A REVIEW ON FPGA IMPLEMENTATION OF RECONFIGURABLE DIGITAL FIR FILTER

1PRADNYA D. SHAHARE, 2SAMRAT S. THORAT

1M. Tech.(Electronic system and communication), Department of Electronics and Telecommunication Engineering, Government College of Engineering, Amravati, MH
2Assistant Professor, Department of Electronics and Telecommunication Engineering, Government College of Engineering, Amravati,MH
E-mail: 1pradnyashahare29@gmail.com, 2samratthorat@gmail.com

Abstract—This paper presents, the different methods viz. conversion based approaches and memory based methods that used for efficient implementation of FIR filter. The formulation of distributed arithmetic (DA) is also discussed. The distributed arithmetic is an efficient technique of FIR filter implementation in terms of area. By using Look Up Table (LUT), shift registers and scaling accumulator the DA based techniques design. Our ultimate goal is to minimize the parameters namely, area and power, with the reduction of hardware in terms of multipliers. The literature review of FIR filter by using DA and other technique is discussed.

Index Terms—Distributed Arithmetic, RAM based LUT; Conversion based approach, FIR Filter, Field Programmable Gate Array, MAC.

I. INTRODUCTION

In various signal processing, image processing and communication system applications low-complexity and high-speed digital finite impulse response (FIR) filter is commonly used as a basic tool because of less area, low cost, low power and high speed of operation and their absolute stability and linear phase property. The implementation cost and power consumption are also high because of complexity of computation. The Finite Impulse Response (FIR) filters are a class of digital filter and a process that removes unwanted parts of the signal, such as random noise, some undesirable component or features from a signal and attenuates or boosts the regions of a sound spectrum. FIR filters have finite impulse response and are used in applications like noise reduction, image enhancement, channel equalization, echo cancellation, speech and waveform synthesis etc. The disadvantage is that to process a signal, the number of computations is more, it requires high order and it causes requirement of more hardware, area and power consumption. As the filter order increases, the complexity of implementation raises with the accuracy of computation, so the realization of these filters in real-time with desired level of accuracy is required [10].

By convolving the input data samples with the desired unit response of the filter, a FIR filtering can be achieved [17]. Filters are frequently used in electronic systems to highlight signals in definite frequency ranges and removes signals in other frequency ranges. Real-time implementation of filters of large orders is a challenging task, the filter order increases therefore the complexity of FIR filter implementation increases as the number of multiply-accumulate (MAC) operations required per filter output also increases. The ultimate goal of the implementation of an efficient FIR filter is to reduce the parameters namely hardware, area and power by the reduction of arithmetic in terms of multipliers and hence DA algorithm can be used for implementation of high order FIR filter. In a Dolby system post filtering and pre-filtering are used to reduce the noise effect. In hi-fi audio a compensating filter may be involved in the preamplifier to compensate for the non-ideal frequency response characteristics of the speaker [18].

Field programmable gate array (FPGA) is usually used as hardware platform to implement high speed digital signal processor (DSP) systems. There is another implementation for this purpose, which is Application Specific Integrated Circuits (ASIC) used overwhelmingly. Reconfiguration ability and reliability and flexibility of FPGA are good as compared to traditional DSPs and ASICs and product cost is also less. FPGAs can give enhanced speed and permits reconfigurable architectures for realization of FIR filter. So, it is the need of the day to implement the FIR filters on FPGAs.

Various multipliers-less schemes had been proposed, in literature. Conversion-based approach is the first type of multiplier-less technique and the second type includes use of memories (RAMs, ROMs) or Look-Up Tables (LUTs) to store pre-computed values of coefficient operations which are discussed below in section II.

II. RELATED WORK

Over the past years, the technique has been proposed which is software radio whose final idea was the
development of digital signal processing towards the antenna [2]. For replacing analog signal processing with digital signal processing, the main reason was the possibility to softly reconfigure the system, where important functionalities of digital front-end defined and concentrated on unities between different signal processing operations and the signal characteristics of mobile communications signals. The technique was popular for fixed-coefficient FIR filter implementation. In this encoding the filter coefficients using CSD representation reduces the number of partial products and thus in hardware implementation saves silicon area and power consumption. Then for implementation of FIR filter, the digit processing unit has designed. By properly arranging the multiplexers in several cascaded DPUs, and summing up all the output of the DPUs and then accumulated sum, an FIR filter with variable number of CSDs in each tap have implemented. The multiplexer in the corresponding DPU selected the buffered data as the output for the last digit of each tap [3].

Memory based FIR filter architecture using low complexity LUT multiplier for a set of fixed coefficients and FIR filter based on low-complexity CSHM for programmable coefficient have proposed. In the memory based FIR filter architecture, a low-complexity memory based FIR filter using optimized LUT multiplier is suggested. The memory based FIR filter could be more efficient in terms of area and latency. In the LUT based multiplier the complexity is reduced by eliminating the need of decoder compare to DA based design. The reconfigurable FIR filter architecture using the low complexity CSHM where the redundant computations in the proposed CSHM based FIR filter is reduced [10].

In recent years, the distributed arithmetic algorithm based techniques has been very popular worldwide because it is used to produce very efficient filter design. But if the filter order increases then the memory requirement for DA-based implementation of FIR filters exponentially increases. So to minimize the problem of such large memory requirement, systolic decomposition techniques are presented. For the efficient implementation of FIR filter in terms of area-power-delay, the 1-D and 2-D fully pipelined structures have presented [6]. This scheme suggested the use of address length of LUT’s for computation based on DA. Memory size is reduced when the smaller address length for DA-based computing units is used. But because of that the adder complexity and latency increases. Novel 1-D and 2-D structures are also designed for calculation of circular convolution and linear convolution using distributed arithmetic (DA). The structures achieved less memory and less area-delay complexity compared with the existing structures of circular convolution [5]. The input of DA is received in digital form and its analog coefficients are set by using the floating-gate voltage references. How the offset and gain errors affected on DA computational accuracy is analyzed and for the limitations of this design, theoretical results have presented. Further, the low power, high speed architecture of a reconfigurable root raised cosine (RRC) filter has designed. The RRC helps as major component of a digital UP converter. This proposed filter can be reconfigured at any time. The multiplexer based design ensured that the reduction in power consumption.

The two multipliers-less schemes according to how they manipulate the filter coefficients are conversion based method and memory based methods. Canonic Sign Digit (CSD) method and Dempster-McLeod method are the examples of conversion based approach, where the coefficients are characterized by a combination of powers of two in such a way that with adder/subtractors and shifters, multiplication can be simply implemented [13]. The second type is memory based methods in which we can only use memories (RAMs, ROMs) or Look-Up Tables (LUTs) that store pre-computed values of filter coefficient operations. The examples of such methods are Constant Coefficient Multiplier method and the very popular Distributed Arithmetic method [12]. The FIR filter by using DA algorithm is easy to implement on FPGAs. The reconfigurable FIR filter whose filters coefficients dynamically change required to be implemented where the rewritable RAM based LUT can be used. And shared LUT structure can be designed by that the hardware complexity and required area could be less.

A. Distributed Arithmetic

The realization of FIR filter needs more number of additions and multiplications we need to consider the parameters like area, power and timing in hardware implementation. Distributed Arithmetic (DA) algorithm appeared as a very effective solution especially suitable for LUT-based FPGA architectures. The fig.1 shows the DA implementation which consist of LUT’s. The multiplier less architecture of DA algorithm had proposed by Croisier et al [14] and it is based on an effective partition of the function in partial terms using 2’s complement binary representation of data. The partial terms can be calculated first and stored in LUTs. The requirement of memory/LUT capacity increases exponentially with the filter order, and it is detected by Yoo et al. [15] given that DA implementations need 2K words (K is the number of taps of the filter).

In the case of FPGA implementation, the area specification of the FPGA is fixed, so we cannot consider that much exchange in area. That is why we need to study the efficient implementation algorithm. We presented the Distributed Arithmetic algorithm to reduce the number of addition and multiplication. By using this method we can make sure that the area of the FPGA is decreased.
Distributed arithmetic is a bit level readjustment of a MAC to hide the multiplications [12]. It is a powerful and popular technique for reducing the size of a parallel hardware of MAC that is well suited to FPGA designs. The DA objects the products of sums which cover all filtering application and frequency transformation functions. Look-Up Table (LUT) which stores the constant coefficients of FIR Filter used in DA. Distributed Arithmetic (DA) Algorithm can be used to replace MAC unit. Replacement of MAC with LUT-Based DA algorithm is having efficiency, more speed, less area usage, low power consumption and less hardware complexity. By the minimization of arithmetic in terms of multipliers, our ultimate goal is to minimize the parameters namely, hardware, area and power for high order FIR filter. To reduce the hardware complexity the lookup tables can be used and hence the design of FIR filter using Distributed Arithmetic algorithm is more effective [1].

1. The formulation of Distributed Arithmetic algorithm

The following expression represents a multiply and accumulate operation.

\[ y = \sum_{k=1}^{K} H_k x_k \]  

H=[H₁, H₂, ..., Hₖ] is a matrix of constant values

\[ x=[x_1, x_2, ..., x_K] \]

Each \( a_k \) is of M-bits. Each \( x_k \) is of N-bits, y should be able large enough to accommodate the result

Let \( x_k \) be a N-bits scaled two’s complement number

\[ | x_k | < 1 \]

\[ x_k: \{ a_{k0}, a_{k1}, a_{k2}, ..., a_{k(N-1)} \} \]

We can express \( x_k \) as

\[ x_k = -a_{k0} + \sum_{n=1}^{N-1} a_{kn} 2^{-n} \]  

Substituting \( x_k \)in above equation 1

\[ y = \sum_{k=1}^{K} H_k \left[ -a_{k0} + \sum_{n=1}^{N-1} a_{kn} 2^{-n} \right] \]  

We get

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

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

Expand equation (5)

\[ y = [a_{10} \cdot H_1 + a_{20} \cdot H_2 + \cdots + a_{K0} \cdot H_K] \]

\[ + [(a_{11} \cdot H_1) + (a_{21} \cdot H_2) + \cdots + (a_{K1} \cdot H_K)] 2^{-1} \]

\[ + [(a_{12} \cdot H_1) + (a_{22} \cdot H_2) + \cdots + (a_{K2} \cdot H_K)] 2^{-2} \]

\[ + [(a_{1(N-1)} \cdot H_1) + (a_{2(N-1)} \cdot H_2) + \cdots + (a_{K(N-1)} \cdot H_K)] 2^{-(N-1)} \]

From (6) we get

\[ y = \sum_{k=1}^{K} (a_{k0} \cdot H_k) + \sum_{n=1}^{N-1} a_{kn} \cdot H_k \]  

\[ y = \sum_{k=1}^{K} (a_{k0} \cdot H_k) + \sum_{n=1}^{N-1} a_{kn} \cdot H_k \]  

The Final Reformulation

\[ y = \sum_{k=1}^{K} I_{k0} \cdot (a_{k1}) + \sum_{n=1}^{N-1} \sum_{k=1}^{K} [I_{kn}] \cdot (a_{kn}) \cdot 2^{-n} \]  

III. PROPOSED SYSTEM

Fig. 1 shows the proposed structure of the high-throughput reconfigurable FIR filter implementation by using DA algorithm whose filter coefficients of this filter change during runtime.

For reconfigurable DA-based implementation of FIR filter the LUT need to be implemented in RAM. RAM based LUT is costly so a shared LUT architecture is designed for FIR filter. The DRAM (distributed RAM) based design is proposed for FPGA implementation of the reconfigurable FIR Filter which is shown in figure. 1[1].
Registers are rare resource in FPGAs since each LUT in many FPGA devices contains only two bits of registers. Therefore, the LUTs are needed to be implemented by distributed RAM (DRAM) for FPGA implementation. Each bit slice will initiate very high resource consumption using a DRAM to implement LUT. Thus, the partial inner-product generator into Qparallel sections has decomposed and each section has Rtime-multiplexed operations corresponding to Rbit slices. We have R time slots of the same duration so that we can take one filter output every Rcycles.

The proposed structure has Qsections, and each section involves of PDRAM-based RRPGs (DRPPGs) and the PAT which use to compute the rightmost summation of final equation, which followed by shift-accumulator that performs over Rcycles along with the second summation. However, we can use dual-port DRAM to minimize the total size of LUTs by half since two DRPPGs from two different sections can share the single DRAM to be added by the PAT. Shift-accumulator will be accumulated the output of the PAT over Rcycles. Finally, the PSAT creates the filter output using the output from each section every Rcycles. At every Rcycles the accumulated value is reset by the control signal to keep the accumulator register ready and to be used for calculation of the next filter output.

**CONCLUSION**

The distributed arithmetic is an efficient technique of FIR filter implementation in terms of area. The distributed arithmetic algorithm is very simple and its applications are broad. Slicing of LUT of desired length can be effective for high order filter designs. The parameters namely area, power, hardware complexity can be reduced by high throughput reconfigurable FIR digital filter implemented by using Distributed arithmetic. The hardware cost could be reduced by sharing the same LUT by the DA units for different bit slices. In future work, the reconfigurable FIR filter whose filters coefficients dynamically change required to be implemented where the rewritable RAM based LUT would be used.

**REFERENCES**


