**Freescale Semiconductor**
Engineering Bulletin

Document Number: EB790
Rev. 0, 03/2013

# Configuration of Phase Fractional Dividers

## 1 Clock generation using PFDs

The i.MX 6 series has several phase-locked loops (PLLs) used for clock generation. Certain PLLs also support phase fractional dividers (PFDs) to generate additional clock outputs. The use of these PFD clock outputs requires that users ensure they are correctly configured. Incorrect configuration can cause a loss of the PFD output clock, leading to processor malfunction including boot up failure.

This document briefly describes PFD and PLL functionality and proper configuration, and provides a pseudocode example. Although the document focuses on the i.MX 6Dual/6Quad SoC, the underlying PFD configuration principles discussed here apply to the entire i.MX 6 series.

### 1.1 System and USB1 PLLs

The System PLL (PLL2) on i.MX 6Dual/6Quad SoC synthesizes a low-jitter clock from the 24 MHz reference oscillator clock. The PLL has one primary output clock which defaults to 528 MHz with an additional three programmable PFD outputs. The three PFD outputs generated by the System PLL can be at different frequencies, as shown in Table 1. The clock outputs can be routed to various modules depending on the clock configuration performed in the Clock Controller Module (CCM). Typically, this PLL is a clock source for internal system buses, internal processing logic, the DDR memory interface, NAND/NOR memory interface modules, etc.

PLL2 on i.MX 6Dual/6Quad SoC supports spread spectrum modulation for use in applications to minimize radiated emissions. Due to this feature support, the associated lock time of this PLL is longer than that of PLLs in the SoC which do not support spread spectrum modulation.

| Clock Output | Default Frequency[1] (MHz) | Description |
|---|---|---|
| PLL2 | 528 | Default PLL output frequency when not in Bypass mode |
| PFD0 | 352 | PFD0 Output sourced from PLL2 |
| PFD1 | 594 | PFD1 Output sourced from PLL2 |
| PFD2 | 396 | PFD2 Output sourced from PLL2 |

**Table 1. i.MX 6Dual/6Quad System PLL and associated PFDs**

1 The default PFD frequency is the value configured at boot. Software uses them as fixed-frequency sources but users may change these values to suit their application.

The USB1 PLL (PLL3) synthesizes a low-jitter clock from the 24MHz reference clock. PLL3 has an output clock (480 MHz) which is used by the USB module and has four additional frequency-programmable PFD outputs, as shown in Table 1-2. The four PFD outputs generated by PLL3 can be routed to various modules, depending on the clock configuration performed in the CCM.

The main PLL output is for the USB module, while its PFD outputs are used as inputs for many clock roots which require a constant frequency, such as UART, CAN and other serial & audio interfaces. PLL3's lock time is shorter than that of the System PLL, as it does not support spread spectrum modulation.

| Clock Output | Default Frequency[1] (MHz) | Description |
|---|---|---|
| PLL3 | 480 | Default USB 1PLL output frequency when not in Bypass mode. Dedicated fixed frequency clock for the USB module. |
| PFD0 | 720 | PFD0 Output sourced from PLL3 |
| PFD1 | 540 | PFD1 Output sourced from PLL3 |
| PFD2 | 508.2 | PFD2 Output sourced from PLL3 |
| PFD3 | 454.7 | PFD3 Output sourced from PLL3 |

**Table 1-2. i.MX 6Dual/6Quad USB1 PLL and associated PFDs**

1 The default PFD frequency is the value configured at boot. Software uses them as fixed-frequency sources but users may change these values to suit their application. On other i.MX 6 series SoCs, the PFDs are reprogrammed to different frequencies at boot.

Out of reset, the i.MX 6Dual/6Quad System and USB1 PLLs are configured as follows:

- Powered Down mode—most of the PLL circuitry is switched off
- Bypass mode—directly routes input reference clocks to output
- Output Enabled—PLL clock output is enabled

When configured in Bypass mode the PLL directly routes its input reference clocks to the PLL output. The 24 MHz Oscillator clock (bypass clock) is selected at the output. Bypassing the PLL is performed by setting the analog bypass bit in the respective control register for that PLL.

On the i.MX 6Dual/6Quad SoC all PFDs are sourced of the System and USB1 PLLs, as shown in Figure 1. For the PLLs equipped with PFDs, the input reference clock is also bypassed to all PFD outputs.

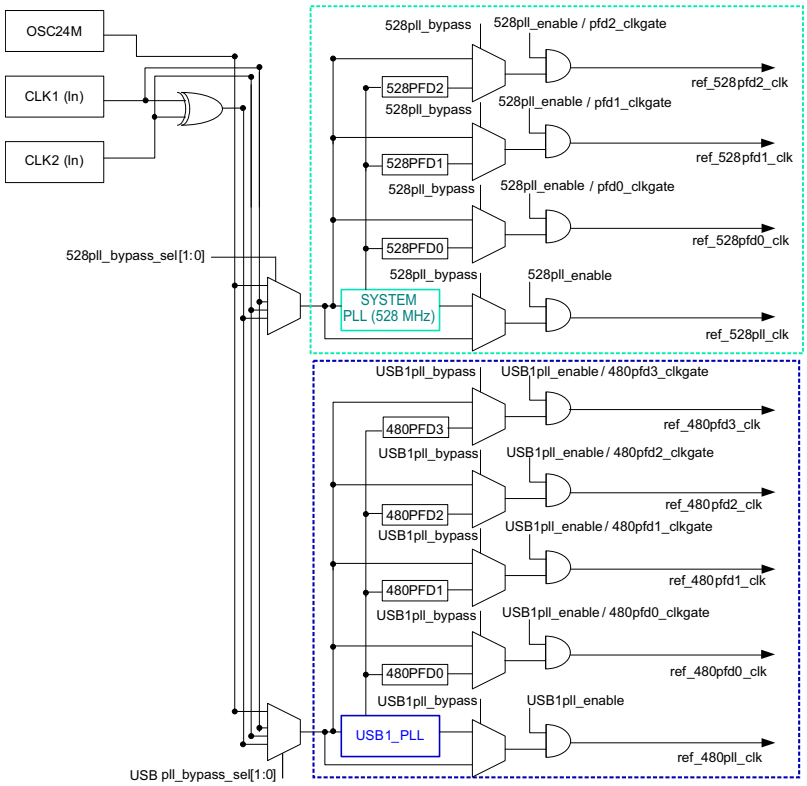


**Figure 1. System and USB1 PLL clock generation**

## 1.2 PFD operation

Each PFD output generates a fractional multiplication of the associated PLL's VCO frequency. The PFDs allow for clock frequency changes without forcing the re-lock of the root PLL. The PFD is a completely digital design with no analog components or feedback loops. The frequency switch is much faster than a PLL, and this feature is useful for supporting dynamic voltage and frequency scaling (DVFS).

The PFDs each source clocks from either the System PLL or the USB1 PLL. The i.MX 6Dual/6Quad PFDs, out of reset, are configured as follows:

- Bypass mode—Input reference clock is bypassed to all PFD outputs
- Clock Un-gated—PFD output is not gated
- Output Enabled—Shared PLL output enabled outputting the 24 MHz clock

Users should note that there is only one bypass bit and one enable bit associated with each PLL and PFD group, but both are applied to all the PFDs that are source from it. Each PFD has individual clock gate bits (PFDx_CLKGATE).

### 1.2.1 Enabling PFDs

When the PLL is powered on, it may output an erratic clock until lock is achieved. Because the PLL clock output is un-gated, this clock can go unimpeded to the un-gated PFDs and can put them in an unrecoverable state. In this state the PFD will not output a clock, affecting all modules sourced from the PFD.

The longer the lock time of the PLL, the higher the chance that an erratic clock pulse may place the PFD in an unrecoverable state. The PFDs sourced from the System PLL with the longer lock time are more susceptible to this issue. This possible loss of clock issue is limited to the PFDs that have clocks sourced from either the System PLL or the USB1 PLL. Only the PFDs of the source PLL which are being locked are affected.

To prevent this random scenario, PFDs should be reset after the PLL lock is achieved. This should be done before PFDs are used (bypass disabled). The PFD reset is achieved by setting and clearing the PFDx_CLKGATE bit in CCM_ANALOG_PFD_480n or CCM_ANALOG_PFD_528n registers. Once the PLLs lock and PFDs are reset, the clocks generated from the PFD can be used to safely source other clock trees within the SoC.

Steps to correctly configure the PFDs sourced from the System PLL are as follows:

1. PLL is powered down, bypassed and output enabled out of reset.
2. Power on the PLL to start the lock process.
3. Wait for the PLL to achieve lock by polling the PLL LOCK bit.
4. Switch the bus fabric clock from periph_clk to the 24 MHz Oscillator clock.
5. Wait for the periph_clk_sel_busy bit to clear indicating handshake is complete.
6. Gate the PFD by setting all the PFDx_CLKGATE bits in the CCM_ANALOG_PFD_528n register.
7. Un-gate the PFD by clearing all the PFDx_CLKGATE bits in the CCM_ANALOG_PFD_528n register.
8. Remove the PLL bypass.
9. Switch back to periph_clk.
10. Wait for the periph_clk_sel_busy bit to clear indicating handshake is complete.
11. It is safe to use all PFD-generated clocks.

The pseudocode example below configures the PFDs sourced from the i.MX 6Dual/6Quad System PLL.

```
/* Clear the System PLL Power Down bit to Power up PLL */
hapi_anatop_reg.hw_anadig_pll_528 &= ~ANATOP_PLL_PWDN_MASK;

/* Wait for the System PLL to Lock */
while(!(hapi_anatop_reg.hw_anadig_pll_528 & ANATOP_PLL_LOCK));

/* Use glitch-less mux to switch to the 24 MHz Oscillator Clock */
hapi_clock_reg.cbcdr |= CBCDR_PERIPH_CLK_SEL_MASK;
```

```
/* Wait for mux to go idle after the handshake */
while(hapi_clock_reg.cdhipr & CDHIPR_PERIPH_CLK_SEL_BUSY);

/* Reset PFD logic by clock gating and un-gating them */
hapi_anatop_reg.hw_anadig_pfd_528 |= ANATOP_PFD528_CLKGATE_MASK;
hapi_anatop_reg.hw_anadig_pfd_528 &= ~(ANATOP_PFD528_CLKGATE_MASK);

/* Disable the PLL bypass */
hapi_anatop_reg.hw_anadig_pll_528 &= ~ANATOP_PLL_BYPASS_MASK;

/* Use glitch-less mux to switch back */
hapi_clock_reg.cbcdr &= ~CBCDR_PERIPH_CLK_SEL_MASK;

/* Wait for mux to go idle after the handshake */
while(hapi_clock_reg.cdhipr & CDHIPR_PERIPH_CLK_SEL_BUSY);
```

**NOTE**

Additional handshaking procedures with the MMDC are required for other i.MX 6 series devices when the MMDC clock is being changed and sourced from a PFD, instead of the default System PLL2.

### 1.2.2 Disabling PFDs

As long as the PLL is not in Bypass mode, the PLL will automatically disable its clock output when the PLL unlocks during power down. This is done to prevent an unstable clock from propagating if the user forgets to disable the output manually. When the PLL's output is disabled, it also automatically clock-gates the PFDs, in addition to disabling their outputs. When the PLL is enabled again, it un-gates the PFDs and enables the clocks to propagate again. Because the PFDs go through a clock gate/un-gate cycle they are essentially reset and can be used safely in the future.

There is only one bypass bit and one enable bit associated with the PLL/PFD group. These bits reside in the respective PLL configuration registers but are applied to all the PFDs. If the PLL output is disabled (CCM_ANALOG_PLL_SYSn_ENABLE =0) then all the PFD outputs will also be disabled regardless of the clock gate bits (PFDx_CLKGATE).

**NOTE**

If when the PLL is bypassed (CCM_ANALOG_PLL_SYSn_BYPASS =1), an unlock event is caused, users must set the PFD clock gate bits (PFDx_CLKGATE) manually.

Users should not assert the PFDx_CLKGATE bit while its output is being used to source key clock roots in the CCM (e.g., Fabric clocks, AHB) as this will kill the clock and hang the system. This operation must be done before the switch is made to the PFDs in the CCM, or by using the periphx_clk_sel muxes to switch to a PLL or OSC clock before resetting the PFDs.

PLLs have certain requirements when being disabled. For more information, see the i.MX 6 series reference manuals (IMX6DQRM, IMX6SDLRM, or IMX6SLRM), available on www.freescale.com.

### 1.2.3 PFD operation in suspend and resume

As long as the PLL is not bypassed when the system enters STOP mode, the CCM automatically clock-gates the PFD before disabling the PLL, and then enables the PLL before un-clock-gating the PFD. This mechanism effectively resets the PFDs; an explicit PFD reset sequence is not required in the suspend/resume case, as long as the PLL is not bypassed. If the PLL is bypassed when entering suspend, users must ensure that the respective PFDx_CLKGATE bits are set.

### 1.2.4 PFD restrictions

A few restrictions on the usage of the PFDs are summarized below:

1. PFDs are not usable before reset unless PLL is in Bypass mode
2. PFDs must be reset by clock-gating and un-gating them after the source PLL locks
3. Although a PFD can be configured using the fractional dividers, they are asynchronous and will not maintain a constant phase offset between clocks
4. PFDs can be gated when bypassed (PFDs output no clock when they are both bypassed and gated)
5. Users should not assert PFDx_CLKGATE bits while the PFD output is being used to source key clock roots, as this can hang the system
6. When changing PFD clock sources with muxes, users should ensure that the correct clock-switching procedures are used (normal and glitchless)
   - Switch must occur only when the MUX output is not in use
   - Switch must occur only when both MUX inputs are available
   - Appropriate MMDC handshaking is performed for the respective channels
7. The respective PFDx_CLKGATE bits must be asserted before the PLL is powered down

***How to Reach Us:***

**Home Page:**
freescale.com

**Web Support:**
freescale.com/support



Document Number: EB790
Rev. 0
03/2013