International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-6, August 2013

An Implementation on 32-Bit High Speed Truncation- Error -Tolerant Adder with Low power Consumption

Tadgiri Aruna, R.Bhadraiah

Abstract: In this study, we had proposed architecture for high speed Truncation Adder Algorithm. In modern VLSI technology, the occurrence of all kinds of errors has become inevitable. By adopting an emerging concept in VLSI design and test, error tolerance (ET), a novel error-tolerant adder (ETA) is proposed. The ETA is able to ease the strict restriction on accuracy, and at the same time achieve tremendous improvements in both the power consumption and speed performance. When compared to its conventional counterparts, the proposed ETA is able to attain more than 74% improvement. One important potential application of the proposed ETA is in digital signal processing systems that can tolerate certain amount of errors. 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.

Key words: High speed arithmetic, error tolerant technique, image processing, power dissipation, Digital Signal Processing (DSP), Least Significant Bit (LSB), adder cells, high-speed integrated circuits, low-power design, VLSI.

I. INTRODUCTION

In conventional digital VLSI design, one usually assumes that a usable circuit/system should always provide definite and accurate results. But in fact, such perfect operations are seldom needed in our non digital worldly experiences. The world accepts "analog computation," which generates "good enough" results rather than totally accurate results. The data processed by many digital systems may already contain errors. In many applications, such as a communication system, the analog signal coming from the outside world must first be sampled before being converted to digital data. The digital data are then processed and transmitted in a noisy channel before converting back to an analog signal.

During this process, errors may occur anywhere. Furthermore, due to the advances in transistor size scaling, factors such as noise and process variations which are previously insignificant are becoming important in today's digital IC design. Based on the characteristic of digital VLSI design, some novel concepts and design techniques have been proposed. The concept of error tolerance (ET) [3]–[10] and the PCMOS technology are two of them. According to the definition, a circuit is error tolerant if: 1) it contains defects that cause internal and may cause external errors and 2) the system that incorporates this circuit produces acceptable results [3].

Manuscript published on 30 August 2013.
* Correspondence Author (s)
Tadgiri Aruna (M.Tech), Nimra College Of Engineering & Technology, Jupudi, Ibrahimpatnam (Mandal), Vijayawada, Krishna (Dist)-521456., India.
R.Bhadraiah, M.Tech, Assistant Professor, Nimra College Of Engineering & Technology Jupudi, Ibrahimpatnam (Mandal), Vijayawada, Krishna (Dist)-521456, India.

The "imperfect" attribute seems to be not appealing. However, the need for the error-tolerant circuit [3]–[10] was foretold in the 2003 International Technology Roadmap for Semiconductors (ITRS) [2]. To deal with error-tolerant problems, some truncated adders/multipliers have been reported [14], [15] but are not able to perform well in either its speed, power, area, or accuracy. The "flagged prefixed adder" [14] performs better than the non flagged version with a 1.3% speed enhancement but at the expense of 2% extra silicon area.

As for the "low-error area-efficient fixed-width multipliers" [15], it may have an area improvement of 46.67% but has average error reaching 12.4%. Of course, not all digital systems can engage the error-tolerant concept. In digital systems such as control systems, the correctness of the output signal is 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 [3], [6], [7].

II. ERROR-TOLERANT ADDER

A. Need for Error-Tolerant Adder

Increasingly huge data sets and the need for instant response require the adder to be large and fast. The traditional ripple-carry adder (RCA) is therefore no longer suitable for large adders because of its low-speed performance. Many different types of fast adders, such as the carry-skip adder (CSK) [16], carry-select adder (CSL) [17], and carry-look-ahead adder (CLA) [18], have been developed. Also, there are many low-power adder design techniques that have been proposed [19]. However, there are always trade-offs between speed and power. The error-tolerant design can be a potential solution to this problem. By sacrificing some accuracy, the ETA can attain great improvement in both the power consumption and speed performance.

B.Error Tolerant Addition

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

• Overall error (OE): OE=|Rc-Re |, where Re is the result obtained by the Error tolerant addition technique, and Rc 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/Rc)) x 100. Its value ranges from 0-100%.

C.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.

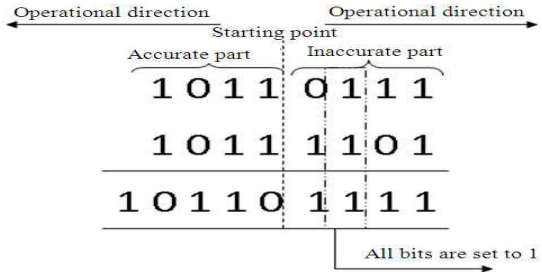

Fig. 1: Arithmetic procedure for 8 bit error tolerant adder

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:

- Check every bit position from left to right (MSB - LSB) starting from right of demarcation line;
- If both input bits are "0" or different, normal one-bit addition is performed and the operation proceeds to next bit position;
- 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 adds multiplier design.

III. DESIGN OF A 32-BIT ERROR-TOLERANT ADDER

The first step of designing a proposed ETA is to divide the adder into two parts in a specific manner. The dividing strategy is based on a guess-and-verify stratagem, depending on the requirements, such as accuracy, speed, and power.

With this partition method defined, we then check whether the accuracy performance of the adder meets the requirements preset by designer/ customer. This can be checked very quickly via some software programs. For example, for a specific application, we require the minimum acceptable accuracy to be 95% and the acceptance probability to be 98%. The proposed partition method must therefore have at least 98% of all possible inputs reaching an accuracy of better than 95%. If this requirement is not met, then one bit should be shifted from the inaccurate part to the accurate part and have the checking process repeated. Also, due to the simplified circuit structure and the elimination of switching activities in the inaccurate part, putting more bits in this part yields more power saving.

Having considered the above, we divided the 32-bit adder by putting 12 bits in the accurate part and 20 bits in the inaccurate part.

A. Design of the Accurate Part

In our proposed 32-bit ETA, the inaccurate part has 20 bits as opposed to the 12 bits used in the accurate part. The overall delay is determined by the inaccurate part, and so the accurate part need not be a fast adder. The ripple-carry adder, which is the most power-saving conventional adder, has been chosen for the accurate part of the circuit 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.

B. Design of the Inaccurate Part

The inaccurate part consists of two blocks: the carry free addition block and the control block. The carry-free addition block is made up of 20 modified XOR gates, and each of which is used to generate a sum bit. The block diagram of the carry-free addition block and the schematic implementation of the modified XOR gate are presented in Fig. 5

In a 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). Meanwhile, a significant proportion of the power consumption of an adder is due to the glitches that are caused by the carry propagation. Therefore, if the carry propagation can be eliminated or curtailed, a great improvement in speed performance and power consumption can be achieved.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

In this paper, we propose for the first time, an innovative and novel addition arithmetic that can attain great saving in speed and power consumption. This new addition arithmetic can be illustrated via an example shown in Fig. 1. We first split the input operands into two parts: an accurate part that includes several higher order bits and the inaccurate part that is made up of the remaining lower order bits. The length of each part need not necessary be equal. The addition process starts from the middle (joining point of the two parts) toward the two opposite directions simultaneously. In the example of Fig. 1, the two 16-bit input operands, "1011001110011010" (45978) and "0110100100010011" (26899), are divided equally into 8 bits each for the accurate and inaccurate parts.

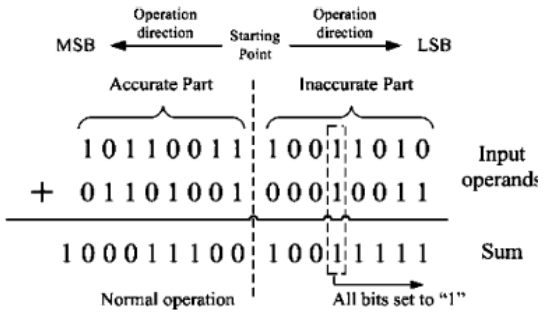

Fig. 2: Arithmetic procedure for 16 bit error tolerant adder

The addition of the higher order bits (accurate part) of the input operands is performed from right to left (LSB to MSB) and normal addition method is 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 the input operands (inaccurate part) require a special addition mechanism. No carry signal will be generated or taken in at any bit position to eliminate the carry propagation path. final result of "10001110010011111" (72863).

$$P(ACC=100\%) = 4^{Nt-Ni} \times 3^{Nt} + 2^{Nt-Ni} / 4^{Ni} + 2^{Nt} \quad \text{.....(1)}$$

B. Relationships Between Minimum Acceptable Accuracy

Acceptance Probability, Dividing Strategy, and Size of Adder The accuracy of the adder is closely related to the input pattern. Assume that the input of an adder is random; there exists a probability that we can obtain an acceptable result (i.e., the acceptance probability). The accuracy attribute of an ETA is determined by the dividing strategy and size of adder. In this subsection, the relationships between the minimum acceptable accuracy, the acceptance probability, the dividing strategy, and the size of adder are investigated.

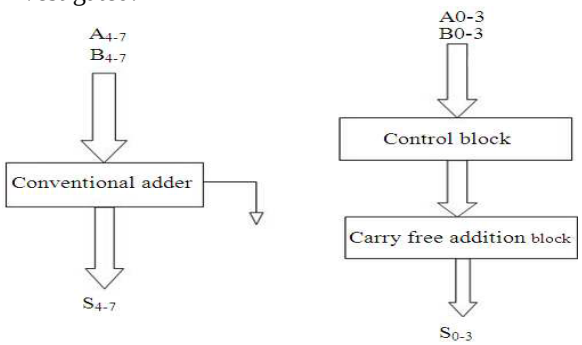

Fig.3: Block diagram of Error tolerant adder

We first consider the extreme situation where we accept only the perfectly correct result. The minimum acceptable accuracy in this "perfect" situation is 100%. According to the proposed addition arithmetic, we can obtain correct results only when the two input bits on every position in the inaccurate part are not equal to "1" at the same time. Wecan therefore derive an equation to calculate the acceptance probability associated with the proposed ETA with different bit sizes and dividing strategies. This equation is given as follows where is the total number of bits in the input operand (also regarded as the size of the adder) and is the number of bits in the inaccurate part (which is indicating the dividing strategy).

In situations where the requirement on accuracy can be somewhat relaxed are investigated, the result will be different. C program is engaged to simulate a 16-bit adder that had adopted the proposed addition mechanism.

As modern VLSI technology advances, the size of the adder has to increase to cater to the application need. The trend of the accuracy performance of an ETA is therefore investigated in Fig. 3. The five curves are associated with different minimum acceptable accuracies, 95%, 96%, 97%, 98%, and 99%, respectively. Note that all adders follow the same dividing strategy whereby the inaccurate part is three times larger than that of the accurate part. Since small numbers will be calculated at the inaccurate part of the adder, the proposed ETA is best suited for large input patterns.

The block diagram of the Error Tolerant adder that adapts to our proposed addition arithmetic is shown in Fig. 3. 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.

D. 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.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

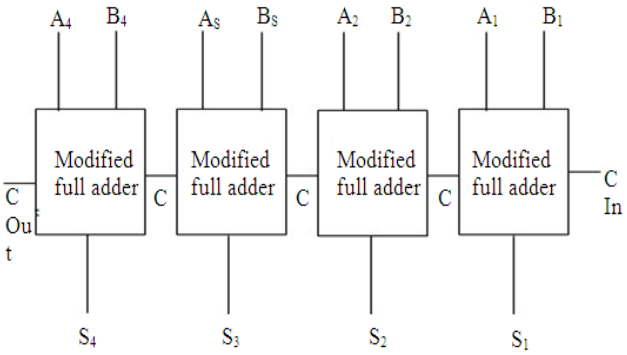

Fig. 4: Implementation of accurate part modified ripple carry adder.

The Above figure4 shows the implementation of accurate part modified ripple carry adder.

Table 1: Simmulation Results for ETA and Conventional Adder.

Type of Adder	Power (mW)	Delay (ns)	PDP (pJ)	PDP saving (%)	Transistor Count
RCA	0.22	4.04	0.89	66.29	896
CSK	0.46	2.90	1.33	77.44	1728
CSL	0.60	3.06	1.84	83.70	2176
CLA	0.51	2.37	1.21	75.21	2208
ETA	0.13	2.29	0.30	N.A.	1006

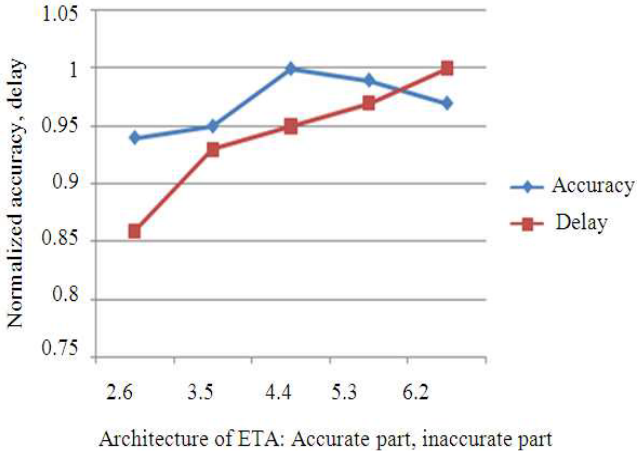

Fig. 5: Normalized graph of accuracy and delay for error tolerant adder

E. 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.

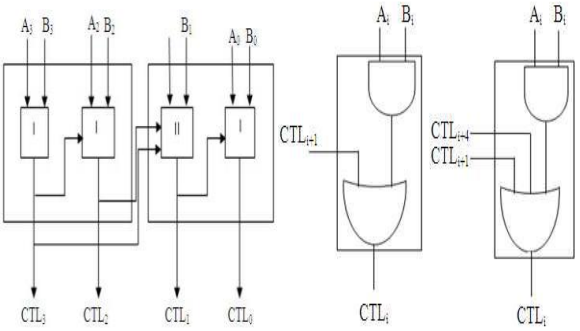

Fig 6: Implementation of control block (a) over all architecture (b) schematic implementation of CSGC.

F. Delay in adder

The combinational logic circuits can't compute the outputs instantaneously. There is some delay between the time the inputs are sent to the circuit and the time the output is computed. While the adders are working in parallel, the carries must "ripple" their way from the least significant bit and work their way to the most significant bit. It takes T units for the carry out of the rightmost column to make it as input to the adder in the next to rightmost column.

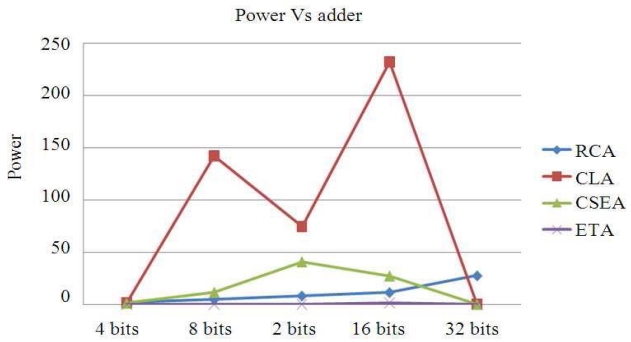

Fig. 7: Power Vs Adders

G. Power consumption in adders

Addition is an operation common in circuits designed for portable equipment and is typical of the digital processing carried out in computer systems. In CMOS circuits most of the energy consumed is due to switching activity, with the number of nodes in the circuit, the stored energy per node and the number of switching operations per second all contributing to the total power consumption. Power consumption was paid more and more attention to by IC designers. The motive of low power design comes from two reasons:· For those chips used in products supplied by battery, such as portable computers and hand-held devices, lower power consumption is one of the key features surpassing their competitors With the steadily increasing of chip's capacity and density, low power consumption becomes a vital feature for chip's functionality and reliability. High power density will make chip's temperature increasing, thus cause path delay increasing and problem of metal immigration. Building low power VLSI system has emerged as significant performance goal because of the fast technology in mobile communication and computation. The advances in battery technology have not taken place as fast as the advances in electronic devices. So the designers are faced with more constraint; high speed, high throughput and at the same time, consuming as minimal power as possible.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The goal is to extend battery life span of portable electronics is to reduce the energy expended per arithmetic operation, but low power consumption does not necessarily result in low energy dissipation. To execute an arithmetic operation, a circuit can consume very low power by clocking at extremely low frequency but it may take a very long time to complete the operation. We measure the energy consumption by the product of average power and worst case delay (power delay- product). Thus ETA is found to have less delay and have less power consumption.

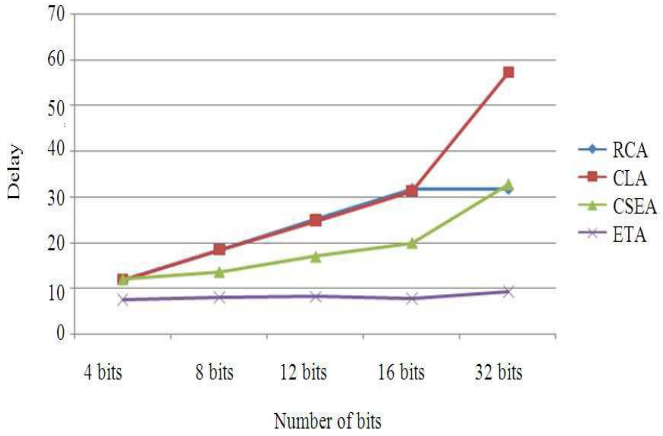

Fig. 8: Delay Vs no of bits in adders

IV.RESULTS

The proposed 32 bit ET Adder is designed in XILINX 9.2 using VERILOG HDL code and simulated using Modelsim 6.5e To evaluate the efficiency of the proposed architecture,

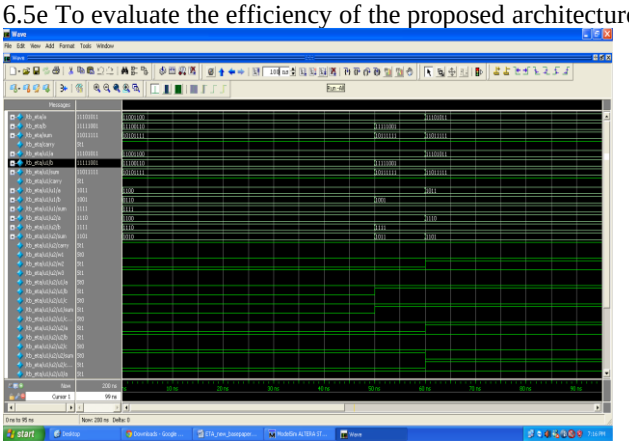

Fig 9: Simulation Wave form for Error Tolerant Adder.

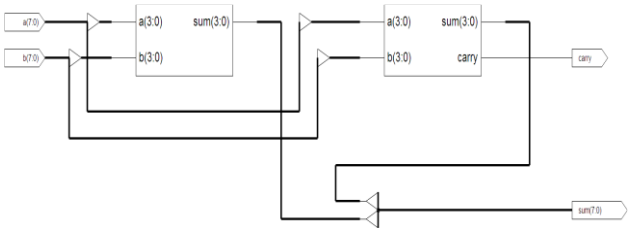

Fig 10: Synthesis block for Error Tolerant Adder Top Module.

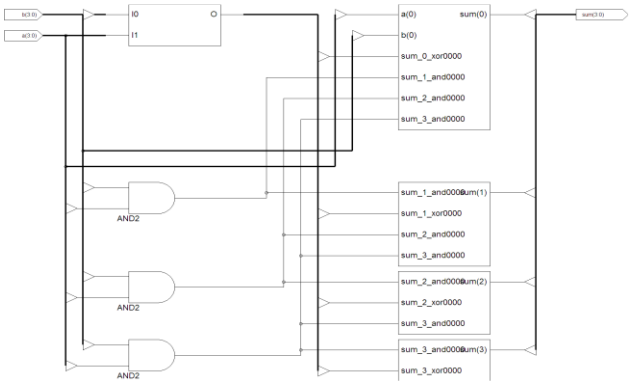

Fig 10: Synthesis block for Error Tolerant Adder Sub Module

To demonstrate the advantages of the proposed ETA, we simulated the ETA along with four types of conventional adders, i.e., the RCA ,CSK, CSL, and CLA, using HSPICE. All the circuits were implemented using Xilinx Spartarn 3 XC3S50 FPGA Family. The input frequency was set to 100 MHz, and the simulation results Shown Below Modelsim 6.5e Tool used for simulation of the project and Xilinx ISE 9.2 used synthesize the design, The synthesized Results of shown blow figures .Here we get the 96 to 98% accuracy results in addition of two numbers.

V. APPLICATIONS

In image processing and many other DSP applications, fast Fourier transformation (FFT) is a very important function. The computational process of FFT involves a large number of additions and multiplications. It is therefore a good platform for embedding our proposed ETA. To prove the feasibility of the ETA, we replaced all the common additions involved in a normal FFT algorithm with our proposed addition arithmetic. As we all know, a digital image is represented by a matrix in a DSP system, and each element of the matrix represents the color of one pixel of the image. To compare the quality of images processed by both the conventional FFT and the inaccurate FFT that had incorporated our proposed ETA.

VI. CONCLUSION

In this study, the concepts of error tolerance are used in design of shift-and add multiplier and Image processing applications. The proposed Error Tolerant Adder trades a certain amount of accuracy for significant power saving and performance improvement. Extensive comparisons with conventional Adders showed that the proposed ETA outperformed the conventional Adders Applications 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 performance are more important than accuracy. Few such applications are in Digital Image processing and DSP architectures for portable devices such as cell phones and laptops. In this paper, the concept of error tolerance is introduced in VLSI design. The potential applications of the ETA fall mainly in areas where there is no strict requirement on accuracy or where super low power consumption and high-speed performance are more important than accuracy.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

VII. ACKNOWLEDGWMENT

Tadigiri Aruna would like to thank Mr.R.Bhadraiah, Assistant professor, in the Department of ECE, Nimra college of Engg&Technology who had been guiding throughout the project and supporting me in giving technical ideas about the paper and motivating me to complete the work efficiently and successfully.

REFERENCES

1. M. A. Breuer, "Intelligible test techniques to support error-tolerance," in *Proc. Asian Test Symp.*, Nov. 2004, pp. 386–393.
2. K. J. Lee, T. Y. Hsieh, and M. A. Breuer, "A novel testing methodology based on error-rate to support error-tolerance," in *Proc. Int. Test Conf.*, 2005, pp. 1136–1144.
3. S. Chong and A. Ortega, "Hardware testing for error tolerant multimedia compression based on linear transforms," in *Proc. Defect and Fault Tolerance in VLSI Syst. Symp.*, 2005, pp. 523–531.
4. H. Chung and A. Ortega, "Analysis and testing for error tolerant motion estimation," in *Proc. Defect and Fault Tolerance in VLSI Syst. Symp.*, 2005, pp. 514–522.
5. J. E. Stine, C. R. Babb, and V. B. Dave, "Constant addition utilizing flagged prefix structures," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, 2005.
6. L.-D. Van and C.-C. Yang, "Generalized low-error area-efficient fixed width multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 25,no. 8, pp. 1608–1619, Aug. 2005.
7. M. Lehman and N. Burla, "Skip techniques for high-speed carry propagation in binary arithmetic units," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 691–698, Dec. 1962.
8. O. Bedrij, "Carry select adder," *IRE Trans. Electron. Comput.*, vol. EC-11, pp. 340–346, 1962.
9. O. MacSorley, "High speed arithmetic in binary computers," *IRE Proc.*, vol. 49, pp. 67–91, 1961.
10. Y. Kiat-Seng and R. Kaushik, *Low-Voltage, Low-Power VLSI Subsystems.* New York: McGraw-Hill, 2005.
11. 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.