Cascaded Integrator-Comb (CIC) Filter User's Guide

August 2010
IPUG42_02.6

Table of Contents

Chapter 1. Introduction 4
Quick Facts 4
Features 8
Chapter 2. Functional Description 9
Interfacing with the CIC Filter 10
Signal Descriptions 12
Timing Diagrams 13
Decimator Timing 13
Interpolator Timing 15
Chapter 3. Parameter Settings 18
CIC Parameter Tab 18
Parameter Descriptions 19
Filter Type 19
Options 19
Differential Delay 19
Decimation/Interpolation Rates 20
Memory Type 20
Optional Port 20
Chapter 4. IP Core Generation 21
Licensing the IP Core 21
Getting Started 21
IPexpress-Created Files and Top Level Directory Structure 23
Instantiating the Core 25
Running Functional Simulation 25
Synthesizing and Implementing the Core in a Top-Level Design 25
Hardware Evaluation 26
Enabling Hardware Evaluation in Diamond 26
Enabling Hardware Evaluation in ispLEVER 26
Updating/Regenerating the IP Core 26
Regenerating an IP Core in Diamond 26
Regenerating an IP Core in ispLEVER 27
Chapter 5. Support Resources 28
Lattice Technical Support 28
Online Forums 28
Telephone Support Hotline 28
E-mail Support 28
Local Support 28
Internet 28
References 28
LatticeECP/EC 28
LatticeECP2M 28
LatticeECP3 28
LatticeSC/M 29
LatticeXP 29
LatticeXP2 29
Revision History 29
Appendix A. Resource Utilization 30
LatticeECP and LatticeEC FPGAs 30
Ordering Part Number 30

© 2010 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.

LatticeECP2 Devices 31
Ordering Part Number 31
LatticeECP2M Devices 31
Ordering Part Number 31
LatticeECP3 Devices 32
Ordering Part Number 32
LatticeSC/SCM Devices 32
Ordering Part Number 32
LatticeXP Devices 33
Ordering Part Number 33
LatticeXP2 Devices 33
Ordering Part Number 33

Introduction

Cascaded Integrator-Comb (CIC) filters, also known as Hogenauer filters, are used to achieve arbitrary and large sample rate changes in digital systems. These filters are used as decimation or interpolation filters and can be efficiently implemented without multipliers, utilizing only adders and subtractors.

A CIC filter is typically used in applications where the system sample rate is much larger than the bandwidth occupied by the signal. They are commonly used to build Digital Down Converters (DDCs) and Digital Up Converters (DUCs). Some applications that use the CIC filter include software design radios, cable modems, satellite receivers, 3G base stations, and radar systems.

Lattice provides a widely parameterizable CIC filter that supports multiple channels with run-time programmable rates and differential delay parameters.

Quick Facts

Table 1-1 through Table 1-8 give quick facts about the CIC Filter IP core for LatticeECP™, LatticeECP2™, LattceECP2M™, LatticeECP3™, LattticeSC™, LatticeSCM™, LatticeXP™, and LatticeXP2™ devices.

Table 1-1. CIC Filter IP Core for LatticeECP Devices Quick Facts

		CIC IP Configuration		
		Decimator with rate is 48 and data with is 8 and stage is 4	**Decimator with rate is 4096 and data with is 16 and stage is 8**	**Interpolator with rate is 35 and data with is 15 and stage is 7**
Core Requirements	FPGA Families Supported	Lattice ECP		
	Minimal Device Needed	LFEC1E	LFEC3E	LFEC3E
Resource Utilization	Targeted Device	LFECP33E-5F672C		
	LUTs	218	1514	878
	sysMEM EBRs	0	0	0
	Registers	301	1253	1982
Design Tool Support	Lattice Implementation	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. CIC Filter IP Core for LatticeECP2 Devices Quick Facts

		CIC IP Configuration		
		Decimator with rate is 48 and data with is 8 and stage is 4	**Decimator with rate is 4096 and data with is 16 and stage is 8**	**Interpolator with rate is 35 and data with is 15 and stage is 7**
Core Requirements	FPGA Families Supported	Lattice ECP2		
	Minimal Device Needed	LFE2-6E		
Resource Utilization	Targeted Device	LFE2-50E-7F672C		
	LUTs	221	1594	985
	sysMEM EBRs	0	0	0
	Registers	301	1253	1982
Design Tool Support	Lattice Implementation	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 ModelSim SE 6.3F		

Table 1-3. CIC Filter IP Core for LatticeECP2M Devices Quick Facts

		CIC IP Configuration		
		Decimator with rate is 48 and data with is 8 and stage is 4	**Decimator with rate is 4096 and data with is 16 and stage is 8**	**Interpolator with rate is 35 and data with is 15 and stage is 7**
Core Requirements	FPGA Families Supported	Lattice ECP2M		
	Minimal Device Needed	LFE2M20E		
Resource Utilization	Targeted Device	LFE2M35E-7F672C		
	LUTs	221	1594	985
	sysMEM EBRs	0	0	0
	Registers	301	1253	1982
Design Tool Support	Lattice Implementation	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 ModelSim SE 6.3F		

Table 1-4. CIC Filter IP Core for LatticeECP3 Devices Quick Facts

		CIC IP Configuration		
		Decimator with rate is 48 and data with is 8 and stage is 4	**Decimator with rate is 4096 and data with is 16 and stage is 8**	**Interpolator with rate is 35 and data with is 15 and stage is 7**
Core Requirements	FPGA Families Supported	Lattice ECP3		
	Minimal Device Needed	LFE3-35EA		
Resource Utilization	Targeted Device	LFE3-95E-8FN672CES		
	LUTs	220	1593	983
	sysMEM EBRs	0	0	0
	Registers	301	1253	1982
Design Tool Support	Lattice Implementation	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 ModelSim SE 6.3F		

Table 1-5. CIC Filter IP Core for LatticeSC Devices Quick Facts

		CIC IP Configuration		
		Decimator with rate is 48 and data with is 8 and stage is 4	**Decimator with rate is 4096 and data with is 16 and stage is 8**	**Interpolator with rate is 35 and data with is 15 and stage is 7**
Core Requirements	FPGA Families Supported	Lattice SC		
	Minimal Device Needed	LFSC3GA15E		
Resource Utilization	Targeted Device	LFSC3GA25E-7F900C		
	LUTs	212	1509	887
	sysMEM EBRs	0	0	0
	Registers	301	1256	1987
Design Tool Support	Lattice Implementation	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 ModelSim SE 6.3F		

Table 1-6. CIC Filter IP Core for LatticeSCM Devices Quick Facts

		CIC IP Configuration		
		Decimator with rate is 48 and data with is 8 and stage is 4	**Decimator with rate is 4096 and data with is 16 and stage is 8**	**Interpolator with rate is 35 and data with is 15 and stage is 7**
Core Requirements	FPGA Families Supported	Lattice SCM		
	Minimal Device Needed	LFSCM3GA15EP1		
Resource Utilization	Targeted Device	LFSCM3GA25EP1-7F900C		
	LUTs	212	1509	887
	sysMEM EBRs	0	0	0
	Registers	301	1256	1987
Design Tool Support	Lattice Implementation	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 ModelSim SE 6.3F		

Table 1-7. CIC Filter IP Core for LatticeXP Devices Quick Facts

		CIC IP Configuration		
		Decimator with rate is 48 and data with is 8 and stage is 4	**Decimator with rate is 4096 and data with is 16 and stage is 8**	**Interpolator with rate is 35 and data with is 15 and stage is 7**
Core Requirements	FPGA Families Supported	Lattice XP		
	Minimal Device Needed	LFXP3E		
Resource Utilization	Targeted Device	LFXP20E-5F484C		
	LUTs	218	1514	878
	sysMEM EBRs	0	0	0
	Registers	301	1253	1982
Design Tool Support	Lattice Implementation	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 ModelSim SE 6.3F		

Table 1-8. CIC Filter IP Core for LatticeXP2 Devices Quick Facts

		CIC IP Configuration		
		Decimator with rate is 48 and data with is 8 and stage is 4	**Decimator with rate is 4096 and data with is 16 and stage is 8**	**Interpolator with rate is 35 and data with is 15 and stage is 7**
Core Requirements	FPGA Families Supported	Lattice XP2		
	Minimal Device Needed	LFXP2-5E		
Resource Utilization	Targeted Device	LFXP2-17E-7F484C		
	LUTs	221	1594	985
	sysMEM EBRs	0	0	0
	Registers	301	1253	1982
Design Tool Support	Lattice Implementation	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 ModelSim SE 6.3F		

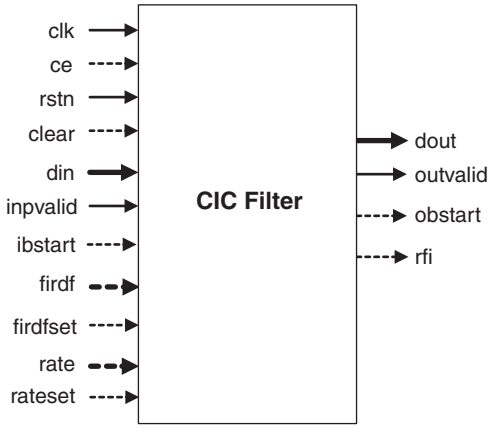
Features

- 1-32-bit input data width
- 1-8 cascaded stages
- 1-4 cycles differential delay, run-time programmable for both decimation and interpolation
- 2-16,384 decimation and interpolation sampling rate factor, run-time programmable rates for both decimation and interpolation
- Multi-channel (up to 40 channels) support for both decimation and interpolation
- Fully synchronous, single-clock design

Functional Description

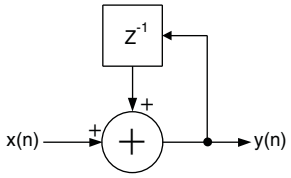
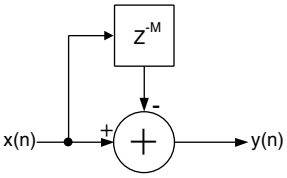
This chapter provides a functional description of Lattice's CIC Filter IP core. Figure 2-1 shows a top-level inteface diagram for the CIC Filter.

Figure 2-1. Top-level Interface Diagram for CIC Filter



A CIC filter is constructed using two kinds of blocks: an integrator and a comb. An integrator is a single-pole IIR filter with a unity feedback coefficient operating at a higher sampling rate, f_S. A comb is a FIR filter with M unity differential delays operating at a lower sampling rate, f_S/R, where M and R are integers. The main features of the integrator and the comb are summarized in Table 2-1.

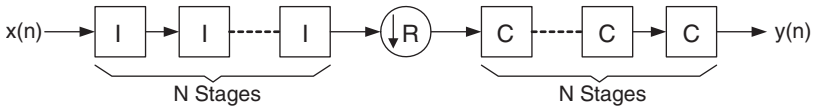
Table 2-1. Main Features of Integrator and Comb

Feature	Integrator	Comb
Structure		
Sampling Rate	f_s	f_s/R
y[n]	y[n-1] + x[n]	x[n] -x[n-RM]
$H_I(z)$	$\frac{1}{1 - z^{-1}}$	$1 - z^{-RM}$
$\lvert H(e^{jw})\rvert^2$	$\frac{1}{2(1 - \cos\omega)}$	$2(1 - \cos RM\omega)$

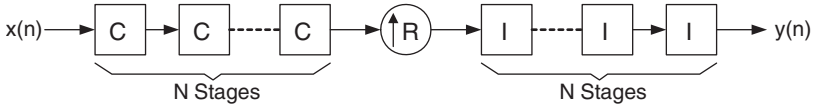
A CIC decimator and interpolator with N stages are shown in Figure 2-2, where I and C represent an integrator and a comb, respectively. The symbols, ↓R and ↑R, represent down-sampling and up-sampling, respectively.

Figure 2-2. CIC Decimator and Interpolator

(a) CIC Decimator



(b) CIC Interpolator



The system transfer function of the CIC decimator and interpolator in the z-plane is

$$H(z) = \frac{(1 - z^{-RM})^N}{(1 - z^{-1})^N} = \left(\sum_{k=0}^{RM-1} z^{-k} \right)^N \quad (1)$$

The frequency response can be derived by substituting $z=j^{2¼f}$ in Equation 1, where f is the frequency relative to the low sampling rate, f_S/R.

$$H(f) = \left(\frac{\sin \pi Mf}{\sin \frac{\pi f}{R}} \right)^N \quad (2)$$

By using sin xÝx for small x, we can approximate Equation 2 for a large R as

$$H(f) = \left(\frac{\sin \pi Mf}{\frac{\pi f}{R}} \right)^N = \left(RM \frac{\sin \pi Mf}{\pi Mf} \right)^N = [RM\text{sinc}(\pi Mf)]^N \quad \text{if } f << M \quad (3)$$

Equation 3 indicates that the frequency response has nulls (zeros) at integer multiples of f=1/M.

As given in Equation 3, the frequency response of the CIC filter resembles a sinc waveform. The main lobe of the sinc wave is the passband and the side lobes are the aliasing or imaging bands. The parameters N, M, and R decide the characteristics of the passband and the aliasing bands.

Interfacing with the CIC Filter

To ensure proper functioning of the CIC filter for different usage scenarios, the handshake signals must be correctly used and the rules of interfacing must be followed. For a CIC filter, the input and output data rates are not the same and one of the rates is higher than the other by a factor of R, the sampling rate factor. For multi-channel operations, the channel index changes more frequently than the data index, in both input and output data streams. For example, for a three-channel decimator, data 0 of channel 0 is followed by data 0 of channel 1 and data 0 of channel 2. After data 0 of all the channels are applied, data 1 for the channels follows. For easier understanding, the terms “input data cycle” and “output data cycle” are used and explained for each case.

For a decimator, R input samples are read for every output sample. For a single-channel decimator, the inputs are applied sequentially and one output data is available every R cycles. Here the input data cycle is one cycle long and the output data cycle is R cycles long. The availability of valid output data is indicated by the `outvalid` signal, which goes high for the first clock cycle of an output data cycle. There could be arbitrary breaks in the input data

and such breaks have to be indicated by pulling the inpvalid low during the duration of the break. There will be corresponding breaks in the output, indicated by the outvalid signal going low.

For a multi-channel decimator, an input data cycle consists of the set of identically indexed data for each channel. The input block start signal (ibstart) is checked only for the first clock cycle in an input data cycle and input valid signal (inpvalid) should be high for all valid input data. Once an ibstart is received, the filter sequentially reads the input data if inpvalid is valid and assigns them sequential channel indices. For example, for a three-channel decimator, the input data during the first valid ibstart signal is read as data 0 from channel 0. The input data during the next two successive clocks are read as data 0 for channel 1 and data 0 for channel 2. After reading all the data for an input data cycle, the filter waits for the next input data cycle by scanning the ibstart signal. For maximum throughput, the input data must be applied without breaks. The additional output signal, obstart, available for a multi-channel decimator, indicates the start of an output data cycle or channel 0 of the output data stream.

An interpolating CIC filter reads one input sample for every R output samples it computes. For a single-channel interpolator, an input data cycle is R clock cycles long and the data is read during the first clock cycle of a data cycle (when inpvalid is high). The inpvalid signal is not scanned for the rest of the input data cycle. After the end of the input data cycle, the CIC filter waits for new data by scanning the inpvalid signal. The output signal rfi is asserted high one cycle before the filter is ready to receive the next input data. When the input data cycles are continuous, without gaps in between them, the outvalid signal is always high. If there are gaps between the input data cycles, there will be corresponding gaps in the output, which is indicated by outvalid going low. For maximum throughput, input data must be applied every time the rfi goes high.

For a multi-channel interpolator, an input data cycle is equal to R*C clock cycles, where C is the number of channels. The cycle consists of the set of identically indexed data for each channel, followed by R-1 clock cycles of blank data per channel, during which no input data is read. The ibstart signal is checked only during the first data of an input data cycle and is not scanned for the rest of the cycle and inpvalid should be high for each valid input data. After the end of the input data cycle, the CIC filter waits for the new data by scanning the ibstart and ipvalid signals. The output signal rfi is asserted high one cycle before the filter is ready to receive the next block of input data. When the input data cycles are continuous, without gaps in between them, the outvalid signal is always high. If there are gaps between the input data cycles, there will be corresponding gaps in the output, which is indicated by outvalid going low. For continuous operation, the driving system can check the rfi signal and start an input cycle after rfi goes high. The output data cycle has as many clock cycles as the number of channels, plus any gaps in the input data stream. The start of an output data cycle, which is also the output data for channel 0, is indicated by the output signal, obstart, going high.

Signal Descriptions

The I/O port definitions are given in Table 2-2. The optional ports are represented as `portname` and may be selected based on the parameters given in Table 3-1 on page 18.

Table 2-2. Interface Signal Descriptions

Name	Bits	Active	I/O	Description
All Configurations				
`clk`	1	↑	I	System Clock. This is the reference clock for input and output data.
`rstn`	1	L	I	Asynchronous System Reset.
`din`	1 to 32	—	I	Input Data in Signed Format.
`dout`	2 to 160	—	O	Output Data in Signed Format.The width of this port is equal to the next higher integer value of [`Input data width` + `Stages` ($\log_2$ (RATE x DF_DELAY)], where, • RATE = `Rate` when `Programmable rate` is not selected. RATE = `Max rate` when `Programmable rate` is selected. • DF_DELAY = `Delay` when `Programmable delay` is not selected. DF_DELAY = `Max delay` when `Programmable delay` is selected.
`outvalid`	1	H	O	Output Data Valid. Indicates that output port `dout` contains a valid sample.
`inpvalid`	1	H	I	Input Data Valid. Indicates that the input port `din` contains a valid sample.
`rfi`	1	H	O	Ready For Input. Indicates that the CIC filter is ready to accept an input sample in the next clock cycle. If `rfi` is low, the input sample in the next clock cycle will be ignored even if `inpvalid` is asserted high. Available only if `Filter type` = “Interpolator”.
Multi-channel Mode Only (when `Number of channels` > 1)				
`ibstart`	1	H	I	Input Block Start. Indicates that the first valid data of an input data cycle is being presented at the input port `din`.
`obstart`	1	H	O	Output Block Start. Indicates that output port `dout` contains a valid sample for the first channel.

Table 2-2. Interface Signal Descriptions (Continued)

Name	Bits	Active	I/O	Description
Programmable Sampling Rate Mode Only (when `Programmable delay = "Yes"`)				
rate	2 to 145	—	I	Sampling Rate. This port is used for feeding the dynamic sampling rate factor of the CIC filter. The width of this port is equal to the next higher integer value of $\log_2$ (`Max rate +1`).
rateset	1	H	I	Set Sampling Rate. Indicates that input port `rate` contains a valid rate factor.
Programmable Differential Delay Mode Only (when `Programmable rate = "Yes"`)				
firdf	1 or 3	—	I	Differential Delay. This port is used for feeding the dynamic differential delay of the FIR stages. The width of this port is equal to the next higher integer value of $\log_2$ (`Max delay +1`).
firdfset	1	H	I	Set Differential Delay. Indicates that input port `firdf` contains a valid differential delay factor.
Other Optional Ports				
ce	1	H	I	Clock Enable. This signal has the highest priority after `rstn`. The CIC filter operation halts as long as `ce` is held low. Choosing this option will increase resource utilization. Available only if `Clock enable` is selected.
clear	1	H	I	System Clear. The optional signal `ce`, if used, must be held high for `clear` to be effective. Choosing this option will increase resource utilization. Available only if `System clear` is selected.

Timing Diagrams

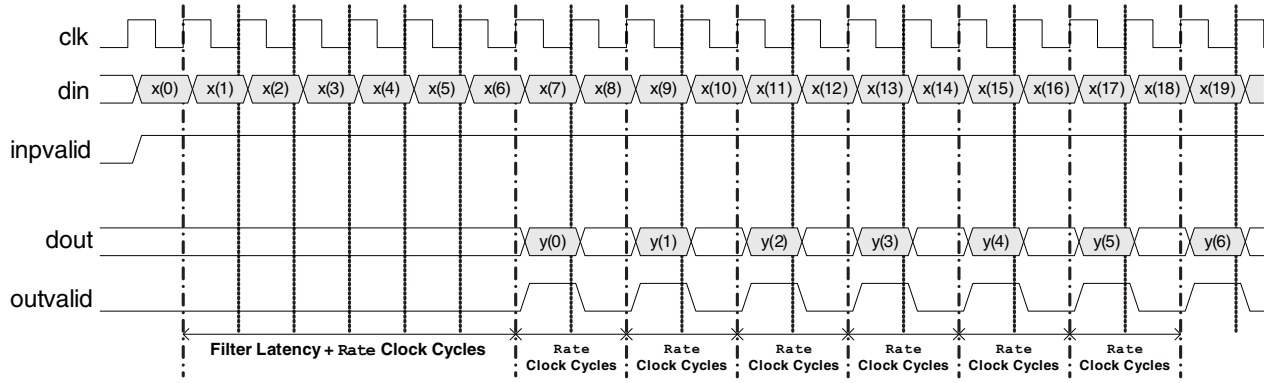
Decimator Timing

Interface timing for four cases of decimators are presented in Figure 2-3 and a brief description of each case is given below. The down-sampling rate is equal to 2 (`Rate` = 2) for all cases described.

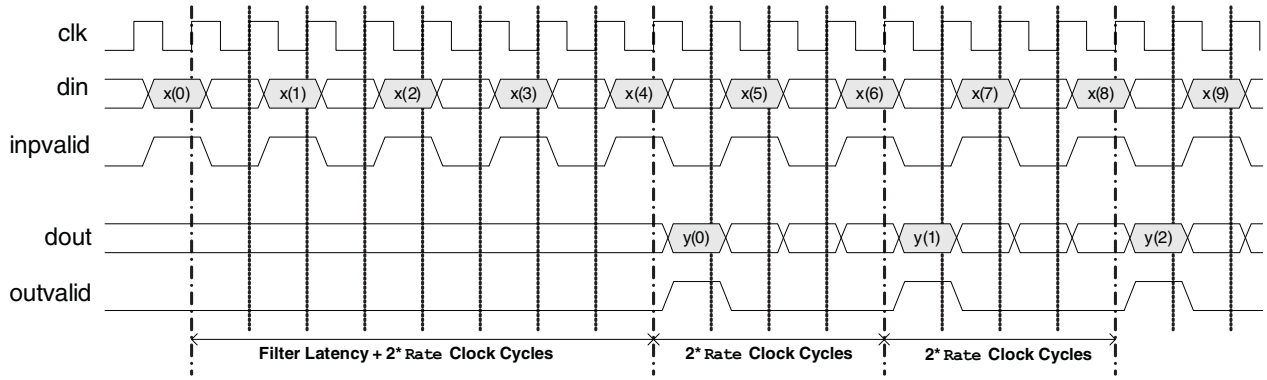
- Figure 2-3(a): A new input sample is available at every clock cycle in a continuous manner, so `inpvalid` signal is continuously held high. There is one valid output for every `Rate` clock cycles, as indicated by the `outvalid` signal.
- Figure 2-3(b): A new input sample is available once in every two clock cycles in an intermittent manner. As one input is given for every two clock cycles, there is one valid output for every four (2*`Rate`) clock cycles, as indicated by the `outvalid` signal.
- Figure 2-3(c): Interlaced dual-channel input samples x and u are supplied in every clock cycle in a continuous manner, so `ibstart` signal is high during the first channel sample x input. For the second channel sample u, `ibstart` must be low. `Inpvalid` is high for each valid input sample. After a few clock cycles of the first input sample to the IP core, the `outvalid` signal will be asserted for each valid output data. The signal `obstart` indicates data for the first channel. Then, subsequent output samples for each channel will be available every 2*`Rate` clock cycles.
- Figure 2-3(d): Interlaced dual-channel input samples x and u are supplied in every two-clock clock cycles in an intermittent manner. After a few clock cycles of the first input sample to the IP core, the `outvalid` signal will be asserted for each valid output data. The signal `obstart` indicates data for the first data. Then, subsequent output samples for each channel will be available every 4*`Rate` clock cycles.

Figure 2-3. CIC Decimator Timing Diagrams

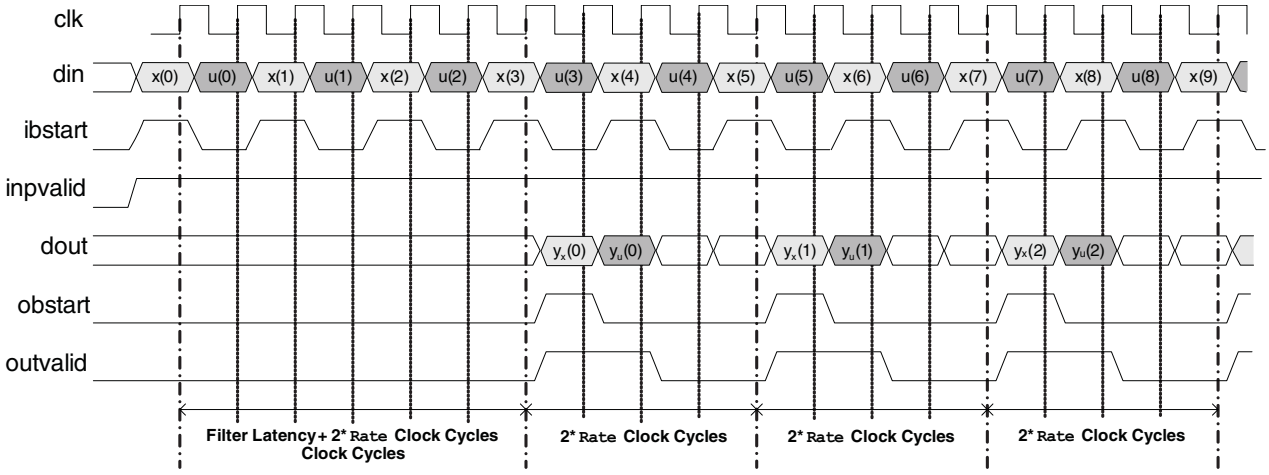
(a) Single-Channel: Inputs are sampled in every clock cycle:



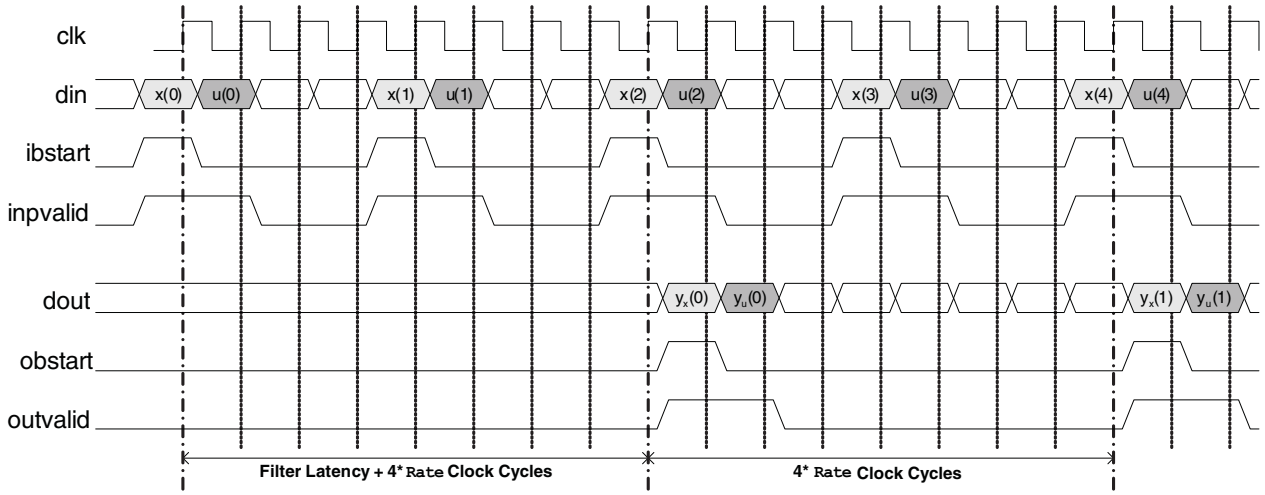
(b) Single-Channel: Inputs are sampled in every two-clock cycle:



(c) Dual-Channel: Inputs are sampled in every clock cycle:



(d) Dual-Channel: Inputs are sampled in every two clock cycles:



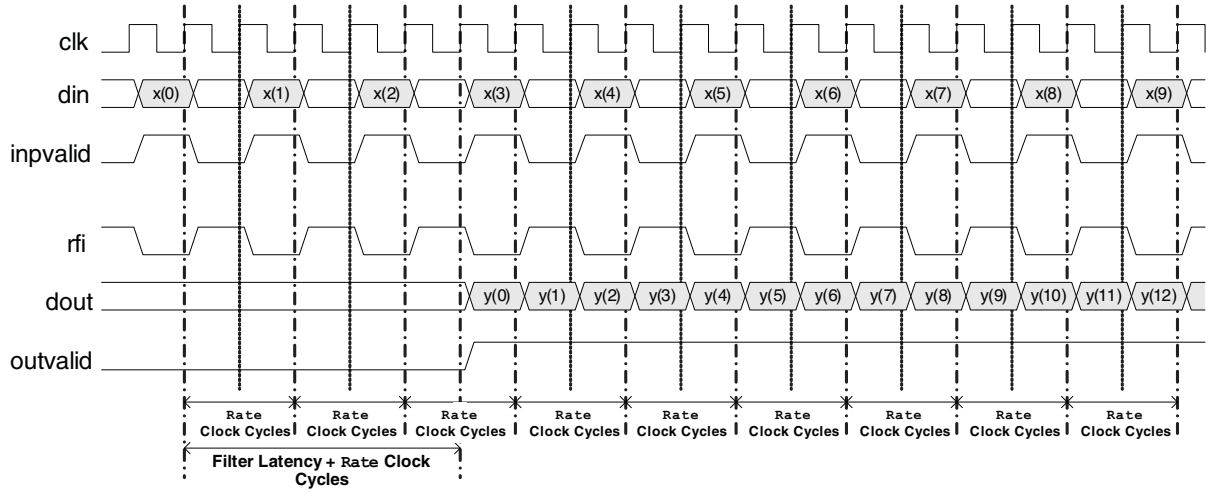
Interpolator Timing

The interface timing for four cases of interpolators are presented in Figure 2-4 and a brief description of each case is given below. The up-sampling rate is equal to 2 (`Rate` = 2) for all cases described.

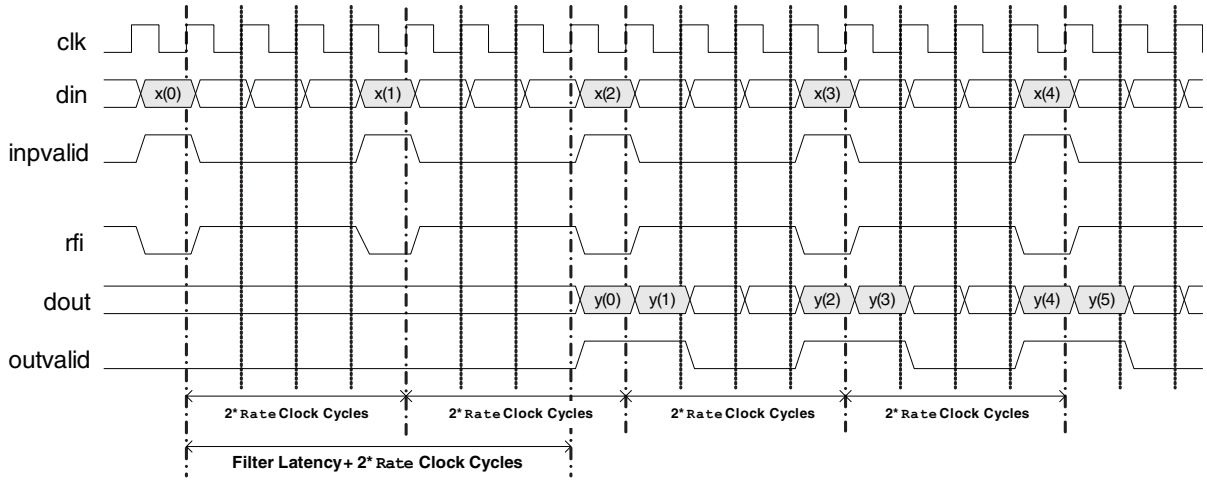
- Figure 2-3(a): A new input sample is available in every `Rate` clock cycles, so `inpvalid` signal during those cycles. The CIC will activate the `rfi` signal when it is ready to read the next sample. When `rfi` is inactive, the CIC cannot accept a new input sample in the next clock cycle. The output samples are available continuously, after the initial latency.
- Figure 2-3(b): A new input sample is available in every 2*`Rate` clock cycles. The IP core will deactivate the `rfi` signal for (`Rate` - 1) clock cycles when a sample is inputted. When `rfi` is inactive, IP core cannot accept a new input sample in the next clock cycle. After a few clock cycles of the first input sample to the IP core, the `outvalid` signal will be asserted for `Rate` clock cycles to indicate the output samples interpolated from the first input sample are available. Then, subsequent output samples will be available for `Rate` clock cycles in every 2*`Rate` clock cycles intermittently.
- Figure 2-3(c): Interlaced dual-channel input samples x and u are supplied in every 2*`Rate` clock cycles, so `ibstart` signal is high in every `Rate` clock cycles during the first channel sample x input. The IP core will deactivate `rfi` signal for 2*(`Rate` - 1) clock cycles happened before the last channel input. When `rfi` is inactive, IP core cannot accept a new block of input samples in the next clock cycle. After a few clock cycles of the first input sample to the IP core, the `outvalid` signal will be asserted to indicate the output sample yx for x and yu for u are available. The signal `obstart` indicates data for the first channel. Then, subsequent output samples will be available continuously.
- Figure 2-3(d): Interlaced dual-channel input samples x and u are supplied in every 4*`Rate` clock cycles. The IP core will deactivate the `rfi` signal for 2*(`Rate` - 1) clock cycles before the last channel input. When `rfi` is inactive, the IP core cannot accept a new input sample in the next clock cycles. After a few clock cycles of the first input sample to the IP core, the `outvalid` signal will be asserted to indicate the output sample yx for x and yu for u are available. The signal `obstart` indicates data for the first channel. Then, subsequent output samples will be available for 2*`Rate` clock cycles in every 4*`Rate` clock cycles intermittently.

Figure 2-4. CIC Interpolator Timing Diagrams

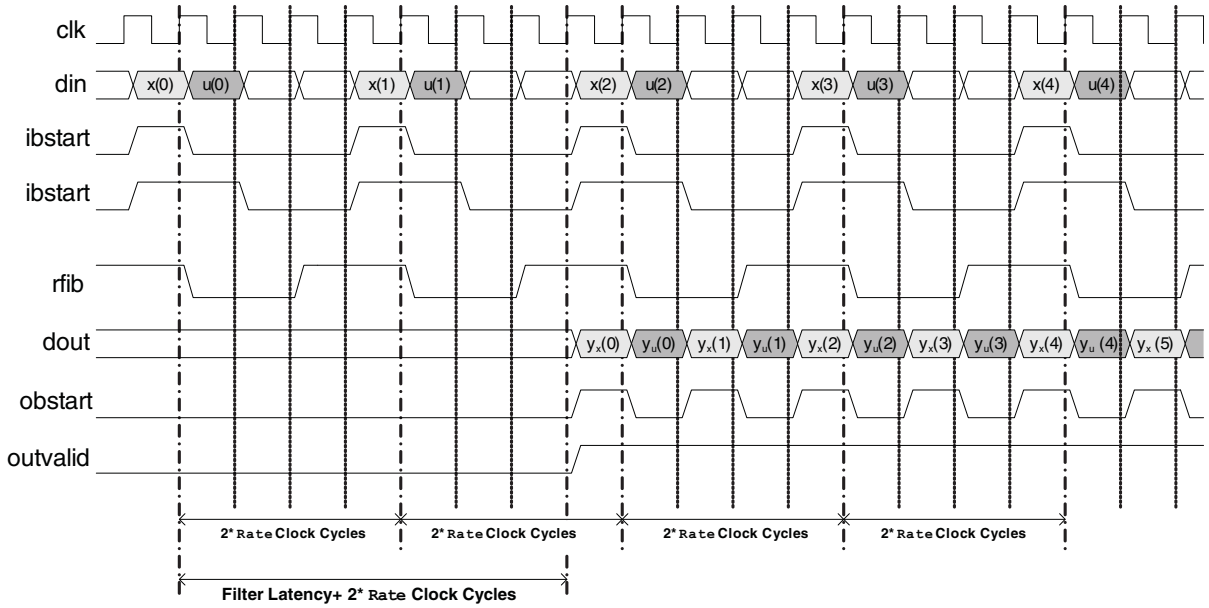
(a) Single-Channel: Inputs are sampled in every `Rate` clock cycles:



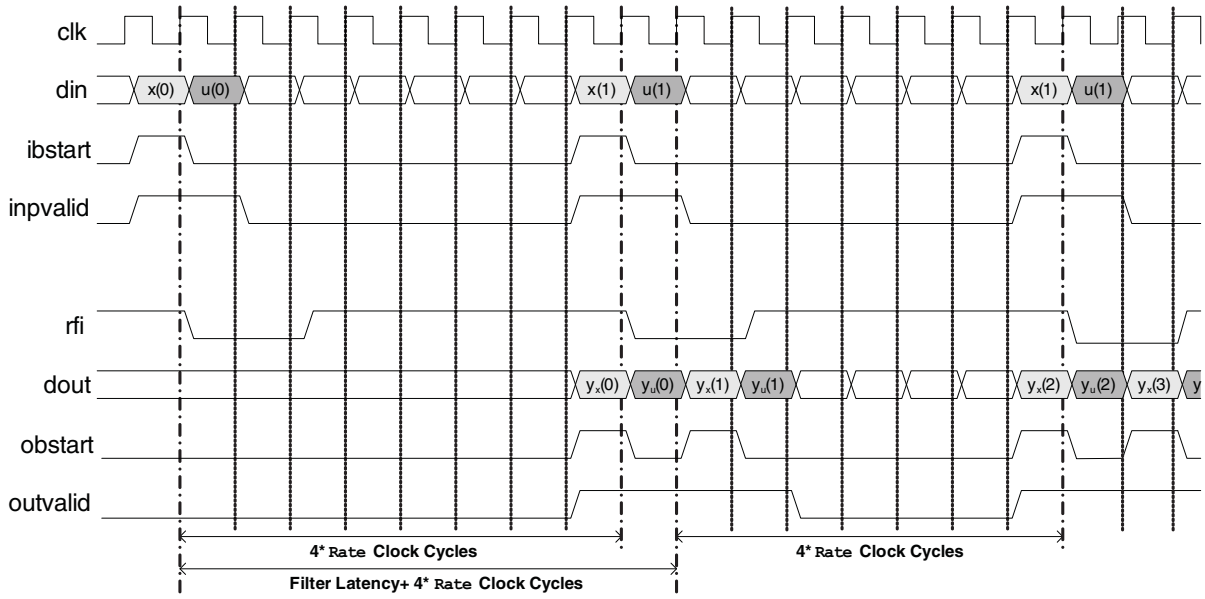
(b) Single-Channel: Inputs are sampled in every 2*`Rate` clock cycles:



(c) Dual-Channel: Inputs are sampled in every `Rate` clock cycles:



(d) Dual-Channel: Inputs are sampled in every 2*`Rate` clock cycles:



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 21 for a description on how to generate the IP.

Table 3-1 provides the list of user configurable parameters for the CIC Filter IP core. The parameter settings are specified using the CIC Parameter Tab in IPexpress.

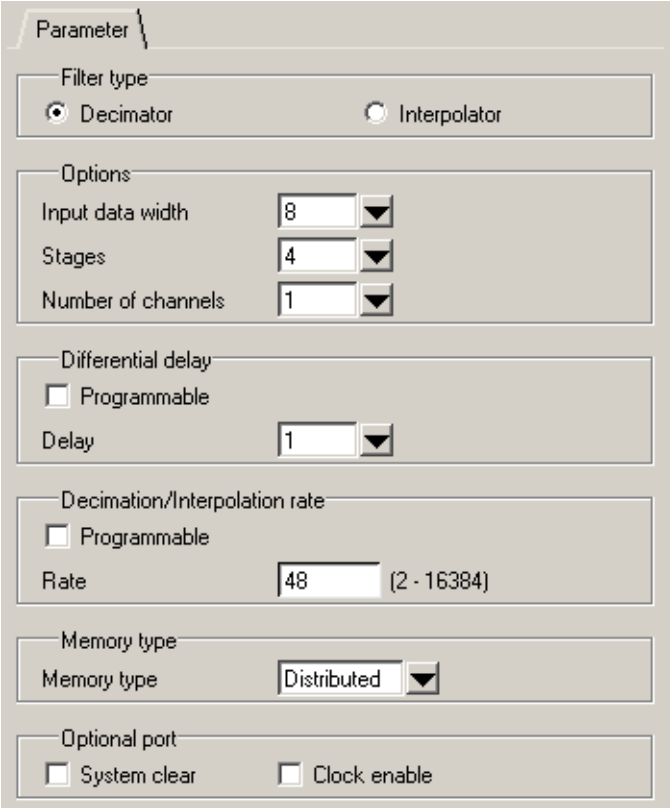
Table 3-1. CIC Filter IP Core Configuration Parameters

Parameter	Range/Options	Default
Filter Type		
Filter type	Decimator, Interpolator	Decimator
Options		
Input data width	1 to 32 bits	8 bits
Stages	1 to 8	4
Number of channels	1 to 40	1
Differential Delay		
Programmable delay	Yes, No	No
Delay	1 to 4	1
Max delay	2 to 4	2
Decimation/Interpolation Rates		
Programmable rate	Yes, No	No
Rate	2 to 16384	48
Max rate	4 to 16384	64
Min rate	2 to 8	2
Memory Type		
Memory Type	Distributed, EBR	Distributed
Optional Port		
System clear	Yes, No	No
Clock enable	Yes, No	No

CIC Parameter Tab

Figure 3-1 shows the Parameter Tab..

Figure 3-1. CIC Parameter Tab



Parameter Descriptions

This section describes the available parameters for the CIC Filter IP core.

Filter Type

This parameter determines whether the CIC filter is configured as an interpolator or a decimator.

Options

Input Data Width

This parameter specifies the width of the input data in bits.

Stages

This parameter specifies the number of integrator and comb stages.

Number of Channels

This parameter specifies the number of time-shared data channels.

Differential Delay

Programmable Delay

This parameter determines whether the differential delay is fixed or programmable. When `Programmable delay` is selected, ports `firdf` and `firdfset` will be added.

Delay

This parameter is only available when Programmable delay is not selected. This parameter sets the differential delay factor in each FIR filter.

Max Delay

This parameter is only available when Programmable delay is selected. This parameter specifies the max differential delay factor in each FIR filter.

Decimation/Interpolation Rates

Programmable Rate
This parameter determines whether the sampling rate is fixed or programmable. When `Programmable rate` is selected, ports `rate` and `rateset` will be added.

Rate
This parameter is only available when Programmable rate is not selected. This parameter specifies down-sampling or up-sampling rate.

Max Rate
This parameter is only available when Programmable rate is selected. This parameter specifies the maximum value for the down-sampling or the up-sampling rate.

Min Rate
This parameter is only available when Programmable rate is selected. This parameter specifies the minimum down-sampling or up-sampling rate. Note that more logic resources will be used as a smaller Min rate is selected.

Memory Type

Memory Type
This parameter indicates which type of memory is used in the core, distributed (LUT-based) or EBR.

Optional Port

System Clear
This parameter determines whether the system clear port clear is present.

Clock Enable
This parameter determines whether the clock enable port ce is present.

IP Core Generation

This chapter provides information on how to generate the CIC Filter 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 CIC 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 CIC Filter IP core and fully evaluate the core through functional simulation and implementation (synthesis, map, place and route) without an IP license. The CIC 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 "Hardware Evaluation" on page 26 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.

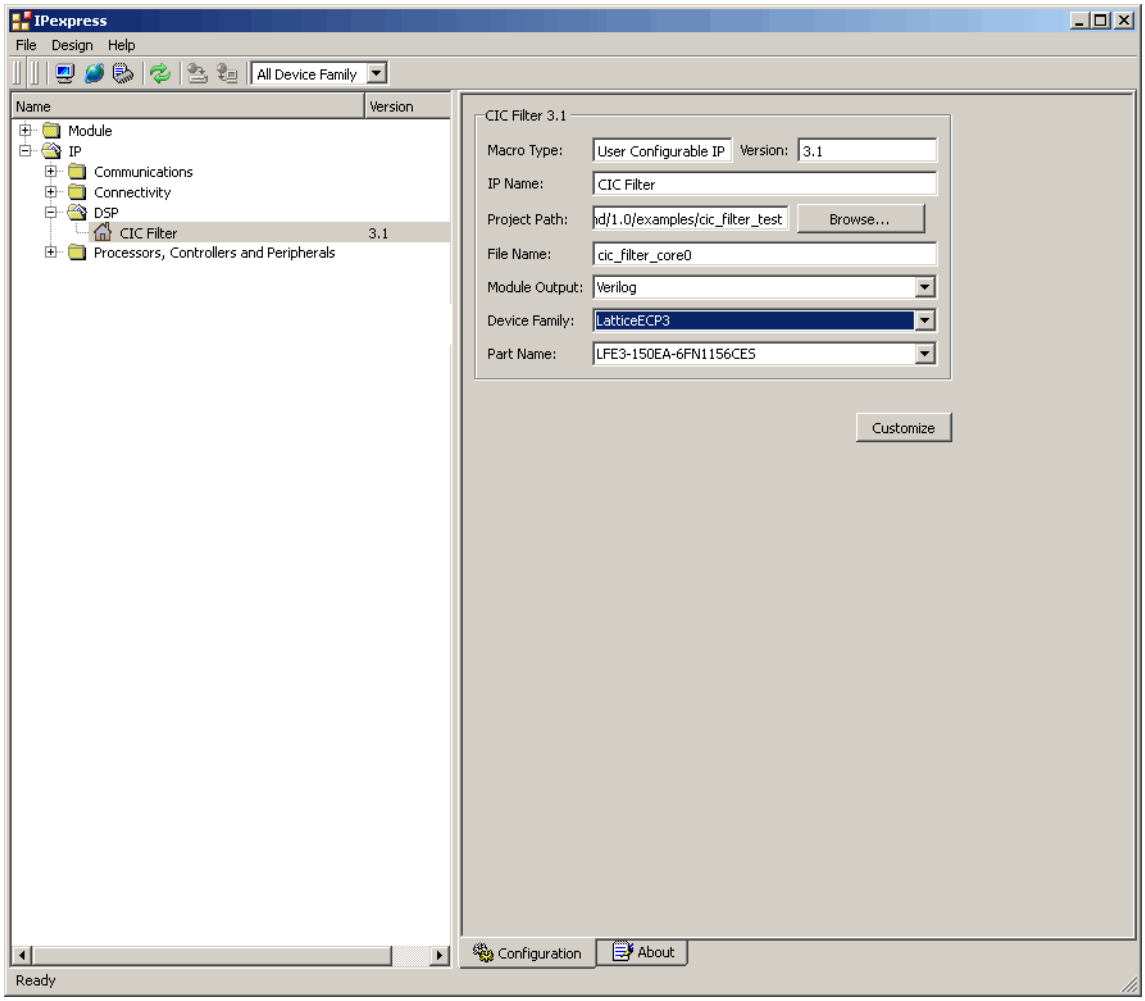
Getting Started

The CIC Filter 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 CIC Filter 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. LatticeSCM, Lattice ECP2M, LatticeECP3, etc.). Only families that support the particular IP core are listed.
- **Part Name** – Specific targeted part within the selected device family.

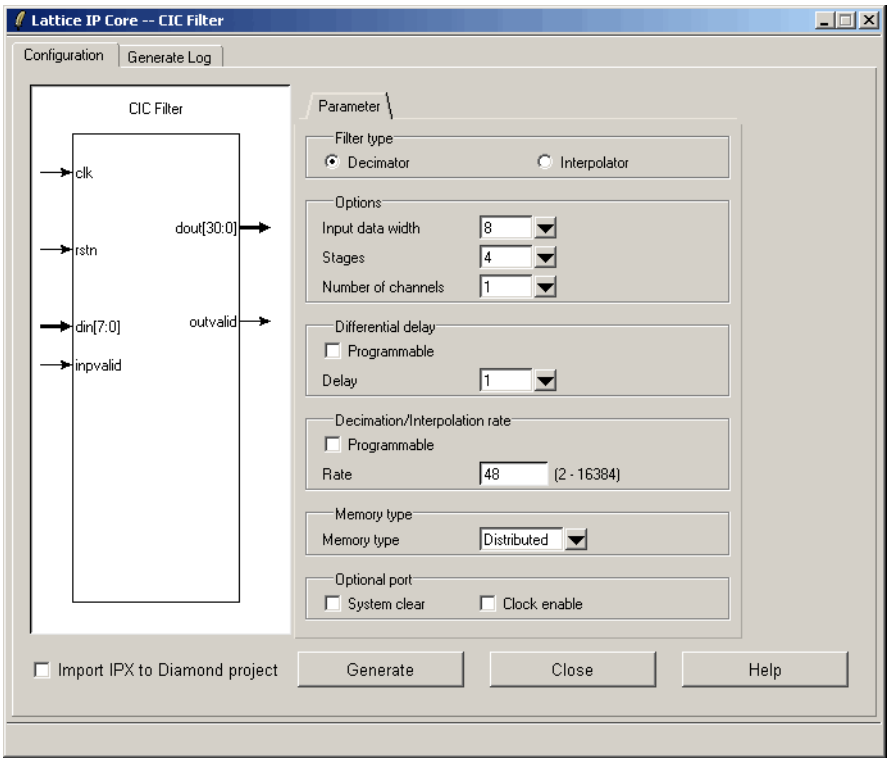
Figure 4-1. IPexpress Dialog Box (Diamond Version)



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 CIC Filter 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 18 for more information on the CIC Filter IP core parameter settings.

Figure 4-2. Configuration Dialog Box (Diamond Version)



IPexpress-Created Files and Top Level Directory Structure

When the user clicks the **Generate** button in the IP Configuration dialog box, 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 4-3.

Figure 4-3. LatticeECP3 CIC Filter IP Core Directory Structure

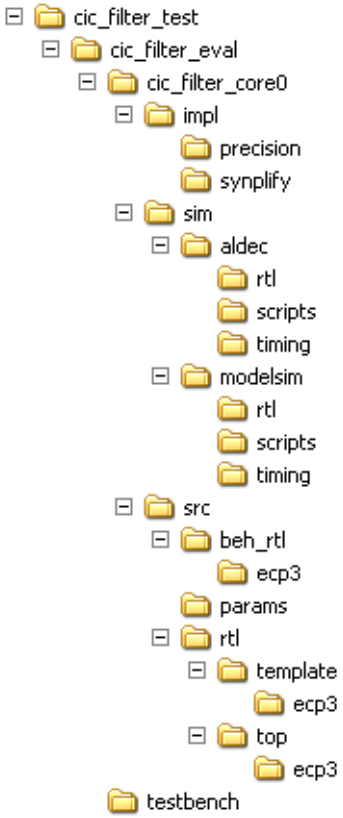


Table 4-1 provides a list of key files created by the IPexpress tool and how they are used. The IPexpress tool creates several files that are used throughout the design cycle. The names of most of the created files are customized to the user's module name specified in the IPexpress tool.t

Table 4-1. File List

File	Description
<*username*>_inst.v	This file provides an instance template for the IP.
<*username*>_bb.v	This file provides the synthesis black box for the user's synthesis.
<*username*>.ngo	The ngo files provide the synthesized IP core.
<*username*>.ipx	The IPX file holds references to all of the elements of an IP or Module after it is generated from the IPexpress tool (Diamond version only). The file is used to bring in the appropriate files during the design implementation and analysis. It is also used to re-load parameter settings into the IP/Module generation GUI when an IP/Module is being re-generated.
<*username*>.lpc	This file contains the IPexpress tool options used to recreate or modify the core in the IPexpress tool.
<*username*>_top.[v,vhd]	This file provides a module which instantiates the CIC IP core. This file can be easily modified for the user's instance of the CIC IP core. This file is located in the `cic_filter_eval/<username>/src/rtl/top/` directory.
<*username*>_generate.tcl	Created when the GUI **Generate** button is pushed, this file invokes generation, may be run from command line.
<*username*>_generate.log	This is the IPexpress scripts log file.
<*username*>_gen.log	This is the IPexpress IP generation log file

Instantiating the Core

The generated CIC 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*>\cic_filter_eval\<*username*>\src\rtl\top. Users may also use this top-level reference as the starting template for the top-level for their complete design.

Running Functional Simulation

Simulation support for the CIC IP core is provided for Aldec Active-HDL (Verilog and VHDL) simulator and Mentor Graphics ModelSim simulator. The functional simulation includes a configuration-specific behavioral model of the CIC 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>\cic_filter_eval\<*username*>\sim\modelsim\scripts. The simulation script supporting Aldec evaluation simulation is provided in \<project_dir>\cic_filter_eval\<*username*>\sim\aldec\scripts. Both ModelSim and Aldec simulation is supported via test bench files provided in \<*project_dir*>\cic_filter_eval\testbench. Models required for simulation are provided in the corresponding \models folder. Users may run the Aldec evaluation simulation by doing the following:

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

Users may run the ModelSim evaluation simulation by doing the following:

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

Note: *When the simulation completes, a pop-up window will appear asking "Are you sure you want to finish?" Answer* ***No*** *to analyze the results. (Answering* ***Yes*** *closes ModelSim).*

Synthesizing and Implementing the Core in a Top-Level Design

Synthesis support for the CIC IP core is provided for Mentor Graphics Precision or Synopsys Synplify. The CIC IP core itself is synthesized and is provided in NGO format when the core is generated in IPexpress. Users may synthesize the core in their own top-level design by instantiating the core in their top-level as described previously 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 files <*username*>_top.v is provided in \<*project_dir*>\cic_filter_eval\<*username*>\src\rtl\top. Push-button implementation of the reference design is supported via Diamond or ispLEVER project files, <*username*>.syn, located in the following directory: \<*project_dir*>\cic_filter_eval\<*username*>\impl\(synplify or precision).

To use this project file in Diamond:

1. Choose **File > Open > Project**.

2. Browse to \<*project_dir*>\cic_filter_eval\<*username*>\impl\synplify (or precision) 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 GUI window.

5. Implement the complete design via the standard Diamond GUI flow.

To use this project file in ispLEVER:

1. Choose **File > Open Project.**

2. Browse to \<*project_dir*>\cic_filter_eval\<*username*>\impl\synplify (or precision) in the Open Project dialog box.

3. Select and open <*username*>.syn. At this point, all of the files needed to support top-level synthesis and implementation will be imported to the project.

4. Select the device top-level entry in the left-hand GUI window.

5. Implement the complete design via the standard ispLEVER GUI flow.

Hardware Evaluation

The CIC Filter IP core supports 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.

Enabling 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.

Enabling Hardware Evaluation in ispLEVER

In the Processes for Current Source pane, right-click the **Build Database** process and choose **Properties** from the dropdown menu. The hardware evaluation capability may be enabled/disabled in the Properties dialog box. It is enabled by default.

Updating/Regenerating the IP Core

By regenerating an IP core with the IPexpress 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.

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 T**arget** 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 dialog box, choose the desired options. To get 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 will need.

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. Then go to step 6 above.

Regenerating an IP Core in ispLEVER

To regenerate an IP core in ispLEVER:

1. In the IPexpress tool, choose **Tools > Regenerate IP/Module**.

2. In the Select a Parameter File dialog box, choose the Lattice Parameter Configuration (.lpc) file of the IP core you wish to regenerate, and click **Open**.

3. The Select Target Core Version, Design Entry, and Device dialog box shows the current settings for the IP core in the Source Value box. Make your new settings in the Target Value box.

4. If you want to generate a new set of files in a new location, set the location in the LPC Target File box. The base of the .lpc file name will be the base of all the new file names. The LPC Target File must end with an .lpc extension.

5. Click **Next**. The IP core's dialog box opens showing the current option settings.

6. In the dialog box, choose desired options. To get information about the options, click **Help**. Also, check the About tab in the IPexpress tool for links to technical notes and user guides. The IP core might come with additional information. As the options change, the schematic diagram of the IP core changes to show the I/O and the device resources the IP core will need.

7. Click **Generate**.

8. Click the **Generate Log** tab to check for warnings and error messages.

Chapter 5:

Support Resources

This chapter contains information about Lattice Technical Support, additional references, and document revision history.

Lattice Technical Support

There are a number of ways to receive technical support.

Online Forums

The first place to look is Lattice Forums (http://www.latticesemi.com/support/forums.cfm). Lattice Forums contain a wealth of knowledge and are actively monitored by Lattice Applications Engineers.

Telephone Support Hotline

Receive direct technical support for all Lattice products by calling Lattice Applications from 5:30 a.m. to 6 p.m. Pacific Time.

- For USA & Canada: 1-800-LATTICE (528-8423)
- For other locations: +1 503 268 8001

In Asia, call Lattice Applications from 8:30 a.m. to 5:30 p.m. Beijing Time (CST), +0800 UTC. Chinese and English language only.

- For Asia: +86 21 52989090

E-mail Support

- techsupport@latticesemi.com
- techsupport-asia@latticesemi.com

Local Support

Contact your nearest Lattice Sales Office.

Internet

www.latticesemi.com

References

- *An Economical Class of Digital Filter for Decimation and Interpolation*, Eugene B. Hogenauer, IEEE. Trans. ASSP, Vol. 29, No. 2, pp.155-162, April 1981.
- *CIC Filter Introduction*, Matthew P. Donadio, Iowegian, www.users.snip.net/~donadio/cic.pdf.

LatticeECP/EC

- HB1000, *LatticeECP/EC Family Handbook*

LatticeECP2M

- HB1003, *LatticeECP2M Family Handbook*

LatticeECP3

- HB1009, *LatticeECP3 Family Handbook*

LatticeSC/M

- DS1004, *LatticeSC/M Family Data Sheet*

LatticeXP

- HB1001, *LatticeXP Family Handbook*

LatticeXP2

- DS1009, *Lattice XP2 Data Sheet*

Revision History

Date	Document Version	IP Version	Change Summary
—	—	2.0	Previous Lattice releases.
September 2006	02.1	2.1	Added IPexpress User-Configurable Core section.
			Updated LatticeECP/EC appendix.
			Added LatticeECP2, LatticeSC and LatticeXP appendices.
December 2006	02.2	2.2	Updated appendices. Added support for LatticeECP2M device family.
May 2007	02.3	2.3	Added support for LatticeXP2 FPGA family.
			Updated LatticeXP and LatticeECP2M appendices.
November 2008	02.4	2.4	Updated appendices.
July 2010	02.5	3.0	Divided document into chapters. Added table of contents. Updated content for 3.0 version of IP core
			Added Quick Facts tables in Chapter 1, “Introduction.”
			Added new content in Chapter 3, “Parameter Settings.”
			Added new content in Chapter 4, “IP Core Generation.”
August 2010	2.6	3.1	Added support for Diamond software throughout.

Appendix A:

Resource Utilization

This appendix gives resource utilization information for Lattice FPGAs using the CIC Filter IP core.

IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the Diamond and ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and Diamond or ispLEVER help system. For more information on the Diamond or ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

LatticeECP and LatticeEC FPGAs

Table A-1. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	sysMEM™ EBRs	f_{MAX} (MHz)
Decimator, 8-bit data, 4 stages, 1 channel	166	216	301	0	212
Decimator, 16-bit data, 8 stages, 1 channel	837	1509	1253	0	109
Interpolator, 15-bit data, 7 stages, 4 channels	1205	898	1980	0	160

1. Performance and utilization data are generated using an LFECP33E-5F672C device, with Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP/EC family.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the CIC targeting LatticeECP/EC devices is CIC-FILTR-E2-U2. Table A-2 lists the parameter settings that are available for the CIC.

Table A-2. Parameter Settings of the Standard Configurations

Parameter Name	Config 1	Config 2	Config 3
ARC_INTERPOLATOR	No	No	Yes
ARC_INW	8	16	15
ARC_STAGE	4	8	7
ARC_CHANNEL	1	1	4
PROG_FIRDF	No	No	No
ARC_FIRDF	1	2	4
PROG_RATE=No	No	No	No
ARC_RATE=35	48	4096	35
PORT_CLEAR=No	No	No	No
PORT_CE=No	No	No	No
PORT_CHOUT=No	No	No	No
PORT_OBSTART=No	No	No	No
PORT_RFI=No	No	No	No
ARC_RATE_MIN=2	2	2	2

LatticeECP2 Devices

Table A-3. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	sysMEM EBRs	f_{MAX} (MHz)
Decimator, 8-bit data, 4 stages, 1 channel	168	219	301	0	272
Decimator, 16-bit data, 8 stages, 1 channel	878	1589	1253	0	144
Interpolator, 15-bit data, 7 stages, 4 channels	1238	985	1980	0	230

1. Performance and utilization data are generated using an LFE2-50E-7F672C device, with Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP2 family.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the CIC targeting LatticeECP2 devices is CIC-FILTR-P2-U2. Table A-2 lists the parameter settings that are available for the CIC.

LatticeECP2M Devices

Table A-4. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	sysMEM EBRs	f_{MAX} (MHz)
Decimator, 8-bit data, 4 stages, 1 channel	168	219	301	0	284
Decimator, 16-bit data, 8 stages, 1 channel	878	1589	1253	0	143
Interpolator, 15-bit data, 7 stages, 4 channels	1238	985	1980	0	234

1. Performance and utilization data are generated using an LFE2M-35E-7F672C device, with Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP2M family.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the CIC targeting LatticeECP2M devices is CIC-FILTR-PM-U2. Table A-2 lists the parameter settings that are available for the CIC.

LatticeECP3 Devices

Table A-5. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	sysMEM EBRs	f_{MHz} (MHz)
Decimator, 8-bit data, 4 stages, 1 channel	167	218	301	0	320
Decimator, 16-bit data, 8 stages, 1 channel	873	1588	1253	0	145
Interpolator, 15-bit data, 7 stages, 4 channels	1216	983	1980	0	237

1. Performance and utilization data are generated using an LFE3-95E-8FN672CES device, with Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP2M family.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the CIC targeting LatticeECP3 devices is CIC-FILT-E3-U2. Table A-2 lists the parameter settings that are available for the CIC.

LatticeSC/SCM Devices

Table A-6. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	sysMEM EBRs	f_{MAX} (MHz)
Decimator, 8-bit data, 4 stages, 1 channel	163	212	301	0	397
Decimator, 16-bit data, 8 stages, 1 channel	839	1510	1259	0	202
Interpolator, 15-bit data, 7 stages, 4 channels	1162	887	1981	0	284

1. Performance and utilization data are generated using an LFSC3GA25E-7F900C device, with Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeSC family.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the CIC targeting LatticeSC devices is CIC-FILTR-SC-U2. Table A-2 lists the parameter settings that are available for the CIC.

LatticeXP Devices

Table A-7. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	sysMEM EBRs	f_{MAX} (MHz)
Decimator, 8-bit data, 4 stages, 1 channel	166	216	301	0	196
Decimator, 16-bit data, 8 stages, 1 channel	837	1509	1253	0	103
Interpolator, 15-bit data, 7 stages, 4 channels	1205	898	1980	0	155

1. Performance and utilization data are generated using an LFXP20E-5F484C device, with Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. When using this IP core in a different density, speed, or grade within the LatticeXP family, performance and utilization may vary.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the CIC targeting LatticeXP devices is CIC-FILTR-XP-U2. Table A-2 lists the parameter settings that are available for the CIC.

LatticeXP2 Devices

Table A-8. Performance and Resource Utilization[1]

IPexpress User-Configurable Mode	SLICEs	LUTs	Registers	sysMEM EBRs	f_{MAX} (MHz)
Decimator, 8-bit data, 4 stages, 1 channel	168	219	301	0	285
Decimator, 16-bit data, 8 stages, 1 channel	878	1589	1253	0	152
Interpolator, 15-bit data, 7 stages, 4 channels	1238	985	1980	0	234

1. Performance and utilization data are generated using an LFXP2-17E-7F484C device, with Lattice Diamond 1.0 and Synplify Pro D-2009.12L-1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeXP2 family.

Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the CIC targeting LatticeXP2 devices is CIC-FILTR-X2-U2. Table A-2 lists the parameter settings that are available for the CIC.

LifeElectronics
Живое партнерство

ООО «ЛайфЭлектроникс» **“LifeElectronics” LLC**

ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

Тел: +7 (812) 336 43 04 (многоканальный)
Email: org@lifeelectronics.ru

www.lifeelectronics.ru