Designing and Comparison of FIR Filter using Xilinx System Generator

Ritu Saroha[1], Surender Dhiman[2]

[1]*M.TECH. Scholar, ECE,, B.M.I.E.T,* [2]*A .P. Department of EEE, B.M.I.E.T, Sonepat*

[1]ritusaroha2284@gmail.com

[2]suren001@gmail.com

***Abstract*—The developments in electronic technology are taking place at a very fast speed. DSP is one of the fields where developments are taking place at faster rate. Digital filtering is one of the most powerful tools of DSP. Digital filters are widely used in the world of communication and computation. On the other hand to design a digital filter that satisfies all the required conditions is a challenging one. Several techniques for designing digital filters have been available in the past many years. In this paper design and comparison of FIR filter using Xilinx ISE is the concerned. Digital filtering is one of the most powerful tools of DSP. Filters are a basic component of all signal processing and telecommunication systems. This research proposes the implementation and comparison of FIR filter using Xilinx System Generator software. Xilinx System generator is used to design efficient DSP algorithm on FPGA. Finite Impulse Response (FIR) filter is designed and the responses are compared with the responses obtained from MATLAB using Simulink in Xilinx System generator.**

Keywords— **DSP, FIR, FPGA, Simulink, Xilinx system generator**

I. INTRODUCTION

Digital signal processing is defined as the processing or manipulation of signals using digital techniques [1]. Main characteristics of DSP systems are specialized high speed arithmetic, Data transfer to and from the real world, multiple access memory architectures, High data bandwidth, Predictable program flow, Sensitivity to small numeric errors.

The field of DSP has always been driven by the advances in DSP applications and in scaled VLSI technologies [5]. Filter is a frequency selective circuit that allows a certain band of frequency to pass while attenuating the others frequencies.

A. DIGITAL FILTERS

A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. Based on the length of the impulse response, digital filters are classified into two types:

1) Finite impulse response (FIR) filters
2) Infinite impulse response (IIR) filters

Although FIR filters are more complex, they have certain advantages over IIR filters due to which they are more widely used in filtering applications. IIR filters do not provide stability at higher orders whereas the FIR counterparts are always stable and are particularly useful for applications which require exact linear phase response.

B. FPGA

FPGA arrived in 1984 as an alternative to programmable logic devices (PLD) and ASICs. FPGA offers the significant benefits of being readily programmable. FPGA can be programmed again and again, giving designers multiple opportunities to tweak their circuits. FPGA consists of an array of logic blocks that are configured using software. Programmable input/output blocks surround these logic blocks. Both are connected by programmable interconnects. Field programmable gate array has become an extremely cost effective means of off-loading computationally intensive digital signal processing algorithms to improve overall system performance.

II. FIR FILTER DESIGNING

FIR filters have finite impulse response and these filters are also called as non-recursive filter because they have no feedback. A non-recursive filter is a function only of the input signal. The response of such a filter to an impulse consists of a finite sequence of $M+1$ sample, where M is the filter order [3]. Hence, the filter is known as a Finite-Duration Impulse Response (FIR) filters.

FIR filter design essentially consists of two parts:

(1)Approximation problem

(2) Realization problem

There are essentially three well-known methods for FIR filter design namely:

(1) The window method

(2) The frequency sampling technique

(3)Optimal Filter Design Method

A .IMPLEMENTATION ISSUES IN DESIGNING OF FIR FILTER

There are essentially two steps in filter design process which completes the process of filter designing. First of all we have to generate coefficients for the filter to be designed, and then we simulate the filter with the help of coefficients generated.

(a) Determination of the coefficients a_k and b_k that produce a desired response

(b) Implementation of the digital filter given a set of coefficients $\{a_k, b_k\}$.

The following factors are used for comparing different filter structures:

Computational Complexity:

This is the number of arithmetic operations (multiplications, and additions) required to compute an output value y (m) for the system.

Memory Requirement:

This refers to the number of memory locations required to store the filter coefficients, past inputs, past outputs and any intermediate values.

B .IMPLEMENTATION METHODOLOGY

The following implementation methodology is identified. Various steps are as follows:

Step 1: Design the Filter

Step 2: Create Simulink Model.

Step 3: Implement Simulink model with the help of MATLAB.

Step 4: Implement Simulink model with the help of XILINX system generator

Step 5: Compare both models.

Step 6: Code generation using System generator.

Step 7: Simulate and debug the logic program and make necessary correction to design of Step 3.

III. DESIGNING ON SIMULINK

Simulink is a block diagram environment for multi-domain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. It provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB and enables us to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis. For the study we have to become familiar with the MATLAB and Simulink environments (software) [4, 5].

The 'Xilinx system generator' [6] is a high-level tool for designing high-performance DSP systems using FPGAs. The system generator tool enables us to integrate Xilinx with Simulink; it creates a .ise file which is used in Xilinx using the model file of Simulink.

'FDA tool' [4] is the basic tool of MATLAB used to design a filter of required specifications. There are different response types (High pass, Low pass, Band stop, Differentiator, Integrator etc.) and Design Methods (IIR, FIR) to implement the filter. These windows can be customized by providing order of the filter, cut-off, sampling, pass-band and stop-band frequencies and magnitude specifications. Through the specifications provided, the tool creates coefficients that are saved as matrix in MATLAB workspace

The major block set used in the design is 'FIR Compiler 5.0.' [4]. It implements a MAC and DA FIR filter as shown in Figure 4. It accepts a stream of input data and computes filtered output with a fixed delay, based on filter configuration. The filter specification tab enables us to provide the coefficient vector as a single MATLAB row vector directly through FDA tool from the workspace.

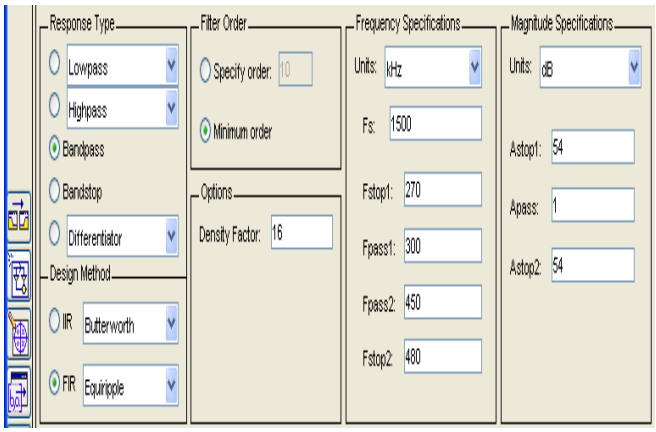

Figure1.Filter specifications

The 'Spectrum Scope/ B-FFT' [8] block computes and displays the mean-square spectrum or power spectral density of each signal. The signal can be a vector or a matrix. The Spectrum type parameter of the block is specified to be two-sided ((-Fs/2...Fs/2), here *Fs* is the sampling frequency of the original time-domain signal.

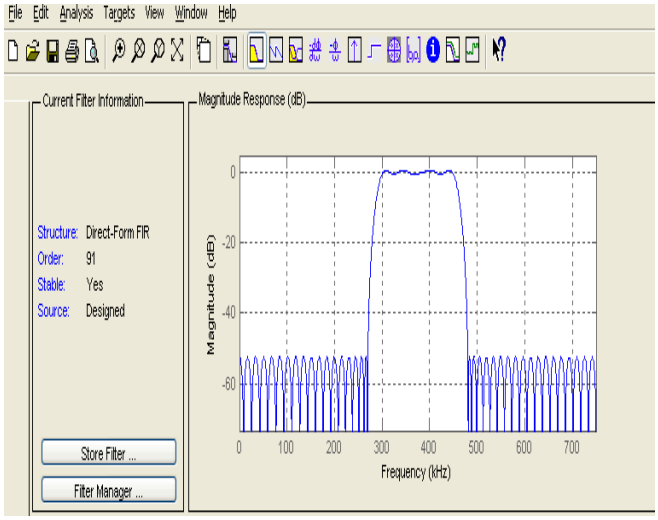

Figure2. Magnitude response of filter

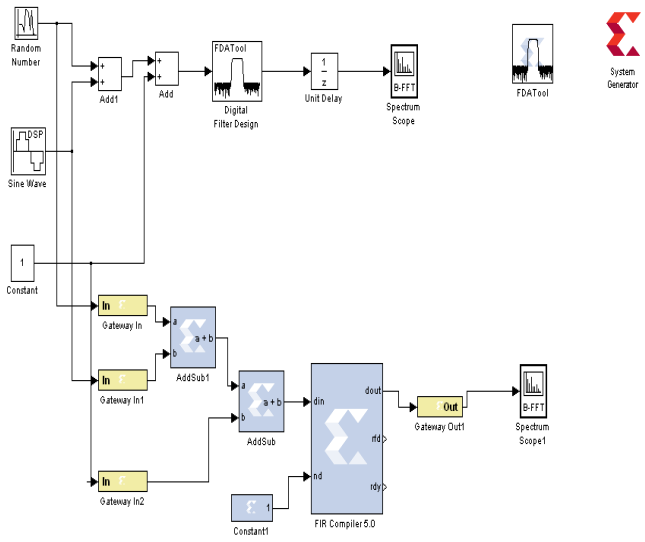

Figure3. Comparison of FIR filter using MATLAB and XILINX

IV. RESULTS

The following figure illustrates the simulation results of the FIR filter with MATLAB. The plot is between the magnitude and the frequency. This can also be called as the hypothetical implementation simulation results because the filter is simulated with the help of MATLAB. The numbers of ripples in the theoretical implementation are more than the practical implementation.

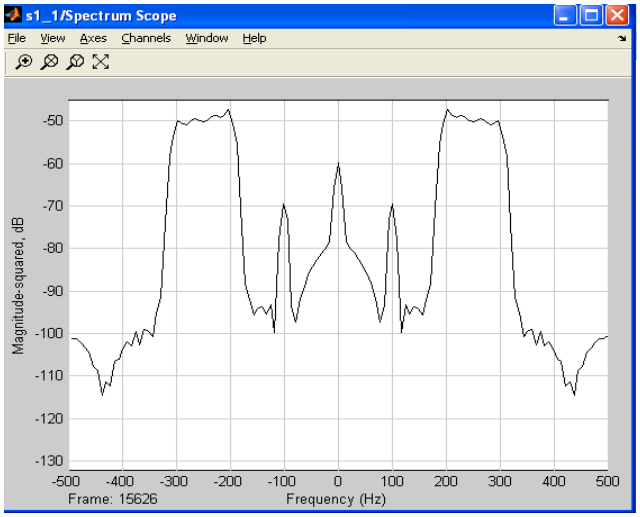

Figure 4. Simulation results using MATLAB

The following figure illustrates the simulation results of the FIR filter with Xilinx. The simulation results given below can be converted into the VHDL codes and then, can be fed directly into FPGA kits for the hardware implementation. Results of hypothetical and the realistic implementation are different and the difference of them gives the errors in the hypothetical implementation. So we can compare them also with the help of a subtractor. The output of both the models can be fed into the subtractor and the error would be available. The output of the subtractor would give the errors in the hypothetical implementation.

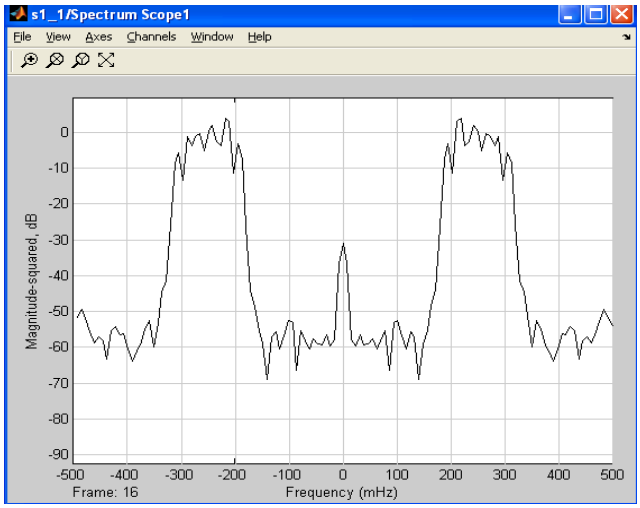

Figure 5. Simulation results using XILINX

V.CONCLUSIONS

The results show that with a single model it would be possible to find out the results from both the software, therefore practical and theoretical results would be available from a single model. Basically this technique will allow us to design digital filter using simulink/ modeling technique which is hardware implementable and much easier than any other coding technique.

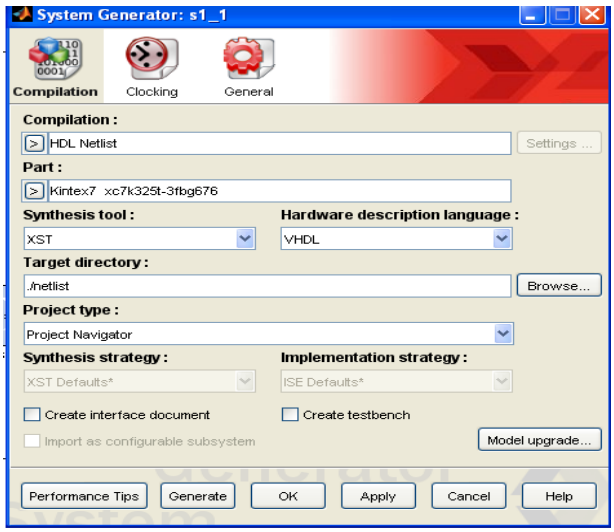

Figure 6. System generator setting

VI. FUTURE DIRECTIONS:

The codes generated from the system generator token can be fed into the FPGA kit for the hardware co-simulation. When the generation is successfully completed, a new Simulink library window will open up and compiled block with appropriate number of inputs and outputs will be displayed. Copy the compiled block and connect it in the design. Double-click on the hardware co-simulation block and perform following configurations and then click OK.

While FPGAs used to be selected for lower speed/ complexity/ volume designs in the past, today's FPGAs easily push the 500MHz performance barrier .With unprecedented logic density increases and a host of other features, such as embedded processors, DSP blocks, clocking, and high-speed serial at ever lower price points, FPGAs are a compelling proposition for almost any type of design.

REFERENCES

[1] http://www.dsptutor.freeuk.com/dfilt1.htm

[2] S.K.Mitra, "Digital signal processing", Tata McGraw Hills publications

[3] C. Ramesh Babu Durai," Digital signal processing", Laxmi publications, 2007(second edition)

[4]Mathworks,simulink,http://www.mathworks.com/products/simulink

[5] MATLAB "The language of technical computing", the Mathworks Inc., Natick, m.a. 1997

[6] Xilinx system generator, basic tutorial, www. Xilinx.com

[7] Anurag aagarwal, Aastha satya, Tushar nagpal," FIR filter designing using Xilinx system generator", International journal of computer applications, volume 68 no. 11, April 2013

[8] Harish V. Dixit, Dr. Vikas Gupta, V.C.E.T," IIR filter using Xilinx system generator for FPGA implementation"' International journal of engineering research and applications (IJERA) ISSN: 2248-9622, vol. 2, issue 5, Sept-Oct 2012, pp.303-307