AN13675

S32Z2/E2 Clock Configuration Guide

Rev. 1.0 — 7 April 2026 **Application note**

Document information

Information	Content
Keywords	S32Z2/E2, PLL, Aurora, LFAST, S32DS, CGM, Clock Gating, EB Tresos
Abstract	This application note describes the S32Z2/E2 clocking architecture and how to configure each component within it using S32DS, EB Tresos, and the S32Z2E2 Clock Configuration Tool.

1 Introduction

S32E2 and S32Z2 are 32-bit Arm®-based microcontrollers (MCUs). They target real-time applications that require low latency and high performance, including applications for:

- Braking
- Hybrid and electric vehicles (HEVs/EVs)
- Safety
- Chassis
- Domain control

S32E2 builds on and expands NXP's HEV/EV portfolio that includes products such as MPC5744P, MPC5777C, MPC5775B, and MPC5775E. S32E2 is an EV integration platform that supports applications such as:

- Electric traction motor control
- Hybrid Control Unit (HCU) control
- Advanced combustion engine management

Multiple clock sources are supported for clock generation:

- FIRC
- SIRC
- FXOSC
- PLLs
- DFS modules

This application note is intended to provide the user guidance on how to configure clock settings for S32Z2/E2. This document includes PLL, DFS, CGM and clock gating modules and introduces configuration with EB Tresos[ref.[6]], S32DS Config Tool[ref.[5]], and the S32Z2E2 Clock Configurator. This document complements the S32Z2/E2 Reference Manual[ref.[1]ref.[2]] and S32Z2/E2 Data Sheet[ref.[3]ref.[4]]. Readers are advised to read through “Clocking” chapter from S32Z2/E2 Reference Manual[ref.[1]ref.[2]] before diving further into this document. Table 1 shows the abbreviations used throughout the document.

Table 1. Acronyms and Abbreviations

Abbreviation	Explanation
PLL	Phase Locked Loop
DFS	Digital Frequency Synthesizer
CGM	Clock Generation Module
FIRC	Fast Internal RC Oscillator
SIRC	Slow Internal RC Oscillator
FXOSC	Fast Crystal Oscillator Digital Controller
SMU	System Manager Unit
RTU	Real-Time Unit
PCFS	Progressive Clock Frequency Switching

Note: *For how to configure and use PCFS please refer to AN13631: S32Z2/E2 Progressive Clock Frequency Switching (PCFS).*

2 PLL – CORE, PERIPH, DDR

A PLL can multiply or divide the frequency of a given clock input. S32Z2/E2 supports multiple PLLs, including:

- CORE_PLL
- PERIPH_PLL
- DDR_PLL
- AURORA PLL
- LFASTn_PLL

This section covers the following PLLs that S32Z2/E2 supports:

- CORE_PLL
- PERIPH_PLL
- DDR_PLL

See PLL - AURORA and PLL - LFAST for information on those PLLs.

The data flow of the CORE_PLL, PERIPH_PLL, and DDR_PLL blocks can be found in the chapter “PLL Digital Interface (PLLDIG)” in the Reference Manual[ref.[1] ref.[2]]. It is also shown in the figure below.

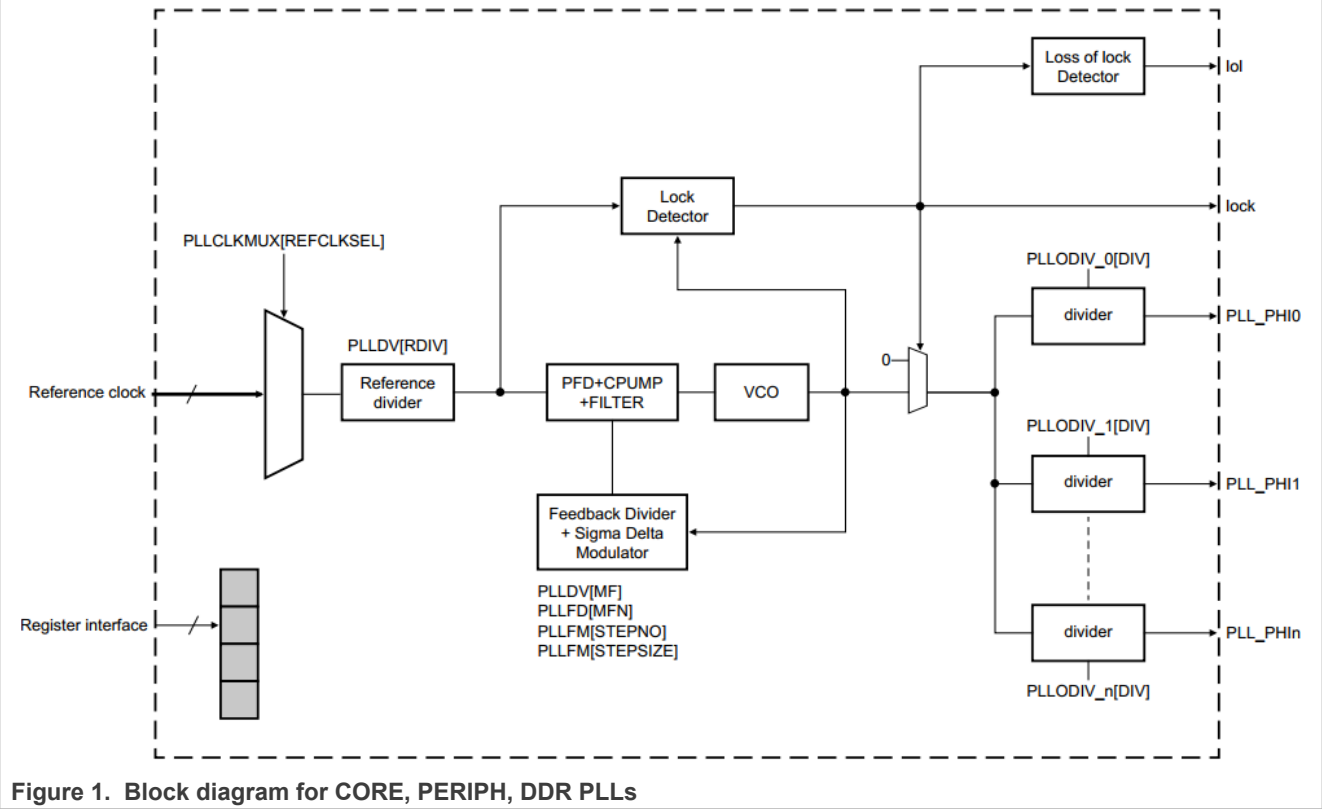


Figure 1. Block diagram for CORE, PERIPH, DDR PLLs

The user needs to configure the values for the parameters below to achieve the target frequencies for PLL_VCO and PLL_PHIn.

- 1. Reference clock: The clock source for the PLLs can either be the 20 – 40 MHz FXOSC or 48 MHz FIRC. During boot, FIRC_CLK is used as the default PLL reference clock. After boot, the PLL reference must be changed to FXOSC_CLK. Ensure that PLLCLKMUX[REFCLKSEL] is selected accordingly.
- 2. RDIV: The PLL input reference clock frequency after the reference divider should be between 20 – 40 MHz, therefore the valid values for RDIV are shown below.

Table 2. RDIV values

Frequency	RDIV
FXOSC – 20MHz	1
FXOSC – 24MHz	1
FXOSC – 40MHz	1 or 2
FIRC – 48MHz	2

Note: *For a crystal of 40 MHz, NXP recommends using RDIV = 1 for better jitter performance.*

- 3. MFI: Integer Portion of Loop Divider
- 4. MFN: Numerator of Fractional Loop Division Factor
- 5. DIV: Division value
- 6. STEPSIZE: Step size for modulation depth and frequency in frequency modulation mode
- 7. STEPNO: Number of steps to achieve modulation depth in frequency modulation mode

Note: *STEPSIZE and STEPNO are FM parameters and are not applicable to PERIPH_PLL.*

2.1 PLL Calculation

2.1.1 Modes of Operation

The PLL digital interface supports 4 modes of operation:

1. Disable
2. Integer Functional mode - PLL operates in Integer-only mode.
3. Fractional Functional mode - PLL operates in Fractional mode.
4. FM Functional mode - PLL operates in Frequency Modulation mode. (PERIPH PLL does not support FM)

For Integer-only mode:

$$f_{pll\_VCO} = \frac{f_{pll\_ref}}{PLLDV[RDIV]} \times PLLDV[MFI]$$

If $\boldsymbol{PLLDV}[\boldsymbol{RDIV}] = 0$,this should be calculated as $\boldsymbol{PLLDV}[\boldsymbol{RDIV}] = 1$.

For Fractional mode:

$$f_{pll\_VCO} = \frac{f_{pll\_ref}}{PLLDV[RDIV]} \times \left(PLLDV[MFI] + \frac{PLLFD[MFN]}{18432} \right)$$

If $\boldsymbol{PLLDV}[\boldsymbol{RDIV}] = 0$, this should be calculated as $\boldsymbol{PLLDV}[\boldsymbol{RDIV}] = 1$.

For Frequency Modulation mode:

The f_{pll_VCO} is calculated the same way as Fractional mode. To achieve the expected modulation frequency (f_{MOD}) and modulation depth (MD), the following additional parameters are needed.

The expected f_{MOD} should be between 30KHz and 64KHz. The expected MD should meet the following restrictions:

$$Max\left(MD\ \%\right) = \frac{\left(f_{REF} \times 100\right)}{PLLDV[RDIV] \times f_{pll\_VCO}}$$

If $\boldsymbol{PLLDV}[\boldsymbol{RDIV}] = 0$, this should be calculated as $\boldsymbol{PLLDV}[\boldsymbol{RDIV}] = 1$.

$PLLFM[STEPNO]$ and $PLLFM[STEPSIZE]$ can be obtained from the following equations:

$$LDF = PLLDV[MFI] + \frac{PLLFD[MFN]}{18432}$$

$$PLLFM[STEPNO] = \frac{f_{REF}}{(2 \times f_{MOD} \times PLLDV[RDIV])}$$

If $\boldsymbol{PLLDV[RDIV]} = 0$, this should be calculated as $\boldsymbol{PLLDV[RDIV]} = 1$.

$$PLLFM[STEPSIZE] = \frac{MD \times LDF}{(100 \times PLLFM[STEPNO])} \times 18432$$

Frequency modulation is only possible if the condition shown in the below equation is met.

$$PLLFM[STEPNO] \times PLLFM[STEPSIZE] < 18432$$

PLL also provides a control field PLLFM[SPREADCTL] to select the modulation spread mode to be center-spread modulation. Figure 2 illustrates center spread modulation mode.

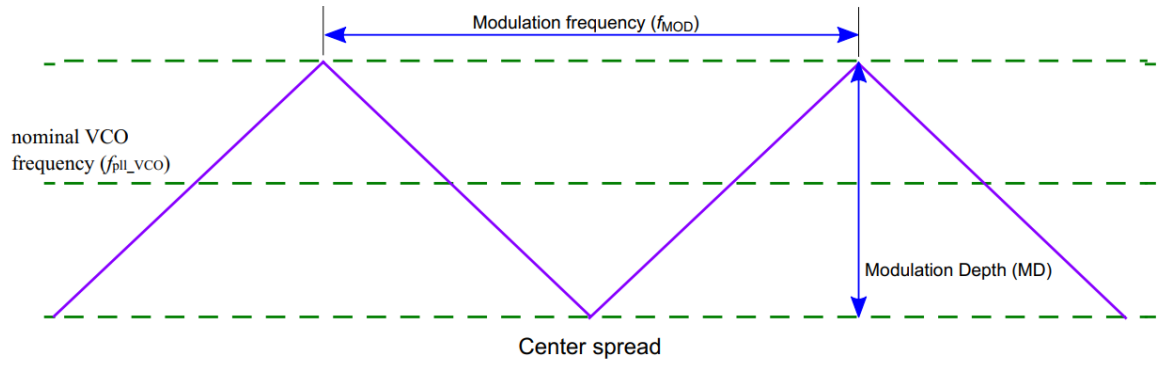


Figure 2. Frequency modulation

2.1.2 PLL PHI clock

The field DIV in the PLLODIV_n register divides the PLL clock into the respective PLL_PHI clocks, where each PLL_PHIn corresponds to a PLLODIV_n register.

$$f_{pll\_phi} = \frac{f_{pll\_VCO}}{PLLODV\_n[DIV+1]}$$

2.2 PLL Outputs and Constraints

2.2.1 CORE PLL

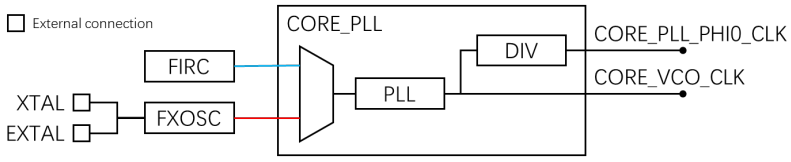


Figure 3. CORE_PLL

The CORE_PLL shown in Figure 3 outputs 1 CORE_VCO_CLK and 1 CORE_PLL_PHI0_CLK. For constraints, please refer to the Data Sheet[ref.[3]ref.[4]].

2.2.2 PERIPH PLL

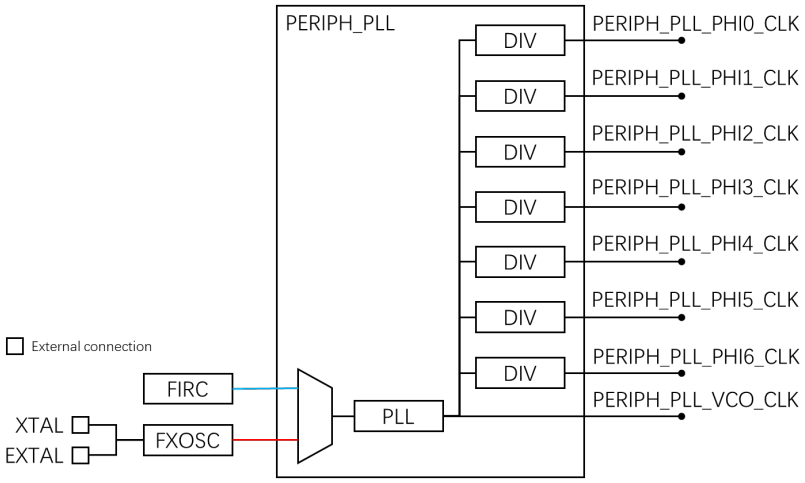


Figure 4. PERIPH_PLL

The PERIPH PLL shown in Figure 4 outputs 1 PERIPH_PLL_VCO_CLK and 7 PERIPH_PLL_PHIn_CLK (0≤n≤6). For constraints, please refer to the Data Sheet[ref.[3]ref.[4]].

2.2.3 DDR PLL

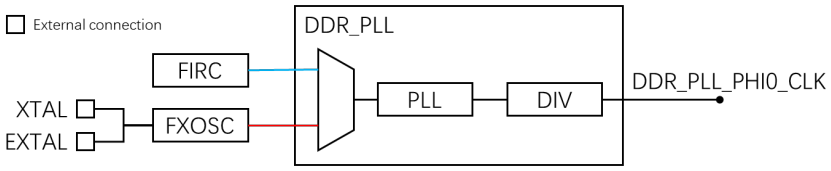


Figure 5. DDR_PLL

The DDR PLL shown in Figure 5 outputs 1 DDR_PLL_PHI0_CLK. For constraints, please refer to the Data Sheet[ref.[3]ref.[4]].

3 DFS

The programmable DFS block generates multiple clock outputs of different frequencies from an input clock source provided by a corresponding PLL. The DFS block provides clock signals as outputs on its ports as per the respective configuration of the ports. S32Z2/E2 supports the following DFSs:

1. CORE_DFS
2. PERIPH_DFS

The data flow of the DFS block is shown in Figure 6.

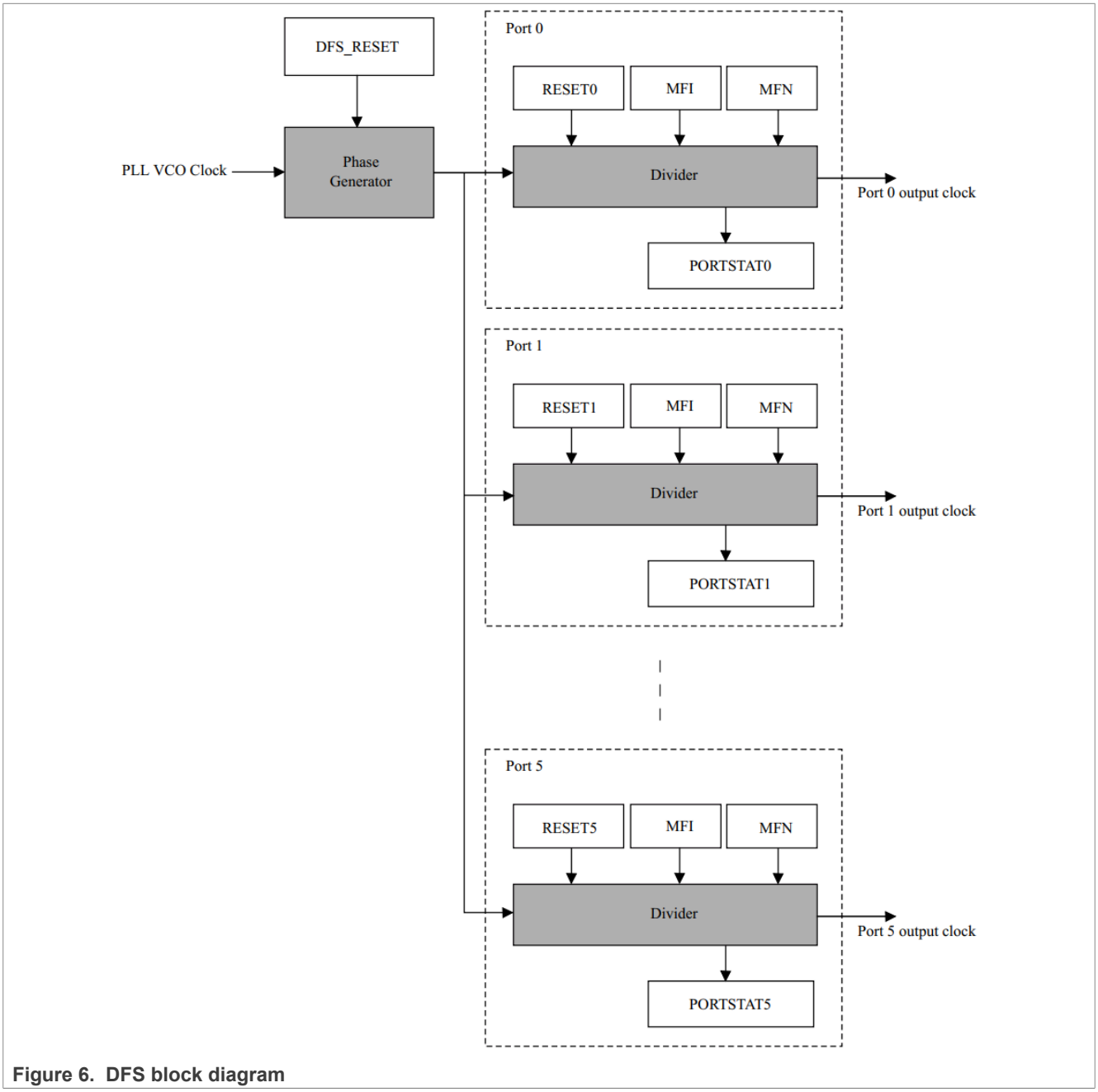


Figure 6. DFS block diagram

The user needs to configure the values for the parameters below to achieve the target frequencies for CORE_DFSn and PERIPH_DFSn (i.e. the Port n output clocks from Figure 6.

1. PLL_VCO: The respective PLL_VCO frequency serves as an input clock source to the DFS block.
2. MFI: Integer Portion of Loop Divider
3. MFN: Numerator of Fractional Loop Division Factor

3.1 DFS Calculation

The DFS block takes an input clock from PLL and generates multiple phases of clock. Six independent phase dividers then use these phases. You control the division factor of each port's divider by programming the DVPORTn[MFI] and DVPORTn[MFN] fields. The following equation describes the relationship between the input and output clock of each phase divider:

$$f_{dfsclkout}n = \frac{f_{dfsclkin}n}{2\times\left(DVPORTn[MFI]+\frac{DVPORTn[MFN]}{36}\right)}$$

3.2 DFS Outputs and Constraints

3.2.1 CORE DFS

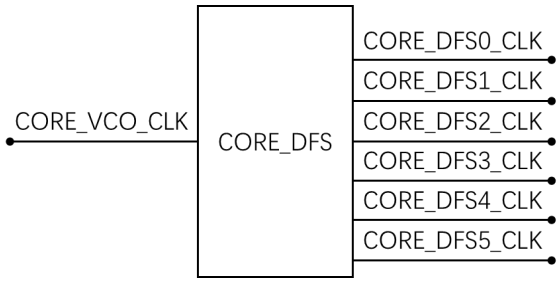


Figure 7. CORE_DFS

The CORE DFS shown in Figure 7 outputs 6 CORE_DFSn_CLK (0≤n≤6) clocks. For constraints, please refer to the Data Sheet[ref.[3]ref.[4]].

3.2.2 PERIPH DFS

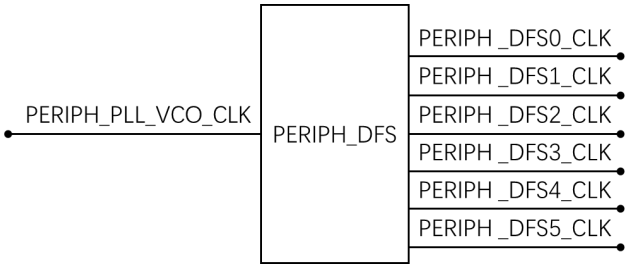


Figure 8. PERIPH_DFS

The PERIPH DFS shown in Figure 8 outputs 6 PERIPH_DFSn_CLK (0≤n≤6) clocks. For constraints, please refer to the Data Sheet[ref.[3]ref.[4]].

4 PLL – AURORA

This section covers the Aurora PLL. The data flow of the AURORA_PLL block is shown in Figure 9 below.

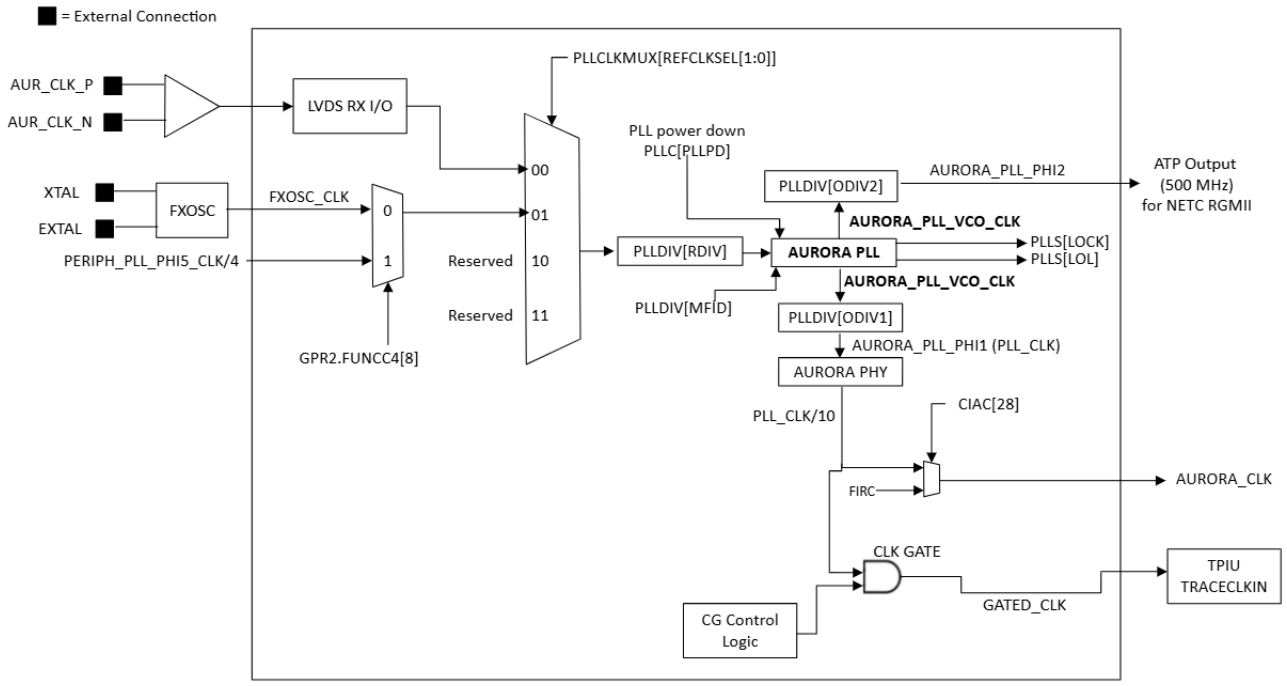


Figure 9. Block Diagram for Aurora PLL

The user needs to configure the values for the below parameters to achieve the target frequencies for AURORA_PLL_VCO_CLK, AURORA_PLL_PHIn, and AURORA_CLK.

1. Reference clock: The clock source for the AURORA_PLL can be the 20MHz – 40MHz FXOSC, the 100MHz external Aurora clock, or PERIPH_PLL_PHI5/4.
2. RDIV: The PLL input reference clock frequency after the pre-divider should be between 40MHz– 100MHz.
3. MFID: Multiplication Factor Integer Divider. This field provides the multiplication factor applied to the reference clock frequency.
4. ODIV: Divider that is applied after VCO.
5. TPIU_CLK_SEL (AURORA_CLK source): The clock source for the AURORA_CLK can be the PLL_CLK/10, or the 48 MHz FIRC.

4.1 AURORA PLL Calculation

When PLLDIV[RDIV] is 0:

$$f_{pll\_VCO} = f_{pll\_ref} \times PLLDIV[MFI] \times 2$$

When PLLDIV[RDIV] is not 0:

$$f_{pll\_VCO} = \frac{f_{pll\_ref}}{PLLDV[RDIV]} \times PLLDIV\left[MFI\right] \times 2$$

4.1.1 AURORA PLL PHI Clock

The fields ODIVn in the PLLDIV register divide the PLL clock into the respective PLL_PHI clocks.

$$f_{pll\_phi} = \frac{f_{pll\_VCO}}{PLLDIV[ODIV\_n]}$$

Note: *PLLDIV[30:25] refers to ODIV2, which maps to AURORA_PLL_PHI2_CLK. This clock is for generating a 500 MHz ATP output for NETC RGMII.*

4.2 AURORA PLL Outputs and Constraints

The AURORA_PLL outputs 1 AURORA_CLK, 1 500 MHz ATP Output for NETC RGMII, and 1 TPIU TRACECLKIN. For constraints, please refer to the Data Sheet[ref.[3]ref.[4]].

5 PLL – LFAST

This section covers the LFAST PLL. There are 2 LFAST PLLs, LFAST0_PLL and LFAST1_PLL. The data flow of the LFAST_PLL block is shown in Figure 10 below.

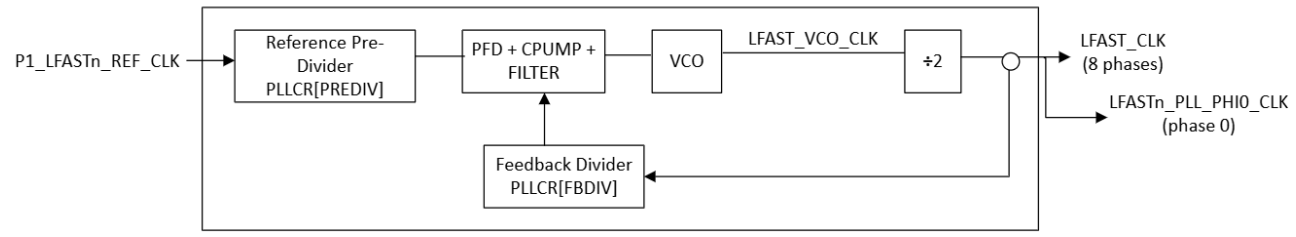


Figure 10. LFAST PLL Block Diagram

The user needs to configure the values for the below parameters to achieve the target frequencies for LFASTn_VCO_CLK and LFASTn_CLK.

1. Reference clock: The clock source for the LFASTn_PLL is P1_LFASTn_ REF_CLK, which is configured in the CGM section.
2. FDIVEN: Enable fraction division mode in feedback divider. Enables the division of VCO output clock by a factor of (FBDIV + 0.5).
3. FBDIV: Feedback Division factor for VCO output clock/2.
4. PREDIV: The PLL input reference clock frequency after the pre-divider should be between 10 – 26 MHz.

5.1 LFAST PLL Calculation

The LFASTn_PLL frequency is calculated with the following formula.

When PLLCR[FDIVEN] = 1:

$$f_{LFAST\_CLK} = \frac{f_{pll\_ref}}{PLLCR[PREDIV]} \times \left(PLLCR[FBDIV] + 0.5 \right)$$

When PLLCR[FDIVEN] = 0:

$$f_{LFAST\_CLK} = \frac{f_{pll\_ref}}{PLLCR[PREDIV]} \times \left(PLLCR[FBDIV] + 1 \right)$$

To calculate the VCO_PLL frequency, use the following equation:

$$f_{pll\_VCO} = f_{LFAST\_PLL} \times 2$$

5.2 LFAST PLL Outputs and Constraints

The LFAST PLL outputs 1 LFASTn_CLK. LFAST0_PLL_PHI0_CLK and LFAST1_PLL_PHI0_CLK correspond to the clock frequency of phase 0 from LFASTn_CLK. For constraints, please refer to the Data Sheet[ref.[3]ref.[4]].

6 CGM

The clock generation module (MC_CGM) configures the clock domains used by various chip blocks as per the application needs. It includes the clock multiplexers that allow software to select the desired clock sources for these domains. This is managed by the MC_CGM to ensure that the changing of the clock selection from one source to another occurs in a glitch-less fashion. In addition, the MC_CGM includes clock dividers that can be configured by software. The CGM data flow is shown in Figure 11.

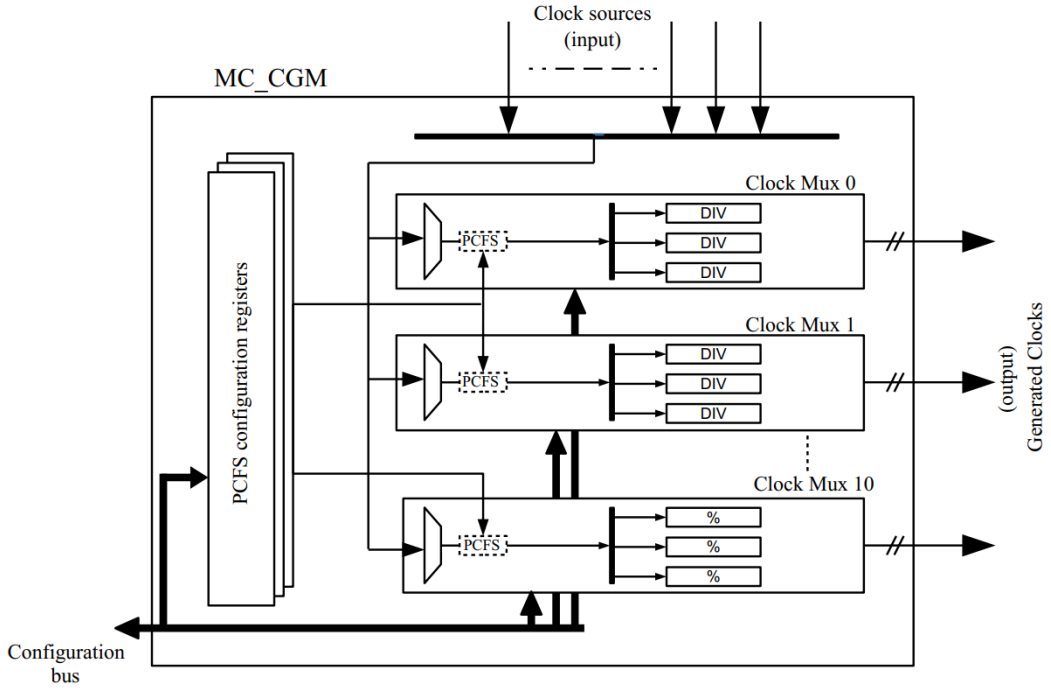


Figure 11. CGM block diagram

The user needs to configure the value for the parameters below to achieve the target frequencies for output clocks generated by MC_CGM.

1. Clock source: Selects the source clock for clock mux N.
2. DIV: Division value.

6.1 CGM calculation

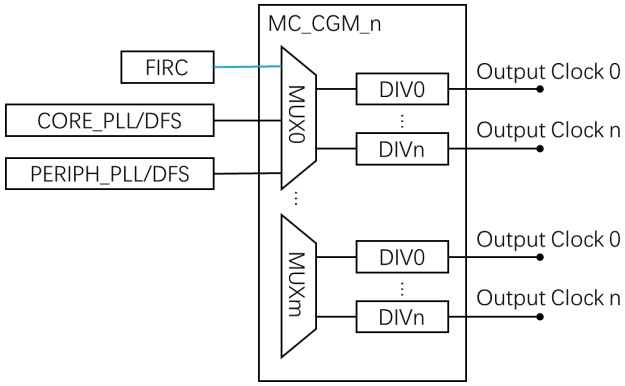


Figure 12. MC_CGM

MC_CGM block takes input clocks from FIRC/PLL/DFS and generates multiple phases of clock, as shown in Figure 12. There are 9 MC_CGM instances . For each MC_CGM instance, there are several multiplexers, and each multiplexer has several dividers. The user selects the input clock source by programming the MUX_m_CSC[SELCTL] field and controls the division factor of each multiplexer's divider by programming the MUX_m_DC_n[DIV] field. The following equation describes the relationship between the input and output clock of each multiplexer's divider:

$$f_{clkout}n = \frac{f_{clkin}}{\text{MUX\_m\_DC\_n[DIV]}}$$

Note: *For diPortSD Module, 2 clock inputs are needed: AE_clock and diPortSD clock transfer, both controlled by CGM_5 MUX5. The maximum of the* $f_{\text{diPortSD clock transfer}}$ *should not exceed 960MHz, and the* $f_{\text{AE\_clock}}$ *should not be less than* $f_{\text{diPortSD clock transfer}}/8$.

7 Clock Gating

7.1 Partition Clock Control

The MC_ME module controls the partition clocks, and there are 3 partitions:

- partition 0 includes SMU, CE cores, CEVA_SPF2 and HSE M7 core;
- partition 1 includes RTU0 cores;
- partition 2 includes RTU1 cores.

Before using any core or core's peripherals, the corresponding partition control should properly configured, as shown in Figure 13.

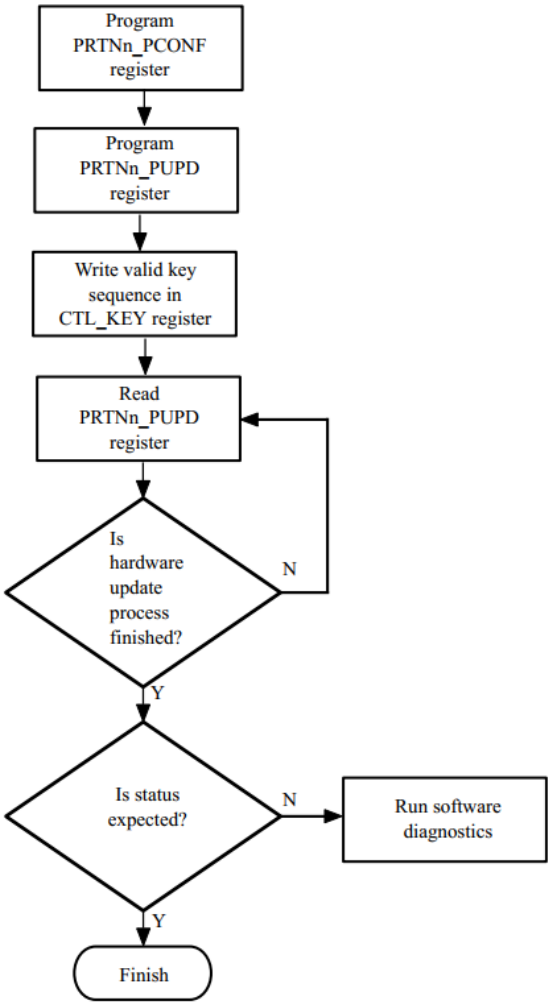


Figure 13. Partition process setup procedure

7.2 Peripheral Clock Gating

GPRn_PCTL registers in the GPRs module are used to control gating of peripheral clocks. The user can disable a module's clock with the GPRn_PCTL registers shown in Table 3 to achieve power saving targets in specific application scenarios.

CAUTION: *When the PCTL register for a module disables the module's clocks, accesses to the module's address space do not return an error and might result in undefined behavior.*

Table 3. Peripheral clock gating group and reset status

Peripheral	Group	Reset Status
SPI_0	GPR0_PCTL	Disabled
FlexRay_0	GPR0_PCTL	Enabled
FlexRay_1	GPR0_PCTL	Enabled
I3C_0	GPR0_PCTL	Enabled
eDMA_0	GPR0_PCTL	Enabled
MSC_0_DSPI	GPR0_PCTL	Disabled

Table 3. Peripheral clock gating group and reset status*...continued*

Peripheral	Group	Reset Status
SPI_1	GPR0_PCTL	Disabled
LINFlex_0	GPR0_PCTL	Enabled
LINFlex_1	GPR0_PCTL	Enabled
LINFlex_2	GPR0_PCTL	Enabled
MSC_0_LIN	GPR0_PCTL	Enabled
ADC_1	GPR0_PCTL	Enabled
ADC_0	GPR0_PCTL	Enabled
PSI5_0	GPR0_PCTL	Enabled
CTU	GPR0_PCTL	Enabled
SIUL_2_0	GPR0_PCTL	Enabled
GTM NANO	GPR0_PCTL	Enabled
PSI5_S_0	GPR0_PCTL	Enabled
SINC	GPR0_PCTL	Enabled
LPI2C_1	GPR1_PCTL	Enabled
eDMA_1	GPR1_PCTL	Enabled
SPI_2	GPR1_PCTL	Disabled
SPI_3	GPR1_PCTL	Disabled
SPI_4	GPR1_PCTL	Disabled
LINFlex_3	GPR1_PCTL	Enabled
LINFlex_4	GPR1_PCTL	Enabled
LINFlex_5	GPR1_PCTL	Enabled
SIPI_0	GPR1_PCTL	Enabled
SIPI_1	GPR1_PCTL	Enabled
SRX_0	GPR1_PCTL	Enabled
ENET_0	GPR1_PCTL	Disabled
SIUL2_1	GPR1_PCTL	Enabled
eDMA_3	GPR3_PCTL	Enabled
CE_eDMA	GPR3_PCTL	Enabled
CE_CAN_0 - CE_CAN_23	GPR3_PCTL	Enabled
CE_PIT_0 - CE_PIT_5	GPR3_PCTL	Enabled
RXLUT	GPR3_PCTL	Enabled
QuadSPI_0	GPR4_PCTL	Enabled
QuadSPI_1	GPR4_PCTL	Enabled
eDMA_4	GPR4_PCTL	Enabled
SPI_5	GPR4_PCTL	Disabled
SPI_6	GPR4_PCTL	Disabled

Table 3. Peripheral clock gating group and reset status*...continued*

Peripheral	Group	Reset Status
SPI_7	GPR4_PCTL	Disabled
LINFlex_6	GPR4_PCTL	Enabled
LINFlex_7	GPR4_PCTL	Enabled
LINFlex_8	GPR4_PCTL	Enabled
uSDHC	GPR4_PCTL	Disabled
SRX_1	GPR4_PCTL	Enabled
LPI2C_2	GPR4_PCTL	Enabled
PSI5_1	GPR4_PCTL	Enabled
SIUL2_4	GPR4_PCTL	Enabled
PSI5_S_1	GPR4_PCTL	Enabled
eDMA_5	GPR5_PCTL	Enabled
SPI_8	GPR5_PCTL	Disabled
SPI_9	GPR5_PCTL	Disabled
LINFlex_9	GPR5_PCTL	Enabled
LINFlex_10	GPR5_PCTL	Enabled
LINFlex_11	GPR5_PCTL	Enabled
SIUL2_5	GPR5_PCTL	Enabled
DDRC	GPR6_PCTL	Disabled

8 Clock Configuration using EB Tresos

EB Tresos[ref.[6]] can be used to configure clock settings. It provides complete configuration items and automatic calculation, validity check, and C code generation. Clock configuration is included in the MCU module and can be added under the tab “McuClockSettingConfig” shown in Figure 14.

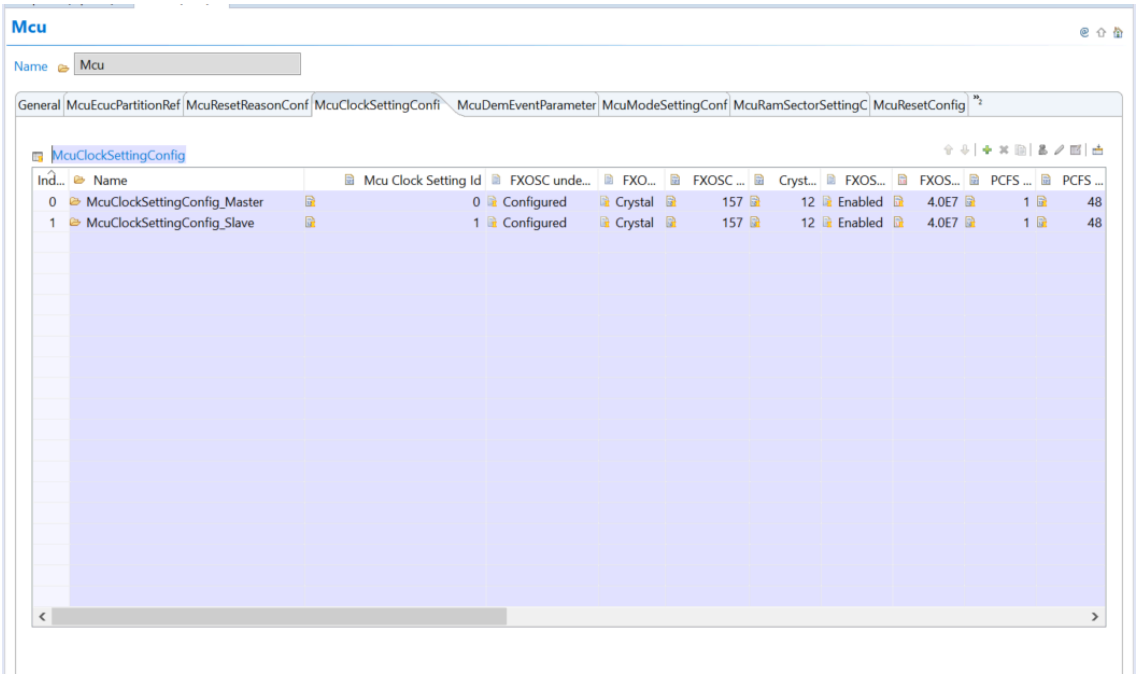


Figure 14. McuClockSettingConfig

For external clock configuration settings, the page layout is shown in Figure 15.

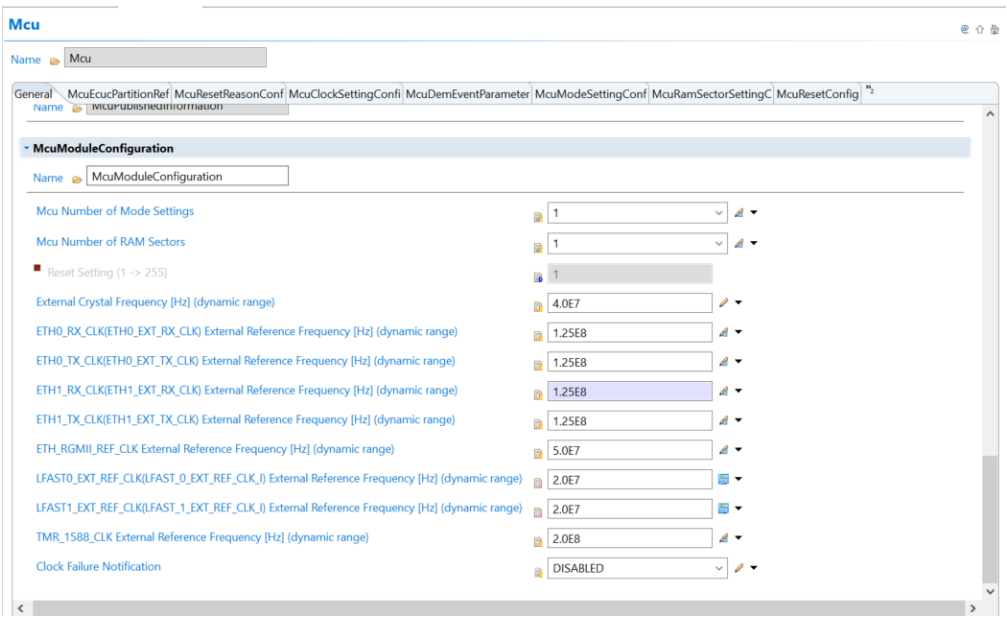


Figure 15. External Clock Settings

For CORE, PERIPH, and DDR PLL settings, the page layout is shown in Figure 16. The user can configure the PLL source (reference clock), RDIV, MFI, MFN, frequency modulation parameters and so on.

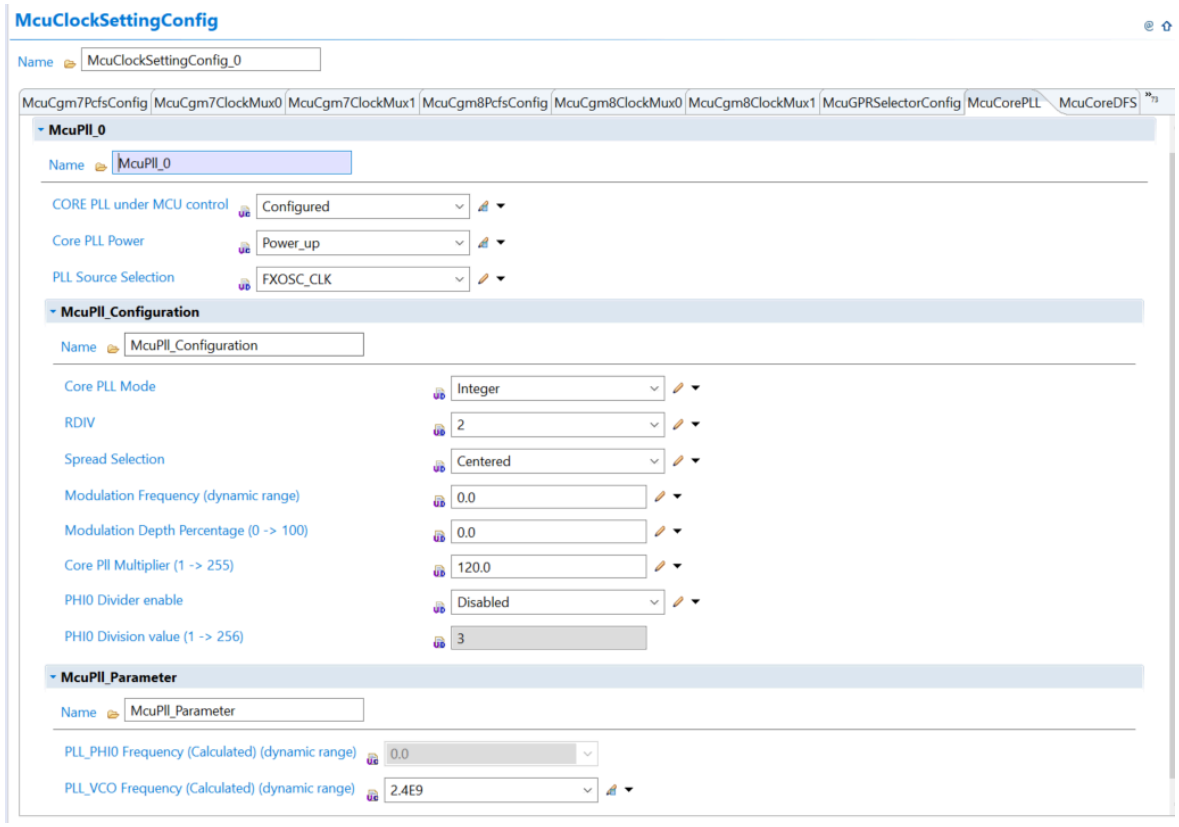


Figure 16. PLL settings

For CORE and PERIPH DFS settings, the page layout is shown in Figure 17. The user can configure DFS enable, MFI, MFN and get the calculated frequency with its validity checked.

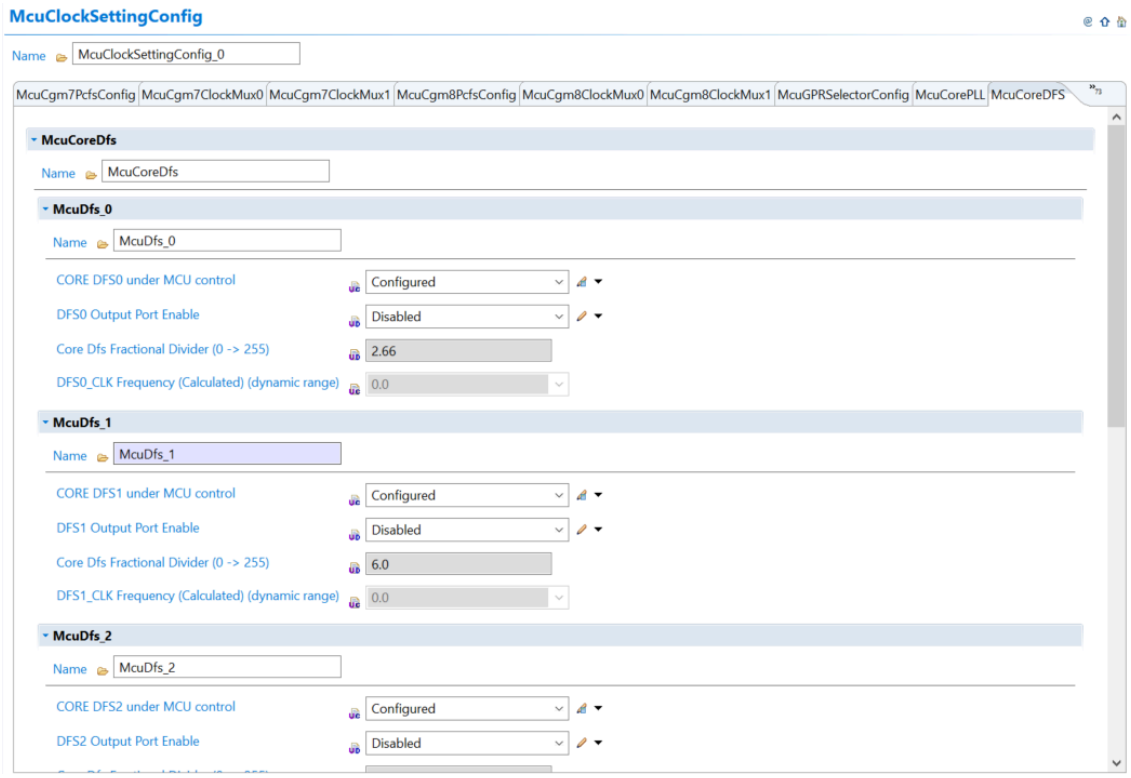


Figure 17. DFS settings

For every CGM multiplexer, there is a page to configure. The page layout is shown in Figure 18. The user can configure the clock source, divider value and obtain the calculated frequency with its validity checked.

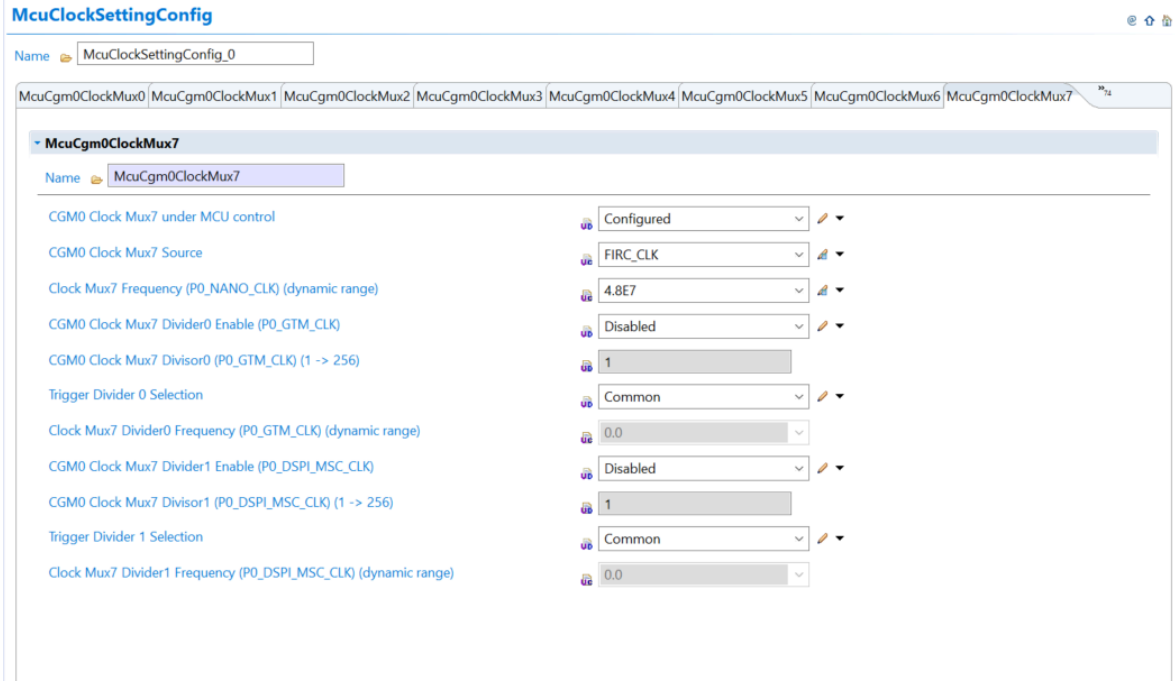


Figure 18. CGM multiplexer settings

For LFASTn_PLL settings, the page layout is shown in Figure 19. The user can configure the PLL mode, PREDIV, the feedback divider (FBDIV), and obtain the calculated frequency with its validity checked.

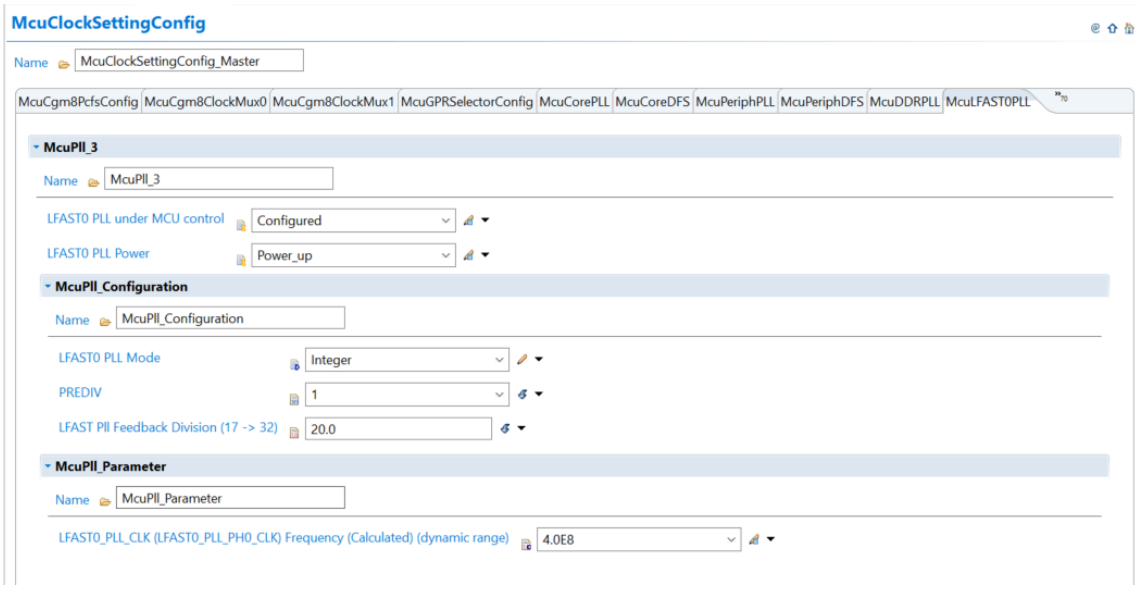


Figure 19. LFAST Settings

Additionally, look in S32DS\software\PlatformSDK_S32ZE\RTD\Mcu_TS_T31D53M20I0R0\doc for “RTD_MCU_UM.pdf” for more information on how to use EB Tresos for clock configuration.

9 Clock Configuration using S32DS Config Tool

The Clock Tool in the S32DS Config Tool[ref.[5]], as shown in Figure 20, provides an intuitive way to configure the clocks. It is formatted in a clock tree view and has a table view option in the upper right corner. The table view may be more helpful than the clock tree view, and the image with the arrow demonstrates how to navigate to it. The user can see what clock source can be used and what output is obtained from the current configuration. The Clock Tool supports validity check and C code generation.

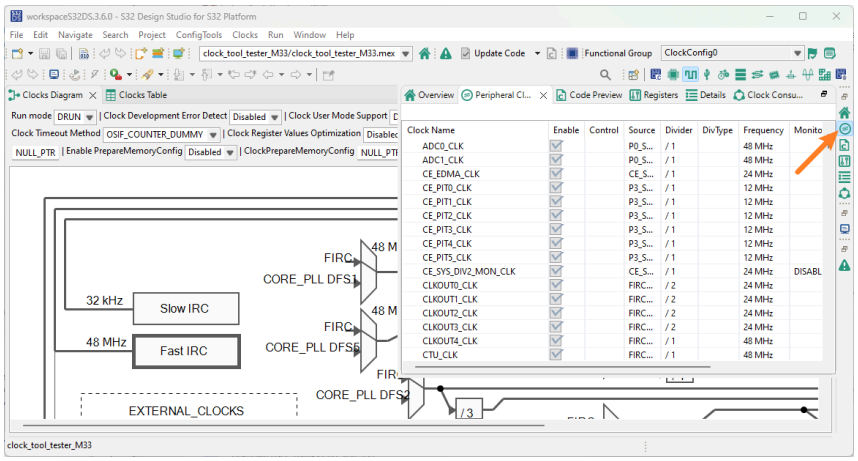


Figure 20. Config Tool clock setting with Table View

For PLL settings, right clicking on the corresponding PLL rectangle and select "Edit the settings of: Core PLL. This will open the Settings interface as shown in Figure 21. The user can configure PLL-related settings here and obtain the calculated frequency with its validity checked.

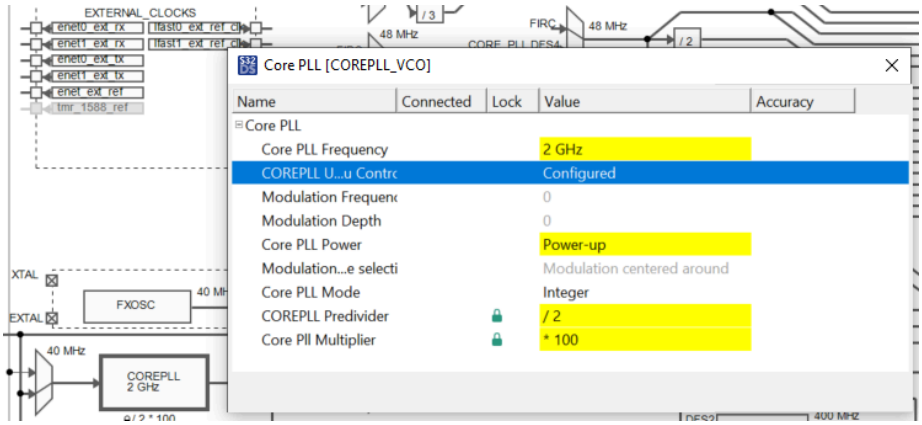


Figure 21. Config Tool - PLL setting

For DFS settings, right clicking on the corresponding DFS rectangle will open the Settings interface as shown in Figure 22. The user can configure DFS enable, divider value (auto-convert to MFI and MFN values) and obtain the calculated frequency with its validity checked.

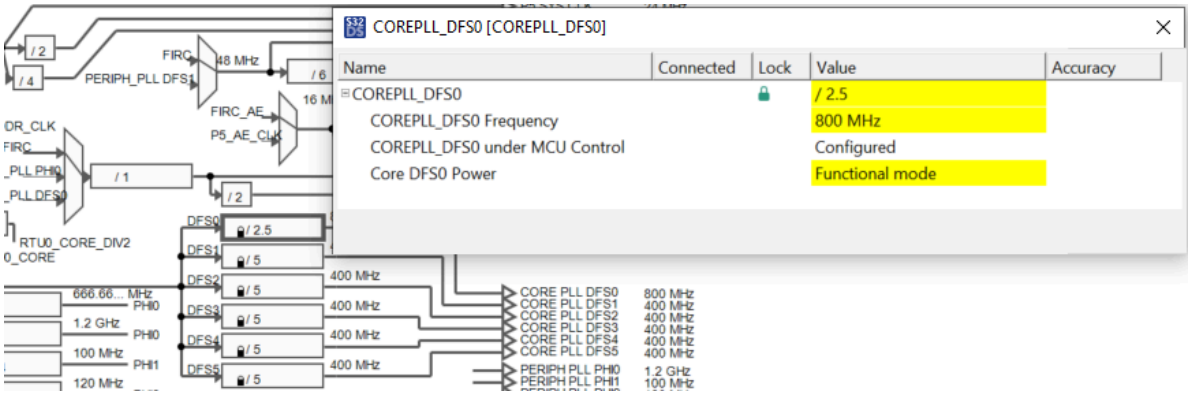


Figure 22. Config Tool – DFS setting

For CGM settings, every CGM multiplexer configuration has a tree part as shown in Figure 23. Right clicking on each part will open the corresponding Settings interface. The trapezoid on the left is for clock source selection, the rectangle in the center is for the divider value, and the square on the right side is the automatic calculated output clock frequency.

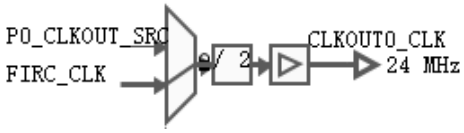


Figure 23. Config Tool – CGM setting

Additionally, look in S32DS\software\PlatformSDK_S32ZE\RTD\Mcu_TS_T31D53M20I0R0\doc for a file called “RTD_MCU_UM.pdf” for more information on how to use the Clocks Tool.

There is a guide within S32DS that describes how to use the Clocks Tool. Navigate to the Help button at the top, select Help Contents. In the new popup window, look for S32 Configuration Tools Getting Started and find Clocks Tool in the dropdown. Refer to Figure 24 for these steps.

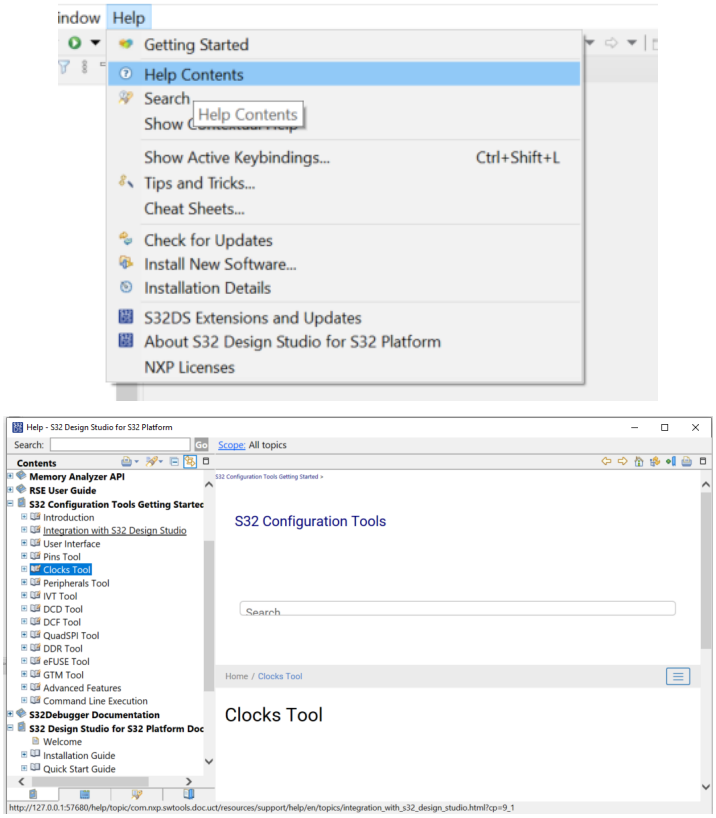


Figure 24. Config Tool – Clocks Tool Help Guide

10 Clock Configuration using S32Z2E2 Clock Configurator

This is an Excel tool that can be used to configure the full clock tree of the S32Z2/E2 processors without installing additional software, such as EB Tresos or S32DS. Users can download multiple copies of this tool, and each copy can be edited independently. Thus, multiple different configurations can be saved. Note that this Excel does not generate source code and can be used standalone or in concert with the other clock configuration tools available for the S32Z2/E2 devices. Configuration should be done sheet-by-sheet following the order shown in this guide to configure the full clock tree. Users may have to click “Enable Content” in the Excel to access drop-down menus.

Find the tool at this link: S32Z2E2_Clock_Configurator.xlsx

10.1 General Notes and Suggestions

- Before using the tool or doing any calculations, review the clocking section of the Reference Manual[ref.[1]ref.[2]] to understand the flow of the clock tree
- This tool’s primary purpose is to validate user inputs and outputs for various clock tree settings. It is not intended to be a solver.
- Calculations should be completed before using this tool.
- Parameters which don't have a drop-down field have to be manually adjusted until the desired output is reached.

- When a cell displays #N/A or #VALUE, this indicates that some formulas used in the spreadsheet have been mistakenly edited. If this happens, download a new copy of the tool and restart.
- Blue cells can be edited (either manually or with a drop-down).
- Cells with manual inputs have input checks to ensure the entered data is valid. Users can click on cells with manual inputs for a “tool tip” with a description of the valid ranges. If the entered value is not valid, an error message will appear and show the valid range and prompt the user to re-enter the value.
- Red and green cell colors show if the cell value is valid.
- Clock source names may have a number in parentheses after the name; this number is for software mapping to registers (see Software Mapping section for more information).

10.2 Use Cases

1. Write custom clock configuration code for bare-metal applications, to configure the full clock tree or a subset of the clock tree. The following sections illustrate the steps needed to configure the full clock tree.
2. Trace and calculate paths in the clock tree for specific clocks without installing software tools such as S32 Design Studio or EB Tresos.
3. A tool for learning about the S32Z2/E2 clock tree.

10.3 Clock Configurator Overview

1. Manual Clocks sheet: select the value of source clocks generated by the device.
2. PLL sheet: select values for PLL parameters MFN, fMOD, and modulation depth (MD) to calculate STEPNO and STEPSIZE for CORE and DDR PLLs. Select ODIV and RDIV values for CORE, PERIPH and DDR PLLs.
3. DFS sheet: select the values of MFI, MFN and MFI/MFN Index to compute the output values for CORE and PERIPH DFS.
4. CLKOUT sheet: select the input sources for P0_CLKOUT_SRC_CLK, P1_CLKOUT_SRC_CLK, P3_CLKOUT_SRC_CLK, P4_CLKOUT_SRC_CLK, and P5_CLKOUT_SRC_CLK.
5. CGM sheet: Select the input source and compute the output for each CGM_MUX.
6. AE Clocking sheet: Select the input source for some AE die modules (only applicable to S32E2 device).
7. Clock Frequency Ranges sheet: Uses inputs from other sheets to compute each value and provides information on whether the computed value is valid.
8. Module Clocks sheet: Computes output values for Module, Register Interface, and other clocks via input clocks.

10.4 Manual Clocks

Clock Name	Frequency (MHz)
FIRC_CLK	48
SIRC_CLK	0.032
FXOSC_CLK	40
ETH_0_RX_CLK	40
ETH_1_RX_CLK	50
ETH_0_TX_CLK	50
ETH_1_TX_CLK	50
ETH_RGMII_REF_CLK	125
TMR_1588_CLK_I	50
LFAST_0_EXT_REF_CLK_I	20
LFAST_1_EXT_REF_CLK_I	20
AURORA_EXT_CLK	100
TCK	50
FIRC_AE	16

Figure 25. Manual Clocks sheet

Edit source clocks that are either clock inputs to the device or generated on die by internal oscillators. In the **Manual Clocks** sheet as shown in Figure 25, edit only the blue highlighted cells. All other clocks without blue highlighting have fixed values that should not be modified. Note that the FIRC and SIRC clocks are generated by the device.

10.5 PLL

Use the **PLL** sheet and follow the steps below to configure this section.

1. Select the clock sources for CORE PLL, PERIPH PLL, DDR PLL, AURORA PLL, and LFAST PLLs using the drop-down list for the REFCLKSEL field, as shown in Figure 26. Do this for each PLL.

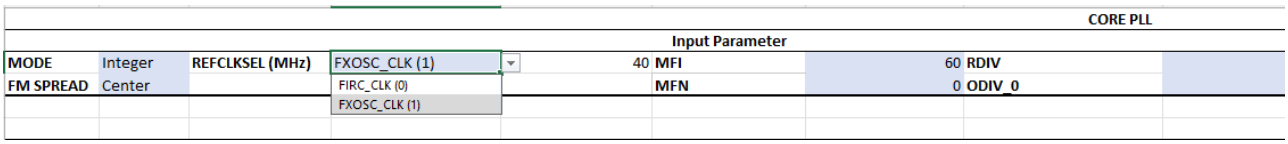


Figure 26. Source Clock Selection with REFCLKSEL field

2. Select the operating modes for CORE, PERIPH and DDR PLLs using the drop-down list for the MODE field for each PLL. For CORE and DDR PLLs, if the selected mode is not FM (Frequency Modulation), the cells in the Min Frequency FM (MHz) and Max Frequency FM (MHz) columns will show “FM NOT ENABLED” since min/ max FM frequencies will not apply.)

3. Enter values for RDIV/ PREDIV as they apply to each PLL. Make note of the table “REFCLKSEL/ RDIV (MHz)” in Figure 27, as the color denotes whether the selected REFCLKSEL and RDIV (or PREDIV, for LFAST0 and LFAST1 PLLs) values are within specification. These ranges for REFCLKSEL/ RDIV (or REFCLKSEL/ PREDIV) are Data Sheet[ref.[3]ref.[4]] specification limits that reflect the valid range of inputs to each PLL.

REFCLKSEL/ RDIV (MHz)					
CORE PLL	PERIPH PLL	DDR PLL	AURORA PLL	LFAST0 PLL	LFAST1 PLL
40	40	40	40	20	40

Figure 27. REFCLKSEL/ RDIV Table

4. Enter values for MFI /MFID/ FBDIV, MFN and ODIV_n as they apply to each PLL (note, FDIVEN for both LFAST PLLs uses a drop-down list as shown in Figure 28, the rest are entered manually).

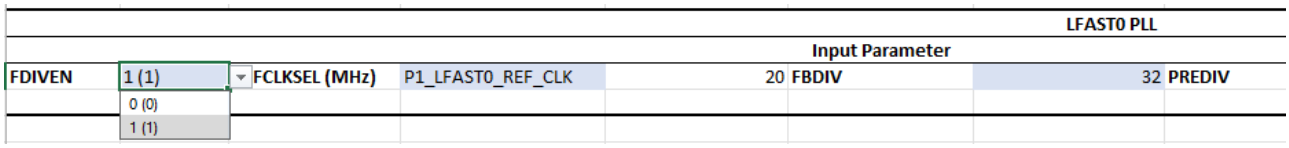


Figure 28. FDIVEN Drop-Down List

5. For tables CORE PLL FM and DDR PLL FM, if utilizing FM for these two PLLs, enter values for fMOD and MD. The calculated STEPNO and STEPSIZE under "Output Parameters" will change colors to indicate valid values. This is also reflected by "FM VALID?", which will change to TRUE if the values provided for the input parameters are valid or FALSE otherwise. Both are seen in Figure 29.

O	P	Q
CORE PLL FM		
Input Parameter		
Description	**Parameter**	**Value**
Expected Modulation Frequency (Hz)	**fMOD (Hz)**	10000
Expected Modulation Depth (%)	**MD (%)**	1.5
Output Parameter		
Description	**Parameter**	**Value**
Loop Division Factor	**LDF**	60.00
Number of Steps of Modulation Period	**STEPNO**	2400.00
Frequency Modulation Step Size	**STEPSIZE**	6.91
Calculated FM Parameters Valid?	**FM VALID?**	TRUE
Maximum Possible Modulation Depth (%)	**Max MD (%)**	1.67

O	P	Q
CORE PLL FM		
Input Parameter		
Description	**Parameter**	**Value**
Expected Modulation Frequency (Hz)	**fMOD (Hz)**	10000
Expected Modulation Depth (%)	**MD (%)**	3
Output Parameter		
Description	**Parameter**	**Value**
Loop Division Factor	**LDF**	60.00
Number of Steps of Modulation Period	**STEPNO**	2400.00
Frequency Modulation Step Size	**STEPSIZE**	13.82
Calculated FM Parameters Valid?	**FM VALID?**	FALSE
Maximum Possible Modulation Depth (%)	**Max MD (%)**	1.67

Figure 29. Output parameters

6. The LFAST PLL uses inputs from the **CGM**. See note at the end of CGM Configuration Steps section for configuring the LFAST PLL.

10.6 DFS

Follow the steps provided below to configure the **DFS** section.

1. Use the dropdowns under "Input Frequency (MHz)" to select input clocks for CORE DFS and PERIPH DFS as shown in Figure 30.
2. Enter values for MFI and MFN for each MFI/MFN Index value.

					CORE DFS	
Input Frequency (MHz)		Input Parameter				
CORE_PLL_VCO_CLK	2400.00	MFI/MFN Index	MFI	MFN	CORE DFS Output	Frequency
CORE_PLL_VCO_CLK		0	1	18	CORE_DFS_0_CLK	
		1	3	0	CORE_DFS_1_CLK	
		2	3	0	CORE_DFS_2_CLK	
		3	2	14	CORE_DFS_3_CLK	
		4	3	0	CORE_DFS_4_CLK	
		5	3	0	CORE_DFS_5_CLK	

Figure 30. Input Clock for DFS

For CORE DFS, the values of the cells in the “Min Frequency FM (MHz)” and “Max Frequency FM (MHz)” columns will depend on if FM mode is selected for CORE PLL on the **PLL** sheet.

10.7 CLKOUT

For each clock under “CLKOUT,” select the desired input clock source as shown in Figure 31 These clocks are used as inputs to some CGM muxes in the **CGM** sheet. Clock source options include CGM output clocks, CORE DFS output clocks, CORE PLL output clocks, and DDR PLL output clocks. The values of the Min Frequency FM (MHz) and Max Frequency FM (MHz) columns depend on if the input clock source selected is derived from a PLL with FM support. “NOT FM CLOCK!” will be displayed if the selected clock is not derived from an FM path. “FM NOT ENABLED” will be displayed if the selected clock is derived from an FM path but FM mode is disabled for the PLL that generated the clock.

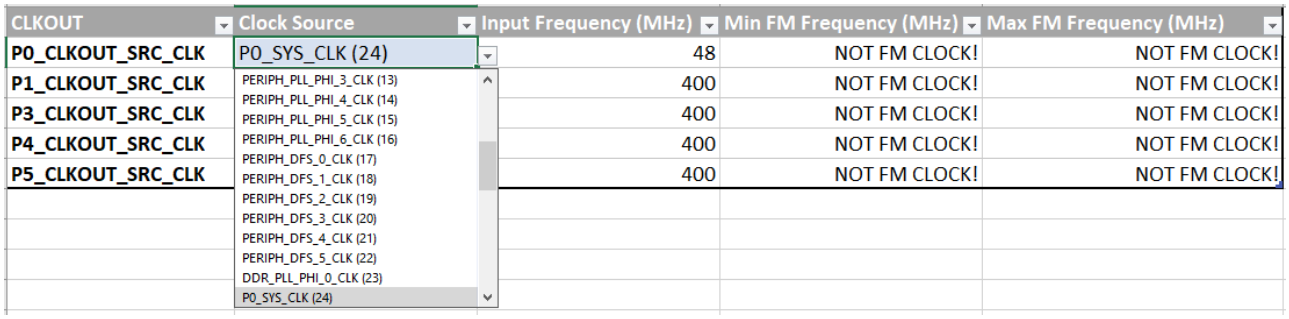


Figure 31. Select Input Clock Source in CLKOUT Sheet

10.8 CGM

10.8.1 Unique CGM Features

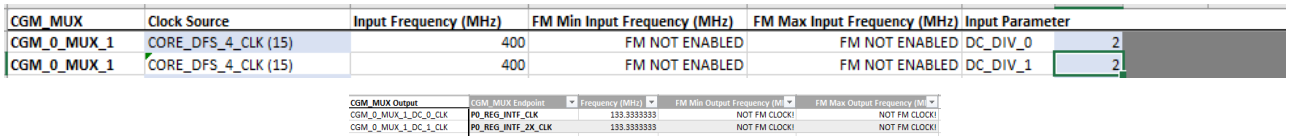
These are unique features of the **CGM** sheet the user should familiarize themselves with prior to following the configuration steps for the CGM section.

1. Notice that some outputs do not have a divider. In these cases, the field DC_DIV_n will be grayed out, as shown in Figure 32.

CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	FM Max Input Frequency (MHz)	Input Parameter	
CGM_0_MUX_0	FIRC_CLK (0)	48	NOT FM CLOCK!	NOT FM CLOCK!		
CGM_0_MUX_1	CORE_DFS_4_CLK (15)	400	FM NOT ENABLED	FM NOT ENABLED	DC_DIV_0	2

Figure 32. CGM_0_MUX_0 has no divider, CGM_0_MUX_1 has 1 (DC_DIV_0)

2. All CGM outputs are unique, but some outputs share a common input. The entries with their clock sources highlighted white have their clock sources updated by the first blue cell above them, as shown in Figure 33.

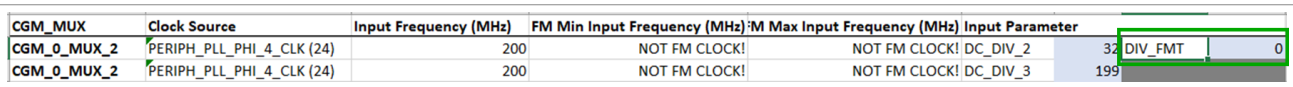


CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	FM Max Input Frequency (MHz)	Input Parameter	
CGM_0_MUX_1	CORE_DFS_4_CLK (15)	400	FM NOT ENABLED	FM NOT ENABLED	DC_DIV_0	2
CGM_0_MUX_1	CORE_DFS_4_CLK (15)	400	FM NOT ENABLED	FM NOT ENABLED	DC_DIV_1	2

CGM_MUX Output	CGM_MUX Endpoint	Frequency (MHz)	FM Min Output Frequency (M	FM Max Output Frequency (M
CGM_0_MUX_1_DC_0_CLK	P0_REG_INTF_CLK	133.3333333	NOT FM CLOCK!	NOT FM CLOCK!
CGM_0_MUX_1_DC_1_CLK	P0_REG_INTF_2X_CLK	133.3333333	NOT FM CLOCK!	NOT FM CLOCK!

Figure 33. CGM_0_MUX_1 leads to outputs CGM_0_MUX_1_DC_0_CLK and CGM_0_MUX_1_DC_1_CLK

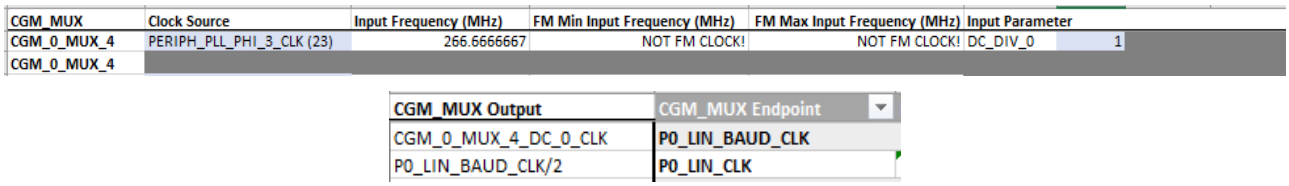
3. Only some CGM muxes include DIV_FMT as an Input Parameter as shown in Figure 34. This field allows for setting the division factor of the clock divider. For more information about this field, refer to the Reference Manual[ref.[1]ref.[2]].



CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	M Max Input Frequency (MHz)	Input Parameter		
CGM_0_MUX_2	PERIPH_PLL_PHI_4_CLK (24)	200	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_2	32	DIV_FMT 0
CGM_0_MUX_2	PERIPH_PLL_PHI_4_CLK (24)	200	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_3	199	

Figure 34. DIV_FMT field

4. Some CGM mux outputs are not directly configurable as shown in Figure 35. In these cases, the entire row corresponding to that output will be grayed out.

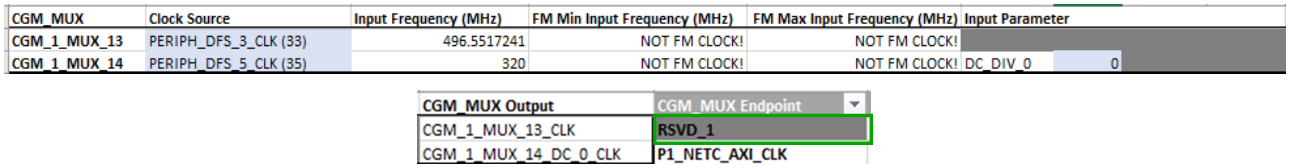


CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	FM Max Input Frequency (MHz)	Input Parameter	
CGM_0_MUX_4	PERIPH_PLL_PHI_3_CLK (23)	266.6666667	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_0	1
CGM_0_MUX_4						

CGM_MUX Output	CGM_MUX Endpoint
CGM_0_MUX_4_DC_0_CLK	P0_LIN_BAUD_CLK
P0_LIN_BAUD_CLK/2	P0_LIN_CLK

Figure 35. CGM mux output P0_LIN_CLK is not directly configurable

5. CGM_MUX Endpoints that are named RSVD_n do not go to any module. However, their Clock Sources and Input Parameters may still be configurable (i.e. not grayed out) as shown in Figure 36.

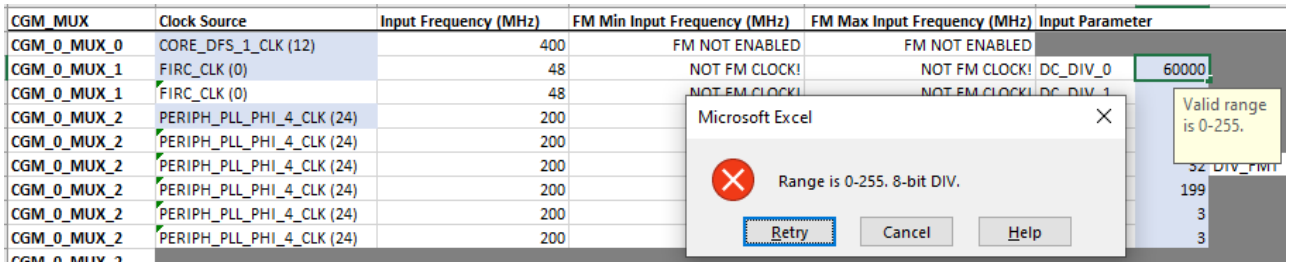


CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	FM Max Input Frequency (MHz)	Input Parameter	
CGM_1_MUX_13	PERIPH_DFS_3_CLK (33)	496.5517241	NOT FM CLOCK!	NOT FM CLOCK!		
CGM_1_MUX_14	PERIPH_DFS_5_CLK (35)	320	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_0	0

CGM_MUX Output	CGM_MUX Endpoint
CGM_1_MUX_13_CLK	RSVD_1
CGM_1_MUX_14_DC_0_CLK	P1_NETC_AXI_CLK

Figure 36. RSVD_1 and RSVD_2 (not shown) do not go to any modules

6. CGM dividers do not all have the same ranges/ limit. There may be more divider ranges possible than shown in Figure 37. Click on the cell corresponding to the CGM divider field that is being configured for a description of the valid range of values.



CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	FM Max Input Frequency (MHz)	Input Parameter	
CGM_0_MUX_0	CORE_DFS_1_CLK (12)	400	FM NOT ENABLED	FM NOT ENABLED		
CGM_0_MUX_1	FIRC_CLK (0)	48	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_0	60000
CGM_0_MUX_1	FIRC_CLK (0)	48	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_1	
CGM_0_MUX_2	PERIPH_PLL_PHI_4_CLK (24)	200				
CGM_0_MUX_2	PERIPH_PLL_PHI_4_CLK (24)	200				
CGM_0_MUX_2	PERIPH_PLL_PHI_4_CLK (24)	200				
CGM_0_MUX_2	PERIPH_PLL_PHI_4_CLK (24)	200				199
CGM_0_MUX_2	PERIPH_PLL_PHI_4_CLK (24)	200				3
CGM_0_MUX_2	PERIPH_PLL_PHI_4_CLK (24)	200				3

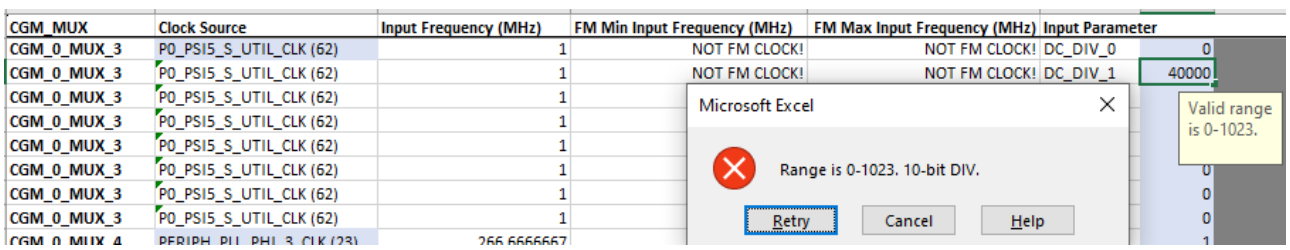


Figure 37. CCGM_0_MUX_2 has a DC_DIV_0 which only has the range 0-255, CMG_0_MUX_3 has a DC_DIV_1 with a range of 0-1023.

7. DC_DIV_nPS dividers cannot be configured separately and are highlighted white as shown in Figure 38. These values are updated by the blue DC_DIV_n field directly above it. (Note: PS refers to “phase-shifted.”)

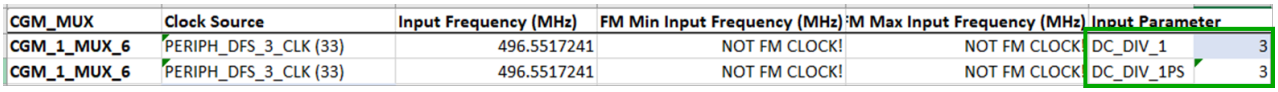


Figure 38. DC_DIV_1PS is updated by DC_DIV_1 directly above it

10.8.2 CGM Configuration Steps

Follow the steps provided below for configuring the CGM section to achieve max clock frequencies.

1. For each mux under "CGM_MUX," use the drop-down to select a clock source (only the blue cells have drop-down menus) as shown in Figure 39. This will update the "Input Frequency (MHz)" entry, and the entries for FM Min Input Frequency (MHz) and FM Max Input Frequency (MHz). The FM min and max input columns are updated depending if the input clock source selected is derived from a PLL with FM support.

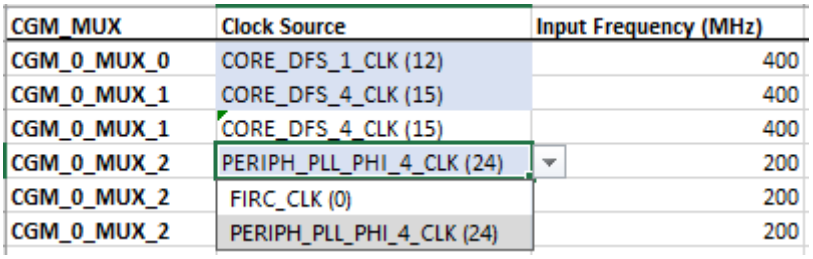
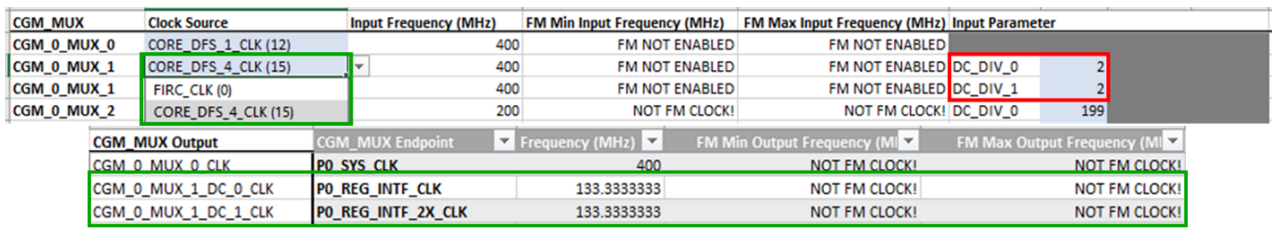


Figure 39. Drop Down to Select Clock Source for CGM_MUX

2. Configure the dividers (DC_DIV_0, DC_DIV_1, etc.) and “DIV_FMT” fields under “Input Parameter” as applicable for each CGM mux with an output divider. Changes to fields under “Input Parameter” update the output clocks under "CGM_MUX Endpoint" that correspond to each CGM mux, including the corresponding values under FM Min Output Frequency (MHz) and FM Max Output Frequency (MHz), depending on if the FM min and max input columns have text or numerical values.

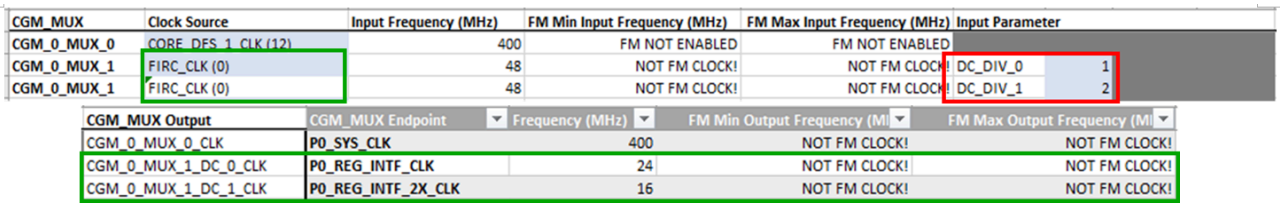
Refer to Figure 40 and Figure 41 for a complete example that shows the result of all the CGM configuration steps. Notice the CGM_MUX entries that repeat if they feed into multiple output clocks. The repeated entries' clock source will be updated by making a selection with the drop down in the blue cell, and any corresponding input parameters will also be updated by modifying the value in a corresponding blue cell.



CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	FM Max Input Frequency (MHz)	Input Parameter	
CGM_0_MUX_0	CORE_DFS_1_CLK (12)	400	FM NOT ENABLED	FM NOT ENABLED		
CGM_0_MUX_1	CORE_DFS_4_CLK (15)	400	FM NOT ENABLED	FM NOT ENABLED	DC_DIV_0	2
CGM_0_MUX_1	FIRC_CLK (0)	400	FM NOT ENABLED	FM NOT ENABLED	DC_DIV_1	2
CGM_0_MUX_2	CORE_DFS_4_CLK (15)	200	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_0	199

CGM_MUX Output	CGM_MUX Endpoint	Frequency (MHz)	FM Min Output Frequency (MI	FM Max Output Frequency (MI
CGM_0_MUX_0_CLK	P0_SYS_CLK	400	NOT FM CLOCK!	NOT FM CLOCK!
CGM_0_MUX_1_DC_0_CLK	P0_REG_INTF_CLK	133.3333333	NOT FM CLOCK!	NOT FM CLOCK!
CGM_0_MUX_1_DC_1_CLK	P0_REG_INTF_2X_CLK	133.3333333	NOT FM CLOCK!	NOT FM CLOCK!

Figure 40. CGM_0_MUX_1 entries repeat, with each entry having a different divider. Source clock is CORE_DFS_4_CLK. P0_REG_INTF_CLK has DC_DIV_0 divider and P0_REG_INTF_2X_CLK has DC_DIV_1 divider.

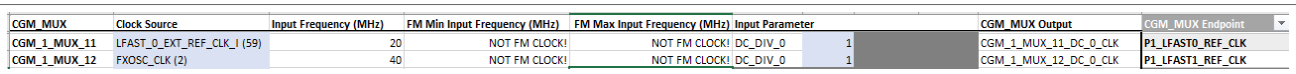


CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	FM Max Input Frequency (MHz)	Input Parameter	
CGM_0_MUX_0	CORE_DFS_1_CLK (12)	400	FM NOT ENABLED	FM NOT ENABLED		
CGM_0_MUX_1	FIRC_CLK (0)	48	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_0	1
CGM_0_MUX_1	FIRC_CLK (0)	48	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_1	2

CGM_MUX Output	CGM_MUX Endpoint	Frequency (MHz)	FM Min Output Frequency (MI	FM Max Output Frequency (MI
CGM_0_MUX_0_CLK	P0_SYS_CLK	400	NOT FM CLOCK!	NOT FM CLOCK!
CGM_0_MUX_1_DC_0_CLK	P0_REG_INTF_CLK	24	NOT FM CLOCK!	NOT FM CLOCK!
CGM_0_MUX_1_DC_1_CLK	P0_REG_INTF_2X_CLK	16	NOT FM CLOCK!	NOT FM CLOCK!

Figure 41. Changing Clock Source to FIRC_CLK updates P0_REG_INTF_CLK and P0_REG_INTF_2X_CLK in CGM_MUX Endpoint Column. Changing DC_DIV_0 causes additional changes to P0_REG_INTF_CLK value.

Note: *To configure LFAST0 and LFAST1 PLLs, look at CGM_1_MUX_11 for LFAST0 and CGM_1_MU_11 for LFAST1 as shown in Figure 42. Select source clocks for each one and configure the divider value for each, which will affect the values for P1_LFAST0_REF_CLK and P1_LFAST1_REF_CLK. These clocks are source clocks for LFAST0 PLL and LFAST1 PLL, respectively. Go back to PLL and configure the LFAST PLLs according to the steps in that section.*

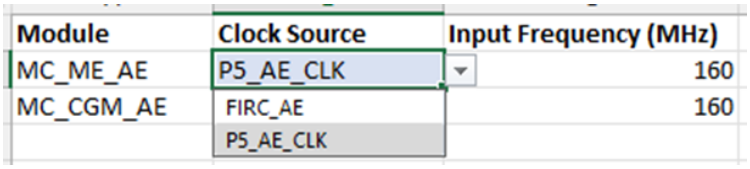


CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	FM Max Input Frequency (MHz)	Input Parameter		CGM_MUX Output	CGM_MUX Endpoint
CGM_1_MUX_11	LFAST_0_EXT_REF_CLK_I (59)	20	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_0	1	CGM_1_MUX_11_DC_0_CLK	P1_LFAST0_REF_CLK
CGM_1_MUX_12	FXOSC_CLK (2)	40	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_0	1	CGM_1_MUX_12_DC_0_CLK	P1_LFAST1_REF_CLK

Figure 42. LFAST0 and LFAST 1 PLL Reference Clock Mux Configuration

10.9 AE Clocking

Use the drop-down to select a clock source for MC_ME_AE. This updates the Clock Source and Input Frequency values for both MC_ME_AE and MC_CGM_AE.



Module	Clock Source	Input Frequency (MHz)
MC_ME_AE	P5_AE_CLK	160
MC_CGM_AE	FIRC_AE	160
	P5_AE_CLK	

Figure 43. MC_ME_AE and MC_CGM_AE clock configuration

10.10 Clock Frequency Ranges

The **Clock Frequency Ranges** sheet, as shown in Figure 44, summarizes the calculated output frequencies from the prior sheets and validates them against the data sheet min/max. Here you will see descriptions, names, and calculated values of all clocks based on the inputs provided in the previous steps. Calculated values will change color if they are within the correct range for that clock, including their corresponding FM min and max frequencies. Check this sheet to ensure the configured clock settings are valid.

Clock Description	Clock Name	DS Min	FM Min Frequency	Calculated Frequency	FM Max Frequency	DS Max	Unit
CTU Clock Frequency	P0_CTU_PER_CLK	0	NOT FM CLOCK!	4	NOT FM CLOCK!	81	MHz
SPI_0 and SPI_1 Clock Frequency	P0_DSPI_CLK	0	NOT FM CLOCK!	5	NOT FM CLOCK!	102	MHz
Micro Second Channel Clock Frequency	P0_DSPI_MSC_CLK	0	NOT FM CLOCK!	0.17	NOT FM CLOCK!	135	MHz
eMIOS_0 Clock Frequency	P0_EMIOS_LCU_CLK	0	NOT FM CLOCK!	8	NOT FM CLOCK!	162	MHz
FlexRay Protocol Engine Clock Frequency	P0_FR_PE_CLK	0	NOT FM CLOCK!	4	NOT FM CLOCK!	81	MHz
GTM Clock Frequency	P0_GTM_CLK	0	NOT FM CLOCK!	0.51	NOT FM CLOCK!	405	MHz
GTM NoC Interface Clock Frequency	P0_GTM_NOC_CLK	0	NOT FM CLOCK!	0.51	NOT FM CLOCK!	405	MHz
GTM Time Stamp Clock Frequency	P0_GTM_TS_CLK	0	NOT FM CLOCK!	0.1	NOT FM CLOCK!	81	MHz
LINFlexD Baud Clock Frequency	P0_LIN_BAUD_CLK	0	NOT FM CLOCK!	6.67	NOT FM CLOCK!	135	MHz
LINFlexD Clock Frequency	P0_LIN_CLK	0	NOT FM CLOCK!	3.33	NOT FM CLOCK!	67.5	MHz
Digital NanoEdge Clock Frequency	P0_NANO_CLK	0	NOT FM CLOCK!	1.52	NOT FM CLOCK!	1620	MHz
PSI5 125K Clock Frequency	P0_PSI5_125K_CLK	0	NOT FM CLOCK!	0.2	NOT FM CLOCK!	4.05	MHz
PSI5 189K Clock Frequency	P0_PSI5_189K_CLK	0	NOT FM CLOCK!	0.3	NOT FM CLOCK!	6.0606	MHz
PSI5 1US Clock Frequency	P0_PSI5_1US_CLK	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
PSI5_S Baud Clock Frequency	P0_PSI5_S_BAUD_CLK	0	NOT FM CLOCK!	2.5	NOT FM CLOCK!	50.625	MHz
PSI5_S Core Clock Frequency	P0_PSI5_S_CORE_CLK	0	NOT FM CLOCK!	1.25	NOT FM CLOCK!	25.3	MHz
PSI5_S_0 Trigger Clock 0 Frequency	P0_PSI5_S_TRIG_CLK0	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
PSI5_S_0 Trigger Clock 1 Frequency	P0_PSI5_S_TRIG_CLK1	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
PSI5_S_0 Trigger Clock 2 Frequency	P0_PSI5_S_TRIG_CLK2	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
PSI5_S_0 Trigger Clock 3 Frequency	P0_PSI5_S_TRIG_CLK3	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
PSI5_S UART Clock Frequency	P0_PSI5_S_UART_CLK	0	NOT FM CLOCK!	2.5	NOT FM CLOCK!	50.625	MHz
PSI5_S UTIL Clock Frequency	P0_PSI5_S_UTIL_CLK	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
PSI5_S_0 Watchdog Clock 0 Frequency	P0_PSI5_S_WDOG_CLK0	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
PSI5_S_0 Watchdog Clock 1 Frequency	P0_PSI5_S_WDOG_CLK1	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
PSI5_S_0 Watchdog Clock 2 Frequency	P0_PSI5_S_WDOG_CLK2	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
PSI5_S_0 Watchdog Clock 3 Frequency	P0_PSI5_S_WDOG_CLK3	0	NOT FM CLOCK!	0.05	NOT FM CLOCK!	1.01	MHz
P0 Register Interface 2X Clock Frequency	P0_REG_INTF_2X_CLK	0	2.909406566	2.91	2.909406566	133.33	MHz
P0 Register Interface Clock Frequency	P0_REG_INTF_CLK	0	2.909406566	2.91	2.909406566	133.33	MHz
P0 System Clock Frequency	P0_SYS_CLK	0	8.000868056	8	8.000868056	405	MHz
SPI_3 and SPI_4 Clock Frequency	P1_DSPI60_CLK	0	NOT FM CLOCK!	6	NOT FM CLOCK!	120	MHz
SPI_2 Clock Frequency	P1_DSPI_CLK	0	NOT FM CLOCK!	5	NOT FM CLOCK!	102	MHz
LINFlexD Baud Clock Frequency	P1_LIN_BAUD_CLK	0	NOT FM CLOCK!	6.67	NOT FM CLOCK!	135	MHz
LINFlexD Clock Frequency	P1_LIN_CLK	0	NOT FM CLOCK!	3.33	NOT FM CLOCK!	67.5	MHz
ETH_0 RMII Mode Reference Clock Frequency (P1_NETC0_REF_RMII_CLK)	ETH_0_REF_RMII_CLK	0	NOT FM CLOCK!	2.48	NOT FM CLOCK!	50	MHz
ETH_0 MII Mode Receive Clock Frequency (P1_NETC0_RX_MII_CLK)	ETH_0_RX_MII_CLK	0	NOT FM CLOCK!	1.18	NOT FM CLOCK!	25	MHz
ETH_0 RGMII Mode Receive Clock Frequency (P1_NETC0_RX_RGMII_CLK)	ETH_0_RX_RGMII_CLK	0	NOT FM CLOCK!	6.21	NOT FM CLOCK!	125.75	MHz
ETH_0 MII Mode Transmit Clock Frequency (P1_NETC0_TX_MII_CLK)	ETH_0_TX_MII_CLK	0	0.478457226	0.48	0.478457226	25	MHz
ETH_0 RGMII Mode Transmit Clock Frequency (P1_NETC0_TX_RGMII_CLK)	ETH_0_TX_RGMII_CLK	0	2.511900436	2.51	2.511900436	125.75	MHz
ETH_1 RMII Mode Reference Clock Frequency (P1_NETC1_REF_RMII_CLK)	ETH_1_REF_RMII_CLK	0	NOT FM CLOCK!	2.48	NOT FM CLOCK!	50	MHz
ETH_1 MII Mode Receive Clock Frequency (P1_NETC1_RX_MII_CLK)	ETH_1_RX_MII_CLK	0	NOT FM CLOCK!	1.18	NOT FM CLOCK!	25	MHz
ETH_1 RGMII Mode Receive Clock Frequency (P1_NETC1_RX_RGMII_CLK)	ETH_1_RX_RGMII_CLK	0	NOT FM CLOCK!	6.21	NOT FM CLOCK!	125.75	MHz
ETH_1 MII Mode Transmit Clock Frequency (P1_NETC1_TX_MII_CLK)	ETH_1_TX_MII_CLK	0	0.478457226	0.48	0.478457226	25	MHz

Figure 44. Clock Frequency Ranges Sheet

10.11 Module Clocks

The **Module Clocks** sheet shows a summary of the clock tree distributed to all the modules on the device. The following describes different features of the sheet and how to use it.

1. To only see the clocks for a specific module, use the “+” and “-” symbols on the left side of the sheet to expand and collapse each section.

2. The frequency modulated minimum and maximum values (if applicable) are not shown in this sheet since any source clocks in this sheet can be checked in the **Clock Frequency Ranges** sheet to make sure they are in the correct range. The values shown here are nominal clock frequencies for both output clocks that have FM source clocks and output clocks that do not.

3. Some modules have more “Other Clocks” than others or may not have a “Module Clock” or “Reg Interface Clock.” As a result, some of the cells corresponding to these clocks’ frequency values may have values of “---” (and not #N/A or #VALUE!) in the frequency columns if there are no clocks, as shown in Figure 45. This is normal functionality and not a bug.

	Group	Module	Module Clock Source	Module Clock Frequency (MHz)	Reg Interface Clock Source	Reg Interface Clock Frequency (MHz)
2	RTU Modules					
53	SMU Modules					
74	System Modules					
136	Reset and Boot Modules					
140	Power Management Modules					
143	Memory-Controller Modules					
144		QuadSPI_0	P4_REG_INTF_2X_CLK	133.3333333	P4_REG_INTF_2X_CLK	133.3333333
145		QuadSPI_1	P4_REG_INTF_2X_CLK	133.3333333	P4_REG_INTF_2X_CLK	133.3333333
146		OTFAD		---	P4_REG_INTF_2X_CLK	133.3333333
147		MEW_0	P4_REG_INTF_2X_CLK	133.3333333	P4_REG_INTF_2X_CLK	133.3333333
148		MEW_1	P4_REG_INTF_2X_CLK	133.3333333	P4_REG_INTF_2X_CLK	133.3333333
149		uSDHC	P4_SDHC_IP_DIV2_CLK	133.3333333		---
150		DDR		---	P6_REG_INTF_CLK	133.3333333
151	Timer Modules w/ GTM Subsystem Modules					
167	Communication Interface Modules					
213	FlexLLCE Modules					
319	Analog Modules					
323	Motor Control Modules					
324		CTU	P0_REG_INTF_CLK	24	P0_REG_INTF_CLK	24
325		LCU_0	P4_EMIOS_LCU_CLK	160	P4_REG_INTF_CLK	133.3333333
326		LCU_1	P0_EMIOS_LCU_CLK	160	P0_REG_INTF_CLK	24
327		TRGMUX_0	P4_REG_INTF_CLK	133.3333333	P4_REG_INTF_CLK	133.3333333
328		TRGMUX_1	P0_REG_INTF_CLK	24	P0_REG_INTF_CLK	24

Figure 45. Module Clocks Sheet (Scroll right to find “Other Clock” entries)

10.12 Examples

This section presents a few examples of common usage scenarios for the tool.

10.12.1 Configure a Clock for a specific module

One usage scenario for the tool is as an aid to calculate a specific module clock. This example will show the steps to trace and configure the CE_CAN_0 CAN clock source to 16MHz.

Note: *CE_CAN_0 CAN clock source is P3_CAN_PE_CLK.*

1. Go to **Module Clocks** sheet.
   a. Find “FlexLLCE Modules” dropdown.
   b. Find CE_CAN_0 and identify its CANCLK source as shown in Figure 46.

Module	Module Clock Source	Module Clock Frequency (MHz)	Reg Interface Clock Source	Reg Interface Clock Frequency (MHz)	Other Clock 1 Name	Other Clock 1 Source	Other Clock 1 Frequency (MHz)
CE_CAN_0	CE_SYS_DIV2_CLK	200	CE_SYS_DIV2_CLK	200	CANCLK	P3_CAN_PE_CLK	80

Figure 46. Find CE_CAN_0

2. Go to the **CGM** sheet. Refer to Figure 47 for the following steps.
   a. Find P3_CAN_PE_CLK under “CGM_MUX_Endpoint.”
   b. In that same row under “Clock Source” select “PERIPH_PLL_PHI_5_CLK.”
   c. In the same row, set DC_DIV_0 to 5.

CGM_MUX	Clock Source	Input Frequency (MHz)	FM Min Input Frequency (MHz)	FM Max Input Frequency (MHz)	Input Parameter		CGM_MUX Output	CGM_MUX Endpoint
CGM_3_MUX_3	PERIPH_PLL_PHI_5_CLK (25)	96	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_0	5	CGM_3_MUX_3_DC_0_CLK	P3_CAN_PE_CLK
CGM_3_MUX_4	FIRC_CLK (0)	400	NOT FM CLOCK!	NOT FM CLOCK!	DC_DIV_0	9	CGM_3_MUX_4_DC_0_CLK	P3_CLKOUT_4_CLK
CGM_4_MUX_0	FXOSC_CLK (2)	400	FM NOT ENABLED	FM NOT ENABLED			CGM_4_MUX_0_CLK	P4_SYS_CLK
CGM_4_MUX_0	PERIPH_PLL_PHI_5_CLK (25)						P4_SYS_CLK/2	P4_SYS_DIV2_CLK

Figure 47. Configure CGM_3_MUX_3 in CGM Sheet

3. Go to the **PLL** sheet. See Figure 48 for the steps listed.
   a. Under “PERIPH PLL,” select MODE to be “Integer”
   b. For REFCLKSEL select “FXOSC_CLK.”
   c. Set MFI to 60, MFN to 0, and RDIV to 1.
   d. Make note of the value in and the color of the PERIPH PLL cell under “REFCLKSEL/ RDIV (MHz).” For this PLL, a valid input, computed by REFCLKSEL/ RDIV, can only be 20-40 MHz, and this cell will be green if the computed input value falls in this valid range.
   e. Set ODIV_5 to 24.

PERIPH PLL

					Input Parameter			PERIPH PLL Output	Frequency (MHz)
MODE	Integer	REFCLKSEL (MHz)	FXOSC_CLK (1)	40	MFI	60	RDIV	1	PERIPH_PLL_VCO_CLK	2400.00
					MFN	0	ODIV_0	1	PERIPH_PLL_PHI_0_CLK	1200.00
							ODIV_1	23	PERIPH_PLL_PHI_1_CLK	100.00
							ODIV_2	19	PERIPH_PLL_PHI_2_CLK	120.00
							ODIV_3	8	PERIPH_PLL_PHI_3_CLK	266.67
							ODIV_4	11	PERIPH_PLL_PHI_4_CLK	200.00
							ODIV_5	24	PERIPH_PLL_PHI_5_CLK	96.00
							ODIV_6	2	PERIPH_PLL_PHI_6_CLK	800.00

REFCLKSEL/ RDIV (MHz)				
CORE PLL	PERIPH PLL	DDR PLL	AURORA PLL	LFAST PLL
40	40	40	40	20

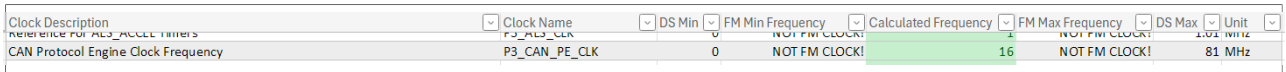
Figure 48. Configure each PERIPH PLL field and REFCLKSEL/ RDIV Validity

4. Check the **Manual Clocks** sheet. Ensure FXOSC CLK is 40MHz, as shown in Figure 49.

Clock Name	Frequency (MHz)
FIRC_CLK	48
SIRC_CLK	0.032
FXOSC_CLK	40
ETH_0_RX_CLK	40
ETH_1_RX_CLK	50
ETH_0_TX_CLK	50
ETH_1_TX_CLK	50
ETH_RGMII_REF_CLK	125
TMR_1588_CLK_I	50
LFAST_0_EXT_REF_CLK_I	20
LFAST_1_EXT_REF_CLK_I	20
AURORA_EXT_CLK	100
TCK	50

Figure 49. FXOSC_CLK in Manual Clocks

5. Go to the **Clock Frequency Ranges** sheet and look for P3_CAN_PE_CLK. The calculated frequency is 16MHz as desired and is highlighted green and is within the acceptable range of values, as shown in Figure 50.



Clock Description	Clock Name	DS Min	FM Min Frequency	Calculated Frequency	FM Max Frequency	DS Max	Unit
CAN Protocol Engine Clock Frequency	P3_CAN_PE_CLK	0	NOT FM CLOCK!	16	NOT FM CLOCK!	81	MHz

Figure 50. Check P3_CAN_PE_CLK Value

10.12.2 Configure the full Clock Tree

For this use case, follow the configuration steps for each sheet in the order provided above. The tool by default configures the full clock tree to the max value for every clock without FM enabled for CORE PLL and DDR PLL, so make note of the default clock configurations when the tool is first downloaded.

10.13 Software Mapping

This section maps configurable fields in each sheet to device registers and register fields. For source clock selection in the PLL, CLKOUT, and CGM sections, the numbers in parenthesis for clock source options in the drop-down list correspond to the value that needs to be written to select that particular clock as a source. For more information, refer to the Reference Manual[1,2].

10.13.1 PLL Register Mappings

The base registers for the CORE, PERIPH and DDR PLLs registers are CORE_PLL, DDR_PLL, and PERIPH_PLL, respectively. The base register for the AURORA PLL registers is ATP. The base registers for LFAST0 and LFAST1 registers are LFAST_0 and LFAST_1, respectively.

Table 4. CORE PLL

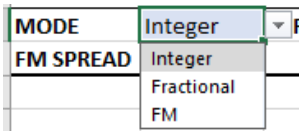
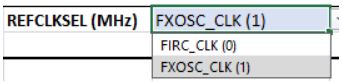
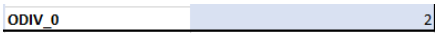
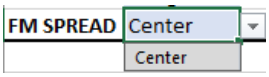
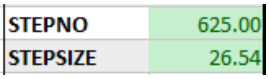
Field in Tool	Corresponding Register(s) and Field(s)
	PLLFD.SDMEN, PLLFM.SSCGBYP
	PLLCLKMUX.REFCLKSEL
	PLLDV.MFI
	PLLFD.MFN
	PLLDV.RDIV
	PLLLODIV_0.DIV
	PLLFM.SPREADCTL
	PLLFM.STEPNO, PLLFM.STEPSIZE

Table 5. PERIPH PLL

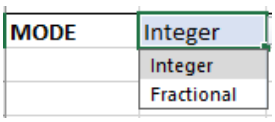
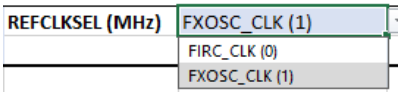
Field in Tool	Corresponding Register(s) and Field(s)
	PLLFD.SDMEN
	PLLCLKMUX.REFCLKSEL
	PLLDV.MFI
	PLLFD.MFN
	PLLDV.RDIV
	PLLLODIV_0.DIV, PLLLODIV_1.DIV, PLLLODIV_2.DIV, PLLLODIV_3.DIV, PLLLODIV_4.DIV, PLLLODIV_5.DIV, PLLLODIV_6.DIV

Table 6. DDR PLL

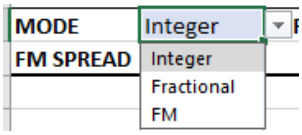
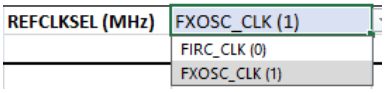
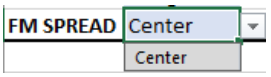
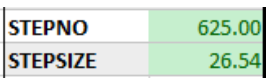
Field in Tool	Corresponding Register(s) and Field(s)
 “Mode” field	PLLFD.SDMEN, PLLFM.SSCGBYP
	PLLCLKMUX.REFCLKSEL
	PLLDV.MFI
	PLLFD.MFN
	PLLDV.RDIV
	PLLLODIV_0.DIV
	PLLFM.SPREADCTL
	PLLFM.STEPNO, PLLFM.STEPSIZE

Table 7. AURORA PLL

Field in Tool	Corresponding Register(s) and Field(s)
REFCLKSEL (MHz) FXOSC_CLK (1) FXOSC_CLK (1) AURORA_EXT_CLK (0) PERIPH_PLL_PHI_5_CLK (0)	PLLCLKMUX.REFCLKSEL, GPR2.FUNCC4[8] (select between FXOSC or PERIPH_PLL_PHI_5)
RDIV 1 ODIV_0 8 ODIV_1 1	PLLDIV.RDIV, PLLDIV[30:25] (PLLDIV.ODIV0), PLLDIV.ODIV1
MFID 50	PLLDIV.MFID
AURORA_CLK SRC (MHz) AURORA_PLL_PHI_1_CLK_DIV_10 (1) FIRC_CLK (0) AURORA_PLL_PHI_1_CLK_DIV_10 (1)	CIAC.TPIU_CLK_SEL

Table 8. LFASTn PLLs

Field in Tool	Corresponding Register(s) and Field(s)
FDIVEN 1 FBDIV 32	PLLCR.FDIVEN, PLLCR.FBDIV
PREDIV 0	PLLCR.PREDIV

10.13.2 DFS Register Mappings

The base registers for the given registers are CORE_DFS and PERIPH_DFS.

Table 9. CORE DFS

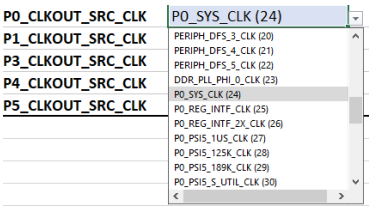
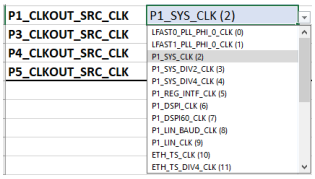
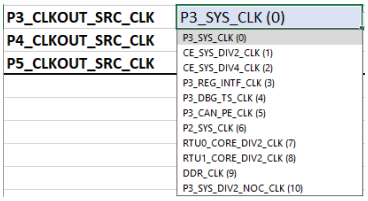
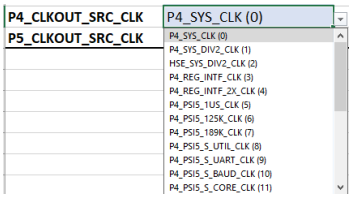
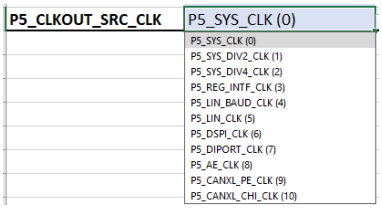
Field in Tool	Corresponding Register(s) and Field(s)
MFI/MFN Index MFI MFN 0 1 18 1 3 0 2 3 0 3 2 14 4 3 0 5 3 0	DVPORT0.MFI, DVPORT0.MFN, DVPORT1.MFI, DVPORT1.MFN, DVPORT2.MFI, DVPORT2.MFN, DVPORT3.MFI, DVPORT3.MFN, DVPORT4.MFI, DVPORT4.MFN, DVPORT5.MFI, DVPORT5.MFN

Table 10. PERIPH DFS

Field in Tool	Corresponding Register(s) and Field(s)
MFI/MFN Index MFI MFN 0 1 18 1 3 0 2 3 0 3 2 14 4 3 0 5 3 0	DVPORT0.MFI, DVPORT0.MFN, DVPORT1.MFI, DVPORT1.MFN, DVPORT2.MFI, DVPORT2.MFN, DVPORT3.MFI, DVPORT3.MFN, DVPORT4.MFI, DVPORT4.MFN, DVPORT5.MFI, DVPORT5.MFN

10.13.3 CLKOUT Register Mappings

Table 11. CLKOUT

Field in Tool	GPR Base Register	Corresponding Register(s) and Field(s)
	GPR0	CLKOUT0SEL.MUXSEL
	GPR1	CLKOUT1SEL.MUXSEL
	GPR4	CLKOUT2SEL.MUXSEL
	GPR5	CLKOUT3SEL.MUXSEL
	GPR3	CLKOUT4SEL.MUXSEL

10.13.4 CGM Register Mappings

Table 12. CGM

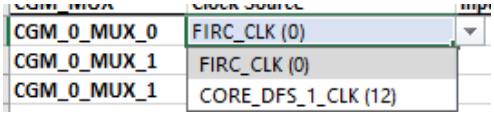
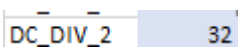
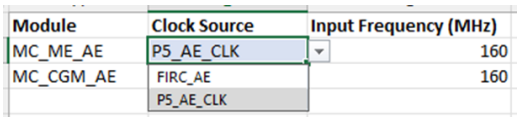
Field in Tool	CGM Base Register	Corresponding Register(s) and Field(s)
	MC_CGM_x	MUX_n_CSC.SELECTCL
	MC_CGM_x	MUX_n_DC_m.DC_DIV

Table 12. CGM...*continued*

Field in Tool	CGM Base Register	Corresponding Register(s) and Field(s)
	MC_CGM_x	MUX_n_DC_m.DIV_FMT

10.13.5 AE Register Mappings

Field in Tool	Module	Corresponding Register(s) and Field(s)
	MC_ME_AE	DRUN_MC.SYSCLK

11 References

[1] S32Z2 Reference Manual
[2] S32E2 Reference Manual
[3] S32Z2 Data Sheet
[4] S32E2 Data Sheet
[5] S32DS3.5.12
[6] EB Tresos 29.0.0

12 Revision history

Table 13. Revision history

Document ID	Release date	Description
AN13675 v.1.0	7 April 2026	Initial release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents