

# Design and Simulation of FIR Filter

Sushmitha.C<sup>1</sup>, Swathy.R<sup>2</sup>, Veena Devi.S<sup>3</sup>, Esther Jeba Rani.S.A<sup>4</sup>, Nagaraju.N<sup>5</sup>

U.G. Scholars, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India<sup>1,2,3,4</sup>

Assistant Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India<sup>5</sup>

**ABSTRACT:** Digital filtering algorithms are most commonly implemented using general purpose digital signal processing chips for audio applications, or special purpose digital filtering chips and application specific integrated circuits(ASICs).A filter may be required to have a given frequency response, or a specific response to an impulse, step, or ramp, or simulate an analog system. Depending on the response of the system, digital filters can be classified into Finite Impulse Response (FIR) filters & Infinite Impulse Response (IIR) filters. The Finite Impulse Response (FIR) is used as a fundamental processing element in any Digital signal processing (DSP) system.. The Project is focused on Designing FIR filter using XILINX 12.3.

**KEYWORDS:** DSP, FIR, Tap, Impulse Response, Delay element, non-recursive.

## I INTRODUCTION

The basic operation in digital signal processing is filtering. This operation is widely used in many electronic devices to cancel part of signal that is redundant or damages the signal. Filters have two uses: signal *separation* and signal *restoration*. Signals which are corrupted by interference and noise require separation techniques. A device for measuring the electrical activity of a baby's heart inside the mother's womb will be corrupted by breath signal and heartbeat signal of the mother. At such times filters are used to separate the signals and analyse them individually. When signal gets distorted the process of signal restoration is used. Analog filters, cheaper, faster, and large dynamic range in both amplitude and frequency. Digital filters [1] in comparison, vastly superior performance level that can be achieved. Quality is better than analog filters to digital filters can achieve performance unique. The filtering problem is approached makes a dramatic difference. With analog filters, emphasizing precision and stability, such as resistors and capacitors in electronics, controls have limitations. In comparison, digital filters are often ignored in order to better filter performance. The emphasis shifts signal constraints, and the processing of theoretical issues. The easiest way to implement a digital filter response is convolution of the digital impulse response with the discrete time signal input. All linear filters can be made in this manner.

## II. FILTERS

Filters are useful for separating, extracting, and restoring signals. They can help remove contaminating sources of noise (like 60 Hz) or isolate tones and patterns within a carrier signal. They can also improve a signal such as poorly sampled audio to better represent the original intent (equalizing).

There are two main types of filter,

- a) Analog filters
- b) Digital filters

### 2.1 Analog Filter

An analog filter has an analog signal at both its input  $x(t)$  and its output  $y(t)$ . Both  $x(t)$  and  $y(t)$  are functions of a continuous variable time ( $t$ ) and can have an infinite number of values. An analog filter uses analog electronic circuits made up from components such as resistors, capacitors and op amps to produce the required filtering

effect. Such filter circuits are widely used in applications such as noise reduction, video signal enhancement, graphic equalizers in hi-fi systems, and many other areas.

## 2.2 Digital Filter

Digital filters are linear systems that in their most straightforward form can be implemented by convolving the discrete samples of an input signal with the filter's impulse response. The impulse response is basically a sequence of numbers that represents the response of the filter to an impulse input. These numbers are also known as the filter coefficients. Finite Impulse Response (FIR) filters use a method where each output sample is calculated by first multiplying the samples of the input signal by the coefficients, and then adding them together (accumulating). A characteristic of FIR filters [2] is that their response to an impulse ultimately settles to zero thus is finite in length. Another approach uses recursion to perform digital filtering. Here previously calculated values in the output as well as values in the input signal are used. These are Infinite Impulse Response (IIR) filters and are defined by a set of recursive coefficients not given by the impulse response but by other means such as a bilinear transform of an analog filter response. Since the output is used in performing these filters, their response to an impulse is non-zero over an infinite length of time, thus is infinite in length. The basic building blocks used to implement digital filters are multipliers, adders, and storage cells (i.e., RAM). The availability of these resources and how efficiently they are used is a key part of the overall performance of the filter. Here is some additional information on the two types (FIR/IIR) of digital filters.

### 2.2.1 Types of Digital Filter

Filters can be classified in several different groups, depending on what criteria are used for classification. The two major types of digital filters are: finite impulse response digital filters (FIR filters) and infinite impulse response digital filters (IIR).

#### 2.2.1.1 FIR Filter

FIR means Finite Impulse Response. In the common case, the impulse response is finite because there is no feedback in the FIR. A lack of feedback guarantees that the impulse response will be finite [3]. Therefore, the term "finite impulse response" is nearly synonymous with "no feedback". However, if feedback is employed yet the impulse response is finite, the filter still is a FIR.

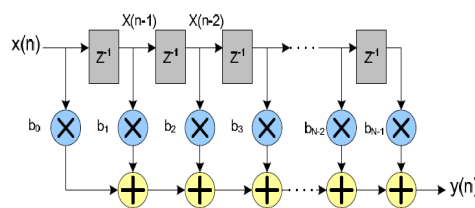


Figure 1: FIR filter

The difference function of FIR filter that defines how the input signal is related to the output signal is:

$$y[n] = x[n]b[0] + x[n-1]b[1] + x[n-2]b[2] + \dots + x[n-o-1]b[o-1]$$

where  $b[i]$  are coefficients of the filter,

$x[n]$  is an input signal,

$y[n]$  is an output signal ,

'o' is the order of the filter.

The output of a filter to an input response of  $x[n]$  is determined by the convolution function

$$y[n] = h[n] * x[n]$$

The key attributes of an FIR filter are as follows:

- FIR filters have no analog equivalent.
- FIR filters can only be implemented as an "all-zero" filter, since the FIR has no poles.
- FIR filters are therefore inherently stable, since their poles are located at the origin in the z-plane.

- FIR filters have a linear phase response if their filter taps are symmetrical about the center tap position. This results in zero phase distortion in the filter output.
- FIR filters can achieve very high roll-off rates when a large number of taps are used. More taps increase the steepness of the filter roll-off while also increasing calculation time (delay) and for high order filters, limiting bandwidth.
- FIR filters are insensitive to coefficient quantization errors since they are non-recursive.
- FIR filters have no limit cycle issues.
- FIR filters are easy to design and can be implemented using fractional coefficients, which makes fixed point calculations easier.
- FIR filter latency =  $(\text{Number of Taps} - 1) / 2$  sample periods.

### III. RELATED WORK

A multiplier is an interesting design example because it affords a rich set of design choices. We can build a multiplier that uses a large array of adders or an alternative that cycles a smaller array of adders several times to complete the product. We will use logical effort to seek a design with minimum delay for a given array configuration.

#### 3.1 Array Multiplier

Digital multiplication is the most extensively used operation especially in Signal processing, people who design digital signal processors sacrifice a lot of chip area in order to make the multiply as fast as possible.

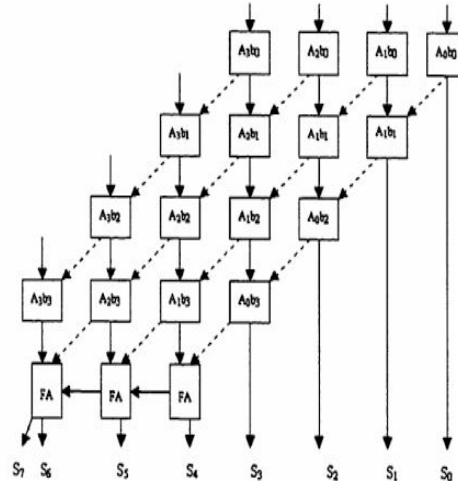


Figure 2: Array multiplier

The array consists of three rows of five adder cells each. Each row is responsible for adding a shifted form of the multiplicand to the partial product and passing the partial product to the row below[4]. Each adder cell contains a 1-bit full adder with three inputs of equal binary value, labeled  $a$ ,  $b$ , and  $c$ . Each cell has two outputs representing the sum ( $s$ ) and carry ( $cr$ ) that result from adding together the three inputs. The sum output represents the same binary weight as each of the three inputs. The carry output represents twice the binary weight of the sum output. As you can see from the figure, each cell combines a sum and a carry input from cells earlier in the array with a product bit. Note that the longest path through the array is three adder cells, corresponding to the number of rows in the array.

#### IV. EXPERIMENTAL OPERATION

Due to the explosive growth of the multimedia applications, the demand of the high performance and low power digital signal processing (DSP) is getting higher and higher. Finite Impulse Response (FIR) digital filters [5] are one of most widely used fundamental devices performance in DSP systems ranging from wireless to video and image processing. The three basic module performance in FIR filters are Adder, Delay element and Multiplier. An FIR filter is based on a feed forward difference, where feed forward means that there is no feedback of past or future outputs to form the present output. We have analysed that FIR filters are linear phase filters, and they have less delay compared to other filters. Fir filters are signal conditioners.

Finally the number of taps in an FIR filter [6] affects the sharpness of the filter. This can be easily understood by recognizing that the number of taps determines the length of time over which the filter operates, and sharp features in frequency require longer time intervals to represent. The 4-tap filter means the order of the filter is 4 and so it has four coefficients. We have defined the input as signed type of 8 bits wide. The output is also of signed type with 16 bit width. The design contains two files. One is the main file with all multiplications and adders defined in it, and another for defining the D flip flop operation. Code is synthesizable and with a few changes can be ported for a higher order filter. Similarly we have designed the 8-tap filter where it means the order of the filter is 8 so it has eight coefficients.

The design used for filter design has only vectors with positive coefficients alone because as in Xilinx software it is not advised to give negative numbers while assigning the input values. So we have designed the FIR filter using the array multiplier. Since array multipliers are very effective to use and also they provide a better performance compared to other dedicated multipliers.

#### V. RESULTS AND DICUSSION

The 4-tap FIR filter is implemented using D flip flops, adders and multipliers. The different combinations of adders and multipliers are used for the design of filter. The input data and multiplier coefficients are used as 8-bit unsigned representation.

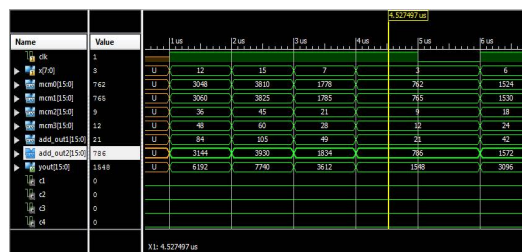


Figure 3: 4 tap FIR filter output

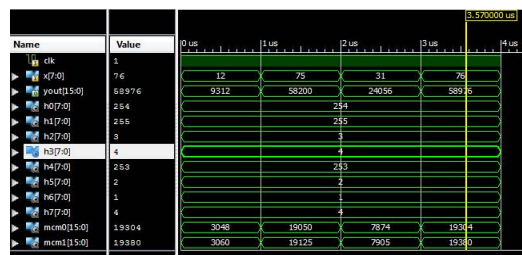


Figure 4: 8 tap FIR filter output

Order of filter	parameters	Filter design using Dedicated Multipliers	Filter design using Array Multipliers
4-tap FIR filter	4 input LUT's	10	10
	Delay	9.485ns	9.476ns
8-tap FIR filter	4 input LUT's	70	65
	Delay	13.194ns	12.926ns

Table 1: Comparison of FIR filter using Dedicated and Array Multiplier

## VI. CONCLUSION

The FIR filters are widely used in digital signal processing but for realizing the large order filters the speed, cost and flexibility is affected because of complex computations the design is suitable for applications that require low power consumption and less delay. The proposed paper consists of multipliers, where multipliers outweigh adders in cost. So it is profitable to replace dedicated multipliers with array multiplier. Moreover, the number of increased adders stays still when the length of FIR filter becomes large. The designed FIR filter is used in numerous DSP applications where accuracy is not the main issue. It can be used in speech and image processing fields as well, to increase battery life performance in the portable devices such as Laptop, smart-phones etc. For the implementation of N-tap FIR filter the top-down design methodology is used with optimized algorithm. Design of the generic 4-tap and 8-tap FIR filter in Xilinx 12.3 was successful.

## REFERENCES

1. Yu-Chi Tsao and Ken Choi, 2011. "Hardware-Efficient Parallel FIR Digital Filter Structures For Symmetric Convolutions" 978-1-4244-9474-3/11/\$26.00 ©2011 IEEE.
2. Pramod Kumar Meher. 2010. "New Approach to Look-Up-Table Design and Memory-Based Realization of FIR Digital Filter" IEEE Transactions on circuits and systems—irregular papers, Vol.57, No. 3, March.
3. Anurag Aggarwal, Astha Satija, Tushar Nagpal, "FIR filter designing using Xilinx system generator" International Journal of Computer Application, vol. 68, no.11, April. 2013.
4. N. Ravi1, A.Satish, .T.Jayachandra Prasad, T.Subba Rao, "A new design for array multiplier with trade off in power and area" International Journal of Computer Science Issues, vol.8, May 2011.
5. N. Parijatha, K.R.A. Hinduja and A.C. Shaker, "FPGA Optimized Low Power and High Speed FIR Filter Structures for DSP Applications", *International Journal of Engineering Research & Technology*, vol. 2, pp. 1- 5, Mar. 2013.
6. C. R. Baugh and B. A. Wooley. 1973. "A two's complement parallel array multiplication algorithm," IEEE Trans. Comput., Vol. C-22, pp. 1045–1047, December.