Functional Safety Manual for AWR2188 v 0.5

Functional Safety Information

DRAFT ONLY
TI Confidential – NDA Restrictions

Literature Number: SFFSAQ4
AUGUST 2025 – REVISED NOVEMBER 2025

DRAFT ONLY
TI Confidential – NDA Restrictions

Table of Contents

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

List of Figures

List of Tables

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Chapter 1
Introduction

This document is a *Functional Safety Manual* for the Texas Instruments' AWR2188 component. The specific orderable part numbers supported by this functional safety manual are as follows:

- AWR2188 (Orderable Part Number: OrderablePartNumber1)
- DeviceName2 (Orderable Part Number: OrderablePartNumber2)
- DeviceName3 (Orderable Part Number: OrderablePartNumber3)

This functional safety manual provides information needed by system developers to help in the creation of a functional safety system using a AWR2188 component. This document includes:

- An overview of the component architecture
- An overview of the development process used to decrease the probability of systematic failures
- An overview of the functional safety architecture for management of random failures
- The details of architecture partitions and implemented functional safety mechanisms

The following information is documented in the *Functional Safety Analysis Report* and is not repeated in this document:

- Summary of failure rates (FIT) of the component
- Summary of functional safety metrics of the hardware component for targeted standards (for example IEC 61508, ISO 26262, and so forth)
- Quantitative functional safety analysis (also known as FMEDA, Failure Modes, Effects, and Diagnostics Analysis) with detail of the different parts of the component, allowing for customized application of functional safety mechanisms
- Assumptions used in the calculation of functional safety metrics

The following information is documented in the *Functional Safety Report* and is not repeated in this document:

- Results of assessments of compliance to targeted standards

The user of this document needs a general familiarity with the AWR2188 component. For more information, refer to the AWR2188 data sheet. This document is intended to be used in conjunction with the pertinent data sheets, technical reference manuals, and other component documentation.

For information that is beyond the scope of the listed deliverables, contact your TI sales representative or go to TI Functional Safety.

ADVANCE INFORMATION for preproduction products; subject to change without notice.

Trademarks

TI E2E™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Chapter 2

AWR2188 Hardware Component Functional Safety Capability

This section summarizes the component functional safety capability.

This hardware component:

- Was developed as a functional Safety Element out of Context (SEooC)
- Was developed according to the relevant requirements of IEC 61508:2010
- Was developed according to the relevant requirements of ISO 26262:2011
- Was developed according to the relevant requirements of ISO 26262:2018
- Achieves systematic integrity of SIL 1234
- Achieves systematic integrity of ASIL ABCD
- Includes sufficient functional safety mechanisms for random fault integrity requirements of SIL 1234
- Includes sufficient functional safety mechanisms for random fault integrity requirements of ASIL ABCD
- Has passed a functional safety assessment by a certified third party
- Has passed a functional safety assessment by Texas Instruments

Note

This component is still undergoing development and the functional safety assessment is not yet complete.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Chapter 3

Development Process for Management of Systematic Faults

For functional safety development, it is necessary to manage both systematic and random faults. Texas Instruments follows a new-product development process for all of its components which helps to decrease the probability of systematic failures. This new-product development process is described in Section 3.1. Components being designed for functional safety applications will additionally follow the requirements of TI's functional safety development process, which is described in Section 3.2.

3.1 TI New-Product Development Process

Texas Instruments has been developing components for automotive and industrial markets since 1996. Automotive markets have strong requirements regarding quality management and product reliability. The TI new-product development process features many elements necessary to manage systematic faults. Additionally, the documentation and reports for these components can be used to assist with compliance to a wide range of standards for customer's end applications including automotive and industrial systems (e.g., ISO 26262-4, IEC 61508-2).

This component was developed using TI's new product development process which has been certified as compliant to ISO 9001 / IATF 16949 as assessed by Bureau Veritas (BV).

The standard development process breaks development into phases:

- Assess
- Plan
- Create
- Validate

Figure 3-1 shows the standard process.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

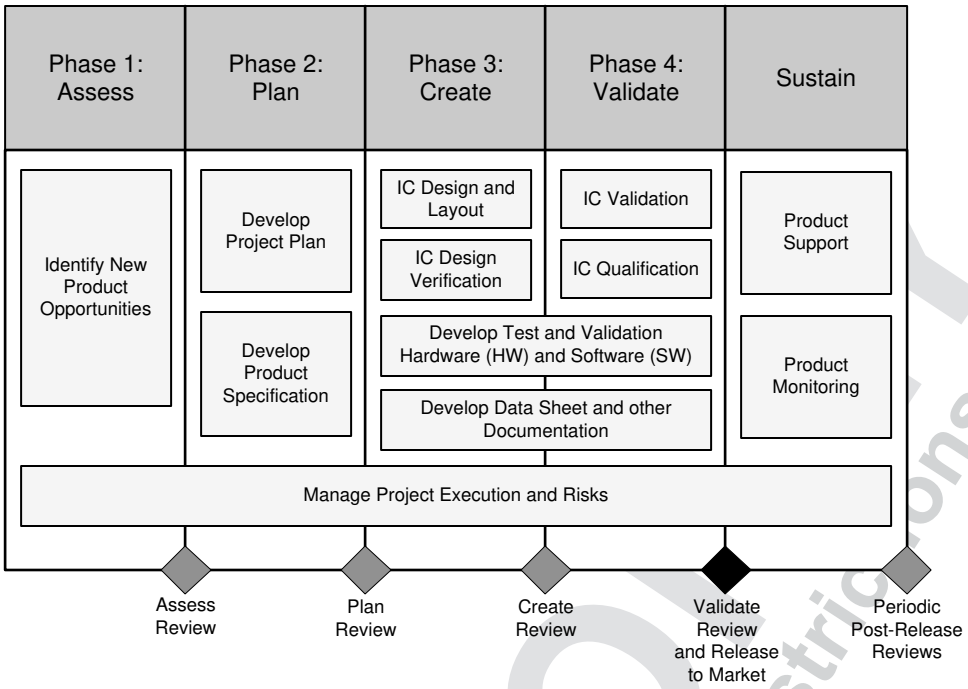

Figure 3-1. TI New-Product Development Process

3.2 TI Functional Safety Development Process

The TI functional safety development flow derives from ISO 26262 and IEC 61508 a set of requirements and methodologies to be applied to semiconductor development. This flow is combined with TI's standard new product development process to develop TI functional safety components. The details of this functional safety development flow are described in the TI internal specification - Functional Safety Hardware.

Key elements of the TI functional safety-development flow are as follows:

- Assumptions on system level design, functional safety concept, and requirements based on TI's experience with components in functional safety applications
- Qualitative and quantitative functional safety analysis techniques including analysis of silicon failure modes and application of functional safety mechanisms
- Base FIT rate estimation based on multiple industry standards and TI manufacturing data
- Documentation of functional safety work products during the component development
- Integration of lessons learned through multiple functional safety component developments, functional safety standard working groups, and the expertise of TI customers

Table 3-1 lists these functional safety development activities which are overlaid atop the standard development flow in Figure 3-1.

Refer to Appendix B for more information about which functional safety lifecycle activities TI performs.

The customer facing work products derived from this TI functional safety process are applicable to many other functional safety standards beyond ISO 26262 and IEC 61508.

Table 3-1. Functional Safety Activities Overlaid on top of TI's Standard Development Process

Assess	Plan	Create	Validate	Sustain and End-of-Life
Determine if functional safety process execution is required	Define component target SIL/ASIL capability	Develop component level functional safety requirements	Validate functional safety design in silicon	Document any reported issues (as needed)
Nominate a functional safety manager	Generate functional safety plan	Include functional safety requirements in design specification	Characterize the functional safety design	Perform incident reporting of sustaining operations (as needed)

Copyright © 2025 Texas Instruments Incorporated

Table 3-1. Functional Safety Activities Overlaid on top of TI's Standard Development Process (continued)

Assess	Plan	Create	Validate	Sustain and End-of-Life
End of Phase Audit	Verify the functional safety plan	Verify the design specification	Qualify the functional safety design (per AEC-Q100)	Update work products (as needed)
	Initiate functional safety case	Start functional safety design	Finalize functional safety case	
	Analyze target applications to generate system level functional safety assumptions	Perform qualitative analysis of design (i.e. failure mode analysis)	Perform assessment of project	
	End of Phase Audit	Verify the qualitative analysis	Release functional safety manual	
		Verify the functional safety design	Release functional safety analysis report	
		Perform quantitative analysis of design (i.e. FMEDA)	Release functional safety report	
		Verify the quantitative analysis	End of Phase Audit	
		Iterate functional safety design as necessary		
		End of Phase Audit		

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Chapter 4
AWR2188 Component Overview

The AWR2188 device is an integrated single-chip FMCW transceiver capable of operation in the 76GHz to 81GHz band. The device enables unprecedented levels of integration in an extremely small form factor. AWR2188 is designed for low power, self-monitored, ultra-accurate radar systems in the automotive space.

The AWR2188 device is a self-contained FMCW transceiver single-chip device that simplifies the implementation of Automotive Radar sensors in the band of 76GHz to 81GHz. The device is built on TI's low-power 45nm RFCMOS process, which enables a monolithic implementation of a 8TX, 8RX system with built-in PLL and ADC converters. Simple programming model changes can enable a wide variety of sensor implementation (Short, Mid, Long) with the possibility of dynamic reconfiguration for implementing a multimode sensor. Additionally, the device is provided as a complete platform device including reference hardware design, software drivers, sample configurations, API guide, and user documentation.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

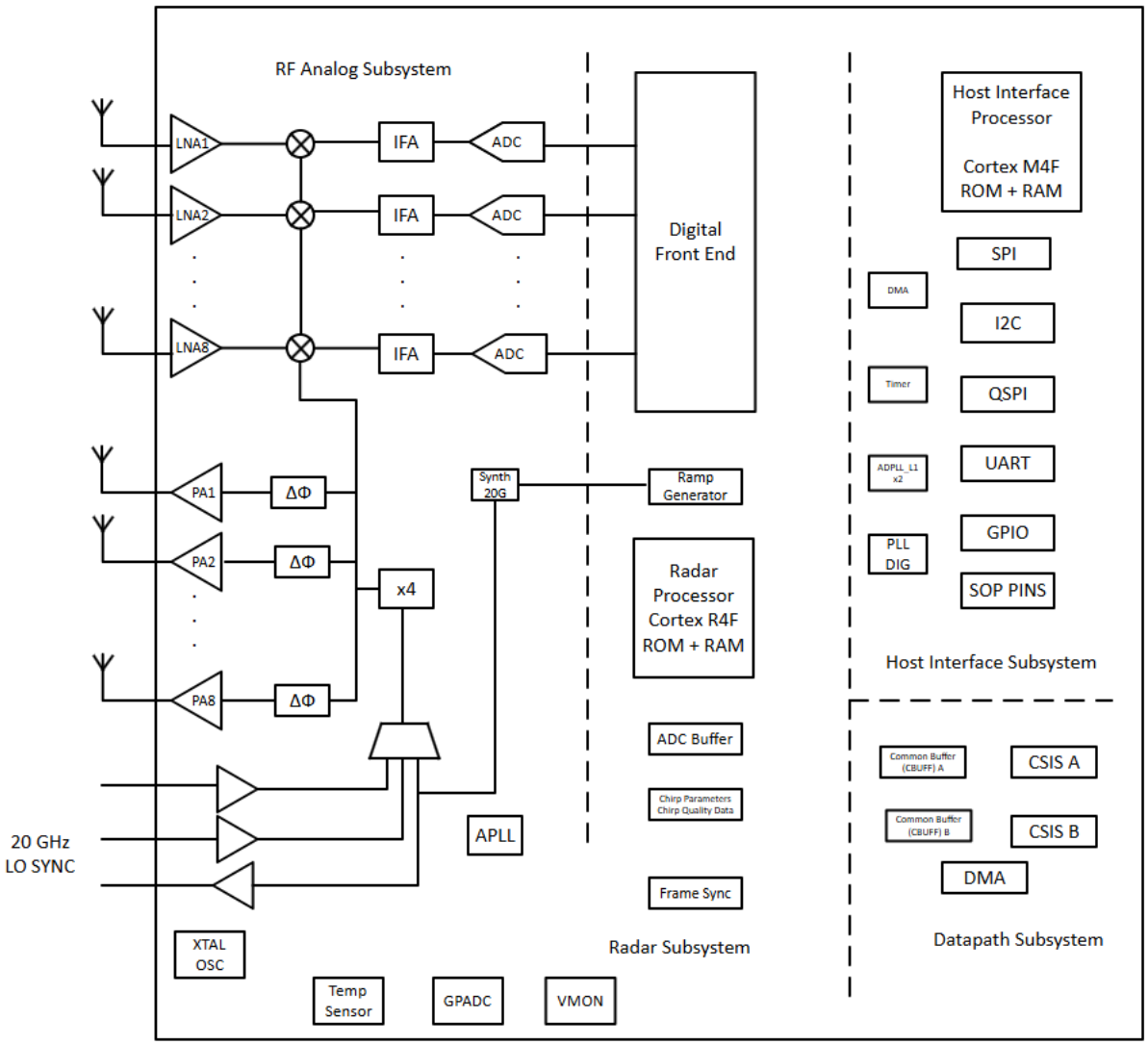

Figure 4-1. AWR2188 Block Diagram

4.1 Targeted Applications

The AWR2188 component is targeted at general-purpose functional safety applications. This is called Safety Element out of Context (SEooC) development according to ISO 26262-10. In this case, the development is done based on assumptions on the conditions of the semiconductor component usage, and then the assumptions are verified at the system level. This method is also used to meet the related requirements of IEC 61508 at the semiconductor level. This section describes some of the target applications for this component, the component safety concept, and then describes the assumptions about the systems (also know as Assumptions of Use or AoU) that were made in performing the safety analysis.

Example target applications include, but are not limited to, the following:

- Automated Highway Driving
- Automatic Emergency Braking
- Adaptive Cruise Control
- Imaging Radar using cascading configuration

Copyright © 2025 Texas Instruments Incorporated

Figure 4-2 shows a generic block diagram for a system. This diagram is only an example and does not necessarily represent a complete system.

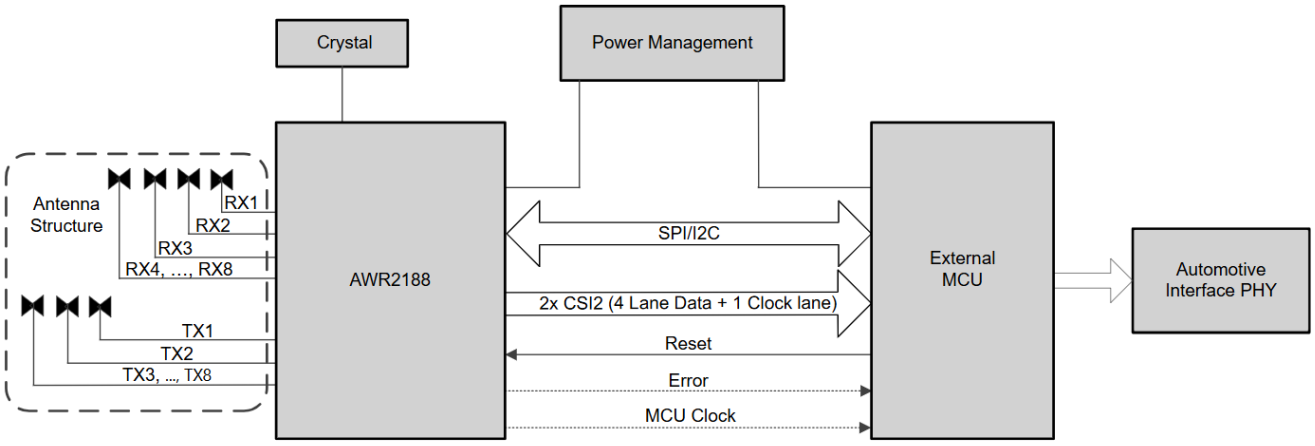

Figure 4-2. AWR2188 Typical Application

4.2 Hardware Component Functional Safety Concept

The AWR2188 device follows a safety philosophy concept termed as “Hierarchical Safety Monitoring by Host”. The basic concept involves a balance between application of hardware diagnostics and software diagnostics to manage functional safety, while balancing cost concerns. In the “Safety Monitoring by Host” approach, the overall safety monitoring responsibilities of the total Radar sensor block (that includes the front end Radar sensors (AWR2188) and the interfacing Host ECU) is predominantly residing in the Host ECU. The front end Radar sensors (AWR2188) also have built-in safety mechanisms to assist the overall safety monitoring by the host by offloading some of the safety monitoring from the host and relying on the host only to monitor in a hierarchical manner. Within the AWR2188 device, a core set of elements are allocated continuously/periodically operating hardware safety mechanisms. This core set of elements include Power Supply, Clock and Reset, M4, R4F Processor and other data processing and Timing engine units, Program and Data memory, Vectored Interrupt Module and associated interconnect to assure functionally correct execution of software. Once correct operation of these elements is confirmed, software can be executed on these elements in order to provide software-based diagnostics on other device elements, such as peripherals. Hierarchically, the interfacing Host ECU finally performs the continuous and/or periodic monitoring of the front-end Radar sensor (AWR2188) be performing temporal and logical monitoring of the control and data paths that is the interface to the front-end Radar sensor. It can also monitor the output signals from the sensor to determine if there are any safety errors (for ex. nError indication). This concept has been proven viable through multiple generations of safety-critical mmWave Radar Family products in the automotive passenger vehicle space.

4.3 Hardware Component Configuration

Ensuring freedom from interference of different subsystems and modules is of primary importance and the key enablers that would assist ensuring the same are tabulated below. In addition to the built in hardware mechanisms that have been provisioned, there are certain SW configuration sequences/strategies that would further assist the FFI concept as allocated below.

The enabler functions to realize Freedom From Interference are listed below:

- **[DFA_FFI_1] - Separate interconnect per subsystem** - A separate interconnect is provisioned per subsystem such that errors in one subsystem does not affect the operation in the other subsystems. It is recommended to avoid the use of the processor core to initiate transactions across sub-systems.EDMAs may be deployed. Alternatively, the mailbox based handshake may be performed.

Copyright © 2025 Texas Instruments Incorporated

- **[DFA_FFI_2] - Heirarchical interconnect structure within the subsystem** - The heirarchical structure of the interconnect is implemented and ensures the fault in one module does not impact the connectivity to the other module within the subsysem.
- **[DFA_FFI_3] - L2 interconnect (PCR) with peripheral access-permission-gating controls** - The L2 interconnect (PCR) supports a Transaction ID based access permissions (read/write, user/supervisor, debug) configurations. It also allows the clock gating of the unused peripherals.
- **[DFA_FFI_4] - L2 interconnect (PCR) with timeout mechanism** - The L2 interconnect (PCR) supports a timeout mechanism if a particular module encounters a fault and is unable to recover from the fault and respond to a transaction request.
- **[DFA_FFI_5] - Processor controlled MPUs** - The software deploys the MPUs supported by the processor cores (M4, R4) to isolate the safety and non-safety (QM) code execution flow.
- **[DFA_FFI_7] - Mailbox based inter-subsysem/processor communication** - The software deploys a Mailbox based inter-processor communication to minimize/limit the interactions between the subsystem to the access to Mailbox alone.
- **[DFA_FFI_8] - Subsystem level infrastructural components** - The infrastructural components/safety enablers (WDT, RTI, Mailbox, CRC, ESM, DCC) are provisioned per subsystem. The software deploys the mechanisms to localize the faults and take appropriate actions.
- **[DFA_FFI_9] - Finer Clock gating controls of unused peripherals** - The hardware supports finer clock gating controls to keep unused peripherals gated based on the deployment usecase scenarios.
- **[COEX_FFI_1]** - Before start of actual Application, all Non-Safety Modules/peripherals must be taken to Reset, disabled in PCR or other isolation methods as per the Spec to not interfere with other Modules.
- **[COEX_FFI_2]** - Before start of actual Application, all Non-Safety Modules(QM)/peripherals Interrupts must be disabled in VIM Modules.
- **[COEX_FFI_3]** - The RS232 interfaces should not be used during safety critical operation - In functional mode
- **[COEX_FFI_4]** - Hardware disable of RS232 interface is implemented.
- **[COEX_FFI_5]** - Before start of actual Application, all Non-Safety Modules/peripherals DMA requests must be disabled in respective EDMA Modules.
- **[COEX_FFI_6]** - The JTAG interfaces should not be used during safety critical operation - In functional mode
- **[COEX_FFI_9]** - The BSS logger interfaces (including BSS SCI, UDMA) should not be used during safety critical operation - In functional mode. It is possible to enable-disable the BSS logger using an appropriate handshake.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

4.4 Operating States

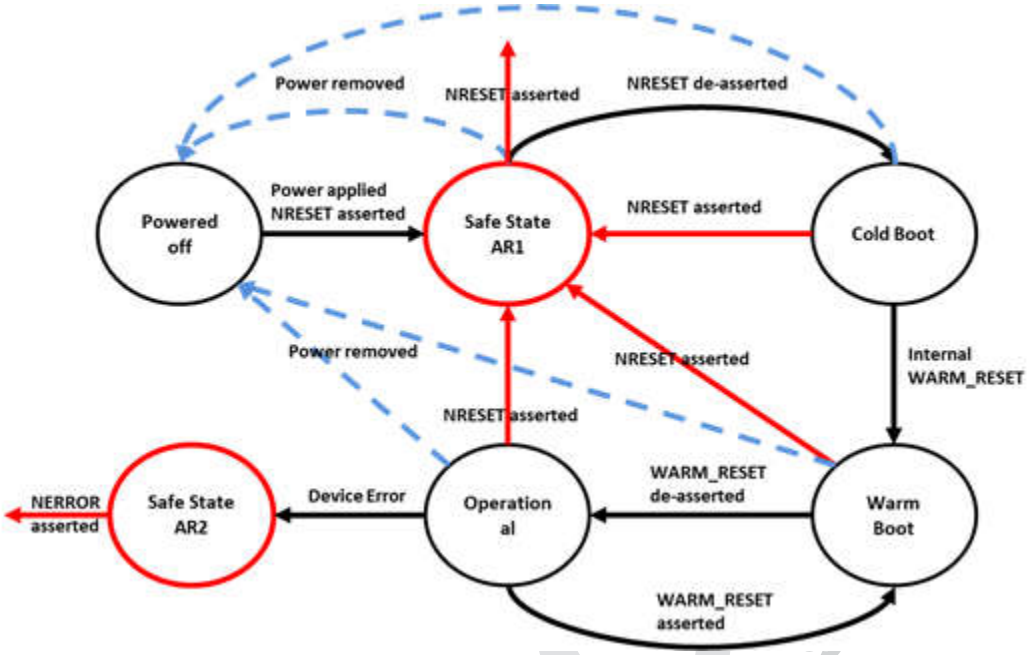

Figure 4-3. Operating States of the Device

Powered Off State

This is the initial operating state of the AWR2188 device. No power is applied to either core or I/O power supply and the device is non-functional. This state can only transition to the safe state AR1, and can only be reached from the safe state.

Safe State

In the safe state, the AWR2188 is powered but non-operational. The NRESET (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. The internal voltage monitor (VMON) safety mechanism also continues to assert the NRESET internal to the device if power supplies are not within a minimum operational range. When the product is in the safe state AR1, Output drivers are tri-stated and IO pins are kept in an input mode with default PU/PD values with few exceptions on SOP pins (PMIC_CLKOUT, SYNC_OUT and TDO), WARM_RESET (open drain) and “NERRER_OUT” pin (open drain).

Cold Boot State

In the cold boot state, key components such as oscillator, digital control logic- Wakeup state machine, and debug logic are functioning. When the cold boot process is completed, the WARM_RESET signal (internal “warm_reset” signal) is internally released, leading to the warm boot stage.

Warm Boot

The warm boot mode resets complete digital logic. The CPU (CM4) begins executing software from Program memory and software initialization of the device can begin. Warm boot completion has to be indicated by application software. Warm boot state can also be entered into by SW based reset trigger or by CM4 watchdog - this mechanism is not used in the device although supported by HW.

Operational

During the operational mode, the device is capable of supporting safety critical functionality. Note that the safety critical functionality in the device commences after the boot time safety tests are performed by the Functional Application Firmware successfully.

Copyright © 2025 Texas Instruments Incorporated

4.5 Functional Safety Constraints and Assumptions

In creating a functional Safety Element out of Context (SEooC) concept and doing the functional safety analysis, TI generates a series of assumptions on system level design, functional safety concept, and requirements. These assumptions (sometimes called Assumptions of Use) are listed below. Additional assumptions about the detailed implementation of safety mechanisms are separately located in Section 6.3 .

The AWR2188 Functional Safety Analysis was done under the following system assumptions:

- **[SA_DEV_1]** The system integrator follows all requirements in the component data sheet.
- **[SA_DEV_2]** The system shall be configured to allow this device to communicate with assigned interfacing ECU to maintain proper operating state of the external system based on input from RF sensors.
- **[SA_DEV_3]** If the system is in a fault detected state, software may attempt to recover from the fault before a safety goal is violated. Software shall be configured to put the system into a safe state, if unable to recover from a fault, before the violation of a safety goal occurs.
- **[SA_DEV_4]** The system integrator shall review the recommended diagnostics in the Safety Analysis Report (FMEDA) and Safety Manual and determine the appropriate diagnostics to include in their system. These diagnostics shall be implemented according to the device Safety Manual and datasheet.
- **[SA_DEV_5]** The software shall initialize the continuous diagnostics and periodically run the test-fordiagnostics in alignment with the system safety concept including Fault Tolerant Time Interval (FTTI).
- **[SA_DEV_6]** The power supply to this device shall provide the appropriate power on each of the power inputs. These rails shall be monitored for deviations outside the device specifications.
- **[SA_DEV_7]** The power supply or other external monitoring device(s) shall monitor the device error pins and transition the system to a safe state after an unrecoverable error is indicated.
- **[SA_DEV_8]** The power supply or other external monitoring device shall monitor the Sensor Device to provide coverage for loss of availability. A watchdog is an example technique to achieve this availability monitoring.
- **[SA_DEV_9]** The power supply or other external monitoring device shall be used as a parallel path to disable downstream actuators in situations such as before the Sensor Device is capable to perform a safety function (i.e. Device safety function startup in the Host Application) or when Device faults have been detected that would compromise the decision making capability of the Device (i.e. Device fault conditions).
- **[SA_DEV_10]** A system concept shall be chosen which uses the available reciprocal comparison by software, coded processing, or other system concept in the host device to detect faults in the execution of the code running on the CPUs in AWR2188 device related to the safety function. This concept shall also take into consideration-shared modules (i.e. RAM, ADCs, etc.).
- **[SA_DEV_11]** The system integrator shall perform system configuration and implementation checks.
- **[SA_DEV_12]** Availability of system function is not a safety requirement. When the system is off, it shall be in a safe state.
- **[SA_DEV_13]** Device power sequencing requirements shall not be considered to be safety critical.
- **[SA_DEV_14]** The system integrator shall review the Device SEooC analysis and integrate it into the system level safety analysis, the diagnostics shall be applied as needed with respect to the system safety goals and requirements, and integration testing shall be performed.
- **[SA_DEV_15]** The system integrator shall analyze the other components in the system with respect to the safety concept and will implement diagnostics on those components as needed with respect to that safety concept.
- **[SA_DEV_19]** The power supply or other external monitoring device shall monitor the Sensor Device for program flow and put the system into a safe state and force a reset of the Sensor Device if the program flow is disturbed. Logical and temporal monitoring by the host device is an example technique to achieve this program flow monitoring.
- **[SA_DEV_20]** The system integrator shall use two independent clocks to ensure integrity using Clock compare modules.
- **[SA_DEV_21]** The system integrator shall provide independence in clocking between the Sensor Device and the interfacing ECU.
- **[SA_DEV_22]** The safety function shall not begin until the software enables it after this device has successfully completed its startup sequence, ROM based bootloading sequence, run any required integrity

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

checks, and is in a normal mode of operation. The safety function is assumed to be started in the Host device Application including all the necessary handshake and monitoring with the AWR2188 device.

- **[SA_DEV_23]** Debug and Design For Test (DFT) logic shall be disabled during operation of a safety function.
- **[SA_DEV_24]** The assumed ASIL allocation for the device developed as SEooC is ASILB.
- **[SA_DEV_25]** The system integrator must ensure external crystal oscillator is highly reliable and monitored continuously.
- **[SA_DEV_26]** The system integrator must ensure configurations that place redundant signals on neighboring pads or pins of the sensor device should be avoided to ensure robustness due to Dependent Failure
- **[SA_DEV_27]** If system noise testing is required, the system integrator is responsible for the implementation and monitoring of this testing.
- **[SA_DEV_28]** If an RX test is required to verify beyond observed amplitudes or phases provided by TI's monitoring test, the system integrator is responsible for the implementation and monitoring of this test.
- **[SA_DEV_28_A]** If an RX test is required to monitor the idle channel characteristics (e.g. cleanliness of spectrum and noise floor), the system integrator is responsible for the implementation and monitoring of this test.
- **[SA_DEV_29]** If an IF Loopback test is required to verify beyond observed amplitudes or phases provided by TI's monitoring test, the system integrator is responsible for the implementation and monitoring of this test
- **[SA_DEV_31]** If a TX test is required to verify beyond observed power or phase provided by TI's monitoring test, the system integrator is responsible for the implementation and monitoring of this test.
- **[SA_ANA_AOU1] :** (SA_DEV_27) : If system noise testing is required, the system integrator is responsible for the implementation and monitoring of this testing.

  Optional Diagnostics:

  SYNTH_SM6 (SYNTH Phase Noise/Spur Detection)* : Externally monitor the phase noise performance at the TX output.

  SYNTH_SM8 (SYNTH Phase Noise Fault Injection) : Test of diagnostic for SYNTH_SM6 which monitors phase noise performance by inducing high phase noise in the TX spectrum. The SYNTH FAULT feature of the AWR ANALOG FAULT INJECTION CONF SB is one possible way to control the injection of phase noise into the TX spectrum for detection.

  TX_SM6 (Bumper Reflection Amplitude Noise Test) : Similar to RF loopback test, however, with PA ON use an external object such as a bumper or antenna coupling to generate a strong signal at the receiver. Detect significant degradation in amplitude noise by observing the spectrum (skirt) of the received tone.

  TX_SM9 (Bumper Reflection Amplitude Noise Consistency Check) : Similar to RF loopback test, however, with PA ON use an external object such as a bumper or antenna coupling to generate a strong signal at the receiver. Detect significant degradation in amplitude noise by observing the spectrum (skirt) of the received tone. Repeat the tests across multiple RX chains (upto 4 RX). A simultaneous failure on all 4 RX chains is attributed to failure in the loopback signal itself.
- **[SA_ANA_AOU3] :** (SA_DEV_28) : If an RX test is required to verify beyond observed amplitudes or phases provided by TI's monitoring test, the system integrator is responsible for the implementation and monitoring of this test.

  Optional Diagnostics:

  RFANA_SM02 (RF Loopback Test – Two Tone)* : Detect non-linearities in the RX signal path (No API support today). Externally inject 2-tones into the receiver such that the intermodulation products fall within the pass band RX base-band. The measured levels of the intermodulation product represents the nonlinearities of the RX signal chain.

  Note

  RFANA_SM01 is identified alternate Safety mechanism.

  RFANA_SM15 (RF Loopback Test – Spectrum monitor)* : Detect the frequency spectrum at the RX output with an external RF input. Externally inject a signal into the RX input and collect ADC samples to perform an FFT. Analyze the FFT output spectrum to assess if acceptable.

Copyright © 2025 Texas Instruments Incorporated

Note

RFANA_SM01 is identified alternate Safety mechanism.

ADC_SM1 (ADC Linearity Test)* : Detect non-linearities arising from the ADC present in the RX signal path. Externally inject 2-tones into the receiver such that the intermodulation products fall within the stopband of the IFA HPF. The measured intermodulation product represents the non-linearities of the ADC.

Note

RFANA_SM01 is identified alternate Safety mechanism.

- **[SA_ANA_AOU4] : (SA_DEV_29) :** If an IF Loopback test is required to verify beyond observed amplitudes or phases provided by TI's monitoring test, the system is responsible for the implementation and monitoring of this test.
- **[SA_ANA_AOU5] : (SA_DEV_28_A) :** If an RX test is required to monitor the idle channel characteristics (e.g. cleanliness of spectrum and noise floor), the system is responsible for the implementation and monitoring of this test.

Optional Diagnostics:

RX_SM01 (Idle channel test)* : Detect the frequency spectrum at the RX output w/o an external RF input. With no input signal to the RX, collect ADC samples to perform an FFT. Analyze the FFT output spectrum to check if the spectrum is clean (no spurious signals) with noise floor not significantly higher than expected.

- **[SA_ANA_AOU6] : (SA_DEV_31) :** If a TX test is required to verify beyond observed power or phase provided by TI's monitoring test, the system is responsible for the implementation and monitoring of this test.

Optional Diagnostics:

SYNTH_SM7 (SYNTH Transient Frequency Monitor) : System analysis of the IF spectrum detects the presence of an incorrectly constructed ramp.

TX_SM3 (PA Loopback Spurs Test)*: Externally loopback from the output of PA to the input of LNA and analyze the spectrum of received signal to identify in-band spurious oscillations.

Optional Test for Diagnostics :

TX_SM23 (PA Loopback Spur Consistency Check)* : Use PA loopback across each RX channel to see if tones are detected.

- **[SA_ANA_AOU7]: (SA_DEV_6) :** If detection of an under-voltage or over-voltage condition of an input voltage supply is required, the system is responsible for the implementation and monitoring of this test.

Recommended Diagnostics:

PM_SM10 to PM_SM14 (External Voltage Monitoring)

The ROM bootloader is developed following the Automotive Quality (AQ) flow and the assumption is that the safety function starts after the higher level

image (MSS Functional FW) performs an end to end safing check on the loaded image and completes the boot time safety tests on the device.

- **[SA_RBL_1]:** System shall implement External Watchdog to monitor the Device Booting process. The external watchdog in this case shall be the logical and temporal monitoring of the AWR2188 device operation.
- **[SA_RBL_2] :** Device will not notify critical errors during the bootloading process and system shall handle these error by using an external watchdog timeout followed by Reset or Restart of the sequence.
- **[SA_RBL_3]:** System shall implement Power Isolation Techniques using PMIC to ensure no AWR2188 power line failures propagate and impact the System. There shall be voltage monitors deployed on the power rails that are deployed at a system level at the PMIC or external to theAWR2188 device. Any failures detected shall take the AWR2188 device and the other components powered by the PMIC to a safe state.
- **[SA_RBL_4] :** One of ISO26262:2018 recommended Safety Mechanism "End to End Safing" shall be deployed to make sure the integrity check of the Final Image (Functional FW) that are downloaded by the ROM bootloader is performed to ensure the correct operation of the ROM bootloader that is developed using the AQ flow.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

- **[SA_RBL_5]:** Functional FW as first step shall perform a check of the CRC computed by the ROM with the expected CRC value.
- **[SA_RBL_6] :** Incase of Corrupt/Faulty FW operation due to error during loading to memory or actual Memory error which caused unsuccessfully FW execution / stuck at / Abort etc, External Watchdog shall monitor such failure and necessary action shall be carried out by System.
- **[SA_RBL_7]:** PBIST on memories that the ROM bootloader loads to shall be performed. The ROM bootloader shall retain the HW/redundant status of the PBIST module as an indication that the PBIST was performed together with the SW status (pass/fail) of the execution.

Recommendations are based on TI's Assumption of use of the system, this may or may not suit actual application. It's system integrator's responsibility to evaluate and consider these recommendations, or find alternate system level solutions. During integration activities these assumptions of use and integration guidelines described for this component shall be considered. Use caution if one of the above functional safety assumptions on this component cannot be met, as some identified gaps may be unresolvable at the system level.

The system integrator may choose to reject one of the above assumptions in their system level safety analysis. In that case, the system integrator shall perform a gap analysis and determine what additional diagnostics, or alternative diagnostics need to be implemented as the system level to close the identified gap. For example, if the system integrator chooses to reject SA_AWR2188_19, then they will need to implement an alternative diagnostic for Sensor Device program flow. Additionally, the system integrator would need to disable the removed diagnostic (in this case the watchdog) inside the FMEDA and add a system integrator defined diagnostic that can provide similar coverage. Use caution when determining to reject one of the safety assumptions on this device, as some identified gaps may be unresolvable at the system level.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Chapter 5

Description of Hardware Component Parts

DRAFT
TI Confidential

Copyright © 2025 Texas Instruments Incorporated

5.1 Introduction

A semiconductor component can be divided into parts to enable a more granular functional safety analysis. This can be useful to help assign specific functional safety mechanisms to portions of the design where they provide coverage ending up with a more complete and customizable functional safety analysis. This section includes a brief description of each hardware part of this component and lists the functional safety mechanisms that can be applied to each. This section is intended to provide additional details about the assignment of functional safety mechanisms that can be found in the Safety Analysis Report. The content in this section is also summarized in Appendix A.

5.2 ADCBUF Memory

The ADC buffer is on-chip memory arranged as a ping-pong buffer, with ECC support for each ping and pong memory. The raw ADC output data from RADAR-SS is stored on this memory, to be consumed by the DSP, or by the hardware FFT accelerator for the post processing. For the application software, the ADC buffer (either ping or pong) is seen as a single memory at the base address. RadarSS generates Chirp Parameter (CP) and Chirp Quality (CQ) data along with ADC data. Interface Control Document (ICD) explains CP/CQ data format and steps to enable it.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- ADCBUF1 - SRAM ECC
- ADCBUF3 - PBIST check of memory
- ADCBUF6 - Periodic software readback of static configuration registers
- ADCBUF7 - Software readback of written configuration
- ADCBUF8 - Software test of basic functionality

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- ADCBUF3A - Software Test of PBIST
- ADCBUF4 - Software Test of Memory ECC
- ADCBUF5 - PBIST Auto-coverage

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.3 CBUFF

[Description of IP#1]

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- CBUFF2 - CBUFF DATA FIFO ECC
- CBUFF3 - A periodic software test of static configuration registers
- CBUFF4 - Software read back written configuration
- CBUFF5 - Transmission Timeout at Chirp Boundary
- CBUFF6 - A PBIST check of CBUFF functionality
- CBUFF7 - Software test of Basic functionality

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- CBUFF8 - PBIST auto coverage
- CBUFF9 - Software test of memory ECC

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

5.4 Clock

[Description of IP#1]

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- CLK2/ APLL_SM4 - PLL lock detection
- CLK3 - Dual Clock Comparator (DCC)
- CLK4 - External monitoring of OSC_CLKOUT and MCU_CLKOUT
- CLK5A - A DWWD watchdog (windowed)
- CLK5B - An external watchdog (program sequence, windowed, Q&A recommended)
- CLK6 - A periodic software test of static configuration registers
- CLK7 - Software shall read back written configuration
- PM_SM01 - Internal RC Osc
- XOSC_SM2 - XO/Slicer Frequency Detection

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- CLK8 - A test of DCC function
- CLK9 - A test of DWWD function

The following tests can be applied for fault avoidance on this module:

- CLK10 - Multi-bits keys on clock path mux selects, clock gating and divider options

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.5 CM4 MCU Subsystem

This core resides in the master subsystem. It is responsible for controlling the master sub system activites plus the the radar sub system actvities.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- CM4.CPU1B - Diversified software redundancy
- CM4.CPU1C - Reciprocal comparison by software
- CM4.CPU1E - Program Sequence Monitoring
- CM4.CPU2 - A boot time LBIST of the CPU core functionality
- CM4.CPU3 - A periodic software test of static configuration registers
- CM4.CPU4 - Software read back of written configuration
- CM4.CPU5 - A Periodic HW CRC check of SRAM contents shall be implemented
- CM4.CPU6 - Illegal operation and instruction trapping shall be implemented in the CPU
- CM4.CPU7 - A memory protection unit (MPU) shall be implemented in the CPU

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- CM4.CPU-T1 - LBIST auto coverage
- CM4.CPU-T2 - A test of Software test of illegal operation and instruction trapping including error tests
- CM4.CPU-T3 - A test of Software test of MPU functionality

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.6 CM4 MCU RAM Subsystem

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

- CM4.RAM1 - SRAM ECC on the L1 SRAM memories
- CM4.RAM2 - A hardware SRAM BIST (PBIST) engine shall be implemented to support check of SRAM functionality

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- CM4.RAM-T1 - Software test of memory ECC
- CM4.RAM-T2 - Software test of PBIST
- CM4.RAM-T3 - PBIST auto coverage

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.7 CSI2-Tx

AWR2188 integrates one two-lane CSI2 receiver interfaces in Radar processing subsystem. The prime functionality of these interfaces is to transfer raw ADC data received by the CSI2 digital physical layer receiver to the external controller for further processing.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- CSITX1 - CSI Header ECC
- CSITX2 - CRC in Message Payload
- CSITX3 - A periodic software test of static configuration registers
- CSITX4 - Software shall read back written configuration
- CSITX5 - Data Underflow detection
- CSITX6 - A PBIST check of CSI2 Tx SRAM functionality
- CSITX7 - A software test of basic functionality including error tests
- CSITX9 - Byte Parity check on CSI FIFO

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- CSITX8 - PBIST auto coverage

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.8 Device Control Registers

The device control registers of this device include registers device level configuration.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- REG1 - A periodic software test of static configuration registers
- REG3 - Software read back written configuration

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.9 Enhanced Direct Memory Access

The enhanced direct memory access module, also called EDMA, performs high-performance data transfers between two slave points, memories and peripheral devices without microprocessor unit. EDMA transfer is programmed through a logical EDMA channel, which allows the transfer to be optimally tailored to the requirements of the application.

The EDMA controller is based on two major principal blocks:

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

- EDMA third-party channel controller (EDMA_TPCC)
- EDMA third-party transfer controller (EDMA_TPTC)

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- EDM4 - DMA TPCC SRAM parity
- EDM5 - A boot time PBIST check of memory
- EDM7 - Software readback of written configuration
- EDM8 - A periodic software test of static configuration registers
- EDM16 - EDMA Legal TR Check
- EDM18 - DMA TPTC FIFO ECC
- EDM20 - Temporal monitoring of DMA shall be implemented by the Host CPU

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- EDM12 - A software test of memory parity
- EDM15 - PBIST auto coverage
- EDM19 - A software test of TPTC FIFO ECC

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.10 EFUSE

The EFUSE supports a boot time configuration of certain functionality (such as trim values for analog macros) with the help of Efuse structures. The Efuses are read automatically after power-on reset by an autoload function.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- EFU1 - Autoload self-test
- EFU2 - Efuse ECC

The following tests can be applied for fault avoidance:

- EFU6 - One Time Programmable memory shall be implemented
- EFU7- Clock Gating of EFUSE Controller

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.11 Error Signaling Module (ESM)

The ESM provides unified aggregation and prioritization of on-board hardware diagnostic errors.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- ESM1 - A periodic software test of static configuration register
- ESM2A - A boot time test of error path reporting
- ESM2B - A periodic test of error path reporting
- ESM4 - Software read back of written configuration

The error paths checks happens while performing test of other diagnostics in the design.

The following fault avoidance mechanism are used to reduce fault probability in the ESM.

- ESM6 - Multi-bit Key for MMR Fields

See Section 6.3 for more information regarding the functional safety mechanisms.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.12 General-Purpose Input/Output (GIO) Module

The General-Purpose Input/Output (GIO) module provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an output, user can write to an internal register to control the state driven on the output pin. When configured as an input, user can obtain the state of the input by reading the state of an internal register.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- GIO3 - A periodic software test of static configuration registers
- GIO4 - Software shall read back written configuration

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.13 Inter-Integrated Circuit (I2C) Interface

The multimaster Inter-Integrated Circuit (I2C) module provides a multi-master serial bus compliant to the I2C protocol.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- IIC2 - A periodic software test of static configuration registers
- IIC3 - Software shall read back written configuration
- IIC5 - Information redundancy techniques including end to end safing shall be implemented on I2C operation.

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.14 Interconnect

The device implements a system interconnect based on TI's common bus architecture, comprising of VBUSM and VBUSP protocols. The system interconnect is designed for the high-performance needs of the system.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- INC1 - Transaction error trapping (including peripheral slave error trapping)
- INC2 - PCR access management by transaction attribute. This diagnostic is applicable to the L2 interconnect.
- INC3 - Information redundancy technique
- INC4 - A periodic software test of static configuration registers
- INC5A - A boot time test of basic functionality including error test
- INC6 - Software read back written configuration
- INC8 - Access timeout capability in the interconnect in case of unresponsive slaves. This diagnostic is applicable to the L2 interconnect.
- INC9 - Temporal monitoring of other Bus masters shall be implemented by the Host CPU.

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- INC-T1 - Software test of PCR access management. This test for diagnostic is applicable to the L2 interconnect.

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

5.15 IOMUX

Includes the pin muxing logic for each IO/pin on the chip.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- IOM1 - Locking control registers shall be implemented for IO pad and muxing configuration registers.
- IOM4 - A periodic software test of static configuration registers
- IOM6 - Software shall read back written configuration

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.16 JTAG

For more information, see the *Debug Architecture* chapter of the device-specific TRM.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- JTG1 - Hardware disable of JTAG port

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.17 Power Supplies

There are multiple power supplies which are required for the right functionality of the AWR device. Voltage Monitors (VMON) are provided on all these primary supplies to detect grossly out of range supply voltages. The VMON operates periodically and requires software configuration. When the supplies are out of specified voltage range the response is as shown below in the table in “Action” column. When power supplies are in range, the VMON will not interfere with the functionality.

The response of the voltage monitor is basically categorized into two categories. One which takes the devices immediately into safe state AR1 and other which raises an interrupt to the MSS CPU core (ANA RF Supplies) and appropriate response can be programmed by the application as shown below in the table.

In addition, a temperature is provided next to Power management module to detect any kind of faults or conditions which causes excessive heat in the device. Temperature sensor readings can be accessed by the application using an API command. More details on this can be found in the later sections in the document.

It is highly recommended to have external monitoring for finely controlled thresholds as primary diagnostics for input supply and fault coverage for primary supply. In a typical system usage there are two or three supply rails from the PMIC) - as described in System Requirements – Hardware requirements section.

Along with these primary supplies, AWR device contains multiple LDOs to provide clean supply to Analog RF components of the device. “Internal Analog Signals Monitoring” implements periodic, GPADC-based internal primary supply, LDO output, and bias voltage monitoring.

Note

IO Supply VIOIN, is a special case where voltage monitor inside the device is primarily to decide the IO voltage mode for the device. System requires an external voltage monitor on this supply. One of the main reasons is also due to the fact that IO supply would be common between the xWR and other peripheral device on the sensor board. It is anyway required to monitor the IO supplies for these devices externally and take the system into the power down mode since taking the xWR device into Safe state AR1 would not be sufficient from the system perspective.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

- PM_SM11 - External Voltage Monitoring of 1.8 V input voltage
- PM_SM12 - External Voltage Monitoring of 1.0 V input voltage
- PM_SM13 - External Voltage Monitoring of 3.3 V input voltage
- PM_SM14 - External Voltage Monitoring of 1.2 V input voltage
- PM_SM111 - 1.8 V input voltage check (Internal GPADC based VMON)
- PM_SM112 - 1.0 V input voltage check (Internal GPADC based VMON)
- PM_SM113 - 3.3 V input voltage check (Internal GPADC based VMON)
- PM_SM114 - 1.2 V input (for digital) voltage check (Internal GPADC based VMON)

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.18 Quad Serial Peripheral Interface (QSPI)

The quad serial peripheral interface (QSPI) module is a kind of SPI module that allows single, dual, or quad read access to external SPI devices. This module has a memory mapped register interface, which provides a direct interface for accessing data from external SPI devices and thus simplifying software requirements. The QSPI works as a master only.The one QSPI in the device is primarily intended for fast booting from quad-SPI flash memories(if required).

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- QSPI2 - QSPI flash contents (Firmware/Application images, key store, Calibration parameters,diagnosis data) CRC shall be implemented
- QSPI12 - Temporal and Logical monitoring method to detect faults in QSPI

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.19 Multi-Buffered Serial Peripheral Interface Module (MibSPI)

The MibSPI is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (2 to 16 bits) to be shifted in and out of the device at a programmed bit-transfer rate. Typical applications for the SPI include interfacing to external peripherals, such as I/Os, memories, display drivers, and analog-to-digital converters.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- SPI2 - Information redundancy techniques including end to end safing
- SPI3 - A periodic software test of static configuration registers
- SPI4 - Software read back written configuration
- SPI5 - Transmission redundancy techniques
- SPI6 - Data overflow detection
- SPI7 - Data underflow detection

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.20 RESET

Two device resets are available that can be controlled from the device pins: the power-on reset pin NRESET and the bidirectional WARM_RESET signal. The warm reset signal is implemented as an I/O, so that an external monitor can be used to detect changes to the state of the internal warm reset control signal.

Device registers can capture recent reset events, and can be used by software to manage failure recovery. Certain registers in the device are immune to WARM_RESET, and can only be reset by power-on reset.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- RST2 - Software check of last reset reason shall be implemented
- RST5 - Multibit shall be implemented for Reset Generation Control register
- RST6 - Glitch filtering on reset pin
- RST8 - A periodic software test of static configuration registers
- RST9 - Software shall read back written configuration

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.21 Real Time Interrupt (RTI)

The real time interrupt (RTI) module instance present in each subsystem provides the operating system timer for the Subsystems. The OS timer function is used to generate internal event triggers or interrupts as needed to provide periodic operation of safety critical functions.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- RTI1 - 1oo2 timer voting using the secondary free running counter
- RTI2 - A periodic software test of static configuration register
- RTI3 - Software shall read back written configuration
- RTI5 - Software test of basic functionality including error tests

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.22 Radar Subsystem

The AWR2188 product relies on the ARM® Cortex-R4F CPU in RSS for controlling the Analog/RF components and monitoring the functionality of these components. BSS executes digital diagnostics and test of diagnostics at boot up however all the applicable diagnostics should be enabled (if abpplicable) by application through API configuration as mentioned in ICD document.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- CPU1B - Diversified software redundancy
- CPU1E - Program Sequence Monitoring diagnostic
- CPU2 - A hardware logic built in self test controller (LBIST STC) shall be implemented to support boot time test of both the CPU core functionality
- CPU3 - A memory protection unit (MPU)
- CPU4 - CPU operation profiling via the hardware performance monitoring unit (PMU)
- CPU5 - Illegal operation and instruction trapping shall be implemented in the CPU
- CPU6 - A periodic software test of static configuration registers
- CPU7 - Software shall read back written configuration
- RAM1 - ECC on the TCMs for use on SRAM data
- RAM2 - TCM interface address and control signals shall incorporate a parity
- RAM5B - A PBIST check of SRAM functionality
- RAM10 - Software test of TCM interface parity
- ROM1 - ECC on the TCMs for use on ROM data
- ROM2 - TCM interface address and control signals shall incorporate a parity
- ROM5B - A PBIST check of ROM functionality
- ROM10 - Software test of TCM interface parity

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- CPU9 - LBIST auto coverage
- RAM11 - PBIST auto coverage
- RAM12 - Software test of memory ECC
- ROM9 - Software test of CRC module
- ROM11 - PBIST auto coverage
- ROM12 - Software test of memory ECC

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.23 RF/Analog Modules

The radar device's RF analog and digital front end subsystems include monitoring features described in this section. The monitoring architectures are broadly classified as follows.

The first set of monitors uses a shared ADC to measure various internal voltage levels.

- Internal Analog Signals Monitoring:
  - Various signals of static or slow varying nature are multiplexed and forwarded to a shared ADC, called GPADC (a General Purpose 10 bit ADC).
  - The internal analog signals thus monitored include power supply nodes and other DC voltages (e.g. common mode and gate biases voltages) of various analog circuits, mmwave signal swing levels at various nodes in the signal chain, and the oscillator control voltages of APLL and FMCW synthesizer.
- External Analog Signals Monitoring:
  - External pin voltages can be monitored using the shared GPADC. Please refer GPADC(Digital) section for the same
- Temperature Sensors:
  - Multiple on-chip temperature sensors are placed near power hungry subsystems and their outputs are measured using the shared GPADC for this monitoring.

The next set of monitors relies on on-chip mmwave power detectors.

- TX Power Monitoring and TX Ball Break Monitoring:
  - Power Detectors are placed at various nodes in the mmwave signal chain in the TX, RX and LO subsystems to measure and monitor mmwave swing levels.
  - These are used to measure RX absolute gain, TX output power and also to detect TX ball break.

The next set of monitors relies on various loopback signals in the RF Analog sub system.

- RX Loopback Test:
  - A test RF signal is generated from the LO subsystem and fed symmetrically to all RX channels' LNA inputs through a weak capacitive coupling.
  - This signal's amplitude and phase at the RX ADC outputs, and power at LNA inputs are determined to estimate various properties of the RX subsystem (e.g. absolute RX gain, inter RX imbalances in gain and phase, noise figure).
- RX IF Loopback Test:
  - A test square wave signal is generated using a square wave DAC and fed to the RX IF stages.
  - This signal's amplitude at the RX ADC output is determined to estimate various properties of the RX IF stages (e.g. gain and filter cutoff frequencies).
- TX Loopback Test:
  - The PA outputs from all TX channels are symmetrically coupled to a combiner and QPSK modulator and coupled to the RX channels' LNA inputs.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

  - This loopback signal's amplitude and phase from different TX channels under different settings are used to estimate various properties of the TX subsystem (e.g. inter TX imbalances in gain and phase, Tx Gain phase monitor and phase accuracy).

The next set of monitors relies on estimating relative frequency of clocks in BSS.

- Synthesizer Chirp Frequency Monitor:
  - During radar chirping, the FMCW synthesizer output signal's frequency is instantaneously measured using a reference signal's frequency and compared with the ideal ramp frequency.
- BSS Clock Monitors:
  - These monitors observe the relative frequency of various clock pairs in the relevant systems. The clocks thus monitored include internal RC oscillator clock, XTAL clock and APLL derived clocks such as GPADC input clock, BSS processor clock, Digital Front End clock, Chirp Timing Engine clock.

There are miscellaneous monitors not summarized here but described subsequently.

All these monitors are configurable by the application layer. The BSS processor performs these monitors during the inter-burst times (when chirps are not ongoing) and reports the results, at a configurable periodicity. The periodicity is configurable through the message AWR_CALIB_MON_TIME_UNIT_CONF_SB, based on the user's choice of FTTI. The set of monitors to execute cyclically at this periodicity is configurable through the message AWR_MONITOR_ANALOG_ENABLES_CONF_SB. The BSS processor has automatic scheduling capability: it breaks down the monitoring functions into multiple small units (lesser than approximately 200us) and executes that many steps which can be executed in the available time in each inter-burst gap and schedules the remaining steps for the subsequent inter-burst gaps, till all the monitors are completed. A simplistic illustration of this is provided in the figure below. The approximate total time consumed in execution of various monitors is documented in the sub section "Monitoring duration" in the section "Calibration and monitoring durations" in the ICD. The user should ensure that the monitoring periodicity and frame configuration programmed are such that there is sufficient time for the BSS processor to complete all the enabled monitors in the inter-burst durations.

These monitors are further described in multiple sub sections which follow. The messages to configure the device's monitoring functionality are described in the ICD under the section "Radar Monitoring APIs". That section also indicates the corresponding reporting messages, which too are described in the ICD.

5.23.1 GPADC and BIST FFT

GPADC captures the different tapping points in the Analog/RF transmit and receive chain for monitoring and calibration. GPADC stands for General Purpose ADC and would be used to sample the outputs from the various monitoring modules in the Analog/RF Subsystem. The output samples would be fed into the GPADC buffer in BISTSS and would be used for post processing by BIST R4F. Typical monitoring modules sampled by GPADC are temperature sensors and Power detectors which are sprinkled all across the device.

BIST FFT module is used by BSS CPU for RF/analog calibration and monitoring functions that require tones outside of the 15 MHz bandwidth (e.g., LPF cut-off monitoring).

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

GPADC

- GPADC1 - PBIST check of GPADC RAMs
- GPADC2 - A repeated test and sanity checking with previous results test of function
- GPADC5 - PBIST auto coverage
- GPADC6 - Software read back written configuration
- GPADC7 - ECC on the GPADC Buffer
- GPADC9 - Software Selftest of GPADC with know reference voltage values

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

- GPADC8 - A software test of GPADC ECC logic

BIST FFT

- FFT1 - Boot time PBIST of FFT RAMs
- FFT2 - A repeated test and sanity checking with previous results test of function
- FFT3 - A boot time test of function using Loopback pattern
- FFT6 - Software read back written configuration

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- FFT5 - PBIST auto coverage

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.23.1.1 Temperature Sensors

The device supports monitoring of on-chip temperature. Temperature sensors are placed near significant power consuming sub systems in the device as summarized below.

Temprature Sensor	Placement
TEMP_RX1	Near Receiver RX1 Analog
TEMP_RX2	Near Receiver RX2 Analog
TEMP_RX3	Near Receiver RX3 Analog
TEMP_RX4	Near Receiver RX4 Analog
TEMP_RX5	Near Receiver RX5 Analog
TEMP_RX6	Near Receiver RX6 Analog
TEMP_RX7	Near Receiver RX7 Analog
TEMP_RX8	Near Receiver RX8 Analog
TEMP_TX1	Near Transmitter TX1 Analog
TEMP_TX2	Near Transmitter TX2 Analog
TEMP_TX3	Near Transmitter TX3 Analog
TEMP_TX4	Near Transmitter TX4 Analog
TEMP_TX5	Near Transmitter TX5 Analog
TEMP_TX6	Near Transmitter TX6 Analog
TEMP_TX7	Near Transmitter TX7 Analog
TEMP_TX8	Near Transmitter TX8 Analog
TEMP_PM	Near Power Management & Reference Generation Analog
TEMP_DIG1A	Near Digital cluster A
TEMP_DIG1B	Near Digital cluster B

The outputs of temperature sensors can be measured using the GPADC. The measured temperatures can be compared against thresholds programmed by the application layer. The threshold checks possible include minimum and maximum tolerable temperatures, separately configurable for sensors associated with the analog and digital sub systems; and one for a tolerable difference across all temperature sensor readings (for latent fault coverage).

The relevant configuration and soft reporting messages are described in the ICD in the section “mmWaveLink Monitor APIs”.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

- PM_SM03 - Temperature Monitor : Detect over-temperature or under-temperature operating conditions near RX, TX, Power Management and Digital Main Subsystem
- PM_SM50 - Temp sensor consistency check : Provision for multiple temperature sensors at different spots in the die.

5.23.2 Internal Analog Signals Monitoring

The voltage levels of various internal nodes in the device can be monitored using a GP (General Purpose) ADC. The parameters that can be monitored using this include power supply nodes and other DC voltages (e.g. common mode and gate biases voltages) of various analog circuits, mmwave signal swing levels at various nodes in the signal chain, and the oscillator control voltages of APLL and FMCW synthesizer.

The monitors falling under this category are further described in the following sub sections. For each of them, the host can configure the set of parameters to be monitored. The BSS can alert the host when the measured voltage exceed safety thresholds, and send soft reports containing information on which parameters set failed and/or measured values, once every monitoring periodicity.

5.23.2.1 PLL Control Voltage Monitoring

The device contains two Phase Lock Loops in Analog Sub System: the APLL and FMCW Synthesizer. The APLL (Analog PLL) generates the clock for the digital subsystems as well as RX ADC sampling and FMCW synthesizer reference.

The VCO's input control voltage adapts to compensate for variations in process, temperature and required output frequency. In the case of the FMCW synthesizer, there are two different VCOs to cater to different frequency ranges.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- APLL_SM5 - APLL VCO Control Voltage Detection
- SYNTH_SM5 - SYNTH VCO Control Voltage Detection

5.23.2.2 TX Internal Analog Signals Monitoring

The device supports monitoring of various internal analog signals in the transmitter analog subsystem.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- PM_SM23 - PM DCBIST (Internal VMON) : Detect under-voltage/over-voltage condition on internal PA PM [0-3] block's output
- PM_SM112 - Input voltage check (Internal VMON) : Internal voltage monitor for 1.0V input voltage (from PMIC) using GPADC
- TX_SM7 - TX Phase Shifter DAC Monitor and Fault Injection
- TX_SM21 - TX Gain and Phase Monitor Fault Injection

5.23.2.3 RX Internal Analog Signals Monitoring

The device supports monitoring of various internal analog signals in the receiver analog subsystem.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- PM_SM24 - PM DCBIST(Internal VMON) : Detect under-voltage/over-voltage condition on internal ADC PM block's output
- PM_SM25 - PM DCBIST(Internal VMON) : Detect under-voltage/over-voltage condition on internal LNA PM block's output
- PM_SM27 - PM DCBIST(Internal VMON) : Detect under-voltage/over-voltage condition on internal IFA PM block's output
- PM_SM112 - Input voltage check (Internal VMON) : Internal voltage monitor for 1.0V input voltage (from PMIC) using GPADC

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

- PM_SM40 - Band-gap Reference Voltage Check : Detect under-voltage/over-voltage condition on Band-gap reference

5.23.2.4 PM, CLK and LO Systems Internal Analog Signals Monitoring

The device supports monitoring of various internal analog signals in the CLK (Clock generation) and LO (local oscillator signal generation) analog subsystem.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- PM_SM21 - PM DCBIST (Internal VMON) - Detect under-voltage/over-voltage condition on internal Synth PM block's output
- PM_SM22 - PM DCBIST (Internal VMON) - Detect under-voltage/over-voltage condition on internal APLL PM block's output
- PM_SM111 - Input voltage check (Internal VMON) - Internal voltage monitor for 1.8V input voltage (coming from PMIC) using GPADC
- PM_SM112 - Input voltage check (Internal VMON) - Internal voltage monitor for 1.0V input voltage (coming from PMIC) using GPADC
- PM_SM113 - Input voltage check (Internal VMON)- Internal voltage monitor for 3.3V input voltage (coming from PMIC) using GPADC
- PM_SM114 - Input voltage check (Internal VMON) - Internal voltage monitor for 1.2 V input voltage using GPADC(coming from PMIC)

5.23.2.5 GPADC Monitoring

All the analog signals in the internal analog signals monitoring architecture are multiplexed into the shared GPADC for monitoring. The device supports monitoring of the GPADC itself using the same architecture. For this monitoring, the BSS processor can forward analog signals of different known voltage levels to the GPADC and observe the GPADC output levels. It can alert the host if the measured levels deviate from expected levels and can provide the measured values as soft information too. The BSS processor performs these measurements once in every monitoring period as configured.

The relevant configuration and soft reporting messages are described in the ICD in the section “mmWaveLink Monitori APIs”.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- PM_SM61 - GPADC Consistency Check : GPADC diagnostic test comparing multiple known supply voltages for consistency.

5.23.3 TX Power Monitoring

The device supports the monitoring of TX output power using internal mmwave power detectors.

For each transmitter, the TX output power can be monitored using the incident power detector coupled to it. This monitoring can be done when the functional chirps are not active, with the BSS scheduling non-functional test chirps.

The host can configure the tolerable power deviation thresholds for this monitor. The BSS can alert the host if the measured power deviates from the programmed power by more than the configured threshold, and also send soft reports containing measured power statistics, once every monitoring periodicity.

The relevant configuration and soft reporting messages are described in the ICD in the section “mmWaveLink Monitor APIs”. There are different messages to cater to each transmitter.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- TX_SM10 - PA output power detector : Monitor the power at the PA output to detect degradation in output power.

DRAFT ONLY TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

5.23.4 TX Ball Break Monitoring

The device enables detection of TX ball break using internal mmwave power detectors.

The device has built-in capability to measure the incident and reflected powers using power detectors at each TX output. The magnitude of the reflection coefficient at the transmitter's output port can be determined using these. This monitoring can be done when the functional chirps are not active, with the BSS scheduling a test chirp (static frequency). TX ball breaks are assumed to result in a high degradation in the reflection coefficient.

The host can configure the tolerable thresholds for the reflection coefficient magnitude. The BSS can alert the host when the measured reflection coefficient exceeds the configured threshold, and also send soft reports containing the measured parameters, once every monitoring periodicity.

The relevant configuration and soft reporting messages are described in the ICD under the sub section "TX Ball Break Monitor" in the section "Radar Monitoring APIs". There are different messages to cater to each transmitter.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- TX_SM1 - Impedance Detector at PA Pin : The directional coupler at the PA output pins are used to measure the incident and reflected power to infer impedance variance on the board due to failures such as ball break.

5.23.5 RX Loopback Test

The receivers' gain, phase and noise figure, and imbalance in gain and phase across receivers can be monitored for gross failures, using the Internal RX RF Loopback architecture in the device.

The loopback architecture includes capacitive coupling of the output of an internal node in the transmitter subsystem to the inputs of the LNAs of all the receivers, symmetrically.

5.23.5.1 RX Gain, Phase and Noise Monitoring

When no functional chirps are ongoing, for the purpose of measuring RX gain and noise figure, inter-RX gain and phase imbalances, the BSS can schedule test chirps (narrow bandwidth), with the TX PAs switched off to suppress external emissions and to avoid parasitic reflections. During this, the loopback signal reaching the LNAs is intentionally offset from the mixer LO frequency by a few MHz using built-in frequency shifter, to avoid corruption by RX baseband DC offsets and loopback signal attenuation by RX baseband high pass filters. The RF power measured by the power detectors before the LNA and the ADC output power are used to estimate the RX gain. The imbalances in the amplitudes and phases of the loopback signal across the RX channels, as measured at the ADC output, are reported as the inter-RX gain and phase imbalances. Failures of the balls will typically cause inter-RX imbalances that would be detected by this monitor. The gain measurements can be performed at different RF frequencies in order to compute the RX gain flatness. The ADC output noise power with the TX and loopback signal generation switched off (idle channel) is computed to find the RX noise figure.

Though the signal of interest during these measurements is only the loopback signal, it should be noted that these internal measurements are impacted by inter-RX antenna coupling and the antenna loading the RX LNAs. Due to these, the signals that get measured for each channel are not just the signal of interest in that channel but additionally, a small amount of coupled signal from the neighboring channels and also a small amount of the signal of interest further reflected by the antennas due to impedance mismatch (non-ideal loading of the RX LNAs by the antenna). These cause a minor deviation in the estimated values of the RX properties (gain and inter channel imbalances) compared to the actual RX properties.

The gain and phase monitors detect failures in RX RF and ADC sections while suitably reconfiguring RX IFA (assumed ideal) for the measurements. The noise monitor reuses the gain monitor's estimated gain and measures the noise in the RX RF, IFA and ADC.

The host can configure the RF frequencies to be monitored, the tolerable thresholds for deviation of gain, noise figure and imbalances from expectations. The BSS can alert the host when the measured RX parameter deviations exceed the configured thresholds, and also send soft reports containing statistics related to the measured parameters, once every monitoring periodicity.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

The relevant configuration and soft reporting messages are described in the ICD in the section “mmWaveLink Monitor APIs”.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- RFANA_SM01 - RF loopback Test : Device supports internally loopback of the RF signal from before the TX PA to the RX LNA input, this feature helps to process the loopback signal in digital baseband to monitor its power level (Gain) and Noise Figure. Any failures in this path impacts Gain, Phase and Noise measured. In order to avoid the loopback signal from falling at DC at the output of the RX mixer, "non-zero IF" mode is enabled for the loopback signal test, using the I/Q modulators in the TX path.
- RFANA_SM04 - RF Loopback Signal Consistency Check : The RF loopback test is executed across multiple RX chains (upto 4 RX). A simultaneous failure of the gain/phase/linearity on all 4 RX chains is attributed to failure in the loopback signal itself. The decision is taken by the Host looking at the monitoring data.

5.23.6 RX IF Loopback Test

The receivers’ IF Amplifier (IFA) gain and filter attenuations can be monitored using the RX IF Loopback architecture in the device. The loopback architecture includes generation of square waves of programmable frequency and feeding them to the IFAs.

When no functional chirps are ongoing, for the purpose of measuring RX IFA gain and filter attenuations, the BSS can schedule loopback test data collection. During this, the loopback signal of known amplitude is enabled with the loopback frequency set to be in-band and in high pass and low pass filter transition bands, and the loopback signal’s amplitude at the ADC output is measured. The IFA gain and filter attenuation parameters are computed from these measurements.

The host can configure the tolerable thresholds for deviation of the IFA parameters. The BSS can alert the host when the measured IFA filter parameters deviate from expectations by more than the configured threshold, and also send soft reports containing the measured parameters, once every monitoring periodicity.

The relevant configuration and soft reporting messages are described in the ICD in the section “mmWaveLink Monitor APIs”.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- RFANA_SM05 - IF Loopback Signal Consistency Check : The IF loopback test is executed across multiple IFchains. A simultaneous failure of the gain/phase/linearity on all 4 RX chains is attributed to failure in the loopback signal itself. The decision is taken by the Host looking at the monitoring data.
- RX_SM02 - IF loopback Test : IF test tone is fed into the IF chain (IFA1 input) on both I and Q channels. This test signal will be processed in digital baseband to ascertain its level, Noise Figure, etc.
- RX_SM04 - IF filter response Test : Similar to the IF loopback test, IF test tone is fed at multiple frequencies, covering the rolloff region of the HPF, passband and rolloff region of the LPF. Ascertain the power level of the test tone through digital baseband processing to check the IF filter's frequency response behavior.

5.23.7 Synthesizer Chirp Frequency Monitoring

This device supports monitoring of Synthesizer’s output frequency during active chirping. This is executed during dummy radar chirps. This is done by a hardware digital state machine which compares the synthesizer’s instantaneous output frequency with the ideally expected ramp frequency by using a clock derived from the APLL as reference.

During a normal chirp, the FMCW synthesizer generates the LO clock with the start frequency and slope as programmed by the application layer. Each rising edge of the synthesizer output clock causes a high speed counter to increment. The counter is sampled on the monitor’s reference clock derived from the APLL. Difference of successive samples is computed to convert the counter value (an indicator of synthesizer phase) to estimated synthesizer frequency. There is a state machine that generates the ideally expected instantaneous frequency words on every reference clock. The difference of the estimated instantaneous frequency and the ideal is passed through a noise averaging filter and compared against user programmed thresholds. The threshold

Copyright © 2025 Texas Instruments Incorporated

comparison results (number of threshold violations in a monitoring duration and maximum frequency error during the monitoring duration) are maintained in a running statistics logger.

The host can program the tolerable frequency error thresholds and also when the threshold based monitoring should begin with respect to the start of the ramp. It should be noted that the threshold comparison continues till the end of the ramp. If the frequency error threshold comparison failed during the preceding monitoring period, the BSS processor alerts the application layer and sends soft reports containing frequency error threshold comparison statistics (number of threshold violations in a monitoring duration and maximum frequency error during the monitoring duration) once in every monitoring periodicity as configured.

The relevant configuration and soft reporting messages are described in the relevant ICD in the section “mmWaveLink Monitor APIs”.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- SYNTH_SM3 - SYNTH VCO Clock Frequency Linearity Detection Using TDC: (SYNTH Chirp Frequency Monitoring): : Monitor SYNTH with frequency linearity measurement circuitry. Frequency linearity measurement circuitry compares a counter clocked at 20G effective rate within one 100MHz sample.

5.23.8 Clock Monitoring (RSS)

The device supports autonomous monitoring of relative frequency between various clock pairs in the RSS using built-in dual clock comparators. The list of clocks thus monitored and their respective ideal frequencies are tabulated below.

Clock Name	Comments
APLL	
FRC	
DFE	
RSS Core	
RC Osc.	
GPADC	

The BSS processor performs this monitor during inter frame duration once in a monitoring duration. It alerts the host processor if any of these clock pairs are found to be violating the error thresholds and sends as soft reports the measured frequency values, once every monitoring periodicity as configured. Details on the Diagnostics mechanism are captured as part of Clock generation section.

The relevant configuration and soft reporting messages are described in the ICD in the section “mmWaveLink Monitoring APIs”.

5.23.9 RX Saturation Detector and Its Monitoring

This section describes the device’s RX saturation detector feature and also the monitoring of that RX saturation detector.

5.23.9.1 RX Saturation Detector

The device has provision to detect saturation in the RX IF stages and ADCs during the active chirping, due to excessive reflections and interference. Saturations in IFA1 output as well as ADC inputs are detectable through independent comparators.

The host can configure the division of the chirp durations into multiple pieces (time slices) for the purpose of detecting and reporting saturation events. It can also configure whether IFA saturations or ADC saturations or both are needed to be monitored and which of the RX channels need to be monitored. For each time slice, a summary of the saturation counts across all enabled RX channels is communicated along with the chirp’s ADC data as part of Chirp Quality (CQ) Metrics through the high speed data lanes.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

The relevant configuration message is described in the ICD in the section “mmWaveLink Monitor APIs”.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- RX_SM06 - RX Saturation Detector (ADC saturation check) : Analog hardware monitor which checks the stability of the ADC and flags any onset of instability.

5.23.9.2 RX Saturation Detector Monitoring

The device’s fault injection feature may be used if the user chooses to find if the RX Saturation detector itself failed. The fault injection feature repurposes the square wave IFA loopback signal generator explained in the RX IF Stages Monitoring section for injecting a high signal condition which triggers saturation events in the IFA and ADC. During this test, the user may configure high RX gain so as to cause saturation events.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- RX_SM23 - RX Saturation Detector Monitor : RX Saturation Monitor Fault Injection test configuring high RX gain test signal which triggers saturation events in IFA and ADC.

5.23.10 TX Loopback Test

The primary diagnostic mechanisms to detect TX faults are implemented in TX power monitor and TX internal analog signal monitor. The TX Loopback Tests described in this section provide redundancy to these monitors.

The transmitters can be monitored for gross failures in their gain, phase and inter-channel imbalances, using the Internal TX Loopback architecture in the device.

The loopback architecture includes capacitive coupling and symmetric combining of outputs of PAs of all transmitters, on-off-keying modulation (OOK Mod.) of the combiner output and coupling to the inputs of the receiver LNAs.

5.23.10.1 TX Gain and Phase Mismatch Monitoring

When no functional chirps are ongoing, for the purpose of measuring inter-TX gain and phase imbalances, the BSS can schedule test chirps (narrow bandwidth). During this, the loopback signal reaching the LNAs is intentionally offset from the mixer LO frequency by a few MHz using the QPSK modulator. The purpose of QPSK is to separate out the desired internal loopback signal from any external reflections with respect to IF frequency. The test chirps are scheduled for one TX at a time, and the loopback signal’s amplitude and phase is measured from the RX ADC outputs in each TX’s test chirp. The imbalances in the amplitude and phases across TXs is reported as inter-TX gain and phase imbalances.

Though the signal of interest during these measurements is only the loopback signal, it should be noted that these internal measurements are impacted by inter-TX antenna coupling and the antenna loading the TX PAs. Due to these, the signals that get measured for each channel are not just the signal of interest in that channel but additionally, a small amount of coupled signal from the neighboring channels and also a small amount of the signal of interest further reflected by the antennas due to impedance mismatch (non-ideal loading of the TX PAs by the antenna). These cause a minor deviation in the estimated values of the TX properties (gain and inter channel imbalances) compared to the actual TX properties.

The host can configure the RF frequencies, output power and phase shift under which the TX is to be monitored, the tolerable thresholds for deviation of imbalances from expectations. The BSS can alert the host when the measured TX parameter deviations exceed the configured thresholds, and also send soft reports containing statistics related to the measured parameters, once every monitoring periodicity.

The relevant configuration and soft reporting messages are described in the ICD in the section “mmWaveLink Monitor APIs”.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

- TX_SM2 - PA Loopback Gain/Phase Test : Loopback test from the output of PA to the input of LNA is used to measure the Gain/[phase of the transmitted signal wrt phase of LO signal at the receiver. Running this test sequentially for all transmitters provides an estimate of Gain/phase balance between transmitters.

5.23.10.2 TX BPM Phase Monitoring

TX BPM phase monitoring is similar to the inter-TX gain and phase mismatch monitor described above, with the following minor differences. Accuracy of TX BPM is measured for each TX using a pair of successive test chirps with opposite BPM settings. The loopback signal's amplitude and phase difference between the test chirps is computed and deviation from ideal expectation (0dB and 180 degree difference) is found and reported.

The host can configure the RF frequencies, output power and phase shift under which the TX is to be monitored, the tolerable thresholds for deviation of imbalances from expectations. The BSS can alert the host when the measured TX parameter deviations exceed the configured thresholds, and also send soft reports containing statistics related to the measured parameters, once every monitoring periodicity.

The relevant configuration and soft reporting messages are described in the ICD in the section "mmWaveLink Monitor APIs".

5.23.11 Analog Fault Injection

The device supports software based fault injection to mimic the impact of faults in various RF, analog and front end subsystems.

Faults are expected to be injected or un-injected only when no radar frames are ongoing. Further, only one fault is expected to be activated at a time. Faults can be injected and un-injected using the fault injection API message, AWR_ANALOG_FAULT_INJECTION_CONF_SB, which is described in the ICD under the section "Analog Fault injection".

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- PM_SM74 - DCC Clock Frequency Monitor Fault Injection
- RX_SM20 - RX Gain and Phase Monitor Fault Injection
- SYNTH_SM20 - SYNTH Frequency Error Monitor Fault Injection
- TX_SM21 - TX Gain and Phase Monitor Fault Injection

5.23.12 RF and Analog Configuration Register

The entire configuration registers corresponding to Analog and RF modules are collectively called as BSS_ANA. BSS Firmware is only software that configures RF and Analog Configuration registers, RSS shall ensure safety mechanisms mentioned below are executed and this needs to be enabled through issuing respective API.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- ANA18 - A periodic software test of static configuration registers
- ANA19 - Software shall read back written configuration

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.24 Digital Front-end Filters

ADC data from the Receiver chains is passed through the Digital front end filter (DFE) chain before sending it outside device. The main purpose of this chain is to do pre-conditioning of the data from the ADC before feeding it external to device for radar signal processing.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- DFE1 - Boot Time Execution of LBIST
- DFE2 - Periodic Time Execution of LBIST

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

- DFE4 - A Parity test of function
- DFE5 - SRAM ECC
- DFE6 - A periodic software test of static configuration registers
- DFE7 - Software shall read back written configuration
- DFE8 - Boot time PBIST check of DFE SRAM
- DFE12 - Dual Clock Comparator (DCC)

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- DFE9 - PBIST auto coverage
- DFE10 - A test of SRAM ECC
- DFE13 - A test of DCC function
- DFE18 - LBIST auto coverage

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.25 Ramp Generation

Ramp Sequencer functionality is split across BSS_RAMPGEN (in RSS) and Free running counter (in RSS) modules. RAMPGEN allows additional flexibility of updating the configuration of specific filters in real time. Sequencer extension SEQ_EXTN module inside RAMPGEN reads the address and corresponding data from the configuration RAMs (ADDRESS_RAM and DATA) and writes into corresponding registers at the defined timings. BSS_RAMPGEN module is the main timing engine of the chip. It initiates and controls the chirp cycles in the frame/burst while FRC (Free running counter) provides the ticks for the frame or burst boundaries. The engine is designed such that no SW intervention is needed throughout the frame by direct interaction with the Synthesizer and many other HW blocks in the chip. Engine contains memories which are programmed by CPU for various chirp profile and chirp parameters and other FE configurations (Extension to Sequencer module) required to perform the functionality.

The following diagnostic mechanisms are available for this module. Note that these modules (BSS_RAMPGEN and FRC) are controlled by the BIST R4F processor running TI's application, all the Diagnostic mechanisms are managed by the TI SW itself and does not require any intervention from the application.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

Sequencer

- SEQ2 - Boot time PBIST check of Chirp Profile and Chirp Config SRAM functionality
- SEQ3 - Chirp profile and Chirp config SRAM ECC
- SEQ4 - A periodic software test of static configuration registers
- SEQ5 - Software read back written configuration
- SEQ6 - Lockstep Ramp Gen compare diagnostic
- SEQ10 - Parity on the Data Path flops

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- SEQ7 - A self-test of lockstep compare logic
- SEQ8 - PBIST auto coverage
- SEQ9 - Software Test of Memory ECC

Sequencer Extension

- EXT2 - Boot time PBIST of Sequencer Extension RAM
- EXT3 - SEQ EXT SRAM ECC
- EXT5 - Software read back previous written configuration
- EXT6 - Time out mechanism for the SEQ EXT functionality

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- EXT7 - PBIST auto coverage
- EXT9 - Lockstep for Sequencer Extension FSM
- EXT10 - Lockstep compare self-test
- EXT11 - Software Test of Memory ECC

The following test can be appliaed as fault avoidance on this module

- EXT8 - Bit multiplexing in memory array

Free Running Counter(FRC)

- FRC2 - Lockstep diagnostic mechanism
- FRC4 - Logical monitoring diagnostic mechanism shall be implemented at a system level
- FRC5 - A periodic software test of static configuration registers
- FRC6 - Software shall read back written configuration

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- FRC8 - A test of Lockstep compare function

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

5.26 Vectored Interrupt Manager (VIM)

The Vectored Interrupt Manager (VIM) aggregates device interrupts and sends them to the Core. It can be used in either split or lockstep configuration.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- VIM1 - VIM SRAM ECC
- VIM2 - A hardware SRAM BIST (PBIST) engine shall be implemented to support boot time check of VIM SRAM functionality
- VIM5A - A boot time test of function including error tests
- VIM6 - A periodic software test of static configuration registers
- VIM7 - Software read back written configuration
- VIM11 - A hardware logic built in self test controller (LBIST STC) shall be implemented to support boot time test of both the VIM module functionality

The following tests can be applied as test-for-diagnostics on this module to test the functionality of other diagnostics on the primary function (that are listed above):

- VIM8 - A test of VIM SRAM ECC logic
- VIM9 - Software test of PBIST
- VIM10 - PBIST auto coverage

See Section 6.3 for more information regarding the functional safety mechanisms.

See the *Table of Contents* for Section 5 for more descriptions of hardware component parts.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Chapter 6

AWR2188 Management of Random Faults

For a functional safety critical development it is necessary to manage both systematic and random faults. The AWR2188 component architecture includes many functional safety mechanisms, which can detect and respond to random faults when used correctly. This section of the document describes the architectural functional safety concept for each sub-block of theAWR2188 component. The system integrator shall review the recommended functional safety mechanisms in the functional safety analysis report (FMEDA) in addition to this safety manual to determine the appropriate functional safety mechanisms to include in their system. The component data sheet or technical reference manual (if available) are useful tools for finding more specific information about the implementation of these features.

6.1 Fault Reporting

When a diagnostic mechanism detects a fault, the error must be reported. The device's architecture provides aggregation of fault indication from internal safety mechanisms using a module/peripheral logic known as the Error signaling module (ESM). The ESM module provides mechanisms to classify errors by severity and to provide programmable error response. Faults during the power up sequence are mapped directly so that they can be reported without the software intervention, which might not be possible if the fault occurs before the CPU gains the control. The error classifications in the ESM are summarized in table below.

Error Group	Interrupt Response	Error Pin Response	Notes
1	Programmable interrupt and Priority	Programmable Response	For generally non crtical errors
2	Non-maskable interrupt generated	Activated	For errors that are generally of crtical severity
3	No interrupt response	Activated	For crtical errors that are seen by diagnostic in CPU

The device uses a hierarchical approach in Error Reporting Mechanism. The device as mentioned earlier is split into three Subsystems (RADAR Subsystem (RSS), Master Subsystem (MSS) and DSP Subsystem (DSS)). Each Subsystem can be configured to handle the low severity interrupts/errors within the Subsystem and report the high severity errors/interrupts to one level up controller in the hierarchy, to be handled over there. The hierarchy assumed is as shown in the figure above and as

RSS, DSS -> MSS -> External Monitoring (Higher) (e.g Power Management)

Main CPU CM4 handles the RSS High severity Errors, DSS High severity errors (as configured) and all MSS errors itself. Low severity errors from MSS diagnostics are handled by CN4 itself and reported to ECU via SPI/I2C communication. High Severity Errors from MSS can be directly configured to generate 'Nerror' output signal which need to be monitored by external monitoring circuitry to take the sensor unit into safe state.

RSS R4F controls and monitors the functionality of the Analog/RF, DFE, Timing engine and other RSS logic through various hardware and software based diagnostics. The software running on RSS R4F is TI proprietary and all low severity errors are handled by RSS R4F itself and reported to Main CPU CM4 via MAILBOX. High severity errors can be configured to be reported to MSS CM4 via MSS ESM's Error group #1 input (described in next section).

All errors from the DSS are also routed to the input channels of MSS ESM.

DRAFT
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

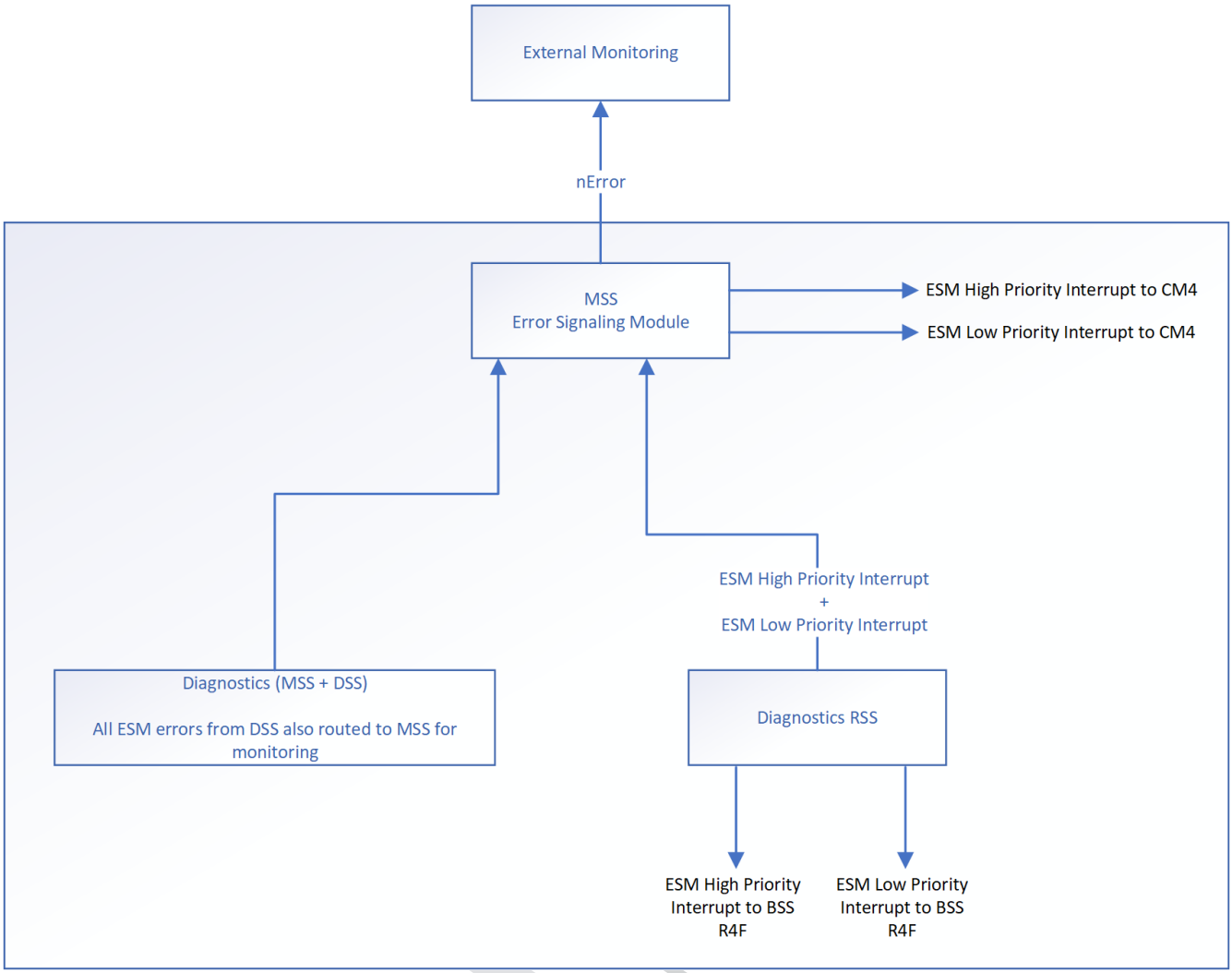

Figure 6-1. Error reporting and response flowchart

The error response is action that is taken by the device when an error is indicated. There are multiple mechanisms of error response possible. The system integrator is responsible to determine what error response should be taken and to ensure that this is consistent with the system level 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 Interrupt - This response can be implemented for diagnostics outside 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 nPORRST - This response allows the device to change state to safe state from any other state. It is also possible to move to the powered-down state, if desired, to implement a system level safe state (with the help of external PMIC and MCU). This response can be generated from the internal voltage monitor (particularly when there is a problem with the supply- nPORRST is not propagated down into the design hierarchy), but is primarily driven by monitors external to the AWR2188 device.

The ESM provides multiple registers that can be read by the CPU to determine the current status of diagnostics and the state of the nERROR pin. For the severe group 2 errors, a shadow register is provided that is not reset by warm reset. This allows the possibility of warm reset re-initialization to identify that a group 2 error initiated the external reset.

Copyright © 2025 Texas Instruments Incorporated

It is possible for the CPU to trigger the nERROR pin response manually to test system behavior or to notify external logic of an internal fault not automatically indicated to ESM. The CPU is responsible to clear indicated errors in the ESM, including clearing of the nERROR pin response.

The external error reporting scheme is based on defined behavior of the nERROR pin. When nPORRST is active, the nERROR pin is in a high impedance state (external Pullup on board). When warm reset is active, the nERROR pin is in an output active state with pull down disabled; behavior does not change when warm reset is deactivated.

Upon a detected fault the nERROR pin will drive a low signal state to indicate a failure. AWR2188 device has a fixed time window to service the request, which can be programmed during device initialization (in an external monitoring logic- PMIC device) by software. If the AWR2188 device resets the error state within the time window, the nERROR signal will transition to a logical high state when the window expires. If the window expires and the nERROR pin remains in low state, external monitoring logic (Host CPU) should take action to reset or disable the device.

Care should be taken to match the device's external error response to the capabilities of existing or planned system basis chips. System level management of the external error response with currently defined products can be simplified using a TI power supply and safety companion device developed for use with the Radar and camera sensor/ADAS family.

6.2 Functional Safety Mechanism Categories

This section includes a description of the different types of functional safety mechanisms that are applied to the design blocks of the AWR2188 component.

The functional safety mechanism categories are defined as follows:

Component Hardware Functional Safety Mechanisms	A safety mechanism that is implemented by TI in silicon which can communicate error status upon the detection of failures. The safety mechanism may require software to enable its functionality, to take action when a failure is detected, or both.
Component Hardware and Software Functional Safety Mechanisms	A test recommended by TI which requires both, safety mechanism hardware which has been implemented in silicon by TI, and which requires software. The failure modes of the hardware used in this safety mechanisms are analyzed or described as part of the functional safety analysis or FMEDA. The system implementer is responsible for analyzing the software aspects for this safety mechanism.
Component Software Functional Safety Mechanisms	A software test recommended by TI. The failure modes of the software used in this safety mechanism are not analyzed or described in the functional safety analysis or FMEDA. For some components, TI may provide example code or supporting code for the software functional safety mechanisms. This code is intended to aid in the development, but the customer shall do integration testing and verification as needed for their system functional safety concept.
System Functional Safety Mechanisms	A safety mechanism implemented externally of this component. For example an external monitoring IC would be considered to be a system functional safety mechanism.
Test for Safety Mechanisms	This test provides coverage for faults on a safety mechanism only. It does not provide coverage for the primary function.
Alternative Safety Mechanisms	An alternative safety mechanism is not capable of detecting a fault of safety mechanism hardware, but instead is capable of recognizing the primary function fault (that another safety mechanism may have failed to detect). Alternate safety mechanisms are typically used when there is no direct test for a safety mechanism.

6.3 Description of Functional Safety Mechanisms

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Copyright © 2025 Texas Instruments Incorporated

DRAFT
TI Confidential – NDA Restr

Copyright © 2025 Texas Instruments Incorporated

6.3.1 Introduction

This section provides a brief summary of the functional safety mechanisms available on this component.

6.3.2 1oo2 Software Voting Using Secondary Free Running Counter

The RTI module contains at minimum two up-counters that can be used to provide an operating system time-tick. While one up-counter is used as the operating system timebase, it is possible to use the second up counter as a diagnostic on the first, via periodic check via software of the counter values in the two timers. The PMU CPU cycle counter inside the CPU can also be used to support such a diagnostic. Error response, diagnostic testability, and any necessary software requirements are defined by the software implemented by the system integrator. The use of a second counter to diagnose faults in RTI is recommended.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.3 A boot time test of error path reporting

Application Boot time test of error path reporting shall be incorporated by a combination of hardware and software. Nerror Pin is monitored based on the Error path selected. A software test can be utilized to test basic functionality including the interrupt triggers,status updates and clears. Software requirements necessary are defined by the software implemented by the system integrator.

Boot time diagnostics typically cannot be claimed during Normal operation to cover Single Point Failures, nevertheless it is essential for Safety Island approach, of using a module only after it's tested. This will help in capturing any faults upfront.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.4 A boot time test of function using Loopback pattern

The FFT can be set to a particular configuration, and a time domain signal of a known frequency (complex/real single tone) can be passed to the FFT using the test source in the DFE chain. The output of the FFT can be analyzed by the CPU to ascertain correct FFT operation. This test is recommended to be performed during device boot time and at desired interval based on the system usecase.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.5 Access timeout capability in the interconnect

The interconnect subsystem includes a number of mechanisms to detect and trap errors. Address decoders in the PCR interconnect respond with a bus error to the initiator if a bus transaction does not decode to a valid target. Logic is also present in PCR that can detect the timeout of certain transactions and respond with a bus error to the transaction initiator. Peripheral slaves which are connected to PCR interconnect can also support error trapping. The interconnect error trapping functionality is enabled by default and cannot be disabled by the software.

Use of this safety mechanism is mandatory. These features can be tested via software through the insertion of invalid bus transactions.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.6 A periodic test of error path reporting

A software test can be utilized to test basic functionality. Such a test can be executed periodically. Software requirements necessary are defined by the software implemented by the system integrator. Nerror Pin is monitored based on the Error path selected. A software test can be utilized to test basic functionality including

Copyright © 2025 Texas Instruments Incorporated

the interrupt triggers, status updates and clears. Software requirements necessary are defined by the software implemented by the system integrator.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.7 A repeated test and sanity checking with previous results test of function

To detect permanent and transient errors in any module or system for example say the FFT computation for calibration, repeated FFT computation (Kind of Transmission redundancy) can be triggered by CPU on the same input data and outputs compared and evaluated for consistent results by the CPU. This input is fed from the Analog Subsystem.

To detect permanent and transient errors in the GPADC computation for calibration, repeated GPADC computation can be triggered by CPU on the same input data and outputs compared and evaluated for consistent results by the CPU.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.8 A test of lockstep compare function

The lockstep comparator diagnostic includes self test features to check for proper operation of the lockstep comparator. A basic test of Compare Module functionality (including error generation) is possible via software by configuring appropriate XOR input/output inversion of the outputs of the second core.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.9 Autoload Self-Test

Efuse autoload self-test continuously operates (upon each power-on reset which causes efuse read). Efuse Controller has a self-test logic that automatically executes at reset after autoload completion. No software configuration of hardware shall be needed to initialize this diagnostic.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.10 Boot Time and Periodic Software Test of Error Path Reporting

A software test is recommended to inject diagnostic errors and check for proper reporting. Such a test can be executed at boot or periodically. Software testing of the ESM error path can be combined with boot time latent tests of hardware diagnostics to reduce startup time. Necessary software requirements are defined by the software implemented by the system integrator. Testing of ESM error path may result in assertion of the Nerror diagnostic output rather NERROR output can itself be tested simultaneously. System integrator should ensure that the system can manage or recover gracefully from the Nerror event.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.11 Boot Time LBIST of CPU core and Vectored Interrupt Controller

The architecture supports the use of a hardware logic BIST (LBIST) engine. This logic is used to provide a very high diagnostic coverage on selected modules (CPUs and designated hardware accelerators) at a transistor level. This logic utilizes the same design for test (DFT) structures inserted into the device for rapid execution of high quality manufacturing tests, but with an internal test engine rather than external automated test equipment (ATE). This technique has proven to be drastically more effective than software-based tests of logic, particularly for the complex logic structures seen in a modern CPU.

See Section 5 for more information regarding the hardware component parts.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.12 Byte Parity check on CSI FIFO

The CSI2 FIFO has a byte Parity implemented on the memory. This FIFO is used to store the data received over the slave port before it is packed as per the format and split across the lanes. Error response is triggered upon Parity error.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.13 Clock Gating of EFUSE Controller

The EFUSE Controller is clock gated after EFUSE Shift is complete after NRESET.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.14 Coded processing

Coded processing is a software method described in ISO 26262-5:2011 Annex D and IEC 61508-7 Annex A. The exact implementation is left for the system integrator to determine.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.15 CPU operation profiling via the hardware performance monitoring unit (PMU)

The Cortex-R4F CPU includes a performance monitoring unit (PMU). This logic is intended to be used for debug and code profiling purposes, but it can also be utilized as a safety mechanism. The PMU includes a CPU cycle counter as well as three 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-R4 Technical Reference Manual.

Examples of the CPU events that can be monitored include:

- Number of cycles in which an ECC or parity error is detected by diagnostics in CPU
- Number of cycles in which the CPU is in the livelock state
- Number of instructions executed
- Number of cycles in which an exception is taken

With such information available, it is possible to generate a software routine that periodically checks the PMU counter values and compares these values to the profile expected during normal operation. The PMU is not enabled by default and must be configured via software. As the PMU is implemented internal to the CPU, it is checked for proper operation on a cycle by cycle basis by the lockstep diagnostic when the CPU is operating in lockstep mode. The PMU functionality can be checked with a software test of function when the CPU is operating in lockstep mode. It can also be checked in both modes via execution of the LBIST diagnostic. Use of the PMU for online diagnostic profiling is optional.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.16 CRC in Message Payload

As per MIPI CSI2 Specification, to detect possible errors in transmission, a checksum is calculated over each data long packet. The checksum is realized as 16-bit CRC. The generator polynomial is x16+x12+x5+x0.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

6.3.17 Data overflow detection

If the McPI RX buffer receives new data before the previous data has been read, the existing data will be overwritten and lost. If this occurs, the hardware can flag the error and generate an interrupt to the CPU. This feature must be enabled and configured in software. Use of Data Overflow Detection is highly recommended.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.18 Data underflow detection

If the McSPI is enabled but the buffer didnot receive any data to be transmissted the underflow condition happens. If this occurs, the hardware can flag the error and generate an interrupt to the CPU. This feature must be enabled and configured in software. Use of Data Underflow Detection is highly recommended.

The CSI2 Protocol Engine will generate an interrupt to processor if the FIFO used on the slave port for buffering the data received on the slave port has under-flowed in the middle of a packet transfer.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.19 Diversified software redundancy

Diversified software redundancy is a software method described in ISO 26262-5:2011 Annex D. The exact implementation is left for the system integrator to determine.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.20 Dual Clock Comparator (DCC)

One or more Dual Clock Comparators (DCCs) are implemented as multi-purpose safety diagnostics. The DCC can be used to detect incorrect frequencies and drift between clock sources. The DCC is composed of two counter blocks: one is used as a reference timebase and a second is used for the clock under test. Both reference clock and clock under test may be selected via software, as can the expected ratio of clock frequencies. Deviation from the expected ratio generates an error indication to the ESM. For more information on the clock selection options implemented, see the device-specific data sheet.

The DCC diagnostic is not enabled by default and must be enabled via software. It is possible to disable and configure this diagnostic via software. Use of the DCC as a diagnostic for clocks is highly recommended. The cyclical check applied by the DCC module provides an inherent level of self-checking (autocoverage), which can be considered for application in latent fault diagnostics.

RSS internal clocks including sequencer clocks are also checked against a reference XTAL and APLL clocks. This is done using the DCC. This should capture if there are faults in the clock used in the DFE.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.21 DWWD (Internal Watchdog)

This device supports the use of an internal watchdog that is implemented in the Real Time Interrupt (RTI) module. The internal watchdog is used in Digital Windowed Watchdog (DWWD) mode of operation.

The DWWD is a traditional windowed watchdog. The user programs an upper bound and lower bound to create a time window during which the software must provide a predetermined "pet" response to the watchdog. Failure to receive the correct response within the time window or an incorrect "pet" response triggers an error response. The DWWD can issue either an internal (warm) system reset or a CPU nonmaskable interrupt upon detection of a failure. The DWWD is not enabled after reset. Once enabled by the software, the DWWD cannot be disabled

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

except by system reset or power-on reset. The use of the time window allows detection of additional clocking failure modes as compared to the DWD implementation. Use of the DWWD functionality is recommended.

Internal Watchdog works on the CPU clock, An additional clock monitoring logic is used to monitor the watchdog clock using an independent reference clock like XTAL or RCOSC .If the Watchdog clock indicates a deviation from the expected frequency, a WD reset or a WD NMI can be issued.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.22 EDMA Legal TR Check

The EDMA provides checks on fields of the TR (Transfer Request) to ensure that the EDMA is capable of performing the requested transaction. This can be used to catch some errors if the error results in the TR being submitted no longer have legal values.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.23 Efuse ECC

Each FROM row is of 41 bits wide with a total of 32 bits of data (31:0). This 32 bits of data is divided into 26 bits of actual data and 6 bits of SECDED ECC. ECC covers the data alone and for every 26-bits of data there is a 6 bit ECC. ECC can be globally enabled with a tie-off at the controller level. Or alternately with per row ECC enable which is in the 32nd bit of each row. Radar devices enforce ECC per row with the global override.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.24 Error Trapping (including peripheral slave error trapping)

The system interconnect includes a number of mechanisms to detect and trap errors. Address decoders in the diagnostic respond with a bus error to the initiator if a bus transaction does not decode to a valid target. Logic is also present that can detect the timeout of certain transactions and respond with a bus error to the transaction initiator. The interconnect error trapping functionality is enabled by default and cannot be disabled by the software. These features can be tested via software through the insertion of invalid bus transactions.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.25 External monitoring of OSC_CLKOUT and MCU_CLKOUT

MCU_CLKOUT and OSC_CLKOUT is externally monitored by clock monitoring logic externally. This is applicable only in the cases that the MCU_CLKOUT and OSC_CLKOUT is used to clock external on board components.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.26 External Watchdog

When using an External Watchdog, there is a possibility to reduce common mode failure with the Multi-Core Processor (MCP) clocking system, as the watchdog can utilize 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. It is highly recommended to use an External Watchdog in addition to or instead of the internally provided watchdogs.An internal or External Watchdog can provide an indication of inadvertent activation of logic which results in impact to safety critical execution. Any watchdog added externally should include a combination of temporal and logical monitoring of program sequence or other appropriate methods such that high diagnostic effectiveness can be claimed.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.27 Glitch Filtering on Reset Pins

Glitch filters are implemented on the cold and warm reset pins of the device. These structures filter out noise and transient signal spikes on the input reset pins in order to reduce unintended activation of the reset circuitry. The glitch filters are continuously operating and their behavior cannot be changed by the software. Use of the glitch filters is mandatory.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.28 Hardware disable of JTAG port

The JTAG debug port can be physically disabled to prevent JTAG accesses in deployed systems. The recommendation scheme is to Hold Test Clock (TCK) to ground and hold Test Mode Select (TMS) high, although alternate schemes are possible.

Disabling the JTAG port also provides coverage for inadvertent activation of many debug and trace activities, since these are often initiated via an external debug tool which writes commands to the device using the JTAG port.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.29 Illegal operation and instruction trapping

The Cortex-M4F / Cortex-R4F 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 the software. Installation of software handlers to support the hardware illegal operation and instruction trapping is highly recommended. For more information on enabling traps, see the Cortex-M4 Technical Reference Manual.

Examples of CPU illegal operation and instruction traps include:

- Illegal instruction
- Floating-point underflow and overflow
- Floating point divide by zero
- Privilege violation

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.30 Information redundancy techniques

Information redundancy techniques can be applied via software as an additional runtime diagnostic for this module. In order 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. There are many different schemes applied, such as additional message checksums, message or frame counters or other similar mechanisms. 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 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.

Error response, diagnostic testability, and any necessary software requirements are defined by the system integrator.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

6.3.31 LBIST auto coverage

The LBIST diagnostic is based on a N-bit signature capture. For a given test, only one code is valid out of 2^N possibilities. Therefore, if there is a fault in the LBIST logic, it is extremely unlikely that the correct passing code will be generated via the fault. The cyclical check applied by the LBIST module provides an inherent level of self checking (auto-coverage), which can be considered for application in latent fault diagnostics.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.32 Logical monitoring diagnostic mechanism

CPU shall implement an independent check FRC timing signals by comparing the FRC counter values with its own PMU/RTI reference counters. These checks will be initiated at periodic chirp/frame start or end interrupts that constitutes the critical timing parameters of the radar waveform.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.33 Lockstep diagnostic mechanism

The entire FRC counter and the various output generation logic is duplicated and the final count values and output signals from both "Original" and "Duplicate" module are compared. If there is any difference in count values or any of the output signals between "Original" and "Duplicate" module in any clock, an error interrupt is flagged to the CPU and the status will indicate which of the enables are causing the error interrupt.This comparison will be running always whenever FRC is ON. Both cores run off same clock without any clock skew between them.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.34 Lockstep Ramp Gen compare diagnostic

Sequencer is very critical from the safety perspective and hence even the control logic is well covered by the delayed lock step Safety mechanisms. A diagnostic sequencer runs in parallel to the main sequencer with a delay of 2.5 clock cycles. Both the sequencer cores generate the same output. A comparator compares both the outputs and flags any mis-compares to the ESM.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.35 Lock Mechanism for Control Registers

The module contains a lock mechanism for protection of all IO PAD control registers. There are lock registers implemented as a part of configuration registers that has to programmed with specific value and sequence, to enable access to other IO PAD configuration registers.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.36 Multi-bit Key for MMR Fields

Three ESM registers include multi-bit keys to protect the operation of ESM.

Global Enable Register (base + 0x08) has a 4-bit key with one state defining disable, one state defining enable, and all other states resulting in enable.

Global Soft Reset Register (base + 0x0C) has a 4-bit key with one state (not the default state) clearing all raw status and enable bits and all other states having no effect.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Error Pin Control Register (base + 0x40) has a 4-bit key with one state (the default state) defining the normal mode, one state defining the force error mode, one state defining the clear event mode, and all other states resulting in continued normal mode operation.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.37 Memory ECC

ECC is a mechanism for providing increased protection against soft errors by allowing single bit errors to be detected and corrected and double bit errors to be detected. An ECC memory always protects against single-bit errors in the data. Address decode logic for the memory is not covered by the memory ECC logic. There is an ECC aggregator module inside the module that aggregates status from the ECC memories into a single interrupt to the host. The aggregator also supports software readable status of ECC single/double-bit errors and associated info such as RAM address and data bit(s) that are in error.

The ADCBUF RAM is supported by SECDED ECC diagnostic. It is connected by a 128-bit wide data bus interface to the Interconnect. In this SECDED scheme, an 9-bit word is used to store the ECC data as calculated over the 128-bit data bus. The ECC generation logic for the ADCBUF is activated on a Write from DFE. ECC evaluation is done by the ECC control logic on a Read from the Bus Matrix.

As per MIPI CSI2 Specification, the correct interpretation of the data identifier and word count values is vital to the packet structure. The Packet Header Error Correction Code byte allows single-bit errors in the data identifier and the word count to be corrected and two-bit errors to be detected.

The GPADC Data RAM is supported by SECDED ECC diagnostics.In this SECDED scheme,.The ECC logic is diabled at reset and must be configured in the CPU.The memory wrapper for RAMs has separate controls for checking,correction and read-modify and write functionality that must be enabled via software .The ECC self test performed by the CPU by introducing different kinds of erros and checking the expected response is used as test for diagnostics.The data and ECC are stored in separate physical memory banks, hence it indirectly covers the address decode logics checks.

The on-chip ROM is supported by SECDED ECC diagnostic. It is connected by a 64-bit wide data bus interface (ATCM0/ATCM1, BTCM0/BTCM1) to the Cortex-R4F CPU. In this SECDED scheme, an 8-bit word is used to store the ECC data as calculated over the 64-bit data bus. The ECC logic for the TCM ROM address is located in the Cortex - R4F CPU. All TCM transactions have ECC on the data payload. ECC evaluation is done by the ECC control logic inside the CPU. This scheme provides end-to-end diagnostics on the transmissions between CPU and ROM. Detected uncorrectable errors in a processor abort or bus error depending on the requesting master. Detected correctable errors can be corrected or not corrected, depending on whether correction functionality is enabled. The address of the memory that includes the ECC error will be logged in the CPU. For more details, see the Cortex-R4 and Cortex-R4F Technical Reference Manual.

The Chirp Config RAM and Profile RAM is supported by SECDED ECC diagnostics. In this SECDED scheme, a 9-bit code word and 8-bit code word is used to store the ECC data as calculated over 128-bit data bus and 64-bit data bus for Chirp Config and Profile RAM respectively. The ECC logic is disabled at reset and must be configured in the CPU. The memory wrapper for RAMs has separate controls for checking, correction and read-modify and write functionality that must be enabled via software. ECC self-test performed by the CPU by introducing different kinds of errors and checking the expected response is used as test for diagnostics.The data and ECC are stored in separate physical memory banks. ECC is supported only for the RAM data not for address.

Redundant Address Decoder Logic: The address decode for the data memory and ECC memory has been implemented separately. This gives a redundancy in terms of address decode. Any error in the address decode between data memory and ECC memory will result in an ECC error being indicated.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

6.3.38 Memory Parity

Parity is a mechanism for providing protection against single bit errors, either permanent or transient (soft errors), in memory registers. Parity errors detected in protected memory registers will result in an event being asserted to indicate the detection of the error and information identifying the location of the error is logged for software to use in diagnosis. A shared module interface, which is referred to as an ECC aggregator, provides the system integrator with access to the error detection status and logging information.

For the memories inside the IPs (TPCC), parity is implemented for each byte of the memory word. The errors from these modules are sent to ESM along with the parity error address.

Parity is disabled by default on reset. The parity bit is updated on writes even when parity is disabled. When Test mode is enabled, on a write, the parity is inverted and written. In the case of multiple errors, the addr status is not updated till the previous error is cleared.

When parity is disabled there is no interrupt even on parity error and the addr status bits reflect 0x0.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.39 Memory Protection Unit (MPU)

This device implementation of the Cortex-M4F CPU includes an MPU. The MPU logic can be used to provide spatial separation of software tasks in the device memory by doing virtual to physical address translation and enforcing access policies. It is expected that the operating system controls the MPU and changes the MPU settings based on the needs of each task. A violation of a configured memory protection policy results in a CPU abort. Use of the MPU is highly recommended.

The MPU functionality can be checked with a software test of function for proper operation and error response. In addition, the LBIST diagnostic provides a check of the MPU when it performs a test of the CPU.

The mWave implementation of the Cortex-R4F CPU includes an MPU. The MPU logic can be used to provide spatial separation of software tasks in the device memory. It is expected that the operating system controls the MPU and changes the MPU settings based on the needs of each task. A violation of aconfigured memory protection policy results in a CPU abort. Use of the MPU is highly recommended.The MPU can also be used to configure the memory ordering policies of the memory system. By default,all peripheral accesses are strongly ordered - meaning that all transactions complete in the sequence areissued and no write transactions are buffered. If desired, the operating system can configure accesses tobe device - meaning that writes are buffered. This can improve performance over a strongly orderedmodel, at the cost of some determinism. It is highly recommended that the system module and othermodules deemed to have critical configurations be set to a strongly ordered access model.As the MPU is internal to the CPU core, proper operation is checked via the lockstep CPU mechanism. Inaddition, the LBIST STC diagnostic provides a check of the MPU when it performs a test of the CPU.Additional software-based testing of the MPU for proper operation and error response is optional.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.40 Multi-bit Key for MMR Fields

One ESM register has multi-bit keys to protect the operation of ESM. Error Pin Control Register (base + 0x38) has a 4-bit key with one state (the default state) defining the normal mode, one state defining the force error mode, one state defining the clear event mode, and all other states resulting in continued normal mode operation.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

Copyright © 2025 Texas Instruments Incorporated

6.3.41 Multi-Bits Keys on Clock Path mux Selects, Clock Gating and Divider Options

Important configuration control registers are implemented as multi-bit(3 bits).A '1' is encoded as '111' and '0' as '000'. A single bit flip due to transient error will not result in change of clock mux and clock gating configuration.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.42 Multi-bits on critical flops

Important configuration control registers are implemented as multi-bit (3 bits). A '1' is encoded as '111' and '0' as '000'. A single bit flip due to transient error will not result in change of trigger source select and FSM enable.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.43 Multi-Bits Keys on Reset Generation Control Registers

Important configuration control registers are implemented as multi-bit(3 bits). A '1' is encoded as '111' and '0' as '000'. A single bit flip due to transient error will not result in reset generation.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.44 One Time Programmable memory

A one time programmable memory is implemented to store the Efuse settings so that they do not get altered over time or by the user.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.45 Parity test of function

MSB flip flops in the datapath which can cause high noise floor increase if faulty are checked for transient faults using a parity check mechanism. Any bit flip in one of those flip flops will cause the parity bit to be different from the expected value and be detectable as a fault.RADAR Devices supports parity logic on the msb bits across all the modules in the DFE chain. The intent is to catch any kind of gross failures which impacts the msb in the data pre-conditioning logic and it is assumed that the lsb bits corruption would anyway get absorbed due to averaging.Note that the Radar system provides inherent redundancy due to multiple Tx and Rx chains and any kind of gross transient failures in one of the chains can be easily caught at the application level. Though parity on the MSB bits is assumed to give sufficient coverage on such failures, any kind of concern on the LSB bits should be handled at the application level.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.46 PBIST Auto-coverage

The PBIST diagnostic is based on a 32-bit signature capture. For a given test, only one code is valid out of 2^32 possibilities. Therefore, if there is a fault in the PBIST logic, it is extremely unlikely that the correct passing code will be generated via the fault.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.47 PBIST Check of SRAM

Hardware programmable memory BIST (PBIST) engine can be used to test SRAM. PBIST can be triggered by the application and any fault detected by the PBIST results in an error indicated in PBIST status registers. This

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

logic is used to provide a very high diagnostic coverage on the implemented SRAMs at a transistor level. This logic utilizes the same design for test (DFT) logic and algorithms used by TI to provide rapid execution of high quality manufacturing tests. This technique has proven to be drastically more effective than software-based tests of SRAM, particularly for the devices with complex CPUs whose addressing modes do not enable an optimal software-based test.

The PBIST tests are destructive to memory contents, and as such are typically run only at the boot-up. However, the user has the freedom to initiate the tests at any time when the CPU is operable. The most effective SRAM tests known to TI require test across a full physical memory module and are destructive to memory contents. If PBIST is to be implemented during operation, it is recommended to copy the data from the SRAM to be tested to a non-tested memory before test execution and to restore the data once the test is complete. When test execution is complete, the SRAM can be utilized for normal operation. The remainder of device logic continues normal operation during SRAM test. Any fault detected by the PBIST results in an error indicated in PBIST status registers. It is also possible to log faults in the PBIST logic.

Use of the PBIST logic at device boot-up is highly recommended. Use of the PBIST logic for periodic execution during normal execution is optional. The cyclical check applied by the PBIST logic provides an inherent level of self checking (autocoverage), which can be considered for application in latent fault diagnostics.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.48 PCR Access Management

The peripheral central resource (PCR) provides two diagnostic mechanisms that can limit access to peripherals. Peripherals can be clock gated per peripheral chip select in the PCR. This can be utilized to disable unused features such that they cannot interfere with active safety functions. In addition, each peripheral chip select can be programmed to limit access based on privilege level of transaction. This feature can be used to limit access to entire peripherals to privileged operating system code only. These safety mechanisms are disabled after reset. Software must configure and enable these mechanisms. Use of these mechanisms is highly recommended.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.49 Periodic HW CRC check of SRAM contents

A software test can be utilized to test basic functionality of the SRAM contents covering the interconnect. Such a test can be executed at boot or periodically. Software requirements necessary are defined by the software implemented by the system integrator.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.50 Periodic Software Readback of Static 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 implementation. The use of read back of configuration registers mechanism is recommended.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.51 Periodic test of function using LBIST

The mmWave family architecture supports the use of a hardware Logic Built In Self Test (LBIST) engine using Self Test Controller (STC). This logic is used to provide diagnostic coverage on the DFE data path at a transistor level. This technique has proven to be drastically more effective than software-based tests of logic, particularly for the complex logic structures seen in a modern CPU. The LBIST tests must be triggered by the software. User may elect to run the test based on the execution time, which can be allocated to the LBIST diagnostic. The

Copyright © 2025 Texas Instruments Incorporated

LBIST mechanism requires isolation of the DFE from the remainder of device logic while under test. It is also necessary to perform a complete context save before the LBIST.

LBIST logic includes capabilities for testing proper operation of the diagnostic. As the test time for the diagnostic is deterministic, a timeout counter has been included that can detect a failure to complete the test within expected time.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.52 PLL Lock Detection

Digital Clock comparator module - DCCA is used to monitor the PLL lock detection with respect to reference clock source. This has to be enabled by software and should be used in a continuous operating mode to check the availability of the PLL clock and the PLL frequency. If lock detect fails, because of PLL issue or any other logic in the clock path, the muxes would switch to limp mode (System switches to RCOSC clock) and generate an ESM error and propagate NERROR to host using RCOSC clock. Presence of an RCOSC clock and clock switch circuitry is required to propagate the NERROR. MSS_DCCA is dedicated for APLL lock detection monitoring comparing the APLL output divided version with the Reference input clock of the device.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.53 Program Sequence Monitoring

Program sequence monitoring (temporal and logic) are a software methods described in ISO 26262-5:2011 Annex D and IEC 61508-7 Annex A. The exact implementation is left for the system integrator to determine.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.54 QSPI flash contents CRC

A CRC is computed on the data written into the flash. When this data is received by QSPI the CRC is computed again and compared in Hardware.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.55 Reciprocal comparison by software diagnostic

Reciprocal comparison by software is a software method described in ISO 26262-5:2011 Annex D and IEC 61508-7 Annex A. The exact implementation is left for the system integrator to determine.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.56 Sequencer extention FSM Lockstep compare diagnostic

Sequencer extension FSM (Finite State Machine) is very critical from the safety perspective and hence even the control logic is well covered by lock step Safety mechanisms. A diagnostic FSM runs in parallel to the main sequencer with no cycle delay. Both the sequencer cores generate the same output. A comparator compares both the outputs and flags any mis-compares to the ESM.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

6.3.57 Software Check of Last Reset

The RCM module provides a status register SYSRSTCAUSE that latches the cause of the most recent reset event. A boot software that checks the status of this register to determine the cause of the last reset event is typically implemented by software developers. This information can be used by the software to manage failure recovery.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.58 Software readback of written configuration

In order to ensure proper configuration of memory-mapped control registers in the module, it is highly recommended that software implement a test to confirm proper operation of all control register writes. To support this software test, it is highly recommended to configure the target memory space as a strongly ordered, non-bufferable memory region for the CPU using the memory protection unit (MPU) or memory management unit (MMU) for the CPU. This ensures that the register write has completed before the read back is initiated.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.59 Software selftest of GPADC with know reference voltage values

The Firmware in the BSS subsystem, periodically performs the BIST operation for GPADC function. The know Reference Voltage value is converted and verified against the expected results.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.60 Software test of Basic functionality including error tests

A software test can be utilized 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. Software requirements necessary are defined by the software implemented by the system integrator.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.61 Software test of CRC

Software test of CRC is done by the ROM by taking a known code and creating CRC and checking the CRC output.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.62 Software Test of DCC Operation

A basic test of Dual Clock Comparator (DCC) functionality (including error generation) is possible via software by programming a sequence of good and bad expected clock ratios and executing DCC operations with software confirming expected results.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.63 Software Test of DWWD Operation

A basic test of DWWD operation can be performed via software including checking of error response by programming expected "pet" thresholds/windows and servicing or not servicing the "pet" requests. For example the preload value being set with a given value and window size is defined to max100%t. Reconfigure the DWWD

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

within the active period and check the expected value. For error path check, the DWWD is not reconfigured within active period window with NMI disable to avoid spurious interrupt halt and WDT status flag is check to see the failed watchdog status. If the status is not set than the test of diagnostic is considered failed.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.64 Software test of illegal operation and instruction trapping including error tests logic

A software test can be utilized to test basic functionality of the illegal operation and instruction trapping as well as inject errors and check for proper error response. Such a test can be executed at boot or periodically. Software requirements necessary are defined by the software implemented by the system integrator.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.65 Software Test of Memory ECC

It is possible to test the functionality of ECC detection logic by forcing an ECC error into the data output from the memory, and seeing if the ECC detection logic reports an error. A shared module interface, which is referred to as an ECC aggregator, provides the system integrator with access to configure the logic and force errors. Reporting of forced errors uses same mechanism that reports unforced errors.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.66 Software Test of Memory Parity and ECC

It is possible to test the functionality of parity and/or ECC detection logic by forcing a bit error into the data output from the memory, and seeing if the parity and/or ECC detection logic reports an error. A shared module interface, which is referred to as an ECC aggregator, provides the system integrator with access to configure the logic and force errors. Reporting of forced errors uses same mechanism that reports unforced errors.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.67 Software test of MPU functionality including error tests logic

A software test can be utilized to test basic functionality of the MPU as well as inject errors and check for proper error response. Such a test can be executed at boot or periodically. Software requirements necessary are defined by the software implemented by the system integrator.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.68 Software Test of PBIST

It is possible to configure the PBIST logic by selecting an algorithm that should fail, and seeing if the PBIST Logic reports an error under this condition. For example, a read-write test could be performed on a read-only memory to ensure that a failure is reported. An alternative scheme is to run a test for a memory with more bits than are actually present on the target memory.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.69 Software test of PCR access management

As a test-for-diagnostic, software can configure and enable safety mechanisms. The peripheral central resource (PCR) provides two safety mechanisms that can limit access to peripherals. Peripherals can be clock gated per peripheral chip select in the PCR. This can be utilized to disable unused features such that they cannot interfere

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

with active safety functions. In addition, each peripheral chip select can be programmed to limit access based on privilege level of transaction. This feature can be used to limit access to entire peripherals to privileged operating system code only. These safety mechanisms are disabled after reset.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.70 Software test of TCM interface parity

As a test-for-diagnostic, it is possible to test the functionality of Parity Error Detection logic by forcing a parity error into the Data or Parity RAM, and seeing if the Parity Error Detection logic reports an error.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.71 TCM interface address and control signals parity diagnostic

The on-chip TCM bus connections to ROM are supported by a parity diagnostic on the address and control signals. The parity is generated by the CPU and evaluated by the ROM. Detected errors are signaled to the ESM by the ROM and the error address is captured in the ROM wrapper. This diagnostic is enabled at reset. Use of the BTCM address and control bus parity diagnostic is highly recommended.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.72 Temporal and Logical Monitoring

Temporal and logical monitoring is done by the CPU. For EDMA temporal monitoring, once CPU configures the DMA, it knows which transfer shall complete and how much transfer time it would take to complete. The HWA can be programmed to generate interrupts after the execution of every instruction in the parameter set RAM. The software can monitor these interrupts to happen within the expected time widow to ensure correct program sequence during the chirping and inter-frame periods. For QSPI, program sequence monitoring (temporal and logical) are a software methods described in ISO 26262-5:2011 Annex D and IEC 61508-7 Annex A. The integrator/users software can monitor interrupts/completion events to happen within the expected time window to ensure correct program sequence. Examples : QSPI is not responding in a timely manner and boot did not complete on time.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.73 Temporal monitoring of other Bus masters

Host shall monitor data transfer interrupt line to ensure the data transfer operation by other masters like DMA is complete. Monitoring the other bus masters whether they have finished their tasks correctly within the defined time would detect the faults in the interconnect or the Bus masters.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.74 Time out mechanism for the SEQ EXT functionality

SEQ EXTN FSM writes the datapath registers and reads back the written values every chirp. This operation is expected to be completed within the IDLE time of a chirp. A timeout counter is programmed to ensure that the FSM operation is completed within the timeout period configured.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

6.3.75 Transmission Timeout at Chirp Boundary

The CBUFF IP will generate an error interrupt if the previous Chirp data transfer has not completed before the next DSS_CHIRP_AVIALABLE interrupt is received. This helps in Detection of erroneous incomplete or No transfer when the trigger for the next transfer is received that includes CSI2 or LVDS layers too.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.76 Transmission redundancy

The information is transferred several times in sequence and compared. If the same data path is used for the duplicate transmissions, then transmission redundancy will only by useful for detecting transient faults. Error response, diagnostic testability, and any necessary software requirements are defined by the system integrator.

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

6.3.77 XO/Slicer Frequency Detection

Monitor REFCLK oscillator/slicer output buffer with "XO/Slicer Frequency Detection" circuitry. XO/Slicer Frequency Detection circuitry compares a Slicer-clocked counter within a window determined by a counter clocked by the Reference Detection Oscillator (RC/ring oscillator).

See Section 5 for more information regarding the hardware component parts.

See the *Table of Contents* for Section 6.3 for more descriptions of functional safety mechanisms.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Chapter 7

An In-Context Look at This Safety Element out of Context (Recommended)

This section contains a Safety Element out of Context (SEooC) analysis of the AWR2188 component. Texas Instruments has made assumptions on the typical safety system configurations using this component. System level safety analysis is the responsibility of the developer of these systems and not Texas Instruments. As such, this section is intended to be informative only to help explain how to use the features of this component to assist the system designer in integrating this component into a system. This section describes example use cases and goes into more detail on how to identify hardware parts that are critical to the safety function and how to configure the associated functional safety mechanisms. Please note that the system designer may choose to use this component in safety-relevant systems beyond those mentioned below.

7.1 System Functional Safety Concept Examples (Recommended)

[System Functional Safety Concept Example]

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Appendix A

Summary of Recommended Functional Safety Mechanism Usage (Optional)

Table A-2 summarizes the functional safety mechanisms present in hardware or recommended for implementation in software or at the system level as described in Section 5 . Table A-1 describes each column in Table A-2 and gives examples of what content can appear in each cell.

Table A-1. Legend of Functional Safety Mechanisms

Functional Safety Mechanism	Description
TI Safety Mechanism Unique Identifier	A unique identifier assigned to this safety mechanism for easier tracking.
Safety Mechanism Name	The full name of this safety mechanism.
Safety Mechanism Category	**Safety Mechanism** - This test provides coverage for faults on the primary function. It may also provide coverage on another safety mechanism. **Test for Safety Mechanism** - This test provides coverage for faults of a safety mechanism only. It does not provide coverage on the primary function. **Fault Avoidance** - This is typically a feature used to improve the effectiveness of a related safety mechanism.
Safety Mechanism Type	Can be either hardware, software, a combination of both hardware and software, or system. See Section 6.2 for more details.
Safety Mechanism Operation Interval	The timing behavior of the safety mechanism with respect to the test interval defined for a functional safety requirement / functional safety goal. Can be either continuous, or on-demand. **Continuous** - the safety mechanism constantly monitors the hardware-under-test for a failure condition. **Periodic or On-Demand** - the safety mechanism is executed periodically, when demanded by the application. This includes Built-In Self-Tests that are executed one time per drive cycle or once every few hours.
Test Execution Time	Time period required for the safety mechanism to complete, not including error reporting time. Note: Certain parameters are not set until there is a concrete implementation in a specific component. When component specific information is required, the component data sheet should be referenced. Note: For software-driven tests, the majority contribution of the Test Execution Time is often software implementation-dependent.
Action on Detected Fault	The response that this safety mechanism takes when an error is detected. Note: For software-driven tests, the Action on Detected Fault may depend on software implementation.
Time to Report	Typical time required for safety mechanism to indicate a detected fault to the system. Note: For software-driven tests, the majority contribution of the Time to Report is often software implementation-dependent.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
ADCBUF1	SRAM ECC	Diagnostic	Hardware	Continuous	No Hardware Overhead	ESM Interrupt for Single/Multi bit error.	Hardware error reporting is typically < 1us.
ADCBUF3	PBIST check of memory	Test for Diagnostic	Hardware	Boot time	Typically <10us	Error Flag in the PBIST status Register	Hardware error reporting is typically < 1us.
ADCBUF3A	Software Test of PBIST	Test for Diagnostic	Software	Periodic / On-demand	Software Dependent	Software Defined	Software Dependent
ADCBUF4	Software Test of Memory ECC	Test for Diagnostic	Hardware/ Software	Periodic / On-Demand	Software Dependent	ESM Interrupt for Single/Multi bit error.	Typical <1us (Implementation Dependent)
ADCBUF5	PBIST Auto-coverage	Test for Diagnostic	Hardware	Boot time	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation
ADCBUF6	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic / On-Demand	Software Dependent Typically a single CPU clock per 32-bit Register Read	Software Defined	Software Dependent
ADCBUF7	Software readback of written configuration	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent
ADCBUF8	Software test of basic functionality	Test for Diagnostic	Software	Boot time	Software Dependent (Hardware overhead is typically negligible)	System Defined	Hardware error reporting is typically < 1us.
ANA18	Software test of basic functionality including error tests	Diagnostic	Software	Periodic / On-Demand	Software-Dependent	BSS shall Report to MSS through Mailbox	Software dependent
ANA19	Periodic Read Back of Configuration Registers	Diagnostic	Software	Periodic / On-Demand	Software-Dependent	BSS shall Report to MSS through Mailbox	Software dependent
APLL_SM5	APLL VCO Control Voltage Detection	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Typically 210us	BSS shall Report to MSS through Mailbox	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
CBUFF2	CBUFF Data FIFO ECC	Diagnostic					
CBUFF3	Periodic software test of static configuration registers	Diagnostic					
CBUFF4	Software readback of written configuration	Diagnostic					
CBUFF5	Transmission Timeout	Diagnostic					

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
CBUFF6	PBIST Check of CBUFF	Diagnostic					
CBUFF7	Software test of basic functionality	Diagnostic					
CBUFF8	PBIST auto coverage	Test-for-Diagnostic					
CBUFF9	Software test of memory ECC	Test-for-Diagnostic					
CLK10	Multi-bits keys on clock path mux selects, clock gating and divider options	Fault Avoidance	Hardware	Continuous	NA (Fault Avoidance)	NA (Fault Avoidance)	NA (Fault Avoidance)
CLK2/ APLL_SM4	PLL lock detection	Diagnostic	Hardware-Software	Continuous	<50us	Trigger to MSS ESM Group1 and Switches the device onto RCOSC Clock.	Hardware error reporting is typically < 1us.
CLK3	Dual Clock Comparator (DCC)	Diagnostic	Hardware-Software	Continuous / On Demand	Software Dependent	System Defined	Hardware error reporting is typically < 1us.
CLK4	External monitoring of OSC_CLKOU T and MCU_CLKOU T	Diagnostic	System	Typically Continuous	System defined	System defined	System defined
CLK5A	Internal watchdog - DWWD	Diagnostic	Hardware-Software	Continuous	Software Dependent	System Defined (BSS). Nerror for MSS/DSS.	Reset immediately or notifying CPU (interrupt) in typically < 1us after detection.
CLK5B	External watchdog	Diagnostic	Hardware-Software	Continuous	System defined	System defined (Typically Take Device to Reset)	System defined
CLK6	Periodic software readback of static clock configuration registers	Diagnostic	Software	Periodic (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS /DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
CLK7	Software readback of written configuration	Diagnostic	Software	After Every Register Write (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	System Defined	Hardware error reporting is typically < 1us.
CLK8	Software test of DCC operation	Test-for-Diagnostic	Software	Boot Time	Typically 100us	Reported to MSS /DSS for BSS. ESM Interrupts for MSS/DSS	Hardware error reporting is typically < 1us.
CLK9	Software test of DWWD operation	Test-for-Diagnostic	Software	Periodic/ On-Demand	Software dependent	Based on software	Software dependent
CM4.CPU1 B	Software diversified redundancy	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
CM4.CPU1C	Reciprocal comparison by software	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent
CM4.CPU1E	Program Sequence Monitoring	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent
CM4.CPU2	Boot Time LBIST of CPU core and Vectored Interrupt Controller	Diagnostic	Hardware/ Software	Continuous	No hardware overhead	Instruction Abort/Data Abort	1 CPU cycle after detection
CM4.CPU3	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic / On-Demand	Software Dependent Typically a single CPU clock per 32-bit Register Read	Software Defined	Software Dependent
CM4.CPU4	Software readback of written configuration	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent
CM4.CPU5	Periodic HW CRC check of SRAM	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent
CM4.CPU6	Illegal operation and instruction trapping	Diagnostic	Hardware	Continuous	No hardware overhead	Instruction Abort/Data Abort	1 CPU cycle after detection
CM4.CPU7	MPU	Diagnostic	Hardware/ Software	Continuous	No hardware overhead	Instruction Abort/Data Abort	1 CPU cycle after detection
CM4.RAM1	SRAM ECC	Diagnostic	Hardware	Continuous	No Hardware Overhead	ESM Interrupt for Single bit error. NERROR	Hardware error reporting is typically < 1us.
CM4.RAM2	Periodic PBIST Check of SRAM	Diagnostic	Hardware	Boot time (Executed as part of PBIST Memory test)	<5ms	NERROR	Error handling is Software defined
CM4.CPU-T1	LBIST Auto-coverage	Test for Diagnostic	Hardware	Periodic / On-Demand	NA	NA	NA
CM4.CPU-T2	Software test of illegal operation and instruction trapping including error tests	Test for Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent
CM4.CPU-T3	Software test of MPU functionality including error tests	Test for Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent
CM4.RAM-T1	Software Test of Memory Parity and ECC	Test for Diagnostic	Hardware/ Software	Periodic / On-Demand	Software Dependent	Error event asserted	Typical <1us (Implementation Dependent)
CM4.RAM-T2	Software Test of PBIST	Test for Diagnostic	Software	Periodic / On-demand	Software Dependent	Software Defined	Software Dependent
CM4.RAM-T3	PBIST Auto-Coverage	Test for Diagnostic	Hardware	Boot time	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
CPU1B	Diversified software redundancy	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Reported to MSS/DSS	Error handling is Software defined
CPU1C	Reciprocal comparison by software	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent
CPU1E	Program Sequence Monitoring	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Reported to MSS/DSS	Software Dependent
CPU2	Execution of LBIST	Diagnostic	Hardware/ Software	Boot time	No hardware overhead	Reported to MSS/DSS	1 CPU cycle after detection
CPU3	MPU	Diagnostic	Hardware/ Software	Continuous	No hardware overhead	Instruction Abort/Data Abort	1 CPU cycle after detection
CPU4	Online profiling using PMU-R4F	Diagnostic	Hardware/ Software	Continuous	No hardware overhead	Instruction Abort/Data Abort	1 CPU cycle after detection
CPU5	Illegal operation and instruction trapping for R4F	Diagnostic	Hardware	Continuous	No hardware overhead	Instruction Abort/Data Abort	1 CPU cycle after detection
CPU6	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic / On-Demand	Software Dependent Typically a single CPU clock per 32-bit Register Read	Reported to MSS/DSS	Software Dependent
CPU7	Software readback of written configuration	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Reported to MSS/DSS	Software Dependent
CPU9	LBIST Auto-coverage	Test for Diagnostic	Hardware	Periodic / On-Demand	NA	NA	NA
CSITX1	CSI Header ECC	Diagnostic					
CSITX2	CRC in message payload	Diagnostic					
CSITX3	Periodic software readback of static configuration registers	Diagnostic					
CSITX4	Software readback of written configuration	Diagnostic					
CSITX5	Data Underflow detection	Diagnostic					
CSITX6	PBIST of CSI SRAM	Diagnostic					
CSITX7	Software test of basic functionality including error tests	Diagnostic					
CSITX9	Parity Check on CSI FIFO	Diagnostic					
CSITX8	PBIST Auto Coverage	Test of Diagnostic					

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
DFE1	Boot Time Execution of LBIST	Diagnostic	Hardware	Boot time	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
DFE10	Software Test of Memory ECC	Test-for-Diagnostic	Software	Boot time	<10ms (Overall Boot time)	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
DFE12	DCC	Diagnostic	Hardware	Boot time / Periodic	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
DFE13	Software Test of DCC Operation	Test-for-Diagnostic	Software	Periodic	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
DFE18	LBIST Auto-coverage	Test-for-Diagnostic	Hardware	Boot time	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
DFE2	Periodic Execution of LBIST	Diagnostic	Hardware	Periodic (For BSS : Enabled by the TI's firmware)	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
DFE4	Parity test of function	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
DFE5	SRAM ECC	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
DFE6	Periodic Software Readback of Static Configuration Registers	Diagnostic	Software	Periodic (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
DFE7	Software Readback of Written Configuration	Diagnostic	Software	After Every Register Write (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
DFE8	Boot time PBIST check of DFE SRAM	Diagnostic	Hardware	Boot time (Executed as part of PBIST Memory test)	<200us	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
DFE9	PBIST Auto-coverage	Test-for-Diagnostic	Hardware	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation
EDM12	Software Test of Memory Parity	Test for Diagnostic	Software	Boot time	Software Dependent (Hardware overhead is typically negligible)	System Defined	Hardware error reporting is typically < 1us.
EDM15	PBIST Auto-coverage	Test for Diagnostic	Hardware	Boot time (Executed as part of PBIST Memory test)	Typically <10us	System Defined	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting
EDM16	EDMA Legal TR Check	Diagnostic	Hardware	Not Available	Typically <10us	System Defined	System Defined

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
EDM18	ECC on DMA TPTC FIFO Memory	Diagnostic	Hardware	Continuous	No Hardware Overhead	DSS ESM Interrupt for Single /Multi bit error	Hardware error reporting is typically < 1us.
EDM19	Test of ECC on DMA TPTC FIFO Memory	Test for Diagnostic	Hardware/ Software	Periodic / On-Demand	Software Dependent	DSS ESM Interrupt for Single /Multi bit error	Typical <1us (Implementation Dependent)
EDM20	Temporal and Logical monitoring	Diagnostic	Software	Periodic / On Demand	Software Dependent	System Defined	Software Dependent
EDM4	Memory Parity	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS ESM group 1 via DSS ESM group 1 and MAILBOX	Hardware error reporting is typically < 1us.
EDM5	PBIST check of memory	Test for Diagnostic	Hardware	Boot time	Typically <10us	Error Flag in the PBIST status Register	Hardware error reporting is typically < 1us.
EDM7	Software readback of written configuration	Diagnostic	Software	Periodic /On Demand	Software Dependant	Software Defined	Software Dependant
EDM8	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic	Software Dependent (Typically < 15 cycles of 200MHz clock for 32- bit register read)	System Defined	Hardware error reporting is typically < 1us.
EFU1	Autoload self- test	Test for Diagnostic	Hardware	Boot time	<10us	Boot ROM execution will not commence.	Hardware error reporting is typically < 1us.
EFU2	Efuse ECC	Test for Diagnostic	Hardware	Boot time	<10us	Boot ROM execution will not commence.	Hardware error reporting is typically < 1us.
EFU6	Efuse ECC logic self-test	Test for Diagnostic	Hardware	Boot time	<10ns	Boot ROM execution will not commence.	Hardware error reporting is typically < 1us.
EFU7	Clock Gating of EFUSE Controller	Fault avoidance	Hardware	Continuous	No Hardware Overhead	No impact of any inadvertent trigger	Not Reported
ESM1	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic/ On Demand	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
ESM2A	Boot time test of error path	Test for Diagnostic	Software	Boot time	<10us	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
ESM2B	Periodic test of error path	Diagnostic	Software	Periodic (trigerred by host)	<10us	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
ESM4	Software readback of written configuration	Diagnostic	Hardware	Continuous	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS	Hardware error reporting is typically < 1us.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
ESM6	Multi-bit Key for MMR Fields	NA (Fault Avoidance)	Hardware	Continuous	NA (Fault Avoidance)	NA (Fault Avoidance)	NA (Fault Avoidance)
EXT10	Lockstep compare self- test	Test-for-Diagnostic	Hardware-Software	Boot time	<10ms (Overall Boot time)	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
EXT11	Software Test of ECC Logic	Test-for-Diagnostic	Software	Boot time	Typically <10us	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
EXT2	Boot time PBIST of Chirp Configuration RAM, Chirp Profile RAM, Sequencer Extension RAM	Diagnostic	Hardware	Boot time (Executed as part of PBIST Memory test)	<12us	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
EXT3	ECC on Chirp Configuration RAM, Chirp profile RAM, Sequencer 4.5.2Extensio n RAM	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
EXT5	Software readback of written configuration	Diagnostic	Software	After Every Register Write (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	System Defined	Hardware error reporting is typically < 1us.
EXT6	Time out mechanism for the SEQ EXT functionality	Diagnostic	Software	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
EXT7	PBIST Auto-coverage	Test-for-Diagnostic	Hardware	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation
EXT8	Bit multiplexing in memory array	NA (Fault Avoidance)	Hardware	Continuous	NA (Fault Avoidance)	NA (Fault Avoidance)	NA (Fault Avoidance)
EXT9	Lockstep for Sequencer Extension FSM	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
FFT1	PBIST Check of FFT RAM	Diagnostic					
FFT2	Repeated test and sanity check	Diagnostic					
FFT3	Loopback pattern test	Diagnostic					
FFT5	PBIST auto coverage	Test-for-Diagnostic					
FFT6	Software readback of written configuration	Diagnostic					
FRC2	FRC Lockstep compare	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
FRC4	Logical FRC Counter monitoring at a system level	Diagnostic	Software	On Demand	System defined	System defined	System defined
FRC5	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
FRC6	Software readback of written configuration	Diagnostic	Software	After Every Register Write (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	System defined	Hardware error reporting is typically < 1us.
FRC8	Lockstep compare self- test	Test-for-Diagnostic	Hardware-Software	Boot time	<10ms (Overall Boot time)	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
GIO3	Periodic Software Readback of Static Configuration Registers	Diagnostic	Software	Periodic	Typically less than 20 CPU clock cycles for 32-bit register read	System Defined	Hardware error reporting is typically < 1us.
GIO4	Software Readback of Written Configuration	Diagnostic	Software	After Every Register Write	Typically less than 20 CPU clock cycles for 32-bit register read	System Defined	Hardware error reporting is typically < 1us.
GPADC1	PBIST check of GPADC RAMs	Test for Diagnostic	Hardware	Boot time (Executed as part of PBIST Memory test)	Typically <10us	Reported to MSS/DSS for BSS. Error Flag in the PBIST status Register for MSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
GPADC2	A repeated test and sanity checking with previous results test of function	Diagnostic	Software	System defined	System defined	System defined	System defined
GPADC5	PBIST Auto-coverage	Test for Diagnostic	Hardware	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation
GPADC6	Software readback of written configuration	Diagnostic	Software	After Every Register Write (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS for BSS.	Hardware error reporting is typically < 1us.
GPADC7	SRAM ECC on GPADC Data RAM	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS for BSS. ESM Error for MSS	Hardware error reporting is typically < 1us.
GPADC8	Software Test of ECC Logic	Test for Diagnostic	Software	Boot time	Typically <10us	Reported to MSS/DSS for BSS. ESM Error for MSS	Hardware error reporting is typically < 1us.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
GPADC9	Software Selftest of GPADC with known reference voltage values	Test for Diagnostic	Refer RF/ Analog Module section	Refer RF/ Analog Module section	Refer RF/ Analog Module section	Refer RF/ Analog Module section	Refer RF/Analog Module section
IIC2	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic / On Demand	Software Dependent. Typically <25 CPU clocks per 32-bit Register Read	Software Defined	Software Defined
IIC3	Software readback of written configuration	Diagnostic	Software	Periodic / On Demand	Software Dependent	Software Defined	Software Defined
IIC5	Information Redundancy	Diagnostic	Software	Periodic	System defined	System defined	System defined
INC1	Error trapping (including peripheral slave error trapping)	Diagnostic	Hardware	Continuous	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS for BSS. MSS/DSS errors need to be handled in application	Hardware error reporting is typically < 1us.
INC2	PCR access management	Diagnostic	Hardware	Periodic	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS for BSS.MSS/DSS errors need to be handled in application	Hardware error reporting is typically < 1us.
INC3	Information redundancy	Diagnostic	Software	Continous	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS for BSS.MSS/DSS errors need to be handled in application	Hardware error reporting is typically < 1us.
INC4	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic (For BSS : Enabled by the TI's Firmware	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS for BSS.MSS/DSS errors need to be handled in application	Hardware error reporting is typically < 1us.
INC5A	Boot time software test of basic functionality including error tests	Test for Diagnostic	Software	Boot time	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS for BSS.MSS/DSS errors need to be handled in application	Hardware error reporting is typically < 1us.
INC6	Software readback of written configuration	Diagnostic	Software	After Every Register Write (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS for BSS.MSS/DSS errors need to be handled in application	Hardware error reporting is typically < 1us.
INC8	Access timeout capability	Diagnostic	Hardware	Continuous	Typically less than 10 CPU clock cycles.	MSS/DSS errors need to be handled in application. In BSS this will lead to aborts	System defined

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
INC9	Hardware redundancy	Diagnostic	Hardware	Continuous	Typically less than 10 CPU clock cycles.	MSS/DSS errors need to be handled in application. In BSS this will lead to aborts	System defined
INC-T1	Software test of PCR access management	Test for Diagnostic	Software	On-Demand	Software Defined	System defined	Hardware error reporting is typically < 1us.
IOM1	Lock mechanism for control registers	NA (Fault Avoidance)	Hardware	Continuous	NA (Fault Avoidance)	NA (Fault Avoidance)	NA (Fault Avoidance)
IOM4	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic	Typically less than 20 CPU clock cycles for 32-bit register read	System Defined	Hardware error reporting is typically < 1us.
IOM6	Software readback of written configuration	Diagnostic	Software	After Every Register Write	Typically less than 20 CPU clock cycles for 32-bit register read	System Defined	Hardware error reporting is typically < 1us.
JTG1	Hardware disable of JTAG port	Fault Avoidance	Hardware	Continuous	Fault Avoidance	Fault Avoidance	Fault Avoidance
PM_SM01	RCOSC Test during bootup	Test for Diagnostic	Hardware-Software	Wakeup / Boot time	<10us	Trigger to MSS ESM Group1 and Switches the device onto RCOSC Clock.	Hardware error reporting is typically < 1us.
PM_SM03	Temperature Monitor	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Typically 200us	Reported to MSS R5F via SW Interrupt, MAILBOX	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
PM_SM11 / PM_SM12 / PM_SM13 / PM_SM14	External Voltage Monitor for (VIO, VDDIN, VIN_SRAM, VIOIN-18DIF F, VIN18_CLK, VIOIN_18)	Diagnostic	System	Continuous	System defined	System defined	System defined
PM_SM111 / PM_SM112 / PM_SM113 / PM_SM114	Internal Voltage Monitor using GPADC	Test of Diagnostic	Hardware	Periodic	1.6us (Glitch filter logic filters glitches < 1.6us)	Reported to MSS / DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
PM_SM21, PM_SM22, PM_SM23, PM_SM24, PM_SM25, PM_SM27	PM DCBIST (Internal VMON)	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Typically 2300us (TX + RX + PM)	Reported to MSS / DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
PM_SM40	Band-gap Reference Voltage Check	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Typically 200us	Reported to MSS / DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
PM_SM50	Temperature Sensor Consistency Check	Test-for-Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Typically 200us	Reported to MSS / DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
PM_SM61	GPADC Consistency Check	Test-for-Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Typically 50us	Reported to MSS / DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
PM_SM74	DCC Clock Frequency Monitor Fault Injection	Test-for-Diagnostic	Hardware-Software	System Dependent	Hardware works during Test chirp. Typically, 100us for post processing by CPU.	Async Event sent to Host with Error Status.	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
QSPI12	Temporal and Logical monitoring	Diagnostic	Software	Periodic/On-demand	Software Dependent	Software Defined	Software Dependent
QSPI2	CRC in Message	Diagnostic	Hardware	Continuous	No Overhead (Same clock cycle)	Error event asserted	Typical <1µs (Implementation Dependent)
RAM1	SRAM ECC	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
RAM10	Software Test of Parity	Test for Diagnostic	Hardware/ Software	Periodic / On-Demand	Software Dependent	Reported to MSS/DSS	Software Dependent
RAM11	PBIST Auto Coverage	Test for Diagnostic	Hardware	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation
RAM12	Software Test of Memory ECC	Test-for-Diagnostic	Software	Boot Time	Typically <10us	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
RAM2	Memory Parity	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
RAM5B	PBIST check of memory	Test for Diagnostic	Hardware	Boot Time (Executed as part of PBIST Memory Test)	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
REG1	Periodic Read Back of Configuration Registers	Diagnostic	Software	Periodic / On-Demand	Software-Dependent	BSS shall Report to MSS through Mailbox	Software dependent

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
REG3	Software Readback of Written Configuration	Diagnostic	Software	Periodic / On-Demand	Software Dependent	Software Defined	Software Dependent
RFANA_SM 01	RF loopback Test	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Depends on user programmin g, typically 1250us	Async Event sent to Host with Error Status. Reported as part RX gain phase monitor results.	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
RFANA_SM 04	RF Loopback Signal Consistency Check.	Test-for-Diagnostic	Hardware-Software	System Dependent	Depends on user programmin g, typically 1250us	Async Event sent to Host with Error Status. Reported as part RX gain phase monitor results.	Once in FTTI
RFANA_SM 05	IF Loopback Signal Consistency Check	Test-for-Diagnostic	Hardware-Software	System Dependent	Depends on user programmin g, typically 1000us	Async Event sent to Host with Error Status. Reported as part of IF Stage Monitor results.	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
ROM1	Memory ECC	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
ROM10	Software Test of Parity	Test-for-Diagnostic	Software	Periodic / On-Demand	Software Dependent	Reported to MSS/DSS	Software Dependent
ROM11	PBIST Auto-coverage	Test-for-Diagnostic	Hardware	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation
ROM12	Software Test of Memory ECC	Test-for-Diagnostic	Software	Boot time	<10ms (overall boot time)	Reported to MSS/DSS	Hardware error reporting is typically <1us
ROM2	Memory Parity	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
ROM5B	PBIST check of memory	Test for Diagnostic	Hardware	Boot time (Executed as part of PBIST Memory test)	N/A	Reported to MSS/DSS	Error handling is Software defined
ROM9	Software test of CRC	Test for Diagnostic	Software	Periodic / On-Demand	Software Dependent	Reported to MSS/DSS	Software Dependent
RST2	Software check of last reset	Fault Avoidance	Software	Boot time	Fault Avoidance	Fault Avoidance	Fault Avoidance
RST5	Multi-bits keys on reset generation control registers	Fault Avoidance	Hardware	Continuous	NA (Fault Avoidance)	NA (Fault Avoidance)	NA (Fault Avoidance)
RST6	Glitch filtering on reset pins	Fault Avoidance	Hardware	Continuous	2us (Glitch filter logic filters the glitches < 2us)	NA (Fault Avoidance)	NA (Fault Avoidance)

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
RST8	Periodic Software Readback of Static Configuration Registers	Diagnostic	Software	Periodic	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS R5F via SW Interrupt, MAILBOX	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
RST9	Software Readback of Written Configuration	Diagnostic	Software	After Every Register Write	Typically less than 20 CPU clock cycles for 32-bit register read	System Defined	Hardware error reporting is typically < 1us.
RTI1	1oo2 software voting using secondary free running counter	Test for Diagnostic	Software	Boot time	Typically <10us	System Defined for MSS/DSS. Reported to MSS/DSS for BSS	Hardware error reporting is typically < 1us.
RTI2	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	System Defined for MSS/DSS. Reported to MSS/DSS for BSS	Hardware error reporting is typically < 1us.
RTI3	Software readback of written configuration	Diagnostic	Software	After Every Register Write (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	System Defined for MSS/DSS. Reported to MSS/DSS for BSS	Hardware error reporting is typically < 1us.
RTI5	Software test of basic functionality including error tests	Test for Diagnostic	Software	Boot time / On Demand (periodically driven by Application)	Software Dependent	System Defined for MSS/DSS. Reported to MSS/DSS for BSS	System Defined
RX_SM02	IF loopback Test	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Depends on user programmin g, typically 1000us	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
RX_SM04	IF filter response Test	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Depends on user programmin g, typically 500us	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
RX_SM06	RX Saturation Detector	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Depends on user programmin g, typically 200us	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
RX_SM20	RX Gain and Phase Monitor Fault Injection	Test-for-Diagnostic	Hardware-Software	System Dependent	Depends on user programmin g, typically 1ms for gain, phases and noise.	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
RX_SM23	RX Saturation Detector Monitor	Test-for-Diagnostic	Hardware-Software	On Demand	Depends on user programmin g, typically 200us	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
SEQ2	Boot time PBIST of Chirp Configuration RAM, Chirp Profile RAM, Sequencer Extension RAM	Diagnostic	Hardware	Boot time (Executed as part of PBIST Memory test)	<700us	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
SEQ3	ECC on Chirp Configuration RAM, Chirp profile RAM, Sequencer 4.5.2Extensio n RAM	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
SEQ4	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
SEQ5	Software readback of written configuration	Diagnostic	Software	After Every Register Write (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
SEQ6	Lockstep Sequencer compare	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
SEQ7	Lockstep compare self- test	Test-for-Diagnostic	Hardware-Software	Boot time	<10ms (Overall Boot time)	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
SEQ8	PBIST Auto-coverage	Test-for-Diagnostic	Hardware	Boot time (Executed as part of PBIST Memory test)	<700us	Reported to MSS/DSS	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
SEQ9	Software Test of ECC Logic	Test-for-Diagnostic	Software	Boot time	Typically <10us	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
SEQ10	Parity on the Data Path flops	Diagnostic	Hardware	Continuous			
SPI2	Information Redundancy	Diagnostic					
SPI3	Periodic software readback of static configuration registers	Diagnostic					

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
SPI4	Software readback of written configuration	Diagnostic					
SPI5	Transmission redundancy	Diagnostic					
SPI6	Data overflow detection	Diagnostic					
SPI7	Data underflow detection	Diagnostic					
SYNC_20G_SM3	20GHz SYNCOUT signal power detector	Diagnostic					
SYNC_20G_SM4	20GHz SYNCIN signal power detector	Diagnostic					
SYNC_20G_SM9	Fault Injection in 20G Path to Detect Low Input/ Output Power	Test-for-Diagnostic					
SYNTH_SM20	SYNTH Frequency Error Monitor Fault Injection	Test-for-Diagnostic	Hardware-Software	System Dependent	Hardware works during Test chirp. Typically, 100us for post processing by CPU.	Async Event sent to Host as per Reporting Mode Configuratio n	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
SYNTH_SM3	SYNTH VCO Clock Frequency Linearity Detection Using TDC	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. during dummy chirp)	Depends on user programmin g, typically 200us	Async Event sent to Host as per Reporting Mode Configuratio n	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
SYNTH_SM5	SYNTH VCO Control Voltage Detection	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Typically 210us	Async Event sent to Host as per Reporting Mode Configuratio n	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
SYS1/ APP_SM1							
TX_SM1	Impedance Detector at PA Pin	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Depends on user programmin g, typically 300us for TX Ball Break monitor execution.	Async Event sent to Host as per Reporting Mode Configuratio n	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
TX_SM10	PA Output Power Detector	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Depends on user programmin g, typically 300us for TX Ball Break monitor execution.	Async Event sent to Host as per Reporting Mode Configuratio n	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
TX_SM2	PA Loopback Gain/Phase mismatch test	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Depends on user programmin g, typically 300us for TX Ball Break monitor execution.	Async Event sent to Host as per Reporting Mode Configuratio n	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
TX_SM21	TX Gain and Phase Monitor Fault Injection	Test-for-Diagnostic	Hardware-Software	System Dependent	Hardware works during Test chirp. Typically, 100us for post processing by CPU.	Async Event sent to Host as per Reporting Mode Configuratio n	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
TX_SM7	TX Phase shifter DAC monitor and fault injection	Diagnostic	Hardware-Software	Periodic (Enabled by TI's BSS firmware. Once per configured FTTI during Inter frame period)	Typically 200us	Async Event sent to Host as per Reporting Mode Configuratio n	Hardware error reporting is typically < 1us. Additional delay due to BSS Mailbox reporting*
VIM1	VIM SRAM Data ECC with DED Vector	Diagnostic	Hardware	Continuous	No Hardware Overhead	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
VIM2	Boot time PBIST check of VIM SRAM	Diagnostic	Hardware	Boot time	<15us	To be handled in application	Error handling is Software defined
VIM6	Periodic software readback of static configuration registers	Diagnostic	Software	Periodic (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
VIM7	Software readback of written configuration	Diagnostic	Software	After Every Register Write (For BSS : Enabled by the TI's firmware)	Typically less than 20 CPU clock cycles for 32-bit register read	Reported to MSS/DSS	Hardware error reporting is typically < 1us.
VIM8	Software Test of Memory ECC	Test-for-Diagnostic	Software	Boot time	Typically <10us	Error flag asserted for MSS. For BSS reported to MSS/DSS	Hardware error reporting is typically < 1us.
VIM9	Software Test of PBIST	Test-for-Diagnostic	Software	On Demand	Typically <10us	Error flag in the PBIST status register	Hardware error reporting is typically < 1us.
VIM10	PBIST Auto-coverage	Test-for-Diagnostic	Hardware	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation	N/A; implicit in PBIST operation
VIM11	Boot Time LBIST of CPU core and Vectored Interrupt Controller	Diagnostic	Hardware/ Software	Periodic / On-Demand	< 800 us	Error flag asserted for MSS. For BSS reported to MSS/DSS	Typically < 3us (implementation dependent)
XOSC_SM2	XO/Slicer Frequency Detection	Diagnostic	Hardware-Software	Continuous / On Demand	Software dependent	System Defined	Software defined

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

See Section 5 and Section 6.3 for more details on the individual safety mechanisms.

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Appendix B
Distributed Developments

A Development Interface Agreement (DIA) is intended to capture the agreement between two parties towards the management of each party's responsibilities related to the development of a functional safety system. TI functional safety components are typically designed for many different systems and are considered to be Safety Elements out of Context (SEooC) hardware components. The system integrator is then responsible for taking the information provided in the hardware component safety manual, safety analysis report and safety report to perform system integration activities. Because there is no distribution of development activities, TI does not accept DIAs with system integrators.

TI functional safety components are products that TI represents, promotes or markets as helping customers mitigate functional safety related risks in an end application and/or as compliant with an industry functional safety standard or FS-QM. For more information about TI functional safety components, go to TI Functional Safety.

B.1 How the Functional Safety Lifecycle Applies to TI Functional Safety Products

TI has tailored the functional safety lifecycles of ISO 26262 and IEC 61508 to best match the needs of a functional Safety Element out of Context (SEooC) development. The functional safety standards are written in the context of the functional safety systems, which means that some requirements only apply at the system level. Since TI functional safety components are hardware or software components, TI has tailored the functional safety activities to create new product development processes for hardware and for software that makes sure state-of-the-art techniques and measures are applied as appropriate. These new product development processes have been certified by third-party functional safety experts. To find these certifications, go to TI Functional Safety.

B.2 Activities Performed by Texas Instruments

The TI functional safety products are hardware components developed as functional Safety Elements out of Context. As such, TI's functional safety activities focus on those related to management of functional safety around hardware component development. System level architecture, design, and functional safety analysis are not within the scope of TI activities and are the responsibility of the customer. Some techniques for integrating the SEooC safety analysis of this hardware component into the system level can be found in ISO 26262-11.

Table B-1. Activities Performed by Texas Instruments versus Performed by the customer

Functional Safety Lifecycle Activity[1]	TI Execution	Customer Execution
Management of functional safety	Yes	Yes
Definition of end equipment and item	No	Yes
Hazard analysis and risk assessment (of end equipment/ item)	No	Yes
Creation of end equipment functional safety concept	No. Assumptions made for internal development.	Yes
Allocation of end equipment requirements to sub-systems, hardware components, and software components	No. Assumptions made for internal development.	Yes

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Table B-1. Activities Performed by Texas Instruments versus Performed by the customer (continued)

Functional Safety Lifecycle Activity[1]	TI Execution	Customer Execution
Definition of hardware component safety requirements	Yes	No
Hardware component architecture and design execution	Yes	No
Hardware component functional safety analysis	Yes	No
Hardware component verification and validation (V&V)	V&V executed to support internal development.	Yes
Integration of hardware component into end equipment	No	Yes
Verification of IC performance in end equipment	No	Yes
Selection of safety mechanisms to be applied to IC	No	Yes
End equipment level verification and validation	No	Yes
End equipment level functional safety analysis	No	Yes
End equipment level functional safety assessment	No	Yes
End equipment release to production	No	Yes
Management of functional safety issues in production	Support provided as needed	Yes

(1) For component technical questions, ask our TI E2E™ support experts.

B.3 Information Provided

Texas instruments has summarized what it considers the most critical functional safety work products that are available to the customer either publicly or under a nondisclosure agreement (NDA). NDAs are required to protect proprietary and sensitive information disclosed in certain functional safety documents.

Table B-2. Product Functional Safety Documentation

Deliverable Name	Contents
Functional Safety Product Preview	Overview of functional safety considerations in product development and product architecture. Delivered ahead of public product announcement.
Functional Safety Manual	User guide for the functional safety features of the product, including system level assumptions of use.
Functional Safety Analysis Report	Results of all available functional safety analysis documented in a format that allows computation of custom metrics.
Functional Safety Report[1]	Summary of arguments and evidence of compliance to functional safety standards. References a specific component, component family, or TI process that was analyzed.
Assessment Certificate[1]	Evidence of compliance to functional safety standards. References a specific component, component family, or TI process that was analyzed. Provided by a 3rd party functional safety assessor.

(1) When an Assessment Certificate is available for a TI functional safety product, the Functional Safety Report may not be provided. When a Functional Safety Report is provided, an Assessment Certificate may not be available. These two documents fulfill the same functional safety requirements and will be used interchangeably depending on the TI functional safety product.

DRAFT ONLY
TI Confidential - NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATE	REVISION	NOTES
August 2025	*	Initial Release

DRAFT ONLY
TI Confidential – NDA Restrictions

Copyright © 2025 Texas Instruments Incorporated

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 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. IMPORTANT NOTICE

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2025, Texas Instruments Incorporated

DRAFT ONLY
TI Confidential – NDA Restrictions