Introduction to Approximate Computing

Jie Han
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta, Canada
jhan8@ualberta.ca

***Abstract*—Approximate computing has emerged as a new paradigm for energy-efficient design of circuits and systems. This paper presents a brief introduction to approximate computing as well as to the challenges faced by approximate computing with respect to its prospects for applications in energy-efficient and error-resilient computing systems.**

Keywords—approximate computing; arithmetic circuit; power; energy; performance; fault tolerance, error resilience

I. Introduction

The continuous miniaturization of electronic devices requires fault-tolerant and variation-resilient designs to ensure operational reliability during their lifetime and to accommodate the inevitable variations in nanoscale manufacturing processes. Conventional fault-tolerant techniques, however, rely on the use of hardware, time and/or information redundancies, which can lead to significant overhead in energy consumption. The conflict between reliability and energy efficiency seems to be inevitable and presents significant design challenges. Indeed, as eloquently stated by John von Neumann some sixty years ago, "Our present treatment of error is unsatisfactory and ad hoc. … Error is viewed (in this work), therefore, not as an extraneous and misdirected or misdirecting accident, but as an essential part of the process under consideration …" [1] This view, that noise or error is an integral part of a system, is just as valid today as it was in the early days of computers.

In contrast to the *passive* use of redundancies, approximate computing (AC) employs *active* design methodologies that exploit the nature that many systems and applications can tolerate some loss of accuracy in the computation result [2]. Such applications include multimedia processing (audio, video, graphics and image), recognition, search, data mining and learning. AC leverages the unique feature of error resilience in these applications and searches for solutions that allow computing systems to trade quality for energy.

AC spans a wide range of research activities from circuits to programming languages. It includes arithmetic circuit design at the transistor and logic levels [3], approximate memory and storage [4] (including SRAM, DRAM and non-volatile memories), and various approximate processor architectures [5] (including neuron networks, general-purpose and reconfigurable processors such as instruction set architectures (ISAs), graphic processing units (GPUs) and FPGAs). Applications of AC have included image and signal processing, classification and recognition, machine learning, among others.

II. Prospects and Challenges

In parallel with the advances in AC, brain-inspired computing has gained momentum in the past few years [6]. A brain-inspired computing system explores the organization and functions of neurons and connecting synapses at different levels of hierarchy and abstraction. The objective is to build a new class of computers that implement massive device parallelism and circuit adaptability that can tolerate unreliable operations and, at the same time, achieve greater energy efficiency [7]. AC techniques appear promising to be integrated into the algorithms and architectures of a brain-inspired computing system.

In spite of the recent progress, questions remain as to how an approximate circuit component can be most effectively integrated into a large computing system for meeting the various quality requirements of error-tolerant applications. How much of the gains in performance and energy efficiency that have been observed at the component level would remain in the system that provides a quality assurance? These are essential questions to answer before approximate computing can be made a viable and practically relevant paradigm for energy-efficient and error-resilient computing.

References

[1] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," Automata Studies, Shannon C.E. & McCarthy J., eds., Princeton University Press, pp. 43-98, 1956.

[2] J. Han and M. Orshansky, "Approximate Computing: An Emerging Paradigm For Energy-Efficient Design," in ETS'13, Proc. 18th IEEE European Test Symp., Avignon, France, May 27-31, 2013.

[3] S. Venkataramani, S.T. Chakradhar, K. Roy, and A. Raghunathan, "Computing approximately, and efficiently," In Design, Automation & Test in Europe Conference & Exhibition, pp. 748-751, 2015.

[4] A. Sampson, J. Nelson, K Strauss, and L. Ceze, "Approximate storage in solid-state memories," ACM Transactions on Computer Systems (TOCS), vol. 32, no. 3, 9, 2014.

[5] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Architecture support for disciplined approximate programming," ACM SIGPLAN Notices, vol. 47, no. 4, pp. 301-312, 2012.

[6] G. Snider, R. Amerson, D. Carter, H. Abdalla, M.S. Qureshi, J. Léveillé, M. Versace, H. Ames, S. Patrick, B. Chandler, A. Gorchetchnikov, and E. Mingolla, "From synapses to circuitry: using memristive memory to explore the electronic brain," Computer, 44 (2), 21, 2011.

[7] Reports of IEEE Rebooting Computing Summits. http://rebootingcomputing.ieee.org/