International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075, Volume-8 Issue-9, July 2019

Design and Analysis of a Low Power Binary Counter-based Approximate Multiplier Architecture

Anish Fathima. B, Mahaboob

Abstract*- Multiplication has become the fundamental arithmetic operation in modern electronic world. Many researches in Signal processing and image processing applications are looking for energy efficient architectures. These applications exhibit error tolerance thus laying foundation for approximation techniques. The proposed work utilizes a new binary counter for partial product accumulation in a segmentation based approximate multiplication technique. The fundamental building block of the counter is a 3-bit stack circuit, which combines each of "Logic 1" bits collectively, after which a stacking process is done to convert two 3-bit modules into a 6-bit stack module. The counter circuit is obtained by converting the bit stacks to binary counts, without any XOR gates on the critical line of operation. This leads to design of binary counters with effective yield of power and delay. Moreover, applying these counters for partial product accumulation in the approximate multipliers found to be more effective when compared with conventional techniques. In future, these counter based approximate multipliers can be utilized to design energy efficient filters for image processing and signal processing applications.

***Keywords*- Binary counter based Approximate Multiplication Segmentation based Approximate Multiplier, Symmetric Stacking Technique.**

I. INTRODUCTION

High speed performance, compact size and long lasting battery have become the tag line for many electronic gadgets in the current scenario. Many filtering applications in such gadgets use real time signal and image processing applications where multiplier plays a very vital role. Multiplier on the other hand consumes more power and area when the operands are 64 bit or more. Various results of the previous work show that approximation of the end product by compensating architectures or by compensating accuracy will lead to an energy efficient structure. As the signal and image processing applications give equally a better performance by accepting tolerable errors, Approximate multiplication came into existence. Column wise counting and compression is commonly used technique to combine the partial products efficiently. Many researches [8-10] concluded that such compression techniques are highly efficient and are used in tree multiplier architectures such as Dadda tree structures or Wallace tree structures or even their improved versions.These architectures have full adders working as binary counters to replace clusters of 3 bits of the identical mass to 2 bits of diverse mass using a carry-save adder. Consequently, the number of addends are only two, which can then be accumulated using a simple adder.

Revised Manuscript Received on July 05, 2019.

AnishFathima.B, Assistant Professor, Electronics and Communication Engineering, Sri Krishna college of Engineering and Technology, Coimbatore, India

Mahaboob.M, Assistant Professor, Electronics and Communication Engineering, Sri Eshwar college of Engineering, Coimbatore, India

In multipliers the multiplication process involving higher order bits such as 8-bits and 16-bits the addition of partial products become very difficult, the multiplication of integer numbers or fixed-point arithmetic gives partial products that must be accumulated to the end result. In such cases it was advised to use the binary counters. These binary counters are optimized and used in the prevailing multipliers such as Dadda and Wallace multipliers.In this work, an approximate multiplier is enhanced by using an energy efficient binary counter for partial product accumulation. Many DSP applications [6, 11] utilize mathematical equations and modules to work on the information that is fed into a system which mostly comprise noise. Critical failure never happens with such approximate multiplication since the end product does not degrade quality of those operations. [5,7]. Energy efficiency can be achieved by exploring error tolerating capacity. The rest of this work includes five sections as follows: Section II explains in detail about the binary counter utilized for partial product accumulation purpose in our multiplier architecture; Section III describes the segmentation based approximate multiplier structure; Section IV proposes a new architecture by incorporating the techniques explained in sections II and III. Section V analyses the simulation results using 180nm Tanner Tools – the power consumption, propagation delay and hence the energy consumed have been tabulated and represented graphically. Section VI concludes the entire brief stating the advantages and future scope of the work.

II. BINARY COUNTER TECHNIQUE

The 6 to 3 binary counter [1] is obtained by primarily combining all input bit in such a way that every "Logic 1" bits are Clustered into a single group. Later, this cluster will get transferred to a binary counter to unleash the 6-bit count value. Tiny 3-bit clustering circuits are made to be used to get 3-bit Clusters. These 3-bit Clusters later converted to get a 6-bit cluster by symmetric methodology.

A. 3-Bit Clustering technique

For X0, $X1$, and $X2$ being the three inputs, a three-bit cluster module can have the responses $Y0$, $Y1$, and $Y2$ like a way that the quantity of "Logic 1" bits in the responses is similar to the quantity of "Logic 1" bits in from input side, however the "logic 1" bits get clubbed with the leftmost side tracking "logic 0" bits.

The responses obtained follow the below postulates [1]

$$Y0 = X0 + X1 + X2 \quad (1)$$

$$Y1 = X0X1 + X0X2 + X1X2 \quad (2)$$

$$Y2 = X0X1X2. \quad (3)$$

Here, the primary response may be "Logic 1" if either of the input parameters is Logic 1, the secondary response results "Logic 1" if either two of the input parameters are "Logic 1", also the final response results "Logic 1", stating all three parameters are "Logic 1." The $Y1$ response is found to be the critical module and hence will be incorporated with an intricate CMOS gate arrangement. The 3-bit clustering module is depicted in figure 1.

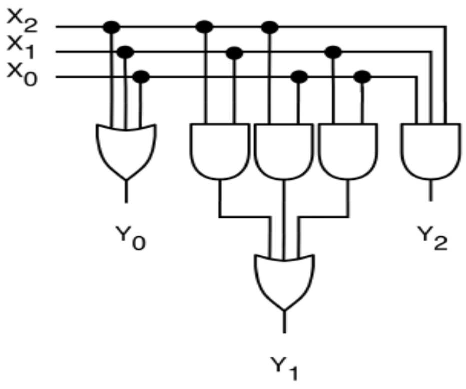


Fig-1 Three-bit Cluster

B. *Six Bit Clustering Technique*

It is required to frame 6-bit clusters with the already discussed 3-bit clusters. For six parameters $X0$ *to* $X5$, we first segregate them as two small bunches of 3 bits which are clustered using 3-bit clusters. Assume $X0$, $X1$, and $X2$ are stacked into signals named $H0$, $H1$, and H2 and $X3$, $X4$, and $X5$ are stacked into $I0$, I, and $I2$. Initially, inverse the responses of the primary cluster bits $H2$, $H1$, $H0$, $I0$, $I1$, and I2. We find that inside these six cluster bits, there will be a sequence of "logic 1" bits enclosed by "Logic 0" bits. For a perfect cluster, "Logic 1" bits will begin from the extreme left end. For an even better 6-bit Cluster, another two versions of 3-bit cluster of bits are created namely $J0$, $J1$, $J2$ and $K0$, $K1$, $K2$. The intention is to put the 1 bits first to the J vector rather before putting in the K vector.

Thus the postulates are

$$J0 = H2 + I0 \quad (4)$$

$$J1 = H1 + I1 \quad (5)$$

$$J2 = H0 + I2 \quad (6)$$

Likewise, the initial three "Logic 1" bits of the sequence are assured to occupy the J sector. To monitor no two bits are counted again and again, the K sector is created with the exactly same parameters however with the AND operations as given below$K0 = H2\ I0$ (7)

$$K1 = H1\ I1 \quad (8)$$

$$K2 = H0\ I2. \quad (9)$$

In case the sequence of "Logic 1" s is not above three continuous positions, then every K sector bits will be "Logic 0" because inputs of AND gate are three places apart. The sequence of "Logic 1" s is above three continuous positions, then few AND gates get all inputs as "Logic 1" s. The quantity of AND gates with this quality will be three less than the sequence length of "1" s. Consequently, $J0\ J1\ J2$ and $K0K1K2$ will contain the identical quantity of "Logic 1" bits like input bits as a whole however *J sector* bits will have Logic 1s well before the K sector. Then club $J0\ J1\ J2$and $K0K1K2$ with another two 3-bit clusters. Response of these two clusters can then be obtained to unleash the end output $Y5$ *to* $Y0$. An illustration showing four "1" bits as input vector in Figure 2 is given below. This figure shows the H and I Sector obtained by grouping clusters of three inputs. Now, H Sector is inverted, framing a sequence of four "logic 1" bits enclosed by "Logic 0" bits. Certain bits are OR-ed to frame the J vector which is made of High bits. Certain bits are AND-ed to frame the K vector where exactly one intersection is found. Then, both the J and K vectors are clubbed again get a 6-bit cluster.

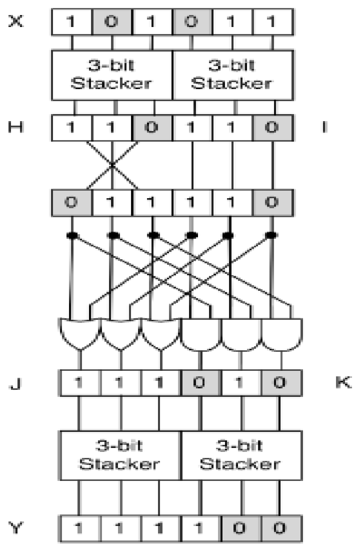


Fig-2 Six-bit Cluster

C. *Design of the 6 to 3 binary Counter*

To design a binary counter, the 6-bit cluster discussed in previous illustration to be deciphered into a binary number. Fast and effective counter can be obtained by using H, I, and K intermediate values to swiftly calculate every response eliminating last level of clusters. Retrieve the response $C2$, $C1$, and S, the binary equivalence of the "Logic 1" input binary digits. From the initial level of 3-bit clusters, the parity of response can be determined and in turn S can be estimated. H vector gives even parity in case "logic 0" or two "logic 1" bits get through in $X0$, $X1$, and $X2$. Even parities of H and I are given by the following postulates:

$$He = H0 + H1H2 \quad (10)$$

$$Ie = I0 + I1\ I2. \quad (11)$$

If S gives odd parity for every input bit, and since the sum of two binary digits with irrelevant parities is naturally odd, the parameter S can be defined as:

$$S = He \oplus Ie. \quad (12)$$

(12) encounters one delay of Exclusive OR gate, the critical path is not affected. Moreover, whenever the count is 2, 3, or 6, it is assumed $C1 = 1$. Thus, two possibilities are there. Initially, it is essential to check whether there are two inputs and the number of inputs should not go beyond three. H, I, and K vectors which are temporary can be utilized here. For two inputs, length two Clusters from either top level cluster, or two lengths one clusters such as $H1 + I1 + H0\ I0$ have to be monitored. As mentioned earlier, three inputs set cannot be allowed. For this, we just ensure that no K bits are set which is given by $(K0 + K1 + K2)'$. Next, it is important to investigate whether all six inputs are "Logic-1." If the H and I bits are set, then the above condition can be taken care of. Since these are bit clusters, it is enough to just investigate whether the rightmost bit in the cluster gives $H2\ I2$.

Finally, this yield

$C1 = (H1 + I1 + H0\,I0)\,(K0 + K1' + K2) + H2\,I2$ (13)

$C2$ can easily be computed while it has to be set for which the postulate would be:

$C2 = K0 + K1 + K2.$ (14)

With (12) – (14), the 6 to 3 Binary counter have been framed as shown in figure 3.

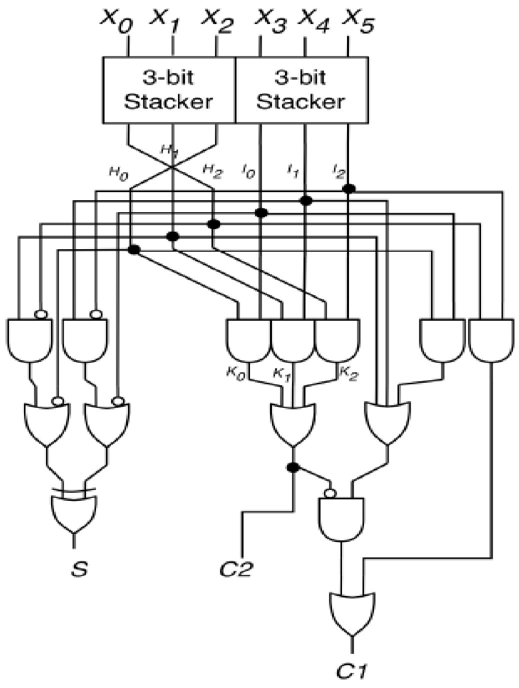


Fig-3 A 6:3 Binary Counter circuit

Using huge Complementary MOS structures, the delay of critical path is very much in control and limited to 7 gates. The designed 6:3 counter overtakes existing counters since no Exclusive OR structures available on the operating path, complexity in wiring and routing is one disadvantage of this design.

III. SEGMENT BASED MULTIPLICATION

The main principle is that only significant bits from the original operands are segmented and the multiplication is executed only for those segments. This method [2-3] exploits only the significant segments of operands unlike the existing techniques such as aggressive voltage scaling; truncation of bit-width and use of inaccurate building blocks for approximate multiplication. The flow of operation for a typical segmentation oriented approximate multiplication is as follows:

The methodology for multiplication in case of two n- bit numbers to be multiplied by segment oriented multiplication is explained as follows:

- Two n-bit (Eg: 16-bit) operands are given
- From each n-bit operand ,Select m-bit (Eg: 8-bit) segment
- The selected segment must comprise the leading 1 bit
- These two m-bit segments are then multiplied
- 2n-bit product is obtained by expanding the 2m-bit product

A. *Static Segment Method (Ss Method)*

An effective method for approximate multiplication is the Static Segment Method (SSM). Four possible combinations of taking two m-bit segments from two n-bit operands for a multiplication using the m-bit segment are found irrespective of m and n. To multiply, consider the m-bit segment that contains the leading 1 bit of each operand and use the selected segments from both operands to the m×m multiplier. Two possible static segment methods are:

- SS Method8 –the size of segment (m) is 8 bit
- SS Method10 – the size of segment (m) is 10 bit

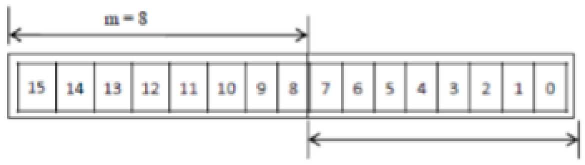


Fig-4: Segmentation of SS Method with m=8

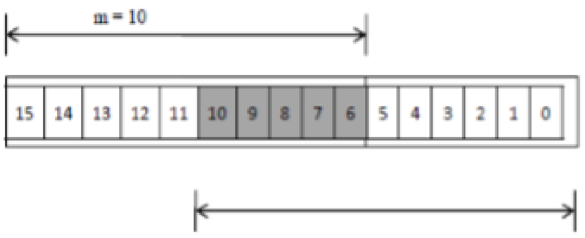


Fig-5: Segmentation of SS Method with m=10

This SS Method simplifies the process by selecting m-bit segments and directing them to the m bit multiplier by flipping two n-bit LODs and shifters with two (n–m) input OR gates and m-bit 2-to-1 multiplexers; if the first (n– m) bits starting from the MSB are all zeros, the lower m-bit segment must contain the leading one. Furthermore, the SS method allows us to swap the 2n-bit shifter with a 2n-bit 3-to-1 multiplexer. Since the segment for each operand is taken from one of two possible segments in an n bit operand, a 2m-bit result can be extended to a 2nbit result by left-shifting the 2m-bit result by one of three possible shift processes as given below:

1) When both segments are from the lower m-bit segments then perform no operation;

2) When two segments are obtained from the upper and lower ones respectively, perform (n–m) shift;

3) When both segments are obtained from the upper ones, perform 2 × (n–m) shifts

Figure 6 shows SSM to take an m-bit segment from two probable bit positions of an n-bit operand.

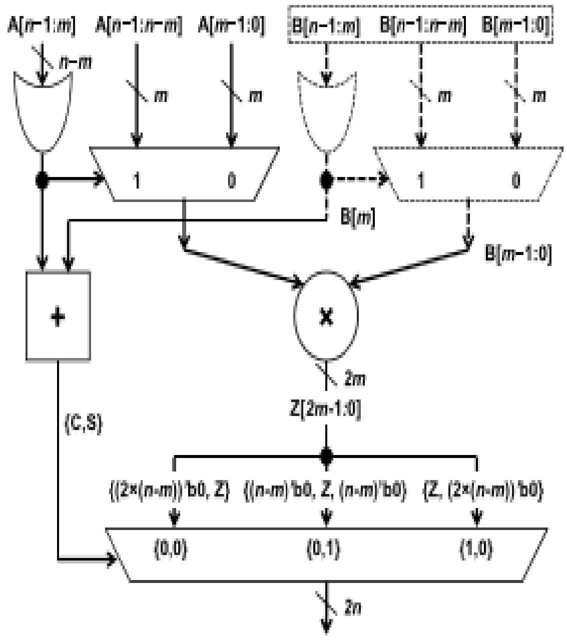
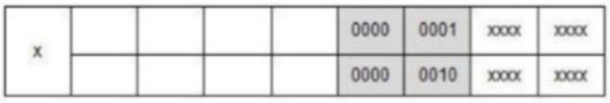


Fig-6: SS architecture

For operands as shown in figure 7, the accuracy factor with m = n/2 is significantly low:



x					0000	0001	xxxx	xxxx
					0000	0010	xxxx	xxxx

Fig-7: An example for low accurate operands

B. *Enriched Static Segment Method (ESS Method)*

To incorporate 3 feasible initiating bit locations for choosing an m-bit segment where m = n/2, the two 2-to-1 multiplexers at the input stage and one 3-to-1 multiplexer at the output stage are swapped with 3-to-1 and 5-to-1 multiplexers, respectively, along with some minor changes in logic operations generating multiplexer control signals. These variations lead to an enhanced architecture method called Enriched Static Segment Method (ESSM) which is shown in figure 8. This enriched SSM design for m = 8 and n = 16 (denoted by ESSM8×8) can provide as better accuracy as SSM10×10 at notably lower energy consumption.

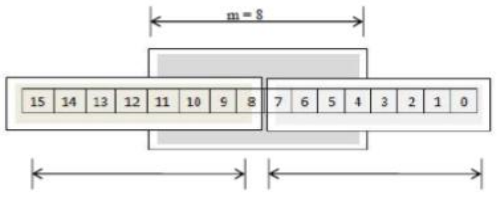


Fig-8: Segmenting in ESSM where m=8

IV. COUNTER BASED APPROXIMATE MULTIPLIER

The proposed work involves the combination of two novel techniques explained in section II and section IV. By using the binary counter for partial product accumulation in the true multiplication part (as mentioned in figure 5) of SSM and ESSM, an approximate multiplier with comparatively low power, less delay and hence low energy had been achieved with the same computational accuracy error of ~1%. The working module of the 6 to 3 binary counter as partial product accumulator in the true multiplier part of ESSM architecture is shown in figure 9 and 10 below:

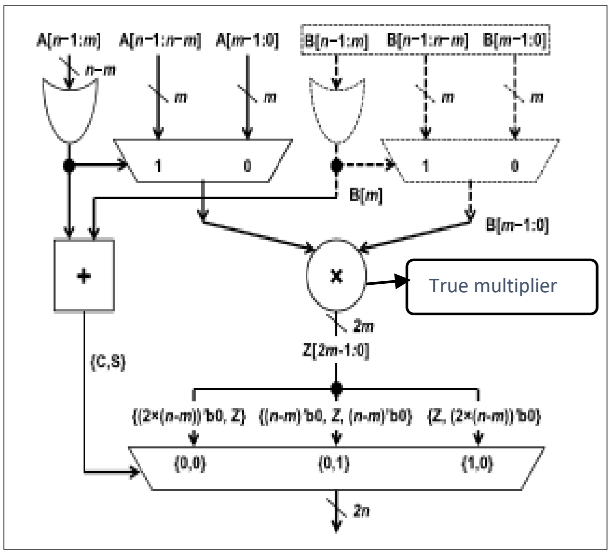


Fig-9: Proposed- Binary counter-based Multiplier of SSM and ESSM highlighted

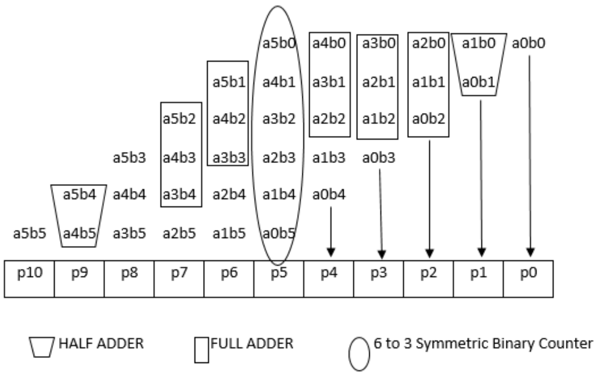


Fig-10: Binary Counter as Partial Product Accumulator in True Multiplier part of ESSM

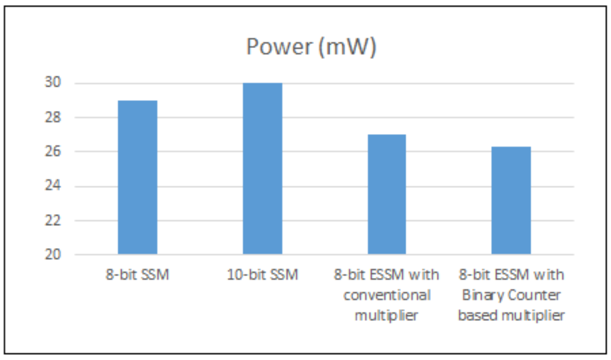
The binary counter reacts much faster when compared to most of the conventional full adder circuits [4] in accumulation, thus leading to a faster operation of multiplication.

V. SIMULATION AND IMPLEMENTATION RESULTS

In this work, design and implementation of energy efficient multiplier architectures were designed and simulated in TANNER TOOLS with 180nm technology. The performance parameters of these architectures were analysed in detail. The performance parameters include power, delay and computational accuracy.

Table 1- Comparison of Existing and Proposed Counters

Binary Counters	Power (m W)	Delay (s)	Power Delay Product (m J)
Conventional 6 to 3 counter	4.093	2.74	11.214
Symmetric stacking based 6 to 3 counter	2.372	1.61	3.818



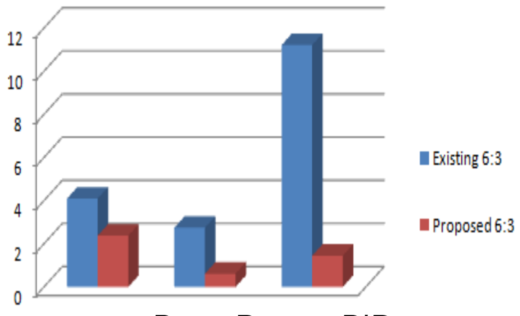


Fig-11 Comparison Chart for Counters

Where P represents Power (in mW), D represents Delay (in seconds) and P*D-Power*Delay (in mJ). The above shown comparison proves the efficiency of proposed counters in terms of power and delay. Thus the symmetric stacking based 6:3 counter proves to be more efficient for counter based multiplication architectures.

Table 2- Comparison of Approximate Multiplier Architectures with and without binary counters

Multiplier Architectures	Delay (ns)	Power (mW)	Energy (p J)	Average Computational Error (%)
Existing				
8-bit SSM	13.073	29	379.12	~6
10-bit SSM	12.451	30	373.53	~1
8-bit ESSM with conventional multiplier	13.691	27	369.66	~1
Proposed				
8-bit ESSM with Binary Counter based multiplier	**11.01**	**26.3**	**289.563**	**~1**

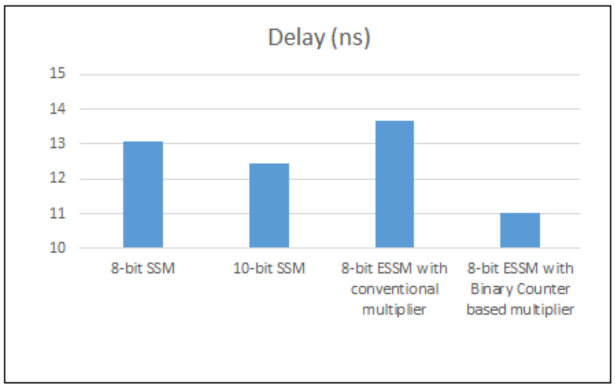


Fig-12 Comparison Chart for Delay variations

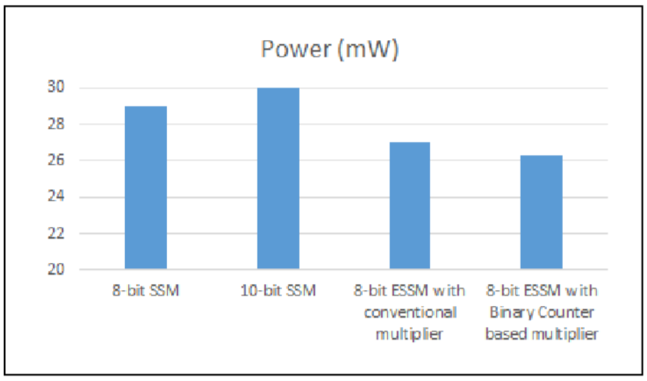


Fig-13 Comparison Chart for Power variations

Table 1 and figure 11 depicts the comparison between the symmetric stacking-based counter and the conventional counter. Table 2 shows the various approximate multiplier architectures and their performance parameters. The binary counter based ESSM approximate multiplier proves that it is more efficient in propagation delay as well as power and hence the energy efficient for real time signal and image processing applications.

VI. CONCLUSION

In this research paper, a novel binary counter based approximate multiplier was proposed and simulated in Tanner Tools with 180nm technology. The proposed multiplier architecture is 22% more energy efficient when compared to the previously proposed architecture. This is mainly because of the use of 6:3 binary counter that is used in place of mere full adders for partial product accumulation. Since no Exclusive OR Structures and multiplexers are there on the critical path, this binary counter achieves better speed with reduced latency and power consumption. This approximate multiplier besides being energy efficient, is also error tolerant as it has an average computational error of just 1% which in turn make this architecture ready for real time signal and image processing applications.

REFERENCES

1. Christopher Fritz and Adly T. Fam, "Fast Binary Counters Based on Symmetric Stacking" IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 25, No. 10, October 2017
2. Anish Fathima and C.Vasanthanayaki, " Design and Implementation of Energy Efficient Approximate

Multiplier", International Journal of Computer Applications (0975 – 8887)

3. Srinivasan Narayanamoorthy, Hadi Asghari Moghaddam, Zhenhong Liu, Taejoon Park, and Nam Sung Kim, July 2014, "Energy-Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Early Access Articles,pp. 1 – 5.
4. B. AnishFathima, "Design and Analysis of Low Power High Speed Hybrid Logic 8-T Full Adder Circuit", Asian Journal of Applied Science and Technology (AJAST), Volume 1, Issue 2, Pages 206-210, March 2017
5. Zdenek Vasicek and Lukas Sekanina, April 2014, "Evolutionary Design of Approximate Multipliers under Different Error Metrics", Design and Diagnostics of Electronic Circuits & Systems, 17th International IEEE Symposium, pp. 135 – 140.
6. K. S. Ganesh Kumar, J. Deva Prasannam, M. Anitha Christy, March 2014, "Analysis of Low Power, Area and High Performance Multipliers for DSP applications", International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 3, pp.278-382.
7. Vinay K. Chippa, Srimat T. Chakradhar, Kaushik Roy and Anand Raghunathan, May- June 2013, "Analysis and Characterization of Inherent Application Resilience for Approximate Computing", IEEE Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE, pp. 1- 9.
8. S. Wallace, "a suggestion for a fast multiplier," *ieee trans. Electron.Comput.*, vol. Ec-13, no. 1, pp. 14–17, feb. 1964.
9. Z. Wang, G. A. Jullien, and W. C. Miller, "a new design technique for Column compression multipliers," *ieee trans. Comput.*, vol. 44, no. 8, Pp. 962–970, aug. 1995.
10. M. Mehta, V. Parmar, and E. Swartzlander, "High-speed multiplier Design using multi-input counter and compressor circuits," in *proc. 10th Ieeesymp. Comput. Arithmetic*, jun. 1991, pp. 43–50.
11. P.Maheswari, K.Sankaragomathy, M.Ramya and B.AnishFathima, "Improved Fuzzy C Means Algorithm Based on Robust Information Clustering for Image Segmentation",Asian Journal of Applied Science and Technology (AJAST),Volume 1, Issue 3 ,Pages 70-74, April 2017

AUTHORS PROFILE

B. Anish Fathima completed her under graduation in Electronics and Communication Engineering. She pursued her post-graduation in VLSI design from Anna University, Chennai. She is currently working as an Assistant Professor in the department of Electronics and Communication Engineering at Sri Krishna college of Engineering and Technology, Coimbatore. She has published research papers in conferences and international journals. She has won the award for best paper in a National conference. Her research area includes Low power VLSI architecture design, Image processing in bio medical applications. She and her team of students are also interested in projects with rural development and women empowerment categories.

M. Mahaboob completed his under graduation in Electronics and Communication Engineering and post-graduation in Applied Electronics from Anna University, Chennai. He is currently working as an Assistant Professor in the department of Electronics and Communication Engineering at Sri Eshwar college of Engineering, Coimbatore. He has published papers in conferences and international journals. He has a Patent for his research work. His research area includes Low power VLSI architecture design, Real time Embedded Systems, Wearable technology and Networks. He is pursuing his research in the field of image processing.