International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Novel Approach to 32-Bit Approximate Adder

Shalini Singh[1], Ghanshyam Jangid[2]

[1]Department of Electronics and Communication, Gyan Vihar University, Jaipur, Rajasthan, India

[2]Assistant Professor, Department of Electronics and Communication, Gyan Vihar University, Jaipur, Rajasthan, India

Abstract: ***The probability of errors in the present VLSI technology is very high and it is increasing with technology scaling. Removing all errors is not required for certain applications and it is also a very expensive task. There are certain applications where the approximate result is acceptable e.g. image processing and video processing. For these applications Approximate Adder is proposed which provide approximate result at very high speed than the conventional adder. This error-tolerant adder is easy to develop the accuracy output and simultaneously achieves excellent improvement in both the area and speed performance. By comparing previous conventional counter parts and approximate adder the proposed approximate adder is to be attaining more than 20% improvements in speed. One important potential application of the proposed adder in digital signal processing systems is that can tolerate certain amount of errors. The proposed adder provides improvement in delay and area at the same time at the cost of accuracy. Simulation result shows improvement in speed and area respectively over conventional adder.***

Keywords: Adders, digital signal processing (DSP), approximate adder, high-speed integrated circuits, VLSI

1. 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 [1]. A large and growing number of applications are inherently error-tolerant, which do not require “strict” correctness but rather approximate correctness. Applications of such kind include multimedia, DSP, wireless communication, data mining and synthesis. They may process noisy data sets and the associated algorithms are stochastic or involve a human interface with limited perceptual capability. Based on the characteristic of digital VLSI design, some novel concepts and design techniques have been proposed. 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]. 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 process sing and speech processing systems, the error-tolerant circuits may be applicable [3], [6], [7]. Approximate adder provides high speed by cutting down the carry propagation. It can be used where high speed is required and small error may be tolerated. So, approximate adder is also known as application specific adder.

In this paper, main aim is to design an adder that enhances the speed and performance. The remainder of this paper is organised as follows. In section II, a brief review of approximate adder is discussed. In section III, previous work is discussed. Next, section IV presents the proposed work. Finally, section V concludes this paper.

2. Approximate 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) [8], carry-select adder (CSL) [9], and carry-look-ahead adder (CLA) [10], have been developed. By sacrificing some accuracy, the approximate adder can attain great improvement in both the area and speed performance. There are some well defined parameters by which approximate designs can be evaluated.

- Overall error (OE): OE=|Rc-Ra |, where Ra is the result obtained by the approximate 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 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%.
- Minimum Acceptable Accuracy (MAA): It is specified according to the application and the accuracy of an acceptable output should be high enough. The result whose accuracy is higher than the minimum acceptable accuracy is known as acceptable result.
- Acceptable Probability(AP) : If 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.

2.1 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, a great improvement in speed performance can be achieved. This new addition arithmetic can be illustrated via an example shown below.

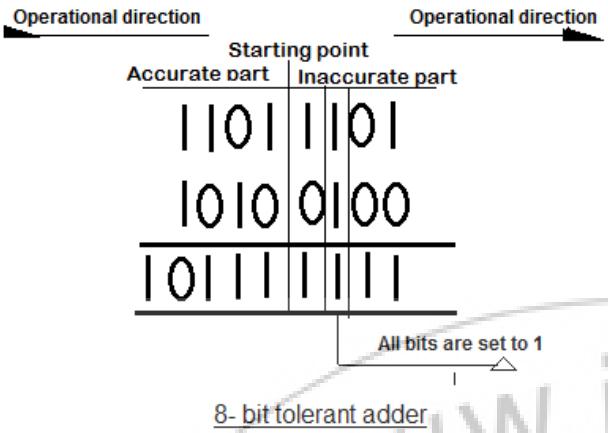

Figure 1: 8-bit approximate adder example

Here, 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 two 8-bit input operands, A= "11011101" (221) and B= "10100100" (164), 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. A special strategy is adapted to eliminate the carry chain which minimizes the overall error. It 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. 1 with a final result of "11011101" (221) which should actually yield "10100100" (164) if normal arithmetic has been applied. The overall error generated can be computed as OE=385-383=2. The accuracy of the adder with respect to these two input operands is ACC=(1- (2/385))×100=99.48%. This accuracy level is acceptable for most of the image processing applications. Therefore by eliminating carry in the inaccurate part performing addition in two separate parts simultaneously, the overall delay time and area is greatly reduced.

3. Previous Work

The first step of designing a proposed adder 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. 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.

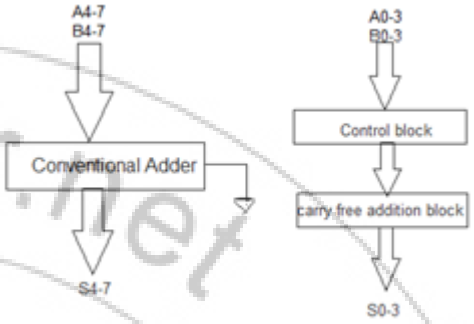

Figure 2: Block diagram of approximate adder

A. Design of the Accurate Part:

In our proposed 32-bit approximate adder, 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 approximate adder as it determines the speed, performance and area 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. 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 paper, we propose for the first time, an innovative and novel addition arithmetic that can attain great saving in speed and power consumption. 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. 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.

4. Proposed Approximate Adder

We have seen in previous approximate adder that if both the inputs bits were high then the all input bits were set high. The traversing of generated control signal from a[19] to a[0] had propagation delay which increased the size of adder and reduced its speed. It can be understood by the following equation.

sum[i] = ctrli | (a[i]^b[i]……. (1)

In this equation, control signal is generated and when it is 1the approximate part automatically becomes 1. In order to overcome these limitations we proposed two new architectures of approximate adder that is more efficient in terms of speed and area. Block diagram of proposed approximate adder is shown in fig. 3. The proposed 32 bit approximate adder is designed in XILINX 9.2 using VERILOG HDL code.

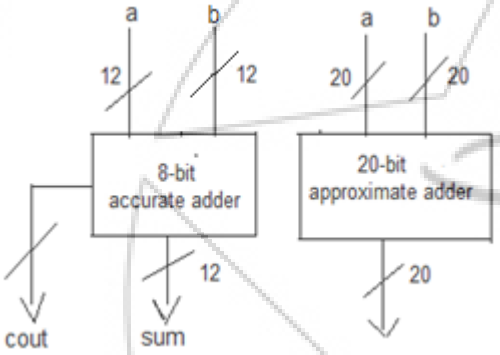

Figure 3: Block diagram of proposed adder

In proposed work 1, the 20 bit approximate adder is divided into 5 equal parts. Previously, inputs were not divided but in new logic each part is divided into 4 bits. This dividing strategy greatly reduced the delay. The basic architecture of proposed approximate adder is shown in fig.4

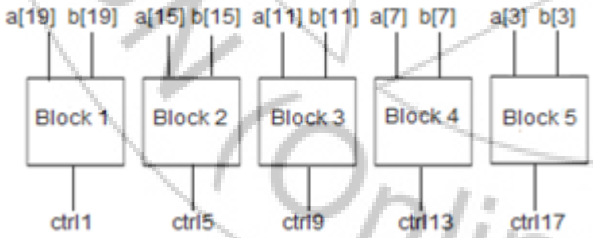

Figure 4: Basic architecture of proposed approximate adder 1

From the architecture, it is clear that control signal propagates from a[19]- a[16], a[15]- a[12].....a[3]- a[0]. By dividing into blocks, delay is reduced to a great extent. The value of control signal becomes 1 whenever the two inputs are logic high i.e '1', output generated will be high and following that all bits will be high. It can be better understood by the control logic of type 1 as shown in fig.5.

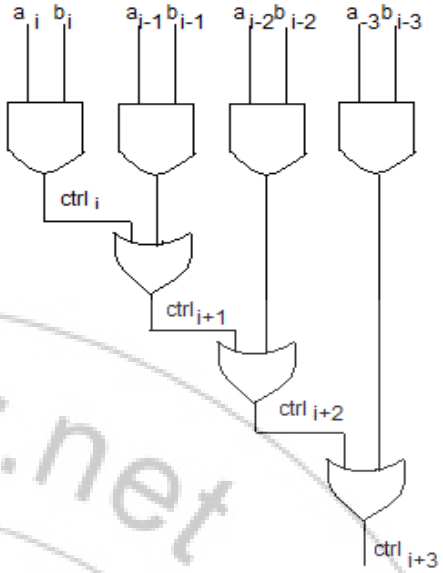

Figure 5: Control logic of proposed adder type 1

In the proposed work 1, the value of control signal became 1 whenever the two inputs were logic high i.e "1" but in the proposed work 2, control signal will become 1 if either of the inputs are 01, 10 or 11and further sum become automatically 1 when any control signal is 1as shown in table 2. This logic also improves the speed and area of the approximate adder. In fig. 6 control logic of type 2 is shown that helps to understand in better way.

Table 2

Logic input	Logic output
00	0
01	1
10	1
11	1

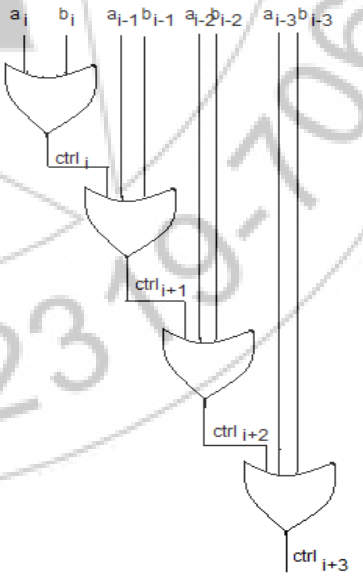

Figure 6: Control logic of approximate adder type 2

5. Result and Comparison

In this paper, performance of base paper and proposed approximate adder is evaluated and is implemented. We

synthesise this approximate adder using Xilinx ISE 9.2 and use Verilog as hardware description language. From the table1, comparison can be done between previous and proposed approximate adder.

Table 1: Comparison between previous and approximate adder

	Previous approximate adder	Proposed approximate adder 1	Proposed approximate adder 2
No. of slices	47	43	38
Delay (ns)	7.906	6.120	6.120

We can see that the speed of proposed work is improved by 22.5% than the conventional adder and area of proposed work 1 is also reduced by 8.5% and in proposed work 2, area is reduced by 19%. Comparing the simulation results of our proposed approximate adder with those of conventional adders, it is proved that the approximate adder performed best in terms of speed and area. The schematic of proposed approximate adder 1 is shown in fig.7 and the simulation waveform is also shown in fig.8.

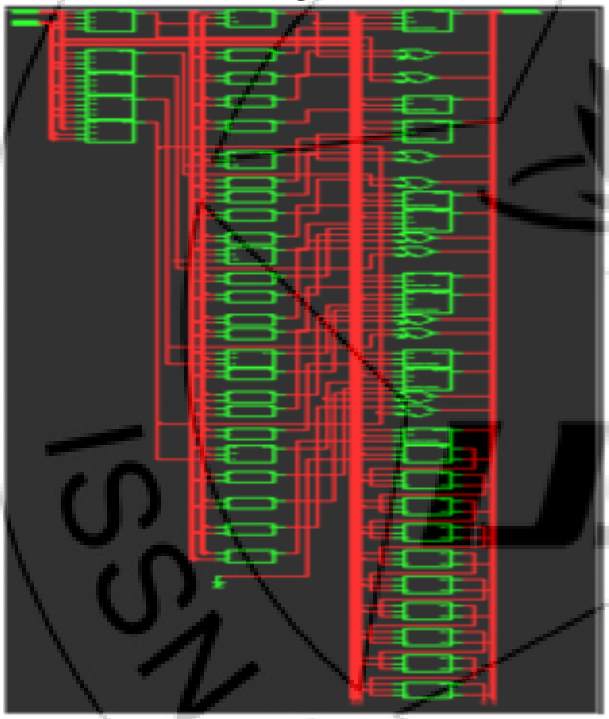

Figure 7: Proposed 32- bit approximate adder type 1

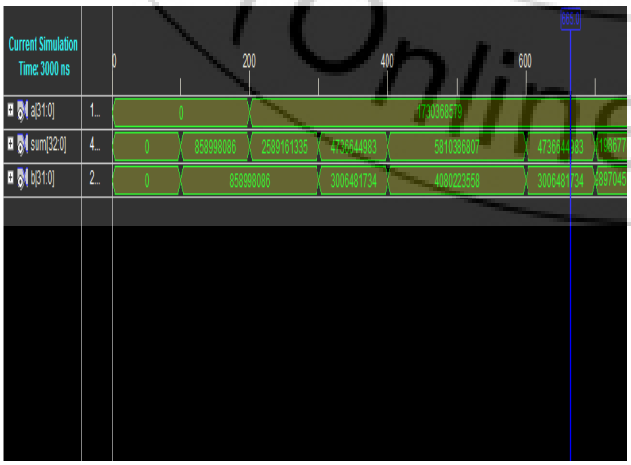

Figure 8: Simulation result 1

The schematic of proposed approximate adder 2 is shown in fig.9 and the simulation waveform is also shown in fig.10.

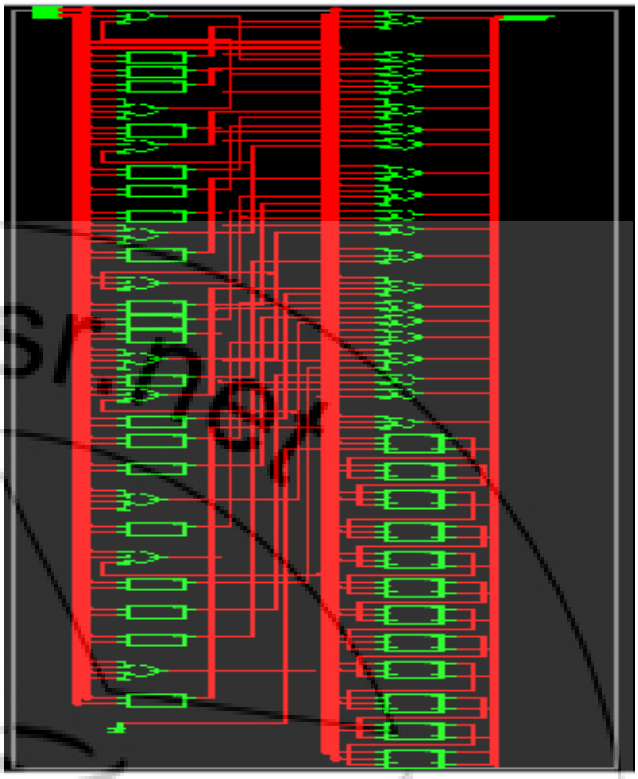

Figure 9: Proposed 32- bit approximate adder 2

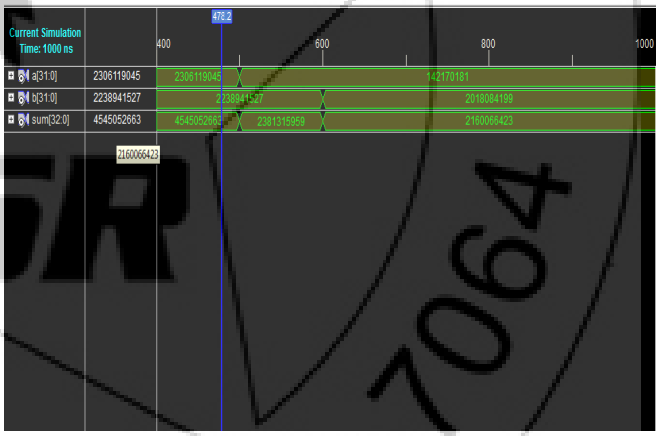

Figure 10: Simulation result 2

6. Conclusion and Future Scope

In this paper, the conventional and modified approximate adders are designed using Verilog. The speed of modified adder is more as compared to conventional adder. We use ripple carry adder in the approximate part of adder. It increases the speed of adder by eliminating the carry. The proposed Approximate Adder trades a certain amount of accuracy and provides improved speed and area. In future, this algorithm can be used in multiplier and BCD adder which may further improve its speed and also its area may get reduced. Ripple carry adder increases speed significantly but other adders also improves the speed. In future adders like Carry select adder and Carry look ahead adder can be used to lower the delay thus enhancing the speed and performance.

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 highspeed carry propagation in binary arithmetic units," *IRETrans. Electron. Comput*, vol. EC-10, pp. 691–698, Dec. 1962.
[8] V. K. Chippa, D. Mohapatra, A. Raghunathan, K. Roy, S. T. Chakradhar, Scalable effort hardware design: exploiting algorithmic resilience for energy efficiency. In *Proc. IEEE/ACM Design Automation Conference (DAC)*, pp.555-560, 2010.
[9] B. Kahng, S. Kang, Accuracy-configurable adder for approximate arithmetic designs. In *Proc. IEEE/ACM Design Automation Conference (DAC)*, pp.820-825, 2012.
[10] S. Yu; E. E. Swartziander Jr., DCT implementation with distributed arithmetic. In *IEEE Transactions on Computers*, vol.50, no.9, pp.985-991, Sep 2001.
[11] M. Narasimha Rao, P. Ganesh Kumar, B. Ratna Raju " Design of high speed 32 bit truncation-error-tolerant adder in international journal of advanced research in electronics and communication engineering(IJARECE), Vol. 1, Issue 5, November 2012

Author Profile

Shalini Singh, final year student of in M. Tech (dual), EC+VlSI studying in Suresh Gyan Vihar University.

Ghanshyam Jangid has completed his B.tech from Global Institute of Technology in 2008 and completed his M. Tech from MNIT, Jaipur in 2013. Currently he is working in Suresh Gyan Vihar University as Assistant Professor.