

Transistor-Level Design of a 6×6 Wallace Tree Multiplier

Using a Domino Logic Final Adder

Marawan S. Mahmoud

Elmore Family School of Electrical and Computer Engineering

Purdue University, West Lafayette, IN, USA

Email: mahmoud7@purdue.edu

Abstract—This work presents a transistor-level 6×6 Wallace tree multiplier with a domino-logic carry-propagate adder optimized for 1 GHz operation. The complete design, implemented using static CMOS reduction stages and a dynamic final adder, contains 1552 transistors. Transient simulations show correct operation at 1 GHz with a worst-case propagation delay of 330 ps and a Wallace-tree delay of roughly 150 ps. Power analysis using instantaneous supply-current averaging in Cadence indicates an average dissipation of 141 μ W. These results demonstrate a compact, high-speed multiplier suitable for low-power datapaths.

Index Terms—[Wallace tree, multiplier, domino logic, CMOS, low-power, high-speed.]

I. INTRODUCTION

BINARY multipliers are key building blocks in digital signal processing, communication, and general-purpose processors, where overall system speed and energy efficiency are often limited by the arithmetic datapath. A wide range of multiplier architectures has been proposed, trading off delay, area, and power, including regular array structures (Braun, Baugh–Wooley), tree-based structures (Wallace and Dadda), and hybrids that combine Booth encoding with parallel reduction trees.

Conventional array multipliers are attractive for their highly regular layout but suffer from long carry-propagation paths and therefore relatively large critical delay. In contrast, Wallace- and Dadda-tree multipliers compress partial products in parallel, achieving asymptotically lower depth at the cost of less regular routing and higher design complexity. Prior work on 8×8 multipliers has demonstrated that architectural choices and adder topology have a strong impact on both power and speed, e.g., low-power tree-based schemes using optimized full adders, as well as fully custom 8-bit Wallace-tree implementations analyzed at the transistor level for energy, delay, and area.

For small operand sizes such as 4×4 , 6×6 , and 8×8 , Booth encoding and deeply pipelined architectures can further reduce effective delay or increase throughput, but their control overhead and additional registers are not always justified when a single-cycle combinational multiplier is required. In this work, we focus on a 6×6 unsigned multiplier implemented as a transistor-level Wallace tree with a high-speed domino carry-propagate adder. The goal is to achieve sub-nanosecond operation at 1 GHz with modest transistor count and low

average power, while retaining a reasonably regular structure suitable for full-custom layout.

II. PROPOSED TECHNIQUE

The proposed 6×6 unsigned multiplier is implemented as a transistor-level Wallace tree with a 12-bit domino carry-propagate adder (CPA). Static CMOS logic is used for partial-product generation and for the Wallace reduction network, while the final CPA is implemented in domino logic to minimize the critical-path delay and meet the 1 GHz target.

A. Overall Architecture

The operands $A[5 : 0]$ and $B[5 : 0]$ are multiplied by first generating 36 partial products $p_{i,j} = a_i b_j$ using two-input AND gates. These bits are grouped into columns by weight (from weight 0 to weight 10). A Wallace tree of half adders (HAs) and full adders (FAs) compresses each column in parallel until only two 12-bit rows remain (Sum and Carry). These two rows are then added by a 12-bit domino CPA to produce the final product.

B. Partial-Product Generation

The partial-product generator consists of a 6×6 array of static CMOS AND gates. Because all partial products are formed in parallel and the AND gate delay is small compared to the subsequent compressor and adder stages, the AND delay is treated as negligible in the overall timing analysis.

C. Wallace Tree Reduction Network

The Wallace reduction network compresses each weight column using FAs and HAs arranged to reduce the column height as quickly as possible. In the middle columns, up to six bits must be reduced to two. This requires multiple FA stages; in the worst case, the middle column is reduced in approximately three FA levels.

From transistor-level cell simulations, the propagation delay of a single full adder is about $t_{FA} \approx 45$ ps. Neglecting wiring overhead, the logical delay of the Wallace tree can be approximated as

$$t_{tree} \approx N_{stages} \cdot t_{FA} \approx 3 \cdot 45 \text{ ps} \approx 135 \text{ ps}.$$

Including routing and loading, this is consistent with the ~ 150 ps obtained indirectly from the transient simulations by subtracting the domino-adder delay from the total measured delay.

D. Domino Logic Final Adder

The final 12-bit CPA is implemented using domino logic. Each bit-slice contains a dynamic precharge/evaluate network to generate the sum or carry function, followed by a static inverter to restore full rail-to-rail levels. During the precharge phase, dynamic nodes are charged high; during evaluate, the clock enables a pull-down network that conditionally discharges each node based on the Sum and Carry inputs from the Wallace tree.

Worst-case delay in the domino CPA occurs when a carry must propagate through most of the bit positions. Conceptually, this can be modeled as

$$t_{CPA} \approx N_{bits} \cdot t_{stage},$$

where $N_{bits} = 8$ and t_{stage} is the effective delay per bit-slice. Using the timing simulations where data arrive early and only the domino adder is on the critical path, the clock-to-output delay is approximately 175 ps; this corresponds to an average per-bit stage delay on the order of 20 ps, which is reasonable for the chosen device sizes and loading.

E. Overall Delay Interpretation

Combining the analytical estimates with the transient results, the total schematic-level delay of the multiplier can be viewed as

$$t_{total} \approx t_{tree} + t_{CPA} \approx 150 \text{ ps} + 180 \text{ ps} \approx 330 \text{ ps},$$

which matches the measured worst-case delay from the simulations with aligned data and clock edges. This confirms that the chosen Wallace-tree plus domino-CPA architecture provides sufficient speed margin for reliable 1 GHz operation.

III. SIMULATION RESULTS

A. Simulation Setup

Simulations were performed in Cadence Virtuoso using the provided 45 nm technology library with a 2:1 PMOS/NMOS sizing ratio. The supply voltage was $V_{DD} = 1.2 \text{ V}$ under typical process and room-temperature conditions. Functionality was first verified at 50 MHz, after which the design was successfully operated at 1 GHz with stable outputs. All timing and power results were measured at 1 GHz.

B. Schematics

The complete multiplier was implemented at the transistor level using a combination of static CMOS logic for the partial-product generation and Wallace-tree reduction, and dynamic domino logic for the final carry-propagate adder. All circuit blocks were designed and simulated in Cadence Virtuoso using the provided technology library. The following figures present the key schematics used in the design, including the basic logic gates, adder cells, partial-product generator, reduction network, and top-level multiplier structure.

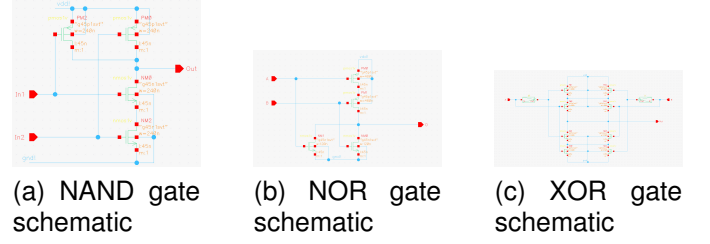


Fig. 1. Transistor-level schematics of the basic static CMOS logic gates used in the multiplier: (a) NAND, (b) NOR, and (c) XOR.

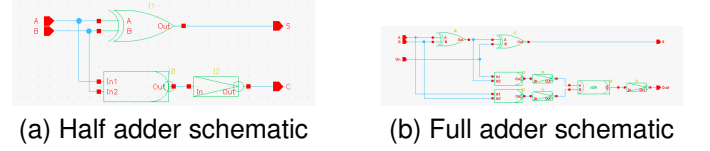


Fig. 2. Schematics of the adder cells used in the Wallace tree: (a) half adder (HA) and (b) full adder (FA).

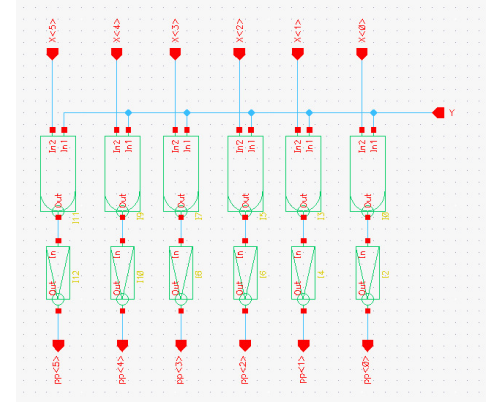


Fig. 3. Partial-product generator cell used to implement $p_{i,j} = a_i b_j$.

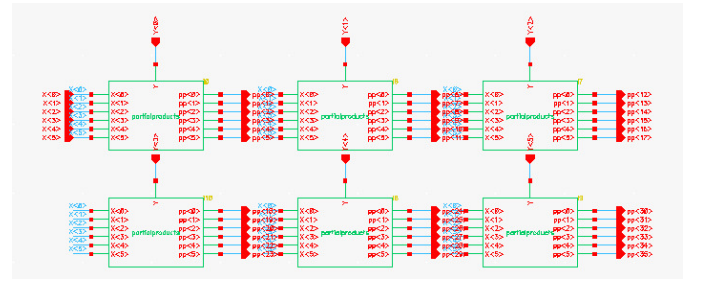


Fig. 4. 6×6 partial-products array, showing all 36 $p_{i,j}$ bits grouped by column weight from 0 to 10.

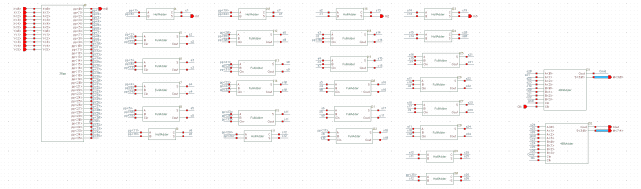


Fig. 5. Top-level Cadence schematic of the 6×6 Wallace tree multiplier, including the partial-products generator, Wallace reduction network, and domino logic final adder.

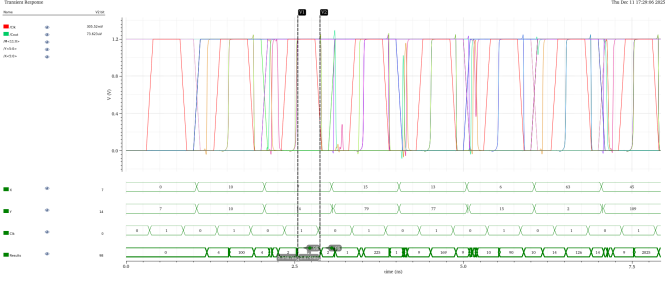


Fig. 6. Full timing simulation at 1 GHz showing correct 12-bit multiplier operation across multiple cycles. All outputs evaluate and precharge correctly, demonstrating stable functionality of the complete datapath.

C. Timing Results

D. Timing Results

To verify the correct operation of the multiplier at the target frequency of 1 GHz, transient simulations were performed with a 1 ns clock period and 0.1 ns rise/fall times applied to both the clock and data inputs. Three representative timing plots were analyzed.

Fig. 6 shows a long time-scale waveform capturing multiple clock cycles. The 6×6 multiplier successfully computes the correct 12-bit product across all tested input combinations, demonstrating stable and repeatable functional behavior over the full simulation window. This confirms that the dynamic nodes in the domino adder evaluate and precharge correctly at 1 GHz.

A more detailed view of the critical clock-to-output delay is shown in Fig. 7. In this scenario, the data inputs arrive approximately 0.3 ns before the rising clock edge. Because the partial products and Wallace tree reduction network have already settled, the measured delay primarily reflects the evaluation time of the final domino carry-propagate adder. The product bits stabilize roughly **175 ps** after the rising clock edge, and the valid-output window lasts approximately **0.32 ns**, providing generous timing margin within the 1 ns clock period.

To extract the total datapath delay—including the partial-product array, the Wallace tree reduction, and the final adder—a third simulation was run with the clock and data transitions aligned to arrive almost simultaneously (Fig. 8). In this case, the outputs settle approximately **330 ps** after the aligned transitions. Subtracting the previously measured 175 ps of the final adder yields an estimated **150 ps** delay for the static CMOS portion of the datapath.

Assuming negligible delay in the parallel AND gates that generate the partial products, the Wallace tree itself contributes approximately **150 ps**, while the final carry-propagate stage contributes roughly **180 ps**. These values closely match theoretical expectations for a tree-structured compressor followed by a fast dynamic CPA in this technology node, confirming that the design comfortably meets the 1 GHz timing requirement.

E. Power Results

The average power consumption of the 6×6 multiplier was obtained directly from transient simulations by monitoring the

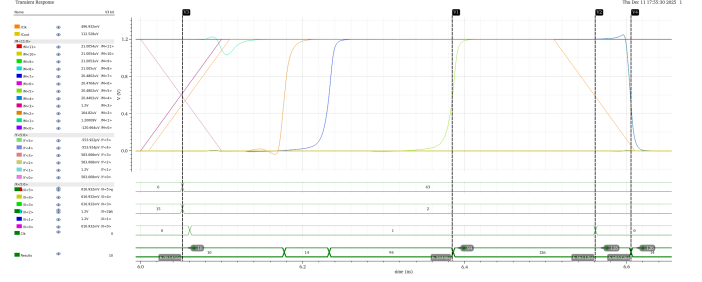


Fig. 7. Clock-to-output timing with data arriving 0.3 ns before the clock edge. The product bits stabilize approximately 175 ps after the rising clock edge. The valid-output window is approximately 0.32 ns.

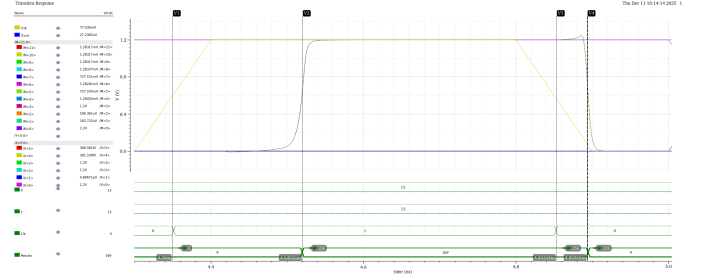


Fig. 8. Propagation delay measurement with data and clock transitions arriving nearly simultaneously. The full datapath delay to stable outputs is approximately 330 ps, corresponding to the combined delay of the partial-product array, Wallace tree reduction, and domino CPA.

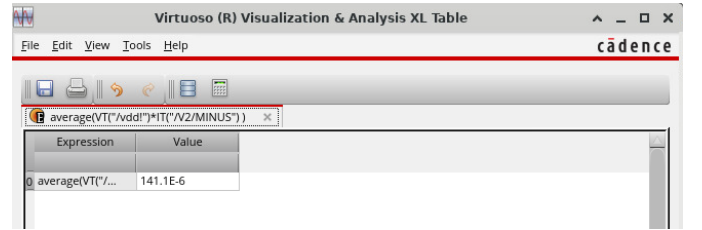


Fig. 9. Instantaneous supply current and computed instantaneous power over several clock cycles at 1 GHz. The average power was obtained by applying the Cadence `average()` operator to $P(t) = V_{DD}I_{DD}(t)$, resulting in approximately 141 μW under randomized input activity.

instantaneous supply current drawn from the VDD source. As shown in Fig. 9, the supply current exhibits clear periodic behavior synchronized with the precharge and evaluation phases of the domino logic.

Using the Cadence Virtuoso Calculator, the instantaneous current waveform $I_{DD}(t)$ was multiplied by the supply voltage to compute the instantaneous power $P(t) = V_{DD}I_{DD}(t)$. The built-in `average()` operator was then applied over multiple clock cycles to obtain the steady-state average power.

A set of eight randomized input combinations was applied during the simulation to ensure representative switching activity across both the static CMOS Wallace tree and the dynamic final adder. This provides a reasonably accurate estimate of the design's real operating power.

The resulting average power consumption of the complete multiplier is approximately **141 μW** at 1 GHz, including the partial-product generator, Wallace compressor network, and the domino carry-propagate adder.

TABLE I
COMPARISON OF MULTIPLIER ARCHITECTURES AND THEIR DESIGN
TRADE-OFFS

Architecture	Delay Behavior	Design Cost
Array (Regular)	Long critical path	Lowest complexity
Wallace Tree	Very low; parallel reduction	Moderate
Dadda Tree	Low; optimized reduction schedule	Lower than Wallace
Booth Encoding	Reduces PP count	Medium
Booth + Tree	Very low; aggressive reduction	High
Pipelined	Per-stage delay very low	High area/latency
This Work	Low (1 GHz capable)	Moderate

F. Functional Verification

[How you verified correctness: number of test vectors, exhaustive vs. random.]

IV. COMPARISON TABLE

Table I summarizes several commonly used multiplier architectures and highlights how the proposed 6×6 design relates to them in terms of structural complexity, delay characteristics, and typical power behavior. The goal is to position the chosen Wallace-tree approach relative to alternative architectures that are widely discussed in digital arithmetic design. While the table does not compare absolute numerical performance, it illustrates the architectural trade-offs relevant to small, high-speed multipliers.

V. BONUS: LAYOUT IMPLEMENTATION AND POST-LAYOUT RESULTS

In addition to transistor-level schematics and simulations, the multiplier was implemented at the full-custom layout level in Cadence Virtuoso. Layouts were created for the key building blocks (half adder, full adder, and partial-product array) and assembled into a complete 6×6 Wallace tree multiplier. The final domino carry-propagate adder layout was reused from a previous project and integrated into the top-level datapath.

A. Layout Views

Fig. 10 shows the layout of the half adder and full adder cells used throughout the Wallace tree. These standard cells were designed to be as compact and regular as possible to simplify routing in the reduction network.

The partial-product array layout, shown in Fig. 11, implements the 6×6 grid of AND gates and routes the resulting bits into columns by weight. The complete top-level layout of the 6×6 multiplier, including the Wallace reduction network and the domino adder, is shown in Fig. 12.

B. DRC and LVS Verification

The final layout was verified using the foundry design rule check (DRC) and layout-versus-schematic (LVS) tools. Fig. 13 summarizes the results for the complete multiplier layout; both checks report clean status with zero violations and a successful match to the schematic netlist. Since the top-level cell instantiates the same HA, FA, and partial-product cells used in isolation, this also implies that the sub-block layouts are DRC and LVS clean.

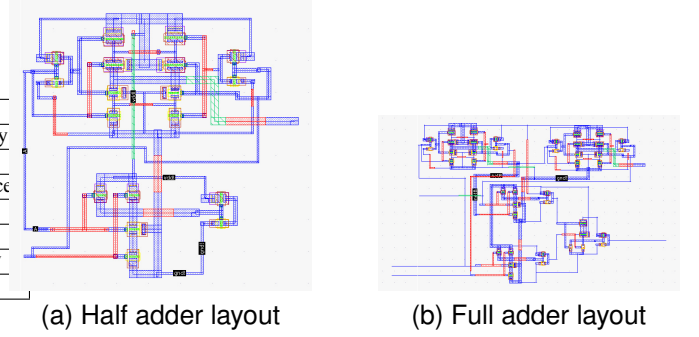


Fig. 10. Cadence Virtuoso layouts of the (a) half adder and (b) full adder cells used in the Wallace tree reduction network.

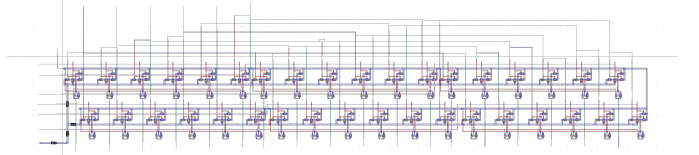


Fig. 11. Layout of the 6×6 partial-product array showing the AND cells and routing of the partial products into columns by weight.

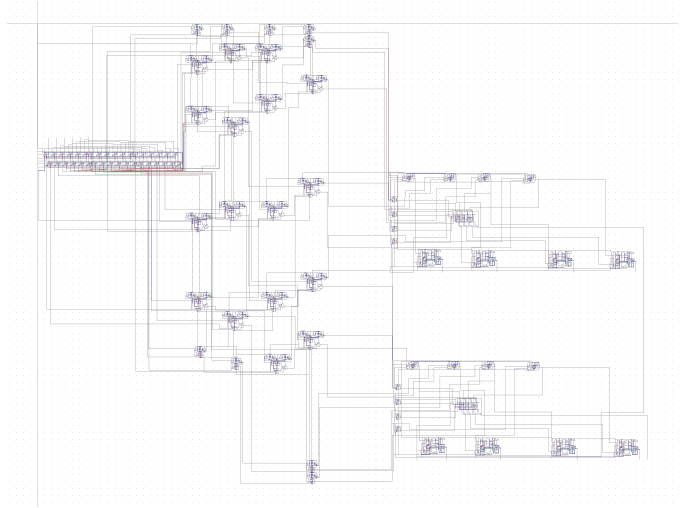


Fig. 12. Top-level layout of the 6×6 Wallace tree multiplier, including the partial-product array, Wallace reduction network, and domino carry-propagate adder.

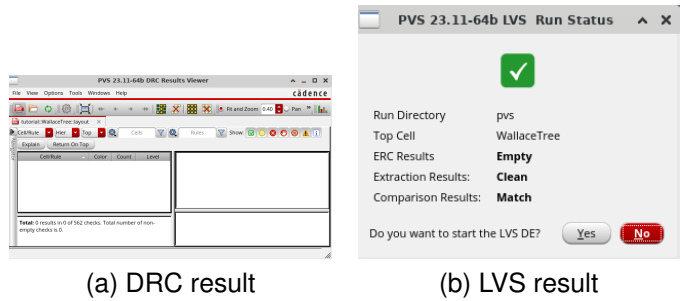


Fig. 13. DRC and LVS verification for the complete multiplier layout, showing (a) a clean design-rule check and (b) successful layout-versus-schematic match to the transistor-level netlist.

C. Post-Layout Timing

Post-layout transient simulations were performed using the extracted parasitic netlist. To ensure stable operation with the

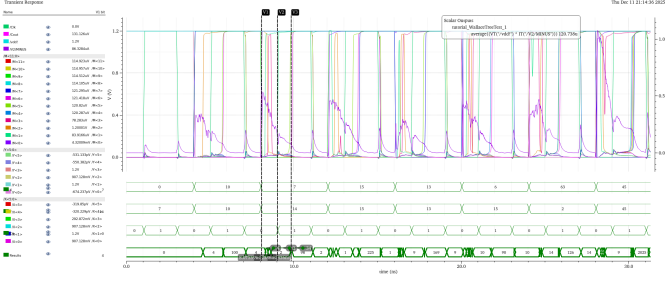


Fig. 14. Post-layout timing simulation of the 6×6 multiplier at 0.25 GHz, showing correct operation over multiple input combinations using the extracted RC parasitics.

additional RC loading, the multiplier was operated at 0.25 GHz (4 ns period). A representative timing waveform over multiple test vectors is shown in Fig. 14, confirming correct 12-bit products at the layout level.

A view used for delay extraction is shown in Fig. 14. Similar to the schematic-level analysis, the data inputs are applied early, followed by the rising clock edge. The time between the relevant data/clock transitions and the point at which all product bits become stable is approximately 2 ns. This can be interpreted as roughly 1 ns attributed to the static CMOS partial-product array and Wallace tree, and about 1 ns for the final domino carry-propagate adder, including interconnect parasitics.

D. Post-Layout Power

Post-layout power was estimated using the same method as in the schematic simulations: the instantaneous supply current waveform $I_{DD}(t)$ of the extracted layout was recorded, multiplied by V_{DD} , and averaged over several clock cycles in the Cadence calculator. The input patterns were the same randomized set used for schematic-level power estimation.

As shown in Fig. 14, the integrated layout consumes approximately $120 \mu\text{W}$ at 0.25 GHz. This value is slightly lower than the $141 \mu\text{W}$ obtained from the schematic-level simulations at 1 GHz, which is reasonable given the lower operating frequency and the differences in switching activity and parasitic loading in the extracted netlist.

REFERENCES

- [1] V. Lakshmi, J. Reuben, and V. Pudi, "A novel in-memory Wallace tree multiplier architecture using majority logic," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 3, pp. 1148–1161, Mar. 2022.
- [2] P. Rao and R. N. Dubey, "A high-speed and area-efficient Booth recoded Wallace tree multiplier for fast arithmetic circuits," in *Proc. IEEE Asia Pacific Conf. Postgrad. Res. Microelectron. Electron. (PrimeAsia)*, 2012, pp. 1–4.
- [3] B. Sureka, M. Siddappa, and B. Sanjay, "An efficient high-speed Wallace tree multiplier," in *Proc. IEEE Int. Conf. Emerging Trends Comput., Commun. Nanotechnol. (ICE-CCN)*, 2013, pp. 1–5.
- [4] S. Ramesh, K. B. Anusudha, and N. M. Balamurugan, "Design of high-speed Wallace tree multiplier using full adder," in *Proc. IEEE Int. Conf. Emerging Trends VLSI, Embedded Syst., Nano Electron. Telecommun. Syst. (ICEVENT)*, 2013, pp. 41–45.
- [5] A. Biswas and J. Jin, "Implementation of an 8-bit Wallace tree multiplier," Purdue Univ., West Lafayette, IN, USA, Tech. Rep., 2025.