

## Introduction

The iCE40™ devices are SRAM-based FPGAs. The iCE40LP, iCE40HX, iCE40 Ultra™ and iCE40 UltraLite™ devices also have an on-chip, one-time programmable NVCM (Non-volatile Configuration Memory) to store configuration data. The SRAM memory cells are volatile, meaning that once power is removed from the device, its configuration is lost, and must be reloaded on the next power-up. This behavior has the advantage of being reprogrammable in the field which provides flexibility for products already deployed to the field. But it also requires that the configuration information be stored in a non-volatile device and loaded each time power is applied to the device. The on-chip NVCM allows the device to configure instantly and greatly enhances the design security by eliminating the need to use an external memory device. The configuration data can also be stored in an external SPI Flash from which the FPGA can configure itself upon power-up. This is useful for prototyping the FPGA or in situations where reconfigurability is required. Additionally, the device can be configured by a processor in an embedded environment.

**Table 1. iCE40 Devices Configuration Features Comparison**

Features	iCE40LM	iCE40LP/iCE40HX/iCE40 Ultra/iCE40 UltraLite
NVCM (one time programmable)	—	Yes
Multiple Configuration Image	—	Yes
Master SPI Configuration Mode	Yes	Yes
Slave SPI Configuration Mode	Yes	Yes

## Configuration Overview

The iCE40 devices contain two types of memory, SRAM and NVCM (one-time programmable). The iCE40LM devices, however, contain only one type of memory, which is SRAM. The SRAM memory contains the active configuration. The NVCM and the external SPI Flash provides a non-volatile storage for the configuration data. Additionally, the iCE40 configuration data can be downloaded from an external processor, microcontroller, or DSP processor using the SPI interface. In this document, the term “programming” refers to the programming of the NVCM and the term “configuration” refers to the configuration of SRAM memory. For either programming or configuration, the iCE40 FPGA utilizes the SPI configuration interface.

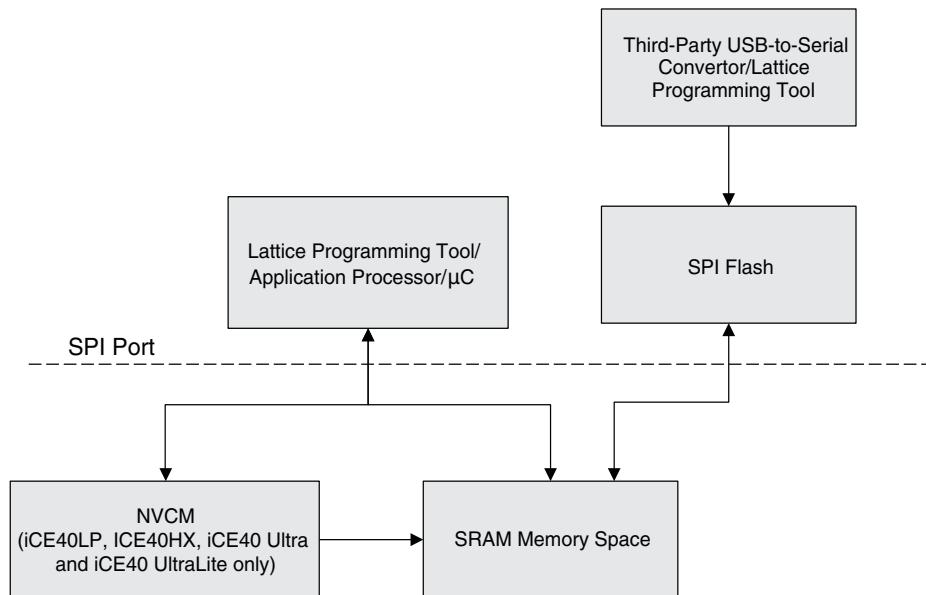
As described in Table 2, iCE40 components are configured for a specific application by loading a binary configuration bitstream image, generated by the Lattice development system. For high-volume applications, the bitstream image is usually permanently programmed in the on-chip Nonvolatile Configuration Memory (NVCM). However, the bitstream image can also be stored externally in a standard, low-cost commodity SPI serial Flash PROM. The iCE40 component can automatically load the image using the SPI Master Configuration Interface. Similarly, the iCE40 configuration data can be downloaded from an external processor, microcontroller, or DSP processor using a SPI-like serial interface.

**Table 2. iCE40 Device Configuration Modes**

Mode	Analogy	Configuration Data Source
NVCM <sup>1</sup>	ASIC	Internal, lowest-cost, secure, one-time programmable Nonvolatile Configuration Memory (NVCM).
Master SPI	Microprocessor	External, low-cost, commodity, SPI serial Flash PROM.
Slave SPI	Processor Peripheral	Configured by external device, such as a processor, microcontroller, or DSP using practically any data source, such as system Flash, a disk image, or over a network connection.

1. iCE40LP, iCE40HX, iCE40 Ultra and iCE40 UltraLite devices only.

Figure 1 provides an overview of the configuration and programming of the iCE40 FPGA. For configuration and programming, the device can be accessed using the SPI interface/protocol described in later sections of this technical note. The SRAM can configure itself (device in master mode) from the on-chip NVCM (iCE40LP, ICE40HX, iCE40 Ultra, and iCE40 Ultra Lite devices only), external SPI Flash or Lattice programming hardware. An external processor or programming hardware can also configure the SRAM with the FPGA in slave SPI mode. The NVCM can be programmed using the Lattice Diamond® Programmer (version 2.2 or later) or an external processor.

**Figure 1. Configuring and Programming the iCE40 Device**


Below is a standard configuration procedure:

1. At the beginning of the configuration, all the SRAM has been cleared to “0”, so Pullup is enabled by SRAM bit “0” and IO HiZ is enabled by configuration controller.
2. In the process of bitstream downloading, all the SRAM bits will be replaced with the fresh SRAM bits gradually. And the new SRAM bits will take effect immediately.
3. At the end of configuration, configuration controller will release IO (except configuration SPI pins) tristate and IO becomes active after CDONE goes high (17 clocks), only 4 configuration SPI pins will become user IO after 49 clocks.

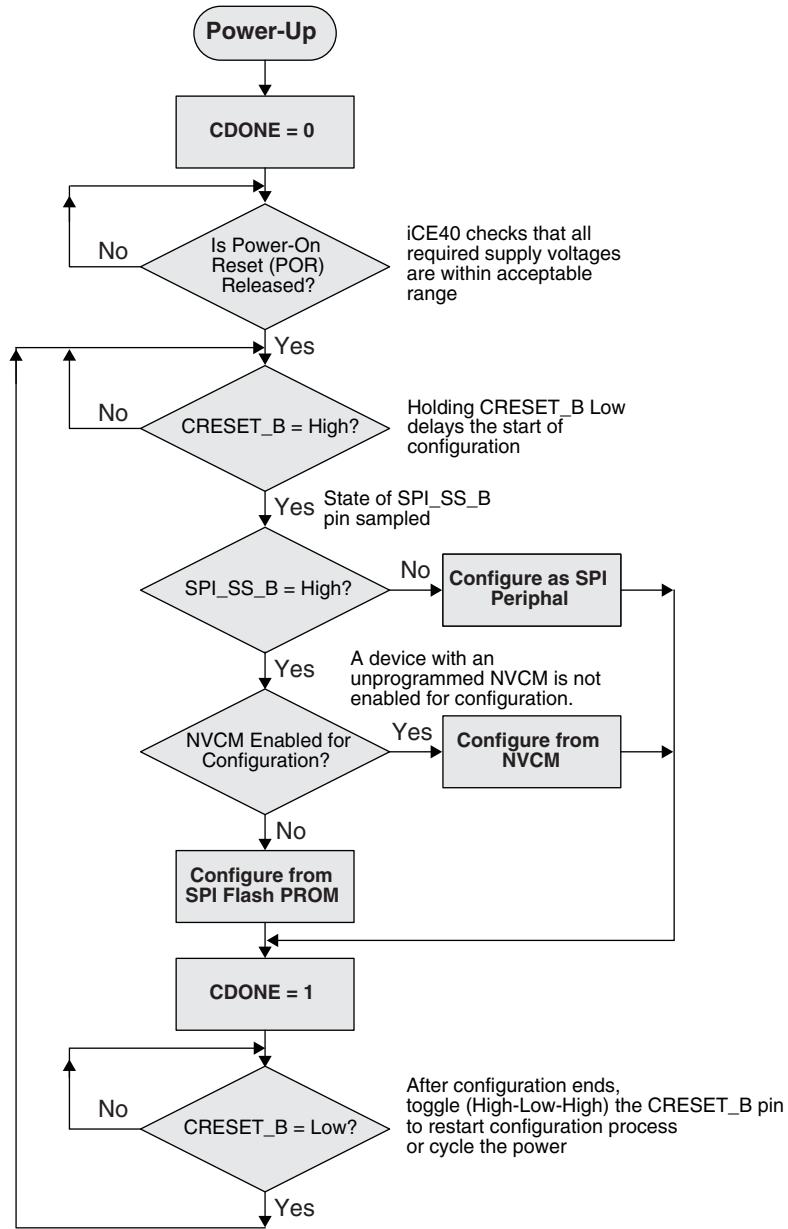
## Configuration Mode Selection

### Mode Selection for iCE40LP/HX, iCE40 Ultra and iCE40 UltraLite Devices

The iCE40LP/HX, iCE40 Ultra and iCE40 UltraLite device configuration mode is selected according to the following priority described below and illustrated in Figure 2.

- After exiting the Power-On Reset (POR) state or when CRESET\_B returns High after being held Low, the iCE40 device samples the logical value on its SPI\_SS\_B pin. Like other programmable I/O pins, the SPI\_SS\_B pin has an internal pull-up resistor. Refer to the iCE40LP/HX Family Data Sheet for the minimum pulse width requirement of CRESET\_B.
- If the SPI\_SS\_B pin is sampled as a logic ‘1’ (High), then ...
  - Check if the device is enabled to configure from the Nonvolatile Configuration Memory (NVCM). If the NVCM is programmed, the device will configure from NVCM.
    - If enabled to configure from NVCM, the device configures itself using the Nonvolatile Configuration Memory (NVCM).
    - If not enabled to configure from NVCM, then the device configures using the SPI Master Configuration Interface.
- If the SPI\_SS\_B pin is sampled as a logic ‘0’ (Low), then the device waits to be configured from an external controller or from another device in SPI Master Configuration Mode using an SPI-like interface.

Figure 2. iCE40 Device Configuration Control Flow

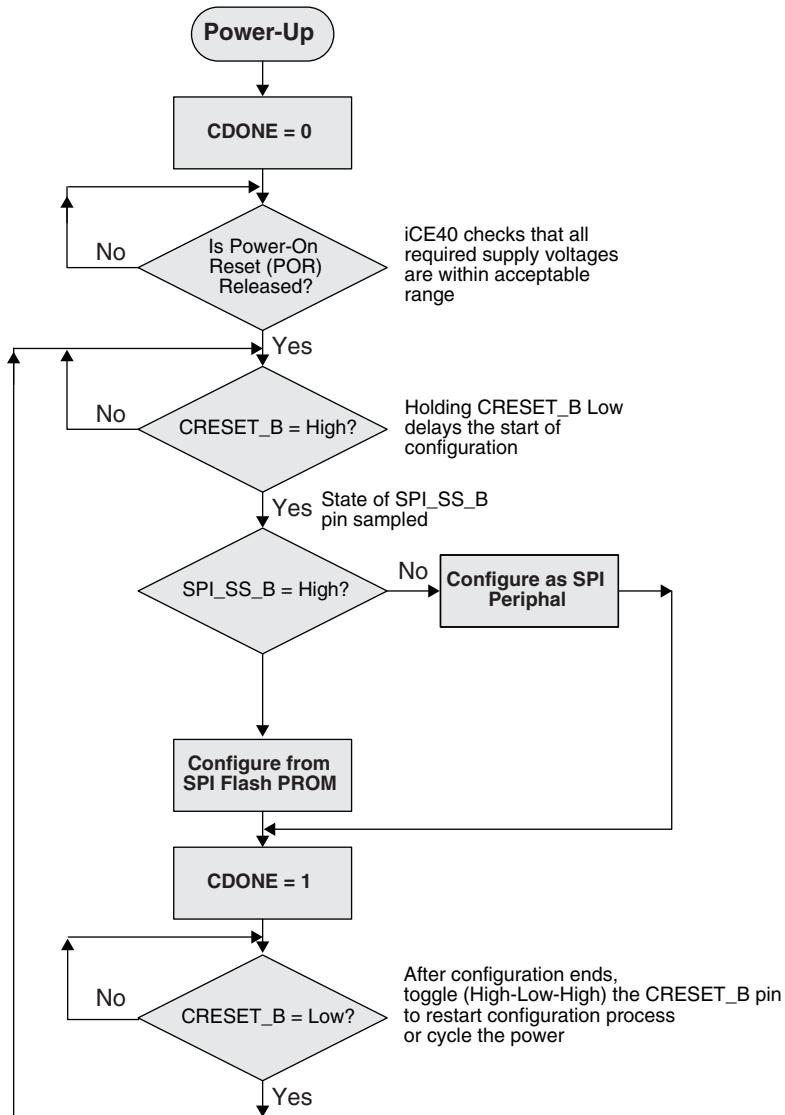


## Mode Selection for iCE40LM

The iCE40LM configuration mode is selected according to the following priority described below and illustrated in Figure 3.

- After exiting the Power-On Reset (POR) state or when CRESET\_B returns High after being held Low, the device samples the logical value on its SPI\_SS\_B pin. Like other programmable I/O pins, the SPI\_SS\_B pin has an internal pull-up resistor. Refer to the iCE40LM Family Data Sheet for the minimum pulse width requirement of CRESET\_B.
- If the SPI\_SS\_B pin is sampled as a logic '1' (High), then the device configures using the SPI Master Configuration Interface.
- If the SPI\_SS\_B pin is sampled as a logic '0' (Low), then the device waits to be configured from an external controller or from another device in SPI Master Configuration Mode using an SPI-like interface

**Figure 3. iCE40LM Device Configuration Control Flow**



## Nonvolatile Configuration Memory (NVCM) (*Applies to iCE40LP, iCE40HX, iCE40 Ultra and iCE40 UltraLite Devices Only*)

All standard iCE40 devices have an internal, nonvolatile configuration memory (NVCM). The NVCM is large enough to program a complete iCE40 device, including initializing all Embedded Block RAM. The NVCM memory also has very high programming yield due to extensive error checking and correction (ECC) circuitry.

The NVCM is ideal for cost-sensitive, high-volume production applications, saving the cost and board space associated with an external configuration PROM. Furthermore, the NVCM provides exceptional design security, protecting critical intellectual property (IP). The NVCM contents are entirely contained within the iCE40 device and are not readable once protected by the one-time programmable Security bits. Furthermore, there is no observable difference between a programmed or un-programmed memory cell using optical or electron microscopy.

The NVCM memory has a programming interface similar to a 25-series SPI serial Flash PROM. Consequently, it can be programmed using Diamond Programmer (version 2.2 or later) before or after circuit board assembly or programmed in-system from a microprocessor or other intelligent controller. The NVCM can also be pre-programmed at the factory. Contact Lattice Technical Support or your local Lattice sales office for assistance.

### NVCM Programming

The NVCM can be programmed in the following ways:

- Diamond Programmer
  - Programming using the Diamond Programmer (Diamond 2.0.1 or later) is recommended for prototyping. Programming is supported using the Lattice programming cable. For more information refer to the Diamond Programmer Online Help and UG48, [Programming Cables User's Guide](#).
- Factory Programming
  - The Lattice factory offers NVCM programming. For more information contact your local Lattice sales office.
- Embedded Programming

The NVCM can be programmed using a processor. For more information contact your local Lattice sales office.

### Configuration Control Signals

The iCE40 configuration process is self-timed and controlled by a few internal signals and device I/O pins, as described in Table 3.

**Table 3. iCE40 Configuration Control Signals**

Signal Name	Direction	Description
POR	Internal Control	Internal Power-On Reset (POR) circuit.
OSC	Internal Control	Internal configuration oscillator.
CRESET_B	Input	Configuration Reset input. Active-Low. No internal pull-up resistor.
CDONE <sup>1</sup>	Open-drain Output	Configuration Done output. Permanent, weak pull-up resistor to VCCIO_2.

1. The iCE40-1KLP SWG16 package CDONE pin can be used as a user output. Please see iCE40LP/HX Family Data Sheet for details.

The Power-On Reset circuit, POR, automatically resets the iCE40 component to a known state during power-up (cold boot). The POR circuit monitors the relevant voltage supply inputs, as shown in Figure 5. Once all supplies exceed their minimum thresholds, the configuration controller can start the configuration process.

The configuration controller begins configuring the iCE40 device, clocked by the Internal Oscillator, OSC. The OSC oscillator continues controlling configuration unless the iCE40 device is configured using the SPI Peripheral Configuration Interface.

**Figure 4. iCE40 Configuration Control Pins**

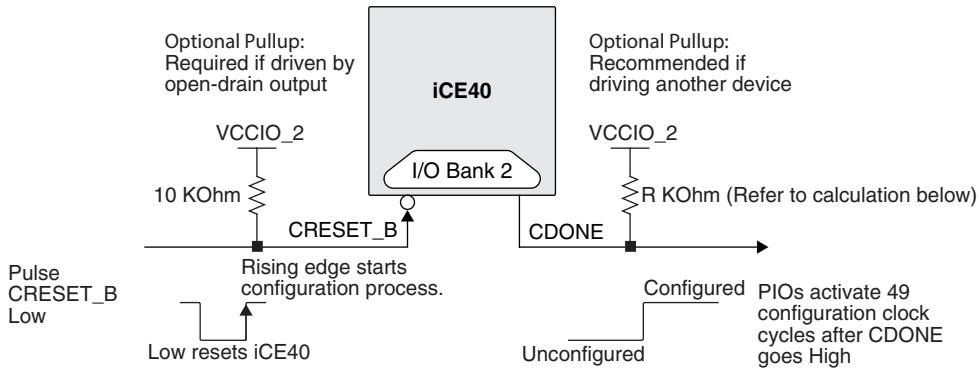


Figure 4 shows the two iCE40 configuration control pins, **CRESET\_B** and **CDONE**. When driven Low, the dedicated Configuration Reset input, **CRESET\_B**, resets the iCE40 device. When **CRESET\_B** returns High, the iCE40 FPGA restarts the configuration process from its power-on conditions (Cold Boot). The **CRESET\_B** pin is a pure input with no internal pull-up resistor. If driven by open-drain driver or un-driven, then connect the **CRESET\_B** pin to a 10 kOhm pull-up resistor connected to the **VCCIO\_2** supply.

The iCE40 device signals the end of the configuration process by actively turning off the internal pull-down transistor on the Configuration Done output pin, **CDONE**. The pin has a permanent, weak internal pull-up resistor to the **VCCIO\_2** rail. However, depending on the system capacitance and configuration frequency, the **CDONE** pin must be tied to an external pullup resistor connected to the **VCCIO\_2** supply. The maximum resistor size can be calculated knowing the configuration clock frequency (SCLK or MCLK) and the **CDONE** trace capacitance with the following formula:

$$R_{pullup} = 1/(2 * ConfigFrequency * CDONETraceCap)$$

The PIO pins activate according to their configured function after 49 configuration clock cycles. The internal oscillator is the configuration clock source for the SPI Master Configuration Interface and when configuring from Nonvolatile Configuration Memory (NVCM). When using the SPI Peripheral Configuration Interface, the configuration clock source is the SPI\_SCK clock input pin.

## Internal Oscillator

During SPI Master or NVCM (iCE40LP, iCE40HX, iCE40 Ultra and iCE40 UltraLite only) configuration mode, the controlling clock signal is generated from an internal oscillator. The oscillator starts operating at the default frequency. During the configuration process, however, bit settings within the configuration bitstream can specify a higher-frequency mode in order to maximize SPI bandwidth and reduce overall configuration time. See data sheet for the specified oscillator frequency range.

Using the SPI Master Configuration Interface, internal oscillator controls all the interface timing and clocks the SPI serial Flash PROM via the SPI\_SCK clock output pin.

The oscillator output, which also supplies the SPI SCK clock output during the SPI Flash configuration process, has a 50% duty cycle.

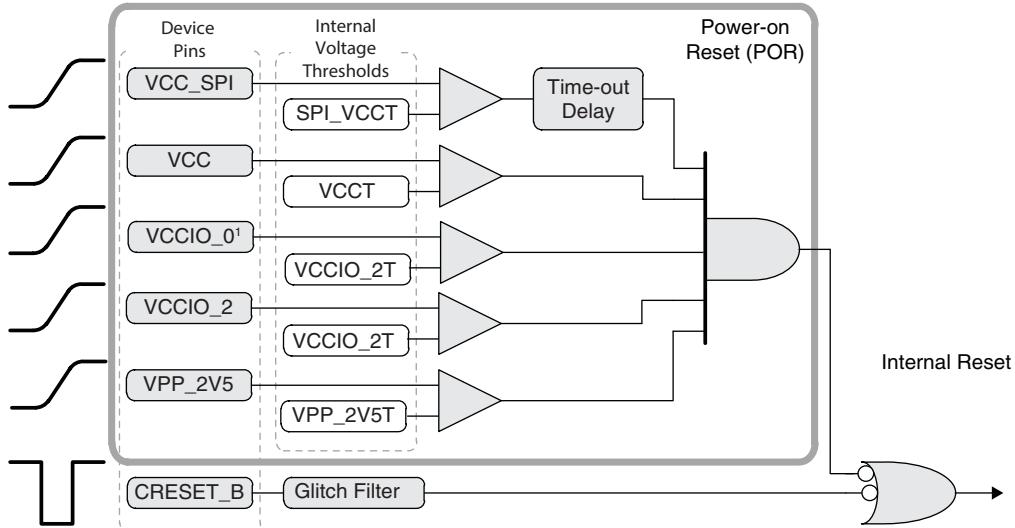
The Oscillator settings can be found in the iCEcube™ software by selecting the **Tools > Tool Options** pull down menu and then the **Bitstream** tab.

## Internal Device Reset

Figure 5 presents the various signals that internally reset the iCE40 internal logic.

- Power-On Reset (POR)
- CRESET\_B Pin

**Figure 5. iCE40 Internal Reset Circuitry**



<sup>1</sup> Applicable for iCE40LM only.

### Power-On Reset (POR)

The Power-on Reset (POR) circuit monitors specific voltage supply inputs and holds the device in reset until all the relevant supplies exceed the internal voltage thresholds. The VCC\_SPI supply also has an additional time-out delay to allow an attached SPI serial PROM to power up properly. Table 4 shows the POR supply inputs. The Non-volatile Configuration Memory (NVCM) requires that the VPP\_2V5 supply be connected, even if the application does not use the NVCM.

Notes:

- It is recommended that Bank1 should be powered before the last supply gating POR.
- All banks must be powered prior to configuration.

**Table 4. Power-on Reset (POR) Voltage Resources**

Supply Rail	iCE40 Production Devices
VCC	Yes
VCC_SPI	Yes
VCCIO_0 <sup>2</sup>	Yes
VCCIO_2	Yes
VPP_2V5 <sup>1</sup>	Yes

1. Only needed for iCE40LP, iCE40HX, iCE40 Ultra and iCE40 UltraLite devices.  
2. Only needed for iCE40LM device.

### CRESET\_B Pin

The CRESET\_B pin resets the iCE40 internal logic when Low.

## sysCONFIG Port

The sysCONFIG port is used to program and configure the iCE40 FPGA. The device has a SPI configuration interface as the sysCONFIG port which can be used to configure the device.

**Table 5. sysCONFIG Ports**

Interface	Port	Description
sysCONFIG	SPI Master Configuration interface	In this mode, the FPGA configures itself from an external SPI Flash or Diamond Programmer (version 2.2 or later). The FPGA behaves as master, generates internal clock and drives the clock to the external SPI Flash.
	SPI Slave Configuration interface	In this mode, the FPGA behaves as a Slave device. An external Application Processor, µC or Diamond Programmer (version 2.2 or later) configures or programs the device.

## sysCONFIG Pins

The iCE40 FPGA has a set of sysCONFIG pins that are used to program and configure the device. The sysCONFIG pins are grouped together to create the sysCONFIG port, as discussed above. The sysCONFIG pins are dual-function, meaning they can be recovered as user I/O after configuration is complete. Table 6 shows the sysCONFIG pins.

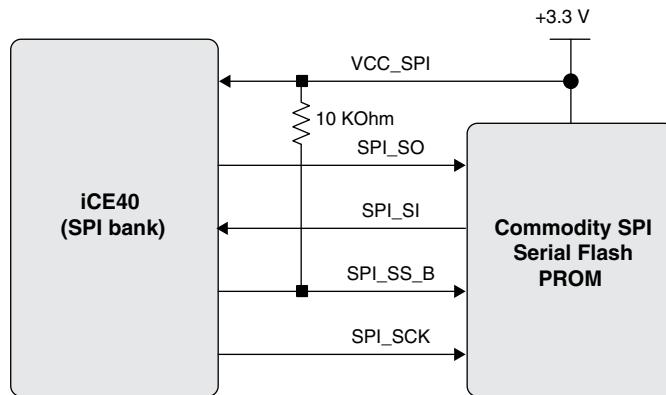
**Table 6. sysCONFIG Pins**

Pin Name	Associated sysCONFIG Port	Pin Direction	Description
CRESET_B	—	Input	Configuration Reset input, active-low. No internal pull-up resistor.
CDONE	—	Output	Configuration Done output. The pin has a permanent, weak internal pull-up resistor to the VCCIO_2 rail. Depending on the frequency of configuration and the capacitance on CDONE node, then CDONE pin must be tied to an external pullup resistor connected to the VCCIO_2 supply. The resistor size can be calculated knowing the configuration clock frequency (SCLK or MCLK) and the CDONE trace capacitance with the following formula: $R_{pullup}=1/(2*ConfigFrequency*CDONETraceCap)$ The iCE40-1KLP SWG16 package CDONE pin can be used as a user output.
SPI_SS_B	SPI Master/Slave configuration interface	Input/Output	An important dual-function, active-low slave select pin. After the device exits POR or CRESET_B is toggled (High-Low-High), it samples the SPI_SS_B to select the configuration mode (an output in Master mode and an input in Slave mode). iCE40LM devices have this pin shared with hardened SPI IP SPI1_CS pin.
SPI_SI	SPI Master/Slave configuration interface	Input/Output	A dual-function, serial input pin in both configuration modes. iCE40LM devices have this pin shared with hardened SPI IP SPI_MOSI pin.
SPI_SO	SPI Master/Slave configuration interface	Input/Output	A dual-function, serial output pin in both configuration modes. iCE40LM devices have this pin shared with hardened SPI IP SPI_MISO pin.
SPI_SCK	SPI Master/Slave configuration interface	Input/Output	A dual-function clock signal. An output in Master mode and input in Slave mode. iCE40LM devices have this pin shared with hardened SPI IP SPI_SCK pin.

## SPI Master Configuration Interface

All iCE40 devices can be configured from an external, commodity SPI serial Flash PROM, as shown in Figure 6. The SPI configuration interface is essentially its own independent I/O bank, powered by the VCC\_SPI supply input. Presently, most commercially-available SPI serial Flash PROMs require a 3.3V supply.

**Figure 6. iCE40 SPI Master Configuration Interface**



The SPI configuration interface is used primarily during development before mass production, where the configuration is then permanently programmed in the NVCM configuration memory (only available in iCE40LP, iCE40HX, iCE40 Ultra and iCE40 UltraLite devices). However, the SPI interface can also be the primary configuration interface allowing easy in-system upgrades and support for multiple configuration images.

The SPI control signals are defined in Table 7.

**Table 7. SPI Master Configuration Interface Pins (SPI\_SS\_B High Before Configuration)**

Signal Name	Direction	Description
VCC_SPI	Supply	SPI Flash PROM voltage supply input.
SPI_SO	Output	SPI Serial Output from the iCE40 device.
SPI_SI	Input	SPI Serial Input to the iCE40 device, driven by the select SPI serial Flash PROM.
SPI_SS_B	Output	SPI Slave Select output from the iCE40 device. Active Low.
SPI_SCK	Output	SPI Slave Clock output from the iCE40 device.

After configuration, the SPI port pins are available to the user-application as additional PIO pins, supplied by the VCC\_SPI input voltage.

## SPI PROM Requirements

The iCE40 SPI Flash configuration interface supports a variety of SPI Flash memory vendors and product families. However, Lattice does not specifically test, qualify, or otherwise endorse any specific SPI Flash vendor or product family. The iCE40 SPI interface supports SPI PROMs that they meet the following requirements.

- The PROM must operate between 1.8V to 3.3V in order to trigger the iCE40 FPGA's power-on reset circuit.
- The PROM must support the 0x0B Fast Read command, using a 24-bit start address and has 8 dummy bits before the PROM provides first data (see Figure 8).
- The PROM must have enough bits to program the iCE40 device.
- The PROM must support data operations at the upper frequency range for the selected iCE40 internal oscillator frequency (see data sheet). The oscillator frequency is selectable when creating the FPGA bitstream image.

- For lowest possible power consumption after configuration, the PROM should also support the 0xB9 Deep Power Down command and the 0xAB Release from Deep Power-down Command (see Figure 7 and Figure 9). The low-power mode is optional. The PROM must be powered and ready to be accessed following iCE40 POR
- The PROM must be ready to accept commands 10  $\mu$ s after meeting its power-on conditions. In the PROM data sheet, this may be specified as  $t_{VSL}$  or  $t_{VCSL}$ . It is possible to use slower PROMs by holding the CRESET\_B input Low until the PROM is ready, then releasing CRESET\_B, either under program control or using an external power-on reset circuit.

The iCEblink40™ development board and associated programming software uses an ST Micro/Numonyx M25Pxx SPI serial Flash PROM. Table 8 gives the bitstream sizes for different densities of the iCE40 FPGA that can be used to select a SPI Flash.

**Table 8. Bitstream Sizes for Different iCE40 FPGA Densities Used to Select a SPI Flash**

Device	Bytes	Bits
iCE40-LP384	7872	62,976
iCE40-LP/HX1K	34,112	272,896
iCE40-LP/HX4K	136,448	1,091,584
iCE40-LP/HX8K	136,448	1,091,584
iCE40LP640	34,112	272,896
iCE40LM1K	34,112	272,896
iCE40LM2K	68,224	545,792
iCE40LM4K	68,224	545,792
iCE5LP1K	71,255	570,040
iCE5LP2K	71,255	570,040
iCE5LP4K	71,255	570,040

## Enabling SPI Configuration Interface

To enable the SPI configuration mode, the SPI\_SS\_B pin must be allowed to float High. The SPI\_SS\_B pin has an internal pull-up resistor. If SPI\_SS\_B is Low, then the iCE40 component defaults to the SPI Slave configuration mode.

## SPI Master Configuration Process

The iCE40 SPI Master Configuration Interface supports a variety of modern, high-density, low-cost SPI serial Flash PROMs. Most modern SPI PROMs include a power-saving Deep Power-down mode. The iCE40 component exploits this mode for additional system power savings.

The iCE40 SPI interface starts by driving SPI\_SS\_B Low, and then sends a Release from Power-down command to the SPI PROM, hexadecimal command code 0xAB. Figure 7 provides an example waveform. This initial command wakes up the SPI PROM if it is already in Deep Power-down mode. If the PROM is not in Deep Power-down mode, the extra command has no adverse affect other than that it requires a few additional microseconds during the configuration process. The iCE40 device transmits data on the SPI\_SO output, on the falling edge of the SPI\_SCK output. The SPI PROM does not provide any data to the iCE40 device's SPI\_SI input. After sending the last command bit, the iCE40 device de-asserts SPI\_SS\_B High, completing the command. The iCE40 device then waits a minimum of 10  $\mu$ s before sending the next SPI PROM command.

**Figure 7. SPI Release from Deep Power-down Command**

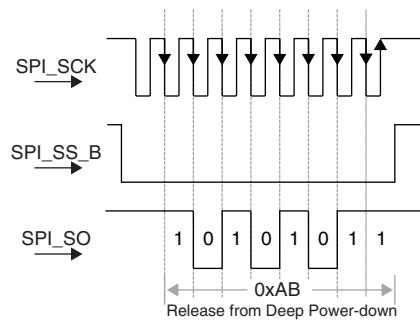
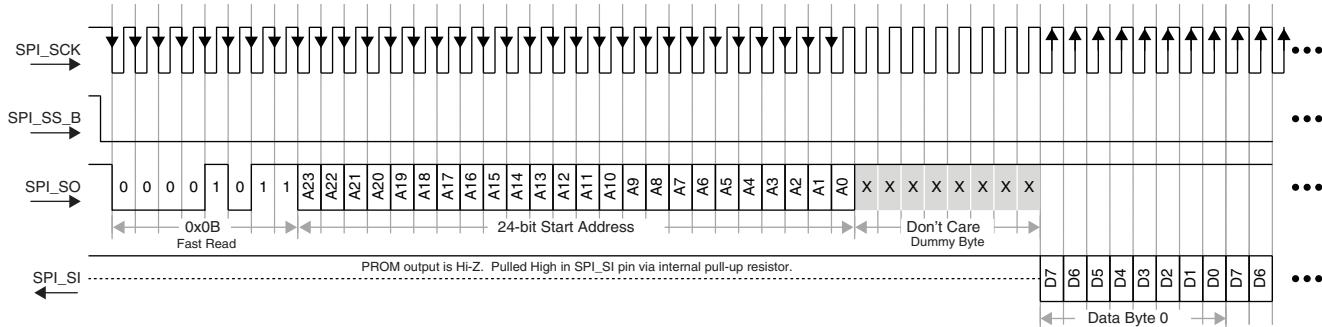


Figure 8 illustrates the next command issued by the iCE40 device. The iCE40 SPI interface again drives SPI\_SS\_B Low, followed by a Fast Read command, hexadecimal command code 0x0B, followed by a 24-bit start address, transmitted on the SPI\_SO output. The iCE40 device provides data on the falling edge of SPI\_SS\_B. Upon initial power-up, the start address is always 0x00\_0000. After waiting eight additional clock cycles, the iCE40 device begins reading serial data from the SPI PROM. Before presenting data, the SPI PROM's serial data output is high-impedance. The SPI\_SI input pin has an internal pull-up resistor and sees high-impedance as logic '1'.

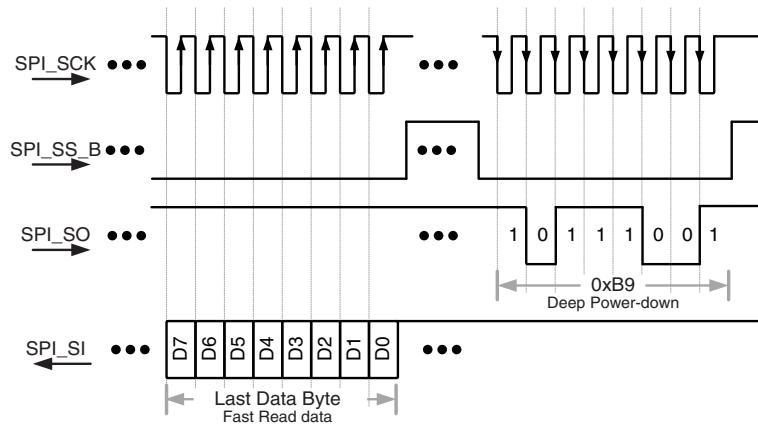
**Figure 8. SPI Fast Read Command**



The external SPI PROM supplies data on the falling edge of the iCE40 device's SPI\_SCK clock output. The iCE40 device captures each PROM data value on the SPI\_SI input, using the rising edge of the SPI\_SCK clock signal. The SPI PROM data starts at the 24-bit address presented by the iCE40 device. PROM data is serially output, byte by byte, with most-significant bit, D7, presented first. The PROM automatically increments an internal byte counter as long as the PROM is selected and clocked.

After transferring the required number configuration data bits, the iCE40 device ends the Fast Read command by de-asserting its SPI\_SS\_B PROM select output, as shown in Figure 9. To conserve power, the iCE40 device then optionally issues a final Deep Power-down command, hexadecimal command code 0xB9. After de-asserting the SPI\_SS\_B output, the SPI PROM enters its Deep Power-down mode. The final power-down step is optional; the application may wish to use the SPI PROM and can skip this step, controlled by a configuration option.

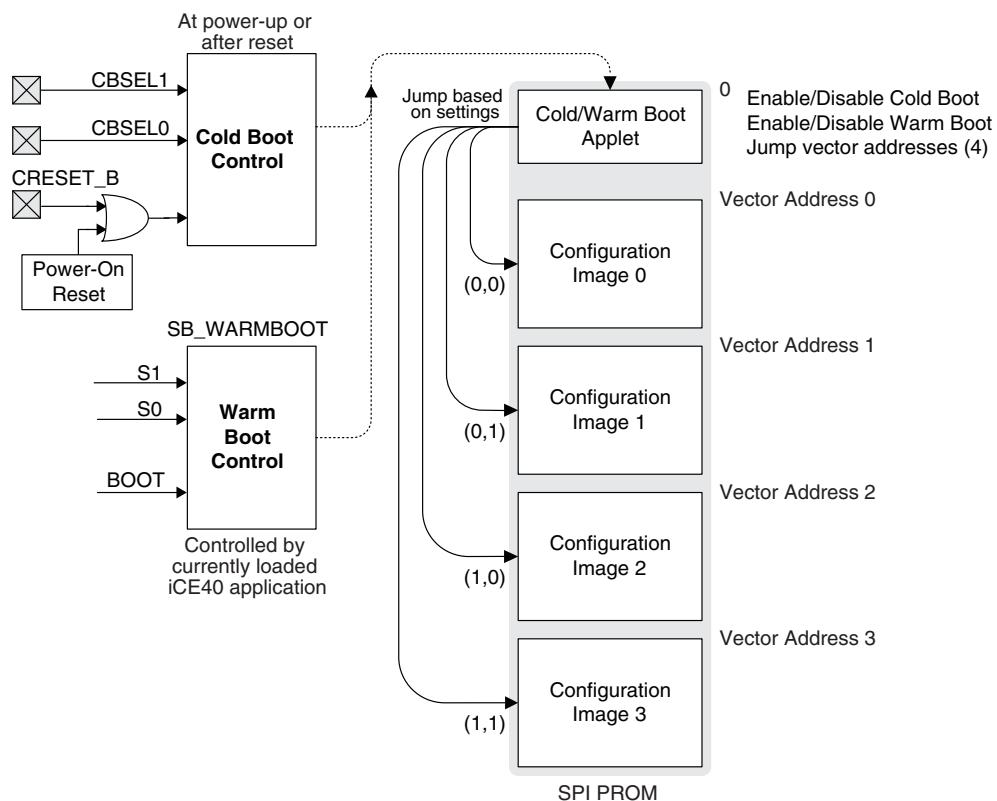
**Figure 9. Final Configuration Data, SPI Deep Power-down Command**



## Cold Boot Configuration Option (*Applies to iCE40LP and iCE40HX Devices Only*)

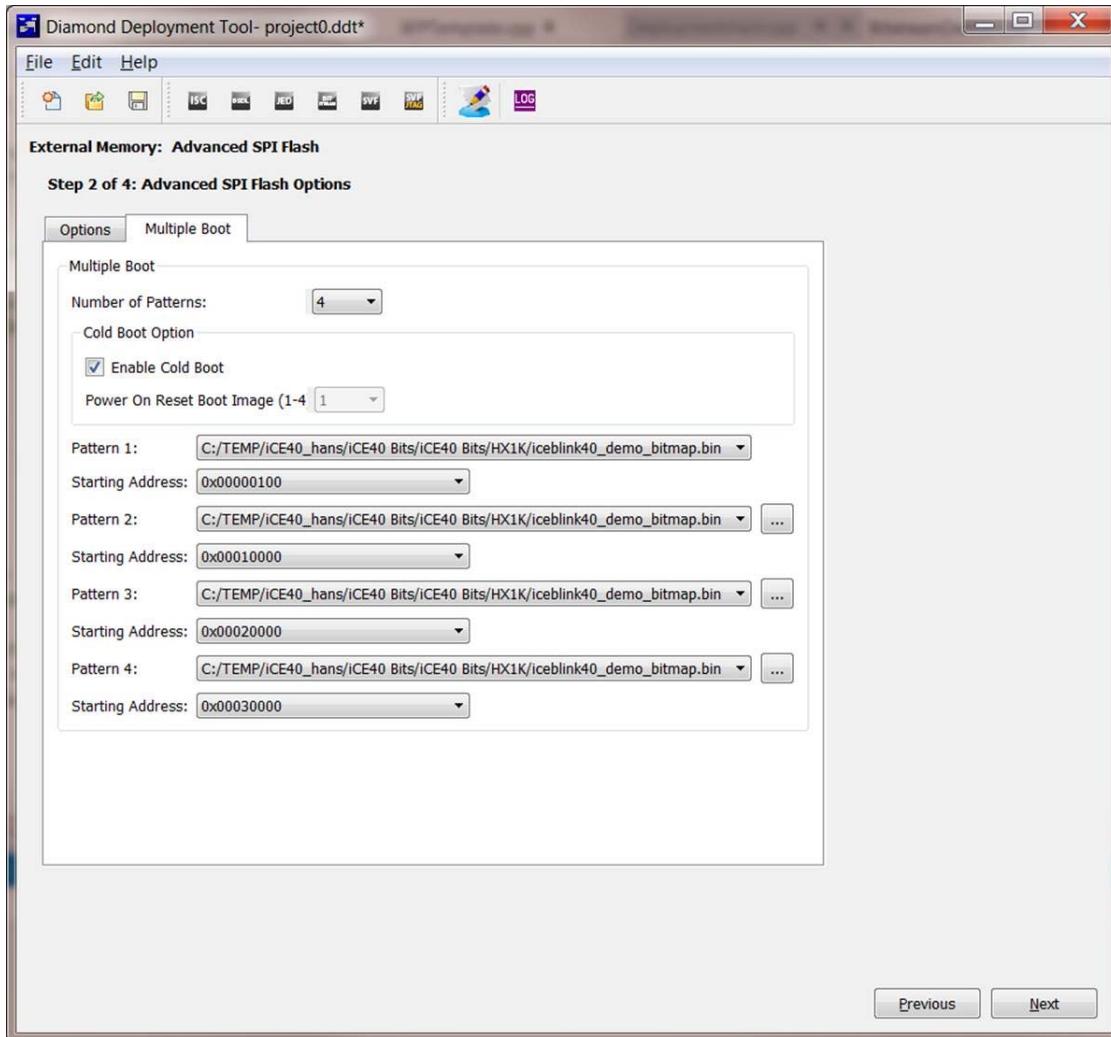
By default, the iCE40 FPGA is programmed with a single configuration image, either from internal NVCM memory, from an external SPI Flash PROM, or externally from a processor or microcontroller.

**Figure 10. Cold Boot and Warm Boot Configuration**



When self-loading from an SPI Flash PROM, the FPGA supports an additional configuration option called Cold Boot mode. This option can be implemented by creating an applet from Deployment Tool as shown in Figure 11. When this option is enabled in the applet, the iCE40 FPGA boots normally from power-on or a master reset (**CRESET\_B** = Low pulse), but monitors the value on two PIO pins that are borrowed during configuration, as shown in Figure 10. These pins, labeled **PIO2/CBSEL0** and **PIO2/CBSEL1**, tell the FPGA which of the four possible SPI configurations to load into the device.

**Figure 11. Diamond Deployment Tool - Advanced SPI Flash Options**



- For Cold Boot or Warm Boot applications, the initial configuration image must be stored at SPI Flash PROM and the cold boot/warm boot applet must be stored at SPI Flash PROM address 0.
- Check if Cold Boot configuration feature is enabled in the applet.
  - If Cold Boot is not enabled, the FPGA configuration starts from the default location (image 0) defined in the Warm Boot applet.
  - If Cold Boot is enabled, the FPGA reads the logic values on pins CBSEL[1:0]. The FPGA uses the value as a vector and then reads from the indicated vector address.
  - At the selected CBSEL[1:0] vector address, there is a starting address for the selected configuration image.
    - For SPI Flash PROMs, the new address is a 24-bit start address in Flash.

**Table 9. CBSEL[1:0] Vector Address**

CBSEL[1:0]	Vector Address
00	0
01	1
10	2
11	3

- Using the new start address, the FPGA restarts reading configuration memory from the new location.

When creating the initial configuration image, the Lattice development software loads the start address for up to four configuration images in the multiple combined bitstreams. The multiple combined bitstreams are stored in the SPI flash. The value on the CBSEL[1:0] pins tells the configuration controller to read a specific start address, then to load the configuration image stored at the selected address.

After configuration, the CBSEL[1:0] pins become normal PIO pins available to the application.

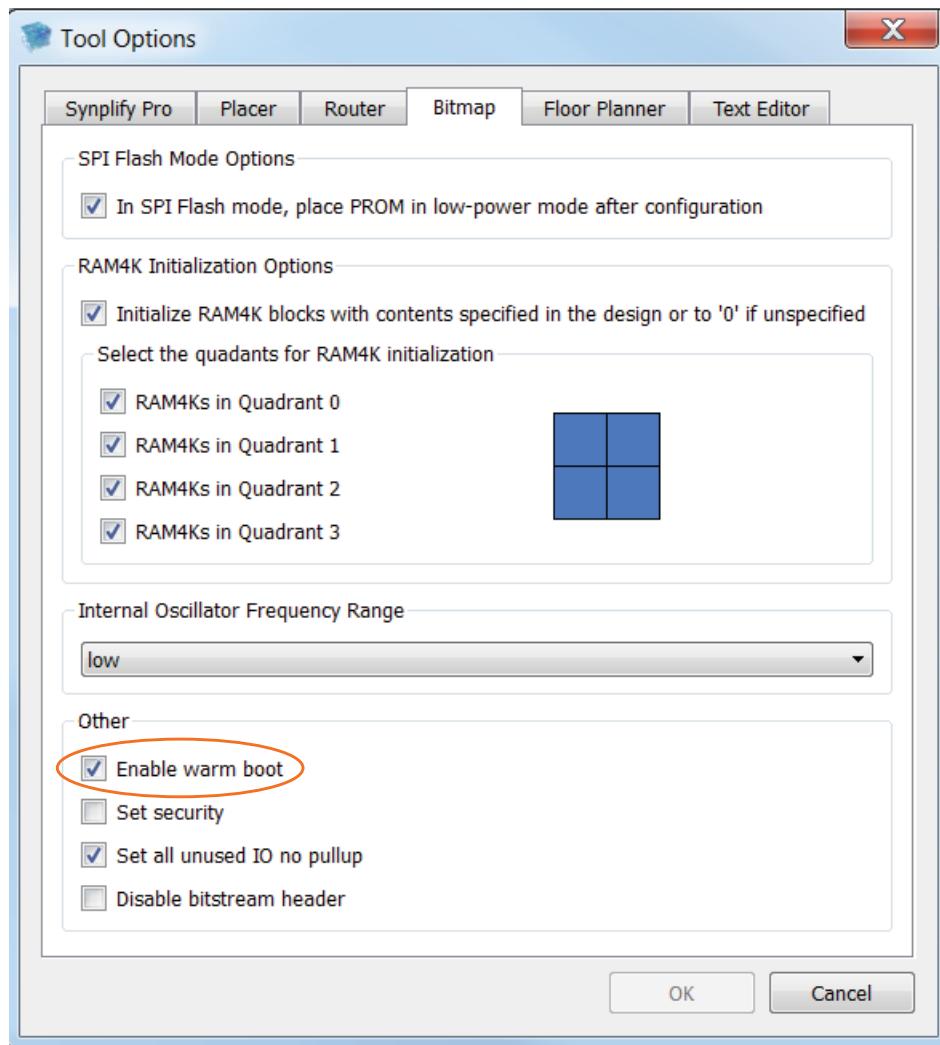
The Cold Boot feature allows the iCE40 to be reprogrammed for special application requirements such as the following.

- A normal operating mode and a self-test or diagnostics mode.
- Different applications based on switch settings.
- Different applications based on a card-slot ID number. Use external SPI Flash PROMs only.

## Warm Boot Configuration Option

The Warm Boot configuration is similar to the Cold Boot feature, but is completely under the control of the FPGA application. To enable Warm Boot, set **Enable warm boot** in Tool Options in Lattice iCEcube. Upon power up, the default image points to Image 0 stored in the SPI flash.

**Figure 12. Enable Warm Boot Option**



A special design primitive, SB\_WARMBOOT, allows an FPGA application to choose between four configuration images using two internal signal ports, S1 and S0, as shown in Figure 10. These internal signal ports connect to programmable interconnect, which in turn can connect to PLB logic and/or PIO pins. S1 and S0 are used in the same way as CBSEL[1:0] as of the vector address indicators. The Warm Boot applet is created same way as shown in Figure 11.

After selecting the desired configuration image, the application then asserts the internal signal BOOT port High to force the FPGA to restart the configuration process from the specified vector address stored in PROM.

## Time-Out and Retry

When configuring from external SPI Flash, the iCE40 device looks for a synchronization word. If the device does not find a synchronization word within its timeout period, the device automatically attempts to restart the configuration process from the very beginning. This feature is designed to address any potential power-sequencing issues that may occur between the iCE40 device and the external PROM.

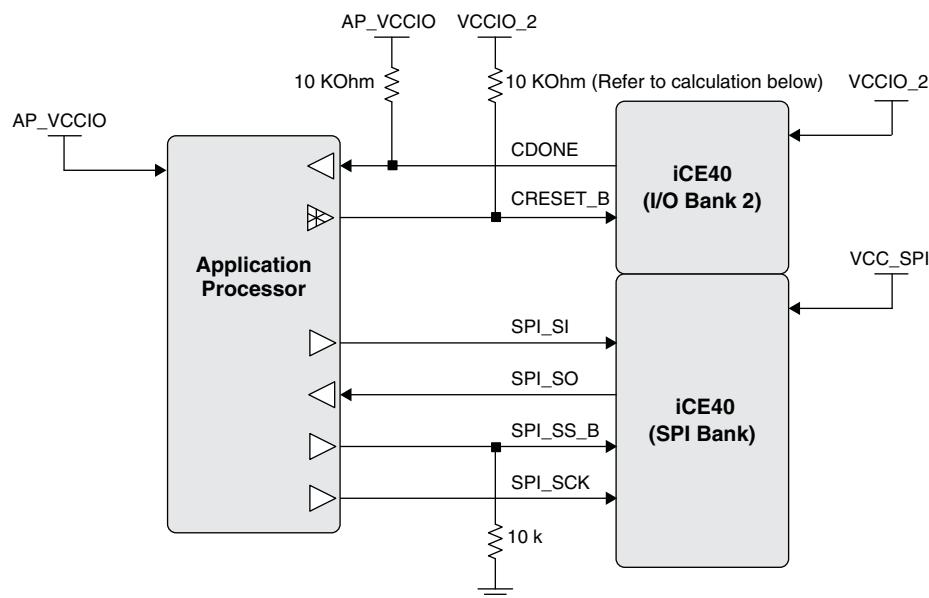
The iCE40 device attempts to reconfigure six times. If not successful after six attempts, the iCE40 FPGA automatically goes into low-power mode.

## SPI Slave Configuration Interface

Using the SPI slave configuration interface, an application processor (AP) serially writes a configuration image to an iCE40 FPGA using the iCE40's SPI interface, as shown in Figure 6. The iCE40's SPI configuration interface is a separate, independent I/O bank, powered by the VCC\_SPI supply input. Typically, VCC\_SPI is the same voltage as the application processor's I/O. The configuration control signals, CDONE and CRESET\_B, are supplied by the separate I/O Bank 2 voltage input, VCCIO\_2.

This same SPI slave interface supports the programming of the Nonvolatile Configuration Memory (NVCM) of the iCE40.

**Figure 13. iCE40 SPI Slave Configuration Interface**



The SPI control signals are defined in Table 9.

**Table 10. SPI Slave Configuration Interface Pins (SPI\_SS\_B Low when CRESET\_B Released)**

Signal Name	Direction	iCE40 I/O Supply	Description
CDONE	Output	VCCIO_2	Configuration Done output from iCE40. Connect to a external pull-up resistor to the application processor I/O voltage, AP_VCC. The resistor size can be calculated knowing the configuration clock frequency (SCLK or MCLK) and the CDONE trace capacitance with the following formula: $R_{pullup}=1/(2 * ConfigFrequency * CDONETraceCap)$ The iCE40-1KLP SWG16 package CDONE pin can be used as a user output.
CRESET_B	Input		Configuration Reset input on iCE40. Typically driven by AP using an open-drain driver, which also requires a 10 KOhm pull-up resistor to VCCIO_2.
VCC_SPI	Supply	VCC_SPI	SPI Flash PROM voltage supply input.
SPI_SI	Input		SPI Serial Input to the iCE40 FPGA, driven by the application processor.
SPI_SO	Output		SPI Serial Output from CE65 device to the application processor. Not actually used during SPI peripheral mode configuration but required if the SPI interface is also used to program the NVCN.
SPI_SS_B	Input		SPI Slave Select output from the application processor. Active Low. Optionally hold Low prior to configuration using a 10 KOhm pull-down resistor to ground.
SPI_SCK	Input		SPI Slave Clock output from the application processor.

After configuration, the SPI port pins are available to the user-application as additional PIO pins, supplied by the VCC\_SPI input voltage.

## Enabling SPI Configuration Interface

The optional 10 KOhm pull-down resistor on the SPI\_SS\_B signal ensures that the iCE40 FPGA powers up in the SPI peripheral mode. Optionally, the application processor drives the SPI\_SS\_B pin Low when CRESET\_B is released, forcing the iCE40 FPGA into SPI peripheral mode.

## SPI Slave Configuration Process

Figure 14 illustrates the interface timing for the SPI slave mode and Figure 15 outlines the resulting configuration process. The actual timing specifications appear in the data sheet. The application processor (AP) begins by driving the iCE40 CRESET\_B pin Low, resetting the iCE40 FPGA. Similarly, the AP holds the iCE40's SPI\_SS\_B pin Low. The AP must hold the CRESET\_B pin Low for at least 200 ns. Ultimately, the AP either releases the CRESET\_B pin and allows it to float High via the 10 KOhm pull-up resistor to VCCIO\_2 or drives CRESET\_B High. The iCE40 FPGA enters SPI peripheral mode when the CRESET\_B pin returns High while the SPI\_SS\_B pin is Low,

After driving CRESET\_B High or allowing it to float High, the AP must wait a minimum of 300  $\mu$ s, allowing the iCE40 FPGA to clear its internal configuration memory.

After waiting for the configuration memory to clear, the AP sends the configuration image generated by the Diamond Programmer (version 2.2 or later). An SPI slave mode configuration image must not use the Cold Boot or Warm Boot options. Send the entire configuration image, without interruption, serially to the iCE40's SPI\_SI input on the falling edge of the SPI\_SCK clock input. Once the AP sends the 0x7EAA997E synchronization pattern, the generated SPI\_SCK clock frequency must be within the data sheet specified range while sending the configuration image. Send each byte of the configuration image with most-significant bit (msb) first. The AP sends data to the iCE40 FPGA on the falling edge of the SPI\_SCK clock. The iCE40 FPGA internally captures each incoming SPI\_SI data bit on the rising edge of the SPI\_SCK clock. The SPI\_SO output pin in the iCE40 is not used during SPI slave mode but must connect to the AP if the AP also programs the Nonvolatile Configuration Memory (NVCN) of the iCE40 device.

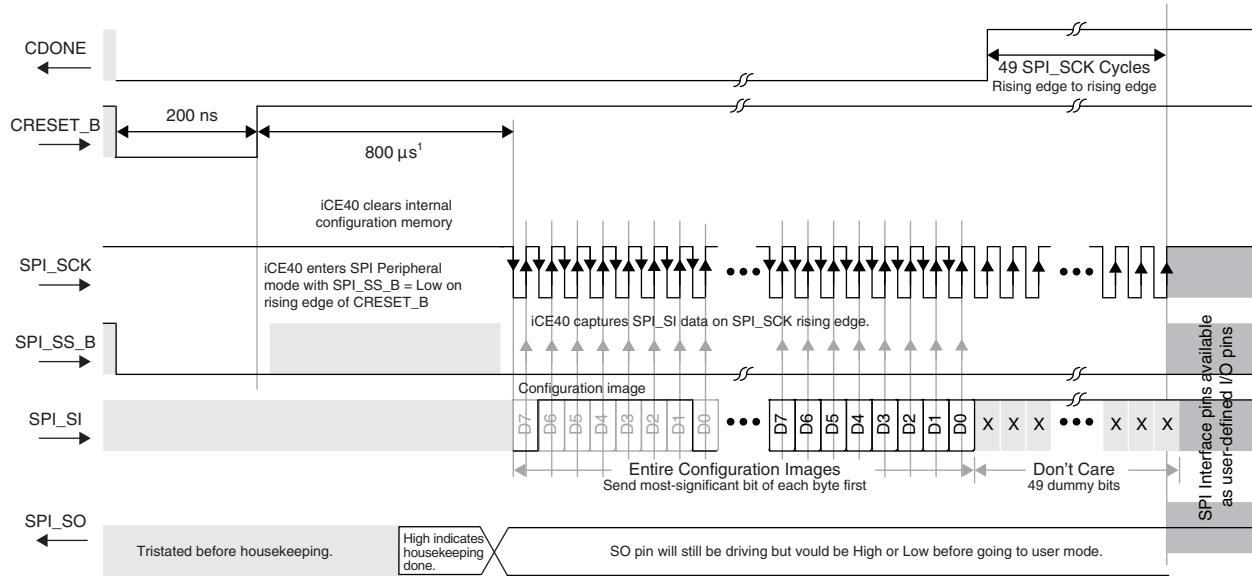
After sending the entire image, the iCE40 FPGA releases the CDONE output allowing it to float High via the external pull-up resistor to AP\_VCC. If the CDONE pin remains Low, then an error occurred during configuration and the AP should handle the error accordingly for the application.

After the CDONE output pin goes High, send at least 49 additional dummy bits, effectively 49 additional SPI\_SCK clock cycles measured from rising-edge to rising-edge.

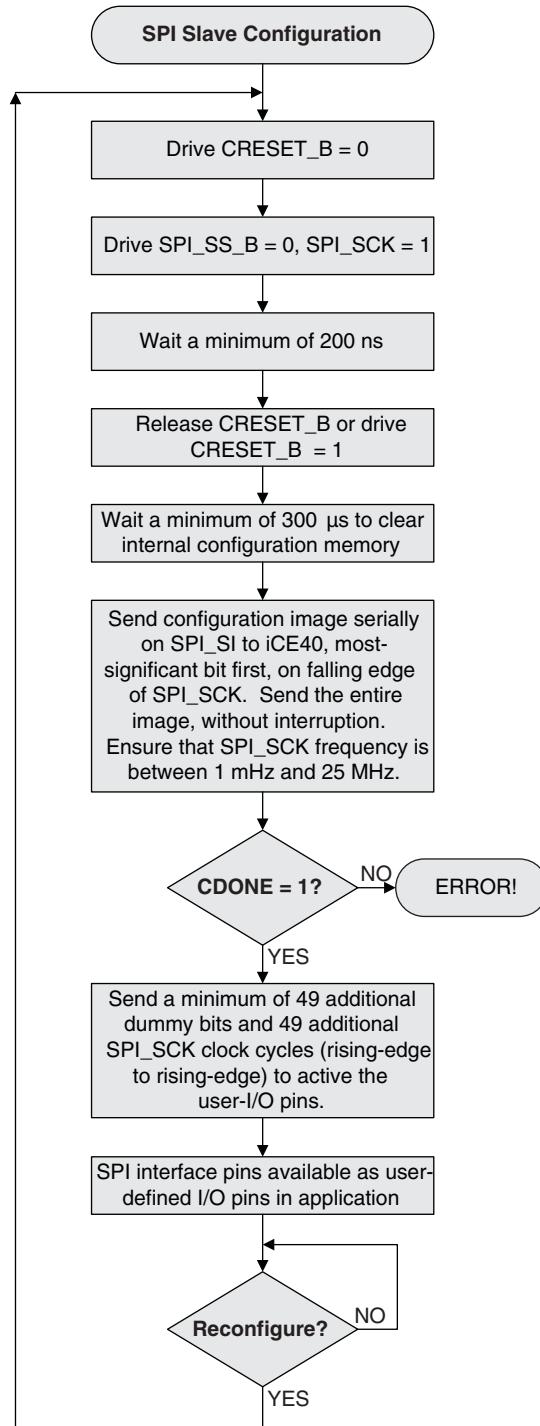
After the additional SPI\_CLK cycles, the SPI interface pins then become available to the user application loaded in FPGA. In the iCE40-1KLP SWG16 package, the CDONE pin can be used as a user output.

To reconfigure the iCE40 FPGA or to load a different configuration image, merely restart the configuration process by pulsing CRESET\_B Low or power-cycling the FPGA.

**Figure 14. Application Processor Waveforms for SPI Peripheral Mode Configuration Process**



**Figure 15. SPI Slave Configuration Process**



Refer to Appendix A for the SPI peripheral configuration procedure.

## Voltage Compatibility

As shown in Figure 6, there are potentially three different supply voltages involved in the SPI Peripheral interface, described in Table 11.

**Table 11. SPI Peripheral Mode Supply Voltages**

Supply Voltage	Description
AP_VCCIO	I/O supply to the Application Processor (AP)
VCC_SPI	Voltage supply for the iCE40 SPI interface
VCCIO_2	Supply voltage for the iCE40 I/O Bank 2

Table 12 describes how to maintain voltage compatibility for two interface scenarios. The easiest interface is when the Application Processor's (AP) I/O supply rail and the iCE40's SPI and VCCIO\_2 bank supply rails all connect to the same voltage. The second scenario is when the AP's I/O supply voltage is greater than the iCE40's VCCIO\_2 supply voltage.

**Table 12. CRESET\_B and CDONE Voltage Compatibility**

Condition	CRESET_B			CDONE Pull-up	Requirement
	Direct	Open- Drain	Pull-up		
AP_VCCIO = VCC_SPI AP_VCCIO = VCCIO_2	OK	OK with pull-up	Required if using open- drain output	Recommended	AP can directly drive CRESET_B High and Low although an open-drain output recommended is if multiple devices control CRESET_B. If using an open-drain driver, the CRESET_B input must include a 10 KOhm pull-up resistor to VCCIO_2. The 10 KOhm pull-up resistor to AP_VCCIO is also recommended.
AP_VCCIO > VCCIO_2	N/A	Required, requires pull-up	Required	Required	The AP must control CRESET_B with an open-drain output, which requires a 10 KOhm pull-up resistor to VCCIO_2. The 10 KOhm pull-up resistor to AP_VCCIO is required.

## Technical Support Assistance

e-mail: [techsupport@latticesemi.com](mailto:techsupport@latticesemi.com)

Internet: [www.latticesemi.com](http://www.latticesemi.com)

## Revision History

Date	Version	Change Summary
December 2014	2.6	Updated <a href="#">Internal Device Reset</a> section. — Revised Figure 5, iCE40 Internal Reset Circuitry. Added VCCIO_0 device pin. — Revised Table 4, sysCONFIG Pins. Added VCCIO_0 in Supply Rail.
		Updated <a href="#">SPI PROM Requirements</a> section. Corrected device in last row of Table 8, Bitstream Sizes for Different iCE40 FPGA Densities Used to Select a SPI Flash.
		Updated <a href="#">Table 6</a> , sysCONFIG Pins. — Changed SPI_SI and SPI_SO Pin Direction. — Changed SPI_CSN pin to SPI_CS1 in the SPI_SS_B description.
		Updated <a href="#">Cold Boot Configuration Option (Applies to iCE40LP and iCE40HX Devices Only)</a> section. General revision.
July 2014	2.5	Updated <a href="#">Warm Boot Configuration Option</a> section. Added information on enabling warm boot.
		Updated Figure 15-12, Application Processor Waveforms for SPI Peripheral Mode Configuration Process. Removed "...always starts with 0x7EAA997E synchronization word."
June 2014	2.4	Added support for iCE40 Ultra.
January 2014	02.3	Added information to the Configuration Mode Selection section.
		Added Appendix B. Configuration Data Format.
October 2013	02.2	Updated for iCE40LM support.
August 2013	02.1	Added information regarding resistor size calculation in the Configuration Control Signals section; updated the iCE40 Configuration Control Pins figure.
		Added notes in the Power-On Reset (POR) section.
		Updated description of CDONE pin in the sysCONFIG Pin and SPI Peripheral Configuration Interface Pins (SPI_SS_B Low when CRESET_B Released tables).
		Updated the iCE40 SPI Master Configuration Interface figure.
		Updated CDONE information in the SPI Configuration Process section,
		Updated SPI PROM Requirements section.
July 2013	02.0	Updated Technical support Assistance information.
June 2013	01.9	Updated the Cold Boot Configuration Option section to provide more accurate information regarding the Cold Boot mode and the storage of multiple bitstreams.
		Changed VCCIO_AP to AP_VCCIO in Table 5-10.
April 2013	01.8	Updated the SPI Peripheral Configuration Process section.
	01.7	Updated the Pseudo Code for configuring an iCE40 device.
		Added information on how to access the Oscillator settings.
		Defined iCE40LP/HX4K wait time to be 1200us.
February 2013	01.6	Updated the Bitstream Sizes for Different iCE40 FPGA Densities Used to Select a SPI Flash table.
September 2012	01.5	Updated based on latest iCE40 information: — Included configuration and programming information — Updated bit file sizes — Updated Table 3. Power-on Reset (POR) Voltage Resources — Removed JTAG references — Include configuration algorithm in Appendix A
June 2012	01.4	Updated document with new corporate style.
March 2012	01.3	Initial release.

## Appendix A. SPI Slave Configuration Procedure

### CPU Configuration Procedure

The sequence for configuring the iCE40 SRAM follows.

**Table 13. iCE40 SRAM Configuration Sequence**

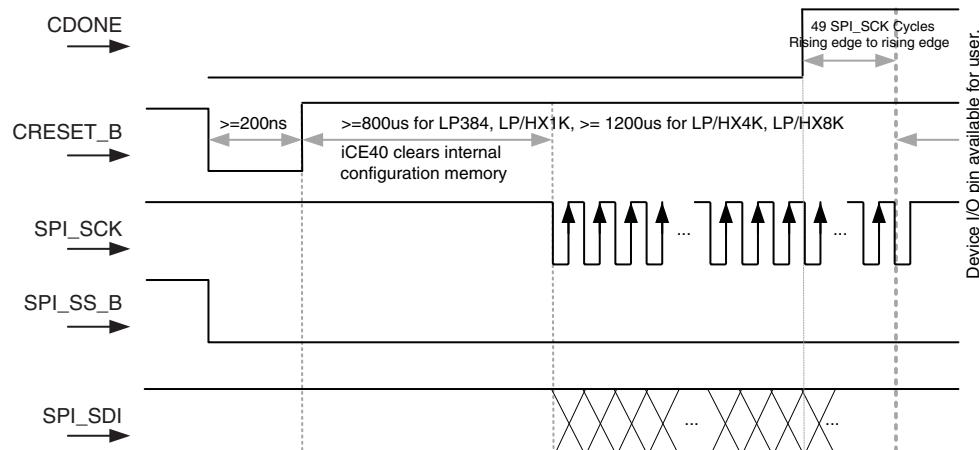
Index	Step	Action	Description
1	Power up the iCE40 FPGA or toggle its CRESET signal low for 200ns and toggle back to high with SPI_SS_B = 0, forcing the device to enter Slave mode. Keep SPI_SS_B low until step 4.	Hold SPI_SS_B =0 and toggle CRESET	See Figure 16 below.
2	Wait > = 800us to 1200us until the iCE40 FPGA completes internal housekeeping work and is ready to receive CPU bitmap data and instructions.	Wait >= 1200us for iCE40LP/HX4K, iCE40LP/HX8K	
3	From this point on, the iCE40 FPGA requires CPU provides operation clock through the SPI_SCK pin of the iCE40 FPGA until configuration is complete.	Feed clock to the SPI_CLK pin	
4	Read and start sending the FPGA bitmap to the iCE40 device. Data goes to the SPI_SDI pin.	Tx bitmap data from user memory	Each clock shifts one bit of data at the clock falling edge. Hold previous SPI_SS_B state, no toggle. Complete sending all bits in bitmap file. Important: Continuous clock is required.
5	Wait for 100 clocks.	Shifting 100 clocks	Configuration complete after 100 clocks.
6	Monitor the CDONE pin. It should go to high. Otherwise, the configuration fails and stops.	Check CDONE Hi	Device in operation when CDONE = 1; device fails when CDONE = 0.

### Configuration Waveforms

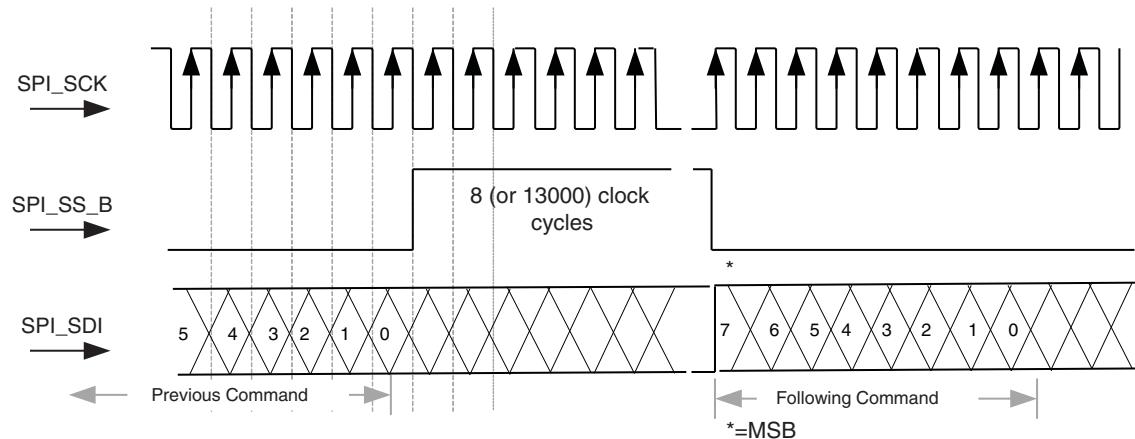
#### iCE40 Reset Waveforms

The reset timing waveforms for initiating NVCM programming are shown below.

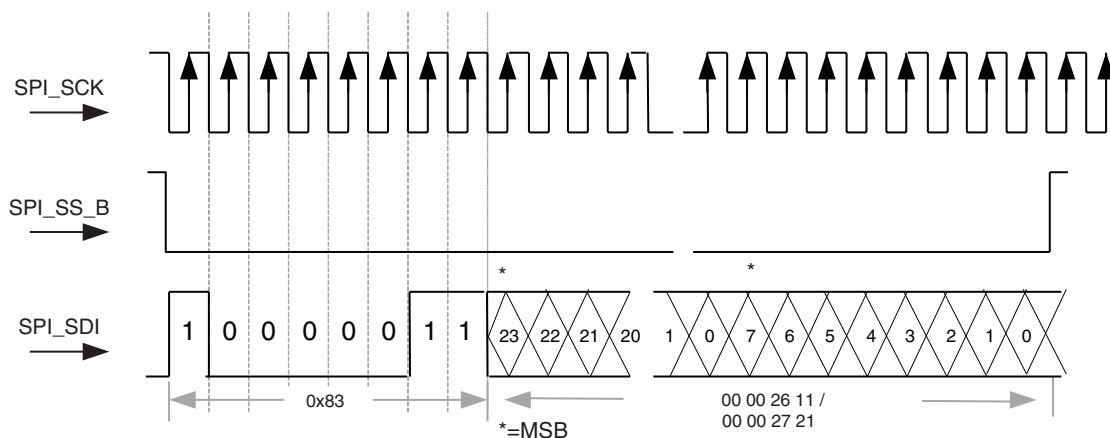
**Figure 16. iCE40 Reset Waveform 1**



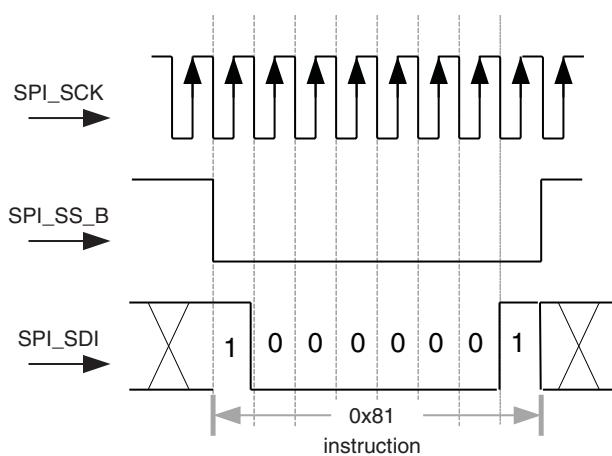
**Figure 17. iCE40 Timing Waveform 2**



**Figure 18. iCE40 Timing Waveform 3**



**Figure 19. iCE40 Timing Waveform 4**



## Pseudo Code

### Configuration

The pseudo code included below will configure an iCE40 device. It assumes a raw binary file generated from iCEcube will be used (\*.bin). Alternatively, a hex file is also generated from iCEcube and can be used as well. The implementation should only make the necessary text-to-binary conversions.

```
//  
//  iCE40 Configuration Pseudo-code  
  
void Config_iCE40 (type, file)  
{  
    //  
    // Open Hex File early to avoid clock delay later  
    //  
    file_pointer = fopen(file); // Open bin file  
  
    //  
    // Reset the iCE40 Device  
    //  
    Set_Port(SPI_SS_B, false); // Set SPI_SS_B low  
    Set_Port(CRESET, false); // Set CRESET low  
    Set_Port(SPI_CLK, true); // Set SPI_CLK high  
    nSec_Delay(200); // Delay 200 nsec  
    Set_Port(CRESET, true); // Set CRESET high  
    if (type == L1K or L4K)  
        uSec_Delay(800); // Delay 800 usec if L1K,L4K  
    else if (type == L8K)  
        uSec_Delay(1200); // Delay 1200 usec for L8K  
    Set_Port(SPI_SS_B, true); // Set SPI_SS_B high  
    Send_Clocks (8); // Send 8 clocks  
    //  
    // Send data from bin file  
    //  
    Send_File(file_pointer); // Send bin file  
    Send_Clocks (100); // Send 100 clocks  
    //  
    // Verify successful configuration  
    //  
    if (Get_Port(CDONE))  
        Return PASS; // PASS if CDONE is true  
    else  
        Return FAIL; // FAIL if CDONE is false  
}  
  
//  
// Clock Generation 10MHz  
//  
void Send_Clocks(num_clocks)  
{  
    for (i = 0; i < num_clocks; i++)  
    {  
        Set_Port(SPI_CLOCK, false); // Set SPI_CLK low  
        nSec_Delay(50); // Delay 50 nsec
```

```
    Set_Port(SPI_CLOCK, true); // Set SPI_CLK high
    nSec_Delay(50); // Delay 50 nsec
}
}

//  

// Send Data from file  

//  

void Send_File(file_pointer)
{
    byte = getc(file_pointer); // Read first byte from file
    while (byte != EOF)
    {
        Send_Byte (byte); // Send data byte
        byte = getc(file_pointer); // read next byte from file
    }
}
```

## Appendix B. Configuration Data Format

The base binary file format is the same for all configuration modes. Different File types (hex, binary, ASCII and others) may ultimately be used to configure the device, but the data in the file is the same. Figure 14 shows the format of a bitstream. The bitstream consists of a comment field, a header, the preamble, and the configuration setup and data.

**Table 14. Configuration Data**

Frame	Contents	Description
Comments	start with ff 00	Comments including company name, software name, parts name etc.
Header	00 ff	32 bits dummy bits
	7e aa 99 7e	32 bits Preamble
config frequency		32 bits
reset CRC		32 bits
boot mode		48 bits
frame size		96 bits
start to write frame		48 bits
data 0		64 bits
	...	
dummy bits	00 00	32 bits
data 1		64 bits
	...	
dummy bits	00 00	32 bits
data 2		64 bits
	...	
dummy bits	00 00	32 bits
data 3		64 bits
	...	
dummy bits	00 00	32 bits
frame size		96 bits
bank address		32 bits
start to write frame		48 bits
ebr write command		32 bits
ebr data	...	
dummy bits	00 00	32 bits
CRC check		48 bits
wake up	01 06	32 bits

1. The data in this table is intended for reference only.