Automatic Partition for Hybrid Stochastic-Binary-Based Circuits

Zexi Li[1], Ruogu Ding[1], Haojia Sun[1], Donghao Chen[1], and Weikang Qian[1,2,]*
[1]University of Michigan-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai, China
[2] MoE Key Lab of AI, Shanghai Jiao Tong University, China
Email: {lzx12138, rg.ding, judy.sun, donghaochen, qianwk}@sjtu.edu.cn ; *corresponding author

ABSTRACT

Stochastic computing (SC) is an emerging computing paradigm with a smaller hardware cost than conventional binary computing (BC). In this paper, we study the effects of circuit partitions that determine which parts are implemented in the SC or BC domain on several applications. We further propose an automatic partition algorithm to find the best partition meeting the accuracy and area requirements. Experimental results demonstrate the trade-off among accuracy, area, and energy using different partitions and an average of 46.53% in energy reduction of the circuits generated by the proposed partition algorithm.

I. INTRODUCTION

Stochastic computing (SC), an emerging computing paradigm, has gained increasing attention in recent years due to its simple computing units and strong fault tolerance [1]. Compared to conventional binary computing (BC) where a number is encoded in binary radix form, SC encodes a number by the ratio of 1s in the bit stream. For example, the bit stream "0111" encodes 7 in BC and encodes 3/4 in SC, as there are three 1s in the 4-bit stream. An advantage of SC is that it has simple computing units. For example, an SC multiplier is implemented by a single AND gate. In comparison, a 2-bit BC multiplier consists of 5 AND gates and 2 XOR gates. However, SC suffers from two main disadvantages. First, due to random fluctuation, the computation in SC is not 100% accurate. Second, the conversion between BC and SC needs additional units, resulting in area and power overhead.

With the inherent inaccuracy, SC is mainly used in error-tolerant applications such as image processing [2] and neural networks (NN) [3]. In early days, researchers design entire NN circuits in SC [3]. However, research in [4] shows that implementing all computations in NN in SC causes a large accuracy degradation. To address this issue, the work [5] implements only the first layer of the NN in SC. Nevertheless, it manually determines which parts should be implemented in SC without thoroughly analyzing the impact of partition between BC and SC. Therefore, determining the optimal BC/SC partition of the circuit to maximize performance remains an open question and should be further studied.

In this paper, we study the effects of different SC/BC partitions and design an automatic partition algorithm. The main contributions of this work are as follows:

This work is supported by the National Key R&D Program of China under grant number 2020YFB2205501.

1) We implement different SC/BC partitions on several applications and compare their performance.
2) We propose an iterative algorithm to find the best SC/BC partition automatically.

Experimental results show that given the area and accuracy of a manually partitioned circuit as the threshold, our automatic partition algorithm can generate circuits with 46.53% smaller energy than the input circuits on average.

The rest of the paper is organized as follows. Section II describes the manual partitions and the automatic partition algorithm. Section III shows the experimental results. Section IV concludes the paper and discusses the future work.

II. METHODOLOGY

Manual SC/BC Partitions of Hybrid Circuits

Manually partitioned circuits of several applications are implemented, including Roberts edge detection, Gaussian blur (GB), matrix-vector product (MV), and neural network (NN). Taking GB as an example, we implement it with a 3×3 kernel size. The computation of GB is

$$z = \sum_{i=1}^{9} w_i \cdot x_i,$$

where z is the output pixel value, x_i is the flattened i-th input, and w_i is the corresponding weight. The circuit can be viewed as a two-stage computation, with the first stage having 9 multiplications and the second having 1 summation. For the manual partition, we specify that computations at the same stage are implemented either all in BC or all in SC, which is the widely used design choice. For multiplication, there is only one implementation in the SC domain. For addition, there are two implementations in the SC domain based on MUX and accumulative parallel counter (APC) [3]. Combining these design choices in the SC domain and the choice in the BC domain, there are 6 partition ways for the GB circuit. Similarly, there are 6 partition ways for Roberts and MV circuits. For NN, a three-layer multi-layer perceptron (MLP) is implemented for digit recognition on the MNIST dataset. It has 64, 16, and 10 nodes at the three layers, respectively, with tanh being the activation function, forming a MV-Tanh-MV-Tanh structure. The tanh function has BC and SC implementations, and we specify that the multiplications and additions in the MV are implemented in the same domain, resulting in $2^4 = 16$ partition ways.

These manual designs are simulated using the SC simulator *SCGen* [6] to get their accuracies and synthesized to get their areas and energies. The results are compared to study the effects of different partitions.

Automatic SC/BC Partition Algorithm

We further propose an SC/BC partition algorithm to find the best SC/BC partition way automatically, as shown in Algo. 1. It takes the circuit description as the input, as well as accuracy and area thresholds as they are common considerations in SC circuit designs. The algorithm starts from the pure BC implementation of the circuit and iteratively replaces BC components with their equivalent SC implementations. During each iteration, possible replacements are applied. The new circuits after replacement are simulated to get accuracies and estimated for areas. Circuits satisfying the accuracy and area requirements will be saved as candidate circuits, and the circuit with the highest accuracy will be chosen to enter the next iteration. When the iteration finishes, i.e., there are no BC components in the circuit, all candidate circuits are synthesized to get their actual areas and energy consumptions.

The automatic partition algorithm no longer needs to follow the rule of keeping the components at the same stage in the same domain, which enlarges the design space and makes it possible to find new designs on the Pareto front. However, for large circuits such as the neural network, the design space is large and it is infeasible to replace components one by one. Therefore, we implement an option to enable the replacement of all equivalent components at the same stage during one iteration, which results in a smaller design space and a smaller runtime.

Algorithm 1: Automatic Partition

Input : Circuit description, area and accuracy threshold $\texttt{Acc}_t$ and $\texttt{D}_t$.
Output: Generated circuit.

```
current = the pure BC implementation of the circuit;
candidates = [];
// Iterative replacement
while current is not pure SC do
    replaced = [];
    foreach BC component c in current do
        Replace c with its SC version;
        Adding necessary conversions between BC and SC;
        Perform circuit optimization;
        Adding the circuit to replaced;
    foreach circuit C in replaced do
        Simulate C to get its accuracy Acc;
        if Acc > Acc_t then
            Estimate the area A of C;
            if A < A_t then
                Add C to candidates;
    current = the circuit in replaced with the largest accuracy;
// Filter invalid candidates
foreach circuit C in candidates do
    Synthesis C to get its area A and energy;
    if A > A_t then
        Remove C from candidates;
return the circuit in candidates with the smallest energy;
```

III. EXPERIMENTAL RESULTS

Manual SC/BC Partition of Hybrid Circuits

We compare the performance of manually partitioned circuits for the above applications. Due to space limit, we only show results of the GB 3 × 3 and NN circuits. The bit width is set to 8, which corresponds to a bit stream length of 256, and the accuracy of NN is calculated by 1000 image inputs. Results of the GB 3 × 3 circuit are shown in Table I, where the circuit error is measured by the mean absolute error (MAE) and the circuit name indicates the partition. For example, *Pure_BC* indicates the circuit is purely implemented in BC, while *BC_SC_MUX* represents the circuit with BC multipliers and the MUX-based SC adder. There are several observations from the results. First, the *Pure_BC* design achieves the highest accuracy (the minimum error) and the smallest energy consumption, while having the largest area. Although it has the largest power, since the BC computation can be done in one clock cycle, its energy consumption is smaller than the others by orders of magnitude. Second, the APC-based adder outperforms the MUX-based adder in accuracy but has larger area and energy. Third, *Pure_SC_MUX* has the smallest area and the worst accuracy. In conclusion, the results of the GB 3 × 3 circuit show the trade-off between accuracy and area, and which partition to use should be determined by the requirements.

Results of the NN circuit are shown in Fig. 1, where the name of each design point represents the partition. For example, SBBS means that the MV operation in the first layer is in SC (“S”), the activation function in the first layer is in BC (“B”), the MV operation in the second layer is in BC (“B”), and the activation function in the second layer is in SC (“S”). From the experiment on GB circuits, we find that MUX-based adder causes large accuracy degradation, and therefore, we only consider the APC-based adder. The results show that 1) the BBBB partition has the largest area and high accuracy; 2) the SSSS partition has

TABLE I. PERFORMANCE COMPARISON OF MANUAL PARTITIONS FOR THE GB 3 × 3 CIRCUIT

Circuit Name	Error (*MAE*)	Area (μm^2)	Power (*W*)	Energy (*J*)
Pure_BC	0.001787	3534.342	1.70E-4	3.40E-12
BC_SC_MUX	0.098812	3171.518	8.81E-5	4.51E-10
BC_SC_APC	0.049162	3213.812	8.86E-5	4.54E-10
SC_BC	0.030113	1543.864	1.15E-4	5.89E-10
Pure_SC_MUX	0.120616	736.820	7.25E-5	3.71E-10
Pure_SC_APC	0.016938	778.050	7.40E-5	3.79E-10

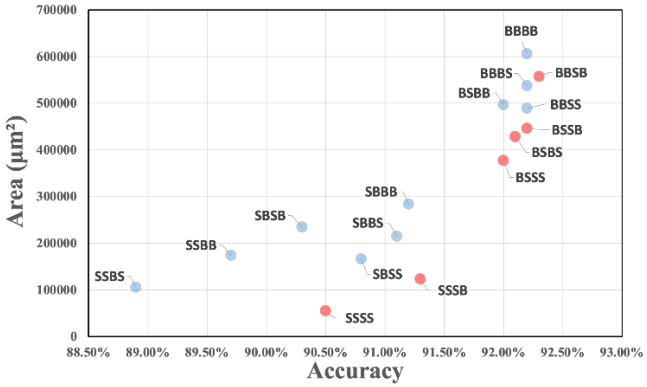

Figure 2: Accuracy and area comparison of manual partitions for the NN circuit, where partitions on the Pareto front are marked in red.

the smallest area and relatively low accuracy. Additionally, it is observed that all partitions with the first layer's MV in BC (names starting with "B") have larger accuracies than partitions with the first layer's MV in SC (names starting with "S"), which may be explained by that the first layer's MV is of great importance and small errors in it will result in a large accuracy drop.

Automatic SC/BC Partition Algorithm

To evaluate the performance of the proposed automatic partition algorithm, for each manual partition of the applications, we take its accuracy and area as the input thresholds to the algorithm and compare its energy with that of the output circuit. The energy ratio of the output circuit relative to the input circuit is calculated and recorded in Fig. 2. For all manual designs, the proposed automatic partition algorithm can generate the circuit satisfying the accuracy and area threshold, and having smaller or equal energy. On average, the energy ratio is 53.47%, indicating a 46.53% energy reduction. Moreover, for all pure BC circuits as the inputs, the algorithm can at least find the same design, which is because pure BC circuits generally have the highest accuracy and the smallest energy. Furthermore, some circuits have energy ratios close to 1%, which is because the algorithm generates the pure BC implementations, and as explained previously the pure BC implementations have much smaller energies due to shorter delays.

IV. CONCLUSION

In this paper, we build circuits with different SC/BC partitions for several applications and compare their performance to study the effect of different partitions. We further propose an SC/BC partition algorithm based on iterative replacement to automatically find the best SC/BC partitions for given applications. Experimental results demonstrate the trade-off among accuracy, area, and energy. They also show the proposed algorithm can achieve an average energy reduction of 46.53%, when given the accuracy and area of a manually partitioned circuit as the input

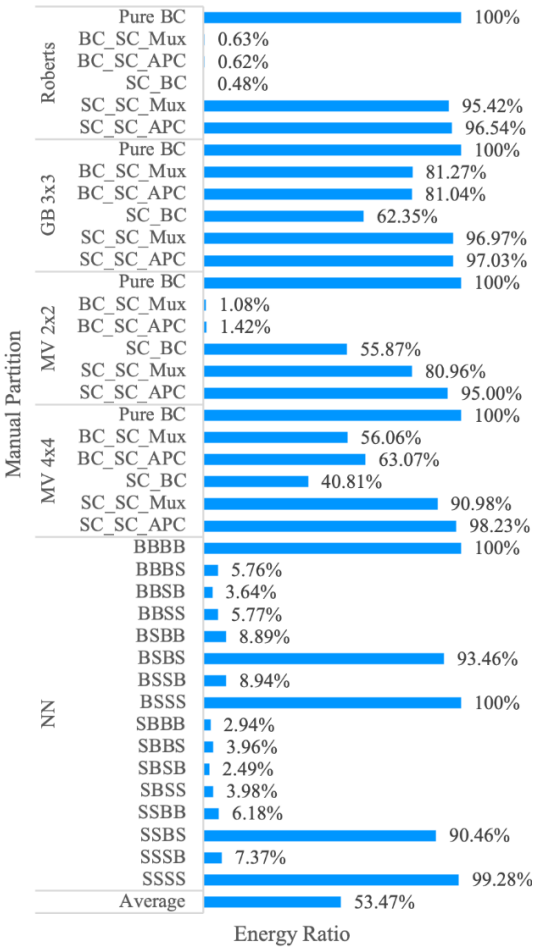

Figure 1: Energy ratio of the circuits generated by the automatic partition algorithm to the input circuits.

threshold. Our future work will enhance the efficiency of the automatic partition algorithm by applying advanced searching methods and support more applications.

REFERENCES

[1] A. Alaghi and J. P. Hayes, "Survey of stochastic computing," *ACM TECS*, vol. 12, no. 2s, pp. 1–19, 2013.

[2] A. Alaghi *et al.*, "Stochastic circuits for real-time image-processing applications," in *DAC'13*, pp. 1–6.

[3] N. Kim *et al.*, "Dynamic energy-accuracy trade-off using stochastic computing in deep neural networks," in *DAC'16*, pp. 1–6.

[4] A. Ren *et al.*, "SC-DCNN: Highly-scalable deep convolutional neural network using stochastic computing," *SIGPLAN Not.*, vol. 52, no. 4, pp. 405–418, 2017.

[5] V. T. Lee *et al.*, "Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing," in *DATE'17*, pp. 13–18.

[6] Z. Li *et al.*, "SCGen: A versatile generator framework for agile design of stochastic circuits," in *DATE'24*, pp. 1–6.