

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 5, May 2019

Design of Reconfigurable Adder by Using Carry Maskable Technique

M.Ramya ^{#1}, R.Saranpriya ^{*2}

Associate Professor, Department of ECE, Sasurie College of Engineering, Vijayamangalam, Tirupur,
Tamilnadu, India^{#1}

II-M.E-VLSI Design, Department of ECE, Sasurie College of Engineering, Vijayamangalam, Tirupur,
Tamilnadu, India^{*2}

ABSTRACT: Approximate computing is a promising approach for low power IC design and has recently received considerable research attention. To accommodate dynamic levels of approximation, a few accuracy configurable adder designs have been developed in the past. However, these designs tend to incur large area overheads as they rely on either redundant computing or complicated carry prediction. Some of these designs include error detection and correction circuitry, which further increases area. Moreover, we propose a delay-adaptive self-configuration technique to further improve accuracy-delay-power trade off. Addition is a key fundamental function for many error tolerant applications. Approximate addition is considered to be an efficient technique for trading off energy against performance and accuracy. This paper proposes a carry maskable adder whose accuracy can be configured at runtime. The proposed scheme can dynamically select the length of the carry propagation to satisfy the quality requirements flexibly. Compared with a conventional ripple carry adder and a conventional carry look-ahead adder, the proposed 16-bit adder reduced power consumption by 54.1% and 57.5% and critical path delay by 72.5% and 54.2%, respectively. In addition, results from an image processing application indicate that the quality of the processed images can be controlled by the proposed adder.

I. INTRODUCTION

Addition is a key fundamental function for many error tolerant applications. Approximate addition is considered to be an efficient technique for trading off energy against performance and accuracy. The objective of the project is to avoid the high power consumption and reduce path delay in the adder circuit design. The result of these adders is implementing in the image. We get the results from an image processing application indicate that the quality of the processed images can be controlled by these adder. Approximate adders are widely being advocated for developing hardware accelerators to perform complex arithmetic operations. Most of the state-of-the-art accuracy configurable approximate adders utilize some integrated Error Detection and Correction (EDC) circuitry. Consequently, the accumulated area overhead due to the EDC (integrated within individual adders) is significant. In this paper, we propose a low-cost Consolidated Error Correction (CEC) unit, that essentially corrects the accumulated error at the accelerator output. The proposed CEC is based on a mathematical model of approximation error. We integrate our CEC unit in approximate hardware accelerators deployed in different applications to demonstrate its area savings and speed enhancement compared to state-of-the-art.

Addition is a key fundamental function for many error tolerant applications. Approximate addition is considered to be an efficient technique for trading off energy against performance and accuracy. This paper proposes a carry maskable adder whose accuracy can be configured at runtime. The proposed scheme can dynamically select the length of the carry propagation to satisfy the quality requirements flexibly. We design 12-bit adder reduced power consumption and critical path delay respectively.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 5, May 2019

II. LITERATURE REVIEW

In this section, we discuss the ProSpeculative adders have attracted strong interest for reducing critical path delays to sub-logarithmic delays by exploiting the tradeoffs between reliability and performance. Speculative adders also find use in the design of reliable variable latency adders, which combine speculation with error correction to achieve high performance for low area overhead over traditional adders. This paper describes speculative carry select addition (SCSA), a novel function speculation technique for the design of low error-rate speculative adders and low overhead, high performance, reliable variable latency adders. We develop an analytical model for the error rate of SCSA to facilitate both design exploration and convergence. We show that for an error rate of 0.01% (0.25%), SCSA-based speculative addition is 10% faster than the DesignWare adder with up to 43% (56%) area reduction. Further, on average, variable latency addition using SCSA-based speculative adders is 10% faster than the DesignWare adder with area requirements of -19% to 16% (17% to 29%) for unsigned random (signed Gaussian) inputs posed speculative addition, called speculative carry select addition (SCSA). SCSA comes from the observation of the carry chain in addition. During the discussion, we employ unsigned binary addition and random inputs to illustrate SCSA. An example of 16-bit addition.

Several recently proposed design techniques leverage the inherent error resilience of applications for improved efficiency (energy or performance). Hardware and software systems that are thus designed may be viewed as scalable effort system, since they offer the capability to modulate the effort that they expend towards computation, thereby allowing for tradeoffs between output quality and efficiency. We propose the concept of Dynamic Effort Scaling (DES), which refers to dynamic management of the control knobs that are exposed by scalable effort systems. We argue the need for DES by observing that the degree of resilience often varies significantly across applications, across datasets, and even within a dataset. We propose a general conceptual framework for DES by formulating it as a feedback control problem, wherein the scaling mechanisms are regulated with the goal of maintaining output quality within a certain specified limit.

We present an implementation of Dynamic Effort Scaling in the context of a scalable-effort processor for Support Vector Machines, and evaluate it under various application scenarios and data sets. Our results clearly demonstrate the benefits of the proposed approach — statically setting the scaling mechanisms leads to either significant error overshoot or significant opportunities for energy savings left on the table unexploited. In contrast, DES is able to effectively regulate the output quality while maximally exploiting the time-varying resiliency in the workload.

Approximate computing has recently emerged as a promising approach to energy-efficient design of digital systems. Approximate computing relies on the ability of many systems and applications to tolerate some loss of quality or optimality in the computed result.

By relaxing the need for fully precise or completely deterministic operations, approximate computing techniques allow substantially improved energy efficiency. This paper reviews recent progress in the area, including design of approximate arithmetic blocks, pertinent error and quality measures, and algorithm-level techniques for approximate computing.

Approximate adders are widely being advocated for developing hardware accelerators to perform complex arithmetic operations. Most of the state-of-the-art accuracy configurable approximate adders utilize some integrated Error Detection and Correction (EDC) circuitry. Consequently, the accumulated area overhead due to the EDC (integrated within individual adders) is significant. In this paper, we propose a low-cost Consolidated Error Correction (CEC) unit, that essentially corrects the accumulated error at the accelerator output. The proposed CEC is based on a mathematical model of approximation error. We integrate our CEC unit in approximate hardware accelerators deployed in different applications to demonstrate its area savings and speed enhancement compared to state-of-the-art.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 5, May 2019

III. METHODOLOGY

CARRY MASKABLE ADDER

A conventional half adder is shown in above diagram. A 2input XOR gate is used to generate sum s and a 2-input AND gate is used to generate carry C_{out} . An equivalent circuit of a half adder and the dashed frame represents an equivalent circuit of a 2-input XOR gate. Since there is a 2-input NAND gate in the dashed frame, we reuse it and add an INV gate to generate the carry signal C_{out} . The outputs of the 2-input NAND and OR gates in the dashed frame are named u and w , respectively. Table 1 is the truth table for the equivalent circuit of a half adder. when the internal signal u is 1, the sum s is equal to a OR b and the carry C_{out} is 0. This means that, if u is controllable and can be controlled to 1, the carry propagation will be masked and the sum s will be equal to a OR b . The sum $s = a \text{ OR } b$ is different from the accurate sum ($=a \text{ XOR } b$) only when both a and b are 1. In other words, the sum $s = a \text{ OR } b$ can be considered as an approximate sum. The selectivity between the accurate and approximate sums can be achieved by a control signal, which is used to control u to be a NAND b , or to be 1.

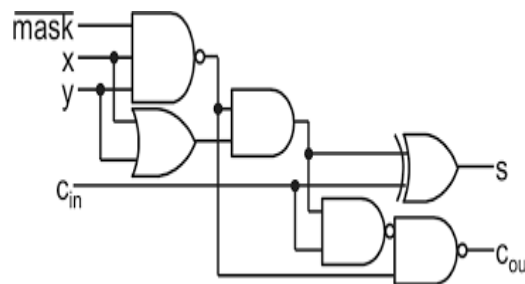


Fig 3.7.3 Schematic diagram of carry maskable adder

We add a signal named “mask_x” as the control signal and use a 3-input NAND gate to replace the 2-input one in the dashed frame. This is called a carry-maskable half adder (CMHA) and shown in Fig. 3. When $mask_x = 0$, the sum $s = a \text{ OR } b$, and the carry $C_{out} = 0$; otherwise, when $mask_x = 1$, the sum $s = a \text{ XOR } b$, and the carry $C_{out} = a \text{ AND } b$. Similar considerations apply to a full adder.

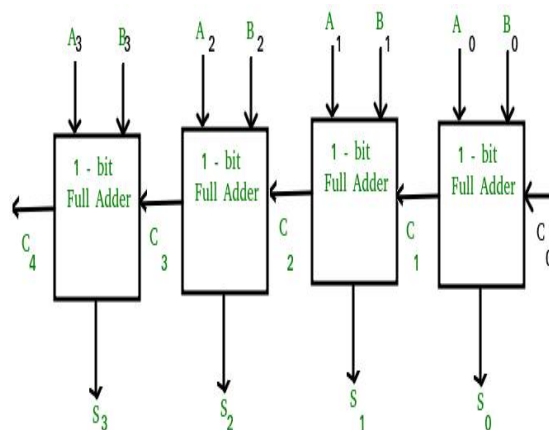


Fig 3.7.4: Schematic Diagram of 8 bit Carry Maskable Adder

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

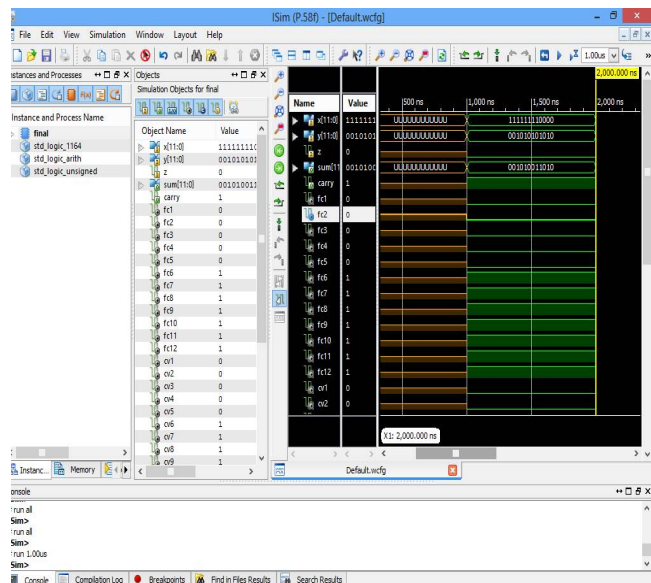
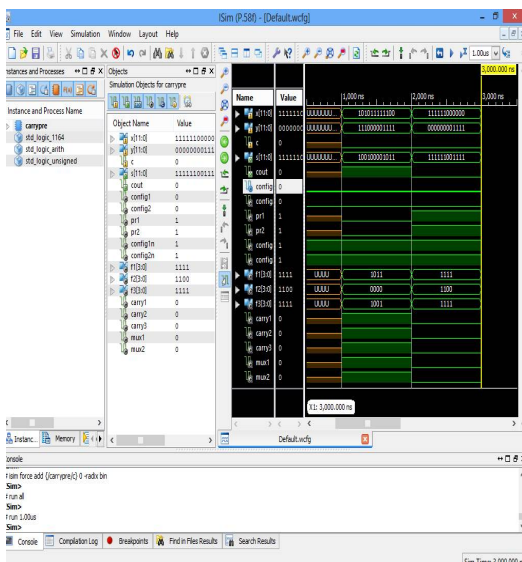
Visit: www.ijirset.com

Vol. 8, Issue 5, May 2019

When $mask_x = 0$ and $Cin = 0$, the sum $s = a$ OR b , and the carry $Cout = 0$, then obviously switching activities become smaller, and dynamic power consumption is reduced. This full adder is called a carry-maskable full adder (CMFA). An n -bit adder, which is implemented using one CMHA and $(n-1)$ CMFA, is called an n -bit carry-maskable adder (CMA).

IV. SIMULATION OF ADDER DESIGN

This part explains about the creation of adder circuit, calculation of addition from we get a more number of bits in the adder circuits.



IV. CONCLUSION AND FUTURE ENHANCEMENTS

A Modified Carry Adder proposes an accuracy-configurable approximate adder that does not require any additional logic blocks to achieve accuracy configuration. The experimental results demonstrate that the proposed adder is able to deliver more significant energy savings than the conventional RCA and CLA while maintaining a significantly small circuit area.

Compared to other previously studied adders, the experimental results from both the circuit and application levels demonstrate that our proposed adder delivers greater improvements in energy saving, design area, and accuracy. The choice of a particular adder depends on the intended application. In CLA and RCA the delay performance of the circuit is improved in the design without affecting the power dissipation of the circuit.

REFERENCES

- [1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in IEEE European Test Symposium, pp. 1–6, 2013.
- [2] V. Chippa, A. Raghunathan, K. Roy, and S. Chakradhar, "Dynamic effort scaling: Managing the quality-efficiency tradeoff," in DAC, pp. 603–608, 2011.
- [3] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," TCAD, vol. 32, no. 1, pp. 124–137, 2013.
- [4] S. Venkataramani, K. Roy, and A. Raghunathan, "Substitute-and-simplify: A unified design paradigm for approximate and quality configurable circuits," in DATE, pp. 1367–1372, 2013.
- [5] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in DATE, pp. 1257–1262, 2012.
- [6] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in ISIC, pp. 69–72, 2009.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 5, May 2019

- [7] J. Hu and W. Qian, "A new approximate adder with low relative error and correct sign calculation," in DATE, pp. 1449–1454, 2015.
- [8] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," in ICCAD, pp. 728–735, 2012.
- [9] S. Mazahir, O. Hasan, R. Hafiz, M. Shafique, and J. Henkel, "An area-efficient consolidated configurable error correction for approximate hardware accelerators," in DAC, pp. 1–6, 2016.
- [10] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in DAC, pp. 820–825, 2012.
- [11] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," in DAC, pp. 1–6, 2015.
- [12] V. Benara and S. Purini, "Accurus: A fast convergence technique for accuracy configurable approximate adder circuits," in IEEE Computer Society Annual Symposium on VLSI, pp. 577–582, 2016.
- [13] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," in ICCAD, pp. 48–54, 2013.
- [14] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: A reconfigurable approximate carry look-ahead adder," IEEE Transactions on Circuits and Systems II: Express Briefs, accepted.
- [15] M.-T. Sun, T.-C. Chen, and A. M. Gottlieb, "VLSI implementation of a 16*16 discrete cosine transform," IEEE Transactions on Circuits and Systems, vol. 36, no. 4, pp. 610–617, 1989.