Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing

Ning Zhu, Wang Ling Goh, Weija Zhang, Kiat Seng Yeo, and Zhi Hui Kong

***Abstract*—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 more than 65% improvement in the Power-Delay Product (PDP). One important potential application of the proposed ETA is in digital signal processing systems that can tolerate certain amount of errors.**

***Index Terms*—Adders, digital signal processing (DSP), error tolerance, high-speed integrated circuits, low-power design, VLSI.**

I. 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 [1]. 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 [2].

Based on the characteristic of digital VLSI design, some novel concepts and design techniques have been proposed. The concept of error tolerance (ET) [3]–[10] and the PCMOS technology [11]–[13] 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 [3]. The "imperfect" attribute seems to be not appealing. However, the need for the error-tolerant circuit [3]–[10] was foretold in the 2003 International Technology Roadmap for Semiconductors (ITRS) [2].

To deal with error-tolerant problems, some truncated adders/multipliers have been reported [14], [15] but are not able to perform well in either its speed, power, area, or accuracy. The "flagged prefixed adder" [14] 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" [15], 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. 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], [6], [7].

The rest of the paper is organized as follows. Section II proposes the addition arithmetic as well as the structure 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.

II. 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):* $OE = |R_c - R_e|$, 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: $ACC = (1 - (OE/R_c)) \times 100\%$. 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.

A. 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 fast adders, such as the carry-skip adder (CSK) [16], carry-select adder (CSL) [17], and carry-look-ahead adder (CLA) [18], have been developed. Also, there are many low-power adder design techniques that have been proposed [19]. 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.

B. 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 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. 1.

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

Manuscript received May 28, 2008; revised February 12, 2009; accepted March 16, 2009. First published October 13, 2009; current version published July 23, 2010.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 639798 (e-mail: zhuning@ntu.edu.sg).

Digital Object Identifier 10.1109/TVLSI.2009.2020591

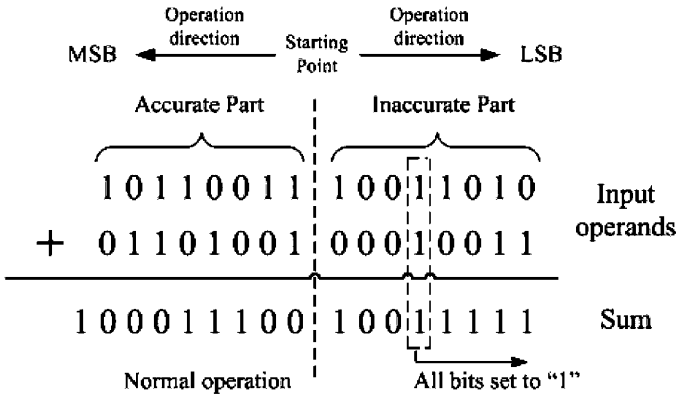

Fig. 1. Proposed addition arithmetic.

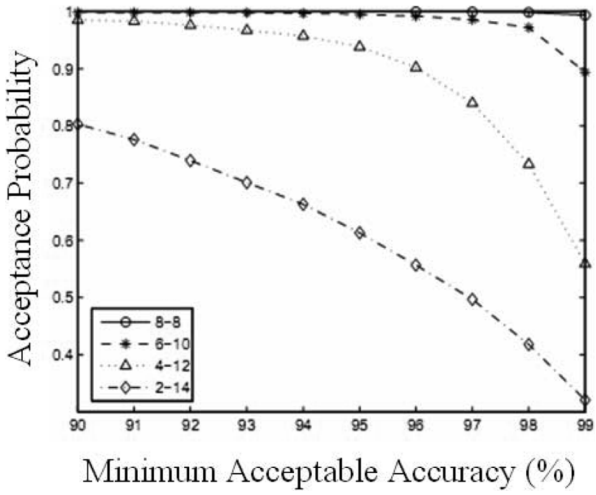

Fig. 2. Relationship between AP and MAA.

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, A = "1011001110011010" (45978) and B = "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." The addition mechanism described can be easily understood from the example given in Fig. 1 with a final result of "10001110010011111" (72863).

The example given in Fig. 1 should actually yield "10001110010101101" (72877) if normal arithmetic has been applied. The overall error generated can be computed as $OE = 72877 - 72863 = 14$. The accuracy of the adder with respect to these two input operands is $ACC = (1 - (14/72877)) \times 100\% = 99.98\%$.

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.

C. Relationships Between Minimum Acceptable Accuracy, Acceptance Probability, Dividing Strategy, and Size of Adder

The accuracy of the adder is closely related to the input pattern. Assume that the input of an adder is random; there exists a probability that we can obtain an acceptable result (i.e., the acceptance probability). The accuracy attribute of an ETA is determined by the dividing strategy and size of adder. In this subsection, the relationships between the minimum acceptable accuracy, the acceptance probability, the dividing strategy, and the size of adder are investigated.

We first consider the extreme situation where we accept only the perfectly correct result. The minimum acceptable accuracy in this "perfect" situation is 100%. According to the proposed addition arithmetic, we can obtain correct results only when the two input bits on every position in the inaccurate part are not equal to "1" at the same time. We can therefore derive an equation to calculate the acceptance probability associated with the proposed ETA with different bit sizes and dividing strategies. This equation is given as follows:

$$P(ACC = 100\%) = \frac{4^{N_t - N_l} \times 3^{N_t} + 2^{N_t - N_l}}{4^{N_t} + 2^{N_t}} \quad (1)$$

where N_t is the total number of bits in the input operand (also regarded as the size of the adder) and N_l is the number of bits in the inaccurate part (which is indicating the dividing strategy).

In situations where the requirement on accuracy can be somewhat relaxed are investigated, the result will be different. C program is engaged to simulate a 16-bit adder that had adopted the proposed addition mechanism. By checking the output results, we can derive the relationship between the minimum acceptable accuracy and acceptance probability, as depicted in Fig. 2. The four curves represent four different dividing strategies, and each of which has been assigned a name "$N-M$" where "N" denotes the size of the accurate part and "M" for the size of the inaccurate part. For the input patterns, we randomly select 10 000 inputs from all possible input patterns (i.e., 0–65 535). It can be deduced from Fig. 2 that the lower the minimum acceptable accuracy set, the higher the acceptance probability for the adder. Fig. 2 also shows that different dividing strategies lead to different accuracy performance.

As modern VLSI technology advances, the size of the adder has to increase to cater to the application need. The trend of the accuracy performance of an ETA is therefore investigated in Fig. 3. The five curves are associated with different minimum acceptable accuracies, 95%, 96%, 97%, 98%, and 99%, respectively. Note that all adders follow the same dividing strategy whereby the inaccurate part is three times larger than that of the accurate part. Since small numbers will be calculated at the inaccurate part of the adder, the proposed ETA is best suited for large input patterns.

D. 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. In Section III, a 32-bit adder is

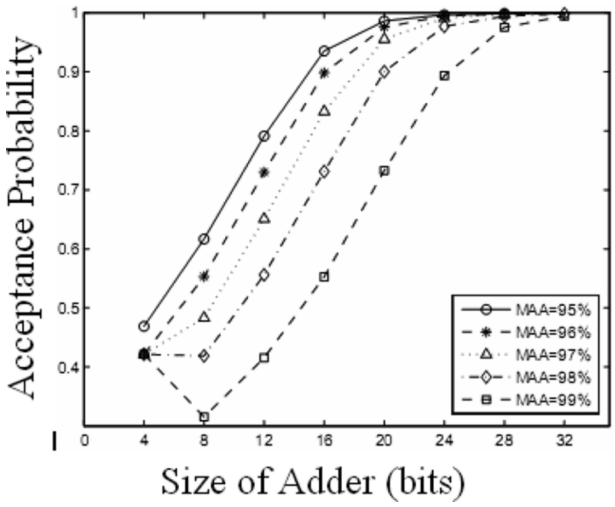

Fig. 3. Relationship between AP and size of adder.

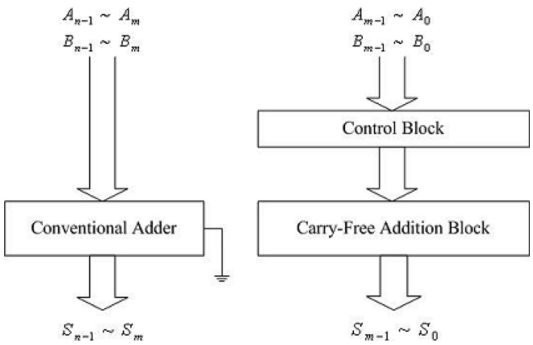

Fig. 4. Hardware implementation of the proposed ETA.

used as an example for our illustration of the design methodology and circuit implementation of an ETA.

III. Design of a 32-Bit Error-Tolerant Adder

A. 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.

First, we define the delay of the proposed adder as $T_d = \max(T_h, T_l)$, 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. For example, 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.

Having considered the above, we divided the 32-bit adder by putting 12 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 Fig. 5. 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. 6 and is used to set the operational mode of the circuit. When $\mathrm{CTL} = 0$, M1 and M2 are turned on, while M3 is turned off, leaving the circuit to operate in the normal XOR mode. When $\mathrm{CTL} = 1$, 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. 6(a), are designed, and the schematic implementations of these two types of CSGC are provided in Fig. 6(b). 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. 6(a)] consists of only ten cells.

IV. Experimental Results

To demonstrate the advantages of the proposed ETA, we simulated the ETA along with four types of conventional adders, i.e., the RCA, CSK, CSL, and CLA, using HSPICE. All the circuits were implemented using Chartered Semiconductor Manufacturing Ltd's 0.18-μm CMOS process. The input frequency was set to 100 MHz, and the simulation results are all tabulated in Table I.

HSPICE software was used to construct the models of our proposed ETA and the conventional adders. 100 sets of inputs were randomly created using the C program "random()" function. For each set of input, we ran the simulation for each adder and recorded the power consumption. With 100 sets of results, average power consumption was determined. The worst case input was calculated and used to simulate the delay. The transistor count was derived directly from the HSPICE software.

Comparing the simulation results of our proposed ETA with those of the conventional adders (see Table I), it is evident that the ETA performed the best in terms of power consumption, delay, and Power-Delay Product (PDP). The PDP of the ETA is noted to be 66.29%, 77.44%, 83.70%, and 75.21% better than the RCA, CSK, CSL, and

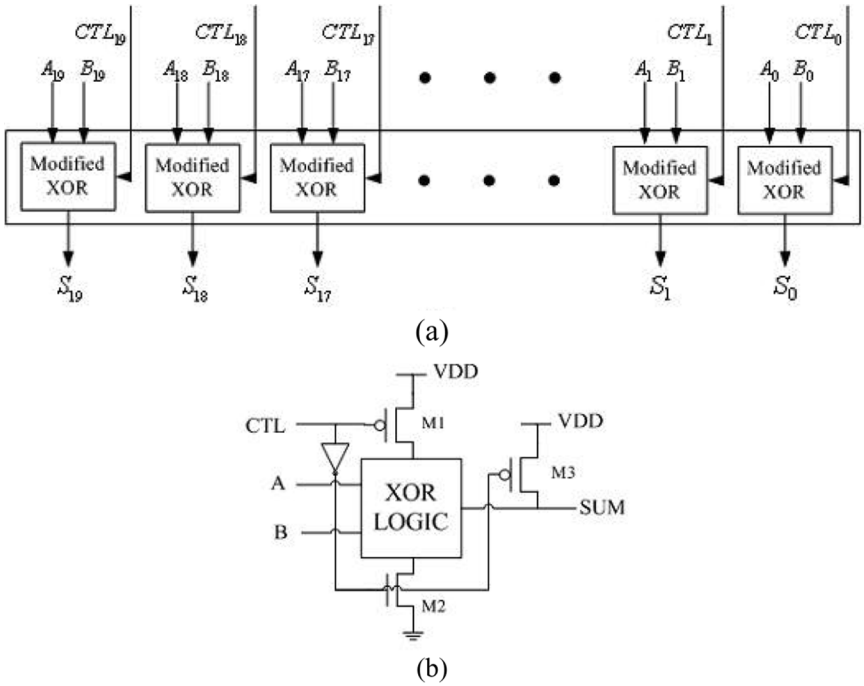

Fig. 5. Carry-free addition block. (a) Overall architecture and (b) schematic diagram of a modified XOR gate.

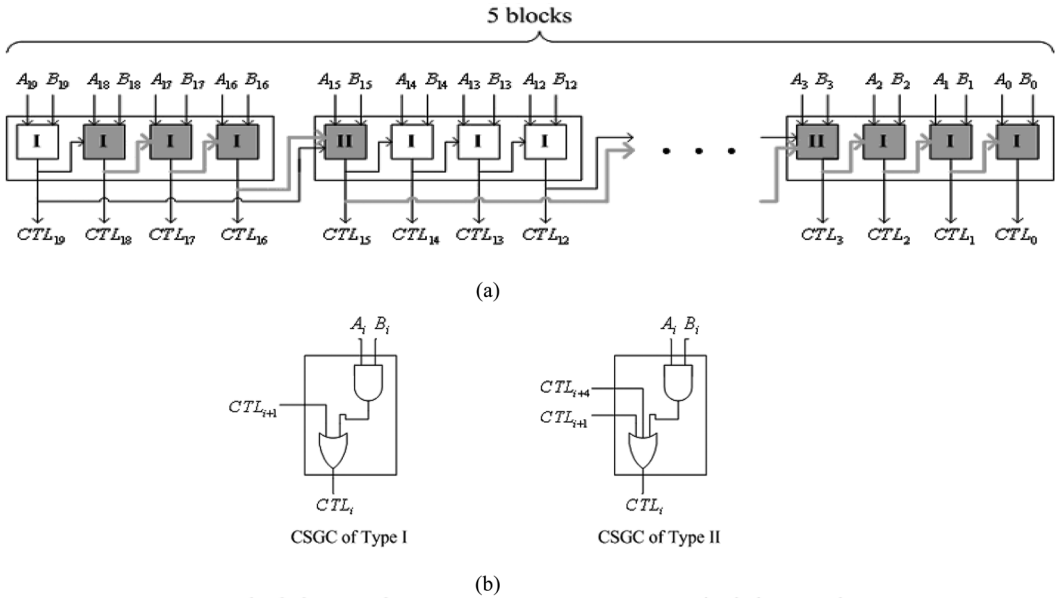

Fig. 6. Control block. (a) Overall architecture and (b) schematic implementations of CSGC.

TABLE I
SIMULATION RESULT FOR ETA VERSUS CONVENTIONAL ADDERS

Type of Adder	Power (mW)	Delay (ns)	PDP (pJ)	PDP saving (%)	Transistor Count
RCA	0.22	4.04	0.89	66.29	896
CSK	0.46	2.90	1.33	77.44	1728
CSL	0.60	3.06	1.84	83.70	2176
CLA	0.51	2.37	1.21	75.21	2208
ETA	0.13	2.29	0.30	N.A.	1006

CLA, respectively. As for transistor count, the proposed ETA is almost as good as the RCA.

V. APPLICATION OF ERROR-TOLERANT ADDER IN DIGITAL SIGNAL PROCESSING

In image processing and many other DSP applications, fast Fourier transformation (FFT) is a very important function. The computational process of FFT involves a large number of additions and multiplications. It is therefore a good platform for embedding our proposed ETA. To prove the feasibility of the ETA, we replaced all the common additions involved in a normal FFT algorithm with our proposed addition arithmetic.

As we all know, a digital image is represented by a matrix in a DSP system, and each element of the matrix represents the color of one pixel of the image. To compare the quality of images processed by both the conventional FFT and the inaccurate FFT that had incorporated our proposed ETA, we devised the following experiment. An image was

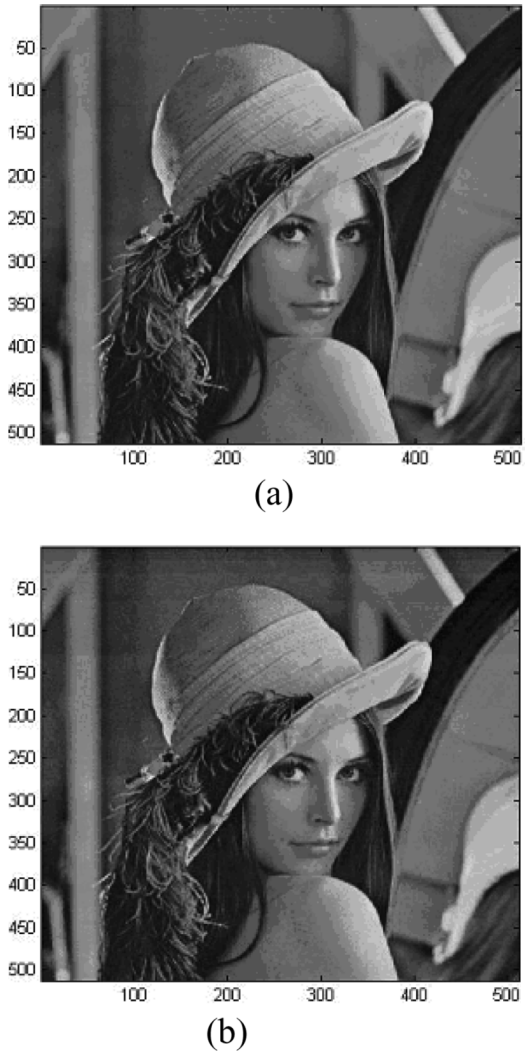

Fig. 7. Images after FFT and inverse FFT. (a) Image processed with conventional adder and (b) image processed with the proposed ETA.

first translated to a matrix form and sent through a standard system that made used of normal FFT and normal reverse FFT. The matrix output of this system was then transformed back to an image and presented in Fig. 7(a). The matrix of the same image was also processed in a system that used the inaccurate FFT and inaccurate reverse FFT, where both FFTs had incorporated the 32-bit ETA described in Section III, with the processed image given in Fig. 7(b).

Although the two resultant matrices of the same image were different, the two pictures obtained (see Fig. 7) look almost the same. Fig. 7(b) is slightly darker and contains horizontal bands of different shades of gray. With a MAA setting of 95%, the AP of the matrix representation of Fig. 7(b) is 98.3% as compared to the matrix representation of Fig. 7(a).

The comparison between the two images in Fig. 7 shows that the quality loss to the image using our proposed ETA is negligible and can be completely tolerated by human eyes. These simulation results have proven the practicability of the ETA proposed in this paper.

VI. Conclusion

In this paper, the concept of error tolerance is introduced in VLSI design. A novel type of adder, the error-tolerant adder, which trades certain amount of accuracy for significant power saving and performance improvement, is proposed. 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 superlow power consumption and high-speed performance are more important than accuracy. One example of such applications is in the DSP application for portable devices such as cell phones and laptops.

Acknowledgment

W. Zhang would like to thank the Nanyang Technological University (NTU) of Singapore for providing the graduate research scholarship and support. The authors appreciate also the help rendered by Mr. L. Y. Loy.

References

[1] A. B. Melvin, "Let's think analog," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2005, pp. 2–5.
[2] International Technology Roadmap for Semiconductors [Online]. Available: http://public.itrs.net/
[3] 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.
[4] M. A. Breuer, S. K. Gupta, and T. M. Mak, "Design and error-tolerance in the presence of massive numbers of defects," *IEEE Des. Test Comput.*, vol. 24, no. 3, pp. 216–227, May-Jun. 2004.
[5] M. A. Breuer, "Intelligible test techniques to support error-tolerance," in *Proc. Asian Test Symp.*, Nov. 2004, pp. 386–393.
[6] 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.
[7] I. 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.
[8] 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.
[9] H. H. Kuok, "Audio recording apparatus using an imperfect memory circuit," U.S. Patent 5 414 758, May 9, 1995.
[10] T. Y. Hsieh, K. J. Lee, and M. A. Breuer, "Reduction of detected acceptable faults for yield improvement via error-tolerance," in *Proc. Des., Automation and Test Eur. Conf. Exhib.*, 2007, pp. 1–6.
[11] K. V. Palem, "Energy aware computing through probabilistic switching: A study of limits," *IEEE Trans. Comput.*, vol. 54, no. 9, pp. 1123–1137, Sep. 2005.
[12] S. Cheemalavagu, P. Korkmaz, and K. V. Palem, "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, Sep. 2004, pp. 402–403.
[13] P. Korkmaz, B. E. S. Akgul, K. V. Palem, and L. N. Chakrapani, "Advocating noise as an agent for ultra-low energy computing: Probabilistic complementary metal-oxide-semiconductor devices and their characteristics," *Jpn. J. Appl. Phys.*, vol. 45, no. 4B, pp. 3307–3316, 2006.
[14] 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.
[15] 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.
[16] 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.
[17] O. Bedrij, "Carry select adder," *IRE Trans. Electron. Comput.*, vol. EC-11, pp. 340–346, 1962.
[18] O. MacSorley, "High speed arithmetic in binary computers," *IRE Proc.*, vol. 49, pp. 67–91, 1961.
[19] Y. Kiat-Seng and R. Kaushik, *Low-Voltage, Low-Power VLSI Subsystems*. New York: McGraw-Hill, 2005.