LogiCORE IP CIC Compiler v2.0

DS613 March 1, 2011 **Product Specification**

Introduction

The Xilinx LogiCORE™ IP CIC Compiler core provides the ability to design and implement Cascaded Integrator-Comb (CIC) filters.

Features

- Drop-in module for Virtex®-7 and Kintex™-7, Virtex®-6, Virtex-5, Virtex-4, Spartan®-6, Spartan-3/XA, Spartan-3E/XA, Spartan-3A/AN/3A DSP/XA FPGAs
- Decimation or interpolation
- Fixed or programmable rate change from 4 to 8192
- Three to six CIC stages
- One or two differential delays
- Support of signed, two's complement input data from 2 bits to 20 bits
- Full or limited precision output data
- Single or multi-channel support for up to 16 channels
- Hardware folding for small footprint implementations
- Optional mapping to XtremeDSP™ Slices
- Synchronous clear input
- Clock enable input
- Use with Xilinx CORE Generator™ software and Xilinx System Generator for DSP v13.1

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	Virtex-7 and Kintex-7, Virtex-6, Virtex-5, Virtex-4, Spartan-6, Spartan-3/XA, Spartan-3E/XA, Spartan-3A/3AN/3A DSP/XA
Supported User Interfaces	Not Applicable
Provided with Core	
Documentation	Product Specification
Design Files	Netlist
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	N/A
Simulation Model	VHDL and Verilog
Tested Design Tools	
Design Entry Tools	CORE Generator 13.1 System Generator for DSP 13.1
Simulation	Mentor Graphics ModelSim 6.6d Cadence Incisive Enterprise Simulator (IES) 10.2 Synopsys VCS and VCS MX 2010.06 ISIM 13.1
Synthesis Tools	N/A
Support	
Provided by Xilinx, Inc.	

[1] For a complete listing of supported devices, see the release notes for this core.

Overview

Cascaded Integrator-Comb (CIC) filters, also known as Hogenauer filters, are multi-rate filters often used for implementing large sample rate changes in digital systems. They are typically employed in applications that have a large excess sample rate. That is, the system sample rate is much larger than the bandwidth occupied by the processed signal as in digital down converters (DDCs) and digital up converters (DUCs). Implementations of CIC filters have structures that use only adders, subtractors, and delay elements. These structures make CIC filters appealing for their hardware-efficient implementations of multi-rate filtering.

Core Symbol and Port Definitions

Figure 1 and Table 1 illustrate and define the schematic symbol signal names. All control inputs are active high.

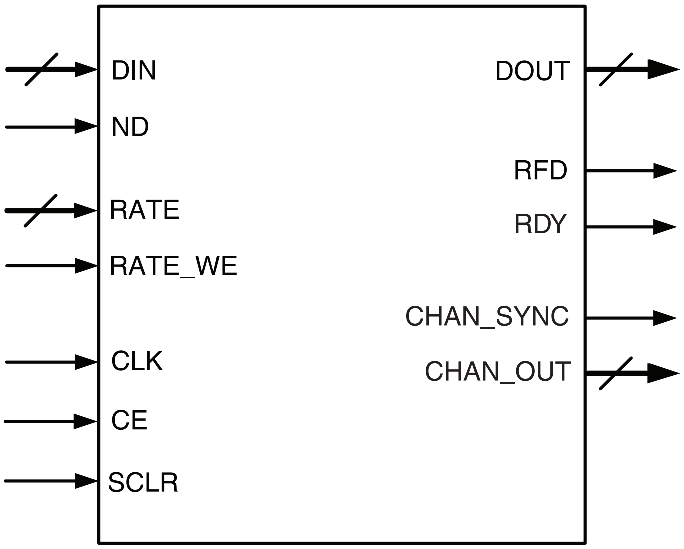


Figure 1: **Core Schematic Symbol**

Table 1: **Core Signal Pinout**

Name	Direction	Description
CLK	Input	Clock – active rising edge.
DIN[B-1:0]	Input	Data Input Port – filter input (B bits wide).
ND	Input	New Data (active high) – when this signal is asserted, the data sample presented on the DIN port is loaded into the filter.
DOUT[W-1:0]	Output	Data Output Port – filter output (W bits wide).
RDY	Output	Filter Output Sample Ready (active high) – indicates that a new filter output sample is available on the DOUT port.
RFD	Output	Ready for Data (active high) – indicates when the filter can accept a new input sample.
CE	Input	Clock Enable – core clock enable (active high).
SCLR	Input	Synchronous Clear – synchronous reset (active high). Asserting SCLR synchronously with CLK resets the filter internal state.
CHAN_SYNC	Output	Channel Synchronization – channel synchronization signal (active high). This is present only in multi-channel implementations.

Table 1: **Core Signal Pinout *(Cont'd)***

Name	Direction	Description
CHAN_OUT[$\log_2$(channels)-1:0]	Output	Channel Output – binary value that indicates the channel number for the current data output. This is present only in multi-channel implementations. ($\log_2$(channels) bits wide)
RATE[R-1:0]	Input	Rate – rate specification input binary value that specifies the rate change for interpolation or decimation. This is present only in programmable rate implementations. (R bits wide).
RATE_WE	Input	Rate Write Enable – enable signal (active high) to latch the rate specification value. This is present only in programmable rate implementations.

CORE Generator Graphical User Interface

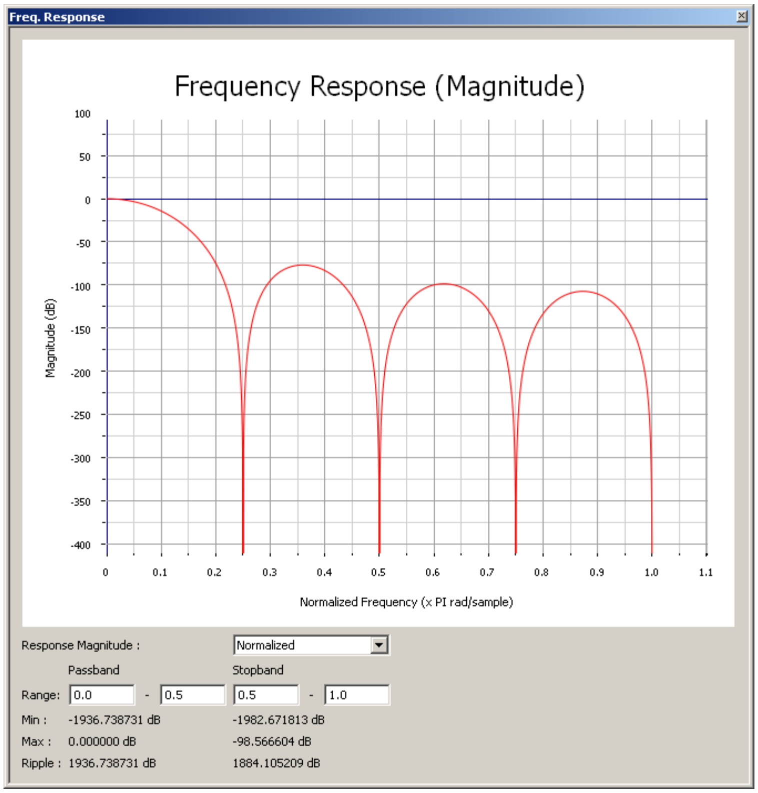
The CIC Compiler core GUI has three pages used to configure the core plus three informational/analysis tabs.

Tab 1: IP Symbol

The IP Symbol tab illustrates the core pinout.

Tab 2: Frequency Response

The Freq. Response tab (Figure 2), the default tab when the CORE Generator software is started, displays the filter frequency response (magnitude only). The content of the tab can be adjusted to fit the entire window or un-docked (as shown) into a separate window.



Response Magnitude : Normalized

	Passband		Stopband	
Range:	0.0	- 0.5	0.5	- 1.0
Min :	-1936.738731 dB		-1982.671813 dB	
Max :	0.000000 dB		-98.566604 dB	
Ripple :	1936.738731 dB		1884.105209 dB	

Figure 2: **Frequency Response tab**

It is important to note that the frequency axis in this plot is also normalized frequency as in other plots shown in this data sheet. Although the values in the GUI plot range from 0 to 1.0, they represent the same range of frequencies as in the other figures, that is, the range from 0 to 1/2 the sampling frequency. It is also important to note that the normalizing sampling frequency implied in the GUI plot depends on the type of filter. For a CIC decimator, the normalizing sampling frequency is the higher, input sampling frequency. For a CIC interpolator, the normalizing frequency is the higher, output sampling frequency.

- Response Magnitude: Specifies the magnitude scaling of the frequency response: Normalized; Full Precision (the absolute filter gain) and Output Quantization (the effective filter gain given the core output width). In previous versions of the core, the frequency response was always normalized. All plots shown in Theory of Operation, page 8, use normalized magnitude.
- **Passband Range:** Two fields are available to specify the passband range, the left-most being the minimum value and the right-most the maximum value. The values are specified in the same units as on the graph x-axis (for example, normalized to pi radians per second). For the specified range, the passband maximum, minimum and ripple values are calculated and displayed (in dB).
- **Stopband Range:** Two fields are available to specify the stopband range, the left-most being the minimum value and the right-most the maximum value. The values are specified in the same units as on the graph x-axis (for example, normalized to pi radians per second). For the specified range, the stopband maximum value is calculated and displayed (in dB).

Note: The user can specify any range for the passband or stopband, allowing closer analysis of any region of the response.

Tab 3: Resource Estimation

The number of DSP slices is displayed in addition to a count of the number of block RAM elements required to implement the design. Usage of general slice logic is not currently estimated.

Filter Specification

- **Component Name**: The name of the core component to be instantiated. The name must begin with a letter and be composed of the following characters: a to z, 0 to 9, and "_".
- **Filter Type**: The CIC core supports both interpolator and decimator types. When the filter type is selected as *decimator,* the input sample stream is down-sampled by the factor *R*. When an *interpolator* is selected, the input sample is up-sampled by *R*.
- **Number of Stages**: Number of integrator and comb stages. If *N* stages are specified, there will be *N* integrators and *N* comb stages in the filter. The valid range for this parameter is 3 to 6.
- **Differential Delay**: Number of unit delays employed in each comb filter in the comb section of either a decimator or interpolator. The valid range of this parameter is 1 or 2.
- **Number of Channels**: Number of channels to support in implementation. The valid range of this parameter is 1 to 16
- **Fixed/Programmable**: Type of rate change is fixed or programmable.
- **Fixed or Initial Rate**: Rate change factor (for fixed type) or initial rate change factor (for programmable type). For an interpolation filter, the rate change specifies the amount of up-sampling. For a decimator, it specifies the amount of down-sampling.
- **Minimum Rate**: Minimum rate change factor for programmable rate change.
- **Maximum Rate:** Maximum rate change factor for programmable rate change.
- **Hardware Oversampling Specification format:** Selects which format is used to specify the hardware oversampling rate, the number of clock cycles available to the core to process an input sample and generate an output. This value directly affects the level of parallelism in the core implementation and resources used. When "Frequency Specification" is selected, the user specifies the Input Sampling Frequency and Clock Frequency. The ratio between these values along with other core parameters determine the hardware

oversampling rate. When "Sample Period" is selected, the user specifies the integer number of clock cycles between input samples.

- **Input Sample Frequency:** This field can be an integer or real value. It specifies the sample frequency for one channel. The upper limit is set based on the clock frequency and filter parameters such as Interpolation Rate and number of channels.
- **Clock Frequency:** This field can be an integer or real value. The limits are set based on the sample frequency, interpolation rate and number of channels. **This field influences architecture choices only; the specified clock rate may not be achievable by the final implementation.**
- **Input Sample Period:** Integer number of clock cycles between input samples. When the multiple channels have been specified, this value should be the integer number of clock cycles between the time division multiplexed input sample data stream.

Implementation Options

- **Input Data Width:** Number of bits for input data. The valid range of this parameter is 2 to 20.
- **Quantization:** Type of quantization for limited precision output, Full Precision or Truncation. This quantization applies only to the output and is not applied in the intermediate stages of the CIC filter.
- **Output Data Width:** Number of bits for output data. The valid range of this parameter is up to 48 bits with the minimum value set to the input data width.
- **Use XtremeDSP Slice:** Use DSP hardware primitive slices in the filter implementation.
- **Use Streaming Interface:** Specifies if a streaming interface is used for multiple channel implementations. Refer to the Control Signals and Timing, page 17, for further details.
- **ND:** Specifies if the core has a New Data signal.
- **SCLR:** Specifies if the core has a synchronous clear signal.
- **CE:** Specifies if the core has a clock enable signal.

Summary

In addition to all the parameterization values of the core, the summary page displays:

- **Bits per Stage:** The number of bits used in each of the stages of the CIC filter implementation. These numbers are computed based on the register growth analysis presented in [Ref 1].
- **Latency:** The input to output latency in the CIC core implementation.

Using the CIC Compiler IP Core

The CORE Generator GUI performs error-checking on all input parameters. Resource estimation and optimum latency information are also available.

Several files are produced when a core is generated, and customized instantiation templates for Verilog and VHDL design flows are provided in the `.veo` and `.vho` files, respectively. For detailed instructions, see the CORE Generator software documentation.

Simulation Models

The core has a number of options for simulation models:

- VHDL behavioral model in the xilinxcorelib library
- VHDL UniSim-based structural simulation model
- Verilog UniSim-based structural simulation model

The models required may be selected in the CORE Generator software project options.

Xilinx recommends that simulations utilizing UniSim-based structural models are run using a resolution of 1 ps. Some Xilinx library components require a 1 ps resolution to work properly in either functional or timing simulation. The UniSim-based structural simulation models may produce incorrect results if simulated with a resolution other than 1 ps. See the "Register Transfer Level (RTL) Simulation Using Xilinx Libraries" section in *Chapter 6 of the Synthesis and Simulation Design Guide* for more information. This document is part of the ISE® Software Manuals set available at www.xilinx.com/support/documentation/dt_ise.htm.

XCO Parameters

Table 2 defines valid entries for the XCO parameters. Parameters are not case sensitive. Default values are displayed in bold. Xilinx strongly suggests that XCO parameters are not manually edited in the XCO file; instead, use the CORE Generator GUI to configure the core and perform range and parameter value checking. The XCO parameters are useful for defining the interface to other Xilinx tools.

Table 2: **XCO Parameters**

XCO Parameter	Valid Values
Component_Name	ASCII text using characters: a..z, 0..9 and '_' ; starting with a letter
Filter_Type	**Interpolation**, Decimation
Number_Of_Stages	**3**,4,5,6
Differential_Delay	**1**,2
Number_Of_Channels	1 - 16 Default value is **1**
Sample_Rate_Changes	**Fixed**, Programmable
Fixed_Or_Initial_Rate	4 - 8192 Default value is **4**
Minimum_Rate	4 - 8191
Maximum_Rate	5 - 8192
RateSpecification	**Frequency_Specification**, Sample_Period
Input_Sample_Frequency	0.000001 - 600.0
Clock_Frequency	0.000001 - 600.0
SamplePeriod	1 - 10000000
Input_Data_Width	2 - 20 Default is **18**
Quantization	**Full_Precision**, Truncation
Output_Data_Width	2 - 104
Use_Xtreme_DSP_Slice	**false**, true
Use_Streaming_Interface	**true**, false
ND	**false**, true
CE	**false**, true
SCLR	**false**, true

The CORE Generator software core update feature may be used to update an existing CIC Compiler XCO file to version 2.0 of the CIC Compiler core. The core may then be regenerated to create a new netlist. See the CORE Generator software documentation for more information on this feature.

Port Changes

There are no differences in port naming conventions, polarities, priorities or widths between versions.

Updating from CIC Compiler v1.1 and v1.2

Latency Changes

CIC Compiler v2.0 generally has a lower latency than previous versions of the core. To verify the core latency, the updated XCO file may be loaded into CORE Generator system and the new latency value found on page 3 of the GUI.

Updating from CIC Compiler v1.3

Timing Changes

Following the application of a new rate change value, using the `RATE_WE` and `RATE` signals, the core updates to the new rate on the next input sample, for a single channel implementation, or the next input to the first channel, for multiple channel implementations. The update now causes the core internal state and output signals to be reset.

Core Use through System Generator

The CIC Compiler core is available through Xilinx System Generator for DSP, a design tool that enables the use of the model-based design environment, Simulink® product for FPGA design. The CIC Compiler core is one of the DSP building blocks provided in the Xilinx blockset for Simulink. The core can be found in the Xilinx Blockset in the DSP section. The block is called “CIC Compiler v2.0.” See the System Generator User Manual for more information.

System Generator for DSP Graphical User Interface

This section describes each tab of the System Generator for DSP GUI and details the parameters that differ from the CORE Generator GUI. See CORE Generator Graphical User Interface, page 3, for detailed information about all other parameters.

Tab 1: Filter Specification

The Filter Specification tab is used to define the basic filter configuration as on the Filter Specification, page 4, of the CORE Generator GUI.

- **Hardware Oversampling Specification format:** Selects which method is used to specify the hardware oversampling rate. This value directly affects the level of parallelism of the core implementation and resources used. When “Maximum Possible” is selected, the core uses the maximum oversampling given the sample period of the signal connected to `DIN` port. When “Hardware Oversampling Rate” is selected, the user can specify the oversampling rate. When “Sample Period” is selected, the core clock is connected to the system clock, and the value specified for the Sample Period parameter sets the input sample rate the core supports. The Sample Period parameter also determines the hardware oversampling rate of the core. When “Sample Period” is selected, the core is forced to use the `ND` control port. Refer to Control Signals and Timing, page 17, for more details on the core control ports.
- **Sample Period:** Specifies the input sample period supported by the core.
- **Hardware Oversampling Rate:** Specifies the hardware oversampling rate to be applied to the core.

Tab 2: Implementation Options

The Implementation tab is used to define implementation options. Refer to Implementation Options, page 5, of the CORE Generator GUI for details of all the core parameters on this tab.

- **rst:** Specifies if the core has a reset pin (the equivalent of selecting the SCLR option in the CORE Generator GUI).
- **nd:** This parameter is not available in System Generator. This control pin is available only when "Sample Period" has been selected for the Hardware Oversampling Specification format.
- **en:** Specifies if the core has a clock enable pin (the equivalent of selecting the CE option in the CORE Generator GUI).
- **FPGA Area Estimation:** See the System Generator documentation for detailed information about this option.

Theory of Operation

The following description of the CIC decimator and interpolator is based closely on that provided in [Ref 1]. The general concept of a CIC filter is the low-pass response that results from filtering an input signal with a cascade of *N* unit-amplitude, rectangular windows of length *R*M*. The system response of such filter is

$$H(z) = [\sum_{k=0}^{R*M-1} z^{-k}]^N$$

or *Equation 1*

$$H(z) = \frac{(1-z^{-R*M})^N}{(1-z^{-1})^N}$$

where

N is the number of CIC stages

R is the rate change (decimation or interpolation)

M is the *differential delay* in the comb section stages of the filter

The implementation of this filter response with a clever combination of comb filter sections, integrator sections, and up-sampling (for interpolation) and down-sampling (for decimation) gives rise to the hardware-efficient implementation of CIC filters.

Frequency Response Characteristics

The frequency response of a CIC filter is obtained by evaluating Equation 1 at:

$$z = e^{2j\pi f} \qquad \text{Equation 2}$$

Where *f* is the discrete-time frequency, normalized to the *higher* frequency in a rate changing filter - input sampling frequency in a CIC decimation, or output sampling frequency in a CIC interpolator. Evaluating Equation 1 in the z-plane at the sample points defined by Equation 2 gives a magnitude frequency response as shown in Equation 3.

$$|H(f)| = \left[\frac{\sin(\pi RMf)}{\sin(\pi f)}\right]^N \qquad \text{Equation 3}$$

This magnitude response is low-pass. In the design process of a CIC filter implementation, the parameters *R*, *M*, and *N* are selected to provide adequate passband characteristics over the frequency range from zero to a predetermined cutoff frequency *fc*. This passband frequency range is typically the bandwidth of interest occupied by the signal undergoing processing by the CIC filter. Figure 3 shows the frequency response of a 3-stage (*N* = 3) CIC filter with unity differential delay (*M* = 1) and a sample rate change *R* = 7.

According to Equation 3 and as seen in Figure 3, there are nulls in the magnitude response (transfer function zeros) at integer multiples of $f = 1/(RM)$. Thus, the differential delay parameter, *M*, can be used as a design parameter to control the placement of the nulls.

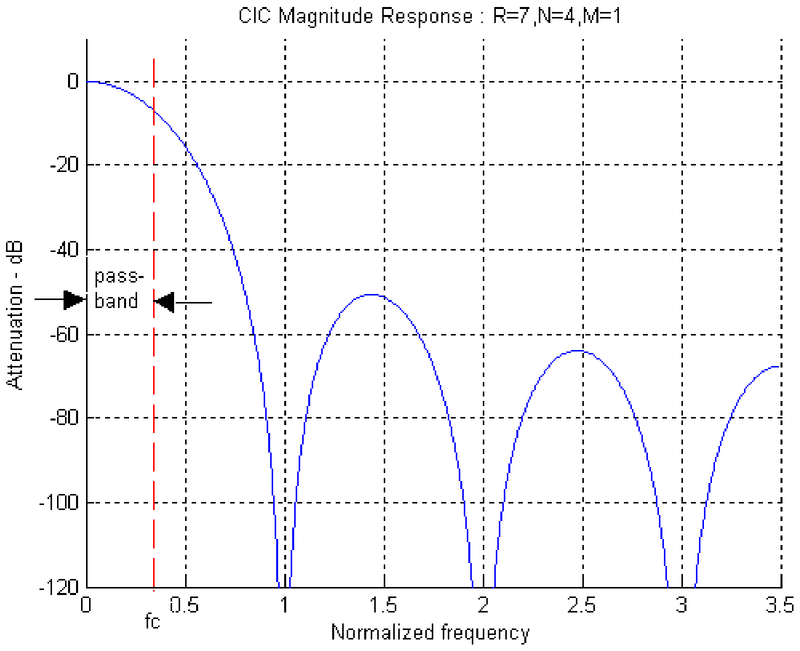


Figure 3: **CIC Magnitude Response**

Figure 4 shows the effect of the differential delay *M* on the magnitude response of a filter with three stages ($N = 3$) and a sample rate change $R = 7$. Besides the effect on the placement of the response nulls, increasing *M* also increases the amount of attenuation in side lobes of the magnitude response.

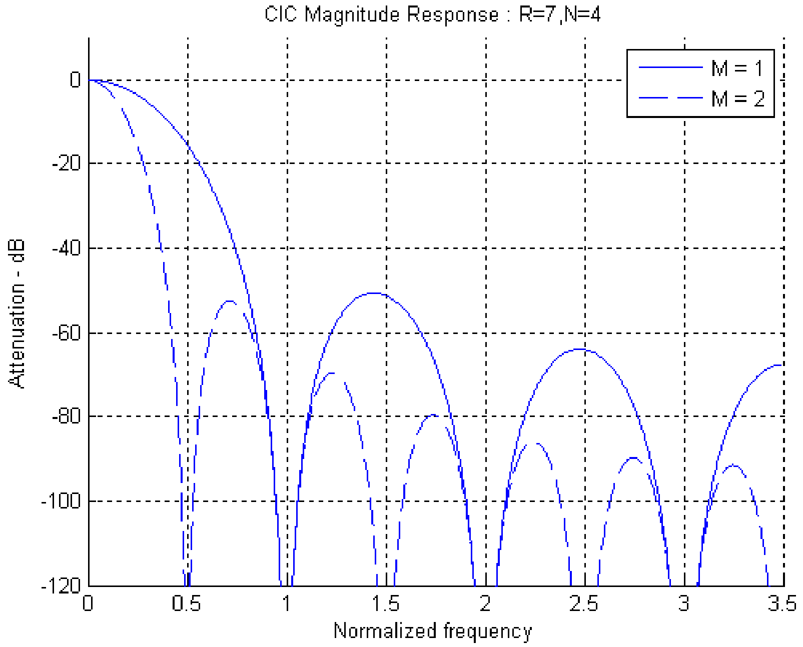


Figure 4: **CIC Magnitude Response – Effect of Differential Delay M**

The rate change parameter *R* can also be used to control the frequency response of the CIC filter. The effect of *R* on the magnitude response can be seen in Figure 5. In essence, increasing the rate change increases the length of the cascaded unit-amplitude, rectangular window of length *R*M*. This results in an increase in attenuation and decrease of the width of the response side lobes.

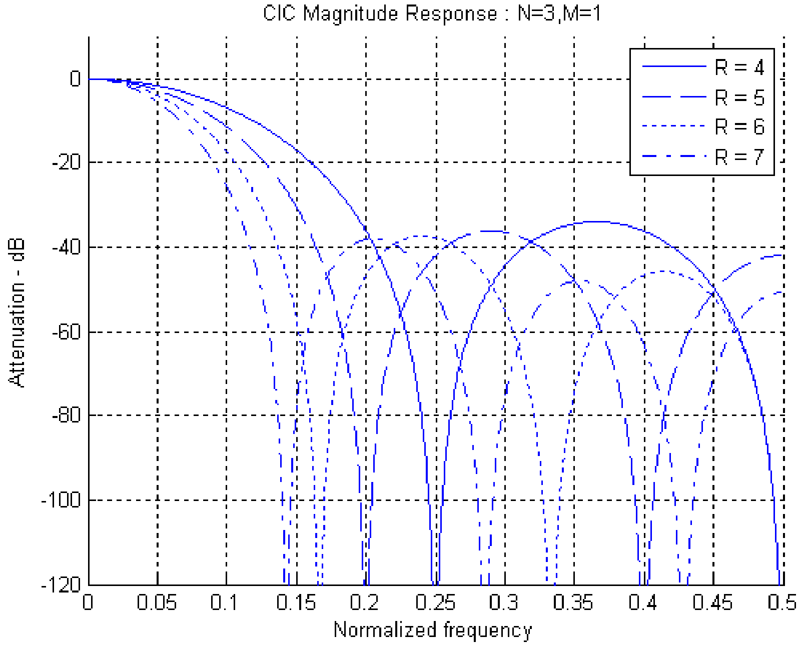


Figure 5: **CIC Magnitude Response – Effect of Rate Change R**

The number of stages parameters, *N*, can also be used to affect the CIC filter magnitude response. This effect can be understood from the fundamental concept of a cascade of *N* filtering stages, each with an impulse response of a unit-amplitude, rectangular window. The larger the number of cascaded stages, the more attenuated the magnitude response side lobes become. This can be seen in Figure 6.

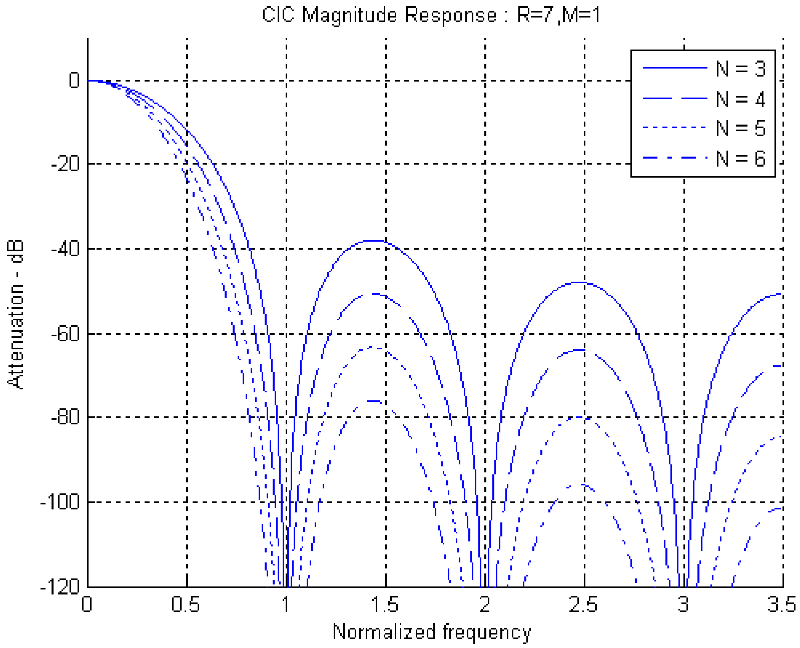


Figure 6: **CIC Magnitude Response – Effect of Number of Stages N**

Increasing *N* has the effect of increasing the order of the zeros in the frequency response. This, in turn, increases the attenuation at frequencies in the locality of the zero. This effect is clearly illustrated in Figure 6 where we see increasing attenuation of the filter side lobes as *N* is increased.

As the order of the zeros increase, the passband droop also increases, thus narrowing the filter bandwidth. The increased droop may not be acceptable in some applications. The droop is frequently corrected using an additional (non-CIC-based) stage of filtering after the CIC decimator. In the case of a CIC interpolator, the signal may be pre-compensated to account for the impact in the passband as the signal is up-sampled by the CIC interpolator.

A compensation filter (not part of the CIC compiler) can be used to flatten the passband frequency response. For a CIC decimator, the compensation filter operates at the decimated sample rate. The compensation filter provides $(x/\sin(x))^N$ shaping. An example of a third order ($N = 3$) $R = 64$ compensated CIC system is shown in Figure 7. The plot shows the uncompensated CIC frequency response, the compensation filter frequency response, and the compensated CIC. In this case, since the number of CIC stages is three, the compensation filter has a cubic response of the form $(x/\sin(x))^3$.

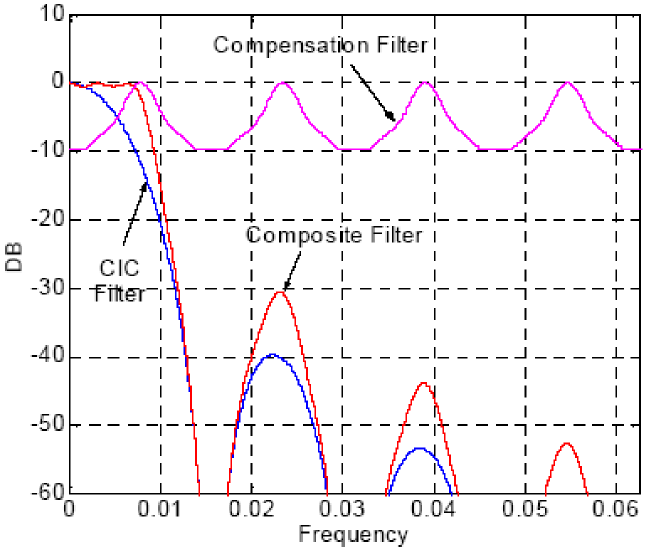


Figure 7: **CIC Droop Compensation**

The compensation filter coefficients employed were [-1, 4, 16, 32, –64, 136, –352, 1312, –352, 136, –64, 32, –16, 4, -1]. Figure 8 provides an exploded view of the compensated filter passband.

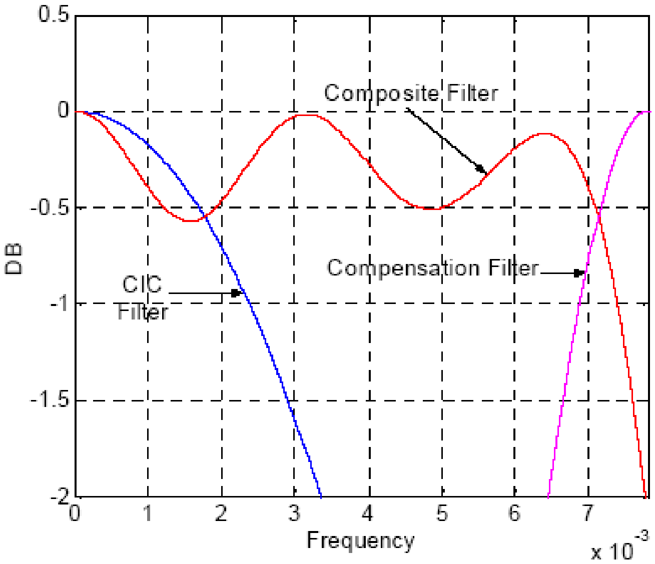


Figure 8: **CIC Droop Compensation – Exploded View**

CIC Decimator

When the output of the filter given by Equation 1 is decimated (down-sampled) by a factor *R*, the response of the filter referenced to the lower, down-sampled output rate is expressed in Equation 4 as:

$$H(z) = \frac{(1 - z^{-M})^N}{(1 - z^{-1})^N} \qquad \text{Equation 4}$$

This response can be viewed as a cascade of *N* integrators and *N* comb filters.

$$H(z) = \frac{1}{(1 - z^{-1})^N} * (1 - z^{-M})^N \qquad \text{Equation 5}$$

A block diagram of a realization of this response can be seen in Figure 9. There are two sections to the CIC decimator filter: an integrator section with *N* integrator stages that processes input data samples at a sampling rate *fs,* and a comb section that operates at the lower sampling rate *fs* / *R*. This comb section consists of *N* comb stages with a differential delay of *M* samples per stage. The down sampling operation decimates the output of the integrator section by passing only every *Rth* sample to the comb section of the filter.

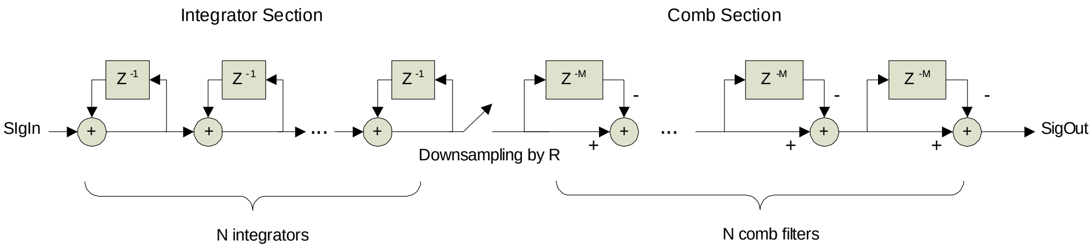


Figure 9: **CIC Decimation Filter**

Referring back to Figure 3, when the CIC filter is employed as a decimator, the frequency bands in the interval

$$\frac{k}{RM} \pm f_c, k = 1,2,\ldots\left\lfloor \frac{R}{2} \right\rfloor \qquad \text{Equation 6}$$

alias back into the filter passband. Care must be taken to ensure that the integrated side lobe levels do not impact the intended application. Figure 10 shows an example of a CIC decimator response prior to down-sampling to help illustrate the effect of aliasing. In Figure 10, the ideal response of a decimator with sampling rate change of *R* = 8, number of stages *N* = 3, and differential delay *M* = 1 is shown. The spectrum of the decimator input is also shown containing energy in the intended passband (low frequencies up to a cutoff frequency fc = 1/32 cycles/sample) and in the stopband (around 1/4 cycles/sample). The output of the decimator (without down-sampling) is shown to demonstrate the attenuation produced by this CIC filter. The dashed vertical lines in Figure 10 indicate the frequency ranges that alias to the passband when down-sampling. In this figure, the frequency axis is normalized to the (higher) sampling frequency prior to down-sampling.

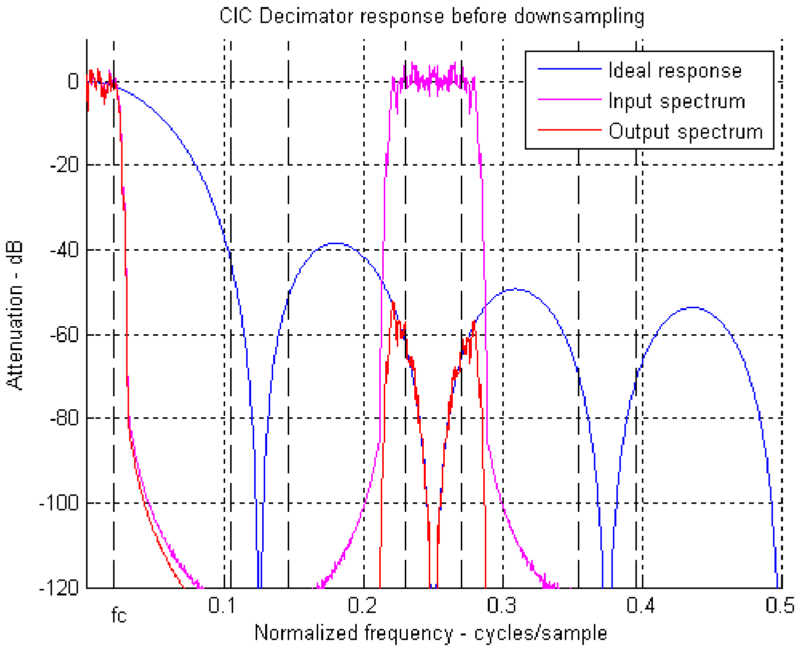


Figure 10: **CIC Decimator Response before Down-sampling**

Figure 11 shows the output spectrum of the CIC decimator example. In Figure 11, the frequency axis is normalized relative to the lower sampling rate obtained after down-sampling. Because of this re-normalization of frequencies, the plots in Figure 11 can be conceptualized as a zoomed view of the frequency range from 0 to 1/(2*R) = 1/16 cycles/sample of Figure 10.

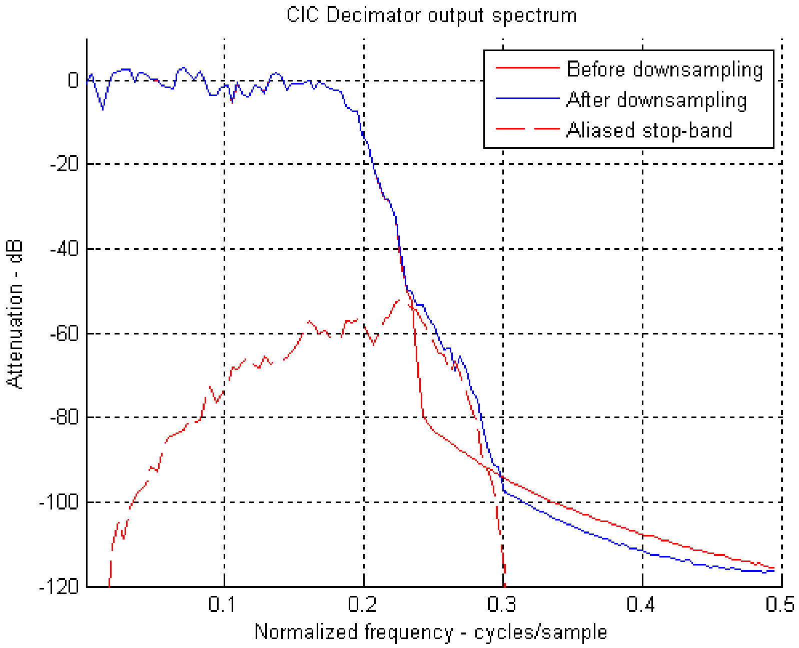


Figure 11: **CIC Decimator Output Spectrum**

The important points to note from Figure 11 are the following:

- The solid red plot shows the CIC output spectrum if no aliasing occurred.
- The dashed red plot shows the stopband output spectrum when aliased due to down-sampling. This aliased spectrum affects the final output of the CIC decimator by contributing additively to the output spectrum.
- The solid blue plot is the actual output of the CIC decimator which clearly shows the contribution of the aliased spectrum from down-sampling.

Again, care must be taken to ensure that the CIC decimator parameters are properly chosen to avoid detrimental effects from aliasing.

Pipelined CIC Decimator

To support high system clock frequencies, the CIC decimator is implemented using the pipelined architecture shown in Figure 12.

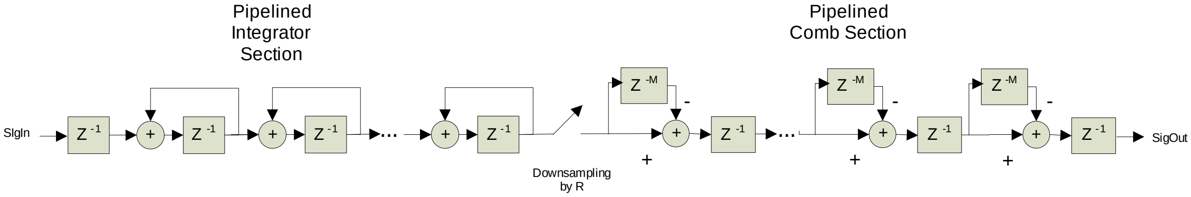


Figure 12: **Pipelined CIC Decimator**

Register Growth in CIC Decimator

The CIC data path undergoes internal register growth that is a function of all the design parameters: *N, M, R* in addition to the input sample precision *B*. As shown in [Ref 1], the output bit width of a CIC decimator with full precision is given by

$$B_{\max} = \lceil N \log_2 RM + B \rceil \qquad \textit{Equation 7}$$

where ⌈ ⌉ denotes the ceiling operator. The CIC compiler supports both full and limited precision output. For full precision, the CIC decimator implementation uses B_{max} bits internally for each of the integrator and differentiator stages. This introduces no quantization error at the output. For limited precision (that is, output bit width less than B_{max}), the registers in the integrator and comb stages are sized to limit the quantization noise variance at the output as described in [Ref 1]. Consequently, the hardware resources in a CIC decimator implementation can be reduced when using limited precision output at the cost of quantization noise. This ability to trade off resources and quantization noise is important to achieve an optimum implementation.

CIC Interpolator

The structure for a CIC interpolator filter is shown in Figure 13. This structure is similar to that of a CIC decimator with the integrator and comb sections in reverse order. In this case, there is an up-sampling of data by a factor, *R*, between the comb and integrator sections. This rate increase is done by inserting *R-1* zero-valued samples between consecutive samples of the comb section output. The up-sampled and filtered data stream is output at the sample rate *fs*.

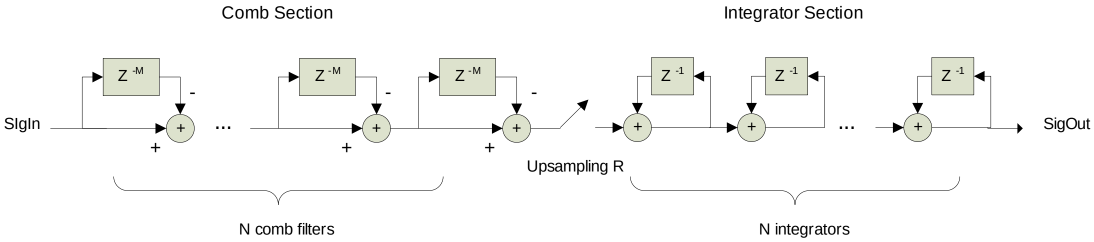


Figure 13: **CIC Interpolator**

For interpolation, the response of the CIC filter is applied to the up-sampled (zero-valued samples inserted) input signal. The effect of this processing is shown in Figure 14 in a filter with rate change *R* = 7, number of stages *N* = 4, and differential delay *M* = 1. The peaks in the output interpolated signal show the effect of the magnitude response of the CIC filter applied to the spectrum images of the up-sampled input signal.

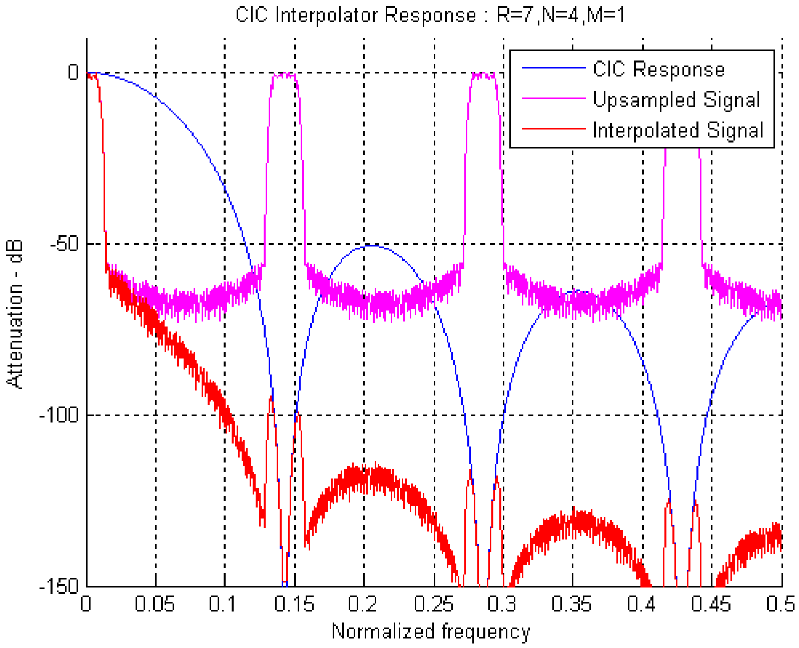


Figure 14: **CIC Interpolator Response**

Pipelined CIC Interpolator

Similarly to the CIC decimator, the CIC interpolator core implementation uses a pipelined structure to support high system clock frequencies. This pipelined structure is shown in Figure 15.

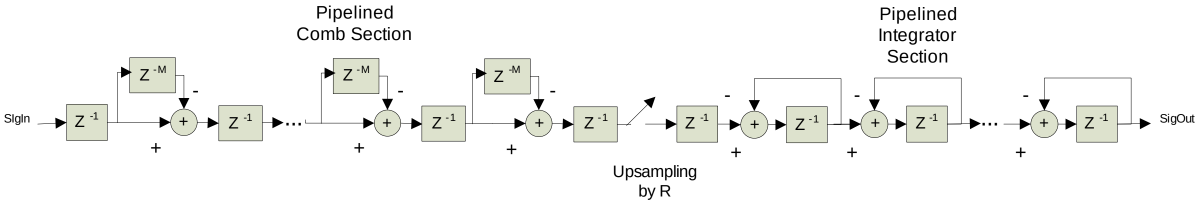


Figure 15: **Pipelined CIC Interpolator**

Register Growth in CIC Interpolator

The data path in a CIC interpolator also undergoes internal register growth that is a function of all the design parameters: N, M, R, in addition to the input sample precision B. As shown in [Ref 1], the registers in the comb and integrator sections grow monotonically with the maximum register size occurring at the output of the last stage (output of the CIC filter). The maximum register width is given by Equation 8:

$$B_{\max} = \left\lceil \log_2 \frac{(RM)^N}{R} + B \right\rceil \qquad \textit{Equation 8}$$

where ⌈ ⌉ denotes the ceiling operator. The CIC compiler always sizes the internal stage registers according to the register growth as described in [Ref 1]. The output of the filter can be selected to be full or limited precision (with truncation or rounding) to accommodate an output width specific to an application. Using limited precision does not affect the internal register sizes and only the final stage output is scaled, and rounded if desired, to provide the selected output width.

Output Width and Gain

As illustrated by Equation 7 and Equation 8, the gain of a CIC filter is a function of all the key design parameters. When the output width is equal to the maximum register width, the core outputs the full precision result and the magnitude of the core output reflects the filter gain. When the output width is set to less than the maximum register width, the output is truncated with a corresponding reduction in gain.

When the core is configured to have a programmable rate change, there is a corresponding change in gain as the filter rate is changed. When the output is specified to full precision, the change in gain is apparent in the core output magnitude as the rate is changed. When the output is truncated, the core shifts the internal result, given the B_{max} for the current rate change, to fully occupy the output bits.

Control Signals and Timing

The CIC filter employs a data-flow style interface for supplying input samples to the core and for reading the filter output port. ND (New Data), RFD (Ready For Data), and RDY (Ready) are used to coordinate I/O operations. The core output status signal RFD signals to the system that the filter is ready for data. RFD is active high. Asserting ND High indicates to the core the availability of a new input sample on the DIN port. The RDY output signal indicates that a new filter output sample is available on the DOUT port.

The interface signals are typically used in the following manner: The user system first waits for RFD = 1, which signals that a new input sample can be written to the filter. The new input sample is placed on the DIN port and ND is placed in the active state (ND = 1) for a single clock cycle. Asserting ND indicates to the core that it should sample the DIN port. The filter samples DIN on the rising edge of the clock (*CLK*) qualified with ND = 1. A filter read operation can occur when the core asserts RDY = 1. RDY can be used as a new data signal for a down-stream processing block that is consuming the filter output samples.

For multiple-channel implementations, the CIC Compiler core supports time-multiplexed input and output. The filter input data in the DIN port is expected to have an ordered, time-multiplexed format. The core produces time-multiplexed output data on the DOUT port. Two additional ports are included in multi-channel implementation. The CHAN_SYNC port signal indicates the time of the output corresponding to the first channel in the time-multiplexed stream. The CHAN_OUT port indicates the time for each channel output in the time-multiplexed steam.

For programmable rate implementations, the RATE and RATE_WE input ports allow control of the rate change in the CIC filter core. The RATE port is sampled when RATE_WE is asserted high. The core uses the new RATE value on the next input sample, for a single channel implementation, or the next input to the first channel, for multiple channel implementations.

Decimator

The timing for a CIC decimator with a down-sampling factor $R = 4$ is shown in Figure 16. In this example, the core is not oversampled and can accept a new input sample on every clock edge. Some number of clock cycles after the first input sample has been written to the filter, RDY is asserted by the filter to indicate that the first output sample is available. This time interval is a function of the down-sampling factor R and a fixed latency that is related to internal pipeline registers in the core. The number of pipeline stages depends on the core parameters. After the first output sample has been produced, subsequent outputs are available every R clock cycles. It is strongly recommended that designers employ the RDY signal as a qualifying signal for any processes that consume the filter output samples.

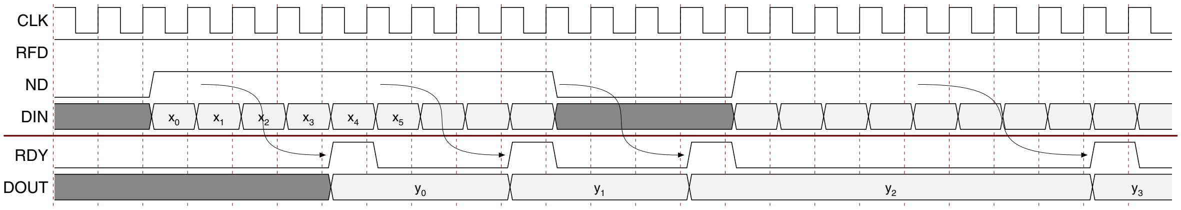


Figure 16: **CIC Decimator – Fixed Rate, Single Channel**

Figure 17 shows the timing for the same filter configuration with an input sample period of 3.

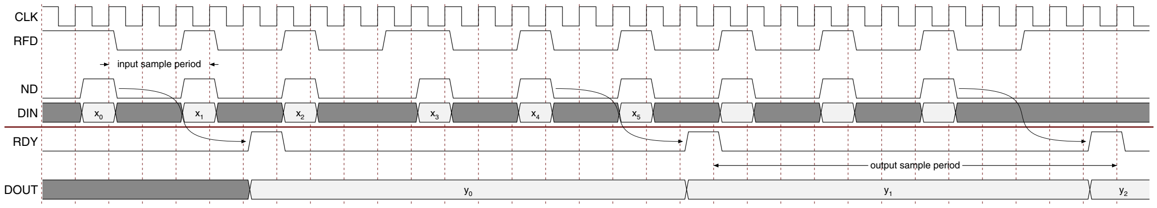


Figure 17: **CIC Decimator – Fixed Rate, Single Channel, Oversampled**

Figure 18 shows the timing for a multi-channel CIC decimator with a rate change $R = 4$. In this example the decimator filter handles three channels of data and is configured to use the block-based interface. The input to the decimator DIN shows the time-multiplexed samples with labels to indicate the corresponding channel number. The output of the decimator DOUT shows the time-multiplexed data.

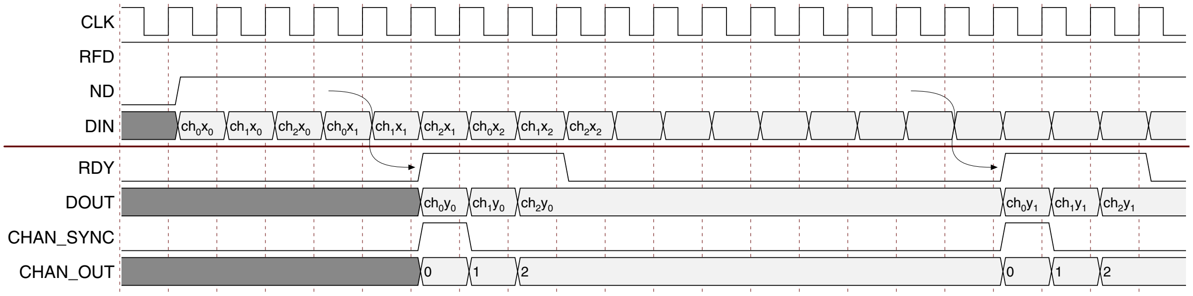


Figure 18: **CIC Decimator – Fixed Rate, Multi-Channel, Block interface**

Figure 19 shows the timing for the same filter configuration using the streaming interface.

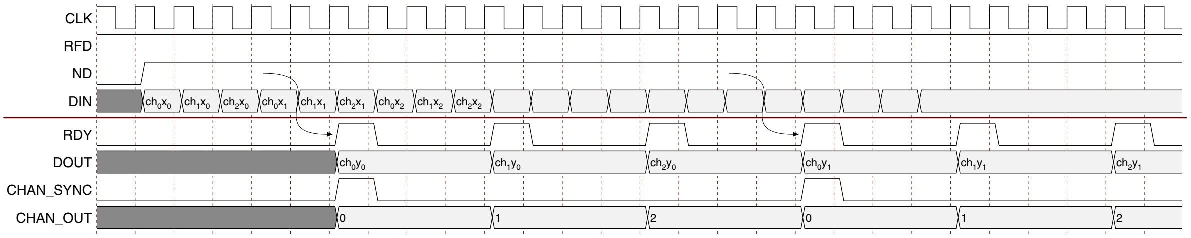


Figure 19: **CIC Decimator – Fixed Rate, Multi-Channel, Streaming interface**

Figure 20 shows the timing for a CIC decimator with programmable rate. In the timing diagram, the decimator is shown with an initial down-sampling rate value of 4. After some time, the down sampling rate is changed to 7 by setting the value on the RATE port to 7 and asserting RATE_WE.

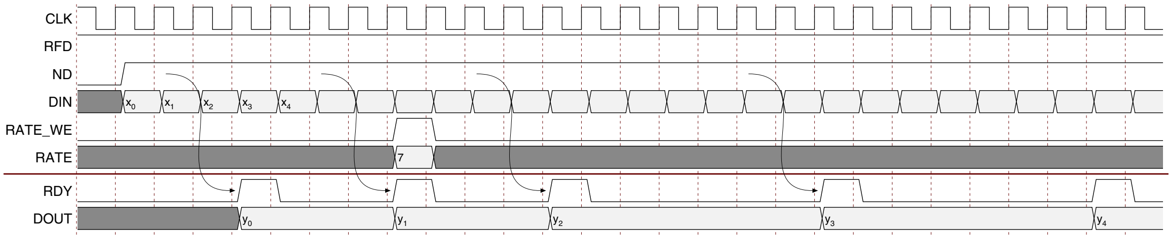


Figure 20: **CIC Decimator with Programmable Rate**

Interpolator

Figure 21 shows the timing for a CIC interpolator with an up-sampling factor R = 4. A new input sample can be accepted by the core every 4th cycle of the clock. After the initial start-up latency, RDY is asserted, and a new filter output is available on every subsequent clock edge. For every input delivered to the filter core, four output samples are generated.

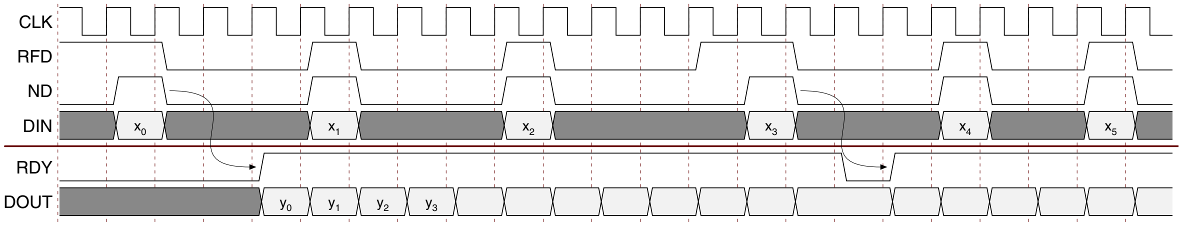


Figure 21: **CIC Interpolator – Fixed Rate, Single Channel**

Figure 22 shows the same filter configuration with an input sample period of 8.

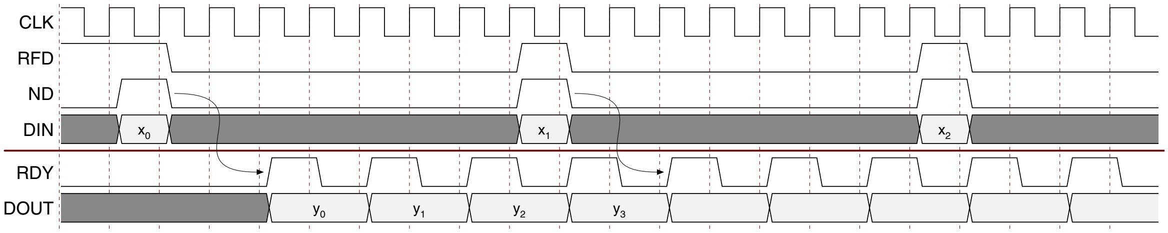


Figure 22: **CIC Interpolator – Fixed Rate, Single Channel, Oversampled**

Figure 23 shows the timing for a multi-channel CIC interpolator with a rate change R = 4. In this example the interpolator filter handles two channels of data and uses the block-based interface. The input DIN shows the time-multiplexed samples with labels to indicate the corresponding channel number. The output DOUT shows the time-multiplexed data samples.

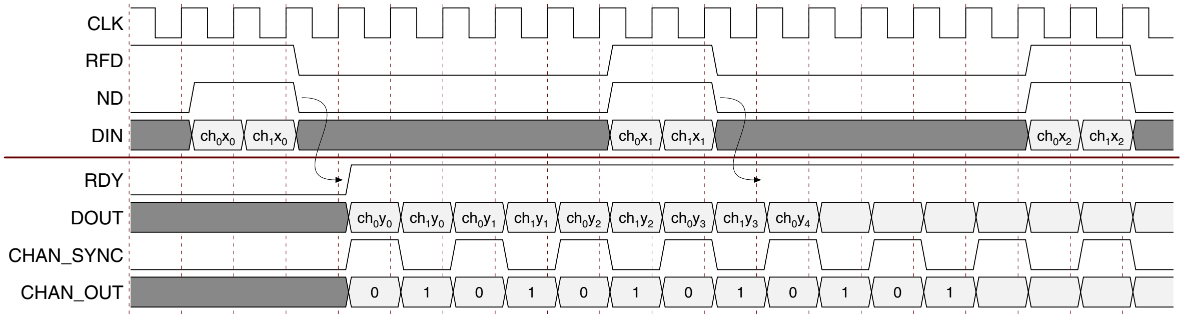


Figure 23: **CIC Interpolator – Fixed Rate, Multi-Channel, Block interface**

Figure 24 shows the same filter configuration using the streaming interface.

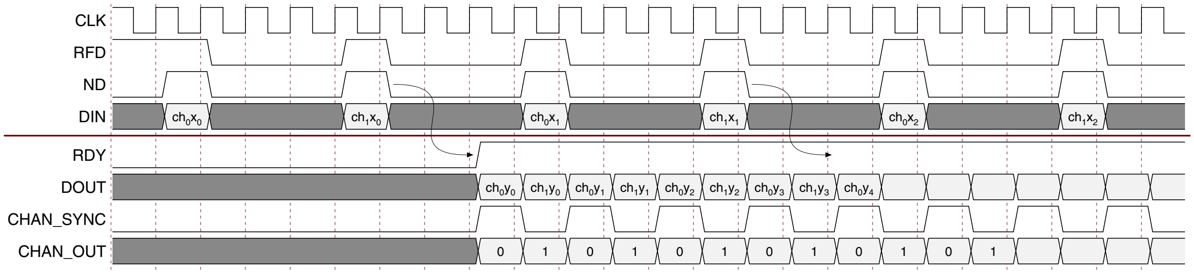


Figure 24: **CIC Interpolator – Fixed Rate, Multi-Channel, Streaming interface**

Figure 25 shows the timing for a CIC interpolator with programmable rate. In the timing diagram, the interpolator is shown with an initial up-sampling rate value of 4. After some time, the up-sampling rate is changed to 7 by setting the `RATE` port to 7 and asserting `RATE_WE`.

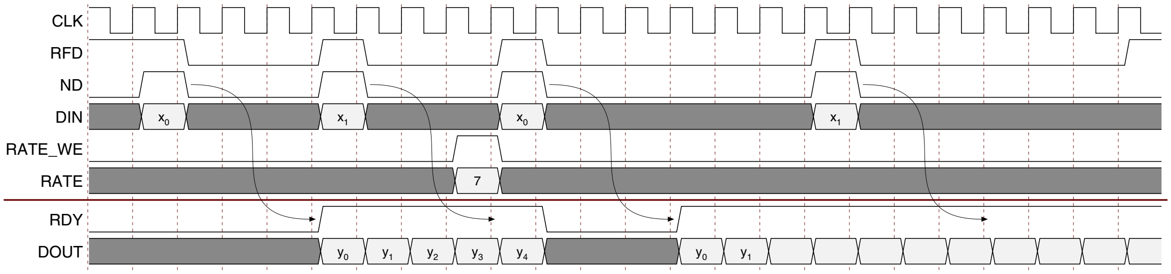


Figure 25: **CIC Interpolator with Programmable Rate**

Performance and Resource Utilization

Table 3 through Table 10 provide performance and resource usage information for a number of different filter configurations.

The maximum clock frequency results were obtained by double-registering input and output ports to reduce dependence on I/O placement. The inner level of registers used a separate clock signal to measure the path from the input registers to the first output register through the core.

The resource usage results do not include the preceding "characterization" registers and represent the true logic used by the core to implement a single multiplier. LUT counts include SRL16s or SRL32s (according to device family).

The map options used were: "`map -ol high`"

The par options used were: "`par -ol high`"

Clock frequency does not take clock jitter into account and should be derated by an amount appropriate to the clock source jitter specification.

The maximum achievable clock frequency and the resource counts may also be affected by other tool options, additional logic in the FPGA device, using a different version of Xilinx tools, and other factors.

CIC Decimator

The Virtex-6 FPGA test cases in Table 3 used ISE speed file version “ADVANCED 1.01e 2009-07-13.”

Table 3: **CIC Decimator: Virtex-6 XC6VLX75T-1-FF784**

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Input Width										
Decimation	4	3	1	2	8	1	1	55	4	452
Decimation	4	3	1	4	10	1	1	63	4	452
Decimation	4	3	1	8	14	1	1	77	4	452
Decimation	4	3	1	12	18	1	1	93	4	452
Decimation	4	3	1	16	22	1	1	109	4	452
Decimation	4	3	1	20	26	1	1	121	4	452
Varying Stages										
Decimation	4	3	1	18	24	1	1	113	4	452
Decimation	4	4	1	18	26	1	1	122	5	452
Decimation	4	5	1	18	28	1	1	151	7	452
Decimation	4	6	1	18	30	1	1	202	8	452
Varying Rate										
Decimation	4	3	1	18	24	1	1	113	4	452
Decimation	8	3	1	18	27	1	1	130	4	452
Decimation	16	3	1	18	30	1	1	135	4	452
Decimation	32	3	1	18	33	1	1	153	4	452
Decimation	64	3	1	18	36	1	1	170	4	452
Decimation	128	3	1	18	39	1	1	187	4	452
Decimation	256	3	1	18	42	1	1	192	4	452
Varying Differential Delay										
Decimation	4	3	1	4	10	1	1	63	4	452
Decimation	4	3	2	4	13	1	1	76	4	452
Decimation	4	3	1	8	14	1	1	77	4	452
Decimation	4	3	2	8	17	1	1	87	4	452
Decimation	4	3	1	12	18	1	1	93	4	452
Decimation	4	3	2	12	21	1	1	106	4	452
Decimation	4	3	1	16	22	1	1	109	4	452
Decimation	4	3	2	16	25	1	1	119	4	452
Decimation	4	3	1	20	26	1	1	121	4	452
Decimation	4	3	2	20	29	1	1	137	4	452

Table 3: **CIC Decimator: Virtex-6 XC6VLX75T-1-FF784** ***(Cont'd)***

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Channels										
Decimation	4	3	1	18	26	1	1	113	4	452
Decimation	4	3	1	18	26	8	1	246	4	452
Decimation	4	3	1	18	26	16	1	276	4	452
Programmable Rate										
Decimation	4 to 32	3	1	18	22	1	1	160	4	452
Decimation	8 to 128	3	1	18	22	1	1	193	4	452
Decimation	5 to 50	3	1	18	22	1	1	170	4	452
Varying Input Sample Period										
Decimation	4	6	1	18	30	1	1	202	8	452
Decimation	4	6	1	18	30	1	2	155	4	452
Decimation	4	6	1	18	30	1	6	181	2	452

The Virtex-5 FPGA test cases in Table 4 used ISE speed file version “PRODUCTION 1.65 2009-07-13, STEPPING level 0."

Table 4: **CIC Decimator: Virtex-5 XC5VSX35T-1-FF665**

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Input Width										
Decimation	4	3	1	2	8	1	1	73	4	449
Decimation	4	3	1	4	10	1	1	85	4	445
Decimation	4	3	1	8	14	1	1	109	4	445
Decimation	4	3	1	12	18	1	1	133	4	445
Decimation	4	3	1	16	22	1	1	157	4	445
Decimation	4	3	1	20	26	1	1	181	4	445
Varying Stages										
Decimation	4	3	1	18	24	1	1	169	4	445
Decimation	4	4	1	18	26	1	1	180	5	445
Decimation	4	5	1	18	28	1	1	225	7	445
Decimation	4	6	1	18	30	1	1	303	8	445
Varying Rate										
Decimation	4	3	1	18	24	1	1	169	4	445
Decimation	8	3	1	18	27	1	1	186	4	445
Decimation	16	3	1	18	30	1	1	203	4	445
Decimation	32	3	1	18	33	1	1	224	4	431
Decimation	64	3	1	18	36	1	1	243	4	445
Decimation	128	3	1	18	39	1	1	260	4	445
Decimation	256	3	1	18	42	1	1	276	4	445
Varying Differential Delay										
Decimation	4	3	1	4	10	1	1	85	4	445
Decimation	4	3	2	4	13	1	1	100	4	445
Decimation	4	3	1	8	14	1	1	109	4	445
Decimation	4	3	2	8	17	1	1	124	4	445
Decimation	4	3	1	12	18	1	1	133	4	445
Decimation	4	3	2	12	21	1	1	148	4	445
Decimation	4	3	1	16	22	1	1	157	4	445
Decimation	4	3	2	16	25	1	1	172	4	445
Decimation	4	3	1	20	26	1	1	181	4	445
Decimation	4	3	2	20	29	1	1	196	4	445

Table 4: **CIC Decimator: Virtex-5 XC5VSX35T-1-FF665** ***(Cont'd)***

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Channels										
Decimation	4	3	1	18	26	1	1	169	4	445
Decimation	4	3	1	18	26	8	1	293	4	438
Decimation	4	3	1	18	26	16	1	323	4	445
Programmable Rate										
Decimation	4 to 32	3	1	18	22	1	1	309	4	450
Decimation	8 to 128	3	1	18	22	1	1	339	4	400
Decimation	5 to 50	3	1	18	22	1	1	317	4	450
Varying Input Sample Period										
Decimation	4	6	1	18	30	1	1	303	8	445
Decimation	4	6	1	18	30	1	2	227	4	445
Decimation	4	6	1	18	30	1	6	255	2	445

The Spartan-6 FPGA test cases in Table 5 used ISE speed file version "ADVANCED 1.01d 2009-07-13."

Table 5: **CIC Decimator: Spartan-6 XC6SLX150-2-FGG484**

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Input Width										
Decimation	4	3	1	2	8	1	1	48	4	251
Decimation	4	3	1	4	10	1	1	52	4	251
Decimation	4	3	1	8	14	1	1	67	4	251
Decimation	4	3	1	12	18	1	1	81	4	251
Decimation	4	3	1	16	22	1	1	89	4	251
Decimation	4	3	1	20	26	1	1	106	4	251
Varying Stages										
Decimation	4	3	1	18	24	1	1	96	4	251
Decimation	4	4	1	18	26	1	1	103	5	251
Decimation	4	5	1	18	28	1	1	139	7	251
Decimation	4	6	1	18	30	1	1	168	8	251
Varying Rate										
Decimation	4	3	1	18	24	1	1	96	4	251
Decimation	8	3	1	18	27	1	1	108	4	251
Decimation	16	3	1	18	30	1	1	116	4	251
Decimation	32	3	1	18	33	1	1	129	4	251
Decimation	64	3	1	18	36	1	1	146	4	251
Decimation	128	3	1	18	39	1	1	153	4	251
Decimation	256	3	1	18	42	1	1	165	4	251
Varying Differential Delay										
Decimation	4	3	1	4	10	1	1	52	4	251
Decimation	4	3	2	4	13	1	1	64	4	251
Decimation	4	3	1	8	14	1	1	67	4	251
Decimation	4	3	2	8	17	1	1	78	4	251
Decimation	4	3	1	12	18	1	1	81	4	251
Decimation	4	3	2	12	21	1	1	97	4	251
Decimation	4	3	1	16	22	1	1	89	4	251
Decimation	4	3	2	16	25	1	1	101	4	251
Decimation	4	3	1	20	26	1	1	106	4	251
Decimation	4	3	2	20	29	1	1	118	4	251

Table 5: **CIC Decimator: Spartan-6 XC6SLX150-2-FGG484** ***(Cont'd)***

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Channels										
Decimation	4	3	1	18	26	1	1	96	4	251
Decimation	4	3	1	18	26	8	1	207	4	244
Decimation	4	3	1	18	26	16	1	232	4	236
Programmable Rate										
Decimation	4 to 32	3	1	18	22	1	1	138	4	250
Decimation	8 to 128	3	1	18	22	1	1	163	4	251
Decimation	5 to 50	3	1	18	22	1	1	155	4	251
Varying Input Sample Period										
Decimation	4	6	1	18	30	1	1	168	8	251
Decimation	4	6	1	18	30	1	2	135	4	251
Decimation	4	6	1	18	30	1	6	148	2	251

The Spartan-3DSP FPGA test cases in Table 6 used ISE speed file version "PRODUCTION 1.33 2009-07-13."

Table 6: **CIC Decimator: Spartan-3ADSP XC3SD1800A-4-FG676**

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	Slices	XtremeDSP Slices	F_{max} (MHz)
Varying Input Width										
Decimation	4	3	1	2	8	1	1	51	4	250
Decimation	4	3	1	4	10	1	1	58	4	244
Decimation	4	3	1	8	14	1	1	72	4	244
Decimation	4	3	1	12	18	1	1	86	4	244
Decimation	4	3	1	16	22	1	1	100	4	244
Decimation	4	3	1	20	26	1	1	114	4	244
Varying Stages										
Decimation	4	3	1	18	24	1	1	107	4	244
Decimation	4	4	1	18	26	1	1	111	5	244
Decimation	4	5	1	18	28	1	1	123	7	244
Decimation	4	6	1	18	30	1	1	178	8	244
Varying Rate										
Decimation	4	3	1	18	24	1	1	107	4	244
Decimation	8	3	1	18	27	1	1	119	4	244
Decimation	16	3	1	18	30	1	1	129	4	244
Decimation	32	3	1	18	33	1	1	143	4	244
Decimation	64	3	1	18	36	1	1	151	4	244
Decimation	128	3	1	18	39	1	1	164	4	244
Decimation	256	3	1	18	42	1	1	171	4	244
Varying Differential Delay										
Decimation	4	3	1	4	10	1	1	58	4	244
Decimation	4	3	2	4	13	1	1	69	4	244
Decimation	4	3	1	8	14	1	1	72	4	244
Decimation	4	3	2	8	17	1	1	83	4	244
Decimation	4	3	1	12	18	1	1	86	4	244
Decimation	4	3	2	12	21	1	1	97	4	244
Decimation	4	3	1	16	22	1	1	100	4	244
Decimation	4	3	2	16	25	1	1	111	4	244
Decimation	4	3	1	20	26	1	1	114	4	244
Decimation	4	3	2	20	29	1	1	125	4	244

Table 6: **CIC Decimator: Spartan-3ADSP XC3SD1800A-4-FG676** ***(Cont'd)***

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	Slices	XtremeDSP Slices	F_{max} (MHz)
Varying Channels										
Decimation	4	3	1	18	26	1	1	107	4	244
Decimation	4	3	1	18	26	8	1	198	4	244
Decimation	4	3	1	18	26	16	1	261	4	236
Programmable Rate										
Decimation	4 to 32	3	1	18	22	1	1	150	4	249
Decimation	8 to 128	3	1	18	22	1	1	172	4	249
Decimation	5 to 50	3	1	18	22	1	1	156	4	249
Varying Input Sample Period										
Decimation	4	6	1	18	30	1	1	178	8	244
Decimation	4	6	1	18	30	1	2	145	4	244
Decimation	4	6	1	18	30	1	6	157	2	244

CIC Interpolator

The Virtex-6 FPGA test cases in Table 7 used ISE speed file version "ADVANCED 1.01e 2009-07-13."

Table 7: **CIC Interpolator: Virtex-6 XC6VLX75T-1-FF784**

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Input Width										
Interpolation	4	3	1	2	6	1	4	46	4	452
Interpolation	4	3	1	4	8	1	4	55	4	452
Interpolation	4	3	1	8	12	1	4	78	4	452
Interpolation	4	3	1	12	16	1	4	99	4	452
Interpolation	4	3	1	16	20	1	4	118	4	452
Interpolation	4	3	1	20	24	1	4	141	4	452
Varying Stages										
Interpolation	4	3	1	18	22	1	4	131	4	452
Interpolation	4	4	1	18	24	1	4	140	5	452
Interpolation	4	5	1	18	26	1	4	160	7	452
Interpolation	4	6	1	18	28	1	4	201	8	452
Varying Rate										
Interpolation	4	3	1	18	22	1	4	131	4	452
Interpolation	8	3	1	18	24	1	8	136	4	452
Interpolation	16	3	1	18	26	1	16	140	4	452
Interpolation	32	3	1	18	28	1	32	145	4	452
Interpolation	64	3	1	18	30	1	64	158	4	452
Interpolation	128	3	1	18	32	1	128	154	4	452
Interpolation	256	3	1	18	34	1	256	154	4	452
Varying Differential Delay										
Interpolation	4	3	1	4	8	1	4	55	4	452
Interpolation	4	3	2	4	11	1	4	64	4	452
Interpolation	4	3	1	8	12	1	4	78	4	452
Interpolation	4	3	2	8	15	1	4	85	4	452
Interpolation	4	3	1	12	16	1	4	99	4	452
Interpolation	4	3	2	12	19	1	4	105	4	452
Interpolation	4	3	1	16	20	1	4	118	4	452
Interpolation	4	3	2	16	23	1	4	125	4	452
Interpolation	4	3	1	20	24	1	4	141	4	452
Interpolation	4	3	2	20	27	1	4	150	4	452

Table 7: **CIC Interpolator: Virtex-6 XC6VLX75T-1-FF784** ***(Cont'd)***

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Channels										
Interpolation	4	3	1	18	26	1	4	131	4	452
Interpolation	4	3	1	18	26	8	4	254	4	452
Interpolation	4	3	1	18	26	16	4	284	4	452
Programmable Rate										
Interpolation	4 to 32	3	1	18	22	1	4	229	4	452
Interpolation	8 to 128	3	1	18	22	1	8	258	4	452
Interpolation	5 to 50	3	1	18	22	1	5	234	4	452
Varying Input Sample Period										
Interpolation	4	6	1	18	28	1	4	201	8	452
Interpolation	4	6	1	18	28	1	8	155	4	452
Interpolation	4	6	1	18	28	1	24	180	2	452

The Virtex-5 FPGA test cases in Table 8 used ISE speed file version “PRODUCTION 1.65 2009-07-13, STEPPING level 0."

Table 8: **CIC Interpolator: Virtex-5 XC5VSX35T-1-FF665**

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Input Width										
Interpolation	4	3	1	2	6	1	4	60	4	445
Interpolation	4	3	1	4	8	1	4	76	4	445
Interpolation	4	3	1	8	12	1	4	108	4	445
Interpolation	4	3	1	12	16	1	4	140	4	445
Interpolation	4	3	1	16	20	1	4	172	4	445
Interpolation	4	3	1	20	24	1	4	205	4	445
Varying Stages										
Interpolation	4	3	1	18	22	1	4	188	4	445
Interpolation	4	4	1	18	24	1	4	197	5	445
Interpolation	4	5	1	18	26	1	4	231	7	445
Interpolation	4	6	1	18	28	1	4	290	8	445
Varying Rate										
Interpolation	4	3	1	18	22	1	4	188	4	445
Interpolation	8	3	1	18	24	1	8	192	4	445
Interpolation	16	3	1	18	26	1	16	196	4	445
Interpolation	32	3	1	18	28	1	32	204	4	445
Interpolation	64	3	1	18	30	1	64	212	4	445
Interpolation	128	3	1	18	32	1	128	216	4	445
Interpolation	256	3	1	18	34	1	256	219	4	445
Varying Differential Delay										
Interpolation	4	3	1	4	8	1	4	76	4	445
Interpolation	4	3	2	4	11	1	4	83	4	445
Interpolation	4	3	1	8	12	1	4	108	4	445
Interpolation	4	3	2	8	15	1	4	115	4	445
Interpolation	4	3	1	12	16	1	4	140	4	445
Interpolation	4	3	2	12	19	1	4	147	4	445
Interpolation	4	3	1	16	20	1	4	172	4	445
Interpolation	4	3	2	16	23	1	4	179	4	445
Interpolation	4	3	1	20	24	1	4	205	4	445
Interpolation	4	3	2	20	27	1	4	215	4	445

Table 8: **CIC Interpolator: Virtex-5 XC5VSX35T-1-FF665 *(Cont'd)***

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Channels										
Interpolation	4	3	1	18	26	1	4	188	4	445
Interpolation	4	3	1	18	26	8	4	303	4	445
Interpolation	4	3	1	18	26	16	4	332	4	445
Programmable Rate										
Interpolation	4 to 32	3	1	18	22	1	4	309	4	450
Interpolation	8 to 128	3	1	18	22	1	8	339	4	400
Interpolation	5 to 50	3	1	18	22	1	5	317	4	450
Varying Input Sample Period										
Interpolation	4	6	1	18	28	1	4	290	8	445
Interpolation	4	6	1	18	28	1	8	213	4	445
Interpolation	4	6	1	18	28	1	24	234	2	445

The Spartan-6 FPGA test cases in Table 9 used ISE speed file version “ADVANCED 1.01d 2009-07-13.”

Table 9: **CIC Interpolator: Spartan-6 XC6SLX150-2-FGG484**

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Input Width										
Interpolation	4	3	1	2	6	1	4	41	4	251
Interpolation	4	3	1	4	8	1	4	47	4	251
Interpolation	4	3	1	8	12	1	4	64	4	251
Interpolation	4	3	1	12	16	1	4	85	4	251
Interpolation	4	3	1	16	20	1	4	100	4	251
Interpolation	4	3	1	20	24	1	4	115	4	251
Varying Stages										
Interpolation	4	3	1	18	22	1	4	110	4	251
Interpolation	4	4	1	18	24	1	4	115	5	251
Interpolation	4	5	1	18	26	1	4	130	7	251
Interpolation	4	6	1	18	28	1	4	165	8	212
Varying Rate										
Interpolation	4	3	1	18	22	1	4	110	4	251
Interpolation	8	3	1	18	24	1	8	113	4	251
Interpolation	16	3	1	18	26	1	16	114	4	251
Interpolation	32	3	1	18	28	1	32	115	4	251
Interpolation	64	3	1	18	30	1	64	122	4	251
Interpolation	128	3	1	18	32	1	128	131	4	251
Interpolation	256	3	1	18	34	1	256	135	4	251
Varying Differential Delay										
Interpolation	4	3	1	4	8	1	4	47	4	251
Interpolation	4	3	2	4	11	1	4	57	4	251
Interpolation	4	3	1	8	12	1	4	64	4	251
Interpolation	4	3	2	8	15	1	4	71	4	251
Interpolation	4	3	1	12	16	1	4	85	4	251
Interpolation	4	3	2	12	19	1	4	91	4	251
Interpolation	4	3	1	16	20	1	4	100	4	251
Interpolation	4	3	2	16	23	1	4	111	4	251
Interpolation	4	3	1	20	24	1	4	115	4	251
Interpolation	4	3	2	20	27	1	4	120	4	251

Table 9: **CIC Interpolator: Spartan-6 XC6SLX150-2-FGG484** ***(Cont'd)***

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	LUT-FF pairs	XtremeDSP Slices	F_{max} (MHz)
Varying Channels										
Interpolation	4	3	1	18	26	1	4	110	4	251
Interpolation	4	3	1	18	26	8	4	206	4	228
Interpolation	4	3	1	18	26	16	4	231	4	236
Programmable Rate										
Interpolation	4 to 32	3	1	18	22	1	4	200	4	198
Interpolation	8 to 128	3	1	18	22	1	8		4	
Interpolation	5 to 50	3	1	18	22	1	5		4	
Varying Input Sample Period										
Interpolation	4	6	1	18	28	1	4	165	8	251
Interpolation	4	6	1	18	28	1	8	126	4	251
Interpolation	4	6	1	18	28	1	24	135	2	251

The Spartan-3ADSP FPGA test cases in Table 10 used ISE speed file version "PRODUCTION 1.33 2009-07-13."

Table 10: **CIC Interpolator: Spartan-3ADSP XC3SD1800A-4-FG676**

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	Slices	XtremeDSP™ Slices	F_{max} (MHz)
Varying Input Width										
Interpolation	4	3	1	2	6	1	4	41	4	244
Interpolation	4	3	1	4	8	1	4	50	4	244
Interpolation	4	3	1	8	12	1	4	68	4	244
Interpolation	4	3	1	12	16	1	4	86	4	244
Interpolation	4	3	1	16	20	1	4	105	4	251
Interpolation	4	3	1	20	24	1	4	123	4	251
Varying Stages										
Interpolation	4	3	1	18	22	1	4	114	4	244
Interpolation	4	4	1	18	24	1	4	122	5	244
Interpolation	4	5	1	18	26	1	4	141	7	244
Interpolation	4	6	1	18	28	1	4	173	8	244
Varying Rate										
Interpolation	4	3	1	18	22	1	4	114	4	244
Interpolation	8	3	1	18	24	1	8	117	4	244
Interpolation	16	3	1	18	26	1	16	122	4	244
Interpolation	32	3	1	18	28	1	32	127	4	244
Interpolation	64	3	1	18	30	1	64	131	4	244
Interpolation	128	3	1	18	32	1	128	131	4	244
Interpolation	256	3	1	18	34	1	256	135	4	244
Varying Differential Delay										
Interpolation	4	3	1	4	8	1	4	50	4	244
Interpolation	4	3	2	4	11	1	4	57	4	244
Interpolation	4	3	1	8	12	1	4	68	4	244
Interpolation	4	3	2	8	15	1	4	75	4	244
Interpolation	4	3	1	12	16	1	4	86	4	244
Interpolation	4	3	2	12	19	1	4	93	4	244
Interpolation	4	3	1	16	20	1	4	105	4	244
Interpolation	4	3	2	16	23	1	4	112	4	244
Interpolation	4	3	1	20	24	1	4	123	4	244
Interpolation	4	3	2	20	27	1	4	130	4	244

Table 10: **CIC Interpolator: Spartan-3ADSP XC3SD1800A-4-FG676** ***(Cont'd)***

	Rate	Stages	Diff. Delay	Input Width	Output Width	Chan.	Input Sample Period	Slices	XtremeDSP™ Slices	F_{max} (MHz)
Varying Channels										
Interpolation	4	3	1	18	26	1	4	114	4	244
Interpolation	4	3	1	18	26	8	4	203	4	244
Interpolation	4	3	1	18	26	16	4	254	4	244
Programmable Rate										
Interpolation	4 to 32	3	1	18	22	1	4	196	4	246
Interpolation	8 to 128	3	1	18	22	1	8	249	4	199
Interpolation	5 to 50	3	1	18	22	1	5	202	4	217
Varying Input Sample Period										
Interpolation	4	6	1	18	28	1	4	173	8	244
Interpolation	4	6	1	18	28	1	8	138	4	244
Interpolation	4	6	1	18	28	1	24	144	2	244

References

1. Eugene B. Hogenauer, *An Economical Class of Digital Filters for Decimation and Interpolation*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-29, No. 2, April 1981.

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY.*

Refer to the IP Release Notes Guide (XTP025) for further information on this core. There is a link to all the DSP IP and then to each core. For each core, there is a master Answer Record that contains the Release Notes and Known Issues list for each core. The following information is listed for each version of the core:

- New Features
- Bug Fixes
- Known Issues

Ordering Information

This LogiCORE IP module is included at no additional cost with the Xilinx ISE Design Suite software and is provided under the terms of the Xilinx End User License Agreement. Use the CORE Generator software included with the ISE Design Suite to generate the core. For more information, please visit the core page.

Information about additional Xilinx LogiCORE IP modules is available at the Xilinx IP Center. For pricing and availability of other Xilinx LogiCORE IP modules and software, please contact your local Xilinx sales representative.

Revision History

The following table shows the revision history for this document:

Date	Version	Revision
10/10/07	1.0	Initial Xilinx release.
03/24/08	1.1	Updated for core version 1.1.
04/25/08	1.2	Updated for core version 1.2.
09/16/09	1.3	Updated for core version 1.3 including support for Virtex-6 and Spartan-6.
04/19/10	2.0	Updated for core version 2.0.
03/01/11	2.1	Updated to include support for Kintex-7 and Virtex-7. ISE Design Suite 13.1.

Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter “Information,” to you “AS IS” with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.