iCE40LM On-Chip Strobe Generator Usage Guide

January 2014 Technical Note TN1275

Introduction

The iCE40LM family is an ultra-low power FPGA and sensor manager designed for ultra-low power mobile applications, such as smartphones, tablets and handheld devices. The iCE40LM family of devices includes integrated SPI and I^2C blocks to interface with virtually all mobile sensors and application processors.

An ultra-low power 10KHz strobe generator 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 12MHz strobe generator is provided for sensor management and pre-processing functions. These generators are intended for general clocking of internal logic and state machines.

Key Features

Two strobe generators are available to users:

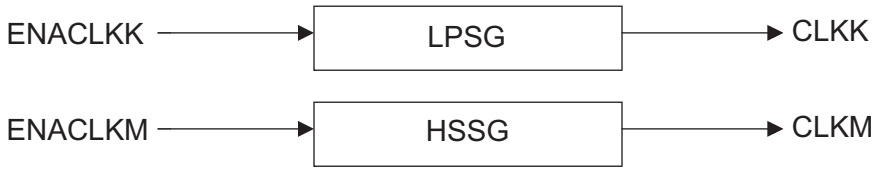
- LPSG – Low Power Strobe Generator
- HSSG – High Speed Strobe Generator

On-Chip Strobe Generator Overview

You can access the two modules: LPSG and HSSG with enabled inputs and which you can dynamically control as shown in Figure 20-1.

LPSG runs at 10KHz and HSSG runs at 12MHz. LPSG and HSSG provide internal clock sources to user designs. These clocks can directly route to the global clock network or to local fabric.

Figure 20-1. On-Chip Strobe Generator



I/O Port Description

Table 20-1. LPSG I/O

Pin Name	Pin Direction	Description
ENACLKK	I	Enable LPSG
CLKK	O	LPSG Clock Output.

Table 20-2. HSSG I/O

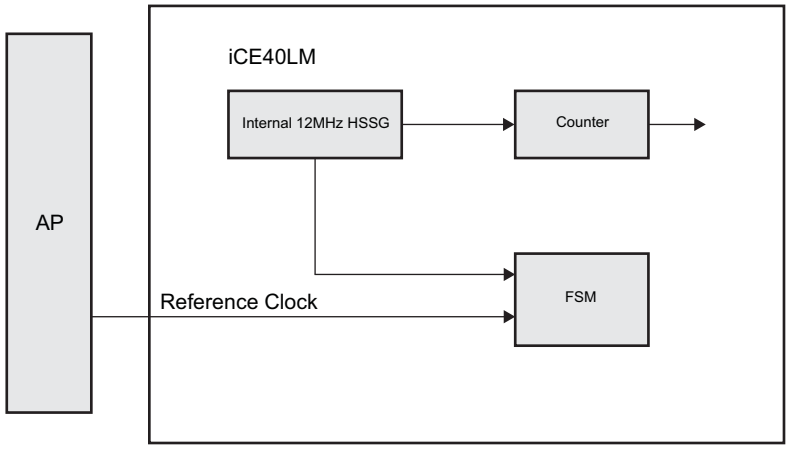
Pin Name	Pin Direction	Description
ENACLKM	I	Enable HSSG
CLKM	O	HSSG Clock Output.

Connectivity Guideline

The LPSG and HSSG 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.

Note that Strobe Generator cannot provide accurate frequency. For applications that require more accuracy, it is recommended to use calibration circuit to support the strobe generator used as clock source. Figure 20-2 shows an example of the use of a reference clock that is only temporarily available for calibration.

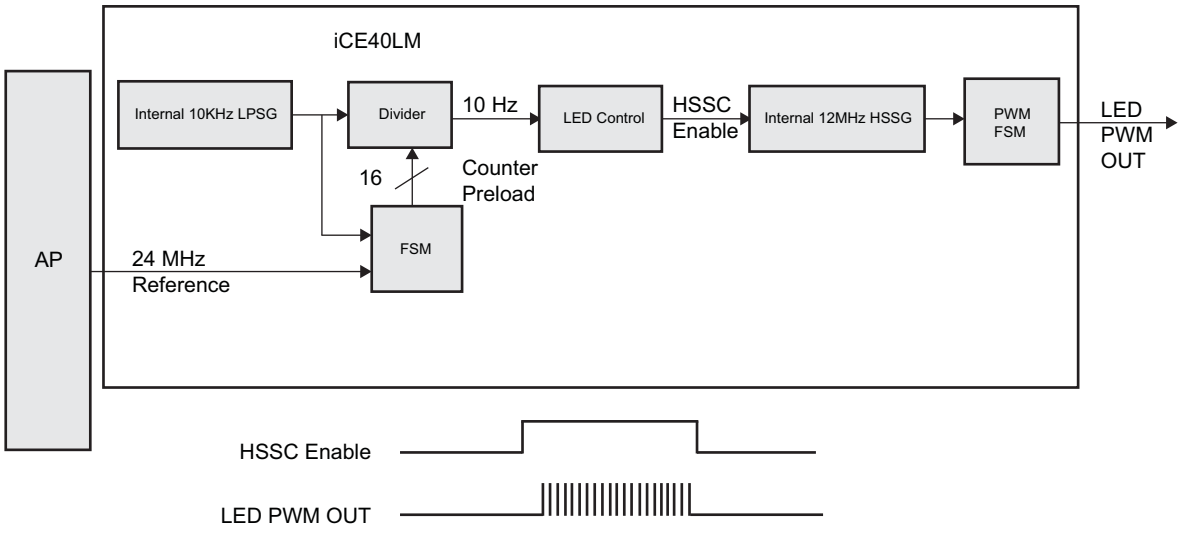
Figure 20-2. Strobe Generator Calibration Example



The calibration circuit for strobe generator can be improved for the purpose of power saving as shown in Figure 20-3.

In this example, 10KHz Strobe Generator is always on. Calibrated divider provides timing for LED on-off. When LED is on, LPSG Enable turns on 12MHz Strobe Generator (HSSG turns on in two cycles). PWM FSM provides accurate PWM for LED. Power benefit is12MHz only when LED is on and minimum power when LED is off

Figure 20-3. Strobe Generator Used for Dynamic Clock Calibration That Can Be Used On Service LED



Power Management Options

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

Technical Support Assistance

e-mail: techsupport@latticesemi.com

Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
October 2013	01.0	Initial release.
January 2014	01.1	Updated Appendix: Design Entry section.

Appendix: Design Entry

The following examples illustrate LPSG and HSSG usage with VHDL and Verilog.

LSOSC (LPSG) Usage with VHDL

```
COMPONENT SB_LSOSC
PORT (
  ENACLKK    : IN  std_logic;
  CLKK       : OUT std_logic);
END COMPONENT;

begin
OSCInst0: SB_LSOSC
PORT MAP (ENACLKK    => ENACLKK,
            CLKK     => CLKK);
```

LSOSC(LPSG) Usage with Verilog

```
module SB_LSOSC(ENACLKK, CLKK);

input  ENACLKK;
output CLKK;

SB_LSOSC OSCInst0 (.ENACLKK(ENACLKK),.CLKK(CLKK));

Endmodule
```

HSOSC(HSSG) Usage with VHDL

```
COMPONENT SB_HSOSC
PORT (
  ENACLKM    : IN  std_logic;
  CLKM       : OUT std_logic);
END COMPONENT;

begin
OSCInst0: SB_HSOSC
PORT MAP (ENACLKM   => ENACLKM,
            CLKM     => CLKM);
```

HSOSC(HSSG) Usage with Verilog (route through fabric)

```
module SB_HSOSC(ENACLKM, CLKM);

input ENACLKM;
output CLKM;

SB_HSOSC OSCInst0 (.ENACLKM(ENACLKM),.CLKM(CLKM))/* synthesis ROUTE_THROUGH_FABRIC=1 */;

endmodule
```