



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





***Lattice*CORE™**

Interleaver/De-interleaver IP Core User's Guide

Chapter 1. Introduction	4
Quick Facts	4
Features	8
Chapter 2. Functional Description	10
Block Diagrams	10
Convolutional Interleaver/de-interleaver	11
Rectangular Interleaver/De-interleaver	12
Latency.....	13
Signal Descriptions	13
Timing Diagrams	16
Chapter 3. Parameter Settings	17
Type and Mode Tab	18
Type	18
Mode	18
Convolutional Parameters Tab.....	19
Interleaver/Deinterleaver.....	19
Rectangular Parameters Tab	20
Interleaver/Deinterleaver.....	20
Block Size Type	20
Permutations.....	21
Convolutional Optional Pins Tab.....	21
Rectangular Optional Pins Tab	22
Chapter 4. IP Core Generation.....	24
Licensing the IP Core.....	24
Getting Started.....	24
IPexpress-Created Files and Top Level Directory Structure.....	26
Permutation Pattern Input File Format	28
Instantiating the Core	28
Running Functional Simulation	28
Synthesizing and Implementing the Core in a Top-Level Design	29
Hardware Evaluation.....	30
Enabling Hardware Evaluation in Diamond.....	30
Enabling Hardware Evaluation in ispLEVER.....	30
Updating/Regenerating the IP Core.....	30
Regenerating an IP Core in Diamond	30
Regenerating an IP Core in ispLEVER	31
Chapter 5. Support Resources	32
Lattice Technical Support.....	32
Online Forums.....	32
Telephone Support Hotline	32
E-mail Support	32
Local Support	32
Internet.....	32
References.....	32
LatticeEC/ECP	32
LatticeECP2/M	32
LatticeECP3	32
LatticeSC/M.....	32
LatticeXP	33

LatticeXP2.....	33
Revision History	33
Appendix A. Resource Utilization	34
LatticeECP3 FPGAs.....	34
Ordering Part Number.....	34
LatticeECP and LatticeEC FPGAs	34
Ordering Part Number.....	34
LatticeECP2 Devices	35
Ordering Part Number.....	35
LatticeECP2M Devices	35
Ordering Part Number.....	35
LatticeXP Devices	35
Ordering Part Number.....	35
LatticeXP2 Devices	36
Ordering Part Number.....	36
LatticeSC/M Devices.....	36
Ordering Part Number.....	36

Interleaving is a technique commonly used in communication systems to overcome correlated channel noise such as burst error or fading. The interleaver rearranges input data such that consecutive data are spaced apart. At the receiver end, the interleaved data is arranged back into the original sequence by the de-interleaver. As a result of interleaving, correlated noise introduced in the transmission channel appears to be statistically independent at the receiver and thus allows better error correction.

The Lattice Interleaver/de-interleaver IP core supports rectangular block type and convolutional architectures. Rectangular interleaving arranges the input data row-wise in a matrix. The interleaved data is obtained by reading the columns of the matrix. Convolutional interleaving feeds the input data to a number of branches, each of which has a shift register with pre-defined length. The output data is taken from the branch outputs. Lattice’s Convolutional Interleaver/de-interleaver IP Cores are compliant with ATSC and DVB standards, while the Rectangular Interleaver/de-interleaver is compliant with IEEE 802.16a standard.

Quick Facts

Table 1-1 through Table 1-9 give quick facts about the Interleaver/de-interleaver IP core for LatticeEC™, LatticeECP™, LatticeECP2™, LatticeECP2M™, LatticeECP3™, LatticeSC™, LatticeSCM™, LatticeXP™, and LatticeXP2™ devices, respectively.

Table 1-1. Interleaver/De-interleaver IP Core for LatticeEC Devices Quick Facts

		Interleaver/de-interleaver IP Configuration			
		Convolutional Interleaver	Convolutional De-interleaver	Rectangular Interleaver	Rectangular De-interleaver
Core Requirements	FPGA Families Supported	LatticeEC			
	Minimal Device Needed	LFEC3E			
Resource Utilization	Targeted Device	LFEC20E-5F672C			
	LUTs	200	200	100	100
	sysMEM EBRs	2	2	4	4
	Registers	200			
Design Tool Support	Lattice Implementation	Lattice Diamond™ 1.0 or ispLEVER® 8.1			
	Synthesis	Synopsys® Synplify™ Pro for Lattice D-2009.12L-1			
	Simulation	Aldec® Active-HDL™ 8.2 Lattice Edition			
		Mentor Graphics ModelSim™ SE 6.3F			

Table 1-2. Interleaver/De-interleaver IP Core for LatticeECP Devices Quick Facts

		Interleaver/de-interleaver IP Configuration			
		Convolutional Interleaver	Convolutional De-interleaver	Rectangular Interleaver	Rectangular De-interleaver
Core Requirements	FPGA Families Supported	LatticeECP			
	Minimal Device Needed	LFECP6E			
Resource Utilization	Targeted Device	LFECP20E-5F672C			
	LUTs	200	200	100	100
	sysMEM EBRs	2	2	4	4
	Registers	200			
Design Tool Support	Lattice Implementation	Lattice Diamond 1.0 or ispLEVER 8.1			
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1			
	Simulation	Aldec Active-HDL 8.2 Lattice Edition			
		Mentor Graphics ModelSim SE 6.3F			

Table 1-3. Interleaver/De-interleaver IP Core for LatticeECP2 Devices Quick Facts

		Interleaver/de-interleaver IP Configuration			
		Convolutional Interleaver	Convolutional De-interleaver	Rectangular Interleaver	Rectangular De-interleaver
Core Requirements	FPGA Families Supported	LatticeECP2			
	Minimal Device Needed	LFE2-6E			
Resource Utilization	Targeted Device	LFE2-50E-7F672C			
	LUTs	200	200	100	100
	sysMEM EBRs	1	1	2	2
	Registers	200			
Design Tool Support	Lattice Implementation	Lattice Diamond 1.0 or ispLEVER 8.1			
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1			
	Simulation	Aldec Active-HDL 8.2 Lattice Edition			
		Mentor Graphics ModelSim SE 6.3F			

Table 1-4. Interleaver/De-interleaver IP Core for LatticeECP2M Devices Quick Facts

		Interleaver/de-interleaver IP Configuration			
		Convolutional Interleaver	Convolutional De-interleaver	Rectangular Interleaver	Rectangular De-interleaver
Core Requirements	FPGA Families Supported	LatticeECP2M			
	Minimal Device Needed	LFE2M20E			
Resource Utilization	Targeted Device	LFEC2M35E-7F672C			
	LUTs	200	200	100	100
	sysMEM EBRs	1	1	2	2
	Registers	200			
Design Tool Support	Lattice Implementation	Lattice Diamond 1.0 or ispLEVER 8.1			
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1			
	Simulation	Aldec Active-HDL 8.2 Lattice Edition			
		Mentor Graphics ModelSim SE 6.3F			

Table 1-5. Interleaver/De-interleaver IP Core for LatticeECP3 Devices Quick Facts

		Interleaver/De-interleaver IP Configuration			
		Convolutional Interleaver	Convolutional De-interleaver	Rectangular Interleaver	Rectangular De-interleaver
Core Requirements	FPGA Families Supported	Lattice ECP3			
	Minimal Device Needed	LFE3-17			
Resource Utilization	Targeted Device	LFSC3GA25E-7F900C			
	LUTs	150	150	100	100
	sysMEM EBRs	1	1	2	2
	Registers	200	200	150	150
Design Tool Support	Lattice Implementation	Lattice Diamond 1.0 or ispLEVER 8.1			
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1			
	Simulation	Aldec Active-HDL 8.2 Lattice Edition			
		Mentor Graphics ModelSim SE 6.3F			

Table 1-6. Interleaver/De-interleaver IP Core for LatticeSC Devices Quick Facts

		Interleaver/de-interleaver IP Configuration			
		Convolutional Interleaver	Convolutional De-interleaver	Rectangular Interleaver	Rectangular De-interleaver
Core Requirements	FPGA Families Supported	LatticeSC			
	Minimal Device Needed	LFSC3GA15E			
Resource Utilization	Targeted Device	LFSC3GA25E-7F900C			
	LUTs	200	200	100	100
	sysMEM EBRs	1	1	2	2
	Registers	200			
Design Tool Support	Lattice Implementation	Lattice Diamond 1.0 or ispLEVER 8.1			
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1			
	Simulation	Aldec Active-HDL 8.2 Lattice Edition Mentor Graphics ModelSim SE 6.3F			

Table 1-7. Interleaver/De-interleaver IP Core for LatticeSCM Devices Quick Facts

		Interleaver/de-interleaver IP Configuration			
		Convolutional Interleaver	Convolutional De-interleaver	Rectangular Interleaver	Rectangular De-interleaver
Core Requirements	FPGA Families Supported	LatticeSCM			
	Minimal Device Needed	LFSCM3GA15EP1			
Resource Utilization	Targeted Device	LFSCM3GA25EP1-7F900C			
	LUTs	200	200	100	100
	sysMEM EBRs	1	1	2	2
	Registers	200			
Design Tool Support	Lattice Implementation	Lattice Diamond 1.0 or ispLEVER 8.1			
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1			
	Simulation	Aldec Active-HDL 8.2 Lattice Edition Mentor Graphics ModelSim SE 6.3F			

Table 1-8. Interleaver/De-interleaver IP Core for LatticeXP Devices Quick Facts

		Interleaver/de-interleaver IP Configuration			
		Convolutional Interleaver	Convolutional De-interleaver	Rectangular Interleaver	Rectangular De-interleaver
Core Requirements	FPGA Families Supported	LatticeXP			
	Minimal Device Needed	LFXP3C			
Resource Utilization	Targeted Device	LFXP20C-5F484C			
	LUTs	200	200	100	100
	sysMEM EBRs	2	2	4	4
	Registers	200			
Design Tool Support	Lattice Implementation	Lattice Diamond 1.0 or ispLEVER 8.1			
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1			
	Simulation	Aldec Active-HDL 8.2 Lattice Edition Mentor Graphics ModelSim SE 6.3F			

Table 1-9. Interleaver/De-interleaver IP Core for LatticeXP2 Devices Quick Facts

		Interleaver/de-interleaver IP Configuration			
		Convolutional Interleaver	Convolutional De-interleaver	Rectangular Interleaver	Rectangular De-interleaver
Core Requirements	FPGA Families Supported	Lattice XP2			
	Minimal Device Needed	LFXP2-5E			
Resource Utilization	Targeted Device	LFXP2-30E-7FT256CES			
	LUTs	200	200	100	100
	sysMEM EBRs	1	1	2	2
	Registers	200			
Design Tool Support	Lattice Implementation	Lattice Diamond 1.0 or ispLEVER 8.1			
	Synthesis	Synopsys Synplify Pro for Lattice D-2009.12L-1			
	Simulation	Aldec Active-HDL 8.2 Lattice Edition Mentor Graphics ModelSim SE 6.3F			

Features

- High performance and area efficient symbol interleaver/de-interleaver
- Supports multiple standards, such as DVB, ATSC and IEEE 802.16
- Convolutional and rectangular block type architectures available
- Fully synchronous design using a single clock
- Symbol size from 1 to 256 bits
- Full handshake capability for input and output interfaces
- Rectangular block type features
 - Variable block size
 - Variable number of rows
 - Variable number of columns
 - Row permutations

- Column permutations
- Convolutional type features
 - User-configurable number of branches
 - User-configurable branch length

Functional Description

The functionality of the Convolutional and Rectangular Interleaver/de-interleaver cores are described in this chapter. [Figure 2-1](#) shows a convolutional interleaver/de-interleaver block diagram. [Figure 2-2](#) shows a rectangular interleaver/de-interleaver block diagram.

Block Diagrams

Figure 2-1. Convolutional Interleaver/De-interleaver Block Diagram

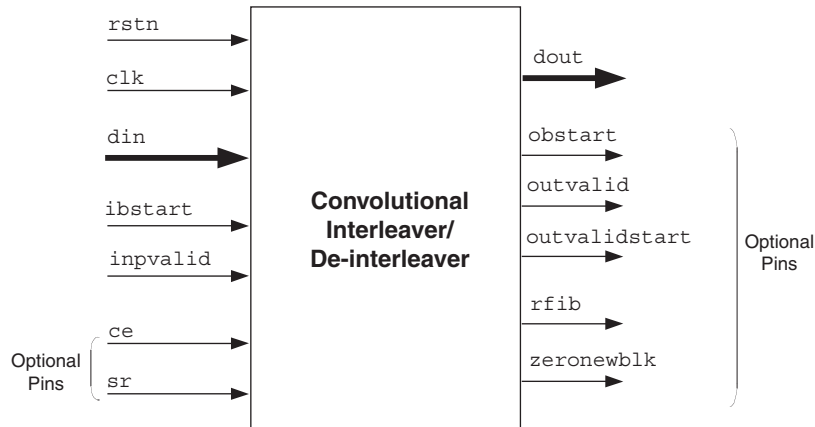


Figure 2-2. Rectangular Interleaver/De-interleaver Block Diagram

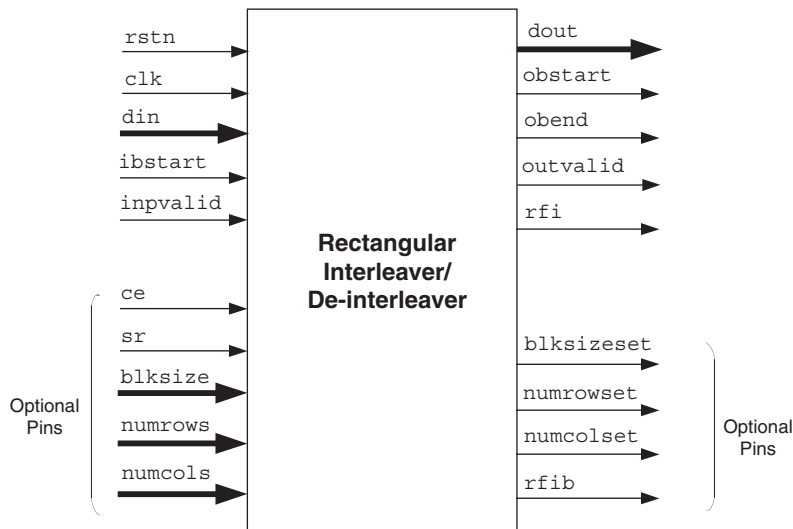
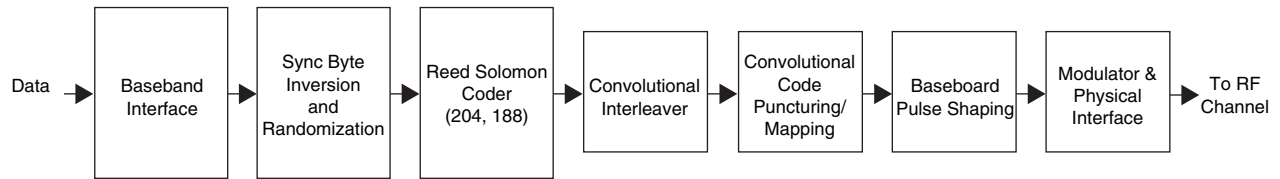


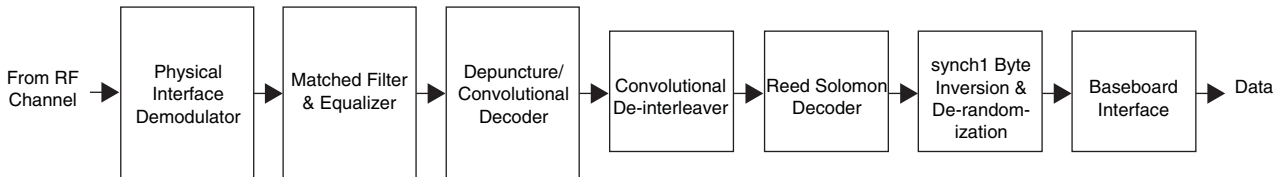
Figure 2-3 shows the role of interleaving and de-interleaving in a broadband wireless access system.

Figure 2-3. Broadband Wireless Access System

Base Station



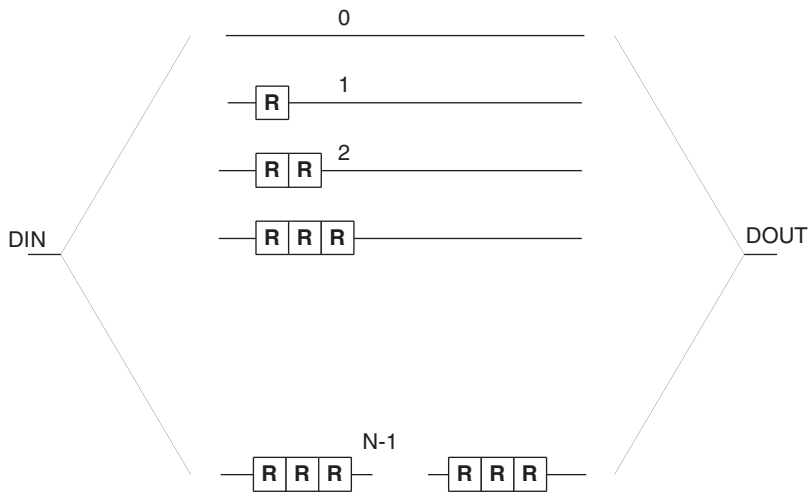
Subscriber Station



Convolutional Interleaver/de-interleaver

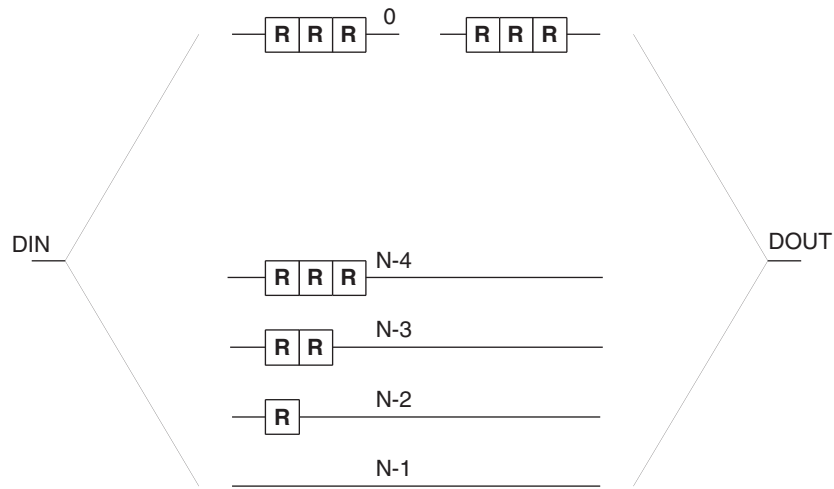
The convolutional interleaver consists of a bank of N branches. The value of N is set by the number of branches parameter. The interleaver pushes input data into the first shift register of a branch and reads the data from the output of the last register of the branch. The first branch has no delay, and every succeeding branch thereafter has a delay increase of R up to (N-1)*R, where R is the depth of the interleaver. The value of R is set by the branch_length parameter. The input and output are both connected to a commutator that synchronously rotates to each branch for each input symbol starting at branch zero. After switching to the final branch, the commutator rotates back to branch zero and repeats the process.

Figure 2-4. Convolutional Interleaver Functional Diagram



The de-interleaver is constructed similar to the interleaver, but its branches are arranged opposite to those of the interleaver. The first branch has a delay of (N-1)*R and the last branch has no delay.

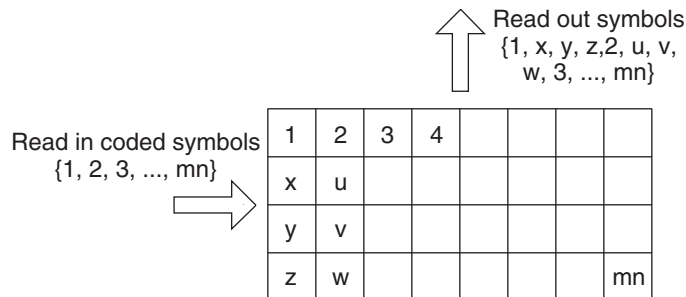
Figure 2-5. Convolutional De-Interleaver Functional Diagram



Rectangular Interleaver/De-interleaver

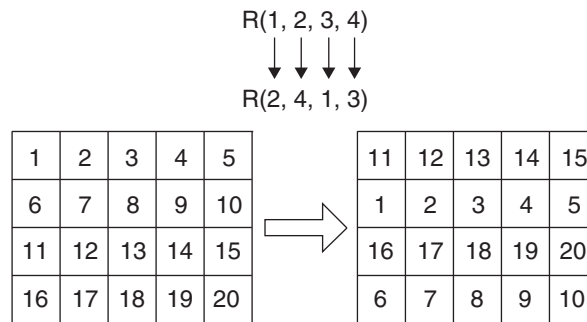
The rectangular interleaver receives the encoded data and formats it into a rectangular array of m rows and n columns. Typically, each row of the array composes a code word of length n . Symbols are read in row-wise and written out column-wise as indicated in Figure 2-6.

Figure 2-6. Rectangular Interleaver Matrix



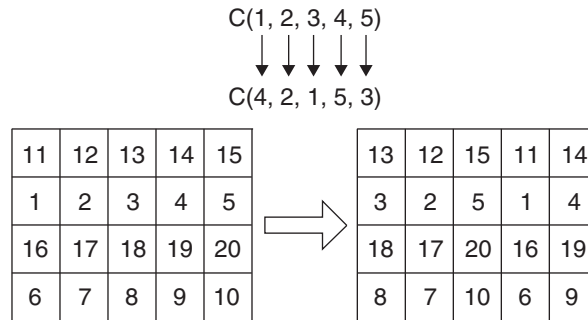
Inter-row and inter-column permutation formats may also be incorporated, but the matrix must be completely filled to do that. An example is shown in Figure 2-7 with a 4 x 5 array.

Figure 2-7. Row Permutation



The notation $R(2,4,1,3)$ for row permutation means row 1 is changed to row 2, row 2 to row 4, row 3 to row 1 and row 4 to row 3.

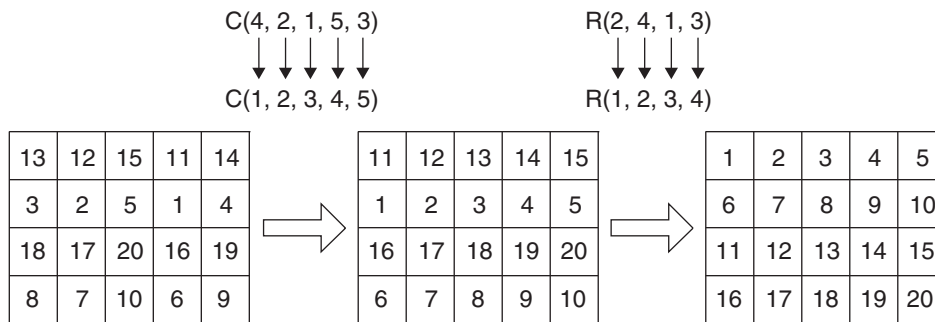
Figure 2-8. Column Permutation



The notation C(4,2,1,5,3) used to specify column permutation means column 1 is changed to column 4, column 2 to column 2, column 3 to column 1, column 4 to column 5 and column 5 to column 3.

Block de-interleaving is also a permutation formatting that is opposite of the permutation done for interleaving. This insures that the original data is correctly restored from the interleaver’s output data. De-interleaving of the last generated array from Figure 2-8 is illustrated in Figure 2-9.

Figure 2-9. De-interleaving



In the last array the rows are read out left-to-right and top-to-bottom to give the output data {1, 2, 3,...,20}. The block interleaver reads in one block of symbols and then outputs the same block with symbols rearranged. No new inputs can be read until the previous block’s interleaved symbols are output.

Latency

Latency for convolutional core, defined as the number of clock cycles between the sampling of the first input data and the availability of the first interleaved data at the output port, is six clock cycles.

Latency for rectangular core, defined as the number of clock cycles between the sampling of the last input data in the block and the availability of the first interleaved data at the output port, is five clock cycles.

Signal Descriptions

Table 2-1 and Table 2-2 list the input and output signals for the Interleaver/de-interleaver IP cores.

Table 2-1. Convolutional Interleaver/de-interleaver Signal Description

Port Name	I/O Type	Width	Signal Description
clk	Input	1	System Clock
rstn	Input	1	Asynchronous Reset. When this signal is asserted, all the core’s registers are cleared. This is an active low signal.

Table 2-1. Convolutional Interleaver/de-interleaver Signal Description (Continued)

Port Name	I/O Type	Width	Signal Description
ibstart	Input	1	This signal must be asserted when the first data is being presented on input. It must be asserted only if <code>r fib</code> is high or the current output stream is aborted. <code>ibstart</code> also initializes the commutator arms to branch zero. The signal <code>in pvalid</code> must be high for this signal to be effective.
in pvalid	Input	1	Qualifies input data <code>din</code> to be a valid new data symbol.
din	Input	1-256	Input Data. The input symbols present on this port are interleaved/de-interleaved.
dout	Output	1-256	Output Data. The interleaved/de-interleaved symbols are output on this port.
Optional Signals			
ce	Input	1	Clock Enable. This signal has the highest priority after <code>r stn</code> . The core operation freezes if this signal is low. Use of this port increases the size of the core. It should only be selected if necessary.
sr	Input	1	Synchronous Reset. When this signal is high, all the registers in the core are cleared synchronously. Use of this port increases the size of the core. It should only be selected if necessary.
obstart	Output	1	This signal indicates that the first interleaved data is present on the <code>dout</code> output port after an output latency of few cycles.
outvalid	Output	1	This signal indicates that a valid data is present on the <code>dout</code> output port. However, some of the output data may be from the initial values stored in the branch registers, as it takes a few cycles for the branch registers to get filled up with input data from the <code>din</code> port.
outvalidstart	Output	1	This signal indicates that a valid data is present on the <code>dout</code> output port and it corresponds to a valid data from the <code>din</code> port.
r fib	Output	1	This signal is asserted for one cycle as soon as the interleaver is ready to accept a new data. After <code>r fib</code> goes high, a new data stream and <code>ibstart</code> can be applied at the input.
zeronewblk	Output	1	This signal is available in the convolutional interleaver mode only. This signal is similar to <code>r fib</code> but with the following differences. The signal <code>zeronewblk</code> is only asserted when the input data source has filled all the storage elements in the interleaver. It indicates that the input source can again start feeding the data. This signal can be used for Block Boundary Synchronization. This signal is also asserted high at synchronous and asynchronous resets.

Table 2-2. Rectangular Interleaver /De-interleaver Signal Description

Port Name	I/O Type	Width	Signal Description
clk	Input	1	System Clock
r stn	Input	1	Asynchronous Reset. This is an active low signal. When this signal is asserted, all the registers in the core are cleared.
ibstart	Input	1	This signal signifies the first data on input <code>din</code> . It must be asserted only after <code>r fib</code> goes high. Otherwise the core will terminate the processing of the current block and start processing the new block. The signals <code>in pvalid</code> and <code>r fi</code> must be high for <code>ibstart</code> to be effective.
in pvalid	Input	1	Qualifies the input data <code>din</code> as a valid data symbol. If <code>r fi</code> is low, <code>in pvalid</code> will be ignored except for the case when the last input data symbol is placed at the <code>din</code> port.
din	Input	1-256	Input Data. The input symbols present on this port are interleaved/de-interleaved.
dout	Output	1-256	Output Data. The interleaved/de-interleaved symbols are output on this port.
obstart	Output	1	Assertion of this signal indicates that first data is present on the <code>dout</code> output port.

Table 2-2. Rectangular Interleaver /De-interleaver Signal Description (Continued)

Port Name	I/O Type	Width	Signal Description
outvalid	Output	1	Assertion of this signal indicates that valid data is present on the <code>dout</code> output port.
obend	Output	1	Assertion of this signal indicates that the last output data is present on the <code>dout</code> output port.
rfi	Output	1	Ready for Input. This signal is asserted when the core is ready to accept data. It is de-asserted one clock cycle before the last input data is read.
Optional Signals			
blksize	Input	4-16	Block size value is supplied through this port for variable block size configurations. The core reads the value on this port when <code>ibstart</code> is high.
numcols	Input	2-9	Number of columns in the current block. The number of columns value is provided through this port for a variable number of columns configuration. The core reads the value on this port when <code>ibstart</code> is high.
numrows	Input	3-9	Number of rows in current block. The number of rows value is provided through this port for a variable number of rows configuration. The core reads the value on this port when <code>ibstart</code> is high.
ce	Input	1	Clock Enable. This signal has the highest priority after <code>rstn</code> . The core operation freezes if this signal is low. This port increases the size of the core and should only be selected if necessary.
sr	Input	1	Synchronous Reset. When this signal is high, all the registers in the core are cleared synchronously. This port increases the size of the core and should only be selected if necessary.
numcolset	Output	1	This signal is high if the number of columns value provided on the <code>numcols</code> port is valid. If the value on <code>numcols</code> is not valid, <code>numcolset</code> is low. It is active three clock cycles after the number of columns value is sampled. If invalid, the signal <code>rfi</code> (and the optional <code>rfib</code> signal, if selected) will be asserted in the next clock cycle.
numrowset	Output	1	This signal is high if the number of rows value provided on the <code>numrows</code> port is valid. If the value on <code>numrows</code> is not valid, <code>numrowset</code> is low. It is active three clock cycles after the number of rows value is sampled. If invalid, the signal <code>rfi</code> (and the optional <code>rfib</code> signal, if selected) will be asserted in the next clock cycle.
blksizeet	Output	1	This signal is high if the block size value provided on <code>blksize</code> port is valid. If the block size value on <code>blksize</code> is not valid, <code>blksizeet</code> is low. It is active three clock cycles after the block size value is sampled. If invalid, the signal <code>rfi</code> (and the optional <code>rfib</code> signal, if selected) will be asserted in the next clock cycle.
rfib	Output	1	This signal is asserted for one cycle as soon as the interleaver is ready to accept a new data block. After <code>rfib</code> goes high, a new data block and <code>ibstart</code> can be applied at the input.

Timing Diagrams

Figure 2-10. Convolutional Interleaver Interface Timing

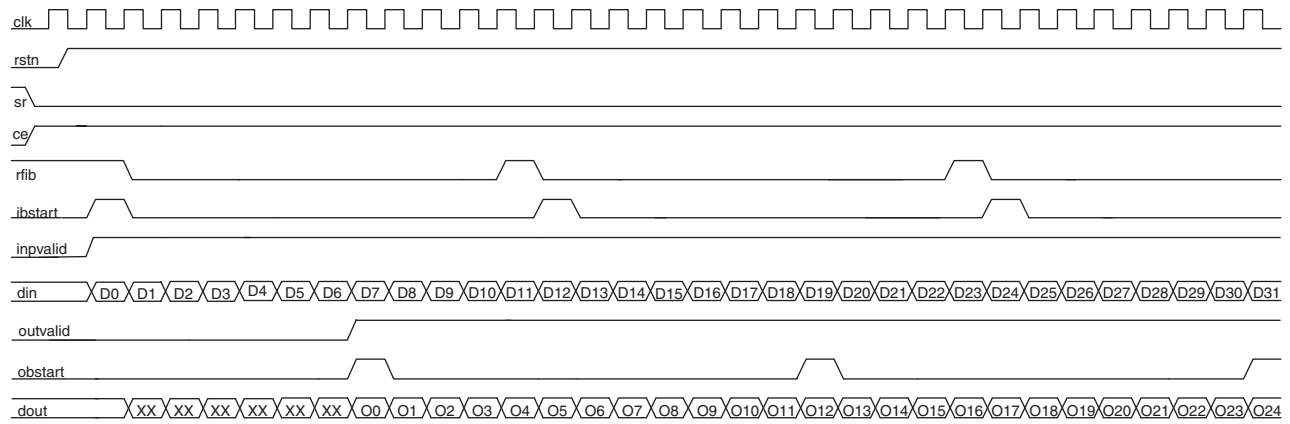
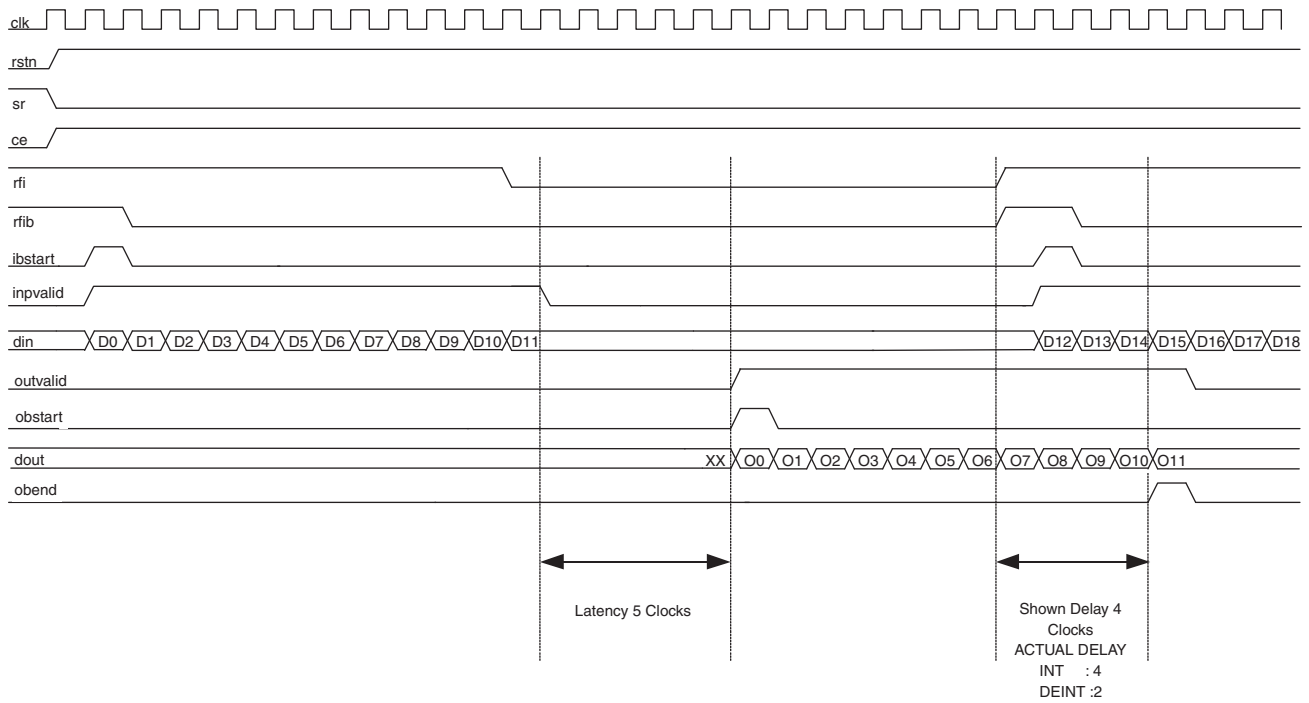


Figure 2-11. Rectangular Interleaver Interface Timing



Parameter Settings

The IPexpress™ tool is used to create IP and architectural modules in the Diamond or ispLEVER software. Refer to [“IP Core Generation” on page 24](#) for a description of how to generate the IP. The Interleaver/de-interleaver IP core can be customized to suit a specific application by adjusting parameters prior to core generation. Since the values of some parameters affect the size of the resultant core, the maximum value for these parameters may be limited by the size of the target device.

[Table 3-1](#) and [Table 3-2](#) provide the list of user configurable parameters for the Interleaver/de-interleaver IP core. The parameter settings are specified using the Interleaver/de-interleaver IP core Configuration GUI in IPexpress.

Table 3-1. Convolutional Interleaver/De-interleaver Parameter Descriptions

Parameter	Range/Options	Default
Type	Convolutional/Rectangular	
Mode	Interleaver/de-interleaver	Interleaver
Symbol Width	1-256	8
Number of Branches	ECP/EC: 7-256	12
Branch Length	1-780	17

Table 3-2. Rectangular Interleaver De-interleaver Parameter Descriptions

Parameter	Range/Options	Default
Mode	Interleaver or De-interleaver	Interleaver
Symbol Width	1-256	8
Block Size Type	Constant Row*Col Variable	Constant
Block Size	ECP/EC: 9-65536	4096
Number of Columns	2-256	256
Number of Rows	4-256	16
Row Permutations	Yes or No	No
Column Permutations	Yes or No	No
Row Type	Constant/Variable	Constant
Column Type	Constant/Variable	Constant
Row Width	3-9	5
Column Width	2-9	9
Block Width	ECP/EC: 4-16	13

Type and Mode Tab

Figure 3-1 shows the Type and Mode tab of the Interleaver/de-interleaver IP core configuration dialog box. The user can select Convolutional or Rectangular type and Interleaver or De-interleaver mode.

- If Convolutional type is selected, when the user clicks **Next**, the Convolutional tab displays.
- If Rectangular type is selected, when the user clicks **Next**, the Rectangular tab displays.

Figure 3-1. Type and Mode Tab



Type

This parameter allows the user to choose either Convolutional or Rectangular type.

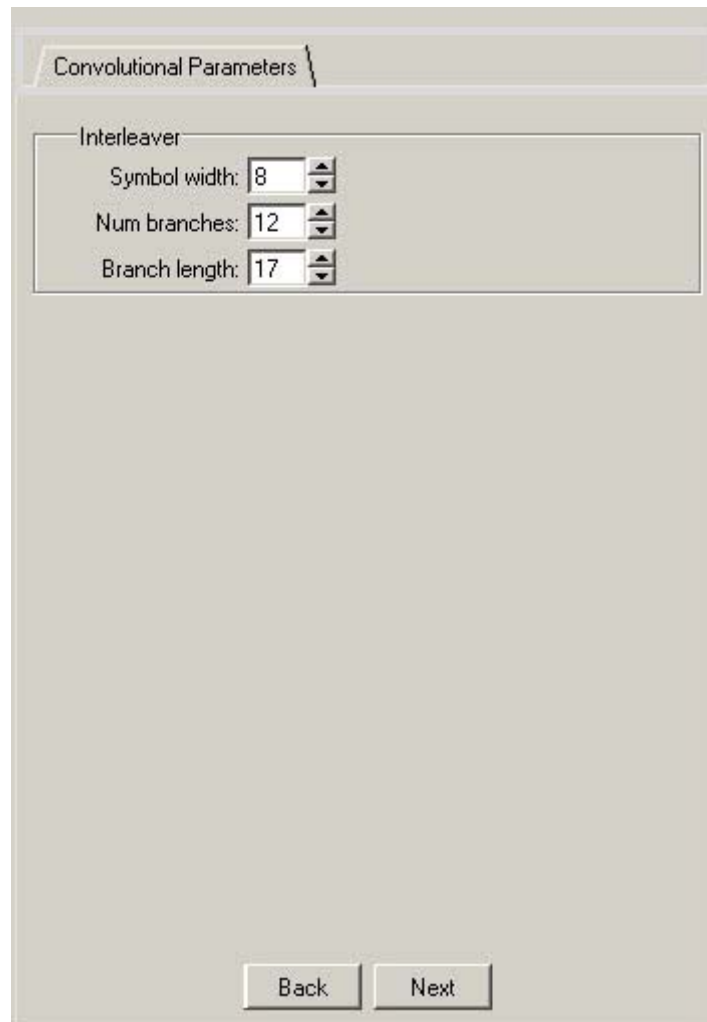
Mode

This parameter determines allows the user to choose either Interleaver or De-interleaver mode.

Convolutional Parameters Tab

If Convolutional was selected in the Type and Mode tab, the Convolutional Parameters tab is displayed when the **Next** button is clicked. [Figure 3-2](#) shows the Convolutional Parameters tab of the Interleaver/de-interleaver IP core configuration dialog box.

Figure 3-2. Convolutional Parameters Tab



Interleaver/Deinterleaver

The following options are available in the Convolutional Parameters tab for both Interleaver or De-interleaver modes.

Symbol Width

This options sets the data width of the input symbols.

Num Branches

This options sets the number of branches of the commutator arm.

Branch Lengths

This options sets the difference in storage elements for consecutive branches.

Rectangular Parameters Tab

If Rectangular was selected in the Type and Mode tab, the Rectangular Parameters tab is displayed when the **Next** button is clicked. [Figure 3-3](#) shows the Rectangular Parameters tab of the Interleaver/de-interleaver IP core configuration dialog box.

Figure 3-3. Convolutional Parameters Tab

The screenshot shows the 'Rectangular Parameters' configuration window. It is divided into several sections:

- Interleaver:** A 'Symbol width' spinner is set to 8.
- Block size type:** Three radio buttons are present: 'Constant' (selected), 'Row*Col', and 'Variable'.
- Block size constant parameters:** Three spinner controls are shown:
 - 'Num columns': 256 (range 2-256)
 - 'Num rows': 16 (range 4-256)
 - 'Block size': 4096 (range 3841 - 4096)
- Permutations:** Two sets of radio buttons:
 - 'Row Permutations': 'Yes' and 'No' (selected)
 - 'Column Permutations': 'Yes' and 'No' (selected)

At the bottom of the window are 'Back' and 'Next' buttons.

Interleaver/Deinterleaver

The following options are available in the Rectangular Parameters tab for both Interleaver or De-interleaver modes.

Symbol Width

This parameter specifies the data width of the input symbols.

Block Size Type

This parameter specifies the block size input to the core. There are three options for giving block size value.

- **Constant:** In this configuration the block size value is constant and assigned a value during core configuration.
- **Row*Col:** In this configuration, block size value is computed from the Row and Column values.
- **Variable:** In this configuration, block size value is given through an input port.

Block Size Constant Parameters

Block Size

This parameter is only available if Block Size Type = Constant. Block size input to the core: The block size value should be such that the last symbol in the block should come on the last row. Therefore block size value should be greater than $(Col*(Row-1))$ and less than or equal to $(Col*Row)$. When block size type is constant, number of rows and number of columns is also constant. Inter-row permutations and inter-column permutations are supported when block size value is $Col*Row$. Col is number of columns and Row is number of rows.

Number of Columns

This parameter specifies the number of columns used in the core. Inter-row permutations and inter-column permutations are supported when block size value is $Col*Row$.

Number of Rows

This parameter specifies the number of rows used in the core. Inter-row permutations and inter-column permutations are supported when block size value is $Col*Row$.

Permutations

Row Permutations

Rows can be permuted if required. Inter-row permutations and inter-column permutations are supported when block size value is $Col*Row$.

Column Permutations

Columns can be permuted if required. Inter-row permutations and inter-column permutations are supported when block size value is $Col*Row$.

Row Type

There are two options for giving number of rows.

- Constant: In this configuration, number of rows value is constant and assigned a value during core configuration.
- Variable: In this configuration, number of rows value is given through an input port. Row permutations are not supported when Row type is variable.

Column Type

There are two options for giving number of columns.

- Constant: In this configuration, number of columns value is constant and assigned a value during core configuration.
- Variable: In this configuration, number of columns value is given through the input port. Column permutations are not supported when Column type is variable.

Row Width

This parameter is only available if Row Type = Variable. Width for number of rows port (numrows).

Column Width

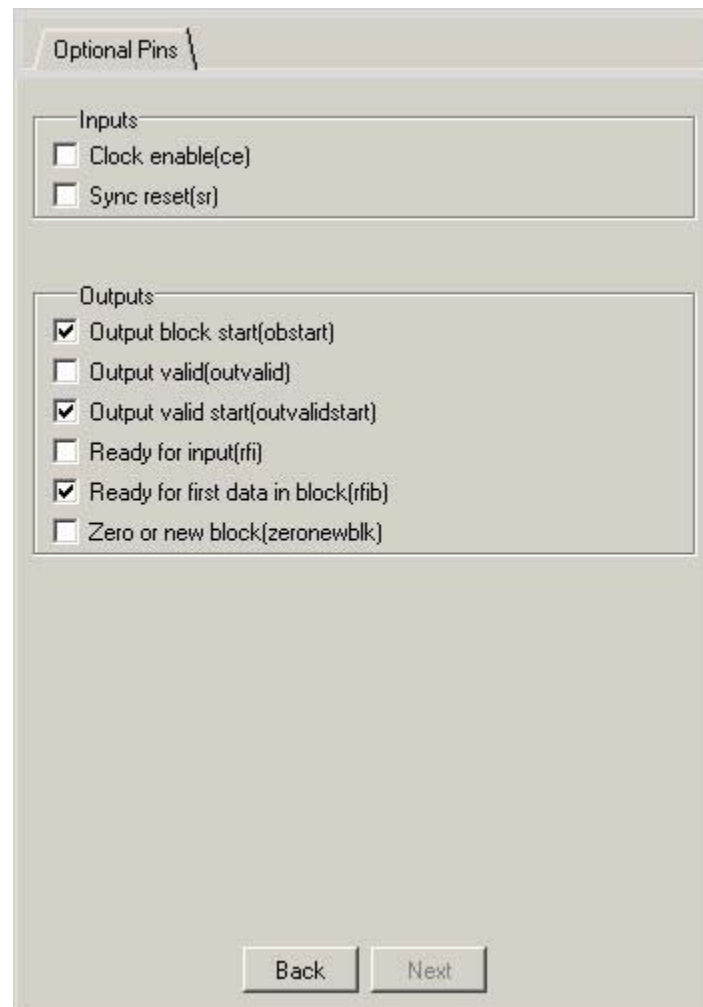
This parameter is only available if Column Type = Variable. Width for number of columns port (numcols)

Block Width

This parameter is only available if Block Size Type = Variable. Block Width required for the block size value. For variable block size type, either rows or columns or both can be selected to be variable. Both rows and columns cannot be constant. Row and column permutations are not supported when block size type is variable.

Convolutional Optional Pins Tab

If Convolutional was selected in the Type and Mode tab, the Convolutional Optional Pins tab is displayed when the **Next** button is clicked twice. [Figure 3-4](#) shows the Convolutional Parameters tab of the Interleaver/de-interleaver IP core configuration dialog box.

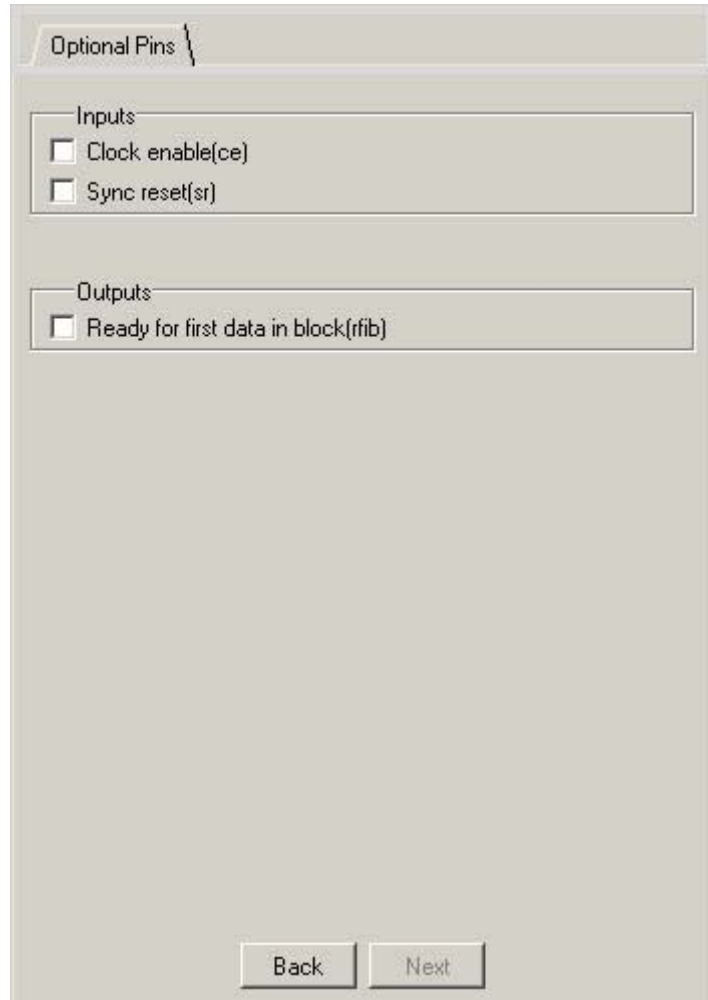
Figure 3-4. Convolutional Optional Pins Tab

In this tab, all ports are optional in convolutional mode, users can select or de-select them to interface with other modules.

Rectangular Optional Pins Tab

If Rectangular was selected in the Type and Mode tab, the Rectangular Optional Pins tab is displayed when the **Next** button is clicked twice. [Figure 3-5](#) shows the Convolutional Parameters tab of the Interleaver/de-interleaver IP core configuration dialog box.

Figure 3-5. Rectangular Optional Pins Tab



In this tab, all ports are optional in rectangular mode, users can select or de-select them to interface with other modules.

This chapter provides information on how to generate the Lattice Interleaver/de-interleaver IP core using the Diamond or ispLEVER software IPexpress tool, and how to include the core in a top-level design.

Licensing the IP Core

An IP core- and device-specific license is required to enable full, unrestricted use of the Interleaver/de-interleaver IP core in a complete, top-level design. Instructions on how to obtain licenses for Lattice IP cores are given at:

<http://www.latticesemi.com/products/intellectualproperty/aboutip/isplevercoreonlinepurchas.cfm>

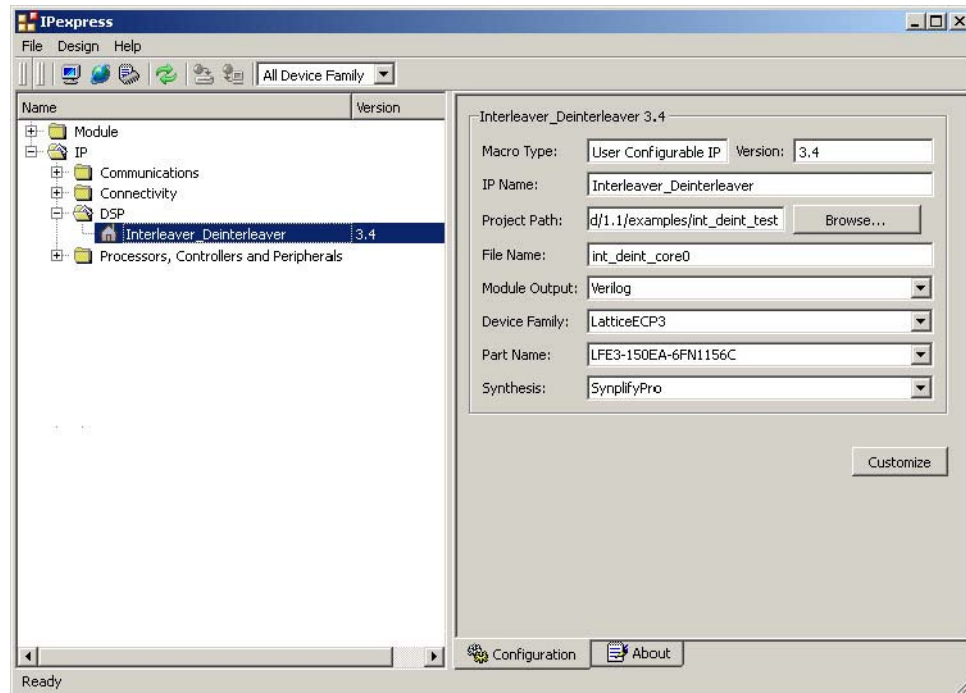
Users may download and generate the Interleaver/de-interleaver IP core and fully evaluate the core through functional simulation and implementation (synthesis, map, place and route) without an IP license. The Interleaver/de-interleaver IP core also supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core that operate in hardware for a limited time (approximately four hours) without requiring an IP license. See "To use this project file in Diamond:" on page 29 for further details. However, a license is required to enable timing simulation, to open the design in the Diamond or ispLEVER EPIC tool, and to generate bitstreams that do not include the hardware evaluation timeout limitation.

Getting Started

The Interleaver/de-interleaver IP core is available for download from the Lattice IP Server using the IPexpress tool. The IP files are automatically installed using ispUPDATE technology in any customer-specified directory. After the IP core has been installed, the IP core will be available in the IPexpress GUI dialog box shown in [Figure 4-1](#).

The IPexpress tool GUI dialog box for the Interleaver/de-interleaver IP core is shown in [Figure 4-1](#). To generate a specific IP core configuration the user specifies:

- **Project Path** – Path to the directory where the generated IP files will be loaded.
- **File Name** – "username" designation given to the generated IP core and corresponding folders and files.
- **(Diamond) Module Output** – Verilog or VHDL.
- **(ispLEVER) Design Entry Type** – Verilog HDL or VHDL
- **Device Family** – Device family to which IP is to be targeted (e.g. LatticeECP2M, LatticeECP3, etc.). Only families that support the particular IP core are listed.
- **Part Name** – Specific targeted part within the selected device family.

Figure 4-1. IPexpress Dialog Box (Diamond Version)

Note that if the IPexpress tool is called from within an existing project, Project Path, Module Output (Design Entry in ispLEVER), Device Family and Part Name default to the specified project parameters. Refer to the IPexpress tool online help for further information.

To create a custom configuration, the user clicks the **Customize** button in the IPexpress tool dialog box to display the Interleaver/de-interleaver IP core Configuration GUI, as shown in [Figure 4-2](#). From this dialog box, the user can select the IP parameter options specific to their application. Refer to [“Parameter Settings” on page 17](#) for more information on the Interleaver/de-interleaver IP core parameter settings.