ISSN: 2454-9940

IJASEM

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940
www.ijasem.org
Vol 18, Issue 3, 2024

IMPLEMENTATION OF FIR FILTER USING VHDL

[1]PUSULURI JYOTHI, [2]DR M SATYA NARAYANA

[1]PG Scholar, **ECE,** HOLY MARY INSTITUTE OF TECHNOLOGY AND SCIENCE

[2]Professor, ECE, HOLY MARY INSTITUTE OF TECHNOLOGY AND SCIENCE

ABSTRACT-

One of the most crucial foundational elements of digital signal processing is digital filtering. Because it settles to zero in a finite amount of time, a filter with a finite impulse response (FIR) has a finite duration in signal processing.

We are going to implement a window-based FIR filter. For our design, we have employed fixed point arithmetic and symmetric direct form structure. We have utilized data with a length of 18 bits. Using Xilinx and VHDL, we will be putting this idea into practice on an FPGA kit. Matlab will be used for the design verification process.

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

1.INTRODUCTION

In digital signal processing applications, digital filters are crucial. Applications for digital signal processing, including digital signal filtering, noise filtering, signal frequency analysis, speech and audio compression, biomedical signal processing, etc., frequently use digital filters. A digital filter is a device that reduces or enhances specific characteristics of a signal by passing some desired signals more than others. It can be used to reject frequencies that do not fall within the given pass-band and pass signals according to the set frequency pass-band. Four categories can be used to group the basic filter types: band-pass, band-stop, high-pass, and low-pass. There are two primary categories of digital filters based on impulse response: finite impulse response (FIR) filters and infinite impulse response (IIR) filters. [1] The digital filter known as Finite Impulse Response is characterised by precise linear phase, high stability, high computational demand, reduced sensitivity to finite word-length effects, variable amplitude-frequency characteristic, and steady signal processing requirements in real-time. As a result, it finds extensive applicability in many digital signal processing applications. There are numerous simple methods, such the window design approach or frequency sampling techniques, for creating FIR digital filters that satisfy any frequency and phase response requirements.

Because it is straightforward, practical, quick, and simple to comprehend, the Window method is the most often used and successful approach. The primary benefit of this design method is that it does not require the solution of difficult optimization problems in order to acquire the impulse response coefficient in closed form. Fixed and adjustable window functions fall into two groups [2]. Rectangular, Hanning, Hammering, and Blackman windows are examples of fixed window functions that are frequently utilized. Kaiser Window, on the other hand, is a type of changeable window feature.

2. DESIGN OF FIR FILTER

In order to calculate the input sequence, we are taking into consideration the following sample example for designing, given the specified parameters:

• 5 KHz is the cut-off frequency. The stop band attenuation is 50 dB, while the pass band frequency is 1.5 KHz.
• 8 KHz is the sampling frequency; 0.5 KHz is the transition width.

We had determined a specific value for the input sequence for this collection of data.

We used the acquired input signal to inform our design.

3. FIR FILTER DESIGN METHOD

The real process for creating digital FIR filters begins with characterizing the intended filter responses and computing the filter coefficient values for a causal FIR filter. Several techniques exist for deriving the digital filter coefficients from frequency parameters. They are as follows: [2] The window method; the Fourier series method • The method of frequency sampling • The best way to design filters The window approach is an easy and effective way to create a FIR filter. By applying the inverse Fourier transform to the ideal frequency characteristics of the digital filter, the Window Design Method yields the unit impulse response of the ideal filter. At some point, the unit sample response needs to be shortened; multiplying it by a window function with a finite length is the equivalent of doing this. An FFT is used to produce the corresponding frequency response of the FIR filter after windowing and truncation. Altering the window functions can also change the frequency response. Here, the window approach is being used for both additional analysis and to determine the filter's order.

4. BASIC ENTITY

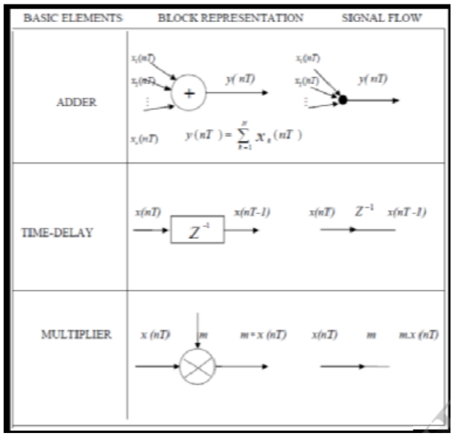

Fig.1: Entities used in our designing

[1][2] The necessary fundamental entity blocks are:

1. Static point addition
2. Perturbative subtraction
3. Fixed multiplier of points
4. The time delay block

5. 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

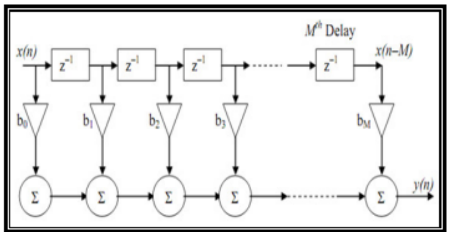

Fig.2 Direct form structure realization

(b) CASCADE STRUCTURE

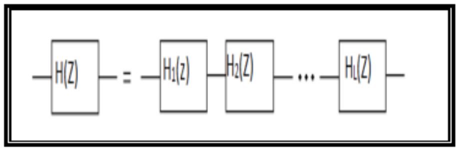

Fig.3 Cascade structure realization

(c) TRANSPOSE FORM STRUCTURE

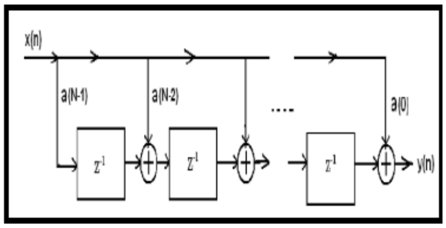

Fig.4 Transpose structure realization

(d) LATTICE STRUCTURE

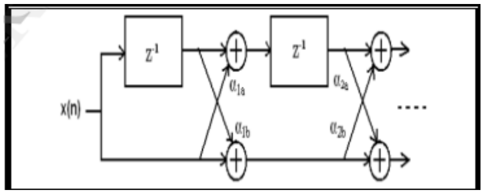

Fig.5 Lattice structure realization

The direct form realisation structure is the most basic of these structures. Nonetheless, alternative more useful structures provide some clear benefits, particularly when quantisation effects are taken into account. Finite word-length implementations demonstrate the robustness of the cascade, parallel, and lattice structures. When compared to other FIR realisations, frequency sampling has the advantage of being computationally efficient. Using the state-space formulation for a linear time-invariant system yields more filter structures. State-space structures are rarely employed because of space constraints. For our design, we have opted for the direct form symmetric structure.

6. IMPLEMENTATION

There are two kinds of arithmetic available for designing purposes. They are as follows: [8]

Arithmetic at fixed points and floating points

In contrast to floating point arithmetic, which needs more computations, fixed point arithmetic is simpler to construct. Although floating point arithmetic is more accurate than fixed point arithmetic due to design issues, fixed point arithmetic is still used. The input sequence is assumed to be 18 bits in size during the design process. The illustration is displayed as follows:

S	OF	10 BIT DATA

Fig.6 Data format considered for our design

S = sign bit (1 bit); the provided integer is positive when S = 0 and negative when S = 1. When the result is greater than 10 bits, the values are stored in the 7 bits known as OF (overflow bits). The 10 bit data is regarded as the input sequence. The fixed-point notation is used to represent the data. The numbers in the fixed-point format are typically taken to be appropriate fractions.

7. CONCLUSION

An FPGA kit has been utilised to implement the structure of the FIR filter. The design and simulation of the FIR filter, which is based on FPGA, Xilinx tools, and VHDL, are the primary topics of this work. It takes a lot less time to achieve the desired results when these tools are used. MATLAB has been used to design the FIR filter coefficients. To enter the hardware description, VHDL was utilised. The observed output and the computed output from MATLAB have been compared in order to assess the design's accuracy. This aids in verifying the efficacy of the design. It has been possible to write, synthesise, map, configure, and prototype VHDL codes. The FIR filter design satisfies all design specifications.

8. 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.