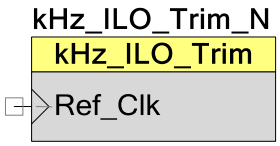
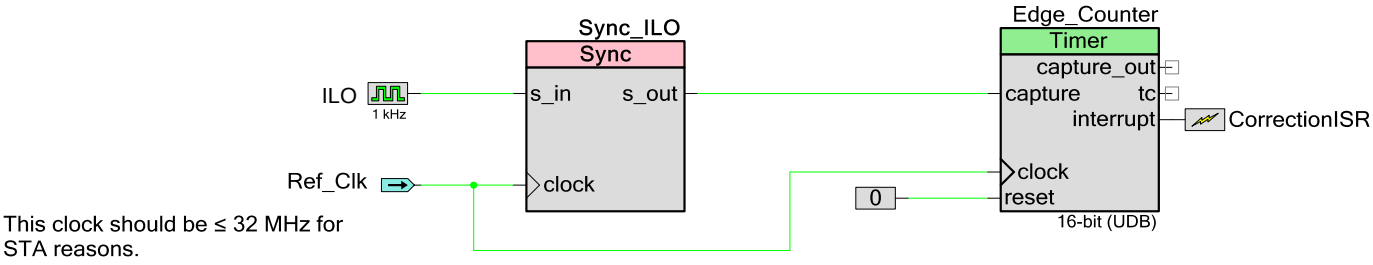
kHz_ILO_Trim
1.40

Features



- Measures frequency error of Internal Low-speed Oscillator (ILO)
- Trims the ILO using an external higher-accuracy clock

Schematic



General Description

The ILO is calibrated by using a higher accuracy clock source like the IMO or an external crystal oscillator as a reference. In this implementation, the number of reference clock edges per cycle of the ILO are counted and compared to the ideal number of reference edges that should have been present. If the reference edges are fewer than ideal, then the frequency of the ILO is reduced. If the reference edges are greater than ideal, then the frequency of the ILO is increased.

When to Use the kHz_ILO_Trim Component

Use the kHz_ILO_Trim component to achieve tighter 1 kHz ILO clock accuracy.

One of the major uses of the kHz ILO is as a clock in sleep mode. A more accurate ILO allows for more precise event timing.

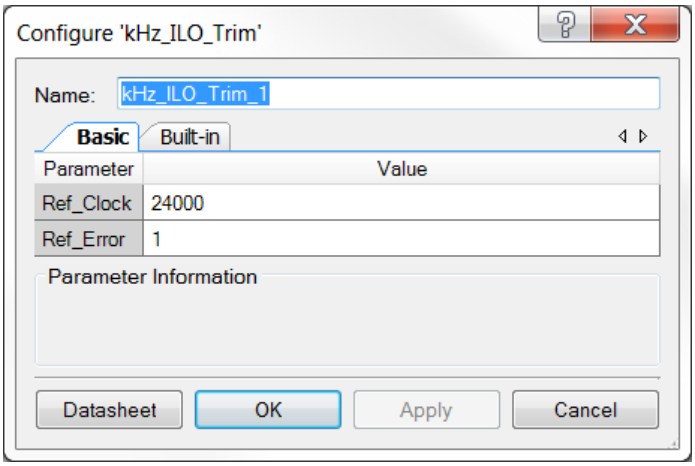
Input/Output Connections

There is one input and no outputs to the component.

Input	May Be Hidden	Description
Ref_Clk	N	Reference clock from which ILO derives higher accuracy. For better accuracy, this should be >1 MHz.

Parameters

Drag a kHz_ILO_Trim component on to your design and double-click it to open the Configure dialog box.



Ref_Clock

The frequency in kHz of Ref_Clock input.

Ref_Error

The error in percentage of the reference clock.

Placement

The timer used in this component is UDB-based. PSoC Creator automatically places it within the UDB array in the best possible configuration.

Resources

The kHz_ILO_Trim component uses the following device resources:

Resources	Count
Datapath	2 of 24
Status cells	1 of 24
Control/count7 cells	1 of 24
PLDs	2 of 48
Macrocells	6 of 192
Interrupts	1 of 32
Digital clock dividers	0 of 8
Sync blocks	1 of 92
Flash (bytes)	776
RAM (bytes)	23
CPU cycles consumed per iteration	PSoC 3: 314 PSoC 5: 148

Application Programming Interface

Application Programming Interface (API) routines allow you to interact with the component using firmware. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name kHz_ILO_Trim_1 to the first instance of the component in a given design. You can rename it to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name of the component is taken to be kHz_ILO_Trim.

Function	Description
kHz_ILO_Trim_Start()	Starts the component but does not begin the trim iterations
kHz_ILO_Trim_Stop()	Stops the component
kHz_ILO_Trim_BeginTrimming()	Enables the ISR to begin trimming the ILO
kHz_ILO_Trim_StopTrimming()	Disables trimming

Function	Description
kHz_ILO_Trim_ClearFIFO()	Clears the edge counter's capture FIFO
kHz_ILO_Trim_CheckStatus()	Returns current status of ILO Trim
kHz_ILO_Trim_CheckError()	Returns ILO frequency error value in parts per thousand
kHz_ILO_Trim_RestoreTrim()	Restores ILO factory trim
kHz_ILO_GetTrim()	Returns current ILO trim value
kHz_ILO_SetTrim()	Sets the ILO trim
kHz_ILO_Trim_Sleep()	Prepares the component for sleep
kHz_ILO_Trim_Wakeup()	Prepares the component for wakeup

void kHz_ILO_Trim_Start(void)

Description: Starts the Counter, and then starts and assigns a vector to CorrectionISR.

Parameters: None

Return Value: None

Side Effects: Sets the isrOff global variable.

void kHz_ILO_Trim_Stop(void)

Description: Stops the Timer and the ISR.

Parameters: None

Return Value: None

Side Effects: None

void kHz_ILO_Trim_BeginTrimming(void)

Description: Enables the CorrectionISR. This allows the trimming procedure to begin.

Parameters: None

Return Value: None

Side Effects: Clears the isrOff global variable.

void kHz_ILO_Trim_StopTrimming(void)

Description: Disables the CorrectionISR.

Parameters: None

Return Value: None

Side Effects: Sets the isrOff global variable.

void kHz_ILO_Trim_ClearFIFO(void)

Description: Clears the edge counter's capture FIFO. Typically used before the CheckError function to ensure that the CheckError function does not return a stale result.

Note: If the FIFO is full, and subsequent writes occur (overflow), the new data overwrites the front of the FIFO (the data currently being output, the next data to read)

Parameters: None

Return Value: None

Side Effects: None

uint8 kHz_ILO_Trim_CheckStatus(void)

Description: Checks whether the ILO frequency is within the acceptable error range. It requires the Edge_Counter to be running.

Parameters: None

Return Value:

Value	Description
kHz_ILO_Trim_IS_INACCURATE	ILO frequency is outside error range. Typically if the return is IS_INACCURATE, BeginTrimming() is called.
kHz_ILO_Trim_IS_ACCURATE	ILO is within error range.
kHz_ILO_Trim_NOT_READY	Timer Capture is not Full.
kHz_ILO_Trim_ISR_ON	CorrectionISR is running .

Side Effects: Clears Timer capture FIFO - so should not be used immediately after the CheckError function.

int16 kHz_ILO_Trim_CheckError(void)

Description: Calculates ILO frequency error in parts per thousand.

Parameters: None

Return Value: Returns an error value in parts per thousand if the CorrectionISR is not running and the timer capture is full. Otherwise, it returns NOT_READY or ISR_ON.

Value	Description
Error in parts per thousand	ILO frequency error
kHz_ILO_Trim_NOT_READY	Timer Capture is not Full.
kHz_ILO_Trim_ISR_ON	CorrectionISR is running .

Side Effects: Clears Timer capture FIFO. Therefore, it should not be used immediately after the CheckStatus function.

void kHz_ILO_Trim_RestoreTrim(void)

Description: Restores factory trim value.

Parameters: None

Return Value: None

Side Effects: Stops CorrectionISR and sets the isrOff global variable.

uint8 kHz_ILO_Trim_GetTrim(void)

Description: Returns current ILO Trim value.

Parameters: None

Return Value: uint8 number which contains the concatenated 4-bit fine trim and 3-bit coarse trim value for the ILO.

Side Effects: None

void kHz_ILO_Trim_SetTrim(uint8 TrimValue)

Description: Sets the ILO trim registers with trim specified in TrimValue

Parameters: TrimValue - which should contain the concatenated 4-bit fine trim and 3-bit coarse trim value to be written to the ILO trim registers.

Return Value: None

Side Effects: None

void kHz_ILO_Trim_Sleep(void)

Description: Prepares the component for sleep by disabling the CorrectionISR and putting the Timer to sleep.

Parameters: None

Return Value: None

Side Effects: Stops CorrectionISR and sets the isrOff global variable.

void kHz_ILO_Trim_Wakeup(void)

Description: Wakes up the Timer and Enables the CorrectionISR.

Parameters: None

Return Value: None

Side Effects: None

DC and AC Electrical Characteristics

AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	ILO frequency			1		kHz
	ILO accuracy wrt reference clock			6.5*		%

* Not characterized

Reference Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes
1.0	Initial design.
1.10	Added CheckError, RestoreTrim APIs.
1.20	Added side-effect of Sleep API to datasheet
1.30	Added StopTrimming and ClearFIFO APIs, updated components for PSoC Creator 2.1 SP1, PSoC 5LP
1.40	Added GetTrim, SetTrim APIs, updated Component for PSoC Creator 2.2.

© Cypress Semiconductor Corporation, 2013. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark , and PSoC Creator™, and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.