

# Reactive Clocks with Variability-Tracking Jitter

Jordi Cortadella\* Luciano Lavagno† Pedro López\* Marc Lupon\* Alberto Moreno\* Antoni Roca\* Sachin S. Sapatnekar‡  
\*Department of Computer Science, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain.

Email: {jordicf, plopez, mlupon, albertomv, aroca}@cs.upc.edu

†Department of Electronics, Politecnico di Torino, 10129 Torino, Italy. Email: luciano.lavagno@polito.it

‡Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA. Email: sachin@umn.edu

**Abstract**—The growing variability in nanoelectronic devices, due to uncertainties from the manufacturing process and environmental conditions (power supply, temperature, aging), requires increasing design guardbands, forcing circuits to work with conservative clock frequencies. Various schemes for clock generation based on ring oscillators and adaptive clocks have been proposed with the goal to mitigate the power and performance losses attributable to variability. However, there has been no systematic analysis to quantify the benefits of such schemes and no sign-off method has been proposed for timing correctness. This paper presents and analyzes a *Reactive Clocking* scheme with *Variability-Tracking Jitter* (RCIk) that uses variability as an opportunity to reduce power by continuously adjusting the clock frequency to the varying environmental conditions, and thus, reduces guardband margins significantly. Power can be reduced between 20% and 40% at iso-performance and performance can be boosted by similar amounts at iso-power. Additionally, energy savings can be translated to substantial advantages in terms of reliability and thermal management. More importantly, the technology can be adopted with minimal modifications to conventional EDA flows.

## I. INTRODUCTION

It is widely recognized that the ultimate limit to Moore’s law is not technology but economics. Every generation requires an enormous increase in the non-recurring engineering (NRE) and fabrication costs. As a result, the cost per transistor, which had been decreasing at every technology node for several decades, may have been increasing over the last few nodes [1].

Even in the past decade, purely geometrical scaling has been limited by physical challenges and lithography issues. As the supply voltage has stagnated, enhanced performance has been enabled by the notion of *equivalent-scaling* [2] in the International Technology Roadmap for Semiconductors, whereby clever “tricks” have been used to achieve better performance at the next node.

Further, the benefits of scaling are showing diminishing returns. While moving to a new technology node implies a 3–4× increase in manufacturing cost, key metrics such as speed, power, and density are only confined to a 20–20–20% improvement, respectively [3]. The modest progress in performance metrics is largely determined by *variability*: the increasing gap between worst-case and nominal delays that must be covered by conservative guardband margins.

Today’s methodologies actively fight off variability. Leading-edge designs use low-variability phase-locked loops (PLLs) for low-jitter clocks and near-zero-skew trees for clock distribution. They employ a strict discipline to maintain the rigidity of timing boundaries, with conservative guardbands used for each combinational block to ensure that these rigid boundaries are not violated. As the magnitude of variability increases, such guardbands incur prohibitive overheads.

This paper presents *Reactive Clocks with Variability-Tracking Jitter* (RCIk), a novel design-based equivalent scaling paradigm that embraces dynamic variability instead of fighting against it. This new

\*This work has been partially supported by funds from the Spanish Ministry for Economy and Competitiveness and the European Union (FEDER funds) under grant TIN2013-46181-C2-1-R, the Generalitat de Catalunya (2014 SGR 1034 and FI-DGR 2015) and a Fulbright award.

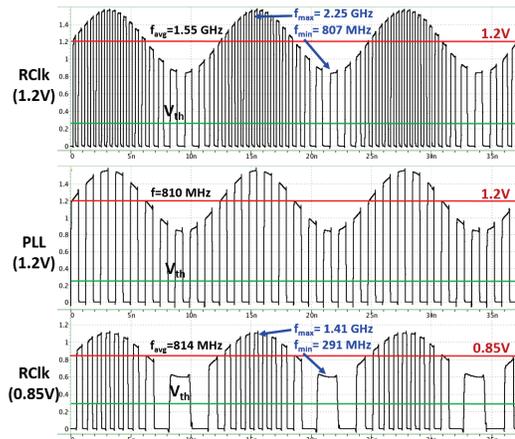


Fig. 1. Clock generation with PLL and RCIk.

method leverages common-mode variability between the circuit and the clock source and develops a clocking scheme that is an innovative alternative to PLL-based approaches. This clocking scheme, coupled with a chip-wide design methodology, mitigates the margins required to tolerate dynamic variability, thus reducing the overall chip power.

Unlike the classical goal of attenuating jitter to provide more robust clock generators, the proposed approach *intentionally generates jitter that closely tracks logic delay variations* to accommodate dynamic delay variations. Note that traditionally jitter is minimized because it is *assumed to be uncorrelated* with the delay of the combinational logic. We proposed to *maximize this correlation*, and use it to improve performance or reduce power. Fig. 1 illustrates the effect of using reactive clocks to reduce margins. The waveforms have been obtained with SPICE simulations and show the clock signal arriving at the flip-flops with a power supply fluctuating with  $\pm 30\%$  noise. Although this is larger than the voltage droops seen in typical systems, it helps to emphasize the benefits of this technology and could potentially model scenarios in an energy harvesting context.

The PLL (middle) maintains a conservative fixed frequency (810 MHz) to cover the delay variability of the circuit. In RCIk, however, the clock source suffers the same variability delay as the circuit and, as a result, it is able to instantaneously adjust its clock period according to operating conditions. By preserving the same nominal voltage (1.2V), it can achieve an average frequency of 1.55 GHz, ranging between 807 MHz and 2.25 GHz. This speed-up can be converted into power savings by scaling voltage. In the bottom waveform we can observe the clock signal working at 0.85V and maintaining an average frequency (814 MHz) similar to one of the PLL working at 1.2V. It is interesting to point out the very low frequency of the clock (291 MHz) when approaching  $V_{th}$  (0.28V).

The proposed technology is based on the exploitation of the sensitivity of ring oscillators to PVT variability. This idea has

TABLE I  
A TAXONOMY OF THE SOURCES OF VARIABILITY.

|        | Static | Slow (ms) | Fast (ns) |
|--------|--------|-----------|-----------|
| Global | PV     | VTA       | V         |
| Local  | PV     | VTA       | V         |

been tentatively explored in the past [4]–[6] but no approach with quantifiable benefits has been analyzed. RClk relies on a robust timing model for assembling a *suitable* Variability-Tracking Ring Oscillator (VTRO) that is used as a clock source (Section III). The paper also introduces a new methodology for designing and validating the clocking scheme using conventional sign-off procedures and a path synthesizer (Section IV). Power and performance benefits of RClk have been evaluated through electrical simulations (Section V) and an FPGA prototype (Section VI).

The main features of RClk can be summarized as follows:

- **Efficient:** 1.2–1.4× speed-up or 20%–40% power savings with regard to worst-case sign-off.
- **Not invasive:** the original circuit is not modified. RClk is an alternative to PLLs—both clocks can live together without any need to modify the clock tree.
- **Practical:** adoptable in commercial design flows with conventional sign-off procedures.
- **Reliable:** bringing substantial improvements in terms of thermal management and reliability as a byproduct of power reduction.

## II. VARIABILITY, MARGINS AND STATIC TIMING ANALYSIS

Among the different taxonomies for classifying the sources of variability, we select one that helps to easily identify the margins used for timing sign-off. Table I classifies variability according to two parameters: locality and variation speed.

In terms of locality, global variability affects all devices uniformly whereas local variability has a different impact for each device. Some elements of local variability (e.g., voltage or temperature) may exhibit spatial correlation, i.e., their impact may be similar for devices located in the same region. In terms of variation speed, we can distinguish between static and dynamic variability. Process (P) variability is always static and can be either global (systematic) or local (random). Temperature (T) and aging (A) have slow variability. Both sources have global and local variability components.

Voltage (V) has a diversity of variability components and deserves a special discussion. On the one hand, it has DC components produced by static IR drops that can be either global (off-chip resistance) or local (on-chip power delivery network). On the other hand, voltage variability also has AC components determined by the activity of the system. The largest components of voltage noise occur at middle and low frequencies, as can be observed from Intel’s Nehalem microprocessor data [7] or from the analysis of GPU architectures [8]. In multi-core architectures, first-order droops have an impact at the level of small clusters of cores. Second-order droops have a chip-wide impact across all cores of the die. In both cases, these droops are global at the level of individual cores, which is the level of granularity considered in this paper for RClk.

Timing sign-off must take into account all the possible sources of variability and add margins to cover them. Nowadays, the mechanisms to model variability during STA are the following: (1) *Library corners* to model global variability, (2) *On-Chip Variability* (OCV) derating factors to model local variability and (3) *Clock uncertainty* to model jitter and any other safety margin included to account for any uncovered variability (e.g., aging) or inaccuracy in the analysis.

The two bars shown in Fig. 2 depict a typical delay distribution for timing sign-off. Data has been obtained by simulation of a critical

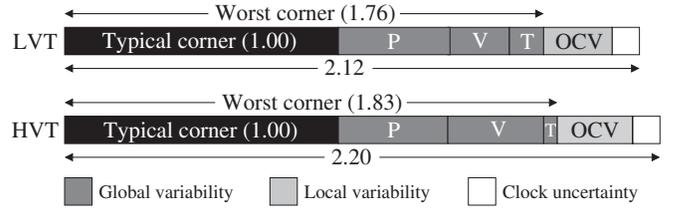


Fig. 2. Delay distribution for timing sign-off.

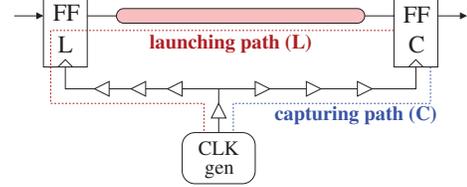


Fig. 3. Launching and capturing paths for timing sign-off.

path using SPICE models from a 65nm commercial library. The delays for two cell libraries are reported: Low- $V_t$  (LVT) and High- $V_t$  (HVT). For each library, the delays have been normalized to its typical corner (1.00ns for TT, 1.2V, 25°C; the delays for HVT are about 2× those for LVT) and the contributions of the process (P), voltage (V) and temperature (T) components have been estimated by means of SPICE simulations at the appropriate corners.

Global variability accounts for most of the guardband margins, with P and V being the dominating components. The worst corner covers the worst conditions for PVT global variability (e.g., SS devices, 1.08V, 125°C). Local variability (OCV) also requires a margin modeled as a derating factor that typically ranges between 5% and 15% in conventional sign-off (15% has been chosen in the example). Finally, some fixed margin is usually added for clock uncertainty and aging (0.10ns in the example). Overall, timing sign-off is done at more than 2× the delay of the typical corner.

We next review the basic timing analysis to check a setup constraint for a critical path that goes from flip-flop  $F_L$  to flip-flop  $F_C$  (see Fig. 3). Two competing paths are involved: launching ( $L$ ) and capturing ( $C$ ). For the circuit to operate correctly, the cycle period ( $P$ ) must be sufficiently long to meet the setup constraint for all pairs of launching/capturing paths (denoted by set  $LC$ ):

$$P - J > \max_{i \in LC} (L_i - C_i) \quad (1)$$

where  $J$  is the maximum clock jitter (that here is assumed to be uncorrelated with  $L_i$  and  $C_i$ ). Timing sign-off must guarantee that the clock frequency does not violate any timing constraint under any operating condition. In the presence of variability, margins have to be added to prevent timing failures, as follows.

During STA, global variability is modeled by library corners. Let us assume that we have a set of corners that cover different PVT configurations for devices (fast/typical/slow process, high/typical/low temperature, high/nominal/low voltage) and interconnect ( $RC_{max}$ ,  $RC_{min}$ , ...). Let us call  $K$  the set of corners. For every timing path  $p$ , we denote by  $p^k$  the delay of the path at corner  $k$ .

Constraint (1) can now be quantified for all corners and all pairs of paths to derive the minimum cycle period:

$$P - J > \max_{k \in K, i \in LC} (L_i^k - C_i^k) \quad (2)$$

Any clock period  $P$  satisfying (2) guarantees a correct behavior for all PVT corners considered for STA.

Local OCV is modeled by applying derating factors to the launching and capturing paths. These factors can be different for each corner.

Let us denote  $\delta_L$  and  $\delta_C$  the derating factors applied to launching and capturing paths, respectively. Typically,  $\delta_L \geq 1$  and  $\delta_C \leq 1$ . When incorporating local variability, the setup constraint (2) is as follows<sup>1</sup>:

$$P - J > \max_{k \in K, i \in LC} (\delta_L L_i^k - \delta_C C_i^k) \quad (3)$$

### III. REACTIVE CLOCKS WITH VARIABILITY-TRACKING JITTER

Various techniques have been proposed to mitigate the impact of variability. Parametric binning [9] performs at-speed testing to eliminate margins associated with global static variability, while Adaptive Clocks (AClk) attack dynamic variability by modifying the clock frequency when sensing changes in the operating conditions [7], [10], [11]. Unfortunately, the aforementioned techniques cannot get rid of some guardband margins because they cannot handle fast variability efficiently—more details are provided in Section VIII.

Several studies pointed out the ability of ring oscillators to react immediately and without requiring feedback control to any source of variability, and even some proposals have suggested to incorporate them as clock generators [4]–[6]. Nonetheless, none of the previous schemes described how ring oscillators should behave or which constraints they must satisfy. The following section introduces a timing model that sets up the foundations of RClk, a novel scheme that benefits from having an *appropriate* ring oscillator as a clock source that responds to variations likewise the rest of the circuit logic.

#### A. Timing model for Reactive clocks

Because of the sensitive nature of ring oscillators, a new sign-off method must be formalized. The next paragraphs describe a timing model that certifies the soundness of the ring oscillator and proves its correct functionality when it oversees a circuit.

Previous STA models assume that the period  $P$  is obtained from a PLL with a fixed frequency. PLLs are attractive because they can sustain the same frequency even in the presence of variability, and hence they cannot adapt to it. Ring oscillators are often shunned because they supposedly have a large jitter. However, even at the core of a PLL, there is a Voltage Controlled Oscillator (VCO) built with logic gates (e.g., current-starved inverters which are used to control its frequency). These gates will suffer from the same variation sources as the ring oscillator that we discuss here, but the resulting jitter will be *minimized* rather than *exploited*. Moreover, they will have similar sources of noise, also resulting in unwanted jitter, as the ring oscillator that we use.

Let us now assume that the clock generator is designed as an oscillator using the same type of components as the ones used for the combinational logic and clock trees (e.g., logic gates and buffers). Let us also assume, for the sake of simplicity, that the delays of all components in the circuit scale uniformly with voltage and temperature<sup>2</sup>. In this case, the period of the clock would naturally adapt to the process corner and operating conditions of the circuit.

Fig. 4 depicts a symbolic representation of the components affected by variability when using RClk. The horizontal dimension represents time. The top and bottom paths represent the launching and capturing paths, respectively. The launching path includes the clock tree (shaded) and the critical path delay (white) from flip-flop  $F_L$  to flip-flop  $F_C$  (flip-flops are assumed to have zero delay in this model). The capturing path includes the delay of the ring oscillator (white)

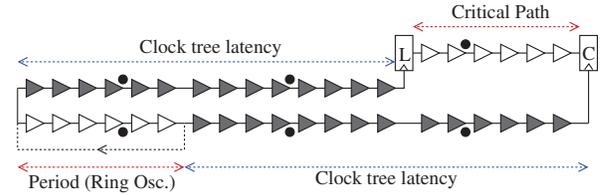


Fig. 4. Symbolic timing model for RClk.

and the clock tree (shaded). The paths in the model are equivalent to the ones shown in Fig. 3, explicitly substituting the clock generator by a ring oscillator. The bullets in the diagram represent signal pulses flying in the launching and capturing paths. In general, the clock tree may contain several flying pulses. Let us assume that the top and bottom bullets are perfectly aligned under the absence of variability and that all components have the same delay  $d$ .

Let us call  $P$  the time separation (period) between consecutive bullets. With this assumption, an infinite stream of pairs of bullets will arrive synchronized at flip-flop  $F_C$  every  $P$  time units. Now assume that voltage drops to a point in which all components are slowed down by a factor  $s$ , i.e., every component has delay  $s \cdot d$ . Then, the time separation between bullets (period) will be increased to  $s \cdot P$  but the bullets would still be perfectly aligned in time, i.e., all the bullets will run in slow motion but at the same speed. It is important to notice that the alignment/misalignment of the bullets is independent from the clock tree latency in this model, as clock pulses can amend the variability delay when traveling in the clock distribution network—a phenomenon known as *clock-data compensation* [12].

The model shows how *margins for global variability can be eliminated when using ring oscillators and only margins for local variability are required*. Using STA terminology, *the ring oscillator transforms the clock-based setup/hold checks into data-to-data checks* (also called zero-cycle checks), as described in Sect. 10.3 of [13]. The ring oscillator is just a component of one of the competing paths in the data-to-data checks. Henceforth, timing analysis with ring oscillators can be done with existing timing checks in conventional STA tools.

Let us now study how this effect can be formally modeled in terms of STA constraints. By using a ring oscillator, a different cycle period  $P^k$  is generated at every corner  $k$ . The setup constraint can now hold at every corner  $k$  with a different period  $P^k$ , i.e.,

$$\forall k \in K, i \in LC : \delta_L L_i^k < \delta_C (P^k + C_i^k) \quad (4)$$

where the term  $J$  (jitter) has been removed. The reason is because the portion of the fluctuations of the ring oscillator that is different from those of  $L_i^k$  and  $C_i^k$  is accounted as local variability, by applying the derating factor  $\delta_C$  to  $P^k$ . The previous inequality can be rewritten as follows:

$$\forall k : P^k > \max_{i \in LC} \left( \frac{\delta_L L_i^k}{\delta_C} - C_i^k \right) \quad (5)$$

A fundamental difference with (3) is that *a different clock period  $P^k$  is obtained at every corner*. This means that the causes of delay change in the clock generator (ring oscillator) and the circuit are *exploited, as long as they are correlated*, instead of being minimized in the clock generator (PLL VCO) and taken as margin in the circuit. This allows us to reduce margins substantially, as will be shown in the experiments.

#### B. Variability-Tracking Ring Oscillators

Let us now define *variable-tracking* ring oscillators (VTRO): a structure composed of a closed chain of different types of logic gates that is built upon the rules established in (5). The VTRO must fulfill

<sup>1</sup>For simplicity, we assume the same derating factors for all corners.

<sup>2</sup>These assumptions are only made to simplify the conceptual discussion about RClk and global variability. All deviations from this assumption, including those due to gate versus wire delays, are included into the derating factors used in constraints (3).

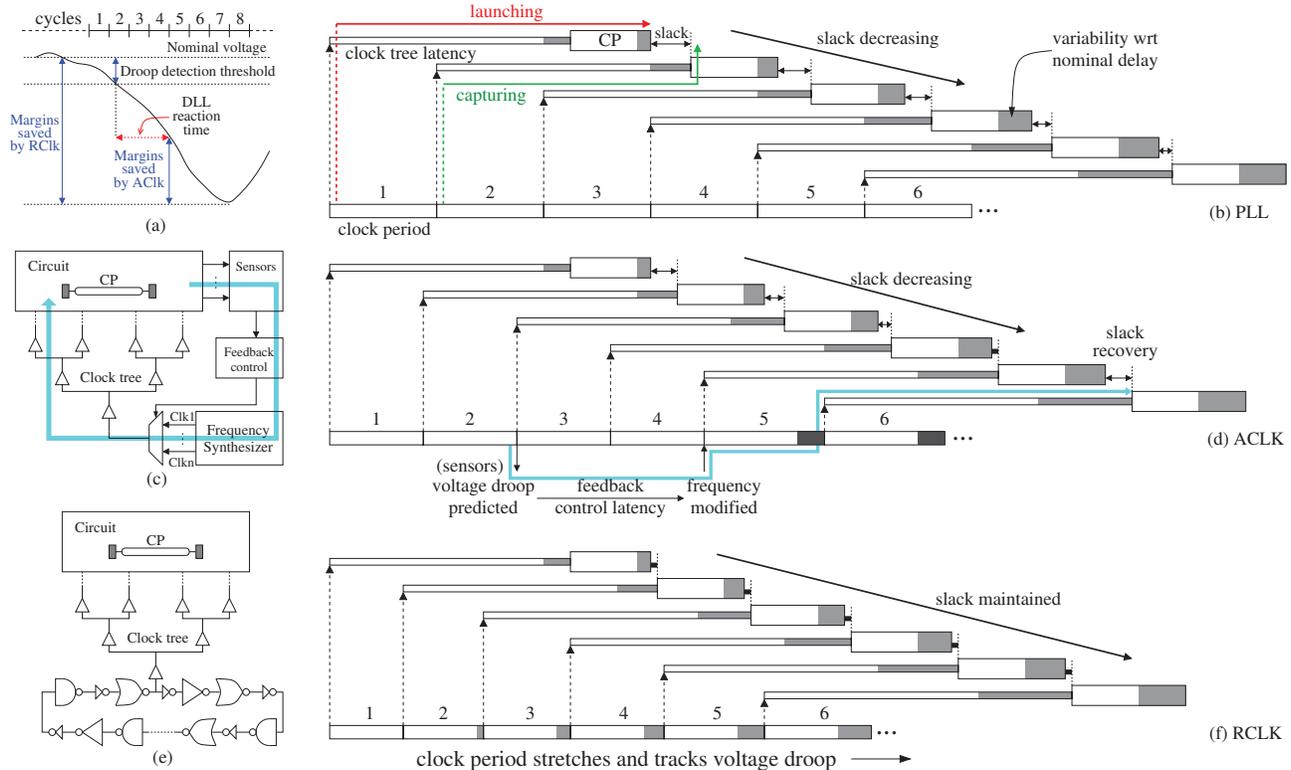


Fig. 5. Effects of a voltage droop on three clocking schemes: PLL, ACIk and RClk

the following properties, which can be certified using conventional sign-off methods:

- its clock period almost fits but is somewhat slower than any flop-to-flop path of the circuit at any operating condition.
- its oscillating pulses must correct the delay deviation introduced by the clock tree and design unknowns (e.g., on-chip variability).
- it can be built through methodologies that mix circuit timing analysis and algorithms that explore the delay produced by different types of gates.
- its timing correctness can be validated using EDA tools.
- its integration should be non-invasive and smooth.

A VTRO can be physically located at any place within the clock domain and, unlike PLLs, it suffers from the same global variability as the circuit, as detailed in Fig. 1. Centering the VTRO in the middle of the layout permits RClk to track fast variability more locally and minimize clock tree correction factors. The proposal is non-intrusive in the sense that it does not interfere with the circuit design flow, as the VTRO can be physically implemented separately from the logic circuit. Moreover, the VTRO is treated as an additional clock source, and thus, it can be multiplexed with a PLL. The only required margins that the VTRO must cover are those dedicated to the local variations between the critical paths, the clock tree and the ring oscillator. RClk can be applied to any clock domain. Each clock domain will have to be isolated with clock-domain crossing (CDC) to prevent synchronization errors with the neighboring domains.

### C. Clocking scheme's response to dynamic variability

Fig. 5 shows how three different clock sources (PLL, ACIk and RClk) respond to a voltage droop as characterized in Fig. 5(a). The timing diagrams (b), (d) and (f) in Fig. 5 illustrate the delays of the launching (red) and capturing (green) paths in different scenarios. The

rectangles at the base represent clock periods. The thin rectangles  $\square$  represent the delay of the clock tree, which is longer than 2 clock periods in the example. The thick rectangles  $\square$  represent the maximum delay of the critical paths (CP) in the circuit. The delays of the clock tree and critical paths have been split into two components  $\square$ : nominal delay  $\square$  and extra delay produced by the voltage droop  $\square$ . Both the CP and the clock tree suffer from extra delays as voltage drops. Slack is measured as the difference between the launching path and the capturing path, and is represented by  $\leftrightarrow$ .

**PLL:** Fig. 5(b) shows a rigid clock that is agnostic to variability. The clock period must include sufficient slack to cover the most adverse operating conditions. As a result, the PLL delivers a period that is far from optimal in predominant scenarios.

**ACIk [7]:** to save guardband margins, ACIk implements the feedback mechanism depicted in Fig. 5(c). As shown in Fig. 5(d), sensors detect the droop when voltage goes below a certain threshold (cycle 2). Once the droop has been detected, some control logic must be activated to modify the clock frequency (DLL reaction time). A new, longer period is then generated (cycle 5) and the slack is recovered after the new clock pulses traverse the clock tree (blue line). Nonetheless, the slack decreases while the new clock period is not issued, meaning that some margins are still required. Thus, ACIk is penalized by the latency of sensing voltage droops and selecting a clock period that is a better choice for the new operating conditions.

**RClk:** this clocking scheme attaches a VTRO to the clock tree as detailed in Fig. 5(e), stretching the clock period when the voltage drops. As displayed in Fig. 5(f), the three important components that determine the slack (critical path, VTRO and clock tree) are affected by the voltage droop in unison. This means that all delays are stretched simultaneously, thus maintaining the slack between the launching and capturing paths along all cycles. Henceforth, the slack

is preserved even in the presence of fluctuations, and thereby the clock period of RClk is close to optimal in all operating conditions.

Note that variability does not only alter critical paths, it also affects the in-flight clock pulses because clock-data compensation takes place in *all* the scenarios of Fig. 5. This phenomenon allows the clock pulses that are traveling in the clock tree to modify their period when operating conditions shift, preventing timing violations in the critical paths. Experiments presented in Section V confirm that RClk is agnostic to clock tree latency, as clock pulses generated by the VTRO readjust their delay when traversing the clock tree.

#### IV. IMPLEMENTING A RELIABLE REACTIVE CLOCK

If we compare the top and bottom paths of Fig. 4, we observe the need to match the delay of the VTRO with that of the critical path. However, the critical path of the picture is just an abstraction of the multiple critical paths that may determine the clock period under different operating conditions. It is worth noticing that the VTRO design goes beyond a critical path replica (CPR [14]) implementation. While a CPR just mirrors the delay of a single circuit's path by replicating most of its logic, the VTRO reproduces the *slowest* delay of *any* critical path of the circuit. This can be archived because the VTRO implementation is independent from the internal configuration of the critical paths, and therefore the concatenation of logic gates that it uses may be different from the ones used in circuit's paths.

##### A. Method for generating Variability-Tracking Ring Oscillators

A circuit that operates with RClk must employ a clock source that satisfies equation (5). One option is to design an *ad hoc* ring oscillator that is manually adjusted to the critical paths of the circuit at different operating conditions. However, an automated solution is desired for a practical design flow.

Fortunately, the task of creating VTROs can easily be integrated in conventional EDA tools. It is enough to implement a method which includes a path synthesizer that generates a chain of logic gates that closely matches the delay of the circuit under different operating conditions—the path synthesizer shares similarities with the one in [15]. The method must cover the following steps:

- 1) use all the PVT corners available for STA to calculate the clock period of the circuit at the corresponding conditions.
- 2) apply OCV derating factors and other margins (design unknowns, clock tree variability correction) to obtain the minimum value for  $P^k$  at each corner according to constraint (5).
- 3) synthesize a path for the VTRO that meets  $P^k$  requirements.
- 4) run standard STA after the synthesis of the VTRO to perform sign-off at every library corner as defined in equation (5).

A path synthesizer has been designed to explore different chains of gates and closely match the different values of  $P^k$  at each library corner. The core of the path synthesizer heuristically solves a combinatorial problem that selects a suitable mix of library cells. The Nonlinear Delay Model (NLDLM) is used to estimate the delay of the path. The search is guided by a cost function consisting of a weighted sum of the delay differences between the synthesized path and  $P^k$  at each corner. Further details of the path synthesizer are out of the scope of the paper.

##### B. A Variability-Tracking Ring Oscillator for an AES module

To demonstrate the effectiveness of the previous method, we synthesized an AES encryptor module [16] with the Synopsys Design Compiler<sup>®</sup> and a 65nm low- $V_t$  commercial library. This module is coupled to a VTRO generated according to the method described above, which was synthesized using gates from the same cell library.

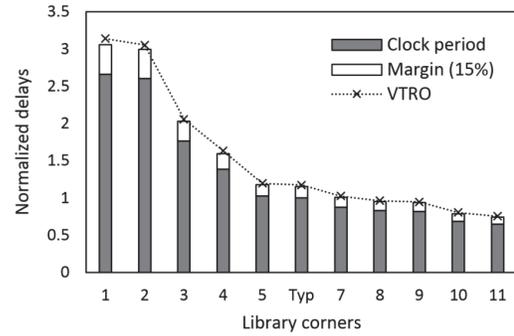


Fig. 6. VTRO delay covering PVT variations of AES module critical paths.

TABLE II  
CELL COMPOSITION OF THE VTRO USED IN THE AES MODULE.

| Cell         | Driving strength |           |          |          |          |          |          |          | Total    |          |           |
|--------------|------------------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
|              | 0                | 1         | 2        | 3        | 4        | 5        | 8        | 10       |          | 12       | 26        |
| CKINV        | -                | 6         | -        | 2        | -        | -        | -        | -        | -        | -        | 8         |
| ND2          | 3                | 2         | -        | -        | -        | 1        | -        | -        | -        | -        | 6         |
| INV          | -                | -         | -        | -        | -        | -        | 2        | 1        | 1        | -        | 4         |
| CKBUF        | -                | 3         | -        | -        | -        | -        | -        | -        | -        | -        | 3         |
| AOI21        | -                | -         | -        | -        | 1        | -        | 1        | -        | -        | -        | 2         |
| AOI31        | 2                | -         | -        | -        | -        | -        | -        | -        | -        | -        | 2         |
| ND3          | -                | -         | -        | 2        | -        | -        | -        | -        | -        | -        | 2         |
| BUF          | -                | -         | -        | -        | -        | -        | -        | -        | -        | 1        | 1         |
| MAOI2223     | -                | -         | 1        | -        | -        | -        | -        | -        | -        | -        | 1         |
| MXB3         | -                | 1         | -        | -        | -        | -        | -        | -        | -        | -        | 1         |
| OAI2113      | -                | 1         | -        | -        | -        | -        | -        | -        | -        | -        | 1         |
| <b>Total</b> | <b>5</b>         | <b>13</b> | <b>1</b> | <b>4</b> | <b>1</b> | <b>1</b> | <b>3</b> | <b>1</b> | <b>1</b> | <b>1</b> | <b>31</b> |

Synopsys PrimeTime<sup>®</sup> was used to perform multi-corner sign-off and extract delays related to the AES module and the ring oscillator.

Fig. 6 illustrates the delay constraints of the AES module and the delay of the VTRO generated by the path synthesizer. The discrete set of points on the horizontal axis represent PVT corners (in decreasing order of delay). The delay has been normalized to the typical corner (labeled Typ). Dark bars (Clock Period) represent critical path constraints extracted directly from PrimeTime<sup>®</sup>. Each bar connects the points of an  $(L_i, C_i)$  pair that may have a different sensitivity to PVT variations. White bars (Margin) add a fixed margin (15%) on top of  $(L_i, C_i)$  pairs as defined by (5). This margin includes OCV, design unknowns, clock uncertainties, and a small guardband to correct the imbalance between the effects of variability in the VTRO and in the clock tree.

The delay extracted by STA for the ring oscillator is displayed with crosses in Fig. 6 (VTRO). As it can be seen, it is very close to the minimum value of  $P^k$  as reported by equation (5). The average difference between the VTRO and the minimum value for  $P^k$  is around 1.6%, with a maximum difference of 2.6% and a minimum difference of 0.7% in certain library corners.

Table II shows the types of gates that are embodied in the VTRO generated by the path synthesizer. First, we can observe that the ring oscillator includes cells of different kinds and driving strengths. This makes the VTRO robust even when encountering million-path circuits that exhibit different sensitivities to VT fluctuations. Second, from a total of 31 gates, 11 are optimized for the clock tree (e.g., CKINV). Thus, the path comprised within the VTRO contains logic gates that react to variability much alike both the critical paths and the clock tree.

#### V. EXPERIMENTAL RESULTS

The benefits of the scheme presented in this paper have been evaluated through electrical simulations, which test the timing of the digital circuit described in Section IV-B. Synopsys PrimeTime<sup>®</sup> was

TABLE III  
F<sub>AVG</sub> (GHZ) FOR PVT PARAMETERS USING PLL, AClk AND RClk.

| Process variability → |       | Typical (TT) |             |             | Worst (SS)  |             |             |
|-----------------------|-------|--------------|-------------|-------------|-------------|-------------|-------------|
| Voltage               | Temp. | PLL          | AClk        | RClk        | PLL         | AClk        | RClk        |
| 1.2V                  | 25°C  | 1.59         | 1.59        | 1.56        | 1.22        | 1.22        | 1.21        |
|                       | 75°C  | 1.48         | 1.48        | 1.46        | 1.13        | 1.13        | 1.13        |
|                       | 125°C | <b>1.40</b>  | 1.40        | <b>1.38</b> | 1.08        | 1.08        | 1.08        |
| 1.2V ± 10%            | 25°C  | 1.36         | 1.39        | 1.56        | 1.00        | 1.03        | 1.21        |
|                       | 75°C  | 1.27         | <b>1.30</b> | <b>1.46</b> | 0.94        | <b>0.97</b> | <b>1.13</b> |
|                       | 125°C | <b>1.22</b>  | 1.23        | 1.38        | <b>0.91</b> | 0.93        | 1.08        |
| 1.2V ± 20%            | 25°C  | 1.12         | 1.17        | 1.54        | 0.80        | 0.84        | 1.20        |
|                       | 75°C  | 1.06         | 1.10        | 1.44        | 0.76        | 0.79        | 1.13        |
|                       | 125°C | 1.01         | 1.05        | 1.37        | 0.75        | 0.76        | 1.07        |
| 1.2V ± 30%            | 25°C  | 0.85         | 0.88        | 1.51        | 0.60        | 0.61        | 1.18        |
|                       | 75°C  | 0.84         | 0.86        | 1.42        | 0.58        | 0.59        | 1.11        |
|                       | 125°C | <b>0.82</b>  | 0.84        | <b>1.35</b> | 0.55        | 0.57        | 1.06        |

used to generate a SPICE netlist including the top 5 critical paths of the AES module and the VTRO. Those paths were totally disjoint and they were obtained from different library corners. This is a good trade-off between selecting representative timing paths and making the SPICE simulations affordable. The simulations were customized to toggle the inputs of the launching flip-flops at every cycle. Global voltage variations were modeled by applying sinusoidal fluctuations with different amplitudes at frequencies fully misaligned with the clock frequency, as shown in Fig. 1. Voltage fluctuations reflected the behavior of first-order droops, following a 200 MHz waveform as suggested in [7]. No local variability was assumed in the simulations.

#### A. Rigid clocks vs. Reactive clocks

Table III reports the maximum (PLL and AClk) and the average (RClk) frequency achieved without timing violations at each PVT corner. To evaluate the benefits of RClk with regard to PLL, different scenarios must be considered. We assume that the chip would work at a nominal voltage with ±10% fluctuations and at an average temperature of 75°C.

- **Worst-case sign-off:** the frequency of the PLL (0.91 GHz) would be determined by the worst corner (SS, 1.08V, 125°C). The average frequency of RClk would depend on the process parameters of the die and the average operating conditions. For a typical die (TT) the frequency would be 1.46 GHz. Even in the case of a slow die (SS), the frequency would be 1.13 GHz.
- **Speed binning:** the margins for process variations would be mostly reduced for the PLL. Still, the margins for dynamic variability should be kept. For a typical die, the PLL could run at 1.22 GHz (−10%, 125°C) whereas RClk would run at 1.46 GHz.

Therefore, speed-ups ranging from 1.24× to 1.60× are obtained with regard to worst-case sign-off depending on the process parameters (delays between TT and SS). Compared with speed binning, the speed-ups range from 1.20× to 1.24×. An important observation is the high robustness to voltage noise, even when the supply changes by ±30%. While the PLL has to drastically reduce frequency to tolerate voltage droops (e.g., from 1.40 down to 0.82 GHz for a typical die), RClk only needs a very small reduction (from 1.38 down to 1.35 GHz). Thus, RClk is a resilient solution for systems living in hostile environments with unreliable power supplies, like low-cost regulators or energy scavenging scenarios. When considering local variability, some derating factors would be applied to PLL and RClk, but the benefits and conclusions of the study would be similar.

#### B. Adaptive clocks vs. Reactive clocks

Table III also compares the maximum frequency achieved by an AClk [7] in the same evaluation framework as the PLL and the RClk.

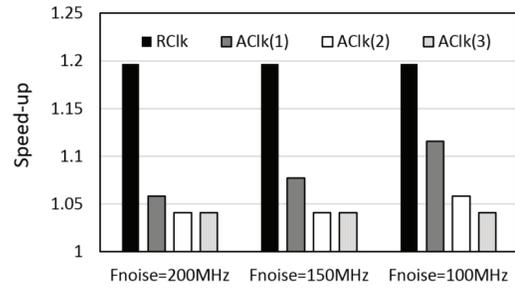


Fig. 7. Speed-ups for AClk and RClk on different frequencies of voltage noise and adapting latencies.

TABLE IV  
F<sub>AVG</sub> (GHZ) FOR DIFFERENT CLOCK TREE LENGTHS IN RClk.

|          | 1.2V ± 0% | 1.2V ± 10% | 1.2V ± 20% | 1.2V ± 30% |
|----------|-----------|------------|------------|------------|
| Short CT | 1.3942    | 1.3822     | 1.3679     | 1.3491     |
| Long CT  | 1.3775    | 1.3723     | 1.3653     | 1.3401     |

For this characterization, we assumed *one* cycle latency for sensing changes in the operating conditions and selecting a new clock period in the frequency synthesizer.

AClk responds differently to fast and slow variability. For instance, temperature varies in the order of milliseconds, and thus AClk is able to track these gradual alterations. Nonetheless, voltage fluctuations are abrupt and unexpected, dropping from nominal to lowest voltage level in few nanoseconds [17]. For the AES module, the voltage dropping time is close to the clock period of the circuit, forcing AClk to operate assuming conservative margins to survive first-order voltage droops. On a typical die (TT), running at 75°C and −10% voltage, AClk would run at 1.30 GHz whereas RClk would operate at 1.46 GHz. Hence, RClk achieves speed-ups ranging from 1.12× to 1.17× with regard to AClk, assuming the latter uses speed binning.

Fig. 7 illustrates the speed-ups obtained by AClk and RClk compared to the PLL. Different frequencies of voltage noise (200 MHz–100 MHz) have been evaluated in typical conditions. AClk has been characterized with distinct sensing and reaction latencies—from 1 cycle to 3 cycles. As it can be seen in Fig. 7, AClk performance gains are moderate when dealing with sudden fluctuations. Similarly, the performance of AClk is closer to that of the PLL as its feedback loop latency increases. Hence, AClk is only as effective as RClk if voltage variations are slow compared with its feedback loop latency, which is not the case for large modern chips, as shown in [7].

#### C. Impact of clock-data latency in Reactive clocks

Table IV shows the maximum (*zero-slack*) average frequency at which RClk could operate when implementing two different clock trees: a first one that is extracted from a critical path of the AES module (Short CT in the table), and a second one that has been elongated intentionally, requiring 200+ extra-buffers and around 6 cycles from the clock source to the flip-flops (Long CT in the table). To quantify the impact of variability when RClk is connected to long clock trees, electrical simulations introduce different levels of voltage noise while operating at 125°C. The maximum frequency was computed by subtracting the *slack* obtained in the launching path from the period originated in the capturing path.

Two conclusions can be drawn from Table IV. First, the difference between using a short or a long clock tree is negligible, even in the presence of deep voltage droops (less than 0.7% difference), since RClk benefits from exploiting clock-data compensation—the clock tree’s ability to track variability for in-flight clock pulses [12]. Second, the frequency for different clock tree implementations slightly

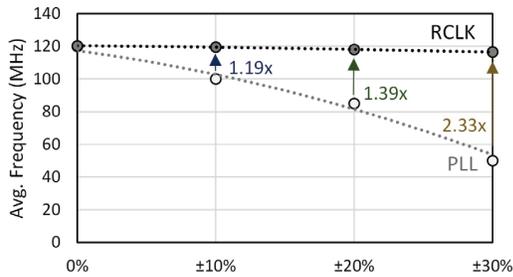


Fig. 8. FPGA: Voltage Noise vs. Frequency for 1.2V nominal voltage.

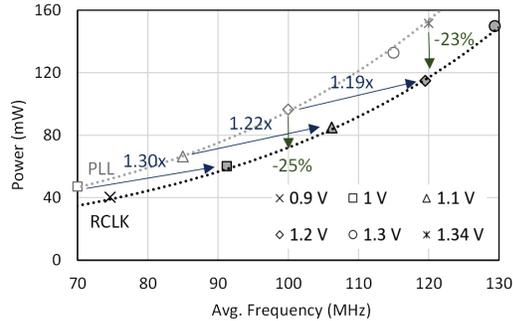


Fig. 9. FPGA: Frequency-Power plot for  $\pm 10\%$  voltage noise.

varies as operating conditions change. The reason is that clock trees are designed using specific cells that have different variability sensitivity from standard cells. Therefore, the clock tree may not adjust the clock period exactly as the VTRO would do it. Although the VTRO is constructed to match the variability of the critical paths (including the clock distribution network), it is still necessary to add a small margin to take these inaccuracies into consideration.

## VI. PROOF OF CONCEPT: FPGA PROTOTYPE

The benefits of RCLK were also estimated in an FPGA prototype. The tolerance to global variability was evaluated using an FPGA (Xilinx Spartan 3E) implementing the same AES module [16] and connected to an oscillating power supply. The clock tree was connected to a multiplexer capable of selecting between a PLL and a ring oscillator implemented as a chain of CLBs. The maximum error-free frequency achievable by the FPGA under different voltages and fluctuations was measured. As the experiments were performed on the same die, no process variations were measured. Thus, the experiments estimated the benefits of RCLK when compared to a perfect speed binning. The impact of temperature was negligible.

The results were consistent with the ones estimated by SPICE simulations under the assumption that dies are perfectly binned and demonstrate significant advantages in terms of performance and power when using RCLK. Fig. 8 plots the maximum frequency that was achieved under different amplitudes of voltage noise. The power supply was generated as a low-frequency sinusoidal signal, simulating the effect of an unregulated power supply (higher frequency variations would be cut by the on-chip decoupling capacitors). The noise amplitude ranged from 0% to  $\pm 30\%$ , i.e.,  $[0.84V \dots 1.56V]$ . As expected, the PLL frequency had to be reduced to keep the circuit operating correctly. However, RCLK could sustain an *almost constant average frequency* across a broad range of voltage noise. The speed-up of RCLK with regard to a PLL was 1.19 $\times$ , 1.39 $\times$  and 2.3 $\times$  for  $\pm 10\%$ ,  $\pm 20\%$  and  $\pm 30\%$  voltage noise, respectively.

Fig. 9 reports the power benefits for different voltage levels and  $\pm 10\%$  voltage noise. The vertical arrows ( $\downarrow$ ) indicate the power

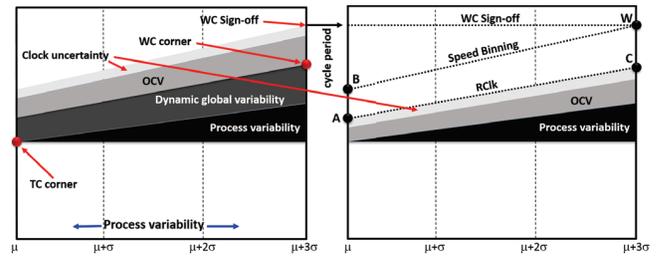


Fig. 10. Margins for sign-off in different scenarios.

savings obtained by voltage scaling at a given average frequency ( $-23\%$  at 120 MHz and  $-25\%$  at 100 MHz). The diagonal arrows ( $\nearrow$ ) connect iso-voltage points and represent the speed-up obtained by simply using RCLK instead of a PLL without changing voltage.

## VII. BENEFITS OF USING REACTIVE CLOCKS

The reduction of margins offered by RCLK results in substantial power and performance benefits that depend on the technology and the application domain. Fig. 10 (left) shows the margins used in STA for a corner-based sign-off. The horizontal axis represents the  $[\mu \dots \mu + 3\sigma]$  range of process variability for a particular distribution of dies<sup>3</sup> and the vertical axis represents the cycle period. The bullets represent the delay obtained at the typical corner (TC) and worst-case corner (WC). On-chip variability is added as a derating factor applied to the delay determined by the corner. Finally, a constant margin is added for clock uncertainty (jitter, timing models inaccuracies, etc.). The clock period for Worst-Case Sign-off is given by the addition of all the previous margins to the delays determined in the WC corner.

Fig. 10 (right) depicts the margins required for RCLK and the performance difference with regard to “Speed Binning” and “WC Sign-off”. The benefits come from eliminating margins for dynamic global variability. A few points have a special interest for analysis. Point **W** represents the cycle period for WC Sign-off. Point **A** serves as the cycle period for a die with typical process variation using RCLK. This point assumes the circuit working at a nominal *average voltage and temperature*. The difference between **A** and **W** represents the benefits for a typical die when no speed binning is applied.

When comparing RCLK with WC Sign-off, the benefits depend on the process characteristics of each die. For WC Sign-off, all dies are specified to run at a unique clock period that is calculated to guarantee a certain yield. RCLK allows each die to run at its natural speed, which is mostly limited by the process characteristics of the manufactured devices. With regard to the environmental parameters (voltage and temperature) the performance is determined by their average value instead of their worst value. Point **C** represents the cycle period achievable by a worst-case-process die using RCLK. The difference between **W** and **C** is determined by the global VT variability.

Speed binning [9] can reduce margins for process variability according to the process attributes of each die. In many cases, binning is used to classify dies and assign different prices according to their performance metrics. An ideal binning procedure would determine the clock frequency by using only the margins required for dynamic and local variability. Point **B** represents the achievable performance for a typical die. Again, the difference between **A** and **B** is determined by global VT variability. It can also be observed that no benefits are obtained between speed binning and WC Sign-off for worst-case dies.

The region between lines **B-W** (Speed Binning) and **A-C** (RCLK) represents the benefits of adapting to global VT variability. With

<sup>3</sup>For simplicity, we focus on the positive segment of the distribution and disregard the negative interval approaching the *best* corner.

RClk, every die runs at its *natural speed*, which is determined by its process characteristics and instantaneously reacts to the dynamic operating conditions. There is no need to do binning for a die to run at its natural speed (no at-speed testing is required), and margins can also be reduced for global dynamic variability.

### VIII. RELATED WORK

Various techniques have been proposed to mitigate the impact of dynamic variability. One of the most aggressive is Razor [18] and some variants based on a similar concept (e.g., [19]). They reduce the clock period at the expense of adding the non-trivial capability, both at the architectural and at the flip-flop level, to recover from timing errors. The main drawback of Razor-like techniques is the significant area overhead for error detection and correction, which involves intricate schemes to cope with metastability and architectural support for flushing the pipeline and replaying instructions. Blade [20] reduces the overheads of Razor by incorporating reconfigurable delay lines, error detecting latches and asynchronous structures, albeit it still requires modifications in the circuitry. Along the same lines, Tribeca [21] proposes to use ECC-protected data and local recovery mechanisms to reduce margins and work at nominal conditions.

All the previous techniques can only be applied in advanced microprocessors that incorporate schemes for error detection and recovery. The benefits oscillate around 30-50% power reduction, similar to those of the approach presented in this paper.

The most important dynamic variations are produced by voltage supply droops. Recently, various approaches have been proposed based on techniques for droop detection and adaptive clocking [7]. Based on the fact that voltage droops may last several cycles, droop detectors can be used to anticipate the arrival of the cycles with the largest droop amplitude. All the previously cited techniques propose digital schemes for droop detection based on perceiving differences or timing violations in delay lines or critical path monitors. After detection, different reaction schemes are proposed. One possible reaction is to quickly modify the clock frequency generated by a DLL [10], [11]. Another possibility is to stop the clock during the droop until the voltage returns to a stable level [22].

The main limitation of the mechanisms based on droop detection is the reaction latency to modify the clock frequency. During that time, voltage continues falling down and margins are also needed to compensate the increasing delays. After that, the margins required to tolerate the maximum droop amplitude can be saved. Moreover, these schemes do not exploit the fact that short-term voltage variations typically have zero average value over relatively short time intervals (e.g., a few  $\mu s$ ), because they are due to second-order inductive effects of the power distribution network. As discussed in Section III-A, *the performance of a circuit driven by RClk can be guaranteed over that time interval*. This is essential to ensure functionality of circuits that must satisfy hard external performance constraints.

### IX. CONCLUSIONS

After the happy-scaling days, it is time to find mechanisms that can maximally exploit the capabilities of technology nodes at nanometric scale. Reactive Clocks with Variability-Tracking Jitter (RClk) emerges as an innovative paradigm to handle variability and an alternative to paying the exorbitant costs of guardband margins.

### REFERENCES

[1] H. Jones, "Why migration to 20nm bulk CMOS and 16/14nm FinFETs is not best approach for the semiconductor industry," International Business Strategies, Los Gatos, CA, Tech. Rep., Jan. 2014.

[2] A. Kahng, "Scaling: More than Moore's law," *IEEE Design & Test of Computers*, vol. 27, no. 3, May/June 2010.

[3] A. B. Kahng, "Lithography-induced limits to scaling of design quality," in *Proc. SPIE*, vol. 9053, 2014, pp. 1–14.

[4] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, 2000.

[5] T. Fischer, J. Desai, B. Doyle, S. Naffziger, and B. Patella, "A 90-nm variable frequency clock system for a power-managed Itanium architecture processor," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 218–228, Jan 2006.

[6] J. Perez-Puigdemont, A. Calomarde, and F. Moll, "Variation tolerant self-adaptive clock generation architecture based on a ring oscillator," in *IEEE Int. SOC Conference*, 2012, pp. 387–392.

[7] N. Kurd, P. Mosalikanti, M. Neidengard, J. Douglas, and R. Kumar, "Next generation Intel core micro-architecture (Nehalem) clocking," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1121–1129, 2009.

[8] J. Leng, Y. Zu, and V. Reddi, "GPU Voltage Noise: Characterization and hierarchical smoothing of spatial and temporal voltage noise interference in gpu architectures," in *Int. Symp. on High Performance Computer Architecture*, Feb. 2015, pp. 161–173.

[9] B. Cory, R. Kapur, and B. Underwood, "Speed binning with path delay test in 150-nm technology," *IEEE Design & Test of Computers*, vol. 20, no. 5, pp. 41–45, Sep. 2003.

[10] K. Chae and S. Mukhopadhyay, "All-digital adaptive clocking to tolerate transient supply noise in a low-voltage operation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 12, pp. 893–897, Dec. 2012.

[11] C. Lefurgy, A. Drake, M. Floyd, M. Allen-Ware, B. Brock, J. Tierno, J. Carter, and R. Berry, "Active guardband management in Power7+ to save energy and maintain reliability," *IEEE Micro*, vol. 33, no. 4, pp. 35–45, Jul. 2013.

[12] K. Wong, T. Rahal-Arabi, M. Ma, and G. Taylor, "Enhancing microprocessor immunity to power supply with clock-data compensation," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 749–758, 2006.

[13] J. Bhasker and R. Chadha, *Static Timing Analysis for Nanometer Designs*. Springer, 2009.

[14] A. Drake, R. Senger, H. Deogun, G. Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, and V. Pokala, "A distributed critical-path timing monitor for a 65nm high-performance microprocessor," in *IEEE Int. Solid-State Circuits Conference*, Feb 2007, pp. 398–399.

[15] 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 *Int. Symp. on VLSI Circuits*, 2009, pp. 112–113.

[16] M. Litochevski and L. Dongjun, "High throughput and low area AES," 2012. [Online]. Available: [http://opencores.org/project,aes\\_highthroughput\\_lowarea](http://opencores.org/project,aes_highthroughput_lowarea)

[17] J. Jang, O. Franza, and W. Bursleson, "Compact expressions for period jitter of global binary clock trees," in *IEEE Electrical Performance of Electronic Packaging*, 2008, pp. 47–50.

[18] D. Ernst, S. N. S. Kim, Das, S. Pant, R. Rao, T. Pham, C. Zieslera, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation," in *IEEE Micro*, 2003, pp. 7–18.

[19] K. Bowman, J. Tschanz, N. Kim, J. Lee, C. Wilkerson, S. Lu, T. Karnik, and V. De, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 49–63, Jan. 2009.

[20] D. Hand, M. Trevisan, H. Hsin-Ho, C. Danlei, F. Butzke, L. Zhichao, M. Gibiluka, M. Breuer, N. L. V. Calazans, and P. Beerel, "Blade – a timing violation resilient asynchronous template," in *IEEE Int. Symp. on Asynchronous Circuits and Systems*, May 2015, pp. 21–28.

[21] M. S. Gupta, J. A. Rivers, P. Bose, G.-Y. Wei, and D. Brooks, "Tribeca: design for PVT variations with local recovery and fine-grained adaptation," in *Int. Symp. on Microarchitecture*, 2009, pp. 435–446.

[22] K. Bowman, C. Tokunaga, T. Karnik, V. De, and J. Tschanz, "A 22 nm all-digital dynamically adaptive clock distribution for supply voltage droop tolerance," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 4, pp. 907–916, Apr. 2013.