

# Safety Manual for TDA3x Automotive Vision safety-critical Applications Processors

## User's Guide



Literature Number: SPRUIG2  
September 2017

<b>Preface</b> .....	<b>6</b>
<b>1 Introduction</b> .....	<b>7</b>
1.1 Overview .....	7
1.2 Product Scope.....	7
1.2.1 Purpose of the Product and Safety Constraints .....	7
1.2.2 Application Sectors .....	7
1.2.3 Product Overview .....	8
<b>2 TI Standard DSP Development Process</b> .....	<b>11</b>
2.1 Overview.....	11
<b>3 Product Safety Architecture and User Requirements</b> .....	<b>13</b>
3.1 Safety Function Overview .....	13
3.2 Electrical Specifications and Environment Limits .....	13
3.3 Mechanical Environment Constraints.....	13
3.4 Operating Frequency Limits.....	13
3.5 Other Foreseeable Environment Disturbances.....	13
3.6 Safety Application Overview .....	14
3.7 Derivation of Technical Safety Concept Based on Identified Safety Goals.....	16
3.7.1 Front Camera Analytics System (FCA).....	16
3.7.2 Rear View Camera Analytics System (RVC) .....	21
3.7.3 Surround View System (SRV) .....	26
3.7.4 Radar and Camera Fusion System (RAD).....	31
3.7.5 Camera Mirror Replacement System (CMS) .....	36
3.7.6 System and Software Level Diagnostics Details in FMEDA, With Fail Modes .....	41
3.7.7 Operating States .....	83
3.7.8 Management of Errors.....	84
3.8 Discussion About Conceptual Safety Mechanisms and Assumptions of Use.....	84
3.8.1 Power Supply .....	85
3.8.2 Power Management.....	85
3.8.3 Software Diagnostic for Configuration Checks .....	85
3.8.4 Clocks.....	86
3.8.5 Reset.....	87
3.8.6 PRCM and Control Module .....	87
3.8.7 CPU Subsystems.....	88
3.8.8 Network-on-Chip L3 Interconnect Subsystem .....	92
3.8.9 On-Chip RAM Subsystem.....	92
3.8.10 General-Purpose Timer Subsystem .....	94
3.8.11 Interprocessor Communication (IPC).....	94
3.8.12 Serial Peripheral Interface (SPI) .....	95
3.8.13 Controller Area Network.....	95
3.8.14 (LP)DDR2/3 Memory Controller (EMIF) .....	96
3.8.15 Enhanced Direct Memory Access (EDMA) .....	97
3.8.16 Video Input Port (VIP) .....	98
3.8.17 Video Processing Engine (VPE) .....	98
3.8.18 Display Subsystem (DSS).....	98
3.8.19 Inter-Integrated Circuit.....	98

3.8.20	General-Purpose Input/Output (GPIO) .....	99
3.9	Tester On Chip (TesOC) .....	99
3.10	Memory Cyclic Redundancy Check Module .....	100
3.11	Dual-Clock Comparator .....	100
3.12	Error Signaling Module.....	100
3.13	Embedded ADC.....	101
<b>4</b>	<b>Software Diagnostic Library .....</b>	<b>102</b>
4.1	TI Diagnostics Library Architecture .....	102
4.2	ECC Fault Injection Test.....	102
4.3	ADC Tests .....	103
4.4	SPI API.....	103
4.5	CRC API .....	104
4.6	DCC API .....	104
4.7	ESM API .....	106
4.8	DCAN Loopback API .....	107
4.9	QSPI (SPI for Boot in 4-Bit Data Mode) .....	107
4.10	TI AutoSAR TDA3x MCAL Architecture .....	108
4.11	Vision SDK AutoSAR Integration Proposal on TDA3x.....	108
4.12	Vision SDK SBL Concept.....	109
4.13	Vision SDK AutoSAR Boot Sequence .....	109
4.14	Vision SDK AutoSAR Stack Initialization With Validation of Resource Conflicts.....	110
4.15	IPC Concept Between AutoSAR OS and SYS/BIOS .....	111
4.16	SYS/BIOS Data Integrity Check Concept .....	111
4.17	Freedom From Interference and Isolation for Software Components .....	112
4.17.1	Memory Isolation High-Level Concept .....	112
4.17.2	Memory Isolation on DSP .....	114
4.17.3	Memory Isolation on Cortex-M4 .....	115
4.17.4	Memory Isolation on EVE.....	116
<b>5</b>	<b>Dependent Fail Analysis Information.....</b>	<b>118</b>
<b>A</b>	<b>Development Interface Agreement .....</b>	<b>119</b>
A.1	Appointment of Safety Managers.....	119
A.2	Tailoring the Safety Lifecycle.....	119
A.3	Activities Performed by TI .....	119
A.4	Information to Exchange.....	120
A.5	Parties Responsible for Safety Activities .....	120
A.6	Supporting Processes and Tools .....	120
A.7	Supplier Hazard and Risk Assessment .....	121
A.8	Creation of Functional Safety Concept .....	121

## List of Figures

1-1.	High-Level Device Diagram .....	9
2-1.	TI Standard DSP QM Development Process and ISO 26262 Compliant Process (1 of 2) .....	11
2-2.	TI Standard DSP QM Development Process and ISO 26262 Compliant Process (2 of 2) .....	12
3-1.	Front Camera Analytics System Block Diagram .....	14
3-2.	Driver Monitoring System Block Diagram .....	15
3-3.	Surround View and Camera Mirror System Conceptual Block Diagram .....	15
3-4.	Radar and Camera Fusion Block Diagram .....	16
3-5.	Operating States .....	83
4-1.	Diagnostics Library High-Level Architecture .....	102
4-2.	Testing ECC Logic.....	103
4-3.	Using On-Chip ADC for System Monitoring .....	103
4-4.	SPI Loopback Test Mechanism.....	104
4-5.	Using CRC to Check for Data Accuracy .....	104
4-6.	Using DCC for Clock Monitoring (1 of 3) .....	105
4-7.	Using DCC for Clock Monitoring (2 of 3) .....	105
4-8.	Using DCC for Clock Monitoring (3 of 3) .....	106
4-9.	Error Interrupts for System Monitoring in ESM (1 of 3) .....	106
4-10.	Error Interrupts for System Monitoring in ESM (2 of 3) .....	107
4-11.	Error Interrupts for System Monitoring in ESM (3 of 3) .....	107
4-12.	Loopback Mechanism for Testing DCAN IP.....	107
4-13.	Confirming Correctness of Boot Over QSPI (1 of 2) .....	108
4-14.	Confirming Correctness of Boot Over QSPI (2 of 2) .....	108
4-15.	High-Level AutoSAR Integration .....	109
4-16.	Safety-Enabled Boot Sequence.....	109
4-17.	Safety-Enabled Boot Sequence Timing.....	110
4-18.	Safety-Enabled Software Partitioning and Core Ownership .....	110
4-19.	Safety-Enabled Interprocessor Communication (IPC) .....	111
4-20.	Ensuring Critical Safety Data Integrity in SYSBIOS.....	112
4-21.	Ensuring Isolation Between an ASIL and QM Task on DSP (1 of 2) .....	114
4-22.	Ensuring Isolation Between an ASIL and QM Task on DSP (2 of 2) .....	115
4-23.	Ensuring Isolation Between an ASIL and QM Task on M4.....	116
4-24.	Ensuring Isolation Between an ASIL and QM Task on EVE .....	116

## List of Tables

3-1.	FCA Analytics .....	17
3-2.	Front Camera Safety-Critical Components .....	17
3-3.	RVC Analytics .....	21
3-4.	Rear Camera Safety-Critical Components .....	22
3-5.	SRV Analytics.....	26
3-6.	SRV Safety-Critical Components.....	27
3-7.	RAD Analytics .....	32
3-8.	RAD Safety-Critical Components .....	32
3-9.	CMS Analytics .....	37
3-10.	CMS Safety-Critical Component .....	37
3-11.	System Diagnostics .....	42
3-12.	Spinlock States .....	94
4-1.	Implementing FFI for Processor Cores .....	113
A-1.	Activities Performed by TI and Performed by Customer .....	119
A-2.	Product Safety Documentation.....	120
A-3.	Product Safety Documentation Tools and Data Formats.....	120

---

---

---

## Scope

This safety manual for the TDA3x device product family specifies the responsibilities of the user for installation and operation to maintain the desired safety level.

This document contains the following:

- Product scope
  - Purpose of product
  - Intended application sectors
  - Product safety constraints
- Information for each safety-related subsystem
  - Functions, interfaces, and parameters of each safety-related subsystem
  - Safety application overview
  - Lifetime, environment, and application limits
  - Application limits of each safety-related subsystem
  - Built-in device safety logic
  - User operation requirements to maintain the desired ASIL level, including a set of proof tests, diagnostic tests, and test intervals
- Summary of responsibilities of the user to integrate TDA3x device family products into safety system
- Safety-related characteristics
- Terms and definitions
- Document revision status

The following information is documented in the *Safety Analysis Report Summary for the TDA3x Device* (part of the FMEDA xls sheet), and is not repeated in this document:

- Failure rate summary of the SoC estimated at the chip level
- Failure rates and diagnostic coverage for each subsystem
- Assumptions of use in calculation of safety metrics
- Fault model used to estimate device failure rates suitable to enable calculation of customized failure rates
- Quantitative FMEA (also known as FMEDA: Failure Modes, Effects, and Diagnostics Analysis) with detail to the sub-module level of the device, suitable to enable calculation based on customized application of diagnostics

Users should have a general understanding of the TDA3x device product family by reading the [TDA3x Data Manual](#) and the [TDA3x Technical Reference Manual](#). This document is intended to be used in conjunction with the pertinent data sheets, technical reference manuals, and other documentation for the products under development.

For more information regarding the Safety Report, contact your TI sales representative.

## Trademarks

C66x, Embedded Vision Engine, OMAP are trademarks of Texas Instruments.  
Cortex is a registered trademark of ARM Limited.  
network-on-chip is a registered trademark of Arteris, Inc.  
MIPI is a registered trademark of MIPI Alliance, Inc.  
Microsoft is a registered trademark of Microsoft Corporation.

## Introduction

---

---

### 1.1 Overview

You, as a system and equipment manufacturer or designer, must ensure that your systems (and any TI hardware or software components incorporated in your systems) meet all applicable safety, regulatory, and system-level performance requirements. All application and safety-related information in this document (including application descriptions, suggested safety measures, suggested TI products, and other materials) is provided for reference only. You understand and agree that your use of TI components in safety-critical applications is entirely at your risk, and that you (as the buyer) agree to defend, indemnify, and hold harmless TI from any and all damages, claims, suits, or expense resulting from such use.

This safety manual provides information needed by system developers to assist in creating a safety-critical system using a TDA3x device. TI does not claim any compliance to any industry safety standard or Automotive Safety Integrity Level at the system level (ASIL). The devices of this family are targeted to support systems which are QM, ASIL A, and ASIL B. The safety manual specifies the responsibilities of the user for installation and operation to maintain the desired safety level.

For more information regarding the product documentation, contact your TI sales representative.

### 1.2 Product Scope

#### 1.2.1 Purpose of the Product and Safety Constraints

The purpose of the TDA3x device is to function as a digital signal processor (DSP) in embedded automotive applications in the driver assistance space. Some of these applications may be safety critical.

Multiple safety applications were analyzed during the concept and design phase for this product, to support Safety Element Out of Context (SEoC) development according to ISO 26262-10:2011. The product was developed per ISO 26262 standard processes. You, as the system and equipment manufacturer or designer, must ensure that your systems (and any TI hardware or software components incorporated in your systems) meet all applicable safety, regulatory, and system-level performance requirements.

TI provides safety manuals, safety analysis reports (part of FMEDA), configurable FMEDAs, and certificates of compliance to enable customers to perform item (system) level safety analysis. No assumptions have been made in the FMEDA for whether a particular item uses a block of TDA3x. This document includes the failure rates for all the blocks, without making any assumptions on the use of that particular block for a given system. The customer must use the reports provided in context of their own usage and system requirements. This safety manual highlights five use cases that customers can use as a starting point for their safety concept. These use cases should be treated as examples only, for TI customers when they use the FMEDA tool and other collateral to measure metrics and perform their own due diligence for item-level ASIL assessment.

The TDA3x device was developed according to ISO 26262 safety standards, to ease customer adoption of the product for safety applications; however, no compliance to these standards is claimed.

MidTDA3x silicon is targeted for a variety of safety applications, thus there is no fixed FTTI that TI as a supplier could propose. It is the responsibility of the customer to use the methods and safety features outlined in the safety manual and the device TRM to meet their own identified FTTI requirements.

#### 1.2.2 Application Sectors

The TDA3x device is intended to be used in automotive Advanced Driver Assistance Systems (ADAS). Specific, targeted-application segments include, but are not limited to the following:

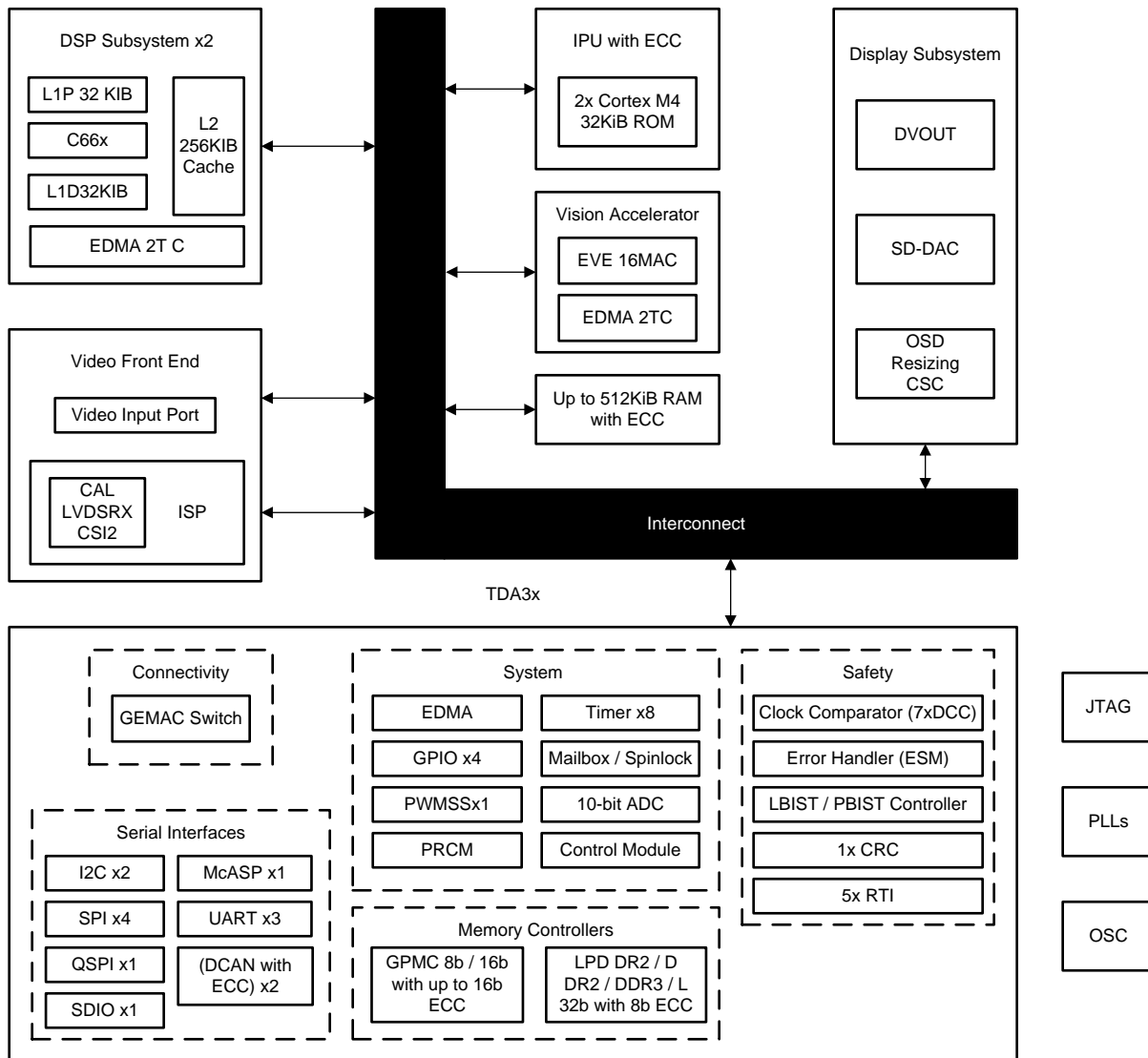
- Front camera
  - Lane departure warning
  - Traffic sign recognition
  - High beam assist
  - Collision mitigation
- Backup camera
  - Obstacle detection
  - Park assist
- Surround View Systems
  - Ethernet surround view
  - LVDS surround view
- Radar
  - Long-range radar
  - Short-range radar
- Mirror replacement

TI has designed the TDA3x device and corresponding safety documentation to be applicable to applications beyond those mentioned above. In these cases, special attention is required when following the requirements specified in this document, to ensure the correct use of this safety information.

### 1.2.3 Product Overview

[Figure 1-1](#) provides a high-level overview of the TDA3x device.





Consult the individual TRM of the device you use to identify the correct configuration for the module in the block diagram.

**Figure 1-1. High-Level Device Diagram**

The device is composed of the following main subsystems:

- Up to DSP C66x™ subsystems
- Embedded Vision Engine™ (EVE) accelerator subsystem
- Dual Cortex®-M4 microprocessor unit subsystem
- Video input capture port (VIP)
- Display subsystem (DSS)
- Imaging subsystem (ISS) with imaging signal processor (ISP)
- Debug subsystem

The device provides a rich set of connectivity peripherals, including the following among others:

- Camera parallel interface (CPI)
- Camera serial interface (MIPI® CSI-2)
- Display parallel 24-bit output (for MIPI DPI 2.0 or BT.656/BT.1120 support), and one NTSC/PAL standard definition digital-to-audio converter (DAC) (for composite video support)

- Gigabit Ethernet (GEMAC) subsystem
- Two DCAN subsystems
- SDIO controller
- QSPI module
- McASP module
- PWMSS module

The device also integrates the following features:

- On-chip memory
- External memory interfaces
- Memory management
- Level 3 (L3) and level 4 (L4) interconnects
- System and serial control peripherals
- Phase-locked loop (PLL) for internal clock generation

The device includes comprehensive support for the following functional safety system requirements:

- Dual-core ECC-protected M4
- ECC-protected 32-bit DDR interface
- Error detection and correction:
  - Parity bit per byte on C66x DSP level 1 (L1) program cache and single-error correction dual-error detection (SECDED) on level 2 (L2) memories
- SECDED on large L3 on-chip RAM
- Dedicated memory management units per CPU (Cortex-M4 MCU, C66x DSP, and EDMA) to implement freedom from interference
- Memory protection units inside Cortex-M4 MCU subsystem allow freedom from interference
- Memory protection units internal to DSP cache controller allow freedom from interference
- Firewalls to enable isolation and inadvertent access
- Real-time interrupt (RTI) module supports a windowed watchdog feature
- Dedicated hardware accelerated CRC block
- Embedded CRC handles MIPI pixel data on CSI-2 interface
- Two C66x DSP subsystems for redundant calculations
- MISR test schemes for EVE
- Parity-protected EVE internal memories on minimum access size granularity
- LBIST controller for testing device processors and PBIST controller for all on-chip memories
- ESM module to enable error monitoring off-chip
- DCC module for on-line clock monitoring
- Temperature monitoring sensors
- Embedded 8-channel analog-to-digital converter (ADC) for system monitoring

The ADAS TDA3x device is a high-performance, automotive, vision application device, based on enhanced OMAP™ architecture integrated on a 28-nm technology.

## TI Standard DSP Development Process

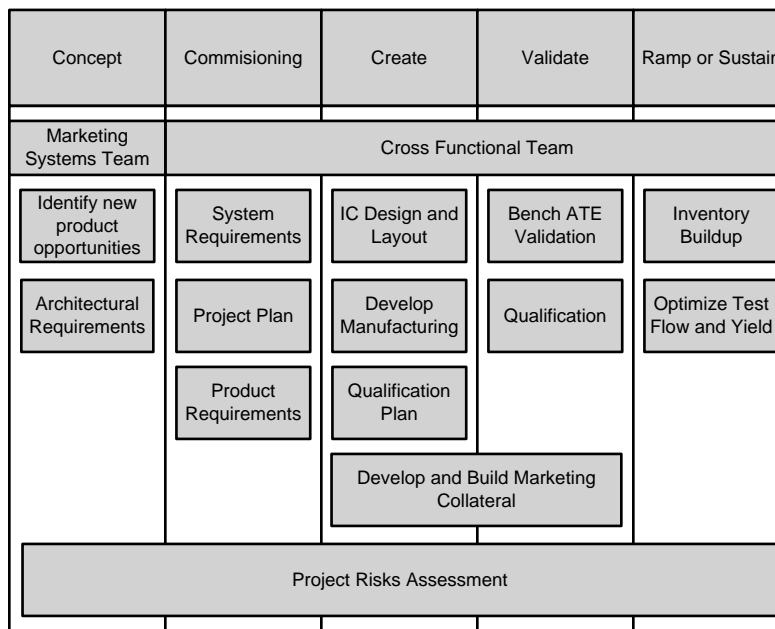
### 2.1 Overview

TI has been developing products for the automotive applications for more than twenty years. Automotive markets have strong requirements on quality management and high-reliability of products. Though not explicitly developed for compliance to a functional safety standard, the TI standard DSP development process already features many elements necessary to manage systematic faults. This development process can be considered to be quality managed (QM), but it does not achieve an IEC 61058 Safety Integrity Level (SIL) or ISO 26262 Automotive Safety Integrity Level (ASIL).

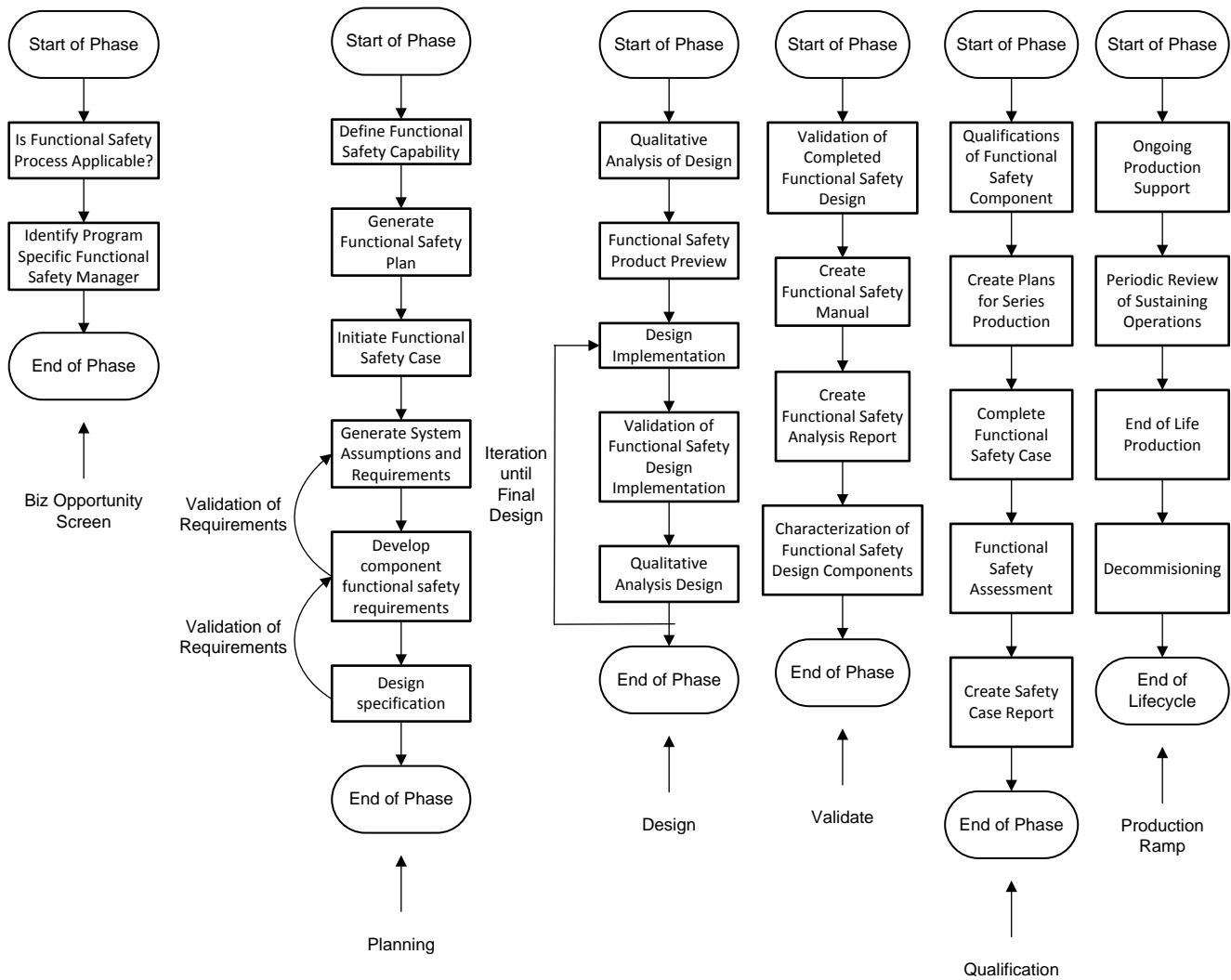
The standard process breaks development into the following phases:

- Concept
- Commissioning
- Create
- Validation
- Ramp

In addition to following the standard development process, TDA3x has also followed a functional safety certified development process based on ISO 26262. TI engaged with an external assessor (Exida) for evaluating whether the practices followed during the development processes meet the requirements of ISO 26262. Summary phases for both of the development processes followed are illustrated in [Figure 2-1](#) and [Figure 2-2](#).



**Figure 2-1. TI Standard DSP QM Development Process and ISO 26262 Compliant Process (1 of 2)**



**Figure 2-2. TI Standard DSP QM Development Process and ISO 26262 Compliant Process (2 of 2)**

## Product Safety Architecture and User Requirements

---

---

### 3.1 Safety Function Overview

Assume a useful lifetime, based on experience. The failure rate stated in the FMEDA only applies within the useful lifetime of the component. Beyond the useful lifetime, the result of the probabilistic calculation method is therefore meaningless, because the probability of failure significantly increases with time. The useful lifetime is highly dependent on the subsystem and its operating conditions.

This assumption of a constant failure rate is based on the bathtub curve.

The TDA3x does not have any significant factors that are known to limit the useful lifetime. A 10-year useful lifetime can be assumed as a conservative measure. The manufacturer of a safety product using the TDA3x device must determine any useful lifetime limitations of other components used in the product design and disclose them in their product specific FMEDA and user manual.

When experience indicates a shorter useful lifetime than indicated in this section, use the number based on actual experience.

### 3.2 Electrical Specifications and Environment Limits

The Recommended Operating Conditions section in the [TDA3x Data Manual](#) and the PRCM Subsystem Environment section in the [TDA3x TRM](#) specify the environmental conditions.

The user must ensure these constraints are kept. Only the Q version device should be considered in the Automobile industry.

### 3.3 Mechanical Environment Constraints

The constraints on mechanics, humidity, and temperature in the reflow oven are stated in AEC-Q100 Rev G. The user must ensure these constraints are kept.

### 3.4 Operating Frequency Limits

The Operating Performance Points section in the [TDA3x Data Manual](#) specifies the operating performance point for processor clocks and device core clocks. The user must ensure these constraints are kept.

### 3.5 Other Foreseeable Environment Disturbances

- Radiation

All the soft error rate (SER) calculations in the FMEDA are based on the NYC sea-level cosmic particles and ultra-low alpha emission package. The soft error caused by the cosmic particles must be reevaluated if this product applies to a higher altitude. If necessary for your product safety analysis, contact your TI sales representative for more details on failure rates at higher altitudes. This product is not allowed to be used in outer space safety applications. Failures due to protons and other heavy nuclei must be considered in outer space. All TI failure rates driven by alpha particles are based upon the use of an ultra-low alpha mold compound in packaging (<0.002 alpha / (cm<sup>2</sup>hr)).

- EM immunity

The ESD tolerance of this product is documented in the Absolute Maximum Ratings section of the [TDA3x Data Manual](#). Any ESD event higher than this limit may cause permanent damage to the IC. The user must implement measures to prevent ESD events higher than the ESD tolerance limit. This product should not be exposed under any high frequency (<1 MHz) electric field larger than 400 V/m or any magnetic field larger than 1.07 A/m. Exposure to such a field may cause communication interrupt, soft error, hard error (can be recovered by a power cycle), or permanent damage to the IC. The user

must implement shielding, grounding, filtering, or other methods to ensure the electromagnetic field strength is under the limits previously mentioned.

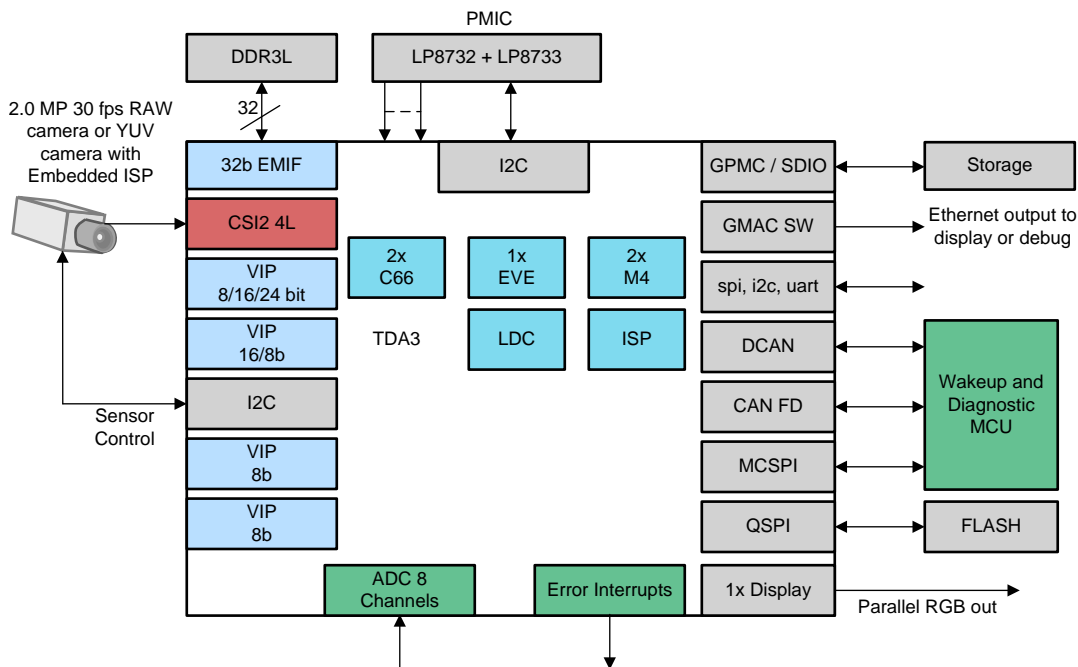
### 3.6 Safety Application Overview

The TDA3x device is intended for use in a number of safety-critical automotive applications, such as:

- Front camera analytics system
- Rear view camera analytics system
- Surround view system
- Radar system
- Radar and camera fusion processing system
- Driver monitoring system
- Camera mirror replacement system
- And more

The purpose of this overview is to help you as a customer use the diagnostic features available in TDA3x silicon and supporting software, and achieve the required system-level ASIL performance rating. The figures below illustrate how the available sensors and varied integrations of supporting logic can result in multiple device interfaces, which must be considered safety critical to comprehend all common system implementations.

Figure 3-1 shows how a front camera analytics system can be realized using the TDA3x device.



**Figure 3-1. Front Camera Analytics System Block Diagram**

Figure 3-2 shows how a driver monitoring system could be realized using the TDA3x device.

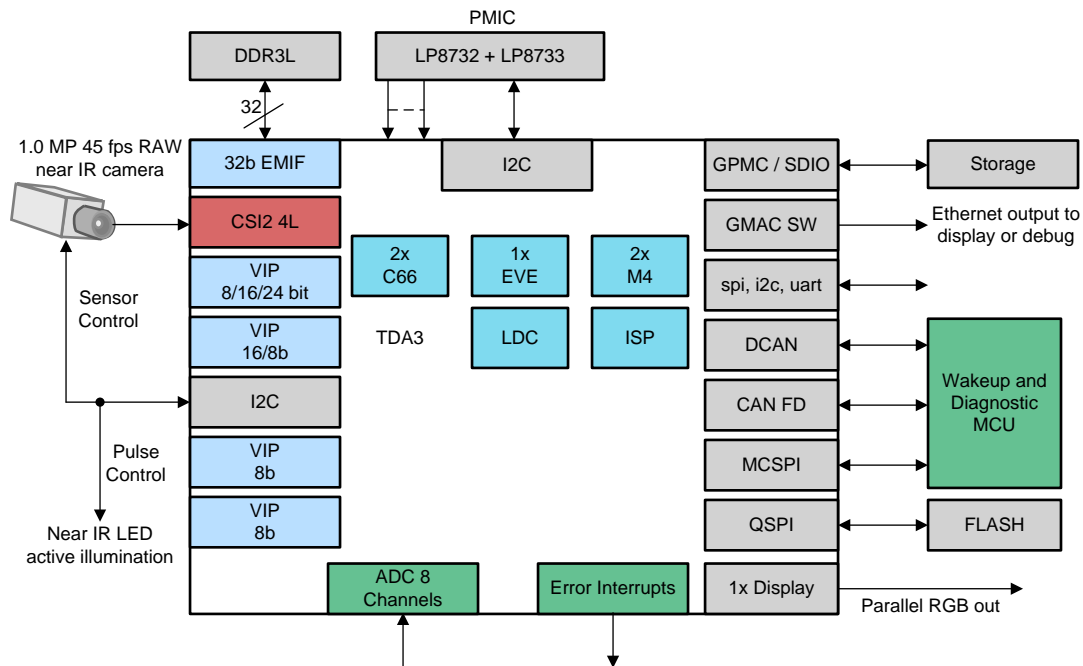


Figure 3-2. Driver Monitoring System Block Diagram

Figure 3-3 shows how a surround view system or CMS (Camera Mirror System) could be realized using the TDA3x device. A camera mirror system typically does not use fisheye lenses, and may require higher frame rates of video. The CMS system also typically uses 1 or 2 cameras, unlike the surround view system that requires 4.

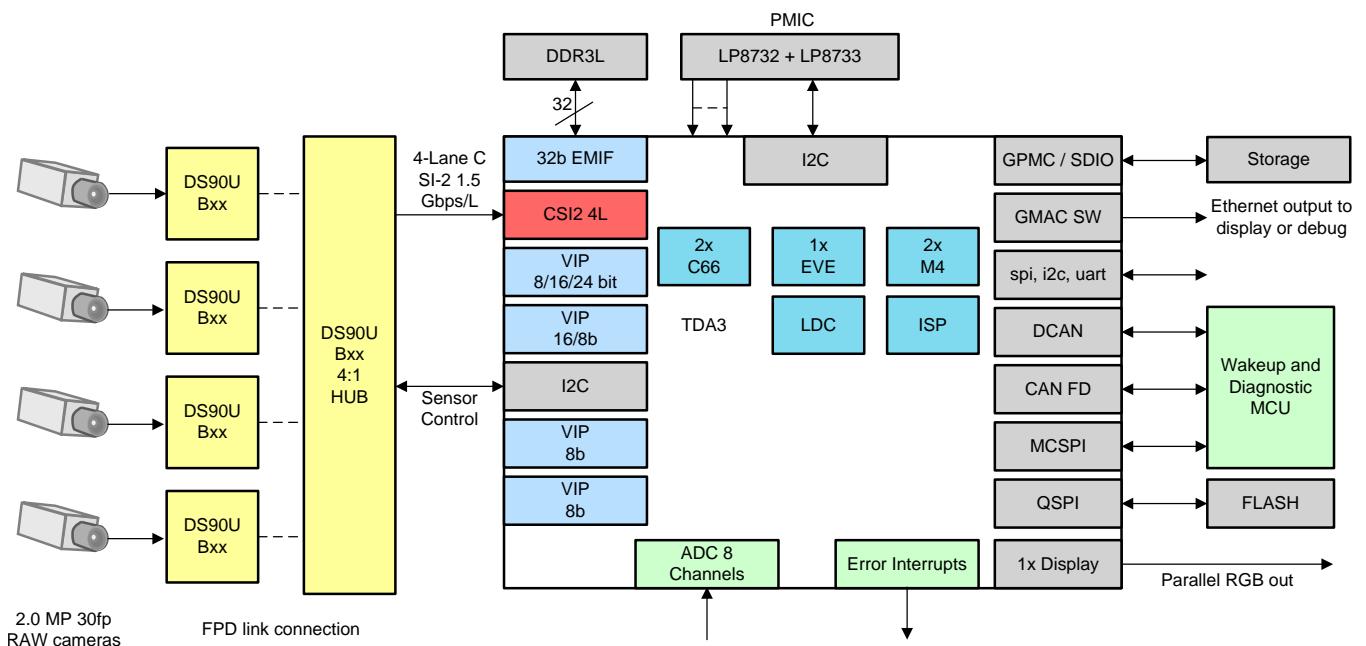


Figure 3-3. Surround View and Camera Mirror System Conceptual Block Diagram

Figure 3-4 shows how a radar and camera fusion processing system can be realized using the TDA3x device. If all the radar (AWR1x) and camera devices except one AWR1x are removed from Figure 3-4, then it is a simple example of an MRR or LRR radar processing system.

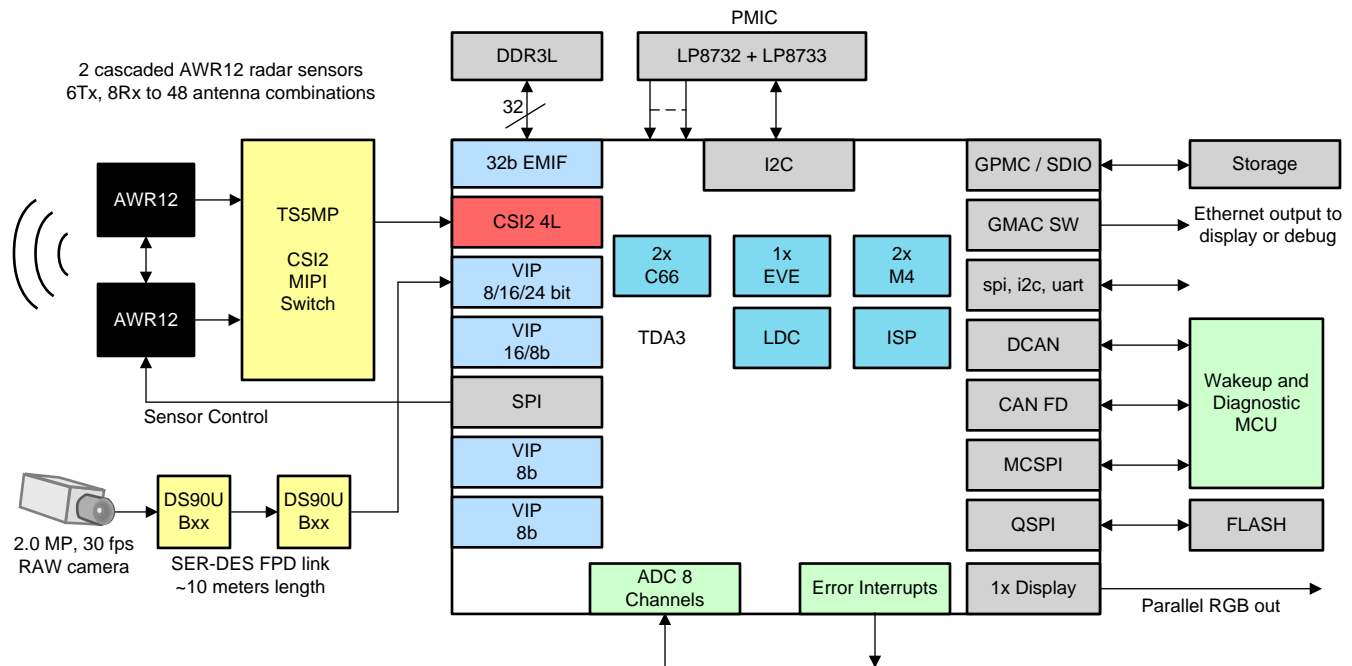


Figure 3-4. Radar and Camera Fusion Block Diagram

### 3.7 Derivation of Technical Safety Concept Based on Identified Safety Goals

System integrators are responsible for the safety analysis at the item level for all systems. This analysis includes identification of the safety goals through formal HARA (HAZard and Risk Analysis) methods. Using severity, exposure, and controllability factors, hazardous events can be assigned ASIL levels and safety goals can be derived. When safety goals are identified, fail modes that can potentially violate the safety goals must be identified. This typically creates a functional safety concept where software- or hardware-based detection mechanisms must be designed and assigned to detect violation of the safety goals, as a result of the variety of fail modes that have been identified.

This analysis does not lay any claim toward thoroughness or comprehensive detailing of failure types that may be systematic or otherwise. As a customer and system designer, you must ensure thorough safety analysis, development of the required safety concept, and assessment of applicability of any recommendations in the safety manual, FMEDA, or any other TI document, to achieve a robustness and high ASIL rating for your part.

#### 3.7.1 Front Camera Analytics System (FCA)

Such analysis may lead to identification of some of the following sample safety goals and fail modes for FCA system. Table 3-1 lists the ways in which a typical technical safety concept may lead to identification of certain modules on the processor as safety critical.



**Table 3-1. FCA Analytics**

Sample Safety Goal / Identified ASIL Level	Potential Transient Fail Modes That May Violate the Safety Goal	Technical Safety Concept Derivation to Detect Safety Goal Violation
Lane detection works properly to identify lanes	Obstructed camera lens	Monitor mechanical obstructions
	Power supply failure, Temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that is synchronized with the lane detection system
Automatic emergency braking system does not work in case of an imminent pedestrian collision / ASIL xx The system does not activate false alarms.	Obstructed camera lens	Monitor mechanical obstructions
	Power supply failure, temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that is synchronized with the lane detection system (in case of a warning-only system)

Based on such an analysis, in general the following components in [Table 3-2](#) may be considered safety critical in this safety analysis for the front camera analytics system for the TDA3x device. The user must perform a detailed analysis based on the actual system implementation, device TRM, and FMEDA.

**Table 3-2. Front Camera Safety-Critical Components**

Hierarchical Part Level	Safety-Related Element
<b>Mission - Digital SRAM</b>	
ADCSS	Yes
DCAN1	Yes
DEBUGSS_hp	No
DEBUGSS_hd	No
MMC	No
TESOC	No
BENELLI unicache_data	Yes
BENELLI unicache_tag	Yes
BENELLI L2	Yes
MCAN	Yes
OCCM	Yes
OCP_WP_NOC	No
GMAC_hp	No
GMAC_hd	No
UART1	No
UART2	No
UART3	No
DSS	No
ISS_hp	Yes
ISS_hd	Yes
EDMA tpcc	Yes

**Table 3-2. Front Camera Safety-Critical Components (continued)**

Hierarchical Part Level	Safety-Related Element
EDMA tptc1	Yes
EDMA tptc2	Yes
VIP_hp	No
VIP_hd	No
EVE dmem	Yes
EVE ibuf	Yes
EVE wbuf	Yes
EVE pcache	Yes
EVE pacahe_tag	Yes
EVE edma tpcc	Yes
EVE edma tptc	Yes
TURING1 edma tpcc	Yes
TURING1 edma tptc	Yes
TURING1 L2	Yes
TURING1 L1D	Yes
TURING1 L1P	Yes
TURING1 umc/pmc/dmc	Yes
TURING2 edma tpcc	Yes
TURING2 edma tptc	Yes
TURING2 L2	Yes
TURING2 L1D	Yes
TURING2 L1P	Yes
TURING2 umc/pmc/dmc	Yes
<b>Mission – Digital Logic</b>	
i_adas_io_pad	Yes
u_vd_core_logic.ADCSS_inst	Yes
u_vd_core_logic.ATB_ASYNC_	No
u_vd_core_logic.CAMERARX_inst	Yes
u_vd_core_logic.CM_CORE_AON_inst	Yes
u_vd_core_logic.CM_CORE_inst	Yes
u_vd_core_logic.COUNTER_32K_inst	Yes
u_vd_core_logic.CTRL_MODULE_CORE_inst	Yes
u_vd_core_logic.CTRL_MODULE_WKUP_inst	Yes
u_vd_core_logic.CUST_EFUSE_inst	Yes
u_vd_core_logic.DAC_inst	No
u_vd_core_logic.DCAN1_inst	Yes
u_vd_core_logic.DCC1_inst	Yes
u_vd_core_logic.DCC2_inst	Yes
u_vd_core_logic.DCC3_inst	Yes
u_vd_core_logic.DCC4_inst	Yes
u_vd_core_logic.DCC5_inst	Yes
u_vd_core_logic.DCC6_inst	Yes
u_vd_core_logic.DCC7_inst	Yes
u_vd_core_logic.DEBUGSS_ATCLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_ICLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_inst	No
u_vd_core_logic.DIV_32_inst	Yes

**Table 3-2. Front Camera Safety-Critical Components (continued)**

Hierarchical Part Level	Safety-Related Element
u_vd_core_logic.DPLL_CORE_inst	Yes
u_vd_core_logic.DPLL_DDR_inst	Yes
u_vd_core_logic.DPLL_DSP_inst	Yes
u_vd_core_logic.DPLL_GMAC_inst	Yes
u_vd_core_logic.DPLL_PER_inst	Yes
u_vd_core_logic.EMIF1_inst	Yes
u_vd_core_logic.ESM_inst	Yes
u_vd_core_logic.GPIO1_inst	Yes
u_vd_core_logic.GPIO2_inst	Yes
u_vd_core_logic.GPIO3_inst	Yes
u_vd_core_logic.GPIO4_inst	Yes
u_vd_core_logic.HSDIVIDER_	Yes
u_vd_core_logic.I2ASYNC_	Yes
u_vd_core_logic.I2C1_inst	Yes
u_vd_core_logic.I2C2_inst	Yes
u_vd_core_logic.ICEMELTER_inst	No
u_vd_core_logic.IEEE1500_2_OCP_inst	No
u_vd_core_logic.INTELLIPHY_EMIF1_inst	Yes
u_vd_core_logic.ISS_ATB_async_bridge_m_inst	No
u_vd_core_logic.L4_WKUP_inst	Yes
u_vd_core_logic.MCASP1_inst	Yes
u_vd_core_logic.MCSPI1_inst	Yes
u_vd_core_logic.MCSPI2_inst	Yes
u_vd_core_logic.MCSPI3_inst	Yes
u_vd_core_logic.MCSPI4_inst	Yes
u_vd_core_logic.MMC_inst	No
u_vd_core_logic.MMC_mdp_inst	No
u_vd_core_logic.PRM_inst	Yes
u_vd_core_logic.RTI1_inst	Yes
u_vd_core_logic.RTI2_inst	Yes
u_vd_core_logic.RTI3_inst	Yes
u_vd_core_logic.RTI4_inst	Yes
u_vd_core_logic.RTI5_inst	Yes
u_vd_core_logic.SC_PWR_CLK_DIV_IO_SC_pd_	Yes
u_vd_core_logic.SMARTREFLEX_CORE_inst	Yes
u_vd_core_logic.SMARTREFLEX_DSPEVE_inst	Yes
u_vd_core_logic.T2ASYNC_CM_CORE_	Yes
u_vd_core_logic.T2ASYNC_L3_	Yes
u_vd_core_logic.T2ASYNC_L4_	Yes
u_vd_core_logic.T2ASYNC_OCP_L4_	Yes
u_vd_core_logic.T2ASYNC_PRM_TO_L3_	Yes
u_vd_core_logic.TESOC_	Yes
u_vd_core_logic.TIMER1_inst	Yes
u_vd_core_logic.TOP_SSYNC_PBIST_	No
u_vd_core_logic.TOP_SYNC_PBIST_	No
u_vd_core_logic.adas_ddr_bscan_inst	No
u_vd_core_logic.adas_efuse_autoload_handler_inst	No

**Table 3-2. Front Camera Safety-Critical Components (continued)**

Hierarchical Part Level	Safety-Related Element
u_vd_core_logic.dftss_adas_low_inst	No
u_vd_core_logic.dma_xbar_inst	Yes
u_vd_core_logic.i_adas_io_bsr	Yes
u_vd_core_logic.intelliphy_	Yes
u_vd_core_logic.ipu_core_reset_	Yes
u_vd_core_logic.tesoc_inst	Yes
u_vd_core_logic.test_glue_adas_inst	Yes
u_vd_core_logic.testcm_adas_wrap_inst	Yes
u_vd_core_logic.top_cdr_dcdr_inst	No
u_vd_core_logic.u_sc_L3	Yes
u_vd_core_logic.u_top_et_	No
u_sc_common.DCAN2_	Yes
u_sc_common.DFT_	No
u_sc_common.ELM_inst	No
u_sc_common.GPMC_inst	No
u_sc_common.HEAD_POS_FOR_ASYNC_BRIDGE_	Yes
u_sc_common.I2ASYNC_	Yes
u_sc_common.L4_CFG_inst	Yes
u_sc_common.L4_PER2_inst	Yes
u_sc_common.L4_PER3_inst	Yes
u_sc_common.L4_PER_inst	Yes
u_sc_common.MAILBOX1_inst	Yes
u_sc_common.MAILBOX2_inst	Yes
u_sc_common.OCMC_DMLED_	No
u_sc_common.OCMC_RAM1_	Yes
u_sc_common.OCP_WP_NOC_	No
u_sc_common.PWMSS1_inst	Yes
u_sc_common.QSPI_inst	Yes
u_sc_common.SC_COMMON_PBIST_	No
u_sc_common.SC_PWR_CLK_DIV_SC_COMMON_pd_	Yes
u_sc_common.SPINLOCK_inst.SPINLOCK_	Yes
u_sc_common.TIMER2_inst	Yes
u_sc_common.TIMER3_inst	Yes
u_sc_common.TIMER4_inst	Yes
u_sc_common.TIMER5_inst	Yes
u_sc_common.TIMER6_inst	Yes
u_sc_common.TIMER7_inst	Yes
u_sc_common.TIMER8_inst	Yes
u_sc_common.UART1_	No
u_sc_common.UART2_	No
u_sc_common.UART3_	No
u_sc_common.sc_common_cdr_dcdr_inst	No
u_sc_common.u_GMAC.GMAC_	Yes
u_sc_common.u_sc_common_et_	No
u_sc_common.ATL_inst	No
u_sc_common.MCAN_	Yes
u_sc_common.MCASP	Yes

**Table 3-2. Front Camera Safety-Critical Components (continued)**

Hierarchical Part Level	Safety-Related Element
u_vd_core_logic.u_sc_benelli	Yes
u_vd_core_logic.u_sc_dss	No
u_vd_core_logic.u_sc_iss	Yes
u_vd_core_logic.u_sc_vip	No
u_vd_dspeve_logic.u_sc_eve	Yes
u_vd_dspeve_logic.u_sc_turing1	Yes
u_vd_dspeve_logic.u_sc_turing2	Yes
misc	Yes
<b>Mission - Memories - ROM</b>	
TESOC	Yes
TOP	Yes
BENELLI	No
BENELLI (pbist)	No
PBIST	No
ISS	No
EVE	No
TURING1	No
TURING2	No
<b>Mission - Analog</b>	
vslido	Yes
vbbldo	Yes
anatop	Yes
afe	Yes
VBGAPTSV2	Yes
RCOSC32K	Yes
HSDIVIDER	Yes
avdac1bgtvv1	No

### 3.7.2 Rear View Camera Analytics System (RVC)

Such analysis may lead to the identification of the following sample safety goals and fail modes for RVC system. [Table 3-3](#) lists ways a typical technical safety concept may lead to identification of certain modules on the processor as safety critical.

**Table 3-3. RVC Analytics**

Sample Safety Goal / Identified ASIL Level	Potential Transient Fail Modes That May Violate the Safety Goal	Technical Safety Concept Derivation to Detect Safety Goal Violation
Object or pet detection system works / ASIL x in variety of environmental conditions	Obstructed camera lens	Monitor mechanical obstructions
	Power supply failure, temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that are synchronized with the lane detection system

**Table 3-3. RVC Analytics (continued)**

Sample Safety Goal / Identified ASIL Level	Potential Transient Fail Modes That May Violate the Safety Goal	Technical Safety Concept Derivation to Detect Safety Goal Violation
Automatic emergency braking system works in case of an imminent collision with a pedestrian, pet, or child / ASIL xx	Obstructed camera lens	Monitor mechanical obstructions
	Power supply failure, temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that are synchronized with the lane detection system
Customer system may require a certain latency for sensor to display	Obstructed lens	Monitor voltage, clock, temperature
	Power supply failure, temperature beyond allowed limits, clock drift	Monitor software execution of the algorithms
	Processor computation failure	Implement time-out mechanisms
	Processor lockout	Wrap critical messages with signatures
	Message corruption for lane detect	Feedback to the system based on speakers/mic that are synchronized with the lane detection system (in case of a warning-only system)
	Audio warning system fails	

Based on such an analysis, in general the following components in [Table 3-4](#) may be considered safety critical in this safety analysis for the rear camera analytics system for the TDA3x device. The user must perform the detailed analysis based on the actual system implementation, device TRM, and FMEDA.

**Table 3-4. Rear Camera Safety-Critical Components**

Hierarchical Part Level	Safety Related Element
<b>Mission - Digital SRAM</b>	
ADCSS	Yes
DCAN1	Yes
DEBUGSS_hp	No
DEBUGSS_hd	No
MMC	No
TESOC	No
BENELLI unicast_data	Yes
BENELLI unicast_tag	Yes
BENELLI L2	Yes
MCAN	Yes
OCCM	Yes
OCP_WP_NOC	No
GMAC_hp	No
GMAC_hd	No
UART1	No
UART2	No
UART3	No
DSS	Yes
ISS_hp	Yes
ISS_hd	Yes
EDMA tpcc	Yes

**Table 3-4. Rear Camera Safety-Critical Components (continued)**

Hierarchical Part Level	Safety Related Element
EDMA tptc1	Yes
EDMA tptc2	Yes
VIP_hp	No
VIP_hd	No
EVE dmem	Yes
EVE ibuf	Yes
EVE wbuf	Yes
EVE pcache	Yes
EVE pacahe_tag	Yes
EVE edma tpcc	Yes
EVE edma tptc	Yes
TURING1 edma tpcc	Yes
TURING1 edma tptc	Yes
TURING1 L2	Yes
TURING1 L1D	Yes
TURING1 L1P	Yes
TURING1 umc/pmc/dmc	Yes
TURING2 edma tpcc	Yes
TURING2 edma tptc	Yes
TURING2 L2	Yes
TURING2 L1D	Yes
TURING2 L1P	Yes
TURING2 umc/pmc/dmc	Yes
<b>Mission - Digital Logic</b>	
i_adas_io_pad	Yes
u_vd_core_logic.ADCSS_inst	Yes
u_vd_core_logic.ATB_ASYNC_	No
u_vd_core_logic.CAMERARX_inst	Yes
u_vd_core_logic.CM_CORE_AON_inst	Yes
u_vd_core_logic.CM_CORE_inst	Yes
u_vd_core_logic.COUNTER_32K_inst	Yes
u_vd_core_logic.CTRL_MODULE_CORE_inst	Yes
u_vd_core_logic.CTRL_MODULE_WKUP_inst	Yes
u_vd_core_logic.CUST_EFUSE_inst	Yes
u_vd_core_logic.DAC_inst	No
u_vd_core_logic.DCAN1_inst	Yes
u_vd_core_logic.DCC1_inst	Yes
u_vd_core_logic.DCC2_inst	Yes
u_vd_core_logic.DCC3_inst	Yes
u_vd_core_logic.DCC4_inst	Yes
u_vd_core_logic.DCC5_inst	Yes
u_vd_core_logic.DCC6_inst	Yes
u_vd_core_logic.DCC7_inst	Yes
u_vd_core_logic.DEBUGSS_ATCLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_ICLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_inst	No
u_vd_core_logic.DIV_32_inst	Yes

**Table 3-4. Rear Camera Safety-Critical Components (continued)**

Hierarchical Part Level	Safety Related Element
u_vd_core_logic.DPLL_CORE_inst	Yes
u_vd_core_logic.DPLL_DDR_inst	Yes
u_vd_core_logic.DPLL_DSP_inst	Yes
u_vd_core_logic.DPLL_GMAC_inst	Yes
u_vd_core_logic.DPLL_PER_inst	Yes
u_vd_core_logic.EMIF1_inst	Yes
u_vd_core_logic.ESM_inst	Yes
u_vd_core_logic.GPIO1_inst	Yes
u_vd_core_logic.GPIO2_inst	Yes
u_vd_core_logic.GPIO3_inst	Yes
u_vd_core_logic.GPIO4_inst	Yes
u_vd_core_logic.HSDIVIDER_	Yes
u_vd_core_logic.I2ASYNC_	Yes
u_vd_core_logic.I2C1_inst	Yes
u_vd_core_logic.I2C2_inst	Yes
u_vd_core_logic.ICEMELTER_inst	No
u_vd_core_logic.IEEE1500_2_OCP_inst	No
u_vd_core_logic.INTELLIPHY_EMIF1_inst	Yes
u_vd_core_logic.ISS_ATB_async_bridge_m_inst	No
u_vd_core_logic.L4_WKUP_inst	Yes
u_vd_core_logic.MCASP1_inst	Yes
u_vd_core_logic.MCSPI1_inst	Yes
u_vd_core_logic.MCSPI2_inst	Yes
u_vd_core_logic.MCSPI3_inst	Yes
u_vd_core_logic.MCSPI4_inst	Yes
u_vd_core_logic.MMC_inst	No
u_vd_core_logic.MMC_mdp_inst	No
u_vd_core_logic.PRM_inst	Yes
u_vd_core_logic.RTI1_inst	Yes
u_vd_core_logic.RTI2_inst	Yes
u_vd_core_logic.RTI3_inst	Yes
u_vd_core_logic.RTI4_inst	Yes
u_vd_core_logic.RTI5_inst	Yes
u_vd_core_logic.SC_PWR_CLK_DIV_IO_SC_pd_	Yes
u_vd_core_logic.SMARTREFLEX_CORE_inst	Yes
u_vd_core_logic.SMARTREFLEX_DSPEVE_inst	Yes
u_vd_core_logic.T2ASYNC_CM_CORE_	Yes
u_vd_core_logic.T2ASYNC_L3_	Yes
u_vd_core_logic.T2ASYNC_L4_	Yes
u_vd_core_logic.T2ASYNC_OCP_L4_	Yes
u_vd_core_logic.T2ASYNC_PRM_TO_L3_	Yes
u_vd_core_logic.TESOC_	Yes
u_vd_core_logic.TIMER1_inst	Yes
u_vd_core_logic.TOP_SSYNC_PBIST_	No
u_vd_core_logic.TOP_SYNC_PBIST_	No
u_vd_core_logic.adas_ddr_bscan_inst	No
u_vd_core_logic.adas_efuse_autoload_handler_inst	No



**Table 3-4. Rear Camera Safety-Critical Components (continued)**

<b>Hierarchical Part Level</b>	<b>Safety Related Element</b>
u_vd_core_logic.dftss_adas_low_inst	No
u_vd_core_logic.dma_xbar_inst	Yes
u_vd_core_logic.i_adas_io_bsr	Yes
u_vd_core_logic.intelliphy_	Yes
u_vd_core_logic.ipu_core_reset_	Yes
u_vd_core_logic.tesoc_inst	Yes
u_vd_core_logic.test_glue_adas_inst	Yes
u_vd_core_logic.testcm_adas_wrap_inst	Yes
u_vd_core_logic.top_cdr_dcdr_inst	No
u_vd_core_logic.u_sc_L3	Yes
u_vd_core_logic.u_top_et_	No
u_sc_common.DCAN2_	Yes
u_sc_common.DFT_	No
u_sc_common.ELM_inst	No
u_sc_common.GPMC_inst	No
u_sc_common.HEAD_POS_FOR_ASYNC_BRIDGE_	Yes
u_sc_common.I2ASYNC_	Yes
u_sc_common.L4_CFG_inst	Yes
u_sc_common.L4_PER2_inst	Yes
u_sc_common.L4_PER3_inst	Yes
u_sc_common.L4_PER_inst	Yes
u_sc_common.MAILBOX1_inst	Yes
u_sc_common.MAILBOX2_inst	Yes
u_sc_common.OCMC_DMLED_	No
u_sc_common.OCMC_RAM1_	Yes
u_sc_common.OCP_WP_NOC_	No
u_sc_common.PWMSS1_inst	Yes
u_sc_common.QSPI_inst	Yes
u_sc_common.SC_COMMON_PBIST_	No
u_sc_common.SC_PWR_CLK_DIV_SC_COMMON_pd_	Yes
u_sc_common.SPINLOCK_inst.SPINLOCK_	Yes
u_sc_common.TIMER2_inst	Yes
u_sc_common.TIMER3_inst	Yes
u_sc_common.TIMER4_inst	Yes
u_sc_common.TIMER5_inst	Yes
u_sc_common.TIMER6_inst	Yes
u_sc_common.TIMER7_inst	Yes
u_sc_common.TIMER8_inst	Yes
u_sc_common.UART1_	No
u_sc_common.UART2_	No
u_sc_common.UART3_	No
u_sc_common.sc_common_cdr_dcdr_inst	No
u_sc_common.u_GMAC.GMAC_	Yes
u_sc_common.u_sc_common_et_	No
u_sc_common.ATL_inst	No
u_sc_common.MCAN_	Yes
u_sc_common.MCASP	Yes

**Table 3-4. Rear Camera Safety-Critical Components (continued)**

Hierarchical Part Level	Safety Related Element
u_vd_core_logic.u_sc_benelli	Yes
u_vd_core_logic.u_sc_dss	Yes
u_vd_core_logic.u_sc_iss	Yes
u_vd_core_logic.u_sc_vip	No
u_vd_dspeve_logic.u_sc_eve	Yes
u_vd_dspeve_logic.u_sc_turing1	Yes
u_vd_dspeve_logic.u_sc_turing2	Yes
misc	Yes
<b>Mission - Memories - ROM</b>	
TESOC	Yes
TOP	Yes
BENELLI	No
BENELLI (pbist)	No
PBIST	No
ISS	No
EVE	No
TURING1	No
TURING2	No
<b>Mission - Analog</b>	
vslido	Yes
vbbldo	Yes
anatop	Yes
afe	Yes
VBGAPTSV2	Yes
RCOSC32K	Yes
HSDIVIDER	Yes
avdac1bgtv1	No

### 3.7.3 Surround View System (SRV)

Such analysis may lead to the identification of the following sample safety goals and fail modes for SRV system. [Table 3-5](#) lists ways a typical technical safety concept may lead to identification of certain modules on the processor as safety critical.

**Table 3-5. SRV Analytics**

Sample Safety Goal / Identified ASIL Level	Potential Transient Fail Mode	Technical Safety Concept Derivation to Detect Safety Goal Violation
Object on the seamlines of the 4 cameras does not disappear	An obstacle or object in the path of the vehicle that is not considered by the scene merge algorithm disappears in the seam line.	For higher safety rating, introduce another sensor modality
	Power supply failure, Temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that are synchronized with the lane detection system

**Table 3-5. SRV Analytics (continued)**

Sample Safety Goal / Identified ASIL Level	Potential Transient Fail Mode	Technical Safety Concept Derivation to Detect Safety Goal Violation
Object detection or pet detection system works in a variety of environmental conditions / ASIL x	Obstructed camera lens,	Monitor mechanical obstructions
	Power supply failure, Temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that are synchronized with the lane detection system (in case of a warning-only system)
Automatic emergency braking system malfunctions in imminent collision with a pedestrian, pet, or child / ASIL xx	Obstructed camera lens,	Monitor mechanical obstructions
	Power supply failure, Temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that are synchronized with the lane detection system
System may require a certain latency for sensor to display	Obstructed lens	Monitor voltage, clock, temperature
	Power supply failure, Temperature beyond allowed limits, clock drift	Monitor software execution of the algorithms
	Processor computation failure	Implement time-out mechanisms
	Processor lockout	Wrap critical messages with signatures
	Message corruption for lane detect	Feedback to the system based on speakers/mic that are synchronized with the lane detection system
	Audio warning system fails	

Based on such an analysis, in general the following components in [Table 3-6](#) may be considered safety critical in this safety analysis for the SRV analytics system for the TDA3x device. The user must perform the detailed analysis based on the actual system implementation, device TRM, and FMEDA.

**Table 3-6. SRV Safety-Critical Components**

Hierarchical Part Level	Safety Related Element
<b>Mission - Digital SRAM</b>	
ADCSS	Yes
DCAN1	Yes
DEBUGSS_hp	No
DEBUGSS_hd	No
MMC	No
TESOC	No
BENELLI unicache_data	Yes
BENELLI unicache_tag	Yes
BENELLI L2	Yes
MCAN	Yes
OCCM	Yes
OCP_WP_NOC	No
GMAC_hp	No

**Table 3-6. SRV Safety-Critical Components (continued)**

Hierarchical Part Level	Safety Related Element
GMAC_hd	No
UART1	No
UART2	No
UART3	No
DSS	Yes
ISS_hp	Yes
ISS_hd	Yes
EDMA tpcc	Yes
EDMA tptc1	Yes
EDMA tptc2	Yes
VIP_hp	No
VIP_hd	No
EVE dmem	Yes
EVE ibuf	Yes
EVE wbuf	Yes
EVE pcache	Yes
EVE pacache_tag	Yes
EVE edma tpcc	Yes
EVE edma tptc	Yes
TURING1 edma tpcc	Yes
TURING1 edma tptc	Yes
TURING1 L2	Yes
TURING1 L1D	Yes
TURING1 L1P	Yes
TURING1 umc/pmc/dmc	Yes
TURING2 edma tpcc	Yes
TURING2 edma tptc	Yes
TURING2 L2	Yes
TURING2 L1D	Yes
TURING2 L1P	Yes
TURING2 umc/pmc/dmc	Yes
<b>Mission - Digital Logic</b>	
i_adas_io_pad	Yes
u_vd_core_logic.ADCSS_inst	Yes
u_vd_core_logic.ATB_ASYNC_	No
u_vd_core_logic.CAMERARX_inst	Yes
u_vd_core_logic.CM_CORE_AON_inst	Yes
u_vd_core_logic.CM_CORE_inst	Yes
u_vd_core_logic.COUNTER_32K_inst	Yes
u_vd_core_logic.CTRL_MODULE_CORE_inst	Yes
u_vd_core_logic.CTRL_MODULE_WKUP_inst	Yes
u_vd_core_logic.CUST_EFUSE_inst	Yes
u_vd_core_logic.DAC_inst	No
u_vd_core_logic.DCAN1_inst	Yes
u_vd_core_logic.DCC1_inst	Yes
u_vd_core_logic.DCC2_inst	Yes
u_vd_core_logic.DCC3_inst	Yes

**Table 3-6. SRV Safety-Critical Components (continued)**

Hierarchical Part Level	Safety Related Element
u_vd_core_logic.DCC4_inst	Yes
u_vd_core_logic.DCC5_inst	Yes
u_vd_core_logic.DCC6_inst	Yes
u_vd_core_logic.DCC7_inst	Yes
u_vd_core_logic.DEBUGSS_ATCLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_ICLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_inst	No
u_vd_core_logic.DIV_32_inst	Yes
u_vd_core_logic.DPLL_CORE_inst	Yes
u_vd_core_logic.DPLL_DDR_inst	Yes
u_vd_core_logic.DPLL_DSP_inst	Yes
u_vd_core_logic.DPLL_GMAC_inst	Yes
u_vd_core_logic.DPLL_PER_inst	Yes
u_vd_core_logic.EMIF1_inst	Yes
u_vd_core_logic.ESM_inst	Yes
u_vd_core_logic.GPIO1_inst	Yes
u_vd_core_logic.GPIO2_inst	Yes
u_vd_core_logic.GPIO3_inst	Yes
u_vd_core_logic.GPIO4_inst	Yes
u_vd_core_logic.HSDIVIDER_	Yes
u_vd_core_logic.I2ASYNC_	Yes
u_vd_core_logic.I2C1_inst	Yes
u_vd_core_logic.I2C2_inst	Yes
u_vd_core_logic.ICEMELTER_inst	No
u_vd_core_logic.IEEE1500_2_OCP_inst	No
u_vd_core_logic.INTELLIPHY_EMIF1_inst	Yes
u_vd_core_logic.ISS_ATB_async_bridge_m_inst	No
u_vd_core_logic.L4_WKUP_inst	Yes
u_vd_core_logic.MCASP1_inst	Yes
u_vd_core_logic.MCSPI1_inst	Yes
u_vd_core_logic.MCSPI2_inst	Yes
u_vd_core_logic.MCSPI3_inst	Yes
u_vd_core_logic.MCSPI4_inst	Yes
u_vd_core_logic.MMC_inst	No
u_vd_core_logic.MMC_mdp_inst	No
u_vd_core_logic.PRM_inst	Yes
u_vd_core_logic.RTI1_inst	Yes
u_vd_core_logic.RTI2_inst	Yes
u_vd_core_logic.RTI3_inst	Yes
u_vd_core_logic.RTI4_inst	Yes
u_vd_core_logic.RTI5_inst	Yes
u_vd_core_logic.SC_PWR_CLK_DIV_IO_SC_pd_	Yes
u_vd_core_logic.SMARTREFLEX_CORE_inst	Yes
u_vd_core_logic.SMARTREFLEX_DSPEVE_inst	Yes
u_vd_core_logic.T2ASYNC_CM_CORE_	Yes
u_vd_core_logic.T2ASYNC_L3_	Yes
u_vd_core_logic.T2ASYNC_L4_	Yes

**Table 3-6. SRV Safety-Critical Components (continued)**

Hierarchical Part Level	Safety Related Element
u_vd_core_logic.T2ASYNC_OCP_L4_	Yes
u_vd_core_logic.T2ASYNC_PRM_TO_L3_	Yes
u_vd_core_logic.TESOC_	Yes
u_vd_core_logic.TIMER1_inst	Yes
u_vd_core_logic.TOP_SSYNC_PBIST_	No
u_vd_core_logic.TOP_SYNC_PBIST_	No
u_vd_core_logic.adas_ddr_bscan_inst	No
u_vd_core_logic.adas_efuse_autoload_handler_inst	No
u_vd_core_logic.dftss_adas_low_inst	No
u_vd_core_logic.dma_xbar_inst	Yes
u_vd_core_logic.i_adas_io_bsr	Yes
u_vd_core_logic.intelliphy_	Yes
u_vd_core_logic.ipu_core_reset_	Yes
u_vd_core_logic.tesoc_inst	Yes
u_vd_core_logic.test_glue_adas_inst	Yes
u_vd_core_logic.testcm_adas_wrap_inst	Yes
u_vd_core_logic.top_cdr_dcdr_inst	No
u_vd_core_logic.u_sc_L3	Yes
u_vd_core_logic.u_top_et_	No
u_sc_common.DCAN2_	Yes
u_sc_common.DFT_	No
u_sc_common.ELM_inst	No
u_sc_common.GPMC_inst	No
u_sc_common.HEAD_POS_FOR_ASYNC_BRIDGE_	Yes
u_sc_common.I2ASYNC_	Yes
u_sc_common.L4_CFG_inst	Yes
u_sc_common.L4_PER2_inst	Yes
u_sc_common.L4_PER3_inst	Yes
u_sc_common.L4_PER_inst	Yes
u_sc_common.MAILBOX1_inst	Yes
u_sc_common.MAILBOX2_inst	Yes
u_sc_common.OCMC_DMLED_	No
u_sc_common.OCMC_RAM1_	Yes
u_sc_common.OCP_WP_NOC_	No
u_sc_common.PWMSS1_inst	Yes
u_sc_common.QSPI_inst	Yes
u_sc_common.SC_COMMON_PBIST_	No
u_sc_common.SC_PWR_CLK_DIV_SC_COMMON_pd_	Yes
u_sc_common.SPINLOCK_inst.SPINLOCK_	Yes
u_sc_common.TIMER2_inst	Yes
u_sc_common.TIMER3_inst	Yes
u_sc_common.TIMER4_inst	Yes
u_sc_common.TIMER5_inst	Yes
u_sc_common.TIMER6_inst	Yes
u_sc_common.TIMER7_inst	Yes
u_sc_common.TIMER8_inst	Yes
u_sc_common.UART1_	No

**Table 3-6. SRV Safety-Critical Components (continued)**

Hierarchical Part Level	Safety Related Element
u_sc_common.UART2_	No
u_sc_common.UART3_	No
u_sc_common.sc_common_cdr_dcdr_inst	No
u_sc_common.u_GMAC.GMAC_	Yes
u_sc_common.u_sc_common_et_	No
u_sc_common.ATL_inst	No
u_sc_common.MCAN_	Yes
u_sc_common.MCASP	Yes
u_vd_core_logic.u_sc_benelli	Yes
u_vd_core_logic.u_sc_dss	Yes
u_vd_core_logic.u_sc_iss	Yes
u_vd_core_logic.u_sc_vip	No
u_vd_dspeve_logic.u_sc_eve	Yes
u_vd_dspeve_logic.u_sc_turing1	Yes
u_vd_dspeve_logic.u_sc_turing2	Yes
misc	Yes
<b>Mission - Memories - ROM</b>	
TESOC	Yes
TOP	Yes
BENELLI	No
BENELLI (pbist)	No
PBIST	No
ISS	No
EVE	No
TURING1	No
TURING2	No
<b>Mission - Analog</b>	
vsldo	Yes
vbbldo	Yes
anatop	Yes
afe	Yes
VBGAPTSV2	Yes
RCOSC32K	Yes
HSDIVIDER	Yes
avdac1bgtvv1	No

### 3.7.4 Radar and Camera Fusion System (RAD)

Such analysis may lead to the identification of the following sample safety goals and fail modes for RAD system. [Table 3-7](#) lists ways a typical technical safety concept may lead to identification of certain modules on the processor as safety critical.

**Table 3-7. RAD Analytics**

Sample Safety Goal / Identified ASIL Level	Possible Fail Modes That May Violate the Safety Goal	Technical Safety Concept Derivation to Detect Safety Goal Violation
Different sensor modalities remain synchronized correctly or do not lose synchronization, leading to missed obstacle detection.	An obstacle or object in the path of the vehicle may be missed because of synchronization errors from any sensor.	For higher safety rating, introduce global synchronization mechanisms that are statistically correlated.
	Power supply failure, temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that is synchronized with the lane detection system
Object detection or pet detection system works in a variety of environmental conditions / ASIL x	Obstructed camera lens or a broken radar antenna or PCB connection	Monitor mechanical obstructions
	Power supply failure, temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that is synchronized with the lane detection system
Automatic emergency braking system works in an imminent collision with a pedestrian, pet, or child ASIL xx	Obstructed camera lens or a broken radar antenna or PCB connection	Monitor mechanical obstructions
	Power supply failure, temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that is synchronized with the lane detection system (in case of a warning-only system)

Based on such an analysis, in general the following components in [Table 3-8](#) may be considered safety critical in this safety analysis for the RAD analytics system for the TDA3x device. The user must perform the detailed analysis based on the actual system implementation, device TRM, and FMEDA.

**Table 3-8. RAD Safety-Critical Components**

Hierarchical Part Level	Safety-Related Element
<b>Mission - Digital SRAM</b>	
ADCSS	Yes
DCAN1	Yes
DEBUGSS_hp	No
DEBUGSS_hd	No
MMC	No
TESOC	No
BENELLI unicache_data	Yes
BENELLI unicache_tag	Yes
BENELLI L2	Yes
MCAN	Yes



**Table 3-8. RAD Safety-Critical Components (continued)**

Hierarchical Part Level	Safety-Related Element
OCCM	Yes
OCP_WP_NOC	No
GMAC_hp	No
GMAC_hd	No
UART1	No
UART2	No
UART3	No
DSS	Yes
ISS_hp	Yes
ISS_hd	Yes
EDMA tpcc	Yes
EDMA tptc1	Yes
EDMA tptc2	Yes
VIP_hp	No
VIP_hd	No
EVE dmem	Yes
EVE ibuf	Yes
EVE wbuf	Yes
EVE pcache	Yes
EVE pacache_tag	Yes
EVE edma tpcc	Yes
EVE edma tptc	Yes
TURING1 edma tpcc	Yes
TURING1 edma tptc	Yes
TURING1 L2	Yes
TURING1 L1D	Yes
TURING1 L1P	Yes
TURING1 umc/pmc/dmc	Yes
TURING2 edma tpcc	Yes
TURING2 edma tptc	Yes
TURING2 L2	Yes
TURING2 L1D	Yes
TURING2 L1P	Yes
TURING2 umc/pmc/dmc	Yes
<b>Mission - Digital Logic</b>	
i_adas_io_pad	Yes
u_vd_core_logic.ADCSS_inst	Yes
u_vd_core_logic.ATB_ASYNC_	No
u_vd_core_logic.CAMERARX_inst	Yes
u_vd_core_logic.CM_CORE_AON_inst	Yes
u_vd_core_logic.CM_CORE_inst	Yes
u_vd_core_logic.COUNTER_32K_inst	Yes
u_vd_core_logic.CTRL_MODULE_CORE_inst	Yes
u_vd_core_logic.CTRL_MODULE_WKUP_inst	Yes
u_vd_core_logic.CUST_EFUSE_inst	Yes
u_vd_core_logic.DAC_inst	No
u_vd_core_logic.DCAN1_inst	Yes

**Table 3-8. RAD Safety-Critical Components (continued)**

Hierarchical Part Level	Safety-Related Element
u_vd_core_logic.DCC1_inst	Yes
u_vd_core_logic.DCC2_inst	Yes
u_vd_core_logic.DCC3_inst	Yes
u_vd_core_logic.DCC4_inst	Yes
u_vd_core_logic.DCC5_inst	Yes
u_vd_core_logic.DCC6_inst	Yes
u_vd_core_logic.DCC7_inst	Yes
u_vd_core_logic.DEBUGSS_ATCLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_ICLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_inst	No
u_vd_core_logic.DIV_32_inst	Yes
u_vd_core_logic.DPLL_CORE_inst	Yes
u_vd_core_logic.DPLL_DDR_inst	Yes
u_vd_core_logic.DPLL_DSP_inst	Yes
u_vd_core_logic.DPLL_GMAC_inst	Yes
u_vd_core_logic.DPLL_PER_inst	Yes
u_vd_core_logic.EMIF1_inst	Yes
u_vd_core_logic.ESM_inst	Yes
u_vd_core_logic.GPIO1_inst	Yes
u_vd_core_logic.GPIO2_inst	Yes
u_vd_core_logic.GPIO3_inst	Yes
u_vd_core_logic.GPIO4_inst	Yes
u_vd_core_logic.HSDIVIDER_	Yes
u_vd_core_logic.I2ASYNC_	Yes
u_vd_core_logic.I2C1_inst	Yes
u_vd_core_logic.I2C2_inst	Yes
u_vd_core_logic.ICEMELTER_inst	No
u_vd_core_logic.IEEE1500_2_OCP_inst	No
u_vd_core_logic.INTELLIPHY_EMIF1_inst	Yes
u_vd_core_logic.ISS_ATB_async_bridge_m_inst	No
u_vd_core_logic.L4_WKUP_inst	Yes
u_vd_core_logic.MCASP1_inst	Yes
u_vd_core_logic.MCSPI1_inst	Yes
u_vd_core_logic.MCSPI2_inst	Yes
u_vd_core_logic.MCSPI3_inst	Yes
u_vd_core_logic.MCSPI4_inst	Yes
u_vd_core_logic.MMC_inst	No
u_vd_core_logic.MMC_mdp_inst	No
u_vd_core_logic.PRM_inst	Yes
u_vd_core_logic.RTI1_inst	Yes
u_vd_core_logic.RTI2_inst	Yes
u_vd_core_logic.RTI3_inst	Yes
u_vd_core_logic.RTI4_inst	Yes
u_vd_core_logic.RTI5_inst	Yes
u_vd_core_logic.SC_PWR_CLK_DIV_IO_SC_pd_	Yes
u_vd_core_logic.SMARTREFLEX_CORE_inst	Yes
u_vd_core_logic.SMARTREFLEX_DSPEVE_inst	Yes

**Table 3-8. RAD Safety-Critical Components (continued)**

Hierarchical Part Level	Safety-Related Element
u_vd_core_logic.T2ASYNC_CM_CORE_	Yes
u_vd_core_logic.T2ASYNC_L3_	Yes
u_vd_core_logic.T2ASYNC_L4_	Yes
u_vd_core_logic.T2ASYNC_OCP_L4_	Yes
u_vd_core_logic.T2ASYNC_PRM_TO_L3_	Yes
u_vd_core_logic.TESOC_	Yes
u_vd_core_logic.TIMER1_inst	Yes
u_vd_core_logic.TOP_SSYNC_PBIST_	No
u_vd_core_logic.TOP_SYNC_PBIST_	No
u_vd_core_logic.adas_ddr_bscan_inst	No
u_vd_core_logic.adas_efuse_autoload_handler_inst	No
u_vd_core_logic.dftss_adas_low_inst	No
u_vd_core_logic.dma_xbar_inst	Yes
u_vd_core_logic.i_adas_io_bsr	Yes
u_vd_core_logic.intelliphy_	Yes
u_vd_core_logic.ipu_core_reset_	Yes
u_vd_core_logic.tesoc_inst	Yes
u_vd_core_logic.test_glue_adas_inst	Yes
u_vd_core_logic.testcm_adas_wrap_inst	Yes
u_vd_core_logic.top_cdr_dcdr_inst	No
u_vd_core_logic.u_sc_L3	Yes
u_vd_core_logic.u_top_et_	No
u_sc_common.DCAN2_	Yes
u_sc_common.DFT_	No
u_sc_common.ELM_inst	No
u_sc_common.GPMC_inst	No
u_sc_common.HEAD_POS_FOR_ASYNC_BRIDGE_	Yes
u_sc_common.I2ASYNC_	Yes
u_sc_common.L4_CFG_inst	Yes
u_sc_common.L4_PER2_inst	Yes
u_sc_common.L4_PER3_inst	Yes
u_sc_common.L4_PER_inst	Yes
u_sc_common.MAILBOX1_inst	Yes
u_sc_common.MAILBOX2_inst	Yes
u_sc_common.OCMC_DMLED_	No
u_sc_common.OCMC_RAM1_	Yes
u_sc_common.OCP_WP_NOC_	No
u_sc_common.PWMSS1_inst	Yes
u_sc_common.QSPI_inst	Yes
u_sc_common.SC_COMMON_PBIST_	No
u_sc_common.SC_PWR_CLK_DIV_SC_COMMON_pd_	Yes
u_sc_common.SPINLOCK_inst.SPINLOCK_	Yes
u_sc_common.TIMER2_inst	Yes
u_sc_common.TIMER3_inst	Yes
u_sc_common.TIMER4_inst	Yes
u_sc_common.TIMER5_inst	Yes
u_sc_common.TIMER6_inst	Yes

**Table 3-8. RAD Safety-Critical Components (continued)**

Hierarchical Part Level	Safety-Related Element
u_sc_common.TIMER7_inst	Yes
u_sc_common.TIMER8_inst	Yes
u_sc_common.UART1_	No
u_sc_common.UART2_	No
u_sc_common.UART3_	No
u_sc_common.sc_common_cdr_dcdr_inst	No
u_sc_common.u_GMAC.GMAC_	Yes
u_sc_common.u_sc_common_et_	No
u_sc_common.ATL_inst	No
u_sc_common.MCAN_	Yes
u_sc_common.MCASP	Yes
u_vd_core_logic.u_sc_benelli	Yes
u_vd_core_logic.u_sc_dss	Yes
u_vd_core_logic.u_sc_iss	Yes
u_vd_core_logic.u_sc_vip	No
u_vd_dspeve_logic.u_sc_eve	Yes
u_vd_dspeve_logic.u_sc_turing1	Yes
u_vd_dspeve_logic.u_sc_turing2	Yes
misc	Yes
<b>Mission - Memories - ROM</b>	
TESOC	Yes
TOP	Yes
BENELLI	No
BENELLI (pbist)	No
PBIST	No
ISS	No
EVE	No
TURING1	No
TURING2	No
<b>Mission - Analog</b>	
vsldo	Yes
vbbldo	Yes
anatop	Yes
afe	Yes
VBGAPTSV2	Yes
RCOSC32K	Yes
HSDIVIDER	Yes
avdac1bgtvv1	No

### 3.7.5 Camera Mirror Replacement System (CMS)

Such analysis may lead to the identification of the following sample safety goals and fail modes for the CMS. [Table 3-9](#) lists ways a typical technical safety concept may lead to identification of certain modules on the processor as safety critical.

**Table 3-9. CMS Analytics**

Sample Safety Goal / Identified ASIL Level	Potential Transient Fail Modes That May Violate the Safety Goal	Technical Safety Concept Derivation to Detect Safety Goal Violation
Blind spot detection works in various environmental conditions / ASIL x	Obstructed camera lens	Monitor mechanical obstructions
	Power supply failure, temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that are synchronized with the lane detection system
Automatic emergency braking system works in case of an imminent collision with vehicle in adjacent lane while managing a maneuver / ASIL xx	Obstructed camera lens	Monitor mechanical obstructions
	Power supply failure, Temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that are synchronized with the lane detection system
System may require a certain latency for sensor to display <sup>(1)</sup>	Obstructed lens or broken sensor or non-functioning display	Monitor mechanical obstructions
	Power supply failure, temperature beyond allowed limits, clock drift	Monitor voltage, clock, temperature
	Processor computation failure	Monitor software execution of the algorithms
	Processor lockout	Implement time-out mechanisms
	Message corruption for lane detect	Wrap critical messages with signatures
	Audio warning system fails	Feedback to the system based on speakers/mic that are synchronized with the lane detection system

<sup>(1)</sup> This goal is more restricted for a CMS system because of lower latency requirements (60 fps) compared to other systems.

Based on such an analysis, in general the following components in [Table 3-10](#) may be considered safety critical in this safety analysis for the CMS analytics system for the TDA3x device. The user must perform a detailed analysis based on the actual system implementation, device TRM, and FMEDA.

**Table 3-10. CMS Safety-Critical Component**

Hierarchical Part Level	Safety-Related Element
<b>Mission - Digital SRAM</b>	
ADCSS	Yes
DCAN1	Yes
DEBUGSS_hp	No
DEBUGSS_hd	No
MMC	No
TESOC	No
BENELLI unicache_data	Yes
BENELLI unicache_tag	Yes
BENELLI L2	Yes
MCAN	Yes
OCCM	Yes
OCP_WP_NOC	No

**Table 3-10. CMS Safety-Critical Component (continued)**

Hierarchical Part Level	Safety-Related Element
GMAC_hp	No
GMAC_hd	No
UART1	No
UART2	No
UART3	No
DSS	Yes
ISS_hp	Yes
ISS_hd	Yes
EDMA tpcc	Yes
EDMA tptc1	Yes
EDMA tptc2	Yes
VIP_hp	No
VIP_hd	No
EVE dmem	Yes
EVE ibuf	Yes
EVE wbuf	Yes
EVE pcache	Yes
EVE pacache_tag	Yes
EVE edma tpcc	Yes
EVE edma tptc	Yes
TURING1 edma tpcc	Yes
TURING1 edma tptc	Yes
TURING1 L2	Yes
TURING1 L1D	Yes
TURING1 L1P	Yes
TURING1 umc/pmc/dmc	Yes
TURING2 edma tpcc	Yes
TURING2 edma tptc	Yes
TURING2 L2	Yes
TURING2 L1D	Yes
TURING2 L1P	Yes
TURING2 umc/pmc/dmc	Yes
<b>Mission - Digital Logic</b>	
i_adas_io_pad	Yes
u_vd_core_logic.ADCSS_inst	Yes
u_vd_core_logic.ATB_ASYNC_	No
u_vd_core_logic.CAMERARX_inst	Yes
u_vd_core_logic.CM_CORE_AON_inst	Yes
u_vd_core_logic.CM_CORE_inst	Yes
u_vd_core_logic.COUNTER_32K_inst	Yes
u_vd_core_logic.CTRL_MODULE_CORE_inst	Yes
u_vd_core_logic.CTRL_MODULE_WKUP_inst	Yes
u_vd_core_logic.CUST_EFUSE_inst	Yes
u_vd_core_logic.DAC_inst	No
u_vd_core_logic.DCAN1_inst	Yes
u_vd_core_logic.DCC1_inst	Yes
u_vd_core_logic.DCC2_inst	Yes

**Table 3-10. CMS Safety-Critical Component (continued)**

Hierarchical Part Level	Safety-Related Element
u_vd_core_logic.DCC3_inst	Yes
u_vd_core_logic.DCC4_inst	Yes
u_vd_core_logic.DCC5_inst	Yes
u_vd_core_logic.DCC6_inst	Yes
u_vd_core_logic.DCC7_inst	Yes
u_vd_core_logic.DEBUGSS_ATCLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_ICLK_mdp_inst	No
u_vd_core_logic.DEBUGSS_inst	No
u_vd_core_logic.DIV_32_inst	Yes
u_vd_core_logic.DPLL_CORE_inst	Yes
u_vd_core_logic.DPLL_DDR_inst	Yes
u_vd_core_logic.DPLL_DSP_inst	Yes
u_vd_core_logic.DPLL_GMAC_inst	Yes
u_vd_core_logic.DPLL_PER_inst	Yes
u_vd_core_logic.EMIF1_inst	Yes
u_vd_core_logic.ESM_inst	Yes
u_vd_core_logic.GPIO1_inst	Yes
u_vd_core_logic.GPIO2_inst	Yes
u_vd_core_logic.GPIO3_inst	Yes
u_vd_core_logic.GPIO4_inst	Yes
u_vd_core_logic.HSDIVIDER_	Yes
u_vd_core_logic.I2ASYNC_	Yes
u_vd_core_logic.I2C1_inst	Yes
u_vd_core_logic.I2C2_inst	Yes
u_vd_core_logic.ICEMELTER_inst	No
u_vd_core_logic.IEEE1500_2_OCP_inst	No
u_vd_core_logic.INTELLIPHY_EMIF1_inst	Yes
u_vd_core_logic.ISS_ATB_async_bridge_m_inst	No
u_vd_core_logic.L4_WKUP_inst	Yes
u_vd_core_logic.MCASP1_inst	Yes
u_vd_core_logic.MCSPI1_inst	Yes
u_vd_core_logic.MCSPI2_inst	Yes
u_vd_core_logic.MCSPI3_inst	Yes
u_vd_core_logic.MCSPI4_inst	Yes
u_vd_core_logic.MMC_inst	No
u_vd_core_logic.MMC_mdp_inst	No
u_vd_core_logic.PRM_inst	Yes
u_vd_core_logic.RTI1_inst	Yes
u_vd_core_logic.RTI2_inst	Yes
u_vd_core_logic.RTI3_inst	Yes
u_vd_core_logic.RTI4_inst	Yes
u_vd_core_logic.RTI5_inst	Yes
u_vd_core_logic.SC_PWR_CLK_DIV_IO_SC_pd_	Yes
u_vd_core_logic.SMARTREFLEX_CORE_inst	Yes
u_vd_core_logic.SMARTREFLEX_DSPEVE_inst	Yes
u_vd_core_logic.T2ASYNC_CM_CORE_	Yes
u_vd_core_logic.T2ASYNC_L3_	Yes

**Table 3-10. CMS Safety-Critical Component (continued)**

Hierarchical Part Level	Safety-Related Element
u_vd_core_logic.T2ASYNC_L4_	Yes
u_vd_core_logic.T2ASYNC_OCP_L4_	Yes
u_vd_core_logic.T2ASYNC_PRM_TO_L3_	Yes
u_vd_core_logic.TESOC_	Yes
u_vd_core_logic.TIMER1_inst	Yes
u_vd_core_logic.TOP_SSYNC_PBIST_	No
u_vd_core_logic.TOP_SYNC_PBIST_	No
u_vd_core_logic.adas_ddr_bscan_inst	No
u_vd_core_logic.adas_efuse_autoload_handler_inst	No
u_vd_core_logic.dftss_adas_low_inst	No
u_vd_core_logic.dma_xbar_inst	Yes
u_vd_core_logic.i_adas_io_bsr	Yes
u_vd_core_logic.intelliphy_	Yes
u_vd_core_logic.ipu_core_reset_	Yes
u_vd_core_logic.tesoc_inst	Yes
u_vd_core_logic.test_glue_adas_inst	Yes
u_vd_core_logic.testcm_adas_wrap_inst	Yes
u_vd_core_logic.top_cdr_dcdr_inst	No
u_vd_core_logic.u_sc_L3	Yes
u_vd_core_logic.u_top_et_	No
u_sc_common.DCAN2_	Yes
u_sc_common.DFT_	No
u_sc_common.ELM_inst	No
u_sc_common.GPMC_inst	No
u_sc_common.HEAD_POS_FOR_ASYNC_BRIDGE_	Yes
u_sc_common.I2ASYNC_	Yes
u_sc_common.L4_CFG_inst	Yes
u_sc_common.L4_PER2_inst	Yes
u_sc_common.L4_PER3_inst	Yes
u_sc_common.L4_PER_inst	Yes
u_sc_common.MAILBOX1_inst	Yes
u_sc_common.MAILBOX2_inst	Yes
u_sc_common.OCMC_DMLED_	No
u_sc_common.OCMC_RAM1_	Yes
u_sc_common.OCP_WP_NOC_	No
u_sc_common.PWMSS1_inst	Yes
u_sc_common.QSPI_inst	Yes
u_sc_common.SC_COMMON_PBIST_	No
u_sc_common.SC_PWR_CLK_DIV_SC_COMMON_pd_	Yes
u_sc_common.SPINLOCK_inst.SPINLOCK_	Yes
u_sc_common.TIMER2_inst	Yes
u_sc_common.TIMER3_inst	Yes
u_sc_common.TIMER4_inst	Yes
u_sc_common.TIMER5_inst	Yes
u_sc_common.TIMER6_inst	Yes
u_sc_common.TIMER7_inst	Yes
u_sc_common.TIMER8_inst	Yes



**Table 3-10. CMS Safety-Critical Component (continued)**

Hierarchical Part Level	Safety-Related Element
u_sc_common.UART1_	No
u_sc_common.UART2_	No
u_sc_common.UART3_	No
u_sc_common.sc_common_cdr_dcdr_inst	No
u_sc_common.u_GMAC.GMAC_	Yes
u_sc_common.u_sc_common_et_	No
u_sc_common.ATL_inst	No
u_sc_common.MCAN_	Yes
u_sc_common.MCASP	Yes
u_vd_core_logic.u_sc_benelli	Yes
u_vd_core_logic.u_sc_dss	Yes
u_vd_core_logic.u_sc_iss	Yes
u_vd_core_logic.u_sc_vip	No
u_vd_dspeve_logic.u_sc_eve	Yes
u_vd_dspeve_logic.u_sc_turing1	Yes
u_vd_dspeve_logic.u_sc_turing2	Yes
misc	Yes
<b>Mission - Memories - ROM</b>	
TESOC	Yes
TOP	Yes
BENELLI	No
BENELLI (pbist)	No
PBIST	No
ISS	No
EVE	No
TURING1	No
TURING2	No
<b>Mission - Analog</b>	
vsldo	Yes
vbbldo	Yes
anatop	Yes
afe	Yes
VBGAPTSV2	Yes
RCOSC32K	Yes
HSDIVIDER	Yes
avdac1bgtvv1	No

### 3.7.6 System and Software Level Diagnostics Details in FMEDA, With Fail Modes

TDA3x devices include a varied and rich set of peripherals and processing elements which let customer systems program and run system-level diagnostics. These diagnostics can be designed to achieve higher levels of safety and robustness at the system level. Depending on the system-level safety objectives, development teams can design their own diagnostics as well. The FMEDA tool lets you insert your own diagnostic and expected coverage to calculate the effective ISO 26262 metrics.

FMEDA already suggests the software-level diagnostics provided in [Table 3-11](#). These diagnostics can be designed and deployed at run time. The detailed sequence of the diagnostics should be derived from the TRM. A comprehensive analysis for fail modes of the system, assignment of correct diagnostics, and achievable conservative coverage numbers for a fail modes (either listed in [Table 3-11](#) or identified by system integrator) is the responsibility of the system integrator.

**Table 3-11. System Diagnostics**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
<b>Mission - Digital SRAM</b>					
ADCSS	ADC1	ADC channel fails due to a transient or permanent fault. Dedicate one ADC channel for diagnostic.	Ensure that ADC interpreted data is correct.	Dedicate one ADC channel to a known reference on board. Measure that to confirm that the ADC conversion is working correctly. For 100% coverage, instantiate an external analog mux that could route the known reference to any of the 8 channels.	If the FTTI is 10s of ms, a GPIO and dedicated timer can easily control the external mux to check the required channels. Depending upon the FTTI, the coverage can range from 12% to 99%. For transient faults that require immediate detection, TI recommends using redundancy of channels.
DCAN1	DCAN1, ECC	Operation fails due to a transient or permanent fault. Check DCAN message signature in the driver.	Ensure that the DCAN message buffer has the correct message.	Ensure that the DCAN driver has the protocol in place and checks enabled for message signatures per individual message. Coverage can be improved to 100% by embedding additional CAN messages that contain the signatures for the multiple message data being sent.	Every message can be checked for correctness, hence coverage can be greater than 99%.
DEBUGSS_hp	NA				
DEBUGSS_hd	NA				
MMC	MMC1	Operation fails due to a transient or permanent fault. End-to-end protection on data transferred through MMC interface.	Ensure that the card data transferred on MMC interface is correct.	Ensure that the MMC driver checks for the data signature check interrupt, which is part of the protocol and IP. Follow TRM section 20.4.9 for the sequence.	Every data stream received can be checked for correctness, hence coverage can be greater than 99%.
TESOC	NA				
BENELLI unicache_data	ECC				
BENELLI unicache_tag	ECC				
BENELLI L2	ECC				

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
MCAN	MCAN 1, ECC	Operation fails due to a transient or permanent fault. Check DCAN message signature in the driver.	Ensure that the DCAN message buffer has the correct message.	Ensure that the DCAN driver has the protocol in place and checks enabled for message signatures per individual message. Coverage can be improved to 100% by embedding additional CAN messages that contain the signatures for the multiple message data being sent.	Every message can be checked for correctness, hence coverage can be greater than 99%.
OCCMC	ECC				
OCP_WP_NO C	NA				
GMAC_hp	GMAC 1	Operation fails due to a transient or permanent fault. Check Ethernet message signature in the driver. IP only accesses the memory space that is allowed for it to access.	Ensure that the Ethernet message buffer has the correct message.	Ensure that the Ethernet driver has the protocol in place and checks enabled for message signatures per individual message. Coverage can be improved to 100% by embedding additional Ethernet messages for black channel communication that contain the signatures for the multiple message data being sent. Follow TRM section 19.7.4, 19.7.5, and 19.7.6 for detailed sequences.	Every message can be checked for correctness, hence coverage can be greater than 99%.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
GMAC_hd	GMAC 1	Operation fails due to a transient or permanent fault. Check Ethernet message signature in the driver. IP only accesses the memory space that is allowed for it to access.	Ensure that the Ethernet message buffer has the correct message.	Ensure that the Ethernet driver has the protocol in place and checks enabled for message signatures per individual message. Coverage can be improved to 100% by embedding additional Ethernet messages for black channel communication that contain the signatures for the multiple message data being sent. Follow TRM section 19.7.4, 19.7.5, and 19.7.6 for detailed sequences.	Every message can be checked for correctness, hence coverage can be greater than 99%.
UART1	NA				
UART2	NA				
UART3	NA				
DSS	DSS1	Operation fails due to a transient or permanent fault, as follows: <ul style="list-style-type: none"> <li>• DSS frame freeze</li> <li>• Latency violations beyond acceptable limits</li> <li>• DSS does not read the frame from the correct location and has written the correct frame data.</li> <li>• IP only accesses the memory space that is allowed for it to access.</li> </ul>	Ensure that the DSS operation is correct.	<ol style="list-style-type: none"> <li>1. Ensure that with every new DSS frame a unique signature is created and transmitted. This signature should be checked through DSS write back DMA and a DSP or M4 routine checking for the expected frame values.</li> <li>2. Ensure a timer is part of the latency measurement .</li> <li>3. Confirm the DSS frame on the receive side as well outside the chip and the ECU.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level. This is especially possible for atomic safety goals such as lens-to-lens latency measurement and/or freeze frame detection.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
ISS_hp	ISS1	<p>Operation fails due to a transient or permanent fault. RAW2YUV RGB conversion is correctly done.</p> <ul style="list-style-type: none"> <li>Image sensor configuration is correct.</li> <li>IP only accesses the memory space that is allowed for it to access.</li> <li>IP misses any latency budget deadlines.</li> <li>Check if the functionality [WB RSZ WDR]} is configured correctly and working.</li> <li>Check if the LUTs for ISS operations that are stored in the DDR are not corrupted.</li> </ul>	Ensure that ISS operation is correct.	<ol style="list-style-type: none"> <li>Ensure that for a known signature pixel data, ISP is working correctly for all intended operations within FTTI.</li> <li>Ensure that the timer is part of the continuous latency measurements.</li> <li>Ensure that the sensor configuration is read over I2C frequently (FTTI driven) and compared against the target known good value at that point.</li> <li>Read the configuration of the module and compare against expected values.</li> <li>Create LUT signatures, read the data from RAM at regular intervals, and compare against the signature.</li> </ol>	<p>Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.</p> <p>RAW2YUV conversion is not safety critical, as the impact of single bit flips is not catastrophic on the quality of results. On the other hand, a watchdog is absolutely necessary to ensure that the ISP processing is within the latency range. A signature analyzer for a sample frame is necessary to confirm that all ISP functional modules are working correctly.</p> <p>Configuration must be checked and if necessary reprogrammed, every frame to achieve higher coverage.</p>

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
ISS_hd	ISS1	<p>Operation fails due to a transient or permanent fault. RAW2YUV RGB conversion is incorrectly done.</p> <ul style="list-style-type: none"> <li>Image sensor configuration is incorrect.</li> <li>IP does not restrict accesses the memory space that is allowed for it to access.</li> <li>IP misses any latency budget deadlines.</li> <li>Check if the functionality [WB RSZ WDR] is configured correctly and working.</li> <li>Check if the LUTs for ISS operations that are stored in the DDR are not corrupted.</li> </ul>	Ensure that ISS operation is correct.	<ol style="list-style-type: none"> <li>Ensure that for a known signature pixel data, ISP is working correctly for all intended operations within FTTI.</li> <li>Ensure that the timer is part of the continuous latency measurement s.</li> <li>Ensure that the sensor configuration is read over I2C frequently (FTTI driven) and compared against the target known good value at that point.</li> <li>Read the configuration of the module and compare against expected values.</li> <li>Create LUT signatures, read the data from RAM at regular intervals, and compare against the signature.</li> </ol>	<p>Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.</p> <p>RAW2YUV conversion is not safety critical as the impact of single bit flips is not catastrophic on the quality of results. On the other hand, a watchdog is absolutely necessary to ensure that the ISP processing is within the latency range. A signature analyzer for a sample frame is necessary to confirm that all ISP functional modules are working correctly.</p> <p>Configuration must be checked and if necessary reprogrammed, every frame to achieve higher coverage.</p>

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
EDMA tpcc	EDMA TPCC1	<p>PARAM set of the TPCC (Channel Controller) gets corrupted:</p> <ul style="list-style-type: none"> <li>TPCC has not programmed the correct values in the TPTC program register set.</li> <li>Mapping of DMA events to Param sets gets corrupted.</li> <li>Completion detection or event handling does not complete within expected latency.</li> </ul>	Ensure that the control and configuration of the DMA transfer works correctly.	<ol style="list-style-type: none"> <li>Read the Param set at regular intervals and ensure that it matches the expected golden values.</li> <li>Ensure that a timer controlled by CPU monitors completion of DMA events to compare it against target latency values.</li> <li>Read the configuration of TPCC and either compare it regularly against the expected values &lt;OR&gt; reconfigure with the correct data regularly.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
EDMA tptc1	EDMA TPTC1	Data is corrupted while it passes through the local TPTC buffers and FIFOS, when it is being moved from source address to destination address. Transfer takes much longer than the expected latency.	Ensure that the DMA-based data transfer does not corrupt the data being moved.	Implement black channel communication at higher layers of software as suggested for DCAN, Ethernet, and so on.	Depending upon the implemented black channel communication, coverage of more than 99% can be achieved for diagnostics designed at system level. If black channel communication is not implemented, it is not possible to ensure data correctness for EDMA TPTC-based transfers. In some cases, if data being moved is such that a single bit flip will not make a material difference to the results being computed, then a safety criticality can be derated (pixel data, for example). On the other hand, if the data being moved is safety critical (such as the location of identified objects and other descriptors), it is advisable to design adequate safety layers for the data transfer to ensure coverage of more than 99%. For example, the data could be transferred once by TPTC1 and then again by TPTC2. When a CPU (DSP/EVE) uses the data, a thread should compare the contents of both moved buffers to ensure that computation progresses on good data.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
EDMA tptc2	EDMA TPTC1	Data is corrupted while it passes through the local TPTC buffers and FIFOs, when it is being moved from source address to destination address. Transfer takes much longer than the expected latency.	Ensure that the DMA-based data transfer does not corrupt the data being moved.	Implement black channel communication at higher layers of software as suggested for DCAN, Ethernet, and so on.	Depending upon the implemented black channel communication, coverage of more than 99% can be achieved for diagnostics designed at system level. If black channel communication is not implemented, it is not possible to ensure data correctness for EDMA TPTC-based transfers. In some cases, if data being moved is such that a single bit flip will not make a material difference to the results being computed, then a safety criticality can be derated (pixel data, for example). On the other hand, if the data being moved is safety critical (such as identified objects' location and other descriptors), it is advisable to design adequate safety layers for the data transfer to ensure coverage of more than 99%. For example, the data could be transferred once by TPTC1 and then again by TPTC2. When a CPU (DSP/EVE) uses the data, a thread should compare the contents of both moved buffers to ensure that computation progresses on good data.
VIP_hp	NA				
VIP_hd	NA				
EVE dmem	Parity				
EVE ibuf	ECC				
EVE wbuf	ECC				
EVE pcache	ECC				
EVE pcache_tag	ECC				



**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
EVE edma tpcc	EVETP CC	<p>PARAM set of the TPCC (Channel Controller) gets corrupted:</p> <ul style="list-style-type: none"> <li>TPCC has not programmed the correct values in the TPTC program register set.</li> <li>Mapping of DMA events to Param sets gets corrupted.</li> <li>Completion detection or event handling does not complete within expected latency.</li> </ul>	Ensure that the control and configuration of the DMA transfer works correctly.	<ol style="list-style-type: none"> <li>Read the Param set at regular intervals and ensure that it matches the expected golden values.</li> <li>Ensure that a timer controlled by CPU monitors the completion of DMA events to compare it against target latency values.</li> <li>Read the configuration of TPCC and either compare it regularly against the expected values &lt;OR&gt; reconfigure with the correct data regularly.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
EVE edma tptc	EVETP TC	<p>Data is corrupted while it passes through the local TPTC buffers and FIFOS, when it is being moved from source address to destination address. Transfer takes much longer than the expected latency.</p>	Ensure that the DMA-based data transfer does not corrupt the data being moved.	Implement black channel communication at higher layers of software as suggested for DCAN, Ethernet, and so on.	Depending upon the implemented black channel communication, coverage of more than 99% can be achieved for diagnostics designed at system level. If black channel communication is not implemented, it is not possible to ensure data correctness for EDMA TPTC-based transfers. In some cases, if data being moved is such that a single bit flip will not make a material difference to the results being computed, then a safety criticality can be derated (pixel data, for example). On the other hand, if the data being moved is safety critical (such as identified objects' location and other descriptors), it is advisable to design adequate safety layers for the data transfer to ensure coverage of more than 99%. For example, the data could be transferred once by TPTC1 and then again by TPTC1 but by using a second param set. When a CPU (EVE) uses the data, there a thread should compare the contents of both moved buffers to ensure that computation progresses on good data.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
TURING1 edma tpcc	DSPTP CC	PARAM set of the TPCC (Channel Controller) gets corrupted <ul style="list-style-type: none"> <li>TPCC has not programmed the correct values in the TPTC program register set.</li> <li>Mapping of DMA events to Param sets gets corrupted.</li> <li>Completion detection or event handling does not complete within expected latency.</li> </ul>	Ensure that the control and configuration of the DMA transfer works correctly.	<ol style="list-style-type: none"> <li>Read the Param set at regular intervals and ensure that it matches the expected golden values.</li> <li>Ensure that a timer controlled by CPU monitors the completion of DMA events to compare it against target latency values.</li> <li>Read the configuration of TPCC and either compare it regularly against the expected values &lt;OR&gt; reconfigure with the correct data regularly.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
TURING1 edma tptc	DSPTC TC	Data is corrupted while it passes through the local TPTC buffers and FIFOS, when it is being moved from source address to destination address. Transfer takes much longer than the expected latency.	Ensure that the DMA-based data transfer does not corrupt the data being moved.	Implement black channel communication at higher layers of software as suggested for DCAN, Ethernet, and so on.	Depending upon the implemented black channel communication, coverage of more than 99% can be achieved for diagnostics designed at system level. If black channel communication is not implemented, it is not possible to ensure data correctness for EDMA TPTC-based transfers. In some cases, if data being moved is such that a single bit flip will not make a material difference to the results being computed, then a safety criticality can be derated (pixel data, for example). On the other hand, if the data being moved is safety critical (such as the location of identified objects and other descriptors), it is advisable to design adequate safety layers for the data transfer to ensure high coverage of more than 99%. For example, the data could be transferred once by TPTC1 and then again by TPTC1 but by using a second param set. When a CPU (DSP) uses the data, a thread should compare the contents of both moved buffers to ensure that computation progresses on good data.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
TURING1 L2	ECC	ECC	Data is not corrupted in L2 RAM.	Implement write scrubbing regularly on the L2 regions of DSP RAM. Review section 4.3 of the TRM for sequences to be implemented.	If the scrubbing is implemented, once every FTTI, then coverage of more than 99% can be achieved.
TURING1 L1D	DSPL1 D	Avoid using L1\$ to minimize FIT rate.	Minimize the FIT rate impact of L1\$ in C66x.	Bypass L1\$ in C66x memory configuration.	For the suggested configuration, the coverage is greater than 99% as L1 \$ is fully bypassed.
TURING1 L1P	DSPL1 P	Parity	Minimize the FIT rate impact of L1\$ in C66x.	Ensure that listeners are implemented for parity errors.	For the suggested configuration, the coverage is greater than 99% as L1P \$ memory is implemented using memories with mux factors = 4.
TURING1 umc/pmc/dmc	DSPM C	Reconfigure L2 to minimize FIT rates for umc/dmc/pmc blocks.	Minimize the FIT rate impact of UMC/DMC/PMC memories on C66x operation.	Reconfigure L2 memory as 32KB cache and 256 KB RAM. This will reduce the FIT rate of the memories by 1/8, since the 7/8 memories will not be used/read any more.	For the suggested configuration, the coverage is 87.5% (that is, 7/8th of these memories are used and accessed).
TURING2 edma tpcc	DSPTP CC	<p>PARAM set of the TPCC (Channel Controller) gets corrupted.</p> <ul style="list-style-type: none"> <li>TPCC has not programmed the correct values in the TPTC program register set.</li> <li>Mapping of DMA events to Param sets gets corrupted.</li> <li>Completion detection or event handling does not complete within expected latency.</li> </ul>	Ensure that the control and configuration of the DMA transfer works correctly.	<ol style="list-style-type: none"> <li>Read the Param set at regular intervals and ensure that it matches the expected golden values.</li> <li>Ensure that a timer controlled by CPU monitors the completion of DMA events to compare it against target latency values.</li> <li>Read the configuration of TPCC and either compare it regularly against the expected values &lt;OR&gt; reconfigure with the correct data regularly.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
TURING2 edma tptc	DSPTC TC	Data is not corrupted while it passes through the local TPTC buffers and FIFOs, when it is being moved from source address to destination address. Transfer takes much longer than the expected latency.	Ensure that the DMA based data transfer does not corrupt the data being moved.	Implement black channel communication at higher layers of software as suggested for DCAN, Ethernet, and so on.	Depending upon the implemented black channel communication, coverage of more than 99% can be achieved for diagnostics designed at system level. If black channel communication is not implemented, it is not possible to ensure data correctness for EDMA TPTC based transfers. In some cases, if data being moved is such that a single bit flip will not make a material difference to the results being computed, then a safety criticality can be derated (pixel data, for example). On the other hand, if the data being moved is safety critical (such as the location of identified objects and other descriptors), it is advisable to design adequate safety layers for the data transfer to ensure high coverage of more than 99%. For example, the data could be transferred once by TPTC1 and then again by TPTC1 but by using a second param set. When a CPU (DSP) uses the data, a thread should compare the contents of both moved buffers to ensure that computation progresses on good data.
TURING2 L2	ECC	ECC	Data is not corrupted in L2 RAM.	Implement write scrubbing regularly on the L2 regions of DSP RAM. Review section 4.3 of the TRM for sequences to be implemented.	If the scrubbing is implemented, once every FTTI, then coverage of more than 99% can be achieved.
TURING2 L1D	DSPL1 D	Avoid using L1\$ to minimize FIT rate.	Minimize the FIT rate impact of L1\$ in C66x.	Bypass L1\$ in C66x memory configuration.	For the suggested configuration, the coverage is greater than 99% because L1\$ is fully bypassed.
TURING2 L1P	DSPL1 P	Parity	Minimize the FIT rate impact of L1\$ in C66x.	Ensure that listeners are implemented for parity errors.	For the suggested configuration, the coverage is greater than 99% because L1P\$ memory is implemented using memories with mux factors = 4.
TURING2 umc/pmc/dmc	DSPM C	Reconfigure L2 to minimize FIT rates for umc/dmc/pmc blocks.	Minimize the FIT rate impact of UMC/DMC/PMC memories on C66x operation.	Reconfigure L2 memory as 32-KB cache and 256 KB RAM. This reduces the FIT rate of the memories by 1/8, because the 7/8 memories will not be used/read any more.	For the suggested configuration, the coverage is 87.5% (that is, 7/8th of these memories are used and accessed).
<b>Mission - Digital Logic</b>					

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
i_adas_io_pad	IOPAD 1	Voltage required for the correct I/O pad operation is not available. The I/O pad logical configuration is corrupted.	Ensure that the I/O pads are operating correctly.	Implement voltage monitors on 1.8 V, 1.1 V and/or 3.3 V as required. Ensure that the configuration of the I/O pad mux modes is correct by reading and comparing the programmed register values to a known good signature within FTTI.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%. It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on an I/O pad.
u_vd_core_logi c.ADCSS_inst	ADC1	Operation fails due to a transient or permanent fault. Dedicate one ADC channel for diagnostic.	Ensure that ADC interpreted data is correct.	Dedicate one ADC channel to a known reference onboard. Measure that to confirm that the ADC conversion is working correctly. For 100% coverage, instantiate an external analog mux that could route the known reference to any of the 8 channels.	If the FTTI is 10s of mS, a GPIO and dedicated timer can easily control the external mux to check the required channels. Depending upon the FTTI, the coverage can range from 12% to 99%.
u_vd_core_logi c.ATB_ASYNC _	NA				
u_vd_core_logi c.CAMERARX _inst	CSIPH Y1	Voltage required for the correct I/O pad operation is not available. The I/O pad logical configuration is corrupted.	Ensure that the I/O pads are operating correctly.	<ol style="list-style-type: none"> <li>1. Implement voltage monitors on 1.8 V, 1.1 V and/or 3.3 V as required.</li> <li>2. Ensure that the configuration of the I/O pad mux modes is correct by reading and comparing the programmed register values to a known good signature within FTTI.</li> <li>3. Ensure that the CRC signatures that are part of the protocol are used for data extraction.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level. It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on an I/O pad.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logic.CM_CORE_AON_inst	PRCM1	PRCM clocks and power management is not configured correctly.	Ensure that all modules are getting the correct clock, power, and rest controls	Ensure that the configuration of the PRCM is regularly checked against a golden signature.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%. It is difficult to achieve higher level coverage with this particular diagnostic because it does not necessarily cover a stuck-at fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as clock monitoring, watchdog timer-based monitoring, and black channel communication to achieve higher coverage for such scenarios.
u_vd_core_logic.CM_CORE_inst	PRCM1	PRCM clocks and power management is not configured correctly.	Ensure that all modules are getting the correct clock, power, and rest controls.	Ensure that the configuration of the PRCM is regularly checked against a golden signature.	Depending upon the identified safety goals, coverage of greater than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%. It is difficult to achieve higher level coverage with this particular diagnostic because it does not necessarily cover a stuck-at fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as clock monitoring, watchdog timer-based monitoring, and black channel communication to achieve higher coverage for such scenarios.
u_vd_core_logic.COUNTER32K_inst	COUNTER32K1	Operation fails due to a transient or permanent fault. COUNTER32K1 timer information is wrong.	Ensure that the 32K counter is operating correctly.	32K counter keeps counting up after a device power on for about 37 days before it loops back from 0xFFFF FFFF to 0x0. Read the CR register, once within every FTTI and ensure that it is showing expected value.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.CTRL_MODULE1	CTRL MODULE1	Control module config gets corrupted due to a transient or permanent fault. Read the control module registers regularly and compare against golden signature.	Ensure that the chip controls are operating correctly.	Read the control module registers regularly and compare against golden signature. Follow guidelines of section 14.5 in the TRM for checking values of all the safety-critical registers.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.  It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower-level fault on an I/O pad <OR> any other component controlled by this module configuration.
u_vd_core_logi c.CTRL_MODULE1	CTRL MODULE1	Control module config gets corrupted due to a transient or permanent fault. Read the control module registers regularly and compare against golden signature.	Ensure that the chip controls are operating correctly.	Read the control module registers regularly and compare against golden signature. Follow guidelines of section 14.5 in the TRM for checking values of all the safety-critical registers	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.  It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower-level fault on an I/O pad <OR> any other component controlled by this module configuration.
u_vd_core_logi c.CUSTOM_EFUSE_inst	NA				
u_vd_core_logi c.DAC_inst	NA				
u_vd_core_logi c.DCAN1_inst	DCAN1 , ECC	DCAN data is corrupted due to a transient or permanent error. Check DCAN message signature in the driver.	Ensure that the DCAN message buffer has the correct message.	Ensure that the DCAN driver has the protocol in place and checks enabled for message signatures per individual message. Coverage can be improved to 100% by embedding additional CAN messages that contain the signatures for the multiple message data being sent.	Every message can be checked for correctness, hence coverage can be greater than 99%.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.DCC1_inst	DCC1	DCC clock comparison operation is incorrect due to a permanent or transient error. Create a test config to confirm correct operation of the DCC module. DCC module does network correctly because of a transient or permanent fault.	Ensure that DCC module can monitor the clock quality correctly.	Choose SYS_CLK1 as the source for both counter 0 and 1 of the DCC module. Configure the seed counters to a small test value. Confirm that counting down for both counters is happening as expected and cross-reference against a timer value.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logi c.DCC2_inst	DCC1	DCC clock comparison operation is incorrect due to a permanent or transient error. Create a test config to confirm correct operation of the DCC module. DCC module does network correctly because of a transient or permanent fault.	Ensure that DCC module can monitor the clock quality correctly.	Choose SYS_CLK1 as the source for both counter 0 and 1 of the DCC module. Configure the seed counters to a small test value. Confirm that counting down for both counters is happening as expected and cross-reference against a timer value.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logi c.DCC3_inst	DCC1	DCC clock comparison operation is incorrect due to a permanent or transient error. Create a test config to confirm correct operation of the DCC module. DCC module does network correctly because of a transient or permanent fault.	Ensure that DCC module can monitor the clock quality correctly.	Choose SYS_CLK1 as the source for both counter 0 and 1 of the DCC module. Configure the seed counters to a small test value. Confirm that counting down for both counters is happening as expected and cross-reference against a timer value.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.



**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.DCC4_inst	DCC1	DCC clock comparison operation is incorrect due to a permanent or transient error. Create a test config to confirm correct operation of the DCC module. DCC module does network correctly because of a transient or permanent fault.	Ensure that DCC module can monitor the clock quality correctly.	Choose SYS_CLK1 as the source for both counter 0 and 1 of the DCC module. Configure the seed counters to a small test value. Confirm that counting down for both counters is happening as expected and cross-reference against a timer value.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logi c.DCC5_inst	DCC1	DCC clock comparison operation is incorrect due to a permanent or transient error. Create a test config to confirm correct operation of the DCC module. DCC module does network correctly because of a transient or permanent fault.	Ensure that DCC module can monitor the clock quality correctly.	Choose SYS_CLK1 as the source for both counter 0 and 1 of the DCC module. Configure the seed counters to a small test value. Confirm that counting down for both counters is happening as expected and cross-reference against a timer value.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logi c.DCC6_inst	DCC1	DCC clock comparison operation is incorrect due to a permanent or transient error. Create a test config to confirm correct operation of the DCC module. DCC module does network correctly because of a transient or permanent fault.	Ensure that DCC module can monitor the clock quality correctly.	Choose SYS_CLK1 as the source for both counter 0 and 1 of the DCC module. Configure the seed counters to a small test value. Confirm that counting down for both counters is happening as expected and cross-reference against a timer value.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logic.DCC7_inst	DCC1	DCC clock comparison operation is incorrect due to a permanent or transient error. Create a test config to confirm correct operation of the DCC module. DCC module does not work correctly because of a transient or permanent fault.	Ensure that DCC module can monitor the clock quality correctly.	Choose SYS_CLK1 as the source for both counter 0 and 1 of the DCC module. Configure the seed counters to a small test value. Confirm that counting down for both counters is happening as expected and cross-reference against a timer value.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logic.DEBUGSS_A_TCLK_mdp_inst	NA				
u_vd_core_logic.DEBUGSS_I_CLK_mdp_inst	NA				
u_vd_core_logic.DEBUGSS_inst	NA				
u_vd_core_logic.DIV_32_inst	NA				
u_vd_core_logic.DPLL_CORE_inst	DPLL1	Voltage required for the correct DPLL operation is not available. <ul style="list-style-type: none"> <li>The DPLL clock quality does not meet DM requirements because of a permanent or transient fault.</li> <li>DPLL loses lock because of a fault or because a temperature/input clock does not meet the required specification.</li> </ul>	Monitor the DPLL operation to confirm correctness of clocks and its quality.	1. Use the DCC modules to continuously measure the clock quality in real time. Also monitor the DPLL-generated interrupts in case of a loss of lock. 2. Monitor the DPLL 1.8-V and 1.2-V voltages.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logic.DPLL_DDR_inst	DPLL1	Voltage required for the correct DPLL operation is not available. <ul style="list-style-type: none"> <li>The DPLL clock quality does not meet DM requirements because of a permanent or transient fault.</li> <li>DPLL loses lock because of a fault or because a temperature/input clock does not meet the required specification.</li> </ul>	Monitor the DPLL operation to confirm correctness of clocks and its quality.	1. Use the DCC modules to continuously measure the clock quality in real time. Also monitor the DPLL-generated interrupts in case of a loss of lock. 2. Monitor the DPLL 1.8-V and 1.2-V voltages.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logic.DPLL_DSP_inst	DPLL1	<p>Voltage required for the correct DPLL operation is not available</p> <ul style="list-style-type: none"> <li>The DPLL clock quality does not meet DM requirements because of a permanent or transient fault.</li> <li>DPLL loses lock because of a fault or because a temperature/input clock does not meet the required specification.</li> </ul>	Monitor the DPLL operation to confirm correctness of clocks and its quality.	<ol style="list-style-type: none"> <li>Use the DCC modules to continuously measure the clock quality in real time. Also monitor the DPLL-generated interrupts in case of a loss of lock.</li> <li>Monitor the DPLL 1.8-V and 1.2-V voltages.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logic.DPLL_GMAC_inst	DPLL1	<p>Voltage required for the correct DPLL operation is not available</p> <ul style="list-style-type: none"> <li>The DPLL clock quality does not meet DM requirements because of a permanent or transient fault.</li> <li>DPLL loses lock because of a fault or because a temperature/input clock does not meet the required specification.</li> </ul>	Monitor the DPLL operation to confirm correctness of clocks and its quality.	<ol style="list-style-type: none"> <li>Use the DCC modules to continuously measure the clock quality in real time. Also monitor the DPLL-generated interrupts in case of a loss of lock.</li> <li>Monitor the DPLL 1.8-V and 1.2-V voltages.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logic.DPLL_PER_inst	DPLL1	<p>Voltage required for the correct DPLL operation is not available</p> <ul style="list-style-type: none"> <li>The DPLL clock quality does not meet DM requirements because of a permanent or transient fault.</li> <li>DPLL loses lock because of a fault or because a temperature/input clock does not meet the required specification.</li> </ul>	Monitor the DPLL operation to confirm correctness of clocks and its quality.	<ol style="list-style-type: none"> <li>Use the DCC modules to continuously measure the clock quality in real time. Also monitor the DPLL-generated interrupts in case of a loss of lock.</li> <li>Monitor the DPLL 1.8-V and 1.2-V voltages.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logic.EMIF1_inst	EMIF1	EMIF does not work correctly for transfer to/from DDR.	Ensure that the EMIF interface is working correctly.	<p>Create a test config for transferring different types of data buffers across variety of source and destination configuration that captures the intent of the application.</p> <p>Confirm by writing the data to DDR memory and read it back to check for correctness.</p> <p>Ensure that the test config includes coverage for the memory protection constraints as needed.</p>	<p>Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.</p> <p>It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on an I/O pad &lt;OR&gt; any other component controlled by this module configuration.</p>
u_vd_core_logic.ESM_inst	NA				
u_vd_core_logic.GPIO1_inst	GPIO1	GPIO does not work correctly for system control due to a permanent or transient fault.	Ensure that the GPIO works as intended.	<p>Recommendation is to use more than one GPIO for critical system monitoring and control. A loopback from GPO to GPI or vice versa can be implemented for a highly critical system control to monitor the system.</p>	<p>Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.</p>
u_vd_core_logic.GPIO2_inst	GPIO1	GPIO does not work correctly for system control due to a permanent or transient fault.	Ensure that the GPIO works as intended.	<p>Recommendation is to use more than one GPIO for critical system monitoring and control. A loopback from GPO to GPI or vice versa can be implemented for a highly critical system control to monitor the system.</p>	<p>Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.</p>

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.GPIO3_inst	GPIO1	GPIO does not work correctly for system control due to a permanent or transient fault.	Ensure that the GPIO works as intended.	Recommendation is to use more than one GPIO for critical system monitoring and control. A loopback from GPO to GPI or vice versa can be implemented for a highly critical system control to monitor the system.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logi c.GPIO4_inst	GPIO1	GPIO does not work correctly for system control due to a permanent or transient fault.	Ensure that the GPIO works as intended.	Recommendation is to use more than one GPIO for critical system monitoring and control. A loopback from GPO to GPI or vice versa can be implemented for a highly critical system control to monitor the system.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logi c.HSDIVIDER_1	HSDIV 1	Voltage required for the correct HSDIV operation is not available <ul style="list-style-type: none"> <li>The DPLL clock quality does not meet DM requirements because of a permanent or transient fault.</li> <li>DPLL loses lock because of a fault or because a temperature/input clock does not meet the required spec</li> </ul>	Monitor the DPLL operation to confirm correctness of clocks and its quality.	<ol style="list-style-type: none"> <li>Use the DCC modules to continuously measure the clock quality in real time. Also monitor the DPLL-generated interrupts in case of a loss of lock.</li> <li>Monitor the DPLL 1.8-V and 1.2-V voltages.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logi c.I2ASYNC_	NA				
u_vd_core_logi c.I2C1_inst	I2C1	I2C does not operate correctly due to a transient or permanent fault.	Monitor the I2C transferred data for correctness and latency.	<ol style="list-style-type: none"> <li>Ensure that I2C message is enwrapped with high level software layers for signature creation and management.</li> <li>Monitor I2C operation is monitored through a watchdog timer.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.I2C2_inst	I2C1	I2C does not operate correctly due to a transient or permanent fault.	Monitor the I2C transferred data for correctness and latency.	<ol style="list-style-type: none"> <li>1. Ensure that I2C message is enwrapped with high level software layers for signature creation and management.</li> <li>2. Monitor I2C operation is monitored through a watchdog timer.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logi c.ICEMELTER_inst	NA				
u_vd_core_logi c.IEEE1500_2_OCP_inst	NA				
u_vd_core_logi c.INTELLIPHY_EMIF1_inst	EMIF1	EMIF does not work correctly for transfer to/from DDR.	Ensure that the EMIF interface is working correctly.	<p>Create a test config for transferring different types of data buffers across variety of source and destination configuration that captures the intent of the application. Confirm by writing the data to DDR memory and read it back to check for correctness. Ensure that the test config includes coverage for the memory protection constraints as needed.</p>	<p>Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.</p> <p>It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on an I/O pad &lt;OR&gt; any other component controlled by this module configuration.</p>
u_vd_core_logi c.ISS_ATB_async_bridge_m_inst	NA				

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logic.L4_WKUP_inst	L41	Configuration read and written over L4 is wrong due to a permanent or transient fault.	Ensure correct module configuration for all L4 targets.	Read back the configuration registers for an IP to ensure correctness of configuration data being transferred. Design software that responds to the L3 interconnect errors and time-outs. Use firewall mechanisms to block unintended accesses. Test the firewalls by trying to make an unallowed data transfer or access and ensure that the error mechanisms are operating correctly.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.  It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on the interconnect <OR> any other component controlled by this module configuration.
u_vd_core_logic.MCASP1_inst					
u_vd_core_logic.MCSP1_inst	SPI1	SPI does not operate correctly due to a transient or permanent fault.	Monitor the SPI transferred data for correctness and latency.	<ol style="list-style-type: none"> <li>1. Ensure that SPI message is wrapped with high-level software layers for signature creation and management.</li> <li>2. Monitor SPI operation through a watchdog timer.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logic.MCSP2_inst	SPI1	SPI does not operate correctly due to a transient or permanent fault.	Monitor the SPI transferred data for correctness and latency.	<ol style="list-style-type: none"> <li>1. Ensure that SPI message is wrapped with high-level software layers for signature creation and management.</li> <li>2. Monitor the SPI operation through a watchdog timer</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logic.MCSPI3_inst	SPI1	SPI does not operate correctly due to a transient or permanent fault.	Monitor the SPI transferred data for correctness and latency.	<ol style="list-style-type: none"> <li>1. Ensure that SPI message is enwrapped with high level software layers for signature creation and management.</li> <li>2. Monitor the SPI operation through a watchdog timer.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logic.MCSPI4_inst	SPI1	SPI does not operate correctly due to a transient or permanent fault.	Monitor the SPI transferred data for correctness and latency.	<ol style="list-style-type: none"> <li>1. Ensure that SPI message is enwrapped with high level software layers for signature creation and management.</li> <li>2. Monitor the SPI operation through a watchdog timer.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
u_vd_core_logic.MMC_inst	MMC1	End-to-end protection on the data transferred through the MMC interface.	Ensure that the card data transferred on the MMC interface is correct.	Ensure that the MMC driver checks for the data signature check interrupt which is part of the protocol and IP. Follow TRM section 20.4.9 for the sequence.	Every data stream received can be checked for correctness, hence coverage can be greater than 99%.
u_vd_core_logic.MMC_mdp_inst	NA				
u_vd_core_logic.PRCM_inst	PRCM1	PRCM clocks and power management is configured correctly.	Ensure that all modules are getting the correct clock, power, and rest controls.	Ensure that the configuration of the PRCM is regularly checked against a golden signature.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%. It is difficult to achieve higher level coverage with this particular diagnostic because it does not necessarily cover a stuck-at fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as clock monitoring, watchdog timer-based monitoring, and black channel communication to achieve higher coverage for such scenarios.



**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.RTI1_inst	RTI1	Watchdog timers do not operate correctly due to a transient or permanent fault.	Ensure that the watchdog timers are operating correctly to monitor the latency of critical system tasks.	Create a test config on RTI by configuring RTIWWDRXNCT RL to generate an interrupt to the CPU, configure RTIWWDSIZECT RL to the window size needed for the application, preload the DWD sequence and go through the process where the key is not written in to generate a DWD NMI. See Section 24.6.2 for design of the sequence.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level and executed once every FTTI. A small preload value and window can be designed to complete the test quickly. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.
u_vd_core_logi c.RTI2_inst	RTI1	Watchdog timers do not operate correctly due to a transient or permanent fault.	Ensure that the watchdog timers are operating correctly to monitor the latency of critical system tasks.	Create a test config on RTI by configuring RTIWWDRXNCT RL to generate an interrupt to the CPU, configure RTIWWDSIZECT RL to the window size needed for the application, preload the DWD sequence and go through the process where the key is not written in to generate a DWD NMI. See Section 24.6.2 for design of the sequence.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level and executed once every FTTI. A small preload value and window can be designed to complete the test quickly. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.RTI3_inst	RTI1	Watchdog timers do not operate correctly due to a transient or permanent fault.	Ensure that the watchdog timers are operating correctly to monitor the latency of critical system tasks.	Create a test config on RTI by configuring RTIWWDRXNCT RL to generate an interrupt to the CPU, configure RTIWWDSIZECT RL to the window size needed for the application, preload the DWD sequence and go through the process where the key is not written in to generate a DWD NMI. See Section 24.6.2 for design of the sequence.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level and executed once every FTTI. A small preload value and window can be designed to complete the test quickly. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.
u_vd_core_logi c.RTI4_inst	RTI1	Watchdog timers do not operate correctly due to a transient or permanent fault.	Ensure that the watchdog timers are operating correctly to monitor the latency of critical system tasks.	Create a test config on RTI by configuring RTIWWDRXNCT RL to generate an interrupt to the CPU, configure RTIWWDSIZECT RL to the window size needed for the application, preload the DWD sequence and go through the process where the key is not written in to generate a DWD NMI. See Section 24.6.2 for design of the sequence.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level and executed once every FTTI. A small preload value and window can be designed to complete the test quickly. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.RTI5_inst	RTI1	Watchdog timers do not operate correctly due to a transient or permanent fault.	Ensure that the watchdog timers are operating correctly to monitor the latency of critical system tasks.	Create a test config on RTI by configuring RTIWWDRXNCT RL to generate an interrupt to the CPU, configure RTIWWDSIZECT RL to the window size needed for the application, preload the DWD sequence and go through the process where the key is not written in to generate a DWD NMI. See Section 24.6.2 for design of the sequence.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level and executed once every FTTI. A small preload value and window can be designed to complete the test quickly. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.
u_vd_core_logi c.SC_PWR_CL K_DIV_IO_SC _pd_	NA				
u_vd_core_logi c.SMARTREFL EX_CORE_inst	SMARTREFL EX1	AVS class0 voltage is outside the accepted DM range because of a transient or permanent fault.	Ensure that the voltage is within the accepted range.	Deploy a voltage monitor mechanism by using one channel of the on-chip ADC or an external PMIC diagnostic or any other method. Ensure that the system software gracefully handles the unacceptable voltage range events.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level, if the safety mechanism is polling the voltage monitoring instrumentation every FTTI <OR> the instrumentation generates a high priority interrupt immediately upon the error.
u_vd_core_logi c.SMARTREFL EX_DSPEVE_inst	SMARTREFL EX1	AVS class0 voltage is outside the accepted DM range because of a transient or permanent fault.	Ensure that the voltage is within the accepted range.	Deploy a voltage monitor mechanism by using one channel of the on-chip ADC or an external PMIC diagnostic or any other method. Ensure that the system software gracefully handles the unacceptable voltage range events.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level, if the safety mechanism is polling the voltage monitoring instrumentation every FTTI <OR> the instrumentation generates a high priority interrupt immediately upon the error.
u_vd_core_logi c.T2ASYNC_C M_CORE_	NA				
u_vd_core_logi c.T2ASYNC_L 3_	NA				

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.T2ASYNC_L4_	NA				
u_vd_core_logi c.T2ASYNC_OCP_L4_	NA				
u_vd_core_logi c.T2ASYNC_P RM_TO_L3_	NA				
u_vd_core_logi c.TESOC_	TESOC C1	Ensure correctness of TESOC based tests in ROM.	Ensure that the TESOC ROM has the correct data. Also ensure that the TESOC configuration is correctly done.	Software should schedule the TESOC ROM and ensure that the signature matches the expected values. This could be done after the boot and TESOC tests are completed to expedite boot time. Ensure that the TESOC configuration is correct.	Signature analysis coupled with configuration checks ensures more than coverage of more than 50% by detecting an anomaly immediately. The FTTI for this test will be dependent on how often customers run TESOC. If it is run just once at boot time, then running the diagnostic once is a reasonable customer decision.
u_vd_core_logi c.TIMER1_inst	TIMER 1	Timer does not work correctly due to a transient or permanent fault.	Ensure that timer operates correctly.	Create a test config on GPTimer by configuring the IP as defined in 18.2.5.2 of the TRM. Read the counter registers or profile one timer against another.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic because it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.
u_vd_core_logi c.TOP_SSYNC_P BIST_	NA				
u_vd_core_logi c.TOP_SYNC_P BIST_	NA				
u_vd_core_logi c.adas_ddr_bs can_inst	NA				
u_vd_core_logi c.adas_efuse_ autoload_ handler_inst	NA				
u_vd_core_logi c.dftss_adas_lo w_inst	NA				

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logic.dma_xbar_inst	NA				
u_vd_core_logic.i_adas_io_block	NA				
u_vd_core_logic.intelliphy_	NA				
u_vd_core_logic.ipu_core_reset_	NA				
u_vd_core_logic.tesoc_inst	TESOC1	Ensure correctness of TESOC-based tests in ROM.	Ensure that the TESOC ROM has the correct data. Also ensure that the TESOC configuration is correctly done.	Software should schedule the TESOC ROM and ensure that the signature matches the expected values. This could be done after the boot and TESOC tests are completed to expedite boot time. Ensure that the TESOC configuration is correct.	Signature analysis coupled with configuration checks ensures coverage of more than 50% by detecting an anomaly immediately. The FTTI for this test will be dependent on how often customers run TESOC. If it is run just once at boot time, then running the diagnostic once is a reasonable customer decision.
u_vd_core_logic.test_glue_adas_inst	NA				
u_vd_core_logic.testcm_adas_wrap_inst	NA				
u_vd_core_logic.top_cdr_dcdr_inst	NA				

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.u_sc_L3	L31	Data does not reach the correct end point and/or gets corrupted during transfer due to a permanent or transient fault.	Ensure that data does reach the correct end point and data corruption, if any, is diagnosed during transfer due to a permanent or transient fault.	Implement end-to-end data signatures to ensure data is transferred correctly through the bus. Read back the configuration registers for an IP to ensure correctness of configuration data being transferred. Design software that responds to the L3 interconnect errors and time-outs. Use firewall mechanisms to block unintended accesses. Test the firewalls by trying to make an unallowed data transfer or access and ensure that the error mechanisms are operating correctly.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.  It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on the interconnect <OR> any other component controlled by this module configuration.
u_vd_core_logi c.u_top_et_	NA				
u_sc_common. DCAN2_	DCAN1 , ECC	Check DCAN message signature in the driver.	Ensure that the DCAN message buffer has the correct message.	Ensure that the DCAN driver has the protocol in place and checks enabled for message signatures per individual message. Coverage can be improved to 100% by embedding additional CAN messages that contain the signatures for the multiple message data being sent.	Every message can be checked for correctness, hence coverage can be greater than 99%.
u_sc_common. DFT_	NA				
u_sc_common. ELM_inst	NA				

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_sc_common. GPMC_inst	GPMC 1	Data corruption on GPMC interface due to a permanent or transient fault	Ensure that the card data transferred on GPMC interface is correct.	Ensure that the GPMC driver checks for the data signature check interrupt which is part of the protocol and IP. Follow TRM Section 11.3.5 for the sequence and reference to use ELM for connecting to memories that do not have error correction capabilities.	Every data stream received can be checked for correctness, hence coverage can be greater than 99%.
u_sc_common. HEAD_POS_F OR_ASYNC_B RIDGE_	NA				
u_sc_common. I2ASYNC_	NA				
u_sc_common. L4_CFG_inst	L41	Configuration read and written over L4 is wrong because of a permanent or transient fault.	Ensure correct module configuration for all L4 targets.	Read back the configuration registers for an IP to ensure correctness of configuration data being transferred. Design software that responds to the L3 interconnect errors and time-outs. Use firewall mechanisms to block unintended accesses. Test the firewalls by trying to make an unallowed data transfer or access and ensure that the error mechanisms are operating correctly.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.  It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on the interconnect <OR> any other component controlled by this module configuration.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_sc_common. L4_PER2_inst	L41	Configuration read and written over L4 is wrong because of a permanent or transient fault.	Ensure correct module configuration for all L4 targets.	Read back the configuration registers for an IP to ensure correctness of configuration data being transferred. Design software that responds to the L3 interconnect errors and time-outs. Use firewall mechanisms to block unintended accesses. Test the firewalls by trying to make an unallowed data transfer or access and ensure that the error mechanisms are operating correctly.	<p>Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.</p> <p>It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on the interconnect &lt;OR&gt; any other component controlled by this module configuration.</p>
u_sc_common. L4_PER3_inst	L41	Configuration read and written over L4 is wrong because of a permanent or transient fault.	Ensure correct module configuration for all L4 targets.	Read back the configuration registers for an IP to ensure correctness of configuration data being transferred. Design software that responds to the L3 interconnect errors and time-outs. Use firewall mechanisms to block unintended accesses. Test the firewalls by trying to make an unallowed data transfer or access and ensure that the error mechanisms are operating correctly.	<p>Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.</p> <p>It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on the interconnect &lt;OR&gt; any other component controlled by this module configuration</p>



**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_sc_common.L4_PER_inst	L41	Configuration read and written over L4 is wrong because of a permanent or transient fault.	Ensure correct module configuration for all L4 targets.	Read back the configuration registers for an IP to ensure correctness of configuration data being transferred. Design software that responds to the L3 interconnect errors and time-outs. Use firewall mechanisms to block unintended accesses. Test the firewalls by trying to make an unallowed data transfer or access and ensure that the error mechanisms are operating correctly.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.  It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on the interconnect <OR> any other component controlled by this module configuration.
u_sc_common.MAILBOX1_inst	NA				
u_sc_common.MAILBOX2_inst	NA				
u_sc_common.OCMC_DMLED_	NA				
u_sc_common.OCMC_RAM1_	OCMC1	OCMC configuration or data transfer is wrong due to a transient or permanent fault.	Ensure correct module configuration for all OCMC registers.	Read back the configuration registers for OCMC to ensure correctness of configuration data being written. Use firewall mechanisms to block unintended accesses. Test the firewalls by trying to make an unallowed data transfer or access and ensure that the error mechanisms are operating correctly.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90%.  It is definitely more robust to design multiple layers of diagnostics. For example, a higher software layer that implements a black channel communication automatically ensures detection of a lower level fault on the interconnect <OR> any other component controlled by this module configuration.
u_sc_common.OCP_WP_NOC_	NA				
u_sc_common.PWMSS1_inst	NA				
u_sc_common.QSPI_inst	NA				
u_sc_common.SC_COMMON_PBIST_	NA				

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_sc_common. SC_PWR_CLK _DIV_SC_CO MMON_pd_	NA				
u_sc_common. SPINLOCK_ins t.SPINLOCK_	NA				
u_sc_common. TIMER2_inst	TIMER 1	Timer does not work correctly due to a transient or permanent fault.	Ensure that timer operates correctly.	Create a test config on GPTimer by configuring the IP as defined in Section 18.2.5.2 of the TRM. Read the counter registers or profile one timer against another.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.
u_sc_common. TIMER3_inst	TIMER 1	Timer does not work correctly due to a transient or permanent fault.	Ensure that timer operates correctly.	Create a test config on GPTimer by configuring the IP as defined in Section 18.2.5.2 of the TRM. Read the counter registers or profile one timer against another.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_sc_common. TIMER4_inst	TIMER 1	Timer does not work correctly due to a transient or permanent fault.	Ensure that timer operates correctly	Create a test config on GPTimer by configuring the IP as defined in Section 18.2.5.2 of the TRM. Read the counter registers or profile one timer against another.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.
u_sc_common. TIMER5_inst	TIMER 1	Timer does not work correctly due to a transient or permanent fault.	Ensure that timer operates correctly	Create a test config on GPTimer by configuring the IP as defined in Section 18.2.5.2 of the TRM. Read the counter registers or profile one timer against another.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.
u_sc_common. TIMER6_inst	TIMER 1	Timer does not work correctly due to a transient or permanent fault.	Ensure that timer operates correctly	Create a test config on GPTimer by configuring the IP as defined in Section 18.2.5.2 of the TRM. Read the counter registers or profile one timer against another.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_sc_common. TIMER7_inst	TIMER 1	Timer does not work correctly due to a transient or permanent fault.	Ensure that timer operates correctly	Create a test config on GPTimer by configuring the IP as defined in Section 18.2.5.2 of the TRM. Read the counter registers or profile one timer against another.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.
u_sc_common. TIMER8_inst	TIMER 1	Timer does not work correctly due to a transient or permanent fault.	Ensure that timer operates correctly	Create a test config on GPTimer by configuring the IP as defined in Section 18.2.5.2 of the TRM. Read the counter registers or profile one timer against another.	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults as well as transient faults with a reasonably high FTTI however, the coverage can be greater than 90% It is difficult to achieve higher level coverage with this particular diagnostic as it does not necessarily cover a stuck-at transient fault outside of a configuration register to affect the correct operation of the device. It is necessary to design higher layers of software checks such as use multiple timers based monitoring to achieve coverage of more than 99%.
u_sc_common. UART1_	NA				
u_sc_common. UART2_	NA				
u_sc_common. UART3_	NA				
u_sc_common. sc_common_c dr_dcdr_inst	NA				

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_sc_common. u_GMAC.GMA C_	GMAC 1	Check Ethernet message signature in the driver. IP accesses only the memory space that is allowed for it to access.	Ensure that the Ethernet message buffer has the correct message.	Ensure that the Ethernet driver has the protocol in place and checks enabled for message signatures per individual message. Coverage can be improved to 100% by embedding additional Ethernet messages for black channel communication that contain the signatures for the multiple message data being sent. Follow TRM Sections 19.7.4, 19.7.5, and 19.7.6 for detailed sequences	Every message can be checked for correctness, hence coverage can be greater than 99%.
u_sc_common. u_sc_common _et_	NA				
u_sc_common. ATL_inst	NA				
u_sc_common. MCAN_	MCAN 1, ECC	Check the DCAN message signature in the driver.	Ensure that the DCAN message buffer has the correct message.	Ensure that the DCAN driver has the protocol in place and checks enabled for message signatures per individual message. Coverage can be improved to 100% by embedding additional CAN messages that contain the signatures for the multiple message data being sent.	Every message can be checked for correctness, hence coverage can be greater than 99%.
u_sc_common. MCASP	NA				
u_vd_core_logi c.u_sc_benelli	M41- SWLS	M4 fails to operate correctly due to a transient or permanent fault.	Ensure that the safety-critical parts of the software are computed robustly on M4.	Implement software lockstep for highest coverage. Also ensure that the memory protection units in Benelli as well as system firewalls are correctly configured to ensure isolation between an ASIL task and a QM task.	Lockstep implementation will achieve coverage of more than 99%.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.u_sc_dss	DSS1	Detect DSS frame freeze. <ul style="list-style-type: none"> <li>• Detect Latency violations beyond acceptable limits.</li> <li>• Ensure that DSS reads the frame from the correct location and has written the correct frame data.</li> <li>• IP only accesses the memory space that is allowed for it to access.</li> </ul>	Ensure that the DSS operation is correct.	<ol style="list-style-type: none"> <li>1. Ensure that with every new DSS frame a unique signature is created and transmitted. This signature should be checked through DSS write-back DMA and a DSP or M4 routine checking for the expected frame values.</li> <li>2. Ensure a timer is part of the latency measurement .</li> <li>3. Confirm the DSS frame on the receive side as well outside the chip and the ECU.</li> </ol>	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults, higher coverage is achievable.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_core_logi c.u_sc_iss	ISS1	RAW2YUV RGB conversion is correctly done. <ul style="list-style-type: none"> <li>Image sensor configuration is correct.</li> <li>IP only accesses the memory space that is allowed for it to access.</li> <li>IP misses any latency budget deadlines.</li> <li>Check to determine if the functionality [WB RSZ WDR]} is configured correctly and working.</li> <li>Check to determine if the LUTs for ISS operations that are stored in the DDR are not corrupted.</li> </ul>	Ensure that ISS operation is correct.	<ol style="list-style-type: none"> <li>Ensure that for a known signature pixel data, ISP is working correctly for all intended operations within FTTI.</li> <li>Ensure that the timer is part of the continuous latency measurement s.</li> <li>Ensure that the sensor configuration is read over I2C frequently (FTTI driven) and compared against the target known good value at that point.</li> <li>Read the configuration of the module and compare against expected values.</li> <li>Create LUT signatures, read the data from RAM at regular intervals, and compare against the signature.</li> </ol>	Depending upon the identified safety goals, coverage of more than 50% can be achieved for diagnostics designed at system level. For permanent faults, higher coverage is achievable.
u_vd_core_logi c.u_sc_vip	NA				
u_vd_dspeve_l ogic.u_sc_eve	EVE1-SWLS	EVE fails to operate correctly due to a transient or permanent fault.	Ensure that the safety-critical parts of the software are computed robustly on EVE.	Implement software lockstep for highest coverage. Also ensure that the memory protection units in C66x as well as system firewalls are correctly configured to ensure isolation between an ASIL task and a QM task.	Lockstep implementation will achieve coverage of more than 99%.

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
u_vd_dspeve_logic.u_sc_turing1	DSP1-SWLS	DSP fails to operate correctly due to a transient or permanent fault.	Ensure that the safety-critical parts of the software are computed robustly on DSP.	Implement software lockstep for highest coverage. Also ensure that the memory management units in C66x as well as system firewalls are correctly configured to ensure isolation between an ASIL task and a QM task.	Lockstep implementation will achieve coverage of more than 99%.
u_vd_dspeve_logic.u_sc_turing2	DSP1-SWLS	DSP fails to operate correctly due to a transient or permanent fault.	Ensure that the safety-critical parts of the software are computed robustly on DSP.	Implement software lockstep for highest coverage. Also ensure that the memory protection units in C66x as well as system firewalls are correctly configured to ensure isolation between an ASIL task and a QM task.	Lockstep implementation will achieve coverage of more than 99%.
misc	NA				
<b>Mission - Memories - ROM</b>					
TESOC	TESOC1	Ensure correctness of TESOC-based tests in ROM.	Ensure that the TESOC ROM has the correct data. Also ensure that the TESOC configuration is correctly done.	Software should schedule the TESOC ROM and ensure that the signature matches the expected values. This could be done after the boot and TESOC tests are completed to expedite boot time.	Signature analysis ensures more than coverage of more than 99% by detecting an anomaly immediately. The FTTI for this test will be dependent on how often customers run TESOC. If it is run just once at boot time, then running the diagnostic once is a reasonable customer decision.
TOP	TOPROM	Ensure correct boot code is in the ROM.	Ensure that the boot ROM has the correct data.	Software should schedule the boot ROM signature computation securely and ensure that the signature matches the expected values. This could be done after the boot and TESOC tests are completed to expedite boot time.	Signature analysis ensures more than coverage of more than 99% by detecting an anomaly immediately. Because this ROM code is only executed once per boot cycle, hence the diagnostic must run only once.
BENELLI	NA				



**Table 3-11. System Diagnostics (continued)**

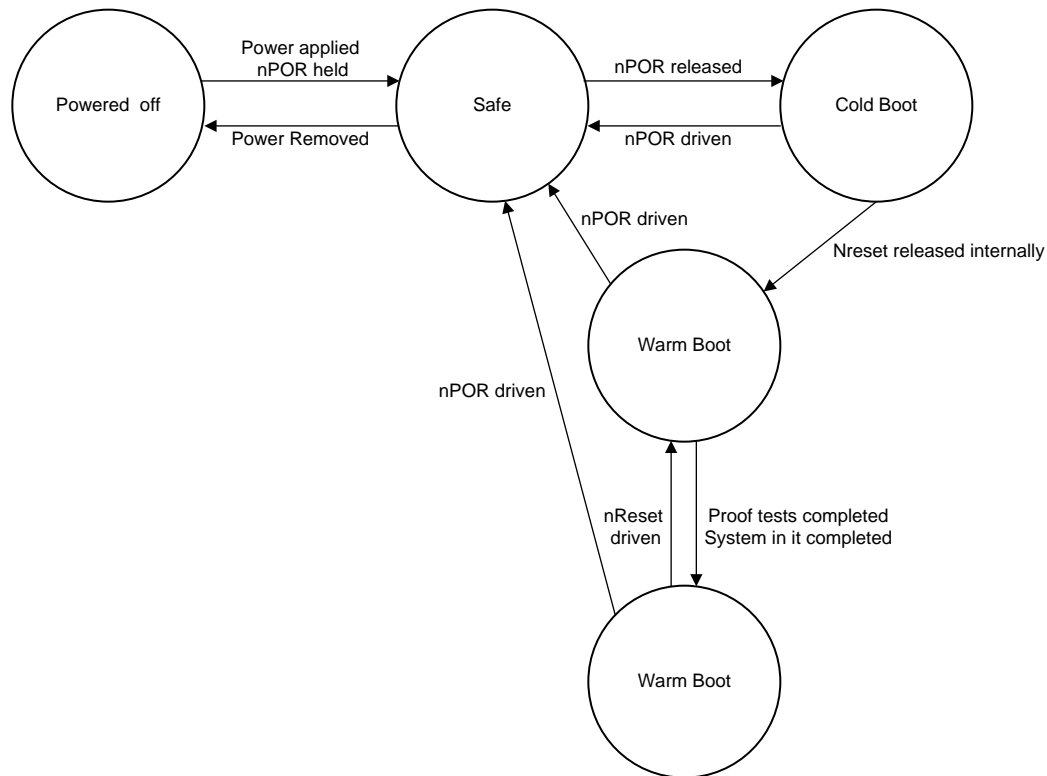
Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
BENELLI (pbist)	NA				
PBIST	NA				
ISS	NA				
EVE	NA				
TURING1	NA				
TURING2	NA				
vslido	VSLDO 1	SRAM voltage is outside the accepted DM range because of a transient or permanent fault.	Ensure that the voltage is within the accepted range.	Deploy a voltage monitor mechanism by using one channel of the on-chip ADC or an external PMIC diagnostic or any other method. Ensure that the system software gracefully handles the unacceptable voltage range events.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level, if the safety mechanism is polling the voltage monitoring instrumentation every FTTI <OR> the instrumentation generates a high priority interrupt immediately upon the error.
vbbldo	VBBLD O1	SRAM voltage is outside the accepted DM range because of a transient or permanent fault.	Ensure that the voltage is within the accepted range.	Deploy a voltage monitor mechanism by using one channel of the on-chip ADC or an external PMIC diagnostic or any other method. Ensure that the system software gracefully handles the unacceptable voltage range events.	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level, if the safety mechanism is polling the voltage monitoring instrumentation every FTTI <OR> the instrumentation generates a high priority interrupt immediately upon the error.
anatop	NA				
afe	NA				

**Table 3-11. System Diagnostics (continued)**

Hierarchical Part Level	Tag	Description of Functional Fail Mode and Diagnostic	Diagnostic Objective	Suggested Sequence	Achievable Coverage (over one FTTI)
VBGAPTSV2	VBGA P1	VBGAP reference to temperature sensor and on-chip LDOs is corrupted because of a transient or a permanent fault.	Ensure that the VBGAP reference is within the accepted range.	<ol style="list-style-type: none"> <li>1. Deploy a voltage monitor mechanism by using one channel of the on-chip ADC or an external PMIC diagnostic or any other method. Ensure that the system software gracefully handles the unacceptable voltage range events.</li> <li>2. For temperature sensing, use another temperature sensor outside the chip and calibrate the readings of that against the on-chip sensor, which references the VBGAP. This mechanism ensures that any anomaly on the VBGAP is identified quickly.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level, if the safety mechanism is polling the voltage monitoring instrumentation every FTTI <OR> the instrumentation generates a high priority interrupt immediately upon the error.
RCOSC32K	NA				
HSDIVIDER	HSDIV 1	Voltage required for the correct HSDIV operation is not available. <ul style="list-style-type: none"> <li>• The DPLL clock quality does not meet DM requirements because of a permanent or transient fault.</li> <li>• DPLL loses lock because of a fault or because a temperature/input clock does not meet the required specifications.</li> </ul>	Monitor the DPLL operation to confirm correctness of clocks and its quality.	<ol style="list-style-type: none"> <li>1. Use the DCC modules to continuously measure the clock quality in real time. Also monitor the DPLL-generated interrupts in case of a loss of lock.</li> <li>2. Monitor the DPLL 1.8-V and 1.2-V voltages.</li> </ol>	Depending upon the identified safety goals, coverage of more than 99% can be achieved for diagnostics designed at system level.
avdac1bgtv1	NA				

### 3.7.7 Operating States

The TDA3x device products have a common architectural definition of operating states. The system developer must observe these operating states in their software and system-level design concepts. [Figure 3-5](#) shows and describes the operating states state machine.



**Figure 3-5. Operating States**

- **Powered Off** – This is the initial operating state of the TDA3x device. No power is applied to the different power supply rails and the device is nonfunctional. This state can only transition to the safe state, and can only be reached from the safe state.
- **Safe** – In this state, the TDA3x device is powered but nonoperational. The nPOR (power-on reset, also known as cold reset) is asserted by the system, but is not released until power supplies have ramped to a stable state. When the product is in the safe state, the CPU and peripherals are nonfunctional. The Terminal Description section in the [TDA3x Data Manual](#) provides details about the pin behavior during reset. Typically an external element may be required to control and hold nPOR, to ensure that TDA3x is in safe state.
- **Cold Boot** – In this state, key analog elements, digital control logic, and debug logic are initialized for future use. The CPU remains powered but nonoperational. When the cold boot process is completed, the nRESET signal is internally released, leading to the warm boot stage. The nRESET signal transition change can be monitored externally on the nRSTOUT output pin.
- **Warm Boot** – This mode resets digital logic and enables the Cortex-M4. The Cortex-M4 begins executing software from the internal boot ROM and software initialization of the device begins. There is no hardware interlock to say that warm boot is completed; this is a software decision. The user must determine the entry and exit of this state.
- **Operational** – During the this mode, the device can support safety-critical functionality. The user must determine the entry and exit of this state.

### 3.7.8 Management of Errors

The error response is an action taken by the TDA3x device or system when an error is indicated in an operational state. Multiple potential error responses are possible for the TDA3x product family. The system integrator must determine which error response should be taken, and ensure that the response is consistent with the system safety concept.

- CPU abort – This response is implemented directly in the CPU, for diagnostics implemented in the CPU. During an abort, the program sequence transfers context to an abort handler and software has an opportunity to manage the fault.
- CPU interrupts – This response can be implemented for diagnostics outside of the CPU. An interrupt allows events external to the CPU to generate a program sequence context transfer to an interrupt handler where software has an opportunity to manage the fault.
- Generation of warm reset – This response lets the device change from operational state to warm boot state. The warm reset could be generated from an external monitor on nRESET or internally by the software global warm reset, emulation warm reset, or watchdog. Re-entry to the warm reset state allows possibility for software recovery when recovery in the operational state was not possible.
- Generation of cold reset – This response lets the device change from cold boot, warm boot, or operational states to safe state. From this state, it is possible to reenter cold boot to attempt recovery when recovery through warm boot is not possible. It is also possible to move to the powered-down state, if desired, to implement a system-level safe state. This response can be generated by the software global cold reset or nPOR pin, but is primarily driven by monitors external to the TDA3x device.

## 3.8 Discussion About Conceptual Safety Mechanisms and Assumptions of Use

You, as the system and equipment manufacturer or designer, must ensure that your systems (and any TI hardware or software components incorporated in your systems) meet all applicable safety, regulatory, and system-level performance requirements. All application and safety-related information in this document (including application descriptions, suggested safety measures, suggested TI products, and other materials) is provided for reference only. You understand and agree that your use of TI components in safety-critical applications is entirely at your risk, and that you (as a buyer) agree to defend, indemnify, and hold TI harmless from any and all damages, claims, suits, or expense resulting from such use.

In this section, the safety mechanisms for each major functional block of the TDA3x device architecture are summarized and general assumptions of use are provided. Use this information to determine the strategy for using safety mechanisms. The details of each safety mechanism are in the device-specific TRM for the TDA3x device used.

TI classifies technical recommendations for the use of safety mechanisms in this section into several categories. The TI recommendations should not be considered infallible. There are many diverse ways to implement safe systems, and alternate safety mechanisms provide support to achieve desired safety metrics. The categories of recommendation are:

- Mandatory – This notation indicates a safety mechanism that is always operable during normal functional operation and cannot be disabled by user action.
- Highly Recommended – This notation indicates a safety mechanism that TI believes provides a high value of diagnostics that are difficult to implement by other means. The user retains the choice whether or not to use the safety mechanism in their design, because user action is either needed to enable the safety mechanism or user action can disable the safety mechanism.
- Recommended – This notation indicates a safety mechanism that TI believes provides a valuable diagnostic that can also be implemented by other means. The user retains the choice whether or not to use the safety mechanism in their design, because user action is either needed to enable the safety mechanism or user action can disable the safety mechanism.
- Optional – This notation indicates a safety mechanism that TI believes provides a lower value diagnostic that can also be implemented by other means. The user retains the choice whether or not to use the safety mechanism in their design, because user action is either needed to enable the safety mechanism or user action can disable the safety mechanism.

Depending on the safety standard and end equipment targeted, it may be necessary to manage not only single-point faults, but also latent faults. According to ISO 26262:2011, the latent faults to consider are when it may cause a fault in the safety mechanism's ability to detect violation of a safety goal. Latent fault testing does not need to occur during the fault-tolerant time interval, but can be performed at boot time, at shut down, or periodically as determined by the system developer. Many of the safety mechanisms described in this section can be used as primary diagnostics, diagnostics for latent fault, or both. When considering system design for management of latent faults, take care to include failure of CPU and memories in consideration for any primary diagnostic that is executed using software, by using the TESOC LBIST and MBIST functionality provided in TDA3x. Customers may also decide to design their own functional self tests of CPU and memories that reflect the system software accurately.

### 3.8.1 Power Supply

The TDA3x device family products require an external device to supply the necessary voltages and currents for proper operation. Separate voltage rails are available for core logic, I/O, PLL, and other functions.

#### 3.8.1.1 External Voltage Supervisor

TI recommends using an external voltage supervisor to monitor all voltage rails. The voltage supervisor must be configured with overvoltage and undervoltage thresholds matching the voltage ranges supported by the target device (as noted in the device-specific data sheet). Error response, diagnostic testability, and any necessary software requirements are defined by the external voltage supervisor selected by the system integrator.

#### 3.8.1.2 Power Sequencing

For the TDA3x device to function properly, TI recommends implementing the power sequencing requirements, as outlined in the *Power Sequence* section of the [TDA3x Data Manual](#).

#### 3.8.1.3 Notes

- TDA3x management of voltage supervision at the system level can be simplified by using a TI system basis chip, developed for use with the family.
- Devices can be implemented with multiple power rails that are intended to be ganged together on the system PCB. For proper operation of power diagnostics, TI recommends implementing one voltage supervisor per ganged rail.
- Common mode failure analysis of the external voltage supervisor may be useful to determine dependencies in the voltage generation and supervision circuitry.

### 3.8.2 Power Management

The power, reset, and clock management (PRCM) module controls switchable power domains. Dependent on the family variant used, one or more power domains can be implemented. Power domains can be permanently configured at manufacturing time by TI or they can be programmed by the user. To determine the power domains supported on your device, see the device-specific data sheet. For programming information, see the device-specific technical reference manual.

### 3.8.3 Software Diagnostic for Configuration Checks

This test is relevant for all safety-critical configuration register writes done by any layer of software. The configuration register may belong to any functional, control, or peripheral interface module in the SOC. TI recommends running this test for all safety-critical operations.

### 3.8.3.1 Software Read Back of Written Configuration for Clock Control

To ensure proper configuration of memory-mapped configuration registers, TI recommends that software implement a test to confirm proper operation of all register writes. To support this software test, configure the memory space as a strongly ordered, non-bufferable memory region using the memory management units (MMU). This ensures that the register write has completed before the read back is initiated. TI recommends performing the readback test immediately after configuration being written.

### 3.8.3.2 Periodic Read Back of Configuration Registers

Periodic read back of configuration registers can provide a diagnostic for inadvertent writes or disturb of these registers. Error response, diagnostic testability, and any necessary software requirements are defined by the software implemented by the system integrator. TI recommends using read back of configuration registers mechanism.

## 3.8.4 Clocks

The TDA3x device family products are primarily synchronous logic devices and as such require clock signals for proper operation. The clock management logic includes clock sources, clock generation logic (including clock multiplication by PLLs), clock dividers, and clock distribution logic. The registers that are used to program the clock management logic are in the PRCM. The customer software or hardware ensures monitoring of the relevant temperature, lock, clock quality, and any other signal relevant to ensure the correct clock (as required in the data sheet) through the mechanisms listed in the TRM. The relevant sections in the TRM that will help design the safety concept for clocks are: Section 3.6, subsystem integration chapters on all modules, and Section 24.4. In addition, safety engineers can use the “clock tree tool” to understand the clock routing and connections of the device. To access the tool, contact your TI support channel.

### 3.8.4.1 Monitoring External Clock Outputs

The TDA3x device family products provide the capability to export certain internal clocking signals for external monitoring. This feature can be configured by programming registers in the PRCM module. To determine the number of external clock outputs implemented and the register mapping of internal clocks that can be exported, see the device-specific data sheet. Export of internal clocks on the dedicated outputs is not enabled by default and must be enabled through software. It is possible to disable and configure this diagnostic through software. Use of the CLKOUT feature for external monitoring of internal clocks is optional.

### 3.8.4.2 Internal Window Watchdog

The TDA3x device family products support the use of internal window watchdog timers. For details of programming the internal watchdog, see the device-specific TRM (Section 24.6, 24.6.2.2, and 24.6.2.3). The watchdog is a digital window dual-threshold watchdog. The user programs a window size and other relevant configuration to the watchdog. The user program must provide a predetermined “pet” response to the watchdog within the allowed time window. Violation of this rule triggers an error response. The watchdog can issue either an internal (warm) system reset or a CPU interrupt when a failure is detected. When the watchdog generates a reset, the RSTOUTn pin is driven low. The watchdog is not enabled after reset. Once enabled by the software, the watchdog can be disabled or enabled by writing the proper disable or enable sequence. Using the watchdog functionality is optional but recommended.

### 3.8.4.3 External Watchdog

When using an external watchdog, there is a possibility to reduce common-mode failure with the TDA3x clocking system, because the watchdog can use clock, reset, and power that are separate from the system being monitored. Error response, diagnostic testability, and any necessary software requirements are defined by the external watchdog selected by the system integrator. The TDA3x platform highly recommends the use of an external watchdog in addition to the internally-provided watchdog.

### 3.8.4.4 Notes

- There are many possible implementations of watchdogs for use in providing clock and CPU

diagnostics. In general, TI recommends using an external watchdog in addition to an internal watchdog for reasons of reduced common mode failure. TI also recommends using a program sequence, windowed, or question-and-answer watchdog as opposed to a single threshold watchdog due to the additional failure modes that can be detected by a more advanced watchdog.

- Driving a high-frequency clock output on the CLKOUT pins may have EMI implications.

### 3.8.5 Reset

The TDA3x device family products require an external reset at cold and power-on (porz) to place all asynchronous and synchronous logic into a known state. The power-on reset generates an internal warm reset signal to reset the majority of digital logic as part of the boot process. Optionally, the resetn pin can be driven to generate an internal warm reset. The rstoutn signal is provided at device level as an I/O pin. The rstoutn pin asserts low in response to any global reset condition on the device. For more information on the reset functionality, see the device-specific data sheet and the device-specific technical reference manual.

#### 3.8.5.1 Software Cold and Warm Reset Generation

The system module provides the software the ability to generate an internal cold or warm reset. This is accomplished by writing appropriate control bits in the PRCM Reset Control (PRM\_RSTCTRL) register. Software can use this feature to attempt failure recovery. Use of the software cold or warm reset is optional.

#### 3.8.5.2 External and Internal Watchdog

An external watchdog can provide secondary diagnostic, if used together with internal watchdog timers. For more information on this diagnostics, see [Section 3.8.4.2](#) or [Section 3.8.4.3](#). It is also a highly effective diagnostic against the device being stuck in reset or a functional loop that it cannot break out of.

#### 3.8.5.3 External Monitoring of Warm Reset (RSTOUTn)

The RSTOUTn warm reset signal is implemented as an output. An external monitor can be used to detect expected or unexpected changes to the state of the internal warm reset control signal. Error response, diagnostic testability, and any necessary software requirements are defined by the external monitor selected by the system integrator. This feature is considered optional.

#### 3.8.5.4 Software Check of Cause of Last Reset

The PRCM provides a status register (PRM\_RSTST) that latches the cause of the most recent reset event. Boot software that checks the status of this register to determine the last reset event is typically implemented by software developers. This information can be used by the software to manage failure recovery. Software use of the PRM\_RSTST to check last reset cause of highly recommended to uncover a latent or permanent fault.

#### 3.8.5.5 Notes

Internal watchdogs are not a viable option for reset diagnostics because the monitored reset signals interact with the internal watchdogs.

### 3.8.6 PRCM and Control Module

The PRCM and control modules provide general system configuration control. To provide protection for unintended change of control module registers, certain address regions in the control module register address space can be locked. For more information, see the control module description in the device-specific TRM.

### 3.8.6.1 Notes

Depending upon the targeted metrics, users can elect to implement a periodic software test of static configuration registers in the PRCM and control modules. Such a test can provide additional diagnostic coverage for disruption by soft error.

## 3.8.7 CPU Subsystems

### 3.8.7.1 C66x™ DSP Subsystem

#### 3.8.7.1.1 CPU Diagnostic

The C66x DSP subsystem implements one C66x DSP instantiation. To detect faults in the CPU internal blocks (ALU, register bank, floating-point unit, and so forth), TI recommends implementing appropriate software strategies. Redundant calculation on two different CPU subsystems or software-diversified redundant calculation on the C66x DSP subsystem of safety-critical portions of the application could be implemented by the system.

#### 3.8.7.1.2 Illegal Operation and Instruction Trapping

The C66x DSP includes diagnostics for illegal operations and instructions that can serve as safety mechanisms. Many of these traps are not enabled after reset and must be configured by the software. TI recommends installing software handlers to support the hardware illegal operation and instruction trapping. Examples of CPU illegal operation and instruction traps include the following:

- Illegal instruction
- Illegal behavior within an instruction
- Resource conflicts

#### 3.8.7.1.3 L1 and L2 Memory System

The C66x DSP subsystem implements a separate L1 instruction (L1P) and data cache (L1D) and a unified L2 cache for instruction and data. The L1P cache is protected by 1 parity bit for 256 data bits. The L1D cache is not protected by any dedicated diagnostic hardware mechanisms to detect faults. The L2 cache implements 10 parity bits per 256 data bits for Single Error Correction Double Error Detection (SECDED). CPU accesses, but also certain accesses of other masters (DMA and IDMA) to the SRAM and Cache can use these diagnostic measures. TI recommends enabling the parity and SECDED feature for the application.

#### 3.8.7.1.4 Memory Protection Architecture

The C66x megamodule memory protection architecture provides these benefits through a combination of CPU privilege levels and a memory system permission structure.

Code running on the CPU executes in one of two privilege modes: supervisor mode or user mode. Supervisor code is considered more trusted than user code. Examples of supervisor threads include operating system kernels and hardware device drivers. Examples of user threads include vocoders and end applications.

Supervisor mode is generally granted access to peripheral registers and the memory protection configuration. User mode is generally confined to the memory spaces that the OS specifically designates for its use.

CPU accesses, as well as internal DMA and other accesses, have a privilege level associated with them. The internal DMA accesses that are initiated by the CPU inherit the privilege level of the CPU at the time they are initiated.

The C66x memory protection architecture divides the DSP internal memory (L1P, L1D, and L2) into pages. An associated set of permissions for each page can be set differently for each requestor ID. For more information, see the TMS320C66x DSP Megamodule Reference Guide.

TI recommends using the features of the memory protection architecture in the application.



### 3.8.7.1.5 CPU Memory Management Unit (MMU)

The C66x DSP subsystem includes two Memory Management Units (MMUs), on the EDMA L2 interconnect and DSP MDMA paths, for accessing the device L3\_MAIN interconnect address space. The MMUs enable mapping of only the necessary application space to the processor.

Both DSP MMUs generate interrupts that are internally mapped to the DSP interrupt controller and output to the device IRQ crossbar. Both DSP MMUs (on MDMA and EDMA paths) have identical functionalities, as follows:

- 32-bit input and output address width (to match L3\_MAIN address width)
- 32 TLB cache entries
- 32 + 1 tags
- 128-bit data bus for MDMA and EDMA ports

For more detailed information, see the device-specific TRM. TI recommends using the features of the MMUs in the application.

### 3.8.7.1.6 Internal or External Watchdog

An internal or external watchdog can provide secondary diagnostic. For more information on these diagnostics, see [Section 3.8.4.2](#) or [Section 3.8.4.3](#).

## 3.8.7.2 Cortex-M4 IPU Subsystem

The Cortex-M4 IPU subsystem consists of two independent Cortex-M4 CPUs working off a unified cache for instruction and data that serves both CPUs. The unicache subsystem and all other relevant memories of the IPU are ECC protected.

### 3.8.7.2.1 CPU Diagnostics

The Cortex-M4 IPU subsystem implements two Cortex-M4 instantiations. Both CPUs are working independently from each other. To detect faults in the CPU internal blocks (ALU, register bank, floating-point unit, and so forth), TI recommends implementing appropriate software strategies. The system can implement redundant calculation on two different CPU subsystems or software-diversified redundant calculation on the Cortex-M4 subsystem of safety-critical portions of the application.

### 3.8.7.2.2 Illegal Operation and Instruction Trapping

The Cortex-M4 CPU includes diagnostics for illegal operations and instructions that can serve as safety mechanisms. Many of these traps are not enabled after reset and must be configured by software. TI recommends installing software handlers to support the hardware illegal operation and instruction trapping. Examples of CPU illegal operation and instruction traps include the following:

- An undefined instruction
- An illegal unaligned access
- Invalid state on instruction execution
- Errors on exception return
- Unaligned addresses on word and halfword memory accesses
- Division by zero

### 3.8.7.2.3 Unified Cache and MMU

The Cortex-M4 IPU subsystem implements a unified cache for instruction and data. The cache (SECDED) ECC-protected to detect and fix memory bit error transient faults. Other strategies can be used if strict isolation is required at task level, as follows:

- TI recommends flushing the cache in between execution of redundant programs.
- Storing data twice in the main memory also leads to duplicates in the data cache. A comparison of the redundant data before its use can detect faults in the cache.
- Mark regions of code or data that are safety-critical as noncacheable in the MMU.

The unified cache includes an MMU. The MMU logic can be used to provide spatial separation of software tasks in the device memory. It is expected that the operating system controls the MMU and changes the settings based on the needs of each task. A violation of a configured memory protection policy results in a CPU abort. TI recommends using the MMU.

Software-based testing of the MMU for proper operation and error response is optional.

#### **3.8.7.2.4 Online Profiling Using Data Watchpoint and Trace Unit (DWT)**

The Cortex-M4 CPU includes a performance monitoring unit called Data Watchpoint and Trace (DWT). This logic is intended to be used for debug and code profiling purposes, but it can also be used as a safety mechanism. The DWT includes a CPU cycle counter as well as five additional counters, which can be programmed to count a number of different CPU events. For a complete list of CPU events that can be monitored, see the Cortex-M4 TRM at <http://infocenter.arm.com/help/topic/com.arm.doc.subset.cortexm.m4/index.html#cortexm4>. Examples of the CPU events that can be monitored include:

- Folded instructions
- Load store unit (LSU) operations
- Sleep cycles
- CPI (all instruction cycles except for the first cycle)
- Interrupt overhead

With such information available, it is possible to generate a software routine that periodically checks the DWT counter values and compares these values to the profile expected during normal operation. The DWT is not enabled by default and must be configured using software. Use of the DWT for online diagnostic profiling is optional.

#### **3.8.7.2.5 Internal or External Watchdog**

An internal or external watchdog can provide secondary diagnostic. For more information on these diagnostics, see [Section 3.8.4.2](#) or [Section 3.8.4.3](#).

### **3.8.7.3 EVE Subsystem**

#### **3.8.7.3.1 Diagnostic Measures**

The EVE subsystem implements a single EVE instantiation. To detect faults in the EVE internal blocks (ARP32, VCOP, EDMA, and so forth), TI recommends implementing appropriate software strategies. Redundant calculation on two different subsystems or software diversified redundant calculation on the EVE subsystem of safety-critical portions of the application could be implemented by the system.

#### **3.8.7.3.2 Illegal Operation and Invalid Instruction Detection**

The EVE subsystem includes diagnostics for invalid instruction detection on ARP32 and VCOP, and illegal address detection on the EVE interconnect operations that can serve as safety mechanisms. Many of these features may not be enabled after reset and must be configured by software. TI recommends installing software handlers to support the hardware illegal operation and instruction trapping.

#### **3.8.7.3.3 Internal Memory System**

The EVE subsystem implements the following memories:

- P\$/PMEM: ARP32 program cache
- DMEM: ARP32 data memory
- WBUF: VCOP working buffer
- IBUFLA: image buffer low copy A
- IBUFLB: image buffer low copy B
- IBUFHA: image buffer high copy A

- IBUFHB: image buffer high copy B

The EVE subsystem supports parity-based error detection on all of the previously mentioned memories on the minimum access size granularity of 8 bits, therefore 1 bit of parity per byte of memory. The EVE subsystem supports the following:

- Single error detect (parity bit per byte) on DMEM, WBUF, IBUF\*
- Double bit error detect on program cache
  - Distance 3 hamming code – detection only, no correction
  - 10 bits per 256-bit cache line. The 10-bit hamming code is also applied to the tag and the address for a particular cache line.

TI recommends enabling the parity feature for the application.

### **3.8.7.3.3.1 Memory Management Unit**

The EVE subsystem includes two MMUs. This allows for virtual addresses to map critical buffers to different memory regions, and allows isolation among different tasks running on EVE.

TI recommends using MMUs.

### **3.8.7.3.3.2 Internal or External Watchdog**

An internal or external watchdog can provide secondary diagnostic. For more information on these diagnostics, see [Section 3.8.4.2](#) or [Section 3.8.4.3](#).

### **3.8.7.3.3.3 Locking Mechanism For Control Registers**

The EVE subsystem provides a lock and unlock mechanism that helps to prevent unintended access to the various control registers of the EVE control module, or EVE subcomponents. Ten lock and unlock registers are defined. Each register is used to lock and unlock access to a specific area of the memory map. See the TRM for more details. TI recommends using the lock and unlock mechanism.

### **3.8.7.3.4 Hardware-Assisted Software Self-Test – MISRs**

To facilitate software self-test, MISRs are instantiated on address and data buses at key points in the system, such as ARP32 interfaces and the Interconnect/WBUF interface. ARP32 is covered because it is the key control engine. WBUF coverage is provided as a convenient central destination that can be used to indirectly provide coverage for a majority of EVE logic.

The MISRs monitor the address and/or data buses and calculate a signature based on the data/address pattern on valid address or data phases. The signature registers reset to a value of 0x0. A different seed value can be manually written through software to each signature register. Based on a known memory access data pattern, the MISR signature can be predicted or calculated (or recorded on a known good system) and used as a reference for subsequent tests that can occur at boot time or during runtime in a safety-critical application.

The MISR calculation is a shift-register/XOR tree calculation using classical CRC algorithms.

TI recommends using the hardware-assisted software self test feature. More details of this feature are available in Section 6.1.3.12 of the TRM.

### **3.8.7.3.5 Error Recovery – ARP32 and OCP Disconnect**

To prevent runaway code from corrupting the remainder of the system, and provide a clean reset/recovery mechanism, the EVE subsystem provides a mechanism to disconnect ARP32 from the remainder of the EVE subsystem, and to disconnect L3 initiator buses from the remainder of the device.

When ARP32 or OCP initiator buses are disconnected, the MPU or debugger can see the EVE MMRs and memories through the interconnect target bus.

When the ARP32 and OCP buses are disconnected, a full reset and reboot cycle are issued to resume normal ARP32<-> EVE operation. This is required to avoid any asynchronous timing paths due to asynchronous reset assertion to ARP32.

Software must wait for the disconnected state before issuing a reset to the ARP32 core. This ensures that the neighboring system is not in a corrupted state. TI recommends enabling this error recovery feature in the application.

### **3.8.8 Network-on-Chip L3 Interconnect Subsystem**

The L3 interconnect is an instantiation of the network-on-chip® (NoC) interconnect.

#### **3.8.8.1 Diagnostic Measures**

TI recommends implementing the appropriate software strategies to perform a periodic dataflow-independent cyclical test of data paths. Defined test patterns to compare observations with corresponding expected values could be implemented by the system.

#### **3.8.8.2 Time-out Monitoring**

The NoC interconnect incorporates a generic target time-out feature. The interconnect generates a time-out event when transactions take too long to execute, either because a request exceeded a time threshold before being accepted by the target, or because a response exceeded a time threshold issued by the target since the corresponding request. TI recommends using the time-out feature for interconnect monitoring.

#### **3.8.8.3 Statistics Collectors**

The NoC interconnect includes a performance monitoring unit that includes statistics collectors. This programmable unit is intended for performance monitoring capability by probing interconnect links, recording events, and transmitting results to a debug unit, but it can also be used as a safety mechanism.

It is possible to collect statistics for any of the initiators in the system.

#### **3.8.8.4 Quality of Service Units**

The NoC interconnect includes several quality of service (QoS) units like bandwidth regulators and bandwidth limiters for several key initiators in the system. These units are intended to ensure an average target bandwidth to an initiator or to prevent initiators from consuming too much bandwidth of a link, or a target, that is shared between dataflows, but they can also be used as safety mechanisms.

#### **3.8.8.5 Error Handling**

Error logging is enabled for the NoC interconnect.

The three major types of errors reported follow:

- Slave network interface unit (NIU) errors
- Firewall errors
- Flag mux errors

TI recommends that the software performs necessary actions when a NoC-reported error is received.

#### **3.8.8.6 Firewalls**

Firewalls provide a strong mechanism to implement FFI. The hardware firewalls ensure that no unintentional accesses are allowed to the isolated memory regions. For more details on the firewall setup and configuration, check chapter 10 of the device TRM.

### **3.8.9 On-Chip RAM Subsystem**

On-chip RAM (OCMC) is memory that can be used by multiple masters for data storage. ORAM is supported by an SECDED error correcting code (ECC) diagnostic.

Three OCM controllers (OCMCs) are associated with the ORAM. The RAM associated controllers are as follows:

- OCMC\_RAM1 with 512KB of dedicated memory space

### 3.8.9.1 Error Correcting Code (ECC)

The ORAM is supported by an SECDED ECC diagnostic. When enabled, the ECC generated is Hamming(155,146) code and has a Hamming distance of 4. The OCMC supports the following four modes of operation:

- Non-ECC mode (data access)
- Non-ECC mode (code access)
- Full-ECC mode
- Block-ECC mode

The default mode of operation for the OCMC is the non-ECC data access mode. For detailed information, see the device-specific TRM.

TI recommends enabling the ECC feature while using the OCMC RAMs. This diagnostic feature can be tested through the software sequences outlined in the TRM for the OCMC section.

### 3.8.9.2 Circular Buffer Mode Error Handling

The OCMC provides up to 12 programmable circular buffers (CBUFs) that are mapped to virtual video frames to support slide-based video frame processing. For each write or read access associated with the virtual frame buffer, the OCMC performs various address error checks to prevent illegal CBUF accesses from causing false overflow and underflow conditions.

The CBUF provides support for the following:

- Detection of VBUF address not mapped to a CBUF memory space
- Detection of VBUF access not starting at the base address
- Illegal address change between two same type accesses
- Detection of illegal frame size (short frame detection)
- Detection of CBUF overflow
- Detection of CBUF underflow

For detailed information regarding all CBUF events, see the device-specific TRM.

TI recommends using this feature.

### 3.8.9.3 Correctable ECC Profiling

The OCM RAMs include a capability to count the number of correctable ECC errors detected. This counter is 16 bits wide and keeps track of all SEC error events. When the error count exceeds a user programmed threshold, an exception (SEC error found) is asserted by the OCM RAM controller.

Three counters are used to count different types of errors that occur when the ECC mode is enabled. The counters are:

- SEC counter – for the single errors occurred
- DED counter – for double error detections
- ADDRERR counter – for address errors found when a single error occurs

For detailed information, see the device-specific TRM.

TI recommends using of this feature.

### 3.8.10 General-Purpose Timer Subsystem

The general-purpose (GP) timer can be used for operating system scheduling and other timer functionality. The logic block does not implement diagnostic features to ensure the correctness of the timer and the corresponding interrupt generation. For consistency checks, a second timer can be used. This could be implemented by reading the primary and secondary counter values and comparing them. Another possibility to do some consistency checks would be to generate two independent interrupts and checking the plausibility of both. This scheme would ensure a scenario in which the primary interrupt might not be generated or recognized by the system. Care must be taken to account for possible jitter due to program execution differences (cache versus noncached accesses) or other exceptions that may interrupt reading the primary and secondary counter values.

#### 3.8.10.1 Internal or External Watchdog

An internal or external watchdog can provide secondary diagnostic. For more information on these diagnostics, see [Section 3.8.4.2](#) or [Section 3.8.4.3](#).

### 3.8.11 Interprocessor Communication (IPC)

The IPC is used to communicate between the different CPUs in this heterogeneous architecture. The IPC consists of mailboxes and spinlocks.

#### 3.8.11.1 Mailboxes

The queued mailbox-interrupt mechanism lets the software establish a communication channel between two processors through a set of registers and associated interrupt signals by sending and receiving messages (mailboxes). Three mailbox module instances are in the device:

- System mailbox (two instances) – used for communications across various CPU cores on the device (DSP and IPU subsystem)
- EVE mailbox (two mailboxes per EVE instance) – used for communication between a) EVE local user (ARP32) and three external users (selected among DSP1 and DSP2\*).

The user can develop test patterns to test the functionality of the mailboxes at start-up or shutdown to ensure the functionality of the mailbox. During application run time, plausibility checks can be made when receiving interrupts from the mailbox on the timing of the mailbox interrupt as well as on the data that is passed through the mailbox. Mailboxes have FIFO full and empty status registers that could be used to diagnose and prevent loss of messages. For detailed recovery mechanisms and use of mailbox for safety-critical message passing, refer to the device TRM.

#### 3.8.11.2 Spinlocks

The spinlock module provides hardware assistance for synchronizing the processes running on multiple processors in the device. The spinlock provides hardware semaphores to lock or unlock access to data structures in the application.

To ensure the functionality of the spinlock module, the user can develop test patterns to set the locks to a taken or not-taken state and afterwards check the state through another CPU read. These test patterns, which cover permanent faults, can be run at either system start-up or shutdown. A time-out mechanism could be implemented in software to test for transient errors. A transient error could set the lock to a taken state. Because no task has claimed the semaphore, the lock will be permanently set to taken. Other tasks will not be able to claim the semaphore, but could implement a time-out mechanism for the application to handle the situation appropriately. Another way to diagnose transient faults would be to assign two spinlocks to each data structure. [Table 3-12](#) provides the descriptions of the possible states and the error conditions.

**Table 3-12. Spinlock States**

State Returned by First Lock Read	State Returned by Second Lock Read	Comment
0	0	No error, data structure is safe to use.

**Table 3-12. Spinlock States (continued)**

State Returned by First Lock Read	State Returned by Second Lock Read	Comment
0	1	Either a transient error has cleared the first lock or software has not cleared the second lock. Time-out may be implemented to check the second lock multiple times to make sure the owning process has taken more time to clear it.
1	0	Either a transient error has cleared the second lock or software has not cleared the first lock. Time-out may be implemented to check the first lock multiple times to make sure the owning process has taken more time to clear it.
1	1	No error, data structure is used by another task.

### 3.8.12 Serial Peripheral Interface (SPI)

The SPI modules provide serial I/O compliance to the SPI protocol. SPI communications are typically used for communication to smart sensors and actuators, serial memories, and external logic, such as a watchdog device. The SPI modules contain internal SRAM buffers.

#### 3.8.12.1 System Test Mode

The SPI module supports a system test mode that lets the user test the internal interrupt connection as well as the external I/O connections. TI highly recommends implementing such a test at application start-up or shutdown.

#### 3.8.12.2 Information Redundancy Techniques

Information redundancy techniques can be applied through software as an additional run-time diagnostic for SPI communication. Many techniques can be applied, such as read back of written values and multiple reads of the same target data with comparison of results. Alternatively redundancy can be achieved by implementing multiple channels in the system. Error response, diagnostic testability, and any necessary software requirements are defined by the software implemented by the system integrator. TI recommends using information redundancy techniques in McSPI transactions.

### 3.8.13 Controller Area Network

The DCAN interface provides medium throughput networking with event-based triggering, compliant to the controller area network (CAN) protocol. The DCAN module requires an external transceiver to operate on the CAN network.

#### 3.8.13.1 Software Test Function Using I/O Loopback

A software test can be used to inject diagnostic errors and check for proper error response. Such a test can be executed at boot or periodically. The software requirements are defined by the software implemented by the system integrator. TI recommends using the boot time software test of basic functionality. The use of a periodic software test of basic functionality reporting is optional.

The DCAN implementation supports both digital and analog loopback capabilities for the I/Os. Digital loopback tests the signal path to the module boundary. Analog loopback tests the signal path from the module to the I/O cell with output driver disabled. For best results any tests of the DCAN functionality should include the I/O loopback.

There may be other I/O functionalities where the analog PHY is implemented outside the TDA3x device (such as Ethernet and other similar interfaces). For scenarios in which diagnostic coverage is required for the customer use scenarios, TI strongly recommends implementing a board test scenario using loopback.

#### 3.8.13.2 Information Redundancy Techniques Using End-to-End Safing

Information redundancy techniques can be applied using software as an additional run-time diagnostic for CAN communication. Many techniques can be applied, such as read back of written values and multiple reads of the same target data with comparison of results.

To provide diagnostic coverage for network elements outside the device (wiring harness, connectors, transceiver) end-to-end safing mechanisms are applied. These mechanisms can also provide diagnostic coverage inside the device. Many different schemes are applied, such as additional message checksums, redundant transmissions, time diversity in transmissions, and so forth. Most commonly, checksums are added to the payload section of a transmission to ensure the correctness of a transmission. These checksums are applied in addition to any protocol level parity and checksums. As the checksum is generated and evaluated by the software at either end of the communication, the whole communication path is safed, resulting in end-to-end safing.

The system integrator defines error response, diagnostic testability, and any necessary software requirements. TI highly recommends using this mechanism.

The end-to-end, black-channel communication may still lead to missed messages at the system level. Therefore, the communication link and software protocol must ensure that the sender receives an acknowledge from the destination of the CAN message.

### **3.8.13.3 DCAN SRAM ECC**

The DCAN SRAM includes a SECDED ECC diagnostic that can detect and correct bit errors in the memory. This feature is disabled after reset. Software must configure and enable this feature. TI recommends using the DCAN SRAM parity feature.

### **3.8.13.4 DCAN SRAM testing**

The DCAN SRAM contents can be tested periodically using appropriate memory tests (for example, March13N). TI recommends using this diagnostic at application start-up or shutdown. Because DCAN SRAM contents tend to be more dynamic, use of this diagnostic during normal operation is optional.

## **3.8.14 (LP)DDR2/3 Memory Controller (EMIF)**

The DDR2/3 memory controller is used to interface with JESD79-2E/JESD79-3C standard-compliant DDR2/3 SDRAM devices, respectively. Memory types such as DDR1 SDRAM, SDR SDRAM, SBSRAM, and asynchronous memories are not supported.

### **3.8.14.1 ECC**

For data integrity, the EMIF1 supports ECC on the data written or read from the SDRAM. The users must enable the ECC feature by writing to appropriate registers inside the EMIF subsystem. ECC accesses are allowed for both SYS and the MPU ports. A 7-bit ECC is calculated over 32-bit data when in 32-bit DDR mode. 6-bit ECC is calculated over 16-bit data when in 16-bit DDR mode. The ECC is calculated for all accesses that are within the address ranges protected by ECC. These address ranges are software configurable. The ECC must be enabled and only aligned writes with byte count in multiples of 8 bytes (ECC quanta) must be used to preload the ECC protected region. The ECC is read and verified during reads. For detailed information, see the device specific TRM. This diagnostic feature can be tested through the software sequences outlined in the OCMC section of the TRM.

### **3.8.14.2 Correctable ECC Profiling**

The EMIF1 includes a capability to count the number of correctable ECC errors detected. When the error count exceeds a user-programmed threshold, an interrupt is generated by the EMIF controller. The software can use this interrupt to gauge the degree of 1-bit ECC errors occurring in the system.

The EMIF also supports a 1-bit ECC data error distribution register that represents whether an error has occurred in a given data channel location. This is advantageous to detect whether errors are random or permanent.

For 2-bit ECC errors in the data, the EMIF generates a 2-bit error interrupt. For any bit errors in the address, the EMIF generates an address error interrupt.

For detailed information, see the device-specific TRM.

TI recommends using this feature.



### 3.8.14.3 Use of Performance Counters

The EMIF controller also supports two performance counters to let users monitor or calculate the EMIF controller bandwidth and efficiency. These counters are able to count events such as total SDRAM accesses, SDRAM activates, reads, writes, and other events. Each counter counts independent of the other. This programmable unit is intended for performance monitoring capability, but it can also be used as a safety mechanism.

For detailed information, see the device-specific TRM.

Use of this feature is optional.

### 3.8.15 Enhanced Direct Memory Access (EDMA)

The enhanced direct memory access (EDMA) module is used to move data from one location to another inside the system. This is typically used for peripheral configuration (SRAM to peripheral transfer), peripheral data update (peripheral buffer memory transfer to SRAM for processing) and memory to memory transfers. The DMA is typically used by the operating system to offload bus transactions from the CPU to improve overall system performance. The DMA has a local SRAM that is used for channel control information.

#### 3.8.15.1 Memory Protection

The EDMA3 channel controller supports two kinds of memory protection: active and proxy.

Active memory protection is a feature that allows or prevents read and write accesses (by any EDMA3 programmer) to the EDMA3 Channel Controller Register (EDMA3CC) (based on permission characteristics that you program).

Proxy memory protection allows an EDMA3 transfer programmed by a given EDMA3 programmer to have its permissions travel with the transfer through the EDMA3 Transfer Controller (EDMA3TC). The permissions travel along with the read transactions to the source and the write transactions to the destination endpoints.

TI recommends using the memory protection techniques.

#### 3.8.15.2 Error Detection

Errors are generated, if enabled, under the following three conditions:

- EDMA3TC detection of an error signaled by the source or destination address
- Attempt to read or write to an invalid address in the configuration memory map
- Detection of a constant addressing mode TR violating the constant addressing mode transfer rules (the source and destination addresses and source and destination indexes must be aligned to 32 bytes.)

TI recommends enabling these error detection features.

#### 3.8.15.3 Information Redundancy Techniques

Information redundancy techniques can be applied using the EDMA module. Many techniques can be applied, such as read back of written values and multiple reads of the same target data with comparison of results.

Error response, diagnostic testability, and any necessary software requirements are defined by the software implemented by the system integrator. TI recommends using the implementation of information redundancy techniques on EDMA transactions.

#### 3.8.15.4 Parameter Memory Testing

The EDMA3 controller is a RAM-based architecture. The transfer context (source and destination addresses, count, indexes, and so forth) for DMA or QDMA channels is programmed in a parameter RAM table within EDMA3CC, referred to as PaRAM.

The PaRAM contents can be tested periodically using appropriate memory tests (for example, March13N). TI recommends using this diagnostic at application start-up or shutdown. Because PaRAM contents tend to be static, TI recommends performing a periodic CRC check of the PaRAM during run time of the application.

### 3.8.15.5 Periodic Read Back of Configuration Registers

Periodic read back of configuration registers can provide a diagnostic for inadvertent writes or disturb of these registers. Error response, diagnostic testability, and any necessary software requirements are defined by the software implemented by the system integrator. TI recommends using the read back of configuration registers mechanism.

### 3.8.15.6 Optional use of MMU1

MMU1 on the device is dedicated to EDMA Transfer Controller 0 (TC0) and EDMA Transfer Controller 1 (TC1). Requests initiated by EDMA TC0 and TC1 (both read and write ports) for system MMU1 can optionally be routed through MMU1. Use of MMU1 by EDMA TC0 and TC1 is independently controllable using the control module.

TI recommends using MMU1.

### 3.8.16 Video Input Port (VIP)

The VIP subsystem does not provide any dedicated hardware diagnostics to detect faults in the module. TI recommends that the user implements dedicated measures on the system level.

#### 3.8.16.1 VIP Overflow Detection and Recovery

An overflow is possible in the VIP\_PARSER. Overflow detection is determined by reading one of the VIP status registers. The status register bits can indicate if not all of the incoming video data was sent to DDR. For detailed information, see the device-specific Technical Reference Manual.

### 3.8.17 Video Processing Engine (VPE)

The VPE subsystem does not provide any dedicated hardware diagnostics to detect faults in the module. TI recommends that the user implements dedicated measures on the system level.

### 3.8.18 Display Subsystem (DSS)

The DSS subsystem does not provide any dedicated hardware diagnostics to detect faults in the module. TI recommends that the user implements dedicated measures on the system level. These diagnostics should target the specific safety goals required by the system. These goals, for example, may include latency requirements for the sensor to display per frame as well as detection of frame freeze or bad display regions. The diagnostics are designed in software.

### 3.8.19 Inter-Integrated Circuit

The interintegrated circuit (I2C) module provides a multimaster serial bus compliant to the I<sup>2</sup>C protocol.

#### 3.8.19.1 Software Test Function

A software test can be used to test basic functionality as well as to inject diagnostic errors and check for proper error response. Such a test can be executed at boot or periodically. The software requirements are defined by the software implemented by the system integrator. TI recommends using a boot time software test of basic functionality. The use of a periodic software test of basic functionality reporting is optional.

#### 3.8.19.2 Information Redundancy Techniques

Information redundancy techniques can be applied using software as an additional run-time diagnostic for I2C communication. Many techniques can be applied, such as read back of written values and multiple reads of the same target data with comparison of results.

The software implemented by the system integrator defines the error response, diagnostic testability, and any necessary software requirements.

TI recommends using information redundancy techniques in I2C transactions.

### 3.8.20 General-Purpose Input/Output (GPIO)

The GPIO module provides digital input capture and digital I/O. There is no processing function in this block. The GPIO is typically used for static or rarely changed outputs, such as transceiver enable signals, and so forth. The GPIO can also be used to provide external interrupt input capabilities.

#### 3.8.20.1 Software Test of Function Using I/O Checking

A software test can be used to test basic functionality as well as to inject diagnostic errors and check for proper error response. Such a test can be executed at boot or periodically. The software requirements are defined by the software implemented by the system integrator. TI recommends using a boot time software test of basic functionality. The use of a periodic software test of basic functionality reporting is optional.

The GPIO module does not support a distinct I/O loopback mode. However, it is possible to support I/O checking using normal functionality. To do this, software generates output and reads back and checks for the same value in the input registers. For best results, any tests of the GPIO functionality should include the I/O loopback.

For highly safety-critical operations, TI recommends using redundancy on the GPIO controls.

#### 3.8.20.2 Information Redundancy Techniques

Information redundancy techniques can be applied via software as an additional run-time diagnostic on GPIO function. Many techniques can be applied, such as multiple inputs and read back of output with an input channel. Signals from many other peripherals can be used as GPIO if not used for primary function. Use of a GPIO module signal and a non-GPIO module signal for multichannel implementation can reduce probability of common mode failures.

The software implemented by the system integrator defines the error response, diagnostic testability, and any necessary software requirements.

TI recommends using information redundancy techniques on GPIO functions.

#### 3.8.20.3 Notes

To reduce the probability of common mode failure, users must consider implementing multiple channels using nonadjacent pins.

### 3.9 Tester On Chip (TesOC)

The tester on chip (TesOC) provides the capability of the device to software initiate structural (logic and memory) testing on cores and safety-critical memories. The TesOC module has the following features:

- Supports software-initiated field tests during device start-up
- Supports structural Logical Build-In Self-Test (LBIST) on EVE, DSP, IPU with up to 90% stuck-at coverage
- Supports memory test of critical memories on EVE, DSP, IPU, ISS and DSS
- Provides built-in ROM with test vectors to achieve up to 90% coverage on EVE, DSP, and 85% on IPU. TI recommends using functional tests in addition to the logic BIST capability of TeSOC to augment coverage. These tests should be designed by the customer such that it reflects the actual functional software instruction sequence that runs on a given CPU.
- Provides an interconnect interface with 128 data width Read/Write with burst write

For more information on TeSOC use and programming model, see Section 24.2 of the device TRM.

### 3.10 Memory Cyclic Redundancy Check Module

The memory cyclic redundancy check (MCRC) module within the device is designated to perform memory checks to verify the integrity of memory system. In the background, the MCRC reads memory contents and generates signature representing them. Software must be written by the system integrator to enable this diagnostic. If the software is written, then the MCRC must calculate these signatures and generate interrupt to MPU for comparison against predetermined good signature value. The interrupt again must be handled in software by the system integrator. The module provides four channels to perform CRC calculation on multiple memories in parallel and can be used with any memory system. The following features are supported by the MCRC module:

- Four channels to perform background signature verification on any memory subsystem
- Data compression on 8-, 16-, 32-, and 64-bit data size with active data trace mode
- Maximum-length of the parallel signature analysis (PSA) register constructed based on 64-bit primitive polynomial
- Each channel has its own CRC registers containing the predetermined CRC value.
- Each channel has its own programmable 20-bit pattern counter to count the number of data patterns for compression.
- Semi-CPU and full-CPU operation modes
- Generate interrupt to CPU in semi-CPU mode to allow CPU to perform signature verification.
- Generate time-out interrupt if CRC is not performed within the time limit.

For more information on MCRC use and programming model, see Section 24.3 of the device TRM.

### 3.11 Dual-Clock Comparator

The dual-clock comparator (DCC) is used to determine the accuracy of a clock signal during the time execution of an application. Specifically, the DCC is expected to detect 2% drift (depending upon the clock ratio setup) from the expected frequency within a period of 100 ms. Moreover, the exact accuracy can also be programmed based on calculation for each application. The DCC can also measure the frequency of a selectable clock source using another input clock as a reference.

Seven instances of the DCC within the device specifically are targeted to perform verification on the DPLL\_CORE, DPLL\_DDR, DPLL\_PER, DPLL\_DSP, DPLL\_EVE, SYS\_CLK2, and one of the DPLL\_CORE HS-Dividers. It is system integrator's responsibility to set up the DCC configuration and clock ratios using software to monitor the quality of safety-critical clocks.

The main features of the DCC module include:

- Two independent counter blocks count clock pulses from each clock source.
- Each counter block is programmable, however, for proper operation the counters must be programmed with seed values that respect the ratio of the two clock frequencies.
- Error signal generation
- Clock frequency measurement

For more information on DCC use and programming model, see Section 24.4 of the device TRM.

### 3.12 Error Signaling Module

The error signaling module (ESM) is used to indicate a severe device failure at an external pad. The error pad is normally used as a second indication path to switch off (or reset) the device CPUs by an external device. Therefore the external controller can reset the device or keep the system in a fail-safe state by disabling the peripherals outside of the ECU. The following features are supported by the ESM.

- Up to 384 interrupt/error channels are supported, divided into the following three different groups:
  - 128 channels with configurable interrupt and error pad behavior
  - 128 error channels with predefined interrupt and error pad behavior
  - 128 channels with predefined error pad behavior
- Error pad to signal severe device failure

- Configurable timebases for error signal
- Error forcing capability

For more information on ESM use and programming model, see Section 24.5 of the device TRM.

### 3.13 Embedded ADC

The analog-to-digital converter (ADC) module is a successive-approximation-register (SAR) general-purpose ADC.

The main features of the ADC include the following:

- 10-bit data
- 750 KSPS at 13.5-MHz ADC\_CLK
- 8 channels
- Programmable FSM sequencer that supports 16 steps, as follows:
  - Software register bit for start of conversion
  - Single conversion (one-shot)
  - Continuous conversions
  - Sequence through all input channels based on a mask
  - Programmable OpenDelay before sampling each channel
  - Programmable sampling delay for each channel
  - Programmable averaging of input samples – 16/8/4/2/1
  - Store data in either of two FIFO groups – 64-deep each
  - Option to encode channel number with data
  - Support for servicing FIFOs through DMA or CPU
  - Programmable DMA Request event (for each FIFO)
  - Dynamically enable or disable channel inputs during operation
  - Stop bit to end conversion
  - Support for error offset (internal calibration or external calibration through eFuse) inside the AFE
- Support for the following interrupts and status, with masking:
  - Interrupt after a sequence of conversions (all nonmasked channels)
  - Interrupt for FIFO threshold levels
  - Interrupt if sampled data is out of a programmable range
  - Interrupt for FIFO overflow and underflow conditions
  - Status bit to indicate if ADC is busy converting

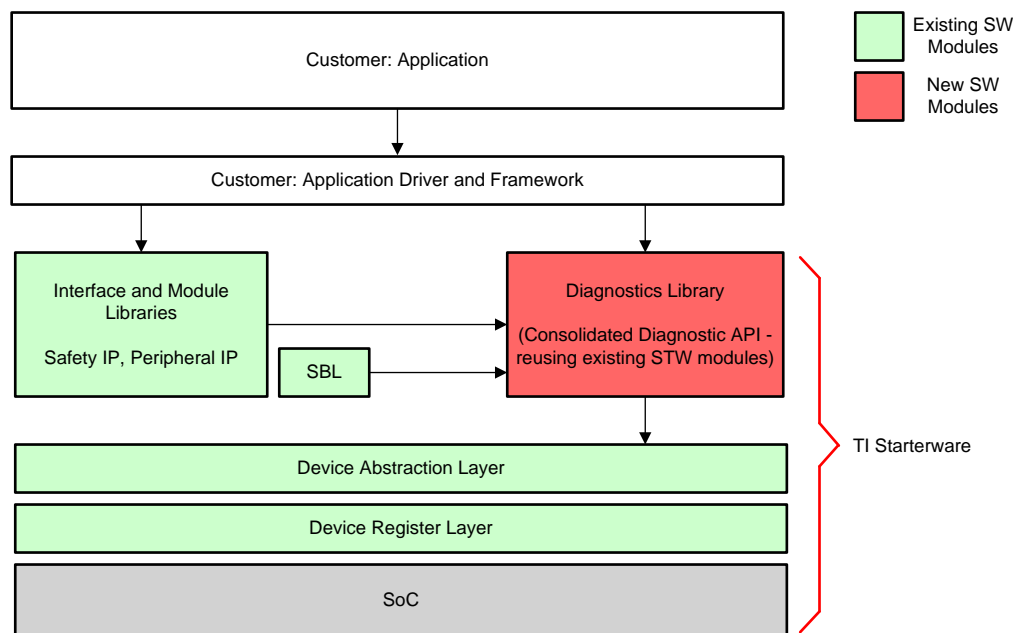
For more information on ADC use and programming model, see Section 23 of the device TRM.

## Software Diagnostic Library

TI is delivering a reference software diagnostic library as reference for customers to design their own diagnostics. The details of this library are captured in this section. This diagnostics library is a collection of software functions for diagnosing the health of various modules and interfaces of the device. The functions of this library can be called by the user at system start, run time, or shutdown. Based on the final system requirements, the system integrator can use these APIs to incorporate appropriate mechanisms in the final system to meet safety requirements.

### 4.1 TI Diagnostics Library Architecture

Figure 4-1 shows the architecture of the library. It is important to point out that diagLib will not save entry state and restore state before exit. It is assumed that the caller will restore context. This means the caller must call DiagLib after driver deinit and call driver init after DiagLib test.



**Figure 4-1. Diagnostics Library High-Level Architecture**

### 4.2 ECC Fault Injection Test

Figure 4-2 and capture the ECC fault injection concept and API.

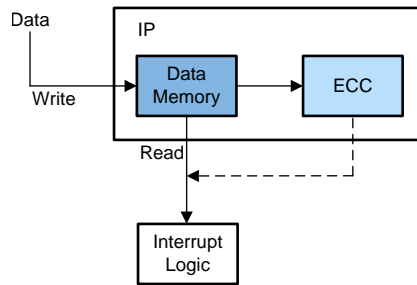


Figure 4-2. Testing ECC Logic

### 4.3 ADC Tests

Figure 4-3 and capture the ADC diagnostic concept and API.

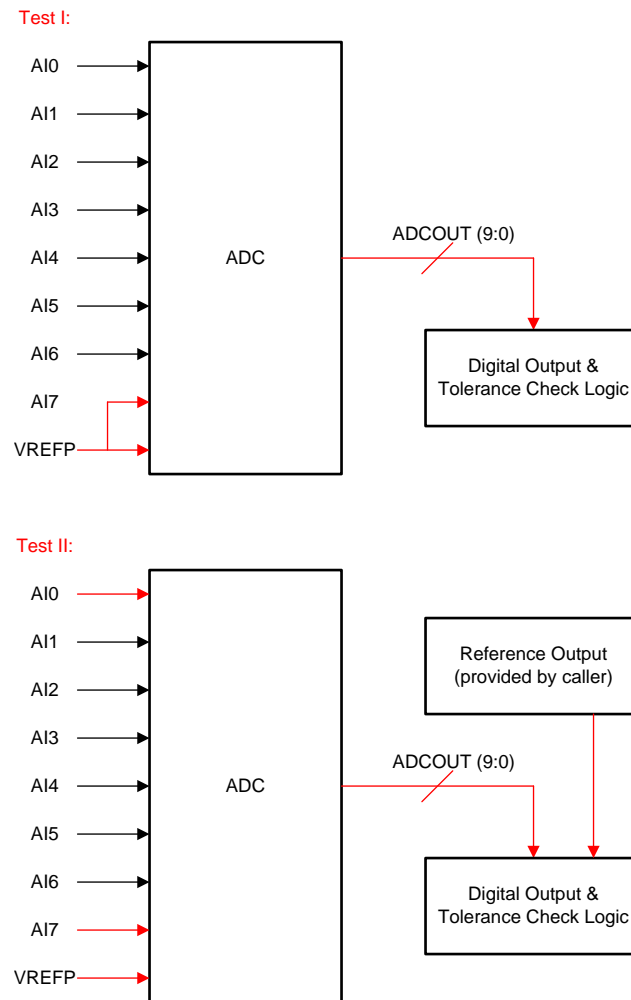


Figure 4-3. Using On-Chip ADC for System Monitoring

### 4.4 SPI API

Figure 4-4 captures the SPI diagnostic concept and API.

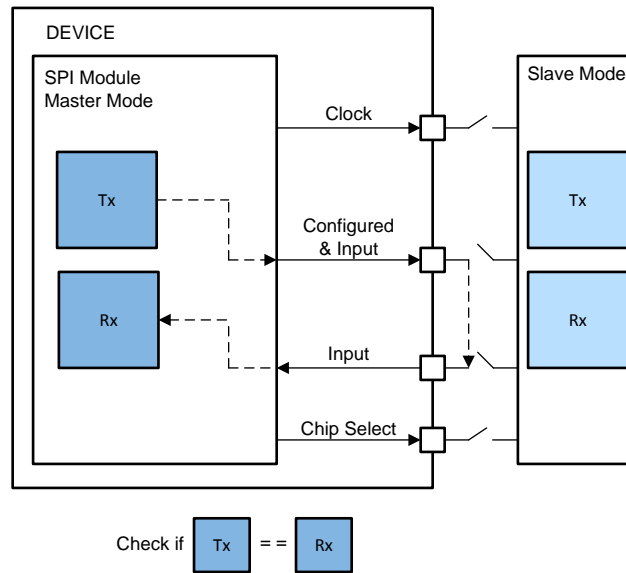


Figure 4-4. SPI Loopback Test Mechanism

#### 4.5 CRC API

Figure 4-5 and capture the CRC diagnostic concept and API.

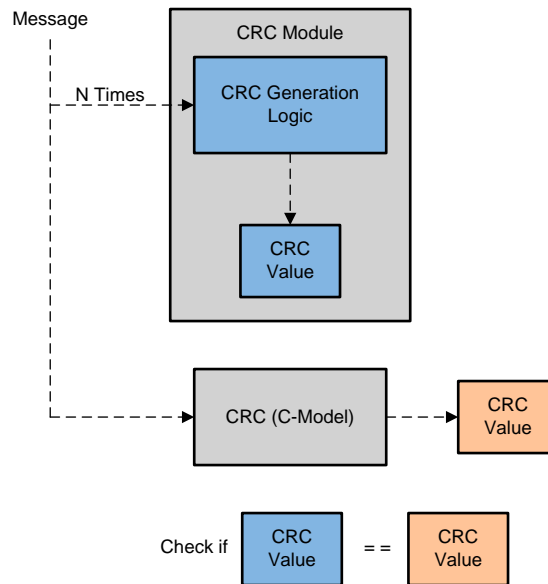


Figure 4-5. Using CRC to Check for Data Accuracy

#### 4.6 DCC API

Figure 4-6 and Figure 4-7 capture the DCC diagnostic concept and API.



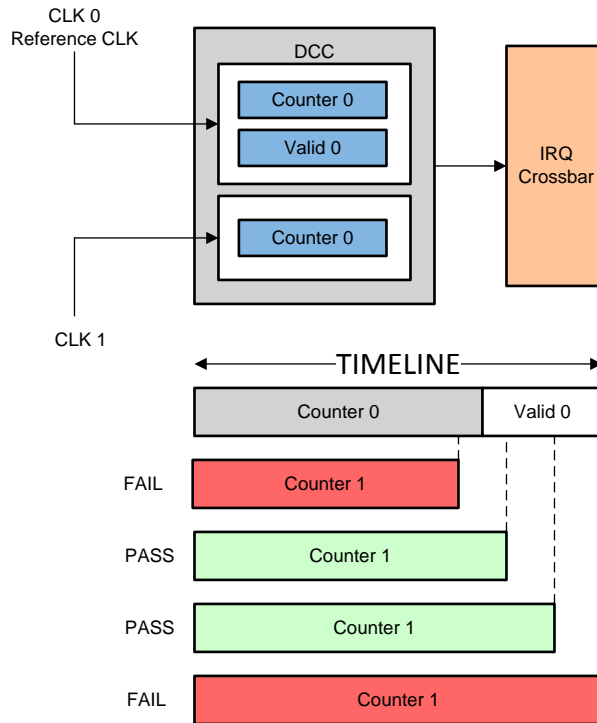


Figure 4-6. Using DCC for Clock Monitoring (1 of 3)

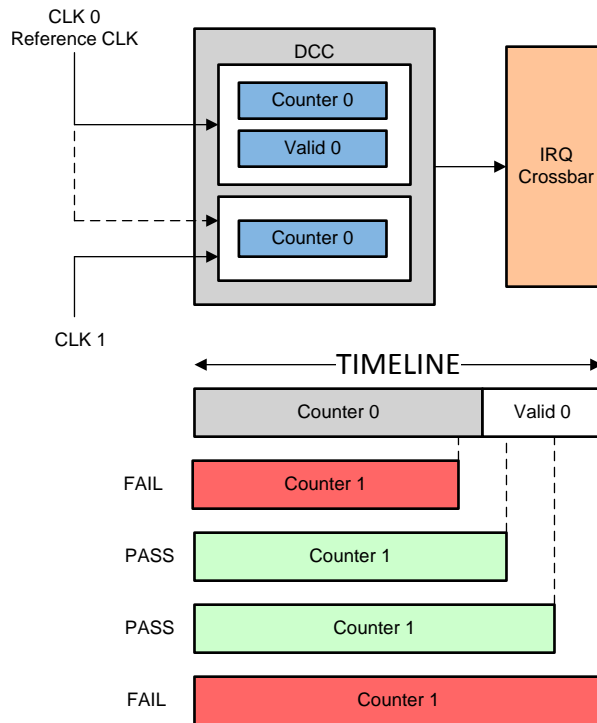


Figure 4-7. Using DCC for Clock Monitoring (2 of 3)

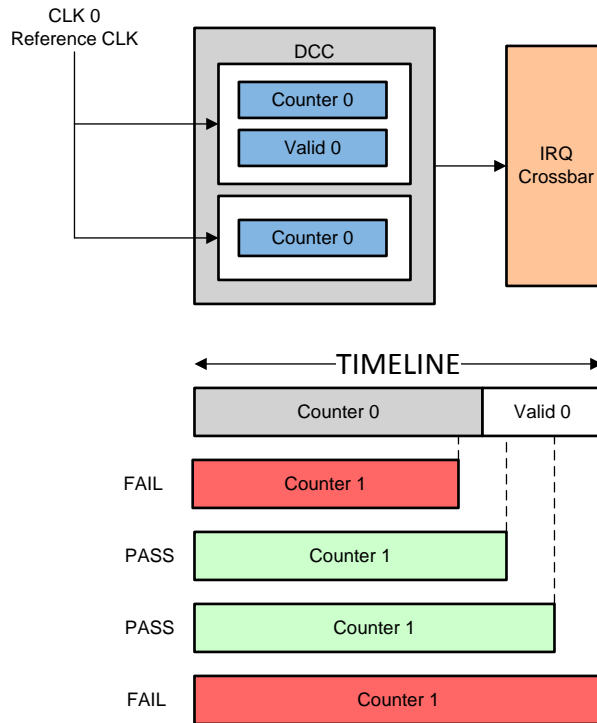


Figure 4-8. Using DCC for Clock Monitoring (3 of 3)

#### 4.7 ESM API

Figure 4-9, Figure 4-10, and Figure 4-11 capture the ESM diagnostic concept and API.

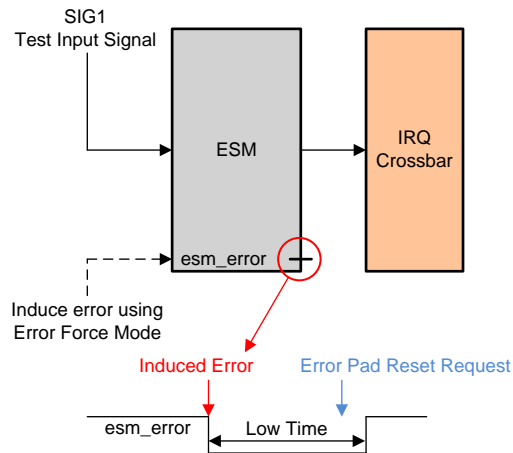


Figure 4-9. Error Interrupts for System Monitoring in ESM (1 of 3)

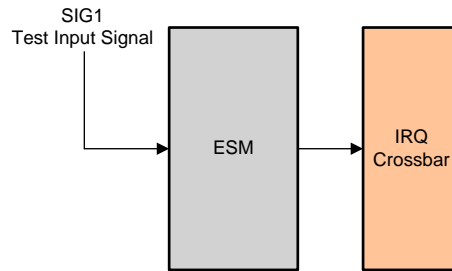


Figure 4-10. Error Interrupts for System Monitoring in ESM (2 of 3)

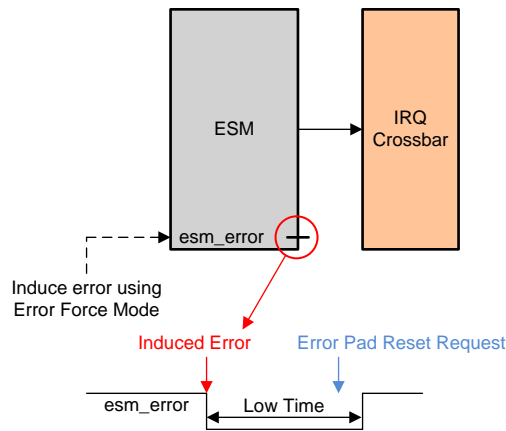


Figure 4-11. Error Interrupts for System Monitoring in ESM (3 of 3)

#### 4.8 DCAN Loopback API

Figure 4-12 and capture the CAN diagnostic concept and API.

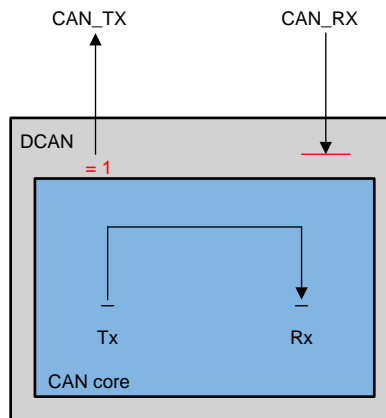


Figure 4-12. Loopback Mechanism for Testing DCAN IP

#### 4.9 QSPI (SPI for Boot in 4-Bit Data Mode)

Figure 4-13, , and Figure 4-14 capture the QSPI diagnostic concept and API.

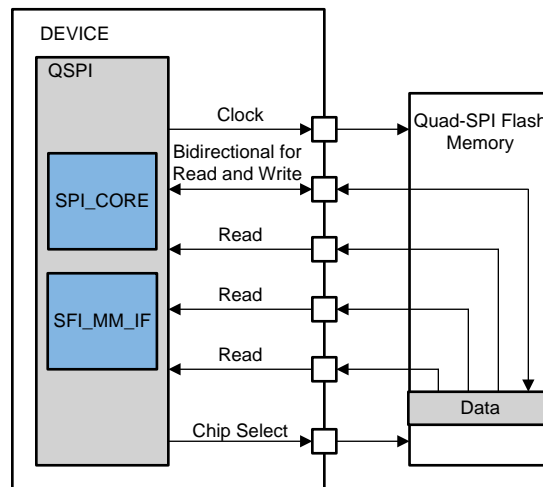


Figure 4-13. Confirming Correctness of Boot Over QSPI (1 of 2)

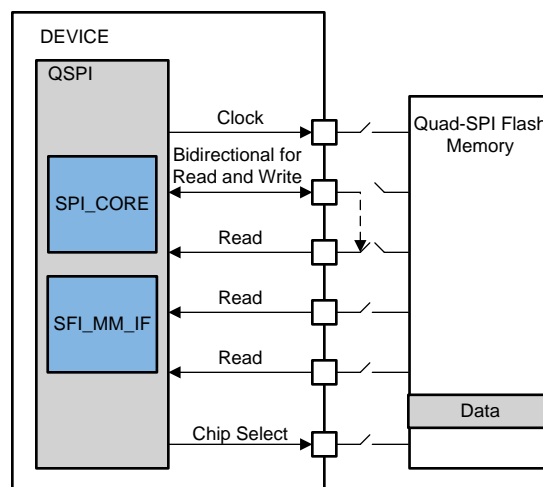


Figure 4-14. Confirming Correctness of Boot Over QSPI (2 of 2)

#### 4.10 TI AutoSAR TDA3x MCAL Architecture

The following subsections for AutoSAR integration must be treated as one possible guideline. Customers are responsible to do their own due diligence to ensure that their software and system architecture of choice fits the safety objectives of their system.

#### 4.11 Vision SDK AutoSAR Integration Proposal on TDA3x

Figure 4-15 describes the AutoSAR integration concept on TDA3x.

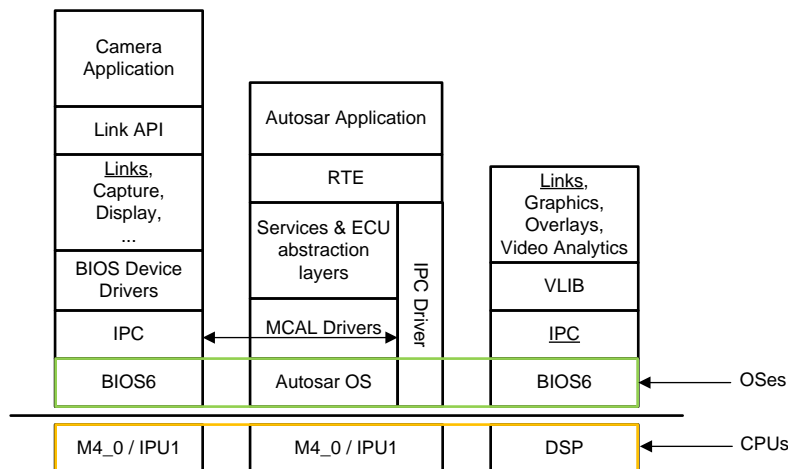


Figure 4-15. High-Level AutoSAR Integration

### 4.12 Vision SDK SBL Concept

Figure 4-16 describes the SBL (secondary bootloader) contents and concept on TDA3x. This clarifies the sequence of boot for MCAL drivers.

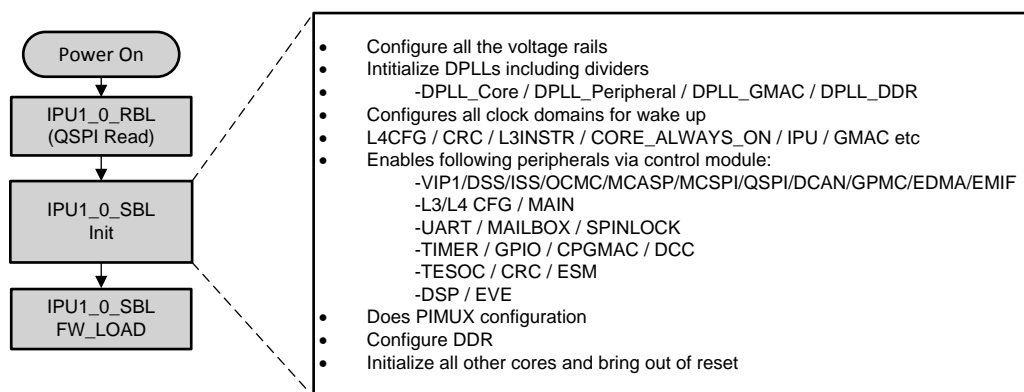


Figure 4-16. Safety-Enabled Boot Sequence

### 4.13 Vision SDK AutoSAR Boot Sequence

Figure 4-17 describes the AutoSAR boot sequence concept on TDA3x.

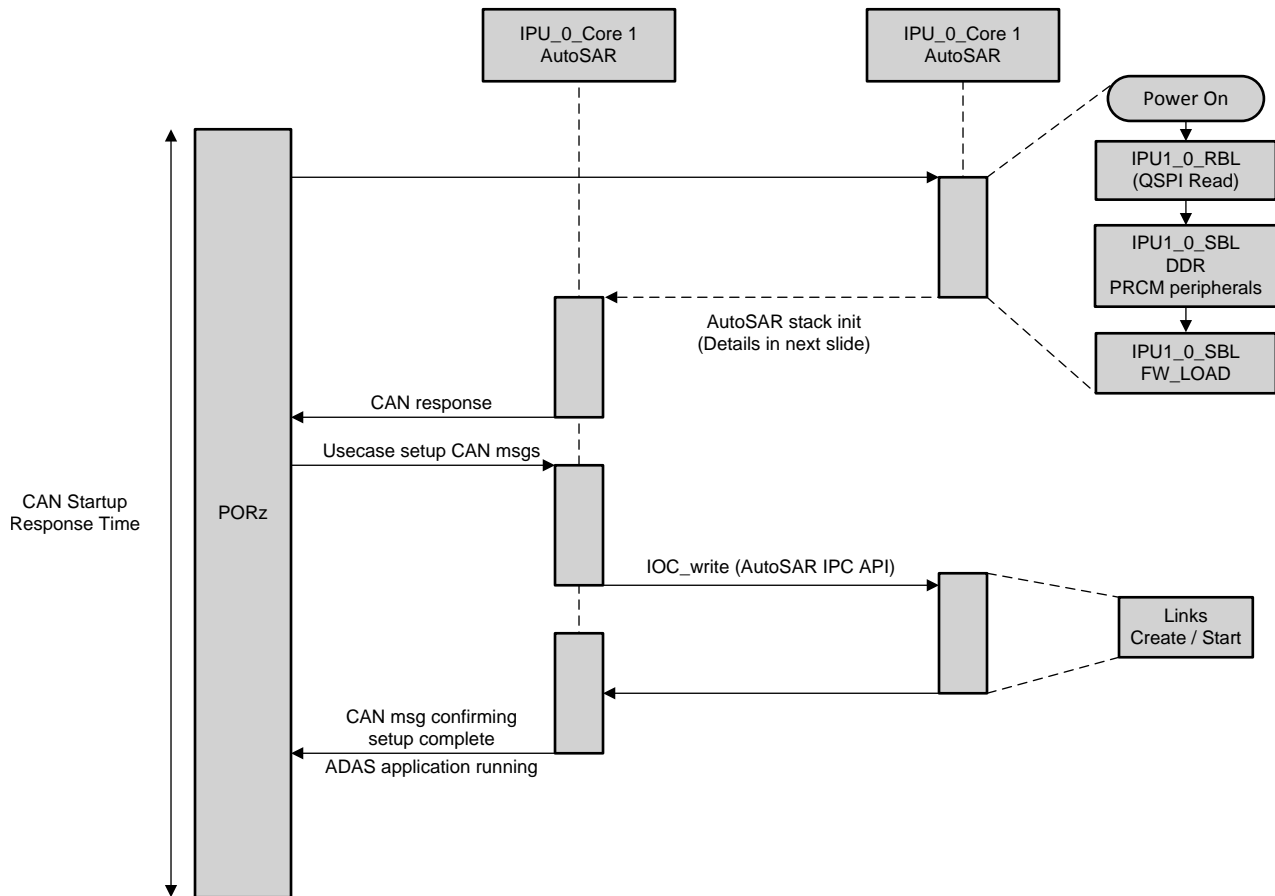


Figure 4-17. Safety-Enabled Boot Sequence Timing

#### 4.14 Vision SDK AutoSAR Stack Initialization With Validation of Resource Conflicts

Figure 4-18 describes the AutoSAR stack initialization on TDA3x.

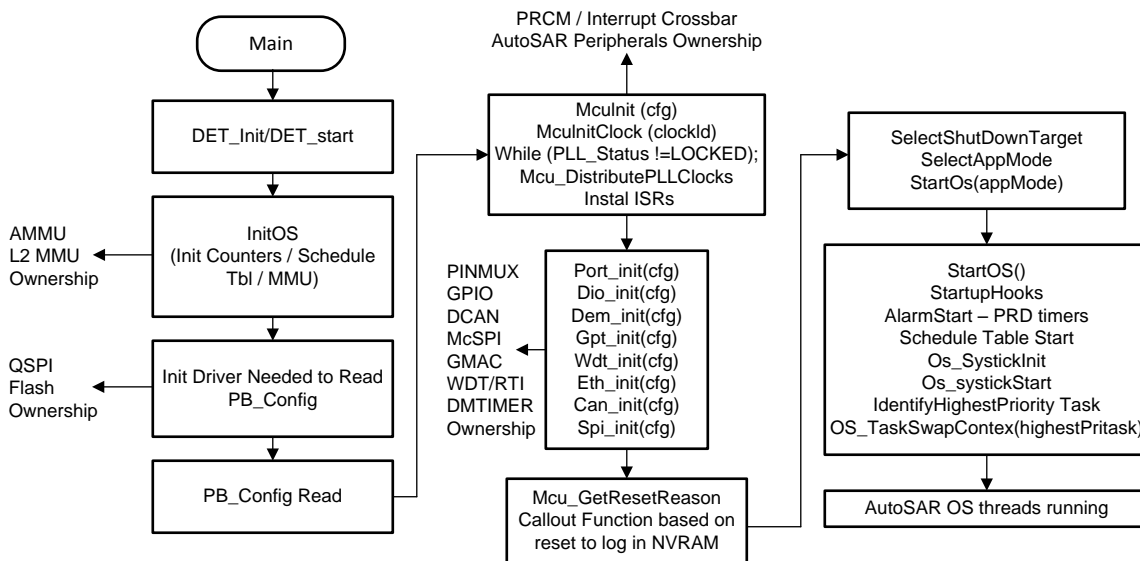


Figure 4-18. Safety-Enabled Software Partitioning and Core Ownership

## 4.15 IPC Concept Between AutoSAR OS and SYS/BIOS

Figure 4-19 describes the IPC communication and isolation concept on TDA3x between AutoSAR and SYS/BIOS.



**Figure 4-19. Safety-Enabled Interprocessor Communication (IPC)**

## 4.16 SYS/BIOS Data Integrity Check Concept

Because SYS/BIOS is an OS that was legacy software, it was not developed according to the development standards of ISO 26262 and IEC 61508. Customers must make an appropriate decision to either continue using SYS/BIOS or choose another RTOS to achieve the safety targets of their system. In any case, TI proposes following concepts to perform run-time data integrity check on the sysbios task objects to detect any data corruption.

The proposal is to generate a checksum of the Task object fields when a Task is created or constructed and use the checksum to perform a data integrity check every time the Task scheduler is invoked.

This integrity check should help catch most cases of Task object corruption due to stack overflows, buffer overflows, incorrect pointer into memory, and so forth.

The following describes the implementation:

- Add a new "checkTaskObjectFlag" module-wide config parameter and a new checkSequence field to the Task object.
- During a Task create or construct, generate a checksum of the Task object fields that do not change during the lifetime of a task and store the result in the checkSequence field of the Task object.  

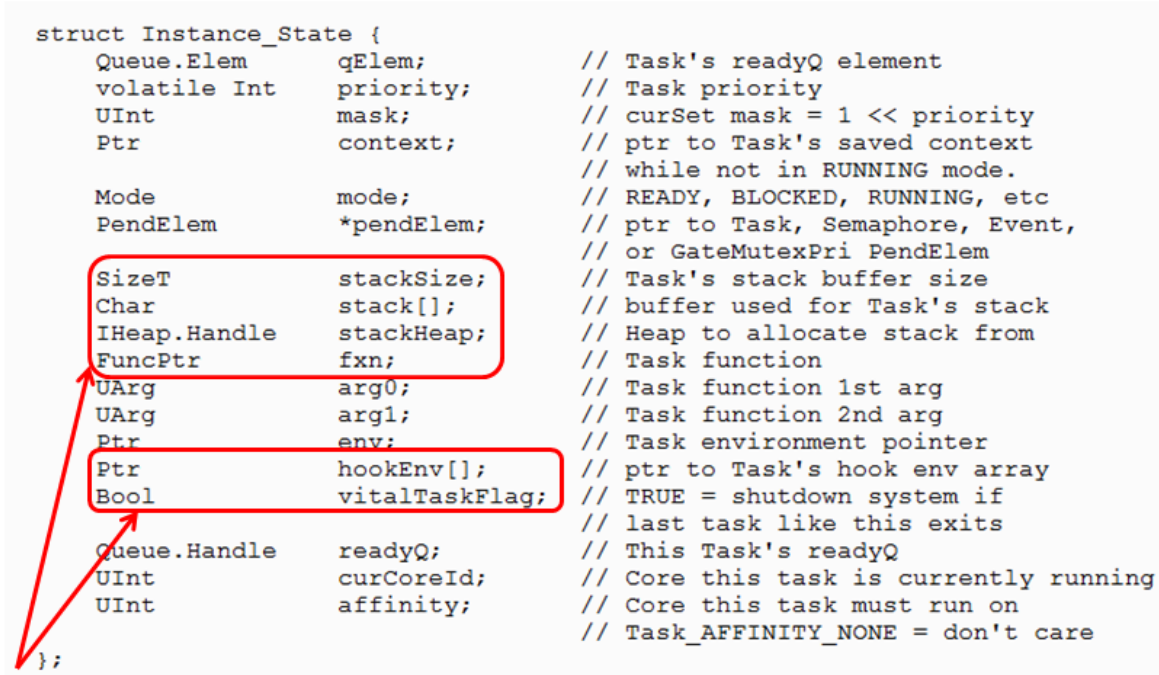
$$\text{tsk} \rightarrow \text{checkSequence} = \sim(\text{unsigned sum of task object fields}) + 1$$
- Add a data integrity check to the Task scheduler. The check will ensure the Task object fields have not been corrupted.

```
if (Task_checkTaskObjectFlag) {
    if ((Unsigned sum of task object fields) + (Check sequence) !=0){
        Error_raise(NULL, Task_E_objectCorrupted, newTask, 0);
    }
}
```

# Task Object Integrity Check

```

struct Instance_State {
    Queue.Elem    qElem;           // Task's readyQ element
    volatile Int  priority;        // Task priority
    UInt          mask;           // curSet mask = 1 << priority
    Ptr          context;         // ptr to Task's saved context
                                // while not in RUNNING mode.
    Mode          mode;           // READY, BLOCKED, RUNNING, etc
    PendElem     *pendElem;       // ptr to Task, Semaphore, Event,
                                // or GateMutexPri PendElem
    SizeT        stackSize;       // Task's stack buffer size
    Char         stack[];         // buffer used for Task's stack
    IHeap.Handle stackHeap;       // Heap to allocate stack from
    FuncPtr      fxn;             // Task function
    UArg         arg0;            // Task function 1st arg
    UArg         arg1;            // Task function 2nd arg
    Ptr          env;             // Task environment pointer
    Ptr          hookEnv[];       // ptr to Task's hook env array
    Bool         vitalTaskFlag;   // TRUE = shutdown system if
                                // last task like this exits
    Queue.Handle readyQ;          // This Task's readyQ
    UInt         curCoreId;       // Core this task is currently running
    UInt         affinity;        // Core this task must run on
                                // Task_AFFINITY_NONE = don't care
};
    
```



Task object fields that do not change during the lifetime of a task.

Figure 4-20. Ensuring Critical Safety Data Integrity in SYSBIOS

## 4.17 Freedom From Interference and Isolation for Software Components

Customers must decide the certification requirements for their safety-sensitive ADAS systems with required ISO 26262 qualification. Because all software modules in a system cannot be ASIL certified, the mixed ASIL components must coexist in a system. It is necessary to ensure freedom from interference (FFI) of the lower ASIL component against the higher ASIL component. The freedom or noninterference must ensure:

- Independence of memory accesses and safety isolation
- Independence of timing behavior and execution order
- Exchange of information among software components with different ASIL levels

### 4.17.1 Memory Isolation High-Level Concept

FFI in the mixed ASIL system is about providing selective memory access restriction to the QM task so that it does not contaminate ASIL memory. In engineering terms, one possible set of methods are as follows:

- QM tasks and ASIL tasks must coexist on multiple CPUs sharing a common memory.
- ASIL tasks therefore can be provided R/W access for all memory.
- QM tasks need read access to ASIL regions, because they may share data with ASIL tasks.
- QM tasks must not be given write permission to ASIL regions.
- All tasks will have R/W access to QM regions.



Table 4-1 describes the hardware modules in TDA3x that support memory isolation implementation as well as the cache inheritance features.

**Table 4-1. Implementing FFI for Processor Cores**

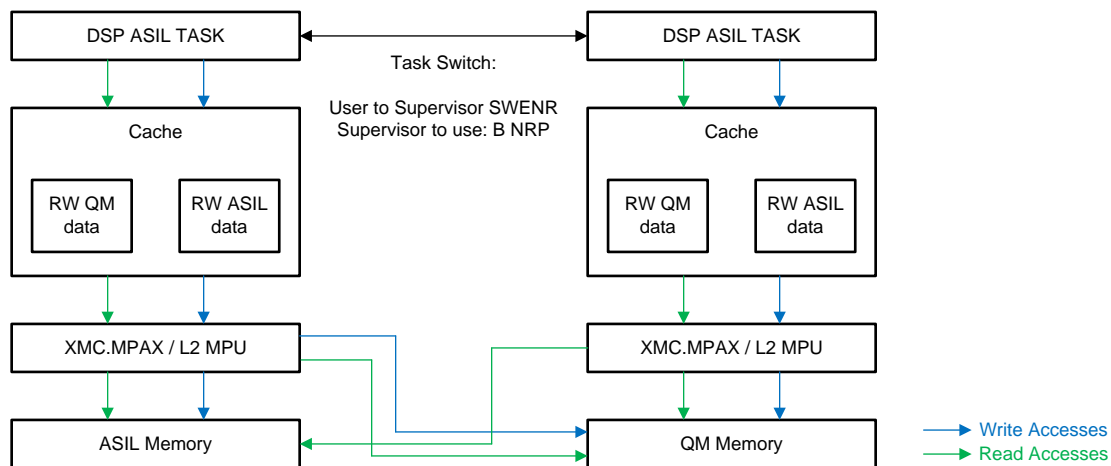
Modules	Advantages	Considerations
EVE, DSP, ISU, and SYS MMU	Allow or disallow access to a given region by mapping and unmapping a region.	Does not have user/supervisor or read/write based restriction. If there is a read permission, write permission is also present.
EMIF and OCMC firewall	<ul style="list-style-type: none"> <li>Memory protection is based on master ID and CPU mode (supervisor versus user).</li> <li>Memory like EMIF and OCMC can be divided into multiple regions, each having different access restrictions.</li> </ul>	EVE does not have supervisor mode.
DSP MPU/XMC	<ul style="list-style-type: none"> <li>Allows access restriction on internal and external memories of the DSP.</li> <li>Memories can be divided into multiple regions, each having different permission with regard to the user and supervisor.</li> </ul>	
DSP cache	<ul style="list-style-type: none"> <li>DSP cache inherits the permission of the DSP MPU when the cache line from a given MPU region is allocated.</li> <li>Because the MPU has permission control, like user and supervisor, cache also has the same permission control.</li> </ul>	<ul style="list-style-type: none"> <li>Firewall permissions are not inherited in the cache line. Therefore, if changing the firewall permission, TI recommends flushing the cache.</li> <li>MMU of the DSP is after the cache, therefore if the MMU permission is changed dynamically, TI recommends a cache flush.</li> </ul>
IPU cache	M4 UNICACHE inherits the permission of the AMMU when the cache gets allocated.	<ul style="list-style-type: none"> <li>Core-wise distinction of the permission is not possible, because the cache/A,,U/L2MMU of IPU are common to both.</li> <li>Both AMMU and L2MMU of the IPU are after the cache; therefore, if the MMU permission is changed dynamically, TI recommends flushing the cache.</li> </ul>
EVE cache	EVE cache is only for programming; therefore, no protection is explicitly needed.	
EDMA (EVE, DSP, and SYS)	EDMA transfers inherit the CPU privilege of the task which programs the PaRam space. This EDMA.PaRam.OPT.PRI V. mode can be used by the firewall to differentiate access permission.	

### 4.17.2 Memory Isolation on DSP

, Figure 4-21, and Figure 4-22 show one possible approach to implement the memory isolation feature among separate tasks running on the DSP.

Example solution – DSP

- Defining memory sections in DDR and L2RAM with varying permission
  - L2RAM section permissions are defined using the memory protection unit (MPU) inside the DSP subsystem.
    - Protection is based on privilege level and has control for read/write/execute
    - Protection is also based on Access ID
  - OCMC and DDR access permission is defined using memory protection with the extended memory controller.
    - The MPAX unit supports 16 user-defined address ranges (MPAX segments) to apply memory protection and address extension.
    - The MPAXL.PERM register can be used to program the permission of these 16 sections: user versus supervisor and read/write/execute control
    - The L2 and L1 cache line inherits the permission as defined in MPAXL.PERM; all access control occurs in cache using this inherited permission, thus simplifying flow control.
- Switch off supervisor and user mode in sysBIOS:
  - SysBIOS is always in supervisor mode; native tasks are all in supervisor mode
  - Add custom APIs in QM task to momentarily switch mode to user
    - Use B NRP to enter user mode
    - Use SWENR to enter supervisor mode
  - No shadow registers for stack pointer in C66x for interrupts and exceptions
- Allows interrupt to have ASIL permission, even if it is called in QM task
  - This is achieved as interrupts and exceptions in the C66x always receive supervisor privilege.
  - B IRP instruction: returns to saved mode from interrupt service rout, and thus reinstates the previous permission



**Figure 4-21. Ensuring Isolation Between an ASIL and QM Task on DSP (1 of 2)**

## DSP: Proof of Concept

```

/****DSP Code****/
main()
{
    TaskCreate(Task1, (priority = 0)); /* QM */
    TaskCreate(Task2, (priority = 1)); /* ASIL */
    TaskCreate(Task3, (priority = 2)); /* QM */
    Semaphore_Create(mySem); /* to enable task switch */
    setL2MemProtPerms(); /* No change at run-time */
    setXHCPerms(); /* No change at run-time */
    setL3FwNode(); /* No change at run-time - reqd due to EDMA */
    iteration = 0;
    BIOS_start();
}

Task1() /* QM */
{
    while(iteration < MAX_ITER_COUNT)
    {
        switchCpuMode(USR);

        < Show QM permissions to DDR from CPU with cache >
        < Show QM permissions to DDR from EDMA >
        < Show QM permissions to L2 RAM using CPU >

        switchCpuMode(PRV);
        Semaphore_pend(mySem);
        < Task3 will be scheduled, completed and control returns here >
        switchCpuMode(USR);

        < Show QM permissions to DDR from CPU with cache >
        < Show QM permissions to DDR from EDMA >
        < Show QM permissions to L2 RAM using CPU >

        switchCpuMode(PRV);
        Task_disable(); /* Disable scheduler */
        Task_setPri(Task2, priority = 1);
        Task_restore(); /* Restart scheduler */
        < Task2 will be scheduled, completed and control returns here >
    }
}

/****DSP Code****/
Task3() /* QM */
{
    while(iteration < MAX_ITER_COUNT)
    {
        switchCpuMode(USR);

        < Show QM permissions to DDR from CPU with cache >
        < Show QM permissions to DDR from EDMA >
        < Show QM permissions to L2 RAM using CPU >

        switchCpuMode(PRV);
        Semaphore_post(mySem);
        < Task1 resumes, control returns here in next iteration >
    }
}

Task2() /* ASIL */
{
    while(iteration < MAX_ITER_COUNT)
    {
        < Show ASIL permissions to DDR from CPU with cache >
        < Show ASIL permissions to DDR from EDMA >
        < Show ASIL permissions to L2 RAM using CPU >

        iteration++;

        Task_disable(); /* Disable scheduler */
        Task_setPri(Task2, priority = 1);
        Task_restore(); /* Restart scheduler */
        < Task1 resumes, control returns here in next iteration >
    }
}

```

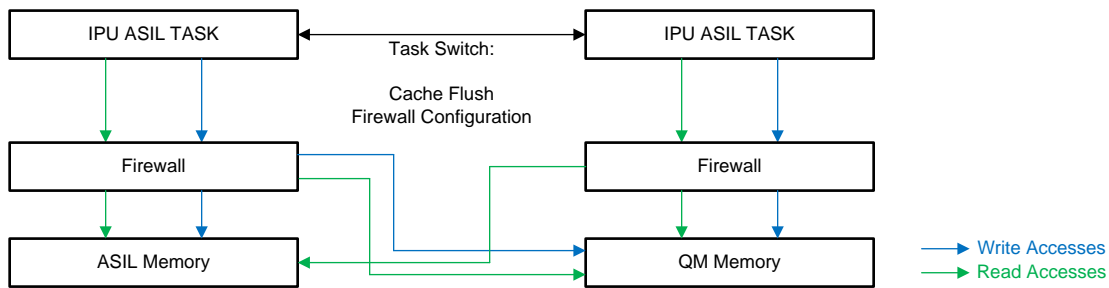
Figure 4-22. Ensuring Isolation Between an ASIL and QM Task on DSP (2 of 2)

### 4.17.3 Memory Isolation on Cortex-M4

Figure 4-23 and show one possible implementation of memory isolation for multiple tasks running on the Cortex-M4.

Example solution – IPU

- FFI solution:
  - Define memory regions in external memory using L3 firewalls
  - Firewalls are reconfigured to QM mode in the following cases:
    - QM tasks entry
  - Firewalls are reconfigured to ASIL mode in the following cases:
    - QM tasks exit
  - Perform cache flush at entry of the QM task to remove cache entries from previous permission
  - Disable interrupt in QM tasks
  - Require task synchronization between both the coes; ASIL task is in one core, and the other core cannot run QM task



**Figure 4-23. Ensuring Isolation Between an ASIL and QM Task on M4**

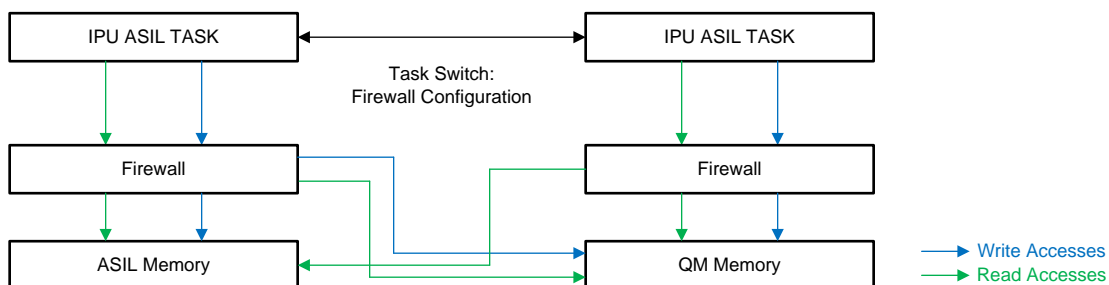
- Considerations
  - IPU UNICACHE MMU
    - UNICACHE MMU provides read, write, and execute attributes to memory regions.
    - This attribute is shared by both cores.
  - IPU L2 MMU
    - L2 MMU can be used for further translation of L3 address space
    - This attribute is shared by both cores, and the access comes from IPU UNICACHE.

#### 4.17.4 Memory Isolation on EVE

and Figure 4-24 show one possible implementation of memory isolation for tasks running on EVE.

Example solution – EVE

- FFI solution:
  - Define memory regions in external memory using L3 firewalls
  - Firewalls are reconfigured to QM mode in the following cases:
    - QM tasks entry
  - Firewalls are reconfigured to ASIL mode in the following cases:
    - QM tasks exit
  - Save and restore the current mode (ASIL or QM) upon entry and exit of the interrupt handler.



**Figure 4-24. Ensuring Isolation Between an ASIL and QM Task on EVE**

- Considerations
  - EVE DMEM
    - EVE internal DMEM is not protected, and thus should not be used for scratch. No persistent data related to ASIL tasks should be stored.
    - If stack or data of the ASIL tasks must be preserved across QM tasks, it should be stored in DDR or OCMC memory.
  - ARP32
    - ARP32 does not have concept of user and supervisor mode, or any privilege mode of execution, unlike DSP and A15.

- Because EVE is used for SIMD acceleration along with DSP or IPU, the system can be realized using a standalone (no OS) framework, which consists of a simple command queue executing EVE kernels.

## ***Dependent Fail Analysis Information***

---

---

The following are possibilities of dependent failures in the system.

- Any failure in the safety diagnostic logic, such as ECC circuits. This check can be done using a simple fault injection diagnostic sequence for ECC circuitry, where:
  1. Data is written in a memory location with ECC enabled
  2. ECC is disabled and a different data is written
  3. ECC circuitry is enabled further
  4. The corrupted data is readThis process should lead to detection of an ECC circuitry failure.
- Any failure in the ADC VREF may lead to missed or incorrect detection of a safety event on a parameter being monitored. Therefore, the VREF voltage for the ADC must be monitored by an external mechanism.
- Similarly any failure on a CPU that is supposed to respond to a safety-critical interrupt or an IP that is generating that interrupt (such as DCC, RTI, timers, or ADC) can be covered through the handling of the interrupt by different CPUs, or at the minimum by using more than one timer to watch over the time-out errors of the safety monitoring process that runs on the CPU.
- In general, TI recommends that customers perform an independent analysis of their system for identification of dependent and multipoint failures and design mechanisms for monitoring the safety mechanisms.

## Development Interface Agreement

A development interface agreement (DIA) is intended to capture an agreement between a customer and supplier regarding the management of shared responsibilities when developing a functional safety system. In custom developments, the DIA is a key document executed between the customer and supplier early in the development process. Because the TDA3x device family is a commercial, off-the-shelf (COTS) product, TI is open to engaging with customers for DIAs for customer developments. Requests for custom DIAs must be referred to your local TI sales office for disposition.

### A.1 Appointment of Safety Managers

TI has appointed a safety manager for the development of the TDA3x device family.

### A.2 Tailoring the Safety Lifecycle

The development of the TDA3x device family adheres to the requirements outlined by ISO 26262:2011. The standard DSP development flow from TI, along with the safety-compliant development process, has been followed. This fact does not eliminate the customer's responsibility to perform their own due diligence for implementing a functional safety lifecycle.

### A.3 Activities Performed by TI

The TI DSP products covered by this DIA are hardware components, developed with any safety standard in mind (see [Table A-1](#)). System-level architecture, design, and safety analysis are not in the scope of TI activities, and are the responsibility of the customer.

**Table A-1. Activities Performed by TI and Performed by Customer**

Safety Lifecycle Activity	TI Execution	SEoC Customer Execution
Management of functional safety	N/A	Yes
Definition of end equipment and item	N/A	Yes
Hazard and risk analysis of end equipment and item	N/A	Yes
Development of end equipment safety concept	Assumptions made	Yes
Allocation of end equipment requirements to subsystems, hardware components, and software components	Assumption made	Yes
Definition of DSP safety requirements	N/A	No
DSP architecture and design execution	Yes	No
DSP-level safety analysis	N/A	No
DSP-level verification and validation	Yes	No
Integration of DSP into end equipment	Support provided	Yes
End equipment-level safety analysis	No	Yes
End equipment-level verification and validation	No	Yes
End equipment-level safety assessment	Support provided	Yes
End equipment release to production	No	Yes
Management of safety issues in production	Support provided	Yes

## A.4 Information to Exchange

In a custom development, there is an expectation under ISO 26262 that all development documents related to work products are made available to the customer. In a COTS product, this approach is unsustainable. TI has summarized the most critical development items into a series of documents that can be made available to customers either publicly or under a nondisclosure agreement (NDA). NDAs are required to protect proprietary and sensitive information disclosed in certain safety documents.

[Table A-2](#) summarizes the product safety documentation that TI can provide to customers to assist in development of safety systems.

**Table A-2. Product Safety Documentation**

Deliverable Name	Contents	Confidentiality	Availability
Safety product review	Overview of safety considerations in product development and product architecture. Delivered ahead of public product announcement.	NDA required	Not circulated because the product is already released to market and the safety manual is available
Safety manual	User guide for the safety features of the product, including system-level assumptions of use	NDA required	Available
Safety analysis report summary for the TDA3x device family	Summary of FIT rates and device safety metrics according to ISO 26262 at device level	NDA required	Part of the FMEDA
Safety case report	Summary of the conformance of the product to the ISO 26262 standard	NDA required	Audited by an external certification agency - Exida
Safety case database	Clause-by-clause detailing of compliance to ISO 26262 standards	NDA required	Audited by an external certification agency - Exida

## A.5 Parties Responsible for Safety Activities

TI has developed the product according to the ISO 26262 standard requirements to the best of our ability. This implies no guarantees that the part can automatically enable ISO 26262 compliant systems. Customers are fully responsible for the performance of their products, as well as robustness and safety lifecycle. TI will support customers in achieving their system targets with information and data about the device as needed and when possible.

## A.6 Supporting Processes and Tools

TI uses a variety of tools and corresponding data formats for internal and external documents. [Table A-3](#) lists the tools and data formats that are relevant to the safety-related documents shared with SEOc customers.

Apart from these tools and data formats, any updates to the safety architecture or reports shall be communicated through the CDDS system to all customer stakeholders. The customer project managers are advised to ensure that the correct stakeholders have the CDDS portal access that can be enabled through your concerned customer program manage (CPM).

**Table A-3. Product Safety Documentation Tools and Data Formats**

Deliverable Name	Creation Tools	Output Formats
Safety product review	N/A	N/A
Safety manual	XML	Adobe PDF
Safety analysis report summary for the TDA3x device family	XML, Microsoft® Excel	Adobe PDF, Microsoft Excel 2003
Safety case report	N/A	N/A
Safety case database	N/A	N/A



## **A.7 Supplier Hazard and Risk Assessment**

Hazard and risk assessments under ISO 26262 are targeted at the system level of abstraction. When developing a hardware component out of context, the system implementation is not known. Therefore, TI has not executed a system hazard and risk analysis. Instead, TI has made assumptions that are fed into the component design. The system integrator is ultimately responsible for determining if the TI component is suitable for use in the system.

## **A.8 Creation of Functional Safety Concept**

The functional safety concept under ISO 26262 is targeted at the system level of abstraction. When developing a hardware component out of context, the system implementation is not known. Therefore, TI cannot generate a system functional safety concept. Instead, TI has made assumptions that have been fed into the component design. The system integrator is ultimately responsible for determining if the TI component is suitable for use in the system.

## IMPORTANT NOTICE AND DISCLAIMER

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

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2022, Texas Instruments Incorporated