XMC4000

Microcontroller Series
for Industrial Applications

Fail-safe Features

✓ Voltage Supervision
✓ Clock Supervision
✓ Memory Integrity
✓ Fail-safe Flash
✓ Software Supervision
✓ System Traps
✓ Special Peripheral Features

Application Guide

V1.0 2013-04

Microcontrollers

Edition 2013-04

Published by
Infineon Technologies AG
81726 Munich, Germany

© 2013 Infineon Technologies AG
All Rights Reserved.

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Revision History

Page or Item	Subjects (major changes since previous revision)
V1.0, 2013-04	

Trademarks of Infineon Technologies AG

AURIX™, C166™, CanPAK™, CIPOS™, CIPURSE™, EconoPACK™, CoolMOS™, CoolSET™, CORECONTROL™, CROSSAVE™, DAVE™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPIM™, EiceDRIVER™, eupec™, FCOS™, HITFET™, HybridPACK™, I²RF™, ISOFACE™, IsoPACK™, MIPAQ™, ModSTACK™, my-d™, NovalithIC™, OptiMOS™, ORIGA™, PRIMARION™, PrimePACK™, PrimeSTACK™, PRO-SIL™, PROFET™, RASIC™, ReverSave™, SatRIC™, SIEGET™, SINDRION™, SIPMOS™, SmartLEWIS™, SOLID FLASH™, TEMPFET™, thinQ!™, TRENCHSTOP™, TriCore™.

Other Trademarks

Advance Design System™ (ADS) of Agilent Technologies, AMBA™, ARM™, MULTI-ICE™, KEIL™, PRIMECELL™, REALVIEW™, THUMB™, µVision™ of ARM Limited, UK. AUTOSAR™ is licensed by AUTOSAR development partnership. Bluetooth™ of Bluetooth SIG Inc. CAT-iq™ of DECT Forum. COLOSSUS™, FirstGPS™ of Trimble Navigation Ltd. EMV™ of EMVCo, LLC (Visa Holdings Inc.). EPCOS™ of Epcos AG. FLEXGO™ of Microsoft Corporation. FlexRay™ is licensed by FlexRay Consortium. HYPERTERMINAL™ of Hilgraeve Incorporated. IEC™ of Commission Electrotechnique Internationale. IrDA™ of Infrared Data Association Corporation. ISO™ of INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. MATLAB™ of MathWorks, Inc. MAXIM™ of Maxim Integrated Products, Inc. MICROTEC™, NUCLEUS™ of Mentor Graphics Corporation. Mifare™ of NXP. MIPI™ of MIPI Alliance, Inc. MIPS™ of MIPS Technologies, Inc., USA. muRata™ of MURATA MANUFACTURING CO., MICROWAVE OFFICE™ (MWO) of Applied Wave Research Inc., OmniVision™ of OmniVision Technologies, Inc. Openwave™ Openwave Systems Inc. RED HAT™ Red Hat, Inc. RFMD™ RF Micro Devices, Inc. SIRIUS™ of Sirius Satellite Radio Inc. SOLARIS™ of Sun Microsystems, Inc. SPANSION™ of Spansion LLC Ltd. Symbian™ of Symbian Software Limited. TAIYO YUDEN™ of Taiyo Yuden Co. TEAKLITE™ of CEVA, Inc. TEKTRONIX™ of Tektronix Inc. TOKO™ of TOKO KABUSHIKI KAISHA TA. UNIX™ of X/Open Company Limited. VERILOG™, PALLADIUM™ of Cadence Design Systems, Inc. VLYNQ™ of Texas Instruments Incorporated. VXWORKS™, WIND RIVER™ of WIND RIVER SYSTEMS, INC. ZETEX™ of Diodes Zetex Limited.

Last Trademarks Update 2011-02-24

Table of Contents

Voltage Supervision

1 Voltage Supervision

Voltage Supervision fail-safe features enable monitoring of the voltage levels and automatic reaction of the system to abnormal supply conditions in the application.

1.1 Introduction

The following fail-safe features are described in the subsequent sections:

- Embedded Voltage Regulator
- Power-on Reset
- PORST pin
  - Power Validation
  - Supply Watchdog
- Supply Voltage Brown-out Detection
  - Temporary Loss of V_{DDP} Supply
  - Prevention of Premature Coin Battery Discharging
- Hibernate Domain Power Management

1.2 Embedded Voltage Regulator

The Embedded Voltage Regulator (EVR) generates the core voltage V_{DDC} out of the external supplied voltage V_{DDP}. A power-on reset is automatically generated on power-up when V_{DDP} and V_{DDC} have reached valid levels. The EVR provides a Supply Watchdog (SWD) for the input voltage V_{DDP}. The generated core voltage V_{DDC} is monitored by a power validation circuit (PV).

1.3 Power-on Reset

The EVR starts operation as soon as V_{DDP} is above the defined minimum level.

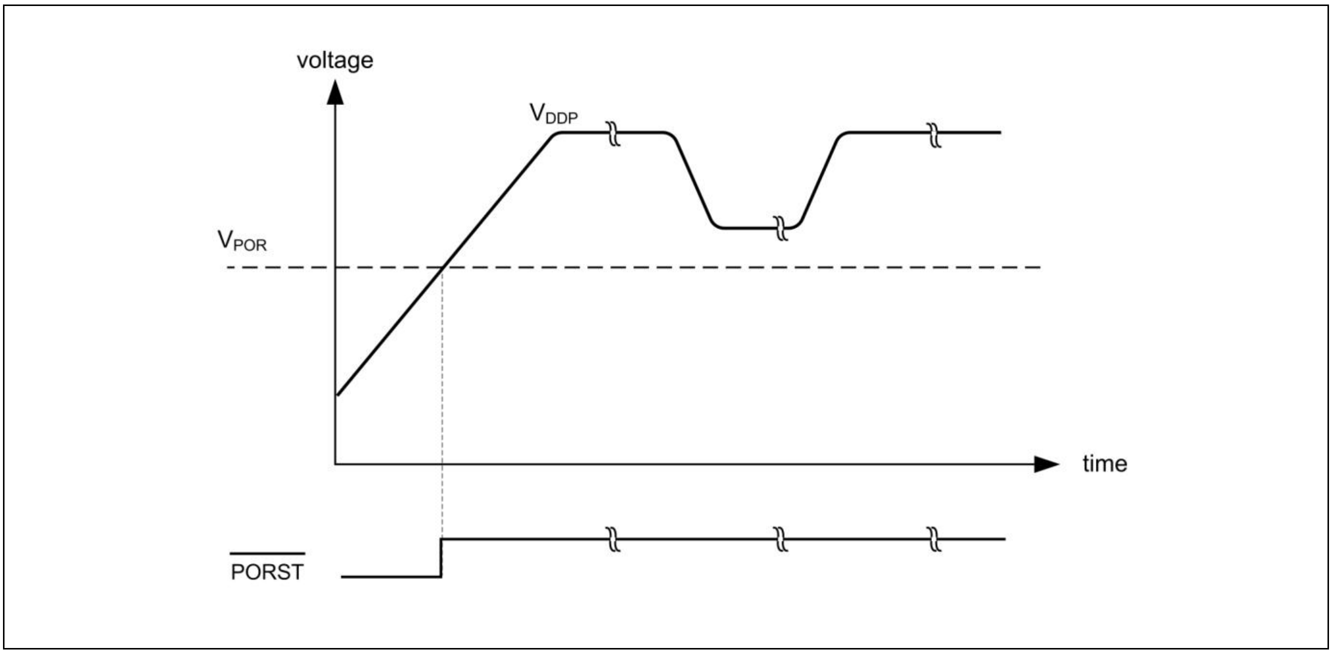

Figure 1 Power-on Reset

The system is released from PORST reset when the external voltage V_{DDP} and the generated voltage V_{DDC} are above the reset thresholds and have reached the nominal values (see Figure 1).

Failure condition	During power-up V_{DDP} voltage still too low to ensure proper function of modules supplied with V_{DDP}, which may potentially lead to for example to Flash malfunction, I/O pads misbehavior, or unreliable V_{DDC} generation in EVR.
Fail-safe effect	PORST reset remains active preventing any activity when VDDP is too low.

1.4 PORST pin

The XMC4000 devices allow an unlimited long ramping of V_{DDP} voltage without any side effects. However, if the XMC4000 device is the reset master on the application board, it is required to ensure a fast PORST reset release (below 2 µs), as specified in the Datasheet. In some application scenarios this requirement may impose limitations that can be easily overcome.

The bi-directional PORST pin needs to be able to overcome an external load imposed by other components on the application board. This may be realized with a pull-up resistor to V_{DDP} on the PORST pin. Typically it would be sufficient to apply a 90kΩ pull-up resistor if no other application components on the board are connected to the PORST pin. The pull-up resistance needs to be adapted to the application constraints and in case of a greater load imposed on the PORST pin, a stronger pull-up may need to be applied (see example scenario in Figure 2). This requirement applies in the case of V_{DDP} ramping times longer than 500 µs. A pull-down resistance and/or capacitor to V_{SS} may severely degrade the ramping release speed of the PORST and result in multiple low spikes on the PORST.

Note: *XMC4000 devices do not require an external capacitor to filter noise on the PORST pin as they implement an on-chip spike filter.*

Exceptions

The PORST ramping requirement does not apply when:

- V_{DDP} ramping is faster than 500 µs
- the XMC4000 device is not a reset master on the application board and is held in reset state by an external reset control device until the power-up sequence is completed

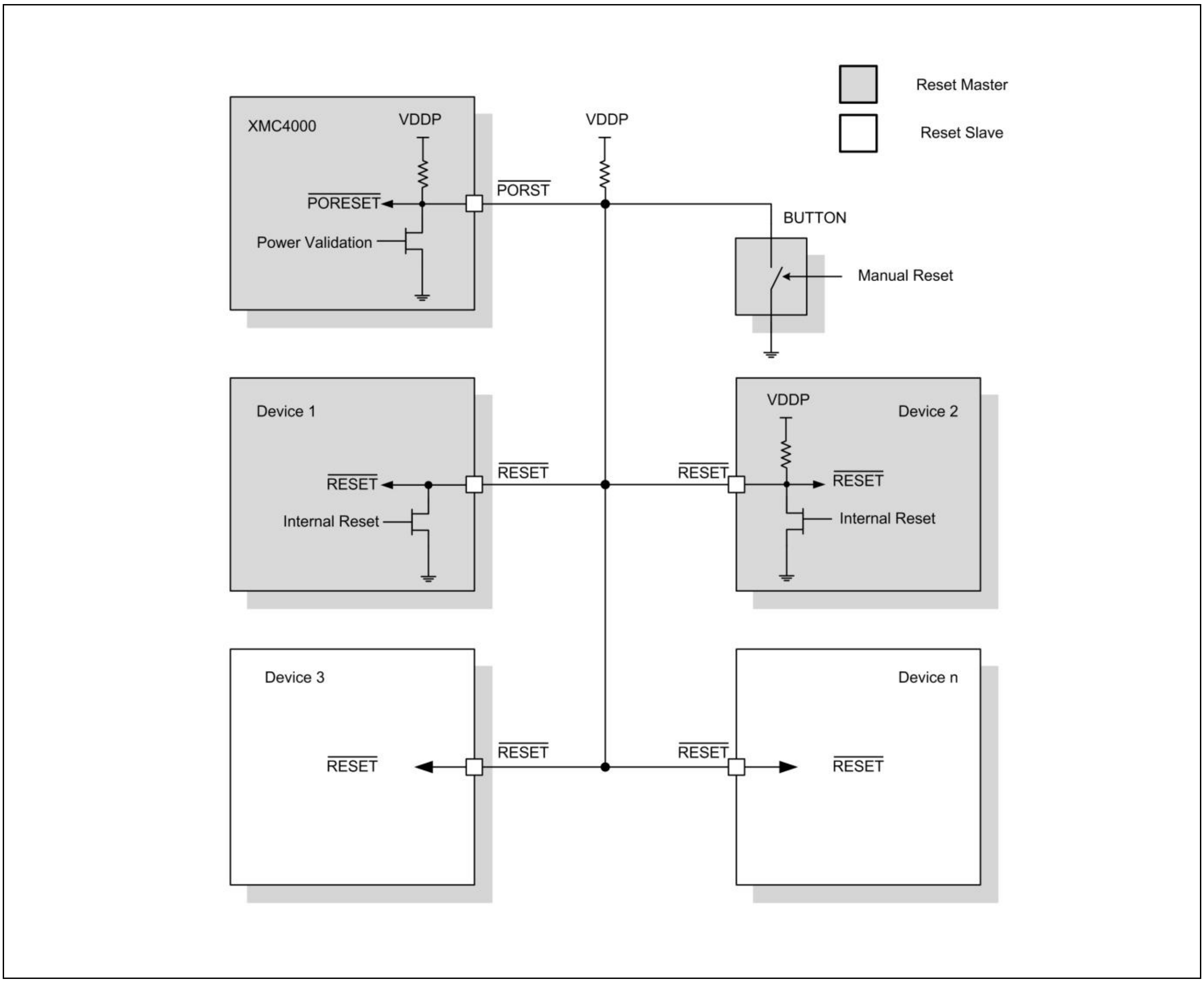

Figure 2 **Pull-up on PORST**

Failure condition	A load on the PORST reset pin prevents the bi-directional PORST from releasing reset faster that 2 µs during V_{DDP} ramping (time longer than 500 µs) leading to a risk of multiple low spikes on the PORST
Fail-safe effect	Improve PORST release time (below 2 µs) by applying a sufficient pull-up to the PORST pin

1.4.1 Power Validation

A power validation circuit monitors the internal core supply V_{DDC} voltage of the core domain.

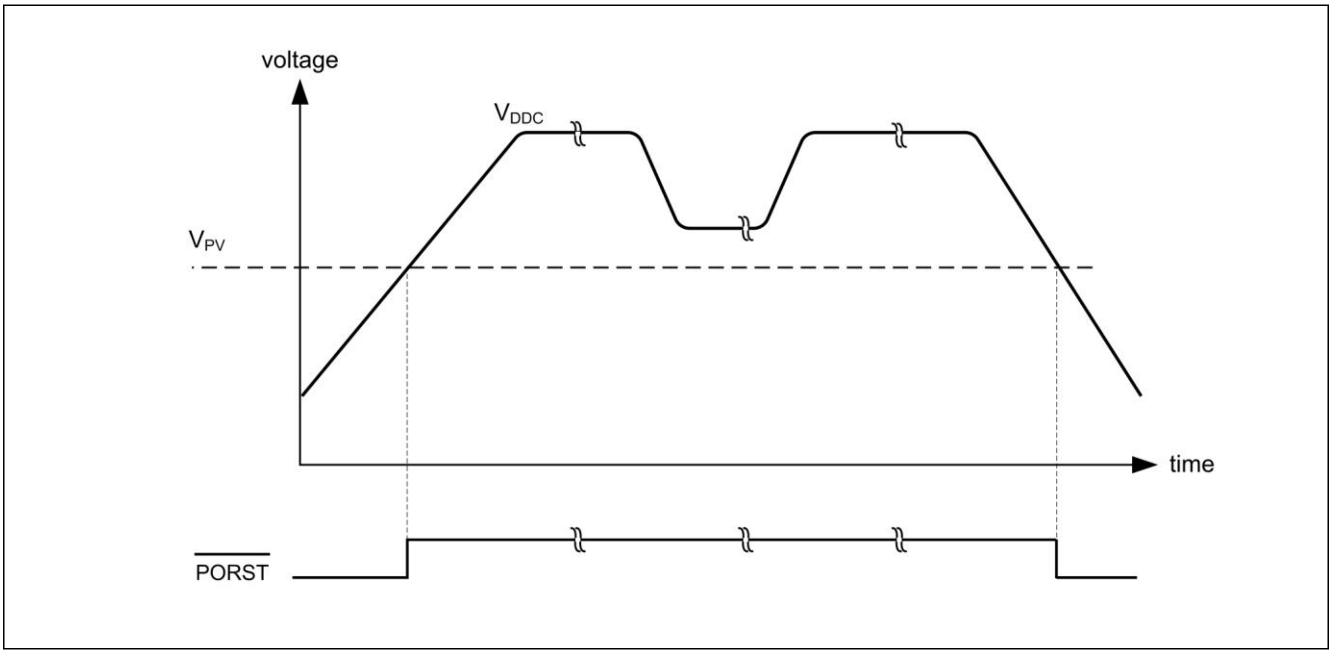

Figure 3 **Power Validation**

It monitors that the core voltage is above the voltage threshold V_{PV} which guarantees a save operation. Whenever the voltage falls below the threshold level a power-on reset is generated.

Failure condition	V_{DDC} voltage remains, or has dropped down below a minimum level V_{PV}, which may lead to severe core logic malfunction, that may result in a system hang-up condition.
Fail-safe effect	PORST reset is asserted and remains active as long as the core voltage V_{DDC} has not recovered to a level that guarantees proper functionality.

1.4.2 Supply Watchdog

The supply watchdog compares the supply voltage against the reset threshold V_{POR} (see Figure 4). The Datasheet defines the nominal value and applied hysteresis.

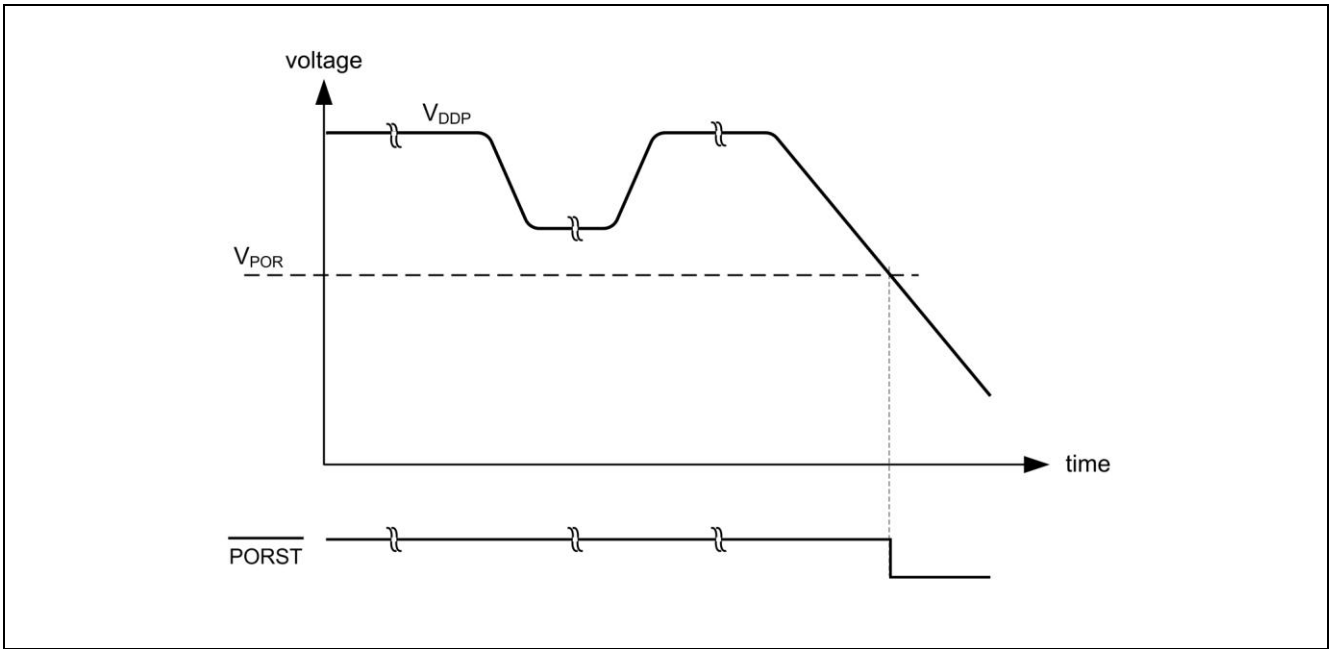

Figure 4 Supply Voltage Monitoring

While the supply voltage is below V_{POR} the device is held in reset.

As soon as the voltage falls below V_{POR} a power on reset is triggered.

Failure condition	During normal operation V_{DDP} voltage drops to a level that is too low to ensure the proper function of modules supplied with V_{DDP}, which may potentially lead to Flash malfunction, I/O pads misbehavior, or unreliable V_{DDC} generation in EVR for example.
Fail-safe effect	PORST reset is activated to prevent any activity when V_{DDP} is too low.

Note: *Recovery from this supply state results in Power-on Reset release.*

1.5 Supply Voltage Brown-out Detection

Brown-out detection is an additional voltage monitoring feature that enables the user software to perform some corrective action that brings the chip into safe operation in case of a critical supply voltage drop, and avoids a System Reset generated by the Supply Voltage Monitoring (see Figure 5).

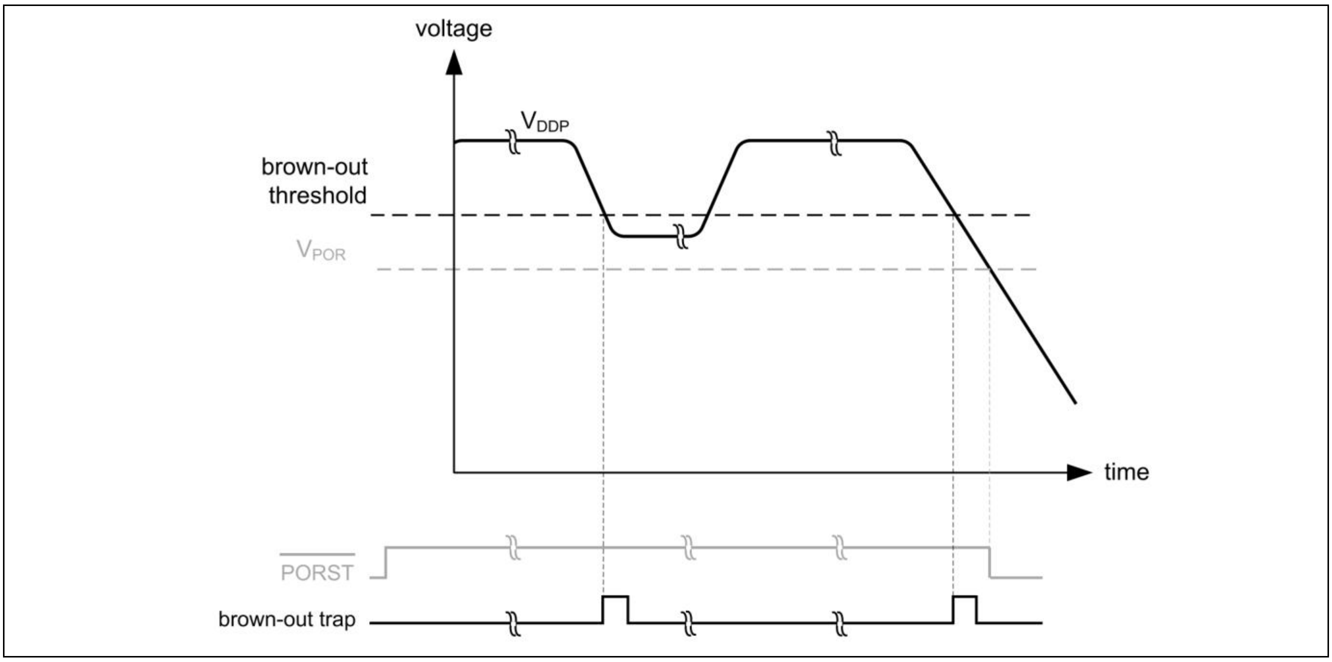

Figure 5 Power Validation

A drop of supply voltage to a critical threshold level, programmed by the user, can be signaled to the CPU with a Trap. An emergency corrective action may involve for example, the reduction of current consumption by switching off some modules, or some interaction with external devices that should result in recovery of the supply voltage level. The threshold value and the inspection interval are configured in the **PWRMON** register.

Automatic monitoring of the voltage against programmed voltage allows efficient operation without a need for software interaction if the supply voltage remains at a safe level. The supply voltage can also be monitored directly by software in the register **EVRVADCSTAT**.

Failure condition	Supply voltage V_{DDP} has dropped down to a level that may indicate a severe overload of the external voltage regulator caused by excessive activity of the XMC4000 device resulting in high current consumption. A further drop of the V_{DDP} voltage may occur unless a corrective action is taken in order to bring the device into safe operation.
Fail-safe effect	A system Trap is generated on crossing the brown-out threshold and is propagated to the CPU as an NMI interrupt. User software shall immediately apply a corrective action that results in a reduction of current consumption, such as clock rate reduction, inactivation of some modules, reduction of CPU activity, and so on. In effect PORST reset assertion may be avoided and normal device activity may be resumed without context loss with minimum effect on the overall system.

1.6 Hibernate Domain Power Management

Hibernate domain must be supplied with a valid V_{BAT} level during startup and while in active mode. This can be ensured by a direct V_{DDP} connection to V_{BAT}.

Invalid V_{BAT} voltage level may lead to unpredictable behavior of the device.

1.6.1 Temporary Loss of V_{DDP} Supply

Hibernate domain reset is asserted whenever V_{BAT} drops below a certain minimum level. Hibernate domain reset release needs to be performed with software access.

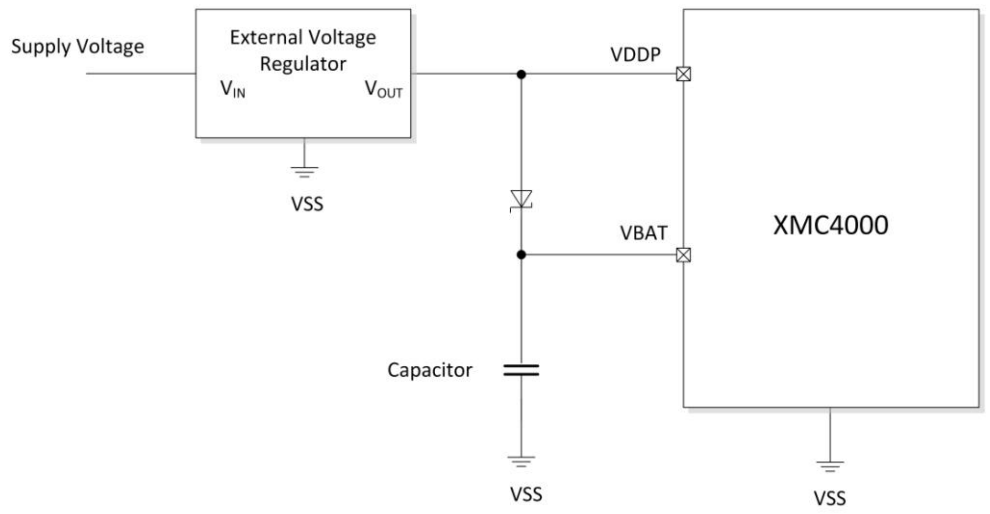

Figure 6 Hibernate domain voltage supply with V_{DDP}

If the Real-Time Counter (RTC) is activated and counting time is required in the application, it may be necessary to reduce the potential risk of spikes or temporary loss of power that might result from a Hibernate domain reset. This can be avoided with a capacitor connected between V_{BAT} and V_{SS} (see Figure 6). Typically 100 µF would allow compensating for several seconds of V_{DDP} loss without a risk of losing RTC state. The schottky diode between V_{DDP} and V_{BAT} prevents discharge of the capacitor towards V_{DDP} when V_{DDP} is lower than V_{BAT}.

Failure condition	Unexpected short drop of V_{DDP} below minimum V_{BAT} level that may lead to loss of RTC state.
Fail-safe effect	Preservation of RTC state upon V_{DDP} supply voltage shortage by supplying V_{BAT} voltage from the capacitor.

1.6.2 Prevention of Premature Coin Battery Discharging

Typically, the V_{BAT} supply voltage of the Hibernate domain (e.g. a coin battery) is lower than V_{DDP} (see Figure 7).

Note: *It is strongly recommended to supply Hibernate domain with V_{DDP} when available in order to extend the battery life time.*

An external supply voltage switching solution based on schottky diodes is shown in Figure 7. The scenario may apply to applications where RTC is activated and will also preserve its operation during longer periods of device inactivity, while in hibernate mode for example, or when simply powered-off for a period of time.

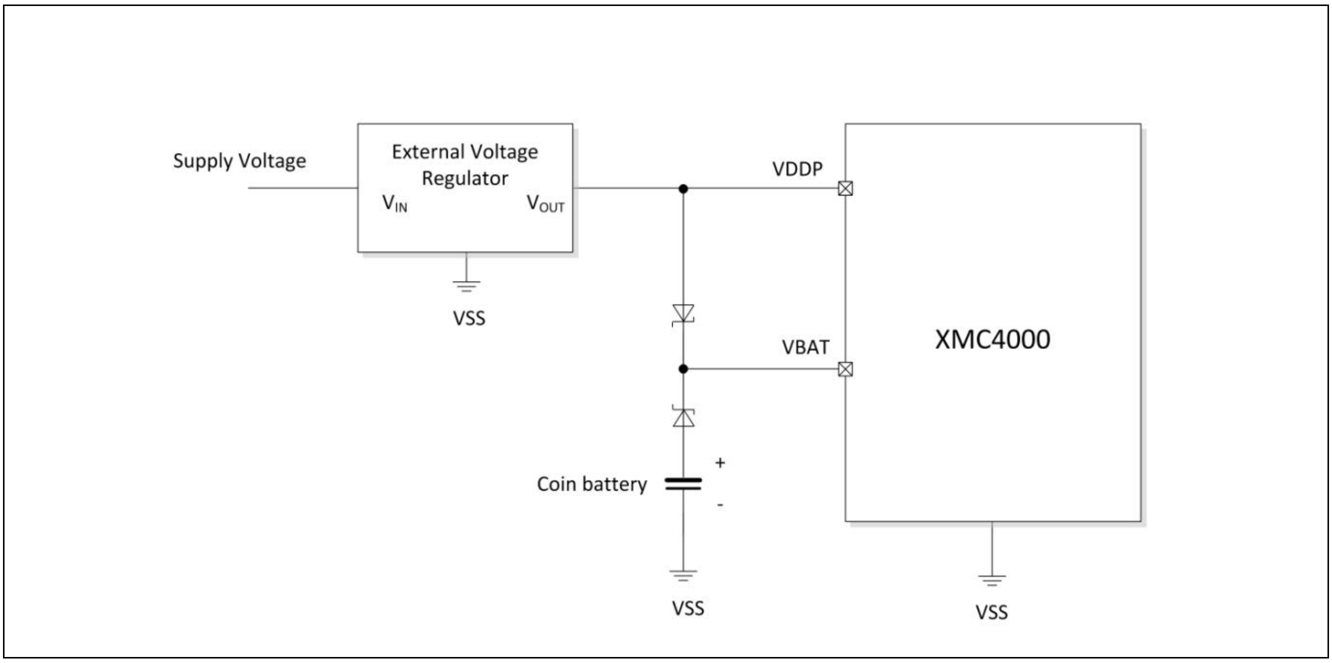

Figure 7 Hibernate domain voltage supply with a coin battery

Failure condition	Direct battery connected to V_{BAT} voltage supplied to Hibernate domain during active operation will result in draining the battery permanently. This would lead to dramatically shortened battery life, and a drop of the V_{BAT} voltage to a level that may result in severe system failure (valid V_{BAT} must always be supplied when in active mode).
Fail-safe effect	Applying a schottky diode between V_{DDP} and V_{BAT} prevents the battery draining while in active operation mode (i.e. when V_{DDP} is supplied), and prevents discharge of the battery towards the V_{DDP} pin when the main supply voltage is not available. In case of end-of-life of the coin battery, the Hibernate domain can still be properly supplied while in active mode, even if the hibernate mode and RTC module are reset during power-off. The schottky diode between a coin battery and V_{BAT} input prevents reverse current through the battery which might cause battery damage.

Clock Supervision

2 Clock Supervision

2.1 Introduction

The XMC4000 clocking system implements various clock sources and operating modes. Fail-safe features cover a wide range of aspects of operation that can be utilized in order to ensure stable operation.

The following sections cover the basic introduction to the clocking system elements and associated fail-safe mechanisms:

- Fail-safe System Clock
  - Backup Clock Source
  - High Precision Oscillator Watchdog Trap
  - System PLL Loss-of-Lock Trap
  - System PLL Loss-of-Lock RecoverySystem PLL Loss-of-Lock Recovery
  - Emergency Mode
- Fail-safe USB Clock
  - USB PLL Loss-of-Lock Trap
- Fail-safe RTC ClockFail-safe RTC Clock
  - RTC Clock Watchdog Trap
  - Emergency Mode Clock for RTC

2.2 Fail-safe System Clock

The central element of the system clock fail-safe mechanism is clock monitoring. The monitoring is performed using a High Precision Oscillator Watchdog implemented as a part of the PLL module.

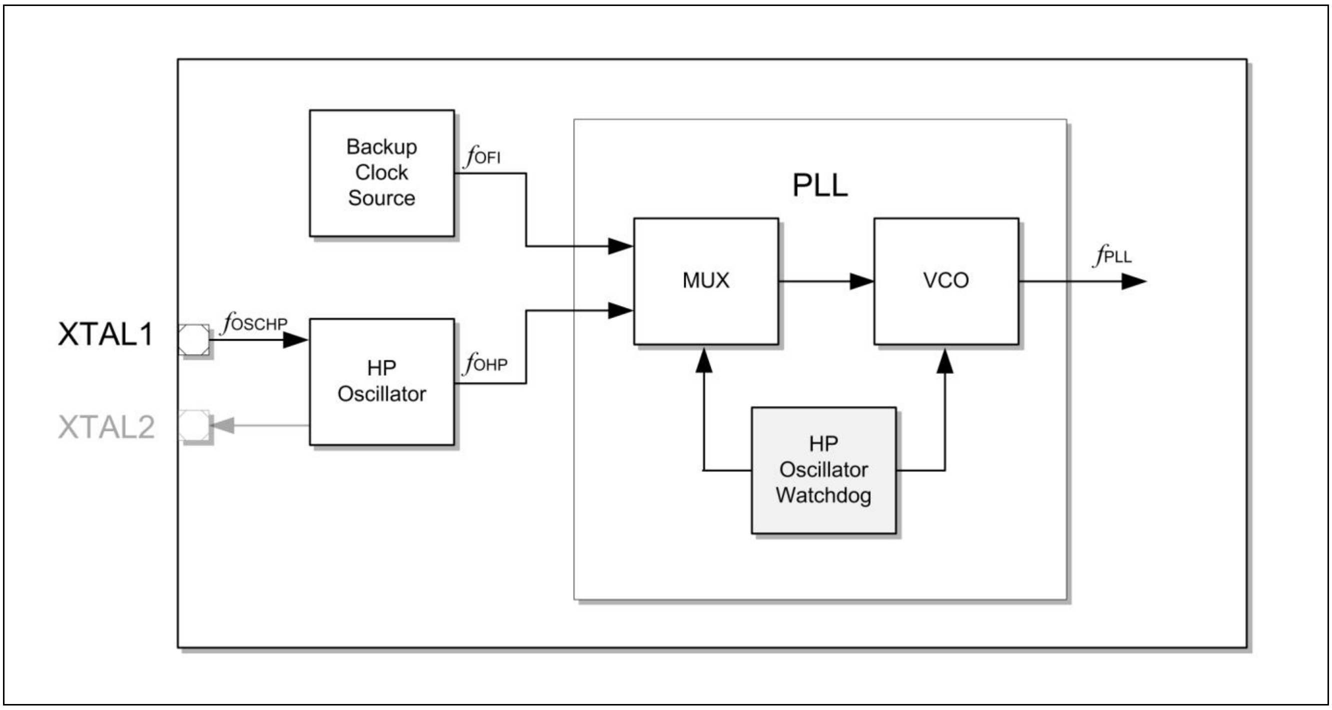

Figure 8 System Clock Supervisor

The watchdog circuit is capable of detecting the malfunction of the external crystal oscillator and to trigger a service request and/or a corrective reaction in hardware.

The following sections describe various aspects of the fail-safe mechanism.

2.3 Backup Clock Source

The backup clock f_{OFI} generated internally, is the default clock after start-up. It is used for by-passing the PLL for startup of the system without an external clock. Furthermore it can be used as an independent clock source for the watchdog module, or even as the system clock source during normal operation. While in prescaler mode this clock is automatically used as the emergency clock if external clock failure is detected.

Clock adjustment is required to reach the desired level of f_{OFI} precision. The backup clock source provides two adjustment procedures:

- loading of adjustment value during start-up
- continuous adjustment using the high-precision f_{STDBY} clock as reference

Failure condition	External clock input or crystal resonator/oscillator not present, not configured, or damaged.
Fail-safe effect	The internal Backup Clock Source is the most reliable clock source and remains active for all conditions while the chip is in normal operation mode. This clock cannot be disabled accidently by mistake with software, as long as the power supply is valid.

2.3.1 High Precision Oscillator Watchdog Trap

The oscillator watchdog monitors the incoming clock frequency f_{OSC}. A stable and defined input frequency is a mandatory requirement for operation in both Prescaler Mode and Normal Mode. It is required that the input frequency f_{OSC} is in a certain frequency range to obtain a stable master clock.

The expected input frequency range is selected via the bit field **OSCHPCTRL.OSCVAL**. The oscillator watchdog checks for spikes, too low frequencies, and for too high frequencies. The clock that is monitored is $f_{OSCREF,}$ which is derived from f_{OSC}.

The monitored frequency is too low if it is below 1.25 MHz and too high if it is above 7.5 MHz. This leads to the following two conditions:

- Too low: f_{OSC} < 1.25 MHz × (**OSCHPCTRL.OSCVAL**+1)
- Too high: f_{OSC} > 7.5MHz × (**OSCHPCTRL.OSCVAL**+1)

The divider value **OSCHPCTRL.OSCVAL** shall to be selected in a way that f_{OSCREF} is approximately 2.5 MHz.

Note: f_{OSCREF} shall be within the range of 2 MHz to 3 MHz and should be as close as possible to 2.5 MHz.

Failure condition	External clock outside of expected frequency range and/or spikes present (e.g. failing crystal).
Fail-safe effect	A Trap generated and flagged in the TRAPSTAT register. An appropriate corrective action can be taken in user software.

2.3.2 System PLL Loss-of-Lock Trap

The System PLL Loss-of-Lock is signaled as a system Trap in case VCO lock has been lost and it continues in a free running mode. The user software shall apply a corrective action in order to re-lock the PLL and bring the system into safe operation.

Failure condition	PLL configuration and/or input clock rate causes unstable operation of the VCO, leading to a loss-of-lock condition.
Fail-safe effect	A system Trap is generated and handled as an NMI request. System can be brought into safe operation with user software.

2.3.3 System PLL Loss-of-Lock Recovery

The System PLL Loss-of-Lock is signaled as a system Trap in case VCO lock has been lost. However, the PLL will try to re-lock without software interaction if the **SCU_PLLCON0.VCODISDIS** bit is set. The user software may still apply some corrective action but does not need to re-lock PLL.

Failure condition	System PLL configuration and/or input clock rate causes unstable operation of the VCO leading to a loss-of-lock condition.
Fail-safe effect	The PLL will re-lock automatically, without software interaction. A system Trap is generated and handled as an NMI request. System can be brought into safe operation with user software.

2.3.4 Emergency Mode

An Emergency in the clocking system occurs when the clock source stops working correctly; i.e. the frequency is outside of the tolerance range, or because of the presence of spikes.

A typical element prone to cause Emergency mode is an external crystal on the application board. The high precision oscillator circuit can drive an external crystal or accept an external clock source.

External Crystal Mode

For the external crystal mode, external oscillator load circuitry is required. The circuitry must be connected to both pins, XTAL1 and XTAL2 (see Figure 9). It consists normally of the two load capacitances C1 and C2. For some crystals a series damping resistor might be necessary. The exact values and related operating range depend on the crystal and have to be determined and optimized together with the crystal vendor using the negative resistance method. The crystal frequency must be in the range from 4 to 25 MHz.

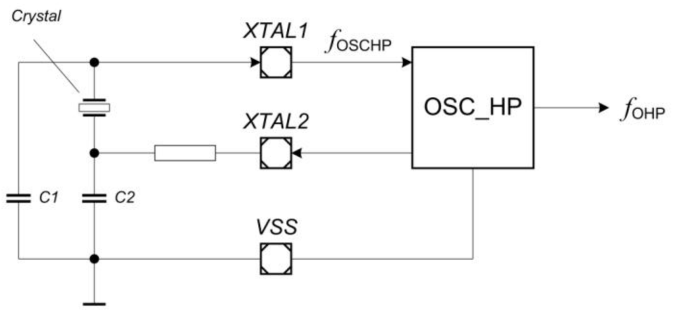

Figure 9 External Crystal Mode Circuitry for the High-Precision Oscillator

2.3.4.1 Emergency System Clock for PLL Normal Mode

The main PLL converts a low-frequency external clock signal to a high-speed internal clock. The PLL also has fail-safe logic that detects de-generative external clock behavior such as abnormal frequency deviations or a total loss of the external clock. The PLL triggers autonomously emergency action if it loses its lock on the external clock and switches to a free-running VCO of the PLL (see Figure 8).

PLL Normal Mode

In PLL Normal Mode the Voltage Controlled Oscillator (VCO) receives input clock from f_{OSC} and multiplies its frequency. The output clock from the VCO is phase-locked to the input clock.

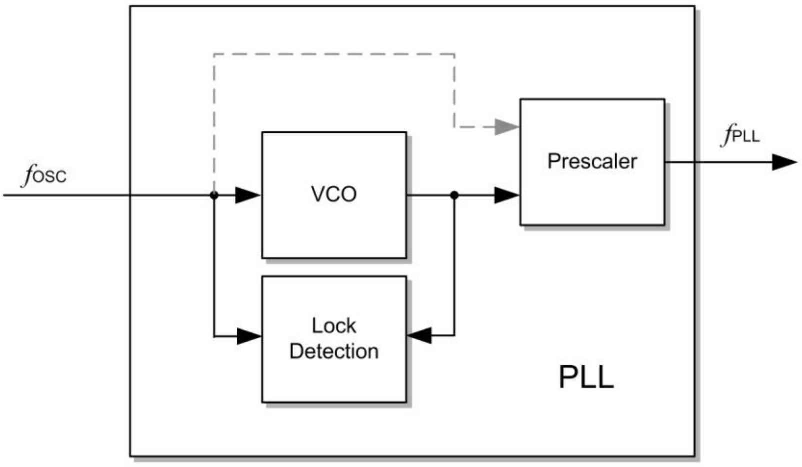

Figure 10 PLL Normal Mode

The high frequency output clock is then scaled down accordingly with the Prescaler (see Figure 10). The phase–locked output is supervised with the Lock Detection circuit.

Failure condition	The external clock is outside of the expected frequency range and/or spikes are present (e.g. caused by a failing crystal) while PLL is running in Normal Mode; i.e. VCO output is configured to be phase locked to the input clock.
Fail-safe effect	The PLL will unlock the VCO and provide a free running VCO output as the system clock f_{SYS}. A Trap is also generated and flagged in the TRAPSTAT register. An appropriate corrective action can be taken in user software.

2.3.4.2 Emergency System Clock for PLL Prescaler Mode

The PLL offers a VCO Power-Down mode. This mode can be entered to save power within the PLL. The VCO Power-Down mode is entered by setting bit PLLCON0.VCOPWD. While the PLL is in this mode only the Prescaler mode is operable.

Note: *Selecting the VCO Power-Down mode does not automatically switch to the Prescaler mode. Before the VCO Power-Down mode is entered, the Prescaler mode must be active.*

PLL Prescaler mode

In Prescaler mode the VCO is by-passed (and, optionally powered-down). The output clock is a direct scale down of the input clock (see Figure 11).

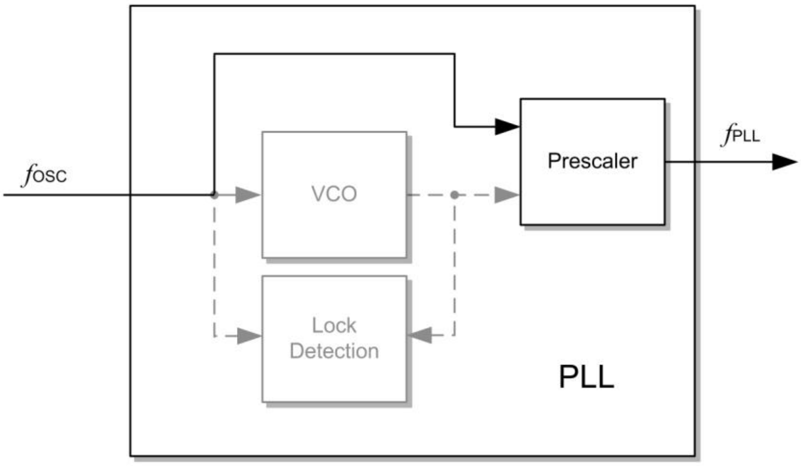

Figure 11 PLL Prescaler mode

Failure condition	The external clock is outside of the expected frequency range and/or spikes are present (e.g. caused by a failing crystal) while PLL running in Prescaler mode; i.e. VCO by-passed and/or in power-down.
Fail-safe effect	The system clock f_{SYS} will automatically switch to the Back-up Clock Source output f_{OFI}. A Trap is also generated and flagged in the **TRAPSTAT** register. An appropriate corrective action can be taken in user software.

2.3.5 Fail-safe Clock Ratio Configuration

XMC4000 devices support a set of different clock ratio configurations for different groups of on-chip resources. A simplified rule that generally applies here is that the ratio between any of the clocks (any combination) f_{CPU}, f_{CCU} and f_{PRIPH} is never greater than 2 (for more details please refer to "Clock System Architecture" chapter of the Reference Manual). Not conforming to the rule may lead to unpredictable system behavior, such as missed service requests for example.

A fail-safe mechanism implemented on XMC4400 and XMC4200 devices will automatically prevent invalid configuration of the clock. The mechanism assumes that all clock configuration properties are accessible in a single register **MLINKCLKCR**. An attempt to write an invalid clock configuration will be rejected and will result in a bus error exception.

Failure condition	An attempt to configure clock system properties such that a clock ratio between any combination of the clock signals f_{CPU}, f_{CCU} and f_{PRIPH} is greater than 2.
Fail-safe effect	A fail-safe mechanism implemented in MLINKCLKCR will reject the invalid configuration. A system bus error will be generated and cause a CPU exception.

2.4 Fail-safe USB Clock

The USB PLL operation is similar (reduced) to the system PLL. The USB PLL typically gets clocked with a clock generated with the High Precision Oscillator and en external high quality crystal (see Figure 12).

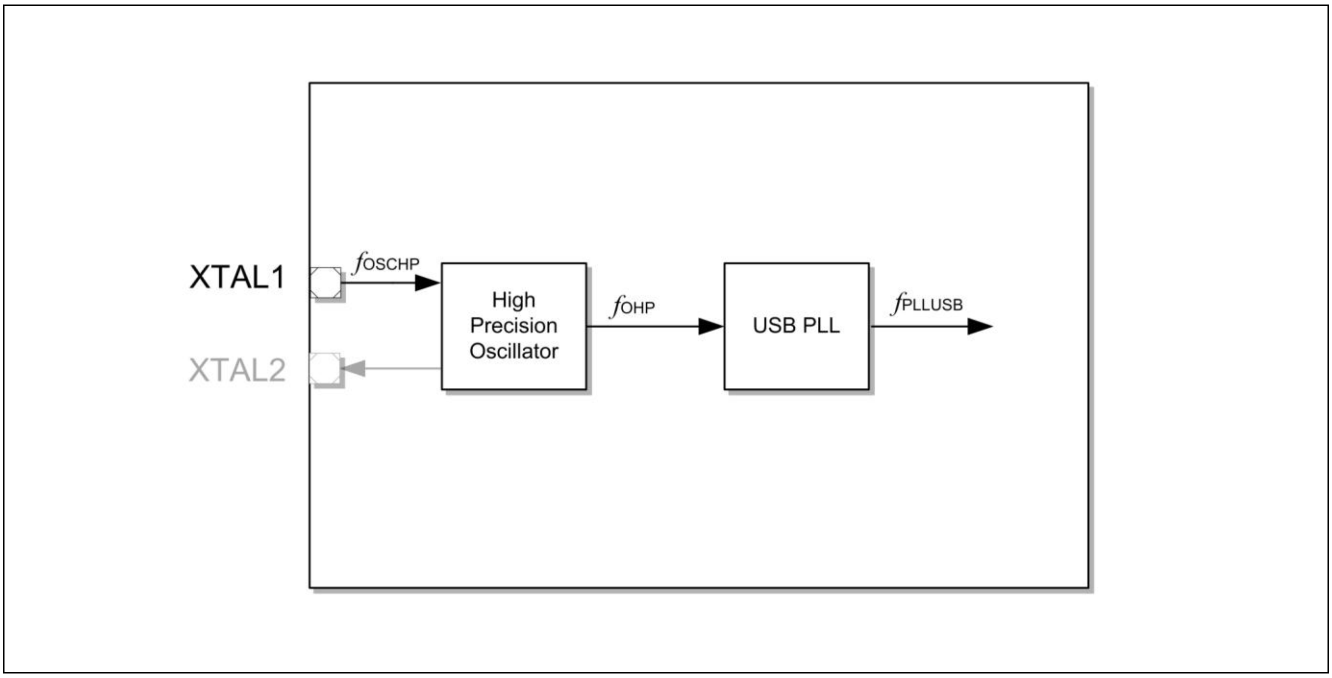

Figure 12 USB PLL

The USB PLL may also be clocked with a direct clock (very high precision required).

2.4.1 USB PLL Loss-of-Lock Trap

The USB PLL Loss-of-Lock is signaled as a system Trap when VCO lock has been lost, and it continues in a free running mode. The user software shall apply a corrective action in order to re-lock the USB PLL to bring the system into safe operation.

Failure condition	USB PLL configuration and/or input clock rate causes unstable operation of the VCO leading to a loss-of-lock condition.
Fail-safe effect	A system Trap is generated and handled as an NMI request. System can be brought into safe operation with user software.

2.5 Fail-safe RTC Clock

The RTC clock of 32.768 kHz can be generated in Hibernate Domain with an Ultra-Low Power Oscillator (OSC_ULP) and external crystal, fed directly via RTC_XTAL1 input, or generated internally in the Internal Slow Clock Oscillator.

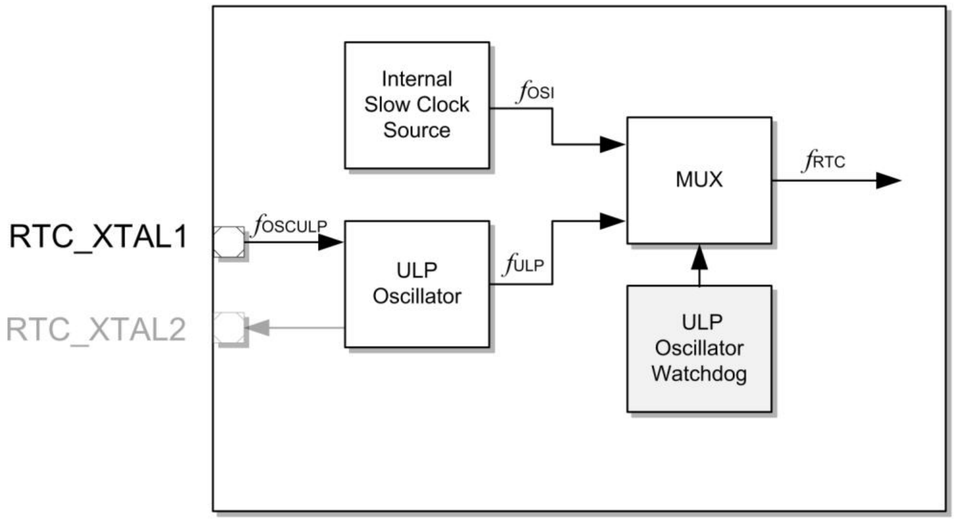

Figure 13 RTC Clock

The importance of the RTC Clock stability is determined by the fact that it may be used for precise, real-time keeping. The Standby Clock Watchdog prevents loss of clock in case of the external crystal failure. The Internal Slow Clock Source will remain active and f_{RTC} will automatically switch to this clock in the absence of the precise 32.768 kHz clock. A system Trap will also be generated to allow the user software to apply a corrective action, such as to request a maintenance service of the device. Time keeping with a reduced accuracy can still be maintained until the failure cause has been corrected.

The Internal Slow Clock Oscillator can only be switched off with user software if the Ultra-Low Power Oscillator has been working in a stable state. Otherwise, attempts to switch off the oscillator will be ignored.

2.5.1 RTC Clock Watchdog Trap

Failure condition	Externally generated clock outside of expected frequency range due for example to external crystal failure.
Fail-safe effect	A Trap generated and flagged in the TRAPSTAT register while operating in active mode. While in Hibernate mode the Trap may be configured to cause a wake-up of the system. An appropriate corrective action can be taken in user software.

2.5.2 Emergency Mode Clock for RTC

Failure condition	The external clock outside of expected frequency range (e.g. failing crystal).
Fail-safe effect	The RTC clock f_{SYS} will automatically switch to the Backup Clock Source output f_{OFI}. A Trap is also generated and flagged in the TRAPSTAT register. An appropriate corrective action can be taken in user software.

Memory Integrity

3 Memory Integrity Protection

3.1 Introduction

The following of memory fail-safe features are described in the subsequent sections:

- Principle of Parity Check Operation
- Parity Error on System SRAMs
- Parity Error on Peripheral Module SRAMs
- System Reset Upon Parity Error

3.2 Principle of Parity Check Operation

Memory Integrity Protection is performed using memory parity checking. The parity logic generates additional parity bits which are stored along with each data word at a write operation. A read operation implies checking the previous stored parity information (see Figure 15).

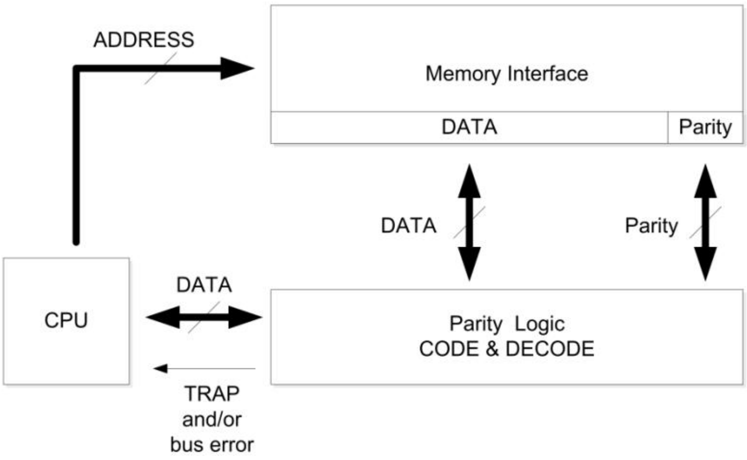

Figure 14 Parity Logic

An occurrence of a parity error can be signaled with a bus error and/or system Trap. A parity error is also observable in the Trap raw status register **SCU_TRAPRAW**. It is configurable whether a memory error should trigger an NMI or System Reset.

Parity check can be enabled individually for each instance of memory with the **SCU_PEEN** register. A trap generation can be individually enabled with the **SCU_PETE** register in order to have the Trap flag reflected in the **SCU_TRAPRAW** register, or to generate System Reset if enabled in the **SCU_PERSTEN** register.

The following aspects of fail-safe of memory integrity are covered in the subsequent sections:

3.3 Parity Error on System SRAMs

System SRAMs like PSRAM, DSRAM1 and DSRAM2, can be read/write accessed directly with user software. These memories are attached directly to the bus system and are mapped into system address space. Parity errors are signaled with a bus error.

Failure condition	Parity error detected upon a read from PSRAM, DSRAM1 or DSRAM2.
Fail-safe effect	Bus error exception occurs. A corrective action needs to be performed in the user exception routine.

3.4 Parity Error on Peripheral Module SRAMs

Peripheral SRAMs cannot be accessed directly and are not directly attached to the bus system. Access to these memories is performed indirectly with internal module logic, via dedicated registers. Parity errors are signaled with Trap signals.

Failure condition	Parity error detected on a read from an internal peripheral module memory. The parity error may be caused by a hardware failure, or by a read access to an initialized memory.
Fail-safe effect	System Trap signal generated. A corrective action needs to be performed in the user exception routine.

3.5 System Reset Upon Parity Error

A system reset can be optionally generated in case any of the parity check enabled modules detect a parity error. This requires enabling the feature with the **SCU_PERSTEN** register. This feature may be useful in case of code execution from system SRAMs for example, where the occurrence of a parity error may indicate a severe integrity problem of the memory content, that in turn might lead to system hang-up. In some application cases it may be safer to trigger immediate system reset, resulting in complete SRAM re-initialization.

Failure condition	Parity error detected upon a read from any parity enabled memory. The parity error may be caused by a hardware failure or by a read access to an initialized memory.
Fail-safe effect	System reset is generated resulting in re-boot and system initialization. The reset cause is recorded in the RSSTAT register.

Fail-safe Flash

4 Fail-Safe Flash

4.1 Introduction

The Flash Module implements various mechanisms to limit or prevent the danger from misbehaviour under critical conditions, and provides the means for supervising different aspects of operation. The error conditions may be caused by a hazardous environment potentially affecting data integrity for example, poor supply voltage, or software bugs.

The following aspects of fail-safe Flash behavior are covered in the subsequent sections:

- Error Correction Codes (ECC)
  - Single-bit Error
  - Double-bit Error
- Reset During Flash Operation
  - Boot Fallback Mode (ABM)
- Flash Clock
- Wear-leveling
- Flash Write and OTP Protection
- Service Request Generation
  - Interrupt Control
  - Trap Control
- Handling Errors During Operation
- Handling Flash Errors During Startup

4.2 Error Correction Codes (ECC)

ECCs are used to ensure the data is always consistent if used within the chip specification. The strength of the required ECC is defined by the worst-case bit failure rate.

The data in Flash is stored with ECC in order to protect against data corruption. The ECC is automatically generated when programming the PFLASH.

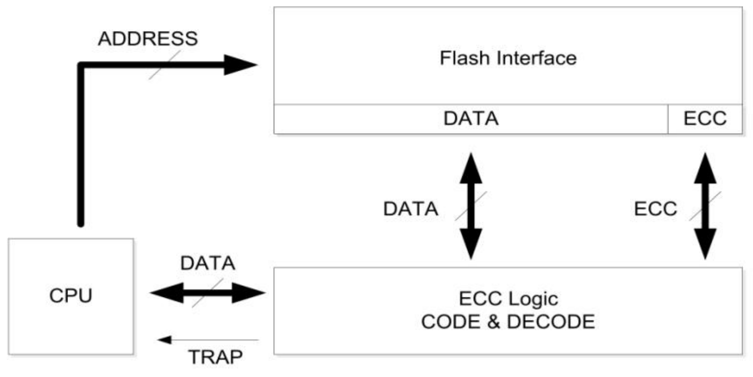

Figure 15 **Error Correction Code Logic**

When data is read these codes are evaluated. Data in PFLASH uses an ECC code with SEC-DED (Single Error Correction, Double Error Detection) capabilities. Each block of 64 data bits is accompanied with 8 ECC bits over the 64 data bits.

An erased data block (all bits ‘0’) has an ECC value of 00_H. Therefore an erased sector is free of ECC errors. A data block with all bits ‘1’ has an ECC value of FFH.

The ECC is automatically evaluated when reading data, as depicted in Figure 15.

4.2.1 Single-bit Error

The occurrence of an error in a single-bit of 64-bit data results in an automatic correction in hardware and may be considered transparent for the software.

Failure condition	Single-bit error detected in ECC circuit.
Fail-safe effect	Data and ECC value are corrected automatically in hardware without user software interaction. The occurrence of the error is noted in FSR.PFSBER and an Interrupt is triggered if enabled in FCON.PFSBERM.

4.2.2 Double-bit Error

The occurrence of an error in two bits of 64-bit data results in a bus error exception. A corrective action needs to be performed in user software.

Failure condition	Double-bit error detected in ECC circuit.
Fail-safe effect	A bus error is generated (if not disabled by MARP.TRAPDIS) and the occurrence of the error is noted in FSR.PFDBER.

4.3 Reset During Flash Operation

A reset or power failure during a Flash operation (i.e. program or erase) must be considered as a violation of stable operating conditions. However the Flash was designed to prevent damage to non-addressed Flash ranges when the reset is applied, as defined in the device datasheet. The exceptions are erasing logical sectors and UCBs.

Aborting the erase process of a logical sector can leave the complete physical sector unreadable. When a UCB erase is aborted the entire Flash can become unusable. So the UCBs must only be erased in a controlled environment. The addressed Flash range is left in an undefined state.

When an erase operation is aborted the addressed logical or physical sector can still contain data and it can be in a state that doesn’t allow this range to be programmed.

When a page programming operation is aborted the page can:

- can still appear as erased (but contain some programmed bits)
- can appear as being correctly programmed (but the data has a lowered retention)
- contains garbage data.

It is also possible that the read data is unstable, which means that depending on the operating conditions, different data is read.

For the detection of an aborted Flash process the flags **FSR.PROG** and **FSR.ERASE** could be used as indicators but only if the reset was a System Reset. Power-on resets cannot be determined from any flags.

It is not possible to detect an aborted operation simply by reading the Flash range. Even the margin reads do not offer a reliable indication.

When erasing or programming the **PFLASH**, an external instance can usually notice the reset and simply restart the operation by erasing the Flash range and programming it again.

However, in cases where this external instance does not exist, a common solution is to detect an abort by performing two operations in sequence and determine after reset from the correctness of the second, the completeness of the first operation.

For example, after erasing a sector a page is programmed. After reset the existence of this page proves that the erase process was performed completely.

Detection of aborted programming processes can be handled similarly. After programming a block of data an additional page is programmed as a marker. If after reset the block of data is readable and the marker exists, this ensures that the block of data was programmed without interruption.

If a complete page can be used as a marker, the following steps show how to reduce the marker size to 8 bytes. Note that these steps violate the rule that a page may be programmed only once, but this violation is only allowed for this purpose and only when the algorithm is robust against disturbed pages (see also recommendations for handling single-bit errors) by repeating a programming step when it detects a failure.

Robust programming of a page of data with an 8 byte marker

1. After reset program preferably always first to an even page (“Target Page”).
2. If the Other Page on the same Word line contains active data, save it to SRAM (the page can become disturbed because of the 4 programming operations per Word line).
3. Program the data to the Target Page.
4. Perform a strict check of the Target Page (see below).
5. Program an 8-byte marker to the Target Page.
6. Perform a strict check of the Target Page.
   - In case of any error of the strict check go to the next word-line and program the saved data and the target data again following the same steps.
7. Ensure that the algorithm doesn’t repeat unlimited in case of a violation of operating conditions.

Strict checking of programmed data

1. Ignore single-bit errors and the **VER** flag.
2. Switch to tight margin 0.
3. If the data (check the complete page) is not equal to the expected data report an error.
4. If a double-bit error is detected report an error.

After reset the algorithm has to check the last programmed page to see if it was programmed completely:

1. Read with normal read level. Ignore single-bit errors.
2. Read 8-byte marker and check for double-bit errors.
3. Read data part and verify its consistency (e.g. by evaluating a CRC). Check for double-bit errors.
4. If the data part is defective do not use it (e.g. by invalidating the page).
5. If the data part is ok:
   - If the marker is erased the data part could have been programmed incorrectly. Either the data part should not be used, or alternatively it could be programmed again to a following page.
   - If the marker contains incorrect data the data part was most likely programmed correctly but the marker was programmed incompletely. The page could be used as is, or alternatively the data could be programmed again to a following page.
   - If the marker is ok the data part was programmed successfully. However this does not guarantee the marker part itself. Therefore the algorithm must be robust enough for the situation in which the marker later becomes unreadable.

4.3.1 Boot Fallback Mode (ABM)

When this boot mode is selected, ABM Address-0 header is audited first.

- A positive audit results in SSW ceding control to the user application pointed to by the header.
- A negative audit results in the evaluation of ABM Address-1.

Should the audit of ABMA address1 header fail, SSW launches diagnostics monitor mode.

SSW evaluates the two ABM headers in succession. Control is given to the application referenced by the first valid ABM header found by SSW. Should the headers be found to be unusable, SSW aborts further execution and places the CPU into a safe mode. A PORST is required to exit the safe mode.

Failure condition	Update of user code in Flash interrupted by a reset (or, power down).
Fail-safe effect	Alternative user software executed if the primary code is corrupt.

4.4 Flash Clock

The Flash interface is operating at the same clock speed as the CPU, f_{CPU}. For proper operation of command sequences and when entering or waking up from Deep-Sleep mode, f_{CPU} shall be equal or above 1 MHz. A safe f_{CPU} clock frequency and Flash start-up is guaranteed after reset release. The f_{CPU} can be changed by the user software and special attention needs to be paid on this setting before entering Deep-Sleep. An incorrect clock rate configuration in the **SCU_DSLEEPCR** register of the SCU module may result in a hang-up of the system on wake-up.

4.5 Wear-leveling

Flash cells have a limited life (typically 20 years of memory retention for XMC4000) and can only be erased/programmed a certain number of times (up to 1000 times on XMC4000) before becoming unreliable. In effect they ‘wear out’. Wear-leveling algorithms caN be applied to maximize the life of the chip by moving the data between physical blocks to ensure some cells are not over-used in comparison to others.

The Flash memory offers a “margin check feature”: the limit which defines if a Flash cell is read as logic ‘0’ or logic ‘1’ can be shifted. This is controlled by the register **MARP**. The Margin Control Register **MARP** is used to change the margin levels for read operations to find problematic array bits. The array area to be checked is read with more restrictive margins. “Problematic” bits will result in a single or double-bit error that is reported to the CPU by an error interrupt or a bus error trap. The double-bit error trap can be disabled for margin checks and also re-directed to an error interrupt.

After changing the read margin, it is necessary to wait for at least $t_{FL\_MarginDel}$ before reading the affected Flash module. During erase or program operation only the standard (default) margins are allowed.

Flash failures lead to data retention problems, which can be caused by:

- Invalid programming / erase algorithms (e.g. due to programmer fail at customer backend)
- Too high endurance numbers (e.g. due to software fail)
- Aborted erase / programming (e.g. due to power fail)

Failure condition	Data errors indication data retention occurrence signaled with excessive ECC error rate.
Fail-safe effect	Software routine utilizing algorithm enabled with the wear-leveling mechanism deployed in order to detect failing cells and diagnose their condition. Corrective action may include re-programming the cells with the original data for example. In case of severe issues the cells may be by-passed with redundancy cells.

4.6 Flash Write and OTP Protection

The Flash write protection mechanism prevents an un-intended overwrite of the Flash by the user application software. A write sequence applied to Flash protected sectors will be ignored and no data will be altered.

The Flash memory can be read and write protected. The protection is configured by programming the User Configuration Blocks (UCB).

4.6.1 Configuring Flash Protection in the UCB

The effective protection is determined by the content of the Protection Configuration Indication **PROCON0–2** registers. These are loaded during startup from the **UCB0–2**. Each UCB comprises 1 Kbyte of Flash organized in 4 UC pages of 256 bytes. The UCBs have the following structure:

Table 1 UCB Content

UC Page	Bytes	UCB0	UCB1	UCB2
0	[1:0]	PROCON0	PROCON1	PROCON2
	[7:2]	unused	unused	unused
	[9:8]	PROCON0 (copy)	PROCON1 (copy)	PROCON2 (copy)
	[15:10]	unused	unused	unused
	[19:16]	PW0 of User 0	PW0 of User 1	unused
	[23:20]	PW1 of User 0	PW1 of User 1	unused
	[27:24]	PW0 of User 0 (copy)	PW0 of User 1 (copy)	unused
	[31:28]	PW1 of User 0 (copy)	PW1 of User 1 (copy)	unused
	others	unused	unused	unused
1		unused	unused	BMI and configuration data (details in Startup Mode chapter)
2	[3:0]	confirmation code	confirmation code	confirmation code
	[11:8]	confirmation code (copy)	confirmation code (copy)	confirmation code (copy)
	others	unused	unused	unused
4	unused	unused	unused	unused

A Flash range can be write-protected in the following ways:

- The complete **PFLASH** can be write-protected by read protection.
- Groups of sectors of **PFLASH** can be write-protected by three different “users” (i.e. UCBs):
  - **UCB0**: Write protection that can be disabled with the password of **UCB0**.
  - **UCB1**: Write protection that can be disabled with the password of **UCB1**.
  - **UCB2**: Write protection that cannot be disabled anymore (ROM or OTP function: “One-Time Programmable”).

4.6.2 Write and OTP Protection Status

Active write-protection is indicated with the **WPROIN** bits in the **FSR** register, and causes the program and erase command sequences to fail with a **PROER**.

A range "x" (i.e. a group of sectors; see **PROCON0**) of the **PFLASH** is write-protected if any of the following conditions are true:

- **FCON.RPA**
- **PROCON2.SxROM**
- **PROCON0.SxL** and not (**FSR.WPRODIS0**)
- **PROCON1.SxL** and not (**PROCON0.SxL**) and not (**FSR.WPRODIS1**)

The password **UCB0** disables the write protection of sectors protected from user 0 and user 1.

The password **UCB1** only disables the write protection of sectors protected from user 1.

The write protection of user 2 (OTP) can not be disabled.

The global write-protection applied when read-protection is active, can be disabled by using the password **UCB0** (i.e. read-protection is disabled).

4.7 Service Request Generation

Access and/or operational errors (e.g. wrong command sequences) may be reported to the user by interrupts, and they are indicated by flags in the Flash Status Register **FSR**. Additionally, bus errors may be generated resulting in CPU Traps.

4.7.1 Interrupt Control

The PMU and Flash modules support the supply of immediate error and status information to the user through interrupt generation. One CPU interrupt request is provided by the Flash module.

The Flash interrupt can be issued because of the following events:

- End of busy state: program or erase operation finished
- Operational error (OPER): program or erase operation aborted
- Verify error (VER): program or erase operation not correctly finished
- Protection error
- Sequence error
- Single-bit error: corrected read data from PFLASH delivered
- Double-bit error in Program Flash.

Note: *In the event of an OPER or VER error, the error interrupt will not be issued until the Flash 'busy' state is de-activated.*

The interrupt source is indicated in the Flash Status Register **FSR** by the error flags, or by the **PROG** or **ERASE** flag for an 'end of busy' interrupt. An interrupt is also generated for a new error event, even if the related error flag is still set from a previous error interrupt.

Every interrupt source is masked (disabled) after reset and can be enabled via dedicated mask bits in the Flash Configuration Register **FCON**.

4.7.2 Trap Control

CPU Traps are triggered because of bus errors, and are generated by the PMU for erroneous Flash accesses. Bus errors are generated synchronously to the bus cycle requesting the erroneous Flash access or the disturbed Flash read data.

Bus errors are issued for any of the following events:

- Non-correctable double-bit error of 64-bit data read from PFLASH (if not disabled for margin check)
- Invalid write-access to a read-only register
- Invalid write-access to Privileged Mode protected register
- Invalid data or instruction read-access if read-protection is active
- Access to un-implemented addresses within the register or array space
- Read-modify-write access to the Flash array

Write accesses to the Flash array address space are interpreted as command cycles and do not initiate a bus error but a sequence error if the address or data pattern is not correct. However, command sequence cycles, which address a busy Flash bank, are serviced with busy cycles, not with a sequence error. If the Trap event is a double-bit error in **PFLASH**, it is indicated in the **FSR**. With the exception of this error Trap event, no other Trap sources can be disabled within the PMU.

4.8 Handling Errors During Operation

The previous sections described the functionality of bits indicating errors in the Flash Status Register **FSR**. In this section we elaborate on this with a more in-depth explanation of the error conditions and provide recommendations as to how these should be handled by customer software. We look first at handling error conditions that occur during operation (i.e. after issuing command sequences), and then at error conditions detected during startup.

4.8.1 SQER "Sequence Error"

Fault conditions	• Invalid command cycle address or data; i.e. incorrect command sequence. • New "Enter Page" in Page Mode. • "Load Page" and not in Page Mode. • "Load Page" results in buffer overflow. • First "Load Page" addresses 2. Word. • "Write Page" with buffer underflow. • "Write Page" and not in Page Mode. • "Write Page" to wrong Flash type. • Byte transfer to password or data. • "Clear Status" or "Reset to Read" while busy. • Erase UCB with wrong UCBA.
New state	Read mode is entered with the following exceptions: • "Enter Page" in Page Mode re-enters Page Mode. • "Write Page" with buffer underflow is executed. • After "Load Page" causing a buffer overflow, the Page Mode is not left, a following "Write Page" is executed.
Proposed handling by software	Usually this bit is only set because of a bug in the software. Therefore in the development code the responsible error tracer should be notified. In production code this error should not occur. However it is possible to clear this flag with "Clear Status" or "Reset to Read" and simply issue the corrected command sequence again. With a SQER after the "Write Page" sequence it is possible to verify the written data in the Flash. It is sufficient to clear the flag with the "Clear Status" command if the written data is correct. If the written data is wrong, the whole sector must be erased and re-programmed.

4.8.2 PFOPER "Operation Error"

Fault conditions	ECC double-bit error detected in the Flash modules internal SRAM during a program or erase operation in **PFLASH**. **Cause:** This can be a transient event due to alpha-particles or illegal operating conditions. Alternatively it is a permanent error due to a hardware defect.
New state	The Flash operation is aborted, the BUSY flag is cleared and read mode is entered.
Proposed handling by software	The flag should be cleared with "Clear Status". The last operation can be determined from the PROG and ERASE flags. For an erase operation the affected physical sector must be assumed to be in an invalid state. For a program operation only the affected page is assumed to be in an invalid state. Other physical sectors can still be read, but new program or erase commands must not be issued before the next reset. So a reset must be performed. The reset performs a new Flash ramp-up with initialization of the microcode SRAM. The application must determine from the context which operation failed and react accordingly. Usually erasing the addressed sector and re-programming its data content is the most appropriate action. If a "Program Page" command was affected and the sector cannot be erased, the Word-line could be invalidated if required by marking it with all-one data and the data could be programmed to another empty Word-line. If there is a defect in the microcode SRAM, this error will occur again in the next program or erase operation. Although this error indicates a failed operation it is possible to ignore it and rely on a data verification step to determine if the Flash memory has valid data. Before re-programming the Flash the flow must ensure that a new reset is applied.

Note: *Even when the flag is ignored it is still recommended to clear it, otherwise all following operations, including "sleep", could trigger an interrupt even when they are successful.*

4.8.3 PROER “Protection Error”

Fault conditions	• Password failure. • Erase/Write to protected sector. • Erase UCB and protection active. • Write UC-Page to protected UCB. *Attention:* ***A protection violation can occur even when a protection was not explicitly installed by the user. This is the case when the Flash startup detects an error and starts the user software with read-only Flash. Trying to change the Flash memory will then cause a PROER.***
New state	Read mode is entered. The protection violating command is not executed
Proposed handling by software	Usually this bit is only set during runtime because of a bug in the software. For a password failure fault condition, a reset must be performed. In any other of the fault conditions the flag can be cleared with “Clear Status” or “Reset to Read” and the corrected sequence can then be executed

4.8.4 VER “Verification Error”

Fault conditions	This flag is a warning indicator and not an error. It is set when a program or erase operation was completed but with a sub-optimal result. This bit is already set when only a single bit is left over-erased or weakly programmed which would any way be corrected by the ECC.
New state	Read mode is entered. The protection violating command is not executed
Proposed handling by software	This bit can be ignored. It should be cleared with “Clear Status” or “Reset to Read”. It must be ensured that the Flash memory is operating within the operating specification. If the application allows (i.e. timing and data logistics), a more elaborate procedure can be used to remove a VER: • VER after program: − Erase the sector and program the data again. This is only recommended when there are more than 3 program VERs in the same sector. − When programming the Flash in the field, ignoring program VER is normally the best solution because the most likely cause is violated operating conditions. − Take care to never program a sector in which the erase was aborted. • VER after erase: − The erase operation can be repeated until VER disappears. − Repeating the erase more than 3 times consecutively for the same sector is not recommended. After that it is better to ignore the VER, program the data and check its readability. The most likely cause is violated operating conditions, therefore it is recommended to repeat the erase at most once, or ignore it altogether.

Note: *Even when this flag is ignored it is recommended to clear it otherwise all following operations, including “sleep”, could trigger an interrupt even when they are successful.*

4.8.5 PFSBER/DFSBER "Single-Bit Error"

Fault conditions	When reading data or fetching code from PFLASH, the ECC evaluation detected a Single Bit Error (SBE) which was corrected. This flag is a warning indication and not an error. A certain amount of single-bit errors must be expected because of known physical effects.
New state	No state change
Proposed handling by software	This flag can be used to analyze the state of the Flash memory. During normal operation it should be ignored. In order to count SBEs, the flag must be cleared by "Clear Status" or "Reset to Read" after each occurrence. Usually it is sufficient after programming data to compare the programmed data with its reference values, ignoring the SBEs. When there is a comparison error the sector is erased and programmed again. When programming the **PFLASH** (end-of-line programming or software updates) customers can further reduce the probability of future read errors by performing the following check after programming: • Change the read margin to "high margin 0". • Verify the data and count the number of SBEs. • When the number of SBEs exceeds a certain limit (e.g. 10 in 2Mbyte) the affected sectors could be erased and programmed again. • Repeat the check for "high margin 1". − Please note that the ECC is evaluated when the data is read from the PMU. When counting single-bit errors always use the non-cached address range otherwise the error count can depend on cache hit or miss and it refers to the complete cache line. As the ECC covers a block of 64 data bits, take care to evaluate the **FSR** only once per 64-bit block. • Each sector should be re-programmed at most once. Afterwards SBEs can be ignored. • Due to the specific nature of each application the appropriate use and implementation of these measures (together with the more elaborate **VER** handling) must be chosen according to the context of the application.

4.9 Handling Flash Errors During Startup

During startup, a fatal error during Flash ramp-up forces the Firmware to terminate the startup process and to end in the Debug Monitor Mode (see Firmware chapter). The reason for a failed Flash startup can be a hardware error or damaged configuration data.

FSR.PFOPER can indicate a **problem** of a program/erase operation before the last system reset, or an error when restoring the Flash module internal SRAM content after the last reset. In both cases it is advised to clear the flag with the command sequence "Clear Status" and trigger a system reset. If the error shows up again it is an indication of a permanent fault which will limit the Flash operation to read accesses. Under this condition program and erase operations are forbidden, although not prevented by hardware.

.

Software Supervision

5 Software Supervisory

5.1 Introduction

The following aspects of fail-safe software surveillance are covered in the subsequent sections:

- Windowed Watchdog Timer (WDT)
- Flexible CRC Engine (FCE)

5.2 Windowed Watchdog Timer (WDT)

Purpose of the Window Watchdog Timer module is improvement of system integrity. WDT triggers the system reset or other corrective action like e.g. non-maskable interrupt if the main program, due to some fault condition, neglects to regularly service the watchdog (also referred to as “kicking the dog”, “petting the dog”, “feeding the watchdog” or “waking the watchdog”). The intention is to automatically bring the system back from unresponsive state into normal operation in case a severe lock-up condition occurred and only reset and reboot is an option.

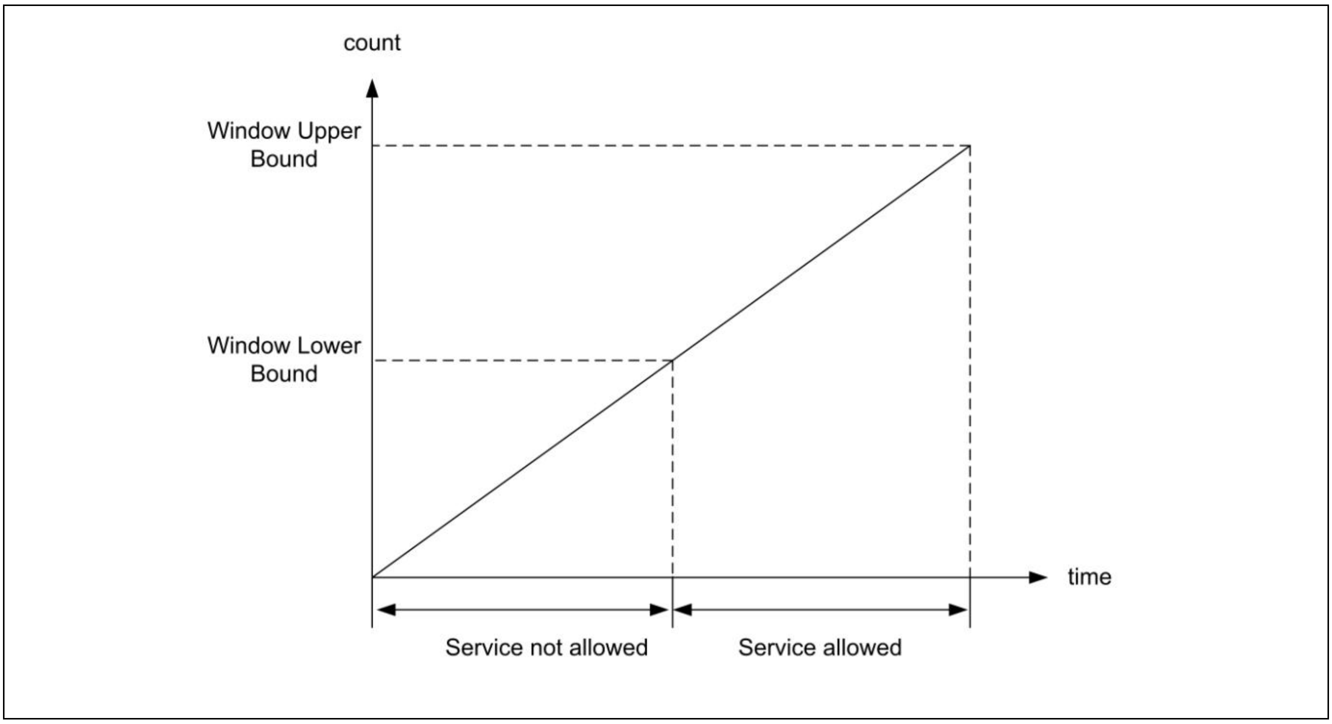

Figure 16 System Clock Supervisory

- programmable WDT with open/close window
- period and window can be programmed
- has to be refreshed in open window otherwise it will generate a RESET
- uses same clock source as CPU
  - clock fails are detected by Clock Supervisory
- WDT refresh should be done within main or low priority routines
  - usually embedded software has a predictable flow with constant periods and cycles
  - a global variable can indicate the software state
  - several user routines can manipulate one global state-variable in a way that wrong execution sequence will lead to a none-refresh of the WDT

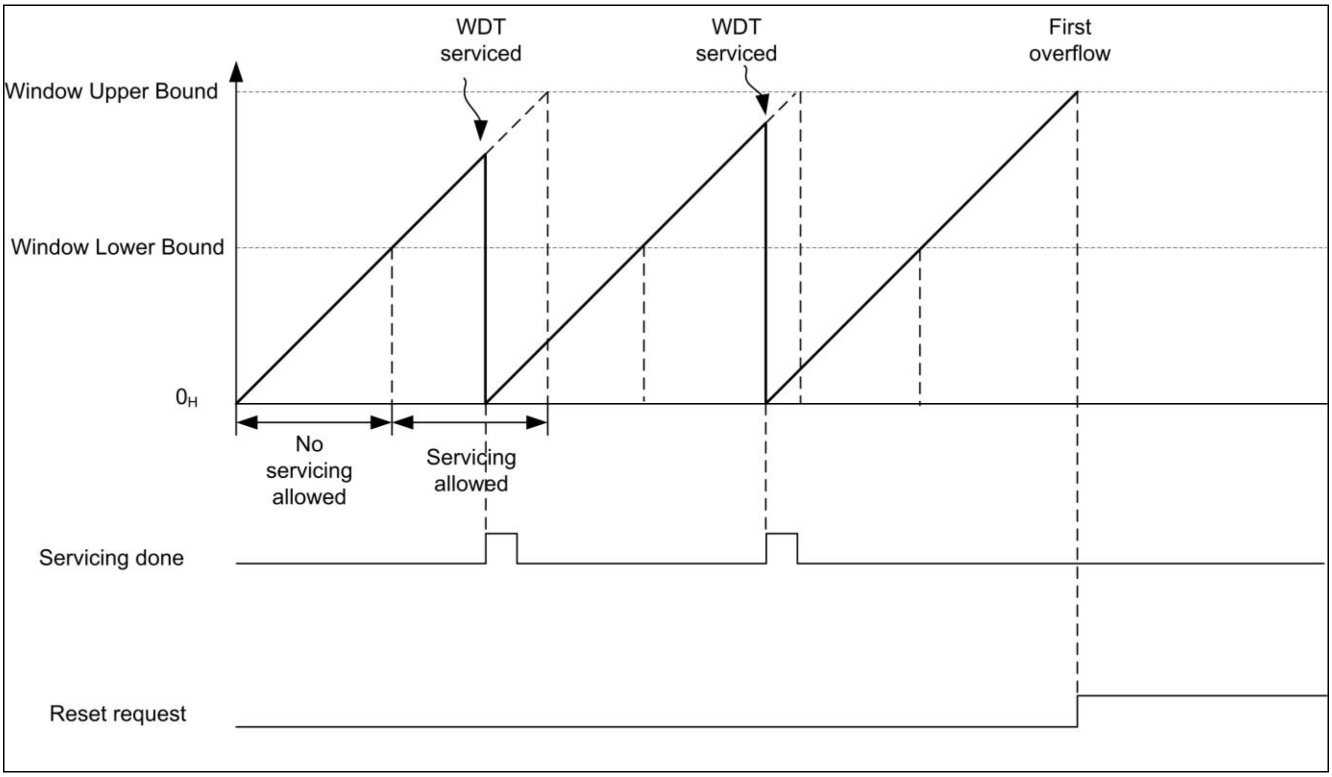

Figure 17 System Clock Supervisory

Failure condition	System un-recoverable hang up occurred.
Fail-safe effect	Triggers system reset when WDT not serviced on-time or serviced in the wrong way.

5.3 Flexible CRC Engine (FCE)

The FCE provides a parallel implementation of Cyclic Redundancy Code (CRC) algorithms. The current FCE version for the XMC4000 microcontrollers implements the IEEE 802.3 ethernet CRC32, the CCITT CRC16 and the SAE J1850 CRC8 polynomials.

The primary target of FCE is to be used as a hardware acceleration engine for software applications or operating systems services using CRC signatures.

Among other applications, CRC algorithms are commonly used to calculate message signatures to:

- Check message integrity during transport over communication channels such as internal buses or interfaces between microcontrollers
- Sign blocks of data residing in variable or invariable storage elements
- Compute signatures for program flow monitoring

Failure condition	Errors injected into communication data.
Fail-safe effect	CRC calculation of the data allows detection of errors.

System Traps

6 Traps and Interrupts

6.1 Introduction

The following aspects of fail-safe od traps and interrupt requests are covered in the subsequent sections:

- System Traps
- External Traps
- System Critical Service Requests

6.2 System Traps

Abnormal system events listed in Table 2 can result in the assertion of a NMI. Typically the occurrence of these Traps would indicate a severe malfunction that requires immediate corrective reaction in order to maintain safe system operation.

Table 2 System Traps

Short Name	**Long Name**	**Cause/Description**
SOSCWDGT	High Precision Oscillator (OSC_HP) Watchdog Trap	OSC_HP clock frequency outside of valid range or spikes present
SVCOLCKT	USB VCO Lock Trap	VCO of USB PLL lock lost
UVCOLCKT	System VCO Lock Trap	System of USB PLL lock lost
PET	Parity Error Trap	Parity error occurred on any memory configured to perform parity check
BRWNT	Brownout Trap	Supply voltage level dropped below programmed minimum level
ULPWDGT	Ultra Low Power Oscillator (OSC_ULP) Watchdog Trap	OSC_ULP clock not toggling, which may indicate for example a broken crystal
BWERR0T	Peripheral Bus 0 Write Error Trap	A bufferable write error on Peripheral Bridge 0 occurred
BWERR1T	Peripheral Bus 1 Write Error Trap	A bufferable write error on Peripheral Bridge 1 occurred
TEMPHIT	Die temperature too high Trap	Die temperature measured with DTS is above upper user limit
TEMPLOT	Die temperature too low Trap	Die temperature measured with DTS is below upper user limit

Failure condition	Abnormal system behavior indicating a risk of severe malfunction and/or hang-up.
Fail-safe Effect	System Traps are generated and propagated to CPU as an NMI interrupt. A corrective action may be required in order to bring the system back into safe operation.

6.3 External Traps

External events (Service Requests to the chip I/O pins) routed via the ERU0 module listed in Table 3, can optionally result in assertion of a NMI if configured with the **SCU_NMIREQEN** register. Otherwise, the ERU0 module Service Requests are routed to the interrupt controller (NVIC) and processed as regular interrupt requests. Occurrence of these Traps and their criticality for the system is application specific and it is a user decision what external circuit may trigger them.

Table 3 External Traps

Short Name	Long Name	Cause/Description
ERU00	Channel 0 of ERU0 NMI Request	External Service Request routed to the ERO0 triggered NMI on CPU. The status of the ERU0 Service Request is reflected in the ERU0 module register in EXICON0-3 registers.
ERU01	Channel 1 of ERU0 NMI Request	
ERU02	Channel 2 of ERU0 NMI Request	
ERU03	Channel 3 of ERU0 NMI Request	

Failure condition	External circuit/monitor detects a critical situation that requires an immediate reaction at the highest priority
Fail-safe Effect	External Service Request generated on an I/O gets propagated via ERU0 module to the system as NMI and triggers immediate CPU reaction. The user software corrective reaction may be required in order to handle a potentially critical external (application specific) condition.

6.4 System Critical Service Requests

Service Request signals typically generate system interrupts. A number of Service Request signals are intended to indicate events nearly as severe as the System Traps. Some of the Service Requests may be configured with the **SCU_NMIREQEN** register to cause NMI requests rather than an ordinary interrupt.

Table 4 System critical Service Requests

Short Name	Long Name	Cause/Description
PRWARN	WDT pre-warning	Widowed Watchdog Timer pre-warning (optional setting on WDT) if a service of the WDT did not occur in the expected time window Note: *This Service Request priority can be configured to NMI level*
PI	RTC Periodic Event	Periodic RTC event detected. This event may be required to trigger a system relevant periodic task. Note: *This Service Request priority can be configured to NMI level*
AI	RTC Alarm	RTC alarm detected. This event may be required to trigger a system relevant scheduled task. Note: *This Service Request priority can be configured to NMI level*
DLROVR	DLR Request Overrun	GPDMA Service Request overrun detected. The GPDMA module may require re-initialization.
HDCLR	HDCLR Mirror Register Updated	A write to a register in hibernated domain completed. These Service Requests are asserted after data forwarded from a writable registers in the Register Mirror has been effectively written to a target register in the hibernated domain. These registers avoid a need for polling the Register Mirror status and enable de-coupling of real-time critical routines from slow accesses to the hibernate domain.
HDSET	HDSET Mirror Register Updated	
HDCR	HDCR Mirror Register Updated	
OSCSICTRL	OSCSICTRL Mirror Register Updated	
OSCULCTRL	OSCULCTRL Mirror Register Updated	
RTC_CTR	RTC CTR Mirror Register Updated	
RTC_ATIM0	RTC ATIM0 Mirror Register Updated	
RTC_ATIM1	RTC ATIM1 Mirror Register Updated	
RTC_TIM0	RTC TIM0 Mirror Register Updated	
RTC_TIM1	RTC TIM1 Mirror Register Updated	
RMX	Retention Memory Mirror Register Updated	

Failure condition	Abnormal application behavior indicating a risk of severe malfunction of the user code.
Fail-safe Effect	Service Requests are generated and propagated to the CPU as interrupt requests or (optional in some cases) as a NMI. A corrective action may be required in order to bring the application back into safe operation

Special Peripheral Features

7 Special Peripheral Features

7.1 Introduction

The following aspects of fail-safe software surveillance are covered:

- WDT Special Functions
  - Pre-warning Service Request
  - Pre-warning Service Request
- Capture/Compare & PWM
  - Hardware Emergency Shutdown
  - Dead-Time Generation
  - Output Parity Checker
  - Shadow Register and Hardware Register Update
  - High Resolution PWM (HRPWM) Overload Protection
- Digital I/Os
  - Power Up and RESET
- ADC Analog Input Out-of-Range Monitor
- Die Temperature Measurement

7.2 WDT Special Functions

The Windowed Watchdog Timer (WDT) apart from regular monitoring of the CPU real-time behavior, offers additional features that may significantly improve its effectiveness.

7.2.1 Pre-warning Service Request

The first crossing of the upper bound triggers the outgoing alarm signal wdt_alarm when pre-warning is enabled. Only the next overflow results in a reset request (see Figure 18).

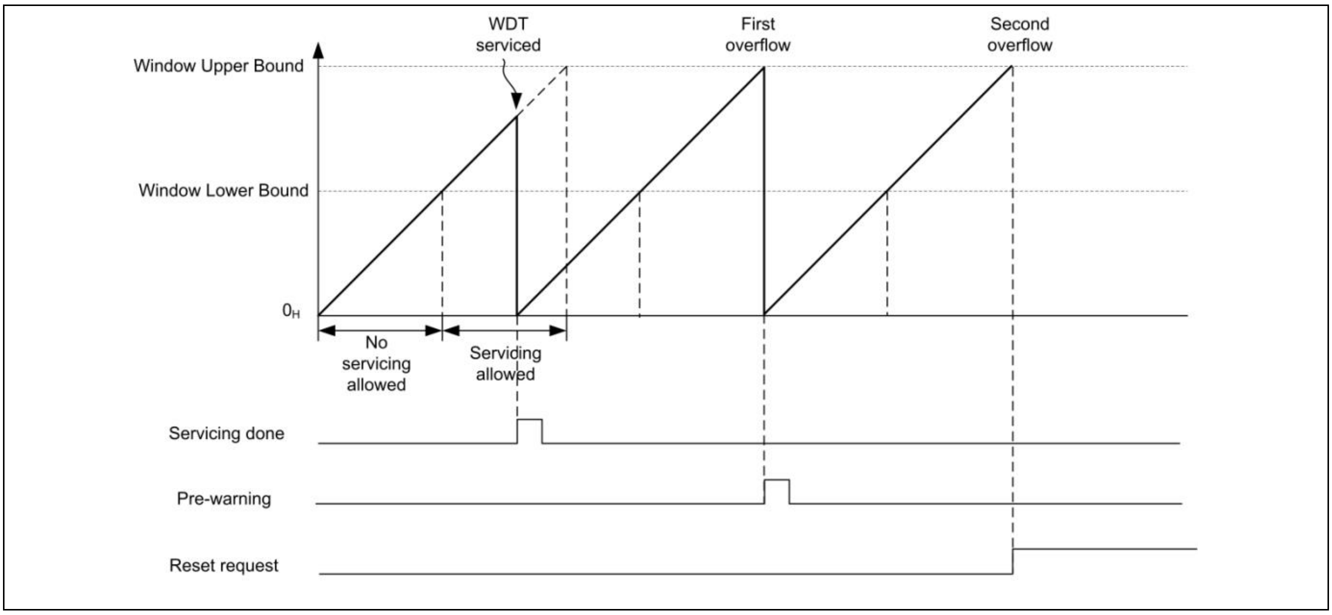

Figure 18 System Clock Supervision

The alarm status is shown via register **WDTSTS** and can be cleared via register **WDTCLR**.

A clear of the alarm status will bring the WDT back to the normal state. The alarm signal is routed as a request to the SCU, where it can be promoted to NMI level.

Failure condition	WDT did not get served on time due to lack of real-time CPU responsiveness.
Fail-safe effect	The system will not be reset on the first missing WDT service of the WDT and can still be brought into safe operation. The pre-warning signal serves a role of a “reminder” to the system that the watchdog has not been serviced and may also indicate that the real-time behavior of the system has been compromised, by too high a CPU load for example. Appropriate corrective action may be applied to reduce the CPU load and improve its real-time responsiveness.

7.2.2 WDT Service External Monitoring

As depicted in Figure 18, a correct servicing of the Windowed Watchdog Timer generates a signal that can be propagated to an external I/O and observed by an external application monitoring circuit (an external watchdog timer).

Failure condition	Severe system lock-up caused by a unrecoverable hardware or software failure where WDT did not get served on time due to a lack of real-time responsiveness of the CPU and/or complete de-activation of the WDT.
Fail-safe effect	The occurrence of a correct WDT service response may be indicated to the outside application components via an I/O (HIB_IO_0 or HIB_IO_1). A missing WDT servicing indication detected by an external watchdog may trigger a corrective hardware action, such as an external reset of the XMC4000 device for example.

7.3 Capture/Compare & PWM

The Capture/Compare and PWM modules of XMC4000 devices are typically used typically for motor control and power conversion applications. In some industrial environments the fail-safe aspect becomes particularly important as the application may involve very high power device control and potential malfunction may result in severe physical damages to equipment, injuries and/or loss of life. Therefore XMC4000 implements mechanisms that enable monitoring redundancy and emergency application shutdown.

7.3.1 Hardware Emergency Shutdown

A so-called CTRAP function allows the PWM outputs to react to the state of an input pin. This functionality can be used to switch off the power devices if a TRAP input becomes active.

- all selected PWM outputs switch immediately to its pre-programmed passive level (without software interaction)
- interrupt will be generated to inform software about fault

Failure condition	An external (redundant) monitor of the application board detects an abnormal condition that may include for example, over-current, a forbidden combination of control signals, incorrect timing, and so on. In practice this might cause severe damage, or even life threatening danger in high power applications.
Fail-safe effect	A failure occurrence can be indicated via a TRAP signal to a corresponding module of the XMC4000 device and cause a hardware shutdown of the module, bringing the control signals to en external component into a safe state that avoids any destructive effects.

7.3.2 Dead-Time Generation

The XMC4000 capture/compare modules implement dead-time insertion to prevent short circuits in the external switches. There are independent dead-time values for rising and falling transitions and an independent channel dead-time counter.

- Each channel can operate independently with their own dead-time values. This enables the control of up to 2 Half Bridges with different dead-time values and the same frequency.
- Different dead-time values for rising and falling transitions can be used to optimize the switching activity of the MOSFETs.

Failure condition	An external MOSFET switch will not function correctly and/or will get damaged if switching the transistors causes shorts or overloads due to unmatched timing of the critical control signal sequences.
Fail-safe effect	A dead-time is inserted in order to prevent critical signal transitions and to minimize undesired cross-current effects. In applications with multiple switches running independently, the dead-time parameters can be maintained individually according to their specifications.

7.3.3 Output Parity Checker

The parity checker function cross-checks the value at the output of the CCU8 module against an input signal that should be connected to a driver XOR structure.

The automated MOSFET signal monitoring functions:

- Parity checker used to monitor the output of the IGBTs and comparing them against the complete set of PWM outputs of CCU8.
- Avoiding short circuits in a multi-MOSFET system.

Failure condition	A miss-match between the driver output and the parity checker is detected.
Fail-safe effect	Interrupt is generated a Timer Slice. The interrupt status bit that stores the information is in the **CC8yINTS** Register.

7.3.4 Shadow Register and Hardware Register Update

The period and compare registers of the CCU implement an aggregated shadow register, which enables the update of the PWM period and duty cycle 'on-the-fly'. This facilitates a concurrent update by software for these parameters, with the objective of modifying the PWM signal period and duty cycle during run-time and decoupling hardware and software interaction.

Failure condition	CCU update requires an update of parameters for PWM generation within a precise and short time slot that is hard to meet using a standard software polling method, and might compromise the real-time behavior of the system by consuming a vast amount of CPU performance.
Fail-safe effect	Shadow registers can be uploaded within a more relaxed time-slot and allow decoupling of the hardware and software interaction. The new values will become effective at the right time and control of the update is performed automatically in hardware, while offloading the CPU and maintaining real-time behavior of the system.

7.3.5 High Resolution PWM (HRPWM) Overload Protection

It is possible, with a simple arrangement of resources, to have a dedicated timer for maximum ON time control. With this mechanism it is possible to limit the maximum time where the switch is ON (overload situations due to wrong measurement), avoiding a premature burn of the switch.

This feature is possible due to the fact that each High Resolution Channel (HRC), can decode two pairs of SET and CLEAR signals.

In the example of Single Phase DC/DC with fail-safe control in Figure 19, one pair can be used within the normal operating conditions while the other pair is used to over-ride and clear the PWM output to an inactive state.

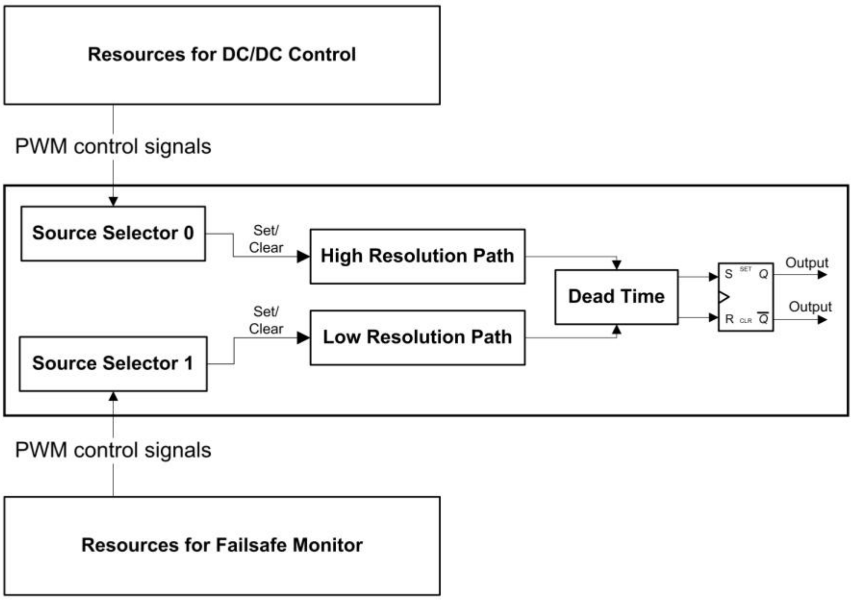

Figure 19 HRPWM Overload Protection

Failure condition	An overload condition occurs due to an incorrect measurement with control signals to a switch generated in HRPWM.
Fail-safe effect	Automatically limit the maximum time where the switch is ON, avoiding a premature burn of the switch.

7.4 Digital I/Os

7.4.1 Power Up and RESET

All I/Os on input (tri-state or weak pull-up).

- in tri-state (PWM outputs)
- all others with weak pull-up

Failure condition	XMC4000 power-up or reset occurs while other application board components are active and are driving I/Os of the device which may lead to potential damage of I/O pads (on XMC4000 or other components) if conflicting direction is configured (as outputs).
Fail-safe effect	All digital I/O of the XMC4000 are configured as inputs (high impedance), which avoids driving conflicts with external application components.

7.5 ADC Analog Input Out-of-Range Monitor

The 'out of range' comparator monitors over-voltage for the chip analog input pins. A number of analog channels are associated with dedicated pads connected to the inputs of the analog modules. Detection of input voltage exceeding VAREF triggers a Service Request to the ERU0, and (optionally) promoted to a NMI.

Dedicated registers SCU_G0ORCEN and SCU_G1ORCEN provide the means of control for enabling and disabling the monitoring of analog channels.

Failure condition	Signal voltage level on analog inputs to ADC is outside of specification (exceeding V_{AREF}).
Fail-safe effect	Service Request is triggered via the ERO0 module. The Service Request can be promoted to NMI. Application specific software reaction may be required in order to minimize the effect of an incorrect signal level on the application.

7.6 Die Temperature Measurement

The Die Temperature Sensor (DTS) generates a measurement result that indicates the current temperature. The temperature monitoring is intended to provide an additional measure for ensuring system stable operation.

Failure condition	Die temperature TJ is outside of configured limits (typically - specification limits), which may lead to a general system malfunction, erroneous Flash programming, and damage.
Fail-safe effect	Die temperature monitoring performed with the Die Temperature Sensor (DTS) allows for the detection of critical temperature operating conditions. The occurrence of a critical condition results in a System Trap. An appropriate software interaction may be required in order to bring the system back into safe operation

www.infineon.com

Published by Infineon Technologies AG