AN4287
Application note

Safety application guide for SPC564Axx/RPC564Axx family

Introduction

This document contains guidelines on how to configure and use the SPC564A7x/ SPC564A80/RPC564A80 device for safety relevant applications.

Contents

List of tables

List of figures

1 Preface

This document contains the guidelines on how to configure and use the SPC564A7x/SPC564A80/RPC564A80 device for safety relevant applications. These guidelines are preceded by one of the following bold text statements:

- Suggested;
- Implementation hint;
- Rationale.

These guidelines are considered to be useful approaches for the specific topics under discussion, but are not mandatory. The user needs to use discretion in deciding whether these measures are appropriate for their applications.

This document is valid only under the assumption that the MCU is used in automotive applications for use cases requiring a fail-silent or a fail-indicate MCU and if the environmental conditions specified in the SPC564A7x/SPC564A80/RPC564A80 device datasheet are maintained.

Together with the standard documentation as the reference manual and the datasheet, also SPC564A7x/SPC564A80/RPC564A80 device errata must be taken into account during system design and implementation.

2 General information

2.1 Mission profile

The assumed mission profile is:

- Lifetime: 20 years;
- Total operating hours: 12000 hours;
- Trip time[a]: 10 hours;
- Process Safety Time[b]: 10 ms.

2.2 Safe state

By definition, the Safe states of the SPC564A7x/SPC564A80/RPC564A80 are as follows:

- Completely unpowered;
- Reset;
- Operating correctly;
- Explicitly indicating an internal error.

If the SPC564A7x/SPC564A80/RPC564A80 signals an internal failure, the surrounding subsystem shall no longer use the SPC564A7x/SPC564A80/RPC564A80 outputs for safety functions since these signals are no longer considered reliable. If an error is indicated, the system must be able to remain in a Safe state without any additional actions. Depending on its configuration, the system may disable, or reset, the SPC564A7x/SPC564A80/RPC564A80 as a reaction to the error signal.

Suggested: the system must must bringthe system itself to a safe state when an error is indicated.

2.3 Failure indication time

SPC564A7x/SPC564A80/RPC564A80 failure indication time must be taken into consideration when determining application safety strategies, because it must be less than the FTTI.

2.4 Error handling

Error handling can be split into two categories:

- Handling of errors during run-time;
- Handling of errors during boot-time.

Suggested: run-time failures shall be handled in a time shorter than the FTTI.

Suggested: boot-time failures shall be handled before the safety application starts.

a. Trip time is defined as the maximum MCU operation time without Power-On Reset.

b. Process Safety Time (PST), also named Fault Tolerance Time Interval (FTTI), is maximum time between the first faulty output and a failure indication or reset.

3 Functional safety requirements for application software

This section gives an overview of suggested measures when using the individual modules of the SPC564A7x/SPC564A80/RPC564A80.

It is possible to ignore aspects of the text if equivalent measures that are taken can be shown to manage the same failures. Modules not explicitly covered by this document are assumed not safety relevant and do not require any software measures.

3.1 Application software requirements

Application software shall be developed according to safety requirements.

The following sections contain suggested assumptions and requirements for using the SPC564A7x/SPC564A80/RPC564A80 devices in a safety application.

3.2 Core

Suggested: all exception shall be enabled, if not enabled by default, and managed. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF.

Suggested: this safety mechanism consists of two redundant diverse software implementations in one hardware channel. Using different hardware resources (e.g. different RAM, ROM memory ranges) can increase the diagnostic coverage. For more details see ISO26262-5 D.2.3.4 technique.

Suggested: this safety mechanism checks the sequence of executed program tasks in order to detect a defective program sequence. A defective program sequence exists if the individual tasks of a program (e.g. software modules, functions or statements) are processed in the wrong sequence. For more details see ISO26262-5 D.2.9.5 technique. These specific software countermeasures can run once per FTTI.

Suggested: this safety mechanism is intended to supervise the reliability of program execution in consideration of periodicity and maximum timing constraints of periodicity. For more details see ISO26262-5 D.2.9.5 technique. These specific software countermeasures can run once per FTTI.

Suggested: specific software countermeasures shall be implemented to detect Core permanent faults. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF and/or once per FTTI.

3.3 System Clock and Frequency-Modulated Phase-Locked Loop (FMPLL)

External oscillator (XOSC) and FMPLL output are monitored by the hardware module called Clock Quality Monitor (CQM).

Suggested: FMPLL shall be configured to use the external oscillator (XOSC) as their source clock and all safety relevant modules shall be clocked with the FMPLL generated

clock signal. The CQM loss-of-clock (XOSC failure, i.e. FMPLL reference failure) and the CQM loss-of-lock (FMPLL failure) detection shall be enabled with relevant ISR request. The management of these errors is application-dependent. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF.

Rationale: to reduce the impact of glitches stemming from the external quartz crystal or the IRCOSC and to check the FMPLL clock integrity.

Suggested: It shall be checked that the device is using FMPLL clock as system clock and the integrity of the CQM module before running the SIF.

Rationale: to check the correctness of FMPLL configuration and the integrity of the CQM module.

Implementation hint: e.g. a wrong PLL configuration is set in order to inject a loss-of-lock and then the CQM response will be tested.

3.4 General-Purpose Static RAM (SRAM)

Suggested: the system SRAM is protected by a single error correction/dual error detection (SEC/DED) ECC scheme. The SEC/DED ECC scheme reporting shall be configured (interrupt request). The SRAM SEC/DED concerns data and not the addresses. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF.

Suggested: in order to increase the diagnostic coverage, specific software countermeasures shall be implemented to detect RAM address logic faults. These specific software countermeasures can run once per FTTI.

Rationale: to verify the integrity of RAM address logic.

Implementation hint: e.g. known pattern can be written and then read-back.

Suggested: in order to increase the diagnostic coverage, specific software countermeasures shall be implemented to detect fault in the RAM ECC logic. Aim is to assure that correct data are not accidentally modified and that bit errors are properly corrected/detected. These specific software countermeasures can run once per FTTI.

Rationale: to verify the integrity of RAM ECC logic.

Implementation hint: ECSM module can force the generation of single-bit and/or doublebit data inversions in RAM allowing the check of the ECC logic. In particular ECSM module can generate errors during data write cycles, such that subsequent reads of the corrupted address locations generate ECC events, either single-bit corrections or double-bit noncorrectable errors that are terminated with an error response.

Suggested: in order to increase the diagnostic coverage, one or more industry-standard MBIST algorithms (such as the “March” algorithm, the checkerboard algorithm, the varied pattern background algorithm and the array BIST) shall be implemented to protect the system SRAM against hardware dormant faults. The implemented MBIST algorithms can run once after the Power-On Reset (POR).

Rationale: to check the integrity of the RAM memory.

3.5 FLASH Memory

Suggested: FLASH memory is protected by a single error correction/dual error detection (SEC/DED) ECC scheme. The SEC/DED ECC scheme reporting shall be configured (interrupt request). The FLASH memory SEC/DED concerns data and not the addresses. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF.

Suggested: in order to increase the diagnostic coverage, specific software countermeasures shall be implemented to detect FLASH memory address logic faults. These specific software countermeasures can run once per FTTI.

Rationale: to verify the integrity of FLASH memory address logic.

Implementation hint: e.g. known pattern can be read.

Suggested: in order to increase the diagnostic coverage, specific software countermeasures shall be implemented to detect FLASH ECC logic faults. The aim is to assure that correct data are not accidentally modified and that single bit errors are rightly corrected. These specific software countermeasures can run once per FTTI.

Rationale: to verify the integrity of FLASH ECC logic.

Implementation hint: see *Chapter 5: ECC logic test* for further details.

Suggested: in order to increase the diagnostic coverage, an MBIST algorithms shall be implemented to protect the system FLASH memory against hardware dormant faults. The implemented MBIST algorithms can run once after the Power-On Reset (POR).

Rationale: to check the integrity of the FLASH memory.

Implementation hint: hardware support test called Array Integrity Self Check can be used (refer to the SPC564A7x/SPC564A80/RPC564A80 Reference Manual to have all additional details).

Suggested: in order to check the correctness of the writing process, written data shall be read-back and compared with the intended ones. These specific software countermeasures can run once after every write operation or after a series of write operations.

Rationale: to verify that written data are coherent with the intended ones.

3.6 Interrupt Controller (INTC)

Suggested: integrity of the INTC module shall be checked. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF. Implementation hint: e.g. the INTC module is configured to generate some interrupt requests and the expected behavior is verified.

Suggested: considering that no specific hardware protection is implemented against failures in the Interrupt Controller, spurious/missing interrupt requests caused by Electromagnetic Interference (EMI) or by bit flips in the interrupt registers of the peripherals, applications not resilient against such errors shall include detection or protection software countermeasures. These specific software countermeasures can run for each interrupt request.

Rationale: to verify interrupt requests are serviced correctly.

Implementation hint: e.g. spurious interrupts can be detected checking corresponding interrupt status in the interrupt status register of the related peripheral before executing the Interrupt Service Routine (ISR) code and missing interrupts, if they are synchronous, can be detected checking the program flow.

3.7 Enhanced Direct Memory Access (eDMA)

Suggested: considering that no specific hardware protection is implemented against failures in the eDMA, spurious/missing eDMA requests caused by Electromagnetic Interference (EMI) or by bit flips in the interrupt registers of the peripherals, applications not esilient against such errors shall include detection or protection software countermeasures. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF.

Rationale: to verify eDMA requests are serviced correctly.

Implementation hint: specific software countermeasures shall be implemented to detect eDMA permanent faults. These specific software countermeasures are applicationdependent.

3.8 Communication peripherals

The SPC564A7x/SPC564A80/RPC564A80 includes the following communication peripherals:

- FlexCAN;
- Deserial Serial Peripheral Interface (DSPI);
- FlexRay Communication Controller (FlexRay);
- Enhanced Serial Communication Interface (eSCI).

Suggested: an appropriate safety software protocol should be utilized (e.g. Fault Tolerant Communication Layer or FTCOM) for any communication peripheral employed to meet safety application requirements.

3.9 I/O peripherals

The SPC564A7x/SPC564A80/RPC564A80 includes the following I/O peripherals:

- System Integration Unit Lite (SIUL);
- Configurable Enhanced Modular IO Subsystem (eMIOS200);
- Enhanced Time Processing Unit (eTPU2).

These modules shall be used to implement the following functions if they are part of the application:

- Read digital inputs;
- Read PWM inputs;
- Write digital outputs;
- Write PWM outputs.

3.9.1 Read digital inputs

Suggested: digital inputs shall be acquired redundantly. Each double acquisition can be implemented using two pads configured as GPIs by the SIU unit. Digital input signal shall be applied on selected pads in order to be acquired and then the acquired values shall be compared by software.

Rationale: to verify that the two input values match.

Implementation hint: if, for a specific application, a plausibility check on a single acquisition assures sufficient diagnostic coverage, it can replace the redundant acquisition. This hint is a special case of deviation from recommended requirements as described in the Preface. The two selected pads shall not be physically adjacent to minimize CCFs. Each pad not dedicated to a specific function can be configured as GPIO with the exception of ADC pads, as they can only be configured as GPIs. SIU pads can be configured via the relevant pad configuration registers (PCRn).

3.9.2 Read PWM inputs

Suggested: digital inputs shall be acquired redundantly. Each double acquisition can be implemented using two pads configured as eMIOS200 channel by the SIU unit and configured with input capture feature by eMIOS200 module. PWM input signal shall be applied on selected pads in order to be acquired and then the acquired set of data (duty cycle and period) shall be compared by software.

Rationale: to verify that the two sets of data match.

Implementation hint: the comparison must take into account possible approximation because of different capturing of the input asynchronous signals. Each pad not dedicated to a specific function can be configured as GPIO with the exception of ADC pads, as they can only be configured as GPIs. SIU pads can be configured via the relevant pad configuration registers (PCRn). The two selected pads shall not be physically adjacent to minimize CCFs. PWM input signal can be generated using one other eMIOS200 channel correctly configured or one eTPU2 channel correctly configured and the eTPU2 channels instead of the eMIOS200 channels can be used.

3.9.3 Write digital outputs

Suggested: digital outputs shall be written either redundantly or with read-back. Write digital output operation can be implemented as single write digital output with read-back or double write digital output.

Rationale: to verify that the two output values match.

Implementation hint: if, for a specific application, a plausibility check on a single write assures sufficient diagnostic coverage, it can replace the redundant write or the write with read-back. This hint is a special case of deviating from recommended requirements as described in the Preface. Each pad not dedicated to a specific function can be configured as GPIO with the exception of ADC pads, as they can only be configured as GPIs. SIU pads can be configured via the relevant pad configuration registers (PCRn).

Single write digital output with read-back

Implementation hint: SIU pads are used to perform a single write digital output with readback. The read-back shall be done using the external configuration or the internal. SIU pads shall be configured as follows:

- External read-back: one SIU pad is configured to allow read-back of the output write on the selected SIU pad and the loop-back is done with an external connection outside the device. Using this configuration, only half of the available digital outputs can be used as safety outputs. In case of External read-back, the two selected pads shall not be physically adjacent;
- Internal read-back[c]: one SIU pad is configured to allow read-back of the output write on the selected SIU pad via an internal read path. Using this configuration, all available digital outputs can be used as safety outputs.

Double Write Digital Outputs

Implementation hint: SIU pads are used to perform a double write digital output. SIU pads shall be correctly configured and the output write of the selected channels shall be implemented following these guidelines:

- The two outputs are written with a single instruction to the appropriate register;
- The output register is read-back.

The two selected pads shall not be physically adjacent to minimize CCFs. Each pad not dedicated to a specific function can be configured as GPIO with the exception of ADC pads, as they can only be configured as GPIs. SIU pads can be configured via the relevant pad configuration registers (PCRn).To write two (or more) GPIOs with a single write instruction, the Parallel GPIO Pad Data Out (PGPDOx) register can be used. User has to take care that the two selected GPIOs are controlled by the same PGPDOx register. To protect the value of the other GPIOs that belong to the same PGPDOx register, the Masked Parallel GPIO Pad Data Out (MPGPDOx) register shall be properly configured before writing the PGPDOx register.

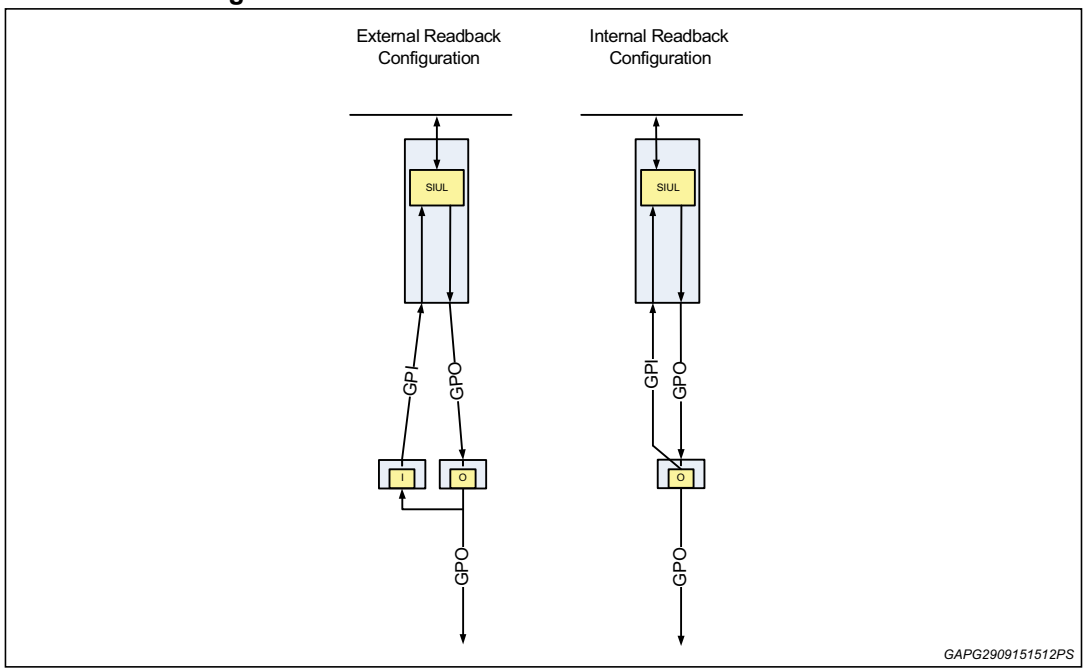
3.9.4 Write PWM outputs

Suggested: some PWM outputs shall be written either redundantly or with read-back. Write PWM output operation can be implemented as single write PWM output with read-back or double write PWM output.

Rationale: to verify that the two sets of data match.

c. Internal read back does not cover package faults (e.g. wire bond). Refer to the specific reference manual to verify the availability of the internal read path.

Figure 1. Block scheme of external/internal read-back



Single write PWM output with read-back

Implementation hint: Two pins are used to implement this function. One pin to generate the PWM output and another pin to read-back this PWM to check its integrity.

Each single write with read-back can be implemented using pads configured as eMIOS200 channel by the SIU unit and configured as PWM by eMIOS200 module.

PWM set of data (duty cycle and period) shall be applied by software. The read-back shall be done using the external configuration or the internal. SIU pads shall be configured as follows:

- External read-back: one SIU pad is configured as eMIOS200 channel by the SIU unit and configured with input capture feature by eMIOS200 module to allow read-back of the output written on the output pad. The loop-back is done with an external connection outside the device. In case of External read-back, the two selected pads shall not be physically adjacent;
- Internal read-back[(d)]: one SIU pad is configured as eMIOS200 channel by the SIU unit and configured with input capture feature by eMIOS200 module to allow read-back of the output write on the selected pad. The loop-back is done via an internal read path.

The two selected pads shall not be physically adjacent to minimize CCFs. SIU pads can be configured via the relevant pad configuration registers (PCRn). eTPU2 channels instead of eMIOS200 channels can be used.

d. Internal read back does not cover package faults (e.g. wire bond). Refer to the specific reference manual to verify the availability of the internal read path.

Double Write PWM Outputs

Implementation hint: Each double write can be implemented using two pads configured as eMIOS200 channel by the SIU unit and configured as PWM by eMIOS200 module. PWM set of data (duty cycle and period) shall be applied by software. The two selected pads shall not be physically adjacent to minimize CCFs. SIU pads can be configured via the relevant pad configuration registers (PCRn). eTPU2 channels instead of eMIOS200 channels can be used.

3.10 Enhanced Queued Analog-to-Digital Converter (eQADC)

The SPC564A7x/SPC564A80/RPC564A80 device is equipped with one eQADC module integrating two analog-to-digital converter macro-cells.

Suggested: acquisition of some reference voltages shall be done. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF and/or once per FTTI.

Rationale: to check the integrity of the eQADC module.

Implementation hint: some eQADC channels are internally connected to some reference voltages as Buffered Band Gap, Reference Voltage for 1.2 V LVD and so on (refer to the SPC564A7x/SPC564A80/RPC564A80 Reference Manual to have all additional details).

Moreover recommendation is to acquire analog input redundantly.

This module, if safety relevant, shall be used to implement the following functions:

- Double read analog inputs;
- Additional mechanisms.

3.10.1 Double read analog inputs

Suggested: safety relevant analog inputs shall be acquired redundantly using both analog-to-digital converter macro-cells integrated in the eQADC module. The measured values shall be compared by software.

Rationale: to verify that the two measured analog input values match.

Implementation hint: shared channels shall not be used for double read operation in order to avoid CCFs due to the pad sharing.

Implementation hint: SIU pads can be configured via the relevant pad configuration registers (PCRn).

3.10.2 Additional mechanisms

The two analog-to-digital converter macro-cells share the same digital interface. To increase the diagnostic coverage against failures impacting this common logic, some additional counter measures can be developed. For example:

- Oversampling;
- Plausibility check.

Suggested: the analog inputs shall be acquired redundantly in time.

Rationale: to increase the diagnostic coverage.

Implementation hint: the sampling rate shall be significantly higher than the Nyquist Frequency related to the input signal. The acquired values shall be compared by software in order to verify the correlation. In case of fault, the acquired values are not correlated with themselves. Against random faults, at least three consecutive analog values shall be acquired for each analog input.

3.11 Temperature sensor

SPC564A7x/SPC564A80/RPC564A80 devices are equipped with a temperature sensor in order to monitor the device temperature. This temperature sensor generates a voltage that increases linearly with temperature and that can be read by software using the on-board eQADC module, so the read value can be used with the band-gap voltage and constants stored in flash memory during factory test to calculate device junction temperature.

Suggested: the temperature sensor output voltage shall be read by software and the corresponding temperature shall be compared with the upper limit of the operating range. In case an over-temperature fault is detected, the device shall be moved to a safe state. This check shall run once per FTTI.

Rationale: to detect over-temperature faults.

Implementation hint: to set the proper operating range threshold, the temperature sensor accuracy of 10° C and the maximum operating junction temperature of 150 °C (see device data sheet) shall be considered.

Note: External temperature sensor could be used to check internal temperature sensor output.

3.12 Software Watchdog Timer (SWT)

Suggested: SWT module shall be used to implement control flow monitoring function. The SWT shall be clocked by oscillator clock. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF. However, other control flow monitoring approaches that do not use the SWT may also be used.
SPC564A7x/SPC564A80/RPC564A80 devices provide the hardware support (SWT) to implement both control flow monitoring and temporal flow monitoring methods.

Rationale: to detect a defective program sequence.

Implementation hint: SWT can be enabled asserting the bit SWT_MCR[WEN] and the configuration registers can be hard-locked asserting the bit SWT_MCR[HLK]. The timeout register (SWT_TO) must contain a 32-bit value that represents a timeout less than the FTTI Before the safety function is executed, software must verify that the SWT is enabled checking the bit SWT_MCR[WEN]. If Windowed mode and Keyed Service mode (two pseudo-random key values used to service the watchdog) are enabled, it is possible to reach a high effective temporal flow monitoring.

3.13 Cyclic Redundancy Checker Unit (CRC)

CRC module shall be used to detect accidental alteration of data during storage or transmission operations. This shall be done for each storage or transmission operation.

Suggested: correct working of the CRC module shall be checked. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF.

Implementation hint: e.g. the CRC signature of a known data pattern shall be calculated and it is compared to the expected one, i.e. the off line calculated CRC signature or the CRC signature of a random data pattern shall be redundantly calculated by software and by the CRC module and then the two CRC signatures are compared.

Suggested: CRC module shall be used to check the correctness of the content of the configuration registers of each safety-related module. If CRC module is used by SEF, specific software countermeasures shall be implemented to detect or to protect against possible faults of the CRC module. This check shall run once per FTTI.

Implementation hint: e.g. the CRC signature of the content of the configuration registers of each safety-related module shall be calculated off line. At run time, the same CRC signature shall be calculated by the CRC module within the SPT. The run-time calculated CRC signature is then compared to the expected one, i.e. the off line calculated CRC signature. These operations allow also checking the integrity of the CRC module. Theoretically, CRC signature could be calculated by software using one or more industry-standard CRC algorithms, but practically, using the CRC module is more effective. To avoid CPU overloading, the EDMA module can be used to support the data transfer from the registers under check to the CRC module.

3.14 Multi-Layer AHB Crossbar Switch (XBAR)

Suggested: the configuration and the integrity of the XBAR shall be checked. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF and/or once per FTTI.

Implementation hint: e.g. the integrity of the XBAR module can be checked reading a checking pattern (stored in the FLASH memory) with the master Core and eDMA, calculating the CRC of the checking pattern and comparing this with the expected one. Different checking patterns (stored in different locations of the FLASH memory) could be chosen for each FTTI.

3.15 Memory Protection Unit (MPU)

The MPU provides hardware access control for all device memory locations.

Suggested: MPU shall be configured in order to ensure that all bus masters (Core, eDMA and FlexRay) can access only their allocated resources according to their access rights. The configuration and the correct working of the MPU shall be checked. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF and/or once per FTTI.

Rationale: to avoid giving access to the device resources to unauthorized master and denying access to authorized master.

Implementation hint: e.g. the integrity of the MPU module can be checked reading data from reserved and not-reserved FLASH memory locations with the master Core and eDMA and verifying if the MPU module gives access or not. Different FLASH memory locations could be chosen for each FTTI.

3.16 Peripheral Bridge (PBRIDGE)

Suggested: PBRIDGE shall be configured in order to ensure that all bus masters (Core, eDMA and FlexRay) can access only their allocated resources according to their access rights. The configuration and the correct working of the PBRIDGE shall be checked. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF and/or once per FTTI.

Rationale: to avoid giving access to the device resources to unauthorized master and denying access to authorized master.

Implementation hint: e.g. the integrity of the PBRIDGE module can be checked calculating the CRC of the configuration registers value of 3 IPs and comparing each one with the expected one. Different IPs could be chosen for each FTTI.

3.17 Power Management Controller (PMC)

SPC564A7x/SPC564A80/RPC564A80 devices use three supply voltages, nominally 5 V, 3.3 V and 1.2 V. The 5 V supply voltage must be supplied from the outside while the other supply voltages are supplied by internal regulators. Moreover, SPC564A7x/SPC564A80/RPC564A80 devices embed LVI for all supply voltages. The PMC controls the internal regulators and the LVI circuits.

Suggested: LVI failure reaction for all supply voltages shall be configured (system reset or interrupt request). These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF.

Rationale: to check if supply voltages are in the correct operation range.

Suggested: LVI circuits operation (for supply voltages generated by internal regulators, i.e. 3.3 V and 1.2 V) shall be checked. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF.

Implementation hint: the output of each internal regulator can be set to a value lower than the LVI threshold value configuring the PMC_ TRIMR register. According to this, enabling only the interrupt request as LVI failure reaction, the generation of the LVI interrupt requests confirms the correctness of LVI circuits operation. Then, the correct value of the PMC_ TRIMR register can be restored.

Suggested: correct execution of Power-On Reset sequence shall be checked. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF.

Implementation hint: e.g. reserved RAM is used to store a key which can be used if the current reset is a POR or not according to the POR bit in the ECSM_MRSR register. Moreover the default reset value of the registers of each IP can be checked.

3.18 Error Correction Status Module (ECSM)

The ECSM is able to detect data storage failures in memory (FLASH and SRAM) and address these.

The ECSM can detect and correct single-bit errors, detect double-bit faults and detect faults affecting more than two bits. ECC functionality concerns data and not the addresses. ECC

is automatically calculated on memory write accesses and is checked while read accesses are executed on memory.

Suggested: to enable ECC reporting logic in the ECSM in order to provide an optional managing failures interrupt mechanism. In addition to the interrupt generation, the ECSM captures specific information (memory address, attributes and data, bus master number, etc.) which can be useful for subsequent failure analysis.

Rationale: to manage failures and perform failure analysis.

3.19 Other modules

Suggested: all other modules, if safety relevant, shall be protected at application level.

4 Functions of external devices for safety applications

This section gives an overview of the external components suggested to use with the SPC564A7x/SPC564A80/RPC564A80 device.

Suggested: at system level some countermeasures have to be placed in order to bring the safety-critical outputs to their safe state (e.g., by pull-up or pull-down resistors).

It should be noted that the failure rates of external services are not included in FMEDA of the SPC564A7x/SPC564A80/RPC564A80 device and have to be included in the system FMEDA by the user himself.

4.1 External Watchdog function (EXWD)

Suggested: an external low-cost device, acting as system supervisor, shall also provide a watchdog to cover CCFs of the SPC564A7x/SPC564A80/RPC564A80 device. It shall be triggered periodically by the SPC564A7x/SPC564A80/RPC564A80 device.

Rationale: to detect CCF as a complete failure of the power supply.

Some common causes of failure (e.g., failure on power supply) are detected because the software no longer triggers the watchdog.

If a failure is detected, the EXWD moves, and maintains, the system (ECU level) to a Safe state condition within the FTTI (e.g., the EXWD disconnects from the power supply the SPC564A7x/SPC564A80/RPC564A80 device).

The user can choose how to implement the watchdog communication between the SPC564A7x/SPC564A80/RPC564A80 device and the external device (for example, communication via serial link or via toggling pin).

4.2 Power Supply Monitor function (PSM)

The SPC564A7x/SPC564A80/RPC564A80 device embeds LVI for all internal supplies. Latent failures impacting these LVIs can’t be detected.

Suggested: an external low-cost device, acting as system supervisor, shall provide also over-/under-voltage monitor for the SPC564A7x/SPC564A80/RPC564A80 on all supplies available externally.

Rationale: to ensure voltage power supply is within the defined operating range.

If the voltage power supply is out of the defined operating range, the PSM moves, and maintains, the system (ECU level) to a Safe state condition within the FTTI (e.g., the PSM disconnects from the power supply the SPC564A7x/SPC564A80/RPC564A80 device).

For the voltage power supply operating range, please refer to the SPC564A7x/SPC564A80/RPC564A80 device data sheet.

It should be noted that an over voltage outside the specified range may cause permanent damage to the SPC564A7x/SPC564A80/RPC564A80 device even if kept in reset.

4.3 PWM Output Monitor function (PWMM)

The eMIOS200 module and the eTPU2 module integrated in the SPC564A7x/SPC564A80/RPC564A80 device can generate PWM output signals.

In general, if the safety application uses these PWM output signals to control an actuator with short safety time against wrong control (such as the inverter of a three-phase motor control application with a dead-time requirements to avoid short circuits destroying the inverter and the motor), those requirements shall be supervised externally if the failure reaction delay within the SPC564A7x/SPC564A80/RPC564A80 device can exceed the safety time of the actuator.

The distinctive features that should be managed by the external device are the correctness of inserted dead-time and the occurrence of an open-circuit and/or short-circuit to supply or ground.

Suggested: an external low-cost device, acting as system supervisor, shall provide also a PWM monitor to check the generated PWM output signals.

Rationale: to check the accuracy of the PWM output signals.

If a failure is detected, the PWMM moves, and maintains, the system (ECU level) to a Safe state condition within the FTTI (e.g., the PWMM disconnects from the power supply the SPC564A7x/SPC564A80/RPC564A80 device).

Implementation hint: in case PWM signals drive the switches of a power stage, eMIOS200 channels or eTPU2 channels cannot be used to detect a dead-time fault because its failure indication time is normally greater than the time enough to produce a physical permanent failure of the power stage.

5 ECC logic test

5.1 Overview

This section describes the required information on how to develop the software for such ECC logic test.

The goal is to ensure high coverage of the ECC logic faults with minimum performance penalty to customer's application. Thus, the performance penalty must be less than 2% (e.g. the test time should be less than 200 ìf considering a FTTI of 10 ms).

The SPC564A7x/SPC564A80/RPC564A80 FLASH memory has a UTEST (user-test) mode ECC logic check feature which can be utilized for this ECC logic test. A data pattern with walking 0 through data and ECC parity bits can be applied during the ECC logic check procedure to achieve high fault coverage of the ECC logic and fast execution.

5.2 Data pattern – Walking 0

To reach the needed performances the use of the data pattern with walking 0 through data and ECC parity bits must be used. *Table 1* shows the data pattern.

Table 1. Data pattern used by the ECC logic test

Data vector number	8-bit ECC parity bits	64-bit data bits
0	0xFF	0xFFFF_FFFF_FFFF_FFFE
1	0xFF	0xFFFF_FFFF_FFFF_FFFD
2	0xFF	0xFFFF_FFFF_FFFF_FFFB
3	0xFF	0xFFFF_FFFF_FFFF_FFF7
4	0xFF	0xFFFF_FFFF_FFFF_FFEF
5	0xFF	0xFFFF_FFFF_FFFF_FFDF
6	0xFF	0xFFFF_FFFF_FFFF_FFBF
7	0xFF	0xFFFF_FFFF_FFFF_FF7F
…	…	…
62	0xFF	0xBFFF_FFFF_FFFF_FFFF
63	0xFF	0x7FFF_FFFF_FFFF_FFFF
64	0xFE	0xFFFF_FFFF_FFFF_FFFF
65	0xFD	0xFFFF_FFFF_FFFF_FFFF
…	…	…
71	0x7F	0xFFFF_FFFF_FFFF_FFFF
72	0xFF	0xFFFF_FFFF_FFFF_FFFF

5.3 UTEST mode ECC logic check

- The procedure to use the UTEST mode ECC logic check is listed as below:
- Enable UTEST mode (Write 0xF9F9_9999 to UT0 register, UT0[UTE] will be set).
- Write UT0[SBCE] to 1 (to enable single-bit error correction visibility).
- Write UT0[EIE] to 1.
- Write UT0[DSI], UT1[DAI] and/or UT2[DAI] bits to provide data and check bit values to be read. Single or Double bit detections/corrections can be simulated by properly choosing Data and Check Bit combinations.
- Write double word address to receive the data inputted in step 3 into the ADR register.
- Reads can now be done through the BIU in a Read Request type fashion. In the event of a BIU read requested from an address that matches the address in the ADR register, expected data, and corrections or detections should be observed based on data written into the UT0[DSI], UT1[DAI] and/or UT2[DAI] registers. MCR[EER] and MCR[SBCSBC] can be checked to evaluate the status of reads done.
- Repeat steps 4 to 6 for all the data vectors in the proposed test data pattern.
- Once completed, clear the UT0[EIE] bit to 0.

5.4 Fault coverage and execution time

The described ECC logic test reaches a 92.7% fault coverage of ECC decode logic.

The execution of the test code takes about 176 ìs at 80 MHz, room temperature and nominal voltages.

Appendix A Further information

5.5 Conventions and terminology

Table 2 shows the list of conventions for this document.

Table 2. List of conventions and terminology

Convention	Description
Error	Discrepancy between a computed, observed, or measured value or condition and the true, specified or theoretically correct value or condition.
Fault	Abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function.
Failure	The termination of the ability of a functional unit to perform a required function.

5.6 Acronyms and abbreviations

A short list of acronyms and abbreviations used in this document is reported in the following *Table 3*.

Table 3. Acronyms and abbreviations

Term	Meaning
CCF	Common Cause Failure
CRC	Cyclic Redundancy Check
DED	Dual Error Detection
ECC	Error Correcting Code
ECSM	Error Correction Status Module
eDMA	Enhanced Direct Memory Access
EXWD	External Watchdog function
eQADC	Enhanced Queued Analog-to-Digital Converter
FMEDA	Failure Modes, Effects and Diagnostic Analysis
FMPLL	Frequency-Modulated Phase-Locked Loop
FOM	Failure Output Monitor function
FTTI	Fault Tolerant Time Interval
GPIO	General Purpose Input/Output
LBIST	Logic Built-In Self-Test
LVI	Low Voltage Inhibit
MBIST	Memory Built-In Self-Test
MCU	Microcontroller Unit
MPU	Memory Protection Unit
PMC	Power Management Controller

Table 3. Acronyms and abbreviations (continued)

Term	Meaning
PSM	Power Supply Monitor function
PST	Process Safety Time
PWM	Pulse Width Modulation
RAM	Random Access Memory
SEC	Single Error Correction
SWT	Software Watchdog Timer

Appendix B Reference documents

1. SPC564A70B4, SPC564A70L7 32-bit MCU family built on the embedded Power Architecture® (RM0068, DocID18132)
2. 32-bit Power Architecture® based MCU for automotive powertrain applications (SPC564A70B4, SPC564A70L7 — DocID18078)
3. SPC564A74xx, SPC564A80xx/RPC564A80xx 32-bit MCU family built on the embedded Power
4. Architecture® (RM0029, DocID15177) 32-bit MCU family built on the embedded Power Architecture® (SPC564A74B4, SPC564A74L7, SPC564A80B4, SPC564A80L7 — DocID15399)
5. SPC564A70x device errata JTAG_ID = 0x0AE03041 (SPC564A70B4, SPC564A70L7 – DocID022776)
6. SPC564A70x device errata JTAG_ID = 0x1AE03041 (SPC564A70B4, SPC564A70L7 – DocID022787)
7. SPC564A80 device errata JTAG_ID = 0x0AE02041 (SPC564A80 — DocID16797)
8. SPC564A80 device errata JTAG_ID = 0x0AE02041 (SPC564A80 — DocID17624)
9. SPC564A80x device errata JTAG_ID = 0x1AE02041 (SPC564A80 — DocID18436)

Revision history

Table 4. Document revision history

Date	Revision	Changes
09-Apr-2013	1	Initial release.
24-Sep-2013	2	Updated Disclaimer.
02-Oct-2015	3	Robust root part numbers added.