Minimizing System Interruption During Configuration Using TransFR Technology

October 2015 **Technical Note TN1087**

Introduction

One of the fundamental benefits of using an FPGA is the ability to reconfigure its functionality without removing the device from the system. A number of elaborate mechanisms to provide field updates have been implemented. Accessibility to the system FPGAs can be as simple as a direct cable connection or something as complex as remote access using wireless links or high-level communication protocols.

Current update methods generally require a significant disruption to the system during the configuration update. It is desirable to reduce or eliminate the downtime resulting from reconfiguration due to an update, especially for non-redundant and mission-critical equipment.

Lattice provides TransFR™ (Transparent Field Reconfiguration) Technology to help minimize system interruption. TransFR Technology support is provided in Lattice Diamond® Programmer software. Devices supporting TransFR are listed in Table 1.

Table 1. Device Support for TransFR

Device Family	Non-Volatile Memory
ECP5™	External
ECP5-5G™	External
LatticeECP3™	External
LatticeECP2™ and LatticeECP2M™	External
LatticeXP2™	On-chip Flash
LatticeXP™	On-chip Flash
MachXO3L™	On-chip NVCM
MachXO3LF™	On-chip Flash
MachXO2™	On-chip Flash
MachXO™	On-chip Flash

Note: Device-specific details are listed in the appendices at the end of this document.

Background Programming

Lattice non-volatile Flash FPGAs feature two sets of configuration storage. The SRAM contains the working configuration, and non-volatile Flash memory or NVCM retains the configuration for use as necessary. The contents of the Flash memory or NVCM can be loaded into SRAM automatically at power-up or at any desired time, replacing the need for external boot memory.

Devices without internal non-volatile storage configure their SRAM contents directly from an external device. Such a device might be an SPI serial Flash, a microprocessor, or an EEPROM.

Whether the non-volatile storage is on-chip or external, it can be programmed independently of the SRAM memory space and one can be modified while the other remains intact. One powerful use of this arrangement is programming of the non-volatile configuration memory, while the SRAM continues to operate uninterrupted. This is referred to as background programming.

Boundary Scan Control

Lattice FPGAs also feature a rich set of IEEE 1149.1 (Boundary Scan Test) capabilities, providing additional control when accessing the device through the ispJTAG™ port. Boundary scan cells have the ability to be sampled and preloaded, allowing controllable I/O behavior during programming or configuration.

TransFR Technology

Minimizing system interruption using TransFR Technology utilizes a sequence that combines background programming capabilities with TransFR. The result is a process in which systems can be upgraded with very little disruption. Careful system design allows TransFR to be completely transparent to the application.

Lattice supports JTAG and non-JTAG mode TransFR operation, depending on device family. Refer to appendices at the end of the document for device specific implementation details. The JTAG port uses BSCAN cells to either capture the current state on the I/O, or to force the I/O to a known state. JTAG port controlled TransFR operations are the only TransFR sequences able to control the boundary scan cells. Non-JTAG mode TransFR operations are bitstream driven. Here are some advantages and disadvantages for each TransFR operation mode:

JTAG Mode TransFR

Advantages:

- Allows user to release I/Os at a user-defined time during TransFR
- Allows user to customize I/O value during TransFR

Disadvantages:

- JTAG port on board is required because the operation is JTAG command based
- Operation is relatively complicated because it is command based

The following is a detailed description of JTAG mode TransFR.

Phase 1: Background Programming. The non-volatile memory (internal or external) is reprogrammed while the SRAM is running undisturbed, allowing the system to continue operating without any disruption.

Phase 2: I/O states are captured and held or driven to a user-defined level using JTAG commands. Outputs will retain these levels throughout the reconfiguration process. As far as the system is concerned, this effectively pauses the FPGA, keeping any critical control and status outputs in their desired states during the system update.

Phase 3: While the I/O states remain under the control of boundary scan, JTAG commands are used to initiate the transfer of the new functionality from non-volatile memory to the SRAM configuration space. For device-specific implementation details, refer to the appendices at the end of this document.

After the SRAM is configured, the I/O settings will return to those specified by the user. The GSR signal is asserted internally to place the device into a predictable state.

After reconfiguration is completed and prior to the exit of boundary scan mode in Phase4, the internal device logic is actively interpreting input signals. This time period can be used for a number of purposes to allow for a custom reactivation. Common uses include:

- Synchronization of PLLs to incoming clock sources
- Manipulation of event counters and state machines into desired operational states
- Ensuring status and error indicators are properly initialized

Phase 4: I/Os are released from boundary scan control and back to the new desired function. The internal logic seamlessly reassumes control of the I/Os.

Design Considerations for JTAG Mode TransFR

The clock source controlling the JTAG state machine and boundary scan circuitry is often asynchronous to the system clocks. To maintain desired output levels, it is recommended that an “indicator” input be included in the design to force the internal logic to the desired state. This input can be asserted during the TransFR process to ensure that sequential design elements are initialized to their desired values.

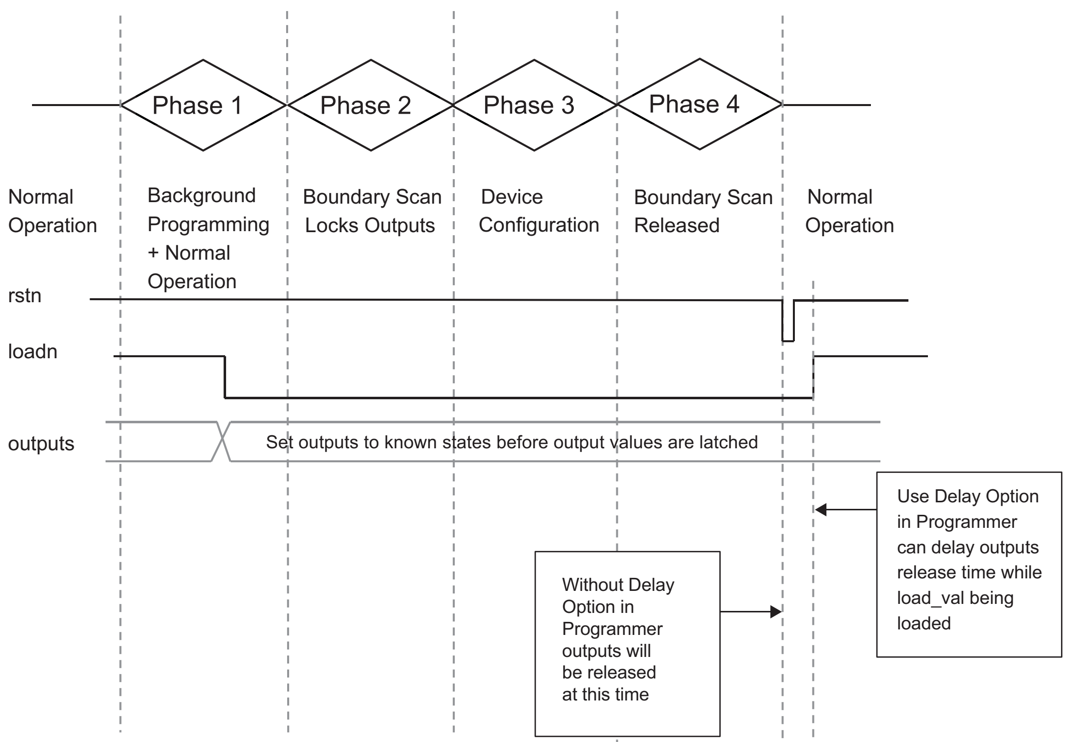
An example of this, as shown in Figure 1, is a synchronous load function for a counter. Determining the number of clocks required to reach the target value and the synchronization with the JTAG clock is often not practical. Inclusion of a load function for the counter allows it to be placed in a desired state during the TransFR process for return to normal post-configuration device operation, as illustrated in Figure 2.

Figure 1. Verilog Code Fragment for Synchronous Load

```
always@ (posedge clk or negedge rstn)
begin
  if (~rstn)
    cnt <= 32'h00000000;
  else begin
      if (loadn == 1'b0)
        cnt <= load_val;
      else
        if (cnten == 1'b1)
          cnt <= cnt + 1;
  end
end
```

Note: loadn is the "indicator" signal.

Figure 2. Example JTAG Mode TransFR Sequence



Non-JTAG Mode TransFR

Advantages:

- JTAG port is NOT required; any CONFIG port may be used
- Operation is relatively simple; most of these only need to refresh the device

Disadvantages:

- Does not allow user to release I/Os at a specific time during TransFR
- Does not allow user to customize I/O value during TransFR. I/O values are always kept as is.

The following is a detailed description of non-JTAG mode TransFR:

Phase 1: Background Programming. The non-volatile memory (internal or external) is reprogrammed while the SRAM is running undisturbed, allowing the system to continue operating without any disruption.

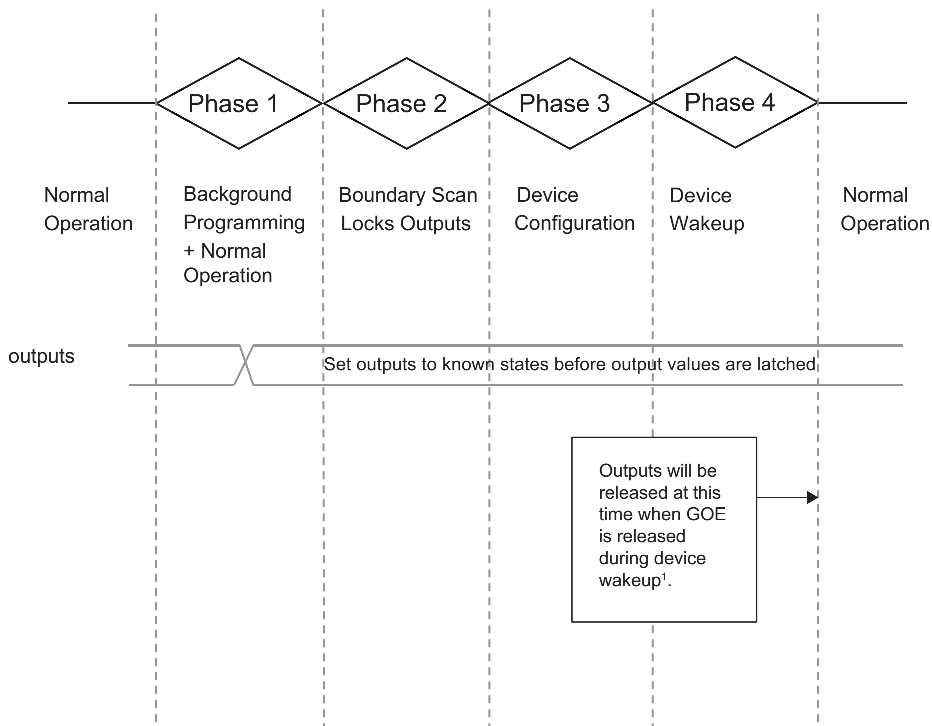
Phase 2: Device starts to refresh. I/O states are captured into I/O latches and held. Outputs retain these levels throughout the reconfiguration process. As far as the system is concerned, this effectively pauses the FPGA, keeping any critical control and status outputs in their desired states during the system update.

Phase 3: I/Os remain under the control of I/O latches while the new functionality transfers from non-volatile memory to SRAM configuration space.

Phase 4: I/Os are released from I/O latch to user function only when GOE is released during wakeup stage.

Design Consideration for Non-JTAG Mode TransFR
In Non-JTAG Mode TransFR, I/Os get released at the same time as Global Output Enable (GOE) signal get released during device wake up. To maintain all output states after device wake up, resetting on the output registers needs to be avoided after wake up. In MachXO2 as an example, GOE signal get released before Global Set/Reset (GSR) signal. If design involves GSR, then asserting GSR could reset all the output registers.

Figure 3. Example Non-JTAG Mode TransFR Sequence



1. For details on device wakeup, please refer to the Wake-up section in TN1104, MachXO2 Programming and Configuration Usage Guide.

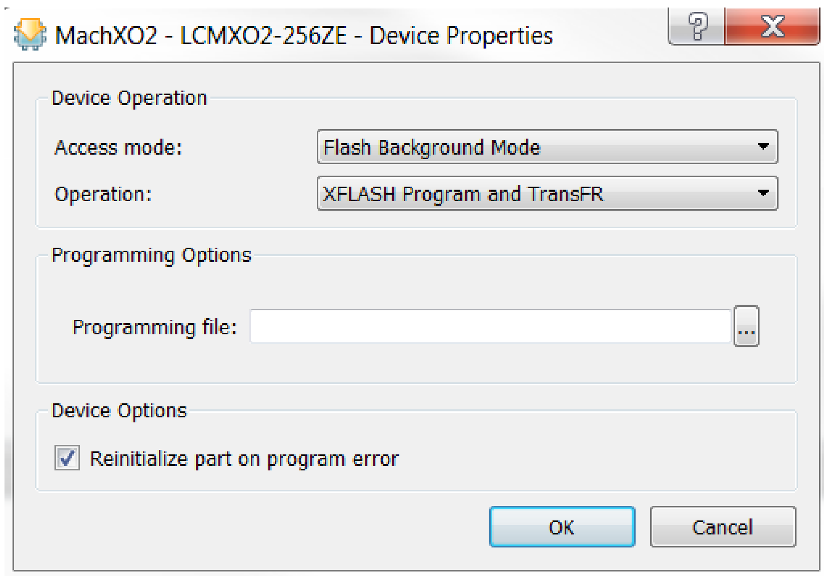
TransFR Using Programmer

Programmer incorporates the TransFR flow for Lattice FPGAs in two operations. The first operation targets the background programming of the non-volatile memory, while the second controls the I/O states and initiates the configuration process.

Note: External configuration memory programming can also be accomplished through other means such as an on-board microprocessor. Provided that this does not disrupt the operation of the FPGA, the TransFR operation can be utilized.

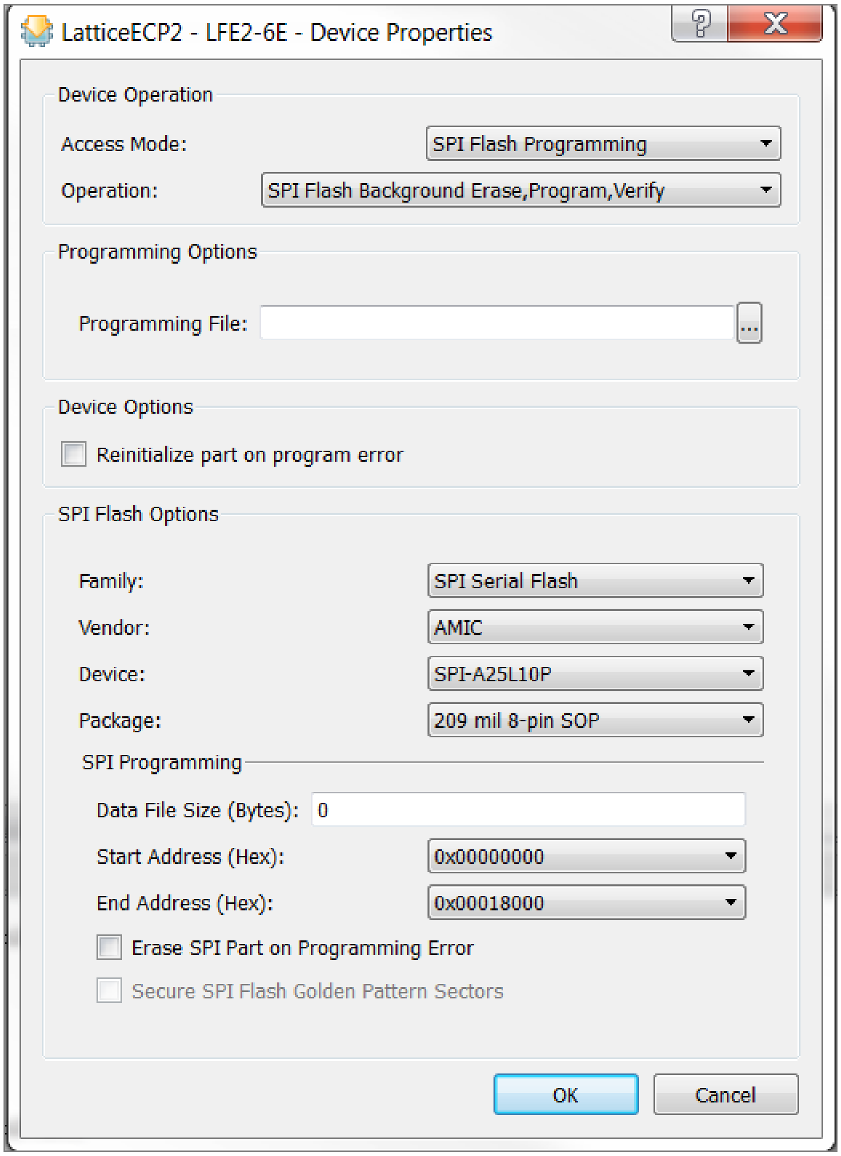
For applicable devices, background programming is specified in Programmer from the “Operation” drop-down list in the Device Properties dialog box. This is shown in Figure 4.

Figure 4. Background Programming Selection in Programmer (MachXO2)



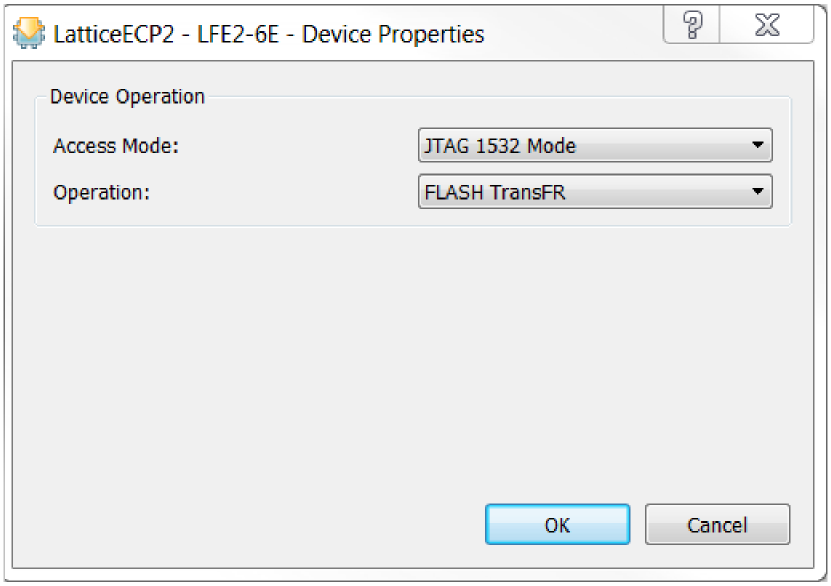
Programmer also provides the capability to transparently program SPI serial Flash devices through the connected SRAM-based FPGA, as shown in Figure 5.

Figure 5. SPI Flash Programming Selection in Programmer (LatticeECP2)



The remaining portion of the TransFR procedure is selected by choosing a dedicated device operation, as shown in Figure 6. For device-specific details, refer to the appendices near the end of this document.

Figure 6. TransFR Operation Selection in Programmer (LatticeECP2)



Embedded and Third Party Support

The Diamond Embedded source code supports TransFR Technology. The embedded instructions for TransFR can be exported from the Diamond embedded directory to a VME file for use with a microprocessor.

In addition to embedded routines, support is provided for the creation of standard SVF and ATE file formats for use outside of the Programmer environment from Deployment Tool.

Technical Support Assistance

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

Revision History

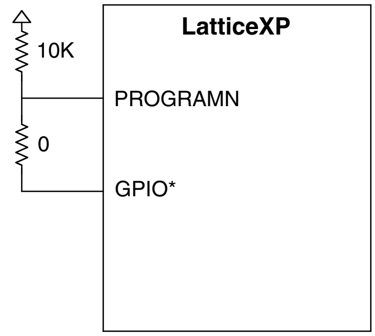
Date	Version	Change Summary
October 2015	3.7	Updated Introduction section. Revised Table 1, Device Support for TransFR. Added ECP5-5G.
		Updated Appendix G. ECP5 and ECP5-5G section. Added ECP5-5G.
		Updated Technical Support Assistance section.
March 2015	3.6	Product name/trademark adjustment.
		Updated Introduction section. Revised Table 1, Device Support for TransFR. Added MachXO3LF.
		Added NVCM information.
March 2014	3.5	Updated Table 1, Device Support for TransFR. Changed LatticeECP4UM to ECP5.
		Updated Appendix G. ECP5, Device Support for TransFR. Changed LatticeECP4UM to ECP5.
February 2014	3.4	Added support for LatticeECP4UM.
		Updated Table 1, Device Support for TransFR
		Added Appendix G. LatticeECP4UM.
September 2013	3.3	Introduced the Non-JTAG mode TransFR.
		Separated device families based on TransFR mode
		Updated example figures using Diamond Programmer and Deployment Tools screen shots.
		Updated Technical Support Assistance information.
June 2013	3.2	Added toggling PROGRAMN pins as an option for initiating the reconfiguration process for TransFR.
		Updated the Selecting TransFR for LatticeECP3 screen shot.
		Deleted the descriptions of I/O State options.
May 2013	3.1	Updated the Enabling TransFR in the Design Planner screen shot for the LatticeECP2/M.
April 2013	3.0	Updated document with new corporate logo.
		Created separate appendices for MachXO and MachXO2 and updated figures for selecting TransFR for MachXO2.
		Updated the Enabling TransFR in the Design Planner screen shot.
August 2011	02.9	Added User SPI during background programming caution.
May 2011	02.8	Documented ispVM “TransFR Options...” for MachXO/XO2, LatticeXP2, LatticeECP2, and LatticeECP3 devices listed.
November 2010	02.7	Updated for LatticeECP3 and MachXO2 device support.
October 2008	02.6	Updated LatticeXP2 appendix.
May 2008	02.5	Updated screen shots.
		Enhanced description of I/O state options.
		Corrected MachXO behavior description.
May 2007	02.4	Added LatticeXP2 appendix.
January 2007	02.3	LatticeECP2/M appendix: added note about using dual-purpose pins and added ispLEVER software setting.
September 2006	02.2	Added LatticeECP2M information.
	02.1	Added LatticeECP2 information.
		Updated screen shots
		Added Table 1 to list device family support.
—	—	Previous Lattice releases.

Appendix A. LatticeXP

In the LatticeXP device family, the Flash to SRAM transfer occurs on the low-to-high transition of the PROGRAMN pin. To allow control of the PROGRAMN pin from the ispJTAG port, an external connection is required to a General Purpose I/O (GPIO) pin, as shown in Figure 7. The operation is supported by the ispVM System Software and is not supported by Programmer. LatticeXP devices are featured with JTAG mode TransFR.

Note: This GPIO pin should not be used for any other purpose than updating via TransFR. During operational mode, a low on this pin will result in undesired reconfiguration cycles.

Figure 7. PROGRAMN to GPIO Connection for LatticeXP



*This pin is any available user-specified I/O.

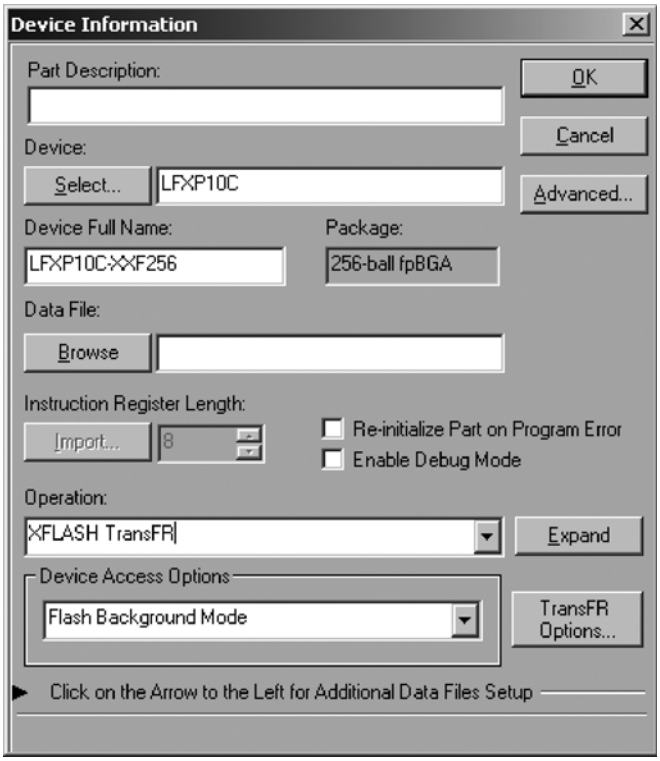
The location of this GPIO pin is user-selectable using ispVM System. Without specification, this pin is assigned to a default location, shown in Table 2. Alternatively, ispVM System may also be configured to directly drive the PROGRAMN pin from the ispEN signal of the ispDOWNLOAD® cable. The PROGRAMN pin can also be asserted at the appropriate time by an external source, such as a microprocessor.

Table 2. LatticeXP Default GPIO for PROGRAMN Control

LatticeXP Device	Pin Function	100 TQFP	144 TQFP	208 PQFP	256 fpBGA	388 fpBGA	484 fpBGA
XP3	PROGRAMN	3	1	3	—	—	—
	GPIO	8	9	14	—	—	—
XP6	PROGRAMN	—	1	3	C2	—	—
	GPIO	—	9	14	E2	—	—
XP10	PROGRAMN	—	—	—	C2	F4	—
	GPIO	—	—	—	D1	H4	—
XP15	PROGRAMN	—	—	—	C2	F4	F5
	GPIO	—	—	—	E2	H3	H3
XP20	PROGRAMN	—	—	—	C2	F4	F5
	GPIO	—	—	—	E2	H3	H3

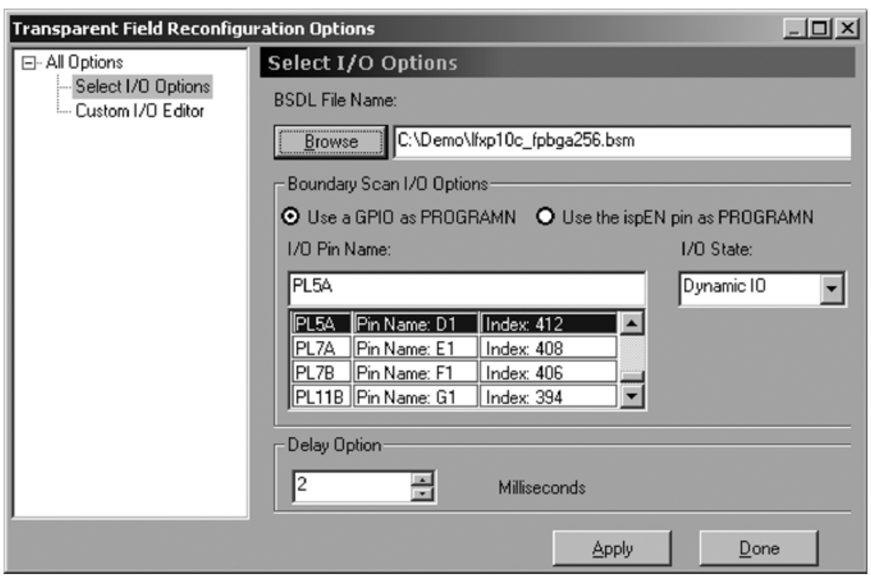
To select the control source of the LatticeXP PROGRAMN pin, the device package must be specified when selecting the device. The 'TransFR Options...' button in the 'Device Information' dialog, then becomes available, as shown in Figure 8.

Figure 8. Device Information Dialog with TransFR



The resulting dialog appears as shown in Figure 9.

Figure 9. TransFR Options Dialog



BSDL File Name: A BSDL file must be supplied when specifying the state of I/Os during the TransFR operation. BSDL files can be downloaded at www.latticesemi.com.

I/O State: Provides control of the I/O behavior during the TransFR operation.

Dynamic I/O: Specifies outputs are to be sampled and held to their last values (Leave-Alone), unless overridden using the Custom I/O Editor.

Custom I/O: Outputs are tri-stated, unless overridden using the Custom I/O Editor.

Delay Option: Specifies the delay (in milliseconds) from the start of reconfiguration of the SRAM from Flash to the release of the I/Os.

To select a particular GPIO for connection with the PROGRAMN pin:

1. Select the ‘Use a GPIO as PROGRAMN’ option
2. Provide a BSDL file for the device. This can be downloaded at www.latticesemi.com.
3. Select the desired pin from the list.

To enable the ispEN output of the ispDOWNLOAD cable for control of PROGRAMN, choose the ‘Use the ispEN pin as PROGRAMN’ option. The ‘PROG Pin Connected’ option must also be enabled. This can be found from the menu as Options->Cable and I/O Port Setup.

The following device limitations apply from the time the boundary scan cells are locked in Phase2 until their release in Phase4:

- Open-drain outputs revert to standard, totem-pole drivers.
- Differential outputs are driven as independent single-ended drivers during boundary scan operations. When a differential output is captured and held, the positive (true) pin is driven to that logical value using an LVCMOS output driver. The negative (complement) pin of the differential pair is tri-stated.

Additionally, during the brief configuration process the programmable I/Os default to the following characteristics, while still under the control of boundary scan:

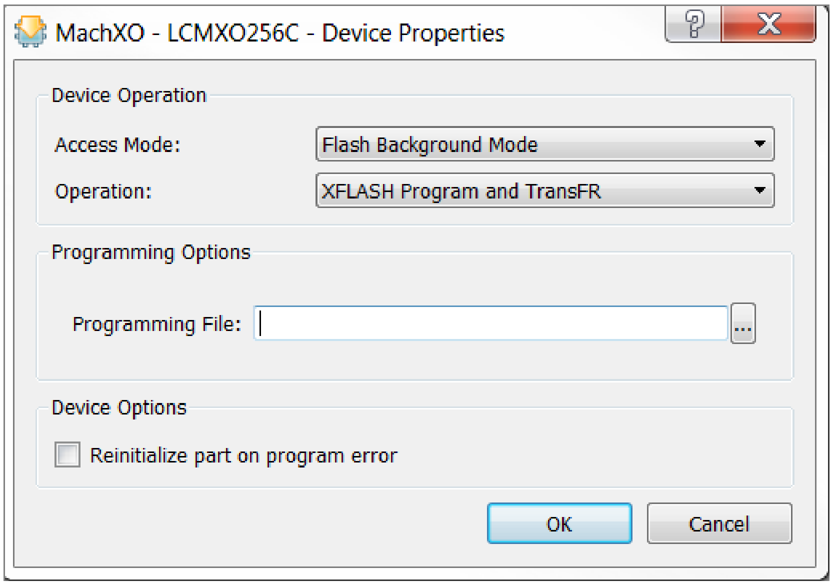
- Outputs are single-ended, totem-pole drivers with a maximum VOH of VCCIO and 8mA drive strength.
- Internal weak pullup resistors are enabled.

Appendix B. MachXO

In the MachXO device family, Flash to SRAM configuration can be initiated by an instruction from the ispJTAG port. No additional external connections are required to facilitate TransFR capabilities. MachXO devices are featured with JTAG mode TransFR.

To specify TransFR in Programmer, choose the corresponding operation, as shown in Figure 10.

Figure 10. Selecting TransFR for MachXO



Edit the I/O state by right-clicking the device and selecting **Edit I/O State** as shown in Figure 11 and Figure 12.

Figure 11. Edit I/O State Menu Option

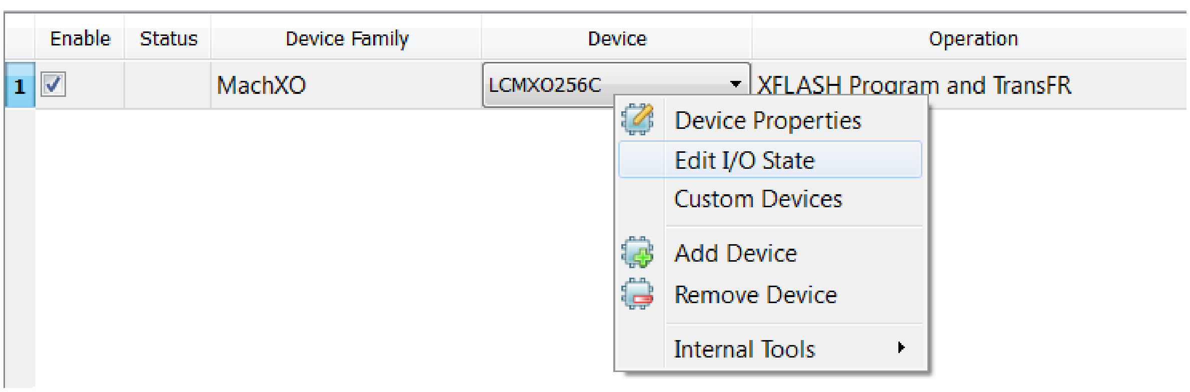
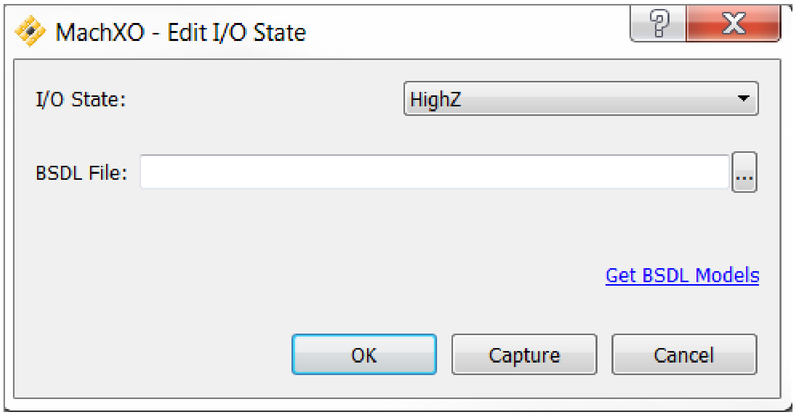


Figure 12. Edit I/O State Dialog Box



I/O State: Provides control of the I/O behavior during the TransFR operation.

- **HighZ** – Specifies all outputs are tri-stated.
- **All 1s** – Specifies all outputs are high.
- **All 0s** – Specifies all outputs are low.
- **Leave Alone** – Specifies outputs are to be sampled and held to their last values.
- **Dynamic I/O** – Specifies outputs are to be sampled and held to their last values (Leave Alone), unless a value is explicitly specified using the Custom I/O Editor. A BSDL file is required for this option.
- **Custom** – Outputs are tri-stated, unless a value is explicitly specified using the Custom I/O Editor. A BSDL file is required for this option.
- **TransFR Options** – Brings up a Delay Option pop-up menu. Specifies the delay (in milliseconds) from the start of reconfiguration of the SRAM from Flash to the release of the I/Os. (*Temporarily not available in Programmer*)

The following device limitations apply from the time the boundary scan cells are locked in Phase2 until their release in Phase4:

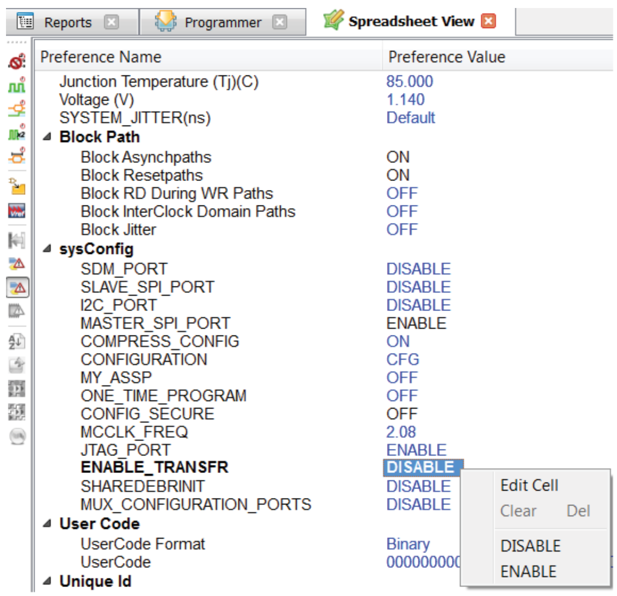
- Differential outputs are driven as independent single-ended drivers during boundary scan operations. When a differential output is captured and held, the positive (true) pin is driven to that logical value using an LVCMOS output driver. The negative (complement) pin of the differential pair is tri-stated.

Appendix C. MachXO2 and MachXO3

In the MachXO2 and MachXO3 device families, Flash to SRAM configuration or NVCM to SRAM configuration can be initiated by an instruction from any configuration port, not just from ispJTAG port. No additional external connections are required to facilitate TransFR capabilities. MachXO2 and MachXO3 devices are featured with non-JTAG mode TransFR.

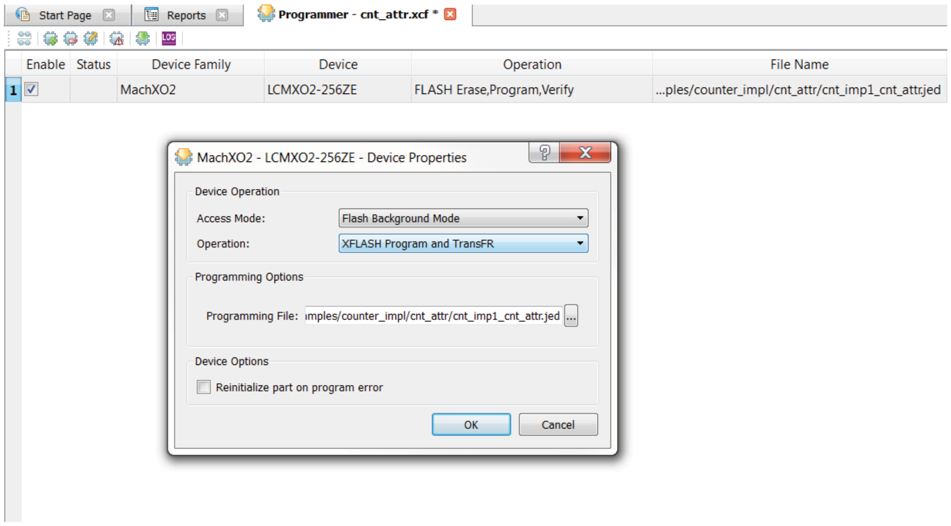
To enable TransFR, set ENABLE_TRANSFR to ENABLE through Global Preference, as shown in Figure 13, in both patterns (the current one in the flash and the new one to be programmed into the flash).

Figure 13. Selecting TransFR for MachXO2



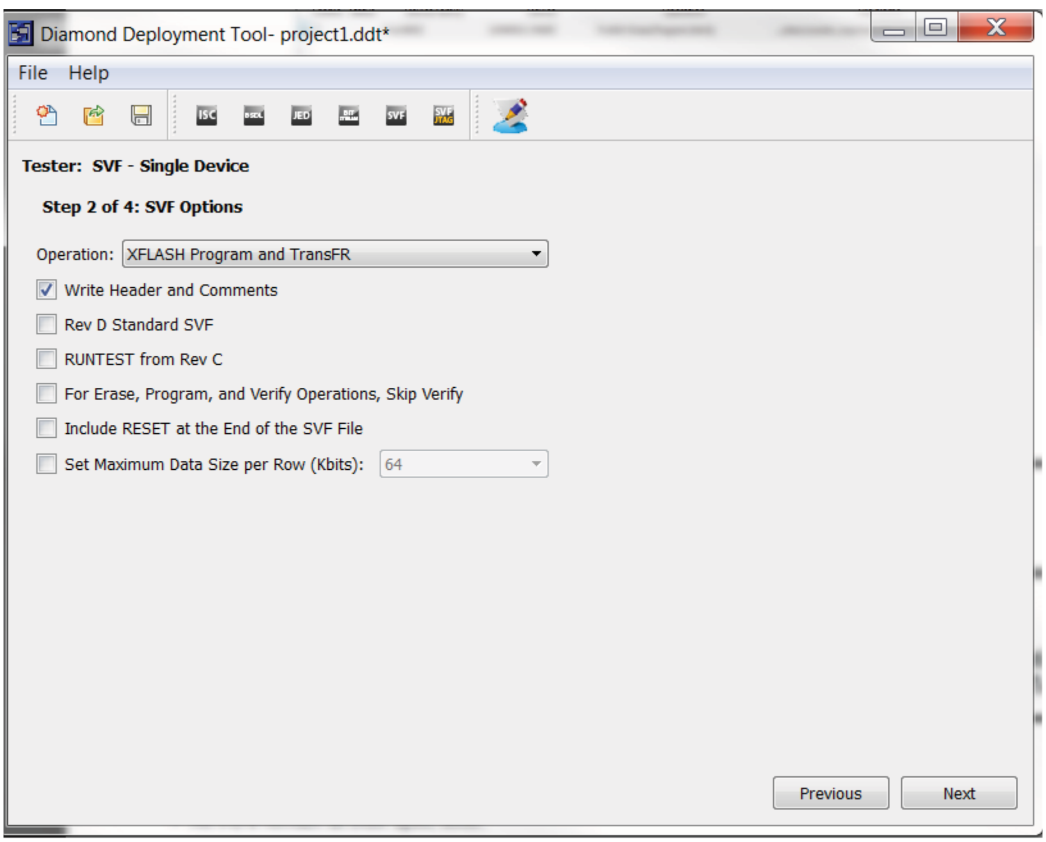
To specify TransFR in Programmer, choose the corresponding operation in Figure 14. In MachXO3, choose **NVCM background Mode** as Access Mode and **XNVCM Program and TransFR** as Operation.

Figure 14. Selecting TransFR for MachXO2



To specify TransFR in Deployment Tool, choose the corresponding operation in Figure 15.

Figure 15. Selecting TransFR for MachXO2



Key Features:

- I/O's are automatically sampled and held at their last state at the falling edge of INITN.
- All I/O's remain at their spec level.
- I/O refresh is automatic.
- If implemented, the hardened user SPI port outputs are disabled (tri-stated). (The configuration SPI port is not impacted.)

Limitation:

- Whereas the MachXO allows the user to release the TransFR IOs at a specific time via a BSCAN operation, the MachXO2 and MachXO3 do not support this ability.
- Whereas the MachXO allows the user to preload specific values into I/O through BSCAN operation during TransFR, the MachXO2 and MachXO3 do not support this ability.

Appendix D. LatticeECP2/M

The LatticeECP2/M device family of SRAM-only FPGAs relies on non-volatile external memory to store configuration data. TransFR allows configuration via the sysCONFIG port while the I/O pins remain in a locked state for minimizing system disruption.

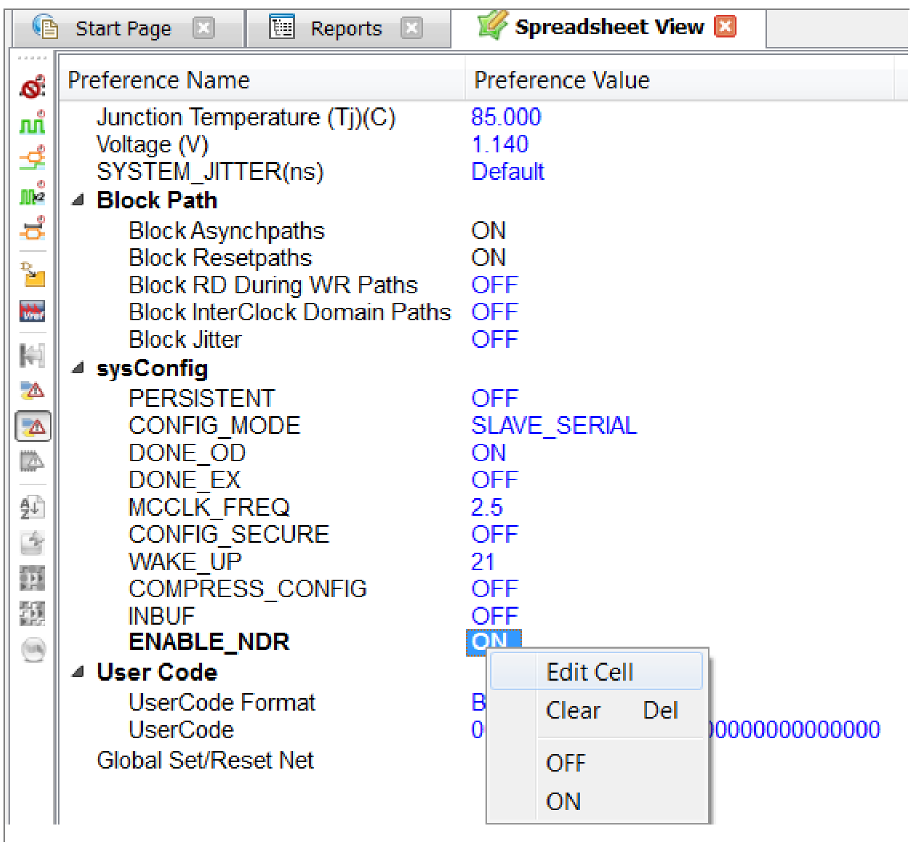
Note: The dual-purpose I/O pins in bank 8 should not be used for signals critical to the system during TransFR. These pins are not controlled by boundary scan in phases 2 through 4 to allow full operation of the sysCONFIG port. If background programming of the SPI Flash from the LatticeECP2/M is desired, these pins cannot be used as I/O pins since the PERSISTENT preference must be set to ON.

The LatticeECP2/M supports a number of configuration sources, such as SPI serial Flash and parallel modes. All sysCONFIG modes are supported with TransFR. Refer to TN1108, LatticeECP2/M sysCONFIG Usage Guide for the details and connection requirements for each mode.

Below is an example of a JTAG mode TransFR procedure.

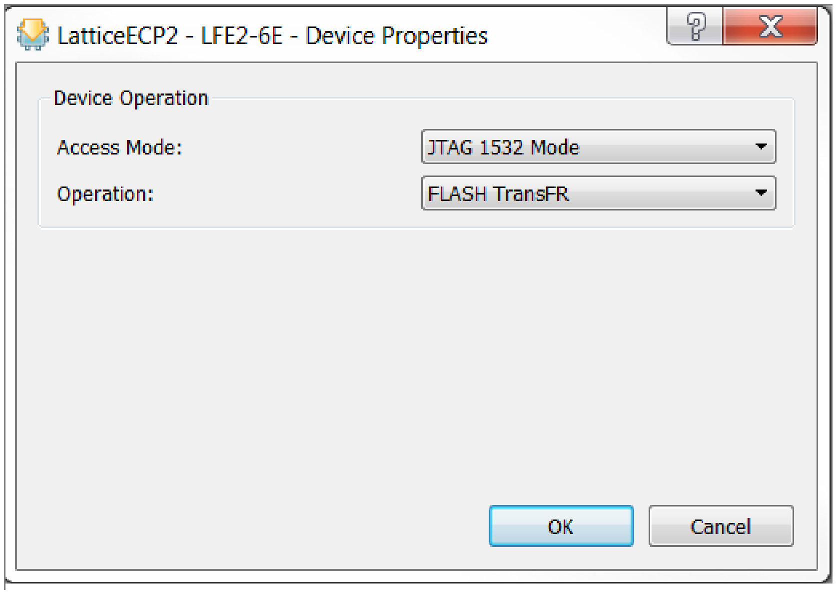
To enable TransFR for the LatticeECP2/M, a preference in Diamond must be set. This can be done using the Global tab in the Spreadsheet View, as shown in Figure 16. To enable TransFR, the ENABLE_NDR option must be set to ON.

Figure 16. Enabling TransFR in Diamond



The reconfiguration process for TransFR is initiated via commands through the ispJTAG port. To specify TransFR in Programmer, choose the corresponding option, as shown in Figure 17.

Figure 17. Selecting TransFR for LatticeECP2



Edit the I/O state by right-clicking the device and selecting **Edit I/O State** as shown in Figure 18 and Figure 19.

Figure 18. Edit I/O State Menu Option

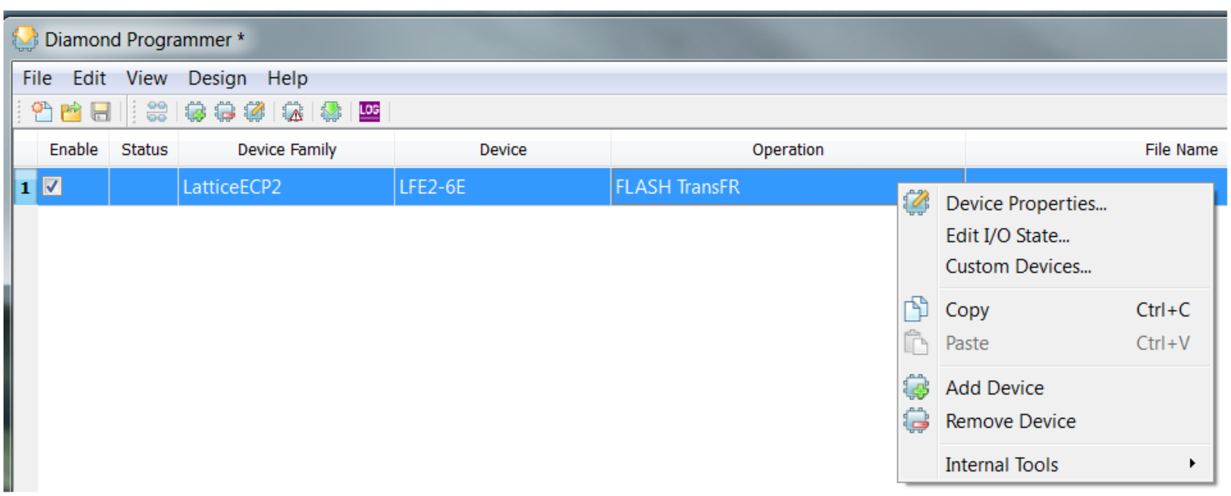
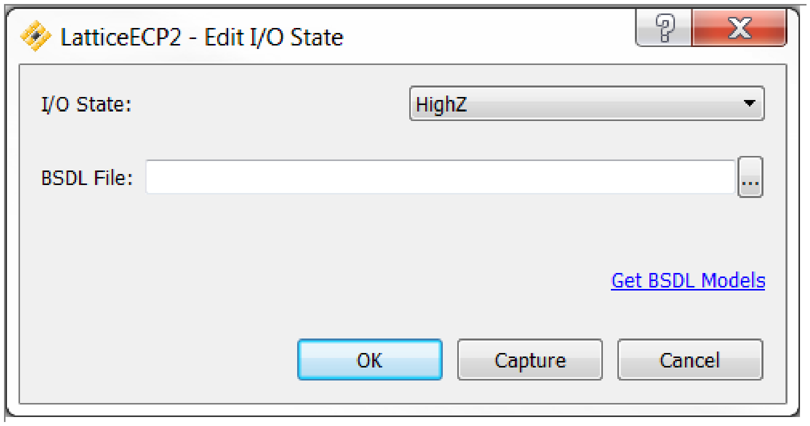


Figure 19. Edit I/O State Dialog Box



I/O State: Provides control of the I/O behavior during the TransFR operation.

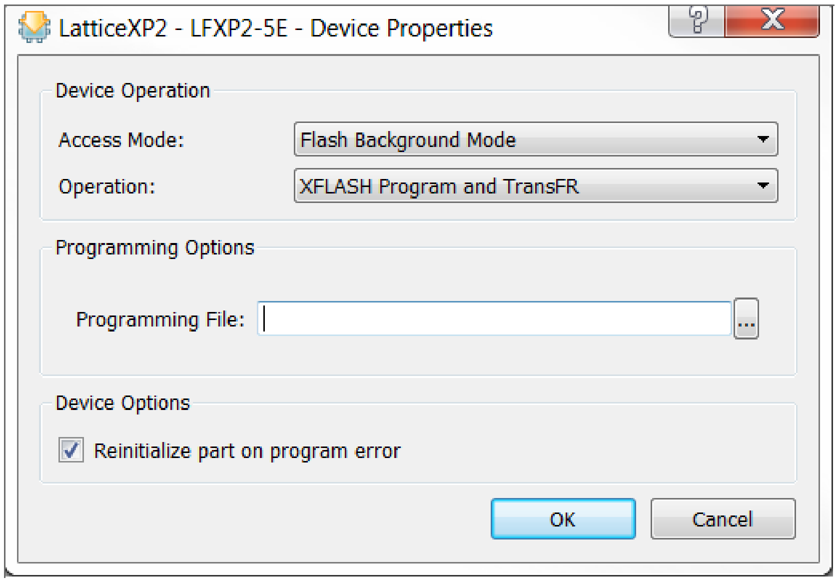
- **HighZ** – Specifies all outputs are tri-stated.
- **All 1s** – Specifies all outputs are high.
- **All 0s** – Specifies all outputs are low.
- **Leave Alone** – Specifies outputs are to be sampled and held to their last values.
- **Dynamic I/O** – Specifies outputs are to be sampled and held to their last values (Leave Alone), unless a value is explicitly specified using the Custom I/O Editor. A BSDL file is required for this option.
- **Custom** – Outputs are tri-stated, unless a value is explicitly specified using the Custom I/O Editor. A BSDL file is required for this option.
- **TransFR Options** – Brings up a Delay Option pop-up menu. Specifies the delay (in milliseconds) from the start of reconfiguration of the SRAM from Flash or NVCM to the release of the I/Os.

Appendix E. LatticeXP2

The LatticeXP2 device family supports JTAG mode TransFR from the on-chip Flash memory.

The reconfiguration process for TransFR is initiated via commands through the ispJTAG port. To specify TransFR in Programmer, choose the corresponding option, as shown in Figure 20.

Figure 20. Selecting TransFR for LatticeXP2



Edit the I/O state by right-clicking the device and selecting **Edit I/O State** as shown in Figure 21 and Figure 22.

Figure 21. Edit I/O State Menu Option

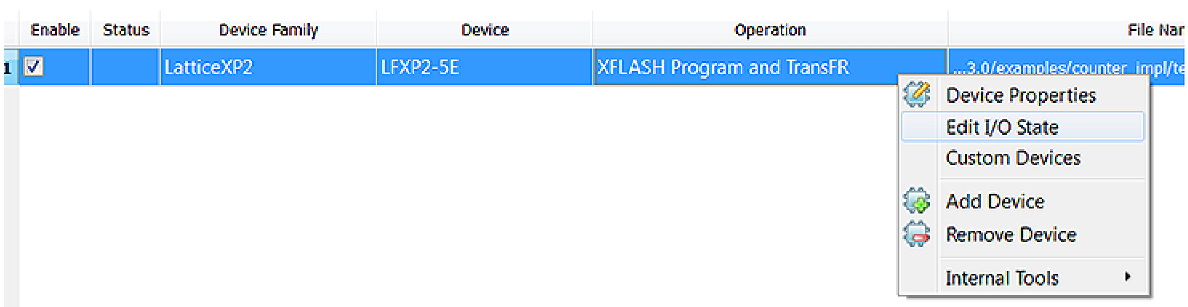
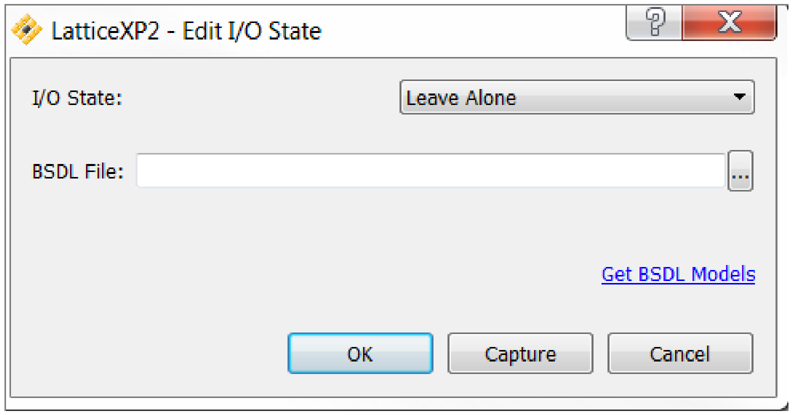


Figure 22. Edit I/O State Dialog Box



I/O State: Provides control of the I/O behavior during the TransFR operation.

- **HighZ** – Specifies all outputs are tri-stated.
- **All 1s** – Specifies all outputs are high.
- **All 0s** – Specifies all outputs are low.
- **Leave Alone** – Specifies outputs are to be sampled and held to their last values.
- **Dynamic I/O** – Specifies outputs are to be sampled and held to their last values (Leave Alone), unless a value is explicitly specified using the Custom I/O Editor. A BSDL file is required for this option.
- **Custom** – Outputs are tri-stated, unless a value is explicitly specified using the Custom I/O Editor. A BSDL file is required for this option.
- **TransFR Options** – Brings up a Delay Option pop-up menu. Specifies the delay (in milliseconds) from the start of reconfiguration of the SRAM from Flash to the release of the I/Os.

Appendix F. LatticeECP3

The LatticeECP3 device family of SRAM-only FPGAs relies on non-volatile external memory to store configuration data. TransFR allows configuration via the sysCONFIG port while the I/O pins remain in a locked state for minimizing system disruption.

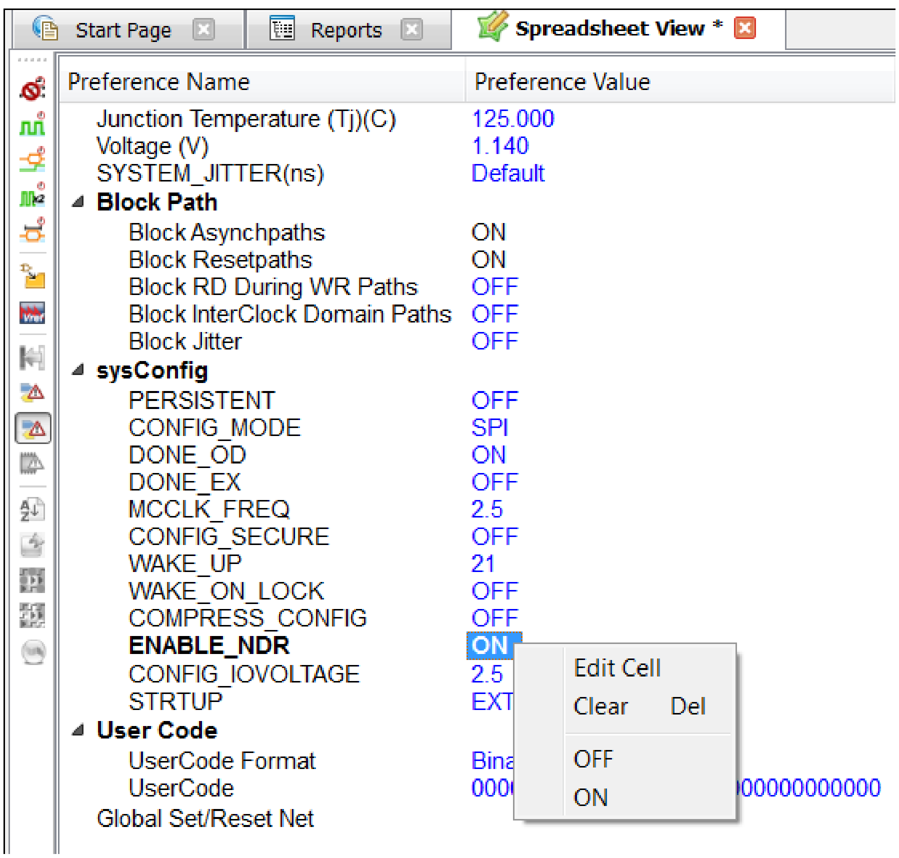
Note: The dual-purpose I/O pins in bank 8 should not be used for signals critical to the system during TransFR. These pins are not controlled by boundary scan in phases 2 through 4 to allow full operation of the sysCONFIG port. If background programming of the SPI Flash from the LatticeECP3 is desired, these pins cannot be used as I/O pins since the PERSISTENT preference must be set to ON.

The LatticeECP3 supports a number of configuration sources, such as SPI serial Flash and parallel modes. All sysCONFIG modes are supported with TransFR. Refer to TN1169, LatticeECP3 sysCONFIG Usage Guide for the details and connection requirements for each mode.

Below is an example of a non-JTAG mode TransFR procedure.

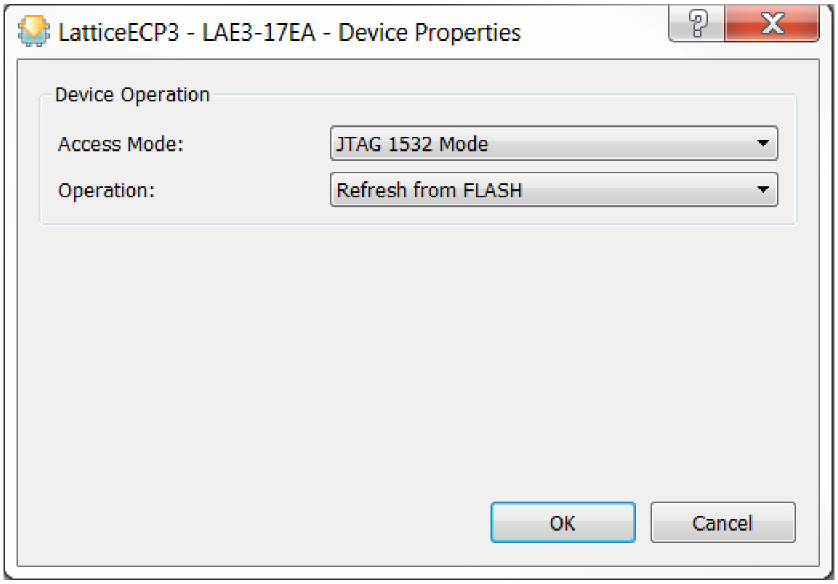
To enable TransFR for the LatticeECP3, a preference in Diamond must be set. This can be done using the Global tab in the Spreadsheet View, as shown in Figure 23. To enable TransFR, the ENABLE_NDR option must be set to ON.

Figure 23. Enabling TransFR in Diamond



The reconfiguration process for TransFR is initiated via refresh command or by toggling PROGRAMN pins. To specify TransFR in Programmer, choose the corresponding option, as shown in Figure 24.

Figure 24. Selecting TransFR for LatticeECP3



Appendix G. ECP5 and ECP5-5G

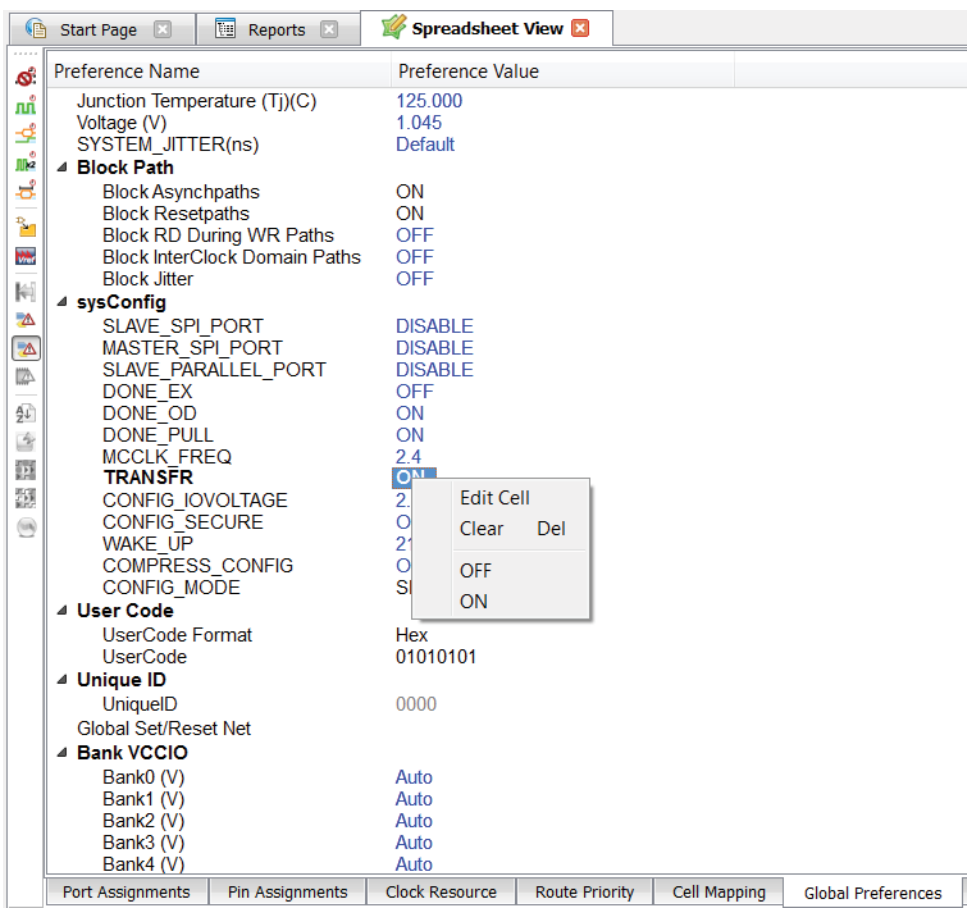
The ECP5 and ECP5-5G families of SRAM-only FPGAs rely on non-volatile external memory to store configuration data. TransFR allows configuration via the sysCONFIG port while the I/O pins remain in a locked state for minimizing system disruption.

Note: The dual-purpose I/O pins in bank 8 should not be used for signals critical to the system during TransFR. These pins are not controlled by boundary scan in phases 2 through 4 to allow full operation of the sysCONFIG port. If background programming of the SPI Flash from the ECP5 and ECP5-5G devices is desired, these pins cannot be used as I/O pins since the PERSISTENT preference must be set to ON.

The ECP5 and ECP5-5G devices support both JTAG mode and non-JTAG TransFR modes.

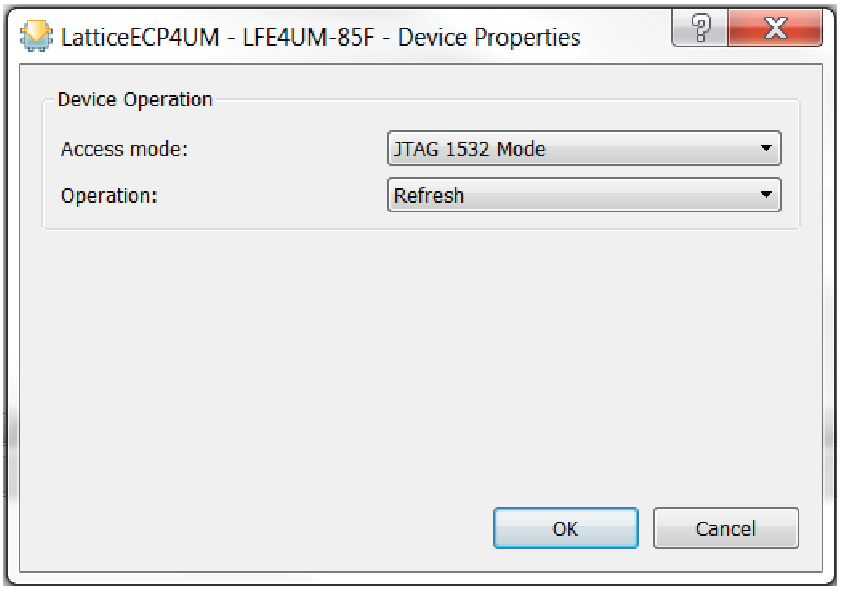
Below is an example of a non-JTAG mode TransFR procedure. To enable TransFR for the ECP5 and ECP5-5G devices, a preference in Diamond must be set. This can be done using the Global tab in the Spreadsheet View, as shown in Figure 25. To enable TransFR, the TRANSFR option must be set to ON.

Figure 25. Enabling TransFR in Diamond



The reconfiguration process for TransFR is initiated via refresh command or by toggling PROGRAMN pins. To specify TransFR in Programmer, choose the corresponding option, as shown in Figure 26.

Figure 26. Selecting TransFR for the ECP5 and ECP5-5G Devices



Glossary

Configuration: The act of writing to volatile configuration memory (such as SRAM) to set the device behavior. Configuration can occur from external sources or via internal transfer from non-volatile memory.

JTAG: Joint Test Action Group. IEEE 1149.1 boundary scan access port for board-level continuity and testing. Also used as an access method to program programmable logic.

FPGA: Field Programmable Gate Array. A high density programmable logic device containing small logic cells interconnected through a distributed array of programmable switches.

GPIO: General Purpose Input/Output pin.

PLL: Phase-Locked Loop. Used in programmable logic for clock management. Common uses include clock multiplication/division and time/phase adjustment.

Programming: The act of writing to non-volatile memory, such as Flash. Configuration can occur directly after programming, or at a later time.

SRAM: Static Random Access Memory. A volatile storage array, generally used in FPGAs for configuration memory.

SVF File: Serial Vector Format file. A standard file format to describe IEEE1149.1 operations.

TransFR: Transparent Field Reconfiguration. A configuration method to allow minimization of system downtime.

VME: Diamond Embedded. Also refers to the data file used by the embedded source code.