MPC5746R Functional Safety Library TRS

|  |  |
| --- | --- |
| Version | 0.1 |
| Date | Jan 06, 2019 |
| Status |  |

Table of Contents

[1 Introduction 1](#_Toc536050985)

[1.1 Purpose 1](#_Toc536050986)

[1.2 Scope 1](#_Toc536050987)

[1.3 Definitions and Acronyms 1](#_Toc536050988)

[1.4 References 2](#_Toc536050989)

[2 General Description 3](#_Toc536050990)

[2.1 Target 3](#_Toc536050991)

[2.2 System Topology 3](#_Toc536050992)

[2.3 Assumptions 3](#_Toc536050993)

[2.4 Release Scope 4](#_Toc536050994)

[3 System Requirements 5](#_Toc536050995)

[3.1 FCCU Module Requirements 5](#_Toc536050996)

[3.2 RGM Module Requirements 5](#_Toc536050997)

[3.3 STCU Module Requirements 6](#_Toc536050998)

[3.4 TSENS Module Requirements 6](#_Toc536050999)

[3.5 SWT Module Requirements 6](#_Toc536051000)

[3.6 CRC Module Requirements 6](#_Toc536051001)

[3.7 OSC Module Requirements 7](#_Toc536051002)

[3.8 PLLDIG Module Requirements 7](#_Toc536051003)

[3.9 CMU Module Requirements 7](#_Toc536051004)

[3.10 MC\_ME Module Requirements 8](#_Toc536051005)

[3.11 PMC Module Requirements 8](#_Toc536051006)

[3.12 MPU Module Requirements 8](#_Toc536051007)

[3.13 PBRIDGE Protection Requirements 8](#_Toc536051008)

[3.14 BIST Module Requirements 9](#_Toc536051009)

[3.15 PBIST Module Requirements 9](#_Toc536051010)

[3.16 INTC Module Requirements 9](#_Toc536051011)

[3.17 EDMA Module Requirements 9](#_Toc536051012)

[3.18 STM Module Requirements 10](#_Toc536051013)

[3.19 PIT Module Requirements 10](#_Toc536051014)

[3.20 SSCM Module Requirements 10](#_Toc536051015)

[3.21 MEMU Module Requirements 10](#_Toc536051016)

[3.22 FLASH Module Requirements 11](#_Toc536051017)

[3.23 BCTU Module Requirements 11](#_Toc536051018)

[3.24 REG\_PROT Module Requirements 11](#_Toc536051019)

[3.25 WKPU Module Requirements 11](#_Toc536051020)

[3.26 XBAR Module Requirements 12](#_Toc536051021)

[3.27 ADC Module Requirements 12](#_Toc536051022)

[Appendix A. Version Tracking 13](#_Toc536051023)

# Introduction

## Purpose

This document describes the technical requirements for Functional Safety Library on NXP MPC5746R. The Technical Requirements are used to generate the functional, architecture and design requirements. It’s intended for product development engineer and business management.

## Scope

These technical requirements cover the inputs refined from marketing, sales and CAS team on a functional safety practice, safety library requirements on AMP PowerPC platform. The descriptions here are for MPC5746R but intended to be technology independent so that the details can be reused in future programs.

## Definitions and Acronyms

|  |  |
| --- | --- |
| C2N | Configuration State to Normal State |
| CF | Critical Fault |
| EOUT | Error Out |
| FCCU | Fault Collection and Control Unit |
| FOSU | FCCU Output Supervision Unit |
| FSM | Finite State Machine |
| INTC | Interrupt Controller |
| IRQ | Interrupt Request |
| MC\_ME | Mode Entry Module |
| N2C | Normal State to Configuration State |
| NCF | Non-Critical Fault |
| NFFS | Non-Critical Fake Fault Source |
| NMI | Non-Maskable Interrupt |
| STCU | Self-test Control Unit |
| TRS | Technical Requirement Specification |
| WKPU | Wake-up Unit |
| SSCM | System Status and Control Module |
| MC\_ME | Mode Entry |
| PMC | Power Management Controller |
| MPU | Memory Protection Units |
| PIT | Periodic Interrupt Timer |
| CMU | Clock Monitor Unit |
| STM | System Timer Module |
| eDMA | Enhanced Direct Memory Access |
| INTC | Interrupt Controller |
| PBIST | Built-In Self-Test |
| BIST | Peripheral Built-In Self-Test |
| NA | Non-initial check and Non-runtime check |

## References

MPC5746R Reference Manual

MPC5746R Safety Manual

# General Description

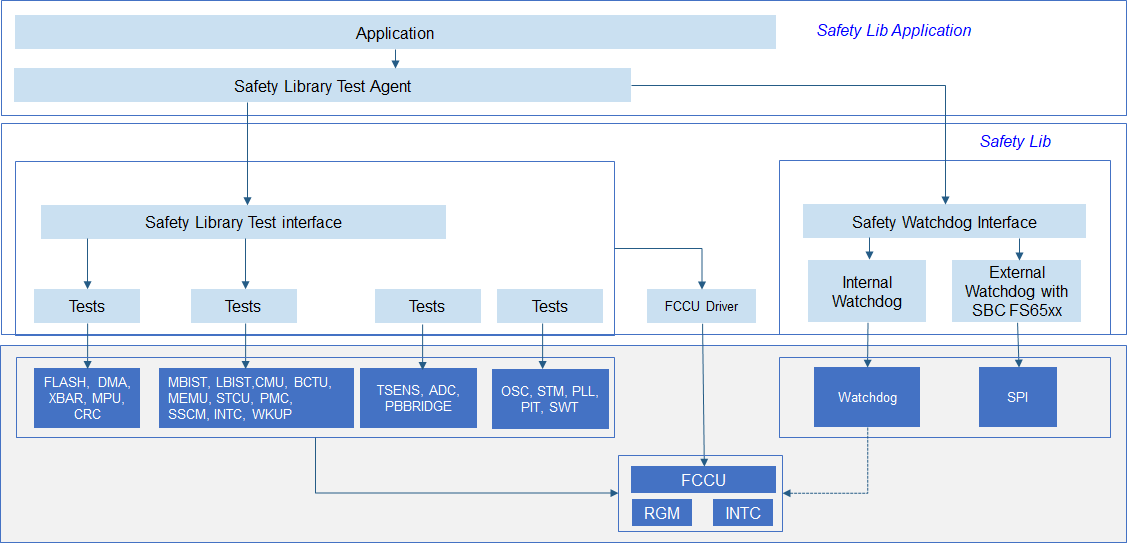
## Target

Functional Safety Library is an NXP direction to meet increasing functional safety requirement in automotive industry. Its primary objective is to provide a safety SDK to facilitate the automotive application designer and developer to take use of safety mechanism supported by NXP product.

## System Topology

Functional Safety Library on MPC5746R is a software component above SDK driver, which bases on the safety mechanism supported by MPC5746R. It provides a series of test, monitor and react functionalities related to the error detection, report and simulation of a hardware safety mechanism.

FCCU plays a critical role in safety library architecture, almost all hardware safety mechanisms have dependencies on FCCU under the path of error reaction and report. The figure below shows the architecture of safety library and typical MCU SDK components.



*Figure 1. Functional Safety Library Component*

## Assumptions

**[FS\_AS\_001]** MCU base system including RAM section, BIST, shall be done before invoking any Functional Safety Library APIs.

**[FS\_AS\_002]** Functional Safety Library shall be executed on safety core, normally the safety core is lockstep core.

**[FS\_AS\_003]** The application shall NOT access the hardware functionality being tested during Functional Safety Library access cycle.

**[FS\_AS\_004]** The application shall handle all exceptions reported from Functional Safety Library test execution.

## Release Scope

The release addresses all requirements in this document. It includes the items below:

* Functional Safety Library SDK
* Functional Safety Library demo application
* Functional Safety Library SDK User Manual

# System Requirements

## FCCU Module Requirements

**[TR-001]** The FSLIB shall support the fault collection with FCCU module.

**NOTE**: SM\_052 NA. For faults detection.

**[TR-002]** The FSLIB shall support fault control with the FCCU module.

**NOTE**: SM\_052 NA. For the fault reaction configuration.

**[TR-003]** The FSLIB shall support to configure ERROR[0] to the error out pin if MCU is used in a single error out pin mode.

**NOTE**: SM\_294 NA.

**[TR-004]** The FSLIB shall support to verify fault reaction has been configured for a specified fault.

**NOTE**: SM\_053, initial check.

**[TR-005]** The FSLIB shall support to force system I/O to a high impedance.

**NOTE**: SM\_054, initial check.

**[TR-006]** The FSLIB shall support to register a response if the MCU signals an internal failure via its error out signals.

**NOTE**: SM\_166, initial check. Fault response, such as disable or reset MCU.

**[TR-007]** The FSLIB shall support to check the status of system is continuously switching mode.

**NOTE**: SM\_055, runtime check. switch between a standard operating state and reset, or fault state.

**[TR-008]** The FSLIB shall support to store functional reset counter for a specified NCF.

**NOTE**: SM\_148, runtime check. If the functional reset counter is too many, safety software might be causing a destructive reset.

## RGM Module Requirements

**[TR-009]** The FSLIB shall support to detect a permanent cycling between a safe state and an unsafe state.

**NOTE**: SM\_057, initial check.

**[TR-010]** The FSLIB shall support to program value to generate a destructive reset if the number of functional resets reaches this value.

**NOTE**: SM\_058, initial check.

**[TR-011]** The FSLIB shall support to clean the functional reset counter.

**NOTE**: SM\_059, initial check. Will be called after finish TR-010.

**[TR-012]** The FSLIB shall support to enable a reset counter.

**NOTE**: SM\_060, initial check. By program a non-zero value to MC\_RGM\_FRET register.

## STCU Module Requirements

**[TR-013]** The FSLIB shall support to verify the functional safety integrity of STCU2.

**NOTE**: SM\_062, initial check. Integrity include detect random faults, latent fault detection, and single-point fault detection.

## TSENS Module Requirements

**[TR-014]** The FSLIB shall support to test substrate temperature with two temperature sensors.

**NOTE**: SM\_063, NA.

**[TR-015]** The FSLIB shall support to configure the react to over-temperature faults with FCCU and PMC module.

**NOTE**: SM\_064, initial check.

**[TR-016]** The FSLIB shall support to verify the conversion values are similar between two temperature sensors.

**NOTE**: SM\_066, initial check.

## SWT Module Requirements

**[TR-017]** The FSLIB shall support to test functional reset or interrupt then reset of the SWT module.

**NOTE**: SM\_161, NA.

**[TR-018]** The FSLIB shall support to test the trigger mechanism of FCCU fault from SWT no service sequence.

**NOTE**: SM\_162, NA.

**[TR-019]** The FSLIB shall support to verify the correctness of SWT initialization.

**NOTE**: SM\_067, NA. need to enable SWT and hard-lock configuration register.

**[TR-020]** The FSLIB shall support to detect lost and significantly slow clocks with SWT modules.

**NOTE**: SM\_068, NA.

**[TR-021]** The FSLIB shall support to insert control flow checkpoints with SWT modules.

**NOTE**: SM\_069/SM\_163/SM\_164, NA. Control flow include a fix service and a specified key service.

## CRC Module Requirements

**[TR-022]** The FSLIB shall support to verify the content of the safety-relevant configuration registers with CRC.

**NOTE**: SM\_070, runtime check. To verify by executing CRC calculation.

**[TR-023]** The FSLIB shall support to check correct initialization of the MCU.

**NOTE**: SM\_071, runtime check. Check configuration register with CRC calculation.

## OSC Module Requirements

**[TR-024]** The FSLIB shall support to test accuracy of the IRCOSC.

**NOTE**: SM\_073, initial check. Measure and compare with the expected frequency.

**[TR-025]** The FSLIB shall support to detect failure of the IRCOSC.

**NOTE**: SM\_074, runtime check. utilize the CMU's frequency meter to read the IRCOSC frequency and compare it against the expected value of 16 MHz.

**[TR-026]** The FSLIB shall support to check the status of the FlexCAN clocked by XOSC.

**NOTE**: SM\_075, initial check. FlexCAN should not be clocked directly by the XOSC in normal operation unless the effects of clock glitches are sufficiently detected by the applied FTCOM layer.

**[TR-027]** The FSLIB shall support test running status of the system clock sourced by XOSC.

**NOTE**: SM\_076, runtime check. Make sure system clock is available sourced by XOSC.

## PLLDIG Module Requirements

**[TR-028]** The FSLIB shall support to configure and generate a high-quality clock with dual PLL module.

**NOTE**: SM\_078, initial check.

**[TR-029]** The FSLIB shall support to switch to another high-quality clock if clock glitches endanger safety goal.

**NOTE**: SM\_079, initial check. High quality clock such as a FMPLL or external clock.

## CMU Module Requirements

**[TR-101]** The FSLIB shall support the Clock Monitor mechanism of CMU. It can be configured to check the clock monitors of the clocks listed below and signal an error to the FCCU in case the monitored clock is out of a programmable bounds or loss of reference clock.

CMU monitored the clock list:

|  |  |
| --- | --- |
| **CMU** | **Monitored Clock** |
| CMU\_FXBAR | Fast Crossbar |
| CMU\_SXBAR | Slow Crossbar |
| CMU\_AIPS | Peripheral Bus |
| CMU\_PER | Peripheral Clock |
| CMU\_ADCSD | Sigma Delta ADC |
| CMU\_SARADC | SAR ADC |
| CMU\_SENT | SENT Module |
| CMU\_EMIOS | eMIOS Module |
| CMU\_ETPU2 | ETPU2 Module |
| CMU\_CLKOUT | CLKOUT Signal |
| CMU\_PLL | PLL Module |

**NOTE:** SM\_298/SM\_156/SM\_157/SM\_080/SM\_081, Initial check. The CMUs should be used for each clock that is being monitored and used by a functional safety-relevant module. Application software shall check that the CMUs are enabled, and their faults managed by the FCCU.

## MC\_ME Module Requirements

**[TR-102]** The FSLIB shall support to monitor the duration of LP mode. If the system does not wakeup within a specified period, the system will be reset by the monitoring circuitry.

**NOTE:** SM\_082, NA. If the application uses the LP mode, this test should be used as a mechanism to overcome faults in the wake-up and interrupt inputs to the MC\_ME.

## PMC Module Requirements

**[TR-103]** The FSLIB shall support to 1) initiate the hardware-assisted self-test to detect LVD/HVD failures after startup 2) check the status registers of the FCCU and MC\_RGM for the results of the hardware-assisted self-test.

**NOTE:** SM\_144/SM\_084/SM\_204/SM\_085/SM\_089, NA. If the application uses the LP mode, this test should be used as a mechanism to overcome faults in the wake-up and interrupt inputs to the MC\_ME.

## MPU Module Requirements

**[TR-104]** The FSLIB shall support to use the CMPU to protect all memory regions that require protection against accesses from other applications.

**NOTE:** SM\_092, NA.

**[TR-105]** The FSLIB shall support to prevent write accesses to the SMPU's registers from all other masters except the safety core.

**NOTE:** SM\_093/SM\_094, initial check.

**[TR-106]** The FSLIB shall support to use MPUs to ensure that only authorized software routines can configure modules and all other bus masters can access only their allocated resources according to their access rights

**NOTE:** SM\_095, initial check.

## PBRIDGE Protection Requirements

**[TR-107]** The FSLIB shall support to configure the PBRIDGEs to define the access permissions for each slave module that requires access protection. Also, should configure the PBRIDGE to prevent write accesses to the MC\_RGM address space for all masters except the core.

**NOTE:** chapter 5.2.15, initial check.

**[TR-108]** The FSLIB shall support to periodically check the contents of configuration registers (more than 10 registers) of modules attached to the PBRIDGEs to help detect faults in the PBRIDGE.

**NOTE:** chapter 5.2.15, runtime check.

## BIST Module Requirements

**[TR-109]** The FSLIB shall support to confirm all LBISTs and MBISTs finished and check the BIST results.

**NOTE:** SM\_096/SM\_097, initial check.

**[TR-110]** The FSLIB shall support the flash memory array integrity self-check (with Flash memory margin read enabled) and check the results.

**NOTE:** chapter 5.2.16.3, initial check.

## PBIST Module Requirements

**[TR-111]** The FSLIB shall support the ADC BISTs run during initialization (during boot) and optionally during normal operation (software actions are required run those tests).

**NOTE:** chapter 5.2.16.5, initial check and optionally during normal operation.

## INTC Module Requirements

**[TR-112]** The FSLIB shall support to detect the critical failure modes of the INTC for all interrupts not supervised by the high priority interrupt monitor.

**NOTE:** SM\_098, runtime check. According to application software and interrupts.

**[TR-113]** The FSLIB shall support to ensure the call period of Periodic low latency IRQs is expected by using a running timer/counter.

**NOTE:** SM\_099, runtime check. According to application software and interrupts.

**[TR-114]** The FSLIB shall support to manage spurious or missing interrupt requests.

**NOTE:** SM\_100, runtime check.

## EDMA Module Requirements

**[TR-115]** The FSLIB shall support to supervise spurious, too often, or constant activation of the eDMA.

**NOTE:** SM\_101, runtime checks. Prevent the DMA from stealing transfer bandwidth on the XBAR, as well as prevent it from copying data at a wrong point in time.

**[TR-116]** The FSLIB shall support to prevent applications (that are not resilient to spurious, or missing functional safety-relevant) to raise eDMA requests that use the PIT module to trigger functional safety-relevant eDMA transfer requests.

**NOTE:** SM\_102, runtime check. To reduce the likelihood of a faulty PIT (which is not redundant) from triggering an unexpected eDMA transfer

**[TR-117]** The FSLIB shall support to protect peripheral lake eDMA transfers by using peripherals in both peripheral lakes or use other detection mechanisms.

**NOTE:** SM\_103, runtime check. To reduce the likelihood of a faulty PIT (which is not redundant) from triggering an unexpected eDMA transfer

**[TR-118]** The FSLIB shall support to protect safety-relevant software using the eDMA to transfer data to a non-replicated peripheral or within the RAM by: 1) Preferably, "always on" channels of the eDMA Channel Mux shall not be used. OR 2) ensure that the eDMA transfer was triggered as expected at the correct rate and the correct number of times. This test shall detect unexpected, spurious interrupts.

**NOTE:** SM\_104/SM\_90/SM\_91, runtime check. To reduce the likelihood of a faulty PIT (which is not redundant) from triggering an unexpected eDMA transfer

## STM Module Requirements

**[TR-119]** The FSLIB shall support to detect a possible functional safety-relevant failure of STM module by the Software Watchdog Timer (SWT). The FSLIB can check whether the interrupt time is consistent with the STM setting.

**NOTE:** SM\_105/SM\_106, runtime check. To protect STM IRQ handler by SWT.

## PIT Module Requirements

**[TR-120]** The FSLIB shall support to detect a possible functional safety-relevant failure of PIT module by the Software Watchdog Timer (SWT). Also, the FSLIB should check the checksum of its configuration registers using the CRC and compared with the expected one to verify that the PIT configuration is correct.

**NOTE:** SM\_107/SM\_108, runtime check. To catch possible PIT failures and check that the PIT remains at its expected configuration

## SSCM Module Requirements

**[TR-121]** The FSLIB shall support to 1) inhibit any execution of the BAF code (or part of it), or 2) provide appropriate measures to validate the safety of executing the BAF code.

**NOTE:** SM\_141, initial check. It is required that application software checks the configuration of the SSCM once after boot.Since BAF code was neither developed nor qualified according to the ISO26262-6, any execution of the BAF, or part of it, needs to be inhibited or validated by appropriate measures.

## MEMU Module Requirements

**[TR-201]** The FSLIB shall support to detect single-bit errors and remaining multi-bit errors after MBIST execution.

**NOTE**: [SM\_109], initial check.

**[TR-202]** The FSLIB shall support to detect permanent multi-bit error sources which are recorded in a new RAM ECC error report and are caused by multiple address selections within the FTTI.

**NOTE**: [SM\_110], runtime check.

## FLASH Module Requirements

**[TR-203]** The FSLIB shall support to validate the content of the array.

**NOTE**: [SM\_112], NA. by array Integrity self-check calculates a MISR signature over the

array content.

**[TR-204]** The FSLIB shall support to use checks to detect incorrect data returned from the EEPROM emulation.

**NOTE**: [SM\_114], NA. what is checks????

**[TR-205]** The FSLIB shall support to test potential multi-bit errors introduced by permanent failures in the flash memory control logic.

**NOTE**: [SM\_116] , runtime check.

**[TR-206]** The FSLIB shall support to verify the correctness of any write operation to both the flash memory and the overlay.

**NOTE**: [SM\_117], runtime check.

**[TR-207]** The FSLIB shall support to check the flash memory ECC fail reporting path if detected ECC faults are correctly reported.

**NOTE**: [SM\_119], runtime check.

## BCTU Module Requirements

**[TR-208]** The FSLIB shall support to configure BCTU to output triggers are generated within the desired time schedule with respect to the input event.

**NOTE**: SM\_120, runtime check.

## REG\_PROT Module Requirements

**[TR-209]** The FSLIB shall support register protection with a hard lock.

**NOTE**: [SM\_125], runtime check.

**[TR-210]** The FSLIB shall support register protection with software lock.

**NOTE**: [SM\_285], runtime check

## WKPU Module Requirements

**[TR-211]** The FSLIB shall support to detect latent faults of WKPU and NMI.

**NOTE**: [SM\_126], NA.

## XBAR Module Requirements

**[TR-212]** The FSLIB shall support verify the priority of masters of the XBAR which are not safety-related modules is lower than safety-relevant masters.

**NOTE**: SM\_127, NA.

**[TR-213]** The FSLIB shall support to check the XBAR configuration once after programming.

**NOTE**: SM\_111, runtime check. Need more notes.

**[TR-214]** The FSLIB shall support to detect failures of the XBAR configuration affecting system performance.

**NOTE**: SM\_128, runtime check. By using the configuration readback and SWT monitoring described above.

## ADC Module Requirements

**[TR-215]** The FSLIB shall support to implement functional safety integrity measures of the ADC module.

**NOTE**: SM\_130, initial check.

**[TR-216]** The FSLIB shall support to copy the reference thresholds from "Test flash memory" into the watchdog registers (STAWnR) with system integrator.

**NOTE**: SM\_131, initial check. Should do this before running the ADC hardware self-test.

**[TR-217]** The FSLIB shall support to enable the analog watchdog for CPU and CTU modes with timeout is smaller than FTTI.

**NOTE**: SM\_132, initial check. For integrated self-test.

1. Version Tracking

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Comments** | **Author** |
| Jan 6, 2019 | 0.1 | Initial draft | Eugene Zhang |