iCE40 Oscillator Usage Guide

June 2016 **Technical Note TN1296**

Introduction

The iCE40™ family, specifically iCE40 Ultra™, iCE40 UltraLite™ and iCE40 UltraPlus™, features two on-chip oscillators. An ultra-low power 10 kHz oscillator is provided for Always-On applications and background polling that allow higher power processors to remain in power-down or sleep mode, conserving overall power consumption. A low power 48 MHz oscillator with output divider is provided for sensor management and pre-processing functions. These oscillators are intended for general clocking of internal logic and state machines.

Key Features

Two oscillators are available to users:

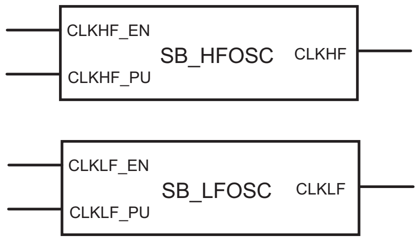
- SB_LFOSC – Low Frequency Oscillator
- SB_HFOSC – High Frequency Oscillator with output divider

On-Chip Oscillator Overview

You can access the two modules: SB_LFOSC and SB_HSOSC with enabled inputs and which you can dynamically control as shown in Figure 1.

SB_LFOSC runs at 10 kHz and SB_HFOSC runs at maximum 48 MHz with output divider by 1, 2, 4 or 8. SB_LFOSC and SB_HFOSC provide internal clock sources to user designs. These clocks can directly route to the global clock network or to local fabric.

Figure 1. On-Chip Oscillator



I/O Port Description

Table 1. SB_HFOSC I/O

Pin Name	Pin Direction	Description
CLKHF_EN	I	Enabling CLKHF output to be oscillating. This does not stop the oscillator, but only disables the output.
CLKHF	O	Oscillator Clock Output.
CLKHF_PU	I	Powering up the SB_HFOSC.

Table 2. SB_LFOSC

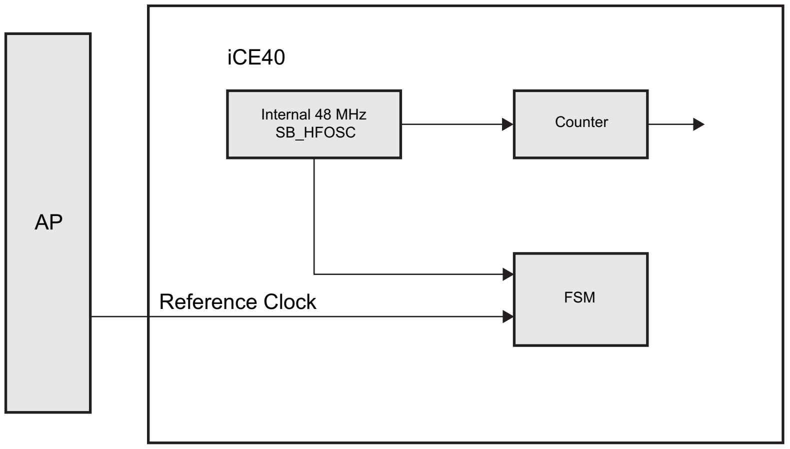
Pin Name	Pin Direction	Description
CLKLF_EN	I	Enabling CLKLF output to be oscillating. This does not stop the oscillator, but only disables the output
CLKLF	O	Oscillator clock output
CLKLF_PU	I	Powering up the SB_LFOSC

Connectivity Guideline

The SB_HFOSC and SB_LFOSC can be used as clock source. Their outputs are available for the user. They should be connected to the global clock network or local fabric. By default, the outputs will be routed to global clock network. To route to local fabric, please see the examples in the Appendix: Design Entry section.

Note that Oscillator cannot provide accurate frequency. For applications that require more accuracy, it is recommended to use calibration circuit to support the oscillator used as clock source. Figure 2 shows an example of the use of a reference clock that is only temporarily available for calibration.

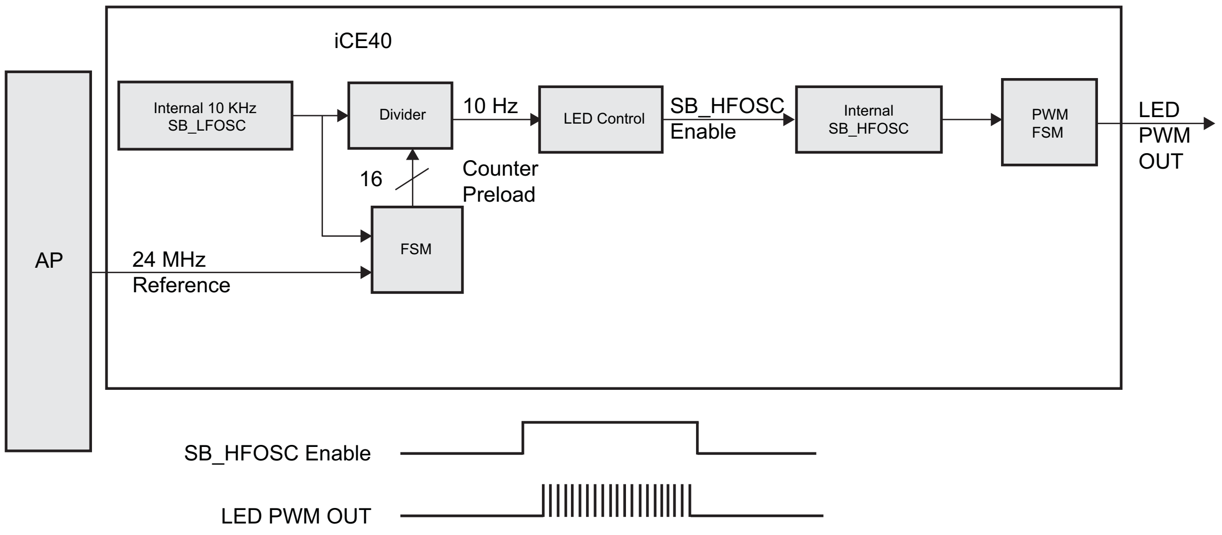
Figure 2. Oscillator Calibration Example



The calibration circuit for Oscillator can be improved for the purpose of power saving as shown in Figure 3.

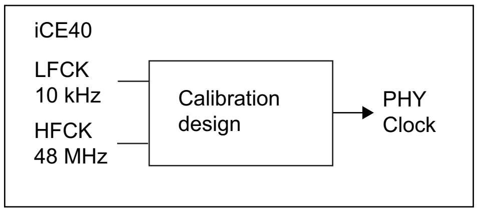
In this example, 10 kHz oscillator is always on. Calibrated divider provides timing for LED on-off. When LED is on, SB_LFOSC Enable turns on 48 MHz oscillator (SB_HFOSC turns on in two cycles). PWM FSM provides accurate PWM for LED. Power benefit is 48 MHz only when LED is on and minimum power when LED is off.

Figure 3. Oscillator Used for Dynamic Clock Calibration That Can Be Used On Service LED



For applications that need clocks within + –10% tolerance, such as USB Type-C PHY, a compensated clock generation logic must be used. An example clock generation logic for generating compensated 600 kHz clock is shown in Figure 4. The target clock is generated by dividing the 48 MHz high speed clock by a variable divisor. The value of the divisor is dynamically calculated from the ratio of the high speed clock (48 MHz) frequency to the low speed clock (10 kHz) frequency. The circuit described here uses some approximations to the calculations to achieve lower LUT utilization. The 600 kHz output clock from this circuit is expected to be within + –10% range as the 10 kHz slow speed clock is known to have under + –10% tolerance.

Figure 4. Example Clock Generation Logic



Please refer to UGxx for further details of this clock calibration design.

Power Management Options

When disabled, the SB_LFOSC and SB_HFOSC are in standby mode by default and consume only DC leakage. It is suggested to always enable SB_LFOSC and enable SB_HFOSC after there is an activity detected and the products return to full power mode for data analysis/processing.

Technical Support Assistance

Submit a technical support case via www.latticesemi.com/techsupport.

Revision History

Date	Version	Change Summary
June 2016	1.3	Updated Introduction section. Added iCE40 UltraPlus.
April 2015	1.2	Updated Connectivity Guideline section. Added example of clock generation logic.
		Updated Technical Support Assistance section.
January 2015	1.1	Added support for iCE40 UltraLite.
June 2014	01.0	Initial release.

Appendix: Design Entry

The following examples illustrate SB_HFOSC and SB_LFOSC usage with VHDL and Verilog.

SB_LFOSC Usage with Verilog

Synthesis Attributes

```
/* synthesis ROUTE_THROUGH_FABRIC = <value> */
```

Value:

0: Use dedicated clock network. Default option.

1: Use fabric routes.

Verilog Instantiation

```
SB_LFOSC OSCInst1 (
.CLKLF_EN(ENCLKLF),
.CLKLF_PU(CLKLF_POWERUP),
.CLKLF(CLKLF)
) /* synthesis ROUTE_THROUGH_FABRIC= [0|1] */;
```

SB_HFOSC Usage with Verilog

Synthesis Attributes

```
/* synthesis ROUTE_THROUGH_FABRIC = <value> */
```

Value:

0: Use dedicated clock network. Default option.

1: Use fabric routes.

Parameter Values

The SB_HFOSC primitive contains the following parameter and their default values:

Parameter CLKHF_DIV = 2'b00 : 00 = div1, 01 = div2, 10 = div4, 11 = div8 ; Default = “00”

Verilog Instantiation

```
SB_HFOSC OSCInst0 (

.CLKHF_EN(ENCLKHF),

.CLKHF_PU(CLKHF_POWERUP),

.CLKHF(CLKHF)

) /* synthesis ROUTE_THROUGH_FABRIC= [0|1] */;

Defparam OSCInst0.CLKHF_DIV = 2`b00;
```