Increasing the Performance of Integrated Circuits by Using Error Lenient Adder

R. Seetha, G. Lakshmi Bhavani and CH. Ramesh

***Abstract*—The probability of errors in the present VLSI technology is very high and it is increasing with technology scaling. Removing all errors is very expensive task and is not required for certain applications. There are certain application where the approximate result is acceptable e.g. image processing and video processing. For these applications Error Tolerant Adder (ETA) is proposed which provide approximate result at very high speed than the convention adder. The proposed adder provides improvement in delay, power and area at the same time at the cost of accuracy. Simulation result shows improvement in delay, power and area respectively over convention adder.**

***Keywords*:** Adder, Error Tolerant, low power, Performance

I. INTRODUCTION

As per the generally accepted digital VLSI design, one assumes that a usable circuit should always provide definite and accurate results. But generally, such perfect operations are rarely and not often required in our non-digital worldly experiences. The world accepts the analog communication for its tendency of generating good results. Most of the digital systems generate errors. In many applications, such as a communication system, the analog signal undergoes the process of sampling before converting into a digital data. The digital data is then processed and transmitted in a noisy channel before converting back to an analog signal. During this process of transmission, errors may occur anywhere. Based on the characteristic of digital VLSI design, some new concepts and design techniques have been proposed. The concept of error tolerance (ET) is one 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.. However, the need for the error-tolerant circuit was foretold in the 2003 International Technology Roadmap for Semiconductors (ITRS). To handle the error-tolerant problems, some truncated adders/multipliers have been described, but are not able to perform well either in its speed, power, area, or accuracy. Of course, not all digital systems can engross the error-tolerant concept.

R.Seetha, G.Lakshmi Bhavani and CH.Ramesh are with Bandari Srinivas Institute of Technology, chevella, R.R.Dist, Email: seetha2409@gmail.com, lakshmisrinivas.2010@gmail.com, ramesh.chinni414@gmail.com

In digital systems such as control systems, the correctness of the output signal is extremely important, and this refuses 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.

II. CONVENTIONAL ADDER

RIPPLE CARRY ADDER :

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.

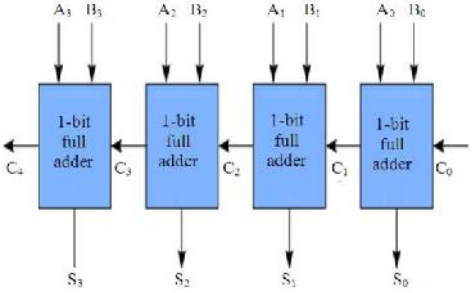

Fig. 1 Ripple carry adder

CARRY LOOK AHEAD:

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.

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.

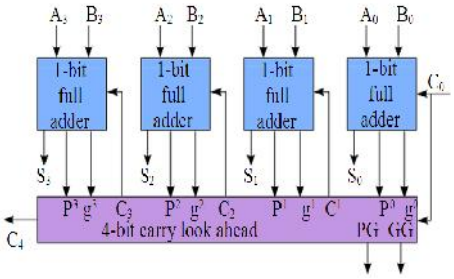

Fig.2 Carry look ahead

Table 1: Comparison of delay in adders

No.of bits/adder	RCA(ns)	CLA(ns)	ETA(ns)
4bits	11.953	11.989	7.570
8bits	18.607	18.453	8.156
12 bits	25.247	24.917	8.300
16 bits	31.887	31.381	7.913
32 bits	38.815	37.237	9.411
64 bits	45.455	43.837	10.139
128 bits	52.095	51.297	10.984

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
64 bits	0.139	19.178	0.007
128 bits	0.025	333.002	0.014

III. ERROR TOLERANT ADDER

Before detailing the ETA, the definitions of some commonly used terminologies shown in this paper are given as follows.

• Overall Error (OE):

OE=|Rc -Re|, where 'Re' is the result obtained by the adder, 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 adder is used to indicate how "correct" the output of an adder is for a particular input.

It is defined as: ACC = (1-(OE / Rc))*100%. Its value ranges from 0% to 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. It can be expressed as

AP = P (ACC>MAA), with its value ranging from 0 to 1.

IV. NEED OF ERROR TOLERANT ADDER

As per the present requirements such as instant and precise responses also considering the huge data sets to be transferred the conventional adders could not satisfy all these. The traditional Ripple-Carry Adder (RCA) is therefore no longer suitable for large adders because of its low-speed performance. Many different types of RC fast adders, such as the Carry-Skip Adder (CSK),Carry-Select adder (CSL) and Carry-Look-Ahead adder (CLA) , have been developed. Also, there are many low-power adder design techniques that have been suggested However, there are always compromise between speed and power. The error-tolerant design can be a influential solution to this problem. By sacrificing some accuracy, the ETA can obtain great improvement in both the power consumption and speed performance. ETA design was proposed in Zhu et al. (2010).

V. PROPOSED ADDITION

Generally, in a adder circuit, the carry chain along the critical path, from the least significant bit (LSB) to the most significant bit (MSB) generated is the main cause for the delay. 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 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.3

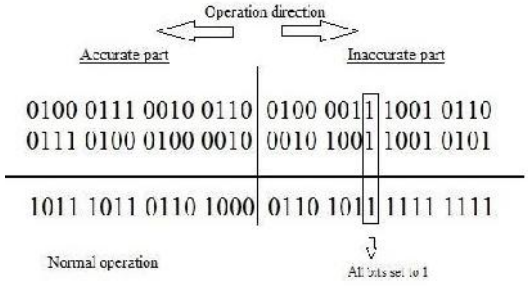

Fig. 3 Proposed additioin

.

We first split the input operands into two parts:1) An accurate part that includes several higher order bits and 2) 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 Fig1.1, the two 32-bit input operands, A= "01000111001001100100001110010110" (47264396) and B= "01110100010000100010100110010101" (74422995), 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."

VI. Hardware Implementation

The block diagram of the ETA that adopts our proposed addition arithmetic is provided in Fig. 2. This 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 contains 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.

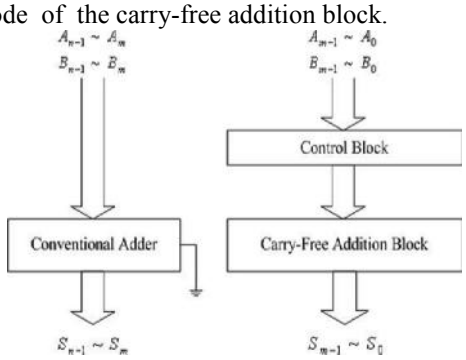

Fig. 4 Hardware implementation

Design of 32-bit ETA

A) Concept to divide into Accurate and Inaccurate part

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. First, we define the delay of the proposed adder as T_d= max (T_h, Tl) $T_d = \max(T_h, T_l)$, where T_h is the delay in the accurate part and T_l is the delay in the inaccurate part. With the proper dividing strategy, we can make T_h approximately equal to T_l and hence achieve an optimal time delay. 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. We divided the 32-bit adder by putting12 bits in the accurate part and 20 bits in the inaccurate part

B) 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.

C) 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 Figure 5.

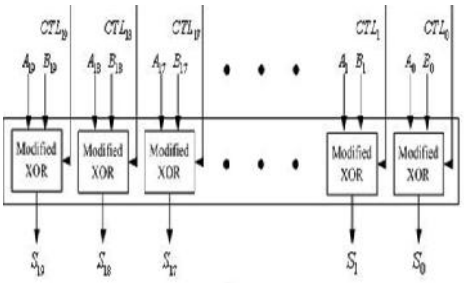

Fig. 5 Carry free addition overall architecture

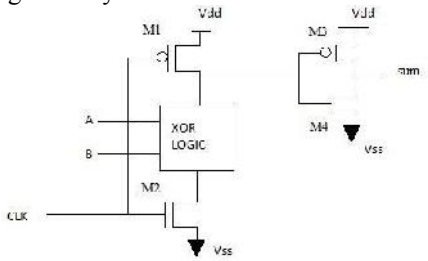

Fig. 6 modified XOR logic using CMOS domino logic

In the modified XOR gate, four extra transistors, M1, M2, M3 and M4 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 and M4 is turned on, 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 and M4 is turned off, 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.

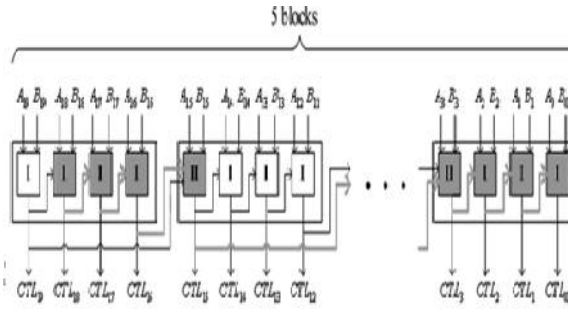

Fig. 7 Control block overall architecture

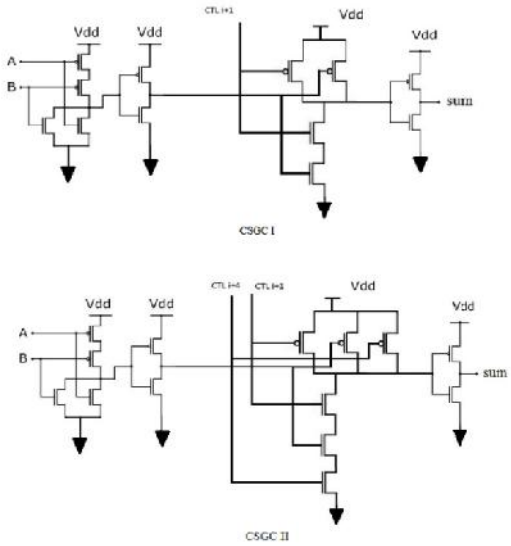

Fig. 8 Schematic implementation of CSGC

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. 7 are designed, and the schematic implementations of these two types of CSGC are provided in Fig. 8. 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. Hence, the worst case propagation path [shaded in gray in Fig. 7] consists of only ten cells.

VII. DELAY IN ADDERS

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.

VIII. POWER CONSUMPTION IN ADDERS

Addition is a common operation 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[6], 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 [2]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[4], 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 energydissipation. 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 [2].

IX. SIMULATION ENVIRONMENT AND RESULTS

To show the effectiveness of the proposed ETA adder we have designed and simulated no. of accurate adders (RCA, CSK, CSL) using TANNER tool. The Simulation result as shown in the table 3, show that the proposed adder performs better than all the available convention adders. Tanner tool software was used to construct Error Tolerant Adder (ETA) and other conventional adders.45nm technology is used for designing the all adders. For each set of input, we run the simulation for each adder and recorded the power consumption. The worst case input was calculated and used to simulate the delay. The transistor count was derived directly from the Tanner tool software. Comparing all the simulation results of our proposed ETA with those of the conventional adders (see Table III), it is evident that the ETA performed the best in terms of power consumption, delay, and Power-Delay Product (PDP).Error Tolerant Adder has same cost as Ripple Carry Adder.

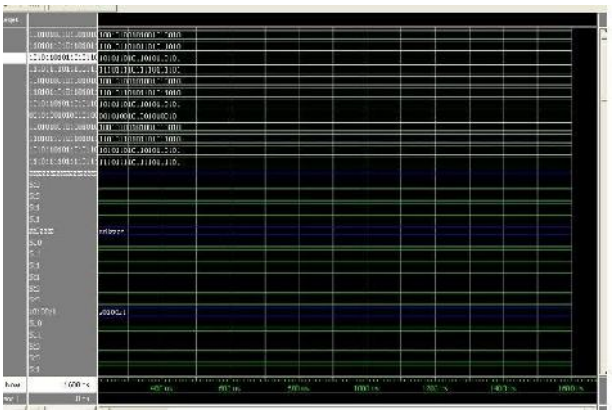

Fig 9: Simulation results

Table 3:

Type of Adder	Power (uw)	Delay (ns)	PDP	Transistor Count
RCA	27.25	5.18	158.32	896
CSK	38.13	2.25	85.79	1442
CSL	42.92	1.96	84.26	2100
ETA	31.63	2.15	68.00	924

X. CONCLUSION

In this paper, we proposed the concept of Error Tolerant Adder (ETA).With small loss in accuracy; provide tremendous improvement in power, delay and area. The Error Tolerant Adder (ETA) provides high the speed by cutting down the carry propagation. Comparisons with conventional digital adders showed that the proposed ETA performed better than the all conventional adders in both power consumption and speed performance. ETA can be used in applications where there is no strict requirement of accuracy or where power consumption and high-speed performance low are more important than accuracy. Example of such applications such as the image processing and speech processing ,DSP application for portable devices such as cell phones and laptops.

REFERENCE(S)

[1]. Y. Kiat-Seng and R. Kaushik, *"*Low-Voltage, Low-Power VLSI Subsystems*"*. New York: McGraw-Hill, 2005.

[2]. A. B. Melvin, "*Let's think analog*," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, 2005, pp. 2–5.

[3]. Ning Zhu, et. al "Design of Low-Power High-Speed Truncation-Error- Tolerant Adder and Its Application in Digital Signal Processing," IEEE Trans. on VLSI., vol. 18, no. 8, pp. 1225–1229, Aug. 2010.

[4]. Ning Zhu, et. al "An Enhanced Low-Power High-Speed Adder For Error-Tolerant Application," International symposium on IC., vol. 18, no. 8, pp. 69–72, Aug. 2009.

[5]. A. B. Melvin and Z. Haiyang, "Error-tolerance and multi-media,"in Proc. 2006 Int. Conf. Intell. Inf. Hiding and Multimedia Signal Process. 2006, pp. 521–524.

[6]. M. A. Breuer, S. K. Gupta, and T. M. Mak, "Design and error tolerance in the presence of massive numbers of defects," IEEE Des.TestComput., vol. 24, no. 3, pp. 216–227, May-Jun. 2004.

[7]. M. A. Breuer, "Intelligible test techniques to support errortolerance," in Proc. Asian Test Symp., Nov. 2004, pp. 386–393.

[8]. 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.

[9]. I. S. Chong and A. Ortega, "Hardware testing for error tolerant multimedia compression based on linear transforms," in Proc. Defect andFault Tolerance in VLSI Syst. Symp., 2005, pp.

[10]. 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.

[11]. M. Lehman and N. Burla, "Skip techniques for high-speed carry propagation in binary arithmetic units," IRE Trans. Electron.Comput., vol. EC-10, pp.

FIRST AUTHOR: R.SEETHA, received the B.TECH degrees in electronics and communication engineering and M.TECH degree in VLSI system design from JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD in 2005 and 2010 respectively.I am working as associate professor in Bandari Srinivas Institute Of Tech.Chevella,R.R.Dist.

SECOND AUTHOR: G.LAKSHMI BHAVANI, received the B.TECH degree in Electronics and Communication Engineering and M.TECH degree in VLSI System Design from JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD in 2010 and 2012 respectively. I am working as an assistant professor in Bandari Srinivas Institute Of Tech.Chevella,R.R.Dist.

THIRD AUTHOR: CH.RAMESH, received the B.TECH degrees in electronics and communication engineering from JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD in 2010 ,presently working as an assistant professor in Bandari Srinivas Institute Of Tech.Chevella,R.R.Dist.