A Bi-Directional, Zero-Latency Adaptive Clocking Circuit in a 28-nm Wide AVFS System

Weiwei Shan, *Member, IEEE*, Wentao Dai, *Student Member, IEEE*, Liang Wan, Minyi Lu, Longxing Shi, *Senior Member, IEEE*, Mingoo Seok, *Senior Member, IEEE*, and Jun Yang, *Member, IEEE*

***Abstract*— Resilient circuits based on *in situ* timing monitoring adaptive voltage–frequency scaling (AVFS) eliminate excess time margins caused by process, voltage, and temperature (PVT) variations but suffer from 50% throughput loss during error recovery when operating at a half frequency. We propose a bi-directional adaptive clocking circuit to provide fine frequency tuning with zero latency for AVFS system. It can either stretch the clock cycle when there are timing errors to ensure correct function or compress the cycle when there are excess timing margins. To support a wide frequency range, we generate multiple phase clocks based on two delay lines and select one appropriate phase clock to obtain a stretched output clock, where balanced clock paths are obtained by a time-to-digital converter and dynamic-OR gates. Applied to a wide-operating-range AVFS system of an SHA-256 accelerator with transition detector (TD) latches, the whole AVFS system is able to respond to errors in one clock cycle, with dynamic-OR gates collecting all the errors in half a cycle and adaptive clocking circuit stretching at the current cycle. Fabricated in 28-nm CMOS, chip measurement shows that it achieves 38.6%–69.4% power gains at near threshold while reducing throughput loss during error recovery.**

***Index Terms*— Adaptive clocking, adaptive voltage–frequency scaling (AVFS), error detection, wide operating range.**

I. Introduction

IN TODAY'S nano-scale digital VLSI circuits, variations become severe, especially for low-voltage applications. As shown in Fig. 1(a), variations including process, voltage, and temperature (PVT) and skew, cause excess timing margins in design time in order to ensure correct timing across various operating conditions, including the worst case PVT condition. However, this leads to a waste of power and performance because the worst case condition rarely happens. To make it worse, PVT variations become even worse at low voltages, especially at the sub-threshold or near-threshold voltage (NTV) region, where the delay distributions are more dispersed across typical, slow, and fast conditions. Adaptive voltage–frequency scaling (AVFS) techniques based on *in situ* timing error monitoring are able to reduce or eliminate the excess margins [1], [2], [7]–[9], [21]–[28], [32]–[35]. Typical representative work includes Razor-lite [1], iRazor [2], Razor II [9], HEPP [21], Bubble Razor [35], and R-Processor [8].

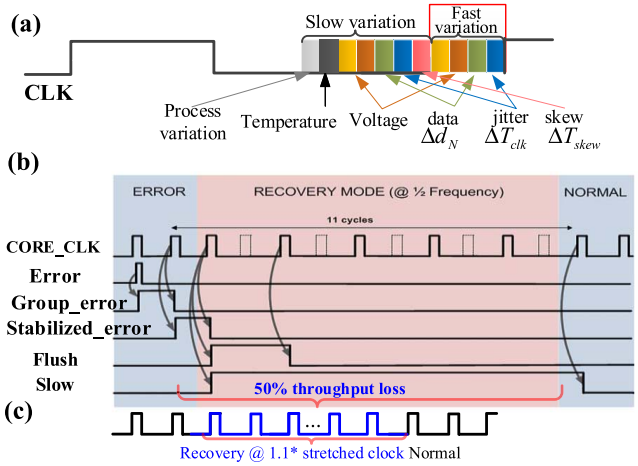


Fig. 1. (a) Worst case timing margins due to PVT variations. (b) AVFS error recovery with 50% throughput loss (Razor lite [1]). (c) Adaptive clocking in reducing the throughput.

In the resilient circuits, timing errors are detected and then corrected by replaying erroneous operations (such as instruction replay), usually at a half frequency by clock gating. Thus, they suffer from a large throughput loss during recovery. As the error recovery of Razor-lite [1] shown in Fig. 1(b), it used 11 clock cycles for error correction, causing 50% throughput rate loss during correction. Although this overhead is relatively low in the long time since the recovery rate is usually low, it is still a severe throughput loss during the error correction period. On the other hand, local voltage boosting was employed to correct errors within a cycle [8]. However, it needed an extra boosted voltage with modified power grid design, causing complex design and verification.

Recently proposed adaptive clock stretching circuits [3]–[6], [10]–[20] provide a fast stretched clock in one or a few clock cycles in a fine-grained scale. If used for error recovery, 10% clock stretching only induces 9% throughput loss, as shown in Fig. 1(c), which shows the prospect in AVFS systems. Adaptive clocking techniques generated multi-phase clocks through

Manuscript received August 29, 2019; revised November 14, 2019; accepted December 6, 2019. Date of publication December 27, 2019; date of current version February 25, 2020. This article was approved by Associate Editor Vivek De. This work was supported in part by the National Natural Science Foundation of China under Grant 61574033 and Grant 61774038 and in part by the China Major S&T Project under 2017ZX01030101 and Grant 2018ZX01031-101. *(Corresponding author: Jun Yang.)*

W. Shan, W. Dai, M. Lu, L. Shi, and J. Yang are with the National ASIC System Engineering Research Center, School of Electronic Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: wwshan@seu.edu.cn; waes1313@163.com; wsallen123@126.com; lmy_seuee@163.com; lxshi@seu.edu.cn; dragon@seu.edu.cn).

L. Wan is with ASR Microelectronics (Shanghai) Company, Ltd., Chengdu 610041, China.

M. Seok is with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: ms4415@columbia.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2019.2959494

a delay-locked loop (DLL) or a phase-locked loop (PLL) or a digital PLL, and then choose a different phase clock in each cycle to stretch the clock cycle continuously [3]–[6]. Thus, they were able to provide a fast clocking for the supply droop mitigation [5], [6], [10]–[12], [39]–[41], by increasing clock cycle quickly to accommodate the worst case droop voltage. Besides these droop detection-based adaptive clock stretching circuits, there are adaptive frequency/voltage tracking circuits [36]–[38], which may either decrease or increase the frequency in response to the supply droop. Recently, unified voltage and frequency regulators were proposed to let the supply voltage recover from the droop besides tuning the frequency [42], [43].

The existing adaptive clocking circuits are fast enough for droop detection/mitigation, because the first droop resonant frequency is usually quite slow compared to the circuit's working frequency. However, there are some obstacles to applying adaptive clocking circuits in an AVFS system. First, the response time to fast variations was not short enough for the AVFS system. For example, the adaptive clocking circuit with 1–3 cycles response time (depending on the configuration) [3] needs to budget the time required to avoid metastability, which is a universal way for production design. Second, some previous work supported only a limited range of clock frequency. An adaptive clock with fast response time as short as 1 cycle and supporting a wide frequency range is preferable for the wide-voltage-range AVFS system.

Therefore, we propose a bi-directional adaptive clock circuit that responds at the current cycle and apply it in an AVFS system. Based on our previous work of using a series of delay cells to replace the DLL in phase clock generator (PCG) [17], here we replace its PVT monitor by using a time-to-digital converter (TDC) to help choose the optimal phase clock. We also make it be able to shorten the clock cycle. Implemented on a resilient circuit in a 28-nm CMOS process, our proposed adaptive clock stretching circuit works together with timing error detection that when timing violations are detected, the clock is stretched immediately at the same cycle to prevent the potential functional error. Bi-directional clocking scheme is useful that besides slowing down the frequency when there is timing violation due to fast variations (i.e., droop), it is also able to increase its frequency rapidly in case when the timing turns better. Our main contributions include the following.

1) Having a zero latency to achieve clock stretching/compression at the current cycle, it is suitable for *in situ* timing monitor-based AVFS system. It is able to eliminate timing margins at run time, with little throughput loss during error recovery.
2) It supports a wide range of clock frequency down to tens of megahertz while being able to adaptively tune the stretching/compression value for maximum energy efficiency.

Fabricated in a 28-nm CMOS process, the measurement results of the prototype resilient chip demonstrate that our adaptive clocking circuit is able to tune the clock cycle in a fine-grain way with a response time of one clock cycle, whose tuning range is from 1/40 Tclk $\sim$ 1Tclk. Its minimum tuning step is 1/40, but its tuning step is variable with the operating

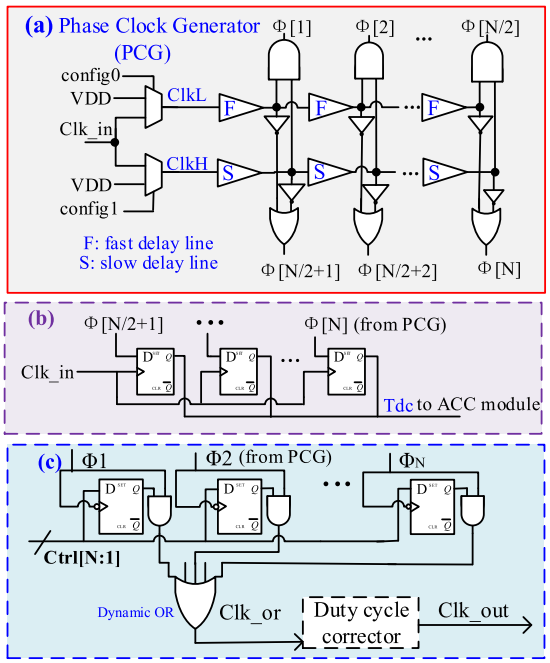


Fig. 2. Circuit architecture of the adaptive clocking, mainly composed of (a) PCG with dual delay lines, (b) PCD, and (c) PCS.

frequency that becomes coarser when working at a higher frequency in a certain range. With the proposed technique and timing monitoring AVFS, the chip achieves up to 69.4% power gain compared to the baseline with a constant 0.55-V supply voltage. Applied to error recovery in AVFS tuning, it incurs only 9% throughput loss when 10% clock stretching is applied, which is much less than the 50% throughput loss of the clock-gating way.

The remainder of this article is organized as follows. Section II describes the adaptive clocking circuit for either clock stretching or clock compression. Section III shows its application on an AVFS system, including the design of a holosymmetrical transition detector (HTD) and the AVFS system. Next, we show the measurement results in Section IV. Finally, Section V concludes this article.

II. Adaptive Clocking Circuit Design

A. Clock Stretching/Compression Architecture and Principle

The function of the fast adaptive clocking circuit is realized by selecting only one particular phase clock successively from the generated multi-phase clocks to tune the clock cycle. Therefore, it needs to fulfill the following requirements: 1) to generate multi-phase clocks ($\Phi 0$, $\Phi 1$, . . ., and ΦN) in a wide frequency range; 2) to select one of the appropriate clocks according to the AVFS requirement; and 3) to continuously circulate among the limited phase clocks when reaching the last available clock until clock stretching/compression is disabled.

We propose an adaptive clocking circuit supporting all the above-mentioned requirements, plus the bi-directional tuning ability. As the architecture shown in Fig. 2, it is mainly composed of: 1) a PCG to generate multiple phase clocks; 2) a phase clock detector (PCD) to detect the timing of the

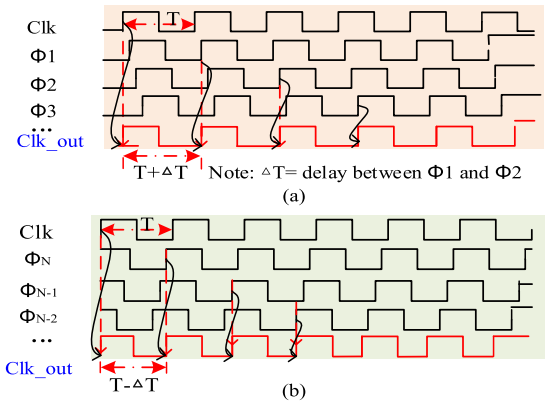


Fig. 3. Illustrated timing diagrams of clock stretching/compression. (a) Clock stretching. (b) Clock compression (ΔT is the delay of a delay cell).

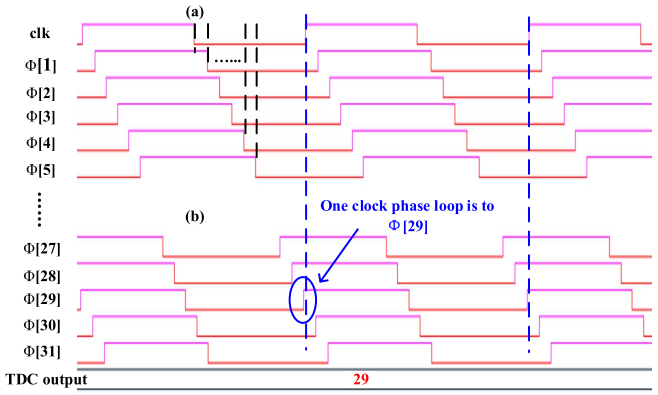


Fig. 4. Post-simulation results of the delay line. (a) Generated phase clocks at (SS corner, 0.95 V and 125 °). (b) TDC output to locate the 2π phase shift position.

last available phase clock; and 3) a phase clock selector (PCS) to select one of the phase clocks as the output, with control signals coming from an adaptive clock controller (ACC). Design details of each component are shown in Sections II-B–II-E.

The timing diagram of clock stretching is shown in Fig. 3(a), and the timing diagram of clock compression is shown in Fig. 3(b). Here, $\Phi 1, \Phi 2, \ldots$, and ΦN are the generated phase clocks from a system clock "Clk." They have an equal phase difference between two successive phase clocks. Take the clock stretching of $1\Delta T$ (ΔT is the delay between two successive phase clocks) as an example, as shown in Fig. 3(a), and the key point is to switch from the current phase clock to a lagged phase clock in each cycle. Once the output clock switches from "Clk" to $\Phi 1$ in the first cycle, the negative phase of output clock "Clk_out" is stretched immediately because of the phase difference between Clk and $\Phi 1$. In the next cycle, the system switches from $\Phi 1$ to $\Phi 2$ to keep its stretching status. Repeating this operation, a continuously stretched output clock is realized.

On the other hand, the clock compression is obtained by switching reversely from the current phase clock to a previous one. As shown in Fig. 3(b), starting from ΦN, $\Phi(N-1)$ is selected whose phase is ahead of ΦN, making the clock cycle compressed. The other parts are the same as clock stretching.

Different stretching/compression values can also be achieved. For example, in order to stretch $3\Delta T$ in a cycle, we choose $\Phi(i+3)$ in each cycle until reaching the last available phase clock, and here, $\Phi(i)$ is the previous phase clock.

B. PCG

The function of PCG is to generate multi-phase clocks of the same frequency but different phases with an equal phase difference. Instead of using DLL to provide multiple phase clocks, we use a series of delay cells for a compact design. In this article, to support a wide frequency range from megahertz to gigahertz, one delay line is not enough. We propose using two delay lines (one fast line and one slow line) to generate multiple phase clocks, as shown in Fig. 2(a). Each delay line is composed of a series of identical delay cells. Here, the CLKBUFV4 buffer is chosen as the basic component in the delay cell because it has a small difference between the rising delay and the falling delay. The slow or the fast delay line contains delay cells with 30 or 5 buffers in each cell, respectively. To make one delay line responsible for a certain frequency range, $N = 40$ delay cells are placed in each delay line.

By the way, the delay lines seem to be open loop, but it is controlled by the PCD and PCS modules. Although they are subject to PVT, it is acceptable for the AVFS system since the critical path's timing is also subject to PVT. The delay line being affected by PVT makes tuning value of the clock cycle also changing at the same time as the main circuit.

To save some area, each delay line is shortened to generates only half of the clock cycle's delay from $\Phi 1$ to $\Phi(N/2)$, and the other half cycle of phase clocks from $\Phi(N/2+1)$ to ΦN is generated by adding inverters after the preceding (N/2) phase clocks. The delay of the inverter is quite small compared to that of delay cells and thus can be neglected. Therefore, N phase clocks ($\Phi 1, \Phi 2, \ldots$, and ΦN) are generated.

Among these two delay lines, only one is activated each time, while the other delay line is not activated. Initially, the fast delay line is selected by default during the power-ON time. Once the working frequency decreases slow enough to exceed the range of the fast delay line, the slow delay line is activated instead. The monitoring and switching of the delay lines are realized by the TDC, which measures the delay time and controls the configuration bits of two delay lines. When the delay time exceeds the range of the fast delay line, TDC will generate an output signal with all bits to be 1, and the slow delay line is activated accordingly with its configuration signal set to high. This delay lines' transition time needs one clock cycle, but it will not influence the whole chips' performance since it usually happens during the initialization phase.

The functions and design details of each functional blocks are verified by post-simulation after layout with back annotation of timing at the worst case (SS corner, 0.95 V and 125 °), using the 28-nm CMOS process. Fig. 4(a) shows the generated phase clocks, and each phase increases equally.

C. PCD

During clock stretching, the clock cycle needs to be kept being stretched continuously. Since there are only a limited number of phase clocks, the selection of a new phase clock cannot continue when it reaches the last available phase clock. To solve this problem, when the accumulated clock phase reaches a 360° (2π) phase shift, we restart the whole clock picking process to let it start from the beginning phase clock again. In a word, a new phase clock is picked regularly in each cycle until (1) the stretch is disabled or (2) it reaches a 2π phase shift (called loop_end) so that one loop is finished and then it restarts from the beginning again.

Detecting this 2π phase-shifting timing is fulfilled by PCD, which is mainly composed of a TDC and some control logic, as shown in Fig. 2(b). Here, the TDC is composed of a series of FFs to measure the delay of the delay line in the PCG module to a digital signal. It is then sent to the ACC module to help locate the last available phase clock when reaching 2π phase shifting.

In order to use fewer resources, we only monitor the last half of the delayed phase clocks ($\Phi(N/2)$ to ΦN) in PCG. To be clear, if the actual 2π phase shift is less than $\Phi(N/2)$, the loop_end will be the 4π phase shift point, which also works. The output bits of DFFs are a series of "0s" and "1s," where the location of the first "0" is the loop_end. For example, if the outputs of the DFFs in TDC are 20'b X0_1111_1111, it means that the location of the 2π phase shift is the 9th FF and the phase clock should be $\Phi 29$ ($29 = 20 + 9$ because detection is in the last half of the delay line).

The post-simulation result of PCS is shown in Fig. 4(b), where the output of TDC is 29, and the 2π phase shift position agrees with the 29th phase clock. By the way, since we cannot guarantee that a 2π shift is exactly an integer number of ΔT, we choose to restart the phase clock selection just before the actual 2π shift time, in order to avoid an erroneous stretching.

To be clear, the last available clock phase is close to 2π phase shift position but not perfectly aligned. Assume that the last available clock phase (2π phase shift point by TDC) is between Φi and $\Phi i + 1$, and after the selection of Φi, the next clock phase will recycle to the beginning again instead of $\Phi i + 1$. By doing this, the clock cycle at this time is a little bit larger than the previous one, since the phase shift of this cycle is larger than the required one, with an increment of less than ΔT. Thus, it has a small impact on the clock cycle, while the circuit timing correctness is ensured.

D. PCS

PCS is used to select only one of the multi-phase clocks in each cycle as the output clock, with the requirement of making no race or hazard. Its control signals are from the ACC module. Here, only 1 bit of Ctrl signal is high each time so that only one phase clock is selected as the output. In the next cycle, a different phase clock is selected to keep the output clock being stretched. Since the control signals (N-bits Ctrl) are asynchronous to the corresponding phase clocks, they are first synchronized by the phase clocks ($\Phi 0, \Phi 1, \ldots$, and ΦN) to avoid the possible glitches at the clock output.

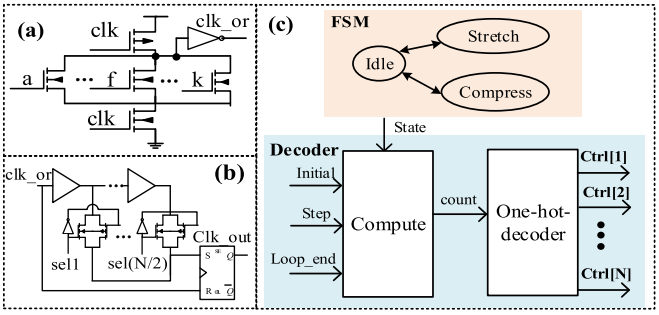


Fig. 5. Circuit schematics of (a) dynamic-OR gate and (b) DCC. (c) Architecture of ACC.

In order to keep its super-fast response ability, here, we use a simple synchronization of sampling each Ctrl bit by FFs at the negative edge of the corresponding phase clocks of $\Phi 1, \ldots$, and ΦN in the PCS module, as shown in Fig. 2(c). There are no setup time violations here since the clocks of these DFFs are from the delayed phase clocks so that the input data always arrive before the clocks, and since the clock stretching is realized by picking one of the delayed phase clocks at the negative clock phase, it does not affect the response time.

Then, the synchronized Ctrl signal, denoted as Ctrl_syn, is ANDed with the corresponding Φi to obtain a synchronized phase clock (Clk_and[i]) whose negative clock phase is stretched if Ctrl_syn[i] $= 1$. Those synchronized phase clocks are connected to a multi-input OR gate, whose output is the selected phase clock. However, a regular multi-input OR gate made the clock path unbalanced during selecting different phase clocks in each cycle [3], [4]. To provide balanced clock paths, instead of using traditional multi-staged OR gates, we introduce a dynamic-OR gate, as shown in Fig. 5(a). It has a balanced clock path no matter which path is selected. As seen from its schematic, it is actually a NAND-INV gate of the clk and the OR of all the delayed phase clocks. The OR of all the phase clocks is actually one of the phase clocks because only one phase clock is picked. Thus, its delay is quite short. Here, we use two stages of dynamic-OR gates for N ($N = 40$) inputs, and it does not degrade the clock slope.

However, using dynamic-OR gate causes the output clock having an irregular duty cycle. It reduces the positive clock width to the width of the overlap between Clk and the selected phase clock. The output clock may have a very narrow positive phase based on different phase clocks. Thus, we further use a simple duty-cycle corrector (DCC) to recover the clock duty cycle. It is composed of a delay line, an RS latch, and several transmission gates (TGs), as shown in Fig. 5(b). Here, the ORed clock signal "clk_or" is delayed by some delay cells and passed from one of the TGs to the "Set" port of an set-reset (SR) latch, which makes the output of SR latch to be 1 during the positive clock period. Thus, the positive clock period is lengthened by the delayed signal. Here, one and only one of the TGs is turned on, while their control signals are reused from the output of the TDC in PCD and transferred to a one-hot-based signal. Since the TDC can provide the timing of the 180 phase shift, it also gives the DCC a half-a-cycle timing to correct its duty cycle. Therefore, after the DCC, the output

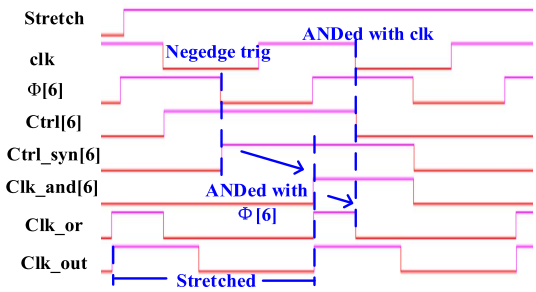


Fig. 6. Post-simulation results of the PCS module, showing results after synchronization, AND, dynamic-OR, and DCC.

clock has an enlarged positive phase that is close to the original clock but different negative clock phase, as shown in Fig. 6.

Post-simulation results of PCS are shown in Fig. 6. Here, the phase clock of Φ6 is chosen as an example. When Stretch is enabled in the positive clock period, Ctrl[6] is synchronized as Ctrl_syn[6] by a DFF sampling at the negative edge of Φ6, and then, it is AND ed with Φ6 to obtain a stretched negative clock phase. Then, it is sent to the dynamic-OR gate whose output signal Clk_or has a short positive phase but then enlarged after the DCC. Finally, Clk_out is stretched at the current clock cycle with a normal duty cycle.

E. ACC

The function of ACC is to generate the 40-bit Ctrl signal according to the stretching/compression requirements for PCS to pick one of the phase clocks. It is composed of a finite-state machine (FSM) and a decoder, as shown in the structure of Fig. 5(c). For the FSM, there are three states as "Idle," "Stretch," and "Compress." Here "Idle" means no frequency change, so we just pick any of Φi $(i = 1 - 40)$ and keep picking it to let it have the same clock cycle as the input clock. "Stretch" means clock stretching, which comes from the AVFS system when timing errors are detected. "Compress" means clock compression, which is also from the AVFS system.

The decoder computes the location of Φi in each cycle in the form of a count number and then turns it to the 40-bit Ctrl signal in a one-hot way that only one of the bits is 1. It is based on the FSM state and some settings such as Initial, Step, and Loop_end. Here, Initial is the default setting for PCS that Φ Initial is selected in the "Idle" state, and Step determines how long the clock should be stretched, where Step $= K$ means the clock cycle is stretched by $K * \Delta T$ in each cycle. Loop_end is the location of the delay cell that reaches the 2π phase shift.

Here, its computation rules for calculating the location of the next appropriate phase clock are given, denoted as count

$$\begin{aligned}
&\text{if state} = \text{Idle}, \quad \text{count} = \text{initial} \\
&\text{if state} = \text{Stretch} \\
&\quad \text{count} = \begin{cases} \text{count} + \text{step}, & \text{If count} < \text{Loop_end} \\ \text{count} - \text{Loop_end} + \text{Step}, & \text{else} \end{cases} \\
&\text{if state} = \text{Compress} \\
&\quad \text{count} = \begin{cases} \text{count-Step}, & \text{If count} > \text{Step} \\ \text{count} + \text{Loop_end} - \text{Step}, & \text{else.} \end{cases}
\end{aligned} \tag{1}$$

First, if the state is idle, we set count as the value of Initial.

Second, if clock stretching is enabled, count is added up with Step in each cycle if it does not meet the loop_end point (2π phase shift). Otherwise, it needs to start from the beginning point with consideration of the Step value, as in (1).

Third, for the clock compression state, count is calculated similarly as in (1), by choosing Φi backward. Finally, a one-hot 40-bit Ctrl signal is generated according to the count value.

F. Adaptive Clock Stretching/Compression Value Design

When used in an AVFS system, how much the clock cycle should be stretched/compressed is an issue. First, we offer the tuning ability of different stretching/compression values by controlling the phase difference between the phase clocks to be switched from one to another through the ACC module. In our design, the minimum stretching/compression value is one delay cell's delay (ΔT), while the maximum value is a whole clock cycle. This characteristic provides sufficient frequency tuning ability for AVFS. Thus, in the real AVFS control system, it is able to be tuned according to various strategies.

Here, we give an exemplary clock stretching strategy as follows, which starts from a conservative value and gradually adjusts the tuning value according to the detected timing errors. Initially, we stretch Tclk by 1/4. Then, if the clock stretching is not long enough such that an error occurs again during the instruction replay, we do the clock gating immediately to avoid further error and increase the stretching value at the same time; if the clock stretching is too much, we gradually decrease the stretching amount until getting to a suitable value. For the initial stretching amount of 1/4 Tclk, we did some simulations about the critical path's delay variance in two consecutive clock cycles and found that the late-arriving signal usually happens in the first 15% of Tclk. Very rarely should there be an abrupt delay variation that exceeds 25% of Tclk, because the timing delay variance caused by fast variations in two consecutive clock cycles is already considered in the design time.

III. AVFS System Design

A. Application Circuit and Overall AVFS System

Our adaptive clock is applied to a cryptographic accelerator of SHA256 circuit [22] designed with AVFS. SHA256 [29] is a widely used secure hashing algorithm released by NIST, which generates a unique 256-bit "digest" from an arbitrary length of the input message. It is a computation-intensive circuit.

Then, we design an AVFS system to eliminate the excess timing margins by AVFS tuning. The overall architecture of our AVFS system is shown in Fig. 7, composed of a main circuit, timing error detectors in critical paths, an AVFS FSM, and a clocking module. Here, our previous HTD [7] latch is used to detect the timing errors in the selected critical paths. It is used in parallel with a latch to replace the endpoint FFs in the critical paths. The detected timing signals are sent to dynamic-OR gates to get the total timing error signal (Pre_error), which is sent to the AVFS and the clocking modules. If there are timing errors, the adaptive clocking and

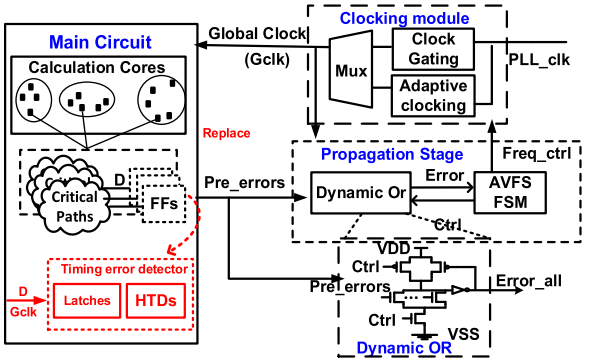


Fig. 7. Architecture of the resilient chip with AVFS tuning.

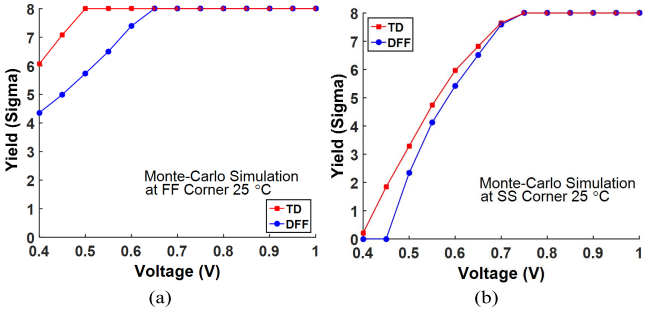


Fig. 9. Yield simulation of our HTD-latch and a standard DFF at 50 MHz and 25 °C from 0.4 to 1.0 V (a) FF corner. (b) SS corner.

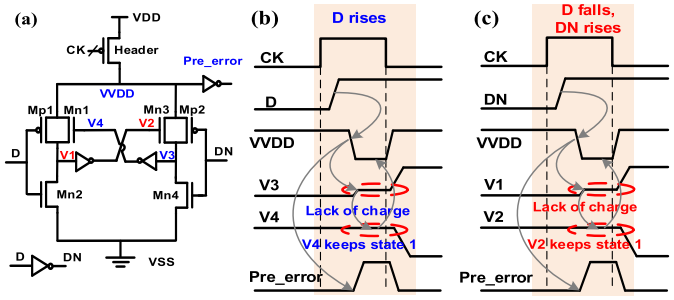


Fig. 8. (a) Schematic of an (HTD) [7]. (b) Timing diagram of HTD ($D = 0- > 1$). (c) Timing diagram of HTD ($D = 1- > 0$).

AVFS are activated to tune the system to eliminate the timing margins.

B. HTD Design

The function of HTD [see Fig. 8(a)] is to detect the late-arriving transitions in the positive clock phase by adding a PMOS header on the two-stage CMOS inverters. The short-circuit current would discharge the floating inner node (VVDD) when the header is turned off. Thus, the output inverter generates a positive Pre_error signal when detecting a late-arriving data transition.

The timing diagrams of HTD are shown in Fig. 8(b) and (c), respectively, when D rises from 0 to 1 and falls from 1 to 0. Taking the detection of falling D [see Fig. 8(c)] as an example, the negative clock phase is the charging time to charge the virtual power node (VVDD) to 1, while the positive clock phase is for detection. During the positive clock phase, if there is a data transition from 1 to 0, VVDD is discharged from 1 to 0 through Mn4 and Mn3 at first and then Mn2 gradually, inducing a positive pulse as the generated Pre_error signal.

An HTD is used in parallel with a latch for timing error detection, called HTD-latch. Multiple HTD latches are inserted in the endpoints of the selected critical paths to replace the original endpoint FFs. All the generated Pre_error signals are ORed together through dynamic-OR gates. AVFS control module controls the clocking module to activate either clock gating or adaptive clocking accordingly.

Metastability at the near-/sub-threshold voltages usually lasts much longer than nominal voltages, as described in [32]. Therefore, EDACs must be carefully designed to avoid the timing error detector metastable. A good example to solve this problem is the SafeRazor, which is metastability robust [33]. Flip-flop is a typical cell, which can easily be made metastable when toggling its input data simultaneously with the clock sampling edge. Therefore, those error-detection circuits based on flip-flop/latch suffer from metastability in the error-detection logic, as analyzed in [30] and [31].

Our HTD is able to work reliably at NTV that it outperforms FF/latch as well as FF/latch-based EDAC such as Razor lite [1] in terms of metastability, because it is not based on FF/latch sampling. Instead, it is a transition detector (TD) that relies on discharging of the floating node of VVDD. However, it may also suffer from metastability when the transition happens just as the sampling window opens, where HTD has to decide whether or not the transition happens before the positive clock edge when the detection window is open (no error) or after (an error). The source of metastability comes from VVDD partially discharging to the point where the output of HTD can go into a non-0/non-1 state for a prolonged period of time.

Our proposed HTD-latch, however, is not the bottleneck of the whole circuit. To prove it, we quantify its metastability by studying the yield of our HTD and compare it with a typical D flip-flop (DFF) [44]. The yield of HTD means its ability to detect data transition correctly in the positive clock phase. Yield simulations are done by NanoSpice Monte Carlo simulation at 50-MHz frequency, FF/SS corner, 25 °C, and the wide VDDs (from 0.4 to 1.0 V). For the DFF yield simulation, D-Clk is set as a large enough value of 50% Tclk and Clk-Q is set as 50% Tclk to allow the output to transit to a correct value. Here, NanoSpice is a yield simulation tool.

The yields' simulation results across 0.4–1 V are shown in Fig. 9, which clearly shows that our HTD has a higher yield than a DFF at NTV, no matter in SS or FF corner. Here, the red squared line is the yield of TD and the blue dotted line is the yield of a DFF, and the higher σ means a better yield. It shows that our HTD is not the bottleneck of the whole circuit.

C. AVFS Design

There are two kinds of AVFS tuning methods: one is to use voltage scaling as the main tuning method and adaptive clocking in error recovery, and the other one is to use adaptive clocking to tune the frequency without voltage scaling. Our adaptive clocking circuit is applicable to both methods. Here,

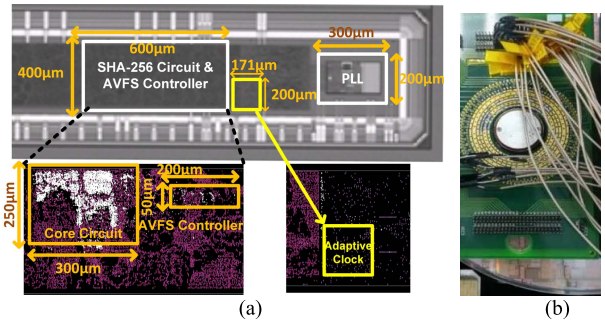


Fig. 10. (a) Microphotograph of our adaptive clocking circuit and AVFS chip. (b) Tested wafer on the probe card in the foundry.

our AVFS tuning strategy is implemented in this chip. When there is no timing error for a long time, the supply voltage is decreased gradually for lower power consumption. Until a timing error appears, the clock is gated immediately, and in this case, this is an accidental error. When there are two errors in a short time (set as 8 cycles here), viewed as timing intense, the adaptive clock is activated to stretch start the clock cycle immediately. Due to the time borrow of HTD-latch, the real timing violation is avoided in the current cycle when an error appears, but clock stretching is needed immediately to ensure correct operation for the next cycle, and a tuning mechanism of stretching/compression value is designed to find a proper value until reaching a point before the first failure (PBFF). Therefore, the resilient circuit is able to operate at an optimum point.

IV. Circuit Implementation and Measurement

A. Circuit Implementation Details

Fabricated in a 28-nm PolySiON (PS) CMOS process, this resilient chip is composed of the SHA-256 cryptographic accelerator, TD latches, adaptive clocking, and an AVFS tuning module. It has a core area of 0.24 mm^2 (400 μm $\times$ 600 μm), as shown in Fig. 10(a), while the adaptive clocking circuit has only 4000 gates. Its core area is 0.034 mm^2 with an ultra-low cell density of 4.0% because we did not optimize it due to sufficient circuit area in design time. Thus, it has an equivalent area of 1824 μm^2 if converted to a typical 75% cell density design.

The whole AVFS circuit is fabricated as a part of the foundry test circuits, and thus, it is tested under a probe card in foundry directly, as shown in Fig. 10(b). This allows us to measure the wafers with real SS, FF corners, which is more practical than MPW chips. We first measure the adaptive clocking circuit and then the whole AVFS system with adaptive clocking.

B. Measurement of the Adaptive Clocking Circuit

The adaptive clocking circuit is measured by an oscilloscope. An example of stretching 1/2 clock cycle is shown in Fig. 11 when working at a high frequency of 1 GHz, where it is divided by 16 to the output I/O due to the frequency limit of I/O. It can be seen that when Stretch is enabled, the output

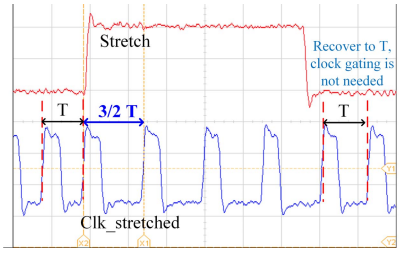


Fig. 11. Measured clock stretching at 1.0 V/1 GHz, divided by 16.

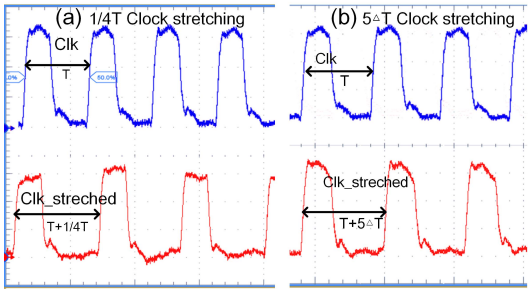


Fig. 12. Measured clock stretching at 1.0 V/50 MHz. (a) Stretched clock with 1/4 cycle stretching value. (b) Stretched clock with 5ΔT stretching value.

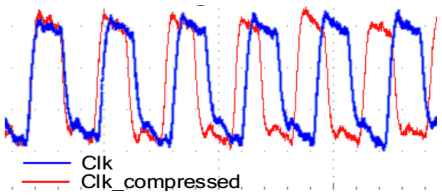


Fig. 13. Measured compressed clock with 2T value at 1.0 V/80 MHz.

clock is stretched immediately at the current clock cycle. Here, the Stretch signal is given from outside, not from the AVFS module. The power consumption of the adaptive clock circuit is 0.92 mW at 1.0 V/1 GHz for a slow die.

It allows stretching the clock at a negative clock phase by any value ranging from one delay cell's delay to a whole clock period. Here, we show the test results of stretching 1/4 clock cycle as well as 5ΔT at 50 MHz as examples. As shown in Fig. 12(a) and (b), respectively, output clocks are stretched to the configured value, where ΔT is the minimum delay of the delay cell. Here, the upper wave is the original clock and the lower wave is the stretched output clock.

The adaptive clocking circuit can also be employed to compress the clock cycle to allow the resilient circuit to obtain a higher performance. The result of the compressed clock is shown in Fig. 13 with a compression value of 2ΔT as an illustration, where the blue/light line is the input Clk and the red/dark line is the output Clk. It can be seen that the clock cycle is reduced effectively. Other compression values are also applicable with proper configurations.

Comparisons with other clocking stretching circuits [3], [4], [6] are shown in Table I. Some works provided bi-directional clocking to follow the supply droop [36]–[38], [42], [43]. Most of the current adaptive

TABLE I
ADAPTIVE CLOCKING CIRCUIT COMPARISONS

	Ours	ISSCC14 [3]	JSCC16 [4]	ISSCC17 [6]
Process	28 nm	28 nm	28 nm	14 nm
Response time	Current cycle	1-3 cycles	1 cycle	1.8 ns
Adaptive Function	Stretch & Compress	Stretch	Stretch	Stretch
Frequency range (Hz)	40M-over 1GHz	2G@0.86V 4G@1.45V	35M@0.4V 2.26G@1V	1G-4G
Area (μm^2)	1,824*	NA	1,120	4,995

* Equivalent area when converted to a typical 75% cell density design for the adaptive clocking circuit.

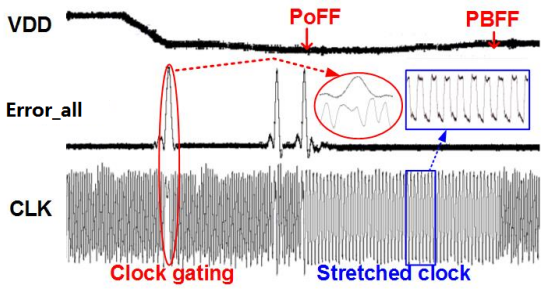


Fig. 14. AVFS measurement as VDD decreases with adaptive clocking.

clocking circuits need one or more clock cycles for tuning, while ours has zero latency that it is able to stretch the clock cycle at the current cycle. In addition, it has a wide-operating-frequency range to as low as 40 MHz to over 1 GHz. Therefore, our adaptive clocking circuit is applicable in the AVFS system, while it may also be used in coping with the voltage droop problem as other work.

C. Measurement of the Whole Resilient Circuit

The AVFS function of the chip is measured with adaptive clock tuning, with an exemplary tuning process shown in Fig. 14. At first, VDD decreases gradually when there is excess timing margin. When an error first occurs, clock gating is enabled. Then, it is disabled in the next clock cycle if it is a spontaneous error. When two errors occur in a short time, it is considered as a timing tense. Then, clock stretching is activated to avoid actual timing errors. After a long period with no timing errors, the adaptive clock recovers to the original. This long period is user-configurable, where it is set as 200 cycles here.

The resilient chip is designed for wide operating range down to NTV based on standard cell pruning and recharacterization under low voltages. The minimum supply voltage is measured as 0.39 V for TT wafer at 25 °. Total 24 dies are measured with the voltage range of 0.39–1.0 V. At normal VDD, power gains of 22.5% (slow die) to 42% (fast die) are obtained when compared to a fixed supply voltage, and at near-threshold region, 38.6%–69.4% power gains are obtained when compared to the baseline with a constant 0.55-V supply voltage.

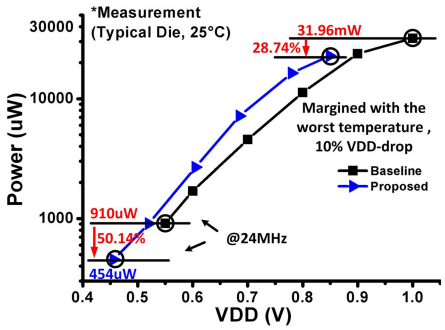


Fig. 15. Measured AVFS power savings for a typical die.

Here, the baseline frequencies are measured at the worst case conditions of 10% VDD drop, the slowest process corner, and the worst temperature, that is, 85 °C at the super-threshold voltage and −20 °C at NTV due to the temperature reverse effect at NTV.

Here, the power savings of a typical die under wide operating range are shown in Fig. 15. The baseline frequencies are tested to be 293.75 MHz at 1.0 V and 24 MHz at 0.55 V. At the near-threshold region, the typical die has a power gain of 50.14% and 28.74% at the super-threshold region compared to the baseline with a constant voltage. It can be seen that it achieves a much higher power saving when the supply voltage is lower due to severer variations at low voltages. Therefore, AVFS provides a good solution to kill margins to improve the performance/power at low voltages.

We compare our AVFS effect of the resilient chip with other recent resilient circuits [1], [2], as shown in Table II. While all the resilient circuits provide remarkable power savings, this article has the finest frequency tuning precision of 1/40 ∼ 1Tclk and bi-directional adaptive frequency tuning ability, which saves throughput loss during error recovery. It also has the widest voltage range of 0.39–1.0 V and, therefore, the largest power gain of 50.14% at NTV. In addition, it does not suffer from metastability issue because it relies on discharging the floating node rather than based on the flip-flop sampling mechanism, thus achieving better a yield than the DFF.

To be clear, there are two ways to recover from timing errors. One is instruction replay in CPU-based systems as Razor lite. The other is frequency tuning based on the latch-based timing monitor since time-borrow ability lets it be able to avoid real timing violations, such as iRazor. However, frequency needs to be decreased immediately to avoid the accumulated timing errors. Here, our clock stretching can be used in either of these two recovery ways. We use it by the second way here, and thus, the total recovery time is one cycle for clock stretching.

D. Response Time Analysis and Applicability Discussion

Response time is essential for AVFS systems. We do two kinds of optimization to make sure that the response time is one clock cycle. First, make sure that HTDs generate the timing errors and sum up all the timing errors in the positive clock cycle. Second, make sure that the adaptive clocking

TABLE II
COMPARISONS OF OUR EDAC SYSTEM AND PREVIOUS EDAC WORKS

	Razor-lite [1]	**iRazor [2]**	**Bubble Razor[35]**	**Tw tuning [22]**	**TEM [34]**	**This work**
Technology	45nm SOI	40nm	45nm SOI	40nm	40nm	28nm
NTV enabled	No	No	No	Yes	Yes	**Yes**
Sequencing	Flip-Flop	Latch	Latch	Flip-Flop	Flip-Flop	Latch
EDAC Extra transistors over a 24-T FF	8	1.46	20+dynOR +cluster	48	46	**5**
Metastability	Yes	Yes	No	No	No	**No**
EDAC Insertion rate	19.82%	8.67%	100%	5.5%	5.7%	12.4%
Freq tuning during recovery	Clock gating	Clock gating	Clock gating	Clock gating	Clock gating *	**Adaptive clocking**
Throughput loss @recovery	50%	50%	50%	50%	50%	**9% typical****
Typical Power gain	45.4%	41%	54%	33.3%	26%	50.14% @ NTV

* Clock gating is used for multi-stage time borrowing;
** Throughput loss of 9% comes from clock cycle being stretched by 10% in typical case, while the minimum stretching amount is 1/40Tclk.

circuit changes its clock period at the current cycle with zero latency, by stretching the negative clock cycle immediately. If considering that the timing error is detected at the second cycle when timing violations occur, the total response time of the AVFS system is one cycle, with the clock being stretched in the next cycle. In addition, due to the time borrowing of latches, timing violations do not really occur when errors are detected.

As for the applicability of our technique, it is able to work in a system with clock insertion delay less than half of a clock cycle. In another word, the total delay time from timing error generation to passing to the adaptive clocking circuit should be less than 0.5 clock cycle considering global clock distribution. Here, the response time for HTD-latch is about two times of inverter delay. The summing up time of total timing errors depends on how many stages of dynamic-ORs are used. Our using of dynamic-OR gate makes the sum of total errors be able to finish in a very short time. For example, for a big system that requires to monitor 100 000 path endpoints, it needs five stages of dynamic-OR, whose delay is approximately five OR gates. The last parts of the total delay are the wire delay and clock distribution mismatch, which limits the applicability of our technique in that they should be kept in the limit of half a cycle.

V. CONCLUSION

A bi-directional adaptive clock circuit is proposed and applied on a resilient chip with timing error detection and AVFS tuning in a 28-nm CMOS process. It supports a wide range of clock frequency from megahertz to gigahertz while being able to adaptively modulate the stretching/compression value for maximal energy efficiency in the AVFS system. This all-digital design is friendly to digital circuits, leading to a new solution for fast, fine, and bi-directional frequency tuning of timing monitor-based AVFS systems.

REFERENCES

[1] I. Kwon, S. Kim, D. Fick, M. Kim, Y.-P. Chen, and D. Sylvester, "Razor-lite: A light-weight register for error detection by observing virtual supply rails," *IEEE J. Solid-State Circuits*, vol. 49, no. 9, pp. 2054–2066, Sep. 2014.

[2] Y. Zhang *et al.*, "iRazor: 3-transistor current-based error detection and correction in an ARM Cortex-R4 processor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan./Feb. 2016, pp. 160–162.

[3] A. Grenat, S. Pant, R. Rachala, and S. Naffziger, "Adaptive clocking system for improved power efficiency in a 28 nm x86-64 microprocessor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 106–107.

[4] J. Kwak and B. Nikolić, "A self-adjustable clock generator with wide dynamic range in 28 nm FDSOI," *IEEE J. Solid-State Circuits*, vol. 51, no. 10, pp. 2368–2379, Oct. 2016.

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

[6] M. S. Floyd *et al.*, "Adaptive clocking in the POWER9 Processor for voltage droop protection," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 444–446.

[7] W. Dai, W. Shan, X. Shang, X. Liu, H. Cai, and J. Yang, "HTD: A light-weight holosymmetrical transition detector for wide-voltage-range variation resilient ICs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 11, pp. 3907–3917, Nov. 2018.

[8] S. Kim *et al.*, "Variation-tolerant, ultra-low-voltage microprocessor with a low-overhead, within-a-cycle, *in-situ* timing-error detection and correction technique," *IEEE J. Solid-State Circuits*, vol. 50, no. 6, pp. 1478–1490, Jun. 2015.

[9] S. Das *et al.*, "RazorII: *In situ* error detection and correction for PVT and SER tolerance," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan. 2009.

[10] M. Cho, C. Tokunaga, S. Kim, J. Tschanz, M. Khellah, and V. De, "Adaptive clocking with dynamic power gating for mitigating energy efficiency & performance impacts of fast voltage droop in a 22nm graphics execution core," in *Proc. Symp. VLSI Circuits (VLSIC)*, 2016, pp. 1–2.

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

[12] K. A. Bowman *et al.*, "A 16 nm all-digital auto-calibrating adaptive clock distribution for supply voltage droop tolerance across a wide operating range," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 8–17, Jan. 2016.

[13] K. Chae, C. Lee, and S. Mukhopadhyay, "Timing error prevention using elastic clocking," in *Proc. IEEE Int. Conf. IC Design Technol.*, May 2011, pp. 1–4.

[14] J. Kwak and B. Nikolić, "A 550-2260 MHz self-adjustable clock generator in 28 nm FDSOI," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2015, pp. 1–4.

[15] X. Zhang, T. Tong, D. Brooks, and G. Wei, "Supply-noise resilient adaptive clocking for battery-powered aerial microrobotic System-on-Chip in 40 nm CMOS," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2013, pp. 1–4.

[16] K. Wilcox *et al.*, "Steamroller module and adaptive clocking system in 28 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 24–34, Jan. 2015.

[17] W. Shan, L. Wan, X. Liu, X. Shang , W. Dai, S. Shao, J. Yang, and L. Shi, "A low overhead, within-a-cycle adaptive clock stretching circuit with wide operating range in 40 nm CMOS," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 11, pp. 1718–1722, Nov. 2018.

[18] K. Chae and S. Mukhopadhyay, "A dynamic timing error prevention technique in pipelines with time borrowing and clock stretching," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 1, pp. 74–83, Jan. 2014.

[19] S. Ghosh, D. Mohapatra, G. Karakonstantis, and K. Roy, "Voltage scalable high-speed robust hybrid arithmetic units using adaptive clocking," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 9, pp. 1301–1309, Sep. 2010.

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

[21] J. Zhou *et al.*, "HEPP: A new *in-situ* timing-error prediction and prevention technique for variation-tolerant ultra-low-voltage designs," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2013, pp. 129–132.

[22] W. Shan, X. Shang, L. Shi, W. Dai, and J. Yang, "Timing error prediction AVFS with detection window tuning for wide-operating-range ICs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 7, pp. 933–937, Jul. 2018.

[23] W. Dai, P. Liu, and W. Shan, "Short-path padding method for timing error resilient circuits based on transmission gates insertion," in *Proc. IEEE Great Lakes Symp. VLSI (GLSVLSI)*, May 2018, pp. 105–110.

[24] W. Shan, X. Shang, X. Wan, H. Cai, C. Zhang, and J. Yang, "A wide-voltage-range half-path timing error-detection system with a 9-transistor transition-detector in 40-nm CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 6, pp. 2288–2297, Jun. 2019.

[25] J. Zhou, C. Wang, X. Liu, X. Zhang, and M. Je, "An ultra-low voltage level shifter using revised wilson current mirror for fast and energy-efficient wide-range voltage conversion from sub-threshold to I/O voltage," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 3, pp. 697–706, Mar. 2015.

[26] S. Kim, J. P. Cerqueira, and M. Seok, "A 450 mV timing-margin-free waveform sorter based on body swapping error correction," in *Proc. Symp. VLSI Circuits (VLSIC)*, 2016, pp. 1–2.

[27] W. Jin, S. Kim, W. He, Z. Mao, and M. Seok, "*In situ* error detection techniques in ultralow voltage pipelines: Analysis and optimizations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 1032–1043, Nov. 2017.

[28] C. Wang *et al.*, "Near-threshold energy and area efficient reconfigurable DWPT/DWT processor for healthcare monitoring applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 1, pp. 70–74, Jan. 2015.

[29] I. Giechaskiel, C. Cremers, and K. B. Rasmussen, "When the crypto in cryptocurrencies breaks: Bitcoin security under broken primitives," *IEEE Security Privacy*, vol. 16, no. 4, pp. 46–56, Jul. 2018.

[30] S. Beer, M. Cannizzaro, J. Cortadella, R. Ginosar, and L. Lavagno, "Metastability in better-than-worst-case designs," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC)*, May 2014, pp. 101–102.

[31] R. Ginosar, "Metastability and synchronizers: A tutorial," *IEEE Des. Test Comput.*, vol. 28, no. 5, pp. 23–35, Sep. 2011.

[32] D. Hand *et al.*, "Blade—A timing violation resilient asynchronous template," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC)*, May 2015, pp. 21–28.

[33] M. Cannizzaro, S. Beer, J. Cortadella, R. Ginosar, and L. Lavagno, "SafeRazor: Metastability-robust adaptive clocking in resilient circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 9, pp. 2238–2247, Sep. 2015.

[34] H. Reyserhove and D. Wim, "Margin elimination through timing error detection in a near-threshold enabled 32-bit microcontroller in 40 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 53, no. 7, pp. 2101–2113, Jul. 2018.

[35] M. Fojtik *et al.*, "Bubble razor: Eliminating timing margins in an ARM cortex-M3 processor in 45 nm CMOS using architecturally independent error detection and correction," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 66–81, Jan. 2013.

[36] N. Kurd, J. Douglas, P. Mosalikanti, and R. Kumar, "Next generation Intel micro-architecture (Nehalem) clocking architecture," in *Proc. IEEE Symp. VLSI Circuits*, Hillsboro, OR, USA, Jun. 2008, pp. 62–63.

[37] D. Jiao and C. H. Kim, "A programmable adaptive phase-shifting PLL for clock data compensation under resonant supply noise," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2011, pp. 272–274.

[38] X. Sun *et al.*, "A combined all-digital PLL-buck slack regulation system with autonomous CCM/DCM transition control and 82% average voltage-margin reduction in a 0.6-to-1.0V cortex-M0 processor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, 2018, pp. 302–304.

[39] D. Jiao, J. Gu, and C. H. Kim, "Circuit design and modeling techniques for enhancing the clock-data compensation effect under resonant supply noise," *IEEE J. Solid-State Circuits*, vol. 45, no. 10, pp. 2130–2141, Oct. 2010.

[40] 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 J. Solid-State Circuits*, vol. 41, no. 1, pp. 218–228, Jan. 2006.

[41] C. Gonzalez *et al.*, "The 24-core POWER9 processor with adaptive clocking, 25-Gb/s accelerator links, and 16-Gb/s PCIe Gen4," *IEEE J. Solid-State Circuits*, vol. 53, no. 1, pp. 91–101, Jan. 2018.

[42] S. Gangopadhyay *et al.*, "UVFR: A unified voltage and frequency regulator with 500 MHz/0.84V to 100 KHz/0.27V operating range, 99.4% current efficiency and 27% supply guardband reduction," in *Proc. IEEE ESSCIRC*, Lausanne, Switzerland, Sep. 2016, pp. 321–324.

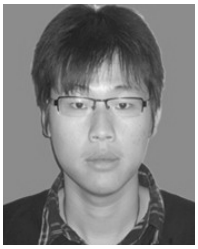
[43] F. U. Rahman *et al.*, "An all-digital unified clock frequency and switched-capacitor voltage regulator for variation tolerance in a sub-threshold ARM cortex M0 processor," in *Proc. Symp. VLSI Circuits*, Honolulu, HI, USA, 2018, pp. 65–66.

[44] W. Shan, W. Dai, C. Zhang, H. Cai, P. Liu, J. Yang, and L. Shi, "TG-SPP: A one-transmission-gate short-path padding for wide-voltage-range resilient circuits in 28 nm CMOS," *IEEE J. Solid-State Circuits*, to be published.

Weiwei Shan (M'09) received the B.S. degree in microelectronics from Tianjin University, Tianjin, China, in 2003, and the Ph.D. degree in microelectronics from Tsinghua University, Beijing, China, in January 2009.

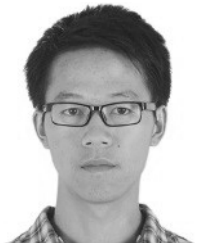
She did joint research at the University of Maryland, College Park, MD, USA, from 2007 to 2008. She was a Visiting Professor with Columbia University, New York, NY, USA, from November 2017 to February 2019. She is currently a Professor with the National ASIC Center, Southeast University, Nanjing, China. She has published over 50 technical articles in conferences and journals, including the IEEE International Solid-State Circuits Conference (ISSCC) and the *Journal of Solid-State Circuits* (JSSC). She has authorized over 30 invention patents. Her research mainly focuses on variation resilient adaptive VLSI circuits, ultra-low-power SoC design, and countermeasure techniques of security circuits.

Dr. Shan was a recipient of the 2014 State Scientific and Technological Progress Award, the 2017 A-SSCC Distinguished Design Award, and the 2018 GLSVLSI Best Paper Candidate.



Wentao Dai (S'17) received the B.S. degree in electronics engineering from Yangzhou University, Yangzhou, China, where he is currently pursuing the Ph.D. degree.

His current research interests include the low-power integrated circuit design, and adaptive circuits and systems.



Liang Wan received the B.S. degree in electronic science and technology from Tianjin University, Tianjin, China, in 2015, and the master's degree in microelectronics and solid-state electronics from Southeast University, Nanjing, China, in 2018.

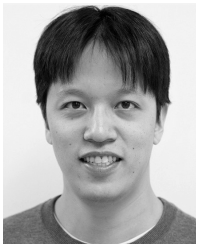
He is currently with ASR Microelectronics (Shanghai) Company, Ltd., Chengdu, Sichuan, China. His research mainly focuses on low-power digital integrated circuit design and adaptive clocking design.

Minyi Lu received the B.S. degree in electronic science and engineering from Southeast University, Nanjing, China, where he is currently pursuing the master's degree in microelectronics and solid-state electronics.

His current research interests include the low-power digital integrated circuit design and adaptive voltage scaling-based circuits and systems.

Longxing Shi (M'09–SM'18) received the Ph.D. degree in microelectronics from Southeast University, Nanjing, China, in 1988.

He is currently a Professor and the Dean of the Electrical Science and Technology School, Southeast University. He has published over 120 technical articles in conferences and journals and authorized over 200 Chinese invention patents and 12 USA patents. His research interests include system-on-chip design and RF and mixed-signal integrated circuit design.



Mingoo Seok (S'05–M'11–SM'18) received the B.S. degree (*summa cum laude*) in electrical engineering from Seoul National University, Seoul, South Korea, in 2005, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, AnnArbor, MI, USA, in 2007 and 2011, respectively.

He joined Texas Instruments, Dallas, TX, USA, as a Technical Staff Member, in 2011. He joined Columbia University, New York, NY, USA, in 2012, where he is currently an Associate Professor with the Department of Electrical Engineering. His current research interests include various aspects of very-large-scale integration circuits and architecture.

Dr. Seok received the 1999 Distinguished Undergraduate Scholarship and the 2005 Doctoral Fellowship from the Korea Foundation for Advanced Studies, and the 2008 Rackham Pre-Doctoral Fellowship from the University of Michigan. He was a recipient of the 2009 AMD/CICC Scholarship Award for picowatt voltage reference work, the 2009 Design Automation Conference (DAC)/IEEE International Solid-State Circuits Conference (ISSCC) Design Contest for the 35-pW sensor platform design, and the 2015 NSF CAREER Award. He has served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I from 2013 to 2015. He has been serving as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS since 2015.

Jun Yang (M'13) received the B.S. and Ph.D. degrees in electronic engineering from Southeast University, Nanjing, China, in 1999 and 2004, respectively.

He is currently a Professor with National ASIC System Engineering Research Center, Southeast University. He has published over 50 technical articles in conferences and journals, including IEEE International Solid-State Circuits Conference (ISSCC), Design Automation Conference (DAC), JSSC, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: Regular Papers (TCASI), IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: Express Briefs (TCASII), and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (TVLSI) SYSTEMS. He has authorized over 100 Chinese and USA invention patents. His current research focuses on SRAM design, in-memory computing, and near-threshold design.