and load of each gate on critical paths, we use Monte Carlo (MC) simulation to analyze the degraded delay distribution. Although this method will increase the simulation time for timing analysis, considering that the length of critical paths is usually not too long, the MC simulation time is acceptable.

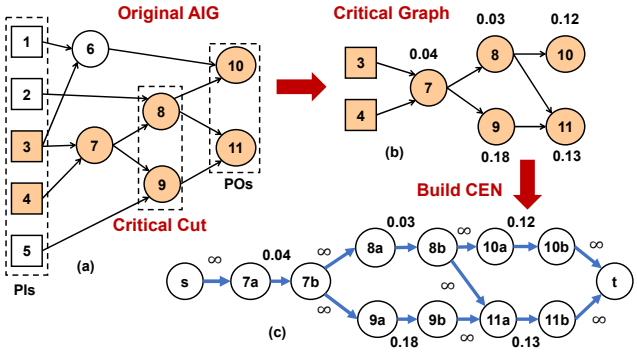

Figure 9: Illustration of delay-driven approximate logic synthesis algorithm.

5 APPROXIMATE SYNTHESIS

Approximate synthesis is a general method for approximate circuit design. In the proposed REA design framework, the approximate synthesis finds the optimal approximate local changes, which can reduce the delay and have the least error impact. Our synthesis algorithm works on the AIG representation of a circuit, and the basic procedure of the algorithm is shown in Fig. 9. In AIG representation, the delay of a circuit is proportional to the depth of AIG. The subgraph containing all critical paths is defined as the critical graph. To reduce the delay of the circuit, the depth of the critical graph needs to be reduced. Because there are usually more than one critical path, multiple approximate local changes need to be performed simultaneously on a cut of the critical paths, which is defined as a critical cut. For example, the cut with nodes 8 and 9 is a critical cut. Therefore, the problem of the approximate algorithm can be described as how to find the optimal critical cut in the critical graph, which has the minimum error impact on the circuit.

The most straightforward approach is to enumerate all sets of ALC and critical cut and then choose the combination with the minimum error. However, the total number of combinations increases exponentially with the size of the graph. To reduce the complexity, it proposes to transform this problem into a network flow problem. First, it enumerates the error impact of applying a set of ALCs. Then, it maps the original critical graph into a critical error network (CEN). Finally, it obtains the minimum cut of the CEN by solving a maximum flow problem on CEN. The minimum cut gives the optimal critical cut to be approximated.

6 EXPERIMENTAL RESULTS

In this section, the experimental results of the proposed REA design method are presented. The benchmark applications are discrete cosine transformation (DCT) and inverse discrete cosine transformation (IDCT), which are usually deployed in multimedia design to encode and decode images or videos. We deploy an 8-bit multiplier and a 16-bit adder. We used Yosys open synthesis suit [19] to synthesize the Verilog HDL codes to BLIF files as the inputs to our in-house program. Technology mapping is performed by the logic synthesis tool ABC [20] using the MCNC generic standard cell library [21]. MED is used as the error metric, which is a widely-used error metric for approximate arithmetic circuits.

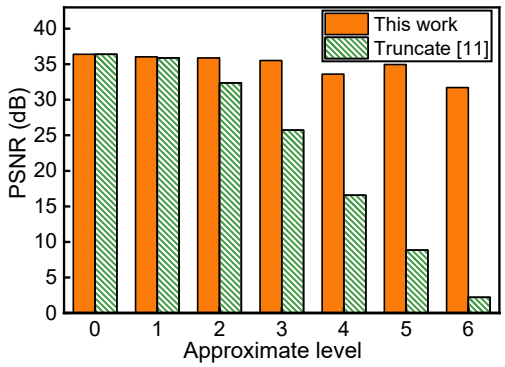

Figure 10: PSNR of circuits designed by different approximate method at the different approximate levels.

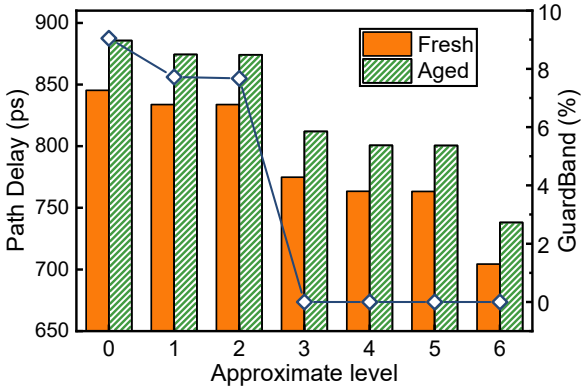

Figure 11: The fresh and aged path delay of the circuit under different approximate level and the required guardband.

Fig. 10 shows the peak signal to noise ratio (PSNR) at the different approximate levels of the 16-bit adder (approximate level is the reduced logic depth). The results indicate that, our general synthesis method can provide a finer level of approximation than simply truncating the low-significance bits [11], which is more friendly to the low-precision arithmetic circuit. The better circuit approximation is because our synthesis algorithm always adopts the ALC that has the least error impact, while truncating is not the best ALC in most cases.

Fig. 11 shows the mean circuit delay before and after aging at the different approximate levels of the adder, as well as the required guardband under these levels. The mean and variance of the delay of critical path are obtained from accurate reliability simulation, and the guardband is calculated by 3-σ standard. As

mentioned before, a cut on AIG does not necessarily means a significant reduction in critical path delay, but the delay is monotonically decreasing with the approximate level. When the approximate level is greater than 3, in other words, after 3 iterations of the design flow, the timing guardband is no longer needed, which means that the timing guardband can be completely eliminated using logic approximation.

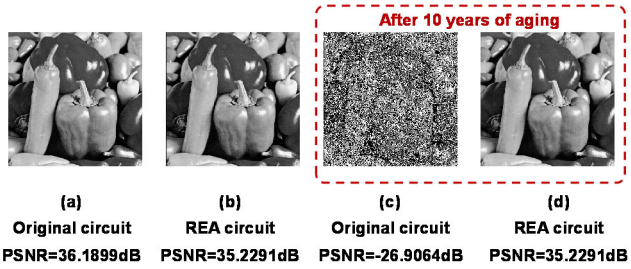

Figure 12: Images processed by the fresh original circuit (a), fresh REA circuit (b), aged original circuit (c) and aged REA circuit (d)

Based on the results of SSTA and the timing constraints, the failure rate of each path can be calculated. Fig. 12 shows the output picture of the DCT-IDCT circuit. After 10 years of aging, the image quality of the original circuit is greatly reduced due to timing error. Because timing errors are more likely to occur on longer paths, that is, the more significant bits of the adder, it will seriously affect the computing result. In contrast, for the circuit with REA design, although its initial PSNR decreases slightly (less than 1 dB), its performance after degradation does not decrease due to the shortening of the critical path. On the other hand, the logic approximation starts from the low-significance bit first, so the drop in PSNR is relatively small. The result proves that the proposed REA design can transform timing errors that have fatal effect on the circuit into less significant bit logic errors to improve resilience.

7 CONCLUSION

In this paper, a reliability-enhanced circuit design framework is proposed, which contains forward reliability simulation flow and backward approximate logic synthesis. We test our design flow in DCT-IDCT circuit design. The result shows that the aged path failure rate of the approximate circuit is much lower than the original circuit, which demonstrates that the proposed approximate logic synthesis framework can enhance the robustness of circuit and completely eliminate the aging guardband. This work provides a new perspective in reliable circuit design, especially for error-tolerant applications: the computing precision can be traded off with higher speed and longer lifetime. It also suggests that the cross-layer design framework is particularly needed in advanced technology nodes.

ACKNOWLEDGMENTS

This work is partly supported by the National Key R&D Program of China (2020YFB2205502), NSFC (61874005, 61421005, 61927901) and the 111 Project (B18001).

REFERENCES

[1] R. Huang *et al.*, "Variability-and reliability-aware design for 16/14nm and beyond technology," in *2017 IEEE International Electron Devices Meeting (IEDM)*, San Francisco, CA, USA, 2017, pp. 12.4.1-12.4.4.

[2] Z. Ji, H. Chen *et al.*, "Design for reliability with the advanced integrated circuit (IC) technology: challenges and opportunities," *Sci. China Inf. Sci.*, vol. 62, no. 12, p. 226401, Dec. 2019.

[3] Y. Cao *et al.*, "Cross-Layer Modeling and Simulation of Circuit Reliability," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 1, pp. 8-23, Jan. 2014.

[4] C. Chen *et al.*, "System-Level Modeling of Microprocessor Reliability Degradation Due to Bias Temperature Instability and Hot Carrier Injection," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 8, pp. 2712-2725, Aug. 2016.

[5] J. P. Kulkarni *et al.*, "A 409 GOPS/W Adaptive and Resilient Domino Register File in 22 nm Tri-Gate CMOS Featuring In-Situ Timing Margin and Error Detection for Tolerance to Within-Die Variation, Voltage Droop, Temperature and Aging," in *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 117-129, Jan. 2016.

[6] S. Arasu *et al.*, "Controlling Aging in Timing-Critical Paths," in *IEEE Design & Test*, vol. 33, no. 4, pp. 82-91, Aug. 2016.

[7] M. Ebrahimi *et al.*, "Aging-aware logic synthesis," in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, 2013, pp. 61–68.

[8] Z. Zhang *et al.*, "Circuit Reliability Comparison between Stochastic Computing and Binary Computing," in IEEE Transactions on Circuits and Systems II: Express Briefs, doi: 10.1109/TCSII.2020.2993273.

[9] J. Han *et al.*, "Approximate computing: An emerging paradigm for energy-efficient design," 2013 18th IEEE European Test Symposium (ETS), Avignon, 2013, pp. 1-6.

[10] M. Shafique *et al.*, "Invited - Cross-layer approximate computing: from logic to architectures," in *Proceedings of the 53rd Annual Design Automation Conference on - DAC '16*, Austin, Texas, 2016, pp. 1–6.

[11] H. Amrouch *et al.*, "Towards Aging-Induced Approximations," in *Proceedings of the 54th Annual Design Automation Conference 2017 on - DAC '17*, Austin, TX, USA, 2017, pp. 1–6.

[12] V. Gupta *et al.*, "Low-Power Digital Signal Processing Using Approximate Adders," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 1, pp. 124–137, Jan. 2013.

[13] V. Gupta *et al.*, "Low-Power Digital Signal Processing Using Approximate Adders," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 1, pp. 124–137, Jan. 2013.

[14] M. Shafique *et al.*, "A low latency generic accuracy configurable adder," *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, CA, 2015, pp. 1-6. doi: 10.1145/2744769.2744778

[15] P. Kulkarni *et al.*, "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," *2011 24th Internatioal Conference on VLSI Design*, Chennai, 2011, pp. 346-351. doi: 10.1109/VLSID.2011.51

[16] S. Su *et al.*, "A Novel Heuristic Search Method for Two-level Approximate Logic Synthesis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, pp. 1–1, 2019.

[17] Z. Zhou *et al.*, "DALS: delay-driven approximate logic synthesis," in *Proceedings of the International Conference on Computer-Aided Design - ICCAD '18*, San Diego, California, 2018, pp. 1–7.

[18] S. Guo *et al.*, "Towards reliability-aware circuit design in nanoscale FinFET technology: — New-generation aging model and circuit reliability simulator," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Irvine, CA, 2017, pp. 780–785.

[19] CliffordWolf.2016. Yosys open synthesis suite. (2016).

[20] Berkeley Logic Synthesis and Verification Group. 2018. ABC: A system for sequential synthesis and verification, release 70330. (2018).

[21] S.Yang. 1991. Logic synthesis and optimization benchmarks user guide: version 3.0. Microelectronics Center of North Carolina (MCNC).

[22] H. Amrouch *et al.*, "Estimating and Optimizing BTI Aging Effects: From Physics to CAD," 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Diego, CA, 2018, pp. 1-6.

[23] A. Mishchenko *et al.*, "DAG-aware AIG rewriting: a fresh look at combinational logic synthesis," 2006 43rd ACM/IEEE Design Automation Conference, San Francisco, CA, 2006, pp. 532-535.