Synthesis of Hardware Performance Monitoring and Prediction Flow Adapting to Near-Threshold Computing and Advanced Process Nodes

Jeongwoo Heo[1], Kwangok Jeong[2], Taewhan Kim[3], and Kyumyung Choi[4]

[1,3,4]School of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
[2]Foundry Business, Samsung Electronics Co., Ltd., Hwaseong-si, Gyeonggi-do, Korea
e-mail: [1]jwheo@snucad.snu.ac.kr, [2]kwang@samsung.com, [3]tkim@snucad.snu.ac.kr, [4]kmchoi@snu.ac.kr

Abstract— An elaborate hardware performance monitor (HPM) has become increasingly important for handling huge performance variation of near-threshold computing and recent process technologies. In this paper, we propose a new approach to the problem of predicting critical path delays (CPDs) using HPM. Precisely, for a target circuit or system, we formulate the problem of finding an efficient combination of ring oscillators (ROs) for accurate prediction of CPDs on the circuit as a mixed integer second-order cone programming and propose a method of minimizing the total number of ROs for a given pessimism level of prediction. Then, we propose a prediction flow of CPDs through statistical estimation of process parameters from measurements of the customized HPM and machine learning based delay mapping from the estimation. For a set of benchmark circuits tested using 28nm PDK and 0.6V operation, it is shown that our approach is very effective, reducing the pessimism of CPDs and minimum supply voltages by 6.7∼52.9% and 20.6∼50.8% over those of conventional approaches, respectively.

I. Introduction

Most power consumption of current CMOS circuits occurs in the charging/discharging process of capacitance, and it increases with supply voltage (V_{dd}) quadratically, in spite of the non-scalability of threshold voltage (V_{th}). Therefore, it could be possible to reduce energy per operation dramatically by lowering V_{dd}, as shown in Fig. 1(a), compared to the super-threshold voltage (super-V_{th}) regime ($V_{\mathrm{dd}} \gg V_{\mathrm{th}}$). Contrary to severe performance degradation in the sub-threshold voltage (sub-V_{th}) regime ($V_{\mathrm{dd}} < V_{\mathrm{th}}$), the near-threshold voltage (NTV) regime ($V_{\mathrm{dd}} \gtrsim V_{\mathrm{th}}$) provides a well-balanced trade-off between performance and energy efficiency and could be a more practical alternative to low-voltage operation [1]. However there are several barriers to the use of NTV operation, one of which is how to handle the significant increase in performance variation, illustrated in Fig. 1(b). Simply adding a margin for handling such a large performance variation sacrifices significant performance loss for near-threshold computing (NTC).

Wide performance spread is not the only concern of NTC. Fig. 2(a) shows normalized variations of an effective channel length (L_{eff}) at different process technologies, indicating that the variation at 5nm is expected to be increased by 62% over that at 28nm. Besides, Meinhardt, Zimpeck, and Reis [3] reported that the impact of the variation of gate workfunction from metal gate granularity on I_{ON} current is increased by up to 2X at 7nm in comparison with that at 20nm, as shown by the blue curve in Fig. 2(b). These results cause large performance variation, and as a result, handling it is increasingly important for advanced process technologies as well as NTC.

For handling the variation for NTC and recent process nodes, speed binning could be used, where each die is classified by its maximum operating frequency (f_{max}). However, its usage is restricted to specific areas, such as a processor chip. Another solution is applying adaptive voltage scaling (AVS), in which, after timing closure is completed with the worst corner, lower V_{dd} ($V_{\mathrm{dd,min}}$) is applied to fast dies to save their power consumption. The essential condition for the success of AVS is that the measurement for estimating the amount of variation for each die should be sufficiently accurate. As a result, much research is being actively conducted on hardware performance monitor (HPM) to meet the condition in academia and industry [4], [5], [6], [7], [8].

One of the most reasonable approaches using HPM is to exploit statistical analyses concerning physical parameters (PPs). It is an extension of statistical static timing analysis (SSTA) and starts from a linear model between HPM measurements and critical path delays (CPDs) to the variations of PPs. Liu and Sapatnekar [4] estimated the amount of spatial variation

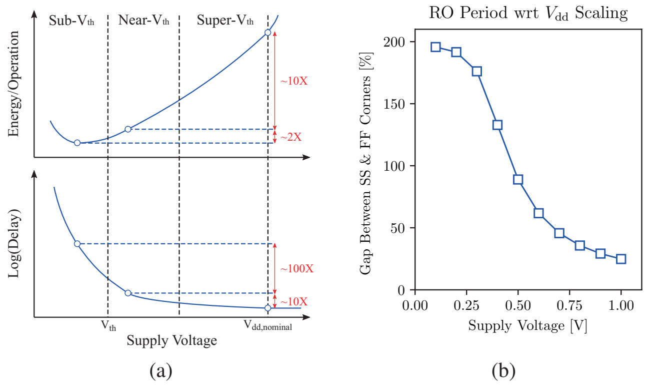

Fig. 1: (a) Energy per operation and delay in different V_{dd} regimes [1]. (b) Impact of V_{dd} scaling on the period of an inverter-based ring oscillator at 28nm process technology.

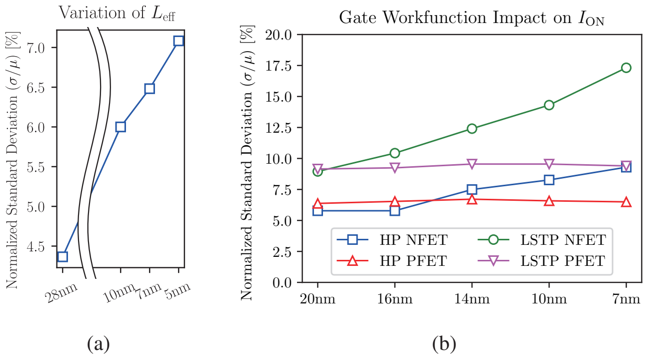

Fig. 2: (a) Variation of L_{eff} at 28nm, 10nm, 7nm, and 5nm process technologies taken from 28nm industry PDK and IRDS 2017 roadmap [2]. (b) Impact of gate workfunction variation on I_{ON} for High Performance (HP) and Low Standby Power (LSTP) version of sub-22nm FinFET PTM-MG models [3].

978-1-7281-4123-7/20/$31.00 © 2020 IEEE

by measuring the same type of ring oscillators (ROs) allocated over a chip, and later, they modeled the maximum delay of a target circuit through SSTA and analyzed its correlation with an arbitrary path [5]. Chan, Gupta, Kahng, and Li [6] observed the sensitivity of CPDs to the variations of PPs forms clusters and suggested to insert one representative RO per each cluster. The biggest advantage of these approaches is that its interpretation is easy enough to be properly exploited for further analysis and optimization. However, a basic premise is that target variables, e.g., CPDs, $f_{\max}$, etc. depend on the variations of PPs *linearly*, which is not true for NTC and advanced process nodes. Severe asymmetry for NTV regime in Fig. 3 clearly shows that the premise is no longer valid.

On the other hand, the approaches with advanced machine learning techniques have emerged recently. The most distinguishing feature is that they use the dataset only consisting of the pairs of HPM measurements and their corresponding $f_{\max}$ with *no intervention of any PP information*. From the dataset of RO frequencies and their corresponding $f_{\max}$ of target circuit, Mu, Chao, Chen, and Wang [7] first selected important parameters using stepwise regression and trained their $f_{\max}$ prediction model with Bayesian linear regression. Sadi, Kannan, Winemberg, and Tehranipoor [8], on the other hand, employed path slack sensors as their HPM and trained a speed binning model by applying various kinds of machine learning techniques. Contrary to the statistical model based approaches, it can be easily applied to the modeling of behaviors for the NTV regime and advanced process technologies since it requires no assumption. However, it is much harder to interpret trained models, and consequently, deeper analysis and optimization on HPM become a daunting task. For instance, Mu, Chao, Chen, and Wang [7] did not consider the impact of the total number of ROs in HPM, their design, allocation, etc., and Sadi, Kannan, Winemberg, and Tehranipoor [8] could not provide enough evidence to support the efficacy of their algorithm theoretically and empirically.

To the best of our knowledge, no work has considered the optimization on HPM taking into account the prediction of target CPDs, under wide and non-linear performance variation, such as the NTV regime and advanced process technologies. In this paper, we propose a highly effective HPM synthesis and CPD prediction methodology for handling such a large and complex variation, with a little amount of pessimism. Precisely, our contributions are twofold, namely:

- formulating the problem of finding an efficient combination of ROs for accurate estimation of PPs related to global variation (GPPs) as a mixed integer second-order cone programming while considering design dependency, and proposing a method of minimizing the total number of ROs under a pessimism level constraint of prediction;
- proposing a prediction flow of CPDs by combining statistical estimation of GPPs and a neural network based CPD prediction model so as to deal with wide and non-linear performance variation, through exploitation of the optimized HPM.

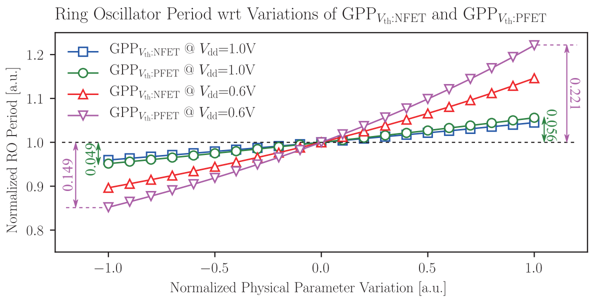

Fig. 3: Change of the period of an inverter-based RO for variation of GPPs concerning threshold voltages at 28nm for N-type and P-type MOSFETs at V_{dd}=1.0V (super-V_{th} regime) and V_{dd}=0.6V (NTV regime). Note that the normalized variations -1.0 and +1.0 mean -3σ and $+3\sigma$ value of a parameter, respectively.

Simulation results demonstrate that our proposed flow outperforms conventional approaches by 6.7∼52.9% and 20.6∼50.8% in terms of the average prediction errors of CPDs and $V_{\mathrm{dd,min}}$, respectively, and tracks ground truth[1] $f_{\max}$ and $V_{\mathrm{dd,min}}$ with little pessimism. We also provide the importance of our optimization on HPM composition by showing that it is able to reduce the average prediction error of $V_{\mathrm{dd,min}}$ by 22.0∼57.1% over the existing methods.

II. Proposed HPM Methodology

A. Overall Flow

Fig. 4 shows the overall flow of our proposed HPM methodology. During a design phase, we construct the following two models:

- *HPM2PP* (Sec. II-B): A model for estimating GPPs from measurements of ROs in HPM;
- *PP2CPD* (Sec. II-E): A model for predicting the variation of CPDs from the estimation of GPPs.

Before constructing an *HPM2PP* model, we solve the problem on the composition of HPM (Sec. II-C) and optimize the total number of ROs in it (Sec. II-D). We handle the local random variation effect on target CPDs separately because it is purely independent of HPM measurements. During a production stage, we executed the following procedures for each die sequentially (Sec. II-F):

1) Estimate GPPs of each die from its HPM measurements using the *HPM2PP* model constructed during the design phase.
2) Infer changes of CPDs from the estimation result of GPPs using the *PP2CPD* model constructed during the design phase.
3) Add margins for local random variation to the inference.
4) Decide $V_{\mathrm{dd,min}}$ of the die from the final prediction result of CPDs.

B. Construction of an HPM2PP Model

The fundamental assumption of our *HPM2PP* model is that HPM measurements can be represented as a linear combination of GPPs [4], [5], [6], i.e.,

$$\widetilde{\mathbf{d}} = \overline{\mathbf{d}} + \Xi^{\mathrm{T}}\mathbf{x} + \boldsymbol{\varepsilon} \tag{1}$$

where $\widetilde{\mathbf{d}}$, $\overline{\mathbf{d}}$, Ξ, $\mathbf{x}$, and $\boldsymbol{\varepsilon}$ denote HPM measurements and their expectations, sensitivities of them with respect to GPPs, changes of GPPs, and errors from local random variation, measurement resolution, etc., respectively. Note that the sensitivity means the change of a measurement divided by the change of a parameter apart from the impact of local random variation. For the validity of linearity, we monitor ROs in HPM with *a sufficiently high level of V_{dd}*.

[1] *Ground truth* means the target we aimed to estimate through prediction in our experiments. For example, in our experiments, we prepared the ground truth CPDs from SPICE simulation on circuit netlists considering variations of GPPs.

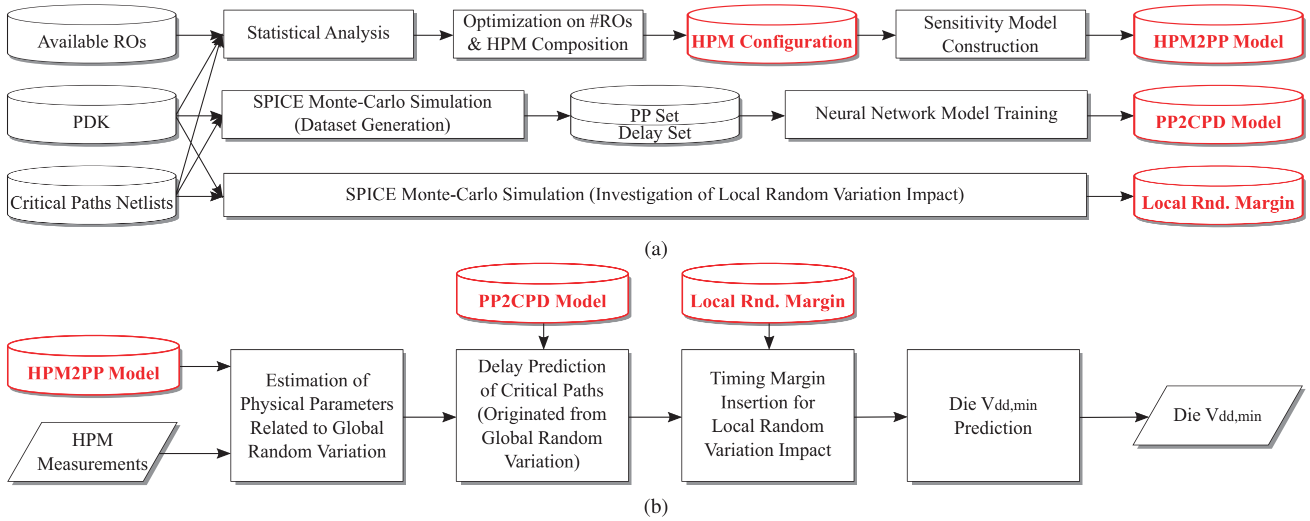
Available ROs
PDK
Critical Paths Netlists
Statistical Analysis
Optimization on #ROs & HPM Composition
HPM Configuration
Sensitivity Model Construction
HPM2PP Model
SPICE Monte-Carlo Simulation (Dataset Generation)
PP Set
Delay Set
Neural Network Model Training
PP2CPD Model
SPICE Monte-Carlo Simulation (Investigation of Local Random Variation Impact)
Local Rnd. Margin
(a)
PP2CPD Model
Local Rnd. Margin
HPM2PP Model
HPM Measurements
Estimation of Physical Parameters Related to Global Random Variation
Delay Prediction of Critical Paths (Originated from Global Random Variation)
Timing Margin Insertion for Local Random Variation Impact
Die Vdd,min Prediction
Die Vdd,min
(b)

Fig. 4: The overall flow of our proposed HPM methodology during (a) a design phase and (b) a production stage.

An *HPM2PP* model is a linear inverse problem of which objective is to find $\mathbf{x}$ from $\widetilde{\mathbf{d}}$, but it is hard to solve accurately because of a limited number of measurements and involvement of purely random components $\boldsymbol{\varepsilon}$. Therefore, it is reasonable to select the most probable one among all solutions of $\mathbf{x}$. We adopt Ridge regression and justify it through Bayesian interpretation [9]. Let us assume that for M GPPs, $\mathbf{x}$ follows a multivariate normal distribution $N\left(\mathbf{0},\Sigma_{\mathbf{x}}\right)$. Then the probability density function (pdf) of a prior distribution for $\mathbf{x}$ is

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^M \det\Sigma_{\mathbf{x}}}} \exp\left[-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\Sigma_{\mathbf{x}}^{-1}\mathbf{x}\right]$$

and for N measurements of a die, the likelihood of the observation $\widetilde{\mathbf{d}}$ is also a normal distribution by Eq. (1), of which pdf is

$$p\left(\widetilde{\mathbf{d}}|\Xi,\mathbf{x}\right) = \frac{1}{\sqrt{(2\pi)^N \det\Sigma_{\boldsymbol{\varepsilon}}}} \times \\ \exp\left[-\frac{1}{2}\left\{\widetilde{\mathbf{d}} - \left(\overline{\mathbf{d}} + \Xi^{\mathrm{T}}\mathbf{x}\right)\right\}^{\mathrm{T}}\Sigma_{\boldsymbol{\varepsilon}}^{-1}\left\{\widetilde{\mathbf{d}} - \left(\overline{\mathbf{d}} + \Xi^{\mathrm{T}}\mathbf{x}\right)\right\}\right]$$

According to Bayes' theorem, the posterior distribution of $\mathbf{x}$ given $\widetilde{\mathbf{d}}$ is proportional to the multiplication of the prior and the likelihood, so it follows $N\left(\overline{\boldsymbol{\mu}}_{\mathbf{x}},\overline{\Sigma}_{\mathbf{x}}\right)$, where $\overline{\boldsymbol{\mu}}_{\mathbf{x}}$ and $\overline{\Sigma}_{\mathbf{x}}$ are

$$\overline{\boldsymbol{\mu}}_{\mathbf{x}} = \overline{\Sigma}_{\mathbf{x}}\Xi\Sigma_{\boldsymbol{\varepsilon}}^{-1}\left(\widetilde{\mathbf{d}} - \overline{\mathbf{d}}\right) \tag{2}$$

$$\overline{\Sigma}_{\mathbf{x}}^{-1} = \Sigma_{\mathbf{x}}^{-1} + \Xi\Sigma_{\boldsymbol{\varepsilon}}^{-1}\Xi^{\mathrm{T}} \tag{3}$$

respectively. Note that $\Sigma_{\boldsymbol{\varepsilon}}$ is a diagonal matrix consisting of $\boldsymbol{\varepsilon}$. Eq. (2) indicates the maximum a posteriori (MAP) estimation of $\mathbf{x}$, which is linearly dependent on $\widetilde{\mathbf{d}}$, and Eq. (3) is related to the estimation uncertainty of $\mathbf{x}$, so it should be taken into account for the pessimistic prediction of GPPs and the consequent changes of CPDs.

From the prior distribution, we can calculate a MAP estimation of variations for each die from HPM measurements using Eq. (2), and taking account of the direction of increase of a CPD to GPPs, i.e., sensitivities of a CPD with respect to GPPs $\mathbf{k}$ and the estimation uncertainty $\overline{\Sigma}_{\mathbf{x}}$ simultaneously, we can find out the pessimistic estimation of the parameters $\mathbf{x}^*$ as

$$\mathbf{x}^* = \overline{\boldsymbol{\mu}}_{\mathbf{x}} + \mathrm{CL}\cdot\mathbf{p} \tag{4}$$

It should be noted that CL denotes a user-defined parameter related to confidence level and $\mathbf{p}$ represents the amount of additional pessimism per unit confidence level, which can be calculated as

$$\mathbf{p} = \overline{\Sigma}_{\mathbf{x}}\mathbf{k}/\sqrt{\mathbf{k}^{\mathrm{T}}\overline{\Sigma}_{\mathbf{x}}\mathbf{k}} \tag{5}$$

For the fast computation of $\mathbf{x}^*$ during a production stage, we prepare $\overline{\Sigma}_{\mathbf{x}}\Xi\Sigma_{\boldsymbol{\varepsilon}}^{-1}$, $\overline{\mathbf{d}}$, and $\mathbf{p}$ in advance through SPICE simulation during a design phase and collect them into our *HPM2PP* model.

C. Optimization on the Composition of HPM

The distance between $\mathbf{x}^*$ and $\overline{\boldsymbol{\mu}}_{\mathbf{x}}$ along $\mathbf{k}$, i.e., $\mathbf{k}^{\mathrm{T}}\left(\mathbf{x}^* - \overline{\boldsymbol{\mu}}_{\mathbf{x}}\right)$, can be exploited as a metric of the amount of pessimism that a given *HPM2PP* model has. Therefore, it could be possible to improve the accuracy of an *HPM2PP* model by selecting the combination of ROs that minimizes it.

Problem (*Finding the Composition of HPM for a Single Direction of Increase of CPDs*)**:** Given T types of ROs, total number of their instances N, sensitivities with respect to GPPs, mean and standard deviation of measurements for each type of RO, and a single direction of increase of CPDs $\mathbf{k}$, find the combination of ROs in HPM which is the most accurate for an *HPM2PP* model, i.e., the combination that minimizes $\mathbf{k}^{\mathrm{T}}\left(\mathbf{x}^* - \overline{\boldsymbol{\mu}}_{\mathbf{x}}\right)$.

The *Problem* is analogous to the optimal experiment design (OED) problem [10], [11], [12], [13], which is to find the best combination of experiments for estimating underlying parameters accurately with the limited number of observations. According to Sagnol and Harman [14], the OED problem

$$\max_{\mathbf{w}\in\mathcal{W}} \quad \Phi_{\mathrm{D|K}}\left(\mathrm{M}(\mathbf{w})\right) = \Phi_{\mathrm{D|K}}\left(\sum_{i=1}^{s} w_i \mathrm{A}_i \mathrm{A}_i^{\mathrm{T}}\right) \tag{6}$$

where $\Phi_{\mathrm{D|K}} : \mathrm{M} \rightarrow \left(\det\ \mathrm{K}^{\mathrm{T}}\mathrm{M}^{-1}\mathrm{K}\right)^{-1/k}$ into a mixed integer second-order cone programming (MISOCP). Using Eq. (4) and Eq. (5), the objective of the *Problem* can be transformed into

$$\min \mathbf{k}^{\mathrm{T}}\left(\mathbf{x}^* - \overline{\boldsymbol{\mu}}_{\mathbf{x}}\right) \Leftrightarrow \min \mathrm{CL}\sqrt{\mathbf{k}^{\mathrm{T}}\left(\Sigma_{\mathbf{x}}^{-1} + \Xi\Sigma_{\boldsymbol{\varepsilon}}^{-1}\Xi^{\mathrm{T}}\right)^{-1}\mathbf{k}} \\ \Leftrightarrow \max\left\{\det\ \mathbf{k}^{\mathrm{T}}\left(\Sigma_{\mathbf{x}}^{-1} + \Xi\Sigma_{\boldsymbol{\varepsilon}}^{-1}\Xi^{\mathrm{T}}\right)^{-1}\mathbf{k}\right\}^{-1}$$

From structural similarity between $\mathrm{M}\left(\mathbf{w}\right) = \sum_{i=1}^{s} w_i \mathrm{A}_i \mathrm{A}_i^{\mathrm{T}}$ in Eq. (6) and $\Sigma_{\mathbf{x}}^{-1} + \Xi\Sigma_{\boldsymbol{\varepsilon}}^{-1}\Xi^{\mathrm{T}} = \mathrm{Q}_{\mathbf{x}}\mathrm{Q}_{\mathbf{x}}^{\mathrm{T}} + \sum_{t=1}^{T} n_t\left(\frac{\boldsymbol{\xi}_t}{\sigma_t}\right)\left(\frac{\boldsymbol{\xi}_t^{\mathrm{T}}}{\sigma_t}\right)$, we can write an MISOCP formulation for the *Problem* by modifying the original one [14]. Meanwhile, upon the observation of Chan, Gupta, Kahng, and Lai [6], we can extend our formulation to optimize the average of the metrics associated with each cluster of CPDs. Thus, the final formulation is

$$
\begin{aligned}
\text{maximize} \quad & \sum_{w=1}^{W} J_w / W & \\
\text{subject to} \quad & \mathrm{Q}_{\mathbf{x}} \mathbf{z}_{0,w} + \sum_{t=1}^{T} \frac{\boldsymbol{\xi}_t}{\sigma_t} z_{t,w} = \mathbf{k}_w J_w, & w \in [W] \\
& \{z_{t,w}\}^2 \le q_{t,w} n_t / (N+1), & t \in [T], w \in [W] \\
& \|\mathbf{z}_{0,w}\|^2 \le q_{0,w} / (N+1), & w \in [W] \\
& q_{0,w} + \sum_{t=1}^{T} q_{t,w} \le J_w, & w \in [W] \\
& q_{t,w} \ge 0, & t \in [T], w \in [W] \\
& q_{0,w} \ge 0, & w \in [W] \\
& \sum_{t=1}^{T} n_t \le N &
\end{aligned}
$$

where W denotes the number of clusters, $[W]$ and $[T]$ represent $\{1, \cdots, W\}$ and $\{1, \cdots, T\}$, respectively, and $\mathbf{k}_w$ is the direction of increase of the w-th CPDs cluster. In the formulation, $\mathbf{z}_{0,w}$, $q_{0,w}$, J_w for $w \in [W]$ and $z_{t,w}$, $q_{t,w}$ for $\{t, w\} \in [T] \times [W]$ are intermediate variables, and n_t for $t \in [T]$ are the target variables we aim to find. Note that n_t means the number of instances of the t-th type RO.

D. Optimization on the Total Number of ROs in HPM

A large number of ROs reduces the uncertainty of predictions at the cost of additional area and measurement overhead. Therefore, engineers should choose the proper number of ROs carefully with consideration of them. To resolve this, we propose a search flow described in Algorithm 1. Our search flow first increases the total number of ROs and finds out the pessimism level of predictions for each case. When L_{crit} consecutive searches fail to achieve relative update r_{crit}, the algorithm stops to solve the MISOCP formulation and increase the number of ROs with the same ratios of ROs in HPM obtained at last. Note that to take into account non-linearity, we run Monte-Carlo simulation using the *PP2CPD* models constructed in Sec. II-E.

E. Construction of a PP2CPD Model

A *PP2CPD* model is used to predict changes of CPDs that correspond to changes of GPPs $\mathbf{x}^*$ obtained from an *HPM2PP* model. To consider non-linearity, we construct a *PP2CPD* model using a neural network [15] which is one of the most representative machine learning techniques today. For the preparation of training and validation datasets, we first generate the set of GPPs and apply them into SPICE simulation of each critical path to capture its delay change. It should be noted that we exclude the impact of local random variation on changes of CPDs in this step since it is completely independent of change caused by GPPs. We also use an exhaustive grid search method to decide hyper-parameters of a *PP2CPD* model, i.e., the number of hidden layers and units in them, a regularization parameter, and stepsize of weights updating, for each critical path. Specifically, we collect average μ and standard deviation σ of prediction errors of CPDs concerning the validation dataset for each hyper-parameter set and select the one whose $-\mu +$ CL$\cdot\sigma$ is smaller than that of the others. We use that value as our margin of inference results obtained from the *PP2CPD* model for guaranteeing pessimistic prediction of CPDs. Note that the time required to generate datasets does not matter since it can proceed while fabricating and characterizing target design.

Algorithm 1 Optimal number of ROs in HPM and its composition

```
In: Target prediction pessimism level $e_{\text{target}}$
    PP2CPD model for each CPD
    Sensitivities and local random variation of each RO
    Initial value of the total number of ROs $N_0$
    Stepsize of RO number increase for new composition search $\Delta N$
    Stopping criterion for new HPM composition search ($L_{\text{crit}}$, $r_{\text{crit}}$)
Out: Optimal number of ROs in HPM and its composition

$N \leftarrow N_0$    // Initialize the total number of ROs in HPM
$L \leftarrow 0$      // Initialize a counter for HPM composition search
$e_0 \leftarrow \epsilon$     // $\epsilon$: Infinitesimal for numerical stability
while True:
    if $L < L_{\text{crit}}$:
        Obtain new HPM composition using MISOCP solver for $N$
    else:
        Increase RO instances with the same ratios of HPM(#ROs=$N_{\text{f}}$)
    Analyze uncertainty of a MAP estimation using Eq. (3)
    Run Monte-Carlo simulation with the PP2CPD models
    Calculate pessimism level of predictions $e$
    if $e \le e_{\text{target}}$:
        Return $N$ and the current HPM composition
    else if $L < L_{\text{crit}}$:
        if $(e_0 - e)/e_0 > r_{\text{crit}}$:
            $L \leftarrow 0$          // Reset the counter
        else:
            $L \leftarrow L + 1$      // Increase the counter
            $N_{\text{f}} \leftarrow N$        // Save the current HPM composition
        $e_0 \leftarrow e + \epsilon$
    if $L < L_{\text{crit}}$:
        $N \leftarrow N + \Delta N$
    else:
        $N \leftarrow N + N_{\text{f}}$
```

F. Procedures During a Production Stage

During a production stage, we first estimate pessimistic values of GPPs for each cluster of CPDs of a die, i.e., $\mathbf{x}^*$ in Eq. (4), from HPM measurements using the *HPM2PP* model constructed in its design phase. After that, starting from the lowest level of V_{dd} among candidates, we increase it until the operation meets a target frequency f_{target}, i.e., $f_{\max} \ge f_{\text{target}}$, to find $V_{\text{dd,min}}$. For calculating $f_{\max}$ for a given V_{dd} level, we infer changes of CPDs originated from GPPs using its *PP2CPD* model and cover their local random variation by adding margins. Then reciprocal of maximum among the predictions is $f_{\max}$ at that level.

III. Experimental Results

To validate our HPM methodology, we used our industry partner's 28nm PDK and DK characterized at 0.6V of V_{dd}. We could not conduct the experiments with an advanced process node due to confidentiality issues, but we expect that its overall trend would be similar to the results. We considered total 19 types of GPPs such as polysilicon gate length defined in the PDK and assumed that 12 types of ROs are available as candidates of our HPM components, by combining a few kinds of cell types, driving strengths, etc. as follows: BUF:1∼3, INV:1∼4, DELAY:1∼2, MUX, NAND2, and NOR2. Sizes of our training and validation datasets for *PP2CPD* model construction were 3,000 and 1,000, respectively, for each critical path, and note that we did not take into account GPPs related to back-end-of-line (BEOL) process, which are remained as our future work.

A. Effectiveness of Our Optimization on HPM Composition

We found the optimized combination of ROs by solving the MISOCP formulation using IBM CPLEX Optimizer [16] through PICOS interface [17] with a total of 50 ROs in HPM for the benchmark circuits listed in Table I, and the results are summarized in Table II. Interestingly, only a few types of ROs, i.e., Buf:3, Inv:3, Mux, and Nor2, accounted for the majority of HPM while five types of ROs, i.e., Buf:1, Inv:1∼2, Inv:4, and Delay:1, were not used at all. Besides, the instance number of ROs were changed remarkably depending on a target design. We also investigated the efficacy of our optimization on prediction of changes of CPDs, $f_{\max}$, and $V_{\mathrm{dd,min}}$ in our methodology using five different combinations: HPMOpt (our optimization result), Random (randomly generated), AllInv:2 (consisting of Inv:2 only), AllNand2 (consisting of Nand2 only), and AllNor2 (consisting of Nor2 only). Our target design was SPARC core of Oracle OpenSPARC T1 processor, and we assumed five candidate levels of V_{dd} between 0.52V and 0.60V. We finished timing closure of the target circuits and then extracted and analyzed 100 setup timing critical paths using Synopsys PrimeTime with Synopsys HSPICE.

We excluded local random variation margins since they are common in all of them, and results are shown in Table III. The first five rows show the statistics on difference between ground truth explained in the introduction and prediction of CPDs, $f_{\max}$, and $V_{\mathrm{dd,min}}$, respectively, for the five combinations, and the column with #Pess represents the number of dies that do not include any optimistic predictions among total 1K dies, so the value divided by 1,000 means yield. For a fair comparison, we set a yield of each combination to 0.999 by adding or subtracting a proper amount of margin. From the table, it is shown that the prediction using our HPMOpt outperforms the others by 24.8∼49.0% and 19.2∼48.5% in terms of average and standard deviation of prediction errors of CPDs, respectively. The accurate prediction of CPDs by our HPMOpt led precise predictions of $f_{\max}$ and $V_{\mathrm{dd,min}}$ as well; specifically, our HPMOpt reduces prediction errors by 21.2∼52.2% of $f_{\max}$ and 22.0∼57.1% of $V_{\mathrm{dd,min}}$ compared to the others. Note that the numbers in the Max. column of $\Delta V_{\mathrm{dd,min}}$ in Table III denote the worst case misprediction of $V_{\mathrm{dd,min}}$ level. For example, the maximum difference between the prediction using our HPMOpt and the ground truth is one level of V_{dd}, while Random predicts two levels higher in the worst case.

TABLE I: The benchmark circuits used in our experiments.

Design	Description	Freq. [MHz]	#Cells
SPARC	Microprocessor core of OpenSPARC T1	282	130,605
aes_cipher	AES cipher (encrypt) block	282	17,012
aes_inv_cipher	AES inverted cipher (decrypt) block	238	23,214
des_perf_opt	Performance-optimized DES block	300	20,702
usb_phy	USB 1.1 PHY	667	510
wb_dma	DMA/Bridge IP Core	500	3,315

B. Effectiveness of Our Proposed CPD Prediction Flow

To verify the effectiveness of our CPD prediction flow, we compared it with other previously published ones and conventional signoff results. Specifically, we compared the quality of prediction with four methods: Statistical, ML-Based, HPM-Average, and Slow-Slow. Statistical is a method that exploits statistical model utilized in [4], [5], [6], and ML-Based is a neural network model with no consideration on GPPs, like [7], [8]. On the other hand, HPM-Average interpolates changes of CPDs using the average of HPM measurements, and Slow-Slow is the signoff results with SS corner. Note that Slow-Slow does not use HPM methodology, so the assigned supply voltages for all dies are the same regardless of an amount of global variation. We applied them to our SPARC core with the same HPM composition used in Sec. III-A, i.e., HPMOpt, in common.

The results are shown in the last four rows of Table III. Our method (the row starts with HPMOpt) achieves a 20.6% pessimism reduction for prediction of $V_{\mathrm{dd,min}}$ on average in comparison with Statistical. In the case of ML-Based, the standard deviation of prediction error itself is slightly smaller than that of ours, but the average prediction error is much larger; consequently, it predicted $V_{\mathrm{dd,min}}$ for each die by 41.4% more pessimistic than ours, on average. HPM-Average produced the poorest result among them due to its excessive simplicity and no consideration of characteristics of each RO type. Compared with conventional signoff results at SS corner, i.e., Slow-Slow, ours recovers 70.1% and 67.5% of pessimism for $f_{\max}$ and $V_{\mathrm{dd,min}}$ on average, respectively. As a result, it is expected to reduce about 6.67% dynamic power consumption on average by lowering V_{dd} additionally with our proposed HPM methodology.

C. Exploration of the Number of ROs and Prediction Quality

For exploring the trade-off between the total number of ROs and prediction quality of CPDs, we applied Algorithm 1 in Sec. II-D. We intentionally set N_0, ΔN, and e_{target} to 1, 1, and 0, respectively, and applied no stopping criterion for a new HPM composition search $(L_{\mathrm{crit}}, r_{\mathrm{crit}})$ to investigate its impact. The results for SPARC core with 100K Monte-Carlo samples are shown in Fig. 5(a). We observed that a sharp decline of CPDs prediction pessimism when the number of ROs is insufficient. For example, 99% quantile value of maximum CPD prediction decreases by 202.1ps and 29.6ps when the total number of ROs increases from 0 to 1 and from 3 to 4, respectively. Therefore, it is important to find out the range of RO numbers carefully in which the pessimism of CPD prediction drops significantly for further exploration.

On the other hand, because of saturation of prediction quality, searching for a new HPM composition has almost no effect in comparison to simply increasing the number of ROs with the same ratios of previously obtained HPM composition when they are sufficient, as shown in Fig. 5(b). Hence, we believe

TABLE II: The optimized composition of HPMs for the benchmark circuits listed in Table I.

Design	Runtime [sec]	#Buf:2	#Buf:3	#Delay:2	#Inv:3	#Mux	#Nand2	#Nor2	#Others	#Total
SPARC	1.54	0	12	1	9	14	4	10	0	50
aes_cipher	1.83	2	19	2	0	21	2	4	0	50
aes_inv_cipher	1.88	3	18	2	0	19	2	6	0	50
des_perf_opt	2.30	0	11	1	10	13	5	10	0	50
usb_phy	1.24	0	10	2	11	12	5	10	0	50
wb_dma	1.79	0	5	2	17	9	6	11	0	50

TABLE III: Statistics on the difference between ground truth and prediction of CPDs, $f_{\max}$, and $V_{\mathrm{dd,min}}$ for 1,000 dies. All values in parentheses are normalized to the results of our HPMOpt, and in bold is the best result for each criterion.

Method	#Pess	ΔCPDs [ps] ([a.u.])		$\Delta f_{\max}$ [MHz] ([a.u.])		$\Delta V_{\mathrm{dd,min}}$ [mV] ([a.u.])	
		Avg.	Std.	Avg.	Std.	Avg.	Max.
HPMOpt	999	**175.584 (1.000)**	51.812 (1.000)	**41.216 (1.000)**	**15.979 (1.000)**	**9.700 (1.000)**	**20 (1)**
Random	999	236.140 (1.330)	64.104 (1.237)	52.322 (1.269)	17.607 (1.102)	12.440 (1.282)	40 (2)
AllInv:2	999	342.822 (1.930)	94.831 (1.830)	83.485 (2.026)	28.796 (1.802)	21.860 (2.254)	40 (2)
AllNand2	999	348.273 (1.961)	100.612 (1.942)	81.146 (1.969)	29.162 (1.825)	21.020 (2.167)	40 (2)
AllNor2	999	334.404 (1.883)	91.838 (1.773)	86.313 (2.094)	29.562 (1.850)	22.620 (2.332)	40 (2)
Statistical ([4], [5], [6])	999	190.346 (1.072)	55.602 (1.073)	52.126 (1.265)	17.239 (1.079)	12.220 (1.260)	40 (2)
ML-Based ([7], [8])	999	222.058 (1.250)	**50.743 (0.979)**	53.186 (1.387)	16.928 (1.059)	13.720 (1.414)	40 (2)
HPM-Average	999	376.734 (2.121)	96.472 (1.862)	78.370 (1.901)	27.717 (1.735)	19.700 (2.031)	40 (2)
Slow-Slow	1,000	583.091 (3.283)	148.875 (2.796)	137.975 (3.348)	44.733 (2.799)	29.840 (3.077)	40 (2)

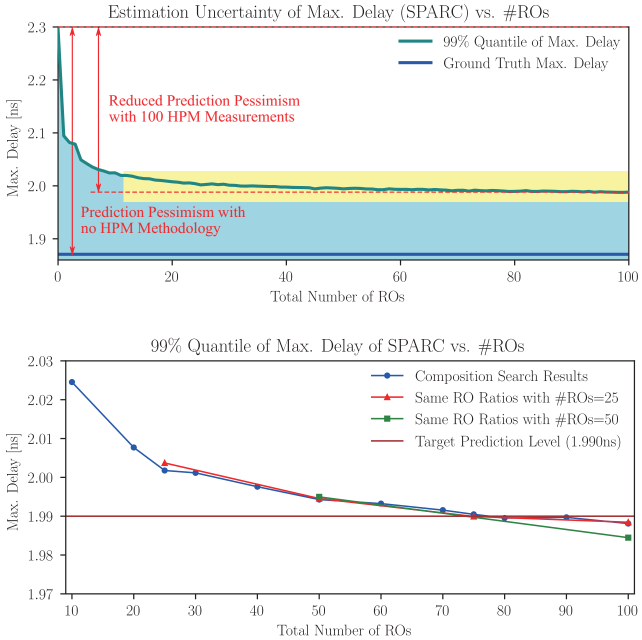

Fig. 5: (Upper) Changes of maximum delay estimation for SPARC core. The blue line represents ground truth of maximum delay we assumed, i.e., maximum CPD at a typical corner in our experiment, and the light blue region denotes the distribution of delays. (Lower) Changes of 99% quantile value of maximum delay estimation for a few kinds of RO combinations. Note that the figure is an enlargement of the yellow region in the upper plot.

that our stopping strategy for a new HPM composition search in Algorithm 1 will save excessive computation time for that case. In the case of SPARC core, for example, it would be efficient to find out the combination of ROs using the MISOCP formulation when their number is 25, and starting from this, increase the RO numbers in HPM by 25, 50, 75, etc., until target prediction pessimism level e_{target} is met. If e_{target} is 1.99ns, the total number of ROs would be 75 at last, as shown in Fig. 5(b).

IV. Conclusion

In this paper, we proposed a highly effective HPM methodology for accurate CPD prediction. Precisely, we formulated the optimization problem of selecting an efficient combination of ROs in HPM for accurate estimation of GPPs as an MISOCP formulation and proposed a method of minimizing the total number of ROs under a pessimism level constraint of prediction. Then we proposed a prediction flow of CPDs by combining a statistical estimation of GPPs and a neural network based CPDs prediction model to deal with wide and non-linear performance variation. From the experiments using a 28nm industry PDK and 0.6V operation, we validated the efficacy of our HPM methodology. We are currently extending our methodology to consider the variation of parasitic resistance and capacitance in back-end-of-line (BEOL) process. We are also investigating and researching the adaptation of the proposed prediction flow to actual silicon measurements beyond PDK-based model.

Acknowledgements

This research was supported by the Samsung-Seoul National University Collaboration project (Foundry-201710DD027, 201812DD005F) and the BK21 Plus Project in 2019.

References

[1] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[2] "International roadmap for device and systems." https://irds.ieee.org, 2017.

[3] C. Meinhardt, A. L. Zimpeck, and R. A. L. Reis, "Predictive evaluation of electrical characteristics of sub-22 nm finfet technologies under device geometry variations," *Microelectronics Reliability*, vol. 54, no. 9-10, pp. 2319–2324, 2014.

[4] Q. Liu and S. S. Sapatnekar, "A framework for scalable postsilicon statistical delay prediction under process variations," *IEEE TCAD*, vol. 28, no. 8, pp. 1201–1212, 2009.

[5] Q. Liu and S. S. Sapatnekar, "Capturing post-silicon variations using a representative critical path," *IEEE TCAD*, vol. 29, no. 2, pp. 211–222, 2010.

[6] T.-B. Chan, P. Gupta, A. B. Kahng, and L. Lai, "Synthesis and analysis of design-dependent ring oscillator (ddro) performance monitors," *IEEE TVLSI*, vol. 22, no. 10, pp. 2117–2130, 2014.

[7] S.-P. Mu, M. C.-T. Chao, S.-H. Chen, and Y.-M. Wang, "Statistical framework and built-in self-speed-binning system for speed binning using on-chip ring oscillators," *IEEE TVLSI*, vol. 24, no. 5, pp. 1675–1687, 2016.

[8] M. Sadi, S. Kannan, L. Winemberg, and M. Tehranipoor, "Soc speed binning using machine learning and on-chip slack sensors," *IEEE TCAD*, vol. 36, no. 5, pp. 842–854, 2017.

[9] J. Chung and J. Kim, "Segment delay learning from quantized path delay measurements," *IEEE TCAD*, vol. 34, no. 6, pp. 1038–1042, 2015.

[10] V. V. Fedorov, *Theory of optimal experiments*. Academic Press, New York, 1972.

[11] H. P. Wynn, "Results in the theory and construction of *d*-optimum experimental designs," *J. Royal Stat. Soc., Series B (Methodological)*, vol. 34, no. 2, pp. 133–147, 1972.

[12] R. Harman and E. Benková, "Barycentric algorithm for computing d-optimal size- and cost-constrained designs of experiments," *Metrika*, vol. 80, no. 2, pp. 201–225, 2017.

[13] Y. Yu, "D-optimal designs via a cocktail algorithm," *Stat. Comput.*, vol. 21, no. 4, pp. 475–481, 2011.

[14] G. Sagnol and R. Harman, "Computing exact *d*-optimal designs by mixed integer second-order cone programming," *Ann. Statist.*, vol. 43, no. 5, pp. 2198–2224, 2015.

[15] M. van Gerven and S. Bohte, *Artificial neural networks as models of neural information processing*. Frontiers Media SA, 2018.

[16] "Ibm ilog cplex optimizer 12.8.0." https://www.ibm.com/analytics/cplex-optimizer, 2017.

[17] G. Sagnol, "Picos, a python interface to conic optimization solvers," tech. rep., Technical Report 12-48, ZIB, 2012. http://picos.zib.de.