AURIX™ TC23x errata sheet

Marking/Step: ES-AC, AC

10506AERRA

About this document

Scope and purpose

This document describes the deviations of the device from the current user documentation, to support the assessment of the effects of these deviations on your custom hardware and software implementations.

Please take note of the following information:

- This errata sheet applies to all temperature and frequency versions and to all memory size variants, unless explicitly noted otherwise. For a derivative synopsis, see the latest datasheet or user manual
- Multiple device variants are covered in this one document. If an issue is related to a particular module, and this module is not specified for a specific device variant, then the issue does not apply to that device variant
  - For example, issues with the identifier "EMEM" (extension memory) do not apply to devices for which no extension memory is specified ("EMEM" is used only as a generic example and may not be a feature of the device that this document covers)
- Devices marked with EES or ES are engineering samples which may not be completely tested in all functional and electrical characteristics and are therefore only suitable for evaluation
  - The specific test conditions for EES and ES are documented in a separate status sheet
- Some of the errata have workarounds which may be supported by the tool vendors. Some corresponding compiler switches may need to be set. Please refer to the respective documentation of your compiler
- To understand the effect of issues relating to the on-chip debug system, please refer to the respective debug tool vendor documentation

Table 1 Current documentation

TC21x/TC22x/TC23x User’s Manual	V1.1	2014-12
TC233/TC234/TC237 AC-Step Data Sheet	V1.0	2017-03
TriCore™ TC1.6P & TC1.6E: • Core Architecture • Instruction Set	 V1.0D10 V1.0D15	 2012-02 2013-07
OCDS User’s Manual (distribution under NDA, only relevant for tool development not for application development)	V2.9.1	2014-11-24

Newer versions replace older versions, unless specifically stated otherwise.

Please always refer to the corresponding documentation for this device available in the category 'Documents' at www.infineon.com/AURIX™ and www.myInfineon.com.

Conventions used in this document

Each erratum identifier follows the pattern [Module]_[Arch].[Type][Number]:

- [Module] = subsystem, peripheral, or function affected by the erratum
- [Arch] = microcontroller architecture where the erratum was initially detected
  - AI = Architecture Independent
  - TC = TriCore™
- [Type] = category of deviation
  - [none] = Functional deviation

  - P = Parametric deviation
  - H = Application hint
- [Number] = ascending sequential number within the three previous fields

***Note**: [Number] As this sequence is used over several derivatives, including already solved deviations, gaps can occur inside this numbering sequence*

Table of contents

1 Errata overview

List of errata referenced in this document.

Table 2 Functional deviations

Issue title	Change	Page
[ADC_AI.016] No Channel Interrupt in Fast Compare Mode with GLOBRES		16
[ADC_TC.068] Effect of VAGND Cross Coupling on Conversion Result		16
[ASCLIN_TC.004] SLSO in SPI mode still active after module disable		19
[ASCLIN_TC.005] Unjustified collision detection error in half-duplex SPI mode		19
[ASCLIN_TC.006] Unjustified response timeout in LIN slave mode		20
[ASCLIN_TC.007] Break Detected in LIN Frames in Soft Suspend mode		20
[ASCLIN_TC.008] Response timeout in LIN Mode in case of header only		20
[ASCLIN_TC.009] RFL flag set in Buffer Mode when Receive FIFO Inlet is disabled		20
[ASCLIN_TC.010] Flush of TXFIFO leads to frame transmission		21
[BROM_TC.008] Sporadic Power-on Reset after Wake-up from Standby Mode		21
[BROM_TC.015] DSPR Data Integrity after Wake-up from Standby Mode		21
[CPU_TC.123] Data Corruption possible when CPU GPR accesses made via SRI slave with CPU running		22
[CPU_TC.127] Pending Interrupt Priority Number PIPN in Register ICR		23
[CPU_TC.132] Unexpected PSW values used upon Fast Interrupt entry		23
[DAP_TC.002] DAP client_blockread has Performance issue in Specific Operation Modes		24
[DAP_TC.003] DAP CRC32 definition and algorithm		24
[DAP_TC.004] DAP client_blockwrite telegram with CRC6 and CRC32 protection options		25
[DAP_TC.005] DAP client_read: dirty bit feature of Cerberus’ Triggered Transfer Mode		26
[DAP_TC.006] CRC6 error in telegram following a get_CRCdown telegram prevents reset of CRC32 calculator		26
[DAP_TC.007] Incomplete client_blockread telegram in DXCM mode when using the “read CRCup” option		27
[DAP_TC.009] CRC6 error in client_blockwrite telegram		27
[DMA_TC.015] DMA Double Buffering: No Timestamp Support		27
[DMA_TC.016] Byte and Half-word Write Accesses to specific Registers not supported		27
[DMA_TC.017] Pattern Detection Double Interrupt Trigger when INTCT = 11_B		28
[DMA_TC.018] FPI timeout can cause pipelined register reads to break		28
[DMA_TC.019] CBS Accesses with Large SPB:SRI Clock Ratios Configured		29
[DMA_TC.020] DMA Conditional Linked List: Circular Buffer Enabled		29
[DMA_TC.021] Combined Software/Hardware Controlled Mode Spurious Errors		29

(table continues...)

Table 2 (continued) Functional deviations

Issue title	Change	Page
[DMA_TC.022] Conditional Linked List: Bus Error		29
[DMA_TC.024] Suspend Request coincident with Channel Activation		30
[DMA_TC.025] Conditional Linked List: new non-CLL mode TCS load can corrupt SDCRC RAM write		30
[DMA_TC.026] Linked List: Failed TCS load can trigger wrap interrupt		30
[DMA_TC.028] Transaction Request Lost (TRL) Interrupt Service Request Behaviour		31
[DMA_TC.031] CHCSR.ICH can be incorrectly set after pattern match		31
[DMA_TC.034] DMA Timestamp and Destination Circular Buffer		31
[DMA_TC.035] Last DMA Transaction in a Linked List triggers a DMA Daisy Chain		32
[DMA_TC.036] Linked List: SADR/DADR can be overwritten when loading a non-LL TCS		33
[DMA_TC.037] Conditional Linked List: Bit TSR.CH not cleared for a CLL transaction upon pattern match		33
[DMA_TC.038] Linked List: SIT interrupt when SIT bit set in newly loaded TCS		33
[DMA_TC.039] Read Data CRC		33
[DMA_TC.040] DMA Linked Lists: Intermittent Clearing of Hardware Transaction Request Enable with mixed mode Transaction Control Sets		34
[DMA_TC.041] DMA Circular Buffer Wrap Interrupt		34
[DMA_TC.042] DMA Interrupt from Channel reported before Completion of DMA Transaction		35
[DMA_TC.043] DMA Write Move Data Corruption for non 32-byte Aligned Cacheable Source Address		35
[DMA_TC.044] Clock Switch after SPB Error Reported results in Spurious SRI Error		36
[DMA_TC.045] DMA Reconfigures DMA Channels Lockup		36
[DMA_TC.046] Shadow Operation Read Only Mode		36
[DMA_TC.049] Bus Error Reported During LL TCS Load		36
[DMA_TC.050] Clearing CHCSR.FROZEN during Double Buffering		37
[DMA_TC.052] SER and DER During Linked List Operations		37
[DMA_TC.053] TS16_ERR Type of Error Reporting Unreliable		37
[DMA_TC.054] DMA Channel Halt Acknowledge Unreliable		38
[DMA_TC.055] ICU to DMA Interface in Sleep Mode		38
[DMA_TC.056] TSR and SUSENR Access Protection Unreliable		38
[DMA_TC.058] Linked List Load Transaction Control Set (TCS) Integrity Error		39
[DMA_TC.061] DMA Double Buffering Operations		40
[DMA_TC.062] Termination of DMA Transaction for Pattern Match		41
[DMA_TC.063] DMA Timestamp Destination Address		42

(table continues...)

Table 2 (continued) Functional deviations

Issue title	Change	Page
[DMA_TC.064] DMA Daisy Chain Request		42
[DMA_TC.065] DMA Move Concurrent Bus Accesses		42
[DMA_TC.066] DMA double buffering operations - Update address pointer		43
[DTS_TC.001] Temperature Sensor Formula		43
[FLASH_TC.052] Use of Write Page Once command		43
[FlexRay_AI.087] After reception of a valid sync frame followed by a valid non-sync frame in the same static slot the received sync frame may be ignored		44
[FlexRay_AI.088] A sequence of received WUS may generate redundant SIR.WUPA/B events		45
[FlexRay_AI.089] Rate correction set to zero in case of SyncCalcResult=MISSING_TERM		45
[FlexRay_AI.090] Flag SFS.MRCS is set erroneously although at least one valid sync frame pair is received		45
[FlexRay_AI.091] Incorrect rate and / or offset correction value if second Secondary Time Reference Point (STRP) coincides with the action point after detection of a valid frame		46
[FlexRay_AI.092] Initial rate correction value of an integrating node is zero if pMicroInitialOffsetA,B = 0x00		46
[FlexRay_AI.093] Acceptance of start-up frames received after reception of more than gSyncNodeMax sync frames		47
[FlexRay_AI.094] Sync frame overflow flag EIR.SFO may be set if slot counter is greater than 1024		47
[FlexRay_AI.095] Register RCV displays wrong value		48
[FlexRay_AI.096] Noise following a dynamic frame that delays idle detection may fail to stop slot		48
[FlexRay_AI.097] Loop back mode operates only at 10 MBit/s		48
[FlexRay_AI.099] Erroneous cycle offset during start-up after abort of start-up or normal operation		49
[FlexRay_AI.100] First WUS following received valid WUP may be ignored		49
[FlexRay_AI.101] READY command accepted in READY state		50
[FlexRay_AI.102] Slot status vPOC!SlotMode is reset immediately when entering HALT state		50
[FlexRay_AI.103] Received messages not stored in Message RAM when in Loop Back Mode		51
[FlexRay_AI.104] Missing start-up frame in cycle 0 at coldstart after FREEZE or READY command		51
[FlexRay_AI.105] RAM select signals of IBF1/IBF2 and OBF1/OBF2 in RAM test mode		52
[FlexRay_AI.106] Data transfer overrun for message transfers Message RAM to Output Buffer (OBF) or from Input Buffer (IBF) to Message RAM		52

(table continues...)

Table 2 (continued) Functional deviations

Issue title	Change	Page
[GTM_AI.132] GTM_TOP level: AEI write to BRIDGE_MODE register can result in blocking of AEI configuration interface		54
[GTM_AI.141] TIM: Incorrect data captured to GPR registers and routed via ARU when EGPRi_SEL,GPRi_SEL= 100 in TIM channel mode TIEM, TPWM, TIPM, TPIM, TGPS		55
[GTM_AI.142] TIM: Incorrect data captured to GPR registers and routed via ARU when EGPRi_SEL,GPRi_SEL= 100 in TIM channel mode TBCM		55
[GTM_AI.143] GTM_TOP level: AEI pipelined write to GTM_BRIDGE_MODE register directly after setting aei_reset='0' can result in blocking of AEI configuration interface		56
[GTM_AI.144] TIM: TIM interrupts as trigger source from TIM to TOM/ATOM not functional		56
[GTM_AI.153] TIM: Incorrect data captured to CNTS register when TIM channel operates in mode TPWM or TPIM and CNTS_SEL = 1 and selected CMU_CLK ≠ sys_clk		57
[GTM_AI.154] TOM: Incorrect duty cycle in PCM mode (bit reversed mode)		57
[GTM_AI.157] CMU: Incorrect AEI status by writing 1 to bit 24 of register CMU_CLK_6/7_CTRL		58
[GTM_AI.163] TIM: timeout signaled when TDU unit is reenabled		58
[GTM_AI.164] TIM: capturing of data into TIM[i]_CH[x]_CNTS with setting CNTS_SEL=1 not functional in TPWM and TPIM mode		59
[GTM_AI.181] TIM: Incorrect signal level bit ECNT[0] in mode TIEM, TPWM, TIPM, TPIM, TGPS		59
[GTM_AI.202] (A)TOM: no CCU1 interrupt in case of CM1=0 or 1 and RST_CCU0=1		59
[GTM_AI.205] TIM: unexpected CNTS register update in TPWM OSM mode		60
[GTM_AI.209] TOM/ATOM: no update of CM0/CM1/CLK_SRC via trigger signal from preceding instance if selected CMU_CLKx is not SYS_CLK		60
[GTM_AI.260] TOM/ATOM: Async. update in SOMP mode with CM1=0 and selected CMU clock unequal sys_clk not functional		61
[GTM_AI.270] (A)TOM: output signal is postponed one period for the values CM0=1 and CM1>CM0 if CN0 is reset by the trigger of a preceding channel (RST_CCU0=1)		61
[GTM_AI.298] TOM/ATOM: wrong output behaviour in SOMP oneshot mode when oneshot pulse is triggered by TIM_EXT_CAPTURE(x)		62
[GTM_AI.299] TOM/ATOM: wrong output behaviour in SOMP oneshot mode when oneshot pulse is triggered by trig_[x-1]		63
[GTM_AI.336] GTM Bus Bridge: Incorrect AEI access execution in case the previous AEI access was aborted with the access timeout abort function		63
[GTM_AI.340] TOM/ATOM: Generation of TRIG_CCU0/TRIG_CCU1 trigger signals skipped in initial phase of A/TOM SOMP one-shot mode		64
[GTM_AI.341] TOM/ATOM: False generation of TRIG_CCU1 trigger signal in SOMP one-shot mode with OSM_TRIG=1 when CM1 is set to value 1		65

(table continues...)

Table 2 (continued) Functional deviations

Issue title	Change	Page
[GTM_AI.347] TOM/ATOM: Reset of (A)TOM[i]_CH[x]_CN0 with TIM_EXT_CAPTURE are not correctly synchronized to selected CMU_CLK/CMU_FXCLK		66
[GTM_AI.361] IRQ: Missing pulse in single-pulse interrupt mode on simultaneous interrupt and clear event		67
[GTM_AI.380] (A)TOM: potentially wrong output signal in case of RST_CCU0=1 and CM0=1 on triggered channel in SOMP mode		67
[GTM_AI.408] (A)TOM-RTL: Missing edge on output signal (A)TOM_OUT when CN0 is reset with force update event		68
[GTM_AI.410] GTM_AEI: The AEI bridge might not execute an accepted write transaction	New	69
[GTM_AI.411] A change of the BRIDGE_MODE register might be delayed indefinitely		69
[GTM_AI.419] TIM: Potentially wrong capture values		70
[GTM_AI.421] GTM_AEI: Changing BRIDGE_MODE.MSK_WR_RSP in pipeline mode can lead to violation of pipeline protocol		71
[GTM_AI.429] TIM: Missing glitch detection interrupt event		72
[GTM_AI.430] TIM: Unexpected increment of filter counter		72
[GTM_AI.431] TIM: Glitch detection interrupt event of filter is not a single cycle pulse		73
[GTM_AI.462] (A)TOM: Missing CCU0TC_IRQ interrupt signal		74
[GTM_AI.487] GTM_AEI: Changing BRIDGE_MODE[2:0] in pipeline mode can lead to violation of pipeline protocol		74
[GTM_AI.488] GTM_AEI: Turning off BRIDGE_MODE.MSK_WR_RSP in asynchronous mode might lead to following transactions being corrupted		75
[GTM_AI.517] (A)TOM: Missing edge on output signal (A)TOM_OUT	New	75
[GTM_AI.522] (A)TOM: Edge at output signal (A)TOM_OUT does not occur	New	76
[GTM_TC.009] TBU signals not wired to debug logic		77
[GTM_TC.012] Read Access Control by Register ODA		77
[IOM_TC.002] Missed or spurious IOM events when pulse length exceeds Event Window counter range		77
[IOM_TC.003] Unexpected Event upon Kernel Reset		78
[IOM_TC.004] Write to IOM register space when IOM_CLC.RMC > 1		78
[MTU_TC.005] Access to MCx_ECCD and MCx_ETRRi while MBIST disabled		78
[MTU_TC.007] Error Overflow Indication ECCD.EOV		79
[MTU_TC.011] MBIST Bitmap not working for w0 - r1		80
[MTU_TC.012] Security of CPU cache memories during runtime is limited		80
[MultiCAN_TC.041] Clock used in Bit Timing Mode	New	80
[MultiCAN_TC.043] CAN FD: Idle Condition		81

(table continues...)

Table 2 (continued) Functional deviations

Issue title	Change	Page
[MultiCAN_TC.044] CAN FD: Missing Hardsync		81
[MultiCAN_TC.045] Simultaneous communication of ISO 11898-1:2015 CAN FD and Non-ISO CAN FD nodes		83
[OCDS_TC.038] Disconnecting a debugger without device reset (“hot detach”) may require reading of OCS registers		83
[OCDS_TC.042] OTGS capture registers can miss single clock cycle triggers		84
[OCDS_TC.043] Read-Modify-Write Bus Transactions to Cerberus Registers		84
[PINNING_TC.001] Port functions and pinning tables for TC234* devices without ETH module in TQFP-144 package - Documentation update		84
[PLL_ERAY_TC.001] PLL_ERAY Initialization after Cold Power-up or Wake-up from Standby mode		85
[PLL_TC.005] PLL Initialization after Cold Power-up or Wake-up from Standby mode		86
[PLL_TC.007] PLL Loss of lock when oscillator shaper is used		87
[PMC_TC.002] Switch Capacitor Regulator Mode, Frequency Spreading - Documentation Update to Register EVRSDCTRL1		87
[PMC_TC.003] Usecase limitation of LDO mode with on chip pass device for SAL devices		88
[QSPI_TC.006] Baud rate error detection in slave mode (error indication in current frame)		88
[QSPI_TC.017] Slave: Reset when receiving an unexpected number of bits		89
[RESET_TC.005] Indication of Power Fail Events in SCU_RSTSTAT		89
[SCU_TC.034] $\overline{\text{TESTMODE}}$ pin shall be held at static high level during LBIST		89
[SMU_TC.006] OCDS Trigger Bus OTGB during Application Reset		89
[SMU_TC.007] Size and Position of Field ACNT in Register SMU_AFCNT		90
[SMU_TC.008] Behavior of Action Counter ACNT		90
[SMU_TC.010] Transfer to SMU_AD register not triggered correctly		91
[SMU_TC.012] Unexpected alarms when registers FSP or RTC are written		91
[SRI_TC.003] XBAR_PRIOL/H Register Layout and Reset Values		92

Table 3 Parametric deviations

Issue title	Change	Page
[ADC_TC.P010] Increased Gain Error (EA_{GAIN}) for T_J < 0°C		95
[IDD_TC.H001] IPC Limits used in Production Test for IDD Max Power Pattern		95
[IEVRSB_TC.P001] Test Condition for I_{EVRSB} (sum of all currents in standby mode) - Data Sheet correction		95
[PADS_TC.H004] PN-Junction Characteristics for Pad Type S		96

(table continues...)

Table 3 (continued) Parametric deviations

Issue title	Change	Page
[RTH_TC.H001] Thermal characteristics of the package - Footnote update for LF-BGA-292-6 package		96
[VDDPPA_TC.H001] Voltage to ensure defined pad states - Footnote update		96

Table 4 Application hints

Issue title	Change	Page
[ADC_AI.H003] Injected conversion may be performed with sample time of aborted conversion		97
[ADC_TC.H011] Bit DCMSB in register GLOBCFG		97
[ADC_TC.H014] VADC Start-up Calibration		98
[ADC_TC.H015] Conversion Time with Broken Wire Detection		98
[ADC_TC.H020] Minimum/Maximum Detection Compares 12 Bits Only		99
[ADC_TC.H022] Sample Time Control - Formula		99
[ADC_TC.H024] Documentation: Filter control only in registers GxRCR7/GxRCR15		100
[ADC_TC.H031] High precision bandgap voltage - documentation update		100
[ADC_TC.H038] Multiplexer Diagnostics Connection - Documentation update		100
[ADC_TC.H041] Offset address of register GxTRCTR - Correction to table “Registers Overview” in User’s Manual		101
[ADC_TC.H042] Precharging of capacitor CAINSW - Documentation update		101
[ASCLIN_TC.H001] Bit field FRAMECON.IDLE in LIN slave tasks		101
[ASCLIN_TC.H003] Behavior of LIN Autobaud Detection Error Flag		102
[ASCLIN_TC.H004] Changing the Transmit FIFO Inlet Width / Receive FIFO Outlet Width		102
[ASCLIN_TC.H005] Collision detection error reported twice in LIN slave mode		103
[ASCLIN_TC.H006] Sample point position when using three samples per bit		103
[ASCLIN_TC.H008] SPI master timing – Additional information to Data Sheet characteristics		103
[BCU_TC.H001] HSM Transaction Information not captured		104
[BROM_TC.H003] Information related to Register FLASH0_PROCOND		104
[BROM_TC.H009] Re-enabling lockstep via BMHD		105
[BROM_TC.H010] Interpretation of value UNIQUE_CHIP_ID_32BIT		105
[BROM_TC.H019] CRC32 ethernet polynomial - Footnote correction		105
[BUS_TC.H001] CPU access latency for TC21x/TC22x/TC23x - Documentation update		105
[BUS_TC.H002] Reset value for register XBAR_IDINTEN - Documentation update		106
[CCU6_AI.H001] Update of Register MCMOUT		106
[CCU6_AI.H002] Description of Bit RWHE in Register ISR		106

(table continues...)

Table 4 (continued) Application hints

Issue title	Change	Page
[CCU6_AI.H003] Bit TRPCTR.TRPM2 in Manual Mode - Documentation Update		107
[CCU_TC.H001] Clock Monitor Check Limit Values		107
[CCU_TC.H002] Oscillator Gain Selection via OSCCON.GAINSEL		108
[CCU_TC.H005] References to f_{PLL2}, f_{PLL2_ERAY} and K3 Divider in User's Manual		108
[CCU_TC.H006] Clock Monitor Support - Documentation Update		109
[CCU_TC.H007] Oscillator Watchdog Trigger Conditions for ALM3[0]		109
[CCU_TC.H010] Oscillator Mode control in register OSCCON - Documentation Update		110
[CPU_TC.H006] Store Buffering in TC1.6/P/E Processors		110
[CPU_TC.H008] Instruction Memory Range Limitations		112
[CPU_TC.H009] Details on CPU Clock Control		112
[CPU_TC.H012] Behavior of bit-wise operations on certain peripheral register bits which need to be written back with the same value		112
[CPU_TC.H014] ACCEN* Protection for Write Access to Safety Protection Registers - Documentation Update		114
[CPU_TC.H015] Register Access Modes for Safety Protection Registers - Documentation Update		114
[CPU_TC.H017] MSUB.Q does not match MUL.Q+SUB - Documentation Update		114
[DAP_TC.H002] DAP client_blockread in Combination with TGIP and all Parcels with CRC6		115
[DAP_TC.H003] Not acknowledged DAP telegrams in noisy environments		115
[DMA_TC.H002] Bit CHCSRz.BUFFER can be toggled when not in Double Buffer Mode		116
[DMA_TC.H004] Transaction Request Lost upon software trigger with pattern match		116
[DMA_TC.H005] Linked List Transfer leading to loading of non-Linked List TCS causes corruption		116
[DMA_TC.H006] Clearing of HTRE when DMA channel is configured for Single Mode		116
[DMA_TC.H007] Selecting the Priority for DMA Channels		117
[DMA_TC.H008] Transaction Request State		118
[DMA_TC.H009] Resetting Bits ICH and IPM in register CHCSRz		118
[DMA_TC.H010] Calculation of DMA Address Checksum for DMA read moves to Cacheable Addresses		118
[DMA_TC.H011] DMA_ADICRz.SHCT - Reserved Values		118
[DMA_TC.H012] TCS Update in Halt State		119
[DMA_TC.H013] MExSR.WS and MExSR.RS Status Bits		119
[DMA_TC.H016] DMARAM ECC Error Disable		119
[DMA_TC.H017] DMA Channel Request Control - Documentation Update		119

(table continues...)

Table 4 (continued) Application hints

(table continues...)

Table 4 (continued) Application hints

Issue title	Change	Page
[IOM_TC.H001] How to clear the IOM_LAMEWCm register		136
[IOM_TC.H002] IOM Clock Control		136
[IOM_TC.H003] Configuration of LAMCFG.IVW and LAMEWS.THR		137
[IOM_TC.H004] Behavior of LAMEWCn.CNT when LAMEWSn.THR is 0		138
[IOM_TC.H006] ACCEN* Protection for Write Access to IOM Registers		138
[IOM_TC.H007] Write Access to FPCESR		138
[LBIST_TC.H004] Update reset behavior of LBISTCTRL2 register - Additional information		138
[LMU_TC.H002] On-the-fly BBB:SRI clock ratio switching		139
[LMU_TC.H003] Function of Bit MEMCON.PMIC (Protection Bit for Memory Integrity Control Bit)		139
[MTU_TC.H003] AURIX™ Memory Tests using the MTU		140
[MTU_TC.H004] Handling the Error Tracking Registers ETRR		140
[MTU_TC.H005] Handling SRAM Alarms		140
[MTU_TC.H006] Alarm Propagation to SMU via Error Flags in MCx_ECCD		141
[MTU_TC.H008] Memory Controllers for DSPR		142
[MTU_TC.H009] Reset Value for Register ECCD		142
[MTU_TC.H010] Register MCONTROL - Bit Field Res4		145
[MTU_TC.H011] Access Protection for Memory Control Registers		145
[MTU_TC.H012] Kernel Reset triggers Reset of MBIST Registers		145
[MTU_TC.H014] Access to SRAM while MTU operations are underway		145
[MultiCAN_AI.H005] TxD Pulse upon short disable request		146
[MultiCAN_AI.H006] Time stamp influenced by resynchronization		146
[MultiCAN_AI.H007] Alert Interrupt Behavior in case of Bus-Off		147
[MultiCAN_TC.H003] Message may be discarded before transmission in STT mode		147
[MultiCAN_TC.H004] Double remote request		147
[MultiCAN_TC.H007] Oscillating CAN Bus may Disable the CAN Interface		148
[MultiCAN_TC.H008] Changes due to CAN FD protocol ISO 11898-1:2015		148
[MultiCAN_TC.H009] Limitation on Secondary Sample Point (SSP) Position (ISO CAN FD nodes only)		151
[MultiCAN_TC.H010] Limitation on maximum SJW Range for CAN FD Data Phase (ISO CAN FD nodes only)		152
[MultiCAN_TC.H011] Transmitter Delay Compensation Behaviour (CAN FD only)		153
[MultiCAN_TC.H012] Delayed time triggered transmission of frames		153
[OCDS_TC.H010] JTAG requires two initial clock cycles after $\overline{\text{PORST}}$		153

(table continues...)

Table 4 (continued) Application hints

Issue title	Change	Page
[OCDS_TC.H012] Minimum Hold Time for Inputs OCDS_TGIx		153
[OCDS_TC.H019] System or Application Reset while OCDS and lockstep monitoring are enabled		154
[PACKAGE_TC.H008] Exposed pad dimensions and package outlines for QFP packages - Updates to TC23x Data Sheet		154
[PLL_ERAY_TC.H002] Correction in Figure "PLL_ERAY Block Diagram"		156
[PMC_TC.H001] Check for permanent Overvoltage during Power-up		156
[PMC_TC.H004] Selecting the WUT Clock Divider		157
[PMS_TC.H002] Sensitivity to supply voltage ripple during start-up		157
[PMS_TC.H008] Interaction of interrupt and power management system - Additional information		158
[PMU_TC.H002] Impact of Application Reset on register FLASH0_FCON		160
[PORTS_TC.H006] Using P33.8 while SMU is disabled		160
[PORTS_TC.H016] Oscillating signal may enable DXCPL and reconfigure the functionality of the port pins P14.0 and P14.1		161
[QSPI_TC.H005] Stopping Transmission in Continuous Mode		161
[QSPI_TC.H006] Corrections to Figures "QSPI - Frequency Domains" and "Phase Duration Control, Overview"		162
[QSPI_TC.H007] RXFIFO Overflow Bit Behavior in Slave Mode		162
[QSPI_TC.H008] Details of the baud rate and phase duration control - Documentation update		162
[QSPI_TC.H009] Dummy frame required after changing SCLK polarity and phase in three wire mode		162
[QSPI_TC.H011] Missing information on SLSI misplaced inactivation enable error	New	163
[RESET_TC.H002] Unexpected SMU Reset Indication in SCU_RSTSTAT		163
[RESET_TC.H003] Usage of the Prolongation Feature for ESR0 as Reset Indicator Output		164
[RESET_TC.H004] Effect of Power-on and System Reset on DSPR		164
[SCU_TC.H009] LBIST Influence on Pad Behavior		164
[SCU_TC.H010] LBIST Signature Depends on Debug Interface Configuration		165
[SCU_TC.H013] Correction to Register References in Chapter "Watchdog Timers"		165
[SCU_TC.H014] Reset Value of Bit Field IOCR.PC1 - Control for Pin $\overline{\text{ESR1}}$		166
[SCU_TC.H027] Bit field INP0 and INP1 in register EICRi - Documentation correction		166
[SCU_TC.H028] ERU configuration changes may lead to ERU reactions	New	167
[SENT_TC.H003] First Write Access to Registers FDR and TPD after ENDINIT Status Change		167

(table continues...)

Table 4 (continued) Application hints

2 Functional deviations

2.1 [ADC_AI.016] No Channel Interrupt in Fast Compare Mode with GLOBRES

Description

In fast compare mode, the compare value is taken from bit-field RESULT of the selected result register and the result of the comparison is stored in the respective bit FCR.

A channel event can be generated when the input becomes higher or lower than the compare value.

In case the global result register GLOBRES is selected, the comparison is executed correctly, the target bit is stored correctly, source events and result events are generated, but a channel event is not generated.

Workaround

If channel events are required, choose a local result register GxRESy for the operation of the fast compare channel.

2.2 [ADC_TC.068] Effect of VAGND Cross Coupling on Conversion Result

Description

Due the implementation of the clock dividers as fractional dividers, a statistical phase shift of one f_{VADC} clock can occur between the operation of different converter groups. If the last f_{VADC} clock of the sample phase of a converter group Gx coincides with the first f_{VADC} clock of a conversion step of (one or more) other converter groups Gy, the Total Unadjusted Error (TUE) of the conversion result of Gx is increased due to cross coupling via VAGND.

Effects

- For TC29x and TC27x, the TUE is increased up to ± 80 LSB_{12}
- For TC26x, TC23x, TC22x, and TC21x, the TUE is increased up to ± 25 LSB_{12}

Workarounds - Introduction

***Note**:*
- *Workaround 1..3 may be used with any device step*
- *Workaround 4 can only be used with TC21x, TC22x, TC23x ≥ step AB, TC26x ≥ step BB, TC27x ≥ step DB, TC29x ≥ step BB.*

  For TC29x ≥ step BB, workaround 4 is the preferred workaround for this device, since for Workaround 1 in particular increased error limits would have to be accepted for high temperature / high pin-count variants.

Workaround 1

Synchronize the trigger events of different converter groups as follows:

Workaround 1: Implementation details for TC27x and TC29x

- Operate the arbiters and the analog parts of the VADC at the same clock frequency, i.e. select the divider factors DIVA and DIVD in register GLOBCFG such that $f_{ADCD} = f_{ADCI}$ for all converter groups:
  - For $f_{VADC} = f_{SPB} = 100$ MHz, reduce f_{ADCD} to 25 MHz (with DIVD = 3), and increase f_{ADCI} to 25 MHz (with DIVA = 3). Depending on supply/reference voltage and junction temperature T_J, this will result in the increased error limits shown in Table 5 (for TC27x in all package variants and for TC297 devices

in BGA-292 package) and Table 6 (for TC298/9 devices in BGA-416/516 package and TC290 bare die variants) below

- Otherwise, to achieve $f_{ADCD} = f_{ADCI}$ = 20 MHz with the error limits specified in the Data Sheet, $f_{VADC} = f_{SPB}$ must be reduced to 80 MHz

- Enlarge the length of an arbitration round to a minimum of 16 arbitration slots (i.e. bit field GxARBCFG.ARBRND ≥ 2 for any x)
- Select the conversion time (including sample time) of the longest conversion of any group Gx to be shorter than two arbitration rounds. This ensures that all converters are idle when the arbiters have determined the next conversion request
- Synchronize the digital and the analog clock by switching off/on the Module Disable Request bit, i.e. set CLC.DISR = 1_B and then CLC.DISR = 0_B
- Initiate the start-up calibration by setting bit GLOBCFG.SUCAL = 1_B (mandatory after switching off/on VADC clocks via CLC.DISR)

Table 5 **TC27x (all packages) and TC297 (in BGA-292 package): Increased VADC Error Limits for f_{ADCI} = 25 MHz, 12-Bit Resolution**

Total Unadjusted Error TUE	DNL Error EA_{DNL}	INL Error EA_{INL}	RMS Noise EN_{RMS}	Unit	Condition	
					VAREF/VDDM	**T_J**
4.4	3.5	3.5	0.95	LSB	*5V ± 10%*	≤ 150°C
5.7	3.5	3.5	0.95			> 150°C
10	12	12	1.5	LSB	3.3 V ± 10 %	≤ 125°C
14	14	16	1.5			*125°C < T_J* ≤ 150°C
This workaround is not recommended for T_J > 150°C and VAREF/ VDDM = 3.3 V ± 10 %						> 150°C

Table 6 **TC298/9 in BGA-416/516 Package and TC290 bare die: Increased VADC Error Limits for f_{ADCI} = 25 MHz, 12-Bit Resolution**

Total Unadjusted Error TUE	DNL Error EA_{DNL}	INL Error EA_{INL}	RMS Noise EN_{RMS}	Unit	Condition	
					VAREF/VDDM	**T_J**
6.4	6.0	5.4	0.95	LSB	*5V ± 10%*	≤ 150°C
8.4	7.9	7.1	0.95			> 150°C
10	12	12	1.5	LSB	3.3 V ± 10 %	≤ 125°C
14	14	16	1.5			*125°C < T_J* ≤ 150°C
This workaround is not recommended for T_J > 150°C and VAREF/ VDDM = 3.3 V ± 10 %						> 150°C

***Note**: For error types (offset, gain, ..) not listed in the table above see corresponding values specified in the Data Sheet for f_{ADCI} = 20 MHz.*

***Note**: For available combinations of package type and temperature range see the corresponding AURIX™ TC2x Variants Data Sheet Addendum.*

Workaround 1: Implementation details for TC26x and TC23x, TC22x, TC21x

- Operate the arbiters and the analog parts of the VADC at the same clock frequency, i.e. select the divider factors DIVA and DIVD in register GLOBCFG such that $f_{ADCD} = f_{ADCI}$ for all converter groups:
  - Note: As $f_{ADCD} = f_{VADC}/4$ with the maximum divider (DIVD = 3), this implies that $f_{VADC} = f_{SPB}$ must be limited to 80 MHz to achieve $f_{ADCD} = f_{ADCI}$ with the error limits specified for $f_{ADCI} = 20$ MHz in the Data Sheet
- Enlarge the length of an arbitration round to a minimum of 16 arbitration slots (i.e. bit field GxARBCFG.ARBRND ≥ 2 for any x)
- Select the conversion time (including sample time) of the longest conversion of any group Gx to be shorter than two arbitration rounds. This ensures that all converters are idle when the arbiters have determined the next conversion request
- Synchronize the digital and the analog clock by switching off/on the Module Disable Request bit, i.e. set CLC.DISR = 1_B and then CLC.DISR = 0_B
- Initiate the start-up calibration by setting bit GLOBCFG.SUCAL = 1_B (mandatory after switching off/on VADC clocks via CLC.DISR)

Workaround 2

Ensure that conversions never overlap for any two converter groups Gx and Gy.

This may be achieved under software control, or by exclusively using the VADC background request source.

For this workaround, no restrictions apply on clock and arbitration round settings.

Workaround 3

Use the converters within a synchronization group in master/slave configuration, such that they are synchronized for parallel sampling, triggered by one common master. In this case, the cross coupling effect will not occur as long as only one synchronization group is performing conversions.

For devices that support more than one synchronization group, operate the synchronization groups in an interleaving manner.

For this workaround, no restrictions apply on clock and arbitration round settings.

Workaround 4

To avoid the cross coupling effect, this device step (see “Workarounds - Introduction” above) supports selection of signal CCU6061_TRIG1 to synchronize the start of the converter groups to a raster of $1/f_{ADCI}$ (e.g. $5/f_{SPB} = 50$ ns @f_{SPB} = 100 MHz and $f_{ADCI} = 20$ MHz, or $4/f_{SPB} = 64$ ns @$f_{SPB} = 62.5$ MHz and $f_{ADCI} = 12.5$ MHz). The resulting jitter (delay from trigger to start of conversion) is thus limited to max. $1/f_{ADCI}$.

For this workaround, either CCU60_T13 or CCU61_T13 is configured (reserved) to provide the synchronization signal. The selection is performed via bit field TRIG1SEL in register CCU60_MOSEL:

- TRIG1SEL = 000_B: signal CCU60_COUT63 from CCU60_T13 is selected
- TRIG1SEL = 001_B: signal CCU61_COUT63 from CCU61_T13 is selected

The synchronization signal is enabled inside the VADC module by setting bit GLOBCFG.DCMSB = 1_B. The default function of this bit (DCMSB = 0_B: one clock cycle for MSB conversion step) is hardwired and thus stays unaffected.

The following examples describe the initialization of CCU60 or CCU61, respectively, to provide a 20 MHz synchronization signal @f_{SPB} = 100 MHz:

Example for CCU60 initialization

```
CCU60_CLC = 0x0;          // enable CCU60 kernel
CCU60_T13PR = 0x4;        // 4+1 clock periods with ..
CCU60_CC63SR = 0x1;       // duty cycle 40 ns low / 10 ns high
CCU60_PSLR |= 0x0080;     // passive state level of COUT63 = 1
CCU60_MODCTR |= 0x8000;   // ECT130 = 1 enables T13 output (CC63ST -> COUT63)
CCU60_TCTR4 |= 0x4200;    // set bit T13STR and T13RS to enable shadow transfer and start T13
CCU60_MOSEL &= 0x1C7;     // CCU6061_TRIG1 is CCU60_COUT63
```

Example for CCU61 initialization

***Note**: In case an application only uses kernel CCU61, ensure that kernel CCU60 is also clocked until register CCU60_MOSEL is configured.*

```
CCU61_CLC = 0x0;          // enable CCU61 kernel
CCU60_CLC = 0x0;          // ensure CCU60 kernel is clocked until CCU60_MOSEL is configured
CCU61_T13PR = 0x4;        // 4+1 clock periods with ..
CCU61_CC63SR = 0x1;       // duty cycle 40 ns low / 10 ns high
CCU61_PSLR |= 0x0080;     // passive state level of COUT63 = 1
CCU61_MODCTR |= 0x8000;   // ECT130 = 1 enables T13 output (CC63ST -> COUT63)
CCU61_TCTR4 |= 0x4200;    // set bit T13STR and T13RS to enable shadow transfer and start T13
CCU60_MOSEL |= 0x8;       // CCU6061_TRIG1 is CCU61_COUT63
```

2.3 [ASCLIN_TC.004] SLSO in SPI mode still active after module disable

Description

It is expected that in SPI mode, after module disable, the Slave Select Output signal SLSO should be in idle state according to configuration of Slave Polarity in Synchronous mode (IOCR.SPOL).

However, in this design step, when the module is disabled, the Slave Select Output signal SLSO is always 0 (low) independent of IOCR.SPOL, i.e., it is still active even when IOCR.SPOL = 1_B.

Workaround

Before disabling the ASCLIN module, set SLSO to the desired level in the corresponding Port control registers.

2.4 [ASCLIN_TC.005] Unjustified collision detection error in half-duplex SPI mode

Description

In Half Duplex SPI mode, when collision detection is enabled and the number of stop bits in SPI frame is configured as any value from 1 to 7 in FRAMECON.STOP, a Collision Error (FLAGS.CE) is triggered during the trailing phase (i.e., during stop bits), although RX and TX signal are identical.

Workaround

In half-duplex SPI mode, set FRAMECON.STOP = 0 if trailing phase is irrelevant, or ignore/disable collision error if FRAMECON.STOP > 0.

2.5 [ASCLIN_TC.006] Unjustified response timeout in LIN slave mode

Description

When ASCLIN is configured as LIN slave and Response timeout is configured as DATCON.RM = 1_B, Response timeout is triggered even when an incomplete LIN Header frame is received. The timeout counter runs further after Header timeout detection without reset and triggers Response Timeout when it reaches the Response Timeout Threshold value defined by DATCON.RESPONSE.

Workaround

Ignore the Response Timeout which comes directly after a Header Timeout has occurred and before the next break is detected.

2.6 [ASCLIN_TC.007] Break Detected in LIN Frames in Soft Suspend mode

Description

When ASCLIN has entered Soft Suspend mode (OCS.SUS = 0x2), it still detects a Break Field in LIN frames and triggers an interrupt if enabled (FLAGSENABLE.BDE = 1_B).

Workaround

Ignore a detected break event when the module has been soft-suspended (for example set FLAGSENABLE.BDE = 0_B when using soft suspend mode).

2.7 [ASCLIN_TC.008] Response timeout in LIN Mode in case of header only

Description

In LIN (Master/Slave) mode, when Header Only (DATCON.HO = 1_B) is configured, Response timeout could occur even though no Response frame is expected.

Workaround

To avoid the unwanted interrupt, disable the interrupt on Response Timeout by FLAGSENABLE.RTE = 0_B whenever Header Only (DATCON.HO = 1_B) is configured.

2.8 [ASCLIN_TC.009] RFL flag set in Buffer Mode when Receive FIFO Inlet is disabled

Description

When RXFIFO is configured in Buffer Mode (RXFIFOCON.BUF = 1_B) and Receive FIFO Inlet is disabled (RXFIFOCON.ENI = 0_B), the receive FIFO level flag is set (FLAGS.RFL = 1_B) even though RXFIFO is not filled with new incoming data.

Workaround

To avoid the unwanted Receive FIFO Level interrupt, disable it by setting FLAGSENABLE.RFLE = 0_B whenever Receive FIFO Inlet is disabled (RXFIFOCON.ENI = 0_B),

2.9 [ASCLIN_TC.010] Flush of TXFIFO leads to frame transmission

Description

When the TXFIFO is flushed (TXFIFIOCON.FLUSH = 1_B), it triggers transmission of a frame in the following corner case:

- Starting condition:
  - TXFIFO is not empty and TXFIFOCON.ENO = 0_B
- Triggering condition:
  - Write to TXFIFOCON with both TXFIFOCON.FLUSH = 1_B and TXFIFOCON.ENO = 1_B

Workaround

Do not flush TXFIFO and change bit TXFIFOCON.ENO from 0_B to 1_B in one single write to TXFIFOCON if TXFIFO is not empty.

2.10 [BROM_TC.008] Sporadic Power-on Reset after Wake-up from Standby Mode

Description

On a wake-up from Standby mode, the Standby RAM redundancy installation procedure is executed. In case there is a sporadic Power-on reset in a time window between 600 µs - 1 ms after Standby mode wake-up, it can happen that the application data stored in specific Standby RAM cells are overwritten.

***Note**: This effect can occur only on devices where non-zero data are stored in CPU0 DSPR at locations D000 2000$_H$ to D000 203F$_H$ by the Startup Software (SSW) after cold power-on.*

For TC26x and TC27x B-step, see for example section “Entering Standby Mode ..” in the PMC chapter within chapter “System Control Units” of the corresponding User’s Manual.

For all other TC2xx devices, see for example section “Preparation before to enter Stand-by mode” in the BootROM chapter of the corresponding User’s Manual.

Only CPU0 DSPR Standby RAM is affected, EMEM in ADAS or ED devices is not affected.

Workarounds

1. Calculate CRC over critical Standby RAM data and store result before Standby mode entry. On a consequent wake-up, CRC of the critical data shall be carried out. The CRC is a general recommended measure for improved robustness of Standby RAM handling.

   Or / and
2. Keep a copy of the critical data at a second location in Standby RAM. On wake-up, compare data from both locations to ascertain their integrity

2.11 [BROM_TC.015] DSPR Data Integrity after Wake-up from Standby Mode

Description

The CPU0 DSPR memory can be used as Standby RAM to save data required after resuming from standby mode.

Applications using DSPR powered in standby mode (to save data) may face data loss within DSPR after wake-up on TC23x product variants.

***Note**: The following types of applications are NOT affected:*

- *Applications which do not power DSPR during standby mode*
- *Applications initializing the entire used DSPR address range with 32-bit word accesses after PORST and after wake-up from standby mode*
- *Applications which do not use the standby feature at all*

Detailed Problem Description

AURIX™ TC2xx devices use repair cells to substitute defective SRAM cells in the DSPR address range. Repair cells are not powered during standby mode and therefore will lose data.

After cold power-on reset, the start-up software (SSW) creates a list of repaired DSPR addresses which is used to restore data to repair cells after wake-up from standby mode (see chapter “RAM overwrite during start-up” in the BootROM chapter of the User’s Manual).

On affected devices, the list of repaired RAM addresses is not created correctly, with the following consequences:

- After wake-up from standby mode, read accesses to repaired addresses may deliver incorrect content, and in most cases there will be a bus error if the V_{DDP3} voltage domain is unpowered during standby mode
- Some RAM locations - incorrectly treated by the SSW as defective - may be overwritten, or the previous contents (before standby mode entry) may be lost, respectively

Scope

The failure rate depends on the application impact of up to 16 corrupted data words which are randomly distributed in the DSPR address range. About 1% of the devices shipped use between 1 and 16 repaired words in the DSPR address range and are therefore affected.

Identification of affected devices

Affected applications can recognize the problem using the following sequence:

- Calculate a checksum on the DSPR address range (used by the application) and store it in DSPR before entering standby mode
- After wake-up, perform a plausibility check (re-calculate checksum and compare it with stored checksum value)
- If the plausibility check passes: the device is not affected by the problem

Workaround

None

2.12 [CPU_TC.123] Data Corruption possible when CPU GPR accesses made via SRI slave with CPU running

Description

Data corruption may occur when another master accesses a TriCore CPU’s General Purpose Registers (GPRs) via its SRI slave port whilst the CPU is running (i.e. not Idle, Halted or Suspended). The TriCore GPRs are A0-A15 and D0-D15. The scenarios in which data corruption may occur are different for the TC1.6P and TC1.6E processors as described below.

TC1.6P - Data corruption may occur when one of the CPU GPRs is **written** via the SRI slave port whilst the CPU is running. Both AGPR and DGPR writes may be affected.

TC1.6E - Data corruption may occur when one of the CPU Address GPRs (A0-A15) is **read** via the SRI slave port whilst the CPU is running. However, data corruption can only occur when the slave AGPR read interacts with the execution of a specific form of store instruction. The store instructions affected by this issue are ST.A and ST.DA, where the address register to be stored is modified by the addressing mode of the store instruction. For example:

```
ST.A [+A0], A0
```

However, such store instructions are architecturally undefined and should not be being used. In the case of this errata all data written to memory by this store instruction may be corrupted.

Workaround

Writes to a CPU's GPRs via its SRI slave port must never be performed whilst the CPU is running. If it is necessary for an external master to write to a CPU's GPR then that CPU must first be placed in Idle, Halt or Suspend mode.

If it is necessary for an external master to read a TC1.6E CPU's AGPR whilst that CPU is running then store instructions of the form above (where any source register is modified by the addressing mode of the store instruction) are not allowed.

2.13 [CPU_TC.127] Pending Interrupt Priority Number PIPN in Register ICR

Description

In the TriCore™ Architecture Manual, it is described for the Pending Interrupt Priority Number ICR.PIPN that it is reset to 0x0 in case there is no request pending.

However, the AURIX™ hardware implementation behaves differently, as the value of PIPN is not changed after the interrupt is serviced in case there is no further request pending.

2.14 [CPU_TC.132] Unexpected PSW values used upon Fast Interrupt entry

Description

Under certain conditions, unexpected PSW values may be used during the first instructions of an interrupt handler, if the interrupt has been taken as a fast interrupt. For a description of fast interrupts, see the "CPU Implementation-Specific Features" section of the relevant User's Manual.

When the problem occurs, the first instructions of the interrupt handler may be executed using the PSW state from the end of the previous exception handler, rather than that which is being loaded by the fast interrupt entry sequence. The TC1.6E, TC1.6P and TC1.6.2P processors are all affected by this problem as follows:

- TC1.6E (in TC21x..TC27x):
  - Only the first instruction of the ISR is affected
- TC1.6P (in TC26x..TC29x), TC1.6.2P (in TC3xx):
  - Up to 4 instructions at the start of the ISR may be affected.
  - However, if the following pre-condition is not met, then there is no issue for these processor variants:

    A11 must point to the first instruction of the fast interrupt handler at the end of the previous exception handler, i.e. the return value from the previous exception must be pointing to the very first instruction of the new interrupt handler. Note that this case should not occur normally, unless software updates the A11 register to a value corresponding to the start of an interrupt handler

Workaround 1

When the PSW fields PSW.PRS, PSW. S, PSW.IO or PSW.GW need to be changed in an exception handler, the change should be wrapped in a function call.

```
_exception_handler:
  CALL _common_handler
  RFE

_common_handler:
  MOV.U d0, #0x0380
  MTCR #(PSW), d0 // PSW.IO updated to User-0 mode
  ...
  RET
```

Note that this workaround assumes SYSCON.TS == SYSCON.IS such that the workaround functions correctly for both traps and interrupts. If this is not the case it is possible for bus accesses to use an incorrect master Tag ID, potentially resulting in an access to be incorrectly allowed, or an unexpected alarm to be generated. In this case it should be ensured that for all interrupt handlers the potentially affected instructions do not produce bus accesses.

Workaround 2

Do not use any instructions dependent upon PSW settings (for example BISR or ENABLE, dependent on PSW.IO) as the first instruction of an ISR in TC1.6E, or as one of the first 4 instructions in an ISR for TC1.6P or TC1.6.2P.

***Note**: The workarounds need to be applied in TC1.6P and TC1.6.2P only in case software modifies the A11 register in an exception handler, as described in the pre-conditions above.*

2.15 [DAP_TC.002] DAP client_blockread has Performance issue in Specific Operation Modes

Description

For achieving the highest block read bandwidth, the following word is already read chip internally while a word is transmitted on DAP. This read ahead is under certain conditions disabled in the case that the “All parcels with CRC6” bit is set in the telegram. In this case the distance between the reply parcels becomes significantly longer, due to the missing read ahead. This effect occurs also in Wide Mode.

The data values in the parcels are always correct, it is just a performance issue.

Workaround

Don’t use the “All parcels with CRC6” option, use “Read CRCup” instead.

This mode is anyway better in terms of performance for larger blocks (no CRC6 overhead for each parcel) and data protection (32 bit CRC). For a few words, the impact of this performance issue might be tolerable. For the first word a read ahead is not possible anyway.

2.16 [DAP_TC.003] DAP CRC32 definition and algorithm

Description

The DAP CRC32 algorithm is different from the IEEE 802.3 Ethernet CRC.

Workaround

Use the following (VHDL) algorithm for each incoming data bit. The CRC32 value is initialized with all ones. In Wide Mode the function is called for both DAP data bits in each DAP0 clock cycle.

```
subtype  crc32_t is std_ulogic_vector(31 downto 0);
function calc_crc32_f(crc_now : crc32_t;
                      bit_new : std_ulogic)
                     return crc32_t is
  variable crc : crc32_t;
begin
  crc(31 downto 1) := crc_now(30 downto 0);
  crc(0)  := bit_new xor crc_now(31);
  crc(1)  := bit_new xor crc_now(0)  xor crc_now(31);
  crc(2)  := bit_new xor crc_now(1)  xor crc_now(31);
  crc(4)  := bit_new xor crc_now(3)  xor crc_now(31);
  crc(5)  := bit_new xor crc_now(4)  xor crc_now(31);
  crc(7)  := bit_new xor crc_now(6)  xor crc_now(31);
  crc(8)  := bit_new xor crc_now(7)  xor crc_now(31);
  crc(10) := bit_new xor crc_now(9)  xor crc_now(31);
  crc(11) := bit_new xor crc_now(10) xor crc_now(31);
  crc(12) := bit_new xor crc_now(11) xor crc_now(31);
  crc(16) := bit_new xor crc_now(15) xor crc_now(31);
  crc(22) := bit_new xor crc_now(21) xor crc_now(31);
  crc(23) := bit_new xor crc_now(22) xor crc_now(31);
  crc(26) := bit_new xor crc_now(25) xor crc_now(31);
  return crc;
end calc_crc32_f;
```

2.17 [DAP_TC.004] DAP client_blockwrite telegram with CRC6 and CRC32 protection options

Description

***Note**: This problem is only relevant for tool development, not for application development.*

When issuing a DAP client_blockwrite telegram from the tool to the device several CRC protection options are available, namely CRC6 and CRC32.

Expected Behavior

- For CRC6 the expected behavior is:
  - (1) A CRC6 will be appended to the reply of only the last parcel of the telegram
  - (2) An optional CRC6 can be appended to the devices “single startbit response” by setting DAPISC.RC6
- For CRC32 the expected behavior is:
  - (3) The telegram can optionally send the CRCdown value as the last parcel

Actual Implementation

- For the actual implementation the CRC6 slightly differs as follows:
  - (1) The CRC6 of the last parcel will be erroneous if DAPISC.RC6 is set or if the CRCdown option is enabled
  - (2) If DAPISC.RC6 = 1_B, an unintentional CRC6 will be appended to the device response of parcels which are not the last parcel
- For the actual implementation the CRC32 option slightly differs as follows:
  - (3) If also the CRC6option is set, the CRCdown option will not return the correct CRCdown value

Workaround for (3)

Workaround for (3) is not to use the CRCdown feature of the client_blockwrite telegram, but to use the dedicated get_CRCdown telegram.

2.18 [DAP_TC.005] DAP client_read: dirty bit feature of Cerberus' Triggered Transfer Mode

Description

***Note**: This problem is only relevant for tool development, not for application development.*

The DAP telegram client_read reads a certain number of bits from an IOclient (for example Cerberus). The parameter k can be selected to be zero, which is supposed to activate reading of 32 bits plus dirty bit.

However, in the current implementation, the dirty bit feature does not work correctly.

It is recommended not to use this dirty bit feature, meaning the number k should not evaluate to "0".

2.19 [DAP_TC.006] CRC6 error in telegram following a get_CRCdown telegram prevents reset of CRC32 calculator

Description

***Note**: This problem is only relevant for tool development, not for application development.*

If a CRC6 error occurs in the telegram following a get_CRCdown telegram the AURIX™ internal CRC32 calculator does not get reset, as is the expected behavior for get_CRCdown.

This effect can lead to unexpected CRC32 values for the next get_CRCdown telegram. This corresponds to the perception of the tool that there has been a CRC32 error, even if the data was transmitted correctly.

Workaround 1

Accept extra traffic for a required retransmission : In this case the tool could see a CRC32 error which is not based on a wrong transmission, but on the missing reset of the AURIX™ internal CRC32 calculator. This would trigger the retransmission of correctly sent data.

Workaround 2

Check for no-reply after a get_CRCdown telegram : If the tool does not receive an answer for the telegram following a get_CRCdown, it needs to re-send the get_CRCdown telegram and ignore the data.

2.20 [DAP_TC.007] Incomplete client_blockread telegram in DXCM mode when using the “read CRCup” option

Description

In DXCM (DAP over CAN Messages) mode, the last parcel containing the CRC32 might be skipped in a client_blockread telegram using the “read CRCup” option.

Workaround

Do not use CRCup option with client_blockread telegrams in DXCM mode.

Instead the CRCup can be read by a dedicated getCRCup telegram.

2.21 [DAP_TC.009] CRC6 error in client_blockwrite telegram

Description

***Note**: This problem is only relevant for tool development, not for application development.*

If a CRC6 error happens in a client_blockwrite telegram, the DAP module will not execute the write and the tool will run into timeout according to the DAP protocol.

But in this case a following client_blockwrite (with start address) will be ignored by the DAP module.

Workaround

If the tool is running into a timeout after a client_blockwrite telegram it should transmit a dummy client_blockread telegram (for example len=0, arbitrary address) which will clean up the DAP client_blockwrite function.

2.22 [DMA_TC.015] DMA Double Buffering: No Timestamp Support

Description

When a DMA channel is configured for DMA Double Buffering, and flow control (or appendage of time stamp) is selected, i.e. DMA_ADICRz.STAMP = 1_B, the Move Engine may lock up.

Workaround

When a DMA channel is configured for DMA Double Buffering then flow control (or appendage of time stamp) should not be selected, i.e. bit DMA_ADICRz.STAMP must be = 0_B.

2.23 [DMA_TC.016] Byte and Half-word Write Accesses to specific Registers not supported

Description

***Note**: This erratum might affect the SFR C Header Definitions. In such cases, SFR usage in the software shall be analyzed within the applications for their correct handling.*

Byte and half-word write accesses via the SPB (System Peripheral Bus) to the Regfile and Request Control logic are not supported.

This affects the following registers:

- DMA_OTSS (OCDS Trigger Set Select)
- DMA_ERRINTR (Error Interrupt)

- DMA_PRR0 (Pattern Read Register 0)
- DMA_PRR1 (Pattern Read Register 1)
- DMA_MODEy (Hardware Resource Mode)
- DMA_HRRz (Hardware Resource Partition)
- DMA_SUSENRz (Channel Suspend Enable)
- DMA_TSRz (Transaction State)

Workaround

Make sure only 32-bit word data is written to the registers listed above by selecting the appropriate data types.

2.24 [DMA_TC.017] Pattern Detection Double Interrupt Trigger when INTCT = 11_B

Description

A DMA channel z is configured for pattern detection by programming the DMA_CHCFGRz.PATSEL to reference a data value set in one of the pattern read registers DMA_PRR0 or DMA_PRR1. If DMA_ADICRz.INTCT = 11_B then DMA channel z will generate a channel interrupt trigger and set CHSRz.ICH each time TCOUNT is decremented.

If a pattern match is detected then a channel interrupt trigger will be correctly generated but a second channel interrupt trigger will be generated when TCOUNT decrements. The second interrupt trigger is a bug and should not occur.

If the DMA channel z interrupt trigger is directed via the Interrupt Router to generate a DMA hardware request to another DMA channel then the second interrupt trigger may result in a Transaction Request Lost event.

Workaround

Workaround is to ignore the generation of the Transaction Request Lost event:

- Either disable the generation of error interrupt service requests by setting ADICRz.ETRL = 0_B
- Or if the error interrupt service request is enabled, check all error status bits. If only the TRL bit for DMA channel x (pattern detection channel) is set then clear TRL and continue normal DMA operation

2.25 [DMA_TC.018] FPI timeout can cause pipelined register reads to break

Description

Due to a problem in the FPI slave interface (SIF) to the System Peripheral Bus (SPB) in the DMA module, a register access which is pipelined behind an access which is timed-out may terminate early and return the wrong data to the bus.

The scenario for this problem to occur is as follows:

1. An FPI read transaction is performed which takes a long time in the data phase. Pipelined behind this is a register access to DMA or Cerberus
2. The first transaction is timed out, and in the same cycle the register access is taken by the SIF

Workaround

Timeout indicates a severe problem, meaning that something took unexpectedly long. In the event of an FPI timeout on the SPB, an error routine should be run to determine the error, and perform a system reset.

2.26 [DMA_TC.019] CBS Accesses with Large SPB:SRI Clock Ratios Configured

Description

When operating in debug mode and a large SPB:SRI clock ratio is configured then Cerberus accesses to the SRI address space may be unreliable and result in the Cerberus hanging.

Workaround

Limit the SPB:SRI clock ratio to 1:1, 2:1, 3:1 or 4:1, and do not perform Cerberus accesses to the SRI address space while switching the SPB:SRI clock ratio.

2.27 [DMA_TC.020] DMA Conditional Linked List: Circular Buffer Enabled

Description

When a DMA channel is configured for Conditional Linked List (i.e. ADICRz.SHCT = 1111_B) and circular buffer operation (i.e. ADICRz.SCBE = 1_B OR ADICRx.DCBE = 1_B) then if the source and destination addresses are not set to wrap boundaries then the behavior will not be as intended, for example the wrap bits CHCSRz.WRPS and CHCESRz.WRPD may be spuriously set.

Workaround

If a DMA channel is configured for Conditional Linked List and circular buffers are enabled then the user must set the source and destination addresses to wrap boundaries.

2.28 [DMA_TC.021] Combined Software/Hardware Controlled Mode Spurious Errors

Description

A DMA channel is configured for combined software/hardware controlled mode. If the Move Engine is servicing a DMA channel software request and a DMA channel hardware trigger is received then a Transaction Request Lost event is set. When the Move Engine completes the current DMA access the TSRz.CH bit is not cleared. The DMA channel will continue to request channel arbitration as the CH bit is set. If the DMA channel wins arbitration then the Move Engine will continue to service the DMA channel.

In summary, 2 DMA requests (software and hardware) have resulted in 2 X DMA transfers and 1 X Transaction Request Lost (i.e. 3 X DMA actions for 2 X DMA triggers) i.e. a spurious error is generated.

Workaround

If a DMA channel is configured for combined software/hardware mode then increased attention must be paid to de-conflict the triggering of DMA channels from the servicing of DMA requests. The workaround will remove the source of spurious errors.

2.29 [DMA_TC.022] Conditional Linked List: Bus Error

Description

When a DMA channel is configured for Conditional Linked List (i.e. ADICRz.SHCT = 1111_B) then if a bus error is reported then:

- If there is a pattern match then the number of DMA moves subsequently executed may not be as intended
- If there is an error during the loading of a new Transaction Control Set then the DMA channel does not clear the TSRz.CH bit and begins the next DMA transaction with an erroneous Transaction Control Set

Workaround

If a DMA channel is configured for Conditional Linked List then the user must enable the error interrupt service request. On receiving notification of an error interrupt service request the user must read the Move Engine Error Status Registers to confirm that no bus errors were reported:

- If DMA_ERRSRx.DER = 0_B and DMA_ERRSRx.SER = 0_B then no bus errors reported
- If a bus error is reported then check the last error channel DMA_ERRSRx.LEC
- If DMA_ERRSRx.DLLER = 1_B then there was an error during the loading of a new Transaction Control Set

2.30 [DMA_TC.024] Suspend Request coincident with Channel Activation

Description

If DMA channel z is suspend enabled (SUSENRz.SUSEN = 1_B) and the DMA receives a suspend request then if during the same clock cycle the DMA channel becomes active in a Move Engine, the following effects will occur:

- SUSACRz.SUSAC is set for a cycle and then cleared
- A DMA transfer is performed for DMA channel z
- SUSACRz.SUSAC is set again on completion of the DMA transfer and the DMA channel is finally suspended

Workaround

When polling SUSACRz.SUSAC in software, additionally check whether DMA channel z is active in a Move Engine x by reading bit-field MExSR.CH.

2.31 [DMA_TC.025] Conditional Linked List: new non-CLL mode TCS load can corrupt SDCRC RAM write

Description

When a Conditional Linked List (CLL) transaction is running and gets a CLL pattern match, this will stop the running transaction and cause a transaction control set (TCS) load.

In case the new TCS load is set up so that it is not in CLL mode, then the SDCRC value of the new TCS may get corrupted.

Workaround

Avoid selection of non-CLL mode in the TCS loaded after a CLL pattern match.

2.32 [DMA_TC.026] Linked List: Failed TCS load can trigger wrap interrupt

Description

When a Transaction Control Set (TCS) linked list load is performed, and an error is received during the load process, this terminates the load. A DMA linked list error is indicated by the error status flag ERRSRx.DLLER.

If the DADR address left in the register matches the destination wrap boundary, this results in the issuing of a destination wrap interrupt in case the destination wrap interrupt enable is set. Hence a failed TCS load has triggered an interrupt.

Note: *This only happens for destination interrupts. Logic is already in place to exclude source interrupts.*

Workaround

An error interrupt for the DMA linked list error is triggered by the status flag ERRSRx.DLLER if enabled by EERx.ELER. Therefore the destination wrap buffer interrupt can be ignored in this case.

2.33 [DMA_TC.028] Transaction Request Lost (TRL) Interrupt Service Request Behaviour

Description

The DMA channel TRL error interrupt service request is a DMA safety measure signalling a lost DMA request to the system. For each DMA channel TRL event, the DMA may trigger one or more error interrupt service requests.

The application software should include a DMA error handler to resolve all DMA errors including TRL.

Workaround

None.

2.34 [DMA_TC.031] CHCSR.ICH can be incorrectly set after pattern match

Description

If a pattern match is seen during a transaction, the transaction is halted for the current active channel. The move engine zeroes its internal move counter, and holds the transfer count status MEx_CHCSR_TCOUNT at the last value. However, the MEx_CHCSR.ICH bit will still be set indicating a TCOUNT decrement.

Workaround

As there is a pattern match, a DMA channel pattern match interrupt service request will be generated. The pattern match interrupt routine can service the interrupt and clear the status bits including ICH.

2.35 [DMA_TC.034] DMA Timestamp and Destination Circular Buffer

Description

The DMA must not write a DMA timestamp at an address that overwrites DMA move data stored at a DMA destination address. If the DMA channel is configured for linear DMA destination address generation (DMA channel ADICRz.DCBE = 0_B), the DMA appends the DMA timestamp to the end of a DMA transaction (i.e. beyond the last DMA write move data).

If the DMA channel is configured for destination circular buffer (DMA channel ADICRz.DCBE = 1_B), there are three use cases:

- **Use Case 1:** the size of the DMA transaction **equals** the size of the destination circular buffer. If the DMA writes the last DMA write move data at the last address in the destination circular buffer, the DMA correctly writes the DMA timestamp beyond the destination circular buffer
- **Use Case 2:** the size of the DMA transaction is **less than** the size of the destination circular buffer. If the DMA writes the last DMA write move data NOT at the last address in the destination circular buffer, the DMA writes the DMA timestamp inside the destination circular buffer. Erroneously, the DMA may store the DMA timestamp at an address that overwrites DMA write move data
- **Use Case 3:** the size of the DMA transaction is **greater than** the size of the destination circular buffer. After the DMA destination address has wrapped, the DMA will overwrite DMA write move data with fresh DMA write move data

***Note**: DMA Timestamp works as specified when using only source circular buffer.*

Workaround 1

If a DMA channel is configured

- for destination circular buffering (ADICRz.DCBE = 1_B) AND
- the appendage of a DMA timestamp (ADICRz.STAMP =1_B), AND
- the size of the DMA transaction (defined by CFCFGRz.TREL) **equals** the size of the destination circular buffer (defined by ADICRz.CBLD)

the DMA shall append the DMA timestamp beyond the destination circular buffer if

- For increment of DMA destination address (ADICRz.INCD = 1_B), the initial DMA destination address is at the bottom of the destination circular buffer
- For decrement of DMA destination address (ADICRz.INCD = 0_B), the initial DMA destination address is at the top of the destination circular buffer

Workaround 2

If DMA channel z is configured

- for destination circular buffering (ADICRz.DCBE = 1_B) AND
- increment of DMA destination address (ADICRz.INCD = 1_B) AND
- the appendage of a DMA timestamp (ADICRz.STAMP = 1_B) AND
- the size of the DMA transaction (defined by CFCFGRz.TREL) is **less than** the size of the destination circular buffer (defined by ADICRz.CBLD)

the DMA shall append the DMA timestamp to the DMA write move data if the following DMA channel parameters are configured:

- ADICRz.DMF = 001_B (address offset is 2 x CHCFGRz.CHDW) AND
- CHCFGRz.CHDW = 010_B (32-bit data width for moves, SDTW)

In all other DMA destination circular buffer use cases, the DMA channel shall be configured to disable the appendage of DMA timestamp (ADICRz.STAMP = 0_B).

2.36 [DMA_TC.035] Last DMA Transaction in a Linked List triggers a DMA Daisy Chain

Description

DMA Channels can be daisy chained by setting the bit CHCFGRz.PRSEL = 1_B. When a higher priority DMA channel z completes a DMA transaction then it will initiate a DMA transaction on the next lower priority DMA channel z-1 by setting the access pending bit TSRz-1.CH.

However, if the current transaction was the last one in a linked list, and PRSEL is set to daisy chain, TSRz-1.CH of the next lower channel z-1 is set just after the TCS (transaction control set) load, that is, before the last transaction of the linked list has even started. Therefore the last TCS is not executed by the Linked List.

Workaround

Do not use Daisy Chain with Linked Lists (i.e. if ADICRz.SHCT[3:2] = 11_B then CHCFGRz.PRSEL = 0_B).

If the use case needs to trigger a further TCS in the next lower DMA channel then the trigger should be routed via the Interrupt Router.

2.37 [DMA_TC.036] Linked List: SADR/DADR can be overwritten when loading a non-LL TCS

Description

If a Linked List (LL) loads in a non-LL Transaction Control Set (TCS) which has a shadow mode selected (ADICRz.SHCT = 0001_B or 0010_B or 0100_B or 0101_B), during the write-back it can overwrite the contents of SADR/DADR in the newly loaded TCS before the DMA transaction has been run.

Workaround

Do not use shadow address modes with DMA Conditional Linked List.

***Note**: The Application Note AP32245 “DMA Linked List” will highlight that shadow address modes are not required.*

2.38 [DMA_TC.037] Conditional Linked List: Bit TSR.CH not cleared for a CLL transaction upon pattern match

Description

When a Conditional Linked List (CLL) pattern match is found, the transaction ends. TSR.CH should be cleared, and set later during write-back of the Transaction Control Set (TCS) if the newly loaded TCS is auto-starting (i.e. CHCSRz.SCH = 1_B).

Due to an internal problem TSR.CH is not cleared in this case.

Workaround

There is no workaround.

The assessment is that a DMA CLL transaction that does not get a match will transition to the next DMA transaction. The CH bit will be cleared.

2.39 [DMA_TC.038] Linked List: SIT interrupt when SIT bit set in newly loaded TCS

Description

The Set Interrupt Trigger (SIT) bit is a means of generating a DMA channel interrupt service request via software. It is a debug feature that allows to trigger the Interrupt Router, without configuring the DMA channel and executing a DMA transaction.

When a new Transaction Control Set (TCS) is loaded in linked list mode, and the SIT bit in the new TCS being loaded is set in the value written to register CHCSRz, a channel interrupt trigger will be activated.

Therefore, the SIT bit should always be set to 0_B when using linked lists.

***Note**: The latest versions of the documentation are/will be updated to reflect this.*

2.40 [DMA_TC.039] Read Data CRC

Description

The Read Data CRC (RDCRC) calculates an IEEE 802.3 ethernet CRC32 checksum as DMA moves read data through the DMA. The DMA implementation of the algorithm does not zero extend the read data for SDTB (8-bit) and SDTH (16-bit) accesses resulting in the calculation of a wrong checksum value.

The RDCRC must only be used with STDW (32-bit), SDTD (64-bit), BTR2 (128-bit) and BTR4 (256-bit) access sizes. It must be noted that SDTD, BTR2 and BTR4 are only supported for SRI-source to SRI-destination transactions.

2.41 [DMA_TC.040] DMA Linked Lists: Intermittent Clearing of Hardware Transaction Request Enable with mixed mode Transaction Control Sets

Description

When a DMA channel is configured for linked list operation, if a Transaction Control Set (TCS) is configured for Continuous Mode (DMA_CHCFGRz.CHMODE = 1_B) and the next TCS is configured for Single Mode (DMA_CHCFGRz.CHMODE = 0_B) then DMA_TSRz.HTRE may be intermittently cleared disabling the servicing of DMA hardware requests.

Workaround

If a DMA channel is configured for linked list operation then all application DMA transactions must be configured for Continuous Mode (DMA_CHCFGRz.CHMODE = 1_B). If there is a need for the application to clear the Hardware Transaction Request Enable (DMA_TSRz.HTRE = 0_B) then two additional dummy DMA transactions should be serviced by the DMA in the linked list:

- Dummy Transaction 1:

  The TCS is configured as a linked list TCS (DMA_ADICRz.SHCT = 0xC, 0xD or 0xE) in Single Mode (DMA_CHCFGRz.CHMODE = 0) and auto start (DMA_CHCSRz.SCH = 1_B). The TCS should configure a single DMA move to read a word from memory in order to write DMA_TSRz.DCH = 1_B and disable subsequent DMA hardware requests
- Dummy Transaction 2:

  The TCS is configured for normal shadow control mode (DMA_ADICRz.SHCT = 0000_B) and Single Mode. A dummy DMA move is performed

2.42 [DMA_TC.041] DMA Circular Buffer Wrap Interrupt

Description

If a DMA channel is configured for source circular buffer operation (ADICRz.SCBE = 1_B), the DMA shall correctly calculate the DMA source addresses. When the DMA source address wraps, the DMA is unreliable in updating the wrap source buffer status (CHCSRz.WRPS). If the wrap source buffer interrupt is enabled (ADICRz.WRPSE = 1_B), the DMA is unreliable in triggering a source wrap buffer interrupt.

If a DMA channel is configured for destination circular buffer operation (ADICRz.DCBE = 1_B), the DMA shall correctly calculate the DMA destination addresses. When the DMA destination address wraps, the DMA is unreliable in updating the wrap destination buffer status (CHCSRz.WRPD). If the wrap destination buffer interrupt is enabled (ADICRz.WRPDE = 1_B), the DMA is unreliable in triggering a destination wrap buffer interrupt.

Workaround

The source wrap buffer interrupt shall be disabled (ADICRz.WRPSE =0_B).

The destination wrap buffer interrupt shall be disabled (ADICRz.WRPDE =0_B).

If a DMA channel is configured for circular buffer operation (ADICRz.SCBE = 1_B or ADICRz.DCBE = 1_B), the DMA channel shall be configured as follows:

- The size of the DMA transaction shall equal the size of the circular buffer
- If a source circular buffer is configured (ADICRz.SCBE = 1_B), the initial DMA source address shall be the start address of the source circular buffer

- If a destination circular buffer is configured (ADICRz.DCBE = 1_B), the initial DMA destination address shall be the start address of the destination circular buffer
- The DMA channel interrupt control shall be configured to trigger an interrupt on completion of the DMA transaction (DMA_ADICRz.INTCT = 10_B and DMA_ADICRz.IRDV = 0000_B)

If a DMA channel is configured for both source circular buffer operation (ADICRz.SCBE = 1_B) AND destination circular buffer operation (ADICRz.DCBE = 1_B), the size of the source circular buffer shall equal the size of the destination circular buffer.

2.43 [DMA_TC.042] DMA Interrupt from Channel reported before Completion of DMA Transaction

Description

The Interrupt from Channel (ICH) status bit should be set on completion of a DMA transaction. If the DMA channel is configured to append a DMA Timestamp then validation have discovered that the ICH bit is set before the DMA timestamp has been written.

Workaround 1

On receipt of a DMA channel interrupt service request software shall poll the Move Engine (ME) Status Register(s) to confirm the DMA channel is no longer active.

1. Check active DMA channel in ME SR
2. Check Write Status in ME SR

If these fields in both ME are no longer the DMA channel that triggered the DMA channel interrupt service request then the DMA transaction has completed.

Workaround 2

To avoid polling the Move Engine status, the user may use a DMA linked list to execute the following DMA transactions:

- DMA transaction 1:
  - move operation (DMA timestamp shall not be selected)
- DMA transaction 2:
  - single 32-bit DMA move to copy DMA timestamp from DMA TIME register to next 32-bit aligned destination after DMA transaction 1

2.44 [DMA_TC.043] DMA Write Move Data Corruption for non 32-byte Aligned Cacheable Source Address

Description

If the DMA channel TCS selects a 256-bit channel data width and a non 32-byte aligned source address then the beat order of the DMA write move will be different for DMA read moves to cacheable (segments 8 and 9) and non-cacheable (segments A and B) source addresses. The effect is data corruption for accesses to cacheable addresses.

Workarounds

1. Use 32-byte aligned source addresses for DMA read move to cacheable addresses (segments 8 and 9)
2. Use non-cacheable source addresses (segments A and B)

2.45 [DMA_TC.044] Clock Switch after SPB Error Reported results in Spurious SRI Error

Description

If an SPB error is reported, and then immediately the SRI:SPB clock ratio is changed, then if the next DMA read move is to an SRI source address a spurious error may be reported.

Workaround

1. The system shall not change the SRI:SPB clock ratio while the DMA is active
2. The DMA error handler should monitor the reporting of SPB and SRI errors after a clock switch

2.46 [DMA_TC.045] DMA Reconfigures DMA Channels Lockup

Description

If two or more DMA channels are used to re-configure other DMA channels (i.e. perform a DMA write move to DMA address space) the DMA may lock up if the re-configuration DMA channels are assigned to different DMA hardware resource partitions.

The effect of the DMA lock up is to lock up other SPB master interfaces which attempt a write access to DMA address space.

Workaround

All DMA channels used to re-configure other DMA channels shall be assigned to the same hardware resource partition in their corresponding DMA Channel Hardware Resource Registers HRRz.

2.47 [DMA_TC.046] Shadow Operation Read Only Mode

Description

If a DMA channel is configured for Source Address Buffering Read Only (ADICR.SHCT = 0001_B) or Destination Address Buffering Read Only (ADICR.SHCT = 0010_B), the DMA is unreliable when performing a shadow address update. In these modes, the SADR/DADR registers may get directly updated (instead of SHADR) in the middle of a transaction, potentially resulting in a DMA data transfer corruption.

Workaround

The DMA channel configuration for Read Only Modes (SHCT = 0001_B or SHCT = 0010_B) must not be used.

Instead, to update the SADR/DADR in the middle of a transaction, use the corresponding Direct Write Mode for Source Address Buffering (ADICR.SHCT = 0101_B) or Destination Address Buffering (ADICR.SHCT = 0110_B), and write the new address to the SHADR register.

2.48 [DMA_TC.049] Bus Error Reported During LL TCS Load

Description

If a DMA channel is configured for Linked List (LL) operation AND a bus error is reported during the load of a new Transaction Control Set (TCS), the DMA shall set the DMA_ERRSRx.DLLER status bit (Move Engine x DMA Linked List Error).

Erroneously, the DMA additionally sets the DMA_ERRSRx.SER status bit (Move Engine x Source Error).

Workaround

None.

2.49 [DMA_TC.050] Clearing CHCSR.FROZEN during Double Buffering

Description

If a DMA channel is configured for one of the following Double Buffering operations:

- 1001_B Double Source Buffering Automatic Hardware and Software Switch
- 1011_B Double Destination Buffering Automatic Hardware and Software Switch

AND the active buffer fills/empties before software has cleared the DMA channel CHCSRz.FROZEN bit, the DMA shall overflow/underflow the active buffer.

Erroneously, the DMA will not trigger a Transaction Request Lost (TRL) error.

Workaround

Software shall clear DMA channel CHCSRz.FROZEN before the active buffer overflows/underflows.

2.50 [DMA_TC.052] SER and DER During Linked List Operations

Description

Software may configure a DMA channel for one of the DMA linked list operations:

- DMA linked list
  - (DMA channel DMA_ADICRz.SHCT = 1100_B)
- Accumulated linked list
  - (DMA channel DMA_ADICRz.SHCT = 1101_B)
- Safe linked list
  - (DMA channel DMA_ADICRz.SHCT = 1110_B)
- Conditional linked list
  - (DMA channel DMA_ADICRz.SHCT = 1111_B)

If the DMA is servicing a DMA request for a DMA channel configured for one of the linked list operations and the DMA indicates a Source Error (SER) (i.e. DMA_ERRSRx.SER = 1_B) or a Destination Error (DER) (i.e. DMA_ERRSRx.DER = 1_B), the DMA completes the current DMA transaction. If the DMA channel is configured for conditional linked list, the DMA disables pattern matching for each DMA read move reporting a SER. When the DMA completes the current DMA transaction, the DMA stops servicing the linked list operation and the DMA will not load the next transaction control set to allow debug of the current DMA transaction.

Erroneously, upon a SER or DER, the DMA does not reliably stop the linked list operation (when it should) on completion of the current DMA transaction.

If the Move Engine is configured to enable DMA error interrupt service request for SER (DMA_EERx.ESER =1_B) and for DER (DMA_EERx.EDER = 1_B), the DMA triggers a DMA error interrupt service request.

The application software should include a DMA error handler to resolve all DMA errors including SER and DER.

Workaround

None.

2.51 [DMA_TC.053] TS16_ERR Type of Error Reporting Unreliable

Description

During debugging, the error trigger set (TS16_ERR) may be used to identify the type of DMA error and the number of the DMA channel. After TS16_ERR reports an error the error type bits (ME0SE, ME0DE, ME1SE and ME1DE) are not cleared. If TS16_ERR reports a subsequent error, the type of error reporting is unreliable.

Workaround

After TS16_ERR reports an error, the error type bits must be cleared.

2.52 [DMA_TC.054] DMA Channel Halt Acknowledge Unreliable

Description

Software may halt a DMA channel by writing to the halt request bit (TSRz.HLTREQ = 1_B). When a DMA channel enters the halt state, the DMA reports DMA channel halt acknowledge (TSRz.HLTACK = 1_B).

The reporting of DMA channel halt acknowledge is unreliable when software sets the TSRz.HLTREQ bit just as channel z is about to be scheduled to a move engine. In this case, the DMA may report a DMA channel is halted when the DMA channel is active in a move engine.

Workaround

If the DMA reports a DMA channel is halted, the software should check the DMA channel is not active in a move engine by monitoring the active channel in the move engine status register(s).

2.53 [DMA_TC.055] ICU to DMA Interface in Sleep Mode

Description

The Interrupt Router triggers DMA hardware requests via the ICU interface. If the DMA is in sleep mode, the DMA will not acknowledge DMA hardware requests. The effect is to lock up the ICU to DMA interface.

Workaround

The application must disable the triggering of DMA hardware requests before placing the DMA in sleep mode.

2.54 [DMA_TC.056] TSR and SUSENR Access Protection Unreliable

Description

The DMA access protection is part of a system wide access protection scheme to restrict write accesses to DMA registers to individual on-chip bus masters.

If the application software configures DMA freedom from interference measures (i.e. when any on-chip bus master write to the DMA is prohibited by a DMA access enable setting), then on-chip bus master writes to the DMA channel TSR and SUSENR registers are unreliable and may result in the following effects:

1. Safety Related Effects

- 1.1. An illegal write access to a DMA channel TSR register will succeed with no indication

The safety related effects (in point 1.1) relate to the DMA channel reset, halt and hardware request control functions in the TSR register. The most severe safety effect is that a DMA operation may be lost.

Workaround (for 1.1)

If the application software implements temporal monitoring of DMA transactions (for example using DMA timestamp) to detect lost DMA operations, the application software will detect the effect of the illegal access to DMA channel TSR register.

2. Non Safety Related Effects

- 2.1. An illegal write access to a DMA channel SUSENR register may succeed with no indication.
  - Impact of 2.1: The SUSENR register is a debug only register. No impact is foreseen during a normal application
- 2.2. A legal write access to a DMA channel TSR register may fail with an indication - this means unexpected bus errors may be triggered when accessing TSR registers
- 2.3. A legal write access to a DMA channel SUSENR register may fail with an indication - this means unexpected bus errors may be triggered when accessing SUSENR registers.
  - Impact of 2.2 & 2.3: Unexpected SPB bus errors and hence CPU traps and SPB error alarms may occur during application run

Workaround (for 2.2 & 2.3)

If the system implements DMA freedom from interference measures, then the Impact of 2.2 & 2.3 will occur, and cause unexpected SPB bus errors and hence CPU traps and SPB error alarms when writing to TSR and SUSENR registers.

In order to work around this problem, the application software shall implement all of the following steps:

- W1: Before an intended write access to a DMA channel TSR or SUSENR register, perform an additional preceding write access to a DMA channel Transaction Control Set (TCS) register of the same DMA channel.
  - TCS registers include the DMA channel RDCRC, SDCRC, SADR, DADR, SHADR, ADICR, CHCSR and CHCFGR registers
- W2: Ensure that this additional preceding write access to a DMA channel TCS register has no real effect. Recommendation: Simply read and write back the RDCRCR register
- W3: Perform the write access to the DMA channel TSR register

Ensure that no other on-chip bus master can access any DMA register of a different resource partition between steps W2 and W3 in the workaround above.

Example Code Snippet

To update TSR register of DMA channel 25 with value:

1. Uint32 temp = DMA_RDCRCR25.U;
2. DMA_ RDCRCR25.U = temp;
3. DMA_TSR25.U = value;

2.55 [DMA_TC.058] Linked List Load Transaction Control Set (TCS) Integrity Error

Description

If DMA channel z is configured for one of the following linked list operations:

- DMA Linked List
  - (DMA channel ADICRz.SHCT = 1100_B)
- Accumulated Linked List
  - (DMA channel ADICRz.SHCT = 1101_B)
- Safe Linked List
  - (DMA channel ADICRz.SHCT = 1110_B)
- Conditional Linked List
  - (DMA channel ADICRz.SHCT = 1111_B)

Then on completion of a DMA transaction a new TCS is loaded into DMA channel z from the on-chip bus.

The DMA ignores data integrity errors in the new TCS:

- The DMA does not trigger an alarm to the SMU
- The DMA does not store any DMA error status
- The DMA may execute a corrupted DMA transaction

Detection of most corrupted DMA transactions is provided by the DMA safety mechanisms as follows:

- Use of the DMA address checksum to detect address generation faults
- Use of the DMA timestamp[1]to detect temporal faults

Workaround

None.

2.56 [DMA_TC.061] DMA Double Buffering Operations

Description

***Note**: This erratum DMA_TC.061 (DMA Double Buffering Operations) substitutes the following errata text modules*

- *DMA_TC.029 (DMA Double Buffering Overflow),*
- *DMA_TC.047 (DMA Double Buffering Buffer Switch), and*
- *DMA_TC.057 (Double Buffering Overflow Causes Other Channel Corruption)*

included in previous TC2xx errata sheet releases.

Software may configure a DMA channel for one of the DMA double buffering operations:

- DMA Double Source Buffering Software Switch Only
  - (DMA channel DMA_ADICRz.SHCT = 1000_B)
- DMA Double Source Buffering Automatic Hardware and Software Switch
  - (DMA channel DMA_ADICRz.SHCT = 1001_B)
- DMA Double Destination Buffering Software Switch Only
  - (DMA channel DMA_ADICRz.SHCT = 1010_B)
- DMA Double Destination Buffering Automatic Hardware and Software Switch
  - (DMA channel DMA_ADICRz.SHCT = 1011_B)

If the DMA is servicing a DMA request for a DMA channel configured for one of the double buffering operations AND the software executes a Software Buffer Switch operation (DMA_CHCSRz.SWB = 1_B), the DMA will not perform the buffer switch reliably.

The following sections provide recommendations for the implementation of DMA double buffering operations.

Supported DMA double buffering operations

As a consequence, the software should configure for a limited number of DMA double buffering operations:

- DMA Double Source Buffering Automatic Hardware Switch
  - (DMA channel DMA_ADICRz.SHCT = 1001_B)
- DMA Double Destination Buffering Automatic Hardware Switch
  - (DMA channel DMA_ADICRz.SHCT = 1011_B)

The software must

- NOT perform a Software Buffer Switch (DMA_CHCSRz.SWB = 0_B)
- NOT set the frozen bit (DMA_CHCSRz.FROZEN = 1_B)

[1] Conditional Linked List does not support the appendage of timestamps (ADICRz.STAMP = 0_B).

DMA channel ETRL configuration

The software must set the Enable Transaction Request Lost (ETRL) bit (DMA_ADICRz.ETRL = 1_B) to prevent the DMA locking up during a DMA double buffering operation.

DMA channel monitoring

The software should configure the DMA to trigger a DMA channel interrupt service request when the DMA empties (source buffering) or fills (destination buffering) a buffer on the completion of a DMA transaction. The software must service the DMA channel interrupt service requests. As soon as the software has analysed a buffer, the software must clear the frozen bit (DMA_CHCSRz.FROZEN = 0_B) and re-initialise the buffer address pointer.

DMA channel underflow or overflow

If the software fails to analyse a frozen buffer before the next DMA channel interrupt service request, the DMA channel will underflow (source buffering) or overflow (destination buffering) on receiving the next DMA request. Erroneously, the DMA will not trigger a DMA error interrupt service request.

As soon as the CPU receives a DMA channel interrupt service request, the software must check for an underflow or overflow by monitoring the DMA transaction count. If the software reads a zero transaction count (DMA_CHCSRz.TCOUNT = 0_D), the DMA channel is in an underflow or overflow state.

DMA channel interference

Erroneously a DMA channel underflow or overflow may cause the setting of the TRL flag and the clearing of a DMA request in one or more other DMA channels (note: dependent on the scheduling of DMA channels around this DMA request). The DMA channel interference is independent of resource partition assignment.

DMA channel reset

If the software detects a DMA channel underflow or overflow, the software must apply a DMA channel reset to all used DMA channels. On completion of the DMA channel reset, the software must re-configure all used DMA channels.

Alternatively, the software may apply an application reset.

Workaround

None.

2.57 [DMA_TC.062] Termination of DMA Transaction for Pattern Match

Description

If a DMA channel is configured for pattern detection and the DMA detects a pattern match, the DMA should terminate the DMA transaction. The DMA should provide the software with the capability to use the DMA channel status to identify the transfer number of the DMA move data.

Erroneously, the DMA may decrement 1 from the TCOUNT value making identification of the DMA move data unreliable.

Workaround

None.

2.58 [DMA_TC.063] DMA Timestamp Destination Address

Description

If software configures a DMA channel

- for increment of DMA destination address (DMA_ADICRz.INCD = 1_B) AND
- to append a DMA timestamp (DMA_ADICRz.STAMP = 1_B);

and the intended write address of the DMA timestamp is in a different 32 Kbyte page to the last DMA destination address to write DMA move data, the DMA erroneously calculates the DMA timestamp write address. The DMA writes the DMA timestamp to an incorrect address inside the same 32 Kbyte page as the last DMA destination address.

Workaround

The last DMA destination address and the write address of the DMA timestamp shall exist in the same 32 Kbyte page (i.e. and shall not cross the 32 Kbyte page boundary).

2.59 [DMA_TC.064] DMA Daisy Chain Request

Description

If software configures a DMA channel for one of the following DMA operations:

- DMA Pattern Detection
  - (DMA channel DMA_CHCFGRz.PATSEL[1:0] != 00_B)
- DMA Double Source Buffering Software Switch Only
  - (DMA channel DMA_ADICRz.SHCT = 1000_B)
- DMA Double Source Buffering Automatic Hardware and Software Switch
  - (DMA channel DMA_ADICRz.SHCT = 1001_B)
- DMA Double Destination Buffering Software Switch Only
  - (DMA channel DMA_ADICRz.SHCT = 1010_B)
- DMA Double Destination Buffering Automatic Hardware and Software Switch
  - (DMA channel DMA_ADICRz.SHCT = 1011_B)
- DMA linked list
  - (DMA channel DMA_ADICRz.SHCT = 1100_B)
- Accumulated linked list
  - (DMA channel DMA_ADICRz.SHCT = 1101_B)
- Safe linked list
  - (DMA channel DMA_ADICRz.SHCT = 1110_B)
- Conditional linked list
  - (DMA channel ADICRz.SHCT = 1111_B)

the software must not select daisy chain (DMA channel CHCFGRz.PRSEL = 0_B).

2.60 [DMA_TC.065] DMA Move Concurrent Bus Accesses

Description

The highest number DMA channel always wins arbitration to shared DMA resources (Move Engine and DMA on-chip bus master interfaces). The configuration of the DMA priority (DMA_CHCFGRx.DMAPRIO) has no effect on internal DMA arbitration.

The DMA priority is used by the System Peripheral Bus (SPB) controller to arbitrate between requests from all the SPB master interfaces.

Workaround

None.

2.61 [DMA_TC.066] DMA double buffering operations - Update address pointer

Description

Software may configure a DMA channel for one of the DMA double buffering operations:

- DMA Double Source Buffering Software Switch Only
  - (DMA channel DMA_ADICRz.SHCT = 1000_B)
- DMA Double Source Buffering Automatic Hardware and Software Switch
  - (DMA channel DMA_ADICRz.SHCT = 1001_B)
- DMA Double Destination Buffering Software Switch Only
  - (DMA channel DMA_ADICRz.SHCT = 1010_B)
- DMA Double Destination Buffering Automatic Hardware and Software Switch
  - (DMA channel DMA_ADICRz.SHCT = 1011_B)

If the software updates a buffer address pointer by BYTE or HALF-WORD writes, the resulting value of the address pointer is corrupted.

Workaround

If the software updates a buffer address pointer, the software should only use a 32-bit WORD access.

2.62 [DTS_TC.001] Temperature Sensor Formula

Description

The formula documented in older Data Sheet versions may result in an increased temperature error when calculating the junction temperature Tj of the device from a DTS temperature measurement.

To properly calculate the temperature measured by the DTS in [°C] from the RESULT bit-field of register SCU_DTSSTAT, it is recommended to use the following formulas depending on the contents of bit-field SCU_DTSCON[30:29]:

- While bit-field SCU_DTSCON[30:29] = 00_B: Tj = (RESULT - 607_D) / 2.13
- While bit-field SCU_DTSCON[30:29] = 01_B: Tj = (RESULT - 646_D) / 2.11

Bit field SCU_DTSCON[30:29] can only deliver one of the two values (00_B, 01_B) listed above (constant for a given device).

Make sure the application software does not modify the values installed during device start-up in register SCU_DTSCON.

***Note**: The description in the Data Sheet will be updated appropriately.*

2.63 [FLASH_TC.052] Use of Write Page Once command

Description

When applying a Write Page Once (WPO) command to a pre-programmed or incompletely erased PFlash location, the WPO command will fail as expected, with both EVER (Erase Verify Error) and PVER (Program Verify Error) error flags being raised.

For an EVER failure in the WPO command, the read bias conditions on the NVM cells for the subsequent read operations will be incorrect. The incorrect bias conditions at the NVM cell terminals may lead to single-bit or multi-bit errors in the PFlash. Only zeroes (erased cells) will be affected by this phenomenon.

The physical content of the flash cells is not damaged by the incorrect read bias conditions, or by the WPO command failure.

***Note**: As per the safety manual's Architecture for Management of Faults [SM_AURIX_PMU_3], it is assumed that the WPO command is not used during application run time.*

Workaround

The incorrect NVM read bias conditions can be fully recovered by performing one of the following actions immediately after the WPO failure:

- Request Flash module sleep mode and wake-up immediately after the WPO failure:
  - Request Sleep mode by setting bit FCON.SLEEP = 1_B
  - Poll the Flash Sleep Mode status bit FSR.SLM to make sure that the Flash is in sleep mode
  - Initiate wake-up by clearing FCON.SLEEP = 0_B
  - Poll status bit FSR.SLM to make sure that the flash is in normal state again

***Note**: For more details about AURIX™ power-down modes, please refer to Application Note "AURIX™ standby power mode" (AP32332).*

- Perform System Reset immediately after the WPO failure

2.64 [FlexRay_AI.087] After reception of a valid sync frame followed by a valid non-sync frame in the same static slot the received sync frame may be ignored

Description

If in a static slot of an even cycle a valid sync frame followed by a valid non-sync frame is received, and the frame valid detection (prt_frame_decoded_on_X) of the DEC process occurs one sclk after valid frame detection of FSP process (fsp_val_syncfr_chx), the sync frame is not taken into account by the CSP process (devte_xxs_reg).

Scope

The erratum is limited to the case where more than one valid frame is received in a static slot of an even cycle.

Effects

In the described case the sync frame is not considered by the CSP process. This may lead to a SyncCalcResult of MISSIMG_TERM (error flag SFS.MRCS set). As a result the POC state may switch to NORMAL_PASSIVE or HALT or the Start-up procedure is aborted.

Workaround

Avoid static slot configurations long enough to receive two valid frames.

2.65 [FlexRay_AI.088] A sequence of received WUS may generate redundant SIR.WUPA/B events

Description

If a sequence of wake-up symbols (WUS) is received, all separated by appropriate idle phases, a valid wake-up pattern (WUP) should be detected after every second WUS.The E-Ray detects a valid wake-up pattern after the second WUS and then after each following WUS.

Scope

The erratum is limited to the case where the application program frequently resets the appropriate SIR.WUPA/B bits.

Effects

In the described case there are more SIR.WUPA/B events seen than expected.

Workaround

Ignore redundant SIR.WUPA/B events.

2.66 [FlexRay_AI.089] Rate correction set to zero in case of SyncCalcResult=MISSING_TERM

Description

In case a node receives too few sync frames for rate correction calculation and signals a SyncCalcResult of MISSING_TERM, the rate correction value is set to zero instead of to the last calculated value.

Scope

The erratum is limited to the case of receiving too few sync frames for rate correction calculation (SyncCalcResult=MISSING_TERM in an odd cycle).

Effects

In the described case a rate correction value of zero is applied in NORMAL_ACTIVE / NORMAL_PASSIVE state instead of the last rate correction value calculated in NORMAL_ACTIVE state. This may lead to a desynchronisation of the node although it may stay in NORMAL_ACTIVE state (depending on gMaxWithoutClockCorrectionPassive) and decreases the probability to re-enter NORMAL_ACTIVE state if it has switched to NORMAL_PASSIVE (pAllowHaltDueToclock=false).

Workaround

It is recommended to set gMaxWithoutClockCorrectionPassive to 1. If missing sync frames cause the node to enter NORMAL_PASSIVE state, use higher level application software to leave this state and to initiate a re-integration into the cluster. HALT state can also be used instead of NORMAL_PASSIVE state by setting pAllowHaltDueToClock to true.

2.67 [FlexRay_AI.090] Flag SFS.MRCS is set erroneously although at least one valid sync frame pair is received

Description

If in an odd cycle 2c+1 after reception of a sync frame in slot n the total number of different sync frames per double cycle has exceeded gSyncNodeMax and the node receives in slot n+1 a sync frame that matches with a sync frame received in the even cycle 2c, the sync frame pair is not taken into account by CSP process. This may cause the flags SFS.MRCS and EIR.CCF to be set erroneously.

Scope

The erratum is limited to the case of a faulty cluster configuration where different sets of sync frames are transmitted in even and odd cycles and the total number of different sync frames is greater than gSyncNodeMax.

Effects

In the described case the error interrupt flag EIR.CCF is set and the node may enter either the POC state NORMAL_PASSIVE or HALT.

Workaround

Correct configuration of gSyncNodeMax.

2.68 [FlexRay_AI.091] Incorrect rate and / or offset correction value if second Secondary Time Reference Point (STRP) coincides with the action point after detection of a valid frame

Description

If a valid sync frame is received before the action point and additionally noise or a second frame leads to a STRP coinciding with the action point, an incorrect deviation value of zero is used for further calculations of rate and/or offset correction values.

Scope

The erratum is limited to configurations with an action point offset greater than the static frame length.

Effects

In the described case a deviation value of zero is used for further calculations of rate and/or offset correction values. This may lead to an incorrect rate and / or offset correction of the node.

Workaround

Configure action point offset smaller than static frame length.

2.69 [FlexRay_AI.092] Initial rate correction value of an integrating node is zero if pMicroInitialOffsetA,B = 0x00

Description

The initial rate correction value as calculated in figure 8-8 of protocol spec v2.1 is zero if parameter pMicroInitialOffsetA,B was configured to be zero.

Scope

The erratum is limited to the case where pMicroInitialOffsetA,B is configured to zero.

Effects

Starting with an initial rate correction value of zero leads to an adjustment of the rate correction earliest 3 cycles later (see figure 7-10 of protocol spec v2.1). In a worst case scenario, if the whole cluster is drifting away too fast, the integrating node would not be able to follow and therefore abort integration.

Workaround

Avoid configurations with pMicroInitialOffsetA,B equal to zero. If the related configuration constraint of the protocol specification results in pMicroInitialOffsetA,B equal to zero, configure it to one instead. This will lead to a correct initial rate correction value, it will delay the start-up of the node by only one micro tick.

2.70 [FlexRay_AI.093] Acceptance of start-up frames received after reception of more than gSyncNodeMax sync frames

Description

If a node receives in an even cycle a start-up frame after it has received more than gSyncNodeMax sync frames, this start-up frame is added erroneously by process CSP to the number of valid start-up frames (zStartupNodes). The faulty number of start-up frames is delivered to the process POC. As a consequence this node may integrate erroneously to the running cluster because it assumes that it has received the required number of start-up frames.

Scope

The erratum is limited to the case of more than gSyncNodeMax sync frames.

Effects

In the described case a node may erroneously integrate successfully into a running cluster.

Workaround

Use frame schedules where all start-up frames are placed in the first static slots. gSyncNodeMax should be configured to be greater than or equal to the number of sync frames in the cluster.

2.71 [FlexRay_AI.094] Sync frame overflow flag EIR.SFO may be set if slot counter is greater than 1024

Description

If in the static segment the number of transmitted and received sync frames reaches gSyncNodeMax and the slot counter in the dynamic segment reaches the value cStaticSlotIDMax + gSyncNodeMax = 1023 + gSyncNodeMax, the sync frame overflow flag EIR.SFO is set erroneously.

Scope

The erratum is limited to configurations where the number of transmitted and received sync frames equals to gSyncNodeMax and the number of static slots plus the number of dynamic slots is greater or equal than 1023 + gSyncNodeMax.

Effects

In the described case the sync frame overflow flag EIR.SFO is set erroneously. This has no effect to the POC state.

Workaround

Configure gSyncNodeMax to number of transmitted and received sync frames plus one or avoid configurations where the total of static and dynamic slots is greater than cStaticSlotIDMax.

2.72 [FlexRay_AI.095] Register RCV displays wrong value

Description

If the calculated rate correction value is in the range of [-pClusterDriftDamping .. +pClusterDriftDamping], vRateCorrection of the CSP process is set to zero. In this case register RCV should be updated with this value. Erroneously RCV.RCV[11:0] holds the calculated value in the range [-pClusterDriftDamping .. +pClusterDriftDamping] instead of zero.

Scope

The erratum is limited to the case where the calculated rate correction value is in the range of [-pClusterDriftDamping .. +pClusterDriftDamping].

Effects

The displayed rate correction value RCV.RCV[11:0] is in the range of [-pClusterDriftDamping .. +pClusterDriftDamping] instead of zero. The error of the displayed value is limited to the range of [-pClusterDriftDamping .. +pClusterDriftDamping]. For rate correction in the next double cycle always the correct value of zero is used.

Workaround

A value of RCV.RCV[11:0] in the range of [-pClusterDriftDamping .. +pClusterDriftDamping] has to be interpreted as zero.

2.73 [FlexRay_AI.096] Noise following a dynamic frame that delays idle detection may fail to stop slot

Description

If (in case of noise) the time between ’potential idle start on X’ and ’CHIRP on X’ (see Protocol Spec. v2.1, Figure 5-21) is greater than gdDynamicSlotIdlePhase, the E-Ray will not remain for the remainder of the current dynamic segment in the state ’wait for the end of dynamic slot rx’. Instead, the E-Ray continues slot counting. This may enable the node to further transmissions in the current dynamic segment.

Scope

The erratum is limited to noise that is seen only locally and that is detected in the time window between the end of a dynamic frame’s DTS and idle detection (’CHIRP on X’).

Effects

In the described case the faulty node may not stop slot counting and may continue to transmit dynamic frames. This may lead to a frame collision in the current dynamic segment.

Workaround

None

2.74 [FlexRay_AI.097] Loop back mode operates only at 10 MBit/s

Description

The looped back data is falsified at the two lower baud rates of 5 and 2.5 MBit/s.

Scope

The erratum is limited to test cases where loop back is used with the baud rate prescaler (PRTC1.BRP[1:0]) configured to 5 or 2.5 MBit/s.

Effects

The loop back self test is only possible at the highest baud rate.

Workaround

Run loop back tests with 10 MBit/s (PRTC1.BRP[1:0] = 00_B).

2.75 [FlexRay_AI.099] Erroneous cycle offset during start-up after abort of start-up or normal operation

Description

An abort of start-up or normal operation by a READY command near the macro tick border may lead to the effect that the state INITIALIZE_SCHEDULE is one macro tick too short during the first following integration attempt. This leads to an early cycle start in state INTEGRATION_COLDSTART_CHECK or INTEGRATION_CONSISTENCY_CHECK.

As a result the integrating node calculates a cycle offset of one macro tick at the end of the first even/odd cycle pair in the states INTEGRATION_COLDSTART_CHECK or INTEGRATION_CONSISTENCY_CHECK and tries to correct this offset.

If the node is able to correct the offset of one macro tick (pOffsetCorrectionOut >> gdMacrotick), the node enters NORMAL_ACTIVE with the first start-up attempt.

If the node is not able to correct the offset error because pOffsetCorrectionOut is too small (pOffsetCorrectionOut ≤ gdMacrotick), the node enters ABORT_STARTUP and is ready to try start-up again. The next (second) start-up attempt is not effected by this erratum.

Scope

The erratum is limited to applications where READY command is used to leave STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state.

Effects

In the described case the integrating node tries to correct an erroneous cycle offset of one macro tick during start-up.

Workaround

With a configuration of pOffsetCorrectionOut >> gdMacrotick • (1+cClockDeviationMax) the node will be able to correct the offset and therefore also be able to successfully integrate.

2.76 [FlexRay_AI.100] First WUS following received valid WUP may be ignored

Description

When the protocol engine is in state WAKEUP_LISTEN and receives a valid wake-up pattern (WUP), it transfers into state READY and updates the wake-up status vector CCSV.WSV[2:0] as well as the status interrupt flags SIR.WST and SIR.WUPA/B . If the received wake-up pattern continues, the protocol engine may ignore the first wake-up symbol (WUS) following the state transition and signals the next SIR.WUPA/B at the third instead of the second WUS.

Scope

The erratum is limited to the reception of redundant wake-up patterns.

Effects

Delayed setting of status interrupt flags SIR.WUPA/B for redundant wake-up patterns.

Workaround

None

2.77 [FlexRay_AI.101] READY command accepted in READY state

Description

The E-Ray module does not ignore a READY command while in READY state.

Scope

The erratum is limited to the READY state.

Effects

Flag CCSV.CSI is set. Cold starting needs to be enabled by POC command ALLOW_COLDSTART (SUCC1.CMD = 1001_B).

Workaround

None

2.78 [FlexRay_AI.102] Slot status vPOC!SlotMode is reset immediately when entering HALT state

Description

When the protocol engine is in the states NORMAL_ACTIVE or NORMAL_PASSIVE, a HALT or FREEZE command issued by the Host resets vPOC!SlotMode immediately to SINGLE slot mode (CCSV.SLM[1:0] = 00_B). According to the FlexRay protocol specification, the slot mode should not be reset to SINGLE slot mode before the following state transition from HALT to DEFAULT_CONFIG state.

Scope

The erratum is limited to the HALT state.

Effects

The slot status vPOC!SlotMode is reset to SINGLE when entering HALT state.

Workaround

None

2.79 [FlexRay_AI.103] Received messages not stored in Message RAM when in Loop Back Mode

Description

After a FREEZE or HALT command has been asserted in NORMAL_ACTIVE state, and if state LOOP_BACK is then entered by transition from HALT state via DEF_CONFIG and CONFIG, it may happen that acceptance filtering for received messages is not started, and therefore these messages are not stored in the respective receive buffer in the Message RAM.

Scope

The erratum is limited to the case where Loop Back Mode is entered after NORMAL_ACTIVE state was left by FREEZE or HALT command.

Effects

Received messages are not stored in Message RAM because acceptance filtering is not started.

Workaround

Leave HALT state by hardware reset.

2.80 [FlexRay_AI.104] Missing start-up frame in cycle 0 at coldstart after FREEZE or READY command

Description

When the E-Ray is restarted as leading coldstarter after it has been stopped by FREEZE or READY command, it may happen, depending on the internal state of the module, that the E-Ray does not transmit its start-up frame in cycle 0. Only E-Ray configurations with start-up frames configured for slots 1 to 7 are affected by this behavior.

Scope

The erratum is limited to the case when a coldstart is initialized after the E-Ray has been stopped by FREEZE or READY command. Coldstart after hardware reset is not affected.

Effects

During coldstart it may happen that no start-up frame is sent in cycle 0 after entering COLDSTART_COLLISION_RESOLUTION state from COLDSTART_LISTEN state.

The next coldstart attempt is no longer affected. Coldstart sequence is lengthened but coldstart of FlexRay system is not prohibited by this behavior.

Workaround

Use a static slot greater or equal 8 for the start-up/sync message.

2.81 [FlexRay_AI.105] RAM select signals of IBF1/IBF2 and OBF1/OBF2 in RAM test mode

Description

When accessing Input Buffer RAM 1, 2 (IBF1, 2) or Output Buffer RAM 1, 2 (OBF1, 2) in RAM test mode, the following behavior can be observed when entering RAM test mode after hardware reset.

- Read or write access to IBF2:
  - In this case also IBF1 RAM select eray_ibf1_cen is activated initiating a read access of the addressed IBF1 RAM word. The data read from IBF1 is evaluated by the respective parity checker.
- Read or write access to OBF1:
  - In this case also OBF2 RAM select eray_obf2_cen is activated initiating a read access of the addressed OBF2 RAM word. The data read from OBF2 is evaluated by the respective parity checker.

If the parity logic of the erroneously selected IBF1 resp. OBF2 detects a parity error, bit MHDS.PIBF resp. MHDS.POBF in the E-Ray Message Handler Status register is set although the addressed IBF2 resp. OBF1 had not error. The logic for setting MHDS.PIBF / MHDS.POBF does not distinguish between set conditions from IBF1 or IBF2 resp. OBF1 or OBF2.

Due to the IBF / OBF swap mechanism as described in section 5.11.2 in the E-Ray Specification, the inverted behavior with respect to IBF1, 2 and OBF1, 2 can be observed depending on the IBF / OBF access history.

Scope

The erratum is limited to the case when IBF1, 2 or OBF1, 2 are accessed in RAM test mode. The problem does not occur when the E-Ray is in normal operation mode.

Effects

When reading or writing IBF1, 2 / OBF1, 2 in RAM test mode, it may happen, that the parity logic of IBF1, 2 / OBF1, 2 signals a parity error.

Workaround

For RAM testing after hardware reset, the Input / Output Buffer RAMs have to be first written and then read in the following order: IBF1 before IBF2 and OBF2 before OBF1

2.82 [FlexRay_AI.106] Data transfer overrun for message transfers Message RAM to Output Buffer (OBF) or from Input Buffer (IBF) to Message RAM

Description

The problem occurs under the following conditions:

1) A received message is transferred from the Transient Buffer RAM (TBF) to the message buffer that has its data pointer pointing to the first word of the Message RAM's Data Partition located directly after the last header word of the Header Partition of the Last Configured Buffer as defined by MRC.LCB

2) The Host triggers a transfer from / to the Last Configured Buffer in the Message RAM with a specific time relation to the start of the TBF transfer described under 1)

Under these conditions the following transfers triggered by the Host may be affected:

a) Message buffer transfer from Message RAM to OBF

When the message buffer has its payload configured to maximum length (PLC = 127), the OBF word on address 00h (payload data bytes 0 to 3) is overwritten with unexpected data at the end of the transfer.

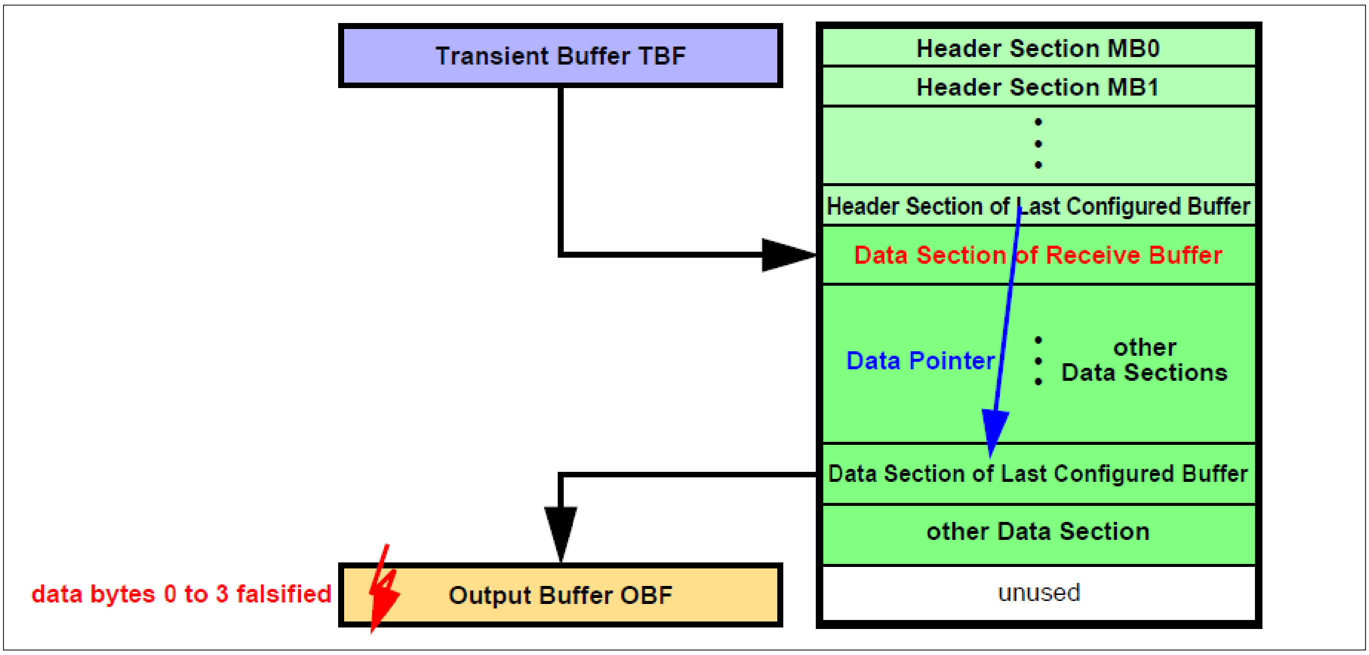

Transient Buffer TBF
Header Section MB0
Header Section MB1
Header Section of Last Configured Buffer
Data Section of Receive Buffer
Data Pointer
other Data Sections
Data Section of Last Configured Buffer
other Data Section
unused
data bytes 0 to 3 falsified
Output Buffer OBF

Figure 1 Message buffer transfer from Message RAM to OBF

b) Message buffer transfer from IBF to Message RAM

After the Data Section of the selected message buffer in the Message RAM has been written, one additional write access overwrites the following word in the Message RAM which might be the first word of the next Data Section.

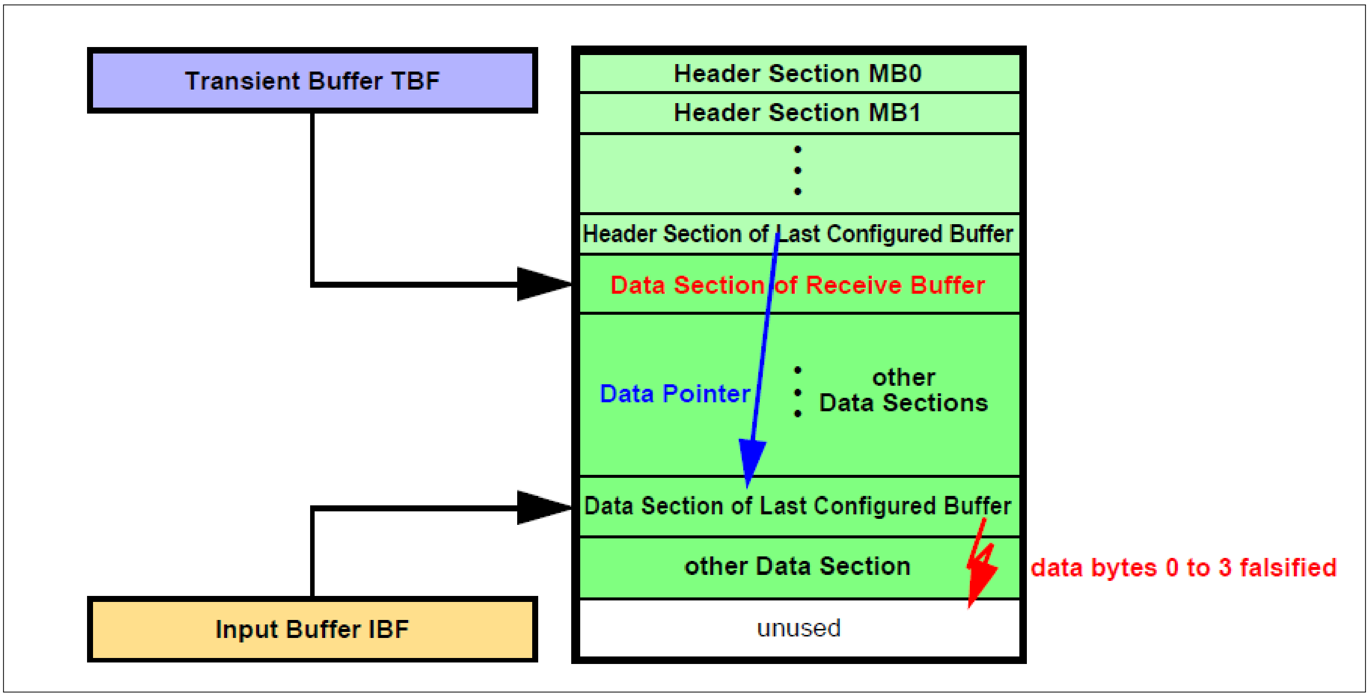

Transient Buffer TBF
Header Section MB0
Header Section MB1
Header Section of Last Configured Buffer
Data Section of Receive Buffer
Data Pointer
other Data Sections
Data Section of Last Configured Buffer
other Data Section
data bytes 0 to 3 falsified
Input Buffer IBF
unused

Figure 2 Message buffer transfer from IBF to Message RAM

Scope

The erratum is limited to the case when (see Figure 3 “Bad Case”):

1) The first Data Section in the Data Partition is assigned to a receive buffer (incl. FIFO buffers)

AND

2) The Data Partition in the Message RAM starts directly after the Header Partition (no unused Message RAM word in between)

Effects

a) When a message is transferred from the Last Configured Buffer in the Message RAM to the OBF and PLC = 127 it may happen, that at the end of the transfer the OBF word on address 00h (payload data bytes 0 to 3) is overwritten with unexpected data (see Figure 1)

b) When a message is transferred from IBF to the Last Configured Buffer in the Message RAM, it may happen, that at the end of the transfer of the Data Section one additional write access overwrites the following word, which may be the first word of another message's Data Section in the Message RAM (see Figure 2)

Workaround 1

Leave at least one unused word in the Message RAM between Header Section and Data Section.

Workaround 2

Ensure that the Data Section directly following the Header Partition is assigned to a transmit buffer.

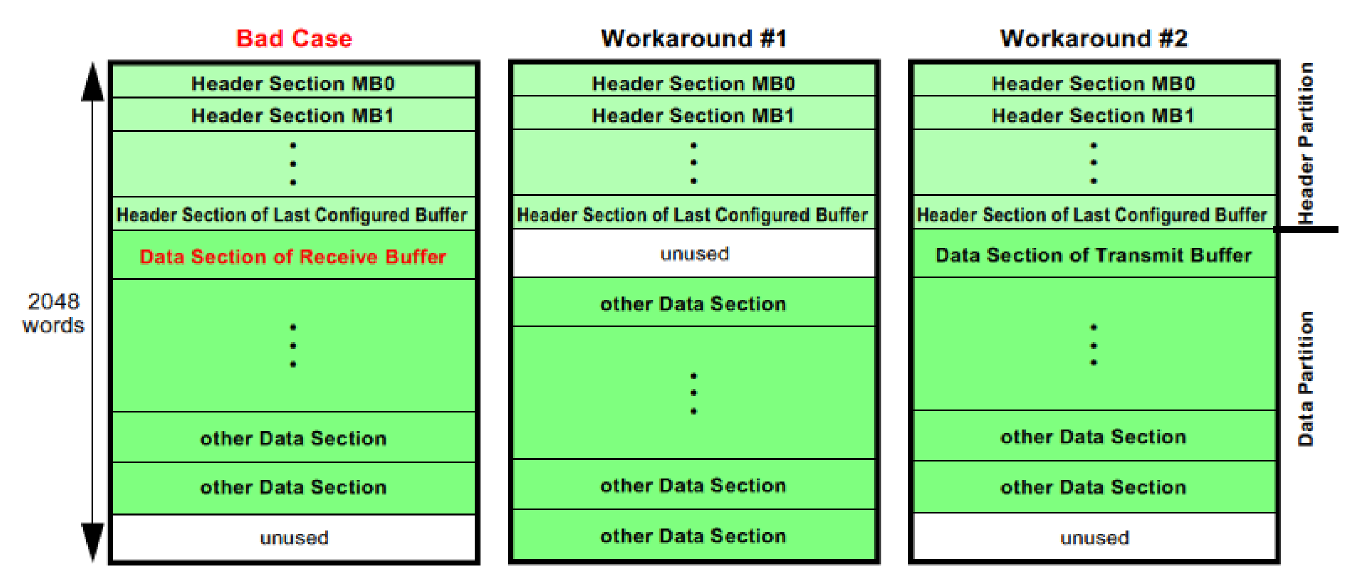


Figure 3 Message RAM configurations

2.83 [GTM_AI.132] GTM_TOP level: AEI write to BRIDGE_MODE register can result in blocking of AEI configuration interface

Description

If the GTM bus bridge operates in MSK_WR_RESP=1 mode, a requested change of the GTM_IP bridge mode (Bit BRG_MODE) can result in blocking of the bus interface.

Scope

All AEI protocols.

Effects

GTM Bus interface does not issue aei_ready/aei_response_ready which could lead to bus timeout of the serving bus master.

Workaround

Ensure that the write command to the BRIDGE_MODE register bit BRG_MODE which switches the mode of the bridge (ASYNC/SYNC) is assigned only when in addition the bit BRG_RST is set to '1'.

2.84 [GTM_AI.141] TIM: Incorrect data captured to GPR registers and routed via ARU when EGPRi_SEL,GPRi_SEL= 100 in TIM channel mode TIEM, TPWM, TIPM, TPIM, TGPS

Description

In case of a TIM channel capture event issued by a rising edge at TIM[i]_CH[x]_FOUT the capturing of the TIM[i]_CH[x]_ECNT register to the TIM[i]_CH[x]_GPRi register is incorrect. The captured value will be ECNT_REG+2; bit 0 (signal level) will be 0. The correct operation would be to capture ECNT_REG+1; bit 1 (signal level) would be 1.

Scope

TIM.

***Note**: The effects described below related to the ARU do not apply to TC23x .. TC21x devices (no ARU is implemented).*

Effects

a) Inconsistency of ARU signal level bit and bit[0] of ARU word which shows the captured ECNT.

b) Reading of TIM[i]_CH[x]_GPRi shows inconsistency when comparing bits [31:24] to [7:0]. At the point in time of capture event the bits [31:24] contain the correct value and are subject to be changed with new incoming edge.

Workaround

a) When using captured data via ARU routing the correct data can be reconstructed by:

IF ARU_SIGNAL_LEVEL ==1 AND ARU_DATA[0] == 0 THEN ARU_DATA = ARU_DATA -1;

b) When reading TIM[i]_CH[x]_GPRi by configuration interface the data can be corrected as long as there is no GPR overflow and no new edge by:

IF TIM[i]_CH[x]_GPRi[24] == 1 AND TIM[i]_CH[x]_GPRi[0] == 0 THEN TIM[i]_CH[x]_GPRi[23:0] = TIM[i]_CH[x]_GPRi[23:0] -1

2.85 [GTM_AI.142] TIM: Incorrect data captured to GPR registers and routed via ARU when EGPRi_SEL,GPRi_SEL= 100 in TIM channel mode TBCM

Description

In case of a TIM channel capture event issued by an input pattern match to condition TIM[i]_CH[x]_CNTS the capturing of the TIM[i]_CH[x]_ECNT register to the TIM[i]_CH[x]_GPRi register can be incorrect. Starting at t=0 with counter value ECNT_REG(t=0), the captured values of two consecutive edges can be ECNT_REG(t=0)+2 followed by ECNT_REG(t=0)+2 instead of ECNT_REG(t=0)+1 followed by ECNT_REG(t=0)+2.

Scope

TIM.

Note: *The effects described below related to the ARU do not apply to TC23x .. TC21x devices (no ARU is implemented).*

Effects

a) In 2 following ARU transfers the ARU word which shows the captured ECNT do not increment by 1.

b) Reading of TIM[i]_CH[x]_GPRi shows inconsistency between [31:24] and [7:0]

Workaround

a) Ignore captured data via ARU and build with MCS independent counter which increments on each ARU transfer.

b) When reading TIM[i]_CH[x]_GPRi by configuration interface use only TIM[i]_CH[x]_GPRi[31:24] as EDGE counter; don't use TIM[i]_CH[x]_GPRi[23:0].

2.86 [GTM_AI.143] GTM_TOP level: AEI pipelined write to GTM_BRIDGE_MODE register directly after setting aei_reset='0' can result in blocking of AEI configuration interface

Description

If the GTM bus bridge is reset with aei_reset= '0' (this means reset by application or module/kernel reset) and the next AEI transfer is a write command to GTM_BRIDGE_MODE register the AEI configuration interface can be blocked.

Scope

AEI pipelined protocol.

Effects

GTM Bus interface does not issue aei_ready which could lead to bus timeout of the serving bus master.

Workaround

Ensure that after setting aei_reset to inactive state (this means after resetting the GTM by application or module/kernel reset) the next command must be a read to any other register except GTM_BRIDGE_MODE. Issue desired write to GTM_BRIDGE_MODE register afterwards.

2.87 [GTM_AI.144] TIM: TIM interrupts as trigger source from TIM to TOM/ATOM not functional

Description

According to specification one could select with the configuration bits EXT_CAP_SRCx(2:0) of register TIM[i]_CH[x]_ECTRL one of six TIM channel x+1 interrupts as a source for signal TIM_EXT_CAPTURE(x).

The signal is used internally in TIM channel x and forwarded to a corresponding ATOM/TOM channel.

For the signal path TIM_EXT_CAPTURE(x) which is forwarded to ATOM/TOM the selection is incorrect for the values of EXT_CAP_SRCx(2:0) = 000, 010, 100, 101, 110, 111.

Only the selection of TIM_IN(x-1), TIM_IN(x) or AUX_IN(x) is possible with the values EXT_CAP_SRCx(2:0) = 001 or 011.

For the signal path TIM_EXT_CAPTURE(x) which is used inside TIM channel x, the selection works as specified.

Scope

TIM.

Effects

The selection of an interrupt of TIM channel x+1 by EXT_CAP_SRCx(2:0) = 000, 010, 100, 101, 110, 111 to trigger corresponding TOM/ATOM channel leads to erroneous trigger behavior.

As a result the TOM/ATOM does not react on the intended interrupt.

Workaround

None.

Do not use the configuration EXT_CAP_SRCx(2:0) = 000, 010, 100, 101, 110, 111.

2.88 [GTM_AI.153] TIM: Incorrect data captured to CNTS register when TIM channel operates in mode TPWM or TPIM and CNTS_SEL = 1 and selected CMU_CLK ≠ sys_clk

Description

In case of CNTS_SEL = 1 and TIM_MODE = TPWM or TPIM in the CNTS_REG register the value of TBU_TS0 shall be captured. This does not happen when the selected CMU_CLK ≠ sys_clk.

Scope

TIM.

Effects

Unexpected values in CNTS_REG.

Workaround

Setup the TIM channel to operate on a CMU_CLK (Divider =1) which is identical to sys_clk. Please notice that the measurement with TIM_CNT has resolution of sys_clk.

2.89 [GTM_AI.154] TOM: Incorrect duty cycle in PCM mode (bit reversed mode)

Description

The generated duty cycle on the TOM output in PCM mode is always one smaller than the configured value in the CM1 register. So if the value 1 is configured, a duty cycle of 0% will be generated. Configuring the max value (0xFFFF) in the CM1 register results in a duty cycle of max-1. Expected is 100% duty cycle in this case. A zero in CM1 register results in 100% duty cycle.

Scope

TOM.

Effects

Unexpected duty cycle in PCM mode.

Workaround

Configure always the value for the expected duty cycle in the CM1 register with expected duty cycle + 1.

To get 0% duty cycle, value 1 has to be configured. To get 100% duty cycle, 0 has to be configured to CM1 register while CM0 is always configured with max. value of 0xFFFF. Configuring CM0=0x1000 and CM1=0xFFFF will also get a duty cycle of 100%.

2.90 [GTM_AI.157] CMU: Incorrect AEI status by writing 1 to bit 24 of register CMU_CLK_6/7_CTRL

Description

If according to GTM device configuration no DPLL is available, bit 24 of register CMU_CLK_6_CTRL and CMU_CLK_7_CTRL is reserved.

Erroneously, writing a '1' to bit 24 is possible and leads to AEI status 0.

Scope

CMU: GTM device configurations without DPLL.

Effects

No functional influence to specified GTM.

After writing a ‘1’ to bit 24 of register CMU_CLK_6_CTRL or CMU_CLK_7_CTRL, a ‘1’ is read back from this register bit.

Writing a ‘1’ to bit 24 of register CMU_CLK_6_CTRL or CMU_CLK_7_CTRL leads to AEI status 0.

Workaround

Do not write ‘1’ to bit 24 of register CMU_CLK_6/7_CTRL.

2.91 [GTM_AI.163] TIM: timeout signaled when TDU unit is reenabled

Description

In the following situation an undesired timeout event is signaled:

After stopping the TDU the TO_CNT bit-field will have an arbitrary value TO_CNT0 <= TOV0 bit-field. Assume TOV will be reconfigured to value TOV1 with TOV1 <= TO_CNT0. If the TDU will be enabled again by writing to TOCTRL a value !=0 and at the same time the TCS selected CMU_CLK has an active edge an unintended timeout is signaled. This results due to the fact that for one clock cycle TO_CNT0 >= TOV1.

Scope

TIM.

Effects

Unexpected timeout event when TIM TDU is enabled.

Workaround

If TDU unit has to be reenabled with a TOV value TOV1 which is less than the previous one in use TOV0 (2 alternatives are available):

a) Wait with disabling TDU until condition TOV1 > TO_CNT is fulfilled. Configure TOV with TOV1 reenable TDU Unit.

b) Disable TDU; if TOV1 <= TO_CNT write TOV with FF_H; enable TDU unit; reconfigure TOV to desired value TOV1.

2.92 [GTM_AI.164] TIM: capturing of data into TIM[i]_CH[x]_CNTS with setting CNTS_SEL=1 not functional in TPWM and TPIM mode

Description

If CNTS_SEL=1 is selected and a new input edge is signaled by the TIM Filter unit while the selected CMU_CLK has no rising edge the register TIM[i]_CH[x]_CNTS will capture data TIM[i]_CH[x]_CNT instead of TBU_TS0.

Scope

TIM.

Effects

Captured data in TIM[i]_CH[x]_CNTS is not as expected.

Workaround

a) Select with CLK_SEL a CMU_CLK which is identical to sys_clk (clock divider=1 applied in CMU channel and for global fractional divider).

b) Use TIEM mode to capture TBU_TS0 for rising and falling input edges.

c) PWM mode: Use CNTS_SEL=0 with CMU_CLK source selected as in use for TBU_TS0 counting. Capture with EGPR0_SEL=0, GPR0_SEL=0 in GPR0_REG TBU_TS0 and with EGPR1_SEL=0, GPR1_SEL= 3 in GPR1_REG CNT. Calculate the desired timestamp with GPR0_REG - GPR1_REG + CNTS_REG.

2.93 [GTM_AI.181] TIM: Incorrect signal level bit ECNT[0] in mode TIEM, TPWM, TIPM, TPIM, TGPS

Description

In case of re-enabling a previously disabled TIM channel the bit ECNT[0] might not reflect the actual signal level of the corresponding input TIM[i]_CH[x]_FOUT until the next input edge occurs. This situation can only occur if between disabling and re-enabling the ECNT register is not read.

Scope

TIM.

Effects

Inconsistency of input signal level with ECNT bit[0].

Workaround

- After disabling the TIM channel, ensure that the ECNT register is read at least once and afterwards the TIM channel can be re-enabled
- Before re-enabling a TIM channel, issue a TIM channel reset and reconfigure the TIM channel control registers

2.94 [GTM_AI.202] (A)TOM: no CCU1 interrupt in case of CM1=0 or 1 and RST_CCU0=1

Description

In case of channel x has configuration of RST_CCU0=1 (i.e. CN0 is reset by trigger input) and CN0 counts from 0 to MAX:

- if CM1=0, CM0>0 -> no CCU1 interrupt is generated
- if CM1=1, CM0=MAX+1 -> only one time a CCU1 interrupt is generated

Scope

TOM / ATOM SOMP mode.

Effects

For the described configuration no CCU1 interrupt is generated.

Workaround

Use for triggering channel y (i.e. the channel that triggers on channel x the reset of counter CN0) the configuration of CM0=MAX, CM1=1.

In case of duty cycle configuration of CM1=0 and CM0>0 on channel x use instead of CCU1 interrupt on channel x the CCU0 interrupt of triggering channel y.

In case of duty cycle configuration of CM1=1 and CM0=MAX+1 on channel x use instead of CCU1 interrupt on channel x the CCU1 interrupt of triggering channel y.

2.95 [GTM_AI.205] TIM: unexpected CNTS register update in TPWM OSM mode

Description

If OSM=1 and TIM_MODE=”000” (TPWM) an active edge defined by DSL will stop the measurement. In case of an inactive edge following after 1 GTM system clock cycle the active edge the CNTS register will be reset unexpected.

Scope

TIM.

Effects

Unexpected CNTS register content.

Workaround

a) Use CMU clock in TIM channel with frequency lesser than system clock.

b) Enable filter and configure filter parameter in a way that two consecutive edges will never occur with distance of GTM system clock.

2.96 [GTM_AI.209] TOM/ATOM: no update of CM0/CM1/CLK_SRC via trigger signal from preceding instance if selected CMU_CLKx is not SYS_CLK

Description

The trigger signal between (A)TOM instances (for example signal TOM_TRIG_[i]) is registered between each TOM and between each 2nd ATOM and with this delayed by one SYS_CLK period to break long combinational path.

For each register in the trigger path between (A)TOM instance i and the succeeding (A)TOM instance i+1, this trigger from instance i does not trigger the update of register CM0, CM1 and CLK_SRC with content of SR0, SR1 and CLK_SRC_SR if the triggered channel of instance i+1 is not running with a selected CMU_CLKx = SYS_CLK.

Scope

TOM/ATOM.

Effects

In the described configuration no update of CM0, CM1 and CLK_SRC is done although the update is enabled by register TOM[i]_TGC[y]_GLB_CTRL / ATOM[i]_AGC_GLB_CTRL.

Workaround

For each register in trigger path between (A)TOM instance i and (A)TOM instance i+1, the channel of instance i+1 that should be triggered has to use a clock of period identical to SYS_CLK period.

A second workaround could be to set up on instance i+1 a redundant channel to trigger other channel of instance i+1 like it was set up on instance i to trigger other channel. Then, start both instances synchronously by using the TBU time base comparator of AGC/TGCx unit (i.e. the ATOM[i]_AGC_ATC_TB / TOM[i]_TGC[y]_ACT_TB register).

2.97 [GTM_AI.260] TOM/ATOM: Async. update in SOMP mode with CM1=0 and selected CMU clock unequal sys_clk not functional

Description

***Note**: In **TC23x/TC22x/TC21x** devices, this problem relates to the following scenario in TOM: Async. update with CM1=0 and selected CMU_FXCLK unequal to sys_clk not functional.*

An asynchronous update of the duty cycle by writing value 0 to CM1 register while a CMU clock unequal sys_clk is selected is not working. It is expected that the output signal level is set immediately to inactive level but it will remain at actual level.

Scope

TOM/ATOM.

Effects

The output signal level is not set to inactive level. It will remain at actual level.

Workaround

Writing value 1 instead of 0 to CM1 register will set the output to inactive level in the actual generated PWM period.

If the duty cycle duration should be zero also for the following period, the user has to take care, that the CM1 register is loaded with a 0 at the beginning of the next PWM period.

Otherwise, if the content of register CM1 remains at 1, a peak of one clock cycle with the selected CMU clock will be observed, with the next PWM period.

2.98 [GTM_AI.270] (A)TOM: output signal is postponed one period for the values CM0=1 and CM1>CM0 if CN0 is reset by the trigger of a preceding channel (RST_CCU0=1)

Description

If counter CN0 is reset by the trigger of a preceding channel (bit RST_CCU0 of register TOM[i]_CH[x]_CTRL/ ATOM[i]_CH[x]_CTRL is set), then the value of CM0 defines the signal edge to SL (signal level), whereas CM1 defines the edge to !SL (inverted signal level).

If - in this case - the value 1 is configured for the output edge to SL (CM0=1) and CM1 is configured to greater than CM0 (CM1>CM0) the expected output edge will be postponed by one period.

Scope

TOM, ATOM SOMP mode

Effects

The expected output edge will be postponed by one period.

Workaround

Instead of configuring CM0=1 it is also possible to configure CM1=1 and to invert SL to get the expected edge at counter value 1 (CN0=1).

2.99 [GTM_AI.298] TOM/ATOM: wrong output behaviour in SOMP oneshot mode when oneshot pulse is triggered by TIM_EXT_CAPTURE(x)

Description

If TOM/ATOM is configured in SOMP oneshot mode (OSM = 1) and the oneshot trigger is configured to TIM_EXT_CAPTURE(x) (OSM_TRIG = 1, EXT_TRIG = 1) the output behaviour is not as expected depending on the selected CMU clock.

1. If the selected CMU clock is configured to sys_clk (ATOM: CMU_CLK_[z]_CTRL = 0, TOM: CMU_FXCLK0 used) no initial oneshot period (CN0 is set to zero and then counts until CN0 >= CM0) is executed and the output is set to SL immediately and not as expected after the first initial period
2. If the selected CMU clock is configured to CMU_CLK_[z]_CTRL > 0 (ATOM)/CMU_FXCLK[1..n] (TOM) then an initial period is executed but the output is set immediately to SL and not as expected when the second oneshot period starts

Scope

TOM/ATOM SOMP oneshot mode

Effects

The TOM/ATOM output is set immediately to SL and not as expected with a delay of the first initial oneshot period.

Workaround TC2xx

For GTM generation v2 no workaround is available.

If it is possible configure the selected CMU clock to sys_clk period. Then the generated oneshot pulse length is correct but without executing of the initial period.

Workaround TC3xx

For GTM generation v3 following workaround is possible:

Use up/down counter mode (UDMODE > 0) instead of up counter mode (UDMODE = 0).

It has to be taken into account that in up/down counter mode the oneshot cycles ends if the counter CN0 counts down and value zero is reached.

A second trigger of TIM_EXT_CAPTURE(x) in the up counting phase will be ignored but a second trigger while the counter CN0 counts down will trigger the next oneshot cycle, which will be executed directly afterwards without the initial period.

2.100 [GTM_AI.299] TOM/ATOM: wrong output behaviour in SOMP oneshot mode when oneshot pulse is triggered by trig_[x-1]

Description

If TOM/ATOM is configured in SOMP oneshot mode (OSM = 1) and the oneshot trigger is configured to trigger signal from trigger chain trig_[x-1] (OSM_TRIG = 1, EXT_TRIG = 0) the output signal is set immediately to SL and not as expected after a delay of the first initial oneshot period (CN0 counts from 0 until it reaches the value of CM0). The first initial oneshot period isn't executed.

Scope

TOM/ATOM SOMP oneshot mode

Effects

The TOM/ATOM output is set immediately to SL and not as expected with a delay of the first initial oneshot period.

Workaround TC2xx

For GTM generation v2 no workaround is available.

If it is possible work without the initial period for GTM generation v2 because the generated pulse length is correct.

Workaround TC3xx

For GTM generation v3 and later following workaround is possible:

Use up/down counter mode (UDMODE > 0) instead of up counter mode (UDMODE = 0).

It has to be taken into account that in up/down counter mode the oneshot cycles ends if the counter CN0 counts down and value zero is reached.

A second trigger of from trigger chain by trig_[x-1] in the up counting phase will be ignored but a second trigger while the counter CN0 counts down will trigger the next oneshot cycle, which will be executed directly afterwards without the initial period.

2.101 [GTM_AI.336] GTM Bus Bridge: Incorrect AEI access execution in case the previous AEI access was aborted with the access timeout abort function

Description

In case the GTM internal AEI access timeout abort function is in use (GTM_CTRL.TO_VAL != 0 and GTM_CTRL.TO_MODE=1), a following AEI access can be corrupted:

a) A write access might not be executed (register/ memory not written to the specified value)

b) A read access can return random data (read value does not reflect the content of the addressed register / memory).

Hint: As a timeout based abort of a GTM register access is assumed to be an error scenario, the internal state of the GTM might be exposed. To ensure the proper behavior after such a severe incident, the GTM IP should be re-initialized as part of a recovery action on system level.

Scope

CPU interface accesses

Effects

Read access returns random data.

Write access does not change the content of the target address.

Workaround

Do not use the AEI access abort mode, use the observe mode instead (Set GTM_CTRL.TO_MODE=0).

Enable additionally the timeout observe IRQ by setting GTM_IRQ_EN.AEI_TO_XPT_IRQ=1 to invoke higher level recovery mechanisms for GTM re-initialization.

(e.g. abort the pending access to the GTM and re-initialize the GTM_IP from hardware reset).

2.102 [GTM_AI.340] TOM/ATOM: Generation of TRIG_CCU0/TRIG_CCU1 trigger signals skipped in initial phase of A/TOM SOMP one-shot mode

Description

Configuration in use:

- A/TOM[i]_CH[x]_CTRL.OSM=1
- A/TOM[i]_CH[x]_CTRL.OSM_TRIG=0
- A/TOM[i]_CH[x]_CTRL.UDMODE=00
- ATOM[i]_CH[x]_CTRL.MODE=10

Expected behavior

The generation of one-shot pulses in A/TOM can be initiated by a write to CN0. In this case the pulse generation comprises of an initial phase where the signal level at A/TOM output is inactive followed by a pulse. The duration of the initial phase can be controlled by the written value of CN0, where the duration is defined by CM0-CN0. After the counter CN0 reaches the value of CM0-1, the pulse starts with its active edge, CN0 is reset, and starts counting again. When CN0 reaches CM1-1, the inactive edge of the pulse occurs. Due to the fact, that the capture compare units CCU0 and CCU1 compare also in the initial phase of the pulse generation, the trigger conditions for these comparators apply also in this initial phase. Thus, the TRIG_CCU0 and TRIG_CCU1 signals also occur in the initial phase of the one-shot pulse. When these trigger signals are enabled in the A/TOM[i]_CH[x]_IRQ_EN, an interrupt signal is generated by A/TOM on the CCU0TC and CCU1TC trigger conditions and the corresponding A/TOM[i]_CH[x]_IRQ_NOTIFY bits are set.

Observed behavior

For certain start values of CN0 and dependent on the history of pulse generation, the trigger signals TRIG_CCU0 and TRIG_CCU1 are skipped. As a consequence, this can led to missing interrupts CCU0TC and CCU1TC on behalf of their missing trigger signals TRIG_CCU0 and TRIG_CCU1.

For the first pulse generation after enabling the channel, all trigger signals TRIG_CCU0 and TRIG_CCU1 appear as expected and described in the section expected behavior. If the channel stays enabled and a new value CN0 is written to trigger a subsequent one-shot pulse, the TRIG_CCU0/TRIG_CCU1 triggers in the initial phases of subsequent one-shot pulses are skipped under the following conditions:

- For TRIG_CCU0 trigger: if the one-shot pulse is started by writing a value to CN0 greater or equal to CM0-1
- For TRIG_CCU1 trigger: if the one-shot pulse is started by writing a value to CN0 greater or equal to CM1-1

Scope

TOM/ATOM

Effects

Missing TRIG_CCU0 and TRIG_CCU1 trigger signals in initial phase of subsequent pulses in A/TOM one-shot mode, when one shot-mode is started with writing to CN0 values greater equal CM0-1 or CM1-1.

Workaround 1

Disabling, resetting (channel reset), re-enabling and initializing of the channel between each one-shot pulse will ensure the correct behavior of CCU0TC and CCU1TC interrupt source.

Workaround 2

Starting a new one-shot pulse by writing twice the counter CN0 whereas the first value, which is written to CN0 should be zero followed by the value which defines the length of the initial phase.

Be aware that in this case, the total length of the initial phase until the pulse is started, is influenced by the time between the two write accesses to CN0.

2.103 [GTM_AI.341] TOM/ATOM: False generation of TRIG_CCU1 trigger signal in SOMP one-shot mode with OSM_TRIG=1 when CM1 is set to value 1

Description

Configuration in use:

- A/TOM[i]_CH[x]_CTRL.OSM=1
- A/TOM[i]_CH[x]_CTRL.OSM_TRIG=1
- A/TOM[i]_CH[x]_CTRL.UDMODE=00
- ATOM[i]_CH[x]_CTRL.MODE=10

Expected behavior

The generation of one-shot pulses in A/TOM can be initiated by the trigger event TRIG_[x-1] from trigger chain or by TIM_EXT_CAPTURE(x) trigger event from TIM, whereas the counter CN0 is reset to zero and starts counting. In this case the pulse generation comprises of an initial phase where the signal level at A/TOM output is inactive followed by a pulse. The duration of the initial phase is always as long until the counter CN0 reaches CM0-1.

After the counter CN0 reaches the value of CM0-1, the pulse starts with its active edge, CN0 is reset, and starts counting again. When CN0 reaches CM1-1, the inactive edge of the pulse occurs. Due to the fact, that the capture compare units CCU0 and CCU1 compare also in the initial phase of the pulse generation, the trigger conditions for these comparators apply also in this initial phase. Thus, the TRIG_CCU0 and TRIG_CCU1 signals also occur in the initial phase of the one-shot pulse. When these trigger signals are enabled in the A/TOM[i]_CH[x]_IRQ_EN, an interrupt signal is generated by A/TOM on the CCU0TC and CCU1TC trigger conditions and the corresponding A/TOM[i]_CH[x]_IRQ_NOTIFY bits are set.

Observed behavior

If the compare register CM1 is set to 1 and a new one-shot pulse is triggered, two effects can be observed:

- The first observed behavior is that the capture compare unit doesn't generate the TRIG_CCU1 trigger signal in the initial phase of the one-shot cycle
- The second observed behavior is that at the end of the operation phase of the one-shot cycle, where CN0 reaches CM0-1 a second time, the capture compare unit generates a TRIG_CCU1 trigger signal which is not expected at this point in time

Scope

TOM/ATOM

Effects

Missing TRIG_CCU1 trigger signal in initial phase of the one-shot cycle and unexpected TRIG_CCU1 trigger signal at the end of the operation phase of the one-shot cycle.

Workaround

Instead of using value 1 for CM1 it could be possible to generate the same pulse length by using a higher CMU_FXCLK/CMU_CLK frequency. Then, to get the same pulse length, the value of CM1 has to be multiplied by the difference of the two CMU_FXCLK/CMU_CLK frequencies.

Be aware that this workaround is only possible, if you are not already using the CMU_FXCLK(0) because there is no higher CMU_FXCLK frequency to select.

Example for TOM : Instead of using CMU_FXCLK(1), which has the divider value 2**4, use CMU_FXCLK(0), which has the divider value 2**0. In this case, CM1 has to be configured with value 2**4 minus 2**0 which is equal to 2**4=16.

Hint : To get the same length of period, which defines the length of the initial phase, the value for the period in CM0 has to be multiplied by the same value.

A second limitation is that the maximum length of the period, which is configured in CM0, is limited. Using a higher CMU_FXCLK/CMU_CLK frequency reduces the maximum possible period.

2.104 [GTM_AI.347] TOM/ATOM: Reset of (A)TOM[i]_CH[x]_CN0 with TIM_EXT_CAPTURE are not correctly synchronized to selected CMU_CLK/CMU_FXCLK

Description

To reset the counter (A)TOM[i]_CH[x]_CN0 (SOMP mode in ATOM), the input signal TIM_EXT_CAPTURE can be used by configuration of (A)TOM[i]_CH[x]_CTRL.EXT_TRIG=1 and (A)TOM[i]_CH[x]_CTRL.RST_CCU0=1.

The reset of the counter (A)TOM[i]_CH[x]_CN0 should happen synchronously to the internal selected CMU clock CMU_CLK/CMU_FXCLK. Therefore a synchronisation stage is implemented to synchronize the input signal TIM_EXT_CAPTURE to the internal selected CMU clock CMU_CLK/CMU_FXCLK.

It can be observed, that the reset of the counter is done immediately with the occurrence of the input signal TIM_EXT_CAPTURE and not as expected synchronously to the selected CMU clock enable CMU_CLK/CMU_FXCLK.

As a consequence of this, the output signal for the compare values 0 and 1 of (A)TOM[i]_CH[x]_CM1.CM1 and (A)TOM[i]_CH[x]_CM0.CM0 will not be set correctly.

Scope

ATOM, TOM

Effects

The output signal (A)TOM[i]_CH[x]_OUT is not set correctly for the compare values 0 and 1 of the operation register bitfields (A)TOM[i]_CH[x]_CM1.CM1 and (A)TOM[i]_CH[x]_CM0.CM0.

Workaround 1

Do not use clock dividing for the affected (A)TOM channels, so the undivided cluster clock is used. For this configure the control registers in the CMU and CCM to generate non-dividing CMU_FXCLK and/or CMU_CLK signals. Select within the (A)TOM the non-dividing CMU_FXCLK0 (for TOM) and/or CMU_CLK0..7 (for ATOM) via the settings for CLK_SRC in the (A)TOM[i]_CH[x]_CTRL register(s).

Workaround 2

Avoid the compare values 0 and 1 for the operation register bitfields (A)TOM[i]_CH[x]_CM1.CM1 and (A)TOM[i]_CH[x]_CM0.CM0.

2.105 [GTM_AI.361] IRQ: Missing pulse in single-pulse interrupt mode on simultaneous interrupt and clear event

Description

In single-pulse interrupt mode ([MODULE]_IRQ_MODE = 0b11) only the first interrupt event of the interrupt bits of the interrupt notify register inside this module generates a pulse on the output signal IRQ_line, if the associated interrupt is enabled ([MODULE]_IRQ_EN=1). All further interrupt events have no effect on the output signal IRQ_line until all enabled interrupts are cleared, except when an interrupt and a clear event (HW_clear or a SW_clear) occur at the same time.

Expected behaviour

On simultaneous occurrence of an interrupt and clear event, a pulse on the output signal IRQ_line is generated.

Observed behaviour

If the associated notify register bit of the interrupt event is not set and another bit of the same notify register is set and this interrupt is enabled, no pulse on the output signal IRQ_line is generated.

All modules ([MODULE]) are affected by this ERRATUM, which are able to generate interrupts and which have multiple interrupt sources which are ORed to the output. Not affected are the modules DPLL and ARU.

Scope

IRQ

Effects

Missing pulse on interrupt signal IRQ_line.

All modules, which deliver an interrupt signal and have more than one internal interrupt source which are ORed are affected. The only exceptions are the modules ARU and DPLL.

Workaround

On a SW clear prevent HW clear events and read the interrupt notify register to check on new interrupts without a received interrupt pulse on IRQ_line. In this case repeat the SW clear step to enable interrupt generation again.

When disabling the HW clear is not an option refrain from using the single-pulse interrupt mode.

2.106 [GTM_AI.380] (A)TOM: potentially wrong output signal in case of RST_CCU0=1 and CM0=1 on triggered channel in SOMP mode

Description

When the reset of (A)TOM_CHx_CN0 of a TOM or ATOM channel is triggered by a preceding channel or assigned TIM module (RST_CCU0=1) and the ATOM channel is configured in SOMP mode, the CM0 value defines the edge to SL and CM1 defines the edge to !SL.

Expected behavior

When SR0 is configured to '1', and CM0 is updated with SR0=1 on trigger signal coming from previous channel, an edge to SL is expected, when CN0=CM0=1.

Observed behavior

When CM0 is updated synchronously from SR0 for the next period, and CM0>1 at the actual period, no edge to SL is generated when CM0=CN0=1 for the first period after CM0=1 becomes active (was updated to CM0=1 from SR0).

Scope

TOM, ATOM

Effects

For the configuration RST_CCU0=1 and CM0=1, CM0 < CM1 no edge is generated for the first period, after CM0 is updated from SR0 with ’1’ and CM0 > 1 in the period before.

Workaround

In addition to configuring SR0=1 and letting the (A)TOM channel update CM0 with ‘1’ at the start of the next period, a hot reconfiguration of CM0=1 can be done. However, the hot reconfiguration needs to be done after the edge to SL was performed in the actual period. Otherwise the CM0 value would be overwritten by ‘1’ and the edge to SL would be generated immediately after hot reconfiguration and not at the intended old CM0 value.

The workaround is applicable where the system can update the CM0 value in time; otherwise the setting of CM0=1 should not be used.

2.107 [GTM_AI.408] (A)TOM-RTL: Missing edge on output signal (A)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.

Configuration for TOM:

TOM[i]_CH[x]_CTRL.UDMODE = 0

TOM[i]_TGC[g]_FUPD_CTRL.FUPD_CTRL[k] = 10_B

TOM[i]_TGC[g]_FUPD_CTRL.RSTCN0_CH[k] = 10_B

Configuration for ATOM:

ATOM[i]_CH[x]_CTRL.MODE = 10_B (SOMP mode)

ATOM[i]_CH[x]_CTRL.UDMODE = 0

ATOM[i]_AGC_FUPD_CTRL.FUPD_CTRL[k] = 10_B

ATOM[i]_AGC_FUPD_CTRL.RSTCN0_CH[k]= 10_B

Expected behavior

After the counter (A)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 the SL value (ATOM[i]_CH[x]_CTRL_SOMP.SL, TOM[i]_CH[x]_CTRL.SL), and after the counter reaches (A)TOM[i]_CH[x]_CM1.CM1, an edge on (A)TOM_OUT to the 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 (A)TOM_OUT to the 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 (A)TOM_OUT holds its last value.

A second observation is in case 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 (A)TOM_OUT to the inverted SL value (ATOM[i]_CH[x]_CTRL_SOMP.SL, TOM[i]_CH[x]_CTRL.SL) when (A)TOM[i]_CH[x]_CN0.CN0 reaches (A)TOM[i]_CH[x]_CM1.CM1 does not happen.

Scope

TOM, ATOM

Effects

Missing edge and false output signal level on (A)TOM_OUT.

Workaround

None.

2.108 [GTM_AI.410] GTM_AEI: 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. The maximum number of non-executed transactions depends on the buffer depth (BRIDGE_BUFF_DPT).

Scope

GTM_AEI

Effects

Write transaction is signaled to be accepted but will never be executed.

Workaround

Issue a read access to any address after the soft reset.

2.109 [GTM_AI.411] A change of the BRIDGE_MODE register might be delayed indefinitely

Description

After a write access to the BRIDGE_MODE register, the bit-fields BRG_MODE and BYPASS_SYNC will not be updated 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 as a result prevents the actual update of the described bit-fields.

***Note**:* *Bit-field BYPASS_SYNC is not specified for TC2xx.*

Scope

GTM_AEI

Effects

Frequently polling the BRIDGE_MODE register ends in a deadlock.

Workaround 1

After every failed attempt to read back the new values, increase the wait time before issuing the next read transaction.

Workaround 2

Use standard mode (which is entered by setting AEI_PIPE and AEI_SPLIT at zero while asserting AEI_SEL) to write and read back the affected bits.

***Note**: This workaround is only possible in devices without AXIS.*

2.110 [GTM_AI.419] 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 = {010_B, 011_B, 101_B, 110_B}. 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 behavior 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 behavior 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 behavior 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 behavior for TSSM mode

The register 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 = {000_B, 001_B, 100_B}) are not affected.*

Scope

TIM

Effects

TIM[i]_CH[x]_CNT register is not reset and the wrong values could be captured into TIM[i]_CH[x]_GPR0 and TIM[i]_CH[x]_GPR1 registers.

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 = 101_B (TGPS)
   - ISL = 1
   - OSM = 1
   - ARU_EN = 0
   - select a fast CMU_CLK_RES, e.g. CLK_SEL = 000_B
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

2.111 [GTM_AI.421] 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.

***Note**: Please also check on errata GTM_AI.487 and GTM_AI.488.*

Scope

GTM_AEI

Effects

Transaction not terminated according to protocol, user might be stuck waiting for AEI_READY to be set.

Workaround

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

2.112 [GTM_AI.429] 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 behavior

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 behavior

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.

Scope

TIM

Effects

The TIM[i]_CH[x]_IRQ_NOTIFY.GLITCHDET bit is not set. Thus, no interrupt is triggered. Furthermore, the external capture source EXT_CAPTURE(x) is not triggered if its source is set to TIM_GLITCHDET_IRQ.

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.

2.113 [GTM_AI.430] 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 behavior

When the input signal level changes, the filter counter should not increment.

Observed behavior

When the input signal level changes, the filter counter increments by one and is not reset.

Scope

TIM

Effects

If an input edge occurs during the acceptance time, the following output signal change will happen one or more selected CMU clock cycles earlier than expected. This depends on the initial configuration and the reconfiguration of the filter mode and the filter counter threshold. If the filter mode for both edges is configured to immediate edge propagation and both filter counter thresholds are set to zero, the counter falsely can count up to a higher value than one without resetting. If one or both filter modes and/or thresholds are reconfigured during the application, the higher count of the filter counter can lead to a difference of more than one CMU clock cycle between the expected and actual output signal change at the next occurring input edge. If only one filter counter threshold is set to zero, the difference of the expected and actual output signal change is one CMU clock cycle.

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.

2.114 [GTM_AI.431] TIM: Glitch detection interrupt event of filter is not a single cycle pulse

Description

Configuration: The TIM filter must be enabled by setting TIM[i]_CH[x]_CTRL.FLT_EN = 1.

Expected behavior

As long as the filter threshold is not reached and the input signal level changes unexpectedly, the glitch detection interrupt event signal (TIM_GLITCHDET_IRQ) should have a single cycle HIGH pulse.

Observed behavior

When the input signal level changes unexpectedly for longer than one clock cycle, the glitch detection interrupt event signal (TIM_GLITCHDET_IRQ) is HIGH as long as the unexpected signal change is present.

Scope

TIM

Effects

- Effect 1: The longer lasting HIGH signal of the glitch detection interrupt event signal (TIM_GLITCHDET_IRQ) may lead to an unexpected behavior within the GTM only if TIM_GLITCHDET_IRQ is used for the external capture signal EXT_CAPTURE(x).
- Effect 2: If the related interrupt notify register (TIM[i]_CH[x]_IRQ_NOTIFY) is cleared by software while the TIM_GLITCHDET_IRQ signal is still HIGH, the interrupt will unexpectedly retrigger.

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.

2.115 [GTM_AI.462] (A)TOM: Missing CCU0TC_IRQ interrupt signal

Description

Configuration:
The channel is configured in SOMP (ATOM) up-counter mode with up/down counter mode disabled ((A)TOM[i]_CH[x]_CTRL.UDMODE=0) or not existing and triggering by a preceding channel with configuration of (A)TOM[i]_CH[x]_CTRL.RST_CCU0=1.

Expected behavior

When the counter (A)TOM[i]_CH[x]_CN0.CN0 reaches the value of (A)TOM[i]_CH[x]_CM0.CM0, the interrupt signal CCU0TC_IRQ must be triggered.

Observed behavior

In the first period after (A)TOM[i]_CH[x]_CM0.CM0 is changed to the value 0 or 1, no CCU0TC_IRQ interrupt signal is triggered.

***Note**: When the second period starts after (A)TOM[i]_CH[x]_CM0.CM0 is changed to the value 0 or 1 and stays at that value, then the CCU0TC_IRQ interrupt signal generation works correctly.*

Scope

TOM, ATOM

Effects

Interrupt signal CCU0TC_IRQ is not triggered.

Workaround

No workaround available.
It needs to be checked if the application can accept the interrupt occurring with the second period.

2.116 [GTM_AI.487] GTM_AEI: Changing BRIDGE_MODE[2:0] in pipeline mode can lead to violation of pipeline protocol

Description

The issue from erratum GTM_AI.421 (“GTM_AEI: Changing BRIDGE_MODE.MSK_WR_RSP in pipeline mode can lead to violation of pipeline protocol”) not only appears when BRIDGE_MODE.MSK_WR_RSP changes, but also when it stays '1' while the other configuration bit-fields in BRIDGE_MODE.BYPASS_SYNC and/or BRIDGE_MODE.BRG_MODE change.
Please also check on erratum GTM_AI.488

Scope

GTM_AEI

Effects

Transaction not terminated according to protocol, user might be stuck waiting for AEI_READY to be set.

Workaround

Make sure the transaction preceding the write of the mentioned BRIDGE_MODE bit-fields is a read transaction.

This workaround matches the workaround from GTM_AI.421.

2.117 [GTM_AI.488] 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_B) and the BRIDGE_MODE.MSK_WR_RSP is turned off (by writing BRIDGE_MODE[2:0]=001_B), the following transaction might be corrupted by the AEI_READY not being set.

This is an issue like in GTM_AI.421 and GTM_AI.487 but a different workaround is needed.

Scope

GTM_AEI

Effects

Transaction not terminated according to protocol, user might be stuck waiting for AEI_READY to be set.

Workaround

Change BRIDGE_MODE.MSK_WR_RSP together with setting the software reset (pipeline writing BRIDGE_MODE[16:0]=10001_H).

2.118 [GTM_AI.517] (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.CM<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.

Scope

ATOM, TOM

Effects

Missing edge on output signal (A)TOM_OUT.

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, 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.

2.119 [GTM_AI.522] (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=1_B (SOMP mode for ATOM) and the duty cycle switches from 100% duty cycle with (A)TOM[i]_CH[x]_CM0.CM0=0_H 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=0_H 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.

***Note**: 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.*

Scope

ATOM, TOM

Effects

Output signal (A)TOM_OUT remains at (A)TOM[i]_CH[x]_CTRL.SL value.

Workaround

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.

2.120 [GTM_TC.009] TBU signals not wired to debug logic

Description

The TBU signals are not wired from the GTM kernel to the OTGB interface. Therefore, TBU signals are not available for debug purposes.

The other GTM signals are connected to the debug system as specified.

***Note**: This limitation will not be present on TC23x Emulation Devices.*

2.121 [GTM_TC.012] Read Access Control by Register ODA

Description

Specific GTM registers have by default “destructive read” behavior as their normal read behavior (see section “GTM Software Debugger Support” in the GTM chapter of the User’s Manual for further details.)

Depending on the reading master and the configuration of bits DREN and DDREN in register GTM_ODA (OCDS Debug Access Register), the read can be performed “non-destructive” for debug related read operation.

According to the User’s Manual the read is performed “non-destructive” (i.e. debug related read operation)

- for all masters when ODA.DREN = 1_B
- for the Cerberus (OCDS) FPI master when ODA.DREN = 0_B and ODA.DDREN = 0_B

Problem Description

In the current implementation the read is performed “non-destructive” (i.e. debug related read operation)

- for all masters when ODA.DREN = 1_B
- for the DMA Partition 2 FPI master when ODA.DREN = 0_B and ODA.DDREN = 0_B

Workaround

The problem described above has 2 aspects:

1. For DMA Partition 2 Access to GTM

When the DMA Partition 2 FPI master is used to perform a normal (“destructive”) read of the GTM registers that by default have “destructive read” behavior as their normal read behavior, setting ODA.DREN = 0_B and ODA.DDREN = 1_B is required to avoid an unintended debug related (“non-destructive”) read access that would be caused by this issue.

2. For Cerberus (OCDS) Access to GTM

When ODA.DREN = 0_B and ODA.DDREN = 0_B, any read access of the Cerberus (OCDS) FPI master to the registers that by default have “destructive read” behavior as their normal read behavior will cause the normal (“destructive”) read behavior. To get the intended debug related (“non-destructive”) read behavior, ODA.DREN needs to be set to 1_B before each access of the Cerberus and set back to 0_B afterwards to not affect the access of other FPI masters on the registers described above.

2.122 [IOM_TC.002] Missed or spurious IOM events when pulse length exceeds Event Window counter range

Description

When using the Logic Analyzer Module (LAM) of the IOM, if the 24-bit counter for the Event Window exceeds its maximum value (0xFFFFFF) it wraps around and starts counting again from 0x0.

If the Event Window is not inverted (LAMCFG.IVW = 0_B), for example for measuring long pulses, and the edge that generates an event comes after the counter exceeded its maximum value, the event will not be generated if the counter, due to the rollover, is again below the threshold value (LAMEWS.THR), outside of the Event Window.

As an additional side effect of the wraparound, spurious events may be generated when expecting an alarm only in case of pulses that are too short, if a pulse is longer than the counter can handle.

Workaround

Avoid measuring pulses longer than the Event Window counter range.

2.123 [IOM_TC.003] Unexpected Event upon Kernel Reset

Description

If a kernel reset (via bits RST in registers KRST0/1) is performed on the IOM, an unexpected event may be signalled to the SMU.

Workaround

Before triggering a kernel reset via software, set the alarm reaction in SMU to “No Action” to avoid reaction on the unexpected event.

2.124 [IOM_TC.004] Write to IOM register space when IOM_CLC.RMC > 1

Description

If a clock divider value RMC > 1 is selected in register IOM_CLC, more than one write access may be performed to the IOM register address space within one IOM clock cycle.

This will cause unpredictable effects on the internal state for the following scenarios where two (or even multiples of 2) write accesses are performed within one IOM clock cycle to the following register groups:

- ECM registers ECMCCFG and/or ECMSELR, or
- ECM Event Trigger History registers ECMETH0 and/or ECMETH1, or
- FPC registers FPCESR, FPCCTRk and/or FPCTIMk, or
- LAM registers LAMCFGm and/or LAMEWSm

***Note**: No problem will occur for read accesses.*

Workaround

Set IOM_CLC.RMC = 1 when configuring (writing to the registers of) the IOM.

During runtime (not configuring IOM) IOM_CLC.RMC > 1 is not an issue.

2.125 [MTU_TC.005] Access to MCx_ECCD and MCx_ETRRi while MBIST disabled

Description

It is possible to access the memory controller registers MCx_ECCD and MCx_ETRRi without the need of the MBIST mode being enabled (i.e. without MTU_MEMTEST.MEMxEN = 1_B). This may be used to avoid a complete SRAM initialization on certain security relevant SRAMs.

However, when a MBIST controller is disabled (MTU_MEMTEST.MEMxEN = 0_B), there is an inevitable corner case that causes the value read/written from/to registers MCx_ECCD and MCx_ETRRi of a disabled MBIST controller to be wrong. There is also a possibility that an SPB error is triggered when accessing the MCx_ECCD and MCx_ETRRi registers if other masters concurrently use the SPB bus in this situation.

***Note**: No workaround is required to access the registers of an enabled MBIST controller.*

Workaround

When MBIST mode is disabled (MTU_MEMTEST.MEMxEN = 0_B) for a MBIST controller,

- ensure that the module kernel clock is enabled for the access to MCx_ECCD and MCx_ETRRi
- and perform a dummy write to MCx_ECCD with value $780F_H$ before any read/write access to MCx_ECCD or MCx_ETRRi

***Note**: The module kernel clock (of the module in which the SRAM is present) does not need to be enabled if it can be ensured that no concurrent SPB bus accesses by other masters (CPU, DMA, HSM, debugger, ..) to other modules are performed during the MCx_ECCD/ETRRi access while the module kernel clock is disabled.*

The module kernel clock is enabled under the following conditions:

1. For CPU memories, the clock is enabled after reset (for CPUx with x>0 even when CPUx is still in BOOT-HALT mode), when the CPU is not explicitly put into IDLE mode by software
2. For SRAMs in peripherals, the module kernel clock is enabled when the module clock is enabled via the CLC register

The value $780F_H$ has been chosen as an example based on the following use cases and assumptions:

- If error reporting is turned on (i.e. notification enable bits *ENE are set), it does not disturb the system to write back $780F_H$ to register ECCD (write back of reset values, write to read-only bits and write of 1_B to error indication bits has no effect)
- If error reporting is turned off (i.e. notification enable bits *ENE are cleared), write back of $780F_H$ to register ECCD may trigger SMU alarms (if SMU is configured). It is assumed that the corresponding errors are already known by the system since error reporting had previously been deactivated

2.126 [MTU_TC.007] Error Overflow Indication ECCD.EOV

Description

The Error Overflow Indication bit EOV in register ECCD does not work correctly in specific cases for the following modules:

E-Ray, ETH, GTM, CIF (in ADAS devices), MCDS (in emulation devices), DAM.

The problem occurs in the following cases:

- If an error (correctable, uncorrectable, address error) was detected at address 0, this error is correctly stored in the ETRR register (ETRR.ADDR = 0x0, flags SERR, CERR, UERR are set accordingly), and bit ECCD.5 (least significant bit of bit-field VAL) is set to 1_B
  - However, a subsequent error on a different address doesn’t generate an error overflow, i.e. bit ECCD.EOV isn’t set to 1_B
- If an error (correctable, uncorrectable, address error) was detected at an address addr_x > 0, this error is correctly stored in the ETRR register (ETRR.ADDR = addr_x, flags SERR, CERR, UERR are set accordingly), and bit ECCD.5 (least significant bit of bit-field VAL) is set to 1_B
  - However, a subsequent read from same addr_x which is still erroneous will erroneously generate an error overflow, i.e. bit ECCD.EOV is set to 1_B

Workaround

Test address 0x0 by software to identify whether the device is sensitive to the first effect described above.

Periodically check bit-field VAL in register ECCD. If VAL ≠ 0, save the contents stored in ETRR (ADDR and MBI), and clear ECCD and ETRR afterwards (via ECCD.TRC = 1_B). If an error overflow is signaled, compare the current contents of ETRR with the saved value to identify an unmotivated error overflow (second effect described above).

2.127 [MTU_TC.011] MBIST Bitmap not working for w0 - r1

Description

The simple test case of writing all 0 and checking for 1 should return a full bitmap.

However, in this device step, only one (the last) address of the SRAM is returned.

Workaround

Use the reverse test w1 - r0, which is working as expected and returns the full bitmap.

2.128 [MTU_TC.012] Security of CPU cache memories during runtime is limited

Description

MTU chapter “Security Applications” in the User’s Manual describes that selected memories with potentially security relevant content are initialized under certain conditions to prevent reading of their data or supplying manipulated data.

The description is correct, but the initialization of CPU cache and cache tag memories triggered by MBIST enable/disable and when mapping/un-mapping these memories to/from system address space using MEMMAP register is of limited value:

- These memories stay functional as cache in the address mapped state. Therefore software can enable address mapping and afterwards watch cache usage of the application (this is a debug feature). Even manipulation of the cache content is feasible
- It is possible to abort an ongoing memory initialization

The security of memory initialization during startup is not affected. Also protection of FSI0 and HSM memories is not limited.

Workaround

Handle security relevant data exclusively inside HSM. Protect the application code by locking external access (for example lock debug interface, prevent boot via serial interface). Consider validation of application code by HSM secure boot.

2.129 [MultiCAN_TC.041] Clock used in Bit Timing Mode

Description

Unlike described in some parts of the documentation, in Bit Timing Mode f_{CLC} is used instead of f_{CAN}. This means that the time information stored in bit-field NFCRx.CFC is measured in f_{CLC} clock cycles.

Workaround

No workaround is available.

2.130 [MultiCAN_TC.043] CAN FD: Idle Condition

Description

***Note**: **TC23x**: This problem does not affect the nodes in TC23xED and ADAS devices. In TC23x step AC, this problem only affects node 2 of modules MultiCAN and MultiCAN1.*

The CAN FD ISO draft hardens the idle condition for the integration phase. Now it is also required that the 11 consecutive recessive bits occur without any synchronization in between. This is to ensure that integration of a node during a fast CAN FD data phase is not irritated by the fast baudrate.

Problem description

If a fast CAN FD data phase is overlapping with the integration phase, the CAN module might end the integration phase too early.

As a result, the node is not properly synchronized and might produce an error on the bus.

Workaround

None

2.131 [MultiCAN_TC.044] CAN FD: Missing Hardsync

Description

***Note**: **TC23x**: This problem does not affect the nodes in TC23xED and ADAS devices. In TC23x step AC, this problem only affects node 2 of modules MultiCAN and MultiCAN1.*

The CAN FD specification requires hard synchronization from the transition recessive FDF bit to dominant res bit being enabled. Hard-synchronization insures perfect synchronization, even for large timing offsets, due to arbitration loss.

Current implementation

Soft synchronization is implemented at FDF bit, no hard synchronization as required.

Problem description

Soft-synchronization is enabled for the transition recessive FDF bit to dominant res bit. In case of a time offset higher than the SJW (Synchronization Jump Width), only partial synchronization is achieved. This might not be sufficient for high speed CAN FD data phase.

Incorrect behavior

If a transmitter is losing arbitration late (for example in an extended frame - see Figure 4 the node which wins arbitration and Figure 5 the transmitting node loosing arbitration), then the FDF bit to res bit is the last chance to synchronize to the winning node (here as a receiver Figure 6 the same node as in Figure 5). A timing offset will occur between the now receiving node and the actual transmitter.

Consequence

The device is producing more error frames within a network than a device with a hardsync at the FDF bit.

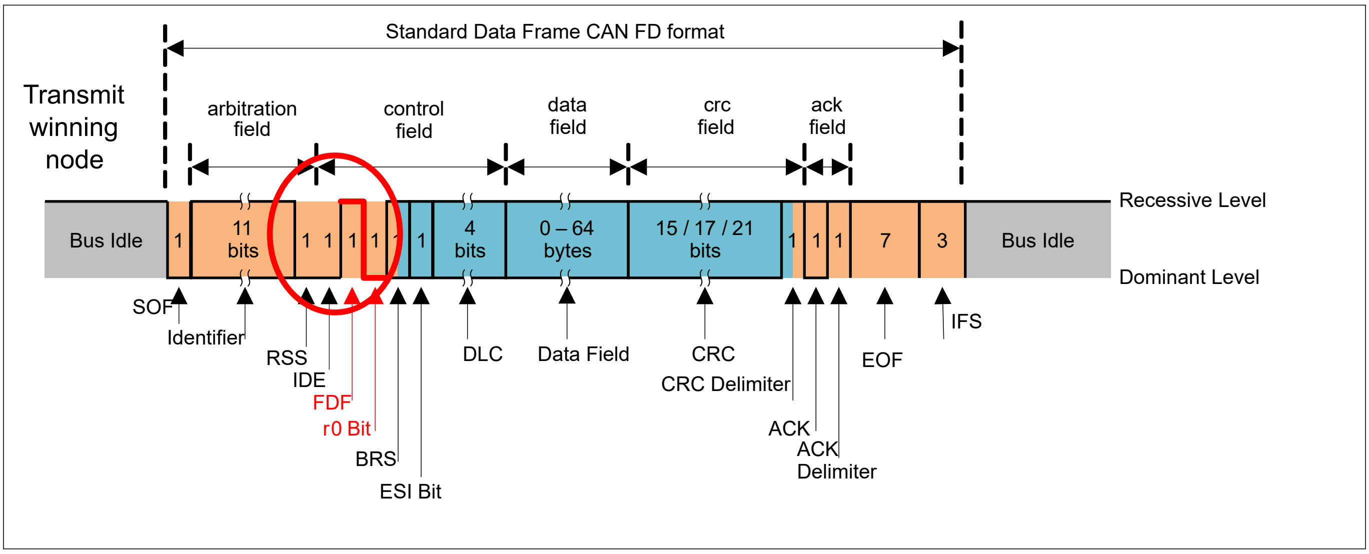


Figure 4 Transmitter - winning CAN FD node

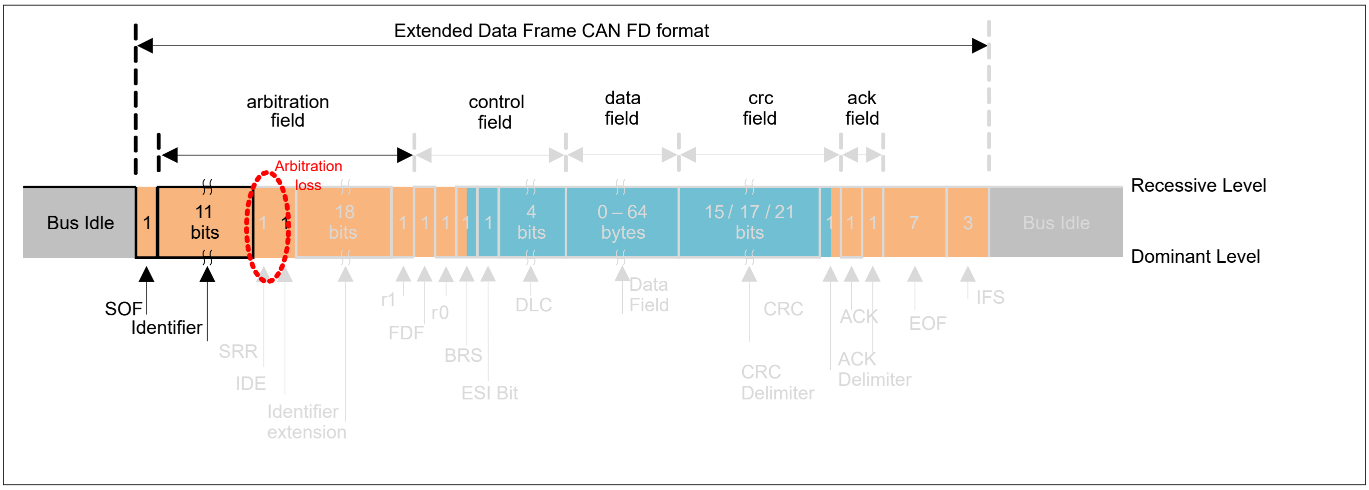


Figure 5 Transmitter - loosing CAN FD node

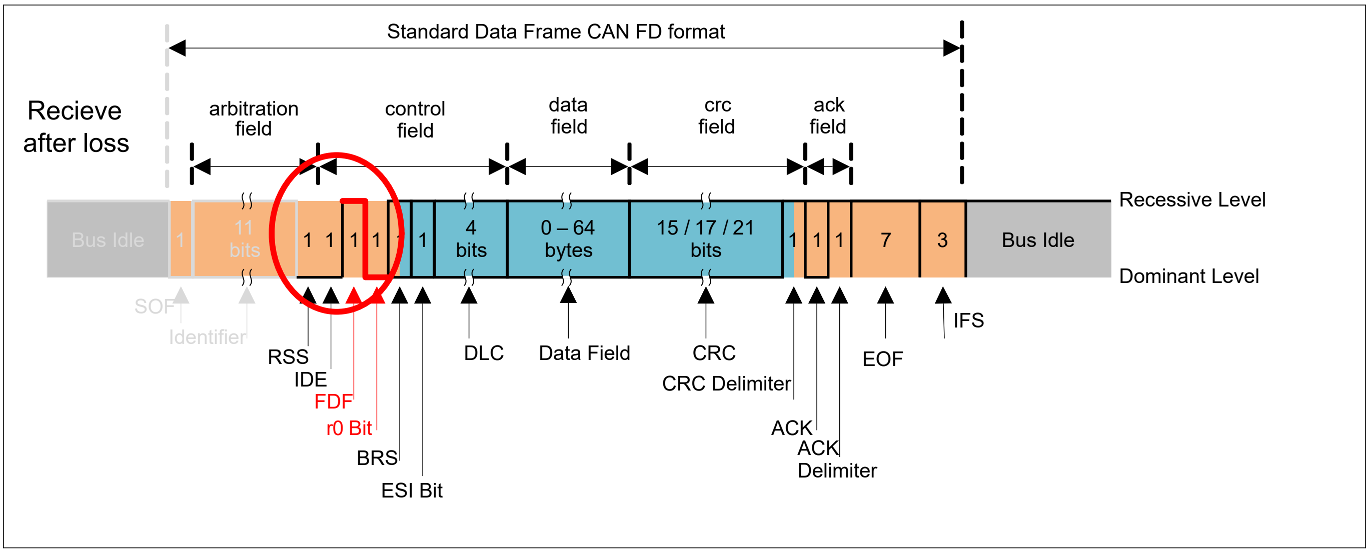


Figure 6 Now receiver (same node as previous figure) - loosing CAN FD node

Workaround

This workaround is not 100% solving the "Missing Hardsync" issue, but setting the SJW to the highest possible value will hide the problem in most cases.

2.132 [MultiCAN_TC.045] Simultaneous communication of ISO 11898-1:2015 CAN FD and Non-ISO CAN FD nodes

Description

When ISO CAN FD communication is enabled for CAN node 0, then ongoing CAN protocol activities of a Non-ISO CAN FD supporting node disturb the communication of CAN node 0 (see table at the end of MultiCAN_TC.H008 for a summary of nodes supporting ISO/Non-ISO CAN FD).

- For **TC23x**, this means:
  - When ISO CAN FD communication is enabled for CAN node 0, then ongoing CAN protocol activities of CAN **node 2** disturb communication of CAN node 0
- For **TC26x**, this means:
  - When ISO CAN FD communication is enabled for CAN node 0, then ongoing CAN protocol activities of CAN **node 4** disturb communication of CAN node 0

Thus, no overlapping communication of ISO CAN FD in node 0 and Classical CAN in a Non-ISO CAN FD supporting node is possible.

***Note**: Other ISO CAN FD supporting nodes (node 1 in TC23x, node 1..3 in TC26x) are not impacted by this issue.*

Workarounds

Option 1

Do not use the Non-ISO CAN FD supporting node of the device.

- For **TC23x**: disable CAN **node 2**
- For **TC26x**: disable CAN **node 4**

Option 2

Do not use CAN node 0 in the CAN FD mode. Use CAN node 0 only in Classical CAN mode.

2.133 [OCDS_TC.038] Disconnecting a debugger without device reset ("hot detach") may require reading of OCS registers

Description

If a debugger disconnects, it should activate at least the Debug Reset. This will reset all the main OCDS resources like CPUs, Cerberus, etc. However for peripherals having a BPI interface, there is the following issue: The Debug Reset is implemented as a synchronous clear on this level. If the OCDS registers are not clocked (for example for power saving reasons), the effect of this synchronous clear will be delayed to the next activation of the clock.

In general this will be more a theoretical problem. It's very unlikely that there is a use case, where a hot detach is required and critical OCDS resources of peripherals were used before. In nearly all cases this effect is invisible for a user, since any register access of the peripheral will generate the clock cycles which are required for the synchronous clear.

Workaround

In case of a hot detach, a tool should - after the Debug Reset activation - read the OCS registers of all peripherals where it used critical OCDS resources. These reads will initiate the required peripheral kernel clocks for the synchronous clear of the OCDS resources.

2.134 [OCDS_TC.042] OTGS capture registers can miss single clock cycle triggers

Description

The Cerberus OTGS capture registers (TCTL, TCCB, TCCH, TCIP, TCTGB, TCM) can fail to capture a trigger if the trigger is of single clock cycle duration and arrives in the same cycle as the same trigger register is being read by the bus.

Workaround

Avoid polling of OTGS capture registers while the system is running.

If polling while running can't be avoided use TLCCx counters for capturing critical Trigger Lines.

2.135 [OCDS_TC.043] Read-Modify-Write Bus Transactions to Cerberus Registers

Description

During read-modify-write (RMW) bus transactions to writable registers in the Cerberus (CBS), the target register is incorrectly updated with an undefined value during the Read-part. The correct value is always returned to the bus master for the Read-part, and the correct value is written to the register when the Write-part completes. But the register may contain an undefined value for a number of clock cycles between the Read-part and the Write-part.

The bus master (CPU) will see the RMW complete normally, but any logic driven by the hardware register's writable bits may be unexpectedly toggled.

This effects all registers that can be written by the SPB (using the FPI protocol) in the CBS block. It does not effect external access from the tool via JTAG/DAP.

Workaround

Do not use RMW bus operations targeting the CBS registers.

2.136 [PINNING_TC.001] Port functions and pinning tables for TC234* devices without ETH module in TQFP-144 package - Documentation update

Description

The port function and pinning tables included in chapter “PG-TQFP-144-27 Package Variant Pin Configuration of TC23x-ADAS” of the current version of the Data Sheet correctly only apply to the TC234* variants that include an ETH module. They are identified by the sales names

- TC234**LA** -* and TC234**LX**-*

and are only available for TC23x design step AB. In addition, these devices include 4 VADC kernels (G0..G3) and EMEM.

Other variants of TC23x design step AB do not have an ETH and EMEM module, and only have 2 VADC kernels (G0 and G1).

TC23x design step AC does not have an ETH and EMEM module, and only has 2 VADC kernels (G0 and G1).

Port functions and pinning tables for TC234* devices without ETH module in TQFP-144 package - Documentation update

For these devices, the following modifications must be considered in the respective rows and columns when using the port function and pinning tables in chapter “PG-TQFP-144-27 Package Variant Pin Configuration of TC23x-ADAS”:

Table 7 Modifications to port function and pinning tables for TC23x without ETH module in TQFP-144 package

Pin	Symbol	Buffer Type	Function
	ETH0_*		All ETH0_* inputs and outputs are not existing (device does not contain an ETH module)
	P10.1	A1 pad (instead of A1+)	
	P10.2	A1 pad (instead of A1+)	
	P10.3	A1 pad (instead of A1+)	
	P13.2	A1 pad (instead of A1+)	
	P13.3	A1 pad (instead of A1+)	
	P15.2	A1 pad (instead of A1+)	
	P15.3	A1 pad (instead of A1+)	
	P15.4	A1 pad (instead of A1+)	
	P40.1		Input VADCG2_0 is not existing (device does not contain VADC group G2)
	P41.1		Input VADCG3_0 is not existing (device does not contain VADC group G3)
10	VDD		Pin 10 is only used as VDD but not as VDDSB because EMEM is not available

2.137 [PLL_ERAY_TC.001] PLL_ERAY Initialization after Cold Power-up or Wake-up from Standby mode

Description

When the PLL_ERAY is configured by the application software after cold power-on reset or wake-up from Standby mode, it may not always reach the intended target frequency (either lock at a lower frequency, or go into unlock state), in particular at high temperature.

Workaround

The following code sequence, executed after power-on reset or wake-up from Standby mode and before initializing the PLL_ERAY, avoids the problem:

```
SCU_PLLERAYCON0.B.PLLPWD = 0;  // set PLL_ERAY to power saving mode
wait(10);                      // wait 10µs
SCU_PLLERAYCON0.B.PLLPWD = 1;  // set PLL_ERAY to normal behavior
...                            // initialize PLL_ERAY
```

2.138 [PLL_TC.005] PLL Initialization after Cold Power-up or Wake-up from Standby mode

Description

When the system PLL is configured by the application software after cold power-on reset or wake-up from Standby mode, it may not always reach the intended target frequency (either lock at a lower frequency, or go into unlock state), in particular at high temperature.

Workaround

The following code sequence, executed after power-on reset or wake-up from Standby mode and before initializing the system PLL, avoids the problem:

```
SCU_CCUCON0.B.CLKSEL = 0;       // switch system clock to another source different from PLL,
                                // e.g. back-up clock
SCU_CCUCON0.B.UP = 1;           // request update
while(SCU_CCUCON0.B.LCK == 1);  // wait for update handshake (see separate Note below)
SCU_PLLCON0.B.PLLPWD = 0;       // set PLL to power saving mode
wait(10);                       // wait 10µs
SCU_PLLCON0.B.PLLPWD = 1;       // set PLL to normal behavior
...                             // initialize PLL
```

***Note**: For devices with PLL_ERAY, see also problem PLL_ERAY_TC.001*

Note on update handshake

LCK = 0 indicates the end of the update handshake. Instead of polling LCK, other instructions may be executed that bridge this time.

The minimum number of instruction cycles n_{UH} (cycles of f_{SRI}) required to bridge the maximum time for the update handshake depends on the least common multiple of the active clock divider factors > 0 that are effective in CCUCON0/1/2/5 before the update by CCUCON0.B.UP = 1 in the sequence above is requested. For LPDIV = 0, this set includes FSIDIV, FSI2DIV, SPBDIV, SRIDIV, BAUD2DIV, BAUD1DIV, ASCLINSDIV, ASCLINFDIV, GTMDIV, STMDIV, CANDIV, MAXDIV, BBDIV.

This results in the following range when SRIDIV = 1 ($f_{SRI} = f_{SOURCE}$):

- $n_{UH} \geq 17$ if the active clock divider factors are any of the elements of the set {1, 2, 3, 4, 6, 12}
- $n_{UH} \geq 29$ if factor **8** is included in the set of {1, 2, 3, 4, 6, **8** , 12}
- $n_{UH} \geq 65$ if factor **5** (or multiples of 5) are included in the set of {1, 2, 3, 4, **5** , 6, **10** , 12, **15**}
- $n_{UH} \geq 125$ if factors **5** (or multiples of 5) and **8** are included in the set of {1, 2, 3, 4, **5** , 6, **8** , **10** , 12, **15**}

When SRIDIV = n > 1, only n_{UH} /n instruction cycles are required to bridge the maximum time for the update handshake, as the instructions take n times longer.

For LPDIV > 0, the divider factors for f_{SRI}, f_{SPB}, f_{BBB} and f_{MAX} are determined by LPDIV. As instruction execution is slowed down by the ratio defined by LPDIV, the number of instructions to bridge the time for the update handshake is scaled accordingly.

***Note**: For the allowed clock ratios see table "CCU allowed Clock Ratios" in the User's Manual.*

2.139 [PLL_TC.007] PLL Loss of lock when oscillator shaper is used

Description

Under certain conditions the PLL loses lock when the oscillator shaper is used (OSCCON.SHBY = 0_B, recommended system configuration, default after reset).

The fail behavior is not observed for oscillator frequencies $f_{OSC} \leq 25$ MHz when using an external crystal/ceramic resonator or supplying the clock signal directly.

Workaround

It is recommended to use input clock frequencies $f_{OSC} \leq 25$ MHz.

***Note**: For devices with PLL_ERAY, the problem also applies to PLL_ERAY.*

2.140 [PMC_TC.002] Switch Capacitor Regulator Mode, Frequency Spreading - Documentation Update to Register EVRSDCTRL1

Description

The documentation of bit-fields SDFREQSPRD, TON, TOFF in register EVRSDCTRL1 will be updated with the next revision of the User's Manual.

1. Documentation Update to SDFREQSPRD

The correct encoding of bit-field SDFREQSPRD (Frequency Spread Mode) in register EVRSDCTRL1 is as documented in Table 8:

Table 8 EVR13 SD Control Register 1 (EVRSDCTRL1) - Frequency Spread Mode Encoding

Field	Bits	Type	Description
SDFREQSPRD	[3:0]	rw	**Frequency Spread Mode** This bit-field defines the maximum number of back-up clock cycles (f_{BACK}) which are added to both charge (TON) and discharge (TOFF) switching phases thus increasing the average switching period. The number of clock cycles added are randomized equally over the listed cycle count range. The resulting TON and TOFF phase lengths are the same and 50% duty cycle is maintained. 0_H No frequency spreading activated. 1_H 0 to 3 clock cycles are added (default). 2_H 0 to 7 clock cycles are added. All other values are reserved.

2. Documentation Update to TON/TOFF

The effect of bit-field EVRSDCTRL1.SDFREQSPRD on the SC DCDC switching period is as documented in Table 9:

Table 9 EVR13 SD Control Register 1 (EVRSDCTRL1) - Switching Period

Field	Bits	Type	Description
TON	[15:8]	rw	**Charge Phase Length** The charge phase length is defined in back-up clock cycles (f_{BACK}, nominal 100 MHz): • In case SDFREQSPRD = 0: - Switching period (in cycles) = TON + TOFF + 18 • In case SDFREQSPRD ≠ 0: - Switching period (in cycles) = TON + TOFF + 18 + 2 * $[2^{(SDFREQSPRD + 1)} - 1]$
TOFF	[23:16]	rw	**Discharge Phase Length** The discharge phase length in clock cycles should be the same as the charge phase length.

2.141 [PMC_TC.003] Usecase limitation of LDO mode with on chip pass device for SAL devices

Description

***Note**: This problem only affects devices with temperature range classification "SAL" (ambient temperature range -40/150°C). For devices classified as "SAK" (-40/125°C), T_J is limited to 150°C anyway (see Data Sheet).*

The LDO mode of the internal EVR13 regulator with on-chip pass device shall not be used at condition $T_J > 150°C$.

Junction temperature T_J > 150°C leads to degradation of the pass device characteristics resulting in a deviation of the static accuracy (V_{OUTT}) of the EVR13 core regulator.

If the junction temperature is increased above 150°C in an application with the on-chip pass device, deviation of the static accuracy down to -3% of the nominal voltage is possible for I_{DD} load currents above 150 mA at the respective device junction temperature. The application shall be tolerable to such deviation of the output voltage accuracy considering limits for the dynamic regulation of the EVRC.

2.142 [QSPI_TC.006] Baud rate error detection in slave mode (error indication in current frame)

Description

According to the specification, a baud rate error is detected if the incoming shift clock supplied by the master has less than half or more than double the expected baud rate (determined by bit-field GLOBALCON.TQ).

However, in this design step, a baud rate error is detected not only if the incoming shift clock has less than half the expected baud rate (as specified), but also already when the incoming shift clock is somewhat (i.e. less than double) higher than the expected baud rate.

In this case, the baud rate error is indicated in the current frame.

Workaround

It is recommended not to rely on the baud rate error detection feature, and not to use the corresponding automatic reset enable feature (i.e. keep GLOBALCON.AREN = 0_B).

The baud rate error detection feature in slave mode is of conceptually limited use and is not related to data integrity. Data integrity can be ensured for example by parity, CRC, etc., while clocking problems of an AURIX™ master are detected by mechanisms implemented in the master.

Protection against the effects of high frequency glitches is provided by the spike detection feature in slave mode.

2.143 [QSPI_TC.017] Slave: Reset when receiving an unexpected number of bits

Description

A deactivation of the slave select input (SLSI) by a master is expected to automatically reset the bit counter of the QSPI module when configured as a slave.

This reset should help slaves to recover from messages where faults in the master or glitches on SCLK lead to an incorrect number of clocks on SCLK (= incorrect number of bits per SPI frame).

However, in this design step, the reset of the bit counter is unreliable.

Workaround

The slave should enable the Phase Transition interrupt (PT2EN = 1_B in register GLOBALCON1) to be triggered after the PT2 event “SLSI deselection” (PT2 = 101_B).

- **TC3xx**: In the interrupt service routine, after ensuring that the receive data has been copied, the software should issue a reset of the bit counter and the state machine via GLOBALCON.RESETS = 01_B
- **TC2xx**: In the interrupt service routine, after ensuring that the receive data has been copied, the software should issue a reset of the bit counter and the state machine via GLOBALCON.RESETS = 0111_B

2.144 [RESET_TC.005] Indication of Power Fail Events in SCU_RSTSTAT

Description

In case of consecutive cold resets triggered by EVR13, EVR33 or SWD power fail events, then only the last power fail event is registered in register SCU_RSTSTAT. It is not possible to distinguish individually between EVR13, EVR33 or SWD power fail events from RSTSTAT information.

Workaround

In case any power fail reset indication bit is set among EVR13, EVR33 or SWD power fail events in register SCU_RSTSTAT, it has to be assumed that all power fail events may have happened before.

2.145 [SCU_TC.034] $\overline{\text{TESTMODE}}$ pin shall be held at static high level during LBIST

Description

For a stable MISR signature, the level on pin $\overline{\text{TESTMODE}}$ (P20.2) must not change during LBIST execution.

Therefore, always keep pin $\overline{\text{TESTMODE}}$ (P20.2) at a static **high** level during LBIST execution.

2.146 [SMU_TC.006] OCDS Trigger Bus OTGB during Application Reset

Description

The SMU provides an alarm trigger and trace interface (Trigger Set TS16_SMU) using the OCDS Trigger Bus OTGB.

While the Application Reset is active, the SMU outputs the reset state of the OTGB interface instead of TS16_SMU.

This OTGB interface reset state is identical to TS16_SMU when no alarm is active.

After the Application Reset TS16_SMU is output again.

Workaround

Just ignore the phase in the OTGB trace where an alarm seems to become inactive while the Application Reset is active.

2.147 [SMU_TC.007] Size and Position of Field ACNT in Register SMU_AFCNT

Description

***Note**: This erratum might affect the SFR C Header Definitions. In such cases, SFR usage in the software shall be analyzed within the applications for their correct handling.*

In the SMU chapter of the User's Manual, in the description of register SMU_AFCNT (Alarm and Fault Counter),

- Size and position of field ACNT (Alarm Counter) are incorrectly described as SMU_AFCNT.[15:8], and
- Bits SMU_AFCNT.[7:4] are incorrectly shown as "Reserved; read as 0"

The **correct** size and position of field ACNT (Alarm Counter) in register SMU_AFCNT is SMU_AFCNT.[**15:4**], as shown in the following Table 10. The position of the "Reserved" bits is aligned accordingly.

Table 10 Field ACNT in Register SMU_AFCNT - Correction

Field	Bits	Type	Description
ACNT	[15:4]	rh	**Alarm Counter** This field is incremented by hardware when the SMU processes an **internal** action related to an alarm event (see Figure **"Alarm operation"**). The counter value holds if the maximum value is reached.
0	[29:16]	r	**Reserved** Read as 0; should be written with 0.

***Note**: The other fields (ACO, FCO, FCNT) of register SMU_AFCNT are correctly described in the User's Manual.*

2.148 [SMU_TC.008] Behavior of Action Counter ACNT

Description

Register SMU_AFCNT (Alarm and Fault Counter) implements a Fault Counter (FCNT) that counts the number of transitions from the RUN state to the FAULT state. Register AFCNT is only reset by a power-on-reset.

Whenever a pending alarm event is processed, the corresponding status bit is set to 1_B by hardware in the Alarm Status register AG<x>.

If an internal SMU action is configured for this alarm, the Action Counter (ACNT) in register AFCNT is incremented anytime the SMU processes this internal action.

Corner Case

In this device step, some of the alarm signals may increment the Action Counter ACNT multiple times for a single alarm event.

Workaround

Do not rely on the value in the action counter ACNT.

2.149 [SMU_TC.010] Transfer to SMU_AD register not triggered correctly

Description

The SMU contains Alarm Debug registers which can be used for diagnostic purposes. If an alarm which is configured to generate a reset (application or system reset) is sent to the SMU, a copy of the Alarm Status registers – AGi – into the Alarm Debug registers – ADi – is automatically triggered.

The AGi are reset by Application reset while the ADi are reset only by power-on reset.

Corner Case

In the case that a first SMU alarm AGi[j] generates a reset request, and a second alarm AGx[y] (where x=i and y=j is possible) configured for a reset occurs a few cycles before the reset is actually executed, then the reset values of the AGi registers will be transferred to the ADi register.

In this case, the ADi registers will not reflect the root cause that lead to a SMU alarm/reset.

***Note**: This corner case will always be met for level alarms.*

2.150 [SMU_TC.012] Unexpected alarms when registers FSP or RTC are written

Description TC2xx

Due to a synchronization issue, ALM3[27] is sporadically triggered if the PRE2 field of register FSP is written while the SMU is configured in Time Switching protocol (FSP.MODE = 10_B) and FSP[0] is toggling with a defined T_{SMU_FFS} period.

Also, ALM3[27] is sporadically triggered if the PRE1 or TFSP_HIGH fields of register FSP are written while the SMU is in the Fault State and T_{FSP_FS} has not yet been reached (STS.FSTS=0_B) (regardless of the FSP.MODE configuration).

In addition, an unexpected ALM2[29] or ALM2[30] is sporadically triggered if field FSP.PRE1 or RTC.RTD is written, and at least one recovery timer is running based on a defined T_{SMU_FS} period (regardless of the FSP.MODE configuration).

The alarms can only be cleared with cold or warm Power-On reset.

Description TC3xx

Due to a synchronization issue, ALM6[7] and ALM10[21] are sporadically triggered if the PRE2 field of register FSP is written while the SMU is configured either

- in Time Switching protocol (FSP.MODE = 10_B) and FSP[0] is toggling with a defined T_{SMU_FFS} period
- or in Dual Rail protocol (FSP.MODE = 01_B) and FSP[1:0] are toggling with a defined T_{SMU_FFS} period

Also, ALM6[7] and ALM10[21] are sporadically triggered if the PRE1 or TFSP_HIGH fields of register FSP are written while the SMU is in the Fault State and T_{FSP_FS} has not yet been reached (STS.FSTS=0_B) (regardless of the FSP.MODE configuration).

In addition, an unexpected ALM10[16] or ALM10[17] is sporadically triggered if field FSP.PRE1 or RTC.RTD is written, and at least one recovery timer is running based on a defined T_{SMU_FS} period (regardless of the FSP.MODE configuration).

The alarms can only be cleared with cold or warm Power-On reset.

Workaround TC2xx

To avoid unexpected alarms, perform the configuration of the PRE1, PRE2 or TFSP_HIGH fields only when the SMU is not in the Fault State and FSP is in Bi-stable protocol mode (FSP.MODE = 00_B). Mode switching and configuration shall not be done with the same write access to register FSP.

This means that in the Fault Free State:

- before writing to PRE1, PRE2 or TFSP_HIGH while Time Switching protocol is enabled:
  - disable Time Switching protocol by setting FSP in Bi-stable protocol mode (FSP.MODE = 00_B);
  - wait until Bi-stable protocol mode is active (read back register FSP twice);
  - write desired value to PRE1, PRE2 or TFSP_HIGH;
  - then switch FSP.MODE to the desired protocol (optional step)
- If the mode shall be changed after writing to PRE1, PRE2 or TFSP_HIGH while in Bi-Stable protocol mode (FSP.MODE = 00_B):
  - write desired value to PRE1, PRE2 or TFSP_HIGH;
  - then switch FSP.MODE to Time Switching protocol

If field FSP.PRE1 or RTC.RTD shall be written, make sure no recovery timer is running. It is not allowed to write to the PRE1 or RTD field when at least one recovery timer is running (indicated by bits RTS0 and RTS1 in the STS register).

Workaround TC3xx

To avoid unexpected alarms, perform the configuration of the PRE1, PRE2 or TFSP_HIGH fields only when the SMU is not in the Fault State and FSP is in Bi-stable protocol mode (FSP.MODE = 00_B). Mode switching and configuration shall not be done with the same write access to register FSP.

This means that in the Fault Free State:

- before writing to PRE1, PRE2 or TFSP_HIGH while Time Switching or Dual Rail protocol is enabled:
  - disable Time Switching or Dual Rail protocol by setting FSP in Bi-stable protocol mode (FSP.MODE = 00_B);
  - wait until Bi-stable protocol mode is active (read back register FSP twice);
  - write desired value to PRE1, PRE2 or TFSP_HIGH;
  - then switch FSP.MODE to the desired protocol (optional step)
- If the mode shall be changed after writing to PRE1, PRE2 or TFSP_HIGH while in Bi-Stable protocol mode (FSP.MODE = 00_B):
  - write desired value to PRE1, PRE2 or TFSP_HIGH;
  - then switch FSP.MODE to Time Switching or Dual Rail protocol

If field FSP.PRE1 or RTC.RTD shall be written, make sure no recovery timer is running. It is not allowed to write to the PRE1 or RTD field when at least one recovery timer is running (indicated by bits RTS0 and RTS1 in the STS register).

2.151 [SRI_TC.003] XBAR_PRIOL/H Register Layout and Reset Values

Description

***Note**: This erratum might affect the SFR C Header Definitions. In such cases, SFR usage in the software shall be analyzed within the applications for their correct handling.*

The CPU0 SRI masters (CPU0.DMI, CPU0.PMI) are mapped to the XBar_SRI Master Connection Interfaces MCI12 and MCI13 as described in table “Mapping of TC21x/TC22x/TC23x SRI master devices to MCI” of the User’s Manual.

***Note**: This implementation in the TC23x .. TC21x devices is compatible with the other devices (TC29x .. TC26x) of the AURIX™ family.*

However, the description of the register layout and reset values for the XBAR_PRIOL/H registers in chapter “TC21x/TC22x/TC23x Control Registers” of the TC21x/TC22x/TC23x Family User’s Manual V1.1 is partially incorrect.

The corrected parts of the description are shown in the following tables.

TC23x

Table 11 XBAR_PRIOL Registers - Reset Values TC23x

Short Name	Description	Reset Value
XBAR_PRIOLD	Arbiter Priority Register D	$0020\ 0002_H$
XBAR_PRIOL0	Arbiter Priority Register 0	$0020\ 0002_H$
XBAR_PRIOL4	Arbiter Priority Register 4	$0020\ 0002_H$
XBAR_PRIOLx (x = 6-7)	Arbiter Priority Register x	$0020\ 0002_H$

Table 12 XBAR_PRIOL Registers - Fields TC23x

Field	Bits	Type	Description
MASTER0	[2:0]	rw	**Master 0 Priority** (Priority of DMA Access)
MASTER5	[22:20]	rw	**Master 5 Priority** (Priority of SFI Access)
0	[31:23], [19:3]	r	**Reserved** Read as 0; should be written with 0.

TC22x, TC21x

Table 13 XBAR_PRIOL Registers - Reset Values TC22x, TC21x

Short Name	Description	Reset Value
XBAR_PRIOLD	Arbiter Priority Register D	$0000\ 0002_H$
XBAR_PRIOL0	Arbiter Priority Register 0	$0000\ 0002_H$
XBAR_PRIOL4	Arbiter Priority Register 4	$0000\ 0002_H$
XBAR_PRIOLx (x = 6-7)	Arbiter Priority Register x	$0000\ 0002_H$

Table 14 XBAR_PRIOL Registers - Fields TC22x, TC21x

Field	Bits	Type	Description
MASTER0	[2:0]	rw	**Master 0 Priority** (Priority of DMA Access)
0	[31:3]	r	**Reserved** Read as 0; should be written with 0.

TC23x .. TC21x

Table 15 XBAR_PRIOH Registers - Reset Values TC23x .. TC21x

Short Name	Description	Reset Value
XBAR_PRIOHD	Arbiter Priority Register D	$0055\ 0000_H$
XBAR_PRIOH0	Arbiter Priority Register 0	$0055\ 0000_H$
XBAR_PRIOH4	Arbiter Priority Register 4	$0055\ 0000_H$
XBAR_PRIOHx (x = 6-7)	Arbiter Priority Register x	$0055\ 0000_H$

Table 16 XBAR_PRIOH Registers - Fields TC23x .. TC21x

Field	Bits	Type	Description
MASTER12	[18:16]	rw	**Master 12 Priority** (Priority of CPU0.DMI Access)
MASTER13	[22:20]	rw	**Master 13 Priority** (Priority of CPU0.PMI Access)
0	[31:23], 19, [15:0]	r	**Reserved** Read as 0; should be written with 0.

3 Parametric deviations

3.1 [ADC_TC.P010] Increased Gain Error (EA_{GAIN}) for $T_J < 0°C$

Description

For devices with Analog-Digital-Converters (VADC) providing 16:1 analog multiplexers (TC26x, TC23x..TC21x), the maximum Gain Error (EA_{GAIN}) increases as follows for $T_J < 0°C$:

- from ±3.5 LSB_{12} to ±4.5 LSB_{12} when V_{DDM} = 4.5 V to 5.5 V (upper voltage range) and sample time t_S < 200 ns
- from ±5.5 LSB_{12} to ±6.5 LSB_{12} when V_{DDM} = 2.97 V to 4.5 V (lower voltage range) and sample time t_S < 400 ns

Note

1. The resulting Total Unadjusted Error (TUE) is not affected and remains as specified in the corresponding Data Sheet
2. For temperatures $T_J \geq 0°C$, the Gain Error (EA_{GAIN}) remains as specified in the corresponding Data Sheet
3. For $t_S \geq 200$ ns (upper voltage range) or $t_S \geq 400$ ns (lower voltage range), the Gain Error (EA_{GAIN}) remains as specified in the corresponding Data Sheet

3.2 [IDD_TC.H001] IPC Limits used in Production Test for IDD Max Power Pattern

Description

Instructions per cycle for a CPU is measured by dividing ICNT instruction counter value with the CCNT clock counter value.

***Note**: For a complete description of registers ICNT and CCNT refer to the TriCore™ Architecture Manual, chapter "Performance Counter Registers".*

Parameters using the max power pattern for device individual testing of power consumption limits (IDD) are tested for the following maximum IPC rates:

- **TC29x, TC27x**: for a maximum IPC rate of 1.2 for all CPUs available in the device
- **TC26x, TC23x, TC22x, TC21x**: for a maximum IPC rate of 1.3 for all CPUs available in the device

3.3 [IEVRSB_TC.P001] Test Condition for I_{EVRSB} (sum of all currents in standby mode) - Data Sheet correction

Description

In table "Power Supply" in the current version of the Data Sheet, in column "Note/Test Condition" for parameter "Sum of all currents (STANDBY mode)" (symbol I_{EVRSB}), the following term is **incorrect**:

- V_{EVRSB} = 5 V

Correction

The correct value for V_{EVRSB} in the test condition for I_{EVRSB} shall be

- V_{EVRSB} = 3.3 V

3.4 [PADS_TC.H004] PN-Junction Characteristics for Pad Type S

Description

As described in chapter “Package and Pinning Definitions” in the Data Sheet, symbol “S” in column “Type” is defined as class D ADC input with digital input.

Consequently, for pad type S, the PN-junction characteristics for pad type D apply.

The corresponding values for U_{IN} are listed in tables “PN-Junction Characteristics for positive Overload” and “PN-Junction Characteristics for negative Overload” in chapter “Pin Reliability in Overload” in the Data Sheet.

3.5 [RTH_TC.H001] Thermal characteristics of the package - Footnote update for LF-BGA-292-6 package

Description

- In the **TC27x** and **TC26x** Data Sheets, the JEDEC JESD51-1 standard for RQJA listed in the footnote for the LF-BGA-292-6 package in table “Thermal characteristics of the package” is not correct
- In the **TC23x** Data Sheet, the references to the JEDEC standards (JESD51-3/5/7) for RQJA in the footnote for the LF-BGA-292-6 package in table “Thermal characteristics of the package” are not correct. They only apply to the TQFP package

Correction

The correct footnote for the LF-BGA-292-6 package is: Value is defined in accordance with JESD51-**9**

3.6 [VDDPPA_TC.H001] Voltage to ensure defined pad states - Footnote update

Description

In the footnote for parameter “Voltage to ensure defined pad states” (symbol V_{DDPPA}) in table “Operating Conditions” of the Data Sheet, V_{DDP3} is mentioned as representative for “non-core supply voltages” in the text.

Update

The footnote for V_{DDPPA} should be extended to include all “non-core supply voltages” as follows:

*) This parameter is valid under the assumption the PORST signal is constantly at low level during the power-up/power-down of the “non-core supply voltages” (V_{DDP3}, V_{EXT}, V_{FLEX}, V_{DDFL3}, V_{DDM}, ..., depending on the respective TC2x device version).

4 Application hints

4.1 [ADC_AI.H003] Injected conversion may be performed with sample time of aborted conversion

Description

For specific timing conditions and configuration parameters, a higher prioritized conversion c_i (including a synchronized request from another ADC kernel) in cancel-inject-repeat mode may erroneously be performed with the sample time parameters of the lower prioritized cancelled conversion c_c. This can lead to wrong sample results (depending on the source impedance), and may also shift the starting point of following conversions.

The conditions for this behavior are as follows (all 3 conditions must be met):

1. **Sample Time setting**: injected conversion c_i and cancelled conversion c_c use different sample time settings, i.e. bit-fields STC* in the corresponding Input Class Registers for c_c and for c_i (GxICLASS0/1, GLOBICLASS0/1) are programmed to different values
2. **Timing condition**: conversion c_i starts during the first f_{ADCI} clock cycle of the sample phase of c_c
3. **Configuration parameters**: the ratio between the analog clock f_{ADCI} and the arbiter speed is as follows:

   $N_A > N_D *(N_{AR} +3)$,

   with

   - N_A = ratio f_{ADC}/f_{ADCI} (N_A = 1 .. 32, as defined in bit-field DIVA)
   - N_D = ratio f_{ADC}/f_{ADCD} = number of f_{ADC} clock cycles per arbitration slot (N_D = 1 .. 4, as defined in bit-field DIVD)
   - N_{AR} = number of arbitration slots per arbitration round (N_{AR} = 4, 8, 16, or 20, as defined in bit-field GxARBCFG.ARBRND)

Bit fields DIVA and DIVD mentioned above are located in register GLOBCFG.

As can be seen from the formula above, a problem typically only occurs when the arbiter is running at maximum speed, and a divider $N_A > 7$ is selected to obtain f_{ADCI}.

Recommendation 1

Select the same sample time for injected conversions c_i and potentially cancelled conversions c_c , i.e. program all bit-fields STC* in the corresponding Input Class Registers for c_c and for c_i (GxICLASS0/1, GLOBICLASS0/1) to the same value.

Recommendation 2

Select the parameters in register GLOBCFG and GxARBCFG according to the following relation:

$N_A \leq N_D *(N_{AR} +3)$.

4.2 [ADC_TC.H011] Bit DCMSB in register GLOBCFG

Description

The default setting for bit DCMSB (Double Clock for the MSB Conversion) in register GLOBCFG is 0_B, i.e. one clock cycle for the MSB conversion step is selected.

DCMSB = 1_B is reserved in future documentation and must not be used.

***Note**: In devices supporting Workaround 4 of problem ADC_TC.068, DCMSB = 1_B may be used to control synchronization of converter groups. TC27x step BC is not affected by problem ADC_TC.068.*

4.3 [ADC_TC.H014] VADC Start-up Calibration

Description

The formula for the duration of the start-up calibration in some versions of the TC2x User's Manuals is incorrect with respect to the used frequency, or missing.

In the following, the contents of chapter "Calibration" is reprinted, including the correct Formula for Start-up Calibration below.

Calibration

Calibration automatically compensates deviations caused by process, temperature, and voltage variations. This ensures precise results throughout the operation time.

An initial start-up calibration is required once after a reset for all converters. All converters must be enabled (ANONS = 11_B). The start-up calibration is initiated globally by setting bit SUCAL in register GLOBCFG. Conversions may be started after the initial calibration sequence. This is indicated by bit CALS = 1_B AND bit CAL = 0_B.

Formula for Start-up Calibration

The start-up calibration phase takes 4352 f_{ADCI} cycles (4352 × **50 ns** = 217.6 µs for f_{ADCI} = 20 MHz).

After that, postcalibration cycles will compensate the effects of drifting parameters. The postcalibration cycles can be disabled.

***Note**: The ADC error depends on the temperature. Therefore, the calibration must be repeated periodically.*

4.4 [ADC_TC.H015] Conversion Time with Broken Wire Detection

Description

As described in a note in section "Broken Wire Detection" of the User's Manual, the duration of the complete conversion is increased by the preparation phase (same as the sample phase) if the broken wire detection is enabled, i.e. the sample time doubles for standard conversions when broken wire detection is enabled (GxCHCTRy.BWDEN = 1_B):

Formula for Standard Conversions without Broken Wire Detection

- $t_{CN} = t_S + (N + PC) \text{ x } t_{ADCI} + 2 \text{ x } t_{VADC}$ (see also User's Manual/Data Sheet)

Formula for Standard Conversions with Broken Wire Detection

- $t_{CN} = 2 \, x \, t_S + (N + PC) \text{ x } t_{ADCI} + 2 \text{ x } t_{VADC}$

where:

t_S = (2 + STC) x t_{ADCI} for STC ≤ 15, and
t_S = (2 + (STC-15) x 16)) x t_{ADCI} for STC ≥ 16;
N = result width (8/10/12 bits);
PC = 2 if post-calibration selected, PC = 0 otherwise.

Examples

Conversion times for different configurations are shown in the following Table 17 (without broken wire detection) and Table 18 (with broken wire detection):

Table 17 Conversion Time for Standard Conversions - Without Broken Wire Detection - Examples

Result	Symbol	Time	Conditions
12-bit result	t_{C12}	(16 + STC) x t_{ADCI} + 2 x t_{VADC}	Post-calibration enabled, STC ≤ 15
10-bit result	t_{C10}	(12 + STC) x t_{ADCI} + 2 x t_{VADC}	Post-calibration disabled, STC ≤ 15
8-bit result	t_{C8}	(10 + STC) x t_{ADCI} + 2 x t_{VADC}	Post-calibration disabled, STC ≤ 15

Table 18 Conversion Time for Standard Conversions - With Broken Wire Detection - Examples

Result	Symbol	Time	Conditions
12-bit result	t_{C12B}	(18 + 2 x STC) x t_{ADCI} + 2 x t_{VADC}	Post-calibration enabled, STC ≤ 15
10-bit result	t_{C10B}	(14 + 2 x STC) x t_{ADCI} + 2 x t_{VADC}	Post-calibration disabled, STC ≤ 15
8-bit result	t_{C8B}	(12 + 2 x STC) x t_{ADCI} + 2 x t_{VADC}	Post-calibration disabled, STC ≤ 15

4.5 [ADC_TC.H020] Minimum/Maximum Detection Compares 12 Bits Only

Description

In minimum or maximum detection mode (FEN = 11_B or 10_B) new results are compared to the lower 12 bits of the respective result register bit-field RESULT.

Therefore, a value RESULT = $XFFF_H$ (X > 0_H) will not be updated for a new result value of $0FFF_H$ in minimum detection mode.

In a real application, this should be no problem, as the minimum detection usually sees values below $0FFF_H$.

Recommendation

For minimum detection, use the start value $0FFF_H$ (instead of $FFFF_H$ as mentioned in the User's Manual).

For maximum detection, use the start value 0000_H as mentioned in the User's Manual.

4.6 [ADC_TC.H022] Sample Time Control - Formula

Description

Table "Sample Time Coding" in section "Input Class Registers" of the VADC chapter in the User's Manual describes the additional clock cycles (selected in bit-fields STCS and STCE) to be added to the minimum sample time of two analog clock cycles.

As can be seen from the table in the User's Manual, the step width in the coding depends on the MSB of STCi (i = S or E). The following Table 19 has been copied from the User's Manual, with the corresponding formula added in the last column:

Table 19 Sample Time Coding

STCS / STCE	Additional Clock Cycles[1)]	Resulting Sample Time	Clock Cycle Formula
0 0000_B	0	2 / f_{ADCI}	2 + STCi
0 0001_B	1	3 / f_{ADCI}	
...	...	...	
0 1111_B	15	17 / f_{ADCI}	
1 0000_B	16	18 / f_{ADCI}	2 + (STCi - 15) x 16

(table continues...)

Table 19 (continued) Sample Time Coding

STCS / STCE	Additional Clock Cycles[1]	Resulting Sample Time	Clock Cycle Formula
1 0001_B	32	34 / f_{ADCI}	
…	…	…	
1 1110_B	240	242 / f_{ADCI}	
1 1111_B	256	258 / f_{ADCI}	

[1] The number of resulting additional clock cycles listed in this column corresponds to the term “STC” used in the conversion timing formulas in the Data Sheet.

4.7 [ADC_TC.H024] Documentation: Filter control only in registers GxRCR7/GxRCR15

Description

In sections “Finite Impulse Response Filter Mode (FIR)” and “Infinite Impulse Response Filter Mode (IIR)” of the VADC chapter in the User’s Manual,

- replace this sentence:
  “Several predefined sets of coefficients can be selected via bit-field DRCTR (coding listed in Table xx-6) in registers G0RCRy (y = 0 - 15)ff and GLOBRCR.”
- with this sentence:
  “Several predefined sets of coefficients can be selected via bit-field DRCTR (coding listed in Table xx-6) in registers **GxRCR7** and **GxRCR15**.”

4.8 [ADC_TC.H031] High precision bandgap voltage - documentation update

Description

The VADC is capable of providing measurement of the internal High Precision Bandgap Reference (HPBG) output voltage Vhpb shared by the PMS subsystem for additional safety supervision. The valid range of the Vhpb signal values currently is not specified.

Detailed description

The output voltage Vhpb of the HPBG is mapped on VADC channel G0CH12 via the double buffer. The expected nominal value of the signal seen by VADC equals to 2.4 V which is 2 x Vhpb.

The complete range of expected values during normal operation is 1.075 V .. 1.325 V, which corresponds to the range of ADC result values as 1761 ($6E1_H$) .. 2171 ($87B_H$) assuming VAREF = 5.0V.

The supervision signals are enabled by setting bit GLOBTF.RCEN. For operation at f_{ADCI} = 20 MHz, the recommended sample time setting for this measurement is STC = 0x13.

4.9 [ADC_TC.H038] Multiplexer Diagnostics Connection - Documentation update

Description

The multiplexer diagnostics feature can pull up the channel input line to V_{DDM} or pull it down to V_{SS}.

Figure “Signal Path Test” in the VADC chapter of the User’s Manual erroneously shows a connection to V_{DDP} instead of V_{DDM}. Pull-up to V_{DDP} is not possible.

Correction

In figure “Signal Path Test” in the VADC chapter of the User’s Manual, symbol “V_{DDP}“ shall be replaced by “V_{DDM}”.

4.10 [ADC_TC.H041] Offset address of register GxTRCTR - Correction to table “Registers Overview” in User’s Manual

Description

In table “Registers Overview” in the VADC chapter of the TC22x/TC23x Family User’s Manual V1.0 and V1.1, the offset address for register GxTRCTR (Trigger Control Register, Group x) is incorrectly documented as X550$_H$.

***Note**: The offset address of register GxTRCTR is correctly specified in the register description of this register at the end of chapter “27.5.1 Queued Request Source Handling”.*

Documentation correction

The offset address for register GxTRCTR in table “Registers Overview” in the VADC chapter of the TC22x/TC23x Family User’s Manual V1.0 and V1.1 shall be corrected as follows:

Table 20 Registers Overview - Correction of offset address for register GxTRCTR

Register Short Name	Register Long Name	Offset Address
GxTRCTR	Trigger Control Register, Group x	X554$_H$

4.11 [ADC_TC.H042] Precharging of capacitor CAINSW - Documentation update

Description

The following paragraph, which is the last paragraph in section “Input Signal Path” in the VADC chapter of the TC22x/TC23x Family User’s Manual V1.0 and V1.1:

“The capacitor C_{AINSW} is automatically precharged to a voltage of approximately half the standard reference voltage V_{AREF} to minimize the average difference between V_{AINx} and V_C at the beginning of a sample phase. Due to varying parameters and parasitic effects, the precharge voltage of C_{AINSW} is typically smaller than V_{AREF} / 2.”

shall be replaced as described in the following.

Documentation update

The capacitor C_{AINSW} is automatically precharged to a voltage of approximately half the standard reference voltage V_{AREF} while the converter is idle. Due to varying parameters and parasitic effects, the precharge voltage of C_{AINSW} is typically smaller than V_{AREF} / 2.

***Note**: When conversions are executed in a sequence, or when a conversion cancels a running conversion, the sample phase starts immediately. The converter does not become idle in this case and C_{AINSW} is not precharged!*

4.12 [ASCLIN_TC.H001] Bit field FRAMECON.IDLE in LIN slave tasks

Description

For LIN performing slave tasks, bit-field FRAMECON.IDLE has to be set to 000$_B$ (default after reset), i.e. no pause will be inserted between transmission of bytes.

If FRAMECON.IDLE > 000_B, the inter-byte spacing of the ASCLIN module is not working properly in all cases in LIN slave tasks (no bit errors are detected by the ASCLIN module within the inter-byte spacing).

4.13 [ASCLIN_TC.H003] Behavior of LIN Autobaud Detection Error Flag

Description

- Expected behavior:

  In ASCLIN, when auto baud detection (LINCON.ABD) is deactivated, the auto baud measurement should still be active and the Autobaud Detection Error Flag FLAGS.LA should be set when the value measured is outside the BRD.LOWERLIMIT and BRD.UPPERLIMIT range.
- Actual behavior:

  The Autobaud Detection Error Flag FLAGS.LA is not set, as the auto baud measurement is not active when auto baud detection is deactivated (LINCON.ABD = 0).

4.14 [ASCLIN_TC.H004] Changing the Transmit FIFO Inlet Width / Receive FIFO Outlet Width

Description

- **Expected behavior**:

  The Transmit FIFO should write the data to intended location of TxFIFO, even though the Transmit FIFO inlet width TXFIFOCON.INW is changed between the write operations.

  The Receive FIFO should read the data from intended location, even though the Receive FIFO outlet width RXFIFOCON.OUTW is changed between the read operations.
- **Actual behavior (Transmit FIFO)**:

  The Transmit FIFO does not write the data in the intended location when TXFIFOCON.INW is changed in an increasing order (from 1 to 2 to 4) between write operations.

  The Transmit FIFO writes the data only to aligned write index based on the number of bytes to be written (TXFIFOCON.INW).

  Example: Assuming that the write index of TxFIFO is from 0 to 15 (16 bytes), when TXFIFOCON.INW = 2, the TxFIFO writes two bytes of data starting only from half-word aligned write index (0, 2, 4, …, 14). Similarly when TxFIFO writes four bytes of data starting only from word aligned write index (0, 4, 8, 12).

  ***Note**: This misbehavior is seen only when TXFIFOCON.INW is changed in-between write operations.*
- **Actual behavior (Receive FIFO)**:

  The Receive FIFO does not read the data from intended location when RXFIFOCON.OUTW is changed in an increasing order (from 1 to 2 to 4) between read operations.

  The Receive FIFO reads the data only from aligned read index based on the number of bytes to be read (RXFIFOCON.OUTW).

  Example: Assuming that the read index of RxFIFO is from 0 to 15 (16 bytes), when RXFIFOCON.OUTW = 2, the RxFIFO reads two bytes of data starting only from half-word aligned write index (0, 2, 4, …, 14). Similarly when RxFIFO reads four bytes of data starting only from word aligned read index (0, 4, 8, 12).

  ***Note**: This misbehavior is seen only when RXFIFOCON.OUTW is changed in-between read operations.*

Effect

Previously written data in TxFIFO will be over-written by the new data, when the TxFIFO write index is not aligned with number of data bytes to be written.

Previously read data will be read again, when the RxFIFO read index is not aligned with number of data bytes to be read.

Recommendation

Flush the TxFIFO (TXFIFOCON.FLUSH) or RxFIFO (RXFIFOCON.FLUSH) before TXFIFOCON.INW or RXFIFOCON.OUTW is changed respectively.

4.15 [ASCLIN_TC.H005] Collision detection error reported twice in LIN slave mode

Description

An ASCLIN module configured as LIN slave node could report a wrong collision detection error during reception of LIN header after detecting a first correct collision detection error during the transmission of a response field of the previous LIN frame.

This misbehavior is observed under the following sequence:

- The LIN slave node detects a collision detection error when there is a bit error in its transmitted response frame, and then it goes to the idle state as expected
- The master transmits a header onto the LIN bus, and the LIN slave node receives header and tries to capture the identifier inside the header
- Then the LIN slave node reports another collision error which is wrongly detected during the reception of identifier although there is no corruption of LIN header on the bus

Recommendation

Ignore the collision detection error which happened during reception phase of a LIN slave node.

4.16 [ASCLIN_TC.H006] Sample point position when using three samples per bit

Description

As documented in the description of field BITCON.SAMPLEPOINT, "... if three sample points at position 7, 8, 9 are required, this bit-field would contain 9".

In general, if three samples per bit are selected (BITCON.SM = 1_B), field BITCON.SAMPLEPOINT defines the position of the last sample point.

Documentation update

The text related to three sample points in figure "ASCLIN Bit Structure" in the ASCLIN chapter of the user manual should be updated as follows:

- 16x Oversampling, 3 sample points, relevant sample position 7, 8, 9 (BITCON.OVERSAMPLING = 16, BITCON.SM = 1, BITCON.SAMPLEPOINT = 9)
  - instead of "16x Oversampling, 3 sample points, relevant sample position 8"
- 8x Oversampling, 3 sample points, relevant sample position 3, 4, 5 (BITCON.OVERSAMPLING = 8, BITCON.SM = 1, BITCON.SAMPLEPOINT = 5)
  - instead of "8x Oversampling, 3 sample points, relevant sample position 4"

4.17 [ASCLIN_TC.H008] SPI master timing – Additional information to Data Sheet characteristics

Description

The following note shall be added to chapter "ASCLIN SPI Master Timing" in the Data Sheet:

***Note**: The specified timings describe the pad capabilities for the respective driver strength configuration. For the maximum achievable baud rate in a given application, the MRST input timings need to be considered in particular.*

Background information

Chapter “ASCLIN SPI Master Timing” in the Data Sheet contains separate tables for different output driver configurations. As can be seen from these tables, the master output timings directly depend on the selected driver strength. The corresponding parameters are marked as controller characteristics with symbol “CC”.

The setup and hold timings for input data received from the slave are marked as system requirements with symbol “SR”. They must be provided by the system in which the device is designed in.

In a given application, the maximum rate at which data can be received from a slave on the master receive input MRST may be limited by the required setup time t_{52} (MRST setup to ASCLKO latching edge). As data is shifted by the slave on one edge of ASCLKO and latched by the master on the opposite edge, one phase of ASCLKO must always be greater than the minimum required MRST setup time (assuming the sampling point is in the middle). This means the ASCLKO period t_{50} must be > 2 x t_{52}.

4.18 [BCU_TC.H001] HSM Transaction Information not captured

Description

No HSM transaction information is captured by the System Bus Control Unit (SBCU). Therefore the following HSM related control/status register bits in the SBCU do not have any function:

- Register **SBCU_DBGRNT** (SBCU Debug Grant Mask Register):
  - **HSMCMI**: this control bit has no function. Behavior as described for SBCU_DBGRNT.ONE0
  - **HSMRMI**: this control bit has no function. Behavior as described for SBCU_DBGRNT.ONE0
- Register **SBCU_DGNTT** (SBCU Debug Trapped Master Register):
  - **HSMCMI**: this control bit has no function. Behavior as described for SBCU_DBGNTT.ONE0
  - **HSMRMI**: this control bit has no function. Behavior as described for SBCU_DBGNTT.ONE0

4.19 [BROM_TC.H003] Information related to Register FLASH0_PROCOND

Description

Chapters “TC2x BootROM Content” of the User’s Manuals contain a description of parts of the FLASH0_PROCOND register as used by the firmware. This description in subchapter “Configuration by Boot Mode Index (BMI)” shows an incorrect address F800 1030_H.

Correct is the description of this register in the PMU chapter with address F800 2030_H (FLASH0 base address F800 1000_H + offset 1030_H).

Additionally for TC26x

Furthermore, in the TC26x User’s Manual, the description for bit-field RAMIN = $0x_B$ is incorrect in chapter “TC26x Boot ROM Content”: RAMs are not initialized by system reset, RAMs are only initialized after power-on resets.

For a correct description, see the description of register FLASH0_PROCOND in the PMU chapter “Protection Configuration”.

4.20 [BROM_TC.H009] Re-enabling lockstep via BMHD

Description

For all CPUs with lockstep option, the lockstep functionality is controlled by Boot Mode Headers (BMHD) loaded during boot upon a reset trigger.

If lockstep is disabled for a CPUx with lockstep functionality, re-enabling (for example via a different BMHD) is not reliably possible if warm PORST, System or Application reset is executed.

Recommendation

Use cold PORST if lockstep is disabled and shall be re-enabled upon the reset trigger.

4.21 [BROM_TC.H010] Interpretation of value UNIQUE_CHIP_ID_32BIT

Description

As described in chapter “Debug System handling” in the AURIX™ TC2xx BootROM chapter, the value UNIQUE_CHIP_ID_32BIT is written to the COMDATA register by firmware.

***Note**: Unlike the name “UNIQUE_CHIP_ID_32BIT” may suggest, this value only identifies a particular product variant, but not an individual device.*

4.22 [BROM_TC.H019] CRC32 ethernet polynomial - Footnote correction

Description

As documented in the FCE chapter of the User’s Manual, CRC calculation is based on IEEE 802.3, the CRC32 ethernet polynomial used is 0x04C11DB7.

In footnote [2)] below table “Boot Mode Header (BMHD) structure” in the BootROM chapter, the CRC32 ethernet polynomial is erroneously documented as $04C11DB71_H$.

Documentation correction

Footnote [2)] below table “Boot Mode Header (BMHD) structure” in the BootROM chapter shall be corrected (trailing “1” deleted) as follows:

- [2)] CRC calculation is based on IEEE 802.3, the CRC32 ethernet polynomial used is $04C11DB7_H$

4.23 [BUS_TC.H001] CPU access latency for TC21x/TC22x/TC23x - Documentation update

Description

TC21x/TC22x/TC23x devices have one TC1.6E CPU core with one PSPR and DSPR.

Documentation update

The rows in table “CPU access latency in CPU clock cycles for TC21x/TC22x/TC23x” in chapter “On-Chip System Buses and Bus Bridges” in the TC21x/TC22x/TC23x Family User’s Manual referring to

- “.. other PSPR” and
- “.. other DSPR”

do not apply to these devices and shall be ignored.

4.24 [BUS_TC.H002] Reset value for register XBAR_IDINTEN - Documentation update

Description

The reset value of the Transaction ID Interrupt Enable register XBAR_IDINTEN is determined by the configured or enabled masters and slaves.

As no master device is connected to SRI Master Connection Interface MCI4 in TC23x/TC22x/TC21x, the corresponding bit XBAR_IDINTEN.20 is specified as “reserved” (read as 0, should be written with 0) in the register description in the TC23x/TC22x/TC21x User’s Manual. This contradicts the documented XBAR_IDINTEN reset value of 3031 80D1$_H$, where bit 20 is shown as 1$_B$.

Documentation update

The reset value for register XBAR_IDINTEN in the TC23x/TC22x/TC21x User’s Manual shall be changed as follows:

- XBAR_IDINTEN reset value = 3021 80D1$_H$

As the TC23x/TC22x/TC21x User’s Manual is a family user’s manual for all device variants, for specific device variants that do not include the full feature set less bits may be set to 1$_B$ after reset in register XBAR_IDINTEN.

See also the notes below the description of register XBAR_IDINTEN in the TC23x/TC22x/TC21x User’s Manual.

4.25 [CCU6_AI.H001] Update of Register MCMOUT

Description

At every correct Hall event (CM_CHE), the next Hall patterns are transferred from the shadow register MCMOUTS into MCMOUT (Hall pattern shadow transfer HP_ST), and a new Hall pattern with its corresponding output pattern can be loaded (for example from a predefined table in memory) by software into MCMOUTS. For the Modulation patterns, signal MCM_ST is used to trigger the transfer.

Loading this register can also be done by writing MCMOUTS.STRHP = 1$_B$ (for EXPH and CURH) or MCMOUTS.STRMCMP = 1$_B$ (for MCMP).

***Note**: If in a corner case a hardware event occurs simultaneously with a software write where MCMOUTS.STRHP = 1$_B$ or MCMOUTS.STRMCMP = 1$_B$, the current contents of MCMOUTS is copied to the corresponding bit-fields of MCMOUT. The new value written to MCMOUTS will be loaded upon the next event.*

4.26 [CCU6_AI.H002] Description of Bit RWHE in Register ISR

Description

Register ISR (Interrupt Status Reset Register) contains bits to individually clear the interrupt event flags by software. Writing a 1$_B$ clears the bit(s) in register IS at the corresponding bit position(s), writing a 0$_B$ has no effect.

In some versions of the User’s Manual, the description of bit RWHE (Reset Wrong Hall Event Flag) in column “Description” of register ISR is wrong (description for status 0$_B$ and 1$_B$ inverted).

The correct description for bit RWHE is (like for all other implemented bits in register ISR) as shown in the following Table 21:

Table 21 Bit RWHE in register ISR

Field	Bits	Type	Description
RWHE	13	w	**Reset Wrong Hall Event Flag** 0$_B$ No action 1$_B$ Bit WHE will be cleared

4.27 [CCU6_AI.H003] Bit TRPCTR.TRPM2 in Manual Mode - Documentation Update

Description

In CCU6 chapter “Trap Control Register” of the User’s Manual, the description for bit TRPCTR.TRPM2 = 1_B (Manual Mode) incorrectly states:

“Manual Mode:

Bit TRPF stays **0** after the trap input condition is no longer valid. It has to be cleared by SW by writing ISR.RTRPF = 1.”

Correction

The correct description is as follows:

Manual Mode:

Bit TRPF stays **1** after the trap input condition is no longer valid. It has to be cleared by SW by writing ISR.RTRPF = 1.

4.28 [CCU_TC.H001] Clock Monitor Check Limit Values

Description

The values for the check limits of the clock monitor have been updated as shown in the tables below.

Documentation update for TC29x step BB, TC27x ≤ step DB, TC26x step BB, TC23x .. TC21x step AB

For TC29x step BB, TC27x ≤ step DB, TC26x step BB, TC23x .. TC21x step AB, Table 22 replaces the corresponding table in chapter “Clock Monitors” of the User’s Manual.

Table 22 Target trimmed Check limits for TC29x step BB, TC27x ≤ step DB, TC26x step BB, TC23x .. TC21x step AB

Target Frequency	LOWER value	UPPER value	SELXXX[1]	Error can be detected for min. deviation	Error is detected for min. deviation
7.5 MHz	0x24	0x27	11_B	-1.26% +1.54%	-6.45% +6.35%
$6.\overline{6}$ MHz	0x20	0x23	10_B	-0.91% +2.75%	-6.09% +7.50%
6 MHz	0x1C	0x1F	01_B	-3.35% +1.54%	-8.43% +6.35%
5 MHz	0x17	0x1A	00_B	-2.76% +4.07%	-9.41% +7.50%

[1] refers to corresponding bit-field xxxSEL in respective CCUCON register

Documentation update for TC29x step BC, TC27x step DC, TC26x step BC, TC23x .. TC21x step AC

For TC29x step BC, TC27x step DC, TC26x step BC, TC23x .. TC21x step AC, Table 23 replaces the corresponding table in chapter “Clock Monitors” of the User’s Manual.

Table 23 Target trimmed Check limits for TC29x step BC, TC27x step DC, TC26x step BC, TC23x .. TC21x step AC

Target Frequency	LOWER value	UPPER value	SELXXX[1]	Error can be detected for min. deviation	Error is detected for min. deviation
7.5 MHz	0x23	0x27	11_B	-4.07% +1.54%	-9.40% +6.35%
$6.\overline{6}$ MHz	0x1F	0x23	10_B	-4.07% +2.75%	-9.40% +7.50%
6 MHz	0x1C	0x1F	01_B	-3.35% +1.54%	-8.43% +6.35%
5 MHz	0x17	0x1A	00_B	-2.76% +4.07%	-9.41% +7.50%

[1] refers to corresponding bit-field xxxSEL in respective CCUCON register

4.29 [CCU_TC.H002] Oscillator Gain Selection via OSCCON.GAINSEL

Description

The reset value of OSCCON.GAINSEL = 11_B provides the default and recommended setting for the oscillator gain. It is not required to modify this value, as the adaptation to a crystal frequency is done via the external circuitry.

Therefore, all other gain selections should be regarded as reserved for special application topics, as shown in the following Table 24.

Table 24 Oscillator Gain Selection via OSCCON.GAINSEL

Field	Bits	Type	Description
GAINSEL	[4:3]	rw	**Oscillator Gain Selection** This value should not be changed from the reset value 11_B. 00_B Low gain 1: reserved for adaptations 01_B Low gain 2: reserved for adaptations 10_B Low gain 3: reserved for adaptations 11_B **Maximum gain: default setting**

Recommendation

Always to keep the default configuration of OSCCON.GAINSEL = 11_B.

4.30 [CCU_TC.H005] References to f_{PLL2}, f_{PLL2_ERAY} and K3 Divider in User’s Manual

Description

The VADC incorporated in this device uses clocks derived from f_{SPB}.

Previous design steps (for example TC27x Bx, TC26x Ax, TC29x Ax) incorporated a different VADC module also clocked by f_{ADC}, which could be derived via the K3 divider from f_{PLL2}, f_{PLL2_ERAY}. These clocks were selected in CCUCON0.[27:26], which is described as “Reserved/Should be written with 0” in the present version of the User’s Manual.

Clocks f_{PLL2}, f_{PLL2_ERAY} and the K3 divider are still described in the present version of the User's Manual.

Recommendation

- New software implementations should not consider f_{PLL2}, f_{PLL2_ERAY} and the K3 divider
- Software ported from previous design steps with a VADC module clocked by f_{ADC} may be reused on this device step (for TC29x, TC27x, and TC26x see also SMU_TC.H004)

4.31 [CCU_TC.H006] Clock Monitor Support - Documentation Update

Description

The note at the end of section "Operating the Clock Monitors" in chapter "Clock Monitors":

***Note**: This feature is supported by the Infineon safety driver [safTlib] and there is no additional customer software required.*

should state more precisely:

***Note**: The Infineon SafeTlib provides a test for the clock monitor. The clock monitor shall be configured by the application software.*

4.32 [CCU_TC.H007] Oscillator Watchdog Trigger Conditions for ALM3[0]

Description

As described in the User's Manual in section "Oscillator Watchdog", the divider value OSCCON.OSCVAL has to be selected in a way that f_{OSCREF} is within the range of 2 MHz to 3 MHz, and should be as close as possible to 2.5 MHz.

The Oscillator Watchdog (OSC_WDT) will trigger the "input clock out of range" alarm ALM3[0] under the following conditions:

- Boundary for **too high** frequencies:
  - for (OSCVAL+1) x 6.25 ≤ f_{OSC} [MHz] ≤ (OSCVAL+1) x 7.5, an alarm can be generated, but there is no guarantee that it is generated
  - for f_{OSC} [MHz] > (OSCVAL+1) x 7.5, an alarm is always generated
- Boundary for **too low** frequencies:
  - for (OSCVAL+1) x 1.25 ≤ f_{OSC} [MHz] ≤ (OSCVAL+1) x 1.67, an alarm can be generated, but there is no guarantee that it is generated
  - for f_{OSC} [MHz] < (OSCVAL+1) x 1.25, an alarm is always generated

The accuracy of these limits [in %] depends on the variation [in %] of the back up clock (see specification of f_{BACKUT} and f_{BACKT} in the Data Sheet).

Example

- For f_{OSC} = 20 MHz, selecting OSCVAL = 7 results in f_{OSC} = 2.5 MHz.
  - An alarm for too high frequencies can be generated for f_{OSC} ≥ 50 MHz
  - An alarm for too high frequencies is always generated for f_{OSC} > 60 MHz
  - An alarm for too low frequencies can be generated for f_{OSC} ≤ 13.36 MHz
  - An alarm for too low frequencies is always generated for f_{OSC} < 10 MHz

4.33 [CCU_TC.H010] Oscillator Mode control in register OSCCON - Documentation Update

Description

The description for setting OSCCON.MODE = 00_B in register OSCCON must be changed from

- “External Crystal / Ceramic Resonator Mode and External Input Clock Mode. The oscillator Power-Saving Mode is not entered.”

to:

- “External Crystal / Ceramic Resonator Mode. The oscillator Power-Saving Mode is not entered.”

Recommendation

When using an external input clock signal connected to XTAL1 (XTAL2 open), do not use setting OSCCON.MODE = 00_B. Instead, use setting OSCCON.MODE = 10_B.

4.34 [CPU_TC.H006] Store Buffering in TC1.6/P/E Processors

Description

Store buffering is a method of increasing processor performance by decoupling memory write operations from the instruction execution flow within the CPU. All write data is placed in a FIFO buffer (known as the store buffer) by the CPU prior to being read by the memory/bus interfaces and written to memory. This allows the processor to continue execution without waiting for the write data to be written to the target memory location. Data is written to the store buffer at processor speed and read from the store buffer at memory/bus speed. Typically the read bandwidth from the store buffer will exceed the write bandwidth from the processor, only if the store buffer fills will the processor stall.

To further increase performance memory read operations are prioritized ahead of memory write operations from the store buffer. This ensures that the processor does not stall on data loads while data writes are pending in the store buffer. A side effect of this prioritizing is that memory may not be accessed in program order.

Operational Details

The function of the store buffer is designed to be invisible to the end user under normal operation:

- All CPU load operations are checked against the store buffer contents. Data for matching load addresses is either immediately forwarded to the CPU from the store buffer (TC1.6, TC1.6P) or written to memory prior to the load operation proceeding (TC1.6E)
- All loads and store operations to peripheral regions (typically segments E_H and F_H) are performed in strict program order (no load prioritization)

The operation of the store buffer can become visible when in-order memory access is required to non-peripheral segments.

This can occur under the following circumstances:

- When programming flash memory
- When performing memory testing with the processor
- When data is required to be in memory for inter-core/inter-module communication

In such cases the following solutions may be employed:

- The store buffer may be explicitly flushed by use of a DSYNC instruction
- In-order data transactions may be enabled by setting SMACON.IODT = 1_B. This should not be done during normal operation, but should only be performed by test routines at start-up or shut-down, as it significantly impacts performance

***Note**:* *In this context, the following statements included in chapter "Store buffers" and in the description of register SMACON for IODT =1_B in the user manual may be misleading:*

- *"Store buffer operation may be disabled by setting the SMACON.IODT bit"*
- *SMACON.IODT = 1_B: "... processor store buffer disabled"*

Effectively, setting SMACON.IODT = 1_B results in memory operations to be performed in program order, where loads always flush preceding stores.

Examples

The following examples refer to memory accesses to non-peripheral regions (i.e. segments 0_H .. D_H):

Example-1a Out of order memory access due to load prioritization

```
Program Flow    -    Memory Access
 st-1                 ld-4
 st-2                 ld-5
 st-3                 ld-6
 ld-4                 st-1
 ld-5                 st-2
 ld-6                 st-3
```

Example-1b In order memory access enforced by DSYNC

```
Program Flow    -    Memory Access
 st-1                 st-1
 st-2                 st-2
 st-3                 st-3
 dsync
 ld-4                 ld-4
 ld-5                 ld-5
 ld-6                 ld-6
```

Example-2a Load forwarding from store buffer - no memory read (TC1.6/1.6P)

```
Program Flow    -    Memory Access
 st.w [a0], d0
 ld.w d1,  [a0]       st.w [a0], d0
```

Example-2b In order memory access enforced by DSYNC (TC1.6/1.6P)

```
Program Flow    -    Memory Access
 st.w [a0], d0        st.w [a0], d0
 dsync
 ld.w d1,  [a0]       ld.w d1,  [a0]
```

4.35 [CPU_TC.H008] Instruction Memory Range Limitations

Description

To ensure the processor cores are provided with a constant stream of instructions the Instruction Fetch Units will speculatively fetch instructions from up to 64 bytes ahead of the current Program Counter (PC).

If the current PC is within 64 bytes of the top of an instruction memory the Instruction Fetch Unit may attempt to speculatively fetch instructions from beyond the physical range. This may then lead to error conditions and alarms being triggered by the bus and memory systems.

Recommendation

It is therefore recommended that either the MPU is used to define the allowable executable range or that the upper 64 bytes of any memory be initialized but unused for instruction storage for the TC1.6.* class processors. For TC1.3.* class processors this may be reduced to 32 bytes.

4.36 [CPU_TC.H009] Details on CPU Clock Control

Description

As described in chapter "Clock Control Unit" of the User's Manual, the effective CPU execution frequency may be reduced by programming the associated bit-field CPUxDIV in register CCUCONn (where x is the core number, and n = x+6).

The effective execution frequency f_{CPUx} seen by CPUx is given by the following equation (where f_{SRI} is the base SRI frequency):

- $f_{CPUx} = f_{SRI}$ * (64 - CPUxDIV) / 64

A CPUxDIV value of 0 results in the core CPUx being clocked at the SRI frequency (no frequency reduction).

To avoid synchronisation issues typically associated with clock division the clock control mechanism stalls the issue of instructions into the processor pipeline rather than by modifying the actual applied clock. An incoming instruction fetch packet is stalled for the number of cycles required to approximate the required execution frequency. The stall is seen by the processor as a stall in the instruction stream in the same way a stalling instruction memory would be seen.

In most scenarios this mechanism provides a good approximation to clock division based control. The actual reduction in effective frequency will be dependent on the code executed.

When determining IPC rates as described in AP32168 (Application Performance Optimization for TriCore V1.6 Architecture), note that for CPUxDIV > 0, field Count Value in register CCNT still represents SRI clock cycles.

4.37 [CPU_TC.H012] Behavior of bit-wise operations on certain peripheral register bits which need to be written back with the same value

Description

The LDMST, ST.T, CMPSWAP.W, SWAPMSK.W and SWAP.W instructions in the AURIX™ microcontrollers are instructions intended to provide atomicity as well as bit-wise operations to a targeted memory location or peripheral register. They are also referred to as Read-Modify-Write (RMW) instructions.

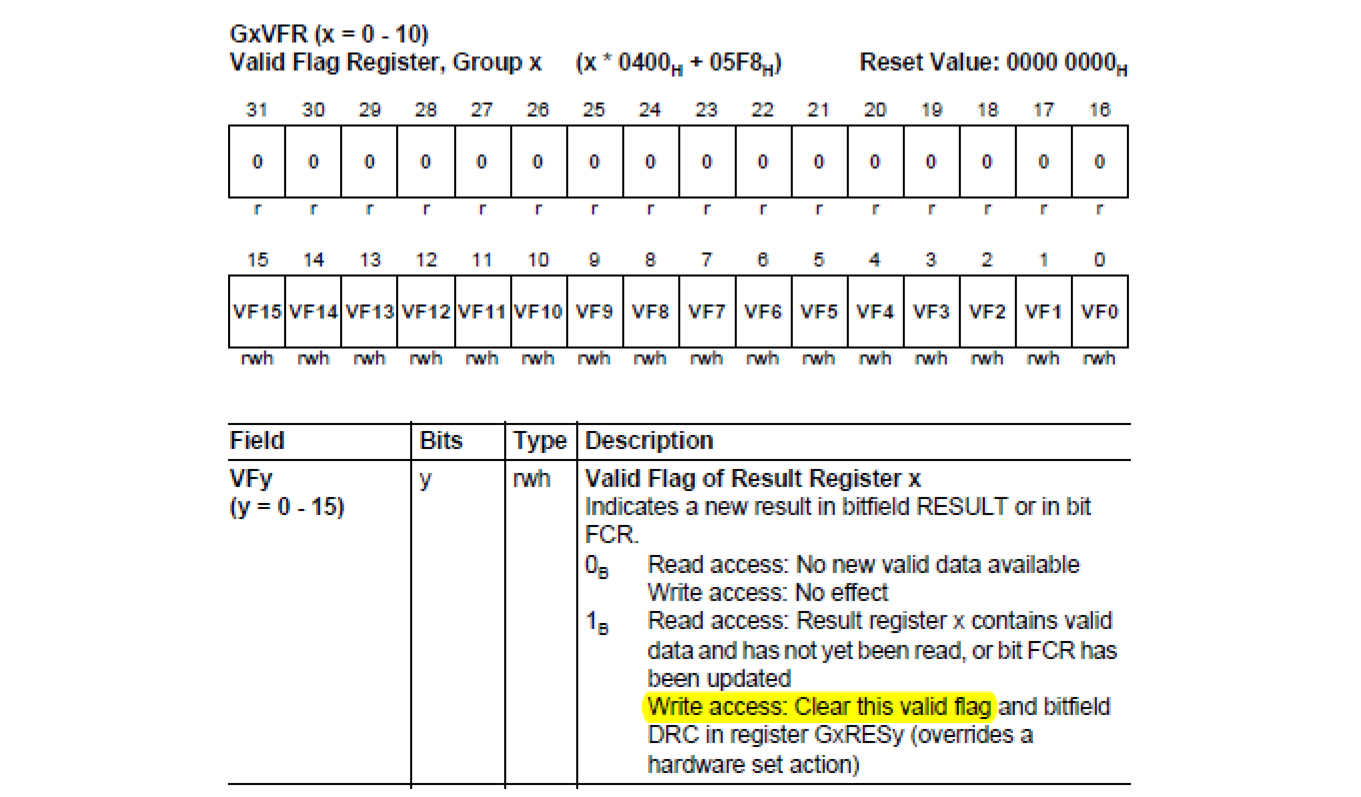
In some registers in certain modules, a bit has to be written with the same value (for example a bit set to 1_B has to be written with a 1_B to perform an operation).

When using a RMW instruction to write to such a bit, the write is masked away and will not happen at all.

***Note**: Writing a different value (for example writing a 1_B to a bit currently at 0_B) is not affected, and works as expected to modify only the selected bit.*

Example

Consider the GxVFR register in the VADC module:



GxVFR (x = 0 - 10)
Valid Flag Register, Group x (x * 0400_H + $05F8_H$) Reset Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VF15	VF14	VF13	VF12	VF11	VF10	VF9	VF8	VF7	VF6	VF5	VF4	VF3	VF2	VF1	VF0
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
VFy (y = 0 - 15)	y	rwh	**Valid Flag of Result Register x** Indicates a new result in bitfield RESULT or in bit FCR. 0_B Read access: No new valid data available Write access: No effect 1_B Read access: Result register x contains valid data and has not yet been read, or bit FCR has been updated Write access: Clear this valid flag and bitfield DRC in register GxRESy (overrides a hardware set action)

Figure 7 Register GxVFR in the VADC Module of TC2xx Devices

The bits in the GxVFR register have to be written with 1_B to clear a valid flag VFy indicating a valid result. Assuming VFy = 1_B, if one of the RMW instructions listed above is used, the write to VFy would never happen since VFy is already set to 1_B. This means that the next read of VFy may lead to incorrect conclusions by software.

Affected Modules and Registers in the AURIX™ Platform

- CCU6: IMON
- VADC: GxVFR, GxSEFLAG, GxCEFLAG, GxREFLAG, GLOBEFLAG

***Note**: VADC is located outside the addressable range of ST.T, so ST.T need not be considered in the context of VADC.*

Recommendation

In the affected modules, use only direct writes (i.e, write the whole register as a 32-bit word), and do not use RMW operations to write to such bits.

For example, to clear bit VF0 in the GxVFR register, the software should write:

```
VADC_GxVFR.U = 0x00000001;
```

Here .U implies writing the whole 32-bit register as an unsigned integer.

4.38 [CPU_TC.H014] ACCEN* Protection for Write Access to Safety Protection Registers - Documentation Update

Description

The access protection symbol ‘P’ to indicate protection by the ACCEN* register mechanism is missing in column “Access Mode - Write” in table “Safety Protection Registers” in the CPU chapter of the User’s Manual for RGN*x registers with an index x ≥4, and for register ACCENA.

Actually, these registers also have write access attribute ‘P’.

4.39 [CPU_TC.H015] Register Access Modes for Safety Protection Registers - Documentation Update

Description

The access protection symbol ‘U’ is erroneously included and should be removed in column “Access Mode - Write” for all registers in table “Safety Protection Registers” in the CPU chapter of the User’s Manual.

The note below this table is rephrased as follows:

***Note**: A disallowed access to any CPU register (for example attempted write to non-existent register, attempted write to read only register, attempted access to E without Endinit, etc.) will NOT result in a Bus Error*

4.40 [CPU_TC.H017] MSUB.Q does not match MUL.Q+SUB - Documentation Update

Description

The AURIX™ implementation of MSUB.Q uses infinitely precise intermediate results. In contrast with AUDO™ devices this can lead to different observable results for MSUB.Q when compared with a MUL.Q+SUB sequence.

The following table describes these differences in the MSUB.Q behaviour in AURIX™ 1st and 2nd generation products.

***Note**:*
- *For 1st generation AURIX™ devices (TC2xx), this is a documentation update to the TriCore™ TC1.6P & TC1.6E Core Architecture Manual V1.0D15 (Vol.2 Instruction Set)*
- *The TriCore™ TC1.6.2 Core Architecture Manual (Vol.2 Instruction Set) V1.1 and following for 2nd Generation AURIX™ (TC3xx) contains these new definitions*

Table 25 **MSUB.Q Definitions in AURIX™ different from AUDO™**

Secondary Opcode [23:18]	Instruction Mnemonic	Updated Description
0x00	**MSUB.Q D[c], D[d], D[a], D[b] U, n** 32 - (32 * 16U)Up --> 32	result = ({D[d], 16’h0000} - ((D[a] * D[b][31:16]) << n)) >> 16;D[c] = result[31:0]; // Fraction
0x01	**MSUB.Q D[c], D[d], D[a], D[b] L, n** 32 - (32 * 16L)Up --> 32	result = ({D[d], 16’h0000} - ((D[a] * D[b][15:0]) << n)) >> 16;D[c] = result[31:0]; // Fraction
0x02	**MSUB.Q D[c], D[d], D[a], D[b], n** 32 - (32 * 32)Up --> 32	result = ({D[d], 32’h0000_0000} - ((D[a] * D[b]) << n)) >> 32;D[c] = result[31:0]; // Fraction
0x20	**MSUBS.Q D[c], D[d], D[a], D[b] U, n** 32 - (32 * 16U)Up --> 32	result = ({D[d], 16’h0000} - ((D[a] * D[b][31:16]) << n)) >> 16;D[c] = ssov(result, 32); // Fraction

(table continues...)

Table 25 (continued) MSUB.Q Definitions in AURIX™ different from AUDO™

Secondary Opcode [23:18]	Instruction Mnemonic	Updated Description
0x21	**MSUBS.Q D[c], D[d], D[a], D[b] L, n** 32 - (32 * 16L)Up --> 32	result = ({D[d], 16’h0000} - ((D[a] * D[b][15:0]) << n)) >> 16;D[c] = ssov(result, 32); // Fraction
0x22	**MSUBS.Q D[c], D[d], D[a], D[b], n** 32 - (32 * 32)Up --> 32	result = ({D[d], 32’h0000_0000} - ((D[a] * D[b]) << n)) >> 32;D[c] = ssov(result, 32); // Fraction

4.41 [DAP_TC.H002] DAP client_blockread in Combination with TGIP and all Parcels with CRC6

Description

***Note**: This problem is only relevant for tool development, not for application development.*

When issuing a DAP client_blockread telegram together with the TGIP (Trigger in Protocol) option (DAPISC.TGIP = 1) the TGIP extra bit is appended for each parcel in case “all parcels with CRC6” is enabled. This causes a slight increase in the communication length compared to the correct behavior of having a TGIP bit only for the last parcel.

Recommendation

Do not use the TGIP and “CRC6 for all parcels” features together in case this extra bit can not be tolerated. If the Trigger in Protocol and increased communication safety is required TGIP can be used together with the CRC32 option (see also DAP_TC.002 DAP client_blockread has Performance issue in Specific Operation Modes).

4.42 [DAP_TC.H003] Not acknowledged DAP telegrams in noisy environments

Description

***Note**: This problem is only relevant for tool development, not for application development.*

DAP telegrams always follow a request-reply scheme. The request is driven by the tool, the reply by the AURIX™. The AURIX™ acknowledges a correctly received telegram always by a reply, which consists at least of a start-bit. DAP communication in noisy environments might result in invalid telegrams. This can leave the IOClient in an intermediate state which requires an IOClient reset.

If AURIX™ receives an invalid telegram with a wrong CRC6 or length field, it does not reply at all and in some cases the selected IOClient might be left in an intermediate state in case of a detected client_write/blockwrite/readwrite tool request.

Recommendation

If a tool does not receive a start bit as an acknowledge for an IOClient request, a client_reset must be sent as the next telegram for the selected IOClient. Tool interaction with the DAP module itself is not affected and can be done in between.

4.43 [DMA_TC.H002] Bit CHCSRz.BUFFER can be toggled when not in Double Buffer Mode

Description

The purpose of bit CHCSRz.BUFFER is to indicate which buffer is read or filled during DMA double buffering (selected in bit-field ADICRz.SHCT).

However, bit CHCSRz.BUFFER can also be toggled by writing bit CHCSRz.SWB = 1_B when not in Double Buffer Mode.

Recommendation

Do not write bit CHCSRz.SWB = 1_B when not in Double Buffer Mode.

4.44 [DMA_TC.H004] Transaction Request Lost upon software trigger with pattern match

Description

If a DMA channel is configured for pattern detection and software triggering of each DMA transfer (CHCSRz.RROAT = 0_B), then if there is a new DMA software request received while a DMA transfer is executing then a Transaction Request Lost event may be lost.

Recommendation

The loss of TRL status is a debug feature. A DMA channel should be used such that TRL is not set.

The user must ensure that the CPU triggers a new DMA software request when no DMA access is pending.The software could poll the TSRz.CH bit to confirm it is 0_B before issuing a DMA software trigger.

4.45 [DMA_TC.H005] Linked List Transfer leading to loading of non-Linked List TCS causes corruption

Description

If on completion of a Linked List (LL) a non-LL Transaction Control Set (TCS) is loaded with shadow address buffering enabled (read only and direct write) then the new non-LL TCS can be corrupted.

Recommendation

Shadow address buffering must be disabled in the non-LL TCS (SHCT[3:0] = 0000_B)

4.46 [DMA_TC.H006] Clearing of HTRE when DMA channel is configured for Single Mode

Description

The DMA may be used to support a peripheral with a high interrupt rate where the interrupts are generated in quick succession (for example a QSPI filling a TXFIFO).

The DMA channel z is configured with the following settings:

- Single Mode (HTRE is reset by hardware on completion of a DMA transaction)
  - TSRz.CHMODE = 0_B
- Request required for each DMA Transfer
  - TSRz.RROAT = 0_B

If the DMA channel is configured to execute a DMA transaction of 1 x DMA transfer of 2 x DMA moves:

- Block Mode: 2 x DMA Move per DMA transfer
  - DMA_CHCFGRz.BLKM = 001_B
- Transfer Reload Value: 1 x DMA transfer
  - DMA_CHCFGRz.TREL = 1_B

then additional DMA moves are executed unexpectedly.

Explanation of Effect

If the peripheral generates two interrupt service requests in relatively quick succession then the first DMA hardware request is serviced by the DMA and performs one DMA transfer comprising two DMA moves. The second DMA hardware request arrives before the completion of the first DMA transfer (i.e. before the clearing of HTRE at the end of the DMA transaction). The second hardware request is serviced by the DMA and performs a second DMA transfer comprising two DMA moves.

Recommendation

If the second DMA hardware request arrives before completion of the first DMA transfer then the DMA channel Block Mode must limit a DMA transfer to one DMA move:

- DMA_CHCFGRz.BLKM = 000_B; //1 x DMA move/DMA transfer

The total number of DMA moves must be defined by the Transfer Reload Value DMA_CHCFGRz.TREL.

4.47 [DMA_TC.H007] Selecting the Priority for DMA Channels

Description

All used DMA channels should be configured with the **highest** priority on SPB in respect to other used SPB master agents (CPUs, HSSL, ETH) to enable a robust execution of the configured DMA transactions.

The DMA channels are configured per default with the **lowest** priority on SPB:

- DMA_CHCFGRz.DMAPRIO = 00_B --> maps DMA channel z SPB requests to SPB priority DMAL
- SBCU_PRIOH.DMAL = 1111_B --> configures DMAL with the lowest priority on SPB

Recommendation

There are several ways to configure used DMA channels with the highest priority on SPB with respect to other SPB master agents. Two examples follow:

Example1

Map the used DMA channels to SPB priority DMAH by setting DMA_CHCFGRz.DMAPRIO = 11_B and keep the configuration of the DMAH priority (SBCU_PRIOL.DMAH = 0000_B).

Example2

Keep the mapping of the used DMA channels to DMAL (DMA_CHCFGRz.DMAPRIO = 00_B) and change the priority configuration of DMAL (for example set SBCU_PRIOH.DMAL = 0001_B).

Background

The DMA can request for SPB access with three different requests (DMAH, DMAM, DMAL) that are configured with different SPB priorities with respect to the other SPB master agents (CPUx, HSCT, ETH). The priority of the DMA requests DMAH, DMAM and DMAL on the SPB in respect to the priority of other SPB master agents can be configured via the SBCU registers SBCU_PRIOL / SBCU_PRIOH.

Each DMA channel z can be configured via DMA_CHCFGRz.DMAPRIO regarding which of three priorities (DMAH, DMAM or DMAL) it uses for SPB access.

The default configuration of DMA_CHCFGRz.DMAPRIO = 00_B. This means that the channels will request for SPB access with the DMAL priority.

The priority of a DMAL request on SPB is configured per default with the lowest priority (SBCU_PRIOH.DMAL = 1111_B).

4.48 [DMA_TC.H008] Transaction Request State

Description

The DMA Transaction Request State bit DMA_TSRz.CH is cleared when the DMA transfer starts (RROAT = 0_B) or at the end of a DMA transaction (RROAT = 1_B).

Figure "Channel Request Control" and RROAT bit-field description of register DMA_MExCHCR in chapter "Register Description" of the User's Manual are wrong.

4.49 [DMA_TC.H009] Resetting Bits ICH and IPM in register CHCSRz

Description

The Clear Interrupt from Channel bit (CICH) is accessible via the DMA channel CHCSR register.

The AURIX™ TC2xx User Manuals are incorrect with respect to the following statement:

- The DMA channel DMA_CHCSRz ICH and IPM bit-field description states: "is reset by software when writing a 1 to ADICRz.CICH"

Correction

- The text should read: "is reset by software when writing a 1 to **CHCSRz**.CICH"

4.50 [DMA_TC.H010] Calculation of DMA Address Checksum for DMA read moves to Cacheable Addresses

Description

The DMA Move Engine (ME) stores the DMA read move data in eight 32-bit read registers. If a DMA read move is to a cached address (Segment 8 or 9), the ME shall translate the DMA read move access to the on chip bus into an SRI BTR4 access to a 32-byte aligned address. The DMA shall calculate the DMA address checksum from the on chip bus address i.e. the 32-byte aligned address. The DMA shall store the DMA address checksum in the SDCRCR.

Recommendation

If an expected DMA address checksum is pre-calculated to test the DMA address generation, the user shall take note of the address translation to 32-byte aligned addresses when calculating the expected DMA address checksum from a cacheable DMA source address.

Alternatively, DMA read moves should be performed to non-cacheable source addresses (segments A and B).

4.51 [DMA_TC.H011] DMA_ADICRz.SHCT - Reserved Values

Description

The DMA channel shadow control bit-field DMA_ADICRz.SHCT controls the function of the shadow address register. If software programs a reserved value in DMA_ADICRz.SHCT, the DMA may deadlock the operation of the DMA.

Therefore, software shall not program DMA_ADICRz.SHCT with the following reserved values:

- 0011_B Reserved

- 0100_B Reserved
- 0111_B Reserved

4.52 [DMA_TC.H012] TCS Update in Halt State

Description

If a DMA channel is in halt state,

- The DMA shall stop performing DMA moves to the destination location
- Software may perform a background test on the destination location
- Software may modify the DMA channel Transaction Control Set (TCS)

Recommendation

If software modifies the DMA channel TCS, software shall only modify the DMA channel source address (DMA_SADRz.SDAR) and the DMA channel destination address (DMA_DADRz.DADR).

4.53 [DMA_TC.H013] MExSR.WS and MExSR.RS Status Bits

Description

As documented in the User’s Manual, the Move Engine (ME) status bits RS/WS in register MExSR are set when the ME is performing a read move or DMA write move. This means:

- MExSR.RS = 1_B when the ME is performing a DMA read move for the active DMA channel
- MExSR.WS = 1_B when the ME is performing a DMA write move for the active DMA channel

It should be noted that the setting of these bits is not restricted to DMA read move and DMA write move. Additionally the status bits may be set when the ME is performing other operations:

- MExSR.RS = 1_B when the ME is loading a new Transaction Control Set in a linked list
- MExSR.WS = 1_B when the ME is writing a DMA timestamp

***Note**: The additional setting of the ME status bits may be observed when debugging the operation of the DMA. There is no effect on the operation of the DMA.*

4.54 [DMA_TC.H016] DMARAM ECC Error Disable

Description

If software disables SPB bus errors caused by DMARAM ECC errors (DMA_MEMCON.ERRDIS = 1_B), the DMA will not correctly acknowledge a Read Modify Write (RMW) access on the SPB bus.

Recommendation

The application software must always enable the reporting of SPB errors (DMA_MEMCON.ERRDIS = 0_B; default after reset).

4.55 [DMA_TC.H017] DMA Channel Request Control - Documentation Update

Description

The following text (located below figure “Channel Request Control” in section “DMA Channel Request Control” of the DMA chapter in the User’s Manual):

“If CHCFGRz.PRSEL = 1 in the current DMA channel z can bypass the ICU and trigger a DMA hardware request in the next lower DMA channel z-1. The latency to service a DMA channel z-1 request is reduced. DMA channel z interrupt service requests are disabled.”

should read as :

“If DMA_CHCFGRz.PRSEL = 1 **is selected** in the current DMA channel z, **a DMA channel trigger** can bypass the ICU and trigger a DMA hardware request in the next lower DMA channel z-1. The latency to service a DMA channel z-1 request is reduced. DMA channel z interrupt service requests are disabled.”

4.56 [DTS_TC.H001] Update of Bit DTSSTAT.BUSY

Description

The following statement in the description of bit BUSY in register DTSSTAT in the SCU chapter “Die Temperature Measurement” is incorrect:

***Note**: This bit is updated 2 cycles after bit DTSCON.START is set.*

Correction

The correct description is as follows:

***Note**: This bit is updated **7** cycles after bit DTSCON.START is set.*

4.57 [ENDINIT_TC.H001] Endinit Protection for Registers KRST0, KRST1, KRSTCLR

Description

The access protection symbol ‘E’ to indicate Endinit-protection is missing in column “Access Mode - Write” in table “Register Overview” in the User’s Manual for the following registers:

- KRST0, KRST1, KRSTCLR

of the following modules (if implemented):

- E-Ray, ETH, PSI5

4.58 [FLASH_TC.H007] Advice for using Suspend and Resume

Description

As documented in the User’s Manual section “Operation Suspend and Resume”, an operation is suspended by writing ‘1’ to MARD.SPND. The Flash operation stops when it reaches an interruptible state. After that the flag FSR.SPND is set and BUSY is cleared.

The 1-to-0 transition of MARD.SPND alone is not indicating if the suspend request has been executed and the Flash can accept a new command. The BUSY flags have to be checked to determine if the Flash is still busy with the current operation. Only after the 1-to-0 transition of the BUSY flags the flag FSR.SPND indicates if the operation has finished or if it is in suspended state.

The following recipe describes the best practice for using suspend and resume.

Suspending an Erase Operation

In case of a request for suspending an ongoing erase operation:

As documented in the User's Manual: Please ensure that between start or resume of an erase process and the suspend request normally at least ~1 ms erase time can pass.

- Check if the corresponding BUSY flag has already cleared. If yes, no suspend is necessary

- Request the suspend with control flag MARD.SPND = 1_B
- Wait until the BUSY flag clears
- After that check FSR.SPND. If this is 1_B then the operation was suspended and needs to be resumed later. If this is 0_B the operation has already finished, therefore no resume is necessary
- Now new Flash operations are allowed with the restrictions documented in User's Manual section “Operation Suspend and Resume”

Note for PFlash erase operations in bank x that PxBUSY and D0BUSY are set at the beginning. The D0BUSY is cleared early after updating the Erase Counters, and PxBUSY is cleared when the erase operation has finished. Therefore, for PFlash the PxBUSY flag has to be used. (Polling for PxBUSY and DxBUSY can be a generic solution for suspend sequences before checking the SPND state.) Interrupt driven software receives two interrupts!

Resuming a Suspended Erase Operation

The resume of the suspended erase operation is done in these steps:

- Resume the operation with the command sequence “Resume Prog/Erase”
- Wait until FSR.SPND is 0_B
- After that wait for the end of the operation signalled by BUSY going to 0_B

Suspending a Program Operation

In case of a request for suspending an ongoing programming operation:

- Request the suspend with control flag MARD.SPND = 1_B
- Wait until the BUSY flag clears
- After that check FSR.SPND. If this is 1_B then the operation was suspended and needs to be resumed later. If this is 0_B the operation has already finished, therefore no resume is necessary
- Now new Flash operations are allowed with the restrictions documented in User's Manual section “Operation Suspend and Resume”

Resuming a Suspended Program Operation

The resume of the suspended programming operation is done in these steps:

- Resume the operation with the command sequence “Resume Prog/Erase”
- Wait until FSR.SPND is 0_B
- After that wait for the end of the operation signalled by BUSY going to 0_B

4.59 [FLASH_TC.H008] Understanding Flash Retention/Endurance Figures in the Data Sheet

Description

Flash retention/endurance is documented in the Data Sheet by the following parameters

- Program Flash Retention Time t_{RET} for PFlash
- UCB Retention Time t_{RTU} for the UCBs
- Data Flash Endurance per EEPROMx sector N_{E_EEP10} for DFlash0
- Data Flash Endurance per HSMx sector N_{E_HSM} for DFlash1 (if available)

Retention

To emphasize the importance of retention, the PFlash and UCB parameters are described as retention time under the condition of a maximum number of cycles.

The value “Min. x years” has to be interpreted as: the data retention is at least x years, i.e. x years or longer after the last programming data stays readable.

The condition “Max. y erase/program cycles” means: this data retention figure is valid if there were not more than y erase/program cycles.

Endurance

For the DFlash the endurance is most important, therefore as parameter the number of cycles under the condition of the retention is given.

The value “Min. x cycles” has to be interpreted as: at least x cycles can be applied.

The condition “Max. data retention time y years” means: this endurance figure is valid if the expected data retention after the last programming is maximum y years.

***Note**: As general remark, these figures are only valid if the parameters given in the Data Sheet are adhered to in their entirety.*

4.60 [FLASH_TC.H022] Flash Wait State configuration

Description

Configuring flash wait states in your application is critical for correct operation.

Refer to these parts of the documentation of the respective TC2*x design step for guidance on avoiding data read errors over the lifetime of the device:

- Data Sheet, chapter “Flash Parameters”:
  - minimum access times t_{PF} / t_{PFECC} for PFLASH
  - and t_{DF} / t_{DFECC} for DFLASH
- AURIX™ TC2*x User’s Manual, PMU chapter “Configuring Flash Wait Cycles”

When **increasing** the SRI and FSI clock frequencies: first set the wait state bit-fields (WSECPF, WSPFLASH, WSECDF, and WSDFLASH) in register FCON to the correct values, and then change the clock configuration.

When **decreasing** the SRI and FSI clock frequencies: first change the clock configuration, and then set the wait state bit-fields (WSECPF, WSPFLASH, WSECDF, and WSDFLASH) in register FCON to the correct values.

***Note**: Applications that omit configuration of FCON may work in the development phase, but encounter data read errors in the field.*

4.61 [FlexRay_AI.H004] Only the first message can be received in External Loop Back mode

Description

If the loop back (TXD to RXD) will be performed via external physical transceiver, there will be a large delay between TXD and RXD.

A delay of two sample clock periods can be tolerated from TXD to RXD due to a majority voting filter operation on the sampled RXD.

Only the first message can be received, due to this delay.

To avoid that only the first message can be received, a start condition of another message (idle and sampling '0' -> low pulse) must be performed.

The following procedure can be applied at one or both channels:

- wait for no activity (TEST1.AOx = 0 -> bus idle)
- set Test Multiplexer Control to I/O Test Mode (TEST1.TMC = 2), simultaneously TXDx = TXENx = 0
- wait for activity (TEST1.AOx = 1 -> bus not idle)
- set Test Multiplexer Control back to Normal signal path (TEST1.TMC = 0)
- wait for no activity (TEST1.AOx = 0 -> bus idle)

Now the next transmission can be requested.

4.62 [FlexRay_AI.H005] Initialization of internal RAMs requires one eray_bclk cycle more

Description

The initialization of the E-Ray internal RAMs as started after hardware reset or by CHI command CLEAR_RAMS (SUCC1.CMD[3:0] = 1100_B) takes 2049 eray_bclk cycles instead of 2048 eray_bclk cycles as described in the E-Ray Specification.

Signalling of the end of the RAM initialization sequence by transition of MHDS.CRAM from 1_B to 0_B is correct.

4.63 [FlexRay_AI.H006] Transmission in ATM/Loopback mode

Description

When operating the E-Ray in ATM/Loopback mode there should be only one transmission active at the same time. Requesting two or more transmissions in parallel is not allowed.

To avoid problems, a new transmission request should only be issued when the previously requested transmission has finished. This can be done by checking registers TXRQ1/2/3/4 for pending transmission requests.

4.64 [FlexRay_AI.H007] Reporting of coding errors via TEST1.CERA/B

Description

When the protocol engine receives a frame that contains a frame CRC error as well as an FES decoding error, it will report the FES decoding error instead of the CRC error, which should have precedence according to the non-clocked SDL description.

This behavior does not violate the FlexRay protocol conformance. It has to be considered only when TEST1.CERA/B is evaluated by a bus analysis tool.

4.65 [FlexRay_AI.H009] Return from test mode operation

Description

The E-Ray FlexRay IP-module offers several test mode options

- Asynchronous Transmit Mode
- Loop Back Mode
- RAM Test Mode
- I/O Test Mode

To return from test mode operation to regular FlexRay operation we strongly recommend to apply a hardware reset via input eray_reset to reset all E-Ray internal state machines to their initial state.

***Note**: The E-Ray test modes are mainly intended to support device testing or FlexRay bus analyzing. Switching between test modes and regular operation is not recommended.*

4.66 [FlexRay_AI.H011] Behavior of interrupt flags in FlexRay™ Protocol Controller (E-Ray)

Description

In the corner case described below, the actual behavior of the interrupt flags of the FlexRay™ Protocol Controller (E-Ray) differs from the expected behavior.

***Note**: This behaviour only applies to E-Ray interrupts INT0 and INT1. All other E-Ray interrupts are not affected.*

Expected behavior

When clearing an interrupt flag by software, the resulting value of the flag is expected to be zero.

A hardware event that occurs afterwards then leads to a zero to one transition of the flag, which in turn leads to an interrupt service request.

Actual behavior in corner case

When the interrupt flag is being cleared by software in the same clock cycle as a new hardware event sets the flag again, then the hardware event wins and the flag remains set without being cleared.

As interrupt requests are generated only upon zero to one transitions of the flag, no interrupt request will be generated for this flag until the flag is successfully cleared by software later on.

Workaround

After clearing the flag, the software shall read the flag and repeat clearing until the flag reads zero.

4.67 [FlexRay_TC.H002] Initialization of E-Ray RAMs

Description

After Power-on reset the ECC codes in the E-Ray RAMs may be set to an arbitrary state. Therefore the E-Ray RAM must be cleared and the ECC codes set to a defined state to avoid unintended traps.

To achieve this the following alternative methods are proposed:

Method 1 using the MTU/MBIST

- Clear all E-Ray RAMs and the related ECC code storage by executing writes to all RAM locations using the AURIX™ MBIST engine. The MBIST engine supports filling the E-Ray RAM with ECC-correct patterns. For this purpose the AURIX™ MBIST auto-initialization algorithm can be used. See section “Filling a Memory with Defined Contents” in the corresponding User’s Manual/Target Specification. The following E-Ray RAM blocks have to be initialized with correct data:
  - Output Buffer
  - Input Buffer
  - Message Buffers The MBIST function to be executed for each buffer is the same, only the function parameters have to be adapted
- Execute one read from each E-Ray RAM block using the AURIX™ MBIST engine (reading from all E-Ray RAM locations is an alternative but not necessary solution). For this purpose the AURIX™ MBIST engine can also be used
- Insert at the end of all MBIST function calls a status check, which makes sure that the launched MBIST tests are finished (check MSTATUS.DONE status flag)
- Clear all ECC error flags in the E-Ray module: these are flag EERR in register EIR, flags EIBF, EOBF, EMR, ETBF1, ETBF2 in register MHDS. The flags are cleared by writing a ‘1’ to the according bit position in the flag register

After these steps the E-Ray RAM can be used for further operation, for example for initialization of the E-Ray buffer.

***Note**: For **TC27x step BC**, due to issue MTU_TC.008 (Reading register MSTATUS while a test is running), the following delay times (including a 100 clock cycle "safety margin" for communication overhead between CPU and MBIST) must be respected before checking bit MSTATUS.DONE:*

- *Output Buffer fill operation: 164 clocks*
- *Input Buffer fill operation: 228 clocks*
- *Message Buffer fill operation: 2148 clocks*

Method 2 using "CLEAR RAMS" Command

Step 1 to 4: Enable the clock of the module:

1. Remove EINIT protection for the writing of the CLC register
2. Enable the clock in the CLC register
3. Read the CLC register
4. Enable the EINIT protection

Enable the test mode, check if the state of the module is according to the expected settings and start clearing the RAMs.

5. Take care of the unlock sequence. See description of LCK.TMK and TEST1.WRTEN in User's Manual:

- Test Mode Key: To set bit TEST1.WRTEN the write operation has to be directly preceded by two consecutive write accesses to the Test Mode Key
- If the write sequence is interrupted by other write accesses between the second write to the Test Mode Key and the write access to the TEST1 register, bit TEST1.WRTEN is not set to 1 and the sequence has to be repeated.

  First write: LCK.TMK = 75_H = $0111\ 0101_B$

  Second write: LCK.TMK = $8A_H$ = $1000\ 1010_B$

  Second write: TEST1.WRTEN = 1_B

6. Check if CCSV.POCS is either 0x0 (DEFAULT_CONFIG) or 0xF (CONFIG). If not in any of these states, perform the according command to get to CONFIG state
7. Check if SUCC1.PBSY is equal 0x0. If 0x1 wait until 0x0
8. Set SUCC1.CMD to 0xC meaning that the CLEAR_RAMS command is entered
9. Read SUCC1.CMD. If 0x0 the command has not been accepted. Repeat up from step 7. Otherwise continue
10. Wait 1024 module cycles
11. Enable RAM Test mode: TEST1.TMC = 01_B. This mode enables access of all RAM blocks in E-Ray modules to the host
12. CUST1.IBF1PAG := 1_B
13. CUST1.IBF2PAG := 1_B
14. Repeat steps 7 to 10
15. Read at least one address in all the RAM blocks within E-Ray module
16. Switch off Test mode: TEST1.TMC = 00_B and TEST1.WRTEN = 0_B
17. Clear ECC error flags in MHDS and EIR registers
18. From here you can start the normal initialization process of the module

***Note**: In order to ensure proper FlexRay communication, RAM test mode must be explicitly disabled via TEST1.TMC = 00_B in step 16 at the end of the initialization sequence.*

4.68 [FlexRay_TC.H004] Bit WRECC in register TEST2 has no function

Description

In the AURIX™ implementation of the E-Ray module, bit WRECC in register TEST2 has no function.

Recommendation

The value read from WRECC should not be evaluated by software, the value written (0_B or 1_B) to it is irrelevant. For new software projects, keep bit WRECC at its reset value (0_B) for easier migration to future AURIX™ generations.

4.69 [FlexRay_TC.H005] E-Ray OTGB2 trigger set active even if disabled

Description

The trigger set TS32_SCSC of the E-Ray IP-module is associated with OTGB2. An internal “valid” signal should be asserted only in case the trigger set is selected via OTSS.OTGB2.

Expected behavior

The OTGB2 trigger set valid signal should be gated by the bit-field OTSS.OTGB2.

Actual behavior

The E-Ray IP does not gate the valid signal with the OTSS.OTGB2 state, but only the data are gated. Meaning the OTGB2 trigger valid signal is only dependent on the slot counter and transfer buffer state changes, irrespective of the OTSS.OTGB2 value.

Recommendation

Ignore all OTGB2 E-Ray triggers when data reported is only 0s.

4.70 [FPI_TC.H002] Write access to register ACCEN1

Description

The ACCEN1 (Access Enable Register 1) registers in the AURIX™ devices are reserved for future expansion. The bits in the ACCEN1 registers are described as "Reserved", read-only. There is no need for software to configure (write to) the ACCEN1 registers.

Note*:* *For a write access to the ACCEN1 registers in the following modules, a bus error will be generated:*

- ***TC2xx****: MTU, SMU, ETH, I2C, FFT, CIF*
- ***TC3xx****: MTU, SMU (documented in TC3xx user manual), CIF*

4.71 [GPT12_TC.H001] Timer T5 Run Bit T5R - Documentation Correction

Description

In the current version of the User’s Manual, the lines for T5R=0_B and T5R=1_B in the register description of the Timer T5 Run Bit (T5R) erroneously have been swapped.

Correction

The correct behavior of bit T5R is as shown in Table 26: T5R=0_B (Timer T5 stops; default after reset), T5R=1_B (Timer T5 runs).

Table 26 Timer T5 Control Register T5CON, Bit T5R - Correction

Field	Bits	Type	Description
T5R	6	rw	**Timer T5 Run Bit** 0_B Timer T5 **stops** 1_B Timer T5 **runs** ***Note***: *This bit only controls timer T5 if bit T5RC = 0.*

4.72 [GPT12_TC.H002] Bits TxUD and TxUDE in incremental interface mode - Additional information

Description

The present description of the incremental interface mode for timers T2, T3, T4 in the User's Manual, including figures and tables, implicitly refers to the following configuration of bits TxUD and TxUDE (x = 2, 3, 4):

- TxUD = 0_B
- TxUDE = 1_B

This is the recommended and validated setting for these bits in incremental interface mode.

Additional information

When bit TxUD = 1_B, the count direction of timer Tx is inverted compared to the setting with TxUD = 0_B in incremental interface mode.

The setting of bit TxUDE is irrelevant in incremental interface mode, the behavior of Tx for TxUDE = 0_B and TxUDE = 1_B is identical. The figures related to incremental interface mode shall be interpreted as if TxUDE is permanently tied to 1_B.

4.73 [GTM_AI.H519] SPEC-(A)TOM: Misleading description of Continuous Counting Up Mode

Description

In the third list item of the paragraph, where some statements are given for Continuous Counting Up Mode with RST_CCU0=1, the following statement for the case (A)TOM[i]_CH[x]_CM0.CM0=(A)TOM[i]_CH[x]_CM1.CM1 is given:

"If (A)TOM[i]_CH[x]_CM0.CM0=(A)TOM[i]_CH[x]_CM1.CM1, the output switches to (A)TOM[i]_CH[x]_CTRL_SOMP.SL if (A)TOM[i]_CH[x]_CN0.CN0=(A)TOM[i]_CH[x]_CM0.CM0=(A)TOM[i]_CH[x]_CM1.CM1 ((A)TOM[i]_CH[x]_CM0.CM0 has higher priority."

or in the older specification versions (before GTM3 generations):

"If (A)TOM[i]_CH[x]_CM0.CM0=(A)TOM[i]_CH[x]_CM1.CM1, the output is 100% (A)TOM[i]_CH[x]_CTRL.SL ((A)TOM[i]_CH[x]_CM0.CM0 has higher priority."

Both statements are misleading and have to be replaced by the following statement:

"As soon as (A)TOM[i]_CH[x]_CN0.CN0 reaches the value of (A)TOM[i]_CH[x]_CM0.CM0 while (A)TOM[i]_CH[x]_CM1.CM1 is equal to (A)TOM[i]_CH[x]_CM0.CM0, an edge to (A)TOM[i]_CH[x]_CTRL.SL is generated at the output or the output remains at (A)TOM[i]_CH[x]_CTRL.SL level, depending on the former level of the output ((A)TOM[i]_CH[x]_CM0.CM0 has higher priority)."

***Note**: The above configuration is not suitable for generating 100% duty cycle.*

Scope

ATOM, TOM

Effects

Textual description can be erroneously interpreted as the configuration of (A)TOM[i]_CH[x]_CM0.CM0=(A)TOM[i]_CH[x]_CM1.CM1 is suitable to generate 100% duty cycle for the current PWM period.

This is because the potential value change to SL will happen as soon as (A)TOM[i]_CH[x]_CN0.CN0 reaches the value of (A)TOM[i]_CH[x]_CM0.CM0 while (A)TOM[i]_CH[x]_CM1.CM1 is equal to (A)TOM[i]_CH[x]_CM0.CM0.

Recommendation

For a setup of 100% duty cycle for Continuous Counting Up Mode with RST_CCU0=1, the following setting must be used:

- ATOM[i]_CH[x]_CM0.CM0 = 0 and ATOM[i]_CH[x]_CM1.CM1 > MAX

4.74 [GTM_AI.H803] SPEC-(A)TOM: Missing priority information for register update

Description

The following information is missing in the specification and has to be placed inside the TGC Sub-unit/AGC Sub-unit chapter:

- Inside ATOM chapter: The trigger condition has always priority over the bus write access to the ATOM[i]_AGC_OUTEN_STAT and ATOM[i]_AGC_ENDIS_STAT registers, even if ATOM[i]_AGC_OUTEN_CTRL.OUTEN_CTRL[k] / ATOM[i]_AGC_ENDIS_CTRL.ENDIS_CTRL[k] is set to 00_B. This means that the bus write access to ATOM[i]_AGC_OUTEN_STAT and ATOM[i]_AGC_ENDIS_STAT register is ignored in the clock cycle when the trigger condition is active
- Inside TOM chapter: The trigger condition has always priority over the bus write access to the TOM[i]_TGC[g]_OUTEN_STAT and TOM[i]_TGC[g]_ENDIS_STAT registers, even if TOM[i]_TGC[g]_OUTEN_CTRL.OUTEN_CTRL[k] / TOM[i]_TGC[g]_ENDIS_CTRL.ENDIS_CTRL[k] is set to 00_B. This means that the bus write access to TOM[i]_TGC[g]_OUTEN_STAT and TOM[i]_TGC[g]_ENDIS_STAT register is ignored in the clock cycle when the trigger condition is active

***Note**: The trigger override does not happen if the trigger is a HOST_TRIG, as this is initiated by a bus write itself and cannot happen at the same time as another bus write to the register.*

***Note**: This AppHint is published as GTM-IP-523 by Bosch.*

Scope

ATOM, TOM

Effects

In (A)TOM the bus write access to the "OUTEN_STAT" and "ENDIS_STAT" registers is overridden by a trigger update and the desired values are not written into the register.

Recommendation

To nevertheless ensure that the desired value is actually stored in the target register, consider one of the following hints:

1. Write first the channel k within "ENDIS_CTRL" ("OUTEN_CTRL") and write the desired channel k in "ENDIS_STAT" ("OUTEN_STAT") afterward. This way, either the asynchronous write or the synchronous write becomes effective
2. If recommendation 1 is not the case, then read back the value of the "OUTEN_STAT" and "ENDIS_STAT" register to ensure the written value is actually present in the register

4.75 [GTM_TC.H004] Correction to Bit Fields GTM_TIMi_IN_SRC.VAL_x

Description

In the description of bit-field VAL_0, the encoding 01_B was erroneously repeated while 10_B and 11_B were missing.

The affected registers are

- GTM_TIMi_IN_SRC in the TC29x, TC27x, and TC26x User’s Manual
- GTM_TIM0_IN_SRC in the TC21x/TC22x/TC23x Family User’s Manual

The correct description is included in the following tables. As the description of bit fields VAL_x, x>0 refers to VAL_0, this description is valid for all VAL_x bit fields in registers GTM_TIMi_IN_SRC and GTM_TIM0_IN_SRC, respectively.

Correction for TC29x, TC27x, and TC26x

Table 27 TC29x .. TC26x: Corrected Description of Bit Field VAL_0 in Registers GTM_TIMi_IN_SRC

Field	Bits	Type	Description
VAL_0	[1:0]	rw	**Value to be fed to Channel 0** multicore encoding in use (VAL_x(1) defines the state of the signal) 00_B State is 0 (ignore write access) 01_B Change state to 0 10_B Change state to 1 11_B State is1 (ignore write access) ...

Correction for TC21x/TC22x/TC23x

Table 28 TC23x .. TC21x: Corrected Description of Bit Field VAL_0 in Register GTM_TIM0_IN_SRC

Field	Bits	Type	Description
VAL_0	[1:0]	rw	**Value to be fed to Channel 0** 00_B Input signal 0 (ignore write access) 01_B Input signal is set to 0 10_B Input signal is set to 1 11_B Input signal 1 (ignore write access) ...

4.76 [GTM_TC.H005] External Capture in TIM Pulse Integration Mode (TPIM)

Description

In table “TIM integration Mode” in section “External Capture in TIM Pulse Integration Mode (TPIM)” of the GTM chapter in the User’s Manual, the information that CNT is cleared upon external capture is missing in column “Action description”.

The corrected Table 29 is shown below:

Table 29 TIM integration Mode

Input signal F_OUTx	selected CMU clock	External capture	ISL	DSL	Action description
0	1	0	-	0	CNT++
1	1	0	-	0	no
1	1	0	-	1	CNT++
0	1	0	-	1	no
-	-	rising edge	-	-	do GPRx capture; issue NEWVAL_IRQ; **CNT = 0**
-	0	0	-	-	no

4.77 [GTM_TC.H007] GTM to CAN Timer Triggers

Description

The CAN transmit trigger inputs of the individual CAN nodes are connected to GTM trigger outputs as specified in table “CAN Transmit Trigger Inputs” in the MultiCAN+ chapter of the User’s Manual.

The corresponding GTM TOM/ATOM channel is selected in register GTM_CANOUTSEL as specified in tables “CAN Timer Triggers” in the GTM chapter. Note that not all specified SELx bit-fields in register CANOUTSEL are used for trigger selection.

The following GTM to CAN connections are implemented:

Table 30 GTM to CAN Connections in TC29x

CAN Node	GTM Trigger Selection via Bit Field
CAN Node 0	CANOUTSEL.SEL0
CAN Node 1	CANOUTSEL.SEL1
CAN Node 2	CANOUTSEL.SEL2
CAN Node 3	CANOUTSEL.SEL3
CANR Node 0	No GTM trigger
CANR Node 1	No GTM trigger

Table 31 GTM to CAN Connections in TC27x

CAN Node	GTM Trigger Selection via Bit Field
CAN Node 0	CANOUTSEL.SEL0
CAN Node 1	CANOUTSEL.SEL1
CAN Node 2	CANOUTSEL.SEL2

(table continues...)

Table 31 (continued) GTM to CAN Connections in TC27x

CAN Node	GTM Trigger Selection via Bit Field
CAN Node 3	CANOUTSEL.SEL3

Table 32 GTM to CAN Connections in TC26x

CAN Node	GTM Trigger Selection via Bit Field
CAN Node 0	CANOUTSEL.SEL0
CAN Node 1	CANOUTSEL.SEL1
CAN Node 2	CANOUTSEL.SEL2
CAN Node 3	CANOUTSEL.SEL3
CAN Node 4	CANOUTSEL.SEL4

Table 33 GTM to CAN Connections in TC23x

CAN Node	GTM Trigger Selection via Bit Field
CAN Node 0	CANOUTSEL.SEL0
CAN Node 1	CANOUTSEL.SEL1
CAN Node 2	CANOUTSEL.SEL2
CAN1 Node 0	CANOUTSEL.SEL0
CAN1 Node 1	CANOUTSEL.SEL1
CAN1 Node 2	CANOUTSEL.SEL2

Table 34 GTM to CAN Connections in TC22x/TC21x

CAN Node	GTM Trigger Selection via Bit Field
CAN Node 0	CANOUTSEL.SEL0
CAN Node 1	CANOUTSEL.SEL1
CAN Node 2	CANOUTSEL.SEL2

4.78 [GTM_TC.H009] TIM0 Channel x Input Selection - Mapping for QFP-80 and QFP-100 Packages

Description

Basically, the mapping of TIM0 input channels to port pins follows a strict family concept: functions available in a lower pin-count package are located on the same port pin in the next higher pin-count package.

In tables “TIM 0 Mapping for QFP-80” and “TIM 0 Mapping for QFP-100” in chapter “Port to GTM Control Registers” of the GTM chapter in the User’s Manual, some rows are incorrect. The following Table 35 and Table 36 show the corresponding corrections.

***Note**: Table “TIM 0 Mapping for QFP-144/BGA-292” in the User’s Manual is correct, as well as the tables in chapter “Port Connections” of the GTM chapter, and the GTM connections listed in the Data Sheet.*

Recommendation

For the correct port connections on QFP-80 and QFP-100 packages, use tables “GTM to Port Mapping for QFP-80” and “GTM to Port Mapping for QFP-100” in chapter “Port Connections” of the GTM chapter, or the Data Sheet.

Corrections

The following Table 35 and Table 36 show the **corrected** rows of tables “TIM 0 Mapping for QFP-80” and “TIM 0 Mapping for QFP-100”.

***Note**: Connections for CHxSEL encodings not listed in Table 35 or Table 36 are correctly printed in the corresponding tables in the User’s Manual.*

Table 35 Corrections to Table “TIM 0 Mapping for QFP-80”

Field CHxSEL	QFP-80: Pad / Input	Name
CH0SEL		
0100_B	Reserved	-
0101_B	Reserved	-
0111_B	Reserved	TIN53
1000_B	Reserved	-
1011_B	P02.8	TIN8
1100_B	Reserved	-
CH1SEL		
0011_B	Reserved	-
0100_B	P14.6	TIN86
0101_B	Reserved	-
0110_B	Reserved	TIN54
1000_B	P33.5	TIN27
CH2SEL		
0001_B	Reserved	-
0011_B	Reserved	-
0100_B	P10.5	TIN107
0101_B	Reserved	-
0110_B	Reserved	-
1000_B	Reserved	-
1001_B	P33.6	TIN28
CH3SEL		
0001_B	Reserved	-
0011_B	Reserved	-
0100_B	P10.6	TIN108
0110_B	Reserved	-
1001_B	P33.7	TIN29
CH4SEL		
0010_B	Reserved	-
0100_B	Reserved	-

(table continues...)

Table 35 (continued) Corrections to Table "TIM 0 Mapping for QFP-80"

Field CHxSEL	QFP-80: Pad / Input	Name
0101_B	Reserved	-
CH5SEL		
0011_B	Reserved	-
0100_B	Reserved	-
0110_B	Reserved	-
CH6SEL		
0100_B	P23.1	TIN42
0101_B	Reserved	-
0110_B	Reserved	-
CH7SEL		
0010_B	P14.4	TIN84
0011_B	P20.8	TIN64
0100_B	Reserved	-
0101_B	Reserved	-
0110_B	Reserved	-

Table 36 Corrections to Table "TIM 0 Mapping for QFP-100"

Field CHxSEL	QFP-100: Pad / Input	Name
CH0SEL		
0001_B	Reserved	-
1010_B	Reserved	-
1100_B	Reserved	-
CH2SEL		
1000_B	Reserved	-
CH4SEL		
1001_B	Reserved	-
1010_B	Reserved	-

4.79 [GTM_TC.H011] First CM0 updates in case of SR0=1 and (A)TOM used as Triggered Channel

Description

In case the CM0 register should be updated from the shadow register with 1, the Force Update mechanism (FUPD(x) signal) has to be enabled on the (A)TOM channel. Otherwise the first edge triggered from CM0 will not be generated after 1 appears in CM0.

4.80 [GTM_TC.H014] Synchronous Bridge Mode Restrictions

Description

The reset value for register GTM_BRIDGE_MODE is specified as $0400\ 1001_H$, and should never be changed according to the User’s Manual, i.e. the AEI bridge should always operate in async_bridge mode.

Exception

In order to improve access latency, operation in synchronous bridge mode is possible if it is ensured that the SPB frequency is identical to the GTM frequency:

- $f_{SPB} == f_{GTM}$

Sequence to configure the bridge in synchronous mode (pseudocode):

```
/* ensure that no data are read or written in the GTM */
if(fSPB == fGTM)
{
GTM_BRIDGE_MODE = 0x04011000; /* switch to sync mode, reset bridge*/
while(GTM_BRIDGE_MODE & 0x100) /* wait till mode change completed */
;
}
else
;
```

4.81 [GTM_TC.H015] Register TIMi_CHx_CTRL - Correction to Register Image

Description

The register image of the following registers erroneously shows bit 19 as “Reserved” with type “r” (read only):

- TIMi_ CHx_CTRL in the TC29x, TC27x and TC26x User’s Manual
- TIM0_ CHx_CTRL in the TC21x/TC22x/TC23x Family User’s Manual

Correction

Actually, bit 19 has type “rw” and is correctly described in the register table as copied from the User’s Manual in the table below:

Table 37 Bit EXT_CAP_EN in Register TIMi_CHx_CTRL and TIM0_CHx_CTRL, respectively

Field	Bits	Type	Description
EXT_CAP_EN	19	**rw**	**Enables external capture mode** The selected TIM mode is only sensitive to external capture pulses, the input event changes are ignored 0_B External capture disabled 1_B External capture enabled

4.82 [GTM_TC.H020] GTM can cause unintended bus errors after enabling when SPB or GTM frequency is very low

Description

When the SPB frequency is low compared to the CPU frequency, or the GTM frequency is low compared to the SPB frequency, the GTM can cause an FPI bus error when it is accessed too early after being enabled.

Recommendation

To avoid an FPI bus error, after enabling the GTM via the DISR bit in register CLC, a time delay of 10 SPB clock cycles and 10 GTM clock cycles must be inserted before accessing any GTM kernel register.

4.83 [GTM_TC.H025] Field TOCTRL in register GTM_TIM0_CHx_CTRL - Documentation correction

Description

In the GTM chapter of the TC21x/TC22x/TC23x User's Manual V1.1, the description of the edge selection in field TOCTRL of register GTM_TIM0_CHx_CTRL is incorrect.

Correction

The correct encoding for field TOCTRL is shown below:

Table 38 Encoding of field TOCTRL - Correction

Field	Bits	Type	Description
TOCTRL	[31:30]	rw	**Timeout Control** 00_B Timeout feature disabled 01_B Timeout feature enabled for rising edge only 10_B Timeout feature enabled for falling edge only 11_B Timeout feature enabled for both edges

4.84 [INT_TC.H004] Corrections to the Interrupt Router Documentation

Description

The following corrections apply to chapter "Interrupt Router (IR)" of the TC21x/TC22x/TX23x Family User's Manual:

Figure "Block Diagram of the TC21x/TC22x/TC23x Interrupt System" erroneously shows ICU3 related to DMA.

- **Correction**:
  - Only ICU0 and ICU1 are implemented, with **ICU1** related to the DMA

Table "Registers Overview - System, OTGM and ICU Control Registers" erroneously shows ICU1 registers INT_LWSR1, INT_LASR1, INT_ECR1 related to CPU1.

- **Correction**:
  - Registers INT_LWSR1, INT_LASR1, INT_ECR1 are related to the **DMA**

4.85 [IOM_TC.H001] How to clear the IOM_LAMEWCm register

Description

The Logic Analyzer Module Event Window Count Status register IOM_LAMEWCm stores the window count value reached prior to being cleared in the LAM block once an event has been generated.

Writing to IOM_LAMEWCm by software will result in a bus error.

The IOM_LAMEWCm register can be reset (cleared) by software with a write to the IOM_LAMCFGm or IOM_LAMEWSm registers, for example by writing the same configuration data that have been read to either of these registers.

***Note**: The clock divider should be set to IOM_CLC.RMC = 1 when configuring the IOM (see issue IOM_TC.004 "Write to IOM register space when IOM_CLC.RMC > 1").*

4.86 [IOM_TC.H002] IOM Clock Control

Description

Contrary to the named clocks given within the subsections of the IOM chapter, the entire IOM operates at the higher of the SPB or GTM clock frequencies. This may be further divided via the RMC bit-field of the IOM_CLC register, where the physical RMC value represents the divisor. For example, RMC = 00000001_B divides clock by 1, RMC = 00000010_B divides clock by 2, and so on. Note that RMC = 00000000_B disables the clock.

See also the following revised description of the IOM_CLC register.

IOM Clock Control Register (IOM_CLC)

The Clock Control Register CLC allows the programmer to adapt the functionality and power consumption of the module to the requirements of the application. The description below shows the clock control register functionality which is implemented in the BPI_FPI for the module. Where a module kernel is connected to the CLC clock control interface, CLC controls the f_{IOM} module clock signal, sleep mode and disable mode for the module.

Table 39 Description of Fields in IOM Clock Control Register (IOM_CLC)

Field	Bits	Type	Description
DISR	0	rw	**Module Disable Request Bit** Used for enable/disable control of the module. 0_B Module disable is not requested 1_B Module disable is requested
DISS	1	rh	**Module Disable Status Bit** Bit indicates the current status of the module. 0_B Module is enabled 1_B Module is disabled
0	2	rw	**Reserved** Read as 0; should be written with 0.
EDIS	3	rw	**Sleep Mode Enable Control** Used to control module's sleep mode. 0_B Sleep mode request is regarded. Module is enabled to go into Sleep Mode. 1_B Sleep mode request is disregarded. Sleep Mode cannot be entered upon a request.

(table continues...)

Table 39 (continued) Description of Fields in IOM Clock Control Register (IOM_CLC)

Field	Bits	Type	Description
RMC	[15:8]	rw	**Clock Divider Value in Run Mode** 00000000_B No clock signal f_{IOM} generated (default after reset) 00000001_B Clock f_{IOM} = max (f_{SPB}, f_{GTM}) selected 00000010_B Clock f_{IOM} = max (f_{SPB}, f_{GTM})/2 selected 00000011_B Clock f_{IOM} = max (f_{SPB}, f_{GTM})/3 selected ... 11111111_B Clock f_{IOM} = max (f_{SPB}, f_{GTM})/255 selected
0	[31:16], [7:4]	r	**Reserved** Read as 0; should be written with 0.

4.87 [IOM_TC.H003] Configuration of LAMCFG.IVW and LAMEWS.THR

Description

As shown in figure “Logic Analyzer Module (LAM) block diagram” in the IOM chapter of the User’s Manual, an EVENT will be generated if the required edge is detected and the XOR between the Event Window value and the invert bit (LAMCFG.IVW) is 1.

When the edge to be detected arrives at LAMEWSn.THR value of the counter, the EVENT will be generated depending on LAMCFG.IVW value:

- If LAMCFG.IVW==0 event will be generated
- if LAMCFG.IVW==1 event will not be generated

Taking this behavior into account, the description of the LAMCFG.IVW and/or LAMEWS.THR configuration in examples 2, 4, 5 and 6 of section “Example Monitor/Safety Measures” is misleading.

Correction

The corrected description, including the case “equal to”, is as follows (only modified lines are printed):

Example 2 - Pulse or duty cycle too long

LAMCFG.IVW: 0x0 ; don’t invert window, capture events when the counter is **equal or** above the threshold.

LAMEWS.THR: select appropriate threshold (maximum duty cycle length required. If duty cycle is longer than this value then an event will be triggered).

Example 4 - Period too long

LAMCFG.IVW: 0x0 ; don’t invert window, capture events when the counter is **equal or** above the threshold.

LAMEWS.THR: select appropriate threshold (maximum period length required. If period is longer than this value then an event will be triggered).

Example 5 - Diagnosis of Command and Feedback - acceptable propagation window and/or signal consistency check

LAMCFG.IVW: 0x0 ; don’t invert window, capture events when the counter is **equal or** above the threshold.

LAMCFG.THR: set to max delay allowed (if the delay between corresponding edges of reference and monitor signals is longer than this value, the event will be triggered).

Example 6 - Diagnosis of Set-up and Hold times

- Example settings for LAM block registers for Set-up

LAMCFG.IVW: 0x0 ; don’t invert window, capture events when the counter is **equal or** above the threshold.

- Example settings for LAM block registers for Hold

LAMCFG.IVR: **0x1** ; invert reference signal (use for gating).

LAMCFG.THR: Acceptable Hold (ref Threshold 2 on waveforms shown, changes in monitor signal will generate an alarm if they occur inside the “THR” cycles after a falling edge in the reference signal).

4.88 [IOM_TC.H004] Behavior of LAMEWCn.CNT when LAMEWSn.THR is 0

Description

When LAMEWSn.THR is set to 0, no event will be sent from the Logic Analyzer Module (LAM) to the Event Combiner Module (ECM) and no ALARM towards the SMU will be generated.

The rest of the effects derived from the cause generating the event inside the LAM will be maintained, for instance copying the counter to LAMEWCn.CNT (this means LAMEWCn.CNT also may change when LAMEWSn.THR is 0).

4.89 [IOM_TC.H006] ACCEN* Protection for Write Access to IOM Registers

Description

The access protection symbol ‘P’ to indicate protection by the ACCEN* register mechanism is missing in column “Access Mode - Write” in table “Register Overview” in the User’s Manual for IOM registers with an offset address ≥ 30_H. Actually, these registers have write access attributes ‘U,SV,P’.

Exception

In this design step, a write access to register LAMEWCm will result in a bus error, as correctly reflected by symbol ‘BE’ in column “Access Mode - Write” in table “Register Overview” in the User’s Manual.

4.90 [IOM_TC.H007] Write Access to FPCESR

Description

The Filter and Prescaler Edge Status Register FPCESR stores the state of detected rising and falling edges from each of the Filter and Prescaler Channels k (k = 0..15).

The flags in this register can be selectively cleared by writing a 0 in the respective bit-field.

However, writing to register FPCESR with a sub-word granularity (for example byte or half-word) leads to undefined behavior.

Recommendation

Individual bits for channel k in FPCESR are cleared with a write to the control register (FPCCTRk) or timer register (FPCTIMk).

Writing to FPCESR directly shall be done always to the whole register (32-bit writes), with bits that should not be modified set to 1_B. In particular, LDMST or SWAPMSK.W should be used only with bit mask enabled for all ´rwh´ bits in register FPCESR.

4.91 [LBIST_TC.H004] Update reset behavior of LBISTCTRL2 register - Additional information

Description

Even though the LBISTCTRL2.[31:0] register bits are cleared by a power-on reset they will automatically recover their values from stored contents of the central LBIST controller in the TCU (Test Control Unit) afterwards.

So on first software access the user will never see the initial reset values, but the updated LBIST done status and MISR result from the TCU LBIST controller.

The stored LBIST done status and MISR result in the central TCU LBIST controller will be cleared only through an externally applied warm power-on reset or during any cold power-on reset (triggered from EVR voltage monitors).

4.92 [LMU_TC.H002] On-the-fly BBB:SRI clock ratio switching

Description

***Note**: This problem only occurs in an ADAS or Emulation Device (ED), but may already need to be considered during software development for the target device.*

When switching the clock ratio for f_{BBB} relative to f_{SRI}, make sure that no MMES (Memory Mapped Emulation System) access to EMEM is performed by an SRI master via the LMU. Otherwise, data read/written may be incorrect.

Recommendation

After a MMES read is complete, allow at least 12 SRI clock cycles before initiating a clock ratio change.

After a MMES write is complete, allow at least 20 SRI clock cycles before initiating a clock ratio change.

After a clock ratio change, allow the clock ratio change to become effective before performing any MMES transfer (for example read back control register that was written for the clock ratio change).

4.93 [LMU_TC.H003] Function of Bit MEMCON.PMIC (Protection Bit for Memory Integrity Control Bit)

Description

In the LMU chapter of the User’s Manual, the following text (last paragraph in section “Local Memory (LMU SRAM)”) is incorrect: Some bit-fields of the LMU_MEMCON register are protected by LMU_MEMCON.PMIC bit. If the data written to the register has the bit-field set to 0_B, no change will be made to bits 15_D to 9_D of the register regardless of the data written to these fields.

Correct Description

For the correct description (only bit 9 (ERRDIS) is protected) see the description of bit PMIC in the LMU Memory Control Register in section “LMU Registers”, copied in Table 40 below:

Table 40 Bit PMIC in Register LMU_MEMCON

Field	Bit	Type	Description
PMIC	8	w	**Protection Bit for Memory Integrity Control Bit** Will always return 0_B when read 0_B Bit Protection: Bit 9 remains unchanged after LMU_MEMCON write. 1_B Bit 9 will be updated by the current write to LMU_MEMCON
ERRDIS	9	rw	**ECC Error Disable** When set SRI bus errors caused by ECC errors in data read from the SRAM will be disabled. …

4.94 [MTU_TC.H003] AURIX™ Memory Tests using the MTU

Description

The use of destructive tests such as March-U and Checkerboard etc. in conjunction with FAILDMP mode to get detailed failure information (errors, fail addresses) will cause the SRAM redundancy information to be overwritten.

Therefore, the MTU/MBIST module effectively only supports the Non-Destructive Inversion Test (NDIT).

Recommendation

To avoid overwriting the SRAM redundancy information, only use Non-Destructive Inversion Test. In this case, failure is detected by ECC and the detailed information can be obtained from ETRR and ECCD registers.

Refer to the latest version of Application Note AP32197 “AURIX™ Memory Tests using the MTU” for more details on MTU/MBIST usage and fault coverage.

4.95 [MTU_TC.H004] Handling the Error Tracking Registers ETRR

Description

CPU and on-chip peripheral SRAMs are capable of detecting errors and generating SMU alarms for correctable, uncorrectable, and address errors. The failing addresses are stored in Error Tracking Registers (ETRR), and the corresponding indicator (CERR/UERR/AERR and SERR) and valid bits (VAL) are set in the Memory ECC Detection Register (ECCD). Only new errors will be considered; i.e. errors at already stored addresses will be ignored. In case the maximum number of ETRR for a memory is used up and a new error occurs, the error overflow bit ECCD.EOV is set, and the corresponding “address buffer overflow” SMU alarm is generated. For peripheral SRAMs, the second error will cause a buffer overflow, and for CPU SRAMs, up to five errors can be registered before the buffer overflow alarm is triggered.

Bit ECCD.TRC (Tracking Clear) allows to clear the EOV and VAL bits in register ECCD and the associated ETRR registers, for example in response to a tolerated corrected single bit error.

Corner Case

If in an exceptional corner case software would set TRC at the same time an error overflow occurs, then the EOV bit is not set, and the SMU alarm is not generated.

Recommendation

- It is not necessary to clear the Error Tracking Registers ETRR by software as part of an SRAM error handling concept. For correctable errors, the application software should only react on the address buffer overflow alarm (for example with a reset). Single correctable error events may be ignored (within limits) to increase the fault tolerance of the system without impacting the safety
- If a different concept is used requiring clearing of the ETRR registers by software via ECCD.TRC, make sure that the corresponding SRAM instance is not functionally accessed while the application software writes ECCD.TRC, so that an overflow error cannot be generated during the clear operation

Information on using the MTU for memory diagnosis is given in Application Note AP32197 “AURIX™ Memory Tests using the MTU”.

4.96 [MTU_TC.H005] Handling SRAM Alarms

Description

Alarms are generated for CPU and on-chip peripheral SRAMs when correctable, uncorrectable, and address errors are detected.

The failing addresses are stored in Error Tracking Registers (ETRR), and information on the error type is stored in the Memory ECC Detection Register (ECCD). Only new errors will be considered, i.e. errors at already stored addresses will be ignored. In case the maximum number of ETRR for a memory is used up and a new error occurs, the error overflow bit ECCD.EOV is set, and the corresponding “address buffer overflow” SMU alarm is generated.

For peripheral SRAMs, the second error will cause a buffer overflow, and for CPU SRAMs, up to five errors can be registered before the buffer overflow alarm is triggered.

In addition, traps and bus errors are generated for uncorrectable errors, depending on the bus master and type of access.

Corner Case

If in an exceptional corner case

- two errors at different locations are present in the same SRAM
- and accesses are made to both locations within a time window of ~ 10 CPU clock cycles

then the first access to the location with an error will correctly trigger an SMU alarm, while the second access to the other location with an error will not trigger an SMU alarm. In the worst case, a correctable error may thus mask an uncorrectable or address error.

***Note**: In case the second error would result in an address buffer overflow, the corresponding bit ECCD.EOV is set and the “address buffer overflow” SMU alarm is correctly generated. Therefore, this problem is **not** relevant for peripheral SRAMs that only have one ETRR, as the second error will always cause an SMU alarm.*

Recommendations

- As recommended in Application Hint MTU_TC.H004 (Handling the Error Tracking Registers ETRR), for correctable errors, the application software should only react on the address buffer overflow alarm (for example with a reset). Single correctable error events may be ignored (within limits) to increase the fault tolerance of the system without impacting the safety
- In case an uncorrectable error for a CPU SRAM would neither generate an “address buffer overflow” nor an “uncorrectable” or “address error” SMU alarm, the error handling (typically resulting in a reset) should be performed in the corresponding trap routine
- In particular for EMEM or FFT SRAMs used in Emulation, ADAS or Extended SRAM devices of the AURIX™ family, a workaround is possible by triggering a correctable error before application startup. This would result in the ECCD.CERR bit of the corresponding MBIST to be set. Any future correctable alarms will not be forwarded[2]and this issue can be avoided completely

4.97 [MTU_TC.H006] Alarm Propagation to SMU via Error Flags in MCx_ECCD

Description

Upon any correctable, un-correctable or address error alarm in an SRAM, the corresponding error flags (CERR, UERR or AERR bits) in the MCx_ECCD register are set, and the corresponding alarm is forwarded to the SMU.

However, in case these bits are set to 1_B, and a further error of the same type occurs, then the corresponding alarm is no longer forwarded to the SMU.

If in a corner case software writes to Mx_ECCD in the same cycle where an error event would set one of the CERR, UERR or AERR bits from 0_B to 1_B, the software write has priority and the status flags remain at 0_B. In this case, however, the alarm is correctly propagated to the SMU.

***Note**: This behavior does not endanger the concept recommended in Application Hints MTU_TC.H004 and MTU_TC.H005 (ignore correctable errors, react on first uncorrectable/address error/buffer overflow alarm).*

[2] see MTU_TC.H006 (Alarm Propagation to SMU via Error Flags in MCx_ECCD)

Recommendation

Upon any alarm from an SRAM/MBIST, if a further alarm of the same type is required to be sent to the SMU and processed, then the software shall clear the error flag (CERR, UERR, AERR) in the ECCD register.

The flags can be cleared by writing MCx_ECCD.CERR (or UERR or AERR, respectively) with 0_B.

4.98 [MTU_TC.H008] Memory Controllers for DSPR

Description

TC23x

In **TC23x** devices, due to its implementation, the Data Scratch Pad RAM (DSPR) of CPU0 has two Memory Controller instances, described as MC14 (MC_CPU0_DSPR) and MC27 (MC_CPU0_DSPR2) in the MTU chapter of the User's Manual.

TC29x

In **TC29x** devices, due to their implementation, the Data Scratch Pad RAMs (DSPR) of CPU1 and CPU2 each have two Memory Controller instances, described in the MTU chapter of the User's Manual as:

- MC6 (MC_CPU1 TC16P_DSPR), MC20 (MC_CPU1 TC16P_DSPR2) and
- MC0 (MC_CPU2 TC16P_DSPR), MC21 (MC_CPU2 TC16P_DSPR2)

Each Memory Controller covers one half of the SRAM. In order to fully test the DSPR, the test[3] has to be executed once on each Memory Controller (i.e. only one of the two Memory Controllers is enabled at a time).

As both Memory Controllers share the same ECC decoders, any error detected by a test executed on one of the Memory Controllers will be logged in the Error Tracking Registers of both Memory Controllers.

Note that once an error status bit is set, further alarms of the same type are not forwarded to the SMU until the flag is cleared (see MTU_TC.H006 "Alarm Propagation to SMU via Error Flags in MCx_ECCD").

Recommendation

- Enable only one of the two Memory Controllers at a time
- Before executing a test on CPUx_DSPR (respectively CPUx_DSPR2), clear the Error Tracking Registers and the error status bits of CPUx_DSPR (respectively CPUx_DSPR2), so that the test reflects the results of only the one memory which is being tested
- It is also recommended to clear the Error Tracking Registers and the error status bits of both CPUx_DSPR and CPUx_DSPR2 after executing each test
- Alternatively, before executing a test on CPUx_DSPR (respectively CPUx_DSPR2), disable the error notifications in CPUx_DSPR2 (respectively CPUx_DSPR) and reenable them after the test. It is also recommended to clear the Error Tracking Registers and the error status bits of CPUx_DSPR (respectively CPUx_DSPR2) after executing a test on it

Regarding configuration and use of the two Memory Controllers, see also the latest version of Application Note AP32197 "AURIX™ Memory Tests using the MTU".

4.99 [MTU_TC.H009] Reset Value for Register ECCD

Description

The reset value of the ECC Detection Register ECCD is documented as 7800_H in the User's Manual. This is always the case for the SRAMs listed in the tables below (if available in the corresponding product).

For other SRAMs the ECCD reset value may either be $7C00_H$ or 7800_H.

[3] Test in this context means Non-Destructive Inversion Test (NDIT, see also MTU_TC.H003).

Bit ECCD.10 is marked as ‘Reserved’ in the User’s Manual:

- When writing to ECCD, bit ECCD.10 should be written as 0_B
- When reading register ECCD, bit ECCD.10 should not be evaluated. Memory errors will be reported by the notification bits CERR, UERR, AERR and EOV in register ECCD

TC29x SRAMs with ECCD Reset Value = 7800_H

Table 41 TC29x SRAMs with ECCD Reset Value = 7800_H

Memory Controller No.	Associated SRAM	Comments / Memory available in
2	CPU2 TC16P_DTAG	
5	CPU2 TC16P_PTAG	
8	CPU1 TC16P_DTAG	
11	CPU1 TC16P_PTAG	
17	CPU0 PTAG	
19	CPU0 TC16P_DTAG	
30	GTM MCS1	
31	GTM DPLL RAM1A	
32	GTM DPLL RAM1B	
34	PSI5	
38	ERAY0 OBF	
39	ERAY0 IBF_TBF	
80	CIF1	ADAS products only
81	CIF2	ADAS products only
83	DMA	

TC27x SRAMs with ECCD Reset Value = 7800_H

Table 42 TC27x SRAMs with ECCD Reset Value = 7800_H

Memory Controller No.	Associated SRAM	Comments / Memory available in
2	CPU2 TC16P_DTAG	
5	CPU2 TC16P_PTAG	
8	CPU1 TC16P_DTAG	
11	CPU1 TC16P_PTAG	
17	CPU0 PTAG	
30	GTM MCS1	
31	GTM DPLL RAM1A	
32	GTM DPLL RAM1B	
34	PSI5	
38	ERAY0 OBF	

(table continues...)

Table 42 (continued) TC27x SRAMs with ECCD Reset Value = 7800_H

Memory Controller No.	Associated SRAM	Comments / Memory available in
39	ERAY0 IBF_TBF	
80	CIF1	ADAS products only
81	CIF2	ADAS products only
83	DMA	

TC26x SRAMs with ECCD Reset Value = 7800_H

Table 43 TC26x SRAMs with ECCD Reset Value = 7800_H

Memory Controller No.	Associated SRAM	Comments / Memory available in
8	CPU1 TC16P_DTAG	
11	CPU1 TC16P_PTAG	
17	CPU0 PTAG	
30	GTM MCS1	
31	GTM DPLL RAM1A	
32	GTM DPLL RAM1B	
34	PSI5	
38	ERAY0 OBF	
39	ERAY0 IBF_TBF	
80	CIF1	ADAS products only
81	CIF2	ADAS products only
83	DMA	

TC23x SRAMs with ECCD Reset Value = 7800_H

Table 44 TC23x SRAMs with ECCD Reset Value = 7800_H

Memory Controller No.	Associated SRAM
17	CPU0 PTAG
38	ERAY0 OBF
39	ERAY0 IBF_TBF

TC22x/TC21x SRAMs with ECCD Reset Value = 7800_H

Table 45 TC22x/TC21x SRAMs with ECCD Reset Value = 7800_H

Memory Controller No.	Associated SRAM
17	CPU0 PTAG

4.100 [MTU_TC.H010] Register MCONTROL - Bit Field Res4

Description

The position of the 3-bit-field Res4 within register MCONTROL is incorrectly described as [14:10] in the register description of the User's Manual.

Documentation correction

The correct position of the 3-bit-field Res4 is MCONTROL.[14:12], as shown in the register image in the User's Manual, and in the following Table 46:

Table 46 Register MCONTROL - Position of Bit Field Res4

Field	Bits	Type	Description
Res	15	r	**Reserved** Read returns 0_B, should be written with 0_B
Res4	14:12	**rw**	**Reserved** Read returns 0x4 Must always be written with 0x4
Res	11:10	r	**Reserved** Read returns 00_B, should be written with 00_B

4.101 [MTU_TC.H011] Access Protection for Memory Control Registers

Description

The access protection symbol 'P' to indicate Access Enable Register protection is missing in column "Access Mode - Write" in table "Register Overview of each MTU Memory Control register block" of the MTU chapter in the User's Manual.

The MTU Memory Control register block actually has protection via the Access Enable registers (ACCEN0/1).

4.102 [MTU_TC.H012] Kernel Reset triggers Reset of MBIST Registers

Description

When a kernel reset is executed (via bit RST in registers KRST0/1) for a module equipped with Memory Controllers (MC) for its internal RAMs, also the corresponding MTU Memory Control (MBIST) registers are reset.

Recommendation

If required, analyze/save the contents of the MBIST registers before executing a kernel reset.

After a kernel reset, reconfigure the MBIST registers.

4.103 [MTU_TC.H014] Access to SRAM while MTU operations are underway

Description

When MTU operations on the SRAM are underway, the memories cannot be accessed. MTU operations in this context include:

1. Running an MBIST test (for example Non-destructive test)
2. Performing an SRAM initialization using the MTU
3. When an Auto-data-initialization is underway

During these operations, the SRAM shall not be accessed. If the SRAM is accessed during this time, unexpected behavior may occur (for example access timeout).

Cases 1. and 2. are easily identified, i.e. whenever the application has triggered an MBIST test or SRAM initialization.

Case 3. occurs whenever bit-field PROCOND.RAMIN is not equal to 0x3. Whenever this is the case in specific MBIST controllers, the SRAM is fully or partially cleared under certain conditions:

- When MTU_MEMTEST.*EN bit is enabled or disabled
- When MTU_MEMMAP.*MAP bit is set or cleared (applicable only to cache memories)

This means, when the above mentioned bits are set or cleared, it takes some time (~hundreds of clock cycles) for the associated SRAMs to be (fully or partially) initialized. During this time the SRAM is not accessible.

Affected SRAMs are:

- CPUx DMEM (DSPR+DCACHE)
- CPUx PMEM (PSPR + PCACHE)

Recommendation

- For all memories, ensure that the SRAM is not accessed when any MTU operation is underway
- For the specific memories listed above, ensure that the SRAM is not accessed:
  - When setting MTU_MEMTEST.*EN bit: as long as MEMSTAT.*AIU bit is set or as long as the MEMTEST.*EN bit is not yet set
  - When clearing MTU_MEMTEST.*EN bit: as long as MEMSTAT.*AIU bit is set or as long as the MEMTEST.*EN bit is not yet cleared
  - When setting or clearing MTU_MEMMAP.*MAP bit for DMEM/PMEM: as long as MEMSTAT.*AIU bit is set

4.104 [MultiCAN_AI.H005] TxD Pulse upon short disable request

Description

If a CAN disable request is set and then canceled in a very short time (one bit time or less) then a dominant transmit pulse may be generated by MultiCAN module, even if the CAN bus is in the idle state.

Example for setup of the CAN disable request:

CAN_CLC.DISR = 1 and then CAN_CLC.DISR = 0

Workaround

Set all INIT bits to 1 before requesting module disable.

4.105 [MultiCAN_AI.H006] Time stamp influenced by resynchronization

Description

The time stamp measurement feature is not based on an absolute time measurement, but on actual CAN bit times which are subject to the CAN resynchronization during CAN bus operation.The time stamp value merely indicates the number of elapsed actual bit times. Those actual bit times can be shorter or longer than nominal bit time length due to the CAN resynchronization events.

Workaround

None.

4.106 [MultiCAN_AI.H007] Alert Interrupt Behavior in case of Bus-Off

Description

The MultiCAN module shows the following behavior in case of a bus-off status:

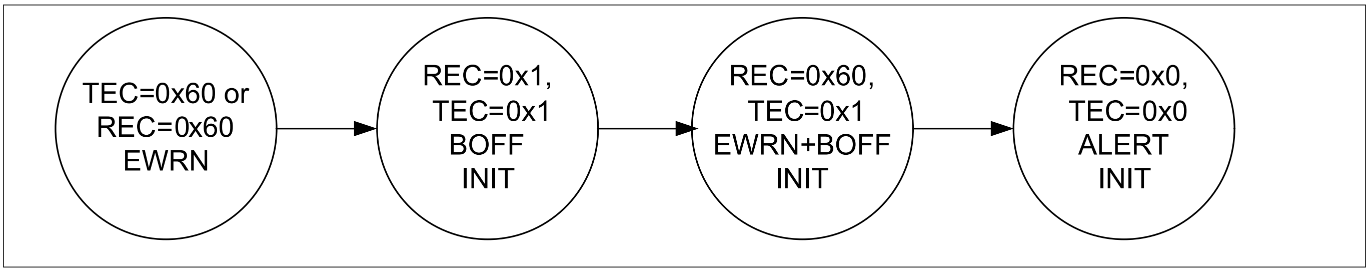


Figure 8 Alert Interrupt Behavior in case of Bus-Off

When the threshold for error warning (EWRN) is reached (default value of Error Warning Level EWRN = 0x60), then the EWRN interrupt is issued. The bus-off (BOFF) status is reached if TEC > 255 according to CAN specification, changing the MultiCAN module with REC and TEC to the same value 0x1, setting the INIT bit to 1_B, and issuing the BOFF interrupt. The bus-off recovery phase starts automatically. Every time an idle time is seen, REC is incremented. If REC = 0x60, a combined status EWRN+BOFF is reached. The corresponding interrupt can also be seen as a pre-warning interrupt, that the bus-off recovery phase will be finished soon. When the bus-off recovery phase has finished (128 times idle time have been seen on the bus), EWRN and BOFF are cleared, the ALERT interrupt bit is set and the INIT bit is still set.

4.107 [MultiCAN_TC.H003] Message may be discarded before transmission in STT mode

Description

If MOFCRn.STT =1 (Single Transmit Trial enabled), bit TXRQ is cleared (TXRQ=0) as soon as the message object has been selected for transmission and, in case of error, no retransmission takes places.

Therefore, if the error occurs between the selection for transmission and the real start of frame transmission, the message is actually never sent.

Workaround

In case the transmission shall be guaranteed, it is not suitable to use the STT mode. In this case, MOFCRn.STT shall be 0.

4.108 [MultiCAN_TC.H004] Double remote request

Description

Assume the following scenario: A first remote frame (dedicated to a message object) has been received. It performs a transmit setup (TXRQ is set) with clearing NEWDAT. MultiCAN starts to send the receiver message object (data frame), but loses arbitration against a second remote request received by the same message object as the first one (NEWDAT will be set).

When the appropriate message object (data frame) triggered by the first remote frame wins the arbitration, it will be sent out and NEWDAT is not reset. This leads to an additional data frame, that will be sent by this message object (clearing NEWDAT).

There will, however, not be more data frames than there are corresponding remote requests.

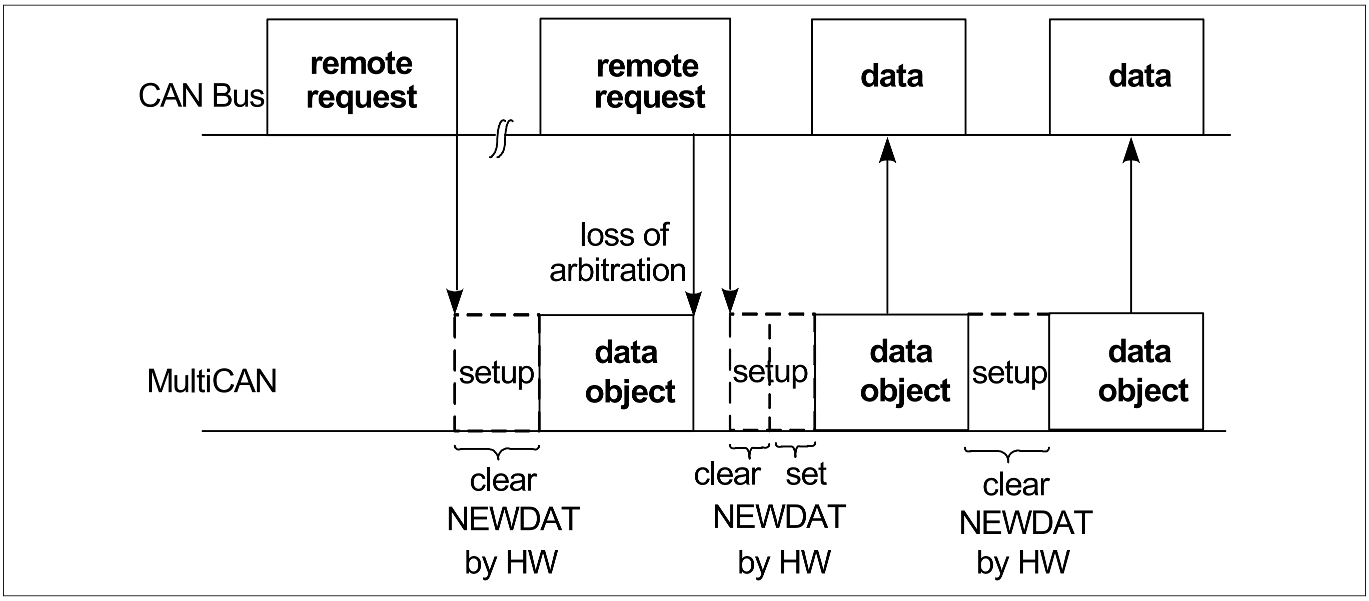


Figure 9 Loss of Arbitration

4.109 [MultiCAN_TC.H007] Oscillating CAN Bus may Disable the CAN Interface

Description

If the connected CAN network is in an unspecified oscillating state for more than 512 cycles this can result in disabling the CAN interface of the device. Enabling the CAN interface again requires then a Power-on Reset.

Recommendation

Please refer to application note AP32264 “DXCPL DAP over CAN Physical Layer” for further information and how this situation can be prevented.

4.110 [MultiCAN_TC.H008] Changes due to CAN FD protocol ISO 11898-1:2015

Description

***Note**: This Application Hint might affect the SFR C Header Definitions. In such cases, SFR usage in the software shall be analyzed within the applications for their correct handling.*

Specific variants and device steps of the AURIX™ TC2xx family support the CAN FD frame format according to standard version ISO 11898-1:2015. These variants are identified by the feature type code **'N'** as last letter in the device name, for example SAL-TC299TP-128F300**N** or SAK-TC223L-16F133**N**.

***Note**: In **TC29x** variants with feature type code 'N', MultiCAN nodes 0..3 support this feature, while MultiCANR nodes don’t.*

*In **TC27x** variants with feature type code 'N', all MultiCAN nodes (0..3) support this feature.*

*In **TC26x** variants with feature type code 'N', MultiCAN nodes 0..3 support this feature, while node 4 does not.*

*In **TC23x** variants with feature type code 'N', nodes 0 and 1 in MultiCAN and MultiCAN1 support this feature, while nodes 2 don’t.*

*In **TC22x/TC21x** variants with feature type code 'N', all MultiCAN nodes (0..2) support this feature.*

For a summary of the devices and nodes supporting this feature see Table 50 at the end of this text module.

For availability of the variants with this feature see the corresponding “AURIX™ TC2xx Variants / Data Sheet Addendum”.

Detailed Description

ISO 11898-1:2015 improves the failure detection capabilities of the ISO11898-1 DIS version 2014. Information about the number of stuff bits in the data field is added to the CRC field. These added bits are called **'Stuff Count'**.

The Stuff Count contains 4 bits, including

- 3 bits gray code to represent the modulo-8 of number of stuff bits in the data field
- and 1 bit for the parity

Since the Stuff Count bits are part of the CRC field, fixed stuff bits will be added before and after the Stuff Count bits. Figure 10 and Figure 11 show the frame format of the ISO 11898-1:2015 CAN FD protocol. There is no change in the classical CAN frame format.

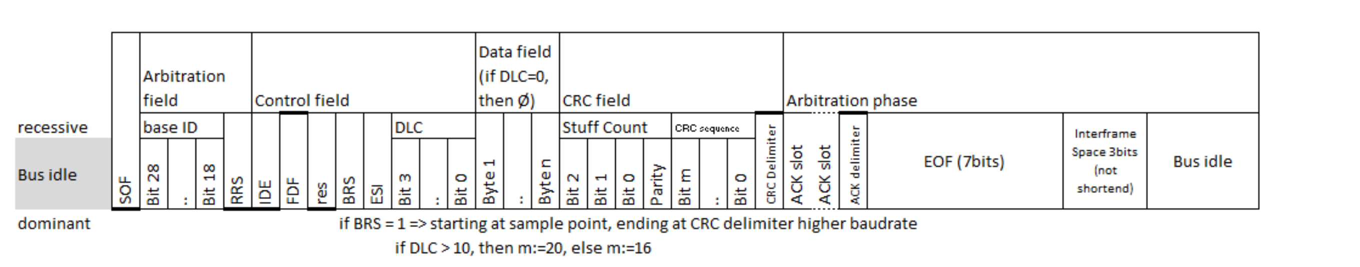


Figure 10 **ISO CAN FD 11-bit ID Data Frames**

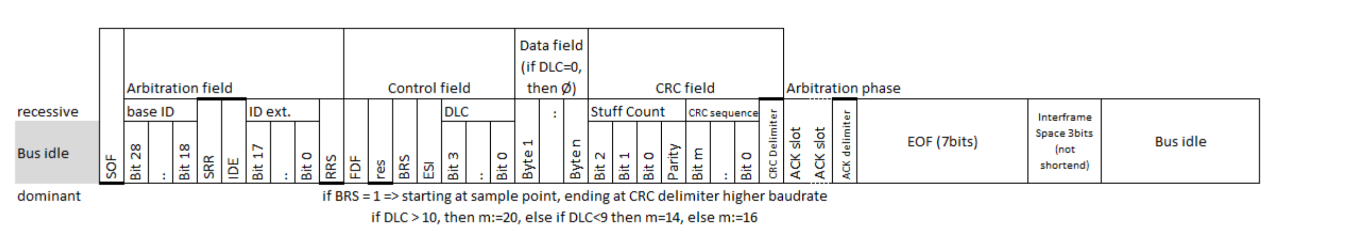


Figure 11 **ISO CAN FD 29-bit ID Data Frames**

From here on,

- the ISO 11898-1:2015 frame format will be referred to as **ISO** CAN FD format
- the previous frame format will be referred to as **Non-ISO** CAN FD format

***Note**: The ISO CAN FD frame format is incompatible with Non-ISO CAN FD frame format.*

AURIX™ devices (with feature type code 'N') support both ISO and Non-ISO CAN FD formats. The format can be selected by modified functionality of bits NBTR0.15 and NBTR1.15:

Functionality of Bit NBTR0.15

NBTR0.15 is changed from NBTR0.DIV8 (Divide Prescaler Clock by 8) to **NBTR0.NISO**[4] (Non-ISO operation) as shown in Table 47:

[4] The symbolic names NISO and PED are only used for explanation in this context. If desired, the register definition file could be modified.

Table 47 Functionality of Bit NBTR0.15

Field	Bit	Type	Description
NISO	15	rw	**Non-ISO Operation** If this bit is set, the MultiCAN+ uses the non-ISO CAN FD frame format. This bit is CCE protected. 0_B CAN FD frame format according to ISO 11898-1:2015 (default after reset) 1_B CAN FD frame format non-ISO.

Functionality of Bit NBTR1.15

NBTR1.15 is changed from NBTR1.DIV8 (Divide Prescaler Clock by 8) to **NBTR1.PED**[4] (Protocol Exception Disable) as shown in Table 48:

Table 48 Functionality of Bit NBTR1.15

Field	Bit	Type	Description
PED	15	rw	**Protocol Exception Disable** The protocol exception event is described in the ISO 11898-1:2015 as option. The error frame on the res bit can be controlled with this option. This bit is CCE protected. 0_B Protocol Exception Event is enabled (default after reset). 1_B Protocol Exception Event is disabled.

Note: *Both NBTR0.NISO and NBTR1.PED are global register bits. This means they affect all the ISO 11898-1:2015 compliant CAN FD nodes in the respective MultiCAN+ module. The former DIV8 function of nodes 0 and 1 is hard-wired to 0_B (i.e. a time quantum lasts (BRP+1) clock cycles).*

The DIV8 function (Divide Prescaler Clock by 8) for all other nodes x (x>1) remains the same, irrespective of the setting of NBTR0.NISO and NBTR1.PED.

Table 49 describes the CAN FD behavior for different configurations of the NBTR0.NISO and NBTR1.PED bits. By default, the CAN FD behaves in compliance with ISO 11898-1:2015 if CAN FD is enabled (bit FDEN = 1_B for corresponding node).

Table 49 Configurations of PED and NISO

PED	NISO	CAN FD Enabled
0	0	Default values - ISO 11898-1:2015 CAN FD compliant
0	1	Non-ISO CAN FD format - same behavior as previous AURIX™ devices
1	0	CAN FD with protocol exception event disabled - ISO 11898-1:2015 CAN FD compliant
1	1	Reserved

Note: *Nodes where FDEN = 0_B will operate using the classical CAN frame format.*

Summary of Devices and Nodes supporting ISO CAN FD

The following table summarizes the nodes of devices with feature type code 'N' which have the ISO 11898-1:2015 CAN FD functionality.

[4] The symbolic names NISO and PED are only used for explanation in this context. If desired, the register definition file could be modified.

Table 50 AURIX™ TC2xx Devices and Nodes supporting ISO CAN FD

Device / Step	ISO CAN FD supporting nodes	Non-ISO CAN FD supporting nodes
TC29x ≥ BC	MultiCAN - Nodes 0,1,2,3	MultiCANR: Nodes 0, 1
TC27x ≥ DC	MultiCAN - Nodes 0,1,2,3	all nodes support ISO CAN FD
TC26x ≥ BC	MultiCAN - Nodes 0,1,2,3	MultiCAN - Node 4
TC23x[1] ≥ AC	MultiCAN - Nodes 0,1 MultiCAN1 - Nodes 0,1	MultiCAN - Node 2 MultiCAN1 - Node 2
TC22x ≥ AC	MultiCAN - Nodes 0,1, 2	all nodes support ISO CAN FD
TC21x ≥ AC	MultiCAN - Nodes 0,1, 2	all nodes support ISO CAN FD

[1] TC23x Emulation (ED) and ADAS Devices only support Non-ISO CAN FD

4.111 [MultiCAN_TC.H009] Limitation on Secondary Sample Point (SSP) Position (ISO CAN FD nodes only)

Description

***Note**: This Application Hint only applies to ISO CAN FD nodes. For devices and nodes supporting the ISO CAN FD format, see MultiCAN_TC.H008.*

The MultiCAN+ of AURIX™ TC2xx has passed the ISO/DIS 16845-1(E), 2015 CAN Conformance test performed by an external test house C&S group GmbH and the test reports are available. The limitation on the range of SSP position is described in the Conformance test report.

In AURIX™ TC2xx devices, there are two limitations with the Secondary Sample Point (SSP) position for CAN FD with respect to ISO 11898-1, 2015 specification:

1. Granularity of the Transmitter loop delay measurement (only when CAN_FNBTRx.FBRP = 1)

Limitation

The Transmitter loop delay measurement is based on data-phase time quantum ($t_{q(D)}$) and not by minimum time quanta (mtq) or CAN clock period as specified in ISO 11898-1 2015. Hence the granularity of the transmitter loop delay measurement is +1 $t_{q(D)}$ in worst case scenario.

***Note**: According to ISO 11898-1 – 2015, when Transmitter Delay Compensation is enabled (CAN_NTDCR.TDC = 1), then the CAN_FNBTRx.FBRP shall be either 0 or 1.*

Effect

In worst case scenario, the SSP could be delayed by +1 $t_{q(D)}$.

Recommendation

It has to be taken care that the SSP offset (CAN_NTDCR.TDCO) is configured accordingly by including the granularity of the transmitter loop delay measurement of +1 $t_{q(D)}$ in worst case scenario.

2. Range of SSP position (only when CAN_FNBTRx.FBRP = 0)

Limitation

The Secondary Sample Point Position is limited to 31 t_q or 31 mtq (bit-field CAN_NTDCRx.TDCV), when compared to 63 mtq as required by ISO 11898-1, 2015.

Note: *When CAN_FNBTRx.FBRP = 0, then 1 time-quantum (t_q) = 1 minimum time-quantum (mtq).*

CAN FD applications with fast data baud rate greater than 2 Mbit/s require Fast Baud Rate Prescaler setting CAN_FNBTRx.FBRP = 0 and f_{CAN} at 80 MHz to ensure reliable CAN communication in long networks. In such a scenario, the max SSP position achievable by the TDC is limited to 31 t_q, i.e. 388 ns (31 * 12.5 ns).

Effect

In scenarios where the sum of transmitter loop delay and SSP offset (CAN_NTDCRx.TDCO) is more than 31 time quanta, the SSP value saturates at 31 time quanta, leading to SSP placed (at 31 time quanta) earlier than required.

Recommendation

It has to be taken care to ensure that the sum of transmitter loop delay and SSP offset (CAN_NTDCRx.TDCO) is within the limit of 31 time quanta.

4.112 [MultiCAN_TC.H010] Limitation on maximum SJW Range for CAN FD Data Phase (ISO CAN FD nodes only)

Description

Note: *This Application Hint only applies to ISO CAN FD nodes. For devices and nodes supporting the ISO CAN FD format, see MultiCAN_TC.H008.*

Note: *Register names in the text follow the MultiCAN syntax (CAN_FNBTRx, CAN_NBTEVRx). Corresponding registers in MultiCAN1 in TC23x: CAN1_FNBTRy, CAN1_NBTEVRy.*

The MultiCAN+ of AURIX™ TC2xx has passed the ISO/DIS 16845-1(E), 2015 CAN Conformance test performed by an external test house C&S group GmbH and the test reports are available.

ISO 11898-1, 2015 specifies the configuration range of the CAN FD Data phase (re-)synchronization jump width (SJW) as 1-8 $t_{q(D)}$.

In AURIX™ TC2xx devices, the CAN FD Data phase SJW is limited to 1-4 $t_{q(D)}$, as bit-field CAN_FNBTRx.FSJW is 2 bits wide.

Effect

Configuring a MultiCAN+ node for CAN FD communication with CAN FD Data Phase SJW less than required, could result in wrong sampling of the received bit of CAN FD Data Phase, thus causing a Receive Error.

Recommendation

Choose the CAN FD configuration in such a way that

- The period of time-quanta in Arbitration phase is equal to the period of time-quanta in data phase. This can be achieved by configuring CAN_NBTEVRx.BRP = CAN_FNBTRx.FBRP
- CAN_FNBTRx.FSJW = min(CAN_FNBTRx.TSEG2, 3)

By this configuration the effect of limited Data SJW range offered by MultiCAN+ on maximum oscillator tolerance required (as given by conditions described in ISO 11898-1) is minimized.

4.113 [MultiCAN_TC.H011] Transmitter Delay Compensation Behaviour (CAN FD only)

Description

When using Transmitter Delay Compensation consider the following points:

1. The transmitter delay compensation does not take the Fractional Divider into account. This means that the values of CAN_NTDCR.TDCO and CAN_NTDCR.TDCV always correspond to CAN_FDR.DM = 01_B and CAN_FDR.STEP = 1023, even though a different setting of the fractional divider is actually in place. Therefore, it is recommended to use setting DM = 01_B and STEP = 1023 in register CAN_FDR so that the granularity of the transmitter loop delay measurement is depending only on the fast baud rate prescaler (CAN_FNBTRx.FBRP)
2. If $2*f_{CAN} < f_{CLC}$, then the transmitter delay compensation measurement value of the previous measurement may be uploaded to bit-field CAN_NTDCR.TDCV instead of the measured delay of the current message, i.e. the measured delay will appear in bit-field CAN_NTDCR.TDCV with a delay of one CAN message

4.114 [MultiCAN_TC.H012] Delayed time triggered transmission of frames

Description

The value written in the bit-field RELOAD of register NTATTRx(x=0-3), NTBTTRx(x=0-3), NTCTTRx(x=0-3) represents the reload counter value for the timer used for triggered transmission of message objects (Classical CAN or CAN FD frames).

The timer source and the prescaler value is defined in the NTCCRx(x=0-3) register.

Once a value is written to bit-field RELOAD with bit STRT=1 the timer starts counting. This timer counts one value more than the written value in bit-field RELOAD, then it triggers the transmission of a message object.

Effect

The message object transmission is delayed by one counter cycle with respect to the desired count time written in bit-field RELOAD.

Recommendation

In order to transmit a message object at a specific time, when using one of these registers:

- NTATTRx(x=0-3), NTBTTRx(x=0-3), NTCTTRx(x=0-3)

set bit-field RELOAD one value less than the calculated counter value.

4.115 [OCDS_TC.H010] JTAG requires two initial clock cycles after $\overline{\text{PORST}}$

Description

For a proper selection of the chip internal TCK clock path, two TCK clock cycles are needed after $\overline{\text{PORST}}$ release. They can be executed with TMS Low or High. A following TCK clock cycle with TMS High will always bring the JTAG TAP state to Run-Test/Idle. This sequence is compliant to standard JTAG and can be used for all TriCore™ devices.

4.116 [OCDS_TC.H012] Minimum Hold Time for Inputs OCDS_TGIx

Description

Inputs OCDS_TGIx (x=0..7, depending on device/package type) may be used to trigger the On-Chip Debug System (OCDS) for example for break or interrupt from an external source.

To ensure the external trigger is sampled correctly and not missed, the trigger should be asserted for a minimum of two SPB clock cycles.

4.117 [OCDS_TC.H019] System or Application Reset while OCDS and lockstep monitoring are enabled

Description

After a System or Application Reset the Lockstep Alarm ALMx[0] gets activated if all of the following conditions are met (x = index of CPU with checker core):

1. Lockstep monitoring is enabled by BMI.LCLxLSEN = 1_B for CPUx, AND
2. Debug System is enabled (CBS_OSTATE.OEN = 1_B), AND
3. CPUx Performance Counters are enabled, AND
4. CPUx Clock Cycle Count register CCNT is read

Recommendation

To avoid the unintended ALMx[0] under the conditions described above, either:

- Keep the debug system disabled. OR
- Ensure CPUx Performance Counters are disabled for all CPUs that have lockstep monitoring enabled before executing a System or Application reset. OR
- Use PORST instead of a System or Application reset

4.118 [PACKAGE_TC.H008] Exposed pad dimensions and package outlines for QFP packages - Updates to TC23x Data Sheet

Description

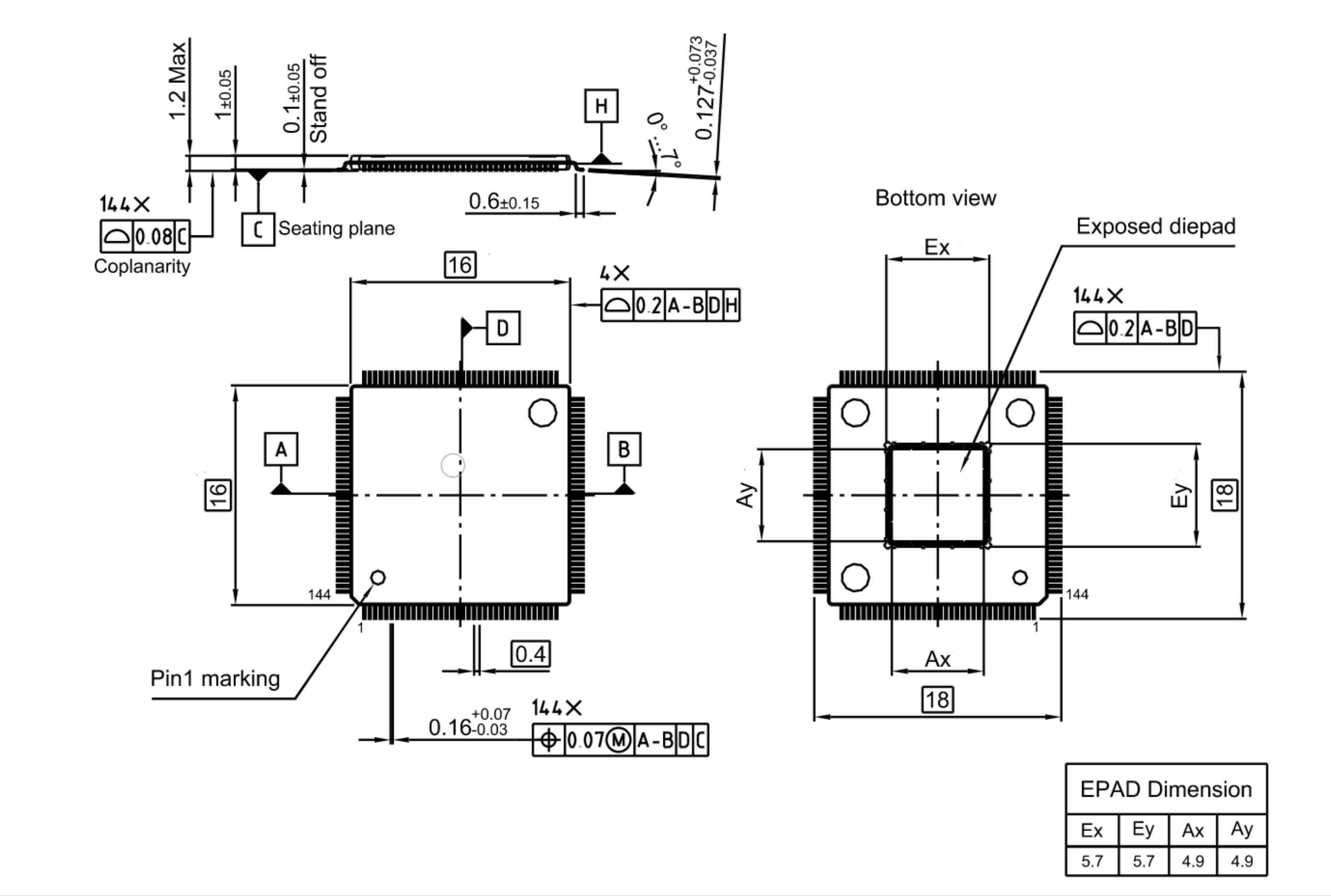
In the scope of the harmonization of the package drawings, the drawings for the TQFP packages of the TC23x have been updated. No change of form, fit or function is implied.

The dimensions for the exposed pads are included in the respective figures.

Furthermore, for the exposed pads, the maximum boundary of the structural corner protrusions to be considered during system design and integration has been added.

This information shall substitute the corresponding information in the TC23x AC-step Data Sheet V1.0 and in the TC23x A-step Data Sheet V1.1.

Package Outlines TQFP-144 for TC23x



Ex	Ey	Ax	Ay
5.7	5.7	4.9	4.9

Figure 12 **Package Outlines TQFP-144 for TC23x**

***Note**: For the exposed pad of the TQFP-144 package of the TC23x, structural corner protrusions have to be considered for purposes of system design and integration with a maximum boundary of 6.2 mm.*

Package Outlines TQFP-100 for TC23x

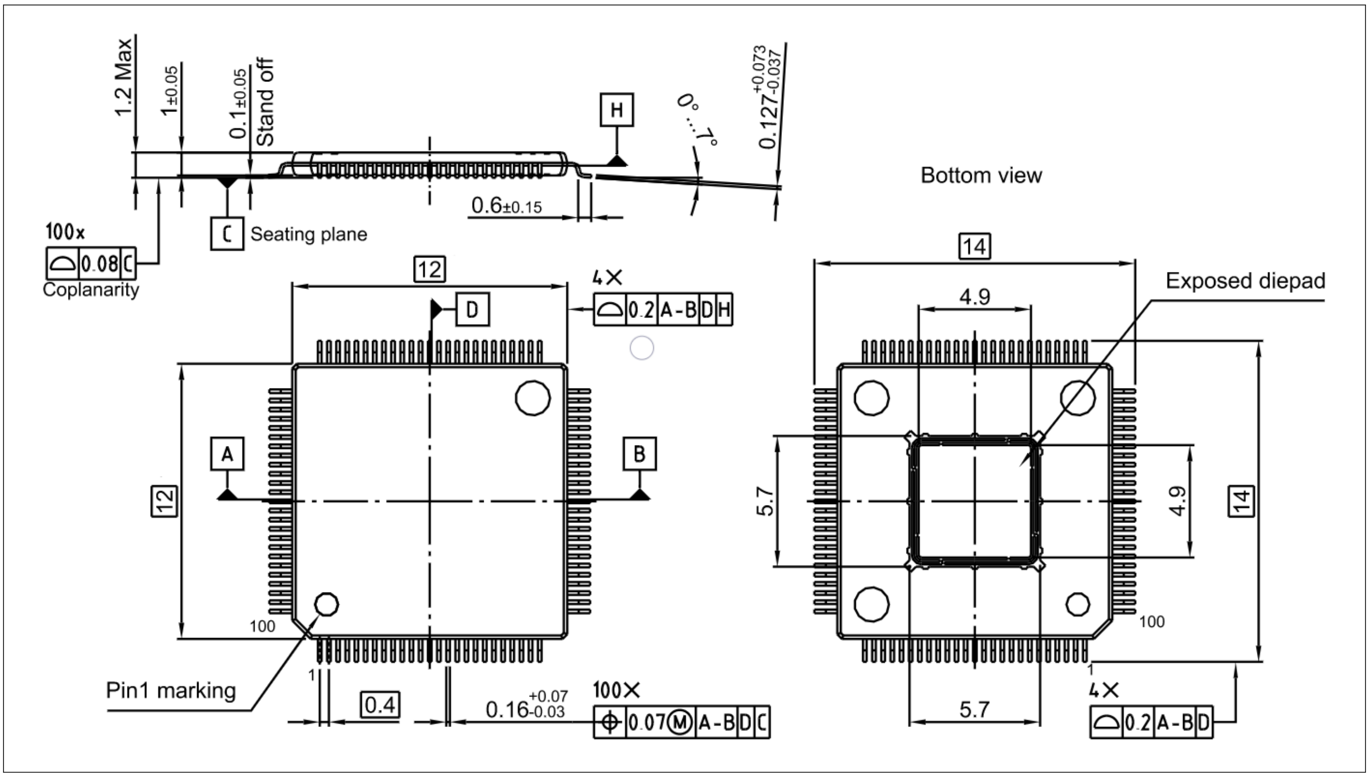


Figure 13 **Package Outlines TQFP-100 for TC23x**

***Note**: For the exposed pad of the TQFP-100 package of the TC23x, structural corner protrusions have to be considered for purposes of system design and integration with a maximum boundary of 6.3 mm.*

4.119 [PLL_ERAY_TC.H002] Correction in Figure “PLL_ERAY Block Diagram”

Description

The signal originating from block “K2-Divider” in figure “PLL_ERAY Block Diagram” in chapter “ERAY Phase-Locked Loop” of the User’s Manual is incorrectly labeled as PLLERAYSTAT.K**1**RDY.

Correction

The correct name of the signal originating from block “K2-Divider” is PLLERAYSTAT.K**2**RDY.

4.120 [PMC_TC.H001] Check for permanent Overvoltage during Power-up

Description

After an initial power-on with a permanent overvoltage condition on either V_{EXT}, V_{DDP3} or V_{DD} supply rails, no overvoltage alarm may be generated by the SMU after configuration of the alarms, as the threshold transition condition has already happened.

However, in case an overvoltage condition was present, it will be indicated by flags OV13, OV33, and OVSWD, respectively, in register EVRSTAT.

Recommendation

Check the OV13, OV33, and OVSWD flags in register EVRSTAT by software at start-up to identify an overvoltage condition.

4.121 [PMC_TC.H004] Selecting the WUT Clock Divider

Description

Wake-up timer usage with PMSWCR3.WUTDIV = 1_B (10 ms count) for PMSWCR3.WUTREL values up to 20 ms is exposed to synchronization issues. The WUT counter PMSWUTCNT.WUTCNT is counted down to 0 every 10 ms, and reloading of WUTREL happens 10 ms later. If Standby request is sent before reloading PMSWCR3.WUTREL, regardless of PMSWSTATCLR.WUTWKPCLR, wake-up request is issued without counting down. This leads to immediate wake-up.

Recommendation

Use WUT with setting PMSWCR3.WUTDIV = 1_B only for longer time periods (more than 20 ms).

For shorter periods the 10 µs clock should be used with setting PMSWCR3.WUTDIV = 0_B (default after reset).

4.122 [PMS_TC.H002] Sensitivity to supply voltage ripple during start-up

Description

The internal back-up clock is sensitive to specific power supply voltage disturbance/ripple caused by a voltage ripple intrinsic to DC-DC converters. Specific conditions such as insufficient filtering of the ripple may lead to improper behavior of the start-up scheme of the back-up clock, and thus stuck-at state during the start-up of the microcontroller until this condition is removed.

The acceptable voltage vs. frequency characteristic is portrayed below on the chart:

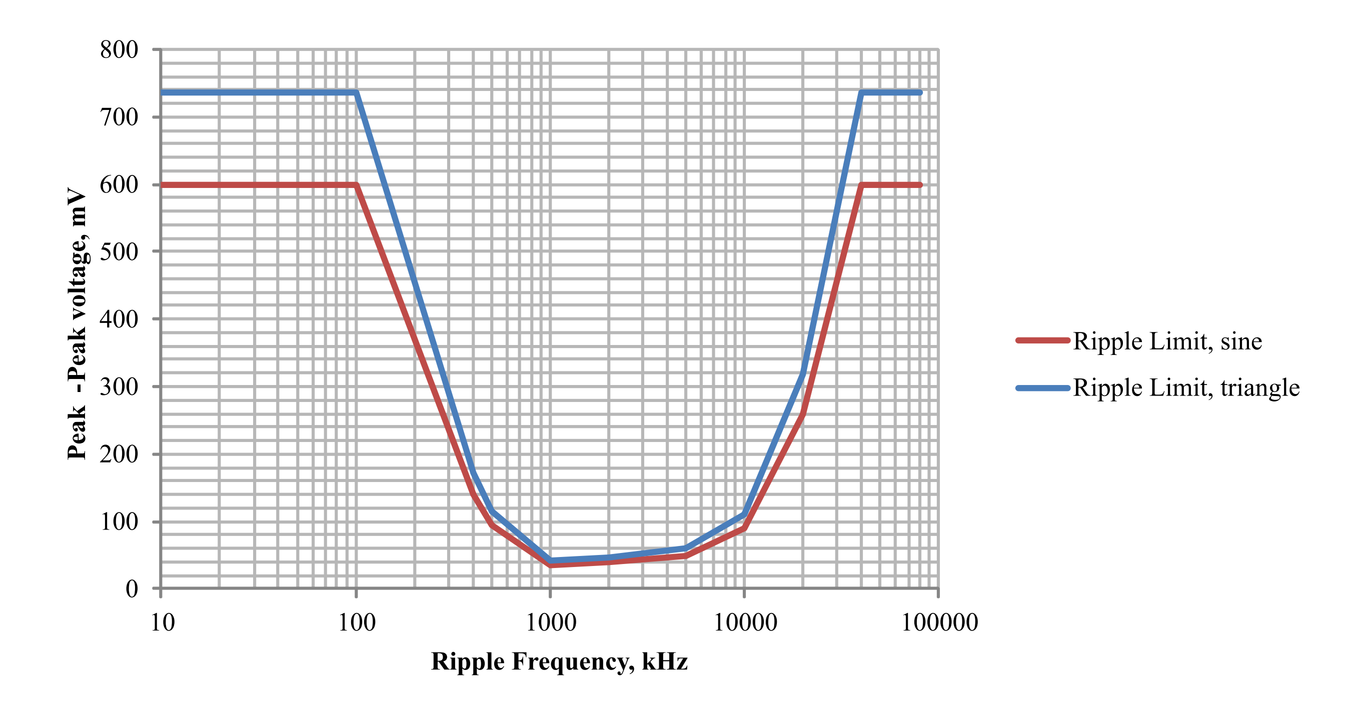


Figure 14 Ripple Voltage vs. Frequency Characteristic

The diagram reflects acceptable ripple level during the cold start of the microcontroller at the respective VDDP3/VEXT/VEVRSB supply of the PMS subsystem, depending on the device and package type, as shown in the following table.

Table 51 Pads/Pins sensitive to supply voltage ripple during start-up

Device	Package	Pad/Pin	Symbol
TC29x	BGA-516	AA16	VEVRSB
TC29x	BGA-416	AD9	VEVRSB
TC29x, TC27x, TC26x	BGA-292	T11	VEVRSB
TC27x, TC26x	QFP-176	69	VEXT
TC26x	QFP-144	59	VEXT
TC23x	BGA-292	T11	VDDP3
TC23x, TC22x, TC21x	QFP-144	69	VDDP3
TC23x, TC22x, TC21x	QFP-100	47	VDDP3
TC22x, TC21x	QFP-80	37	VDDP3

Recommendation 1

Apply an additional ceramic capacitor at the respective VDDP3/VEXT/VEVRSB supply input (at pins specified above) to attenuate the residual ripple of the buck converter. The resonant frequency of the additional filter capacitor shall be chosen in accordance with the amplitude-frequency characteristic given above and the switching frequency of the DC-DC converter in order to provide a proper attenuation in the range of interest.

The amount of ripple voltage can be approximated by $V_{\text{pk-pk}} = I_{\text{load}}$ / (f*C) and therefore the necessary nominal value of the blocking capacitance can be estimated as C = I_{load} / (f*$V_{\text{pk-pk}}$)

It is recommended to take the I_{load} value as approximately 10 mA for the start-up load at the respective VDDP3/VEXT/VEVRSB domain before the internal regulator starts.

The frequency shall be taken same as the switching frequency of the external DC-DC voltage regulator. For example:

C = (0.010 A) / (10^6 Hz * 0.040 V) = 0.25 * 10^{-6} F

Recommendation 2

Dimension the output LC filter of the external DC-DC converter to meet the limit of the ripple below the specified limit at the switching frequency. The effective value of ripple current flowing in and out of the buffer capacitor is calculated in accordance with standard formulas for the DC-DC buck converters. Selection of the low-ESR buffer capacitor is crucial in such applications, as the ESR value is directly proportional to the voltage drop caused by inductor current ripple.

Recommendation 3

Supply the respective VDDP3/VEXT/VEVRSB rail by an external post LDO power stage.

4.123 [PMS_TC.H008] Interaction of interrupt and power management system - Additional information

Description

- **TC2xx**: The description of steps to enter Idle, Sleep and Standby Mode in chapter “Power Management Overview” of the PMC chapters in the current TC2xx User’s Manuals is not comprehensive in explaining the dependency on pending interrupts as well as received interrupts. Hence, more explanation is provided here.

- **TC3xx**: The description of steps to enter Idle, Sleep and Standby Mode in chapter "Power Management Overview" of the PMS and PMSLE chapters in the current TC3xx User's Manual is not comprehensive in explaining the dependency on pending interrupts as well as received interrupts. Hence, more explanation is provided here.

For a CPU to enter Idle Mode, it must have no interrupts pending. If it is in Idle Mode it will stay in Idle Mode until one of the specified wake-up events occurs – one of these is to have a pending interrupt.

Any SRN targeting a specific CPU (i.e. TOS set to that CPU), which is enabled, i.e. has SRE set, and has received a trigger event, i.e. has SRR set (whether by a received trigger from a peripheral or a master using the SETR control bit in the SRN) is a pending interrupt. Thus, even if a peripheral is shut down by having its clocks gated off, if it has presented a trigger event to the IR, and the SRE bit for that SRN is set, there will be a pending interrupt to the specified CPU.

It is not necessary for the priority of the pending interrupt to allow it to be taken, nor is it necessary for the CPU to have interrupt servicing enabled. It is possible and valid for Idle Mode to be entered with interrupts disabled, and to only re-enable interrupt acceptance subsequent to resuming execution. Equally, the CPU's priority may well dictate that the interrupt cannot be serviced immediately on re-enabling interrupts.

There may be some interrupts in a system that a CPU will be required to service and must exit Idle Mode (or Sleep Mode) or prevent entry to Idle Mode (or Sleep or Standby Mode) on their arrival. If one of these interrupts is raised prior to, or just as Idle Mode, Sleep Mode or Standby Mode is requested then that mode will not be entered.

The description for the REQSLP field states

- "In Idle Mode or Sleep Mode, these bits are cleared in response to an interrupt for the CPU, or when bit 15 of the corresponding CPU Watchdog Timer register (bit WDTCPUxSR.TIM[15]) changes from 0 to 1."

For clarity, this also means, if a write to PMCSRx.REQSLP occurs while the IR has a pending interrupt for CPUx the write data will be ignored and the REQSLP value will remain as 00_B "Run Mode".

For the system to enter Sleep or Standby Mode by writing to PMCSRx.REQSLP (as opposed through an external low voltage condition), all CPUs must be in Idle Mode. Typically, first other CPUs will be brought into Idle Mode and then the master CPU will be the last to enter to Idle Mode as a transitional state of the request for the system mode Sleep or Standby. Consequently any pending interrupts for any CPU will prevent the entry into Sleep or Standby Mode.

Recommendation

To ensure the transition to a power save mode, for a CPU intended to enter Idle Mode or for a system entering Sleep or Standby mode, all interrupts that are not intended to cause Run Mode to be re-entered or retained, should either have the SRE bit cleared in the respective SRN or be guaranteed to have the SRR bit clear.

- **TC2xx**: If modifying the SRE bit of an SRN, to ensure the new state is reflected in IR arbitration information conveyed to the PMC and CPUs, sufficient time for an arbitration must have elapsed. Hence, a subset of the synchronisation described in subsection "Changing the SRN configuration" of the IR chapter in the corresponding TC2xx User's Manual is required.
- **TC3xx**: If modifying the SRE bit of an SRN, to ensure the new state is reflected in IR arbitration information conveyed to the PMS and CPUs, sufficient time for an arbitration must have elapsed. Hence, a subset of the synchronisation described in subsection "Changing the SRN configuration" of the IR chapter in the TC3xx User's Manual is required.

After the last SRN (for CPUx) has been updated

- Read back the last SRN
- Read the LWSRx register

Clearing the SRR bit or disabling the source of the trigger can also be used if there are no timing hazards; i.e. no risk of a trigger being raised just before reconfiguring the peripheral (to not raise triggers), or no risk of an SRN that has had SRR cleared being set again while other SRNs are accessed. If the timing behaviour of these interrupt sources allows them to be disabled at source or in the SRN these are also valid methods. So long as the SRE bit and SRR bit are not both set, there will not be a pending interrupt. If the SRR bits are cleared, after the last SRN is modified there also needs to be a synchronisation step for the IR outputs to reflect the update before the PMCSRx is written.

Once there are no pending interrupts, request the power saving mode by writing to the respective PMCSRx.

***Note**: **TC2xx**: There will still be several system clock cycles till the power saving mode is enabled by the PMC during which the CPU will continue to execute instructions.*

***Note**: **TC3xx**: There will still be several system clock cycles till the power saving mode is enabled by the PMS during which the CPU will continue to execute instructions.*

To ensure a deterministic boundary for execution to end after the power saving mode request, the write to PMCSRx should be followed by a DSYNC and a WAIT instruction.

4.124 [PMU_TC.H002] Impact of Application Reset on register FLASH0_FCON

Description

Register FLASH0_FCON is described in PMU chapter “Flash Configuration Control” as being reset by Application Reset with reset value 0091 $XXXX_H$ with a footnote adding the information

“[1] The wait-cycles WSECDF, WSDFLASH, WSECPF and WSPFLASH are changed by the startup after system and power-on resets. **Attention: the configured value is only sufficient for the clock configuration used during startup**. The wait-cycles have to be configured after startup as described in <reference to the PMU section “Configuring Flash Wait Cycles”> before changing to higher clock frequencies.”

In this section the user is informed that after System Reset and Power-On Reset the wait cycles are configured to have a maximum allowed frequency of 100 MHz for f_{FSI} and f_{FSI2}.

In summary this results in the following reset behavior:

- Power-on reset and system reset: both change the wait-cycles to a value sufficient for f_{FSI} and f_{FSI2} at max 100 MHz
- Application reset: changes the wait-cycles to a value not disclosed in the User’s Manual. This value is WSPFLASH=10, WSECPF=2, WSDFLASH=45, WSECDF=2

Recommendation

Consequently after each reset the application software shall write values adapted to the clock configuration as described in the section “Configuring Flash Wait Cycles”.

4.125 [PORTS_TC.H006] Using P33.8 while SMU is disabled

Description

Per default, the SMU is enabled (SMU_CLC = 0x0) and collects the alarms from the safety mechanisms defined by the safety concept. The SMU may optionally use P33.8 to output the Fault Signaling Protocol (FSP), selectable via register SMU_PCTL. To satisfy safety requirements, it is ensured that the pad configuration of this pin is not affected by an application or system reset after the first 0-to-1 transition of bit SMU_PCTL.PCS.

If the SMU is enabled, but is not using P33.8 for the FSP function, this pin may be used as general purpose input/output (GPIO) or alternate function input/output, controlled via the corresponding P33 registers.

However, if the SMU is disabled by software (SMU_CLC.DISR = 1_B, i.e. not clocked), configuration of P33.8 (pull devices, driver settings, selection of alternate function, etc.) requires special considerations as described in the following, otherwise the configuration change may not become effective.

Recommendations

- If P33.8 shall be used as GPIO or alternate function input/output, do not disable the SMU, i.e. keep SMU_CLC = 0x0 (default after reset). In this case, the configuration of P33.8 may be changed by software at any time
- Alternatively, configure P33.8 before the SMU is disabled by software (SMU_CLC.DISR = 1_B). After the SMU is disabled, the configuration of P33.8 can no longer be modified by software
- Alternatively, if the SMU is disabled by software (SMU_CLC.DISR = 1_B, i.e. not clocked), clear bit position 8 at address 0xF003 D364 in the P33 address space once after any reset (Application, System Reset, PORST) before configuring P33.8. Controlling P33.8 as FSP by SMU is possible only once after a reset

***Note**: Write access to address 0xF003 D364 is Safety ENDINIT protected.*

4.126 [PORTS_TC.H016] Oscillating signal may enable DXCPL and reconfigure the functionality of the port pins P14.0 and P14.1

Description

The port pin P14.1 can be configured as input for different modules such as GTM input, CAN input, FlexRay input or General Purpose Input. In case oscillations are appearing on this input, DXCPL may get enabled unintentionally on P14.0 and P14.1 and disable the module previously assigned to the pins.

Recommendation

Please refer to application note AP32264 “DXCPL DAP over CAN Physical Layer” for further information and how this situation can be prevented.

***Note**: See also MultiCAN_TC.H007 (Oscillating CAN Bus may Disable the CAN Interface).*

4.127 [QSPI_TC.H005] Stopping Transmission in Continuous Mode

Description

The QSPI module supports the following mechanisms to (temporarily) suspend its operation:

- Pause by setting bit GLOBALCON.EN = 0_B via software
- Disable by setting bit CLC.DISR = 1_B via software
- Sleep Mode (enabled with CLC.EDIS = 0) requested by hardware
- Suspend Mode requested by hardware (debugger)

These modes and their handling is described in detail in section “Operation Modes” of the QSPI chapter in the User’s Manual.

In **Continuous Mode** , the following specific behavior of QSPI module has to be considered:

- In case the QSPI module is put into **Pause** state by setting bit GLOBALCON.EN = 0_B via software, it continues transmission until the end of the TRAIL phase of the frame with BACON.LAST = 1_B
- In case the QSPI module is put into **Disable**, **Sleep**, or **Suspend** mode, the frame is stopped after the next trailing delay (character n). In case BACON.LAST was not =1_B at that time, transmission continues with character n+2 when operation from Disable/Sleep/Suspend state is resumed, i.e. data loss (character n+1) will occur

Recommendation

Ensure that software does not put the QSPI module into Pause or Disable state (via GLOBALCON.EN or CLC.DISR) while a transmission in Continuous Mode is ongoing.

If Sleep Mode is used in the system, disable acceptance of sleep requests (set CLC.EDIS = 1_B) before starting data transmission in Continuous Mode.

During debugging, ensure that the QSPI is not suspended while it is transmitting in Continuous Mode.

4.128 [QSPI_TC.H006] Corrections to Figures “QSPI - Frequency Domains” and “Phase Duration Control, Overview”

Description

In the current version of the User’s Manual,

- Figure “QSPI - Frequency Domains” erroneously uses the term “f_{PER}” instead of “f_{BAUD2}”, and
- Figure “Phase Duration Control, Overview” erroneously uses the term “T_{PER}” instead of “T_{BAUD2}”

Correction

- $f_{SCLK} = 1/f_{BAUD2}$ in Figure “QSPI - Frequency Domains”, and
- $T_{BAUD2} = 1/f_{BAUD2}$ in Figure “Phase Duration Control, Overview”

4.129 [QSPI_TC.H007] RXFIFO Overflow Bit Behavior in Slave Mode

Description

In slave mode, if no data word has been written to TXFIFO during initialization before the master starts sending data, the error flag corresponding to an RXFIFO overflow (bit STATUS.5) is set to 1_B.

Recommendation

To avoid this RXFIFO overflow event, write (at least) one word to TXFIFO during initialization and after each reset in slave mode. For following transmissions, no data need to be written to TXFIFO to avoid this effect.

4.130 [QSPI_TC.H008] Details of the baud rate and phase duration control - Documentation update

Description

To enhance readability, the last part of the second paragraph in the QSPI chapter “Details of the Baud Rate and Phase Duration Control”, starting with “Variations in the baud rates of the slaves ..”, shall be rephrased as shown below.

For further details see also the formulas in the chapter mentioned above and in the figures in chapter “Calculation of the Baud Rates and the Delays” in the User’s Manual.

Documentation update

Variations in the baud rates of slaves of one module are supported by the ECONz.Q and the ECONz.A/B/C bit-field settings allowing for a flexible bit time variation between the channels in one module.

4.131 [QSPI_TC.H009] Dummy frame required after changing SCLK polarity and phase in three wire mode

Description

When three wire mode is used, and the SCLK polarity (bit ECONz.CPOL) or phase (bit ECONz.CPH) of the master is changed by software, the state of the clock and data signals is not defined before the first data is transmitted.

This may result in wrong data being received or transmitted by the slave.

Recommendation

After the SCLK polarity (bit ECONz.CPOL) or phase (bit ECONz.CPH) is changed by software, transmission of a dummy frame is required. The pad enable shall be after transmission of the dummy frame, such that the slave will not notice the dummy frame.

***Note**: In four wire mode where the slave is controlled by a select signal from the master, this issue has no effect, because the output signals from the master are at the correct levels by the time the slave select signal gets active.*

4.132 [QSPI_TC.H011] Missing information on SLSI misplaced inactivation enable error

Description

Missing information for error interrupt "SLSI misplaced inactivation" in the Status Register.

Recommendation

The documentation will be updated as follows:

- SLSI misplaced inactivation error interrupt is raised when SLSI is deactivated by the master while the data transfer is still ongoing

4.133 [RESET_TC.H002] Unexpected SMU Reset Indication in SCU_RSTSTAT

Description

Under certain conditions the Reset Status Register SCU_RSTSTAT can show an SMU reset indication in addition to the real reset trigger (for example a SW reset).

The explanation of this behavior refers to section “Reset Generation” and following pages in chapter “RCU” of the User’s Manual.

Figure “Reset Overview” shows that all warm resets are executed in a defined sequence. This sequence ensures that first the active CPUs are ramped down, then at 80μs the Flash receives an idle request and at 180μs the reset is executed.

The idle request to the Flash makes it immediately busy, all read requests after this point fail with a bus error. All non-CPU masters (HSM, Ethernet, HSSL, DMA and DAM) however continue operation from 80μs to 180μs. When one of these masters reads the busy Flash, a bus error is signaled to the SMU as alarm ALM3[30] (SRI) and/or ALM3[31] (SPB).

If the SMU is configured to react on this by a reset request, this will be noted in the SCU_RSTSTAT register in addition to the original warm reset.

This applies mainly to the master HSM which fetches its code from PFlash.

Recommendations

- Generally a different alarm handling can be configured in the SMU for the mentioned alarms, for example trigger an NMI trap but not a reset
- When the application detects after reset that SCU_RSTSTAT has an additional SMU reset indication it might ignore it and proceed based on the other reset indication
- In case of SW resets the application can prepare the system just before activating the reset:
  - The non-CPU masters can be disabled or in case of HSM it can be informed about the imminent SW reset and continue execution from RAM

- The mentioned alarms can be disabled or the alarm reaction can be changed to trigger an NMI trap
- The SMU module reset can be used to reconfigure the SMU into its initial state in which only watchdog timeout alarms are handled

4.134 [RESET_TC.H003] Usage of the Prolongation Feature for ESR0 as Reset Indicator Output

Description

The ESR0 pin can be used as reset indicator output and in such a case its active low state can be prolonged upon user-configurable selection as described in section “ESRx as Reset Output” of chapter “Reset Control Unit (RCU) in the User’s Manual.

According to this description, an ESR0CNT value of 0 defines “as soon as possible after start of Boot Code execution”, where “as soon as possible” means:

- about 500 µs after cold power-on
- not less than 20 µs after other types of reset

Warning

In case of ESR0CNT = 2, the ESR0 pin will never be released by the device and the user code will never start.

***Note**: On the other hand - as explained before - configuring an ESR0CNT value of 1 or 2 would anyhow not be effective as a prolongation time below 20 µs is conceptually unachievable.*

Recommendation

Do not configure ESR0CNT = 2.

If prolongation of about 20 µs or below is needed, configure ESR0CNT = 3 or 0 instead.

4.135 [RESET_TC.H004] Effect of Power-on and System Reset on DSPR

Description

This documentation issue applies to

- footnote **3)** on Table “Effect of Reset on Device Functions” in the RCU chapter “Module Reset Behavior” in the **TC29x**, **TC27x**, and **TC26x** User’s Manuals
- footnote **2)** on Table “Effect of Reset on Device Functions” in the RCU chapter “Module Reset Behavior” in the **TC21x/TC22x/TC23x Family** User’s Manual

The following part of this footnote regarding the effect of startup firmware on Data Scratchpad RAM (DSPR) is incorrect:

- “DSPR is partially used as a scratchpad by the startup firmware. Previous data stored in the upper 32kB will be overwritten on start-up”

The correct effect is described in the Boot ROM chapter “RAM overwrite during start-up“:

- Start-up procedure upon power-on and system reset can overwrite up to 8 Kbyte at the beginning of CPU0 DSPR

4.136 [SCU_TC.H009] LBIST Influence on Pad Behavior

Description

The behavior of the GPIO and ESR0/1 pads during LBIST execution is as follows:

- ESR0 is switched to input direction during LBIST with weak pull-up and pull-down driver disabled (i.e. pad is tri-stated)

- ESR1 is switched to input direction during LBIST with weak pull-down driver enabled
- Other GPIO pins are switched to input direction with weak pull-up devices either stable active or inactive (depending on LBIST user configuration)

4.137 [SCU_TC.H010] LBIST Signature Depends on Debug Interface Configuration

Description

The following three cases generate different sets of LBIST MISR signatures:

1. Pin $\overline{\text{TRST}}$ is held high during $\overline{\text{PORST}}$ rising edge (DAP operation): In this case the further values of $\overline{\text{TRST}}$ will have no influence on the MISR signature
2. Pin $\overline{\text{TRST}}$ is held low during $\overline{\text{PORST}}$ rising edge (JTAG operation): $\overline{\text{TRST}}$ is held continuously low also during LBIST operation
3. Pin $\overline{\text{TRST}}$ is held low during $\overline{\text{PORST}}$ rising edge (JTAG operation): $\overline{\text{TRST}}$ is switched to high after $\overline{\text{PORST}}$ has been released and at least one pulse occurred at TCK before LBIST starts

If DXCM/DXCPL (Debug over CAN) is not needed it is recommended to keep pin $\overline{\text{TRST}}$ always at a high level during $\overline{\text{PORST}}$ rising-edge in the application environment (also in final application). This makes the MISR signature independent from further $\overline{\text{TRST}}$ behavior and still allows debug access via DAP or to completely disable the debug IF via software (by setting OIFM.DAPMODE = 111_B).

In the DXCM/DXCPL enabled case it is recommended to keep pin $\overline{\text{TRST}}$ always at a low level (also after $\overline{\text{PORST}}$ has been released). In this operation case a different set of MISR signatures will be received (case 2 in above list). Consequently the application software needs to be prepared to accept LBIST MISR signature results from case 1 or from case 2 as pass criteria.

4.138 [SCU_TC.H013] Correction to Register References in Chapter “Watchdog Timers”

Description

Some references to register names in chapter “Watchdog Timers” of the User’s Manual are incorrect.

The corrected references and their section headers are listed in **bold** below.

Section Password Access to WDTxCON0

.. To ensure that a CPU fault could not allow a fault to be ignored an option is provided to prevent watchdog unlocking if the Safety Management Unit (SMU) is not in the RUN state. This option may be enabled by bit **WDTxCON1**.UR. If the password is valid and the SMU state meets the requirements of the **WDTxSR**.US bit then WDTxCON0 will be unlocked as soon as the Password Access is completed. ..

Section Timer Operation

.. The parameter divider represents the user-programmable source clock division selected by **WDTxCON1**.IRx, which can be 64, 256 or 16384.

Section Watchdog Timer Registers

- **WDTSCON1** - Safety WDT Control Register 1:
  - References to WDTxCON0 and WDTxSR should be consequently to **WDTSCON0** and **WDTSSR** in the context of WDTSCON1
- **WDTCPUxCON1** - CPUx WDT Control Register 1:
  - References to WDTSCON0 and WDTSSR should be consequently to **WDTCPUxCON0** and **WDTCPUxSR** in the context of WDTCPUxCON1

4.139 [SCU_TC.H014] Reset Value of Bit Field IOCR.PC1 - Control for Pin $\overline{\text{ESR1}}$

Description

The reset value of register SCU_IOCR is documented as 0000 $20E0_H$ in chapter "Reset Control Units" of the User's Manual, i.e. the reset value of bit-field PC1 = 2_H.

This is not always correct under all circumstances:

The actual SCU_IOCR reset value should be considered as 0000 $X0E0_H$ with the explanations given in the following Documentation Update.

Documentation Update

The reset value of bit-field SCU_IOCR.PC1 is influenced by pin HWCFG6 and bit PMSWCR0.TRISTREQ:

- When a cold reset is activated and HWCFG6=1 then PC1 is reset to 2_H and pin $\overline{\text{ESR1}}$ will have input pull-up mode
- If HWCFG6=0 then PC1 is reset to 0_H and $\overline{\text{ESR1}}$ will have tri-state mode

PC1 and the $\overline{\text{ESR1}}$ reset state can also be configured by software with the PMSWCR0.TRISTREQ bit. PMSWCR0.TRISTREQ is not affected by warm reset or wake-up from standby so the IOCR.PC1 reset value is configured as per the state of the TRISTREQ bit prior to the warm reset.

4.140 [SCU_TC.H027] Bit field INP0 and INP1 in register EICRi - Documentation correction

Description

In the SCU chapter of the current user manual, for settings INP0 = 100_B to 111_B and INP1 = 100_B to 111_B in the description of register EICRi, the last index y of signal TRxy is erroneously shown a 0.

In the description for INP0, the enable bit is erroneously referenced as EIEN(2i) instead of EICRi.EIEN0, and as EIEN(2i+1) instead of EICRi.EIEN1 in the description for INP1.

Documentation correction

The last index y of signal TRxy shall be identical to the OGUy index. The corrected description for INP0 and settings INP0 = 100_B to 111_B and for INP1 and settings INP1 = 100_B to 111_B is shown in the following table.

Table 52 **Field INP0 and INP1 in register EICRi (i=0-3) - Correction**

Field	Bits	Type	Description
INP0	14:12	rw	**Input Node Pointer** This bit-field determines the destination (output channel) for trigger event (2i) (if enabled by **EICRi.EIEN0**). 100_B An event from input ETL 2i triggers output OGU4 (signal TR(2i) **4**) 101_B An event from input ETL 2i triggers output OGU5 (signal TR(2i) **5**) 110_B An event from input ETL 2i triggers output OGU6 (signal TR(2i) **6**) 111_B An event from input ETL 2i triggers output OGU7 (signal TR(2i) **7**)
INP1	30:28	rw	**Input Node Pointer** This bit-field determines the destination (output channel) for trigger event (2i+1) (if enabled by **EICRi.EIEN1**). 100_B An event from input ETL 2i+1 triggers output OGU4 (signal TR(2i+1) **4**)

(table continues...)

Table 52 (continued) Field INP0 and INP1 in register EICRi (i=0-3) - Correction

Field	Bits	Type	Description
			101_B An event from input ETL 2i+1 triggers output OGU5 (signal TR(2i+1) **5**) 110_B An event from input ETL 2i+1 triggers output OGU6 (signal TR(2i+1) **6**) 111_B An event from input ETL 2i+1 triggers output OGU7 (signal TR(2i+1) **7**)

***Note**: In the table above, only rows that include corrections are shown.*

4.141 [SCU_TC.H028] ERU configuration changes may lead to ERU reactions

Description

The External Request Unit (ERU) may react on changes of control registers even if there is no edge at its inputs.

For example, if one of the inputs of an input channel x is '1' and this is switched to another input of this channel (by EICRy.EXISz) that is '0', then ERU recognizes an edge if configured for this input channel x and the corresponding EIFR.INTFx is set and the trigger is propagated to the ERU output as configured.

Recommendation

Clear EIFR.INTFx bits after (re-)configuration.

If an ERU reaction is to be suppressed on configuration changes (and you suspect there might be two different levels at the two ERU inputs to be switched), then:

- Clear bits EICRy.RENz, EICRy.FENz without changing EICRy.EXISz (so potential edges are swallowed at the 'Detect Event (edge)' block)
- With a 2nd write access to EICRy set bits EICRy.EXISz as needed without changing the EICRy.RENz, EICRy.FENz
- Wait long enough

  The wait time depends on the ERU input filter setting

  In case the filter is active, the 3rd access to EICRy has to happen after EIFILT.DEPTH * (EIFILT.FILTDIV + 1) SPB (100 MHz) clock cycles, otherwise the edge is still traveling through the filter and has not arrived at the 'Detect Event (edge)' block yet, to be swallowed as intended
- Then with a 3rd write access set EICRy.RENz, EICRy.FENz as needed without changing the EICRy.EXISz

4.142 [SENT_TC.H003] First Write Access to Registers FDR and TPD after ENDINIT Status Change

Description

Due to an extra registering stage of the ENDINIT signal from the SCU inside the SENT kernel, the behavior of the first write access to SENT registers FDR and TPD protected by the Endinit write protection scheme after an ENDINIT status change is as follows:

- After unlocking protection (ENDINIT change from 1 to 0), if the first access to the SENT module is a write to FDR or TPD, it will still view ENDINIT as locked (value 1). The contents of FDR or TPD is not changed, but no BCU alarm will be generated, as the ENDINIT does not indicate a protected status in case of the access
- By setting protection again (ENDINIT change from 0 to 1), if the first access to the SENT module is a write to FDR or TPD, it will still be effective, i.e., the value will be written. Nevertheless a SMU alarm through BCU will be generated as the protection status is ENDINIT

***Note**: After the first read of any SENT register, or first write to any SENT register, the ENDINIT change will be correctly considered for all following accesses. The CLC, KRST0/1 and KRSTCLR registers (that also have Endinit protection) are not affected at all. An initial value of 0 for ENDINIT is seen by SENT after reset before the first access.*

Recommendation

After a change of the ENDINIT protection status, first perform a read of any SENT register or a write to a non-Endinit-protected SENT register. The second access is then always equipped with correct information of ENDINIT.

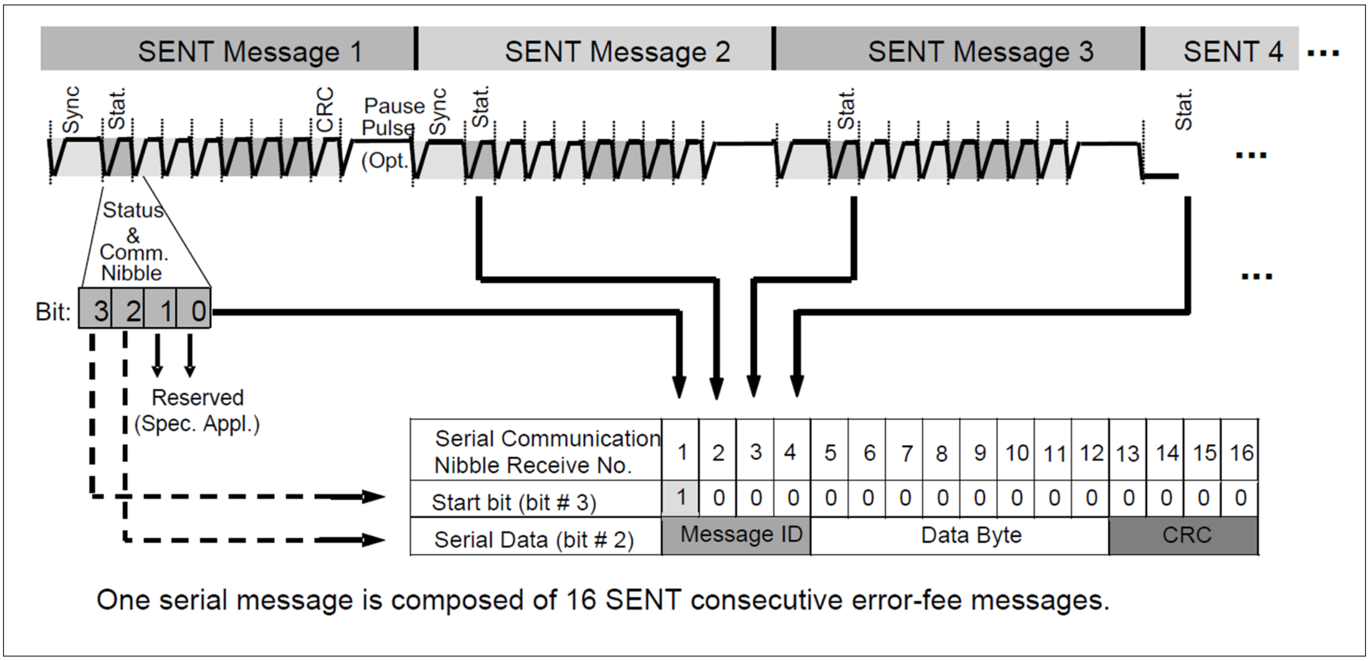
4.143 [SENT_TC.H004] Short Serial Message - Figure Correction

Description

In Figure “Short Serial Message, Serial Data Encoding over 16 messages” of the SENT chapter, the arrows originating from bits 2 and 3 of the Status & Comm Nibble are routed incorrectly and must be swapped.

Correction

Figure 15 shows a corrected version of this figure.



Serial Communication Nibble Receive No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Start bit (bit # 3)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Serial Data (bit # 2)	Message ID				Data Byte								CRC			

One serial message is composed of 16 SENT consecutive error-fee messages.

Figure 15 Short Serial Message, Serial Data Encoding over 16 messages

4.144 [SENT_TC.H005] Interface Connections of the SENT Module - Documentation Correction

Description

The following corrections apply to chapter “Interface Connections of the SENT Module” in the SENT chapter of the TC29x, TC26x, and TC21x/TC22x/TC23x Family User’s Manuals:

Corrections to TC29x User’s Manual

- **Figure “SENT Module Implementation and Interconnections” for TC29x**
  The range of index n for connected trigger inputs TRIGn in TC29x is n = 0..10.

- **Interrupt and DMA Controller Service Requests for TC29x**

  Table “Service Request Lines of SENT” in the TC29x User’s Manual specifies 10 request lines (SR0..3 and TRIGO0..5) as connected to the Interrupt Router.

  In table “Registers Overview - Service Request Control Registers” in the Interrupt Router chapter of the TC29x User’s Manual, registers SRC_SENT10..14 should be regarded as “Reserved”.
- **Trigger Inputs for TC29x**

  In TC29x, the SENT module has 15 SENT channels and 11 trigger inputs TRIG[10:0] selectable in bit-field IOCRx.ETS. Values ≥ 1010_B are reserved and should not be used for bit-field ETS in register IOCRx.

  The trigger inputs (TRIG[10:0]) of the SENT module are connected to the GTM. The rows shown in the table below replace the rows in the corresponding table of the User’s Manual:

Table 53 Trigger Input Lines of SENT - Correction for TC29x

Request Line	Connected to	Description
TRIG9	TRIG9	GTM.DSADC_0_TRIG_0
TRIG10	TRIG10	GTM.ADC_10_TRIG_0
TRIG11..TRIG15	-	Not Connected

Corrections to TC26x User’s Manual

- **Figure “SENT Module Implementation and Interconnections” for TC26x**

  The range of index n for connected trigger inputs TRIGn in TC26x is n = 0..5.
- **Interrupt and DMA Controller Service Requests for TC26x**

  The trigger outputs of the SENT module are connected via the Interrupt Router and can be selected in register INPx. The row shown in the table below replaces the row in the corresponding table of the TC26x User’s Manual:

Table 54 Service Request Lines of SENT - Correction for TC26x

INP value	Request Line	Connected to	Description
0101_B	TRIGO1	SRC_SENT5	Interrupt Router SENT Request 5

Corrections to TC21x/TC22x/TC23x Family User’s Manual

- **Figure “SENT Module Implementation and Interconnections” for TC21x/TC22x/TC23x**
  - In TC23x/TC22x/TC21x, no TRIGOn signals are connected from the SENT module to the Interrupt Router (IR). All references to TRIGOn should be ignored in this figure
  - The range of index n for connected trigger inputs TRIGn in TC23x/TC22x/ TC21x is n = 0..3
- **Interrupt and DMA Controller Service Requests for TC21x/TC22x/TC23x**

  In TC23x/TC22x/TC21x, request lines SR0..3 of the SENT module are connected via the Interrupt Router and can be selected in register INPx accordingly. Values ≥ 0100_B are reserved and should not be used for the bit-fields in register INPx.

4.145 [SENT_TC.H009] Unexpected NNI error behavior

Description

The NNI interrupt is triggered when the actual number of transmitted nibbles exceeds the expected count predefined in RCRx.FRL. Specifically, when IEP = 0 and no pause pulse is used, NNI interrupt performs as expected. However, when IEP = 1 and a pause pulse is used, the interrupt is not triggered if the number of transmitted nibbles surpasses the expected value by one nibble. In this case, the NNI interrupt is only triggered when the number of nibbles transmitted surpasses the expected value by two or more nibbles.

Recommendation

Due to this issue, SENT messages could be missed. This can be detected by implementing timeout or message rate checking mechanisms.

4.146 [SMU_TC.H001] Write all bit-fields of SMU_PCTL with one write access

Description

When configuring the FSP pin (for example P33.8), all bit-fields (HWDIR, HWEN and PCS) of register SMU_PCTL must be written with the same write access.

Otherwise, when first writing a 1_B to HWEN before writing a 1_B to PCS, the pad configuration will be modified to push/pull configuration before it is latched into field PCFG.

***Note**: When PCS = 1_B, the bit-fields PCFG and PCS are protected against any changes until the next power on reset. HWEN and HWDIR may still be modified by SW, unless locked via register SMU_KEYS.*

4.147 [SMU_TC.H005] Correction to Figure “SMU Register Map”

Description

The start address “@SMU + 0x0E0” for the SMU System Registers shown in the lower part of figure “SMU Register Map” in the SMU chapter of the User’s Manual is incorrect.

The correct start address is “@SMU + 0x**7E0**”.

Addresses listed in table “Registers Overview” of the SMU chapter are correct.

4.148 [SMU_TC.H006] Description of Bit EFRST in Register SMU_AGC

Description

In the SMU chapter of the User’s Manual, the description of the encoding of bit EFRST (Enable FAULT to RUN State Transition) in register SMU_AGC (Alarm Global Configuration) is missing.

The complete description should be as shown in Table 55:

Table 55 Bit EFRST in Register SMU_AGC

Field	Bits	Type	Description
EFRST	29	rw	**Enable FAULT to RUN State Transition** 0_B FAULT to RUN State Transition disabled 1_B FAULT to RUN State Transition enabled See section “**FSP Fault State** ” for the usage of this field.

4.149 [SMU_TC.H007] SPB Bus Control Unit (SBCU) Alarm Signalling to SMU

Description

ALM3[31] is dedicated to System Peripheral Bus (SPB) alarms. As described in table “Alarm Mapping related to ALM3 group” in the SMU chapter of the User’s Manual, an SPB bus error can result from multiple root causes, including protocol violation, incorrect address, register access protection violation.

More details on the SPB related error conditions can be found in the “On-Chip Bus System” chapter:

The SBCU signals an alarm to the SMU whenever it detects

- a SPB transaction that was finished with a Bus Error (Error Acknowledge)
- an un-implemented Address (no slave responds to a transaction request)
- a SPB transaction that was finished by a Time-out

The alarm signaling to the SMU is independent of the BCU configuration (for example BCU interrupt configuration, BCU debug status).

4.150 [SMU_TC.H009] Alarm Table Corrections

Description

Some alarm tables were unintentionally changed between User's Manual (UM) version V1.0 and V1.1.

In the following, the issues are described and the correct table parts are included.

Table 10-2 HwAlarmOut[3:0]: CPU0 DCACHE/DSPR SRAM

The PreAlarms 1, 3, 5, 7 of the DSPR2 part (TC23x only) were accidentally removed in UM V1.1.

The following Table 56 copied from UM V1.0 is correct:

Table 56 Table 10-2 HwAlarmOut[3:0]: CPU0 DCACHE/DSPR SRAM

Function

HwAlarmOut[0]=
PreAlarm[0=CPU0.DMI.DSPR.(ECC single bit correction)] or
PreAlarm[1=CPU0.DMI.DSPR2.(ECC single bit correction)]

HwAlarmOut[1]=
PreAlarm[2=CPU0.DMI.DSPR.(ECC uncorrectable error)] or
PreAlarm[3=CPU0.DMI.DSPR2.(ECC uncorrectable error)]

HwAlarmOut[3]=
PreAlarm[4=CPU0.DMI.DSPR.(Address error)] or
PreAlarm[5=CPU0.DMI.DSPR2.(Address error)]

HwAlarmOut[4]=
PreAlarm[6=CPU0.DMI.DSPR.(Address buffer overflow)] or
PreAlarm[7=CPU0.DMI.DSPR2.(Address buffer overflow)]

Table 10-3 HwAlarmOut[7:4]: CPU1 DCACHE/DSPR SRAM

HwAlarmOut[7:4] are reserved. The following Table 57 copied from UM V1.0 is correct:

Table 57 Table 10-3 HwAlarmOut[7:4]: CPU1 DCACHE/DSPR SRAM

Function
HwAlarmOut[7:4] are reserved

Table 10-4 HwAlarmOut[15:8]: CPU2 SRAMs

HwAlarmOut[15:8] are reserved. The following Table 58 copied from UM V1.0 is correct:

Table 58 Table 10-4 HwAlarmOut[15:8]: CPU2 SRAMs

Function
HwAlarmOut[15:8] are reserved

Table 10-5 HwAlarmOut[19:16]: GTM SRAMs

HwAlarmOut[19:16] are reserved. The following Table 59 copied from UM V1.0 is correct:

Table 59 Table 10-5 HwAlarmOut[19:16]: GTM SRAMs

Function
HwAlarmOut[19:16] are reserved

Table 10-7 HwAlarmOut[27:24]: CAN SRAM

The PreAlarms of the second CAN module RAMs (TC23x only) were accidentally removed in UM V1.1; these are: 81, 83, 85, 87.
The following Table 60 copied from UM V1.0 is correct:

Table 60 Table 10-7 HwAlarmOut[27:24]: CAN SRAM

Function

HwAlarmOut[24]=
PreAlarm[80=CAN.SRAM.MCAN0.(ECC single bit correction)] or
PreAlarm[81=CAN.SRAM.MCAN1.(ECC single bit correction)]

HwAlarmOut[25]=
PreAlarm[82=CAN.SRAM.MCAN0.(ECC uncorrectable error)] or
PreAlarm[83=CAN.SRAM.MCAN1.(ECC uncorrectable error)]

HwAlarmOut[26]=
PreAlarm[84=CAN.SRAM.MCAN0.(Address error)] or
PreAlarm[85=CAN.SRAM.MCAN1.(Address error)]

HwAlarmOut[27]=
PreAlarm[86=CAN.SRAM.MCAN0.(Address buffer overflow)] or
PreAlarm[87=CAN.SRAM.MCAN1.(Address buffer overflow)]

Table 10-8 HwAlarmOut[31:28]: LMU sub-system SRAMs

The PreAlarms of the FFT RAMs (TC23x ADAS only) were accidentally removed in UM V1.1; these are: 230, 231, 233, 234, 236, 237, 239, 240.
The following Table 61 copied from UM V1.0 is correct:

Table 61 Table 10-8 HwAlarmOut[31:28]: LMU sub-system SRAMs

Function

HwAlarmOut[28]=
PreAlarm[88=LMU.DAM.SRAM(ECC single bit correction)] or
PreAlarm[230=LMU.FFT0.SRAM(ECC single bit correction)] or
PreAlarm[231=LMU.FFT1.SRAM(ECC single bit correction)] or

HwAlarmOut[29]=
PreAlarm[90=LMU.DAM.SRAM(ECC uncorrectable error)] or
PreAlarm[233=LMU.FFT0.SRAM(ECC uncorrectable error))] or
PreAlarm[234=LMU.FFT1.SRAM(ECC uncorrectable error))] or

(table continues...)

Table 61 (continued) Table 10-8 HwAlarmOut[31:28]: LMU sub-system SRAMs

Function

HwAlarmOut[30]=
PreAlarm[92=LMU.DAM.SRAM(Address error)] or
PreAlarm[236=LMU.FFT0.SRAM(Address error)] or
PreAlarm[237=LMU.FFT1.SRAM(Address error)] or

HwAlarmOut[31]=
PreAlarm[94=LMU.DAM.SRAM(Address buffer overflow)] or
PreAlarm[239=LMU.FFT0.SRAM(Address buffer overflow)] or
PreAlarm[240=LMU.FFT1.SRAM(Address buffer overflow)] or

Table 10-9 HwAlarmOut[34:32]: SRI Agents

PreAlarm[116] is accidentally listed as HSSL.SRI_MASTER(SRI Read Data Phase Error) in UM V1.1.
Actually PreAlarm[116] is reserved.

Table 9-11 HwAlarmOut[44:41]: Misc. SRAMs

The PreAlarms accidentally listed as assigned to PSI5 and CIF are actually reserved; these are 145, 147-149, 153, 155-157, 161, 163-165, 169, 171-173.

Table 10-12 HwAlarmOut[45]: Watchdogs Timeout

The PreAlarms accidentally listed as assigned to WDTCPU1 and WDTCPU2 in UM V1.1 are actually reserved; these are 175, 176.

Table 10-13 HwAlarmOut[50:46]: PMU Alarms

The PreAlarms accidentally listed as assigned to PFLASH1 in UM V1.1 are actually reserved; these are: 179, 187, 195, 203, 211.

Table 10-18 TC21x/TC22x/TC23x Alarm Mapping related to ALM1 Group

This table incorrectly shows alarms for CPU1 in UM V1.1.
These alarms actually are reserved, as shown in the following Table 62 copied from UM V1.0:

Table 62 Table 10-18 Alarm Mapping related to ALM1 Group

Alarm Index	Module	Description
ALM1[31:0]	Reserved	Reserved

Table 10-20 TC21x/TC22x/TC23x Alarm Mapping related to ALM3 Group

The rows in the following Table 63 replace the corresponding rows of the table in UM V1.1:

Table 63 Part of Table 10-20 with corrected ALM3 Group Mapping

Alarm Index	Module	Description
ALM3[9]	Reserved	Reserved
ALM3[13]	Reserved	Reserved
ALM3[14]	Reserved	Reserved
ALM3[19]	Reserved	Reserved

(table continues...)

Table 63 (continued) Part of Table 10-20 with corrected ALM3 Group Mapping

Alarm Index	Module	Description
ALM3[20]	Reserved	Reserved
ALM3[27]	**Registers**	**Safety Mechanism: Register Monitor** **Alarm: register error detection**
ALM3[28]	SCU/LSCU	Safety Mechanism: Lockstep Dual Rail Monitor Alarm: dual rail error Note: monitors the dual-rail property (inverted signals) from the lockstep comparator unit (LSCU) alarms.

Table 10-21 TC21x/TC22x/TC23x Alarm Mapping related to ALM4 Group

There are no GTM SRAMs in this device. The rows in the following Table 64 replace the corresponding rows of the table in UM V1.1:

Table 64 Part of Table 10-21 with corrected ALM4 Group Mapping

Alarm Index	Module	Description
ALM4[3:0]	Reserved	Reserved
ALM4[7:4] connects to pre-alarms specified by table “HwAlarmOut[27:24]: CAN SRAM”		

Table 10-23 TC21x/TC22x/TC23x Alarm Mapping related to ALM6 Group

This table incorrectly shows alarms for CPU2 in UM V1.1.
These alarms actually are reserved, as shown in the following Table 65 copied from UM V1.0:

Table 65 Table 10-23 Alarm Mapping related to ALM6 Group

Alarm Index	Module	Description
ALM6[31:0]	Reserved	Reserved

4.151 [SMU_TC.H010] Clearing individual SMU flags: use only 32-bit writes

Description

The SMU registers shall only be written via 32-bit word accesses (i.e. ST.W instruction), as mentioned in table “Registers Overview” of the SMU chapter in the User’s Manual.

If any other instruction such as LDMST or SWAPMSK.W is used to modify only a few bits in the 32-bit register, then this may have the effect of modifying/clearing unintended bits.

Recommendation (Examples in C Language)

- **Example 1**: To clear status flag SF2 in register AG0, use:
  - SMU_AG0.U = 0x0000 0004;
- **Example 2**: To clear status flags EF2 in register RMEF and RMSTS, use:
  - SMU_RMEF.U = 0xFFFF FFFB;
  - SMU_RMSTS.U = 0xFFFF FFFB;

Here the <REGISTER>.U implies writing to the register as an unsigned integer, which normally results in a compiler translation into an ST.W instruction.

Safety Considerations

As long as software uses only 32-bit writes to the SMU registers, there is no risk of malfunction.

In case the software does not use 32-bit writes (and for example uses bit-wise operations such as LDMST instructions instead) – then potentially unintended flags may be written and modified in the SMU registers. Depending on the application, this may potentially have an impact on safety and/or diagnostics.

***Note**: The SMU reaction itself (for example alarm action triggering) is not affected even if the software unintentionally clears additional bits by not using a 32-bit write as recommended.*

4.152 [SMU_TC.H013] Increased Fault Detection for SMU Bus Interface (SMU_CLC Register)

Description

Transient faults can possibly affect the SMU_CLC register and lead to disabling the SMU_core. This unintended switching off of SMU_core cannot be detected if the FSP protocol is not used at all or used in FSP bi-stable mode.

Recommendation

In order to increase the capability of the microcontroller to detect such faults it is recommended to:

- Option 1:

  Use FSP Dynamic dual-rail or Time-switching protocol only, don’t use FSP bi-stable protocol
- Option 2:

  In case FSP protocol is not used at all or Recommendation Option 1 is not possible, the [Application SW] shall read periodically, once per FTTI, the SMU_CLC register to react on unintended disabled SMU

4.153 [SMU_TC.H014] Unintended short pulse on FSP pins in Time switching or Dual-rail mode

Description

Due to an internal synchronization issue, an unintended short pulse of a duration of around 80 ns can be seen on the FSP pins if the FSP pins are configured for Time switching or Dual-rail mode, and one of the following scenarios happens in the SMU state machine:

- Scenario a): transition from START to RUN state
- Scenario b): transition from FAULT to RUN (Fault-Free) state

Recommendation

- Workaround for scenario a):
  - Enable FSP by writing SMU_PCTL register 10 SPB clock cycles (or more) after sending SMU_ReleaseFSP() command
- Assessment for scenario b):
  - The pulse in scenario b), if it occurs, cannot be avoided but has no safety impact as the unintended pulse happens during the transition from fault state to fault-free state. This state transition is not considered as safety relevant

4.154 [SRI_TC.H001] Using LDMST and SWAPMSK.W instructions on SRI mapped peripheral registers (range 0xF800 0000-0xFFFF FFFF)

Description

The LDMST and SWAPMSK.W instructions in the AURIX™ microcontrollers are intended to provide atomicity as well as bit-wise operations to a targeted memory location or peripheral register. They are also referred to as Read-Modify-Write (RMW) instructions.

The bit-manipulation functionality is intended to provide software a mechanism to write to individual bits in a register, without affecting other bits. The bits to be written can be selected through a mask in the instruction. Please refer to the TriCore™ Architecture Manual for further information about these instructions and their formats.

Restrictions for SRI mapped Peripherals

The bit-manipulation functionality is supported only on registers accessed via the SPB bus, and is not supported on the SRI mapped peripheral range, that is address range 0xF800 0000 to 0xFFFF FFFF

The SRI mapped peripheral range includes the following units (if available):

- In **TC2xx**: EBU, PMU0, SRI Crossbar, LMU, DAM, FFT, CPUx SFRs and CSFRs, MCDS, miniMCDS; see table “On Chip Bus Address Map of Segment 15” in chapter “Memory Map”
- In **TC3xx**: DMU, LMU, EBU, DAM, SRI Crossbar, SPU, CPUx SFRs and CSFRs, AGBT, miniMCDS, ...; see table “On Chip Bus Address Map of Segment 15” in chapter “Memory Map”

On the SRI mapped peripherals, usage of these instructions always results in all the bits of a register being written, and not just specific individual bits.

***Note**: The instructions are still executed atomically on the bus – that is, the SRI is locked between the READ and the WRITE transaction.*

4.155 [STM_TC.H001] Effect of kernel reset on interrupt outputs STMIR0/1

Description

The clock ratio f_{STM} : f_{SPB} is determined by the settings of bit-fields STMDIV and SPBDIV in registers CCUCON1 and CCUCON0, respectively.

If $f_{STM} \leq f_{SPB}$, and a kernel reset of the STM module is performed in the same clock cycle where a compare match of the STM with the CMP0 or CMP1 registers occurs, a transition on the interrupt outputs STMIR0 or STMIR1 may occur. This may for example trigger the External Request Unit (ERU), or set the corresponding Service Request flags SRC_STMmSR0.SRR or SRC_STMmSR1.SRR in the Interrupt Router (m = 0, 1, 2, depending on number of CPUs).

***Note**: For $f_{STM} > f_{SPB}$, this effect will not occur.*

Recommendation

If $f_{STM} \leq f_{SPB}$, set bits ICR.CMP0EN = 0_B and ICR.CMP1EN = 0_B to disable the compare match interrupts before performing the STM kernel reset.

4.156 [STM_TC.H002] Access Protection for STM Control Registers

Description

The access protection symbol ‘P’ to indicate Access Enable Register protection is missing in table “Registers Overview - STM Control Registers” of the STM chapter in the User’s Manual for the STM registers CMP0, CMP1, CMCON, ICR, ISCR.

The STM registers CMP0, CMP1, CMCON, ICR, ISCR actually have protection via the Access Enable registers (ACCEN0/1), as shown in the following Table 66.

Table 66 Correction to Table Registers Overview - STM Control Registers

Short Name	Description	Offset Addr.	Access Mode		Reset
			Read	Write	
CMP0	Compare Register 0	30_H	U, SV	U, SV, P	Application
CMP1	Compare Register 1	34_H	U, SV	U, SV, P	Application
CMCON	Compare Match Control Register	38_H	U, SV	U, SV, P	Application
ICR	Interrupt Control Register	$3C_H$	U, SV	U, SV, P	Application
ISCR	Interrupt Set/Clear Register	40_H	U, SV	U, SV, P	Application

4.157 [STM_TC.H003] Suspend control for STMx - Documentation Update

Description

In contrast to the register description of bit OCS.SUS in the STM chapter of the current User's Manual, the suspend functionality of STMx is controlled by signal CPUxSUSOUT of the corresponding CPUx (and not by the signal coming from the OCDS Trigger Switch (OTGS)).

Therefore, the description for bit OCS.SUS in the STM chapter should read:

- “Controls the sensitivity to the suspend signal coming from the CPU (CPUxSUSOUT)”

4.158 [STM_TC.H004] Access to STM registers while STMDIV = 0

Description

If accesses to STM kernel registers are performed while bit-field STMDIV = 0_H in the corresponding CCU Clock Control register (that is, clock f_{STM} is stopped),

- the SPB bus gets locked after the first access until a timeout (defined in BCU Control register field SBCU_CON.TOUT) occurs;
- after the second access the STM slave will answer with RTY (retry) until the STM is clocked again with STMDIV > 0_H

The corresponding CCU Clock Control register including STMDIV is:

- CCUCON1 in **TC2xx**
- CCUCON0 in **TC3xx**

Recommendation

- In **TC2xx**, do not access any STM kernel register while CCUCON1.STMDIV = 0_H
- In **TC3xx**, do not access any STM kernel register while CCUCON0.STMDIV = 0_H

Revision history

Document version	Date of release	Description of changes
1.0	2016-10-21	• New/updated text modules see columns “Change” in tables 4..6 of errata sheet V1.0 • The following text modules have been removed, as no device variant exists of this design step (AC) that includes EMEM, ETH, FFT modules: - EMEM_TC.H002 (EMEM will raise ECC errors when not properly initialized) - ETH_AI.003 (Overflow Status bits of Missed Frame and Buffer Overflow counters get cleared without a Read operation) - ETH_AI.H001 (Sequence for Switching between MII and RMII Modes) - ETH_TC.004 (DMA Access to Reserved/Protected Resources: FPI Error Response not correctly evaluated) - ETH_TC.H002 (Minimum operation frequency for Ethernet MAC) - ETH_TC.H003 (Interrupt Generation by Wake-up or Magic Packet Frames) - FFT_TC.001 (FFT Access with disabled FFT Module) - FFT_TC.002 (FFT Kernel Reset Function) - FFT_TC.003 (No Error reported upon Write to FFT Registers in User Mode) • The following text module has been removed, as no ADAS variant exists of this design step (AC) - SCU_TC.H012 (Overlay Feature for ADAS Variants)
1.1	2017-03-31	• New/updated text modules see columns “Change” in tables 4..6 of errata sheet V1.1 • Removed reference to “GTM-IP 210 Errata Sheet” in Table 1 - all GTM errata relevant for this design step are considered in this TC23x errata sheet • The following text modules have been included in Table 3 (Errata fixed in this step): - ADC_TC.P007 (Additional Parameter for Data Sheet: Wakeup Time t_{WU}): see specification of t_{WU} in TC23x AC-Step Data Sheet - I0_TC.P003 (Calculating the 1.3 V Current Consumption for TC23x): see corresponding section in TC23x AC-Step Data Sheet - PADS_TC.P007 (Connection of Ball U17 in LFBGA-292 Package): see chapter “Package and Pinning Definitions” in TC23x AC-Step Data Sheet
1.2	2017-11-03	• Update: new/updated text modules see columns “Change” in tables 4..6 of errata sheet V1.2

Revision history

Document version	Date of release	Description of changes
1.3	2018-06-11	• New/updated text modules see columns “Change” in tables 4..6 of errata sheet V1.3 • Replaced: - DMA_TC.029 (DMA Double Buffering Overflow), - DMA_TC.047 (DMA Double Buffering Buffer Switch), - DMA_TC.057 (Double Buffering Overflow Causes Other Channel Corruption) - >> replaced by DMA_TC.061 (DMA Double Buffering Operations) • Removed: - GTM_TC.010 (Effects of GTM Resets) - TC23x..TC21x do not have GTM SRAM
1.4	2019-08-30	• Update: new/updated text modules see columns “Change” in tables 4..6 of errata sheet V1.4
1.5	2020-11-06	• Update: new/updated text modules see columns “Change” in tables 4..6 of errata sheet V1.5
1.6	2022-07-04	• Update: new/updated text modules see columns “Change” in tables 4..6 of errata sheet V1.6
1.7	2023-08-14	• Update: new/updated text modules see columns “Change” in tables 4..6 of errata sheet V1.7 • Removed: - ASCLIN_TC.012: does not apply to design implementation in AURIX™ family - ASCLIN_TC.H007: does not apply to ASCLIN implementation in TC2xx

Document version	Date of release	Description of changes
2.0	2024-06-28	For new and changed errata see also column "Change" in tables 2, 3, and 4. New: • GTM_AI.410, GTM_AI.517, GTM_AI.H519, GTM_AI.522, GTM_AI.H803, MultiCAN_TC.041, QSPI_TC.H011, SCU_TC.H028, SENT_TC.H009 Update to latest errata sheet document template (details see below). Following editorial changes were applied to several (not all) errata (examples): • Misspellings, typos, and case sensitivity • Aligned with latest Infineon writing guidelines • Added 'Description' section title when missing • Added 'TM' where missing (e.g. TriCore™) • Standard footnote numbers are incremented over the entire document (and not per erratum). Table footnotes are numbered per table When an erratum is used by different families or devices, the erratum is now identical in all errata sheets. Differences between the different families or devices are clearly highlighted in the erratum. No update of technical content. TC4xx family specific content has been removed: • ADC_TC.068, ADC_TC.H011, BROM_TC.008, BROM_TC.H003, CCU_TC.H001, CCU_TC.H005, FlexRay_TC.H002, FPI_TC.H002, GTM_AI.141, GTM_AI.142, GTM_AI.260, GTM_AI.298, GTM_AI.299, GTM_TC.H004, GTM_TC.H007, GTM_TC.H015, IDD_TC.H001, MTU_TC.H008, MTU_TC.H009, MultiCAN_TC.043, MultiCAN_TC.044, MultiCAN_TC.H008, MultiCAN_TC.H010, PLL_TC.007, PMS_TC.H008, QSPI_TC.017, RESET_TC.H004, RTH_TC.H001, SENT_TC.H005, SMU_TC.012, SRI_TC.003, SRI_TC.H001, STM_TC.H004 Section 'Severity' is removed from following errata: • FlexRay_AI.104, FlexRay_AI.105, FlexRay_AI.106

Table 67 **Errata fixed in this step**

Errata	Short description	Change
ADC_TC.P007	Additional Parameter for Data Sheet: Wakeup Time t_{WU}	Fixed
CCU_TC.002	Clock Monitors - Target Monitoring Frequency Selection	Fixed
CCU_TC.003	Maximum Amplitude for Frequency Modulation	Fixed
FLASH_TC.044	Repetitive Erase Suspend Requests on Data Flash	Fixed
I0_TC.P003	Calculating the 1.3 V Current Consumption for TC23x	Fixed
MTU_TC.016	Wrong Address(es) Tracked in Registers ETRRx of TC1.6E CPU0 PSPR and DSPR	Fixed
MultiCAN_AI.047	Transmit Frame Corruption after Protocol Exception (CAN FD only)	Fixed
MultiCAN_TC.043	CAN FD: Idle Condition	Fixed (node 0,1 only)
MultiCAN_TC.044	CAN FD: Missing Hardsync	Fixed (node 0,1 only)
PADS_TC.P007	Connection of Ball U17 in LFBGA-292 Package	Fixed
RESET_TC.007	Unexpected SMU Reset during SSW execution if no HARR requested	Fixed
SMU_TC.005	Unexpected/Incorrect Reset caused by SMU Alarms	Fixed

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2024-06-28

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-nxs1688377548512

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”).

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.