Runtime Accuracy-Configurable Approximate Hardware Synthesis Using Logic Gating and Relaxation

Tanfer Alan*, Andreas Gerstlauer†, Jörg Henkel*
*Karlsruhe Institute of Technology †University of Texas at Austin
alan@kit.edu gerstl@ece.utexas.edu henkel@kit.edu

***Abstract*— Approximate computing trades off computation accuracy against energy efficiency. Algorithms from several modern application domains such as decision making and computer vision are tolerant to approximations while still meeting their requirements. The extent of approximation tolerance, however, significantly varies with a change in input characteristics and applications.**

We propose a novel hybrid approach for the synthesis of runtime accuracy configurable hardware that minimizes energy consumption at area expense. To that end, first we explore instantiating multiple hardware blocks with different fixed approximation levels. These blocks can be selected dynamically and thus allow to configure the accuracy during runtime. They benefit from having fewer transistors and also synthesis relaxations in contrast to state-of-the-art gating mechanisms which only switch off a group of logic. Our hybrid approach combines instantiating such blocks with area-efficient gating mechanisms that reduce toggling activity, creating a fine-grained design-time knob on energy vs. area. Examining total energy savings for a Sobel Filter under different workloads and accuracy tolerances show that our method finds Pareto-optimal solutions providing up to 16% and 44% energy savings compared to state-of-the-art accuracy-configurable gating mechanism and an exact hardware block, respectively, at 2x area cost.

I. Introduction

Approximate computing leverages the application error resilience by relaxing exactness in computation towards a primary design goal: improving energy efficiency. Traditionally, a large class of research explored approximate computing at the hardware level targeting a single accuracy in manual [1, 2] and automated design [3–6] of functionally approximate circuits. The hardware is designed to have fewer transistors and shorter critical paths, where boolean functionality deviates from an exact specification to a limited extent. Instantiating such approximate hardware has a two-fold effect on energy: fewer transistors cause less toggling activity and shorter paths allow for voltage scaling or *synthesis relaxations*, i.e., circuits can be composed of smaller transistors that require less power at the same performance. The evident disadvantage is that the approximations on these circuits are fixed and hardwired. It is not possible to configure their accuracy at runtime.

Accuracy configurability is essential in practice for two main reasons: (i) Output quality of approximate hardware strongly depends on its inputs, and (ii) A workload may tolerate significantly different levels of approximation depending on its context and environment [7]. Runtime methods have shown that a fixed accuracy may be too conservative and accuracy configuration is necessary to maximally exploit the opportunities of approximate computing for energy efficiency improvement [7–9]. In particular, an offline profiler in [7] has shown that there is significant variation in precision requirements between different applications and also between different phases of an application.

Existing accuracy configurable hardware proposals [10–14] and design methodologies [15–17] primarily utilize *gating* mechanisms: They disable a configurable portion of the hardware by not propagating data or inserting control circuitry into the exact hardware with a small area overhead. Notwithstanding their potency, such approaches only benefit the from reduced toggling activity. Because they do not structurally simplify the circuit, e.g. shorten the critical path, they cannot exploit the full extent of power savings that static approximate hardware can achieve with synthesis relaxations.

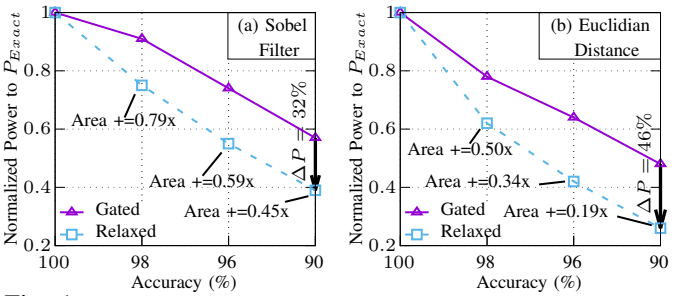

Fig. 1: Power vs. accuracy to compare gating an exact circuit against instantiating relaxed, approximate circuits. Area costs of instantiated circuits are noted. All values are relative to the exact version of the corresponding circuit. ΔP is dynamic power savings of instantiating approach over gating.

A potential solution can be instantiating multiple static accuracy circuits and switching between them despite their area and leakage power costs. In Figure 1, we compare gating an exact hardware, similar to existing work [15–17], against static accuracy approximate hardware by means of precision scaling their inputs, i.e., discarding a number of LSBs. Additional dynamic power savings reach up to 46% when leakage is neglected. Notably, the additional area cost of instantiating a circuit is reduced significantly as the accuracy is reduced. For instance in Figure 1b at 90% accuracy, the circuit requires only $0.19\times$ the area of the exact circuit. This example demonstrates that instantiating and connecting additional approximate circuits can be a lower-power and higher-area overhead alternative to gating. However, the dynamic power benefits of an instantiated logic is proportional to its utilization whereas its leakage power and area costs are fixed. At fine granularity, having many instantiations would reduce the utilization per circuit. Hence, instantiating may not always result in net energy benefits.

In this paper, we propose a novel hybrid approach for the synthesis of runtime accuracy-energy configurable hardware. Our approach combines gating mechanisms and instantiating multiple approximate circuits, to exploit both toggling activity and also synthesis relaxations. It enables fine-grain energy vs. area trade-offs in a design space that is a superset of two distinct approaches. Our work makes the following contributions:

- We propose a novel runtime accuracy configurable hardware design approach that combines joint gating and instantiating to exploit both synthesis relaxations and reduced toggling activity for total energy savings despite the leakage increase.

- Our work demonstrates the existence of a larger design space of accuracy-configurable hardware, with non-obvious trade-offs linked to the workload, hardware architecture, and technology.

In our evaluations, we examine a range of circuits under different workloads and accuracy requirements. Our experiments show at $2\times$ area cost and the same performance, up to 44% energy reduction compared to an exact hardware block and up to 16% energy reduction compared to state-of-the-art accuracy-configurable gated hardware while matching the accuracy.

II. Related Work

Approximate computing has received significant interest with research efforts spanning from software to architecture and circuits. The majority of the hardware research efforts explored targeting a single accuracy in manual [1, 2] and automated design [3–6] of functionally approximate circuits. Runtime monitoring techniques, however, have shown that with temporal variations in input characteristics, the optimal accuracy to meet an application quality target also changes [8, 9]; the single accuracy circuit delivers suboptimal benefits or violates the quality targets [7].

To utilize the temporal variations with approximations, accuracy configurable hardware [10–14] and generic design methodologies are proposed [15–17]. The *de facto* method for this purpose is precision scaling, i.e., not propagating the LSBs of data. Energy-aware precision scaling of floating-point data is proposed in [13]. At the system level, precision scaling is applied in memory controller [14]. A vector coprocessor with data precision reducing FIFO input buffers is proposed in [10]. It is extended with an internal PID controller [11] and later with accuracy aware ISA extensions [12]. All of these designs apply precision scaling on data, not on the circuit. Hence, they only benefit from reduced toggling activity, and not from synthesis relaxations.

Generic design methodologies can offer improving performance by synthesizing partially faster circuits [18–20] or energy efficiency by means of disabling low significance logic groups in hardware [15–17]. Voltage scaling may possibly reduce the energy in [18–20], but it comes at often ignored system level costs of multiple additional voltage supplies, rails and switches. Two gating mechanisms are utilized in [15]: (1) Masking logic groups by inserting control gates to their combinatorial path and (2) power gating the logic groups to partially switch off the hardware. To group the gated logic, a genetic programming based search is proposed in [16]. In [17] clock gating for approximations, *clock overgating*, is proposed. By disabling the clock signal of flip-flops, power savings are achieved in their fan-out cone. Similar to existing accuracy configurable hardware designs, gating mechanisms reduce the energy via reducing the toggling activity only.

Our work distinguishes from existing methodologies as it benefits from both gating and synthesis relaxations. It incorporates the existing design methodologies for accuracy configuration and existing single accuracy hardware designs towards creating a design space, that is a superset of these individual approaches. Hence, it extends the design space of gating approaches for further energy savings at area expense.

III. Accuracy Configurable Hardware Architecture

Dynamic accuracy configuration, as we interpret it in the scope of this paper, aims to maximally exploit energy efficiency benefits of approximate hardware while meeting a given quality target given at runtime. In this section, we first compare *gating* mechanisms and position *instantiating* approximate circuits as a distinct design approach for accuracy configurable hardware architecture. Next, we introduce a *hybrid* design approach that enables a design space with fine-grain energy vs. area trade-offs. Finally, we explain the hardware execution of cycle-by-cycle accuracy configuration, facilitating dynamic runtime adjustments.

A. Gating Groups of Logic in Hardware

We utilize the gating mechanisms discussed in Section II as low area cost accuracy configuration methods [15–17]. When compared, masking by inserting control gates into the combinatorial paths, also increases path delays of the circuit. Thus, either the circuit delay increases or circuit area and power increase to match the same delay with more aggressive synthesis optimizations. These effects are counter-intuitive to our energy optimization design goal. Power and area overheads are reported as up to 7.6% and 8.7% [15]. Power gating mechanism used in [15, 16] requires many cycles, prohibiting cycle-by-cycle dynamic adjustments. In comparison, clock gating for approximations can eliminate such delay overheads and allow dynamic adjustments.

Clock gating is a mean for disabling configurable partitions of a circuit. In [17], *significance constrained overgating* strategy together with clock gating candidates algorithm result in configurable degrees of precision scaling on the input registers. While our design approach will allow for any gating mechanism to be employed, we use clock gating as a baseline in our comparisons to represent the gating approach in the remainder of the paper.

B. Instantiating Approximate Circuits with Different Accuracies

We employ adding and connecting multiple instantiations of a circuit for additional energy savings at area expense. As shown in Figure 1, the approximate instantiations have static accuracies and they exhibit lower power at the same delay as the exact. To design approximate instantiations of a circuit, we inherit the rich set of existing static approximation, i.e., single accuracy hardware design methods. By connecting separate instantiations of a circuit with different accuracies, the hardware is able to offer dynamic accuracy configuration to fulfill the varying requirements of the application at runtime. existing methods on static approximation and transforming them to design accuracy configurable hardware.

Energy savings through functional simplification of the hardware architecture originate from having fewer gates and shorter paths compared to the exact circuit. When the exact and the approximate circuits are synthesized for the same clock delay, the shorter paths allow *synthesis relaxations*: Boolean remapping and undoing gate-level delay optimizations [21]. Boolean remapping converts a large number of parallel gates into a less number of serial gates while maintaining the boolean function (e.g., parallel-prefix adder to ripple carry adder). Undoing gate-level delay optimizations such as inserting buffers (load isolation) and splitting the driving gates on the critical paths (load splitting) reduce the number of gates in the design. Gate resizing re-composes the circuit with smaller transistors that require less energy. Hence, synthesis relaxations generate inherently more energy-efficient circuits.

We connect the new instantiations by splitting the primary inputs to separate input registers and enabling propagation of primary inputs to only one instantiation at a time while clock gating the others. To route the outputs of new instantiations, we propose to extend existing multiplexers. Extending a multiplexer increases its complexity logarithmically. Also, unless all possible inputs are utilized, adding different accuracy circuits would

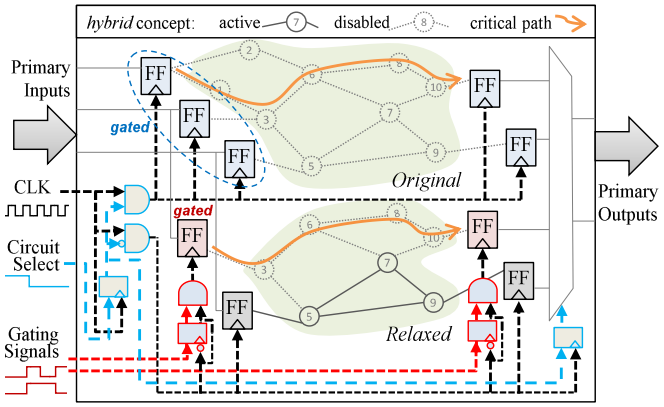

Fig. 2: *hybrid* approach for designing accuracy configurable hardware architecture. The lower positioned circuit is instantiated and gated.

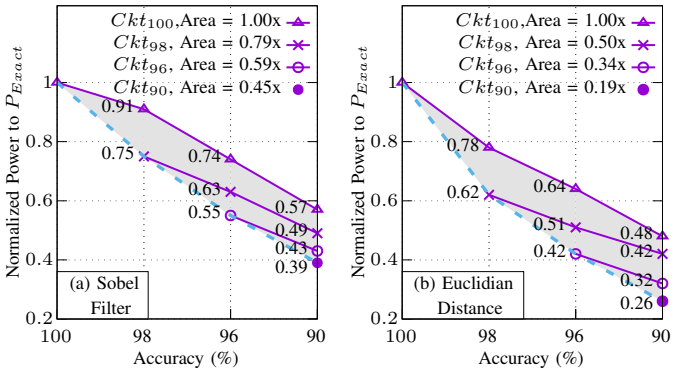

Fig. 3: Power vs. accuracy design space of the hybrid approach. Power values are labeled. Area costs of instantiated circuits are given in the legend. All values are relative to the exact version of the corresponding circuit.

not change the depth of the multiplexer. Therefore we assume the control overhead in terms of extending the multiplexers is negligible. Connecting the new instantiations is a hardware system-specific question. For instance, if the hardware is located in the execution stage of a processor, new instantiations can be connected as new function units. Or, if the hardware is a peripheral, new instantiations can be connected as other peripherals. Therefore we refrain from binding it to a single solution. As a guideline, in case extending a multiplexer is not possible, adding a new multiplexer to the same hardware stage should be avoided for delay reasons.

The instantiated circuits are at the level of one complete pipeline stage. We call this granularity a *hardware block*. Potentially, instantiating at the sub-block level can exploit logic sharing opportunities for area savings across instantiated blocks. This can be done in an automated manner: we leverage the final synthesis tool to find hardware common subexpressions between instantiated combinatorial blocks for sharing logic.

C. Hybrid Design Approach: Accuracy Configurable Hardware Architecture with Jointly Instantiating and Gating

While gating brings some energy benefits at low area overhead, instantiating can significantly improve the energy benefits with a higher area overhead. Our proposed *hybrid* approach combines the two: It instantiates distinct approximated blocks at coarse accuracy levels and selectively gates them. Consequently, it enables fine-grained intermediate accuracies and additional energy savings on their computation, without the area and leakage cost of instantiating each intermediate accuracy circuit. In Figure 2, we illustrate the proposed hybrid approach. Here, an approximate instantiation of the original circuit is placed below it. The instantiation has a shorter critical path and fewer gates. It maximizes the power savings of computation at a particular accuracy. The instantiation is also gated to enable further power savings for a further range of accuracies.

In Figure 3, we extend our motivational example with the hybrid design approach. Starting from exact versions of the Sobel filter and Euclidian distance circuits, we synthesize approximate circuits for each accuracy. Afterward, we gate each synthesized circuit for lower accuracies. The leftmost value of each line represents power values achieved with instantiations. The lines towards the right represent power values achieved with gating each instantiated circuit. Note that the design space of prior, gating-only approaches is limited to the Ckt_{100} line. We show the design space of our proposed *hybrid* approach by the shaded area.

Given an area budget, we can instantiate a set of circuits to address varying accuracy requirements. Energy-optimal selection of such a circuit set is not a trivial task. It necessitates answering the following questions: (1) which accuracy circuits to instantiate (2) which instantiated circuits to gate and (3) how to associate different accuracy requirements of workloads with the hardware in an optimum manner. The energy optimal solution is a function of hardware and workload. Within an area budget, the additional area and leakage cost vs. dynamic power savings of instantiations over gating should be considered. From the workload perspective, the impact of power savings through instantiation is weighted by the utilization of the circuit. Note that the number of solutions increase quadratically with the number of accuracies as we can see in Figure 3. Next we use the Sobel Filter as a case study in our experiments.

IV. Experiments and Results

Profiling: The required circuit accuracies and their utilizations are application and input dependent. To abstract their effect on our methodology, we used 4 different accuracies and 3 different utilization distributions as shown in Table I. We considered circuit accuracies in terms of $1 - MRED$ (Mean Relative Error Distance), as shown in Equation (1), where O_{approx_n} is the nth approximate and O_{exact_n} is the nth exact output value.

$$Accuracy = 1 - \frac{1}{N}\sum_{n=1}^{N}\frac{\left|(O_{approx\_n} - O_{exact\_n})\right|}{O_{exact\_n}} \quad (1)$$

Methodology: For the synthesis of the circuits used in our experiments, we use Synopsys Design Compiler with the ultra high effort option using TSMC 65nm generic plus technology library. We synthesized a circuit for each given accuracy. All instantiations of circuits are synthesized to match the same delay, i.e.,110% of the minimum delay of the corresponding exact circuit ($delay_{synth} = delay_{min} \times 110\%$). Therefore all circuits compared in our experiments have the same performance. We instantiated each circuit directly in the HDL testbench to discard the multiplexing delay as discussed in Section III-B. For gating, we reduced the number of primary inputs supplied to the netlists according to the algorithm given in the state of the art [17]. For the area cost of gating, we assumed a conservative 3% penalty per added accuracy, which is in line with [15]. In characterizing the power consumption of our circuits, we generated toggling activity files from gate-level simulations with ModelSim HDL simulator and provided them to Synopsys Primetime. We used a 512x512 pixel cameraman image as input to the Sobel Filter.

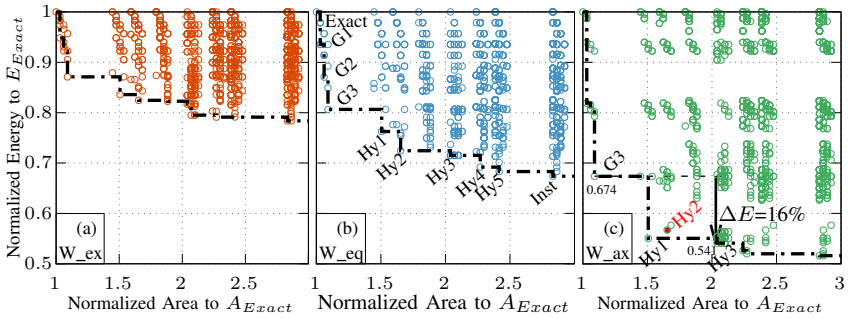

Fig. 4: Design space of an accuracy configurable Sobel Filter under 3 different utilizations given in Table I. Pareto solutions in Figure 4b are labeled to be used in Figure 5.

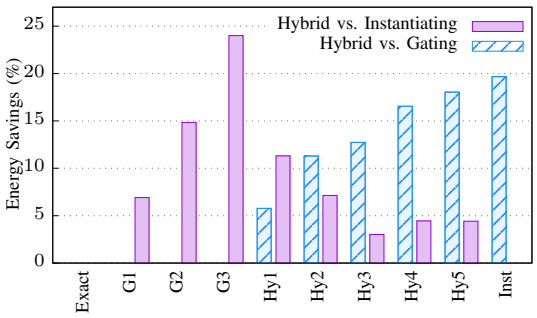

Fig. 5: Pareto front comparison of hybrid against gating and instantiating solutions from Figure 4b.

TABLE I: Utilization distributions (U_i) of 4 different circuit accuracies for 3 synthetic workloads

Utilization Distributions	Accuracy			
Workload	100%	98%	96%	90%
W_eq - *even distribution*	0.25	0.25	0.25	0.25
W_ex - *mostly exact*	0.5	0.2	0.2	0.1
W_ax - *mostly approximate*	0.1	0.15	0.05	0.7

In our experiments on the Sobel Filter, 98%, 96% and 90% accuracies corresponded to PSNR 45, 38 and 31dB, respectively.

A. Design space exploration

We begin our evaluations by presenting the design space of a Sobel Filter in Figure 4 under 3 different workloads. Each solution on the design space corresponds to a particular combination of instantiated circuit, gating and workload accuracy association. Energy values include both dynamic and leakage energy, normalized to the exact circuit. We show the Pareto front with a dashed line. When examined at 2× maximum area constraint, the Pareto-optimum solution in Figure 4b is labeled *Hy2*. At the same area cost, under *W_ax* workload in Figure 4c, *Hy1* is the optimum solution, which dominates *Hy2*. *Thus, Pareto-optimal solutions are workload dependent.*

The dynamic power impact of instantiating an approximate circuit is proportional to the utilization of its accuracy. At the excess area, only low impact circuits remain. As an example, under the mostly approximate workload in Figure 4c, 90% is the dominating accuracy. With solution *Hy1*: [$Ckt\_acc_{100}g_{98}g_{96}$, Ckt_acc_{90}] (i.e., instantiating exact and 90% accuracy circuits, gating the exact for 98% and 96%) we already achieve significant energy savings. Additionally instantiating a 96% accuracy circuit (Ckt_acc_{96}) only reduces the energy consumption by a mere 2.6% at $0.6\times A_{\text{Exact}}$ extra area cost. Similarly, under *W_eq*, where the accuracy utilization is even, an extra 0.65x area at first reduces energy by 28% w.r.t. *Exact* (10% w.r.t. *G3*), and at the end, only 1.3%, w.r.t. *Hy5*. *Thus, energy savings diminish at excess area.*

B. Comparison of Pareto-optimal solutions

In Figure 5, we show the energy savings of hybrid solutions over gating and instantiating for the *W_eq* workload. The figure highlights that, as we increase the area budget, the Pareto solutions of *hybrid* and *instantiating* approaches offer energy reduction over the-state-of-the-art gating approach. *The joint design space offers solutions that are superior or equal to the two individual approaches.*

V. Conclusion

We addressed the necessity of mixed-accuracy hardware systems with the exploration of applying gating mechanisms to existing circuits together with instantiating more efficient circuits. Jointly, they present a larger design space where non-trivial decisions are necessary to find optimal solutions. Our work has demonstrated that dynamic accuracy configurable hardware with considerably (up to 16%) reduced energy compared to existing gating solutions can be synthesized when more circuit area can be utilized.

Acknowledgement

This work was supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre Invasive Computing (SFB/TR 89).

References

[1] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in *DATE*, 2008.
[2] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *ISIC*, 2009.
[3] J. Miao, A. Gerstlauer, and M. Orshansky, "Approximate logic synthesis under general error magnitude and frequency constraints," in *ICCAD*, 2013.
[4] S. Lee and A. Gerstlauer, "Fine grain precision scaling for datapath approximations in digital signal processing systems," in *IFIP/IEEE International Conference on Very Large Scale Integration-System on a Chip*, 2013.
[5] J. Castro-Godínez, S. Esser, M. Shafique, S. Pagani, and J. Henkel, "Compiler-driven error analysis for designing approximate accelerators," in *DATE*, 2018.
[6] I. Scarabottolo, G. Ansaloni, and L. Pozzi, "Circuit carving: A methodology for the design of approximate hardware," in *DATE*, 2018.
[7] S. Yesil, I. Akturk, and U. R. Karpuzcu, "Toward dynamic precision scaling," *IEEE Micro*, vol. 38, no. 4, pp. 30–39, 2018.
[8] W. Baek and T. Chilimbi, "Green: A system for supporting energy-conscious programming using principled approximation," *PLDI, ACM*, 2010.
[9] M. A. Laurenzano, P. Hill, M. Samadi, S. Mahlke, J. Mars, and L. Tang, "Input responsiveness: using canary inputs to dynamically steer approximation," *PLDI, ACM*, 2016.
[10] V. K. Chippa, D. Mohapatra, K. Roy, S. T. Chakradhar, and A. Raghunathan, "Scalable effort hardware design," *TVLSI*, 2014.
[11] V. Chippa, A. Raghunathan, K. Roy, and S. Chakradhar, "Dynamic effort scaling: Managing the quality-efficiency tradeoff," in *DAC*, 2011.
[12] S. Venkataramani, V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Quality programmable vector processors for approximate computing," in *MICRO*, 2013.
[13] C.-C. Hsiao, S.-L. Chu, and C.-Y. Chen, "Energy-aware hybrid precision selection framework for mobile gpus," *Computers & Graphics*, 2013.
[14] A. Jain, P. Hill, S.-C. Lin, M. Khan, M. E. Haque, M. A. Laurenzano, S. Mahlke, L. Tang, and J. Mars, "Concise loads and stores: The case for an asymmetric compute-memory architecture for approximation," in *MICRO*, 2016.
[15] S. Jain, S. Venkataramani, and A. Raghunathan, "Approximation through logic isolation for the design of quality configurable circuits," in *DATE*, 2016.
[16] V. Mrazek, Z. Vasicek, and L. Sekanina, "Design of quality-configurable approximate multipliers suitable for dynamic environment," in *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2018.
[17] Y. Kim, S. Venkataramani, K. Roy, and A. Raghunathan, "Designing approximate circuits using clock overgating," in *DAC*, 2016.
[18] T. Alan and J. Henkel, "Slackhammer: Logic synthesis for graceful errors under frequency scaling," *TCAD*, 2018.
[19] D. J. Pagliari and M. Poncino, "Application-driven synthesis of energy-efficient reconfigurable-precision operators," in *ISCAS*, 2018.
[20] B. Moons and M. Verhelst, "Dvas: Dynamic voltage accuracy scaling for increased energy-efficiency in approximate computing," in *ISLPED*, 2015.
[21] *Design Compiler® User Guide*, Synopsys, www.synopsys.com, 2010.
[22] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," in *DAC*, 2013.