

November 21, 1994

Application Note BY GREG GOSLIN & BRUCE NEWGARD

## Summary

This application note describes the functionality and integration of a 16-Tap, 8-Bit Finite Impulse Response (FIR) filter macro with predefined coefficients (e.g. low pass) and a sample rate of 5.44 mega-samples per second or 784 MIPS using an XC4000-4 device. The application note also describes how to set the coefficients of the FIR Filter to meet the needs of other applications.

## Xilinx Family

XC4000

## Demonstrates

RAM-based Look Up Tables for parallel-to-serial shift registers.

ROM-based Look Up Tables for parallel arithmetic operations

Distributed arithmetic for high-speed parallel calculations.

MEMGEN and ROM data entry

## Table of Contents

<a href="#">Introduction .....</a>	<a href="#">1</a>
<a href="#">FIR Filter Coefficients .....</a>	<a href="#">2</a>
<a href="#">Design Implementation Guidelines .....</a>	<a href="#">3</a>
<a href="#">Converting the 16-Tap into a 15-Tap FIR .....</a>	<a href="#">3</a>
<a href="#">Using the Design Files .....</a>	<a href="#">3</a>
<a href="#">Acknowledgments .....</a>	<a href="#">4</a>

## Introduction

Xilinx provides a wide range of programmable logic devices that can be used to develop high-performance DSP products. The XC4000 family of Field Programmable Gate Arrays (FPGAs) offer designers a diverse selection of functionality to build almost any system design. The FPGA can adapt to last minute design modifications as well as future design iterations without making extensive hardware or software modifications—

saving both time and money.

The Xilinx RPM functional implementation of the 16-Tap FIR filter shown in Figure 1 is a cost-effective building block for the creative DSP engineer. The design is based on the Xilinx XC4000 family architecture. The XC4000 family provides ample density to build a complete DSP system in a single device. The FIR macro, as implemented, is a low-pass filter. It can be reconfigured as a high-pass or band-pass filter by simply reprogramming the filter coefficients. The high sample rate (e.g. 5.44 million samples per second) and the ability to change many characteristics of the design—including the number of taps—makes this FIR filter a versatile macro with a diverse range of DSP applications.

## Functional Description

The 16-Tap 8-Bit FIR filter is based on a distributed arithmetic algorithm pioneered by Les Mintzer. The filter consists of seven major components:

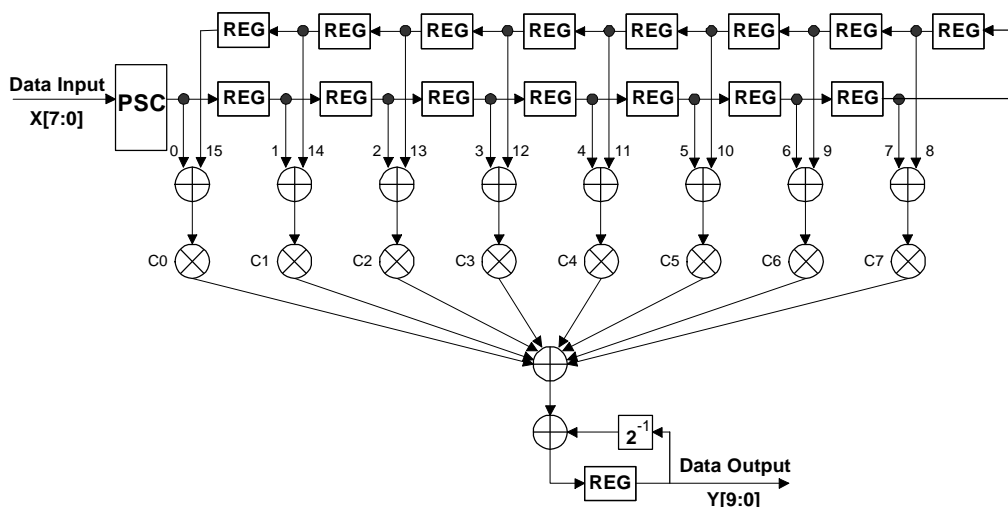


Figure 1. Data Flow Diagram for 16-Tap FIR Filter.

- a parallel to serial converter (PSC\_8),
- a RAM-based shift register (TSB\_8),
- a serial adder (S\_ADD4),
- a ROM-based Look-Up Table (LUT16X8),
- a complementing register (REG\_COMP),
- an adder (REG\_ADD9), and
- a scaling accumulator (SCAL10).

For the following discussion, refer to Figure 5 and Figure 6.

The FORMAT input signal indicates whether the incoming data is unsigned binary or 2's-complement. If FORMAT is High, then the data is unsigned binary.

An 8-bit data sample is loaded into the parallel to serial converter (PSC\_8) at the sample rate. The PSC generates a serial output stream that is supplied to the RAM-based shift registers at the bit clock rate. The bit clock rate is determined by

$$\text{bit clock} = (n + 1) \times (\text{sample rate})$$

where (n+1) represents the number of data bits per sample plus an overflow bit.

The RAM-based shift register (TSB\_8) functions as a time-skew buffer. The data bit stored in a particular address of a RAM-based shift register is first read and stored in a flip-flop. A "new" data bit is then written into the particular address replacing the previous data. The data bit stored in the flip-flop is written to the next location.

The RAM-based shift register approach—rather than a more-traditional cascade of data registers—significantly reduces the overall size of the FIR filter. Note the special configuration of the two RAM-based shift registers to control the distribution of data. The upper RAM-based shift register contains TAPS[3:0] and TAPS[15:12] while the lower RAM-based shift register contains TAPS[7:4] and TAPS[11:8]. This configuration is important because it exploits the inherent symmetrical coefficient values used in a linear-phase FIR filter.

$$3x_0 + 5x_1 + \dots + 5x_{14} + 3x_{15}$$

**Figure 2. Symmetry of Coefficients.**

Since the task of summing the 16 taps is split into two sections, TAPS[3:0] and TAPS[15:12] are added and stored in the upper serial adder. TAPS[7:4] and TAPS[11:8] are added and stored in the lower serial adder.

The outputs of the registered serial adders are presented to the ROM-based lookup tables. Since the coefficients are symmetric, the sums generated by the adders can be multiplied by the same coefficients. Since all possible partial products are pre-computed, the outputs of the serial adders are used to address the ROM-based lookup tables to generate the appropriate multiplication results. The outputs of the registered lookup tables are summed except for the sign result which is complemented before being summed. The registered summation is then fed into a scaling accumulator.

The scaling accumulator (SCAL\_10) is first loaded with the least significant bit (LSB) result. Each successive result is added to the summation which has been bit shifted to perform a divide by two function. This process continues until the most significant bit (MSB), or sign bit, is present. It should be noted that to generate a 2's-complement number the sign result is complemented and a carry is forced at the adder and the scaling accumulator. Since two separate sign values were complemented, a two must be added. Once the sign result is accumulated, the final filter result is registered.

The overall timing is controlled by a four-bit counter running at the bit rate. Decoded counter outputs generate the control signals needed to store the filter values in the proper sequence.

## FIR Filter Coefficients

The FIR filter coefficients are formatted and stored in ROM-based LUTs. Because the FIR filter coefficients are symmetric, the number of unique coefficients is half the number of taps. Consequently, the 16-taps for this example result in eight unique coefficients. The eight coefficients are split into two 16-word x 8-bit ROM-based LUTs formatted as shown in Figure 3.

Address	Data	Address	Data
0000	0	1000	C <sub>3</sub>
0001	C <sub>0</sub>	1001	C <sub>3</sub> + C <sub>0</sub>
0010	C <sub>1</sub>	1010	C <sub>3</sub> + C <sub>1</sub>
0011	C <sub>1</sub> + C <sub>0</sub>	1011	C <sub>3</sub> + C <sub>1</sub> + C <sub>0</sub>
0100	C <sub>2</sub>	1100	C <sub>3</sub> + C <sub>2</sub>
0101	C <sub>2</sub> + C <sub>0</sub>	1101	C <sub>3</sub> + C <sub>2</sub> + C <sub>0</sub>
0110	C <sub>2</sub> + C <sub>1</sub>	1110	C <sub>3</sub> + C <sub>2</sub> + C <sub>1</sub>
0111	C <sub>2</sub> + C <sub>1</sub> + C <sub>0</sub>	1111	C <sub>3</sub> + C <sub>2</sub> + C <sub>1</sub> + C <sub>0</sub>

(e.g. C<sub>0A</sub>=-2, C<sub>1A</sub>=-4, C<sub>2A</sub>=5, C<sub>3A</sub>=-5  
C<sub>0B</sub>=3, C<sub>1B</sub>=3, C<sub>2B</sub>=-18, C<sub>3B</sub>=78)

**Figure 3. ROM-Based Coefficient LUT**

All possible partial products are pre-computed and placed in a memory definition file or .MEM file, as shown in Figure 4 (refer to the MemGen Program chapter in the XACT Reference Guide, Volume 1-April, 1994, pages I-59 through I-67).

```
;DA_LUTA.MEM: A 16-word deep, 8-bit wide ROM
TYPE ROM ;The memory is a ROM
DEPTH 16 ;The memory is 16-words deep
WIDTH 8 ;The memory word is 8-bits wide
SYMBOL NONE
DEFAULT 0 ;Default value for unspecified locations
DATA ;User defined coef. ROM data starts
here
00,FE,FC,FA ; 0, C0, C1, (C0+C1)
05,03,01,FF ; C2, (C0+C2), (C1+C2), (C0+C1+C2)
FB,F9,F7,F5 ; C3, (C0+C3), (C1+C3), (C0+C1+C3), (C1+C3)
00,FE,FC,FA ; (C0+C1+C3), (C2+C3), (C0+C2+C3),
(C0+C1+C2+C3)
;END OF FILE
```

**Figure 4. Memory Definition File**

Note: After the .MEM files are processed by the MEMGEN memory compiler, the RLOC's must be manually changed to match the RLOC's in the original design files.

## Design Implementation Guidelines

The bit clock must use a BUFGS primitive, not a BUFGP. This is because the bit clock is also used to control the write-enable signals in the RAM-based shift registers (TSB\_8). A BUFGS connects to the write control signals on a Configurable Logic Block (CLB) while a BUFGP primitive does not.

The maximum bit clock rate is 49 MHz using an XC4000-4 device. The 49 MHz clock rate corresponds to 5.44 million samples per second because it takes nine bit rate clock periods to add two eight bit numbers with overflow. The critical path is in the carry chain of the final scaling accumulator. The RAM-based shift registers require a minimum clock width for both High and Low of 9.9 ns. Pipelining the arithmetic functions in the design increases the overall performance but also increases the logic resource requirements.

Since the 16-Tap 8-Bit FIR filter design is a Relationally Placed Macro (RPM), it must be placed so that its placement does not conflict with other design modules. The shape of the RPM is a 9 CLB row by 10 CLB column rectangular block and occupies 67 CLBs.

## Converting the 16-Tap into a 15-Tap FIR

The 16-Tap FIR filter design can be changed into a 15-Tap FIR filter with two simple modifications.

First, the center two Taps of the 16-Tap FIR must register the same data. This is done by schematically changing the 8th Tap data source from TAP7 to TAP6 (e.g. change the DI\_4 input of the lower TSB\_8 symbol from TAP7 to TAP6.)

Secondly, since the 7th Coefficient corresponds to a single Tap with a 15-Tap FIR and the data registered at the 7th and 8th Taps are equal, the sum of the two Taps must be divided by two. This is done by dividing the 7th Coefficient by 2 (e.g. C<sub>8</sub> in the lower ROM-based look up table, LUT\_B, must be changed to C<sub>8</sub>/2.)

## Using the Design Files

This design is available from the Xilinx BBS (see "Xilinx Technical Bulletin Board" in Section 6 of the **Xilinx Data Book**.) This section describes what software is required to run the design and the steps involved. Also, please read through the **Limitations and Restrictions** section.

## Software Requirements

The following software is required to process this design:

- VIEWdraw or VIEWdraw-LCA schematic editor. This software is required in order to make modifications to the schematics.
- Xilinx XACT 5.0 FPGA development system, including the PPR place and route program and the X-BLOX module generator.

## Using the Design on Your System

1. Create a new directory on your hard disk called C:\FILTER
2. Copy all of the files and sub-directories from the DSP\DESIGNS\ directory on the **Programmable Logic Breakthrough '95** CD-ROM.
3. Make sure that the VIEWlogic design library pointers in the viewdraw.ini file are set appropriately for your directory structure
4. Invoke XDM.
5. Set the part type to XC4003PC84-4.
6. Run XMAKE on FIR\_LCA.1 to process the design. The schematic file named FIR\_LCA.1 is the top-level schematic.

Limitations and Restrictions

WARNING:

Xilinx, Inc. does not make any representation or warranty regarding this design or any item based on this design. Xilinx disclaims all express and implied warranties, including but not limited to the implied fitness of this design for a particular purpose and freedom from infringement. Without limiting the generality of the foregoing, Xilinx does not make any warranty of any kind that any item developed based on this design, or any portion of it, will not infringe any copyright, patent, trade secret or other intellectual property right of any person or entity in any country. It is the responsibility of the user to seek licenses for such intellectual property rights were applicable. Xilinx shall not be liable for any damages arising out of or in connection with the use of the design including liability for lost profit, business interruption, or any other damages whatsoever.

Design Support and Feedback

This application note may undergo future revisions and additions. If you would like to be updated with new versions of this application note, or if you have questions, comments, or suggestions please send an E-mail to

[dsp@xilinx.com](mailto:dsp@xilinx.com)

or a FAX addressed to "Corporate Applications—FIR Filter Macro" at (408) 879-4442.

**IMPORTANT:** Please be sure to include which version of the application note you are using. The version number is in the lower right-hand corner of page 1.

Acknowledgments

Special thanks to Les Mintzer, research and development consultant and U.C. Irvine professor, for creating the enabling technology described herein.

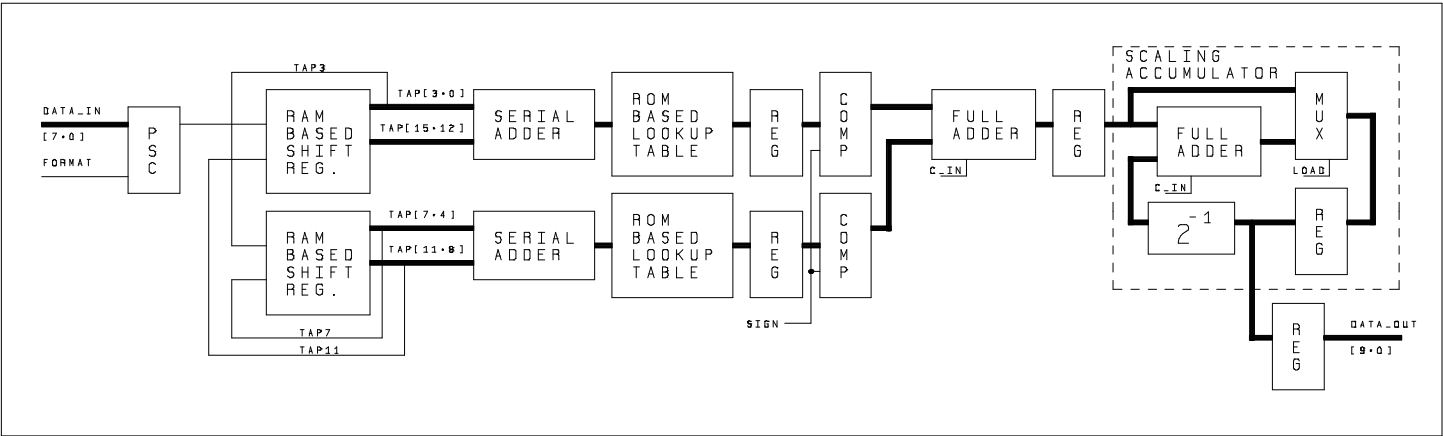


Figure 5. Data Flow Block Diagram for 16-Tap FIR Filter RPM.

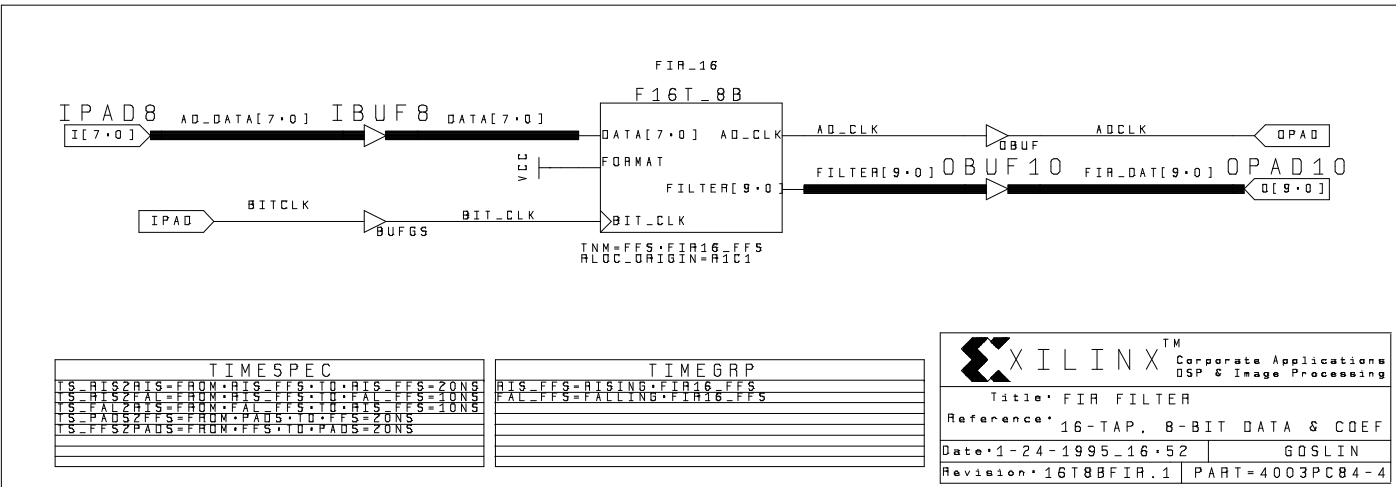


Figure 6. 16-Tap 8-Bit FIR Filter RPM High-Level Schematic.

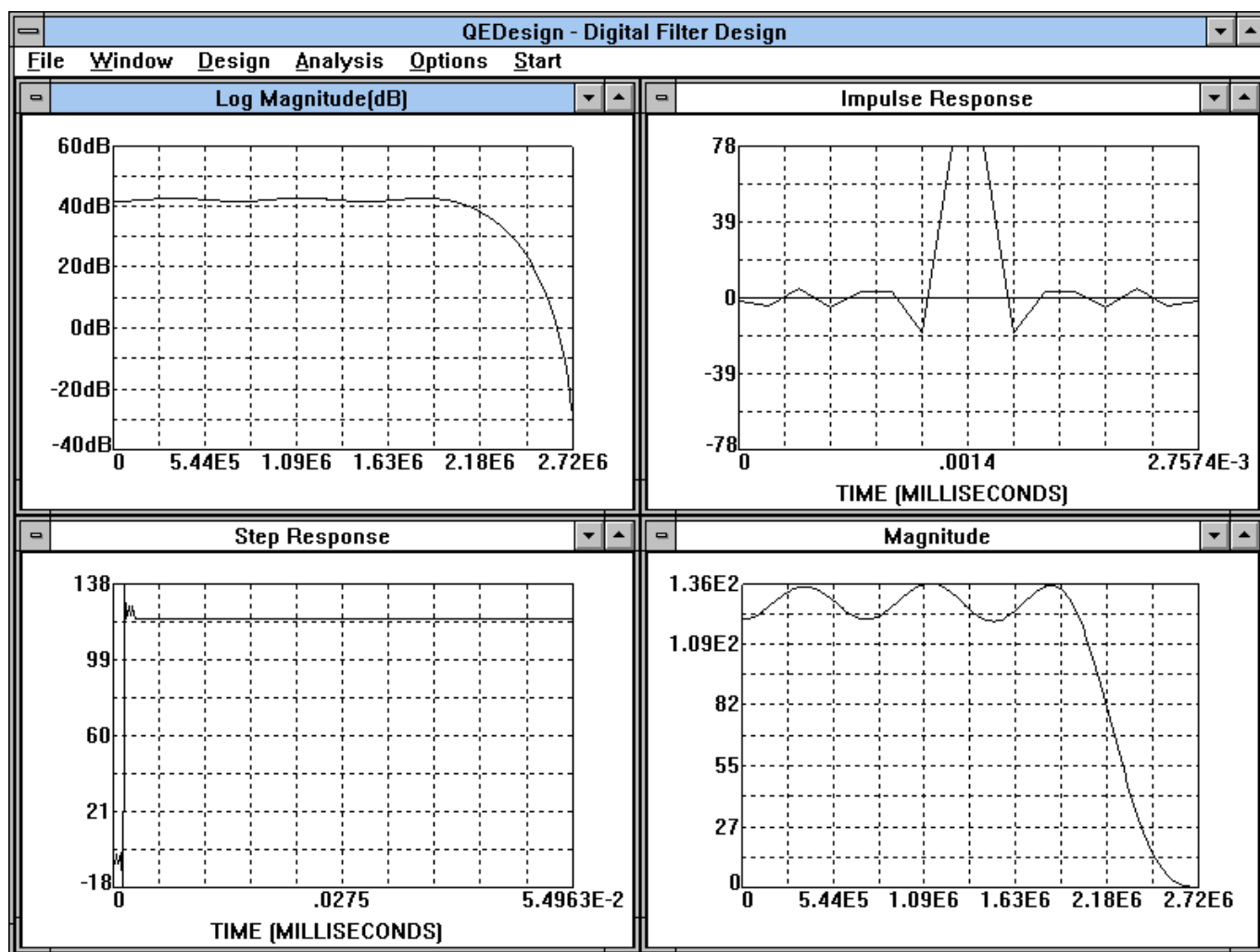


Figure 7. 16-tap, 8-bit FIR low-pass filter characteristics.