Design and Implementation of High Speed FIR filter Based on Common Subexpression Elimination.

Aaisha Siddiqa G.D, Dr. S. Jagannathan, Dr. M. Uttara Kumari

Department of Electronics & Communication, RVCE, Bangalore-59;8123805787; and Professor, Department of Electronics & Communication, RVCE, Bangalore-59; 09449765873.
Professor and HOD, Department of Electronics & Communication, RVCE, Bangalore-59;91 - 080-6717 8021.

ABSTRACT

This paper presents an area-efficient implementation of a High Speed Digital Finite Impulse Response Filter. Here, the problem of Optimizing the area and delay has been addressed. Since a large number of constant multiplications dominate the complexity of many digital signal processing applications, in the proposed method, constant multiplications are replaced by addition/subtraction and shift operations. Based on the design objective of Low complexity and High Speed, the addition/subtractions are implemented using fast Ripple Carry adders and Carry Save Adders. Furthermore, high-level algorithms are used to reduce the complexity of adders used in the design. Thus, the experimental results on FIR filter instances can find better FIR Filter designs in terms of area and delay than those implemented using efficient general multipliers.

Key words: filter, CSA, RCA, High Level Algorithms, Optimization

Corresponding Author: Dr. S. Jagannathan

INTRODUCTION

Finite Impulse Response Filters are widely used due to their high stable nature. The phase characteristics of these filters are also linear, which adds to the advantage. The performance of an FIR filter depends to a larger extent on the multiplier blocks. Multiplication of filter coefficients with the sampled input signals are termed as Multiple Constant multiplications. FPGA features that support design reconfigurability, parallel structures and arithmetic algorithms that minimize the system resources and enhance the design performance allow it to be the choice for implementing DSP structures.

In this paper, Direct-form I based Low pass FIR filter is implemented. This approach gives better performance in terms of cost; and most importantly speed of operation. The main idea is to replace constant multiplications which are complex with addition/subtraction and shift operations. FIR filters are initially implemented using shift-add multiplier architectures. The basic adder blocks used are Ripple Carry Adder (RCA) and Carry Save Adder (CSA). In

order to enhance the performance of RCA blocks, radix-2 and radix-4 Ling adders have been used.

In order to enhance the performance of the filters, Fast Ripple carry adders based on radix-2 and radix-4 Ling and Jackson's theory, have been designed. These adders improve the performance rate by 10 fold. The major drawback is the increase in area by 2fold (Ling adder) and 3 fold (Jackson adder). Furthermore, Common Sub Expression Elimination and Partial Redundancy Algorithms are used to optimize the design. These algorithms aim at eliminating all the redundant expressions and allows for sharing of common coefficients. The coding scheme used is Canonical Signed Digit and Digit Based encoding.

DESIGN METHODOLOGY

The digital FIR Filter proposed for High-Speed DSP applications is presented in Fig.1 below. Each adder block is implemented either using a Carry Save Adder and/or Ripple Carry Adder.Initially, FIR filter coefficient multiplication is carried out using Wallace Tree and Booth Multiplier. Since the coefficients remain constant, the complexity associated with multiplier blocks are replaced using shift-add multiplier architectures. This reduces the hardware complexity to a greater extent, without affecting the design functionality. The basic adder blocks used in the design of shift-add architectures are CSA and RCA adders. RCA has a major drawback of latency associated with it. In order to enhance its speed, modified radix-2 and radix-4 Ling adders have been designed. These adders perform significantly faster than the existing ripple carry adders, but introduce an area overhead.

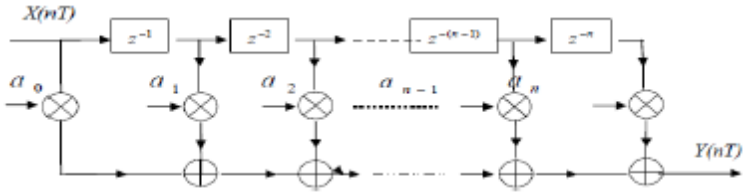

Fig 1. Direct Form 1 FIR Filter structure.

I. SHIFT-ADD MULTIPLIER ARCHITECTURE

The complexity of multipliers can be reduced to a larger extent by replacing it with shift-add architecture. Here, the shifts are represented by only wires in the hardware. The basic adders used are the RCA and CSA. The constant coefficient multiplication can be represented as shown below in the example:

$$10x = (1010)\, x = x << 3 + x << 1$$

Multiplication by a constant of 10 means, shift the input signal to the left by 3 bit positions and 1 bit positions and add the resulting bits. In the above case, only one adder block will be needed, instead of using a large and complex multiplier block. This leads to saving of power and area.

II. LING ADDERS

Ling proposed an algorithm in which the $(i+1)^{th}$ carry output, c_{i+1}, of an adder operating on the binary inputs a and b, may be simplified as follows:

$$c_{i+1} = G_{i:0} = g_i + nk_i \cdot g_{i-1} + nk_i \cdot nk_{i-1} \cdot g_{i-2} + \cdots$$
$$= nk_i \cdot g_i + nk_i . g_{i-1} + nk_i \cdot nk_{i-1} \cdot g_{i-2} + \cdots$$
$$= nk_i (g_i + g_{i-1} + nk_{i-1} \cdot g_{i-2} + \cdots)$$
$$= nk_i \cdot H_{i:0}$$

Here nk_i: not kill term, g_i is the generate and $G_{i:0}$ is the group generate terms. This group generate term is obtained by multiplying all the generate terms from bit position i downto 0. Ling also showed that the corresponding sum bit, s_{i+1} are derived as:

$$s_{i+1} = p_{i+1} \oplus c_{i+1}$$

$$= p_{i+1} \oplus G_{i:0}$$

$$= p_{i+1} \oplus (nk_i \cdot H_{i:0})$$

$$= H_{i:0}\langle p_{i+1} \oplus nk_i \rangle + !H_{i:0} \cdot p_{i+1}$$

Where p denotes propagate, defined as $p_i = a_i \oplus b_i$.

The above equation represents $H_{i:0}$ as the select line of the multiplexer (2:1). This effectively eliminates XOR gate from the critical path of the adder. The main reason behind this logic is that, an XOR gate introduces larger delay than any other logic gate leading to increase in the delay.Thus, replacing most of the XOR2 gates with the NAND2 gate across the ripple carry chain can accelerate the speed of the adder.

III. OPTIMIZATION USING HIGH LEVEL ALGORITHMS

Logic optimization has a direct impact on the macro-cell design styles using programmable-logic arrays (PLAs). The increasing trend towards high performance and portable systems has forced researches to come up with the innovative system level design techniques that can achieve these objectives and meet the strict time to market requirements.

The unfortunate drawback of the flexibility in implementing logic combinational functions is the difficulty in modelling and optimizing the networks themselves. This section describes one of the most basic conventional optimization techniques Common Sub-Expression Elimination (CSE).

A. COMMON SUBEXPRESSION ELIMINATION

The extraction of common subexpression relies on the search of common divisors of two or more expressions. A common subexpression of two functions associated with the vertices can be extracted by creating a new vertex associated with the subexpressions. The variable associated with the new vertex allows us to simplify the representation of the two functions by replacing with the common subexpression.A reason for extracting the common subexpression is to simplify the overall network by exploiting the commonalities.

B. FIR FILTER USING RIPPLE CARRY ADDERS

One of the most widely used techniques for implementing shift/add architectures is by the use of Digit Based Encoding. Here each constant coefficient will be encoded in the binary format. For each 1 in the binary representation of the constant, it will left shift the bit according to the bit position. For example, $15x = 1111_{(2)}$ x = x <<3 + x<<2 +x<<1 +x
So in the above example four adder blocks are needed that will sum up all the shifted bits. So, the final result will be available after 4 clock pulses.

Digit based encoding does not allow for sharing of partial products among the constant coefficients. As a result, optimized number of additions are not obtained from this technique. In order to enhance the speed performance of Ripple carry adders, modified radix-2 Ling adders are used. This speeds up the computation by 10%. This also leads to an increase in area.

C. FIR FILTER DESIGN USING CARRY SAVE ADDERS

Carry save adders are much faster than the ripple carry adders, even though they have a larger area. Since the performance of DSP processors forms the bottleneck of the design, an area overhead can always be acceptable.

In this case, Digit based encoding along with Canonical Signed Digit based encoding both are used. Canonical Signed Digit is a number system that uses two's complement representation for encoding a floating point value. Due to its efficiency, this scheme is most commonly used to add/subtract networks in hardwired Digital Signal Processing. This representation uses a sequence of one or more symbols, -1, 0, and 1 with each position possibly representing the addition or subtraction of a power of 2.

The conversion begins with the least significant bit and moves to the most significant bit. The multiplication of each filter coefficient with x[n] is performed by shift and adds, where "<<" represents the arithmetic shift left operation and represents the multiplication factor of -1.For the filter coefficient say 281_{10} = $0100011001_{(2)}$, using CSD, the encoded value is given by $0100101\bar{1}001$.

Advantages of implementing filter using Canonical signed digit is that, it reduces the number of adder blocks required for computation, which implies the overall circuit complexity is drastically reduced. This technique proves to be the best only in the best case where all the bits are high i.e., logical 1, hence providing maximum efficiency.

IV RESULTS AND DISCUSSION

The circuit proposed is simulated using Xilinx simulator with certain inputs (8, 9 and 10) and for impulse input the corresponding input and output waveform are shown in Fig

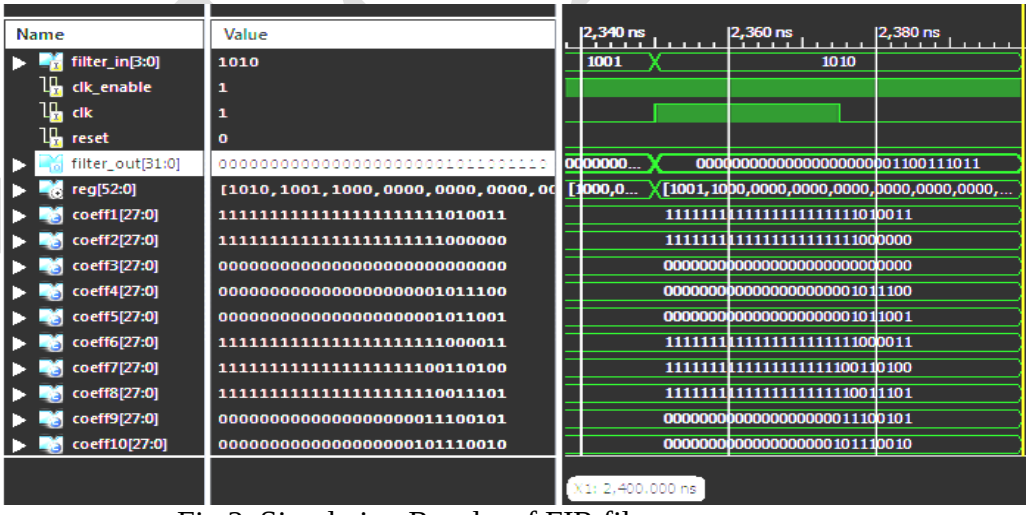

Fig 2. Simulation Results of FIR filter structure.

Fig2 gives comparative analysis of all the adders i.e., ripple carry adder and radix-2 and radix-4 ling and Jackson adder. From the graph, it clearly states that the area of ripple carry adder is the least while all the higher radix adders contribute an increase in the circuit complexity. This is still acceptable as the performance as higher radix adders are significantly faster than ripple carry adder.

Radix-4 Jackson adder has the highest speed although it accounts for a large area. All the higher radix adders output the results at an approximately 4times faster than the existing the adder circuits.

The graphical plot indicated in the figure3 gives the delay and area associated with the FIR Filter implemented using Carry save adder and Canonical Signed digit encoding based FIR filter design.

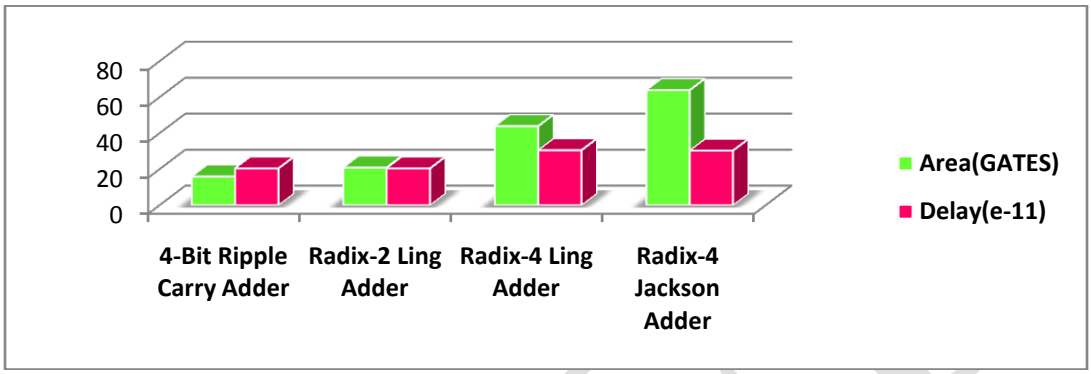

Fig 3. Comparative analysis of various adders

Filters implementing Signed digit based encoding are significantly faster than those with just carry save adders. The above graph shows the best case wherein all the input bits are 1's. in such a case SD encoding proves to be the best.

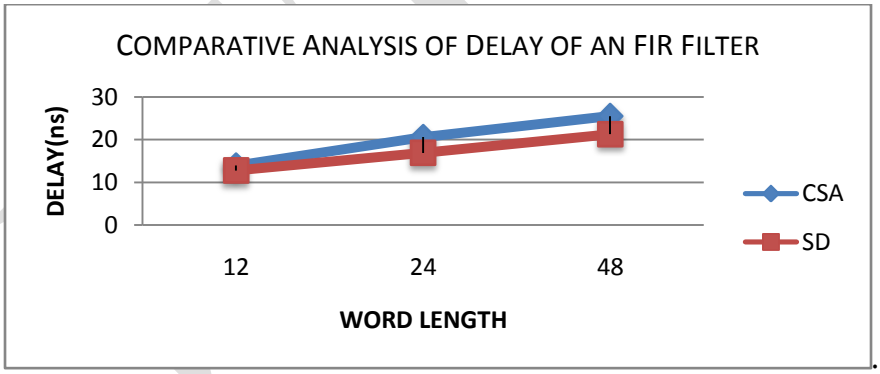

Fig 4. Graphical Plot of an 16th Order FIR Filter designed using Carry Save Adder and Optimized FIR using Canonical Signed Digit-Area Analysis

CONCLUSION

In the proposed design, multiplication by constant coefficients is replaced with shift-add architecture. These shift add multiplier circuits reduce the hardware complexity, as shifting is implementing using only wires, implying no extra logic needed.

Moreover, the basic adders used are ripple carry and carry save adders which also determine the speed of filter. Simulation results indicate that the Carry save adders are faster than ripple carry adders. Couple of enhancements is done to this design. Firstly, use of higher radix Ling and Jackson adders which increase the speed of the adder drastically by about 5 times than the simple adders. The major drawback associated with this is the increase in the hardware complexity. Second, Canonical

Signed digit based encoding is used. This coding scheme coupled with the sharing of common coefficient subexpressions reduces the number of adder blocks required in shift-add architectures. This in turn results in drastic reduction in power consumption.The area overhead is reduced by 50% for best case with an enhanced speed of 10 fold. But in the worst case, area is increased by 10% with a speed enhancement by 2 fold.

REFERENCE

[1] R.Uma and Jebashini Ponnian, "Systolic FIR Filter Design with Various Parallel Prefix Adders in FPGA, Performance Analysis," International Symposium on Electronic System Design, 2012.
[2]Kamaraj, C. K. Sundaram, and J. Senthilkumar, "Pipelined FIR Filter Implementation using FPGA", International Journal of Scientific Engineering and Technology, vol. 1, no. 4, Oct. 2012, pp. 55-60.
[3]V.Sudhakar, N.S.Murthy, and L.Anjaneyulu, "Area Efficient Pipelined Architecture for Realization of FIR Filter Using Distributed Arithmetic", International Conference on Industrial and Intelligent Information, 2012, pp. 169-173.
[4] R. P. Rajput, and M. N .S Swamy, "High speed Modified Booth Encoder multiplier for signed and unsigned numbers", International Conference on Modelling and Simulation, 2012, pp. 649-654.
[5] B. Rashidi, B. Rashidi, and M. Pourormazd, "Design and Implementation of Low Power Digital FIR Filter based on low power multipliers and adders on Xilinx FPGA", International Conference on Electronics Computer Technology, 2011, pp. 18-22.
[6] N. S. Pal, H. P. Singh, R. K. Sarin, and S. Singh, "Implementation of High Speed FIR Filter using Serial and Parallel Distributed Arithmetic Algorithm", International Journal of Computer Applications, vol. 25, no. 7, Jul 2011, pp. 26-32.
[7]. V. Soni, P. Shukla, and M. Kumar, "Application of Exponential Window to Design a Digital Nonrecursive FIR Filter", International Conference on Communications and Signal Processing, 2011,pp. 1015-1019.
[8] W. Yunlong, W. Shihu, and J. Rendong, "An Extreme Simple Method for Digital FIR Filter Design" in proc. Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), IEEE, 2011, pp. 410-413.
[9] J. Li, M. Zhao, and X. Wang, "Design and Simulation of 60-order Filter Based on FPGA" in proc. International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), IEEE, 2011, pp. 113 – 115**.**
[10]Diego Jaccottet and Eduardo Costa, "Design of Low-Complexity and High-Speed Digital Finite Impulse Response Filters", *2010 18th IEEE/IFIP International Conference on VLSI and System-on-Chip (VLSI-SoC 2010)*
[11] X. Jiang, and Y. Bao, "FIR filter design based on FPGA", International Conference on Computer Application and System Modeling, 2010, pp. 621-624.