IMPLEMENTATION OF FIR FILTER USING VHDL

[1]Raksha Jagadale, [2]Digvijaysingh Rathore, [3]Abhishek Satam, [4]Ajit Singh, [5]Ravindra Joshi.
[1,2,3,4] B.E Students, Electronics and Telecommunication, K.C. College of Engineering, & Management Studies & Research
Thane, India.
[5]Assistant Professor, Electronics and Telecommunication, K.C. College of Engineering, & Management Studies & Research
Thane, India.
raksha.jag13@gmail.com, dsr207@gmail.com, abhisatam50@gmail.com, ajit.m93@gmail.com, joshi_ravindra_s@ yahoo.co.in

Abstract- Digital filtering is one of the most important fundamental aspects of Digital Signal Processing. In signal processing, a finite impulse response (FIR) filter is a filter whose impulse response is of finite duration, because it settles to zero in finite time. A FIR filter using the window method is being implemented by us. We have used symmetric direct form structure and fixed point arithmetic for our design. Data length of 18 bits has been used by us. We will be implementing this design on a FPGA kit using Xilinx and VHDL. The verification of the design will be done using matlab.

Keywords- FIR filter, Xilinx, VHDL, FPGA, Digital Signal Processing

I. INTRODUCTION

Digital filter plays an important role in digital signal processing applications. Digital filters are widely used in digital signal processing applications, such as digital signal filtering, noise filtering, signal frequency analysis, speech and audio compression, biomedical signal processing, etc.

A digital filter is a system which passes some desired signals more than others to reduce or enhance certain aspects of that signal. It can be used to pass the signals according to the specified frequency pass-band and reject the frequency other than the pass-band specification.

The basic filter types can be classified into four categories: low-pass, high-pass, band-pass, and band-stop. On the basis of impulse response, there are two fundamental types of digital filters: Infinite Impulse Response (IIR) filters, and Finite Impulse Response (FIR) filters. [1]

Finite Impulse Response digital filter strictly has exact linear phase, highly stable, computationally intensive, less sensitive to finite word-length effects, arbitrary amplitude-frequency characteristic and real-time stable signal processing requirements etc. Thus, it is widely used in different digital signal processing applications.

There are many straight forward techniques for designing FIR digital filters to meet arbitrary frequency and phase response specifications, such as window design method or frequency sampling techniques. The Window method is the most popular and effective method because this method is simple, convenient, fast and easy to understand. The main advantage of this design technique is that the impulse response coefficient can be obtained in closed form without the need for solving complex optimization problems.

Window functions can be divided into two categories; Fixed and Adjustable window function [2]. Commonly used fixed window functions are: Rectangular window, Hanning window, hamming window and Blackman window. On the other hand, Kaiser Window is a kind of adjustable window function.

IJERT

II. DESIGN OF FIR FILTER

For designing, we are considering the following sample example with the given parameters to calculate the input sequence:

- Cut-off frequency = 5 KHz
- Pass band frequency = 1.5 KHz
- Stop band attenuation = 50 dB
- Sampling frequency = 8 KHz
- Transition width = 0.5 KHz

For these set of values we had obtained a particular value for the input sequence. That obtained input signal was used by us for our designing.

III. FIR FILTER DESIGN METHOD

In actual procedure for designing digital FIR filters, first the desired filter responses are characterized and the filter coefficient values are calculated for a causal FIR filter. There are different methods to find the coefficients of digital filter from frequency specifications. They are: [2]

- Fourier series method
- The window method
- Frequency sampling method
- Optimal filter design method

A simple and efficient way to design an FIR filter is window method. In the Window Design Method, the unit impulse response of ideal filter is obtained by applying inverse Fourier transform to the ideal frequency characteristics of digital filter. Then this unit sample response must be truncated at some point, this process is equivalent to multiplying it by a finite length window function. After truncation and windowing, an FFT is used to generate the corresponding frequency response of FIR filter. The frequency response can also be modified by choosing different window functions. Here we are using the window method to obtain order of filter as well as for further analysis.

IV. BASIC ENTITY

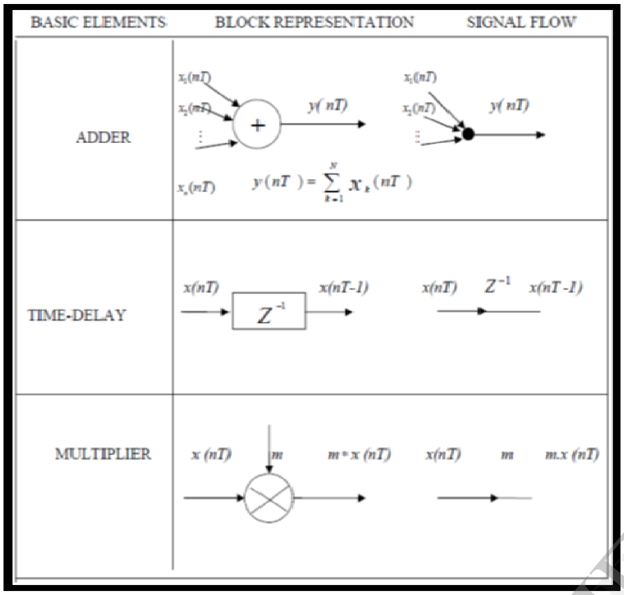

Fig.1: Entities used in our designing

[1][2] The basic entity blocks required are:

1. Fixed point adders
2. Fixed point subtractors
3. Fixed point multiplier
4. Time delay block

V. VARIOUS METHODS OF REALIZATION

The various methods which can be used to implement FIR filters using the basic blocks are as follows: [2]

(a) DIRECT FORM STRUCTURE

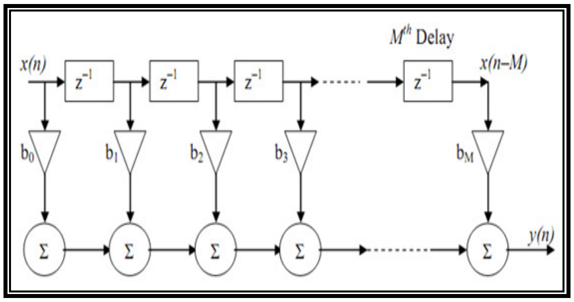

Fig.2 Direct form structure realization

(b) CASCADE STRUCTURE

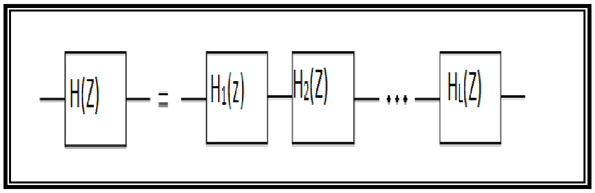

Fig.3 Cascade structure realization

(c) TRANSPOSE FORM STRUCTURE

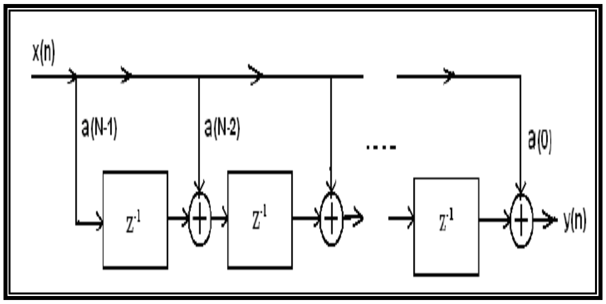

Fig.4 Transpose structure realization

(d) LATTICE STRUCTURE

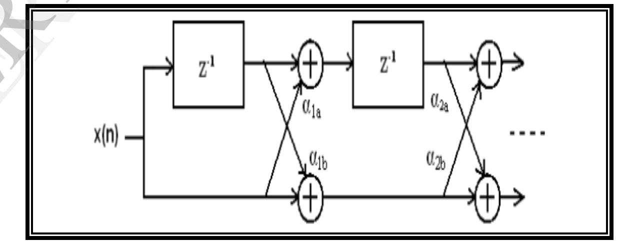

Fig.5 Lattice structure realization

IJERT

The simplest of these structures is the direct form realization structure.

However, there are other more practical structures that offer some distinct advantages, especially when quantization effects are taken into consideration.

The cascade, parallel, and lattice structures exhibit robustness in finite word-length implementations.

The frequency sampling has the advantage of being computationally efficient when compared with alternative FIR realizations.

Other filter structures are obtained by employing state-space formulation for linear time-invariant system.

Due to space limitations, state-space structures are not generally used. We have used the direct form symmetric structure for our design.

VI. IMPLEMENTATION

The arithmetic available for the designing purpose is of two types. They are: [8]

- Fixed point arithmetic
- Floating point arithmetic

Fixed point arithmetic is easy to design, requires less calculations whereas floating point arithmetic is more computationally demanding.
The accuracy of floating point is more than that of the fixed point but considering the complications in designing we are using the fixed point arithmetic.

In our designing the size of the input sequence is taken as 18 bits. The representation is as shown below:

S	OF	10 BIT DATA

Fig.6 Data format considered for our design

S = sign bit (1bit), when S = 0 the given number is positive else when S = 1 the given number is negative.
OF = overflow bits (7bits), these bits are used to store the values when result is more than 10 bits.

The input sequence is considered as the 10 bit data.The data is represented in fixed-point notation. In the fixed-point format, the numbers are usually assumed proper fraction.

VII. BLOCK DIAGRAM

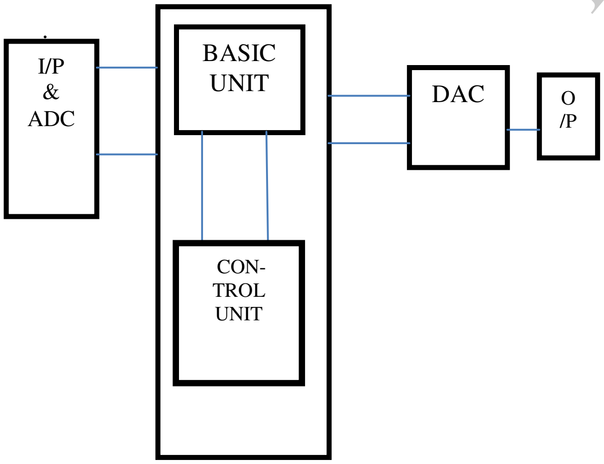

Fig.7 System block diagram used in our design

The above diagram shows the system architecture. Suppose a 10 bit input is applied to the system.

This input is sampled and made to pass through the ADC where the continuous time signal is converted into a discrete signal.

This discrete signal is applied as the input to the main block.

Some additional connections like that of the clkin and clkout are also applied to the main block.

Thus the input is synchronized with the clock.

The main block consists of the basic unit and the control unit where all the processing of the input signal is performed. [5]

The output of the main block is given to the DAC and at the output port of DAC we get our required output.

VIII. DEVICE UTILIZATION SUMMARY

TABLE I. SUMMARY REPORT

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	3851	7168	53%
Number of 4 input LUTs	5766	7168	80%
Number of occupied Slices	3593	3584	100%
Total Number of 4 input LUTs	5766	7168	80%

IJERT

IX. SYNTHESIS TOOLS AND THEIR USE IN DESIGNING

The following tools are used for implementation of FIR filter on FPGA: [3]

- XILINX ISE web pack 11.1i for design, synthesis and implementation.
- Very High Speed Integrated Circuit Hardware Description Language (VHDL) is used as the designing language.
- Hardware Description Languages (HDLs) are used to describe the behaviour and structure of system and circuit designs.

Web PACK ISE design software offers a complete design suite based on the Xilinx ISE series software

We have used XILINX for synthesis and implementation of our design.

Various specifications of the FPGA kit we are using is as follows: [9]

- Device: Xilinx FPGA (XC3S400 PQ208)
- On board +5V, +3.3V, +2.5V supply to FPGA & other hardware circuit.
- On board 2 Crystal 8MHz & 25MHz.
- Master Reset key for hardware reset
- Program Key for FPGA reconfiguration
- On board Flash EPROM for FPGA backup
- JTAG Interface (Boundary Scan)
- PROM Interface (XCF02S)
- 40 pin, 4 header connector for external I/O's
- Number of I/O's 136

X. CONCLUSION

The structure of FIR filter has been implemented on a FPGA kit. This paper mainly describes the design and simulation of FIR filter which is based on FPGA, Xilinx tools and VHDL.

By using these tools, the time required to get the desired results becomes very less. FIR filter coefficients design has been performed using the MATLAB. VHDL has been used to enter hardware description.

To test the correctness of the design, the observed output has been compared with the calculated output from the MATLAB.

This helps us in confirming the effectiveness of the design. VHDL codes have been written, synthesized, mapped successfully, configured and prototyped. The FIR filter designed fully meets the design requirements.

IJERT

REFERENCES

[1] Mitra,S.K., “Digital Signal Processing” 3rd Edition, Tata McGraw Hill Publications.
[2] Proakis, J.G., Manolakis, D.G., “Digital Signal Processing” 3rd Edition, PHI publication 2004.
[3] Spartan-3 Starter Kit Board User Guide UG130 (v1.1) May 13, 2005.
[4] Mark S. Manalo and AshkanAshrafi “Implementing Filters on FPGAs” Department of Electrical and Computer Engineering.
[5] Design of FIR filter using look up table and memory based approach by A.Sirisha, P.Balanagu & N.Suresh Babu , Department of Electronics and Communication Engineering, Chirala Engineering College, Chirala.
[6] Fir filter implementation on a FPGA with coefficients obtained using Remez exchange algorithm by Animesh Panda.
[7] Hardware Approach of a Multipurpose Finite Impulse Response Filter for Real-Time Filtering Applications Md. Syedul Amin, American Journal of Applied Sciences 8 (12): 1272-1281, 2011.
[8] FPGA implementation on digital FIR filter by Harsh Kumar, Thapar University.
[9] FPGA kit manual, Scientech Technologies Pvt.Ltd.
[10] Design of FIR filter using IP cores by Apurva Singh Chaauhan and Vipul Soni.