

FPGA Implementation of Reconfigurable Digital FIR Filter

Pradnya D. Shahare

Student, M. Tech. (Electronic System and Communication)
Department of Electronics and Telecommunication
Government College of Engineering
Amravati, MH, India, 444604
pradnyashahare29@gmail.com

Samrat S. Thorat

Assistant Professor
Department of Electronics and Telecommunication
Government College of Engineering
Amravati, MH, India, 444604
samratthorat@gmail.com

Abstract—This paper presents, high-throughput, reconfigurable implementation of finite-impulse response (FIR) filter. The proposed architecture, whose filter coefficients change during runtime is designed by using efficient Distributed Arithmetic. For the reconfigurable filter, the LUT's are required to be implemented in RAM and RAM based LUT's are costly so the shared LUT's architecture is designed. Distributed RAM from two different sections shares one LUT at a time to realize DA computation. The paper also presents FPGA implementation of the FIR filter, which supports up to 91 MHz input sampling frequency while implemented in the Xilinx FPGA device. In this paper, comparison of single and decomposed RAM structure of FIR filter is discussed. It offers about 50% and 25% less number of slice than the systolic structure and the CSA-based structure, respectively. The parameters namely area, power and hardware reduce in the proposed DA based FIR filter.

Keywords-Distributed Arithmetic; Field Programmable Gate Array; FIR Filter; Look Up Table.

I. INTRODUCTION

The present era of mobile computing and multimedia technology demands the low power and high performance in DSP and VLSI application. And the most widely used function in DSP is FIR filtering which perform weighted summation of input sequences. Design of FIR filter in terms of low power, less area and high speed is the key issue for many applications. Due to intensive use of FIR filter in video and communication systems, high performance in speed, area and power consumption is demanded. In DSP, the design methods were mainly focused in multiplier-based architectures to implement the multiply-and- Accumulate (MAC) blocks that establish the central piece in FIR filters and many functions [3].

Most of the digital signal processing applications includes FIR filters due to its linearity and stability. But the limitation is the large number of taps, for the desired frequency response, and that causes area complexity. To overcome this type of problems FPGAs offer a very effective solution that balance cost, high flexibility, time-to-market and performance. 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. These methods can be classified in two types according to operation of the filter coefficients for the multiply operation. Conversion-based approach is the first type of multiplier-less technique and the second type includes use of memories like Look-Up Tables (LUTs), (RAMs, ROMs) to store pre-computed values of coefficient operations[11],[13].

Over the past years, the software radio technique has been proposed whose ultimate idea was the development of digital signal processing towards the antenna [2]. Due to demand of reconfigurable communication systems, this technique was popular worldwide, which is capable of multistandard operations. Then the canonical signed digit (CSD) representation used in design of filter that reduce the complexity of FIR digital filter implementation. This technique was popular for fixed-coefficient FIR filter implementation [4]. In recent years, the distributed arithmetic algorithm based techniques has been very popular worldwide because of efficient filter design. A sequence of lookup table (LUT) accesses followed by shift accumulation operations of

the LUT output are the main operations required for DA-based computation. 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], [7]. But because of that the adder complexity and latency increases. The reconfigurable FIR filter whose filters coefficients dynamically change, the rewritable RAM based LUT can be used to implement it instead of ROM-based LUT [12]. FPGA technology has extremely grown from a dedicated hardware to a heterogeneous system, which is considered to be a popular choice in communication base stations instead of being just a prototype platform [1]. The FIR filter by using DA algorithm is easy to implement on FPGAs. The advantages of the FPGA approach to digital filter implementation involves higher sampling rates and lower costs than an ASIC for moderate volume applications, and more flexibility than the another approaches.

In the next section, the Distributed Arithmetic is explained.

II. DECOMPOSITION OF FIR FILTER

The output of FIR filter of length N is given by

$$y(n) = \sum_{k=0}^{N-1} h(k) x(n-k) \quad \dots\dots\dots(1)$$

Where, $h(k)$ – Impulse response vector for ($k = 0,1,\dots,N-1$)

$x(n-k)$ – Input vector for ($k = 0,1,\dots,N-1$)

For simplification, let us remove time index n as

$$y(n) = \sum_{k=0}^{N-1} h(k) f(k) \quad \dots\dots\dots(2)$$

where $f(k) = x(n-k)$ – input sample

Assuming L – word length,

$f(k)$ may be expressed in two's complement representation as

$$f(k) = -[f(k)]_0 + \sum_{i=1}^{L-1} [f(k)]_i 2^{-i} \quad \dots\dots\dots(3)$$

where $[f(k)]_i$ denotes the i th bit of $f(k)$. we

can write (2) in an expanded form, by Substituting (3),i.e.

$$y = - \sum_{k=0}^{N-1} h(k) [f(k)]_0 + \sum_{k=0}^{N-1} h(k) \left\{ \sum_{i=1}^{L-1} [f(k)]_i 2^{-i} \right\} \quad \dots\dots(4)$$

Interchanging the order of summations over the indices k and i in (4) to have

$$y = - \sum_{k=0}^{N-1} h(k) [f(k)]_0 + \sum_{i=1}^{L-1} 2^{-i} \left\{ \sum_{k=0}^{N-1} h(k) [f(k)]_i \right\} \quad \dots\dots(5)$$

and the inner product given by (5) can be computed as

$$y = \sum_{i=1}^{L-1} 2^{-i} A_i - A_0 \quad \dots\dots\dots(6)$$

Where

$$A_i = \sum_{k=0}^{N-1} h(k) [f(k)]_i \quad \dots\dots\dots(7)$$

Without a loss of generality, and for simplicity of discussion, the signal samples to be unsigned words of size L may assume, although the proposed algorithm can be used for two's complement coding and offset binary coding also. Unsigned input signal by adding fixed offset can be always obtained when the original input signal is signed. The inner product given by (6) then can be expressed in a simpler form, i.e.

$$y = \sum_{i=0}^{L-1} 2^{-i} A_i \quad \dots\dots\dots(8)$$

We can use (8) directly for straightforward DA-based implementation of FIR filter using the LUT. However, the LUT size becomes too large, and the LUT access time also becomes large for large values of N . Therefore, for large filter orders the straightforward DA-based implementation is not proper. When N is a composite number given by $N = PM$ (P and M may be any two positive integers), one can map the index k into $(m + pM)$ for $m = 0, 1, \dots, M - 1$ and $p = 0, 1, \dots, P - 1$ to express (8) as

$$y = \sum_{l=0}^{L-1} 2^{-l} \left(\sum_{p=0}^{P-1} F_{l,p} \right) \dots\dots\dots(9)$$

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

$$F_{l,p} = \sum_{m=0}^{M-1} h(m + pM) [f(m + pM)]_l \dots\dots\dots(10)$$

III. DISTRIBUTED ARITHMETIC

In DSP applications, distributed arithmetic is an important algorithm, an old technique that has been revived by the wide spread use of Field Programmable Gate Arrays (FPGAs) for Digital Signal Processing (DSP). It is a bit level rearrangement of the multiply and accumulate operation to hide the multiplication. Basically, it targets the sum of products computation that covers many of DSP functions like filtering. It is an efficient technique for calculation of sum of products or vector dot product or inner product. In all Digital Signal Processing Algorithms, DA targets the sum of products computation for MAC operation. For high-throughput processing capability and regularity, which result in cost-effective and area-time efficient computing structures DA based techniques are useful. The algorithm of DA is very easy to understand but its applications are broad. Area savings from using DA can be up to 80% in digital signal processing hardware designs. DA effectively implements the MAC using basic building blocks (Look Up Tables) in FPGAs.

In DA the task of summing product terms is replaced by look-up table procedures that are easily implemented in the

Xilinx configurable logic block (CLB) look-up table architecture. The (ROM) contents are defined as follows:

- Address (000) => 0
- Address (001) => A0
- Address (010) => A1
- Address (011) => A0+A1
- Address (100) => A2
- Address (101) => A0+A2
- Address (110) => A1+A2
- Address (111) => A0+A1+A2

IV. FIR FILTER FOR FPGA IMPLEMENTATION

Fig. 1 shows the structure of the proposed time-multiplexed DA-based FIR filter using DRAM for FPGA implementation. The proposed reconfigurable FIR filter may be also implemented on FPGA. The distributed arithmetic is the popular and efficient technique to reduce the size of a parallel hardware multiply-accumulate that is well suited to FPGA. And DA involves Look-Up Table that stores constant coefficients of FIR filter. For FPGA implementation, we propose a reconfigurable DA based FIR filter.

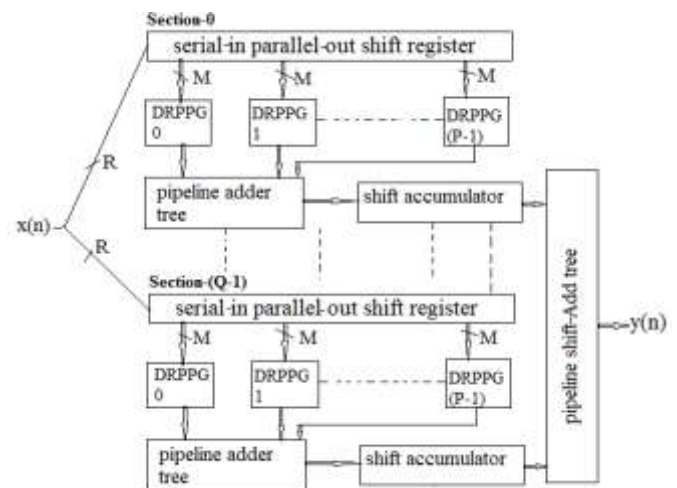


Figure 1. Proposed structure of the DA- based FIR filter for FPGA implementation

However, registers are rare resource in FPGA because 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. So

that single DRAM can be shared by two DRPPGs (i.e. distributed RAM partial product generator) of different sections. Thus, we design the distributed RAM partial inner-product generator into Q parallel sections and each section has R time-multiplexed operations corresponding to R bit slices. The index l in (9) can be mapped into $(r + qR)$ for $r = 0, 1, \dots, R - 1$ and $q = 0, 1, \dots, Q - 1$ to modify (9) as

$$y = \sum_{q=0}^{Q-1} 2^{-Rq} \left[\sum_{r=0}^{R-1} 2^{-r} \left(\sum_{p=0}^{P-1} F_{r+qR,p} \right) \right] \dots\dots\dots (11)$$

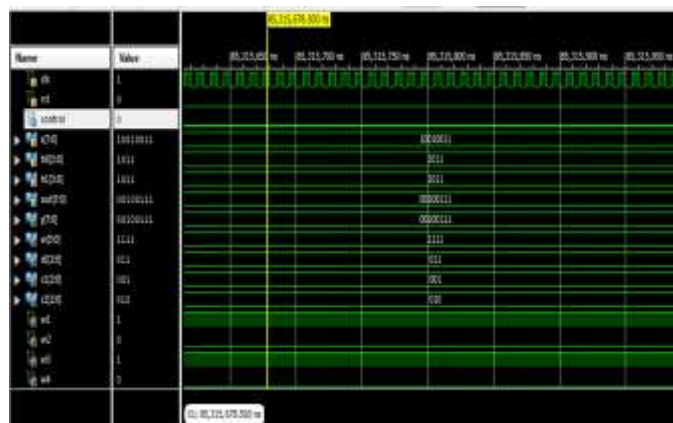
$L = RQ$ (R and Q are two positive integers), where L is a composite number.

In (11), the indices q and r are the section index and time index, respectively. To implement equation (11), the proposed structure divided in Q sections, and each section contains P number of (DRPPGs) and the PAT (pipeline adder tree) to calculate the rightmost summation. Shift -accumulator which performs over R cycles used for the second summation. The output of the PAT is accumulated by a shift-accumulator. The accumulated value is reset every R cycles by the control signal 'acc_rst' to keep the accumulator register ready to be used for calculation of the next filter output. Finally, the PSAT produces the filter output using the output from each section every R cycles. The proposed structure can generate QP partial inner products in a single cycle, In the r th cycle, P DRPPGs in the q th section generate P partial inner products $S_{r+qR,p}$ for $p = 0, 1, \dots, P - 1$. And the products are added by the PAT. By using dual-port DRAM we can reduce the total size of LUTs by half. We have R time slots of the same duration so that we can have one filter output every R cycles.[1]

V. SIMULATION RESULTS

The simulation results are carried out in Xilinx ISE 13.1 and area, speed constraints are synthesized in this tool. The Verilog HDL is used for programming. The simulation of DA based reconfigurable FIR for decomposed RAM and single RAM is shown in fig.2 and fig.3 respectively.

A. Simulation Results



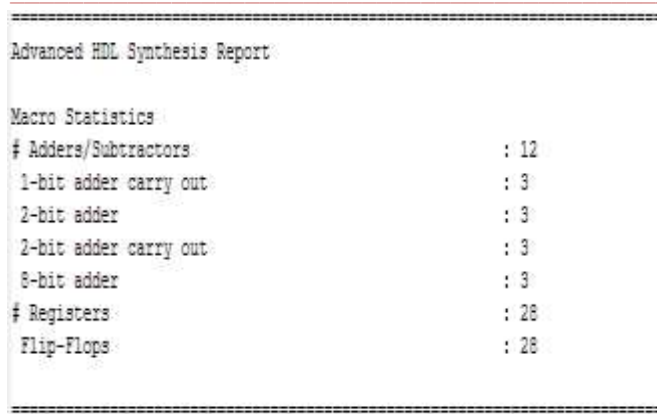


Figure 5. Synthesis report of DA based reconfigurable FIR filter for decomposed RAM structure

C. Comparison results of Conventional Single RAM Structure and Decomposed RAM Structure

In the case of the RPPG in ASIC design i.e single RAM structure, only one LUT value can be read from the DRAM per cycle because of that the multiple number of partial inner products Sl,p cannot be saved from the DRAM simultaneously. Each bit slice will consume very high resource using a DRAM. The structure of ASIC involves $N(2M - 1)/M$ number of registers for the implementation of LUTs for FIR filter of length N [1].

If L is the bit width of input, the duration of the sample period of the design is L times the operating clock period, which may not be suitable for the application requiring high throughput. The comparison of result after simulation is shown in figure 4 and table 1 for 8 bit input.

TABLE I. DEVICE UTILIZATION SUMMARY

DA based reconfigurable FIR filter	Device utilization summary after simulation		
	No. of Slices	No. of 4 input LUT's	No. of bonded IOB's
Conventinal single RAM structure	45	83	21
Proposed decomposed RAM structure	18	30	18

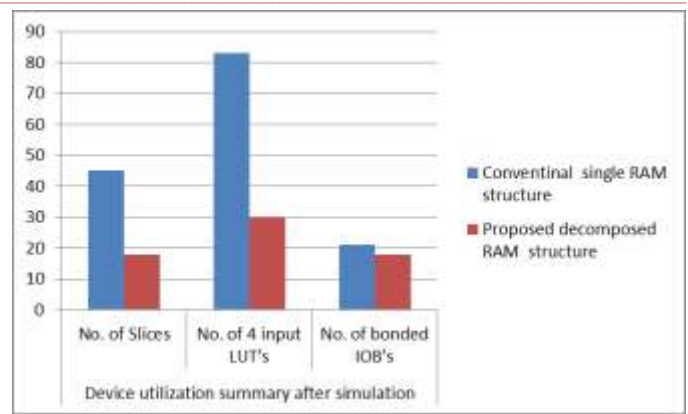


Figure 6. Comparison of result of single RAM and decomposed RAM architecture

VI. CONCLUSION

Efficient technique i.e. distributed arithmetic for high throughput reconfigurable implementation of FIR digital filters has presented. DA based technique has proved to be an area efficient technique. The hardware cost could be reduced because same registers can be shared by DA units for different bit slices. The result generated in Xilinx ISE 13.1 using Verilog HDL. The FIR filter for the ASIC implementation and FPGA implementation has compared. The FIR filter for FPGA implementation requires half the number of registers as compared to ASIC implementation. The proposed structure of FIR filter for FPGA implementation supports up to 91 MHz input sampling frequency. It has offered about 50% and 25% less number of slice than the systolic structure and the CSA-based structure, respectively. The area, power and hardware can be reduced in FIR filter by using distributed arithmetic.

REFERENCES

- [1] Sang Yoon Park and Pramod Kumar Meher, "Efficient FPGA and ASIC Realizations of a DA-Based Reconfigurable FIR Digital Filter", *IEEE transactions on circuits and systems—II: express briefs*, vol. 61, no. 7, July 2014
- [2] 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.
- [3] John G. Proakis, Manolakis, "Digital Signal Processing" Principles, algorithm and applications (Fourth Edition)- 2008.

-
- [4] K.-H. Chen and T.-D. Chiueh, "A low-power digit-based reconfigurable FIR filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 617–621, Aug.
- [5] L. Ming and Y. Chao, "The multiplexed structure of multichannel FIR filter and its resources evaluation," in *Proc. Int. Conf. CDCIEM*, Mar.2012, pp.764-768.
- [6] 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.
- [7] 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.
- [8] E. Ozalevli, W. Huang, P. E. Hasler, and D. V. Anderson, "A reconfigurable mixed-signal VLSI implementation of distributed arithmetic used for finite-impulse response filtering," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 2, pp. 510-521, Mar.2008.
- [9] P. K. Meher and S. Y. Park, "High-throughput pipelined realization of adaptive FIR filter based on distributed arithmetic," in *Proc. IEEE/IFIP 19th Int. Conf. VLSI-SOC*, Oct. 2011, pp. 428-433.
- [10] J. L. Mazher Iqbal, S. Varadarajan "New Approach to Memory Less Design and Look-Up-Table Realization for Low-Complexity Reconfigurable Digital FIR Filter Architectures", WSEAS TRANSACTIONS on SYSTEMS, Issue 3, Volume 12, March 2013, pp.142-153
- [11] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital Filter for PCM Encoded Signals", U.S. Patent No. 3,777,130, issued April, 1973.
- [12] 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.
- [13] H. Yoo, and D. Anderson, "Hardware-Efficient Distributed Arithmetic Architecture for High-Order Digital Filters", in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, Vol. 5, pp. 125 – 128.