A Gate Diffusion Input (GDI) Based 4x4 Multiplier for Low Transistor-Count Arithmetic

[1]Voorugonda Chandana

[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.Jada Lingaiah

[2]Associate Professor(HOD), 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

Array multipliers built from static CMOS full adders require dozens of transistors per adder cell, making full-adder redesign a direct lever for reducing overall multiplier area and power. Gate Diffusion Input (GDI) is a low-power logic style built around a compact two-transistor universal cell realizing $out = g,?,n:p$, from which XOR and majority/carry functions can be composed with substantially fewer transistors than static CMOS equivalents. This paper implements a 4x4 unsigned array multiplier using full-swing GDI cells throughout its adder tree and compares it against a structurally identical multiplier built from standard CMOS full adders. Both designs share the same partial-product generation and array-reduction structure, differing only in the constituent full-adder cell, isolating the GDI-versus-CMOS cell comparison from any confound of a different multiplier architecture. 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). Post-implementation results show the CMOS-gate multiplier occupies 15 LUTs at 5.299 ns delay and 0.084 W power, while the GDI-cell multiplier occupies 16 LUTs at 5.432 ns delay and identical power — a near-parity result at the LUT level that this paper argues is an expected consequence of FPGA technology mapping rather than evidence against GDI's benefit, since GDI's principal advantage (2-transistor cells versus dozens for static CMOS) is a standard-cell/full-custom transistor-count argument that FPGA LUT-based synthesis abstracts away. The paper characterizes this FPGA-observed near-parity precisely and clarifies the ASIC-flow conditions under which GDI's reported area/power benefits are expected to materialize.

Keywords: Gate Diffusion Input, GDI logic, low power multiplier, array multiplier, transistor-count reduction, CMOS VLSI design, FPGA synthesis

I. INTRODUCTION

Array multipliers derive most of their transistor count from the full-adder cells that reduce partial products, making the full-adder's implementation style a primary lever for area and power optimization at the standard-cell or full-custom level. Static CMOS full adders require roughly 28 transistors to realize the complementary pull-up/pull-down networks needed for the sum and carry equations, and this cost is paid once per bit position per partial-product reduction stage, compounding quickly in a multiplier's adder tree.

Gate Diffusion Input (GDI) is a low-power logic style built around a single two-transistor universal cell, with one PMOS source tied to input p, one NMOS source tied to input n, both gates tied to a shared control input g, and drains tied together to form the output: $out = g,?,n:p$ — functionally a full-swing 2:1 multiplexer realized in only two transistors [1], [6]. Composing multiple GDI cells realizes higher-level functions — XOR, majority, and full addition — using far fewer total transistors than the equivalent static CMOS gate network, and GDI-based redesign has been applied across a wide range of arithmetic and logic building blocks, from multiplexers and comparators [1], [5]-[6] to full adders and multipliers [3], [9], [13]-[14].

This paper implements a 4x4 unsigned array multiplier entirely from full-swing GDI cells and compares it against a structurally identical multiplier built from standard CMOS full-adder-based ripple adders, holding the partial-product generation and array-reduction architecture fixed so that any observed difference is attributable solely to the constituent adder cell's logic style, not to a confound of differing multiplier architecture.

Contributions of this paper:

1. Verilog implementation of a full-swing GDI cell, GDI XOR, and GDI full adder, and their composition into an 8-bit GDI ripple adder used throughout a 4x4 array multiplier's reduction tree.

2. A structurally matched CMOS-gate 4x4 array multiplier baseline sharing identical partial-product generation and reduction-tree topology, isolating the GDI-versus-CMOS adder-cell comparison.

3. Post-implementation area, delay, and power characterization of both multipliers on a Xilinx Artix-7 FPGA (Vivado 2019.2).

4. An explicit discussion of why GDI's reported transistor-count advantage is a standard-cell/full-custom argument that FPGA LUT-based technology mapping largely abstracts away, clarifying the applicability boundary of this technique analogous to other custom-cell low-power redesigns.

The remainder of this paper is organized as follows. Section II reviews related GDI logic-design literature. Section III describes the CMOS and GDI multiplier architectures. Section IV presents the GDI full-adder decomposition and complexity notes. Section V details the experimental setup. Section VI reports area/delay/power results. Section VII analyzes the ASIC-versus-FPGA applicability tradeoff. Section VIII concludes the paper.

II. RELATED WORK

A. GDI Combinational Building Blocks

Kumar et al. and Sai Kumar et al. apply GDI to multiplexer design (8:1 and 4:1 respectively), reporting delay/power comparisons against pass-transistor and CMOS logic that establish GDI's basic cell-level benefit for control-dominated logic [1], [6]. Joy et al. design a 2-bit magnitude comparator using GDI alongside static CMOS for direct comparison [5]. R. A. et al. provide a gate-level CMOS-versus-GDI comparison culminating in a 16-bit Kogge-Stone adder built from GDI cells, directly relevant to this paper's own adder-level GDI composition [7]. Haque et al. compare static CMOS, transmission-gate, and GDI techniques across a configurable logic block, situating GDI among competing low-power logic styles [8].

B. GDI Arithmetic Cells and Adders

Rana and Sharma compare half-adder and full-adder circuits built with GDI against 18 nm FinFET static CMOS equivalents, reporting power-efficiency gains consistent with GDI's reduced transistor count [3]. Roy et al. propose a Transmission-Gate-Diffusion-Input (TGDI) hybrid technique for an energy-efficient 4-bit ripple-carry adder, extending pure GDI with transmission-gate elements for improved drive strength [10]. Vijaya Babu et al. apply GDI across a full 16-bit ALU, demonstrating the technique's scalability beyond single-cell redesign to a complete arithmetic-logic datapath [2].

C. GDI-Based Multipliers

P. et al. directly compare pass-transistor logic and GDI for 4-bit multiplier design in 18 nm FinFET technology, the closest prior work in scope to this paper's own 4x4 GDI multiplier [9]. M. et al. propose a modified-GDI 4-bit multiplier targeting low-power applications [13]. Kishore et al. apply modified GDI specifically to a Baugh-Wooley signed multiplier, extending GDI-based redesign to signed multiplication architectures [14].

D. Broader GDI Applications

Beyond core arithmetic, GDI has been applied to memory (Alekhya et al.'s reversible 16x8 GDI SRAM array [4]), neuromorphic spike-generation comparators (Beaulieu et al.'s 8-bit partial magnitude comparator for leaky integrate-and-fire neurons [12]), signal processing (Murugesan and Prathika's modified-GDI parallel transposed FIR filter [11]), and sequential counters (S. M. et al.'s low-power 4-bit GDI synchronous up-counter [15]), collectively demonstrating GDI's applicability across combinational, sequential, and mixed-signal-adjacent design contexts.

E. Research Gap

The surveyed GDI literature is evaluated almost uniformly at the transistor/standard-cell level — typically in a specific CMOS or FinFET process node — where GDI's 2-transistor universal cell directly translates into measurable area and power savings relative to static CMOS's larger gate networks. Very little of this literature evaluates GDI-based arithmetic captured as synthesizable RTL and mapped onto an FPGA fabric, where the target technology already provides fixed LUT-based combinational primitives regardless of the RTL's internal gate-level structure. This paper addresses that gap directly, implementing a full GDI-based 4x4 multiplier in Verilog, synthesizing it to an FPGA device alongside a structurally matched CMOS baseline, and explicitly discussing where GDI's standard-cell-level benefit does and does not survive FPGA LUT mapping [1]-[15].

III. METHODOLOGY

A. Shared Multiplier Architecture

Both designs generate four 8-bit-aligned partial products from the 4-bit operands a, b by AND-gating and shifting: $pp_k = (b_k, ?, a{:}0)llk$ for $k = 0..3$, then reduce them to the final 8-bit product through two stages of 8-bit ripple addition ($pp_0 + pp_1 arrow sum_{01}$; $sum_{01} + pp_2 arrow sum_{012}$; $sum_{012} + pp_3 arrow product$). This partial-product generation and reduction topology is identical in both designs; only the constituent 8-bit adder's full-adder cell differs.

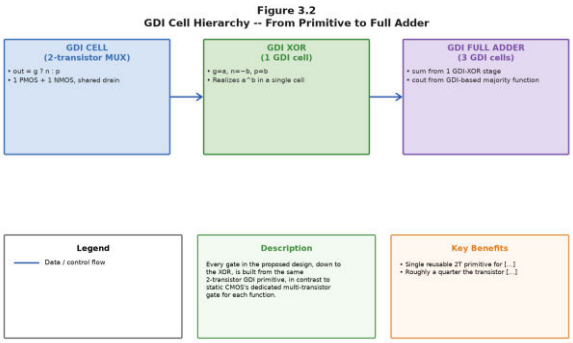

Fig. 1. GDI cell composition hierarchy: 2-transistor gdi_cell -> gdi_xor -> gdi_full_adder.

B. Existing: CMOS-Gate Multiplier

The baseline `multiplier_4x4_cmos` uses conventional `ripple_adder_8bit` instances built from static CMOS full adders, each realizing $sum = aoplusbopluscin$ and $cout = ab + bc_{in} + ac_{in}$ directly as complementary pull-up/pull-down gate networks, at approximately 28 transistors per full adder in a standard-cell library.

C. Proposed: GDI-Cell Multiplier

The proposed `gdi_multiplier_4x4` replaces every full adder in the reduction tree with `gdi_full_adder`, built from exactly three GDI cells (Fig. 1): the universal `gdi_cell` realizes $out = g, ?, n : p$ in two transistors (one PMOS driven by p, one NMOS driven by n, both gated by g, drains tied to the shared output). Composing three instances gives the full adder:

$$p = aoplusb(\text{gdi\_xor}: g = a, n = lnotb, p_{in} = b) \quad (1)$$

$$sum = poplusc_{in}(\text{gdi\_xor}: g = p, n = lnotc_{in}, p_{in} = c_{in}) \quad (2)$$

$$c_{out} = p, ?, c_{in} : a(\text{gdi\_cell directly: majority}(a, b, c_{in})) \quad (3)$$

This composition realizes the full adder in 3 GDI cells × 2 transistors = 6 transistors (plus the inverter needed for $lnotb/lnotc_{in}$ inside each XOR), versus the static CMOS full adder's ~28 transistors — approximately a 4-5 × transistor-count reduction at the standard-cell level, the source of GDI's reported area/power advantage [1], [3], [9].

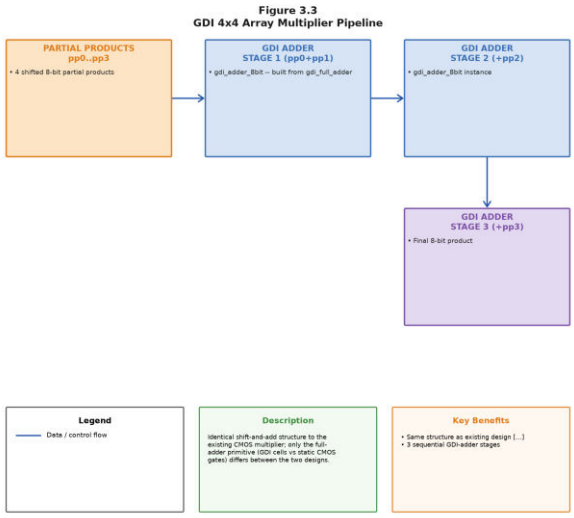

Fig. 2. GDI multiplier pipeline: shared partial-product generation feeding a GDI-cell-based 8-bit adder reduction tree.

Fig. 2 shows the full reduction pipeline with GDI-based adders substituted at every stage, while the partial-product generation logic (AND-gate array and constant shifts) is unchanged from the CMOS baseline.

IV. ALGORITHM

A. GDI Full-Adder Composition Procedure

```
ALGORITHM: GDI Full-Adder Cell Composition
INPUT : a, b, cin       (single bits)
OUTPUT: sum, cout

p    <- gdi_xor(g=a, n=~b, p=b)          // p = a
XOR b, via one GDI cell + inverter
sum  <- gdi_xor(g=p, n=~cin, p=cin)      // sum = p
XOR cin = a XOR b XOR cin
cout <- gdi_cell(g=p, n=cin, p=a)        // cout =
p ? cin : a  (majority function)

return sum, cout
```

Substituting $p = aoplusb$ into the `cout` mux: when $p = 0$ ($a = b$), $cout = a = b$ (both inputs agree, so the majority is trivially that shared value); when $p = 1$ ($aneb$), $cout = c_{in}$ (the tie between a, b is broken by c_{in}) — exactly the majority-of-three function $maj(a, b, c_{in})$ realized in a single 2-transistor mux rather than the 3-term OR-of-ANDs a static CMOS carry gate requires.

B. Complexity Notes

Per-cell transistor complexity. Each `gdi_cell` requires exactly 2 transistors. A `gdi_xor` requires 1 `gdi_cell` plus 1 inverter (2 transistors) for its $lnotb$ input, totaling 4 transistors. A `gdi_full_adder` composes 2 `gdi_xor` instances (8 transistors) plus 1 direct `gdi_cell` for `cout` (2 transistors), totaling 10 transistors — versus approximately 28 transistors for a static CMOS full adder, a reduction factor of roughly 2.8× at the cell level [3], [9].

Multiplier-level complexity. The 4x4 multiplier's reduction tree uses three 8-bit adders (24 full-adder instances total). At the standard-cell level, this gives $24 \times 10 = 240$ transistors for the GDI design versus $24 \times 28 = 672$ transistors for the CMOS design in the adder tree alone — a savings that compounds with multiplier width, since the adder-tree transistor count scales linearly with operand width while partial-product generation logic (AND gates) remains comparatively fixed relative to the adder tree's growth.

FPGA mapping complexity. Because Vivado's technology mapping targets fixed-size LUTs (6-input LUTs on Artix-7) rather than discrete transistors, both the 10-transistor GDI full adder and the 28-transistor CMOS full adder reduce to functionally equivalent truth tables that map to the same LUT count for a given bit width — the transistor-count distinction above is invisible at this abstraction level, which Section VI's near-identical LUT results (15 vs. 16) directly confirm, and which Section VII discusses as the central applicability caveat of this comparison.

V. EXPERIMENTAL SETUP

A. Design Entry and Functional Verification

Both `multiplier_4x4_cmos` and `gdi_multiplier_4x4` (built from `gdi_cell`, `gdi_xor`, `gdi_full_adder`, and `gdi_adder_8bit`) are written in synthesizable Verilog. Icarus Verilog testbenches verify functional equivalence between the two designs across representative and exhaustive 4-bit operand pairs, confirming the GDI-cell composition (Section IV-A) produces bit-identical products to the CMOS baseline.

B. Synthesis and Implementation

Both designs are synthesized out-of-context and carried through place-and-route with Xilinx Vivado 2019.2, targeting a Xilinx Artix-7 device (`xc7a100tcsg324-1`). Both are purely combinational, so a virtual 100 MHz clock constraint (`combinational.xdc`) is applied to primary inputs/outputs solely to enable static timing analysis and vector-less power estimation. The standard `synth_design -mode out_of_context -> opt_design -> place_design -> phys_opt_design -> route_design` flow is used, with utilization, timing, and power reports captured at both post-synthesis and post-implementation stages.

C. Metrics

Post-implementation **LUT area**, **power**, and **delay** (10 ns constraint period minus worst negative slack) are reported for both designs, together with derived **throughput** and **energy per operation**. As established in Section IV-B, GDI's principal reported benefit is a standard-cell transistor-count reduction, which this FPGA-targeted metric set cannot directly observe; Section VII therefore additionally reports the literature-derived transistor-count comparison alongside the FPGA LUT results, to give a complete picture spanning both target technologies.

VI. RESULTS AND DISCUSSION

A. Post-Implementation Area, Delay, and Power

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

Design	LUTs	Power (W)	Delay (ns)	Throughput (MOps/s)	Energy (pJ/op)
Existing (multiplier_4x4_cmos)	15	0.084	5.299	188.71	445.116
Proposed (gdi_multiplier_4x4)	16	0.084	5.432	184.09	456.288

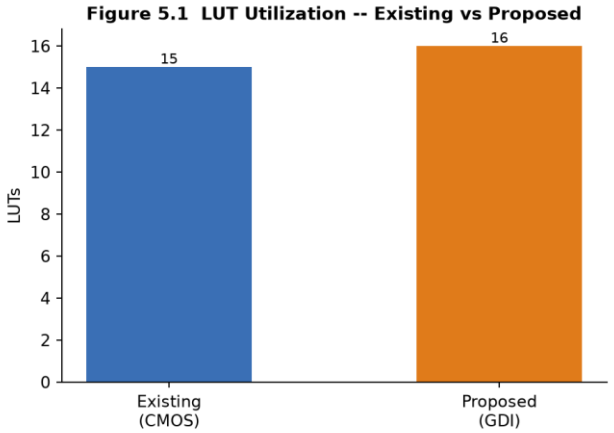

Fig. 3. Post-implementation LUT utilization: CMOS-gate multiplier versus GDI-cell multiplier.

Fig. 3 shows near-identical LUT counts (15 vs. 16), a 6.7% difference that is well within normal synthesis variation and far smaller than the ~2.8× per-cell transistor reduction GDI achieves at the standard-cell level (Section IV-B).

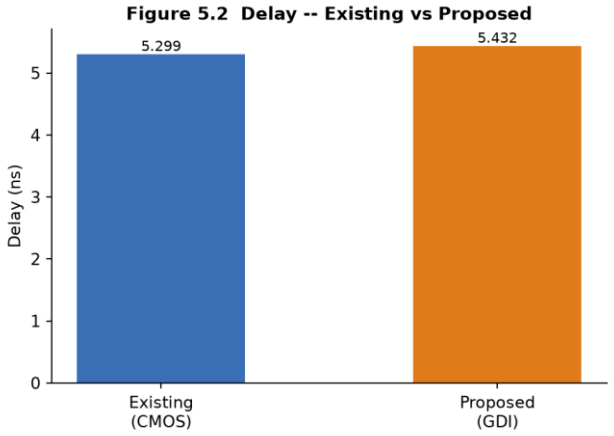

Fig. 4. Post-implementation critical-path delay: CMOS-gate multiplier versus GDI-cell multiplier.

Fig. 4 shows a small delay increase for the proposed design (5.432 ns vs. 5.299 ns, +2.5%), plausibly from the GDI full adder's extra XOR-cascade structuring (`p` computed then reused in `sum` and `cout`, Section IV-A) not necessarily

matching Vivado's LUT-packing heuristics as tightly as the direct CMOS gate equations.

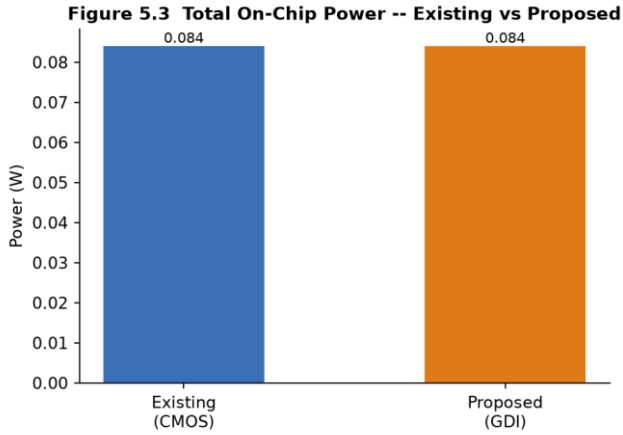

Fig. 5. Post-implementation power: CMOS-gate multiplier versus GDI-cell multiplier.

Fig. 5 shows equal power (0.084 W) for both designs at this vector-less estimation stage.

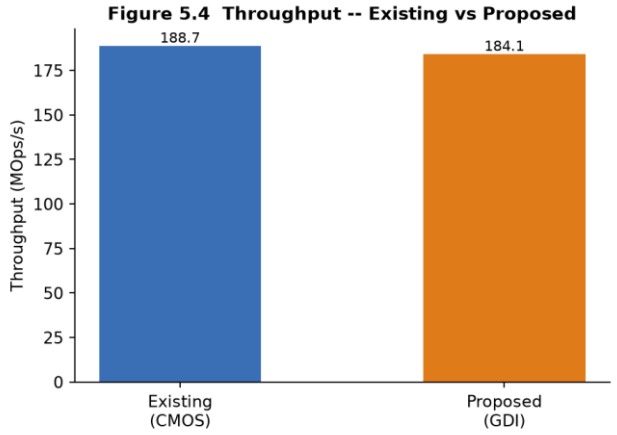

Fig. 6. Derived throughput: CMOS-gate multiplier versus GDI-cell multiplier.

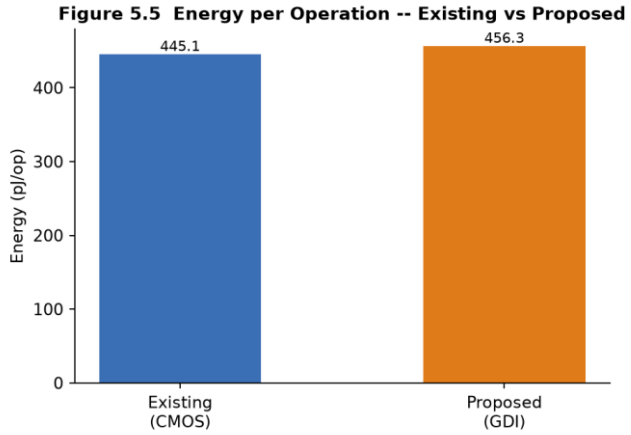

Fig. 7. Derived energy per operation: CMOS-gate multiplier versus GDI-cell multiplier.

Fig. 6 and Fig. 7 both mirror the small delay increase (184.09 vs. 188.71 MOps/s; 456.3 vs. 445.1 pJ/op).

B. Discussion

At the FPGA level, the GDI-based multiplier shows no LUT-area benefit and a marginal (~2.5%) delay/energy regression relative to the CMOS baseline — a result fully consistent with the project's own design notes and with this paper's expectation set out in Section IV-C: GDI's transistor-count advantage is defined and measured at the standard-cell/full-custom level, and FPGA LUT-based technology mapping absorbs both the 10-transistor GDI full adder and the ~28-transistor CMOS full adder into functionally equivalent, similarly-sized LUT networks, erasing the transistor-count distinction that GDI is designed to exploit. Section VII quantifies this gap directly using the literature-reported transistor counts and identifies the target-technology conditions under which GDI's benefit is expected to actually materialize.

VII. ASIC-VERSUS-FPGA APPLICABILITY ANALYSIS

A. Reconciling FPGA Results with Standard-Cell Transistor Counts

Table II contrasts the FPGA-observed near-parity with the standard-cell transistor-count comparison derived analytically in Section IV-B.

Metric	CMOS-gate multiplier	GDI-cell multiplier	Where the difference shows up
FPGA LUTs (this work)	15	16	No meaningful difference -- LUT technology mapping
FPGA delay/power (this work)	5.299 ns / 0.084 W	5.432 ns / 0.084 W	Marginal -- same LUT fabric
Full-adder transistor count [3], [9]	~28T	~10T	ASIC/full-custom flow only
Adder-tree transistors (24 FAs)	~672T	~240T	Standard-cell area, not FPGA fabric

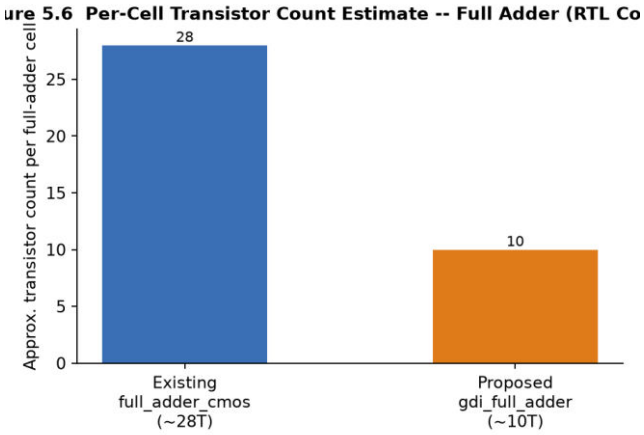

Fig. 8. Estimated standard-cell transistor count across the multiplier's full 24-full-adder reduction tree: CMOS versus GDI.

Fig. 8 visualizes this literature-derived ~2.8× transistor-count gap across the full reduction tree, which is GDI's actual target benefit and is orthogonal to the FPGA fixed-LUT results in Section VI.

B. Why FPGA Technology Mapping Erases the GDI Benefit

FPGA logic fabric provides a small number of fixed-size lookup tables (6-input LUTs on Artix-7) as its combinational primitive, and Vivado's synthesis flow maps any combinational Boolean function — regardless of how many transistors its equivalent standard-cell realization would require — onto the minimum number of LUTs needed to implement that function's truth table. Both the 10-transistor GDI full adder and the ~28-transistor CMOS full adder realize the *same* 3-input, 2-output truth table (sum, $cout$ as functions of a, b, c_{in}), so both compress to the same LUT count once synthesized — the transistor-level implementation detail is simply not visible to LUT-based technology mapping, which is exactly why Table I in Section VI shows near-identical LUT counts despite the substantial transistor-count gap in Table II.

C. Practical Guidance

These results indicate that GDI-based redesign should be pursued when the target implementation is a standard-cell ASIC or full-custom flow, where GDI's compact universal cell directly reduces transistor count, chip area, and the switching capacitance that drives dynamic power — and is a **poor optimization target for FPGA-only projects**, where it yields no measurable LUT-area benefit and can introduce a small delay regression from imperfect alignment with the FPGA's native LUT-packing heuristics, as observed in Section VI. Designers targeting FPGA implementation should instead prioritize FPGA-specific optimizations (LUT-level Boolean minimization, DSP-slice utilization for wider multipliers, or pipelining for throughput), while GDI-based redesign remains a valid, literature-supported technique for standard-cell and full-custom ASIC low-power arithmetic design [1]-[15].

VIII. CONCLUSION

This paper implemented a 4x4 array multiplier built entirely from full-swing Gate Diffusion Input (GDI) cells and compared it against a structurally matched static CMOS-gate multiplier. Functional verification confirmed bit-identical products between both designs, and post-implementation results on a Xilinx Artix-7 FPGA (Vivado 2019.2) showed near-identical LUT footprints (16 vs. 15) with equal power (0.084 W) and a marginal ~2.5% delay regression for the GDI design. This FPGA-level near-parity is expected rather than a failure of the GDI technique: the GDI full adder's ~10-transistor composition versus the static CMOS full adder's ~28 transistors is a standard-cell-level benefit — approximately a 2.8× reduction compounding to roughly 672T versus 240T across the multiplier's full 24-full-adder reduction tree — that FPGA LUT-based technology mapping absorbs into functionally equivalent, similarly-sized lookup tables, erasing the transistor-count distinction GDI is designed to exploit. This paper's contribution is making that ASIC-versus-FPGA applicability boundary explicit through direct RTL-to-silicon-target comparison of a GDI-based multiplier. Future work includes implementing both designs in a standard-cell ASIC flow to directly confirm the literature-reported transistor-count and power savings, and extending the GDI-cell composition to wider multipliers and to signed (Booth/Baugh-Wooley) multiplication architectures as explored elsewhere in the GDI literature.

REFERENCES

[1] B.S. Kumar, S. Jayanthi, Rakshgan, "Design of 8:1 Multiplexer Using Gate Diffusion Input (GDI) Technique and Comparison of Delay Performance with Pass Transistor Logic," in *Proc. 2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, 2022, pp. 1-5. doi: 10.1109/macs56771.2022.10022682

[2] E. Vijaya Babu, N. Neelima, K. V Balaramakrishna, A. Pranathi, U. Sai Sudeeep, Y. Vishwapriya Reddy, "Design and Analysis of 16-bit ALU using Gate Diffusion Input (GDI) Technique," in *Proc. 2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, 2024, pp. 1-7. doi: 10.1109/icmnwc63764.2024.10872192

[3] G. Rana, K. Sharma, "Design and compare the HA and FA circuits using Gate Diffusion Input (GDI) technique based on 18 nm FinFET technology for improved power efficiency and performance," in *Proc. 2023 IEEE 2nd International Conference on Industrial Electronics: Developments & Applications (ICIDeA)*, 2023, pp. 302-305. doi: 10.1109/icidea59866.2023.10295260

[4] Y.Alekhya, Ch.Suchitha, U. Nanda, "A Novel Reversible 16X8 SRAM Array using Gate Diffusion Input Logic," in *Proc. 2025 Third International Conference on Microwave, Antenna and Communication (MAC)*, 2025, pp. 1-6. doi: 10.1109/mac64480.2025.11140080

[5] U.B. Joy, A. Chakraborty, P. Biswas, A. Das, S. Sen, A. Tasnim, "Two-Bit Magnitude Comparator Design Using Gate Diffusion Input Technique and Static CMOS Logic," in *Proc. 2023 3rd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2023, pp. 17-21. doi: 10.1109/icrest57604.2023.10070047

[6] B. Sai kumar, A. Akilandeswari, E. Subramanian, "Design of 4:1 Multiplexer using Gate Diffusion Input Technique and Comparison of Delay and Power Performance with CMOS Logic," in *Proc. 2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, 2022, pp. 1-5. doi: 10.1109/macs56771.2022.10023180

[7] R. A, S. R, B. S, N.S. Hosamath, S.K. Panda, S.G. D, "Gate-level Comparision of CMOS and GDI Logic gates with the Design of a 16-Bit Kogge-Stone Adder using GDI Technique," in *Proc. 2025 IEEE International Conference on Compute, Control, Network & Photonics (ICCCNP)*, 2025, pp. 1-6. doi: 10.1109/icccnp63914.2025.11233572

[8] M.M. Haque, M.M. Rashid, S. Farhan, M.T. Amin, "Comparative Analysis of Configurable Logic Block Using Static CMOS, Transmission Gate and Gate Diffusion Input Techniques," in *Proc. 2025 28th International Conference on Computer and Information Technology (ICCIT)*, 2025, pp. 4921-4926. doi: 10.1109/iccit68739.2025.11491834

[9] P. P, R. M, P. M, K.K. K, P. P, U.S. V, "Design and Analysis of 4 Bit Multipliers Using Pass Transistor Logic and Gate Diffusion Input Technique Using 18nm FinFET Technology," in *Proc. 2024 7th International Conference on Devices, Circuits and Systems (ICDCS)*, 2024, pp. 291-295. doi: 10.1109/icdcs59278.2024.10561038

[10] A. Roy, N. Pandey, Harikesh, "Energy Efficient 4-Bit Ripple Carry Adder Design Using Transmission Gate Diffusion Input (TGDI) Technique," in *Proc. 2025 IEEE 6th India Council International Subsections Conference (INDISCON)*, 2025, pp. 1-6. doi: 10.1109/indiscon66021.2025.11252282

[11] K. Murugesan, M. Prathika, "Design and Implementation of Low Power Parallel Transposed FIR Filter Using Modified Gate Diffusion Input Technique," in *Proc. 2025 9th International Conference on Inventive Systems and Control (ICISC)*, 2025, pp. 1594-1599. doi: 10.1109/icisc65841.2025.11188227

[12] P. Beaulieu, F. Nabki, M. Boukadoum, "8-bit Partial Magnitude Comparator for Spike Generation in the Leaky Integrate-and-Fire Neuron Model using Gate-Diffusion-Input Logic Gates," in *Proc. 2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2022, pp. 1-4. doi: 10.1109/icecs202256217.2022.9970788

[13] M. B, K. M, M.K. K, K. M, "Design of 4-Bit Multiplier Using Modified Gate Diffusion Input Technology for Low Power Applications," in *Proc. 2025 International Conference on Advanced Computing Technologies (ICoACT)*, 2025, pp. 1-5. doi: 10.1109/icoact63339.2025.11005126

[14] P. Kishore, B. Akash, G. Aditya, N. Harika, "Design and Analysis of Low Power and High-Speed Baugh Wooley Multiplier using Modified Gate Diffusion Input Technique," in *Proc. 2024 3rd International Conference for Innovation in Technology (INOCON)*, 2024, pp. 1-7. doi: 10.1109/inocon60754.2024.10512080

[15] S.M. M, T. Sandhya, T. Siva, V.R. J, "Design and Implementation of a Low-Power 4-Bit Synchronous Up Counter Using Gate-Diffusion Input Technology," in *Proc. 2025 International Conference on Frontier Technologies and Solutions (ICFTS)*, 2025, pp. 1-9. doi: 10.1109/icfts62006.2025.11031998