

MTTF-aware Design Methodology of Error Prediction Based Adaptively Voltage-scaled Circuits

Yutaka Masuda Masanori Hashimoto

Department of Information Systems Engineering, Osaka University

Email: {masuda.yutaka, hashimoto}@ist.osaka-u.ac.jp

Abstract—Adaptive voltage scaling is a promising approach to overcome manufacturing variability, dynamic environmental fluctuation, and aging. This paper focuses on error prediction based adaptive voltage scaling (EP-AVS) and proposes an MTTF-aware design methodology for EP-AVS circuits. Main contributions of this work include (1) optimization of both voltage-scaled circuit and voltage control logic, and (2) quantitative evaluation of voltage reduction for practically long MTTF. Evaluation results show that the proposed EP-AVS design methodology achieves 20.8% voltage reduction while satisfying target MTTF.

Index Terms—adaptive voltage scaling, critical path isolation, mean time to failure, timing error predictive FF

I. INTRODUCTION

Aggressive device miniaturization due to technology scaling has been improving the average device performance. However, circuits have become sensitive to static manufacturing variability and dynamic environmental fluctuation. Moreover, device aging, which is another temporal variation and is represented by negative bias temperature instability (NBTI) [1], [2], degrades performance gradually in the field. These static and temporal variations directly lead to circuit reliability degradation. For overcoming variabilities mentioned above, design and operating margins are given in design time and in the field, respectively, for ensuring correct circuit operation. However, as the performance variation becomes significant, such margin tends to be too painful for designers. Therefore, a traditional worst-case (WC) design with guard-banding is inefficient, and an adaptive performance compensation is desired.

The most effective tuning knob for post-silicon compensation is supply voltage control, and adaptive voltage scaling (AVS) is intensively studied [3]–[7]. AVS is expected to minimize process, voltage, temperature, and aging (PVTA) margin of each chip and allocate only a little margin for the entire lifetime as shown in Fig. 1. The excessive conventional PVTA margins in most of the chips can be exploited as the source for power reduction.

There are two AVS strategies in literatures; error detection and recovery based control (e.g. Razor [3]), and error prediction and prevention based control (e.g. canary FF [8], slack monitor [7], error predictive FF [9]¹). In both the strategies, sensors are embedded to detect/predict timing errors, and the supply voltage is controlled according to the sensor outputs. Therefore, these existing works [3]–[9] focus on where to

¹There are several names, but the sensor structure is the same.

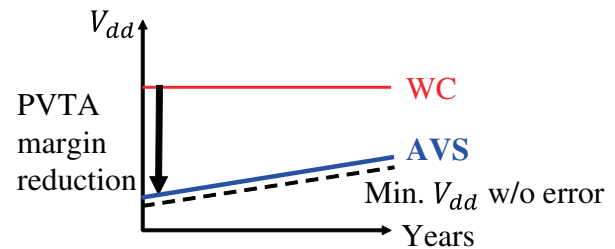


Fig. 1. Supply voltages of AVS and conventional WC in device lifetime. Ideal AVS minimizes PVTA margin of each chip.

insert sensors and how to control supply voltage and discuss the design methodology of voltage control system. Most of the conventional works embed sensors to timing-critical paths to detect/predict setup timing violation. This strategy is reasonable since timing-critical paths can be easily extracted through static timing analysis (STA) tools.

On the other hand, for implementing AVS systems that fully exploit run-time adaptation and eliminate the redundant margin, we have found that we should pay attention to the main logic circuit under AVS in addition to the sensing circuit. In the conventional VLSI design flow, there are many critical paths since the timing slack is exploited for power and area reduction, but we observe that inherent critical paths whose path delays cannot be reduced at all are limited. This observation suggests that adaptive slack assignment (ASA) to the main logic circuit under AVS, which allocates larger slack to highly active paths, could improve the efficacy of the AVS and enable further supply voltage reduction with extremely low error rate.

This work focuses on the error prediction based AVS (EP-AVS) and proposes a design methodology for EP-AVS circuits. The proposed methodology optimizes both the main logic under AVS and sensing circuit. In the main logic design, we perform an MTTF-aware ASA that utilizes critical path isolation (CPI) [10] and estimates MTTF of AVS circuits with a stochastic framework [11]. The MTTF-aware ASA enforces larger slack on the FFs that have frequent input transitions immediately before the clock edge since those FFs tend to cause setup timing errors. The number of FFs likely to fail is reduced in a design phase, and thus the insertion of error prediction sensors is facilitated in the EP-AVS design. As for the sensing circuit design, we propose a novel sensor insertion method that minimizes the sum of gate-wise timing failure

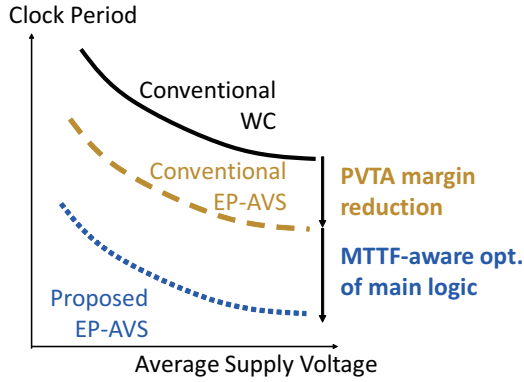


Fig. 2. Expected performance improvement thanks to the proposed EP-AVS design methodology.

probabilities, where the timing failure probability is the joint probability of activation and timing violation probability. By exploiting the information on the paths with higher timing failure probability, the proposed sensor insertion makes EP-AVS efficiently monitor the timing-critical and highly-active FFs. Experimental results show that MTTF-aware main logic design is highly compatible with EP-AVS, and they mutually enhance and provide further supply voltage reduction and performance improvement with margin elimination.

Main contributions of this work include (1) optimization of both main logic under AVS and sensing circuit, and (2) quantitative evaluation of supply voltage reduction for practically long MTTF. To best of our knowledge, this is the first work that optimizes both the main logic under AVS and sensing circuit under the explicit constraint of MTTF.

Fig. 2 illustrates the expected speed-up and/or V_{dd} reduction effects. The top black curve represents the conventional WC design that adds timing margins assuming the worst PVTA condition. The middle yellow and bottom blue curves correspond to the conventional EP-AVS without main logic optimization and the proposed EP-AVS with the ASA. The proposed EP-AVS is expected to attain a better trade-off relation between the clock period and supply voltage. This speed-up and V_{dd} reduction effects under the MTTF constraint of several years will be experimentally demonstrated.

The rest of this paper is organized as follows. Section II describes the proposed design methodology that optimizes both the main logic under AVS and sensing circuit. Section III demonstrates the supply voltage reduction and speed-up of the EP-AVS circuit designed by the proposed methodology. Lastly, concluding remarks are given in Section IV.

II. PROPOSED DESIGN METHODOLOGY FOR EP-AVS

The proposed design methodology for EP-AVS consists of the ASA for the main logic under AVS and the insertion of error prediction sensors. This section first explains the assumed EP-AVS and the overview of the proposed methodology. Then, the ASA and the sensor insertion are presented, separately.

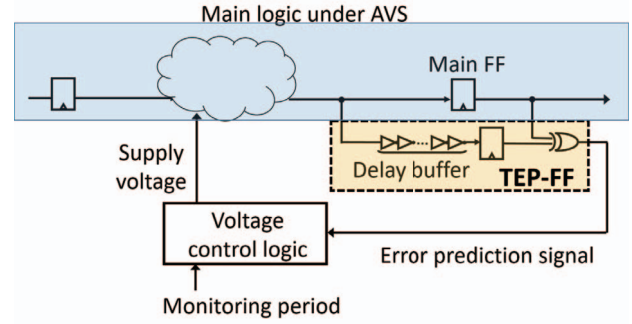


Fig. 3. Assumed EP-AVS.

A. Assumed EP-AVS

Fig. 3 illustrates an EP-AVS circuit assumed in this paper. The EP-AVS circuit is composed of the main circuit, timing error predictive flip-flop (TEP-FF) and voltage control unit. The TEP-FF consists of a flip-flop, delay buffers, and a comparator (XOR gate), and works with the main FF. When the timing margin is gradually decreasing, a timing error occurs at the TEP-FF before the main FF captures a wrong value due to the delay buffer, which enables us to know that the timing margin of the main FF is not large enough. An error prediction signal is generated to predict the timing errors, and this signal is monitored during a specified period. Note that timing errors are predicted, not detected, which is a distinct difference from Razor [3]. Once an error prediction signal is observed, the higher supply voltage is given to reduce circuit delay. Note that clock frequency is fixed throughout this paper. If no error prediction signals are observed during the monitoring period, the circuit is slowed down for power reduction. This proactive AVS is expected to overcome the variation of the timing margin which is different in every chip and varies depending on operating condition and aging.

B. Overview and Problem Definition

The proposed design methodology for EP-AVS consists of the ASA for the main logic under AVS followed by the insertion of TEP-FFs. Fig. 4 illustrates the concept of ASA. In conventional design flow, cell instances that are included in non-critical paths are replaced with smaller cells and high- V_{th} cells for reducing power dissipation and area. Consequently, this replacement decreases timing margin of many paths and may deteriorate MTTF. On the other hand, the ASA increases timing slacks of non-intrinsic critical paths as shown in Fig. 4. Meanwhile, the path-based slack assignment is not efficient since the number of paths in a circuit is huge. Therefore, this work utilizes FF-based CPI proposed in [10] to adjust setup slacks of FFs. For each FF, we assign an individual slack value to be attained after CPI. After the CPI, the paths ending at the FFs whose slack values increased are less likely to fail even when the gate delays in the paths vary, which contribute to MTTF extension. Here, it should be noted that CPI increases area and power since the intentional increase in slack was originally exploited for area and power reduction in conventional design optimization. From this sense, we need to

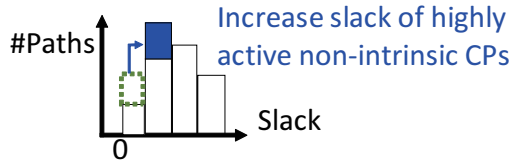


Fig. 4. Adaptive slack assignment. Larger slack is enforced on the paths that are highly active yet non-intrinsically timing critical for mitigating timing errors.

smartly perform ASA, i.e., we need to identify FFs that have less impact on area and power yet contribute to remarkable MTTF extension.

Based on the discussion above, we formulate the design optimization of EP-AVS including CPI-based ASA and TEP-FF insertion.

- Objective
 - Minimize : V_{dd}
- Variables
 - $B_{TEP_i} (1 \leq i \leq N_{FF})$
 - $B_{CPI_i} (1 \leq i \leq N_{FF})$
- Constraints
 - $MTTF \geq MTTF_{const}$
 - $N_{TEP} (= \sum_{i=1}^{N_{FF}} B_{TEP_i}) = N_{TEP}^{max}$
 - $N_{CPI} (= \sum_{i=1}^{N_{FF}} B_{CPI_i}) = N_{CPI}^{max}$

The objective of this problem is to minimize V_{dd} aiming at power minimization. The variables for optimization are B_{TEP_i} and B_{CPI_i} . B_{TEP_i} is a binary variable, and it becomes 1 when i -th FF is replaced to TEP-FF. B_{CPI_i} is also a binary variable, and it is 1 when CPI is applied to i -th FF. The primary constraint is MTTF, and the lower bound of MTTF ($MTTF_{const}$) is given as a constraint. The second constraint gives the upper bound of the number of TEP-FFs (N_{TEP}^{max}), and this limits the area increase due to TEP-FF insertion. Similarly, the upper bound of the number of FFs to which CPI is applied (N_{CPI}^{max}) is also given as a constraint to limit the area increase originating from CPI.

The proposed design methodology solves this problem with a two-stage procedure. The first stage designs the main logic under AVS using CPI [10], i.e., determines B_{CPI_i} , and the second stage performs TEP-FF insertion, i.e., determine B_{TEP_i} . The following subsections explain these two stages.

C. First Stage: Adaptive Slack Assignment in Main Logic under AVS

The CPI is performed referring to [10]. Let us briefly introduce the FF-based CPI in [10]. The CPI method focuses on gate-wise failure probability. Gate-wise failure probability is a metric that expresses the contribution to the timing failure probabilities at the downstream FFs. The detailed computation will be explained in the next subsection. Then, for maximumly reducing the sum of gate-wise failure probabilities, this method selects target FFs by solving the covering problem of instances weighted with the failure probability as an integer linear programming (ILP) problem.

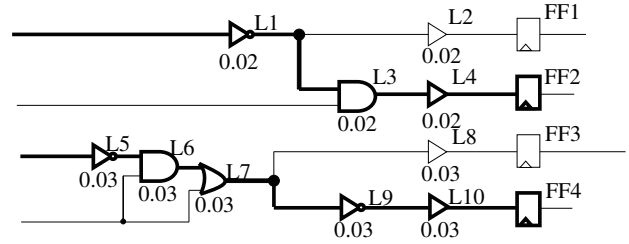
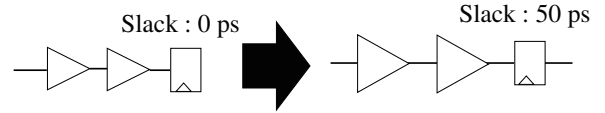


Fig. 5. FF selection strategy.



1. Increase FF setup time
+ re-synthesis
2. Restore setup time

Fig. 6. FF-based CPI.

Fig. 5 shows a simple example, where the circuit is composed of ten combinational logic cells and four FFs. The numbers attached to each gate are the gate-wise failure probabilities. Given $N_{CPI} = 2$, we can see that the most promising pairs of FFs are FF2 and FF4. When the slack times of FF2 and FF4 are increased, the slack times of L1, L3, L4, L5, L6, L7, L9, and L10 are also increased, and the expected reduction of failure probabilities at endpoint FFs corresponds to the sum of gate-wise failure probabilities and it is 0.21 ($= 0.02 + 0.02 + 0.02 + 0.03 + 0.03 + 0.03 + 0.03 + 0.03$). We note that if we choose FF3 and FF4, i.e., descending order of the sum of gate-wise failure probability, the slack times of L5, L6, L7, L8, L9 and L10 are increased. In this case, the expected reduction is 0.18 ($= 0.03 \times 6$) and this is smaller than the previous one.

Once B_{CPI_i} is determined, the FF-based CPI proceeds to the following two steps shown in Fig. 6; (1) increase setup time of the target i -th FF by $\Delta setup_i$ artificially and re-synthesize the design as an engineering change order (ECO) process, and (2) restore the setup time for the successive analysis process. With this FF-based CPI, we enforce the paths ending at the target FF to have the slack of more than $\Delta setup_i$. Note that if there are intrinsic critical paths whose path delays cannot be shortened, such paths cannot increase the slack by $\Delta setup_i$. After the CPI, the circuit area increases since conventional designs exploit such slacks for area reduction. Referring to [10], $\Delta setup_i$ is set to the upper bound value that can satisfy the setup constraint after ECO for simplicity.

D. Second Stage: Sensing Circuit Insertion

For making EP-AVS work well, TEP-FFs need to output the error prediction signals frequently to finely adjust the supply voltage, and hence it is desirable that inserted TEP-FFs are highly activated. Also, FFs with small slacks need fewer delay buffers in TEP-FFs. Here, FFs having higher timing

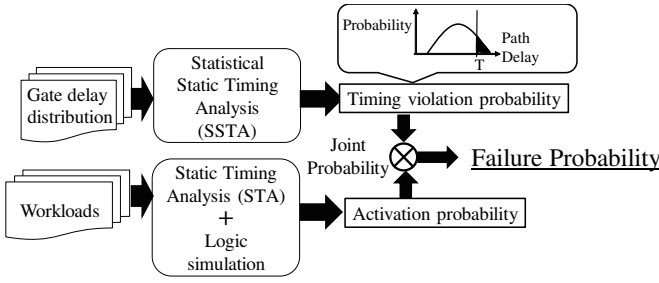


Fig. 7. Failure probability calculation.

failure probability satisfies both the desirable properties above. Therefore, we propose a novel TEP-FF insertion method that minimizes the sum of gate-wise timing failure probability aiming at MTTF maximization, which has a similarity with the CPI previously exemplified in Fig. 5. Our insertion method consists of the following two steps; (1) calculate timing failure probabilities, and (2) find out a set of FFs that maximally reduces the sum of gate-wise failure probability by solving instance covering problem as an ILP problem.

In the first step, the proposed method calculates timing failure probability of FFs, $P_{FF_i_fail}$, where timing failure probability is the joint probability of timing violation and activation. In this work, we calculate the timing violation probability by statistical static timing analysis (SSTA) and derive the activation probability of each path by associating the signal transition time in logic simulation and the path delay in STA. Then, we obtain $P_{FF_i_fail}$ by multiplying the timing violation probability and the activation probability as shown in Fig. 7.

Next, we compute the gate-wise failure probabilities, i.e., $P_{inst_k_fail}$, from the failure probability of FFs, i.e., $P_{FF_i_fail}$, as follows. Remind that $P_{inst_k_fail}$ will be utilized in the covering problem (or FF selection problem) in the second step.

$$P_{inst_k_fail} = \max\left\{\frac{P_{FF_i_fail}}{\sum_{k=1}^{N_{inst}} (B_{FF_i_inst_k})}\right\} \quad (1 \leq i \leq N_{FF}). \quad (2)$$

In Eq. (2), N_{inst} is the number of instances in the circuit. $B_{FF_i_inst_k}$ is a binary valuable which is determined by the circuit topology, and it becomes 1 when k -th instance is included in the paths ending at i -th FF. $\sum_{k=1}^{N_{inst}} (B_{FF_i_inst_k})$ is the total number of instances included in the fan-in cone of i -th FF. The above equation assumes that each instance included in the fan-in cone of i -th FF has the same contribution to the timing error at the FF, and hence the FF_fail_i is divided by $\sum_{k=1}^{N_{inst}} (B_{FF_i_inst_k})$. An instance can be included in the fan-in cones of multiple FFs. For coping with this, the max operation is performed in Eq. (2).

In the second step, we select a set of FFs that maximize the sum of gate-wise timing failure probabilities. We formulate this FF selection problem as an ILP problem to derive the exact solution. Our ILP formulation is as follows:

- Objective
 - Maximize : $\sum_{k=1}^{N_{inst}} (P_{inst_k_fail} \times B_{inst_k})$
- Constraints
 - $0 \leq B_{inst_k} \leq 1 \quad (1 \leq k \leq N_{inst})$
 - $0 \leq B_{TEP_i} \leq 1 \quad (1 \leq i \leq N_{FF})$
 - $\sum_{i=1}^{N_{FF}} B_{TEP_i} \leq N_{TEP}$
 - $B_{inst_k} \leq \sum_{i=1}^{N_{FF}} (B_{TEP_i} \times B_{FF_i_inst_k})$
- Variables
 - $B_{TEP_i} \quad (1 \leq i \leq N_{FF})$

The objective of this ILP problem is to maximize the sum of $(P_{inst_k_fail} \times B_{inst_k})$, where $P_{inst_k_fail}$ is the gate-wise failure probability, and it means how much k -th instance contributes to the timing error. B_{inst_k} is a binary variable, and it becomes 1 when k -th instance is included in paths ending at the target FFs for TEP-FF insertion. Therefore, the sum of $P_{inst_k_fail} \times B_{inst_k}$ represents the gate-wise failure probability reduction. In this problem, we assign binary variables B_{TEP_i} , where B_{TEP_i} becomes 1 when i -th FF is selected to target FFs for TEP-FF insertion.

The first and second constraints are given to restrict B_{inst_k} and B_{TEP_i} to binary numbers. The third constraint means that the number of target FFs for TEP-FF insertion should be equal or less than N_{TEP} . The fourth constraint is a key constraint that defines the relation between B_{inst_k} and B_{TEP_i} . Remind that $B_{FF_i_inst_k}$ is a binary variable which is determined by the circuit topology, and it becomes 1 when k -th instance is included in the paths ending at i -th FF. The product term of $B_{TEP_i} \times B_{FF_i_inst_k}$ becomes 1 when both B_{TEP_i} and $B_{FF_i_inst_k}$ are 1. B_{inst_k} becomes 0 only when the product of B_{TEP_i} and $B_{FF_i_inst_k}$ is 0 for all the FFs. On the other hand, if k -th instance is included in the paths ending at the target FFs, at least one of the products of B_{TEP_i} and $B_{FF_i_inst_k}$ become 1. In this case, B_{inst_k} can be 1. In this ILP formulation, we are maximizing the sum of $(P_{inst_k_fail} \times B_{inst_k})$ and hence B_{inst_k} is necessarily assigned to be 1.

III. EXPERIMENTAL EVALUATION

This section experimentally evaluates the performance improvement from the conventional WC design to the proposed EP-AVS. Section. III-A explains the evaluation setup. Section. III-B demonstrates the performance improvement results regarding supply voltage reduction and speeding-up effects. Besides, Section. III-B shows that the proposed EP-AVS extends MTTF from the conventional TEP-FF implementation, i.e., TEP-FF insertion with ascending order of FF setup slack.

A. Experimental Setup

In this work, we used the advanced encryption standard (AES) circuit and OR1200 OpenRISC processor, which is a 32-bit RISC microprocessor with five pipeline stages, as target circuits. These two circuits were designed by a commercial logic synthesizer with a 45nm Nangate standard cell library. Also, standard cell memories [12], [13] were used as SRAMs in OpenRISC processor. The synthesized circuits

include 871,000 combinational logic cells, 589,800 latches and 2,500 FFs in OpenRISC and 16,470 combinational logic cells and 530 FFs in AES, respectively. Thus, sets of N_{inst} and N_{FF} are 1,460,800 and 2,500 in OpenRISC, 16,470 and 530 in AES, respectively.

We used Gurobi Optimizer 7.0 to solve the ILP problem defined in Sections II-C and II-D. The solver was executed on a 2.4 GHz Xeon CPU machine under the Red Hat Enterprise Linux 6 operating system with 1024 GB memory. The required CPU times were at most 1.12 seconds in AES and 1.32 seconds in OpenRISC. For calculating meaningful MTTF, practical delay variations should be considered. Our evaluation took into account the following variations.

- Dynamic supply noise, which is assumed to temporally fluctuate between -50 mV and 50 mV by 10mV with eleven steps.
- Manufacturing variability, which is assumed to consist of the intra-die random variation and inter-die variation. Both the intra-die random variation and inter-die variation include NMOS and PMOS threshold voltage variation of $\sigma = 30$ mV and gate length variation of $\sigma = 1$ nm, respectively.
- NBTI aging, whose model was obtained by fitting a trapping/de-trapping model [14] to the measured data in [15]. Six degradation states of 0 mV, 0.5 mV, 1 mV, 5 mV, 10 mV and 15 mV are prepared. Note that [15] measures the NBTI degradation with stress probability of 100%, and thus the NBTI model used in our experiment does not cover recovery situation. Our future work includes to investigate the adequacy of degradation status assignment and consider the relationship between degradation and activation probability.

For performing SSTA, we generate probability density functions of gate delay variability according to the assumed variations, execute sensitivity-based SSTA (such as [16] and [17]) to obtain the canonical-form expression of the timing violation probability, and calculate the timing violation probability by integrating the canonical-form expression with MATLAB 2016b.

As for workload in OpenRISC, we selected three benchmark programs (CRC32, SHA1, and Dijkstra) from MIBenchmark [18]. For each program, 30 sets of input data were prepared for MTTF estimation. Totally, we used 90 ($= 3 \times 30$) workloads. In AES, 1,000 random test patterns were used. Fig. 8 shows the

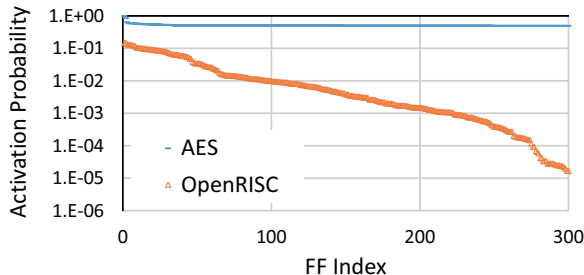


Fig. 8. Activation probability of FF.

distributions of activation probability in AES and OpenRISC. We can see that OpenRISC is a less activated circuit and the activation probability is widely spread, which suggests the CPI-based ASA is more effective to OpenRISC.

We set MTTF of 1.00×10^{17} cycles, i.e., 3.3 years in OpenRISC and 1.6 years in AES, as $MTTF_{const}$. Note that the above $MTTF_{const}$ is just an example, and the proposed design can cope with another constraint of $MTTF_{const}$ similarly. With this setup, we performed CPI-based ASA to both AES and OpenRISC. The constraint of area overhead by ASA is set to 6.0% for AES and 1.0% for OpenRISC, and N_{CPI}^{max} is set to 90 in AES and 150 in OpenRISC respectively.

Next, we inserted several TEP-FFs to the voltage-scaled circuits. The constraint of area overhead for TEP-FF is set to 1.0% for both AES and OpenRISC, and N_{TEP}^{max} is set to 18 in AES and 13 in OpenRISC respectively. When inserting TEP-FF, we need to determine the number of delay buffers for each TEP-FF. In this work, we inserted the delay buffers whose delay were comparable to the delay variation caused by 100 mV supply noise. This determination of the number of delay buffers includes room for improvement.

MTTF and average supply voltage under PVTA variation are evaluated by the stochastic MTTF estimation framework [11]. In our experiment, the monitor period for EP-AVS was set to 10^6 cycles, i.e., if no error prediction signals are outputted for 10^6 cycles, the supply voltage is decreased. This monitor period is about 1 ms in OpenRISC and 0.5 ms in AES, respectively, and it is longer than the response time of the fast transient voltage regulator, e.g., $1.6 \mu s$ in [19]. We prepared nine supply voltages from 1.20 V to 0.80 V with 50 mV interval and swept clock period from 300 ps to 800 ps in AES and from 600 ps to 2000 ps in OpenRISC. For each clock period, EP-AVS dynamically adjust the supply voltage within the range from 1.20 V to 0.80 V.

B. Evaluation Results

This subsection first shows supply voltage reduction and speed-up thanks to the proposed EP-AVS. Then, this subsection examines the effectiveness of the proposed EP-AVS regarding CPI-based ASA and TEP-FF insertion methodology.

1) *Supply Voltage Reduction and speed-up*: Fig. 9 shows the trade-off curves between the minimum average supply voltage and the clock cycle under the MTTF constraint of 10^{17} cycles, where (a) in OpenRISC and (b) in AES, respectively. The black square plots represent the conventional WC design with guard-banding for PVTA variation. The yellow circular and blue triangular plots correspond to the conventional EP-AVS which optimizes only the sensing circuit, and the proposed EP-AVS which optimizes both the main logic under AVS and sensing circuit, respectively. Here, the TEP-FFs in the conventional EP-AVS were inserted by the method in Section II-D. In this section, we examine our evaluation results from the following two aspects; (1) overall V_{dd} and clock period reduction effect thanks to the proposed EP-AVS, and (2) performance difference between the proposed and conventional EP-AVS.

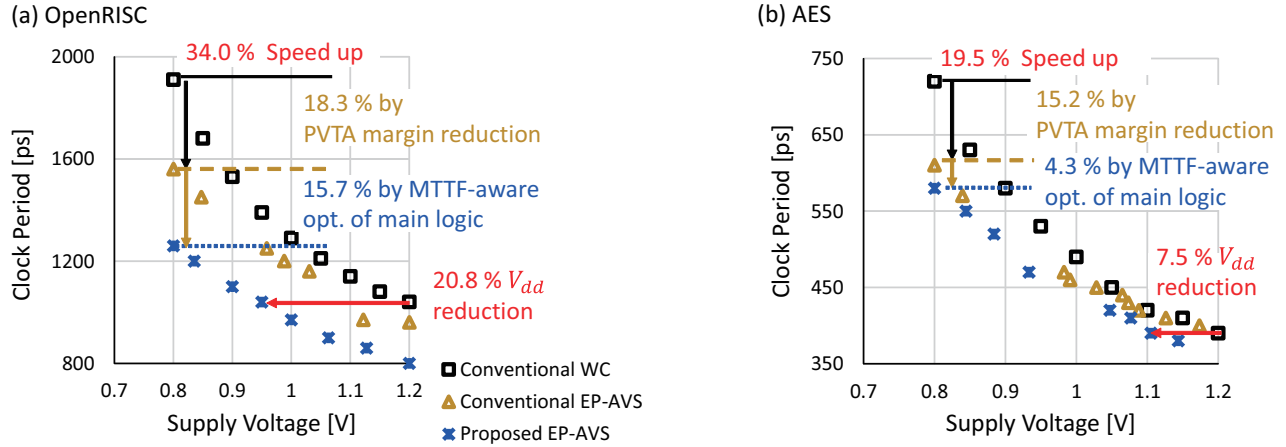


Fig. 9. Trade-off curves between clock period and voltage. (a) OpenRISC, (b) AES.

First, we compare the black square and blue triangular plots for clarifying the overall performance improvement thanks to the proposed EP-AVS. Fig. 9 shows that the proposed EP-AVS reduces average supply voltage and clock cycle time while keeping the target MTTF. For example, in Fig. 9(a), at a clock period of 1040 ps, the proposed EP-AVS achieved the target MTTF with an average supply voltage of 0.95 V, whereas the conventional WC design required 1.20 V operation. In other words, EP-AVS achieved 20.8% V_{dd} reduction from 1.20 V to 0.95 V. Similarly, in Fig. 9(b), at a clock period of 390 ps, the proposed EP-AVS achieved 7.5% V_{dd} reduction from 1.20 V to 1.11 V. As for clock period reduction, the proposed EP-AVS achieved 34.0% speed-up at 0.80 V from 1910 ps to 1260 ps in OpenRISC (Fig. 9(a)) and 19.5% speed-up at 0.80 V from 720 ps to 580 ps in AES (Fig. 9(b)), respectively. We experimentally confirmed that the proposed EP-AVS made the significant performance improvement both in AES and OpenRISC at the cost of 7.0% area increase in AES and 1.4% in OpenRISC.

Next, we compared the conventional EP-AVS and proposed EP-AVS, i.e., yellow circular and blue triangular plots. Fig. 9 shows that the proposed EP-AVS further improves performance from the conventional EP-AVS. For example, at 0.80 V, the proposed EP-AVS achieved 15.7% speed-up from 1560 ps to 1260 ps in OpenRISC and 4.3% speed-up from 610 ps to 580 ps in AES. This performance improvement reveals that the ASA for the main logic works well regarding speed-up and V_{dd} reduction and the simultaneous optimization of the main logic under AVS and the sensing circuit enhances the efficacy of EP-AVS. Here, it should be noted that AES has many FFs with higher activation probability as shown in Fig. 8, which means the paths having the slack of 0 or close to 0 tend to have high timing failure probability. Thus, the effectiveness of the ASA is smaller in AES than in OpenRISC.

2) *PVTA Margin Reduction after CPI*: Secondly, we discuss the PVTA margin reduction thanks to the proposed EP-AVS. We compared the performance of the ASA circuit without EP-AVS and the proposed EP-AVS to experimentally

confirm the compatibility between EP-AVS and the CPI-based ASA for the main logic. Fig. 10 shows the evaluation results in OpenRISC. The green diamond dots correspond to the circuit only with the ASA that takes into account the timing margin assuming PVTA WC, and the blue cross dots correspond to the proposed EP-AVS. From Fig. 10, we can see that the proposed EP-AVS achieved 22.2% speed-up at 0.8 V, i.e., from 1620 ps to 1260 ps. We note that this improvement is similar or even better than the performance improvement from the conventional WC design to the conventional EP-AVS of 18.3% in Fig. 9(a). These results reveal that the ASA for the main logic is highly compatible with EP-AVS, and they mutually enhance the performance with margin elimination.

A possible reason for the mutual enhancement is that the ASA reduces the number of failure FFs and the failure probabilities of FFs. Fig. 11 shows the failure probabilities of FFs in AES and OpenRISC. The CPI-based ASA reduces both the magnitude of the failure probability and the number of failure FFs. The number of failure FFs is reduced from 92 to 13 in OpenRISC and from 261 to 187 in AES. Therefore, the ASA helps insert TEP-FF more efficiently and hence contributes to improving performance. Thus, these results indicate that the ASA to the main logic not only enhances performance but also facilitates TEP-FF insertion.

3) *Effectiveness of the Proposed Failure Probability based TEP-FF insertion*: Lastly, we evaluate the effectiveness of the proposed TEP-FF insertion methodology that takes into account the failure probabilities of individual FFs. For comparison, we also evaluate the performance of the conventional TEP-FF insertion method, e.g., [7]–[9], that selects the insertion locations according to the order of FF setup slacks. Fig. 12 shows the comparison results of OpenRISC, where the monitor period is set to 10^9 cycles. We can see that the proposed method achieved much longer MTTF than the conventional slack-based TEP-FF insertion method. More importantly, the conventional method cannot satisfy the given MTTF constraint at all.

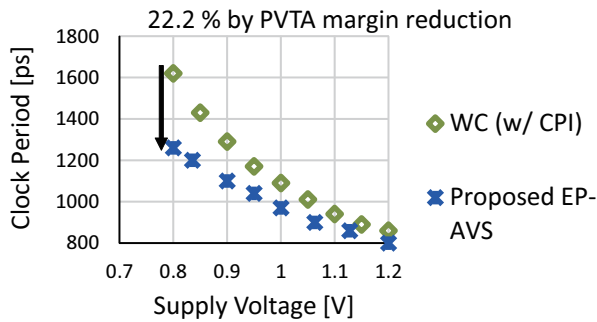


Fig. 10. PVTa margin reduction by EP-AVS after CPI-based ASA.

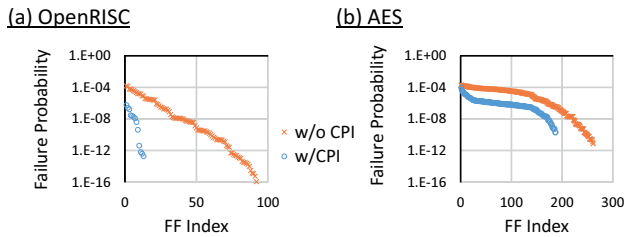


Fig. 11. Failure probability comparison between with and without the CPI-based ASA. (a) OpenRISC, (b) AES.

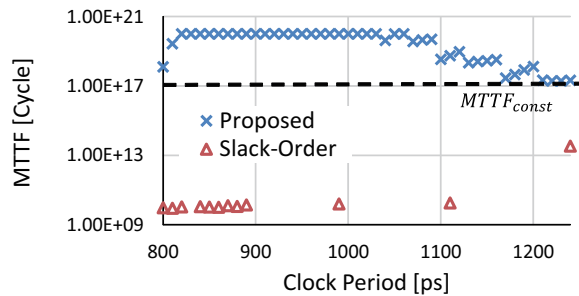


Fig. 12. MTTF comparison between the proposed TEP-FF insertion and conventional slack-based one.

IV. CONCLUSION

This paper focused on error prediction based adaptive voltage scaling (EP-AVS) and proposed a design methodology for EP-AVS circuits. The proposed design methodology optimizes both the main logic under AVS and sensing circuits taking into account the timing failure probabilities of FFs. The quantitative MTTF and supply voltage evaluation results showed that the proposed EP-AVS design methodology achieved 20.8% voltage reduction while satisfying target MTTF thanks to the ASA and failure probability based TEP-FF insertion.

ACKNOWLEDGEMENT

This work is partly supported by STARC, Socionext and ICOM Foundation, Japan.

REFERENCES

[1] B. Zang and M. Orshansky, "Modeling of nbtI-induced pmos degradation under arbitrary dynamic temperature variation," in *Proc. ISQED*, pp. 774–779, 2008.

[2] T. Wang and Q. Xu, "On the simulation of NBTI-Induced performance degradation considering arbitrary temperature and voltage variations," in *Proc. DAC*, pp. 1–6, 2014.

[3] S. Das, D. Roberts, L. Seokwoo, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "A self-tuning DVS processor using delay-error detection and correction," *IEEE Journal Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, 2006.

[4] K. A. Bowman, J. W. Tschanz, S. L. Lu, P. A. Aseron, M. M. Khellah, A. Raychowdhury, B. M. Geuskens, C. Tokunaga, C. B. Wilkerson, T. Karnik, and K. D. Vivek, "A 45nm Resilient Microprocessor Core for Dynamic Variation Tolerance," *IEEE Journal Solid-State Circuits*, vol. 46, no. 1, pp. 194–208, 2011.

[5] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. M. Harris, D. Blaauw, and D. Sylvester, "Bubble Razor: Eliminating Timing Margins in an ARM Cortex-M3 Processor in 45 nm CMOS Using Architecturally Independent Error Detection and Correction," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 66–81, 2013.

[6] S. Kim and M. Seok, "Variation-Tolerant, Ultra-Low-Voltage Microprocessor With a Low-Overhead, Within-a-Cycle In-Situ Timing-Error Detection and Correction Technique," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 6, pp. 1478–1490, 2015.

[7] A. Benhassain, F. Cacho, V. Huard, M. Saliva, L. Anghel, C. Parthasarathy, A. Jain, and F. Giner, "Timing in-situ monitors: Implementation strategy and applications results," in *Proc. CICC*, pp. 1–4, 2015.

[8] T. Sato and Y. Kunitake, "A simple flip-flop circuit for typical-case designs for DFM," in *Proc. ISQED*, pp. 539–544, 2007.

[9] H. Fuketa, M. Hashimoto, Y. Mitsuyama, and T. Onoye, "Adaptive performance compensation with in-situ timing error predictive sensors for subthreshold circuits," *IEEE Trans. VLSI Systems*, vol. 20, no. 2, pp. 333–343, 2012.

[10] Y. Masuda, M. Hashimoto, and T. Onoye, "Critical Path Isolation for Time-to-Failure Extension and Lower Voltage Operation," in *Proc. ICCAD*, 2016.

[11] S. Iizuka, Y. Masuda, M. Hashimoto, and T. Onoye, "Stochastic Timing Error Rate Estimation under Process and Temporal Variations," in *Proc. ITC*, 2015.

[12] A. Teman, D. Rossi, P. Meinerzhagen, L. Benini, and A. Burg, "Controlled placement of standard cell memory arrays for high density and low power in 28nm FD-SOI," in *Proc. ASP-DAC*, pp. 81–86, 2015.

[13] J. Shiomi, T. Ishihara, and H. Onodera, "Fully digital on-chip memory using minimum height standard cells for near-threshold voltage computing," in *Proc. PATMOS*, pp. 44–49, 2016.

[14] B. J. Velamala, K. B. Sutaria, H. Shimizu, H. Awano, T. Sato, G. Wirth, and Y. Cao, "Compact Modeling of Statistical BTI Under Trapping/Detrapping," *IEEE Trans. Electron Devices*, vol. 60, no. 11, pp. 3645–3654, 2013.

[15] H. Awano, M. Hiromoto, and T. Sato, "Variability in device degradations: Statistical observation of NBTI for 3996 transistors," in *Proc. ESSDERC*, pp. 218–221, 2014.

[16] H. Chang and S. Sapatnekar, "Statistical timing analysis under spatial correlations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 9, pp. 1467–1482, Sep. 2005.

[17] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *Proc. DAC*, pp. 331–336, 2004.

[18] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "Mibench: A free, commercially representative embedded benchmark suite," in *Proc. IEEE Workshop on Workload Characterization*, pp. 3–14, 2001.

[19] Y. Li, X. Zhang, Z. Zhang, and Y. Lian, "A 0.45-to-1.2-V Fully Digital Low-Dropout Voltage Regulator With Fast-Transient Controller for Near/Subthreshold Circuits," *IEEE Trans. Power Electronics*, vol. 31, no. 9, pp. 6341–6350, 2016.