# CODA: A Concurrent Online Delay Measurement Architecture for Critical Paths

Yubin Zhang[†‡], Haile Yu[†] and Qiang Xu[†‡]
[†]CUhk REliable Computing Laboratory (CURE)
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
[‡]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Email: {ybzhang,hlyu,qxu}@cse.cuhk.edu.hk

## ABSTRACT

*With technology scaling, integrated circuits behave more unpredictably due to process variation, environmental changes and aging effects. Various variation-aware and adaptive design methodologies have been proposed to tackle this problem. Clearly, more effective solutions can be obtained if we are able to collect real-time information such as the actual propagation delay of critical paths when the circuit is running in normal function mode. Motivated by the above, in this paper, we propose a novel concurrent online delay measurement architecture for critical paths, namely* CODA*, to facilitate this task. Experimental results demonstrate high accuracy and practicality of the proposed technique.*

## I. Introduction

With technology scaling, process, voltage and temperature variations have a high impact on the timing behavior of integrated circuits (ICs), and hence it is increasingly difficult to ensure ICs' timing correctness solely by off-line manufacturing test. What's worse, circuits fabricated with latest technology suffer from ever-increasing aging effects (e.g., negative bias temperature instability (NBTI)), gradually reducing their performance [3, 11]. While there have been some attempts to conduct on-chip delay measurement to tackle the above problems (e.g., [9, 13, 14]), they require a dedicated test mode. Such non-concurrent solutions are hence inherently inaccurate due to the discrepancy between circuits' timing behavior in functional mode and that in test mode.

The ever-increasing non-predictability of IC performance has also motivated a number of research efforts in variation-aware and adaptive design methodologies. Various types of sensors (e.g., ring oscillator for process variation characterization [7, 8] and aging sensor [1]) are introduced on-chip to characterize the circuits' timing behavior, which can then be used for post-silicon tuning. Although helpful, these sensors can only provide some rough timing estimation without directly measuring critical paths in function mode.

Suppose the propagation delay of critical paths can be acquired with high accuracy as the circuit is working in function mode, such on-site information will be of great help for process variation characterization, dynamic management policy design and aging monitoring. For example, in dynamic voltage scaling (DVS), reducing supply voltage too much may result in prolonged delay outside of timing constraint while leaving a large margin wastes the timing slack for energy savings. If real-time path delay can be obtained, we are able to set proper values for the parameters of various dynamic management policies.

The above motivates us to develop a concurrent online delay measurement architecture, namely CODA, which is able to accurately measure the path delay while the circuit is working in normal function mode. As far as we know, this is the first accurate online delay measurement architecture. The innovative CODA holds the following outstanding benefits:

- High accuracy of measurement can be achieved by eliminating the interference from process variation and routing uncertainty.
- CODA measures the actual path delay as the circuit is working, without the need of switching to specific test mode.
- CODA completes delay measurement in a few clock cycles, which enables concurrent response corresponding to the real time changes of circuit status.
- The equipment of CODA requires acceptable hardware overhead and introduces negligible interference to the circuits' normal working, enabling CODA with high practicality.
- CODA is beneficial to a variety of tasks, including dynamic scaling, aging monitoring, speed binning, process variation characterization, etc. CODA itself can work for so multiple purposes that there is no need to equip the dedicated devices for each purpose.

The remainder of this paper is organized as follows. Section II reviews related work and motivates this paper. CODA is overviewed in Section III while its detailed design is presented in Section IV. Section V presents the experimental results and, finally, Section VI concludes this paper.

## II. Preliminaries and Motivations

Traditionally, the timing correctness of ICs is guaranteed by manufacturing test with external automatic test equipment (ATE). With technology scaling, however, this task has become increasingly difficult because: (i). it is quite difficult and costly to generate and apply delay tests for a large number of critical paths; (ii). ATE itself introduces inaccuracy into delay measurement. To address this problem, various on-chip delay measurement techniques and online delay error detection mechanisms have been proposed.

**On-chip delay measurement:** The key issue in on-chip delay measurement is to implement a time-to-digit converter. Vernier Delay Line (VDL) is a popular technique to achieve this objective [4, 9, 12], where two signals propagate through respective delay chain, working as data and clock inputs for a series of Flip-Flops (FFs), respectively. The measurement result is then the sum of the delay difference in all the stages latching value '1'. In [6, 10], Ghosh *et al.* introduced another technique, wherein a capacitor is discharged linearly during measurement, while different delay intervals are converted into different voltage levels and then translated to digital signals. Ring oscillator-based methods have also been applied in path delay measurement by including the target path into an oscillation ring [7, 8].

In the above works, the delay introduced by the measurement circuit, including interconnect wires and logic gates, is assumed to be known or negligible, which severely impacts the measurement accuracy or even disables the practicality, especially considering the routing uncertainty at design stage and the ever-increasing effects of process variation. To tackle this problem, Wang *et al.* proposed a novel on-chip path delay measurement architecture, namely *Path-RO*, which builds an Oscillation Ring (OR) for each target path. Delay of the path and its returning loop is then translated into oscillation frequency while the delay of returning path is firstly set close to one clock cycle.

OR based techniques firstly needs a specific test mode for path delay measurement, where all the side-inputs of the gates along the target path are required to be non-controlling values in consecutive clock cycles to enable oscillation. Not all true paths in the circuit can satisfy such stringent requirement. Secondly, the operation condition in this test mode (e.g.,

power supply voltage, crosstalk and temperature) can be quite different from that in function mode. Therefore, significant deviation can occur between the measurement result and the actual path delay in functional mode. Thirdly, the time needed to measure the delay of a path via OR based methods is in the order of $k$ clock cycles, which is considerably long and costly.

**Online delay error detection:** As discussed earlier, it is increasingly difficult to predict and ensure the circuits' performance at design and manufacturing test stage. To accommodate this problem, various adaptive design methodologies have been presented, by detecting/predicting errors on-chip and compensate their effects.

One representative method is the so-called *Razor* technique used for error-tolerant microarchitecture design [5]. In this technique, a shadow latch is introduced to the receiving FF of each critical path. The value difference between the shadow latch and the FF indicates the corresponding path delay is outside of one clock cycle, and such errors can be corrected by flushing the pipeline. Power supply voltage can then be adjusted according to the timing violation rate to achieve better balance between energy consumption and performance.

As circuit aging has the unique feature that circuit delay increases slowly and steadily, Agarwal *et al.* proposed to conduct circuit failure prediction for this particular failure mechanism so that circuit can take proactive actions before errors actually appear [1]. To achieve this, an aging sensor is integrated inside each target FF for guardband checking, i.e., to check whether there are transitions close to the end of the clock period. Adjustment can then be applied for timing safety, e.g., prolonging the clock cycle or reducing the path delay by increasing supply voltage.

To sum up, on one hand, existing on-chip delay measurement techniques are not accurate enough due to the discrepancy between circuit's timing performance in functional mode and that of test mode. On the other hand, online delay error detection methods can only tell whether the circuit delay exceeds a certain limit. Clearly, if we are able to acquire the delay of critical paths with high accuracy as the circuit is working in function mode, such information will be of great help for process variation characterization, dynamic management and aging monitoring, which motivates the proposed concurrent online delay measurement architecture, namely *CODA*.

## III. Overview of CODA

We firstly overview CODA here, including the functionality of the main components and the delay measurement procedure in CODA.

**CODA Infrastructure:** The schematic structure of a circuit equipped with CODA is shown in Fig. 1. The upper rectangle encircles the circuit under measurement (CUM), where two FFs are selected as targets for delay measurement, while inside the lower rectangle is the measurement circuit. CODA is mainly composed of the following components.

1). *CODA Flip-Flop* (*CODA-FF*) at the receiving ends of critical paths. Two more ports (i.e., $P$ and $M$) and extra circuitries are introduced into CODA-FF, compared with normal FF, to facilitate online delay measurement.

2). *MUX*, which is used to select one signal at a time for delay measurement among multiple sources sharing the same delay measurement circuit.

3). *Source selection block*, which generates codes controlling the MUX, ensuring that at most one signal with transition in a pre-defined time window is selected during each measurement.

4). *Delay measurement unit* (*DMU*). While various types of on-chip delay measurement scheme can be utilized, in this work, we adopt the VDL-based technique, where the target signal and the clock signal are fed into DMU to measure the delay between them.

5). *M-Control block*, which is the main controller to ensure consistent operation of CODA.

6). *Storage module*, which stores the measurement results and outputs the corresponding value when requested.

**Delay Measurement Procedure:** In CODA, the delay measurement procedure is made up of two stages.

1) Probe route delay measurement. We name the route connecting a CODA-FF with DMU as *probe route*. Correspondingly, its delay is denoted as *probe route delay* or *probe delay*. As the IC starts to work in function mode, M-Control block firstly initializes CODA into probe delay measurement mode, wherein port $P$ of each CODA-FF is fed with logic '1' and port $M$ outputs signal for probe delay measurement. The source selection block selects the sources, one by one and orderly, so that their probe delays are measured, respectively. The measurement results of probe delay are then stored into the storage module.

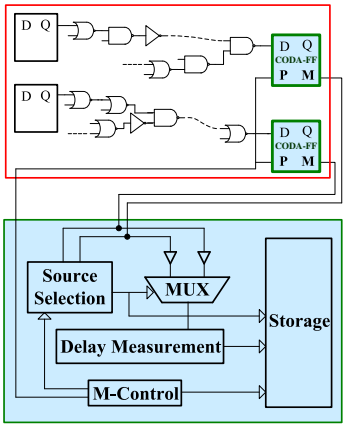



Fig. 1. Structure of the proposed online delay measurement architecture.

2) Online delay measurement. After finishing the probe delay measurement, CODA conducts online delay measurement. Port $P$ of each CODA-FF is fed with logic '0' and the target signals travel through port $M$ to DMU along respective probe route so that their delay is measured. The measurement result is then the total delay that includes the target path delay in the CUM and the probe delay, which is also saved in the storage module. Deducting the probe delay from the total delay can then obtain the target path delay.

**Characteristics of CODA:** Firstly, CODA does not disturb the function of CUM, and in fact CUM does not realize the existence of CODA. The only interference induced to CUM is the negligible delay overhead of CODA-FF compared to normal FF, with the detailed analysis presented in Section C.2.

Secondly, in CODA, only the receiving end of a critical path is connected with DMU, while the traditional techniques need to connect two ends of a path with DMU, which greatly reduces routing overhead.

Thirdly, critical FFs are selected as targets for delay measurement in CODA, instead of critical paths. Considering that frequently multiple critical paths converge at the same FF, CODA significantly reduces the complexity of measurement.

Fourthly, CODA acquires the path delay by deducting the probe delay from the total delay. Such measurement mechanism can tolerate routing uncertainty and most of the variations, guaranteeing high accuracy.

## IV. Detailed Design of CODA

In this section, we present the detailed design of the major components in CODA.

### A. CODA-FF Design

In a FF there are a master latch (ML) and a slave latch (SL). The timing constraint requires that the signal should propagate through the combinational circuit path, including the SL at the driving end and the ML in the receiving end, within one clock cycle. Such propagation delay determines the performance of VLSI circuits and it is just the target delay to be measured in this work.

In CODA, the normal FF at a target location is replaced by the proposed CODA-FF whose design is shown in Fig. 2, which is namely target FF. Compared with normal FF, CODA-FF is with two additional ports ($P$ and $M$) and three addition gates (inverter $E_0$, $E_1$ and multiplexer $M_0$).

Port $M$ is connected with DMU through probe route to transfer signal for delay measurement. Port $P$ is fed with mode signal from the M-Control block to select measuring the probe delay ($P = 1$) or total delay ($P = 0$), as mentioned in section III.

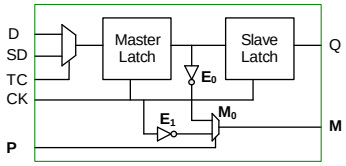



Fig. 2. Design of the proposed CODA-FF.

1) When $P = 1$, clock signal is hence transferred to port $M$. The transition of clock then propagates to DMU through the probe route while the other input of DMU is directly connected with clock. In such circumstance, the transition skew between the two inputs of DMU is just the probe route delay.

2) When $P = 0$, the output of ML is connected to port $M$. Therefore, the signal arriving at CODA-FF through the target path in CUM further transmits to DMU. Now the skew between the two inputs of DMU is therefore the total delay, which equals to the sum of the target path delay in CUM and the probe delay.

Consequently, the target path delay can be obtained by deducting the probe delay from the total delay.

The two inverters $E_0$ and $E_1$ are to isolate the CUM from the measurement circuit so that the only interference, induced to CUM by CODA, is that the output of ML in CODA-FF is with extra capacitance load resulting from the input of inverter $E_0$. Such extra capacitance results in prolonged delay, which in fact is negligible with detailed experimental results shown in section C.2. On the logic level, CUM does not realize the existence of CODA. That is, CUM and CODA work independently without interference.

About the target FFs selection, those FFs whose delay is crucial for circuits' performance and dynamic methodologies, should be selected as targets. Because of limited space, the selection details are out of the scope of this paper.

The proposed delay measurement mechanism and the corresponding CODA-FF design demonstrate two outstanding advantages. 1) The way of obtaining the path delay in CUM by deducting the probe delay from the total delay can eliminate most of the disturbance happening on the probe route, including process variation, routing uncertainty and aging effect. This enables high measurement accuracy and the sharing of one CODA module among multiple target FFs. 2) CODA introduces no interference to the circuit's function, which simplifies the design and operation of circuits equipped with CODA.

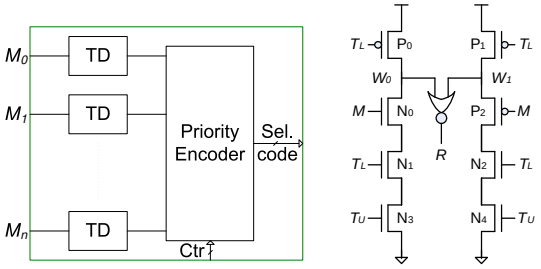
### B. Source Selection

Sharing one DMU among multiple target FFs can significantly reduce hardware overhead. Correspondingly, it must be guaranteed that at most one signal is selected at any time. Here we adopt transition detection and priority based selection mechanism, which firstly checks if there is transition in pre-defined time window for each signal and then select the one with highest priority among all the signals with transition.

Fig. 3(a) shows the design of the proposed signal selection block, which is mainly composed of Transition Detectors (TDs) and a priority encoder. The signals from CODA-FFs go into respective TD for transition checking. The output of a TD becomes active once it detects a transition in the pre-defined time window, which indicates that the corresponding critical path is activated and waiting for delay measurement. When there are multiple signals ready for measurement simultaneously, it is essential to guarantee that only one signal is finally selected. Here we adopt priority encoder to ensure the uniqueness, wherein the order is programmable via signals $Ctr$ controlled via JTAG port.

We adopt the TD design based on the stability checker in [1], as shown in Fig. 3(b). The TD detects whether transition happens on the input signal $M$ during the time window with lower bound $T_L$ and upper bound $T_U$ to guarantee delay measurement on critical paths[1]. $T_L$ and $T_U$ are generated by delaying clock signal $Clk$. To be specific, the rising edge of $T_L$ indicates the start point of the time window while the falling edge of

[1]Those short paths connecting to CODA-FFs need to be patched to have delay exceeding $T_U - Clk - T$ to guarantee the correct delay measurement.




(a) Source selection block. (b) TD.

Fig. 3. Source selection block and transition detector (TD).

$T_U$ indicates the end point. For example, suppose $T_L = Clk + 0.9T$ while $T_U = Clk + 1.1T$ with $T$ denoting clock cycle, only propagation delay in the range of $[0.9T, 1.1T]$ will be detected and measured.

The working mechanism of the proposed TD is as follows.

1): Before the time window, $T_L = 0$ and $T_U = 1$. Consequently, PMOS transistors $P_0$ and $P_1$ are on while NMOS transistors $N_1$ and $N_2$ are off. Therefore, nodes $W_0$ and $W_1$ are charged to logic '1' while output $R = 0$.

2): During the time window, $T_L = 1$ and $T_U = 1$. $P_0$ and $P_1$ are off while $N_1$, $N_2$ $N_3$ and $N_4$ are on. If there is a transition on signal $M$, both $N_0$ and $P_2$ will be on during a period of time so that $W_0$ and $W_1$ will both be discharged, resulting in $R = 1$. If there is no transition, only one of $W_0$ and $W_1$ will be discharged with $R$ staying at 0.

3): After the time window, $T_L = 1$ and $T_U = 0$. Both the pull-up and pull-down networks are off so that the output stays stable.

Extra buffers are added before the MUX (see Fig. 1) to ensure that selection codes are ready before the signals arrive at MUX. Since in CODA the path delay is acquired by subtracting probe delay from the total delay and both of them include the extra delay caused by the buffers, the measurement accuracy is guaranteed.

### C. Delay Measurement Circuit

The delay measurement circuit in CODA is to quantify the time interval between two input signals. There have been multiple kinds of methods targeting at measuring delay on chip that can be applied in CODA. Here we adopt the VDL based measurement method which can achieve high resolution with low hardware overhead.

**Structure of the proposed DMU:** Fig. 4 shows the proposed circuit design of DMU, which digitalizes how late the rising transition of signal $T$ happens after that of signal $R$. That is, $R$ is connected with reference signal while the to be measured signal is fed into $T$, where in CODA $R$ is connected with clock. The falling transition of $R$ and $T$ will be transformed into rising one before measurement [12]. The measurement circuit shown in Fig. 4 is made up of a series of $n+1$ stages, $S_n$ to $S_0$, while the combination of the output at each stage, $Q_n$ to $Q_0$, shows the measurement result.

In each stage there are two input ports (DI and CI) and three output ports (DO, CO and Q). DO from the previous stage is fed into DI of the next stage and CO is connected with CI. The proposed design of a stage module is shown in Fig. 5, where it can be seen that the signal DI always travels through path $p_D$ to DO, while CI travels through path $p_0$ or $p_1$ corresponding to the output of FF $Q = 0$ or $Q = 1$, respectively.

**Working mechanism of DMU:** Considering the timing of the signals in a stage, $t_{DO} = t_{DI} + d_{p_D}$ while $t_{CO} = t_{CI} + d_{p_0}$ or $t_{CO} = t_{CI} + d_{p_1}$; $t_D = t_{DI} + d_{B_i}$ while $t_{CK} = t_{CI}$. A stage will work in one of the following two ways corresponding to different timing relationship with $t_{setup}$ representing the setup time of the FF.

1) $t_D < t_{CK} - t_{setup}$. This timing relationship means that $s = t_{CI} - t_{DI} > d_{B_i} + t_{setup}$, indicating that the skew between CI and DI is longer than $d_{B_i} + t_{setup}$. Hereafter, $d_{B_i} + t_{setup}$ is denoted as $d_i$, representing the character delay of stage $i$. As the result of $t_D < t_{CK} - t_{setup}$, logic '1' will be latched in the FF and the output $Q$ changes to '1' from the initialized value '0', which controls CI to traverse path $p_1$ with buffer $B_1$. Path $p_1$ is so designed that $d_{p_1} = d_{p_D} - d_i$, meaning the delay of path $p_1$ is shorter

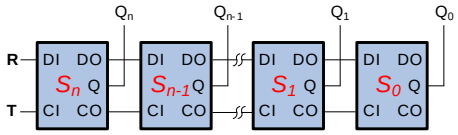



Fig. 4. The proposed delay measurement circuit.

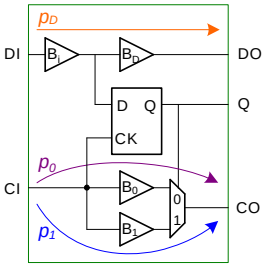



Fig. 5. A stage of the proposed delay measurement circuit.

by $d_i$ than that of path $p_D$, which results in the skew between CO and DO is shorter by $d_i$ than that between CI and DI.

2) $t_D \geq t_{CK} - t_{setup}$. Value '0' will be latched in the FF while $Q$ stays at '0'. Consequently, CI travels through path $p_0$ that is designed with $d_{p_0} = d_{p_D}$, which results in the skew between CO and DO equal to that between CI and DI.

In conclusion of cases 1 and 2, if the skew between two input signals of stage $i$ is longer than its character delay $d_i$, the output $Q_i$ becomes '1' while the skew is reduced by $d_i$ and fed into next stage; If the skew is shorter than $d_i$, $Q_i$ stays '0' and the skew stays unchanged for next stage. Therefore, the measurement range of such measurement circuit equals to the sum of all character delays, while summing the character delay of all the stages outputting '1' generates the measurement result for the skew between the two input signals.

The character delay of each stage in the proposed design can be with any desired value. Here we adopt exponential distribution of character delay along the series of stages where $d_i = d_0 2^i$, with motivation from [9] [14]. Such assignment is with the advantage of low hardware overhead and the measurement result is directly expressed as binary number, easing the following processing.

The waveform from an example of delay measurement is shown in Fig. 6, where the measurement circuit consists of five stages with character delay of 0.32, 0.16, 0.08, 0.04 and 0.02 *ns* from stage $S_4$ to $S_0$, respectively. Compared with $R_{ef}$, the other input signal $V_{in}$ is with 0.45 *ns* delay as shown in the upper part of Fig. 6, while the measurement output is demonstrated in the lower part as $Q_4Q_3Q_2Q_1Q_0 = 10110$. Correspondingly, the measurement result is $d_{VDL} = (10110)_2 \times d_0 = 22 \times 0.02 = 0.44$ *ns*, which is with -0.01 *ns* deviation from the real delay.

From the working mechanism it can be seen that (i) the measurement result will always be no greater than the real delay because the skew between the output signals DO and CO of each stage will always be no less than zero. That is, the output skew of the last stage $S_0$ will be thrown out of measurement and this uncovered part of the skew will result in measurement result less than real delay. Consequently, the measurement resolution is dependent on the character delay of the last stage $d_0$. With exponential distribution of character delay along the stages, the last stage can be designed with small character delay to achieve high measurement resolution while the previous stages with exponentially increasing character delay can provide long measurement range without the need of large number of stages; (ii) if multiple transitions happen in the measurement time window, the states of all measurement stages eventually settle down according to the last transition because a late transition will activate new states and flush the previous ones, which ensures CODA to measures the longest delay in the time window.

**Influence of clock skew:** In CODA, one input of DMU is always connected with clock, which means that the measured delay in DMU is in fact the skew between the other input signal and the clock at DMU. In addition, we use the clock signal inside CODA-FF for probe route delay measurement. Let $t_s$ represent the timing of the signal arriving at CODA-

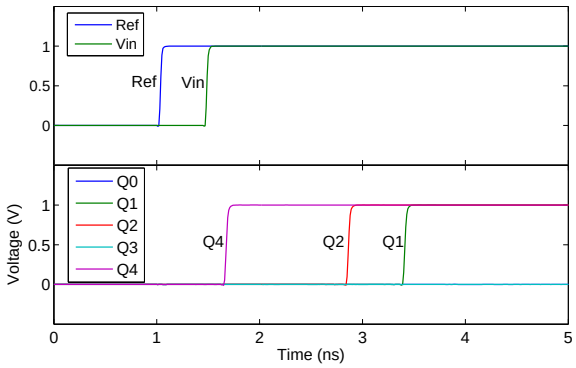



Fig. 6. Operation waveform of the delay measurement circuit.

FF whose delay is to be measured, while $t_{C\_M}$ and $t_{C\_FF}$ represent the timing of the clock at the measurement circuit and that at CODA-FF, respectively. In probe delay measurement mode, the delay $d_p$ measured by CODA can then be expressed as follows with $d_{probe}$ denoting the probe route delay.

$$d_p = t_{C\_FF} + d_{probe} - t_{C\_M} \tag{1}$$

In online delay measurement mode, CODA measures the total delay $d_t$.

$$d_t = t_s + d_{probe} - t_{C\_M} \tag{2}$$

Consequently, the final result from CODA $d_{CODA}$ is calculated by deducting the probe delay from the total delay.

$$d_{CODA} = d_t - d_p = t_s - t_{C\_FF} \tag{3}$$

That is, the final result from CODA represents the skew between the function signal arriving at CODA-FF and the clock at the same CODA-FF. Such comparison is in fact to check whether the timing constraint is satisfied. For example, a result with $d_{CODA} = 1.1T$ indicates that the timing constraint has been violated at the corresponding CODA-FF. Therefore, the final result from CODA is exactly the information needed for timing constraint checking. In other words, CODA can check to what extent the timing constraint can be satisfied at the locations equipped with CODA-FFs, no matter how much clock skew exits.

### *D. Storage of Measurement Results*

The measurement results from CODA include source No., delay value and delay type (probe delay or total delay). If there is no dedicated storage attached to CODA, such measurement results need to be collected each time by such circuits as the master processor on chip. Here we propose an effective storage design with moderate overhead that can significantly ease the processing of measurement results, which is demonstrated in Fig. 7.

In each row of the storage module, there are two storage blocks for one source, one storing the probe delay while the other storing the longest total delay. In probe delay measurement mode, the measurement result is stored in the first storage block. In online delay measurement mode, the R/W control block first compare a new measurement result with the content in the second block of the corresponding source. Only if the new measurement result is with larger value, the second block is refreshed with the new value. Once there comes the request for delay reading, the desired source is selected and the corresponding values are outputted.

## V. Experimental Results

In this section, we report SPICE simulation results for CODA, based on a 90 *nm* IC fabrication technology with nine metal layers and $1V$ power supply voltage.

### *A. Measurement Accuracy*

In the experiments we have implemented CODA with seven stages of delay measurement circuit, where the character delay of stage $i$ $d_i = 0.02 \times 2^i$ *ns*. Table I shows the measurement results, with time unit of *ns*, for eight target FFs in ISCAS'89 benchmark circuit S38417, as a proof of

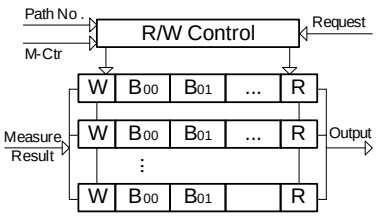



Fig. 7. Storage module for delay measurement results.

concept. The FFs are selected to cover paths with delay in the range of [0.8$L$, $L$] with $L$ as the longest path delay. After the source No. listed in the first column of Table I, the path delay from Spice simulation is shown in the second column as $d_{simu}$. The following three columns specify the results from CODA, where $d_p$, $d_t$ and $d_{CODA}$ denote probe delay, total delay and path delay from CODA, respectively, with $d_{CODA} = d_t - d_p$. The last two columns then shows the comparison between the path delay from Spice simulation and that from CODA, where $\Delta d = d_{CODA} - d_{simu}$ and $Err = |\Delta d / d_{simu}| \times 100\%$.

The measurement inaccuracy in CODA results from (I) the delay variation of the probe route, and (II) the resolution of the delay measurement unit.

Considering part I, firstly, CODA generate path delay by measuring the probe delay and deducting it from the total delay, which can tolerate probe delay variation resulting from process variation and routing uncertainty at design stage. The uncovered measurement inaccuracy is from the operation condition variation. That is, the delay of a probe route can be different during each measurement because of variation in temperature, power degradation and crosstalk. This kind of variation is hard to be thoroughly eliminated, requiring that there is no probe route needed or the probe delay and total delay are measured at the same time, which is a limitation of CODA. However, CODA can mitigate such variation by suffering less probe delay variation benefiting from the short probe routes needed in CODA, while the detailed analysis of the routing in CODA is explained in section C.1.

For part II, the resolution of VDL based delay measurement circuit with exponential distribution of character delay is determined by the smallest character delay $d_0$, as explained in section C. That is, the measurement result $d_{VDL}$ is always in the range of $[d - d_0, d]$, where $d$ is the practical delay value. Consequently, the path delay that is obtained by deducting the probe delay from the total delay is with deviation from the practical value in the range of $[-d_0, d_0]$. The measurement error in our experiments is from -0.005 to 0.014 *ns* corresponding to $d_0 = 0.02ns$, which certifies the effectiveness of the proposed delay measurement circuit and CODA.

### B. Impact of Process Variation

In CODA, process variation on probe route can be eliminated while the process variation on target paths are directly measured, which will not degrade the measurement accuracy. Process variation, however, can also occur on the delay measurement circuit itself, i.e., the pre-defined buffer delay used in VDL circuit may be inaccurate and hence induces measurement inaccuracy. We therefore conduct experiments to show its impact on CODA.

As shown in [2], the variation of multiple types of semiconductor device factors can be unified into changes of the threshold voltage $V_{th}$. In other words, variation of $V_{th}$ can reflect multiple types of variation. Therefore, we introduce $V_{th}$ variation into the proposed circuit and conduct Monte-Carlo simulation to analyze how robust CODA is with the existence of process variation. The $V_{th}$ variation of a transistor is assumed to be inversely linear to the sqaure root of its size [2], as shown in formula 4.

$$\Delta V_{th} \propto \frac{C}{\sqrt{W_{eff} L_{eff}}} \tag{4}$$

In formula 4, $C$ is a constant value which is associated with manufacturing technology. $W_{eff}$ and $L_{eff}$ are the effective channel width and length of a transistor, respectively. For transistor with minimum size, we set $V_{th}$ to follow normal distribution with sigma as 5% of its mean value. Different $\Delta V_{th}$ is then applied onto each transistor according to its size. Nine

| No. | $d_{simu}$ | $d_p$ | $d_t$ | $d_{CODA}$ | $\Delta d$ | $Err$ (%) |
|---|---|---|---|---|---|---|
| 0 | 1.076 | 0.40 | 1.48 | 1.08 | 0.004 | 0.36 |
| 1 | 1.062 | 0.12 | 1.18 | 1.06 | -0.002 | 0.20 |
| 2 | 1.034 | 0.36 | 1.40 | 1.04 | 0.006 | 0.53 |
| 3 | 0.990 | 0.08 | 1.06 | 0.98 | -0.010 | 1.02 |
| 4 | 0.966 | 0.64 | 1.62 | 0.98 | 0.014 | 1.48 |
| 5 | 0.942 | 0.24 | 1.18 | 0.94 | -0.002 | 0.24 |
| 6 | 0.922 | 0.56 | 1.48 | 0.92 | -0.002 | 0.19 |
| 7 | 0.865 | 0.72 | 1.58 | 0.86 | -0.005 | 0.55 |

TABLE I
DELAY MEASUREMENT RESULTS FOR BENCHMARK CIRCUIT S38417

delay values from 0.8 *ns* to 1.2 *ns* are measured by the proposed circuits in the presence of process variation and Monte-Carlo simulation with 1000 iterations is performed for each delay value. We have not enlarged the size of the transistors in the proposed circuit on purpose, i.e., all transistors are with the minimum size that can realize the functionality.

The measurement results under process variation is shown in Table II, where the first column shows the delay to be measured while the second and eighth row lists the classification of measurement error. The other entries of the table show the percentage of measurement with the corresponding error. For example, considering the measurement results for 0.8 *ns*, 13%, 44%, 27% and 16% of the 1000 iterations is with error of -40, -20, 0, 20 *ps*, respectively. The distribution of the measurement result is also shown in Fig. 8 by histogram for clarity, where the one with less shadow is with larger error. Only 0.1% of the total measurement is with the maximum error of 60 *ps*, and 15% is with error larger than 20 *ps*. The remaining can achieve accuracy within 20 *ps*, which accounts for nearly 85% of the total measurements. Such measurement error distribution demonstrates that CODA is insensitive to process variation and therefore can achieve fine measurement resolution in the presence of process variation. Enlargement of the transistor sizes can further improve the robustness of the delay measurement circuit.

### C. Overhead of CODA

CODA introduces two kinds of overhead into the original circuit: (i). the hardware used to implement it; and (ii). the extra delay due to the replacement of normal FF by CODA-FF.

## C.1 Hardware Overhead

The hardware overhead includes the logic gates used to implement the functionality of CODA and the wires needed to connect the target FFs in CUM with the delay measurement unit.

**Transistor overhead:** The transistor overhead required for CODA mainly exists in four parts.

1) Compared to traditional SFF, the proposed CODA-FF needs two more inverters and one more MUX2 gate, which costs extra hardware overhead roughly equivalent to four NAND2 gates of minimum size.

2) One stage in the delay measurement circuit needs hardware overhead equivalent to about 14 NAND2 gates. The total hardware overhead there can then be expressed as $14N_m$ with $N_m$ representing the number of stages in the measurement circuit.

3) For each target, hardware overhead of about 15 NAND2 gates are needed to build the transition detector, encoder, decoder, MUX, etc.

4) To store the measurement result, 28 NAND2 gates equivalent overhead are required to construct the memory part for each target.

Therefore, in the case with $N_t$ target FFs and $N_m$ stages of DMU, the hardware overhead is equivalent to $A = 4N_t + 14N_m + 15N_t + 28N_t = 47N_t + 14N_m$ NAND2 gates. For example, in a design with 64 targets and eight stages of delay measurement, the total transistor overhead is equivalent to 3k NAND2 gates. Such amount of transistor overhead is acceptible for nowaday IC, especially considering that at present IC is mostly routing limited.

**Routing overhead:** Compared to transistor overhead, wire routing overhead is more critical nowadays since the performance of VLSI circuits is generally more dependent on wire routing. Especially, those long wires demonstrate long delay and cost lots of crucial routing resources. For the oscillation ring based delay measurement methods, a returning loop is needed for each target path, traversing backward from the receiving end to

| Delay | Measurement error | | | | | |
|---|---|---|---|---|---|---|
| | -60*ps* | -40*ps* | -20*ps* | 0*ps* | +20*ps* | +40*ps* |
| 0.80 *ns* | | 13% | 44% | 27% | 16% | |
| 0.90 *ns* | 1% | 12% | 42% | 45% | | |
| 1.00 *ns* | | 15% | 33% | 42% | 9% | 1% |
| 1.10 *ns* | | 1% | 51% | 35% | 13% | |
| 1.20 *ns* | | 9% | 45% | 35% | 11% | |
| | -50*ps* | -30*ps* | -10*ps* | 0*ps* | +10*ps* | +30*ps* |
| 0.85 *ns* | 7% | 23% | 43% | | 22% | 5% |
| 0.95 *ns* | | 30% | 44% | | 21% | 5% |
| 1.05 *ns* | 3% | 24% | 40% | | 29% | |
| 1.15 *ns* | 3% | 29% | 46% | | 22% | 3% |

TABLE II
ERROR OF DELAY MEASUREMENT UNDER PROCESS VARIATION

measurement unit and then finally to the driving end of the path. Therefor, the returning loop is with length similar to critical paths. What's worse, it is very possible that a considerable part of returning loops is even much longer than critical paths, considering that multiple paths share one measurement unit which cannot be close to all the paths. Consequently, so long returning loops occupy a lot of routing resources. In CODA, only the receiving ends of the paths are required to be connected with the measurement circuit while there is no routing requirement at the driving ends. Such character can greatly reduce the routing overhead, especially considering that the measurement circuit can be placed near those target FFs.

### *C.2 Delay overhead of CODA-FF*

To equip CODA, CUM needs to replace the traditional SFFs at the target locations by the proposed CODA-FFs. The only interference induced to CUM is the extra delay caused by the extra capacitance load of the extra gates in CODA-FFs.

Here we show the worst case delay overhead resulting from the proposed CODA-FF. First, we build the traditional SFF where all the internal logic gates are with driving capability equivalent to minimum size inverter. Thereafter, we add the extra gates to build the proposed CODA-FF. We compare the path delay with traditional SFF and the proposed CODA-FF attached to the receiving end, respectively, which can then show the extra delay caused by the proposed CODA-FF. In the comparison experiments the last gate before the FF is designedly assigned with minimum size inverter. Such arrangement is to create worst case condition, where all the related gates are with minimum driving capability and maximum extra delay can then result from the extra load in the proposed CODA-FF.

Spice simulation results show that the maximum delay is 4.8 *ps*, which is negligible compared to path delay in the order of *ns*. Such delay value is similar to that of [14], which is much smaller than that in the initial version [13]. What's more, such delay can be greatly reduced by increasing the size of the related gates moderately. Therefore, the proposed CODA-FF introduces negligible interference onto the circuits' operation.

#### *D. Measurement Time*

The time needed to complete a delay measurement in CODA, $d_{measure}$, depends on three factors: I). The probe route delay, $d_{probe}$; II). The desired time window within which the delay is to be measured, $d_{window}$; and III). The response time of DMU, $d_{DMU}$.

I) As discussed earlier, considering the low routing overhead in CODA, $d_{probe}$ is usually smaller than one clock cycle.

II) $d_{window}$ is usually much smaller than one clock cycle as well. For example, suppose we set $T_L = 0.9T$ and $T_U = 1.1T$, then the time window size is $d_{window} = 0.2T$.

III) The response time of DMU is determined by its measurement range $R_{DMU}$ that should be no less than $d_{probe} + d_{window}$, and typically $R_{DMU} = 1T$ is enough to cover the delay variation for the to-be-measured transition arriving at DMU, and the measurement response can be obtained in less than $2T$.

Therefore, $d_{measure} = d_{probe} + d_{window} + d_{DMU}$ is usually smaller than $4T$, i.e., the measurement result is ready within four clock cycles. Compared to the oscillation ring based delay measurement techniques that require several thousands of cycles for each measurement, CODA is clearly much more efficient.

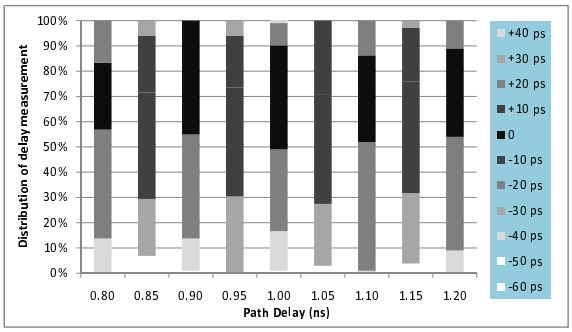



Fig. 8. Distribution of measurement error under process variation

## VI. CONCLUSION

With technology scaling, integrated circuits behave more unpredictable due to process variation, environmental changes and aging effects, and it is important to be able to collect the actual propagation delay of critical paths when the circuit is running in normal functional mode. Motivated by the above, in this paper, we propose a novel concurrent online delay measurement architecture for critical paths, namely CODA. Our proposed technique is able to achieve high measurement accuracy with relatively low cost, as demonstrated in our experimental results.

## VII. ACKNOWLEDGEMENT

This work was supported by the General Research Fund CUHK418708 and CUHK418111 from Hong Kong SAR Research Grants Council (RGC).

## VIII. REFERENCES

[1] M. Agarwal, et al. Circuit Failure Prediction and Its Application to Transistor Aging. In *Proc. IEEE VLSI Test Symposium (VTS)*, pp. 277–286, 2006.

[2] A. Asenov, et al. Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale mosfets. *Electron Devices, IEEE Transactions on*, 50(9):1837–1852, Sept. 2003.

[3] S. Borkar. Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *IEEE Micro*, 25(6):10–16, 2005.

[4] R. Datta, et al. A scheme for on-chip timing characterization. In *Proc. IEEE VLSI Test Symposium (VTS)*, pp. 24–29, 2006.

[5] D. Ernst, et al. Razor: Circuit-level Correction of Timing Errors for Low-power Operation. *IEEE Micro*, 24(6):10–20, Nov.-Dec 2004.

[6] S. Ghosh, et al. A novel delay fault testing methodology using low-overhead built-in delay sensor. *IEEE Transactions on Computer-Aided Design*, 25(12):2934–2943, 2006.

[7] M. Bhushan, et al. Ring Oscillators for CMOS Process Tuning and Variability Control. *IEEE Transactions on Semiconductor Manufacturing*, 19(1):10–18, Feb. 2006.

[8] M. Nourani and A. Radhakrishnan. Testing On-Die Process Variation in Nanometer VLSI. *IEEE Design & Test of Computers*, 23(6):438–451, Nov. 2006.

[9] S. Pei, H. Li, and X. Li. A low overhead on-chip path delay measurement circuit. In *Proc. IEEE Asian Test Symposium (ATS)*, pp. 149–154, 2009.

[10] A. Raychowdhury, S. Ghosh, and K. Roy. A novel on-chip delay measurement hardware for efficient speed-binning. In *Proc. IEEE International On-Line Testing Symposium*, pp. 287–292, 2005.

[11] D. Schroder and J. F. Babcock. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of Appllied Physics*, 94(1):1–18, July 2003.

[12] M.-C. Tsai, C.-H. Cheng, and C.-M. Yang. An all-digital high-precision built-in delay time measurement circuit. In *Proc. IEEE VLSI Test Symposium (VTS)*, pp. 249–254, 2008.

[13] X. Wang, M. Tehranipoor, and R. Datta. Path-RO: A Novel On-Chip Critical Path Delay Measurement Under Process Variations. In *Proc. International Conference on Computer-Aided Design (ICCAD)*, pp. 640–646, 2008.

[14] X. Wang, M. Tehranipoor, and R. Datta. A Novel Architecture for On-chip Path Delay Measurement. In *Proc. IEEE International Test Conference (ITC)*, pp. 1–10, 2009.