A Post-Silicon Hold Time Closure Technique using Data-Path Tunable-Buffers for Variation-Tolerance in Sub-threshold Designs

Divya Akella Kamakshi[1], Xinfei Guo, Harsh N. Patel, Mircea R. Stan, and Benton H. Calhoun
University of Virginia, Charlottesville, USA
[1]Email: dka5ns@virginia.edu

Abstract

This paper presents a post-silicon hold time closure technique for performance-relaxed, sub-threshold digital designs using tunable-buffer insertion in hold-critical data-paths. Hold time closure in flip-flop based digital circuits is highly critical because hold failures cannot be corrected post-fabrication. This criticality increases in the sub-threshold domain, which is highly sensitive to process, voltage, and temperature variations. Design-time hold margins enable robust hold time closure across variations. However, insufficient hold margins can lead to chip failures and overestimated hold margins introduce additional costs in area and power. In this paper, we propose a post-silicon hold time closure methodology that introduces tunable-buffers in the data-path. This enables post-silicon correction of hold violations and therefore, reduces the design effort in estimating design-time hold margins. We design a tunable-buffer, demonstrate the tunable-buffer insertion strategy, and present a physical design flow using standard EDA tools. We verify this technique with measurements of a 130 nm test chip. A design-dependent hold slack improvement in the range of 103%-195% is achieved compared to the traditional buffering technique, with minimal power and area overhead. This technique also has the potential to reduce the number of buffers inserted for hold closure.

Keywords

Digital design, Hold time closure, IoT, Physical design, Post-silicon, Sub-threshold, Tunable-buffers, Ultra-low power, Variation tolerance

1. Introduction

There is a growing market for internet-of-things (IoT) technology that is driven by low-power needs over performance such as wireless sensor nodes (WSNs) and body sensor nodes (BSNs) [1]. Sub-threshold (sub-V_T) operation has gained attention for such ultra-low power, yet performance-relaxed applications. However, the impact of process, voltage, and temperature (PVT) variations is very high at such low supply voltages (V_{DD}) as compared to the super-threshold (super-V_T) region. For instance, Monte Carlo simulations of an X1 inverter (130 nm CMOS bulk technology and 27°C) indicate that its fan out of four (FO4) delay across slow to fast corners varies by ~16X at 0.3 V (sub-V_T) and only by ~2X at 1 V (super-V_T). Figure 1 illustrates this using a plot of the probability density function (PDF) of FO4 delays across different iterations. Similarly, voltage and temperature variations also have a high impact on delay in sub-V_T. Such wide delay distributions make timing closure of digital circuits in sub-V_T challenging.

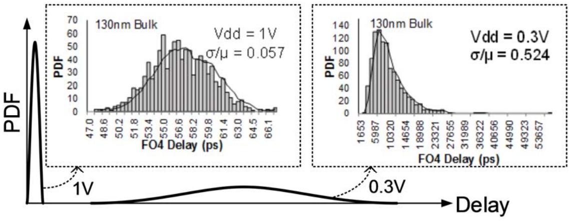

Figure 1: Process variations have a higher impact on the delay in sub-V_T compared to super-V_T.

Especially in flip-flop based digital designs, meeting hold time constraints is highly critical due to the lack of a post-silicon knob to fix failures.

In flip-flop based digital designs, the equation for the flip-flop hold time constraint is:

$$t_{hold} <= t_{clk-q} + t_{logic} - t_{skew}$$

t_{hold} is the flip-flop hold time, t_{clk-q} is the flip-flop clock-to-output delay, t_{logic} is the combinational logic delay, and t_{skew} is the clock skew (clock arrival time at the capture flip-flop, t_{clk2}, minus clock arrival time at the launch flip-flop, t_{clk1}). Hold time violations are caused by insufficient data-path delay (t_{logic}) or excessive positive clock skew (t_{skew}). Traditionally, additional hold margins are allocated during design-time to achieve reliable circuit functionality in the presence of potential PVT variations. In the physical design process, this translates to t_{skew} minimization during the clock-tree synthesis stage and traditional-buffer insertion in hold-critical data paths to increase t_{logic} [2]. This is illustrated in Figure 2. Although traditional-buffer insertion is relatively easy and tool-friendly, it requires a realistic yet worst-case estimation of the design-time hold margins. Even with accurate estimation, the large hold time variability can require insertion of many buffer cells, increasing overhead. This hold margin estimation becomes more challenging with a higher impact of PVT variations in sub-V_T as compared to super-V_T. Underestimation of design-time hold margins can

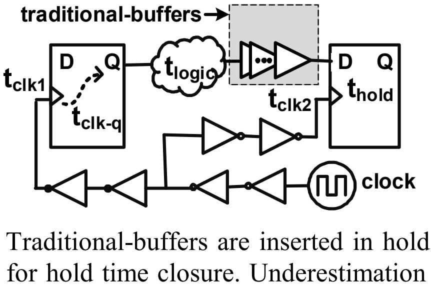

Figure 2: Traditional-buffers are inserted in hold-critical data-paths for hold time closure. Underestimation of hold margins can lead to circuit failure and overestimation leads to power and area overheads.

lead to circuit failure and overestimation leads to power and area overheads. This necessitates a post-silicon knob for hold time closure, especially in sub-V_T, which can mitigate the design effort involved in hold margin estimation.

Traditionally, post-silicon tuning mechanisms such as V_{DD} and frequency tuning to avoid setup time failures and increase energy efficiency [3][5] are available. Recently, the authors in [4] demonstrated post-silicon tunability for hold time closure also, however in latch based designs. Such designs involve two-phase clock generation and distribution. Latch based designs also require additional re-timing and verification efforts compared to flip-flop based designs. Therefore, in this paper, we exclusively focus on variation tolerant flip-flop based designs. For flip-flop based designs, the authors in [6] proposed the use of post-silicon tunable-buffers in the clock path to tweak the clock skew (t_{skew}) and overcome hold violations. However, post-silicon tuning of clock skew in one path may affect the timing in other paths. To overcome this issue, the authors in [6] used dedicated tunable-buffers in the clock paths corresponding to each hold-critical path. This adds to the area and power overhead. Although many CAD algorithms [2] have been proposed to optimize the allocation of traditional-buffer insertion in the hold-critical data-paths and post-silicon tunable-buffers in the clock-paths, design-time hold margin estimation under the impact of variations is still challenging.

In this paper, we propose a post-silicon hold time closure technique that uses tunable-buffers in the data-path instead of traditional-buffers as shown in Figure 3. We demonstrate this technique in the context of performance-relaxed and energy-efficient digital designs. The tunable-buffer is designed to have a wide delay range and therefore, it potentially eliminates the need for long chains of traditional-buffers in the data-path. Compared to traditional-buffering, the tunable-buffer technique mitigates the effort for hold margin estimation by enabling post-silicon hold correction. In Section 2 of this paper, we describe the design of tunable-buffer and the bias voltage generator circuit to control its delay. In Section 3, we describe the methodology for inserting tunable-buffers in hold-critical data-paths and the physical design technique using standard tools. In Section 4, we present the simulation and silicon measurement results that verify the post-silicon hold time closure technique. Section 5 discusses the significance of using the proposed technique in a system-on-chip (SoC) and possible future work. Finally, Section 6 concludes the paper.

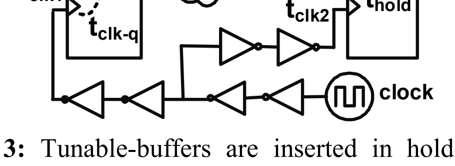

Figure 3: Tunable-buffers are inserted in hold-critical data-paths to enable post-silicon hold correction.

2. Tunable-buffers for hold time closure

2.1. Tunable-buffer structure

Many tunable-buffer structures have been previously explored in literature for different applications. For instance, these structures use tunable output capacitances [6] or configurable switches at the inverter pull-down/pull-up [7], to generate variable delays. However, such designs are not good candidates for our technique. Capacitances may not scale well with technology. Most importantly, capacitances and digital switches have high areas, which makes such tunable-buffers unsuitable for insertion in many hold-critical data-paths. Therefore, we implement tunable-buffers with an analog control as shown in Figure 4(a).

The tunable-buffer implemented in this work has two units, BUF_{bias} and BUF_{drive} as shown in Figure 4(a). BUF_{bias} is current-starved using a PMOS header (biased with voltage V_{bias}), which enables the variable delay. BUF_{drive} operates at V_{DD} to maintain a sharp output transition time. The lowest delay of the tunable-buffer, which is comparable to the delay of two back-to-back traditional-buffers, is obtained with V_{bias} = 0 V. The tunable-buffer functions reliably at different process corners and temperatures (0°C to 100°C) for V_{bias} up to 0.25 V (the highest delay of the tunable-buffer), beyond which the output swing of BUF_{bias} is too low for reliable operation. The substrates of the PMOS devices of all the tunable-buffers in the design are tied to V_{DD}, which makes their insertion into hold-critical paths possible using standard physical design tools. We infer from simulations presented in Section 4.1 that the tunable-buffer power is lower than that of a chain of traditional-buffers of the equivalent delay (for $0 \leq V_{bias} \leq 0.25$ V). This confirms that this tunable-buffer structure is a good candidate for our technique.

2.2. Bias voltage generator

The above tunable-buffer requires a bias voltage (V_{bias}) to vary its delay. Bias generators based on digital-to-analog converters, charge pumps, etc. have been previously explored [8][9] for applications such as fine-grained body-biasing. A bias voltage generator targeted toward energy-efficient systems is required to consume low power. Therefore, we design a voltage-divider based V_{bias} generator as shown in Figure 4(b). A voltage divider is built using a stack of equally-sized diode-connected PMOS transistors (T_m:T_1), with their bulks tied to their sources for similar bias.

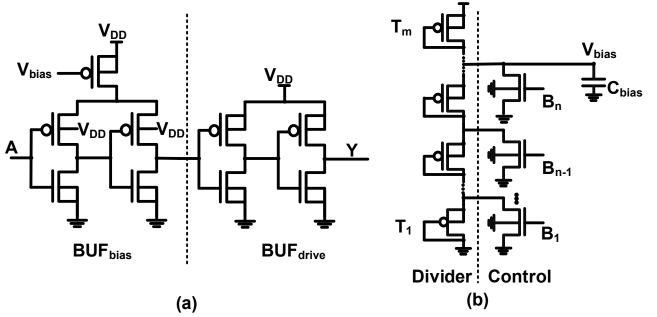

Figure 4: (a) Tunable-buffer structure. (b) Bias voltage generator to control tunable-buffer delay.

V_{bias} is tapped at the center of the stack.

Such a high impedance voltage bias node can be highly susceptible to switching noise from the tunable-buffers, coupling from nearby signals etc. On-chip capacitor, C_{bias}, mitigates such noise. T_m:T_1 operate at sub-V_T, which enables an ultra-low static power. The target bias current is a few nAs to lower switching noise at V_{bias} and effect of variations. In this paper, m = 20 and V_{DD} = 0.5 V. Therefore, when T_{20}:T_1 are all active in the divider circuit, V_{bias} = 0.25 V. A control logic of NMOS switches with a thermometer code-input B_n:B_1 (n=10 in this paper) ties different nodes of the divider to ground (V_{SS}). This is to vary the active number of transistors and generate different V_{bias} as shown in Table I.

The bias generator is custom designed and is not a part of the physical design flow. The design decisions for the bias voltage generator: m, n, and the nodes of the divider at which NMOS switches are placed, are governed by the required range and granularity of the tunable-buffer delay. The tunable-buffer range is its maximum delay and granularity is its different delay steps. Whereas in traditional synthesis flow, additional hold margin leads to a longer buffer chain, it translates to a wider tunable-buffer range in the proposed technique. In low-performance SoCs, the tunable-buffer delay range is more critical than its granularity. In this paper, we allow the maximum tunable-buffer range (V_{bias} = 0.25 V). This range is equivalent to the delay of 20 traditional-buffers at a typical corner (TT:27^{o}C). This high additional margin enables failure-free operation at extreme PVT conditions. At V_{bias} > 0.25 V, the tunable-buffer starts approaching its limit of reliable operation. A granularity of 10 (n=10) is chosen as a sufficient coarse-grained tunability for different PVT conditions. The tunable-buffer is non-linear owing to its current-starved nature and non-linear V_{bias} generation, but this is not critical in performance-relaxed designs.

3. Design implementation

3.1. Tunable-buffer and bias voltage generator

Figure 5(a) shows the tunable-buffer layout in a 130 nm bulk CMOS technology. The two units, BUF_{bias} and BUF_{drive}, along with the PMOS header is shown in Figure 5(a). The tunable-buffer area is ~1.27X higher than two back-to-back traditional-buffers due to the PMOS header. However, a single tunable-buffer can mitigate the need for a chain of two or more traditional-buffers in hold-critical data-paths, as discussed in Section 4.2. This enables potential savings in the number of buffers. Metals M1-M3 are typically used for routing between standard-cells within a block. Therefore, we use the next horizontal metal, M5, for routing V_{bias} over V_{DD} as shown in Figure 5(a). Although this may potentially cause top-level routing blockages in M5, we infer in Section 4.3 that the overheads associated with this are minimal.

Table I: Thermometer code vs. V_{bias}

$B_{10:1}$	V_{bias} mV	$B_{10:1}$	V_{bias} mV	$B_{10:1}$	V_{bias} mV
1111111111	0	1111110000	143	1100000000	222
1111111110	45	1111100000	167	1000000000	237
1111111100	83	1111000000	188	0000000000	250
1111111000	115	1110000000	206		

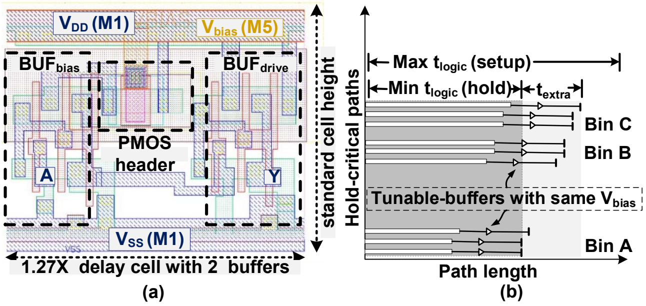

Figure 5: (a) Tunable-buffer layout: PMOS header in addition to traditional back-to-back buffers, M5 is used for V_{bias}. (b) Illustration of tunable-buffer insertion.

The bias voltage generator area is 0.0022 mm^2 (~1% of a 130 nm openMSP430 core). Our strategy uses a single V_{bias} to tune many tunable-buffers as discussed in the next section. We use a metal-insulator-metal (MIM) capacitor for C_{bias} (10 pF of area 0.002 mm^2), which can be placed over device area to avoid overhead. 10 pF is sufficient to mitigate the noise from 1k switching tunable-buffers at 100 kHz.

3.2. Tunable-buffer insertion methodology

In performance-relaxed IoT SoCs, we propose to insert tunable-buffers in the hold-critical paths and tune them with the same V_{bias} despite different data-path delays (t_{logic}). Figure 5(b) illustrates an example scenario. The tunable-buffer delay is tuned with a certain V_{bias} to meet hold time in the most-critical paths (e.g. Bin A). The same delay (due to same V_{bias}) also resolves the hold issues in the less critical paths (e.g. Bin C), which gain extra hold slack (t_{extra}). Despite t_{extra}, the data-path delay remains within the t_{logic} limit to meet setup time. For instance, in an openMSP430 core designed to operate at 32 kHz and 0.5 V V_{DD}, the setup slack is ~45% of the cycle time (14 µs). Therefore, t_{extra} of tens of ns (in 130 nm) will not cause failure. The static timing analysis (STA) is presented in Section 4.2.

3.3. Physical design flow for implementation

In this section, we present a physical design flow for tunable-buffer insertion using conventional EDA tools as shown in Figure 6. First, a conventional synthesis flow is executed starting from RTL design. The proposed physical design flow is performed after the conventional synthesis. The steps involved are:

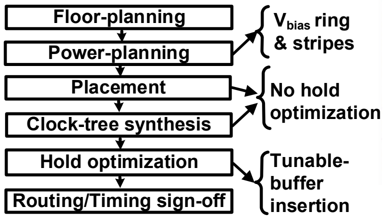

Figure 6: A top-level flow-chart of the proposed physical design flow for tunable-buffers insertion.

(a) Floor-planning: This step involves the deciding the block or design dimensions.

(b) Power-planning: Traditionally, this step involves creating V_{DD} and V_{SS} rings (using metal layers M3 and M4). Metal M1 stripes are also added over standard-cell rows to pre-connect their V_{DD} and V_{SS} to the above rings. In our physical design flow, a V_{bias} ring using metal M3 and M4 is also created additionally, and metal M5 stripes are added to pre-connect the V_{bias} port of the tunable-buffers to the above ring. In this technique, M5 is reserved for V_{bias} (Figure 5(a)) and this indeed may create M5 routing blockage within the block. However, this may be optimized in the future by placing M5 stripes only over tunable-buffer rows.

(c) Placement: Similar to the traditional flow, this step involves placement of different standard-cells and optimization of their locations, sizes, and design rule violations (DRV). Hold time is not optimized in this step.

(d) Clock-tree synthesis: Similar to the traditional flow, this step involves balancing a clock-tree and clock-skew by inserting clock buffers or inverters.

(e) Hold optimization: An important requirement at this stage is the timing information (.lib) of the tunable-buffer corresponding to different V_{bias} values. Therefore, we generate timing the .lib using a characterization tool. A preliminary hold slack target is given as an input to the physical design tool to identify the hold-critical paths. We restrict the hold-buffers to only the tunable-buffer.

Depending on the desired amount of post-silicon tunability, the tunable-buffer .lib corresponding to different V_{bias} can be used in the flow. For maximum tunability, the .lib generated at $V_{bias} = 0$ can be used. However, it can have no impact on the number of the traditional-buffers (no buffer savings) and potentially cause some area and power overhead. In this paper, we use the tunable-buffer .lib generated at $V_{bias} = 0.125$ V in the flow, which enables a trade-off of post-silicon tunability and buffer numbers for minimal power/area impact. We quantify the metrics in Section 4.2. The advantage of this technique is that in case of underestimation in the preliminary target hold margin, the tunable-buffers can be tuned post-silicon for hold correction. Figure 7 shows the layout of an openMSP430 design with tunable-buffers in the data-path.

(f) Routing and timing sign-off: After routing, STA using the tunable-buffer .lib at different V_{bias} can be done as shown in Section 4.2. Finally, the bias generator is custom-placed and routed to the V_{bias} ring.

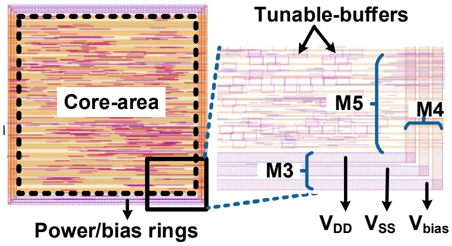

Figure 7: Synthesized openMSP430 design with tunable-buffers. The bias voltage generator is then custom placed adjacent to and routed to the design.

4. Results

4.1. Power Simulations

Figure 8 shows the simulated power of the tunable-buffer for different V_{bias}, compared to a traditional-buffer chain of same delay in both 28 nm FDSOI and 130 nm bulk CMOS nodes, at sub-V_T (0.4 V, 100 kHz). The power of the tunable-buffer is lower compared to the chain of traditional-buffers of the same delay when $0 \leq V_{bias} \leq 0.25$ V. For instance, in a 130 nm technology and at $V_{bias} = 0.167$ V, the tunable-buffer power is 2.61X lower than a chain of traditional buffers of same delay. This is because current-starved BUF_{bias} unit consumes lower power than a full-V_{DD} swing buffer. This holds true despite the short circuit current occurring at the interface of the low-swing BUF_{bias} and full-V_{DD} swing BUF_{drive}. This allows the use of our tunable-buffer without power overheads. As discussed in section 2.1, for $V_{bias} \geq 0.25$, the output swing of BUF_{bias} is too low for reliable operation. Also, beyond this point, short circuit current can cause the tunable-buffer power to shoot up in power and it will no longer be a viable option.

4.2. Timing Simulations

An STA tool is used for timing simulations. Figure 9 shows the hold slack histograms in a fast-fast corner (FF:25ºC) with aggressive on-chip variation (OCV), for a 32-bit/16-stage shift register (SR) designed using the proposed technique, across different V_{bias}. Negative hold slack means timing failure. The number of paths failing hold time decrease as V_{bias} is increased from 0 V to 0.2 V (Figure

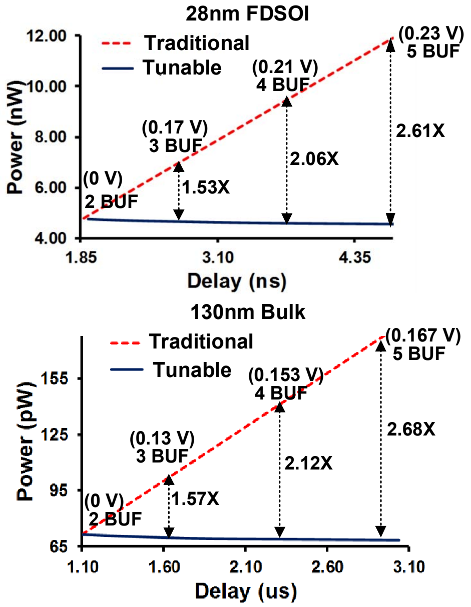

Figure 8: Tunable-buffer is lower power (simulated in 28 nm FDSOI and 130 nm bulk CMOS) compared to the traditional-buffer chain of same delay. In the plot, "X" BUF means both tunable-buffer (V_{bias} in parentheses) and traditional-buffer chain are of "X" buffer delay.

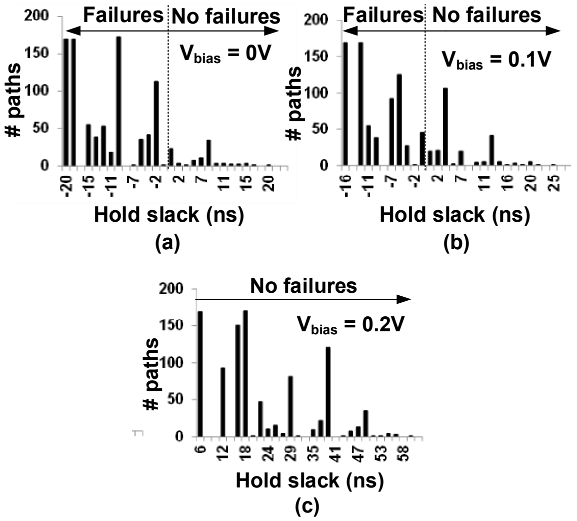

Figure 9: The number of paths vs. hold slack at FF:25°C from STA simulation: Negative slack (failures) at V_{bias} = 0 is made positive by increasing V_{bias}.

9(a)-(c)). This demonstrates hold correction with V_{bias} tuning at a worst-case PVT condition. V_{bias} = 0.25 V is the reliable limit for our tunable-buffer.

Table II summarizes the hold-related benefits of 3 blocks (32-bit/16-stage SR, 8-bit/4-tap FIR, 2-channel direct memory access (DMA)) designed with our technique to operate at 0.5 V and 32 kHz. Post-silicon tunability enables hold-slack improvement, which is maximum at V_{bias} = 0.25 V. We compare the maximum hold-slack improvement for these blocks (at FF:25°C with aggressive OCV) with their traditionally-designed counterparts. We designed the baseline traditional blocks for ~25 ns of hold-slack at this corner. The slack improvement depends on the hold-critical path distribution. A highly hold-critical SR sees a 195% increase in hold slack and a setup-critical FIR sees up to a 103% increase. An added benefit in blocks with many hold-critical paths is the buffer savings (29% for SR and 18% in DMA). We observe only a small buffer overhead for blocks like FIR for a 103% increase in hold slack.

Finally, we note that all the above blocks meet setup time also at V_{bias} = 0.25 V. The worst-case setup slack for the blocks with setup-critical paths (FIR and DMA) at a slow-slow corner with OCV is greater than ~45% of the clock cycle with a much higher setup-slack for hold-critical SR (~90%), which is ample for failure-free operation at 32 kHz.

4.3. Measurement results

Table II: Benefits in blocks with data-path tunable-buffers

Design	Max Hold-Slack Increase (% w.r.t traditional)	Hold-Buffers (% w.r.t traditional)
SR	195%	-29% (savings)
FIR	103%	3% (increase)
DMA	152%	-18% (savings)

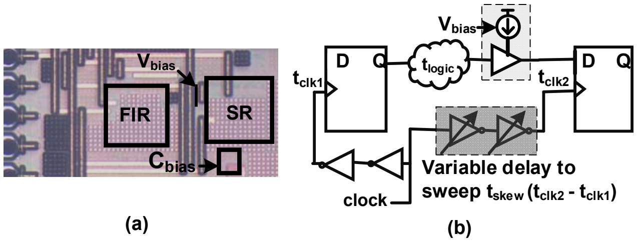

Figure 10: (a) Annotated photo of a 130 nm bulk CMOS test chip (b) Experimental setup: a variable delay in clock-path is used to increase t_{skew} to emulate hold failures.

Figure 10(a) shows an annotated photo of the test chip fabricated in a 130 nm bulk CMOS technology, to verify our post-silicon hold closure technique. We implemented the above discussed SR and FIR blocks with tunable-buffers in hold-critical data-paths and also a V_{bias} generator to control their delay. An on-chip experimental setup as shown in Figure 10(b) is necessary to verify post-silicon hold correction. This is because the occurrence of hold failures are unpredictable due to variations and also due to the availability of our test chips in the typical corner (TT). To overcome this testing challenge, we implemented a variable delay line in the clock-paths corresponding to a few hold-critical data-paths. With this setup, we are able to sweep this clock-path delay to generate different values of t_{skew} and also measure it for a few hold-critical paths. Therefore, we induce the occurrence of hold failures. Finally, we are able to correct these hold failures by increasing V_{bias} of the tunable-buffers by tuning bits B_{10}:B_1 of the V_{bias} generator. We verify post-silicon hold correction in both the SR and FIR.

Figure 11 demonstrates post-silicon hold correction in the SR across 5 chips. It is a plot between the measured t_{skew} of a hold-critical path (using the experimental setup) vs. the corresponding V_{bias} set to correct the subsequent hold failure. Up to t_{skew} of ~100 ns, there are no hold failures in the hold-critical path and therefore, V_{bias} is tuned to 0 V. After that, as we increase t_{skew}, hold failures start occurring. We set a higher V_{bias} to solve the failures. We demonstrated hold correction for ~800 ns of t_{skew} (32X our STA sign-off hold slack of 25 ns at FF:25°C). This verifies that there is high potential for hold-slack improvement using this technique.

Next, we examine the overheads associated with this technique. Figure 12 shows the measured average power of the SR and FIR across 5 chips for different V_{bias}. The power

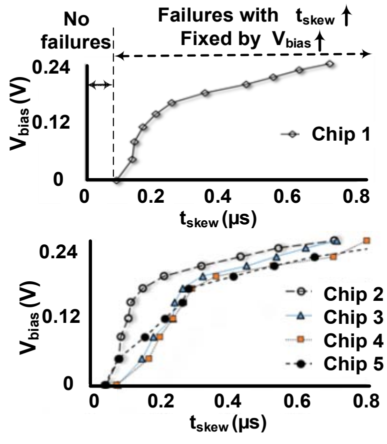

Figure 11: Post-silicon hold correction: hold failure due to t_{skew} corrected by V_{bias} tuning.

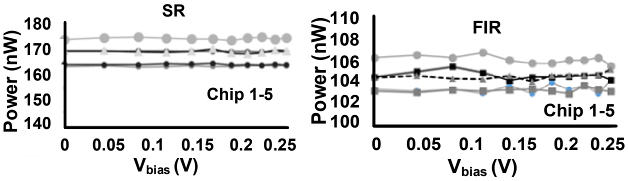

Figure 12: Negligible variation in power with V_{bias} tuning.

across different V_{bias} values stays within ~1% variation, which can be attributed to the inherent variability caused by the measurement process. We infer that the impact of the short circuit currents in the tunable-buffers is negligible. The V_{bias} generator, which can be shared across multiple blocks, has an average measured power of only 4 nW at 0.5 V. We expect the area overhead from V_{bias} ring at block-level and routing at the chip-level to be similar to the fine-grained body-biasing techniques [10], which is about 2%. Owing to the ultra-low power and small area of the bias generator, it has the potential to be used as local bias generators to reduce V_{bias} routing overhead. As shown in Table II, the area and power overhead/savings from the tunable-buffers are design-dependent. For example, we show only a small increase in a FIR, which may be compensated by hold-critical block savings. We conclude that post-silicon hold correction can be achieved with minimal area and power impact.

5. Post-silicon hold time closure in a system-on-chip

In this section, we discuss the significance of the post-silicon hold closure technique in an SoC and possible future work. Timing closure at the block and chip-level at multiple corners is highly challenging and time-consuming in large SoCs or multi-core systems where different blocks have different thermal behavior, supply noise profile, and widespread process mismatch. In the previous sections, we demonstrated the feasibility of a post-silicon hold closure technique and showed that the costs associated with implementing the tuning structure (consisting of a bias voltage generator and its control logic) is low. These low overhead circuits have the potential to be implemented locally for different blocks across the chip. Compared to having only one tuning structure in the whole chip, fine-grained tuning at the block level will increase the tolerance of the chip to spatially varying conditions. These tuning structures can be controlled differently corresponding to the timing behaviors of each block. For instance, a tuning control unit can act as a hub for the multiple tuning structures. Furthermore, automation of post-silicon hold closure using error detection hardware may also be explored in the future. It is to be noted there is an upper limit on the size of the block that can be tuned by one tuning structure. Using the bias voltage generator design described in Section 2.2, V_{bias} can drive at least 1k simultaneously switching tunable-buffers with a decoupling capacitor C_{bias} of 10 pF. The bias voltage generator controls an arrays of tunable-buffers, which add to the load capacitance and switching noise at V_{bias}. A tuning structure can therefore be shared among multiple small blocks and or be dedicated to a big block. Therefore, in a large-scale SoC, post-silicon tunable-buffer insertion will enable opportunities for fine-grained localized tuning for hold closure. Furthermore, local tuning structures will enable avoid V_{bias} routing congestion [10].

6. Conclusions

This paper presents a post-silicon hold time closure technique for sub-V_T designs that are highly sensitive to PVT variations. This technique makes use of post-silicon tunable-buffers in hold-critical data-paths. This enables post-silicon correction of hold violations, mitigates the efforts for design-time hold margin estimation, and has the potential to reduce the number of inserted hold-buffers. In this paper, we discuss the optimal tunable-buffer and the bias voltage generator circuit designs for this technique. We present a physical design flow for tunable-buffer insertion using commercial tools. We present simulation and measurement results to verify this scheme. We show that a design-dependent hold slack improvement in the range of 103%-195% is possible compared to the traditional technique.

7. Acknowledgements

This work was supported and funded in part by NSF NERC ASSIST Center (EEC-1160483), by DARPA under the PERFECT program, by CFAR, one of the StarNET centers, and by NSF.

8. References

[1] A. Klinefelter et al., “A 6.45μW self-powered IoT SoC with integrated energy-harvesting power management and ULP asymmetric radios,” in IEEE ISSCC, pp. 1–3, Feb 2015.

[2] P. Wu et al., “On timing closure: Buffer insertion for hold-violation removal,” in 51st ACM/EDAC/IEEE DAC, pp. 1–6, June 2014.

[3] M. S. Golanbari et al., “Post-fabrication calibration of near-threshold circuits for energy efficiency,” in IEEE ISQED, pp. 385–390, Mar 2017.

[4] Y. Zhang et al., “Hold time closure for subthreshold circuits using a two phase, latch based timing method,” in IEEE S3S, pp. 1–2, Oct 2013.

[5] S. Rusu and S. Tam, “Clock generation and distribution for the first ia-64 microprocessor,” in IEEE ISSCC, pp. 176–177, Feb 2000.

[6] G. Geannopoulos and X. Dai, “An adaptive digital deskewing circuit for clock distribution networks,” in IEEE ISSCC, pp. 400–401, Feb 1998.

[7] M. Maymandi-Nejad and M. Sachdev, “A monotonic digitally controlled delay element,” IEEE JSSCC, vol. 40, pp. 2212–2219, Nov 2005.

[8] N. Kamae et al., “A body bias generator with wide supply-range down to threshold voltage for within-die variability compensation,” in IEEE A-SSCC, pp. 53–56, Nov 2014.

[9] M. Meijer et al., “A forward body bias generator for digital cmos circuits with supply voltage scaling,” in IEEE ISCAS, pp. 2482–2485, May 2010.

[10] S. Narendra et al., “1.1V 1GHz Communications Router with On-chip body bias in 150nm CMOS,” in IEEE ISSCC, pp. 218–482, Feb 2002.