A Distributed Arithmetic (DA) Based Digital FIR Filter Realization

Mr. Mayur B. Kachare¹, Prof. D. U. Adokar²

¹ME Scholar (Digital Electronics), ²Associate Prof. in Electronics and Telecommunication Department
S.S.B.Ts C.O.E.T. Bambhori, Jalgaon, North Maharashtra University, Jalgaon-425001, Maharashtra (India)

Abstract— The Digital filters like Finite Impulse Response (FIR) filter is a widely used in digital signal processing applications in various fields like imaging, communications, instrumentation, etc. Programmable Digital Signal Processors (PDSPs) are used in realizing the FIR filter. Although, in realizing a large-order filter many complex computations are needed which influence the performance of the common digital signal processors in terms of speed, cost, flexibility, etc. Field-Programmable Gate Array (FPGA) has become an especially productive in relation to cost means of off-loading computationally intensive digital signal processing algorithms to enhance overall system performance. The FIR filter implementation in FPGA, utilizing the dedicated hardware schemes can more accurately achieve Application Specific Integrated Circuit (ASIC) like performance while reducing risks and development time cost. In this paper, a band stop FIR filter is implemented on FPGA. Direct-form way in realizing a digital filter is taken into account. This point of view gives a better performance than the common filter structures in terms of power consumption, speed of operation, and cost in real-time. The FIR filter is realized in FPGA and simulated with the help of Xilinx and MATLAB.

Keywords— Distributed Arithmetic (DA), Finite Impulse Response (FIR), Least Mean Square (LMS), Look-Up Table (LUT), Multiply-Accumulate (MAC).

I. INTRODUCTION

In all signal processing and telecommunication systems filters are a basic component. Filters are widely employed in communication systems and signal processing applications such as biomedical signal processing, channel equalization, noise reduction, audio processing, radar, video processing, and analysis of economic and financial data. For example in a radio receiver tuners, or band-pass filters, are used to extract the signals from a radio channel. A digital filter is a mathematical algorithm implemented in a hardware/software that operates on a digital input to provide a digital output. Digital filters provide a major role in DSP. Compared with analog filters they are preferred in number of applications like image processing, speech processing, data compression, etc. Digital filters are divided into two categories, including Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). FIR filters are widely applied to a variety of digital signal processing areas for the integrity of providing linear phase and system stability. However, the famous multiple constant multiplications based technique [7]; that is widely used for the implementation of FIR filters, cannot be used when the filter coefficients are changing dynamically. In another side, a general multiplier based design requires a large chip area and at the same time enforces a limitation on the maximum possible order of the filter which is designed for applications requires high throughput. A distributed arithmetic (DA) based technique [8] has gained considerable importance and popularity in past few years for its increased regularity and high throughput processing capability, which results in profitable and time efficient computing also area saving structures. The main operations required for DA based computation are a sequence of lookup table (LUT) just before shift register accumulation operations of the LUT output. The conventional DA used for an FIR filter design assumes that impulse response coefficients are fixed, and this behavior tends to use ROM-based LUTs. The implementation of FIR filters which are DA based memory requirement exponentially increases with the filter order. To eliminate the problem of such a large memory requirement, systolic decomposition techniques are used for implementation of long length DA based convolutions and FIR filter of large orders [7, 8]. The performance is increased as the filter size increases.

II. BACKGROUND

A digital processor is used to perform numerical calculations on sampled values of the signal in digital filter. The processor may be a specialised DSP (Digital Signal Processor) chip or a general purpose computer such as a PC. Digital filters are required in a large variety of signal processing applications, such as spectrum analysis, and pattern recognition, and digital image processing. Digital filters removes a number of problems associated with their classical analog counterparts and thus are mostly used in place of analog filters. The analog input signal must first be sampled and digitised with the help of an ADC (analog to digital converter). The output binary numbers representing successive sampled values of the input signal are entered to the processor that carries out numerical calculations on them. Fast DSP processors can control complex combinations in cascade (series) or parallel of filters, making the hardware requirements in comparison with the equivalent analog ciruity are relatively simple and compact.
Fig. 1 Block diagram of Digital Filter [10]

Ideal filters give a specified frequency range in regard to pass through while attenuating a specified unwanted frequency range. The filters are classified according to their frequency range characteristics.

Fig. 2 Ideal Frequency Characteristics [9]

Fig. 3 Pass Band and Stop Band [9]

In Fig. 2, the filters exhibit the following behaviour:
- The low pass filter passes all frequencies below \( f_c \).
- The high pass filter passes all frequencies above \( f_c \).
- The band pass filter passes all frequencies between \( f_{c1} \) and \( f_{c2} \).
- The band stop filter attenuates all frequencies between \( f_{c1} \) and \( f_{c2} \).

The frequency points \( f_c, f_{c1}, \) and \( f_{c2} \) specify the cut off frequencies for the different filters. When designing filters, you must specify the cut off frequencies. The pass band of the filter is the frequency range that passes through the filter. We know an ideal filter has a gain of one (0 dB) in the pass band so the amplitude of the signal neither increases nor decreases. In practical applications, ideal filters are not realizable. Ideally, a filter has a unit gain (0 dB) in the pass band and a gain of zero (\( -\infty \) dB) in the stop band. However, real filters cannot fulfill all the criteria of an ideal filter. Practically, a finite transition band always exists between the pass band and the stop band. The gain of the filter changes slowly from one (0 dB) in the pass band and to zero (\( -\infty \) dB) in the stop band [9].

A. Non Recursive or FIR Filters

A non-recursive filter has no feedback and its input-output relation is given by
\[
y(m) = \sum_{k=0}^{M} b_k x(m - k) \quad (1)
\]

As shown in Fig 4 the output \( y(m) \) of a non-recursive filter is a function only of the input signal \( x(m) \). The response of such a filter to an impulse consists of a finite sequence of \( M+1 \) sample, where \( M \) is the filter order. Hence, the filter is known as a *Finite-Duration Impulse Response (FIR)* filters.

Fig. 4 Direct-form Finite Impulse Response (FIR) filter [8]

Other names for a non-recursive filter include all-zero filter, feed-forward filter or moving average (MA) filter a term usually used in statistical signal processing literature.

B. Recursive or IIR Filters

A recursive filter has feedback from output to input, and in general its output is a function of the previous output samples and the present and past input samples as described by the following equation.
\[
y(m) = \sum_{k=0}^{N} a_k y(m - k) + \sum_{k=0}^{M} b_k x(m - k) \quad (2)
\]

Fig. 5 shows a direct form implementation of Eq. 2. In theory, when a recursive filter is excited by an impulse, the output persists forever. Thus a recursive filter is also known as an *Infinite Duration Impulse Response (IIR)* filters. Other names for an IIR filter include feedback filters, pole-zero filters and auto-regressive-moving-average (ARMA) filter a term usually used in statistical signal processing literature.
III. IMPLEMENTATION

There has been remarkable progress in recent software tool development to support DSP applications in FPGAs kits. System Generator is a design tool for Xilinx FPGAs which extends the capabilities of Simulink in high-level to include bit and cycle accurate modelling of FPGA circuits, and generation of an FPGA circuit from a Simulink model [5, 6]. System Generator provides stronger Simulink libraries for memories, arithmetic and logic functions, and DSP functions. Also by supporting high level modelling and automatic code generation, System Generator creates new opportunities to determine the interplay between hardware-centric considerations and mathematical abstraction.

The entire design as realized in Simulink is shown in Fig. 6.

A. System Generator

The System Generator block has control of system and simulation parameters, and is used to assist the code generator. Every Simulink model having any element from the Xilinx blockset must have at least one System Generator block. When a System Generator block is added to a model, it is easy and possible to find how code generation and simulation can be done.

B. Gateway In

The inputs to the Xilinx section of the Simulink design are the Xilinx Gateway In blocks. These blocks convert Simulink double, fixed-point and integer data types into the System Generator fixed-point type. The HDL design generated by System Generator is defined by every block of a top-level input port in.

C. MATLAB Simulink

The design options in MATLAB allow the user to either create a code for designing filters that calls built-in functions, or to design filters in Sptool, a graphical user interface. MATLAB provides all the information necessary for building a hardware replica of the filter designed in software. Simulink is a data flow graphical programming language tool for modeling, simulating and analyzing multi-domain dynamic systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in control theory and digital signal processing for multi-domain simulation and model-based design. Filter Information is as shown below in figure 7.

D. FDATool

The Xilinx FDATool block is MATLAB signal processing toolbox and it provides an interface to the FDATool software available. If the signal processing toolbox is not there then this block does not function properly and is of no use. This block gives a function of defining a FDATool object and stores it as a data of a System Generator model. FDATool gives a graphical user interface to digital filters. FDATool is very important tool in filter design.

E. Gateway Out

The outputs from the Xilinx section of the Simulink design are the Xilinx Gateway Out blocks. This block converts the System Generator fixed-point data type into Simulink Double.
F. Multiplexer

The Xilinx Mux block implements a multiplexer. The block has one select input (type unsigned) and a user-configurable number of data bus inputs, ranging from 2 to 1024.

G. Distributed Arithmetic FIR Filter v9.0 [15]

- High-performance finite impulse response (FIR), half-band, Hilbert transform, interpolated filters, polyphase decimator, polyphase interpolator, half-band decimator and half-band interpolator implementations
- 2 to 1024 taps
- 1- to 32-bit input data precision
- 1- to 32-bit coefficient precision
- Support for interpolation and decimation factors of between 1 and 8 inclusive
- Coefficient symmetry exploited (symmetric/negative-symmetric) to produce compact implementations
- Serial and parallel filters supported. The user may specify the degree of parallelism and trade off FPGA logic resources for sample rate in order to generate an optimal design
- On-line coefficient reload capability
- Incorporates Xilinx Smart-IP™ technology for maximum performance

To be used with v7.1i or later of the Xilinx CORE Generator™ system

IV. RESULTS

The specifications of a 50 order FIR filter in which attenuation obtain is below 40dB are shown in Fig. 8. The magnitude (dB) and phase response are shown in combined view in Fig. 9.

The comparison of given input audio and output audio through FIR filter is shown in Fig. 10. The input audio is a combination of signal with a noise of 2 KHz signal and it’s marked by .wav file. The output audio is the noise free .wav signal.

Fig. 8 Designed Finite Impulse Response (FIR) Filter Specification

Fig. 9 Magnitude and Phase response of Designed Finite Impulse Response (FIR) Filter

Fig. 10 Comparison of Input audio & Output audio for Designed Finite Impulse Response (FIR) Filter

V. CONCLUSION

This FIR filter is implemented using MATLAB Simulink model and Xilinx System Generator for selected audio application. The input audio signal is noisy signal. It is filtered using FIR compiler block that is designed using FDATool according to the sampling rate, passband frequency, stopband frequency, attenuations, transition width and filter orders. Finally noiseless signal is obtained at output of filter. The proposed FIR filter structure is verified using Xilinx system generator model and their performances are verified in terms of noise cancellation.

Author’s Profile:

Mayur B. Kachare has received his B.E. graduation degree in Electronics and Communication in 2012 and now pursuing M.E. degree in Digital Electronics from SSBTs COET Bambhori, Jalgaon. He has participated in conference of special issue of IJECSCE. He has published three papers in International Journals.

Dineshkumar U. Adokar has received his B.E. degree in Electronics from Visvesvarayya Regional College of Engineering,
Nagpur in 1987 and Master’s Degree in Electronics from Government College of Engineering, Amravati in 2001. He is also pursuing PhD from Shri Sant Gadge Baba Amravati University, Amravati. Presently, he is working as Associate Professor in the Department of Electronics and Telecommunication Engineering at SSBT’s COET Bambhori, Jalgaon. He has published 10 research papers in National and International Journals. His interests include Image processing, Microcontrollers etc.

REFERENCES


