American Journal of Applied Sciences 9 (6): 818-824, 2012
ISSN 1546-9239

Design of an Error Tolerant Adder

[1]Jayanthi, A.N. and [2]C.S. Ravichandran
[1]Department of Electronics and Communication Engineering,
Sri Ramakrishna Institute of Technology, Coimbatore, India
[2]Department of Electrical and Electronics Engineering,
SSK College of Engineering and Technology Coimbatore, India

Abstract: Problem statement: 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 improvement in the Power-Delay Product (PDP). **Conclusion/Recommendations:** One important potential application of the proposed ETA is in digital signal processing systems that can tolerate certain amount of errors. Delay and power are compared for various adders like RCA and CLA. It is found that ETA has high speed and less power compared to its counterparts.

Key words: Error-Tolerant Adder (ETA), Error Tolerance (ET), Power-Delay Product (PDP), Minimum Acceptable Accuracy (MAA), Least Significant Bit (LSB)

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 nondigital worldly experiences. The world accepts "analog computation," which generates "good enough" results rather than totally accurate results (Breuer, 2005). 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 International Technology Roadmap for Semiconductors. Based on the characteristic of digital VLSI design, some novel concepts and design techniques have been proposed. The concept of Error Tolerance (ET) (Breuer and Zhu, 2006; Breuer *et al*., 2004; Breuer, 2004; Lee *et al*., 2005; Chong and Ortega, 2005; Chung and Ortega, 2005; Kuok, 1995; Hsieh *et al*., 2007) and the PCMOS technology (Palem, 2005; Cheemalavagu *et al*., 2004; Korkmaz *et al*., 2006) 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. The "imperfect" attribute seems to be not appealing. However, the need for the error-tolerant circuit (Breuer and Zhu, 2006; Breuer *et al*., 2004; Breuer, 2004; Lee *et al*., 2005; Chong and Ortega, 2005; Chung and Ortega, 2005; Kuok, 1995; Hsieh *et al*., 2007) was foretold in the 2003 International Technology Roadmap for Semiconductors (ITRS) International Technology Roadmap for Semiconductors. To deal with error-tolerant problems, some truncated adders/multipliers have been reported (Stine *et al*., 2005; Van and Yang, 2005) but are not able to perform well in its speed, power, area, or accuracy. The "flagged prefixed adder" (Stine *et al*., 2005) performs better than the nonflagged 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" (Van and Yang, 2005), 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.

Corresponding Author: Jayanthi, A.N., Department of Electronics and Communication Engineering,
Sri Ramakrishna Institute of Technology, Coimbatore, India

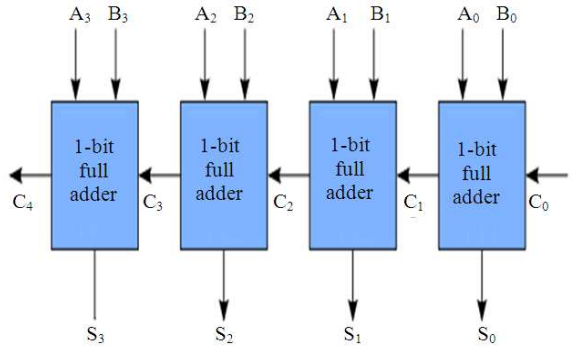

Fig. 1: Block diagram of 4 Bit Ripple Carry

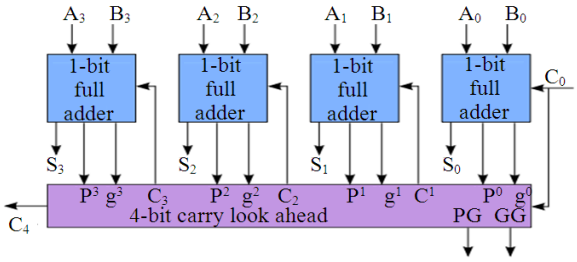

Fig. 2: Block diagram of 4bit carry-look-ahead adder

Table 1: Comparison of delay in adders

No. of bits/adder	RCA (ns)	CLA (ns)	ETA (ns)
4 bits	11.953	11.989	7.570
8 bits	18.607	18.453	8.156
12 bits	25.247	24.917	8.300
16 bits	31.887	31.381	7.913
32 bits	31.815	57.237	9.411

Table 2: Comparison of power in adders

No. of bits/adder	RCA	CLA	ETA
4 bits	2.199	1.787	0.073
8 bits	4.988	142.000	0.016
12 bits	8.237	74.523	0.001
16 bits	0.142	1294.000	0.021
32 bits	0.028	0.135	0.014

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).

Conventional adders:

Ripple-Carry Adder (RCA): The n-bit adder built from n one-bit full adders is known as a ripple carry adder, because of the way the carry is computed. Each full adder inputs a C_{in}, which is the C_{out} of the previous adder. This kind of adder is a ripple carry adder, since each carry bit "ripples" to the next full adder. Block diagram of Ripple Carry Adder is as in Fig. 1.

The layout of ripple carry adder is simple, which allows for fast design time; however, the ripple carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder. The gate delay can easily be calculated by inspection of the full adder circuit. Each full adder requires three levels of logic. In a 32-bit (ripple carry) adder, there are 32 full adders, so the critical path (worst case) delay is 31 * 2(for carry propagation) + 3(for sum) = 65 gate delays. Table 1 and 2 shows the result obtained for RCA.

Carry-look-ahead adder CLA: Carry look ahead logic uses the concepts of generating and propagating carries. The addition of two 1-digit inputs A and B is said to generate if the addition will always carry, regardless of whether there is an input carry. In the case of binary addition, A+B generates if and only if both a and B are 1. The addition of two 1-digit inputs A and B is said to propagate if the addition will carry whenever there is an input carry. The propagate and generate are defined with respect to a single digit of addition and do not depend on any other digits in the sum. In the case of binary addition, A+B propagates if and only if at least one of A or B is 1. Sometimes a slightly different definition of propagate is used. By this definition, A+B is said to propagate if the addition will carry whenever there is an input carry, but will not carry if there is no input carry. For binary arithmetic, or is faster than xor and takes fewer transistors to implement. However, for a multiple-level carry look ahead adder, it is simpler to use. Block Diagram of 4bit carry-look-ahead adder is as in Fig. 2.

The carry look ahead adder represents the most widely used design for high-speed adders in modern Computers. The advantage of using a look-ahead design over a ripple carry adder is that the Look-ahead is faster in computing the solution. The carry-in values in a carry look-ahead design are calculated independent of each other through a series of logic circuits.

Carry look ahead depends on two things:

- Calculating, for each digit position, whether that position is going to propagate a carry if one comes in from the right
- Combining these calculated values so as to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right
- Supposing that groups of 4 digits are chosen

Then the sequence of events goes something like this:

- All 1-bit adders calculate their results. Simultaneously, the look ahead units perform their calculations.

- Suppose that a carry arises in a particular group. Within at most 3 gate delays, that carry will emerge at the left-hand end of the group and start propagating through the group to its left
- If that carry is going to propagate all the way through the next group, the look ahead unit will already have deduced this. Accordingly, before the carry emerges from the next group the look ahead unit is immediately (within 1 gate delay) able to tell the next group to the left that it is going to receive a carry - and, at the same time, to tell the next look ahead unit to the left that a carry is on its way Table 1 and 2 shows the result obtained for CLA

Error-tolerant adder: Before detailing the ETA, the definitions of some commonly used terminologies shown in this study are given as follows:

- Overall error (OE) OE = R_c-R_E, where R_E, is the result obtained by the adder 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 adder is used to indicate how "correct" the output of an adder is for a particular input. It is defined as: ACC = (1-(OE/ R_c))/100%
- Its value ranges from 0-100%.
- Minimum Acceptable Accuracy (MAA): Although some errors are allowed to exist at the output of an ETA, the accuracy of an acceptable output should be "high enough" (higher than a threshold value) to meet the requirement of the whole system. Minimum acceptable accuracy is just that threshold value. The result obtained whose accuracy is higher than the minimum acceptable accuracy is called acceptable result.
- Acceptance Probability (AP): Acceptance probability is the probability that the accuracy of an adder is higher than the minimum acceptable accuracy.

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 R_c fast adders, such as the Carry-Skip Adder (CSK) (Lehman and Burla, 1961), Carry-Select adder (CSL) (Bedrij, 1962) and Carry-Look-Ahead adder (CLA) (MacSorley, 1961), have been developed. Also, there are many low-power adder design techniques that have been proposed (Yeo and Roy, 2005) 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.

ETA design was proposed in Zhu *et al*. (2010). Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing. IEEE Transactions on Very Large Scale Integration (VLSI) Syst., 18: 8. However there are tradeoffs between speed and power.

Proposed addition arithmetic: 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. In this study, 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. 3.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. 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. To minimize the overall error due to the elimination of the carry chain, a special strategy is adapted and can be described as follow: (1) check every bit position from left to right (MSB to LSB); (2) if both input bits are "0" or different, normal one-bit addition is performed and the operation proceeds to next bit position; (3) if both input bits are "1,"the checking process stopped and from this bit onward, all sum bits to the right are set to "1." Table 1 and 2 shows the result obtained for RCA.

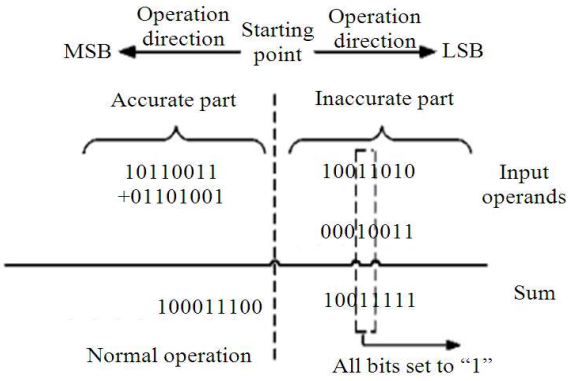

Fig. 3: Proposed addition arithmetic

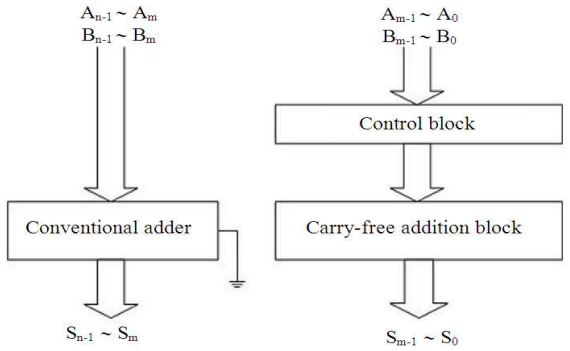

Fig. 4: Hardware implementation of proposed ETA

Hardware implementation: The block diagram of the hardware implementation of such an ETA that adopts our proposed addition arithmetic is provided in Fig. 4 This most straightforward structure consists of two parts: an accurate part and an inaccurate part. The accurate part is constructed using a conventional adder such as the RCA, CSK, CSL, or CLA. The carry-in of this 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.

Design of a 32-bit error-tolerant adder:

Strategy of dividing the 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%.

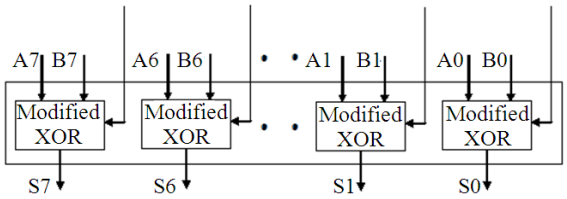

Fig. 5: Overall structure of carry-free addition block

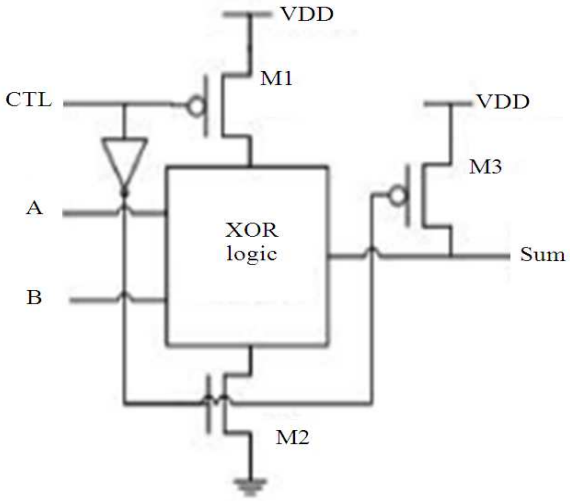

Fig. 6: Schematic diagram of modified XOR gate

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.

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.

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 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 and 6.

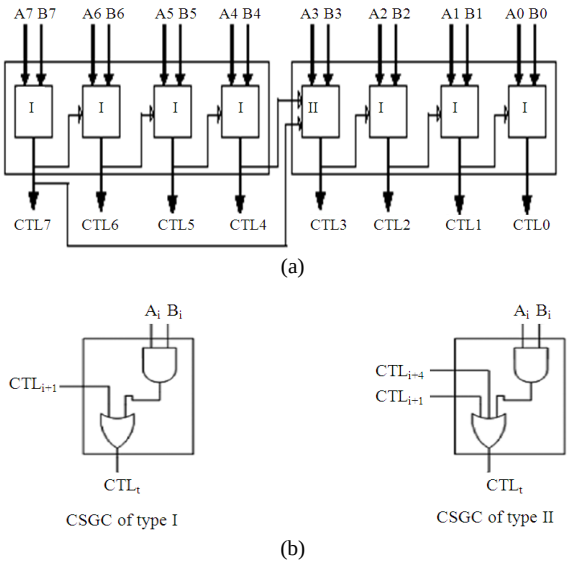

Fig. 7: (a) Overall architecture of control block (b) CSGC

In the modified XOR gate, three extra transistors, M1, M2 and M3, are added to a conventional XOR gate. CTL is the control signal coming from the control block of Fig. 7 and is used to set the operational mode of the circuit. When CTL = 0, M1 and M2 are turned on, while M3 is turned off, leaving the circuit to operate in the normal XOR mode. When CTL = 1 M1 and M2 are both turned off, while M3 is turned on, connecting the output node to VDD and hence setting the sum output to "1." The function of the control block is to detect the first bit position when both input bits are "1," and to set the control signal on this position as well as those on its right to high. It is made up of 20 Control Signal Generating Cells (CSGCs) and each cell generates a control signal for the modified XOR gate at the corresponding bit position in the carry-free addition block. Instead of a long chain of 20 cascaded GSGCs, the control block is arranged into five equal-sized groups, with additional connections between every two neighboring groups. Two types of CSGC, labeled as type I and II in Fig. 7a are designed and the schematic implementations of these two types of CSGC are provided in Fig. 7b.

The control signal generated by the leftmost cell of each group is connected to the input of the leftmost cell in next group. The extra connections allow the propagated high control signal to "jump" from one group to another instead of passing through all the 20 cells. Fig 8 shows the propagation delay that take place in gates.

Xilinx ISE was used to obtain the delay of all the 3 adders and simulation results are taken for 3 adders as shown in Fig. 10-12. Synthesis reports are plotted as in Fig. 9. Power Vs adders are noted in Fig. 13. ETA shows less power consumption.

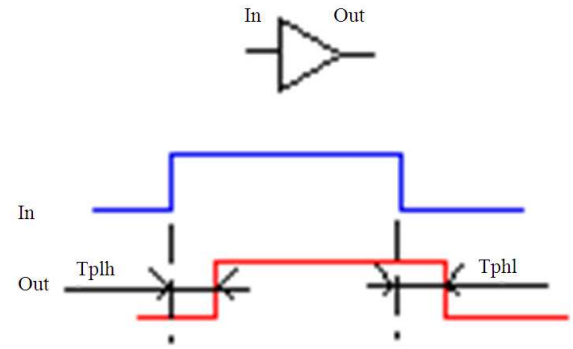

Fig. 8: Propagation delay of gates

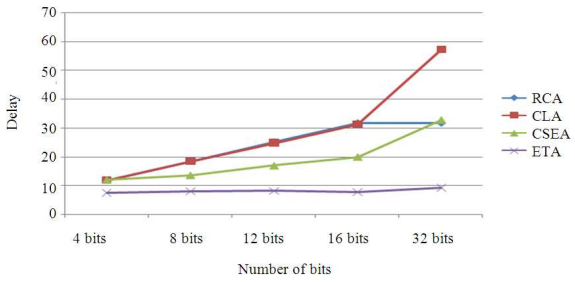

Fig. 9: Delay Vs no of bits in adders

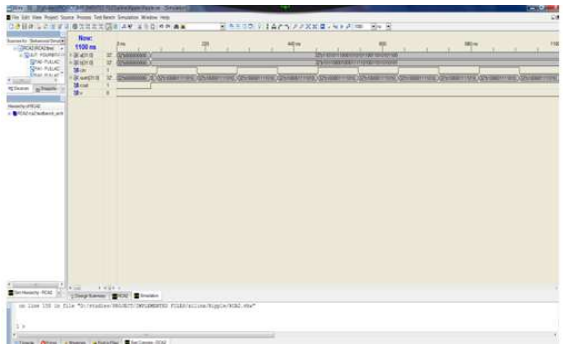

Fig. 10: 32 bit RCA Simulation result

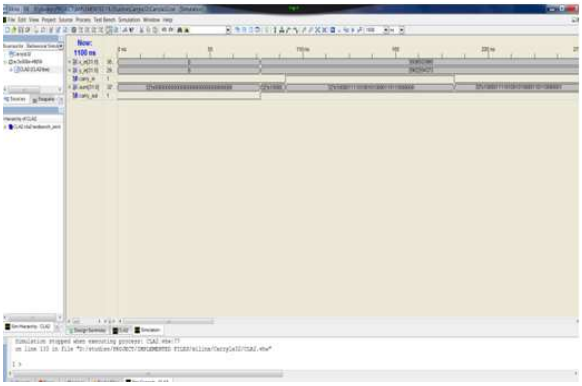

Fig. 11: 32 bit CLA simulation result

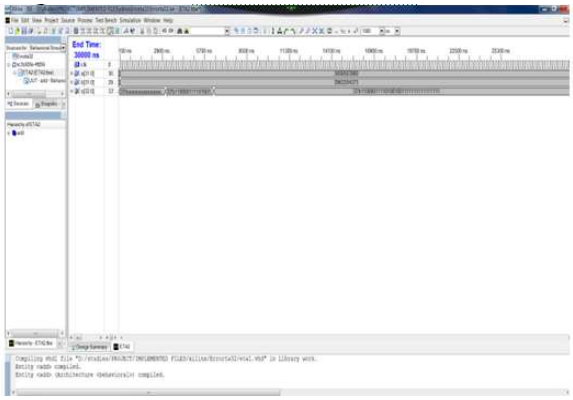

Fig. 12: 32 bit ETA Simulation result

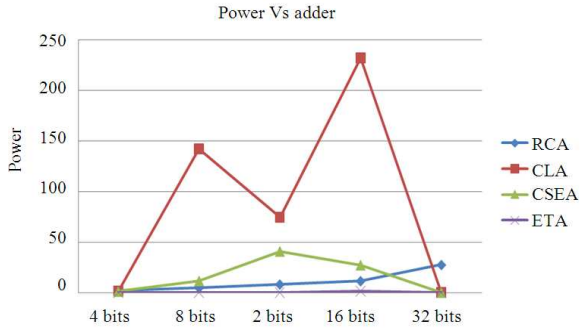

Fig. 13: Power Vs Adders

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.

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.

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.

CONCLUSION

The 'error tolerant adder' was thus designed with an idea to minimize the delay and power consumption .The ETA was tested using the Xilinx ISE and was compared with the other conventional adders such as the Ripple carry adder, carry look ahead adder. The power consumption of the ETA was calculated using the Micro wind/DSCH tool.

Extensive comparisons with conventional digital adders showed that the proposed ETA outperformed the conventional adders in both power consumption and speed performance. 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.

In future the error tolerant adder can be used in the DSP application for portable devices such as cell phones, laptops.

REFERENCES

Breuer, M.A., 2004. Intelligible test techniques to support error-tolerance. Proceedings of the International Conference Asian Test Symposium, Nov. 15-17, IEEE Xplore Press, pp: 386-393. DOI: 10.1109/ATS.2004.51

Breuer, M.A., S.K. Gupta and T.M. Mak, 2004. Defect and error tolerance in the presence of massive numbers of defects. IEEE Des. Test Comp., 21: 216-227. DOI: 10.1109/MDT.2004.8

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

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

Cheemalavagu, S., P. Korkmaz and K.V. Palem, 2004. Ultra low energy computing via probabilistic algorithms and devices: CMOS device primitives and the energy-probability relationship. Proceedings of the International Conference Solid State Devices and Materials, (SSDM' 04), Tokyo, Japan, pp: 402-403.

Chong, I.S. and A. Ortega, 2005. Hardware testing for error tolerant multimedia compression based on linear transforms. Proceedings of the 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Oct. 3-5, IEEE Xplore Press, pp: 523-531. DOI: 10.1109/DFTVS.2005.38

Chung, H. and A. Ortega, 2005. Analysis and testing for error tolerant motion estimation. Proceedings of the Defect Fault Tolerance in VLSI System Symposium, Oct. 3-5, IEEE Xplore Press, pp: 514-522. DOI: 10.1109/DFTVS.2005.19

Hsieh, T.Y., K.J. Lee and M.A. Breuer, 2007. Reduction of detected acceptable faults for yield improvement via error-tolerance. Proceedings of the conference on Design, automation and test in Europe, April 16-20, ACM, USA., pp: 1-6.

Korkmaz, P., B.E.S. Akgul, K.V. Palem and L.N. Chakrapani, 2006. Advocating noise as an agent for ultra-low energy computing: Probabilistic complementary metal-oxide-semiconductor devices and their characteristics. Japan. J. Applied Phys., 45: 3307-3316.

Kuok, H.H., 1995. Audio recording apparatus using an imperfect memory circuit. U.S. Patent, 5: 414-758.

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 Conference Test Conference, Nov. 8-8, IEEE Xplore Press, Austin, pp: 1136-1144. DOI: 10.1109/TEST.2005.1584081

Lehman, M. and N. Burla, 1961. Skip techniques for high-speed carry-propagation in binary arithmetic units. IRE Trans. Elect. Comput., 10: 691-698. DOI: 10.1109/TEC.1961.5219274

MacSorley, O., 1961. High speed arithmetic in binary comp. IRE Proc., 49: 67-91.

Palem, K.V., 2005. Energy aware computing through probabilistic switching: A study of limits. IEEE Trans. Comput., 54: 1123-1137. DOI: 10.1109/TC.2005.145

Stine, J.E., C.R. Babb and V.B. Dave, 2005. Constant addition utilizing flagged prefix structures. Proceedings of the International Conference of Symposium Circuits and Systems (ISCAS), May 23-26, IEEE Xplore Press, pp: 668-671. DOI: 10.1109/ISCAS.2005.1464676

Van, L.D. and C.C. Yang, 2005. Generalized low-error area-efficient fixed-width multipliers. IEEE Trans. Circ. Syst. I, Reg. Papers, 52: 1608-1619. DOI: 10.1109/TCSI.2005.851675

Yeo, K.S. and K. Roy, 2005. Low Voltage, Low Power VLSI Subsystems. 1st Edn., McGraw-Hill, New York, ISBN-10: 007143786X pp: 293.

Zhu, N., W.L. Goh, W. Zhang, K.S. Yeo, Z.H. 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. Syst., 18: 1225-1229. DOI: 10.1109/TVLSI.2009.2020591