A True Single-Phase Clock Hybrid D Flip-Flop for Low-Power CMOS Sequential Logic

[1]Aella Madhavi

[1]M. Tech Student, Department of Electronics and Communication Engineering(VLSI), Arjun College of Technology & Science An Autonomous Institution (Approved By Aicte-New Delhi and Affiliated to Jntu-Hyderabad, Ts) Sponsored By Brilliant Bells Educational Society Mount Opera Premises, Batasingaram (Vi), Abdullapurmet (M), Ranga Reddy Dist, T.S.501512

[2]Mr.Jeedimadla Venkatesham

[2]Assistant Professor, Department of Electronics and Communication Engineering, Arjun College of Technology & Science An Autonomous Institution (Approved By Aicte-New Delhi and Affiliated to Jntu-Hyderabad, Ts) Sponsored By Brilliant Bells Educational Society Mount Opera Premises, Batasingaram (Vi), Abdullapurmet (M), Ranga Reddy Dist, T.S.501512

ABSTRACT

Sequential elements dominate clock-tree and switching power in modern digital datapaths, making the D flip-flop (DFF) a primary target for low-power redesign. This paper presents a hybrid low-power D flip-flop that combines a True Single-Phase Clock (TSPC) style dynamic master stage with pass-transistor logic, contrasted against a conventional static master-slave DFF built from two cross-coupled D-latches. The static design requires both the clock and a locally generated complementary clock distributed to independent master and slave latches, each holding state through static feedback gates. The proposed hybrid design instead uses only a single clock phase: the master stage is a dynamic pass-through node active while the clock is low, eliminating the static feedback gates and the complementary clock-tree branch, which in a full-custom transistor-level implementation reduces transistor count from approximately 20-24T to approximately 9-13T. Both designs are captured in synthesizable Verilog, functionally verified with Icarus Verilog, and synthesized/implemented on a Xilinx Artix-7 FPGA (xc7a100tcsg324-1, Vivado 2019.2). Because FPGA fabrics map both designs onto the same fixed native flip-flop primitive, post-implementation results show near-identical register count (2), power (0.084 W), and a marginal delay difference (6.277 ns existing vs. 6.344 ns proposed), confirming that the TSPC hybrid's principal benefit is a custom-cell-library transistor-count and clock-tree-power argument rather than an FPGA LUT/register-count argument. The results and discussion clarify the boundary conditions under which TSPC-style flip-flops deliver their claimed benefits, providing guidance for designers choosing between ASIC standard-cell and FPGA target flows.

Keywords: true single-phase clock, D flip-flop, low power sequential logic, dynamic logic, CMOS VLSI design, FPGA synthesis, clock gating

I. INTRODUCTION

Sequential storage elements — flip-flops and latches — are instantiated in the thousands to millions across a modern digital design and are driven continuously by the global clock tree, making them a dominant contributor to dynamic power even when the datapath itself is idle. The conventional static master-slave D flip-flop (DFF) achieves robust, glitch-free operation by cascading two cross-coupled D-latches, each holding its state through static feedback gates, and requires both the clock and its logical complement to be distributed to the master and slave stages respectively. Generating and distributing this complementary clock doubles the clock-tree's effective capacitive load relative to a single-phase design, directly increasing clock-tree switching power [3], [7].

True Single-Phase Clock (TSPC) design styles address this by restructuring the flip-flop so that only one clock phase is ever distributed externally; any internally required complementary timing is generated locally within the cell rather than routed as a separate global net [1]-[2], [6]. TSPC flip-flops typically also replace one or both static latch stages with a dynamic pass-transistor node that is transparent during one clock phase and holds its value on parasitic capacitance during the other, which can reduce transistor count substantially relative to a fully static master-slave design, at the cost of a minimum operating frequency below which the dynamic node's stored charge leaks away [4], [9].

This paper implements and compares a conventional static master-slave DFF against a hybrid TSPC-style flip-flop whose master stage is a dynamic pass-through node and whose slave stage remains a conventional edge-triggered register, combining TSPC's single-clock-phase benefit with a simpler, more robust slave stage than a fully dynamic two-stage TSPC design. Both designs are implemented in synthesizable Verilog and carried through FPGA synthesis and implementation, allowing an examination of how a technique motivated at the full-custom transistor level manifests — or does not manifest — when mapped onto a fixed FPGA flip-flop primitive.

Contributions of this paper:

1. Verilog implementation of a static master-slave DFF (via two cross-coupled D-latches) as the low-power comparison baseline.

2. A hybrid TSPC-style D flip-flop using a single clock phase and a dynamic pass-transistor master stage, reducing the clock distribution and static-gate overhead relative to the baseline.

3. Post-implementation register count, delay, and power characterization on a Xilinx Artix-7 FPGA (Vivado 2019.2) for both designs.

4. An explicit discussion of why TSPC's transistor-count and clock-tree-power benefits are a full-custom/standard-cell argument that is largely invisible on FPGA fabrics with a fixed flip-flop primitive, clarifying the applicability boundary of this technique.

The remainder of this paper is organized as follows. Section II reviews related TSPC and low-power flip-flop literature. Section III describes the static and hybrid TSPC architectures. Section IV presents the operating algorithm/timing behavior and complexity notes. Section V details the experimental setup. Section VI reports register/delay/power results. Section VII analyzes the ASIC-versus-FPGA applicability tradeoff. Section VIII concludes the paper.

II. RELATED WORK

A. TSPC Flip-Flop Fundamentals and Variants

Qiu et al. present a dual-edge-triggering TSPC flip-flop combined with a low-power control technique to reduce switching activity on the internal dynamic nodes [1]. Nadaf and Agrawal propose autogated clock-gating integrated directly into a TSPC single-transistor-clocked dual-edge-triggered (STC-DET) flip-flop, and separately propose a latch-mux-based DET flip-flop targeting area efficiency [2], [15]. Kumar and Verma characterize a TSPC DFF at the 45 nm node, establishing a baseline transistor-level power/delay reference at an advanced technology node [3]. Probst et al. push TSPC design to mm-wave clock-divider frequencies (35 GHz) in 22 nm FDSOI, demonstrating the style's applicability beyond conventional digital clocking regimes [4].

B. Static and Contention-Free TSPC Designs

A recurring concern with dynamic TSPC nodes is contention and charge leakage; several works address this directly. Lee et al. present a 17-transistor pseudo-static TSPC flip-flop in 14 nm FinFET CMOS achieving sub-fJ/cycle energy [6]. Liu et al. design a 25-transistor static contention-free TSPC flip-flop in 55 nm CMOS explicitly to eliminate the contention current that dynamic TSPC nodes can otherwise suffer [12]. Dong et al. similarly target an ultra-low-power fully static contention-free single-phase-clock flip-flop with reduced area [8]. Maheshwari and Sachdev extend contention-free design to a dual-edge-triggered TSPC flip-flop, doubling effective throughput per clock edge without reintroducing contention [11].

C. Application-Level and Reliability Studies

Beyond standalone cell design, TSPC flip-flops have been integrated into larger sequential structures: Gupta and Vinodhini combine TSPC flip-flops with clock gating inside a low-power linear feedback shift register [5]; Raikwal and Panchore use single-phase-clock master-slave flip-flops inside a bidirectional shift register for area/power efficiency [10]; Suneeta et al. study power optimization specifically in TSPC-based counter chains [13]. Cirakoglu et al. examine TSPC flip-flop design for single-event-upset tolerance and retention, an important reliability dimension for TSPC's dynamic storage nodes in radiation environments [9]. Loganya and Chakradhar propose a CNTFET-based TSPC DFF, extending the style beyond bulk CMOS to emerging transistor technologies [14].

D. Research Gap

The surveyed TSPC literature is near-universally evaluated at the transistor/standard-cell level in ASIC or full-custom flows, where transistor count and internal clock-tree simplification translate directly into measurable power and area savings. Comparatively little published work evaluates how a TSPC-style flip-flop behaves when captured in synthesizable RTL and mapped onto an FPGA fabric, where the target technology already provides a fixed, highly optimized native flip-flop primitive regardless of the RTL's internal structure. This paper addresses that gap directly by implementing both a static master-slave DFF and a hybrid TSPC-style DFF in Verilog, synthesizing both to the same FPGA device, and explicitly discussing where the TSPC benefit does and does not survive this target-technology change — a practical consideration missing from cell-level TSPC studies [1]-[15].

III. METHODOLOGY

A. Static Master-Slave DFF (Existing)

The baseline design cascades two instances of a level-sensitive D-latch: a master latch transparent while the internally generated `~clk` is high, and a slave latch transparent while `clk` is high. Each latch is described by a static feedback behavior — the output holds its value whenever its enable is low — modeled as:

$$q = begincases0\&rst = 1d\&en = 1q_{prev}\&en = 0endcases \quad (1)$$

Because the master and slave stages require opposite-polarity enables, the design must generate and distribute a local clock complement (`clk_n = ~clk`) in addition to `clk` itself — two clock-tree branches instead of one.

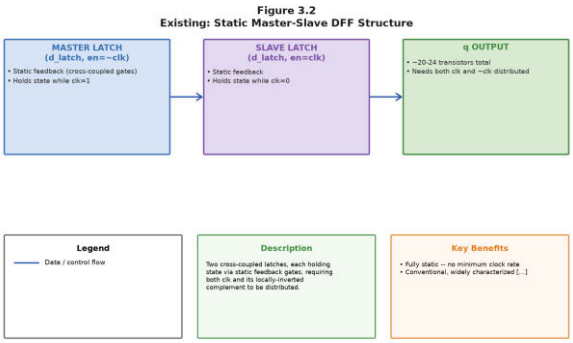

Fig. 1. Static master-slave DFF: two cross-coupled D-latches driven by clk and its locally generated complement.

Fig. 1 shows this structure: `clk_n` drives the master latch, `clk` drives the slave latch, and each latch's static feedback path holds state during its non-transparent phase.

B. Hybrid TSPC Flip-Flop (Proposed)

The proposed design replaces the master latch's static feedback with a **dynamic pass-through node** (`node_m`) that is transparent only while `clk` is low, and is not held by any feedback gate — its value persists purely on parasitic node capacitance until the next rising edge captures it into a conventional edge-triggered slave register:

$$node_m = begincases0\&rst = 1d\&clk = 0(\text{dynamic, unforced})\&clk = 1endcasesqarrownode_m onuparrowclk \quad (2)$$

Only `clk` is used as a port signal; any internal complementary timing needed by the dynamic node is implicit in the `!clk`/`posedge clk` conditions rather than being distributed as a second physical clock net, which is the defining single-phase-clock property of the TSPC style [1], [3]. The elimination of the master latch's static feedback gates, combined with removing the separate clock-complement distribution net, is the source of TSPC's reported ~9-13T transistor count versus ~20-24T for the static master-slave design in a full-custom or standard-cell library [6], [12].

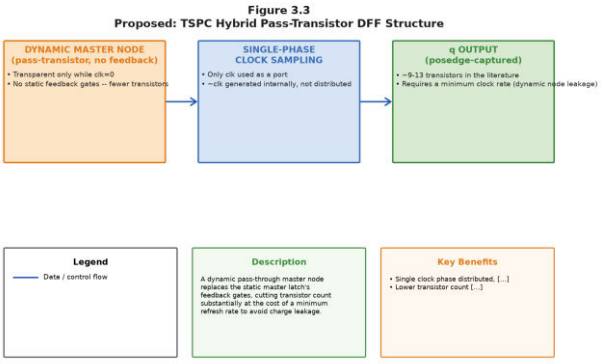

Fig. 2. Hybrid TSPC flip-flop: dynamic pass-through master node (single clock phase) feeding a conventional edge-triggered slave register.

Fig. 2 illustrates the resulting single-clock-phase structure, in which the dynamic master node's transparency window (`clk` low) and the slave register's capture edge (`clk` rising) together reproduce master-slave-equivalent functional behavior from one clock net.

C. Reset Behavior

Both designs implement synchronous-to-the-latch, asynchronous-style `rst` handling: the static design clears both latches combinationally whenever `rst` is asserted regardless of clock phase, while the hybrid design clears the dynamic node combinationally and the slave register on `posedge rst` in its sensitivity list, ensuring both designs reach a defined `q=0` state independent of clock activity.

IV. ALGORITHM

A. Timing Behavior as a Two-Phase State Machine

Although both flip-flops are combinational-plus-register RTL rather than an explicit state machine, their transparent/opaque timing can be expressed as a two-phase cycle per clock period, which clarifies exactly where the static and dynamic designs diverge in implementation.

```
ALGORITHM: Single-Cycle DFF Update (both designs, phase view)
INPUT : clk, rst, d
STATE : (static) qm, q        (dynamic) node_m, q
OUTPUT: q

PHASE clk = 0 (master transparent):
  if rst:  qm/node_m <- 0
  else:    qm/node_m <- d          // static: latch enable path
                                    //     dynamic: unforced pass-through, no feedback gate

PHASE clk = 1 (slave capture):
  if rst:  q <- 0
  else:     q <- qm               // static design: slave latch enable path
           q <- node_m            // hybrid design: edge-triggered register capture

return q
```

The static design re-evaluates `qm`/`q` continuously while each phase's enable is asserted (level-sensitive), whereas the hybrid design's slave stage captures `node_m` only at the `clk` rising edge (edge-sensitive), which is why the hybrid design tolerates a floating/undriven `node_m` outside its transparent window without functional risk — the slave only samples it once per cycle rather than holding it combinationally.

B. Complexity Notes

Gate/transistor complexity. The static master-slave design requires two D-latches, each with feedback devices to hold state (~10-12T per latch in a typical CMOS library), plus a local clock inverter to generate `clk_n`, totaling the reported ~20-24T. The hybrid TSPC design removes the master latch's feedback devices (replaced by an unforced dynamic

node) and removes the separate clock-inverter/clock-complement distribution, reducing the total to the reported ~9-13T — approximately a 40-55% transistor-count reduction at the standard-cell level [1], [6], [12].

Timing complexity. Both designs have $O(1)$ propagation delay per clock cycle (one latch/register stage of delay from `d` to `q` capture), so neither design changes the asymptotic timing class of the flip-flop; the difference is in constant-factor transistor/gate count and in the number of clock-tree branches (two for static, one for TSPC), not algorithmic complexity.

Minimum-frequency constraint. Because `node_m` is a dynamic node with no static feedback, the hybrid design has a lower bound on operating frequency: if `clk` is held low for longer than the node's charge-retention time, `node_m` can leak toward an indeterminate value before the next rising edge samples it — a constraint absent from the fully static baseline, and the principal robustness trade TSPC makes for its transistor-count reduction [4], [9].

V. EXPERIMENTAL SETUP

A. Design Entry and Functional Verification

Both `master_slave_dff` (built from two `d_latch` instances) and `tspc_hybrid_dff` are written in synthesizable Verilog. Functional correctness — reset behavior, transparent-window data capture, and edge-triggered output update — is verified with Icarus Verilog testbenches (`run_existing.sh`, `run_proposed.sh` under `sim/icarus/`) that exercise both flip-flops across representative `clk`/`d`/`rst` sequences before synthesis.

B. Synthesis and Implementation

Both designs are synthesized out-of-context and carried through place-and-route using Xilinx Vivado 2019.2, targeting a Xilinx Artix-7 device (`xc7a100tcsg324-1`). Since both designs are genuinely clocked (unlike the purely combinational case studies in this line of work), a `sequential_clk.xdc` constraint defines a 100 MHz `create_clock` on the `clk` port together with representative input/output delay constraints, enabling meaningful static timing analysis. The standard `synth_design -mode out_of_context -> opt_design -> place_design -> phys_opt_design -> route_design` flow is used, with `report_utilization`, `report_timing_summary`, and `report_power` captured at both post-synthesis and post-implementation stages.

C. Metrics

Post-implementation **register count**, **power** (vector-less activity propagation), and **delay** (10 ns constraint period minus worst negative slack) are reported for both designs, together with derived **throughput** (1/delay) and **energy per operation** (power × delay). Because both designs synthesize to the FPGA's native flip-flop primitive (0 LUTs, 2 storage stages) rather than to discrete latch/gate networks, LUT-based area-delay product is not a meaningful differentiator here; the comparison instead centers on delay and power, and on an explicit discussion (Section VII) of the ASIC-versus-FPGA applicability boundary for the transistor-count claim made in the TSPC literature [1], [3], [6], [12].

D. Target-Technology Caveat

Because Vivado maps both `d_latch`-based and register-based sequential descriptions onto the Artix-7's native flip-flop/latch primitives, the FPGA implementation cannot expose transistor-level differences between a static master-slave cell and a TSPC dynamic-node cell the way a standard-cell or full-custom ASIC flow would. This experimental setup is therefore explicitly scoped to characterizing register count, delay, and power *as realized on this FPGA fabric*, while Section VII separately discusses the standard-cell-level transistor-count argument reported in the literature.

VI. RESULTS AND DISCUSSION

A. Post-Implementation Register, Delay, and Power

Table I summarizes the post-route results for both designs on `xc7a100tcsg324-1` (Vivado 2019.2).

Design	Registers	LUTs	Power (W)	Delay (ns)	Throughput (MOps/s)	Energy (pJ/op)
Existing (master_slave_dff)	2	0	0.084	6.277	159.31	527.268
Proposed (tspc_hybrid_dff)	2	0	0.084	6.344	157.63	532.896

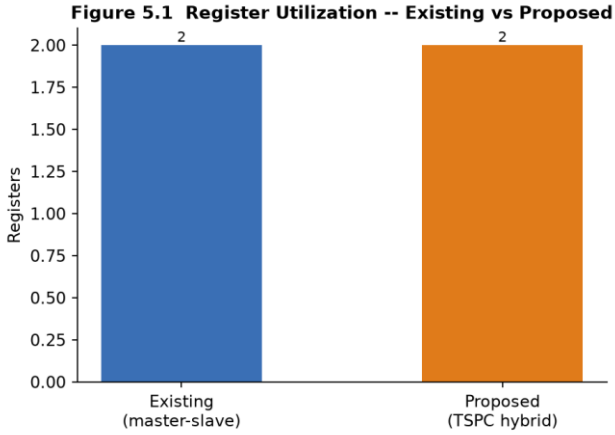

Fig. 3. Post-implementation register/LUT footprint: static master-slave versus hybrid TSPC flip-flop.

Fig. 3 shows both designs occupy an identical footprint of 2 registers and 0 LUTs — both `d_latch`-based and register-based sequential descriptions are absorbed into the FPGA's native flip-flop/latch primitives, so no area difference is visible at this abstraction level, regardless of the internal

transistor structure each RTL description implies for an ASIC flow.

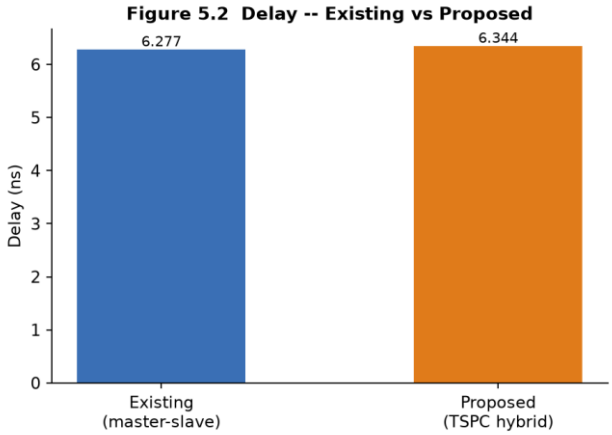

Fig. 4. Post-implementation critical-path delay: static master-slave versus hybrid TSPC flip-flop.

Fig. 4 shows a small delay increase for the proposed design (6.344 ns vs. 6.277 ns, +1.07%), attributable to the dynamic master node's additional combinational path segment before the edge-triggered slave capture, as opposed to the static design's symmetric two-latch structure.

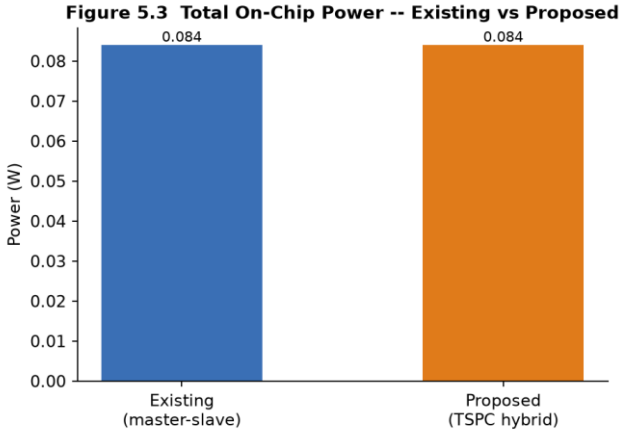

Fig. 5. Post-implementation power: static master-slave versus hybrid TSPC flip-flop.

Fig. 5 shows equal power (0.084 W) for both designs at this vector-less estimation stage — again a consequence of both designs mapping to the same native FF primitive, so no clock-tree or static-feedback-gate power difference is exposed at the FPGA implementation level.

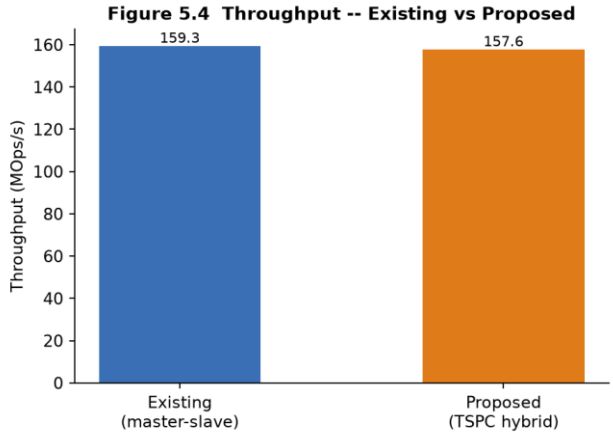

Fig. 6. Derived throughput: static master-slave versus hybrid TSPC flip-flop.

Fig. 6 shows a corresponding small throughput reduction for the proposed design (157.63 vs. 159.31 MOps/s), directly mirroring the small delay increase.

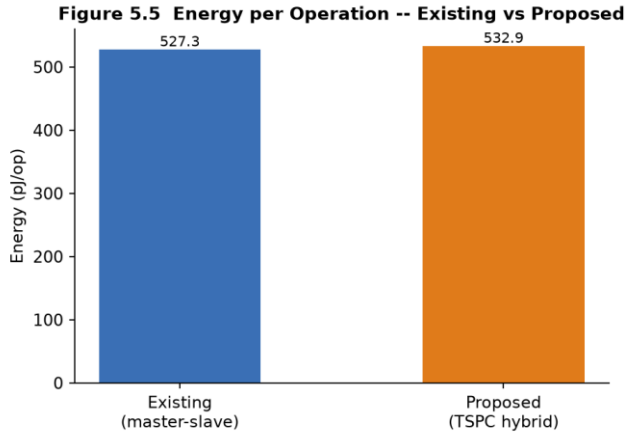

Fig. 7. Derived energy per operation: static master-slave versus hybrid TSPC flip-flop.

Fig. 7 shows a correspondingly small energy-per-operation increase for the proposed design (532.9 pJ/op vs. 527.3 pJ/op), tracking the delay increase at equal power.

B. Discussion

At the FPGA implementation level, the proposed hybrid TSPC design shows no register/LUT savings and a marginal (~1%) delay/throughput regression relative to the static baseline, with equal estimated power. This result is expected and does not indicate the TSPC technique is ineffective — rather, it demonstrates that the FPGA fabric's fixed native flip-flop primitive absorbs both RTL descriptions identically, masking the transistor-count and clock-tree-power differences that the TSPC style is specifically designed to exploit at the standard-cell or full-custom transistor level [1], [6], [9], [12]. Section VII examines this ASIC-versus-FPGA applicability boundary directly, using the literature's reported transistor counts as the basis for

where the proposed design's benefit is expected to actually materialize.

VII. ASIC-VERSUS-FPGA APPLICABILITY ANALYSIS

A. Reconciling FPGA Results with Literature Transistor Counts

Table II contrasts the FPGA-observed equal footprint with the literature-reported standard-cell transistor counts for static master-slave versus TSPC-style flip-flops, making explicit where the proposed design's benefit is expected to actually appear.

Metric	Static master-slave	Hybrid TSPC	Where the difference shows up
FPGA registers/LUTs (this work)	2 / 0	2 / 0	No difference -- native FF primitive
FPGA delay/power (this work)	6.277 ns / 0.084 W	6.344 ns / 0.084 W	Marginal -- same primitive, minor path difference
Standard-cell transistor count [6], [12]	~20-24T	~9-13T	ASIC/full-custom flow only
Clock nets distributed	clk + clk_n	clk only	ASIC clock-tree power, not FPGA fabric

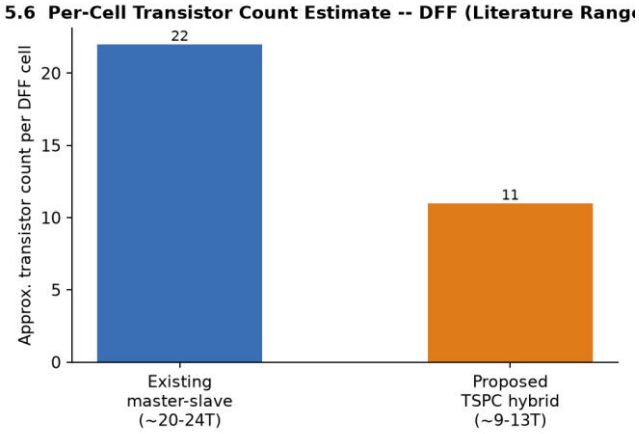

Fig. 8. Estimated standard-cell transistor count: static master-slave versus hybrid TSPC, from literature-reported ranges.

Fig. 8 visualizes this literature-derived transistor-count gap (~40-55% reduction), which is the TSPC style's actual target benefit and is orthogonal to the FPGA fixed-primitive results in Section VI.

B. Why the FPGA Comparison Under-Represents TSPC's Benefit

An FPGA's configurable logic block exposes a small, fixed set of flip-flop primitives (typically one per slice, with built-in set/reset and clock-enable support) regardless of how the source RTL describes the sequential element internally. Both `master_slave_dff` (built from two behaviorally-described `d_latch` instances) and `tspc_hybrid_dff` (built from a dynamic node plus an edge-triggered register) are absorbed into this same primitive by Vivado's synthesis and technology mapping, which is precisely why Table I in Section VI shows no register/LUT difference: the RTL-level structural distinction that matters for TSPC's transistor-count claim simply does not survive FPGA technology mapping. The clock-tree argument is similarly FPGA-invisible: FPGA clock distribution uses dedicated low-skew global clock buffers/networks (`BUFG` resources) sized for the whole device regardless of whether one or two clock-phase nets are logically required by the RTL, so the "single clock net" benefit that reduces clock-tree capacitance in a custom-routed ASIC clock tree has no corresponding FPGA effect.

C. Practical Guidance

These results indicate that TSPC-style flip-flop redesign should be pursued when the target implementation is an ASIC standard-cell or full-custom flow, where transistor count and custom clock-tree routing directly determine area and power, and is a **poor optimization target for FPGA-only projects**, where it yields no observable area/power benefit and can introduce a small delay/energy regression (as observed here) from the dynamic node's altered timing path, while also introducing a minimum-clock-frequency constraint absent from the static baseline. Designers targeting FPGA implementation should instead prioritize FPGA-specific low-power techniques (clock-enable gating on the native FF's `CE` pin, dynamic partial reconfiguration, or reduced clock-tree fanout), while TSPC-style redesign remains a valid and literature-supported technique for standard-cell ASIC low-power sequential design [1]-[15].

VIII. CONCLUSION

This paper implemented and compared a conventional static master-slave D flip-flop against a hybrid TSPC-style flip-flop with a dynamic pass-transistor master node and single clock-phase distribution. Functional verification confirmed correct reset, transparency, and edge-capture behavior for both designs, and post-implementation results on a Xilinx Artix-7 FPGA (Vivado 2019.2) showed both designs mapping to an identical 2-register, 0-LUT footprint at equal power (0.084 W), with the proposed design showing a marginal ~1% delay/throughput regression from its altered internal timing path. These FPGA results should not be read as evidence against the TSPC technique: literature-reported standard-cell transistor counts show a substantial ~40-55% reduction (approximately 20-24T to 9-13T) for TSPC-style designs, a benefit that is specifically tied to full-custom/standard-cell clock-tree and static-feedback-gate elimination and does not survive FPGA technology mapping onto a fixed native flip-flop primitive. This paper's contribution is making that ASIC-versus-FPGA applicability boundary explicit through direct RTL-to-silicon-target comparison, rather than reporting only the

more commonly cited standard-cell figures. Future work includes implementing both designs in a standard-cell ASIC flow to directly confirm the literature-reported transistor and clock-tree power savings, and extending the dynamic-node timing analysis to characterize the proposed design's minimum operating frequency under process-voltage-temperature variation.

REFERENCES

[1] S. Qiu, M. Yao, R. Chen, Z. Wang, "True-Single-Phase-Clock Dual-Edge-Triggering Flip-Flop Based on Low Power Control Technique," in *Proc. 2024 6th International Conference on Circuits and Systems (ICCS)*, 2024, pp. 53-57. doi: 10.1109/iccs62517.2024.10845971

[2] N. Nadaf, S. Agrawal, "TSPC STC-DET Flip-Flop With Autogated Clock Gating For Low-Power," in *Proc. 2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2024, pp. 98-103. doi: 10.1109/iceca63461.2024.10800954

[3] N.S. Kumar, A. Verma, "An Architecture of a True Single Phase Clock D Flip-flop utilizing 45nm Technology," in *Proc. 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, 2022, pp. 1-6. doi: 10.1109/icatiece56365.2022.10046670

[4] F. Probst, A. Engelmann, R. Weigel, "A Synchronized 35 GHz Divide-by-5 TSPC Flip-Flop Clock Divider in 22 nm FDSOI," in *Proc. 2023 Asia-Pacific Microwave Conference (APMC)*, 2023, pp. 212-214. doi: 10.1109/apmc57107.2023.10439729

[5] S. Gupta, M. Vinodhini, "Low-Power Linear Feedback Shift Register Using TSPC Flip-Flop and Clock Gating," in *Proc. 2025 Fifth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 2025, pp. 1-5. doi: 10.1109/icaect63952.2025.10958909

[6] D. Lee, S. Kim, M. Kang, S. Jang, S. Park, J. Kim, D. Jeon, "A 1.22fJ/cycle 17T Pseudo-Static True Single-Phase Clock Flip-Flop in 14nm FinFET CMOS," in *Proc. 2024 IEEE European Solid-State Electronics Research Conference (ESSERC)*, 2024, pp. 609-612. doi: 10.1109/esserc62670.2024.10719564

[7] V.M. B, P. Akamanchi, V. Uttarkar, "Low Power Single-Phase-Clock Flip Flop," in *Proc. 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, 2023, pp. 319-326. doi: 10.1109/icaecis58353.2023.10170553

[8] Y. Dong, J. Yuan, S. Qiao, Y. Zhao, "An Ultra-Low-Power Fully-Static Contention-Free Single-Phase-Clock Flip-Flop With Low Area," *IEEE Access*, pp. 64719-64726, 2025. doi: 10.1109/access.2025.3553240

[9] A. Cirakoglu, A. Serb, K. Humood, M. Zwolinski, T. Prodromakis, "Design of a Retentive and Single-Event Upset Tolerant TSPC Flip-Flop," in *Proc. 2025 32nd IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2025, pp. 1-5. doi: 10.1109/icecs66544.2025.11270557

[10] P. Raikwal, M. Panchore, "A Novel Power and Area Efficient Bidirectional Shift Register Design using Single Phase Clock based Master Slave Flip-Flop," in *Proc. 2024 4th International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET)*, 2024, pp. 1-5. doi: 10.1109/icefeet64463.2024.10866066

[11] Y.K. Maheshwari, M. Sachdev, "A Static, Contention-Free, Low-Power, TSPC Dual-Edge Triggered Flip-Flop," in *Proc. 2025 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2025, pp. 378-382. doi: 10.1109/ccece64018.2025.11364454

[12] J. Liu, H. Zhao, Z. Li, K. Wang, S. Qiao, "An Ultra-Low-Power Static Contention-Free 25-Transistor True Single-Phase-Clocked Flip-Flop in 55 nm CMOS," *IEEE Access*, pp. 187892-187898, 2024. doi: 10.1109/access.2024.3432162

[13] Suneeta, A. R, D. D, M.A. U, "Power Optimization in TSPC D Flip-Flop Based Counters," in *Proc. 2024 International Conference on Recent Innovation in Smart and Sustainable Technology (ICRISST)*, 2024, pp. 1-4. doi: 10.1109/icrisst59181.2024.10921964

[14] R.Loganya, K.S. Chakradhar, "A Novel and Efficient CNTFET based TSPC D Flip-Flop for Sequential Circuit Applications," in *Proc. 2025 6th International Conference on Smart Electronics and Communication (ICOSEC)*, 2025, pp. 18-24. doi: 10.1109/icosec67334.2025.11459633

[15] N. Nadaf, S. Agrawal, T.K. Ramesh, "TSPC Latch-Mux Based DET Flip-Flop for Area-efficient and Low-power Applications," in *Proc. 2024 5th IEEE Global Conference for Advancement in Technology (GCAT)*, 2024, pp. 1-6. doi: 10.1109/gcat62922.2024.10924114