ISSN 2278-9723
doi: 10.31838/ijccts/07.SP01.03
Research Article

SYMMETRIC STACKING BINARY COUNTER

M.NAVIN KUMAR[1], A.S.DEVIKA[2], K.HARI KRISHNAN[3], S.JANE SELCIA[4]
[1]Assistant Professor,Department. of ECE,Sri Krishna College of Technology,Coimbatore.
[2]UG Scholar,Department of ECE,Sri Krishna College of Technology,Coimbatore.
Email id :15tuec039@skct.edu.in
[3]UG Scholar,Department of ECE,Sri Krishna College of Technology,Coimbatore.
Email id :15tuec047@skct.edu.in
[4]UG Scholar,Department of ECE,Sri Krishna College of Technology,Coimbatore.
Email id :15tuec056@skct.edu.in
Received: 03.05.19, Revised: 03.06.19, Accepted: 03.07.19

ABSTRACT
High efficient and fast addition of multiple operands is an essential process in any computational units. The power and speed efficiency of multiplier circuits is one of critical importance in the overall performance of microcontrollers and microprocessors. Multiplier circuits are an essential part of an arithmetic logic unit, or a digital signal processor system for performing convolution, image processing, filtering, and other purposes. The binary multiplication of fixed-point numbers and integers ends up in partial products that is used to provide the ultimate product. Adding those partial products dominates the power consumption and efficiency of the number. A new binary counter design uses 3- bit stacking circuit, which groups all the "1" bits together, to combine pairs of 3- bit stacks into 6- bit stacks through novel symmetric method has been proposed. The bit stacks square measure then reborn to binary counts, producing 6:3 counter circuits with no xor gates on the critical path. This avoidance of xor gates results in faster designs with efficient area and power utilization. Additionally, using the counters present in proposed system in existing counter - based Wallace tree multiplier architectures reduces latency and power consumption for 128 and 64 - bit multipliers. We apply this Counter design in FIR filter Application

Keywords: Stacking, Counter, Wallace, Multiplier, Filtering, Convolution.

INTRODUCTION

In the public key cryptosystems, Multiplication of large operands is one of the most widely used operations. Wallace tree multipliers offer high speed operation and therefore they are used extensively in high performance applications. The partial product tree present in the original Wallace tree multiplier is divided into groups and each group has three rows. Then the addition is performed in every column using Half Adder (HAs) and Full Adders (FAs). This process is repeated until the tree is reduced to set of rows. More number of papers have been published in the literature to improve the overall performance of the Wallace multiplier.

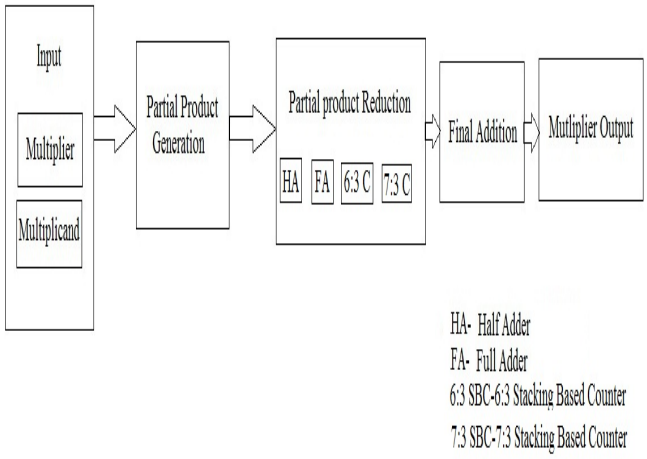

Fig.1 block diagram of symmetric stacking binary counter.

Symmetric Bit Stacking

In order to enhance the partial products expeditiously, column compression is commonly used. Many methods and techniques have been presented to optimize the performance of the partial product summation, such as the well-known row compression techniques in the Wallace tree[1] or Dadda tree, or in any other architecture[4]. The methods involve full adders that performs as counters to reduce groups of 3 bits of the same weight to 2 bits of different weight in parallel using a carry-save adder tree. By the reduction method in several layers, the number of summands is reduced to two, which are then added using a conventional adder circuit. The proposed 6:3 counter is realized by stacking all of the input bits such that all of the "1" bits are grouped together. After stacking the input bits, this stack can be converted into a binary count to output. Small 3-bit stacking circuits are used to form 3-bit stacks. Those 3-bit stacks are then combined to make a 6-bit stack using a symmetric stacking technique that adds extra one layer of logic[8].

A.Three-Bit Stacking Circuit

The given inputs X0, X1, and X2, a 3-bit stacker circuit will have three outputs Y0, Y1, and Y2 such that the number count of "1" bits in the outputs is the same as the number count of "1" bits in the inputs, but the "1" bits are grouped together to the left followed by the "0" bits. It is clear that the outputs are then formed by

Y0 = X0 + X1 + X2 -------(i)

Y1 = X0X1 + X0X2 + X1X2 -------(ii)

Y2 = X0X1X2--------. (iii)

As the first output will be "1" if any one of the inputs is one, the second output will be "1" if any two inputs are one, and the last output will be one if all three of the inputs are "1." The Y1 output is a majority function and can be implemented using one complex CMOS gate. The 3-bit stacking circuit is shown in Fig. 2.

B. Merging Stacks

We formed a 6-bit stacking circuit using the 3-bit stacking circuits [10]. The six inputs X0, . . . , X5, we first divide them into two groups of 3 bits which are stacked by 3-bit stacking circuits. Let X0, X1, and X2 be stacked into signals named H0, H1, and H2 and X3, X4, and X5 be stacked into I0, I, and I2. Initially, we reverse the outputs of the first stacker and consider the 6- bits H2,H1,H0, I0, I1,andI 2. Consider the Fig. 3 for an example of this process. We found that within these six bits, there is a number of "1" bits surrounded by "0" bits.

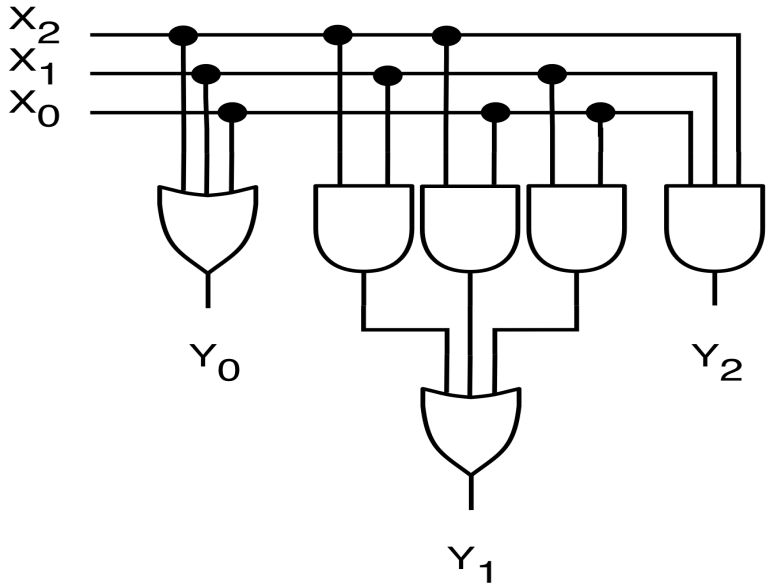

Fig.2.Three-bit stacker circuit.

To design a proper stack, this train of "1" bits must begin from the leftmost bit. In order to form the proper 6-bit stack, two more 3-bit vectors of bits are formed called J0, J1, J2 and K0, K1, K2. The method is to fill the J vector with "1"s in first, before filling the K vector. So we let

J0=H2+I0----------(iv)

J1=H1+I1----------(v)

J2=H0+I2.-------(vi)

In this way, the initial three "1" bits of the count are there to fill into the J bits although they may not be properly ordered. Now to ensure no of bits are counted two times, the K bits are formed using the same inputs but with the AND gates instead

K0=H2I0---------(vii)

K1=H1I1---------(viii)

K2=H0I2---------(ix)

If the number of "1"s is no more than three bits long, then all of the K bits will be zero as the AND gate inputs are three positions away. If the number is longer than three places, then some of the AND gates will have both inputs as "1"s as the AND gate inputs are three positions away. The number of AND gates that will have this property will be three less than the length of the number of "1"s. We found that J0 J1 J2 and K0K1K2 still contain the same number of "1" bits as the input in total but now J bits will be filled with ones before any of the K bits. We must stack J0 J1 J2 and K0K1K2 using two more 3-bit stacking circuits. The outputs of these two circuits can then be concatenated to form the stack outputs Y5,Y4,Y3,Y2,Y1,Y0. An example of this process is shown for an input containing four "1" bits in Fig. 3. In this example, initially the H and I vectors are formed by stacking groups of three input bits. Then, the H vector is reversed, forming a continuous train of four "1" bits surrounded by "0" bits. Simultaneously bits undergo OR operation to produce the J vector which is full of "1" bits. Corresponding bits undergo AND operation to form the K vector which finds exactly one overlap.

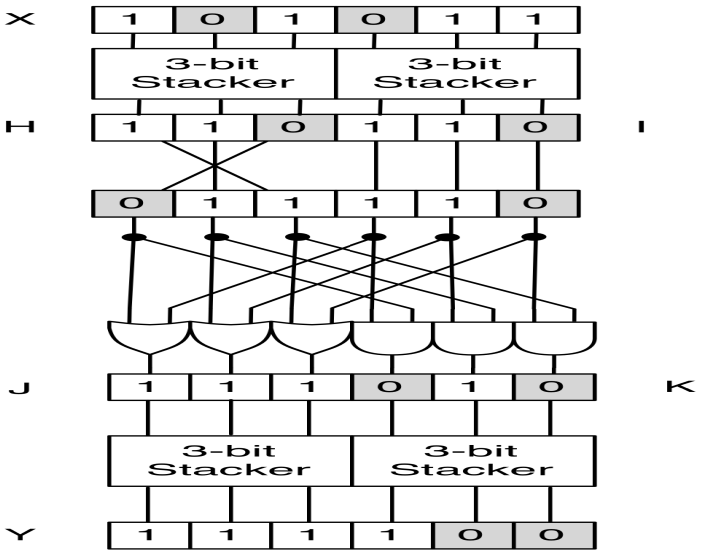

Fig.3. Six-bit stacking example.

Stacking Counter Based Wallace Tree Multiplier

With the increasing applications in portable equipment and mobile communications, there is an increasing demand for low-power VLSI[9] systems. In this, a full adder plays a major role since it forms the basic element in any processor design. The reduction of power and operation at low supply also plays a major role[2]. The multiplier's speed is a critical issue in determining the performance of microcontrollers and microprocessors. For implementing DSP algorithms such as convolution and filtering in digital signal processing systems require multipliers. Also Multipliers are used in microprocessors within their arithmetic logic units. The demand for high-speed multiplier circuits is continuously increasing.

A .The Process Of Fast Multiplication

The fast multiplication process consists of 3 steps:

- Generation of partial product,
- Reduction of partial product and
- Final carry propagating addition.

Approximate Computing

Many re-coding schemes are used to reduce the number of partial products. Compressors have been widely used for reduction process which usually contributes to the delay, power and area of the multiplier[3]. To enhance a better performance, the use of higher order counters instead of conventional

counters, e.g. 3:2 counters, have been considered. The reduction process finally results in a two-row matrix, and then a high speed adder is used to produce the final result from the two rows. Several type of implementation based on the conventional 7:3 model is proposed to be used for fast multiplication process or various addition applications. Thereafter, for the purpose that the rate of the compressor may be increased, other higher order compressors have been developed. This attempt has resulted in a faster partial product compression than the 3:2 counters. Though, the count based design result in area and power reduction, all the LSB part will be considered in the multiplier increasing its area and power consumption. As the comparison table I shows the difference between the multiplier circuit and the multiplier circuit after approximate computing.

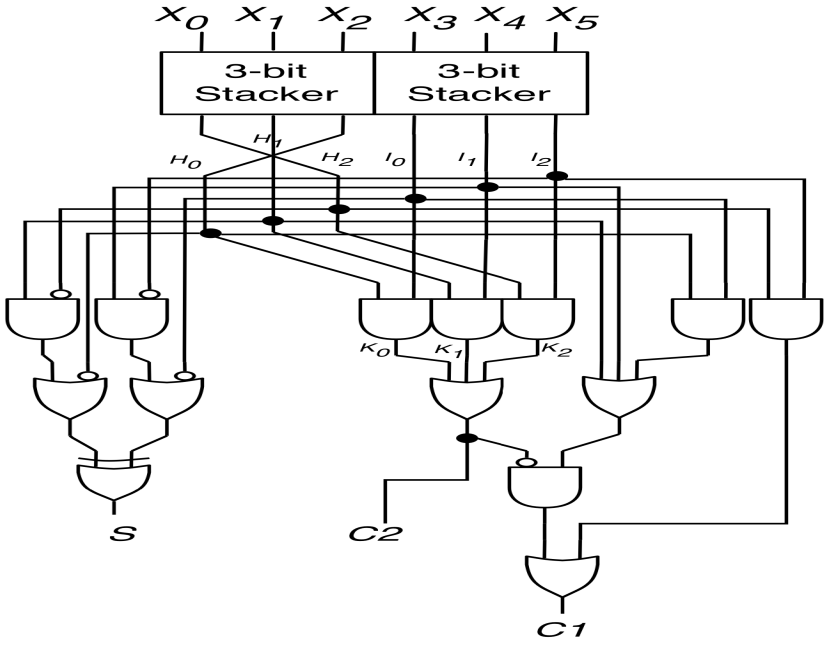

Fig.4 A 6:3 symmetric stacking counter .

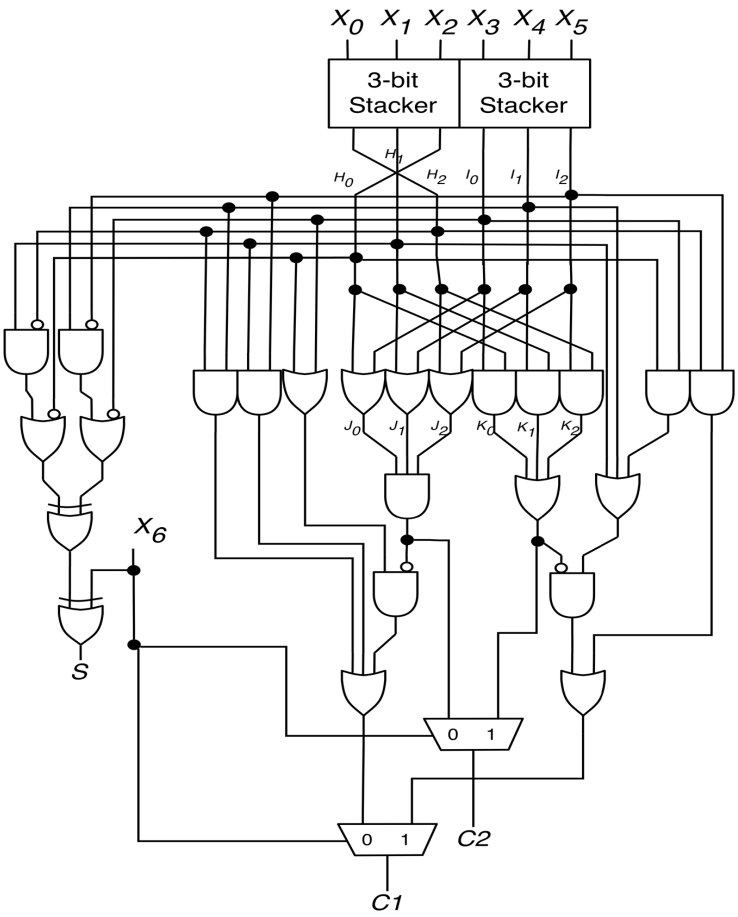

Fig.5 A 7:3 symmetric stacking counter.

Experimental Result

Table –I: Comparison Between Multiplier Circuits

DESIGN	MULTIPLIER	MULTIPLIER (APPR.COMP)
Area(Total no.of.gates)	1221	1146
Power	170 mw	157 mw
Delay	32.974ns	32.902 ns

Thus the approximate computing method by using the stacking method with all "1's" bits in MSB and "0's" bits in LSB has been implemented in a multiplier circuit. And the simulations results are shown in fig.5.

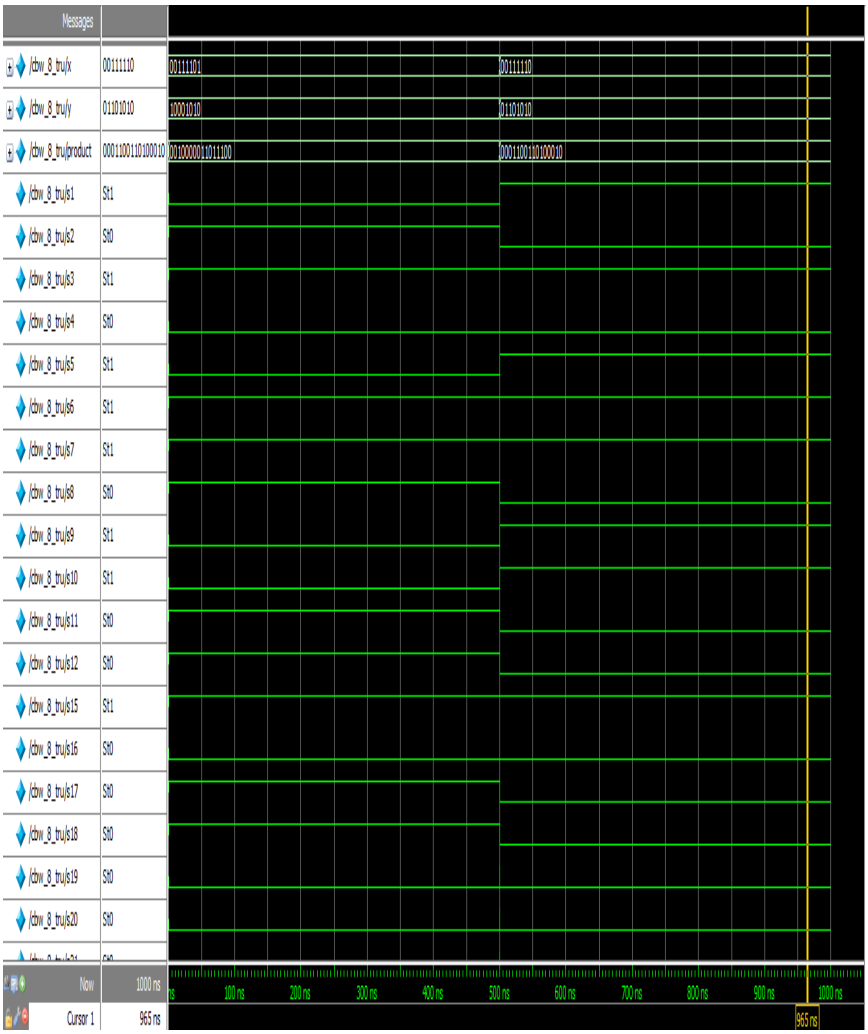

Fig.5 simulation result for multiplier circuit after approximate computing.

Application

filter design

Digital filters of high-performance are all important for the implementation of digital signal processing systems. The speed of a filter realization counts not alone on the potentialities of the hardware platform employed, but also on the computational structure of the code. The design of FIR filters are done using HDL languages since speed is the important parameter in the filter design[13]; the main aim is to improve the speed of the system. Since ultimately, speed, delay, power and chip area are the most often used measures of the efficiency of a system, there has a strong link between the system and technology applied for its implementation. Here we implemented the multiplier circuit using the symmetric stacking and the approximate computing method. The simulation results are shown in fig.6.

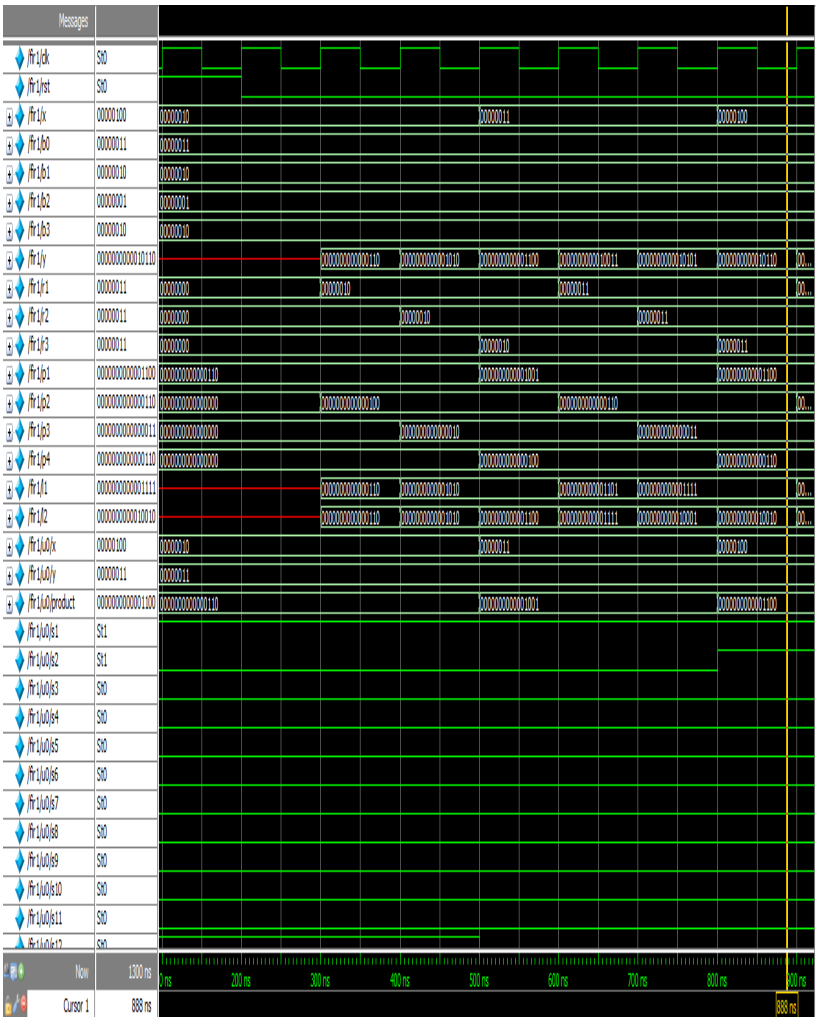

Fig .6. simulation result of multiplier circuit implemented in FIR filter.

Conclusion

A symmetric stacking binary counter based on approximate computing is proposed. The addition of partial products, is done by using 6:3 and 7:3 counters. We demonstrated the multiplier circuit by reducing the LSB part through the approximate computing method. Thus the 6:3 and 7:3 counters implemented with this approximate computing technique achieve higher speed than higher order counter designs and hence reduces power consumption. This is due to the reduction in the number of XOR gates and multiplexers on the critical path. And also this technique has been implemented in FIR filter as an application and simulated the results.

References

1. S. Asif and Y. Kong, "Design of associate degree recursive Wallace multiplier factor exploitation high speed counters," in Proc. IEEE Comput. Eng. Syst. (ICCES), Dec. 2015, pp. 133–138.
2. S. Veeramachaneni, L. Avinash, M. Krishna, and M. B. Srinivas, "Novel architectures for economical (m, n) parallel counters," in Proc.17th ACM Great Lakes Symp. VLSI, 2007, pp. 188–191.
3. S. Veeramachaneni, K. M. Krishna, L. Avinash, S. R. Puppala, and M.B. Srinivas, "Novel architectures for high-speed and low-power 3-2, 4-2 and 5-2 compressors," in Proc. 20th Int. Conf. VLSI Design Held Jointly 6th Int. Conf. Embedded Syst. (VLSID), Jan. 2007, pp. 324–329.
4. S. Asif and Y. Kong, "Analysis of various architectures of counter primarily based Wallace multipliers," in Proc. 10th Int. Conf. Comput. Eng. Syst. (ICCES), Dec. 2015, pp. 139–144.
5. Gu and C.-H. Chang, "Low voltage, low power (5:2) compressor cell for fast arithmetic circuits," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), vol. 2. Apr. 2003, pp. 661–664.

6. M. Rouholamini, O. Kavehie, A.-P. Mirbaha, S. J. Jasbi, and K. Navi, “A new style for 7:2 compressors,” in Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl., May 2007, pp. 474–478.

7. A. Dandapat, S. Ghosal, P. Sarkar, and D. Mukhopadhyay, “A 1.2-ns 16 × 16-bit binary multiplier using high speed compressors,” Int. J. Elect. Electron. Eng., vol. 4, no. 3, pp. 234–239, 2010.

8. Anand Kumar Thangapandi, Anandharaj VijaiRavindar, Boopathy KarthikRaj, Ponnusamy Keerthana, "Design and Implementation of Symmetric Multilevel Inverter", International Journal of Advances in Computer and Electronics Engineering, Vol. 3, No. 4, pp. 10-14, April 2018.

9. Jan M.Rabaey, Anantha Chandrakasan and Borivoje Nikolic, “Digital Integrated Circuits”, Second Edition, Prentice Hall Electronics and VLSI series, 2004.

10. Anand Kumar Thangapandi, Anandharaj VijaiRavindar, Boopathy KarthikRaj, Ponnusamy Keerthana, "Design and Implementation of Symmetric Multilevel Inverter", International Journal of Advances in Computer and Electronics Engineering, Vol. 3, No. 4, pp. 10-14, April 2018.

11. Christopher Fritz and Adly T. Fam, “Fast Binary Counters supported regular Stacking,” IEEE Trans.VLSI, Dec. 2017.

12. Sami Khorbotly, Joan E. Carletta, Robert J. Veillette “A Methodology for Implementing Pipelined Fixed-Point Infinite Impulse Response Filters” 41st Southeastern Symposium on System Theory University of Tennessee Space Institute Tullahoma, TN, USA, March 15-17, 2009

13. A.Mehrnia, A. N. Willson, ”FIR Filter style exploitation best Factoring: A Walkthrough and outline of Benefits”, IEEE Circuits and Systems Magazine, vol.16, no.1,pp.8-21,2016.