

MPC564xA/MPC563xM

FMPLL Initialization

by: NXP Semiconductors

Contents

1 Introduction.....	1
2 Overview.....	1
2.1 FMPLL features.....	2
3 Procedure.....	2
4 Code.....	3

1 Introduction

This application note describes the proper method to initialize and change the frequency of the Frequency Modulated Phase Lock Loop (FMPLL) module for the MPC564xA and MPC563xM families of devices.

2 Overview

The FMPLL allows the users to generate high-speed system clocks from a crystal oscillator or from an external clock generator and supports programmable frequency modulation (FM) of the system clock. The programmable registers control the FMPLL multiplication factor (MFD), reference clock predivider factor (PREDIV), output clock divider ratio (RFD), modulation depth, and multiplication rate. See [Figure 1](#) below.

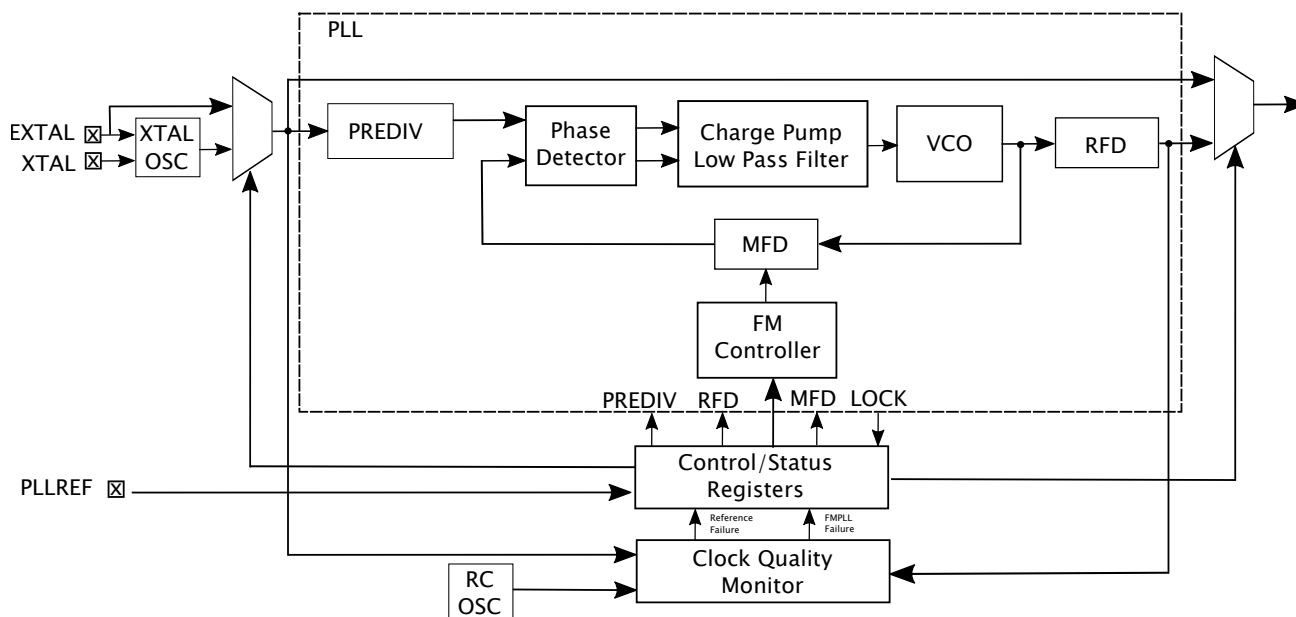


Figure 1. FMPLL Block Diagram



2.1 FMPLL features

Here are the features of FMPLL:

- FMPLL reference clock is pin selectable between either a crystal oscillator or external clock
 - PLLREF pin driven high at reset - crystal oscillator selected as reference clock
 - PLLREF pin driven low at reset - external clock selected as reference clock
- Reference Clock Predivider (PREDIV) for finer frequency synthesis resolution
- Reduced Frequency Divider (RFD) for reducing the FMPLL output clock frequency without forcing the FMPLL to relock
- Four bypass modes: crystal or external reference with PLL on or off
 - Bypass mode is when the reference clock is selected as the system clock
- Two normal modes: crystal or external reference
 - Normal mode is when the FMPLL output is selected as the system clock
- Programmable frequency modulation
- Lock detect circuitry reports when the FMPLL has achieved frequency lock and continuously monitors lock status to report loss of lock conditions
 - can generate an interrupt request upon loss of lock
 - can generate a system reset upon loss of lock
- Clock Quality Monitor (CQM) module provides loss-of-clock detection for the FMPLL reference and output clocks
 - can generate an interrupt request upon loss of clock
 - can generate a system reset upon loss of clock
 - can apply backup block to the system in case of loss of clock

Out of reset, the FMPLL is turned on running in bypass mode. The reference clock is determined by the external pin PLLREF and reflected in the least significant bit of the CLKCFG field in the FMPLL_ESYNCR1 register.

3 Procedure

For the MPC564xA and MPC563xM families, you can run in legacy or enhanced mode. These procedures apply to enhanced mode. The 'E' prefix on the divider names, EPREDIV, EMFD, etc., indicates enhanced mode dividers. These dividers perform the same functions as the corresponding non 'E' prefixed dividers in [Figure 1](#).

The recommended procedure to program the FMPLL and engage normal mode is:

- **Enhanced mode with no FM**
 1. Disable all clock monitoring functions in the FMPLL_ESYNCR2 register.
 2. Power off the FMPLL by writing to FMPLL_ESYNCR1[CLKCFG].
 3. Program the EMODE bit of the FMPLL_ESYNCR1 register for enhanced mode.
 4. Program the EPREDIV and EMFD fields of FMPLL_ESYNCR1 and the ERFD field of FMPLL_ESYNCR2.
 5. Poll FMPLL_SYNSR[LOCK] until it asserts.
 6. Engage normal mode by writing to FMPLL_ESYNCR1[CLKCFG].

- **Enhanced mode with FM**

1. Disable all clock monitoring functions in the FMPLL_ESYNCR2 register.
2. Power off the FMPLL by writing to FMPLL_ESYNCR1[CLKCFG].
3. Program the EMODE bit of FMPLL_ESYNCR1 register for enhanced mode.
4. Program the EPREDIV and EMFD fields of FMPLL_ESYNCR1 and the ERFD field of FMPLL_ESYNCR2.
5. Poll FMPLL_SYNSR[LOCK] until it asserts.
6. Program the FMPLL_SYNMMR with desired FM parameters, poll the BSY bit until it negates, and then enable FM by asserting the MODEN bit.
7. Engage normal mode by writing to FMPLL_ESYNCR1[CLKCFG].

The most robust method for initializing the FMPLL requires that the FMPLL is powered down before programming any configurations. Although it may be possible to leave the FMPLL on while programming it is never recommended to have the FMPLL selected as the system source while configuring it.

Some important information to keep in mind while programming and using the FMPLL:

- EPREDIV must not be set to any value that causes the input frequency to the phase detector to go below 4 MHz.
- The LOCK flag is immediately negated if any of the following fields of FMPLL_ESYNCR1 are changed: EMODE, EPREDIV, EMFD, or the two least significant bits of CLKCFG.

NOTE

Changing only the most significant bit of the CLKCFG field to move from bypass to normal or vice-versa while keeping the values of the other FMPLL_ESYNCR1 fields unchanged will not cause the FMPLL to lose lock or the lock flag to be cleared.

- To prevent an immediate reset, the LOLRE bit must be cleared before doing any of the above-mentioned operations. Only set the LOLRE bit once the FMPLL is locked.
- Changing ERFD does not affect the FMPLL lock status.

4 Code

Here is sample code for initializing the FMPLL on the MPC5644A:

```
*=====*
* COPYRIGHT:                                     *
* NXP Semiconductor, INC. All Rights Reserved. You are hereby       *
* granted a copyright license to use, modify, and distribute the    *
* SOFTWARE so long as this entire notice is retained without alteration *
* in any modified and/or redistributed versions, and that such modified *
* versions are clearly identified as such. No licenses are granted by *
* implication, estoppel or otherwise under any patents or trademarks *
* of NXP Semiconductor, Inc. This software is provided on an        *
* "AS IS" basis and without warranty.                               *
*                                                                    *
* To the maximum extent permitted by applicable law, NXP           *
* Semiconductor DISCLAIMS ALL WARRANTIES WHETHER EXPRESS OR IMPLIED, *
* INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  *
* PARTICULAR PURPOSE AND ANY WARRANTY AGAINST INFRINGEMENT WITH REGARD *
* TO THE SOFTWARE (INCLUDING ANY MODIFIED VERSIONS THEREOF) AND ANY  *
* ACCOMPANYING WRITTEN MATERIALS.                                   *
*                                                                    *
* To the maximum extent permitted by applicable law, IN NO EVENT    *
* SHALL NXP Semiconductor BE LIABLE FOR ANY DAMAGES WHATSOEVER      *
*=====*
```

Code

FMPLL features

```

* (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS,
* BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER
* PECUNIARY LOSS) ARISING OF THE USE OR INABILITY TO USE THE SOFTWARE.
*
* NXP Semiconductor assumes no responsibility for the
* maintenance and support of this software
*
*****/

void ClkInit(void)
{
    /**-----Disable clock Monitoring flags-----*/
    FMPLL.ESYNCR2.R = 0x00000000;    /** Ensure loss of lock reset is disabled */
                                     /** Ensure loss of clock reset is disabled */
                                     /** Ensure Loss of lock IRQ is disabled */

    /**-----Disable PLL-----*/
    FMPLL.ESYNCR1.B.CLKCFG = 0x01;    /** Clock Mode :Bypass w/ Crystal Ref PLL Off */

    /**-----Clock Configuration Mode-----*/
    FMPLL.ESYNCR1.B.EMODE = 0x1;    /** The Clock Configuration Mode : Enhanced Mode */

    /**-----Settings-----*/
    FMPLL.ESYNCR2.B.ERFD = 0x0;    /** PLL Ramping Step 1 Divider: 2 */
    FMPLL.ESYNCR1.B.EPREDIV = 0x01;    /** Enhanced Pre-Divider Selected :2 */
    FMPLL.ESYNCR1.B.EMFD = 0x4B;    /** PLL Ramping Step 1 Multiplier: 75 */
    FMPLL.ESYNCR1.B.CLKCFG = 0x03;    /** Clock Mode :Bypass w/ Crystal Ref and PLL On */

    while (FMPLL.SYNSR.B.LOCK == 0){}; /** Wait for FMPLL to acquire lock */
    FMPLL.SYNSR.B.LOLF = 0x1;    /** Clear the Loss-of-Lock Flag */

    /**-----FMPLL Setup-----*/
    /** FMPLL.SYNFMMR.R = 0x00000000; /** Frequency Modulation : Disabled */
    /** Wait for the FMPLL processing to complete */
    /** while (FMPLL.SYNFMMR.B.BSY == 1){}; */

    /**-----Clock Configuration Mode-----*/
    FMPLL.ESYNCR1.B.CLKCFG = 0x07;    /** Clock Mode :Normal w/ Crystal Ref PLL On */
}

```


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, QorlQ, QorlQ 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.

AN11960
Rev. 0
May 2017

