Design and Application of Approximate Circuit by SAIF Pruning

K.Mahalakshmi
S.Chitra M.E
E.G.S. Pillay Engineering College,Nagapattinam

Abstract
Inexact and approximate circuit design is a promising approach to improve performance and energy efficiency in technology-scaled and low-power digital systems. Such strategy is suitable for error-tolerant applications involving perceptive or statistical outputs.Probabilistic pruning technique has been proposed for an efficient approximate tangent function.The approximation is based on a mathematical analysisconsidering the maximum allowable significance-activity product (SAP) and unessential nodes.Hardware implementation of the proposed approximation schemeis presented, which shows that the proposed structure comparesfavorably with previous architectures in terms of area and delay.

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 ofembedded 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]. Inexact and approximate circuit design [1] is a radical approach to trade this counterproductive quest for perfection for substantial gains in power, speed, area and yield. The primary challenge, however, is to determine *where* and *how* to let an error or an approximation occur in the circuits without compromising the functionality or the user experience. With ever-increasing amount of data being processed, a wide variety of applications could tolerate inaccuracies. For example, in multimedia processing, a small proportion of errors is not perceptible to humans, and in highly computational algorithms such as data mining, search or recognition, the outcome is not required to be a single result but an adequate match. A promising approach to design inexact circuits is to use *speculation* to trade circuit accuracy for better power and speed. Taking advantage of such circuits would help to realize extremely energy-efficient and high-performance DSPs and hardware accelerators at lower integration cost and with higher speed, data rate or duty-cycling.

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.

Pruning

Once the nodes are ranked according to their SAP, significance only or activity only, the gate-level netlist is modified in order to remove *unessential* nodes from the design. For the sake of simplicity, and in order to maximize the use of the existing EDA tools, the probabilistic pruner does not literally remove the gates form the netlist, but it disconnects the corresponding wires. Gates whose outputs are unconnected will automatically be removed by the synthesis tool. However, leaving gate inputs unconnected would fail the resynthesis of
the design. For this reason, and in order to minimize the error, those inputs are set to 0 if they statistically spend most of the time at 0 (i.e., $T0 \geq T1$). Otherwise they are connected to 1 (i.e., $T0 < T1$). This should allow to statistically reduce the error magnitude. The synthesis of the modified netlist therefore
improves the design in two ways.

1) One or more gates having their outputs unconnected are removed, allowing direct area, power, and delay savings.
2) Gates having their inputs set to 1 or 0 can then be replaced by lower complexity ones.

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. Error Rate = Number of Erroneous Computations
Total Number of Computations

RELATED WORK

K.V. Palem et al[1] is proposed here the estimates of the energy saved through PBIT-based probabilistic computing switches and networks developed rely on the constructs and thermodynamic models due to Boltzmann, Gibbs, and Planck, this work has also led to the innovation of probabilistic CMOS-based devices and computing frameworks.

Kaushik Roy et al[2] presents a review of various cross-layer design options that can provide solutions for dynamic voltage over-scaling and can potentially assist in meeting the strict power budgets and yield/quality requirements of future systems.

Ilia Polian et al[3] present an energy-reduction strategy for applications which are resilient, i. e. can tolerate occasional errors, based on an adaptive voltage control. The voltage is lowered, possibly beyond the safe-operation region, as long as no errors are observed, and raised again when the severity of the detected errors exceeds a threshold.

AnandRaghunathan et al[4] is proposed here logic complexity reduction at the transistor level as an alternative approach to take advantage of the relaxation of numerical accuracy. We demonstrate this concept by proposing various imprecise or approximate full adder cells with reduced complexity at the transistor level, and utilize them to design approximate multi-bit adders.

Milos Ercegovac et al[5] propose a novel multiplier architecture with tunable error characteristics, that leverages a modified inaccurate 2×2 building block. Our inaccurate multipliers achieve an average power saving of 31.78% - 45.4% over corresponding accurate multiplier designs, for an average error of 1.39% - 3.32%.

Jie Han et al[6] describe deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. These designs rely on different features of compression, such that imprecision in computation (as measured by the error rate and the so-called normalized error distance) can meet with respect to circuit-based figures of merit of a design (number of transistors, delay and power consumption). Four different schemes for utilizing the proposed approximate compressors are proposed and analyzed for a Dadda multiplier. Extensive simulation results are provided and an application of the approximate multipliers to image processing is presented.

C. Lucas et al[7] is proposed here the conventional digital hardware computational blocks with different structures are designed to compute the precise results of the assigned calculations. The main contribution of our proposed Bio-inspired Imprecise Computational blocks (BICs) is that they are designed to provide an applicable estimation of the result instead of its precise value at a lower cost.

Christian Enz et al[8] is presents here a novel architecture of an Inexact Speculative Adder with optimized hardware efficiency and advanced compensation technique with either error correction or error reduction.

Seokhyeong Kang et al [9] propose an accuracy-configurable approximate (ACA) adder for which the accuracy of results is configurable during runtime. Because of its configurability, the ACA adder can adaptively operate in both approximate (inaccurate) mode and accurate mode.

Christian Enz et al[10] describe an approximate adder architecture based on a digital quasi-feedback technique called Carry CutBack in which high-significance stages can cut the carry propagation chain at lower-significance positions. This lightweight approach prevents activation of the critical path, improving energy efficiency while guaranteeing low worst-case relative error.

SheriefReda et al [11] is proposed here that our design can achieve power savings of 54% - 80%, while introducing bounded errors with a Gaussian distribution with near-zero average and standard deviations of 0.45% - 3.61%.

Christian Piguet et al[12] 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.

Krishna V. Palem et al[13] presents a methodology to automatically generate inexact circuits starting from a conventional design by adding only one small step in the digital design flow.

Vincent Camus et al[14] is proposed here the floating-point unit is one of the most common building block in any computing system and is used for a huge number of applications. By combining two state-of-the-art techniques of imprecise hardware, namely Gate-Level Pruning and Inexact Speculative Adder, and by introducing a novel Inexact Speculative Multiplier architecture, three different approximate FPUs and one reference IEEE-754 compliant FPU have been integrated in a 65 nm CMOS process within a low-power multi-core processor.

Nilanjan Banerjee etal[15] present a novel discrete cosine transform (DCT) architecture that allows aggressive voltage scaling for low-power dissipation, even under process parameter variations with minimal overhead as opposed to existing techniques. Under a scaled supply voltage and/or variations in process parameters, any possible delay errors appear only from the long paths that are designed to be less contributive to output quality.

PROPOSED APPROXIMATION SCHEME

In this section, mathematical analysis of the approximation scheme used for hardware implementation of hyperbolic tangent function is provided. The mathematical analysis in this and the following sections uses the basic properties of hyperbolic tangent function. Hyperbolic tangent is an odd function tanh(−x) = − tanh(x). (2) Using this property, only the absolute value of input is processed and the input sign is directly passed to the output.

Output Approximation in the Pass Region

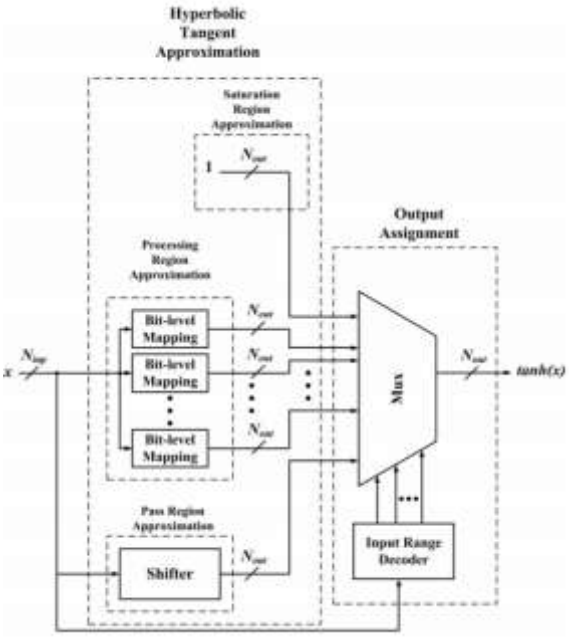
The input and output of hyperbolic tangent function are represented as signed-magnitude notation. Therefore, considering the first basic property of the hyperbolic tangent function discussed in previous section, input sign bit is directly passedto the output sign bit and only the absolute value of input is processed

Output Approximation in the Processing Region

Before going through the proposed approximation scheme in this region, a new parameter named None is introduced. This parameter is an indicator of position of the first occurrence of one in binary input, when scanned from left.

Output Approximation in the Saturation Region

The hyperbolic tangent function reaches its maximum value in the saturation region, while at the same time output variation in this region is low. Therefore, all output values in this region are approximated by the maximum value representable by the output bits. Hyperbolic Tangent Approximation



This block is composed of three main blocks to approximate the hyperbolic tangent function in all three regions, including saturation, processing, and pass region. General arithmetic operations in each region can be described as follows. 1) Pass Region: In this region, fractional part of input is passed to the output. Based on (8), a shift to left by Nout − N f bits before passing the input to output is required. 2) Processing Region: For inputs in the processing region, a bit-level input mapping is required. The number of bit-level mapping blocks required is equal to the number of input ranges in this region. For each input range in the processing region, logN 2 bits after None bit of input should be mapped to output bits using the bit-level mapping. Using logN 2 bits after None bit covers all sub-ranges. The number of sub-ranges (N) is calculated using (13) while the output of each sub-range is found using (12). The bit-level mapping can be implemented using a combinational

circuit. 3) Saturation Region Approximation: In this region, hyperbolic tangent function is approximated by the maximum value representable by output bits, and can be realized by setting all output bits to one.

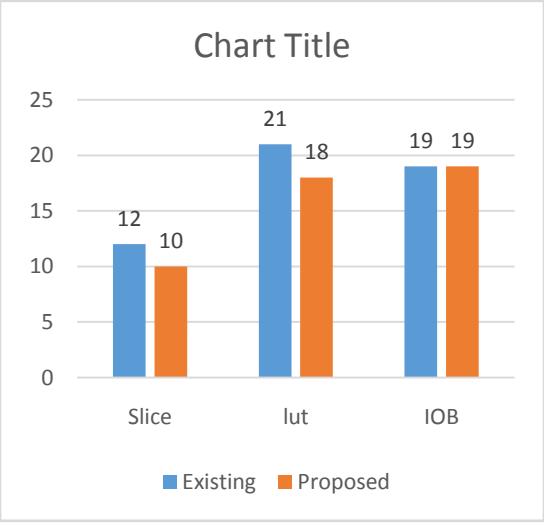
Implementation results

The proposed has been simulated and the synthesis report can be obtained by using Xilinx ISE 12.1i. The various parameters used for computing existing and proposed systems with Spartan-3 processor are given in the table 7.1.

s.no	Parameter	Existing	Proposed
1	Slice	12	10
2	lut	21	18
3	IOB	19	19

PERFORMANCE ANALYSIS

The Figure given below is shown that there is a considerable reduction in time and area based on the implementation results which have been done by using Spartan-3 processor. The proposed algorithm significantly reduces area consumption when compared to the existing system.



CONCLUSION

A new approximation scheme for hyperbolic tangent was proposed in this project. The proposed approximation scheme is based on a mathematical analysis considering maximum allowable error as a design parameter. Based on the proposed approximation scheme, a hybrid architecture for hardware implementation of hyperbolic tangent activation function was presented. The synthesis results showed that the proposed structure compares favorably to the previously developed architectures in terms of area, delay, and area×delay. The proposed structure required less output bits for the same maximum allowable error compared to the previously developed architectures

REFERENCES

[1] K.V. PalemEnergy aware computing through probabilistic switching: a study of limits, IEEE Transactions on Computers.

[2] Kaushik Roy,Voltage over-scaling: A cross-layer design perspective for energy efficient systems, 2011 20th European Conference on Circuit Theory and Design (ECCTD).

[3] Ilia Polian,Adaptive voltage over-scaling for resilient applications,2011 Design, Automation & Test in Europe.

[4]AnandRaghunathan ,Low-Power Digital Signal Processing Using Approximate Adders, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[5] Milos Ercegovac ,Trading Accuracy for Power with an Underdesigned Multiplier Architecture, 2011 24th Internatioal Conference on VLSI Design.

[6] JieHan,Design and Analysis of Approximate Compressors for Multiplication, IEEE Transactions on Computers.

[7] C. Lucas,Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications, IEEE Transactions on Circuits and Systems I: Regular Papers.

[8] Christian Enz,Energy-efficient inexact speculative adder with high performance and accuracy control, 2015 IEEE International Symposium on Circuits and Systems (ISCAS).

[9] Seokhyeong Kang,Accuracy-configurable adder for approximate arithmetic designs, DAC Design Automation Conference 2012.

[10] Christian Enz,A low-power carry cut-back approximate adder with fixed-point implementation and floating-point precision, 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC).

[11] SheriefReda,DRUM: A Dynamic Range Unbiased Multiplier for approximate applications, 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD).

[12] Christian Piguet,Energy parsimonious circuit design through probabilistic pruning, 2011 Design, Automation & Test in Europe.

[13] Krishna V. Palem,Automatic generation of inexact digital circuits by gate-level pruning, 2015 IEEE International Symposium on Circuits and Systems (ISCAS).

[14] Vincent Camus,Approximate 32-bit floating-point unit design with 53% power-area product reduction, ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference.

[15] NilanjanBanerjee ,Process-Variation Resilient and Voltage-Scalable DCT Architecture for Robust Low-Power Computing, IEEE Transactions on Very Large Scale Integration (VLSI) Systems.