Freescale Semiconductor
Application Note

Document Number:AN4880
Rev 0, 03/2014

Handling Crystal Failure for MPC57XX

by: Shruti Maheshwari, Arun Mishra

1 Overview

The MPC57XX system-on-chip (SoC) includes a set of highly flexible modules that provide a large portfolio of internal and external clock sources. In general, crystal devices for car-mounted equipment serve in extreme and more severe conditions than in other applications. Accordingly, they are designed to ensure high reliability. However, when using external crystals or external clock sources, there is always a possibility of the source failing while the application is running. The clock fail detection logic on the MPC57XX SoCs detects clocking failures of the external crystal circuit (XTAL) on the application board but it does not automatically switch to the internal crystal oscillator circuit (IRCOSC) as the clock source. This application note discusses the ways in which loss of clock due to external oscillator (XOSC) failure can be handled correctly through software.

Contents

1.1 Introduction

The MPC57XX device family provides various clock sources; IRCOSC (Internal RC Oscillator), XOSC (External Oscillator), and PLL0/1 (Phased Lock Loop) for driving system and peripheral clocks. After power-on reset (POR), the MPC57xx devices boot from the internal 16 MHz RC oscillator (IRCOSC) and applications may run PLL with IRCOSC or XOSC as reference clock. When using XOSC as the primary or secondary clock source, there could be a situation in which the external crystal stops operating correctly or the corresponding oscillator circuit inside the SoC fails due to an open or short circuit. In the MPC57XX devices, the IRCOSC works as backup clock due to PLL or XOSC failures but there is no automatic switching by hardware. Instead, there are numerous software configurable options which can be selected by the user depending on their applications needs.

This application note is applicable to following MPC57XX devices; MPC5744P, MPC5746M, MPC5746R, MPC5777M, and MPC5775K.

2 Detailed description

MPC57XX devices have built-in mechanisms for detecting loss of the oscillator or PLL clocks, and provide several options for reaction to a loss of clock in the application. The following diagram shows the complete data flow through various blocks of SoC in case a loss of lock event is generated by the PLL.

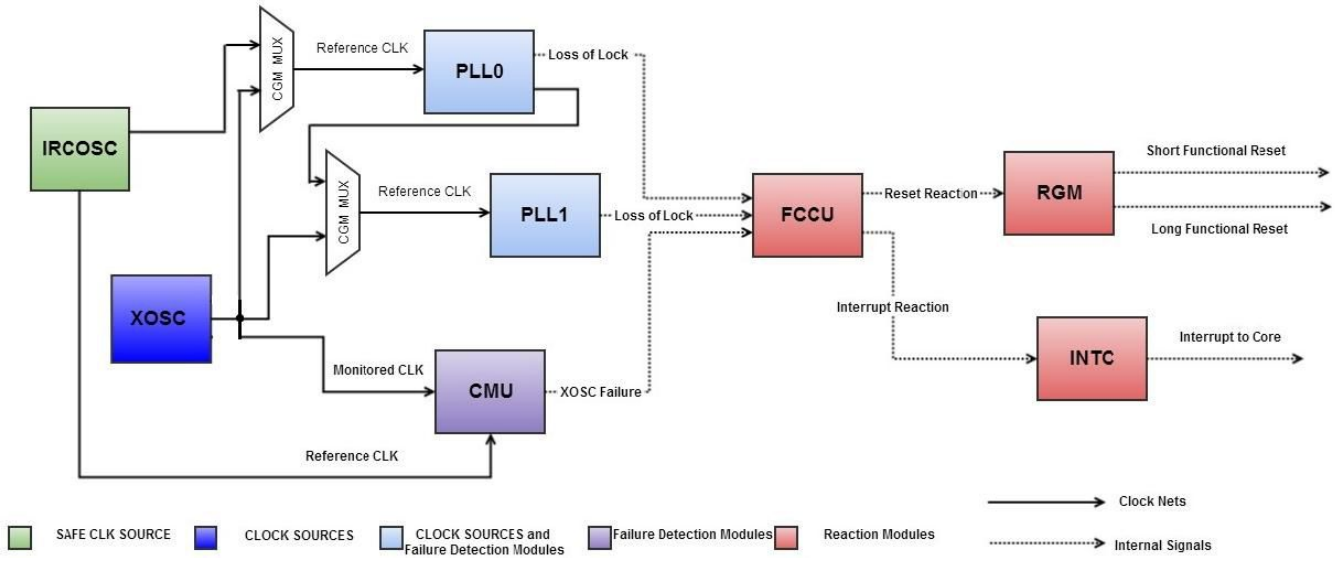

Figure 1. Data Flow Path for Loss of clock events

The data flow can be divided into the following two parts.

- Detection of Failure
- Reaction to Failure

2.1 Detection of failure

As shown in Figure 1, the blocks shown in shades of blue are different failure detection modules. The following sections explain each in more detail.

2.1.1 External Oscillator (XOSC)

The external oscillator (XOSC) is used as a clock source to the peripherals, PLL0 and PLL1 reference source and can also be selected as a system clock source.

2.1.2 Clock Monitor Unit (CMU)

To supervise the integrity of the various clock sources on the chip, there are Clock Monitoring Units (CMU) provided on chip. One such CMU also monitors frequency of XOSC clock with respect to the IRCOSC clock. Following is the block diagram of CMU monitoring XOSC clock with respect to IRCOSC.

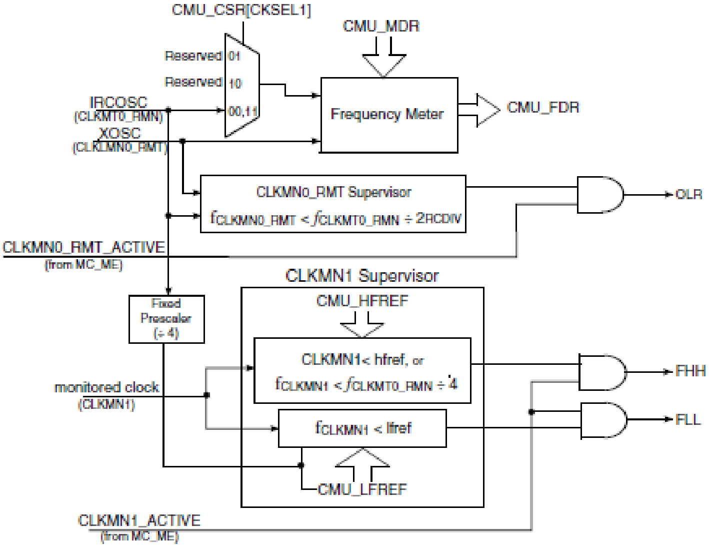

Figure 2. CMU Block Diagram

From the above diagram, it can be seen that if frequency XOSC (CLKMNO_RMT) is less than IRCOSC (CLKMTO_RMN)/ 2^ (RCDIV) then an Oscillator Less than Reference (OLR) event is generated. The CMU_CSR[RCDIV] bit field is programmable to values 0b00, 0b01, 0b10, or 0b11. Thus we can monitor XOSC and generate the OLR event if its frequency is less than IRCOSC, IRCOSC/2, IRCOSC/4, and IRCOSC/8 respectively. This provides a method to detect loss of the XOSC.

2.1.3 PLL0/1

When using PLL0/1 with XOSC as reference, the loss of lock interrupt enable (LOLIE) bit can be set in the PLL0/1_CR registers. This causes an interrupt to be generated on loss of the XOSC clock, and the internal circuits of the PLL also identify the loss of lock/reference clock and signal the Fault Collection and Control Unit (FCCU) if LOLIE bit is set in PLL0/1_CR registers.

2.2 Reaction to failure

As shown in Figure 1, the blocks shown in shades of red are the different failure detection modules. The following sections explain each module in detail.

2.2.1 Fault Collection and Control Unit (FCCU)

The fault collection and control unit (FCCU) offers a redundant hardware channel to collect errors, and as soon as a failure is detected, to lead the device to a safety state in a controlled way. There is no CPU intervention required for collection and control operation. The FCCU also has configurable and graded fault control with the following internal reactions.

- No reaction

- Interrupt
- Functional Reset
- Destructive Reset

FCCU with external reaction is described below.

- Fault report to the external system via configurable output pins.

When configured as mentioned above, each loss of lock signal from the PLLs or XOSC failure is signaled to FCCU. For the event when a clock failure occurs, the FCCU can be programmed to generate a short or long reset sequence, or simply to generate an interrupt.

2.2.2 Interrupt Controller (INTC)

The interrupt controller (INTC) has the following features.

- Provides priority-based preemptive scheduling of interrupt requests.
- Schedules interrupt requests (IRQs) from software and internal peripherals to one or more processors.
- Provides interrupt prioritization and preemption, interrupt masking, interrupt priority elevation, and protocol support.

The FCCU signals the INTC if Loss of Lock of PLL0/1 or OLR event from CMU is configured to generate an Interrupt in the case of a fault latched.

2.2.3 Reset Generation Module (RGM)

The reset generation module (RGM) centralizes the different reset sources and manages the reset sequence of the device. An FCCU reaction is one of the various reset sources for the RGM. On receiving the signal from FCCU, the RGM generates a Long functional or Short functional reset based on the configuration programmed in FCCU.

- Long functional reset implies that the flash and digital circuitry (except FCCU and STCU) undergoes initialization.
- Short functional reset implies that the digital circuitry (except FCCU and STCU) undergoes initialization.

The system clock is switched to IRCOSC after reset in both the above cases.

2.2.4 Recommendation on handling the failure

Following table shows the effects of various configurations on the system clock in case of XOSC failure.

Table 1. Fault handling configuration and system reaction

Current system clock	Reference clock	FCCU reaction configured	CMU OLR generated	PLL loss of lock generated	System clock after fault reaction
IRCOSC	Not applicable	Interrupt/ Short / Long functional reset	Yes	No	IRCOSC – Safe Clock
XOSC	Not applicable	Interrupt	Yes	No	No clock may be available leading to system hang
XOSC	Not applicable	Short /Long Functional Reset	Yes	No	IRCOSC – Safe Clock

Table continues on the next page...

Table 1. Fault handling configuration and system reaction (continued)

Current system clock	Reference clock	FCCU reaction configured	CMU OLR generated	PLL loss of lock generated	System clock after fault reaction
PLL0	IRCOSC	Interrupt	Yes	No	PLL0 maintained at configured frequency
PLL0	IRCOSC	Short /Long Functional Reset	Yes	No	IRCOSC – Safe Clock
PLL0	XOSC	Interrupt	Yes	Yes	PLL0 undetermined free running clock
PLL0	XOSC	Short /Long Functional Reset	Yes	Yes	IRCOSC – Safe Clock
PLL1	XOSC	Interrupt	Yes	Yes	PLL1 undetermined free running clock
PLL1	XOSC	Short /Long Functional Reset	Yes	Yes	PLL1 undetermined free running clock IRCOSC – Safe Clock
PLL1	PLL0 with IRCOSC as reference clock	Interrupt	Yes	No	PLL1 maintained at configured frequency
PLL1	PLL0 with IRCOSC as reference clock	Short /Long Functional Reset	Yes	No	IRCOSC – Safe Clock
PLL1	PLL0 with XOSC as reference clock	Interrupt	Yes	Yes	PLL1 undetermined free running clock
PLL1	PLL0 with XOSC as reference clock	Short /Long Functional Reset	Yes	Yes	IRCOSC – Safe Clock

As shown in above table, when the FCCU is configured to generate interrupt, an unpredictable free running clock is still provided by PLL in case of loss of lock of PLL. When the XOSC is programmed as system clock, a source failure on XOSC can cause no clock to be available to system leading to a catastrophic condition. There is no automatic system clock switch in these cases, thus the user is required to program the switch through the Mode Entry module. Therefore, it is strongly recommended that the FCCU reactions as an interrupt to a PLL loss of lock, or the OLR event from XOSC is not selected when the PLL or XOSC respectively, are used as the source of the system clock. If a long or short reset is selected, PLL, XOSC, MC_ME, and MC_CGM are reset to their default states, and the system clock is switched to the IRCOSC (Safe Clock) source.

3 Software pseudo code

The following pseudo code shows how software can be designed to implement the fault handling and system reaction.

```
//***********************************************************************
// Include the header files required for the XOSC failure handling
//***********************************************************************
#include "MPC57XX.h"
#include "fccu_api.h"
#include "cmu_api.h"
#include "mc_me_api.h"
//***********************************************************************
// The following function enables the XOSC fail detection and should be called in the
```

```
initialization code
//*****************************************************************************
void XOSCFailDetect(void)
{
    CMU_0.CSR.B.RCDIV=0;          //OLR generated when Freq XOSC< Freq IRCOSC
    CMU_0.CSR.B.CME=1              //Enable CMU 0
    PLLDIG. PLL0CR.B.LOLIE=1;      //Enable  detection of loss of lock for PLL0
    PLLDIG. PLL1CR.B.LOLIE=1;      //Enable  detection of loss of lock for PLL1

 }
//*****************************************************************************
// The following function enables the XOSC fail detection and should be called in the
initialization code
//*****************************************************************************

void XOSCFailReaction(void)
{
 /* OLR event parameters passed to FCCU structure   */
fccu_config.NCF_number = CMU0_OLR_NCF_NUM;   //mention the NCF number for FCCU
fccu_config.fault_en = 1;          //Enable fault reaction for OLR event from CMU
fccu_config.t_out_en = 0;          //Set timeout to 0
fccu_config.reset_mode = SHORT_RESET  //Set FCCU reaction as short functional reset
/*This function call does appropriate FCCU configurations based on the above mentioned
structure*/
fccu_channel_config(&fccu_config);

  /* PLL0 loss of lock event parameters passed to FCCU structure   */
fccu_config.NCF_number = PLL0_LOL_NCF_NUM ;  //mention the NCF number for FCCU
fccu_config.fault_en = 1;           //Enable fault reaction for PLL0 loss of Lock
fccu_config.t_out_en = 0;             //Set timeout to 0
fccu_config.reset_mode = SHORT_RESET  //Set FCCU reaction as Short Functional Reset
/*This function call does appropriate FCCU configurations based on the above mentioned
structure*/
fccu_channel_config(&fccu_config);

/* PLL1 loss of lock event parameters passed to FCCU structure   */
fccu_config.NCF_number = PLL1_LOL_NCF_NUM ;  //mention the NCF number for FCCU
fccu_config.fault_en = 1;           //Enable fault reaction for PLL1 loss of Lock
fccu_config.t_out_en = 0;             //Set timeout to 0
fccu_config.reset_mode = SHORT_RESET  //Set FCCU reaction as Short Functional Reset
/*This function call does appropriate FCCU configurations based on the above mentioned
structure*/
fccu_channel_config(&fccu_config);
}
```

4 Conclusion

The crystal failure detection and reaction logic provided by MPC57XX hardware is highly flexible and can be appropriately configured by proper software configuration as per the application needs. This application note also discusses various methods for recovering from a crystal oscillator failure when different clock sources are selected as the source for the system clock.

NOTE

As discussed earlier, this application note is applicable to following MPC57XX devices; MPC5744P, MPC5746M, MPC5746R, MPC5777M, and MPC5775K.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number AN4880
Revision 0, 03/2014