

An Enhanced Carry Elimination Adder for Low Power VLSI Applications

V.Muralidharan

P.G. Scholar - M.E.VLSI Design
Sri Ramakrishna Engineering College
Coimbatore, India

Dr.M.Jagadeeswari

Professor and Head - M.E. VLSI Design
Sri Ramakrishna Engineering College
Coimbatore, India

Abstract— Truncation and round off errors in adders has become unavoidable in modern VLSI technology. A new type of adder i.e. error tolerant adder(ETA) is proposed to tolerate those errors and to attain low power consumption and high speed performance in DSP systems. In conventional adder circuit, delay is mainly certified to the carry propagation chain along the critical path, from the LSB to MSB. If the carry propagation can be eliminated by the technique proposed in this paper, a great improvement in speed performance and power consumption is achieved. By operating shifting and addition in parallel, the error tolerant adder tree compensates for the truncation errors. To prove the feasibility of the ETA, normal addition operation present in the DCT algorithm is replaced by the proposed addition arithmetic and the experimental results are shown.

Index Terms— Adders, digital signal processing (DSP), error tolerance, discrete cosine transform, distributed arithmetic, and XOR gate.

1. INTRODUCTION

The system should always provide correct and accurate results in earlier digital VLSI design. The analog computation produces “good enough” results rather than accurate results [9]. The data processed by many digital systems may already contain errors. The analog signal is 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.

To deal with error-tolerant problems, some truncated adders/multipliers have been reported, [8] [10] but are not able to perform well in its speed, power, area, or accuracy. 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 processes and speech processing systems, the error-tolerant circuits may be applicable [11].

The rest of the paper is organized as follows. Section II proposes the addition arithmetic of the error-tolerant adder

(ETA). In Section III, the detailed design of the ETA is explained. The experimental results are shown in Section IV. Section V provides an application example of the ETA. Lastly, the conclusion of this work is presented in Section VI.

2. LITERATURE SURVEY

Most of the adder structures discussed in this paper is applicable to general purpose designs, with a few exceptions. Increasingly huge data sets and the need for instant response, the adder should 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 adders, such as the carry-skip adder (CSK) [13], The Carry Skip Adder reduce the carry-propagation time by skipping over groups of consecutive adder stages. The carry skip adder is usually comparable in speed to the carry lookahead adders, but it requires less chip area and consumes less power compared to carry lookahead adders.

Then next the carry-select adder (CSL) [1], The Carry Select Adder consists of two ripple carry adders and one multiplexer. The two adders are used to calculate the addition twice; one addition is computed assuming carry input “1” and the other as “0”. The correct output is then selected upon the arrival of carry-in and carry-look-ahead adder (CLA), [14] have been developed. The Carry Look Ahead Adder has lower delay but requires much more complex circuitry in achieving its performance. 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.

3. ERROR TOLERANT ADDER

Error-tolerance deals with the use of defective circuitry that occasionally produces errors, yet provides acceptable performance to end users when executing certain applications [2]. The motivation for using such devices is the related increase in effective yield, and hence lower cost parts. This error tolerance concept is applicable only for the system which doesn't require accurate output results. 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.

3.1 Projected Addition Arithmetic

In a normal adder circuit, the delay is mainly certified to the carry propagation chain along the critical path, from the LSB to MSB. A major proportion of the power consumption of an adder is due to the unwanted signal component that are caused by the carry propagation[7]. Therefore, if the carry propagation can be removed, a great improvement in speed performance and power consumption can be achieved.

Some terminologies used in the ETA are shown below.

- **Total error (TE):** $TE = |Bc - Be|$, where Be is the result obtained by the adder, and Bc denotes the correct result (all the results are represented as decimal numbers)
- **Accuracy (ACC):** 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: $ACC = (1 - TE/Bc) * 100$. 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.

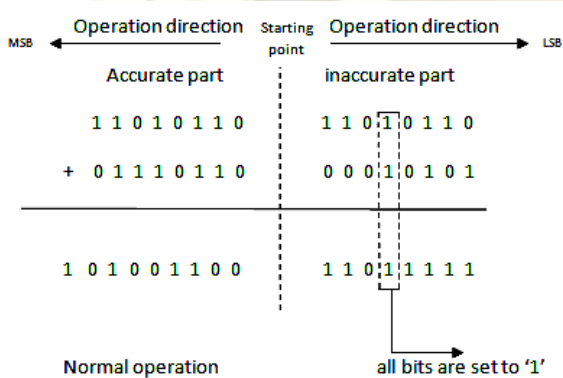


Fig. 1 Proposed Addition Arithmetic

The input operands are split into two parts: an accurate part that includes most significant bits and the inaccurate part that is made up of the least significant bits. The length of each part need not necessary be equal. The two 16-bit input operands, A = "1101011011010110" (54998) and B = "0111011000010101" (30229), are divided equally into 8 bits each for the accurate and inaccurate parts are shown in Fig 1.

In accurate part normal addition method is applied from right to left (LSB to MSB). This is to preserve its correctness since the higher order bits play a more important role than the lower order bits. No carry signal will be generated at any

bit position to eliminate the carry propagation path. To minimize the overall error, 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." The addition mechanism described can be easily understood from the example given with a final result of "1010011001101111" (85215).

The above addition should actually yield "10100110011101011" (85227) if normal arithmetic has been applied. The total error generated can be computed as $TE = 85227 - 85215 = 12$. The accuracy of the adder with respect to these two input operands is $ACC = (1 - 12/85227) * 100\% = 99.98\%$. By eliminating the carry propagation path in the inaccurate part and performing the addition in two separate parts simultaneously, the overall delay time is greatly reduced, so is the power consumption.

4. PROPOSED METHOD

The error tolerant adder consists of two parts: an accurate part and an inaccurate part. Here a 32-bit adder is used as an example for our illustration of the design methodology and circuit implementation of an ETA.

4.1 Dividing Approach of the Adder

The first step is to divide the ETA into two parts in a specific manner. The dividing approach is based on a guess-and-verify trick, depending on the requirements, such as accuracy, speed, and power. First consider the delay of the proposed ETA as $E_d = \max(E_h, E_l)$ where E_h is the delay in the accurate part and E_l is the delay in the inaccurate part. With the proper dividing strategy, we can make E_h approximately equal to E_l and hence achieve an optimal time delay. With this dividing strategy method defined, then test whether the accuracy performance of the adder meets the requirements specified by the designer.

For some application, the requirement of the minimum acceptable accuracy should 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.

Due to the simplicity in structure and the removal of switching actions in the inaccurate part, putting more bits in this part yields more power saving. [7] Here the 32bit adder is divided by putting 12 bits in the accurate part and 20 bits in the inaccurate part.

4.2 Design of the Accurate Part of 32bit ETA

In the proposed 32-bit ETA, the accurate part consists of 12 bits. The ripple-carry adder has been chosen for the accurate part of the circuit since it is the most power-saving conventional adder, The Ripple Carry Adder, being the simplest one, uses the least hardware circuitry when

compared to all other traditional adder circuits in use is shown in Fig 2.

The delay of the ripple carry adder increases linearly with the number of bits with a worst case delay of $O(n)$. This worst case delay makes it slow when large bit sizes are used.

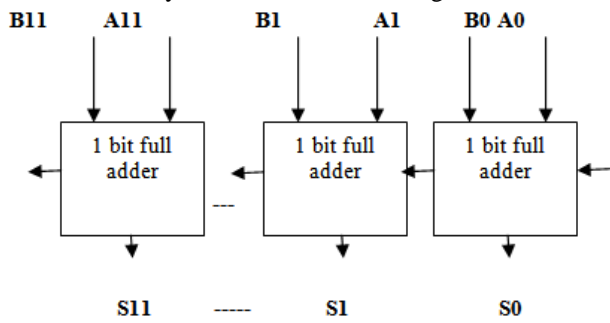


Fig. 2 12-bit ripple carry adder

4.3 Design of the Inaccurate Part of 32bit ETA

The inaccurate part consist of 20 bits and it is the most essential 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 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.

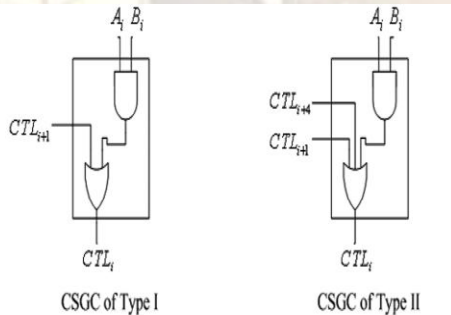


Fig. 3 Graphical Implementations of CSGC

Two types of CSGC, labelled as type I and II are designed, and the schematic implementations of these two types of CSGC are provided.

4.4 Carry free addition 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 shown in Fig 4.

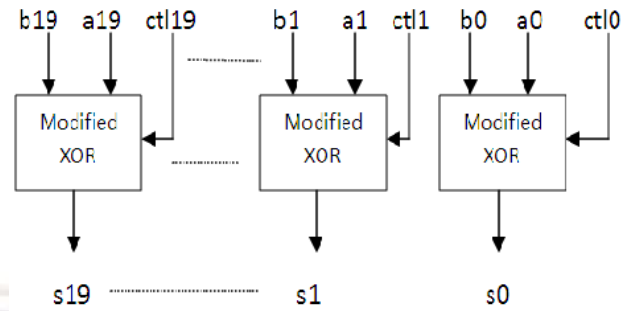


Fig. 4 Carry-free addition block: Overall architecture

Here the modified XOR gate is designed by using the CMOS logic. It consists of three inputs namely A, B and CTL. CTL is the control signal coming from the control block and is used to set the operational mode of the circuit.

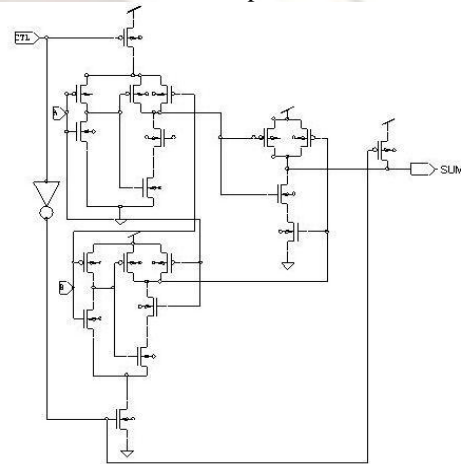
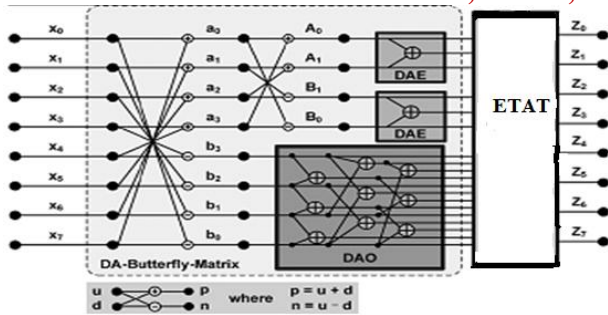


Fig. 5 XOR gate using CMOS logic

When $CTL=0$, leaving the circuit to operate in the normal XOR mode. When $CTL=1$, connecting the output node to VDD, and hence setting the sum output to "1." Hence the novel adder is designed. To prove the feasibility of the ETA, normal addition operation present in the DCT algorithm can be replaced by the proposed addition arithmetic.

5. ARCHITECTURE OF THE 1-D 8-POINT DCT

The 1-D 8-point DCT architecture can be constructed using a DA-Butterfly-Matrix that includes two DA even processing elements (DAEs), a DA odd processing element (DAO) and 12 adders/subtractions, and 8 ETAs.



Architecture of the 1-D 8-Point DCT

5.1 Discrete Cosine Transform

Discrete cosine transform is a widely used tool in image processing and video compression applications [15]. One clear advantage of the DCT over the DFT is that there is no need to manipulate complex numbers. Recently, the high throughput DCT designs have been adopted to fit the requirements of real time applications. When computing a two-dimensional DCT, a large number of multiplications and additions are required in the direct approach. Multiplications, which are the most time-consuming operations in simple processor, can be completely avoided in the proposed architecture for real-time image compression. An area efficient high performance VLSI architecture for DCT based on the distributed arithmetic is proposed in this paper. Minimum number of additions is used to the DCT by exploiting the timing property of the DCT transform based on the distributed arithmetic [16].

5.2 Distributed Arithmetic

Distributed Arithmetic (DA) is an efficient method for computing inner products when one of the input vectors is fixed [19]. It uses look-up tables and accumulators instead of multipliers for computing inner products and has been widely used in many DSP applications such as DFT, DCT, convolution, and digital filters. In particular, there has been great interest in implementing DCT with distributed arithmetic and in reducing the ROM size required in the implementations since the DA-based DCT architectures are known to have very regular structures suitable for VLSI implementations [17]. Most DA-based DCT implementations use the original DCT algorithm or the even-odd frequency decomposition of the DCT algorithm along with some memory reduction techniques such as the partial sum technique and/or the offset binary coding technique [18].

5.3 Error Tolerant Adder Tree (ETAT)

In general, the shifting and addition computation uses a shift-and-add operator in VLSI implementation in order to reduce hardware cost. However, when the number of the shifting and addition words increases, the computation time will also increase. Therefore, the shift-adder-tree operates shifting and addition in parallel by unrolling all the words needed to be computed for high-speed applications. However, a large truncation error occurs. The proposed ETA operates shifting and addition in parallel by unrolling all the

words required to be computed. Furthermore, the error-compensated circuit alleviates the truncation error for high accuracy design. Here the conventional adder is said to be the ripple carry adder (RCA). Here the normal addition operation can be replaced by the proposed addition arithmetic to achieve low power consumption and high speed performance.

Let A = 4 bit right shift register which will give the LSB of the resultant at the 4th clock cycle. B = 4 bit register. Value will never get change. P = 4 bit shift register partial input partial output (PIPO) for storing the partial product. At first T=0; P = 0000; A= 1001; B = 1101. At the first clock cycle i.e., at T = 1. It check the LSB bit of the A register if it if „1“ then B register value will get added with the initial P register value else if LSB bit of A register value is „0“ then initial p value will be get added with the current p register value. After adding we will get a 5 bit output (4:0), in that bit (4:1) is stored in the P register and the (0) LSB value will be shifted in to the A register from right to left. Multiplied result will be obtained at the Nth cycle of the process, where N represents the number of bits in the A and B. here 4 bit of data is used and at 4th clock cycle the output is obtained.

6. EXPERIMENTAL RESULTS AND COMPARISON

To demonstrate the advantages of the proposed ETA, simulate the ETA along with two types of conventional adders namely Ripple Carry Adder and Carry Select Adder by using XILINX ISE10.1 tool.

TABLE I
POWER AND AREA RESULTS

Type	Power(mw)	Delay(ns)	Transistor count
RCA	40	15.35	106
CSLA	41	17.00	150
ETA	33	7.269	120

Table I shows the comparison results of ETA and the conventional adders. Comparing the simulation results of the proposed ETA with other adders shows that the overall performance of the ETA is improved. The power consumed by the ETA (33mw) is less when compared to other two methods. The delay of ETA (7.269ns) is also reduced than others. The transistor is slightly increased due to the design of XOR gate in CMOS logic. To reduce the transistor count transmission gates can be used instead of CMOS logic in XOR gate, the area will be further reduced.

Comparison Results of Adders

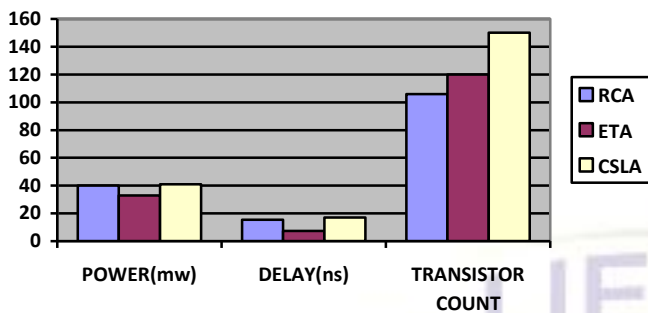


Fig.7 Image processed with the proposed ETA

The comparison between the two images shows that the quality loss to the image using the proposed ETA is negligible and can be completely tolerated by human eyes. These simulation results have proven the practicability of the ETA proposed.

7. APPLICATION OF ERROR TOLERANT ADDER IN DSP

The Discrete Cosine Transformation is a critical function in image processing and many other DSP applications. The analysis process of DCT contains large number of additions and multiplications. To prove the feasibility of the ETA, we replaced all the common additions involved in a normal DCT algorithm with our proposed addition arithmetic.

A digital image was represented by a matrix in a DSP system, and each element of the matrix represents the colour of one pixel of the image. To compare the quality of images processed by both the conventional DCT and the inaccurate DCT that had incorporated our proposed ETA, consider the following experiment, an image was first translated to a matrix form and sent through a standard system that made use of normal DCT and normal reverse DCT.



Fig. 6 Image processed with conventional adder

The matrix output of this system was then transformed back to an image and presented in Fig 6.

The matrix of the same image was also processed in a system that used the inaccurate DCT and inaccurate reverse DCT, where both DCT's had incorporated the 32-bit ETA described in Section III, with the processed image given in Fig 7. Although the two resultant matrices of the same image were different, the two pictures obtained are almost same Fig 7 contains horizontal bands of different shades of gray.

8. CONCLUSION AND FUTURE WORK

In this paper, the concept of error tolerance is introduced in VLSI design. A novel type of adder, (error-tolerant adder), which trades certain amount of accuracy for significant power saving and performance improvement, is proposed. By eliminating the carry propagation path in the inaccurate part and performing the addition in two separate parts simultaneously, the overall delay time is greatly reduced. Then the ETA is applied to the 1-D 8point DCT. The normal addition operation is replaced by the proposed addition arithmetic. The proposed ETA is well in terms of power and area when compared to the conventional ripple carry adder (RCA). In future modification can be made by replacing the XOR gate using CMOS logic by Transmission gates where the number of transistors can be reduced. Therefore transistor gate counts will be reduce so the area is also been reduced.

ACKNOWLEDGEMENT

The authors thank the Management and Principal of Sri Ramakrishna Engineering College, Coimbatore for providing excellent computing facility and encouragement.

REFERENCES

- [1] Bedrij, O. (1962) 'Carry Select Adder', IRE Trans.Electron.Comput. Vol. EC-11, pp.340-346.
- [2] Breuer, M. A. Gupta, S. K. and Mak, T. M. (2004) 'Defect and Error Tolerance in the Presence of Massive Numbers of Defects', IEEE Design and Test Magazine.
- [3] Cheemalavagu, S. Korkmaz, P. and Palem, K.V. Sep (2004) 'Ultra Low Energy Computing via Probabilistic Algorithms and Devices: CMOS device primitives and the energy Probability relationship', in Proc. 2004 Int. Conf. Solid State Devices and Materials, Tokyo, Japan, pp. 402-403.

- [4] Ning Zhu, Wang Ling Goh, and Kiat Seng Yeo (2009) 'An Enhanced Low-Power High-Speed Adder for Error-Tolerant Application', School of Electrical and Electronic Engineering Nanyang Technological University Singapore
- [5] Ning Zhu, Wang Ling Goh, Weija Zhang, (2010) 'Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing'
- [6] International Technology Roadmap for Semiconductors [Online] Available.
- [7] Kiat-Seng and Kaushik, R. (2005) Low-Voltage, Low-Power VLSI Subsystems. New York: McGraw-Hill
- [8] Stine, J.E. Babb, C.R. and Dave, V.B. (2003) 'Constant Addition Utilizing Flagged Prefix Structures', in Proc. IEEE Int. Symp. Circuits and Systems.
- [9] Melvin, A. B. (2005) 'Let's Think Analog', in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, pp. 2–5.
- [10] 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.
- [11] A. B. Melvin and Z. Haiyang, "Error-tolerance and multi-media, "in Proc. 2006 Int. Conf. Intel. Inf. Hiding and Multimedia Signal Process, 2006, pp. 521–524.
- [12] 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.
- [13] O. MacSorley, "High speed arithmetic in binary computers," *IRE Proc.*, vol. 49, pp. 67–91, 1961.
- [14] N. I. Cho and S.U. Lee, "Fast Algorithm and Implementation of 2-D DCT," *IEEE Transactions On Circuits and Systems*, vol. 38 p. 297, March 1991.
- [15] G. Strang, "The Discrete Cosine Transform," *SIAM Review*, Volume 41, Number 1, pp.135-147, 1999.
- [16] S. Yu and E. E. S. , Jr., "DCT implementation with distributed arithmetic," *IEEE Trans. Comput.*, vol. 50, no. 9, pp. 985–991, Sep. 2001.
- [17] A. M. Shams, A. Chidanandan, W. Pan, and M. A. Bayoumi, "NEDA: A low-power high-performance DCT architecture," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 955–964, Mar. 2006.
- [18] C. Peng, X. Cao, D. Yu, and X. Zhang, "A 250 MHz optimized distributed architecture of 2D 8 \times 8 DCT," in *Proc. Int. Conf. ASIC*, 2007, pp. 189–192.