Energy Parsimonious Circuit Design through

Probabilistic Pruning

Christian Enz*, Avinash Lingamneni†, Jean-Luc Nagel*, Krishna Palem† and Christian Piguet*

* Centre Suisse d’Electronique et de Microtechnique (CSEM) SA,
Jaquet- Droz 1, Neuchatel, Switzerland
Email:Christian.ENZ@csem.ch, Jean-Luc.NAGEL@csem.ch, christian.piguet@csem.ch

†NTU-Rice Institute for Sustainable and Applied Infodynamics (ISAID)
50 Nanyang Avenue, Singapore 639798
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 computational components from unreliable sources 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. This technique 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 term inexact design coined in this paper is an umbrella term for the previously proposed probabilistic [7], approximate [8] 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.

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 III 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], [8]).
- We show through extensive simulations in the proposed CAD framework that the potential savings in energy, delay and area obtained by the pruning technique in the context of 64-bit arithmetic adders can be significant, in Section IV. 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 motivated by binary representation of numbers, and about 2X-5.3X savings in energy-delay-area product with corresponding error rates of 0.01%-37% using an uniform significance values where the outputs are interpreted like unary numbers.
- 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 [9], [10], [11]. Exciting new research into novel materials for realizing circuits such as optoelectronics, memristors [12] and molecular electronics [13], [14] have also been investigated successfully. The question is of such great significance to our daily lives that even the articles in New York Times are actively discussing these issues [15]. 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], [16] 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 [17] and was later, extended to realize a system level application through an SoC architecture [18]. 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 [8], [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], [8] 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], [19] 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 [20] 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.

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 $G$ 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 $\sigma$. Let $I$ be the set of all input nodes, $O$ be the set of output nodes, $C$ be the set of all components and $W$ be the set of all wires.

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

$$\text{Er}(G') = \sum_{k=1}^{V} p_k \times |O_k' - O_k| \leq \sigma$$  \hspace{1cm} (1)$$

where $O_k$ and $O_k'$ correspond to value of final output vectors $< O_{k,1}, O_{k,2}, \ldots, O_{k,n} >$ and $< O_{k,1}', O_{k,2}', \ldots, O_{k,n}' >$ of circuits $G$ and $G'$ respectively for a given $n$-bit input vector $I_k$ which occurs with a probability $p_k$ for $1 \leq k \leq V$. In the unweighted case, $|O_k - O_k'|$ is the value of the difference between $O_k$ and $O_k$ treated as unary numbers. In the case where the output bits are weighted, without loss of generality, we could assign a weight $\eta_j$ to the $j^{th}$ output bit $O_j$. In this case, $O_k$ and $O_k'$ will be the difference between the corresponding binary numbers. 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 $G'$ that is optimal in that there is no other $G''$ satisfying the conditions above such that $N_W'' < N_W'$. The average error computation metric used above is not limiting in any sense that it can conveniently 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 [21], and we will adopt this approach. Not surprisingly, it is easy to show that variants of this problem are NP-hard in general [22], [23]. 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 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 bounds on the magnitude of error. We will state each in turn for convenience given $V$.

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

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

C. Proposed Logic Synthesis based CAD Flow

The proposed logic synthesis CAD flow for realizing inexact circuits through Probabilistic Pruning is shown in Figure 2. 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
IV. APPLICATION OF PROBABILISTIC PRUNING TO ARITHMETIC CIRCUIT VIA ADDERS

Binary addition is a fundamental and most frequently used arithmetic operation on microprocessors, digital signal processors and other data-processing ASICs. Numerous algorithms and circuit architectures for binary addition have been proposed over the last 50 years and the usability of these architectures based on the specific constraints (delay/area/fanout/wiring tracks) has been extensively studied [24], [25]. 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, 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 [24] : (a) Serial Prefix (b) Group Prefix (c) Parallel Prefix depending on the order of grouping of bits and carry propagation. The prefix networks of some common adders are shown in Figure 3. 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 $B_j$, $(v, j \neq i$ 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 [26], 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 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:

![Flowchart for the Probabilistic Pruning](image)

![Logic Synthesis based CAD flow for designing Inexact Circuits](image)

![Summary of Various Conventional Adder Architecture Characteristics](image)

![Table I](image)
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. Examples of 16-bit Uniform Pruned Kogge-Stone and Brent-Kung adders are shown in Figure 4. It should be noted that while all the outputs of a pruned Kogge-Stone adder have equal amount of carry propagation paths due to its regular structure, the outputs of other parallel adders (such as Brent-Kung) have varying number of carry propagation paths.

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. Examples of resulting 16-bit Pruned Kogge-Stone and Brent-Kung adders are shown in Figure 5.

V. Experimental Results and Analysis

A. Methodology and Framework

Our logic-synthesis based CAD methodology for applying probabilistic pruning technique on adders is based on Figure 2. 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. Accurate error values
of the resulting pruned circuits are computed through functional simulations using a testbench consisting of uniformly distributed 64-bit random numbers in ModelSim. We will then synthesize the VHDL code of the pruned circuit using Cadence RC Compiler, and perform Place & Route using Cadence SoC Encounter to obtain the layout along with the final netlist and parasitics. We then estimate the power consumption of the pruned circuit by back-annotation of the final netlist and parasitics using Synopsys Power Compiler. The required toggling frequency of each net and cell in the netlist for the Power Compiler are produced through the Switching Activity Interchange Format (.saif) file derived from the Value Change Dump (VCD) file. The VCD file is produced by simulation of the netlist along with the Standard Delay Format (SDF) file in ModelSim using the previously created testbench. All the designs are implemented using TSMC 180nm (Low Power) technology library, and to croscheck 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.

Specifically, in addition to the observations highlighted through italicization in the paragraphs above that are of potential value to circuit design in general, we can also observe that probabilistic pruning is a design level technique which does not involve varying of circuit 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 [27] 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.

REFERENCES


