THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

High Performance 64-bit Error Tolerent Adder Using Cadence Tool

Dr. Tejo Murthy P. H. S.
Associate Professor, Department EIE, GITAM, Viskapatanam, Andhra Pradesh, India
Sanjay V. Chowdhary
Assistant Professor, Department of ECE, SKSVMACET, Lakshmeshwara, Karnataka, India
Srinivasarao Udara
Assistant Professor, Department of ECE, STJIT, Ranebennur, Karnataka, India

Abstract:
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 can be achieved, Delay and Power estimation for different number of bits is estimated.The power and area almost 50% reduced compared to conventional adder.

Keywords: *Cadence for Synthesis (Digital Design) , Xilinx 14.1 for Simulation, Conventional Adder, Error Tolerent Adder*

1. Introduction

Adder 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. In analog computations, generation of "good enough" results is more important than totally accurate results .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.

1.1. Conventional Adder

Ripple-Carry Adder (RCA): The n-bit adder is 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 Cin, which is the Cout 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.

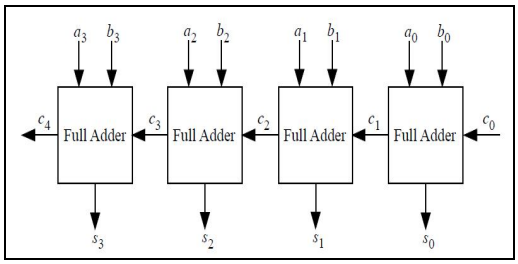


Figure 1: fig .1 4-bit Ripple Carry Adder

The 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. 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.

Carry-look-ahead adder CLA: Carry look a-head 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 a-head adder, it is simpler to use. Block Diagram of 3 bit carry-look-ahead adder is as in Fig. 2.

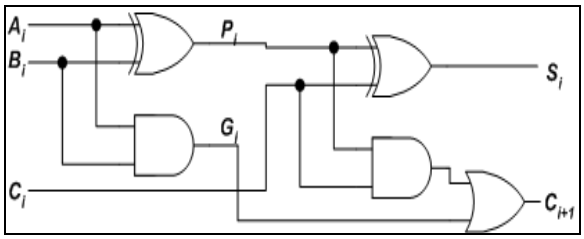


Figure 2: Carry-look-ahead-adder

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.

2. 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 = 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 0-100%.
- 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

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..

To minimize error due to the elimination of the carry chain: 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." This is how this adder Saves carry propagation delay and enhances the overall performance.

Design of the accurate part: Ripple carry addition is the most power saving conventional addition technique .the Ripple carry adder is built from cascading the full adders in series the full adder block diagram is as shown below fig 3.

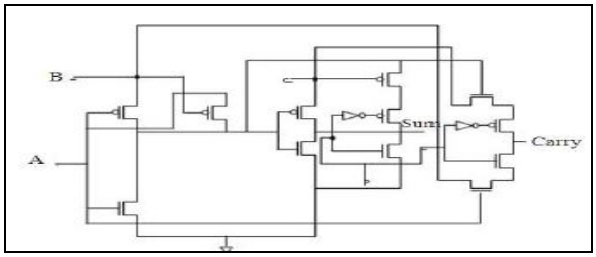
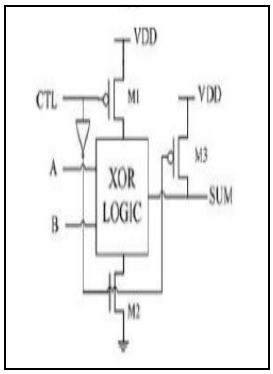


Figure 3: Full Adder

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 function of the control block 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 .The block diagram of the carry free addition block and the schematic implementation of the modified XOR gate are shown in the fig below fig.4 and it's control logic block diagram shown in fig.5.



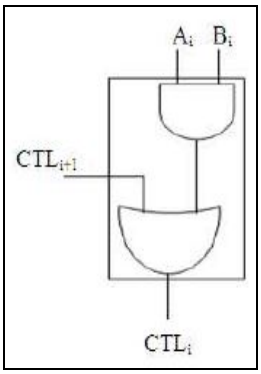


Figure 4: Modified XOR Gate Figure 5: Contrl Block

The modified xor gates are then combined together to obtain the carry free addition block as shown the fig.6.
The Entire BLOCK DIAGRAM Of Carry Free Addition

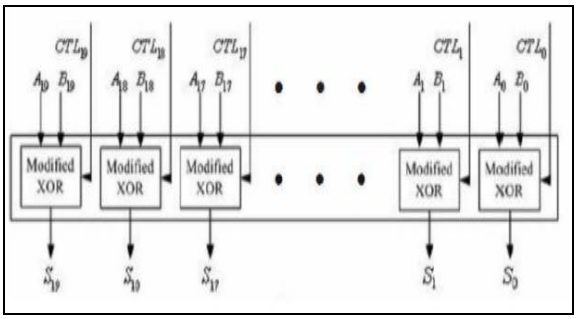


Figure 6: Carry free addition block

Design and results of Accurate Part: (MSB): Since the delay and complexity of the Ripple Carry adder is less when compared to any other adder such as Carry look-ahead adder or Carry select adder, we prefer designing of accurate part of our project with the ripple carry adder. And is shown in fig .1.

3. Simulation & Result Analysis
a) For 4-bit ETA: Simulated Waveform for 4-bit ETA

Figure 4: Simulated result

b) for 8-bit ETA: Simulated waveform for 8-bit ETA

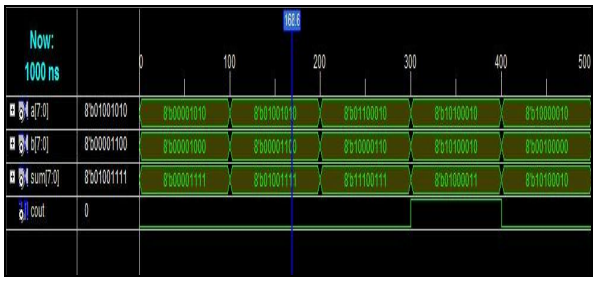


Figure 5: Simulated result

c)The simulated waveform for 16-bit ETA

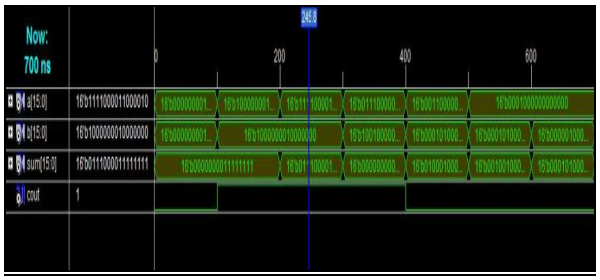


Figure 6: simulated result

d) Simulated waveforms for 16 bit RCA

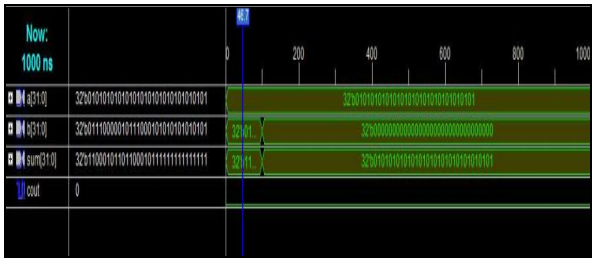


Figure 7: .Simulated result

Results: Comparison Table between 4-bit to 64-bit ETA

5. Performance Comparison

Parameter	Conventional RCA	ETA with RCA	Conventional CLA	ETA with CLA
Power (nano watt)	2653.57	829.97	2653.57	829.977
Delay (pico sec)	524	237	483	319
Area (No. of Cells)	71	38	71	38

5.1. Comparison Table of 4 bit adders

Parameter	Conventional RCA	ETA with RCA	Conventional CLA	ETA with CLA
Power (nano watt)	5720.05	2125.727	2230.09	951.072
Delay (pico sec)	1156	593	1065	553
Area (No. of Cells)	141	82	84	58

5.2. Comparison Table of 8 bit Adders

Parameter	Conventional RCA	ETA with RCA	Conventional CLA	ETA with CLA
Power (nano watt)	10804.955	5298.682	4209	1978.837
Delay (pico sec)	2529	1306	2021	1451
Area (No. of Cells)	276	198	264	126

5.3. Comparison of 16 bit Adders

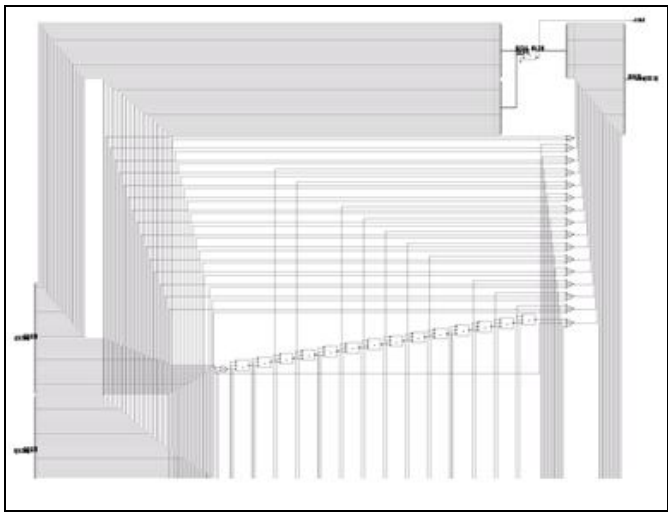
Parameter	Conventional RCA	ETA with RCA	Conventional CLA	ETA with CLA
Power (nano watt)	21408.85	11474.793	16543.65	7591.749
Delay (pico sec)	5529	2730	5005	3283
Area (No. of Cells)	814	418	724	389

5.4. Comparison of 32 bit Adders

Parameter	Conventional RCA	ETA with RCA	Conventional CLA	ETA with CLA
Power (nano watt)	42408.65	22674.694	26543.15	13591.749
Delay (pico sec)	10421	3630	9008	6673
Area (No. of Cells)	1617	916	1323	779

5.4 Comparison of 64 bit Adders

5.5 Synthesis of 64 Bit Error Tolerent Adder



5.6. Advantages, Dis-Advantages And Application of Eta Adder

5.6.1. Advantages

- Since the Carry bit from LSB to MSB is eliminated the overall circuit power Consumption is reduced.
- While propagating the carry signal glitches are generated, since the carry bit is
- Eliminated the glitches are also eliminated.
- Since the adder is built using the error tolerant, the MSB bits need not wait for the Carry bit generated from LSB, due to which the system speed is enhanced.
- As the current world is working on low power devises the power consumed by the
- ETA adder is significantly less when compared to normal conventional adder.

5.7. Dis-Advantages (Limitations)

- The system has to compromise with Accuracy.
- Cannot be used in the system where accuracy is required.

5.8. Applications

- It is used in Laptops, Cell-phones.
- It is used in DSP applications such as image processing and voice processing.
- It is basically used in system where power and speed is important than accuracy.

6. Conclusion

The proposed Error Tolerant Adder trades a certain amount of accuracy for significant power saving and performance improvement. Extensive comparisons with conventional Adders i.e. Ripple Carry Adder is shown in the table.2 indicate that the proposed ETA out-performed the conventional Adders Applications Power Performance.

7. References

1. "Design of low- power high-speed error Tolerant shift and add multiplier" Journal of computer science 7 (12): 1839-1845, 20111ssn 1549- 3636 © 2011 science publications Corresponding author: 1 k.n. vijeyakumar.
2. "Design of low-power high-speed truncationerror- tolerant adder and its application in digital signal processing" ning zhu, wang ling goh, weija zhang, kiat seng yeo, and zhi hui kong ieee transactions on very large scale integration (vlsi) systems, vol. 18, no. 8, august 2010.
3. "Design and error-tolerance in the presence of massive numbers of defects," m.A. Breuer, s. K. Gupta, and t. M Mak, ieee des. Test Comput., vol. 24, no. 3, pp. 216-227, may-jun. 2004.
4. "A novel[4] L. Sterpone, M. SonzaReorda and M. Violante, "Evaluating Different Solutions toDesign Fault Tolerant Sytems with SRAM-based FPGAs," Journal of ElectronicTesting: Theory and Applications, vol. 23, pp. 47-54, 2007.
5. K. Kyriakoulakos and D. Pnevmatikatos, "A Novel SRAM-Based FPGA Architecture for Efficient TMR Fault Tolerance Support," International Conference on Field Programmable Logic and Applications, pp. 193-198, 2009.
6. 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
7. 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

8. 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
9. 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.
10. 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
11. 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