

Energy Parsimonious Circuit Design through *Probabilistic Pruning*

Avinash Lingamneni[†], Christian Enz^{*}, Jean-Luc Nagel^{*}, Krishna Palem[†] and Christian Piguet^{*}

^{*} Centre Suisse d'Electronique et de Microtechnique (CSEM) SA
Jaquet- Droz 1, Neuchatel, Switzerland

[†]NTU-Rice Institute for Sustainable and Applied Infodynamics (ISAID)
Department of Electrical and Computer Engineering
Rice University, Houston, USA
Email: avinash.l@rice.edu, palem@rice.edu

Abstract—*Inexact* Circuits or circuits in which the accuracy of the output can be traded for energy or delay savings, have been receiving increasing attention of late due to invariable inaccuracies in designs as Moore's law approaches the low nanometer range, and a concomitant growing desire for ultra low energy systems. In this paper, we present a novel design-level technique called *probabilistic pruning* to realize inexact circuits. Unlike the previous techniques in literature which relied mostly on some form of scaling of operational parameters such as the supply voltage (V_{dd}) to achieve energy and accuracy tradeoffs, our technique uses pruning of portions of circuits having a lower probability of being active, as the basis for performing architectural modifications resulting in significant savings in energy, delay and area. Our approach yields more savings when compared to any of the conventional voltage scaling schemes, for similar error values. Extensive simulations using this pruning technique in a novel logic synthesis based CAD framework on various architectures of 64-bit adders demonstrate that normalized gains as great as 2X-7.5X in the Energy-Delay-Area product can be obtained, with a relative error percentage as low as $10^{-6}\%$ up to 10%, when compared to corresponding conventionally correct designs.

I. INTRODUCTION

Realizing reliable computations from unreliable components has long been a focus of study [1] and is receiving greater than ever prominence today [2] as diminishing transistor sizes driven by Moore's law are leading to increasing process variations. It is due to these process variations, arising as lithographic scaling lags behind device scaling, and the quest for ultra-low energy circuits, particularly in the domain of embedded systems, that *exact* computing, in which output of the desired circuit is precise, is yielding the way to *inexact* computing, wherein accuracy of the output of circuits can be traded in for significant savings in energy and/or delay parameters pioneered in [3] and subsequently used in [4], [5].

Almost all of the previous papers in literature realizing *inexact* circuits used some form of voltage scaling as the basis for reliability and energy tradeoffs [6], [4], [5]. In this

paper, we introduce a novel design level technique called *Probabilistic Pruning* to realize inexact circuits which focuses on architectural modifications to circuits based on pruning of the circuit elements having a lower probability of being active during operation. The amount of such pruning will be dictated by the application's error tolerance, quantified through a metric.

The major contributions of this paper are summarized below:

- We propose a novel design-level technique called *Probabilistic Pruning* to realize Inexact circuits along with the corresponding mathematical characterization in Section III.
- We propose a logic synthesis based CAD framework in Section V incorporating the *Probabilistic Pruning technique* to help design inexact circuits achieving a faster time to design and fabricate than a conventional design methodology needed by previous approaches in literature ([7], [6]).
- We show through extensive simulations that the significant savings in energy, delay and area obtained by the pruning technique in the context of 64-bit arithmetic adders in Section V. Our experiments demonstrate about 2X-7.5X savings in energy-delay-area product, with corresponding relative error percentage of $10^{-6}\%$ -10% while using weighted significance values, and about 2X-5.3X savings in energy-delay-area product with corresponding error rates of 0.01%-37% using uniform significance values.
- We infer some general principles for the application of *probabilistic pruning* technique to any general circuit beyond adders, and provide some future directions for the design of inexact circuits in Section V and VI.

II. RELATED WORK

Several papers in the past have investigated ways of overcoming the challenges to process and parameter variations

threatening the sustained evolution of Moore's law. Some of the prominent methods include using multicore architectures to increase parallelism without frequency/voltage scaling, designing for average case operation and using temporal and/or spatial redundancy to correct worst-case errors [8], [9]. Exciting new research into novel materials for realizing circuits such as optoelectronics, memristors [10] and molecular electronics have also been investigated successfully. One common principle in all of these approaches is to ensure that the device always functions correctly, either by design, or through an error-correction mechanism.

In a radical departure from these conventional approaches, it was shown in [3] that error can be traded as a commodity as opposed to being viewed as an impediment to glean significant savings (typically energy) – in applications that can accommodate error. Fortunately, a large class of emerging applications, particularly in the domain of embedded and mobile systems, can tolerate varying amounts of errors, more so when it results in significant energy savings. A CMOS realization of this principle called PCMOS was given in [11] and was later, extended to realize a system level application through an SoC architecture [12]. Also, application of this principle in the context of DSP applications was realized using *probabilistic* arithmetic [7] where the errors were caused by (thermal) noise present in ultra-deep submicron technologies, and through *approximate* arithmetic in [6] where the errors were deterministically introduced through critical path violation as a result of voltage scaling. This principle was applied more recently to processor modules through voltage scaling and slack redistribution in [4], [5].

However, while this unconventional principle opens up several novel directions for trading off accuracy for savings, there are serious impediments when one considers integrated circuits based on this principle. For example, one physical realization referred to as Biased Voltage Scaling (BiVOS) [7], [6] is seriously impeded since it involves significant overheads of routing multiple voltage planes, and by necessity for level shifters for parallel structures in which routing exists between all voltage planes. Another important drawback of all previous voltage scaling based works [4], [5], [7], [13] is that the fine-tuning of supply voltage at run-time based on the application requirements might not be feasible due to inherent variations present in the power supply routing [14] and by the large overhead generally required to ensure such an accurate fine-tuning is realized.

In this paper, we overcome these drawbacks while exploiting the principle of trading accuracy for significant energy savings through a architectural redesign technique called *probabilistic pruning*, which also yields improvements in area as well as speed. Finally, we note that this technique yields savings which are significantly more and realized with *zero* overhead, when compared to the previously proposed voltage scaling-based techniques which have associated overheads. The term *inexact* design coined in this paper is an umbrella term for the previously proposed *probabilistic* [7], *approximate* [6] and *stochastic* [4] designs or in general, any design in which

accuracy of the output can be traded off for energy, performance and/or area benefits.

III. PROBABILISTIC PRUNING FOR ENERGY/DELAY/ERROR TRADEOFF IN INEXACT CIRCUITS

Probabilistic Pruning is a design level technique wherein we systematically “prune” or delete components and their associated wires along the paths of the circuit that have a lower probability of being active during circuit operation while staying within the error boundaries dictated by the application.

A. A Formal Definition of Probabilistic Pruning

A circuit can be represented as a *directed acyclic graph* whose nodes are components such as gates, inputs, or outputs and whose edges are wires. Given a circuit \mathcal{G} with N_C components, N_I inputs, N_O outputs and N_W wires, our goal is to prune components in the paths such that the energy, area and speed are reduced while maintaining a bound on error, say σ . Let \mathbf{I} be the set of all input nodes, \mathbf{O} be the set of output nodes, \mathbf{C} be the set of all components and \mathbf{W} be the set of all wires.

We now formulate an optimization problem of computing a circuit \mathcal{G}' , which is a subgraph of \mathcal{G} such that it has the same set of inputs $\{\mathbf{I}\}$ and outputs $\{\mathbf{O}\}$ but with components $\{\mathbf{C}'\}$ where $N_{C'} \leq N_C$ and wires $\{\mathbf{W}'\}$ where $N_{W'} \leq N_W$ such that given \mathcal{V} randomly chosen inputs, the average error

$$\text{Er}(\mathcal{G}') = \sum_{k=1}^{\mathcal{V}} p_k \times |\mathcal{O}'_k - \mathcal{O}_k| \leq \sigma \quad (1)$$

where \mathcal{O}_k and \mathcal{O}'_k correspond to value of final output vectors $\langle \mathcal{O}_{k,1}, \mathcal{O}_{k,2}, \dots, \mathcal{O}_{k,n} \rangle$ and $\langle \mathcal{O}'_{k,1}, \mathcal{O}'_{k,2}, \dots, \mathcal{O}'_{k,n} \rangle$ of circuits \mathcal{G} and \mathcal{G}' respectively for a given n -bit input vector \mathcal{I}_k which occurs with a probability p_k for $1 \leq k \leq \mathcal{V}$. In the unweighted case, $|\mathcal{O}_i - \mathcal{O}_j|$ is the value of the difference between \mathcal{O}_i and \mathcal{O}_j treated as unary numbers. In the case where the output bits are weighted, without loss of generality, we could assign a weight η_j to the j^{th} output bit \mathcal{O}_j . In fact, starting with Section IV, we will be demonstrating the value of probabilistic pruning through circuits for integer addition and therefore, will be considering the case of weights $\eta_j = 2^j$ almost exclusively.

Output: A *pruned* \mathcal{G}' that is optimal in that there is no other \mathcal{G}'' satisfying the conditions above such that $N_{C''} < N_{C'}$.

The average error computation metric used above is not limiting in any sense that it can conveniently be replaced by any other error metric based on the application requirements in using the probabilistic pruning approach. In circuit design, it is considered to be meaningful to evaluate a design using a range of inputs drawn from a distribution and analyze error following approaches to average case analysis [15], and we will adopt this approach. Not surprisingly, it is easy to show that variants of this problem are NP-hard in general. However, our main goal in this paper is to demonstrate the value of applying probabilistic pruning to circuit design. Therefore, we will not emphasize the algorithmic nuances in this paper, but will rather use a simple-minded and (almost) brute force

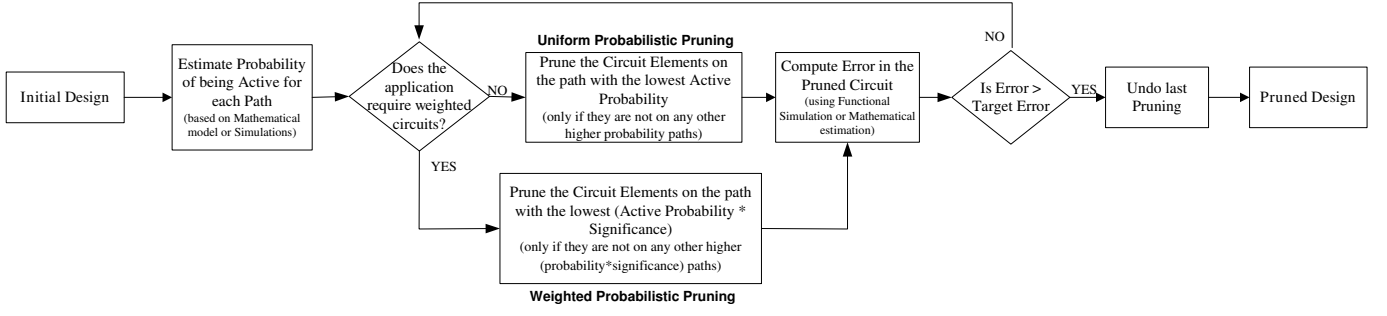


Fig. 1. Flowchart for the Probabilistic Pruning

heuristic here, which is shown in Figure 1. The flowchart is detailed enough to where a pseudo-code for the heuristic can be easily developed using it as a basis.

B. Defining Error

We can broadly classify error resilient applications into two types : ones which have a bound on the total number of erroneous computations (such as number of incorrect memory address computations in a microprocessor) and others (such as computation of the value of a pixel by a graphics processor) which have bounds on the magnitude of error. While in the former type applications, each of the output receives equal importance or “significance” and errors are quantified through the error rate metric, the outputs in the latter applications have a certain importance or *weights* depending on the magnitude of error. We will state each in turn for convenience given \mathcal{V} .

$$\text{Error Rate} = \frac{\text{Number of Erroneous Computations}}{\text{Total Number of Computations}} = \frac{\mathcal{V}'}{\mathcal{V}}$$

$$\text{Relative Error Magnitude} = \frac{1}{\mathcal{V}} \sum_{k=1}^{\mathcal{V}} \frac{|\mathcal{O}_k - \mathcal{O}'_k|}{\mathcal{O}_k}$$

IV. APPLICATION OF PROBABILISTIC PRUNING TO ARITHMETIC CIRCUIT VIA ADDERS

We choose binary adders as the first platform for the application of our *probabilistic pruning* technique as their well-understood yet non-trivial architectures, ranging from serial to highly parallel [16], [17], present us with valuable insights into application of the proposed technique on larger and more complex circuits as outlined in Section V-B.

A. Conventional Adders Designs

It is widely known that binary addition can be formulated as a prefix dependent function i.e. every output is dependent on all inputs of equal or lower magnitude, and every input influences all outputs of equal or higher magnitude. Based on this property, the functioning of a binary adder can be grouped into 3 stages: Precomputation, Prefix and Postcomputation. While the precomputation and postcomputation stages are similar in almost all adder architectures, the prefix computation generally defines the architecture of an adder and is classified into 3 types [16] : (a) Serial Prefix (b) Group Prefix (c) Parallel Prefix depending on the order of grouping of bits and carry

TABLE I
SUMMARY OF VARIOUS CONVENTIONAL ADDER ARCHITECTURE CHARACTERISTICS

Type of Adder	Area	Speed	Power	Prefix Type
Ripple Carry	Lowest	Lowest	Lowest	Serial
Carry Skip	Low	Low	Low	—
Carry Select	Medium	Medium	Medium	Group
Carry Increment	Medium	Medium	Medium	Group
Sklansky	High	Highest	High	Parallel
Brent-Kung	High	High	High	Parallel
Kogge-Stone	Highest	Highest	Highest	Parallel
Han-Carlson	High	Highest	High	Parallel
Ladner-Fischer	High	Highest	High	Parallel
Sparse Tree	High	Highest	High	Parallel

propagation. The prefix networks of some common adders are shown in Figure 2. A brief overview of the characteristics of various conventional adders is given in Table I where we show the type of prefix structure and also contrast the relative area, speed and power intuitively.

B. A Brief Overview of Carry Path Probabilities in an Adder

For notational convenience, we will use the symbols S , A and B to denote the Sum (output) and the two binary inputs to the adder. As all the paths between output S_i and an input A_j or B_j , ($\forall j \neq i$ and $0 \leq j \leq N$) in existing in an N bit adder are due to the propagation of carry bits, we compute the various path probabilities in an adder using a variation of the carry path propagation results derived in [18], to form the basis of the *pruning* technique.

A bit position ‘ i ’ is said to *generate* a carry if both A_i and B_i are equal to 1 and *propagate* a carry if exactly one of A_i or B_i is equal to 1. Hence, a sum output S_i is affected by an input A_j or B_j (where $j < i$) only if there is a carry *generated* at j and the rest of the $i - j$ bits *propagate* the carry. For example, if the summands A and B are chosen uniformly at random, the probability that a bit position j *generates* a carry is $1/4$ and the probability that the rest of the $i - j - 1$ *propagate* the carry is $1/2^{i-j-1}$. Hence, the probability of any particular path from an input A_j or B_j to an output Sum S_i being active is $1/2^{i-j+1}$.

C. Designing Adders using Probabilistic Pruning

Based on the application requirement, we classify the *probabilistic pruning* technique into two forms : *uniform*

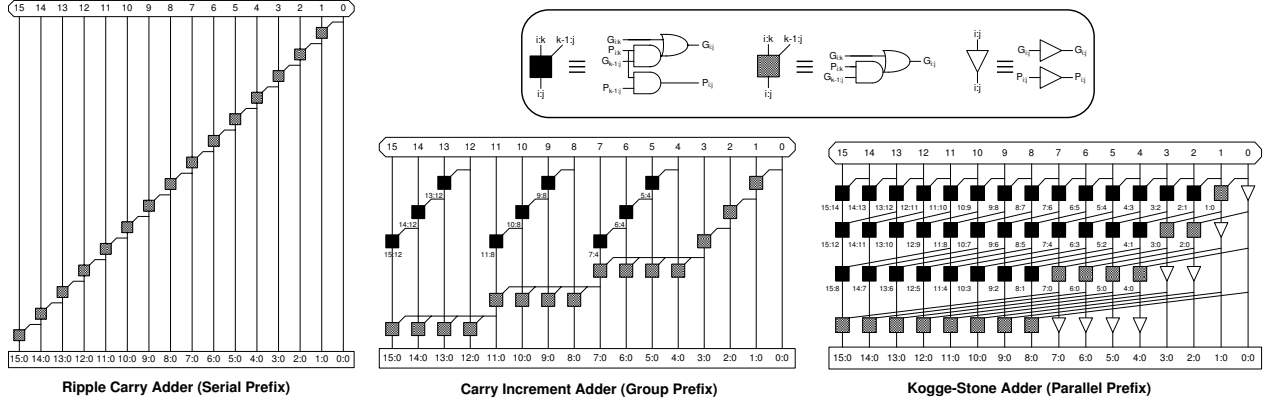


Fig. 2. Prefix Networks of Some Adders

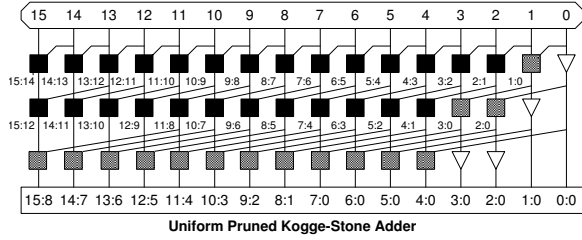


Fig. 3. Example of adder architecture designed using Uniform Probabilistic Pruning

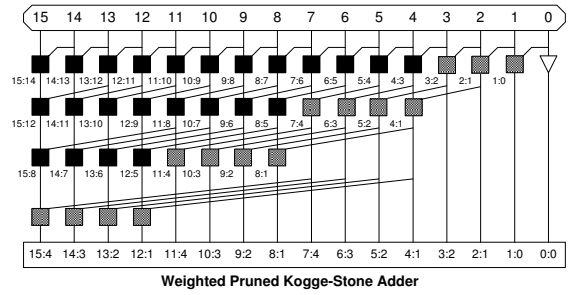


Fig. 4. Example of adder architecture designed using Weighted Probabilistic Pruning

probabilistic pruning (UPP) for applications which require uniform circuits and *weighted probabilistic pruning* (WPP) for applications which require circuits whose outputs have associated weights. We apply the UPP and WPP techniques on adders as follows:

1) *Uniform Probabilistic Pruning based Adders*: Applications for which all bit positions in an adder are treated with equal significance i.e. there is no notion of *Most Significant Bit* (MSB) or *Least Significant Bit* (LSB) and each bit position has equal significance. We use concepts from Section IV-B to calculate the path probabilities in each adder and apply the UPP technique as shown in Figure 1. Due to regular structure of the prefix networks, all components at the same level (or row) are on the paths with equal probability of being active. For example, the components on the 4th level of the Kogge-stone adder are propagating carry information from inputs A_i and B_i to output S_{i+8} and while the components on the 3rd level are propagating the carry information from inputs A_i and B_i to output S_{i+4} , the path probabilities of which are $1/2^9$ and $1/2^5$ respectively. Hence, we start pruning from the 4th level till we reach an error bound of the application. Example of a 16-bit Uniform Pruned Kogge-Stone is shown in Figure 3.

2) *Weighted Probabilistic Pruning based Adders*: Starting from the LSB and moving towards the MSB, each bit position has a significance of 2 times higher than the previous bit position. While it is possible to use the different significance values for each bit position through the WPP technique, in this work for illustrative purposes, we “bin” the bits into four

equal groups with each bin having k consecutive bits. We assume that the bits in the each bin have the same significance and are 2^k times as significant as those in the bin that is immediately following it. Applying our WPP technique, we compute the (significance*path probability) product and prune the components with the least product value. Example of a resulting 16-bit Pruned Kogge-Stone is shown in Figure 4.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Proposed Logic Synthesis based CAD Flow

The proposed logic synthesis CAD flow for realizing *Inexact* circuits through *Probabilistic Pruning* is shown in Figure 5. The main object of interest in the proposed CAD flow is the *Probabilistic Pruner*, which prunes the portions of designs with lower probability of being active and/or with the least significance depending on the application requirement. The overall flowchart summarizing the Probabilistic Pruner is shown in Figure 1 where the pruner interacts with Modelsim based functional simulator and an Error estimator to decide the amount of pruning governed by the error margins dictated by the application. We design the 64-bit adder circuits in VHDL which is sent to the *Probabilistic Pruner* where individual path probabilities are calculated and depending on the error tolerance of the application, circuit elements on paths with lower probabilities and/or lower significance will be pruned. All the designs are implemented using TSMC 180nm (Low Power)

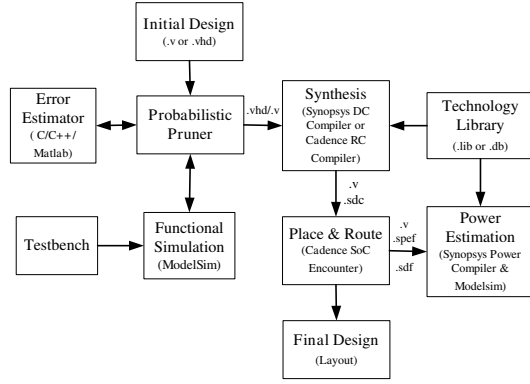


Fig. 5. A Logic Synthesis based CAD flow for designing Inexact Circuits

technology library, and to crosscheck the gains obtained, some of the designs were re-implemented in IBM 90nm (Normal V_t) and TSMC 65nm (High V_t) technologies as well. 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.

B. Results and Analysis

The central results, namely the normalized gains (Conventional/Pruned) for different metrics (area, delay, energy, energy-delay product and energy-delay-area product) obtained by applying WPP and UPP techniques on various 64-bit adders are summarized in Figure 6 and Figure 7 respectively. Due to space constraints, we only show parallel adders with the highest (Kogge-Stone, Han-Carlson) and least (Brent-Kung) amount of gains. From these graphs, it can be observed that: *The (normalized) gains obtained for the applications employing weighted circuits is more than the corresponding uniform circuits for the same error percentage.*

While the normalized gains outlined for parallel adders in Figures 6, 7 are as high as 2X for less than 1% error rate or $10^{-6}\%$ relative error, it is not possible to achieve such high savings in the context of serial adders. For example, WPP technique applied on a Ripple Carry Adder yielded savings of 1.25X, 1.5X and 2X for a high relative error of 19%, 33% and 56% respectively. Hence, *Probabilistic Pruning yields much more significant gains in parallel circuits (with large number of available paths to prune) than the corresponding serial circuits (which generally have very few paths).*

Figure 8 outlines the results obtained for a pruned Kogge-Stone adder using three different technology libraries under two different synthesis constraints. From this, we can conclude that: *For similar operating conditions, the gains achieved in the probabilistic pruned circuits is proportional to the ratio of circuit pruned to the original circuit. It is largely independent on the process technology being used and only depends on the logic synthesis constraints.*

We can also observe that probabilistic pruning is a design level technique which does not involve varying of circuit

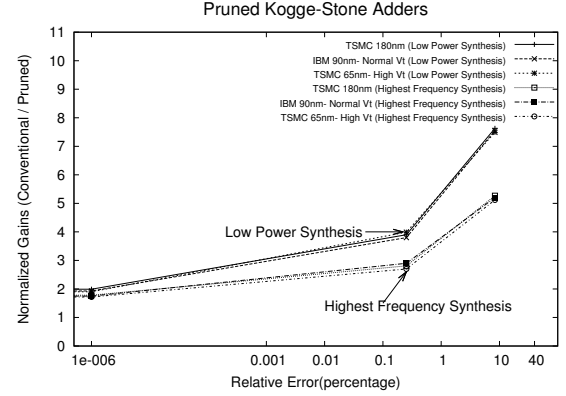


Fig. 8. Energy-Delay-Area Product of Weighted Pruned Kogge-Stone adders implemented in different process technologies and synthesis constraints

parameters during operation, *the amount of error in a probabilistic pruned circuit is independent of varying of parameters (such as V_{dd}) unlike other inexact circuits and is as robust as conventional circuits to process variations.* The amount of such error is generally fixed at design time based on application requirements. The probabilistic pruning technique can be used in conjunction with techniques such as adaptive body bias to address the effects of parameter variations in the more significant portions of the circuits.

Another observation regarding the probabilistic pruned circuits is that *the error (both error rate and relative error magnitude) in probabilistic pruned adders rises sharply beyond a critical amount of pruning akin to the critical voltage scaling point problem mentioned in [5].* We anticipate that this can be fixed by combining a parameter variation approach (such as V_{dd} or V_t variation techniques) with our pruning technique.

VI. CONCLUSION AND FUTURE DIRECTIONS

To the best of our knowledge, this is the first attempt at an architectural design of circuits that explicitly trades error for savings. We believe to have convincingly shown through extensive simulations, that our *Probabilistic Pruning* technique achieves significantly better savings along all 3 dimensions – energy, delay and area – while having comparable error, over the conventional voltage scaling schemes. Other benefits include the lack of dependence of amount of error on any particular process technology and faster design time than conventional voltage scaling schemes as a result of easy integration into a logic synthesis based CAD flow.

These gains achieved through the *Probabilistic Pruning* are *relative* in that they can be combined with standard techniques that achieve energy or performance gains or both, through *absolute* approaches. Specifically, this means that any technique that uses equal voltage (planes) throughout the datapath and yields correct results or voltage scaling to yield tolerable incorrect results can be extended through the insights in this paper to yield *additional* gains simultaneously along the energy, delay and area dimensions by using the *probabilistic pruning* technique.

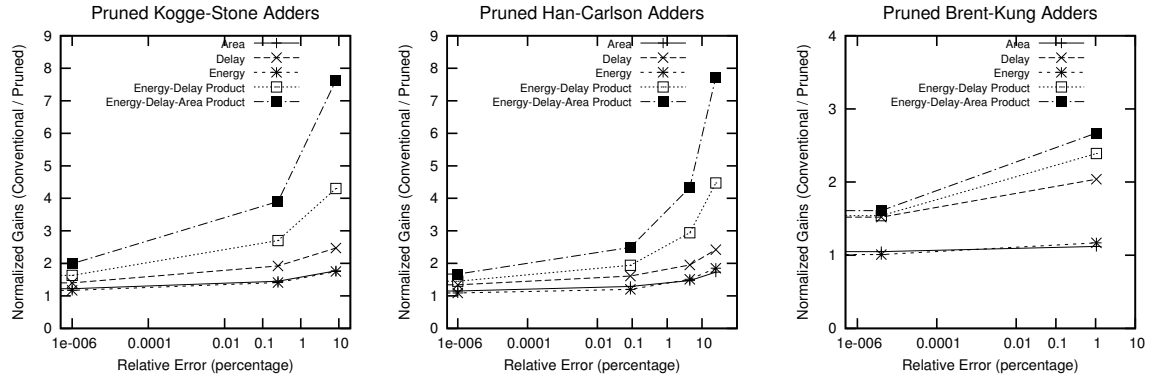


Fig. 6. Normalized gains Vs Relative Error percentage of various Weighted Pruned 64-bit adders implemented in TSMC 180nm technology

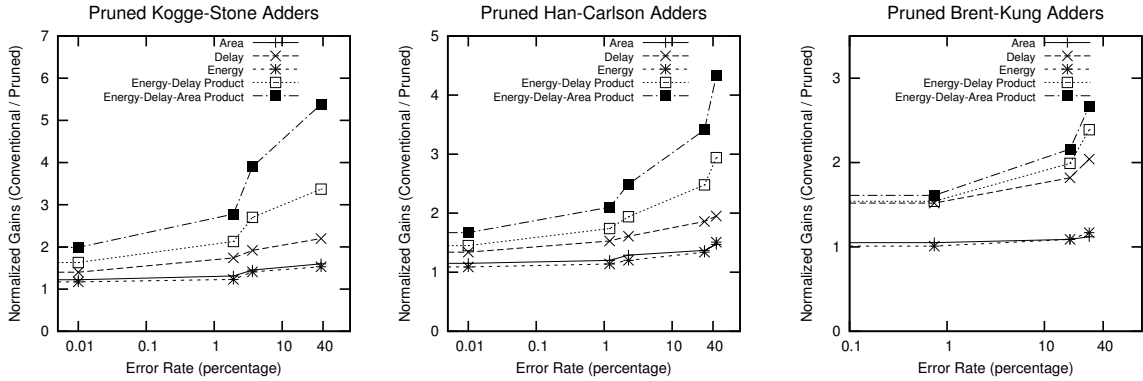


Fig. 7. Normalized gains Vs Error Rate percentage of various Uniform Pruned 64-bit adders implemented in TSMC 180nm technology

REFERENCES

- [1] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies (C.E. Shannon and J. McCarthy eds.)*, Princeton Univ. Press, Princeton, N.J., 1956.
- [2] S. Borkar, "Designing reliable systems from unreliable components: The challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10–16, 2005.
- [3] K. V. Palem, "Energy aware algorithm design via probabilistic computing: from algorithms and models to Moore's law and novel (semiconductor) devices," in *Proc. of the IEEE/ACM Intl. Conf. on Compilers, Architecture and Synthesis for Embedded Systems*, 2003, pp. 113–117.
- [4] S. Narayanan, J. Sartori, R. Kumar, and D. Jones, "Scalable stochastic processors," in *Proc. of the Design, Automation and Test in Europe*, March 2010.
- [5] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," in *Proc. of 15th IEEE/SIGDA Asia and South Pacific Design and Automation conference*, January 2010.
- [6] L. N. B. Chakrapani, K. K. Muntimadugu, A. Lingamneni, J. George, and K. V. Palem, "Highly energy and performance efficient embedded computing through approximately correct arithmetic: A mathematical foundation and preliminary experimental validation," in *Proc. of the IEEE/ACM Intl. Conf. on Compilers, Architecture, and Synthesis of Embedded Systems*, 2008.
- [7] J. George, B. Marr, B. E. S. Akgul, and K. Palem, "Probabilistic arithmetic and energy efficient embedded signal processing," in *Proc. of the IEEE/ACM Intl. Conf. on Compilers, Architecture, and Synthesis for Embedded Systems*, 2006, pp. 158–168.
- [8] J. Ray, J. C. Hoe, and B. Falsafi, "Dual use of superscalar datapath for transient-fault detection and recovery," in *Proceedings of the 34th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2001, pp. 214–224.
- [9] D. Ernst, N. S. Kim, S. Das, S. Pant, T. Pham, R. Rao, C. Ziesler, D. Blaauw, T. Austin, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. of the 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct. 2003, pp. 7–18.
- [10] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, March.
- [11] S. Cheemalavagu, P. Korkmaz, and K. V. Palem, "Ultra low-energy computing via probabilistic algorithms and devices: CMOS device primitives and the energy-probability relationship," in *Proc. of the Intl. Conference on Solid State Devices and Materials*, Sep. 2004, pp. 402–403.
- [12] L. N. Chakrapani, B. E. S. Akgul, S. Cheemalavagu, P. Korkmaz, K. V. Palem, and B. Seshasayee, "Ultra efficient embedded SoC architectures based on probabilistic CMOS technology," in *Proc. of the 9th Design Automation and Test in Europe*, Mar. 2006, pp. 1110–1115.
- [13] N. Banerjee, G. Karakonstantis, and K. Roy, "Process variation tolerant low power DCT architecture," in *Proceedings of Design, Automation and Test in Europe Conference*, Apr. 2007, pp. 1–6.
- [14] M. Alioto and G. Palumbo, "Impact of supply voltage variations on full adder delay: analysis and comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 12, pp. 1322 – 1335, 2006.
- [15] R. Karp, "Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane," *Mathematics of Operations Research*, vol. 2, no. 3, pp. 209–224, 1977.
- [16] R. Zimmerman, "Binary adder architectures for cell-based vlsi and their synthesis," Ph.D. dissertation, Swiss Federal Institute of Technology, 1997.
- [17] D. Harris, "A taxonomy of parallel prefix networks," vol. 2, Nov 2003, pp. 2213 – 2217.
- [18] N. Pippenger, "Analysis of carry propagation in addition: An elementary approach," *Journal of Algorithms*, vol. 42, pp. 317–313, 2002.