Median Filter IP Core User's Guide

December 2010
IPUG87_01.0

Table of Contents

Introduction

This user's guide provides a description of the Median Filter IP core. Median filtering is a popular method of noise removal, employed extensively in applications involving speech, signal and image processing. This non-linear technique has proven to be a good alternative to linear filtering as it can effectively suppress impulse noise while preserving edge information. The core's flexible architecture supports a wide variety of video frame sizes on LatticeECP2/S™, LatticeECP2M/S™, LatticeXP2™, and LatticeECP3™ devices. A simple IO handshake makes the core suitable for either streaming or bursty input video data.

Quick Facts

Table 1-1 gives quick facts about the Median Filter IP core.

Table 1-1. Median Filter IP Core Quick Facts

			Median Filter Core		
			Frame Size:320x240 Window Size: 3x3	**Frame Size:256x256 Window Size: 5x5**	**Frame Size:128x128 Window Size: 7x7**
Core Requirements	FPGA Familes Supported		LatticeECP2/S, LatticeECP2M/S, LatticeXP2, LatticeECP3		
	Minimum Device Required		LFXP2-5E LFE2-6E LFE2M20E LFE3-17EA	LFXP2-5E LFE2-6E LFE2M20E LFE3-17EA	LFXP2-17E LFE2-12E LFE2M20E LFE3-17EA
Resource Utilization	LatticeXP2	LUTs	700	2900	11500
		EBRs	1	1	2
		Registers	550	2200	6900
		sysDSP blocks	0		
	LatticeECP2/S	LUTs	700	2900	11500
		EBRs	1	1	2
		Registers	550	2200	6900
		sysDSP blocks	0		
	LatticeECP2M/S	LUTs	700	2900	11500
		EBRs	1	1	2
		Registers	550	2200	6900
		sysDSP blocks	0		
	LatticeECP3	LUTs	700	2900	11500
		EBRs	1	1	2
		Registers	550	2200	6900
		sysDSP blocks	0		
Design Tool Support	Lattice Implementation		Lattice Diamond™ 1.1 or ispLEVER® 8.1SP1		
	Synthesis		Synopsys® Synplify™ Pro for Lattice D-2010.03L-SP1		
	Simulation		Aldec® Active-HDL™ 8.2 Lattice Edition II		
			Mentor Graphics® ModelSim™ SE 6.3F		

Features

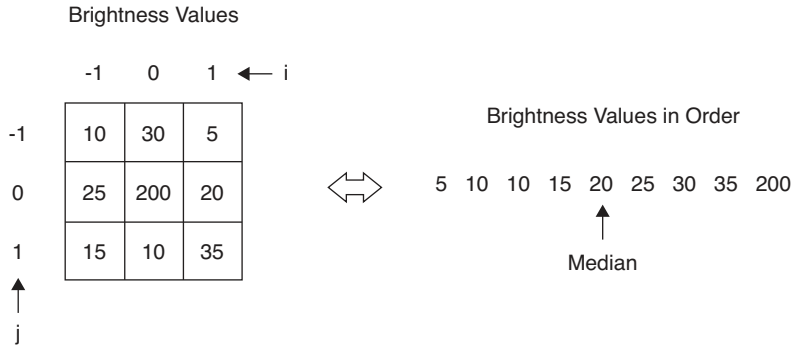
- Single color plane
- Three filter window sizes: 3x3, 5x5 and 7x7
- Configurable input data width

- Input frame size set at compile-time
- Static active region selection
- Edge mode handling: COPY, MIRROR or VALUE
- Optional clock enable and synchronous reset ports

Key Concepts

Median filter is a spatial filtering operation, so it uses a 2-D mask that is applied to each pixel in the input image. To apply the mask means to centre it on a pixel, evaluating the covered pixel brightness and determining which brightness value is the median value. The median value is determined by placing the brightness in ascending order and selecting the centre value. The obtained median value will be the value for that pixel in the output image. An example is shown in Figure 2-1.

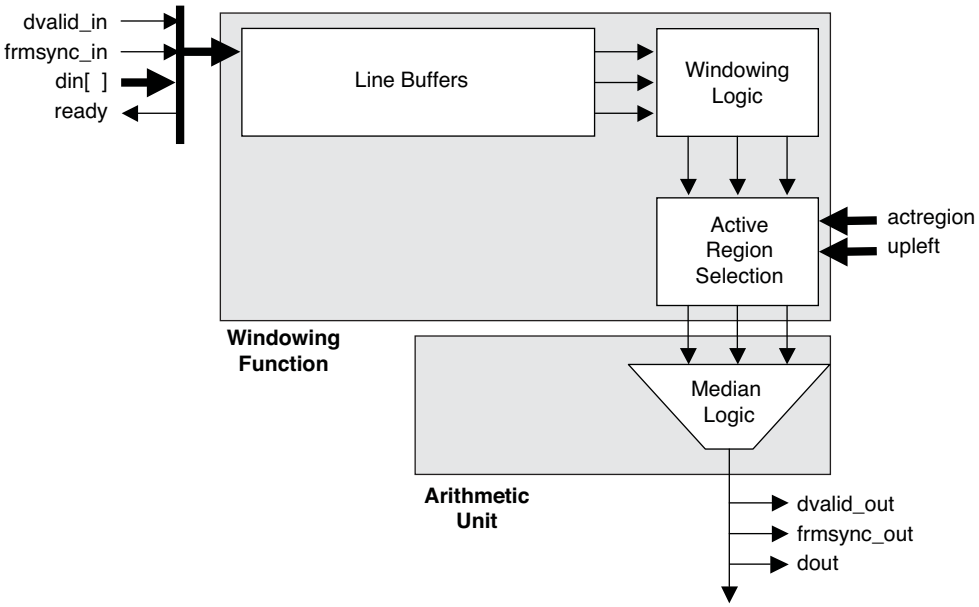
Figure 2-1. Example of Median Value



Block Diagram

The high-level architecture of the Median Filter core is diagrammed in Figure 2-2.

Figure 2-2. Median Filter IP Core Block Diagram

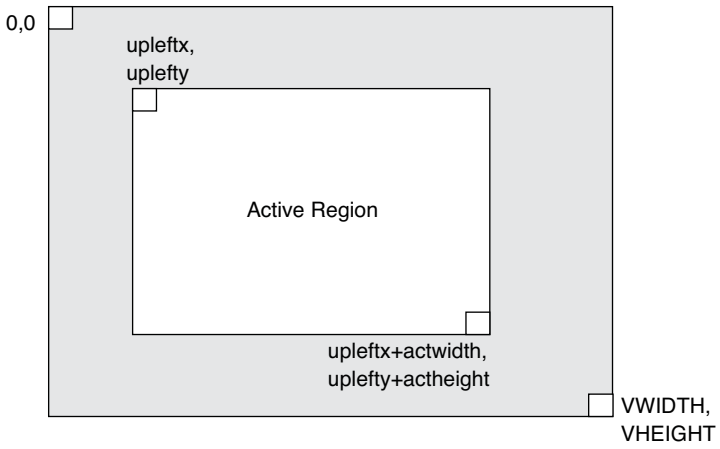


Input data is stored in line buffers, then passed to windowing logic for edge mode handling and data alignment. Optional control inputs allow real-time specification of the portion of the input frame used to generate output pixels (referred to as the “active region”). Windowed data are sent to the median arithmetic unit which chooses the median input pixel value.

Active Region Selection

The Median Filter core may be configured to allow the user to dynamically alter the coordinates of the active region of the input frame. The active region concept is illustrated in Figure 2-3.

Figure 2-3. Active Region



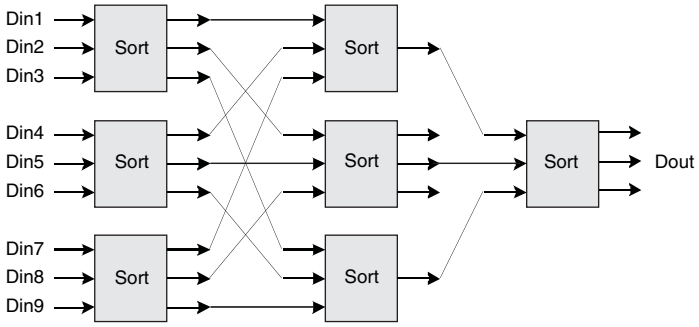
The upleftx and uplefty ports set the coordinates of the first pixel in the input frame that will have a corresponding pixel in the output frame. The actwidth and actheight ports determine the region of pixels in the input frame that will have corresponding pixels in the output frame. Both sets of inputs – upleft and active region – are synchronized internally and delivered to the core logic at the appropriate time to avoid anomalies when moving from frame to frame.

Median Arithmetic Unit

Different window sizes have different fast algorithms and implementations. The following description is based on the 3×3 window size.

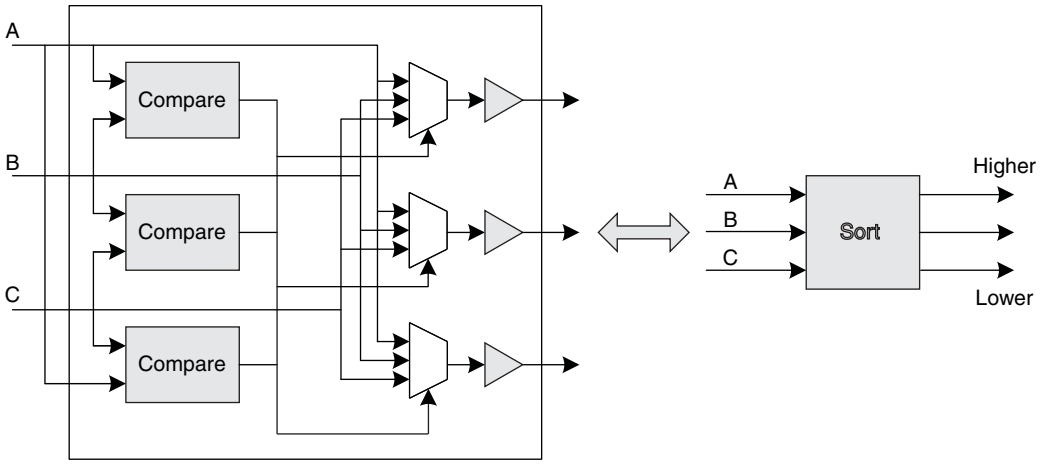
The scheme of median arithmetic unit of 3×3 window size is shown in Figure 2-4, which is much better than other solutions, since it needs a much lower number of basic nodes. This scheme uses the minimum exchange network required to produce the median from nine pixels by performing a partial sorting.

Figure 2-4. 3x3 Median Arithmetic Unit Scheme



Each basic node allows sorting of three elements. To do that, each node compares the three elements by means of three comparators, using its output in three 3:1 multiplexers, as shwon in Figure 2-5.

Figure 2-5. Scheme for Each Basic Node



Primary I/O

Table 2-1. Primary I/O

Port	Size	I/O	Description
Global Signals			
clk	1	I	System clock
rstn	1	I	System wide asynchronous active-low reset signal
ce	1	I	Active high clock enable (optional)
sr	1	I	Active high synchronous reset (optional)
Video Input			
ready	1	O	Core is ready for input
dvalid_in	1	I	Input valid
frmsync_in	1	I	Current pixel is at row 0, column 0
din	4 - 24	I	Pixel data in
Video Output			
dvalid_out	1	O	Output valid
frmsync_out	1	O	Current output pixel is at row 0, column 0
dout	4 - 24	O	Pixel data out
Dynamic Frame Size and Active Region Controls (Optional)			
pwrite	1	I	Internal regs write enable
paddr	4	I	Internal regs address
pwdat	8, 16, 32, 64	I	Internal regs write data
prdat	8, 16, 32, 64	O	Internal regs read data
Miscellaneous			
tags_in	TAGS_WIDTH	I	Tags input
tags_out	TAGS_WIDTH	O	Tags output

Interface Descriptions

Video Input/Output

The Median Filter uses a simple handshake to pass pixel data into the core. The core asserts its ready output when it is ready to receive data. When the driving module has data to give the core, it drives the core's dvalid_in port to a '1' synchronously with the rising edge of the clk signal, providing the input pixel data on port din. The frmsync_in

input should be driven to a '1' during the clock cycle when the first pixel of the first row in the incoming video frame is active.

Correspondingly, dvalid_out is active when valid output pixel data is available on dout, and frmsync_out marks the first pixel, first row of the output video frame.

Timing Specifications

Timing diagrams for the Median Filter IP core are given in Figure 2-6 and Figure 2-7.

Figure 2-6. Timing Diagram for Median Filter with Continuous Inputs

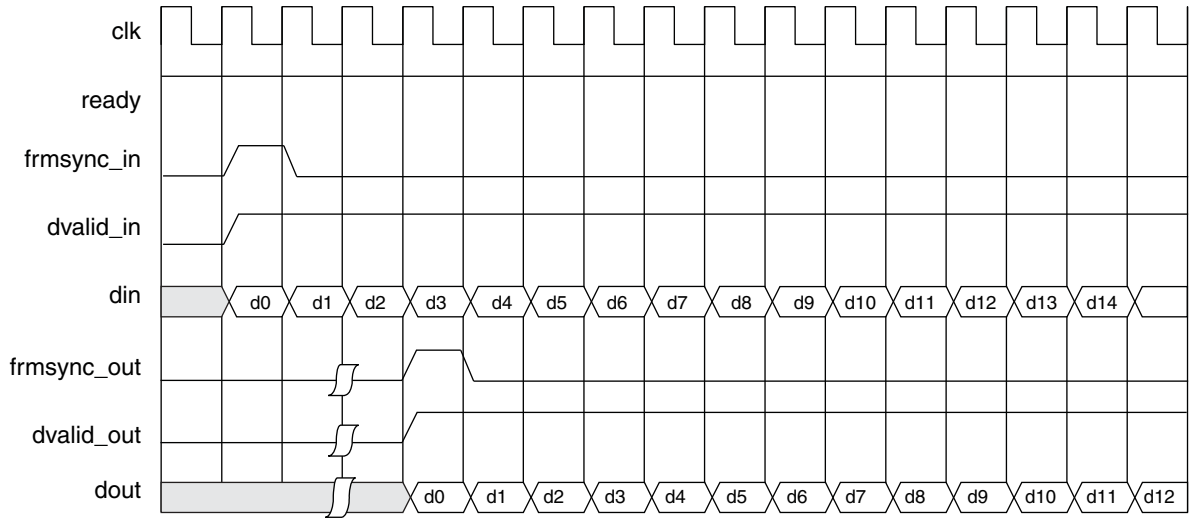
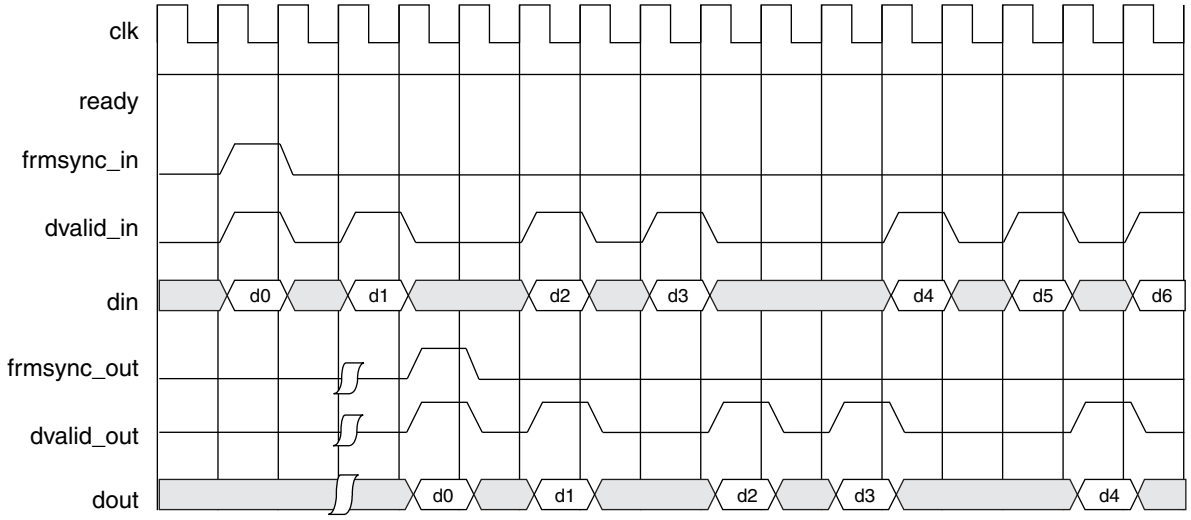


Figure 2-7. Timing Diagram for Median Filter with Gaps in Inputs



Parameter Settings

The IPexpress™ tool is used to create IP and architectural modules in the Diamond or ispLEVER software. Refer to "IP Core Generation" on page 13 for a description of how to generate the IP. The Median Filter IP core can be customized to suit a specific application by adjusting parameters prior to core generation. Since the values of some parameters affect the size of the resultant core, the maximum value for these parameters may be limited by the size of the target device.

Table 3-1 provided the list of user configurable parameters for the Median Filter IP core.

Table 3-1. Median Filter IP Core Parameters

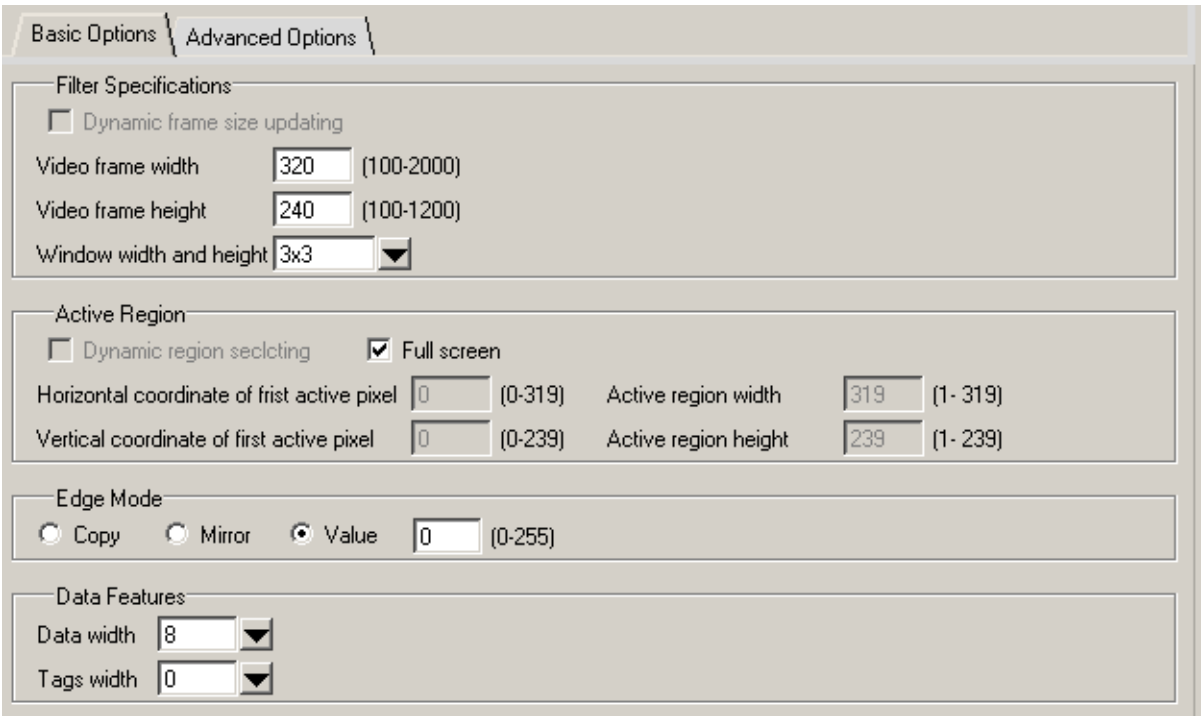
Parameter	Range	Default
Data width	4 - 24	8
Video frame width	100 - 1200	320
Video frame height	100 - 1200	240
Dynamic frame size updating	0 or 1	0
Dynamic region selecting	0 or 1	0
Horizontal coordinates of first active pixel	0 – (VWIDTH-1)	0
Vertical coordinates of first active pixel	0 – (VHEIGHT-1)	0
Active region width	1 – (VWIDTH-UPLEFTX-1)	VWIDTH-1
Active region height	1 – (VHEIGHT-UPLEFTY-1)	VHEIGHT-1
Window width and height	3x3, 5x5, 7x7	3x3
Edge mode	VALUE, COPY, MIRROR	VALUE
Edge value	0 – (1<<DWIDTH)-1	0
Tags width	0	0
Internal regs data bus width	8, 16, 32, 64	8
Input buffer type	EBR, Distributed	EBR

Basic Options Tab

The Basic Options tab provides settings for video frame and window parameters, data width, and edge mode options.

Note: Median Filter core version 1.0 does not support dynamic frame size updating and dynamic region selecting. Figure 3-1 shows the contents of the Basic Options tab.

Figure 3-1. Basic Options Tab



Filter Specifications

This section provides settings that define the input and output frame sizes, as well as what to do when the sampling window straddles the frame edges.

Video Frame Width
This parameter defines the input video frame size width.

Video Frame Height
This parameter defines the the input video frame size height.

Window Width and Height
This parameter defines the filter window size.

Active Region

Dynamic region selection
This checkbox enables the feature and adds the upleftx/y and actwidth/height input ports.

Full screen
This checkbox sets the active region to the video frame size.

Horizontal Coordinates of First Active Pixel/Vertical Coordinates of First Active Pixel
These coordiates set the upper left corner of the active region.

Active Region Width/Active Region Height
These coordiates set the size of the active region.

Edge Mode
Radio buttons select Copy, Mirror or Value for edge mode. If Value is selected, the entry field is the pixel value used for window locations that overlap the frame boundary.

Data Features

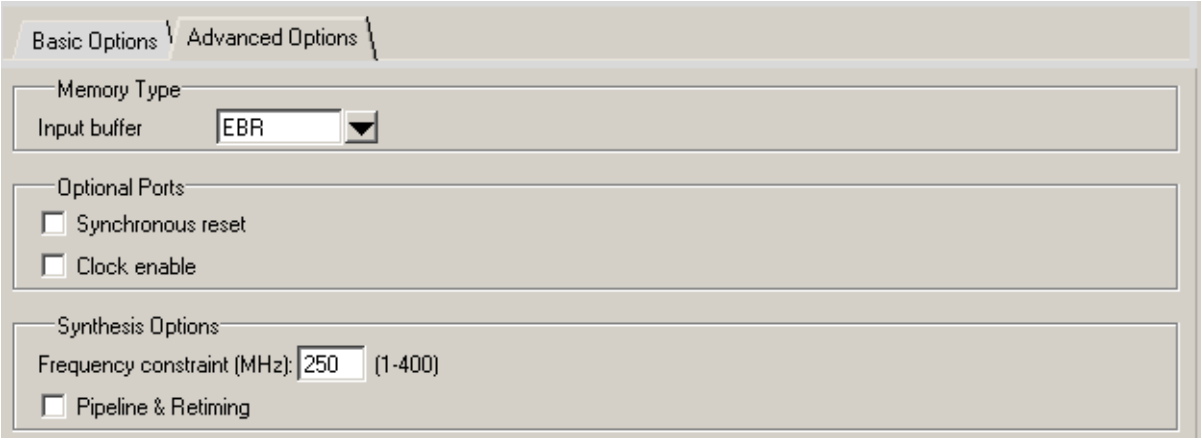
Data width
This parameter sets the width of the incoming pixel data.

Tags width
This parameter sets the width of the tags_in and tags_out ports.

Advanced Options Tab

Figure 3-2 shows the contents of the Advanced Options tab.

Figure 3-2. Advanced Options Tab



Memory Type

Input Buffer
This parameter selects between EBR and Distributed RAM for the line buffers.

Optional Ports

Synchronous Reset
This checkbox enables synchronous reset and synchronous reset input port.

Clock Enable
This checkbox enables clock enable and clock enable input port.

Synthesis Options

Frequency Constraint
This parameter sets the target clock frequency in MHz.

Pipeline & Retiming
This checkbox enables pipeline and retiming features in the generation flow.

IP Core Generation

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

Licensing the IP Core

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

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

Users may download and generate the Median Filter IP core and fully evaluate the core through functional simulation and implementation (synthesis, map, place and route) without an IP license. The Median Filter IP core also supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core that operate in hardware for a limited time (approximately four hours) without requiring an IP license. See "Hardware Evaluation" on page 17 for further details. However, a license is required to enable timing simulation, to open the design in the Diamond or ispLEVER EPIC tool, and to generate bitstreams that do not include the hardware evaluation timeout limitation.

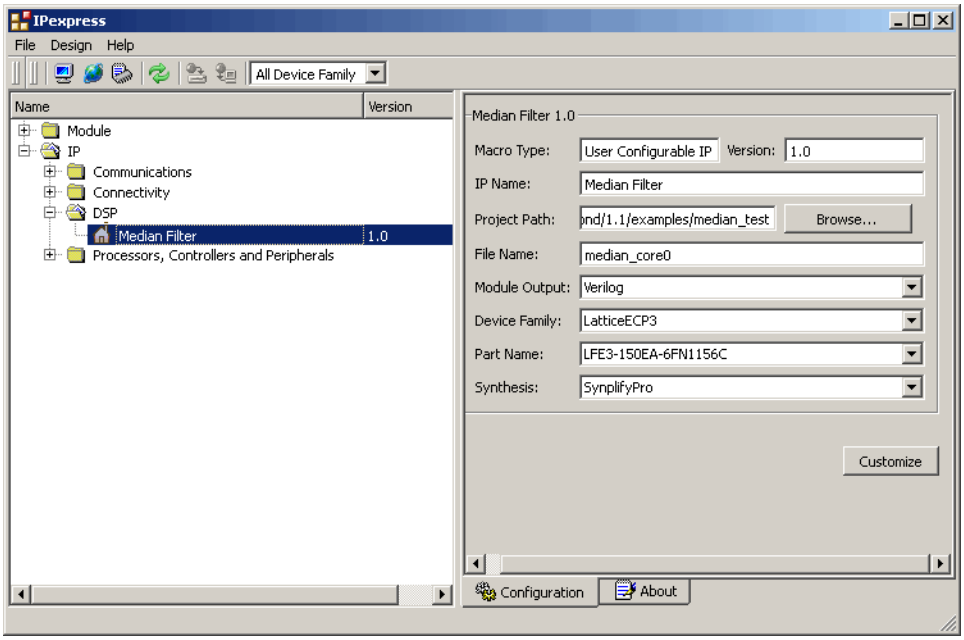
Getting Started

The Median Filter IP core is available for download from the Lattice IP Server using the IPexpress tool. The IP files are automatically installed using ispUPDATE technology in any customer-specified directory. After the IP core has been installed, the IP core will be available in the IPexpress GUI dialog box shown in Figure 4-1.

The IPexpress tool GUI dialog box for the Median Filter IP core is shown in Figure 4-1. To generate a specific IP core configuration the user specifies:

- **Project Path** – Path to the directory where the generated IP files will be located.
- **File Name** – "username" designation given to the generated IP core and corresponding folders and files.
- **(Diamond) Module Output** – Verilog or VHDL.
- **(ispLEVER) Design Entry Type** – Verilog HDL or VHDL.
- **Device Family** – Device family to which IP is to be targeted (e.g. Lattice ECP2M, LatticeECP3, etc.). Only families that support the particular IP core are listed.
- **Part Name** – Specific targeted part within the selected device family.

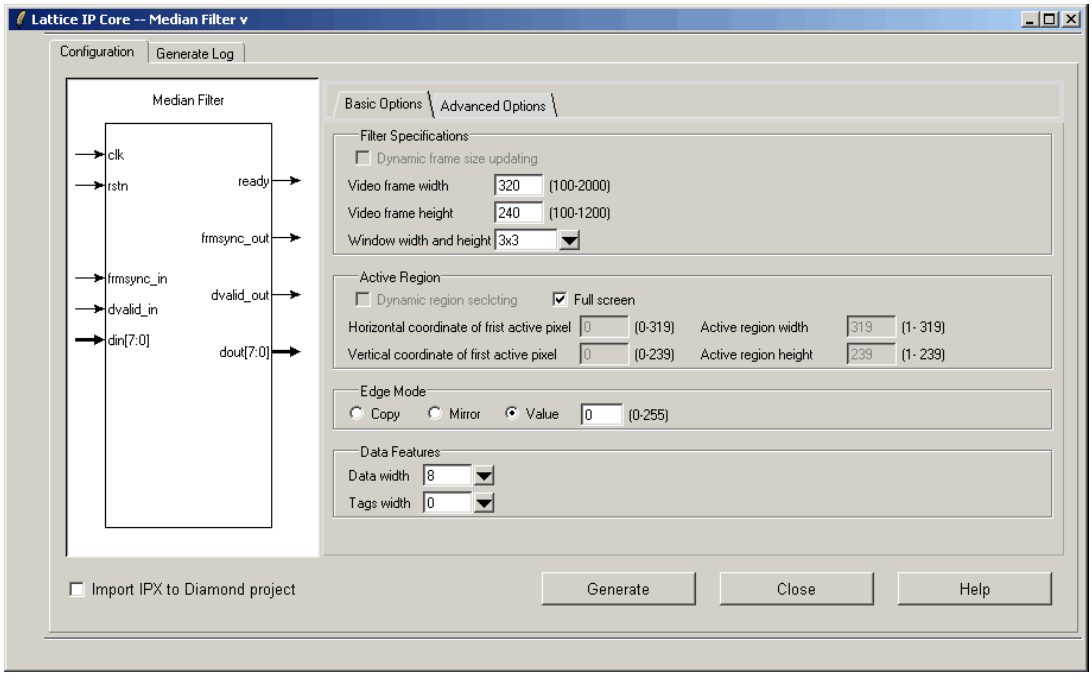
Figure 4-1. IPexpress Dialog Box (Diamond Version)



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

To create a custom configuration, the user clicks the **Customize** button in the IPexpress tool dialog box to display the Median Filter IP core Configuration GUI, as shown in Figure 4-2. From this dialog box, the user can select the IP parameter options specific to their application. Refer to "Parameter Settings" on page 136 for more information on the Median Filter IP core parameter settings.

Figure 4-2. Configuration GUI (Diamond Version)



IPexpress-Created Files and Top Level Directory Structure

When the user clicks the **Generate** button in the IP Configuration dialog box, the IP core and supporting files are generated in the specified "Project Path" directory. The directory structure of the generated files is shown in Figure 4-3. This example shows the directory structure generated with the Median Filter IP for LatticeECP3 device.

Figure 4-3. Median Filter IP Core Directory Structure

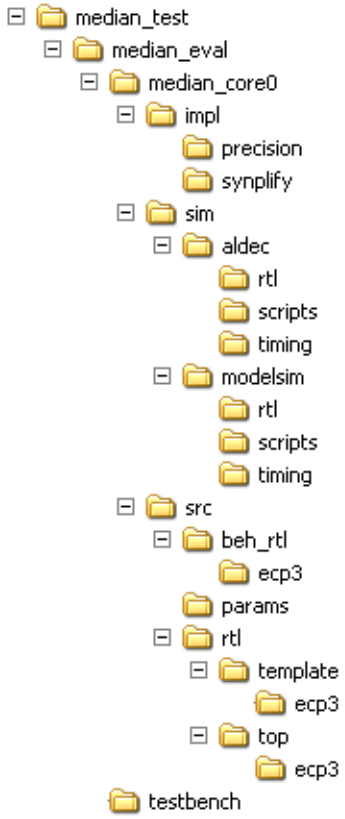


Table 4-1 provides a list of key files and directories created by the IPexpress tool and how they are used. The IPexpress tool creates several files that are used throughout the design cycle. The names of most of the created files are customized to the user's module name specified in the IPexpress tool.

Table 4-1. File List

File	Description
<*username*>.lpc	This file contains the IPexpress tool options used to recreate or modify the core in the IPexpress tool.
<*username*>.ipx	The IPX file holds references to all of the elements of an IP or Module after it is generated from the IPexpress tool (Diamond version only). The file is used to bring in the appropriate files during the design implementation and analysis. It is also used to re-load parameter settings into the IP/Module generation GUI when an IP/Module is being re-generated.
<*username*>.ngo	This file provides the synthesized IP core.
<*username*>_bb.v/.vhd	This file provides the synthesis black box for the user's synthesis.
<*username*>_inst.v/.vhd	This file provides an instance template for the Median Filter IP core.
<*username*>_beh.v/.vhd	This file provides the front-end simulation library for the Median Filter IP core.

Table 4-2 provides a list of key additional files providing IP core generation status information and command line generation capability are generated in the user's project directory.

Table 4-2. Additional Files

File	Description
<*username*>_generate.tcl	This file is created when the GUI "Generate" button is pushed. This file may be run from command line.
<*username*>_generate.log	This is the synthesis and map log file.
<*username*>_gen.log	This is the IPexpress IP generation log file

Instantiating the Core

The generated Median Filter IP core package includes black-box (<*username*>_bb.v) and instance (<*username*>_inst.v) templates that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file that can be used as an instantiation template for the IP core is provided in \<*project_dir*>\median_eval\<*username*>\src\rtl\top. Users may also use this top-level reference as the starting template for the top-level for their complete design.

Running Functional Simulation

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

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

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

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

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

Synthesizing and Implementing the Core in a Top-Level Design

Synthesis support for the Median Filter IP core is provided for Mentor Graphics Precision or Synopsys Synplify. The Median Filter IP core itself is synthesized and is provided in NGO format when the core is generated in IPexpress. Users may synthesize the core in their own top-level design by instantiating the core in their top-level as described previously and then synthesizing the entire design with either Synplify or Precision RTL synthesis.

The top-level file <*username*>_eval_top.v provided in
\<*project_dir*>\median_eval\<*username*>\src\top
supports the ability to implement the Median Filter core in isolation. Push-button implementation of this top-level design with either Synplify or Precision RTL Synthesis is supported via the project files
<username>_eval.ldf (Diamond) or .syn (ispLEVER) located in the
\<*project_dir*>\median_eval\<*username*>\impl\synplify
and the
\<*project_dir*>\median_eval\<*username*>\impl\precision directories, respectively.

To use this project file in Diamond:

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

To use this project file in ispLEVER:

1. Choose **File > Open Project.**
2. Browse to \<*project_dir*>\median_eval\<*username*>\impl\(synplify or precision) in the Open Project dialog box.
3. Select and open <*username*>.syn. At this point, all of the files needed to support top-level synthesis and implementation will be imported to the project.
4. Select the device top-level entry in the left-hand GUI window.
5. Implement the complete design via the standard ispLEVER GUI flow.

Hardware Evaluation

The Median Filter IP core supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core that operate in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. It may also be used to evaluate the core in hardware in user-defined designs.

Enabling Hardware Evaluation in Diamond

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

Enabling Hardware Evaluation in ispLEVER

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

Updating/Regenerating the IP Core

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

Regenerating an IP Core in Diamond

To regenerate an IP core in Diamond:

1. In IPexpress, click the **Regenerate** button.
2. In the Regenerate view of IPexpress, choose the IPX source file of the module or IP you wish to regenerate.
3. IPexpress shows the current settings for the module or IP in the Source box. Make your new settings in the T**arget** box.
4. If you want to generate a new set of files in a new location, set the new location in the **IPX Target File** box. The base of the file name will be the base of all the new file names. The IPX Target File must end with an .ipx extension.
5. Click **Regenerate.** The module's dialog box opens showing the current option settings.
6. In the dialog box, choose the desired options. To get information about the options, click **Help**. Also, check the About tab in IPexpress for links to technical notes and user guides. IP may come with additional information. As the options change, the schematic diagram of the module changes to show the I/O and the device resources the module will need.
7. To import the module into your project, if it's not already there, select **Import IPX to Diamond Project** (not available in stand-alone mode).
8. Click **Generate**.
9. Check the Generate Log tab to check for warnings and error messages.
10. Click **Close**.

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

Regenerating an IP Core in ispLEVER

To regenerate an IP core in ispLEVER:

1. In the IPexpress tool, choose **Tools > Regenerate IP/Module**.
2. In the Select a Parameter File dialog box, choose the Lattice Parameter Configuration (.lpc) file of the IP core you wish to regenerate, and click **Open**.
3. The Select Target Core Version, Design Entry, and Device dialog box shows the current settings for the IP core in the Source Value box. Make your new settings in the Target Value box.
4. If you want to generate a new set of files in a new location, set the location in the LPC Target File box. The base of the .lpc file name will be the base of all the new file names. The LPC Target File must end with an .lpc extension.
5. Click **Next**. The IP core's dialog box opens showing the current option settings.

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

Support Resources

Lattice Technical Support

There are a number of ways to receive technical support as listed below.

Online Forums

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

Telephone Support Hotline

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

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

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

- For Asia: +86 21 52989090

E-mail Support

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

Local Support

Contact your nearest Lattice sales office.

Internet

www.latticesemi.com

References

LatticeECP2/M

- HB1003, *LatticeECP2/M Family Handbook*

LatticeECP3

- HB1009, *LatticeECP3 Family Handbook*

LatticeXP2

- DS1009, *Lattice XP2 Datasheet*

Revision History

Date	Document Version	IP Core Version	Change Summary
December 2010	01.0		Initial release.

Appendix A:

Resource Utilization

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

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

LatticeXP2 FPGAs

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

Frame Size	Window Size	Edge Mode	Data Width	Input Buffer Type	Slices	LUTs	Registers	f_{MAX}
320x240	3x3	VALUE	8	EBR	546	697	568	214
256x256	5x5	VALUE	8	EBR	2198	2943	2211	214
128x128	7x7	VALUE	8	EBR	8132	11482	6934	171

1. Performance and utilization data are generated targeting an LFXP2-30E-7F484C device using Lattice Diamond 1.1 and Synplify Pro D-2010.03L-SP1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeXP2 family.

Ordering Part Number

The Ordering Part Number (OPN) for the Median Filter IP core on LatticeXP2 devices is MED-FILT-X2-U1.

LatticeECP3 FPGAs

Table A-2. Performance and Resource Utilization[1]

Frame Size	Window Size	Edge Mode	Data Width	Input Buffer Type	Slices	LUTs	Registers	f_{MAX}
320x240	3x3	VALUE	8	EBR	534	680	570	255
256x256	5x5	VALUE	8	EBR	2179	2908	2209	231
128x128	7x7	VALUE	8	EBR	8184	11536	6909	191

1. Performance and utilization data are generated targeting an LFE3-70E-8FN484CES device using Lattice Diamond 1.1 and Synplify Pro D-2010.03L-SP1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP3 family.

Ordering Part Number

The Ordering Part Number (OPN) for the Median Filter IP core on LatticeECP3 devices is MED-FILT-E3-U1.

LatticeECP2/S FPGAs

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

Frame Size	Window Size	Edge Mode	Data Width	Input Buffer Type	Slices	LUTs	Registers	f_{MAX}
320x240	3x3	VALUE	8	EBR	546	697	568	225
256x256	5x5	VALUE	8	EBR	2198	2943	2211	223
128x128	7x7	VALUE	8	EBR	8132	11482	6934	206

1. Performance and utilization data are generated targeting an LFE2 35E-7F672C device using Lattice Diamond 1.1 and Synplify Pro D-2010.03L-SP1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP2/S family.

Ordering Part Number

The Ordering Part Number (OPN) for the Median Filter IP core on LatticeECP2/S devices is MED-FILT-P2-U1.

LatticeECP2M/S FPGAs

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

Frame Size	Window Size	Edge Mode	Data Width	Input Buffer Type	Slices	LUTs	Registers	f_{MAX}
320x240	3x3	VALUE	8	EBR	546	697	568	224
256x256	5x5	VALUE	8	EBR	2198	2943	2211	254
128x128	7x7	VALUE	8	EBR	7796	11482	6934	188

1. Performance and utilization data are generated targeting an LFE2M20E-7F484C device using Lattice Diamond 1.1 and Synplify Pro D-2010.03L-SP1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP2M/S family.

Ordering Part Number

The Ordering Part Number (OPN) for the Median Filter IP core on LatticeECP2M/S devices is MED-FILT-PM-U1.