

# Low Latency And Less Power Dissipation Of A 4:2 Compressor Based Distributed Arithmetic Unit FIR Filter Design

S.Karunakaran<sup>1</sup> and S.Rukmanidevi<sup>2</sup>

<sup>1</sup> Department of EEE, RMD Engineering College, Kavaraipettai-601 206,  
Chennai, Tamilnadu, India

<sup>2</sup> Department of CSE, RMD Engineering College, Kavaraipettai-601 206,  
Chennai, Tamilnadu, India

Email:<sup>1</sup>karuna\_vlsi@rediffmail.com, <sup>2</sup>rukmani.ece@rmd.ac.in

## ABSTRACT

This paper presents a design of low latency and less power dissipation of distributed arithmetic FIR filter architecture. In the existing method, FIR filter is implemented using two operand adders based Distributed arithmetic (DA) units, which have less speed due to long carry propagation and also consumes more power due to the generation of more glitches. In order to overcome these issues, three different types of DA FIR filter architectures are proposed using 4:2 compressors, namely XOR-MUX based 4:2 compressors, pure MUX based 4:2 compressors, and XOR-XNOR based 4:2 compressors respectively. The design simulation is performed by using HSpice in Cadence design tool. Speed and power estimation were obtained through transistor level design simulation. By increasing the filter tap size also, the simulations are performed and the results are compared based on delay and power dissipation. From the simulation result, XOR-XNOR using 4:2 compressor based DA unit FIR filter outperforms the other proposed FIR filter design.

**Keywords:** Distributed arithmetic unit, two operand adders, 4:2 compressors, Multiplierless architecture.

## 1.0 INTRODUCTION

DSP systems find applications in many areas such as digital audio, video compression, speech processing, portable video systems and computers. With the increasing number of DSP applications, there is a need for low power and high speed

design of VLSI architecture for DSP systems. In multiplier based FIR filters, MAC block is an important block along with other DSP functions.

FIR filters require KMAC blocks for the implementation, where K is the number of taps. This type of implementation requires more area. Multiplierless architectures were proposed to resolve this issue according to the way in which the manipulation of filter coefficients and multiplierless architectures are classified into two categories. The first technique is the conversion based technique. In this technique, high performance hardware is implemented by transforming the coefficients to other forms of numeric representation. Canonic Sign Digit (CSD) is an example of the conversion based technique. In the CSD method, multipliers can be designed using adders/subtractors and shifters by transforming the filter coefficients into combinations of powers of two [12]. The Dempster-Mcleod (DM) method is also an example of a conversion based approach. In this method, formed partial products are cascaded to reduce the usage of adders. The second type of multiplierless technique is called the Distributed Arithmetic (DA) method [2, 3, 4, 5, 6, 8, 9, 10, 11]. The DA technique utilizes RAMs, ROMs and Look-Up tables (LUTs) for storing the pre-computed values of coefficients. The distributed arithmetic method is the best technique for implementing LUT based FPGA architectures.

LUT based multiplierless FPGA architecture was initially proposed by Croisier et al. [13] and involves partitioning of partial term functions made by utilizing two's complement representation of data. Precomputed partial terms are stored in LUTs. The advantage of this approach is that any bit serial design can be transformed to full parallel implementation. This may enhance the design performance. Since DA implementation requires  $2K$  words ( $K$  being the number of taps of the filter), the memory/LUT requirement increases in an exponential manner when the order of the filter increases. This is one of the main problems while designing higher order FIR filter. Yoo et al. [7, 14] developed an architecture that replaces the LUT requirements with multiplexer/adder pairs. M.Keerthi et al [1] implemented LUT-less DA based higher order FIR filters on an FPGA device. Here the DA base unit is implemented using two operand adders, which are carry save adders. In this approach, area and memory usage is reduced in comparison with the conventional LUT based DA architecture and previous LUT-less DA architecture.

The Distributed Arithmetic (DA) block is comprised of three small units: the shift register unit, the DA base unit, which is constructed using two operand adders, and the adder/shifter unit. In DA unit, computation time is longer since the computation is carried out dynamically on the fly instead of doing it in predefined look up table. Therefore, the addition operation is the limitation for the system speed. Since the output coming from different input paths have uneven delay, which gives rise to the generation of many undesired transitions before the signal settles called glitches, the power consumption is also more. When designing higher order filters, computation time for addition operation also will be increased. Therefore it is essential to reduce the computation time for computing partial products and also power dissipation by introducing compressors for constructing DA base units instead of two operand adders. Three different types of DA FIR filter architectures are

proposed using 4:2 compressor namely XOR-MUX based 4:2 compressors, pure MUX based 4:2 compressors, and XOR-XNOR based 4:2 compressors respectively.

## 2.0 THEORY OF DISTRIBUTED ARITHMETIC UNIT

Distributed Arithmetic is one of the best methods for the implementation of FIR filters on FPGAs, which has high flexibility that permits transformation from serial to full parallel design. Distributed Arithmetic (DA) can be used to develop bit-level architectures for vector-vector multiplications. In distributed arithmetic, each vector word can be expressed as a binary number; the multiplications are mixed and reordered so that the arithmetic unit becomes “distributed” throughout the structure.

Eq. (1) describes an FIR filter of length  $K$ :

$$y[n] = \sum_{k=0}^{K-1} a_k x[n-k] \quad (1)$$

Where:

$x$  and  $y$  are two vectors of size  $K$  that represent the input and transformed data, respectively.  $K$  is the number of taps of the FIR filter.

In a bit-serial DA scheme, assuming that the input  $x$  to the filter is represented in  $L$ -bit 2's complement binary numbers with the sign bit to the left of the radix point, we have:

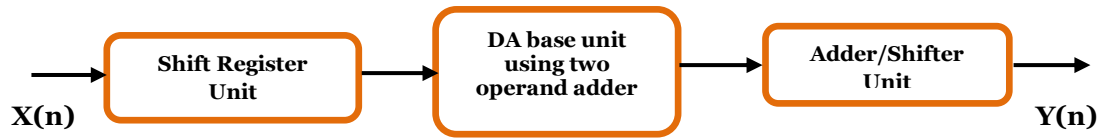
$$x[n-k] = -b_{k,0} \sum_{l=1}^{L-1} b_{k,l} 2^{-l} \quad (2)$$

Replacing the result of Eq. (2) in Eq. (1), we obtain Eq. (3)

$$y[n] = \sum_{k=0}^{K-1} a_k \left( -b_{k,0} + \sum_{l=1}^{L-1} b_{k,l} 2^{-l} \right) \quad (3)$$

$$y[n] = - \left( \sum_{k=0}^{K-1} a_k b_{k,0} \right) + \sum_{l=1}^{L-1} \left( \sum_{k=0}^{K-1} a_k b_{k,l} \right) 2^{-l} \quad (4)$$

The terms inside the brackets in Eq. (4) will take any one of  $2K$  possible combinations, given that  $b \in \{0, 1\}$  and also these values represent all possible sum combinations of FIR filter coefficients. These values are precomputed values that will be stored in memories or LUTS, which may be addressed by  $k, l$ . Thus the MAC algorithm is converted to LUT accesses and summations



**Fig.1. Block diagram of the existing distributed arithmetic FIR Filter**

The block diagram of the existing distributed arithmetic FIR filter is shown in Fig.1. The DA architecture is comprised of three units, namely the shift register unit, the DA base unit, and the adder/shifter unit. In this existing system, distributed arithmetic LUT-less algorithms are implemented using a two operand adder, which is a carry save adder. The carry save adder requirement depends on the order of the filter and the filter coefficient's word length. In the carry save addition concept, the carries are generated during each addition. A pair of operands can be saved and added to the next operand with proper alignment. The carry outputs can be saved and used in the adder in next row. In this case, partial sum and partial carry are saved and passed on to the next row. The time taken for getting a result in two operand adders is proportional to  $\log_2(n)$  where 'n' is the number of inputs. In the two operand adder based DA FIR filters, power consumption is more due to the generation of lot of glitches and due to long carry propagation, the carry save addition process is slow and hence the delay increases for higher order filters. The critical path delay as well as the number of output bits is increased using this logic. In order to overcome these issues, three different types of DA FIR filter architectures are proposed using 4:2 compressors, namely XOR-MUX based 4:2 compressors, pure MUX based 4:2 compressors, and XOR-XNOR based 4:2 compressors respectively.

### 3.0 MATERIALS AND METHODS

The block diagram of the proposed distributed arithmetic FIR filter is shown in Fig. 2. The proposed system consists of three units, namely 1. Shift register unit 2. DA unit using compressors and 3. Adder/Shifter unit.



**Fig.2. Block diagram of the proposed distributed arithmetic FIR Filter**

#### 3.1 Input shift registers

The input shift register unit is used to store the K most recent input samples  $x[n]$  and the data are shifted right during every clock cycle. The output of the input shift

register unit is concatenated, and they become the selection line for 2X1 multiplexers, which are present inside the distributed arithmetic unit.

**3.2 New distributed arithmetic unit based on 4:2 compressors**

Three different types of DA FIR filter architectures are proposed by modifying the architecture of DA unit using 4:2 compressors namely XOR-MUX based 4:2 compressors, pure MUX based 4:2 compressors, and XOR-XNOR based 4:2 compressors respectively.

Local carry compressors are used to produce single sum bits of equal weight from N input bits. During addition operations, N:2 compressors reduce N operands into two, and the sum and carries are kept separate. The operands can be added in parallel without any dependency upon the result of the previous least significant bit. A two-output adder is formed with a time delay, which is also independent of input bit size.

4:2 compressors are used for the design of regularly structured Wallace trees in a low complex manner. 4:2 compressors can be easily constructed by using XORs and Multiplexers (MUX) as building blocks. The advantage of 4:2 compressors is that the previous stage carry will not be propagated to the next stage. When two 4:2 compressors are cascaded, column wise computation can be performed for the partial product. 4:2 compressors could be realized by different combinations of XOR gates, AND gates, and MUXs. Eq. (5), Eq. (6), and Eq. (7) represents the outputs of 4:2 compressor.

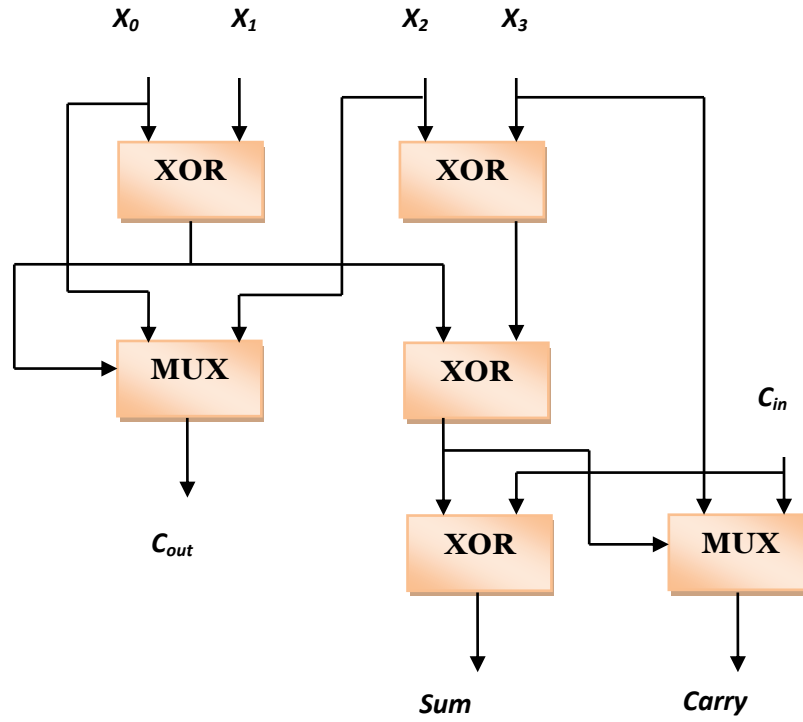
$$S = X_0 \oplus X_1 \oplus X_2 \oplus X_3 \oplus C_{in} \tag{5}$$

$$C = (X_0 \oplus X_3) \cdot C_{in} + S \cdot X_3 \tag{6}$$

$$C_{out} = (X_0 \oplus X_1) \cdot X_2 + X_0 \cdot X_1 \tag{7}$$

where  $X_0, X_1, X_2, X_3, C_{in}$  are the inputs and  $S, C, C_{out}$  are the outputs. Three types of architectures are proposed by logic level decomposition of 4:2 compressors. These types are explained in the following section.

### 3.2.1 XOR-MUX based 4:2 compressor



**Fig.3. Configuration of XOR-MUX based compressor**

The configuration of XOR-MUX based compressor is shown in Fig.3. It is formed by using four 2input XOR gates and two 2:1 MUX gates. The critical path of the compressor is 3 XOR gates.

### 3.2.2. Pure MUX based 4:2 compressors

The configuration of pure MUX based 4:2 compressor is shown in Fig.4. It is formed by six 2:1 MUX gates. All the three outputs (sum, carry, and carryout) are generated using MUX gates. The critical path of the compressor is 3 XOR gates.

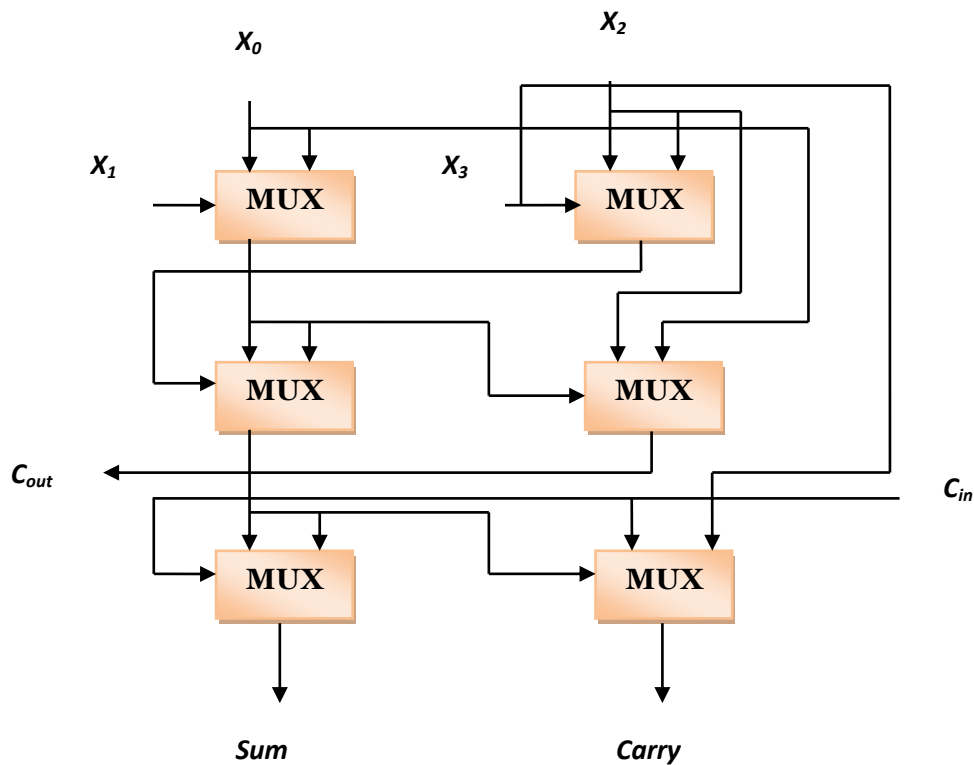
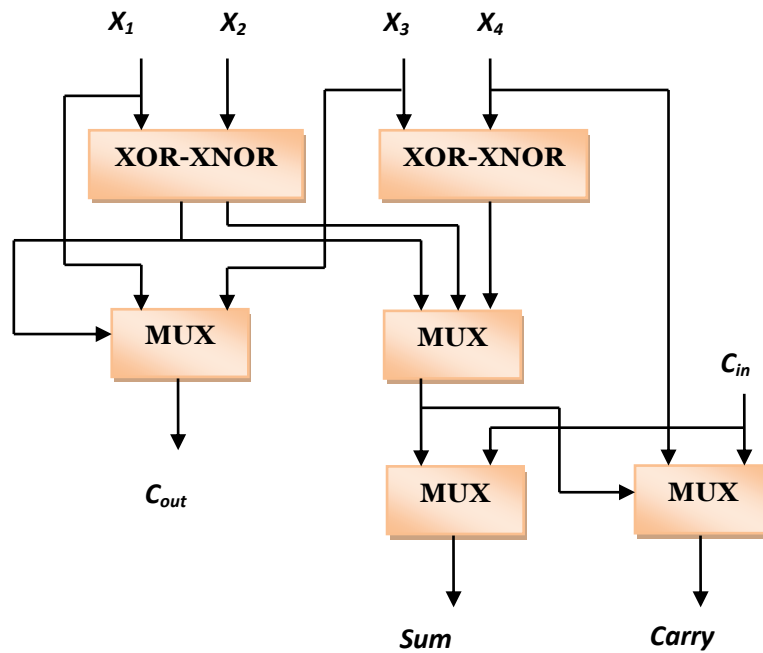


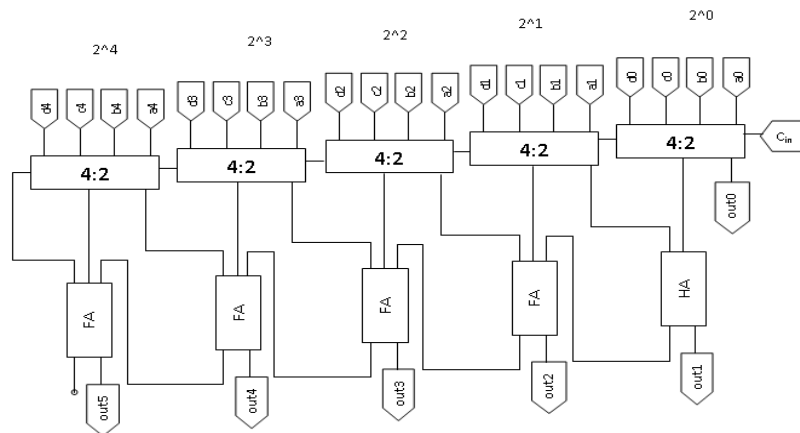
Fig.4. Configuration of pure MUX based compressor

### 3.2.3. XOR-XNOR gate based 4:2 compressors

The configuration of XOR-XNOR gate based 4:2 compressors is shown in Fig.5. The speed of the architecture will be improved when the multiplexers are placed in the critical path. The delay is reduced by the use of computed XOR and XNOR values. Before the inputs arrive, the select bit at the MUX block will be available. So the switching time of the transistors in the critical path is reduced. The critical path delay of the proposed implementation is XOR + 2\*-MUX



**Fig.5. Configuration of XOR-XNOR gates based 4:2 compressor**



**Fig.6. Proposed 4 tap DA base unit using 4:2 Compressor**

In the proposed 4 tap DA base unit shown in Fig. 6, two operand adders are replaced by 4:2 compressor and, due to this, carry propagation delay and critical path delay becomes greatly reduced largely, and so a high speed DA base unit is obtained. After every input bit has been processed by 4:2 compressors, the output signals are then combined using a vector-merging adder to obtain partial term for the corresponding input data.

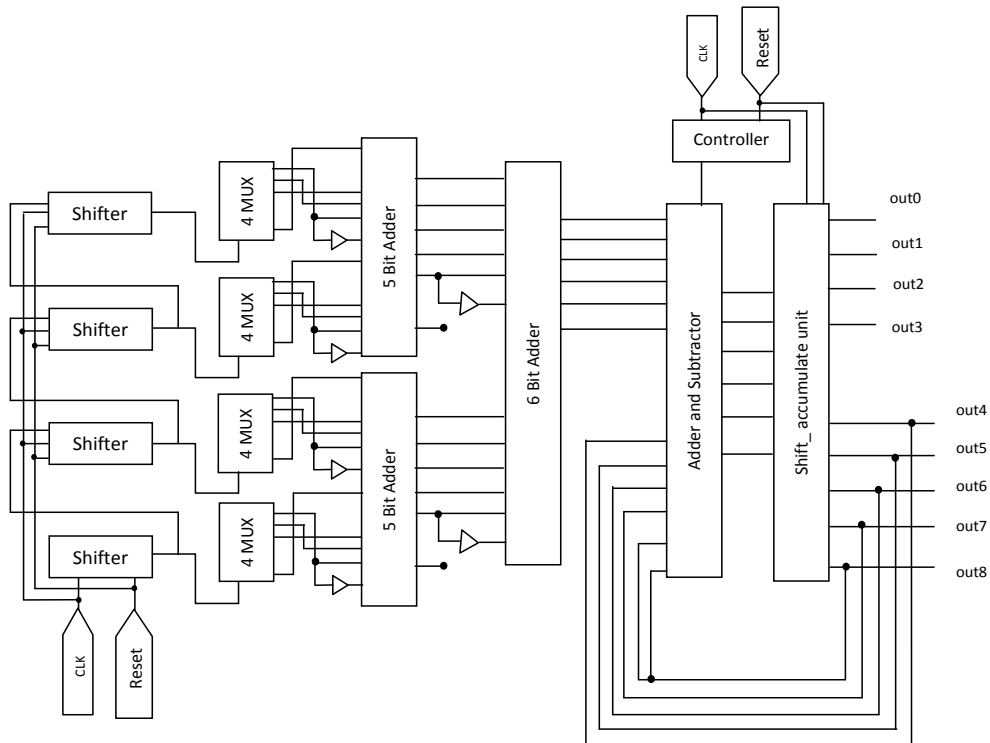


**3.3 Adder/ Shifter**

This stage consists of three units namely adder/subtractor, accumulator, and a shifter. This unit gets a partial term beginning with the Least Significant Bit (LSB) of the input and then shifts this to the right to add it to the next partial result. This involves extra logic to take the LSB of each partial result out and to prepare the result for the summation with the next partial term. Logic is necessary to manipulate the LSB at each iteration. The number of bits in the addition is increased in 1 bit to maintain whole precision of the output. First, the control of the adder/shifter unit should be independent of the input samples stored in the input shifter register unit. Second, that the control of the adder/shifter unit should be independent of the filter coefficients. From the DA FIR Eq. (1), the last partial result must be subtracted in the shifter/adder structure. For this reason, the main component of this stage is actually an adder/subtractor unit.

**4.0 PROPOSED 4 TAP FIR FILTER USING 4:2 COMPRESSOR**

Fig.7 shows proposed DA FIR filter in which the adder tree is constructed by normal 4:2 compressor adders. From above structure, we see that it is basically four major units; they are the input shifter units, DA units, Adder/shifter units, and control units. For testing, carry save adder is used in adder tree units. Adder/subtractor units are constructed by the circuit explained earlier.

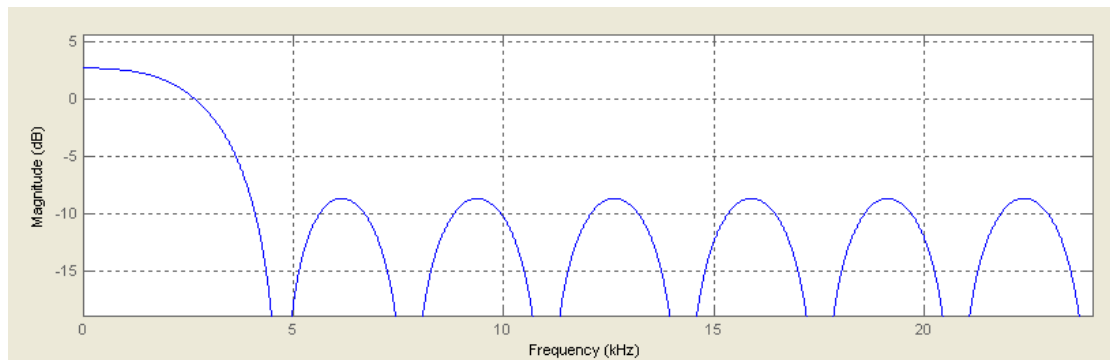


**Fig.7. Proposed architecture of 4 tap DA FIR filter using 4:2 compressor**

## 5.0 RESULTS AND DISCUSSION

In FIR filter implementation, filter coefficients represents the impulse response of the filter. The filter coefficients are generated from the MATLAB code. If the order of the filter is  $N$ , then  $N+1$  will be the coefficients terms of the filter. The filter coefficients are truncated with 8 bits of precision. To verify the correct functionality of the 4:2 compressor based DA filter, simulation was performed using TSPICE. All the circuits are targeted for 0.18 TSMC technologies. Finally, the results were validated with a MATLAB code.

After the construction of DA units, FIR filters can be constructed. In the two operand adder based DA FIR filters, power consumption is more due to the generation of many glitches. Due to long carry propagation, the carry save addition process is slow; hence, the delay increases for higher order filter. These drawbacks are overcome by the proposed 4:2 compressor based DA FIR filter. XOR-XNOR 4:2 compressor based DA FIR filter outperforms the other proposed designs. Fig.8 shows the frequency response of the 4 tap FIR filter



**Fig.8. Frequency response of FIR filter**

The DA unit is constructed and simulated for 4 tap, 8 tap and 16 tap respectively. Four to double compressor using XOR-XNOR outperforms the other proposed DA architecture. After the construction of DA units, FIR filters are constructed. The results are simulated and furnished in Table 1 for 4 tap, 8 tap, and 16 tap FIR filter respectively. In XOR-MUX based 4:2 compressor, the critical path of the compressor is 3 XOR gates.

A pure MUX based implementation of a 4:2 compressor requires CMOS inverters for inverting the input bits and output bits of intermediate MUXs. The inverters present at the input have more switching activity when compared at all the other nodes in the circuit; hence, the power dissipation is more. MUX based implementation requires 40 transistors for the implementation. In XOR-XNOR based DA FIR filters, due to the presence of XOR and XNOR outputs, the carry generation multiplexer does not require any extra inverter and none of their inputs needs any inverters. So it requires only 28 transistors for the implementation; hence, there is a reduction in power consumption. Also there is a reduction in delay because the critical path delay is less than the other proposed designs.

**Table 1 Performance analysis of distributed arithmetic FIR filter designs**

| Parameter                | Proposed FIR Filter using XOR-MUX based 4:2 Compressor |       |       | Proposed FIR Filter using Pure MUX based 4:2 Compressor |       |       | Proposed FIR Filter using XOR-XNOR based 4:2 Compressor |       |       |
|--------------------------|--------------------------------------------------------|-------|-------|---------------------------------------------------------|-------|-------|---------------------------------------------------------|-------|-------|
|                          | 4                                                      | 8     | 16    | 4                                                       | 8     | 16    | 4                                                       | 8     | 16    |
| Number of Taps           | 4                                                      | 8     | 16    | 4                                                       | 8     | 16    | 4                                                       | 8     | 16    |
| Power Dissipation (mW)   | 6.73                                                   | 13.53 | 23.13 | 6.82                                                    | 13.76 | 23.57 | 6.75                                                    | 13.87 | 23.32 |
| Average Delay(ns)        | 0.62                                                   | 0.63  | 0.64  | 0.59                                                    | 0.60  | 0.61  | 0.56                                                    | 0.57  | 0.58  |
| Power Delay Product (pJ) | 4.17                                                   | 8.52  | 14.41 | 4.02                                                    | 8.26  | 13.91 | 3.78                                                    | 7.91  | 13.52 |
| Area (Sq.µm)             | 17233                                                  | 34233 | 51266 | 19528                                                   | 39078 | 58326 | 21002                                                   | 42001 | 63412 |

**Table 2 Performance analysis of other researcher proposed FIR filter using distributed arithmetic unit**

| Parameter                | FIR Filter using Two Operand Adder based 4:2 Compressor |       |       |
|--------------------------|---------------------------------------------------------|-------|-------|
| Number of Taps           | 4                                                       | 8     | 16    |
| Power Dissipation (mW)   | 6.83                                                    | 15.01 | 25.41 |
| Average Delay (ns)       | 0.64                                                    | 0.65  | 0.65  |
| Power Delay Product (pJ) | 4.37                                                    | 9.75  | 16.51 |

The entire proposed DA FIR filter consumes less power and has less delay. The performance analysis of other authors proposed FIR filter using distributed arithmetic unit is shown in Table 2. XOR-XNOR based 4:2 compressor distributed arithmetic FIR filter has nearly 10% power saving and has a 13% reduction in power delay product when compared to the existing two operand adder based 4:2 compressor DA FIR filter. So XOR-XNOR based 4:2 compressor distributed arithmetic FIR filters outperform all the other proposed designs stated above.

## 6.0 Conclusion

In this paper, a new architecture for LUT-less distributed arithmetic FIR filtering has been proposed and compared against conventional multiplierless FIR filter. In the proposed architecture, compressors are actuated instead of two operand adders, which limit the carry propagation and so, the proposed method achieves high speed addition operation without increasing power consumption. The proposed 4:2 compressor based DA FIR filter has less delay and less power dissipation when compared to the existing

two operand adder based DA FIR filters. Three types of 4:2 compressor based FIR filters are proposed, namely XOR-MUX based 4:2 compressor DA unit FIR filters, pure MUX based 4:2 compressor DA FIR filters, and XOR-XNOR based 4:2 compressor DA FIR filters. Among the three proposed FIR filter designs, XOR-XNOR 4:2 compressor based DA FIR filters outperform the other types of proposed FIR filters. It saves nearly 10% of power and has 12% of reduction in delay achieved with the XOR-XNOR based 4:2 compressor distributed arithmetic FIR filter. So XOR-XNOR based 4:2 compressor distributed arithmetic FIR filters outperform all the other proposed design methods.

## REFERENCES

- [1] M.Keerthi, Vasujadev Midasala, S.Nagakishore Bhavanam, Jeevan Reddy“FPGA Implementation of distributed arithmetic for FIR filter”, International Journal of Engineering Research & Technology (IJERT) November 2012, Vol. 1, Issue 9.
- [2] S. M. Badave and A.S. Bhalchandra, “Multiplierless FIR Filter implementation on FPGA”, International Journal of Information and Electronics Engineering, May 2012, Vol. 2
- [3] Prof. S. J. Dashavant, Prof. A. I. Merchant “ASIC implementation of low power filter”, International Journal of Engineering Research & Technology (IJERT), April 2013, Vol. 2, Issue 4
- [4] Shahnammirzaei, Ryan Kastner, and Anup Hosangadi, “Layout aware optimization of high speed fixed coefficient FIR filters for FPGAs, ” International Journal of Reconfigurable Computing, 2010, Article ID 697625
- [5] Allred D.J. and Yoo H. “LMS adaptive filters using distributed arithmetic for high throughput”, IEEE transactions on circuits and systems, regular papers, vol. 52, 2005, pp.1327-1337
- [6] Pramod Kumar, Shrutisagar Chandrasekaran, “FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic”, IEEE transactions on signal processing, 2008.
- [7] Heejong Yoo and David V. Anderson, “Hardware-efficient distributed arithmetic architecture for high-order digital filters”, IEEE conference, 2005
- [8] Wang Sen, Tang Bin, Zhu Jun, “ Distributed arithmetic for FIR filter design on FPGA” in Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing, 2007, Vol. 5, pp620-623.
- [9] Partrick longa, Ali Miri, “Area-efficient FIR filter Design on FPGAs using distributed arithmetic”IEEE International symposium on system Signal Processing and Information Technology, 2006, pp:248-252.
- [10] Abhijit Chandra, Sudipta Chattopadhyay, “Novel approach of designing multiplier-less finite impulse response filter using differential evolution algorithm”, I.J. Intelligent Systems and Applications, 2012, Vol 4, pp 54-62

- [11] Hwang S., Han G., Kang S. and Kim J., “New distributed arithmetic algorithm for low power FIR filter implementation”, *IEEE Signal Processing Letters*, Vol.11, 2004, pp.463-466.
- [12] M. Yamada, and A. Nishihara, “High speed FIR digital filter with CSD coefficients implemented on FPGA”, in *Proc. IEEE Design Automation Conference (ASP-DAC 2001)*, 2001, pp. 7-8
- [13] 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.
- [14] D.J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. Anderson, “A novel high performance distributed arithmetic adaptive filter implementation on an FPGA”, in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP’04)*, Vol. 5, 2004, pp. 161-164.

