Journal of Computer Science 7 (12): 1839-1845, 2011
ISSN 1549-3636

# Design of Low- Power High-Speed Error Tolerant Shift and Add Multiplier

[1]K.N. Vijeyakumar, [2]V. Sumathy
[1]Sriram Komanduri and [1]C. Chrisjin Gnana Suji
[1]Department of Electronics and Communication Engineering,
Anna University of Technology, Coimbatore,
Academic Campus, Jothipuram, Coimbatore-47, India
[2]Department of Electronics and Communication Engineering,
Government College of Technology, Thadagam Road, Coimbatore-13, India

**Abstract: Problem Statement:** In this study, we had proposed a low power architecture for high speed multiplication. **Approach:** The modifications to the conventional shift and add multiplier includes introduction of modified error tolerant technique for addition and enabling of adder cell by current multiplication bit of the multiplier constant. The proposed architecture enables the removal of input multiplexer, switching of adder cells and bypassing adder for zero bit values of the multiplier constant. The architecture makes use of down counter for tracking shift of partial products and multiplier bits. **Results:** When compared to the conventional architecture the simulation results for 8×8 multiplier shows that the proposed design reduces power consumption by 23.8% and delay by 35.6%. **Conclusion:** Enhanced performance of the proposed Error Tolerant shift and add multiplier in terms of power and delay makes it suitable for portable image processing applications where minimum percentage of error is tolerable.

**Key words**: High speed arithmetic, error tolerant technique, down counter, Partial Product (PP), image processing, power dissipation, Digital Signal Processing (DSP), Least Significant Bit (LSB), adder cells

## INTRODUCTION

Multiplier is one among the fundamental components of many digital and non digital systems and hence, their power dissipation and speed are of prime concern. In portable analog applications where power consumption is the most important parameter, one should reduce power dissipation to the possible limit. One of the best ways to reduce dynamic power dissipation is to minimize the total switching activity, i.e., total number of signal transitions of the system.

In analog computations, generation of “good enough” results is more important than totally accurate results (Breuer, 2005). Hence, by adopting error tolerance concept in design and test, it is possible to generate good enough results. To deal with high speed and low power circuits for analog computations , various adders and multipliers have been investigated. Multipliers based on word length reduction for multi-precision multiplication (http://public.itrs.net) showed that power reduction of 56% can be realized in case of 16 bit Wallace tree multipliers and 31% in case of modified 16 bit Booth multiplier for 8 bit truncation. However, power reduction can be achieved only at the expense of precision which exceeds tolerance for minimum bit constants. In (Al Mijalli, 2011) it was shown that the two’s complement multiplication can be realized using area efficient fixed width truncated Baugh-Wooley multipliers using error compensation biasing technique, for portable analog applications. The area of this multiplier is 32.7% less when compared to standard multiplier .However, the average error in the output is more than 10%. In this study, the design of an Error Tolerant (ET) Shift-and Add Multiplier is proposed. It utilizes the concept of error tolerant addition (Zhu *et al.,* 2010;) for accumulation of partial products and a down counter for shifting of multiplier bits and partial product. Since the system that incorporates this circuit produces acceptable results, it is said to be error tolerant.

Not all digital based applications can engage error-tolerant concept. In digital systems such as control systems, the correctness of the output signal is

**Corresponding Author:** [1]K.N.Vijeyakumar, Department of Electronics and Communication Engineering, Anna University of Technology, Coimbatore, Academic Campus, Jothipuram, Coimbatore-47, India extremely important, and this denies the use of the error tolerant circuit. However, for many Digital Signal Processing (DSP) systems that process signals relating to human senses such as hearing, sight, smell and touch, e.g., the image processing and speech processing systems, the error-tolerant circuits may be applicable (Breuer and Zhu, 2006; Lee *et al.*, 2005; Chong and Ortega, 2005 and Teymourzadeh *et al*., 2010).

## MATERIALS AND METHODS

The architecture of conventional shift-and-add multiplier (Marimuth. C.N *et al.*, 2010), which multiplies A by B is shown in Fig. 1. It has an adder, a multiplier (B) register, a multiplicand (A) register, an input multiplexer and a register to store the partial product. One input of the adder is multiplicand bits, and is fed through input multiplexer. The other input to the multiplexer is all zero bits. The select signal for the multiplexer is bit B (0) of multiplier. For B (0) equals to one, multiplicand A will be routed through the multiplexer and for B(0) equals to zero ,input from all zero bit register will be routed through multiplexer to the adder. The multiplexer output and partial product are added by the adder and the result is stored in partial product register. After current computation, the bits of partial product register and multiplicand register are shifted right by one bit position. Thus, the current bit B(0) moves out of register and next bit B(1) will occupy position B(0). The shifting and addition process are continued until all the bits of multiplier occupy position B(0). At the last cycle, the final bit of multiplier is moved out of the register and the result of multiplication is stored in partial product (PP) register and multiplier register (B).

There exists five major sources of switching activity in the multiplier which accounts for power dissipation. They are: (a) shift of B register, (b) activity in the adder, (c) switching between '0' and A in the multiplexer, (d) activity in the mux-select controlled by B(0), and (e) shifts of the partial product (PP) register. Note that the activity of the adder consists of required (when B(0) is nonzero) and unnecessary transitions (when B(0) is zero).

By removing or minimizing any of these switching activity sources, one can lower power consumption. Since, some of the nodes have higher capacitance, the reduction of their switching leads to more power reduction. As an example, elimination of input multiplexer and avoiding transitions in adder for zero value of bit B(0) results in noticeable power saving.

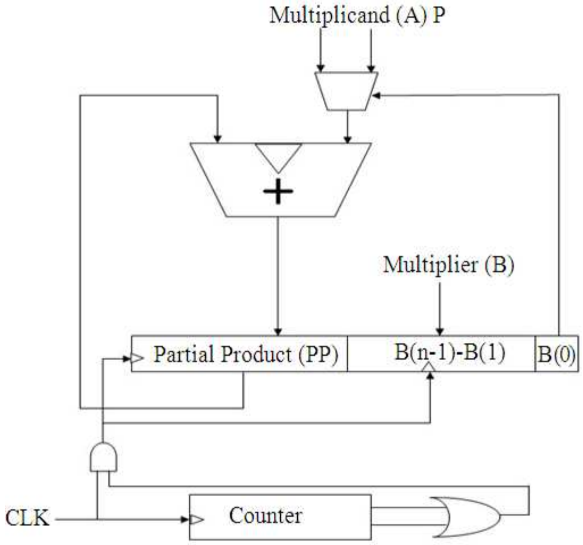


Fig. 1: Conventional shift-and- add Multiplier

A new addition technique based on Error tolerance concept is derived and used to design the proposed low power shift-and-add multiplier .

**Error Tolerant Addition:** The commonly used terminologies in Error Tolerant addition are as follows:

- Overall error (OE): OE=|$R_c$-$R_e$ |, where $R_e$ is the result obtained by the Error tolerant addition technique, and $R_c$ denotes the correct result (all the results are represented as decimal numbers).
- Accuracy (ACC): In the scenario of the error-tolerant design, the accuracy of an addition process is utilized to indicate how "correct" the output of an adder is for a particular input. It is defined as ACC%=(1-(OE/$R_c$)) x 100. Its value ranges from 0-100%.

**Addition Arithmetic:** In the conventional adder circuit, the delay is mainly attributed to the carry propagation chain along the critical path, from the least significant bit (LSB) to the most significant bit (MSB). Also glitches in the carry propagation chain dissipate a significant proportion of dynamic power dissipation. Therefore, if the carry propagation can be eliminated or curtailed, a great improvement in speed performance and power consumption (Zhu *et al.*, 2010) can be achieved.

This new addition arithmetic can be illustrated via an example shown below.

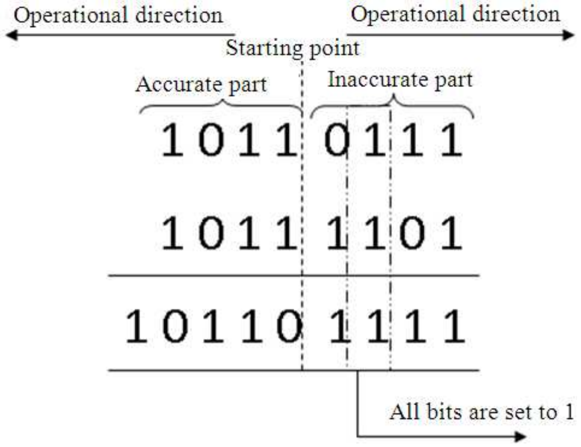


Fig. 2: Arithmetic procedure for error tolerant adder

Here, we discuss about the addition arithmetic proposed in (Zhu *et al*., 2010) where the input operand is split into two parts: with higher order bits grouped into accurate part and remaining lower order bits into inaccurate part. The length of each part need not necessary be equal. The addition process starts from the demarcation line toward the two opposite directions simultaneously. In the example of Fig. 2, the two 8-bit input operands, A= "10110111" (183) and B= "10111101" (189), are divided equally into 4 bits each for the accurate and inaccurate parts. The addition of the higher order bits (accurate part) of the input operands is performed from right to left (LSB to MSB) starting from the demarcation line with normal addition method applied . This is to preserve its correctness since the higher order bits play a more important role than the lower order bits. The lower order bits of input operands (inaccurate part) are added using error tolerant addition mechanism. No carry signal will be generated or taken in at any bit position to eliminate the carry propagation path. To minimize the overall error due to the elimination of the carry chain, a special strategy is adapted (Zhu *et al*., 2010), and can be described as follows: (1) check every bit position from left to right (MSB - LSB) starting from right of demarcation line; (2) if both input bits are "0" or different, normal one-bit addition is performed and the operation proceeds to next bit position; (3) the checking process is stopped when both input bits are encountered as high i.e., 1, and from this bit onwards, all sum bits to the right (LSB) are set to "1." The addition mechanism described can be easily understood from the example given in Fig. 3 with a final result of "101101111" (367) which should actually yield "101110100" (372) if normal arithmetic has been applied. The overall error generated can be computed as OE=372-367=5. The accuracy of the adder with respect to these two input operands is ACC=(1-(5/372))×100=98.66%. This accuracy level is acceptable for most of the image processing applications. Hence by eliminating carry propagation path in the inaccurate part and performing addition in two separate parts simultaneously, the overall delay time and power consumption is greatly reduced.

The plot of accuracy and delay of proposed 8 bit adder with different number of bits in accurate and inaccurate parts is shown in Fig.3.

From the Fig. 3 it is observed that the design with 4 bits in accurate part and 4 bits in inaccurate part yields an average accuracy of more than 98% for 100 samples taken. So the design of 4-4 Error Tolerant adder is considered and is used for our shift and add multiplier design.

**Proposed error tolerant adder:** The block diagram of the Error Tolerant adder that adapts to our proposed addition arithmetic is shown in Fig. 4. This most straightforward structure consists of two parts: an accurate part and an inaccurate part. The accurate part is constructed using conventional adder such as the Ripple- Carry Adder (RCA). The carry-in of this accurate part adder is connected to ground. The inaccurate part constitutes two blocks: a carry-free addition block and a control block. The control block is used to generate the control signals to determine the working mode of the carry-free addition block. In addition, the Least Significant Bit(LSB) of the multiplier(bit B(0)) is used as control bit P for both accurate part and inaccurate part of the proposed adder.

For B(0) is one, the adder cells performs normal addition operation. For B(0) equals to zero, the adder cells are brought into OFF state with NMOS and PMOS transistor driven by P brought into open state and the line from supply to ground is cut off , thus minimizing leakage power dissipation.

Based on the proposed methodology, an 8-bit Error tolerant adder is designed by considering 4 bits in accurate part and 4 bits in inaccurate part.

**Design of the accurate part:** In the proposed 8-bit ETA, the inaccurate and accurate parts consist of 4 bits each. Ripple-carry addition is the most power saving conventional addition technique, hence it has been chosen for the design of accurate part of the adder circuit.

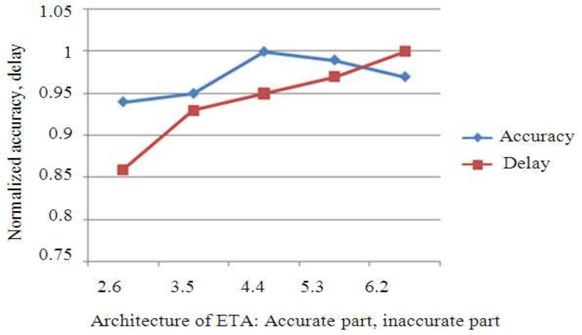


Fig. 3: Normalised graph of accuracy and delay for error tolerant adder

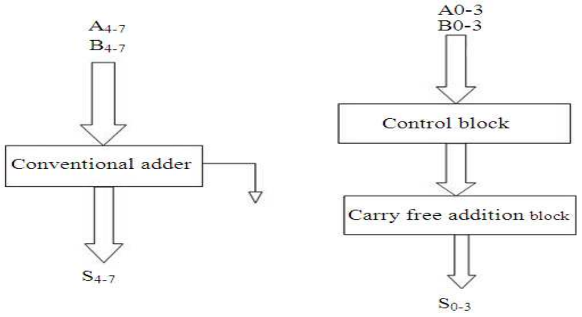
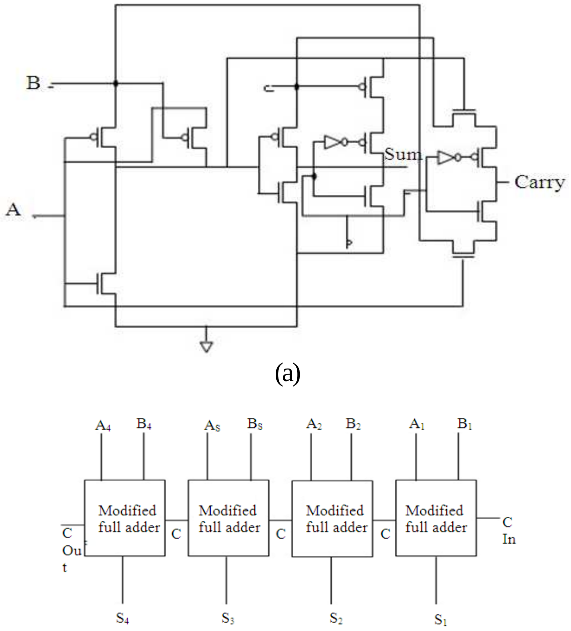


Fig.4: Block diagram of Error tolerant adder



(a)

(b)

Fig. 5: Implementation of accurate part (a) modified full adder (b) modified ripple carry adder

**Design of the inaccurate part:** The inaccurate part is the most critical section in the proposed ETA as it determines the accuracy, speed performance, and power consumption of the adder. The inaccurate part consists of two blocks: the carry free addition block and the control block. The carry-free addition block is designed using 4 modified XOR gates to generate a sum bit individually for LSBs. The block diagram of the carry free addition block and the schematic implementation of the modified XOR gate are shown in Fig.6.

In the modified XOR gate, six extra transistors M1, M2, M3, M4, M5 and M6 are added to the conventional XOR gate. CTL is the control signal coming from the control block and is used to set the state of transistors, while P (bit B(0) of multiplier) is used to set the mode of operation of modified XOR logic block. The state of transistors and the mode of operation for various values of CTL and P is shown in Table 1.

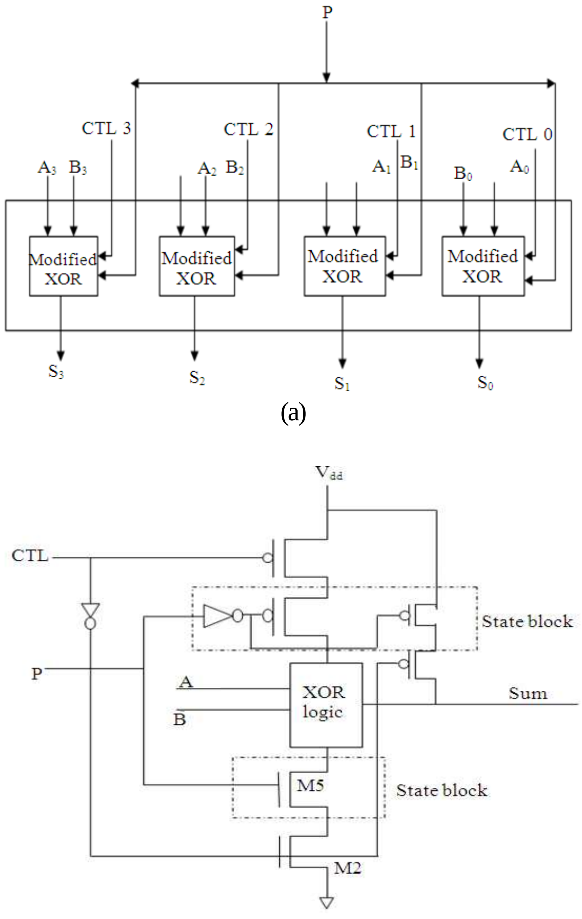
The conventional sum and carry blocks are modified by inserting extra PMOS and NMOS transistor driven by P(bit B(0) of multiplier) as shown in Fig.5. When P equals one PMOS transistor Ps1 and NMOS transistor Ns1 of sum block and PMOS transistor Pc1 and NMOS transistor Nc1 of carry block are in ON state and the cell performs normal addition operation. When P equals zero PMOS transistor Ps1 and NMOS transistor Ns1 of sum block and PMOS transistor Pc1 and NMOS transistor Nc1 of carry block are in OFF state and the cell is brought into high impedance. As the line from supply to ground is open during high impedance state, the chances of leakage power dissipation is minimized.

The function of the control block (Zhu et al., 2010) is to detect the first bit position when both input bits are "1," and to set the control signal CTL to high at this position as well as those to its right up to LSB.

As the proposed adder has 4 bits in inaccurate part, the control block is designed with 4 control signal generating cells (CSGCs) and each cell generates a control signal for the modified XOR gate in the corresponding bit position of carry-free addition block. Two types of CSGC, labeled as type I and II are designed and the schematic implementations of these two types of CSGC are shown in Fig.7. The control signal generated by the leftmost cell in each group is connected to the input of the leftmost cell in the adjacent group. These extra connections allow the propagated high control signal to "jump" from one group to another (Kuok, 1995)

Table 1: Mode of operation and state of transistors of modified xor block

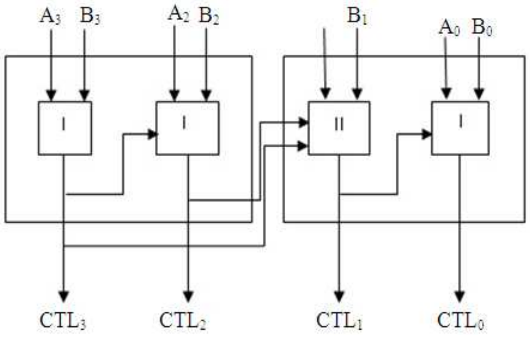
| P | M4 | M5 | M6 | CTL | M1 | M2 | M3 | OP |
|---|---|---|---|---|---|---|---|---|
| 1 | On | On | On | 0 | On | On | On | A oxr B |
| | | | 1 | Off | Off | ON | 1 | |
| 0 | Off | Off | Off | 0/1 | On/Off | On/Off | On/Off | Off |



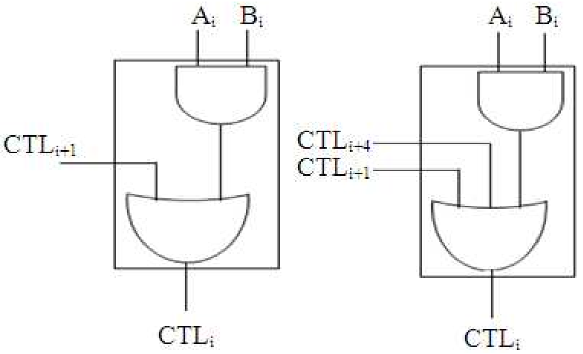
(a)

(b)

Fig 6: Implementation of carry free addition block (a) carry free addition block (b) modified XOR logic



(a)



(b)

Fig 7: Implementation of control block (a) over all architecture (b) schematic implementation of CSGC.

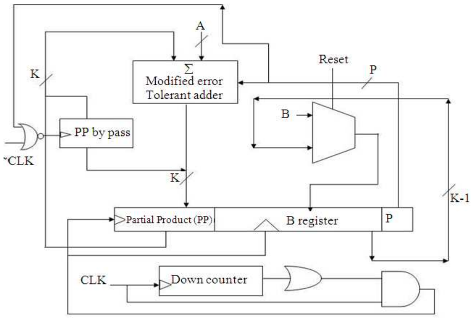


Fig 8: Proposed ET Shift- and add multiplier

**Proposed low power ET shift - and add multiplier:** In this section, the design of proposed shift and add multiplier which multiplies A by B using error tolerant adder for partial product accumulation is shown in Fig.8.The major blocks of the proposed design are (i) Error tolerant adder (ii) Partial product (PP) register (iii) Multiplier (B) register (iv)PP bypass register and (iv)Down counter. Initially PP register will be set to zero, B register is loaded with multiplier bits and A register with multiplicand bits. The B(0) bit(Least significant bit) of B register is used as the control signal P for Error tolerant adder. When P=1 , the multiplier bits in A register will be added with bits of partial product register. When P=0 the Error tolerant adder switches to OFF state and just the shifted bits of PP register is bypassed from adder using bypass register. The shifting of PP register together with B register is achieved using AND signal of down counter output and the clock as shown in Fig. 8.

Initially, on reset down counter will be loaded with all bits high. During each decrement of count values the contents of PP and B register will be shifted by one bit position towards LSB and the shifting procedure is halted when the counter bits attains all low. So the counter has to be designed based on the number of bits of multiplier.

Reducing switching activity of adder block and input multiplexer.

In the conventional multiplier architecture (Fig 1), in each cycle, the current partial product is added to *A* (when *B (0)* is one) or to 0 (when *B(0)* is zero). This leads to unnecessary transitions in the adder when B (0) is zero. For zero value of B (0) ,the Error-Tolerant adder in our proposed architecture (See Fig.5 and Fig.6) is switched OFF and *PP* Bypass register is used to bypass the adder . This reduces the switching activity in the adder and thus saves dynamic power consumption. Bypass register is triggered by a NOR gate output to store the current partial product only when B(0)=0 .The inputs of the NOR gate are the inverted clock (~*Clock* ) and B(0). Finally in each cycle, B (0) determines if the partial product should come from the PP Bypass register or from the Error Tolerant adder output. Since, one input of the Error tolerant adder is always A, which is constant during the multiplication, the input multiplexer is removed and *A* is fed directly to the adder, resulting in noticeable power saving by reducing switching activity of multiplexer. As Error tolerant adder used for accumulation of partial products involves carry free addition, the delay due to carry propagation can be reduced to a greater extent.

## RESULTS

The proposed ET shift-and add multiplier is designed in XILINX 10.2 using VHDL code and simulated using Modelsim5.7.

To evaluate the efficiency of the proposed architecture, we chose conventional shift-and add multiplier and BZ-FAD (By pass Zero Feed A directly) architectures for comparison.

To determine the effectiveness of power dissipation due to reduction in switching ,the transition counts of Conventional shift-and multiplier, BZ-FAD multiplier and our proposed ET shift-and add multiplier are reported in Table 2.

The power dissipation and delay comparison of the multipliers for normally distributed input data are shown in Table 3.

Another parameter that is worth mentioning is the Power-Delay Product (PDP) which gives energy consumption. Since, delay in general can be reduced by increasing power consumption, looking at either power or delay in isolation gives an incomplete picture. Using the obtained values of power and delay, the PDP can be calculated.

Table 2: Comparison of transition counts in conventional, BZ-FAD and ET shift-and add multipliers

| Component | Shift and<br>multiplier | BZFAD shift and<br>add multiplier | Et shift and<br>add multiplier |
|---|---|---|---|
| Adder | 32.435 | 23.15 | 18.574 |
| Multiplier | 5.006 | 28.31 | 7.145 |
| Latch11.487 | 10.598 | 8.546 | |

Table 3: Comparison of (a) Power,(b) Delay and (c) Power Delay Product(PDP) of conventional shift-and add, BZ-FAD and proposed ET Shift-and Add multipliers

| Conventional<br>Shift-and add<br>Multiplier | multiplier | BZ-FAD<br>shift-and add<br>add multiplier | ET shift-and |
|---|---|---|---|
| Power (mw) | 295 | 271 | 228 |
| Delay (ns) | 95 | 61 | 49 |
| PDP (E-9 Joules) | 28.025 | 16.531 | 11.172 |

## DISCUSSION

From Table 2 it can be inferred that switching activity of proposed Error Tolerant shift-and add multiplier is 42.8 % less compared to conventional shift and add multiplier and 19.8 % lower compared to BZ-FAD multiplier for chosen sample size.

From Table 3 it is seen that , the ET Shift-and add multiplier consumes 23.8% and 15.9% less power when compared with conventional shift and add multiplier and BZ-FAD shift and add multiplier respectively. Reduction in power dissipation is mainly due to the reduced number of switching activities in the proposed ET shift- and add multiplier . Since the blocks of Error tolerant adder are brought into high impedance state during zero bit value of multiplier, a constant saving in leakage power is achieved. Delay of proposed ET shift- and add multiplier decreases by 47.4% when compared to the conventional shift and add multiplier and by 23.1% when compared to BZ-FAD shift and add multiplier. The reduced delay of proposed ET shift- and add multiplier is due to the elimination of carry propagation in inaccurate part of the Error Tolerant adder used**.** Also for zero bit values of multiplier constant, the partial products are bypassed without passing through the adder which in addition contributes for the reduction in delay.

PDP of the proposed ET shift- and add multiplier is reduced by 59.9% when compared to the conventional shift- and add multiplier and by 35.3% when compared to BZ-FAD shift -and add multiplier.

On comparing the outputs of proposed ET shift-and add multiplier with actual values for 1000 number of samples, it is found that the percentage of error is 1.4 % i.e., the percentage of accuracy is 98.6 %. This percentage of error is most tolerable for image, speech signal and video processing applications.

## CONCLUSION

In this study,the concept of error tolerance is used in design of shift-and add multiplier. The proposed multiplier trades a certain amount of accuracy for significant power saving and performance improvement. Extensive comparisions with conventional multipliers showed that the proposed ET shift-and add multipier outperformed the conventional shift-and add multiplier and BZ-FAD multiplier in both power consumption and speed performance.The potential applications of the Error Tolerant Multiplier fall mainly in areas where there is no strict restriction on accuracy or where super low power consumption and high-speed peerformance are more important than accuracy. Few such applications are in Digital Image processing and DSP architectures for portable devices such as cellphones and laptops.

## REFERENCES

Breuer, M.A. and H.H. Zhu, 2006. Error-tolerance and multi-media. Proceedings of the International Conference on Intellegent Information Hiding and Multimedia Signal Processing, (IIHMSP’06), IEEE Xplore Press, Pasadena, CA, USA., pp. 521-524. DOI: 10.1109/IIH-MSP.2006.265055.

Breuer, M.A., 2005. Let's think analog. Proceedings of the IEEE Computer Society Annual Symposium on VLSI, May, 11-12, IEEE Xplore Press, CA, USA., pp.2-5. DOI: 10.1109/ISVLSI.2005.48

Kuok, H.H., 1995. Audio recording apparatus using an imperfect memory circuit,” U.S. Patent 5 414 758, May 9, 1995. Thomson Consumer Electronics, Inc.

Chong, I.S. and A. Ortega, 2005. Hardware testing for error tolerant multimedia compression based on linear transforms. Proceedings of the International Symposium on Defect and Fault Tolerance in VLSI Syste, Oct. 3-5, IEEE Xplore Press, Los Angeles, CA, USA., pp. 523-531. DOI: 10.1109/DFTVS.2005.38.

Lee, K.J., T.Y. Hsieh and M.A. Breuer, 2005. A novel test methodology based on error-rate to support error-tolerance. Proceedings of the International Test Conference, Nov. 8-8, IEEE Xplore Press, Austin, TX, pp. 1144-1149. DOI: 10.1109/TEST.2005.1584081.

Al Mijali, M.H., 2011. Spartan-3AN field programmable gate arrays truncated multipliers delay study. Am. J. Applied Sci. 8: 554-557. DOI: 10.3844/ajassp.2011.554.557

Ning Zhu, Wang Ling Goh, Weija Zhang, Kiat Seng Yeo, and Zhi Hui Kong, 2010. Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing. IEEE Trans. Very Large Scale Integrat., 18: 1225-1229. DOI: 10.1109/TVLSI.2009.2020591.

Teymourzadeh, R., Y.S. Algnabi, N. Mahdavi and M.B. Othman, 2010. On-Chip implementation of high resolution high speed floating point adder/subtractor with reducing mean latency for OFDM. Am. J. Eng. Applied Sci., 3: 25-30. **DOI:** 10.3844/ajeassp.2010.25.30.