Path-RO: A Novel On-Chip Critical Path Delay Measurement Under Process Variations

Xiaoxiao Wang[1], Mohammad Tehranipoor[1], and Ramyanshu Datta[2]
[1]Dept. of ECE, University of Connecticut, {xwang,tehrani}@engr.uconn.edu
[2] Texas Instruments, rdatta@ti.com

Abstract— As technology scales to 45nm and below, process variations will present significant impact on path delay. This trend makes the deviation between simulated path delay and actual path delay in a manufactured chip more significant. In this paper, we propose a new on-chip path delay measurement structure called path-based ring oscillator (Path-RO). The proposed method creates an oscillator from a targeted path for which it is used to measure path delay on-chip under the impact of process variations. To alleviate accuracy degradation caused by the architecture itself, a high-accuracy calibration process is presented. Through experimental results on Path-ROs inserted in ITC'99 b19 benchmark, we obtain path delay distribution under different process variations. The accuracy and efficiency of path delay measurement using Path-RO are also verified by comparing the results obtained from post-layout Hspice simulations.

I. INTRODUCTION

The scaling down of semiconductor devices in each technology generation improves performance and integration density. At 45nm technology and below, process and environmental variations will significantly impact product performance and power. Process variations in circuits consisting of nominally identical structures may occur locally, across chip, across wafer, from wafer-to-wafer and from lot-to-lot. As we are entering the "late-CMOS" age [1], deep nanometer scaling causes increased complexity with respect to lithography techniques and dopant concentration. Parameters, such as channel length L, oxide thickness T_{ox}, threshold voltage V_{th}, etc., will have larger relative variations since shrinking device feature sizes lead to reduced control over the manufacturing process. Critical paths, which usually connect one flip-flop/primary input from one corner of a chip to another flip-flop/primary output in another corner, are prone to have larger variation range and less predictability from design stage. As we go into multi-giga hertz era, the number of path delay faults observed during first silicon debug increases significantly [2]. Therefore, the impact of process variations on critical and delay sensitive paths must be accurately measured.

To improve the yield, every new product goes through several revisions and modifications before it becomes a shippable product. Those revisions and modifications are usually done in a step called first silicon debug. The faster faults and performance limiters are identified and the more information is collected in each version of first silicon, the fewer revisions are needed, which results in reduced time-to-market. On-chip test structures used during silicon debug can provide valuable information about type and sources of defects and failures.

When on-chip path delays fall below hundreds of pico seconds, path delay measurement performed by off-chip equipments will be dominated by parasitic capacitance, resistance of probe and transmission line impedance fluctuations. Therefore, precise on-chip delay measurement methodologies are of high demand in semiconductor industry for rapid silicon debug and yield ramp up. By measuring

under actual operating conditions, the designers will be able to perform efficient process tuning and variability control. Such post silicon timing characterization and on-chip measurements are fundamentally closer to technology applications than simulation.

Several types of on-chip delay measurement architectures have already been presented in literatures. Datta et al. [3] measure path delay by Modified Vernier Delay Line (MVDL), which converts path delay into digital "0" and "1" series in flip-flops forming MVDL and improves the measurement resolution below 100ps. However, the premise of high accuracy measurement is delicate by symmetric routing in branches of MVDL, which will constrain the place and route tool [4]. [5] and [6] utilize built-in delay sensor consisting of saw-tooth voltage generator and comparator to convert path delay from analog domain into digital domain. However, accuracy loss may be caused by saw-tooth waveform distortion and analog to digital translation. Su et al. [7] use phase detector to detect phase difference between bus input and output, and translate it to bus delay. However, the different length of wires connecting bus input/output and phase detector makes the detected phase difference inaccurate.

Some on-chip test structures utilizing ring oscillators to monitor process variations have already been presented in literature [8] [9] [10] [11]. These test structures were intended to provide information for chip performance evaluation especially path delay prediction. However, the test structures placed in scribe lines do not adequately capture all the variations to provide insight into their sources [8] [9]. The process monitors placed in various locations in a chip provide some information on variations but little on their actual impact on circuit performance and power [10] [11]. Therefore, design and test engineers must perform a timely analysis to understand these issues due to the uncorrelated data collected using monitors. Such monitors are also unable to address environmental variations effects such as crosstalk, IR-drop and temperature.

However, ring oscillator can be used more efficiently for path delay measurement by making the path under test (PUT) part of the oscillator. With the help of oscillator, delay on path and its returning loop can be translated into oscillation frequency, which can be easily read out by counter with no potential accuracy loss. The need for test equipments and tackling their associated parasitics can be eliminated achieving low overhead as well as high accuracy design aim. Arabi et al. [12] and Wu et al. [13] equal path-delay directly to half of oscillation period. However, the delay caused by returning loop is neglected making the measured path delay inaccurate. Paul et al. [14] make PUT as a part of ring oscillator only to collect path delay variation information under different supply voltages rather than measuring actual path delay and analyzing the impact of process variations.

In this paper, we propose a novel on-chip test structure, called Path-RO, that configures critical and near-critical paths into ring oscillator. A Measurement and Calibration (MC) unit is shared by all of the selected paths, which can perform loop back delay calibration and oscillation frequency measurement. To ensure oscillation, one inverter will be added to the returning loop if the number of inverting logic on path is even. The data collected by counter in MC unit can be

640

used for path delay calculation and process variations evaluation. The contributions this paper makes are:

1) It utilizes existing paths to form Path-RO instead of inserting extra ring oscillators, which ensures better understanding of process variations' impact on path delay. As a result, intra/inter-die process variations' impact on path delay can be obtained by comparing simulated (without any process variations) and measured (with actual process variations) path delay results.

2) With continuous scaling of technology and decreasing clock cycle, delay introduced by returning loop connecting the output and input of PUT will greatly impact the measured oscillation frequency. To improve the accuracy of path delay measurement, the extra delay introduced by returning loop is obtained using a calibration process and could be easily removed from measured data.

3) Path-RO is a low cost and time-efficient on-chip test strategy which requires no test equipment. The small MC unit can be easily synthesized and existing ATPG tools can be used in Path-RO measurement flow.

In this paper, Section II presents the proposed Path-RO architecture. Section III describes the implementation flow of Path-RO based path delay measurement. Section IV presents the calibration strategy and calibration circuit design flow we propose. Section V describes the special calibration launch flip-flop (CLFF) and calibration capture flip-flop (CCFF) used in Path-RO architecture. Simulation results are presented in Section VI. The path delay measurement accuracy under process variations is verified by the simulation results in this section. Finally, Section VII presents the concluding remarks.

II. PATH-RO ARCHITECTURE

The architecture of Path-RO is established in circuit which already has scan chain for manufacturing test and consists of three major parts (see Figure 1).

First part is the basic critical path, which is the (or one of the) longest path(s) chosen as PUT from netlist by statistical or static timing analysis tools [16].

Second part is the MC (measurement and calibration) unit, which is a single four-components unit shared by all paths as shown in Figure 1. MC unit consists of:

1. A calibration circuit (see Figure 2), which calibrates the returning loop delay during the calibration mode, while working as a part of returning loop during oscillation mode. The calibration circuit contains a configurable chain of buffers and multiplexers that can be controlled by user.

2. An M-to-1 Mux aiming at selecting one of M PUTs for calibration or measurement.

3. A small control unit, which is actually a short scan chain made of dummy flip-flops, provides control signals to all Muxes in Path-RO architecture.

4. An n-bit counter which can record up to $2^n - 1$ oscillation cycles.

According to the analysis of Section III, n can be defined based on user's predefined resolution. Since the routing and location of MC unit will not affect both oscillation counting result and calibration process, there will be no constraints when adding MC unit to the layout.

The third part in Path-RO architecture are calibration launch flip-flop (CLFF) and calibration capture flip-flop (CCFF). The scan flip-flops (SFFs) located at the input of the selected critical paths used to construct Path-ROs are replaced by CCFFs, while SFFs at the output of the selected critical paths are replaced by CLFFs. The details of
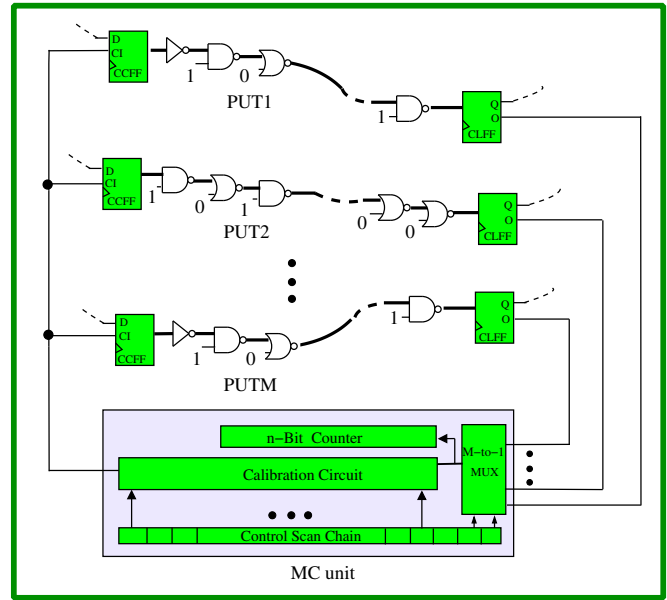

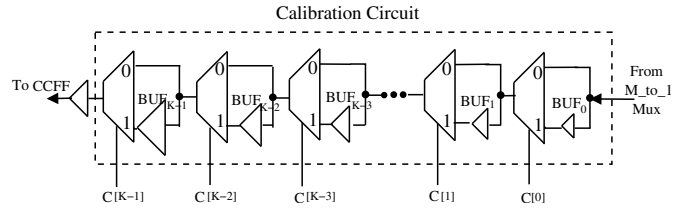Fig. 1.   Path-RO architecture.

Fig. 2.   Calibration circuit.

CLFF and CCFF design and their associated operating modes will be presented in Section V.

III. IMPLEMENTATION FLOW OF PATH-RO BASED PATH DELAY MEASUREMENT

Figure 3 outlines the general flow for Path-RO implementation; the flow can be divided into five steps as following:

● **Step 1:** In this step, critical paths are chosen from netlist by static/statistical timing analysis tool (here, a static timing analysis tool is used [16]). We have performed both pre-layout and post-layout critical path selection for large benchmarks. According to our critical path selection results for ITC'99 b19 benchmark [19], 20 of the top 30 selected pre-layout critical paths still remain in the first 30 post-layout critical path list. That is, the majority of critical paths selected from pre-layout netlist to build Path-RO most likely are post-layout critical or near-critical paths as well. Statistical timing analysis tools can also be used for critical path selection considering process variations.

● **Step 2:** The netlist modifications including the insertion of MC unit and replacing the selected critical paths' starting and ending SFFs by CCFFs and CLFFs are performed in this step.

● **Step 3:** From the data we collected for b19 benchmark in 180nm standard cell technology [17], the contribution of the returning loop delay to overall oscillation period can be up to 10%, and with technology scaling this percentage is expected to increase. Thus calibration process will be applied in this step to eliminate the impact of returning loop during on-chip measurement. The calibration is done by changing the delay of calibration circuit shown in Figure 2 and performing at-speed transition delay fault (TDF) test. By disabling and enabling the buffer with smallest delay step in calibration circuit

and TDF test passes and fails, the delay of returning loop t_{rl} will be set to one clock cycle. The calibration process and calibration circuit design will be described in details in Section IV.

● **Step 4:** In this step, we activate Path-RO in oscillation mode for duration time T, and N oscillation cycles will be recorded by a rising edge effective counter. The counter value increases by 1 when signal completes travel around Path-RO twice. T and N are determined according to predefined path delay measurement resolution. For example, during time T, the counter records N times of oscillation. Thus the oscillation period can be ideally calculated by $\frac{T}{N}$. However, in our experiments, we set the measurement time T as a constant value making ± 1 deviation between recorded value N and actual oscillation iteration number. According to [18], the accuracy of an oscillation system is lower bounded by $(\frac{T}{N-1} - \frac{T}{N})$. If r_o is the desired oscillation period measurement resolution, then

$$\frac{T}{N-1} - \frac{T}{N} = \frac{T}{N(N-1)} \leq r_o \qquad (1)$$

In Equation 1, $\frac{T}{N-1}$ approximates the length of an oscillation period which can easily be estimated by timing analysis tools, therefore,

$$N \geq \frac{t_{period\_est}}{r_o} \qquad (2)$$

where t_{period_est} is the estimated oscillation period during simulation only used for obtaining T. We obtain the minimum number of N through Equation 2, then the minimum oscillation measurement time (T) is obtained by $t_{period\_est} \cdot N$.

To ensure oscillation, a pattern applying non-controlling values to the off-path inputs of the gates on PUT is generated (robust test). The pattern can be easily obtained by ATPG tool [16] for TDF fault test of the critical path on the enabled Path-RO. During oscillation mode we apply the pattern, disable system clock, and keep primary inputs constant, which will ensure same non-controlling values are applied to the off-path inputs during the entire oscillation period.

● **Step 5:** Finally, path delay t_{path} can be calculated by

$$t_{path} = \frac{T}{2N} - t_{rl} \qquad (3)$$

where t_{rl} is the delay of returning loop and has been calibrated to one clock cycle in Step 3 ($t_{rl} = t_{clock}$) using TDF test and configuration of calibration circuit.

As an example, in an actual path delay measurement for b19 benchmark, in Step 3 the returning loop delay has been calibrated to one clock cycle which is $7.1ns$, and in Step 4 the oscillation measurement time is obtained to be $T = 6.9\mu s$. During this measurement time, $N = 246$ oscillation cycles have been recorded by counter. From Equation 3, $t_{path} = 6.92ns$.

IV. CALIBRATION

In sub-100nm technologies, the contribution of interconnect delay to Path-RO oscillation period will be significant. Thus the simulated returning loop delay without considering process variations cannot represent their actual on-chip value. In this section, we present a process-variation robust, low-overhead and time-efficient calibration process to eliminate the impact of returning loop delay (t_{rl}) on oscillation period (t_{period}) and accurately measure path delay (t_{path}).

The oscillation period t_{period} of Path-RO can be seen as:

$$t_{period} = 2(t_{path} + t_{rl}) \qquad (4)$$
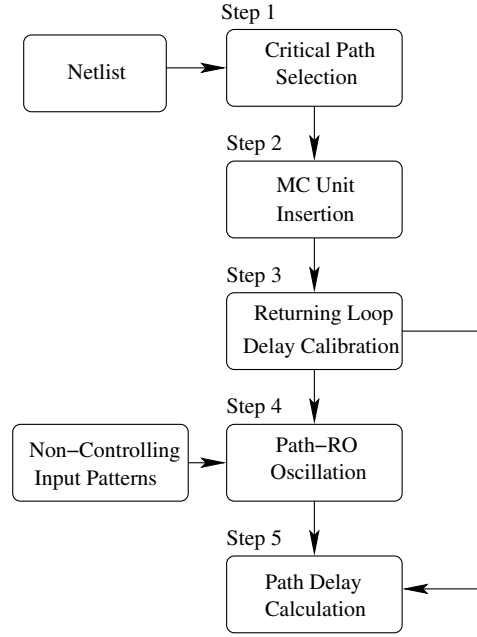
where t_{rl} can be further divided into:

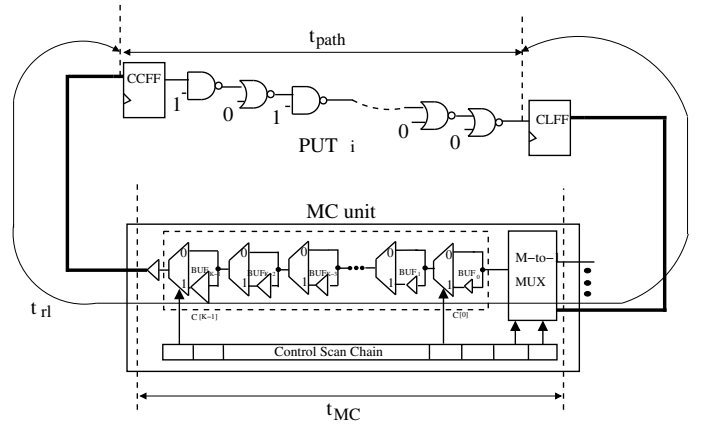Fig. 3. Path-RO based path delay measurement implementation flow.

Fig. 4. Components of oscillation period t_{period}.

$$t_{rl} = t_{rlw} + t_{MC} \qquad (5)$$

where t_{rlw} is the fixed delay introduced by interconnects connecting CLFF to MC unit and MC unit to CCFF (shown as bold lines in Figure 4). t_{MC} is the delay of MC unit (see Figure 4).

Due to process variations, t_{rlw} and t_{MC} are both uncertain to designers. However, by using the calibration circuit that provides a controllable and flexible t_{MC} combined with TDF test, we can make t_{rl} exactly one clock cycle. It is expected that: 1) different PUTs in Path-RO have different t_{rlw}s, therefore different t_{MC}s are needed; 2) because of process variations, same Path-RO on different dies may also have different t_{rlw}s. Thus same Path-RO may also need different t_{MC}s from die to die. A delay controllable and flexible calibration circuit in MC unit can satisfy these demands.

t_{rl} then can be removed from measured t_{period} leaving only t_{path}, which is the actual path delay of PUT, obtained by:

$$t_{path} = \frac{t_{period}}{2} - t_{rl}$$
$$= \frac{t_{period}}{2} - t_{clock} \qquad (6)$$

A. Calibration Process

During calibration, we treat the returning loop as a timing path and apply TDF tests. A passed TDF test means $t_{rl} \leq t_{clock}$ while a failed TDF test indicates that $t_{rl} > t_{clock}$. If the returning loop passes the TDF test, then t_{rl} should be increased by increasing the binary value of the K-bit control vector $\mathbf{C}$=($C[K-1]...C[1]$, $C[0]$) for calibration circuit in Figure 2. Conversely, if the returning loop fails the TDF test, the binary value of control vector $\mathbf{C}$ should be decreased to some extent. As seen in Figure 2, $C[0] = 0$ bypasses the minimum delay buffer (BUF0) and $C[0] = 1$ connects BUF0 to Mux. If by switching the last control bit $C[0]$ from 0 to 1, a previously passed TDF test fails, or switching $C[0]$ from 1 to 0, a previously failed TDF test passes, the delay of returning loop t_{rl} is verified to be one clock cycle, with accuracy loss in pico seconds range (minimum size buffer delay). It should be noted that, these pico seconds range deviation can even further be reduced by decreasing the delay difference between the two inputs of Mux controlled by signal $C[0]$, which is outside the scope of this paper.

The calibration flow is shown in Figure 5. After scanning in the control vector C_i and TDF pattern in test mode, the circuit is switched to calibration mode. Then TDF pattern is applied to make a transition along the returning loop. The TDF test result will then be analyzed to see if increase or decrease in control bits is required. This flow must be repeated for all PUTs.

B. Calibration Circuit Design

The following steps generalize the calibration circuit design flow:
• **Step 1:** In this step, we will estimate the flexible range calibration circuit must offer. Post-layout Hspice simulation [16] is performed in this step to measure $t_{rlw1}, t_{rlw2},..., t_{rlwM}$ for PUT_1, $PUT_2,...,PUT_M$. t_{rlw_max} and t_{rlw_min} are the maximum and minimum wire delays among them, respectively. To ensure that t_{rl} always can reach one clock cycle, the delay range of MC unit should satisfy the following equations:

$$t_{MC\_min} = t_{clock} - t_{rlw\_max}$$
$$t_{MC\_max} = t_{clock} - t_{rlw\_min} \qquad (7)$$

where t_{MC_min} and t_{MC_max} are minimum and maximum delay MC unit should offer. According to Equation 7, the minimum flexible range FR_{min} the calibration circuit in MC unit must be able to offer is,

$$FR_{min} = t_{MC\_max} - t_{MC\_min}$$
$$= t_{rlw\_max} - t_{rlw\_min} \qquad (8)$$

And the fixed delay t_{fixed}, caused by Muxes in calibration circuit and M-to-1 Mux, is obtained by:

$$t_{fixed} = t_{MC\_min}$$
$$= t_{clock} - t_{rlw\_max} \qquad (9)$$

Hence an MC unit with a known t_{fixed} (obtained from Equation 9), and a flexible delay range of FR_{min} can generate a one
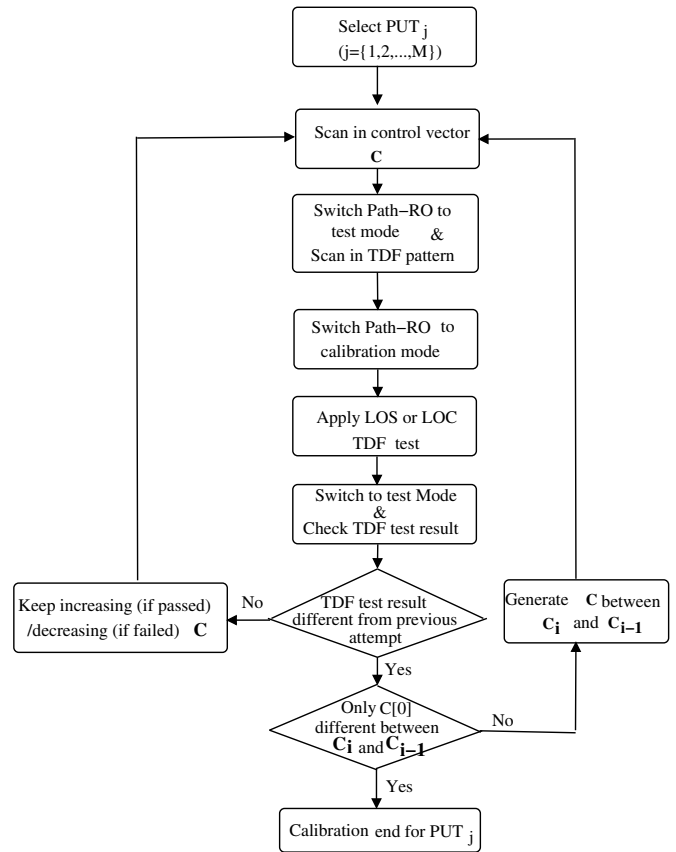

Fig. 5. Calibration flow.

clock-cycle long returning loop for all PUTs on one die. However, FR_{min} obtained from Equation 8 is over optimistic. Since process variations may increase or decrease t_{MC_max}/t_{MC_min} as well as t_{rlw_max}/t_{rlw_min}, FR_{min} should be increased by a factor of x ($x > 1$) to $FR = x \cdot FR_{min}$. x can be based on the severity of process variations in each technology node. It is expected to be larger in lower technologies.

• **Step 2:** In this step, we estimate the number of stages (K) for calibration circuit. Assume that FR is the desired flexible range, r_c is the calibration resolution, which is the minimum delay step increase or decrease when changing least significant bit in control vector C. K is the calibration circuit stage (i.e. the number of buffers) shown in Figure 2. The minimum number of calibration stages K_{min} which makes the calibration circuit to generate any flexible delay in the range of FR with r_c resolution is:

$$K_{min} = int[lg_2(\frac{FR}{r_c})] + 1 \qquad (10)$$

In our calibration circuit, the delay of buffer in $stage_i$ is twice the delay of buffer in $stage_{i-1}$. For FR = 5ns with r_c = 5ps, using Equation 10, K_{min} = 11. If in a design, $K \geq 11$ and the flexible delay of the last stage $t_{BUF0} \leq 5ps$, the flexible range and resolution can satisfy our goal.

V. CLFF AND CCFF DESIGN

Figures 6(a) and 6(b) show the structure of calibration launch flip-flop (CLFF) and calibration capture flip-flop (CCFF), respectively. The cells are designed to provide various operating modes for Path-RO architecture. To ensure PUTs can work in functional mode, test

mode, calibration mode and oscillation mode, small modifications are made in SFFs placed at the input and output of PUTs.
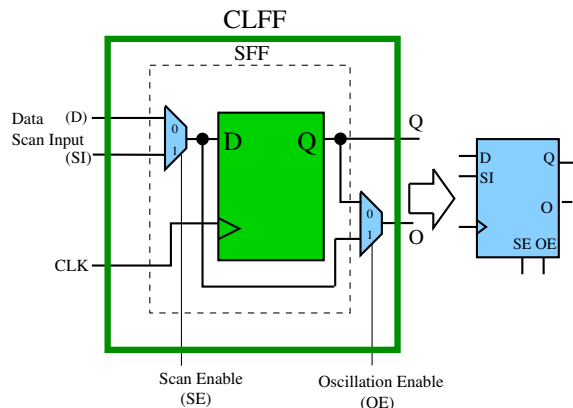
CLFF

(a) CLFF.
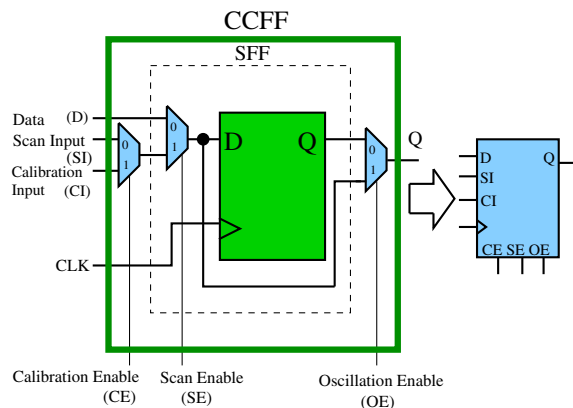
CCFF

(b) CCFF.

Fig. 6.   CLFF and CCFF.

CLFF: During calibration mode, the flip-flop at a PUT output operates as a transition launch flip-flop, therefore we call it calibration launch flip-flop (CLFF). As seen, CLFF is SFF plus a 2-to-1 Mux. Pin O, which is the output of the small Mux, connects to the returning loop. Hence, during oscillation mode signal from PUT can bypass the flip-flop and go out through pin O by setting SE to 0 and OE to 1. During calibration mode, the TDF pattern scanned into SFF can be applied to the returning loop through pin O by switching OE to 0. Since pin Q of CLFF directly connects SFF output to its connecting path, the speed of functional signal will not be affected.

CCFF: The flip-flop at the input of PUT is called calibration capture flip-flop (CCFF). In CCFF, pin CI connects to the returning loop. During calibration, CE and SE are set to 1. Therefore, the transition can be captured by the flip-flop. While during oscillation mode, SE, OE and CE are all set to 1 making the signal bypass the flip-flop.

The functional, calibration and oscillation signal paths are shown in Figure 7. From Figure 7, it can be seen that, the one output Mux in CLFF and two input Muxes in CCFF are included in the calibration signal path, therefore their delay will be included into the one clock cycle returning loop delay. Thus the process variations

TABLE I
CLFF AND CCFF OPERATION MODES

Mode	SE	CE	OE
Functional	0	0	0
Test Pattern Scan In/Out	1	0	0
Calibration	1	1	0
Oscillation	1	1	1

TABLE II
AREA OVERHEAD OF PATH-RO ARCHITECTURE

PUT no. M	K	n	Overhead
5	4	10	0.169%
10	5	10	0.199%
25	7	10	0.228%
50	8	10	0.288%
100	8	10	0.297%
200	8	10	0.311%
500	9	11	0.431%

on these components will be taken into account during calibration process. Also from Figure 7, it can be seen that the delay of PUT under functional mode will only increase by the small 2-to-1 Mux at pin Q in CCFF.

By offering different mode-control signals (scan enable (SE), calibration enable (CE) and oscillation enable (OE)), Path-RO can work in various modes. The control signal combinations and their representing operation modes are shown in Table I.

VI. RESULTS

In this section, on-chip path delay under different process variations will be simulated and measured. During the *simulation* we measure the delay of each path under test (PUT) from start-point SFF to its end-point SFF. However, during *measurement* we use Path-RO architecture to create oscillator for each PUT and use counter value (N) to measure path delay ($t_{path} = \frac{T}{2N} - t_{clock}$). To present the effectiveness of the proposed architecture, we have implemented Path-RO on b19 benchmark which contains 231K gates and 7K flip-flops. 180nm technology node [20] is used in implementation and the design was timing closed for 141MHz frequency. Path-ROs are built on the critical paths for b19 benchmark with different start-points and end-points located at different sites on physical layout.

Since there is an additional Mux in CCFF, which is unavoidable during functional mode, the delay overhead on critical path is about 0.1ns accounting for 1.4% of the top critical path delay, which is 7.02ns. Table II shows Path-RO's area overhead for various Ms (number of PUTs), Ks (number of stages in calibration circuit), and n (size of counter). As seen, the area overhead is extremely low even for large number of PUTs. Note that this ratio is expected to decrease as circuit size increases.

Figure 8 shows 50 path delay measurement results (using Parh-RO) and simulation results (using Hspice Monte Carlo process) for one of the most critical paths in b19 benchmark when considering intra-die process variations only. When using Path-RO, the process variations are applied to the paths and the content of counter is analyzed to measure the actual path delay (t_{path}). Note that the length of the path without considering process variations is 6.80ns. The variations considered during measurements and Monte Carlo simulations are:
- 10% intra-die, 3 sigma variation for Transistor Channel Length L.
- 30% intra-die, 3 sigma variation for Threshold Voltage V_{th}.
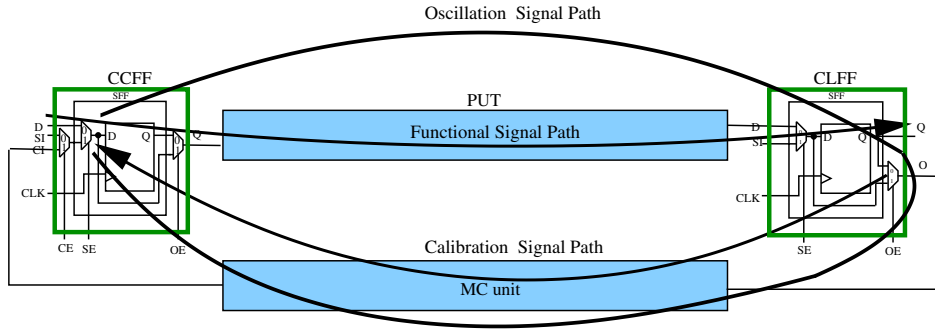- 3% intra-die, 3 sigma variation for Oxide Thickness T_{ox}.

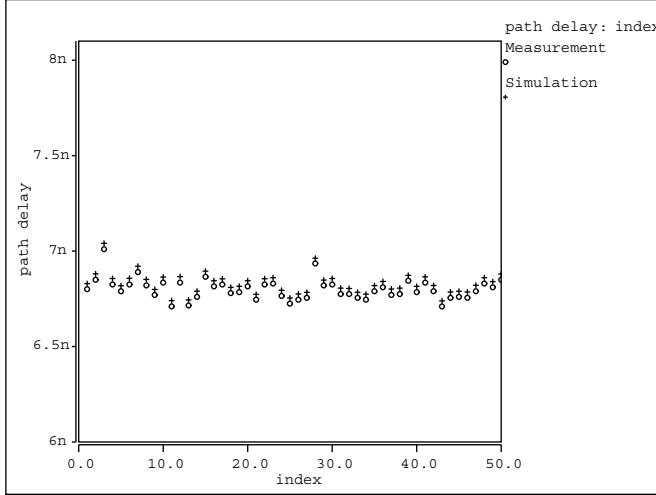Fig. 7. The functional, calibration and oscillation signal paths.

Fig. 8. Path delay for a selected critical path in b19 when only intra-die process variations are considered (50 Monte Carlo iterations).
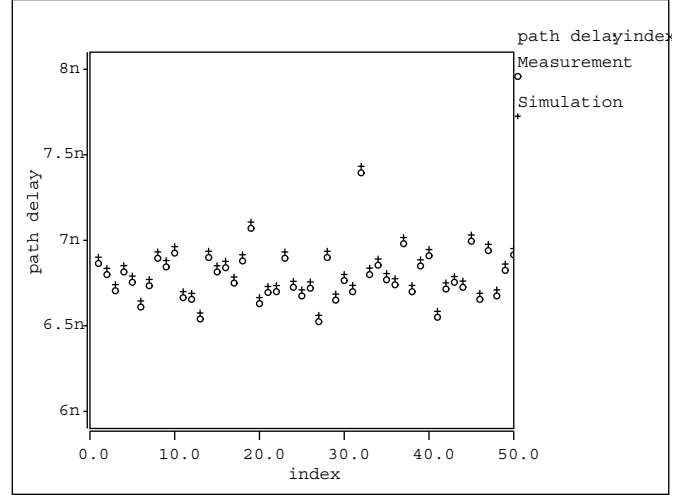

Fig. 9. Path delay for the same critical path in b19 when considering 50% intra-die and inter-die process variations (50 Monte Carlo iterations).

Figure 9 shows 50 path delay measurement and simulation results for the same path when half intra-die and half inter-die process variations are considered. The variations considered during simulations are:

- 5% intra-die, 5% inter-die, 3 sigma variation for L.
- 15% intra-die, 15% inter-die, 3 sigma variation for V_{th}.
- 1.5% intra-die, 1.5% inter-die, 3 sigma variation for T_{ox}.

The single point in Figure 9 far from others represents the case when all variations happen to worsen path delay. As seen in Figure 9, the half intra/inter-die process variations case has a nearly four times larger delay variation range than the intra-die only case. This is because the intra-die process variations have random negative or positive effect on each single gate delay, and may weaken each other when all gate delays are added up to generate path delay. However, inter-die process variations have almost same effect on every gate, which leads to accumulating impacts on path delay and making the delay variation range larger.

Both figures show that the measurement and simulation results are very close under the same process variations. However, the measured path delay for each variation in L, T_{ox} and V_{th} is slightly shorter than the path delay obtained from simulation which is in the range of resolution r_o set for Path-RO architecture.

Table III shows the results obtained from Path-RO (i.e. measurement) and simulation for 10 PUTs under various process variations. We have considered these 10 critical paths such that they have different starting points selected by timing analyzer. As seen, in both cases, the results obtained from the measurement and simulation are

very close. Comparing the results of the two cases, intra-die process variation have shown either positive or negative effect on individual gate delay on a path, however the inter-die process variations have same effect on every gate therefore, same percentage of inter-die process variations will introduce more impact on path delay. In this design, given the same oscillation measurement time $T = 9.8\mu s$ for all 10 Path-ROs, Case1 has led to at least 40 variations on counter value N, while Case2 has induced at least 224 variations on N. Note that different process variations can also be considered on corresponding gates in case of inter-die variations, which is part of our future work.

VII. CONCLUSION

This paper has presented a novel on-chip path delay measurement methodology called Path-RO. First, the design of Path-RO architecture which is of high accuracy and low-overhead was presented. The proposed architecture can be fully synthesized and be integrated into current DFT flow. To achieve high path delay measurement accuracy, a time-efficient and process-variation-robust calibration process aiming at eliminating the returning loop delay from measured oscillation period was then proposed. We also generate an on-chip at-speed path delay measurement flow basing on Path-RO. Measurement and simulation results show different combinations of intra/inter-die process variations' impact on critical path delay, and verify accuracy and timing efficiency of the proposed method as well.

TABLE III

THE IMPACT OF PROCESS VARIATIONS ON PATH DELAY (MEASUREMENT VS. SIMULATION)

PUT	Case 1: Intra Dominant		Case 2: Half Intra/Inter	
	Simulation	Measurement	Simulation	Measurement
1	7.076-7.315	7.025-7.265	6.575-7.781	6.525-7.735
2	6.758-7.050	6.710-7.010	6.575-7.454	6.525-7.795
3	6.028-6.280	5.980-6.240	5.744-6.734	5.710-6.685
4	5.972-6.300	5.935-6.255	5.645-6.734	5.605-6.680
5	5.954-6.180	5.925-6.105	5.365-6.620	5.325-6.575
6	5.883-6.180	5.850-6.145	5.486-6.688	5.450-6.650
7	5.795-5.972	5.750-5.935	5.278-6.467	5.245-6.430
8	5.778-6.084	5.750-6.050	5.548-6.488	5.510-6.450
9	5.264-5.425	5.230-5.385	4.969-5.694	4.939-5.650
10	4.531-4.729	4.516-4.658	4.141-5.021	4.125-4.995

REFERENCES

[1] G. Declerck, "A look into the future of nanoelectronics," in Proc. *2005 Symposium on VLSI Technology*, pp. 6 - 10, 2005.

[2] H. Balachandran, K. Butler N. Simpson, "Facilitating Rapid First Silicon Debug," in Proc. *ITC'02*, pp. 628-637, Oct 2006.

[3] R. Datta, A. Sebastine, A. Raghunathan and J. Abraham, "On-Chip Delay Measurement for Silicon Debug," in Proc. *GLSVLSI'04*, pp. 145-148, Apr 2004.

[4] R. Datta, G. Carpenter, K. Nowka and J. Abraham, "A scheme for on-chip timing characterization" in Proc. *VTS'06*, pp. 24-29, Apr 2004.

[5] S. Ghosh, S. Bhunia, A. Raychowdhury and K. Roy, "A Novel Delay Fault Testing Methodology Using Low-Overhead Built-In Delay Sensor," *IEEE Transactions on Computer-Added Design of Integrated Circuits and Systems*, vol. 25, no. 12, pp. 2934-2943, 2006.

[6] A. Ghosh, R. Rao, C. Chuang and R. Brown, "On-Chip Process Variation Detection and Compensation using Delay and Slew-Rate Monitoring Circuits," in Proc. *ISQED'08*, pp. 815-820, Mar 2008.

[7] C. Su, Y. Chen, M. Huang, G. Chen and C. Lee, "All Digital Built-in Delay and Crosstalk Measurement for On-Chip Buses," in Proc. *DATE'00*, pp. 527-531, Mar 2004.

[8] M. Orshansky, S. Nassif, and D. Boning, "Design for manufacturability and statistical design," *Springer*, USA, 2007.

[9] M. Bhushan, A. Gattiker, M. Ketchen and K. Das, "Ring Oscillators for CMOS Process Tuning and Variability Control," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 1, pp. 10-17, 2006.

[10] M. Nourani and A. Radhakrishnan, "Testing On-die Process Variation in Nanometer VLSI," *IEEE Design & Test of Computers*, pp. 438-451, Nov-Dec 2006.

[11] Z. Abuhamdeh and A. Crouch and J. Remmers, "A Production IR-Drop Screen on a Chip," *IEEE Design & Test of Computers*, pp. 216-224, May-Jun 2007.

[12] K. Arabi, H. Ihs, C. Dufaza and B. Kaminska, "Dynamic digital integrated circuit testing using oscillation-test method," *Electronics Letters*, pp. 762-764, April 1998.

[13] W. Wu, C. Lee, M. Wu, J. Chen and M. Abadir, "Oscillation ring delay test for high performance microprocessors" *Journal of Electronic Testing*, vol 16, no. 1-2, pp. 147-155, Feb. 2000.

[14] S. Paul, S. Krishnamurthy, H. Mahmoodi and S. Bhunia, "Low-overhead design technique for calibration of maximum frequency at multiple operating points," in Proc. *ICCAD'07*, pp. 401-404, Apr 2004.

[15] A. Chan and G. Roberts, "A synthesizable, fast and high-resolution timing measurement device using a component-invariant vernier delay line" in Proc. *ITC'01*, pp. 858-867, 2001.

[16] Synopsys Inc., "User Manuals for SYNOPSYS Toolset Version 2007.03," Synopsys, Inc., 2007.

[17] http://crete.cadence.com, "0.18m standard cell GSCLib library version 2.0," Cadence, Inc., 2005.

[18] B. Mark, G. Roberts, et al, "An introduction to mixed-signal IC test and measurement," *Oxford University Press*, USA , 2000.

[19] http://www.cerc.utexas.edu/itc99-benchmarks/bench.html

[20] http://crete.cadence.com, 0.18m standard cell GSCLib library version 2.0, Cadence, Inc., 2005.