



***Lattice*CORE™**

2D Edge Detector IP Core User's Guide

Chapter 1. Introduction	3
Quick Facts	3
Features	3
Chapter 2. Functional Description	4
Key Concepts	4
Block Diagram	4
Active Region Selection	5
Interface Descriptions	6
Video Input/Output	6
Parameter Update Port	6
Chapter 3. Parameter Settings	7
User Parameters Tab	8
Video Frame	8
Edge Detection Method	9
Data Widths	9
Implementation Options	9
Performance options	9
Chapter 4. IP Core Generation	10
Licensing the IP Core	10
Getting Started	10
IPexpress-Created Files and Top Level Directory Structure	12
Instantiating the Core	13
Running Functional Simulation	13
Synthesizing and Implementing the Core in a Top-Level Design	14
Hardware Evaluation	15
Enabling Hardware Evaluation in Diamond	15
Enabling Hardware Evaluation in ispLEVER	15
Updating/Regenerating the IP Core	15
Regenerating an IP Core in Diamond	15
Regenerating an IP Core in ispLEVER	16
Chapter 5. Support Resources	17
Lattice Technical Support	17
Online Forums	17
Telephone Support Hotline	17
E-mail Support	17
Local Support	17
Internet	17
References	17
LatticeECP2/M	17
LatticeECP3	17
LatticeXP2	17
Revision History	17
Appendix A. Resource Utilization	18
LatticeECP3 Devices	18
Ordering Part Number	18
LatticeECP2M Devices	18
Ordering Part Number	18
LatticeXP2 Devices	19
Ordering Part Number	19

Introduction

The 2D Edge Detector IP core detects edges in incoming video frames using the Sobel or Prewitt algorithms. Its flexible architecture supports a wide variety of video frame sizes on LatticeECP2M™, LatticeXP2™, and LatticeECP3™ devices. A simple I/O handshake makes the core suitable for either streaming or bursty input video data. Coefficients may be set at compile time, or updated in-system via a simple memory interface. Dynamic zoom and pan functions are optionally provisioned at compile time.

Quick Facts

Table 1-1 gives quick facts about the 2D Edge Detector IP core.

Table 1-1. Quick Facts

			2D Edge Detector IP Core			
			320x240	640x480 ¹	720x480	1280x720
Core Requirements	FPGA Families Supported		LatticeXP2, LatticeECP2M, LatticeECP3			
	Minimum Device Required					
Resource Utilization	LatticeXP2	LUTs/Registers	456/393	277/119	465/399	475/399
		EBRs	1	1	1	2
		sysDSP™ blocks	0	5	0	0
	LatticeECP2M	LUTs/Registers	456/393	277/119	465/399	475/399
		EBRs	1	1	1	2
		sysDSP blocks	0	5	0	0
	LatticeECP3	LUTs/Registers	437/391	274/119	462/398	460/402
		EBRs	1	1	1	2
		sysDSP blocks	0	10	0	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			

1. Uses sysDSP blocks for adder functions.

Features

- Single color plane input
- Configurable input data width
- Dynamically variable input frame size
- Dynamic active region selection
- Dynamic selection between Sobel and Prewitt algorithms
- Dynamic detection threshold modification

Functional Description

Key Concepts

Video edge detection is the process of calculating gradients (rates of change) in pixel values in an incoming frame. If the gradient at a given pixel location exceeds a threshold value, the edge detector outputs a '1', '0' otherwise. For the Sobel and Prewitt algorithms, gradients are calculated using convolution filters with 3x3 windows with fixed coefficient values. The coefficient values are 0, 1, -1, and 2, so multiplication operations are performed using simple shifts and inversions.

The 2D Edge Detector IP core simultaneously applies two orthogonal 3x3 gradient filters to its input pixel stream and sums the results. The Sobel gradient filter kernels are separable, and have the following fixed values:

$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

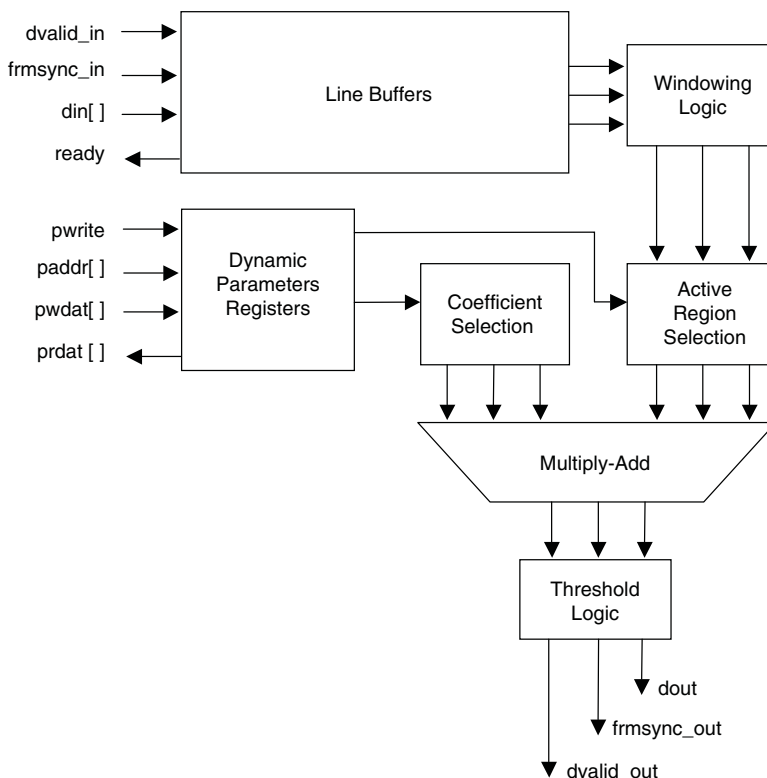
The Prewitt filter kernels are also separable, with the following fixed values:

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

Block Diagram

The high-level architecture of the 2D Edge Detector core is diagrammed in [Figure 2-1](#).

Figure 2-1. 2D Edge Detector IP Core Block Diagram



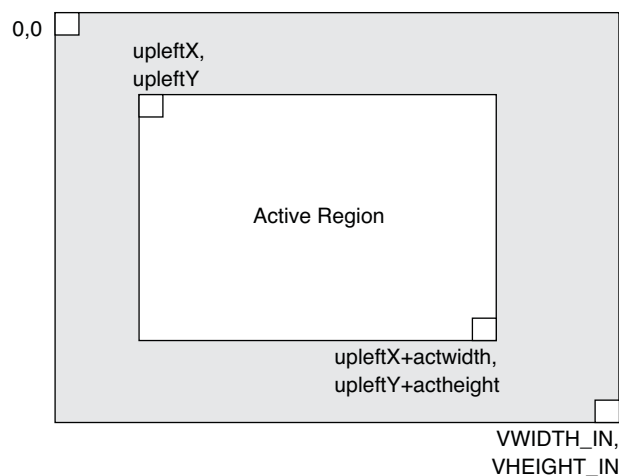
Input data is stored in line buffers, then passed to windowing logic for edge mode handling and data alignment. Optional control inputs allow real-time specification of the portion of the input frame used to generate output pixels (referred to as the “active region”).

Windowed data and coefficients are sent to the arithmetic unit which multiplies the data values by their corresponding coefficients and sums the multiplication results. Horizontal and vertical gradients are calculated, their outputs are summed, and then the sum is compared to the value on the core’s threshold input. Values exceeding the threshold result in a ‘1’ on the dout output port, which is ‘0’ otherwise.

Active Region Selection

The 2D Edge Detector 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 2-2](#).

Figure 2-2. 2D Edge Detector IP Core Active Regions



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.

Primary I/O

Table 2-1. Primary I/O

Port	Size	I/O	Description
Global Signals			
clk	1	I	System clock
rstn	1	I	Asynchronous reset, active low
clken	1	I	Clock enable, active high (optional)
srst	1	I	Synchronous reset, active high (optional)
Video Input			
ready	1	O	Core is ready for input
dvalid_in	1	I	Input valid
frmsync_in	1	I	Frame sync, current pixel is row 0, column 0
din[]	DWIDTH	I	Pixel data in
Video Output			

Table 2-1. Primary I/O (Continued)

Port	Size	I/O	Description
dvalid_out	1	O	Output valid
frmsync_out	1	O	Frame sync out, marks row 0, column 0
dout	1	O	Edge data out
Parameter Update Port			
pwrite	1	I	Parameter write
paddr	4	I	Parameter address
pwdat		I	Parameter write data
prdat		O	Parameter read data
Miscellaneous			
tags_in	TAGS_WIDTH		
tags_out	TAGS_WIDTH		

Interface Descriptions

Video Input/Output

The 2D Edge Detector 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 edge data is available on dout, and frmsync_out marks the first pixel, first row of the output video frame.

Parameter Update Port

The parameter update port is provided when the user enables dynamic update for any of the following: input frame size, active region size and location, edge detection method, or edge detection threshold. The port consists of an active-high write enable, pwrite; a four-bit register address, paddr; write data, pwdat; and read data, prdat.

When pwrite is a '1', the register selected by paddr is written with the value of pwdat. The read output prdat always carries the value of the register selected by paddr. The register addresses are as follows:

- 0 – input video frame width
- 1 – input video frame height
- 2 – x coordinate of upper left corner of active region
- 3 – y coordinate of upper left corner of active region
- 4 – width of active region (minus 1)
- 5 – height of active region (minus 1)
- 6 – edge detection algorithm (0 is Sobel, 1 is Prewitt)
- 7 – edge detection threshold
- 8 – update control (write to '1' to pass new values to the windowing function; the core will clear update_control when the windowing function has begun using the new values)

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 10](#) for a description of how to generate the IP.

The 2D Edge Detector 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](#) provides the list of user configurable parameters for the 2D Edge Detector IP core.

Table 3-1. Edge Detector Parameters

Parameter	Range	Default
Video Frame In and Out		
Input Data Width	4 to 24	8
Frame Width In	200 to 2000	320
Frame Hight In	100 to 1200	240
Active Region Width	1 to 703	320
Active Region Height	24 to 1200	240
Upper Left X Coordinate of Active Region	0 to ACTWIDTH-1	0
Upper Left Y Coordinate of Active Region	0 to ACTHEIGHT-1	0
Edge Mode	VALUE, COPY, MIRROR	COPY
Edge Value	0 to 1<<DWIDTH-1	0
Edge Detection Method	SOBEL, PREWITT	SOBEL
Threshold	0 to 1<<DWIDTH-1	0
Dynamic Frame Size Update	0 or 1	0
Dynamic Method Selection Update	0 or 1	0
Tags Width	0 to 32	0
Adder Hardware Implementation Selection	Logic, DSP	Logic
Line Buffer Type	EBR, PLC	EBR

User Parameters Tab

The User Parameters tab is shown in [Figure 3-1](#).

Figure 3-1. User Parameters Tab

User Parameters

Video Frame

Frame width in (100-2000) Frame height in (100-1200) ☐ Dynamic frame size

☐ Dynamic active region selection ☒ Full screen

Upper left coordinate of active region: X (0-703) Y (0-479)

Active region width (1-703) Active region height (1-479)

Edge Mode: ☐ Copy ☐ Mirror ☒ Value (0-255)

Edge Detection Method

☐ Dynamic method selection

Method ☒ Sobel ☐ Prewitt

Threshold:

Data Widths

Input data width ▼

Tags width ▼

Implementation Options

Line buffer type ▼

Adders: ☒ Logic ☐ DSP Block

☐ Synchronous reset (rst)

☐ Clock enable (clken)

Performance options

Frequency constraint (MHz): (1-400)

Video Frame

Frame Width In

This parameter defines the input video frame width.

Frame Height In

This parameter defines the input video frame height.

Dynamic Frame Size

This parameter defines enables dynamic frame size.

Dynamic Active Region

This checkbox enables dynamic input frame size setting via the parameter update port.

Full Screen

This checkbox automatically sets the active region to the full frame size. Unchecking it allows a user-defined active region.

X Coordinate of Upper Left of Active Region

This specifies the X coordinate of the upper left corner of the active region.

Y Coordinate of Upper Left of Active Region

This specifies the Y coordinate of the upper left corner of the active region.

Active Region Width

This parameter sets the active region width.

Active Region Height

This parameter sets the active region height.

Edge Mode

This section selects between Copy (use the value of the pixels at the edge); Mirror (use the values of pixels the same distance from the edge); and Value (use a fixed value for pixels straddling the boundary).

Edge Detection Method**Dynamic Method Selection**

This checkbox enables dynamic selection of the edge detection algorithm via the parameter update port.

Method

These radio buttons make the algorithm selection at synthesis time.

Threshold

This parameter sets the edge detection threshold level.

Data Widths**Input Data Width**

This parameter chooses the bit width of the pixel data input din.

Tags Width

This parameter sets the bit width of the optional tags_in and tags_out ports.

Implementation Options**Line Buffer Type**

This parameter selects EBR or distributed RAM for the line buffers.

Adders

These radio buttons choose the implementation method for the adder blocks.

Synchronous Reset (srst)

This checkbox enables the synchronous reset feature and adds the srst input port.

Clock Enable (clken)

This checkbox enables the clock enable feature and adds the clken input port.

Performance options**Frequency Constraint (MHz)**

This parameter sets the target clock frequency (in MHz).

IP Core Generation

This chapter provides information on how to generate the 2D Edge Detector 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 2D Edge Detector 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>

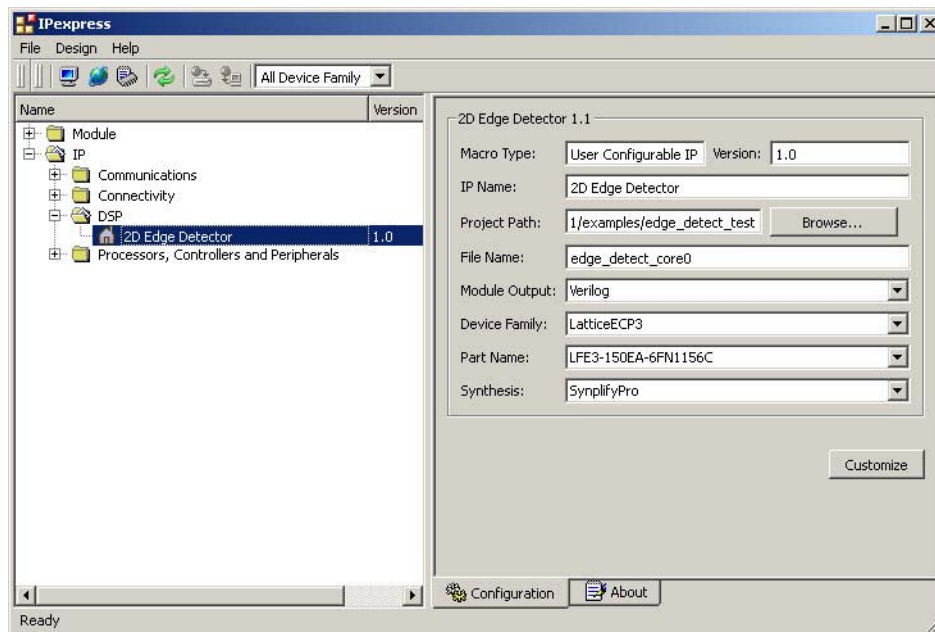
Users may download and generate the 2D Edge Detector IP core and fully evaluate the core through functional simulation and implementation (synthesis, map, place and route) without an IP license. The 2D Edge Detector IP core also supports Lattice's IP hardware evaluation capability, which makes it possible 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 15](#) 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 2D Edge Detector 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](#).

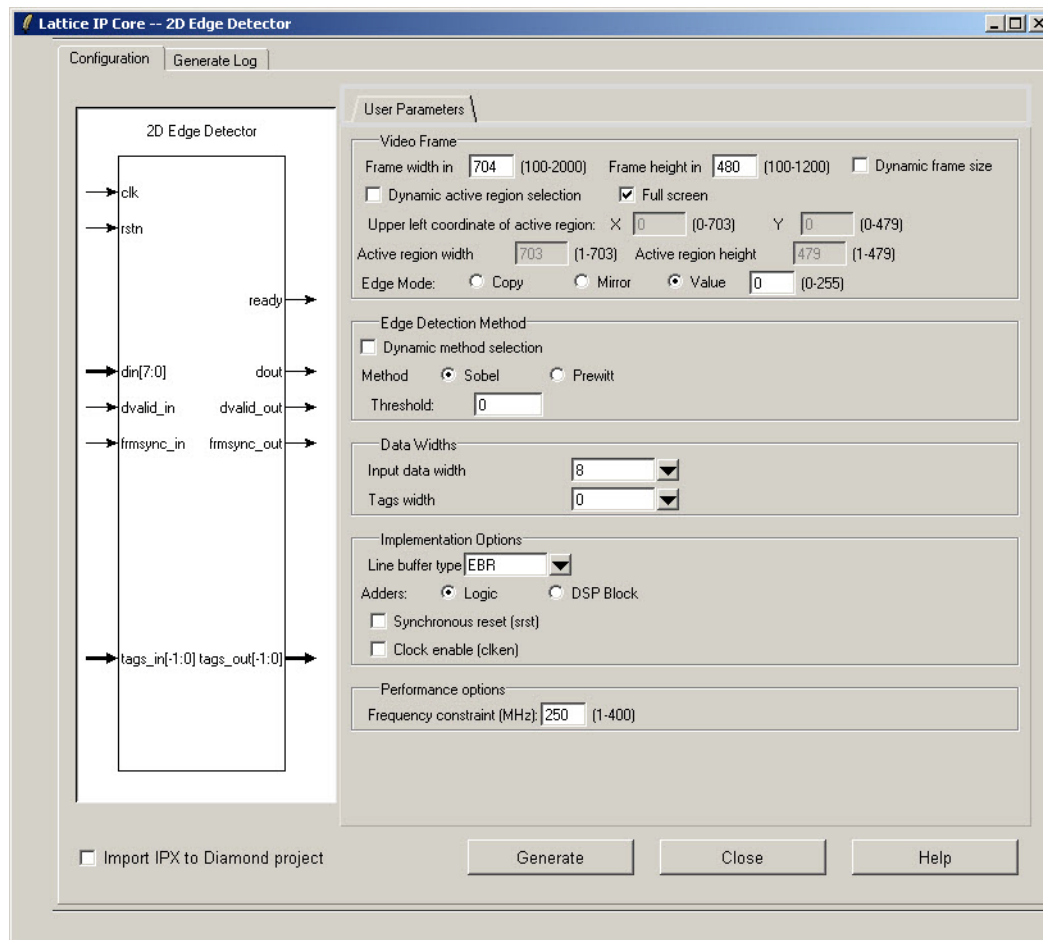
The IPexpress tool GUI dialog box for the 2D Edge Detector IP core is shown in [Figure 4-1](#). To generate a specific IP core configuration the user specifies:

- **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 2D Edge Detector 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 7](#) for more information on the 2D Edge Detector 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 2D Edge Detector IP for LatticeECP3 device.

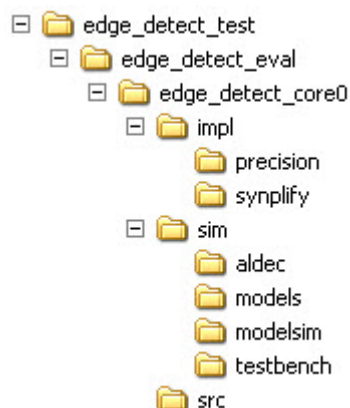
Figure 4-3. 2D Edge Detector IP Core Directory Structure

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 2D Edge Detector IP core.
<username>_beh.v/.vhd	This file provides the front-end simulation library for the 2D Edge Detector 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 2D Edge Detector 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>\edge_detect_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 2D Edge Detector 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 2D Edge Detector 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>\edge_detect_eval\<username>\sim\modelsim\scripts`. The simulation script supporting Aldec evaluation simulation is provided in `\<project_dir>\edge_detect_eval\<username>\sim\aldec\scripts`. Both Modelsim and Aldec simulation is supported via test bench files provided in `\<project_dir>\edge_detect_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>\edge_detect_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>\edge_detect_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. Choosing **Yes** closes ModelSim.

Synthesizing and Implementing the Core in a Top-Level Design

Synthesis support for the 2D Edge Detector IP core is provided for Mentor Graphics Precision or Synopsys Synplify. The 2D Edge Detector 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>\edge_detect_eval\<username>\src\top` supports the ability to implement the 2D Edge Detector 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>\edge_detect_eval\<username>\impl\synplify` and the `\<project_dir>\edge_detect_eval\<username>\impl\precision` directories, respectively.

To use this project file in Diamond:

1. Choose **File > Open > Project**.
2. Browse to `\<project_dir>\edge_detect_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>\edge_detect_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 2D Edge Detector 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 **Target** box.
4. If you want to generate a new set of files in a new location, set the new location in the **IPX Target File** box. The base of the file name will be the base of all the new file names. The IPX Target File must end with an .ipx extension.
5. Click **Regenerate**. The module's dialog box opens showing the current option settings.
6. In the 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
February 2011	01.0	1.0	Initial release.

Resource Utilization

This appendix gives resource utilization information for Lattice FPGAs using the 2D Edge Detector IP core.

The IP configurations shown in this chapter were generated using the IPexpress software tool. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the Diamond and ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and Diamond and ispLEVER help systems. For more information on the Diamond or ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com.

LatticeECP3 Devices

Table A-1. Performance and Resource Utilization^{1, 2}

Frame Size	DWIDTH	DSP Adders	LUTs	Slices	PFU Registers	EBRs	DSP Slices	f _{MAX} (MHz)
320x240	8	No	540	368	397	1	0	226
640x480	8	Yes	399	295	203	1	60	168
720x480	8	No	528	351	403	1	0	204
1280x720	8	No	528	352	407	2	0	154

1. Performance and utilization data are generated targeting an LFE3-70EFPBGA672 device using Diamond 1.1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP3 family.

2. PAR settings:

Placement Effort Level: 5

Routing Passes: 6

Placement Iterations: 3

Routing Delay Reduction Passes: 1

PLC Input Limit: Low

Ordering Part Number

The Ordering Part Number (OPN) for the 2D Edge Detector IP core on LatticeECP3 devices is EDGE-DET-E3-U1.

LatticeECP2M Devices

Table A-2. Performance and Resource Utilization^{1, 2}

Frame Size	DWIDTH	DSP Adders	LUTs	Slices	PFU Registers	EBRs	DSP Slices	f _{MAX} (MHz)
320x240	8	No	559	386	398	1	0	190
640x480	8	Yes	408	296	203	1	40	173
720x480	8	No	530	353	402	1	0	209
1280x720	8	No	553	368	410	2	0	191

1. Performance and utilization data are generated targeting an LFE2M35E-6F672C device using Diamond 1.1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeECP2M family.

2. PAR settings: defaults.

Ordering Part Number

The Ordering Part Number (OPN) for the 2D Edge Detector IP core on LatticeECP2M devices is EDGE-DET-PM-U1.

LatticeXP2 Devices

Table A-3. Performance and Resource Utilization^{1, 2}

Frame Size	DWIDTH	DSP Adders	LUTs	Slices	PFU Registers	EBRs	DSP Slices	f _{MAX} (MHz)
320x240	8	No	559	386	398	1	0	155
640x480	8	Yes	408	296	203	1	40	153
720x480	8	No	530	353	402	1	0	161
1280x720	8	No	553	368	410	2	0	170

1. Performance and utilization data are generated targeting an LFXP2-40E-6F672C device using Diamond 1.1 software. Performance may vary when using a different software version or targeting a different device density or speed grade within the LatticeXP2 family.

2. PAR settings: defaults.

Ordering Part Number

The Ordering Part Number (OPN) for the 2D Edge Detector IP core on LatticeXP2 devices is EDGE-DET-X2-U1.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Lattice:

[EDGE-DET-E3-U1](#) [EDGE-DET-X2-UT1](#) [EDGE-DET-X2-U1](#) [EDGE-DET-P2-UT1](#) [EDGE-DET-E3-UT1](#) [EDGE-DET-PM-U1](#) [EDGE-DET-PM-UT1](#) [EDGE-DET-P2-U1](#)