Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

AN220208

Clock Configuration Setup in Traveo II Body Entry Family

Author: Hirofumi Arakawa

Associated Part Family: Traveo™ II Family CYT2B Series

Related Documents: see Related Documents

AN220208 describes how to set clock sources and PLL/FLL in Traveo™ II family CYT2B series MCUs. AN220208 also provides examples for setting PLL/FLL, how to calibrate ILO, and supplementary information.

Contents

1 Introduction

Traveo II family MCUs, targeted at automotive systems such as body control units, are 32-bit automotive microcontrollers based on the Arm® Cortex®-M4 processor with FPU and manufactured on an advanced 40-nm process. These products enable a secure computing platform, and incorporate Cypress' low-power flash memory along with multiple high-performance analog and digital functions.

Traveo II Clock System supports both the internal and external clock sources, and supports high-speed clock using PLL and FLL. Traveo II Clock System also supports low-speed clock with internal and external clock. The clock source can also use external oscillator, and Traveo II supports clock input mainly used for RTC.

Traveo II also supports the function to monitor clock operation and to measure the clock difference of each clock.

To understand the functionality described and terminology used in this application note, see the Clocking System chapter in the Architecture Technical Reference Manual (Architecture TRM).

In this document, Traveo II family MCU refers to the Body Entry or the CYT2B series.

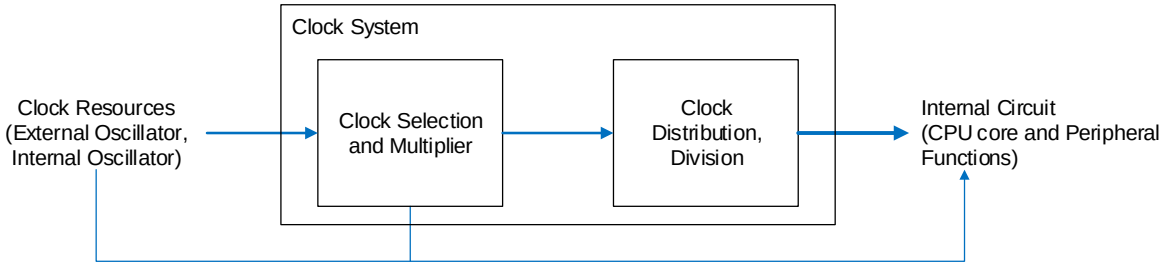
2 Clock System for Traveo II Family MCUs

2.1 Overview of the Clock System

The clock system in this CYT2B series MCU can be divided into two blocks. One block selects the clock resources such as external oscillator and internal oscillator, and multiplies the clock using FLL and PLL. The other block distributes and divides clocks to the CPU core, and peripheral functions. However, there are some exceptions such as RTC, that connect directly from a clock resource to a peripheral circuit.

Figure 1 shows the overview of the clock system structure.

Figure 1. Overview of the Clock System Structure



2.2 Clock Resources

Two kinds of clock resources, internal clock and external clock sources are input to the clock system of this MCU series. There are three types of internal clock and external clock sources:

- Internal clock sources:
  - IMO: Internal Main Oscillator. The IMO is a built-in clock, and its frequency is 8 MHz (TYP). IMO is enabled by default.
  - ILO0: Internal Low-speed Oscillator 0. ILO0 is a built-in clock, and its frequency is 32 kHz (TYP). ILO0 is enabled by default.
  - ILO1: Internal Low-speed Oscillator 1. ILO1 has the same function as ILO0, but ILO1 is available to monitor the clock of ILO0. ILO1 is enabled by default.
- External clock sources:
  - ECO: External Crystal Oscillator. This clock uses an external crystal. Input frequency range is between 3.988 MHz and 33.34 MHz. ECO is disabled by default.
  - WCO: Watch Crystal Oscillator. The WCO is mainly used in RTC. Use a clock frequency of 32.768 kHz. WCO is disabled by default.
  - EXT_CLK: External Clock. The EXT_CLK is a 0.25 MHz to 100 MHz range clock that can be sourced from a signal on a dedicated I/O pin. This clock can be used as the source clock for either PLL or FLL, or can be used directly as the high-frequency clock. EXT_CLK is disabled by default.

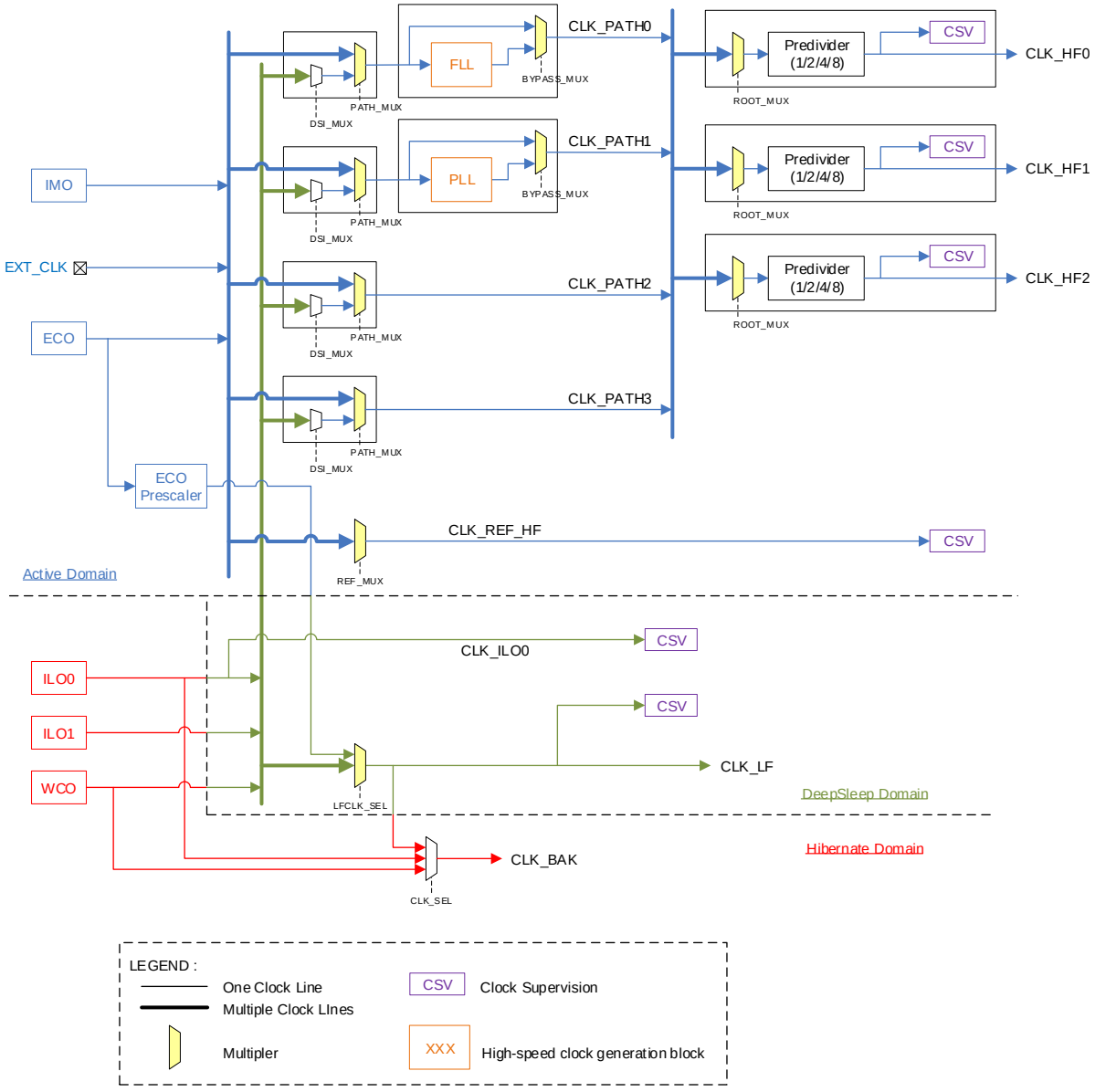
For more details on functions such as IMO, PLL, and so on, and numerical values such as frequency, see the Traveo II Architecture TRM and the Datasheet.

2.3 Functions of Clock System

This section explains the functions of the clock system.

Figure 2 shows the details of the Clock Selection and Multiplier block shown in Figure 1. This block generates CLK_HF0, CLK_HF1, and CLK_HF2 from the clock resources. CLK_HF0, CLK_HF1, and CLK_HF2 are the base clocks for operating this CYT2B series MCU. This block also selects the clock resources, and FLL and PLL to generate high-speed clock.

Figure 2. Block Diagram



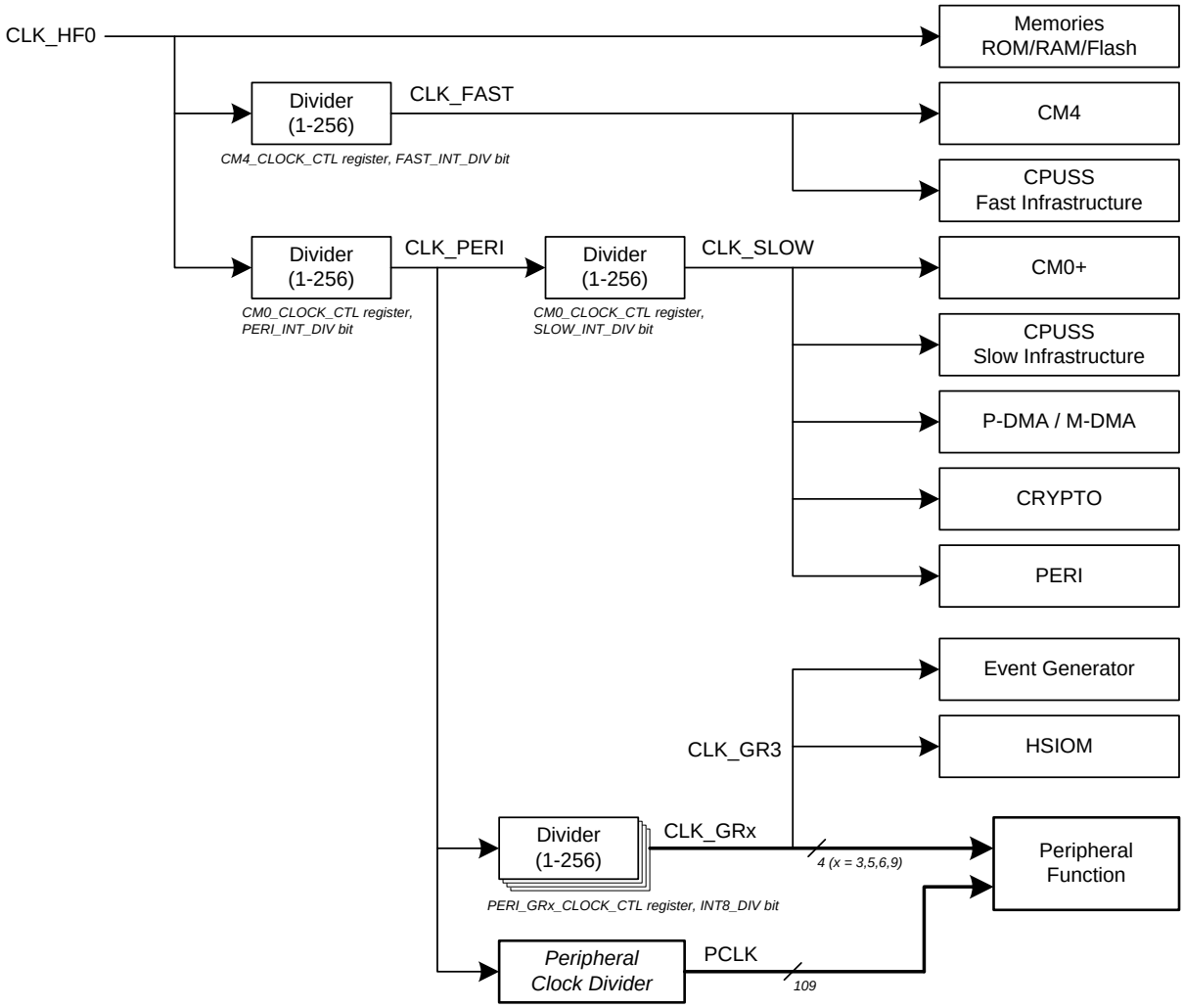
Active Domain	Active Domain is the region for operating only during active power mode.
DeepSleep Domain	DeepSleep Domain is the region for operating only during Active mode and DeepSleep mode.
Hibernate Domain	Hibernate Domain is the region for operating in all Power modes.

ECO Prescaler	ECO_Prescaler divides the ECO and creates a clock that can be used with the LFCLK clock. The division function has a 10-bit integer divider and an 8-bit fractional divider.
DSI_MUX	DSI_MUX has a function to select a clock from ILO0, ILO1, and WCO.
PATH_MUX	PATH_MUX has a function to select a clock from IMO, ECO, EXT_CLK and DSI_MUX outputs.
CLK_PATH	CLK_PATH0, CLK_PATH1, CLK_PATH2, and CLK_PATH3 are used as the input sources for CLK_HF0, CLK_HF1 and CLK_HF2.
CLK_HF	CLK_HF0, CLK_HF1, and CLK_HF2 are high-frequency clocks.
FLL	FLL is a Frequency Locked Loop which can generate high-speed clock.
PLL	PLL is a Phase Locked Loop which can generate high-speed clock.
BYPASS_MUX	BYPASS_MUX has a function to select the clock to be the output of CLK_PATH. It can either choose the output of FLL/PLL or bypass them.
ROOT_MUX	ROOT_MUX has a function to the clock source of CLK_HFx. The clocks that can be selected are CLK_PATH0, CLK_PATH1, CLK_PATH2, and CLK_PATH3.
Predivider	The Predivider is available to divide the selected CLK_PATH. 1, 2, 4, and 8 divisions can be selected.
REF_MUX	REF_MUX selects the CLK_REF_HF clock source.
CLK_REF_HF	CLK_REF_HF monitors CSV of CLK_HF.
LFCLK_SEL	LFCLK_SEL selects the CLK_LF clock source or a ECO divided clock too.
CLK_LF	CLK_LF is the MCWDT source clock.
CLK_SEL	CLK_SEL selects the clock to be input to RTC.
CLK_BAK	CLK_BAK is mainly used by RTC.
CSV	CSV is clock supervision, which monitors the operation of the clock. The clocks that can be monitored are CLK_HFs, CLK_REF_HF, ILO0, and CLK_LF.

Figure 3 shows the distribution of CLK_HF0 and the details of the Clock Distribution and Division block shown in Figure 1.

CLK_HF0 is the root clock for the CPU subsystem (CPUSS) and peripheral clock dividers. For the functions shown in the figure, see the Architecture TRM.

Figure 3. Block Diagram for CLK_HF0



CLK_FAST	CLK_FAST is the clock input for CM4 and CPUSS of the fast infrastructure.
CLK_PERI	CLK_PERI is the clock source for CLK_SLOW, CLK_GR, and peripheral clock divider.
CLK_SLOW	CLK_SLOW is the clock input for CM0+ and CPUSS of the slow infrastructure.
CLK_GR	CLK_GR is the clock input to peripheral functions. CLK_GR is grouped by Clock Gater. CLK_GR has six groups.
Divider	Divider divides each clock and can be configured from 1 to 256 divisions.

Figure 4 shows the distribution of CLK_HF1 and the details of “Clock Distribution, Division” block shown in Figure 1.

CLK_HF1 is an input source for the Event Generator that generates interrupts and triggers. These interrupts and triggers route internal CPU and peripheral signals with the GPIO. Event Generator uses not only the clock of CLK_GR3, but also the clock of CLK_HF1. Figure 4 shows the clock distribution of CLK_HF1 is shown.

For more details on the Event Generator, see the Architecture TRM.

Figure 4. Block Diagram for CLK_HF1

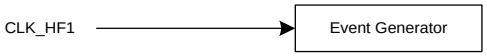


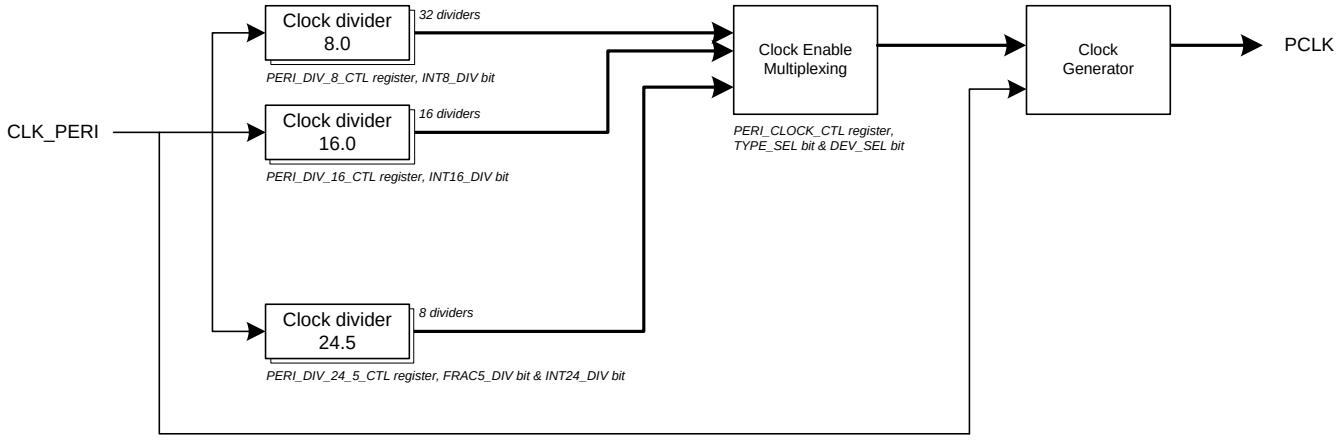
Figure 5 shows the details of the peripheral clock dividers shown in Figure 3.

An operation clock is required for peripheral functions of this CYT2B series MCU, such as the serial communication block (SCB) which is a communication function, and the Timer, Counter, and PWM (TCPWMs) used for waveform output and input signal measurement. These peripherals are clocked by the peripheral clock divider.

This CYT2B series MCU has many peripheral clock dividers to generate PCLK. It has thirty-two 8-bit dividers, sixteen 16-bit dividers, and eight 24.5-bit dividers (24 integer bits, five fractional bits). The output of any of these dividers can be routed to any peripheral.

Figure 5 shows the clock distribution of CLK_PERI. For the functions shown in Figure 5, see the Architecture TRM.

Figure 5. Block Diagram for Peripheral Clock Dividers



Clock divider8.0	Clock divided by 8.
Clock divider16.0	Clock divided by 16.
Clock divider24.5	Clock divided by 24.5.
Clock Enable Multiplexing	Clock Enable Multiplexing enables the signal output from clock divider.
Clock Generator	Clock Generator divides CLK_PERI based on clock divider.

3 Configuring Clock Resources

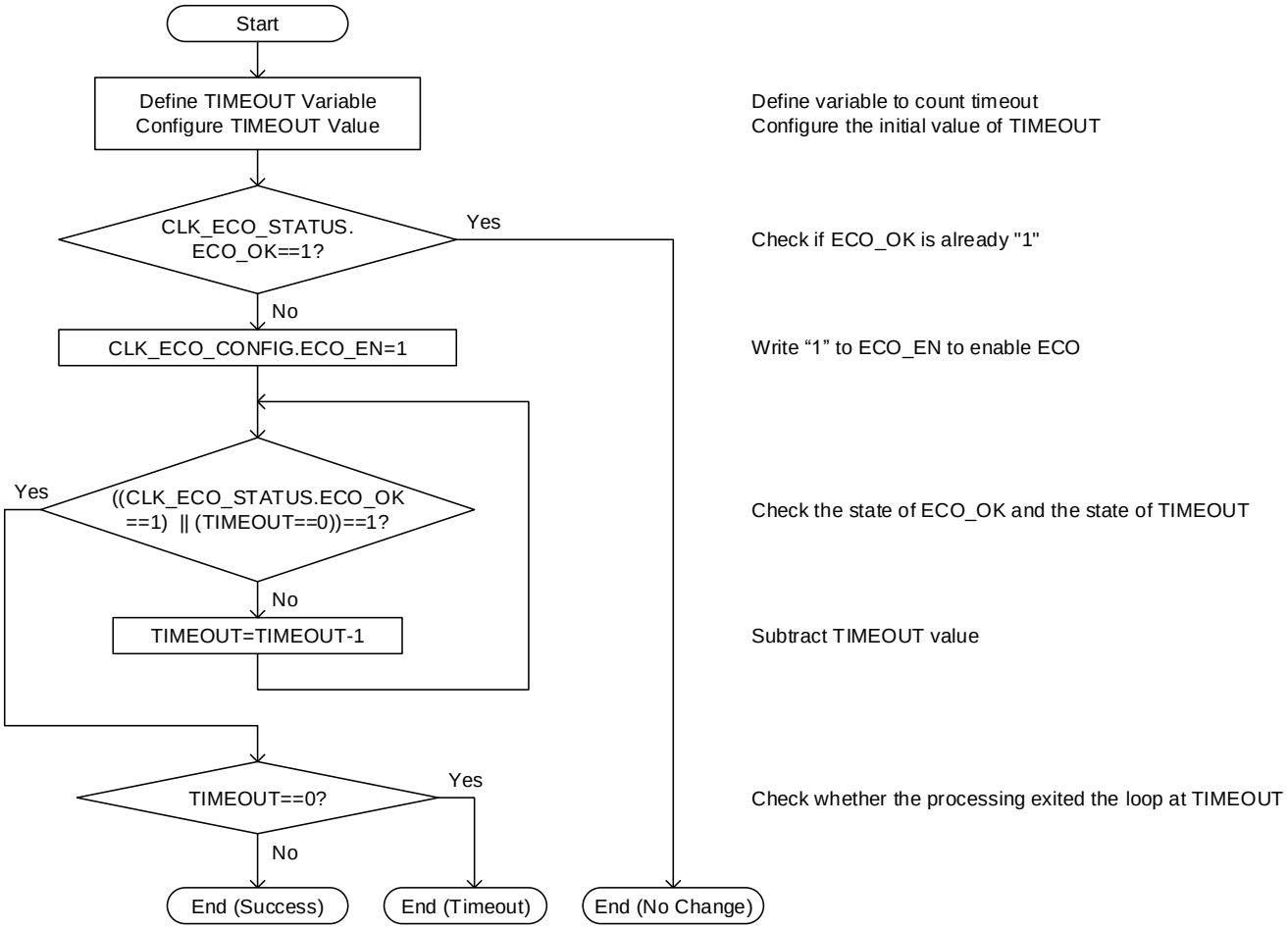
This section explains how to configure the clock resources.

3.1 Setting ECO

ECO is invalid; so, ECO cannot be used unless it is enabled. Figure 6 shows how to configure registers for enabling ECO.

To disable ECO, write ‘0’ to the ECO_EN bit of the CLK_ECO_CONFIG register.

Figure 6. Enabling ECO

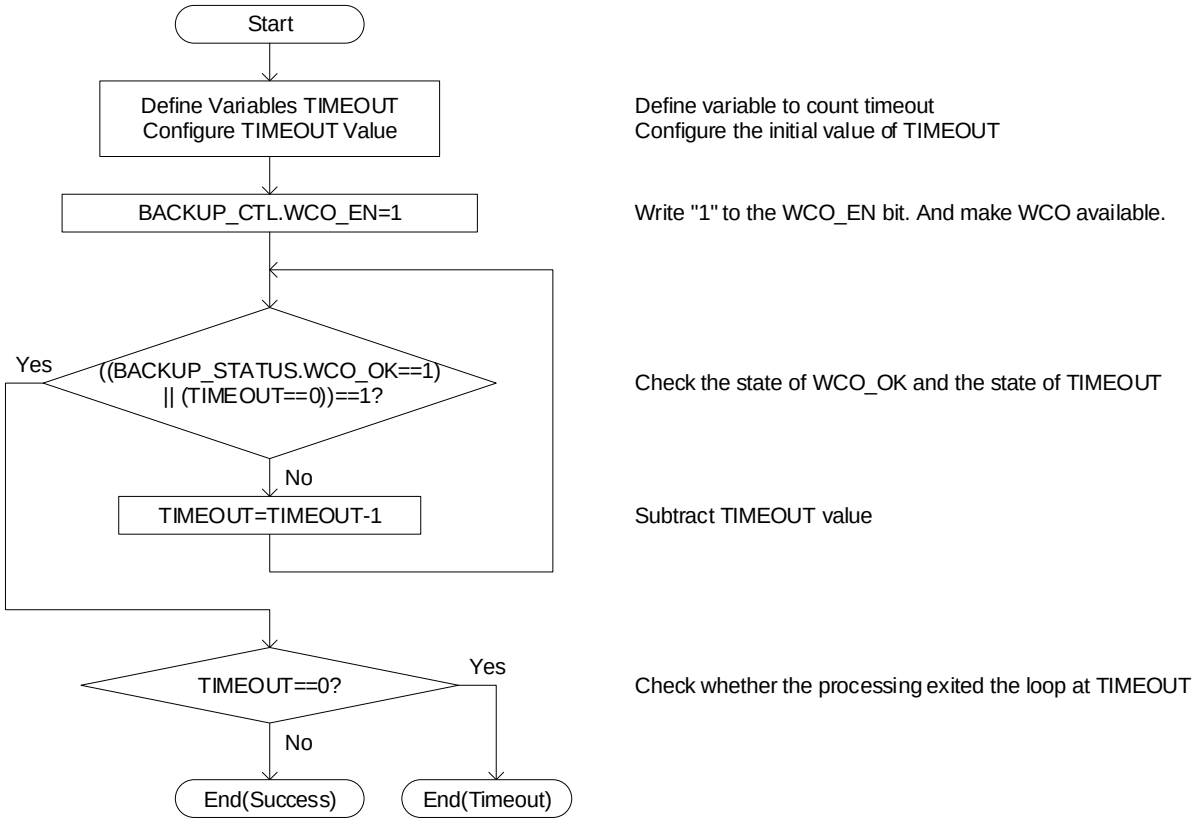


3.2 Setting WCO

WCO is disabled by default. Accordingly, WCO cannot be used unless it is enabled. Figure 7 shows how to configure registers for enabling WCO.

To disable WCO, write '0' to the WCO_EN bit of the BACKUP_CTL register.

Figure 7. Enabling WCO



3.3 Setting IMO

IMO is enabled by default so that all functions operate properly. IMO will be automatically disabled during Deep Sleep, Hibernate, and XRES. Therefore, you do not need to set IMO.

3.4 Setting ILO0/ILO1

ILO0 and ILO1 are enabled by default.

Note that ILO0 is used as the operating clock for the watchdog timer (WDT). Therefore, if ILO0 is disabled, it is necessary to disable WDT. To disable ILO0, write '01b' to the WDT_LOCK bit of the WDT_CTL register, and then write '0' to the ENABLE bit of the CLK_ILO0_CONFIG register.

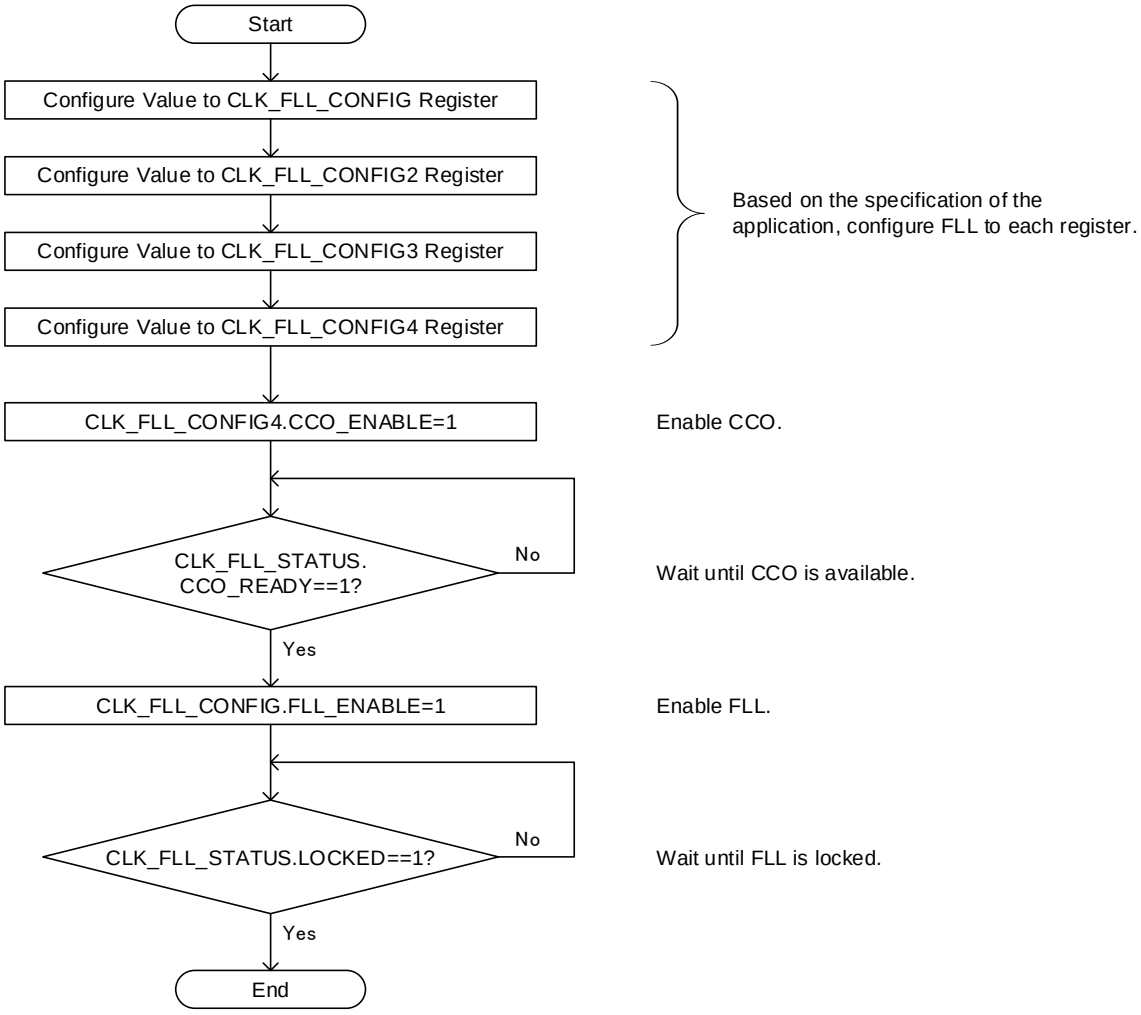
4 Configuring FLL and PLL

This section shows how to set FLL and PLL in the clock system.

4.1 Setting FLL

To use FLL, it is necessary to set FLL. FLL has a current-controlled oscillator (CCO), the output frequency of this CCO is controlled by adjusting the trim of the CCO. Figure 8 shows the steps to set FLL.

Figure 8. Procedure for Setting FLL

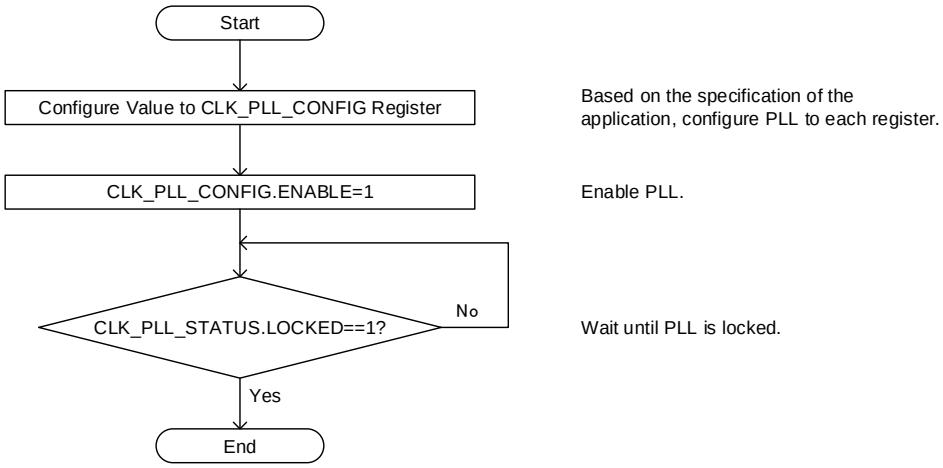


In the blocks Configure Value to CLK_FLL_CONFIGx, configure the FLL output frequency and CCO operating conditions. For details of FLL and FLL setting registers, see the Architecture TRM and Registers Technical Reference Manual (Registers TRM).

4.2 Setting PLL

To use PLL, it is necessary to set PLL. Figure 9 shows the steps to set PLL. For details on PLL, see the Architecture TRM and Registers TRM.

Figure 9. Procedure for Setting PLL



5 Configuring Internal Clock

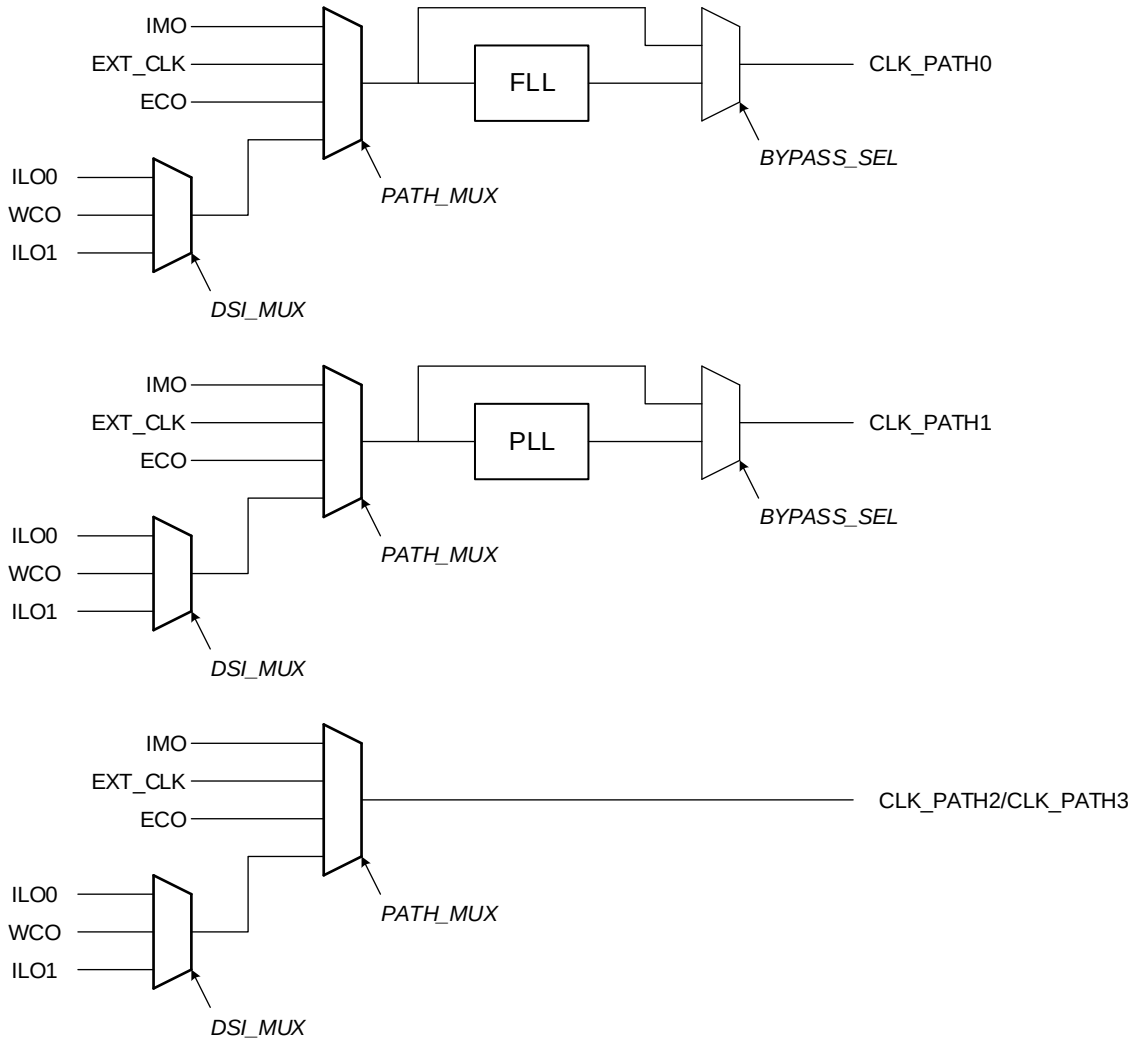
This section explains how to set the internal clock which appears such as a CLK_HF0 and CLK_FAST in the clock system.

5.1 Setting CLK_PATH0, CLK_PATH1, CLK_PATH2, and CLK_PATH3

CLK_PATH0, CLK_PATH1, CLK_PATH2, and CLK_PATH3 are used as the input sources for CLK_HF0, CLK_HF1, and CLK_HF2. CLK_PATH0 and CLK_PATH1 can select all clock resources including FLL and PLL using DSI_MUX and PATH_MUX. CLK_PATH2 and CLK_PATH3 cannot select FLL and PLL, but other clock resources can be selected.

Figure 10 shows the generation block diagram of CLK_PATH0, CLK_PATH1, CLK_PATH2, and CLK_PATH3.

Figure 10. Generation Block for CLK_PATH0, CLK_PATH1, CLKPATH2, and CLK_PATH3



To set CLK_PATH0, CLK_PATH1, CLK_PATH2, and CLK_PATH3, it is necessary to configure DSI_MUX and PATH_MUX. BYPASS_SEL is also required for CLK_PATH0 and CLK_PATH1. Table 1 shows the registers necessary for CLK_PATH. See the Architecture TRM and Registers TRM for more details.

Table 1. Configuring CLK_PATH0, CLK_PATH1, and CLK_PATH2

Register Name	Bit Name	Value	Selected Clock and Item
CLK_PATH_SELECT	PATH_MUX[2:0]	0 (Default)	IMO
		1	EXT_CLK
		2	ECO
		4	DSI_MUX
		Other	Reserved. Do not use.
CLK_DSI_SELECT	DSI_MUX[4:0]	16	ILO0
		17	WCO
		20	ILO1
		Other	Reserved. Do not use.
CLK_FLL_CONFIG3	BYPASS_SEL[29:28]	0 (Default)	AUTO[1]
		1	LOCKED_OR_NOTHING[2]
		2	FLL_REF (bypass mode) [3]
		3	FLL_OUT[4]
CLK_PLL_CONFIG	BYPASS_SEL[29:28]	0 (Default)	AUTO[1]
		1	LOCKED_OR_NOTHING[2]
		2	PLL_REF (bypass mode) [3]
		3	PLL_OUT[4]

5.2 Setting CLK_HF

CLK_HF0, CLK_HF1, and CLK_HF2 can be selected from CLK_PATH0, CLK_PATH1, CLK_PATH2, and CLK_PATH3. A predivider is available to divide the selected CLK_PATH0, CLK_PATH1, CLK_PATH2, and CLK_PATH3. CLK_HF0 is always enabled because it is the source clock for the CPU. It is possible to disable CLK_HF1 and CLK_HF2.

To enable CLK_HF1, write ’1’ to the ENABLE bit of the CLK_ROOT_SELECT register. To disable CLK_HF1 and CLK_HF2, write ‘0’ to the ENABLE bit of the CLK_ROOT_SELECT register.

CLK_PATH0 is the clock output from FLL. CLK_PATH1 is the clock output from PLL. CLK_PATH2 and CLK_PATH3 are the source clock selected by PATH_MUX and DSI_MUX. The ROOT_DIV bit of the CLK_ROOT register sets the predivider values from the options: no division, divide by 2, divide by 4, and or by 8. Figure 11 shows the details of ROOT_MUX and the predivider.

Figure 11. ROOT_MUX and Predivider

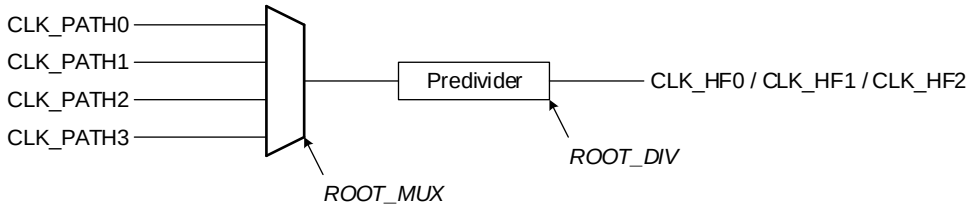


Table 2 shows the registers necessary for CLK_HF0, CLK_HF1, and CLK_HF2. See Architecture TRM and Registers TRM.

[1] Switching automatically according to locked state.

[2] The clock is gated OFF, when unlocked.

[3] In this mode, lock state is ignored.

[4] In this mode, lock state is ignored.

Table 2. Configuring CLK_HF0 and CLK_HF1

Register Name	Bit Name	Value	Selected Item
CLK_ROOT_SELECT	ROOT_MUX[3:0]	0	CLK_PATH0
		1	CLK_PATH1
		2	CLK_PATH2
		3	CLK_PATH3
		Other	Reserved. Do not use.
CLK_ROOT_SELECT	ROOT_DIV[5:4]	0	No division
		1	Divide clock by 2
		2	Divide clock by 4
		3	Divide clock by 8

5.3 Setting CLK_LF

CLK_LF can be selected from WCO, ILO0, ILO1, and ECO_Prescaler. CLK_LF cannot be set when the WDT_LOCK bit in the WDT_CLTL register is disabled, because CLK_LF can select ILO0.

Figure 12 shows the details of LFCLK_SEL.

Figure 12. LFCLK_SEL

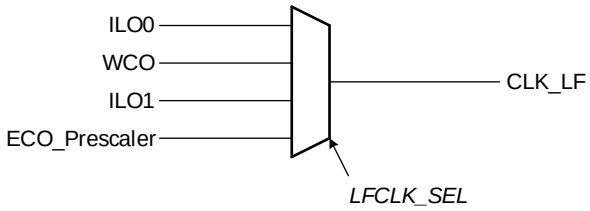


Table 3 shows the registers necessary for CLK_LF. See the Architecture TRM and Technical Reference Manuals for more details.

Table 3. Configuring CLK_LF

Register Name	Bit Name	Value	Selected Item
CLK_SELECT	LFCLK_SEL[2:0]	0	ILO0
		1	WCO
		5	ILO1
		6	ECO_Prescaler
		Other	Reserved. Do not use.

5.4 Setting CLK_FAST

CLK_FAST is generated by dividing CLK_HF0 by (x+1). When configuring CLK_FAST, configure value (x = 0..255) to be divided by the FAST_INT_DIV bit of the CM4_CLOCK_CTL register.

5.5 Setting CLK_PERI

CLK_PERI is the clock input to peripheral clock divider. CLK_PERI is generated by dividing CLK_HF0; its frequency is configured by the value obtained by dividing CLK_HF0 by (x+1). When configuring CLK_PERI, configure value (x = 0..255) to be divided by the PERI_INT_DIV bit of the CM0_CLOCK_CTL register.

5.6 Setting CLK_SLOW

CLK_SLOW is generated by dividing CLK_PERI; its frequency is configured by the value obtained by dividing CLK_PERI by (x+1). After configuring CLK_PERI, configure value (x = 0..255) to be divided by the SLOW_INT_DIV bit of the CM0_CLOCK_CTL register.

5.7 Setting CLK_GR

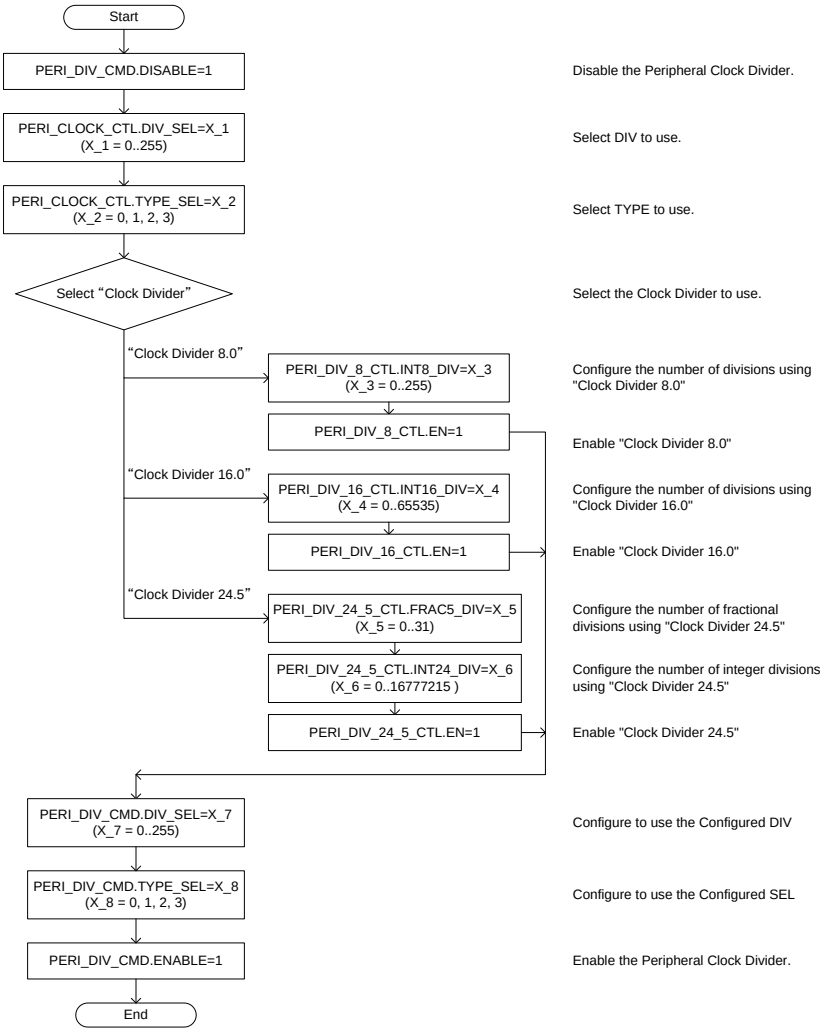
The clock source of CLK_GP is CLK_SLOW in Groups 1 and 2, and CLK_PERI in Groups 3, 5, 6, and 9. Groups 3, 5, 6, and 9 are clocks divided by CLK_PERI. To generate CLK_GR, write the division value (from 1 to 255) to divide the CLOCK_CTL bit of the PERI_GR_CLOCK_CTL register.

5.8 Setting PCLK

Peripheral Clock (PCLK) is a clock that activates each peripheral function. Peripheral clock dividers divide CLK_PERI and generate a clock to be supplied to each peripheral function. For assignment of the peripheral clocks, see the Peripheral Clocks section in the Datasheet.

Figure 13 shows the steps to set peripheral clock dividers. See the Architecture TRM for more details.

Figure 13. Procedure to Set Generate PCLK



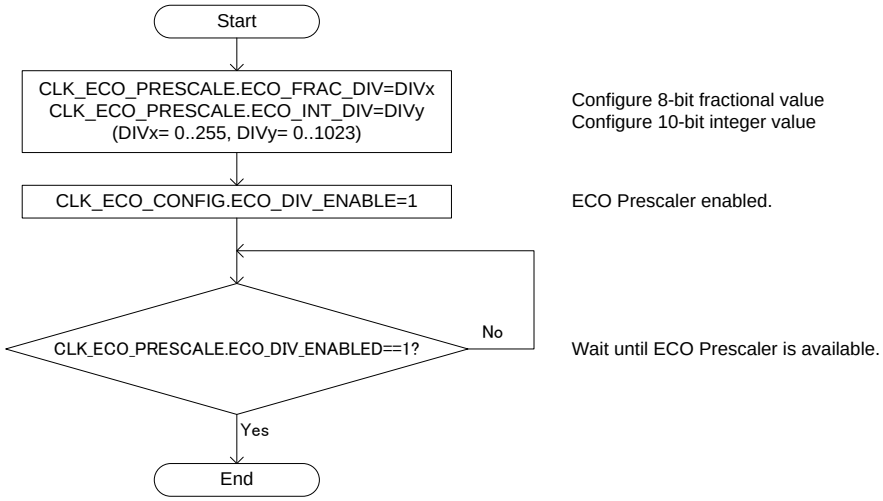
Note: If DIV_SEL is "63" and TYPE_SEL is "3" (default/reset value), no divider is specified and no clock signal(s) are generated.

5.9 Setting ECO_Prescaler

ECO_Prescaler divides ECO, and creates a clock that can be used with the LFCLK clock. The division function has a 10-bit integer divider and 8-bit fractional divider.

Figure 14 shows the steps to enable ECO_Prescaler. below. For details on ECO_Prescaler, see Architecture TRM and Registers TRM.

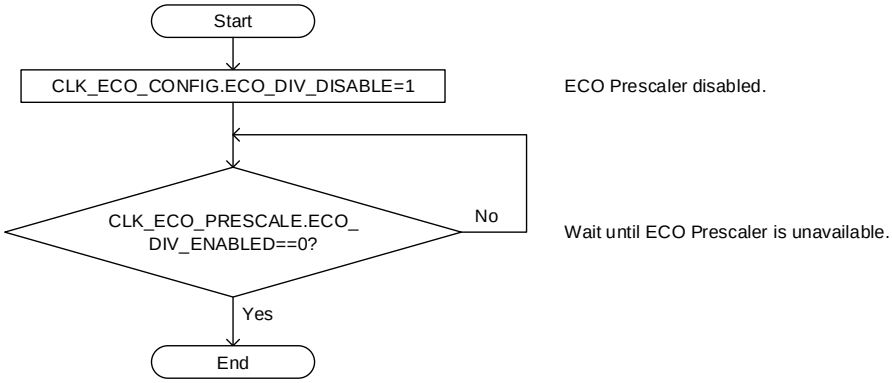
Figure 14. Enabling ECO_Prescaler



Note: Do not change the ECO_FRAC_DIV and ECO_INT_DIV settings when ECO_DIV_ENABLE = 1.

Figure 15 shows the steps to disable ECO_Prescalar. For details on ECO_Prescaler, see Architecture TRM and Registers TRM.

Figure 15. Disabling ECO_Prescaler



6 Example for Configuring Internal Clock

This section explains how to configure the internal clock and others.

6.1 General Setting Flow

Table 4 lists the steps to configure the sample flow in this CYT2B series MCU clock system.

Table 4. Configuring Clock System

Step	Description	Options
1	Select the source clock of the operation clock according to the application specifications.	Select the source clock of the operation clock: • IMO • ECO • EXT_CLK • ILO • WCO
2	Select FLL, PLL, do not use FLL and PLL according to the application specifications.	Select the source clock of the operation clock when using high frequency: • PLL • FLL • Do not use FLL and PLL
3	Select the clock input to CLK_HF0 and CLK_HF1 according to the application specifications. Also, configure the clock frequency of CLK_HF0 and CLK_HF1.	Select the clock input to CLK_HFx: • Predivider • MUX
4	Configure CLK_FAST	Configure the value to be entered in Divider. See 5.4 Setting CLK_FAST.
5	Configure CLK_SLOW	Configure the value to be entered in Divider. See 5.6 Setting CLK_SLOW.
6	Configure CLK_PERI	Configure the value to be entered in Divider. See 5.5 Setting CLK_PERI.
7	Configure PCLK according to the application specifications	Configure the value to be entered in Divider. See 5.8 Setting PCLK.
8	Configure CLK_GR according to the application specifications	Configure the value to be entered in Divider. See 5.7 Setting CLK_GR.

6.2 Example 1: Setting PLL Using ECO

This example demonstrates how to configure a PLL with ECO as the clock source with the parameters listed in Table 5.

The following are the parameters of PLL using ECO:

- Clock source: ECO 4 MHz
- PLL: 100 MHz
- CM4 operation clock(CLK_FAST): 100 MHz
- CM0+ operation clock(CLK_SLOW): 12.5 MHz
- CLK_PERI: 50 MHz

Table 5. Configuring PLL using ECO

Step	Description	Operation	Reference Section
1	Configure ECO	Enable ECO	Setting ECO
2	Configure PATH_MUX	Select ECO	Setting CLK_PATH0, CLK_PATH1, CLK_PATH2
3	Configure PLL	To configure the PLL output as 100 MHz: FEEDBACK_DIV = 50 REFERENCE_DIV = 1 OUTPUT_DIV = 2	Setting PLL
4	Configure CLK_HF	Select CLK_PATH1	Setting CLK_HF
5	Configure CLK_FAST	FAST_INT_DIV = 0	Setting CLK_FAST
6	Configure CLK_SLOW	SLOW_INT_DIV = 3	Setting CLK_SLOW
7	Configure CLK_PERI	PERI_INT_DIV = 1	Setting CLK_PERI
8	Configure PCLK	Set according to peripheral function to be used	Setting PCLK

6.3 Example 2: Setting FLL Using ECO

This example demonstrates how to configure a FLL with ECO as the clock source with the parameters listed in Table 6.

The following are the parameters of FLL using ECO:

- Clock source: ECO 8 MHz
- FLL: 80 MHz
- CM4 operation clock(CLK_FAST): 80 MHz
- CM0+ operation clock(CLK_SLOW): 40 MHz
- CLK_PERI: 40 MHz

Table 6. Configuring of FLL Using ECO

Step	Description	Operation	Reference Section
1	Configure ECO	Enable ECO	Setting ECO
2	Configure PATH_MUX	Select ECO	Setting CLK_PATH0, CLK_PATH1, CLK_PATH2
3	Configure FLL	To configure the FLL output as 80 MHz: FLL_MULT = 1720 FLL_OUTPUT_DIV = 1 FLL_RFE_DIV = 86	Setting FLL
4	Configure CLK_HF	Select CLK_PATH0	Setting CLK_HF
5	Configure CLK_FAST	FAST_INT_DIV = 0	Setting CLK_FAST
6	Configure CLK_SLOW	SLOW_INT_DIV = 3	Setting CLK_SLOW
7	Configure CLK_PERI	PERI_INT_DIV = 1	Setting CLK_PERI
8	Configure PCLK	Set according to peripheral function to be used	Setting PCLK

7 Supplementary Information

7.1 Input Clocks in Peripheral Functions

Table 7 to Table 11 list the clock input to each peripheral function. For detailed values of PCLK, see the Peripheral Clocks section in the Datasheet.

Table 7. Clock Input to TCPWM

Peripheral Function	Operation Clock	Channel Clock
TCPWM (16-bit)	CLK_GR3 (Group 3)	PCLK (PCLK_TCPWM0_CLOCKSx, x=0-62)
TCPWM (16-bit) (Motor Control)		PCLK (PCLK_TCPWM0_CLOCKSy, y=256-267)
TCPWM (32-bit)		PCLK (PCLK_TCPWM0_CLOCKSz, z=512-515)

Table 8. Clock Input to CAN FD

Peripheral Function	Operation Clock (clk_can (cclk))	Channel Clock (clk_sys (hclk))
CAN FD0	CLK_GR5 (Group 5)	Ch0: PCLK (PCLK_CANFD0_CLOCK_CANFD0)
		Ch1: PCLK (PCLK_CANFD0_CLOCK_CANFD1)
		Ch2: PCLK (PCLK_CANFD0_CLOCK_CANFD2)
CAN FD1		Ch0: PCLK (PCLK_CANFD1_CLOCK_CANFD0)
		Ch1: PCLK (PCLK_CANFD1_CLOCK_CANFD1)
		Ch2: PCLK (PCLK_CANFD1_CLOCK_CANFD2)

Table 9. Clock Input to LIN

Peripheral Function	Operation Clock	Channel Clock (clk_lin_ch)
LIN	CLK_GR5 (Group 5)	Ch0: PCLK (PCLK_LIN_CLOCK_CH_EN0)
		Ch1: PCLK (PCLK_LIN_CLOCK_CH_EN1)
		Ch2: PCLK (PCLK_LIN_CLOCK_CH_EN2)
		Ch3: PCLK (PCLK_LIN_CLOCK_CH_EN3)
		Ch4: PCLK (PCLK_LIN_CLOCK_CH_EN4)
		Ch5: PCLK (PCLK_LIN_CLOCK_CH_EN5)
		Ch6: PCLK (PCLK_LIN_CLOCK_CH_EN6)
		Ch7: PCLK (PCLK_LIN_CLOCK_CH_EN7)

Table 10. Clock Input to SCB

Peripheral Function	Operation Clock	Channel Clock
SCB0	CLK_GR6 (Group 6)	PCLK (PCLK_SCB0_CLOCK)
SCB1		PCLK (PCLK_SCB1_CLOCK)
SCB2		PCLK (PCLK_SCB2_CLOCK)
SCB3		PCLK (PCLK_SCB3_CLOCK)
SCB4		PCLK (PCLK_SCB4_CLOCK)
SCB5		PCLK (PCLK_SCB5_CLOCK)
SCB6		PCLK (PCLK_SCB6_CLOCK)
SCB7		PCLK (PCLK_SCB7_CLOCK)

Table 11. Clock Input to SAR ADC

Peripheral Function	Operation Clock	Unit Clock
SAR ADC	CLK_GR9 (Group 9)	Unit0: PCLK (PCLK_PASS_CLOCK_SAR0)
		Unit1: PCLK (PCLK_PASS_CLOCK_SAR1)
		Unit2: PCLK (PCLK_PASS_CLOCK_SAR2)

7.2 Use Case of Clock Calibration Counter Function

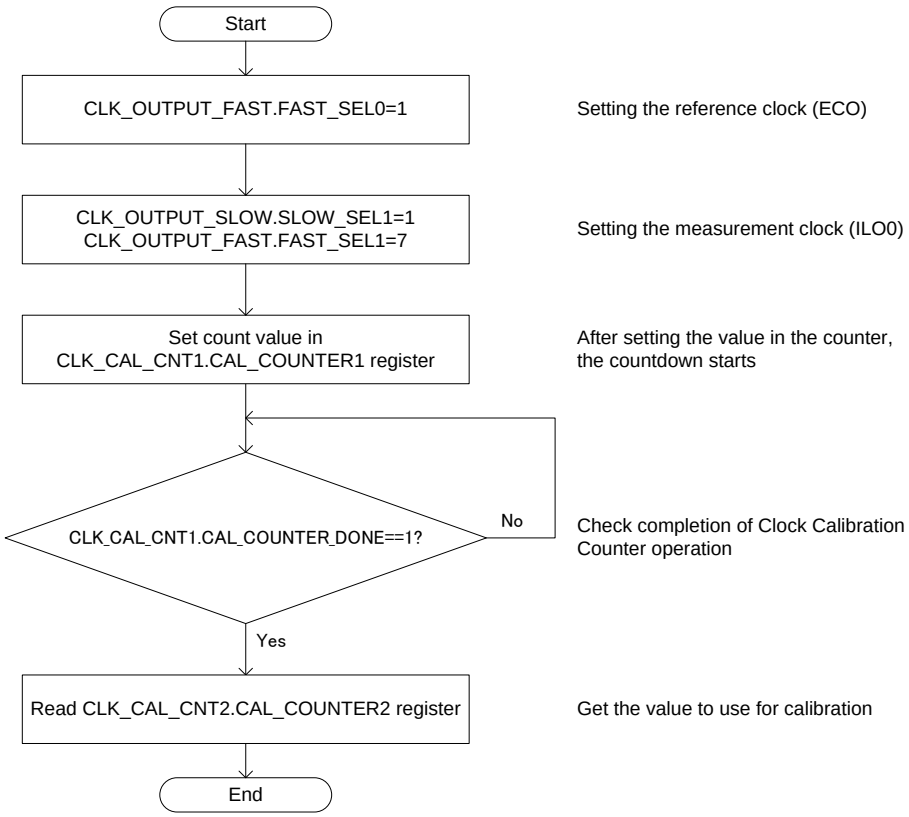
The Clock Calibration Counter has two counters that can be used to compare the frequency of two clock sources. All clock sources are available as a source for these two clocks.

1. Calibration Counter1 counts clock pulses from Calibration Clock1 (the high-accuracy clock used as the reference clock). Counter1 counts in decreasing order.
2. Calibration Counter2 counts clock pulses from Calibration Clock2 (measurement clock). This counter counts in increasing order.
3. When Calibration Counter1 reaches 0, Calibration Counter2 stops counting, and its value can be read.
4. The frequency of Calibration Counter2 can be obtained by using the value and the following equation:

$$\text{CalibrationClock2} = \frac{\text{Counter2value}}{\text{Counter1value}} \times \text{CalibrationClock1}$$

Figure 16 shows an example of the Clock Calibration Counter function when ILO0 and ECO are used. ILO0 and ECO must be enabled. See ILO0 and ECO for Setting ILO0/ILO1 and Setting ECO.

Figure 16. Example of Clock Calibration Counter with ILO0 and ECO



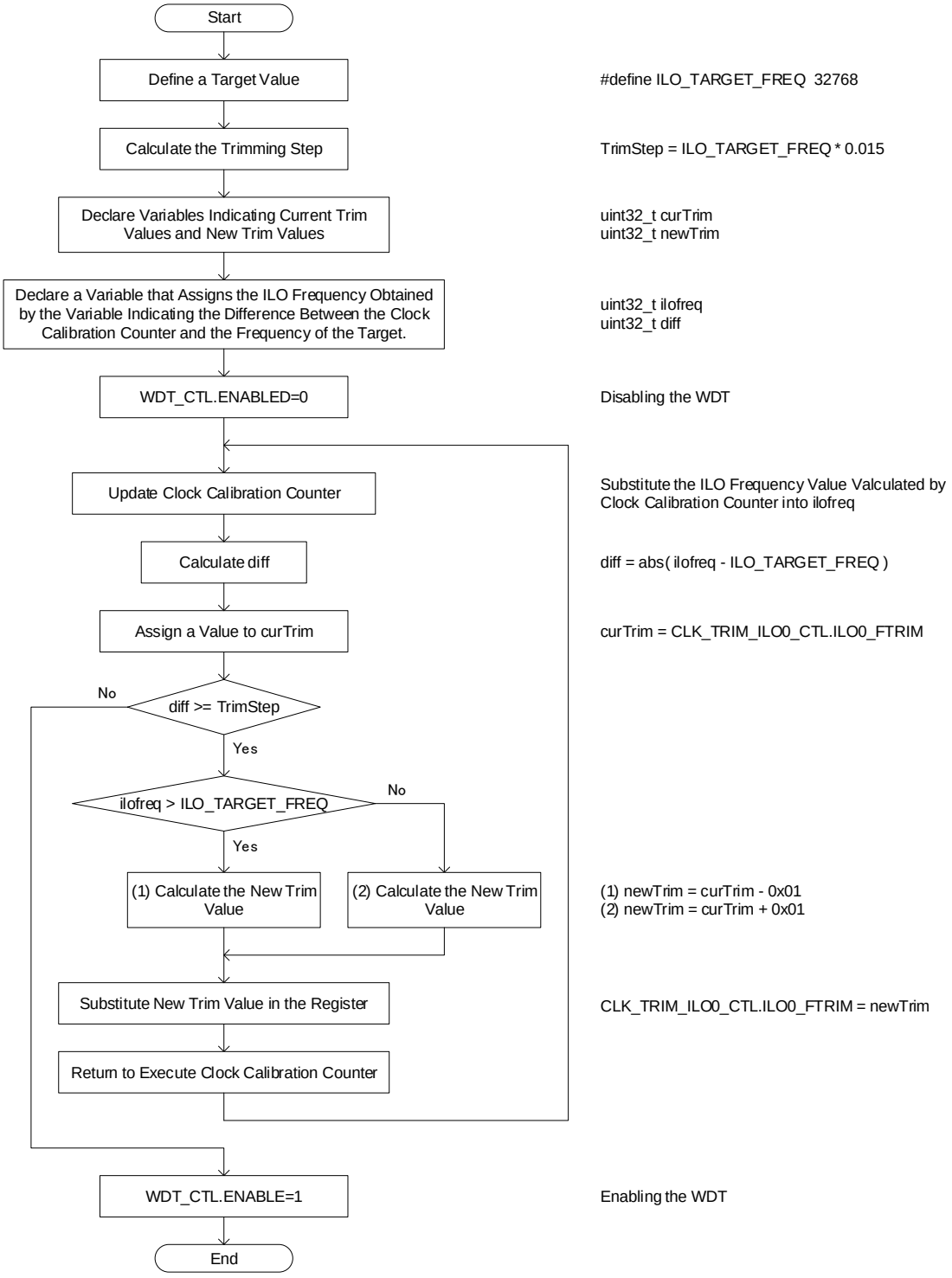
7.2.1 ILO0 Calibration Using Clock Calibration Counter Function

The ILO frequency is determined during manufacturing; however, the ILO frequency can be updated on the field to change according to the voltage and temperature conditions.

The ILO frequency trim can be updated using the ILOx_FTRIM bit of the CLK_TRIM_ILOx_CTL register. The initial value of the ILOx_FTRIM bit is 0x2C. Increasing the value of this bit by 0x01 increases the frequency by 1.5% (typical); decreasing this bit value by 0x01 decreases the frequency by 1.5% (typical). The CLK_TRIM_ILO0_CTL register is protected by WDT_CTL.ENABLE. For the specification of the WDT_CTL register, see the Watchdog Timer section of the Traveo II Architecture TRM.

Figure 17 shows an example flow of ILO0 calibration using Clock Calibration Counter and the CLK_TRIM_ILOx_CTL register.

Figure 17. ILO0 Calibration



Note: Calculation of each variable must be appropriately dealt with so that decimal places do not occur.

8 Glossary

Terms	Description
FPU	Floating Point Unit
RTC	Real Time Clock
IMO	Internal Main Oscillator
ILO	Internal Low-speed Oscillators
ECO	External Crystal Oscillator
WCO	Watch Crystal Oscillator
EXT_CLK	External Clock
PLL	Phase Locked Loop
FLL	Frequency Locked Loop
CLK_HF	High Frequency Clock. The CLK_HF derive both CLK_FAST and CLK_SLOW. CLK_HF, CLK_FAST, and CLK_SLOW are synchronous to each other.
CLK_FAST	Fast Clock. The CLK_FAST is used for the CM4 and CPUSS Fast Infrastructure.
CLK_SLOW	Slow Clock. The CLK_FAST is used for the CM4 and CPUSS Slow Infrastructure.
CLK_PERI	Peripheral Clock. The CLK_PERI is the clock source for CLK_SLOW, CLK_GR, and peripheral clock divider.
CLK_GR	Group Clock. The CLK_GR is the clock input to peripheral functions.
Peripheral Clock Divider	Peripheral clock divider derives a clock to use of each peripheral function.
MCWDT	Multi-Counter Watchdog Timer. See Watchdog Timer chapter of Traveo II Architecture TRM for details.
TCPWM	Timer, Counter, and Pulse Width Modulator. See the Timer, Counter, and PWM chapter of Traveo II Architecture TRM for details.
CAN FD	CAN FD is the CAN with Flexible Data rate, and CAN is the Controller Area Network. See the "CAN FD Controller" chapter of Traveo II Architecture TRM for details.
LIN	Local Interconnect Network. See the Local Interconnect Network (LIN) chapter in Traveo II Architecture TRM for details.
SCB	Serial Communications Block. See the Serial Communications Block (SCB) chapter in Traveo II Architecture TRM for details.
SAR ADC	Successive Approximation Register Analog-to-Digital Converter. See the SAR ADC chapter in Traveo II Architecture TRM for details.
Clock Calibration Counter	Clock Calibration Counter has a function to calibrate the clock using two clocks.

9 Related Documents

- Technical Reference Manuals
  - Traveo II Automotive Body Controller Entry Family Architecture Technical Reference Manual (Contact Technical Support)
  - Traveo II Automotive Body Controller Entry Registers Technical Reference Manual (Contact Technical Support)
- Datasheet
  - Traveo II Device Datasheet (Contact Technical Support)

Document History

Document Title: AN220208 – Clock Configuration Setup in Traveo II Body Entry Family

Document Number: 002-20208

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6100869	HIAR	04/17/2019	New application note.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709