LatticeSC sysCONFIG Usage Guide

October 2008 Technical Note TN1080

Introduction

Configuration is the process of loading or programming a design into volatile memory of an SRAM-based FPGA. This is accomplished via a bitstream file, representing the logical states, that is loaded into the FPGA internal configuration SRAM memory. The device's functional operation after being programmed is determined by these internal configuration RAM settings. The SRAM cells must be loaded with configuration data each time the device powers up.

This technical note is segmented into three main sections: Configuration Modes, Bitstream Generation, and Configuration Process & Flow. The Configuration Modes section shows all the different modes with pin information, schematic diagrams, functional timing waveforms and descriptions. The Bitstream Generation section describes the options available when creating a bitstream file with the ispLEVER® Bitstream Data Generator software program; this section also shows the configuration frame format and content. The Configuration Process & Flow section details the states of operation of the device during configuration, miscellaneous configuration options, configuration frame sizes and bitstream file contents. This document does not contain any configuration performance specifications for devices. Waveforms are presented for reference, however AC timing is not included. Refer to the LatticeSC/M Family Data Sheet for this information.

Configuration Modes

The LatticeSC™ configuration RAM can be loaded in a number of different modes. In these configuration modes, the FPGA can act as a master, a peripheral to a CPU, or a slave of other system devices. It also supports in-system configuration via the JTAG port.

The decision as to which configuration mode to use is a system design concern. There are many methods for configuring the FPGA utilizing four basic schemes.

Master – As a master, the FPGA is the source of the clock, which addresses an external PROM or EPROM storage device through either a serial or parallel data connection, with no additional timing or control signals used. This scheme includes Serial Programming Interface (SPI) that supports a seamless connection for programming using industry-standard external Flash-based memory devices.

Peripheral – In the peripheral mode, the FPGA acts as a CPU peripheral and accepts parallel data and interacts with the source of data, usually with a Ready/Busy handshake. This includes MPI (microprocessor interface) mode where the FPGA receives bus wide data (8, 16, 32-bit), control and a clock from a microprocessor as a direct extension to the processor bus.

Slave – FPGA receives bit-serial or byte-wide data and a clock from an external data and timing source, either from a microprocessor, or from the lead device in an FPGA-daisy chain. As a slave device, a clock is generated externally and provided into the CCLK input.

JTAG – The device can be configured through the ispJTAG™ port. The JTAG port is always on and available regardless of the configuration mode selected.

The system designer should determine the requirements for configuration very early in the design. Many factors must be considered when deciding which configuration mode is best suited for the design. The flexible features for configuration can provide a seamless design to the system.

The following table will assist the user in deciding which configuration mode is best suited for a particular design. The table queries the system features that may help the user understand which programming mode may be best suited for the design.

Table 1. Configuration Mode Overview

Configuration Mode	Microprocessor Support	Programming Latency Cost	Required Interconnection Cost	Supports Bitstream Compression	Key Features
Master Serial	No	High	Low	Yes	
Master Parallel	No	High	Medium	Yes	
Master Byte	No	Low	Medium	Yes	Supports 8-bit parallel daisy-chaining
Asynchronous Peripheral	Yes	High	Medium	Yes	Support for micro-controller and other 8-bit interfaces
MPI -8, -16, -32	Yes	Low	High	Yes	Glue-less interface to PowerPC micro-processors
Slave Serial	No	High	Low	No	
Slave Parallel	No	Low	Medium	Yes	
SPI	No	High	Low	Yes	Inexpensive Flash memory support
SPIX	No	High	Low	Yes	Inexpensive Flash memory support
JTAG	No	High	Low	No	IEEE 1532

All the programming interfaces except JTAG require an initial setting of the Mode[3:0] input pins prior to power up or re-initialization. This setting is latched on the rising edge of INITN and is used by the configuration logic to determine the appropriate clock, data, and control handshake signaling. Table 2 lists the functions of the configuration mode pins.

Table 2. Mode Pin Logic Settings

Configuration	Mode	M[3]	M[2]	M[1]	M[0]	CCLK	Data	Daisy Chaining
Master	Serial	1	0	0	0	Output	1-bit	Serial
Master	Parallel	1	1	0	0	Output	8-bit	Serial
Master	Byte	0	1	1	0	Output	8-bit	8-bit
Asynchronous	Peripheral	1	1	0	1	Output	8-bit	Serial
Slave	Serial	1	1	1	1	Input	1-bit	Serial
Slave	Parallel	1	0	0	1	Input	8-bit	None
SPI	Serial	0	1	0	1	Output	1 or 8-bit	Serial
SPIX	Serial	0	1	0	0	Output	1 or 8-bit	Serial
MPI	8-bit	1	0	1	0	Output	8-bit	Serial
MPI	16-bit	1	0	1	1	Output	16-bit	Serial
MPI	32-bit	1	1	1	0	Output	32-bit	Serial
JTAG 1532	Serial	X	X	X	X	NA	1-bit	Serial

Configuration Pins

The LatticeSC devices support two types of sysCONFIG™ pins, dedicated and dual-purpose. This document will use names based on the purpose used by a specified configuration mode. There are many pins used for various functions across many modes. There are some dedicated device pins that are defined exclusively for configuration functions. The dual-purpose pins are available as extra user-defined I/O pins when not being used for configuration. A programmable option controls the dual-purpose configuration pins. This option is made via a preference in Lattice ispLEVER software or as an HDL source file attribute. The LatticeSC devices also support the ispJTAG port for configuration, including transparent read back and JTAG testing. Appendix A describes the functionality of the sysCONFIG and ispJTAG pins.

Most of the LatticeSC configuration pins are considered dual purpose; they are used as configuration pins during configuration, and then are user I/Os after configuration. Both the dedicated and dual-purpose configuration pins are physically located in VCCIO1 (bank 1). JTAG pins are powered by a dedicated VCCJ supply. Table 3 highlights the use of each pin per configuration mode. VCCO1 can be 1.8V, 2.5V or 3.3V for configuration.

Upon power up or during re-initialization of the FPGA, configuration is done using default LVCMOS buffers (which operate between 1.8V and 3.3V). As configuration is taking place, the buffer type will change to the user-specified buffer type in the FPGA design in all banks except bank 1. Bank 1 buffer types will change only after the configuration is completed. The LatticeSC FPGA offers many high-speed PURESPEED™ I/O types having many different operation levels that may lead to improper behavior after configuration is completed. It is recommended not to use the PURESPEED I/O buffer types that are utilized for a particular programming mode except on these dual-use configuration pins for LVCMOS applications. Consequently, we do not recommend using the configuration pins as dual-use pins in order to avoid potential conflicts during the configuration process. A specific software control is available to the designer to prevent this type of conflict. This will be discussed later in this document.

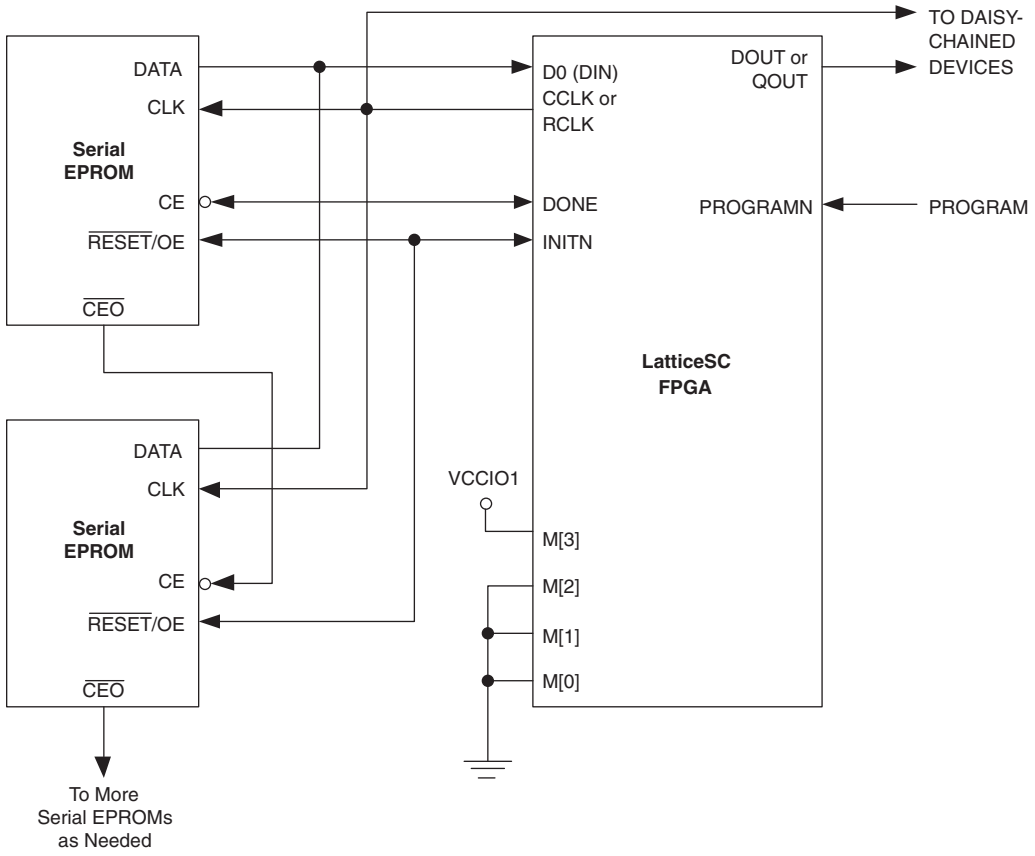
Table 3. sysCONFIG Pin Usage

Signal Name	Configuration Mode										VCCIO Bank 1 Type
	JTAG	Slave Serial	Slave Parallel	Master Serial	Master Parallel	Master Byte	Asynch Periph.	SPI / SPIX	MPI	Serial Read-back	
RESETN	✔	✔	✔	✔	✔	✔	✔	✔	✔		Dedicated
CCLK		✔	✔	✔	✔	✔	✔	✔	✔		
DONE	✔	✔	✔	✔	✔	✔	✔	✔	✔		
PROGRAMN	✔	✔	✔	✔	✔	✔	✔	✔			
RDCFGN	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	
INITN	✔	✔	✔	✔	✔	✔	✔	✔	✔		
CFGIRQN/MPI_IRQ_N		✔	✔	✔	✔	✔	✔	✔	✔		
M[3:0]		✔	✔	✔	✔	✔	✔	✔	✔		
TDO/RDDATA	✔									✔	VCCJ
TCK	✔										
TDI	✔										
TMS	✔										
QOUT/CEON		✔	✔	✔	✔	✔	✔	✔	✔		Dual Use
DOUT		✔	✔	✔	✔		✔	✔	✔		
RDY/BUSYN/RCLK			✔	✔	✔		✔	✔	✔		
HDC		✔	✔	✔	✔	✔	✔	✔	✔		
LDCN		✔	✔	✔	✔	✔	✔	✔	✔		
CS0N, CS1			✔		✔		✔	✔	✔		
RDN							✔		✔		
WRN							✔	✔	✔		
D[0]		✔	✔	✔		✔	✔	✔	✔		
D[7:1]			✔		✔		✔	✔	✔		
A[21:0]					✔				✔		
MPI_TA, MPI_TEA, MPI_RETRY									✔		
MPI_CLK									✔		
D[31:8], DP[3:0]									✔		

Master Serial Mode

In the master serial mode, the FPGA loads the configuration data from an external serial ROM. The configuration data is either loaded automatically at start-up or on a command using the PROGRAMN pin to reconfigure. Serial PROMs can be used to configure the FPGA in the master serial mode and is shown in Figure 1.

Figure 1. Master Serial Configuration Diagram



Dedicated Master Serial Pins	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	O	CCLK is an output which strobes configuration data into daisy-chained devices. CCLK is output for daisy-chaining operation when the lead FPGA is a master.
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	PROGRAMN is an active-low input that forces the restart of configuration. This pin always has an active pull-up.

Dedicated Master Serial Pins	I/O	Description
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.
CFGIRQN	O	During JTAG, slave, master, and asynch. peripheral configuration, assertion on this CFGIRQN (active low) indicates error(s) for block RAM or MACO™ initialization. A programmable pull-up resistor is available on this pin.
M[3:0]	I	During power-up and initialization, M[3:0] are used to select the configuration mode with their values latched on the rising edge of INITN. After configuration these pins are available as user defined inputs. Mode Pin Setting: M[3]=1, M[2]=0, M[1]=0, M[0]=0
Dual-Purpose Pins: After configuration, these pins are user-programmable I/O pins		
DOUT	O	DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
QOUT	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. Data out on QOUT changes on the rising edge of CCLK.
RCLK	O	RCLK is a read clock output signal to an external memory. The RCLK frequency is the same as CCLK when used with uncompressed bitstreams. However, RCLK will be 1/8 the frequency of CCLK when the bitstream is compressed.
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
LDCN	O	Low During Configuration is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
D0(DIN)	I	D[0](DIN) receives configuration data, and the pin is pull-up enabled.

Configuration in the master serial mode can be done at power-up and/or on command. The system or the FPGA must activate the serial ROM's RESET/OE and CE inputs. At power-up, the FPGA and serial ROM each contain internal power-on reset circuitry that allows the FPGA to be configured without the system providing an external signal. The power-on reset circuitry causes the serial ROM's internal address pointer to be reset. After power up, the FPGA automatically enters its initialization phase. The serial ROM/FPGA interface used depends on such factors as the availability of a system reset pulse, availability of an intelligent host to generate a configure command, whether a single serial ROM is used or multiple serial ROMs are cascaded, whether the serial ROM contains a single or multiple configuration programs, etc. Because of differing system requirements and capabilities, a single FPGA/serial ROM interface is not appropriate for all applications.

Data is read by the FPGA sequentially from the serial ROM. The DATA output from the serial ROM is connected directly into the DIN input of the FPGA. The CCLK output from the FPGA is connected to the CLK input of the serial ROM. During the configuration process, CCLK clocks one data bit on each rising edge.

Since the data and clock are direct connects, the FPGA/serial ROM design task is to use the system or FPGA to enable the RESET/OE and CE of the serial ROM(s). There are several methods for enabling the serial ROM's RESET/OE and CE inputs. The serial ROM's RESET/OE is programmable to function with RESET active-high and OE active-low or RESET active-low and OE active-high.

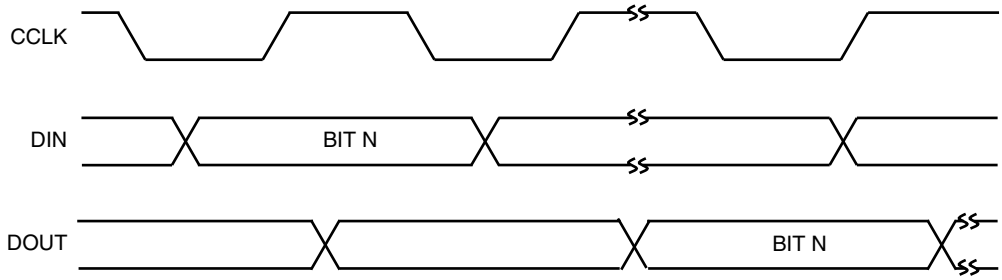
With the before-mentioned PROM options, the FPGA's INITN pin is connected to the serial ROMs' RESET/OE input, which has been programmed to function with RESET active-low and OE active-high. The control host gener-

ates a 500 ns low pulse to the FPGA's PROGRAMN input pin; this will cause INITN to output pulse low to high. The FPGA DONE is connected to the CE pin. The low on DONE enables the serial ROMs. At the completion of configuration, the high on the FPGAs DONE disables the serial ROM.

Serial ROMs can also be cascaded to support the configuration of multiple FPGAs or to load a single FPGA when configuration data requirements exceed the capacity of a single serial ROM. After the last bit from the first serial ROM is read, the serial ROM outputs CEO low and 3-states the DATA output. The next serial ROM recognizes the low on CE input and outputs configuration data on the DATA output. After configuration is complete, the FPGA's DONE output into CE disables the serial ROMs. This FPGA/serial ROM interface is not used in applications in which a serial ROM stores multiple configuration programs. In these applications, the next configuration program to be loaded is stored at the ROM location that follows the last address for the previous configuration program. In some applications, there can be contention on the FPGA's DIN pin. During configuration, DIN receives configuration data, and after configuration, it is a user I/O. If there is contention, an early DONE at start-up (selected with the ispLEVER Bitgen program) may correct the problem.

In master serial mode, the CCLK output of the master device is used for both the external PROM device and daisy-chained devices. The two clocks required for compressed bitstream loading are accomplished by adding the use of the RCLK output, which is a common output pin with master parallel mode and asynchronous peripheral modes. In normal uncompressed daisy-chained devices, the CCLK output is used to provide a clock during loading. However if the bitstream is compressed, the PROM device has to use the RCLK output. If the bitstream is uncompressed, the RCLK follows CCLK.

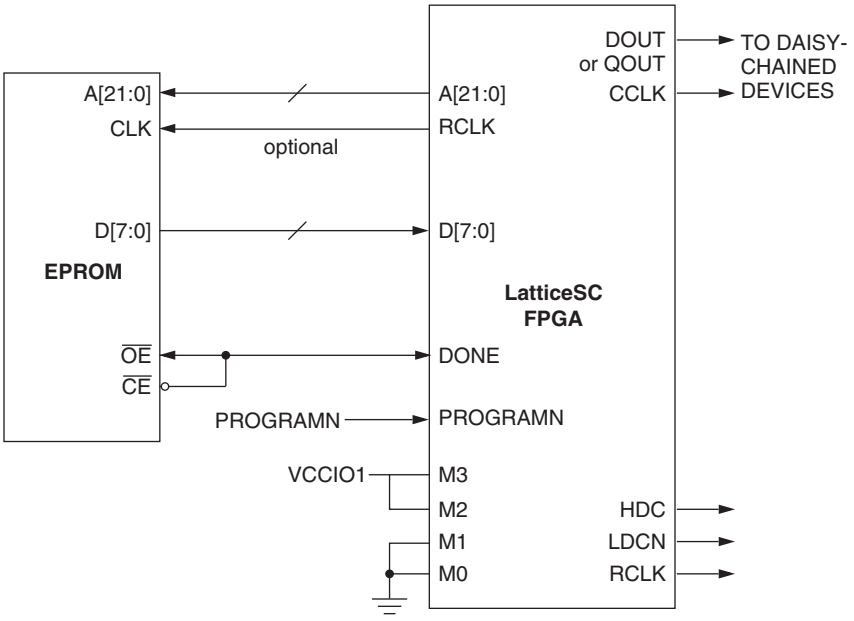
Figure 2. Master Serial Configuration Timing Waveform



Master Parallel Mode

The master parallel configuration mode is generally used to interface to industry-standard, byte-wide memory devices. Figure 3 depicts the interface connections to an EPROM device. The FPGA outputs up to a 22-bit address on A[21:0] to the memory device and reads one byte of configuration data on the rising edge of RCLK. Optionally, RCLK can be driven by the FPGA to the PROM. The parallel bytes are internally serialized starting with the least significant bit, D0. Serial data can be daisy-chained to downstream devices.The 8-bit data bus D[7:0] of the FPGA can be connected to data bus D[7:0] of the microprocessor if a standard PROM file format is used

Figure 3. Master Parallel Configuration Diagram

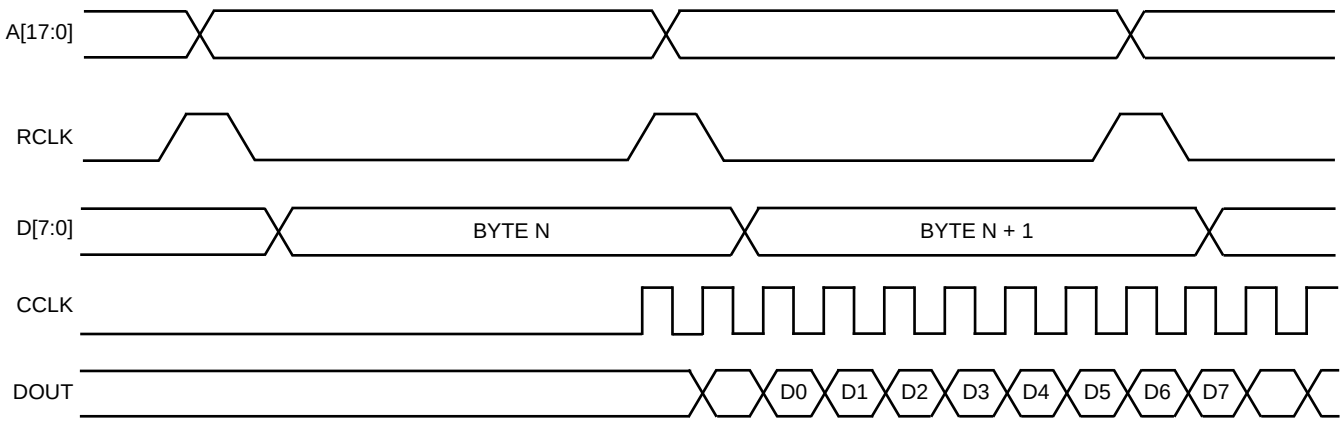


Dedicated Master Parallel Pins	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	O	CCLK is an output which strobes configuration data into daisy-chained devices. CCLK is output for daisy-chaining operation when the lead FPGA is a master.
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	Active-low input forces the restart of configuration. This pin always has an active pull-up.
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.
CFGIRQN	O	During JTAG, slave, master, and asynch. peripheral configuration, assertion on this CFGIRQN (active low) indicates error(s) for block RAM or MACO initialization. A programmable pull-up resistor is available on this pin.
M[3:0]	I	During power-up and initialization, M[3:0] are used to select the configuration mode with their values latched on the rising edge of INITN. After configuration these pins are available as user defined inputs. Mode Pin Setting: M[3]=1, M[2]=1, M[1]=0, M[0]=0

Dedicated Master Parallel Pins	I/O	Description
Dual-Purpose Pins: After configuration, these pins are user-programmable I/O pins		
DOUT	O	During configuration, DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
QOUT	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. Data out on QOUT changes on the rising edge of CCLK.
RCLK	O	RCLK is a read clock output signal to an external memory. The RCLK frequency is the same as CCLK when used with uncompressed bitstreams. However, RCLK will be 1/8 the frequency of CCLK when the bitstream is compressed.
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
LDCN	O	Low During Configuration is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
D[7:0]	I	D[7:0] receives configuration data, and each pin is pull-up enabled.
A[21:0]	O	A[21:0] are optional outputs and will address the configuration PROMs up to 4MB space.

In master parallel mode, the starting memory address is 00000 Hex, and the FPGA increments the address for each byte loaded. One master FPGA can interface to the memory and provide configuration data on DOUT or QOUT to additional FPGAs in a daisy chain. The configuration data on DOUT or QOUT is provided synchronously with the rising edge of CCLK. The frequency of the CCLK output is eight times that of RCLK.

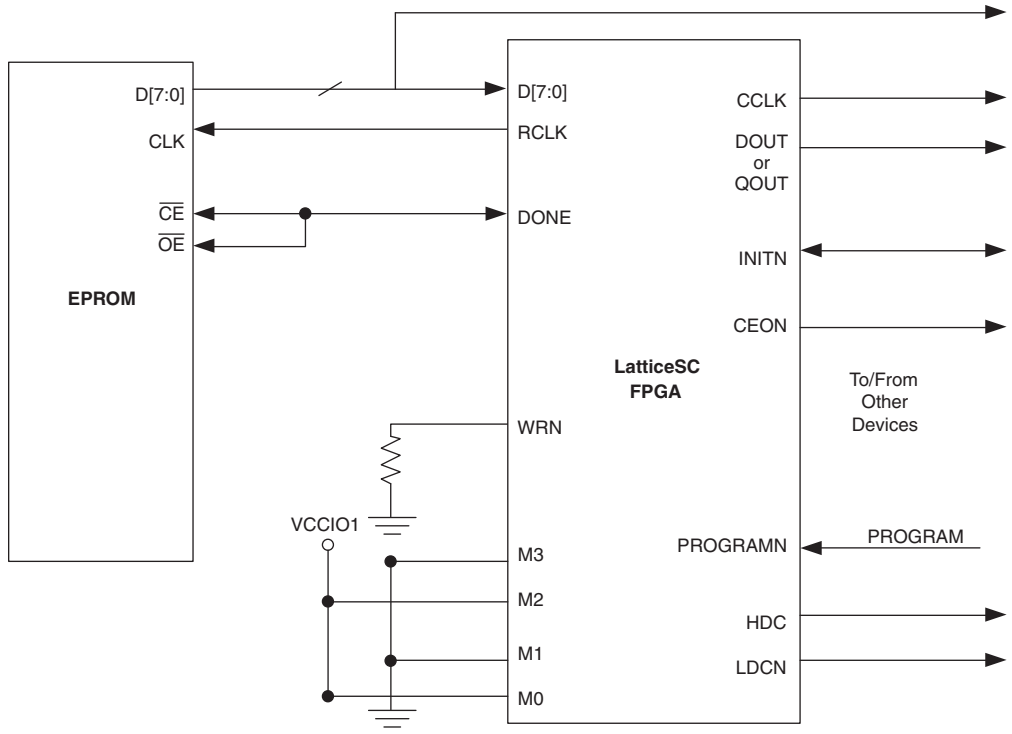
Figure 4. Master Parallel Timing Configuration Waveform



Master Byte Mode

Master byte mode supports byte-wide parallel configuration daisy chaining. In master byte mode the incoming 8-bit data is shifted directly into the FPGA in one clock cycle. The data bus is connected one-to-one with D[7:0] of the device being read. This mode allows for byte wide data and clock to be transmitted to all devices in a chain. The parallel chaining will be discussed later in the document. Figure 5 depicts the connections between the FPGA and a parallel EPROM device.

Figure 5. Master Byte Configuration Diagram



Dedicated Master Byte Pins	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	O	CCLK is an output which strobes configuration data into daisy-chained devices. CCLK is output for daisy-chaining operation when the lead FPGA is a master.
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	Active-low input forces the restart of configuration. This pin always has an active pull-up.
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.
CFGIRQN	O	During JTAG, slave, master, and asynch. peripheral configuration, assertion on this CFGIRQN (active low) indicates error(s) for block RAM or MACO initialization. A programmable pull-up resistor is available on this pin.

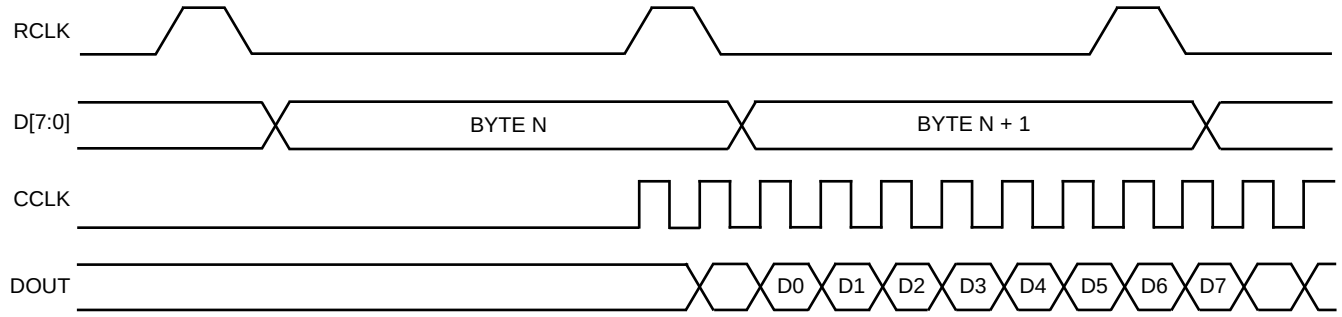
Dedicated Master Byte Pins	I/O	Description
M[3:0]	I	During power-up and initialization, M[3:0] are used to select the configuration mode with their values latched on the rising edge of INITN. After configuration these pins are available as user defined inputs.Mode Pin Setting: M[3]=0, M[2]=1, M[1]=0, M[0]=1
Dual-Purpose Pins: After configuration, these pins are user-programmable I/O pins		
DOUT	O	During configuration, DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
CEON	O	During parallel-chaining configuration, active low CEON enables the cascaded slave device to receive bitstream data. Can be connected to CS0N of daisy-chained slave device.
RCLK	O	RCLK is a read clock output signal to an external memory.
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
LDCN	O	Low During Configuration is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
WRN	I	A low on the write strobe, WRN, loads the data on D[7:0] inputs into an internal data buffer.
D[7:0]	I	D[7:0] receives configuration data, and each pin is pull-up enabled.

The leading master device must be in master byte mode while the down chain slave devices must be in the slave parallel mode. The CEON pin is needed for parallel chaining so that the leading device can enable the following device(s) to receive bitstream data after it completes configuration and asserts CEON output low.

The master byte mode can also support standalone decompression if it uses RCLK to drive the parallel PROM while using CCLK for internal shifting. However, both master byte and slave parallel modes cannot support decompression in parallel chaining configuration. Therefore, if decompression is used, CCLK and DOUT or QOUT must be sent to downstream devices.

The CS1 and WRN inputs for slave parallel mode should be tied to high and low, respectively, in a parallel chaining configuration. This permits CEON output and CS0N input to be linked in a chain.

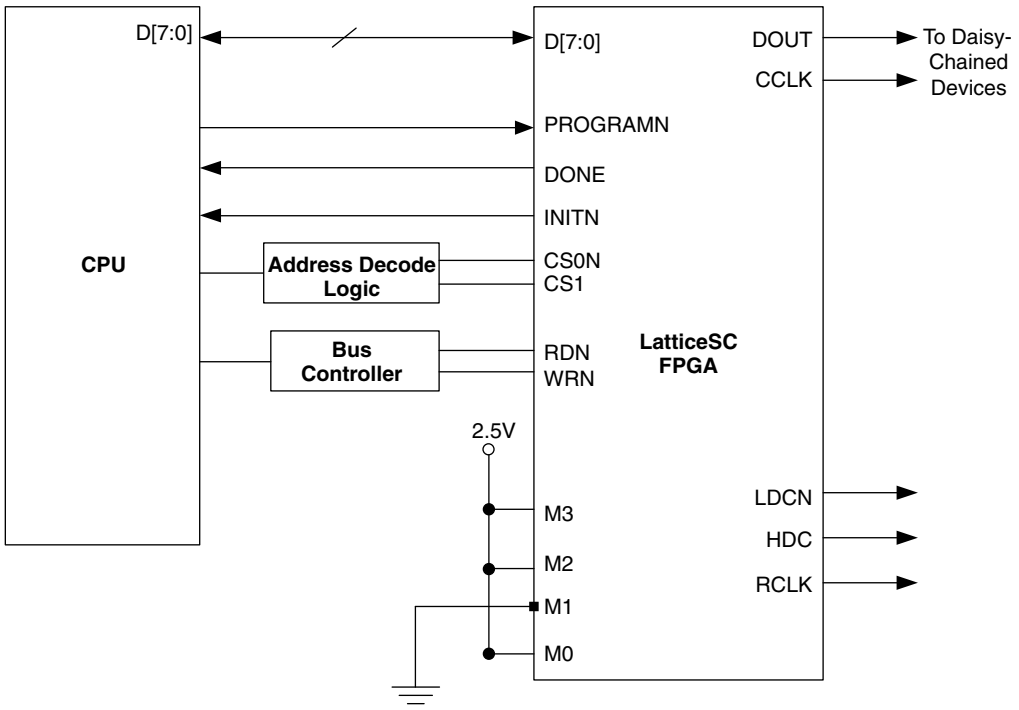
Figure 6. Master Byte Timing Configuration Waveform



Asynchronous Peripheral Mode

In Asynchronous Peripheral mode shown in Figure 7, the FPGA system interface is similar to that of a microprocessor-peripheral interface. This interface is suited for the system that may use any other processor other than the PowerPC or any other microcontroller based system. The microprocessor generates the control signals to write an 8-bit byte into the FPGA. The FPGA control inputs include active-low CS0N and active-high CS1 chip selects and WRN and RDN inputs. The chip selects can be cycled or maintained at a static level during the configuration cycle. Each byte of data is written into the FPGA's D[7:0] input pins. D[7:0] of the FPGA can be connected to D[7:0] of the microprocessor if a standard PROM file format is used.

Figure 7. Asynchronous Peripheral Configuration Diagram



Dedicated Asynchronous Peripheral Pins	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	O	CCLK is an output which strobes configuration data into daisy-chained devices.
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	PROGRAMN is an active-low input that forces the restart of configuration. This pin always has an active pull-up.
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.
CFGIRQN	O	Assertion on this CFGIRQN (active low) indicates error(s) for block RAM or MACO initialization. A programmable pull-up resistor is available on this pin.

Dedicated Asynchronous Peripheral Pins	I/O	Description
M[3:0]	I	During power-up and initialization, M[3:0] are used to select the configuration mode with their values latched on the rising edge of INITN. After configuration these pins are available as user defined inputs. Mode Pin Setting: M[3]=1, M[2]=1, M[1]=0, M[0]=1
Dual-Purpose Pins: After configuration, these pins are user-programmable I/O pins		
DOUT	O	During configuration, DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
QOUT	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. Data out on QOUT changes on the rising edge of CCLK.
BUSYN	O	During configuration in peripheral mode, high on BUSYN indicates another byte can be written to the FPGA. If a read operation is done when the device is selected, the same status is also available on D[7] in asynchronous peripheral mode
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
LDCN	O	Low During Configuration is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
CS0N CS1	I	The FPGA is selected when CS0N is low and CS1 is high.
RDN	I	RDN is used in the asynchronous peripheral configuration mode. A low on RDN changes D[7:3] into status outputs. WRN and RDN should not be used simultaneously. If they are, the write strobe overrides.
WRN	I	When the FPGA is selected, a low on the write strobe, WRN, loads the data on D[7:0] inputs into an internal data buffer
D[7:0]	I/O	D[7:0] receives configuration data, and each pin is pull-up enabled.D[7:3] output internal status for peripheral mode when RDN is low.

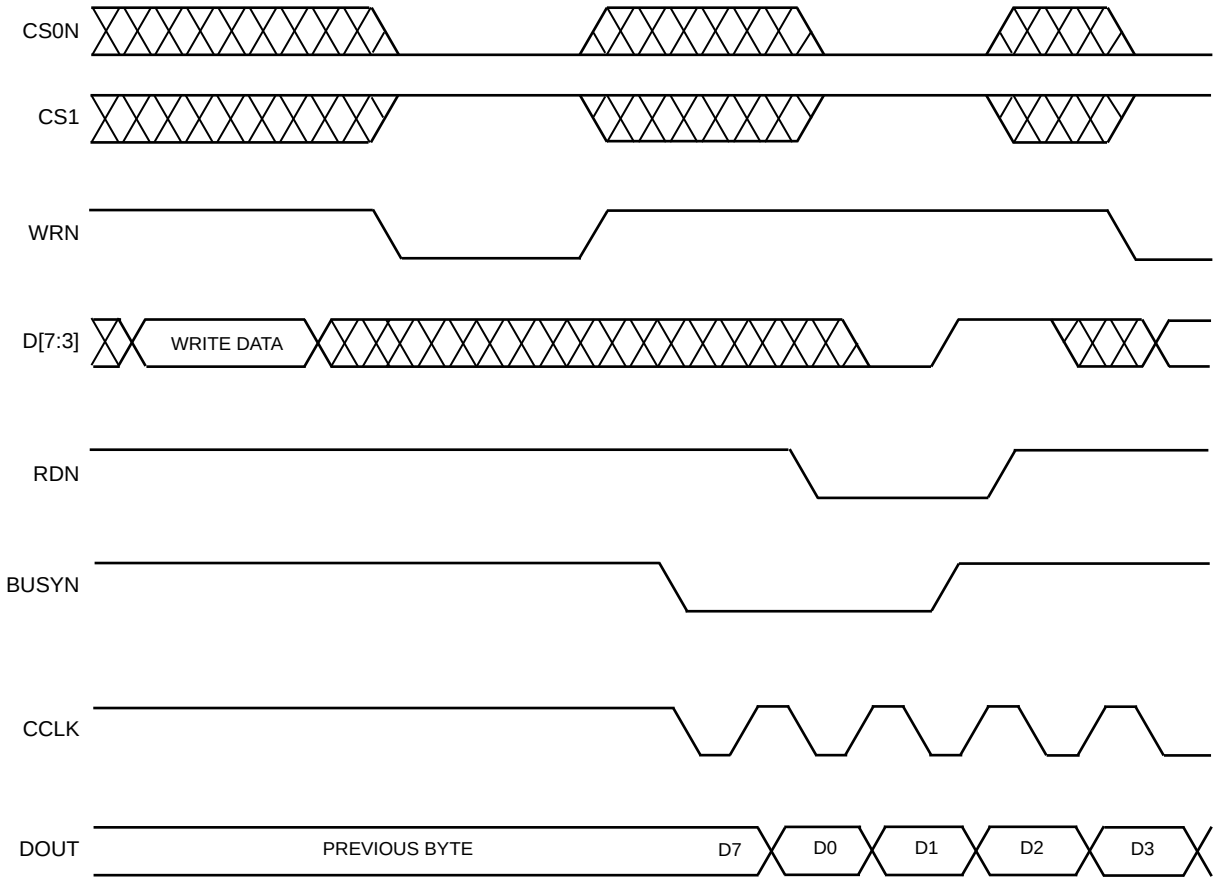
The FPGA provides a BUSYN status output to indicate that another byte can be loaded. A low on BUSYN indicates that the double-buffered hold/shift registers are not ready to receive data, and this pin must be monitored to go high before another byte of data can be written. The shortest time BUSYN is low occurs when a byte is loaded into the hold register and the shift register is empty. In this case, the byte is immediately transferred to the shift register. The longest time for BUSYN to remain low occurs when a byte is loaded into the holding register and the shift register has just started shifting configuration data into configuration RAM.

Status is also available on the D[7:3] pins by enabling the chip selects, setting WRN high, and applying RDN low, where the RDN input provides an output enable for the D[7:3] pins when RDN is low. The following status indicators are available.

D[7]=BUSYN
D[6:5]=ERR_FLAG[1:0], bitstream error flags.
00=No errors, 01=id error, 10= checksum error, 11= alignment error
D[4:3]=CFG_BUS_ERR[1:0], system bus error responses during configuration
00= None, 10=One Error, 11= Multiple Errors

The D[2:0] pins are not enabled to drive when RDN is low and, therefore, only act as input pins in asynchronous peripheral mode. Optionally, the user can monitor the RDY/BUSY status and simply wait until the RDY/BUSY pin to go high, indicating the FPGA is ready for more data, before writing the next data byte. The timing diagram Asynchronous Peripheral configuration is shown in Figure 8.

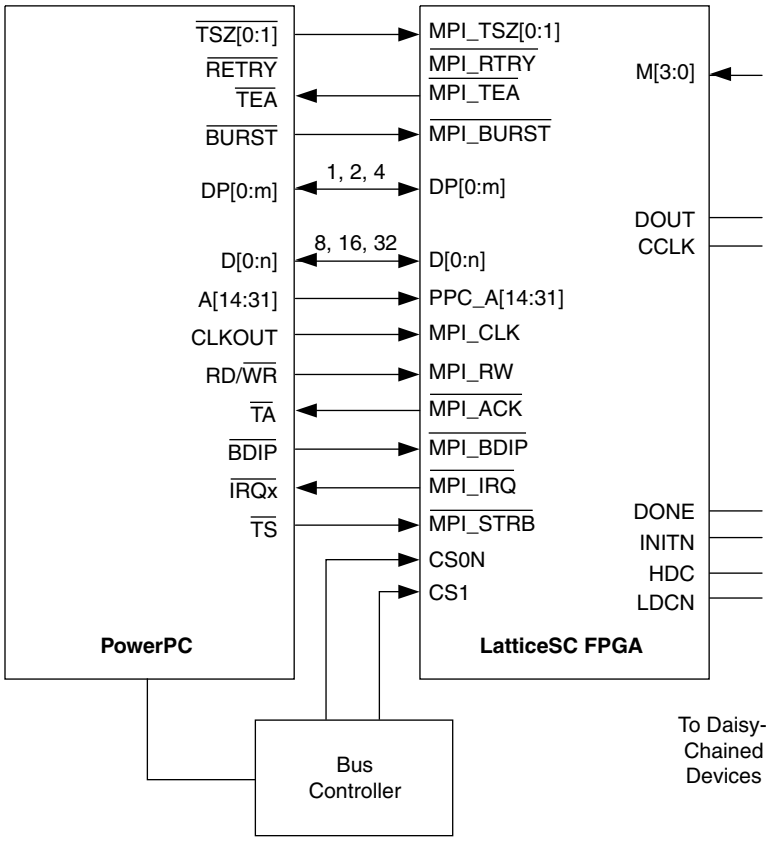
Figure 8. Asynchronous Peripheral Timing Waveform



Microprocessor Interface Mode

The built-in microprocessor interface is designed for use in configuring the FPGA and runtime access of system registers. The MPI permits glueless interface for FPGA configuration and readback from the PowerPC processor. When enabled by the mode pins, the MPI handles all configuration/readback control and handshaking with the host processor. The MPI transfers are clocked from the microprocessor bus via the FPGA MPI_CLK input pin. This interface supports 8, 16 and 32-bit transfers.

Figure 9. Microprocessor Interface Configuration Diagram



Dedicated Microprocessor Interface Pins	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	O	CCLK is an output which strobes configuration data into daisy-chained devices. CCLK is output for daisy-chaining operation when the lead FPGA is in any of the MPI modes.
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	PROGRAMN is an active-low input that forces the restart of configuration. This pin always has an active pull-up.
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.

Dedicated Microprocessor Interface Pins	I/O	Description
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.
MPI_IRQ_N	O	It is the interrupt pin for MPI to external microprocessor. A programmable pull-up resistor is available on this pin.
M[3:0]	I	During power-up and initialization, M[3:0] are used to select the configuration mode with their values latched on the rising edge of INITN. After configuration these pins are available as user defined inputs. Mode Pin Setting: 8-bit M[3]=1, M[2]=0, M[1]=1, M[0]=0 16-bit M[3]=1, M[2]=0, M[1]=1, M[0]=1 32-bit M[3]=1, M[2]=1, M[1]=1, M[0]=0
Dual-Purpose Pins: After configuration, these pins are user-programmable I/O pins		
DOUT	O	During configuration, DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
QOUT	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. Data out on QOUT changes on the rising edge of CCLK.
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
LDCN	O	Low During Configuration is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
CS0N CS1	I	The FPGA is selected when CS0N is low and CS1 is high. During configuration, a pull-up is enabled on both except with MPI DMA access control (see TN1085, LatticeSC MPI/System Bus).
MPI_STRB_N	I	MPI_STRB_N is MPI TS (transfer start) signal.
MPI_RW_N	I	MPI_RW_N is MPI read (high) / write (low) signal.
MPI_DATA	I/O	D[7:0] is also the first byte of MPI data pins.
MPI_BURST_N, MPI_BDIP_N, MPI_TSZ, MPI_ADDR	I	Transfer size = MPI_TSZ Burst mode = MPI_BURST_N Burst in process = MPI_BDIP_N Address MPI_ADDR (connected to FPGA Address pin A[0:17)
MPI_TA, MPI_TEA, MPI_RETRY	O	For MPI configuration modes, these active low signals indicate transfer success (MPI_TA), error (MPI_TEA) or retry (MPI_RETRY).
MPI_CLK	I	Input for MPI clock.
D[0:#], DP[0:#]	I/O	MPI data and parity data buses.

For single FPGA configuration, the host sets the configuration control register MPIPRGM to one then back to zero and, after reading that the configuration write data acknowledge register is high, transfers data 8, 16, or 32 bits at a time to the FPGA's D[0:#] input pins. If configuring multiple FPGAs through serial operation is desired, the SYS_DAISY bit must be set in the configuration control register of the MPI.

The configuration control register offers control bits to enable the interrupt on a bitstream error. The MPI status register may be used in conjunction with, or in place of, the interrupt request option. The status register contains a 2-bit field to indicate the Bitstream error status. A flow chart of the MPI configuration process is shown Figure 10.

Table 4. System Bus Register Map

Absolute Address	Type R=Read W=Write	Bits	Name	Description
0x00000 - 0x00003	R	0:31	DEVICE_ID	32-Bit Manufacturer and Device ID Code: The manufacturer identification code register contains a unique code for each LatticeSC device in the family. The code is comprised of: • ID[0:11]: This is company identification provided by JEDEC. The value for this field is always 0xEA8. • ID[12:19]: Size Identification. The value is the binary number of columns of CIB/PLC in the device. As an example, the LFSC25 device has 72 columns of CIB/PLC so its size identification is 0x48. Refer to the LatticeSC/M Family Data Sheet for the full list of size identification values for different LatticeSC devices. • ID[20:23]:Reserved. • ID[24:27]: Series Identification. LatticeSC is 0x5ID • [28:31]: Reserved to 0x0. For example, the LFSC25 device ID CODE is (ID[0:31]): LFSC25 (72 columns): 0xEA848X50 (X=don't care). The system bus simulation model will always show a LFSC25 device ID. Manufactured devices will have the proper device ID as part of the silicon.
0x00004 - 0x00007	R/W	0:31	SCRATCH_PAD	32-bit scratchpad register. Free register used for debugging purposes.
0x00008- 0x0000A	**R/W**	**Control Registers**		
0x00008	R/W	0:1	RDBK_SIZE	These two bits specify the number of valid bytes in the read back data register (0x00018) during a readback operation. • [0:1]: 00 - 1 byte • 10 - 2 bytes • 01 - 4 bytes
	R/W	2	MPI_USR_ENABLE	Active high. Enables the MPI interface to the user. Used to keep the MPI available during a reconfiguration process. During reconfiguration, the mode pins are not sampled to check for MPC mode. Set this bit before reconfiguration to keep MPI available.
	R/W	3	REPEAT_RDBK	Active high. Inhibits auto-increment of the readback address (0x00014) when the readback data register (0x00018) is read.
	R/W	4	SYS_RD_CFG	Active-high. Initializes the readback logic.
		5		Unused
	R/W	6	UMI_RST	Active high. Asserts system bus reset. Can only be set by user master. Writing to this register bit is only active if "Systembus Reset by User Master" is set in the IPexpress™ GUI for system bus.
	R/W	7	MPI_RST	Active high. Asserts system bus reset. Can only be set by MPI. Writing to this register bit is only active if "Systembus Reset by MPI" is set in the IPexpress GUI for system bus.

Table 4. System Bus Register Map (Continued)

Absolute Address	Type R=Read W=Write	Bits	Name	Description
0x00009	R/W	0		Reserved
	R/W	1	UMI_LOCK	Active high, locks the internal system bus for use by the UMII. Used for multi-cycle operations that must retain bus ownership. Only the UMI can write this bit.
	R/W	2	MPI_LOCK	Active high, locks the internal system bus for use by the MPI. Used for multi-cycle operations that must retain bus ownership. Only the MPI can write this bit.
		3		Unused
	R/W	4	PRGM_UMI	UMI configuration request. Active high, forces reconfiguration of the FPGA logic using the mode specified on the MODE pins during initial power up. Has to be released before sending the bitstream.
	R/W	5	PRGM_MPI	MPI configuration request. Active high, forces reconfiguration of the FPGA logic using the mode specified on the MODE pins during initial power up. Has to be released before sending the bitstream.
	R/W	6	SYS_DAISY	Enables bitstream daisy chaining when configuring more than one device via MPI
	R/W	7	SYS_GSR	Active high, asserts the global set/reset
0x0000A	R/W	0:3	EBR_EXP	For pre-configuration usage of these bits
		4		Unused
		5	MPI_PAR_CHK	Enables MPI to check parity errors for write transfers if MPI parity bus is enabled
	R/W	6		Reserved. Must be written to ‘0’.
	R/W	7	MPI_DFA_EN	When set to 1, this bit enables any fabric DFA controller to use the MPI outputs pads
0x0000B				Unused
0x0000C-0x0000D	**R**	**Status Registers**		
0x0000C	R	0:1	ERR_FLAG	In the event of an error during device configuration these bits will indicate the nature of the error. [0:1]: • 00 - no error • 01 - checksum error indicates one or more corrupt bits in bitstream • 10 - device ID error indicates bitstream does not match target • 11 - framing/alignment error. Data bits may be reversed.
	R	2	INIT_N	Reflects the state of the INITN I/O pad.
	R	3	DONE	Reflects the state of the DONE I/O pad.
	R	4	CFG_DATA_LOST	Indicates that some initialization data was lost because configuration encountered long wait states or too many retries when initializing EBRs/ASB
	R	5:6	CFG_BUSI_ERR	If an internal system bus error occurs during configuration the error is captured in these two bits. The address of the first error in captured in the bus error address register (0x00024). [bit5 bit6]: 00 - no errors 01 - invalid response code 10 - one error occurred 11 - multiple errors occurred
	R	7	RDBK_AOR_ERR	Active-high readback address out of range error alarm.

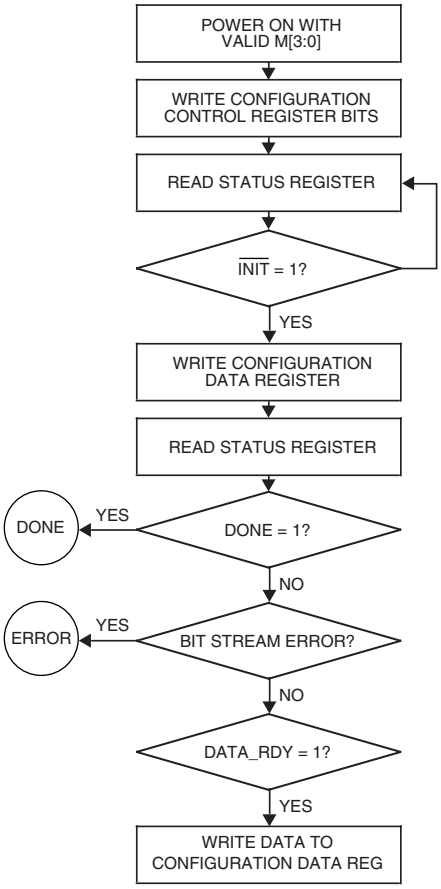
Table 4. System Bus Register Map (Continued)

Absolute Address	Type R=Read W=Write	Bits	Name	Description
0x0000D	R	0:1	WDATA_SIZE	Reflects the HSIZE [0:1] size during write transfers to the configuration data register
	R	2	EBR_BIT_ERR	A 1 indicates the bitstream of system bus configuration contains errors in the initialization data for EBRs.
	R	3	ASB_BIT_ERR	A 1 indicates the bitstream of system bus configuration contains errors in the initialization data for ASB.
		4:5		Unused
	R	6	RDATA_RDY	Active-high indicates that data is pending in the readback data register (0x00018).
	R	7	WDATA_ACK	Active-high indicates that the configuration logic is ready for data to be written into the configuration data register (0x0001C).
0x0000E - 0x0000F				Unused
0x00010	**Interrupt Cause Register**			
		0		Unused.
	R	1	PCS_IRQ	Active-high, interrupt request from the PCS interface. Write 1 to clear this bit.
	R	2	MPI_IRQ	Active-high, interrupt request from the MPI. Write 1 to clear this bit.
	R	3	CFG_ERR_IRQ	Active-high, indicates that the ERR_FLAG bits in the status register were changed during device configuration.
	R	4	CFG_DATA_IRQ	Active-high, interrupt request from the configuration logic requesting another word/byte of data. Write 1 to clear this bit.
	R	5	UMI_IRQ	Active-high, interrupt request from the User Master interface (UMI_IRQ). Write 1 to clear this bit.
	R	6	USI_IRQ	Active-high, interrupt request from the User Slave interface (USI_IRQ). Write 1 to clear this bit.
	R	7	USER_IRQ	Active-high, interrupt request from the USR_IRQ_IN signal. Write 1 to clear this bit.
0x00011				Unused
0x00012	**USER Interrupt Enable Register** (Only UMI can write to this register)			
		0	EN_IRQ_USER	Unused
	R/W	1		Logic 1 enables the PCS interrupt bit from address 0x00010 to generate an interrupt to the USR_IRQ_OUT port.
	R/W	2		Logic 1 enables the MPI interrupt bit from address 0x00010 to generate an interrupt to the USR_IRQ_OUT port.
	R/W	3		Logic 1 enables the CFG_ERR interrupt bit from address 0x00010 to generate an interrupt to the USR_IRQ_OUT port.
	R/W	4		Logic 1 enables the CFG_DATA master interrupt bit from address 0x00010 to generate an interrupt to the USR_IRQ_OUT port.
	R/W	5		Logic 1 enables the USER_MSTR interrupt bit from address 0x00010 to generate an interrupt to the USR_IRQ_OUT port.
	R/W	6		Logic 1 enables the USER_SLAVE interrupt bit from address 0x00010 to generate an interrupt to the USR_IRQ_OUT port.
	R/W	7		Logic 1 enables the USER_IRQ interrupt bit from address 0x00010 to generate an interrupt to the USR_IRQ_OUT port.

Table 4. System Bus Register Map (Continued)

Absolute Address	Type R=Read W=Write	Bits	Name	Description
0x00013	**MPI Interrupt Enable Register** (Only MPI can write to this register)			
	R/W	0	EN_IRQ_MPI	Unused.
	R/W	1		Logic 1 enables the PCS interrupt bit from address 0x00010 to generate an interrupt to the MPI_IRQ_N pin.
	R/W	2		Logic 1 enables the MPI interrupt bit from address 0x00010 to generate an interrupt to the MPI_IRQ_N pin.
	R/W	3		Logic 1 enables the CFG_ERR interrupt bit from address 0x00010 to generate an interrupt to the MPI_IRQ_N pin.
	R/W	4		Logic 1 enables the CFG_DATA master interrupt bit from address 0x00010 to generate an interrupt to the MPI_IRQ_N pin.
	R/W	5		Logic 1 enables the USER_MSTR interrupt bit from address 0x00010 to generate an interrupt to the MPI_IRQ_N pin.
	R/W	6		Logic 1 enables the USER_SLAVE interrupt bit from address 0x00010 to generate an interrupt to the MPI_IRQ_N pin.
	R/W	7		Logic 1 enables the USER_IRQ interrupt bit from address 0x00010 to generate an interrupt to the MPI_IRQ_N pin.
0x00014 - 0x00017	R/W	0:13	CFG_RDBK_ADDR	Configuration memory readback address register (14 bits). Bits [14:31] are reserved.
0x00018 - 0x0001B	R/W	0:31	CFG_RDBK_DATA	Configuration memory readback data register
0x0001C - 0x0001F	R/W	0:31	CGG_DATA	Configuration data register
0x00020 - 0x00023	R	0:31	TRAP_ADDR	24-bit trap address register. Configuration trapped in the address bus is stored here when bitstream initialization for EBR/ASB is lost
0x00024 - 0x00027	R	0:31	BUSI_ERR_ADDR	Bus error address register contains the address of the first configuration error. Indicated by the CFG_ERR bits of register 0x0000C.
0x00028 - 0x0002B	R	0:31	READ_WORD1	Read Only Word #1. Read Only Register with content defined via IPexpress
0x0002C - 0x0002F	R	0:31	READ_WORD2	Read Only Word #2. Read Only Register with content defined via IPexpress
0x00030 - 0x00033	R	0:31	READ_WORD3	Read Only Word #3. Read Only Register with content defined via IPexpress
0x00034 - 0x00037	R	0:31	READ_WORD4	Read Only Word #4. Read Only Register with content defined via IPexpress
0x00038 - 0x0003B	R	0:31	READ_WORD5	Read Only Word #5. Read Only Register with content defined via IPexpress
0x0003C - 0x0003F	R	0:31	READ_WORD6	Read Only Word #6. Read Only Register with content defined via IPexpress

Figure 10. MPI Configuration Flow



In general, a PROM image of the configuration bitstream is generated in EXO (S-record) or MCS (Intel*) format and linked into the firmware image during the compilation process. PROM images can be created using Lattice ispVM® software. At power-up, the LatticeSC device examines the MODE pins, activates the MPI in 8, 16, or 32-bit interfaces and waits for firmware to control and write the configuration data into the device.

If a configuration error occurs during the configuration process, the INIT signal is forced low, the DONE pin will not go high, and the error flags bits are set in the status registers (0x0000C, 0x0000D). The firmware must assert PRGM_MPI (0x00009) to reinitialize the configuration logic before attempting to reconfigure the device. This sequence must be done to clear the interrupt condition caused by the configuration error.

A device ID error indicates that firmware attempted to load a bitstream that was generated for an FPGA other than the target device. A checksum error is generated if correct framing is detected, but one or more bits in the bitstream are corrupted. A framing/alignment error occurs most often when the data bits are reversed in the bitstream generation or firmware compilation processes. Another common indication of this is that the complete bitstream is sent without errors, but DONE does not go high even after several dummy bytes are written into the configuration data register (e.g., the start of frame sequence was never detected).

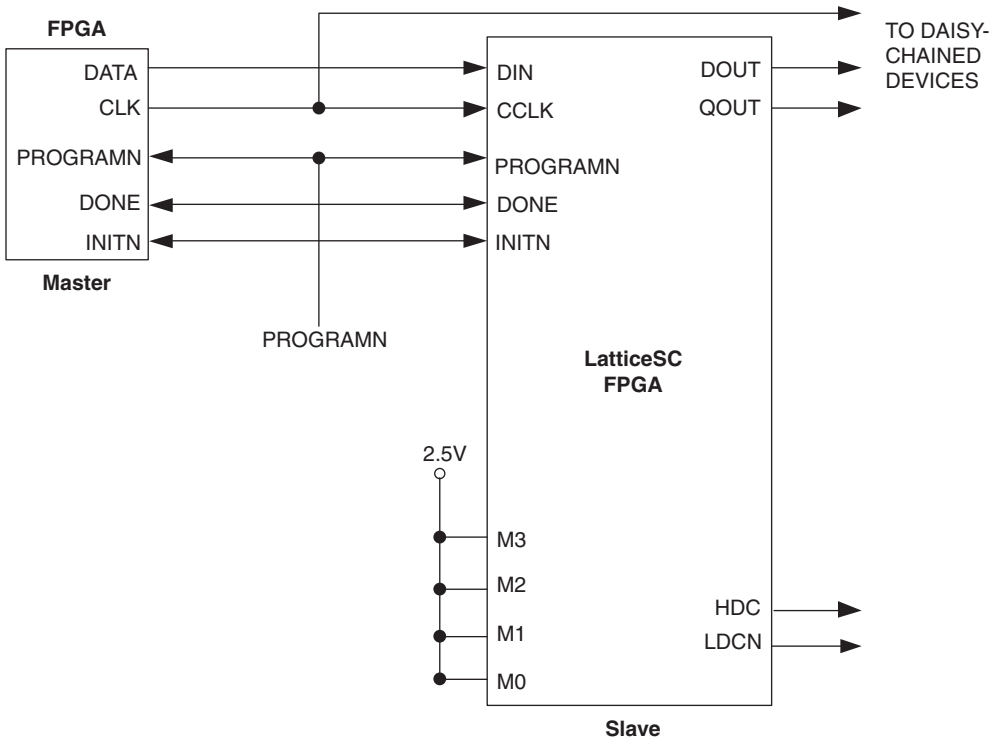
During configuration, initialization errors on the internal system bus are logged in the status register, but they do not prevent the configuration process from attempting to be completed. The first address of a failure is captured in the bus error address register (0x00024). If bus errors occur, the system may appear to be functioning, but block RAM or ASB control elements may not have been properly initialized. It is left to the FPGA designer to determine if bus initialization errors during configuration can be tolerated by the design.

The control and status registers are part of the embedded FPGA system bus. Please refer to Lattice technical note *LatticeSC System Bus* for the complete register mapping, addressing and further descriptions of these registers.

Slave Serial Mode

The slave serial mode is primarily used when multiple FPGAs are configured in a daisy chain. A device in the slave serial mode can be used as the lead device or daisy chain device. Figure 11 shows the connections for the slave serial configuration mode.

Figure 11. Slave Serial Configuration Mode Diagram

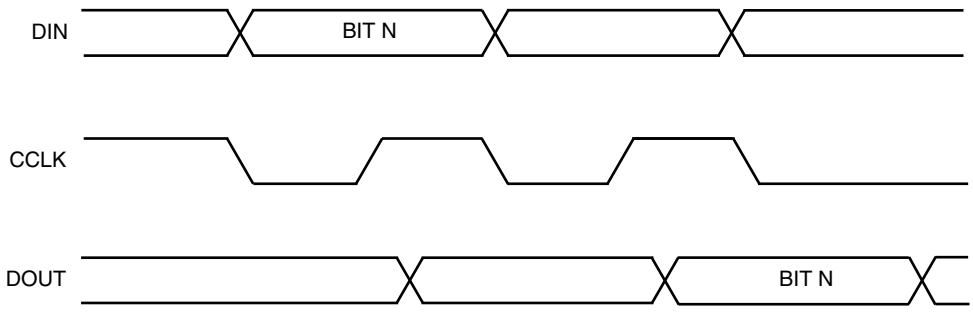


Dedicated Slave Serial Pins	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	I	CCLK is input synchronous with the data on DIN
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	PROGRAMN is an active-low input that forces the restart of configuration. This pin always has an active pull-up.

Dedicated Slave Serial Pins	I/O	Description
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.
CFGIRQN	O	During JTAG, slave, master, and asynch. peripheral configuration, assertion on this CFGIRQN (active low) indicates error(s) for block RAM or MACO initialization. A programmable pull-up resistor is available on this pin.
M[3:0]	I	During power-up and initialization, M[3:0] are used to select the configuration mode with their values latched on the rising edge of INITN. After configuration these pins are available as user defined inputs. Mode Pin Setting: M[3]=1, M[2]=1, M[1]=1, M[0]=1
Dual-Purpose Pins: After configuration, these pins are user-programmable I/O pins		
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
LDCN	O	Low During Configuration is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
DOUT	O	DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
QOUT	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. Data out on QOUT changes on the rising edge of CCLK.
D[0] or DIN	I	D0 also known as DIN is the data input.

The configuration data is provided into the FPGA's DIN (also known as D0) input synchronous with the configuration clock CCLK input. After the FPGA has completely loaded its configuration data, it retransmits the incoming configuration data on DOUT and QOUT. CCLK is routed into all slave serial mode devices in parallel. Multiple slave FPGAs can be loaded with identical configurations simultaneously. This is accomplished by loading the configuration data into the DIN inputs in parallel. The timing diagram for Slave Serial configuration is shown in Figure 12.

Figure 12. Slave Serial Configuration Timing Diagram

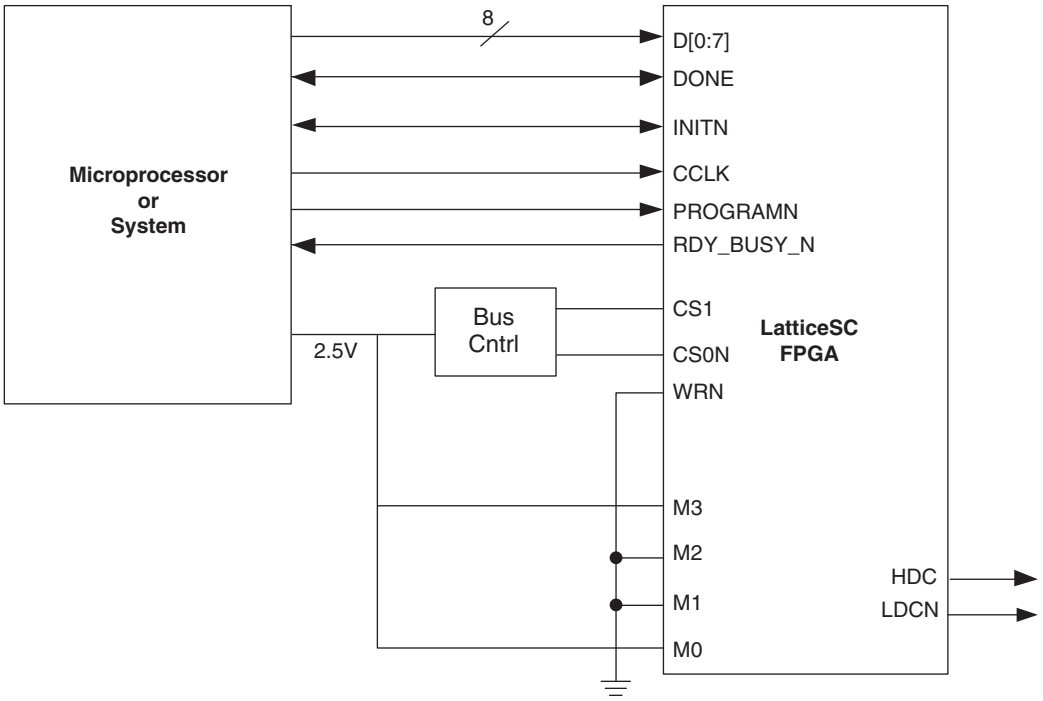


Slave Parallel Mode

The slave parallel mode is essentially the same as the slave serial mode except that 8 bits of data are input on pins D[0:7] for each CCLK cycle. Due to eight bits of data being input per CCLK cycle, the DOUT pin does not contain a

valid bitstream for slave parallel mode. As a result, the lead device cannot be used in the slave parallel mode in a daisy-chain configuration. Slave Parallel mode can be used to accomplish this function.

Figure 13. Slave Parallel Configuration Mode Diagram

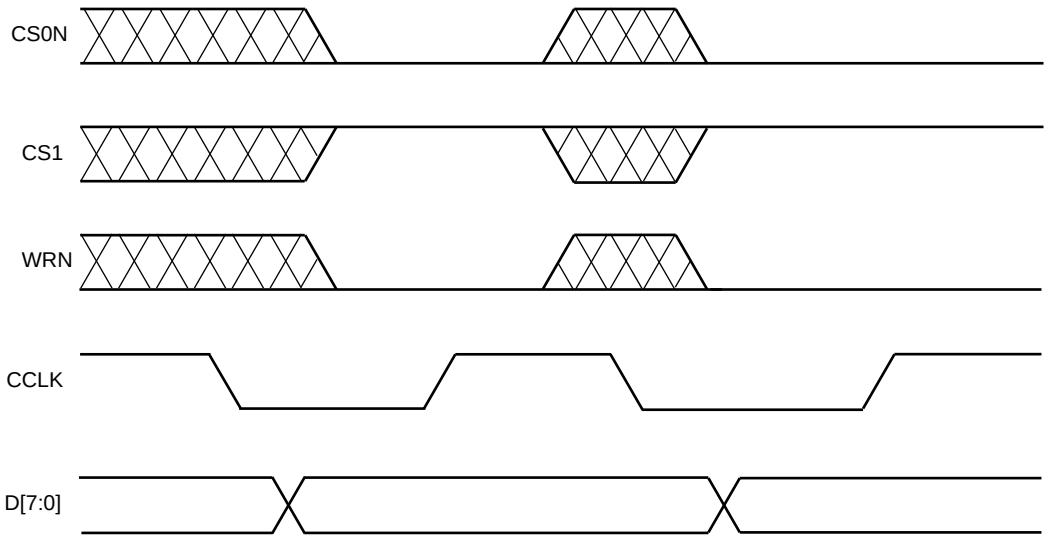


Dedicated Slave Parallel Pins	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	I	CCLK is input synchronous with the data on DIN
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	PROGRAMN is an active-low input that forces the restart of configuration. This pin always has an active pull-up.
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.

Dedicated Slave Parallel Pins	I/O	Description
CFGIRQN	O	During JTAG, slave, master, and asynch. peripheral configuration, assertion on this CFGIRQN (active low) indicates error(s) for block RAM or MACO initialization. A programmable pull-up resistor is available on this pin.
M[3:0]	I	During power-up and initialization, M[3:0] are used to select the configuration mode with their values latched on the rising edge of INITN. After configuration these pins are available as user defined inputs. Mode Pin Setting: M[3]=1, M[2]=0, M[1]=0, M[0]=1
Dual-Purpose Pins: After configuration, these pins are user-programmable I/O pins		
DOUT	O	During configuration, DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
QOUT	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. Data out on QOUT changes on the rising edge of CCLK.
BUSYN	O	For slave parallel mode, active low BUSYN inhibits the external host from sending new data. This output is used by slave parallel and master serial modes only for decompression.
CS0N CS1	I	The FPGA is selected when CS0N is low and CS1 is high. For parallel daisy chaining, CS0N can be used in conjunction with the CEON output of the master device.
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
LDCN	O	Low During Configuration is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
D[0:7]	I	Data inputs that receive data from master.

Figure 13 is a schematic of the connections for the slave parallel configuration mode. WRN and CS0N are active-low chip select signals, and CS1 is an active-high chip select signal. These chip selects allow the user to configure multiple FPGAs in slave parallel mode using an 8-bit data bus common to all of the FPGAs. These chip selects can then be used to select the FPGAs to be configured with a given bitstream. The chip selects must be active for each valid CCLK cycle until the device has been completely programmed. They can be inactive between cycles but must meet the setup and hold times for each valid positive CCLK. D[0:7] of the FPGA can be connected to D[0:7] of the microprocessor if a standard PROM file format is used. The BUSYN handshake signal is used when programming with a compressed formatted bitstream. The timing diagram shown in Figure 14.

Figure 14. Slave Parallel Timing Waveform



Serial Peripheral Memory Interface

The SPI (Serial Peripheral Interface) configuration modes permit programming from SPI Flash memories. The SPI interface consists of three pins (SI, SO and SCK) and a fourth pin for Chip Select (SCS). Three other pins SIZE[2:0] are used to select the number of Flash devices on the interface. When the mode pins have selected SPI or SPIX configuration mode, the sysCONFIG pins are no longer used for their traditional purposes and are mapped for SPI Flash memory programming. Figure 15 shows a typical SPI PCB interface.

The SPI approach provides a number of advantages over traditional FPGA boot memory:

1. SPI devices are available from multiple vendors ensuring stable supply
2. The cost of SPI memory is up to 75% less than traditional FPGA boot memory
3. SPI memory is available in space saving 8-pin packages that are considerably smaller than packages used for traditional FPGA boot memory

Like all boot memories, SPI Serial Flash needs to be loaded with the FPGA configuration data. There are three options for programming an SPI memory used in conjunction with a LatticeSC device. The SPI memory can be configured off-board using a stand-alone programmer, on-board using its SPI interface, or on-board via JTAG through the LatticeSC device.

The SPI interface is a 4-wire serial interface comprised of the following signals:

- **CS – Chip Select, Input:** Enables and disables device operation. When high the device is at standby power levels and the output is tri-stated. When low the device powers up and instructions can be written to and data read from the device.
- **CLK – Serial Clock, Input:** Provides timing for the interface. The Serial Data Input (DI) is latched on the rising edge of CLK. Serial Data Output (DO) changes after the falling edge of CLK.
- **DI – Serial Data In, Input:** When the device is enabled (CS is low) this pin allows instructions, addresses, and data to be serially written to the device. Data is latched on the rising edge of the CLK.
- **DO – Serial Data Out, Output:** When the device is enabled (CS is low) this pin allows data and status to be serially read from the device. Data is shifted out on the falling edge of the CLK.

Figure 15. SPI Programming Interface Diagram (with 2-SPI Memory Devices Used)

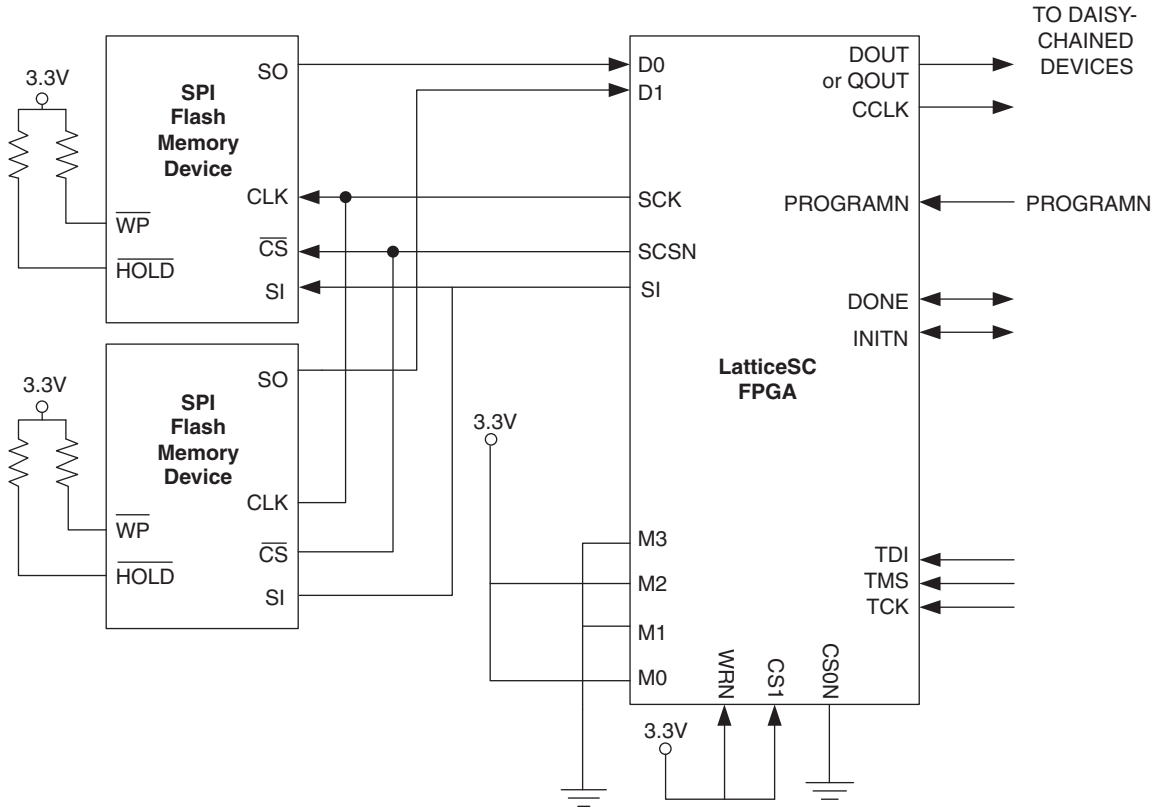


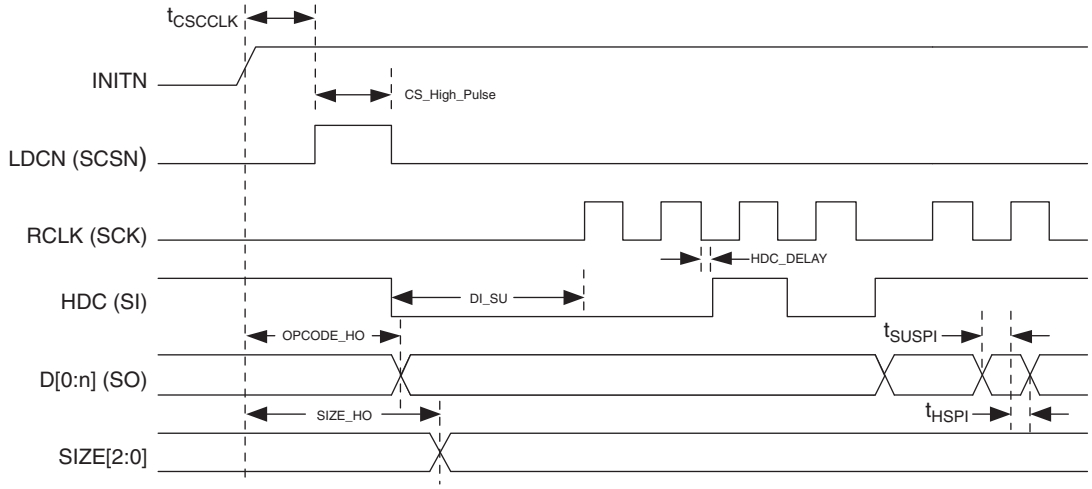
Table 5. Serial Peripheral Interface

Dedicated SPI Pins	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	O	CCLK is output for daisy-chaining
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	PROGRAMN is an active-low input that forces the restart of configuration. This pin always has an active pull-up.
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.

Table 5. Serial Peripheral Interface (Continued)

Dedicated SPI Pins	I/O	Description
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.
CFGIRQN	O	During JTAG, slave, master, and asynch. peripheral configuration, assertion on this CFGIRQN (active low) indicates error(s) for block RAM or MACO initialization. A programmable pull-up resistor is available on this pin.
M[3:0]	I	During power-up and initialization, M[3:0] are used to select the configuration mode with their values latched on the rising edge of INITN. After configuration these pins are available as user defined inputs. Mode Pin Settings: SPI M[3]=0, M[2]=1, M[1]=0, M[0]=1 SPIX M[3]=0, M[2]=1, M[1]=0, M[0]=0
Dedicated JTAG Pins: Power supplied via dedicated VCCJ (1.8V, 2.5V or 3.3V) supply. These pins are used for on-board programming of SPI Flash memory devices.		
TDO	O	JTAG data output.
TCK	I	JTAG clock input.
TDI	I	JTAG data input.
TMS	I	JTAG mode selection input.
Dual-Purpose Pins: After configuration, these pins are user-programmable I/O pins		
DOUT	O	During configuration, DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
QOUT	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. Data out on QOUT changes on the rising edge of CCLK. During parallel-chaining configuration, active low CEON enables the cascaded slave device to receive bitstream data.
SCK	O	For SPI modes, clock generated by the device and connected to the CLK input of the Flash memory.
SI	O	For SPI modes, this pin is used to download the read command and initial read address into the Flash memory device on the falling edge of SCLK. This pin will be connected to SI of the memory. If the SPI mode is used the 8-bit instruction code 0x03 will be downloaded followed by a 24-bit starting address of 0x000000 or a non-zero stat address for partial reconfiguration. If the SPIX mode has been selected the 8-bit instruction captured on D[7:0] at power-up will be shifted in followed by a 32-bit starting address of 0x000000.
SCSN	O	For SPI modes, this is an active low chip select for Flash memories. It will go active after INITN goes high but before SCK begins. During power up SCS will be low. Once INITN goes high, SCS will go high for 100ns - 200ns after which time will go back low and configuration can begin. During the 100ns -200ns period the read instruction will be latched for SPIX mode.
SIZE[2:0] WRN, CS1, CS0N	I	Used to set the number of Flash memory devices used to simultaneously download data into the device. {WRN, CS1, CS0N} = 111 => 1 device, = 110 => 2 devices, 101 => 3 devices...000 => 8 devices.
D[7:0]	I/O	For SPI modes, Data input from the Flash memory device(s). These pins will be connected to the SO pins of the Flash memories. If only a single Flash memory is used the input used will be D[0]. Additional inputs, starting at D[1], are used as additional Flash memories are used. If the SPIX mode is used these pins provide the READ instruction code upon power-up.

Figure 16. SPI Programming Timing Waveform



SCK is used to drive the clock to the flash memory devices. The first rising edge of SCK is sent after SCS switches from high to low. The SCS high pulse width is determined by oscillator speed selection selected via the option selected by Bitgen. The SCS low pulse width varies depending on speed selection, the number of flash devices, and bitstream whether the bitstream is compressed.

The LatticeSC overloads sysCONFIG pins HDC, RCLK, and LDCN to the SI, SCK, and SCSN pins respectively for the SPI modes. This is done via the mode pin setup. These pins will no longer function according to their original function and are always dedicated to SPI programming before and after configuration. Data is read back from an SPI Flash memory by shifting in the 8-bit read command followed by a 24-bit address into SI. The read data will be clocked out onto the D[0:7] (SO) pin after the address and command have been shifted in. SI drives the serial-input of flash devices. The SI is high during configuration except for the first 128 cycles of SCK after INITN going high. In these 128 cycles, the first eight bits of SSI become the 8-bit read command (00000011 for SPI and user-defined on D[0:7] for SPIX), and the rest are all 0 for the initial read address.

The initial read address after power up starts at 0 for a full configuration. If the user only desires to access the Flash starting from another address, a non-zero address can be used. The non-zero start address must be initially set in the prior configuration bitstream. This type of flash indexing could also be used for a partial configuration application.

Configuration with an alternate starting address other than 00 can be stored in a 32-bit wide memory. If the device is reconfigured without initialization, the starting address specified in start address index will be shifted in. This allows for multiple configuration images to be stored in the same SPI Flash memory or memories if multiple devices are used. Start address index value will be latched upon the DONE signal going high so that it does not change during the re-configuration of the device. The memory cells storing the data can be re-written, providing a new starting address, but the address used for the current configuration sequence would not change.

The most common SPI memories use 0x03 as the read command and are directly supported with SPI mode. The majority of SPI Serial Flash on the market support a common read Operation Code (Op Code). The LatticeSC devices offer direct connection to these devices by hardwiring the read Op Code (03H) into the FPGA silicon. These devices are sometimes referred to as SPI3 devices because they support this common Read Op Code.

Other memories will select their own read command values and interface is interfaced using SPIX configuration mode. Some devices also support an Op Code for fast read and the SPIX mode can be used for this capability. In SPIX mode, the user-defined read command will be driven on D[0:7] when SCS (LDCN) is high. SPIX mode allows the LatticeSC to easily interface to SPI Serial Flash devices that support a different read Op Code. This can be done with pull-up and pull-down resistors on the PCB wired to the D[7:0] pins, telling the FPGA which Read Op Code that particular Flash device.

Prior to configuration, the LatticeSC FPGA has a default 10K pull-up resistor in the I/O buffer. This pull-up resistor may cause some excessive loading for the Flash memory drivers. In some cases there may be a need to add external buffering with some Flash memory devices to mitigate the loading concerns.

The inputs WRN, CS1, and CS0N are used to set the number of Flash memory devices used to simultaneously download data into the device. In SPI mode, these pins are used as the SIZE[2:0] function. When these pins are left unconnected, an on chip pull-up will default the Flash selection to 1.

Table 6. SPI Flash Selects

Number of Flash Devices	SIZE[2]WRN	SIZE[1]CS1	SIZE[0]CS0N
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0

INITN may be optionally connected to reset (active low) of flash memory devices that support a reset pin. CCLK is a continuously running clock for daisy-chaining slaves using either the QOUT or DOUT serial data outputs.

The ispJTAG port can be utilized to directly program the one Flash memory device. This is useful for changing the SPI Flash memory via JTAG. After receiving the correct JTAG instruction code, the LatticeSC device will send the programming input from the JTAG port to the appropriate flash device via the SI pins connected to the Flash SI device pins. The SPI programming 8-bit JTAG instruction register is loaded (11011011). The LatticeSC device creates the master clock for the Flash device for writing to the memory or reading from the memory. The rest of the FPGA can continue to operate in a user mode. During system operation the Flash device can be re-written and will be available to the FPGA on the next power-cycle or after toggling the PROGRAMN pin. However assertion of PROGRAMN will not interfere with the Flash being written through the JTAG port.

The LatticeSC supports reading from multiple Flash devices without any extra circuitry or additional logic. This capability is useful for large configuration bitstream loading such as needed for daisy chain designs. The use of the ispJTAG interface to program or write to multiple Flash devices requires additional steering circuitry to control the Flash device programming.

Refer to Lattice technical note number TN1100, SPI Serial Flash Programming Using ispJTAG in LatticeSC Devices for recommended procedures and software usage.

Daisy Chaining

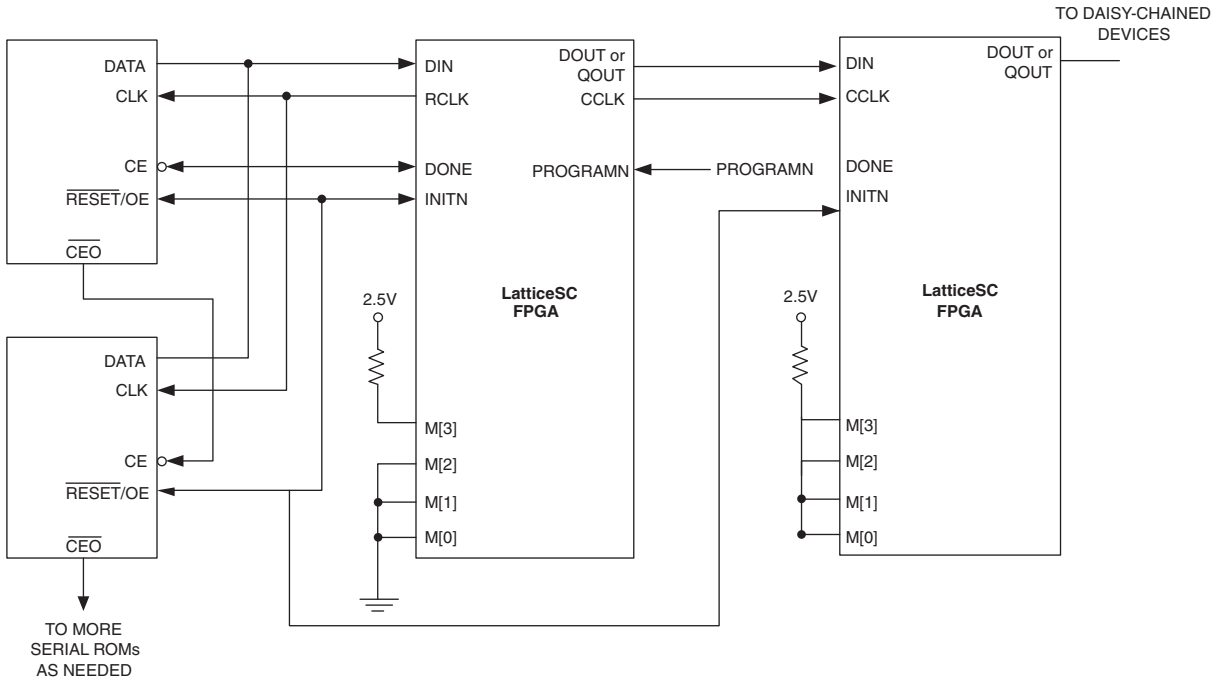
Multiple FPGAs can be configured using a daisy chain of FPGAs. Daisy chaining is available in 8-bit parallel and single bit serial modes. In LatticeSC devices, the term daisy chain defines the ability to pass through the programming data to downstream devices. Daisy chaining typically uses a lead FPGA and one or more FPGAs configured in slave mode. The lead FPGA can be configured in any mode except slave parallel mode for parallel wide chaining. Daisy chaining is available with the boundary scan and is discussed later.

For serialized daisy chaining the first FPGA can receive the bitstream either in serial or parallel format. The downstream FPGAs will receive data in serial format.

Table 7. Pins Used for Various Daisy Chaining Modes

Pin	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	O	CCLK is output from the master to the all the slave devices for daisy-chaining
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up. All DONE pins should be wired-anded together with daisy chaining
PROGRAMN	I	PROGRAMN is an active-low input that forces the restart of configuration. This pin always has an active pull-up. All PROGRAMN pins should be connected together for daisy chaining.
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration. All INITN pins should be connected together for daisy chaining.
RCLK	O	RCLK is a read clock output signal from the master device to an external memory.
DOUT	O	During daisy chain configuration, DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. This is used with Series 2, 3, and 4 FPGAs. Data out on DOUT changes on the rising edge of CCLK.
QOUT/CEON	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. This is used with other industry FPGAs and programmable products. Data out on QOUT changes on the rising edge of CCLK. During parallel-chaining configuration, active low CEON enables the cascaded slave device to receive bitstream data.
D[7:0]	I/O	For parallel daisy chain configuration, D[7:0] receives configuration data for both the master and slave devices, and each pin is pull-up enabled. For slave serial daisy chaining, D0 is the data input. Receives data from QOUT or DOUT.

Figure 17. Serial Daisy Chaining Method



One daisy chain scenario used with many legacy devices is shown in Figure 17. In this scenario, each FPGA reads and shifts the preamble and length count in on the positive-edge of CCLK and out on negative CCLK edges. An upstream FPGA that has received the preamble and length count outputs a high on DOUT until it has received the appropriate number of data frames so that downstream FPGAs do not receive frame start bit pairs. After loading and retransmitting the preamble and length count to a daisy chain of slave devices, the lead device loads its configuration data frames. The loading of configuration data continues after the lead device has received its configuration data if its internal frame bit counter has not reached the length count. When the configuration RAM is full and the number of bits received is less than the length count field, the FPGA shifts any additional data out on DOUT.

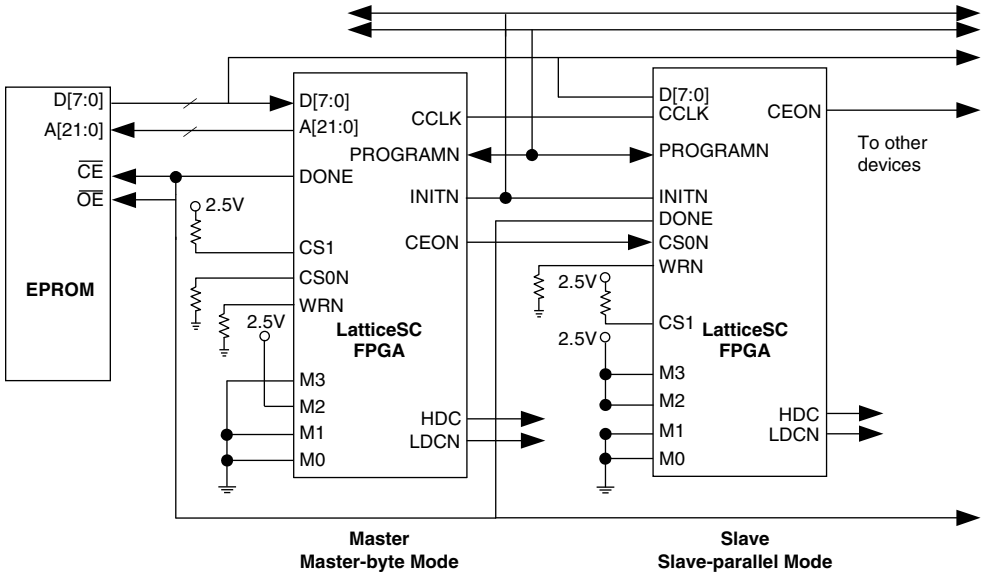
In other serial scenarios used with the latest family of devices, the QOUT pin is used when it is not desired to propagate the preamble with length count in daisy chaining. The output of QOUT is suppressed while DOUT is propagating the preamble. The data transmitted on QOUT is the same as DOUT in daisy-chaining modes.

A daisy chain operation can be done as an 8-bit wide parallel operation shown in Figure 18. The QOUT shares the same pad with CEON because CEON is only used in parallel-chaining modes (master byte and slave parallel). The CEON is used to handshake between chained devices when using a parallel data bus scheme. In this scenario, an upstream device signals to downstream devices via their chip select input when the data bus is available to them.

In master byte mode, the user should preference the bitstream to use DONE_EX=ON and WAKE_UP=1-7. This is needed to produce the correct finish of configuration between the master and slave device. These preferences are also needed for other serial daisy chaining scenarios where the data is sent via the QOUT output.

Other schemes may use a parallel interface to a PROM as shown in Figure 18. The first FPGA device generates the clock used to access the PROM. This FPGA master will interpret the length count and select the down chain devices via the CEON pin. This pin would typically be connected to CS0N of the following chained device. This will use a common CCLK and data bus as the appropriate devices are loaded.

Figure 18. Parallel Data Bus Daisy Chain Method



The configuration data is read into DIN of slave devices on the positive edge of CCLK, and also shifted out DOUT on the positive edge of CCLK. The generation of CCLK for the daisy-chained devices that are in slave serial mode differs depending on the configuration mode of the lead device. A master parallel mode device uses its internal timing generator to produce an internal CCLK at eight times its memory address rate (RCLK). The asynchronous peripheral mode device outputs eight CCLKs for each write cycle. If the lead device is configured in slave mode, CCLK must be routed to the lead device and to all of the daisy-chained devices.

Programming with JTAG

Single and multiple FPGAs can be configured through the JTAG ports. JTAG operation is available upon initial configuration when power-on reset or after pulling the program pin to reset the chip when the INITN pin is high. The mode pins are not needed for use with the JTAG programming feature. Configuration through the JTAG port conforms to the IEEE 1532 Standard. The boundary scan cells take control of the I/Os during any 1532 mode instruction. The boundary scan cells can be set to a pre-determined values whenever using the JTAG 1532 mode.

Table 8. sysCONFIG Pins

Dedicated sysCONFIG Pins	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	PROGRAMN is an active-low input that forces the restart of configuration. This pin always has an active pull-up.
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected (via a bitstream option) to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.

Table 8. sysCONFIG Pins (Continued)

Dedicated sysCONFIG Pins	I/O	Description
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.
CFGIRQN	O	During JTAG configuration, assertion on this CFGIRQN (active low) indicates error(s) for block RAM or MACO initialization. It is also the interrupt pin for MPI to external microprocessor. A programmable pull-up resistor is available on this pin.
Dedicated JTAG Pins: Power supplied via dedicated VCCJ (1.8V, 2.5V or 3.3V) supply		
TDO	O	JTAG data output.
TCK	I	JTAG clock input.
TDI	I	JTAG data input.
TMS	I	JTAG mode select input.
Dual-Purpose Pins: After configuration, these pins are user-programmable pins		
DOUT	O	During configuration, DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
QOUT	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. Data out on QOUT changes on the rising edge of CCLK.
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
LDCN	O	Low During Configuration is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.

All daisy-chained FPGAs are connected in series. Each FPGA reads and shifts the preamble and length-count in on the positive TCK and out on the negative TCK edges. An upstream FPGA that has received the preamble and length count outputs a high on TDO until it has received the appropriate number of data frames so that downstream FPGAs do not receive frame start bit pairs. After loading and retransmitting the preamble and length count to a daisy chain of downstream devices, the lead device loads its configuration data frames. The loading of configuration data continues after the lead device had received its configuration read into TDI of downstream devices. This is done on the positive edge of TCK, and shifted out TDO on the negative edge of TCK.

Partial Reconfiguration

All LatticeSC devices have been designed to allow a partial reconfiguration of the FPGA at any time. Partial configuration is a function permitting a modification to a prior programming bitstream without necessarily reprogramming the entire device configuration memory. Setting a bitstream option in the previous configuration bitstream permits the partial configuration in future downloads of a bitstream. This is accomplished by allowing the FPGA to not reset the entire configuration RAM during a reconfiguration. Only the configuration frames that are to be modified need to be rewritten, thereby reducing the configuration time.

During the partial reconfiguration operation, after toggling the PROGRAMN pin, INITN will pulse high-low-high indicating the programming can commence and the DONE will output low as the partial bitstream is transmitted to the device. Until a new bitstream is supplied to the device, the prior loaded bitstream will remain un-altered. During this operation, other prior bitstream options are also available that keep the user I/O active and allow one portion of the FPGA to remain in operation while a partial reconfiguration is being done. If this is done, the user must be careful to not cause contention between the two configurations (the bitstream resident in the FPGA and the partial re-configuration bitstream) as the second reconfiguration bitstream is being loaded.

Bitstream Generation Software Usage

This section describes the settings for bitstream generation performed by the ispLEVER software program Bitgen. These options are only available to be controlled through the Bit Generation Software tool. Other programming options can be controlled by software preferences and attributes. These will be discussed later in the document.

Bit Generation takes a fully routed Physical Design (.ncd file) as input and produces a configuration bitstream (bit images). The bitstream file contains all of the configuration information from the Physical Design defining the internal logic and interconnections of the FPGA, as well as device-specific information from other files associated with the target device. The data in the bitstream can then be downloaded directly into the FPGA's memory cells or used to generate files for PROM programming (using a separate program, ispVM). Please refer to the ispVM documentation for details on creating PROM image files.

The Bitstream Generation options are listed in Table 9 and descriptions for each option follow the table.

Table 9. Bitstream Generation (Bitgen) Options

Switch Option	Option Name	Description
-d		Don't Run DRC (Design Rules Checker)
-j		Don't create bit file
-b		Create "rawbits" file
-q		Create bit output without header comments (special use only)
-w		Overwrite existing output file
-m <format>		Create "mask" and "readback" file, valid format value is 0
		<format>0: Output files in ASCII
		<format>1: Output files in binary
-g <option:value>		Set option to value, options are (first is default):
	option	value
	LengthBits	24, 32
	CfgMode	Enable, Serial
	ExternalClk	No, Yes
	OutputsCfg	TriStateDuringReConfig, DriveDuringReConfig
	RamCfg	ClearDuringReConfig, NoClearDuringReConfig
	RegisterCfg	ResetDuringReConfig, NoResetDuringReConfig
	StartupClk	Cclk, UserClk
	Address	Increment, Explicit
	ZeroFrames	No (skip zero frames), Yes (output zero frames)
	SysbusCfg	ResetDuringReConfig, NoResetDuringReConfig
	SysbusclkCfg	ResetDuringReConfig, NoResetDuringReConfig
	WaitStateTimeout	5, 0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
	GrantTimeout	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
	SPI_RECONFIG_START_ADDRESS[31:0]	0x00000000-0xFFFFFFFF (default 0)

Note: The first value is the default value

Disable DRC (-d)

This switch causes Bitgen to not run the design rules checker (DRC). Without disabling DRC with this switch, Bitgen will run a physical design rule check and save the output to the Bit Generation report (.bgn file). Running DRC before a bitstream is produced will detect any errors that could cause the FPGA to function improperly. If no fatal errors are detected, it will produce a bitstream file.

Raw Bits (-b)

This switch causes Bitgen to create a Raw Bit (.rbt) file instead of a binary file (.bit). Do not use -b with -m if you want both a Raw Bits file and a mask file (see the -m mask command below). Instead, run Bitgen twice – once with the -b option and once with the-m option.

The Raw Bit File is a text file containing ASCII ones and zeros representing the bits in the bitstream file. If you are using a microprocessor to configure a single FPGA, you can include the Raw Bit file in the source code as a text file to represent the configuration data. The sequence of characters in the Raw Bit file is the same as the bit sequence that will be written into the FPGA.

Mask (-m)

This switch causes Bitgen to create a Mask file (.msk). The Bitgen options are either 0 or 1. Option 0 will generate an ASCII formatted file and 1 a binary file. The Mask file is used to compare relevant bit locations for executing a read-back of configuration data contained in an operating FPGA. The mask file is for the user to compare against read-back data from the device after downloading the bitstream. The ASCII mask file will contain 1's and 0's, and X's.

The file only contains all FPGA data frames. Its format matches the format of .rbt file. It contains no header, ID frames, address frames and any preloaded frames.

For RAM modes the file will include 0 values since it will always readback 0 when in RAM mode.

Mask file will have the dual-port and single-port LUT, EBR, and unused bits are masked out.

OutputsCfg

This directs the bitstream to program the FPGA to drive outputs (not be tristated) during configuration/reconfiguration when set to DriveDuringReConfig. The only time this should be enabled is for partial reconfiguration when it is desired to keep the device operational during reconfiguration, and then care must be taken to ensure that no user outputs are placed at pins required for configuration. TriStateDuringReConfig is defaulted and insures no contention during programming.

RamCfg

When this switch is set it directs the bitstream to reinitialize the device configuration starts. When this is set to NoClearDuringReConfig, it directs the bitstream to program the device to retain the current configuration and allows for additional bitstream configuration.

RegisterCfg

When this is ResetDuringReConfig, the device will reset the PFU registers before configuration. The only time this should be NoResetDuringReConfig is for partial reconfiguration when it is desired to keep the device operational during reconfiguration.

StartupClk

This switch selects whether to use CCLK or a User clock during startup. A User clock is needed to synchronize startup to a system clock. NO is set as default and selects the device generated startup clock.

Address

This switch controls whether the configuration addressing will be in Incremental mode or Explicit mode. The two modes are similar except that Auto-increment mode uses assumed address increments to reduce the Bitstream size, and Explicit mode uses an optional address frame.

In Incremental mode, data frames are written out one after another in increasing addresses. This mode reduces the size of the bitstream.

In Explicit mode, every address frame is written into the bitstream, followed by the data frame for each address. Explicit mode is good for partial reconfiguration where only some addresses of the array are to be written.

Bitgen supports the ability to format a bitstream in either incremental or explicit mode. The LatticeSC family does not require a bitstream to be entirely one format or the other. The configuration data can include address information or not. It will increment when an address is not explicitly defined.

Figure 19. Increment Mode Data Format

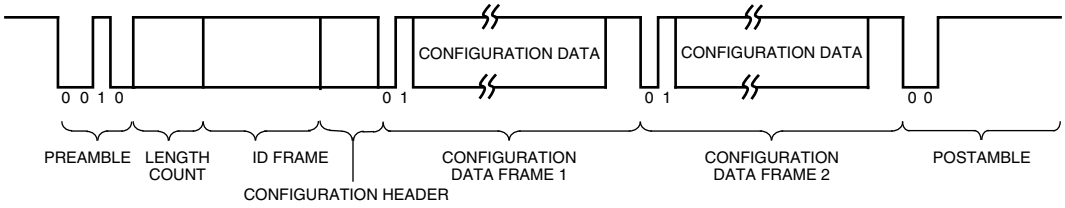


Figure 20. Explicit Mode Data Format

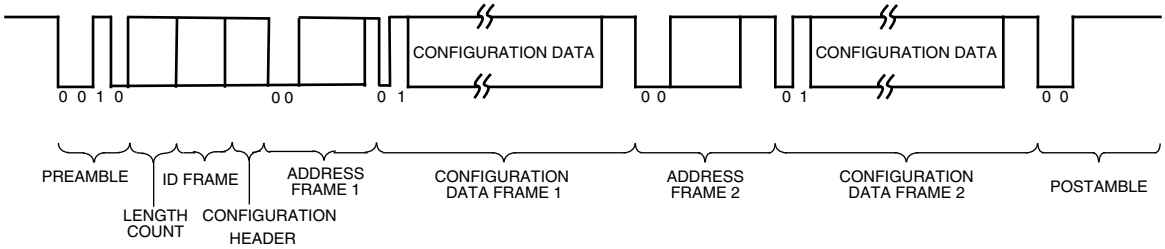


Table 10. Configuration Address Mode Frame Contents

Name	Contents	Description	Notes
FPGA Header	1111 0010	Mandatory Header	
	1111 1111	8 stop bits	
FPGA Address Frame	00	Address Frame Header	Not included in Incremental Mode
	14-bit Address	14-bit Address	
	Checksum	8-bit checksum	
	1111 1111	Stop bits to separate frames	
FPGA Data Frame	01	Frame header	
	Alignment Bits	0-bits added to reach byte boundary at end of data frame	
	Data bits	Data bits for single frame	
	Checksum	8-bit checksum	
	1111 1111	Stop bits	
Post Amble for FPGA	00 or 10	Post amble (00=finish, 10 more bits coming)	
	1111111 11111111	Dummy address	
	1111 1111 1111 1111	16 stop bits	

ZeroFrames

This switch controls whether the bitstream contains null data frames when in Incremental mode. If Yes is selected, every data frame is written out, including zero data frames, and no address frames are written.

If No is selected, all sequential non-zero data frames are written without address frames and zero data frames are skipped. When a zero data frame is skipped, the address frame for the next non-zero data frame is written. Selecting No for this switch is good for reducing bitstream size when you know you'll always do a full configuration (when all the FPGA is already cleared to zeros).

Selecting No while being in Explicit addressing mode is not allowed. A warning will be generated the ZeroFrames: No is not compatible with Explicit addressing, and the option will be ignored.

SystemBusCfg

When this switch is set to Reset, the System Bus is reset at configuration. If this switch is set to NoReset, the System Bus is not reset at configuration.

SystemBusclkCfg

When this switch is set to Reset, the System Bus Clock is reset at configuration. If this switch is set to NoReset, the System Bus Clock is not reset at configuration.

WaitStateTimeout

This switch controls an internal System Bus timeout counter. Table 11 shows how this switch programs the counter to timeout after the given number of HCLK cycles goes by while the System Bus is in Wait States. For example, if the switch is at 5 and the HCLK is running at 50 MHz, the System Bus will timeout after 2^10 * 20 ns ~= 20 microseconds of wait states.

Table 11. Time-out Values

Switch Value	System Bus Cycles
0	Forever (never time out)
1	2^2
2	2^4
3	2^6
4	2^8
5	2^10
6	2^12
7	2^14
8	2^16
9	2^18
10	2^20
11	2^22
12	2^24
13	2^26
14	2^28
15	2^31

GrantTimeout

This switch controls an internal System Bus grant counter. The table for the Wait State Timeout switch above applies for this switch also, showing how the switch setting programs the counter to timeout after the given number of System Bus (HCLK) cycles goes by while there is a grant requested on the System Bus. The Timeout counters provide a way for avoiding a system lockup if something goes wrong on the bus.

SPI_RECONFIG_START_ADDRESS

When the device is initially configured using an external SPI Flash memory device, the start address is assumed to be 0x00000000. An alternate starting address can be stored in a 32-bit register controlled by the SPI_RECONFIG_START_ADDRESS Bitgen option. If the device is re-configured without initialization, the starting address specified by the prior bit stream will be used. This allows for multiple configuration images to be stored in the same SPI Flash memory or memories if multiple devices are used.

The user-defined address is latched upon the DONE signal going high so that it does not change during the re-configuration of the device. The memory cells storing the data can be re-written, providing a new starting address, but the address used for the current configuration sequence would not change. This eliminates the need to preserve the memory contents by using the RAMCFG option.

Some Flash devices use 24-bit addressing. The software can interpret the 24-bit address and base address. For example, if the user specifies 24-bit address and base address 1, the user should set SPI_RECONFIG_START_ADDRESS =0x000001.

Software Preference Control

In order to control the configuration of the LatticeSC device other than by the default bitstream settings, software preferences can be used. These preferences will appropriately modify the settings accordingly. Below is a simple example of the preferences available for setting up configuration settings using the ispLEVER software flow.

Example:

SYSCONFIG CONFIG_MODE=SLAVE_SERIAL DONE_PULL=ON DONE_EX=OFF MCCLK_FREQ=1 CONFIG_SECURE=OFF WAKE_UP=21 PWRSAVE=OFF

The attributes available for sysCONFIG settings are listed below.

Attributes (the **BOLD** value is the default)

CONFIG_MODE	=	**JTAG_NONE**, NONE, SLAVE_PARALLEL, MASTER_SERIAL, MASTER_PARALLEL, MASTER_BYTE, MPI8, MPI32, MPI16, ASYNC_PERIPHERAL, SPI, SPIX
DONE_PULL	=	**ON**, OFF
DONE_EX	=	**OFF**, ON
MCCLK_FREQ	=	**1**, 2, 4, 8, 16, 33, 66, 133 # in MHz
CONFIG_SECURE	=	**OFF**, ON, ONCE
WAKE_UP	=	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, **21**, 22, 23, 24, 25
PWRSAVE	=	**OFF**, ON

CONFIG_MODE

CONFIG_MODE attribute controls the locking of configuration pins for ispLEVER. The software will query the CONFIG_MODE attribute and avoid using any of the defined pins needed for the particular configuration mode.

DONE_PULL

Options are to either ON, the DONE pin with an internal 100 K ohm resistor or OFF, meaning do not use an internal pull-up resistor.

The default for DONE_PULL is ON.

DONE_EX

The DONE Pin can be selected to delay wake up. The device can wake up on its own after the EXTERNAL DONE Bit is set or wait for the EXTERNAL DONE pin to be externally driven. The DONE_EX option controls whether or not an external DONE signal gates the rest of the wake up sequence. This is used to synchronize the startup sequence to the external DONE signal. If DONE_EX = ON then only the wake up sequences that have DONE first are valid which are limited to sequences 0 to 7.

The default for DONE_EX preference is OFF.

MCCLK_FREQ

This setting is used after initialization for programming. Care must be taken to insure that the CCLK being output off chip is appropriate for the memory device being driven. This option is useful to speed up programming time from external memory devices such as SPI Flash. Also, the OSCA library element needs to be in the design for the internal oscillator to function after configuration is completed.

CONFIG_SECURE

This setting allows you to extract the configuration data stored in an FPGA in order to verify the configuration. Settings are OFF (default), ON (multiple readbacks), and ONCE (allow one readback only; after that, readback cannot be invoked again).

In a LatticeSC device, the Readback and Boundary scan functions share the same output pin TDO/RDDATA. If both functions are to be used, the RDDATA output from Readback can be routed to a different pin (or internally) by using the LatticeSC Macro Library element READBK.

Wake-Up

After configuration, the FPGA enters the wake-up phase. This phase is the transition between the configuration and operational states and begins when the number of CCLKs received after INITN goes high is equal to the value of the length count field in the configuration frame and when the end of configuration frame has been written. The system design issue in the wake-up phase is to ensure the user I/Os become active without inadvertently activating devices in the system or causing bus contention. A second system design concern is the timing of the release of global set/reset of the latches/FFs. There are configuration options that control the relative timing of three events: DONE going high, release of the set/reset of internal FFs and user I/Os becoming active.

A commonly used design technique is to release DONE one or more clock cycles before allowing the I/O to become active. This allows other configuration devices, such as PROMs, to be disconnected using the DONE signal so that there is no bus contention when the I/Os become active. In addition to controlling the FPGA during wakeup, other startup techniques that avoid contention include using isolation devices between the FPGA and other circuits in the system, reassigning I/O locations and maintaining I/Os as 3-stated outputs until contentions are resolved.

Each of these wakeup options can be selected during bitstream generation in ispLEVER Bitgen program (see the Bitstream Generation Options section of this document for details). INITN pins for all of the FPGAs are connected together. This is required to guarantee that power up and initialization will work correctly. In general, the DONE pins for all of the FPGAs are also connected together as shown to guarantee that all of the FPGAs enter the wake-up state simultaneously. This may not be required, depending upon the wake-up sequence desired.

When configuration is complete, the device should wake up in a predictable fashion. The following selections determine how the device will wake up. Two synchronous wake-up processes are available. One automatically wakes the device up when the internal Done Bit is set even if the DONE pin is held low externally. The other waits for the DONE pin to be driven high externally before starting the wake-up process. The DONE_EX preference determines if the synchronous wake up will be controlled by the external driving of the DONE pin or ignores the external driving of the DONE pin. If the LatticeSC device is the only device in the chain or the last device in a chain, the wake-up process should be initiated by the completion of the configuration. Once the configuration is complete, the internal Done Bit will be set and then the wake-up process will begin

Two synchronous wake up processes are available. One automatically wakes the device up regardless of whether the DONE pin is held externally. The other waits for the DONE pin to be driven high before starting the wake up process. The DONE_EX attribute determines if the synchronous wake up will be controlled by the external driving of the DONE pin high or ignores the external driving of the DONE pin. If no external source is provided, the DONE pin will need to be set to pulled high by the internal pull up.

In a synchronous wake-up process, the wake-up sequence is controlled with three bits for the internal GSR and GOE signals. The $\overline{\text{DONE}}$ pin is controlled by four. The release of the internal Done bit initiates the process by clearing a counter associated with wake-up. The counter should be cleared on the first rising edge of the clock selected to run the wake up process after the Done bit is set. The programmed bits specify on which cycle of the counter the individual signal wakes up.

When synchronous wake up is selected, the clock phase T0 to T3 will be synchronized to the clock selected by the CONFIG_MODE setting will use a slave clock from CCLK or the internal clock. When the user selects the order and the phase of wake up, then the software shall create the correct bit settings to achieve the desired wake up sequence. Note that in some sequences, more than one signal can be changed at a time.

The user can select the order the signals wake up. A list of 25 options is available to the user to select all logical possible wake-up sequences after programming is complete. When the user selects the sequence to follow, the software should set the proper bits to achieve the user desired wake-up sequence.

Table 12. Wake Up Options

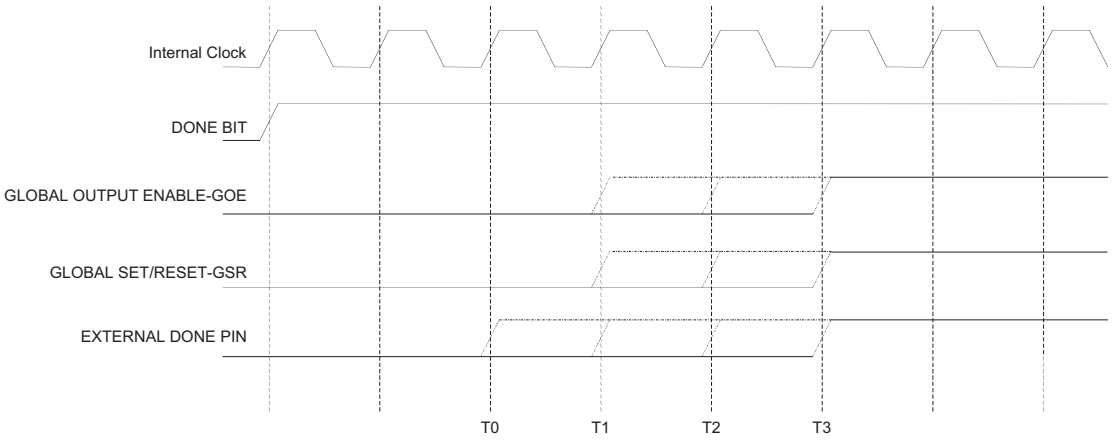
Sequence	Phase T0	Phase T1	Phase T2	Phase T3
1	EXTERNAL DONE	GOE, GSR		
2	EXTERNAL DONE		GOE, GSR	
3	EXTERNAL DONE			GOE, GSR
4	EXTERNAL DONE	GOE	GSR	
5	EXTERNAL DONE	GOE		GSR
6	EXTERNAL DONE	GOE		GSR
7[1]	EXTERNAL DONE	GOE	GSR	
8		EXTERNAL DONE	GOE, GSR	
9		EXTERNAL DONE		GOE, GSR
10		EXTERNAL DONE	GSR	GOE
11		EXTERNAL DONE	GOE	GSR
12			EXTERNAL DONE	GOE, GSR
13		GOE, GSR	EXTERNAL DONE	
14[1]		GOE	EXTERNAL DONE	GSR
15		GOE,	EXTERNAL DONE	GSR
16			EXTERNAL DONE	GOE, GSR
17		GSR	EXTERNAL DONE	GOE
18		GOE, GSR	EXTERNAL DONE	
19			GOE, GSR	EXTERNAL DONE
20		GOE, GSR		EXTERNAL DONE
21 (Default)		GOE	GSR	EXTERNAL DONE
22		GOE	GSR	EXTERNAL DONE
23			GOE, GSR	EXTERNAL DONE
24[1]		GSR	GOE	EXTERNAL DONE
25		GOE, GSR		EXTERNAL DONE

1. Recommended wake-up sequence for EBR designs.

Startup Clock

After configuration, the wake up sequence can be synchronized to the user's system clock. The configuration logic can be programmed to select a user clock as a clock source as opposed to the default CCLK. This allows an off-chip clock to drive the state machine controlling wake up sequence between the configuration and operation states. The input clock can be any single-ended input buffer depending on the VCCIO supply associated with the input pin. The user clock should be less than 200MHz. Once in the operation state, this clock will not affect operation.

Figure 21. Wake Up Timing Waveforms



Compression

The LatticeSC configuration circuitry includes a bitstream de-compressor to support compressed bitstream for most configuration modes. The compression of the bitstream is performed by ispVM software, while the decompression is performed by the dedicated on-chip circuitry. The user has an option after bitstream creation to compress the bitstream format. The compressed bitstream includes the Embedded algorithm data for the on-chip circuitry to utilize the smaller programming bitstream. The feature reduces of the original size of the bitstream up to 50%, however it does not shorten the time to configure the devices.

For uncompressed bitstreams, the output clocks from CCLK and RCLK are identical frequencies. However when the bitstream is compressed, RCLK is 1/8 the frequency of CCLK. When used for daisy chaining, the first or master device will decompress the bitstream and send a non-compressed bitstream to downstream slave devices.

Bitstream compression is available for use in all modes except slave serial and JTAG modes. If a compressed bitstream is sent to a device in either of these excluded modes, it will cause the programming to terminate. Also for daisy chained modes, the master receives compressed bitstreams and sends uncompressed serial data to the downstream devices.

Configuration Process and Flow

Prior to becoming operational, the FPGA goes through a sequence of states, including initialization, configuration, and wake-up.

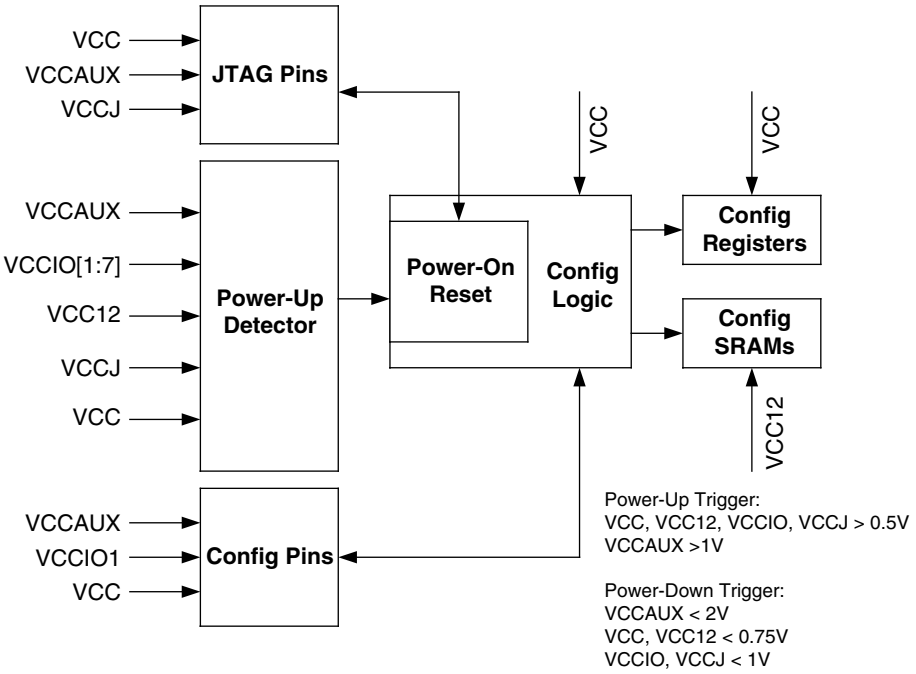
Power-up Sequence

The LatticeSC FPGA device has a power-up reset state machine that depends on various power supplies. A power-up reset counter will begin to count after all of the approximate conditions are met:

- VCC reaches 0.5V or above
- VCC12 reaches 0.5V or above
- VCCAUX reaches 1.0V or above
- VCCIO[1:7] reaches 0.5V or above

- VCCJ reaches 0.5V or above

Figure 22. Power Sequence State Machine



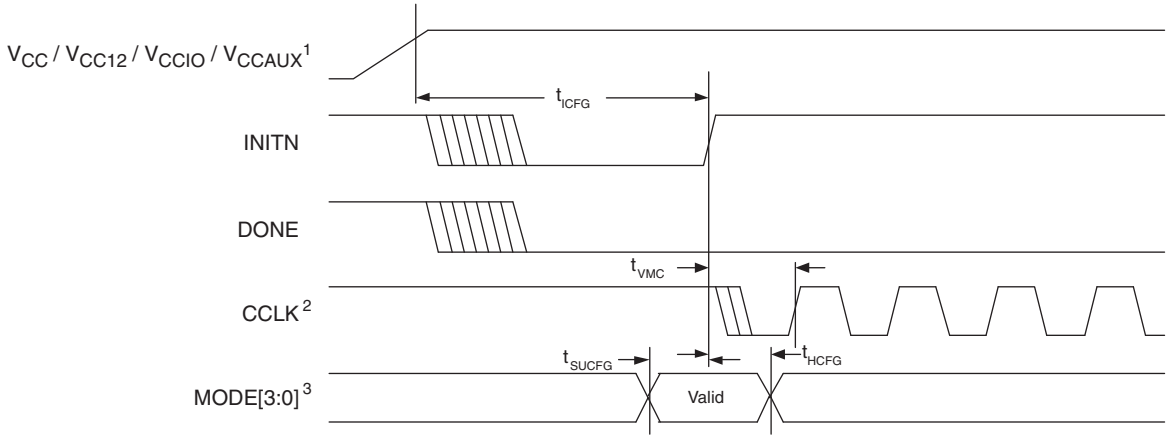
Additional power-up and power-down requirements can be found in the LatticeSC/M Family Data Sheet.

The counter running on about 1MHz clock will take about 75ms to wait for stabilizing of all supplies. This is independent of configuration mode. All power supplies are required to power up monotonically to their particular minimum operating voltage within 75ms. The memory retention voltage is a function of VCC12 and should be above 0.8V to retain the written contents of the configuration memory. To insure stability, the device also self-tests the configuration RAM by making certain it can be written to a 0 and to a 1 before the power-on is considered complete.

On power-up the device tri-states all the I/Os, inputs are pulled up and the INITN and EXTERNAL DONE pin are set to low. The device prepares for configuration by resetting the configuration circuitry. The device clears the configuration memory and gets ready to start configuration.

After the device finishes its power on reset, the device samples the Configuration Mode pins MODE[0:3] and recovers the relevant configuration pins according to the Configuration Mode Pin Settings (refer to Table 2). The device will then release the INITN pin if the PROGRAMN and RESETN pins are high. If a Master Mode is selected, the device starts driving the master clock out of the CCLK pin. The INITN pin can be driven low externally to delay device configuration. Once the INITN pin goes high, the device is ready for configuration to start.

Figure 23. Power-On Reset Timing



1. Time taken on all power supplies, whichever is the last to reach its V_{MIN}.
2. Device is in a Master Mode.
3. The MODE pins are normally static (hard wired).

Initialization

Upon power-up, the device power-up circuitry is triggered, as mentioned above. The VCCIO levels must be operational before any configuration signaling can begin. The I/Os are configured based on the configuration mode, as determined by the mode select inputs M[3:0]. The INITN and DONE outputs are low. At the end of initialization, the default configuration option is that the configuration RAM is written to a low state. This prevents shorts prior to configuration. As a configuration option, after the first configuration (i.e., at reconfiguration), the user can reconfigure without clearing the internal configuration RAM first. This is detailed later in the “Partial Reconfiguration” section.

The active-low, open-drain initialization signal INITN is released and must be pulled high by an external resistor when initialization is complete. To synchronize the configuration of multiple FPGAs, one or more INITN pins should be wire-ANDed. If one or more FPGAs or an external device holds INITN low, the FPGA remains in the initialization state. INITN can be used to signal that the FPGAs are not yet initialized. After INITN goes high for two internal clock cycles, the mode lines (M[3:0]) are sampled, and the FPGA enters the configuration state.

The high during configuration (HDC), low during configuration (LDCN), and DONE signals are active outputs in the FPGA’s initialization and configuration states. HDC, LDCN, and DONE can be used to provide control of external logic signals such as reset, bus enable, or PROM enable during configuration. During the configuration phase, HDC should be output high, LDCN should be output low, and DONE will be output low. For parallel master configuration modes, these signals provide PROM enable control and allow the data pins to be shared with user logic signals.

If configuration has begun, an assertion of RESETN or PROGRAMN initiates an abort, returning the FPGA to the initialization state. The PROGRAMN and RESETN pins must be pulled back high before the FPGA will enter the configuration state. During the start-up and operating states, only the assertion of PROGRAMN causes a reconfiguration. RESETN will not reinitiate a configuration.

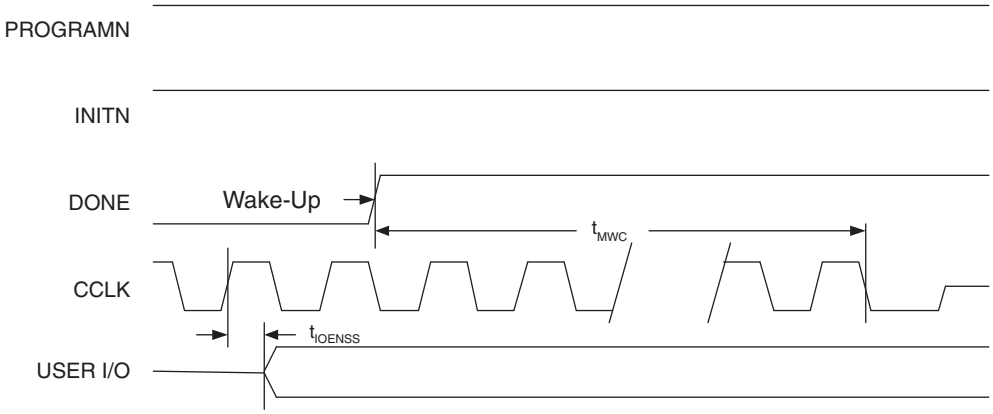
In the master configuration modes, the FPGA is the source of configuration clock (CCLK). In this mode, the initialization state is extended to ensure that, in daisy-chain operation, all daisy-chained slave devices are ready. Independent of differences in clock rates, master mode devices remain in the initialization state an additional six internal clock cycles after INITN goes high.

When configuration is initiated, a counter in the FPGA is set to 0 and begins to count configuration clock cycles applied to the FPGA. As each configuration data frame is supplied to the FPGA, it is internally assembled into data words. Each data word is loaded into the internal configuration memory. The configuration loading process is com-

plete when the internal length count equals the loaded length count in the length count field, and the required end of configuration frame is written.

During configuration, the user programmable I/O and PLC latches are set while FFs are held reset. The combinatorial logic begins to function as the FPGA is configured. Figure 21 shows the general waveform of the initialization, configuration, and wake-up states.

Figure 24. Wake-up Timing

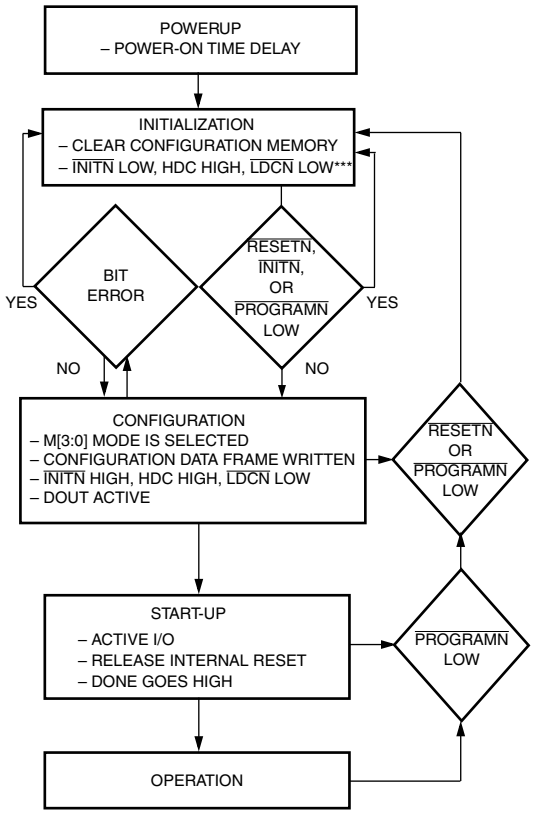


Loading the Configuration Memory

Once the PROGRAMN and INITN pins are high and the initialization phase is completed, configuration can begin. Depending on the configuration mode selected, data will be accepted on either the DI or D[0:7] pins on the rising edge of CCLK for all modes excluding MPI modes. If an error occurs at any time during transfer of the data, the INITN pin will be driven low by the LatticeSC device.

For handshaking configurations, the CS0N and CS1 pin can be driven high to pause configuration and stop the Master clock. Once the full data stream has been shifted in, the wake-up sequence or begin daisy chain flow of data to the next device.

Figure 25. Configuration Process Flow



*** Except for SPI configuration modes.

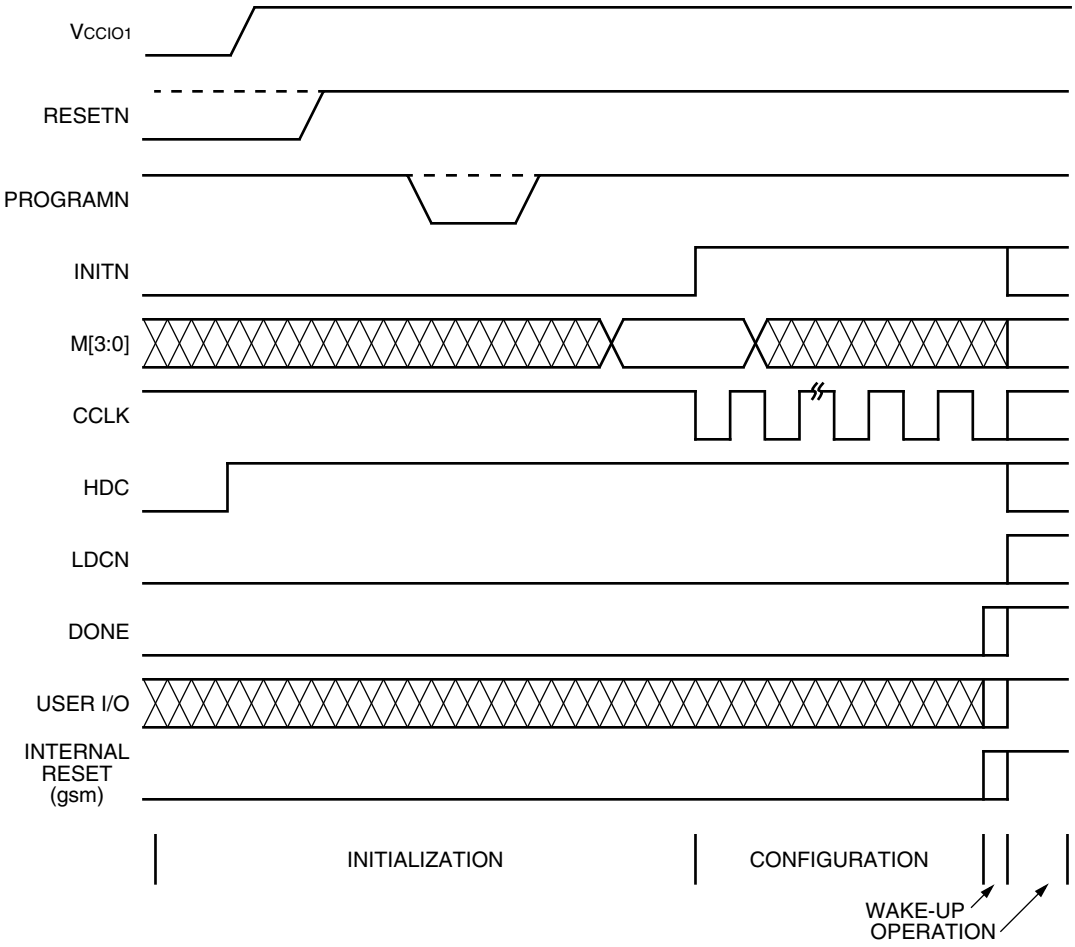
Clearing the Configuration Memory and Re-initialization

Several methods can clear the internal configuration memory of the LatticeSC device. The first is mentioned earlier when the device powers up. A second method is by toggling the PROGRAMN pin or the MPI control register. Also, JTAG can reinitialize the memory through an ISC command.

After a device is powered up, toggling the PROGRAMN pin will initiate a sequence to prepare the LatticeSC device for re-configuration. Upon driving the PROGRAMN pin low, the INITN and EXTERNAL DONE pins will drive low and the memory will start clearing. The I/O pins will become tri-stated and pulled up to VCCIO. The INITN pin will remain low until PROGRAMN is deasserted. The EXTERNAL DONE will remain low until a configuration is re-programmed.

Upon driving the PROGRAMN pin high, the MODE[0:3] are sampled to determine the configuration mode to implement as well as which configuration pins will be used for configuration. If a master mode is selected, the master clock will start to be driven out CCLK. The INITN pin will be released once the configuration memory is cleared and the PROGRAMN pin is driven high. Holding the INITN pin low will delay configuration. Configuration will begin as soon as the INITN pin is released and pulled high.

Figure 26. Configuration Timing Diagram



Configuration Clock/Oscillator Use

There are eight configuration speeds available for the master and asynchronous peripheral modes, controlled by a nominal 133MHz internal oscillator and selected via a bitstream. The internal clock frequency can be one of eight choices, 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 of the internal 133MHz (nominal) oscillator. Initially after power-up or triggering of a re-initialization, the clock frequency uses the slowest 1/128 (about 1.0MHz), while initializing (the period of time until the INITN pin goes high), the clock uses 1/8 (about 16MHz).

When the user has selected a particular master or peripheral configuration via the Mode [3:0] pins, the CCLK and RCLK pin are selected as output clocks with the a determined frequency set by the user. The CCLK/RCLK speed for master and peripheral modes uses three reserved bits written very early in the bitstream to select the clock speed. After the oscillator speed is read from the bitstream, the clock will switch to the speed determined by user selection via the configuration bitstream. The selection of the clock speed is done at bitstream creation using the ispLEVER software tools

Bitstream Error Checking

There are three different types of bitstream error checking performed in the LatticeSC FPGAs: ID frame, frame alignment, and CRC checking. The ID data frame is sent to a dedicated location in the FPGA. This ID frame contains a unique code for the device for which it was generated. This device code is compared to the internal code of the FPGA. Any differences are flagged as an ID error.

Each data and address frame in the FPGA begins with a frame start pair of bits and ends with eight stop bits set to 1. If any of the previous stop bits were a 0 when a frame start pair is encountered, it is flagged as a frame alignment error.

Error checking is also checked on the FPGA for each frame by means of a checksum byte. If an error is found on evaluation of the checksum byte, then a checksum error is flagged. The checksum is the XOR of all the data bytes, from the start of frame up to and including the bytes before the checksum. It applies to the ID, address, and data frames.

When any of the three possible errors occur, the FPGA is forced into an idle state, forcing INITN low. The FPGA will remain in this state until either the RESETN or PROGRAMN pins are asserted. If using either of the MPI modes to configure the FPGA, the specific type of bitstream error is written to one of the MPI registers by the FPGA configuration logic. In asynchronous peripheral mode, the error flags can be read from the data bus. The PROGRAMN bit of the MPI control register can also be used to reset out of the error condition and restart configuration.

Readback

Readback is the process of reading the configuration data from the SRAM memory cells of a programmed FPGA. Performing a readback extracts the contents known as configuration bits to verify that a design was downloaded correctly or has not changed during operation.

In the LatticeSC, readback is accessed through IEEE 1532 compliant JTAG, the system bus interface, or an internal readback interface. All can be used to verify that the current configuration data is correct and the readback operation can be done while the FPGA is in normal system operation. The readback operation cannot be daisy-chained and is only permitted after configuration is completed indicated by the open-collector DONE signal release (high).

Design security is a concern for many designers. Readback is only permitted after the user disables the security feature via the SYSCONFIG preference CONFIG_SECURE. This security register is over-written through the bitstream loaded into the FPGA. This is intended primarily to give the user control over the security of the FPGA's configuration program. The user can prohibit readback, allow a single readback, or allow unrestricted readback. By default, CONFIG_SECURE=ON, not allowing read back. The security attribute can be set via the ispLEVER Design Planner. The Design Planner allows the user to select one of the options in Table 13 from the Global Preference tab.

Table 13. Bitstream Security Attributes

CONFIG_SECURE	
Attribute	**Description**
OFF	Unrestricted read back/on-demand
ON	Never allow read back
ONCE	Only read back one time

JTAG Readback

The JTAG Test Access Port can read back the contents of the configuration memory. This is accomplished using the IEEE 1532 in-system configuration commands. The ISC READ op code 00001010 is issued to the TAP controller. The controller responds by reading back data from the configuration memory and output to TDO.

The readback operation is started at frame x00 after the JTAG TAP controller starts the operation. One bit of data is shifted out on TDO on the falling edge of TCK. The user can be certain of the start of the readback frame by monitoring the data for the 01 frame start bit pair. ispVM software completely supports this feature and can be used to read and save the memory contents in a bit(10) format or HEX format. This data format will be discussed later.

To use this approach, the user needs to disable the bitstream security as mentioned previously. There are no other requirements to use this feature other than the use of the JTAG TAP controller.

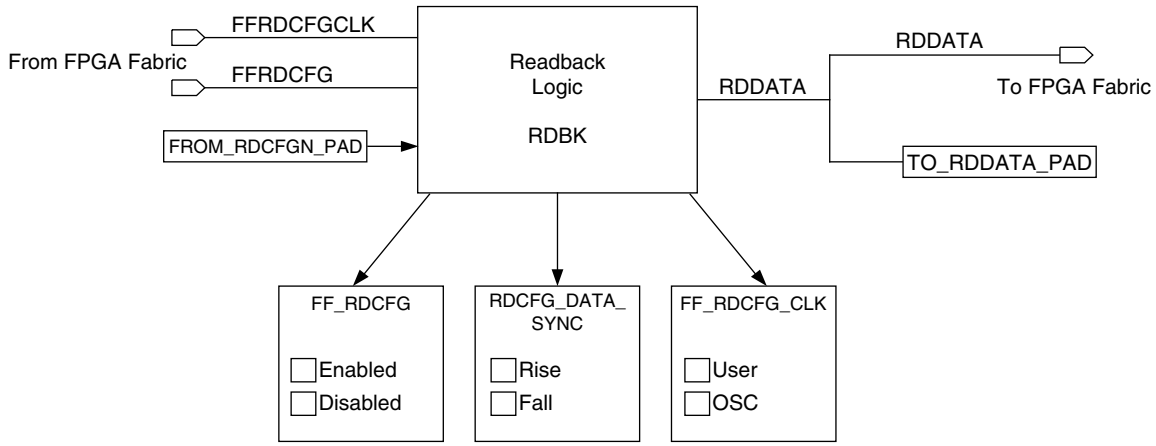
Serial Readback

Readback functionality is also available within the user's design. This function is useful for monitoring the integrity of the configuration stored in the FPGA SRAM memory. An example use of this is for soft-error detection (SED) applications. Soft errors occur when high-energy charged particles alter the stored charge in a memory cell in an electronic circuit. The phenomenon first became an issue in DRAM, requiring error detection and correction for large memory systems in high-reliability applications. As device geometries have continued to shrink, the probability of soft errors in SRAM has become significant for some systems. Although SED may not be an issue for most systems, designers are using a variety of approaches to minimize the effects of soft errors on system behavior when needed.

SRAM-based FPGAs store logic configuration data in SRAM cells. As the number and density of SRAM cells in an FPGA increase, the probability that a soft error will alter the programmed logical behavior of the system increases. A number of approaches have been taken to address this issue, but most involve Intellectual Property (IP) cores that the user instantiates into the logic of their design, using valuable resources and possibly affecting design performance. The LatticeSC includes a full-feature block to be incorporated into designs requiring readback functions.

This LatticeSC serial readback module is instantiated by the user. The self-contained module for readback is added to the user's design and can be used in conjunction with SED IP cores or any user-defined circuitry. The instance includes ports that are connected within the FPGA fabric. The module includes ports required to be included in the users design.

Figure 27. Serial Readback Module



The library element connects to the user design via a three-port interface. The interface uses these connections to control and manage the readback of the bitstream data. The data read is available to be routed to the FPGA fabric. The data is clocked by a user-supplied clock or the on-chip oscillator. The maximum readback is 150MHz.

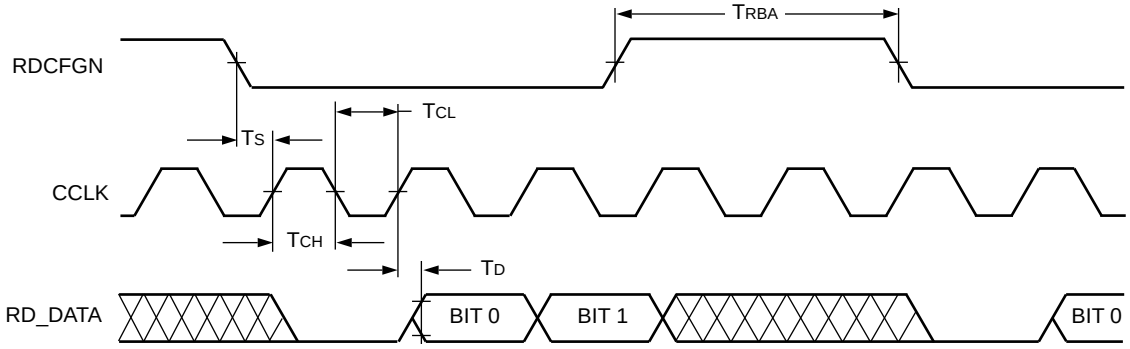
Table 14. RDBK Library Element Ports

Port	Direction	Description
FFRDCFGCLK	Input	FPGA fabric clock used to read back configuration.
FFRDCFG	Input	Enable to start readback.
RDCFG_IN	Input	Dedicated enable to start readback (alternative to FFRDCFG port). Low to high transition starts readback.
RDDATA	Output	Data output of configuration memory.

The readback logic instance cannot be simulated. It has synthesis attributes that can be optionally selected in the design. These attributes shown in Table 15 control user options provided in the hardware logic. Serial readback only permits incremental reading of the configuration data and always begins at 00h whenever readback is initiated.

The FF_RDCFG= Disabled option only permits readback through a fixed interface. This interface utilizes the RDCFGN device pin to start readback by toggling the 1 to 0. This initiates the readback and is clocked by driving a clock into the CCLK pin and data is shifted out on the RDDATA (TDO) output pin. The waveform of this interface is depicted in Figure 28.

Figure 28. Readback Timing Waveform



The RDCFGN input pin is also used to control the global tri-state (TSALL) function. Before and during configuration, the TSALL signal is always driven by the RDCFGN input and read back is disabled. After configuration, the selection as to whether this input drives the readback or global tri-state function is determined by a set of bitstream options. If used as the RDCFGN input for readback, the internal TSALL input can connect internally to be driven by any input pin.

When FF_RDCFG = Enabled, user-defined connections to the FPGA fabric are available. The user can select the active control edge for FFRDCFG port to start a readback by setting the RDCFG_DATA_SYNC attribute. The data is clocked out of the RDDATA port to the fabric synchronous to the user-connected clock port FFRDCFGCLK to the FPGA using a user-supplied clock or the oscillator.

Table 15. RDBK Attributes

Attribute	Option	Description
FF_RDCFG	Enabled/Disabled	Port connection for starting readback Enable – From FPGA fabric Disable – From dedicated RDCFGN pin
RDCFG_DATA_SYNC	Rise/Fall	When the FF_RDCFG is used to control start of readback, this options sets the active edge.
FF_RDCFG_CLK	User/Osc	Determines source of clock to readback. User can be routed from any source in the FPGA fabric. OSC is the on-chip oscillator that can be controlled by the OSC library element.

Verilog instantiation example:

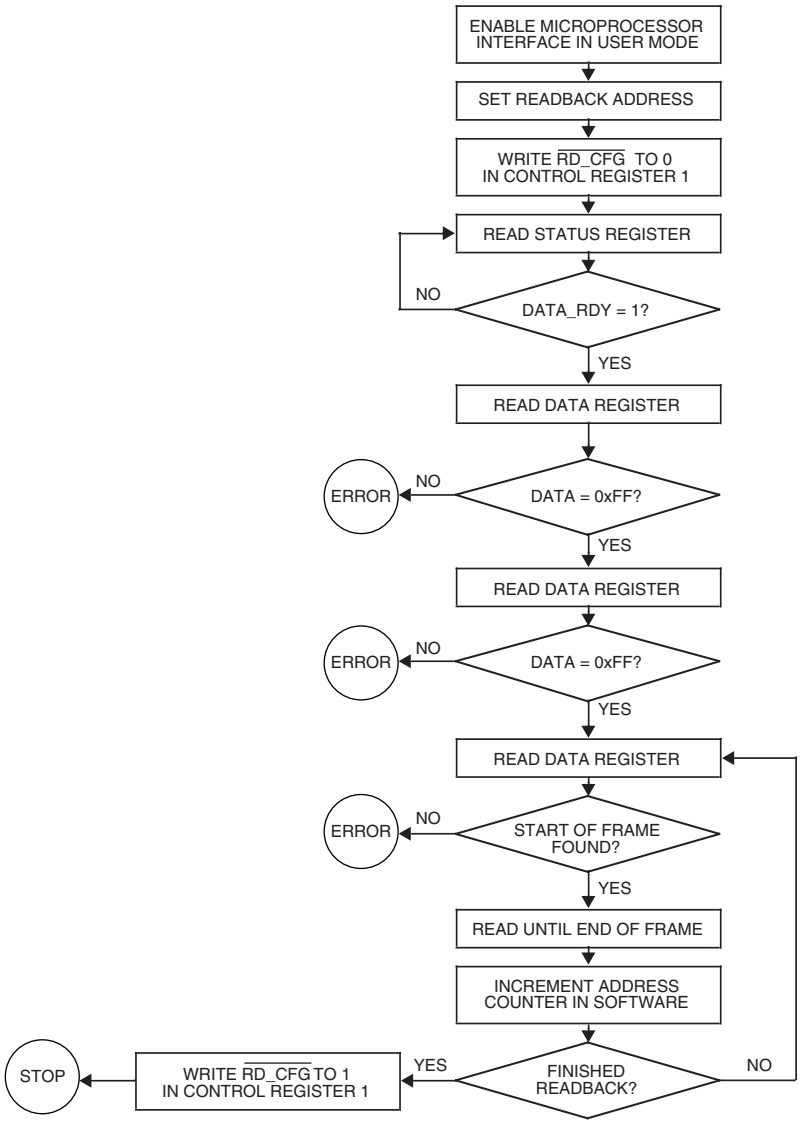
```
//RDBK RDBK_INST(
   //Inputs
   RDCFGN          (1'b1),
   FFRDCFG         (fabric_rdcfgn),
   FFRDCFGCLK      (fabric_clock),

   //Outputs
   RDDATA          (fabric_rdddata)
   //)
   //* synthesis FF_RDCFG="ENABLED" */
   //* synthesis RDCFG_DATA_SYNC="RISE" */
   //* synthesis FF_RDCFG_CLK="OSC" */;
```

Readback via System Bus

Readback can be performed via the system bus from the MPI or user master interface. Readback using the MPI uses the flow diagram shown in Figure 29. Readback can be initiated at an address other than frame 0 via the system bus control registers and may repeatedly read that frame without reading other frames. Serial readback and MPI readback cannot be used simultaneously, but MPI accesses that do not include readback can occur during serial readback. MPI readback is discussed in greater detail in TN1085, LatticeSC MPI/System Bus.

Figure 29. Readback Flow



Readback Data Format

The readback frame has an identical format to that of the configuration data frame, which is discussed in the Configuration Data Format section. If LUT memory is not used as RAM, the readback data (not just the format) will be identical to the configuration data for the same frame. This eases a bitwise comparison between the configuration and readback data. The configuration header, including the length count field, is not part of the readback frame.

The entire configuration memory contents for making a bit-per-bit comparison of the readback data to the bitstream file should not be used directly for comparison. There are data bits included inside a readback that cannot be verified against the original bitstream. The RAM bits stored in configuration memory cells that hold the contents of all PFUs configured as RAM and EBRs contain dynamically changing values per the user design. The readback con-

tents of these cells are automatically transposed as ZERO during read back. All PFUs programmed in RAM mode will read back zero regardless of its programmed value. EBR RAMs cannot be read back in bitstream format even if it is preloaded.

The ispLEVER development system optionally provides a readback data stream known as a mask file to compare to readback data from the FPGA. As previously noted, if any of the LUTs are used, RAM will always be read back as zero. This software process will generate a mask file to overlay against the readback capture which accounts for the configuration space of these unverifiable bits. The mask file is used to determine which bits in the bitstream should be compared to readback data for verification purposes.

The contents of PCS SERDES “performance bits” loaded into the device via the bitstream can be read back. This does not include the data loaded with autoconfig of the PCS and other MACOs. The autoconfig is a register preload to the system bus and not visible in the bitstream so it cannot be read back with the bitstream. It also is not input into the readback mask file.

Table 16 shows the complete frame format for the FPGA if explicit addressing is specified in the bitstream generation software. If incremental addressing will be used then the frame associated with addressing can be eliminated.

Table 16. Configuration Frame Format

FPGA Header	11110010	A mandatory header for RAM bitstream portion
	11111111	8 stop bits (high) to separate frames
FPGA Address Frame	00	Address frame header
	14-bit address	14-bit address
	Checksum	8-bit checksum
	11111111	8 stop bits (high) to separate frames
FPGA Data Frame	01	Data frame header
	Alignment bits	String of 0-bits added to reach a byte boundary
	Data bits	Number of data bits for a single frame1
	Checksum	8-bit checksum
	11111111	8 stop bits (high) to separate frames
Postamble for FPGA	00 or 10	Postamble header 00 = finish, 10 = more bits coming
	111111 11111111	Dummy address
	11111111 11111111	16 stop bits (high)

Table 17 shows the configuration memory sizes if no EBRs are initialized and the bitstream is not compressed (see Table 18 for additional information). It also does not include any auto configuration bits used for SERDES and PCS.

Configuration Memory Sizes

Table 17. Configuration Memory Sizes

FPGA Device Header	Preamble	8
	Length Count	24 or 32
	Trailing Header	8
	ID frame	16
	Reserved	40
	Compression bit	1
	Osc speed	3
	Part ID	20
	Checksum	8
	stop bits	8
	FPGA Header	8
	Stop bits	8
FPGA Address Frame	LFSC15	1043
	LFSC25	1316
	LFSC40	1652
	LFSC80	2156
	LFSC115	2639
	Address Header	2
	14-bit address	14
	Checksum	8
	Address frame stop	8
	Data Header	2
	Alignment bits	2 or 6
FPGA Data Frame	LFSC15	4216
	LFSC25	5816
	LFSC40	7100
	LFSC80	10300
	LFSC115	11580
	Checksum	8
	Stop bits	8
FPGA Postamble	Header	2
	Dummy addr	14
	Stop bits	16
Total PROM Size for FPGA Explicit Bitstream	LFSC15	4,532,120
	LFSC25	7,835,824
	LFSC40	11,822,008
	LFSC80	22,523,272
	LFSC115	35,707,708

Embedded Block RAM Memory Configuration

The Embedded Block RAM (EBR) Memory can be written with non-zero initial values via configuration. Memory initialization, known as INITVAL, can be attributes in the netlist from IPexpress or user generated through HDL. Simulation uses a generic parameter INITVAL, while the synthesis netlist passes an INITVAL attribute to the synthesis

tool to initialize the memory components. The INITVAL values are processed by the software flow and are implemented in the bitstream via ispLEVER.

The preloading of EBRs by configuration is accomplished using a defined format in the bitstream. This preloading data is generated by using software controlled EBR addressing scheme for memory initialization.

A unique INIT_ID is created when the mapping software processes the EBR modules generated from IPexpress or user HDL. Logical memory initialization values defined by the user are mapped to the physical definitions of the INIT_ID by ispLEVER. These EBR modules can be used individually or used in multiple instances that would have identical INITVALS thus writing the same data to multiple EBRs. The preloading will occur as part of the initial programming of the device. The format in Table 18 details the content of the bitstream for EBR memory initialization. The bitstream size is directly affected by the use of INITVAL preloading for each unique INIT_ID. In cases where many of the same EBR instances have been compressed by the mapping algorithm, this will limit the size impact of the bitstream.

Table 18. Embedded Block RAM Bitstream Initialization Contents

Name	Contents	Description
RAM Header	1111 0001	Mandatory EBR RAM Header
	1111 1111	8 stop bits
RAM Address Frame	00	Address Frame Header
	10-bit Address	10-bit Address
	Checksum	8-bit checksum
	1111 1111	Stop bits to separate frames
RAM Data Frame	01	Frame header
	000000	0-bits added to reach byte boundary
	1024 x 18 data bits	EMR data bits
	Checksum	8-bit checksum
	1111 1111	Stop bits
Postamble for EBR	00 or 10	Postamble (00 = finish, 10 more bits coming)
	1111111 11111111	Dummy address
	1111 1111 1111 1111	16 stop bits

Embedded Application Specific Block Configuration

The configuration of an embedded application specific block (ASB) is done in a similar fashion as the EMB initial values mentioned in the previous section. When a design uses any of the available LatticeSC embedded cores such as the PCS or MACO blocks, the ispLEVER design tool will provision the embedded blocks appropriately. This process, which is transparent to the user, is known as auto configuration.

Auto configuration is accomplished via the IPexpress tool and the user generated HDL. The ispLEVER software will interpret the users requirements of the embedded ASBs and provide the initial auto configuration data to the bitstream. The only noticeable difference between EBR and ASB initialization is ASB auto configuration does not do instance compression. The bitstream format is shown in Table 19. This auto configuration data will increase the final size of the default FPGA bitstream. Further details relating to this topic can be found in the *LatticeSC Family Handbook.*

Table 19. Embedded ASB Bitstream Auto Configuration Contents

Name	Contents	Description
ASB Header	11110100	A mandatory header for ASB bitstream portion
	11111111	8 stop bits (high) to separate frames
ASB Address Frame	00	Address frame header. Same as generic
	Transfer size (2 bit)	00=byte, 01=half word, 10=word
	4-bit transfer count	Configuration data frame length, the maximum is 16 words.
	16-bit address	16-bit base address of ASB registers
	Checksum	8-bit checksum
	11111111	Eight stop bits (high) to separate frames
ASB Data Frame	01	Data frame header. Same as generic FPGA
	111111	Dummy control bits
	000000	Six 0-bits added to reach a byte boundary
	Data bits	The number of bits depends on the 6 bits in address frame
ASB Postamble	00 or 10	Postamble header. 00 = finish, 10 = more bits coming
	111111	Dummy control bits
	11111111 11111111	Dummy address
	11111111 11111111	16 stop bits (high)

References

- LatticeSC/M Family Data Sheet
- TN1100, SPI Serial Flash Programming Using ispJTAG in LatticeSC Devices
- TN1085, LatticeSC MPI/System Bus
- ispLEVER software documentation
- ispVM System software documentation

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
—	—	Previous Lattice releases.
December 2006	01.2	Updated to correct errors and formatting in original document.
		Added Configuration Frame Format information.
August 2007	01.3	Updated SPI Programming Interface Diagram.
November 2007	01.4	Figure 15, changed 2.5V to 3.3V for the left side of the SPI Slash memory Device.
December 2007	01.5	Deleted reference to CONFIG_IOVOLTAGE.
February 2008	01.6	Updated Master Byte Timing Configuration Waveform diagram.
March 2008	01.7	Updated sysCONFIG Pin Usage table with check in RDCFGN/JTAG to indicate use of RDCFGN in JTAG Configuration mode.
June 2008	01.8	References to TS_ALL changed to TSALL.
June 2008	01.9	Added footnote to Wake Up Options table for EBR designs.
October 2008	02.0	Corrected routing to Bus Cntrl in Slave Parallel Configuration Mode Diagram.

Appendix A. sysCONFIG Pins

This appendix provides a crossover for the pin utilization of sysCONFIG signals. It is intended to provide reference for all configuration modes.

Table 20. sysCONFIG Crossover Descriptions

Signal Name	I/O	Description
RESETN	I	Active low input prior to configuration done, forces the restart of configuration. An internal active pull up is defaulted prior to configuration. This can be deactivated after configuration. Optionally user-programmed to connect to GSR after configuration.
CCLK	I/O	In the master and asynchronous peripheral modes, CCLK is an output which strobes configuration data into daisy-chained devices. In the slave modes, or read-back after configuration, CCLK is input synchronous with the data on DIN/DOUT or D [7:0]. CCLK is output for daisy-chaining operation when the lead FPGA is in master, peripheral, or MPI modes.
DONE	I/O	As an input, a low level on DONE delays FPGA wake-up after configuration if the appropriate wake-up options are set. As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. Optional user programmable 100K internal pull up
PROGRAMN	I	PROGRAMN is an active-low input that forces the restart of configuration. This pin always has an active pull-up.
RDCFGN	I	This pin must be held high during device initialization until the INITN pin goes high. This pin always has an active pull-up. During configuration, RDCFGN is an active-low input that activates the TSALL function and 3-states all of the I/O. After configuration, it can be selected (via a bitstream option) to activate the TSALL function as described above, or, if read-back is enabled via a bitstream option, a high-to-low transition on it will initiate read-back of the configuration data starting with frame address 0.
M[3:0]	I	During power-up and initialization, M[3:0] are used to select the configuration mode with their values latched on the rising edge of INITN. After configuration these pins are available as user defined inputs.
INITN	I/O	INITN is a dedicated bidirectional signal before and during configuration. A pull-up is always enabled, but an external 10K ohm pull-up resistor is recommended. As an active-low open-drain output, INITN is held low during power stabilization and internal clearing of memory. As an active-low input, INITN holds the FPGA in the wait-state before the start of configuration.
CFGIRQN/MPI_IRQ_N	O	During JTAG, slave, master, and asynch. peripheral configuration, assertion on this CFGIRQN (active low) indicates error(s) for block RAM or MACO initialization. It is also the interrupt pin for MPI to external microprocessor. A programmable pull-up resistor is available on this pin.
Dedicated JTAG Pins: Power supplied via dedicated VCCJ (1.8V, 2.5V or 3.3V) supply.		
TDO/RDDATA	O	JTAG data output. If used for serial read-back, RDDATA provides serial configuration data out which is clocked out using CCLK.
TCK	I	JTAG clock input.
TDI	I	JTAG data input.
TMS	I	JTAG mode selection input.
Dual-Purpose Pins: After configuration, these pins are user-programmable I/O pins.		
DOUT	O	During configuration, DOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices. The data-stream from this output will propagate preamble bits of the bitstream to daisy-chained devices. Data out on DOUT changes on the rising edge of CCLK.
QOUT/CEON	O	During daisy-chaining configuration, QOUT is the serial data output that can drive the D0/DIN of daisy-chained slave devices that do NOT propagate preamble bits. Data out on QOUT changes on the rising edge of CCLK. During parallel-chaining configuration, active low CEON enables the cascaded slave device to receive bitstream data.

Table 20. sysCONFIG Crossover Descriptions (Continued)

Signal Name	I/O	Description
BUSYN/RCLK/SCK	O	During configuration in peripheral mode, high on BUSYN indicates another byte can be written to the FPGA. If a read operation is done when the device is selected, the same status is also available on D7 in asynchronous peripheral mode. For the master modes, RCLK is a read clock output signal to an external memory. For slave parallel mode, active low BUSYN inhibits the external host from sending new data. This output is used by slave parallel and master serial modes only for decompression. For SPI modes, SCK generated by the device and connected to the CLK input of the Flash memory.
HDC/SI	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete. For SPI modes, this pin is used to download the read command and initial read address into the Flash memory device on the falling edge of SCK. This pin will be connected to SI of the memory. If the SPI mode is used the 8-bit instruction code 0x03 will be downloaded followed by a 24-bit starting address of 0x000000 or a non-zero stat address for partial reconfiguration. If the SPIX mode has been selected the 8-bit instruction captured on D[7:0] at power-up will be shifted in followed by a 32-bit starting address of 0x000000.
LDCN/ SCS	O	Low During Configuration is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete. For SPI modes, this is an active low chip select for Flash memories. It will go active after INITN goes high but before SCK begins. During power up LDCN will be low. Once INITN goes high, LDCN will go high for 100ns - 200ns after which time will go back low and configuration can begin. During the 100ns -200ns period the read instruction will be latched for SPIX mode.
CS0N CS1 SIZE[2:0]-SPI Flash	I	CS0N and CS1 are used in the asynchronous peripheral, slave parallel, and MPI modes. The FPGA is selected when CS0N is low and CS1 is high. During configuration, a pull-up is enabled on both except with MPI DMA access control (see Lattice technical note number TN1085, LatticeSC MPI/System Bus). Used to set the number of Flash memory devices used to simultaneously download data into the device. {WRN, CS1, CS0N} = 111 => 1 device, = 110 => 2 devices, 101 => 3 devices...000 => 8 devices.
RDN/MPI_STRB_N	I	RDN is used in the asynchronous peripheral configuration mode. A low on RDN changes D[7:3] into status outputs. WRN and RDN should not be used simultaneously. If they are, the write strobe overrides. MPI_STRB_N is MPI TS (transfer start) signal.
WRN/MPI_RW_N	I	WRN is used in the asynchronous peripheral and slave parallel configuration mode. When the FPGA is selected, a low on the write strobe, WRN, loads the data on D[7:0] inputs into an internal data buffer. MPI_RW_N is MPI read (high)/write (low) signal. Used to set the number of Flash memory devices used to simultaneously download data into the device. {WRN, CS1, CS0N} = 111 => 1 device, = 110 => 2 devices, 101 => 3 devices...000 => 8 devices.

Table 20. sysCONFIG Crossover Descriptions (Continued)

Signal Name	I/O	Description
D[7:0]	I/O	For parallel configuration modes, D[7:0] receives configuration data, and each pin is pull-up enabled. For slave serial mode, D0 is the data input. D[7:3] output internal status for peripheral mode when RDN is low. D[7:0] is also the first byte of MPI data pins. For SPI modes, Data input from the Flash memory device(s). These pins will be connected to the SO pins of the Flash memories. If only a single Flash memory is used the input used will be D[0]. Additional inputs, starting at D[1], are used as additional Flash memories are used. If the SPIX mode is used these pins provide the READ instruction code upon power-up.
A[21:0] MPI_BURST_N MPI_BDIP_N MPI_TSZ MPI_ADDR	I/O	In master parallel mode, A[21:0] is an output and will address the configuration EPROMs up to 4MB space. For MPI configuration mode, A[17:0] will be the MPI address MPI_ADDR, while the A[19:18] will be transfer size and A[21:20] will be burst mode MPI_BURST_N and burst in process MPI_BDIP_N.
MPI_TA, MPI_TEA MPI_RETRY	O	For MPI configuration modes, these active low signals indicate transfer success (MPI_TA), error (MPI_TEA) or retry (MPI_RETRY).
MPI_CLK	I	For MPI configuration mode, it is input for MPI clock.
D[31:8] DP[3:0]	I/O	MPI data and parity data buses.

Note: Signals listed as Signal A / Signal B defines the same physical pin that is used for different functions based on configuration mode.