Completion Detection Based Timing Error Detection and Correction in a Near-Threshold RISC-V Microprocessor in FDSOI 28nm

Roel Uytterhoeven, *Student Member, IEEE* and Wim Dehaene, *Senior Member, IEEE*

***Abstract*—This paper presents a novel timing error detection and correction (EDaC) technique to reduce design margins in a near/sub-threshold RISCV-IM32 microprocessor. The proposed technique takes a snapshot of the datapath's activity just before the launch of the next clock to determine if a timing error will occur. If so, it prevents the error at the last moment by gating the clock with one cycle. This avoids imposing additional hold constraints on the design and removes the need for a complex correction mechanism. The design is implemented in FDSOI 28nm and achieves a margined minimum energy point (MEP) of $1.32\,\mathrm{pJ/cycle}$ at $3\,\mathrm{MHz}$ and $0.434\,\mathrm{V}$. The EDaC technique robustly eliminates all voltage margin, resulting in an improved MEP of $0.92\,\mathrm{pJ/cycle}$ and $0.345\,\mathrm{V}$ at the same frequency.**

***Index Terms*—adaptive voltage scaling, CMOS digital integrated circuits, energy-efficient digital circuits, error detection and correction (EDaC), near/sub-threshold digital circuits, razor, timing error detection, variation resilience**

I. Introduction

Several timing error detection and correction (EDaC) techniques have been demonstrated to reclaim design margins in digital systems [1]. These are especially valuable in ultra-low energy near/sub-threshold designs where the low supply voltage increases the circuit's sensitivity to PVT variations. This in turn demands even larger design margins that oppose the low voltage energy savings [2]. Typically, these techniques rely on a double-sampling (DS) approach where a shadow flop/latch or a clocked transition detector resamples data after a predefined timing window to detect late arriving signals. However, this is prone to false errors that are triggered when a fast signal arrives within the detection window at a monitored endpoint through a short path. To avoid these, all techniques that use a DS approach have to impose a hold constraint that matches the width of their error detection window on all monitored paths. This constraint results in a direct trade-off between area/energy overhead due to hold padding and the size of detection window [3]. This forces designers to limit this window in order to keep the hold padding losses reasonable, often without regard for the effect on the EDaC's reliability.

To mitigate the hold constraints, some techniques have proposed a fully latch based pipeline [4]. Yet, this comes at the cost of a more complex timing closure, the risk of race conditions, and a higher clock load. Further, in [5], sparse insertion of the error detection elements has been explored to reduce the number of monitored endpoints. However, this compromises the detection robustness as with fewer detectors the circuit's activity is more likely to mask critical slow paths which in turn requires a more conservative voltage/frequency scaling as discussed by [6].

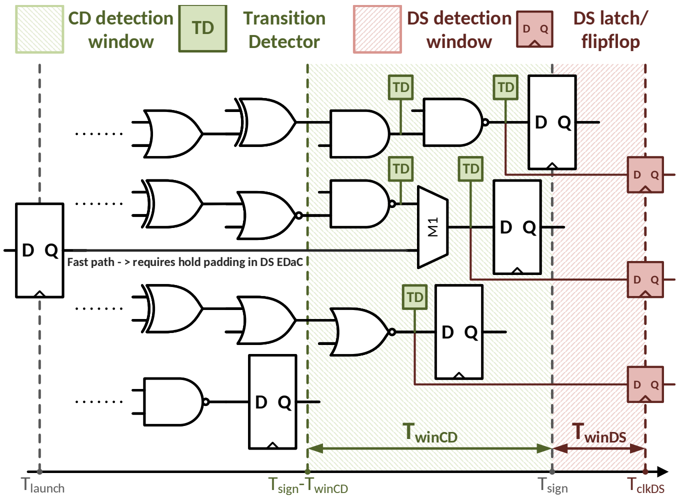

Fig. 1. CD monitoring principle, TDs flag activity within T_{winCD}. Compared with a DS EDaC, no hold padding is required and a large window is thus easier to achieve.

This paper proposes a novel EDaC technique that is inherently robust to false errors and thus avoids the overhead associated with extra hold constraints. Furthermore, the technique achieves a high detection coverage which reduces the impact of activity on the reliability of the error detection. Finally, this work provides a strategy to determine the appropriate size for the error detection window based on statistical timing analysis.

II. Completion detection EDaC system

The presented completion detection (CD) EDaC system is illustrated by Fig. 1. Using transition detectors (TD), the system monitors activity from critical gates in the datapath for late signal toggles that would cause a timing error. To determine the set of critical cells that should be monitored, all cells are placed on a time line based on their worst-case output timing. Adding the desired error detection window to this time line shows which cells should be part of the critical set. In Fig. 1 this matches the cells in the green area covered by T_{winCD}. Note that, although the proposed timing window has the same meaning as a conventional EDaC timing window (i.e. the time range within which late arrivals can be detected), its

The authors thank ST-Microelectronics for chip fabrication, and FWO for the supporting SB-fellowship (1S31817N).

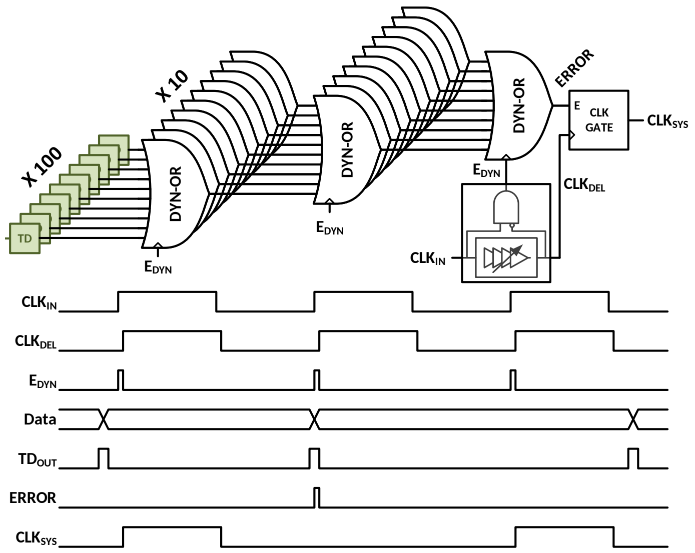

Fig. 2. Three stage dynamic OR-tree triggered by E_{DYN} and configured to evaluate 1000 TDs. Last stage generates $ERROR$ signal which triggers clock gating at the root of the system's clock. Timing-diagram illustrates a correction event under critical activity.

construction is fundamentally different. It relies on the propagation delay along paths in the datapath to create the timing window. This window is then observed instantaneously at the end of each clock cycle, as explained in the next paragraph. This is in contrast with the double sampling approach using a delayed clock to create the detection window T_{winDS}.

When a TD flags activity in the set of critical cells during the instantaneous observation at the end of a clock cycle, a timing error is likely to happen on the rising edge of the clock. To evaluate this condition, a dynamic OR-tree takes a snapshot of all TD outputs just before this rising edge and reduces these outputs to a single error signal as shown in Fig. 2. The dynamic implementation of this OR-tree serves a dual purpose. First, it enables a fast, wide fan-in OR-gate construction that scales well with the amount of TDs required in the design. Second, it enables a quick and brief sampling of the TDs using the E_{DYN} pulse. A programmable delay allows to match the width of this pulse with the OR-tree's propagation delay in order to keep the evaluation time as short as possible.

If the OR-tree evaluation flags a timing error, a clock gate at the root of the clock tree halts the launch of the system's clock (CLK_{SYS}) for one cycle as shown in Fig. 2. This provides an extra clock cycle for late signals to settle and prevents the timing error from being captured by the processor's sequential elements. Thanks to this last-minute error prevention, no additional correction mechanism is required.

Additionally, the presented error detection strategy is capable of detecting critical activity independent of its propagation to an endpoint, i.e. it monitors all critical activity within the error detection window. This helps to prevent activity dependent over-tuning of voltage or frequency when the activity pattern masks critical paths from the monitored endpoints. For instance, imagine that under the current activity pattern only the endpoint connected to MUX M1 in Fig. 1 is activated. Now, the setting of this MUX determines whether an endpoint monitoring EDaC strategy still sees a valid critical path or whether it actually monitors a non-critical path. In the latter case, the voltage or frequency tuning loop receives overly optimistic timing information which allows to operate far beyond the critical point of first failure (PoFF). If under this condition, the activity changes and a critical path triggers one of the endpoints again, the resulting timing error could fall beyond the reach of the error detection window leading to a system failure. Since the proposed EDaC system also monitors the critical activity in front of the M1 MUX, it still provides realistic timing information in this situation and hence it avoids the over-tuning pitfall.

Compared to a double sampling (DS) approach, CD offers three major advantages. First, as it works with a snapshot taken before the launch of the clock, it does not infer any extra hold constraints on monitored cells/paths. This eliminates the trade-off between the width of the error detection window and the area/energy overhead associated with hold padding. Second, the scope of the detection is broader since the system does not depend on the propagation of critical activity towards an endpoint. This makes the EDaC system less dependent on the processor's activity and thus improves its overall robustness. Third, since errors are detected just before the launch of the clock, correction is possible by simply stopping the clock, avoiding the need for a complex correction mechanism.

III. Implementation

The proposed EDaC system is applied to a near-threshold implementation of a RISC-V IM32 microprocessor in FDSOI 28nm. This processor is similar to [7] and its processing capabilities match those of an ARM Cortex-M0. The implementation uses normal std-cells recharacterized at $0.3\,\mathrm{V}$, $0.4\,\mathrm{V}$, and $0.5\,\mathrm{V}$ to facilitate the near/sub-threshold design.

A custom std-cell implements the TD by comparing its input signal with an internally delayed version using an XOR-operation. The TD cell exhibits a latency of less than $0.5\,\mathrm{ns}$ at $0.4\,\mathrm{V}$, has an area footprint of $4.2\,\mathrm{\mu m}^2$ and a leakage power consumption of around $1\,\mathrm{nW}$. Besides, the DYN-OR gates are custom std-cells as well. The OR-operation is implemented with 10 parallel NMOS transistors as pull-down network and a keeper to ensure stability of the internal dynamic node. These OR-gates exhibit less than $0.5\,\mathrm{ns}$ propagation delay with a footprint of $3.75\,\mathrm{\mu m}^2$ and a leakage power of $1.9\,\mathrm{nW}$. The complete error detection has thus a total detection delay (i.e. inherent margin) below $2\,\mathrm{ns}$ at $0.4\,\mathrm{V}$. Further, a NAND-gate ladder network implements the E_{DYN} pulse's programmable delay with a thermometer coded tuning range from $0.25\,\mathrm{ns}$ to $4\,\mathrm{ns}$ in steps of $0.25\,\mathrm{ns}$.

After a low-power synthesis- and P&R-flow, the appropriate size for the error detection window is sought using statistical static timing analysis. This analysis provides both the mean and sigma for the delay of each path under local variations. By sampling random Gaussian distributions based on these values, a Python script emulates full system MC-simulations. These then yield probability-density-functions by binning the slowest timing of each sample, as well as the second slowest, the third slowest, and so on. In Fig. 3 the PDFs of the 20 slowest

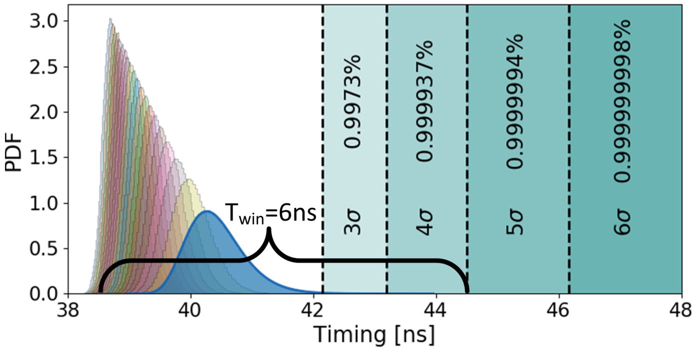

Fig. 3. The 20 most critical PDFs at 50 ns signoff clock period and TT corner. More overlap between PDFs indicates less recovered margin per resolved timing error. Chosen 6 ns window equals 12 % of the clock period.

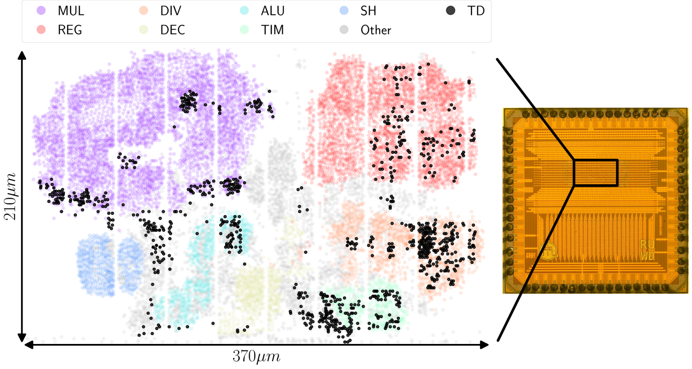

Fig. 4. Die photograph with close-up from the core area and the physical distribution of TDs amongst different functional blocks.

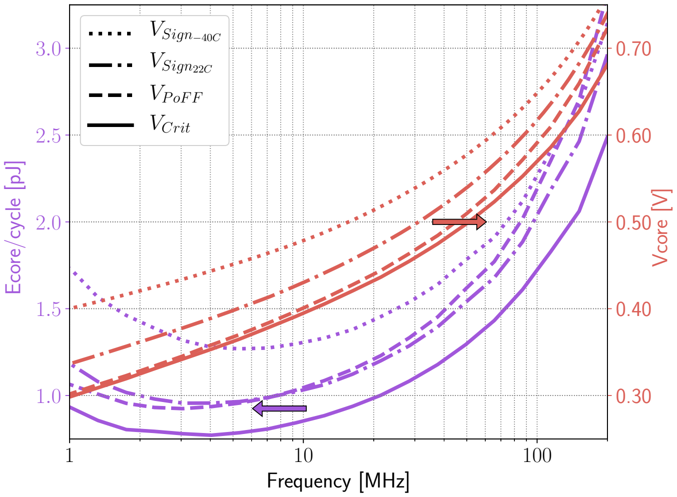

Fig. 5. Core energy and voltage results over a 200x frequency range. Signoff voltages are based on spice results from 200 critical paths.

bins are shown together with the yield that can be expected in respect to the clock period. This allows to reason on the size of the detection window. This should start at the desired yield (4 σ in our case) and end when the PDFs start to fully overlap. At this end point, they create a ‘critical wall’ beyond which there is no added benefit in solving (i.e. removing a PDF from the figure) extra timing errors, as the detection will immediately find the next timing error.

Based on this strategy, an error detection window of 6 ns is chosen as indicated in Fig. 3. This window corresponds to the T_{winCD} parameter in Fig. 1 and encloses 950 critical gates demanding a 3-level OR-tree constructed with 10 to 1 OR-gate stages as shown in Fig. 2. As this results in 1000 TD evaluation slots, the window is expanded slightly to add 50 extra gates. In total, 8 % of the logic cells in the design are monitored. Combined with the OR-tree, this leads to a total EDaC area overhead of 6 % and a leakage overhead of 15 % for the chosen 4 σ yield. The paths that propagate through monitored cells can activate 40 % of the endpoints in the design. The larger relative endpoint coverage results from the convergence of the datapath towards the processor’s register-file.

The TDs are added to the design by placing them as close as possible to their designated critical gate to ensure that only a marginal extra load is added. After this insertion, timing reports show only minor differences from the initial timing and indicate that the cell selection remains valid. Next, a K-means clustering algorithm divides the TDs in groups of 10 based on their placement. This provides the location and connectivity information for the first stage elements of the OR-tree. The same clustering procedure is repeated to connect and place the second stage elements as well. Fig. 4 shows the TD insertions in the core area and the die photograph. From this figure, it becomes clear that TDs have been added to several functional blocks of the processor indicating a good coverage under different activity patterns.

IV. Measurements

All 23 samples are measured and fully operational under four voltage conditions: margined signoff voltage at −40 °C and SS process corner ($V_{Sign_{-40C}}$), margined signoff voltage at 22 °C and SS process corner ($V_{Sign_{22C}}$), Point of First Failure (PoFF) at 22 °C using EDaC tuning (V_{PoFF}) and critical voltage without EDaC or margin at 22 °C (V_{Crit}). Correct operation is verified using a Dhrystone C-program benchmark. Further, the minimum save E_{dyn} pulse width is determined for each sample by shrinking its width until the EDaC fails.

Fig. 5 shows the core’s energy consumption and voltage scaling under these conditions over frequency. The processor has a critical minimum energy point (MEP) of 0.78 pJ/cycle at 3 MHz and 0.345 V supply. In the MEP, the most conservative signoff condition results in a 26 % voltage margin of 89 mV that increases the energy consumption to 1.32 pJ/cycle. The EDaC system safely eliminates this voltage margin and saves 0.4 pJ/cycle in energy consumption. This results in a MEP of 0.92 pJ/cycle at PoFF. In this MEP, only 0.14 pJ/cycle of energy is lost with respect to the unmargined critical MEP. Towards higher clock speeds and higher supply voltages, the overhead of the EDaC system increases due to the dynamic energy cost of the E_{DYN} pulse. Combined with the reduced design margins at higher supplies, this limits the energy gains of the EDaC system to operation speeds below 100 MHz in the near/sub-threshold region.

Besides, the EDaC system operates successfully beyond the

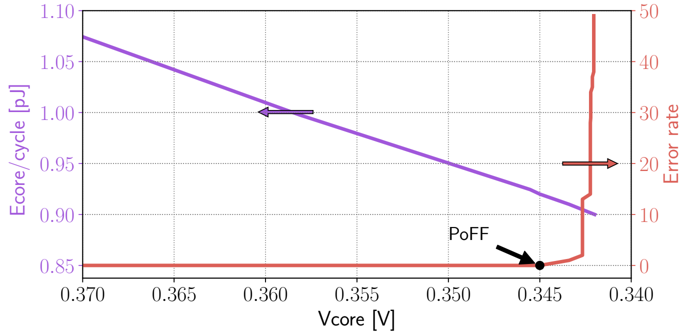

Fig. 6. Voltage scaling beyond the Point of First Failure (PoFF). Error rate indicates the number of detected and corrected timing errors per 10000 cycles.

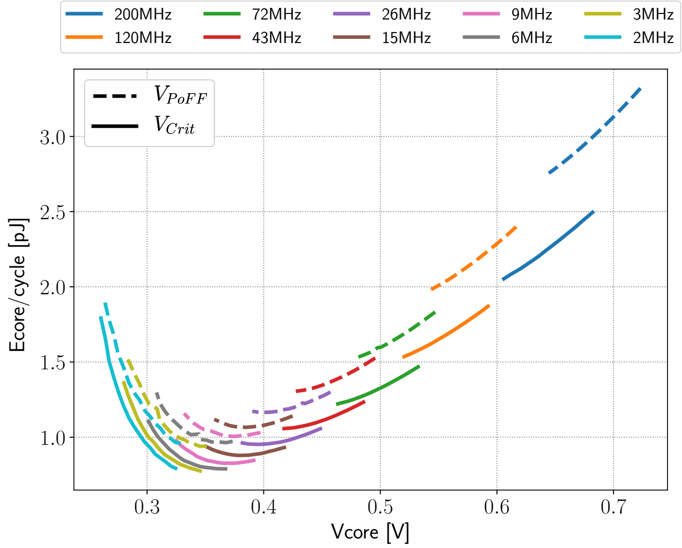

Fig. 7. PoFF and Critical energy along iso-frequency lines created by adapting the body bias to maintain speed whilst scaling Vcore. Vcore and Vbody relate quasi linear with 85mVcore/Vbody. At higher frequencies, dynamic savings result from reduced Vcore and increased body biasing. At lower frequencies, leakage savings result from increased Vcore and reduced body biasing.

PoFF as shown in Fig. 6. Yet, the error rate increases rapidly beyond this point and only a small extra supply reduction is achieved. This is partially caused by the delay's high sensitivity to voltage scaling in the near/sub-threshold regime, but also by the wide detection coverage of the EDaC system. This detects errors even when they are masked from endpoints which allows for a robust voltage tuning, but has the downside that operation beyond PoFF quickly degrades performance.

In Fig. 7 the body bias capabilities of the FDSOI technology are explored. These enable an active trade-off between dynamic and leakage energy creating iso-frequency lines that each have their own MEP, i.e. the most efficient combination of supply voltage and body biasing for that frequency. This allows to operate between 2 to $10\,\mathrm{MHz}$ at the best possible energy efficiency under both critical and PoFF voltage tuning.

Finally, table I compares this EDaC system with other near/sub-threshold capable EDaC implementations. It shows that our system enables a relatively high detection window and coverage with a small area penalty. Further, the presented microprocessor achieves the best energy efficiency for this class of microprocessors equipped with an EDaC system.

TABLE I
COMPARISON OF NEAR/SUB-THRESHOLD CAPABLE EDAC SYSTEMS

	[5] JSSC'15	[2] JSSC'18	[8] JSSC'19	This work
Host processor	R-proc (16bit)	Cortex-M0	Cortex-M0	RISC-V IM32
Technology	65nm CMOS	40nm CMOS	28nm CMOS	28nm FDSOI
Detection technique	DS (2-phase latch pipeline)	DS	DS	CD
Hold Solution	Not required	Buffers	Buffers	Not required
T_{win}	50% T_{clk}	5% T_{clk}	50% T_{clk}	12% T_{clk}
Endpoint coverage	13%	5.7%	19.5%	40%
Area overhead	8.3%	7%	4.17%* 50%**	6%
E_{crit}/cycle	-	8.11pJ	-	0.78pJ
E_{PoFF}/cycle	3.25pJ	11.12pJ	3.99pJ	0.92pJ

* Including 64kB SRAM ** Estimate without SRAM using die photograph

V. CONCLUSION

When operating at PoFF, the presented EDaC system is able to recover all voltage margin in the processor's MEP and regains $0.4\,\mathrm{pJ/cycle}$ of the $0.54\,\mathrm{pJ/cycle}$ ($74\,\%$) energy overhead inferred by conventional voltage margins. It does so without imposing any extra hold constraints during the design's implementation which helps to minimize the area impact. Combined with the simple error correction strategy, this allows to apply the technique easily to other designs without the need for architectural modifications. Furthermore, the wide detection scope guarantees robust error detection over a wide supply and frequency range and aids to prevent overtuning under varying activity patterns.

REFERENCES

[1] S. Das, C. Tokunaga, S. Pant, W. H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RazorII: In situ error detection and correction for PVT and ser tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, 2009.

[2] H. Reyserhove and W. Dehaene, "Margin Elimination Through Timing Error Detection in a Near-Threshold Enabled 32-bit Microcontroller in 40-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 7, pp. 2101–2113, 2018.

[3] X. Shang, W. Shan, J. Xu, M. Lu, Y. Xiang, L. Shi, and J. Yang, "A 0.46V-1.1V Transition-Detector with In-Situ Timing-Error Detection and Correction Based on Pulsed-Latch Design in AES Accelerator," in *2018 IEEE Asian Solid-State Circuits Conference, A-SSCC 2018 - Proceedings*, vol. 1, no. 8041, 2018, pp. 145–148.

[4] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. M. Harris, D. Blaauw, and D. Sylvester, "Bubble razor: Eliminating timing margins in an ARM cortex-M3 Processor in 45 nm CMOS using architecturally independent error detection and correction," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 66–81, 2013.

[5] S. Kim and M. Seok, "Variation-Tolerant, Ultra-Low-Voltage Microprocessor With a Low-Overhead, Within-a-Cycle In-Situ Timing-Error Detection and Correction Technique," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 6, pp. 1478–1490, 2015.

[6] T. Gemmeke, M. Konijnenburg, and C. Bachmann, "In-situ Performance Monitor Employing Threshold Based Notifications (TheBaN)," in *ESSCIRC 2013 - IEEE 39th European Solid State Circuits Conference*. IEEE, 2013, pp. 271–274.

[7] R. Uytterhoeven and W. Dehaene, "A sub 10 pJ/Cycle over a 2 to 200 MHz Performance Range RISC-V Microprocessor in 28 nm FDSOI," in *ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference*. IEEE, 2018, pp. 326–329.

[8] C.-Y. Hong and T.-T. Lui, "A Variation-Resilient Microprocessor with a Two-Level Timing Error Detection and Correction System in 28nm CMOS," *IEEE Journal of Solid-State Circuits*, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8906049