



XAPP933 (v1.1) October 23, 2007

Two-Dimensional Linear Filtering

Author: Robert Turney

Summary

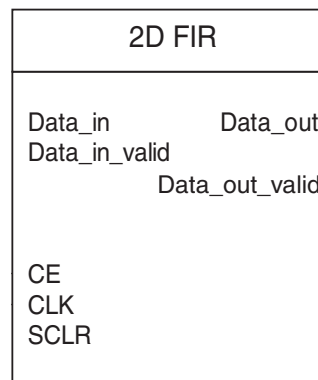
This application note provides a Xilinx FPGA solution to two-dimensional filtering with a parameterized VHDL reference design. Two-dimensional linear filtering (2D FIR) has many applications in imaging and video processing. The range of applications vary from very precise medical imaging systems to low precision industrial imaging and consumer video applications.

For optimized FIR operations, refer to the Xilinx FIR Compiler v1.0 in Coregen or System Generator.

Introduction

This reference design has a fully synchronous interface through the CE, CLK, and SCLR ports.

The Data_In_valid input signal indicates valid pixels on the Data_in bus. The Data_out_valid output signal indicates valid output data on the Data_out bus. The entire module can be stalled with the CE signal at any time, and synchronous clear can be used to reset. [Figure 1](#) is a pinout diagram.



x933_01_033006

Figure 1: Pinout Diagram

Parameterization

The design input parameters (Generics) are listed in [Table 1](#).

Table 1: Design Parameters

Design Parameter	Type	Range	Description
width	Integer	4-16	Input data width
iwidth	Integer	4-32	Intermediate width between vertical and horizontal filters
cwidth	Integer	4-32	Coefficient width for filter coefficients and determines the amount of filtering in the stop band
hsize	Integer	32-4096	Horizontal size of the image
vsize	Integer	32-4096	Vertical size of the image

Table 1: Design Parameters (Continued)

Design Parameter	Type	Range	Description
owidth	Integer	4-32	Sets the number of bits of filtered data that should be output
hcoefs	Integer_array	4-64	The sum of the horizontal coefficients should be equivalent to normalized one in the user's system. For example, if there are 10 bit coefficients which are signed terms, then 512 is normalized one.
vcoefs	Integer_array	4-64	The sum of the vertical coefficients should be equivalent to normalized one in the user's system. For example, if there are 10 bit coefficients which are signed terms, then 512 is normalized one.
Hnum_taps	Integer	4-64	Hnum_taps are integers that indicate how many horizontal coefficient taps exist.
Vnum_taps	Integer	4-64	Vnum_taps are integers to indicate how many vertical coefficient taps exist.
h_symmetry	Integer	0-1	The h_symmetry and values are set to 1 or 0 to indicate if the coefficients are symmetrical and the symmetrical architecture can be used. In this case, only the half of the hcoefs are used in the design and the second half are ignored.
v_symmetry	Integer	0-1	The v_symmetry values are set to 1 or 0 to indicate if the coefficients are symmetrical and the symmetrical architecture can be used. In this case, only the half of the vcoefs are used in the design and the second half are ignored.

Detailed Operation

2D FIR Filtering

Filtering of imaging and video can be performed linearly or using non-linear techniques [Ref 1], [Ref 2], and [Ref 3]. For linear filtering FIR filters are generally used due to their phase response and ease of implementation. For a general input image f_{ij} and expected output image g_{ij} the filtering operation is given by:

$$g_{ij} = \sum_{k=-N}^N \sum_{l=-M}^M H(i-k, j-l) f_{kl} \quad \text{Equation 1}$$

Where H is the filter kernel.

If H can be expressed as an outer product of two vectors, then the filter is said to be separable. This significantly reduces the number of multiplications from N squared to $2N$. If a 2D filter is not separable, it can be expressed as a sum of separable kernels through the use of Singular Value Decomposition (SVD) [Ref 4]. In addition, many filters used for processing image have symmetry in their filter coefficients. This further reduces the number of multiplications from $2N$ to $(N+1)$ for odd filters and N for even filters.

The 2D FIR Separable Architecture

The architecture for a 2D FIR filter mapped into an Xilinx FPGA is shown in Figure 2. In this architecture we see the use of Line Buffers to store up enough lines to hold the vertical size of the kernel. The example shown in Figure 2 is for five vertical filter coefficients. After this first stage, the lines are fed into the Vertical filter followed by the Horizontal filter.

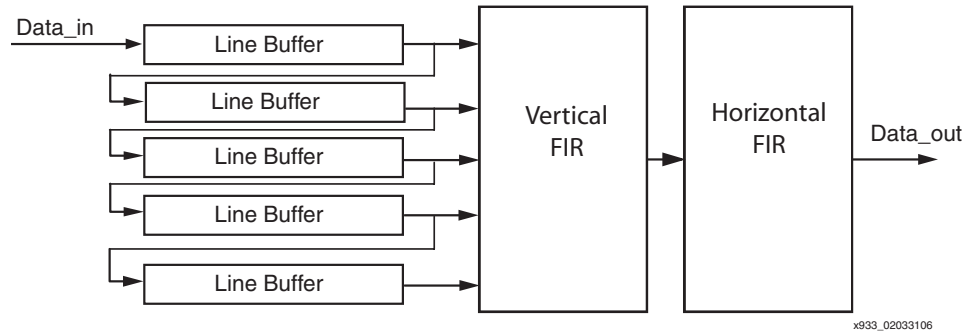


Figure 2: 2D FIR Separable

The vertical filter is implemented by taking the line buffer data in and performing the multiplication with the filter coefficient and using an adder tree to provide the intermediate result. This intermediate result is then filtered horizontally after a series of register delays to provide the appropriate spatial relationship. The entire design has parameters to control the number of bits throughout the processing.

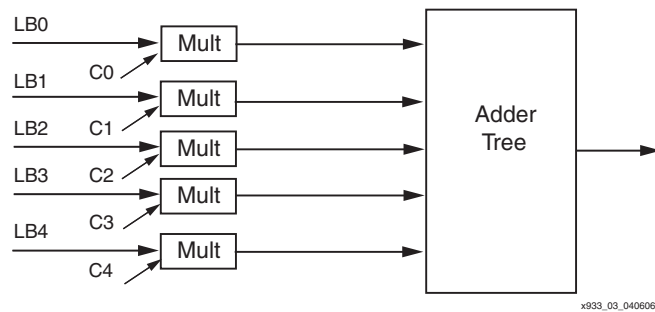


Figure 3: Vertical FIR

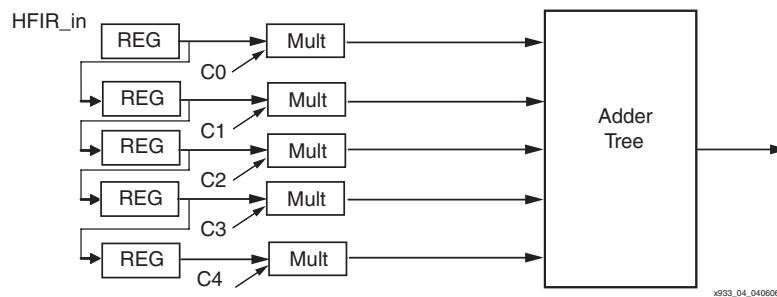


Figure 4: Horizontal FIR

The 2D FIR Symmetrical Separable Architecture

The extension to symmetrical implementation from this architecture is achieved through a Pre-adder stage in the Vertical and Horizontal FIRs. This is illustrated in Figure 5, Figure 6, and Figure 7.

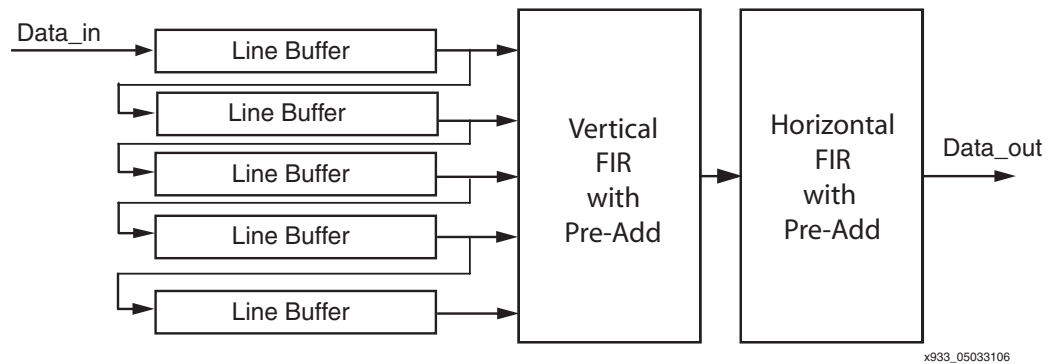


Figure 5: 2D FIR with Symmetry

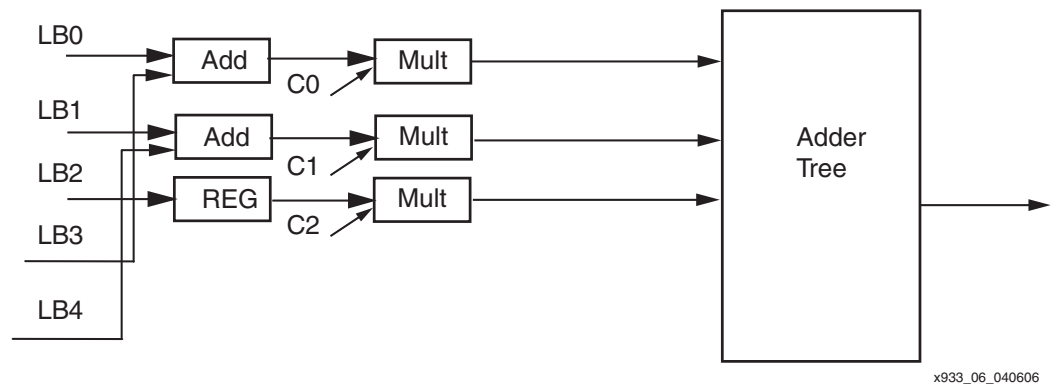


Figure 6: Vertical FIR with Symmetry

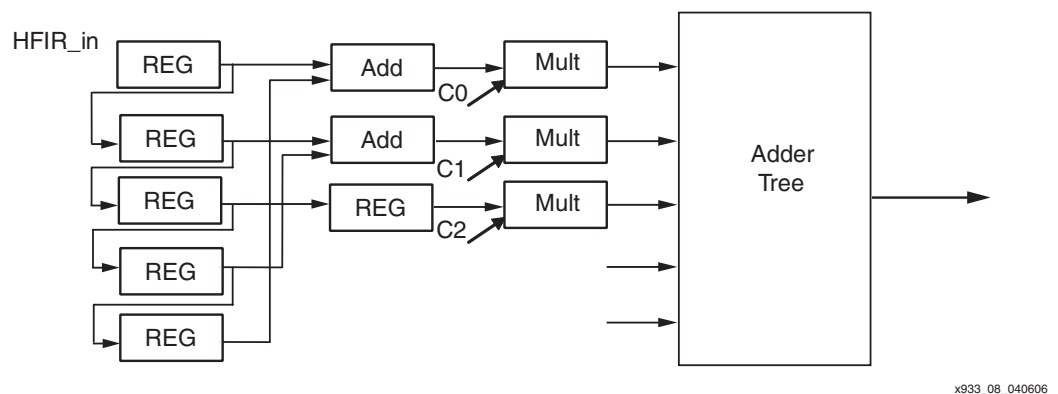


Figure 7: Horizontal FIR with Symmetry

Using the 2D FIR Reference Design

The 2D FIR reference design is intended to be used with a raster scan video system. As such the Data_in_valid should be set high and an entire line of raster data should be presented on the Data_in bus with each pixel coming on one clock. At the end of lines the Data_in_valid can be brought low or lines can be concatenated together if image edges precision is not critical to the design.

Data_out_valid will be driven after a number of lines have been put into the 2D FIR signaling that valid output data is on the Data_out bus. At the end of the image, the user needs to flush out the remaining lines of filtered image by running a flush operation and driving Data_in_valid a number of dummy lines equal to half of the vertical coefficient size. Padding at the beginning of lines, end of lines, beginning of frames and end of frames can be performed by the user by wrapping this 2D FIR with a padding function to serve the needs of the user application.

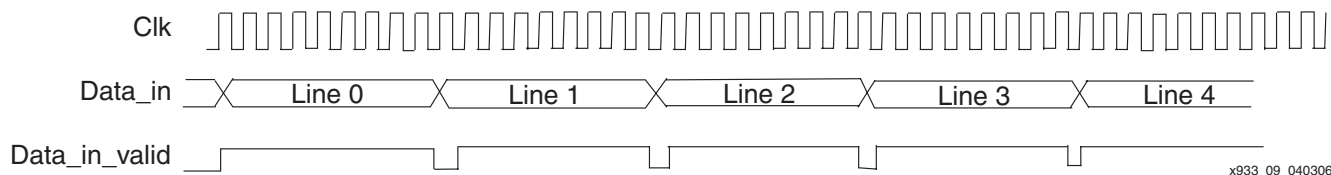


Figure 8: Input Timing Diagram

Verification in System Generator and Hardware Loop Tests

Verification of the 2D FIR VHDL was achieved by utilizing the VHDL co-simulation feature of System Generator. The parameters were varied to test corner cases against the MATLAB function `filter2.m`. The Gold image was also used during testing. Hardware in the loop tests were also run with the WildCard 2 PCMCIA card and WildCard 4 PCMCIA cards in a laptop environment with System Generator.

Note: To run hardware in the loop tests, the .x86 model will have to be regenerated from the System Generator token.

There are over 100 test cases that are run in `regression_testN.m` to test various parameter settings of the design.

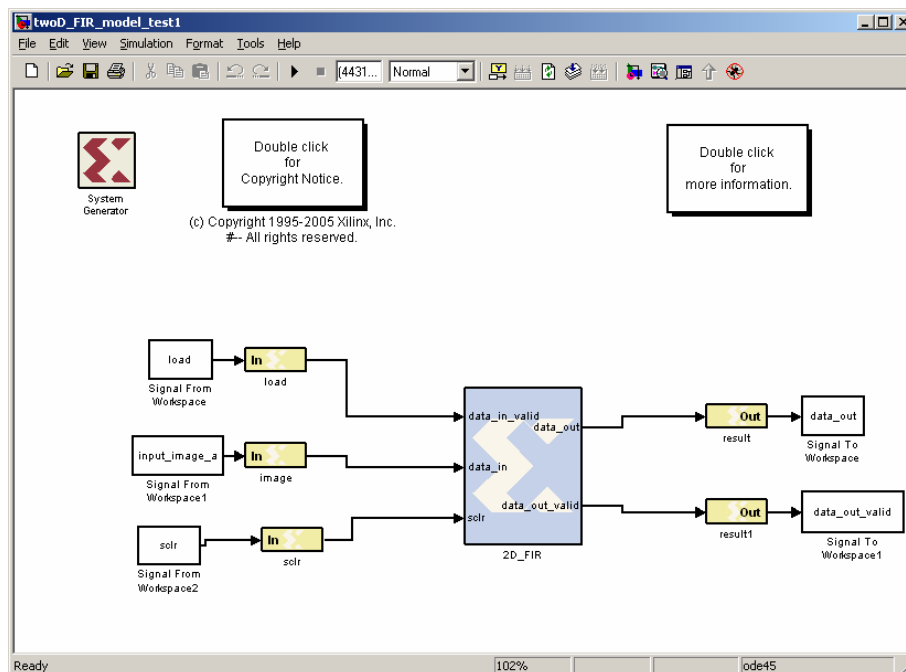


Figure 9: System Generator Design of 2D FIR

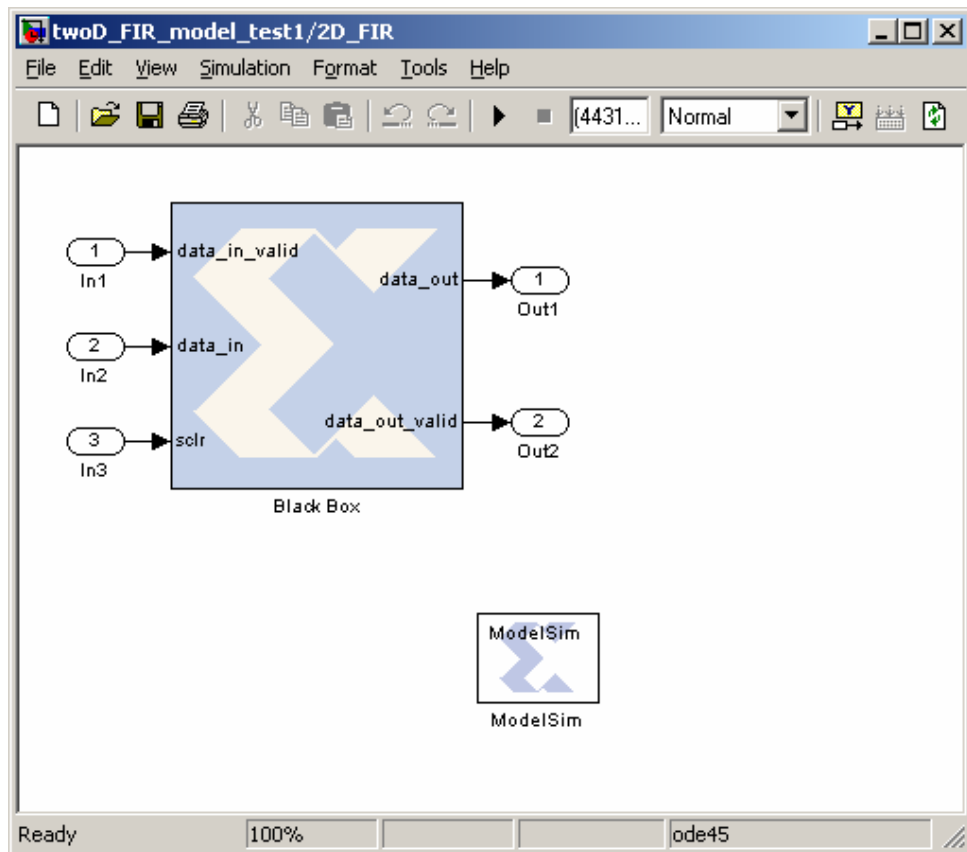


Figure 10: System Generator Design of 2D FIR

2D FIR Characterization

The 2D FIR reference design was characterized with the three test cases listed in Table 2. In these test cases, a set of parameters were desired to show what performance can be expected for representative bit sizes and image sizes.

Table 2: Test Case Characterization

Test	Width	Cwidth	lwidth	Hnum	Vnum	Hsize	Vsize
1	8	10	10	5	5	720	576
2	8	12	12	15	15	512	512
3	10	14	14	31	31	1528	1146

Table 3: Non-Symmetrical 2D FIR Characterization Results

Test	Synthesis	FPGA Family	Slices	Block RAMS	Mults/DSP48s	MHz
1	XST	Virtex™-II Pro	302	5	10	227
1	XST	Spartan™-3	306	5	10	156
1	SYN	Virtex-II Pro	369	5	10	192
1	SYN	Spartan-3	371	5	10	107
2	XST	Virtex-II Pro	849	15	30	203
2	XST	Spartan-3	838	15	30	146
2	SYN	Virtex-II Pro	836	15	30	175

Table 3: Non-Symmetrical 2D FIR Characterization Results (Continued)

Test	Synthesis	FPGA Family	Slices	Block RAMS	Mults/DSP48s	MHz
2	SYN	Spartan-3	1013	15	30	90
3	XST	Virtex-II Pro	2010	62	62	180
3	XST	Spartan-3	1982	62	62	118
3	SYN	Virtex-II Pro	2637	62	62	133
3	SYN	Spartan-3	2630	62	62	69

Table 4: Symmetrical 2D FIR Characterization Results

Test	Synthesis	FPGA Family	Slices	Block RAMS	Mults/DSP48s	MHz
1	XST	Virtex-II Pro	257	5	6	225
1	XST	Spartan-3	401	5	6	137
1	SYN	Virtex-II Pro	301	5	6	214
1	SYN	Spartan-3	302	5	6	104
2	XST	Virtex-II Pro	727	15	16	201
2	XST	Spartan-3	716	15	16	131
2	SYN	Virtex-II Pro	849	15	16	165
2	SYN	Spartan-3	848	15	16	101
3	XST	Virtex-II Pro	1719	62	32	174
3	XST	Spartan-3	1691	62	32	113
3	SYN	Virtex-II Pro	2064	62	32	144
3	SYN	Spartan-3	2071	62	32	65

Reference Design Files

Reference design files are available for download from the Xilinx website at:
<http://www.xilinx.com/bvdocs/apnotes/xapp933.zip>

References

1. Ronald Bracewell, Two Dimensional Imaging, 1995 Prentice-Hall Inc.
2. William Pratt, Digital Image Processing, 2nd Ed. 1991 John Wiley & Sons, Inc.
3. Rafael Gonzalez, Richards Woods, Digital Image Processing, 1992 Addison-Wesley Inc.
4. Klema, V. C. and A. J. Laub, "The Singular Value Decomposition: Its Computation and Some Applications," *IEEE Trans. Autom. Control*, Vol. AC-25, pp. 164-176, April 1980
5. Annapolis Micro Systems Inc., *WildCard-II™ and WildCard-4™ Reference Manual*, 2002-2006

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/09/06	1.0	Initial Xilinx release.
10/23/07	1.1	<ul style="list-style-type: none">Added a note to “Verification in System Generator and Hardware Loop Tests.”Updated Copyright Notice and Notice of Disclaimer.

Notice of Disclaimer

Xilinx is disclosing this Application Note to you “AS-IS” with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.