Approximate Adder with Hybrid Prediction and Error Compensation Technique

Xinghua Yang[1], Yue Xing[1], Fei Qiao, Qi Wei, Huazhong Yang
Institute of Circuits and Systems, Dept of Electronic Engineering, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology
Email: qiaofei@tsinghua.edu.cn

***Abstract*—This paper proposed an approximate adder to accelerate computation and reduce energy consumption for error-resilient applications with a moderate output quality losses. The computation acceleration comes from the prediction scheme for the adder circuit, where the critical path is divided into multiple short fragments and a paralleling addition progress is enabled. The energy consumption is reduced as the result of trimming the registers from the lower predictors of the design. Furthermore, a simple module for error compensation is inserted into the approximate part of the circuit to decrease the relative error with very little hardware cost. Being simulated with 65nm CMOS process, $2.82X$ speedups and 57.8% energy-efficiency improvements have been achieved compared with traditional adders. Compared with the current high performance approximate adders, the proposed adder shows 6.9% energy-savings with 2 orders of reduction in relative error using random test data. At last, the proposed approximate adder is adopted in DCT processing, where more than $10dB$ PSNR increase can be achieved, compared with the current counterpart designs.**

I. Introduction

Approximate computing has become an effective technology in recent years with the booming development of mobile devices and embedded system, where high speed signal processing circuit with energy-efficient property can be achieved with this technique [1]. The main reason is that many practical systems or applications present inherent error tolerance, from which a tradeoff between output quality and energy efficiency can be established. For example, in multimedia domain, most of the image processing output are finally displayed for human eyes, however, the visual sense ability of human beings in perceiving images or videos is limited. Thus, a small portion of erroneous results in image processing will not be detected obviously in practical applications, especially for the image/video sensor network, where an acceptable image output, not with high quality, is just enough. This property, called error-resilient, provides a method to make tradeoffs between high-speed or energy-efficient processing with the precision of the output. Currently, most of the digital signal processing circuits are designed to ensure a precise result. However, when the output quality can be relaxed, those circuits are not efficient, and more improvements could be obtained with the utilization of approximate computing.

Previously, extensive researches have been presented to exploit approximate technique in digital signal processing (DSP) block, including approximate computation and storage, where the data calculation and storage on-chip or off-chip are both carried out in various levels, from design approach to circuit implementation, with the results of improvements on energy efficiency. As pointed out in [2], addition is one of the most important processing in DSP due to the large computation from convolution or kernel functions. Thus, approximate adders have been explored in a wide range of researches from architecture to transistor levels.

In [3], parts of the transistors are trimmed from the original circuit of the adder to shorten the critical path and less power and energy are consumed, however, this design presents large effort of customization, which makes it difficult to be used in automatic synthesis. For architecture-level design, carry look ahead adder (CLA) is a very powerful design to achieve higher speed than traditional ripple carry adder (RCA) by accelerating the process of carry signal propagation. However, most of current digital process unit needs a long bit word and this makes the logic block for the carry signal prediction extremely huge, which is inefficient in practical design. In [4], a new adder named Error-Tolerance Adder-II (ETAII) is presented based on CLA. A paralleling addition process is successfully realized with input segmentation and prediction. However, large error will be introduced to the computation results since the wrong prediction could happen in the upper bits, or in other words, the most significant bits (MSB) of the output. In order to reduce the error magnitude, an approximate carry skip adder (ACSA) is presented in [5], where the error is reduced as the location of the wrong prediction is further pushed back in the output through a multiplexer, however, the error magnitude of the unit is still large and its critical path is relatively long since each fragment for the sum result depends on both carry signal generator and addition process.

In this paper, an approximate adder with hybrid prediction scheme and error compensation is presented. Three innovative techniques are proposed to the adder design to ensure the property of high energy efficiency and low error output

Xinghua Yang and Yue Xing are joint first authors and have the same contribution to this paper.

978-1-4673-9039-2/16 $31.00 © 2016 IEEE
DOI 10.1109/ISVLSI.2016.16

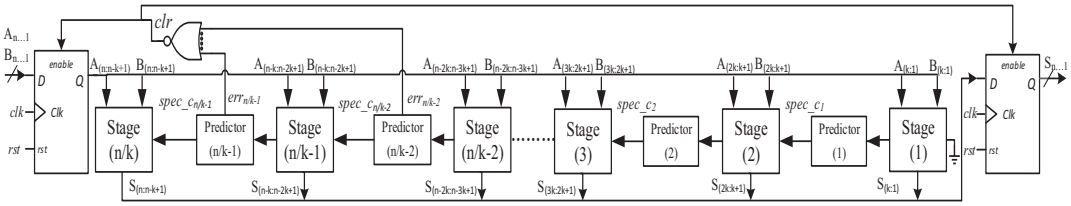

Figure 1. General Scheme of Proposed Approximate Adder

simultaneously. First, the conventional RCA is divided into multiple stages with a series of predictors with an error control method based on multistage latency scheme. Second, in order to further decrease the average cycle consumption and the energy consumption, the original prediction scheme is modified, where the D-Flip-Flops (DFFs) in lower predictors are eliminated to avoid extra clock cycles since the wrong prediction in lower part could propagate to higher bits of the output and decrease the performance of the adder. At last, an error compensation technique is used to further increase the output quality of the unit. The experiments show that $2.82X$ speedups and 57.8% energy-efficiency than traditional RCA and $1.06X$ speedups and 55.7% energy-efficiency than conventional CLA can be achieved. Compared with the ACSA in [5], which is one of the most efficient approximate designs, our proposed adder shows 6.9% energy-savings with 2 orders of reduction in relative error using random data for testbench. At last, the proposed adder is implemented in Discrete Cosine Transform (DCT) processing, where more than $10dB$ increase for PSNR has been achieved compared with ACSA.

The remainder of this paper is organized as follows. Previous related work will be analyzed in the next section. In section 3, we will describe our proposed approximate adder. Two necessary experiments covering both random input data and DCT algorithms are presented in the section 4. Finally, conclusions will be drawn in section 5.

II. Related Work

In this section, the circuit scheme of conventional RCA, CLA, ETAII in [4] and ACSA in [5] will be analyzed in detail. As pointed out in [6], RCA is a common type of adder in digital circuit design, in which a series of one-bit-full adders are connected in sequence and the higher output depends on lower carry signals. The delay of RCA is $O(n)$ and the critical path starts from the lowest bit to the last one. However, the probability to activate this critical path is very small [6], [7], which provides the foundation to design speculative-based adder in following researches. In order to obtain the carry signal in advance, for CLA, the real value of carry signals for higher computation block is calculated using signal generation method. The critical path can be reduced efficiently. However, the process of carry signal generation is complicated in CLA, which could produce large logic area and will result in a big power consumption.

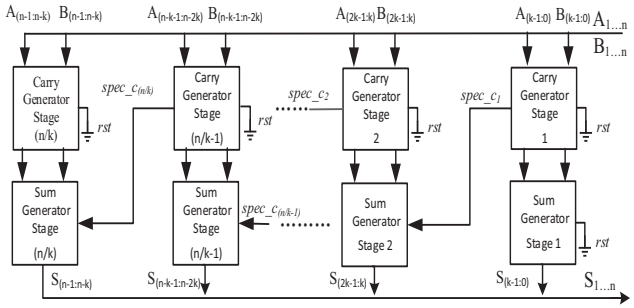

Figure 2. Circuit Scheme of ETAII in [4]

Based on the idea of CLA, ETAII in [4] makes a full use of paralleling calculation and introduced approximation to reduce the power overhead as shown in Fig.2. The adder is divided into several stages. Each of the stages has a carry generator and a sum generator. The output of one stage comes from its sum generator with the previous carry signal. Thus, the critical path is composed of one sum stage and carry signal generator. However, the carry signal generator involves only parts of the lower input data, which could cause large error when a wrong prediction happens in the upper stage of the adder. For 32-bits ETAII with 4 bits for each stage, the maximum error magnitude could be 2^{28}, which is too large that the adder have little practical value in real application.

In [5], an approximate adder with carry skip technique (ACSA) is proposed based on ETAII, in which a multiplexor is used to choose the carry signal for the sum generator in each stage. Different from ETAII, this adder detects the property of carry propagation in previous stage. When all the bits in previous $(i-1)^{th}$-stage is in carry propagation mode, it will select the carry signal from $(i-2)^{th}$-stage. In this way, the error rate of the adder will be decreased. Furthermore, a method for error compensation is also used. However, the error magnitude of this adder is still large. Take 32-bits adder with 4 bits for each stage, the maximum error magnitude could be 2^{20}. Meanwhile, the extra multiplexors could consume more area, energy and delay as well.

III. Proposed Circuits Architecture

In this section, an approximate adder using multistage latency scheme with hybrid prediction scheme and error

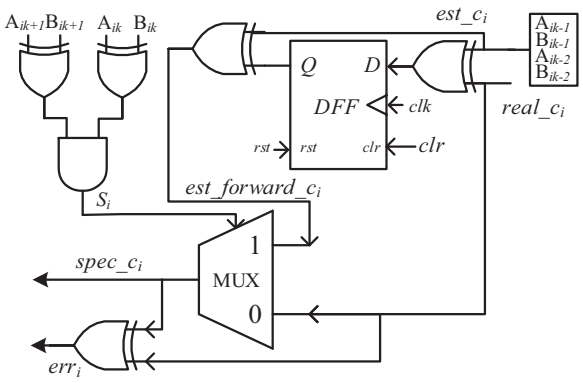

Figure 3. Circuit Scheme of Our Proposed Predictor

compensation technique is presented. Three techniques will be adopted in our proposed adder in order to achieve a high energy-efficiency adder with low error output.

A. Critical path division with predictors and error control for higher bits of the output based on multistage latency scheme

As shown in Fig.1, the n-bit RCA ($A_{n...1}$ and $B_{n...1}$ are the input data) is first divided into multiple stages ($i = 1...n/k$) with k-bits for each fragment. A series of predictors are inserted and for each new input data, every predictor will produce a speculative value to the carry signal ($spec\_c_i$) so that all the stages could be computed in parallel. This prediction process is just like [4] and the performance of the adder will be improved significantly since the critical path is divided into short ones. However, as we analyzed in section II, the wrong prediction in upper stages of the final output will produce a huge error and this disadvantage also exists in [5]. Thus, we utilize the technique of multistage latency computing in [8] to guarantee a correct result for the higher bits of final output. As shown in Fig.1, the err_i signals produced by the upper $\lceil (n/k-1)/2 \rceil$ predictors are converged into the NOR-gate, from which a signal called clr is generated. On the other hand, no err_i signals are converged into the NOR-gate for the remaining $\lfloor (n/k-1)/2 \rfloor$ predictors. With this configuration, the computing process of the proposed adder is as following: for each new input data, every predictor will generate a speculative value to each stage and all the stages will be computed in parallel in first cycle. At the end of the first cycle, another cycle will be used if any predictor makes its err_i high (which means a wrong prediction is detected), at which time the clr signal in Fig.1 is low and D-Flip-Flops (DFFs) for input and output will be held. At the second cycle, the correct value of carry signal will be pushed forward and updated. This process will be operated until the clr signal is high (which means no wrong prediction exists in the upper $\lceil (n/k-1)/2 \rceil$ predictors).

With this computing scheme, two features can be observed. First, the final output from the $(k * \lfloor (n/k-1)/2 \rfloor + 1)^{th}$-bit to the n^{th}-bit will be precise. This is the main reason for our low error output property than [4], [5]. Second, since the wrong prediction in last cycle could be corrected in following one, a DFF should be adopted in each of the predictors to preserve the real value of carry signal. The circuit scheme of our proposed predictor is shown in Fig.3, which is composed of a DFF and other combinational logics. The clr signal in Fig.3 is from the output of NOR-gate in Fig.1. For each new input data ($A_{n...1}$ and $B_{n...1}$), the output Q of the DFF in Fig.3 will be first cleared to '0' with the clr signal, thus, for i^{th}-predictor, the est_c_i signal ($est\_c_i = A_{ik-1}B_{ik-1} + (A_{ik-1} + B_{ik-1})(A_{ik-2}B_{ik-2})$) will be pushed forward through the XOR-gate as $est\_forward\_c_i$ signal (any signal will be itself when XOR with logic '0'). For the MUX in Fig.3, when $S_i = (A_{ik+1} \oplus B_{ik+1}) \cap (A_{ik+2} \oplus B_{ik+2}) =' 0'$, which means the critical path has already been broken naturally with the input data, the value of $est\_forward\_c_i$ will be abandoned and the real carry signal ($real\_c_i$) from lower stage will be used for $spec\_c_i$ straightforwardly. This selection between $real\_c_i$ and $est\_forward\_c_i$ is effective since less cycles will be consumed as $S_i =' 0'$ will remove the possible wrong prediction. When $S_i =' 1'$, the $spec\_c_i$ will be $est\_forward\_c_i$ and whole prediction process will be carried out. At the end of first cycle, the err_i signal will be high if the $real\_c_i$ and $est\_forward\_c_i$ are not equal. In next cycle, the output Q of the DFF will be logic '1' and the wrong prediction in last cycle will be corrected.

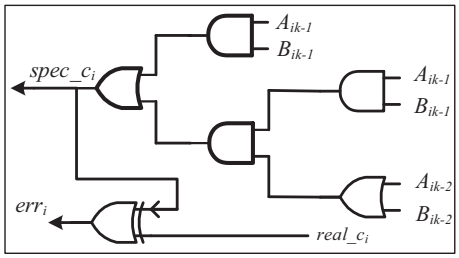

Figure 4. Modified Predictor Scheme with No DFF

B. Hybrid scheme with modified predictor

In previous description, all the predictors in Fig.1 have the same circuit structure even for the predictors with no err_i signals converged into the NOR-gate. In essence, the DFF in Fig.3 is used to preserve the correct value of carry signal. Thus, as no err_i signals are converged into the NOR-gate for the remaining $\lfloor (n/k-1)/2 \rfloor$ predictors, which means that the wrong prediction in these stages will not introduce more computation cycles, the DFFs of these predictors could be eliminated from the circuit in Fig.3 and only combinational logics are reserved as shown in Fig.4 (the err_i signal is reserved for error compensation as described in following part). This modification could obtain more power savings in

practical design. Another important advantage of trimming the DFF is that the wrong prediction from the remaining $\lfloor (n/k-1)/2 \rfloor$ predictors will not propagate to higher output.

Take the example in Fig.5 to illustrate the negative impact of DFFs in lower stages. A 16-bits adder with 3 predictors inserted is used. All the predictors are equipped with the scheme in Fig.3 and no DFFs are trimmed. Only the err_i signal from the 3^{th}-predictor is pushed into the NOR-gate in Fig.1. After the first clock cycle, another cycle will be used to correct the third predictor, however, since the DFF also exists in the first and second predictor, the wrong prediction to carry signal in first predictor will also be corrected. Thus, after second clock cycle, the first and third predictor will produce a correct carry signal, however, wrong prediction in second predictor will be detected. Since the err_i signal for second predictor has not been converged into the NOR-gate in Fig.1, no subsequent cycles will be allocated to the unit, which means that 2^8 error will be introduced into the final result. However, if the first and second predictor are realized with the circuit scheme in Fig.4, the wrong prediction in first predictor will not propagate to higher output. This effect will be more prominent when a long-bit adder with more predictors are used.

By trimming the DFFs for the remaining $\lfloor (n/k-1)/2 \rfloor$ predictors, the wrong prediction will stay in lower output and the power for the adder is also reduced since the DFFs consume a large area compared with one-bit-full adder. Furthermore, the process for modifying the predictors for lower stage is very simple as only the combinational logic are kept. For the accurate part, where the err_i signals are converged into the NOR-gate in Fig.1, the predictors are realized as the circuit scheme in Fig.3.

C. Error compensation

Error compensation is a useful method to further reduce error magnitude with very little area and delay cost. Since the DFFs are trimmed for the approximate stages of the adder, it will be certain that wrong prediction will be generated. Thus, another improvement in our design is that we apply error compensation scheme to further reduce the output error. Considering the prediction method in Fig.4, where the $spec\_c_i$ is computed as $A_{ik-1}B_{ik-1} + (A_{ik-1}+B_{ik-1})(A_{ik-2}B_{ik-2})$. In practical computation, if $spec\_c_i$ is '1', the prediction must be correct because either $A_{ik-1}B_{ik-1} = 1$ or $(A_{ik-1} + B_{ik-1})(A_{ik-2}B_{ik-2}) = 1$ will lead a carry signal '1' to next stage. So the wrong

3th-predictor				2th-predictor				1th-predictor							
0	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0

Figure 5. Illusion of the Negative Impact for the DFF in the remaining $\lfloor (n/k-1)/2 \rfloor$ Predictors

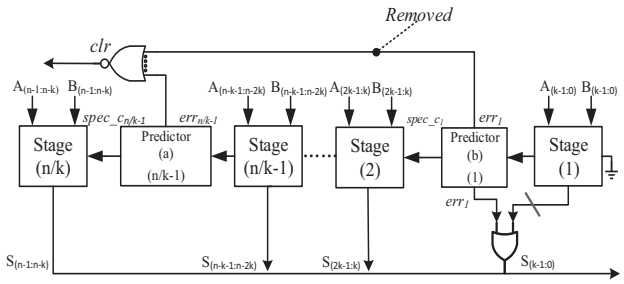

Figure 6. The Scheme of the Proposed Adder with Error Compensation Module. Predictor(a) is realized as shown in Fig.3 and Predictor(b) is realized as Fig.4.

prediction can only happen when $spec\_c_i$ is '0' but the real carry signal is '1'. The error compensation scheme can be described as shown in Fig.6. For the 1^{th}-predictor in approximate stage, when its err_i signal become high, the sum output of 1^{th}-stage will be set to '1111' to compensate the error to the final result. This process can be realized just with OR-gate. Through the OR-gate, the sum output of this stage is changed into '1111' when the err_i signal becomes high. This method of error compensation could be applied to all the the remaining $\lfloor (n/k-1)/2 \rfloor$ predictors so that the final output error could be reduced.

As analyzed in Section 2, the error magnitude of ACSA in [5] can be 2^{20} in maximum for an 32-bits adder with 4 bits for each stage. For our proposed design with all of the above techniques, the final output error will be 2^{12} in maximum, which is far more smaller than previous work. In following section, all the adders will be verified with random test data and implemented in real applications, where the performance, energy and other parameters of the unit will be given.

IV. Experiment Results

To demonstrate the efficiency of the proposed approximate adder, conventional accurate adder of RCA, CLA and high performance approximate adder of ETAII in [4],ACSA in [5] with our proposed adder are simulated using 65nm CMOS process. Two simulations are implemented, which covers both random data test and DCT processing algorithm.

A. Error Metric for Evaluation with Random Test Data

We use relative error (RE) to describe the quality of the results, just like in [9], which is defined as:

$$RE = |\frac{S - S_{cor}}{S_{cor}}| \quad (1)$$

Where S is the approximate result and S_{cor} is the correct value. For this simulation, we use the RMS(Root Mean Square) of the RE as the final parameter to evaluate the output quality. As for DCT processing, the PSNR(Peak Signal to Noise Ratio) is a common standard to describe the quality of a processed picture.

Table I
PERFORMANCE AND AREA OF DIFFERENT ADDERS

	RCA	CLA	ETAII [4]	ACSA [5]	Proposed
Delay(ns)	1.78	0.67	0.43	0.53	0.56
Avg_cycle	1	1	1	1	1.12
Avg_delay(ns)	1.78	0.67	0.43	0.53	0.63
Power(mW)	0.36	0.90	0.37	0.54	0.43
Energy(pJ)	0.64	0.61	0.16	0.29	0.27
Area(μm^2)	601	1076	716	978	840
RMS of RE	0	0	19.0922	0.6768	0.0033
EEP(pJ)	0	0	3.0547	0.1962	0.00089

B. Simulation-I:Random Data Test

In this simulation, 32-bits input data are used for RCA and CLA. For the adders in [4], [5] and our proposed design, all the units are divided into 8 stages with 4 bits for each fragment. As described before, for the proposed adder, the upper four predictors in Fig.1 will be realized as the scheme in Fig.3 and the remaining lower predictors will be equipped with the scheme in Fig.4. Moreover, the error compensation is also adopted in final design. All of these adders are first coded in Verilog and simulated by DesignCompiler [10], where the delay and area of each unit could be achieved. For power analysis, we use PrimeTime [10] to simulate the gate-level netlist from DesignCompiler to obtain the final power consumption. It should be noted that since our proposed adder may consume variable clock cycles for once computation, the average cycle (Avg_cycle) of our design after 10^7 random data simulation in Modelsim is obtained. The final average delay (Avg_delay) of our design should be the value of Delay multiplying Avg_cycle as shown in Table-I. Furthermore, we modeled all the approximate adders with their exact functionality in C++, the RMS of RE for each unit as we described before could be achieved after we simulated all the adders with random test data. At last, the value of EEP (Energy and RMS of RE Product) is listed, which is another parameter to evaluate the efficiency of the proposed design.

All the simulation results are listed in Table-I. Comparing with traditional RCA, our proposed adder can achieve $2.82X$ speedups and 57.8% energy savings. Comparing with conventional CLA, our proposed adder can achieve $1.06X$ speedups and 55.7% energy savings. It can be seen that the ETAII adder has a faster speed and lower power than other adders, however, its RMS of RE is too big. This property makes it inavailble in real applications as we demonstrate in following part. Compared with the ACSA, our design achieves 6.9% less energy consumption. More importantly, our proposed adder presents the lowest RMS of RE. It is 2 orders reduction in RMS of RE of our design than ACSA and even more when the value of EEP is evaluated. This is due to the multistage latency scheme to ensure a precise output for the upper output bits and the error compensation circuit to further reduce the error magnitude. This property of low error output is quite important when the approximate adders are applied into practical design as we demonstrate in following simulation.

C. Simulation-II:DCT

We built a simulator in C++ for DCT processing, in which the RCA, CLA, ACSA, ETAII and our proposed adders are modeled in functionality for addition processing. The DCT coefficient matrix comes from [3]. Since the 2-DCT can be divided into two 1-DCT processing, we use 20-bits long adder for the first 1-DCT processing and 32-bits long adder for the second one. Each time a block of 8x8 pixels from the image will be pushed into the simulator for DCT processing.

As pointed in [3], different elements of the final matrix after 2-DCT processing have the different significance to recover the image. Basically, the left and up elements of the final matrix carries the lower frequency components of the image, which are more important in recovering the original picture. Thus, precise addition should be applied to the computation of these low frequency components. In our simulator, the upper left 2x2 results are calculated in precise mode and the other components are computed in approximate mode. With all of these configurations, the final matrix after 2-DCT could be achieved, which is then pushed into Matlab, where the standard 2-IDCT function is used to obtain the original image and the corresponding PSNR will be computed as shown in Fig.7. It can be seen that our proposed adder provides a large improvement in PSNR compared with ACSA in [5] and ETAII [4]. As we discussed in previous part, the ETAII adder presents a higher performance and energy efficiency than other adders, however, it can hardly be used in practical design. This property can also be seen compared with ACSA. As for RCA/CLA implementation, which means the accurate computing process, our design presents 2dB lost in PSNR. However, as pointed out in [3], this final output quality of image is acceptable in practical application and the tradeoff between energy efficiency and output quality is worthwhile considering the results in Table-I.

V. CONCLUSION

In this paper, an approximate multistage latency adder with hybrid prediction scheme and error compensation technique is presented. Parts of the DFFs in lower parts of the adder are removed from their corresponding predictors to prevent the wrong prediction to propagate into higher output bits and more power savings can also be achieved with this modification. Furthermore, an error compensation module is inserted into the approximate output, which is used to increase the output quality of the adder with little area and power cost. The proposed adder has been implemented in verilog and compared with conventional adders and corresponding counterparts. Improvements in performance, energy efficiency and output error have been demonstrated

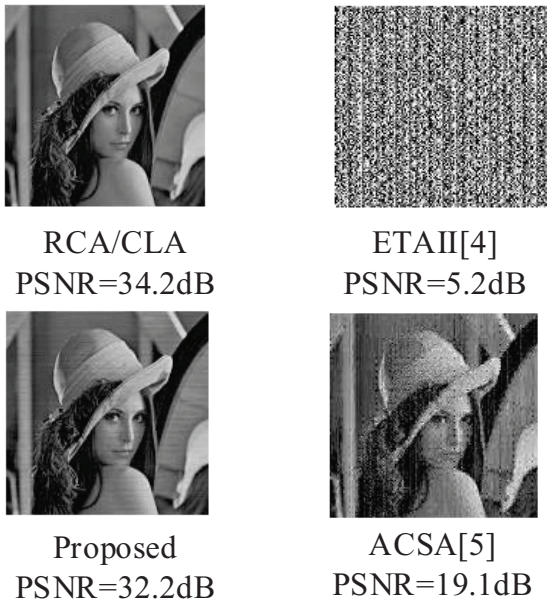

Figure 7. Output Images after using: (a)ACSA;(b)Proposed Adder;(c)RCA

through extensive simulation. In future, more researches will be carried on the model of tradeoffs between output quality and performance improvements and the proposed adder will be applied to more applications, such as machine learning in image/video processing area.

REFERENCES

[1] K. Roy and A. Raghunathan, "Approximate computing: An energy-efficient computing technique for error resilient applications," in *VLSI (ISVLSI), 2015 IEEE Computer Society Annual Symposium on*. IEEE, 2015, pp. 473–475.

[2] A. A. Del Barrio, R. Hermida, S. O. Memik, J. M. Mendias, and M. C. Molina, "Multispeculative addition applied to datapath synthesis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, no. 12, pp. 1817–1830, 2012.

[3] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 1, pp. 124–137, 2013.

[4] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *Integrated Circuits, ISIC'09. Proceedings of the 2009 12th International Symposium on*. IEEE, 2009, pp. 69–72.

[5] Y. Kim, Y. Zhang, and P. Li, "An energy efficient approximate adder with carry skip for error resilient neuromorphic vlsi systems," in *Proceedings of the International Conference on Computer-Aided Design*. IEEE Press, 2013, pp. 130–137.

[6] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Test Symposium (ETS), 2013 18th IEEE European*. IEEE, 2013, pp. 1–6.

[7] S. Yinan, L. Yongpan, W. Zhibo, and Y. Huazhong, "Multi-stage function speculation adders," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 98, no. 4, pp. 954–965, 2015.

[8] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in *Proceedings of the conference on Design, automation and test in Europe*. ACM, 2008, pp. 1250–1255.

[9] J. Schlachter, V. Camus, and C. Enz, "Near/sub-threshold circuits and approximate computing: The perfect combination for ultra-low-power systems," in *VLSI (ISVLSI), 2015 IEEE Computer Society Annual Symposium on*. IEEE, 2015, pp. 476–480.

[10] DesignCompiler and PrimeTime, "Synopsys inc," 2000.