Approximate De-randomizer for Stochastic Circuits

Kyounghoon Kim[1], Jongeun Lee[2], and Kiyoung Choi[1]

[1]Dept. of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
{khkim, kchoi}@dal.snu.ac.kr
[2] School of Electrical and Computer Engineering, UNIST, Ulsan, Korea
jlee@unist.ac.kr

Abstract—**De-randomizer is one of the most important components in stochastic computing. We suggest an approximate parallel counter for the de-randomizer generating a small number of errors, which outperforms a conventional parallel counter in terms of area, delay, and power.**

Keywords-approximate counter; stochastic computing; de-randomizer.

I. Introduction

Stochastic computing (SC) is an alternative paradigm to conventional binary arithmetic computing [1]. SC can boost efficiency in terms of area, power, and error tolerance while relaxing the accuracy of computation in emerging applications such as machine learning, computer vision, and computer graphics. Numbers in SC are represented by the probability of 1's occurrence in a random bit stream. For example, in Fig. 1 (a), since the occurrences of 1's in the three 8-bit streams at *A*, *B*, and *Y* are 6, 4, and 3, respectively, the corresponding *stochastic numbers* (SNs) are $P(A=1)=6/8$, $P(B=1)=4/8$, and $P(C=1)=3/8$. The single AND gate performs multiplication (i.e., 6/8 x 4/8 = 3/8).

For an efficient conversion between *binary numbers* (BNs) and SNs, there have been researches on converting from BNs to SNs (aka randomization) and vice versa (aka de-randomization). In this paper, we focus on the conversion from SNs to BNs. Converting an SN into a BN requires counting the number of 1s in the random bit stream. Fig. 1 (b) and (c) show a serial counter and an *accumulative parallel counter* (APC) [2] [3], respectively, where flip-flops are used with some other logic for accumulation. [1] The *parallel counter* (PC) in an APC uses full adders (FA) in order to generate $2^0{\sim}2^{m-1}$ weighted bits (i.e., a BN) from a stream of 2^0 weighted input bits (i.e., an SN), where *m* is the number of bits of the BN. Fig. 1 (d) shows an example of PC generating 4 bits BN from 15 bits SN.

Considering that SNs are based on the probabilistic nature of random bit streams and thus the accuracy is already compromised during SC, there is no reason to stick to accurate conversion using the conventional PC. Note that using the conventional PC may lead to lots of inefficiency in the aspect of entire workload, since the length of bit streams in SN is much longer than that in BN; only *k* bits in BN becomes 2^k bits in SN.

In this paper, we suggest an approximate PC exploiting two properties of SN: 1) inaccuracy due to randomness and 2) long bit stream. The proposed PC can be implemented with a smaller circuit compared with the conventional accurate PC, where the inaccuracy problem can be alleviated due to the long bit stream.

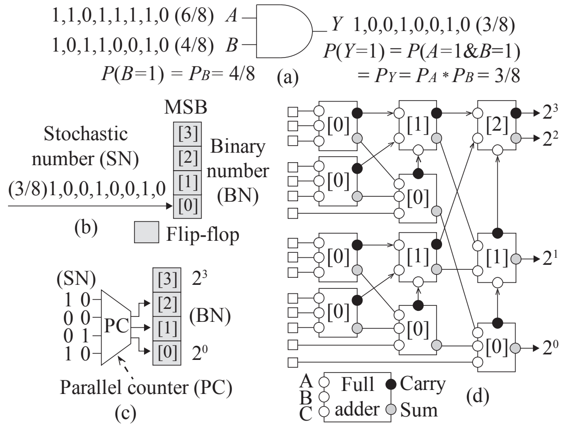


Fig. 1. Stochastic numbers (SNs) and conventional counters, where the numbers in brackets represent the bit index of binary numbers (BNs). (a) Multiplication of two SNs. (b) Accumulative serial counter. (c) Accumulative parallel counter (APC). (d) Example of a parallel counter (PC) converting 15 bits SN into 4 bits BN.

II. Proposed Approximate Parallel Counter

The proposed approximate PC is shown in Fig. 2 (a), which consists of two parts: an approximation unit (AU) and a conventional accurate PC. The former is implemented by simple gates such as AND and/or OR gate; Fig. 2 (b) shows a 2-layer AU. The approximate PC exploiting a 1-layer AU is shown in (c). The input weight of AU is 2^0 while the output weight becomes 2^l, where *l* is the number of layers.[2] (e) shows errors for 1-layer AU using AND or OR gate. Note that AND gates generate negative errors while OR gates generate positive errors.

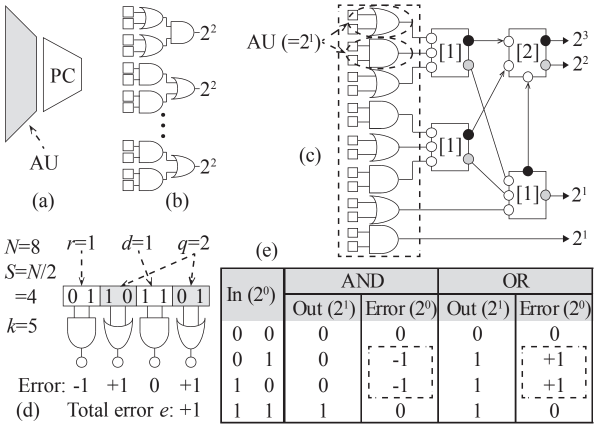
A. Analysis for Gate Count in 1-layer Approximate PC

In order to see how much area reduction can be obtained by the proposed approach, we calculate the number of FAs in Fig. 1 (d) and Fig. 2 (c). Suppose that *N*-input bits become *v*-output binary bits ($N=2^v-1$) and $f(v)$ is the number of FAs. In case of the conventional PC (i.e., Fig. 1 (d)), when *v* increases by one, the number of FAs increases by two times plus $v-1$. Thus, $f(v) = 2\cdot f(v-1)+v-1 = 2^v-v-1$, where $f(2)=1$ (i.e., one FA is needed to generate two-digit binary numbers). The number of gates in the conventional PC is given by

$$G_{conv}(N) = \{(N+1) - \log_2^{N+1} - 1\} \cdot 5, \tag{1}$$

[1] In this paper, we consider only APC because the serial counter consumes large amounts of energy.

[2] Due to space limitation, we explain only 1-layer AU in this paper.



In (2^0)	AND		OR	
	Out (2^1)	Error (2^0)	Out (2^1)	Error (2^0)
0 0	0	0	0	0
0 1	0	-1	1	+1
1 0	0	-1	1	+1
1 1	1	0	1	0

Fig. 2. The proposed parallel counter (PC). (a) Overview of the PC. (b) 2-layer approximate unit (AU). (c) The proposed PC using 1-layer AU, converting 16-bit SN into 4-bit BN. (d) Example of 1s distribution in 1-layer AU. (e) Output and error for all inputs in AND and OR gate.

considering that $v = \log_2^{N+1}$ and FA consists of five gates. The proposed PC with 1-layer AU, as shown in Fig. 2 (c), uses $f(v-1)$ FAs for v-output bits and $N/2$ additional AND or OR gates, where $N=2^v$.[3] Thus, the number of gates in the proposed PC is,

$$G_{prop\text{-}1}(N) = \{N/2 - \log_2^{N}\} \cdot 5 + N/2, \tag{2}$$

Fig. 3 (a) shows the number of gates for (1) and (2), where the proposed PC using 1-layer AU reduces gate count by about 40% compared to the conventional PC.

B. Analysis for Error in 1-layer Approximate PC

In order to analyze the effect of approximation and error, we define e as the output error in number of 1s and $T1(N,k,e)$ as the probability mass function (PMF), where N is the length of the given bit stream and k is the number of 1s in the input. In other words, given k 1s in the N-bit stream, $T1(\cdot)$ shows the probability of error e generated by the proposed approximate PC in Fig. 2 (c). The PMF is given by

$$T1(N,k,e) = \sum_{e=q-r} \frac{\binom{S/2}{q}\binom{S/2}{r}\binom{S-q-r}{d} \cdot 2^q 2^r}{\binom{N}{k}}, \tag{3}$$

where S is the number of output bits (slots) of the AU (S=N/2), and d is the number of slots containing two 1s at the inputs (d=(k−e)/2), and q is the positive error of OR gates while r is the negative error of AND gates. Fig. 2 (d) shows five 1s in the 8-bit input stream. Note that q+r is the number of errors and is the same as the number of slots containing one 1-bit and that the cumulative error e is q−r.

Fig. 3 (b) shows a theoretical result for the error PMF of $T1(.)$, given 1024-bit streams. The mean of error values of $T1(.)$ is zero and the maximum standard deviation of the errors is only 11.3 (about 1.1%). This means that there is no bias, and that the error is within only 1.1% for about 70% among all trials (within 2.2% for 95%).

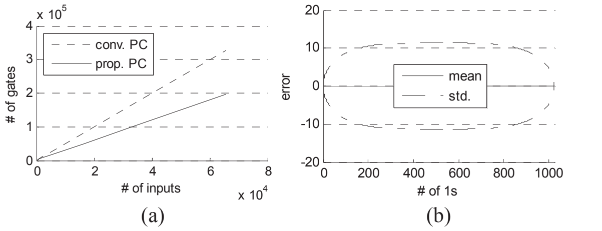
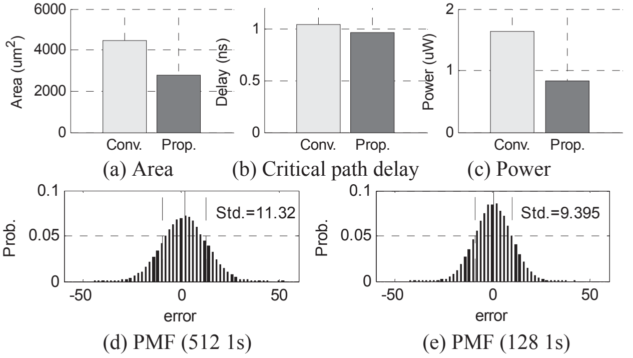


Fig. 3. Theoretical analysis of the proposed scheme. (a) The number of gates for the conventional PC and the proposed PC with 1-layer AU. (b) The mean and standard deviation of number of errors for 1-layer PC with 1024 input bits.



(a) Area (b) Critical path delay (c) Power

(d) PMF (512 1s) (e) PMF (128 1s)

Fig. 4. Experimental results of the proposed approximate PC compared with the conventional PC in 1024-bit stream. (a) Area. (b) Critical path delay. (c) Power. (d) PMF when 512 1s among 1024 bits. (e) PMF when 128 1s.

III. Experimental Results

We compare the conventional PC and the proposed PC with heterogeneous 1-layer AU, given 1024-bit streams. They are implemented in TSMC 45nm technology library with Synopsys Design Compiler using Verilog HDL. As shown in Fig. 4, the proposed PC decreases area ((a)), critical path delay ((b)), and power ((c)) by 38.3%, 7.6%, and 49.4%, respectively. (d) and (e) show the errors of 1024-bit streams containing 512 and 128 1s, where the means of errors are almost zero and the standard deviations are 11.32 and 9.40, respectively. The results match well with the theoretical analysis results.

IV. Conclusion

Although de-randomizer is a very important component for stochastic circuits, it has not been paid attention in the literature of stochastic computing (SC). Considering that SC is based on inaccurate computation, using a conventional accurate parallel counter (PC) in SC leads to inefficiency. We have proposed an approximate PC, which outperforms the conventional PC in terms of area, delay, and power, with no bias and small standard deviation of errors.

Acknowledgment

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1501-08.

References

[1] A. Alaghi and J. P. Hayes, "Survey of stochastic computing," ACM Trans. Embed. Comput. Syst., vol. 12, p. 92, 2013.

[2] B. Parhami and C.-H. Yeh, "Accumulative parallel counters," Proc. Asilomar Conf. Signals, Systems & Computers, pp. 966-970, 1995.

[3] T. Pai-Shun and J. P. Hayes, "Stochastic Logic Realization of Matrix Operations," Proc. Digital System Design, pp. 356-364, 2014.

[3] In this case, we use N=2^v instead of N=2^v−1, since it fits better with the use of two input gates. The generated BN cannot represent the maximum value of SN, but the error is small and thus can be ignored.