

MEDAC: A Metastability Condition Detection and Correction Technique for a Near-Threshold-Voltage Multi-Voltage-/Frequency-Domain Network-on-Chip

Chuxiong Lin^{ID}, *Graduate Student Member, IEEE*, Weifeng He^{ID}, *Member, IEEE*, Yanan Sun^{ID}, *Member, IEEE*, Bingxi Pei, Pavan Kumar Chundi, Zhigang Mao, *Member, IEEE*, and Mingoo Seok^{ID}, *Senior Member, IEEE*

Abstract—This article presents a technique titled metastability condition detection and correction (MEDAC) that is featured in a near-threshold voltage (NTV) network-on-chip (NoC) hardware. The proposed technique introduces a double-sampling-based circuitry to detect whether the input data arrives too close to the receiver's clock edge, which is defined as a metastability condition. The MEDAC also contains a metastability condition mitigation scheme to monitor if the occurrence of such metastability conditions is frequent, and adaptively tune the receiver's clock phase to increase the mean time between metastability conditions. We prototype a test chip of the NoC in a 65-nm low-power process, which supports four independent voltage/frequency (V/F) domains in globally asynchronous locally synchronous (GALS) systems. Measurements across 0.4–1-V supply voltages with 13 different frequency ratios from 0.25 to 4 show that the MEDAC can well detect the metastability conditions and reduce the probability of entering such conditions by one to three orders of magnitude. This reduction, in turn, reduces the potential incorrect packets, thereby improving the data rate and energy efficiency by up to 21.1% and 27.2%, respectively.

Index Terms—Metastability, metastability condition detection and correction (MEDAC), near-threshold voltage (NTV), network-on-chip (NoC), reliability, sub-threshold voltage.

I. INTRODUCTION

DESIGNING reliable, high-performance, yet energy-efficient system-on-a-chip (SoC) for emerging mobile/embedded applications, such as autonomous vehicles, drones, and robots, has been receiving a large amount of research and development attentions. Such an SoC typically integrates a large number of heterogeneous building blocks, each

ideally operating in an independent voltage/frequency (V/F) domain [1].

In such an architecture, a network-on-chip (NoC) plays a key role in enabling high-speed and energy-efficient data communication. However, today's NoCs [2]–[6] adopt complex energy-efficient techniques, such as globally asynchronous locally synchronous (GALS) clocking system and dynamic voltage and frequency scaling (DVFS). Therefore, any two domains may have arbitrary and time-varying clock phase and frequency relationship. Moreover, each V/F domain uses largely different voltage, e.g., from nominal 1 V to near-threshold voltage (NTV), and clock frequency, e.g., from hundreds to sub-MHz. At such a low supply voltage, circuit delay exhibits severe delay variability across process, voltage, and temperature (PVT) variations, requiring a large number of timing margins. Therefore, it imposes a huge challenge to meet a high-reliability target, especially considering the increased chances of entering metastability in an NTV NoC.

Indeed, the metastability in clock-domain crossing (CDC) has been drawing great attention in the past decades. There has been continued research on the fundamentals, such as modeling [7], characterization [8]–[10], and on-chip test structures [11]. These studies show that supply voltages, technology scaling, and loading have a significant impact on the metastability of sequential cells.

On the other hand, multiple works proposed metastability tolerant techniques [12]–[20]. The synchronizer is one of the most adopted techniques, which uses two (or more) cascaded flip-flops to largely reduce the probability of metastability. Recent works also explore synchronizer circuits in ultra-low voltage operation [12]–[14]. Synchronizers help to resolve metastability to a stable state (0 or 1) in a clock cycle. Every additional synchronizer stage gives more time to resolve metastability, exponentially increasing the chance to tolerate metastability. However, synchronizers can neither guarantee metastability elimination nor provide long-term metastability mitigation.

Error detection sequential (EDS) circuits can also mitigate metastability [15]–[20]. Ernst *et al.* [15] and Cannizzaro *et al.* [16] proposed the use of the P/N-skewed gates for detecting metastability and dynamically adjust clock

Manuscript received July 20, 2020; revised October 9, 2020; accepted November 4, 2020. This article was approved by Associate Editor Edith Beigne. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB2202004, in part by NSFC under Grant 61774104, in part by Semiconductor Research Corporation under Grant TxACE 2712.012, and in part by US National Science Foundation under Grant CCF-1453142. (*Corresponding author: Weifeng He.*)

Chuxiong Lin, Weifeng He, Yanan Sun, Bingxi Pei, and Zhigang Mao are with the Department of Micro-Nano Electronics, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: hewf@sjtu.edu.cn).

Pavan Kumar Chundi and Mingoo Seok are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: ms4415@columbia.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JSSC.2020.3036856>.

Digital Object Identifier 10.1109/JSSC.2020.3036856

0018-9200 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

frequency or supply voltage to mitigate metastability. However, they have metastability issues on datapath and also require a reliable metastability detector. References [17]–[20] tried to avoid the metastability on datapath in the first place by providing extra timing margin to separate data and clock edges. However, this approach tends to increase the chance of min-delay violation and also poses an extra metastability risk in generating timing error signals.

Recent work also explores the idea of timing recovery and phase tuning for a two-clock system to avoid metastability and ensure reliable data communication [21]. Here, a phase detector is used to extract the phase relation between incoming data and the receiving clock, which then adjusts the clock phase accordingly to avoid data arriving in a time window that may cause metastability. Unfortunately, this technique requires a complex timing recovery circuit and only considered an unknown phase relationship of the same clock frequency.

Another work leveraged the predictability of relative phases between two ratiochronous clocks and proposed a double-buffering method for metastability mitigation [22]. It tries to maintain the relative phase to avoid metastability. However, as it relies on the initial measurement of the frequency ratio to estimate the clock phase of the transmitter, if the clock frequency varies, it has to re-measure the frequency ratio, which costs thousands of clock cycles.

References [7]–[11] focused on sequential element circuit design. References [12]–[14] proposed new synchronizer circuits optimized to improve its immunity to metastability. References [15]–[22] aimed to develop better synchronization circuits. However, most of these works target only one specific clocking system, such as synchronous [15]–[20], mesochronous [21], or periodic clocks [22]. Besides, they usually try to avoid metastability in the first place based on a known frequency relationship or predictive relative phase. However, in practice, the relative phase can drift, and the frequency relationship may vary over time. Last but not least, most of these works focus on nominal super-threshold voltage.

In this article, we propose a metastability condition detection and correction (MEDAC) technique in the context of an NTV NoC, where the clock frequencies of the V/F domains are different and the phase relationship is unknown and time-varying. Unlike previous synchronizer-based methods, our MEDAC technique, while still employing a synchronizer, aims the in situ detection of metastable conditions, and based on this detection, it aims to automatically set the clock phase such that it can avoid the condition that can potentially cause the metastability in sequencing elements. Therefore, as long as the phase is set properly, it can eliminate the chance of metastability events significantly.

The MEDAC contains mainly two components: metastability condition detection and long-term metastability mitigation. The detection hardware consists of two parallel synchronizers, which double-samples the incoming data at the receiver clock and its delayed version. It can detect if the data arrive too close to the clock edge, which is the necessary condition of metastability errors. If the detector finds that the receiver enters the metastability condition too often, the mitigation hardware tunes the phase of the receiver clock accordingly to separate

the data arrival time from clock edges. We perform an in-depth analysis of the relationship between the meantime to the next metastability condition (MTTM) and the phase differences of two clock domains across several frequency ratios. Based on this analysis, we develop the lookup table style control scheme that tunes the clock phase to maximize MTTM.

For the experiment, we prototype a GALS NoC test chip in a 65-nm low-power process. The NoC contains four independent V/F domains, each of a processing element (PE) and a router. We integrate the MEDAC into the dual-clock asynchronous FIFOs in the routers of the NoC. It can detect and mitigate metastability conditions and optionally initiate packet retransmission upon each detection of the metastability condition, which can ensure extremely high reliability. The NoC can work in a wide range of supply voltage and frequency from 0.4 V and 7.3 MHz to 1 V and 181 MHz. Measurement results show that the proposed MEDAC technique significantly reduces the probability of entering the metastability conditions by $17.7\times$ – $3430\times$ when the supply voltages scale from 1.0 to 0.4 V. The metastability condition rate reduces by about three orders of magnitude when sweeping clock frequency ratios from 0.25 to 4 for a supply voltage of 0.4 V. Furthermore, when enabling the retransmission scheme, the MEDAC improves the data rate and energy efficiency by up to 27.2% and 21.1%, respectively.

The remainder of this article is organized as follows. Section II discusses the metastability problems in CDC in a GALS system. Section III describes the proposed MEDAC circuits. Section IV discusses the architecture of the NoC test-chip employing the proposed MEDAC hardware. The measurement results are shown and discussed in Section V. Finally, Section VI concludes this article.

II. METASTABILITY IN CDC IN GALS

Data transmission across different clock domains in GALS systems can cause metastability in sequential elements. Fig. 1(a) shows such an example where the TX domain transmits data at the clock Tx_clk to the RX domain at the clock Rx_clk. The RX domain has the synchronizer that consists of two flip-flops R1 and R2. Fig. 1(b) shows the example waveforms where R1 undergoes the metastability and the synchronizer helps to resolve it. In this example, the Tx_data arrives at the synchronizer a bit earlier (by ΔP) than the rising edge of Rx_clk. If this phase difference ΔP is too small, Rx_data1 becomes metastable (in other words, R1 experiences metastability). This metastable output will be resolved to a stable “1” or “0” after resolution time t_r [23]. If t_r is roughly less than the RX domain’s clock period T_{RX} , the metastable Rx_data1 will be resolved to a stable state and captured by R2 in the second rising clock edge. However, if t_r is larger than T_{RX} , metastability would not be resolved, and R2 can also become metastable.

Prior works have found t_r as a function of ΔP [7]. As shown in Fig. 2, t_r increases exponentially as ΔP is getting closer to ΔP_{meta} [7]. t_r theoretically becomes infinite if $\Delta P = \Delta P_{meta}$. Note that ΔP_{meta} is typically zero but can be non-zero depending on the synchronizer implementation. Therefore, for a given

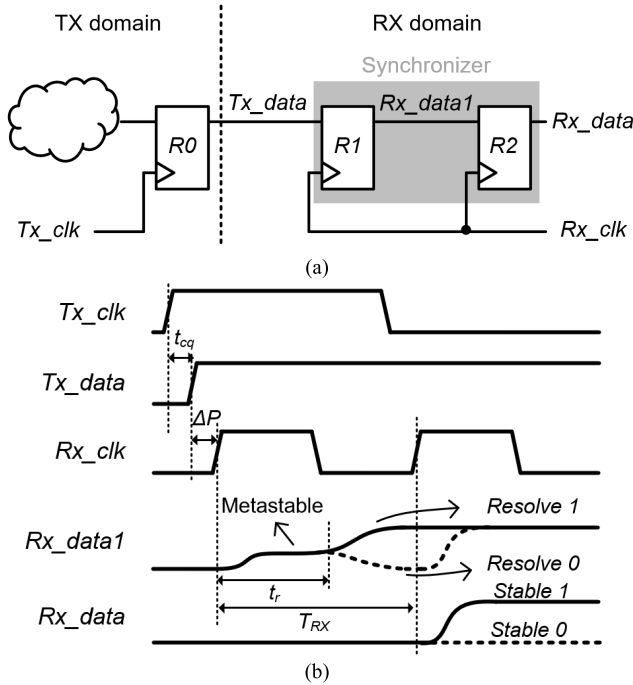


Fig. 1. (a) CDC example. (b) Timing diagram of synchronizer's metastability recovery.

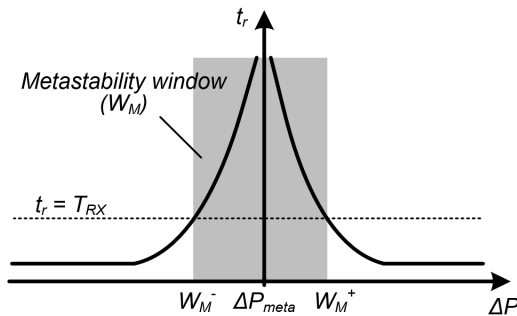


Fig. 2. Relationship between the resolution time t_r and phase difference ΔP .

T_{RX} , we can define the metastability window W_M (W_M^- to W_M^+) as the range of ΔP that makes t_r larger than T_{RX} . Then, we can define the necessary condition of metastability errors as the case that ΔP is within W_M . In this article, we will refer this case to a metastability condition.

In a GALS system, different from synchronous and mesochronous systems, $\Delta P[n]$, the phase difference at the n th Rx_clk cycle, is time-varying. This makes the metastability problem more complex to analyze and also difficult to address.

Fig. 3 shows an exemplary timing diagram on this change of the phase difference at the m th Tx_data and the n th Rx_clk cycle ($\Delta P[n]$). It shows that $\Delta P[n]$ would vary over time periodically (with some randomness, as well, due to clock jitter).

To further analyze, we have formulated $\Delta P[n]$ as

$$\Delta P[n] = \Delta P[0] + \frac{(n \cdot T_{RX} - m \cdot T_{TX})}{T_{RX}} \cdot 2\pi \quad (1)$$

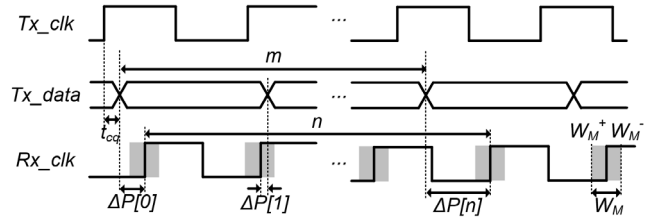


Fig. 3. Phase difference can vary over time in a GALS system, especially if the clock frequencies of TX and RX are different.

where T_{RX} and T_{TX} are, respectively, the clock periods of RX and TX domains. We assume that the m th Tx_data should arrive before the n th Rx_clk edge and its metastability window boundary. We can then formulate the relation of m and n as

$$m = \left\lfloor \left(\frac{\Delta P[0]}{2\pi} + n + \frac{|W_M^-|}{2\pi} \right) \cdot \frac{T_{RX}}{T_{TX}} \right\rfloor \quad (2)$$

In practice, a finite amount of clock jitter and frequency drift appears, causing almost arbitrary phase variations. To include these effects, we reformulate (1) and (2), respectively, to

$$\Delta P[n] = \Delta P[0] + \Delta P_{jitter} + (n - (1 + \varepsilon)k \cdot m) \cdot 2\pi \quad (3)$$

$$m = \left\lfloor \left(\frac{\Delta P[0]}{2\pi} + n + \frac{|W_M^-|}{2\pi} \right) \cdot \frac{1}{(1 + \varepsilon)k} \right\rfloor \quad (4)$$

where ΔP_{jitter} is a normally distributed random variable representing the combined clock jitter of TX and RX domains, k is T_{TX}/T_{RX} , and ε is the variation of k caused by frequency drift, which is assumed to follow a nominal distribution with a range of ± 100 part-per-million (ppm). The resulted (3) and (4) show that $\Delta P[n]$ is a function of $\Delta P[0]$, k , and n .

We also define a metric n_{MTTM} , which refers to the mean number of Rx_clk cycles to enter the next metastability condition for a given $\Delta P[0]$. We assume the mean value of ΔP_{jitter} is zero, and then, we can formulate n_{MTTM} to be

$$n_{MTTM} = \min\{n | n \in N, W_M^- \leq \Delta P[n] \leq W_M^+, \Delta P_{jitter} = 0\}. \quad (5)$$

Using (3)–(5), we develop a model in MATLAB, with which we can compute $\Delta P[n]$ and n_{MTTM} across different k and $\Delta P[0]$ values. Using this model, we generate the pattern of $\Delta P[n]$ across k values from 0.25 to 4 with a step of 0.0001. For each k value, we swept $\Delta P[0]$ from 0 to 2π with a step of $2\pi/10^4$.

Fig. 4 shows the two important patterns that we found from the experiment. The first pattern appears if k is an integer. If we ignore ΔP_{jitter} and ε , $\Delta P[n]$ becomes a constant ($\Delta P[0]$). Therefore, if $W_M^- < \Delta P[0] < W_M^+$ [as shown with the gray zone in Fig. 4(a)], the system undergoes the metastability condition every cycle, and n_{MTTM} becomes one. On the other hand, if $W_M^+ < \Delta P[0] < 2\pi + W_M^-$, no metastability occurs, making n_{MTTM} to be infinite. The second pattern is when k is not an integer. Fig. 4(b) shows the pattern for $k = 1.0001$, where $\Delta P[n]$ varies periodically over time. As $\Delta P[n]$ enters the gray zone, the system undergoes the metastability condition. Fig. 4(c) shows the pattern for $k = 2.56$.

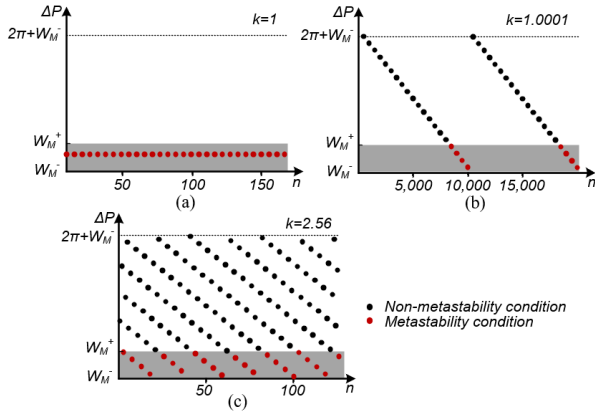


Fig. 4. Phase difference over clock cycles if (a) $k = 1$, (b) $k = 1.0001$, and (c) $k = 2.56$.

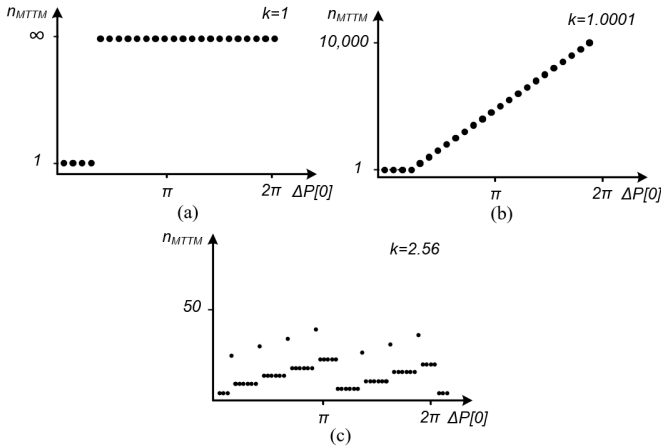


Fig. 5. Mean clock cycles to next metastability condition n_{MTTM} as a function of initial phase difference $\Delta P[0]$ if (a) $k = 1$, (b) $k = 1.0001$, and (c) $k = 2.56$.

Similarly, the system periodically undergoes the metastability condition, which increases the risks of having metastability incurred timing error.

We then investigated n_{MTTM} across different initial phase differences (i.e., $\Delta P[0]$). As shown in Fig. 5(a) and (b), if k is an integer ($k = 1$) or near integer ($k = 1.0001$), some $\Delta P[0]$ values lead to small n_{MTTM} , while other $\Delta P[0]$ enables very large n_{MTTM} . On the other hand, as shown in Fig. 5(c), certain k values make n_{MTTM} small (e.g., < 50) regardless of $\Delta P[0]$ values.

The analysis above shows that it is possible to increase n_{MTTM} by tuning $\Delta P[0]$. The cases shown in Fig. 5(a) and (b) are good candidates for such tuning. For the case shown in Fig. 5(c), however, it is difficult to significantly improve n_{MTTM} . These highly non-linear patterns are found in some k values. Thus, it would be critical to avoid such a k value in the system design phase. With that in mind, for given k values that a system wants to support, the key research question is then how we can dynamically tune $\Delta P[0]$, which we will further discuss in Section III-B.

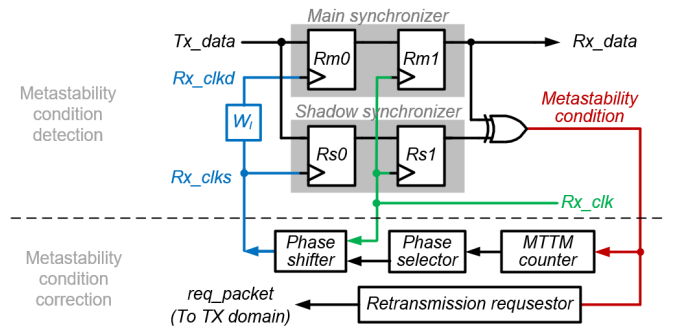


Fig. 6. Proposed MEDAC hardware.

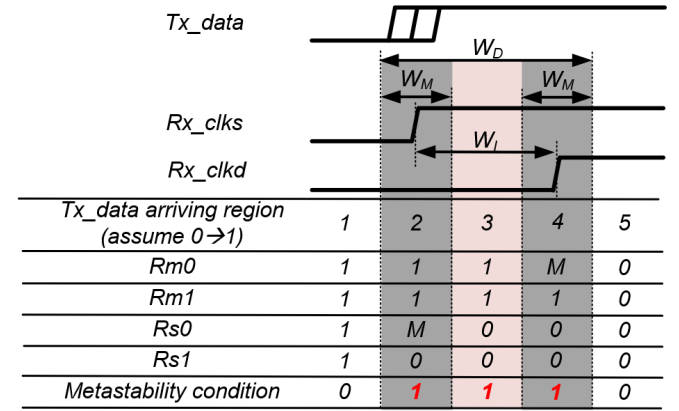


Fig. 7. Operation of the proposed metastability condition detection hardware.

III. METASTABILITY CONDITION DETECTION AND CORRECTION

A. Proposed Metastability Condition Detection

We propose the double-sampling synchronizers (see Fig. 6, top) for detecting the necessary condition of metastability. It contains the main synchronizer along with a shadow one. The main and shadow synchronizers are triggered, respectively, by the Rx_clk_d and the Rx_clk_s ; the former is the delayed version of the latter by W_l delay. The detector can detect if Tx_data arrives closely to either Rx_clk_s or Rx_clk_d , i.e., the metastability condition that may make one of the flip-flops in the synchronizers metastable. In such a case, the two synchronizers capture different values, and the XOR gate detects the difference and flags a metastability condition output.

Fig. 7 shows the example timing diagram of the proposed detector. W_M represents the size of the metastability window. In the proposed detector, Tx_data can arrive in one of the five possible regions. If it arrives at the timing region 1 (5), Tx_data changes ahead of (after) the metastability window of Rx_clk_s .

In this case, both main and shadow synchronizers capture the same and stable value, and the detector, indeed, finds no metastability condition.

On the other hand, Tx_data may arrive in timing region 2 (4). In other words, Tx_data may make a change in

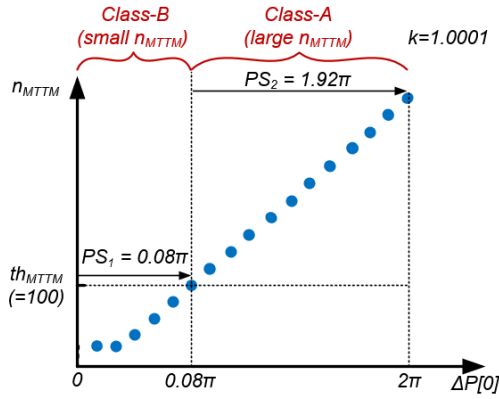


Fig. 8. Class-A and Class-B $\Delta P[0]$ ranges. The former enables large n_{MTTM} . The proposed metastability mitigation scheme aims to tune the RX clock's phase from Class-B into Class-A to increase n_{MTTM} .

the metastability window of Rx_clks (Rx_clkd). In such a case, the main (shadow) synchronizer captures Tx_data as a stable "1" ("0"). However, the first flip-flop in the shadow (main) synchronizer captures Tx_data that is switching, and therefore, its output can be metastable. If the shadow (main) synchronizer resolves the metastable output to "0" ("1"), i.e., differently from the main (shadow) synchronizer, the detector flags it as a metastability condition. If the output resolves to "1" ("0"), identically to the main (shadow) synchronizer, it does not flag. Finally, if Tx_data arrives in timing region 3, both synchronizers capture stable but different values. As a result, the detector makes a false alarm. To eliminate this false alarm, i.e., to remove the timing region 3, we should set W_1 equal to W_M .

B. Proposed Long-Term Metastability Condition Mitigation

The synchronizer is very likely to stabilize received data, but it cannot guarantee the stabilized data to be correct. Besides, the synchronizer does nothing to reduce the chance that input data arrive near the clock edges. This increases the chance that the synchronizer may not be able to resolve the metastability in a clock cycle, thereby increasing the chance to incur a timing error. It is, therefore, critical to make the CDC hardware to avoid having metastability conditions for the long term.

Monitoring and adaptively tuning the phase difference between two clocks can implement such a long-term mitigation scheme. A key observation is that the current phase difference $\Delta P[0]$ can determine CDC hardware to have small or large n_{MTTM} . We can define the range of $\Delta P[0]$ that makes n_{MTTM} greater than th_{MTTM} as Class-A. Here, th_{MTTM} represents a designer-defined threshold value of minimum acceptable n_{MTTM} . On the other hand, the range of $\Delta P[0]$ that makes n_{MTTM} smaller than th_{MTTM} is defined as Class-B. Fig. 8 shows the exemplary case where $k = 1.0001$ and $th_{MTTM} = 100$. The Class-B is found to be $0-0.08\pi$, and the Class-A is $0.08\pi-2\pi$.

However, it is hard to dynamically measure the value of $\Delta P[0]$. Instead, it is much more cost-effective to infer whether the current $\Delta P[0]$ is Class-A or -B. This can be done by counting the number of clock cycles between two flag signals

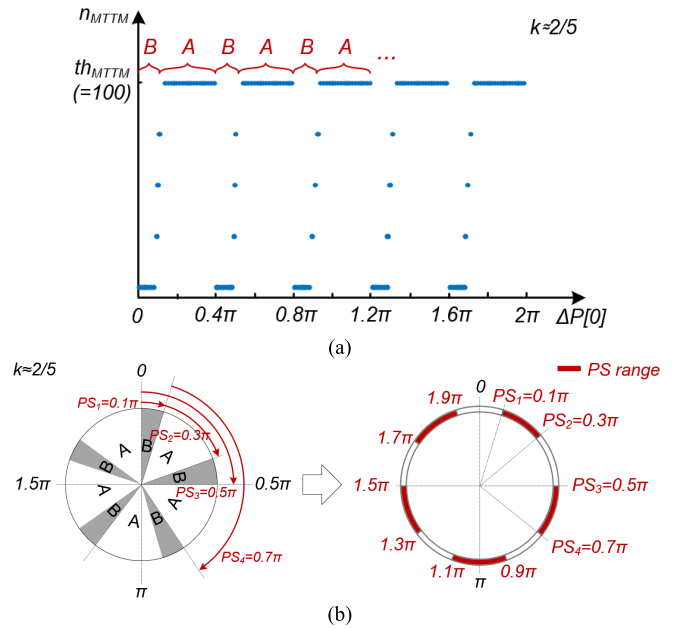


Fig. 9. (a) n_{MTTM} as a function of $\Delta P[0]$ if $k \approx 0.4$. (b) Computed PS ranges when $k \approx 0.4$.

of the metastability condition detection and comparing the clock cycles to the threshold (th_{MTTM}). In other words, if the system entering the metastability condition more frequently than the threshold, we infer that the current $\Delta P[0]$ is Class-B.

With the inference on the current $\Delta P[0]$ being available, the next key question is then how much phase shift (PS) of Rx_clk we need for making $\Delta P[0]$ to be Class-A from now on. Fig. 8 shows the simple example of how we can do this if $k = 1.0001$. Recall Class-B $\Delta P[0]$ is from 0 to 0.08π and Class-A is from 0.08π to 2π . Therefore, the range of PS values is simply calculated: $PS_1 = 0.08\pi$ ($= 0.08\pi - 0$) to $PS_2 = 1.92\pi$ ($= 2\pi - 0.08\pi$). We mark them in Fig. 8.

On the other hand, Fig. 9(a) shows a more complex case with $k \approx 0.4$. In this case, the ranges of Class-A and Class-B $\Delta P[0]$ values alternate (th_{MTTM} is set to be 100 cycles). Note that the th_{MTTM} is configurable. Increasing the threshold can make the range of Class-B larger and the range of Class-A smaller, which means that smaller ranges of PS values are available. To find out PS values more systematically, we make an alternative representation, as shown on the left-hand side of Fig. 9(b), where each region indicates Class-A or -B. Suppose that, currently, $\Delta P[0]$ is in the first Class-B segment (from 0 to 0.1π) to move it to the next Class-A region (from 0.1π to 0.4π), and the PS values should be from 0.1π to 0.3π , i.e., PS_1 to PS_2 in Fig. 9(b). Similarly, to move it to the second next Class-A region (from 0.5π to 0.8π), the PS values should be from 0.5π to 0.7π , i.e., PS_3 to PS_4 in Fig. 9(b). We repeat this process across all the Class-A regions for the first Class-B region, finding all the possible ranges of PS values. Then, we repeat the same process for each of the other Class-B regions. Since the detection hardware knows only the class of the current $\Delta P[0]$, not the exact value of it, chosen PS values should be able to work for all Class-B regions. For this reason, we choose only the overlapping parts of the PS ranges that we

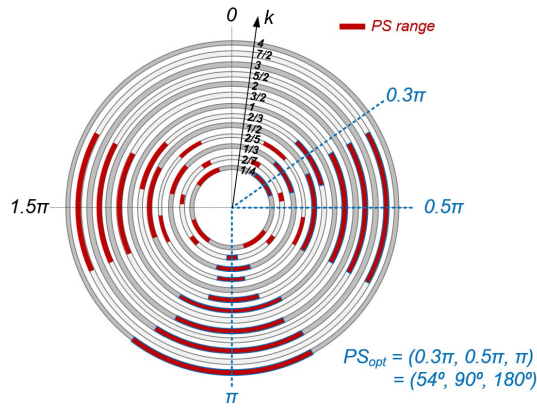


Fig. 10. PS ranges for the 13 k values in red curved strips. The three PS_{opt} values that can cover all the k values (in blue dashed lines).

found across all Class-B regions. Fig. 9(b) shows the set of the PS values for the example of $k \approx 0.4$, which are $0.1\pi - 0.3\pi$, $0.5\pi - 0.7\pi$, $0.9\pi - 1.1\pi$, $1.3\pi - 1.5\pi$, and $1.7\pi - 1.9\pi$.

As indicated in the abovementioned analysis, a different k value leads to a different set of PS values. Therefore, to support a GALS system supporting several different clock frequencies, we must compute PS values for all supported k values. Also, it is ideal to consider the worst case jitter and k variations to ensure that the chosen PS values are optimal and robust for non-ideal clocks. In this example, we assume to support 13 different k values from 0.25 to 4. Fig. 10 shows the computed PS values. Each ring represents the ranges of PS values for a selected k value. To supports all these k values, we need to pick several PS values that collectively include the necessary PS values for different k values. While we may choose as many PS values, only three PS values, 54° , 90° , and 180° , can still do the job in this example since, for any of the k values, at least one of these three PS values can make Class-B $\Delta P[0]$ become Class-A. Moreover, as shown in Fig. 10, the deviation of PS values is tolerable (e.g., $PS = 54^\circ \pm 11^\circ$ can work if k is $1/4$, $1/3$, or $2/3$), which reduces the design complexity of implementing these PS values.

Based on the abovementioned analysis, we design the metastability condition correction hardware consisting of mainly three components (the bottom of Fig. 6). First, the MTTM counter counts the clock cycles and calculates n_{MTTM} based on the metastability condition flag input. The phase selector monitors the n_{MTTM} value, and if n_{MTTM} is less than a user-defined threshold th_{MTTM} , it chooses a PS value among the set of the pre-computed PS values based on a round-robin algorithm.

As shown in Fig. 11, the phase shifter, consisting of variable delay lines, adds the selected PS to Rx_clk . If the PS value is 0° or 180° , the phase shifter selects Rx_clk or inversed Rx_clk , respectively; if the PS value is 54° or 90° , to delay Rx_clk , the phase shifter uses the current-starve inverter-based delay lines. For example, $PS = 54^\circ \pm 11^\circ$ can be translated to acceptable delays of 7.5 ± 1.53 fan-out-of-4 (FO4) delays if the clock period is 50 FO4 delays.

There is a small but finite amount of chance that the “metastability condition” signal can become metastable.

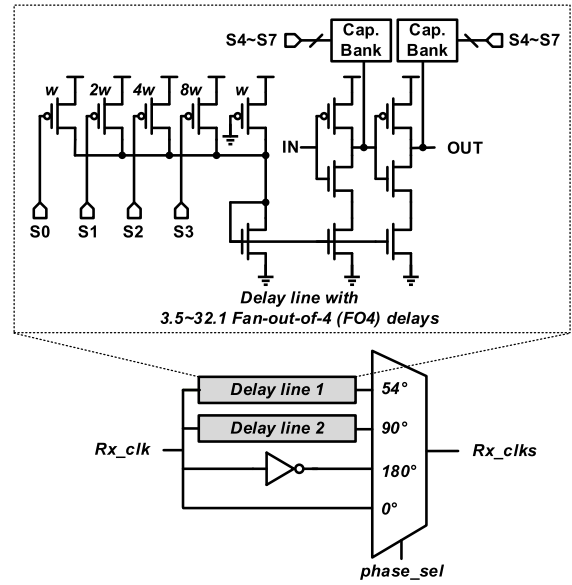


Fig. 11. Structure of the phase shifter. The phase shifter can select clock among an origin Rx_clk ($PS = 0^\circ$), an inversed Rx_clk ($PS = 180^\circ$), and two delayed Rx_clk ($PS = 54^\circ$ or 180°).

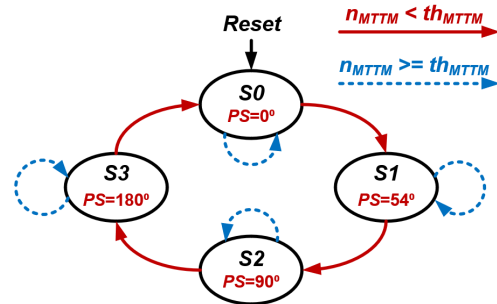


Fig. 12. Phase selector's FSM chart based on the round-robin algorithm.

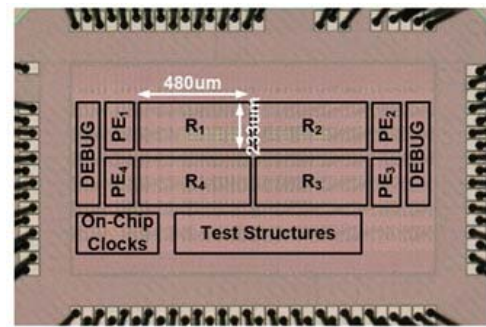


Fig. 13. Die photograph of the prototyped NoC test chip featuring MEDAC.

However, this metastability would only delay the metastability condition flag signal until the synchronizer is resolved from the metastable state. Since this signal is simply to reset the MTTM counter, it will just lead to a slightly incorrect counting for resetting (e.g., one cycle larger than the true value), and the MTTM counter would still function sufficiently well.

Fig. 12 shows the finite-state machine (FSM) diagram used in the PS selection. It is based on the round-robin algorithm.

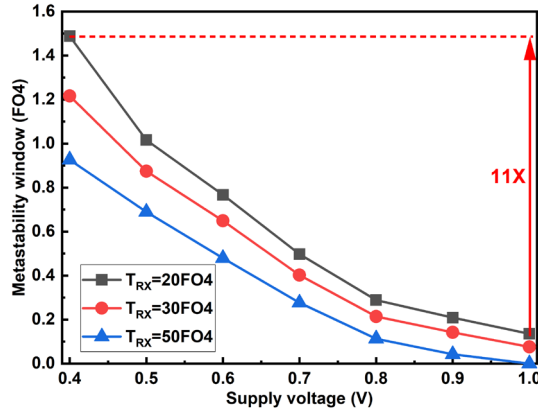


Fig. 16. Metastability window under different supply voltages and clock periods.

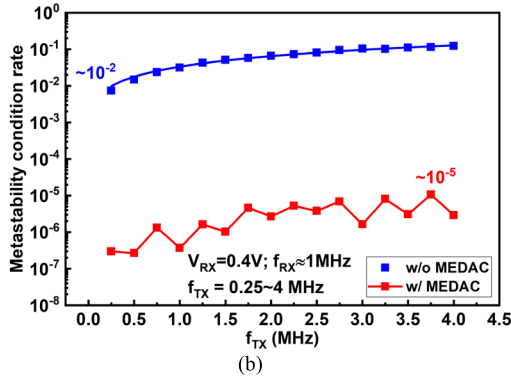
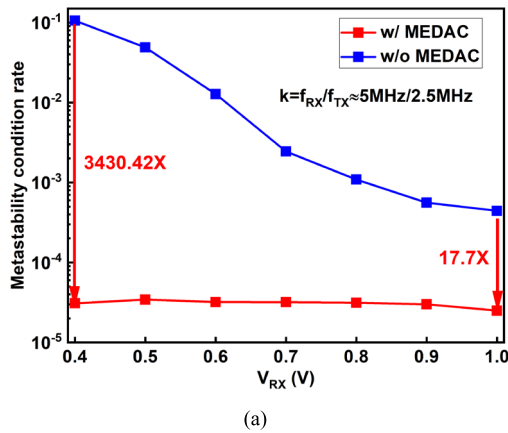


Fig. 17. Measured metastability condition rates across (a) different supply voltages and (b) f_{TX} values.

the results of W_M , with which we set $W_I = W_M$ for all chips in the experiment. It also shows that the voltage scaling from 1 to 0.4 V increases W_M (normalized to the FO4 delay of each supply voltage) by 11 times, confirming the growing importance of metastability in NTV circuits.

To study the impact of supply voltages on the metastability condition rate, i.e., the ratio of the total number of metastability condition cycles to the total number of clock cycles, we scale V_{RX} from 1.0 to 0.4 V. f_{RX} is fixed to 4.9995 MHz, considering ~ 100 -ppm frequency offset from nominal 5 MHz. f_{TX} is set to 2.5 MHz (i.e., $k \approx 2$). W_I is set conservatively based on the W_M from the Monte Carlo simulation. Fig. 17(a) shows the measured metastability condition rate

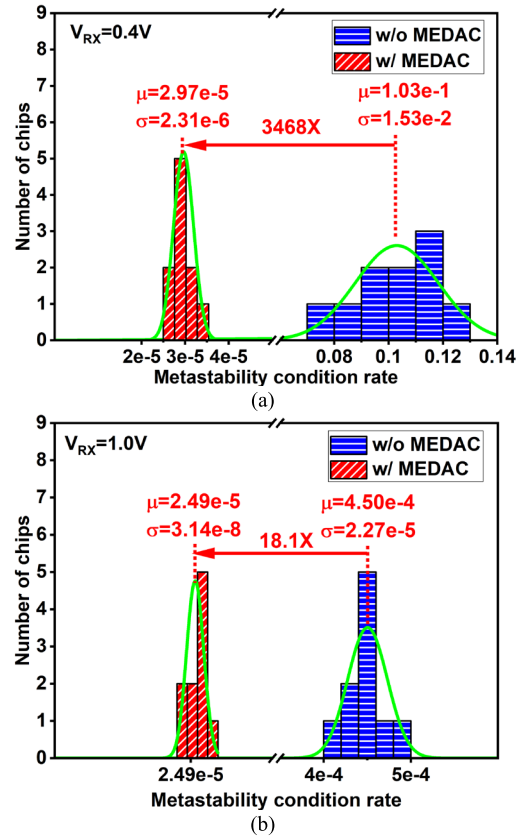
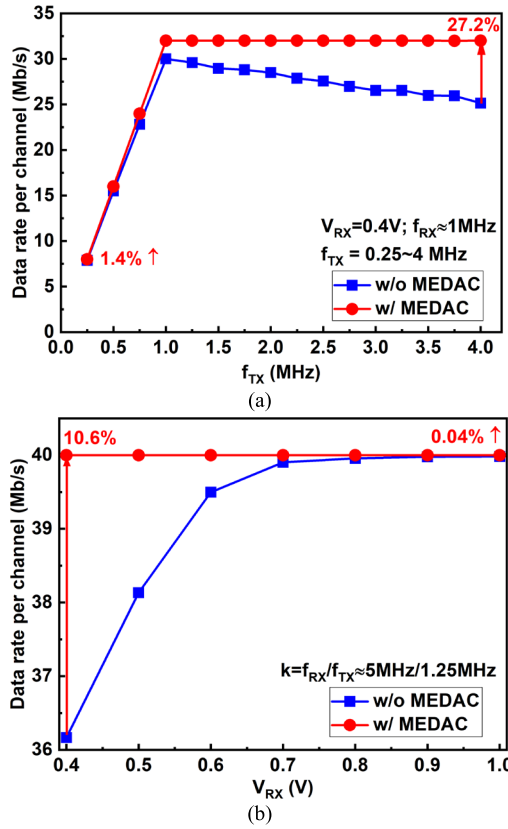


Fig. 18. Metastability condition rates across ten different dies. $k = f_{RX}/f_{TX} \approx 5M/2.5M$. (a) $V_{RX} = 0.4$ V. (b) $V_{RX} = 1.0$ V.

with or without the MEDAC technique. Without the MEDAC, voltage scaling increases the metastability condition rate from 4.43×10^{-4} all the way to 1.06×10^{-1} , by 239 $\times$. The MEDAC can reduce the metastability condition rate by 3430 $\times$ (17.7 $\times$) at $V_{RX} = 0.4$ V (1 V). Note that, in this prototype, at 1.0 V, the buffer chain that we employed to generate W_I has more than one order longer delay than W_M . This increases the metastability condition rate in the experiment.

Besides, we investigate the relationship between k and the metastability condition rate. We sweep k from 0.25 to 4 by fixing f_{RX} to 0.9999 MHz (nominal 1 MHz with 100-ppm frequency offset) and sweeping f_{TX} from 0.25 to 4 MHz. V_{RX} is fixed at 0.4 V. Fig. 17(b) shows the results of this experiment. Without the MEDAC technique, the metastability condition rates increase with f_{TX} . This is because the RX domain is more likely to capture a switching Tx_data and, therefore, suffers more metastability conditions when the TX domain operates at a high frequency. In this experiment, again, the MEDAC can reduce the metastability condition rate from $\sim 10^{-2}$ to $\sim 10^{-5}$.

We also measured the metastability condition rates across ten different dies. We set f_{RX} to 4.9995 MHz and f_{TX} to 2.5 MHz. Fig. 18(a) shows the histogram of the metastability condition rate under 0.4 V. Without the MEDAC, the metastability condition rate has an average value of 0.103 and a standard deviation of 1.53×10^{-2} . The MEDAC reduces the average metastability condition rate to 2.97×10^{-5} , by 3468 $\times$,

Fig. 19. Effective data rate across (a) f_{TX} values and (b) V_{RX} values.

and the standard deviation to 2.31×10^{-6} . Fig. 18(b) shows the histogram for $V_{RX} = 1.0$ V. Again, the MEDAC is able to reduce the average metastability condition rate by $18.1\times$.

We also investigate the data rate and energy efficiency benefits of the proposed long-term mitigation scheme when enabling a retransmission scheme. Fig. 19(a) shows the average data rate of communication. Here, we set V_{RX} to 0.4 V and f_{RX} to 0.9999 MHz. We sweep f_{TX} from 0.25 to 4 MHz (i.e., k is between 0.25 and 4). Without the long-term mitigation scheme, the data rate increases with f_{TX} until f_{TX} approaches f_{RX} , but the data rate saturates and actually decreases when further increasing f_{TX} . This is because the RX domain more frequently experiences metastability if Tx_data switches fast, thereby requesting more data retransmission. With the long-term mitigation scheme enabled, however, the measurement result shows that the NoC can almost maintain the maximum data rate (limited by f_{RX}), 27.2% higher than the case without the long-term mitigation when $k \approx 4$. Similarly, Fig. 19(b) shows the data rate improvement over V_{RX} s. The long-term mitigation improves the data rate by 10.6% at $V_{RX} = 0.4$ V, where the NoC experiences metastability conditions most frequently.

The data rate improvement benefits energy efficiency, as well. Fig. 20(a) shows that the MEDAC-enabled router achieves up to 21.1% energy efficiency improvement with k in the range from 0.25 to 4. Fig. 20(b) also shows energy efficiency improvement across V_{RX} from 1.0 to 0.4 V. The MEDAC reduces energy consumption by up to 9.6%.

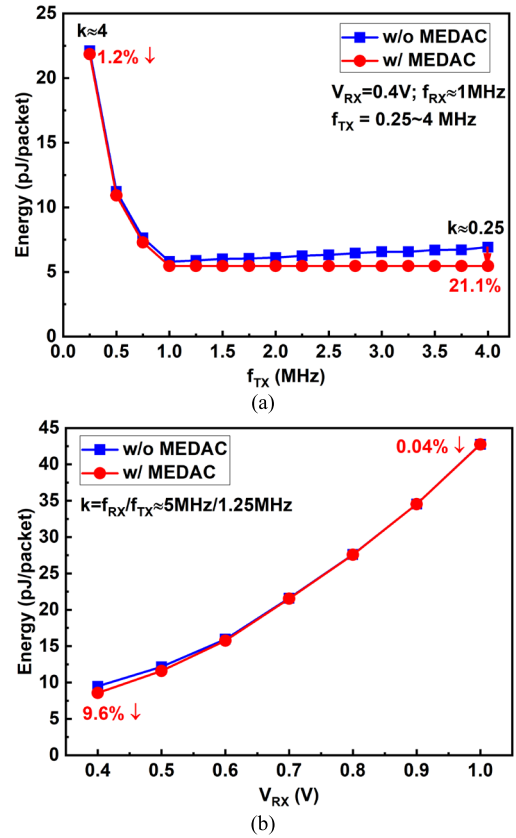
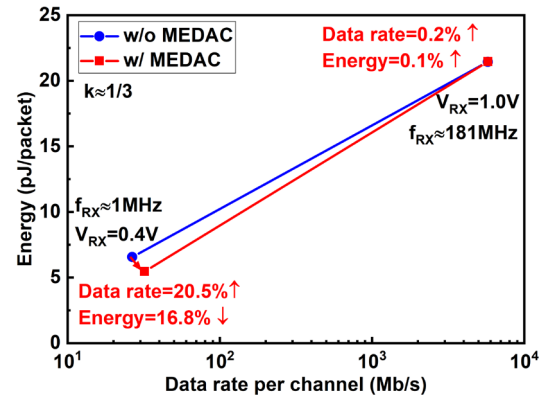
Fig. 20. Energy consumption per packet across (a) f_{TX} values and (b) V_{RX} values.

Fig. 21. Measured energy consumption and data rate for routers with or without MEDAC.

We also measure the data rate and energy efficiency improvement across a wider range of V_{RX} and f_{RX} from 0.4 V, 1 MHz to 1 V, 181 MHz. Fig. 21 shows the results, where the MEDAC offers up to 20.5% data rate improvement and up to 16.8% energy efficiency improvement. The improvement is smaller at 1.0 V because the metastability condition reduces exponentially with increasing supply voltage.

Finally, Table I summarizes the comparison to the recent NTV NoC work. Paul *et al.* [2] proposed the timing EDS and a flow control unit (FLIT) replay correction scheme embedded in a router for improving the resiliency against multi-bit timing

TABLE I
COMPARISONS WITH THE EXISTING NOC PROTOTYPE

	This work	[2]
Technology	65nm low-power	22nm Tri-gate
EDAC	No	Yes
MEDAC	Yes	No
Router area (mm ²)	0.112	0.051
Router power (mW)	0.07	1.3
Data rate improvement	27.2%	N/A
Metastability condition rate reduction	3,468X	N/A
Energy efficiency improvement	21.1%	14.6%
V/F domains	4	1

errors. Paul *et al.* [2] considered a single V/F domain and no specific attention to metastability. The proposed NoC is designed to support four V/F domains in GALS systems and features the MEDAC technique to mitigate metastability related issues.

VI. CONCLUSION

In this article, we propose the MEDAC technique to mitigate the metastability risks and improve the reliability of data transmission among multiple V/F domains, especially for NTV operation. We prototype a 2×2 GALS NoC test chip that contains four independent V/F domains using the proposed MEDAC technique in a 65-nm CMOS low-power process. The MEDAC-enabled NoC reduces the probability of experiencing metastability conditions by up to 3468 $\times$ compared with the baseline. This translates to up to 27.2% higher data rate and 21.1% less energy consumption over baseline. The area overhead of MEDAC is estimated as 4.4%. The proposed MEDAC technique will benefit the design of future multi-V/F near-threshold NoCs that demand both ultra-low power consumption and high reliability.

REFERENCES

- [1] J. P. Cerqueira, T. J. Repetti, Y. Pu, S. Priyadarshi, M. A. Kim, and M. Seok, "Catena: A 0.5-V Sub-0.4-mW 16-core spatial array accelerator for mobile and embedded computing," in *Proc. Symp. VLSI Circuits*, Jun. 2019, pp. 54–55.
- [2] S. Paul *et al.*, "A 3.6GB/s 1.3mW 400mV 0.051mm² near-threshold voltage resilient router in 22nm tri-gate CMOS," in *IEEE Symp. VLSI Circuits (VLSI)*, Jun. 2013, pp. C30–C31.
- [3] E. Beigne *et al.*, "An asynchronous power aware and adaptive NoC based circuit," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1167–1177, Apr. 2009.
- [4] P. Vivet *et al.*, "A 4×2 homogeneous scalable 3D network-on-chip circuit with 326 MFlit/s 0.66 pJ/b robust and fault tolerant asynchronous 3D links," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 33–49, Jan. 2017.
- [5] F. Vitullo *et al.*, "Low-complexity link microarchitecture for mesochronous communication in networks-on-chip," *IEEE Trans. Comput.*, vol. 57, no. 9, pp. 1196–1201, Sep. 2008.
- [6] R. W. Apperson, Z. Yu, M. J. Meeuwsen, T. Mohsenin, and B. M. Baas, "A scalable dual-clock FIFO for data transfers between arbitrary and halttable clock domains," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 10, pp. 1125–1134, Oct. 2007.
- [7] C. L. Portmann and H. Y. Meng, "Metastability in CMOS library elements in reduced supply and technology scaled applications," *IEEE J. Solid-State Circuits*, vol. 30, no. 1, pp. 39–46, Jan. 1995.

- [8] C. Foley, "Characterizing metastability," in *Proc. 2nd Int. Symp. Adv. Res. Asynchronous Circuits Syst.*, Mar. 1996, pp. 175–184.
- [9] A. Cantoni, J. Walker, and T.-D. Tomlin, "Characterization of a flip-flop metastability measurement method," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 5, pp. 1032–1040, May 2007.
- [10] D. J. Kinniment, A. Bystrov, and A. V. Yakovlev, "Synchronization circuit performance," *IEEE J. Solid-State Circuits*, vol. 37, no. 2, pp. 202–209, Feb. 2002.
- [11] J. Zhou, D. J. Kinniment, C. E. Dike, G. Russell, and A. V. Yakovlev, "On-chip measurement of deep metastability in synchronizers," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 550–557, Feb. 2008.
- [12] J. Zhou, M. Ashouei, D. Kinniment, J. Huisken, G. Russell, and A. Yakovlev, "Sub-threshold synchronizer," *Microelectron. J.*, vol. 42, no. 6, pp. 840–850, Jun. 2011.
- [13] J. Zhou, D. Kinniment, G. Russell, and A. Yakovlev, "A robust synchronizer," in *Proc. IEEE Comput. Soc. Annu. Symp. Emerg. VLSI Technol. Archit. (ISVLSI)*, Jun. 2006, pp. 442–443.
- [14] D. Li, P. I. Chuang, D. Nairn, and M. Sachdev, "Design and analysis of metastable-hardened flip-flops in sub-threshold region," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Des.*, Aug. 2011, vol. 1, no. 2, pp. 157–162.
- [15] D. Ernst *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. 22nd Digit. Avionics Syst. Conf. Process.*, Dec. 2003, pp. 7–18.
- [16] 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.
- [17] S. Kim and M. Seok, "Variation-tolerant, Ultra-Low-Voltage micro-processor 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.
- [18] D. Blaauw *et al.*, "Razor II: *In situ* error detection and correction for PVT and SER tolerance," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2008, pp. 400–402.
- [19] K. A. Bowman *et al.*, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 49–63, Jan. 2009.
- [20] 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.
- [21] F. Mu and C. Svensson, "Self-tested self-synchronization circuit for mesochronous clocking," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 2, pp. 129–140, Feb. 2001.
- [22] W. J. Dally and S. G. Tell, "The Even/Odd synchronizer: A fast, all-digital, periodic synchronizer," in *Proc. IEEE Symp. Asynchronous Circuits Syst.*, May 2010, pp. 75–84.
- [23] R. Ginosar, "Metastability and synchronizers: A tutorial," *IEEE Des. Test. Comput.*, vol. 28, no. 5, pp. 23–35, Sep. 2011.



Chuxiong Lin (Graduate Student Member, IEEE) received the B.S. degree in microelectronic science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017, where he is currently pursuing the Ph.D. degree in electronic science and technology.

His current research interests include variation tolerant circuit designs, ultra-low-power circuits, and system design.



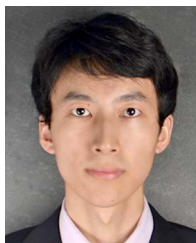
Weifeng He (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in microelectronics and solid-state electronics from the Harbin Institute of Technology, Harbin, China, in 1999, 2001, and 2005, respectively.

He is currently an Associate Professor with the Department of Micro-Nano Electronics, Shanghai Jiao Tong University, Shanghai, China. His current research interests include low-power circuits design, VLSI architecture for video signal processing, and computer system architecture.



Yanan Sun (Member, IEEE) received the B.E. degree in microelectronics from Shanghai Jiao Tong University, Shanghai, China, in 2009, and the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2015.

She is currently an Associate Professor with the Department of Micro-Nano Electronics, Shanghai Jiao Tong University, Shanghai, China. Her research area is low-power, high-performance, and variation-tolerant VLSI circuit and system design with emerging nanoscale device technologies.



Bingxi Pei received the M.S. degree in microelectronics and solid-state electronics from the National University of Defense Technology, Changsha, China, in 2017. He is currently pursuing the Ph.D. degree in electronic science and technology with Shanghai Jiao Tong University, Shanghai, China.

He was a Physical Design Engineer with Hisilicon, Shenzhen, China, from 2017 to 2018. His current research interest focuses on photoelectric hybrid switching network design.



Pavan Kumar Chundi received the B.Tech. degree in electrical engineering from IIT Delhi, New Delhi, India, in 2016. He is currently pursuing the M.S./Ph.D. degree with Columbia University, New York, NY, USA.

His interests include computer architectures and circuits for emerging cognitive systems.



Zhigang Mao (Member, IEEE) received the B.S. degree from Tsinghua University, Beijing, China, in 1986, and the Ph.D. degree from the University of Rennes 1, Rennes, France, in 1992.

From 1992 to 2006, he was with Microelectronics Center, Harbin Institute of Technology, Harbin, China. In 2006, he joined the Department of Micro-Nano Electronics, Shanghai Jiao Tong University, Shanghai, China, where he is currently a Professor. His current research interests include DSP architecture design, video processor design, and reconfigurable processor architecture.



Mingoo Seok (Senior Member, IEEE) 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 from the University of Michigan, Ann Arbor, MI, USA, in 2007 and 2011, respectively, all in electrical engineering.

He joined Texas Instruments Inc., Dallas, TX, USA, in 2011, as a Technical Staff Member. Since 2012, he has been with Columbia University, New York, NY, USA, where he is currently an

Associate Professor of Electrical Engineering. His research interests on VLSI hardware with the foci given to energy efficiency, artificial intelligence and machine learning, and hardware security.

Prof. Seok received the 1999 Distinguished Undergraduate Scholarship from the Korea Foundation for Advanced Studies, the 2005 Doctoral Fellowship from the Korea Foundation for Advanced Studies, the 2008 Rackham Pre-Doctoral Fellowship from the University of Michigan, the 2009 AMD/CICC Scholarship Award for picowatt voltage reference work, the 2009 DAC/ISSCC Design Contest for the 35-pW sensor platform design, the 2015 NSF CAREER Award, and the 2019 Qualcomm Faculty Award. He is/was a Technical Program Committee Member for several conferences including IEEE International Solid-State Circuits Conference (ISSCC) and IEEE Custom Integrated Circuits Conference (CICC). He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: REGULAR PAPERS from 2013 to 2015. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS since 2015 and IEEE SOLID-STATE CIRCUITS LETTERS since 2017. He also served as a Guest Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS.