A Carry-Predicting Full Adder for Accuracy-Scalable Computing

Hiroyuki Baba¹ Tongxin Yang² Masahiro Inoue³
Kaori Tajima⁴ Tomoaki Ukezono⁵ Toshinori Sato⁶
¹Graduate School of Electronics and Computer Science
²Graduate School of Information and Control Systems
³,⁴,⁵,⁶Department of Electronics Engineering and Computer Science
Fukuoka University
¹td172009, ²td166502, ³td141319, ⁴td141216}@cis.fukuoka-u.ac.jp
⁵tukezo@fukuoka-u.ac.jp, ⁶toshinori.sato@computer.org

Abstract - Approximate computing is one of the promising paradigms to enable high speed, small area, and lower power, which are essential properties for modern applications such as IoT devices. This paper proposes an approximate adder, which has scalabilities in accuracy and in power consumption. It is named Carry-Predicting Adder (CPredA). The CPredA is based on a previously proposed adder and improves its accuracy by conducting carry prediction. From the gate level simulations, it is found that the CPredA improves the accuracy over the existing approximate adder by around 50% with the comparable energy efficiency.

I. Introduction

Some of the contemporary applications, such as image processing and machine learning, have the ability of error tolerance. In other words, such computing does not require strictly accurate results. Approximate computing is a promising approach, which improves performance, circuit area, and power efficiency on the computing of the error tolerance applications [1]. While it can be utilized at any layer of computing stack, this paper focuses on elemental circuit level and proposes an accuracy-scalable adder.

Several approximate adders have been proposed [1-5]. Unfortunately, the range of applicable applications is limited, because most of the prior work lacks accuracy-scalability [1]. Dynamic configurability is necessary due to the following two reasons. First, different applications have different accuracy requirements, as do different program phases in an application. If the accuracy is fixed, chances of power reduction will be lost when high accuracy is not required. Second, for applications which rely on iterative methods, their accuracy requirement changes among iterations [6, 7]. While there are some adders which are statically [2] and dynamically [3-5] configurable, some applications might require further scalability in accuracy. In this paper, a low-power and small-area adder is studied for accuracy-scalable approximate computing.

The rest of this paper is organized as follows. The next section introduces the related work. Section III proposes the accuracy-scalable adder. After that, Section IV evaluates the proposed adder. Finally, Section V concludes this paper.

II. Related Work

Mahdiani et al. [2] proposed Lower-part-OR adder (LOA), which hybridizes precise adders for the upper bits and OR gates for the lower bits. This is because the lower bits are less important in accuracy than the upper ones. Although it is very simple, it is moderately accurate and is power- and area-efficient [1]. The LOA does not have dynamic configurability. In other words, its configuration is statically determined at the design time.

Ye et al. [3] proposed Gracefully-Degrading Adder (GDA), which has dynamic configurability. A 16-bit GDA is shown in Fig. 1. It consists of four 4-bit approximate sub-adders, each of which multiplexes a carry predictor and a precise adder. On configuration, one of the predictor and the adder is selected. In the figure, the configuration bits, {F₂, F₁, F₀}, are used for the selection. If F equals 1, the predictor is selected. In this paper, it is named prediction mode. The weaknesses of the GDA are its large area and power consumption. Because it has the predictor and the adder redundantly, its area obviously becomes large. In addition, because both of them are always active, it wastes relatively large power.
Yang et al. [4, 5] proposed Carry-Maskable Adder (CMA), which also has dynamic configurability. Different from the GDA, it relies on neither redundant circuits nor multiplexers but combines a precise adder and an approximate one into a simple structure. Hence its power consumption is comparable to a ripple carry adder (RCA) even when it selects the configuration of the highest accuracy.

The references to the other studies on approximate adders are summarized in [1].

III. Proposed Approximate Adder

This section explains the proposed approximate adder, which is named Carry-Predicting Adder (CPredA). It is inspired from the CMA [4, 5]. Hence, first in this section, the CMA is described. Next, the CPredA is introduced by explaining differences between the two adders.

A. Carry-Maskable Adder

Figure 2 shows a Carry-Maskable Full Adder (CMFA) [4, 5]. It works as the conventional full adder (FA), when the signal ‘mask’ equals 0. Otherwise, Cout becomes 0 and it works as an OR gate, which operates between x and y. In this paper, it is called mask mode. Note that Cin as well as Cout is assumed to be 0 in the mask mode.

Using the CMFA, a 16-bit CMA is constructed. In order to ease configuration, four CMFAs are grouped as a 4-bit CMA. Four FAs also group a 4-bit precise adder. As shown in Fig. 3, the 16-bit CMA is made from three 4-bit CMAs and one 4-bit precise adder. The most significant four bits are almost accurately calculated and the remaining 12 bits are approximated. F2, F1, and F0 are configuration bits, which are connected into ‘mask’ ports in the CMFAs. A single F configures all CMFAs in a 4-bit sub-adder. For example, when the set of \{F2, F1, F0\} is \{0,0,1\}, the lower four bits work as the CMFAs and the remaining 12 bits work as FAs.

B. Carry-Predicting Adder

In order to enhance the accuracy-scalability, the CMA is modified and the CPredA is proposed. Figure 4 introduces a Carry-Predicting Full Adder (CPFA). The ‘mask’ bit of the CMFA is replaced by ‘predict’ bit. When it is set to 0, the CPFA works as a FA. Otherwise, Cout is predicted by AND’ing x and y. Different from the CMFA, the sum ‘s’ is accurately generated. This operation is named prediction mode.

Table I is the truth table of the precise full adder, the mask mode of the CMFA, and the prediction mode of the CPFA. Remember that Cin is always 0 in the mask mode of the CMFA. It is emphasized where the approximate adders generate incorrect outputs. It is expected that the CPFA is more accurate than the CMFA from the following investigations. First, the CPFA always generates correct outputs when Cin is 0. In contrast, the CMFA may sometimes generate incorrect Cout. Second, the CPMA works even when Cin is 1. On the other hand, the CMFA assumes that Cin is always 0.

Similar to the 16-bit CMA, a 16-bit CPredA is constructed. The grouping of four bits is applied to the 16-bit CPredA, which consists of four sub-adders. The most significant four bits are processed by a 4-bit precise adder and remaining 12

---

**Fig. 2.** Carry-maskable full adder.

**Fig. 3.** 16-bit CMA.

**Fig. 4.** Carry-predicting full adder.

---

**TABLE I**

<table>
<thead>
<tr>
<th>Truth Table of FA, CMFA, and CPFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>cin</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
the remaining 12 bits work as the precise FAs in the 16-bit CPredA and the 16-bit CMA. Note that the configuration of \{0,0,0\} means the precise adder in all approximate adders.

First, they are evaluated on accuracy. Mean relative error distance (MRED) [8] is used as the metric for evaluating accuracy. The error distance (ED) is defined as the difference between an accurate sum (M) and its approximate sum (M’), i.e., ED = |M’ – M|. The relative ED (RED) is defined as the ED divided by M, i.e., RED = ED / M = |M’ - M| / M. The mean RED (MRED) is the average of REDs.

Second, they are evaluated on delay, circuit area, and power consumption. Verilog HDL is used to implement the adders. The precise adders, which are RCA and carry lookahead adder (CLA), are also implemented for the purpose of comparison. Synopsys Design Compiler and NanGate 45nm Open Cell Library [9] are used for logic synthesis. The default compiler options are used. The value change dump (VCD) files from VCS simulations are used by Synopsys Power Compiler for power estimation. Using one million outputs obtained from randomly generated inputs, the MRED is calculated and the VCD file is generated.

B. Accuracy

Table II summarizes the MRED for the approximate adders. It is firstly found that all approximate adders have large scalabilities in the accuracy. The configuration of \{0,0,1\} is two orders of magnitude more accurate than that of \{1,1,1\}. When the adders are compared with each other, the CPredA type 1 is comparably accurate with the CMA. In contrast, the CPredA type 2 improves the MRED over the CMA by around 50%. From the observations, the carry propagated from the approximate part to the precise part is most important. Different from the expectation explained in the previous section, the other carries predicted inside the 4-bit sub-adder do not contribute to the accuracy. Hereafter, the CPredA type 2 is chosen as the representative of the CPredAs.

C. Delay and Circuit Area

Figure 6 shows delays and circuit areas of the approximate and the precise adders. For each group of two bars, the left one presents the delay and the right one indicates the area. All values are normalized by those of the RCA and hence the results of the RCA are not included in the figure. Regarding delay, the GDA has the comparable one to that of the CLA and it is 45% smaller than those of the CMA and the CPredA. In contrast, both the CMA and the CPredA has nearly equivalent delay to that of the RCA. While the both of two
are based on the RCA, their overhead due to the configurability is very small and is less than 5%.

Regarding circuit area, the GDA suffers serious overhead from the configurability. The area of the GDA is 122% and 59% larger than those of the RCA and the CLA, respectively. The increase in area is not ignorable, when approximate adders are required from area-constrained applications such as IoT devices. In contrast, the CMA and the CPredA has small area, both of which are less than 10% larger than that of the RCA.

As the configuration changes, the longest path delay also changes. This is because some paths in a configuration are always inactive in the other configurations. Such paths should be regarded as false paths when the longest path delay is evaluated. Figure 7 explains how the delay changes dynamically according to the configuration. As in Fig. 6, all values are normalized by the delay of the RCA. It is easily found that the scalability on delay is larger in the CPredA and in the CMA than in the GDA. Surprisingly, in the configuration of {1,1,1}, both the CPredA and the CMA are faster than the GDA. From the observations, the scalability in performance can be built. Dividing the CPredA into three stages makes a 3-stage pipelined CPredA. If the pipeline stage unification technique [10] is applied to it, the number of the pipeline stages changes dynamically according to the configuration. The configurations of {0,0,0} and {0,0,1} work as the 3-stage pipeline. In contrast, those of {1,1,1} and {1,1,1} work as the unified 2- and 1-stage pipelines, respectively. This makes the scalability in the execution latency of the CPredA. In other words, the performance-scalability is obtained. It should be carefully considered how to divide the CPredA and where to insert pipeline registers. This open question remains as the future study.

When the CPredA is compared with the CMA, their delays and areas are almost the same, respectively.

**D. Power Consumption**

Figure 8 presents how the configuration affects power consumed by the approximate adders. Those of the RCA and the CLA are also shown as straight lines because they are not configurable. It is surprisingly found that the GDA consumes much larger power than the other adders do. Its large circuit area and high activities in its components will cause the large power. This is because the precise sub-adders and the carry predictors always work simultaneously. It is also observed that the power-scalability is larger in the CPredA and in the CMA than in the GDA. From these observations, the GDA is omitted from the comparison hereafter.

Power and accuracy in MRED for every configuration are summarized in Fig. 9. Note the horizontal axe uses
logarithmic scale. It clearly shows the power-scalability is larger in the CMA than in the CPredA. On the other hand, the accuracy-scalability is larger in the CPredA than in the CMA. The MREDs in \{1,0,0\} are 301.1 and 247.0 times smaller than those in \{0,0,0\} for the CPredA and the CMA, respectively. The relationship between PDP (power delay product, which equals energy) and MRED is presented in Fig. 10. As can be seen, the difference between two adders becomes small. From the viewpoint of energy efficiency, the CPredA and the CMA are comparable.

From the results above, it is summarized that application can choose appropriately from the CPredA and the CMA according to their requirements. Applications that prioritize power should use the CMA. On the other hand, those that prioritize accuracy should use the CPredA because it has larger accuracy-scalability than the CMA does.

V. Conclusions

This paper proposed the CPredA, which approximates addition by predicting carry generation. Compared with the CMA, the CPredA achieves higher accuracy with comparable delay and area and with slightly larger power. In the view point of energy efficiency, the CPredA and the CMA are comparable to each other. The CPredA is another promising approximate adder as an alternative to the CMA.

One of the future study is to achieve performance-scalability. The pipelined CPredA will be a candidate and it should be carefully investigated to divide the CPredA into the pipeline stages.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP17K00088, by funds (No.175007 and 177005) from the Central Research Institute of Fukuoka University, and by VLSI Design Institute of Fukuoka University, and by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

References