SPI Flash Programming and Hardware Interfacing Using ispVM System

March 2005 Technical Note TN1081

Introduction

SRAM-based FPGA devices are volatile and require reconfiguration after power cycles. This requires external configuration data to be held in a non-volatile device. Although serial boot PROM devices are commonly used for this purpose, they are often found in a proprietary format from one single vendor. This can lead to high cost and problems with availability.

Lattice has addressed this significant cost issue with its new FPGA product families. The latest generation of FPGAs from Lattice can be configured directly from low-cost, industry-standard third-party SPI (Serial Peripheral Interface) Flash memory devices. These memory devices are available from many sources, allowing them to remain cost-effective.

FPGAs that do not directly support SPI Flash configuration can still benefit from the cost savings of SPI Flash memory with the addition of some simple external logic. This technical note describes how a serial PROM can be emulated using a Lattice PLD and SPI Flash memory device to configure an FPGA in a standard master-serial mode. The accompanying design files can be downloaded from the Lattice web site at www.latticesemi.com.

SPI Flash Programming using ispVM® System

ispVM System software version 15.0 and later provides support for the SPI Flash FPGA Loader. The device selection dialog box contains a device named 'FPGA Loader', as shown in Figure 1. This device represents the FPGA Loader PLD in a standard JTAG chain, along with the SPI Flash device.

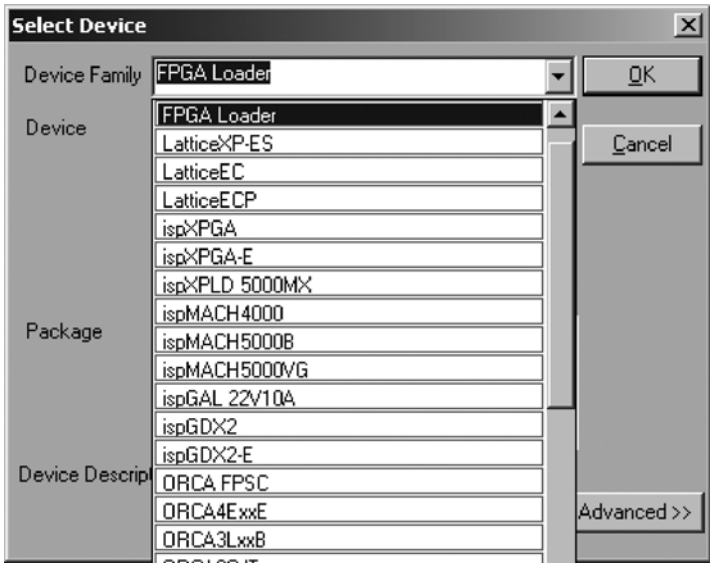
The following are required to accommodate SPI programming from ispVM System:

1. The PLD must be the first device in the JTAG chain.
2. There can be only one PLD/SPI pair per JTAG chain.

ispVM System Setup

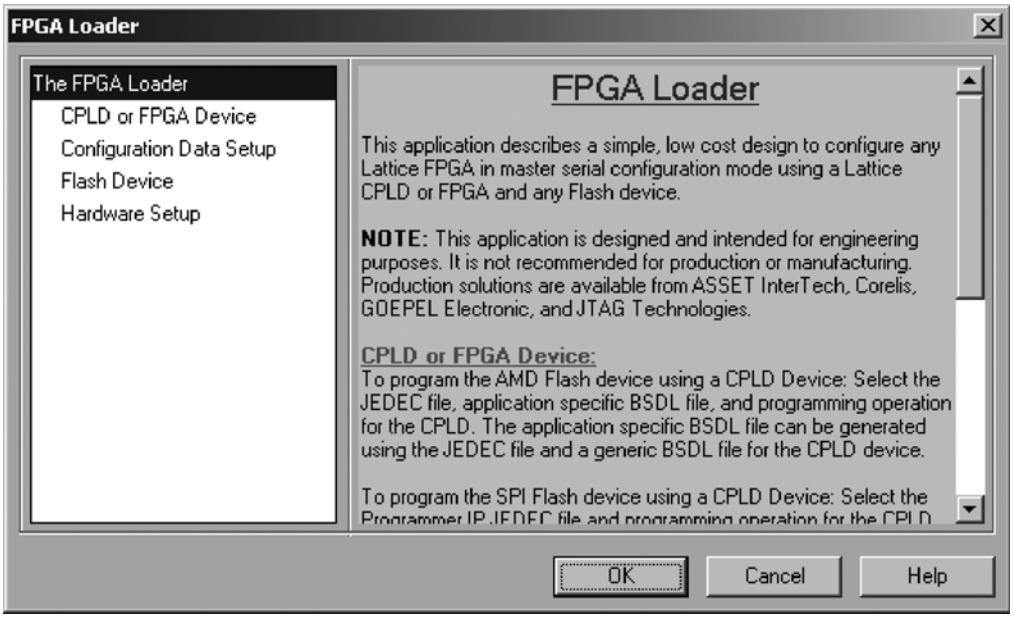
The figures in this section describe the ispVM System setup for programming the SPI Flash device.

Figure 1. Device Selection Dialog



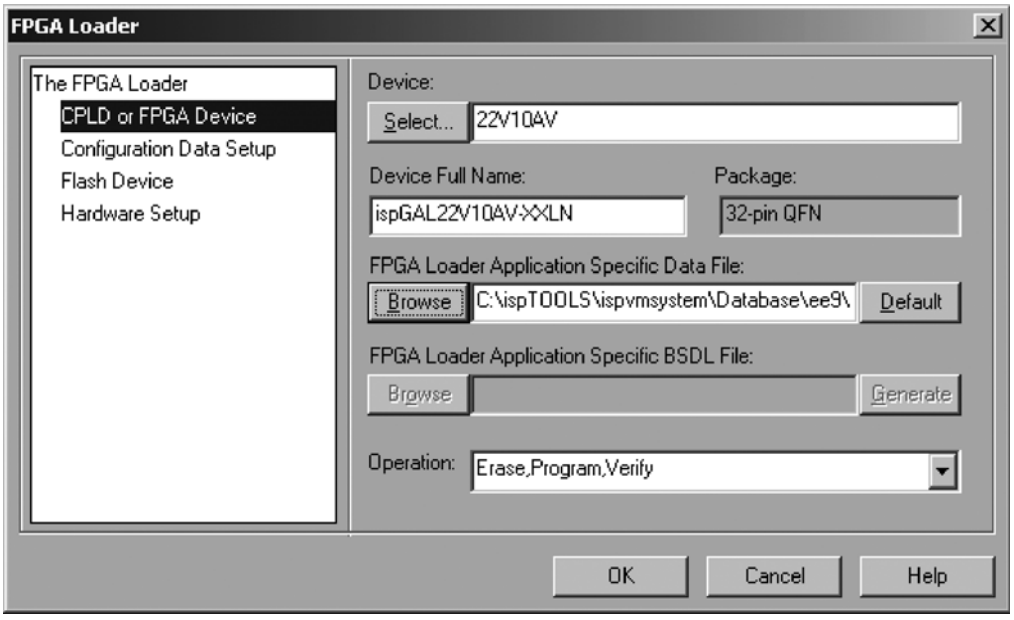
Once the FPGA Loader is selected, the configuration dialog appears, as seen in Figure 2.

Figure 2. FPGA Loader Dialog



The FPGA Loader dialog is comprised of four pages of options to specify the PLD, configuration data, Flash device and hardware setup.

Figure 3. CPLD or FPGA Device Page



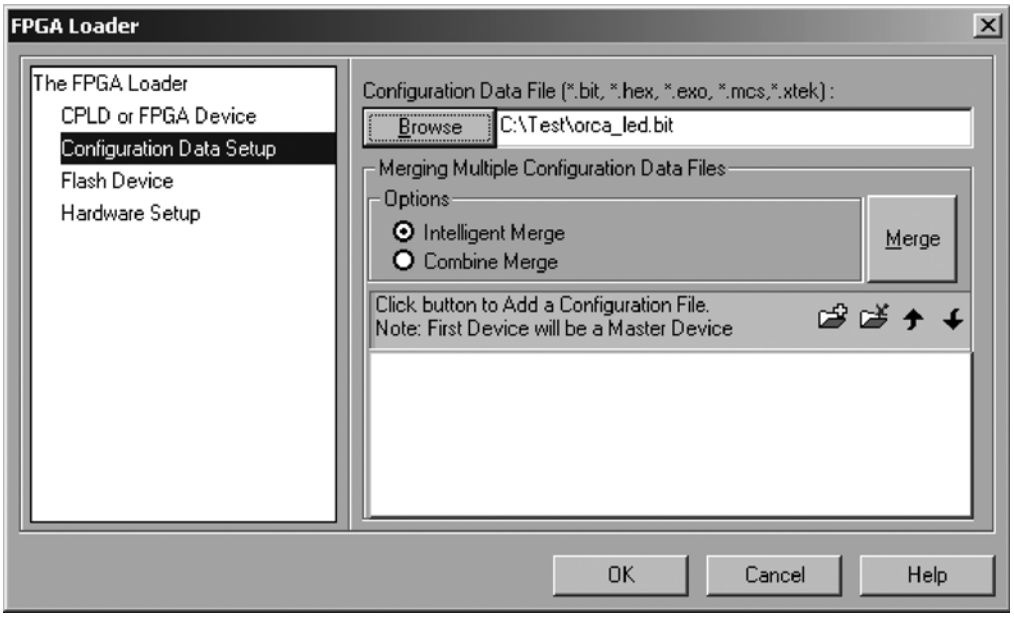
Device: Select the specific PLD that implements the FPGA Loader functionality.

FPGA Loader Application Specific Data File: By default, the .jed file shipped with the ispVM System software will be loaded. Clicking on the Default button will also select the default .jed file. See Table 2 for the pinouts for the default .jed files. Click the Browse button to select the .jed file produced by the ispLEVER® design software after fitting the SPI Loader design files.

FPGA Loader Application Specific BSDL File: This field should be left blank.

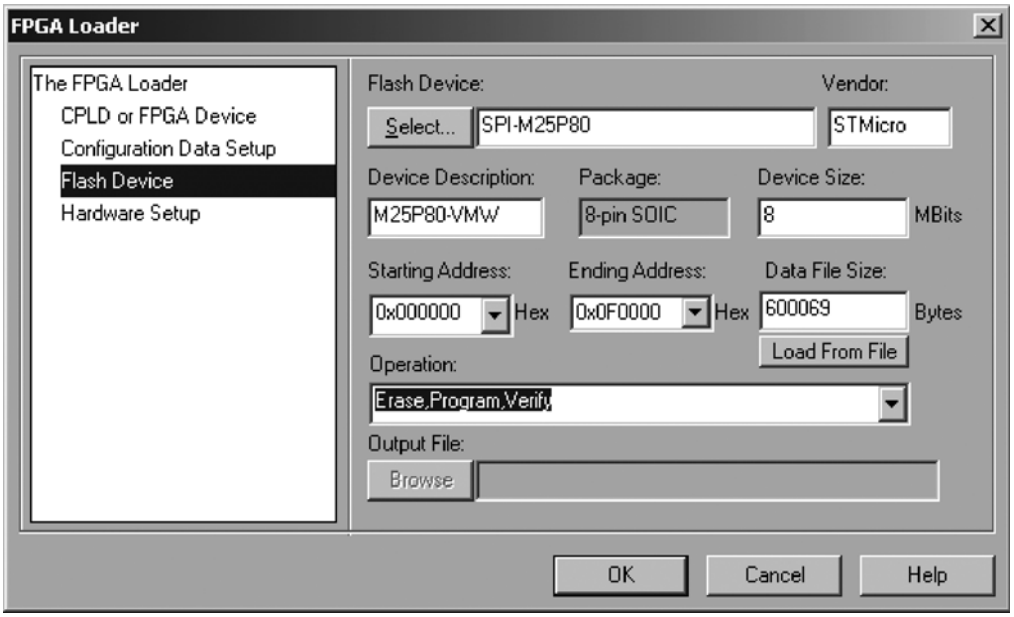
Operation: The operation to be performed on the PLD prior to Flash programming. If the PLD is not yet programmed with the SPI Loader pattern, set this to 'Erase, Program, Verify'. The PLD will then be programmed prior to the Flash. If the PLD has already been programmed, choose 'Bypass'.

Figure 4. Configuration Data Setup



The Configuration Data File contains the data to be programmed into the SPI Flash device. This can be in a generic bitstream (.bit) file, or one of the supported PROM file formats.

Figure 5. Specifying the Flash Device



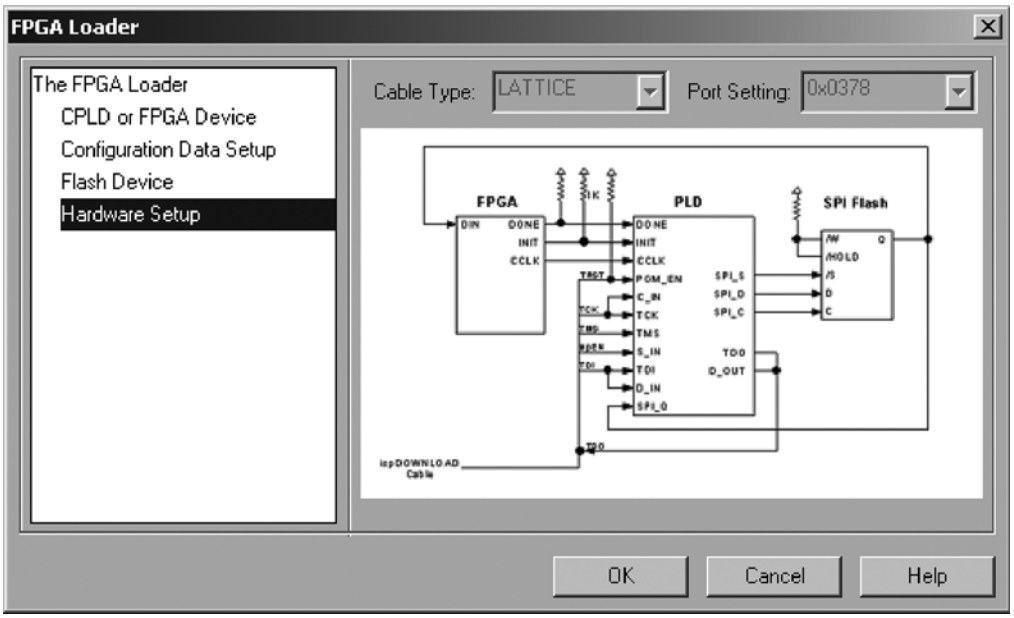
Specify the Flash device by clicking the 'Select' button and choosing from the list of supported Flash devices. The device description and device size fields will be filled out automatically when the device is selected.

Flash Base Address: The address at which to begin the programming of the Flash.

Data File Size: The size of the data file. This field is automatically filled in when a configuration data file is loaded on the previous page.

Operation: The desired operation to perform on the Flash. Normally set to 'Program'.

Figure 6. Hardware Setup Page



For this configuration, the Hardware Setup Page is shown for reference only. No modification is required.

FPGA Configuration

Configuration can be done at any time on a Lattice FPGA device. The FPGA supports several options for configuration, which can be broken down into two categories: Master and Slave. In Master mode, the FPGA provides a configuration clock (CCLK) and signaling to an external memory to read its contents. In Slave mode, an external device must provide the configuration clock and data to the FPGA. For both Master and Slave modes, the device can be configured to accept parallel or serial data. The Master Serial mode is easily utilized for configuration from most serial data sources, and is the foundation for this application.

Master Serial Mode

In master serial mode, the FPGA loads its configuration data automatically from an external ROM on power up or after the PROGRAM pin is toggled. The FPGA provides signaling to the serial PROM. The FPGA device generally has five pins associated with configuration: INIT, PRGM, CCLK, DIN/DI and DONE. DOUT is only necessary if there are multiple FPGAs in a daisy chain configuration. In Master Serial Mode, the FPGA drives the INIT pin low during a configuration cycle if there is an error. The DONE pin will drive high on the completion of a successful configuration.

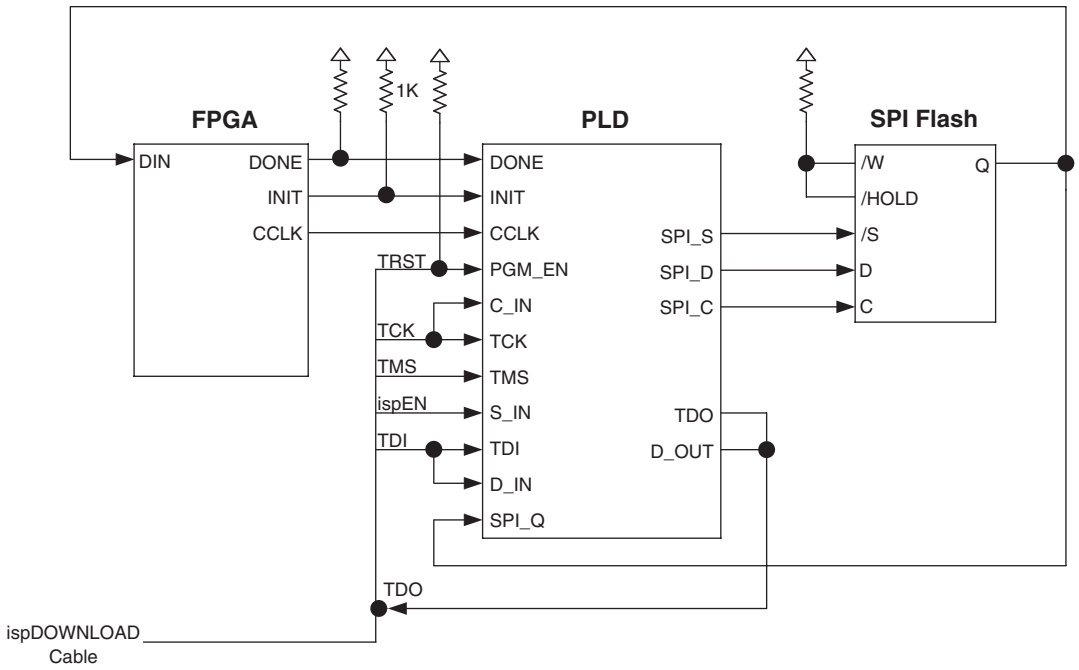
Power Up

On power up, the FPGA device can be configured automatically. While the power is ramping up to the device, a power-on-reset is triggered, external configuration mode pins are sampled, the DONE and INIT pins are driven low, and the internal RAM cleared. Once the device reaches the proper voltage threshold, the INIT pin is released and must be pulled up externally to allow the start of CCLK. In Master mode, the FPGA remains in the initialization state an additional six internal clock cycles after INIT goes high to ensure that any daisy chained devices have a chance to clear.

Emulating a Serial PROM with a PLD and SPI Flash

Figure 7 shows a connection diagram for the PLD-to-SPI Flash interface. The INIT pin from the FPGA holds the PLD in a reset state until it is ready to be configured. When the FPGA releases the INIT pin, the clock CCLK will start shortly thereafter. Internally, the PLD generates the read command for the SPI Flash and specifies a starting address of zero.

Figure 7. Connection Diagram



Notes:

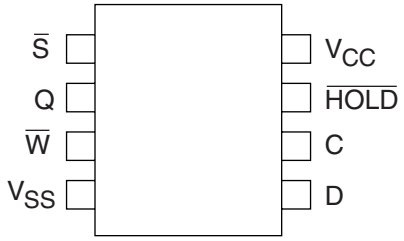
1. The PLD connected to the SPI Flash device must be the first device in the JTAG chain (PLD TDI connected to the ispDOWNLOAD cable).
2. If additional JTAG devices are downstream from the PLD, D_OUT must be directly connected to the TDO of the ispDOWNLOAD cable and the TDO of the last device in the JTAG chain.
3. There can be only one PLD/SPI in the JTAG chain.
4. All resistor values are 5K-10K ohms, unless shown otherwise.
5. TRSTn from the ispDOWNLOAD Cable must not drive TRSTn inputs to devices in the JTAG chain.

The SPI Flash pins are listed in Table 1.

Table 1. SPI Flash Signal Listing

Signal	Description
/S	Chip Select – Enables and disables device operation. When disabled, the device is at standby power levels. When enabled, the device powers up and instructions can be written to and data read from the device.
C	Serial Clock – Provides timing for serial input and output operations.
D	Serial Data In – When the device is enabled, allows instructions, addresses and data to be serially written to the device. Data is latched on the rising edge of the Serial Clock.
Q	Serial Data Out – When the device is enabled, allows data and status to be serially read from the device. Data is shifted out on the falling edge of the Serial Clock.
/HOLD	Allows the device to be paused while it is actively selected. This function is useful when multiple devices are sharing the same SPI signals.
/W	Write Protection – Used to prevent inadvertent writing to the Status Register Block Protect bits.

Figure 8. Standard Pinout for 8-pin SPI Flash Memories



The default .jed file shipped with the ispVM System software targets the ispGAL®22V10AV device and uses the pin connection listed in Table 2. The SPI Flash Loader design files are available on the Lattice web site if a different pin connection or target device is required.

Table 2. Default ispGAL22V10AV Pin Connections

Signal	ispGAL22V10AV	
	32 QFN	28 PLCC
D_IN	3	7
C_IN	14	16
S_IN	15	17
PGM_EN	9	12
D_OUT	22	23
DONE	26	27
INIT	32	4
CCLK	30	2
SPI_D	18	20
SPI_C	17	19
SPI_S	23	24
SPI_Q	1	5

Implementation in a Lattice PLD

The SPI Flash Loader design is implemented using Lattice's ispLEVER design software. The design is distributed in netlist format and can be instantiated in VHDL or Verilog to target the desired PLD with custom pin locations. The design requires fewer than ten macrocells.

Design Fit Process Using ispLEVER

Requirements

- ispLEVER v.4.1 or greater with a valid license

Procedure

Note: This procedure assumes familiarity with the Lattice design software. For more information on using ispLEVER, please see the tutorials included within the help system.

1. Extract the netlist and appropriate instantiation template file from the within the distribution .zip file.
   - spi_loader.bl1 – SPI Loader design in netlist format
   - spi_loader_top.vhd – VHDL instantiation template
   - spi_loader_top.v – Verilog instantiation template

   Note: When targeting an ispGAL22V10 device, support is limited to VHDL for this design.

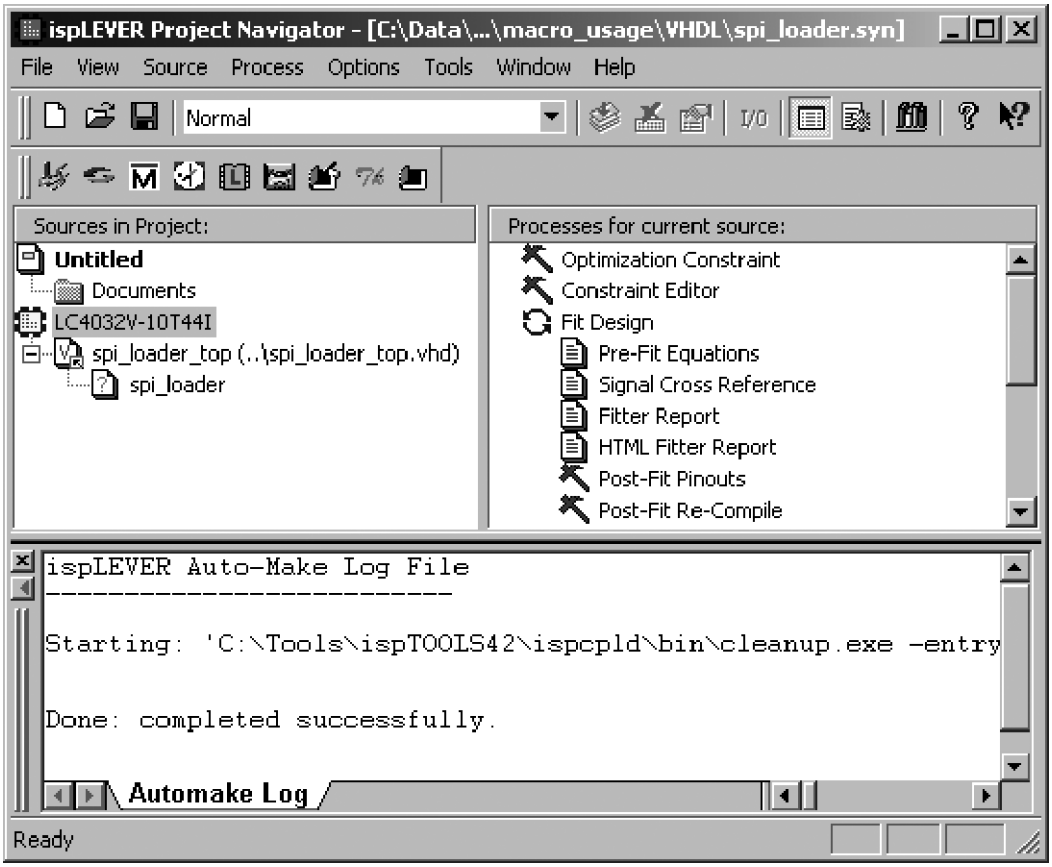
2. Set the file attribute for the spi_loader.bl1 file for read-only access. This will prevent the removal of this file during the clean-up of intermediate files.

3. Launch the ispLEVER design tool and create a new project in the location of the SPI Loader files extracted in step 1. Select the project type according to the desired use of VHDL or Verilog.

4. Select an appropriate PLD device to target.

5. Import the top-level file according to HDL preference. Once imported, the design hierarchy will show the 'spi_loader' module represented as a red question mark as shown in Figure 9.

Figure 9. SPI Loader Design Hierarchy in ispLEVER



Note: The spi_loader module is displayed in the design hierarchy as an unknown entity. Provided that the spi_loader.bl1 file exists in the project directory, the fit process will properly link the logic.

6. Execute the 'Fit Design' process to run the design flow.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-408-826-6002 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com