Fault Tolerance of Approximate Compute Algorithms

Wunderlich, Hans-Joachim; Braun, Claus; Schöll, Alexander

Proceedings of the 34th VLSI Test Symposium (VTS'16) Caesars Palace, Las Vegas, Nevada, USA, 25-27 April 2016

doi: http://dx.doi.org/10.1109/VTS.2016.7477307

Abstract: Approximate computing algorithms cover a wide range of different applications and the boundaries to domains like variable-precision computing, where the precision of the computations can be online adapted to the needs of the application [1, 2], as well as probabilistic and stochastic computing [3], which incorporate stochastic processes and probability distributions in the target computations, are sometimes blurred. The central idea of purely algorithm-based approximate computing is to transform algorithms, without necessarily requiring approximate hardware, to trade-off accuracy against energy. Early termination of algorithms that exhibit incremental refinement [4] reduces iterations at the cost of accuracy. Loop perforation [5] approximates iteratively-computed results by identifying and reducing loops that contribute only insignificantly to the solution. Another group of approximate algorithms is represented by neural networks, which can be trained to mimic certain algorithms and to compute approximate results [6]. Today, approximate computing is predominantly proposed for applications in multimedia and signal processing with a certain degree of inherent error tolerance. However, in order to fully utilize the benefits of these architectures, the scope of applications has to be significantly extended to other computeintensive tasks, for instance, in science and engineering. Such an extension requires that the allowed error or the required minimum precision of the application is either known beforehand or reliably determined online to deliver trustworthy and useful results. Errors outside the allowed range have to be reliably detected and tackled by appropriate fault tolerance measures.

Fault Tolerance of Approximate Compute Algorithms

Hans-Joachim Wunderlich, Claus Braun, Alexander Schöll

Institute of Computer Architecture and Computer Engineering, University of Stuttgart
Pfaffenwaldring 47, D-70569, Germany, Email: {wu,braun,schoell}@informatik.uni-stuttgart.de

I. Approximate Computing Algorithms

Approximate computing algorithms cover a wide range of different applications and the boundaries to domains like *variable-precision computing*, where the precision of the computations can be online adapted to the needs of the application [1, 2], as well as *probabilistic* and *stochastic computing* [3], which incorporate stochastic processes and probability distributions in the target computations, are sometimes blurred. The central idea of purely algorithm-based approximate computing is to transform algorithms, without necessarily requiring approximate hardware, to trade-off accuracy against energy. Early termination of algorithms that exhibit incremental refinement [4] reduces iterations at the cost of accuracy. Loop perforation [5] approximates iteratively-computed results by identifying and reducing loops that contribute only insignificantly to the solution. Another group of approximate algorithms is represented by neural networks, which can be trained to mimic certain algorithms and to compute approximate results [6].

Today, approximate computing is predominantly proposed for applications in multimedia and signal processing with a certain degree of inherent error tolerance. However, in order to fully utilize the benefits of these architectures, the scope of applications has to be significantly extended to other compute-intensive tasks, for instance, in science and engineering. Such an extension requires that the *allowed error* or the *required minimum precision* of the application is either known beforehand or reliably determined online to deliver trustworthy and useful results. Errors outside the allowed range have to be reliably detected and tackled by appropriate fault tolerance measures.

II. Test and Online Test

Conventional test methods which are intended to validate the full functionality of circuits by evaluating the absence of faults are often not directly applicable to approximate hardware. Existing test methods must be extended by appropriate metrics and characterization procedures which assess whether the circuit is either critical, marginal, or non-critical. Moreover, new monitoring and online test methods are required to cover the entire lifecycle of approximate circuits. Regardless of whether such online tests will be performed in a concurrent or non-concurrent manner, some of the underlying metrics will be very similar to those metrics required for the online determination of the required minimum precision and hence for the detection of critical errors [7]. This enables synergies between the required online test and fault tolerance measures.

III. Fault Tolerance

For approximate computations, the algorithm and application level is an attractive insertion point for flexible fault tolerance measures, specifically designed for the target application. The minimum required precision is an essential criterion for the assessment of approximate computational results. In science and engineering, many applications rely on floating-point arithmetic. Hence, the inevitable *rounding error* that affects computed results can be used to determine the minimum required precision since there is typically no need to compute more accurately than what rounding allows.

Probabilistic error functions enable the derivation of rounding error bounds based on the performed arithmetic operations and the processed data. The basic idea is to determine a confidence interval for the rounding error based on its expectation value and variance. Errors which are larger than the probabilistic rounding error bound are harmful and have to be tackled. Probabilistic error functions can be applied for the offline characterization of applications, as well as the online monitoring and checking of approximate results.

Many iterative numerical methods (e.g. solvers) seem to be well-suited for approximate hardware since they impose an inherent resilience to certain numerical errors [8]. However, appropriate *fault tolerance techniques* are required to further increase their resilience and to allow their execution on approximate computing structures with even increased low-power constraints.

Bibliography

[1] M. Schulte and E. Swartzlander, "A Family of Variable-Precision Interval Arithmetic Processors", *IEEE Transactions on Computers*, vol. 49, no. 5, pp. 387–397, May 2000.

[2] C.-C. Hsiao, S.-L. Chu, and C.-Y. Chen, "Energy-aware Hybrid Precision Selection Framework for Mobile GPUs", *Computers & Graphics*, vol. 37, no. 5, pp. 431 – 444, 2013.

[3] A. Alaghi and J. P. Hayes, "Survey of Stochastic Computing", *ACM Trans. Embed. Comput. Syst.*, vol. 12, no. 2, pp. 92:1–92:19, May 2013.

[4] S. H. Nawab *et al.*, "Approximate Signal Processing", *Journal of VLSI Signal Processing Systems For Signal, Image and Video Technology*, vol. 15, no. 1-2, pp. 177–200, 1997.

[5] S. Sidiroglou-Douskos *et al.*, "Managing Performance vs. Accuracy Trade-offs with Loop Perforation", in *Proc. of the 13th Europ. Conf. on Foundations of Software Engineering*, 2011, pp. 124–134.

[6] T. Moreau *et al.*, "SNNAP: Approximate Computing on Programmable SoCs via Neural Acceleration", in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2015, pp. 603–614.

[7] M. Ringenburg *et al.*, "Monitoring and Debugging the Quality of Results in Approximate Programs", in *Proc. Intl. Conf. on Architectural Support for Programming Languages and Operating Systems*, 2015, pp. 399–411.

[8] Q. Zhang *et al.*, "ApproxIt: An Approximate Computing Framework for Iterative Methods", in *Proc. 51st ACM/EDAC/IEEE Design Automation Conference (DAC'14)*, 2014, pp. 1–6.