

FPGA Realization of MUX Based FIR Filter Architecture

K.Sabitha¹, S.Sreehari²

¹ PG Scholar, Department of ECE, MVSR Engineering College, Hyderabad, India

kondamsabitha@gmail.com

² Assistant Professor, Department of ECE, MVSR Engineering College, Hyderabad, India

Abstract

In digital filter implementation, the multiplier usage is avoided by using MUX based multiplier and Look up Table (LUT) based multiplier. Odd multiple storage proposed by stores the product of odd multiple of co-efficient and the input. The advantage of storing odd multiple is that even multiples can be obtained by a simple left shift operation. MUX based multiplication is carried out only with MUX and shifters and CSLA BEC adder. Proposed MUX based multiplication is carried out only with MUX and shifters and CSLA DLATCH adder. Proposed MUX based CSLA DLATCH adder increases the speed over MUX based CSLA BEC. The designs've been implemented using Verilog and synthesized using Xilinx 13.2 Spartan 3e kit.

Keywords: *FIR filter, Look-up Table, Reconfigurable Architecture, and Distributed Arithmetic.*

1. Introduction

Finite Impulse Response (FIR) digital filter is widely used as a basic block in signal and image processing applications. The number of multiply-accumulate (MAC) operations required per filter output increases linearly with the order of filter, but implementation of higher order filters in real-time is another challenging task. Recently with the advent of software defined radio (SDR), the research has been concentrated on realization of FIR filter [1] mainly due to its high flexibility and low complexity [5] [6] [7]. The digit-based reconfigurable architecture presented in [3] provides a flexible and low power solution with a wide range of precision and tap length of FIR filters. Conventionally, the FIR filters are designed based on programmable multiply-accumulate [MAC] architecture [6] and systolic architecture [7]. The performances of the designs are analyzed in terms of hardware complexity, power consumption and throughput. In [6], the programmable MAC architectures consume low power with reduced supply voltage and it requires large area. In [7], even though systolic based architecture reduces the complexity, it increases the latency when the order of the filter increases. Several attempts have been made and it continued to develop low-complexity dedicated VLSI systems for these filters.

There are several issues in the hardware implementation of Digital filters. The direct implementation of N-tap FIR filters which requires N MAC operations are too expensive to implement in hardware due to its logic complexity and area requirement. Distributed Arithmetic (DA) and LUT constitute memory-based techniques. Among them, DA is applied for inner product computation [8]-[17], LUT provides computation of multiplication [18]-[24]. In the LUT based approach, multiplications of input values with a fixed coefficient and consisting of all possible pre-computed product values corresponding to all possible values of input multiplicand, while in the DA-based approach, an LUT is used to store all possible values of inner-products of a fixed-point vector. If the inner-products are implemented directly, the memory-size of LUT-multiplier based implementation increases exponentially with the input word length, whereas the memory size of the DA-based approach increases exponentially with the inner-product-length. It is observed that the memory-size reduction is achieved by such decompositions is accompanied by raise in latency as well as the number of adders and latches.

II. MULTIPLIER LESS STRUCTURES BASED FOR FIR FILTER

In this paper FPGA realization of the two FIR filter architectures such as Odd multiple storage scheme

[4] and MUX based multiplier are discussed. Both Odd multiple storage scheme and MUX based multiplier have multiplier less architecture to reduce complexity in the design, by manipulating the odd multiples of the fixed coefficient in LUT design, and general multipliers replaced by shift and add operations respectively. The performances of the proposed architectures are analyzed in terms of area and time complexities by varying the number of taps.

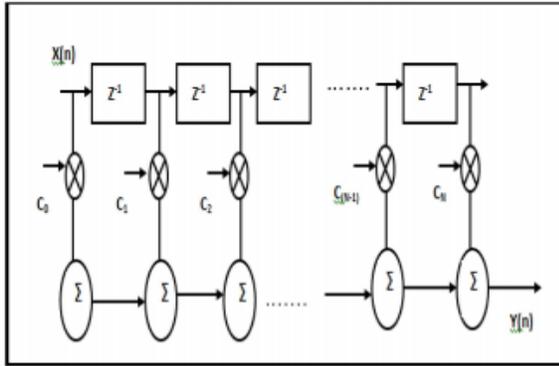


Fig.1. General Structure of an FIR filter

Figure 1 shows an implementation of N tap FIR filter with the usage of three elements namely adders, multipliers and delay elements. Let X (n) and Y (n) are the input and output sequences of the FIR filter respectively. Consider an N-tap FIR filter that can be formulated as

$$Y(n) = \sum_{k=0}^{N-1} h_k X(n-k) \quad (1)$$

Where h_k is the k th coefficient of the filter impulse response.

In the above equation, the filter coefficients do not vary with regard to the input signal or the noise. This implies that the coefficients are fixed. Such filters whose coefficients are fixed are called as fixed filters or filters with fixed characteristics. In the filter the MAC structure and delay blocks are the main building blocks. The performance of the filter can be enhanced by the introduction of two architecture schemes which reduces the complexity and critical path. The two schemes discussed here are ODD and MUX based multiplier.

III. ODD MULTIPLE STORAGE SCHEME

This method proposed by [4] stores the product of odd multiple of co-efficient and the input. The advantage of storing odd multiple is that even multiples can be obtained by a simple left shift operation. For any input vector of N bits the number

of locations in the LUT is identified by $2N$. However, the odd multiple storage scheme reduces the number of locations by a factor of 2. Considering an input vector of 8 bits, uses only 128(27) unlike 256 locations of conventional LUT. Table I gives the complete insight about the look up table of odd multiple storage scheme. The 7-bit address (A6-A0) given in Table I identifies the stored values of LUT and it also gives a complete detail about the number of shifts to be performed for the corresponding input bits Hence for the input bits '10' the LUT stored value A is shifted one time to get 2A. The control bits c_2, c_1, c_0 decides the number of shifts to be performed. The Figure 2 gives the proposed structure of the odd multiple storage schemes.

The architecture has an 8 to 7 bit encoder which maps the 8-bit input word ($x_7: x_0$) to a 7 bit address (A6: A0) according to the logical relations. It chooses one output address value depending upon the output obtained from the encoder. The value obtained from the memory array is an odd multiple value of the coefficient. In order to obtain the even multiple values, the shift operation is to be performed. The control unit generates the control bits according to the relations 3a to 3c

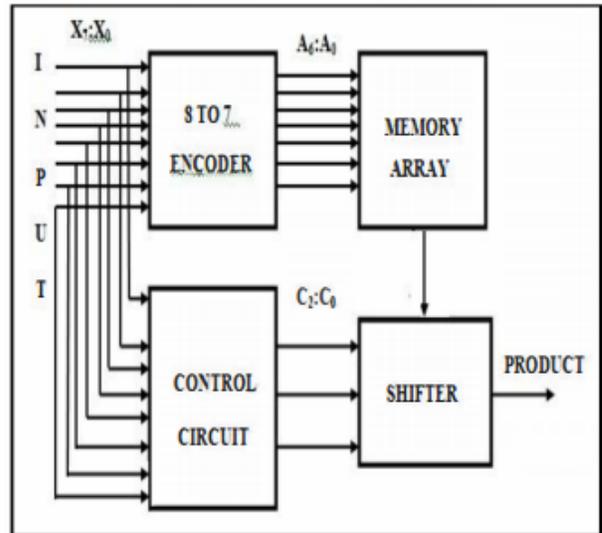


Fig.2. Proposed architecture for odd multiple storage scheme

Sym bol	Address A ₂ A ₁ A ₀ A ₂ A ₁ A ₀	Stored value	Input X ₇ X ₆ X ₅ X ₄ X ₃ X ₂ X ₁ X ₀	Product value	# of shifts	Control C ₂ C ₁ C ₀
P ₀	0	A	1	A	0	0
			10	2A	1	1
			100	4A	2	10
			1000	8A	3	11
			10000	16A	4	100
			100000	32A	5	101
			1000000	64A	6	110
P ₁	1	3A	11	3A	0	0
			110	6A	1	1
			1100	12A	2	10
			11000	24A	3	11
			110000	48A	4	100
			1100000	96A	5	101
			11000000	192A	6	110
P ₁₂₁	1111001	243A	11110011	243A	0	0
P ₁₂₂	1111010	245A	11110101	245A	0	0
P ₁₂₃	1111011	247A	11110111	247A	0	0
P ₁₂₄	1111100	249A	11111001	249A	0	0
P ₁₂₅	1111101	251A	11111011	251A	0	0
P ₁₂₆	1111110	253A	11111101	253A	0	0
P ₁₂₇	1111111	255A	11111111	255A	0	0

Table.1. Odd multiple storage scheme

IV. PROPOSED MUX BASED MULTIPLIER WITH CSLA BEC

Figure 3 shows a shift and add multiplier implemented with the usage of namely adders, and MUX. Since the general multipliers are replaced by MUX, shifters and adders, referred to as multiplier less implementation. Consider an eight bit multiplication of signed numbers. Here the multiplication is carried out only with MUX and shifters. For example consider two signed 8-bit numbers x and c bits, represented in two's complement form.

Table II shows the binary input X is given in powers of 2. As in equation (8) the number C is left shifted based on the given input bits. If MSB bit of input is 1, the shifted value is twos complemented. These shifted values are given through MUX and the corresponding outputs are selected based on the input bits. Finally MUX outputs are added to obtain the product.

CSLA BEC

The structure of the proposed 16-b SQRT CSLA using BEC for RCA with Cin=1 to optimize the area and power is shown in Fig. 4. We again split the structure into five groups. The delay and area estimation of each group 2 is shown in Fig. 5. One input to the mux

goes from the RCA with Cin=0 and other input from the BEC

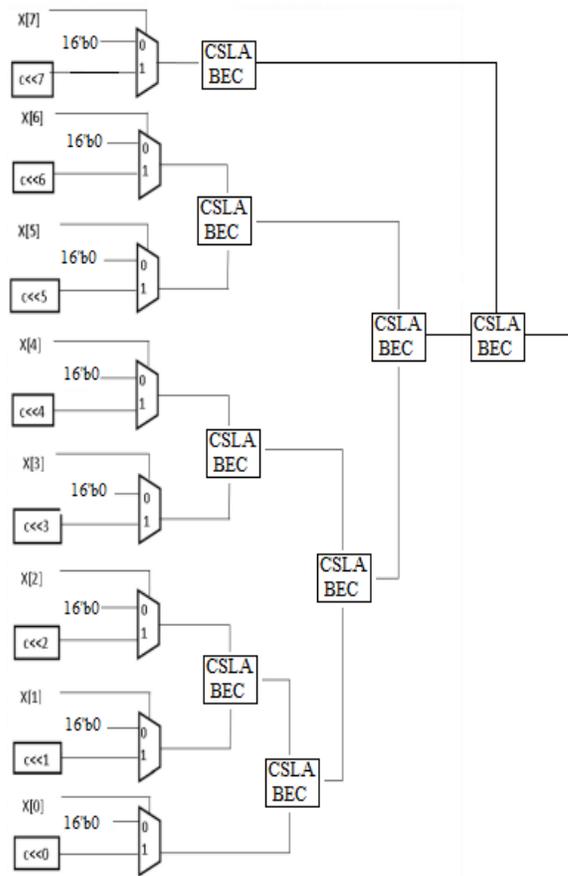


Fig.3. Proposed MUX based Multiplier architecture

Input, X	P = X.C	X in powers of 2
0001	1C	2 ⁰
0010	2C	2 ¹
0011	3C	2 ¹ +2 ⁰
0100	4C	2 ²
0101	5C	2 ² +2 ⁰
0110	6C	2 ² +2 ¹
0111	7C	2 ² +2 ¹ +2 ⁰
1000	-8C	-2 ³
1001	-7C	-2 ³ +2 ⁰
1010	-6C	-2 ³ +2 ¹
1011	-5C	-2 ³ +2 ¹ +2 ⁰
1100	-4C	-2 ³ +2 ²
1101	-3C	-2 ³ +2 ² +2 ⁰
1110	-2C	-2 ³ +2 ² +2 ¹
1111	-1C	-2 ³ +2 ² +2 ¹ +2 ⁰

Table.2.MUX based multiplier

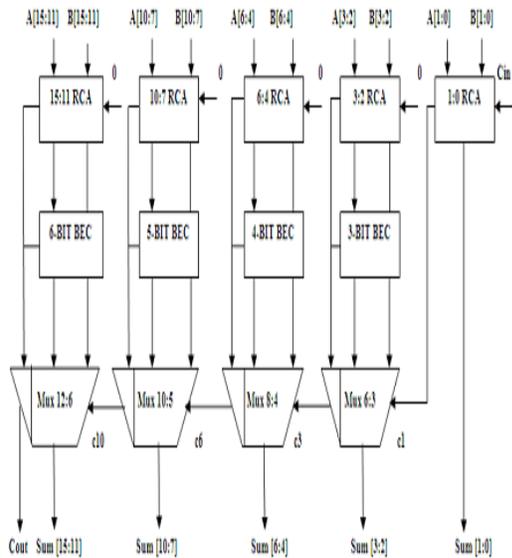


Fig. 4 CSLA using BEC

V. PROPOSED MUX BASED MULTIPLIER WITH CSLA D-LATCH

This method replaces the BEC add one circuit by D-latch with enable signal. Latches are used to store one bit information. Their outputs are constantly affected by their inputs as long as the enable signal is asserted. In other words, when they are enabled, their content changes immediately according to their inputs. D-latch and it's waveforms are shown in Fig.5 The architecture of proposed 16-b CSLA is shown in Fig. 6. It has different five groups of different bit size RCA and D-Latch. Instead of using two separate adders in the regular CSLA, in this method only one adder is used to reduce the area, power consumption and delay. Each of the two additions is performed in one clock cycle.

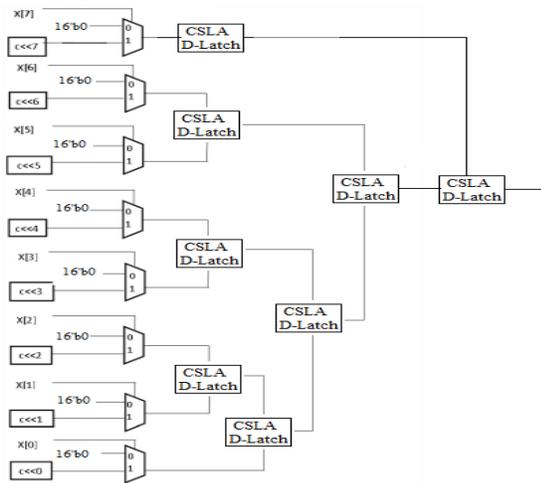


Figure 5. MUX based multiplier with CSLA D-latch

This is 16-bit adder in which least significant bit (LSB) adder is ripple carry adder, which is 2 bit wide. The upper half of the adder i.e., most significant part is 14-bit wide which works according to the clock. Whenever clock goes high addition for carry input one is performed. When clock goes low then carry input is assumed as zero and sum is stored in adder itself. From the Fig. it can understand that latch is used to store the sum and carry for $C_{in}=1$ and $c_{in}=0$. Carry out from the previous stage i.e., least significant bit adder is used as control signal for multiplexer to select final output carry and sum of the 16-bit adder. If the actual carry input is one, then computed sum and carry latch is accessed and for carry input zero MSB adder is accessed. Cout is the output carry.

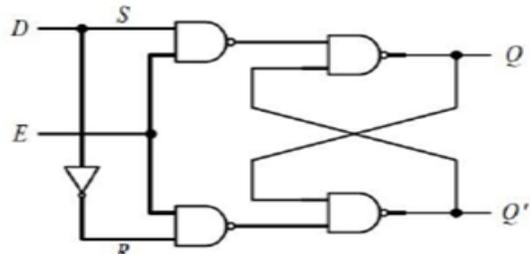


Fig.6 .D-latch

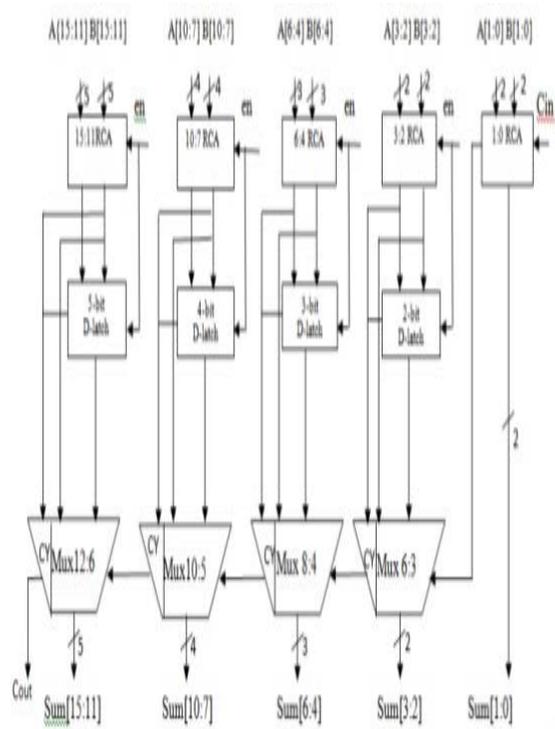


Fig.7.CSLA using D-LATCH

VI. CONCLUSION

In this paper an efficient Low complexity based FIR filter architecture using MUX based multiplier and

odd multiple scheme have been discussed and implemented effectively. Odd multiple storage proposed by stores the product of odd multiple of co-efficient and the input. The advantage of storing odd multiple is that even multiples can be obtained by a simple left shift operation. MUX based multiplication is carried out only with MUX and shifters and CSLA BEC adder. Proposed MUX based multiplication is carried out only with MUX and shifters and CSLA DLATCH adder. Proposed MUX based CSLA DLATCH adder increases the speed over MUX based CSLA BEC. The designs've been implemented using Verilog and synthesized using Xilinx 13.2 Spartan 3e kit.

VII. REFERENCES

1. Digital Signal Processing solution: "Designing for Optimal Results High-Performance DSP using Virtex-FPGAs," Xilinx corporation, pp. 99-103, 2005
2. David V. Anderson, and ErhanOzalevli, "A Reconfigurable Mixed Signal VLSI implementation of Distributed Arithmetic Used for Finite Impulse Response Filtering," IEEE Transactions on Circuits and Systems-I: Regular Papers, Vol. 55, No. 2, March 2008.
3. Kuan-hung, and Tzi-Dar, "A low power digit based Reconfigurable FIR Filter," IEEE Transactions on circuits and systems, Vol. 53, Aug 2006.
4. Pramod Kumar Meher, "New Approach to Look Up Table Design and Memory-Based Realization of FIR Digital Filter," IEEE Transactions on circuits and systems irregular papers, Vol. 57, No. 3, March 2010.
5. R.Mahesh, and AP Vinod, "New Reconfigurable Architectures for implementing FIR Filter with low complexity," IEEE Transactions on computer aided design of integrated circuits and systems, Vol. 29, Feb 2010.
6. T. Solla, and O. Vainio, "Comparison of programmable FIR filter architectures for low power," in Proc. of 28th European Solid State Circuits Conference, pp. 759-762, September 24 - 26, 2002
7. Pramod Kumar Meher, "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. A Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital filter for PCM encoded signals," U.S. Patent 3 777 130, Dec. 4, 1973.
9. 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.
10. Xilinx Incorporation, "The Role of Distributed Arithmetic in FPGA based signal Processing," Xilinx application notes, San Jose, CA

AUTHOR'S BIOGRAPHY:



K.Sabitha has completed her B.tech in Electronics and Communication Engineering from Malla Reddy Institute of Technology, J.N.T.U.H Affiliated College in 2014. She is pursuing her M.Tech in Embedded Systems and VLSI Design from MVSR Engineering College, O.U Affiliated College.



S.Sreehari is Assistant Professor at MVSR Engineering College, Hyderabad in ECE Department. He received his B.Tech degree in Electronics and Communication Engineering from Srikottam Tulasi Reddy Memorial College of Engineering, Hyderabad and M.Tech degree in Embedded Systems from Mahaveer Institute of Science and Technology, Hyderabad. His research interest is Digital Electronics.