ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 8, August 2017

Efficient Design of a Reconfigurable FIR Filter using Distributed Arithmetic for FPGA Implementation

[1] Reshma Ghurde, [2] Mrs. Aparna Shinde
[1][2] Dept. of Electronics and Telecommunication Engg
[1][2] D. Y. Patil College of Engineering, Akurdi, Pune, India

***Abstract* - This paper present Distributed Arithmetic (DA) Algorithm for high-throughput reconfigurable implementation of an FIR Filter. When we directly applied the DA algorithm to FPGA for realization of an FIR filter, it is difficult to achieve the best configuration in the coefficient of FIR filter, the storage resource and the computing speed. For the FPGA implementation, the Dual-Port Distributed RAM based lookup table (LUT) are required for Reconfigurable FIR Filter. Registers are required to store the result of partial inner products of different bit positions for DA processing, but here registers are shared by the DA units for bit slices of different weightage.**

Index Terms— Distributed arithmetic, finite-impulse response (FIR) filters, reconfigurable implementation, lookup-table (LUT).

I.INTRODUCTION

FIR Filter is plays vital role in several Digital Signal Processing Application because of its stable response and linear phase [1]. Now a days FIR Filter is widely used in Wireless Communication such as Software Defined Ratio (SDR) system, Speech Processing, Loud Speaker equalization, Echo cancellation, so therefore FIR Filter require reconfigurablility and low complexity. Complexity of implementation grows with FIR Filter order, and realization of these highly complex filters with desired level of accuracy is a challenging task. Therefore several attempts have been made to developed reconfigurable architecture of FIR Filter in Application Specific Integrated Circuits (ASIC) and Field Programmable Gate Array (FPGA) platform. The FIR structure consist of a series multiplication and addition units and consumes N-MAC blocks of FPGA, which are expensive in high speed system. When DA based algorithm compared with traditional direct arithmetic, bit serial representation save the hardware resources.

Transpose Form is used in Multiple Constant Multiplication (MCM) based technique which can be used only when the filter coefficients of FIR Filter is constant [2]. The MCM technique reduce the total number of additions required for the realization of multiplication, when given input is multiplied with a set of constant coefficients [2]. Power consumption, speed and area are main challenges in Very Large Scale Integration (VLSI), for real time applications less hardware and latency become more important in today's date.

A Distributed Arithmetic (DA) plays a vital role in embedding DSP Functions in the XILINX family of FPGA devices. In recent years DA technique has gained because of its high-throughput processing capabilities and increased regularity, which gives result in area consumption and cost effective computing structure. The memory requirement for FIR Filter by using DA-based implementation is exponentially increases with the filter order. The fundamental operation for DA-based FIR Filter implementation require a sequence of lookup table (LUT) which is followed by adder tree and shift accumulator. When coefficients of FIR Filter are fixed, then this behavior makes it possible to use ROM-based LUTs. For Reconfigurable application, we require DA-based FIR Filter whose filter coefficients dynamically change, and this time we need to use rewritable RAM-based LUT rather than ROM-based LUT [8].

Systolic design represent an architecture for efficient hardware implementation of computation-intensive DSP applications, which supported features like simplicity, regularity and modularity of structure [5]. This systolic structure architecture is flexible choice of the address length of the look-up table (LUT) for DA-based computation [5].

In the section II, we briefly discuss the mathematical background of DA algorithm of FIR filter. In Sections III describe the proposed method for reconfigurable DA-based FIR filter for field-programmable gate array (FPGA) implementations. We provide the synthesis results of the proposed designs in Section IV. Finally, the conclusion is given in Section V.

II. DA ALGORITHM FOR IMPLEMENTATION OF AN FIR FILTER

Distributed Arithmetic is simply a bit serial computational operation which forms an inner dot product of a pair of vectors in a single direct step. It also perform multiply accumulate operation which is very common in DSP applications. DA specially targets the sum of products computation that covers many of the important Digital Signal Processing filtering and frequency transforming functions. The main advantage of DA are best exploited in data path circuit designing. DA hides the explicit multiplications by ROM lookup tables, which is implement on FPGA. The fundamental operation of FIR filter is a arithmetic sum of product of two vectors, expressed as,

$$y(n) = \sum_{k=0}^{n-1} h(k) * x(n-k) \qquad \text{….(1)}$$

Where, h(k) is inner product of impulse response for k=0,1,….,N-1
X(n-1) is input vector for k=0,1,…..,N-1
Let remove the time index n as,

$$y = \sum_{k=0}^{n-1} h(k) * s(k) \qquad \text{….(2)}$$

Assume L is wordlength,
Let s(k) may be expressed in two's complement representation, i.e.,

$$s(k) = -[s(k)]_0 + \sum_{l=1}^{L-1} [s(k)]_i 2^{-l} \qquad \text{….(3)}$$

Substituting (3) in (2) then,

$$y = \sum_{k=0}^{n-1} h(k)[s(k)]_0 + \sum_{k=0}^{N-0} h(k) \{\sum_{i=1}^{L-1} [s(k)]_l 2^{-l}\} \text{ (4)}$$

from equation (4), to convert t0he sum of products form of inner product of equation (2) into a distributed form, we have to change the order of summation,

$$y = \sum_{k=0}^{n-1} h(k)[s(k)]_0 + \sum_{i=1}^{L-1} 2^{-l} \{\sum_{k=0}^{N-1} h\,(k)[s(k)]_l\} \text{ (5)}$$

from equation (5), inner products can be computed as,

$$y = \sum_{l=1}^{L-1} 2^{-l} c_1 - \ c_0 \qquad \text{……(6 a)}$$

where,

$$c_l = \sum_{k=0}^{N-1} h(k)[s(k)]_l \qquad \text{…..(6 b)}$$

Here N point bit sequence [s(k)]l for 0≤k≤N-1 can be either 0 or 1, the partial sum Cl for 0≤l≤L-1can have 2N possible values. If all this values precomputed and stored in LUT, then the partial sum can be read out from the LUT using bit sequence as a address bit for calculating inner products. The inner product given by (6a) then can be expressed in a simpler form i.e.,

$$y = \sum_{l=0}^{L-1} 2^{-l} C_l \qquad \text{……(7)}$$

so that no sign reversal of LUT output is required.
For large values of N, LUT size becomes too large and LUT access time also become large. Therefore this DA based implementation is not suitable for large filter orders. When N is a composite number given by N = PM, where P and M may be any two positive integers, one can map the index k into (m+pM) for m =0, 1, . . . , M − 1 and p = 0, 1, . . . , P − 1 to express (7) as

$$y = \ \sum_{l=0}^{L-1} 2^{-l} \, (\sum_{p=0}^{p-1} S_{l.p} \qquad \text{……(8a)}$$

Where $S_{l,p}$ is the sum of partial product of M samples represented as

$$S_{l,p} = \sum_{m=0}^{M-1} h(m+pM)[s(m+pM)]_1 \qquad \text{….. (8b)}$$

For l=0,1,…..,L-1 and p=0,1,………,P-1.

For any given sequence of impulse response *{h(k)}*, the 2^M possible values of Sl,p corresponding to the 2M permutations of M-point bit sequence *{(s(m + pM))l}*, for m = 0, 1, . . . , M - 1 and l = 0, 1, . . . , L - 1, may be stored in the LUT of 2M words. These values of Sl,p can be read out when the bit sequence is
fed to the LUT as address. Equation (8) may, thus, be written in terms of memory-read operation as

$$y = \sum_{l=0}^{L-1} 2^{-l} [\sum_{p=0}^{P-1} \mathcal{F}(b_{l,p}) \qquad \text{…..(9)}$$

Where Ƒ(*bl,p*) = *Sl,p* and

$$bl,p \ = \ \{[s(pM)]l \ [s(1+pM)]l \ , \dots\dots, [s(M-1+pM)]l \ \} \qquad \text{………(10)}$$

for 0 ≤ l ≤ L - 1 and 0 ≤ p ≤ P - 1. The bit vector bl,p is used as address word for the LUT, and Ƒ(·) is the memory-read operation.
DA technique hides the explicit multiplications in Look Up Table (LUT) and is an efficient technique to implement on Field Programmable Gate Arrays (FPGAs). It uses look-up tables and accumulators instead of multipliers for computing inner products. The DA of FIR filter consists of Look up Table (LUT), Shift Register (SR) and Scaling Accumulator (SA). This algorithm is based on the scaling accumulation algorithm. This accumulator takes one parallel and one serial input. The parallel input in DA algorithm is considered to be a constant. Several Scaling accumulator units can be used in parallel to conduct the MAC operation for many terms. The constants with the AND operators and partial sums can represent product terms that can have predefined values. These equations can be implemented using Read Only Memory (ROM) where its contents are defined by the constants and their addresses are inputs bits. Since only one bit of each input goes to the ROM address The (ROM) contents are defined as follow:

Addr (000) => 0

Addr (001) => C0

Addr (010) => C1

Addr (011) => C0+C1

Addr (100) => C2

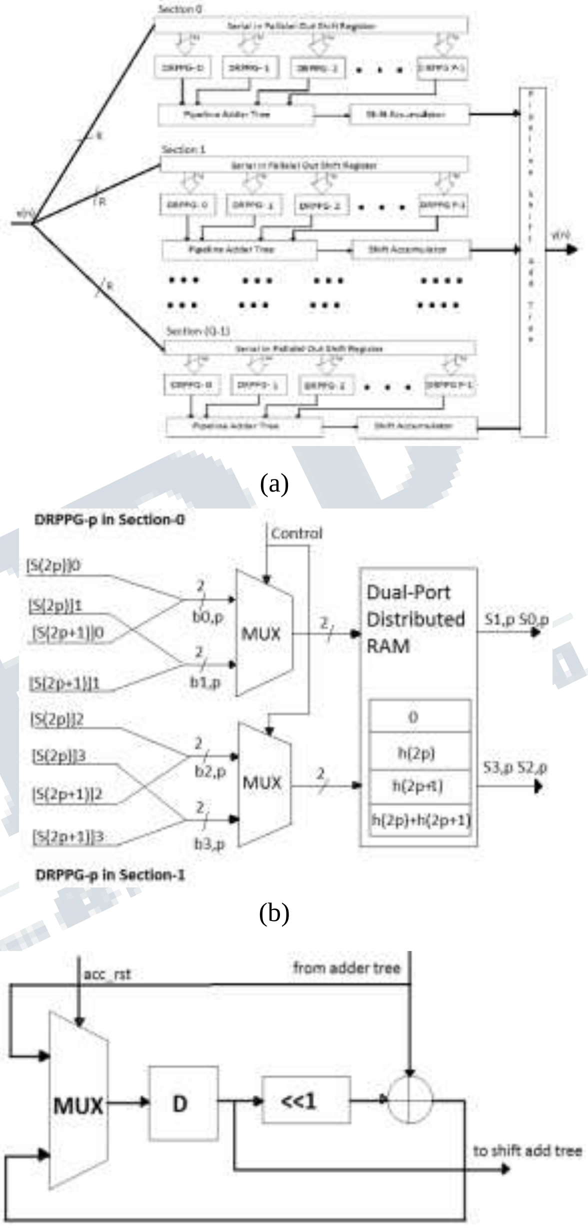
Addr (101) => C0+C2

Addr (110) => C1+C2

Addr (111) => C0+C1+C2

III. PROPOSED RECONFIGURABLE DA-BASED FIR FILTER FOR FPGA IMPLEMENTATION

Now a days FPGA become a crucial part in many wireless communication as well as DSP applications. The proposed structure for reconfigurable FIR filter also implemented on FPGA.



(a)

(b)

(c)

Fig. 1 Proposed structure of the DA-based FIR filter for FPGA implementation. (a) Structure of the DA-based FIR filter. (b) Structure of the DRPPG for M = 2and R = 2. (c) Structure of the shift-accumulator.

By using DA based algorithm the total number of area is reduced, therefore we have propose an efficient FPGA implantation for DA based reconfigurable FIR filter. Registers are scarce resource for FPGA implementation because of each LUT in many FPGA devices contain only two bits of registers. Therefore, the LUT are required to be implemented by Distributed RAM (DRAM) for FPGA implementation.

If L is the bit width of input, then total duration of design is L times the operating clock period, which may not be suitable for application requiring high through-put. If we use DRAM for each bit slice for the implementation of each LUT will lead to very high resource consumption. Therefore, we decompose the partial inner product generator of FIR filter into Q parallel sections, where each section has R time multiplexed operations corresponding to R bit slices. When L is composite number given by L=RQ, where R and Q are two positive integers. Then the index l in (8a) equation can be mapped into (r+qR) for r=0,1,…..R-1 and q=0,1,…...Q-1 then equation (8a) modify as,

$$y = \sum_{q=0}^{Q-1} 2^{-Rq} [\sum_{r=0}^{R-1} |2^{-r} (\sum_{p=0}^{P-1} S_{r+q\,R,p} \quad \ldots(11)$$

where q is section index and r is time index. In the proposed method, we have R time slots of same duration as operating clock period, so that we get one filter output at every R cycle. Fig. (1a) show the DA based FIR filter for FPGA implementation. To implement equation (11), the proposed structure has Q sections and each section is consist of P number of DRAM based Reconfigurable Partial Product Generator (DRPPG). Then the output of DRPPG is fed to the Pat to calculate the rightmost summation, followed by pipeline shift add tree that perform addition of different sections outputs over R cycles according to the second summation.

In this proposed method, dual port DRAM is used to reduce the total size of LUT by half because two DRPPG from two different sections can share the single DRAM. This is shown in fig. (1b). this DRPPG generates QP partial inner products in a single cycle.

In rth cycle, p number of DRPPG generates qth section generates P partial products Sr+qR,p for p=0,1,…..P-1, which is then added to the PAT. The output of PAT are accumulated by shift accumulator over R cycle which is shown in fig. (1c). Finally Pipeline Shift Add Tree (PSAT) produce the filter output by adding the output of different Q sections. This accumulated value is reset every R cycles by control signal to keep the accumulator register ready to use for calculation of next filter output. If the maximum operating clock period is fclk, then proposed structure can support the input sample rate of fclk/R.

IV. RESULTS

If we compare the proposed structure with a conventional DA based structure [3], and DA based systolic structure [4], [5], then hardware and time complexity is reduced in proposed structure. In this proposed structure, the duration of minimum clock periods depends on the delay of final adder stage in PSAT block.

The structure in fig. (1), by using DRPPG for FPGA implementation gives a higher through-put than existing structure [3], [4], [5]. For the proposed structure Verilog language is used for the coding and it is simulated in Xilinx ISE Design Suite 14.1. For the FPGA implementation, Spartan XC6XSL45 board is used.

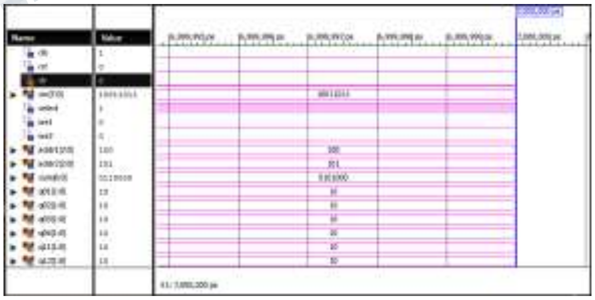


Fig. 2 Output waveform in Xilinxs.

When we go for proposed DA–based reconfigurable FIR filter implementation the output will be same but the complexity get reduced due to decomposed dual port RAM Implementation and it is given in Fig. 2. It is observe that synthesis results shows that the use of decomposed RAM structure reduces the complexity when we go for higher tap and comparatively the area get reduced and speed also get increased due to change in area.

V. CONCLUSION

This paper presents the design and implementation of DA based reconfigurable FIR digital filter design .The simulation results of single LUT based RAM structure gives much complexity when the TAP increases there we cannot implement single structure its quiet difficult task and area consuming process. The proposed structure is easy to implement with higher tap with the help of decomposed Distributed RAM (DRAM) structure. With this structure, we get higher through-put and low complexity design than existing structure.

REFERENCES

[1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications. Upper Saddle River, NJ, USA, Prentice-Hall, 1996.

[2] B. K. Mohanty and P. K. Meher, “A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications” IEEE Tran on VLSI,Feb 2015.

[3] S. A. White, “Applications of distributed arithmetic to digital signal processing: A tutorial review,” IEEE ASSP Mag., vol. 6, no. 3, pp. 4– 19, Jul. 1989

[4] P. K. Meher, “Hardware-efficient systolization of DA-based calculation of finite digital convolution,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 707–711, Aug. 2006.

[5] P. K. Meher, S. Chandrasekaran, and A. Amira, “FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic,” IEEE Trans. Signal Process., vol. 56, no. 7, pp. 3009–3017, Jul. 2008.

[6] M. Kumm, K. Moller, and P. Zipf, “Dynamically reconfigurable FIR filter architectures with fast reconfiguration,” in Proc. 8th Int. Workshop ReCoSoC, Jul, 2013, pp.1-8.

[7] T. Hentschel, M. Henker, and G. Fettweis, “The digital front-end of software radio terminals,” IEEE Pers. Commun. Mag., vol. 6, no. 4, pp. 40–46, Aug. 1999.

[8] S. Y. Park and P. K. Meher “Efficient FPGA Realizations of a DA-Based Reconfigurable FIR Digital Filter” IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 61, NO. 7, JULY 2014