
Safety Manual for MagniV Safety Devices

Document Number: MC9S12ZVxSM
Rev. 4, 02/2018

Review Draft
Confidential Proprietary





Contents

| Section number | Title | Page |
|---|--|------|
| Chapter 1 | | |
| Introduction | | |
| 1.1 | Document Control..... | 5 |
| 1.2 | Vocabulary..... | 5 |
| 1.3 | Preface..... | 5 |
| 1.4 | MagniV Safety Devices Supporting the Safety Concept..... | 7 |
| Chapter 2 | | |
| Functional Safety Concept | | |
| 2.1 | Device Intended Use..... | 9 |
| 2.2 | Safe state..... | 10 |
| 2.3 | Use Context and Item Interface..... | 11 |
| 2.4 | Single-point Fault Tolerant Time Interval | 11 |
| 2.5 | ISO26262 Compliance and Metrics..... | 12 |
| 2.6 | Safety Architecture..... | 13 |
| Chapter 3 | | |
| Hardware and Software Requirements | | |
| 3.1 | Hardware requirements..... | 17 |
| 3.2 | General Information About Software..... | 18 |
| 3.3 | Clock Safety Mechanisms..... | 19 |
| 3.4 | Power Supply Safety Mechanisms..... | 21 |
| 3.5 | Processing Unit Safety Mechanisms..... | 22 |
| 3.6 | Non Volatile Memory Safety Mechanisms..... | 23 |
| 3.7 | Volatile Memory Safety Mechanisms..... | 23 |
| Chapter 4 | | |
| FMEDA | | |
| 4.1 | FMEDA..... | 25 |



Chapter 1

Introduction

1.1 Document Control

1.2 Vocabulary

Unless defined differently, the terms used in this document will have the same definitions as the ones given in standard ISO 26262-1-2011.

In this document, the different MagniV safety devices are referred with the generic identifier "MC9S12ZVxSM".

It is assumed that the user of this document is familiar with the MC9S12ZVx and ISO 26262 standard.

1.3 Preface

This document provides the information required to integrate the MagniV Safety Devices as a safety element out of context in different items.

Please see the chapter [MagniV Safety Devices Supporting the Safety Concept](#) for a list of devices that support the safety concept described here.

The following chapters are included in this document:

- Safety Concept
 - MC9S12ZVx Intended Use
 - Safe state assumptions
 - Use Context and Item Interfaces
 - FTTI
 - ISO26262 Compliance and Target Metrics
 - MCU Safety Architecture

- Principal parts (modules involved in code execution)
- Non Safety Modules
- Allocated Safety Mechanisms
- Hardware and Software Requirements
 - Hardware requirements
 - Safety Mechanisms' Software Configuration
- Failure Modes, Effects and Diagnostic Analysis

The safety assumptions considered during the development of the MagniV Safety Devices appear in the form of paragraphs that start with the tag **Assumption:** this is followed by the assumption, then the end tag [end] and a new line.

Item integration and use requirements appear in the form of paragraphs that start with the tag **Safety requirement:** or **Safety requirement under certain conditions:** this is followed by the requirement, then the end tag [end] and a new line.

If the item do not fulfil a specific requirement, item developers either have to show that their alternative solution is similarly effective as related to the safety requirement in question (for example, provides the same coverage, avoids Common Cause Failure (CCF) as effectively, and so on), or they need to specify the increased failure rate (λ SPF, λ RF, λ MPF,...) and the reduced metrics (SPFM: Single Point Failure Metrics, LFM: Latent Fault Metric) that they estimate will occur due to the deviation.

This document also contains recommendations (or guidelines) on how to configure and operate the MagniV Safety Devices. These guidelines are preceded by one of the following text statements:

- **Recommendation:** A recommendation is a reasonable measure provided when there is no assumption in place. The user has the choice whether or not to adhere to the recommendation.
- **Rationale:** The motivation for a specific recommendation.
- **Implementation hint:** Gives specific hints on the implementation of a recommendation. The user has the choice whether or not to obey the implementation hint.

These recommendations (guidelines) are considered to be useful approaches for the specific topics under discussion. The user will need to use discretion in deciding whether these measures are appropriate for their specific applications.

1.4 MagniV Safety Devices Supporting the Safety Concept

Table 1-1. MagniV Safety Devices

| Family Device Id | Fully Safety Supported Device according to ISO26262:2011, ASIL-A | Safety Enabled Device |
|---|--|-----------------------|
| MC9S12ZVL(S/A) | √ | |
| MC9S12ZVC(A) | √ | |
| MC9S12ZVMC128/64 MC9S12ZVML128/64/32 | | √ |
| MC9S12ZVMC256 MC9S12ZVML31 MC9S12ZVM32/16 | √ | |
| MC9S12ZVMB | √ | |
| MC9S12ZVMA | √ | |
| MC9S12ZVFP | | √ |
| MC9S12ZVH(Y/L) | | √ |

Fully Safety Supported Device according to ISO26262:2011, ASIL-A: The safety architecture is implemented and the device was developed following the ISO26262:2011 development process as a MCU safety element out of context. All work products required by ISO26262:2011, target ASIL-A standard are available. Safety Analysis is available to demonstrate compliance with the ASIL-B, SPFM, LFM and PMHF coverage numbers.

Safety Enabled Devices: The safety architecture is implemented and the safety documentation needed to demonstrate the fulfilment of the assumed requirements on this document is available; however, they were developed following NXP standard development process. Safety Analysis is available to demonstrate compliance with the ASIL-B, SPFM, LFM and PMHF coverage numbers.

Chapter 2

Functional Safety Concept

2.1 Device Intended Use

The MC9S12ZVx was developed as a safety element out of context as defined in section 9 of ISO26262-10. This implies this is a generic part for different applications and customers. This application independent safety function would have to be integrated into an item. (which implements a specific function at the vehicle level).

Assumption: [SM_0001] The MC9S12ZVx shall be used in systems requiring the safety function of code execution. **The safety function is code execution.** [end]

Assumption: [SM_0002] The MC9S12ZVx shall be able to read instructions out of internal Flash, read data from internal RAM or internal Flash, execute instructions, process data and write back result data into internal RAM. [end]

Assumption: [SM_0003] The CPU shall be able to receive interrupt service request, prioritize interrupt service request, calculate instruction address, store context, read respective instructions out of Flash, execute instructions, read data from RAM or flash, process data, and write back result data into RAM and restore context. [end]

Assumption: [SM_0004] Because each application may require the MC9S12ZVx to communicate in a different way with the item, the failure modes and detection mechanisms of the selected I/O modules (used to communicate with the item) shall be analyzed by the user of the MC9S12ZVx and are not analysed by NXP. [end]

Assumption: [SM_0005] Writing to Flash EEPROM shall not be considered a safety function as it shall be performed in a safe and controlled maintenance environment, with the Flash EEPROM content validated multiple times before bringing back to operation. Only intermittent faults and not diagnostic coverage, for example due to weak programming, might cause a safety threat. This threat is a maintenance issue in principle and not a random fault field issue. [end]

2.2 Safe state

Safe state of the system is named Safe state_{item}, whereas a Safe state of the MC9S12ZVx is named Safe state_{MCU}.

The assumptions related to the safe state of the item are:

Assumption: [SM_0006] The item shall operate without unreasonable risk of harm if the MagniV Safety device is operating without unreasonable risk of harm. [end]

Assumption: [SM_0007] If the MC9S12ZVx is un-powered the item shall transition to a Safe state_{item}. [end]

The assumptions related to the safe state of the MC9S12ZVx are:

- **Assumption:** [SM_0008] If the parts involved in the execution of code are operating fault free and Non Safety Modules are not interfering with the code execution, the MC9S12ZVx shall be considered to be in a Safe state_{MCU}. This is the fault free normal operating state. [end]
- **Assumption:** [SM_0009] During the time the MC9S12ZVx is in Non-POR reset, the device shall be considered to be in a Safe state_{MCU}. [end]
- **Assumption:** [SM_0010] During the time the MC9S12ZVx is in POR the device shall be considered to be in a Safe state_{MCU}. [end]

The MC9S12ZVx is assumed not to be in a Safe state_{MCU} when:

- **Assumption:** [SM_0011] During the time the MC9S12ZVx is starting up and shutting-down, the MC9S12ZVx shall NOT be assumed to be in a Safe state_{MCU}. [end]
- **Assumption:** [SM_0012] If the MC9S12ZVx continuously switches between run mode and reset state, it shall NOT be considered to be in a Safe state_{MCU}. [end]

Safety requirement: [SM_1001] During the MC9S12ZVx initialization and shut down, the item must take the necessary actions to maintain Safe state_{item}. [end]

Safety requirement: [SM_1002] The item must be designed in a way it is operating without an unreasonable risk of harm when the MC9S12ZVx is in reset state. (I/O pins in high impedance with the internal pull ups and pull downs enabled/disabled as listed in the PIM chapter in the Reference Manual) [end]

Recommendation: On some applications it is recommended to take the necessary actions to switch the item to a safe state in case the MC9S12ZVx is detected to be continuously switching between reset and normal operating mode.

Recommendation: It is recommended to design the item in a way it is operating without an unreasonable level of risk when the MC9S12ZVx is not powered (VDDX=0V).

2.3 Use Context and Item Interface

Assumption: [SM_0013] The MC9S12ZVx shall be used in automotive applications for use cases requiring a fail-silent code execution function. [end]

Assumption: [SM_0015] The latest MC9S12ZVx errata shall be taken into account during item design, implementation, and maintenance. [end]

Assumption: [SM_0016] The MC9S12ZVx shall be used in applications where the battery voltage is maximum 18 V in normal operation (12 V Nominal). [end]

Assumption: [SM_0017] The MC9S12ZVx shall be able to handle transient conditions like ESD pulses or load dump conditions as specified in the data sheet section of the Reference Manual. [end]

Assumption: [SM_0018] The MC9S12ZVx shall be used in applications that require continuous operation without power removal, setting the MC9S12ZVx in standby when needed. [end]

2.4 Single-point Fault Tolerant Time Interval

The single-point Fault Tolerant Time Interval (FTTI) is the time span in which a single point fault can be present in a system before a hazardous event occurs. It is used to define the sum of worst case fault detection time and the time for execution of a corresponding countermeasure (reaction). The figure below shows the FTTI for a single-point fault in a system.

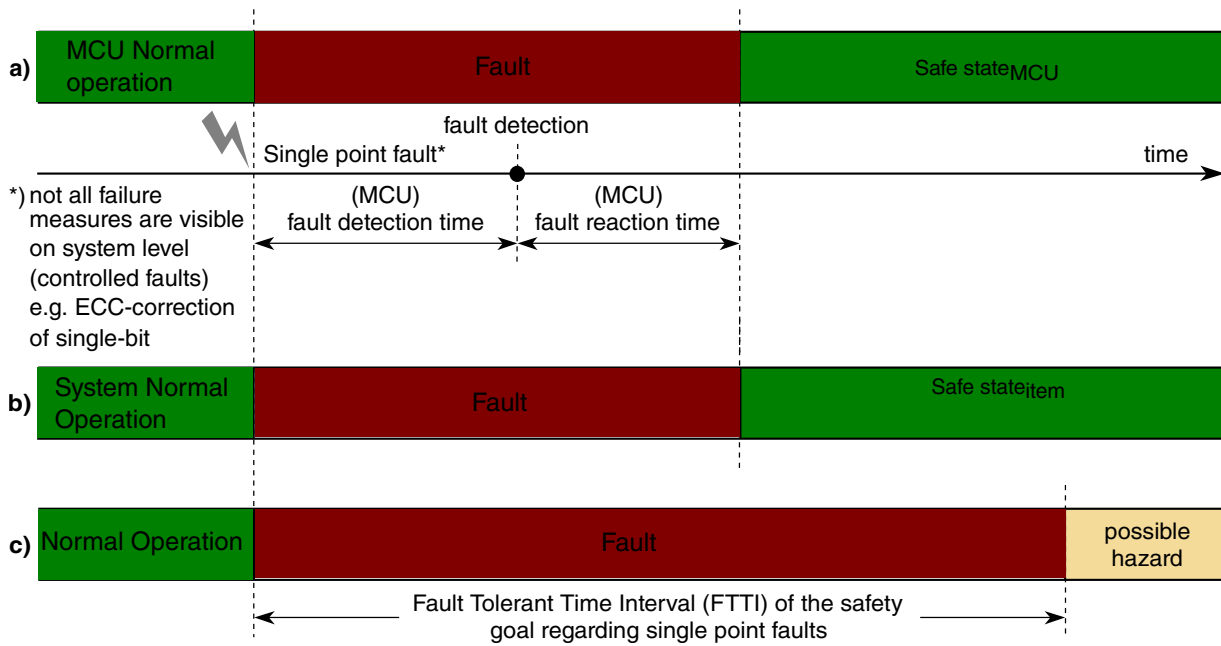


Figure 2-1. Fault tolerant time interval for single point faults

The assumed MC9S12ZVx safety goal is:

Assumption: [SM_0020] Once the MC9S12ZVx has left its Safe state_{MCU} due to a detectable single point fault, it shall automatically return to its Safe state_{MCU} in 100 ms or less. [end]

2.5 ISO26262 Compliance and Metrics

Assumption: [SM_0021] The MC9S12ZVx shall be developed in accordance with the process in ISO26262-10 Safety element out of Context (SEooC) hardware component. [end]

Assumption: [SM_0022] The MC9S12ZVx shall be usable for all applications in safety relevant systems. [end]

Assumption: [SM_0023] At least a 60% single point fault metric coverage for each part involved in the code execution shall be targeted. [end]

Assumption: [SM_0024] There shall not be a minimum target for latent fault metric coverage on any part of the device. [end]

Assumption: [SM_0025] The sum of the single point and residual fault failure rates shall be less than 1×10^{-8} failures/hour. [end]

Table 2-1. ISO26262 and Metrics Compliance

| Assumed Requirement | Fully Safety Supported Devices Compliance | Safety Enabled Devices Compliance |
|--|---|-----------------------------------|
| ISO26262 Development Process Compliance | √ | No |
| Usable in all safety relevant applications | √ | √ |
| Fulfills assumed minimum single point fault metric coverage requirement | √ | √ |
| Fulfills assumed minimum latent fault metric coverage requirement | √ | √ |
| Fulfills assumed maximum single point plus residual failure rate requirement | √ | √ |

2.6 Safety Architecture

Assumption: [SM_0026] The MC9S12ZVx five principal parts needed in the execution of code shall be:

1. Processing unit -S12Z CPU
2. Power supply -5 V Voltage Reg
3. Clock
4. Non-volatile memory -Flash and EEPROM
5. Volatile memory -RAM

[end]

Assumption: [SM_0027] The Non Safety Modules can interfere with execution of code and shall NOT be operated during the execution of safety code. These modules are:

1. Debug Module: May halt the processing unit or alter the result of an operation by altering one or more CPU registers or memory locations within the address space. Active BDM may also halt (or freeze) the watchdog. It also, allows entering the corresponding test modes as per the TEST registers.
2. Test Module: Used for comprehensive factory testing and may interfere with the proper operation of the parts used for code execution.

[end]

Assumption: [SM_0028] During run mode the hardware safety mechanisms active shall be:

1. External Oscillator Stuck at Monitor if the external pierce oscillator is used -Ext Osc. Clock Monitor
2. PLL output clock Stuck at Monitor -PLL Clock Monitor
3. PLL input clock stuck at -PLL Lock Interrupt request
4. Watchdog with separate time base and time window -COP
5. Volatile memory monitoring using error detection correction code EDC -ECC

6. Non Volatile memory monitoring using error detection correction code EDC -ECC
7. Under voltage detection -LVR monitors VDDX, VDD, VDDF
8. Under voltage detection -LVI monitors VDDA
9. Under voltage input detection -BATS

[end]

Assumption: [SM_0029] During stop mode the hardware safety mechanisms active shall be:

1. Under voltage Detection -POR

[end]

Assumption: [SM_0030] During run mode, the parts in the architecture shall relate as follows:

- The parts used in the execution of code shall run item code.
- The non safety modules shall be inactive and they shall not interfere with the code execution or safety mechanisms.
- The corresponding safety measures shall be configured and operating.

[end]

Assumption: [SM_0031] During stop mode, the parts in the architecture shall relate as follows:

- The processing unit shall be waiting to be awoken.
- The non safety modules shall be inactive such as they are not interfering with the parts involved in code execution or safety mechanisms.
- The corresponding safety measures shall be configured and operating.

[end]

In the figure below it can be seen the hardware part included in the MC9S12ZVx:

- In blue the parts used to perform the safety function
- In pink the Non Safety Modules
- In green the safety integrity hardware measures
- In yellow the modules that are used by the hardware integrity measures

MagniV Safety Related Modules

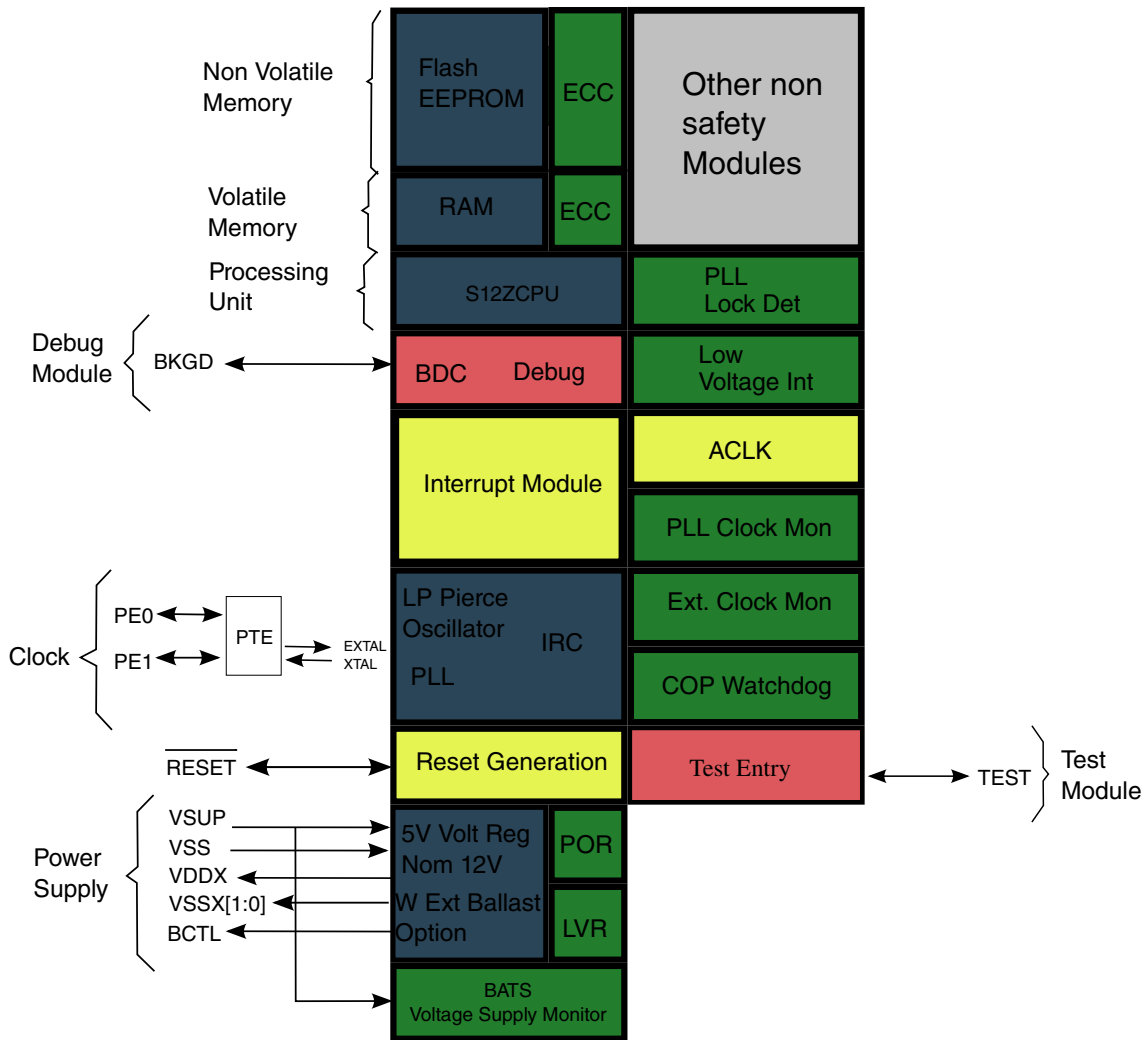


Figure 2-2. Safety Architecture Diagram
 Safety Manual for MagniV Safety Devices, Rev. 4, 02/2018

Table 2-2. Architecture Compliance

| Assumed Requirement | Fully Safety Supported Devices Compliance | Safety Enabled Devices Compliance |
|--|---|-----------------------------------|
| Flash with ECC | √ | √ |
| RAM with ECC | √ | √ |
| S12ZCPU with interrupt and machine exception capabilities | √ | √ |
| Clock Distribution (Pierce Oscillator, PLL, IRC) | √ | √ |
| Internal Voltage Regulator | √ | √ |
| Watchdog with independent clock source and optional window operation | √ | √ |
| Power Supply Safety Measures | √ | √ |
| Clock Distribution Safety Measures | √ | √ |
| Core Self Test Software Available ¹ | √ | √ |
| Interrupt module with programmable priorities | √ | √ |

1. a core self test for ASIL level A or B is available

The requirements related to the architecture are as follows:

Safety requirement: [SM_1003] The Non Safety Modules can interfere with the code execution or safety measures; thus, they must be inactive during the code execution. [end]

Safety requirement: [SM_1004] Safety Integrity measures must be configured and operating while executing safety code. [end]

Chapter 3

Hardware and Software Requirements

3.1 Hardware requirements

This section lists required or recommended hardware measures.

Safety requirement: [SM_1005] Pin TEST must be connected to ground. During the reset de assert, this pin must be in logic state 0 to avoid entering test mode. [end]

Safety requirement: [SM_1006] A 4.7 K ohms to 10 K ohms pull up resistor between VDDX (5 V) and pin BKGD/MODC must be installed. This pull up resistor prevents entering Special single chip mode (Enable the debug functionalities). [end]

Safety requirement: [SM_1007] VDDA and VDDX must be shorted on the board. Internally are shorted but the internal impedance may be larger than the external trace. [end]

Safety requirement: [SM_1008] VSSA, VSS, LGND, VSSX1 and VSSX2 must be shorted on the board. Internally are shorted but the internal impedance may be larger than the external trace. [end]

Recommendation: It is recommended to avoid placing capacitors with a value larger than 680 pF between RESET pin and GND.

Recommendation: If it is necessary place an external resistor consider the internal resistor of 10K Ω to satisfy a maximum value of $R \cdot C = 7.62 \mu s$ relationship

Rationale: Flags COPRF, OMRF and PMRF in CPMURFLG may not set if the reset line is held low by the capacitor for more than 8 μs . The MC9S12ZVxs will interpret this as a reset caused by an external device.

Recommendation: Decouple the ballast transistor with low ESR (around 0.01 ohms) 10nF and 1nF ceramic caps.

Recommendation: In case the period of the API or the GPIO 5 V operating characteristics need to be assured during **stop** mode, it is recommended to use an external POR module triggered at the low voltage required. (e.g. API requires VDDA to be larger than 3 V to fulfil reference manual specs)

3.2 General Information About Software

This section lists the general assumptions, requirements, and recommendations related to software.

Assumption: [SM_0032] Interrupt request prioritization shall be used in the application software to ensure the detection of faults within the fault tolerant time. [end]

Assumption: [SM_0033] The watchdog timer shall be used in the application to detect faults. Use and size of the time window is application dependant. [end]

Assumption: [SM_0034] An ECC fault injection mechanism in the MC9S12ZVx shall be included to allow software unit test (during debug time) of the Non volatile memory and volatile memory fault handler routines. (See the description of FCNFG and the chapter ECC Debug Behavior in the reference manual for more information). [end]

Safety requirement: [SM_1009] The ECC fault injection mechanisms (for NVM and volatile memory) **must be used only during debug mode** to trigger the ECC handler routines. They must NOT be used as a measure against random faults in the application. (e.g. to check the integrity of the ECC safety mechanisms). They were included to make easier the software unit test of the handler routines. [end]

Safety requirement: [SM_1010] Interrupt requests associated with the safety mechanisms (e.g. the low voltage interrupt request or PLL lock interrupt request), must be configured to have a higher interrupt request priority than non safety related interrupt requests. [end]

Safety requirement under certain conditions: [SM_1011] Interrupt service routines that last more than 100 ms must be configured to allow nested interrupts. Use the instruction CLI inside time consuming non safety related interrupt service routines to allow the execution of safety interrupt requests within the FTTI. [end]

Safety requirement: [SM_1012] The reset vector must have a handler where the system application will be initialized. [end]

Safety requirement: [SM_1013] Bit I in S12Z' CCR register must be cleared before starting to execute code to enable maskable interrupt requests. This can be cleared by using instruction CLI. [end]

Recommendation: It is recommended to create a default interrupt handler for all unused interrupts.

Recommendation: It is recommended to create an interrupt handler for all software exceptions (SWI, TRAP, SPARE, SYS).

Recommendation: It is recommended to leave register bit RSBCK in its reset value (not to freeze).

Rationale: Even in the case where the device enters active debug mode, the COP will continue to operate.

Recommendation: After the MC9S12ZVxs has reset, it is recommended to write register MODE with 0x80.

Rationale: Even in the case the MC9S12ZVxs entered by accident in Special Single Chip mode, the MC9S12ZVx will return to Normal Single Chip mode.

Recommendation: It is recommended to enable the high temperature interrupt request.

Rationale: Such interrupt handler can notify the item that the MagniV Safety Device may permanently damage if the temperature continues to rise.

3.3 Clock Safety Mechanisms

This section lists the assumptions, requirements, and recommendations for the safety mechanisms related to the clock part.

Assumption: [SM_0035] The following safety mechanisms shall be included in the MC9S12ZVx to detect clock faults:

- PLL Clock Monitor Reset: Resets the MC9S12ZVxs when the PLL is detected to be operating at a very low frequency or not operating at all (VCO stops). See CPMU PLL Clock Monitor Reset chapter at the reference manual for more information.
- PLL Input Clock Loss Detection: PLL LOCK Interrupt request. Can interrupt the processing unit when the PLL input clock has been detected to be stuck.
- External Oscillator Clock Monitor Reset: Resets the MC9S12ZVxs when the external oscillator is detected to be operating at a very low frequency or not operating at all. See CPMU Oscillator Clock Monitor Reset chapter at the reference manual for more information.
- Watchdog with separate time base and optional time window: Resets the MC9S12ZVx when the watchdog times out or is not triggered inside the time window. See CPMU Computer Operating Properly Watchdog (COP) Reset at the Reference Manual for more information.

[end]

Safety requirement under certain conditions: [SM_1014] When using the PLL in a Bypassed External Mode, the external oscillator clock monitor reset must be enabled. Bit OMRE must be set to 1 in register CPMUOSC2 when in PBE. MC9S12ZVxs may stall in case of a sudden loss of the external oscillator if the external oscillator clock monitor is not enabled. See the description of OMRE bit in CPMUOSC2 in the reference manual. [end]

Safety requirement under certain conditions: [SM_1015] The reset configuration to insert the **maximum** flash wait states during read operations must not be changed unless the bus clock frequency is configured to operate at a frequency below or equal than 25 MHz. See the description of bits WSTAT[1:0] in the FCNFG register and the description of bit WSTATACK in FPSTAT in the reference manual. [end]

Safety requirement under certain conditions: [SM_1016] The PLL must be locked before using the bus clock for time base operations. [end]

Safety requirement: [SM_1017] After the PLL has locked, the LOCK interrupt request must be enabled by clearing the LOCKIF flag and enabling the corresponding interrupt request enable bit. This must be done by the application. See the description of register CPMUIFLG in the Reference Manual for more information. [end]

Safety requirement: [SM_1018] The PLL LOCK interrupt handler must wait for t_{lock} seconds for the PLL to lock again. If the PLL doesn't lock in such period of time, the software must reset the MC9S12ZVxs. If the PLL locks within t_{lock} seconds, the software must clear the LOCKIF again and exit the interrupt handler. See `ipll_1vdd_1l18` Characteristics and Phase Locked Loop with Internal Filter (PLL) in the reference manual for more information. [end]

Safety requirement under certain conditions: [SM_1019] If the watchdog is configured to be clocked with the ACLK but the ACLK COP input clock is gated off during stop mode (CSAD=1), the application must wait for 2 ACLK cycles (100 μ s) before executing the STOP instruction again. This period of time (100 μ s) must be measured from the time the MagniV Safety device starts to execute the first instruction in the interrupt service routine until the STOP instruction starts to execute. If this is not done the operation of the API or COP is not guaranteed. See CPMUCLKS register description in the Reference Manual for more information. [end]

Safety requirement: [SM_1020] The watchdog timer must be used in the application to detect faults in the behavior and plausibility of the program sequence. [end]

Safety requirement: [SM_1021] The watchdog time out period must be set such that it can detect and react to program sequence faults within the FTTI. [end]

Recommendation: It is recommended to write the flash non volatile byte inside the configuration register to automatically load the watchdog during RESET.

Rationale: In case the initialization of the application stalls (before configuring the watchdog by software), the MC9S12ZVx will reset.

Recommendation: If the ACLK clock source is used to clock the watchdog, it is recommended to leave the watchdog on during stop mode. (CSAD bit in CPMUCLKS is recommended to be 0)

Recommendation: Once the bus clock has been configured, it is recommended to protect the register against changes by writing the CPMUPROT register. See the CPMUPROT description in the Reference Manual.

3.4 Power Supply Safety Mechanisms

This section lists the assumptions, requirements, and recommendations for the safety mechanisms related to the power part.

Assumption: [SM_0036] The following safety mechanisms shall be included in the MC9S12ZVx to detect power faults:

- **Input voltage monitoring:** Can interrupt the processing unit in case a low input voltage is detected. See Supply Voltage Sensor in the Reference Manual for more information.
- **Low Voltage Reset:** Resets the MC9S12ZVxs when the power supply output voltage is below the minimum voltage level that is required to operate the MC9S12ZVxs at the maximum bus clock speed (during run mode).
- **Power on Reset:** Performs a Power on Reset during the power ramp up and assures the power supply output voltage is large enough for the volatile memory and register to retain their state during stop mode.
- **Low Voltage Interrupt:** Can interrupt the processing unit when the voltage at VDDA is detected to be smaller than a threshold. VDDA is used to provide several voltage references. See the low voltage interrupt data sheet specifications in the Reference manual for more information.

[end]

Safety requirement: [SM_1022] The low voltage interrupt request must be enabled.

[end]

Safety requirement: [SM_1023] Inside the low voltage interrupt handler, the software application must prepare the device to enter a low voltage safe state (by entering a LVR or POR state). Actions are application dependant. Please see VREG Electrical Specifications in the reference manual for more information about the LVI assert value. [end]

Safety requirement: [SM_1024] The item initialization software must initialize the CPMUVREGCTL register according to the item power supply hardware configuration. More information at the voltage regulator control register description. [end]

Recommendation: When using the internal voltage regulator, it is recommended to configure the Supply Voltage Sensor to assert an interrupt request when the input to the voltage regulator is below the LBI2 level and enable the corresponding interrupt request.

Rationale: When this interrupt request asserts, the internal voltage regulator could be operating with reduced current capabilities. See the VREG Electrical Specifications and the BATS Electrical Specifications in the device's Reference Manual.

Recommendation: When using the external ballast transistor to regulate externally the power, it is recommended to power the ballast transistor with the same net as VSUP.

Rationale: To allow monitoring the input voltage with the supply voltage sensor.

3.5 Processing Unit Safety Mechanisms

This section lists the assumptions, requirements, and recommendations for the safety mechanisms related to the processing unit part.

Assumption: [SM_0037] Runtime patterns shall be run by the item application software to check the integrity of the processing unit. These self-test patterns are sweeps of the instruction set. The diagnostic coverage of these sweeps is estimated to be 60%. [end]

Assumption: [SM_0051] This Assumption is valid for items targeting ASIL-B level only. Runtime patterns shall be run by the item application software to check the integrity of the processing unit. These self-test patterns are sweeps of the instruction set. The advanced Core Self-Test SW with the estimated diagnostic coverage of 90% must be used. [end]

Assumption: [SM_0038] The correct sequence of the individual program sections shall be monitored by item application software only. (Logical monitoring of program sequence will be done in software only) [end]

Recommendation: It is recommended that during code execution, a processing unit self test software is executed. This software will detect failures in the processing unit and its execution frequency is application dependant.

Recommendation: It is recommended to implement a periodic stack over/under flow detection mechanism. For example as described in appendix D section D.2.3.8 of ISO26262-5-2011. The implementation needs to be done in software only.

NOTE

NXP developed a core self test software for the S12Z core and its availability and manual is published in <http://www.NXP.com> under the name S12Z_FCST_Safety_Manual or please contact your local NXP Sales. A core self test for ASIL level A or B is also available.

3.6 Non Volatile Memory Safety Mechanisms

This section lists the assumptions, requirements, and recommendations for the safety mechanisms related to the Non volatile Memory part.

Assumption: [SM_0040] A memory monitoring using error-detection-correction EDC(known as ECC in the Reference Manual) shall be included to detect NVM faults. [end]

Assumption: [SM_0041] The ECC safety mechanism shall generate a maskable interrupt request (with independent enable bit) when a single bit fault has been corrected by the ECC module. There is no software reaction needed to maintain the Safe state_{MCU} when a single bit error occurs. [end]

Assumption: [SM_0042] The ECC safety mechanism shall generate a non maskable machine exception request when an uncorrectable memory corruption has been found while the processing unit is reading an instruction or data in the EEPROM or P-Flash. [end]

Safety requirement: [SM_1027] The machine exception handler must reset the MC9S12ZVxs. For example by attempting to write a value different than 0x55 or 0xAA to the CPMUARMCOP register. See S12CPMU_UHV_V5 COP Timer Arm/Reset Register (CPMUARMCOP) in the reference manual for more information. [end]

3.7 Volatile Memory Safety Mechanisms

This section lists the required configurations for the safety mechanisms.

Assumption: [SM_0043] A memory monitoring using error-detection-correction (known as ECC in the Reference Manual) shall be included to detect RAM faults. [end]

Assumption: [SM_0044] The ECC safety mechanism shall generate a maskable interrupt request (with independent enable bit) when a single bit fault has been automatically corrected. There is no software reaction needed to maintain a Safe state_{MCU} when an automatic correction occurs. [end]

Assumption: [SM_0045] The ECC safety mechanism shall generate a non maskable machine exception when an uncorrectable memory corruption has been found while the processing unit is reading an instruction or data in RAM. [end]

Safety requirement: [SM_1028] There are no software requirements regarding the ECC for the volatile memory. During each POR, the RAM content will be automatically initialized to 0 along with the ECC information. Access to the SRAM is disabled (cpu or peripheral waits) until the complete memory has finished the initialization process. See the ECC Generation Module in the reference manual for more information. [end]

Chapter 4

FMEDA

4.1 FMEDA

This section lists the assumptions related to the FMEDA.

Assumption: [SM_0046] Each MC9S12ZVx shall have a FMEDA for each of the parts involved in the code execution. [end]

Assumption: [SM_0047] Each MC9S12ZVx shall have a fault metrics analysis for each part involved in the code execution to support writing the item FMEDA. [end]

Assumption: [SM_0048] The diagnostic coverage analysis shall concentrate on the failure modes and safety mechanisms listed in ISO26262-5 Annex D for the parts involved in code execution. [end]

Assumption: [SM_0049] Only faults or failures categorized as low in ISO26262-5 Annex-D Table D1 shall be analyzed in the derivation of diagnostic coverage. [end]

Assumption: [SM_0050] The safety mechanisms requirements described in this document shall be fulfilled during the FMEDA analysis. [end]

Table 4-1. FMEDA Compliance

| Assumed Requirement | Fully Safety Supported Devices Compliance | Safety Enabled Devices Compliance |
|-------------------------|---|-----------------------------------|
| FMEDA Available | √ | √ |
| Fault Metrics Available | √ | √ |

All assumed targets were fulfilled, please ask your NXP Semiconductors representative for the updated FMEDA report.

Appendix A

Acronyms and abbreviations

A short list of acronyms and abbreviations used in this document is shown in the table below.

Table A-1. Acronyms and abbreviations

| Terms | Meanings |
|---------|--|
| ACLK | Autonomous Clock |
| ADC | Analog to Digital Converter |
| API | Autonomous Periodic Interrupt |
| BATS | Supply Voltage Sensor |
| BDC | Background Debug Controller |
| COP | Computer Operating Properly Watchdog |
| CSM | Computational Shell Modules |
| DBG | Debug Module |
| ECC | Error Correcting Code |
| EDC | Error Detection Correction |
| FMEDA | Failure Modes, Effects and Diagnostic Analysis |
| FTTI | Fault Tolerant Time Interval |
| GPIO | General Purpose Input and Output |
| HTOL | High Temperature operating life test |
| IIC | Inter-IC Module |
| INT | Interrupt Control Module |
| IRC | Internal 1MHz RC Oscillator |
| LF | Latent Fault |
| LVR | Low Voltage Reset |
| L-FTTI | Latent-Fault Tolerant Time Interval |
| MCU | Micro Controller Unit |
| NDA | Non Disclosure Agreement |
| NMI | Non-Maskable Interrupt |
| NON-POR | Non Power On Reset |
| NoSaMos | Non-Safety Modules |
| NVM | Non-Volatile Memory |

Table continues on the next page...

Table A-1. Acronyms and abbreviations (continued)

| Terms | Meanings |
|--------------|--------------------------------|
| PIM | Port Integration Module |
| PLL | Phase Locked Loop |
| POR | Power On Reset |
| PWM | Pulse Width Modulation |
| RF | Residual Fault |
| RTI | Real Time Interrupt |
| SCI | Serial Communication Interface |
| SEooC | Safety element out of Context |
| SF | Safe Fault |
| SIF | Safety Integrity Function |
| SM | Safety Manual |
| SPF | Single-Point Fault |
| SPFM | Single point fault metric |
| SPI | Serial Peripheral Interface |
| TIM | Timer Module |

Appendix B

Release Notes

Table B-1. Revision history

| Revision | Date | Description of Change |
|----------|------------------|---|
| 1 | 28 April 2014 | Initial document release |
| 2 | 4 August 2015 | <ul style="list-style-type: none"> • Minor editorial updates throughout • Updated front page for new families Updated table: "MagniV Safety Devices" for new families • Added topic: Faults and failures • Updated topic: "Hardware requirements" • Updated topic: "Processing Unit Safety Mechanisms" |
| 3 | 17 October 2017 | <ul style="list-style-type: none"> • Renamed the second column of the "MagniV Safety Devices" table to read "Fully Safety Supported Device according to ISO26262:2011, ASIL-A." • In section "MagniV Safety Devices Supporting the Safety Concept," renamed the first bullet to read "Fully Safety Supported Device according to ISO26262:2011, ASIL-A." Also updated "ISO26262" to read "ISO26262:2011." Additionally, added text related to safety analysis. • Removed assumption [SM_0014] from the "Use Context and Item Interface" section. • Added a note related to core self test to the "Core Self Test Software Available" assumed requirement in the table titled "Architecture Compliance." • Added text to the note in the "Processing Unit Safety Mechanisms" topic. This text is related to core self test. |
| 4 | 02 February 2018 | Added information to the note in Section 3.5 Processing Unit Safety Mechanisms. The revised note reads this: NXP developed a core self test software for the S12Z core and its availability and manual is published in http://www.NXP.com under the name S12Z_FCST_Safety_Manual or please contact your local NXP sales. A core self test for ASIL level A or B is also available. |



How to Reach Us:**Home Page:**nxp.com**Web Support:**nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2017 NXP B.V.

Document Number MC9S12ZVxSM
Revision 4, 02/2018

