ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

ISSN 2350-0174
Vol.02, Issue.11
December -2020
Pages: -283-288

A LOW-POWER HIGH-SPEED ACCURACY CONTROLLABLE APPROXIMATE MULTIPLIER DESIGN

[1] ADDANKI UMA MAHESWARI, [2] ALURI BULLI BABU
[1]PG Scholar, Dept. of ECE, NOVA COLLEGE OF ENGINEERING AND TECHNOLOGY, Vijayawada, A.P
[2]Assistant Professor, Dept. of ECE, NOVA COLLEGE OF ENGINEERING AND TECHNOLOGY, Vijayawada, A.P

ABSTRACT: Approximate circuits have been considered for error-tolerant applications that can tolerate some loss of accuracy with improved performance and energy efficiency. Multipliers are key arithmetic circuits in many such applications such as digital signal processing (DSP). In this paper, a novel approximate multiplier with a lower power consumption and a shorter critical path than traditional multipliers is proposed for high-performance DSP applications This paper proposes an accuracy-controllable multiplier whose final product is generated by a carry-maskable adder. The proposed scheme can dynamically select the length of the carry propagation to satisfy the accuracy requirements flexibly. The partial product tree of the multiplier is approximated by the proposed tree compressor. An 8* 8 multiplier design is implemented by employing the carry maskable adder and the compressor. As an extension of this concept carry skip adder is used instead of carry maskable adder for area and time parameters optimizations. This improvement yields more enhancements in all type of applications.

KEYWORDS: Approximation, Multiplier, Carry maskable adder, Carry skip adder, Partial products

INTRODUCTION: Approximate computing has emerged as a potential solution for the design of energy-efficient digital systems [1]. Applications such as multimedia, recognition and data mining are inherently error-tolerant and do not require a perfect accuracy in computation. For these applications, approximate circuits may play an important role as a promising alternative for reducing area, power and delay in digital systems that can tolerate some loss of accuracy, thereby achieving better performance in energy efficiency. As one of the key components in arithmetic circuits, adders have been extensively studied for approximate implementation (see [1] for a review). New methodologies to model, analyze and evaluate the approximate adders have been discussed in [2]–[4]. However, there has been relatively less effort in the design of approximate multipliers. A multiplier usually consists of three stages: partial product generation, partial product accumulation and a carry propagation adder (CPA) at the final stage. [5] considers using approximate adders to generate the radix-8 Booth encoding 3x with error reduction. In [6], approximate partial products are computed using inaccurate 2 × 2 multiplier blocks, while accurate adders are used in an adder tree to accumulate the approximate partial products. [2] briefly discusses the use of approximate speculative adders for the final stage addition in a multiplier. The error tolerant multiplier (ETM) of [7] is based on the truncation of a multiplier into an accurate multiplicApplications that have recently emerged (such as image recognition and synthesis, digital signal processing, which is computationally demanding, and wearable devices, which require battery power) have created challenges relative to power consumption. Addition is a fundamental arithmetic function for these applications [1] [2]. Most of these applications have an inherent tolerance for insignificant inaccuracies. By exploiting the inherent tolerance feature, approximate computing can be adopted for a tradeoff between accuracy and power. At present, this tradeoff plays a significant role in such application domains [3]. As computation quality requirements of an application may vary significantly at runtime, it is preferable to design quality Configurable systems that are able to tradeoff computation quality and computational effort according to application requirements [4] [5]. The previous proposals for configurability suffer the cost of the increase in power [5] or in delay [12]. In order to benefit such application, a low-power and high-speed adder for configurable approximation is strongly required. In this paper, we propose a configurable approximate adder, which consumes lesser power than [5] does with a comparable delay and area. In addition, the delay observed with the proposed adder is much smaller than that of [12] with a comparable power consumption.

Our primary contribution is that, to achieve accuracy configurability the proposed adder achieved the optimization of power and delay simultaneously and with no bias toward either. We implemented the proposed adder, the conventional carry look-ahead adder (CLA), and the ripple carry adder (RCA) in Verilog HDL using a 45-nm library. Then we evaluated the power consumption, critical path delay, and design area for each of these implementations. Compared with the conventional CLA, with 1.95% mean relative error distance (MRED), the proposed adder reduced power consumption and critical path delay by 42.7% and 56.9%, respectively.

We provided a crosswise comparison to demonstrate the superiority of the proposed adder. Moreover, we implemented two previously studied configurable adders to evaluate power consumption, critical path delay, design area, and accuracy. We also evaluated the quality of these accuracyconfigurable

Copyright @ 2020 **ijearst. All rights reserved.**

adders in a real image processing application. Many increasingly popular applications, such as image processing and recognition, are inherently tolerant of small inaccuracies.

LITERATURE SURVEY:

Gupta et al. [6] discussed how to simplify the complexity of a conventional mirror adder cell at the transistor level. Mahdiani et al. [7] proposed lowerpart-OR adder, which utilizes OR gates for addition of the lower bits and precise adders for addition of the upper bits. Venkatesan et al. [8] proposed to construct an equivalent untimed circuit that represents the behavior of an approximate circuit. The above static approximate designs [6-8] with fixed accuracy may fail to meet the quality requirements of applications or result in wastage of power when high quality is not required. Kahng et al. [4] proposed an accuracy-configurable adder (ACA), which is based on a pipeline structure. The correction scheme of the ACA proceeds from stage 1 to stage 4, if the most significant bits of the results are required to be correct, all the four stages should be performed. Motivated by the above, Ye et al. [5] proposed an accuracy gracefully-degrading adder (GDA) which allows the accurate and approximate sums of its sub adders to be selected at any time. Similar to [5], our adder proposed in this paper does not consider a pipeline structure either. To generate outputs with different levels of computation accuracy and to obtain the configurability of accuracy, some multiplexers and additional logic blocks are required in [5]. However, the additional logic blocks require more area. Furthermore, these blocks will cause power wastage when their outputs are not used to generate a sum. This problem was addressed by [12] based on a low-power configurable adder that generates an approximate sum by using OR gates. Similar to [12], the proposed adder also uses OR gates to generate an approximate sum, but [12] focuses on only power consumption and its delay is large. Thus, it may fail to meet the speed requirement of an application. Approximate Adders for approximate multiplication Document The gap between capabilities of CMOS technology scaling and requirements of future application workloads is increasing rapidly. There are several promising design approaches that jointly can reduce this gap significantly. Approximate computing is one of them and in recent years, has attracted the strongest attention of the scientific community. Approximate computing exploits inherent errorresilience of applications and features high-performance energy-efficient software and hardware implementations by trading-off computational quality (e.g., accuracy) for computational efforts (e.g., performance and energy). Over the decade, several research efforts have explored approximate computing throughout all the layers of computing stack, however, most of the work at hardware level of abstraction has been proposed on adders. In [13], a comparative survey of state- of-the-art approximate adders is provided.

EXISTING TECHNIQUE:

Accuracy-controllable multiplier using carry maskable adder:

A typical multiplier consists of three parts:

1.partial product generation using an AND gate;

2. PPR using an adder tree; and

3.addition to produce the final result using a CPA.

Power consumption and circuit complexity are dominated by the PPR [6], and the multiplier's critical path is dominated by the propagated carry chain in the CPA [7].

Approximate Tree Compressor:

Figure 1(a) shows an accurate half adder, for which the following equation can be obtained:

$$\{c, s\} = a + b = 2c + s = (c + s) + c,$$

where {,} and + denote concatenation and addition, respectively. The value c is generated by a ___ b and s is generated by a ___ b, so (_ _ _) can be generated by a __ b. Based on the above, consider the basic logic cell shown in Fig. 1(b), for which the following equations can be obtained:

$$p = c + s,$$
$$q = c,$$
$$\{c, s\} = a + b = p + q.$$

This is called an incomplete adder cell (iCAC). Table I shows the truth tables for an accurate half adder and an iCAC. Note that the bit position of c and that of s, p, and q are different. As can be seen, q is equal to c. While p is not equal to s, the precise sum can be obtained by adding p and q, so the iCAC is not an approximate adder but an element of a precise adder. By extending the above equation to _ bits, the following equation can be obtained:

$$S = A + B = P + Q.$$

where A, B, P, and Q are _-bit values, the bits of which correspond to a, b, p, and q, respectively. A row of eight iCACs, used for 8-bit inputs, is shown in Fig. 2. Consider the example of an 8-bit adder with the two inputs A = 01011111 and B = 00110110. The accurate sum S is 10010101, while the row of iCACs produces P = 01111111 and Q = 00010110. Again, it is evident that the following holds:

S=P+Q; 1

While S is obtained from P and Q, P can be used as an approximation for S, and Q can be used as an error recovery vector for the approximate sum P.

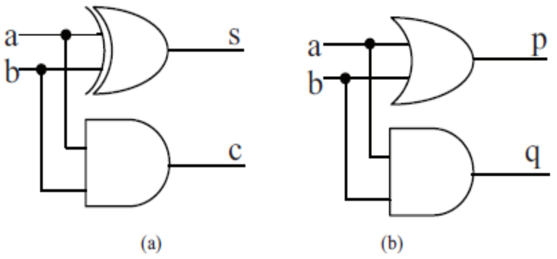


Fig. 1. (a) Accurate half adder and (b) incomplete adder cell.

TABLE I. TRUTH TABLES FOR ACCURATE HALF ADDER AND INCOMPLETE ADDER CELL

Inputs		Outputs			
		Accurate half adder		iCAC	
a	b	c	s	q	p
0	0	0	0	0	0
0	1	0	1	0	1
1	0	0	1	0	1
1	1	1	0	1	1

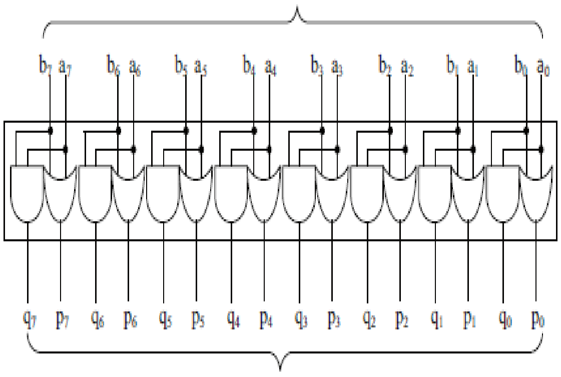


Fig. 2. A row of incomplete adder cells with two 8-bit inputs.

By extending the row of iCACs from two to inputs, /2 Ps and /2 Qs are obtained. If the sum of the /2 Qs is use instead of the /2 Qs themselves, the number of Qs is reduced to one. Remember that P is always greater than or equal to S, and Q is equal to C. By exploiting these facts, OR gates can be used to generate the approximate sum of the /2 Qs without significant loss of accuracy. This approximate sum is called the accuracy compensation vector and is denoted by V. This method is named approximate tree compressor (ATC). An ATC with inputs is called an ATC- , and the structure of an ATC with eight inputs (ATC-8) is shown in Fig. 3. The rectangles represent rows of iCACs and the number of iCACs in each row (rectangle) is dependent on the bit width of the inputs. For example, if there are eight _-bit inputs (D1, D2, ..., D8), four rows of _ iCACs are required to build a _-bit ATC-8. This reconstruction generates four approximate sums, P1, P2, P3, and P4, and four error recovery vectors, Q1, Q2, Q3, and Q4. OR gates generate the accuracy compensation vector V. As a result, the eight inputs have been reduced to five.

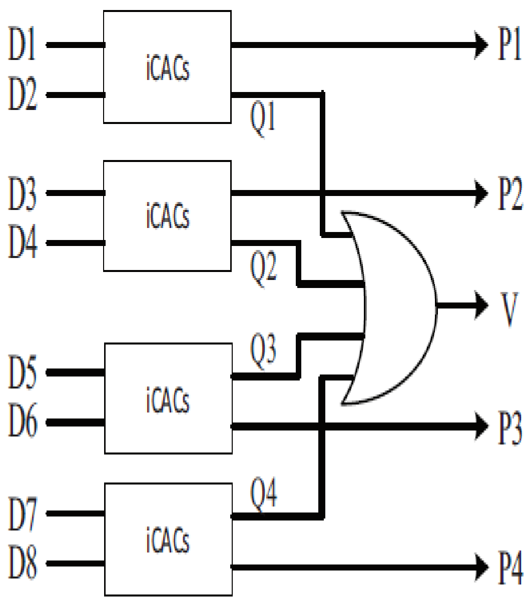


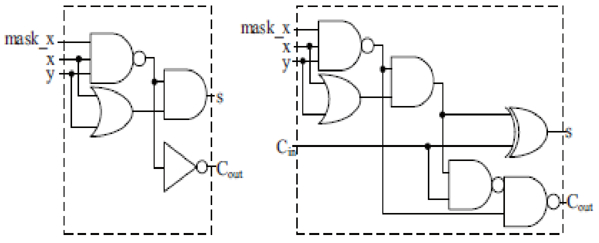
Fig. 3. Structure of an approximate tree compressor with eight inputs.

Copyright @ 2020 ijearst. All rights reserved.

OVER ALL ARCHITECTURE:

An -bit multiplier consists of " rows, each of which has partial products (PP), so there are _ PPs in total. Using the ATC- introduced in the previous section, the rows can be replaced by /2_# rows. Figure 5 shows an example of an 8-bit multiplier with $ _ $ PPs. The PPR is performed in three stages (Stage 1, Stage 2, and Stage 3) and the CPA is performed in Stage 4. The PP generation step is not shown. Each dot represents a PP. The least significant bit (right side) is bit 0, and the most significant bit (left side) is bit 14. The solid rectangles in Stage 1 represent ATCs and the dashed rectangles

represent rows of seven iCACs. Every row of iCACs includes PPs that are not processed: for example, the PP at position 0 in the first row and the one at position 8 in the second row of the first Icac block in ATC-8 are not processed. In Stage 1, eight rows of PPs are reduced to four rows (P1, P2, P3, and P4) and one accuracy compensation vector (V1) by an ATC-8. The four rows are further reduced to two rows (P5 and P6) and another accuracy compensation vector (V2) by an ATC-4. A final row of iCACs then processes P5 and P6 and generates P7 and Q7. In summary, Stage 1 uses an ATC-8, an ATC-4, and a row of seven iCACs to compress the $ _ $ PPs to four rows (P7, V1, V2, and Q7).



(a) (b)

Fig. 4. (a) Carry-maskable half adder, (b) Carry-maskable full adder

PROPOSED TECHNIQUE:

Controllable approximate multiplier using carry-skip adder

A carry-skip adder (also known as a carry-bypass adder) is an adder implementation that improves on the delay of a ripple-carry adder with little effort compared to other adders. The improvement of the worst-case delay is achieved by using several carry-skip adders to form a block-carry-skip adder.

The conventional structure of the CSKA consists of stages containing chain of full adders (FAs) (RCA block) and 2:1 multiplexer (carry skip logic). The RCA blocks are connected to each other through 2:1 multiplexers, which can be placed into one or more level structures [19]. The CSKA configuration (i.e., the number of the FAs per stage) has a great impact on the speed of this type of adder [23]. Many methods have been suggested for finding the optimum number of the FAs [18]–[26]. The techniques presented in [19]–[24] make use of VSSs to minimize the delay of adders based on a singlelevel carry skip logic. In [25], some methods to increase the speed of the multilevel CSKAs are proposed. The techniques, however, cause area and power increase considerably and less regular layout. The design of a static CMOS CSKA where the stages of the CSKA have a variable sizes was suggested in [18]. In addition, to lower the propagation delay of the adder, in each stage, the carry look-ahead logics were utilized. Again, it had a complex layout as well as large power consumption and area usage. In addition, the design approach, which was presented only for the 32-bit adder, was not general to be applied for structures with different bits lengths. Alioto and Palumbo [19] propose a simple strategy for the design of a single-level CSKA. The method is based on the VSS technique where the near-optimal numbers of the FAs are determined based on the skip time (delay of the multiplexer), and the ripple time (the time required by a carry to ripple through a FA). The goal of this method is to decrease the critical path delay by considering a noninteger ratio of the skip time to the ripple time on contrary to most of the previous works, which considered an integer ratio [17], [20]. In all of the works reviewed so far, the focus was on the speed, while the power consumption and area usage of the CSKAs were

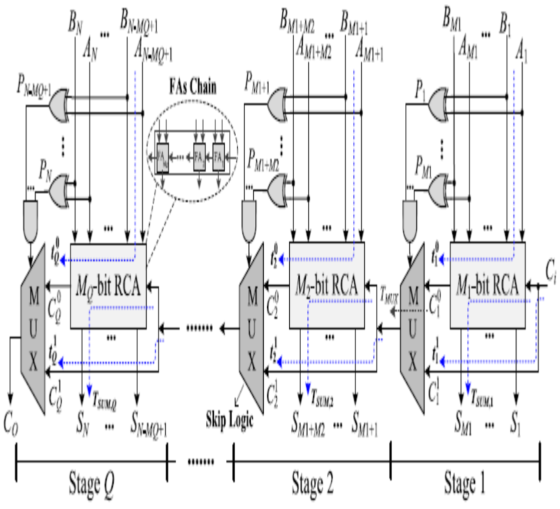
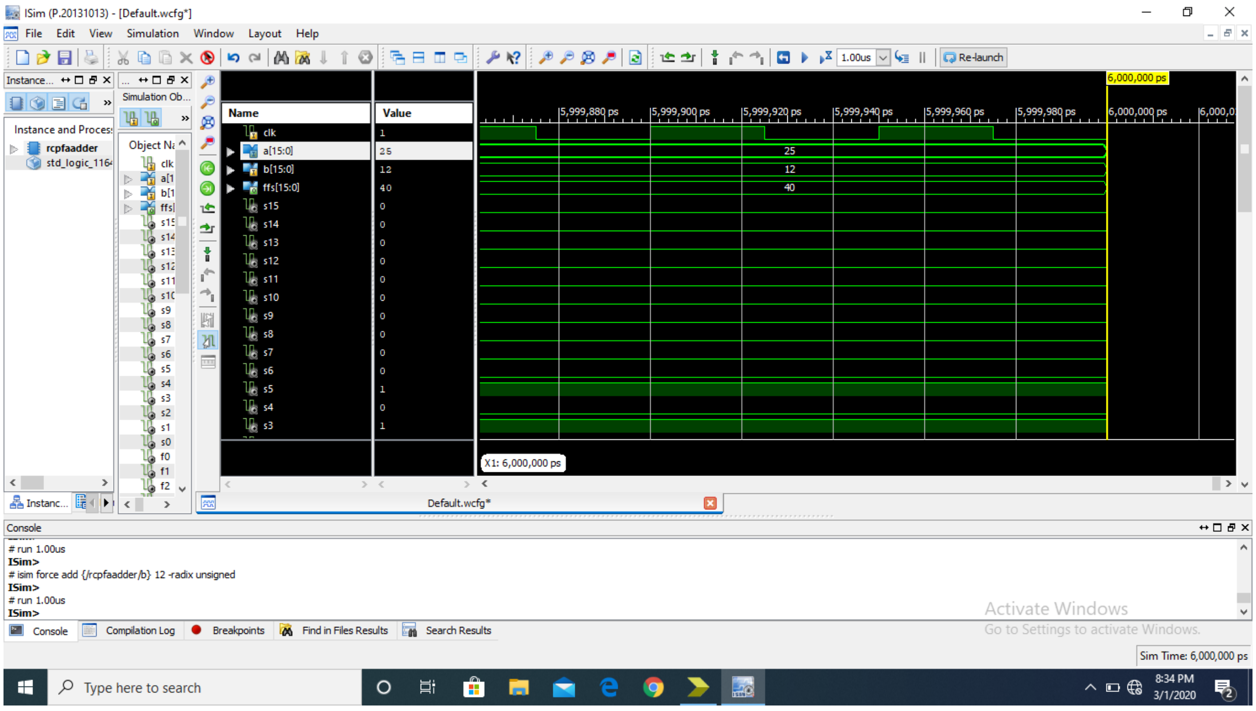


Fig5. Conventional structure of the CSKA

Copyright @ 2020 ijearst. All rights reserved.

not considered. Even for the speed, the delay of skip logics, which are based on multiplexers and form a large part of the adder critical path delay [9], has not been reduced.

RESULT:



Name	Value
clk	1
a[15:0]	25
b[15:0]	12
ffs[15:0]	40
s15	0
s14	0
s13	0
s12	0
s11	0
s10	0
s9	0
s8	0
s7	0
s6	0
s5	1
s4	0
s3	1

```
# run 1.00us
ISim>
# isim force add {/rcpfaadder/b} 12 -radix unsigned
ISim>
# run 1.00us
ISim>
```

CONCLUSION: In this paper, a novel approximate multiplier design is proposed using a newly designed approximate adder. On a statistical basis the proposed multiplier has a very small error distance and thus a high accuracy. Simulations have shown that the proposed design has a shorter critical path delay and a significantly lower area consumption compared to an existing design multiplier. It also uses a configurable error recovery that can produce more accurate results than other state-of-the art approximate multipliers.

REFERENCES:

[1] S. Cotofana, C. Lageweg, and S. Vassiliadis, “Addition related arithmetic operations via controlled transport of charge”, IEEE Transactions on Computers, vol. 54, no. 3, pp. 243-256, Mar. 2005.

[2] V. Beiu, S. Aunet, J. Nyathi, R. R. Rydberg, and W. Ibrahim, “Serial Addition: Locally Connected Architectures”, IEEE Transactions on Circuits and Systems-I: Regular papers, vol. 54, no. 11, pp. 2564-2579, Nov. 2007.

[3] S. Venkataramani, V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, “Quality programmable vector processors for approximate computing”, 46th Annual IEEE/ACM International Symposium on Micro architecture (MICRO), pp. 1- 12, Dec. 2013.

[4] A. B. Kahng, and S. Kang, “Accuracyconfigurable adder for approximate arithmetic designs”, IEEE/ACM Design Automation Conference (DAC), pp. 820-825, Jun. 2010.

[5] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, “On Reconfiguration- Oriented Approximate Adder Design and Its Application”, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 48- 54, Nov. 2013.

6] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, “Low-Power digital signal processing using approximate adders”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 1, pp. 124-137, Jan. 2013.

7] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bio-Inspired imprecise computational blocks for efficient VLSI implementation of Soft- Computing applications”, IEEE Transactions on Circuits and Systems I: Regular papers, vol. 57, no. 4, pp. 850-862, Apr. 2010.

[8] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, “MACACO: modeling and analysis of circuits for approximate computing”, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 667-673, Nov. 2011.

[9] NanGate, Inc. NanGate FreePDK45 Open Cell Library, http://www.nangate.com/?page_id=2325, 2008

10] J. Liang, J. Han, and F. Lombardi, “New metrics for the reliability of approximate and probabilistic adders”, IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760-1771, Sep. 2013.

11] M. S. Lau, K. V. Ling, and Y. C. Chu, “Energy- Aware probabilistic multiplier: Design and Analysis”, International Conference on Compliers, architecture, and synthesis for embedded systems, pp. 281-290, Oct. 2009.

[12] T. Yang, T. Ukezono, and T. Sato, “A Low- Power Configurable Adder for Approximate Applications”, 19th International Symposium on Quality Electronic Design, Mar. 2018. 978-

Copyright @ 2020 **ijearst. All rights reserved.**