

FIR Filter IP Core

User Guide

FPGA-IPUG-02043 Version 1.5

June 2018



Contents

Acronyms in This Document	5
1. Introduction	6
2. Quick Facts	7
3. Features	9
4. Functional Description	10
4.1. Interface Diagram	10
4.2. FIR Filter Architecture	10
4.2.1. Direct-form Implementation	10
4.2.2. Symmetric Implementation	11
4.2.3. Polyphase Interpolation FIR Filter	11
4.2.4. Polyphase Decimation FIR Filter	12
4.2.5. Multi-channel FIR Filters	12
4.3. Implementation Details	12
4.4. Configuring the FIR Filter Core	14
4.4.1. Architecture Options	14
4.4.1.1. Coefficients Specification	14
4.4.1.2. Multiplier Multiplexing Factor	15
4.4.2. I/O Specification Options	15
4.4.2.1. Rounding	15
4.4.3. Implementation Options	15
4.4.3.1. Memory Type	
4.5. Signal Descriptions	
4.6. Interfacing with the FIR Filter IP Core	
4.6.1. Data interface	
4.6.2. Multiple Channels	
4.6.3. Variable Interpolation/Decimation Eactor	
4 6 4 Reloadable Coefficients	
4.7 Timing Specifications	
4.7.1 Timing Specifications Applicable to All Devices	18
4.7.2 Timing Specifications Applicable to Lattice XP2 Lattice FCP3 and Lattice FCP5 Implementations	19
4.7.2. Timing Specifications Applicable to LatticeFCP3 and LatticeFCP5 Implementations	20
5 Parameter Settings	20
5.1 Architecture Tab	21
5.2 I/O Specification Tab	22
5.3 Implementation Tab	24
6 IP Core Generation and Evaluation	20
6.1 Licensing the IP Core	27
6.2 Gatting Startad	27
6.3 IPeynress-Created Files and Ton Level Directory Structure	27 21
6.1 Instantiating the Core	22
6.5 Running Eurotional Simulation	22
6.6 Synthesizing and Implementing the Core in a Top-Level Design	22
6.7 Hardware Evaluation	
6.7.1 Enabling Hardware Evaluation in Diamond	
6.7.1. Endoting /Reconcrating the ID Core	
0.0. Opuallig/negetielallig lie ir Cole	
0.0.1. Regenerating on ID Core in Clarity Designer Tabl	
0.3. Regenerating an ID Care in Clarity Designer Tool	
0.10. Recreating an in Core in Clarity Designer 1001	
Relefences	
rechnical Support Assistance	
Latticeeups devices	



LatticeXP2 Devices	37
ECP5 Devices	37
Revision History	38



Figures

Figure 4.1. Top-Level Interface for the FIR Filter IP Core	10
Figure 4.2. Direct-form FIR Filter	11
Figure 4.3. Symmetric Coefficients FIR Filter Implementation	11
Figure 4.4. Polyphase Interpolator	11
Figure 4.5. Polyphase Decimator	12
Figure 4.6. Functional Block Diagram	13
Figure 4.7. Tap and Coefficient Memory Management for a Sample FIR Filter	13
Figure 4.8. Single Channel, Single Rate FIR Filter with Continuous Inputs	18
Figure 4.9. Single Channel, Single Rate FIR Filter with Gaps in Input	18
Figure 4.10. Factorset Signals	18
Figure 4.11. Coefficient Reloading	18
Figure 4.12. Multi-Channel Single Rate FIR Filter (3 Channels)	19
Figure 4.13. Multi-Channel (3 Channels) Interpolator (Factor of 3)	19
Figure 4.14. Multi-Channel (3 Channels) Decimator (Factor of 3)	19
Figure 4.15. Multi-Channel Single Rate FIR Filter (3 Channels)	20
Figure 4.16. Multi-Channel (3 Channels) Interpolator (Factor of 3)	20
Figure 4.17. Multi-Channel (3 Channels) Decimator (Factor of 3)	20
Figure 5.1. Architecture Tab of the FIR Filter IP Core Interface	22
Figure 5.2. I/O Specification Tab of the FIR Filter IP Core Interface	24
Figure 5.3. Implementation Tab of the FIR Filter IP Core Interface	26
Figure 6.1. IPexpress Dialog Box	27
Figure 6.2. Configuration Dialog Box	28
Figure 6.3. Clarity Designer Tool Dialog Box	28
Figure 6.4. Clarity Designer Catalog Tab	29
Figure 6.5. Fir Filter Dialog Box	29
Figure 6.6. IP Configuration Interface	30
Figure 6.7. FIR Filter IP Core Generated Directory Structure	31

Tables

Table 2.1. FIR Filter IP Core for LatticeXP2 Devices Quick Facts	7
Table 2.2. FIR Filter IP Core for LatticeECP3 Devices Quick Facts	7
Table 2.3. FIR Filter IP Core for LatticeECP5 Devices Quick Facts	8
Table 4.1. Maximum Multiplier Multiplexing Factor for Different Configurations*	
Table 4.2. Top-Level Port Definitions	
Table 5.1. Parameter Specifications for the FIR Filter IP Core	21
Table 5.2. Architecture Tab	23
Table 5.3. I/O Specification Tab	25
Table 5.4. Implementation Tab	
Table 6.1. File List	
Table A.1. Performance and Resource Utilization (LatticeECP3)*	
Table A.2. Performance and Resource Utilization (LatticeXP2)*	
Table A.3. Performance and Resource Utilization (LFE5U)*	



Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
FIR	Finite Impulse Response
FPGA	Field-Programmable Gate Array
LED	light-emitting diode
MLE	Machine Learning Engine
SDHC	Secure Digital High Capacity
SDXC	Secure Digital eXtended Capacity
SPI	Serial Peripheral Interface
VIP	Video Interface Platform
USB	Universal Serial Bus
NN	Neuro Network



1. Introduction

The Lattice FIR (Finite Impulse Response) Filter IP core is a widely configurable, multi-channel FIR filter, implemented using high performance sysDSP[™] blocks available in Lattice devices. In addition to single rate filters, the IP core also supports a range of polyphase decimation and interpolation filters. The utilization versus throughput trade-off can be controlled by specifying the multiplier multiplexing factor used for implementing the filter. The FIR Filter IP core supports as high as 256 channels, with each having up to 2048 taps.

The input data, coefficient and output data widths are configurable over a wide range. The IP core uses full internal precision while allowing variable output precision with several choices for saturation and rounding. The coefficients of the filter can be specified at generation time and/or reloadable during run-time through input ports.

The FIR Filter IP core can also be generated using the Lattice FIR Filter Simulink[®] Model. For information on the Simulink flow, refer to the FPGA Design with ispLEVER tutorial.



2. Quick Facts

Table 2.1 through Table 2.3 provide quick facts about the FIR Filter IP core for LatticeXP2[™], LatticeECP3[™], and LatticeECP5[™] devices.

Table 2.1. FIR Filter IP Core for LatticeXP2 Devices Quick Facts

		FIR IP Configuration			
		1 Channels 64 Taps 16 Multipliers	1 Channel 24 Taps 6 Multipliers	1 Channel 48 Taps 12 Multipliers	
Core Requirements	FPGA Families Supported	LatticeXP2			
	Minimal Device Needed	LFXP2-5E	LFXP2-40E	LFXP2-8E	
Resource Utilization Targeted Device			LFXP2-40E-7F672C		
	LUTs	211	241	246	
	sysMEM EBRs	4	4	4	
	Registers	250	272	281	
	DSP Slice	1	1	1	
Design Tool Support	Lattice Implementation	Lattice Diamond 3.10			
	Synthesis	Synplify Pro F-2012.09L-SP1			
	Simulation	Modelsim SE 10.2c			
			Active-HDL 8.2 Lattice Editio	n	

Table 2.2. FIR Filter IP Core for LatticeECP3 Devices Quick Facts

		FIR IP Configuration		
		4 Channels 64 Taps 1 Multiplier	1 Channel 32 Taps 32 Multipliers	1 Channel 32 Taps 8 Multipliers
Core Requirements	FPGA Families Supported	LatticeECP3		
	Minimal Device Needed		LFE3-35EA	
Resource Utilization	Targeted Device	LFE3-150EA-6FN672C		
	LUTs	866	212	200
	sysMEM EBRs	32	2	4
	Registers	2041	199	303
	MULT18X18	64	4	6
Design Tool Support	Lattice Implementation	Lattice Diamond 3.10		
	Synthesis	Synplify Pro F-2012.09L-SP1		
	Simulation	Modelsim SE 10.2c		
		A	ctive-HDL 8.2 Lattice Edition	n



Table 2.3. FIR Filter IP Core for LatticeECP5 Devices Quick Facts

		FIR IP Configuration		
	-	4 Channels 64 Taps 1 Multiplier	1 Channel 32 Taps 32 Multipliers	1 Channel 32 Taps 8 Multipliers
Core Requirements	FPGA Families Supported	ECP5		
	Minimal Device Needed	LFE5UM-85FEA	LFE5UM-85FEA	LFE5UM-85FEA
Resource Utilization	Targeted Device	LFE5U-85F-6BG756C		
	LUTs	248	202	201
	sysMEM EBRs	2	2	4
	Registers	222	199	303
	DSP Slice	6	6	9
Design Tool Support	Lattice Implementation	ation Lattice Diamond 3.10		
	Synthesis		Synplify Pro F-2012.09L-SP1	L
	Simulation	Alde	c Active-HDL 10.3 Lattice Ec	lition
			ModelSim SE 10.2c	



3. Features

- Variable number of taps up to 2048
- Input and coefficients widths of 4 to 32 bits
- Multi-channel support for up to 256 channels
- Decimation and Interpolation ratios from 2 to 256
- Support for half-band filter
- Configurable parallelism from fully parallel to serial
- Signed or unsigned data and coefficients
- Coefficients symmetry and negative symmetry optimization
- Re-loadable coefficients support
- Full precision arithmetic
- Selectable output width and precision
- Selectable overflow: wrap-around or saturation
- Selectable rounding: truncation, round towards zero, round away from zero, round to nearest and convergent rounding
- Width and precision specified using fixed point notations
- Handshake signals to facilitate smooth interfacing



4. Functional Description

This chapter provides a functional description of the FIR Filter IP core.

4.1. Interface Diagram

The top-level interface diagram for the FIR Filter IP core is shown in Figure 4.1.



CONTROL SIGNALS

Figure 4.1. Top-Level Interface for the FIR Filter IP Core

4.2. FIR Filter Architecture

FIR filter operation on data samples can be described as a sum-of-products operation. For an N-tap FIR filter, the current input sample and (N-1) previous input samples are multiplied by N filter coefficients and the resulting N products are added to give one output sample as shown below.

$$y_n = \sum_{i=0}^{N-1} x_{n-i} h_i = x_n h_0 + x_{n-i} h_1 + \dots + x_{n-N+i} h_{N-i}$$
(1)

In the above equation, h_n , n=0,1,...,N-1 is the impulse response; x_n , $n=0,1,...,\infty$ is the input; and y_n , $n=0,1,...,\infty$ is the output. The number of delay elements (N-1) represents the order of the filter. The number of input data samples (current and previous) used in the calculation of one output sample represents the number of filter taps (N).

4.2.1. Direct-form Implementation

In the direct-form implementation shown in Figure 4.2, the input samples will be shifted into a shift register queue and each shift register is connected to a multiplier. The products from the multipliers are summed to get the FIR filter's output sample.





Figure 4.2. Direct-form FIR Filter

4.2.2. Symmetric Implementation

The impulse response for most FIR filters is symmetric. This symmetry can generally be exploited to reduce the arithmetic requirements and produce area-efficient filter realizations. It is possible to use only one half of the multipliers for symmetric coefficients compared to that used for a similar filter with non-symmetric coefficients. An implementation for symmetric coefficients is shown in Figure 4.3.



Figure 4.3. Symmetric Coefficients FIR Filter Implementation

4.2.3. Polyphase Interpolation FIR Filter

The polyphase interpolation filter option implements the computationally efficient 1-to-P interpolation filter shown below, where P is an integer greater than 1. Figure 4.4 shows a polyphase interpolator, where each branch is referred to as a polyphase.







In this structure, the input data will be loaded into each polyphase at the same time and the output data of each polyphase will be unloaded as an output sample of the FIR. The number of polyphases is equal to the interpolation factor. The coefficients are assigned to all polyphases evenly.

4.2.4. Polyphase Decimation FIR Filter

The polyphase decimation filter option implements the computationally efficient P-to-1 decimation filter shown in Figure 4.5, where P is an integer greater than 1.



Figure 4.5. Polyphase Decimator

In this structure, the input sample is loaded sequentially into each of the polyphases with only one polyphase fed at a time. When all the polyphases are loaded with a sample, the result from the polyphases are summed and unloaded as the FIR filter's output. In this scheme, P input samples generate one output sample, where P is the decimation factor.

4.2.5. Multi-channel FIR Filters

It is very common to see FIR filters used in multi-channel processing scenarios. The maximum possible throughput of a FIR filter implementation is often much higher than the throughput required for a single channel being processed. For such applications, it is desirable to use the same resources in a time multiplexed way to realize multi- channel FIR filters. Except in fully parallel implementations, where enough multipliers are used to perform all the necessary computations in one clock cycle, the FIR filter uses independent tap and coefficient memories to feed each multiplier. Hence, multi-channel implementations result in lower memory usage compared to multiple instantiations of FIR filters. For cases, where all the channels use the same coefficient set, using a multi-channel FIR filter has the clear advantage of requiring smaller coefficient memories.

4.3. Implementation Details

Figure 4.6 shows the functional block diagram of the FIR Filter IP core.





Figure 4.6. Functional Block Diagram

The data and coefficients are stored in different memories shown as tap memory and coefficients memory in the above diagram. The symmetry adder is used if the coefficients are symmetric. The multiplier array contains one or more multipliers depending on the user specification. The adder tree performs the sum of products. Depending on the configuration, the adder tree, or a part of it, is implemented inside DSP blocks. The output processing block performs the output width reduction and precision control. This block contains logic to support different types of rounding and overflow. The block labeled Control Logic manages the scheduling of data and arithmetic operations based on the type of filter (interpolation, decimation or multi-channel) and multiplier multiplexing.

The tap and coefficient memories are managed differently for different configurations of the FIR filter. Figure 4.7 shows the memory assignments for a 16-tap, 3-channel, symmetric FIR filter with two multipliers.



Figure 4.7. Tap and Coefficient Memory Management for a Sample FIR Filter

In the diagram, there are two tap memories and a coefficient memory for each multiplier. The depth of each memory is ceil(taps/2/multiplier) *channel, which is 12 in this example, where the operator ceil(x) returns the next higher integer, if the argument x is fractional.

^{© 2008-2018} Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.



4.4. Configuring the FIR Filter Core

4.4.1. Architecture Options

The options for number of channels, number of taps, and filter type are independent and directly specified in the **Architecture** tab of the IP core interface (see Parameter Settings for details). If a polyphase decimator or interpolator is required, the decimation or interpolation factor can be directly specified in the interface. The decimation or interpolation factor can also be specified through input ports during operation by selecting the corresponding **Variable** option. If the **Variable decimation** (or **Variable interpolation**) factor option is selected, the decimation (or interpolation) factor can be varied from two to **Decimation factor** (or **Interpolation factor**) through the input port.

4.4.1.1. Coefficients Specification

The coefficients of the filter are specified using a coefficients file. The coefficients file is a text file with one coefficient per line. If the coefficients are symmetric, the check box Symmetric Coefficients must be checked so the IP core uses symmetry adders to reduce the number of multipliers used. If the Symmetric Coefficients box is checked, only one-half of the coefficients are read from the coefficient file. For an n-tap symmetric coefficients filter, the number of coefficients read from the coefficients file is equal to ceil(n/2). For multi-channel filters, the coefficients for channel 0 are specified first, followed by those for channel 1, and so on. For multi-channel filters, there is an option to specify whether the coefficients are different for each channel or the same (common) for all the channels.

If the coefficients are common, only one set of coefficients needs to be specified in the coefficients file. The coefficient values in the file can be in any radix (decimal, hexadecimal or binary) selected by the user. A unary negative operator is used only if the coefficients are specified in decimal radix. For hexadecimal and binary radices, the numbers must be represented in twos complement form. An example coefficients file in decimal format for an 11-tap, 16-bit coefficients set is given below. In this example, the coefficients binary point is 0.

-556 -706 -857 -419 1424 5309 11275 18547 25649 30848 32758

An example coefficients file in floating point format for the above case when the Coefficients binary point position is 8, is given below. The coefficients will be quantized to conform to the 16.8 fractional data in which 16 is the full width of coefficients, and 8 is the width of fractional part.

-2.1/19
-2.7578
-3.3477
-1.6367
5.5625
20.7383
44.043
72.45
100.0191
120.5
127.96

2 1710

© 2008-2018 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.



If the check box **Reloadable Coefficients** is checked, the coefficients can be reloaded to the FIR filter during the operation of the core. With this option, the desired coefficients must be loaded before the operation of the filter. The coefficients must be loaded in a specific order that is determined by the program supplied with the IP core. The IP core can also optionally do the reordering internally, albeit using more resources. If this option is desired, the check box **Reorder Coefficients** Inside can be checked. With this option, the coefficients can be loaded in the normal sequential order to the core.

4.4.1.2. Multiplier Multiplexing Factor

The throughput and the resource utilization can be controlled by assigning a proper value to the **Multiplier Multiplexing Factor** parameter. Full parallel operation (one output data per clock cycle) can be achieved by setting the **Multiplier Multiplexing Factor** to 1. If the **Multiplier Multiplexing Factor** is set to the maximum value displayed in the interface, full series operation is supported and it takes up to n clocks to compute one output data sample, where n is the number of taps for a non-symmetric FIR filter and half the number of taps for a symmetric FIR filter. The maximum value of the **Multiplier Multiplexing Factor** for different configurations of an n-tap FIR filter is given in Table 4.1.

FIR Type	Single Rate	Interpolator with Factor=i	Decimator with Factor
Non-symmetric	n	Ceil(n/i)	Ceil(n/d)
Symmetric	Ceil(n/2)	Ceil(n/2i)	Ceil(n/2d)
Half-band	floor((n+1)/4)+1	floor((n+1)/4)	floor((n+1)/8)+1

Table 4.1. Maximum Multiplier Multiplexing Factor for Different Configurations*

*Note: The operator floor (x) returns the next lower integer, if x is a fractional value.

4.4.2. I/O Specification Options

The controls in the I/O Specifications interface tab are used to define the various widths and precision methods in the data path. The width and binary point positions of the input data and coefficients can be defined independently. From the input data width, coefficient width and the number of taps, the full precision output width and true location of the output binary point automatically get fixed. The full precision output is converted to user specified output width by dropping some least significant (LS) and some most significant (MS) bits and by performing the specified rounding and overflow processing. The output is specified by the output width and the output binary point position parameter.

4.4.2.1. Rounding

The following five options are supported for rounding:

- None Discards all bits to the right of the output least significant bit and leaves the output uncorrected.
- **Rounding up** Rounds to nearest more positive number.
- Rounding away from zero Rounds away from zero if the fractional part is exactly one-half.
- Rounding towards zero Rounds towards zero if the fractional part is exactly one-half.
- Convergent rounding Rounds to the nearest even value if the fractional part is exactly one-half.

4.4.3. Implementation Options

4.4.3.1. Memory Type

The FIR Filter IP core uses memories for storing delay tap data, coefficients and for some configurations, input or output data. The number of memory units used depends on several parameters including data width, number of taps, filter type, number of channels and coefficient symmetry. In most cases, each multiplier requires one data memory unit and one coefficient memory unit. Interpolation or decimation filters may additionally use input or output buffers. The memory type interface option can be used to specify whether EBR or distributed memory is used for data, coefficient, input and output storage. The option called Auto leaves that choice to the IP generator tool, which uses EBR if the memory is deeper than 128 locations and distributed memory otherwise.



4.5. Signal Descriptions

A description of the Input/Output (I/O) ports for the FIR Filter IP core is provided in Table 4.2.

Table 4.2. Top-Level Port Definitions

Port	Bits	I/O	Description
General I/O			
clk	1	I	System clock for data and control inputs and outputs.
rstn	1	I	System wide asynchronous active-low reset signal.
din	Input data width	I	Input data.
inpvalid	1	I	Input valid signal. The input data is read-in only when inpvalid is high.
dout	Output width	0	Output data.
outvalid	1	0	Output data qualifier. Output data dout is valid only when this signal is high.
rfi	1	0	Ready for input. This output, when high, indicates that the IP core is ready to receive the next input data. A valid data may be applied at din only if rfi was high during the previous clock cycle.
When Reloadable co	oefficients is selected		
coeffin	Notes 1*	I	Coefficients input. The coefficients have to be loaded through this port in a specific order. Refer to the section Interfacing with the FIR Filter IP core for details.
coeffwe	1	I	When asserted, the value on bus coeffin will be written into coefficient memories.
coeffset	1	I	This input is used to signal the filter to use the recently loaded coefficient set. This signal must be pulsed high for one clock cycle after the loading the entire coefficient set using coeffin and coeffwe.
When Number of ch	annels is greater than 1		
ibstart	1	Ι	Input block start. For multi-channel configurations, this input identifies channel 0 of the input.
obstart	1	0	Output block start. For multi-channel configurations, this output identifies channel 0.
When Variable inter	polation factor or Variab	le decimation factor is	s checked
ifactor	ceil(Log2(Interpolation factor+1))	I	Interpolation factor value
dfactor	ceil(Log2(Decimation factor+1))	Ι	Decimation factor value
factorset	1	I	Sets the interpolation factor or the decimation factor.
Optional I/Os			
се	1	I	Clock Enable. While this signal is de-asserted, the core will ignore all other synchronous inputs and maintain its current state
sr	1	I	Synchronous Reset. When asserted for at least one clock cycle, all the registers in the IP core are initialized to reset state.

Notes:

1. Width for signed type and symmetric interpolation is Coefficients width +1.

2. Width for unsigned and symmetric interpolation is Coefficients width +2.

3. Width for all other cases is Coefficients width.



4.6. Interfacing with the FIR Filter IP Core

4.6.1. Data interface

Data is fed into the core through din and out from the core through dout.

4.6.2. Multiple Channels

For multi-channel implementations, two ports, ibstart and obstart, are available in the IP core to synchronize the channel numbers. The input ibstart is used to identify channel 0 data applied at the inputs. The output obstart goes high simultaneously with channel 0 output data.

4.6.3. Variable Interpolation/Decimation Factor

When the interpolation (or decimation) factor is variable, the ports ifactor (or dfactor) and factorset are added to the IP core. The interpolation (or decimation) factor applied on the port ifactor (or dfactor) is set when the strobe signal factorset is high. When the interpolation (or decimation) factor changes, the output rfi goes low for a few cycles. When it becomes high again, the filter performs as an interpolating (or decimating) filter corresponding to the new factor value.

4.6.4. Reloadable Coefficients

When **Reloadable Coefficients** is selected, the two added ports, coeffin and coeffwe, are used to reload the coefficients. All the coefficients need to be loaded in one batch, while keeping the signal coeffwe high during the entire duration of loading. After all the coefficients are loaded, the input signal coeffset must be pulsed high for one clock cycle for the new coefficients to take effect.

There are two ways in which coefficients can be applied for reloading the coefficients memory, as specified by the **Reorder Coefficients Inside** parameter.

When **Reorder Coefficients Inside** is not selected, the coefficients have to be applied in a particular sequence for reloading the coefficients memory. The raw coefficients, as specified in the coefficients file, can be converted to the reloadable sequence by using the coefficients generation program coeff_gen.exe (for Windows) available under the gui folder in the IP installation directory (for example, under the **C:\LatticeCore\fir_core_v6.0\gui** folder). The names of the coefficient generation program for UNIX and Linux are coeff_gen_s and coeff_gen_l respectively. For Windows, the program is invoked as follows:

coeff_gen.exe <IP_file_name>.lpc

Note: If in lpc file, the value of parameter varcoeff= is Yes, please change it to No before generating ROM files manually.

This command converts the coefficients in the input file, as referred by the coefffile= parameter in the lpc file, to the loadable coefficients sequence file called coeff.mem. Note that the output file may contain more coefficients than there originally were due to inserted zero coefficients. All the coefficients in the output file, including the zeros, have to be applied sequentially through the coeffin port. To obtain the sequence of application of coefficients, edit the input coefficients file with sequential numbers 1,2,... and run the coeff_gen.exe command. In the reloadable coefficients mode, the core will not be ready for operation (the rfi output will not be high) until the coefficients are loaded and coeffset is asserted high.

When the parameter **Reorder Coefficients Inside** is selected, the coefficients will be reordered inside the IP core without requiring manual reordering described previously. With this option, reordering logic is added to the IP core and the user can apply the coefficients in the normal sequence.

In this mode, if the parameter **Symmetric Coefficients** is selected, only half of the coefficients provided will be used. For example, if the raw coefficient input sequence is: 1 2 3 4 5 6 5 4 3 2 1, the coefficients that will be used will be 1 2 3 4 5 6.

Similarly, if **Half Band** is selected, all of the input coefficients in the even locations, except the last one, will be discarded. For example, if the raw coefficient input sequence is: 1 0 2 0 3 0 4 0 5 6 5 0 4 0 3 0 2 0 1, the coefficients that will be used will be 1 2 3 4 5 6.

Note: If the parameter varcoeff= in the lpc file is set to Yes, change it to No before generating the new coefficients file.



4.7. Timing Specifications

Timing diagrams for the FIR Filter IP core are given in Figure 4.8 through Figure 4.17. Note that there are different timing specifications for certain FIR filter applications using Lattice XP2/ECP3/ECP5 devices. Figure 4.8 through Figure 4.11 apply to all FIR applications.

clk	
rfi	
inpvalid	
din	$eq:dl_dl_dl_dl_dl_dl_dl_dl_dl_dl_dl_dl_dl_d$
outvalid	
dout	D0 / D1 / D2 / D3 / D4 / D5 / D6 / D7 / D8 / D9 / D10 / D11 / D12 / D13 / D14 / D15 /
	Figure 4.8. Single Channel, Single Rate FIR Filter with Continuous Inputs
clk	
rfi	
inpvalid	
din	d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11
outvalid	
dout	D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D9
	Figure 4.9. Single Channel, Single Rate FIR Filter with Gaps in Input
clk	
factorset	
lfactor/ dfactor	X 3 X
	Figure 4.10. Factorset Signals
clk	
coeffwe	
coeffin	$\begin{array}{ $
coeffset	
	Figure 4.11. Coefficient Reloading

4.7.1. Timing Specifications Applicable to All Devices



4.7.2. Timing Specifications Applicable to LatticeXP2, LatticeECP3 and LatticeECP5 Implementations

In addition to the previous figures, Figure 4.12 through Figure 4.14 apply in using both LatticeXP2, LatticeECP3, and LatticeECP5 devices: negative symmetry, half band, factor variable interpolation and decimation, and applications using 36x36 multipliers.







4.7.3. Timing Specifications Applicable to LatticeECP3 and LatticeECP5 Implementations

As indicated previously, Figure 4.15 through Figure 4.17 apply to all LatticeECP3 and Lattice ECP5 devices other than those specifically listed in the previous section.



^{© 2008-2018} Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.



5. Parameter Settings

The IPexpress and Clarity Designer tools are used to create IP and architectural modules in the Diamond software. You may refer to the IP Core Generation and Evaluation section on how to generate the IP.

Table 5.1 provides the list of user configurable parameters for the FIR Filter IP core. The parameter settings arespecified using the FIR Filter IP core Configuration interface in IPexpress or Clarity Designer. The numerous FIR Filter IPcore parameter options are partitioned across multiple interface tabs as described in this chapter.

Table 5.1. Parameter Specifications for the FIR Filter IP Core

Parameter	Range	Default				
Filter Specifications						
Number of channels	1 to 256	4				
Number of taps	1 to 2048	64				
Filter type	{Single rate, Interpolator, Decimator}	Single rate				
Interpolation factor	2 to 256	2				
Variable interpolation factor	{Yes, No}	No				
Decimation factor	2 to 256	2				
Variable decimation factor	{Yes, No}	No				
Coefficients Specifications						
Reloadable coefficients	{Yes, No}	Yes				
Reorder coefficients inside	{Yes, No}	No				
coefficients set	{Common, One per channel}	Common				
Symmetric coefficients	{Yes, No}	No				
Negative symmetry	{Yes, No}	No				
Half band	{Yes, No}	No				
Coefficient radix	{Floating point, Decimal, Hex, Binary}	Decimal				
Coefficients file	Type or Browse	-				
Advanced Options						
Multiplier Multiplexing factor	Note 1, Note 2	Note 2				
Number of SysDSP blocks in a row	5 - Note 3	Note 3				
I/O Specifications						
Input data type	{Signed, Unsigned}	Signed				
Input data width	4 to 32	16				
Input data binary point position	-2 to Input data width + 2	0				
Coefficients type	{Signed, Unsigned}	Signed				
Coefficients width	4 to 32	16				
Coefficients binary point position	-2 to Coefficients width + 2	0				
Output width	4 to Max Output Width	38				
Output binary point position	(4+Input data binary point position + coefficient binary point position – Max output width) to (Output width + Input data binary point position + Coefficient binary point position - 4)	0				
Precision control						
Overflow	{Saturation, Wrap-around}	Saturation				
Rounding	{None, Round-up, Round away from zero, Round towards zero, Convergent rounding}	None				
Memory Type	Memory Type					
Data memory type	{EBR, Distributed, Auto}	EBR				
Coefficient memory type	{EBR, Distributed, Auto}	EBR				
Input buffer type	{EBR, Distributed, Auto}	EBR				

© 2008-2018 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.

All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.



Parameter	Range	Default
Output buffer type	{EBR, Distributed, Auto}	EBR
Optimization	{Area, Speed}	{Area}
Optional Ports		
ce	{Yes, No}	No
sr	{Yes, No}	No
Synthesis Options		
Frequency constraint	1-400	300

Notes:

1. The Multiplier Multiplexing Factor is limited by the number of DSP blocks in a device (A) and the actual number of DSP blocks a design needs (B). When A>B, the Multiplier Multiplexing Factor is set to 1; otherwise the value will be greater than 1.

- 2. See Multiplier Multiplexing Factor for details.
- 3. Maximum number of DSP blocks available in a row in the selected device.

The default values shown in the following pages are those used for the FIR Filter reference design. IP core options for each tab are discussed in further detail.

5.1. Architecture Tab

Figure 5.1 shows the contents of the Architecture tab.

Architecture \ 1/0 Specification \ Implementation \						
Filter specifications						
Number of channels 4 (1-256)	Number of taps 64	(1-2048)				
Filter type						
Filter type Single rate 💌						
Variable interpolation factor	Interpolation factor	(2-256)				
Variable decimation factor	Decimation factor	(2-256)				
Coefficients set Common C Sep Symmetric coefficients	parate ative symmetry 🔲 Half B	and				
Coefficients radix Decimal						
Coefficients file						
Advanced Options						
Multiplier Multiplexing Factor	64	(1-64)				
Number of SysDSP Blocks in a Row	16	(5-16)				

Figure 5.1. Architecture Tab of the FIR Filter IP Core Interface



Table 5.2. Architecture Tab

Interface Item	Description
Number of Channels	This option allows the user to specify the number of channels.
Number of Taps	This option allows the user to specify the number of taps.
Filter Type	This option allows the user to specify whether the filter is single rate, interpolator, or decimator.
Interpolation Factor	This option allows the user to specify the value of the fixed interpolation factor. When FIR type is interpolation, the value should be 2 to 256. Otherwise, it will be set to 1 automatically.
Variable Interpolation Factor	This option allows the user to specify whether the interpolation factor is fixed at the time of IP generation, or variable during run-time. If this is checked, the interpolation factor is set through the input port ifactor when factorset is high.
Decimation Factor	This option allows the user to specify the value of the fixed decimation factor. When FIR type is decimation, the value should be 2 to 256. Otherwise, it will be set to 1 automatically.
Variable Decimation Factor	This option allows the user to specify whether the decimation factor is fixed at the time of IP generation or variable during run-time. If this is checked, the decimation factor is set through the input port dfactor when factorset is high.
Reloadable Coefficients	This option allows the user to specify whether the coefficients are fixed or reloadable. If checked, the coefficients can be reloaded during core operation using the input port coeffin.
Reorder Coefficients Inside	When coefficients are reloadable, they need to be entered in a particular order. The reordering can be done using the program supplied along with the IP core. However, the core also provides for optional hardware reordering at the expense of additional hardware resources. If this option is selected, the coefficients can be entered in the normal sequence to the core, and the core will internally reorder hem as required. This option is not available when Filter type is interpolator, and Symmetric coefficients is enabled.
Coefficients set	This option allows the user to specify whether the same coefficient set is used for all channels, or an independent coefficient set is used for each channel.
Symmetric Coefficients	This option allows the user to specify whether the coefficients are symmetric. If this is checked, only one half of the number of coefficients (if number of taps is odd, the half value is rounded to the next higher integer) is read from the initialization file.
Negative Symmetry	If this is checked, the coefficients are considered to be negative symmetric. That is the second half of the coefficients are made equal to the negative of the corresponding first-half coefficients.
Half Band	This option allows the user to specify whether a half band filter is realized. If this is checked, only one half of the number of coefficients (if the number of taps is odd, the half value is rounded to the next higher integer) is read from the initialization file.
Coefficient Radix	This option allows the user to specify the radix for the coefficients in the coefficients file. For decimal radix, the negative values have a preceding unary minus sign. For hexadecimal (Hex) and binary radices, the negative values must be written in 2's complement form using exactly as many digits as specified by the coefficients width parameter. The floating point coefficients are specified in the form <nnn>.<ddd>, where the digits 'n' denote the integer part and the digits 'd', the decimal part. The values of the floating point coefficients must be consistent with the Coefficients width and Coefficients binary point position parameters. For example, if <nnn>.<ddd> is 8.4 and Coefficients type is unsigned, the value of the coefficients should be between 0 and 1111111.1111 (255.9375).</ddd></nnn></ddd></nnn>



Interface Item	Description
Coefficients File	This option allows the user to specify the name and location of the coefficients file. If the coefficients file is not specified, the filter is initialized with a default coefficient set.
Multiplier Multiplexing Factor	This option allows the user to specify the Multiplier Multiplexing Factor. This parameter should be set to 1 for full parallel applications and to the maximum value supported in the interface for full series applications.
Number of sysDSP Blocks in a Row	This parameter allows the user to specify the maximum number of DSP multipliers to be use in a DSP row to achieve optimal performance. For example, if the targeted device has 20 multipliers in a DSP row and the design requires 22 multipliers, the user can select to use all 20 multipliers in one row and two multipliers in another row, or fewer than 20 multipliers in each row (e.g. 8), which may yield better performance. Multipliers spread across a maximum of three DSP rows may be used in a single FIR instance. This parameter is only valid on LatticeECP3 and ECP5 devices.

5.2. I/O Specification Tab

Figure 5.2 shows the contents of the I/O Specification tab.

Data	
Input data type	Signed
Input data width	16 🗨
Input data binary point position	0
Coefficients	
Coefficients type	Signed
Coefficients width	16 🗨
Coefficients binary point position	
Output	
	Full precision
Output width	38 💌 38
Output binary point position	0 🗸 0

Figure 5.2. I/O Specification Tab of the FIR Filter IP Core Interface

© 2008-2018 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.



Table 5.3. I/O Specification Tab

Interface Item	Description
Input Data Type	This option allows the user to specify the input data type as signed or unsigned.
Input Data Width	This option allows the user to specify input data width.
Input Data Binary Point Position	This option allows the user to specify the location of the binary point in the input data. This number specifies the bit position of the binary point from the LSB of the input data. If the number is zero, the point is right after LSB, if positive, it is to the left of LSB and if negative, it is to the right of LSB.
Coefficients Type	This option allows the user to specify the coefficients type as signed or unsigned. If the type is signed, the coefficient data is interpreted as a 2's complement number.
Coefficients Width	This option allows the user to specify the coefficients width.
Coefficients Binary Point Position	This option allows the user to specify the location of the binary point in the coefficients. This number specifies the bit position of the binary point from the LSB of the coefficients. If the number is zero, the point is right after LSB; if positive, it is to the left of LSB and if negative, it is to the right of LSB.
Output Width	This option allows the user to specify the output data width. The maximum full precision output width is defined by Max Output Width = Input data width + Coefficients width +ceil (Log2(Number of taps/Interpolation factor)). The core's output is usually a part of the full precision output equal to the Output width and extracted based on the different binary point position parameters. The format for the internal full precision output is displayed as static text next to the Output width control in the interface. The format is displayed as W.F, where W is the full precision output, counted to the left. For example, if W.F is 16.4, then the output value will be yyyyyyyyyyyyyyy in binary radix.For example, 110010010010.
Output Binary Points	This option allows the user to specify the bit position of the binary point from the LSB of the actual core output. If the number is zero, the point is right after LSB, if positive, it is to the left of LSB and if negative, it is to the right of LSB. This number, together with the parameter Output width, determines how the actual core output is extracted from the true full precision output. The precision control parameters Overflow and Rounding are applied respectively when MSBs and LSBs are discarded from the true full precision output.
Overflow	This option allows the user to specify what kind of overflow control is to be used. This parameter is available when- ever there is a need to drop some of the MSBs from the true output. If the selection is Saturation, the output value is clipped to the maximum, if positive or minimum, if negative, while discarding the MSBs. If the selection is Wrap- around, the MSBs are simply discarded without making any correction.
Rounding	This option allows the user to specify the rounding method when there is a need to drop one or more LSBs from the true output.



5.3. Implementation Tab

Figure 5.3 shows the contents of the Implementation tab.

Memory type			
Data memory type	EBR	-	
Coefficient memory type	EBR	-	
Input buffer type	EBR	-	
Output buffer type	EBR	_	
Synchronous reset (sr)			
Optimization			
🖲 Area			
Speed			
Synthesis options			
Francisco and the state of the	200 (1	400)	

Figure 5.3. Implementation Tab of the FIR Filter IP Core Interface

Table 5.4. Implementation Tab

Interface Item	Description
Data Memory Type	This option allows the user to specify select the type of memory that is used for storing the data. If the selection is EBR, Lattice Embedded Block RAM memories are used for storing the data. If the selection is Distributed, look- up-table based distributed memories are used for storing data. If "Auto" is selected, EBR memories are used for memory sizes deeper than 128 locations and distributed memories are used for all other memories. If the type is signed, the data is interpreted as a two's complement number.
Coefficient Memory Type	This option allows the user to specify the type of memory that is used for storing the coefficients. If the selection is EBR, EBR memories are used for storing the coefficients. If the selection is Distributed, distributed memories are used for storing coefficients. If Auto is selected, EBR memories are used for memory sizes deeper than 128 locations and distributed memories are used for all other memories.
Input Buffer Type	This option allows the user to specify the memory type for the input buffer.
Output Buffer Type	This option allows the user to specify the memory type for the output buffer.
Synchronous Reset (sr)	This option allows the user to specify if a synchronous reset port is needed in the IP. Synchronous reset signal resets all the registers in the FIR filter IP core.
Clock Enable (ce)	This option allows the user to specify if a clock enable port is needed in the IP. Clock enable control can be used for power saving when the core is not being used. Use of clock enable port increases the resource utilization and may affect the performance due to the increased routing congestion.
Optimization	This option specifies the optimization method. If Area is selected, the core is optimized for lower resource utilization. If Speed is selected, the core is optimized for higher performance, but with slightly higher resource utilization.
Synthesis Options	Lattice LSE or Synplify Pro



6. IP Core Generation and Evaluation

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

6.1. Licensing the IP Core

An IP core- and device-specific license is required to enable full, unrestricted use of the FIR Filter 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 FIR Filter IP core and fully evaluate the core through functional simulation and implementation (synthesis, map, place and route) without an IP license. The FIR Filter 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 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.

6.2. Getting Started

The FIR Filter IP core is available for download from Lattice's IP server using the IPexpress or the Clarity Designer 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 Interface or the Clarity Designer tool.

The IPexpress tool interface dialog box for the FIR Filter IP core is shown in Figure 6.1. To generate a specific IP core configuration, the user specifies:

- Project Path Path to the directory where the generated IP files will be located.
- File Name Username designation given to the generated IP core and corresponding folders and files.
- (Diamond) Module Output Verilog or VHDL.
- **Device Family** Device family to which IP is to be targeted (such as LatticeXP2, LatticeECP3, and others). Only families that support the particular IP core are listed.
- Part Name Specific targeted part within the selected device family.

Name	Version		FIR Filter 6.0	
P Multiply_Accumulate	2.8		Macro Type:	Ilser Configurable IP Version: 6.0
Image: A state of the state	1.6		macro rype.	
尋 Subtractor	3.5		IP Name:	FJR Filter
✓ [™] DSP_Modules			Project Path:	ocuments/Diamond/xp2_svnplify Browse
ADDER_TREE	1.4		,	
🐺 BARREL_SHIFTER	1.3		File Name:	
碶 MAC	4.2		Module Output:	Verilog •
WMAC	2.2		Dovico Family:	Lattice/D2
🚇 MULT	4.2		Device runniy.	
寧 MULTADDSUB	4.2		Part Name:	LFXP2-40E-5F484C
寧 MULTADDSUBSUM	4.2		Synthesis:	SynplifyPro
🐺 SLICE	1.6			
🐺 WIDE_MUX	1.4		Please enter a file	name!
✓ [™] Memory_Modules				Customize
✓				
Distributed_DPRAM	3.9			
I Distributed_ROM	2.8			
Distributed_SPRAM	3.9			
✓				
RAM_DP	6.5			
RAM_DP_TRUE	7.5			
👜 RAM_DQ	7.5			
题 ROM	5.4			
寧 FIFO	5.1			
孁 FIFO_DC	5.8			
RAM_Based_Shift_Register	5.2			
V 🔮 IP				
🗸 🚔 DSP				
😰 FIR Filter	5.1			
IP FIR Filter	6.0	~	🍓 Configuration	📑 About

Figure 6.1. IPexpress Dialog Box



Note that if the IPexpress tool is called from within an existing project, Project Path, Module Output, 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 FIR Filter IP core Configuration interface, as shown in Figure 6.2. From this dialog box, the user can select the IP parameter options specific to their application. Refer to Parameter Settings for more information on the FIR Filer IP core parameter settings.

FIR Filter		Architecture \ I/O Specification \ Implementation \	
→ clk		Filter specifications Number of channels 4 (1-256) Number of taps 64	(1-2048
→ rstn		Filter type Single rate	
		Variable interpolation factor Interpolation factor 1	(2-256)
	dout[37:0]	Variable decimation factor Decimation factor 1	(2-256)
➡ din[15:0]			
→inpvalid	outvalid 🔸	Coefficients specifications	
→inp∨alid →ibstart	outvalid 🔶	Coefficients specifications	
→inpvalid →ibstart	outvalid 🔸 obstart 🔸	Coefficients specifications Reloadable coefficients Reorder coefficients inside Coefficients set Common C Separate	
→inp∨alid →ibstart	outvalid	Coefficients specifications Reloadable coefficients Reorder coefficients inside Coefficients set Common C Separate Symmetric coefficients Regative symmetry Relf	band
→ inp∨alid → ibstart	outvalid → obstart →	Coefficients specifications Reloadable coefficients Reorder coefficients inside Coefficients set Common C Separate Symmetric coefficients Regative symmetry Half Coefficients radix Decimal	band
→inpvalid →ibstart	outvalid → obstart →	Coefficients specifications Reloadable coefficients Reorder coefficients inside Coefficients set Symmetric coefficients Negative symmetry Half Coefficients radix Decimal	band
→inpvalid →ibstart	outvalid	Coefficients specifications Reloadable coefficients Reorder coefficients inside Coefficients set Symmetric coefficients Regative symmetry Half Coefficients radix Decimal Advanced Options	band
→ inpvalid → ibstart	outvalid	Coefficients specifications Reloadable coefficients Reloadable coefficients Coefficients set Coefficients set Coefficients radix Decimal Coefficients file Advanced Options Multiplier Multiplexing Factor	band
→ inpvalid → ibstart	outvalid → obstart → rfi →	Coefficients specifications Reloadable coefficients Symmetric coefficients Coefficients radix Coefficients radix Coefficients file Advanced Options Multiplier Multiplexing Factor Number of SysDSP Blocks in a Row	band 64
→ inpvalid → ibstert	outvalid → obstart → rfi →	Coefficients specifications Reloadable coefficients Symmetric coefficients Coefficients radix Coefficients radix Coefficients file Advanced Options Multiplier Multiplexing Factor Number of SysDSP Blocks in a Row	band
⇒inpvalid ⇒ibstart	outvalid → obstart → rfi →	Coefficients specifications Reloadable coefficients Symmetric coefficients Coefficients radix Coefficients radix Coefficients file Advanced Options Multiplier Multiplexing Factor Number of SysDSP Blocks in a Row	bend
→inpvalid → ibstart	outvalid → obstart → rfi →	Coefficients specifications Reloadable coefficients Symmetric coefficients Coefficients radix Coefficients radix Coefficients file Advanced Options Multiplier Multiplexing Factor Number of SysDSP Blocks in a Row	band 64 16
→ inpvalid → ibstart	outvalid → obstart → rfi →	Coefficients specifications Reloadable coefficients Symmetric coefficients Coefficients radix Coefficients radix Coefficients file Advanced Options Multiplier Multiplexing Factor Number of SysDSP Blocks in a Row	band 64 16

Figure 6.2. Configuration Dialog Box

The Clarity Designer tool interface dialog box for the FIR Filter IP core is shown in Figure 6.3.

- Create new Clarity design Choose to create a new Clarity Design project directory in which the FIR IP core will be generated.
- **Design Location** Clarity Design project directory Path.
- **Design Name** Clarity Design project name.
- HDL Output Hardware Description Language Output Format (Verilog or VHDL).
- Open Clarity design Open an existing Clarity Design project.
- Design File Name of existing Clarity Design project file with .sbx extension.

🔠 Clarity Designer		? ×				
Create new Clari	ty design					
Design Location:	C:\sc_env\proj\fir_imp	Browse				
Design Name:						
HDL Output:	Verilog	•				
Open Clarity desi	gn					
Design File: C:/lsc_	env/proj/test/test.sbx	Browse				
Start Clarity Desi	Start Clarity Designer to generate a single Component SBX					
Clarity Designer will module on Clarity De IP generation is not	L start clarity besigner to generate a single component SBX Clarity Designer will enter single module generation mode. In this mode, you can double-dick the module on Clarity Designer catalog to start generating a single module SBX. IP generation is not available in this mode, please do not enable this mode for IP. generation					

Figure 6.3. Clarity Designer Tool Dialog Box



The Clarity Designer Catalog tab is shown in Figure 6.4. To generate FIR IP core configuration, double-click the IP name in the Catalog tab.

Catalog Builder Planner		
Name	Version ^	
Y 🚔 Arithmetic Modules		FIK Filter
Fort adder	3.5	
adder subtractor	3.5	Davides Summark Lattice VD2 Lattice CD2 Lattice CD5111 attice CD51111 Lattice CD5111160
comparator	3.6	Device Support LaureeLOF3, LaureeLOF30, LaureeLOF30M, LaureeLOF30M30
complex multiplier	1.8	
convert	1.3	Overview. Finite impulse Response (First) inter is a common type of inter that provides functionality to remove noise, provide spectral shaping, or perform signal detection in digital
counter	4.6	systems. A FIR filter performs a convolution of an input data sequence with the filter's impulse response (the discrete time inverse Fourier transform of its desired frequency response),
fft_butterfly	1.5	which is stored in memory. FIR filter has the advantages of being inherently stable and having linear phase characteristics. I hese characteristics have led to FIR filters being designed
A Ifsr	3.7	into a large number of applications.
🚑 mult_add_sub	2.8	
🔤 mult_add_sub_sum	2.7	The Lattice FIR (finite impulse response) Filter IP core is a widely configurable, multi-channel FIR filter, implemented using high performance sysDSP blocks available in Lattice
multiplier	4.9	devices. In addition to single rate filters, the IP core also supports a range of polyphase decimation and interpolation filters. The utilization vs. throughput trade-off can be controlled by
multiply_accumulate	2.8	specifying the multiplier multiplexing factor used for implementing the filter. Using the largest multiplexing factor results in the best resource utilization, whereas using the
sin-cos_table	1.6	smallest multiplier multiplexing factor results in the best throughput. The FIR Filter IP supports as high as 256 channels, with each having as high as 2048 taps.
subtractor	3.5	
✓ ^{SP} DSP_Modules		The input data, coefficient and output data widths are configurable over a wide range. The IP core uses full internal precision while allowing the output precision to be defined and
1d_filter	1.2	supports several choices for saturation and rounding. The coefficients of the filter can be specified at generation time and/or reloadable during run-time through input ports.
adder_tree	1.4	
barrel_shifter	1.3	Features:
eff mac	4.2	Variable number of taps up to 2048
4 mmac	2.2	 Input and coefficients widths of 4 to 32 bits
eff mult	4.2	Multi-channel support for up to 256 channels
multaddsub	4.2	Decimation and Internolation ratios from 2 to 256
multaddsubsum	4.2	Configurable parallelism from full to citation in 2 to 200
But slice	1.0	Comparison parameters in norm to single mainplies Single acues parameters in the mainplies
age wide_mux	1.4	
 Wemory_Woodules M. Bistributed RAM 		• Coencient symmetry opunization
Gittibuted doram	2.0	Re-loadable coellicients support
at distributed rom	20	Full precision antimetic
Get distributed coram	3.0	Selectable output width and precision
✓ Set EBR Components		Selectable overflow: wrap-around or saturation
and ram do	6.5	Selectable rounding: truncation, round towards zero, round away from zero, round to nearest and convergent rounding
ram dp true	7.5	Width and precision specified using fixed point notations
ram_dq	7.5	Handshake signals to facilitate smooth interfacing
a rom	5.4	In LatticeECP5U/ LatticeECP5UM/ LatticeECP5UM5G, support high-speed. For lowspeed, support half-band filter
e fifo	5.1	
fifo_dc	5.8	
📴 ram_based_shift_register	r 5.2	Version: 6.0
🗸 🚳 IP		
Y 🖄 DSP		Website: FIR Filter
🕼 fir filter	6.0	
	· · · · · · · · · · · ·	×
Lattice IP 🐨 Lattice IP Server	Import IP	E About

Figure 6.4. Clarity Designer Catalog Tab

In the Fir Filter dialog box shown in Figure 6.5, specify the following:

• Instance Name – The instance module name of FIR IP core.

👂 fir filter				?		×
Name						
Instance Path:	C:/lsc_env/proj/test					
Instance Name:		_				
fir filter 6.0						
Macro Type:	IPCFG Version	: 6	i.0			
Maco Name:	fir filter					
Module Output:	Verilog					•
Device Family:	ECP5UM5G					
Part Name:	LFE5UM5G-85F-8BG756C					
Synthesis:	lse					
Please enter a inst	ance name!					
		Cust	tomize		Cancel	

Figure 6.5. Fir Filter Dialog Box

Note that if the Clarity Designer tool is called from within an existing project, Design Location, Device Family, and Part Name default to the specified project parameters. Refer to the Clarity Designer tool online help for further information.

To create a custom configuration, click the Customize button in the Clarity Designer tool dialog box to display the FIR IP core Configuration interface, as shown in Figure 6.6. From this dialog box, the user can select the IP parameter options specific to their application. Refer to Parameter Settings for more information on the FIR parameter settings.

FIR Filter IP Core User Guide



Hattice IP Core FIR Filter v6.0	- 🗆 X
Configuration Generate Log	
FIR Filter	Architecture \ I/O Specification \ Implementation \
→clk →rstn dout[37:0]→	Filter specifications Number of channels 4 (1-256) Number of taps 64 (1-2048) Filter type Filter type Single rate ✓ ✓ variable interpolation factor Interpolation factor 1 (2-256) ✓ variable decimation factor Decimation factor 1 (2-256)
→ din[15:0] → inpvalid outvalid → ibstart obstart → rfi →	Coefficients specifications Reloadable coefficients Coefficients set Coefficients set Symmetric coefficients Negative symmetry Half band Coefficients radix Decimal Coefficients file Advanced Options Multiplier Multiplexing Factor 64 Number of SysDSP Blocks in a Row
	Configure Close Help

Figure 6.6. IP Configuration Interface



6.3. IPexpress-Created Files and Top Level Directory Structure

When the user clicks the **Generate** button, the IP core and supporting files are generated in the specified Project Path directory. The directory structure of the generated files is shown in Figure 6.7.



Figure 6.7. FIR Filter IP Core Generated Directory Structure

The design flow for IP created with the IPexpress tool uses a post-synthesized module (NGO) for synthesis and a protected model for simulation. The post-synthesized module is customized and created during the IPexpress tool generation.

Table 6.1 provides a list of key files created by the IPexpress tool. The names of most of the created files are customized to the user's module name specified in the IPexpress tool. The files shown in Table 6.1 are all of the files necessary to implement and verify the FIR Filter IP core in a top-level design.

File	Description
<username>_inst.v</username>	This file provides an instance template for the IP.
<username>.v</username>	This file provides a wrapper for the FIR core for simulation.
<username>_beh.v</username>	This file provides a behavioral simulation model for the FIR core.
<username>_bb.v</username>	This file provides the synthesis black box for the user's synthesis.
<username>.ngo</username>	The ngo files provide the synthesized IP core.
<username>.lpc</username>	This file contains the IPexpress tool options used to recreate or modify the core in the IPex- press tool.
<username>.ipx</username>	IPexpress package file (Diamond only). This is a container that holds references to all of the elements of the generated IP core required to support simulation, synthesis and implementation. The IP core may be included in a user's design by importing this file to the associated Diamond project.
pmi_*.ngo	One or more files implementing synthesized memory modules used in the IP core.
*.rom	This file provides filter coefficient memory initialization data.

Table 6.1. File List

FPGA-IPUG-02043-1.5



The following additional files providing IP core generation status information are also generated in the Project Path directory:

- <username>_generate.tcl A TCL scripts which can regenerate the IP from command line.
- <username>_generate.log Synthesis and map log file.
- <username>_gen.log IPexpress IP generation log file.

6.4. Instantiating the Core

The generated FIR Filter IP core package includes black-box (<username>_bb.v) and instance (<*username>_*inst.v) templates that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file that can be used as an instantiation template for the IP core is provided in

\<project_dir>\fir_eval\<username>\src\rtl\top. You may also use this top-level reference as the starting template for the top-level for their complete design. By regenerating an IP core with the Clarity Designer tool, you can modify any of the options specific to an existing IP instance. By recreating an IP core with Clarity Designer tool, you can create (and modify if needed) a new IP instance with an existing LPC/IPX configuration file.

6.5. Running Functional Simulation

Simulation support for the FIR Filter IP core is provided for Aldec Active-HDL (Verilog and VHDL) simulator, Mentor Graphics ModelSim simulator. The functional simulation includes a configuration-specific behavioral model of the FIR Filter IP core. The test bench sources stimulus to the core, and monitors output from the core. The generated IP core package includes the configuration-specific behavior model (<username>_beh.v) for functional simulation in the Project Path root directory. The simulation scripts supporting ModelSim evaluation simulation is provided in **\<project_dir>\fir_eval\<username>\sim\modelsim\scripts**. The simulation script supporting Aldec evaluation simulation is provided in **\<project_dir>\fir_eval\<username>\sim\aldec\scripts**. Both Modelsim and Aldec simulation is supported via test bench files provided in **\<project_dir>\fir_eval\testbench**. Models required for simulation are provided in the corresponding \models folder.

To run the Aldec evaluation simulation:

- 1. Open Active-HDL.
- 2. Under the Tools tab, select Execute Macro.
- 3. Browse to folder **\<project_dir>\fir_eval\<username>\sim\aldec\scripts** and execute one of the **do** scripts shown.

To run the Modelsim evaluation simulation:

- 1. Open ModelSim.
- Under the File tab, select Change Directory and choose the folder <project_dir>\fir_eval\<username>\sim\modelsim\scripts.
- 3. Under the Tools tab, select **Execute Macro** and execute the ModelSim **do** script shown.

Note: When the simulation is complete, a pop-up window appears asking **Are you sure you want to finish?** Choose **No** to analyze the results. Choosing Yes closes ModelSim.

6.6. Synthesizing and Implementing the Core in a Top-Level Design

The FIR Filter IP core itself is synthesized and provided in NGO format when the core is generated through IPexpress. You may combine the core in your own top-level design by instantiating the core in your top-level file as described in Instantiating the Core and then synthesizing the entire design with either Synplify or Precision RTL Synthesis.

The following text describes the evaluation implementation flow for Windows platforms. The flow for Linux and UNIX platforms is described in the Readme file included with the IP core.

The top-level file <userame>_top.v is provided in **\<project_dir>\fir_eval\<username>\src\rtl\top**. Push-button implementation of the reference design is supported via the project file **<username>.ldf** located in **\<project_dir>\fir_eval\<user- name>\impl\synplify**.

To use this project file in Diamond:

^{© 2008-2018} Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice



- 1. Choose File > Open > Project.
- 2. Browse to **\<project_dir>\fir_eval\<username>\impl\synplify** in the Open Project dialog box.
- 3. Select and open **<username>_.ldf**. At this point, all of the files needed to support top-level synthesis and implementation will be imported to the project.
- 4. Select the Process tab in the left-hand interface window.
- 5. Implement the complete design through the standard Diamond interface flow.

6.7. Hardware Evaluation

The FIR Filter IP core 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 period of time (approximately four hours) without requiring the purchase of an IP license. It may also be used to evaluate the core in hardware in user-defined designs. The hardware evaluation capability may be enabled/disabled in the Properties menu of the Build Database setup in Diamond Project Navigator.

6.7.1. Enabling Hardware Evaluation in Diamond

To enable hardware evaluation in Diamond, choose **Project > Active Strategy > Translate Design Settings**. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default.

6.8. Updating/Regenerating the IP Core

By regenerating an IP core with the IP express tool, you can modify any of its settings including: device type, design entry method, and any of the options specific to the IP core. Regenerating can be done to modify an existing IP core or to create a new but similar one.

6.8.1. Regenerating an IP Core in Diamond

To regenerate an IP core in Diamond:

- 1. In IPexpress, click the **Regenerate** button.
- 2. In the Regenerate view of IPexpress, choose the IPX source file of the module or IP you wish to regenerate.
- 3. IPexpress shows the current settings for the module or IP in the Source box. Make your new settings in the Target box.
- 4. If you want to generate a new set of files in a new location, set the new location in the **IPX Target File** box. The base of the file name will be the base of all the new file names. The IPX Target File must end with an .ipx extension.
- 5. Click **Regenerate.** The module's dialog box opens showing the current option settings.
- 6. In the module dialog box, choose the desired options.

For more information about the options, click **Help**. Also, check the About tab in IPexpress for links to technical notes and user guides. IP may come with additional information.

As the options change, the schematic diagram of the module changes to show the I/O and the device resources the module needs.

- 7. To import the module into your project, if it's not already there, select **Import IPX to Diamond Project** (not available in stand-alone mode).
- 8. Click Generate.
- 9. Check the Generate Log tab to check for warnings and error messages.
- 10. Click Close.

The IPexpress package file (.ipx) supported by Diamond holds references to all of the elements of the generated IP core required to support simulation, synthesis and implementation. The IP core may be included in a user's design by importing the .ipx file to the associated Diamond project. To change the option settings of a module or IP that is already in a design project, double-click the module's .ipx file in the File List view. This opens IPexpress and the module's dialog box showing the current option settings. Thengo to step 6 above.

© 2008-2018 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.



6.9. Regenerating an IP Core in Clarity Designer Tool

To regenerate an IP core in Clarity Designer:

- 1. In the Clarity Designer Builder tab, right-click on the existing IP instance and choose Config.
- 2. In the module dialog box, choose the desired options.

For more information about the options, click **Help**. You may also click the About tab in the Clarity Designer window for links to technical notes and user guides. The IP may come with additional information.

As the options change, the schematic diagram of the module changes to show the I/O and the device resources the module needs.

3. Click Configure.

6.10. Recreating an IP Core in Clarity Designer Tool

To recreate an IP core in Clarity Designer:

- 1. In Clarity Designer click the Catalog tab.
- 2. Click the **Import IP** tab (at the bottom of the view).
- 3. Click Browse.
- 4. In the Open IPX File dialog box, browse to the .ipx or .lpc file of the module. Use the .ipx if it is available.
- 5. Click Open.
- 6. Type in a name for Target Instance. Note that this instance name should not be the same as any of the existing
- 7. IP instances in the current Clarity Designer project.
- 8. Click **Import**. The module's dialog box opens.
- 9. In the dialog box, choose desired options.

For more information about the options, click **Help**. You may also check the About tab in the Clarity Designer window for links to technical notes and user guides. The IP may come with additional information.

As the options change, the schematic diagram of the module changes to show the ports and the device resources the module needs.

10. Click Configure.



References

- LatticeXP2[™] Family Data Sheet (DS1009)
- LatticeECP3[™] Family Data Sheet (DS1021)
- ECP5[™] and ECP5-5G[™] Family Data Sheet (FPGA-DS-12012)



Technical Support Assistance

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



Appendix A. Resource Utilization

This appendix provides resource utilization information for Lattice FPGAs using the FIR IP core. The IP configurations shown in this chapter were generated using the IPexpress software tool and Clarity Designer tool. IPexpress and Clarity Designer are the Lattice IP configuration utility, and are included as a standard feature of the Diamond design tool. Details regarding the usage of IPexpress and Clarity Designer can be found in the IPexpress, Clarity Designer and Diamond help systems. For more information on the Diamond design tool, visit the Lattice web site at: www.latticesemi.com/software.

LatticeECP3 Devices

Table A.1. Performance and Resource Utilization (LatticeECP3)*

IPexpress User-Configurable Mode	Slices	LUTs	Registers	DSP Slices	sysMEM EBRs	f _{MAX} (MHz)
4 channels, 64 taps, multiplier multiplexing 64	134	254	222	4	2	227
1 channel, 32 taps, multiplier multiplexing 1	84	155	148	32	0	207
1 channel, 32 taps, multiplier multiplexing 4	260	238	482	10	8	153

***Note:** Performance and utilization characteristics are generated targeting an LFE3-150EA-6FN672C device using Lattice Diamond 3.10.2 and Synplify Pro D-2013.09L beta software. Performance may vary when using this IP core in a different density, speed or grade within the LatticeECP3 family or in a different software version.

Ordering Part Number

The Ordering Part Number (OPN) for the FIR Filter IP Core targeting LatticeECP3 devices is FIR-COMP-E3-U4.

LatticeXP2 Devices

Table A.2. Performance and Resource Utilization (LatticeXP2)*

IPexpress User-Configurable Mode	Slices	LUTs	Registers	18x18 Multipliers	sysMEM EBRs	f _{MAX} (MHz)
4 channels, 64 taps, multiplier multiplexing 64	105	204	165	1	1	197
1 channel, 32 taps, multiplier multiplexing 1	211	418	372	8	0	189
1 channel, 32 taps, multiplier multiplexing 4	159	272	304	2	8	207

***Note:** Performance and utilization characteristics are generated targeting an LFXP2-40E-7F672C device using Lattice Diamond 3.10.2 and Synplify Pro D-2013.09L beta software. Performance may vary when using this IP core in a different density, speed or grade within the LatticeXP2 family or in a different software version.

Ordering Part Number

The Ordering Part Number (OPN) for the FIR Filter IP Core targeting LatticeXP2 devices is FIR-COMP-X2-U4.

ECP5 Devices

Table A.3. Performance and Resource Utilization (LFE5U)*

Clarity User-Configurable Mode	Slices	LUTs	Registers	DSP Slices	sysMEM EBRs	f _{MAX} (MHz)
4 channels, 64 taps, multiplier multiplexing 64	129	248	222	4	2	211
1 channel, 32 taps, multiplier multiplexing 1	80	151	148	32	0	264
1 channel, 32 taps, multiplier multiplexing 4	260	239	482	10	8	177

*Note: Performance and utilization characteristics are generated targeting LFE5UM-85F-8MG756I using Lattice Diamond 3.10.2 and Synplify Pro F-2013.09L beta software. When using this IP core in a different density, speed, or grade within the ECP5 device family or in a different soft- ware version, performance may vary.

Ordering Part Number

The Ordering Part Number (OPN) for the FIR Filter IP Core targeting ECP5 devices is FIR- COMP-E5-U.



Revision History

Revision 1.5, June 2018

Section	Change Summary
All	Changed document number from IPUG79 to FPGA-IPUG-02043.
Introduction	Updated content.
Quick Facts	General update to Quick Facts tables.
Features	• Removed the line, "In ECP5, support high-speed. For low speed, support for half-band filter."
Functional Description	• Updated Figure 4.1. Top-Level Interface for the FIR Filter IP Core.
	Updated equation in FIR Filter Architecture.
	• Updated Figure 4.7 caption.
	Updated Coefficients Specification section.
	Updated Table 4.2 in Signal Descriptions section.
	Updated Interfacing with the FIR Filter IP Core section.
	Added Lattice ECP3 and ECP5 in Timing Specifications section.
Parameter Settings	• Updated Table 5.1. Parameter Specifications for the FIR Filter IP Core.
	• Updated Figure 5.1. Architecture Tab of the FIR Filter IP Core Interface.
	Updated Table 5.2. Architecture Tab.
	Updated Table 5.4. Implementation Tab. Added Synthesis Options description.
IP Core Generation and	Updated Figure 6.1. IPexpress Dialog Box.
Evaluation	Updated Figure 6.2. Configuration Dialog Box.
	Updated Figure 6.3. Clarity Designer Tool Dialog Box.
	Updated Figure 6.4. Clarity Designer Catalog Tab.
	Updated Figure 6.5. Fir Filter Dialog Box.
	Updated Figure 6.6. IP Configuration Interface.
	Updated Figure 6.7. FIR Filter IP Core Generated Directory Structure.
Appendix A. Resource Utilization	• Updated Table A.1. Performance and Resource Utilization (LatticeECP3)*.
	• Updated Table A.2. Performance and Resource Utilization (LatticeXP2)*.
	• Updated Table A.3. Performance and Resource Utilization (LFE5U)*.
Technical Support Assistance	General update.

Revision 1.4, May 2018

Section	Change Summary
All	Added support for ECP5 FPGA family.
	Updated document with new corporate logo.
	Updated Technical Support Information.

Revision 1.3, May 2011

Section	Change Summary
All	Added support for multipliers in multiple DSP rows.
	Changed interface timing for certain configurations in LatticeECP3 devices.



Revision 1.2, June 2010

Section	Change Summary
All	Added support for Diamond software throughout.
	Divided document into chapters. Added table of contents.
Quick Facts	Added Quick Facts tables.
IP Core Generation and Evaluation	Added new content.

Revision 1.1, April 2009

Section	Change Summary
All	Added support for LatticeECP3 FPGA family.
	• Updated appendices for ispLEVER 7.2 SP1.

Revision 1.0, September 2008

Section	Change Summary
All	Initial release.



7th Floor, 111 SW 5th Avenue Portland, OR 97204, USA T 503.268.8000 <u>www.latticesemi.com</u>