



Performance Analysis of FIR Filter Design Using Reconfigurable Mac Unit

Mohammad Aadil

Research Scholar M. Tech.

Takshshila Institute of Engineering & Technology,
Jabalpur (M.P.), [INDIA]

Email: m.aadil078692@gmail.com

Shobhit Verma

Assistant Professor

Takshshila Institute of Engineering & Technology,
Jabalpur (M.P.), [INDIA]

Email: shobhitverma@takshshila.org

Abstract—Digital filters are commonly used as an essential element of everyday electronics such as radios, cell phones, and stereo receivers. Digital filters may be more expensive than an equivalent analog filter due to their increased complexity, but they may lead to many designs which are impractical or impossible as analog filters. Basically there are two types of digital filters, which are FIR (Finite Impulse Response) and IIR (Infinite Impulse Response). FIR filters are highly desirable in digital filter design because of their inherent stability and linear phase. In this paper FIR filter has been designed by using a signed multiplier and a signed Adder. Combination of Signed multiplier and Signed adder makes FIR filter faster. RTL synthesis has been done by using Xilinx 14.1 Proposed unit is efficient in terms of speed and complexity. We have achieved delay for filter unit as 18.752ns.

Keywords:— Low Power, Signed Adder, Signed Multiplier, Multiplexer, VLSI, FPGA.

1. INTRODUCTION

Their impulse response, which can be either finite or infinite. The methods for designing and implementing these two filter classes differ considerably. Finite impulse response (FIR) filters are digital filters whose response to a unit impulse (unit sample function) is finite in duration. This is in contrast to infinite impulse response (IIR)

filters whose response to a unit impulse (unit sample function) is infinite in duration. FIR and IIR filters each have advantages and disadvantages, and neither is best in all situations. FIR filters can be implemented using either recursive or nonrecursive techniques, but usually nonrecursive techniques are used. FIR filters are widely used in digital signal processing (DSP) systems that are characterized by the extensive sequence of multiplication operations. FIR filters are widely used in various DSP applications. In some applications, the FIR filter circuit must be able to operate at high sample rates, while in other applications, the FIR filter circuit must be a low-power circuit operating at moderate sample rates. The low-power or low-area techniques developed specifically for digital filters can be found in. Parallel (or block) processing can be applied to digital FIR filters to either increase the effective throughput or reduce the power consumption of the original filter. While sequential FIR filter implementation has been given extensive consideration, very little work has been done that deals directly with reducing the hardware complexity or power consumption of parallel FIR filters [1]. Traditionally, the application of parallel processing to an FIR filter involves the replication of the hardware units that exist in the original filter. The topology of the multiplier circuit also affects the resultant power consumption. Choosing multipliers with more hardware breadth rather than depth

would not only reduce the delay, but also the total power consumption [2]. A lot of design methods of low power digital FIR filter are proposed, for example, in [3] they present a method implementing fir filters using just registered address and hardwired shifts. They extensively use a modified common sub expression elimination algorithm to reduce the number of adders. In [4] they have proposed a novel approach for a design method of a low power digital baseband processing. Their approach is to optimize the bitwidth of each filter coefficient. They define the problem to find optimized bitwidth of each filter coefficient. In [5] presents the method reduce dynamic switching power of a fir filter using data transition power diminution technique (DPDT). This technique is used on adders, booth multipliers. In [6] this research proposes a pipelined variable precision gating scheme to improve the power awareness of the system. This research illustrates this technique is to clock gating to registers in both data flow direction and vertical to data flow direction within the individual pipeline stage based on the input data precision. The rest of the paper is structured as follow. Section2 gives a summary of fir filter theory and in Section3 presents the architecture adopted in our implementation. Comparison of our implementation with those done is given at section4. Finally section5 provides the conclusion of this paper.

2. FIR FILTER THEORY

A digital filter takes a digital input, gives a digital output, and consists of digital components. In a typical digital filtering application, software running on a DSP reads input samples from an A/D converter, performs the mathematical manipulations dictated by theory for the required filter type, and outputs the result via a D/A converter. An analog filter, by contrast, operates directly on the analog inputs and is built entirely with analog components, such as resistors, capacitors, and inductors. There are many filter types, but the most common are lowpass, highpass, bandpass, and bandstop. A lowpass

filter allows only low frequency signals (below some specified cutoff) through to its output, so it can be used to eliminate high frequencies. A lowpass filter is handy, in that regard, for limiting the upper most range of frequencies in an audio signal; it's the type of filter that a phone line resembles. A highpass filter does just the opposite, by rejecting only frequency components below some threshold. An example highpass application is cutting out the audible 60Hz AC power "hum", which can be picked up as noise accompanying almost any signal in the U.S. The designer of a cell phone or any other sort of wireless transmitter would typically place an analog bandpass filter in its output RF stage, to ensure that only output signals within its narrow, government-authorized range of the frequency spectrum are transmitted. Engineers can use bandstop filters, which pass both low and high frequencies, to block a predefined range of frequencies in the middle. A. Frequency response Simple filters are usually defined by their responses to the individual frequency components that constitute the input signal. There are three different types of responses. A filter's response to different frequencies is characterized as passband, transition band, or stopband. The passband response is the filter's effect on frequency components that are passed through (mostly) unchanged. Frequencies within a filter's stopband are, by contrast, highly attenuated. The transition band represents frequencies in the middle, which may receive some attenuation but are not removed completely from the output signal. In Fig. 1, which shows the frequency response of a

lowpass filter, ω_p is the passband ending frequency, ω_s is the stopband beginning frequency, and A_s is the amount of attenuation in the stopband. Frequencies between ω_p and ω_s fall within the transition band and are attenuated to some lesser degree.

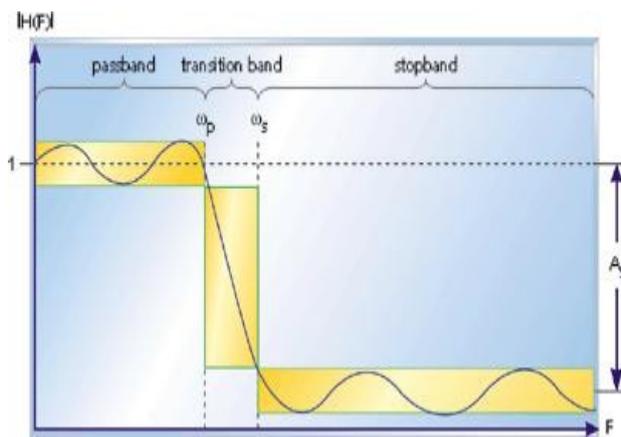


Figure 1. The response of a lowpass filter to various input frequencies.

Given these individual filter parameters, one of numerous filter design software packages can generate the required signal processing equations and coefficients for implementation on a DSP. Before we can talk about specific implementations, however, some additional terms need to be introduced. Ripple is usually specified as a peak-to-peak level in decibels. It describes how little or how much the filter's amplitude varies within a band. Smaller amounts of ripple represent more consistent response and are generally preferable. Transition bandwidth describes how quickly a filter transitions from a passband to a stopband, or vice versa. The more rapid this transition, the higher the transition bandwidth; and the more difficult the filter is to achieve.

Though an almost instantaneous transition to full attenuation is typically desired, real-world filters don't often have such ideal frequency response curves. There is, however, a tradeoff between ripple and transition bandwidth, so that decreasing either will only serve to increase the other.[7]

Digital filters are typically used to modify or alter the attributes of a signal in the time or frequency domain. The most common digital filter is the linear time-invariant (LTI) filter. An LTI interacts with its input signal through a process called linear convolution, denoted by $y = h \times x$ where h is the filter's impulse response, x is the input signal, and y is

the convolved output. The linear convolution process is formally defined by:

$$y[n] = x[n]h[n] = \sum_{k=0}^{L-1} x[n-k]h[k] = \sum_{k=0}^{L-1} h[k]x[n-k]. \quad (1)$$

$$k=0$$

An FIR with constant coefficients is an LTI digital filter. The output of an FIR of order or length L , to an input time-series $x[n]$, is given by a finite version of the convolution sum given in (1), namely:

$$y[n] = \sum_{k=0}^{L-1} x[n-k]h[k] = \sum_{k=0}^{L-1} h[k]x[n-k] \quad (2)$$

where $h[0] \neq 0$ through $h[L-1] \neq 0$ are the filter's L coefficients.[8]

Figure 2 shows the basic block diagram for an FIR filter of length N . The delays result in operating on prior input samples. The h_k values are the coefficients used for multiplication, so that the output at time n is the summation of all the delayed samples multiplied by the appropriate coefficients.

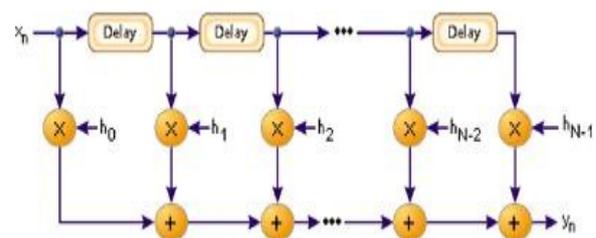


Figure 2. The logical structure of an FIR filter.

The process of selecting the filter's length and coefficients is called filter design. The goal is to set those parameters such that certain desired stop band and pass band parameters will result from running the filter. Most engineers utilize a program such as MATLAB to do their filter design. But whatever tool is used, the results of the design effort should response can be tuned. With the length, N , and coefficients, float $h[N]=\{\dots\}$, decided upon, the implementation of the FIR filter is fairly straightforward. Listing 1 shows how it could be done in C. Running this code

on a processor with a multiply-and-accumulate instruction (and a compiler that knows how to use it) is essential to achieving a large number of taps.

First we arrange decimal coefficients according to negative and positive power of 2 (no need to them into integer). So the filter hardware and power consumption will reduce.

3 . IMPLEMENTATION FIR FILTER

In this paper, MAC architecture is considered. There is design and implementation low power multiplexer based shift/add multiplier for reduce the power consumption. Fir filter is implemented using the shift and add method. We perform all our optimization in the multiplier block. This multiplier is constitution of one shifter block (barrel shifter) one adder block and also control unit. Where in shifter block is based on multiplexer only. And proportional to select signal(shift[2..0]) input number left shift to magnitude 0 to 7 magnitude of shift specification by control unit i.e. based on weight of bits input signal to control unit. If slightly bit is zero we have non-shift and slightly bit is one to magnitude weight of bit shift is apply. That in continue show VHDL code of this block. And will has been explanation in related with shifter block. Adder block add together shifted number, final product favorable output. By implementation of shifter block based on multiplexer hardware and power consumption of FIR filter is very reduced. in Fig. 3 shown block diagram of low power multiplexer based shift/add multiplier. The constant multiplications are decomposed in to additions and shifts and the multiplication complexity is reduced. Its possible to implement the design in the two form described below:

The coefficients are changed to integer getting multiplied to a multiple power of 10, then we arrange these coefficients the positive power of 2. This procedure is shown below in graph form. Graph branches stand for left shift and nothes stand for sum.

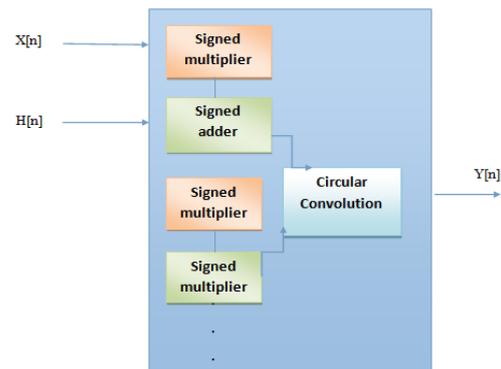


Figure 3. Block diagram of low area signed adder and signed multiplier based fir filter

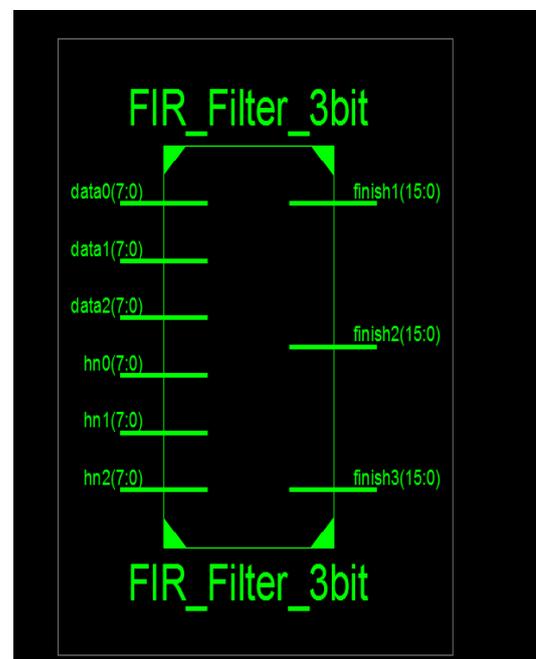


Figure 4: Top view of fir filter

4. COMPARATIVE RESULT

Proposed FIR filter speed solely depends upon effectiveness and efficiency of proposed Signed multiplier and adder. That's why to showcase efficiency and effectiveness of proposed Signed multiplier that has been implemented and compared with other popular commercially available filter lie on different algorithms on the same platform of target FPGA, which has been used to implement these popular FIR filter structures. Proposed module has been compared against different modules with same target FPGA on which previous has been implemented and then

showcased comparison on the basis of availability of result Comparison table are:

Author	Structure (N=3)	Delay(ns)	Area (No. of bonded IOS)
Mr. D. Ashok Kumar et al [6]	Fast FIR Algorithm	25.593	128
	Symmetric convolution	24.289	225
Result obtained	Convolution using signed adder & signed multiplier	18.752	96

5. CONCLUSION

The realization structure of FIR filter consists of a MAC operation which is made up of multipliers and adders. A FIR filter has been designed by signed multiplier and signed adder. Proposed FIR filter has been designed by using behavioral modeling. Synthesis of proposed work has been done successfully. Proposed MAC unit can work for n number of bits. For this paper synthesis has been done for 3 bit and 8 bit input. MAC unit is high speed unit which makes FIR filter faster.

REFERENCES:

[1] Abhijit Chandra, Design of hardware efficient FIR filter: A review of the state-of-the-art Approaches”, ELSEVIER 2016.

[2] Xin Lou, “Novel Structure for Area-Efficient Implementation of FIR Filters” IEEE 2016.

[3] Deepika, “Design of FIR Filter Using Reconfigurable MAC Unit, IEEE 2016.

[4] Nidhi Mishra, Literature Review on FIR Filter Optimization”, IJDACR, Volume 4, Issue 1, August 2015.

[5] Navneet Kaur “Optimal design of sparse fir filter using genetic” IJEST July, 2015.

[6] Mr. D. Ashok Kumar, “Design and Study of Modified Parallel FIR Filter Using Fast FIR Algorithm and Symmetric Convolution” IEEE 2014.

[7] Neha, Design of Linear Phase Low Pass FIR Filter using Particle Swarm Optimization Algorithm”, IJCA, Volume 98– No.3, July 2014

[8] Amanpreet Singh,” Comparison of Different Techniques to Design an Efficient FIR Digital Filter”, Proceedings of the World Congress on Engineering 2014 Vol 1, WCE 2014, July 2 - 4, 2014, London, U.K.