Proactive Supply Noise Mitigation with Low-Latency Minor Voltage Regulator and Lightweight Current Prediction

Jun Chen
Osaka University
Osaka, Japan
j-chen@ist.osaka-u.ac.jp

Masanori Hashimoto
Osaka University
Osaka, Japan
hasimoto@ist.osakau.ac.jp

Abstract—**Power supply noise induces extra timing delay or even malfunctions in modern power-demanding VLSI chips. Traditional reactive noise mitigation is often too late to suppress emergent supply noise due to the long latency of voltage boosting. This paper proposes a proactive method for mitigating emergent supply noises and avoiding unexpected failures in power-hungry VLSI designs with two contributions. First, a major-minor voltage regulator (MMVR) structure, which enables quick and wide-range voltage scaling with small ripples, is proposed. Second, a lightweight current predictor consisting of a six-layer decision tree regressor achieves over 0.98 correlation for 50-cycle-ahead prediction in 25 RISC-V benchmark programs. Experimental results with a multi-core RISC-V design show that the proposed method mitigates the supply noise within 30 mV while the noise exceeds 70 mV with the conventional reactive mitigation. Also, the average supply voltage is compensated during the power-demanding operation.**

I. Introduction

With the scaling down of the technology node, both power consumption and supply noise are continuously increasing, which causes timing degradation or even malfunctions in modern power-hungry VLSI chips. Traditional reactive noise mitigation often fails to compensate for emergent supply noise due to the long latency of voltage boosting through the power delivery network (PDN). For concealing such latency, power or current future prediction is studied toward proactive noise mitigation [1]–[4]. However, existing long-term prediction requires high computation cost and, consequently, longer computation latency, which makes further longer-term prediction requirements. This negative loop makes proactive noise mitigation less effective.

In this paper, we manage to break this negative loop from two aspects. The first is to relieve the prediction length requirement by introducing scalable major-minor voltage regulator (MMVR) structure. The second is to lighten the prediction cost by developing a compact short-term average current predictor. The background and contribution to these two aspects are described in the following.

A. Related work and contribution to voltage regulator

Proactive noise mitigation requires quick and continuous voltage scaling with a wide scaling range and small voltage ripple. Switched capacitor voltage regulator (SCVR) is a popular off-chip power supply solution, but off-chip SCVR has limited voltage scaling flexibility and long response time. For addressing this problem, C. Zhan [5] and Y. Lu [6] use cascade low-dropout (LDO) voltage regulator as a secondary linear regulator for fast voltage regulation purposes. However, LDO energy efficiency drops during power-hungry operation and emergent large voltage droop. Hence, LDO is commonly applied for small-range voltage scaling or light load current scenarios. To extend the voltage scaling range, T. Andersen et al. [7] and J. Jiang et al. [8] try to scale the voltage using multiple-conversion-ratio SCVRs or reconfigurable SCVR. However, these solutions provide low ripple voltages only at a few discrete voltage levels. When dynamically switching the conversion ratio, the output ripple can be beyond 70 mV, which is 8.2% of the load voltage [7]. Meanwhile, J.-H. Lin et al. [9] use a switching regulator to scale the load voltage. However, the inductor component in this solution introduces over 10 μs voltage scaling latency, which is too long to mitigate emergent voltage droop.

To address this challenge, we propose a major-minor voltage regulator (MMVR) structure, which consists of two SCVRs whose flying capacitance is much different. MMVR can provide continuous wide-range voltage scaling capability by modulating the switching frequency of the minor voltage regulator. In our experiment, even during power-hungry operation, the MMVR has achieved over 3X voltage scaling range compared with traditional SCVR while the ripple is within 16 mV, which is 1.6% of the load voltage.

B. Related work and contribution to short-term prediction

Proactive noise mitigation relies on accurate predictions with low hardware and computational cost. Meanwhile, the prediction length, namely how far future is predicted, should be sufficiently long so that noise mitigation can take effect in time for noise occurrence. In [1]–[4], power, voltage drop, and timing delay prediction are studied. These studies commonly use internal hardware signals as input features, and use the neural network (NN), or linear regressor such as support vector machine (SVM), as a prediction engine. However,

978-1-7281-9113-3/20/$31.00 ©2020 IEEE

even though hardware signal features are carefully selected, the computational cost of NN prediction is overwhelmingly high, and the prediction can cost computation time at milliseconds level [4], which is unacceptable for run-time noise mitigation purpose. As for the SVM prediction engine, the prediction length reaches only 16 cycles [3]. Meanwhile, accurate SVM prediction is often achieved with non-linear kernel functions and a large number of support vectors. This expensive computation requires large hardware overhead and longer computation time of over 40 cycles [1], which requires even longer prediction length. Thus, a negative design loop arises and prevents proactive noise mitigation.

To address this negative loop challenge, we propose a lightweight short-term average current predictor which achieves 50-cycle prediction length and over 0.98 correlation with a six-layer decision tree (DT) regressor.

C. *Overall contribution and paper organization*

We combine the above voltage regulator and predictor solutions to proactively mitigate the supply noise in a multi-core RISC-V PDN system. Experimental results show that the proposed method can mitigate the supply noise within 30 mV while the noise exceeds 70 mV with the traditional reactive mitigation. The average supply voltage is also compensated throughout the power-hungry operation period.

The rest of this paper is organized as follows. Section II presents the overall structure of the proposed proactive supply noise mitigation system. Section II-A introduces the MMVR structure. Section II-B presents the structure and training flow of short-term current predictor followed by the noise mitigation control module in Section II-C. Section III shows experimental results, and Section IV draws the conclusion.

II. Proposed proactive noise mitigation method

Fig. 1 shows the overall PDN structure with the proactive supply noise mitigation, where off-chip PDN and a multi-core processor are included in the original design. In this work, RISC-V Rocket core [10] is used in the processor module as an example. It is noteworthy that the proposed method is basically independent of the processor core, while minor ISA-dependent adaptation is necessary.

The first key component is the major-minor voltage regulator (MMVR), which is shown as orange boxes in Fig. 1. The major VR is placed outside the chip and serves as the main power supplier. The minor VR is placed close to the cores, possibly on the chip, and serves as a voltage regulator to mitigate noise. The second key component is the prediction and control units, which are shown as blue boxes in Fig. 1. For each RISC-V core, the dedicated current predictor obtains instruction information from IO ports and then predicts future average current. The controller sums up the prediction results and decides noise mitigation action using a lookup table (LUT). A digital voltage sensor is equipped to override the mitigation action if the voltage is too high or too low for fail-safe purposes. Finally, the action signal is sent to minor VR for noise mitigation. The remaining of this section presents the details of MMVR, current predictor, and controller components, accordingly.

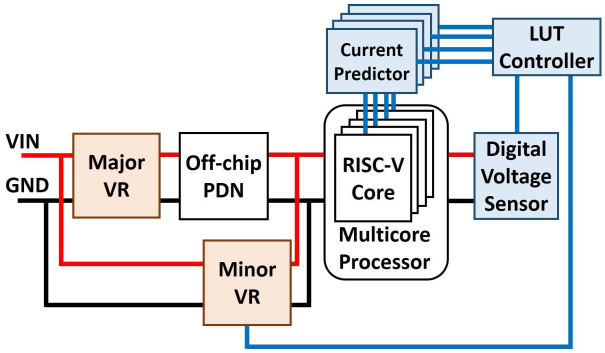


Fig. 1: Proposed structure for proactive supply noise mitigation. Red lines are power wires, black lines are ground wires, and blue lines are control signal wires.

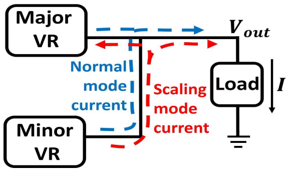


Fig. 2: MMVR connection.

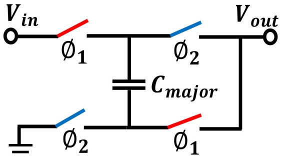


Fig. 3: Major VR with 2:1 conversion ratio.

A. *Scalable major-minor voltage regulator*

We propose a scalable switched capacitor voltage regulator called a major-minor voltage regulator (MMVR). MMVR consists of major VR and minor VR, and its simplified connection is depicted in Fig. 2. Major VR serves as a major power supplier with a fixed conversion ratio and large flying capacitance. A typical 2:1 major VR structure is shown in Fig. 3, where the switches toggle with two-phase pulses ϕ_1 and ϕ_2. C_{major} denotes the flying capacitance of major VR.

The minor VR with smaller flying capacitance is designed for voltage scaling, and it has conversion-ratio reconfigurability. By changing the switches status, the minor VR can operate in 2:1 normal mode (Fig. 4), and 3:2 scaling mode (Fig. 5). When an emergent power requirement arises, the minor VR is switched to the scaling mode. Also, the output voltage is scaled by modulating the switching frequency of minor VR. In this way, the output voltage of MMVR, V_{out}, can be scaled between 1/2 and 2/3 of input voltage V_{in}.

SCVR causes voltage ripple every time the switches are turned on and off due to its operation principle. In MMVR, the ripple depends on the operation mode. When both the major and minor VRs work in normal mode with the same switching frequency, the dynamic current flows like the blue dot line in Fig. 2, and MMVR is equivalent to a traditional SCVR. As is well studied in [11], the output ripple in normal mode can be approximated as:

$$V_{r\_norm} = \frac{\alpha I}{f_{sw}(C_{major} + C_{minor})}, \qquad (1)$$

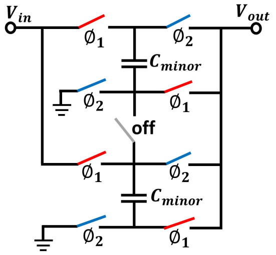


Fig. 4: Minor VR in normal mode with 2:1 conversion ratio.

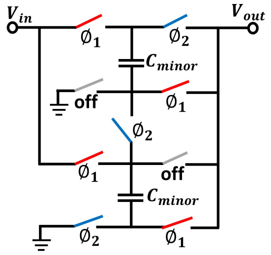


Fig. 5: Minor VR in scaling mode with 3:2 conversion ratio.

where α is a structural coefficient for both major VR and minor VR, I is dynamic load current that includes only AC component, f_{sw} is MMVR switching frequency, and C_{major} and C_{minor} are VR flying capacitance of major VR and minor VR, respectively.

On the other hand, when the minor VR works in voltage scaling mode with 3:2 conversion ratio, the output voltage of the minor VR is higher than that of the major VR. Then, the dynamic current goes from the minor VR to the major VR in addition to the load, which is illustrated as the red dot line in Fig. 2. Considering that the minor VR has a different conversion ratio, structural coefficient, and switching frequency, the ripple discussion in [11] is extended accordingly. The dynamic load current I of MMVR can be approximated as:

$$\begin{aligned} I &= I_{minor} - I_{major} \\ &= \frac{V_{r\_scale} f_{minor} C_{minor}}{\alpha_{minor}} - \frac{V_{r\_scale} f_{major} C_{major}}{\alpha_{major}}, \end{aligned} \tag{2}$$

where I_{major} and I_{minor} are the dynamic currents that go through major VR and minor VR, and V_{r_scale} is the dynamic load voltage, which is the output voltage ripple. Then, the MMVR output ripple can be derived as:

$$V_{r\_scale} = \frac{I}{\frac{f_{major} C_{major}}{\alpha_{major}} - \frac{f_{minor} C_{minor}}{\alpha_{minor}}}, \tag{3}$$

where f_{major} and f_{minor} are switching frequencies of major VR and minor VR. C_{major}/C_{minor} and α_{major}/α_{minor} are VR flying capacitances and VR structural coefficients of major VR and minor VR, respectively. Eq. (3) suggests increasing the capacitance difference between major VR and minor VR to reduce the ripple. Therefore, we intentionally use a small flying capacitance for minor VR. In our experiment in Section III, the capacitance ratio reaches ten. Such a small capacitor can be integrated into the chip package or even on the chip, and hence minor VR can be placed close to cores, and fast voltage response becomes feasible.

B. Lightweight short-term current predictor

Next, this section details the short-term current predictor. Fig. 6 shows the training and prediction flows of the predictor, where the left side illustrates the off-line training stage, and the right side shows the on-line current prediction flow. The key training and prediction procedures are represented in blue blocks.

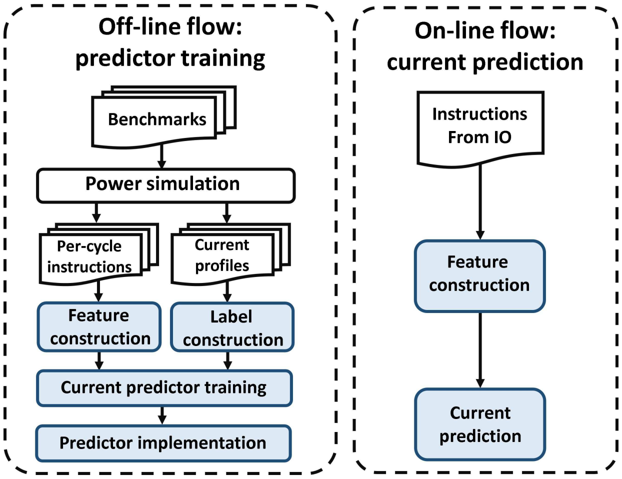


Fig. 6: Training and prediction flows with current predictor.

In the off-line training stage, firstly, the training data is prepared from benchmark programs. Simulation is performed to generate current profiles and obtain the instruction at IO ports for every cycle with logic/circuit simulator or power estimation tools. Then, we construct a set of features and labels from the instructions and raw current profiles. After that, a decision tree-based predictor is trained. The predictor hardware is implemented accordingly using the training result. In the on-line current prediction stage, firstly, the instructions are obtained from IO ports. Next, the features are constructed and given to the predictor. The prediction results are collected to the controller for MMVR noise mitigation.

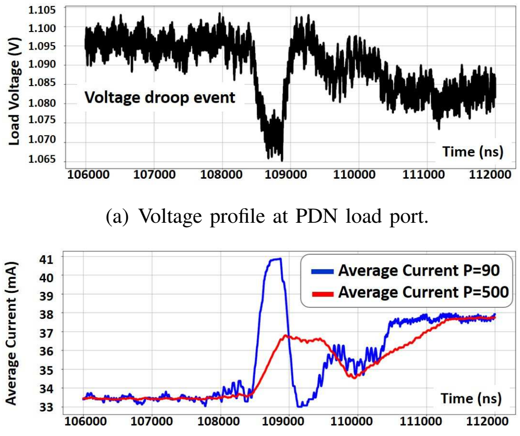
The label and feature construction, and hardware implementation are discussed in the following.

1) Prediction label construction: We use a load current value averaged over a certain duration as the training label because of the following two reasons. Firstly, the load current is independent of PDN, and therefore we can decouple the on-chip current prediction from the design of the noise controller and voltage regulator. Secondly, the averaged current value can be used as the load current at the PDN port since high-frequency cell switching current is naturally smoothed out by the parasitic impedance, especially by on-chip capacitance.

To generate the training label, we use a simple moving average (SMA) algorithm as a low pass filter to generate the average current value. The averaged current at k-th clock cycle is defined by:

$$I_{SMA}(k) = \frac{\sum_{j=(k-P+1)}^{k} I(j)}{P}, \tag{4}$$

where $I(j)$ is the average current within j-th clock cycle and P is the average period represented by clock cycle count. Here, P is determined by maximizing the summation of the correlation coefficients between voltage droop profile V^i and averaged



(a) Voltage profile at PDN load port.

(b) Average current profile with different P values.

Fig. 7: Determination of averaging period P using voltage-current correlation.

current profile I^i_{SMA} multiplied by -1 across N voltage drop events:

$$\underset{P}{\text{maximize}} \quad \sum_{i=1}^{N} \text{correlation}(V^i, -1 \cdot I^i_{SMA}). \tag{5}$$

Fig. 7 exemplifies the P selection process. First, we run a transient simulation and get the voltage profile at the PDN load port with an actual current profile and PDN model. Next, we collect the profiles of voltage droop events like Fig. 7(a) using, for example, a voltage drop threshold. For those events, we derive average current $I_{SMA}(k)$ by varying P, and calculate the correlation with Eq. (5). Then, we choose P that maximizes the average of the correlations. In the RISC-V design that will be explained in Section III-A, the correlation reaches the maximum of 0.924 with P=90. In this case, the correspondence between the voltage in Fig. 7(a) and the current in blue in Fig. 7(b) is well preserved while the high-frequency components are eliminated. If P is not appropriately selected, for example, 500 cycles, the correlation drops to 0.644, and the current pulse becomes much wider than the voltage droop, as shown by the red line in Fig. 7(b). Such a label misleads the noise mitigation action.

Next, $I_{SMA}(k)$ is shifted by $L(>0)$ clock cycles, where L corresponds to future prediction length. Then, the training label, i.e., the future averaged current at k-th clock cycle is:

$$I'_{SMA}(k) = I_{SMA}(k+L). \tag{6}$$

We will determine the prediction length L according to prediction accuracy, correlation, and implementation cost with experimental evaluations in Section III-C.

2) Prediction feature construction: Next, we discuss features suitable for future prediction supposing RISC-V instruction set as a representative one. Inspired by previous work that the processor power consumption is closely related with executed instructions [12], [13], the fundamental idea of this work is to exploit the temporal locality of processor operation and then suppose the average current in the near future has a strong correlation with the present and previous instructions. For example, when the recently fetched instructions include a lot of floating-point calculation, floating-point unit (FPU) is more likely to dominate the power consumption in several cycles. Furthermore, the instructions which will be fetched immediately after now tend to include floating-point instructions. Compared with conventional approaches that use only the current hardware signals, the longer-term prediction is expected to be feasible. On the other hand, the number of available instructions is huge. Then, for facilitating the feature construction, we categorize instructions into a small number of groups, each of which has similar hardware usage, such as FPU, cache, register files, etc., resulting in similar power dissipation.

TABLE I: Instruction categorization for RISC-V.

Type No.	Categorization	Example instruction
1	Memory load instructions	lw, ld, lh, lb
2	Memory write instructions	sw, sd, sh, sb
3	Branch instructions	bne, blt, bge, blt
4	ALU instructions	add, sub, or, and
5	Integer multiply division	mul, div, rem
6	CSR access instructions	csrrw, csrrc, csrrwi
7	PC jump instructions	j, auipc, c.j
8	Floating point instructions	fsub, fadd, fmul
9	Routine switch instructions	ret, addi sp a0 1

To put the above idea into use, we firstly decode the instructions from the RISC-V IO port and then categorize the instructions into nine types, according to Table I. We define instruction type $T_i(k)$ of k-th clock cycle as:

$$T_i(k) = \begin{cases} 1 & \text{if } k\text{-th instruction belongs to type } i, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

We use an exponential moving average (EMA) algorithm to derive features $F_i(k)$ in k-th cycle that represents how frequently i-th instruction type is fetched recently, which does not require the on-chip memory for saving the history of instruction type.

$$F_i(k) = \alpha T_i(k) + (1-\alpha)F_i(k-1), \quad (0 < \alpha < 1). \tag{8}$$

When $F_i(k)$ is close to 1, most of the recently fetched instructions belong to i-th instruction type. α is a coefficient that adjusts the weight on the current and historical instruction type. When α is close to 1, $F_i(k)$ is more sensitive to current instruction type. Conversely, when α is close to 0, longer instruction type history is included. α is determined by maximizing the summation of correlation between feature F_i and averaged current profile I_{SMA}:

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^{M} |\text{correlation}(F_i, I_{SMA})|, \tag{9}$$

where M is the feature dimension. The result of $(1/\alpha)$ can be round to the nearest power-of-two integer to further reduce hardware implementation cost.

3) Predictor implementation cost: We use DT as the prediction engine since the algorithmic complexity, and memory requirements for DT are much lower compared with SVM and NN. This advantage is critical for quick prediction with low hardware cost. Secondly, DT has non-linear regression capability even with simple computation. On the other hand, the SVM regressor, which is used in conventional works [1], [2], uses a linear kernel, which has the limited capability to regress training data. When SVM uses non-linear kernel functions, the regression to non-linear functions becomes possible. However, the computational cost for such kernel functions is usually very high. Therefore, non-linear kernel SVM is not considered in this work. Neither NN prediction is selected in this work because the computational cost of multi-layer NN inference is even higher than the SVM solution.

The hardware cost of DT predictor, denoted by H, consists of two factors:

$$H = H_{feature}(M) + H_{node}(2^D - 1), \tag{10}$$

where $H_{feature}$ is the hardware cost for instruction decoding, categorizing, and feature construction. This cost is roughly proportional to the feature dimension M, which is nine in this work with RISC-V core, and the features are listed in Table I. H_{node} is the cost for decision nodes, and it increases exponentially with decision tree depth D. Therefore, small tree depth is highly desirable. The advantage and the necessary depth of DT will be experimentally discussed in Section III-C.

C. Noise mitigation controller

The noise mitigation controller sums up the predicted values from the predictors and then uses a lookup table (LUT) to decide noise mitigation action, that is, to set the conversion ratio and the switching frequency of the minor VR. As an example, if a current jump is predicted, the controller will set the minor VR to voltage scaling mode and increase the switching frequency according to LUT. Note that the load voltage is affected by both PDN impedance and VR operation. To reduce the mitigation response time, we do not separately construct LUT for PDN and VR. Instead, we use one LUT that incorporates the relationship between the current prediction and the noise mitigation action, which is derived from the simulation that includes PDN and VR models.

For preventing wrong mitigation action at a very high or very low voltage level, an on-chip digital voltage sensor is introduced to override the wrong LUT based prediction action. A simple digital voltage sensor structure is exemplified in Fig. 8, which is found in [14]. Here, the four-bit output varies from 0000 to 1111, depending on the supply voltage level. For example, if the voltage is too low, the sensor output is 1000, voltage scaling down action is prohibited, and only scaling up action is allowed. The entire overriding rule is shown in Table II. Now, all the critical components of the proactive noise mitigation system in Fig. 1 have been prepared.

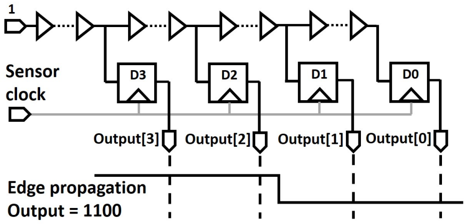


Fig. 8: Digital voltage sensor.

TABLE II: Overriding rule table with sensor output.

Output	Voltage range	Overriding rule
0000	Ultra low voltage	Perform voltage scaling up
1000	Low voltage range	Voltage scaling down prohibited
1100	Normal voltage	Accept all LUT based action
1110	High voltage range	Voltage scaling up prohibited
1111	Ultra high voltage	Perform voltage scaling down

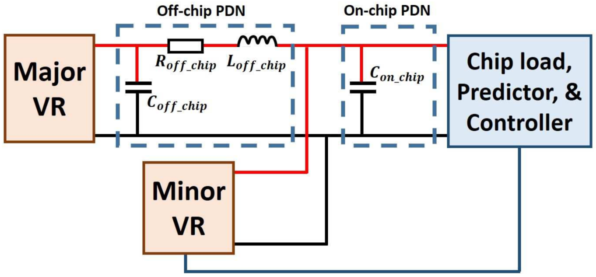


Fig. 9: PDN setup for experiments. Red lines are power wires, black lines are ground wires, and blue lines are control signal wires.

III. Experimental results

This section first introduces the experimental setup and then, presents the performance of MMVR and current predictor. Finally, we perform the system-level experiment by applying the proactive noise mitigation method to a multi-core RISC-V system to demonstrate the effectiveness of the proactive noise mitigation method.

A. Experimental setup

We use 64-bit RISC-V Rocket core [10] as chip load, and OpenRAM [15] for cache implementation. The core logic and cache are synthesized with NanGate 45 nm Open Cell Library. Nominal voltage is 1.1 V. The clock frequency is set to 0.5 GHz for both the core and cache model. Totally 25 test benchmark programs are prepared to cover most of the available functionality and usage scenarios. These benchmark programs are bare-metal C programs, which are derived from RISC-V regression test benchmarks [16], and MiBench benchmarks [17] (i.e. qsort, CRC32, Dijkstra, sha, Stringsearch, Bitcnts, basicmath, FFT, and IFFT). The description of benchmarks, indexed from 1 to 25, are summarized in Table III.

Then, the IO values regarding instructions and current profiles are generated via transistor-level simulation. Next, we perform the feature and label construction, and the total

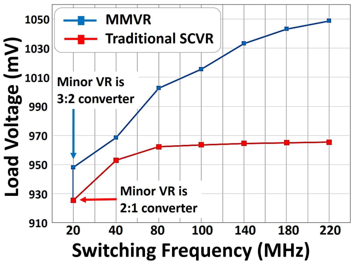


Fig. 10: Comparison in voltage scaling range.

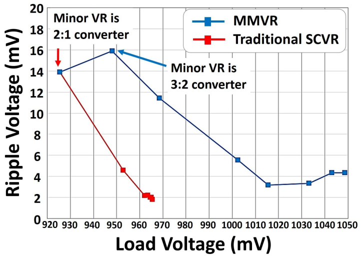


Fig. 11: Comparison in ripple voltage.

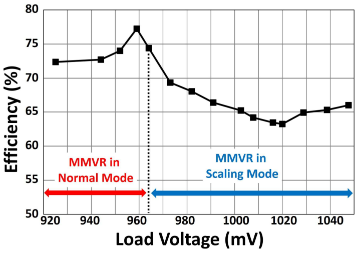


Fig. 12: MMVR efficiency versus load voltage.

TABLE III: Benchmark programs used in the experiment.

Benchmark index	Source	Operation scenario
2,4,6,8,12,13	RISC-V	Basic ALU related operation
1,3,5,11	RISC-V	Floating-point operation
7,9,10	RISC-V	Integer multiply and division
1,2,10,11,14-16	RISC-V	Branch, function call
17, 21-23	MiBench	Automotive, industrial, office
18-20	MiBench	Network
18,24,25	MiBench	Telecomm

number of data samples is 4.59 million, where 50% of the data for training and the rest for testing. P in Eq. (4) is set to 90 according to Eq. (5), and α in Eq. (8) is set to 1/32 according to Eq. (9). DT predictor is trained off-line with Sklearn package [18]. The total flying capacitance of major VR is 50 nF, and that of minor VR is 5 nF.

In the system-level experiment, Fig. 9 is used as an example of simple PDN. The off-chip PDN and on-chip PDN are simplified by lumped RLC components, where R_{off_chip} is $300\,\text{m}\Omega$, C_{off_chip} is $0.2\,\mu$F, L_{off_chip} is 100 pH, and C_{on_chip} is 10 nF.

B. MMVR performance

We compare the performance between the proposed MMVR and traditional SCVR in terms of the voltage scaling range. An 800 mA current source is attached as a load to mimic the power-hungry processor operations.

Fig. 10 shows the output voltage when the VR switching frequency is swept. The traditional SCVR output voltage is bounded at near 970 mV even with a high switching frequency, and the voltage scaling range is limited within 40 mV. On the other hand, MMVR can boost the output voltage to 1048 mV. The scaling range is 3X larger compared with the traditional SCVR.

Fig. 11 shows the output voltage ripple at different output voltage levels. The maximum ripple of MMVR is 15.9 mV. We can find MMVR and SCVR have a comparable ripple magnitude even while the major and minor voltage regulators are operating with different voltage conversion ratios and different switching frequencies.

Fig. 12 shows the MMVR conversion efficiency versus load voltage scaling range. When MMVR works in normal mode, the efficiency is identical to that of the traditional SCVR. When MMVR works in voltage scaling mode, the efficiency slightly drops, yet it is still above 63.5%. Note the scaling mode is only triggered in a short emergent period, and hence this small efficiency drop has the least impact on the overall efficiency.

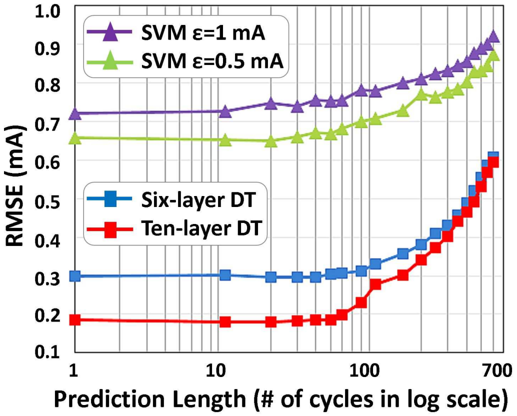


Fig. 13: RMSE versus prediction length.

Finally, we compare the voltage scaling response time. Conventional SCVR takes 226.9 ns to boost 10 mV load voltage, while MMVR takes 15.6 ns. Such a short response time relieves the prediction length requirements and makes the proactive noise mitigation possible with 50-cycle current prediction, which will be shown in Section III-D.

C. Performance and hardware cost of current predictor

We evaluate the performance of short-term current predictor using root-mean-square-error (RMSE) and correlation coefficient. The RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{N}(I'_{SMA}(j) - \hat{I}'_{SMA}(j))^2}{N}}, \qquad (11)$$

where N is the data set size, $I'_{SMA}(j)$ is the training label, which is future averaged current at j-th clock cycle, and $\hat{I}'_{SMA}(j)$ is the prediction result. The correlation coefficient is measured between $I'_{SMA}(j)$ and $\hat{I}'_{SMA}(j)$. SVM prediction engine is chosen as a comparison, where the tolerance margin ϵ is selected as 1 mA and 0.5 mA.

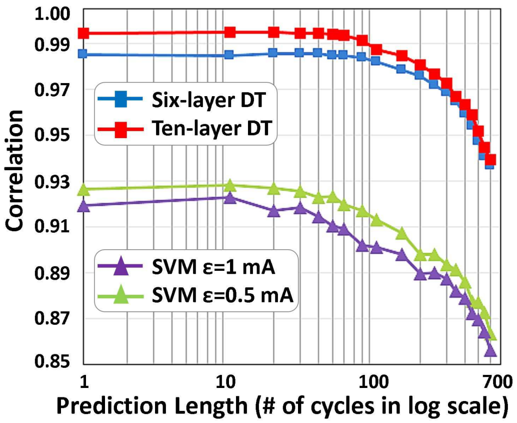


Fig. 14: Correlation versus prediction length.

TABLE IV: Prediction performance and hardware cost.

DT depth	#Nodes	Overhead(%)	RMSE(mA)	Cor.
5	31	1.48	0.327	0.980
6	63	2.51	0.304	0.984
7	127	4.57	0.247	0.989
8	255	8.68	0.227	0.991
9	511	16.91	0.205	0.992
10	1023	33.37	0.184	0.994
SVM $\epsilon(mA)$	#SVs	Overhead(%)	RMSE(mA)	Cor.
1	701	115.41	0.752	0.910
0.5	1442	236.88	0.668	0.923

We first evaluated RSME and correlation varying the prediction length. Figs. 13 and 14 show their results, where the blue and red lines are the results of the six-layer and ten-layer DTs, green and purple lines are the results of SVM predictions, respectively. The deeper DT provides longer prediction length with the same accuracy, but the correlation still drops below 0.98 beyond 100 clock cycles. On the other hand, the SVM predictor shows worse accuracy and correlation at every prediction length. We select 50 clock cycles as the prediction length because both the DTs achieve the correlation higher than 0.98, and the RMSE is almost constant, and the noise can be proactively mitigated, which will be shown in Section III-D.

Next, we compare hardware cost and prediction quality between DT and SVM predictors, where SVM uses a linear kernel. Both the predictors are designed with 8-bit floating-point representation having four fraction bits to save hardware cost and improve the prediction robustness. For minimizing the prediction latency, the predictors are designed for one-cycle completion. The hardware overhead is defined as the predictor area over the RISC-V core area. The comparison results in Table IV show that the deep DT predictor can achieve 0.994 correlation at the cost of 1023 decision nodes and 33.37% hardware overhead. When pursuing a practical lightweight predictor, the six-layer DT with 63 decision nodes is sufficient to achieve over 0.984 correlation with 2.51% overhead. Even though the SVM predictor provides worse accuracy, the number of support vectors (SVs) reaches 701, which requires 115.41% overhead for one-cycle computation. Though hardware cost can be relieved by allowing multi-cycle prediction or adopting pipeline structure, the substantial prediction length that can be used for proactive voltage boosting becomes shorter due to the latency increase and prediction throughput decrease. Thus, the SVM prediction engine cannot be adopted.

TABLE V: Prediction quality for validation set.

Validation case #	RMSE (mA)	Correlation
17, 18	0.311	0.967
18, 19	0.331	0.966
19, 20	0.341	0.981
21, 22	0.322	0.963
23, 24	0.327	0.966
Average	0.326	0.969

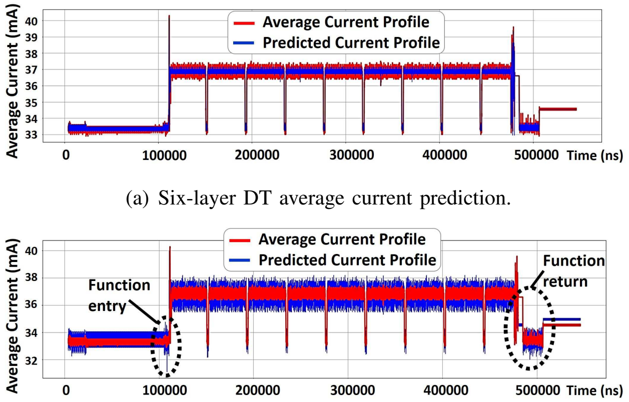
Thirdly, to verify the generality of the predictor, we performed five rounds of validation. Each round picks two Mibench cases as the validation set, the remaining 23 benchmarks as the training set. The prediction quality of the validation set is shown in Table V. While the average RMSE is slightly degraded to 0.326 mA, the average correlation of 0.969 is maintained. The prediction quality of the validation set is comparable with the full-set training result.

Fourthly, Fig. 15 demonstrates the prediction accuracy of the DT and SVM predictors in the time domain. Here, a recursive floating-point calculation benchmark is used as an example. We can see that the six-layer DT shows a better correlation with the average current profile, and the emergent current jump and drop are closely tracked. However, the SVM prediction induces a large variation from the actual future average current. Especially during the entry and return operations of the main function, as is circled in Fig. 15(b), the error of SVM miss-prediction reaches over 7.3%. Such an error can trigger wrong mitigation actions and delay the necessary mitigation action.

D. Proactive versus reactive noise mitigation

Finally, we demonstrate the effectiveness of proactive noise mitigation. For this experiment, we build up a four-core RISC-V PDN system with the proposed proactive noise mitigation method. The system-level structure with proactive mitigation is shown in Fig. 1. As a comparison, we also set up a reactive noise mitigation method that modulates minor VR to boost the voltage if the load voltage drops below the low bound. For both the mitigation methods, we set the low voltage bound as 1010 mV. To compare the worst-case voltage drop, we run the same benchmark in each core, resulting in a large voltage drop.

The voltage waveforms at the load are shown in Fig. 16, where the blue waveform corresponds to the proactive noise mitigation method, and the red waveform is the reactive



(a) Six-layer DT average current prediction.

(b) SVM average curent prediction with ϵ=0.5 mA.

Fig. 15: Current prediction results with DT and SVM.

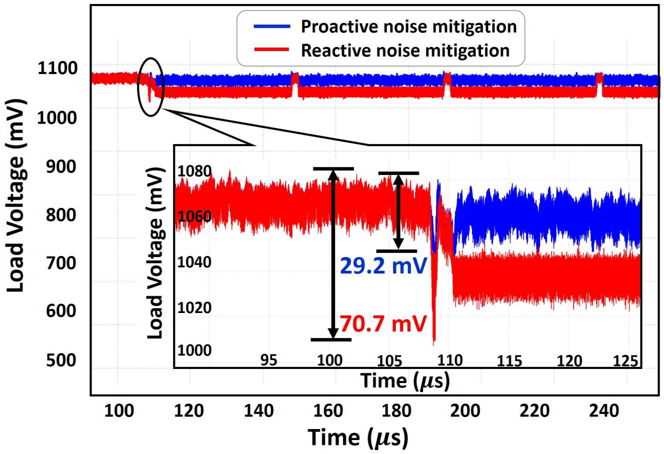


Fig. 16: Noise mitigation result for multi-core RISC-V PDN.

mitigation method. In the proactive noise mitigation case, the voltage is above 1030 mV, and the voltage recovers in 40 ns. Furthermore, the proactive noise mitigation method can stabilize the average load voltage around 1060 mV, with a ripple of less than 30 mV. As for the reactive mitigation, the voltage drop exceeds 70 mV, and the voltage goes below the 1010 mV bound because of the PDN latency. Also, the average voltage drop exceeds 20 mV during the power-hungry operation period after 115 μs. The proposed proactive noise mitigation can contribute to avoiding unexpected failures originating from emergent voltage drop.

IV. Conclusion

In this paper, we have proposed a proactive noise mitigation method that introduces MMVR and lightweight short-term current predictor for enabling quick and continuous voltage scaling. MMVR provides over 3X scaling range compared with traditional SCVR even while the ripple is suppressed within 16 mV. The current predictor, which is implemented with a simple six-layer decision tree and achieves over 0.98 correlation for 50-cycle prediction length with the hardware overhead of 2.51%. Finally, the system-level simulation validates the effectiveness of the proposed method. The voltage drop is mitigated within 30 mV by the proposed proactive mitigation, while it is 70 mV for traditional reactive mitigation. The emergent voltage droop and its risk of unexpected failure are mitigated.

References

[1] F. Ye, F. Firouzi, Y. Yang, K. Chakrabarty, and M. B. Tahoori, “On-Chip Droop-Induced Circuit Delay Prediction Based on Support-Vector Machines,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 4, pp. 665–678, Apr. 2016.

[2] M. Kaliorakis, A. Chatzidimitriou, G. Papadimitriou, and D. Gizopoulos, “Statistical Analysis of Multicore CPUs Operation in Scaled Voltage Conditions,” *IEEE Computer Architecture Letters (CAL)*, vol. 17, no. 2, pp. 109–112, Jan. 2018.

[3] V. J. Reddi, M. S. Gupta, G. Holloway, G.-Y. Wei, M. D. Smith, and D. Brooks, “Voltage emergency prediction: Using signatures to reduce operating margins,” *Proc. IEEE Int. Conf. High-Perform. Comput. Archit.*, pp. 18–29, Feb. 2009.

[4] S. N. Mozaffari et al., “An Efficient Supervised Learning Method to Predict Power Supply Noise During At-speed Test,” *2019 IEEE International Test Conference (ITC)*, pp. 1-10, 2019.

[5] C. Zhan and W. Ki, “Analysis and Design of Output-Capacitor-Free Low-Dropout Regulators With Low Quiescent Current and High Power Supply Rejection,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 2, pp. 625–636, Feb. 2014.

[6] Y. Lu, “A Reconfigurable Switched-Capacitor DC-DC Converter and Cascode LDO for Dynamic Voltage Scaling and High PSR,” *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 509-511, Oct. 2018.

[7] T. Andersen et al., “A 10 W On-Chip Switched Capacitor Voltage Regulator With Feedforward Regulation Capability for Granular Microprocessor Power Delivery,” *IEEE Transactions on Power Electronics*, vol. 32, no. 1, pp. 378–393, Feb. 2016.

[8] J. Jiang, W.-H. Ki, and Y. Lu, “Digital 2-/3-Phase Switched-Capacitor Converter With Ripple Reduction and Efficiency Improvement,” *IEEE J. Solid-State Circuits*, vol. 52, no. 7, pp. 1836–1848, Apr. 2017.

[9] J.-H. Lin et al., “A high-efficiency and fast-transient digital-low-dropout regulator with the burst mode corresponding to the power-saving modes of DC–DC switching converters,” *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pp. 314–315, Feb. 2018.

[10] Asanovic et al., “The Rocket Chip Generator,” Technical Report UCB/EECS-2016-17, EECS Department, University of California, Berkeley, Apr. 2016.

[11] B. Zimmer et al., “A RISC-V Vector Processor With Simultaneous-Switching Switched-Capacitor DC–DC Converters in 28 nm FDSOI,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 930–942, Apr. 2016.

[12] A. Sinha, N. Ickes and A. P. Chandrakasan, “Instruction level and operating system profiling for energy exposed software,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 6, pp. 1044–1057, Dec. 2003.

[13] C. Cernazanu-Glavan et al., “Direct FPGA-based power profiling for a RISC processor,” *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pp. 1578–1583, Jul. 2015.

[14] C. Lefurgy, A. Drake, M. Floyd, M. Allen-Ware, B. Brock, J. Tierno, J. Carter, and R. Berry, “Active guardband management in power7+ to save energy and maintain reliability,” *IEEE Micro*, vol. 33, no. 4, pp. 35–45, Jul. 2013.

[15] M. R. Guthaus, J. E. Stine, S. Ataei, Brian Chen, Bin Wu and M. Sarwar, “OpenRAM: An open-source memory compiler,” *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–6, Nov. 2016.

[16] riscv-tests. [Online]. Available: https://github.com/riscv/riscv-tests

[17] 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,” *Proceedings of the Fourth Annual IEEE International Workshop on Workload Characterization*, pp. 3–14, Dec. 2001.

[18] F. Pedregosa et al., ”Scikit-learn: Machine learning in Python”, *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.