Run-Time Accuracy Reconfigurable Stochastic Computing for Dynamic Reliability and Power Management: Work-in-Progress

Shuyuan Yu*, Han Zhou*, Shaoyi Peng*, Hussam Amrouch†, Joerg Henkel†, Sheldon X.-D. Tan*
* Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521 stan@ece.ucr.edu
† Karlsruhe Institute of Technology, Chair for Embedded Systems (CES), Karlsruhe, Germany

***Abstract*—In this paper, we propose a novel accuracy-reconfigurable stochastic computing (ARSC) framework for dynamic reliability and power management. Different than the existing stochastic computing works, where the accuracy versus power/energy trade-off is carried out in the design time, the new ARSC design can change accuracy or bit-width of the data in the run-time so that it can accommodate the long-term aging effects by slowing the system clock frequency at the cost of accuracy while maintaining the throughput of the computing. We validate the ARSC concept on a discrete cosine transformation (DCT) and inverse DCT designs for image compressing/decompressing applications, which are implemented on Xilinx Spartan-6 family XC6SLX45 platform. Experimental results show that the new design can easily mitigate the long-term aging-induced effects by accuracy trade-off while maintaining the throughput of the whole computing process using simple frequency scaling. We further show that one-bit precision loss for the input data, which translated to 3.44dB of the accuracy loss in term of Peak Signal to Noise Ratio (PSNR) for images, we can sufficiently compensate the NBTI induced aging effects in 10 years while maintaining the pre-aging computing throughput of 7.19 frames per second. At the same time, we can save 74% power consumption by 10.67dB of accuracy loss. The proposed ARSC computing framework also allows much aggressive frequency scaling, which can lead to order of magnitude power savings compared to the traditional dynamic voltage and frequency scaling (DVFS) techniques.**

I. Introduction

Due to the fact that the accurate computing becomes less important for today's emerging computing workloads, as a result, accuracy can be traded off to improve hardware footprint, power/energy efficiencies via so-called approximation computing. One important approach for approximate computing is by means of stochastic computing (SC) [1]. SC is shown to have better error resilience, progressive trade-off among performance, accuracy and energy, as well as cheap implementation of complex arithmetic operations. However, conventional SC suffers long computing time and high randomness of the stochastic numbers for accuracy. Recently, a more efficient and also accurate SC multiplier was proposed to partially mitigate the two mentioned problems in the traditional SC [2]. The whole design is simplified into two counters and a simple bit-stream generator. In this work, we call this design the *counter-based SC multiplier* (CBSC-Multiplier).

Also, reliability issues like bias temperature instability (BTI), hot carrier injection (HCI) for CMOS devices, electromigration (EM) and time dependent dielectric breakdown (TDDB) for interconnects and dielectrics, which are the major consideration for the aging effects [3]–[5], emerge as technology node advances. Fig. 1 shows how BTI affects the maximum frequency of a discrete cosine transformation (DCT) design based on the Nangate 45nm degradation-aware standard cell library from Karlsruhe Institute of Technology (KIT) [6]. Recently using less accurate computing to compensate the NBTI-induced long-term aging effects have been proposed [7]. However this method is targeted at the design time so that sufficient margins can be allocated in advance.

Based on those observations, in this paper, we propose a new accuracy-reconfigurable stochastic computing (ARSC) technique for dynamic long-term reliability management and more power efficient computing. Different than existing stochastic computing works, where the accuracy versus power/energy trade-off is carried out in the design time, the new stochastic computing can change accuracy or bit-width of the data in the run-time so that it can accommodate the long-term aging effects by slowing the system clock frequencies at the cost of accuracy while maintaining the throughput of the computing. As many emerging workloads are error tolerant, the new accuracy-reconfigurable stochastic computing essentially provides viable solution to mitigate the challenging long-term reliability problems due to the increasing degradation effects such as biased temperature instability (BTI) and electromigration (EM) as technology advances. Further more, the proposed reconfigurable SC method can provide new knob to dynamically regulate the active power of a chip as one can scale the frequency in a much larger range (compared to the traditional voltage and frequency scaling techniques) to trade the accuracy for power in a progressive way.

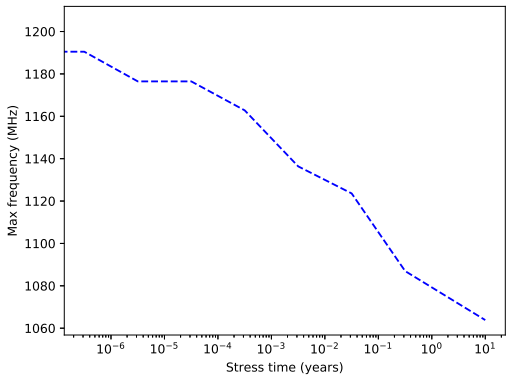

Fig. 1: The maximum working frequency decreases over years because of aging.

We validate the ARSC concept on a DCT and inverse DCT designs for image compressing/decompressing applications, which are implemented on Xilinx Spartan-6 family XC6SLX45 platform. Experimental results show that the new design can easily mitigate the long-term aging induced effects by accuracy trade-off while maintaining the throughput of the whole computing process using simple frequency scaling. We further show that one-bit precision loss for the input data, which translated to 3.44dB of the accuracy loss in term of Peak Signal to Noise Ratio (PSNR) for images, we can sufficiently compensate the NBTI induced aging effects in 10 years while maintaining the pre-aging computing throughput of 7.19 frames per second. At the same time, we can save 74% power consumption by 10.67dB of accuracy loss. The proposed ARSC computing framework also allows much aggressive frequency scaling, which can lead to order of magnitude power savings compared to the traditional dynamic voltage and frequency scaling (DVFS) techniques.

This work is supported in part by NSF grants under No. CCF-1741961, in part by NSF grant under No. CCF-2007135 and No. OISE-1854276.

II. Proposed run-time ARSC for 2D DCT/IDCT

In this section, we present the proposed accuracy-reconfigurable stochastic computing (ARSC) method based on the counter-based SC framework. The key idea is to *dynamically adjust the bit-width* of the coming data for multiplication intensive computing so that we can reduce the accuracy of the computing progressively using SC. At the same time, we also reduce the effective latency of the computing logic so that we can compensate aging-induced delay increases. Since we reduce the effective latency of computing logic, we can reduce the frequency while still being able to maintain the same throughput of the whole computing process as required by the application.

We will illustrate the proposed ARSC method using an image compression application based on computing intensive 2D DCT and inverse DCT algorithms.

A. ARSC architecture for the 2D DCT/IDCT

The primary computing in the 2D DCT/IDCT algorithms are essentially multiply-accumulate operation (MAC). Fig. 2 shows the proposed ARSC-based MAC unit used in the DCT/IDCT applications.

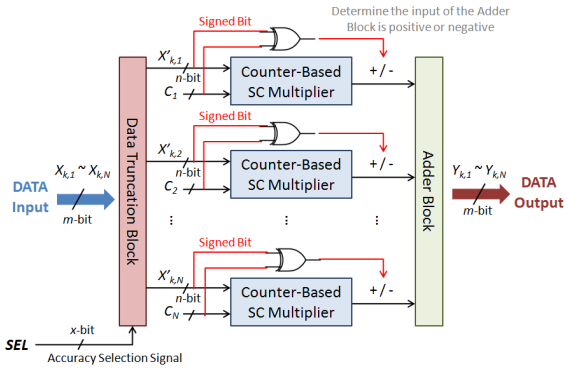

Fig. 2: The proposed ARSC-based MAC unit.

The ARSC MAC unit includes the input data truncation/reconfiguration block, the multipliers and the adder block. We use the CBSC-Multiplier to realize the SC multiplication [2]. The proposed ARSC module does the dynamic accuracy arrangement by adjusting the bit-width of the input data that participate in the CBSC-Multiplier in the ARSC MAC unit, which is realized by the data truncation block. For instance, when the initial data is m-bit, represented in signed-and-magnitude form, $X_{k,i}$ (i = 1, 2, ..., N) in Fig. 2. In this line, an x-bit accuracy selection signal SEL is used to tell the truncation block how many bits it needs to keep. In our design, for instance, we have 5 states representing from 10-bit to 6-bit configurations, so x=3 will be enough to distinguish the 5 states. After data truncation, $X_{k,i}$ is transformed to $X'_{k,i}$, which is a truncated n-bit binary number.

After the ARCS MAC process finishes, we will add 0 at the end of the output number to make it the same bit-width as the input binary number to keep bit-width compatibility between different computing modules. As SC computing time is directly proportional to the bit-width, or more precisely proportional to $O(2^{bit-width})$, one bit-width reduction can dramatically reduce the SC computing time by half, which can be very effective to mitigate the aging effects.

Notice that our CBSC-Multiplier only deals with uni-polar number multiplication, whose range in $[0, 1]$. We keep the sign bits for all the DCT coefficients C_i (i = 1, 2, ..., N). The sign bit of C_i and the sign bit of $X'_{k,i}$ will perform an XOR operation to determine whether the product obtained from the counter-based SC multiplier is positive or negative before participating in the add operation which is carried out at the adder block.

The architecture of the ARSC design consisted of several parts is shown in Fig. 3, including the input buffer, two 2D N-DCT blocks (N is 8 here), logic control unit, an accuracy selection signal and an output buffer. Since SC still has longer computing latency compared to the conventional arithmetic units, we propose to use the parallel computing structure to accelerate the process. The input buffer will send 8 image pixel data to the 2D DCT block once due to the 8-DCT method. The frameworks of the DCT and the inverse DCT block actually are the same. The 2D DCT block is made up of two ARSC MAC units for the 1D DCT process, and an intermediate buffer. The intermediate buffer is used to save the output data from the 1D DCT process as the first ARSC MAC unit get the data from input in row and the second one in column, respectively. The second ARSC MAC unit will not start working until the intermediate data buffer is full. And the Logic Control Unit will send the control signal to all of the blocks to control the data flow of the whole DCT/IDCT process.

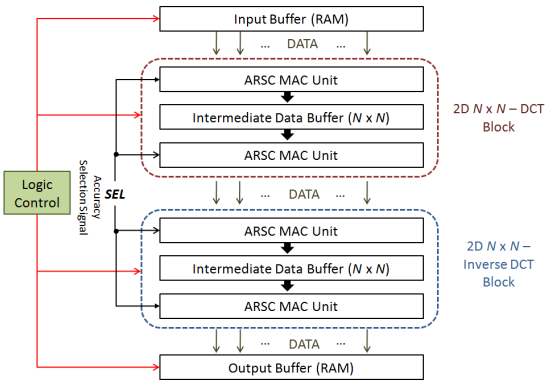

Fig. 3: The top-level diagram for the ARSC-based image compressing/decompressing application.

ARSC Design Hardware Resource Utilization				
Slice Registers	Slice LUTs	LUT FF Pairs	RAMB16BWERS	RAMB8BWERS
11041	17569	18098	66	1

TABLE I: ARSC design hardware resource utilization.

III. Hardware Implementation

To evaluate the hardware cost of the re-configurable stochastic computing module, including the area, delay and power consumption of the module, the proposed design was implemented in Verilog and synthesized using Xilinx ISE 14.7 for XC6SLX45 device of Spartan-6 family. Different from the ASIC-based module, FPGA mainly use the LUT-based operations [8]. So for the design area measurement, we simply count the number of LUTs after the module is synthesized. As mentioned in Sec. II, the design totally utilizes 17569 LUTs to support the parallel SC blocks. We show the details of the FPGA hardware resource utilization information in Table I. To evaluate the power consumption, we use the Xilinx Power Estimator downloaded from the official website, which can easily obtain the total power consumption. We'll discuss the power consumption of the design later in Sec. IV. For the delay measurement, we obtain the critical path of the ARSC design from the Xilinx ISE 14.7 timing summary after the design is synthesized, showing that the hardware delay, which is calculated from the worst case critical path, is 11.348ns. It means that the highest frequency the ARSC design can run is 88.1M. Since we use the digital clock manager (DCM) IP of Xilinx ISE 14.7 to generate the clock signal and the input system clock of the DCM is 100M for the Spartan-6 family boards. The highest frequency DCM can output is 85.7M. So we choose this frequency to be the initial global clock signal of the ARSC design.

IV. Experimental results and discussion

In this section, we present results from the proposed ARSC computing method for the aging mitigation. The proposed ARSC DCT/IDCT image compression algorithms were implemented on the Xilinx XC6SLX45 FPGA platform.

We first show the image compression results with different accuracy in Fig. 4. The Fig. 4(a) shows the original figure without any compression. And Fig. 4(b) ˜Fig. 4(f) show the image quality after the DCT/IDCT sequence computing with different accuracy, from 10-bit to 6-bit. We use PSNR (Peak Signal to Noise Ratio) to evaluate the accuracy of the image compressing/decompressing process, which is shown in Table. II.

(a) (b) (c)

(d) (e) (f)

Fig. 4: (a) The initial image before DCT/IDCT process; (b) Image after DCT/IDCT process using 10-bit stochastic multiplication; (c) 9-bit; (d) 8-bit; (e) 7-bit; (f) 6-bit.

To simulate the aging process by FPGA, we perform the aging analysis for DCT/IDCT design using a degradation aware Nangate 45nm standard cell library from Karlsruhe Institute of Technology (KIT) [6] to calculate the chip frequency changes, about 11.7% change, after 10 years as shown in Fig. 1. Then we adjust the frequency output from the DCM by the same ratio, which is 85.7M to 75.7M (we note that such mapping may not be perfect as the design technologies used in our ASIC and FPGA are different). For the DCT/IDCT process, the aging effect directly affects the throughput.

We show that the throughput with different precision at different clock frequencies in Fig. 5. The x-axis is the bit-width we use in the stochastic multiplication when we perform the DCT/IDCT computing. The y-axis is the throughput, meaning the number of images the design can deal with per second. Fig. 5 also shows the throughput of 7.19 images per second for different frequencies and precision with the dashed black line. As we can see, initially (when the aging process hasn't started yet), if we use the full 10-bit precision, the throughput at 85.7MHz clock frequency is 7.19. When the clock frequency decreases to 43.8MHz, we can still keep the same throughput if we truncate the precision by only one bit (from 10 to 9). Due to the aging process, the throughput will decrease to 6.35. If we degrade the precision by one bit (9-bit), the throughput will increase to 12.42, which obviously, is larger than the initial throughput. The red line of dashes in Fig. 5 shows this very clearly. As the comparison, traditional binary design has to change 3 bits to accommodate the 10 year aging process [7], And, by decreasing the precision of SC multiplication to 6-bit, the throughput will be about 12 times of the 10-bit precision, which shows huge space we can mitigate the aging effects if such accuracy is still accepted in practical applications.

Due to the difficulty of obtaining the hardware delay of the scenarios in which the data bit-width is not 10-bit during SC computing process, we use the effective latency to evaluate the timing performance of our design. The effective latency of the ARSC design is actually the inverse of the throughput, since it represents the time interval between the input and the output. We show the latency at the 5th column of Table II.

Table II shows the effective latency (inverse of the throughput), and power consumption for different bit-width used. From Table II, we show that our design can do very aggressive power management by adjusting the working frequency as well. By doing the trade-off between the throughput and the accuracy mentioned before, we can keep the throughput by sacrificing accuracy when the frequency is cut down due to some low power consumption requirement situation.

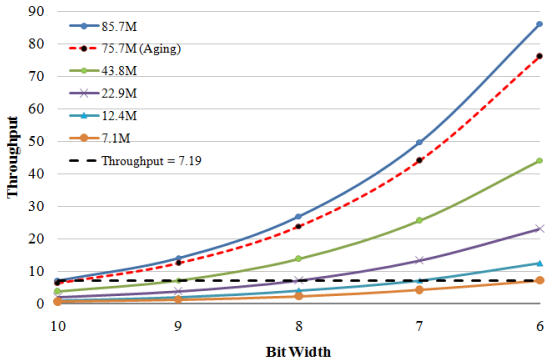

Fig. 5: Accuracy versus throughput considering the aging.

Bit-width	Frequency (MHz)	Power (W)	PSNR (dB)	latency (s)
10	85.7	0.292	38.12	0.139
9	43.8	0.177	34.68	0.071
8	22.9	0.120	31.27	0.037
7	12.4	0.092	28.70	0.020
6	7.1	0.077	27.45	0.012

TABLE II: Performance metric comparison under the same throughput.

For instance, we notice that we can save near 74% of power consumption by sacrificing 10.67dB of the accuracy loss. We also observed that the proposed ARSC computing framework allows much aggressive frequency scaling, which can lead to order of magnitude power savings compared to the traditional dynamic voltage and frequency scaling (DVFS) techniques.

V. Conclusion

In this paper, we have proposed a novel accuracy-reconfigurable stochastic computing (ARSC) framework for dynamic reliability and power management. The new ARSC design can dynamically change accuracy via bit-width change of the data. In this way, the new method can accommodate the long-term aging effects by slowing the system clock frequency at the cost of accuracy while maintaining the throughput of the computing. We designed and validated the ARSC-based discrete cosine transformation (DCT) and inverse DCT designs for image compressing/decompressing applications on the Xilinx Spartan-6 family XC6SLX45 platform.

References

[1] A. Alaghi, W. Qian, and J. P. Hayes, "The promise and challenge of stochastic computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1515–1531, 2018.

[2] H. Sim and J. Lee, "A new stochastic computing multiplier with application to deep convolutional neural networks," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, IEEE, 2017.

[3] S. X.-D. Tan, M. Tahoori, T. Kim, S. Wang, Z. Sun, and S. Kiamehr, *VLSI Systems Long-Term Reliability – Modeling, Simulation and Optimization*. Springer Publishing, 2019.

[4] S. X.-D. Tan, H. Amrouch, T. Kim, Z. Sun, C. Cook, and J. Henkel, "Recent advances in EM and BTI induced reliability modeling, analysis and optimization," *Integration, the VLSI Journal*, vol. 60, pp. 132–152, Jan. 2018.

[5] "Critical Reliability Challenges for The International Technology Roadmap for Semiconductors (ITRS)," 2003. In International Sematech Technology Transfer Document 03024377A-TR, 2003.

[6] "Degradation-aware cell libraries, v1.0." http://ces.itec.kit.edu/dependable-hardware.php.

[7] H. Amrouch, B. Khaleghi, A. Gerstlauer, and J. Henkel, "Towards aging-induced approximations," in *Proceedings of the 54th Annual Design Automation Conference 2017*, pp. 1–6, 2017.

[8] Y. Guo, H. Sun, and S. Kimura, "Small-area and low-power fpga-based multipliers using approximate elementary modules," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 599–604, IEEE, 2020.