OPTIMIZATION OF DISTRIBUTED ARITHMETIC BASED FIR FILTER

Kanu Priya[a], Sarbdeep Singh[b]

[a,b] Chandigarh Engineering College, Landran, Mohali

Abstract: The VLSI design industry has grown rapidly during the last few decades. The complexity of the applications increases day by day due to which the area utilization increases. The tradeoff between area and speed is an important factor. The main focus of continued research has been to increase the operating speed by keeping the area and memory utilization of the design as low as possible. We present the parallel DA approach in which LUT decomposed into small units to enhance the operating speed and to reduce the critical path.

Keywords— Distributed Arithmetic (DA), Digital Signal Processing (DSP), Look Up Table (LUT) and Field Programmable Gate Array (FPGA).

1. Introduction

The implementation and precision of computation of higher order filters becomes complex which is a challenging task for real time realization of these types of filters [5]. The features like simplicity, modularity and regularity of structure is supported by the systolic design which represents efficient hardware architecture for computation intensive DSP applications [7], [9]. To yield high throughput rate they also possess significant potential by implementing high level of concurrency using parallel processing or pipelining [6]. The multipliers acquire large area which enforces limitation of maximum possible number of logic elements. Memory based structures have advantages like reduced latency, high throughput and more regular as compare to multiply accumulate unit structures [3], [4]. For memory read operations, the dynamic power consumption is less due to the less switching activities as compare to conventional multipliers. In order to utilize the memory-based structures, all the elements like adders, multipliers and delay units are removed to reduce the area and system latency. Several architectures have been demonstrated for memory-based structures of Digital Signal Processing applications and digital filters [8]. In this paper, Distributed Arithmetic (DA) technique is used which has high throughput and regularity that makes it cost effective and area time efficient structure [1]. In the conventional FIR filters multipliers and adders are used for the computational workload to calculate the inner products of two vectors. When sum of products are computed sequentially by using multiplies and adders, then the multiplication of two B-bit numbers requires B/2 to B additions which is not an efficient use of time and if the multiplication is computed in parallel with B/2 to B adders then it is area intensive. For K-tap filter whether computed serially or in parallel, it need at least B/2 additions per multiplication and K -1 addition for the summation of products. Therefore, K *(B + 2)/2 -1 additions are required for a K-tap filter using multipliers and adders. In DA, multipliers and adders are removed by the look up table (LUT), shift registers and scaling accumulator that provide high-throughput processing capability, cost-effective and area-time efficient computing structures. By using the DA the computation of a K tap filter can be compressed from K multiplication and K-1 additions into a LUT and provide result in B-bit time duration using B - 1 additions. Number of additions are reduced in DA for filtering operation and this reduction is significant for high bit precision filters. Pre computed partial sums of the filter coefficients are stored in the memory table which results in computational workload reduction [2].

2. DA Filter Background

Distributed Arithmetic (DA) technique is bit-level rearrangement of the multiply and accumulation operation. DA is a bit-serial operation used to compute the inner products or weighted sum of products of a constant coefficient vector and a variable input vector.

$$y=\sum_{k=1}^{K}C_k X_k \qquad (1)$$

where y – output response, C_k – constant filter coefficients, X_k – input data

Assume X_k be a N- bits and can be expressed in scaled two's complement number as

$$X_k=-b_{k0}+\sum_{n=1}^{N-1}b_{kn}2^{-n} \qquad (2)$$

Substituting (2) in (1) we get

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

$$y=-\sum_{k=1}^{K}\left(b_{k0}\bullet C_k\right)+\sum_{k=1}^{K}\sum_{n=1}^{N-1}\left(C_k\bullet b_{kn}\right)2^{-n}$$

$$y=-\sum_{k=1}^{K}\left(b_{k0}\bullet C_k\right)+\sum_{k=1}^{K}\left[\sum_{n=1}^{N-1}\left(b_{kn}\bullet C_k\right)2^{-n}\right] \qquad (3)$$

According to the power terms after rearranging the summation we get the final equation

$$y=-\sum_{k=1}^{K}C_k\bullet(b_{k0})+\sum_{n=1}^{N-1}\left[\sum_{k=1}^{K}C_k\bullet b_{kn}\right]2^{-n} \qquad (4)$$

IJERGS

From the equation (4) it is clear that a binary AND operation is performed between a single bit of input variable and all the bits of the constant coefficient and the exponential factors indicates the scaling of each bit. The expression in the second part of the equation (4) has 2^K possible values. Now these values are pre-calculated for all values of k and are used to store in the look up table of 2^K words and each is addressed by K bits [2].

3. PROPOSED SCHEME

The problems encountered in the serial architecture are solved by using parallel distributed architecture. In parallel architecture the filter has high performance at the cost of larger area. The parallel architecture reduces the number of clock cycles for processing each input sample by a factor of two. This improvement comes at the price of doubling the number of required LUTs and the size of the scaling accumulator which is required to store the intermediate results. By increasing the number of input bits processed at each cycle, the performance can be further improved. Since for the higher order FIR filters the efficiency reduces due to the occupancy of large memory locations in single DALUT as shown in Fig. 4.2. In this method we use the pipelined registers for each decomposed LUT that are further responsible for the operating frequency enhancement. In parallel processing multiple outputs are computed in parallel in a clock period and the effective sampling speed is increased by the level of parallelism. The power consumption is reduced by using parallel processing. The effective critical path is reduced by introducing pipelining latches along the critical data path. Pipelining either increases the clock speed or sampling speed or reduces the power consumption at same speed in a DSP system [5].

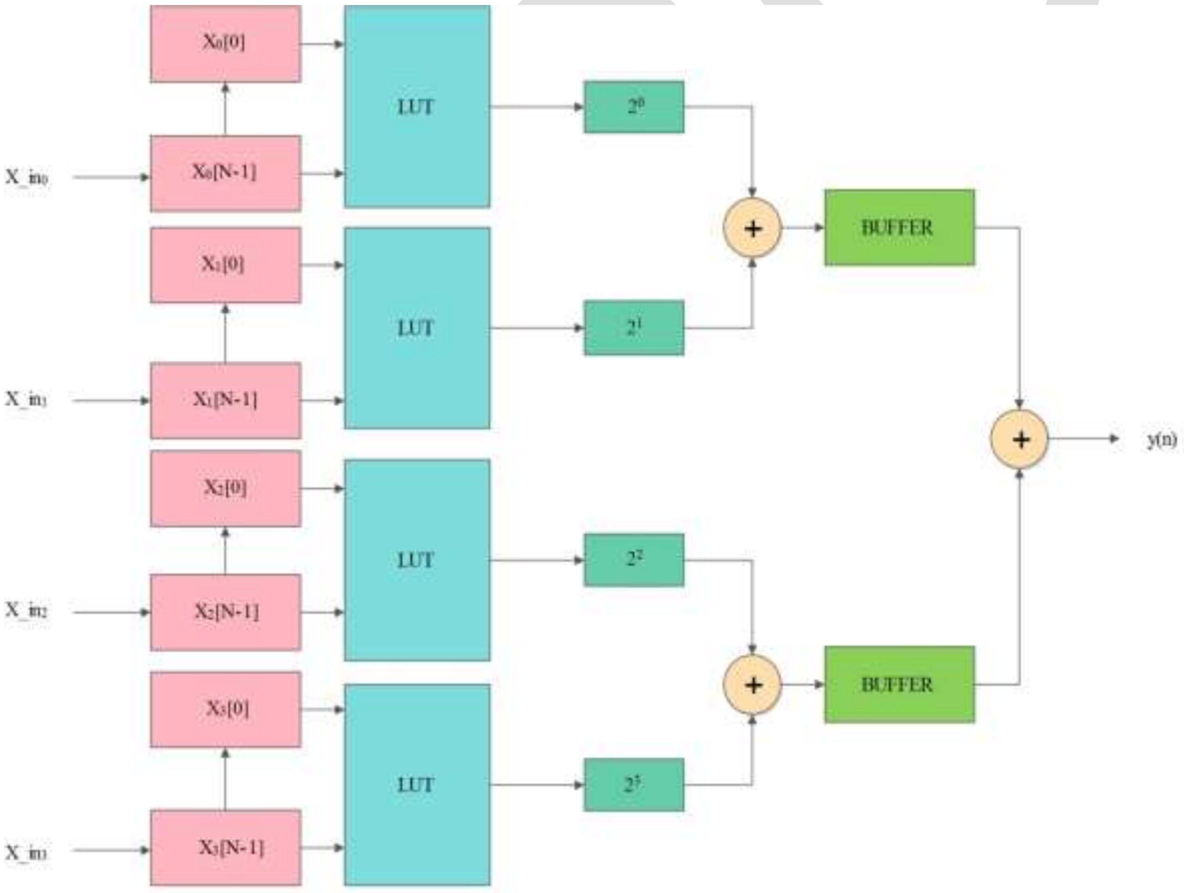

Fig.1. Parallel implementation of DA FIR filter

4. SIMULATION RESULTS

The digital FIR filter architectures are designed based on the existing DA technique and the proposed scheme which is performed by using parallel implementation. VHDL codes are written for these designs and synthesized using Xilinx ISE design tool of version 14.7. The Family of the device was Spartan 6 and the target device was XC6SLX45T. For the design verification purpose, ISim simulator was used. Firstly, the FIR filter using existing DA technique is designed to analyze the thresholds of FIR filters. Then the optimization in terms of speed of the FIR filter is done. After successful synthesis the functional simulation is carried out with the ISim simulator. The results for various architecture of the FIR filter used in this thesis are optimized in terms of speed of the filter. The filter architectures are designed for filter length-5. The input bit precision is of 4 bits for simulation purpose.

Table .1: Comparative analysis of resources utilization

Parameters	**Conventional FIR Filter**	**Signed DA FIR Filter**	**Unsigned DA FIR Filter**	**Proposed Scheme**
Number of Slice Registers	42	38	37	55
Number of Slice LUTs	49	53	50	37
Number of fully used LUT-FF pairs	9	32	37	36
Number of bonded IOBs	17	36	34	12

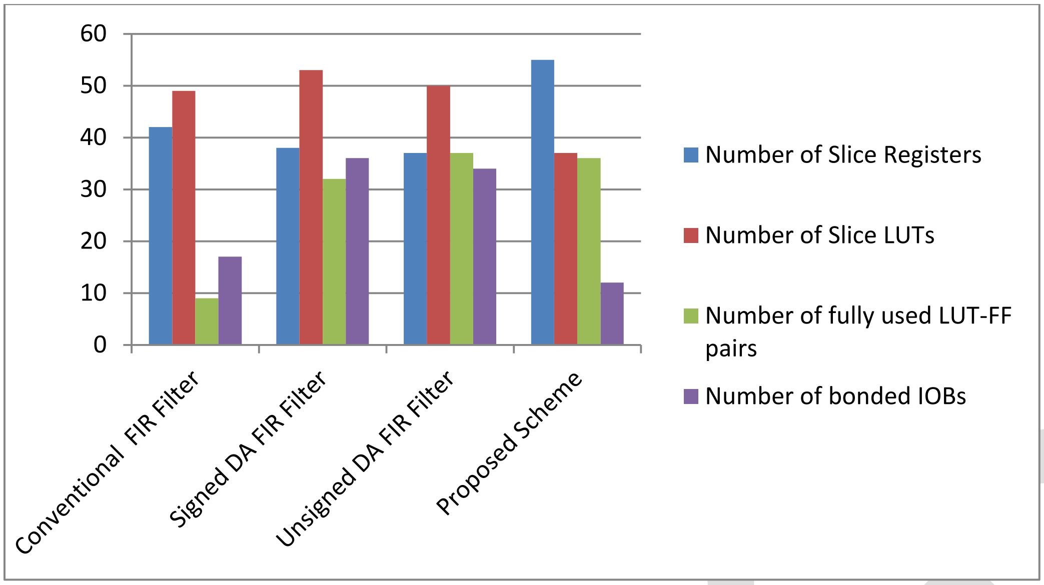

Fig. 2 Comparative analysis of resources utilization

5. CONCLUSION

In this work the optimization of digital FIR filter has been achieved in the direction of operating speed by considering the tradeoff factor between area and speed. In this work the parallel DA technique is proposed to increase the speed of FIR filter at slightly increase the area. The proposed scheme using parallel approach enhanced the operating speed as compare to signed and unsigned DA approach at the cost of slightly increase in the memory requirements

REFERENCES:

[1] C.-F. Chen, “Implementing FIR filters with distributed arithmetic,” IEEE Trans. Acoust., Speech, Signal Process., vol. 33, no. 5, pp. 1318–1321, Oct. 1985.

[2] S. A. White, “Applications of the distributed arithmetic to digital signal processing:Atutorial review,” IEEE ASSP Mag., vol. 6, no. 3, pp. 5–19, Jul. 1989.

[3] H.-R. Lee, C.-W. Jen, and C.-M. Liu, “On the design automation of the memory-based VLSI architectures for FIR filters,” IEEE Trans. Consum. Electron., vol. 39, no. 3, pp. 619–629, Aug. 1993.

[4] K. Nourji and N. Demassieux, “Optimal VLSI architecture for distributed arithmetic-based algorithms,” in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Apr. 1994, vol. 2, pp. II/509–II/512.

[5] H. Yoo and D. V. Anderson, “Hardware-efficient distributed arithmetic architecture for high-order digital filters,” in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Mar. 2005, vol. 5, pp. v/125–v/128.

[6] S.-S. Jeng, H.-C. Lin, and S.-M. Chang, “FPGA implementation of FIR filter using M-bit parallel distributed arithmetic,” in Proc. 2006 IEEE Int. Symp. Circuits Systems (ISCAS), May 2006, p. 4.

[7] Pramod Kumar Meher, Shrutisagar Chandrasekaran, and Abbes Amira ,“ FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic” IEEE transactions on signal processing, vol. 56, no. 7, July 2008.

[8] Pramod Kumar Meher, "New Approach to Look-Up-Table Design and Memory-Based

Realization of FIR Digital Filter" IEEE transactions on circuits and systems, vol. 57, no. 3, march 2010.

[9] Michał Staworko, Mariusz Rawski, "Application of Modified Distributed Arithmetic Concept in FIR Filter Implementations Targeted at Heterogeneous FPGAs" ISSN 0033-2097, R. 88 NR 6/2012

IJERGS