

Approximating Complex Arithmetic Circuits with Guaranteed Worst-Case Relative Error^{*}

Milan Češka jr., Milan Češka, Jiří Matyáš ,
Adam Pankuch, and Tomáš Vojnar

Faculty of Information Technology, Centre of Excellence IT4Innovations
Brno University of Technology, Brno, Czech Republic
`imatyas@fit.vutbr.cz`

Abstract. We present a novel method allowing one to approximate complex arithmetic circuits with formal guarantees on the *worst-case relative error*, abbreviated as WCRE. WCRE represents an important error metric relevant in many applications including, e.g., approximation of neural network HW architectures. The method integrates SAT-based error evaluation of approximate circuits into a verifiability-driven search algorithm based on Cartesian genetic programming. We implement the method in our framework ADAC that provides various techniques for automated design of arithmetic circuits. Our experimental evaluation shows that, in many cases, the method offers a superior scalability and allows us to construct, within a few hours, high-quality approximations (providing trade-offs between the WCRE and size) for circuits with up to 32-bit operands. As such, it significantly improves the capabilities of ADAC.

1 Introduction

In the recent years, reduction of power consumption of computer systems and mobile devices has become one of the biggest challenges in the computer industry. *Approximate computing* has been established as a new research field aiming at reducing system resource demands by relaxing the requirement that all computations are always performed correctly. Approximate computing can be conducted at different system levels with *arithmetic circuit approximation* being one of the most popular as such circuits are frequently used in numerous computations. Approximate circuits exploit the fact that many applications, including image and multimedia processing, machine learning, or neural networks, are *error resilient*, i.e., produce acceptable results even though the underlying computations are performed with a certain error. Chippa et al. [3] claims that almost 80 % of runtime is spent in procedures that could be approximated.

Circuit approximation can be formulated as an optimisation problem where the error and non-functional circuit parameters (such as power consumption or chip area) are conflicting design objectives. Designing complex approximate circuits is a time-demanding and error-prone process, and its automation is challenging too since the

^{*} This work has been partially supported by the Brno Ph.D. Scholarship Program, the Czech Science Foundation (project No. 19-24397S), the IT4Innovations Excellence in Science (project No. LQ1602), and the FIT BUT internal project FIT-S-17-4014.

design space is huge and evaluating candidate solutions is computationally demanding, especially if formal guarantees on the error have to be ensured. In our previous work [10], we proposed a scalable evolutionary circuit optimisation algorithm integrating a SAT-based circuit evaluation method that provides formal guarantees on the *worst-case absolute error* (WCAE).

In this paper, we extend the algorithm towards the *worst-case relative error* (WCRE) that represents another important error metric capturing the worst-case behaviour of the approximate circuits. Bounds on WCRE, in contrast to WCAE, require that the approximate circuits provide results that are close to the correct values even for small input values. This is essential for many application domains including, e.g., approximation of neural network hardware architectures [5]. Designing approximate circuits with WCRE bounds, however, represents a more challenging problem (when compared to WCAE) as the approximation has to preserve a larger part of the circuit logic and the circuit evaluation requires a more complicated procedure. To mitigate these challenges, we propose a novel construction of an auxiliary circuit (so-called miter) enabling an efficient SAT-based circuit evaluation against WCRE bounds. We integrate this evaluation procedure into the verifiability-driven circuit optimisation [10] implemented in our tool ADAC [1] and thus significantly extend the existing capabilities of automated techniques for the circuit approximation. Our experiments on circuits with up to 32-bit operands show that, in many cases, the proposed approach offers a superior scalability compared to alternative methods and allows us to construct, within a few hours, high-quality approximate circuits.

2 Search-Based Circuit Approximation

This section briefly summarises state-of-the-art methods for functional approximation with the focus on search-based approaches with formal error guarantees.

In *functional approximation*, the original system is replaced by a less complex one which exhibits some errors but reduces power consumption, delay, etc. Functional approximation can then be formulated as an optimisation problem where the error and energy efficiency/performance are conflicting design objectives. The approximation process either (1) tries to build an approximate solution from scratch or (2) tries to gradually modify the original system. The goal of the design process is to obtain an approximate solution with the best trade-off between the approximation error and resource savings.

Functional approximation can be performed manually by experts, but the current trend is to develop fully automated functional approximation methods that can be integrated into computer-aided design tools for digital circuits. There exist systematic approaches such as SALSA [11] or SASIMI [12], however, their drawback is an inability to generate novel logic structures. Search-based approximation techniques overcame this problem, and existing literature shows that this approach offers good performance and scalability [7].

Search-based approximation techniques typically iterate over two basic steps until a certain termination criterion is satisfied. The first step is the generation of candidate approximate solutions, and the error of these solutions is evaluated during the second

step. Eventually, the method produces a solution (or a set of solutions) providing a good trade-off between the approximation error and resource consumption.

Our search-based approach builds, in particular, on the *Cartesian genetic programming* [6]—a specialised version of evolutionary algorithms suitable for circuit approximation [8]. The circuits are represented as an oriented acyclic graph where nodes are located in a fixed-size two-dimensional matrix. New solutions are obtained from existing ones by simply changing the functionality and interconnections of the nodes.

To obtain a near-optimal solution, search-based techniques typically have to explore and evaluate a high number of candidate approximations [8]. Therefore, the efficiency of the methods used to evaluate the approximation error of the candidates is essential for the performance of the overall approach.

There exist several *error metrics* characterising different types of errors such as the worst-case error, the mean error, or the error rate. In this work, we primarily focus on the worst-case error that is essential when guarantees on the worst behaviour of the approximate circuits are required. For arithmetic circuits, the worst-case behaviour is typically captured either by the *worst-case absolute error* (WCAE) or by the *worst-case relative error* (WCRE), defined as follows.

For an original golden circuit G computing a function f_G and its approximation C computing a function f_C , we define:

$$\text{WCAE}(G, C) = \frac{\max_{x \in \{0,1\}^n} |\text{int}(f_G(x)) - \text{int}(f_C(x))|}{2^m} \quad (1)$$

$$\text{WCRE}(G, C) = \max_{n \in \mathbb{N}} \frac{|f_G(n) - f_C(n)|}{f_G(n)} \quad (2)$$

Figure 1 illustrates the difference between the approximation process targeting at WCAE and WCRE. This difference, in fact, motivates our work. It shows two sets of circuits approximating 8-bit multipliers optimised for WCRE (green squares) and WCAE (red circles), respectively. The plots show the trade-off between the circuit area (directly effecting the power consumption) and WCRE (left) and WCAE (right), respectively. First, we observe that circuits optimised for WCAE have very bad WCRE (red dots left) and vice versa (green squares right). Second, the plots demonstrate that when optimising 8-bit multipliers circuits for WCAE, we achieve about 50 % area reduction with WCAE = 1 % while we need to set WCRE = 40 % to obtain similar area improvements when optimising for WCRE. This is indeed caused by the fact that a larger part of the circuit logic has to be preserved to obtain approximations with low WCRE.

Methods evaluating the approximation error have a crucial impact on the performance of the approximation process. A popular class of methods employs *circuit simulation* on a set of inputs to evaluate the error. Such methods typically suffer from low scalability (when an exhaustive simulation is applied) or a lack of guarantees (when the circuits are simulated for a subset of the possible inputs only). In order to provide guarantees on the approximation error and scale to complex circuits at the same time, various *formal verification* techniques, such as model-checking, SAT solving, or BDDs have recently been integrated to the approximation process [4, 9]. They typically employ auxiliary circuits, so-called *miters*, that combine the original circuit and the approximate circuit and evaluate the error [2]. In our previous work [10], we proposed

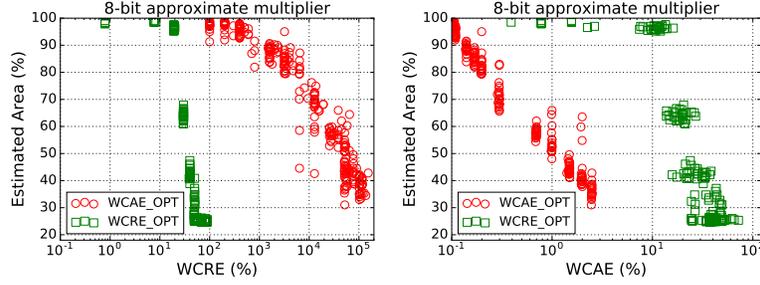


Fig. 1. A comparison of 8-bit multipliers approximated for WCRE and WCAE.

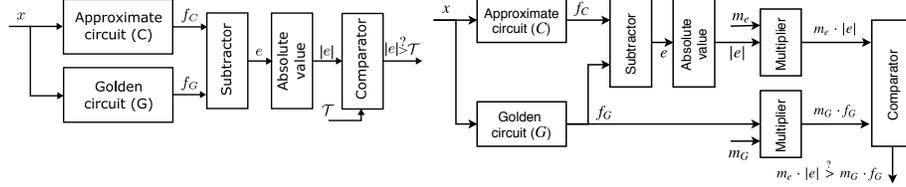


Fig. 2. The approximation miter used for WCAE evaluation (left) and the novel construction for WCRE evaluation (right).

a miter construction allowing one to subsequently use an efficient SAT-based procedure to check whether the approximate circuit satisfies a given WCAE bound. Moreover, we proposed a verifiability-driven search-strategy that drives the search towards promptly verifiable approximate circuits. The strategy introduces a limit on resources that the underlying SAT solver can use to prove that the WCAE bound is met. This approach currently provides the best performance for the circuit approximation with WCAE guarantees.

3 SAT-based WCRE Evaluation

To evaluate whether the given approximate circuit meets the required bound on WCRE, we adapt and extend the miter we designed for WCAE [10]. As shown in Figure 2 (left), the miter interconnects the golden circuit G and the candidate circuit C that both share identical inputs. The subtractor and absolute value blocks allow us to quantify the approximation error between C and G . Finally, the error is compared to a given threshold value T , and the output of the comparator is set to logical *true* if and only if the threshold T is violated. Thus the miter construction allows us to evaluate whether $WCAE(C, G) > T$ in a single SAT query. Note that, for a given approximation scenario, the threshold T is constant and can therefore be built into the structure of the comparator.

3.1 A Generic WCRE Miter

To obtain a WCRE miter, we extend the WCAE miter by adding some components. Recall that we need to check the satisfiability of the following formula:

$$\max_{n \in \mathbb{N}} \frac{|f_G(n) - f_C(n)|}{f_G(n)} > T. \quad (3)$$

Note that we do not need to find the maximum of the left-hand side of the formula, but rather determine if there exists a single input combination for which the bound T is violated. Therefore, we can replace Formula 3 by the following constraint

$$\exists n \in N : |f_G(n) - f_C(n)| * m_e > f_G(n) * m_G \quad (4)$$

where $T = m_G/m_e$. Based on this formula, we build a general WCRE miter using two multipliers by a constant and a generic comparator, see Figure 2 (right).

3.2 Variants of the WCRE Miter

Observe that the resulting WCRE miter is larger and more complex than the WCAE miter. This indeed slows down its evaluation and thus reduces the overall performance of the approximation process. To improve the performance and scalability with respect to the circuit complexity, we simplify the general WCRE miter and propose three variants of the miter that are smaller but can be used for certain values of the bound T only.

As we work with the binary representation of integers, multiplication by the powers of 2 is identical to a bit shift operation. Thus, each of the constants m_G and m_e can be expressed using two values: namely, mc_x denoting a multiplicative constant and bs_x denoting a number of shifted bits. The original values of m_G and m_e are then computed as:

$$m_G = mc_G * 2^{bs_G} \qquad m_e = mc_e * 2^{bs_e}$$

In combinational circuits, a shift by a constant number of bits is represented by a re-connection of wires only and does not contain any logical gates. This setting allows us to remove one or even both of the constant multiplications for a subset of target WCRE error bounds T . If we restrict the values of m_g and m_e to powers of two, we suffice with utilising bit shifts only. This restricts the obtainable values of T to $1/2^{bs_e}$, e.g., 50 %, 25 %, or 12.5 %. Adding one multiplier by a constant significantly broadens the range of supported target values. These can be expressed by one of the formulas: $2^{bs_G}/(mc_e * 2^{bs_e})$ or $(mc_G * 2^{bs_G})/2^{bs_e}$. However, the constants should be kept small. Using higher values leads to larger bit widths representing the compared numbers, and therefore a more complex comparator, thus negating the contribution of this optimisation.

4 Experimental Evaluation

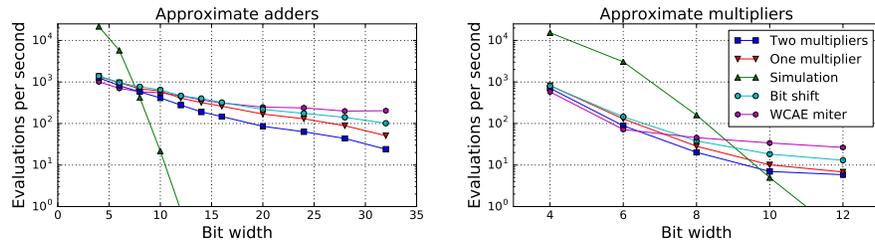
We have integrated the proposed WCRE miters into our tool ADAC [1] and evaluated its performance on a benchmark of circuit approximation problems.

4.1 Comparison of the WCRE Miters

In Table 1, we compare the size of the proposed WCRE miters. We select three target WCRE bounds T for the bit-shift variant and four target error values for one and two multiplier miter designs. The table shows the average sizes of the different variants of the miter obtained for the three chosen bit widths in adder and multiplier approximation. The size is measured in the number of nodes in the AIG graph representation of the miter. Note that AIG is a basic representation of circuits in ADAC and is directly used as the input for the SAT solving procedure. A larger size of AIGs negatively affects the

Table 1. Numbers of AIG nodes for different miters and WCAE bounds T .

T [%]	Bit shifts			One multiplier				Two multipliers			
	12.5	25.0	50.0	10.0	33.3	66.7	80.0	30.0	42.9	71.4	85.7
add8	226	228	233	324	327	342	360	447	510	506	519
add16	497	501	502	770	755	773	756	1079	1225	1181	1220
add32	1120	1090	1114	2074	2116	2084	2106	3125	3249	3315	3354
mult4	268	267	273	335	347	356	347	464	499	492	513
mult8	1175	1177	1183	1393	1421	1436	1414	1685	1833	1803	1841
mult12	2617	2621	2622	3032	3057	3060	3051	3512	3748	3726	3756

**Fig. 3.** Performance of the circuit evaluation using different WCRE miters, the WCAE miter, and simulation. Left: Adders. Right: Multipliers.

performance of the solver. We can see that the bit-shift variant is about a factor 2 smaller than the general construction using two multipliers. For multipliers, the differences in the size between the variants are less significant as the circuits themselves form a bigger part of the miter. Note that the average size of the WCAE miter for a 32-bit adder and 12-bit multiplier is 810 and 2437 AIG nodes, respectively, which is smaller than even the bit-shift variant of the WCRE miters for the corresponding circuits. This clearly indicates that the evaluation against WCRE is considerably harder.

Figure 3 illustrates how the size of the miters affects the performance of the candidate circuit evaluation. In particular, it shows the average number of evaluations per second (taken from 20 independent runs) when the approximation of adders (left) and multipliers (right) with different bit-widths (the x -axis) is performed. We also compare the miter-based methods with full simulation and WCAE miter evaluation.

We can observe that the simulation is considerably faster for small bit-widths, however, its performance significantly drops for circuits with operands larger than 10-bits. The proposed-SAT based approach scales much better. For the adders, it provides very good performance (around 100 evaluations per second) even for 32-bit operands. For the multipliers (representing structurally more complex circuits), the performance is much lower and drops to 10 evaluations per second for 12-bit operands. As expected, the speed of the miter evaluation slows down with increasing miter complexity—the bit-shift variant is the fastest while the version with two multipliers is the slowest. The difference in the evaluation speed is negligible for smaller circuits but becomes more significant for larger bit-widths. Note that the evaluation of the WCAE miters is significantly faster due their smaller sizes (e.g. 4-times smaller for the 32-bit adders and

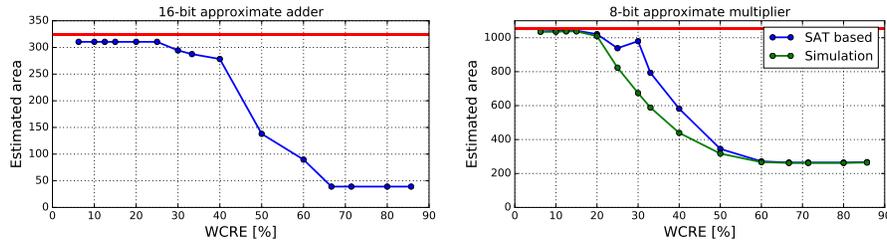


Fig. 4. The median circuit area of approximate adders (left) and multipliers (right). The red line indicates the area of the original circuit.

1.5-times smaller for the 12-bit multipliers in comparison to the two multiplier implementation).

For larger mitters, the bounds on the SAT solver resources get applied, and a small number of circuit evaluation tasks is skipped (e.g., for the WCRE mitters, 0.7 % for the 32-bit adders and 6 % for the 12-bit multipliers). The idea is to skip candidates for which the evaluation takes too long because their successors typically have the same problem and thus they reduce the performance of the overall approximation process. For more details, see our previous work [10], where we introduced this, so-called verifiability-driven strategy.

4.2 Circuit Approximation

In this section, we study how the proposed SAT-based circuit evaluation can be leveraged in circuit approximation. Recall that we integrate the evaluation procedure into the verifiability-driven circuit approximation based on Cartesian genetic programming. The optimisation is formulated as a single-objective optimisation, i.e., for a given threshold on the WCRE bound T , the approximation seeks for a circuit satisfying the bound and having the smallest circuit area¹. For every value of T , we run a 2-hours-long approximation process. To take into account the randomness of the evolutionary optimisation, we report the median of the circuit area obtained from 20 independent runs. Figure 4 illustrates the results of the approximation process, in particular, the obtained approximate circuits for the 16-bit adders (left) and the 8-bit multipliers (right). The circuits form a Pareto front that captures the trade-offs between the area and the approximation error. The red line shows the area of the golden circuit.

For the *adders*, the proposed approximation method works very well and is able to successfully approximate circuits up to 32-bit operands (not presented here). Figure 4 (left) shows that, for 16-bit adders, the most interesting solutions in the terms of accuracy and area savings are located in the interval between 30 % and 60 % WCRE. For smaller target error values, the reduction of the circuit size is negligible. On the other hand, the solutions with larger approximation errors do not feature further improvements. We can also observe a dramatic area reduction between 40 % and 50 % WCRE.

¹ We estimate the area as the sum of sizes of the gates (in the target 45 nm technology) used in the circuit. The estimation tends to be accurate and also adequately captures the circuit power consumption [9].

Approximation of the *multipliers* represents a significantly harder problem. Recall that the size of multipliers (and thus also of the miters) grows quadratically with respect to their bit-widths. Therefore, the design space is larger and candidate evaluation is more complicated as discussed in the previous section. Figure 4 (right) compares the approximate 8-bits multipliers obtained using the simulation-based and SAT-based evaluation procedure. The SAT-based approach slightly lags behind the simulation mainly in the interval between 25 % and 40 % WCRE. This can be explained by the worse performance of the SAT-based evaluation on the 8-bit multipliers (recall Figure 3 (right)).

As the performance of the simulation-based evaluation is very low beyond 10-bit multipliers, the approximation process is not able to provide a good approximation of these circuits within a 2-hours-long run. Although the SAT-based approach (namely the bit-shift solution) is able to evaluate around 10 candidates per second (for the 12-bit multipliers), the approximation process also fails to provide good Pareto sets. This is probably caused by the candidates that are skipped during the evaluation due to the resource limits on the underlying SAT solver. Note that this behaviour was not observed for the WCAE approximation that works very well even for 16-bit multipliers despite many skipped solutions [10]. This again indicates that the WCRE approximation is very challenging, and future research is necessary in this area.

References

1. Češka, M., Matyáš, J., et al.: ADAC: Automated design of approximate circuits. In: CAV'18. LNCS, Springer (2018)
2. Chandrasekharan, A., et al.: Precise error determination of approximated components in sequential circuits with model checking. In: DAC'16. IEEE (2016)
3. Chippa, V.K., et al.: Analysis and characterization of inherent application resilience for approximate computing. In: DAC'13. ACM (2013)
4. Ciesielski, M., et al.: Verification of gate-level arithmetic circuits by function extraction. In: DAC '15. ACM (2015)
5. Judd, P., Albericio, J., et al.: Proteus: Exploiting numerical precision variability in deep neural networks. ICS'16 pp. 1–12 (2016)
6. Miller, J.F., Thomson, P.: Cartesian genetic programming. In: Genetic Programming. Springer Berlin Heidelberg (2000)
7. Mrazek, V., et al.: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods. In: DATE'17 (2017)
8. Vasicek, Z., Sekanina, L.: Circuit approximation using single- and multi-objective Cartesian GP. In: EuroGP'15. pp. 217–229. Springer (2015)
9. Vasicek, Z., et al.: Towards Low Power Approximate DCT Architecture for HEVC standard. In: DATE'17. pp. 1576–1581. IEEE (2017)
10. Češka, M., et al.: Approximating complex arithmetic circuits with formal error guarantees: 32-bit multipliers accomplished. ICCAD'17 pp. 416–423 (2017)
11. Venkataramani, S., et al.: SALSA: systematic logic synthesis of approximate circuits. In: DAC'12. pp. 796–801. ACM (2012)
12. Venkataramani, S., et al.: Substitute-and-simplify: a unified design paradigm for approximate and quality configurable circuits. In: DATE'13. EDA (2013)