Elastic Fidelity: Trading-off Computational Accuracy for Energy Reduction

Sourya Roy[†], Tyler Clemons[‡], S M Faisal[‡], Ke Liu[†], Nikos Hardavellas[†], Srinivasan Parthasarathy[‡]

[†]Department of Electrical Engineering & Computer Science
Northwestern University
{souryaroy, nikos}@northwestern.edu, KeLiu2015@u.northwestern.edu

[‡]Department of Computer Science & Engineering
Ohio State University
{clemonst, faisal, srini}@cse.ohio-state.edu

Abstract

Power dissipation and energy consumption have become one of the most important problems in the design of processors today. This is especially true in power-constrained environments, such as embedded and mobile computing. While lowering the operational voltage can reduce power consumption, there are limits imposed at design time, beyond which hardware components experience faulty operation. Moreover, the decrease in feature size has led to higher susceptibility to process variations, leading to reliability issues and lowering yield. However, not all computations and all data in a workload need to maintain 100% fidelity. In our study, we explore the idea of employing functional or storage units that let go the conservative guardbands imposed on the design to guarantee reliable execution. Rather, these units exhibit Elastic Fidelity, by judiciously lowering the voltage to trade-off reliable execution for power consumption based on the error guarantees required by the executing code. By estimating the accuracy required by each computational segment of a workload, and steering each computation to different functional and storage units, Elastic Fidelity Computing obtains power and energy savings while reaching the reliability targets required by each computational segment. Our preliminary results indicate that even with conservative estimates, Elastic Fidelity can reduce the power and energy consumption of a processor by 11-13% when executing applications involving human perception such as audio, image, and video decoding, as well as various data mining kernels such as clustering algorithms and frequent itemset mining.

Overview

Continued technology scaling in IC design has made power dissipation a major constraint in the design of processors today. Although feature sizes are still scaling, voltage scaling has nearly stopped due to high leakage currents associated with low threshold voltages. This has lead to a dramatic increase in power density with decreasing feature size. Additionally, the scaling of the feature sizes has made chips more susceptible to problems of variability and hardware faults. These faults originate from process variations, soft errors and wear outs, hampering reliable execution.

Traditionally, the operating points of processors have been determined by conservative guardbands based on worst-case scenarios. However, this design approach results in significant overheads in both power and performance, leading to an interesting question: What if we let go of these guardbands and allow components of the processor to fail sometimes with the errors accommodated at the architectural and software levels? By following laws of transistor physics, keeping all else constant, decreasing the operating voltage (V_{dd}) would reduce power consumption at a quadratic rate, at the expense of some timing errors.

Prior research has shown that in every large CMOS processor, there are three operating regions. First, when the supply voltage is at or above the rated voltage, the processor runs at full accuracy without any errors. Second, when the processor operates at a voltage between the rated and critical voltage points, small-scale errors emerge due to timing violations in worst-case situations. And last, operating at a voltage beyond this critical point leads to massive errors.

In our study, we propose the idea of operating processor components (e.g., functional units) at the region of supply voltage between the rated and critical operating points, to attain significant reductions in power while meeting the reliability requirements requested by each section of the executing application. The errors originating due to this are accommodated at the software layer by exploiting the fact that different sections of the code require variable reliability guarantees to present acceptable results to the user.

We envision that programming language constructs can denote the reliability guarantees required by different sections of the code; these requirements are communicated to the hardware during execution, which steers the computation to corresponding functional and storage units operating at the lowest voltage that meets the required reliability constraints. The correlation between voltage and error manifestation can be determined during hardware testing by adding a tuning phase for the components amenable to elastic fidelity, or online through error detection. The reliability guarantees required by each code or data segment can be similarly determined through feedback-directed software optimization and user input.

By not treating all code and all data the same from the viewpoint of reliability requirements, *Elastic Fidelity Computing* exploits sections of the computation that are error-tolerant to lower power and energy consumption, without negatively impacting executions that require full reliability. The level of error tolerance, in fact, is application-dependent and is determined by how accurate a program's output needs to be. There are applications which are highly resilient inherently and there are others which are very little. Important examples of highly-resilient applications come from the class of soft computing. Unlike hard or exact computing, soft computing takes advantage of the tolerance of imprecision, uncertainty and approximation for a given problem – resulting in acceptable rather than exact results. Multimedia applications offer a very interesting example of soft computing as they primarily depend on human perception and allow considerable leeway in terms of accuracy. Another class of examples comes from data mining applications such as the k-means clustering algorithm. Typically, applications of this class converge upon near-optimal solutions. Solutions need not be identical and thus we can offer approximations.

Methodology & Result Highlights

To explore the feasibility of our idea, we examine the error tolerance of a range of applications involving human perception and data mining in the presence of computation errors in ALUs. We simulate the applications under elastic fidelity by injecting errors in the computations at run time through software wrappers. Our preliminary results corroborate our hypothesis that different portions of an application's dataset and code exhibit variable error tolerance. As a result, we demonstrate that even if we allow only the ALUs to exhibit Elastic Fidelity, and without any modifications to the applications' algorithms, Elastic Fidelity Computing reduces the processor's power and energy by 11-13%. We anticipate that expanding this idea to more execution and storage components of a processor would result in much higher power savings.