

# ARCSyn: Aging-Aware Accuracy-Reconfigurable Logic Synthesis

Ruicheng Dai<sup>1</sup>, Feiyang Shu<sup>2</sup>, Pengpeng Ren<sup>2</sup>, Runsheng Wang<sup>3</sup>, and Weikang Qian<sup>1</sup>

<sup>1</sup>Global College and <sup>2</sup>School of Integrated Circuits, Shanghai Jiao Tong University, China

<sup>3</sup>School of Integrated Circuits, Peking University, China

Emails: {ruichengdai, shufy, pengpengren}@sjtu.edu.cn, r.wang@pku.edu.cn, qianwk@sjtu.edu.cn

**Abstract**—As CMOS technology scales down, transistor aging has become a major threat to the long-term reliability of digital circuits. Existing solutions, such as aging-aware synthesis and approximate computing, suffer from either limited optimization space or early-stage accuracy loss. To address the above limitations, we propose ARCSyn, an aging-aware logic synthesis framework that generates accuracy-reconfigurable circuits capable of switching between accurate and approximate modes depending on aging conditions. Experimental results show that ARCSyn effectively extends circuit lifetime by 9.5 times while satisfying user-specified error constraints with only 3.72% area overhead.

## I. INTRODUCTION

As devices continue to scale down, reliability has become a critical challenge in advanced CMOS technologies. Transistor aging induces gradual shifts in the threshold voltage ( $V_{th}$ ), leading to increase of gate delays over time and ultimately causing timing violations [1]–[4]. To address this issue, conventional guardbanding [5], [6] ensures correctness but sacrifices early-stage performance or increases power consumption. Recent studies have proposed aging-aware design methods to reduce the pessimism of static guardbands [1], [2]. However, these methods provide only limited post-aging delay improvement and are insufficient to address severe aging. Since many modern applications are inherently error-tolerant [7]–[10], another line of work leverages *approximate computing* to enable more aggressive delay reduction [11]–[13]. For example, the work [12] employs *approximate logic synthesis (ALS)* [14] to generate lower-delay approximate circuits to improve aging resilience, but it sacrifices accuracy from the beginning.

To address these limitations, we propose ARCSyn, an aging-aware accuracy-reconfigurable synthesis method that generates accuracy-reconfigurable circuits capable of switching between accurate and approximate modes, thereby avoiding early-stage accuracy loss and improving long-term aging resilience.

## II. METHODOLOGY

ARCSyn is designed as an iterative synthesis flow, as shown in Fig. 1(a). It achieves accuracy reconfiguration by inserting multiple *reconfigurable modules (RMs)*, as shown in Fig. 1(b), each applied to a sub-circuit and providing accurate and approximate outputs selected by a mode select signal. To identify an optimal set of sub-circuits for modification, we proposed an iterative flow that generates multiple *approximate candidates* using an ALS method. The ALS process takes an original circuit and a user-defined error constraint  $\xi$  as inputs.

This work is supported by the National Natural Science Foundation of China under Grants T2293700, T2293701, and T2293704 and Natural Science Foundation of Shanghai under Grant 25ZR1401189. Corresponding authors: Runsheng Wang and Weikang Qian.

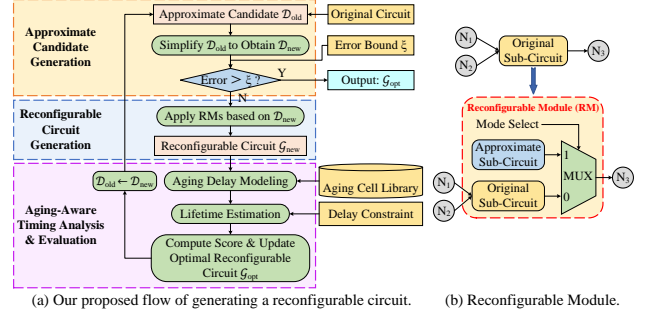


Fig. 1. Overview of ARCSyn.

In each iteration, an approximate candidate  $\mathcal{D}_{new}$  is generated by incrementally replacing a selected sub-circuit in the approximate candidate  $\mathcal{D}_{old}$  from the last iteration. The approximate candidate generation continues until the error constraint is exceeded. For each  $\mathcal{D}_{new}$ , we build an accuracy-reconfigurable circuit  $\mathcal{G}_{new}$  by selectively replacing some sub-circuits with the corresponding RMs. Then we conduct aging-aware timing analysis to estimate the circuit lifetime under a given aging cell library and a delay constraint. We evaluate each  $\mathcal{G}_{new}$  by a performance score considering both lifetime and area. The flow finally outputs the optimal accuracy-reconfigurable circuit  $\mathcal{G}_{opt}$  with the highest score. In the following subsections, we discuss some technical details.

### A. Approximate Candidates Generation

We leverage an iterative ALS flow [15] to generate multiple approximate candidates, which treats each node as a sub-circuit to maximize design flexibility. In each iteration, a constant *local approximate change (LAC)*, which replaces a node by a constant 0 or 1, is applied to the current circuit. As a result, the area of the approximate sub-circuit shown in Fig. 1(b) is zero, introducing the least hardware cost. Moreover, the *arrival time* of the replaced node is reduced to zero, making it one of the most promising approaches to reduce the overall delay. An example of an approximate candidate with three constant LACs applied is shown in Figs. 2(a) and (b). As the circuit becomes progressively approximated across iterations, we obtain a sequence of candidates exhibiting different delay-error trade-offs.

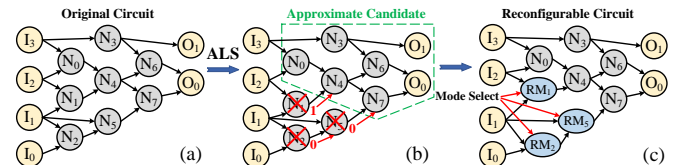


Fig. 2. Approximate circuit and reconfigurable circuit generation: (a) original circuit; (b) approximate candidate; (c) reconfigurable circuit.

## B. Low-Overhead Reconfigurable Circuit Generation

For each approximate candidate, we obtain a set of replaced nodes, denoted as  $N_{rep}$ . A direct way to enable reconfiguration is to insert RMs using a MUX-based structure in Fig. 1(b) at all nodes in  $N_{rep}$ , as shown in Fig. 2(c). Since each RM inevitably incurs area overhead, we propose the following two techniques to reduce the overhead.

1) *Minimizing the hardware cost of each RM*: A constant LAC makes one data input of the RM a constant, which leads to constant propagation opportunities to simplify the RM. Specifically, we consider two MUX-based structures, as shown in Figs. 3(a) and (b), and apply Boolean matching to map each of them to an existing gate in the cell library. If Boolean matching fails, we use the backup solutions in Fig. 3(c).

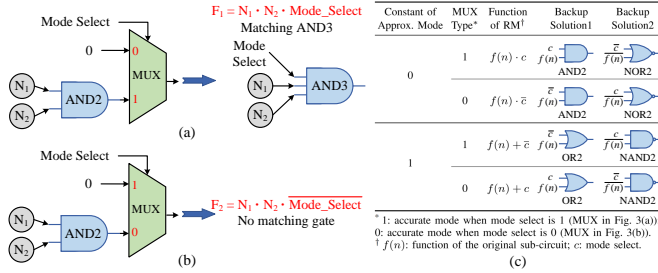


Fig. 3. Two examples of RM optimization, where the RM chooses the accurate mode when the mode select signal is 1 (a) and 0 (b). (c) Target Boolean function and backup solutions to implement an RM.

2) *Reducing the number of nodes replaced by RM*: While RMs can reduce the delay of certain paths, not all replacements at nodes in  $N_{rep}$  contribute to reducing the *critical path delay*. These cases fall into three groups: (i) the RM is not an input of the approximate candidate (e.g.,  $RM_2$  that replaces  $N_2$  in Fig. 2(b)), and therefore does not affect its delay or logic; (ii) the RM lies on paths that are already short; and (iii) the RM is initially on the critical path but becomes non-critical after other RMs are inserted. We first identify the nodes in groups (i) and (ii) within  $N_{rep}$ , and then propose a heuristic that applies RMs in reverse topological order to identify as many nodes as possible in group (iii). Then we remove all such nodes from  $N_{rep}$  to reduce area overhead.

## C. Aging-Aware Timing Analysis for Reconfigurable Circuits

In practical implementations, an aging sensor monitors whether the circuit is approaching a timing violation and triggers the switch from the accurate mode to the approximate mode [16]. However, as real aging behavior is unpredictable at design time, we develop a model to estimate the lifetimes of both accurate and approximate modes (denoted as  $LT_{acc}$  and  $LT_{app}$ ) under ideal conditions to guide design-time evaluation.

We use Cadence Liberate [17] to generate an aging cell library characterized at operation time  $L$ , and we consider the impact of input *signal probabilities* (SPs) on aging by generating multiple aged versions of each standard cell, which enables interpolation to obtain the aged delay for any input SP combination. To extend this to arbitrary operation times, we use a widely adopted degradation model [18]–[20] to derive the delay-versus-time aging curve of each gate under a

specific input SP combination. To support lifetime estimation under mode switching, we follow the accurate-mode aging curve until the switching point, match its aging state on the approximate-mode curve, and continue aging from there to obtain a piecewise aging curve under mode switching.

## III. EXPERIMENTAL RESULTS

This section shows the experimental results. ASAP7, a 7-nm predictive FinFET PDK [21], is used to construct a 10-year aging library for timing analysis. We design accuracy-reconfigurable circuits for some benchmarks and compare them with three baselines: the original circuits and two versions of reconfigurable circuits generated from the approximate circuits in [12], where the RMs are implemented with a MUX-based structure and with our proposed method (*-Opt*), respectively. The delay constraint is set to the 5-year aged delay of the original circuit, and each circuit is assigned a proper error constraint (see Fig. 4) based on its characteristics.

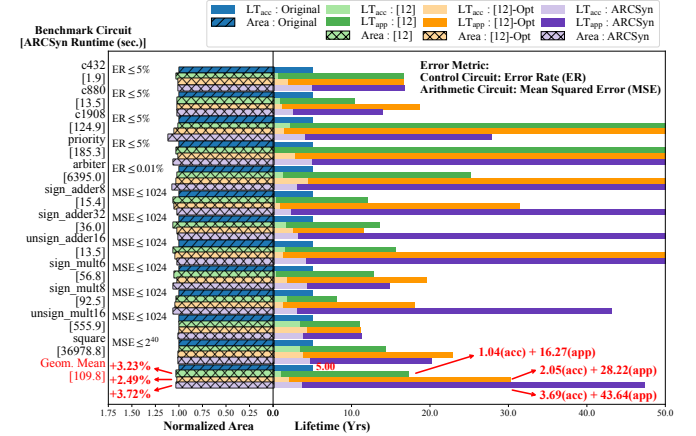


Fig. 4. Comparison among original circuits and reconfigurable circuits generated by ARCSyn and from approximate circuits produced using [12].

Fig. 4 shows the area and lifetime of the circuits under different methods and the runtime of ARCSyn. The result shows that when converting the approximate circuits from [12] into reconfigurable ones, our optimization technique (*-Opt*) reduces the area overhead while simultaneously decreasing the delays in both accurate and approximate modes, resulting in longer  $LT_{acc}$  (2 $\times$ ) and  $LT_{app}$  (1.7 $\times$ ). Moreover, with a comparable circuit area, ARCSyn achieves better delay optimization in both modes, further extending the lifetimes. On average, the circuits produced by ARCSyn maintain an  $LT_{acc}$  of 3.69 years and an  $LT_{app}$  of 43.64 years, achieving a 9.5 $\times$  increase in total circuit lifetime, while introducing only 3.72% area overhead. Moreover, ARCSyn is efficient, requiring only 109.8 seconds on average to generate an accuracy-reconfigurable circuit.

## IV. CONCLUSION

In this work, we proposed ARCSyn, an aging-aware synthesis flow for generating accuracy-reconfigurable circuits that support switching between an accurate mode and a low-delay approximate mode. The experiments show that ARCSyn can significantly extend circuit lifetime with a small area overhead.

## REFERENCES

- [1] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "NBTI-aware synthesis of digital circuits," in *Design Automation Conference*, 2007, pp. 370–375.
- [2] M. Ebrahimi, F. Oboril, S. Kiamehr, and M. B. Tahoori, "Aging-aware logic synthesis," in *International Conference on Computer-Aided Design*, 2013, pp. 61–68.
- [3] H. Amrouch, B. Khaleghi, A. Gerstlauer, and J. Henkel, "Reliability-aware design to suppress aging," in *Design Automation Conference*, 2016, pp. 1–6.
- [4] S. Guo, R. Wang, Z. Yu, P. Hao, P. Ren, Y. Wang, S. Liao, C. Huang, T. Guo, A. Chen *et al.*, "Towards reliability-aware circuit design in nanoscale FinFET technology:—New-generation aging model and circuit reliability simulator," in *International Conference on Computer-Aided Design*, 2017, pp. 780–785.
- [5] K. Bowman, J. Tschanz, C. Wilkerson, S.-L. Lu, T. Karnik, V. De, and S. Borkar, "Circuit techniques for dynamic variation tolerance," in *Design Automation Conference*, 2009, pp. 4–7.
- [6] E. Gunadi, A. A. Sinkar, N. S. Kim, and M. H. Lipasti, "Combating aging with the colt duty cycle equalizer," in *International Symposium on Microarchitecture*, 2010, pp. 103–114.
- [7] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *European Test Symposium*, 2013, pp. 1–6.
- [8] Q. Xu, T. Mytkowicz, and N. S. Kim, "Approximate computing: A survey," *IEEE Design & Test*, vol. 33, no. 1, pp. 8–22, 2015.
- [9] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of approximate radix-4 Booth multipliers for error-tolerant computing," *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1435–1441, 2017.
- [10] A. Agrawal, J. Choi, K. Gopalakrishnan, S. Gupta, R. Nair, J. Oh, D. A. Prener, S. Shukla, V. Srinivasan, and Z. Sura, "Approximate computing: Challenges and opportunities," in *International Conference on Rebooting Computing*, 2016, pp. 1–8.
- [11] H. Amrouch, B. Khaleghi, A. Gerstlauer, and J. Henkel, "Towards aging-induced approximations," in *Design Automation Conference*, 2017, pp. 1–6.
- [12] Z. Zhang, R. Wang, Z. Zhang, R. Huang, C. Meng, W. Qian, and Z. Zhou, "Reliability-enhanced circuit design flow based on approximate logic synthesis," in *Great Lakes Symposium on VLSI*, 2020, pp. 71–76.
- [13] K. Balaskas, G. Zervakis, H. Amrouch, J. Henkel, and K. Siozios, "Automated design approximation to overcome circuit aging," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 11, pp. 4710–4721, 2021.
- [14] C. Meng, Z. Zhou, Y. Yao, S. Huang, Y. Chen, and W. Qian, "HEDALS: Highly efficient delay-driven approximate logic synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 11, pp. 3491–3504, 2023.
- [15] S. Su, C. Meng, F. Yang, X. Shen, L. Ni, W. Wu, Z. Wu, J. Zhao, and W. Qian, "VECBEE: A versatile efficiency–accuracy configurable batch error estimation method for greedy approximate logic synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 5085–5099, 2022.
- [16] Y. Kunitake, T. Sato, H. Yasuura, and T. Hayashida, "Possibilities to miss predicting timing errors in canary flip-flops," in *International Midwest Symposium on Circuits and Systems*, 2011, pp. 1–4.
- [17] Cadence Design Systems, "Liberate characterization tool," Available at: <https://www.cadence.com/>.
- [18] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, 1990.
- [19] W. Wang, Z. Wei, S. Yang, and Y. Cao, "An efficient method to identify critical gates under circuit aging," in *International Conference on Computer-Aided Design*, 2007, pp. 735–740.
- [20] K.-C. Wu and D. Marculescu, "Aging-aware timing analysis and optimization considering path sensitization," in *Design, Automation & Test in Europe*, 2011, pp. 1–6.
- [21] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "ASAP7: A 7-nm FinFET predictive process design kit," *Microelectronics Journal*, vol. 53, pp. 105–115, 2016.