A Metastability Risk Prediction and Mitigation Technique for Clock-Domain Crossing With Single-Stage Synchronizer in Near-Threshold-Voltage Multivoltage/ Frequency-Domain Network-on-Chip

Chuxiong Lin, Graduate Student Member, IEEE, Weifeng He, Senior Member, IEEE, Yanan Sun, Senior Member, IEEE, Lin Shao, Bo Zhang, Jun Yang, Member, IEEE, and Mingoo Seok, Senior Member, IEEE

Abstract— For a network-on-chip (NoC) with multiple voltage/frequency domains, metastability hurts the reliability during the clock-domain crossing, especially in the near-threshold-voltage (NTV) region. Conventional multistage synchronizers reduce the probability of metastability but have a high latency penalty. This article presents a technique titled metastability risk prediction and mitigation (MPAM) that predicts the near-future metastability risks by a triple-phase clock monitoring circuitry and mitigates them by a metastability-free clock scheme. Therefore, the MPAM enables only one flip-flop for data synchronization without degrading the reliability against metastability, thus improving the latency and throughput of NoC. We prototyped a 2-by-2 NoC test chip with four independent voltage/frequency domains in a 40-nm low-power (LP) process, featuring the MPAM technique. The measurement shows that the MPAM significantly reduces the metastability condition rate by 10^{10} times under different clock frequency ratios. Moreover, by enabling only one flip-flop for synchronization, the MPAM-based NoC achieves packet latency reduction, throughput improvement, and energy efficiency gain by 58%, 13.4%, and 8.6%, respectively.

Index Terms— Clock-domain crossing (CDC), metastability, multivoltage/frequency domain, near-threshold voltage (NTV), network-on-chip (NoC).

Manuscript received 25 November 2022; revised 22 March 2023; accepted 2 June 2023. This article was approved by Associate Editor Sanu Mathew. This work was supported in part by NSFC under Grant 62274106, in part by the National Key Research and Development Program of China under Grant 2018YFB2202004, in part by Semiconductor Research Corporation under Grant TxACE 2712.012, and in part by U.S. National Science Foundation under Grant CCF-1453142. (Corresponding author: Weifeng He.)

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

Bo Zhang and Mingoo Seok are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA.

Jun Yang is with the National ASIC Research Center, Southeast University, Nanjing 210096, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2023.3283961.

Digital Object Identifier 10.1109/JSSC.2023.3283961

I. Introduction

THE reliability of a system-on-chip (SoC) has been a critical concern for emerging applications like drones and autonomous vehicles, e.g., a drone typically requires its mean time between failures (MTBF) to be larger than tens of thousands of hours [1]. Such an SoC typically adopts a multicore or even many-core architecture, which employs a network-on-chip (NoC) [2], [3] to enable high throughput and energy-efficient data communication among cores. Each core of the SoC ideally operates in an independent voltage and frequency (V/F) domain to push energy efficiency. Meanwhile, the SoC usually adopts complex low-power (LP) techniques, such as globally asynchronous, locally synchronous (GALS) clocking [4], dynamic voltage and frequency scaling (DVFS) [2], and adaptive clocking [5].

However, these LP techniques could make the supply voltage and clock frequency of each V/F domain vary largely from super-threshold voltage to near-threshold voltage (NTV), and from several GHz to sub-MHz. As a result, two independent V/F domains of the NoC could have arbitrary and time-varying clock phase and frequency relationships. Unfortunately, such an unknown phase relationship brings risks of metastability to the NoC. Moreover, as the core operates in the NTV region for energy saving, the metastability issue becomes even worse since the metastability resolution time increases exponentially [6], [7], [8], and process, voltage, and temperature (PVT) variations greatly impact the circuit's delay variability, badly hurting the reliability of NoC.

There are plenty of prior works proposed to counter metastability. Using a multistage synchronizer [9], which composes of several flip-flops, can provide several clock cycles of metastability resolution time to mitigate metastability during clock-domain crossing (CDC), but it cannot guarantee metastability elimination and has a latency penalty. Optimizing the flip-flop structure [10], [11], [12] is an instinctive way to improve the synchronizer against metastability, however, at the cost of performance, area, and power. Instead, many

prior works [13], [14], [15], [16], [17], [18] actively control the receiver's sampling clock edge to avoid it becoming close to data arrival, thus mitigating metastability. Among these works, Dally and Tell [13] and Buckler and Burleson [14] use the measured clock frequency ratio to infer the next phase relationship and switch the receiver's clock phase accordingly, but the measurement of the clock frequency ratio is time-consuming. Some works [15], [16], [17] detect the occurrence of metastability and then modulate the receiver's clock, thus requiring the two clock domains to have the same or similar clock frequencies. Lin et al. [18] detect the necessary metastability condition and modulate the receiver's clock edge to enlarge the mean time between two metastability (MTTM) conditions, but: 1) it only detects the necessary condition instead of the actual metastability so that it tends to pessimistically alarm metastability; 2) it requires packet retransmission to correct potential errors, which degrades throughput and energy efficiency; and 3) it reduces the probability of entering metastability conditions by only two–three orders of magnitude; therefore, it has limited help to counteract the dramatically worsen metastability issue in the NTV region.

In this work, we aim to predict the near-future metastability risks and mitigate them before the metastability occurs, thus using a single flip-flop instead of a multistage synchronizer for low-latency data synchronization. To this end, we propose a metastability risk prediction and mitigation (MPAM) technique that mitigates the metastability risk in an NTV NoC, where different V/F domains have different clock frequencies and unknown phase relationships. MPAM exploits in situ error detectors (EDs) and three consecutive clocks with different phases to predict the metastability conditions on the synchronizing clock. Once predicted, MPAM selects one metastability-free clock among these three clocks for synchronization before the metastability strikes the clock edge. Therefore, the MPAM can avoid the upcoming metastability condition, let alone the actual metastability. Such effective metastability prevention allows us to use a flip-flop instead of a multistage synchronizer for synchronization, thus reducing transmission latency and energy consumption.

Based on the MPAM technique, we prototyped a 2-by-2 NoC test chip in a 40-nm LP process. The NoC integrates four independent V/F domains that operate widely from 0.4 V/1 MHz to 1 V/200 MHz. Each domain contains a processing element (PE) and an MPAM-based router. Measurement shows that the MPAM enables zero bit error rate (BER) for 10^{13} bits of data transmission on the NoC, reducing the metastability condition rate by at least 10^{10} times. Therefore, the NoC with MPAM enabled can reliably use a single flip-flop for synchronization, thus reducing packet latency by 34.5% (58%) as compared to that of a conventional 3(6)-stage synchronizer. The proposed technique also helps to improve the throughput of NoC by 7.4% (13.4%) and energy efficiency by 3.5% (8.6%). The area overhead of MPAM is 3.4%.

The remainder of this article is organized as follows. Section II introduces the metastability issue and our motivations. Section III describes the proposed MPAM technique and its design constraints. Section IV shows the architecture of MPAM-based NoC. Section V shows the measurement results. Finally, Section VI concludes this article.

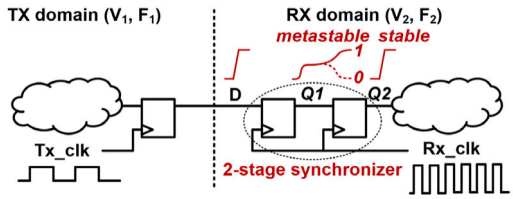


Fig. 1. Two-stage synchronizer for countering the metastability in CDC.

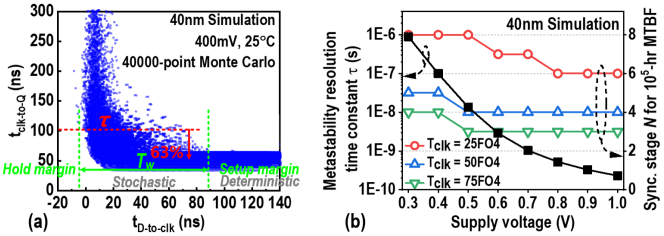


Fig. 2. (a) Two key parameters of metastability: the metastability resolution time constant τ and the metastability window T_W. (b) Simulated τ and required number of flip-flops for achieving 10^5-h MTBF under different supply voltages.

II. Metastability in CDC

A. Metastability Phenomenon

Metastability happens in a flip-flop when the data arrive very close to the sampling clock edge. During CDC, since the clock phase relationship between two domains is arbitrary, the receiving flip-flop is risky to have metastability issues. As shown in Fig. 1, the TX domain sends data D with a clock of Tx_clk to the RX domain with another clock of Rx_clk. The different clock frequencies of Tx_clk and Rx_clk result in an uncertain phase relationship between the two domains. Therefore, the data D may arrive very close to the rising edge of the Rx_clk, making the receiving synchronizer (flip-flop) becomes metastable, i.e., the output stays between 0 and 1. Specifically, as shown in Fig. 2(a), the metastability makes the flip-flop's delay, i.e., $t_{clk-to-Q}$, increases exponentially when the time between data arrival and the clock's edge, i.e., $t_{D-to-clk}$, is smaller than the metastability window T_W. The average $t_{clk-to-Q}$ when metastability occurs is defined as the metastability resolution time constant τ of a flip-flop [20]. The larger the τ is, the higher the possibility of metastability will be.

To avoid the metastable signal entering the combinational logic and incurring errors, a synchronizer with N-stage flip-flops is commonly used for data synchronization since it can provide at most N − 1 clock cycles for metastability resolution. As a result, the MTBF of an N-stage synchronizer can be formulated as follows [21]:

$$\mathrm{MTBF} = \frac{T_{TX} \cdot T_{RX}}{T_W \cdot e^{-(N-1)\cdot T_{RX}/\tau}} \quad (1)$$

where T_{TX} and T_{RX} are the clock period of Tx_clk and Rx_clk, respectively.

As the supply voltage tends to scale down as low as possible in nowadays SoCs, metastability becomes more critical.

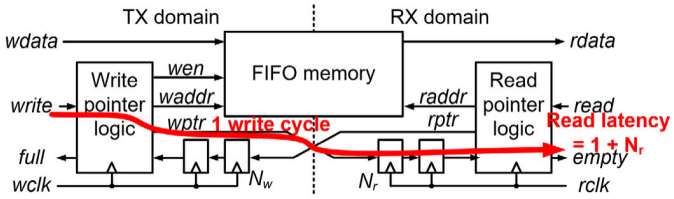


Fig. 3. Stage number N of a synchronizer greatly impacts the latency of CDC.

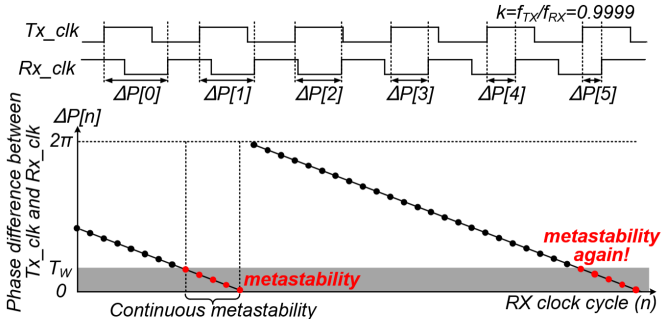


Fig. 4. Phase difference between the clocks of two clock domains changes over time, resulting in periodic metastability conditions.

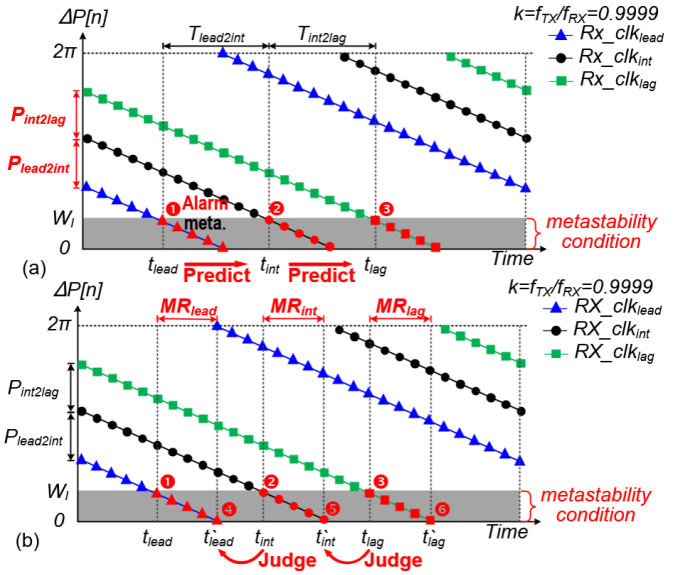


Fig. 5. Proposed metastability prediction scheme can: (a) alert the metastability condition in advance and (b) judge whether the metastability condition has passed.

As shown in Fig. 2(b), considering the process variations, the metastability resolution time constant τ increases by ⍰4000 times when the supply voltage scales from 1.0 down to 0.3 V. According to (1), to maintain the same reliability (e.g., MTBF of 10^5 h) under increased τ, the stage number N of the synchronizer should be up to 4 or even 8 in the NTV region.

While increasing the N of a synchronizer helps to maintain the reliability in the NTV region, it also greatly impacts the latency. Taking a bisynchronous FIFO equipped with N-stage synchronizers as an example, as shown in Fig. 3, the FIFO takes one cycle for the TX domain to write data, and then, it takes N_r cycles for the RX domain to synchronize and access the data, so the total read latency is

$$\text{read latency} = 1\text{TX cycle} + N_r\text{RX cycles}. \quad (2)$$

When the supply voltage reduces to the NTV region, the required N pessimistically grows, making the latency penalty of synchronization high to eight RX cycles. Therefore, mitigating the metastability during CDC is not only essential for enhancing reliability, but it is also important for improving data transmission latency and throughput.

B. Regularity in Metastability During CDC

While the relative clock phase is arbitrary during CDC, the change in the phase difference shows regularity [13], [19], [22]. As shown in Fig. 4, assuming the TX and RX have a slight clock frequency difference ($k = f_{TX}/f_{RX} = 0.9999$), then the clock phase difference, i.e., 1P[n], will change over the RX clock cycles n. As 1P[n] becomes smaller than the metastability window (T_W), the RX is going to have a metastability issue. In this case, we can conclude that the metastability has a regularity as: 1) the RX experiences metastability risks continuously in a short time and 2) with the clock phase changes over time, the metastability will return periodically, hurting the reliability.

Prior works, such as [13], [14], and [18], have tried to exploit phase regularity to mitigate the metastability during CDC. However, as they only initiate metastability mitigation post detecting the 1P[n] is small, the system is already at risk of metastability. Instead, by exploring the regularity as summarized above, we can use the current 1P[n] to infer the future clock phase difference, so that we can predict the metastability risk by clock phase monitoring. Once predicted, we can adapt the receiver clock in advance to always guarantee enough phase margins (i.e., large 1P[n]) between two domains for metastability-free data sampling and synchronization.

III. Metastability Risk Prediction and Mitigation

We proposed the MPAM technique that first predicts near-future metastability risks of synchronizers and then mitigates them before metastability happens.

A. Metastability Risk Prediction

To predict the metastability risk based on the disciplinary phase relationship between clock domains, we proposed a triple-phase clock monitoring scheme. It monitors the phase relationship between Tx_clk and three consecutive clocks (i.e., $\text{Rx\_clk}_{\text{lead}}$, $\text{Rx\_clk}_{\text{int}}$, and $\text{Rx\_clk}_{\text{lag}}$), which are all derived from Rx_clk with the smallest, the intermediate, and the largest phase addition, respectively. As shown in Fig. 5(a), the three derived Rx_clks have different 1P[n] to Tx_clk. Moreover, the clock phase of $\text{Rx\_clk}_{\text{lead}}$ advances that of $\text{Rx\_clk}_{\text{int}}$ by P_{lead2int}, and $\text{Rx\_clk}_{\text{int}}$ advances $\text{Rx\_clk}_{\text{lag}}$ by P_{int2lag}. We can use an advanced clock to predict the metastability risk of a laggard clock. To do so, we detect whether the 1P[n] of the derived Rx_clk is smaller than a

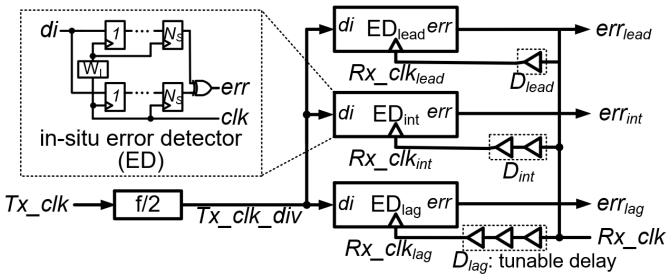


Fig. 6. Triple-phase clock-based metastability risk prediction hardware predicts the metastability condition with three consecutive Rx_clks.

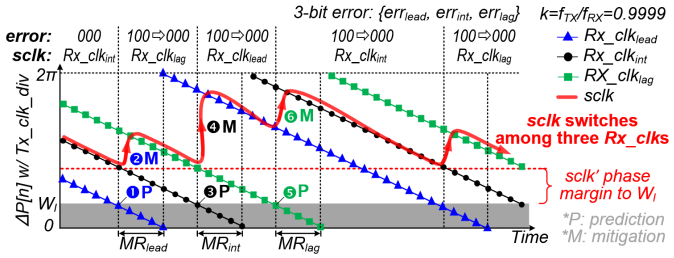


Fig. 7. Synchronization clock sclk dynamically switches among the three Rx_clks to mitigate the metastability.

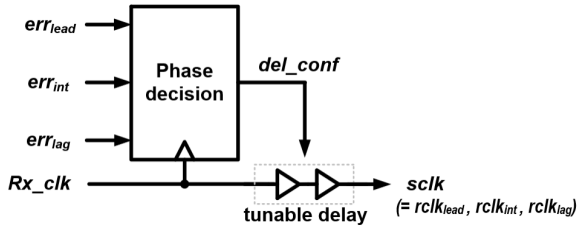


Fig. 8. Proposed metastability mitigation hardware selects a proper phase for sclk to avoid metastability conditions.

metastability detection window W_I by an in situ ED [19]. Here, we set W_I larger than T_W for detecting metastability conditions rather than detecting the actual metastability. As a result, when the ED alarms the metastability condition of Rx_clk_{lead} at t_{lead} (1), we can predict that Rx_clk_{int} is also going to have metastability condition in the near-future moment t_{int} (2). Similarly, the alarm at t_{int} (2) can be used to predict the metastability condition of Rx_clk_{lag} at t_{lag} (3). Note that the prediction advances the upcoming metastability condition by $T_{lead2int}$ or $T_{int2lag}$, which can be adjusted by setting proper phase difference ($P_{lead2int}$ or $P_{int2lag}$) for supporting different frequency relationships during CDC.

On the other hand, the proposed prediction scheme also judges whether the Rx_clk is metastability-free for the next period. As shown in Fig. 5(b), $1P[n]$ of the Rx_clk_{lead}, Rx_clk_{int}, and Rx_clk_{lag} is smaller than the metastability detection window W_I during the metastability risk region MR_{lead}, MR_{int}, and MR_{lag}, respectively. The three metastability regions are nonoverlapped by setting proper $P_{lead2int}$ and $P_{int2lag}$. As a result, when the ED triggered by Rx_clk_{int} alarms metastability condition at t_{int} (2), we can judge that Rx_clk_{lead} is metastability-free since Rx_clk_{lead} has already left MR_{lead} at t'_{lead} (4). Similarly, the alarm generated at t_{lag} (3) can be a flag that Rx_clk_{int} is metastability-free in the next period.

Based on the above scheme, Fig. 6 shows the metastability risk prediction hardware that uses three in situ EDs [19] to monitor the relative phase relation between Tx_clk and Rx_clk. The three EDs, i.e., ED_{lead}, ED_{int}, and ED_{lag}, are triggered by three consecutive Rx_clks, i.e., Rx_clk_{lead}, Rx_clk_{int}, and Rx_clk_{lag}, which are generated from Rx_clk by tunable delay cells. Each ED uses two N_S-stage synchronizers to sample the input data di within a metastability detection window interval W_I. The ED compares the two samples and alerts an error signal err if di arrives within W_I. Therefore, the three EDs generate error signals, i.e., err_{lead}, err_{int}, and err_{lag}, if Rx_clk_{lead}, Rx_clk_{int}, and Rx_clk_{lag} arrive too close to Tx_clk_div, which is the frequency-divided Tx_clk. By configuring the tunable delay cells properly (which will be discussed in Section III-C), the three EDs generate error signals at different time regions, with which we can predict and judge the metastability condition for the three derived Rx_clks.

B. Metastability Risk Mitigation

After predicting the potential metastability risks, we further proposed a metastability risk mitigation scheme to avoid actual metastability in advance by dynamically choosing one metastability-free derived Rx_clk for synchronization. As shown in Fig. 7, the three derived Rx_clks have $1P[n]$ larger than the metastability detection window interval W_I, so the three EDs, i.e., ED_{lead}, ED_{int}, and ED_{lag}, output "000" for indicating no errors. Therefore, we select Rx_clk_{int} as the synchronizing clock sclk. As time goes by, $1P[n]$ of Rx_clk_{lead} first becomes smaller than W_I, so ED_{lead} alarms the case (1), with which we can predict the Rx_clk_{int} is going to have metastability condition soon (3) but Rx_clk_{lag} is relatively distant to the metastability risk region MR_{lag} (5). Therefore, we can mitigate the near-future metastability risk by selecting Rx_clk_{lag} as srclk (2) for the next period. Later on, $1P[n]$ of Rx_clk_{int} becomes smaller than W_I during MR_{int} (3), where ED_{int} alarms the upcoming metastability risks of Rx_clk_{lag} (5). Hence, the synchronizing clock sclk switches from Rx_clk_{lag} to Rx_clk_{lead} (4) to mitigate the potential risks. Similarly, when ED_{lag} gives an alarm (5), sclk switches to Rx_clk_{int} (6) for metastability mitigation. By such continuous metastability risk prediction and dynamic clock switching, the synchronizing clock sclk always keeps a sufficient phase margin to the metastability detection window W_I, thus avoiding any metastability condition, letting alone actual metastability.

Based on the presented scheme, as shown in Fig. 8, the metastability risk mitigation hardware contains a phase decision block that uses the generated error signals to decide a metastability-free clock phase dynamically and then selects the corresponding Rx_clk for synchronization.

C. Timing Constraints of Metastability Prediction and Mitigation

To ensure that the proposed MPAM technique can predict the potential metastability condition in advance and mitigate it by selecting one metastability-free Rx_clk before the metastability really happens, we need to properly set the phase difference (i.e., $P_{lead2int}$ and $P_{int2lag}$) among the three derived Rx_clks by configuring the tunable delay cells.

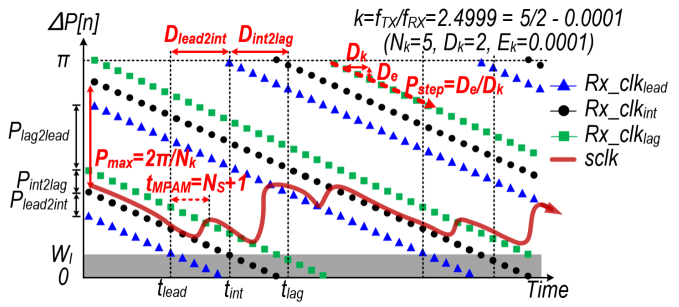


Fig. 9. Requirements of $D_{lead2int}$ and $D_{int2lag}$ between the three derived Rx_clks.

First of all, $P_{lead2int}$ and $P_{int2lag}$ should be large enough to give MPAM sufficient time to act on prediction and mitigation before actual metastability happens. As shown in Fig. 9, the phase differences $P_{lead2int}$ and $P_{int2lag}$ make the Rx_clk_{lead}, Rx_clk_{int}, and Rx_clk_{lag} meet metastability conditions at different moments, i.e., t_{lead}, t_{int}, and t_{lag}, respectively. When the metastability condition happens on Rx_clk_{lead} at t_{lead}, the MPAM latency t_{MPAM} includes N_S clock cycles for the ED to alarm an error and one clock cycle for the mitigation hardware to switch sclk from Rx_clk_{int} to Rx_clk_{lag}, i.e., $t_{MPAM} = N_S + 1$. The clock switching must complete before Rx_clk_{int} meets the metastability condition at t_{int}. Therefore, t_{MPAM} should be less than the timing interval between t_{int} and t_{lead}, i.e., $T_{lead2int} (= t_{int} - t_{lead})$. Similarly, t_{MPAM} should also be smaller than $T_{int2lag} (= t_{lag} - t_{int})$. It can be formulated as follows:

$$\min\left(T_{lead2int}, T_{int2lag}\right) \geq N_S + 1. \tag{3}$$

Here, the timing intervals (i.e., $T_{lead2int}$ and $T_{int2lag}$) depend on the phase differences (i.e., $P_{lead2int}$ and $P_{int2lag}$) between the derived Rx_clks, which can be calculated as follows:

$$T_{lead2int} = \frac{P_{lead2int}}{P_{step}}, \quad T_{int2lag} = \frac{P_{int2lag}}{P_{step}} \tag{4}$$

where P_{step} is the average phase step of $\Delta P[n]$. Then, (3) can be rewritten as

$$\min\left(P_{lead2int}, P_{int2lag}\right) \geq (N_S + 1) \cdot P_{step}. \tag{5}$$

The average phase-changing step P_{step} relates to the frequency ratio k between two clock domains. The k can be formulated as a rational fraction with a bounded denominator plus an error term [13]

$$k = \frac{f_{TX}}{f_{RX}} = \frac{N_k}{D_k} + E_k \tag{6}$$

where N_k and D_k are two coprime integers for representing the nominator and dominator of k, respectively, and E_k is the error term. Then, P_{step} can be formulated as [19]

$$P_{step} = \frac{\Delta P[n + D_k] - \Delta P[n]}{D_k} \tag{7}$$

$$\Delta P[n] = \Delta P[0] + \left(n - \frac{1}{k}\right) \cdot m \cdot 2\pi \tag{8}$$

$$m = \left\lceil \frac{\Delta P[0]}{2\pi} + n + \frac{W_M}{2\pi} \right\rceil \cdot k. \tag{9}$$

With (7)–(10), we can approximate P_{step} as

$$\begin{aligned} P_{step} &= \frac{1}{D_k} - \frac{1}{k} \cdot \left\lceil \frac{\Delta P[0]}{2\pi} + n + D_k + \frac{W_M}{2\pi} \right\rceil \cdot k \\ &\quad - \left\lceil \frac{\Delta P[0]}{2\pi} + n + \frac{W_M}{2\pi} \right\rceil \cdot k \cdot 2\pi \\ &\approx \frac{1}{D_k} - \frac{1}{k} \cdot (\lceil (n + D_k) \cdot k \rceil - \lceil n \cdot k \rceil) \cdot 2\pi \\ &\approx E_k \cdot \frac{N_k}{D_k} \cdot 2\pi. \end{aligned} \tag{10}$$

In addition, $P_{lead2int}$ and $P_{int2lag}$ should be larger than the metastability detection W_I so that the three derived Rx_clks can have nonoverlapping metastability risk regions (MR_{lead}, MR_{int}, and MR_{lag}). Therefore, (5) can be extended as

$$\min\left(P_{lead2int}, P_{int2lag}\right) \geq \max\left((N_S + 1) \cdot P_{step}, W_I\right). \tag{11}$$

At the same time, the phase difference between Rx_clk_{lag} and Rx_clk_{lead}, i.e., $P_{lag2lead}$, should also be larger than the metastability window interval W_I to avoid MR_{lag} overlap with MR_{lead}, which can be formulated as

$$P_{lag2lead} = \frac{2\pi}{N_k} - P_{lead2int} - P_{int2lag} \geq W_I \tag{12}$$

where $2\pi/N_k$ is the maximum step of $\Delta P[n]$ in consecutive clock cycles [13]. Based on (12), $P_{lead2int}$ and $P_{int2lag}$ should satisfy the following constraint:

$$P_{lead2int} + P_{int2lag} \leq \frac{2\pi}{N_k} - W_I. \tag{13}$$

In conclusion, the delay of the tunable delay cells (D_{lead}, D_{int}, and D_{lag}) should make the phase difference, i.e., $P_{lead2int}$ and $P_{int2lag}$ satisfy the constraints of (11) and (13) to ensure the MPAM can predict and mitigate the metastability condition in advance. For example, if $k = 2.4999$, $N_S = 3$, and W_I is 0.03π, we can rewrite k into $k = 5/2 - 0.0001$ (i.e., $N_k = 5$, $D_k = 2$, and $E_k = -0.0001$). Then, $P_{lead2int}$ and $P_{int2lag}$ should be between 0.03π and 0.185π. Therefore, when the circuit operates under a typical clock period T = 20 FO4 (fan-out-of-four inverter delay), $P_{lead2int}$ and $P_{int2lag}$ translate to latency difference $D_{lead2int}$ $(= D_{int} - D_{lead})$ and $D_{int2lag}$ $(= D_{lag} - D_{int})$ that can be any value within 0.3 FO4 – 1.85 FO4. This relaxed delay requirement improves the tolerance of the tunable delay cells against PVT variations.

IV. MPAM-Based NoC Architecture

Based on the proposed MPAM technique, we designed a 2 × 2 GALS NoC. Fig. 10(a) shows the architecture of the proposed MPAM-based 2 × 2 NoC. The NoC contains four PEs (PE_0–PE_3) and four routers (R_0–R_3). The four PEs operate in four independent V/F domains. To communicate with PEs in other V/F domains, each PE has a local link to the adjacent router that shares the supply voltage and the ring-oscillator-based clock generator. Each PE can generate data packets under various packet injection rates from 10% to 100% and different traffic patterns. Moreover, the PE can

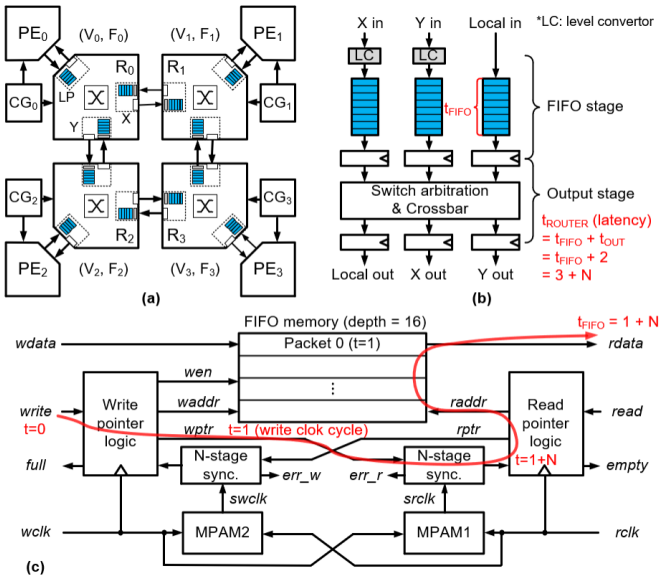


Fig. 10. (a) Proposed MPAM-based 2-by-2 GALS NoC architecture. (b) Two-stage router in the NoC contains. (c) MPAM-based FIFOs for mitigating metastability.

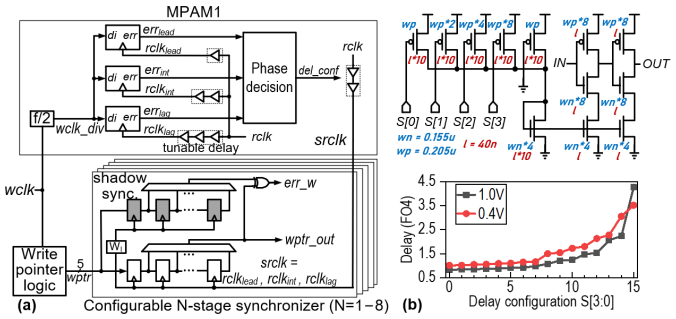


Fig. 11. (a) MPAM block with configurable N-stage pointer synchronizers. (b) Current-starve-based tunable delay cells that have a configurable delay.

also check the correctness of received packets via signature analysis. Each packet has to travel through several routers to its destination. Inside a packet, it contains a field of timing stamps that record the time when the packet is generated. As a result, the destined PE can calculate the packet latency when receiving the packet.

As shown in Fig. 10(b), the router collects packets from different input ports and forwards them to corresponding destinations. For V/F domain crossing, the router converts the packets to its local voltage domain via level converters and buffers the packets via dual-clock FIFOs. Then, the router allocates the packet to a corresponding output port in the second output stage. We can formulate a typical packet latency per router as $t_{ROUTER} = t_{FIFO} + t_{OUT}$, where t_{FIFO} is the latency of a dual-clock FIFO and t_{OUT} is the latency of the router datapath. t_{OUT} is the two clock cycles in our NoC.

As shown in Fig. 10(c), the MPAM-based dual-clock FIFO still contains write/read pointer logic, a dual-port SRAM, and two N-stage synchronizers. We add two MPAM blocks, i.e., MPAM1 and MPAM2, to avoid the metastability of the synchronizer. Each MPAM block can be shared by a 5-bit pointer synchronizer to amortize the power overhead. The latency of FIFO t_{FIFO} can be formulated as $t_{FIFO} = 1 + N$, where "1" is the one write clock cycle for the writer pointer updating, and N is the latency of the N-stage synchronizer. Therefore, the router latency is $t_{ROUTER} = 3 + N$, where the synchronizer's stage N constitutes a large portion of the total latency (40% if N = 2 and 57% if N = 4).

The two MPAM blocks monitor the phase difference to predict the metastability condition and to adaptively adjust the clock phase of the synchronizer to avoid the metastability condition. MPAM1 is for the read domain and MPAM2 is for the write domain. Let us take the MPAM1 as an example. As shown in Fig. 11(a), the MPAM1 block receives the write clock wclk from the write domain and monitors the phase difference by three in situ EDs with three derived rclks. Based on the generated errors, the phase decision block configures the phase of the synchronizing clock srclk to that of the metastability-free clock for the pointer synchronizers. To investigate whether the synchronizer is indeed safe from becoming metastable, we add an extra shadow synchronizer that samples the write pointer wptr after a metastability window interval W_I and compares the sample with that of the main synchronizer. If wptr arrives close to srclk within the window W_I, the shadow synchronizer alarms the error err_w to indicate the occurrence of metastability condition. For studying the impacts of stage number N on router latency, the synchronizer can be configured to have various N that ranges from 1 to 8. Note that the derived clocks are only used in the MPAM block and the pointer synchronizers, whereas the sequential elements in the FIFO stage and the output stage still use the origin rclk. Therefore, generating the three derived clocks has little impact on the global clock tree design.

The phase difference between the derived clocks (i.e., $P_{lead2int}$ and $P_{int2lag}$) of the MPAM should follow the design constraints as discussed in Section III-C. To obtain the target phase difference, as shown in Fig. 11(b), we implement the tunable delay unit based on the current-starve buffer, which has a configurable delay ranging from 1.1 FO4 to 3.46 FO4 under 0.4 V or ranging from 0.88 FO4 to 4.1 FO4 under 1 V to support metastability mitigation under 13 typical frequency ratios [18].

V. Measurement Results

We prototyped the NoC test chip in a 40-nm LP CMOS process. Fig. 12 shows the die photograph of the test chip. The MPAM blocks take 3.4% area overhead of the router's area.

Fig. 13 shows the measurement setup for the NoC test chip. The NoC is measured under two typical traffic patterns: the hotspot traffic pattern [Fig. 13(a)] that PE_0 receives packets from all the other three PEs, and the bit transpose traffic pattern [Fig. 13(b)] that each PE only communicates with one neighbor PE. We initiate the data traffic for transmitting 10^{13}-bit data. We measure the metastability condition rate, packet latency, throughput, and power consumption of the NoC during the data transmission when disabling MPAM and enabling MPAM, respectively.

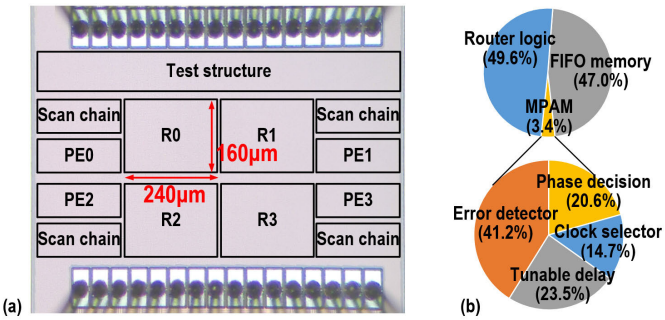


Fig. 12. (a) Die photograph of the proposed MPAM-based NoC test chip. (b) Area breakdown of the router and MPAM block.

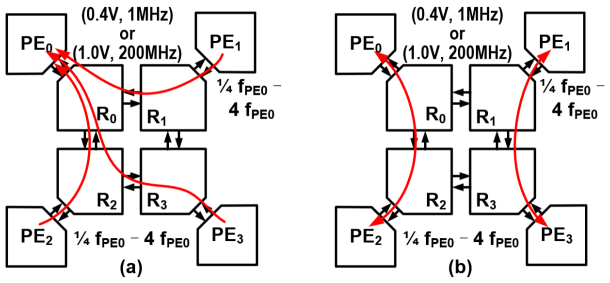


Fig. 13. NoC is measured under two typical data traffic patterns: (a) hotspot pattern and (b) bit transpose pattern.

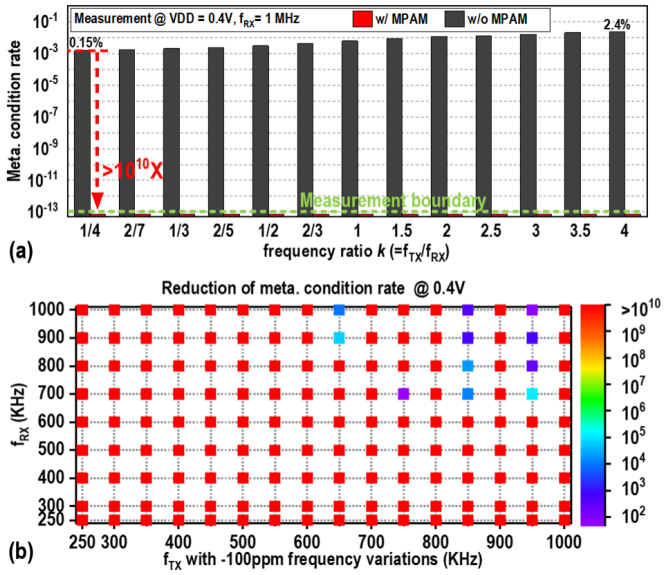


Fig. 14. (a) Measured metastability condition rate when disabling and enabling MPAM across 13 frequency ratios. (b) Reduction of metastability condition rate across clock frequencies.

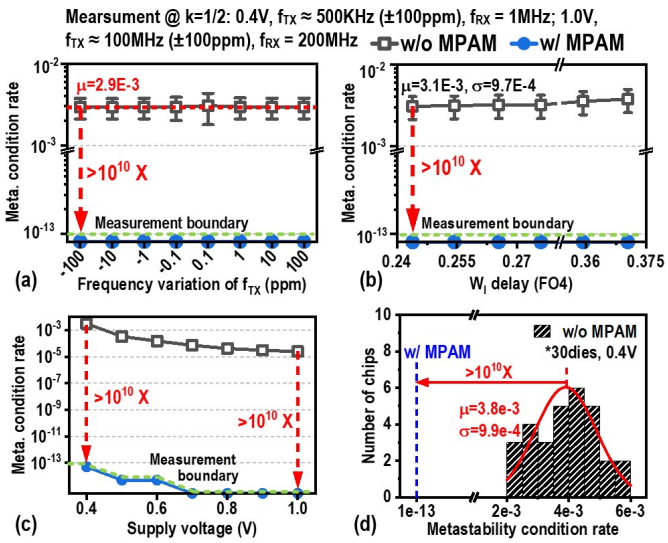


Fig. 15. Robustness of MPAM under: (a) clock frequency variations; (b) metastability detection window variations; (c) different supply voltages from 0.4 to 1 V; and (d) 30 different dies.

To investigate the capability of MPAM for mitigating metastability, we measure the metastability condition rate under a variety of clock frequency relationships. According to the design constraints, the tunable delay units in the MPAM are designed to support 13 typical frequency ratios just like [18]. As shown in Fig. 14(a), we measure the NoC under these supported frequency ratios when disabling MPAM and enabling MPAM. PE_0 operates under 0.4 V and 1 MHz. The clock frequency of the sender PE, i.e., f_{TX}, is swept from 250 kHz to 4 MHz. For each frequency ratio, we considered the frequency variation of ±100 ppm for f_{TX} and measured the average metastability condition rate. When disabling MPAM, the metastability condition rate increases from 0.15% to 2.4% as increasing the frequency ratio. While enabling the MPAM, the NoC is measured to have zero metastability condition and error during 10^{13}-bit data transmission, indicating that MPAM can reduce the metastability condition rate by at least ten orders of magnitude. As shown in Fig. 14(b), we further measure the reduction of metastability condition at 0.4 V across a wider frequency range. When sweeping both f_{TX} and f_{RX} from 250 kHz to 1 MHz, for most f_{TX} and f_{RX} pairs, the MPAM can still meet the timing constraints (see Section III-C), thus reducing the metastability condition rate by $>10^{10}$ times.

However, the tunable delay cells are not designed to meet the timing constraints of all frequency ratios since the required delay under some frequency ratios is very small. For example, for the frequency ratio $k = f_{TX}/f_{RX} \approx 950$ K/1000 K, the required delay range should be 0.3 FO4 – 0.38 FO4. In such cases, the MPAM can only reduce the metastability condition by □10^2–10^5 times. To avoid data errors, we can either expand the delay range of the tunable delay cells further or use the error signal by the shadow synchronizers to initiate packet retransmission as [18] does.

Since MPAM relies on the advanced prediction of metastability conditions rather than accurate metastability detection, MPAM is robust against variations. When the frequency ratio is near 1/2, MPAM can maintain 10^{10} times of reduction in the metastability condition rate even when the clock frequency has ±100-ppm variations [Fig. 15(a)] or the metastability detection window W_l has ±20% delay variations [Fig. 15(b)]. As shown in Fig. 15(c), the metastability condition rate increases greatly as the supply voltage scales down without MPAM. Enabling MPAM reduces the metastability condition rate by $>10^{10}$times under 0.4–1 V. As shown in Fig. 15(d), MPAM maintains the performance across 30 dies.

TABLE I
COMPARISONS WITH THE EXISTING NOC PROTOTYPE

	VLSI'13 [23]	JSSC'17 [2]	JSSC'21 [24]	ISSCC'20 [18]	This work
Technology	22nm Tri-gate	65nm	28nm	65nm LP	**40nm LP**
NoC	2×2	4×4	2×3	2×2	**2×2**
V/F domains	1	4	6	4	**4**
Operating range	0.34V–0.85V	0.65V–1.2V	0.5V–1.1V	0.5V–1V	**0.4V–1V**
Max clock frequency	151MHz–1GHz	50MHz–250MHz	130MHz–1.15GHz	7.3MHz–175MHz	**2.5MHz–416.7MHz**
Router power (mW)	1.3–28.5	-	-	0.07@0.5V	**0.046–10.58**
Design goal	Timing error	Packet congestion	Low latency and flexible communication	Metastability	**Metastability**
Proposed scheme	Error detection	Communication pattern prediction	Active interposer with distributed scalable NoC	Metastability condition detection	**Metastability risk prediction**
	Flit replay	Intelligent task scheduler		Post-metastability clock modulation	**Pre-metastability clock modulation**
Metastability condition rate reduction	-	-	-	500 – 1600× @ 0.5V 6.5 – 40× @ 1V	**>10^{10} ×**
Latency reduction	Flit replay increases latency	-	Accumulation of clock-domain crossing increases latency	Retransmission increases latency	**34.5% (over 3-stage)** **58.0% (over 6-stage)**
Throughput improvement	-	31%	-	19.5%*	**7.4% (over 3-stage)** **13.4% (over 6-stage)**
Energy efficiency improvement	14.6%	-	-	21.1%*	**3.5% (over 3-stage)** **8.6% (over 6-stage)**
Area overhead	2.8%	-	-	4.4%	**3.4%**

* Compared to a baseline with the conservative data retransmission scheme.

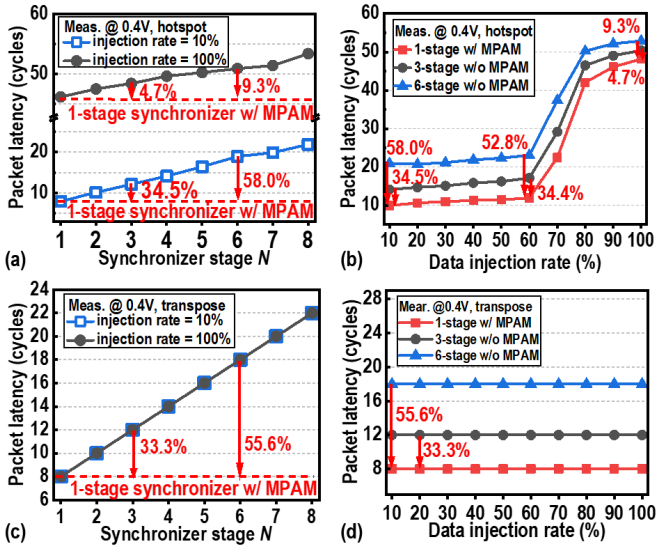


Fig. 16. Packet latency under the hotspot and bit transpose traffic pattern with: (a) and (c) different synchronizer stages and (b) and (d) different data injection rates.

Due to the significant metastability condition mitigation capability of MPAM, we aim to replace the conventional multistage synchronizer in NoC with only one flip-flop to reduce packet latency without scarifying reliability. For a conventional NoC with a multistage synchronizer, we found that the synchronizer typically needs to have three or six stages to achieve enough MTBF (i.e., 10^5 h) for applications like drones [see Fig. 2(b)]. As shown in Fig. 16(a), when the NoC operates under the hotspot data pattern, the packet latency increases linearly with the increase of the synchronizer's stage number. With a 10% packet injection rate, the MPAM-enabled NoC configured with the one-stage synchronizers reduces the packet latency by 34.5% (58%) as compared to that under a 3(6)-synchronizer. When increasing the packet injection rate, as shown in Fig. 16(b), the average packet latency first keeps constant and then increases due to packet congestion in the NoC. At a 100% packet injection rate, since the latency is not dominated by the synchronizers but by the congestions, the MPAM-enabled NoC reduces the latency by only 4.7% (9.3%) as compared to that under a 3(6)-stage synchronizer. When the NoC transmits data under the transpose data pattern, as shown in Fig. 16(c), the packet latency is dominated by the synchronizers. As shown in Fig. 16(d), the MPAM-enabled NoC reduces the latency by 33.3% (55.6%) as compared to that under a 3(6)-synchronizer with different packet injection rates.

The MPAM also helps to improve the throughput of NoC by reducing the synchronization stages. As shown in Fig. 17(a), at a low packet injection rate, e.g., 10%, the throughput is less related to the synchronizer stage. As the data injection

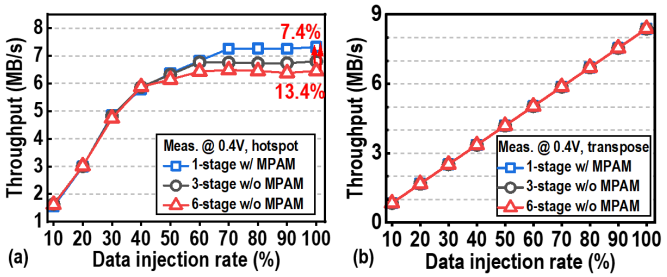


Fig. 17. Throughput of NoC under: (a) hotspot traffic pattern and (b) bit transpose traffic pattern.

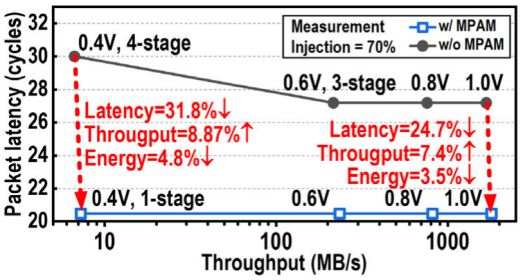


Fig. 18. MPAM improves energy efficiency, latency, and throughput of NoC across supply voltages.

rate goes up, the packets tend to be congested in the hotspot router (i.e., R_0). Since reducing the synchronizer stage can help quickly grant the FIFO access, the MPAM-enabled NoC shows 7.4% (13.4%) higher throughput than that of the 3(6)-stage synchronizer. As shown in Fig. 17(b), the throughput of the NoC with the bit transpose traffic pattern increases linearly with the increase in packet injection rates. Since the NoC has no packet congestion under the bit transpose pattern, it maintains the throughput under different synchronizer stage numbers.

MPAM can improve the throughput and energy efficiency of NoC from NTV to super-threshold voltage. As shown in Fig. 18, when the NoC operates under a supply voltage of 0.4–1.0 V, the MPAM reduces the latency by 31.8% and improves throughput by 8.87% at 0.4 V, and it can also effectively reduce the latency by 24.7% and improve throughput by 7.4% at 1.0 V. Despite the MPAM hardware bringing extra power overhead, the improved throughput reduces the data transmission time, thus also benefiting system energy efficiency across supply voltages.

Finally, Table I summarizes the comparisons of recent NTV NoC works. Prior works [2] and [23] optimize system performance by detecting setup violations or adopting an intelligent task scheduler, which is orthogonal with our proposed scheme. Vivet et al. [24] proposed a distributed scalable NoC to provide low latency and flexible communication among chiplets. However, its CDC latency is still a critical part of packet latency. Metastability condition detection and correction (MEDAC) [18] performs post-metastability clock modulation to reduce the metastability condition rate by three orders of magnitude but increases the latency. In addition, as MEDAC requires data retransmission when detecting the necessary metastability conditions, it tends to overly ask for data retransmission, hurting system throughput. In contrast, MPAM performs metastability prediction and mitigation to reduce the metastability condition greatly by 10^{10} times in a multi-V/F domain NoC at 0.4–1 V. It enables a single flip-flop for low-latency synchronization during CDC. MPAM also improves the throughput and energy efficiency of NoC.

VI. Conclusion

In this article, we propose the MPAM technique to mitigate the metastability risks and reduce the excess synchronization latency during data transmission among multiple V/F domains, especially for NTV operations. We prototype a 2 × 2 GALS NoC test chip that includes four independent V/F domains using the proposed MPAM technique in a 40-nm CMOS LP process. The MPAM reduces the metastability condition rate by at least 10^{10} times, thus enabling a single flip-flop for synchronization during CDC without hurting the reliability against metastability. As a result, the MPAM-enabled NoC reduces packet latency by 34.5% (58%) as compared to that of a 3(6)-stage synchronizer. It also benefits the NoC under heavy data traffic in terms of throughput by 7.4% (13.4%) and energy efficiency by 3.5% (8.6%). The area overhead of MPAM is only 3.4%.

References

[1] E. Petritoli, F. Leccese, and L. Ciani, "Reliability and maintenance analysis of unmanned aerial vehicles," Sensors, vol. 18, no. 9, p. 3171, Sep. 2018.

[2] K. J. Lee et al., "A 502-GOPS and 0.984-mW dual-mode intelligent ADAS SoC with real-time semiglobal matching and intention prediction for smart automotive black box system," IEEE J. Solid-State Circuits, vol. 52, no. 1, pp. 139–150, Jan. 2017.

[3] J. P. Cerqueira, T. J. Repetti, Y. Pu, S. Priyadarshi, M. A. Kim, and M. Seok, "Catena: A near-threshold, sub-0.4-mW, 16-core programmable spatial array accelerator for the ultralow-power mobile and embedded Internet of Things," IEEE J. Solid-State Circuits, vol. 55, no. 8, pp. 2270–2284, Aug. 2020.

[4] J. Bund, M. Fugger, C. Lenzen, M. Medina, and W. Rosenbaum, "PALS: Plesiochronous and locally synchronous systems," in Proc. 26th IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC), May 2020, pp. 36–43.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] C. Liu, S. Cho, and Z. Zhang, "A 2.56 mm^2 718 GOPS configurable spiking convolutional sparse coding processor in 40 nm CMOS," in Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC), Nov. 2017, pp. 233–236.

[10] 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. Design, Aug. 2011, pp. 157–162.

[11] L. Ravezzi and H. Partovi, "Clock and synchronization networks for a 3 GHz 64 bit ARMv8 8-core SoC," IEEE J. Solid-State Circuits, vol. 50, no. 7, pp. 1702–1710, Jul. 2015.

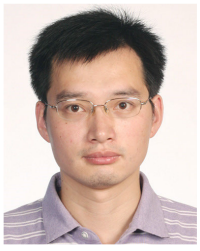
[12] G. Shin, E. Lee, J. Lee, Y. Lee, and Y. Lee, "An ultra-low-power fully-static contention-free flip-flop with complete redundant clock transition and transistor elimination," IEEE J. Solid-State Circuits, vol. 56, no. 10, pp. 3039–3048, Oct. 2021.

[13] W. J. Dally and S. G. Tell, "The even/odd synchronizer: A fast, all-digital, periodic synchronizer," in Proc. IEEE Symp. Asynchronous Circuits Syst. (ASYNC), May 2010, pp. 75–84.

[14] M. Buckler and W. Burleson, "Predictive synchronization for DVFS-enabled multi-processor systems," in Proc. Int. Symp. Quality Electron. Design (ISQED), Mar. 2014, pp. 270–275.
[15] 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.
[16] 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.
[17] I. Loi, F. Angiolini, and L. Benini, "Developing mesochronous synchronizers to enable 3D NoCs," in Proc. Design, Autom. Test Eur., Mar. 2008, pp. 1414–1419.
[18] C. Lin, W. He, Y. Sun, B. Pei, Z. Mao, and M. Seok, "A near-threshold-voltage network-on-chip with a metastability error detection and correction technique for supporting a quad-voltage/frequency-domain ultra-low-power system-on-a-chip," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2020, pp. 394–396.
[19] C. Lin et al., "MEDAC: A metastability condition detection and correction technique for a near-threshold-voltage multi-voltage-/frequency-domain network-on-chip," IEEE J. Solid-State Circuits, vol. 56, no. 7, pp. 2270–2280, Jul. 2021.
[20] Y. Thonnart, "On-chip interconnects: Basic concepts, designs and future opportunities," in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2021, pp. 509–512.
[21] R. Ginosar, "Metastability and synchronizers: A tutorial," IEEE Design Test Comput., vol. 28, no. 5, pp. 23–35, Sep. 2011.
[22] S. Beer, R. Ginosar, R. Dobkin, and Y. Weizman, "MTBF estimation in coherent clock domains," in Proc. IEEE 19th Int. Symp. Asynchronous Circuits Syst., May 2013, pp. 166–173.
[23] S. Paul et al., "A 3.6 GB/s 1.3 mW 400 mV 0.051 mm^2 near-threshold voltage resilient router in 22 nm tri-gate CMOS," in Proc. Symp. VLSI Technol., Jun. 2013, pp. 30–31.
[24] P. Vivet et al., "IntAct: A 96-core processor with six chiplets 3D-stacked on an active interposer with distributed interconnects and integrated power management," IEEE J. Solid-State Circuits, vol. 56, no. 1, pp. 79–97, Jan. 2021.

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 (Senior 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 a Professor with the Department of Micro-Nano Electronics, Shanghai Jiao Tong University, Shanghai, China. His current research interests include low-power circuit design, VLSI architecture for video signal processing, and computer system architecture.

Yanan Sun (Senior 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. Her research area is low-power, high-performance, and variation-tolerant VLSI circuit and system design with emerging nanoscale device technologies.

Lin Shao received the B.S. degree in microelectronic science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2021, where she is currently pursuing the M.S. degree.

Her current research interests include energy-efficient computing, low-power high-performance circuits, and network-on-chip (NoC) architectures.

Bo Zhang received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently pursuing the M.S./Ph.D. degree in electrical engineering with Columbia University, New York, NY, USA.

His research interests include computer architecture for machine learning algorithms and ultra-low-power VLSI circuit and system design.

Jun Yang (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Southeast University, Nanjing, China, in 1999, 2001, and 2004, respectively.

He is currently a Professor with the School of Electronic Science and Engineering, Southeast University. He has coauthored over 50 academic articles and holds 40 patents. His current research interests include near-threshold circuit design and Global Navigation Satellite System (GNSS) algorithm.

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 in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2007 and 2011, respectively.

He was a Technical Staff Member with Texas Instruments Incorporated, Dallas, TX, USA, in 2011. 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, and the 2008 Rackham Pre-Doctoral Fellowship from the University of Michigan. He was a recipient of the 2009 AMD/Custom Integrated Circuits Conference (CICC) Scholarship Award for picowatt voltage reference work, the 2009 DAC/International Solid-State Circuits Conference (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 ISSCC and IEEE CICC. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: REGULAR PAPERS from 2013 to 2015, and 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 of IEEE JOURNAL OF SOLID-STATE CIRCUITS.