Hold Time Closure for Subthreshold Circuits Using a Two-Phase, Latch Based Timing Method

Yanqing Zhang and Benton H. Calhoun
University of Virginia
Email: yz4hz@virginia.edu, bcalhoun@virginia.edu

***Abstract*-This paper presents an ultra low power (ULP) solution to hold time closure for subthreshold circuits across PVT variation and mismatch using a two-phase, latch based timing method. We show that compared to conventional hold buffering, our solution saves up to 37% (at 6σ yield) in energy per operation and allows for post tapeout hold time correction. Replacing registers with latches also permits time borrowing, which we show can save up to 47% (6σ yield) when used for setup time closure.**

I. INTRODUCTION

Supply voltage scaling into the subthreshold (sub-V_T) region is becoming an increasingly attractive solution to save energy and power in cases where performance is not the driving factor, for example in body sensor nodes (BSNs)[1][2]. However, the increased impact of PVT and mismatch variation on gate delay presents a design challenge for timing closure at sub-V_T V_{DD}s. Our Monte Carlo (MC) simulations on a size X1 inverter from a commercial 65nm technology show FO4 delay changing 12X from SS:27ºC corner to FF:27ºC corner at V_{DD}=0.3V, which is ~130mV below V_T (Fig 1.). Delay distributions due to mismatch display log-normal distributions, which have greater σ/μ values than the Gaussian distributions that result from mismatch in superthreshold. Conventional synthesis methodologies are not well designed to cope with this extreme spread in delay, often resulting in greatly increased logic area to meet setup time and excessive hold buffer insertion to fix hold time.

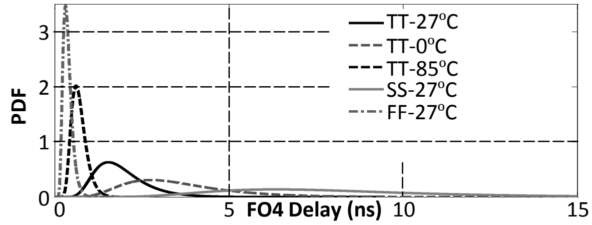

Fig 1. Distribution of FO4 delay across process corners.

Though there is work proposing fixes to setup time failure [3], little work has been done to study hold time closure in sub-V_T. This paper will show through analysis and simulation how hold failures occur and how conventional buffer insertion will lead to great energy overhead. Thus, we propose an alternative two-phase, latch based timing method solution that eliminates hold buffers entirely for robust, low power hold time closure.

II. HOLD FAILURE IN SUB-V_T

To demonstrate how hold failures occur in sub-V_T, we performed 3000 point MC simulations on a size X2 hold time friendly standard cell register (Fig. 2) placed in a shift register (SR) path setting to find the distributions of t_{cq}, t_{hold}, and t_{skew} (Fig. 2). Ideal clock and data slew were used. To find what types of skew exist in digital, synthesized blocks, we studied the clock trees of all digital components of the ULP BSN SoC in [1] and found a deterministic skew between different register sinks equal to 0.5FO4 delay. Fig. 2 shows the additional stochastic skew on top of the deterministic amount of skew. Using these results and equations for log-normal distributions, we calculated the hold margin (eq. 1) at 3σ yield (99.7%) assuming case (b) of Fig. 2 and found the margin to be negative.

$$\textit{Hold margin}=t_{cq}-t_{hold}-t_{skew}(\textit{deterministic})-t_{skew}(\textit{stochastic}) =15.23-2.34-12-7.9 = -7.01\ \textit{ns (TT:27°C corner).} \qquad (1)$$

This means that even a hold friendly register presumably protected by its t_{cq} delay will fail hold time without buffering. It should be noted that two factors, clock jitter and slew, were not included in this dicussion. While it is obvious jitter will make the negative margin worse, slew also contributes negatively through stochastic means [4].

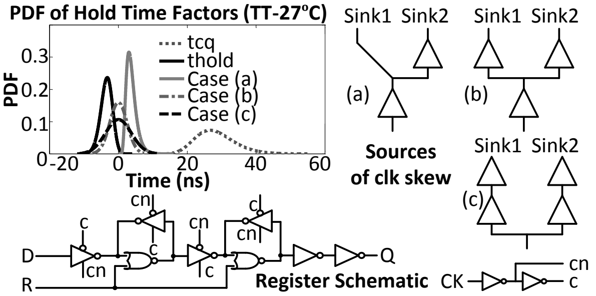

Fig. 2. MC sim results of hold friendly register. Stochastic skew arises from unbalanced clock tree by 1 level (a), or balanced tree but clock paths differ by one level (b) or two (c).

The severity of hold time in sub-V_T is apparent when we took away the hold buffers in the BSN SoC design and used a commercial timing tool to check hold time, with proper timing derates set to reflect the 3σ yield for the SoC's technology. The results are shown in Fig. 3. Fig. 3 shows that on average a significant portion (12%) of energy is attributed to hold buffers to properly meet hold time. Delving deeper, we found through 3000 point MC simulations of extracted paths (both register and clock path) from the SoCs that for the majority of cases (b) and (c) from Fig. 2, either 3 or 4 hold buffers must be inserted per path respectively to meet 3σ yield (Table 1), which leads to the significant energy overhead.

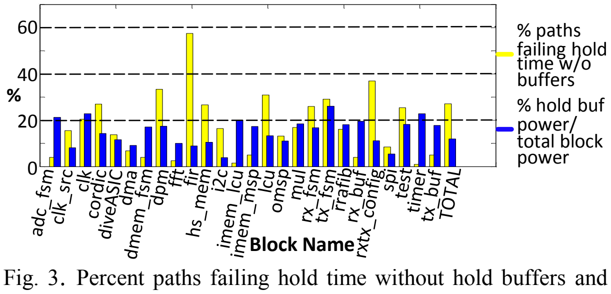

Fig. 3. Percent paths failing hold time without hold buffers and percent hold buffer contribution to total block power, displaying the signficant overhead of conventional hold buffer insertion.

Table 1. # of Buffers Needed to Meet Hold Time

	Yield-no buffer	#Buffs Insert	Yield-w/ buffer
Case(b)	89.7%	3	100%
Case(c)	81.3%	4	100%

In short, hold buffer insertion must be done to meet hold time, which will lead to significant energy increases.

III. TWO-PHASE LATCH BASED TIMING

Eq. 1 reveals that clock skew is a deciding factor in hold time closure. Hold time failure due to skew is diagrammed in Fig. 4. Our latch based method uses non-complementary two-phase clocks for alternating pipe stages whose phases inheritantly deny the possibility of having simultaneous transparency time, eliminating the potential for hold time errors. The proposed method is detailed in Fig. 4. While latch based timing is a known approach, this is to our knowledge the first application of the method to fixing hold time errors in subthreshold.

The implementation of this method has the advantage of being readily integratable with commercial synthesis tools. A register based design is first implemented. The amount of phase offset (the 'hold shoulder') between the two phases can be tuned based on results of timing closure and anticipated jitter, and this can occur after fabrication if necessary. A custom script translates all registers into latches, and re-timing is done so that the number of total pipeline stages in the design is doubled, while the logic between pipe stages is roughly split in half. The doubling of pipe stages retains the original throughput of the design, while register to latch translation reduces the storage element energy. The creation of the two phases assumes the availability of a DLL, which are abundant in an SoC environment. In summary, a well defined 'hold shoulder' provides resiliency against variation, and the hold shoulder can be tuned based on how much variation is present (how many sigma yield to guard against, how much jitter anticipated, etc.)

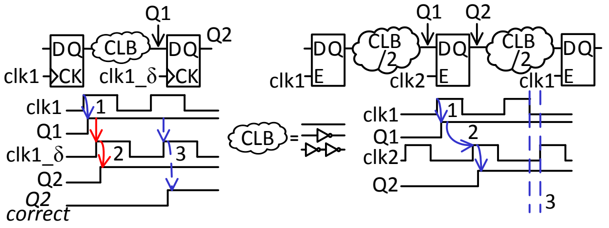

Fig. 4. On left, diagram of register based hold failure. 1: Data launched to Q1. 2: Skew causes Q1 to meet setup time→hold failure. 3: Desired operation. On right, diagram of two-phase method. 1: Data launched to Q1. 2: Non-complementary phases negate skew even with variation and desired correct operation is ensured. 3: The 'hold shoulder'.

IV. RESULTS AND CONCLUSIONS

The two-phase method was implemented on a 16b, 4-stage MAC, on a 32b, 8-stage shift register (SR), and an 8b, 4-tap, 8-stage FIR. These blocks were chosen to represent the setup time critical/hold time non-critical case (MAC), the hold critical/setup time non-critical case (SR), and a mixed logic path length 'common case' (FIR). Several iterations of the designs' register base case and two-phase implementation were done at the 3-, 4-, 5-, and 6σ yield constraint with a V_{DD} of 0.3V. Hold time was checked across all process and temperature corners (SS:0ºC, TT:27ºC, FF:45ºC), and setup time was checked at SS:0ºC. We also imposed 1FO4 inverter delay of clock jitter (50ns). These implementations resulted in hold shoulders of 80, 100, 120, 140ns and operation frequencies of 667k, 555k, 460k, 385kHz for 3-, 4-, 5-, and 6σ yield, respectively. None of the two-phase implementations required any hold buffers. A comparison of energy per operation and its breakdown is given in Fig. 5.

As the yield constraint is made more stringent (from 3 to 6σ), the savings from two-phase latch based design increase gradually. For the SR, whose savings mainly come from hold buffer insertion negation, additional hold buffer energy increases from 3 to 6σ (Fig. 5(b)). The reason is twofold: the main culprit of skew is case (b) of Fig. 2, whose distribution is Gaussian, and the early derated delay of hold buffers does not change much from 3 to 6σ due to the nature of log-normal distributions. The SR is able to net energy savings of 27-37% compared to the register base case.

By allowing time borrowing, the MAC is also able to save 45-47% energy. With time borrowing, setup critical circuits are able to use originally unusable positive slack in register designs to meet setup time instead of upsizing and rebuffering. The amount saved is constant across yield constraints, because the clock period, delay of a logic stage, and amount of slack scale evenly from one yield constraint to another (Fig. 5(c)). The FIR is an interesting case, as it is able to harness both savings from time borrowing for setup optimization, as well as hold buffer negation from the two-phase method. However, these savings are partially compromised due to the latch growth factor (# of latches after two-phase translation/# of registers in base case), which is 2.3 for the FIR, resulting in increased energy in the clock network and latches in the FIR. With all factors accounted, the FIR saves 14-27% in energy per operation across yield constraints.

In conclusion, our two-phase, latch based method provides an ULP solution to subthreshold timing closure in the face of PVT variation and mismatch compared to the conventional register based designs that must resort to upsizing, rebuffering, and hold buffer insertion to meet timing. Our extensive simulations show that our method can save up to 37% in energy per operation for hold critical circuits and 47% for setup critical circuits at 6σ yield.

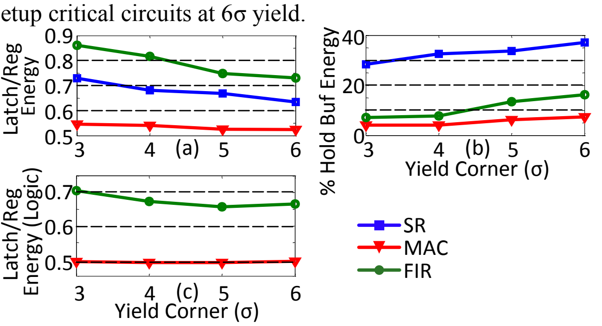

Fig. 5. Implementation results. (a) Relative energy consumption (latch/register based design). (b) % Hold buffer energy/total block energy. (c) Relative energy for logic components (latch/register).

ACKNOWLEDGEMENTS: This project was supported in part by NVIDIA through the DARPA PERFECT program and by the NSF NERC ASSIST Center (EEC-1160483).

References:

[1] Y. Zhang, et al., "A Batteryless 19 uW MICS/ISM-Band Energy Harvesting Body Sensor Node SoC for ExG Applications", *JSSC*, vol. 48, issue 1, pp. 199-213, 01/2013.

[2] G. Chen, et al., "A Millimeter-Scale Nearly-Perpetual Sensor System with Stacked Battery and Solar Cells," *ISSCC*, Feb. 2010.

[3] M. Wieckowski, et al., "Timing Yield Enhancement Through Soft Edge Flip-Flop Based Design,"*CICC*, Sept.2008.

[4] M. H. Abu-Rahma and M. Anis, "A Statistical Design-Oriented Delay Variation Model Accounting for Within-Die Variations," *TCAD*, Nov. 2008.