PROACTIVE SUPPLY NOISE MITIGATION AND DESIGN METHODOLOGY FOR ROBUST VLSI POWER DISTRIBUTION

Masanori Hashimoto and Jun Chen
Department of Information Systems Engineering, Osaka University, Osaka 565-0871, Japan
Email: hasimoto@ist.osaka-u.ac.jp

ABSTRACT

To robustly provide the low-noise supply voltage through a VLSI power distribution network (PDN) system, we have proposed a proactive noise mitigation system and improved the low-noise PDN design methodology. For the proactive noise mitigation, we present a lightweight current prediction solution, which predicts the near future noise, and a major-minor voltage regulator (MMVR) structure to provide the fast and wide-range voltage scaling capability. For the low-noise PDN design methodology, we introduce a chip load model to fill the gap between off-chip PDN design and on-chip timing information.

INTRODUCTION

A high-quality low-noise power distribution system is critically important to ensure the robust performance of next-generation VLSI system. It is because the supply noise magnitude is continuously increasing, and timing sensitivity to noise becomes more and more severe [**1**]. With this trend, extensive efforts are required at both VLSI PDN design and operation stages.

A low-noise PDN structure, which is our proposal, is depicted in Figure 1, where the original PDN components are sounded by dotted lines. Conventionally, reactive noise mitigation is a typical solution [**2**]. However, such solution is usually too late for emergent voltage drop due to systematic issues such as voltage sensing latency and voltage boosting latency through PDN. On the other hand, existing proactive noise mitigation methods are less effective because of considerable hardware overhead, short prediction length [**3**], or limited voltage regulator (LDO, SCVR, etc.) scaling capability [**4**]. Meanwhile, during the off-chip PDN design, designers rely on an over-simplified chip load model, such as current source [**5**] or equivalent RC circuit models [**6**], to refine the PDN circuit without actual on-chip timing information. Hence, such a methodology may result in under- or over-designed PDN.

The objective of this paper is to achieve robust VLSI power distribution by firstly introducing a proactive noise mitigation system [**7**], which consists of lightweight predictor and major-minor voltage regulator (MMVR) corresponding to green boxes in the Figure 1. Secondly for the PDN design methodology, an improved chip load model is derived from [**8**], which can reveal the key on-chip timing information and consider interdependency between voltage, current, and timing.

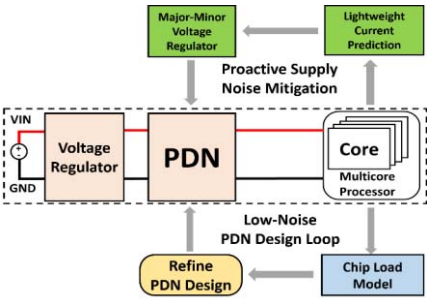

Figure 1: Overall PDN Solution.

PROACTIVE NOISE MITIGATION

Figure 2 shows the overall proactive supply noise mitigation structure, where off-chip PDN and multi-core processor are included as the original design. The first key component is the major-minor voltage regulator (MMVR), which consists of orange boxes in Figure 2. The major VR is placed outside the chip and serves as the main power supplier. The minor VR is placed close to the cores, and serves as a voltage regulator to mitigate noise. The second key component is prediction and control units, which are blue boxes in Figure 2. For each core, the current predictor predicts future average current. The controller sums up the prediction results and decides noise mitigation action. Then, the action signal is sent to minor VR for noise mitigation.

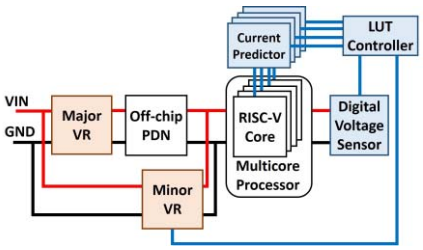

Figure 2: Proposed Proactive Noise Mitigation.

Lightweight Current Prediction

Figure 3 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.

2021 China Semiconductor Technology International Conference (CSTIC) | 978-1-6654-4945-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/CSTIC52283.2021.9461432

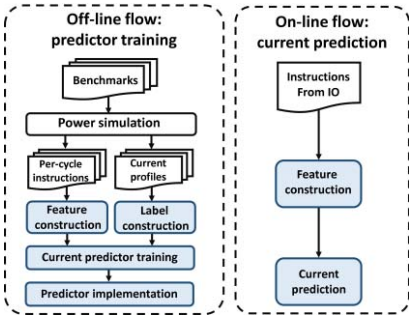

Figure 3: Training and Prediction Flow.

In the off-line training stage, first 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 training label and features 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, first 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.

For prediction label construction, we use a load current value averaged over a certain duration as the training label, mainly because the load current is independent of PDN, and therefore we can decouple the on-chip current prediction from the design of PDN, noise controller and voltage regulator. 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 on-chip capacitance. To generate the training label, we use 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:

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

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

$$\max_P \sum_{i=1}^{N} correlation(V^i, -1 \cdot I^i_{SMA}). \quad (2)$$

For prediction feature construction, we 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. We categorize instructions into nine types, which are memory load, memory write, branch instructions, ALU instructions, integer multiply instructions, CRS access, PC jump, FPU instructions, and routine switch instructions. Each type has the similar hardware usage. We define instruction type $T_i(k)$ of k-th clock cycle as:

$$T_i(k) = \begin{cases} 1 & if\ instruction \in type\ i \\ 0 & otherwise. \end{cases} \quad (3)$$

We use the exponential moving average (EMA) algorithm to derive feature $F_i(k)$ in k-th cycle that represents how frequently i-th instruction type is fetched recently, which requires no on-chip memory for saving the history of instruction type:

$$F_i(k) = \alpha T_i(k) + (1-\alpha) F_i(k-1), \quad (4)$$

where $0 < \alpha < 1$, and α is determined by maximizing the summation of correlation between feature F_i and averaged current profile I_{SMA}.

For predictor implementation, we use DT as the prediction engine since the algorithmic complexity and memory requirements for DT are much lower compared with SVM and NN. Besides, DT has non-linear regression capability even with simple computation. The noise mitigation controller sums up the predicted values from the predictors and then uses lookup table (LUT) to decide noise mitigation action. For preventing wrong mitigation action at very high or very low voltage level, an on-chip digital voltage sensor is introduced to override the wrong LUT based prediction action.

Major-Minor Voltage Regulator (MMVR)

MMVR consists of major VR and minor VR, and its simplified connection is depicted in Figure 4. In normal operation mode, both major and minor VR supply power to the chip load, and the current flow is shown as the blue dot line. When the minor VR works in voltage scaling mode, 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 Figure 4.

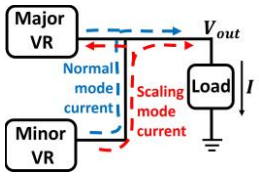

Figure 4: MMVR.

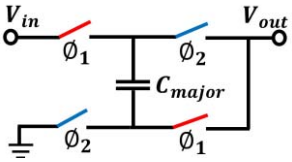

Figure 5: Major VR.

Here, the 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 Figure 5, 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 the voltage scaling, and it has the conversion-ratio reconfigurability. By changing the switches status, the minor VR can operate in 2:1 normal mode (Figure 6), and 3:2 scaling mode (Figure 7). 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, $\boldsymbol{V_{out}}$, can be scaled between 1/2 and 2/3 of the input voltage $\boldsymbol{V_{in}}$. Now, the key components of the proactive noise mitigation system in Figure 2 have been prepared.

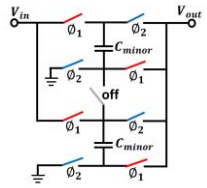

Figure 6: Normal Mode.

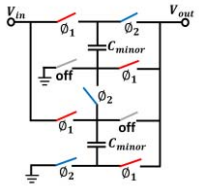

Figure 7: Scaling Mode.

PDN DESIGN METHODOLOGY USING CHIP LOAD MODEL

The main challenge for a chip load model to reveal timing information, is to replay the interdependency between voltage, current, and timing. To address this challenge, we propose a chip load model composed of three sub-models as explained with Figure 8.

The time-voltage-variant resistor model is responsible for reproducing the switching current in time

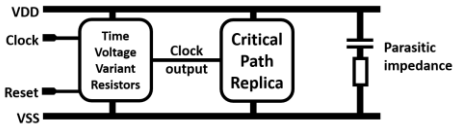

Figure 8: Chip Load Model Overall Structure.

domain. This component is derived from current profile of chip design modules and clock tree. The current profiles are characterized from a range of voltage levels. Then, current profiles are converted to resistance profile (RP) according to Ohm's law. The result $\boldsymbol{RP}$ is expressed as

$$\boldsymbol{RP} = (\boldsymbol{T_N}\ \boldsymbol{R_N}),$$

where $\boldsymbol{T_N}$ and $\boldsymbol{R_N}$ are time and resistance vectors, and can be described in Verilog-A. Further, we can derive RP for various operation mode, and use logic interface, e.g., set and reset signal, to switch operation mode of chip load model. The output of time-voltage-variant resistor model is the clock output signal with latency, which is also characterized from clock tree module.

The critical path replica model takes the output of clock, and reproduces the propagation delay in a set of the representative critical paths. The parasitic impedance is responsible for reproducing the voltage-current response in high frequency-domain. By these components, the interdependency between voltage, current, and timing is addressed, and the on-chip timing information reflecting the off-chip PDN can be provided to off-chip designers.

EXPERIMENTAL RESULTS

This section presents proactive noise mitigation result, including lightweight current prediction, and system level results. Then, the performance of the proposed chip load model is presented. We use 64-bit RISC-V Rocket core [**9**] as chip load, and OpenRAM [**10**] for cache implementation. The chip load is synthesized with NanGate 45 nm Open Cell Library. Training data is from RISC-V regression, and MiBench benchmarks [**11**].

Lightweight Current Prediction Results

We evaluate the performance of predictor using root-mean-square-error (RMSE) and correlation coefficient of near future averaged current. Figures 9 and 10 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.

The hardware overhead is defined as the predictor area over RISC-V core area. When pursuing a practical lightweight predictor, the six-layer DT with 63 decision nodes is sufficient to achieve over 0.984 correlation with

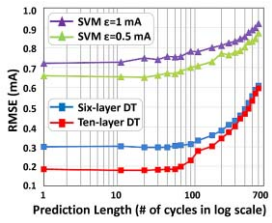

Figure 9: RMSE Comparison DT v.s SVM.

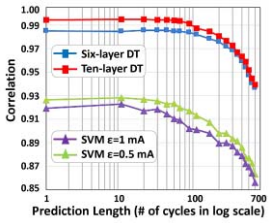

Figure 10: Correlation Comparison DT v.s SVM.

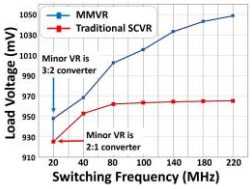

Figure 11: Scaling Range Comparison.

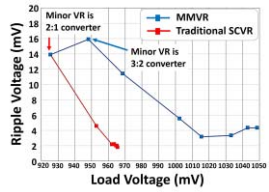

Figure 12: Output Ripple Comparison.

2.51% area overhead.

MMVR Performance Results

We compare the performance between the proposed MMVR and traditional SCVR in terms of the voltage scaling range and output ripple.

Figure 11 shows the output voltage when the VR switching frequency is swept. The traditional SCVR output voltage is bounded at near 970mV, while MMVR can boost the output voltage to 1048 mV. The scaling range is 3X larger. Figure 12 shows the output voltage

ripple at different output voltage levels. The maximum ripple of MMVR is 15.9mV. We can find MMVR and SCVR have a comparable ripple magnitude. Besides, conventional SCVR takes 226.9 ns to boost 10mV 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.

System-level Proactive Noise Mitigation Results

The system-level structure with proactive mitigation is shown in Figure 2. As a comparison, we set up both proactive and reactive noise mitigation methods, the low voltage bound is set as 1010mV. The voltage waveforms at the load are shown in Figure 13, where the blue waveform corresponds to the proactive noise mitigation method, and the red waveform is the reactive mitigation method. In the proactive noise mitigation case, the voltage is above 1030 mV, and the voltage recovers in 40 ns, with ripple of less than 30mV. 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. The proposed proactive noise mitigation can contribute to avoiding unexpected failures from emergent voltage drop.

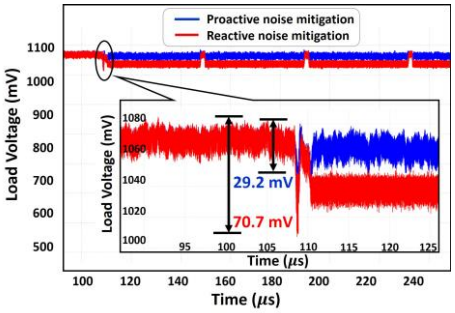

Figure 13: Noise Mitigation Results for PDN.

Chip Load Model Performance

The cycle-by-cycle critical path slack is compared between transistor-level SPICE netlist and the proposed chip load model. The slack comparison result of on-chip critical path is shown in Figure 14, where the average estimation error of the path slack is 0.1%, and the maximum error is 2.6%. In this simulation, the simulation with full transistor-level SPICE netlist takes 68,537 s, while that with the proposed model takes 172 s which means over 300X runtime reduction. The traditional current source model cannot reveal timing information.

CONCLUSION

To achieve robust low-noise PDN VLSI design, this paper presented a proactive noise mitigation system, which consists of lightweight predictor and major-minor voltage regulator (MMVR). The proposed structure can mitigate emergent voltage drop with reasonable hardware cost. For the PDN design methodology, this paper introduced a fast and accurate chip load model, which can replay the on-chip timing information and help off-chip PDN designer to refine of-chip PDN circuit.

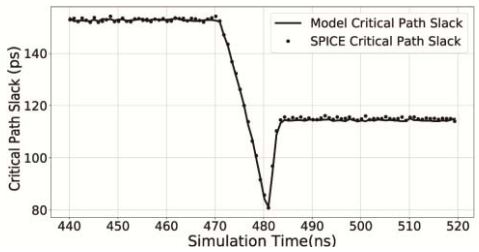

Figure 14: Critical Path Slack in Transient Process.

REFERENCES

[1] M. Swaminathan, "Power distribution networks for system-on-package: status and challenges," *IEEE Trans. Adv. Packaging*, May 2004.

[2] T. Burd et al., "A dynamic voltage scaled microprocessor system," *IEEE J. Solid-State Circuits*, Nov. 2000.

[3] F. Ye, "On-Chip Droop-Induced Circuit Delay Prediction Based on Support-Vector Machines," *IEEE Trans. CAD*, April 2016.

[4] T. Souvignet, "A Fully Integrated Switched-Capacitor RegulatorWith Frequency Modulation Control in 28-nm FDSOI," *IEEE Trans. Power Electronics*, July 2016.

[5] W. Cui et al.,"Modeling the network processor and package for power delivery analysis," *Proc. Symp. on Electromagnetic Compatibility*, 2005.

[6] S. Lin and N. Chang, "Challenges in power-ground integrity," *Proc. ICCAD*, 2001.

[7] Jun Chen and Masanori Hashimoto "Proactive Supply Noise Mitigation with Low-Latency Minor Voltage Regulator and Lightweight Current Prediction," *Proc. ITC*, 2020

[8] Jun Chen et al.,"A Multicore Chip Load Model for PDN Analysis Considering Voltage-Current-Timing Interdependency and Operation Mode Transitions," *IEEE Trans. Components, Packaging and Manufacturing Technology*, Sept. 2019.

[9] K. Asanovic et al., "The Rocket Chip Generator," *Technical Report UCB/EECS-2016-17*, EECS Department, University of California, Berkeley, Apr. 2016.

[10] M. R. Guthaus et al., "OpenRAM: An open-source memory compiler," *Proc. ICCAD*, 2016.

[11] M. R. Guthaus et al.,"MiBench: A free, commercially representative embedded benchmark suite," *Proc. International Workshop on Workload Characterization*, 2001.