FIR Filters

January 1996, ver. 1 | Functional Specification 1

Features

- High-speed operation: up to 105 million samples per second (MSPS)
- 8-, 16-, 24-, 32-, and 64-tap finite impulse response (FIR) filters
- Parameterized data widths, pipelining, and symmetry
- Optimized for FLEX 10K and FLEX 8000 devices
- Coefficients implemented as look-up table (LUT) vectors
- Parallel and serial design versions
- Supported by schematic and text design entry methods, including VHDL, Verilog HDL, and the Altera Hardware Description Language (AHDL)
- Useful for a variety of applications, including high performance, real-time video and image filtering; radio frequency (RF) filtering; radar; magnetic resonance imaging (MRI) applications; multi-rate digital signal processing (DSP); and spread spectrum filtering

General Description

Table 1 shows the performance and size of specific FIR filters in FLEX 8000A devices. The input, coefficient, internal, and output precision values are fully parameterized. The filter name specifies the number of taps in the filter, and the architecture and suffixes follow the naming convention: P = parallel, S = serial. The video convolution filter, `fir_3x3`, is discussed on page 16 of this document.

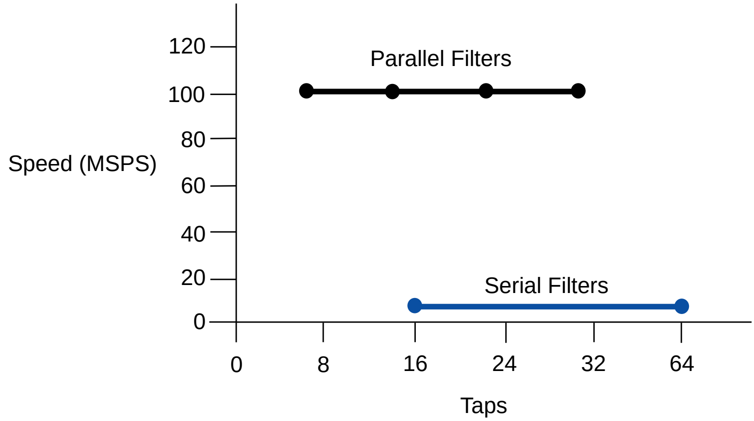
Table 1. FIR Filter Performance & Size in FLEX 8000A Devices

FIR Filter	Input Precision	Taps	Precision/Bits			Size (Logic Elements)	Performance (MSPS)			
			Coef.	Internal	Output		Pipelined		Non-Pipelined	
							A-2	A-4	A-2	A-4
`fir_08tp`	8	8	8	17	17	296	101	66	28	15
`fir_16tp`	8	16	8	10	10	468	101	75	20	13
`fir_24tp`	8	24	8	10	10	653	100	74	18	12
`fir_32tp`	8	32	8	10	10	862	101	75	18	12
`fir_16ts`	8	16	8	18	18	272	7.0	4.9	3.4	2.4
`fir_64ts`	8	64	8	24	24	920	6.5	4.1	2.4	1.7
`fir_3x3`	8	9	8	18	18	327	102	67	24	14

A-FS-01-01

Adding more taps to the filter does not significantly change the speed for any given precision width. See Figure 1.

Figure 1. Speed for 8-Bit Input Filters in FLEX 8000A Devices



Altera's FIR filters provide the designer with a great deal of flexibility because they can be customized for specialized applications by modifying the source code. In addition, the filters' flexibility allows designers to implement decimating or interpolating filters, and to use the vector multiplier from these reference designs to implement any operation that has multiply and accumulate (MACs) functions, where one set of multiplicands is constant.

All linear-phase FIR filters have symmetric coefficients; therefore, the designer can take advantage of this symmetry by adding taps prior to multiplication. For example, an 8-tap FIR filter can be implemented with a 4-input vector multiplier. The vector multiplier performs the following multiplication: $y = (h(1) \times s(1)) + (h(2) \times s(2)) + (h(3) \times s(3)) + (h(4) \times s(4))$, where $h(n)$ are FIR coefficients, and $s(n)$ are pre-added data samples. The coefficients must be fixed prior to compilation, although you can change the coefficients by reconfiguring the device. Thus, it is possible to change the filter function in response to user controls. The device can be reconfigured in less than 100 ms. To be implemented in FLEX devices, all FIR filters require MAX+PLUS II version 6.0 or higher.

See *Application Note 73* (*Implementing FIR Filters in FLEX Devices*) for a complete description of vector multipliers and the FIR filter architecture.

Parameters

The following FIR filter parameters can be defined by the user:

- *Data Width* (`width`)—The width (in bits) of incoming data.
- *Coefficient Precision* (`rom_p`)—The width of the coefficients used for the FIR filter, independent of the data width setting.
- *Internal Precision* (`internal_p`)—The precision (in bits) of internal calculations in the FIR filter.
- *Output Precision* (`output_p`)—The precision (in bits) of the output data. This value can be different from the internal precision.
- *Pipelined/Non-Pipelined Design* (`pipelined`)—When designers choose between a fully pipelined or non-pipelined design, the main filter design program, FIRGEN, creates the **coef.inc** file accordingly. A fully pipelined design results in the maximum operating frequency with no extra device area usage.
- *Symmetrical/Antisymmetrical Filters* ("`sym`" or "`anti`")—Designers can specify whether the filter is symmetric or antisymmetric. Symmetrical FIR filters implement even functions with symmetric tap adders. Designers can create an antisymmetric FIR filter that implements an odd function by using symmetric tap subtractors. FIRGEN allows the designer to specify symmetrical or antisymmetrical coefficients.
- *Coefficients*—FIRGEN automatically generates an AHDL Include File (**coef.inc**) that specifies vector table representations of filter coefficients. Designers specify the original file containing coefficients to FIRGEN, which then computes the vector tables and adds them to **coef.inc**. Although the coefficients are specified at compilation, they are not parameters because they are defined by constants in the **coef.inc** file. If more than one filter exists in the design, modify a copy of the top-level Text Design File (**.tdf**) file (e.g., **fir_08tp.tdf**) and change the name of the Include File (**coef.inc**).

Parameters vs. Ports

Parameters, which are variables established at design time, determine the configuration of the filter. Parameters cannot be changed once the filter has been designed and is operating in-circuit. In contrast, ports refer to the in-circuit I/O ports, including the I/O data, Clock, and Reset.

Design Flow Overview

The 8-tap parallel FIR filter design flow in Figure 2 is common to all FIR filter reference designs supplied with the Altera DSP Design Kit. The designer can specify the filter requirements using any commercially available FIR filter design software that can store the coefficients in an ASCII file. This file should contain the filter coefficients one-per-line and should have the extension **.dat**.
The listing below shows the coefficients for an 8-tap lowpass filter in the **lp8.dat** file, available from the Altera DSP Design Kit:

```
-5.144663E-03
-2.279553E-02
 9.638777E-02
 .4296515
 .4296515
 9.638777E-02
-2.279553E-02
-5.144663E-03
```

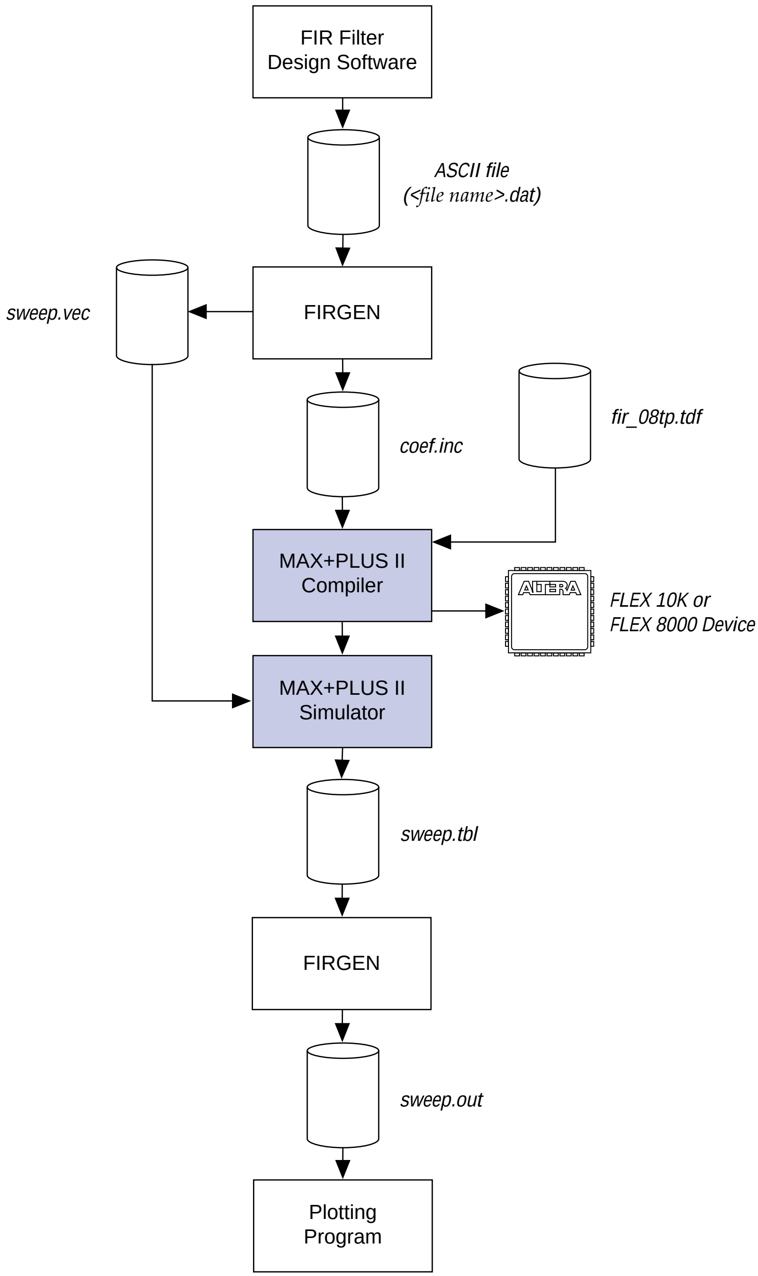
Next, the designer runs FIRGEN to read the coefficients from the data file. Then, FIRGEN calculates the vector table for the LUT and creates **coef.inc**.

Using MAX+PLUS II, the designer can compile the AHDL Include File and set the parameters for the particular filter. To insure the smallest size and maximum performance, set the *Global Project Synthesis Style* to *Fast* in the **Global Project Logic Synthesis** dialog box. This option uses carry chains for high-speed operation.

FIRGEN also generates a swept sine wave (or chirp signal) Vector File, called **sweep.vec**. This file can be used as a simulation source file for the MAX+PLUS II Simulator to generate the filter's frequency response. The response is then viewed with any plotting program, such as Microsoft Excel or Gnuplot, a freely distributed program included with the Altera DSP Design Kit.

☞ For a sample design, see the *FIR Filter Design Walkthrough* in the **readme** file on the Altera DSP Design Kit CD-ROM.

Figure 2. FIR Filter Design Flow



Instantiation

The designer can compile a FIR filter as a stand-alone design or can insert an instance of a FIR filter in a design. The filter function is placed and connected in the MAX+PLUS II Graphic Editor, an AHDL TDF, a third-party Verilog HDL file, or a VHDL file. Figure 3 shows an instance of an 8-tap FIR filter in a Graphic Design File (**.gdf**).

Figure 3. 8-Tap Filter in a GDF

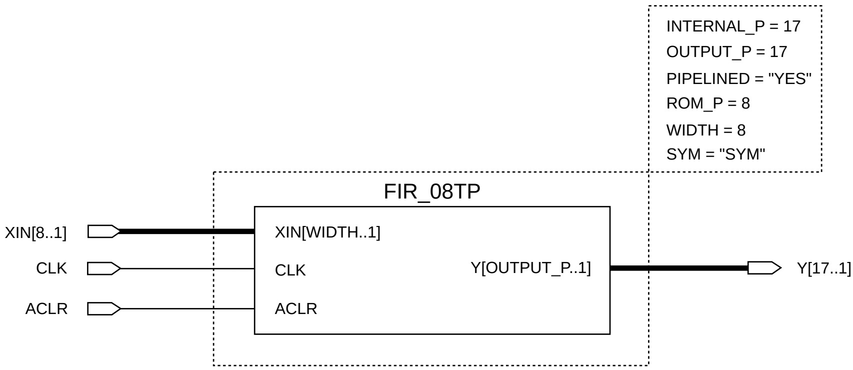


Figure 4 shows an instance of an 8-tap FIR filter implemented in an AHDL TDF.

Figure 4. 8-Tap Filter in an AHDL TDF

```
INCLUDE "fir_08tp"
CONSTANT INPUT_WIDTH = 8;
CONSTANT COEFFICIENT_P = 8;
CONSTANT OUTPUT_P = 17;
CONSTANT INTERNAL_P = 15;
CONSTANT SYM = "sym"

SUBDESIGN test_tdf
(
   xin[width..1]       :INPUT;
   clk                 :INPUT;
   aclr                :INPUT;
   y[output_p..1]      :OUTPUT
)

VARIABLE
    fir : fir_08tp WITH
    (
       width = input_width,
       rom_p = coefficient_p,
       output_p = output_p,
       pipelined = "YES"
       internal_p = internal_p
       sym = "sym"
    );

 BEGIN
    fir.xin[] = xin[];
    fir.aclr = aclr;
    fir.clk = clk;
    y[] = fir.y[];

 END;
```

Features

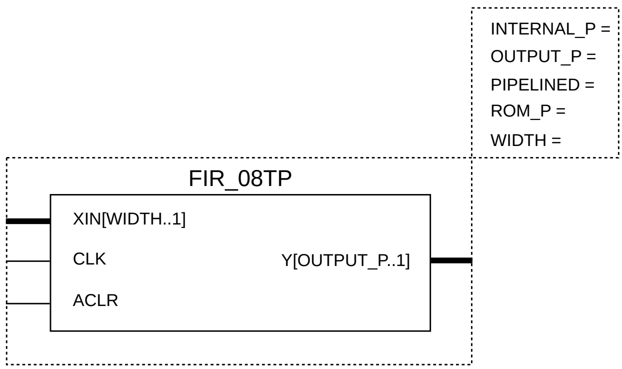
- Parallel FIR filter reference designs
- High-speed operation: > 100 MSPS
- Parameterized data widths, pipelining, and symmetry
- Symmetrical or antisymmetrical coefficients
- Parallel design versions for maximum performance

General Description

Altera parallel FIR filters are linear-phase designs with operation speeds of over 100 MSPS. By efficiently using the Altera FLEX architecture, these filters provide extremely fast, flexible, and cost-effective design solutions. The coefficients are stored in LUTs and their values are determined at compilation; however, the coefficients can be changed by reconfiguring the device.

Figure 5 shows the symbol for the 8-tap parallel FIR filter. All other parallel FIR filters in the Altera DSP Design Kit have the same symbol. Coefficients are defined in an Include File (**.inc**) called **coef.inc**.

Figure 5. 8-Tap Parallel FIR Filter Symbol



Function Prototype

The AHDL Function Prototype for the `fir_08tp` function is shown below and is the same for all other parallel FIR filters in the Altera DSP Design Kit:

```
FUNCTION fir_08tp (xin[width..1] clk, aclr)
   WITH (width, rom_p, internal_p, output_p, pipelined)
   RETURNS (y[output_p..1]);
```

Parameters

Parameters for the parallel FIR filter functions are provided in Table 2. In the default value of the `output_p` parameter, the constant `BLOCKS` refers to the number of 8-tap blocks in the filter.

Table 2. Parallel FIR Filter Parameters

Name	Default	Value	Description
`width`	`8`	Integer	Input word width (in bits)
`rom_p`	`8`	Integer	Coefficient precision
`pipelined`	`"yes"`	`"yes"` or `"no"`	Pipelined/non-pipelined
`sym`	`"sym"`	`"sym"` or `"anti"`	Symmetrical/antisymmetrical
`output_p`	`width+blocks+rom_p`	Integer	Output precision (in bits)
`internal_p`	`output_p`	Integer	Internal precision (in bits)

Ports

Input and output ports for the parallel FIR filter functions are provided in Table 3l.

Table 3. Input & Output Ports

Port Type	Name	Description
Input	`xin[width..1]`	Data input
Input	`clk`	Clock input
Input	`aclr`	Asynchronous Clear
Output	`y[output_p..1]`	Data output

Block Diagrams

Figures 6 through 9 show block diagrams for the `fir_08tp`, `fir_16tp`, `fir_24tp`, and `fir_32tp` filters.

See *Application Note 73* (*Implementing FIR Filters in FLEX Devices*) for more information on how to implement vector multipliers.

Figure 6. FIR_08TP Filter Block Diagram

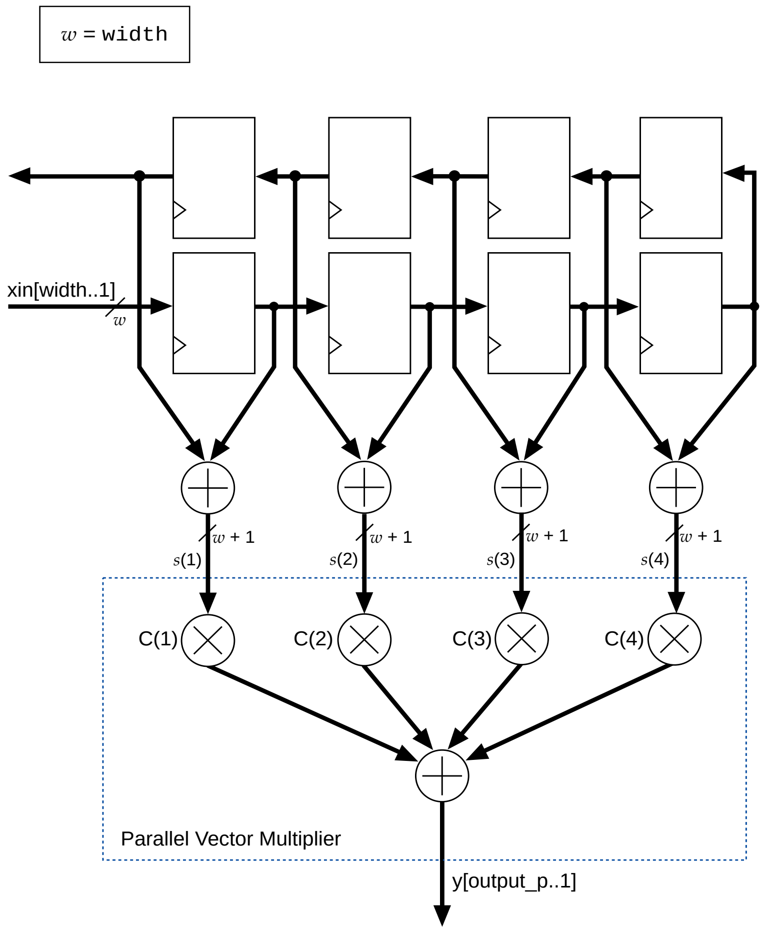


Figure 7. FIR_16TP Filter Block Diagram

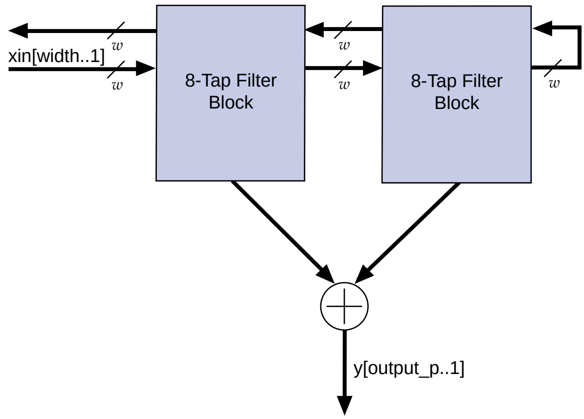


Figure 8. FIR_24TP Filter Block Diagram

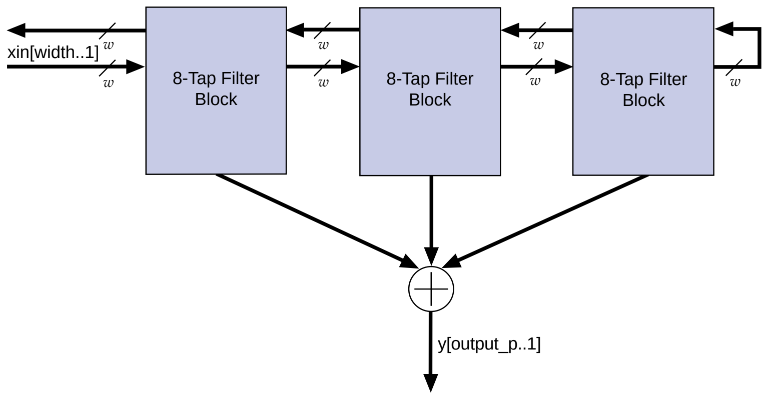
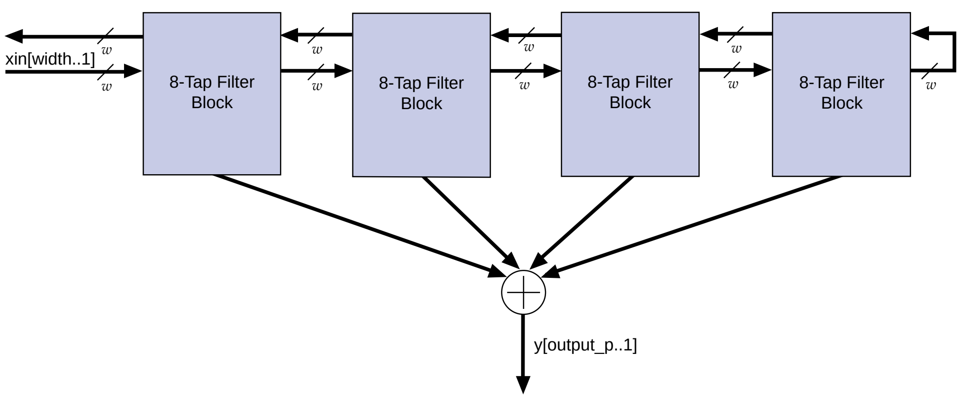


Figure 9. FIR_32TP Filter Block Diagram



Features

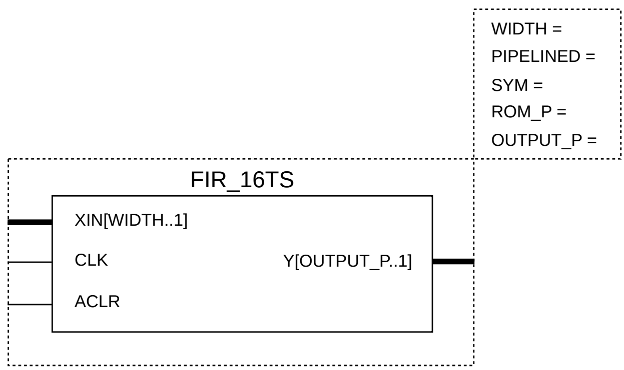
- Serial FIR filter reference designs
- Serial implementation for minimum resource usage
- High performance: > 6 MSPS
- Parameterized data widths, pipelining, and symmetry
- Symmetrical or antisymmetrical coefficients

General Description

The Altera serial FIR filters are small, linear-phase designs with operation speeds of over 6 MSPS. By efficiently using the Altera FLEX architecture, these filters provide extremely fast, flexible, and cost-effective design solutions. In addition, the filters use bit-serial computation, making them significantly smaller than parallel implementations. The filters are implemented using the vector-multiplier approach described in *Application Note 73* (*Implementing FIR Filters in FLEX Devices*). The coefficients are stored in LUTs and their values are determined at compilation; however, the coefficients can be changed by reconfiguring the device. Other tap lengths are available by modifying a copy of the AHDL TDF. These designs convert parallel data input into serial internal data for processing. Thus, each input sample must be applied to the input (`xin`) for `width + 1` clock cycles.

Figure 10 shows the symbol for the 16-tap serial FIR filter. All other serial FIR filters in the Altera DSP Design Kit have the same symbol. Coefficients are defined in an Include File (**.inc**) called **coef.inc**.

Figure 10. Serial FIR Filter Symbol



Function Prototype

The AHDL Function Prototype for the `fir_16ts` function is shown below and is the same for all other serial FIR filters in the Altera DSP Design Kit:

```
FUNCTION fir_16ts (xin[width..1] clk, aclr)
   WITH (width, pipelined, sym, rom_p, output_p)
   RETURNS (y[output_p..1]);
```

Parameters

Parameters for the serial FIR filter functions are provided in Table 4. In the default value for the `output_p` parameter, the constant `blocks` refers to the number of 8-tap blocks in the filter.

Table 4. Serial FIR Filter Parameters

Name	Default	Value	Description
`width`	8	Integers	Input word width (in bits)
`rom_p`	8	Integers	Coefficient precision
`pipelined`	"yes"	"yes" or "no"	Pipelined/non-pipelined
`sym`	"sym"	"sym" or "anti"	Symmetrical/antisymmetrical
`output_p`	`width+blocks+rom_p`	Integers	Output precision (in bits)

Ports

Input and output ports for the serial FIR filter functions are provided in Table 5.

Table 5. Input & Output Ports

Port Type	Name	Description
Input	`xin[width..1]`	Data input
Input	`clk`	Clock input
Input	`aclr`	Asynchronous Clear
Output	`y[output_p..1]`	Data output

Block Diagrams

Figures 11 and 12 show block diagrams for the `fir_16ts` and `fir_64ts` FIR filters.

Figure 11. FIR_16TS Filter Block Diagram

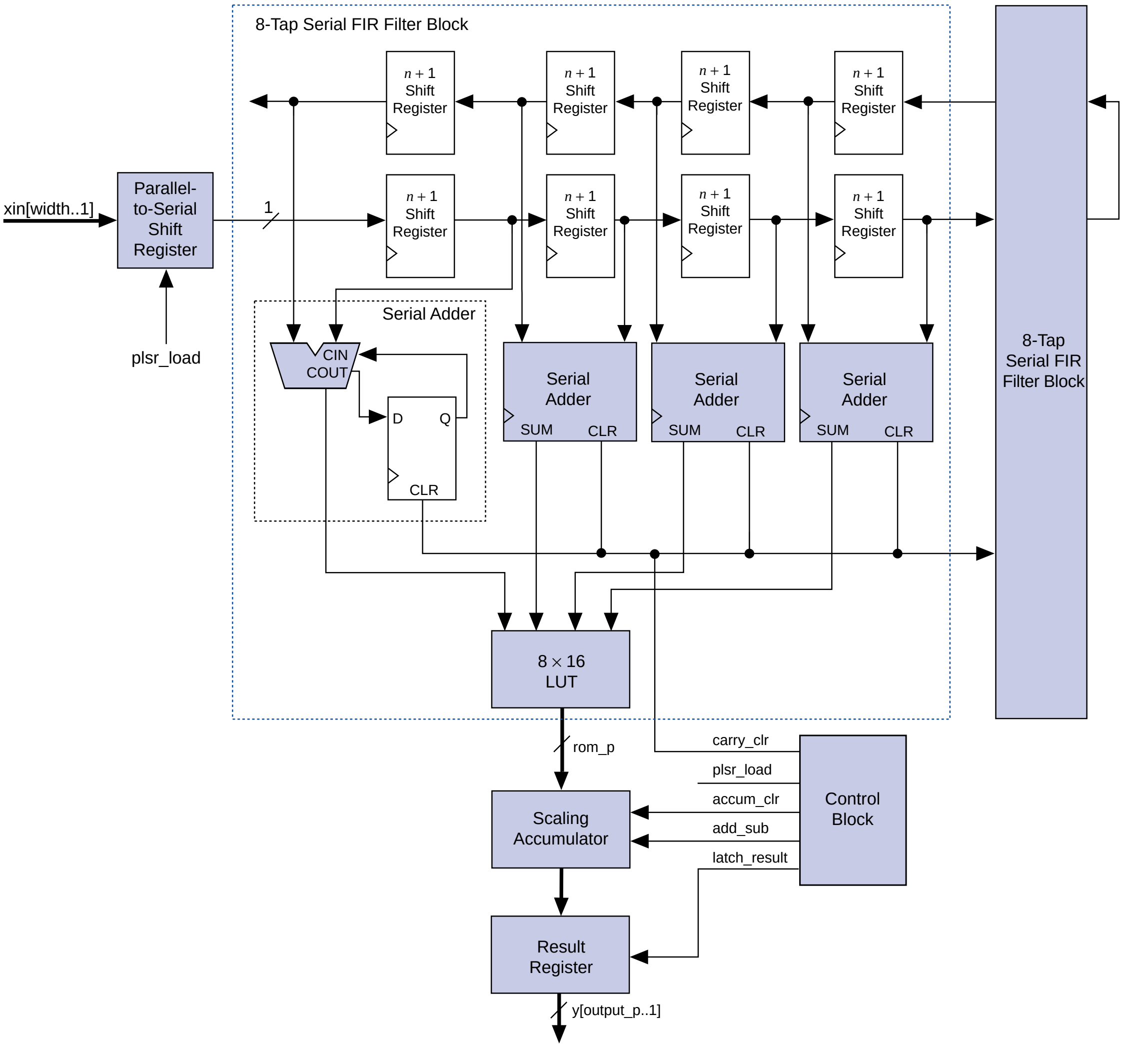
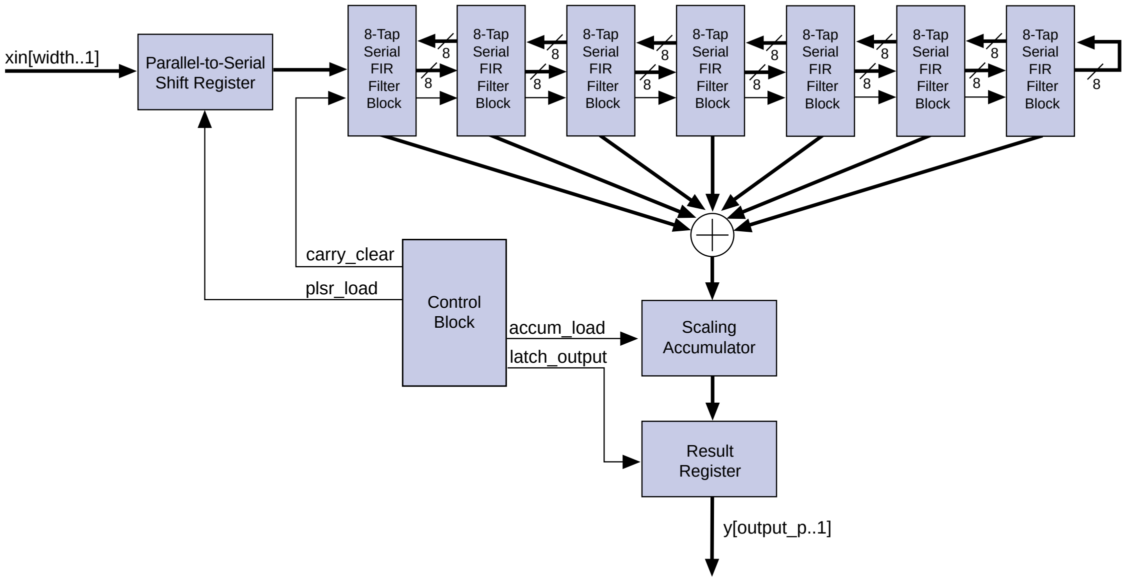


Figure 12. FIR_64TS Filter Block Diagram



Features

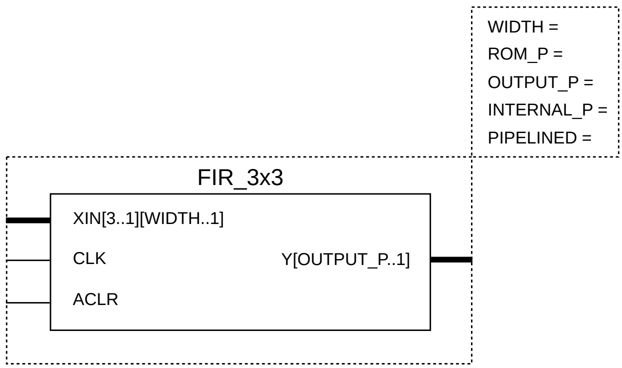
- Symmetrical video FIR filter reference design
- High performance: > 100 MSPS
- Parallel implementation for maximum performance
- Parameterized data widths and symmetry

General Description

The Altera `fir_3x3` symmetrical video FIR filter is a linear-phase design with an operation speed of over 100 MSPS. By efficiently using the Altera FLEX architecture, this filter provides an extremely fast, flexible, and cost-effective convolution operation. These filters are implemented with the vector-multiplier approach described in *Application Note 73 (Implementing FIR Filters in FLEX Devices)*. The coefficients are stored in LUTs and their values are determined at compilation; however, the coefficients can be changed by reconfiguring the device.

Figure 13 shows the symbol for the `fir_3x3` symmetrical video FIR filter.

Figure 13. FIR_3x3 Video Filter Symbol



Function Prototype

The AHDL Function Prototype for the `fir_3x3` function is shown below:

```
FUNCTION fir_3x3 (xin[3..1][width..1], clk, aclr)
    WITH (width, rom_p, output_p, internal_p,
    pipelined)
    RETURNS (y[output_p..1]);
```

Parameters

Parameters for the fir_3x3 function design are provided in Table 7.

Table 6. fir_3x3 Parameters

Name	Default	Value	Description
width	8	Integers	Input word width (in bits)
rom_p	8	Integers	Coefficient precision
pipelined	"yes"	"yes" or "no"	Symmetrical/antisymmetrical
output_p	width+2+rom_p	Integers	Output precision (in bits)
internal_p	= output_p	= output_p	Pipelined/non-pipelined

Ports

Input and output ports for the fir_3x3 function are provided in Table 7.

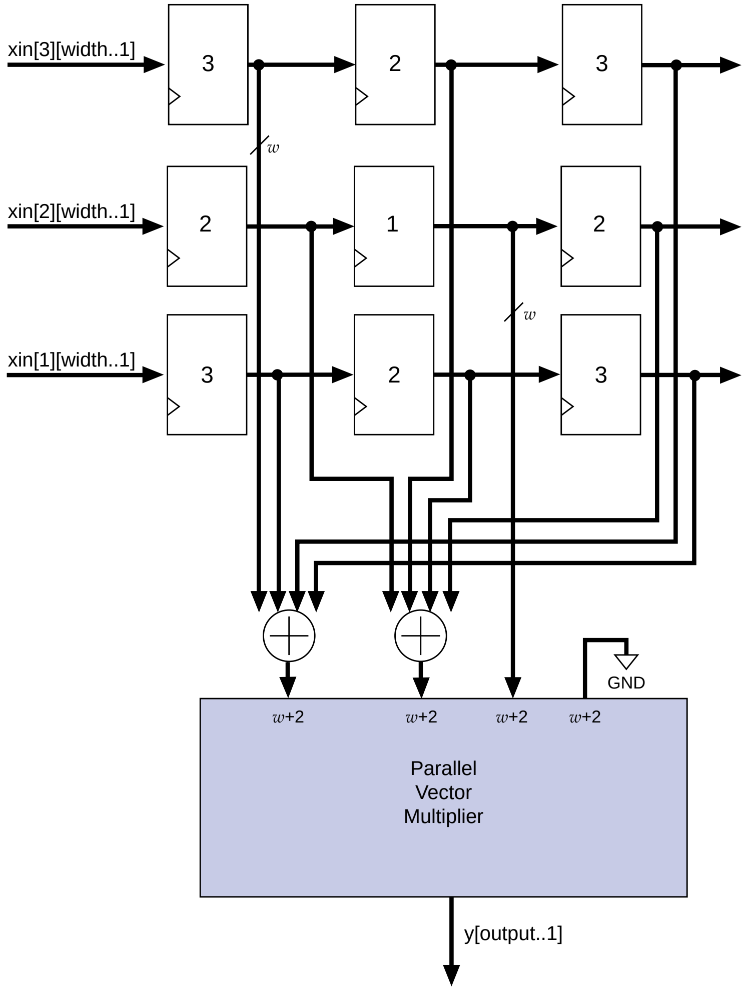
Table 7. Input & Output Ports

Port Type	Name	Description
Input	xin[3..1][8..1]	Data input
Input	clk	Clock input
Input	aclr	Asynchronous Clear
Output	y[output_p..1]	Data output

Block Diagram

Figure 14 shows the block diagram of the fir_3x3 video filter. One input to the vector multiplier is zero and the other three inputs are the sums of the symmetric taps. The n-bit value from pixel 1 is stuffed with two leading zeroes so that all inputs to the parallel vector multiplier are $n + 2$ bits wide.

Figure 14. FIR_3x3 Video Filter Block Diagram



Notes:

ALTERA®

2610 Orchard Parkway
San Jose, CA 95134-2020
(408) 894-7000
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 894-7104
Literature Services:
(408) 894-7144

I.S. EN ISO 9001

Printed on Recycled Paper.