AN4266
Application note

Safety application guide for SPC56xL70xx family

Introduction

This document is the safety application guide for the SPC56xL70xx. It provides the conditions of use for the SPC56xL70xx in ASIL D applications.

Contents

List of tables

List of figures

1 Preface

This document discusses requirements and assumptions for the use of the SPC56xL70xx Microcontroller Unit (MCU) in ASIL D applications. It prescribes several measures as mandatory (or mandatory under certain preconditions, for example, if a certain module is used) whereby the measure described was assumed to be in place when analyzing the safety of the MCU.

This document considers:

● The system assembly that contains the SPC56xL70xx MCU
● The "Safety Element out of Context" section in the "Road vehicles - Functional safety - Part 10: Guideline [ISO/DIS 26262-10]" standard
● Certain assumptions about the assembly's functional safety needs based on that standard

and determines whether a measure is mandatory or not based on these factors.

What this means for designers using the SPC56xL70xx MCU is that if they don't fulfill a specific Safety Application Guide (SAG) prescription they either have to show to their ISO 26262 assessor that the alternative solution is similarly efficient concerning the safety requirement in question (for example, provides the same coverage, avoids Common Cause Failure (CCF) as effectively, and so on), or they have to specify the increased failure rate/reduced Safe Failure Fraction (SFF) they estimate to incur due to the deviation. Otherwise, the assessor will not recognize the MCU certificate that the customer received with the MCU.

This document also contains guidelines on how to configure and operate the SPC56xL70xx for ASIL D applications. These guidelines are preceded by one of the following bold text statements:

● Implementation hint
● Recommended
● Example

These guidelines are considered to be useful approaches for the specific topics under discussion, but are not mandatory. The user will need 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.

Mandatory: This document is only valid if the environmental conditions given in the SPC56xL70xx data sheet are maintained.

The cores in the SPC56xL70xx can be configured to operate in either Lock-Step Mode (LSM) or Decoupled Parallel Mode (DPM). In LSM, the outputs of a set of replicated modules, identified as the Sphere of Replication (SoR, see *Section 2.5, Sphere of Replication* for details), are compared to ensure that the operations or transactions that are executed are identical on a clock per clock basis.

Mandatory: This document is based on the assumption that the SPC56xL70xx is configured to operate in LSM.

As for all devices, device errata must be taken into account during system design and implementation. For a safety-related device such as the SPC56xL70xx, this also concerns safety-related activities such as system safety concept development.

Mandatory: The device shall be handled according to JEDEC standards J-STD-020 and J-STD-033.

Mandatory: To cover the ISO-07-6.5.4 and ISO-07-6.4.2.1, customers shall report all field failures of the devices to silicon supplier.

Mandatory: This document is only valid if the conditions given in the addendum are met (see *Section 8.3: Document references*).

2 General information

2.1 Mission profile

The assumed mission profile is:

● Lifetime: 20 years

● Total operating hours: 12000 hours

● Trip time: 10 hours (Trip time is defined as the maximum time of operation of the MCU without power-on reset)

● Fault Tolerant Time Interval (FTTI, also named Process Safety Time (PST)): 10 ms (maximum time between the first faulty output and a failure indication or reset)

Temperature profiles for packaged devices (*Table 1*) and bare die (*Table 2*) are shown below.

Note: *The temperature profile is an assumption of the SPC56xL70xx safety analysis and shall be fulfilled during integration into an ASIL D compliant system.*

Table 1. Temperature profile for packaged device

Temperature range (°C)	Operation time (h)
125–135	120
110–120	960
90–100	7680
30–40	3240

Table 2. Temperature profile for bare die device

Temperature range (°C)	Operation time (h)
120–125	120
100–110	960
80–90	7680
20–30	3240

2.2 Safe state

By definition, the Safe states of the SPC56xL70xx are as follows:

● Completely unpowered

● Reset

– All pins except possibly the error output pins (FCCU_F[0:1]) are tristated.

● Operating correctly

– Outputs depend on application.

● Explicitly indicating an internal error

– Error output pins FCCU_F[0:1] are in a state indicating an error, and the state of other I/O pins will not be reliable.

Defining these states as safe for the MCU means that the overall system must react safely to the SPC56xL70xx being in, and entering, any of these states. For the 'Completely unpowered' and 'Reset' states the addition of a pullup or pulldown resistor on relevant signals may be necessary. If an 'Explicit indication of internal error' occurs on FCCU_F[0:1], the application must not depend on the MCU for continued operation. This also means that the system must be able to remain in a safe state without any additional actions from the MCU.

Mandatory: The system must transition to a safe state when there is an indication of an error.

Depending on the configuration the system may disable, or reset, the SPC56xL70xx as a reaction to the error signal.

If a system continuously switches between a standard operating state and the reset state, without any device shutdown, the system is not considered to be in a Safe state.

Mandatory: The application must identify and signal such switching as a failure condition.

2.3 Failure indication time

The SPC56xL70xx failure indication time must be taken into consideration when determining application safety strategies, because it must be less than the FTTI.

Failure indication time has three components, two of which are influenced by configuration settings: **recognition time + internal processing time + indication time**.

Each component of failure indication time is described as follows:

● **Recognition time** is the maximum of the recognition time of all involved safety mechanisms. The three mechanisms with the longest time are:
  – ADC[a] recognition time is the most demanding HW test in terms of timing. The self-test requires the ADC conversion to complete a full test. A single full test takes at least 70 μs[b].
  – Recognition time related to the FMPLL loss of clock: it depends on how the FMPLL is configured, but is approximately 20 μs.
  – Diagnostic cycle time of software self-tests. This time depends closely on the software implementation.

● **Internal processing time** lasts maximum 10 RC clock cycles (RC is the internal safe clock with nominal frequency of 16 MHz).

● **Indication time**, the time to notify an observer about the failure, depends on indication protocol configured in the Fault Collection and Control Unit (FCCU):
  – Dual Rail protocol and time switching protocol:
    FCCU configured as "fast switching mode": indication delay is maximum 64 μs. As soon as FCCU receives a fault signal, FCCU reports the failure to the outside world via output pin (if properly configured).
    0 = FCCU configured as "slow switching mode": an indication delay could occur. The maximum delay is equal to period of the error out signal. This parameter shall be configured equal to its minimum which is 128 μs.

---

a. ADC recognition time shall be used only if ADC is used by the safety function.

b. This value takes into account the steps needed to run the three ADC hardware self-tests.

0 = Bi-stable protocol: indication delay is maximum 64 µs. As soon as the FCCU receives a fault signal, it reports the failure to the outside world via output pin (FCCU_F[0:1], if properly configured).

If the configured reaction to a fault is an interrupt, an additional delay (interrupt latency) can occur until the interrupt handler is able to start executing (for example, higher priority IRQs, XBAR contention, register saving, and so on).

General failure rate, or the Failure Modes, Effects and Diagnostic Analysis (FMEDA) report, is available upon request when covered by an NDA (contact your representative).

2.4 Error handling

Error handling can be split into two categories:
● Handling of errors during runtime
● Handling of errors during boot time (for example, Logic Built-In Self-Test (LBIST), Memory Built-In Self-Test (MBIST))

Mandatory: Runtime errors shall be handled in a time shorter than the FTTI.

Mandatory: Boot time failures shall be handled before the safety function starts.

Note: ***Implementation hint:*** *To satisfy this requirement regarding the LBIST/MIBST, Self-Test Control Unit (STCU) status condition shall be checked by application software before safety application starts (See "Integrity SW Operations" section of the "Self-Test Control Unit (STCU)" chapter in the SPC56xL70xxReference Manual for details).*

2.5 Sphere of Replication

Sphere of Replication (SoR) is used for duplicating of critical components on the SPC56xL70xx. The following modules are included in the SoR:
● e200z4 Cores
● Enhanced Direct Memory Access (eDMA)
● Interrupt Controller (INTC)
● Crossbar Switch (XBAR)
● Memory Protection Unit (MPU)
● Flash memory controller
● Static RAM Controller (SRAMC)
● System Timer Module (STM)
● Software Watchdog Timer (WDT)
● Peripheral Bridge (PBRIDGE)

3 Functional safety requirements for application software

This section gives an overview of necessary, or recommended, measures when using the individual modules of the SPC56xL70xx. If a module is implemented without following the text of this section, the safety certificate for the module, or the entire MCU, may not be validated. 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 do not require any software measures.

The modules covered by the SoR reach very high Diagnostic Coverage (DC) without dedicated measures at application or system levels.

3.1 Application software requirements

Application software shall be developed according to ASIL D requirements.

3.1.1 Mandatory software requirements

The following sections contain **Mandatory** design constraints for using the SPC56xL70xx devices in an ASIL D system:

● *Section 3.2, System Status and Configuration Module (SSCM)*
● *Section 3.3, Self-Test Control Unit (STCU)*
● *Section 3.4, Reset Generation Module (MC_RGM)*
● *Section 3.5, Clock configuration*
● *Section 3.7, Flash memory*
● *Section 3.8, Interrupt Controller (INTC)*
● *Section 3.10, Enhanced Direct Memory Access (eDMA) requests*
● *Section 3.11, Periodic Interrupt Timer (PIT)*
● *Section 3.13, I/O peripherals*
● *Section 3.14, Cross Triggering Unit (CTU)*
● *Section 3.15, ADC*
● *Section 3.16, Temperature sensors*
● *Section 3.17, Software Watchdog Timer (SWT)*
● *Section 3.19, Cyclic Redundancy Checker Unit (CRC)*
● *Section 3.20, Clock Monitor Unit (CMU)*
● *Section 3.21, Frequency-Modulated Phase-Locked Loop (FMPLL)*
● *Section 3.22, Internal RC Oscillator (IRCOSC)*
● *Section 3.23, Power Management Unit (PMU)*
● *Section 3.25, Register Protection Module*
● *Section 3.27, Fault Collection and Control Unit (FCCU)*

3.1.2 Recommended software requirements

The following sections contain **Recommended** design constraints for using the SPC56xL70xx devices in an ASIL D system:

- *Section 3.6, SRAM*
- *Section 3.12, Communication peripherals*
- *Section 3.13, I/O peripherals*
- *Section 3.16, Temperature sensors*
- *Section 3.18, Redundancy Control Checking Unit (RCCU)*
- *Section 3.19, Cyclic Redundancy Checker Unit (CRC)*
- *Section 3.24, Memory Protection Unit (MPU)*
- *Section 3.25, Register Protection Module*
- *Section 3.26, Error Correction Status Module (ECSM)*

3.1.3 Implementation details

The following sections contain implementation details for using the SPC56xL70xx devices in an ASIL D system:

- *Section 3.2, System Status and Configuration Module (SSCM)*
- *Section 3.5, Clock configuration*
- *Section 3.7, Flash memory*
- *Section 3.8, Interrupt Controller (INTC)*
- *Section 3.10, Enhanced Direct Memory Access (eDMA) requests*
- *Section 3.13, I/O peripherals*
- *Section 3.14, Cross Triggering Unit (CTU)*
- *Section 3.16, Temperature sensors*
- *Section 3.17, Software Watchdog Timer (SWT)*
- *Section 3.19, Cyclic Redundancy Checker Unit (CRC)*
- *Section 3.20, Clock Monitor Unit (CMU)*
- *Section 3.21, Frequency-Modulated Phase-Locked Loop (FMPLL)*
- *Section 3.23, Power Management Unit (PMU)*
- *Section 3.25, Register Protection Module*
- *Section 3.27, Fault Collection and Control Unit (FCCU)*

Note: *A section may contain **Mandatory** constraints, **Recommended** constraints, **Implementation hints** or any combination of the three.*

3.2 System Status and Configuration Module (SSCM)

3.2.1 Configuration

Mandatory: Before executing the safety functions, the SSCM shall be configured to inhibit unintentional execution of the BAM code.

Note: ***Rationale:*** *Since BAM code is not intended to be executed by ASIL D applications, any execution of the BAM, or part of it, must be inhibited.*

Note: **Implementation hint:** *This requirement is satisfied by writing SSCM_ERROR[PAE] = 1. Each access to the BAM memory area produces a Prefetch or Data Abort exception.*

3.2.2 Checking

Mandatory: After boot, but before executing any safety function, the application software needs to read SSCM_STATUS[LSM] to verify that the device runs in the selected mode of operation:

● Decoupled Parallel Mode (DPM) – SSCM_STATUS[LSM] = 0

● Lock Step Mode (LSM) – SSCM_STATUS[LSM] = 1

Note: **Rationale:** *To check if the MCU started in LSM*

3.3 Self-Test Control Unit (STCU)

3.3.1 Configuration

The STCU does not require any configuration written by application software. The default STCU configuration is to execute LBIST/MBIST and to react to detected faults by triggering a Non-Critical Fault (NCF) that signals the FCCU (See "Self-Test Control Unit (STCU)" chapter in the SPC56xL70xx Reference Manual for details).

Mandatory: LBISTs and MBISTs shall be configured to be executed once per trip time (trip time defined in *Section 2.1, Mission profile*).

3.3.2 Checking

Mandatory: Once after boot, before the safety application starts, application software shall carry out some STCU checking steps for ensuring STCU reliability.

Note: **Implementation hint:** *See "Integrity SW Operations" section of the "Self-Test Control Unit (STCU)" chapter in the SPC56xL70xxReference Manual for details.*

Note: **Rationale:** *STCU manages the execution, and checks the result, of the LBISTs and MBISTs. The STCU's correct behavior must be verified by checking the expected results with software.*

The Integrity SW should confirm that all MBISTs and LBISTs finished successfully with no additional errors flagged. This software confirmation prevents a fault within the STCU itself from incorrectly indicating that the self-test passed.

This is an additional safety layer since the STCU propagates the LBIST/MBIST and internal faults using the NCF signals of the FCCU. So, reading STCU_LBS, STCU_LBE, STCU_MBSL, STCU_MBSH, STCU_MBEL, STCU_MBEH and STCU_ERR registers helps increase the STCU auto-test coverage.

3.4 Reset Generation Module (MC_RGM)

A redundant fault notification path is achieved through the use of the MC_RGM and the FCCU. MC_RGM configuration is application dependent.

Mandatory: However, to have the redundant notification path, both MC_RGM and FCCU shall be configured to react to critical application faults.

Note: **Rationale:** *To have two notification paths in case of an error*

3.5 Clock configuration

The system starts by using the internal RC oscillator clock (IRCOSC) as its source (See "Oscillators" chapter in the SPC56xL70xx Reference Manual and *Section 3.22, Internal RC Oscillator (IRCOSC)* below for details on IRCOSC configuration).

Mandatory: Before safety functions are executed, the FMPLLs must be configured to use the external oscillator (XOSC) as their source clock.

Note: **Rationale:** *Since the IRCOSC is used by the CMUs as reference to monitor the output of the two PLLs, it can not be used as input of these PLLs.*

Note: **Implementation hint:** *MC_CGM_AC3_SC[SELCTL] and MC_CGM_AC4_SC[SELCTL] must be set to 1 to select the XOSC.*

Mandatory: All safety relevant modules shall be clocked with an FMPLL generated clock signal.

Note: **Rationale:** *To reduce the impact of glitches stemming from the external quartz crystal and its hardware connection to the MCU*

Note: **Implementation hint:** *This requirement is fulfilled by appropriately programming the Clock Generation Module (MC_CGM) Clock Divider Configuration and Clock Select Control registers and Mode Entry Module (MC_ME) MC_ME_<mode>_MC registers (See "Clock Generation Module (MC_CGM)" and "Mode Entry Module (MC_ME)" chapters in the SPC56xL70xx Reference Manual for details).*

3.6 SRAM

The system SRAM is protected against hardware dormant faults by hardware BISTs (See "MBIST partitioning" section in the "Self-Test Control Unit (STCU)" of the SPC56xL70xx Reference Manual). This test runs at boot, but some software actions are requested (See *Section 3.3, Self-Test Control Unit (STCU)*).

Moreover, the system SRAM is also protected by a single error correction/dual error detection (SEC/DED) ECC scheme. The SRAM SEC/DED concerns data and addresses and thus provides diagnostic coverage to logic addresses.

3.7 Flash memory

Non-volatile memory (NVM) flash memory is protected with an SEC/DED ECC scheme.

Caution: The single-bit correction reporting functionality is not available as described for flash memory ECC (See errata e3320). In case single-bit corrections need to be tracked, the workaround in the errata shall be used. Be aware that the workaround has a higher probability than the original mechanism to miss corrections if several occur within a short time.

To support the detection of dormant faults in the entire memory array and addressing logic, and to check the integrity of the logic used for flash memory programming, the following BISTs must be enabled by software:

● **Mandatory:** Array Integrity Self Check – This BIST is based on functionality built into the flash memory control logic. It calculates a MISR signature over the array content and thus validates the content of the array as well as the decoder logic. The calculated MISR value is dependent on the array content and must be validated by software. **Frequency:** This check must be performed at boot time.

Note: **Rationale:** *To check the integrity of the flash memory array content*

Note: **Implementation hint:** *This BIST must be started by application software; its result must be validated by reading the corresponding registers in the flash memory controller after it has been finished (See "Array integrity self check" section in the "Flash memory" chapter of the SPC56xL70xx Reference Manual for detailed information about this BIST).*

● **Mandatory:** Write operation – When writing flash memory, the corresponding SW driver must validate the correctness of the programming of flash memory by checking the value of C90FL_MCR[PEG]. Furthermore, the data that was written must be read back, then verified by SW that it compares with the intended data value. **Frequency:** After every write operation or after a series of write operations

Note: **Rationale:** *To verify that the written data is coherent with the expected data*

● **Mandatory:** Flash memory ECC logic test – This BIST tests the (digital) logic within the flash memory that is responsible for detecting and correcting faults (ECC logic) in the read data.

Note: **Rationale:** *The intention of this test is to assure that correct data is not accidently modified, and single-bit errors are correctly updated.*

Reading a set of data words from flash memory and comparing it with expected values is a software initiated function that is controlled by the application. **Frequency:** Once per FTTI

Note: **Implementation hint:** Section 6, ECC logic test *explains how to perform flash memory data compares with SW.*

3.8 Interrupt Controller (INTC)

No specific hardware protection is provided against spurious or missing interrupt requests caused by Electromagnetic Interface (EMI) on the interrupt lines, or bit flips in the interrupt registers of the peripherals[c].

Mandatory: Applications that are not resilient against such errors must include detection or protection measures.

Note: **Rationale:** *To manage spurious or missing interrupt requests*

Note: **Implementation hint:** *A possible way to detect spurious interrupts is to check corresponding interrupt status in the interrupt status register of the related peripheral before executing the Interrupt Service Routine (ISR) service code.*

---

c. INTC is a replicated module. No software action is needed to detect faults inside this module.

3.9 Semaphore Unit (SEMA4)

Semaphore modules are only used in DPM. Failures of the SEMA4 module may cause unwanted interrupts in LSM. Each SEMA4 unit is connected to both replicated INTC modules. This means that even in LSM when SEMA4 units are not used, a corrupted SEMA4 could trigger continuous interrupts to both INTCs. To avoid this possible failure the INTC shall have the SEMA4 interrupt masked (for example, SEMA4 units have the lowest priority in the INTCs).

Mandatory: Application software shall keep these interrupt sources masked by programming the interrupt controller appropriately.

3.10 Enhanced Direct Memory Access (eDMA) requests

Mandatory: For ASIL D applications, protection against spurious or missing safety relevant eDMA requests must be implemented[d]. The methodology used to satisfy this requirement is application dependent.

Note: **Rationale:** *To manage spurious or missing eDMA transfer requests*

Note: **Implementation hint:** *Some implementations which can satisfy these requirements are:*
- Counting the number of eDMA transfers triggered inside a control period and compare this with what is the expected value.
- If the eDMA is used to manage the analog acquisition with the Cross-Triggering Unit (CTU) and ADC, the number of the converted ADC channels is saved in the CTU FIFO together with the acquired value. The eDMA transfers this value from the CTU FIFO to a respective SRAM location. Spurious or missing transfer requests can be detected by comparing the converted channel with what is expected.

Mandatory: Designers must not use the Periodic Interrupt Timer (PIT) module to trigger an eDMA transfer request for ASIL D applications.

Note: **Rationale:** *To avoid a faulty PIT (which is not redundant) from triggering an unexpected eDMA transfer*

3.11 Periodic Interrupt Timer (PIT)

Mandatory: For ASIL D applications the PIT module must be used in such a way that a possible failure is detected by the Software Watchdog Timer (SWT).

Note: **Rationale:** *To catch possible PIT failures*

Mandatory: If the PIT is used by ASIL D applications, a checksum of its configuration registers must be calculated and compared with the expected value to verify that the PIT configuration is correct.
Frequency: Once per FTTI

Note: **Rationale:** *To verify that the PIT remains at its expected configuration*

---

d.   eDMA is a replicated module. No software action is needed to detect faults inside this module.

3.12 Communication peripherals

The SPC56xL70xx includes the following communication peripherals:

● FlexCAN
● DSPI
● FlexRay
● LINFlexD

Recommended: An appropriate safety software protocol should be utilized (for example, Fault Tolerant Communication Layer, FTCOM) for any communication peripheral employed to meet ASIL D application requirements.

3.13 I/O peripherals

The following sections cover the use of the following peripherals:

● System Integration Unit Lite (SIUL)
● eTimer
● FlexPWM

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

● Read Inputs
    – Read Digital Inputs
    – Read PWM Inputs
    – Read Encoder Inputs
● Write Outputs
    – Write Digital Outputs
    – Write PWM Outputs

These are the safety functions assumed during analysis of the SPC56xL70xx.

3.13.1 Read Digital Inputs

For ASIL D applications, digital inputs used for safety purposes are assumed to be acquired redundantly as described in the following section.

Note: *Implementation hint: If sufficient diagnostic coverage can be obtained by a plausibility check on a single acquisition for a specific application, a plausibility check can replace a redundant acquisition. This hint is a special case of deviating from mandatory requirements as described in the Preface.*

Double Read Digital Inputs

Hardware elements

Double read operation of a digital input is implemented by two general purpose inputs (GPI) of the SIUL unit. SIUL must be configured to allow an input signal to be read from it's assigned pad. To minimize CCFs, the two input pads must not be physically adjacent (see *Section 7, I/O pin/ball configuration* for details).

Safety Integrity Functions

Mandatory: Safety integrity is achieved by replicated reading and software comparison by the processing function. The application shall implement the following tests:

- SIUL_SWTEST_REGCRC

Note: *Rationale: To verify that the configuration of the two pads used corresponds with the expected configuration, and to avoid a CCF caused by incorrectly configured pads*

- GPI_SWTEST_CMP

Note: *Rationale: To verify that the two input values compare*
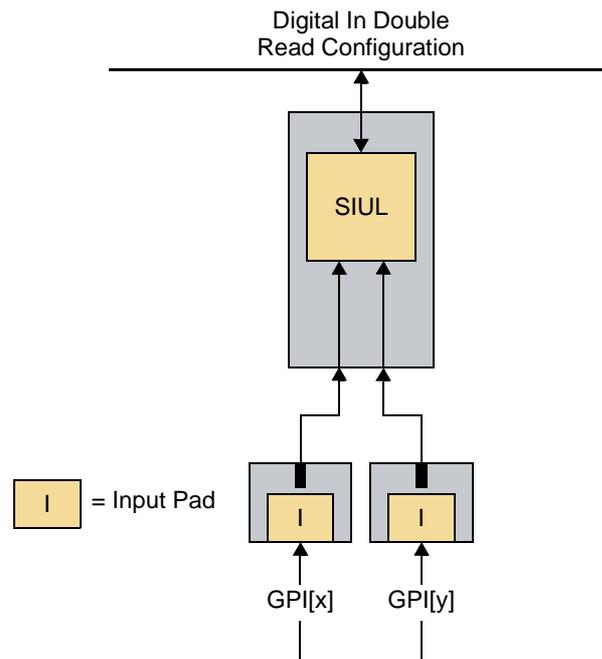
Figure 1.   Double Read Digital Input

Software test implementation

- SIUL_SWTEST_REGCRC

  The SIUL configuration registers are read, then a CRC is calculated. The CRC calculation is compared to the expected CRC value.

Note: *Implementation hint: The eDMA and CRC modules may be used to implement this Safety Integrity Function (SIF) to avoid overloading the CPU.*

- GPI_SWTEST_CMP

  This software test is used to execute the comparison between the double reads performed by the independent channels.

Implementation details

The only hardware element that can be used for the safety function is the general purpose input/output (GPIO).

Note: **Implementation hint:** *Every I/O pad that is not dedicated to a single function can be configured as GPIO (ADC pads are an exception to this rule, as they can only be configured as inputs).*

Caution: Redundant GPIO shall be selected in a non-contiguous way from the pin perspective to minimize CCF (see *Section 7, I/O pin/ball configuration* for details).

Mandatory: The pads shall be configured via the appropriate pad configuration registers (PCR*n*) in the SIUL module.

Note: **Rationale:** *To configure pads used by this safety function, and avoid CCF caused by improper configuration of the pads.*

Table 3. Software BIST and/or test

Software BIST or test	Frequency
SIUL_SWTEST_REGCRC	Once after programming
GPI_SWTEST_CMP	Once for every acquisition

3.13.2 Read PWM Input

For ASIL D applications, digital inputs used for safety purposes are always assumed to be acquired redundantly as described in the following section.

Read PWM Input means any input read related to signal transitions (rise or fall). This may also include the time that the signal was high, low or both.

Double Read PWM Inputs

Hardware elements

A Double Read PWM Input is implemented by two channels, one channel provided by eTimer_0 and the other by eTimer_1. The SIUL module must be configured (via the appropriate SIUL_PCR*n*) to provide configuration and input direction of the input pads. To minimize CCFs, these input pads must not be physically adjacent (see *Section 7, I/O pin/ball configuration* for details).

Safety Integrity Functions

Safety integrity is achieved by reading each input then comparing the values in the processing function (See *Figure 2*).

Mandatory: The software tests that the application must implement are:

● ETIMER0_SWTEST_REGCRC
● ETIMER1_SWTEST_REGCRC
● SIUL_SWTEST_REGCRC

Note: **Rationale:** *To verify that the configuration of the modules used by this safety function compare to the expected configuration*

Mandatory: In addition, the double reads must be compared by the application with the implementation of the following test: ETIMERI_SWTEST_CMP.

Note: **Rationale:** *To verify that the two sets of data compare*

Figure 2.   Double Read PWM Input

Software test implementation

● ETIMER0_SWTEST_REGCRC

The eTimer_0 configuration registers are read and a CRC checksum is computed. The checksum is compared with the expected value.

● ETIMER1_SWTEST_REGCRC

The eTimer_1 configuration registers are read and a CRC checksum is computed. The checksum is compared with the expected value.

● SIUL_SWTEST_REGCRC

The configuration registers of the SIUL are read and a CRC checksum is computed. The checksum is compared with the expected value.

Note: ***Implementation hint:*** *The eDMA and CRC modules should be used to implement these SIFs to avoid overloading the CPU.*

● ETIMERI_SWTEST_CMP

This software BIST is used to execute the comparison between the double reads performed by a channel on eTimer_0 and another channel on eTimer_1. The comparison must take into account possible approximation because of different capturing of the input asynchronous signals.

Implementation details

The following hardware elements shall be used for the safety function:

● eTimer_0 channels
● eTimer_1 channels

Mandatory: The user must select one channel from the eTimer_0 module and another from the eTimer_1.

Note: **Rationale:** *To avoid CCF (eTimer_0 and eTimer_1 belonging to different lakes)*

Mandatory: The pads shall be configured via the appropriate pad configuration registers (SIUL_PCR*n*).

Note: **Rationale:** *To configure pads used by this safety function*

Table 4.   Software BIST and/or test

Software BIST or test	Frequency
ETIMER0_SWTEST_REGCRC	Once after programming
ETIMER1_SWTEST_REGCRC	Once after programming
SIUL_SWTEST_REGCRC	Once after programming
ETIMERI_SWTEST_CMP	Once for every acquisition

3.13.3   Read Encoder Inputs

For ASIL D applications, encoder inputs used for safety purposes are assumed to be acquired redundantly as described in the following section.

Read Encoder Input means any input read related to signal transitions (rise or fall). This may also include signals coming from an encoder.

Double Read Encoder Inputs

Hardware elements

A Double Read Encoder Input is implemented using two channels that can be provided by:

● eTimer_0

● eTimer_1

● SIUL

When both channels are provided by the timer units, the signals of one encoder must be addressed to eTimer_0 and the signals of the other encoder must be addressed to eTimer_1. Alternatively, one or both channels can be provided by the SIUL, which supports interrupt based reading of encoder signals. This means the SIUL must use general purpose inputs which have edge detection interrupts (See *Figure 3* for details).

Mandatory: One channel must be addressed by eTimer_0, and the other by eTimer_1.

Note: **Rationale:** *Two different eTimers must be used to avoid CCF (eTimer_0 and eTimer_1 belonging to different lakes).*

For each signal, the SIUL can provide additional channels to support interrupt-based reading.

Mandatory: In this configuration, the SIUL must be correctly configured to forward one or two interrupt-based event readings.

Note: **Rationale:** *To configure pads used by this safety function*

Mandatory: The input pads must not be physically adjacent (see *Section 7, I/O pin/ball configuration* for details).

Note: ***Rationale:*** *To minimize CCF*

Safety Integrity Functions

The safety integrity is achieved by duplicate reads and software comparison by the processing function (See *Figure 3*).

Mandatory: The application software must implement the following tests:

● ETIMER0_SWTEST_REGCRC
● ETIMER1_SWTEST_REGCRC
● SIUL_SWTEST_REGCRC

Note: ***Rationale:*** *To verify that the configuration of the modules used by this safety function compare with what is expected*

                     Rationale: *To avoid CCF caused by improper configuration of the pads*

                     Mandatory: The application software must implement the test ENCI_SWTEST_CMP, which compares signals acquired from each channel.

Note: ***Rationale:*** *To verify that the two sets of data compare*
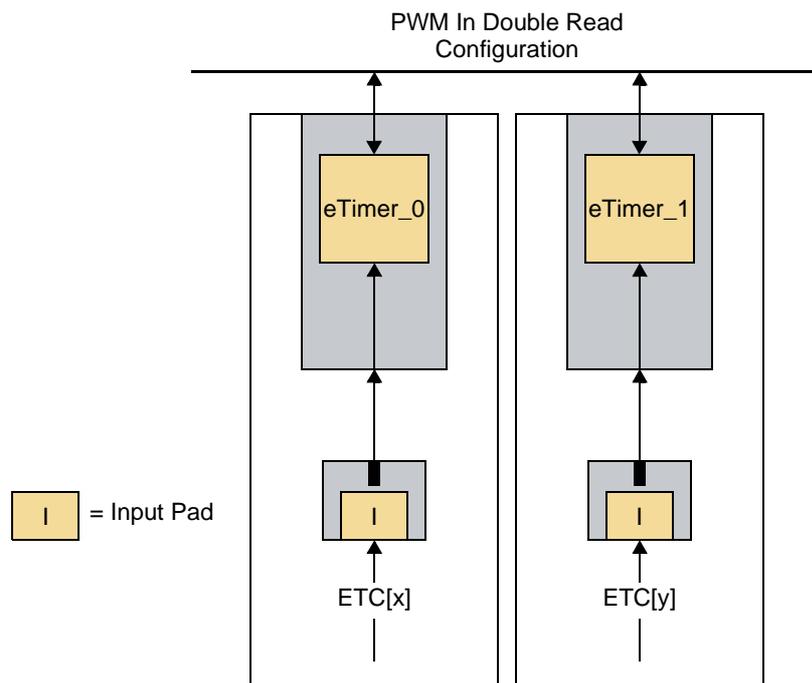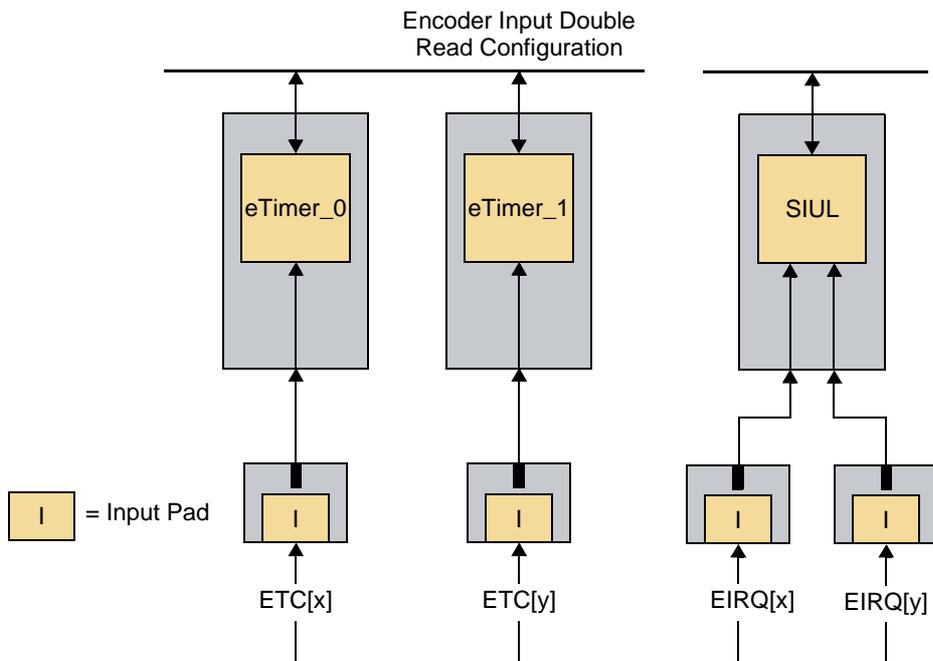

Figure 3.   **Double encoder read input**

Software test implementation

● ETIMER0_SWTEST_REGCRC

The eTimer_0 configuration registers are read, then a CRC checksum is computed. This computed checksum is compared to the expected value.

● ETIMER1_SWTEST_REGCRC

The eTimer_1 configuration registers are read, then a CRC checksum is computed. This computed checksum is compared to the expected value.

● SIUL_SWTEST_REGCRC

The configuration registers of the SIUL are read, then a CRC checksum is computed. This computed checksum is compared to the expected value.

Note: **Implementation hint:** *The eDMA and CRC modules should be used to implement this SIF to avoid overloading the CPU.*

● ENCI_SWTEST_CMP

This software test is used to execute the comparison between the double reads performed by one of the following:

– one channel on eTimer_0 and one channel on eTimer_1
– one channel on eTimer_1 and one channel on the SIUL
– one channel on eTimer_0 and one channel on the SIUL
– two channels on the SIUL

The comparison must take into account possible approximation because of different captured values of the input asynchronous signals and the execution of interrupt based event reads. Approximation required by different behavior of the encoded inputs must be handled at the application level.

Implementation details

The following hardware elements shall be used for the safety function:

● eTimer_0 channels
● eTimer_1 channels
● External interrupt via GPIO pins (configured via the SIUL)

The user must select one channel from eTimer_0 and one from eTimer_1. The external interrupt pins are optional.

Mandatory: The pads shall be configured via the appropriate pad configuration registers (SIUL_PCR*n*).

Note: **Rationale:** *To configure pads used by this safety function*

Table 5. Software BIST and/or test

Software BIST or test	Frequency
ETIMER0_SWTEST_REGCRC	Once after programming
ETIMER1_SWTEST_REGCRC	Once after programming
SIUL_SWTEST_REGCRC	Once after programming
ENCI_SWTEST_CMP	Once for every acquisition

3.13.4 Write Digital Outputs

For ASIL D applications, digital outputs used for safety purposes are assumed to be written either redundantly or with read back as described in the following section.

Note: ***Application-dependent option:*** *If a sufficient diagnostic coverage can be reached by a plausibility check on a single output channel for a specific application, a plausibility check can replace a redundant write or a direct read back.*

The element safety function Write Digital Out is implemented as either:

● Single Write Digital Out With Read Back
● Double Write Digital Out

Single Write Digital Outputs With Read Back

The SIUL hardware element is used to perform a single Write Digital Output With Read Back.

Mandatory: The read back must be implemented in one of the two modes shown in *Figure 4*.

Note: ***Rationale:*** *To verify if written data compares with the expected data*

Mandatory: The SIUL element must be correctly configured to provide the output write and the pad directions as follows:

● External read back – SIUL is configured to read back the signal from an additional pad, and the loopback is performed outside the device. In this configuration, only half of the available digital outputs are available as safety outputs.
● Internal read back[e] – SIUL is configured to read back the pad value via an internal read path. All pads dedicated to digital input/output are capable of reading the pad digital status using the input logic.

Note: ***Rationale:*** *To verify if written data is coherent with the expected data*

Mandatory: The application software must implement the software test to check the correct configuration of the pads, SIUL_SWTEST_REGCRC, and to compare the read back with the digital output write. GPOERB_SWTEST_CMP is used for external read back and GPOIRB_SWTEST_CMP is used for internal read back.

---

e. Internal read back does not cover package faults (e.g., wire bond, etc.).
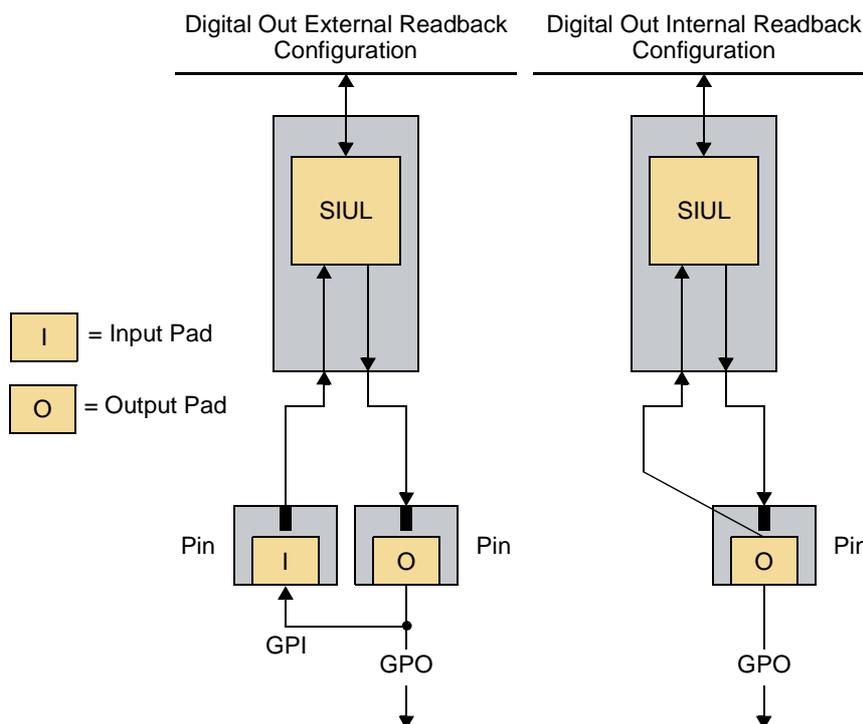
Figure 4.   Write Digital Output With Read Back

Software test implementation

● SIUL_SWTEST_REGCRC

The SIUL configuration registers are read and a CRC checksum is computed. This CRC checksum is compared what is expected.

Note: ***Rationale:*** *To avoid CCF caused by incorrect configuration of the pads*

Note: ***Implementation hint:*** *The eDMA and CRC modules should be used to implement this SIF to avoid overloading the CPU.*

● GPOERB_SWTEST_CMP

This software test is used to execute the comparison between the desired output values and the value read back via external read back configuration. After writing the output value, the test must read the value of the digital input.

Note: ***Rationale:*** *To verify if the read data compares with the written data*

● GPOIRB_SWTEST_CMP

This software test is used to execute the comparison between the desired output values and the value read back via internal read back configuration. After writing the output value, the test must read the status of the digital input.

Note: ***Rationale:*** *To verify if the read data compares with the written data*

Implementation details

The SIUL hardware element shall be used for the safety function. Every pad that is not dedicated to a single function can be configured as GPIO. Pads dedicated to ADC are an exception to this rule, as they can be configured as inputs only.

The pads shall be configured via the appropriate pad configuration registers (PCR*n*) in the SIUL module.

Table 6.   Software BIST and/or test

Software BIST or test	Frequency
SIUL_SWTEST_REGCRC	Once after programming
GPOERB_SWTEST_CMP	Once every write
GPOIRB_SWTEST_CMP	Once every write

Double Write Digital Outputs

The SIUL is used to perform a Double Write Digital Output.

Mandatory: The SIUL must be configured to correctly define the configuration of the output pads used. The software must perform a double write.

Note: ***Rationale:*** *To configure pads used by this safety function*

Mandatory: To guarantee the integrity of the two output channels, the application shall test the SIUL configuration implementing the SIUL_SWTEST_REGCRC.

Note: ***Rationale:*** *To avoid a CCF caused by incorrect configuration of the pads*

Mandatory: The application must implement the double output write as defined by the GPODW_SWAPP_WRITE.

Note: ***Rationale:*** *To write a digital output by exploiting redundancy*
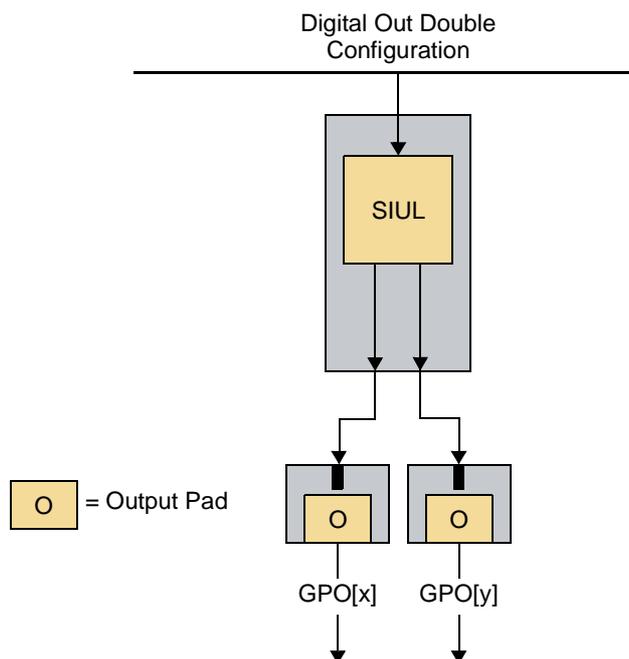
Figure 5.   Double Write Digital Output

Software test implementation

● SIUL_SWTEST_REGCRC

The configuration registers of the SIUL are read and a CRC is computed. This CRC value is compared with what is expected.

Note: ***Implementation hint:*** *The eDMA and CRC modules should be used to implement this SIF to avoid overloading the CPU.*

● GPODW_SWAPP_WRITE

Mandatory: The output write of a redundant channel must be implemented following this guideline:

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

Note: ***Rationale:*** *To minimize CCF of the SIUL*

Note: ***Implementation hint:*** *To write two or more GPIOs with a single instruction, the Masked Parallel GPIO Pad Data Out register (MPGPDOx) register can be used.*

Application software shall verify that the two GPIOs used are in the same MPGPDOx register.

To protect the value of the other GPIOs that belong to the same MPGPDOx, the MASK field of the MPGPDOx register needs to be properly configured.

Implementation details

The only hardware element that can be used for the safety function is the GPIO.

Note: *Every pad that is not dedicated to a single function can be configured as GPIO. ADCs are an exception to this rule, as they can be configured as inputs only.*

The pads shall be configured via the appropriate pad configuration registers (PCR*n*) in the SIUL module.

Table 7. Software BIST and/or test

Software BIST or test	Frequency
SIUL_SWTEST_REGCRC	Once after programming
GPODW_SWAPP_WRITE	Once every write

3.13.5 Write PWM Outputs

For ASIL D applications, PWM outputs used for safety purposes are assumed to be written either redundantly or with read back as described in the following section.

The element safety function Write PWM Output is implemented as Double Write PWM Outputs or Single Write PWM Outputs With Read Back.

Double Write PWM Outputs

The hardware elements eTimer_0 and eTimer_1 or FlexPWM_0 and FlexPWM_1 are used to perform a Double Write PWM Output.

Mandatory: These units must be configured to implement two PWM channels. The SIUL must be configured to define the configuration of the output pads used. The software must perform a double write.

Mandatory: Redundant pads must not be adjacent and pad configuration/data registers must be separate SIUL registers (see *Section 7, I/O pin/ball configuration* for details).

Note: **Rationale:** *To avoid CCF*

Mandatory: To guarantee the integrity of the two output channels, the application should test the SIUL configuration implementing the SIUL_SWTEST_REGCRC.

Note: **Rationale:** *To avoid CCF caused by incorrect configuration of the pads*

Mandatory: The application software must implement a test for the eTimer_0 and eTimer_1 configuration (ETIMER0_SWTEST_REGCRC, ETIMER1_SWTEST_REGCRC) or for the FlexPWM_0 and FlexPWM_1 configuration (FLEXPWM0_SWTEST_REGCRC, FLEXPWM1_SWTEST_REGCRC) and a software write (PWMDW_SWAPP_WRITE).

Note: **Rationale:** *To verify that the configuration of the modules used by this safety function adhere to the expected configuration*
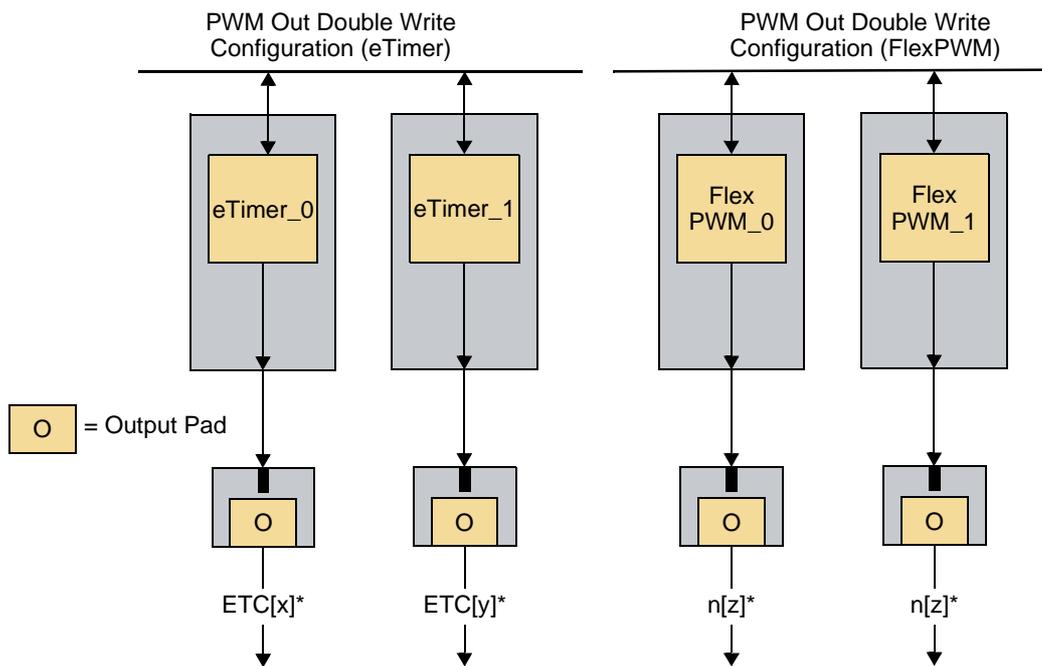
Note: n[z] represents any FlexPWM output (for example, A[z], B[z] or X[z]), but each output must be driven by different FlexPWM modules. The same consideration is valid for the eTimer; any eTimer output may be used, but each output must be driven by different eTimer module.

Figure 6.   Double Write PWM Output configuration

Software test implementation

● SIUL_SWTEST_REGCRC

The SIUL configuration registers are read and a CRC checksum is computed. The CRC checksum is compared to the expected value.

● ETIMER0_SWTEST_REGCRC

The eTimer_0 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

● ETIMER1_SWTEST_REGCRC

The eTimer_1 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

● FLEXPWM0_SWTEST_REGCRC

The FlexPWM_0 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

● FLEXPWM1_SWTEST_REGCRC

The FlexPWM_01 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

Note: *Implementation hint: The eDMA and CRC modules should be used to implement this SIF to avoid overloading the CPU.*

● PWMDW_SWAPP_WRITE

Mandatory: The output write of a redundant PWM channel must be implemented by writing the new output values to both the PWM channels. The customer can decide whether to use both eTimers (eTimer_0, eTimer_1) or both FlexPWMs (FlexPWM_0, FlexPWM_1), See *Figure 6*.

Note: **Rationale:** *To write a digital output by exploiting redundancy, and modules must belong to different lakes to decrease the probability of CCF*

Implementation details

The following hardware elements shall be used for the safety function:

● eTimer_0 channels
● eTimer_1 channels
● FlexPWM_0 channels
● FlexPWM_1 channels

Mandatory: The pads shall be configured via the appropriate pad configuration registers (PCR*n*) in the SIUL module.

Note: **Rationale:** *To configure pads used by this safety function*

Table 8. Software BIST and/or test

Software BIST or test	Frequency
SIUL_SWTEST_REGCRC	Once after programming[1]
ETIMER0_SWTEST_REGCRC[2]	Once after programming
ETIMER1_SWTEST_REGCRC[2]	Once after programming
FLEXPWM0_SWTEST_REGCRC[3]	Once after programming
FLEXPWM1_SWTEST_REGCRC[3]	Once after programming
PWMDW_SWAPP_WRITE	Once every write

1. If a change in a single SIUL configuration register is capable of affecting both the output and the read-back paths, then SIUL_SWTEST_REGCRC must be executed every FTTI. In all other cases configuration errors are covered by the software comparison.

2. This software BIST is needed only if the FlexPWM channels are used for the safety function

3. This software BIST is needed only if the eTimer channels are used for the safety function

Single Write PWM Outputs With Read Back

The hardware elements eTimer_0 and FlexPWM_1 or eTimer_1 and FlexPWM_0 are used to perform a Write PWM Output With Read Back[f]. These units must be configured to implement one PWM output channel and (via internal read back) the eTimer_0 input PWM channel. The SIUL must be configured to define the configuration of the output pads used. The software must perform a write operation followed by a read operation. To guarantee the integrity of the two output channels, the application should test the SIUL configuration implementing the SIUL_SWTEST_REGCRC (to avoid a common failure caused by misconfiguration of the pads).

---

f. eTimer_0 and FlexPWM_0 (eTimer_1 and FlexPWM_1) cannot be used in combination due to the same LBIST partition assignment.

Note: **Implementation hint**: *A single channel of the eTimer is used with a multiplexing of the internal read back of the different output of the FlexPWM. The read back paths are limited to six signals, two for each sub-module of the FlexPWM.*

Mandatory: The application software must implement software tests for eTimer_0 and eTimer_1 configurations:

● FLEXPWM0_SWTEST_REGCRC
● FLEXPWM1_SWTEST_REGCRC
● ETIMER0_SWTEST_REGCRC
● ETIMER1_SWTEST_REGCRC

Note: **Rationale:** *To verify that the configuration of the modules used by this safety function adheres to the expected configuration*

Mandatory: The application software must write to the output port and then compare the written value via the read back (See item *PWMRB_SWTEST_CMP* below).

Note: **Rationale:** *To verify that written data is what is expected*
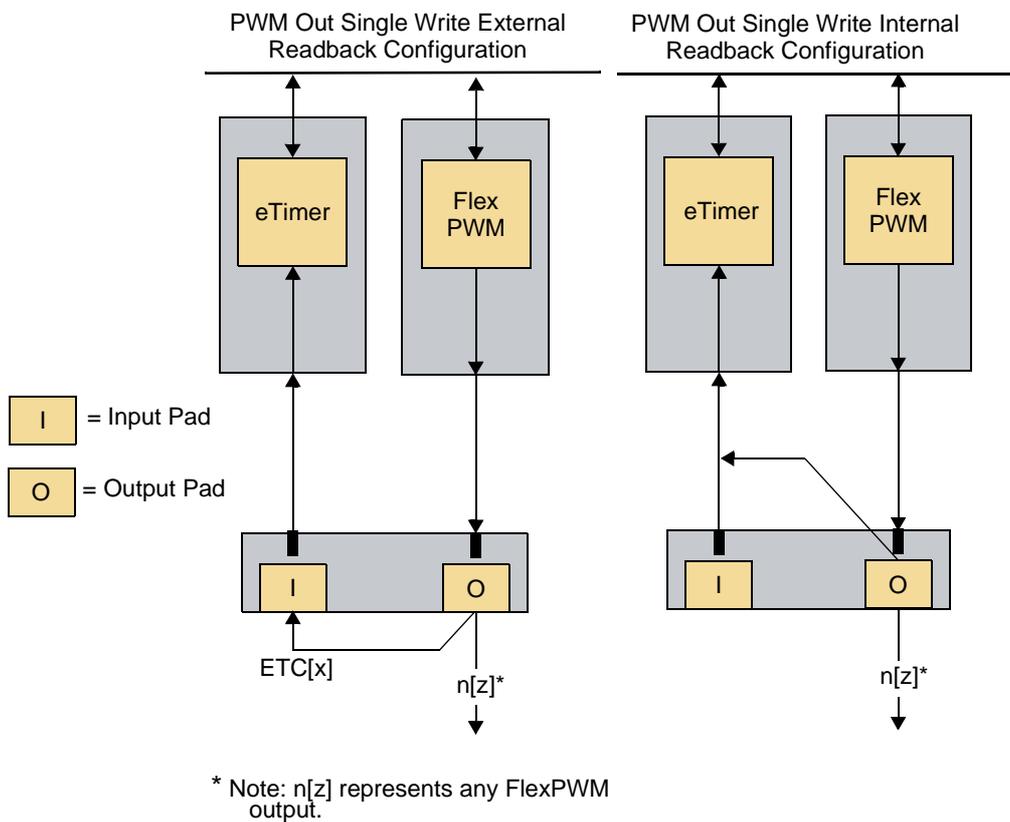
Figure 7. Single Write PWM Output With Read Back configuration

Software test implementation

- SIUL_SWTEST_REGCRC

  The SIUL configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

- ETIMER0_SWTEST_REGCRC

  The eTimer_0 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

- ETIMER1_SWTEST_REGCRC

  The eTimer_1 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

- FLEXPWM0_SWTEST_REGCRC

  The FlexPWM_0 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

- FLEXPWM1_SWTEST_REGCRC

  The FlexPWM_1 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

Note: **Implementation hint:** *The eDMA and CRC modules should be used to implement this SIF to avoid overloading the CPU.*

- PWMRB_SWTEST_CMP

  This procedure output compares the PWM read back provided by a single channel of the eTimer_0 (eTimer_1) with the expected values that have been written to the FlexPWM_1 (FlexPWM_0) output channel.

Implementation details

The following hardware elements shall be used for the safety function:

- eTimer_0 channels
- eTimer_1 channels
- FlexPWM_0 channels
- FlexPWM_1 channels

Mandatory: The pads shall be configured via the appropriate pad configuration registers (PCR*n*) in the SIUL module.

Note: **Rationale:** *To configure pads used by this safety function*

Table 9.   Software BIST and/or test

Software BIST or test	Frequency
SIUL_SWTEST_REGCRC	Once after programming
ETIMER0_SWTEST_REGCRC[1]	Once after programming
ETIMER1_SWTEST_REGCRC[1]	Once after programming
FLEXPWM0_SWTEST_REGCRC[2]	Once after programming
FLEXPWM1_SWTEST_REGCRC[2]	Once after programming
PWMRB_SWTEST_CMP	Once every write

1. This software BIST is needed only if the eTimer channels are used for the safety function.

2. This software BIST is needed only if the FlexPWM channels are used for the safety function.

3.13.6   Other requirements for I/O peripherals

Mandatory: Other requirements related to I/O peripherals include the following:

● In the eTimer module, the capture flag (eTimer_n_STS[ICFn]) must be used.

Note: ***Rationale:*** *To detect missing eTimer_n acquisition*

● If the eTimer counter is used to decode a primary and secondary external input as quadrature encoded signals, the eTimer watchdog must be used (See "Counting Modes" section of the SPC56xL70xx Reference Manual).

Note: ***Rationale:*** *To detect stalled quadrature counting*

3.14   Cross Triggering Unit (CTU)

The CTU generates some triggers based on input events (FlexPWMs, eTimers, and/or external pins).

The trigger can be caused by:

● A pulse

● An interrupt

● An ADC command (or a stream of consecutive commands)

● All of these

Mandatory: The CTU shall be appropriately configured so that the output triggers are generated within the desired time schedule with respect to the input event(s).

Note: ***Rationale:*** *To avoid erratic output trigger generation*

For each trigger, a set of ADC commands and pulses to be generated can be defined.

If the application safety function includes the read of some inputs synchronized with some events (FlexPWMs, eTimers, and/or external pins), the customer can use the CTU module for this purpose. The software needed for targeting the ASIL D is listed in *Section 3.14.1, Synchronize Sequential Read Input*.

For a detailed description on how the CTU works (triggered and sequential mode), its configuration and use, refer to the SPC56xL70xx Reference Manual.

3.14.1 Synchronize Sequential Read Input

The CTU can be used if the customer needs to synchronize the reading of some inputs with some events (FlexPWMs, eTimers, and/or external pins).

Mandatory: If this function is part of the application safety function, the safety integrity is achieved by a mix of hardware mechanisms and software safety integrity functions implemented at the application level:

● CTU_HWSWTEST_TRIGGERNUM
● CTU_SWTEST_TRIGGERTIME
● CTU_HWSWTEST_TRIGGEROVERRUN
● CTU_HWSWTEST_ADCCOMMAND (only if the input is an analog signal)
● CTU_SWTEST_ETIMERCOMMAND
● CTU_HW_CFGINTEGRITY

Note: *These functions are mandatory only if the CTU is used.*

Software test implementation

● CTU_HWSWTEST_TRIGGERNUM

If the reload signal occurs before all the triggers are generated, an overrun indication is flagged and the application software must handle the error indication.

Note: ***Rationale:*** *Tests if all the triggers configured within a control period have been generated and serviced.*

Note: ***Implementation hint:*** *The Cross Triggering Unit Error Flag register (CTUEFR) shows information about the overrun status.*

● CTU_SWTEST_TRIGGERTIME

Application software must configure one eTimer channel to capture the time at which each trigger event occurs.

In triggered mode, the time instant of each trigger within one control period is captured and stored in a FIFO. Application software has to check the FIFO values against the expected ones according to CTU configuration.

In sequential mode, one eTimer channel is needed to check the correct time of a single trigger with respect to the corresponding event.

Note: ***Rationale:*** *To verify if triggers are generated at the correct time*

Note: ***Implementation hint:*** *Some eTimer inputs are internally connected to the CTU output. eTIMER_2 input/outputs are not connected to pins on LQFP144 package. Use eTIMER_2 channels for implementing this safety function to keep the channels from eTIMER_0 or eTIMER_1 units for functions using port pins (See "Enhanced Motor Control Timer (eTimer)" in the SPC56xL70xx Reference Manual for details).*

Note: ***Implementation hint:*** *eTimer capture register implements a two entry FIFO, but in CTU triggered mode up to 8 time values need to be stored. To avoid FIFO overflow condition,*

eTimer can be configured to trigger a eDMA transfer to move the captured value to specific RAM location.

● CTU_HWSWTEST_TRIGGEROVERRUN

This hardware mechanism checks if a new trigger occurs that requires an action by a subunit that is currently busy. In this case, an overrun interrupt is generated and the application software must handle the error condition.

Over-run detection mechanism shall be enabled by software during CTU configuration.

Note: **Rationale:** *Checks if a new trigger occurs that requires an action by a subunit (such as ADC command generator) which is currently busy.*

Note: **Implementation hint:** *To enable the over-run detection the IEE flag in the Cross Triggering Unit Interrupt/eDMA register (CTUIR) register shall be asserted. This interrupt is shared between several sources of error. The user can discriminate among them by reading the CTUEFR register.*

● CTU_HWSWTEST_ADCCOMMAND

The CTU stores in its internal FIFOs both the value provided by each ADC conversion and the channel number. Application software must check the ADC channel number sequence against what is expected for each FIFO. Moreover, invalid commands issued by the CTU are flagged and the corresponding error must be handled by the application software.

Note: **Rationale:** *To detect if the incorrect channel has been acquired, or if the incorrect ADC result FIFO is selected*

Note: **Implementation hint:** *To enable invalid command detection, the IEE flag in the CTUIR register must be asserted.*

This interrupt is shared between several sources of error. The user can discriminate among them by reading the CTUEFR register.

This safety integrity function needs to be implemented only when reading analog signals.

● CTU_SWTEST_ETIMERCOMMAND

Application software must configure one channel of eTimer_0 or eTimer_1 to count the number of eTimer commands generated within a CTU control period and must check the number against the expected one.

Note: **Rationale:** *To verify the correctness of the number of generated commands*

Note: **Implementation hint:** *Some eTimer inputs are internally connected to the CTU output (See the SPC56xL70xx Reference Manual for details).*

● CTU_HW_CFGINTEGRITY

This hardware mechanism ensures the consistency of the CTU configuration at the beginning of each CTU control period.

The configuration registers are all double-buffered. If the configuration is only partial when the control period starts, the previous configuration is used and an error condition is flagged, which must be handled by the application software.

Note: **Rationale:** *Ensures the consistency of the CTU configuration*

Note: **Implementation hint:** *The CTU uses a safe reload mechanism. The General Reload Enable (GRE) bit in the Cross Triggering Unit Control Register (CTUCR) shall be used to detect partial or incomplete CTU update.*

To enable the interrupt in case of error during reload, the IEE flag in the CTUIR register shall be asserted.

This interrupt is shared between several sources of error. The user can discriminate among them by reading the CTUEFR register.

Implementation details

The following hardware elements shall be used for the safety function:
● CTU
● One eTimer channel

Table 10.   Software BIST and/or test

Software BIST or test	Frequency
CTU_HWSWTEST_TRIGGERNUM	Once for every control period (< FTTI)
CTU_SWTEST_TRIGGERTIME	Once for every CTU control period (triggered mode) or every trigger (sequential mode)
CTU_HWSWTEST_TRIGGEROVERRUN	Once for every trigger
CTU_HWSWTEST_ADCCOMMAND	Once for every ADC command
CTU_SWTEST_ETIMERCOMMAND	Once for every control period (< FTTI)
CTU_HW_CFGINTEGRITY	Once for every control period (< FTTI)

Other requirements for CTU module usage

Mandatory: The only other requirement related to the CTU is that if the CTU is used to read an analog signal through the ADC, the software shall verify the Invalid Command Error flag (CTU_CTUEFR[ICR]) after programming the ADC command lists.

Note: **Rationale:** *To check the presence of invalid commands*

3.15   ADC

If the ADC is used in a safety function, the following sections must be observed if an ADC BIST is to be performed.

It is important to note that the ADC is part of the temperature measuring safety integrity function, and it is therefore required that the HWBIST functions be executed once after the boot even if the ADC is not in application use.

3.15.1   Read Analog Inputs

The customer has two options for reading analog inputs:
● Single Read Analog Inputs
● Double Read Analog Inputs

Single Read Analog Inputs

Hardware elements

The single-read analog input uses a single-analog-input channel either of ADC_0 or ADC_1 to acquire an analog voltage signal (See *Figure 8*).

To support a high diagnostic coverage two known reference supply voltages are utilized by two software tests which are described in the following sections (ADC_SWTEST_TEST1 and ADC_SWTEST_TEST2).

The reference supply voltages are the following:

- $V_{DD\_HV\_ADR0}$ (ADC_0 high reference voltage)
- $V_{DD\_HV\_ADR1}$ (ADC_1 high reference voltage)
- $V_{SS\_HV\_ADR0}$ (ADC_0 low reference voltage)
- $V_{SS\_HV\_ADR1}$ (ADC_1 low reference voltage)

The SIUL unit must be configured properly to correctly enable the input pads. The pads used for analog inputs are only of type INPUTS.

Safety Integrity Functions

Mandatory: The safety integrity is achieved by dedicated hardware BIST[g]:

Note: **Rationale:** *Hardware BIST to check the integrity of the ADC, both analog and digital parts:*

- SUPPLY SELF-TESTRESISTIVE-CAPACITIVE SELF-TESTCAPACITIVE SELF-TEST

Mandatory: By dedicated software test implemented at the application level:

- ADC_SWTEST_TEST1
- ADC_SWTEST_TEST2
- ADC_SWTEST_VALCHK
- ADC0_SWTEST_REGCRC
- ADC1_SWTEST_REGCRC
- SIUL_SWTEST_REGCRC
- ADC_SWTEST_ADCOVERSAMPLING

---

g. These hardware BISTs need some software to activate them. This software shall be developed by the customer.

Analog Single Read/Write Internal
Readback Configuration

ADC_x

Reference voltages
(V_{dd}_HV_ADRx and V_{ss}_HV_ADR)

I   = Input Pad

I

AN[x]
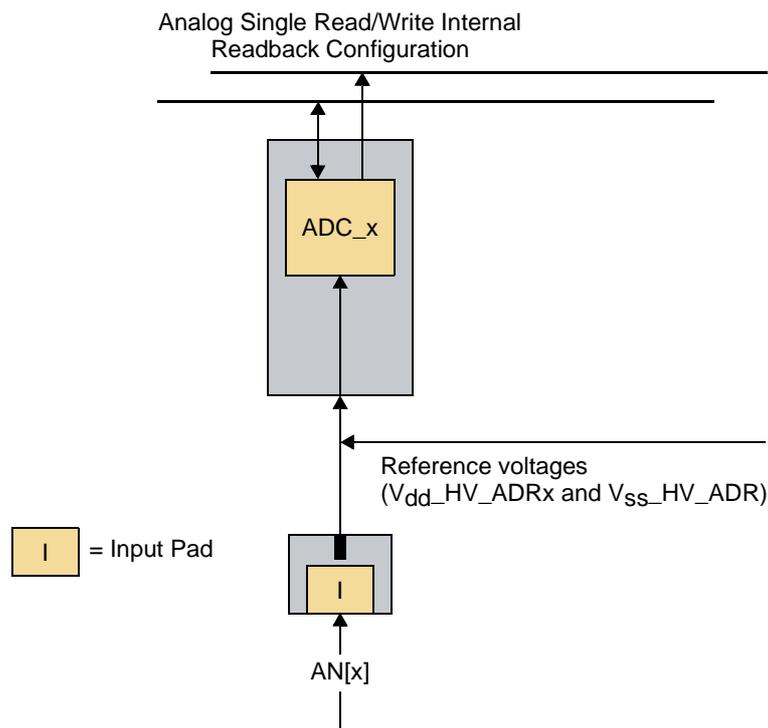
Figure 8.   Single Read Analog Input configuration

Hardware BIST

Three types of self-test algorithms have been implemented in the ADC hardware:

● SUPPLY SELF-TEST

● RESISTIVE-CAPACITIVE SELF-TEST

● CAPACITIVE SELF-TEST

Hardware BIST implementationThe hardware BISTs shall be activated by the application in one of the following modes:

● CPU mode

● CTU mode

In CPU mode, the application software takes care of the hardware self-test activation and checks the test flow and the timing.

In CTU mode, the CTU module takes care of the hardware self-test activation, flow monitoring, and timing. It is important to note that in this operating mode, the CPU does not take part in running the hardware self-test.

HW self-tests use analog watchdogs to verify the outcome of self-test conversions. The reference thresholds of these watchdogs are saved in test sector (See "Test flash memory" section and "Test flash information" table in the SPC56xL70xx Reference Manual).

Mandatory: Before running the HW self-test, the customer must copy these thresholds from the test sector into the watchdog registers (See "Self test analog watchdog" section of the SPC56xL70xx Reference Manual).

Note: **Rationale:** *To set the correct threshold for the self-tests*

Note: **Implementation hint:** *Since user can not directly read the test sector an SSCM feature, called Test Flash Enable, shall be exploited. This action is performed through the following steps:*

1. If code is executing in flash memory, it jumps to execute from RAM.
2. Write SSCM_SCTR[TFE] = 1.
3. Test sector is readable at the offset 0x0 of the flash memory address space (See "System Status and Configuration Module (SSCM)" of the Reference Manual).
4. Thresholds are copied from the test sector to the respective register.
5. Write SSCM_SCTR[TFE] = 0.
6. Code can continue execution from the flash memory.

BAM implements an access method to read the test sector.

Mandatory: Since the BAM is not developed according to the safety standard, a safety application is not allowed to read the test sector through the BAM access method.

Additionally, a watchdog timer is implemented to check the sequence of the self-test algorithms.

Mandatory: The customer must enable the watchdog timer for CPU mode and CTU mode. The programmable watchdog timeout is the FTTI[h].

Note: **Rationale:** *To check the sequence of the self-test algorithms*

Every hardware BIST is activated via a dedicated command sent to the ADC. Refer to the "Self-testing" section in the "ADC" chapter of the SPC56xL70xx Reference Manual to have all detailed instructions for implementing one of these modes.

The supply self-test must be executed without interleaved user conversion.

Software tests

● ADC_SWTEST_TEST1

This software BIST exploits the presampling feature of the ADC. Presampling allows to precharge or discharge the ADC internal capacitor before it starts the sampling and conversion phases of the analog input coming from pads. During presampling phase, the ADC samples the internally generated voltage while in the sampling phase the ADC samples analog input coming from pads (See *Figure*).

Reference voltage which can be used during presampling phase is either $V_{DD\_HV\_ADR0/1}$ or $V_{SS\_HV\_ADR0/1}$.

If there is an open failure in the analog multiplexing circuitry, the signal converted by the ADC is not the analog input coming from the pad, but the presampling reference voltage ($V_{DD\_HV\_ADR0/1}$ or $V_{SS\_HV\_ADR0/1}$).

This BIST must be run for each analog input used by the safety function.

Since the pads dedicated to analog inputs are of type INPUT, a missing enable from the SIUL results in an open failure.

Note: **Rationale:** *To detect open failures of the channel multiplexing circuitry (See Figure 9)*

---

h. This action is not mandatory in case of Double Read Analog Inputs.
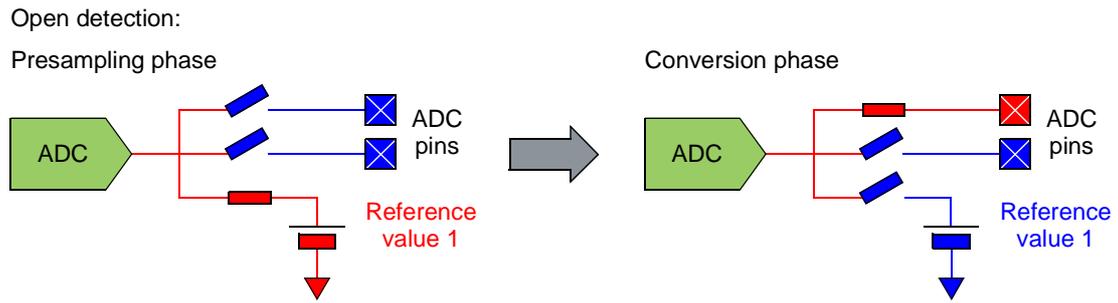
Figure 9.   **Software BISTs to test the multiplexing circuitry (ADC_SWTEST_TEST1)**

Note: **Implementation hint**: *Presampling can be enabled on a per channel basis through the ADC_x_PSR0 register.*

          ADC_x_PSCR[PREVAL0] selects which reference voltage is used to precharge/discharge the ADC internal capacitor.

          ADC_x_PSCR[PRECONV] register shall be 0 (See "Analog-to-Digital Converter (ADC)" chapter in the SPC56xL70xx Reference Manual for details on the presampling feature).

Note: Either $V_{DD\_HV\_ADR0/1}$ or $V_{SS\_HV\_ADR0/1}$ can be used as presampling voltage.
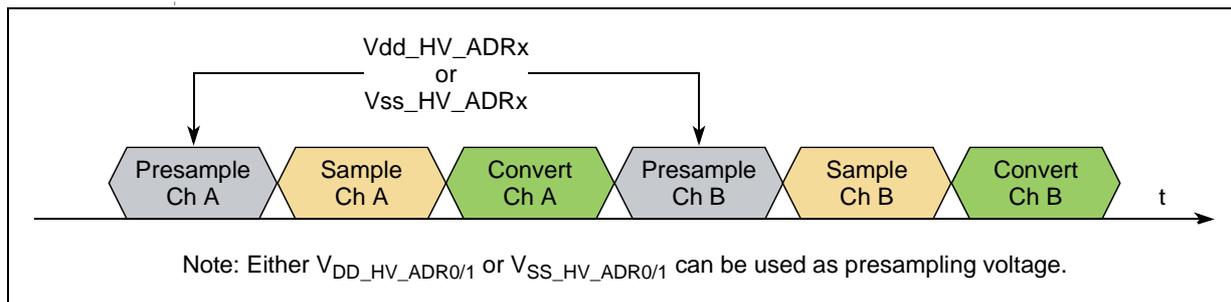
Figure 10. Implementation of ADC_SW_TEST1 through the ADC presample feature

● ADC_SWTEST_TEST2

To detect short failures two different voltages are acquired by the ADC. If these values are different from the expected ones, a short failure on the multiplexed circuitry has been detected.

To implement this test a presampling feature of the ADC can be exploited. The presampling must be configured in such a way that the sampling of the channel is bypassed and the presampling reference supply voltages are converted.

During the first step the $V_{DD\_HV\_ADR0/1}$ is converted and compared with the its expected value, then the $V_{SS\_HV\_ADR0/1}$ is converted and compared with its expected value (See *Figure 12*).

Figure 12 includes the conversion of the 2 different presampling reference voltages ($V_{DD\_HV\_ADR0/1}$ and $V_{SS\_VH\_ADR0/1}$).

Note: **Rationale:** *To detect short failures of the channel multiplexing circuitry (See Figure 11)*
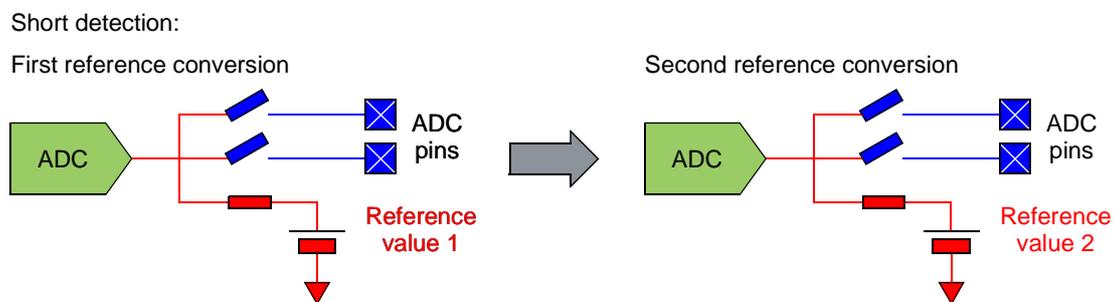
Figure 11. Software BISTs to test the multiplexing circuitry (ADC_SWTEST_TEST2)

Note: **Implementation hint**: *The implementation hints of the ADC_SWTEST_TEST1 applies also to the ADC_SWTEST_TEST2*

To bypass the conversion of the input channel and convert the presampled values, ADC_x_PSCR[PRECONV] register shall be set to 1.

Note: Either $V_{DD\_HV\_ADR0/1}$ or $V_{SS\_HV\_ADR0/1}$ can be used as presampling voltage.
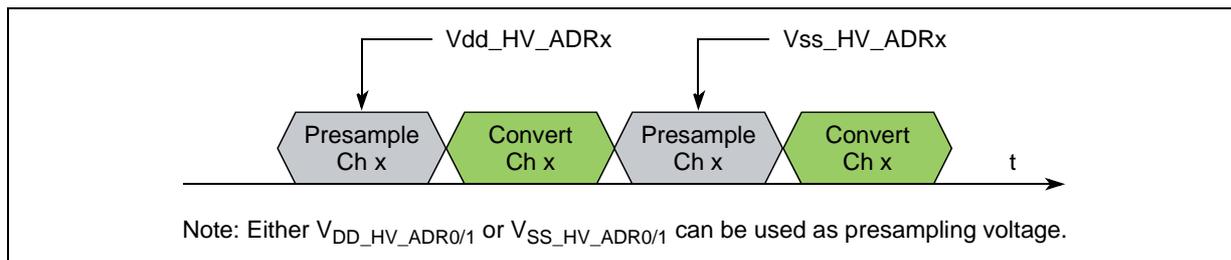
Figure 12.   Implementation of ADC_SW_TEST2 through the ADC presample feature

ADC_SWTEST_VALCHK The goal of this software test is to verify correct operation of the control and queue logic of the ADC, and also the CTU, if used. This software measures implementation is dependant on the ADC configuration (for example, CTU or CPU mode):

When the ADC is used in CPU mode, the acquired value is read by the ADC_CDRn. This register includes ADC_CDRn[VALID] and ADC_CDRn[RESULT] fields as well as channel n converted data (ADC_CDRn[CDATA]). These fields provide status information about the data acquisition. Application software shall read and verify these fields after every acquisition.

When the ADC conversion is triggered by the CTU, the acquired digital sample data are stored in a dual queue along with information about the channel that performed the acquisition. Checking the channel information of the acquisition provides sufficient coverage of the control logic and, in part, the queue logic.

Note: ***Implementation hint:*** *If ADC is configured to work in CTU mode, the conversion results are stored in CTU FIFOs (See "Cross-Triggering Unit (CTU)" chapter in SPC56xL70xx Reference Manual). Along with the converted data, the converted channel number and ADC module are stored. CTU includes two sets of registers to read this information (FIFO Right aligned data, CTU_FRx, and FIFO Left aligned data, CTU_FLx). User must read these registers to verify if the sequence of the acquired channel is what is expected.*

● ADC_SWTEST_OVERSAMPLING

In case of Single Read Analog Inputs the ADC_SWTEST_ADCOVERSAMPLING_CMP must be implemented as counter measure against random fault.

ADC_SWTEST_OVERSAMPLING is an acquisition redundant in time.

It refers to sampling the signal at rate significantly higher than the Nyquist Frequency related to the input signal. In case of fault the acquired values are not correlated with themselves.

This SIF compares the acquired value to verify the correlation.

Against random fault, three consecutive analog values are converted for each acquisition to implement the ADC_SWTEST_OVERSAMPLING The second

acquisition, A_2, is faulty because the first converted value is quite different respect the other two (See *Figure 13*).

Faulty Acquisition
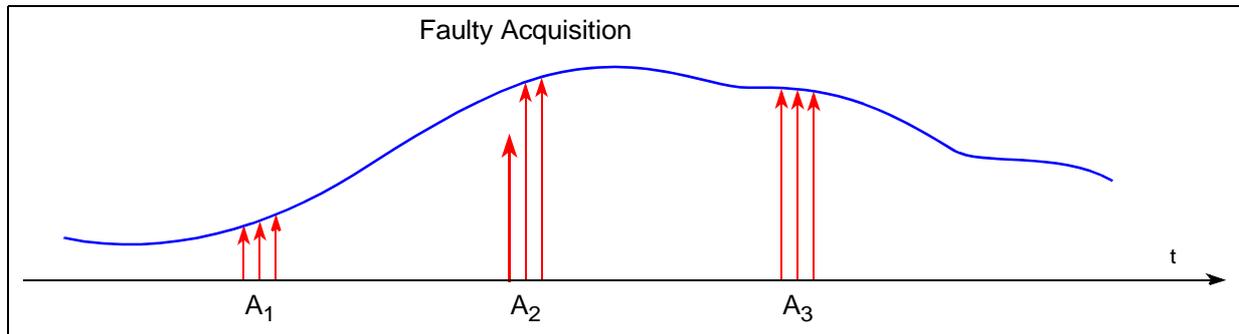
Figure 13. Series of acquired analog values

● ADC0_SWTEST_REGCRC

If ADC_0 is used, the ADC_0 configuration registers are read and CRC checksum is computed. The checksum is compared to the expected value.

● ADC1_SWTEST_REGCRC

If ADC_1 is used, the ADC_1 configuration registers are read and CRC checksum is computed. The checksum is compared to the expected value.

● SIUL_SWTEST_REGCRC

The SIUL configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

Implementation details

The following hardware elements shall be used for the safety function:

● Analog input channels AN[0:8] of ADC_0

● Analog input channels AN[11:14] of ADC_0 and ADC_1 (shared channels)

● Analog input channels AN[0:8] of ADC_1

The user must select one channel from ADC_0 or from ADC_1. Shared channels can be used.

Mandatory: The input pads are configured via the appropriate pad configuration registers (PCR*n*) in the SIUL module.

Table 11. Software BIST and/or test

Software BIST and/or test	Frequency
SUPPLY SELF-TEST	Once in the FTTI
RESISTIVE-CAPACITIVE SELF-TEST	Once in the FTTI
CAPACITIVE SELF-TEST	Once in the FTTI
ADC_SWTEST_TEST1	Once in the FTTI
ADC_SWTEST_TEST2	Once in the FTTI
ADC_SWTEST_VALCHK	Once for every acquisition
ADC_SWTEST_OVERSAMPLING	Once for every acquisition

Table 11.   Software BIST and/or test (continued)

Software BIST and/or test	Frequency
ADC0_SWTEST_REGCRC	Once in the FTTI
ADC1_SWTEST_REGCRC	Once in the FTTI
SIUL_SWTEST_REGCRC	Once in the FTTI

Double Read Analog Inputs

Hardware elements

The Double Read Analog Input uses two analog input channels to acquire a replicated analog input signal. Both ADC units acquire and digitize the two copies of a redundant analog signal connected to the inputs. In this configuration (if applied to all possible analog inputs), only half of the analog inputs are available to the applications (AN[0:8] of ADC_0 for signals, and AN[0:8] of ADC_1 for signal copies).

Mandatory: The shared channels (AN[11:14]) suffer from CCF because they share pads between each ADC module. Therefore, they are omitted (considered not safe) for double reads. The comparison of the results is performed by application software (See *Figure 14*).

Note: **Rationale:** *ADC_0 and ADC_1 share a pad for the channels (AN[11:14]). Omitting them from double read eliminates a possible source of CCF.*

Mandatory: After boot but before executing the safety function the following tests shall be executed to detect latent faults (See *Section , Hardware BIST* and *Section , Hardware BIST implementation*):

- SUPPLY SELF-TEST
- RESISTIVE-CAPACITIVE SELF-TEST
- CAPACITIVE SELF-TEST

Note: **Rationale:** *To check the integrity of the ADC modules*

Mandatory: Before running the HW self-test, the customer must copy the threshold values of the analog watchdogs from test sector into the watchdog registers (See "Self test analog watchdog" section of the "Analog-to-Digital Converter (ADC)" chapter in SPC56xL70xx Reference Manual).

Note: **Rationale:** *To set the correct threshold for the self-test*

Safety Integrity Functions

Safety integrity is achieved by replicated acquisition with separate analog input channels and software comparison by the processing function (See *Figure 14*).

Mandatory: The following software test must be implemented by the application software:
ADC0_SWTEST_REGCRC, ADC1_SWTEST_REGCRC, SIUL_SWTEST_REGCRC

Note: **Rationale:** *To verify that the configuration of the module used by this safety function corresponds with what is expected*

Note: **Rationale:** *To avoid CCF caused by improper configuration of the pads*

Mandatory: In addition, the software test ADC_SWTEST_CMP must be implemented to compare the channel reads.

Note: ***Rationale:*** *To verify that the two sets of read data compare*

It is important to note that this safety integrity function might be applied in addition to Single Analog Read Inputs, which increases diagnostic coverage.
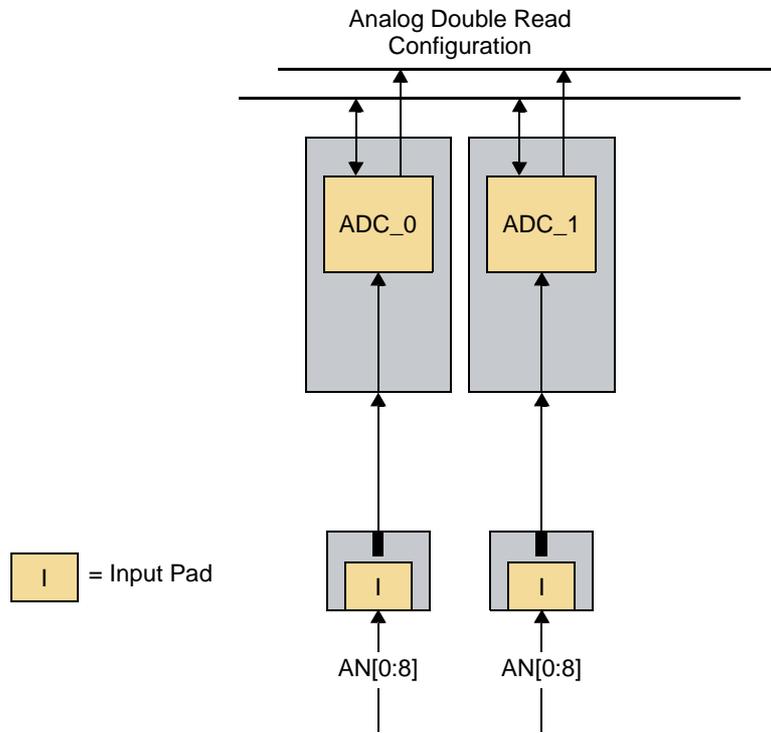
Figure 14.   Double Read Analog Inputs configuration

Software test implementation

● ADC0_SWTEST_REGCRC

The ADC_0 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

● ADC1_SWTEST_REGCRC

The ADC_1 configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

● SIUL_SWTEST_REGCRC

The SIUL configuration registers are read and a CRC checksum is computed. The checksum is compared to the expected value.

● ADC_SWTEST_CMP

This software test is used to execute a comparison between the double acquisition performed by one channel of ADC_0 and one channel of ADC_1. The comparison must be approximated because of conversion differences.

Implementation details

The following hardware elements shall be used for the safety function:

● Analog input channels AN[0:8] of ADC_0
● Analog input channels AN[0:8] of ADC_1

The user must select one channel from ADC_0 and one from ADC_1.

The input pads are configured via the appropriate pad configuration registers, SIUL_PCR*n*.

Table 12.   Software BIST and/or test

Software BIST or test	Frequency
SUPPLY SELF-TEST	Once after boot
RESISTIVE-CAPACITIVE SELF-TEST	Once after boot
CAPACITIVE SELF-TEST	Once after boot
ADC0_SWTEST_REGCRC	Once after programming
ADC1_SWTEST_REGCRC	Once after progamming
SIUL_SWTEST_REGCRC	Once after progamming
ADC_SWTEST_CMP	Once for every acquisition

3.15.2   Other requirements

Other requirements related to the ADC modules are:

● When an application needs to access the ADC result FIFO, a 32-bit read access shall be performed to verify the channel number on which the conversion has been executed.
● If the ADC analog watchdog function is used for function-safety relevant signal, two analog watchdog channels must monitor the same signal.
● If the Sine Wave Generator (SWG) is used, the ADC (in conjunction with CTU) must be used to check the output signal.

3.16   Temperature sensors

There are two temperature sensors: temperature sensor 0 (TSENS_0) mapped to ADC_0 and temperature sensor 1 (TSENS_1) mapped to ADC_1.

Mandatory: During power up, the two temperature sensors need to be read by software (TSENS_0 from ADC_0 channel 15, TSENS_1 from ADC_1 channel 15), which must verify that the read values are similar as a means of assessing the functionality of the sensors. However, nothing prohibits reading the temperature sensors during run time if needed.

Note: **Rationale:** *A means of assessing functionality of the temperature sensors*

Mandatory: In addition, the temperature must be acquired from at least one of the temperature sensors by software every FTTI during run time. In case of a fault, software must move the system to a safe state.

Note: **Rationale:** *To detect over-temperature faults*

To set a proper threshold the customer must consider that the maximum operating junction temperature is 150 °C (See the SPC56xL70xx data sheet) and the temperature sensor accuracy is 10° C.

Note: **Implementation hint:** *See the SPC56xL70xx Reference Manual for details on TSENS_x implementation in relation to the ADC.*

It is important to note that the ADC is part of the temperature measuring safety integrity function. Therefore, it is required that the BIST of the ADC be executed once after boot even if the ADC is not used by the application.

3.17 Software Watchdog Timer (SWT)

Mandatory: These requirements apply to the SWT for ASIL D applications:

● Both of the following must be true:
  – The SWT is enabled
  – Configuration registers hard locked to avoid unwanted modification
● The SWT time window settings must be set to a value less than the FTTI. Detection latency shall be smaller than FTTI.
● Before the safety function is executed, software must verify that the SWT is enabled by reading the SWT control register (SWT_CR[WEN] = 1).

Note: **Rationale:** *To detect a defective program sequence*

Mandatory: Control flow monitoring can be implemented by SWT. However, other control flow monitoring approaches that do not used the SWT may also be used.

SPC56xL70xx provides the hardware support (SWT) to implement both control flow and temporal monitoring methods. Refer to the SPC56xL70xx Reference Manual for the SWT functional description.

Note: **Implementation hint:** *To enable the SWT, and to hard lock the configuration register, SWT_CR[WEN] and SWT_CR[HLK] must be asserted (= 1).*

The timeout register (SWT_TO) must contain a 32-bit value that represents a timeout less than the FTTI.

If Windowed mode and Keyed Service mode (two pseudorandom key values used to service the watchdog) are enabled, it is possible to reach a high effective temporal flow monitoring.

3.18 Redundancy Control Checking Unit (RCCU)

The task of the RCCU unit is to perform a cycle-by-cycle comparison of the outputs of the modules included in the SoR. The SoR is the logical part of the device that contains all the modules that are replicated for functional safety reasons.

The RCCU is able to detect any mismatch between the outputs of two replicated modules. The error information is forwarded to the MC_RGM and FCCU.

For ASIL D applications, use of the RCCU is indispensable. The use of RCCU's is automatically managed by the SPC56xL70xx device, users cannot disable the RCCU.

Note: **Rationale:** *To catch faults in the processing channel*

The RCCUs are only enabled when the SPC56xL70xx is in LSM. Application software must determine whether LSM mode is activate. Please refer to *Section 3.2.2, Checking* for further details.

3.19 Cyclic Redundancy Checker Unit (CRC)

The CRC module computes CRC checksums, which offloads the CPU. The CRC has the capability of processing two CRC calculations simultaneously.

Recommended: The CRC module should be used to detect accidental alteration of data during transmission or storage. The CRC takes as its input a data stream of any length and produces a 32-bit output value.

Mandatory: The CRC calculation shall be executed to verify the content of the registers.

Note: *Rationale: The contents of the configuration registers of the safety-related modules must be checked within the FTTI.*

Note: *Theoretically, the CPU could be used instead of the CRC to verify that the value of the configuration registers have not changed. However, using the CRC is more effective.*

Note: *Implementation hint: The CRC of the configuration registers of the modules involved with the safety function shall be calculated offline.*

At run time, the same CRC value shall be calculated by the CRC module within the safety process time. To avoid overloading the CPU, the eDMA module can be used to support the data transfer from the registers under check to the CRC module.

The result of the runtime computation is then compared to the value of the offline CRC.

The application must include detection, or protection measures, against possible faults of the CRC module only if the CRC module is used by any SEF.

3.20 Clock Monitor Unit (CMU)

The main task of the Clock Monitor Unit (CMU) is to supervise the integrity of various clock sources.

Mandatory: The following supervisor functions shall be used:

● Loss of external crystal oscillator clock
● FMPLL frequency higher than a (programmable) value set as high reference
● FMPLL frequency lower than a (programmable) value set as low reference

Note: *Rationale: To monitor the integrity of the clock signals*

This error information is forwarded to the FCCU and to the MC_RGM.

SPC56xL70xx includes three CMUs:

● CMU_0 monitors the clock signal of the SoR modules and the clock from the XOSC (XOSC_CLK).
● CMU_1 monitors the clock signal used by the motor control related peripherals (such as eTimer, FlexPWM, CTU and ADC).
● CMU_2 monitors the clock signal for the protocol engine of the FlexRay module.

Mandatory: For ASIL D applications, use of the CMU is mandatory. If the related modules are used by the application safety function, the user shall verify that the CMUs are enabled and their faults managed by the FCCU.

Note: **Rationale:** *To monitor the integrity of the various clock signals*

Note: **Implementation hint:** *In general, the following two application-dependent configurations must be executed before CMU monitoring will be enabled:*

● The first configuration is related to the XOSC_CLK monitor of CMU_0. The software shall configure CMU_0_CSR[RCDIV] to select a divider for the IRCOSC. The divided RCOSC frequency will be compared with the XOSC_CLK.

● The second configuration relates to the other clock signals being monitored. The high frequency reference (CMU_*n*_HFREFR_A[HFREF_A]) and low frequency reference (CMU_*n*_LFREFR_A[LFREF_A]) shall be configured depending on the SoR (CMU_0), motor control related peripherals (CMU_1) and FlexRay (CMU_2) clock frequencies.

Once the CMUs are configured, the clock monitoring must be enabled by asserting CMU_*n*_CSR[CME_A] (= 1).

3.21 Frequency-Modulated Phase-Locked Loop (FMPLL)

Mandatory: Application software has the responsibility of checking that the system uses the system FMPLL clock as system clock before running any safety element function (PLL_SWCHECK).

Note: **Rationale:** *To decrease the risk of a glitch from the crystal or IRCOSC*

Note: **Implementation hint:** *Application software can verify the current system clock by checking MC_ME_GS[S_SYSCLK] status. MC_ME_GS[S_SYSCLK] = 0x4 indicates system FMPLL clock is used as system clock.*

Mandatory: Each FMPLL provides a loss of lock error indication which is routed to the MC_RGM and FCCU. The application software must enable the respective fault and configure the FCCU to manage the fault.

Note: **Rationale:** *To check the integrity of the FMPLL clock*

Since the system can be driven by the IRCOSC, if there is a system clock fault, an FMPLL fault is considered a Non-Critical Fault (NCF). If the FMPLL successfully relocks after a clock fault it will typically stay relocked since the locking process includes built in hysteresis between loosing and regaining the lock.

Note: **Implementation hint:** *Software must clear FMPLL_n_CR[PLL_FAIL_MASK] so the pll_fail output is not masked.*

To enable the RGM input related to FMPLL loss of clock, RGM_FERD[D_PLLn] and RGM_FEAR[AR_PLLn] must be configured.

To enable FCCU fault paths, registers in the FCCU must be configured (NCF_CFG0, NCFS_CFG0, NCF_TOE0, etc.). Loss of lock signals from FMPLL_0 and FMPLL_1 provide the FCCU NCF[2] and NCF[3] inputs, respectively.

The MC_RGM and FCCU configuration includes the reaction in case of FMPLL loss of lock. This reaction is application-dependent.

3.22 Internal RC Oscillator (IRCOSC)

The frequency meter of CMU_0 must be exploited to verify the availability and frequency of the IRCOSC. This feature allows measuring the IRCOSC frequency using the external oscillator as the clock source.

Mandatory: Users must measure the IRCOSC frequency and compare it with what is expected (16MHz[i]). This test must be performed at least once every FTTI (IRC_SW_CHECK_SIF).

Note: *Rationale: To check the integrity of the IRCOSC*

Note: *If the IRCOSC is not operating due to a fault, the measurement of the IRCOSC frequency will never complete and the CMU_CSR[SFM] flag will remain set. The application shall manage detecting this condition. For example, implementing a software watchdog which monitors the CMU_CSR[SFM] flag status.*

Safety analysis assumes that this measurement executes at least once every FTTI. Testing frequency can be reduced to once after boot if the customer accepts that most safety mechanisms will be non-functional for the remainder of the operation if the IRCOSC fails.

Safety related modules which work with the RC clock are: FCCU, CMU and SWT. These modules stop working if the IRCOSC fails.

3.23 Power Management Unit (PMU)

The Power Management Units (PMU) manage the supply voltage of modules on the SPC56xL70xx. The supplies monitored by the PMU and naming conventions are found in *Table 13*.

Table 13. PMU monitored supplies

Detector Type	Detector Name	Voltage Monitored	Alternate Name	Comments
Flash memory LVD	LVD_MAIN_3	VDDFLASH	LVD_FLASH	A redundant LVD is embedded
I/O LVD	LVD_MAIN_1	VDDIO	LVD_GPIO	A redundant LVD is embedded
VREG LVD	LVD_MAIN_2	VDDREG	LVD_VREG	A redundant LVD is embedded
Core main LVD	LVD_DIG_MAIN	1.2 V digital	—	—
Core main HVD	HVD_DIG_MAIN	1.2 V digital	—	—
Core backup LVD	LVD_DIG_BKUP	1.2 V digital	—	Assists in the self-test of LVD_DIG_MAIN
Core backup HVD	HVD_DIG_BKUP	1.2 V digital	—	Assists in the self-test of HVD_DIG_MAIN

If one of the monitored voltages falls below or rises above a fixed threshold, a destructive reset is initiated. The Low Voltage Detection (LVD) and High Voltage Detection (HVD) fault indications are forwarded to the MC_RGM.

---

i. Nominal frequency of the IRCOSC is 16 MHz, but a post trim accuracy of $\pm$6% over voltage and temperature must be taken into account.

Since power is critical to the operation of the SPC56xL70xx there is built-in redundancy to the PMU core LVDs and HVDs. LVD_DIG_MAIN and HVD_DIG_MAIN monitor the digital core voltage and have backups for additional safety protection (LVD_DIG_BKUP and HVD_DIG_BKUP). Internal architecture allows for testing of the functionality of the main and back up LVD_DIG and HVD_DIG, as well as trimming circuitries (See *Figure 15*). The PMUCTRL module provides software initialized BISTs which test the digital core supply HVD and LVD (both main and backup).

Note: This scheme represents only the logical configuration and not the actual silicon implementation structure.
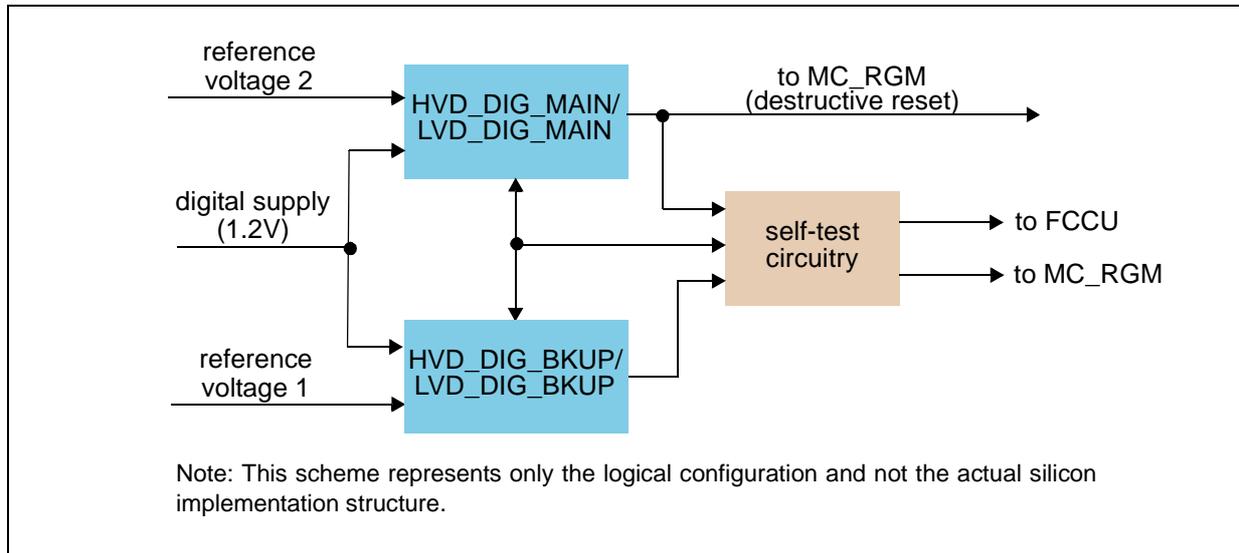
Figure 15.   Logic scheme of the LVD_DIG and HVD_DIG

If the self-test circuitry detects a fault in the main or backup detectors the reaction will be one of the following (See "Built In Self-test (BIST)" subsection of the "Power Management Unit (PMU)" section in the SPC56xL70xx Reference Manual):

● Critical Fault (CF[21]) triggered and one or more of the following:
  – Main Low Voltage Detector Pending – PMUCTRL_IRQS[MLVDP] = 1
  – Backup Low Voltage Detector Pending – PMUCTRL_IRQS[BLVDP] = 1
  – Main High Voltage Detector Pending – PMUCTRL_IRQS[MHVDP] = 1
  – Backup High Voltage Detector Pending – PMUCTRL_IRQS[BHVDP] = 1
● Destructive reset triggered

If the self-test circuitry detects a fault in the main or backup detectors the FCCU will read a CF.

There are dedicated LVD's in the flash memory, I/O and VREG providing additional redundancy. This solution is different from the 1.2 V digital core supply monitoring, but still provides the same level of safety coverage. The outputs of the first and the second LVD are logically AND'd in such a way that a single LVD can trigger a fault, even if the other LVD is not functioning properly (See *Figure 16*).

Note: This scheme represents only the logical configuration and not the actual silicon implementation structure.
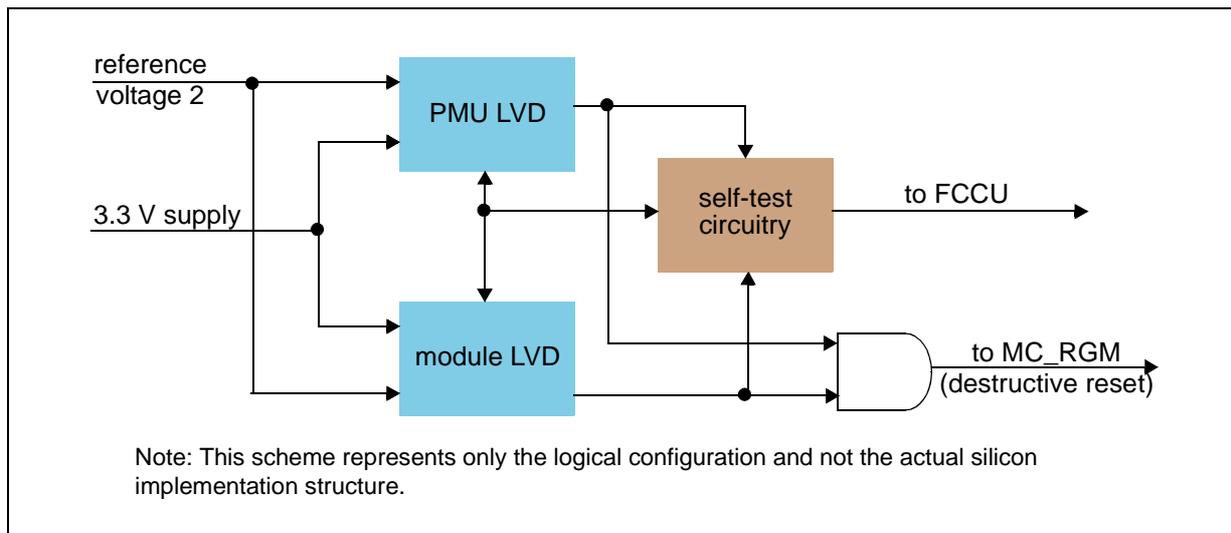
Figure 16.   **Logic scheme of the LVD_FLASH, LVD_GPIO and LVD_VREG**

Operation of the LVD_FLASH, LVD_GPIO and LVD_VREG is as follows (software intervention is not needed):

●   A single LVD (PMU LVD or module LVF) can trigger a fault even if the other LVD is faulty (this event signals the MC_RGM)

●   During each power on cycle self-test circuitry is able to detect failures on one of the two LVD's (this event signals the FCCU).

Mandatory: Core voltage LVD and HVD implement a hardware assisted self-test that needs to be initiated by software once after the boot.

Note: **Rationale:** *To check the integrity of the LVD and HVD*

Note: **Implementation hint:** *The hardware assisted self-tests are initiated by configuring PMUCTRL_CTRL[SILHT[1:0]].*

               If the self-test passes, an NCF is triggered. If the self-test fails, a PMUCTRL_IRQS flag and CF are asserted.

Apart from the self-test, the use of the power management unit for ASIL D applications is transparent to the user, because the operation of the PMU is automatic. The SPC56xL70xx embeds three LVDs which can detect a failure in the 3.3V power supply. Considering the failure mode "Wrong Power Regulation", a diagnostic coverage of 90% is estimated against both a soft error and DC fault.

3.24   Memory Protection Unit (MPU)

The Memory Protection Unit (MPU) provides hardware access control for all memory references generated in a device. Using pre-programmed region descriptors that define memory spaces and their associated access rights, the MPU concurrently monitors all system bus transactions (including those initiated by the eDMA or FlexRay controller) and evaluates the appropriateness of each transfer.

Memory references that have sufficient access control rights are allowed to complete, while references that are not mapped to any region descriptor or have insufficient rights are terminated with a protection error response.

The MPU implements a set of program-visible region descriptors that monitor all system bus addresses. The result is a hardware structure with a two-dimensional connection matrix, where the region descriptors represent one dimension and the individual system bus addresses and attributes represent the second dimension.

Recommended: For ASIL D applications, the MPU should be used to ensure that only authorized software routines can configure modules and all other bus masters (eDMA, core, FlexRay) can access only their allocated resources according to their access rights. For the non-replicated master FlexRay, a correct MPU setup is highly recommended.

3.25 Register Protection Module

The Register Protection module offers a mechanism to protect defined memory mapped address locations in a module that has been write protected. The address locations that can be protected are module specific.

The Register Protection module includes these distinctive features:

● The Register Protection module restricts write accesses for the module under protection to supervisor mode only. This access restriction is in addition to any access restrictions imposed by the protected module.

● A register cannot be written once Soft Lock Protection is set. Soft Lock Protection can be cleared by software or system reset.

● A register cannot be written once Hard Lock Protection is set. Hard Lock Protection can only be cleared by system reset.

Mandatory: For ASIL D applications, all configuration registers that aren't modified during application execution, must be protected with a Hard Lock.

Note: **Rationale:** *Hard Lock is the last access protection against unwanted writes to some predefined memory mapped address locations.*

Mandatory: Access restrictions must be handled at MPU level.

Note: **Rationale:** *Access restriction at the MPU level is protection against unwanted read/write accesses to some predefined memory mapped address locations.*

Recommended: It is recommended that only hardware related software (OS, drivers) run in supervisor mode.

Note: **Implementation hint**: *Most of the off-platform peripherals have their own Register Protection module. Register Protection address space is inside the memory space reserved for the peripherals (please, refer to the "SPC56xL70xx registers under protection" section of the SPC56xL70xx Reference Manual).*

Each peripheral register that can be protected through the Register Protection module has a Set Soft Lock bit reserved in the Register Protection address space. This bit shall be asserted to enable the protection of the related peripheral registers. Moreover, the Hard Lock Bit (REG_PROT_GCR[HLB] = 1) should be set for best write protection.

3.26 Error Correction Status Module (ECSM)

There is no dedicated ECC module on the SPC56xL70xx. ECC functionality is located in, or near, the different storage modules and may vary slightly depending on the needs (and size) of the storage. The ECSM is used to detect failures of data stored in memory (SRAM only) and addressing failures (See "Error Correction Status Module (ECSM)" in the SPC56xL70xx Reference Manual).

The ECSM can detect and correct single-bit errors, detect two bit faults and detect faults affecting more than two bits. For SRAM, addressing information is included in the calculation and evaluation of the ECC to also detect addressing failure of the SRAM arrays. Single-bit addressing failures that are detected are not corrected. Instead, they are treated as a detected multi-bit error.

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

The ECSM corrects read data when a single-bit error is detected. Optionally, the user can raise an interrupt or check the address of last corrected data.

In the case of a multi-bit fault, both the FCCU and MC_RGM modules take appropriate actions:

● Activate error out pins
● Reset
● NMI is triggered.

The reporting functionality of the ECSM is disabled by default.

Mandatory: Before the safety application starts executing, the error reporting shall be enabled.

Note: **Implementation hint:** *Error reporting is enabled by configuring the ECC Configuration Register (ECR) of the ECSM module (for example, ECSM_ECR[EPR1BR] = 1b, see section "ECC Configuration Register (ECR)" in the SPC56xL70xx Reference Manual for details).*

3.27 Fault Collection and Control Unit (FCCU)

The Fault Collection and Control Unit (FCCU) offers a hardware channel to collect faults and to bring the device into a safe state when a failure has occurred.

Besides the possible initial configuration, no CPU intervention is necessary for collection and control operation.

The FCCU offers a systematic approach to fault detection and control. The distinctive features of the module are:

● Collection of redundant hardware checker results (e.g., the RCCU. See *Section 3.18, Redundancy Control Checking Unit (RCCU)*)

● Collection of error information from modules whose behavior is essential with respect to the safety goal

● Configurable and graded fault control:

– Internal reactions

No reset reaction

IRQ

Functional Reset

SPC56xL70xx safe mode entered

– External reaction (failure is reported to the outside world via output pin)

Mandatory: Only functional resets, or a switch to a Safe state, is appropriate as internal reaction for ASIL D applications.

Note: *Rationale:* *Maintain the device in the Safe state in case of failure*

The only exception to this rule is when the CMU monitors a FMPLL that is not used or is used for non-safety critical modules only. In this case, error masking and limited internal reaction can be tolerated.

External reaction of the FCCU is always enabled and can not be disabled.

Note: *Implementation hint:* *The application shall configure the FCCU to enable all reactions related to faults of peripherals used by the application safety function.*

Software shall be implemented to avoid cycling between a functional and a fault state. For example, in case of periodic NCFs, the software could clean the respective status and periodically move the device from fault state to normal state. This looping shall be avoided.

Mandatory: To prevent permanent cycling between a functional and a fault-state, software needs to keep track of cleaned faults, stop cleaning and stay in safe mode instead in case of inacceptable high frequency of necessary fault cleaning. The limit for the number and frequency of clearances is application dependent.

4 Functions of external devices for ASIL D applications

This section describes the external components needed to use the SPC56xL70xx for ASIL D applications.

Mandatory: 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) when an output in high-impedance is not considered safe.

It should be noted that the failure rates of external services are not included in FMEDA of the SPC56xL70xx and have to be included in the system FMEDA by the user himself.

4.1 External Watchdog Function (EXWD)

Mandatory: An external device, acting as the supervisor of operations, must provide a watchdog to cover CCFs of the SPC56xL70xx for ASIL D applications. The watchdog shall be triggered periodically by safety relevant software running on the SPC56xL70xx or other means demonstrating that the SPC56xL70xx is still working.

Note: *Rationale: To detect critical 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 (such as the EXWD disconnects the SPC56xL70xx device from the power supply).

The user can choose how to implement the watchdog communication between the SPC56xL70xx and the external device (for example, communication via serial link, via toggling pin, or via the FCCU error out signals).

Note: *There must be a signalling path from the safety software to the external system through which the software can confirm correct initialization. This is not automatically guaranteed by the FCCU_F[n] signals which communicate the status of the device independently from software. On the other hand, a different communications interface (such as a serial link) can be used to detect incorrect software initialization.*

4.2 Power Supply and Monitor Function (PSM)

The SPC56xL70xx includes some internal monitors which continuously check the various voltage supplies (See *Section 3.23, Power Management Unit (PMU)*).

Mandatory: To prevent over voltage conditions causing malfunctions or possibly permanent damage to the SPC56xL70xx, an external device must provide over voltage monitoring for the SPC56xL70xx external 3.3 V supplies (such as I/O and VREG).

Under voltage conditions on the 3.3V supply may be detected indirectly by measurements from other functionality like the ADC self-test or ECC/ECD logic.

Recommended: To fully monitor all voltage supplies, it is also recommended that an external device provides under voltage monitoring for the SPC56xL70xx external 3.3 V supplies (such as I/O and VREG).

Note: *Rationale: To monitor the power supply voltage to ensure it is within the acceptable range*

If the power supply is out of range, the PSM moves and maintains the system (ECU level) to a Safe state condition within the FTTI (for example, the PSM disconnects the SPC56xL70xx device from the power supply).

Note: *Working outside the specified voltage range may cause permanent damage to the SPC56xL70xx even if the MCU is held in reset (see SPC56xL70xx Data Sheet for correct voltage operating ranges).*

4.3 Error Out Monitor Function (ERRM)

The FCCU has two external pins: FCCU_F[0], FCCU_F[1].

An external device must be connected to the FCCU via FCCU_F[0] and optionally FCCU_F[1] to continually monitor the error output pins of the FCCU.

If a failure is detected, the ERRM moves and maintains the system (ECU level) to a Safe state condition within the FTTI (e.g., the ERRM disconnects the SPC56xL70xx device from the power supply)

Mandatory: Depending on user selection, there are two different ways to interface to the FCCU:

● Both FCCU pins connected to the external device
● Only a single FCCU pin connected to the external device

Note: **Rationale:** *To monitor the error out signals (FCCU_F[x]) for correct functionality*

Mandatory: For ASIL D applications, the user can choose between these FCCU configurations, depending on which best fits the hardware and software system.

Both FCCU configurations work properly with all the supported error out protocols. Refer to the SPC56xL70xx Reference Manual for a list of supported protocols.

Note: *The system (for example, ECU) cannot rely on any pins, other than the SPC56xL70xx error output pins (FCCU_F[n]), when those pins indicate an error.*

4.3.1 Both FCCU pins connected to external device

In this case, both pins FCCU_F[0] and FCCU_F[1] are connected to the external device.

Mandatory: The external device must check both signals, taking into account that FCCU_F[0] = $\overline{\text{FCCU\_F[1]}}$.

Note: **Rationale:** *To check the integrity of the FCCU*

In this configuration the external device continuously monitors the output of the FCCU. Thus it can detect if the FCCU does not work properly.

The advantage of this configuration with respect to the other one is that it does not need any dedicated software.

Note: **Implementation hint:** *Monitoring the error out pins through a combinatorial logic (e.g., XOR port) can generate some glitches. Oversampling these pins reduces the possibility that the glitches occur.*

4.3.2 Single FCCU pin connected to external device

A single pin, FCCU_F[0] (or FCCU_F[1]), is connected to the external device.

If a fault occurs, the FCCU communicates it to the external device through the FCCU_F[0] (or FCCU_F[1]) pin.

The functionality of FCCU_F[0] (or FCCU_F[1]) can be verified in 2 ways:
- FCCU_F[0] (or FCCU_F[1]) output read back (internal connection)
- FCCU_F[0] (or FCCU_F[1]) output connected externally to a normal GPIO.

The customer must choose which solution better fits their requirements.

Mandatory: After boot, but before executing the safety function, the functionality of FCCU_F[0] (or FCCU_F[1]) pin shall be verified[j].

Note: **Rationale:** *To check the integrity of the FCCU error out signals*

Note: **Implementation hint:** *To verify the functionality of FCCU_F[0] (or FCCU_F[1]) pin, a fault may be injected and the behavior of the pin could be checked by FCCU_F[1] (or FCCU_F[0]), or GPIO. It's possible to change the polarity of the error out pin by configuring the FCCU_CFG[FCCU_CFG.PS] bit. Other methods for checking the functionality of FCCU_F[0] (or FCCU_F[1]) may be implemented.*

The advantage of a single FCCU_F[x] signal being used, when compared to using both FCCU_F[x] signals as in the previous section, is that an external device does not need to be used for comparing the FCCU_F[x] signals.

4.4 PWM Output monitored by external ASIC (PWMA)

The FlexPWM module integrated in the SPC56xL70xx can insert dead time in the generated PWMs.

Mandatory: An ASIL D compliant application shall include an external device which checks the PWM output signals.

Note: **Rationale:** *To check the accuracy of the PWM signals*

The distinctive features that must be managed by the external device are:
- Dead-time must be always positive and greater than the maximum value between TON and TOFF of the inverter switches
- Open pins and short to supply or ground shall be detected in case read back is not performed via input capture functionality on the SPC56xL70xx

If a failure is detected, the PWMA moves and maintains the system (ECU level) to a Safe state condition within the FTTI (e.g., the PWMA disconnects the SPC56xL70xx device from the power supply).

In general, if the safety application uses I/Os to control actuator with short safety time against wrong control (for example, a motor control application with dead-time requirements to avoid short circuits destroying the motor), those requirements shall be supervised externally if the error reaction delay within the SPC56xL70xx can exceed the safety time of the actuators.

---

j. Since FCCU is a monitor, it is sufficient to verify the FCCU_F[0] (or FCCU_F[1]) signal only at startup in order to avoid latent faults.

Note: **Implementation hint:** *In case PWM signals drive the switches of a power stage, eTimer can not be used to detect dead-time fault because its failure indication time is normally greater than the time needed to have a physical permanent failure in the power stage.*

5   Scenarios for automotive applications: Motor control

This section shows some examples of safety-related inputs and outputs from some motor control applications.

5.1   Application example 1

- Application: 3-phase electric motor control
- Motor control algorithm: Field Oriented Control (FOC)
- Position sensor(s): Incremental encoder; 3 Hall sensors
- Current sensor(s): 3 shunts on motor phases or on inverter legs
- Current sensor(s) for diagnostic: 1 shunt on direct-current (DC) link

5.1.1   Functional safety related inputs

Table 14.   Functional safety inputs for application example 1

Signal description	Input signal (alias)	Source	Destination (module on SPC56xL70xx)	Comments
FCCU input	FCCU_F[1] (if used)	FCCU output pin FCCU_F[0]	FCCU	FCCU output loop-back signal.
Phase current 1	AN[0]	ASIC or current sensor	ADC_0	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
Phase current 2	AN[15]	ASIC or current sensor	ADC_1	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
Phase current 3	AN[11]	ASIC or current sensor	ADC_0 ADC_1	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
DC voltage for DC ripple compensation	AN[1]	ASIC	ADC_0	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
DC-link current	AN[16]	ASIC or current sensor	ADC_1	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
Non-maskable interrupt	NMI[1]	External component (ASIC)	WKPU	Critical interrupt routine or error/fault signal coming from external device.
Reset signal	RESET B	External component (ASIC, companion chip)	MC_RGM	Reset signal coming from external device.

Table 14. Functional safety inputs for application example 1 (continued)

Signal description	Input signal (alias)	Source	Destination (module on SPC56xL70xx)	Comments
Incremental Encoder management	ETC[0–1] ETC[0–1]	Incremental encoder	eTimer_0 eTimer_1	Precautions for usage are presented in *Section , Double Read Encoder Inputs*.
Hall sensors management	ETC[2–4] ETC[2–4]	Hall sensors	SIUL	Precautions for usage are presented in *Section , Double Read PWM Inputs*.
DSPI receive signal	SIN	External component (ASIC)	DSPI_0	If DSPI_0 is used, an appropriate safety protocol must be utilized.
DSPI receive signal	SIN	External component (ASIC)	DSPI_1	If DSPI_1 is used, an appropriate safety protocol must be utilized.
FlexCAN receive signal	CAN_RX	External component (ASIC)	FlexCAN_0	If FlexCAN_0 is used, an appropriate safety protocol must be utilized.
FlexCAN receive signal	CAN_RX	External component (ASIC)	FlexCAN_1	If FlexCAN_1 is used, an appropriate safety protocol must be utilized.
FlexRay receive signals	FR_CA_RX FR_CB_RX	External component (ASIC)	FlexRay	If FlexRay is used, an appropriate safety protocol must be utilized.

1. The NMI input is not intended or certified for use as the sole mechanism to react to the failure of a system component external to the SPC56xL70xx. For ASIL D certification, additional measures at the system level are necessary to handle failures of non-SPC56xL70xx components beyond notification of the SPC56xL70xx device via NMI.

5.1.2 Functional safety related outputs

Table 15. Functional safety outputs for application example 1

Signal description	Output signal (alias)	Source (module on SPC56xL70xx)	Destination	Comments
FCCU output	FCCU_F[0]	FCCU	External component (ASIC)	Error out signal that indicates the presence of a failure in the device.
FCCU output	FCCU_F[0]	FCCU	Alternative 1: FCCU_F[1]	FCCU output loop-back signal.
	$\overline{FCCU\_F[1]} = FCCU\_F[0]$	FCCU	Alternative 2: External component (ASIC)	Inverted Error out signal that indicates the presence of a failure in the device.
PWM output signal	A[0–2], B[0–2]	FlexPWM	External component (ASIC)	Precautions for usage are presented in *Section , Single Write PWM Outputs With Read Back*.

Table 15. Functional safety outputs for application example 1 (continued)

Signal description	Output sIgnal (alias)	Source (module on SPC56xL70xx)	Destination	Comments
Clockout	CLK_OUT	MC_CGM	External component (ASIC)	Clockout signal to be used if the external components needs the SPC56xL70xx clock for internal usage or for monitoring.
Clockout inverted signal	CLK_OUT	MC_CGM	External component (ASIC)	Inverted clockout signal to be used if the external components needs the SPC56xL70xx clock for internal usage or for monitoring.
Transceiver enable (for communication peripherals)	GPO[–]	SIUL	External component (ASIC, transceiver)	Precautions for usage are presented in *Section , Single Write Digital Outputs With Read Back*.
Reset signal	GPO[0]	SIUL	External component (ASIC, companion chip)	Reset signal for the external component(s) Precautions for usage are presented in *Section , Single Write Digital Outputs With Read Back*.
DSPI transmit signal	SOUT	DSPI_0	External component (ASIC)	If DSPI_0 is used, an appropriate safety protocol must be utilized.
DSPI transmit signal	SOUT	DSPI_1	External component (ASIC)	If DSPI_1 is used, an appropriate safety protocol must be utilized.
FlexCAN transmit signal	CAN_TX	FlexCAN_0	External component (ASIC)	If FlexCAN_0 is used, an appropriate safety protocol must be utilized.
FlexCAN transmit signal	CAN_TX	FlexCAN_1	External component (ASIC)	If FlexCAN_1 is used, an appropriate safety protocol must be utilized.
FlexRay transmit signals	FR_CA_TX FR_CB_TX	FlexRay	External component (ASIC)	If FlexRay is used, an appropriate safety protocol must be utilized.

5.2 Application example 2

- Application: 3-phase electric motor control
- Motor control algorithm: Field Oriented Control (FOC)
- Position sensor(s): Resolver; 3 Hall sensors
- Current sensor(s): 3 shunts on motor phases or on inverter legs
- Current sensor(s) for diagnostic: 1 shunt on DC link

5.2.1 Functional safety related inputs

Table 16. Functional safety inputs for application example 2

Signal description	Input Signal (alias)	Source	Destination (module on SPC56xL70xx)	Comments
FCCU input	FCCU_F[1] (if used)	FCCU output pin FCCU_F[0]	FCCU	FCCU output loop-back signal.
Phase current 1	AN[0]	ASIC or current sensor	ADC_0	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
Phase current 2	AN[15]	ASIC or current sensor	ADC_1	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
Phase current 3	AN[11]	ASIC or current sensor	ADC_0 ADC_1	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
DC voltage for DC ripple compensation	AN[1]	ASIC	ADC_0	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
DC-link current	AN[16]	ASIC or current sensor	ADC_1	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
Non-maskable interrupt	NMI[1]	External component (ASIC)	Wake-up Unit	Critical interrupt routine or error/fault signal coming from external device.
Reset signal	RESET B	External component (ASIC, companion chip)	MC_RGM	Reset signal coming from external device.
Resolver management (sine/cosine)	AN[2–3] AN[17–18]	Resolver	ADC_0 ADC_1	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
Hall sensors management	ETC[0–2] ETC[0–2]	Hall sensors	eTimer_0 eTimer_1	Precautions for usage are presented in *Section , Double Read PWM Inputs*.
DSPI receive signal	SIN	External component (ASIC)	DSPI_0	If DSPI_0 is used, an appropriate safety protocol must be utilized.
DSPI receive signal	SIN	External component (ASIC)	DSPI_1	If DSPI_1 is used, an appropriate safety protocol must be utilized.
FlexCAN receive signal	CAN_RX	External component (ASIC)	FlexCAN_0	If FlexCAN_0 is used, an appropriate safety protocol must be utilized.

Table 16. Functional safety inputs for application example 2 (continued)

Signal description	Input Signal (alias)	Source	Destination (module on SPC56xL70xx)	Comments
FlexCAN receive signal	CAN_RX	External component (ASIC)	FlexCAN_1	If FlexCAN_1 is used, an appropriate safety protocol must be utilized.
FlexRay receive signals	FR_CA_RX FR_CB_RX	External component (ASIC)	FlexRay	If FlexRay is used, an appropriate safety protocol must be utilized.

1. The NMI input is not intended or certified for use as the sole mechanism to react to the failure of a system component external to the SPC56xL70xx device. For ASIL D certification, additional measures at the system level are necessary to handle failures of non-SPC56xL70xx components beyond notification of the SPC56xL70xx device via NMI.

5.2.2 Functional safety related outputs

Table 17. Functional safety outputs for application example 2

Signal description	Output signal (alias)	Source (module on SPC56xL70xx)	Destination	Comments
FCCU output	FCCU_F[0]	FCCU	External component (ASIC)	Error out signal, that indicates the presence of a failure in the device.
FCCU output	FCCU_F[0]	FCCU	Alternative 1: FCCU_F[1]	FCCU output loop-back signal.
	$\overline{FCCU\_F[1]} = FCCU\_F[0]$	FCCU	Alternative 2: External component (ASIC)	Inverted Error out signal that indicates the presence of a failure in the device.
PWM output signal	A[0–2], B[0–2]	FlexPWM	External component (ASIC)	Precautions for usage are presented in *Section , Single Write PWM Outputs With Read Back*.
Resolver excitation	DA [0]	SWG	Resolver	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
Clockout	CLK_OUT	MC_CGM	External component (ASIC)	Clockout signal to be used if the external components need the SPC56xL70xx clock for internal usage or for monitoring.
Clockout inverted signal	CLK_OUT	MC_CGM	External component (ASIC)	Inverted clockout signal to be used if the external components need the SPC56xL70xx clock for internal usage or for monitoring.
Transceiver enable (for communication peripherals)	GPO[-]	SIUL	External component (ASIC, transceiver)	Precautions for usage are presented in *Section , Single Write Digital Outputs With Read Back*.

Table 17. Functional safety outputs for application example 2 (continued)

Signal description	Output signal (alias)	Source (module on SPC56xL70xx)	Destination	Comments
Reset signal	GPO[0]	SIUL	External component (ASIC, companion chip)	Reset signal for the external component(s). Precautions for usage are presented in *Section , Single Write Digital Outputs With Read Back*.
DSPI transmit signal	SOUT	DSPI_0	External component (ASIC)	If DSPI_0 is used, an appropriate safety protocol must be utilized.
DSPI transmit signal	SOUT	DSPI_1	External component (ASIC)	If DSPI_1 is used, an appropriate safety protocol must be utilized.
FlexCAN transmit signal	CAN_TX	FlexCAN_0	External component (ASIC)	If FlexCAN_0 is used, an appropriate safety protocol must be utilized.
FlexCAN transmit signal	CAN_TX	FlexCAN_1	External component (ASIC)	If FlexCAN_1 is used, an appropriate safety protocol must be utilized.
FlexRay transmit signals	FR_CA_TX FR_CB_TX	FlexRay	External component (ASIC)	If FlexRay is used, an appropriate safety protocol must be utilized.

5.3 Application example 3

● Application: 3-phase electric motor control
● Motor control algorithm: Sinusoidal Control (SC) or 6-step mode
● Position sensor(s): Incremental encoder; 3 Hall sensors
● Current sensor(s) for diagnostic: 1 shunt on DC link

5.3.1 Functional safety related inputs

Table 18. Functional safety inputs for application example 3

Signal description	Input signal (alias)	Source	Destination (module on SPC56xL70xx)	Comments
FCCU input	FCCU_F[1] (if used)	FCCU output pin FCCU_F[0]	FCCU	FCCU output loop-back signal.
DC voltage for DC ripple compensation	AN[0]	ASIC	ADC_0	Precautions for usage are presented in *Section , Single Read Analog Inputs*.
DC-link current	AN[15]	ASIC or current sensor	ADC_1	Precautions for usage are presented in *Section , Single Read Analog Inputs*.

Table 18.   Functional safety inputs for application example 3 (continued)

Signal description	Input signal (alias)	Source	Destination (module on SPC56xL70xx)	Comments
Non-maskable interrupt	NMI[1]	External component (ASIC)	Wake-up Unit	Critical interrupt routine or error/fault signal coming from external device.
Reset signal	RESET B	External component (ASIC, companion chip)	MC_RGM	Reset signal coming from external device.
Incremental Encoder management	ETC[0–1] ETC[0–1]	Incremental encoder	eTimer_0 eTimer_1	Precautions for usage are presented in *Section , Double Read Encoder Inputs*.
Hall sensors management	ETC[2–4] ETC[2–4]	Hall sensors	eTimer_0 eTimer_1	Precautions for usage are presented in *Section , Double Read PWM Inputs*.
DSPI receive signal	SIN	External component (ASIC)	DSPI_0	If DSPI_0 is used, an appropriate safety protocol must be utilized.
DSPI receive signal	SIN	External component (ASIC)	DSPI_1	If DSPI_1 is used, an appropriate safety protocol must be utilized.
FlexCAN receive signal	CAN_RX	External component (ASIC)	FlexCAN_0	If FlexCAN_0 is used, an appropriate safety protocol must be utilized.
FlexCAN receive signal	CAN_RX	External component (ASIC)	FlexCAN_1	If FlexCAN_1 is used, an appropriate safety protocol must be utilized.
FlexRay receive signals	FR_CA_RX FR_CB_RX	External component (ASIC)	FlexRay	If FlexRay is used, an appropriate safety protocol must be utilized.

1.   The NMI input is not intended or certified for use as the sole mechanism to react to the failure of a system component external to the SPC56xL70xx device. For ASIL D certification, additional measures at the system level are necessary to handle failures of non-SPC56xL70xx components beyond notification of the SPC56xL70xx device via NMI.

5.3.2   Functional safety related outputs

Table 19.   Functional safety outputs for application example 3

Signal description	Output signal (alias)	Source (module on SPC56xL70xx)	Destination	Comments
FCCU output	FCCU_F[0]	FCCU	External component (ASIC)	Error out signal, that indicates the presence of a failure in the device.

Table 19. Functional safety outputs for application example 3 (continued)

Signal description	Output signal (alias)	Source (module on SPC56xL70xx)	Destination	Comments
FCCU output	FCCU_F[0]	FCCU	Alternative 1: FCCU_F[1]	FCCU output loop-back signal.
	FCCU_F[1] = $\overline{FCCU\_F[0]}$	FCCU	Alternative 2: External component (ASIC)	Inverted Error out signal, that indicates the presence of a failure in the device.
PWM output signal	A[0–2], B[0–2]	FlexPWM	External component (ASIC)	Precautions for usage are presented in *Section , Single Write PWM Outputs With Read Back*.
Clockout	CLK_OUT	MC_CGM	External component (ASIC)	Clockout signal to be used if the external components need the SPC56xL70xx clock for internal usage or for monitoring.
Clockout inverted signal	CLK_OUT	MC_CGM	External component (ASIC)	Inverted clockout signal to be used if the external components need the SPC56xL70xx clock for internal usage or for monitoring
Transceiver enable (for communication peripherals)	GPO[–]	SIUL	External component (ASIC, transceiver)	Precautions for usage are presented in *Section , Single Write Digital Outputs With Read Back*.
Reset signal	GPO[0]	SIUL	External component (ASIC, companion chip)	Reset signal for the external component(s). Precautions for usage are presented in *Section , Single Write Digital Outputs With Read Back*.
DSPI transmit signal	SOUT	DSPI_0	External component (ASIC)	If DSPI_0 is used, an appropriate safety protocol must be utilized.
DSPI transmit signal	SOUT	DSPI_1	External component (ASIC)	If DSPI_1 is used, an appropriate safety protocol must be utilized.
FlexCAN transmit signal	CAN_TX	FlexCAN_0	External component (ASIC)	If FlexCAN_0 is used, an appropriate safety protocol must be utilized.
FlexCAN transmit signal	CAN_TX	FlexCAN_1	External component (ASIC)	If FlexCAN_1 is used, an appropriate safety protocol must be utilized.
FlexRay transmit signals	FR_CA_TX FR_CB_TX	FlexRay	External component (ASIC)	If FlexRay is used, an appropriate safety protocol must be utilized.

6 ECC logic test

6.1 Overview

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

A flash memory ECC logic test is needed to perform a test to check flash memory ECC logic every FTTI (10 ms).

The goal is to ensure high coverage of the faults in ECC logic with minimum performance penalty to customer's application. Thus, the performance penalty must be less than 2% which means that the test lasts less than 200 μs considering a FTTI of 10 ms.

The SPC56xL70xx 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.

6.2 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 20* shows the data vectors.

Table 20.   Data pattern used by the ECC logic test[1]

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

1. Each vector is a 72-bit ECC code-word.

It is important to note that for double word data = 0xFFFF_FFFF_FFFF_FFFF, the correct ECC check bits should be 0xFF. Therefore, every data vector in the data pattern in *Table 20*, except the last one, contains a single-bit ECC error and will result in a single-bit correction.

6.3 UTEST mode ECC logic check

The procedure to use the UTEST mode ECC logic check is listed as below:

1. Write 0xF9F9_9999 to UT0 to enable UTEST mode (UT0[UTE] will be set).
2. Write UT0[SBCE] to 1 to enable single-bit error correction visibility.
3. Write UT0[EIE] to 1.
4. Write UT0[DSI], UT1[DAI] and/or UT2[DAI] bits to provide the current data vector including the double-word data and check bit values to be read. The data and check bit values are from the chosen ECC test data pattern, i.e., walking 0 pattern shown above.
5. Write double-word address to receive the data input in step 4 into the ADR register.
6. Reads the address stored in ADR register via BIU using a CPU instruction. The 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[SBC] will be checked to evaluate the status of reads done.
7. Repeat steps 4 to 6 for all the data vectors in the proposed test data pattern.
8. Once completed, clear the UT0[EIE] bit to 0.

6.4 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 176 µs at 80 MHz.

7 I/O pin/ball configuration

Mandatory: The user must avoid configurations that place redundant signals on neighboring pads or pins.

Whether two functions on two package pins/balls are adjacent to each other can easily be determined by looking at the mechanical drawings of the packages (see the *SPC56xL70xx Data Sheet*) together with the pin/spheres (balls) number information of the packages as seen in the *SPC56xL70xx Reference Manuals* "System Integration Unit Lite (SIUL)" section and the "Pin muxing" table.

The internal die pad sequence can be derived from the package pin sequence of the QFP144 pin package shown in the *SPC56xL70xx Data Sheet*.
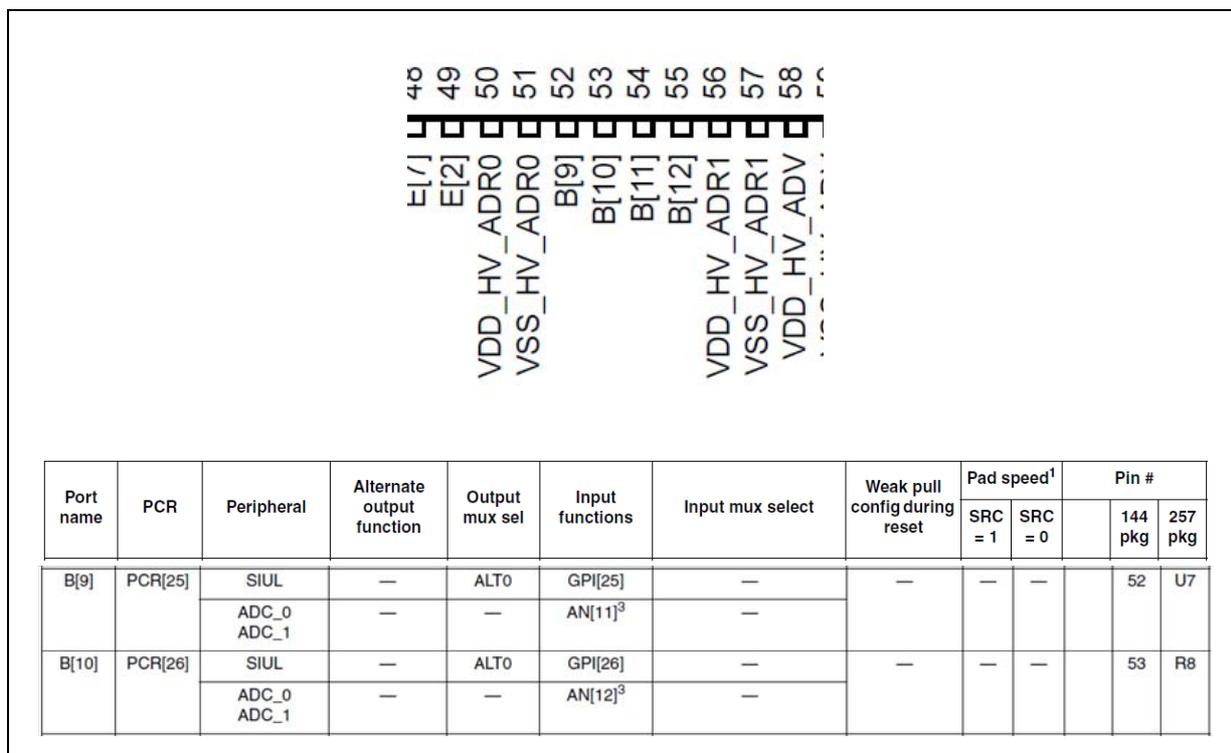

Port name	PCR	Peripheral	Alternate output function	Output mux sel	Input functions	Input mux select	Weak pull config during reset	Pad speed[1] SRC = 1	Pad speed[1] SRC = 0	Pin # 144 pkg	Pin # 257 pkg
B[9]	PCR[25]	SIUL	—	ALT0	GPI[25]	—	—	—	—	52	U7
		ADC_0 ADC_1	—	—	AN[11][3]	—					
B[10]	PCR[26]	SIUL	—	ALT0	GPI[26]	—	—	—	—	53	R8
		ADC_0 ADC_1	—	—	AN[12][3]	—					

Figure 17. Example of QFP144 pin/pad adjacency

For example, the internal die pads supporting the functionality described in *Figure 17* are referred to by "Port Pin" in the first column. From this figure you can see that the port pins are B[9] and B[10]. Since these two port pins are in sequential order on the same port (Port B) the die pads are adjacent to each other. The corresponding two QFP144 package pin numbers are directly adjacent to each other, QFP144 pins 52 and 53. In general, the internal die pads follow the same sequence as the corresponding package pins for QFP144 packages. If pins on the QFP144 pins are adjacent to each other, the corresponding internal die pads are also adjacent. Likewise, if package pins are not adjacent to each other the corresponding die pads are also not adjacent.

An example on the BGA package as shown in *Figure 18* has two balls belonging to port pins B[9] and B[10], which are balls U7 and R8, respectively. They are not directly adjacent to

each other on the BGA package. However, their corresponding die pads are adjacent to each other as described above since the same die is used in the QFP144 and BGA packages.

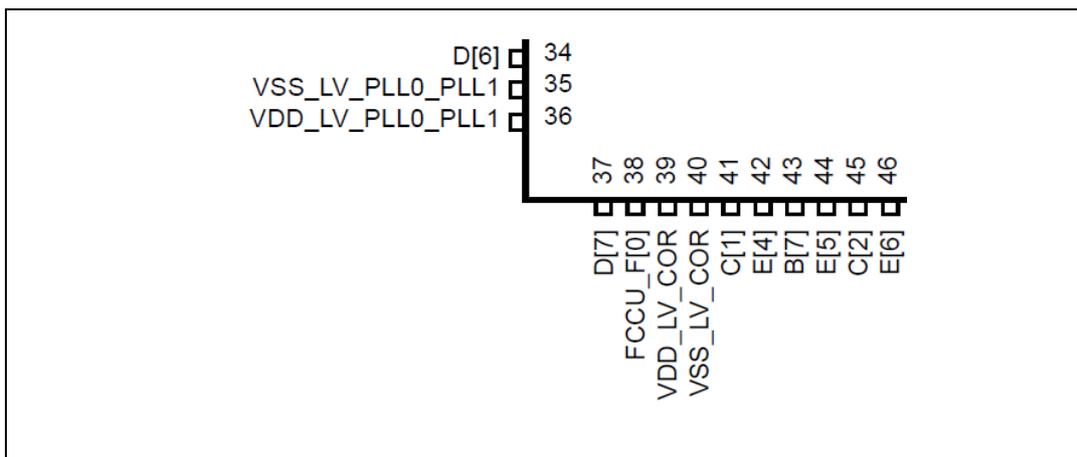**Figure 18. BGA balls non-adjacent, die pads adjacent**

Figure 19. BGA balls adjacent, die pads non-adjacent

In another example looking at balls U4 and U5 in *Figure 18*. Their functionality is implemented by Port Pins E[4] and C[2] (QFP144 pins 42 and 45, respectively, shown in *Figure 19*). These two spheres are adjacent to each other on the BGA, but not on the QFP144. Therefore, the two corresponding die pads are not adjacent to each other.

The above examples are valid for corresponding pins on BGA (257 balls) and QFP144 packages only. For a thorough analysis of pin adjacency related to all signals see . This table can be used to determine whether two pins are adjacent in the internal die for all signals and packages. Two pins, identified by the columns 'Port Name', are adjacent on the internal die if the numbers in the 'Physical Pad Sequence' column are consecutive (for example, pad number n and pad number $n + 1$ are adjacent).

8 Further information

8.1 Conventions and terminology

Table 21 shows the list of conventions for this document.

Table 21.   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.

8.2 Acronyms and abbreviations

A short list of acronyms and abbreviations used in this document is reported below for completeness.

Table 22.   Acronyms and abbreviations

Terms	Meanings
ADC	Analog to Digital Converter
BAM	Boot Assist Module
CCF	Common Cause Failure
CF	Critical Fault
CMU	Clock Monitor Unit
CRC	Cyclic Redundancy Check
CTU	Cross-Triggering Unit
DC	Diagnostic Coverage
DED	Dual Error Detection
ECC	Error Correcting Code
ECSM	Error Correction Status Module
eDMA	Enhanced Direct Memory Access
ERRM	Error Out Monitor function
EXWD	External Watchdog function
FCCU	Fault Collection and Control Unit
FMEDA	Failure Modes, Effects and Diagnostic Analysis
FMPLL	Frequency-Modulated Phase-Locked Loop

Table 22.   Acronyms and abbreviations (continued)

Terms	Meanings
GPIO	General Purpose Input/Output
LBIST	Logic Built-In Self-Test
LSM	Lock Step Mode
MBIST	Memory Built-In Self-Test
MC_CGM	Clock Generation Module
MC_ME	Mode Entry
MCU	(Microcontroller Unit)
MPU	Memory Protection Unit
NCF	Non-Critical Fault
NMI	Non-Maskable Interrupt
NVM	Non-Volatile Memory
PMU	Power Management Unit
PSM	Power Supply and Monitor function
PWM	Pulse Width Modulation
RCCU	Redundancy Control Checking Unit
MC_RGM	Reset Generation Module
SAG	Safety Application Guide
SEC	Single Error Correction
SEF	Safety Element Function
SFF	Safe Failure Fraction
SIF	Safety Integrity Function
SIL	Safety Integrity Level
SoR	Sphere of Replication
SWG	Sine Wave Generator
SWT	Software Watchdog Timer

8.3   Document references

● Safety application guide for SPC56xL70xx family reference manual addendum
(TN0983, Doc ID 024257).

9 Revision history

Table 23. Document revision history

Date	Revision	Changes
18-Feb-2013	1	Initial release
17-Sep-2013	2	Updated disclaimer.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

ST PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE ST PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER'S SOLE RISK, EVEN IF ST HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY ST AS BEING INTENDED FOR "AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL" INDUSTRY DOMAINS ACCORDING TO ST PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com