Designing Energy-Efficient Arithmetic Operators Using Inexact Computing

Avinash Lingamneni1,3, ∗, Christian Enz3, Krishna Palem1,2, and Christian Piguet3

1Department of Electrical and Computer Engineering, Rice University, Houston, Texas 77005, USA
2NTU-Rice Institute for Sustainable and Applied Infodynamics (ISAID), Nanyang Technological University, Singapore
3Wireless and Integrated Systems, Centre Suisse d’Electronique et de Microtechnique (CSEM) SA, Neuchatel 2000, Switzerland

(Received: 22 December 2012; Accepted: 11 February 2013)

It is widely acknowledged that the exponentially improving benefits of sustained technology scaling prophesied by the Moore’s Law would end within the next decade or so, attributed primarily to an understanding that switching devices can no longer function deterministically as feature sizes are scaled down to the molecular levels. The benefits of Moore’s Law could, however, continue provided systems with probabilistic or “error-prone” elements could still process information usefully. We believe that this is indeed possible in contexts where the “quality” of the results of the computation is perceptually determined by our senses—audio and video information being significant examples. To demonstrate this principle, we will show how such “inexact” computing based devices, circuits and computing architectures can be used effectively to realize many ubiquitous energy-constrained error-resilient applications. Further, we show that significant energy, performance and area gains can be achieved, while trading a perceptually tolerable level of error—that will be ultimately determined based on neurobiological models—applied in the context of video and audio data in digital signal processing. This design philosophy of inexact computing is of particular interest in the domain of embedded and (portable) multimedia applications and in application domains of budding interest such as recognition and data mining, all of which can tolerate inaccuracies to varying extents or can synthesize accurate (or sufficient) information even from inaccurate computations!

Keywords: Inexact Computing, Probabilistic and Approximate Computing, Probabilistic Pruning, Probabilistic Logic Minimization, Error Tolerant Applications, Adder, Multiplier, Low Power.

1. INTRODUCTION AND BACKGROUND

Over the last decade, there has been an increasing interest in innovations in realizing “good-enough” and parsimonious systems that could afford an imperceptible degradation in their output quality if they could get significant resource-savings (either energy consumption, critical path delay and/or silicon area) in exchange. This was driven in part by the necessity to innovate and sustain the technology scaling (and as a result, the resource efficiency) prophesied by the Moore’s law which was facing the hurdles of process- and parameter-variation induced errors as well as the forecasts that a large part of the emerging applications would involve Recognition, Mining and Synthesis (RMS) workloads, most of which exhibit varying levels of error resilience. These systems as well as a large part of the present day multimedia and Digital Signal Processing (DSP) systems can tolerate varying amounts of error and still realize potentially useful computations, largely owing to the “cognitive filling” or perceptual limitations of the end systems consuming the output and also, the absence of a single “golden” result owing to their inherent statistical and aggregative characteristics. For example, in applications where the workloads interact with human perception, such as vision and audition, the neurobiological pathways are intimately tied into the information that is being computed and hence, we envision that employing perceptually guided architectures incorporating human tolerance for error based on neurobiological foundations will be an important future direction for innovation for systems, in particular the severely constrained smart systems. Hence, in these systems, any error (either caused probabilistically due to inherent device variations or perturbations, or induced deterministically) can be viewed as a commodity that can be exchanged for substantial savings in hardware
cost (energy, delay and/or area) without the necessity of any error correction mechanisms. This foundational principle of inexact computing along with its thermodynamic interpretations was conceptualized almost a decade ago by Palem in his works.\textsuperscript{1, 2} Since then, this notion of inexact computing found its way into a wide variety of DSP, multimedia and probabilistic applications to realize highly energy-efficient systems and has been considered distinct to techniques driven by necessity for exactness by employing varying schemes for error correction mechanisms (for example, spatial redundancy schemes such as Triple Modular Redundancy (TMR) or temporal redundancy schemes such as RAZOR),\textsuperscript{3} that are applicable to general purpose computing. In other words, we distinguish the phrase inexact circuit design to refer to an approach to realizing information processing frameworks—from transistors, gates, data-path elements all the way up to more domain-specific complex macroscopic engines—which are deliberately designed to be erroneous and used as such without any (hardware) overhead of error-correction mechanisms, in return for significant savings in hardware resources (for example, energy, performance, yield and/or area).\textsuperscript{4}

In this paper, we present an overview of our work (from\textsuperscript{5–7}) in realizing energy-parsimonious arithmetic operators (such as adders and multipliers) through inexact computing and demonstrate the perceptually acceptable results that they deliver when used in designing more complex datapath systems. We choose datapath elements as demonstrative examples for our techniques as they are considered the most energy consuming components in the targeted error tolerant applications (for example, upto 75% power consumption of the motion estimation block, ubiquitous to most video codecs, is attributed to the datapath elements).\textsuperscript{8}

The rest of the paper is organized as follows: in Section 2, we present a succinct literature survey and understand the existing techniques through the lens of a newly coined term–potential lowering. In Section 3, we present a concise overview of our cross-layer co-design framework (CCF) encompassing both physical- and architectural-layer techniques for realizing highly energy-efficient inexact circuits with relevant illustrative examples. In Section 4, we describe the experimental framework used to validate the proposed techniques and highlight the resource savings we could achieve for varying error tolerance levels along with some image examples to demonstrate the qualitative perceptual degradation when used in more complex datapath systems. Section 5 concludes the paper and presents a multitude of possible future research directions in this domain.

2. INEXACT TECHNIQUES THROUGH THE LENS OF POTENTIAL LOWERING

Inexact circuits are parsimonious or “adequately-engineered” in terms of (physical) implementation and cost (quantified through energy, delay and/or area metrics) much lesser than their “over-engineered” conventional correct counterparts while achieving the needed quality of output for a target application. In this section, we aim to assess the previously proposed physical-level (e.g., voltage overscaling)\textsuperscript{8, 9} and architectural-level (e.g., probabilistic pruning\textsuperscript{10} and probabilistic logic minimization\textsuperscript{11} techniques by coining a term called potential lowering, broadly referring to a lowering of the energy configuration.

Let a node\textsuperscript{a} in a circuit have a cost $C_i$ with a corresponding error $\xi_i$ in the output (quantified through error metrics). Interpreting the inexact techniques through the lens of potential lowering implies a lowering of the cost of the node at the expense of being “somewhat” erroneous as shown in Figure 1. Specifically, the node is transformed to have cost $C_i' < C_i$ and error $\xi_i' \geq \xi_i$.

The potential lowering of a node for the accuracy trade-off has been typically done by either voltage overscaling or (switching) capacitance reduction as shown in Figure 1. The voltage overscaling technique provides a fine-grained way (theoretically) to enable energy-accuracy tradeoffs typically by not scaling the sampling frequency (thereby, violating the (increased) critical path delay).\textsuperscript{8–12} This technique was highly popular and widely used by a vast plethora of papers for a multitude of error tolerant applications. For instance, a manifestation of this technique termed biased voltage overscaling (BiVOS) was used to realize arithmetic adders\textsuperscript{9, 12} and its impact was evaluated using transforms on image datasets. In this BiVOS technique, multiple (variable) supply voltage planes were utilized such that the computational blocks corresponding to the most significant bits were placed in the higher voltage plane, thereby incurring lower error magnitudes. Voltage overscaling-based inexact approaches were also used to implement other signal processing and multimedia applications including Discrete Cosine Transform,\textsuperscript{13} Motion Estimation\textsuperscript{8, 10} and Image processing.\textsuperscript{14} More recently, this technique has also been applied at the granularity of processor modules through an approach popularly referred to as stochastic computing.\textsuperscript{11, 15}

However, on the flipside, the voltage overscaling techniques came with associated overheads due to amortized cost of the level-shifters, metastability resolutions circuits and routing of multiple voltage lines. The practical realizations only implement a subset of well characterized voltage levels owing to these overheads and also, due to the impeding power supply variations.\textsuperscript{16} While on the upside, this technique provides more dynamic control and quadratic energy savings, such overheads are seldom amortized at

---

\textsuperscript{a}Henceforth, we use the word node as an overloaded term depending on the circuit under consideration to imply—a collection of basic gates such as full adders or propagate-generate blocks in basic arithmetic blocks; arithmetic blocks such as adders or multipliers in dataflow graphs, for instance FFTs; interconnected dataflow graphs such as a group of FFTs or filter banks in a complex system, for instance a hearing aid.
the level of small-scale building blocks and drastically reduce the gains in the targeted systems of interest.\textsuperscript{5, 17}

It is due to these inhibiting overheads that many recent efforts have moved away from the physical level to explore inexact logic/architecture-level techniques\textsuperscript{4–7, 17, 18} and were able to realize inexact circuits with zero hardware overhead with savings across all 3 dimensions—energy, delay and area. These techniques advocated reduction of logic density by either pruning/deletion of non-significant components\textsuperscript{5} or by transformations to lesser power consuming yet similar logic.\textsuperscript{6, 17, 18}

In our view, the precision reduction or bit-width truncation approaches, widely pervasive in the DSP and embedded systems domain, enabled more coarse-grained energy-accuracy tradeoffs from the perspective of a node as it can only provide the extremum potential (either a node/sub-circuit is deleted or it is not). On the other hand, the probabilistic pruning and logic minimization approaches offer a much more fine-grained control over the energy-accuracy tradeoffs at the design time enabling a much larger design space. This is demonstrated in Figure 2 where the design space (Mean Error versus Energy Savings) of the truncated multipliers and the inexact multipliers is provided. In this figure, a standard 16-bit truncated multiplier with variable correction\textsuperscript{19} (16-bit inputs and 16-bit output) is used as a comparison baseline using uniformly distributed input vector in the range $[-1,1]$. For the truncation approach, the bit-width is reduced from 16-bits down to 10-bits in decrements of 1 and the corresponding mean error versus energy savings point of these 7 architectures are plotted (indicated by red star symbols). On the other hand, the inexact logic minimization schemes generate a richer design space (marked by the green circles) due to their ability to tune logic functions at a much finer scale. As evident from the figure, we always find multiple inexact multiplier architectures that have higher energy gains for a given error budget than the plain truncation scheme.

2.1. Inexact Circuits and Classification of Applications

Broadly, applications can be broadly classified into three classes based on their error-resiliency.\textsuperscript{4}

(i) Applications that profit from the errors (or perturbations) — examples include algorithms with a large number of probabilistic steps such as randomized test for primality, Bayesian inferencing and certain cryptographic blocks (e.g., Hyperencryption).

(ii) Applications that can tolerate but do not benefit from errors—most of the (digital) signal processing and multimedia systems fall into this category. The design techniques for these applications need not attempt to correct the errors introduced by components that are susceptible to perturbations, instead “good enough” systems would suffice.

(iii) Applications that cannot tolerate any errors—safety critical applications (e.g., navigation systems used in automobiles or aviation). In the presence of erroneous outputs (either due to PVT variations or inexact hardware), these applications would require error detection and correction schemes.

It follows from the above classification that the domain of inexact computing is limited to applications belonging to classes (i) and (ii). Also, due to ubiquity and significantly larger number of systems built for class (ii) applications, we focus our efforts into identifying, designing and validating several building blocks and systems targeted at this class of applications. For class (i) applications, interested readers can refer to Refs. [20, 21]. For the more curious readers interested in the background and timelines for

![Fig. 1. Classifying the various inexact techniques as seen by a node through the lens of potential lowering.](image1)

![Fig. 2. A comparison of the design space of bit-width truncated multipliers and inexact multipliers using a 16-bit truncated multiplier as a baseline.](image2)
2.2. Contributions of Our Work

In this subsection, we would to highlight some of the salient features and unique contributions of our work and research focus:

• In a sharp contrast with the popular push for designing “exact” or highly reliable systems from “erroneous” or unreliable components, we advocate the notion of designing “good-enough” or relaxed-reliable systems that are parsimonious in the resources they consume, yet deliver a solution, leveraging the non-linear resource-accuracy trade-off curves, that is acceptable to the system consuming the output (for example, human sensory systems). To push the limits of this resource parsimony through accuracy trade-off, we provide novel approaches and design frameworks for intentionally transforming components into their erroneous manifestations, which come for a much lower resource cost.

• Most of the initial work in realizing inexact hardware systems focused on leveraging the supply voltage as the control knob for gleaning the quadratic energy savings made possible through voltage overscaling. However, as pointed out before, this technique suffers from inefficient amortization of (hardware) overheads while designing smaller (yet critical) building blocks such as the datapath arithmetic circuits. In lieu of this, we were among the first to highlight the drawbacks of using voltage scaling solely as the control knob and push for a shift towards the architecture- and logic-layer techniques (for example, probabilistic pruning and probabilistic logic minimization) with a focus on realizing inexact circuits with zero hardware overheads and savings across all the (hardware) resources.

• In a quest for highly energy-parasimonious systems—a dominant portion of the current mobile, embedded and DSP systems (potential targets for inexact computing)—where energy minimization is the primary concern with area and delay reduction playing a second fiddle, we provide for a symbiotic cross-layer co-design framework (CCF) that still realizes zero-overhead inexact circuits by trading some portion of the delay and area savings for substantially higher energy savings through a technique called confined voltage scaling.

• Contrary to the largely simulation-driven and model-driven validation of the inexact circuits in literature, we provide a industry-standard design flow for designing, implementing and validating the inexact circuits and verify that the measurement results from the silicon tape-outs track the predicted results from our simulation framework.

3. POTENTIAL LOWERING OF ARITHMETIC OPERATORS THROUGH CROSS-LAYER CO-DESIGN TECHNIQUES

One of the central goal of this paper is to demonstrate the potentially significant savings enabled through a cross-layer co-design framework (CCF) wherein we advocate a conducive symbiotic relationship between the architecture and logic-layer approaches on one hand and physical-layer approaches on the other, governed by the application algorithm, as shown in Figure 3.

To elaborate, while the architecture- and logic-layer techniques such as probabilistic pruning do achieve gains in all three aspects—energy, delay and area, the energy gains by themselves are not dramatically high owing to the sub-linear returns on energy gains from reducing the effective switching capacitance. On the other hand, voltage scaling techniques provide an opportunity for quadratic energy gains while having a negative (linear) effect on the delay of the circuit and a negative effect on the circuit area as well due to the necessity for the control circuitry needed to facilitate voltage scaling. Hence, we focus on fostering a co-design between the two levels of abstraction, wherein the logic-layer techniques rely on the physical-layer technique for boosting its modest energy gains while the physical-layer techniques rely on the logic-layer techniques to compensate for its delay and area overheads. We show that CCF would result in more significant cumulative savings across all three cost dimensions (translating to higher yield), while overcoming any of the drawbacks associated with the individual approaches limited to either architectural- or physical-layers. To illustrate the exploration of inexact circuit design space using CCF, we select probabilistic pruning to represent the architectural/logic-layer techniques while we use Confined Voltage Scaling (CVS) to represent physical-layer techniques.

Fig. 3. A Cross-layer co-design framework for potential lowering through inexact design techniques.
3.1. Designing Inexact Operators Through Probabilistic Pruning/Logic Minimization

Probabilistic Pruning is a technique through which we “prune” or delete computational blocks and their connecting wires from a fully functional circuit design. The criterion for deciding that a block can be removed is based on the significance that the node has in contributing to the output value, and also its activity level when the circuit is exercised using a canonical set of inputs. For example, a node whose output has a lot of significance but is pretty dormant across most inputs of interest (determined by the application workload’s distribution in general, assumed to be uniform in our examples) is a candidate. Similarly, a very active node can be a candidate for pruning if the values it computes do not have significant contribution to the outputs. We use the Significance-Activity Product (SAP) as a basis for determining the relative importance of each of the nodes. The pruning technique we use here is versatile and can be applied at varying node granularities: nodes can be a micro-block such as a gate for logic-layer pruning or a macro-block such as an adder or multiplier, for architectural-layer pruning. Conceptually, the pruning algorithm can be broken down into three steps:

(i) Ranking nodes by their significance which determines their relative priorities as inputs to the next step,
(ii) Pruning the nodes in the decreasing order of ranks where higher ranked nodes are preserved at the expense of those that have lower ranks, and
(iii) Healing which reconnects the blocks that might have become isolated due to the pruning in the previous step.

On the other hand, Probabilistic logic minimization is another zero overhead logic-layer technique that systematically minimizes logic blocks to a lesser cost logic while limiting the errors to acceptable margins. While this technique is computationally more intensive than the probabilistic pruning technique, it achieves more savings due to its inherent ability to “finetune” the logic blocks in a circuit as opposed to a “binary notion” of either pruning the block or not.

3.1.1. Demonstrative Example of the Probabilistic Pruning Technique

In order to demonstrate the application of probabilistic pruning, we select the carry path circuit of a typical 16-bit Kogge Stone adder as shown in Figure 4. We present two heuristics (depending on the evaluation error metric) to assigning significance to nodes in an adder: in the first heuristic, we assign significance to a node based on the positional significance of the output nodes directly affected by the current node as shown in Figure 4(a). In the second approach, we assign the same significance to all output nodes irrespective of where they occur in the circuit, as shown in Figure 4(b). Empirically, we have noticed that the former approach leads to lower relative and average error magnitudes, whereas the latter approach leads to lower error rates.

We compute the probability of a node being active, or the activity of a node by calculating the probability of any particular path from a pair of input bits contributing to an output Sum being active. Assuming that the inputs are drawn from a uniform distribution, this probability value is equal to \( \frac{1}{2^i} \) for \( i \geq j \). The rank of a node is the product of its significance and its activity and we will use SAP to denote this product. In Figure 4, the SAP of the nodes in a Kogge-Stone adder, which are the candidates for pruning, are shown inside the boxes for 16 nodes. To reiterate, whenever there is a choice between nodes to be pruned, those with the lowest rank (or lowest SAP value) are chosen first and the greedy algorithm proceeds to pruning from the lowest ranked nodes till it reaches the error bounds.

3.1.2. An Example of Logic Minimization Through Bit-Flips in Boolean Function Minterms

A more generic and methodical way to reduce the effective switching capacitance of the circuits is through the notion of introducing bit flips in the minterms of boolean...
functions to create inexact logic functions based on work of Ref. [22]. This principle can be harnessed to glean cost gains through literal reduction in the context of inexact circuits while causing an error due to such bit flips. However, not all bit flips of minterms would result in expanding the prime implicant (PI) cubes and some of them might result in negative gains. Hence, it is important to identify the “favorable” bit-flips (or the bit-flips which further minimize the function) and discard the non-favorable ones. To illustrate through an example, Figure 5(a) shows a function (Carry logic) that is widely prevalent in most datapath elements. Assuming that the application would only be able to tolerate at most one bit-flip at this logic function (probability of error = 1/8), Figures 5(b) and 5(c) give an example of favorable 0 to 1 and 1 to 0 bit flips respectively as they minimize the logic function whereas Figure 5(d) shows an unfavorable bit flip leading to an increased logic function complexity. Hence, we can conclude that introduction of favorable bit-flips would lead to further minimization of a logic function owing to the expansion of PI cubes, thereby achieving resource cost gains at the expense of error which is proportional to the number of such bit flips introduced.

As the number of such possible bit-flip combinations grow exponentially in the number of input cases, we take advantage of the notion of assigning significance (similar to the ranking schemes used for pruning), input vector probabilities derived from exercising the benchmarks, node clustering algorithms (to limit the node size between 3–10 inputs/outputs) coupled with greedy local search and finally, the highly symmetric nature of the building blocks/nodes needed for the targeted arithmetic operators.

3.2. Pushing the Energy-Efficiency Envelope Through Confined Voltage Scaling

Conventional physical-layer approaches such as voltage overscaling sought to aggressively scale down the supply voltage and strived to limit the resulting (delay) overhead by not reducing the sampling frequency (thereby making errors).8–10,12 However, imprudent use of such voltage overscaling techniques severely mitigates the gains in a physical realization as a result of unaccounted overheads outlined in Section 2. Hence, it is critical to identify the amount of voltage scaling that would be feasible on a given inexact circuit and in this paper, we propose to identify such a bound on voltage scaling. The proposed confined voltage scaling (CVS) technique reduces the supply voltage such that the hardware implementation overheads in terms of delay and area are lower than the gains achieved through architectural-layer approaches.7 In other words, we intend to trade a small portion of the delay and area savings achieved from architecture and logic-layer techniques for realizing energy gains. The CVS technique differs from the voltage overscaling (VOS) in the following three distinguishing aspects:

- The limit imposed on the amount of voltage scaling that CVS would permit is determined through a function of the delay and area savings that were enabled by the architecture- and logic-layer techniques (for example, probabilistic pruning). These limits can be easily estimated at design time through mathematical modeling (shown below) and simulations. On the other hand, the voltage scaling limit imposed on the VOS techniques is often dictated by the end application as well as the selection of circuit architecture (for example, refer to Ref. [11] that describes the existence of a critical voltage point beyond which error escalates rapidly for certain architectures). Such limits are hard to model and often require elaborate and time-consuming simulations.

- The CVS technique does not introduce any additional errors beyond what was introduced already by the architectural- and logic-layer techniques. This can be attributed to the voltage scaling limit that we set in the CVS technique which makes sure that it does not violate the critical path delay of pruned circuit. Hence, the circuit preserves the error characteristics before and after the application of the CVS technique. On the other hand, VOS techniques use voltage scaling as the primary basis for introducing errors in the circuit by violating the critical path delay. As a result, the circuit could transgress into metastable states and requires metastability-resolution circuits, which come with non-negligible overheads. These overheads are not incurred in the CVS technique.

- When compared to the biased VOS approaches such as Refs. [9, 12], the CVS technique has much lower overheads as the former requires more complicated multiple supply plane routing with level shifters as well as power management circuitry.

We present a rough heuristic to estimate the amount of energy reduction that would be possible through the CVS technique. Here, we will use the commonly accepted fact that supply voltage (V) is inversely proportional to delay.

Fig. 5. An example of probabilistic logic minimization of carry logic in full adder cells. K-maps of (a) initial correct function, (b) function with a favorable 0 to 1 bit flip, (c) function with a favorable 1 to 0 bit flip, (d) function with a non-favorable 0 to 1 bit flip.6,8
3.3. Putting it All Together: Cross-Layer Co-Design Framework

Now, having established the foundational techniques at the two layers of abstractions (probabilistic pruning at the architectural-level and CVS at the physical-level), we describe the CCF framework that spans across both the layers and provides for a symbiotic co-design framework that amplifies the energy efficiency of the targeted circuits. The high-level steps involved in the CCF approach using probabilistic pruning and CVS are shown in Figure 6 and the interested readers can Ref. [7] for a detailed pseudocode. The flowchart in Figure 6 describes the transformations under the broad umbrella of CCF that a circuit goes through from the initial (exact) design to a final highly energy-efficient inexact design. The first part consists of applying the probabilistic pruning technique which can be broken down into three steps as mentioned in Section 3.1—ranking steps which sorts the nodes based on their Significance-Activity Product (SAP) values, pruning step which prunes/deletes the nodes iteratively (can employ a “binary search” manner to reduce the complexity from $O(n)$ to $O(\log n)$) until the error bound is reached and finally, the healing step which reconnects isolated wires/blocks. Once this part is complete, the cost gains form the resulting probabilistically pruned circuit are estimated and used as a basis for applying the confined voltage scaling technique as outlined in Section 3.2. This outputs the final cross-layer optimized energy-parsimonious circuit as shown in Figure 6.

4. EXPERIMENTAL FRAMEWORK, RESULTS AND DISCUSSION

4.1. Proposed Flow for Implementation and Testing of Inexact Circuits

The complete design flow incorporating the CCF framework is shown in Figure 7. The unique blocks in this design flow are the probabilistic pruner and the confined voltage scalar which are developed using the flowchart from Figure 6. The desired circuits are described in a hardware description language (VHDL) and then, synthesized using Cadence RTL Compiler. The Probabilistic Pruning/Logic-Minimization is applied to this synthesized netlist using some custom scripts with the aid of a functional simulator (ModelSim) and application specific benchmarks to narrow down the best pruned or logic minimized circuit. The application-specific benchmarks used in our simulations include uniform random distributions from Matlab (for generic applications), image test vectors from Mediabench23 and audio test vectors from NCH Software.24 The resulting pruned or minimized circuit is re-synthesized to glean further savings (if any) and then, we use the Place and Route Tools (Cadence SoC Encounter) to generate the final layout and the GDSII file (for the foundry). The post-layout analysis of our resulting...
Fig. 7. A design flow for designing and implementing inexact circuits using the CCF framework. Reprinted with permission on Ref. [7], A. Lingamneni et al., Algorithmic methodologies for ultra-efficient inexact architectures for sustaining technology scaling, *In the proceedings of the 9th ACM International Conference on Computing Frontiers (2012)*, pp. 3–12 © 2012. Association for Computing Machinery, Inc.

circuit is done by back-annotation of the final netlist and parasitics using V2LVS from Calibre and using Mentor’s ADMS-ADiT suite to generate accurate and quick SPICE-like estimates. Within this framework, we have custom scripts for the CVS block which facilitates and estimates the circuit characteristics at reduced operating voltages.

To illustrate the technology-independent nature of the logic- and architecture-layer design techniques, we have implemented the inexact circuits in a variety of CMOS technology libraries including TSMC 65 nm (High $V_t$), IBM 90 nm (Normal $V_t$) and TSMC 180 nm (Low Power). All the circuits are operated at the nominal supply voltage of their respective technology nodes as indicated by the foundries: 1.8 V for TSMC 180 nm, 1.2 V for IBM 90 nm and 1.2 V for TSMC 65 nm. Also, the synthesis of designs was done targeting highest frequency of operation and lowest power consumption (or loose target frequency) separately, to analyze the gains achieved in each case.

We have also fabricated a prototype chip in TSMC 180 nm (Low Power) technology to validate the proposed CCF technique on a variety of conventional 64-bit arithmetic adders designs. A photograph of the 86-pin fabricated chip is shown in Figure 8. The different adder designs (a combination of both conventional and pruned) are placed in PD2 power domain while the peripheral circuitry (including pseudo random number generators, register banks etc.) to facilitate the functioning and verification of the adder circuits are present on the outer PD1 power domain. The versatile icyboard® platform is used to build the testing framework for the fabricated inexact circuits. It consists of a programmable MAX-IIG CPLD along with peripheral circuitry such as JTAG/USB ports for enabling the data transfer from the mounted PCB board that houses the prototype chip along with probes for testing and measuring power consumption.

4.2. Results and Analysis

The measured normalized gains (calculated as Conventional/Proposed) from the prototype chip for different metrics such as area, delay, energy, energy-delay product and energy-delay-area product for varying relative error magnitude percentages (same metrics of comparison as Ref. [5]) obtained by applying the probabilistic pruning technique on 64-bit Kogge-Stone adder are shown in Figure 9(a). It is evident from this figure that the pruning technique yields savings across all 3 cost dimensions—with the cumulative gains (quantified through the energy-delay-area product metric) varying from $1_{periodor}7 \times 7 \times$ for varying relative error magnitude percentage values.

For the probabilistic logic minimization approach, we present the results of an array multiplier that used two different input benchmarks—uniform test vectors and images as shown in Figure 10. It can be observed that this technique results in highly resource-efficient datapath elements. For example, the uniform test vectors, the inexact
Fig. 8. (a) Die photograph of the fabricated chip with inexact arithmetic adders. (b) Measurement framework of the chip using CSEM’s icyboard platform. Reprinted with permission on Ref. [7], A. Lingamneni et al., Algorithmic methodologies for ultra-efficient inexact architectures for sustaining technology scaling, *In the proceedings of the 9th ACM International Conference on Computing Frontiers* (2012), pp. 3–12 © 2012. Association for Computing Machinery, Inc.

Fig. 9. Measured results of the Kogge-Stone adders designed through (a) Probabilistic pruning alone, and (b) Cross-layer co-design approach. Reprinted with prior permission on Ref. [7], A. Lingamneni et al., Algorithmic methodologies for ultra-efficient inexact architectures for sustaining technology scaling, *In the proceedings of the 9th ACM International Conference on Computing Frontiers* (2012), pp. 3–12 © 2012. Association for Computing Machinery, Inc.

Array Multiplier (Uniform)

Array Multiplier (Image)

Fig. 10. Normalized gains in different metrics for varying relative error magnitude percentages for an inexact array multiplier on two different input benchmarks.
array multiplier resulted in cumulative EDAP savings of about 7X with a relative error magnitude of less than 6.5% as shown in Figure 10(a) and using application specific test vectors (like audio and image), the savings have increased (upto 8.25X in the case of array multipliers) with comparable error values as shown in Figure 10(b).

However, it can be inferred from Figure 9(a) that energy gains obtained by the probabilistic pruning technique vary only between 1.2–1.6X for acceptable error bounds. Combining the probabilistic pruning technique with the proposed CVS technique under the umbrella of the proposed cross-layer co-design approach has a dramatic increase in the energy gains (upto 4.8X) by trading small amounts of delay gains obtained through probabilistic pruning while maintaining the same error values as shown in Figure 9(b). As a result of this co-design approach, the cumulative gain across all 3 dimensions shoots up as high as 15X! The normalized energy gains obtained by the proposed co-design approach for various types of adders with varying error values are summarized in Figure 11(a), and it is evident that the energy gains could be as high as 4.8X by trading off modest amounts of delay gains.

Fig. 12. Applying the proposed inexact techniques on two multimedia benchmarks—(top) an 8-point inexact FFT and exact inverse FFT computation with varying relative error, and (bottom) frames from an H.264 video codec processed using inexact adders with varying relative error values. Reprinted with permission on Ref. [7], A. Lingamneni et al., Algorithmic methodologies for ultra-efficient inexact architectures for sustaining technology scaling, In the proceedings of the 9th ACM International Conference on Computing Frontiers (2012), pp. 3–12 © 2012. Association for Computing Machinery, Inc.
All the adder designs fabricated using TSMC 180 nm (Low Power) technology library were re-implemented in IBM 90 nm and TSMC 65 nm (high $V_t$, low leakage) technologies as well and the post-layout simulations of the adders establish the technology independence of the probabilistic pruning technique as shown in Figure 11(b) providing significant baseline gains for realizing inexact circuits (even in recent/future technology nodes where the headroom for the gate overdrive voltage might limits the amount of feasible voltage scaling levels).

4.3. Application of Pruned Adders to Images.\textsuperscript{7}

Since metrics such as relative error do not provide an intuitive idea as to how much an inexact architecture affects an application, we will provide some preliminary evidence demonstrating the effect of inexactness on perceptual quality in the context of media processing applications. For this purpose, we will choose the Fast Fourier Transform, one of the most ubiquitous building blocks in such applications. We have simulated an 8-point FFT architecture designed using pruned adders. The adders used in the FFT were simulations of 16-bit pruned adders implemented using a MATLAB environment. Specifically, we collected the mean and standard deviation of relative error magnitudes from post-layout simulations of 16-bit pruned adders, and injected these statistically determined error values into our simulator. The resulting images processed using such inexact FFT architectures with various levels of error are shown in Figure 12 (top half). In addition, we have also included some of the frames processed by a H.264 video codec with inexact arithmetic blocks having varying relative error magnitudes on sample videos from\textsuperscript{26} in the bottom half of Figure 12.

Depending on the perceptual quality needed, the inexact adder with the corresponding error properties could be chosen. From this figure, it can be observed that while images corresponding to a relative error of 0.54% would be suitable for most applications, images with relative error of 7.5% would still allow us to visually recognize the images. Thus, depending on the specific quality needed and after accounting for the compensation provided by our own visual and potentially auditory pathways, inexact architectures could be used at relatively high levels of admissible error indeed!

5. CONCLUSION AND FUTURE DIRECTIONS

We have briefly reviewed our work on inexact design approaches on designs of energy-efficient arithmetic operators with a focus on highlighting the benefits of using a cross-layer co-design approach integrating the architecture/logic-level techniques such as probabilistic pruning that guarantees a technology-independent gains across all 3 aspects—energy, delay and area, with the physical-level approaches such as confined voltage scaling technique which trades some portion of these delay and/or area gains to garner dramatic energy savings. Some other future directions of research that can be pursued are outlined below:\textsuperscript{7}

- **Mathematical framework for developing optimization models**: While most inexact techniques outlined in this paper concentrated on proposing heuristics to demonstrate the benefits that could be achieved the accuracy trade-off, we envision more efficient circuits that could be realized through a mathematical optimization framework spanning over all the layers of abstraction.

- **Compilers and Tools for Hardware/Software Co-design**: There is a need for developing a hardware/software co-design framework for efficiently mapping the inexact parts of an application algorithm into corresponding inexact hardware. Efficient software compilers and design automation tools for realizing such a mapping (both statically and at runtime) would be of significant interest as would be the case with research on efficient underlying inexact hardware.

- **Verification and Test of Inexact Systems**: As inexact systems redefine the notion of “correctness” associated with their outputs, conventional verification and test algorithms and their evaluation metrics have to be revisited and modified for inexact circuits.

- **Mixed-Signal Inexact SoC Design**: While most of the current research in inexact design has focused on digital (logic) systems and memories, there hasn’t been any integration of analog components (which generally are inexact by nature) along with these inexact circuits. A mixed-signal SoC integrating all the 3 fabrics—digital logic, memories and analog circuits—along with appropriate system level design techniques would lead to more cost-efficient IC design for emerging error resilient applications.

Acknowledgments: The authors would like to gratefully acknowledge the efforts of their collaborators Richard M Karp and Kirthi Krishna Muntimadugu for their discussions and contributions to the conference paper,\textsuperscript{7} which shaped portions of this paper. The authors would also express their sincere thanks to Jean-Luc Nagel, Pierre-Alain Beuchat and Marc Morgan of CSEM SA for their help with the chip tapeout and measurements. We also extend our gratitude to all of the reviewers and editors for their painstaking efforts to help improve the quality of our paper.

References

Designing Energy-Efficient Arithmetic Operators Using Inexact Computing

Avinash Lingamneni

Avinash Lingamneni is currently a Ph.D. student in Electrical & Computer Engineering (ECE) at Rice University, USA and is working towards his Ph.D. at the Swiss Center for Electronics and Microtechnology (CSEM SA), Switzerland. He received his M.S. degree in ECE from Rice University in May 2011 and B.Tech. (honors) in ECE from IIT Hyderabad, India in 2008. His current research focuses on designing low-power VLSI hardware systems for error-resilient applications, in particular Digital Signal Processing, Wireless Communications and Neuromorphic workloads, through Probabilistic and Inexact/Approximate circuit designs. He has co-authored over 20 refereed publications and received the Best Paper Award at the ACM International Conference on Computing Frontiers in 2012.

Christian C. Enz

Christian C. Enz received his Ph.D. from the Swiss Federal Institute of Technology (EPFL) in 1989. He is currently VP heading the Integrated and Wireless Systems Division of the Swiss Center for Electronics and Microtechnology (CSEM) in Neuchâtel, Switzerland. He is also Professor at EPFL, where he is lecturing and supervising undergraduate and graduate students in the field of analog and RF IC design. Prior to joining the CSEM, he was Principal Senior Engineer at Conexant (formerly Rockwell Semiconductor Systems), Newport Beach, CA, where he was responsible for the modeling and characterization of MOS transistors for RF applications. His technical interests and expertise are in the field of ultralow-power analog and RF IC design, wireless sensor networks and semiconductor device modeling. Together with E. Vittoz and F. Krummenacher he is the developer of the EKV MOS transistor model and the author of the book “Charge-Based MOS Transistor Modeling—The EKV Model for Low-Power and RF IC Design.”
(Wiley, 2006). He is the author and co-author of more than 150 scientific papers and has contributed to numerous conference presentations and advanced engineering courses.

Krishna V. Palem
Krishna V. Palem is the Ken and Audrey Kennedy Professor at Rice University with appointments in Computer Science, in Electrical and Computer Engineering and in Statistics. He founded and directs the NTU-Rice Institute on Sustainable and Applied Infodynamics (ISAID) at the Rice University, where he is a scholar in the Baker Institute for Public Policy, and at the Nanyang Technological University (NTU) in Singapore where he is a Canon Visiting Professor in the School of EEE of NTU. His related work on the foundations of architecture assembly for designing reconfigurable embedded SoC architectures, developed at Proceler Inc. which he co-founded and served as a CTO, was a nominee for the Analysts Choice Awards as one of the (four) outstanding technologies of 2002. He has pioneered a novel technology entitled Probabilistic CMOS (PCMOS) for enabling ultra low-energy embedded computing which was recognized as one of the ten technologies “likely to change the way we live” by MIT’s Technology Review, as one of the seven “emerging world changing technologies” by IEEE. He is a Fellow of the IEEE and of the ACM. He is the recipient of the 2008 W. Wallace McDowell Award, IEEE Computer Society’s highest technical award and one of computing’s most prestigious individual honors.

Christian Piguet
Christian Piguet received his M.S. and Ph.D. degrees in electrical engineering from the Ecole Polytechnique Fédérale de Lausanne (EPFL), respectively in 1974 and 1981. He joined the Centre Electronique Horloger S.A., Neuchâtel, Switzerland, in 1974. He is now Scientific Coordinator at the CSEM Centre Suisse d’Electronique et de Microtechnique S.A. He is presently involved in the design of low-power low-voltage integrated circuits in CMOS technology, including design methodologies, microprocessor architectures and logic design. He is Professor at the Ecole Polytechnique Fédérale Lausanne (EPFL), Switzerland, lectures in VLSI design in the ALaRI Master at the University of Lugano, Switzerland. He was “Distinguished Lecturer” in IEEE CAS for 2004–2005. He holds about 30 patents in digital design, microprocessors and watch systems and co-authored of more than 200 publications in technical journals and of books and book chapters in low-power digital design.