Delay Sensitivity Polynomials Based Design-Dependent Performance Monitors for Wide Operating Ranges

Ruikai Shi
State Key Laboratory of Computer Architecture, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
shiruikai@loongson.cn

Liang Yang
Loongson Technology Co., Ltd
Beijing, China
yangliang@loongson.cn

Hao Wang
Loongson Technology Co., Ltd
Beijing, China
wanghao@loongson.cn

***Abstract*—The downsizing of CMOS technology makes circuit performance more sensitive to on-chip parameter variations. Previous proposed design-dependent ring oscillator (DDRO) method provides an efficient way to monitor circuit performance at runtime. However, the linear delay sensitivity expression may be inadequate, especially in a wide range of operating conditions. To overcome it, a new design-dependent performance monitor (DDPM) method is proposed in this work, which formulates the delay sensitivity as high-order polynomials, makes it possible to accurately track the nonlinear timing behavior for wide operating ranges. A 28nm technology is used for design evaluation, and quite a low error rate is achieved in circuit performance monitoring comparison.**

Keywords—performance monitoring, path delay modeling, frequency estimation, wide operating ranges

I. Introduction

With the development of integrated circuit (IC) manufacturing technology, the influence of process, voltage and temperature (PVT) variations on timing is becoming crucial. As a result, design based on worst-case conditions may not guarantee the post-silicon validity of circuits and it is hard to track real-time performance fluctuations. A conservative guard band is usually added in the design phase to deal with it. Unfortunately, it may affect performance, power and area, resulting in enormous design and test costs. Furthermore, maximum performance or power savings might not be achieved for real chips in various operating conditions. In order to recover design margin, reduce test time and provide reference for energy management systems, it has become important to provide on-chip monitoring [1][2].

The existing performance monitors can be classified into two categories: generic and design-dependent timing monitors. Design-dependent monitors show superiority in comparison with other performance monitors on three dimensions (design costs, the number of monitors and the post processing time) [3]. Representative critical path (RCP) [4] is a typical design-dependent monitor that overcomes the drawbacks of generic monitors in the correlation with targeted circuits. However, it is impossible to identify a single critical path to cover all corners in different parameter conditions [5]. An effective approach called design-dependent ring oscillator (DDRO) [6] has been employed to improve it. Highly correlated monitors are synthesized for critical path clusters, which are formed in accord with delay sensitivities to several variation sources. In the pursuit of performance and power management, novel commercial ICs require a wide range of operating conditions, such as supply voltage, temperature and aging [7]. A nonlinear relationship between path delay and supply voltage is demonstrated and the performance is limited by different paths at different voltage points in [8]. The linear delay variation models used in RCP and DDRO are devoted to capture small fluctuations near some specific parameters, but might have inaccurate predictions in large scope varieties due to the nonlinear essence on timing behavior.

In this work, a new high-order polynomial model for path delay sensitivity is introduced for better prediction in wide operating ranges, and a design-dependent performance monitor (DDPM) method is proposed accordingly. Our experiments show nearly 1.5% error rate is achieved which is a significant improvement compared to previous works.

II. DDPM Framework

Our DDPM Framework is originated from DDRO methodology [6] but with some key steps improved. Fig. 1 gives the framework of DDPM composed by four major steps.

First of all, the critical paths are extracted by static timing analysis (STA), and their delay sensitivities to variation sources are characterized in Step 1. Next, the paths are grouped into multiple clusters according to their different delay sensitivities in Step 2. After that, typical ROs are synthesized to match the delay sensitivity of each cluster as close as possible in Step 3. Finally, ROs delay are measured at manufacturing and/or runtime, and the actual circuit performance can be predicted based on them in Step 4. The main differences between DDRO and DDPM are as follows:

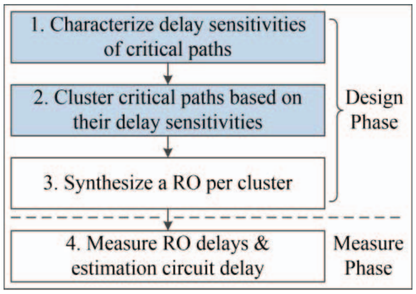
1. Characterize delay sensitivities of critical paths
2. Cluster critical paths based on their delay sensitivities
3. Synthesize a RO per cluster
Design Phase
4. Measure RO delays & estimation circuit delay
Measure Phase

Fig. 1. Proposed DDPM framework. And the steps marked with colored boxes are of the main concerns in this work.

1) In DDRO method the variation model in [9] is used, and the path delay is represented as a linear combination of the effects of variation sources, so called delay sensitivities, using the first-order coefficients. But in DDPM method the high-order polynomials are adopted to represent delay sensitivities, and a more complex representation for similarity is illustrated at the same time. This will be detailed in Section III.

2) In DDRO method, *k-means*++ algorithm [10] is used to cluster the paths, while in DDPM a corresponding clustering algorithm is proposed to adapt the polynomial characterization of delay sensitivity model. This will be given in Section IV.

In fact, the last two steps stay the same as in DDRO. The main focus is to optimize the delay sensitivity model in DDPM.

III. Delay Sensitivity Polynomials Formulation

A. Delay Sensitivity

For better understandings, all the notations in this work are defined in Table I beforehand. As mentioned before, it is not accurate for linear delay sensitivities expression when the range of variations expanding. Therefore, the fluctuation of path delay is represented as high-order polynomials using the Taylor formula, which is shown as

$$\begin{aligned} d_{path,i} &= d_{nom\_path,i}\left(1+PLN_{path,i}\left(\mathbf{G}\right)+l_{path,i}\right) \\ PLN_{path,i}\left(\mathbf{G}\right) &= \sum_{j=1}^{M}\sum_{r=1}^{n_j} a_{i,j,r}\cdot g_j^{\,r} \end{aligned} \tag{1}$$

Where $d_{path,i}$, $d_{nom\_path,i}$, $\mathbf{G}$, $l_{path,i}$, $a_{i,j,r}$ and g_j follow the notations in Table I. Note that the influence of $l_{path,i}$ might be a problem for a single device, but is limited due to the alleviation of a large number of devices in critical paths. Then the delay sensitivity vector of path i is obtained by calculating the first-order derivative of *PLN* function for each variation source:

$$\mathbf{V}_{path,i} = \left[\sum_{r=1}^{n_1} r\cdot a_{i,1,r}\cdot g_1^{\,r\text{-}1} \quad \cdots \quad \sum_{r=1}^{n_M} r\cdot a_{i,M,r}\cdot g_M^{\,r\text{-}1}\right] \tag{2}$$

Accordingly, the same delay model as in (1) is used to represent the delay of ROs as follows:

$$\begin{aligned} d_{ro,x} &= d_{nom\_ro,x}\left(1+PLN_{ro,x}\left(\mathbf{G}\right)\right) \\ PLN_{ro,x}\left(\mathbf{G}\right) &= \sum_{j=1}^{M}\sum_{r=1}^{n_j} b_{x,j,r}\cdot g_j^{\,r} \end{aligned} \tag{3}$$

Where $d_{ro,x}$, $d_{nom\_ro,x}$ and $b_{x,j,r}$ follow the notations in Table I. It is observed that local variation is omitted in delay model of RO. Each RO consists of many identical gates whose local variations are insignificant due to averaging of uncorrelated delay deviation [6]. It is also similar to give the expression of delay sensitivity vector of RO x recorded as $\mathbf{V}_{ro,x}$:

$$\mathbf{V}_{ro,x} = \left[\sum_{r=1}^{n_1} r\cdot b_{x,1,r}\cdot g_1^{\,r\text{-}1} \quad \cdots \quad \sum_{r=1}^{n_M} r\cdot b_{x,M,r}\cdot g_M^{\,r\text{-}1}\right] \tag{4}$$

B. Similarity of Sensitivities

By expressing the path delay in polynomial form, each component of the delay sensitivity vector is a multi-order polynomial. And it is difficult to characterize their difference by Euclidean distance. Therefore, the concepts of function distance and neighborhood are introduced as metrics to measure the similarity of delay sensitivities. Extract an arbitrary element from the delay sensitivity vector and record it as a function of g_j on its change interval C_j, i.e.

TABLE I. Glossary of Terminology

Term	Description
i	Index for critical path
j	Index for variation source
x	Index for cluster/RO
r	Index for power of polynomial
N	Total number of critical paths
M	Total number of variation sources
K	Total number of clusters/ROs
$d_{path,i}$	Delay of path i, i =1,2,…,N
$d_{ro,x}$	Delay of RO x, x=1,2,…,K
$d_{nom\_path,i}$	Nominal delay of path i
$d_{nom\_ro,x}$	Nominal delay of RO x
$\mathbf{V}_{path,i}$	Delay sensitivity of path i to all M variation sources
$\mathbf{V}_{ro,x}$	Delay sensitivity of RO x to all M variation sources
$\mathbf{G}$	Global variation vector with all M variation sources
g_j	The j^{th} variation source component of $\mathbf{G}$
$l_{path,i}$	Local variation of path i
n_j	Highest power of polynomials to variation source j
$a_{i,j,r}$	polynomial coefficient to path i, variation source j and power r
$b_{x,j,r}$	polynomial coefficient to RO x, variation source j and power r
C_j	Operating range of variation source j

$$v_i\left(g_j\right) = \sum_{r=1}^{n_j} r\cdot a_{i,j,r}\cdot g_j^{\,r\text{-}1}, g_j\in C_j \tag{5}$$

Here $v_i(g_j)$ has at most n_j – 1 order nonzero continuous derivatives on C_j, so the m-stage distance (m = 0, 1, …, n_j – 1) between any two $v_{i_1}(g_j)$ and $v_{i_2}(g_j)$ on C_j can be calculated using the equation as follows:

$$dist_m\left[v_{i_1}\left(g_j\right), v_{i_2}\left(g_j\right)\right] = \max_{0\le r\le m}\max_{g_j\in C_j}\left|v_{i_1}^{(r)}\left(g_j\right)-v_{i_2}^{(r)}\left(g_j\right)\right| \tag{6}$$

It is obvious that the following inequality holds:

$$\begin{aligned} & dist_0\left[v_{i_1}\left(g_j\right), v_{i_2}\left(g_j\right)\right] \le dist_1\left[v_{i_1}\left(g_j\right), v_{i_2}\left(g_j\right)\right] \le \cdots \\ & \le dist_{n_j-1}\left[v_{i_1}\left(g_j\right), v_{i_2}\left(g_j\right)\right] \end{aligned} \tag{7}$$

The inequality illustrates that the (n_j – 1)-stage distance gives the upper bound of the arbitrary stage distance, so it is reasonable to quantify the similarity of delay sensitivities by their (n_j – 1)-stage distance. Given a positive number δ, the (n_j – 1)-stage δ neighborhood of $v_i(g_j)$ on C_j is defined as the set of $v(g_j)$ whose (n_j – 1)-stage distance to $v_i(g_j)$ is less than δ:

$$nb_{n_j-1}\left[\delta, v_i\left(g_j\right)\right] = \left\{v\left(g_j\right)\middle| dist_{n_j-1}\left[v\left(g_j\right), v_i\left(g_j\right)\right] < \delta\right\} \tag{8}$$

Here $v\left(g_j\right)\in nb_{n_j-1}\left[\delta, v_i\left(g_j\right)\right]$ is called to have the (n_j – 1)-stage δ proximity with $v_i(g_j)$, and δ is the neighbor radius. Accordingly, find the path neighbors of path i using the following equation:

$$PNB[\delta,i]=\left\{i'\neq i\,\middle|\,v_{i'}\left(g_j\right)\in nb_{n_j-1}\left[\delta,v_i\left(g_j\right)\right]\right\} \tag{9}$$

Finally, extending the discussion to a general case, the path neighbors with similar delay sensitivity to path *i* are defined as

$$PNB[\Delta,i]=\bigcap_{j=1}^{M} PNB\left[\delta_j,i\right] \tag{10}$$

Where $\mathbf{\Delta} = [\delta_1, \delta_2, \ldots, \delta_M]$ is a user-defined proximity vector. Thus the similarity between any two paths can be evaluated based on the function distance of their delay sensitivities. Find a set of paths with similar delay sensitivities to a target path using (10).

IV. The Clustering Process

A. Clustering Algorithm

The *k-means* algorithm is no longer suitable for DDPM. Because the delay sensitivities are now presented in the form of higher-order curves, it is difficult to select an appropriate *K* by mapping method. In order to adapt the new characterization of delay sensitivity, Algorithm 1 has been proposed to ensure clustering results within expectations.

Before the clustering process begins, several input data need to be prepared. It is recommended to perform STA on the original design and get the initial critical paths set CPS_0. After that, perform SPICE simulation on CPS_0 with PVT variation sources, and run polynomial curve fitting to find *PLN* function for each path. Finally compute all the $\mathbf{V}_{path,i}$ using (2). The proximity limit vector $\mathbf{\Delta}$ is given according to target number of clusters, which will be further discussed in subsection B.

In the initialization phase, singular paths whose path neighborhood is empty set are identified and screened out if existed. After that, greedy strategy is adopted to generate the primary clusters for non-singular paths. The key feature is to screen out all the singularities which makes the following clustering iteration stable and convergence rapidly.

In the iteration phase, all paths are redistributed to the nearest cluster based on their sensitivity distance to each centroid, and then centroids of changed clusters get updated accordingly. The optimization continues until convergence conditions get satisfied. The condition for termination should be carefully determined for better convergence. Here a non-negative threshold vector $\mathbf{\Gamma}$ is recommended to end the iteration process, e.g. 1% of $\mathbf{\Delta}$.

In fact, the iterative termination of our algorithm remains the same with *k-means* algorithm, whose time and space complexity are both O(n). But the initialization takes more time especially for *PNB* calculation, which approaches O(n^2). The speed of convergence depends on the constraints of centroid movement, which is user-defined.

B. Discussion on *Δ* and *K*

In Algorithm 1, the granularity of clustering is adjusted by the proximity limit vector $\mathbf{\Delta}$, not as the *K* parameter as in [6]. A relationship between $\mathbf{\Delta}$ and *K* exists intuitively as *K* should decrease when $\mathbf{\Delta}$ increases. However, it seems difficult to get a simple expression due to not only the high-order polynomial form, but also the dispersion of delay sensitivities for specific circuit paths. To get a glimpse of it, a simplified experiment is set up, assuming supply voltage is the only one variation source and cubic polynomials are used for delay expression. Fig. 2 illustrates a roughly exponential relationship between $\mathbf{\Delta}$ and *K*, which means a reasonable *K* value could be obtained when $\mathbf{\Delta}$ is carefully chosen.

```
Algorithm 1
Input: delay sensitivities vector $\mathbf{V}_{path,i}$, proximity limit vector $\mathbf{\Delta}$
Output: cluster result $CL[K]$
Initialization:
  1. $CPS = CPS_0$.
  2. Calculate $PNB[\mathbf{\Delta},i]$ for all $i \in CPS$ using (10).
  3. Screen out all $x_0$ singularities:
     $SG = \{i \mid PNB[\mathbf{\Delta},i] = \varnothing,\ i \in CPS\}$, $x_0 = \mathrm{card}(SG)$.
     for ($x = 1$ to $x_0$)
        $CL[x] = SG(x)$.
     end for
     $CPS = CPS - SG$, $x = x_0 + 1$.
  4. Generate primary clusters:
     while (existing path haven't been clustered)
        Find one $i_{max}$ that $\mathrm{card}(PNB[\mathbf{\Delta},i_{max}]) \geq \mathrm{card}(PNB[\mathbf{\Delta},i])$, $i \in CPS$.
        $CL[x] = \{i_{max}\} \cup PNB[\mathbf{\Delta},i_{max}]$.
        $CPS = CPS - \{i_{max}\} \cup PNB[\mathbf{\Delta},i_{max}]$.
        Recalculate $PNB[\mathbf{\Delta},i]$ for all $i \in CPS$ using (10).
        $x = x + 1$.
     end while
  5. $K = x$, calculate the centroid of each cluster from 1 to $K$.
Optimization iteration:
  6. $CPS = CPS_0$.
  7. Redistribute clusters:
      do
        Reset $CL[K]$.
        for ($i = 1$ to $N$)
           Find the nearest cluster centroid $x_{closest}$.
           $CL[x_{closest}] = CL[x_{closest}] \cup \{i\}$.
        end for
        Recalculate the centroid of each cluster from 1 to $K$.
      while (the centroid changes > $\mathbf{\Gamma}$)
  8: Return $CL[K]$.
```

V. Experimental Results

A 28nm technology is used for evaluation. Totally 1792 critical paths have been extracted from a commercial microprocessor by STA in multiple corners, whose worst timing slack is within 5% of the clock period. For simplicity, only supply voltage in wide range (from 0.7V to 1.45V with normal value 1.1V here) is chosen as variation source, but it can be easily extended to multiple variation sources following the general form presented in Section III.

Table II gives error rates for *PLN* functions fitting with different polynomial orders. The increase of order leads to more runtime due to complicated calculations for function distance and neighbor. The lifting efficiency is defined as the improvement of accuracy (relative to order 2) divided by runtime. As a trade-off between accuracy and computation time, the third-order polynomial with best lifting efficiency is selected to continue the following experiments. Then delay sensitivities are calculated and critical paths are clustered using Algorithm 1 under different values of proximity parameter $\mathbf{\Delta}$. With the constraint that cluster number should not be more than 10 and 20, two parameters $\mathbf{\Delta}$ = 0.046 * $dist_{max}$ and $\mathbf{\Delta}$ = 0.073 * $dist_{max}$ are selected in relative stable intervals in Fig. 2.

TABLE II. POLYNOMIAL APPROXIMATION FOR PATH DELAY

Order	Max Error (%)	Lifting Efficiency	Mean Error (%)	Lifting Efficiency
2	8.32	N/A	3.12	N/A
3	**2.44**	77.75	**0.98**	75.71
4	1.49	45.10	0.27	43.47
5	0.92	33.73	0.07	32.40
6	0.79	25.25	0.02	24.25

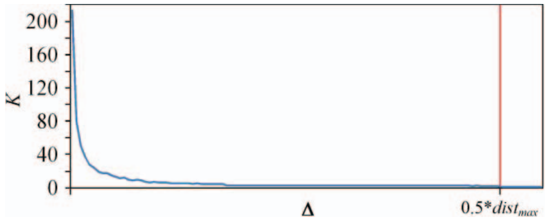

Fig. 2. An approximately exponential relationship between **Δ** and *K*. Where $dist_{max}$ is the maximum function distance between all $\mathbf{V}_{path,i}$.

For comparison, two related references have been realized. The first is REF1 directly from DDRO method [6], where the delay sensitivities are extracted with small variation near the nominal value (1.1V±50mV here) and then expanded for the scope of evaluation. The second is REF2 extended from DDRO method by segmented interception, where the large sliding interval is divided into two segments (from 0.7V to 1.1V with nominal value 0.9V and from 1.1V to 1.45V with nominal value 1.25V, respectively). Then RO synthesis and timing evaluation perform in each interval individually.

The calculations of estimation error rates for circuit delay in Table III show that our method achieves the best result compared to the references. The error rate of our method approaches less than 4% in maximum and nearly 1.5% on average. While REF2 improves the accuracy of REF1 to some degree, it comes at a significant design costs, e.g. the number of ROs increasing. Fig. 3 gives simulation and estimation results on circuit delays from three methods using 17 ROs (doubled for REF2). It can be observed that estimations using linear delay sensitivity keep accurate in small range, but the accuracy decreases dramatically as the variation range expands. It is proved that the proposed delay sensitivity polynomials model improves DDRO approach significantly in wide operating ranges.

VI. CONCLUSIONS

In this work, a novel DDPM method is proposed to support accurate circuit performance prediction in wide operating ranges. To distinguish from previous works, the path delay sensitivity in DDPM is formulated as high-order polynomials and its proximity function is further discussed. In addition, a corresponding clustering algorithm is provided to adapt to the high-order model. A 28nm simulation results demonstrate the average and maximum error rate are 1.53% and 3.92% respectively for performance monitoring.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant No. 61432016 and No. 61521092, and in part by the Key Program of the Chinese Academy of Sciences (ZDRW-XH-2017-1).

TABLE III. ESTIMATION ERROR RATE COMPARISON

Method	Δ	*K*	Max Error (%)	Mean Error (%)
This work	0.073 * $dist_{max}$	9	3.99	1.61
	0.046 * $dist_{max}$	17	3.92	1.53
REF1	N/A	9	34.18	9.24
	N/A	17	33.59	9.01
REF2	N/A	9*2	17.64	3.64
	N/A	17*2	17.24	3.44

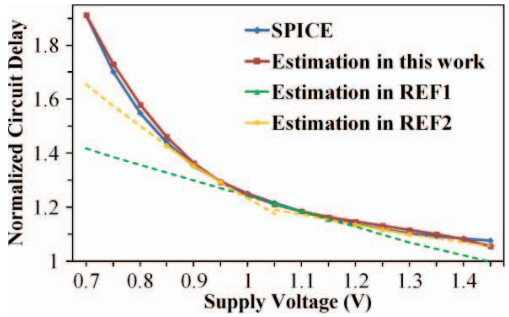

Fig. 3. Simulation and estimation results for comparison. Note that the linear delay sensitivity model used in REF1/2 makes their actual RO delay curves in wide voltage intervals cannot be predicted and are completely out of control at the synthesis step. Thus the estimations out of range are marked with the dotted lines for intuitive illustration.

REFERENCES

[1] A. J. Drake, R. M. Senger, H. Singh, G. D. Carpenter and N. K. James, "Dynamic measurement of critical-path timing", in *IEEE International Conference on Integrated Circuit Design and Technology (ICICDT)*, 2008, pp. 249-252.

[2] J. Tschanz, K. Bowman, S. Walstra, M. Agostinelli, T. Karnik and V. De, "Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance", in *IEEE Symposium on VLSI Circuits*, 2009, pp. 112-113.

[3] J. K. Rangan, N. P. Aryan, L. Wang, J Bargfrede, C. Funke and H. Graeb, "Design-dependent monitors based on delay sensitivity tracking", in *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2018, pp. 633-66.

[4] Q. Liu and S. S. Sapatnekar, "Capturing post-silicon variations using a representative critical path", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 29, no. 2, pp. 211-222, 2010.

[5] M. Zandrahimi, Z. Al-Ars, P. Debaud and A. Castillejo, "Challenges of using on-chip performance monitors for process and environmental variation compensation", in *IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016, pp. 1018-1019.

[6] T. B. Chan, P. Gupta, A. B. Kahng and L. Lai, "Synthesis and analysis of design-dependent ring oscillator (DDRO) performance monitors", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2117-2130, 2013.

[7] S. R. Vangal, S. Jain and V. De, "A solar-powered 280mV-to-1.2V wide-operating-range IA-32 processor", in *IEEE International Conference on IC Design & Technology (ICICDT)*, 2014, pp. 1-4.

[8] J. Kim, K. Choi, Y. Kim, W. Kim, K. Do and J. Choi, "Delay monitoring system with multiple generic monitors for wide voltage range operation", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 1, pp. 37-48, 2018.

[9] L. Cheng, P. Gupta, K. Qian, C. Spanos and L. He, "Physically justifiable die-level modeling of spatial variation in view of systematic across wafer variability", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 30, no. 3, pp. 388-401, 2011.

[10] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding", in *ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 1027-1035.