BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER SYSTEMS
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

AUTOMATED DESIGN METHODOLOGY FOR APPROXIMATE LOW POWER CIRCUITS

METODOLOGIE PRO AUTOMATICKÝ NÁVRH NÍZKOPŘÍKONOVÝCH APROXIMATIVNÍCH OBVODŮ

EXTENDED ABSTRACT OF A PHD THESIS
ROZŠÍŘENÝ ABSTRAKT DISERTAČNÍ PRÁCE

AUTHOR — **Ing. VOJTĚCH MRÁZEK**
AUTOR PRÁCE

SUPERVISOR — **prof. Ing. LUKÁŠ SEKANINA, Ph.D.**
ŠKOLITEL

BRNO 2018

Contents

Chapter 1

Introduction

This chapter starts with a brief motivation for the research conducted as a part of my doctoral study at the Faculty of Information Technology, Brno University of Technology in 2014-2018. In this chapter, the open problems are identified, and the research objectives are formulated. Finally, the thesis outline is introduced.

1.1 Motivation

Despite the rapid developments in the very-large-scale integration (VLSI) circuit technologies and in modern circuit design techniques, the overall energy consumption of integrated circuits is rapidly growing mainly due to their increasing complexity needed in current computing systems. At the same time, many computationally intensive applications, e.g. image recognition, signal processing and data mining, are widely implemented in these systems. Moreover, the expansion of modern battery-powered and smart devices such as mobile systems, IoT nodes and wearable electronics emphasizes the low-power requirements put on the computer-based systems.

It is essential to improve the energy efficiency of these systems implemented. Fortunately, many of the computationally intensive applications feature an intrinsic error-resilience property [3]. Since they process noisy or redundant data and users are willing to accept certain errors in some cases, an emerging paradigm, the so-called *approximate computing*, can be used for design of energy-efficient applications. At the circuit level, approximations (circuit simplifications) are intentionally introduced to find a good tradeoff between power consumption and error.

The approximations can be introduced to the circuit in various steps of the standard circuit design flow. In this thesis, we primarily focus on the technology independent logic synthesis step because it has one important advantage — the approximate circuit can be used in arbitrary ASIC as well as FPGA technology, because we can assume the technology dependent synthesis is performed by means of some well-optimized open source or commercial tool after the approximation is conducted. The common logic synthesis typically starts with a circuit description given at the register-transfer-logic (RTL) level, in the form of a source code written in hardware description language (HDL). The goal of the logic synthesis is to transform a typically suboptimal solution into a close to optimal implementation with respect to given synthesis goals. For example, if a gate-level netlist is the starting point, we speak about logic optimization, because the circuit is described at the gate level which directly represents logic expressions specifying its behavior. The ultimate goal of the logic

optimization is thus to transform this netlist into an optimal gate-level implementation with respect to given synthesis goals.

The performance (delay) and area parameters are usually considered as the key design objectives in the traditional design flow. Since the power consumption becomes more and more important, the power consumption objective was included to the goals in modern circuit design. In the emerging approximate circuit design flow, the error (accuracy) is a new design objective (Fig. 1.1) [35].

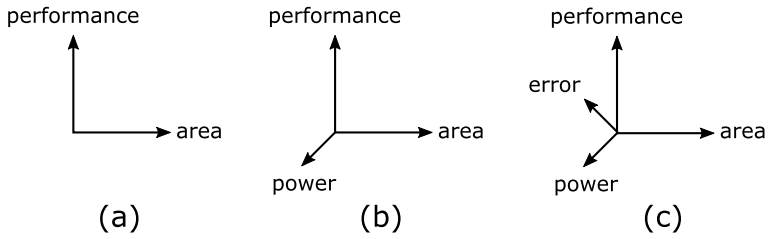

Figure 1.1: Design parameters considered in (a) traditional, (b) low-power and (c) approximate design flows [35]

Recently, a new approach to the synthesis of digital circuits has been proposed. It exploits evolutionary algorithms (EA) — a promising class of optimization algorithms inspired in evolutionary biology. The evolutionary logic synthesis is a method utilizing EAs to design and optimize logic circuits. Many papers showing the merits of the EA in this domain have already been published in the literature [34, 40]. The thesis deals with the use of EA for approximate logic synthesis.

1.2 Open problems

Although the EAs have already been successfully applied to the optimization of complex logic circuits, some open problems still exist. The main issue of the traditional (accurate) logic synthesis based on EA is the limited scalability. It means that EA typically produces a high-quality solution for low complex problem instances, but an insufficient (or even none) solution is provided for complex problem instances. There are two main types of the scalability problem: (i) the scalability of representation — the search-space grows rapidly with the representation size, and (ii) the scalability of evaluation — the increasing number of inputs and circuit components causes longer evaluation time. These problems contribute to a very limited acceptance of the method by the community of hardware designers. Another issue is that the optimization criteria, typically the number of active nodes, do not usually correspond with the hardware parameters such as power consumption or area, and hence, the resulting circuits are not acceptable in real-world applications. Open problems of the EA-based logic synthesis are how to: (i) effectively represent the circuits for the EA, (ii) quickly evaluate the functionality of the circuit, and (iii) perform fast estimation of hardware parameters (delay, power consumption etc.).

Regarding approximate circuits, some of the open problems are how to determine the error of an approximate circuit and which error metric is suitable for a given application. It is also unclear how to perform desired approximations.

1.3 Research objectives

The main objective of the research presented in this thesis is to show that

it is possible to reduce power dissipation of digital combinational circuits described on various levels of abstraction using EA-based circuit approximation algorithms.

First of all, it is important to select representative applications to demonstrate the proposed approximation methods. The next stage is to develop the EA-based circuit approximation method for these applications. As indicated in Section 1.2, such an approximation method has to meet several requirements. It has to be able to design circuits of a sufficient complexity that can be employed in the real-world application. Therefore, the evaluation of candidate circuits has to be effective. To optimize the power dissipation, some power estimation method has to be included in the EA.

The main objective of the thesis can be divided to the following partial goals:

1. To identify real-world applications suitable for approximation.
2. To create fast and efficient power estimation methods for the selected levels of circuit description.
3. To implement an evolutionary algorithm for efficient approximation of digital circuits.
4. To tune and accelerate the evolutionary algorithm in order to speed up the circuit approximation process.
5. To experimentally evaluate the proposed method using the selected real-world application.
6. To compare obtained results with the state-of-the-art solutions.

1.4 Abstract outline

This extended abstract summarizes author's doctoral thesis. The thesis is composed as a collection of selected author's papers with an accompanying introductory part. The seven peer-reviewed papers encapsulate the contribution of the dissertation.

The extended abstract of the thesis is organized as follows. The first chapter provides the introduction to the research area and research objectives. Chapter 2 briefly surveys the state-of-the-art already published in the literature. Chapter 3 summarizes the research contribution and introduces the selected papers. Chapter 4 summarizes the obtained results and proposes directions for the future research.

Chapter 2

Survey of the state of the art

This chapter introduces the background utilized in the presented work. It primarily addresses the power consumption of digital circuits and its estimation, approximate computing techniques and evolutionary methods for the digital circuit design.

2.1 Power consumption of digital circuits

Today it is widely accepted that power efficiency is a key goal of digital circuit design. Minimizing power consumption calls for conscious effort at each abstraction level and at each phase of the design process.

There are three sources of power dissipation in the standard complementary metal-oxide-semiconductor (CMOS) digital combinational circuits. The first one is the existence of logic transitions. As the "wires" switch between two logic (and so electrical) values, the parasitic capacitances are charged and discharged. This component of power dissipation, the so-called switching power P_{switch} is proportional to supply voltage V, wire voltage swing V_{swing}, switching activity coefficient α and switched capacitance C_{LOAD}. As the voltage swing is usually equal to the supply voltage, the dissipation varies with the square of the supply voltage. Short-circuit currents that directly flow from supply to ground when both complementary subnetworks conduct simultaneously are the second source of power dissipation. When inputs of the gate are stable, only one subnetwork is conducting, and no short circuit current can flow. When the output of the gate is changing its value in response to the changes in inputs, both subnetworks conduct simultaneously for a short interval. This power dissipation P_{short} varies with the average short-circuit current I_{SC} and supply voltage V. Both the above-mentioned sources of power dissipation in CMOS circuits are related to the transitions at gate outputs. They are collectively referred to as *dynamic power*. In contrast, the last source of power dissipation is leakage current I_{leak} originating mainly from subthreshold MOS conduction. Nowadays, this leakage power dissipation becomes more important because transistors are smaller and smaller [29].

$$P_{CMOS} = P_{switch} + P_{short} + P_{leakage} \tag{2.1}$$

$$P_{CMOS} = \frac{1}{2} \cdot C_{LOAD} \cdot \alpha \cdot f \cdot V^2 + I_{SC} \cdot V + I_{leak} \cdot V \tag{2.2}$$

This section gives a brief introduction to the main factors having a crucial impact on the power dissipation of digital circuits. In the first section, the impact of selected fabrication technology is discussed. Digital circuits can be described at various levels of abstraction.

In the next sections, two different levels are presented — the transistor level netlist and the gate level netlist. For the both levels the power estimation methods and power saving techniques are presented.

2.1.1 The technology impact

The overall power consumption strictly depends on the selected fabrication technology. The technology is usually specified using the so-called layout design rules. Since migration from one process to a more complex one is difficult, these rules were simplified using scalable rules. Mead and Conway [15] popularized the scalable design rules based on a single parameter λ that characterizes the process resolution. This parameter is generally half of the minimum drawn transistor channel length.

The decreasing transistor sizes lead to a reduction in capacitance of the gates. In addition to that, the supply voltage is decreasing too. For example, the 500 nm technology needs 5 V supply, 180 nm needs 1.8 V and 45 nm technology uses 1.0 V supply only. As the consequence of that, threshold voltages V_{tn} (for n-mos transistors) and V_{tp} (for p-mos transistors) grow up relatively to the supply voltage. The threshold voltages influence the leakage.

The target fabrication technology is usually chosen with respect to price and availability. For the automated design, it is important to consider that there is a trade-off between power consumption and performance (speed). In order to address this, technology libraries typically provide two or more versions of each cell, e.g. one fast, but dissipating, and one slow, but low-power. The designer can then select which implementation fits his/her needs.

2.1.2 Transistor-level digital circuits

As the circuit performance and power dissipation heavily depend on the chosen design style, the low-level description of digital circuits enables to choose different design styles. For example, fully complementary logic (CMOS), NMOS, differential cascade voltage switch or pass-transistor logic design style can be used [29]. The design styles differ in the power dissipation, performance and scalability of circuits that can be obtained. The transistor-level circuits are usually described using a netlist consisting of transistors and additional devices such as capacitors or resistors.

One of the most accurate and straightforward method for power estimation of the circuits described at the level of transistors is to perform a circuit simulation by means of some SPICE-like[1] simulator.

It was shown that simulation results of the circuits utilizing these models are usually strongly pattern-dependent [11]. Hence, all the possible input transitions should be simulated in order to obtain trustworthy results. This can become computationally very expensive, especially for complex circuits, because the number of transitions grows exponentially.

2.1.3 Gate-level digital circuits

A gate level circuit is defined as a set of cells (gates) and their connections. To simulate circuits with millions of gates, the simulation model must be simple enough to handle this complexity, yet accurate enough to give trustworthy results.

The designers cannot change the design style in the same way as it was possible at the low-level (transistor) description, because the elementary gates are defined in the *techno-*

[1] *Simulation Program with Integrated Circuit Emphasis*

logical library. A typical technological library is developed and optimized for a certain fabrication process and contains various implementations and models of elementary gates and other basic components such as one-bit adders, and-or-invert structures, multiplexers etc. Corresponding to equation 2.2, four basic possibilities how to reduce power dissipation can be identified. The following list gives basic power optimization techniques applicable at the gate level:

- reduction of the switching activity α,
- reduction of load capacitance of gates C_{LOAD},
- frequency modification f,
- optimizing power domains V.

The load capacitance comes from the wires and transistors involved in a circuit. Good floorplaning and placement can minimize the wire capacitance. The load capacitance of gates can be reduced by choosing fewer stages of logic and smaller transistors. A common technique is to intentionally increase delay (up to a delay constraint) by choosing a slower, but more energy efficient, version of the gate from the technological library. Available implementations of the same gate (e.g. AND) differ in the channel sizes and threshold voltages.

One of the most powerful power optimization techniques is called *clock gating*. This approach masks (using AND gate) a clock signal with an enable signal to turn off the clock input of idle blocks. It is effective because the clock signal has a high activity factor. There is no clock signal in the combinational digital circuits, but we can select an arbitrary input with high switching activity (Fig. 2.1a).

Dynamic power is proportional to the frequency. Hence, a chip should not obviously run faster than it is necessary. Reducing the frequency also allows downsizing transistors as mentioned above. The performance can be recouped through parallelism, especially if the area is not as important as the power. The circuit may employ multiple frequency domains so that certain portions of the chip can run more slowly than others. However, the wires connecting the frequency domains must be synchronized.

Supply voltage has a quadratic effect on the dynamic power. Therefore, choosing a lower power supply significantly reduces power consumption. But V_{cc} value varies the performance of the circuit and the minimal value is typically specified by fabrication technology. However, the chip may be divided into multiple voltage domains affecting the performance of the circuits. In addition to that, the voltage domains can be turned off entirely to eliminate leakage power the during sleep mode. This technique is called *power gating*. Fig. 2.1b shows that the block is active, when the header switch transistors are connecting the supply voltage to the block. The outputs of the block may take on voltage levels in the forbidden zone. Hence, the output isolation gates must be used. This technique is required for leakage reduction of inactive parts of circuits [30].

It is infeasible to use an accurate SPICE simulator to determine power consumption of complex gate-level circuits. To address this problem, the power consumption of gate-level circuits is typically estimated using some suitable models. There are several models for the analysis of gate-level circuits that are much faster than numeric simulations. For each gate in the cell library the models (of physical behavior) are stored in a lib file satisfying Liberty standard[2]. These models used to be based on k-factor lookup tables, which gave the gate

[2]http://opensourceliberty.org

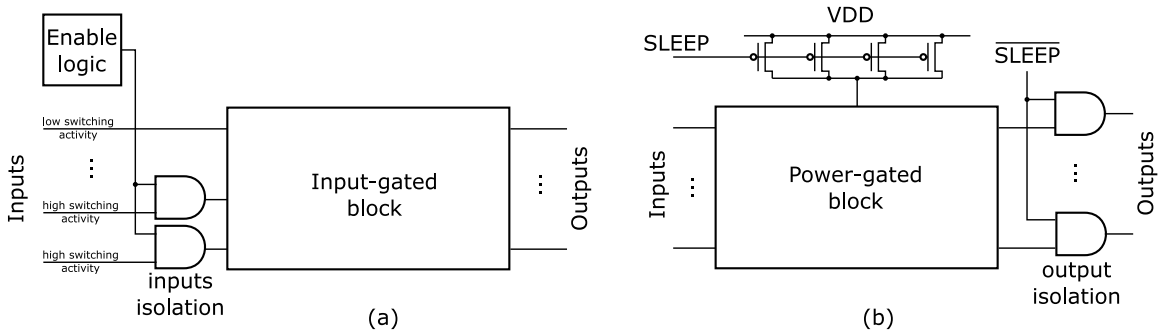

Figure 2.1: Two common low-power design techniques: (a) clock / signal gating, and (b) power gating [43].

delays and the output signal transition times based on the gate input signal transition times and the gate capacitive load [10]. However, these lookup-table delay models did not contain enough information to characterize the parameters of multi-voltage circuits implemented in modern fabrication technologies. These limitations have motivated the development of the current source models. The current source models define the output DC current as a nonlinear function of the input and output voltages of the cell.

When the circuit parameters are determined using a suitable model, the switching activity factor α is estimated. This estimation is a non-trivial task, complicated by the fact that the switching activity depends not only on the circuit structure and functionality, but also on the applied input patterns. Two approaches have been proposed to address this problem – dynamic methods employing a variant of circuit simulator and static methods based on probabilistic techniques analyzed how the switching activity is propagated from inputs to outputs [28]. From the viewpoint of scalability, the probabilistic methods seem to be the preferred ones nowadays. Several variants of binary decision diagrams accompanied with partitioning have been proposed to determine the switching activity [14]. Some assumptions are usually introduced to reduce the computational complexity. When we neglect glitching, for example, we can use a zero-delay model to compute the switching activity [4].

2.2 Approximate arithmetic circuits

Despite of the development of design techniques for energy efficient circuits, the overall energy consumption of computer systems is still rapidly growing. As many important applications are inherently error resilient, precision of the involved computations can be traded for improved energy efficiency, performance, and/or chip area. Various approaches exploiting this fact have been developed in recent years and presented under the umbrella of the so-called approximate computing [3, 18].

In many scenarios, the use of approximate computing is unacceptable, while in others, the approximate computing paradigm can be proactively used to reduce power consumption. The motivations for introducing approximate computing can be: (i) satisfying the requirement for low power dissipations, (ii) some applications are error resilient and a strictly accurate output is not needed, or (iii) eliminating expensive fault-tolerant mechanisms that are needed to ensure reliability of circuits implemented with recent fabrication technology. According to [7, 31], applications suitable for approximate computing can be broadly classified to four classes:

1. Applications with analog inputs which operate on noisy real-world data.
2. Applications with analog output intended for human perception.
3. Applications with no unique answer such as web search and machine learning.
4. Iterative and convergent applications that iteratively process large amounts of data and the equality of results depends on the number of iterations.

These approximations can be conducted at different system levels with circuit approximation being one of the most popular. Circuit approximation techniques can be classified into two main groups. (1) *Frequency/voltage over-scaling*, where timing-induced errors can appear as the circuit is operated on lower voltage than the nominal value [33]. (2) *Functional approximation*, where the original circuit is replaced by a less complex one which exhibits some output errors but improves non-functional circuit parameters such as power consumption or area on the chip.

In this work, we focus on functional approximation only. The principle of this approach is to implement a different function (circuit) with respect to the original one provided that the non-functional parameters such as power dissipation or area are improved, and the error is acceptable. Typically, the goal is to find a design showing a minimum area (or power consumption) and satisfying a given error constraint [44].

2.2.1 Approximation methods for arithmetic circuits

We focus on approximate arithmetic circuits because they are frequently used in the key applications relevant for approximate computing. The methods for functional approximations can be divided into two categories: (1) manual, and (2) automated.

The *manual (ad-hoc) methods* are developed for a specific circuit component. In this work, examples of manual approximation of two key arithmetic circuits — adders and multipliers — are described. These circuits are widely approximated because they realize key operations in applications requiring low power processing. However, other circuits such as dividers and multiply-and-accumulate (MAC) circuits were manually approximated too, but they are not widely employed. Hence, these circuits are not discussed in this chapter. On the other hand, the *automated methods* use some general-purpose circuit resynthesis and approximation techniques and enable us to approximate arbitrary circuits. These methods starts with an original (exact) circuit and, typically iteratively, modifies its structure.

2.2.2 Quality of approximate circuits

The quality of approximate combinational circuits is typically expressed using one or several error metrics. In addition to the error rate, the average-case as well as the worst-case situation can be analyzed. Among others, the mean absolute error (MAE) and the mean relative error (MRE) are the most useful metrics that are based on the average-case analysis. Selection of the right metrics is a key step of the whole design. When an arithmetic circuit is approximated, for example, it is necessary to base the error quantification on an arithmetic error metric. For general logic circuits, where no additional knowledge is available and where there is not a well-accepted error model, Hamming distance or error rate is typically employed.

The following paragraphs summarize the error metrics that have been employed in literature to quantify the deviation between the outputs produced by a functionally correct

design and an approximate design. These metrics are divided to two categories. The category of arithmetic errors consists of metrics that compare integer values of the circuit outputs. The Boolean error metrics are classified as general errors.

Arithmetic error metrics

Let $f : \mathbb{B}^n \to \mathbb{B}^m$ be an n-input m-output Boolean function that describes the correct functionality (the accurate function) and $f' : \mathbb{B}^n \to \mathbb{B}^m$ be an approximation of it, both implemented by two circuits, namely F and F'.

The *worst-case arithmetic error*, sometimes denoted as *error magnitude* or *error significance* [1], is defined as

$$e_{wce}(f, f') = \max_{\forall x \in \mathbb{B}^n} |int(f(x)) - int(f'(x))|, \tag{2.3}$$

where $int(x)$ represents a function $int : \mathbb{B}^m \to \mathbb{Z}$ returning an integer value of the m-bit binary vector x. Typically, a natural unsigned binary representation is considered, i.e. $int(x) = \sum_{i=1}^{m} 2^i \cdot x_i$. The worst-case error represents the fundamental metric that is useful to guarantee that the approximate output differs from the correct output by at most error bound e.

In the literature, the *relative worst-case error*

$$e_{wcre}(f, f') = \max_{\forall x \in \mathbb{B}^n} \frac{|int(f(x)) - int(f'(x))|}{int(f'(x))} \tag{2.4}$$

is frequently employed to constrain the approximate circuit to differ from the correct one by at most a certain margin. Note that a special care must be devoted to the cases for which the output value of the original circuit is equal to zero, i.e. the cases when the denominator approaches zero. This issue can be addressed by either omitting test cases when $int(f(x)) = 0$, or biasing the denominator by 1. The first approach is usually employed in the manual approximation methods where the zero results are accurate [9].

The *average-case arithmetic error* (also known as *mean absolute error*) is defined as the sum of absolute differences in magnitude between the original and approximate circuit, averaged over all inputs:

$$e_{mae}(f, f') = 2^{-n} \sum_{\forall x \in \mathbb{B}^n} |int(f(x)) - int(f'(x))|. \tag{2.5}$$

If we replace the expression in the sum by the equation for relative error distance, we can calculate the *mean relative error*:

$$e_{mre}(f, f') = 2^{-n} \sum_{\forall x \in \mathbb{B}^n} \frac{|int(f(x)) - int(f'(x))|}{int(f'(x))}. \tag{2.6}$$

Note that the values produced by absolute error metrics e_{mae} and e_{wce} can be very large. Hence, these values can be expressed as a part of the output range using multiplication by 2^{-m}. For example, the worst-case arithmetic error 64 for an 8-bit output circuit is equal to 25% error.

General error metrics

In addition to the arithmetic error metrics, there are metrics that are not related to the magnitude of the output of the correct or approximate circuit.

The *error rate*, referred to as the *error probability*, represents the basic measure that is defined as the ratio of inputs vectors for which the output value differs from the original one:

$$e_{prob}(f, f') = 2^{-n} \sum_{\forall x \in \mathbb{B}^n : f(x) \neq f'(x)} 1. \tag{2.7}$$

In many cases, it is also worth to consider the Hamming distance between $f(x)$ and $f'(x)$. The *worst-case Hamming distance*, denoted also as *bit-flip error* [2], is defined as

$$e_{bf}(f, f') = \max_{\forall x \in \mathbb{B}^n} \sum_{i=1}^{m} (f(x) \oplus f'(x))_i, \tag{2.8}$$

and gives the maximum number of output bits that simultaneously output a wrong value. The average number of changed output bits, denoted as the *average Hamming distance*, can be expressed as follows:

$$e_{mhd}(f, f') = 2^{-n} \sum_{\forall x \in \mathbb{B}^n} \sum_{i=0}^{m} (f(x) \oplus f'(x))_i. \tag{2.9}$$

2.2.3 Quality evaluation

In the error metric formulas, the enumeration of all possible input vectors is employed. For larger number of inputs n, it is not feasible to enumerate $\mathbb{B}^n$. We can deal with this issue by (a) enumerating a subset of $\mathbb{B}^n$, or (b) obtaining the exact value using a formal verification approach. Therefore, the evaluation methods can be divided to the following classes.

- *Exact (formal) evaluation* where the error is calculated using formal verification methods or obtained by a simulation tool performing the exhaustive simulation.
- *Inexact evaluation* using just a subset of $\mathbb{B}^n$ or a probabilistic error analysis.

This section introduces several approaches for the evaluation of the quality of approximation.

Simulation

This method can be used in the exact evaluation scenario. A circuit simulator that calculates responses for all input vectors is employed to provide the exact error metric. The simulation method is capable of determining all error metrics introduced in the previous section. In order to maximize the efficiency of the implementation, the simulator usually uses parallel acceleration techniques such as vectorization enabling bit-level parallel simulation. The idea is to utilize bitwise operators operating on multiple bits to perform more than one evaluation of a gate in a single step. This approach benefits from the fact that modern processors are equipped with specialized SIMD instructions.

Since it is unfeasible to perform the exhaustive enumeration for a larger number of inputs due to the exponential complexity, the simulation frequently utilizes a subset of input vectors, typically selected by Monte-Carlo method [9]. To determine the required subset size, statistical models can be used [13].

BDD-based formal verification

To overcome time limitations of the simulation, various formal approaches can be employed for exact error evaluation [36]. In the classical logic synthesis, determining whether two Boolean functions are functionally equivalent represents a fundamental problem solved using formal verification techniques. These widely explored techniques can be extended for the error metrics calculation, in the so-called *relaxed equivalence checking* scenario.

One of the state-of-the-art verification methodologies is based on Reduced Ordered Binary Decision Diagrams (ROBDD). ROBDD is a canonical rooted acyclic graph-based representation of Boolean function, where each node represents either an input variable or terminal node, and the edges represent the value assignment. ROBDDs have been traditionally used to solve the equivalence checking problem due to their canonical property. The ROBDD tools can exactly answer two questions: (1) whether a variable (signal) in the circuit is satisfiable, i.e. whether some input vector exists which generates true value of the signal; and (2) the probability that a selected variable is true. In the literature [32], algorithms determining the worst-case error and the average error of an approximate circuit were published.

Most formal verification approaches developed for testing exact equivalence are not directly extendable for relaxed equivalence checking. A common approach to the error analysis is to construct an auxiliary circuit referred to as an approximation miter. The structures of various approximation miters is shown in Figure 2.2. For arithmetic error metrics, a two's complement subtractor followed by a circuit which determines the absolute value is employed. Such a circuit can be used for formal evaluation of the mean and the worst-case arithmetic errors. For Hamming distance and the error rate, incorporating exclusive-or gates connecting the corresponding outputs is sufficient.

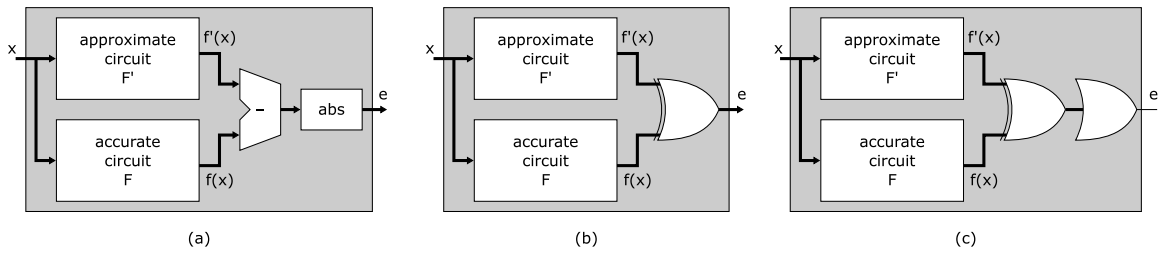

Figure 2.2: Approximation miter for the formal error analysis. Implementation of the error computation block for (a) arithmetic error, (b) Hamming distance and (c) error rate.

SAT-based formal verification

ROBDD based verification is inefficient in representing certain classes of functions. It is a well-known fact that the size of BDD is sensitive to the chosen variable ordering. In addition to that, there are functions, e.g. multipliers, for which the BDD size is always exponential, independent of variable ordering [36].

The advanced relaxed equivalence checking approach transforms the error computation to a satisfiability (SAT) problem which is solved by means of a SAT solver. Modern SAT solving algorithms are extremely effective at coping with large problem instances and large search spaces. The basic principle is to translate the problem of functional equivalence of two combinational circuits to the problem of deciding whether a Boolean formula given in conjunctive normal form (CNF) is satisfiable or not. This formula is constructed using

miters shown in Figure 2.2. With respect to the BDD-based verification, SAT solving can effectively process large circuits, but the satisfiability (e.g. proving, that some signal can or cannot be set) is answered only. The worst-case upper-bound obtained by the miter and a threshold circuit checking that difference e is below threshold τ (formally $e \leq \tau$) was usually determined [36].

Estimation

The errors of approximate circuits described on e.g. Abstract Syntax Tree or RTL levels can be determined using statistical analysis [12]. This approach is suitable when the circuit consists of approximate arithmetic modules (such as adders or multipliers) with known error distributions. For example, if the error of the module can be described using a Gaussian model, then mean μ and variance ν represent the error. At the circuit level, these error characteristics coming from the modules are combined to the resulting error.

2.3 Cartesian Genetic Programming

Cartesian genetic programming (CGP) grew from a method of evolving digital circuits developed by Miller et al. in 1998 [17]. CGP especially differs from other genetic programming branches in (i) the solution representation and (ii) the search mechanism. The key ingredients of CGP are briefly introduced in the following paragraphs. Then, the use of CGP in the synthesis of approximate circuits is discussed. Detailed description of CGP in the digital circuit design is available in [16].

2.3.1 Circuit representation

The key part of CGP is a representation of candidate circuits. A candidate circuit is represented as a special integer netlist describing the constant number of components (N). These components (nodes) are organized in a two-dimensional grid of n_c columns and n_r rows ($N = n_c \cdot n_r$). The number of primary inputs and outputs of the circuit is denoted as n_i and n_o. The function of the nodes depends on the level of abstraction used in modeling, where logic gates and more complex components from the technology library are naturally supported. Every component has up to n_a inputs and n_b outputs. For example, if standard logic gates are used as components, $n_a = 2$ and $n_b = 1$.

A unique address is assigned to all primary inputs and to the outputs of all components to define an addressing system enabling circuit topologies to be specified. The primary inputs are labeled $0, 1, \ldots, n_i - 1$ and the components' outputs are labeled $n_i, n_i + 1, \ldots, n_i + n_b \cdot n_c \cdot n_r - 1$. As no feedback connections are allowed in the basic version of CGP, only acyclic graphs can be encoded. Because of that, only combinational circuits can be created.

The main feature of this encoding is that while the size of the chromosome is constant (for given n_a, n_b, n_o, n_r and n_c), the size of circuits represented by this chromosome is variable (from 0 to $n_c \cdot n_r - 1$ components can be involved) as some components can remain disconnected (see Fig. 2.3). This redundancy has been identified as a crucial property of the efficient search in the space of digital circuits [16].

Figure 2.3 shows a gate level, four-input (a_0, a_1, b_0, b_1) and four-output (o_0, o_1, o_2, o_3), circuit consisting of eight gates implementing the two-bit multiplication. This circuit is represented in the CGP grid with $n_c = n_r = 3$ and the outputs of the components are labeled $4, 5, \ldots, 12$. One component (the exclusive-or gate with the output labeled 10) is

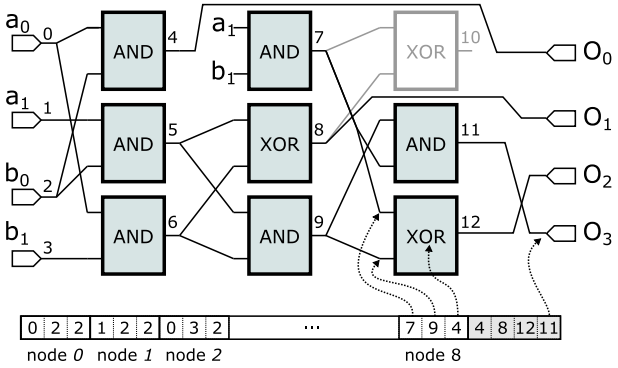

Figure 2.3: A two-bit multiplier represented in CGP with parameters: $n_i = n_o = 4, n_c = n_r = 3, n_a = 2, n_b = 1, \Gamma = \{0^{identity}, 1^{not}, 2^{and}, 3^{or}, 4^{xor}, 5^{nand}, 6^{nor}, 7^{xnor}, 8^{cont0}, 9^{const1}\}$.

inactive, i.e. no path from the output of the component to the primary output of the circuit exists.

In the integer netlist, each component is represented using $n_a + 1$ integers, where n_a integers specify destination addresses for its inputs and one integer refers to the set Γ containing all supported functions of the nodes. A component placed in the j-th column can obtain its input values v either from the primary inputs or from the components placed in previous columns, formally $v \leq n_i + (j-1) \cdot n_r \cdot n_b$.

The last part of the chromosome contains n_o integers specifying the connection of each primary output. The output can be connected either with the output of some node or with the primary input.

The whole circuit is then represented using the so-called chromosome (i.e. simplified integer netlist) whose size is

$$N_g = n_c \cdot n_r \cdot (n_a + 1) + n_o \quad \text{(integers)}. \tag{2.10}$$

2.3.2 Search algorithm

Every candidate circuit represents one design point in the design space. In CGP, new designs are created by introducing small random modifications to the chromosome. This operation is called the mutation and it typically modifies h integers of the chromosome. Note that all modifications must lead to valid circuits, i.e. only valid function codes and connections can be created.

Algorithm 1 presents the search method based on $(1 + \lambda)$ evolutionary strategy usually used for a single-objective circuit approximation by means of CGP [16]. The search algorithm can start either with a randomly generated initial population or with existing designs. The population size is $1 + \lambda$. After evaluating the initial population (i.e. scoring the circuit functionality and the cost while better circuits obtain higher scores), the following steps are repeated until the termination condition is not satisfied: (i) a new parent is selected; (ii) λ offspring circuits are created from the parent by means of mutation; (iii) the population is evaluated.

One mutation can affect either the component function, the component input connection, or the primary output connection. A mutation is called neutral if it does not affect the circuit's fitness. Neutral mutations are very important for the evolutionary design and optimization [16].

Algorithm 1: CGP optimization

Input: CGP parameters, fitness function, intial population P
Output: The highest scored individual and its fitness

```
1 EvaluatePopulation(P);
2 while ⟨terminating condition not satisfied⟩ do
3     α ← SelectHighest-scored-individual(P);
4     if fitness(α) ≥ fitness(p) then
5         p ← α;
6     P ← {p} ∪ {λ offspring of p created by mutation};
7     EvaluatePopulation(P);
8 return p, fitness(p);
```

The fitness function is typically application-specific. If the initial population is initialized with circuits satisfying the functional requirements, the objective of the evolutionary synthesis formulated in the fitness function is to improve non-functional parameters and keep the functionality unchanged.

2.3.3 Circuit approximation using CGP

In the context of approximate computing, three evolutionary approximation strategies (shown in Fig. 2.4) were developed.

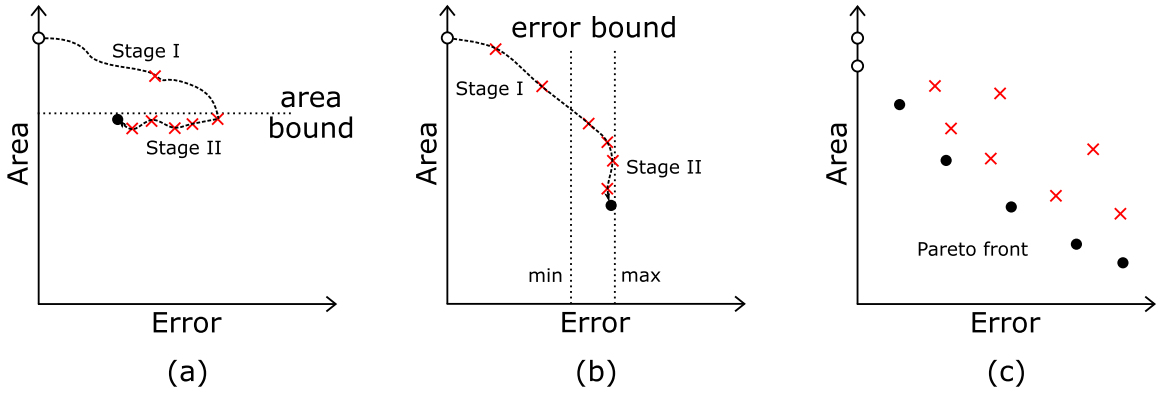

Figure 2.4: Evolutionary approximation strategies (a simplified situation for the area-error optimization): (a) resource-oriented, (b) error-oriented and (c) multi-objective. The white points denote the initial (accurate) circuit, the black are the desired approximate circuits and the crosses denote the valid candidate solutions (design points).

Resources-oriented method

CGP is used to minimize the error criterion under the assumption that only m_i components (gates) are available and m_i is lower than the minimal number of components (gates) needed to implement the accurate function [41].

The strategy is divided to two stages. In the first stage, the accurate initial solution is reduced to contain less than m_i components. It can be done using a random component

elimination algorithm or a suitable heuristic. In the second stage, the goal of CGP is to optimize the error while the number of components is kept lower than m_i. The main advantage is that the user can control the used area (and power consumption) precisely by means of m_i.

Error-oriented method

In the error-oriented method, the target error (e.g. the worst-case error) range, determined by e_{min} and e_{max}, is specified by the user. The goal is to optimize the number of components (or area or power consumption) while the error of the circuits is kept between target errors e_{min} and e_{max} [39]. This strategy is divided to two stages as well. In the first stage, the design objective (area in Fig. 2.4b) is minimized since the error is lower than e_{max}. We can assume that the increasing error typically leads to smaller area. In the second stage, the algorithm optimizes the design objective while the error of circuits kept in the target error range. If various tradeoffs between the error and the number of components are requested, CGP is executed several times with e_{max} as the parameter.

Multi-objective CGP

Compared to previous methods that employ a single-objective optimization (one fitness function with constraints), the multi-objective method allows to optimize the error and other key circuit parameters (area, delay and power consumption) together in one CGP run [8]. We are primarily interested in approximate circuits belonging to the *Pareto set* which contains the so-called *nondominated solutions*. For example, consider two circuits C1 and C2. Circuit C1 dominates another circuit C2 if: (1) C1 is no worse than C2 in all objectives, and (2) C1 is strictly better than C2 in at least one objective.

The multi-objective CGP represents candidate circuits using the standard CGP encoding. The search algorithm uses a modified variant of a multi-objective genetic algorithm, such as Non-dominated Sorting Genetic Algorithm (NSGA-II) [5].

The advantage of multi-objective CGP is that it re-constructs the Pareto front in each CGP generation and tries to cover all possible compromise solutions. Several accurate implementations can be used as a seed. A set of non-dominated circuits is the output of the multi-objective algorithm.

Chapter 3

Research summary

This chapter summarizes the research conducted in the thesis. In the Section 3.1, overall view on the research is given. Section 3.2 describes motivations and contributions of seven selected papers where the most important results of the research were published. Section 3.3 lists the remaining papers of the author related to the research topic that are not included in this thesis.

3.1 Overview

The goal of the research conducted in this doctoral thesis is to improve the evolutionary circuit design methodology based on Cartesian genetic programming (CGP) in order to obtain an advanced method for the approximate circuit design. The method should produce approximate circuits that: (i) show excellent tradeoffs between quality and power-consumption, (ii) have reasonable complexity, (iii) can be defined at various levels of description, and (iv) are applicable in the real-world applications.

In Paper I we presented an advanced simulation technique and a novel modification of CGP representation which is suitable for fast evaluation of transistor-level circuits. The proposed method was utilized in the evolutionary design and optimization of circuits at the transistor-level.

Due to the limited scalability of the transistor-level approach, the gate-level representation was used in CGP approximation of larger circuits. At the gate level, multi-objective CGP was employed for construction of a library of 8-bit approximate arithmetic circuits. This library contains hundreds of adders and multipliers, that are available online for download (Paper II). The approximate multipliers from the library helped us to investigate the properties of approximation in neural image classifiers and introduce a new constraint for the forward path approximation in neural networks. This constraint was implemented to the CGP approximation method which was utilized to evolve approximate multipliers for neural networks. The neural networks employing the proposed approximate multipliers showed a good tradeoff between the power consumption and the overall classification accuracy (Paper III).

In the aforementioned papers, circuit simulators evaluating the exact approximation error were utilized. However, the exact circuit simulation is not feasible for complex circuits. To deal with this issue, formal verification techniques were adapted in the next work. We included several optimization approaches to the verification methodology in order to approximate complex arithmetic circuits (Paper IV). In particular, we utilized SAT solvers

for analysis of the worst-case arithmetic error. An advanced BDD-based analysis was implemented in order to obtain the mean arithmetic error for complex adders. This algorithm was used in CGP to design approximate adders and subtractors that were later applied in discrete cosine transformation blocks of HEVC encoder (Paper V).

We investigated not only the possibilities on how to improve the scalability of the evolutionary approach, but also how to improve quality of the results. In the evolutionary approximation of multipliers, it was shown that the function set Γ covering all important components of the technology library (such as full adders and multiplexors) led to better results. As a side effect, the CGP with the function set including the more complex components led to better convergence compared to the CGP with the function set containing just one- and two-input logic gates (Paper VI).

The last presented work (Paper VII) shows that the evolutionary approximate synthesis conducted at RTL-level is capable of creating complex circuits. In particular, we evolved approximate median filters suitable for image and signal processing. In order to perform a data-independent and fast evaluation of candidate median filters, a new error metric was proposed. A significant improvement was obtained in terms of circuit complexity and quality with respect to CGP guided by the fitness function based on random simulation.

All results enabled us to deeper understand the relations among the error metrics, the level of circuit description and the search algorithm in the context of evolutionary approximations. We showed that evolutionary algorithms can be used for power-aware optimization of digital circuits described on various levels — from the lowest transistor netlist to the gate level description and the circuits utilizing complex building blocks.

3.2 Papers included in the thesis

This section presents details on the motivation and contributions for each paper together with the paper abstract.

3.2.1 Paper I

MRÁZEK Vojtěch and VAŠÍČEK Zdeněk. **Evolutionary Design of Transistor Level Digital Circuits using Discrete Simulation**. In: *Genetic Programming, 18th European Conference, EuroGP 2015.* Berlin: Springer, 2015, pp. 66-77. ISBN 978-3-319-16500-4.

Author participation: 50 %

Conference ranking: B (CORE[1]) / B1 (Qualis[2])

Motivation and contributions

The transistor level description is useful for the design and optimization of small circuits suitable for e.g. technology libraries specialized for approximate computing. In this paper, we presented a fundamental modification of the CGP representation of circuits described on the transistor level. Instead of single-output and typically two-input nodes supporting single-directional data flow, into this representation, we included *nmos* and *pmos transistors* and *junctions*. The junctions are necessary for the description of transistor-level circuits, but their usage can lead to cyclic connections that are unwanted in combinational circuits.

[1] `http://www.core.edu.au/conference-portal`

[2] `http://www.conferenceranks.com/`

The evaluation of candidate circuits was performed by means of the proposed discrete even-driven simulator. Six different discrete values are supported: "strong" as well as "degraded" zeros and ones, a high-impedance state and a forbidden state. The simulation engine stimulated candidate circuits with all possible input vectors and checked, whether the output values were identical with the specification, and no short-circuit connections or values oscillations (caused by cyclic connections) occurred during the simulation.

The proposed approach was employed in the automated design of accurate circuits that could be used as components of the technology library. In some cases, it allowed us to find the same solutions as those published in the literature. For example, an implementation of full-adder employing 14 transistors was rediscovered. In the contrast with genetic programming literature [42], where only small circuits (up to 10 transistors) were evolved, the proposed evolutionary design was able to handle circuits with up to 30 transistors due to the fast evaluation of candidate circuits.

Later, the work was extended with power consumption estimation of candidate circuits [22]. The numeric simulation models of transistors, typically used in power consumption calculation, are very complex and computationally expensive. Hence, we proposed an estimation algorithm that utilized precalculated power dissipations of pmos and nmos transistors for each input transition obtained by means of the SPICE simulator. In the candidate circuit evaluation process, all possible combinations of inputs were simulated using the proposed simulator. During the simulation, probabilities of transitions of all transistors were acquired. These probabilities and the precalculated table of power dissipations were used to estimate the total power consumption.

Since the discrete simulation ignores some facts (e.g. it does not care about a potential timing issues, especially for large circuits containing hundreds of transistors), the accurate SPICE simulator was used time to time to validate the functionality of candidate circuits [22]. If the validation did not pass, the evolutionary algorithm rolled back to the last valid solution.

In order to design approximate arithmetic circuits, we proposed to employ a two-stage design process [22]. In the first stage, the circuit was approximated on the gate level. Then, the resulting approximate circuit was optimized directly on the transistor level. Using this approach, we evolved optimized four-bit approximate multipliers that outperformed the gate-level approximations in terms of power consumption.

In this paper, a new approach suitable for the evolutionary design of digital circuits conducted directly on the transistor level was introduced. The novelty lies in a new circuit representation and a new simulation engine. The approach can be utilized to design or optimize low-power modules of technology libraries specialized for approximate computing.

Abstract

The objective of the paper is to introduce a new approach to the evolutionary design of digital circuits conducted directly at transistor level. In order to improve the time-consuming evaluation of candidate solutions, a discrete event-driven simulator was introduced. The proposed simulator operates on multiple logic levels to achieve reasonable trade-off between performance and precision. A suitable level of abstraction reflecting the behavior of real MOSFET transistors is utilized to minimize the production of incorrectly working circuits. The proposed approach is evaluated in the evolution of basic logic circuits having more than 20 transistors. The goal of the evolutionary algorithm is to design a circuit having the minimal number of transistors and exhibiting the minimal delay. In addition to that,

various parameter settings are investigated to increase the success rate of the evolutionary design.

3.2.2 Paper II

MRÁZEK Vojtěch, HRBÁČEK Radek, VAŠÍČEK Zdeněk and SEKANINA Lukáš. **EvoApprox8b: Library of Approximate Adders and Multipliers for Circuit Design and Benchmarking of Approximation Methods**. In: *Proc. of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Lausanne: European Design and Automation Association, 2017, oo. 258-261. ISBN 978-3-9815370-9-3.

Author participation: 25 %
Conference ranking: B (CORE) / A1 (Qualis)

Motivation and contributions

Due to the limited scalability of transistor-level approach, CGP at the gate-level representation was employed for construction of larger circuits. In this paper, a large library of 8-bit approximate arithmetic circuits was introduced. The approximate circuits were evolved using a multi-objective implementation of CGP based on NSGA-II algorithm proposed in [8]. It considered power consumption, area and mean absolute error as the design objectives. An exhaustive simulation utilizing 256-bit AVX instructions was employed in the candidate circuit evaluation. The simulator calculated the mean absolute error of the candidate circuit and the probability of being in logic zero and one for each CGP node output. These probabilities combined with parameters of logic functions from the technology library were used to estimate the power consumption.

The evolved circuits outperformed the circuits published in the literature. Hence, we constructed hardware as well as software models of the circuits and made them online. This library contains hundreds of approximate adders and multipliers that are available. The library was well accepted by the community, for example, 47 unique users from 13 countries visited the library websites in April 2018.

This paper presented the first large library of approximate arithmetic circuits that are available online and thus everyone can download them and apply in various applications. In addition to that, researchers designing new approximate circuits can use these models for comparison and benchmarking.

Abstract

Approximate circuits and approximate circuit design methodologies attracted a significant attention of researchers as well as industry in recent years. In order to accelerate the approximate circuit and system design process and to support a fair benchmarking of circuit approximation methods, we propose a library of approximate adders and multipliers called EvoApprox8b. This library contains 430 nondominated 8-bit approximate adders created from 13 conventional adders and 471 non-dominated 8-bit approximate multipliers created from 6 conventional multipliers. These implementations were evolved by a multi-objective Cartesian genetic programming. The EvoApprox8b library provides Verilog, Matlab and C models of all approximate circuits. In addition to standard circuit parameters, the error is given for seven different error metrics. The EvoApprox8b library is available at: `www.fit.vutbr.cz/research/groups/ehw/approxlib`.

3.2.3 Paper III

MRÁZEK Vojtěch, SARWAR Syed Shakib, SEKANINA Lukáš, VAŠÍČEK Zdeněk and ROY Kaushik. **Design of Power-Efficient Approximate Multipliers for Approximate Artificial Neural Networks**. In: *Proceedings of the 35th IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. Austin, TX: Association for Computing Machinery, 2016, pp. 811-817. ISBN 978-1-4503-4466-1.

Author participation: 55 %
Conference ranking: A (CORE) / A1 (Qualis)

Motivation and contributions

The image classification using neural networks (NN) is currently a very popular topic. Since the recognition task has been identified as error-resilient, the implementations of neural classifiers are after approximated.

This research started with applying approximate multipliers from the EvoApprox8b benchmark to the neural classifier in order to reduce its power consumption. We observed that the NN with approximate multipliers exhibits significantly worse classification accuracy than the original NN. We analyzed this problem and recognized that almost 80% of multiplications in NN should produce zero. We hypothesized that multiplication by zero must be accurate, but multiplication of non-zero operands can be approximated. We introduced this "accurate zero-multiplication" constraint into the error oriented evolutionary approximation and evolved 7-bit and 11-bit approximate unsigned multipliers. Finally, the evolved multipliers were extended using one-complement to 8- and 12-bit signed ones.

The evolved multipliers were applied in two NNs (a multilayer perceptron and convolutional LeNet-5 network) performing two different benchmark tasks — handwritten number recognition (MNIST dataset) and Street-View House Number classification (SVHN dataset). We recognized that (despite the NNs were trained) a re-training process is able to adapt the pre-trained NN to the inaccurate multipliers, and thus improve the classification accuracy of approximate NNs up to tens of percent. The resulting NNs employing approximate multipliers exhibited significant power consumption reduction and classification accuracy comparable with solutions presented in the literature.

In this application, a specific constraint related to approximate multiplication in NNs was discovered. This paper showed that the proposed methodology of evolutionary approximation can easily handle very specific constraints. Multipliers satisfying this constraint were evolved and applied in the neural networks. The resulting neural networks show a good tradeoff between the power consumption and the overall classification accuracy.

Abstract

Artificial neural networks (NN) have shown a significant promise in difficult tasks like image classification or speech recognition. Even well-optimized hardware implementations of digital NNs show significant power consumption. It is mainly due to non-uniform pipeline structures and inherent redundancy of numerous arithmetic operations that have to be performed to produce each single output vector. This paper provides a methodology for the design of well-optimized power-efficient NNs with a uniform structure suitable for hardware implementation. An error resilience analysis was performed in order to determine key constraints for the design of approximate multipliers that are employed in the resulting structure of NN. By means of a search-based approximation method, approximate multi-

pliers showing desired tradeoffs between the accuracy and implementation cost were created. Resulting approximate NNs, containing the approximate multipliers, were evaluated using standard benchmarks (MNIST dataset) and a real-world classification problem of Street-View House Numbers. Significant improvement in power efficiency was obtained in both cases with respect to regular NNs. In some cases, 91% power reduction of multiplication led to classification accuracy degradation of less than 2.80%. Moreover, the paper showed the capability of the back-propagation learning algorithm to adapt with NNs containing the approximate multipliers.

3.2.4 Paper IV

ČEŠKA Milan, MATYÁŠ Jiří, MRÁZEK Vojtěch, SEKANINA Lukáš, VAŠÍČEK Zdeněk and VOJNAR Tomáš. **Approximating Complex Arithmetic Circuits with Formal Error Guarantees: 32-bit Multipliers Accomplished**. In: *Proceedings of 36th IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. Irvine, CA: Institute of Electrical and Electronics Engineers, 2017, pp. 416-423. ISBN 978-1-5386-3093-8.

Author participation: 17 %
Conference ranking: A (CORE) / A1 (Qualis)

Motivation and contributions

In Paper II and Paper III, the evolutionary approximation of arithmetic circuits such as multipliers and adders was presented. The error was determined exactly in both cases to give the potential users a guarantee regarding the worst-case error. Hence, we used a highly optimized parallel (vectorized) circuit simulator to evaluate the quality of candidate circuits. However, this method is not feasible for exact evaluating of complex circuits because the number of input combinations grow exponentially with increasing the number of inputs. In order to address this issue and approximate larger circuits, we proposed to employ a SAT solver in the fitness function.

We proposed two novel approaches to accelerate the relaxed equivalence checking. One of the key components of the method is the miter circuit that verifies, whether the difference d between the approximate output and the accurate output is below a threshold τ (formally $|d| \leq \tau$). We developed an advanced miter construction that replaced the standard absolute value calculation. The proposed formula $(d \geq 0 \wedge d \leq \tau) \vee (d < 0 \wedge d \geq -\tau)$ reduced the number of SAT clauses and thus the evaluation time. Another crucial improvement lies in the proposed verifiability-driven search strategy. The trick is to control the time which the SAT solver has to decide whether a candidate multiplier satisfies a given error bound or not. This solution allowed us to skip candidate multipliers that are not easily verifiable.

These approaches enabled to significantly improve the scalability limits of the evolutionary circuit approximation. We constructed a large dataset of approximate multipliers (with up to 32-bit inputs) and approximate adders (with up to 128-bit inputs).

In this paper, formal relaxed equivalence checking was employed in the approximation of complex arithmetic circuits. Since the checking can be very complicated and time-consuming for specific circuits, we included several optimization techniques to the verification methodology. In particular, we utilized CGP with SAT solvers to approximate large arithmetic circuits. By means of this approach, we significantly improved the scalability of the evolutionary approximation process.

Abstract

We present a novel method allowing one to approximate complex arithmetic circuits with formal guarantees on the approximation error. The method integrates in a unique way formal techniques for approximate equivalence checking into a search-based circuit optimization algorithm. The key idea of our approach is to employ a novel search strategy that drives the search towards promptly verifiable approximate circuits. The method was implemented within the ABC tool and extensively evaluated on functional approximation of multipliers (with up to 32-bit operands) and adders (with up to 128-bit operands). Within a few hours, we constructed a high-quality Pareto set of 32-bit multipliers providing trade-offs between the circuit error and size. This is for the first time when such complex approximate circuits with formal error guarantees have been derived, which demonstrates an outstanding performance and scalability of our approach compared with existing methods that have either been applied to the approximation of multipliers limited to 8-bit operands or statistical testing has been used only. Our approach thus significantly improves capabilities of the existing methods and paves a way towards an automated design process of provably-correct circuit approximations.

3.2.5 Paper V

VAŠÍČEK Zdeněk, MRÁZEK Vojtěch and SEKANINA Lukáš. **Towards Low Power Approximate DCT Architecture for HEVC Standard**. In: *Proc. of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Lausanne: European Design and Automation Association, 2017, pp. 1576-1581. ISBN 978-3-9815370-9-3.

Author participation: 40 %

Conference ranking: B (CORE) / A1 (Qualis)

Motivation and contributions

Discrete cosine transformation (DCT) is a very important part of widely employed HEVC (H.265) encoders as well as decoders. The goal of this work was to reduce the power consumption of DCT blocks in HEVC coders and encoders while keeping reasonable PSNR of the video processing.

The transformation can be implemented as a multiplication of inputs by a constant matrix. The most effective approach is to utilize multiplier-less constant multipliers (MCMs). Since adders and subtractors are the key components of MCMs, these arithmetic circuits were approximated by means of the proposed CGP approximation algorithm. Three approximate MCM blocks were proposed. Due to the increasing bitwidth of adders/subtractors in MCMs, 16 different instances of adders and subtractors were approximated.

In this work, formal relaxed-equivalence checking was employed in CGP. In contrast with the SAT-based solution, we were interested not only in the worst-case analysis but also in the average-case error. Unfortunately, the SAT-based approach cannot be used to determine the average error. Hence an advanced BDD-based analysis algorithm was developed and implemented in order to obtain the mean and the worst-case arithmetic errors for candidate circuits. The proposed BDD-based analysis was used in the error evaluation in the error-oriented CGP approximation algorithm. Apart from the error analysis, the BDDs were used to determine the switching activity of each gate because they provided the information about the probability that a gate output is in logic one. The switching activity and the information from technology library (liberty file) were used in power consumption

estimation of candidate circuits. The resulting HEVC implementations with approximate circuits in DCT blocks showed better quality/power tradeoffs than relevant implementations available in the literature.

In this paper, we proposed advanced algorithms for the mean and the worst-case error analysis with BDDs. We implemented a fast power-consumption estimation based on BDDs which was employed in CGP. By means of the proposed approach, new designs of complex approximate adders and subtractors applicable in an innovative application were developed.

Abstract

Video processing performed directly on IoT nodes is one of the most performance as well as energy demanding applications for current IoT technology. In order to support real-time high-definition video, energy-reduction optimizations have to be introduced at all levels of the video processing chain. This paper deals with an efficient implementation of Discrete Cosine Transform (DCT) blocks employed in video compression based on the High Efficiency Video Coding (HEVC) standard. The proposed multiplier-less 4-input DCT implementations contain approximate adders and subtractors that were obtained using genetic programming. In order to manage the complexity of evolutionary approximation and provide formal guarantees in terms of errors of key circuit components, the worst and average errors were determined exactly by means of Binary decision diagrams. Under conditions of our experiments, approximate 4- input DCTs show better quality/power tradeoffs than relevant implementations available in the literature. For example, 25% power reduction for the same error was obtained in comparison with a recent highly optimized implementation.

3.2.6 Paper VI

MRÁZEK Vojtěch, VAŠÍČEK Zdeněk and HRBÁČEK Radek. The Role of Circuit Representation in Evolutionary Design of Energy-Efficient Approximate Circuits. *IET Computers & Digital Techniques*. Stevenage: The Institution of Engineering and Technology, (to appear), p. 11. ISSN 1751-8601.

Author participation: 40 %
Impact factor: 0.515 (Q3)

Motivation and contributions

The fitness function is not the only component that determines the quality of the obtained results. The representation of circuits, especially the set of possible node functions (Γ) has a great impact not only on the scalability but also on the quality of obtained approximate circuits and the convergence of the evolutionary design process (i.e. the number of generations needed to find an acceptable solution).

In this paper, we investigate the role of the function set Γ and thereby the number of inputs n_a and outputs n_b of CGP nodes. Two settings were compared: (i) CGP nodes that can act as standard 1- or 2-input gates (*gate-based representation*) and (ii) CGP nodes that have 3 inputs and 2 outputs and can act as standard gates as well as more complex blocks such as full-adders (*cell-based representation*). The circuits were approximated by the error-oriented CGP guided by estimated power consumption.

The evolved circuits were subsequently synthetized (technology mapped) by means of a commercial tool. Then, the quality of the synthetized circuits (power, area and delay) was

evaluated. The circuits showed better quality for the same target error when we employed cell-based representation in the CGP. In addition to that, the convergence speed of the approximate EA-based logic synthesis was analyzed. Although the cell-based representation led to a larger search-space, in the case of approximate multipliers, better results were obtained with respect to the gate-based representation.

In the evolutionary approximation of multipliers, it was shown that the cell-based representation including the efficient implementations of various components such as full adders and multiplexors leads to better results than the representation employing the function set containing just one- and two-input logic gates only. This work illustrated the importance of choosing the right circuit representation in the approximate multiplier synthesis task.

Abstract

Circuit approximation has been introduced in recent years as a viable method for constructing energy efficient electronic systems. An open problem is how to effectively obtain approximate circuits showing good compromises between key circuit parameters – the error, power consumption, area and delay. The use of evolutionary algorithms in the task of circuit approximation has led to promising results. Unfortunately, only relatively small circuit instances have been tackled because of the scalability problems of the evolutionary design method. This paper demonstrates how to push the limits of the evolutionary design by choosing a more suitable representation on the one hand and a more efficient fitness function on the other hand. In particular, we show that employing full adders as building blocks leads to more efficient approximate circuits. We focused on the approximation of key arithmetic circuits such as adders and multipliers. While the evolutionary design of adders represents a rather easy benchmark problem, the design of multipliers is known to be one of the hardest problems. We evolved a comprehensive library of energy-efficient 12-bit multipliers with a guaranteed worst-case error. The library consists of 65 Pareto dominant solutions considering power, delay, area and error as design objectives.

3.2.7 Paper VII

VAŠÍČEK Zdeněk and MRÁZEK Vojtěch. **Trading between Quality and Non-functional Properties of Median Filter in Embedded Systems**. *Genetic Programming and Evolvable Machines*. Berlin: Springer, 2017, vol. 18, no. 1, pp. 45-82. ISSN 1389-2576.

Author participation: 50 %

Impact factor: 1.514 (Q2)

Motivation and contributions

The last paper is devoted to the approximation of circuits represented using a high-level representation. In particular, we approximated median filters implemented as median networks described at the level of so-called *compare-and-swap* components. The approximate median filters were already investigated in the literature [41], but the approximation error was calculated using a randomly generated set of input data. However, the calculated error depends on the input dataset. We proposed a new approach enabling to evaluate the quality of median networks. Firstly, we proposed an error metric that is data independent. Secondly, we introduced a formal method how to determine the error. The proposed metric, the so-called *distance error*, is calculated using the *permutation principle* introduced in this paper.

This new metric enabled to evaluate approximate median filters in $n!$ steps, where n is the number of inputs. The proposed distance error metric was employed in the resource-oriented CGP. The goal of the algorithm was to optimize the average distance error while keeping the number of active CGP nodes below a certain value. In contrast with the gate-level approximations, all operations were conducted with k-bit components. The function set Γ covered all functions that are important in the filtering: k-bit minimum, k-bit maximum and a junction connecting one of the inputs directly to the output of the node.

The evolved filters were applied in two tasks — image filtering of salt-and-pepper noise and processing of the data generated by an accelerator sensor. In addition to that, proposed approximated median structures were transferred to the C-code and implemented as approximate software for microcontrollers.

This paper demonstrated how to approaximate circuit described on a higher level of abstraction. Since the evolved filters were evaluated on microcontrollers, the paper also showed that the proposed approach is applicable for software approximations as well.

Abstract

Genetic improvement has been used to improve functional and non-functional properties of software. In this paper, we propose a new approach that applies a genetic programming (GP)-based genetic improvement to trade between functional and non-functional properties of existing software. The paper investigates possibilities and opportunities for improving non-functional parameters such as execution time, code size, or power consumption of median functions implemented using comparator networks. In general, it is impossible to improve non-functional parameters of the median function without accepting occasional errors in results because optimal implementations are available. In order to address this issue, we proposed a method providing suitable compromises between accuracy, execution time and power consumption. Traditionally, a randomly generated set of test vectors is employed so as to assess the quality of GP individuals. We demonstrated that such an approach may produce biased solutions if the test vectors are generated inappropriately. In order to measure the accuracy of determining a median value and avoid such a bias, we propose and formally analyze new quality metrics which are based on the positional error calculated using the permutation principle introduced in this paper. It is shown that the proposed method enables the discovery of solutions which show a significant improvement in execution time, power consumption, or size with respect to the accurate median function while keeping errors at a moderate level. Non-functional properties of the discovered solutions are estimated using data sets and validated by physical measurements on physical microcontrollers. The benefits of the evolved implementations are demonstrated on two real-world problems—sensor data processing and image processing. It is concluded that data processing software modules offer a great opportunity for genetic improvement. The results revealed that it is not even necessary to determine the median value exactly in many cases which helps to reduce power consumption or increase performance. The discovered implementations of accurate, as well as approximate median functions, are available as C functions for download and can be employed in a custom application (`http://www.fit.vutbr.cz/research/groups/ehw/median`).

3.2.8 Author's contributions to selected papers

The papers presented in this thesis were created in collaboration with the Evolvable Hardware Group led by my supervisor prof. Sekanina. Preliminary results of Paper III were

investigated during my visit to Purdue University, IN, USA. Although all co-authors contributed to the papers, this section explicitly summarizes author's contribution to selected papers.

- **Paper I** — development of discrete simulator, novel circuit representation, implementation of the evolutionary algorithm and experimental evaluation.
- **Paper II** — development of power estimation method, software models of circuits and the library front-end.
- **Paper III** — discovering the zero multiplication constraint, implementation of the evolutionary algorithm and experimental evaluation.
- **Paper IV** — development of the error-oriented method, miter simplification and experimental evaluation.
- **Paper V** — development of BDD-based error evaluation, power-estimation method and implementation of evolutionary algorithm.
- **Paper VI** — development of power estimation functions and experimental evaluation.
- **Paper VII** — proposing the error metric, implementation of the evolutionary algorithm and experimental evaluation.

3.3 List of other publications

2018

- ČEŠKA Milan, MATYÁŠ Jiří, MRÁZEK Vojtěch, SEKANINA Lukáš, VAŠÍČEK Zdeněk a VOJNAR Tomáš. ADAC: Automated Design of Approximate Circuits. In: *Proceedings of 30th International Conference on Computer Aided Verification (CAV'18).*

  Author participation: 17 %

  Conference ranking: A* (CORE) / A1 (Qualis)

- MRÁZEK Vojtěch and VAŠÍČEK Zdeněk. Evolutionary Design of Large Approximate Adders Optimized for Various Error Criteria. In: *GECCO Companion '18 Proceedings of the Companion Publication of the 2018 on Genetic and Evolutionary Computation Conference.* Kyoto: Association for Computing Machinery, 2018, p. 2. ISBN 978-1-4503-5764-7.

  Author participation: 50 %

- MRÁZEK Vojtěch, SÝS Marek, VAŠÍČEK Zdeněk, SEKANINA Lukáš and MATYÁŠ Václav. Evolving Boolean Functions for Fast and Efficient Randomness Testing. In: *GECCO '18 Proceedings of the 2018 on Genetic and Evolutionary Computation Conference.* Kyoto: Association for Computing Machinery, 2018, p. 8. ISBN 978-1-4503-5618-3.

  Author participation: 40 %

  Conference ranking: A (CORE) / A1 (Qualis)

2017

- MRÁZEK Vojtěch and VAŠÍČEK Zdeněk. Parallel Optimization of Transistor Level Circuits using Cartesian Genetic Programming. In: *GECCO Companion '17 Proceedings of the Companion Publication of the 2017 on Genetic and Evolutionary Computation Conference*. Berlin: Association for Computing Machinery, 2017, pp. 1849-1856. ISBN 978-1-4503-4939-0.

  Author participation: 90 %

- SEKANINA Lukáš, VAŠÍČEK Zdeněk and MRÁZEK Vojtěch. Approximate Circuits in Low-Power Image and Video Processing: The Approximate Median Filter.*Radioengineering*. 2017, vol. 26, no. 3, pp. 623-632. ISSN 1210-2512.

  Author participation: 30 %

  Impact factor: 0.945 (Q3)

- SHAFIQUE Muhammad, HAFIZ Rehan, JAVED Muhammad Usama, ABBAS Sarmad, SEKANINA Lukáš, VAŠÍČEK Zdeněk and MRÁZEK Vojtěch. Adaptive and Energy-Efficient Architectures for Machine Learning: Challenges, Opportunities, and Research Roadmap. In: *2017 IEEE Computer Society Annual Symposium on VLSI*. Los Alamitos: IEEE Computer Society Press, 2017, pp. 627-632. ISBN 978-1-5090-6762-6.

  Author participation: 14 %

  Conference ranking: B1 (Qualis)

2016

- HRBÁČEK Radek, MRÁZEK Vojtěch and VAŠÍČEK Zdeněk. Automatic Design of Approximate Circuits by Means of Multi-Objective Evolutionary Algorithms. In: *Proceedings of the 11th International Conference on Design & Technology of Integrated Systems in Nanoscale Era*. Istanbul: Istanbul Sehir University, 2016, pp. 239-244. ISBN 978-1-5090-0335-8.

  Author participation: 25 %

- MRÁZEK Vojtěch and VAŠÍČEK Zdeněk. Automatic Design of Arbitrary-Size Approximate Sorting Networks with Error Guarantee. In: *Power and Timing Modeling, Optimization and Simulation (PATMOS), 2016 26rd International Workshop on*. Bremen: Institute of Electrical and Electronics Engineers, 2016, pp. 221-228. ISBN 978-1-5090-0733-2.

  Author participation: 50 %

  Conference ranking: B2 (Qualis)

- MRÁZEK Vojtěch. Evoluční snižování příkonu: Od obvodů na úrovni tranzistorů po neuronové sítě na čipu. In: *Počítačové architektury a diagnostika PAD 2016*. Bořetice: Faculty of Information Technology BUT, 2016, pp. 61-64. ISBN 978-80-214-5376-0.

  Author participation: 100 %

- NEVORAL Jan, RŮŽIČKA Richard and MRÁZEK Vojtěch. Evolutionary Design of Polymorphic Gates Using Ambipolar Transistors. In: *2016 IEEE Symposium Series on Computational Intelligence*. Athens: Institute of Electrical and Electronics Engineers, 2016, pp. 1-8. ISBN 978-1-5090-4240-1.

  Author participation: 20 %

  Conference ranking: B5 (Qualis)

- VAŠÍČEK Zdeněk, MRÁZEK Vojtěch and SEKANINA Lukáš. Evolutionary Functional Approximation of Circuits Implemented into FPGAs. In: *2016 IEEE Symposium Series on Computational Intelligence*. Athens: Institute of Electrical and Electronics Engineers, 2016, pp. 1-8. ISBN 978-1-5090-4240-1.

  Author participation: 20 %

  Conference ranking: B5 (Qualis)

2015

- MRÁZEK Vojtěch and VAŠÍČEK Zdeněk. Automatic Design of Low-Power VLSI Circuits: Accurate and Approximate Multipliers. In: *Proceedings of 13th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. Porto: Institute of Electrical and Electronics Engineers, 2015, pp. 106-113. ISBN 978-1-4673-8299-1.

  Author participation: 50 %

  Conference ranking: C (CORE) / B2 (Qualis)

- MRÁZEK Vojtěch, VAŠÍČEK Zdeněk and SEKANINA Lukáš. Evolutionary Approximation of Software for Embedded Systems: Median Function. In: *GECCO Companion '15 Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*. New York: Association for Computing Machinery, 2015, pp. 795-801. ISBN 978-1-4503-3488-4.

  Author participation: 40 %

- MRÁZEK Vojtěch. Evoluční návrh nízkopříkonových obvodů. In: *Počítačové architektury a diagnostika PAD 2015*. Zlín: Faculty of Applied Informatics, Tomas Bata University in Zlín, 2015, pp. 1-6. ISBN 978-80-7454-522-1.

  Author participation: 100 %

2014

- MRÁZEK Vojtěch and VAŠÍČEK Zdeněk. Acceleration of Transistor-Level Evolution using Xilinx Zynq Platform. In: *2014 IEEE International Conference on Evolvable Systems Proceedings*. Piscataway: Institute of Electrical and Electronics Engineers, 2014, pp. 9-16. ISBN 978-1-4799-4480-4.

  Author participation: 20 %

  Conference ranking: B5 (Qualis)

- MRÁZEK Vojtěch. Akcelerace evolučního návrhu digitálních obvodů na úrovni tranzistorů s využitím platformy Zynq. In: *Proceedings of the 20th Student Conference, EEICT 2014*. Brno: Brno University of Technology, 2014, pp. 229-231. ISBN 978-80-214-4923-7.

  Author participation: 100 %

Chapter 4

Conclusions

This chapter summarizes the challenges and the contributions of the thesis. Then the outcomes for the research community are discussed. Finally, directions of the future research are proposed.

4.1 Challenges and Methodology

The biggest challenge of the work was to combine two, clearly independent, topics: the design of VLSI circuits and the evolutionary design. Although the evolutionary design has been successfully employed in the digital circuit design, only a few papers had been published on leading hardware conferences. During this work, the gap between hardware design and evolutionary optimization seems to get narrowed, because some of our papers have been accepted on the top conferences such as ICCAD or DATE. Two main approaches helped us in this direction. The first was the considering of real hardware parameters such as area and power consumption in the design process. The second was that we focused on approximate circuits and their applications.

In addition to that, we had to deal with the scalability of evolutionary design, mostly with the scalability of evaluation. We performed several crucial optimizations that led to the fast circuit parameters determination. There were two contradictory requirements in the evaluation of the circuits — the speed and the accuracy. Although various accurate approaches for the power and area estimation had been implemented in synthesis tools, we had to propose and implement own algorithms showing good tradeoffs between the accuracy and the computational intensity. Similarly, the standard equivalence checking algorithms are not sufficient for approximate computing and relaxed equivalence checking algorithms had to be implemented and optimized.

In this work, an automated approximation methodology was proposed for digital circuits. This methodology can work on various level of description and outperforms the manual and systematic approaches. This work brought new ideas for the hardware as well as the evolutionary community. The main contribution for the hardware community is that search-based algorithms can provide better results than the state-of-the-art methods for approximation of digital circuits and various application-specific error metrics or constraints can easily be considered in the design. The new contributions for the evolutionary computing community are: (i) the advanced fitness evaluation can significantly improve the scalability of evolutionary circuit design, and (ii) considering the state-of-the-art solutions

as starting points allows to optimize more complex circuits than the evolutionary design from scratch.

4.2 Contributions

This section summarizes the contributions of my research to automated approximate synthesis on the selected levels of circuit description.

Contributions to the automated synthesis of transistor-level circuits We improved the CGP representation to easily handle the circuits described on the transistor-level. For circuit evaluation, a fast multi-valued discrete simulation was developed, implemented and evaluated. This approach allowed us to design a human-competitive one-bit full-adder and other competitive digital primitives [23]. The discrete simulator was also accelerated on an FPGA [21]. Since the proposed discrete simulator does not capture all relevant physical phenomena, timing issues occurred during the simulation of large circuits. A novel combination of the less accurate, but fast discrete simulation with the accurate, but slow numeric SPICE simulation was introduced [22]. Finally, a two-stage algorithm combining gate-level approximation and transistor-level optimization was introduced [22, 25].

Contributions to the automated synthesis of gate-level circuits In the synthesis of circuits described on the gate-level, we focused on approximation of arithmetic circuits. The objective was to provide circuits showing better tradeoffs between power consumption and error than existing methodologies. We adopted the error-oriented CGP and evolved complex approximate adders and multipliers. These circuits were applied in two real-world applications. We showed, that image classification using neural networks [20] as well as DCT transformation in HEVC encoder [38] can be implemented with significantly reduced power consumption while the error is only slightly worsened.

As a part of these case studies, we utilized and improved three main approaches for the error evaluation — parallel simulation (vectorization) [19, 8, 20], BDD-based formal verification [38, 26] and SAT-based formal verification [6].

In order to obtain the worst-case error of a complex arithmetic circuit, the problem can be formulated as a satisfiability problem and solved with a SAT solver. However, this approach does not scale for multipliers. Hence, we introduced a verifiability driven CGP in which the SAT solver can use only a very short time to prove or disprove the submitted formula representing a candidate solution. Our method enables to generate and evaluate more candidate designs in a single evolutionary run than if there is no time limit for SAT solver [6].

Contributions to the automated synthesis of RTL circuits We selected the design of approximate median filters as an example of the approximate synthesis of circuits described using complex RTL modules. We introduced a new error metric called *distance error* whose main property is that it is a deterministic, input-data independent metric [37]. This metric can be utilized in the approximate sorting-networks characterization [24].

Moreover, it was possible to transform the evolved median filters described on the RTL level to a program for embedded systems (by transformation of RTL modules to software functions). We thus proposed fast (and therefore power-efficient) approximate median

filters for microprocessors. We also showed that the EA-based approximations define a new direction in the promising topic called *genetic improvement of software* [27].

Summary of contributions In Chapter 1, we defined the hypothesis, that *it is possible to reduce power dissipation of digital combinational circuits described on various levels of abstraction using EA-based circuit approximation algorithms.* We selected four different real-world applications described on three levels of abstraction. For each target application, the following steps were taken. At first, an effective power consumption estimation method was developed. This method was employed in the EA-based search algorithm. In order to speed up the circuit synthesis, the evaluation of candidate circuits was accelerated. Table 4.1 summarizes the power-estimation method, the modification of EA-based algorithm and the acceleration technique used in selected applications.

Table 4.1: Overview of approaches taken for selected real-world applications

Target application	Circuit representation	Power estimation	Search algorithm	Fitness acceleration
Tech. library	transistors	switching activity	CGP (two stages)	discrete simulation
Neural networks	gates	area-correlation	error-oriented CGP	vectorization
DCT for HEVC	gates	dynamic power	error-oriented CGP	BDD verification
Median filters	RTL modules	area-correlation	resources-oriented CGP	permutation principle

Finally, the obtained results were compared with conventional solutions. We found the transistor-level circuits realizing approximate multiplication more effectively than circuits obtained using a simple CMOS gate substitution. The neural network image classifiers employing the proposed approximate multipliers traded the power consumption for the overall classification accuracy better than a manual approximation. DCT blocks as well as median filter approximations had lower power dissipation, but acceptable approximation error in comparison with the approximate multiplier-less multipliers. Approximate median filters showed significant power savings with respect to the optimal accurate solutions. Hereby, based on these results, we can confirm, that EA-based approximation algorithms can create circuits with lower power consumption than conventional approaches.

4.3 Developed libraries available online

As a part of the research, we have developed the *EvoApprox8B* library. This library contains approximate hardware (HDL) and software (C/C++) models of arithmetic circuits[1]. The EvoApprox8B library currently contains hundreds of approximate 8-bit adders and 8-, 12-, 16- and 32-bit multipliers. These circuits can be used either for benchmarking of circuit approximation tools or directly in user applications such as image filters, neural network accelerators or special big-data processing systems.

We also made online the SPICE netlists of twelve 4-bit approximate multipliers that were obtained using CGP operating on the transistor level description[2]. The software implementations realizing evolved approximate median filters were also published online[3].

[1] `https://www.fit.vutbr.cz/research/groups/ehw/approxlib`
[2] `http://www.fit.vutbr.cz/~imrazek/euc2015/`
[3] `http://www.fit.vutbr.cz/~imrazek/median2015/`

4.4 Future research directions

This thesis dealt with application-oriented approximation of digital circuits. There are many different ways presented in the following list how to further develop the research results.

- The evolutionary optimization of transistor-level circuits could be used for non-conventional technologies such as ambipolar transistors.
- A comprehensive library of arithmetic circuits has been developed. These circuits could be employed in other real-world applications and enable to determine the most significant functional criteria influencing the accuracy of the target application.
- The SAT-based verification with termination conditions for SAT solver was very successful for evolutionary approximation of arithmetic circuits. Additional improvement could be obtained by introducing an adaptive termination strategy.
- Error-calculation algorithms employing BDD structures could be further improved by an advanced miter construction without absolute values calculators.
- New applications of approximate sorting networks and median filters such as statistic indicators in big-data datasets could be proposed and evaluated.

Bibliography

[1] Chan, W. T. J.; Kahng, A. B.; Kang, S.; et al.: Statistical analysis and modeling for error composition in approximate computation circuits. In *2013 IEEE 31st International Conference on Computer Design (ICCD)*. Oct 2013. ISSN 1063-6404. pp. 47–53.

[2] Chen, T.-H.; Alaghi, A.; Hayes, J. P.: Behavior of stochastic circuits under severe error conditions. *it - Information Technology*. vol. 56, no. 4. jan 2014.

[3] Chippa, V. K.; Chakradhar, S. T.; Roy, K.; et al.: Analysis and characterization of inherent application resilience for approximate computing. In *The 50th Annual Design Automation Conference 2013, DAC'13*. ACM. 2013. pp. 1–9.

[4] Coudert, O.: Gate sizing for constrained delay/power/area optimization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. vol. 5, no. 4. Dec 1997: pp. 465–472. ISSN 1063-8210.

[5] Deb, K.; Pratap, A.; Agarwal, S.; et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. vol. 6, no. 2. Apr 2002: pp. 182–197. ISSN 1089-778X.

[6] Češka, M.; Matyaš, J.; Mrazek, V.; et al.: Approximating complex arithmetic circuits with formal error guarantees: 32-bit multipliers accomplished. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. Nov 2017. pp. 416–423.

[7] Esmaeilzadeh, H.; Sampson, A.; Ceze, L.; et al.: Neural Acceleration for General-Purpose Approximate Programs. In *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO-45. Washington, DC, USA: IEEE Computer Society. 2012. ISBN 978-0-7695-4924-8. pp. 449–460.

[8] Hrbacek, R.; Mrazek, V.; Vasicek, Z.: Automatic design of approximate circuits by means of multi-objective evolutionary algorithms. In *2016 International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS)*. April 2016. pp. 1–6.

[9] Jiang, H.; Liu, C.; Liu, L.; et al.: A Review, Classification, and Comparative Evaluation of Approximate Arithmetic Circuits. *J. Emerg. Technol. Comput. Syst.*. vol. 13, no. 4. August 2017: pp. 60:1–60:34. ISSN 1550-4832.

[10] Kahng, A. B.; Liu, B.; Xu, X.: Constructing Current-Based Gate Models Based on Existing Timing Library. In *Proceedings of the 7th International Symposium on Quality Electronic Design*. ISQED '06. Washington, DC, USA: IEEE Computer Society. 2006. ISBN 0-7695-2523-7. pp. 37–42.

[11] Kumar, R.; Liu, Z.; Kursun, V.: TECHNIQUE FOR ACCURATE POWER AND ENERGY MEASUREMENT WITH THE COMPUTER-AIDED DESIGN TOOLS. *Journal of Circuits, Systems and Computers*. vol. 17, no. 03. 2008: pp. 399–421.

[12] Li, C.; Luo, W.; Sapatnekar, S. S.; et al.: Joint Precision Optimization and High Level Synthesis for Approximate Computing. In *Proceedings of the 52Nd Annual Design Automation Conference*. DAC '15. New York, NY, USA: ACM. 2015. ISBN 978-1-4503-3520-1. pp. 104:1–104:6.

[13] Liu, G.; Zhang, Z.: Statistically certified approximate logic synthesis. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. Nov 2017. pp. 344–351.

[14] Machado, F.; Riesgo, T.; Torroja, Y.: A Method for Switching Activity Analysis of VHDL-RTL Combinatorial Circuits. In *Proceedings of the 16th International Conference on Integrated Circuit and System Design: Power and Timing Modeling, Optimization and Simulation*. PATMOS'06. Berlin, Heidelberg: Springer-Verlag. 2006. ISBN 3-540-39094-4, 978-3-540-39094-7. pp. 645–657.

[15] Mead, C.; Conway, L.: *Introduction to VLSI Systems*. Philiphines: Addison-Wesley. 1980. ISBN 978-0201043587.

[16] Miller, J. F.: *Cartesian Genetic Programming*. Springer-Verlag. 2011.

[17] Miller, J. F.; Thomson, P.; Fogarty, T.: *Designing Electronic Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study*. Wiley. 1998. pp. 105–131.

[18] Mittal, S.: A Survey of Techniques for Approximate Computing. *ACM Comput. Surv.*. vol. 48, no. 4. March 2016: pp. 62:1–62:33. ISSN 0360-0300.

[19] Mrazek, V.; Hrbacek, R.; Vasicek, Z.; et al.: EvoApprox8b: Library of Approximate Adders and Multipliers for Circuit Design and Benchmarking of Approximation Methods. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*. March 2017. pp. 258–261.

[20] Mrazek, V.; Sarwar, S. S.; Sekanina, L.; et al.: Design of Power-efficient Approximate Multipliers for Approximate Artificial Neural Networks. In *Proceedings of the 35th International Conference on Computer-Aided Design*. ICCAD '16. New York, NY, USA: ACM. 2016. ISBN 978-1-4503-4466-1. pp. 81:1–81:7.

[21] Mrazek, V.; Vasicek, Z.: Acceleration of transistor-level evolution using Xilinx Zynq Platform. In *2014 IEEE International Conference on Evolvable Systems*. Dec 2014. pp. 9–16.

[22] Mrazek, V.; Vasicek, Z.: Automatic Design of Low-Power VLSI Circuits: Accurate and Approximate Multipliers. In *2015 IEEE 13th International Conference on Embedded and Ubiquitous Computing*. Oct 2015. pp. 106–113.

[23] Mrazek, V.; Vasicek, Z.: Evolutionary Design of Transistor Level Digital Circuits Using Discrete Simulation. In *Genetic Programming*, edited by P. Machado; M. I. Heywood; J. McDermott; M. Castelli; P. García-Sánchez; P. Burelli; S. Risi; K. Sim. Cham: Springer International Publishing. 2015. ISBN 978-3-319-16501-1. pp. 66–77.

[24] Mrazek, V.; Vasicek, Z.: Automatic design of arbitrary-size approximate sorting networks with error guarantee. In *2016 26th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. Sept 2016. pp. 221–228.

[25] Mrazek, V.; Vasicek, Z.: Parallel Optimization of Transistor Level Circuits Using Cartesian Genetic Programming. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. GECCO '17. New York, NY, USA: ACM. 2017. ISBN 978-1-4503-4939-0. pp. 1849–1856.

[26] Mrazek, V.; Vasicek, Z.: Evolutionary Design of Large Approximate Adders Optimized for Various Error Criteria. In *GECCO Companion '18 Proceedings of the Companion Publication of the 2018 on Genetic and Evolutionary Computation Conference*. Association for Computing Machinery. 2018. ISBN 978-1-4503-5764-7. pp. 1–2.

[27] Mrazek, V.; Vasicek, Z.; Sekanina, L.: Evolutionary Approximation of Software for Embedded Systems: Median Function. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*. GECCO Companion '15. New York, NY, USA: ACM. 2015. ISBN 978-1-4503-3488-4. pp. 795–801.

[28] Najm, F. N.: A survey of power estimation techniques in VLSI circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. vol. 2, no. 4. Dec 1994: pp. 446–455. ISSN 1063-8210.

[29] Roy, K.; Prasad, S. C.: *Low-power CMOS VLSI circuit design*. New York: Wiley. 2000. ISBN 0-471-11488-x.

[30] Rusu, S.; Tam, S.; Muljono, H.; et al.: A 45 nm 8-Core Enterprise Xeon Processor. *IEEE Journal of Solid-State Circuits*. vol. 45, no. 1. Jan 2010: pp. 7–14. ISSN 0018-9200.

[31] Sekanina, L.: Introduction to Approximate Computing: Embedded Tutorial. In *19th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems*. Institute of Electrical and Electronics Engineers. 2016. ISBN 978-1-5090-2467-4. pp. 90–95.

[32] Soeken, M.; Große, D.; Chandrasekharan, A.; et al.: BDD minimization for approximate computing. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. Jan 2016. pp. 474–479.

[33] Srinivasan, G.; Wijesinghe, P.; Sarwar, S. S.; et al.: Significance driven hybrid 8T-6T SRAM for energy-efficient synaptic storage in artificial neural networks. In *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. March 2016. pp. 151–156.

[34] Vasicek, Z.: Cartesian GP in Optimization of Combinational Circuits with Hundreds of Inputs and Thousands of Gates. In *Genetic Programming*. Cham: Springer International Publishing. 2015. ISBN 978-3-319-16501-1. pp. 139–150.

[35] Vasicek, Z.: *New Methods for Synthesis and Approximation of Logic Circuits*. Habilitation thesis. Faculty of Information Technology, Brno University of Technology. 2017.

[36] Vasicek, Z.: Relaxed equivalence checking: a new challenge in logic synthesis. In *2017 IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*. April 2017. pp. 1–6.

[37] Vasicek, Z.; Mrazek, V.: Trading between quality and non-functional properties of median filter in embedded systems. *Genetic Programming and Evolvable Machines*. vol. 18, no. 1. Mar 2017: pp. 45–82. ISSN 1573-7632.

[38] Vasicek, Z.; Mrazek, V.; Sekanina, L.: Towards low power approximate DCT architecture for HEVC standard. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*. March 2017. pp. 1576–1581.

[39] Vasicek, Z.; Sekanina, L.: Evolutionary design of approximate multipliers under different error metrics. In *17th International Symposium on Design and Diagnostics of Electronic Circuits Systems*. April 2014. pp. 135–140.

[40] Vasicek, Z.; Sekanina, L.: How to evolve complex combinational circuits from scratch? In *2014 IEEE International Conference on Evolvable Systems*. Dec 2014. pp. 133–140.

[41] Vasicek, Z.; Sekanina, L.: Evolutionary Approach to Approximate Digital Circuits Design. *IEEE Transactions on Evolutionary Computation*. vol. 19, no. 3. June 2015: pp. 432–444. ISSN 1089-778X.

[42] Walker, J. A.; Hilder, J. A.; Tyrrell, A. M.: Evolving Variability-Tolerant CMOS Designs. In *Evolvable Systems: From Biology to Hardware*. Berlin, Heidelberg: Springer Berlin Heidelberg. 2008. ISBN 978-3-540-85857-7. pp. 308–319.

[43] Weste, N.; Harris, D.: *CMOS VLSI Design: A Circuits and Systems Perspective*. USA: Addison-Wesley Publishing Company. fourth edition. 2010. ISBN 0321547748, 9780321547743.

[44] Xu, Q.; Mytkowicz, T.; Kim, N. S.: Approximate Computing: A Survey. *IEEE Design Test*. vol. 33, no. 1. Feb 2016: pp. 8–22. ISSN 2168-2356.

Appendix A

Curriculum Vitae

Education

Ph.D. of Computer Science and Engineering 2014 — present
Faculty of Information Technology, Brno University of Technology
Supervisor: prof. Ing. Lukáš Sekanina, Ph.D.
Co-supervisor: doc. Ing. Zdeněk Vašíček, Ph.D.

Master of Computer and Embedded Systems 2012 — 2014
Faculty of Information Technology, Brno University of Technology
Acceleration of Transistor-Level Evolutionary Design of Digital Circuits Using Zynq
Supervisor: doc. Ing. Zdeněk Vašíček, Ph.D.
Passed with honour, acquired title Ing.

Bachelor of Information Technology 2009 — 2012
Faculty of Information Technology, Brno University of Technology
Intelligent Energy Measurement Device
Supervisor: doc. Ing. Zdeněk Vašíček, Ph.D.
Passed with honour, acquired title Bc.

Secondary school: Electronic Computer Systems 2005 — 2009
Střední průmyslová škola Jedovnice
Passed with honour

Conferences, Summer Schools, Institutions

2017

DATE (Laussane, CH), GECCO (Berlin, DE), ICCAD (Irvine, CA, US)
MACLOC — Deep learning on Chip summer school (Torino, IT)

2016

HiPEAC (Prague, CZ), SSCI (Athens, GR), ICCAD (Austin, TX, USA), Patmos (Bremen, DE)
Comprehensive Digital IC Implementation & Signoff course, MSC RAL STFC (Didcot, GB)

2015

EuroGP (Copenhagen, DK), EUC (Porto, PT)
Purdue University (West Lafayette, IN, US)

2014

SSCI (Orlando, FL, US)

Awards & Courses

- Comprehensive Digital IC Implementation & Signoff for Synopsys tools course
- The Best IP Award at the Design, Automation and Test in Europe (DATE) conference 2017.
- Prof. Ing. Jan Hlavička, DrSc. award for the excellent results in the PhD study at Computer Architectures & Diagnostics workshop for PhD students 2015.
- 1st prize in the Student EEICT 2014 competition awarded for the paper *Acceleration of Transistor-Level Evolutionary Design of Digital Circuits Using Zynq*.
- 3rd prize in the Student EEICT 2012 competition awarded for the paper *Intelligent energy measurement device*.

Projects

- FIT-S-17-3994 — *Advanced parallel and embedded computer systems, Brno Univesity of Technology*. Team member.
- FIT/FSI-J-17-4294 — *Enhancement of genetic optimization methods for computer engineering, Brno Univesity of Technology*. Investigator.
- GA16-08565S — *Advancing cryptanalytic methods through evolutionary computing, Czech Science Foundation*. Team member.
- LQ1602 — *IT4Innovations excellence in science, Ministry of Education, Youth and Sports of Czech Republic*. Team member.
- GA16-17538S — *Relaxed equivalence checking for approximate computing, Czech Science Foundation*. Team member.
- GA14-04197S — *Advanced Methods for Evolutionary Design of Complex Digital Circuits, Czech Science Foundation*. Team member.
- FIT-S-14-2297 — *Architecture of parallel and embedded computer systems, Brno Univesity of Technology*. Team member.

Teaching

- Introduction to Programming Systems — *labs, projects*
- Microprocessors and Embedded Systems — *labs, projects*
- VHDL Seminar — *lectures, projects*
- Bio-Inspired Computers — *projects*
- Introduction to Software Engineering — *projects*
- Supervised bachelor students
  - Jakub Hanzuš — Monitoring of Temperature for Small Buildings
  - Matěj Šoc — Advanced Measurement of Motorcycle Fuel Consumption
  - Filip Denk — Autopilot of RC Plane
- Supervised master students
  - Michal Tomášek — Neural Networks Classifier Design using Genetic Algorithm
  - Aleš Matěj — Deep Neural Networks: Embedded System Implementation

Work Experience

Junior Researcher 2013–present
Faculty of Information Technology, Brno University of Technology
Information systems developer 2008–2010
Vertigo.cz a.s., Brno, CZ
Embedded systems developer 2007–2017
Rawet s.r.o., Blansko, CZ
Information systems developer 2016–2018
Pasport.eu, Dolní Bojanovice, CZ