CIC IP Core

User Guide

Subscribe

Send Feedback

UG-CIC
2014.12.15

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Contents

About The CIC IP Core 1

2014.12.15

UG-CIC Subscribe Send Feedback

The Altera® CIC IP core implements a cascaded integrator-comb (CIC) filter with data ports that are compatible with the Avalon® Streaming (Avalon-ST) interface. CIC filters (also known as Hogenauer filters) are computationally efficient for extracting baseband signals from narrow-band sources using decimation. They also construct narrow-band signals from processed baseband signals using interpolation.

CIC filters use only adders and registers; they require no multipliers to handle large rate changes. Therefore, CIC is a suitable and economical filter architecture for hardware implementation, and is widely used in sample rate conversion designs such as digital down converters (DDC) and digital up converters (DUC).

Altera DSP IP Core Features

- Avalon Streaming (Avalon-ST) interfaces
- DSP Builder ready
- Testbenches to verify the IP core
- IP functional simulation models for use in Altera-supported VHDL and Verilog HDL simulators

CIC IP Core Features

- Interpolation and decimation filters with variable rate change factors (2 to 32,000), a configurable number of stages (1 to 12), and two differential delay options (1 or 2).
- Single clock domain with selectable number of interfaces and a maximum of 1,024 channels.
- Selectable data storage options with an option to use pipelined integrators.
- Configurable input data width (1 to 32 bits) and output data width (1 to full resolution data width).
- Selectable output rounding modes (truncation, convergent rounding, rounding up, or saturation) and Hogenauer pruning support.
- Optimization for speed by specifying the number of pipeline stages used by each integrator.
- Compensation filter coefficients generation.
- IP functional simulation models for use in Altera-supported VHDL and Verilog HDL simulators.
- DSP Builder ready.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

CIC IP Core Device Family Support

Altera offers the following device support levels for Altera IP cores:

- Preliminary support—Altera verifies the IP core with preliminary timing models for this device family. The IP core meets all functional requirements, but might still be undergoing timing analysis for the device family. You can use it in production designs with caution.
- Final support—Altera verifies the IP core with final timing models for this device family. The IP core meets all functional and timing requirements for the device family. You can use it in production designs.

Table 1-1: Device Family Support

Device Family	Support
Arria® II GX	Final
Arria II GZ	Final
ArriaV	Final
Arria10	Final
Cyclone® IV GX	Final
Stratix® IV GT	Final
Stratix IV GX/E	Final
Stratix V	Final
Other device families	No support

DSP IP Core Verification

Before releasing a version of an IP core, Altera runs comprehensive regression tests to verify its quality and correctness. Altera generates custom variations of the IP core to exercise the various parameter options and thoroughly simulates the resulting simulation models with the results verified against master simulation models.

CIC IP Core Release Information

Table 1-2: CIC IP Core Release Information

Item	Description
Version	14.1
Release Date	December 2014
Ordering Code	IP-CIC
Product ID(s)	00BB
Vendor ID(s)	6AF7

Altera verifies that the current version of the Quartus II software compiles the previous version of each IP core. Altera does not verify that the Quartus II software compiles IP core versions older than the previous version. The *Altera IP Release Notes* lists any exceptions.

Related Information

- Altera IP Release Notes
- Errata for CIC IP core in the Knowledge Base

CIC IP Core Performance and Resource Utilization

The following parameters apply:

- **Number of stages**: 8
- **Rate change factor**: 8
- **Differential delay**: 1
- **Integrator data storage**: Memory (whenever possible)
- **Differentiator data storage**: Memory (whenever possible)
- **Input data width**: 16
- **Output data width**: Full precision
- **Output rounding**: No rounding

The target f_{MAX} is 1 GHz.

Table 1-3: CIC IP Core Performance

Typical performance using the Quartus II software with the Arria V (5AGXFB3H4F40C4), Cyclone V (5CGXFC7D6F31C6), and Stratix V (5SGSMD4H2F35C2) devices

Device	Filter Type	ALM	Memory		Registers		f_{MAX} (MHz)
			M10K	M20K	Primary	Secondary	
Arria V	Decimator	493	2	--	1,149	5	207.34
Arria V	Decimator 5 Channels	1,162	2	--	3,749	6	207
Arria V	Decimator 5 Channels 3 Interfaces	911	37	--	1,722	6	255
Arria V	Decimator Hogenauer Pruning	352	1	--	785	12	304
Arria V	Decimator Trunction	463	2	--	1,055	5	198.69
Arria V	Decimator Variable Rate Change	919	37	--	1,730	7	256
Arria V	Interpolator	326	1	--	728	18	320
Arria V	Interpolator 5 Channels	762	1	--	2,369	27	288

Device	Filter Type	ALM	Memory		Registers		f_{MAX} (MHz)
			M10K	M20K	Primary	Secondary	
Arria V	Interpolator 5 Channels 3 Interfaces	886	27	--	1,776	17	232.61
Arria V	Interpolator Convergent Rounding	352	1	--	785	12	304
Arria V	Interpolator Variable Rate Change	889	27	--	1,772	23	235
Cyclone V	Decimator	492	2	--	1,137	17	182
Cyclone V	Decimator 5 Channels	1,162	2	--	3,748	8	190.15
Cyclone V	Decimator 5 Channels 3 Interfaces	906	37	--	1,719	9	204
Cyclone V	Decimator Hogenauer Pruning	352	1	--	784	14	246
Cyclone V	Decimator Truncation	463	2	--	1,054	4	177
Cyclone V	Decimator Variable Rate Change	917	37	--	1,730	5	193.27
Cyclone V	Interpolator	324	1	--	709	37	264
Cyclone V	Interpolator 5 Channels	760	1	--	2,383	11	235
Cyclone V	Interpolator 5 Channels 3 Interfaces	890	27	--	1,747	48	168
Cyclone V	Interpolator Convergent Rounding	352	1	--	784	14	246.06
Cyclone V	Interpolator Variable Rate Change	894	27	--	1,725	70	165
Stratix V	Decimator	515	--	1	1,152	6	377
Stratix V	Decimator 5 Channels	1,176	--	1	3,750	8	413

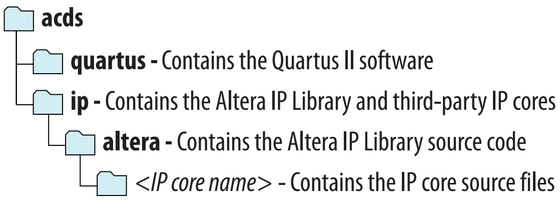
Device	Filter Type	ALM	Memory		Registers		f_{MAX} (MHz)
			M10K	M20K	Primary	Secondary	
Stratix V	Decimator 5 Channels 3 Interfaces	1,891	--	11	5,562	8	450.05
Stratix V	Decimator Hogenauer Pruning	361	--	0	790	13	450
Stratix V	Decimator Truncation	483	--	1	1,059	4	376
Stratix V	Decimator Variable Rate Change	1,900	--	11	5,574	3	450
Stratix V	Interpolator	335	--	0	737	14	450.05
Stratix V	Interpolator 5 Channels	771	--	0	2,390	8	450
Stratix V	Interpolator 5 Channels 3 Interfaces	1,625	--	8	4,635	70	450
Stratix V	Interpolator Convergent Rounding	361	--	0	790	13	450
Arria 10	Interpolator Variable Rate Change	464	--	0	128	128	451

CIC IP Core Getting Started 2

Installing and Licensing IP Cores

The Altera IP Library provides many useful IP core functions for your production use without purchasing an additional license. Some Altera MegaCore® IP functions require that you purchase a separate license for production use. However, the OpenCore® feature allows evaluation of any Altera IP core in simulation and compilation in the Quartus® II software. After you are satisfied with functionality and perfformance, visit the Self Service Licensing Center to obtain a license number for any Altera product.

Figure 2-1: IP Core Installation Path


acds
quartus - Contains the Quartus II software
ip - Contains the Altera IP Library and third-party IP cores
altera - Contains the Altera IP Library source code
<IP core name> - Contains the IP core source files

Note: The default IP installation directory on Windows is **<*drive*>:\altera\**<*version number*>; on Linux it is <*home directory*>**/altera/** <*version number*>.

Related Information

- Altera Licensing Site
- Altera Software Installation and Licensing Manual

OpenCore Plus IP Evaluation

Altera's free OpenCore Plus feature allows you to evaluate licensed MegaCore IP cores in simulation and hardware before purchase. You need only purchase a license for MegaCore IP cores if you decide to take your design to production. OpenCore Plus supports the following evaluations:

- Simulate the behavior of a licensed IP core in your system.
- Verify the functionality, size, and speed of the IP core quickly and easily.
- Generate time-limited device programming files for designs that include IP cores.
- Program a device with your IP core and verify your design in hardware.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

OpenCore Plus evaluation supports the following two operation modes:

- Untethered—run the design containing the licensed IP for a limited time.
- Tethered—run the design containing the licensed IP for a longer time or indefinitely. This requires a connection between your board and the host computer.

Note: All IP cores that use OpenCore Plus time out simultaneously when any IP core in the design times out.

CIC IP Core OpenCore Plus Timeout Behavior

All IP cores in a device time out simultaneously when the most restrictive evaluation time is reached. If there is more than one IP core in a design, the time-out behavior of the other IP cores may mask the time-out behavior of a specific IP core .

All IP cores in a device time out simultaneously when the most restrictive evaluation time is reached. If there is more than one IP core in a design, a specific IP core's time-out behavior may be masked by the time-out behavior of the other IP cores. For IP cores, the untethered time-out is 1 hour; the tethered time-out value is indefinite. Your design stops working after the hardware evaluation time expires. The Quartus II software uses OpenCore Plus Files (**.ocp**) in your project directory to identify your use of the OpenCore Plus evaluation program. After you activate the feature, do not delete these files..

When the evaluation time expires, the data output signal goes low.

Related Information

- AN 320: OpenCore Plus Evaluation of Megafunctions

IP Catalog and Parameter Editor

The Quartus II IP Catalog (**Tools** > **IP Catalog**) and parameter editor help you easily customize and integrate IP cores into your project. You can use the IP Catalog and parameter editor to select, customize, and generate files representing your custom IP variation.

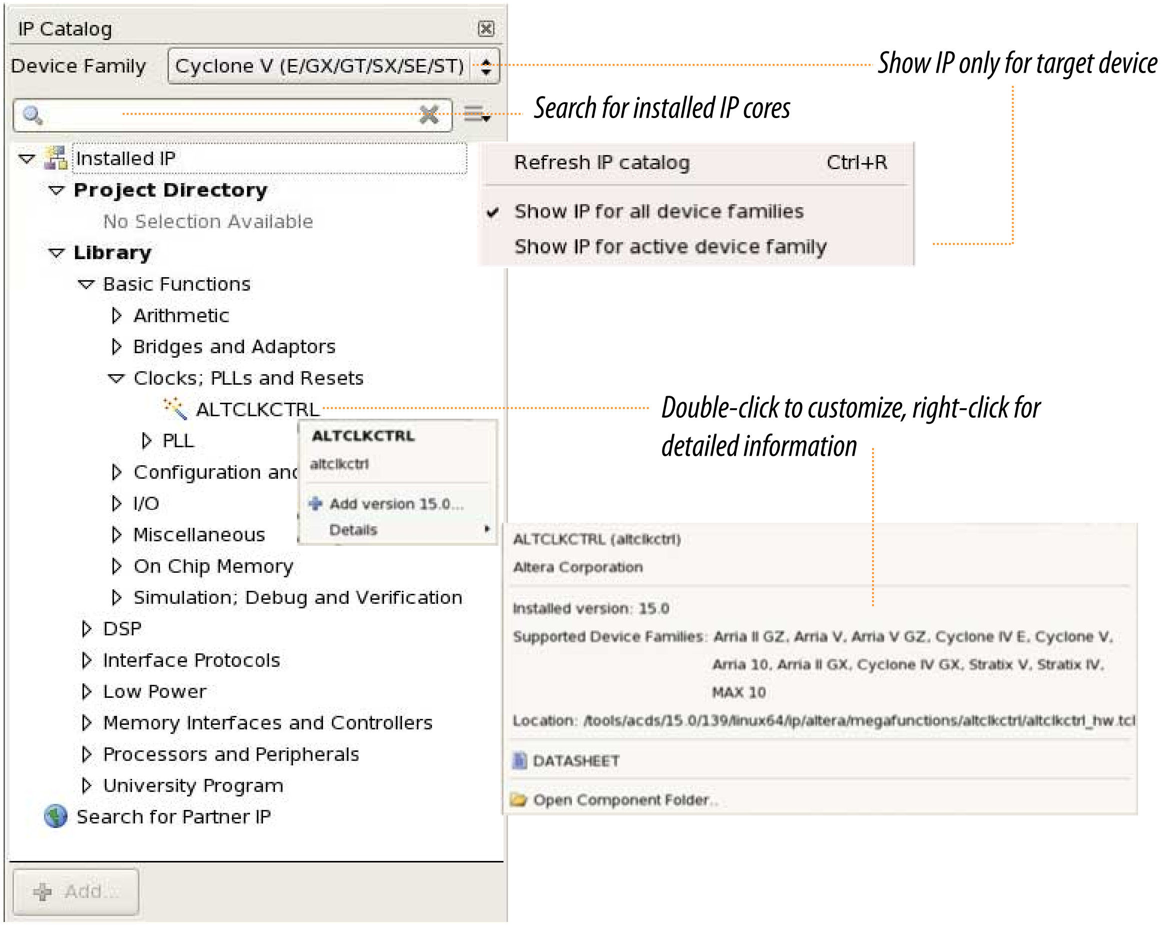
Note: The IP Catalog (**Tools** > **IP Catalog)** and parameter editor replace the MegaWizard™ Plug-In Manager for IP selection and parameterization, beginning in Quartus II software version 14.0. Use the IP Catalog and parameter editor to locate and paramaterize Altera IP cores.

The IP Catalog lists installed IP cores available for your design. Double-click any IP core to launch the parameter editor and generate files representing your IP variation. The parameter editor prompts you to specify an IP variation name, optional ports, and output file generation options. The parameter editor generates a top-level Qsys system file (**.qsys**) or Quartus II IP file (**.qip**) representing the IP core in your project. You can also parameterize an IP variation without an open project.

Use the following features to help you quickly locate and select an IP core:

- Filter IP Catalog to **Show IP for active device family** or **Show IP for all device families**. If you have no project open, select the **Device Family** in IP Catalog.
- Type in the Search field to locate any full or partial IP core name in IP Catalog.
- Right-click an IP core name in IP Catalog to display details about supported devices, open the IP core's installation folder, and view links to documentation.
- Click **Search for Partner IP**, to access partner IP information on the Altera website.

Figure 2-2: Quartus II IP Catalog



ALTCLKCTRL (altclkctrl)
Altera Corporation
Installed version: 15.0
Supported Device Families: Arria II GZ, Arria V, Arria V GZ, Cyclone IV E, Cyclone V, Arria 10, Arria II GX, Cyclone IV GX, Stratix V, Stratix IV, MAX 10
Location: /tools/acds/15.0/139/linux64/ip/altera/megafunctions/altclkctrl/altclkctrl_hw.tcl

Show IP only for target device

Search for installed IP cores

Double-click to customize, right-click for detailed information

Note: The IP Catalog is also available in Qsys (**View** > **IP Catalog**). The Qsys IP Catalog includes exclusive system interconnect, video and image processing, and other system-level IP that are not available in the Quartus II IP Catalog. For more information about using the Qsys IP Catalog, refer to *Creating a System with Qsys* in the *Quartus II Handbook*.

Specifying IP Core Parameters and Options

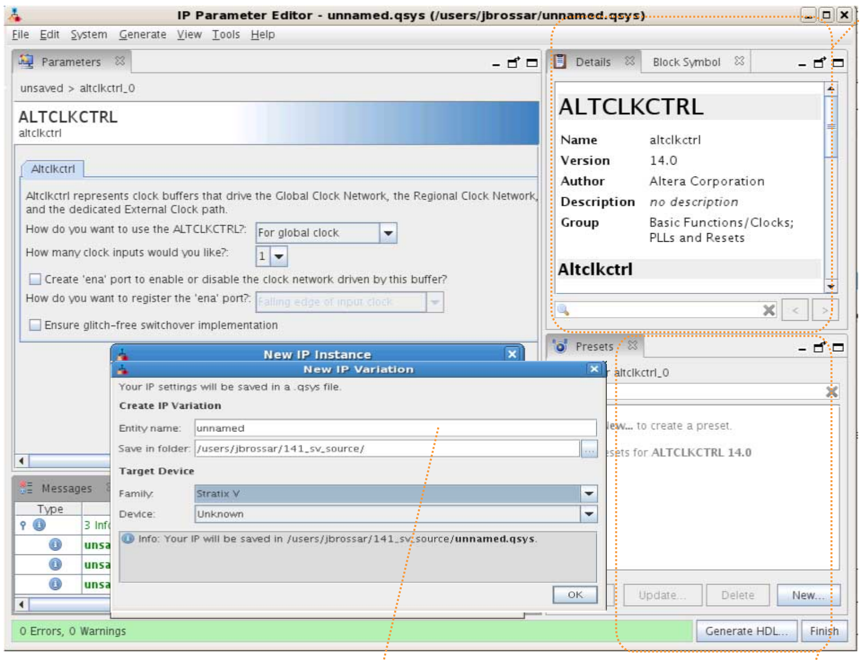
You can quickly configure a custom IP variation in the parameter editor. Use the following steps to specify IP core options and parameters in the parameter editor. Refer to *Specifying IP Core Parameters and Options (Legacy Parameter Editors)* for configuration of IP cores using the legacy parameter editor.

1. In the IP Catalog (**Tools** > **IP Catalog**), locate and double-click the name of the IP core to customize. The parameter editor appears.
2. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named <*your_ip*>**.qsys**. Click **OK**.
3. Specify the parameters and options for your IP variation in the parameter editor, including one or more of the following. Refer to your IP core user guide for information about specific IP core parameters.

- Optionally select preset parameter values if provided for your IP core. Presets specify initial parameter values for specific applications.
- Specify parameters defining the IP core functionality, port configurations, and device-specific features.
- Specify options for processing the IP core files in other EDA tools.

4. Click **Generate HDL**, the **Generation** dialog box appears.
5. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
6. To generate a simulation testbench, click **Generate** > **Generate Testbench System**.
7. To generate an HDL instantiation template that you can copy and paste into your text editor, click **Generate** > **HDL Example**.
8. Click **Finish**. The parameter editor adds the top-level **.qsys** file to the current project automatically. If you are prompted to manually add the **.qsys** file to the project, click **Project** > **Add/Remove Files in Project** to add the file.
9. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

Figure 2-3: IP Parameter Editor



unsaved > altclkctrl_0

ALTCLKCTRL
altclkctrl

Altclkctrl

Altclkctrl represents clock buffers that drive the Global Clock Network, the Regional Clock Network, and the dedicated External Clock path.

How do you want to use the ALTCLKCTRL?: For global clock

How many clock inputs would you like?: 1

□ Create 'ena' port to enable or disable the clock network driven by this buffer?

How do you want to register the 'ena' port?: Falling edge of input clock

□ Ensure glitch-free switchover implementation

ALTCLKCTRL

Name	altclkctrl
Version	14.0
Author	Altera Corporation
Description	no description
Group	Basic Functions/Clocks; PLLs and Resets

Altclkctrl

Your IP settings will be saved in a .qsys file.

Create IP Variation

Entity name: unnamed

Save in folder: /users/jbrossar/141_sv_source/

Target Device

Family: Stratix V

Device: Unknown

Info: Your IP will be saved in /users/jbrossar/141_sv_source/**unnamed.qsys**.

...ew... to create a preset.

...sets for ALTCLKCTRL 14.0

0 Errors, 0 Warnings

View IP port and parameter details

Specify your IP variation name and target device

Apply preset parameters for specific applications

Files Generated for Altera IP Cores

The Quartus II software generates the following IP core output file structure:

Figure 2-4: IP Core Generated Files

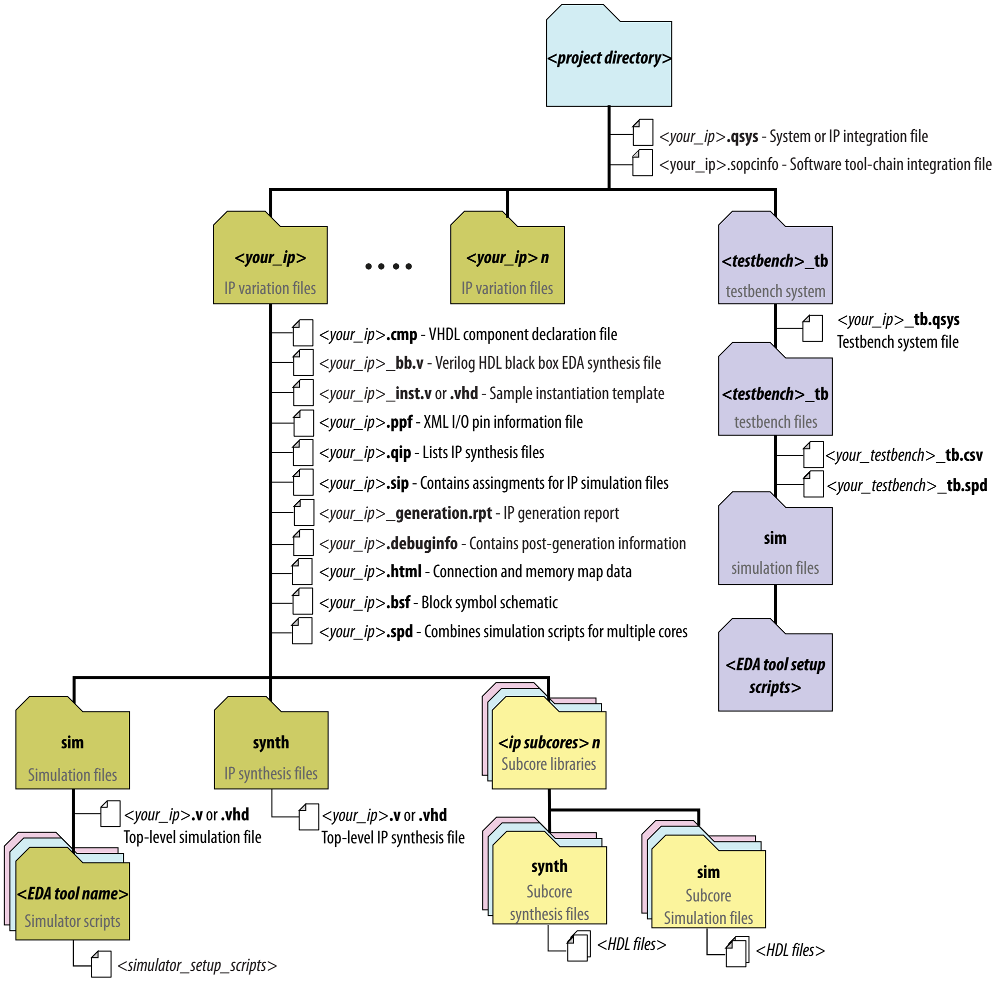

<project directory>
<your_ip>.qsys - System or IP integration file
<your_ip>.sopcinfo - Software tool-chain integration file
<your_ip>
IP variation files
....
<your_ip> n
IP variation files
<testbench>_tb
testbench system
<your_ip>_tb.qsys
Testbench system file
<testbench>_tb
testbench files
<your_testbench>_tb.csv
<your_testbench>_tb.spd
sim
simulation files
<EDA tool setup scripts>
<your_ip>.cmp - VHDL component declaration file
<your_ip>_bb.v - Verilog HDL black box EDA synthesis file
<your_ip>_inst.v or .vhd - Sample instantiation template
<your_ip>.ppf - XML I/O pin information file
<your_ip>.qip - Lists IP synthesis files
<your_ip>.sip - Contains assingments for IP simulation files
<your_ip>_generation.rpt - IP generation report
<your_ip>.debuginfo - Contains post-generation information
<your_ip>.html - Connection and memory map data
<your_ip>.bsf - Block symbol schematic
<your_ip>.spd - Combines simulation scripts for multiple cores
sim
Simulation files
synth
IP synthesis files
<ip subcores> n
Subcore libraries
<your_ip>.v or .vhd
Top-level simulation file
<your_ip>.v or .vhd
Top-level IP synthesis file
<EDA tool name>
Simulator scripts
<simulator_setup_scripts>
synth
Subcore synthesis files
<HDL files>
sim
Subcore Simulation files
<HDL files>

Table 2-1: IP Core Generated Files

File Name	Description
<*my_ip*>.qsys	The Qsys system or top-level IP variation file. <*my_ip*> is the name that you give your IP variation.
<*system*>.sopcinfo	Describes the connections and IP component parameterizations in your Qsys system. You can parse its contents to get requirements when you develop software drivers for IP components. Downstream tools such as the Nios II tool chain use this file. The **.sopcinfo** file and the **system.h** file generated for the Nios II tool chain include address map information for each slave relative to each master that accesses the slave. Different masters may have a different address map to access a particular slave component.

File Name	Description
<*my_ip*>.cmp	The VHDL Component Declaration (**.cmp**) file is a text file that contains local generic and port definitions that you can use in VHDL design files.
<*my_ip*>.html	A report that contains connection information, a memory map showing the address of each slave with respect to each master to which it is connected, and parameter assignments.
<*my_ip*>_generation.rpt	IP or Qsys generation log file. A summary of the messages during IP generation.
<*my_ip*>.debuginfo	Contains post-generation information. Used to pass System Console and Bus Analyzer Toolkit information about the Qsys interconnect. The Bus Analysis Toolkit uses this file to identify debug components in the Qsys interconnect.
<*my_ip*>.qip	Contains all the required information about the IP component to integrate and compile the IP component in the Quartus II software.
<*my_ip*>.csv	Contains information about the upgrade status of the IP component.
<*my_ip*>.bsf	A Block Symbol File (**.bsf**) representation of the IP variation for use in Quartus II Block Diagram Files (**.bdf**).
<*my_ip*>.spd	Required input file for `ip-make-simscript` to generate simulation scripts for supported simulators. The **.spd** file contains a list of files generated for simulation, along with information about memories that you can initialize.
<*my_ip*>.ppf	The Pin Planner File (**.ppf**) stores the port and node assignments for IP components created for use with the Pin Planner.
<*my_ip*>_bb.v	You can use the Verilog black-box (**_bb.v**) file as an empty module declaration for use as a black box.
<*my_ip*>.sip	Contains information required for NativeLink simulation of IP components. You must add the **.sip** file to your Quartus project.
<*my_ip*>_inst.v or **_inst.vhd**	HDL example instantiation template. You can copy and paste the contents of this file into your HDL file to instantiate the IP variation.
<*my_ip*>.regmap	If the IP contains register information, the **.regmap** file generates. The **.regmap** file describes the register map information of master and slave interfaces. This file complements the **.sopcinfo** file by providing more detailed register information about the system. This enables register display views and user customizable statistics in System Console.

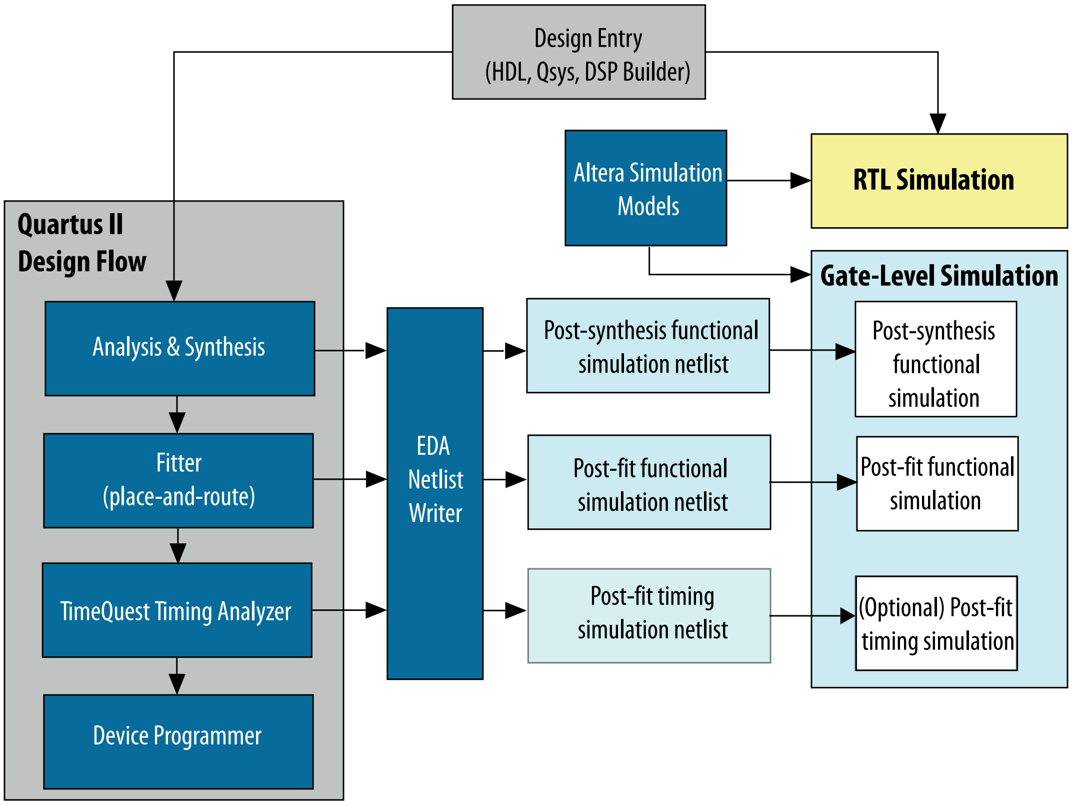
File Name	Description
<*my_ip*>.svd	Allows HPS System Debug tools to view the register maps of peripherals connected to HPS within a Qsys system. During synthesis, the **.svd** files for slave interfaces visible to System Console masters are stored in the **.sof** file in the debug section. System Console reads this section, which Qsys can query for register map information. For system slaves, Qsys can access the registers by name.
<*my_ip*>.v or **<*my_ip*>.vhd**	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
mentor/	Contains a ModelSim® script **msim_setup.tcl** to set up and run a simulation.
aldec/	Contains a Riviera-PRO script **rivierapro_setup.tcl** to setup and run a simulation.
/synopsys/vcs **/synopsys/vcsmx**	Contains a shell script **vcs_setup.sh** to set up and run a VCS® simulation. Contains a shell script **vcsmx_setup.sh** and **synopsys_ sim.setup** file to set up and run a VCS MX® simulation.
/cadence	Contains a shell script **ncsim_setup.sh** and other setup files to set up and run an NCSIM simulation.
/submodules	Contains HDL files for the IP core submodule.
<*child IP cores*>/	For each generated child IP core directory, Qsys generates **/synth** and **/ sim** sub-directories.

Simulating Altera IP Cores in other EDA Tools

The Quartus II software supports RTL and gate-level design simulation of Altera IP cores in supported EDA simulators. Simulation involves setting up your simulator working environment, compiling simulation model libraries, and running your simulation.

You can use the functional simulation model and the testbench or example design generated with your IP core for simulation. The functional simulation model and testbench files are generated in a project subdirectory. This directory may also include scripts to compile and run the testbench. For a complete list of models or libraries required to simulate your IP core, refer to the scripts generated with the testbench. You can use the Quartus II NativeLink feature to automatically generate simulation files and scripts. NativeLink launches your preferred simulator from within the Quartus II software.

Figure 2-5: Simulation in Quartus II Design Flow


Design Entry
(HDL, Qsys, DSP Builder)
Altera Simulation Models
RTL Simulation
Quartus II Design Flow
Analysis & Synthesis
Fitter (place-and-route)
TimeQuest Timing Analyzer
Device Programmer
EDA Netlist Writer
Post-synthesis functional simulation netlist
Post-fit functional simulation netlist
Post-fit timing simulation netlist
Gate-Level Simulation
Post-synthesis functional simulation
Post-fit functional simulation
(Optional) Post-fit timing simulation

Note: Post-fit timing simulation is supported only for Stratix IV and Cyclone IV devices in the current version of the Quartus II software. Altera IP supports a variety of simulation models, including simulation-specific IP functional simulation models and encrypted RTL models, and plain text RTL models. These are all cycle-accurate models. The models support fast functional simulation of your IP core instance using industry-standard VHDL or Verilog HDL simulators. For some cores, only the plain text RTL model is generated, and you can simulate that model. Use the simulation models only for simulation and not for synthesis or any other purposes. Using these models for synthesis creates a nonfunctional design.

Related Information

Simulating Altera Designs

DSP Builder Design Flow

DSP Builder shortens digital signal processing (DSP) design cycles by helping you create the hardware representation of a DSP design in an algorithm-friendly development environment.

This IP core supports DSP Builder. Use the DSP Builder flow if you want to create a DSP Builder model that includes an IP core variation; use IP Catalog if you want to create an IP core variation that you can instantiate manually in your design. For more information about the DSP Builder flow, refer to the

Related Information

Using MegaCore Functions chapter in the DSP Builder Handbook.

CIC IP Core Functional Description

3

2014.12.15

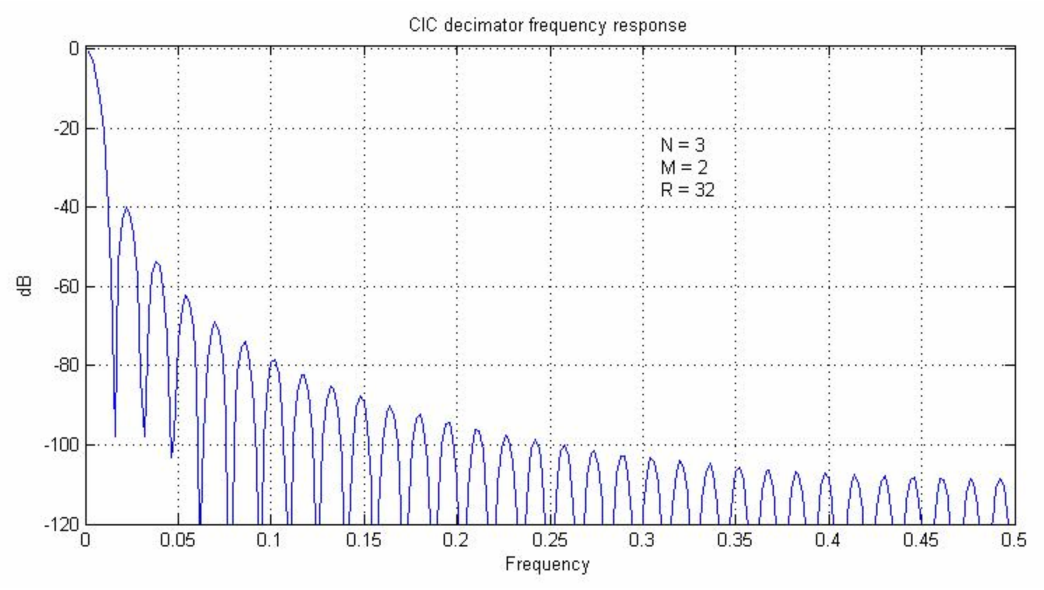
You can select either a decimation or interpolation CIC filter. A decimation CIC filter comprises a cascade of integrators (integrator), followed by a down sampling block (decimator) and a cascade of differentiators (called the differentiator or comb section). Similarly an interpolation CIC filter comprises a cascade of differentiators, followed by an up sampling block (interpolator) and a cascade of integrators.

In a CIC filter, both the integrator and comb sections have the same number of integrators and differentiators. Each pairing of integrator and differentiator is a stage. The number of stages (*N*) has a direct effect on the frequency response of a CIC filter. You determine the response of the filter by configuring:

- The number of stages *N*
- The rate change factor *R*
- The number of delays in the differentiators (differential delay) *M*. Generally, set the differential delay to 1 or 2.

Figure 3-1: Three-stage CIC Decimation Filter Frequency Response

CIC decimation filter with *N* = 3, *M* = 2 and *R* = 32



© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

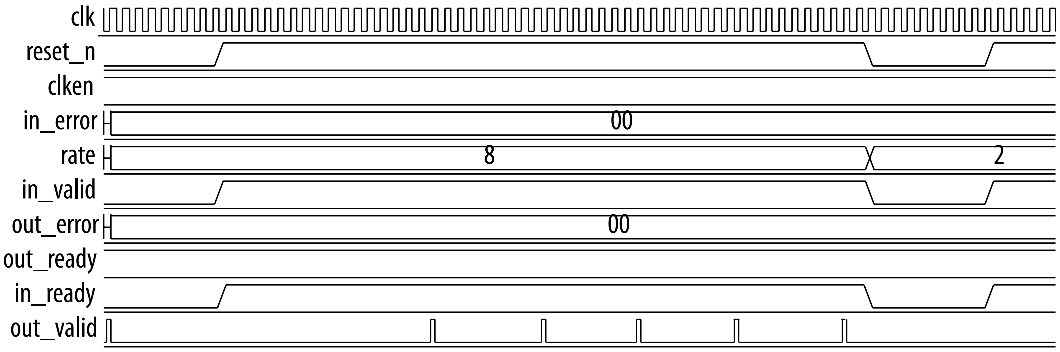
Variable Rate Change Factors

You can optionally set minimum and maximum values for the decimator or interpolator rate change factors and enable the rate change factors to be set at run time. With these options, the CIC provides an additional `rate` port that you can use to specify the rate change factor.

Note: With variable rate change factors, reset the IP core when you change the rate change factor, otherwise the CIC uses previous memory and register values. You cannot change the filter mode (interpolation or decimation) at run time.

Figure 3-2: Variable Rate Change Decimation CIC Filter Timing Diagram

The out_valid signal changes its period according to the variable rate change.



Multichannel Support

Often many channels of data in a digital signal processing (DSP) system require filtering by CIC filters with the same configuration. You can combine them into one filter, which shares the adders that exist in each stage and reduces the overall resource utilization.

Using a combined filter uses fewer resources than using many individual CIC filters. For example, a two-channel parallel filter requires two clock cycles to calculate two outputs. The resulting hardware needs to run at twice the data rate of an individual filter, which is especially useful for higher rate changes where adders grow particularly large.

Note: To minimize the number of logic elements, use a multiple input single output (MISO) architecture for decimation filters, and a single input multiple output (SIMO) architecture for interpolation filters.

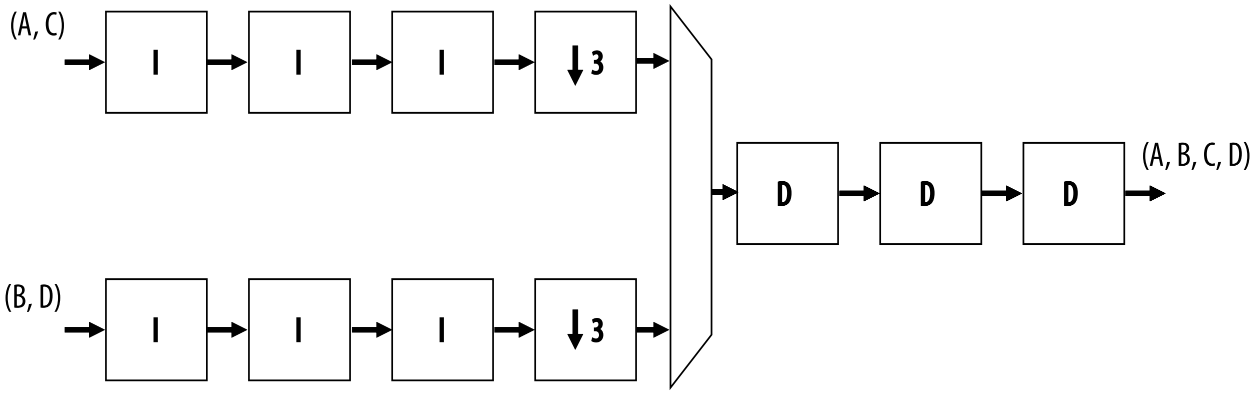
Multiple Input Single Output (MISO)

In many practical designs, channel signals come from different input interfaces. On each input interface, the same parameters including rate change factors apply to the channel data that the CIC filter is going to process. The CIC IP core allows multiple input single output (MISO) decimation filters, which allows the flexibility to exploit time sharing of the low-rate differentiator sections.

The CIC achieves time sharing by providing multiple input interfaces and processing chains for the high rate portions. It then combines all of the processing associated with the lower rate portions into a single processing chain. This strategy can lead to full utilization of the resources and represents the most efficient hardware implementation.

Figure 3-3: Multiple Input Single Output Architecture For Four Channels

The symbols A, B, C, D are multiplexed into one output A, B, C, D



The sampling frequency of the input data only allows time multiplexes of two channels per bus. Therefore, you must configure the CIC filter with two input interfaces. For two interfaces, the rate change factor must also be at least two to exploit this architecture. The CIC support up to 1,024 channels by using multiple input interfaces in this way.

Note: The CIC applies the MISO architecture when you select a decimation filter and the number of interfaces is greater than one.

Single Input Multiple Output (SIMO)

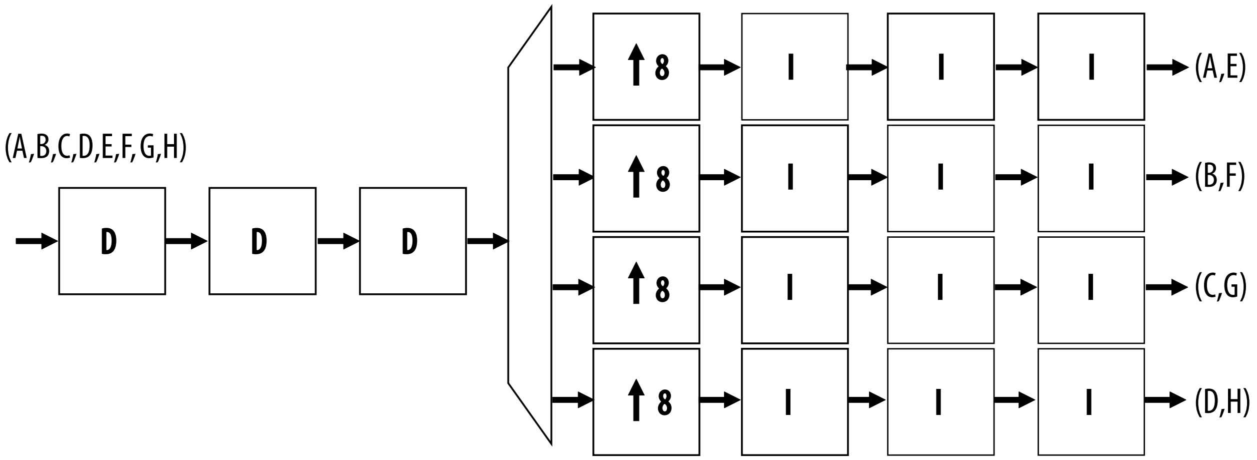
With single input multiple output (SIMO), all the channel signals presented for filtering come from a single input interface.

Like the MISO, you can share the low sampling rate differentiator section among more channels than the higher sampling frequency integrator sections. Therefore, this architecture features a single instance of the differentiator section and multiple parallel instances of the integrator sections.

After processing by the differentiator section, the CIC splits the channel signals into multiple parallel sections for processing in a high sampling frequency by the integrator sections.

Figure 3-4: Single Input Multiple Output Architecture with Eight Channels

The symbols A, B, C, D, E, F, G, H are demultiplexed into four outputs A, E; B, F; C, G; and D, H



The required sampling frequency of the output data only allows time multiplexes of two channels per bus. Therefore, you must configure the CIC filter with four output interfaces. The rate change factor must also be at least four to exploit this architecture, but this example shows a rate change of eight.

Note: The CIC applies a SIMO when you select an interpolation filter and the number of interfaces is greater than one.

The total number of input channels must be a multiple of the number of interfaces. To satisfy this requirement, you may need to either insert dummy channels or use more than one CIC IP core.

The CIC transfers data as packets using Avalon Avalon-ST interfaces.

Related Information

AN442: Tool Flow Design of Digital IF for Wireless Systems
An example design using multichannel MISO and SIMO architectures.

Output Options

You can select output options for the output data bit width and rounding options.

Output Data Width

If you select an output data width that is smaller than the full output resolution data width, apply the Hogenauer pruning technique to reduce the data widths across the filter stages and hence the overall resource utilization.

For a decimation filter, the gain at the output of the filter is:

$G = RM^N$

Therefore, the data width at the output stage for if full resolution is:

$B_{out} = B_{in} + N\log_2(RM)$

where Bin is the input data width.

Note: A data width of B_{out} is required for each integrator and differentiator for no data loss.

For an interpolation filter, the gain at each filter stage is:

$$G_i = \begin{cases} 2^i & i = 1, 2, \ldots, N \\ \dfrac{2^{2N-1}(RM)^{i-N}}{R} & i = N+1, \ldots, 2N \end{cases}$$

Hence the required data width at the ith stage is:

$W_i = [B_{in} + \log_2(G_i)]$

and the data width at the output stage is:

$B_{out} = [B_{in} + N\log_2(RM) - \log2(R)]$

where B_{in} is the input data width.

When the differential delay is one, the bit width at each integrator stage is increased by one to ensure stability.

For more information about these calculations, refer to Hogenauer, Eugene. *An Economical Class of Digital Filters For Decimation and Interpolation,* IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-29, pp. 155-162, April 1981.

Output Rounding

For high rate change factors, the maximum required data width for no data loss is large for many practical cases. To reduce the output data width to the input level, apply quantization at the end of the output stage. the CIC filter offers various rounding or saturation options. You can only apply these rounding options to the output stage of the filter. The data widths at the intermediate stages are not changed.

Table 3-1: Output Rounding Options

Option	Description
Truncation	The CIC drops the LSBs. (Equivalent to rounding to minus infinity.)
Convergent rounding	Also known as unbiased rounding. Rounds to the nearest even number. If the most significant deleted bit is one, and either the least significant of the remaining bits or at least one of the other deleted bits is one, then one is added to the remaining bits.
Round up	Also known as rounding to plus infinity. Adds the MSB of the discarded bits for positive and negative numbers via the carry in.
Saturation	Puts a limit value (upper limit in the case of overflow, or lower limit in the case of negative overflow) at the output when the input exceeds the allowed range. The upper limit is +2n-1 and lower limit is –2n

Hogenauer Pruning

Hogenauer pruning uses truncation in intermediate stages with the retained number of bits decreasing monotonically from stage to stage. The total error introduced is still no greater than the quantization error introduced by rounding the full precision output. This technique helps to reduce the number of logic cells used by the filter and gives better performance.

The existing algorithms for computing the Hogenauer bit width growth for large *N* and *R* values are computationally expensive.

For more information about these algorithms, refer to U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 2nd Edition, Spinger, 2004.

The CIC IP core has precalculated Hogenauer pruning bit widths. The CIC does not have to calculate Hogenauer pruning bit widths if you enable Hogenauer pruning for a decimation filter.

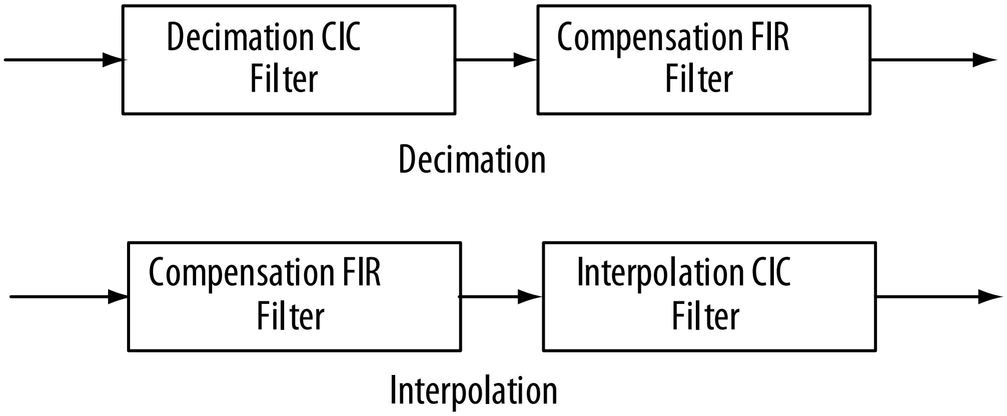
Note: Hogenauer pruning is only available to decimation filters when the selected output data width is smaller than the full output resolution data width.

FIR Filter Compensation Coefficients

CIC filters have a low-pass filter characteristic. Three parameters (the rate change factor *R*, the number of stages *N*, and the differential delay *M*) allow you to change the passband characteristics and aliasing or imaging rejection.

Typically, decimation or interpolation filtering applications require flat passband and narrow transition region filter performance. However, the CIC filter has drooping passband gains and wide transition regions. To overcome these problems connect the decimation or interpolation CIC filter to a compensation FIR filter, which narrows the output bandwidth and flattens the passband gain.

Figure 3-5: Using a CIC Compensation FIR Filter



You can use a frequency sampling method to determine the coefficients of a FIR filter that equalizes the undesirable passband droop of the CIC and construct an ideal frequency response.

Determine the ideal frequency response by sampling the normalized magnitude response of the CIC filter before inverting the response.

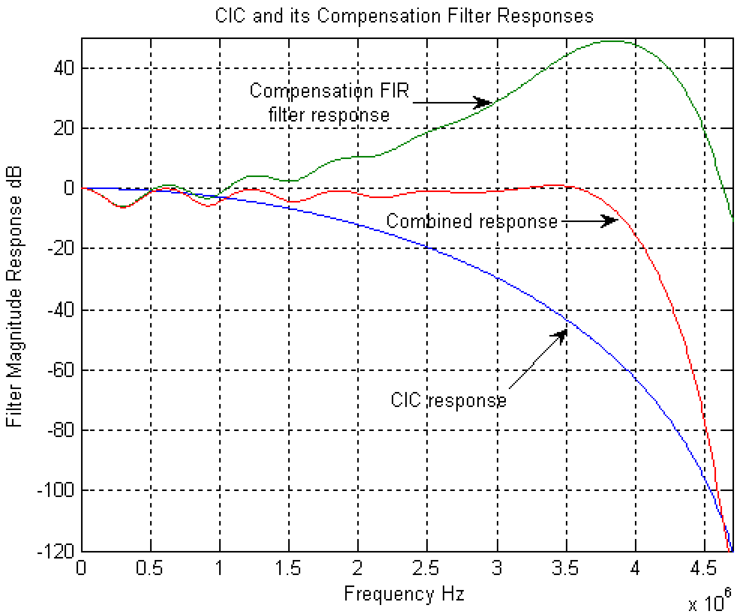
Generally, only equalize the response in the passband, but you can sample further than the passband to fine tune the cascaded response of the filter chain.

The CIC IP core generates a MATLAB script ***<variation_name>*_fir_comp_coeff.m** in the project directory. You can run this script in MATLAB to generate FIR coefficients that provide appropriate passband equalization. The generated coefficients are saved in a text file, for use by the Altera FIR Compiler MegaCore function.

The MATLAB script requires the following parameters for the compensation FIR filter:

- L: FIR filter length, which is same as the number of taps or the number of coefficients
- $F_{S:}$ FIR filter sample rate in Hz before decimation/interpolation
- F_C: FIR filter cutoff frequency in Hz
- B: Coefficient bit width if coefficients are written in fixed-point numbers

Figure 3-6: CIC and Compensation Filter Responses



Related Information

[AN455: Understanding CIC Compensation Filters]

CIC IP Core Parameters

Table 3-2: CIC IP Core Parameters

Parameter	Value	Description
Filter Specification		
Filter type	Decimator, Interpolator	Selects a decimator or interpolator.

Parameter	Value	Description
Number of stages	1 to 12	Specifies the required number of stages.
Differential delay	1, 2	Specifies the differential delay in cycles.
Enable variable rate change factor	On or Off	Turn on to enable a variable rate change factor that you can change at runtime. When this option is on, the **Rate change factor** parameter is not available but you can specify minimum and maximum values.
Rate change factor	2 to 32000	Specifies the rate change factor.
Number of interfaces	1 to 128	Specifies the number of MISO inputs or SIMO outputs The product of the **Number of interfaces** and the **Number of channels per interface** must be no more than 1024.
Number of channels per interface	1 to 1024	Specifies the number of channels per interface. The product of the **Number of interfaces** and the **Number of channels per interface** must be no more than 1024
Interface Specification		
Input data width	1 to 32	Specifies the input data width in bits.
Output Rounding Options	None, Truncation, Convergent rounding, Rounding up, Saturation, Hogenauer pruning	Selects the required rounding output mode. Select **None** for full output resolution. The saturation limit is the maximum value for overflow or the minimum value for negative overflow. Hogenauer pruning is available only when a **Decimator** filter type is selected in the **Architecture** page.
Output data width	1 to calculated maximum data width	Specifies the output data width in bits.
implementation Options		
Integrator data storage	Logic Element, Memory	Selects whether to implement the integrator data storage as logic elements or memory. The **Memory** option is available for integrator data storage when the **Number of channels per interface** is greater than 4.
RAM type of integrator data storage	AUTO, M9K, M10K, M20K, M144K, MLAB	When you select **Memory**, you can select the RAM type for integrator data storage. The **Memory** option is available for integrator data storage when the **Number of channels per interface** is greater than 4.

Parameter	Value	Description
Differentiator data storage	Logic Element, Memory	Selects whether to implement the differentiator data storage as logic elements or memory. The **Memory** option is available for differentiator data storage when the product of the **Differential delay**, **Number of channels per interface** and **Number of interfaces** is greater than 4.
RAM type of differentiator data storage	AUTO, M9K, M10K, M20K, M144K, MLAB	When you select **Memory**, you can select the RAM type for differentiator data storage. The options available depend on the target device family. When AUTO is selected, the Quartus II software automatically selects the optimum RAM type for the currently selected device family.
Pipeline stages per integrator	—	Enter the pipeline stages per integrator. This option is available when the **Number of channels per interface** is greater than or equal to 2 (or greater than or equal to 6, when you select the **Memory** option for integrator data storage). Use this option for multichannel designs that have large input bit width and require high f_{MAX}, but not for designs targeting Cyclone devices.
Pipeline stages per integrator	1 to 4	Specifies the number of pipeline stages used by each integrator. Adding additional integrators can improve f_{MAX} but increases the resource utilization. The maximum number of pipeline stages depends on the number of channels and whether you select **Memory** or **Logic Cells** for integrator data storage. For **Memory**, the maximum number of pipeline stages equals the number of channels minus 5. For **Logic Cells**, the maximum number of pipeline stages equals the number of channels.

CIC IP Core Interfaces and Signals

Table 3-3: Avalon-ST Interface Parameters

All parameters not explicitly listed have undefined values.

Parameter Name	Value
READY_LATENCY	0
BITS_PER_SYMBOL	Data width

Parameter Name	Value
SYMBOLS_PER_BEAT	Single input, single output architectures, have one symbol per beat at the source and the sink. MISO architectures have <*number of interfaces*> symbols per beat at the sink, and a single symbol per beat at the source. SIMO architectures have <*number of interfaces*> symbols per beat at the source, and a single symbol per beat at the sink.
SYMBOL_TYPE	Signed
ERROR_DESCRIPTION	• 00: No error • 01: Missing startofpacket (SOP) • 10: Missing endofpacket (EOP) • 11: Unexpected EOP or any other error

Related Information

Avalon Interface Specifications

For more information about the Avalon-ST interface

Avalon-ST Interfaces in DSP IP Cores

Avalon-ST interfaces define a standard, flexible, and modular protocol for data transfers from a source interface to a sink interface.

The input interface is an Avalon-ST sink and the output interface is an Avalon-ST source. The Avalon-ST interface supports packet transfers with packets interleaved across multiple channels.

Avalon-ST interface signals can describe traditional streaming interfaces supporting a single stream of data without knowledge of channels or packet boundaries. Such interfaces typically contain data, ready, and valid signals. Avalon-ST interfaces can also support more complex protocols for burst and packet transfers with packets interleaved across multiple channels. The Avalon-ST interface inherently synchronizes multichannel designs, which allows you to achieve efficient, time-multiplexed implementations without having to implement complex control logic.

Avalon-ST interfaces support backpressure, which is a flow control mechanism where a sink can signal to a source to stop sending data. The sink typically uses backpressure to stop the flow of data when its FIFO buffers are full or when it has congestion on its output.

Related Information

- Avalon Interface Specifications

CIC IP Core Signals

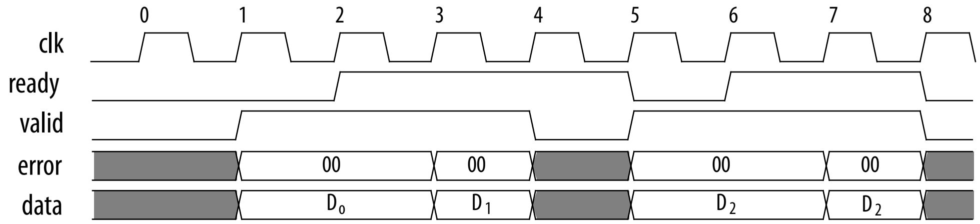
Table 3-4: CIC IP Core Signals

Signal	Direction	Description
av_st_in_data	Output	In Qsys systems, this Avalon-ST-compliant data bus includes all the Avalon-ST input data signals. For multi-interface designs Interface 0 is in the MSB; Interface *N* is the LSB.
clk	Input	Clock signal for all internal registers.
clken	Input	Optional top-level clock enable.
reset_n	Input	Active low reset signal. You must always reset the CIC MegaCore function before receiving data. If not, the CIC filter may produce unexpected results because of feedback signals.
in_data	Input	Sample input. For multiple input cases, the input data ports are in0_data, in1_data, and so on.
in_endofpacket	Input	Marks the end of the incoming sample group. For *N* channels, the end of packet signal must be high when the sample belonging to the last channel, channel *N*-1, is presented at in_data.
in_error	Input	Error signal indicating Avalon-ST protocol violations on input side: • 00: No error • 01: Missing start of packet • 10: Missing end of packet • 11: Unexpected end of packet Other types of error are also marked as 11.
in_ready	Output	Indicates when the IP core can accept data.
in_startof-packet	Input	Marks the start of the incoming sample group. The start of packet is interpreted as a sample from channel 0.
in_valid	Input	Asserted when data at in_data is valid. When in_valid is not asserted, processing is stopped until valid is re-asserted. If clken is 0, in_valid is not be asserted.
av_st_out_data	Output	In Qsys systems, this Avalon-ST-compliant data bus includes all the Avalon-ST output data signals. For multi-interface designs Interface 0 is in the MSB; Interface *N* is the LSB.
out_channel	Output	Specifies the channel whose result is presented at out_data.
out_data	Output	Filter output. The data width depends on the parameter settings. For multiple output cases, the output data ports are named as out0_data, out1_data, and so on.

Signal	Directio n	Description
`out_endofpacket`	Output	Marks the end of the outgoing result group. If '1', a result corresponding to channel N-1 is output, where N is the number of channels.
`out_error`	Output	Error signal indicating Avalon-ST protocol violations on source side: • 00: No error • 01: Missing start of packet • 10: Missing end of packet • 11: Unexpected end of packet Other types of errors may also be marked as 11.
`out_ready`	Input	Asserted by the downstream module if it is able to accept data.
`out_startof-packet`	Output	Marks the start of the outgoing result group. If '1', a result corresponding to channel 0 is output.
`out_valid`	Output	Asserted by the IP core when there is valid data to output.
`rate`	Input	This signal is available when the variable rate change factor option is enabled. You can use it to change the decimation or interpolation rate during run time. It has the size Ceil($\log_2$(maximum rate)) .

Avalon-ST Interface Data Transfer Timing

Figure 3-7: Avalon-ST Interface Timing with READY_LATENCY=0



The source provides data and asserts `valid` on cycle 1, even though the sink is not ready. The source waits until cycle 2, when the sink does assert `ready`, before moving onto the next data cycle. In cycle 3, the source drives data on the same cycle and because the sink is ready to receive it, the transfer occurs immediately. In cycle 4, the sink asserts `ready`, but the source does not drive valid data.

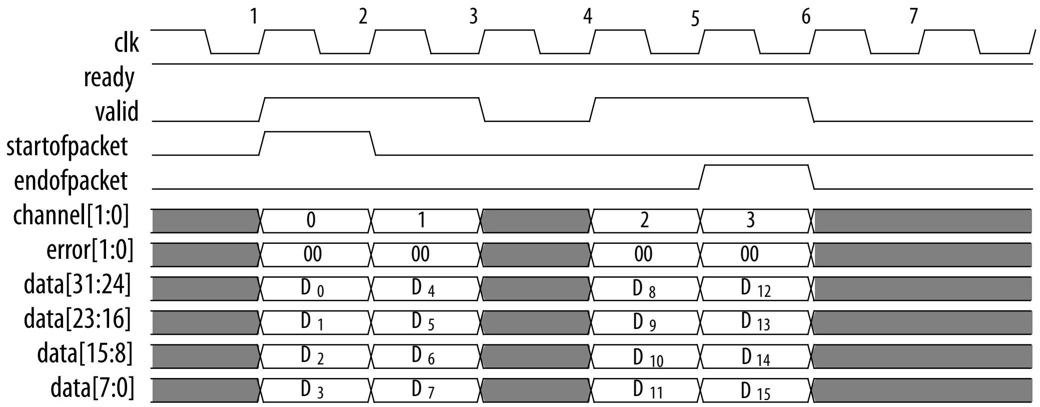
Packet Data Transfers

A beat is the transfer of one unit of data between a source and sink interface. This unit of data may consist of one or more symbols and makes it is possible to support modules that convey more than one piece of information about each valid cycle.

Packet data transfers are used for multichannel transfers. Two additional signals (`startofpacket` and `endofpacket`) are defined to implement the packet transfer.

The multiple symbols per beat scenario applies to both the sink interface on MISO CIC filters and the source interface of SIMO CIC filters. All other interfaces operate with a single symbol per beat, but the interfaces also support multiple channels using packets.

Figure 3-8: Packet Data TransferFour symbols are transferred on each beat. The data transfer occurs on cycles 1, 2, 4, and 5, when both ready and valid are asserted.



During cycle 1, the CIC IP core asserts `startofpacket`, and transfers the first four bytes of packet. During cycle 5, the CIC IP core asserts `endofpacket` indicating that this is the end of the packet. The `channel` signal indicates the channel index associated with the data. For example, on cycle 1, the data D_0, D_1, D_2, and D_3 associated with channel 0 are available.

Document Revision History 4

2014.12.15

UG-CIC Subscribe Send Feedback

CIC IP Core User Guide revision history.

Table 4-1:

Date	Version	Changes Made
2014.12.15	14.1	• Added final support for Arria 10 devices • Reordered parameters tables to match wizard
August 2014	14.0 Arria 10 Edition	• Added support for Arria 10 devices. • Added new av_st_in_data and av_st_out_data bus descriptions. • Added Arria 10 generated files description. • Removed table with generated file descriptions.
June 2014	14.0	• Removed support for Cyclone III and Stratix III devices • Added instructions for using IP Catalog
November 2013	13.1	• Removed support for the following devices: • Arria • Cyclone II • HardCopy II, HardCopy III, and HardCopy IV • Stratix, Stratix II, Stratix GX, and Stratix II GX • Added full support for the following devices: • Arria V • Stratix V
November 2012	12.1	Added support for Arria V GZ devices.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered