Optimized FIR Filter Using Distributed Parallel Architecture for Audio Application

Prathibha P Nair¹, Tintu Mary John²
¹PG Scholar, Department of Electronics and Communication Engineering, Believers Church Caarmel Engineering College, R-Perunad, Pathanamthitta, Kerala, India
²Assistant Professor, Department of Electronics and Communication Engineering, Believers Church Caarmel Engineering College, R-Perunad, Pathanamthitta, Kerala, India
¹prathibha080@gmail.com, ²tintuajin@hotmail.com

Abstract

Digital Signal Processing is an important area having a wide range of applications with Digital Filters as the key element. Different architectures can be used for implementing FIR filters, Distributed Arithmetic is one such architecture. This is a multiplier less architecture, since multipliers are the speed limiting elements in a VLSI circuit. Thus it consumes less area than basic FIR filter by replacing multipliers with shift and add operation. Multimedia requires signal processing to be fast with less power consumption. Audio signal processing is an important application in multimedia. In order to perform this filtering function efficiently, we can optimize the delay, power and area, for that, Distributed Parallel Architecture is used. In this paper an FIR filter is designed using the Filter Design and Analysis (FDA) tool in MATLAB and the filter coefficients are calculated in 16-bit fixed point format. The input values of the audio signal is generated to HDL code by programming in MATLAB R2015a. The filter is implemented for Audio Application using basic filter structure, serial Distributed Arithmetic and parallel Distributed Arithmetic Architecture in Xilinx ISE 14.7 by programming in VHDL and the speed, area and power of these filters are compared. 40% increase in speed is obtained in Parallel Distributed Arithmetic FIR filter compared to the serial structure and when compared to the basic Direct form structure, there is 70% speed improvement.

Keywords- Digital Signal Processing, FIR filter, Distributed Arithmetic, Distributed Parallel filter

1. Introduction

Signal is a function of independent variable that contains information. Signal processing is a method to extract information from a signal. There is Analog Signal Processing and Digital Signal Processing. Analog Signal Processing has both input and output as continuous signals and in Digital Signal Processing, input and output are discrete signals. Digital Signal Processing (DSP) has a wide range of applications. Audio signal processing is an important area in multimedia. We have to remove noise from the audio signals. Filter is a frequency selective network, which modify an input signal in order to facilitate further processing. Filter (here) is an algorithm or a device which removes parts of a signal. A filter selects, suppresses or modifies certain frequency components of the signal, either to reduce noise or to shape the spectrum. There are mainly two types of filters- Analog filter and Digital filter. Analog filter operates on continuous signal or voltages whereas digital filter operates on discrete signals or numbers. Digital filters have better Signal to Noise Ratio (SNR), better signal reproducibility and it performs noiseless mathematical operations than analog filter. Thus digital filters are preferred over analog filters. Finite Impulse Response (FIR) filter and Infinite Impulse Response (IIR) filter are two common types of digital filters. Since impulse response of FIR filter settles to zero in finite time, it is called non recursive filter. IIR filters are recursive filter having internal feedback and may continue to respond indefinitely. FIR filter can be used to implement almost any sort of frequency response digitally and is implemented with delays, multipliers and adders. The general equation of an FIR filter is shown in (1).

\[ y(n) = \sum_{k=0}^{N-1} b_k x(n-k) \]  

Where y(n) is the output signal, x(n) is the input signal, \( b_k \) represents filter coefficient and N is the order of the filter. Digital Filter is to be designed for removing noise. Designing Digital Filter is the process of calculating appropriate filter coefficients and order of the Digital Filter. The designed filter can be implemented on Field Programmable Gate Array (FPGA). This implementation must meet the sampling rates of the corrupted audio signal. Digital Filter coefficients can be represented by fixed and floating point formats. Fixed point implementations have higher speed and lower cost, while floating point implementations have higher dynamic range and no
need of scaling which may be attractive for more complicated algorithms. The research is going on the optimized digital filter in terms of power, area and speed. There are various digital filter architectural optimization approaches like pipelining [5], parallel processing, Distributed Arithmetic [7], folding etc. used to generate efficient digital filter architecture.

In this paper, an FIR filter is designed for audio application using the distributed arithmetic architecture, both serial and parallel. Here the input signal is the audio signal, whose values are obtained by coding in MATLAB R2015a and these values are passed to the VHDL code. The filter coefficients are obtained using the Filter Design and Analysis (FDA) tool in MATLAB. An Equiripple Bandpass FIR filter [8] is designed here. The FIR filter for audio application using Distributed Arithmetic (DA) is obtained by programming in VHDL in Xilinx ISE14.7 design suite. Final synthesis is done for Spartan6-100T FPGA board for optimal utilization of FPGA resources. The delay, area and power is obtained for serial and parallel distributed filters [10]. These values are also compared with the basic traditional FIR filter.

2. Methodology

Finite Impulse Response (FIR) filters are one of the two main type of filters available for signal processing. Output of this filter is finite and it settles down to zero after some time. An Equiripple FIR Bandpass Filter is designed using Filter Design and Analysis (FDA) tool in MATLAB. The Figure 1 shows the designed filter.

Then the filter coefficients are calculated in 16-bit binary fixed point format. The calculated values are generated to HDL and then stored in Look Up Table in the Distributed Arithmetic Filter structure. The input audio values are taken as samples from the audio signal and stored in a text file. These are converted to ASCII and used in processing in the VHDL code to get the output response. After that, a basic FIR digital filter is implemented using the direct form representation using the shift and add method. The Fig. 2 shows the direct form representation of FIR digital filter.

![Figure 2. Direct form representation of an M-tap filter](image)

Figure 2 shows an M tap FIR filter which has N coefficients, \( x(n) \) is the input signal and \( y(n) \) is the output response. Next we have to implement the serial and parallel distributed arithmetic architecture in the design of Filter. Distributed Arithmetic is one of the most important methods of implementing FIR filters. Distributed Arithmetic (DA) is used for the calculation of inner product or multiply and accumulate (MAC) efficiently when the coefficients are pre knowledge, as happens in FIR filters. It is the bit level rearrangement of the multiply and accumulate operation. DA is appropriate when the number of elements in a vector is almost same as the word size. In DA, the explicit process of multiplication is replaced with ROM-LUT. Thus we can efficiently implement an FIR filter on Field Programmable Gate Array (FPGA) [3] [6]. The Figure 3 is the basic block diagram for FIR filter structure using Distributed Arithmetic.

![Figure 3. FIR filter structure using Distributed Arithmetic](image)

It includes mainly three blocks- Parallel and Serial Shift Registers, Look up table memory and right shift accumulator. Here \( x(n) \) is the input signal and \( y(n) \) is the output response. The output response of linear,
time-invariant filter at any discrete time is given by Equation 2 [9].

\[ y = \sum_{k=1}^{K} A_k X_k \]  

(2)

Let \( X_k \) be N-bits scaled 2’s complement number i.e. \( X_k < 1 \) and \( X_k = \{ b_{k0}, b_{k1}, b_{k2}, \ldots, b_{k(N-1)} \} \) where \( b_{k0} \) represents sign bit. Thus can be expressed in Equation 3.

\[ X_k = -b_{k0} \sum_{n=1}^{N-1} b_{kn} \cdot 2^{-n} \]  

(3)

Substituting Eq. 3 in Eq. 2 results in

\[ y = \sum_{k=1}^{K} A_k \left[ -b_{k0} \sum_{n=1}^{N-1} b_{kn} \cdot 2^{-n} \right] \]  

(4)

Solving and reordering, we get the final equation as

\[ y = -\sum_{k=1}^{K} A_k \cdot (b_{k0}) + \sum_{n=1}^{N-1} \left[ \sum_{k=1}^{K} A_k \cdot b_{kn} \right] \cdot 2^{-n} \]  

(5)

Therefore, by interchanging the summing order of \( k \) and \( n \), the initial multiplications in (2) are now distributed to another computational pattern as in (5).

Distributed Arithmetic is used to design bit-level architectures for vector-vector multiplications. In distributed arithmetic, each word in the vectors is represented as a binary number, the multiplications are reordered and mixed such that the arithmetic becomes “distributed” through the structure. Distributed arithmetic is commonly used for implementation of convolution operations and discrete cosine transforms (DCT).

2.1 Serial Distributed Arithmetic Architecture

The FIR structure consists of a series of multiplication and addition (MAC) blocks, which are expensive in high speed systems. Compared with traditional direct arithmetic, Distributed Arithmetic is a different approach for implementing digital filters. The basic idea is to replace multiplications and additions by a table and shift accumulators, which can save considerable hardware resources. Another advantage of this method is that it can avoid the decrease of system speed while the width of input data bit or the filter coefficient bit increases, which may occur in traditional direct method. Distributed Arithmetic relies on the fact that the filter coefficients are known. This is an important difference and a prerequisite for DA design. Figure 4 shows the structure of FIR Filter using Serial Distributed Architecture.

![Figure 4. Serial Distributed Arithmetic FIR filter](image)

Here, the shift register is made as a cascade of flip flops for storing digital data, sharing the same clock and the output of the last flip-flop is connected to the “data” input of the next one, which creates a circuit containing a one dimensional bit array that shifts by one position. Thus it shifts in the data present at its input and shifts out the last bit in the array. All flip-flops are set or reset simultaneously. The storage capacity of a register is determined by the number of stages in the register. The serial in/serial out shift register accepts data serially and generates the stored information on its output also in serial form. Based on this input value, the corresponding filter coefficient which is stored in the ROM look up table is selected and accumulated in the register to get the final output response.

2.2 Distributed Parallel Architecture

In order to optimize the various parameters like Power, Area and Delay, here we use Distributed Parallel Architecture to design an FIR filter. Based on the serial Distributed Arithmetic, for a length \( N \)th sum-of-product computation, accepts one bit from each of \( N \) words. If more bits per word are accepted, then the computational speed can be improved considerably. The maximum speed can be achieved with distributed parallel architecture. Parallel Distributed Arithmetic structure is shown in Figure 5. By using the parallel architecture in Distributed Arithmetic, we can reduce the factors like power, area and speed to a significant amount.
Parallel Distributed Arithmetic allows each of the input data to be addressed at the same time by increasing numbers of ROM, registers and adders, since each level of the addition operation is parallel. The difference between the two structures is that at each clock cycle, all input variables in the serial mode are just handled one bit, while the parallel mode handles all bits, which respectively represent two optimal results of resource and speed optimization design. All bits of the first input signal are applied to the first LUT. Similarly, all input signals are applied to the LUTs at the same time. The input signal got multiplied with the precomputed and stored coefficients in the LUT. These are accumulated in the register as in Serial structure. Finally, the output response is obtained

After designing FIR Filter based on both Serial and Parallel Distributed Arithmetic Architecture, the delay, area and power is calculated and compared.

3. Results and Discussions

An FIR filter for audio application is designed and synthesized using both serial and parallel Distributed Arithmetic in Xilinx ISE 14.7 Design Suite and the programming is done in VHDL for a Spartan 6-100T FPGA as target device. The performance analysis and comparison of the two architectures is performed and the parameters like area, power and delay is calculated and compared for both structures. Also these two architectures are compared with the basic FIR filter architecture which uses MAC units. The output waveform for basic FIR filter is shown in Figure 6.

Here the input signal is the “firin” signal which is 15 bits wide. These input values are obtained from the audio samples through MATLAB coding. These inputs are multiplied with the precomputed filter coefficients which are also of same width to get the filter’s output response “firout” which is 32 bits wide. Thus filtering function is performed on the input signal.

The filter is then synthesized on Spartan6-100T FPGA device to get the delay, area and power.
The comparison of the three FIR filter structure namely the basic FIR filter structure, the serial Distributed Arithmetic structure and the parallel Distributed Arithmetic structure for delay, area and power distribution is shown in Table 1. Basic FIR filter has very high delay. As we move to Serial Distributed Arithmetic Architecture, 50% reduction in delay compared to basic FIR filter is observed. In Parallel Distributed Arithmetic structure, more than 40% reduction in delay is obtained compared to serial architecture. Thus Parallel Architecture can be used for high speed application. Area or device utilization is less and almost same in basic and Serial Distributed Arithmetic architecture. In Parallel Distributed Arithmetic Architecture also it is almost same, but little more than serial structure. Power Distribution is same in all these architectures. Thus we can say that FIR Filter using parallel DA architecture is suitable for high speed application.

4. Conclusions
An FIR filter for audio application is designed using basic traditional shift and add method of filter structure, serial distributed arithmetic architecture and parallel distributed arithmetic architecture. These are multiplier less architectures. Basic filter structure uses shift and add method instead of multiplication. Distributed Arithmetic uses look up tables, shift registers and scaling accumulators and thus consumes less resources. Distributed Arithmetic architecture can be used for high speed implementation of FIR filter. Since the Distributed Arithmetic architecture is basically bit serial in nature, we can further increase the speed by using a distributed parallel architecture with almost the same area and power consumption. From the work, it is found that speed of the filter increased in Parallel Architecture compared to the serial architecture and basic structure. Area remains almost same in all architectures, but parallel architecture has a much higher area compared to others. The Power remains the same in all the architectures. Audio filtering in DSP processors must be done fast. So, Parallel Distributed Arithmetic architecture of FIR filter can be used in these high speed applications.

5. Acknowledgement
I would like to express my gratitude to my Professor, Tintu Mary John, whose knowledge and assistance added considerably to my graduate experience. I would like to thank the Dean of Research and Development Dept. of my college, Dr. Milind Thomas, for the motivation he provided me in my project.

6. References