Implementation of Efficient 4x4 Column by Passing Multiplier Using 2:1 Multiplexer Based Adder

Dr.P,Gayathri[1], G.Nandini[2], K.Jagadeeshwar[3], K.Abhiram[4]
[1]*Associate Professor, Dept of ECE, TKR College of Engineering and Technology.*
[2,3,4] *Student, Dept of ECE, TKR College of Engineering and Technology.*

***Abstract*—Multiplication is a basic computation in digital signal processing and VLSI design for which speed, power, and area need to be optimized. An efficient 4×4 Column Bypassing Multiplier implemented with a 2:1 multiplexer-based adder on Tanner EDA is discussed in this paper. Column bypassing saves dynamic power dissipation by bypassing computations, if some of the multiplier bits are zero, thus improving energy efficiency. To further improve performance, a 2:1 multiplexer-based adder is utilized, using fewer transistors and propagation delay than standard adders.**

The design is simulated in Tanner EDA with CMOS technology, and its performance is compared on the basis of power dissipation, delay, and area usage. Simulation results show that the proposed multiplier architecture has substantial power savings and better computational speed over traditional multipliers and is well suited for low-power VLSI and embedded systems. The method offers a power-efficient and high-speed multiplication solution and opens the door to further energy-constrained digital design optimizations

***Keywords*—Column Bypassing Multiplier, 2:1 Multiplexer-based Adder, Tanner EDA,CMOS Technology**

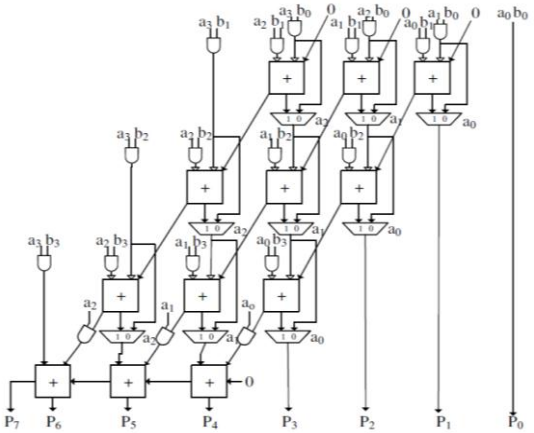
I.INTRODUCTION

Multiplication is an integral arithmetic operation and is utilized across a number of digital applications such as image processing, cryptography, digital signal processing (DSP), and microprocessors. Since advanced computational platforms need increased speed along with decreased power consumption, minimizing multipliers has become vital for improving overall performance. Traditional multipliers, such as array multipliers and booth multipliers, are not suitable for power-critical applications due to their high power usage and propagation delay. To mitigate these problems, some low-power multiplier architectures have been suggested, such as Column Bypassing Multipliers and Row Bypassing Multipliers. These methods assist in eliminating unnecessary calculations when some bits of the multiplier are zero, resulting in dynamic power reduction.

Of these, the Column Bypassing Multiplier (CBM) stands out is particularly noteworthy for its capacity to optimize power consumption and speed by bypassing certain columns of addition operations that are not essential for obtaining the final result. Here, a 4×4 Column Bypassing Multiplier is designed with a 2:1 multiplexer-based adder in Tanner EDA. In contrast to traditional adders, the multiplexer-based adder minimizes transistor usage and enhances speed, thus being a cost-effective option for VLSI-based low-power designs.

The Tanner EDA tool is employed for design, simulation, and performance analysis of the proposed multiplier using CMOS technology for power-efficient implementation. The design is analyzed in terms of power consumption, propagation delay, and area utilization, showing considerable improvements over conventional architectures. This work emphasizes the efficiency of using column bypassing techniques along with multiplexer-based adders to provide a power-efficient and high-speed multiplication unit, which is suitable for energy-constrained embedded systems and next-generation VLSI circuits.

BLOCK DIAGRAM



II. DESIGN PROCEDURE

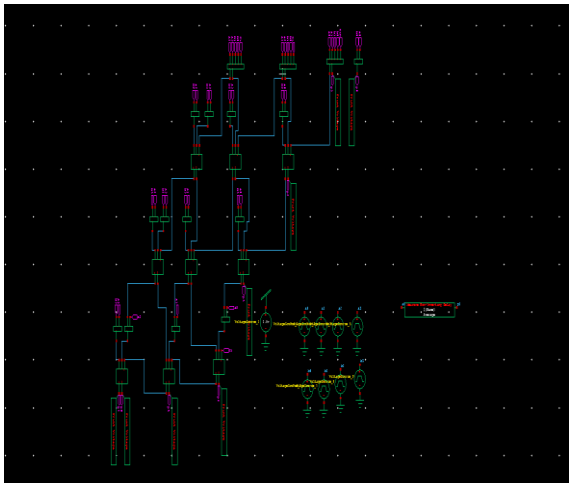


Fig 1: Schematic Diagram

UNDERSTANDING THE BASICS

- Multiplicand (A): A 4-bit number (say A3A2 A1A0).
- Multiplier Bit (Bi): One bit of the multiplier. In a typical parallel multiplier, you would have several such stages for each multiplier bit. This description is concerned with one such stage (a 4x1 bit multiplication).
- Partial Product: The product of the multiplicand and one bit of the multiplier. If the multiplier bit (Bi) is 1, then the partial product(pp) is the same as the multiplicand. If Bi is 0, the partial product is 0000.
- Column Addition: In a multiplier, you generate multiple pps (one for each bit of the multiplier). Here pps are added together in columns after being shifted to obtain the final product. This description is centered on the addition of one such pp to a running sum (or another partial product from an earlier stage).

GENERATING THE PARTIAL PRODUCTS USING 2:1MULTIPLEXERS

- To obtain one multiplier bit (Bi), we have to produce a 4-bit partial product and can be done with four 2:1multiplexers.
- To produce multiplicand's each bit (A j, j = 0, 1, 2, 3), we supply it to one port of a 2:1 multiplexer. The second port of each multiplexer is wired to 0. The multiplier bit (B i) controls the select line of all four multiplexers.
- Here's how it works:
  If Bi=1: The multiplexers take the input attached to multiplicand bits (A3,A2,A1,A0). The result of the multiplexers will be the partial product 1×A=A3 A2 A1 A0.
  If Bi=0: The multiplexers take the input attached to 0. The result of the multiplexers will be the partial product 0×A=0000.

THE BYPASSING FEATURE

- The employment of multiplexers automatically gives a "bypass" mechanism. If the multiplier bit is 0, the partial product is just zero. In a larger multiplier, if a partial product is zero, the next addition stage essentially just passes the prior sum along. This can result in some power optimization and possibly speed in more complicated multiplier architectures.

ADDING THE PARTIAL PRODUCTS TO RUNNING SUM

- Now, suppose this 4-bit partial product is to be combined with a 4-bit (or possibly wider, depends on the step) carry-in running sum from previous partial product sums. This would typically be incorporated into a 4-bit adder, which can be constructed using full adders.

IMPLEMENTING THE 4-BIT ADDER USING 2:1 MULTIPLEXERS

- Though less often done for performance concerns, it is theoretically possible to build a adder using strictly 2:1 multiplexer. A full adder takes three inputs (two to be added and carry-in) and produces two outputs (sum and carry-out). Multiplexers are used to make sum and carry-out logic.
- To build a 4-bit adder, you would string four full adders together, with the carry-out of one stage being the carry-in to the next stage. Any of these full adders could theoretically be constructed using multiplexers.

PUTTINGH IT TOGETHER FOR A 4x4 Column
A 4x4 multiplier would feature four of such stages, each for each of the 4 bits of the 4-bit multiplier.
1. Stage 0 (Multiplication of Multiplier Least Significant Bit):
Produce a 4-bit partial product through the multiplication of the 4-bit multiplicand and the multiplier's least significant bit using the 2:1 multiplexer technique discussed above.
2. Stage 1 (Multiplication of the Next Bit of the Multiplier):
Create another 4-bit partial product by multiplying

the multiplicand with the next bit of the multiplier. This partial product would generally be shifted left one place.

Next, this shifted partial product must be added to the Stage 0 result. This addition would be done conceptually with a 5-bit adder (to allow for possible carry).

3. Stage 2 and Stage 3:

Repeat the same procedure for the other two bits of the multiplier, with corresponding left shifts and additions. The adders in these steps would have to be designed for the increasing number of bits in the partial sums.

THE ROLE OF 2:1 MULTIPLEXERS IN ADDITION

- Although multiplexers are very good for the partial product generation (the "multiplication" by a single bit), having only 2:1 multiplexer to construct area-efficient and high-speed multi-bit adders is in general not the most useful idea in high-speed designs. Common adder architectures such as Carry-Ripple Adders, Carry-Lookahead Adders, etc., are area- and speed-efficient.

But ideologically, the multiplexer selection logic can be employed to realize the truth table of a full adder.

III.RESULTS AND DISCUSSIONS

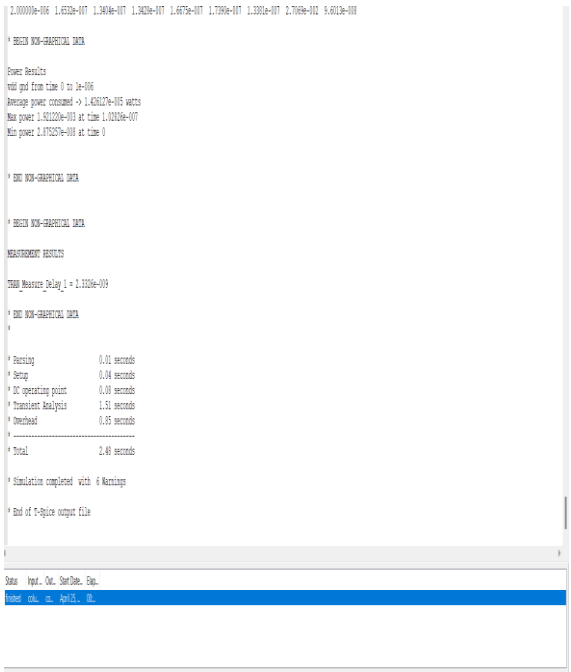


Fig 2:power consumption and delay

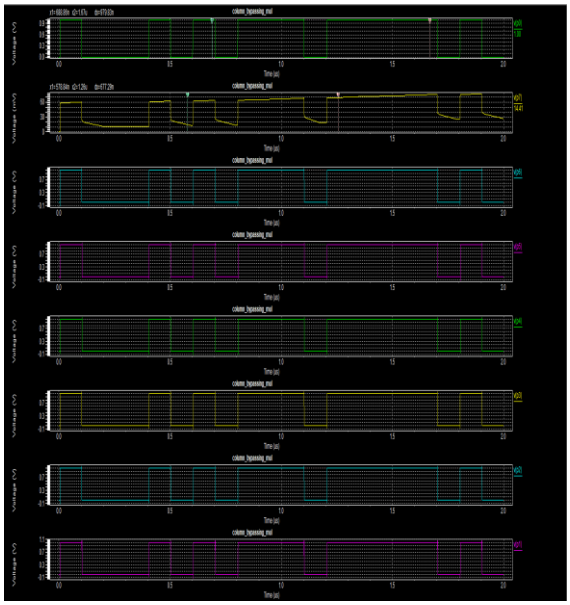


Fig 3: waveforms

IV. FUTURE SCOPE

The proposed design has the following limitations Bypassing Overhead, Delay in the Critical Path, Timing of MUX Control and these can be overcome by the following measures.

Scaling to Larger Bit Widths (8×8, 16×16, etc.): The 4×4 structure can be scaled to 8×8, 16×16, or even larger multipliers .Column bypassing and MUX optimization will still play a role in reducing switching activity and power consumption even for top-level designs.

Application in Low Power VLSI Designs Since column bypassing minimizes redundant switching, it will be particularly well-suited for low-energy ICs, such as portable processors, IoT devices, or implantable bio-medical devices where energy efficiency is highly critical.

Integration in DSP (Digital Signal Processing) Systems:

Multipliers are extensively used in DSP blocks (e.g., FIR filters, FFT processors).

Column bypassing multipliers can increase overall speed and energy efficiency of DSP chips.

ASIC and FPGA Implementations

The structure can be synthesized to ASICs or optimized to FPGAs where power and area are not abundant.

Applying 2:1 MUXes for bypass logic makes it extremely hardware-friendly for implementation. Combination with Other Multiplier Techniques: Can be hybridized with techniques such as Booth encoding, Wallace trees, or Dadda multipliers to

produce faster and even more power-efficient multipliers.

Dynamic Bypassing Logic research:
Rather than static column bypassing, dynamic methods (bypass solely on runtime information) can be researched for additional optimization.

V. ADVANTAGES

1.AREA EFFICIENCY:
A bypassing logic with multiplexer implementation uses fewer transistors than full adder logic.

- Lessens the silicon area, and thus is good for low-area embedded systems.

2.REDUCED POWER CONSUMPTION:

- Bypassing suppresses unnecessary switching activities in non-contributing columns (e.g., because of zero inputs).
- Multiplexers suppress the activation of full adder logic when not required.
- This results in dynamic power saving, particularly in sparse input applications.

3.REDUCED POWER SUPPLY:

- The power usage is reduced.

VI.CONCLUSION

The suggested 4×4 Modified Column Bypassing Multiplier is effectively power-efficient VLSI implementation through minimizing dynamic power consumption by selective bypassing of columns. The design minimizes switching activity to maximize energy efficiency with computational accuracy and performance. The simulation results confirm its low power dissipation, area overhead reduction, and improved efficiency, thus being suitable for low-power DSP, embedded systems, and IoT applications. This paper illustrates the potential of power-optimized arithmetic circuits in contemporary VLSI design, opening the door to future enhancements in high-speed and low-power computing architectures.
A 2:1 multiplexer-based 4x4 column adder is a reliable method of addressing arithmetic computations within digital systems. The multiplexer acts as the control system that allows the system to select skipping the multiplier for a case when a multiplication does not occur, sending one of the inputs through to the adder directly .This degree of flexibility allows for the improvement of the performance of the system by eliminating unwanted computation, meaning that only the necessary operations are carried out as per the exact needs of the task.
In real-time systems, including signal processing, embedded systems, and control systems, this technique has certain benefits. By avoiding multiplication where it can, the system minimizes the complexity of the calculations, and the processing happens faster with less latency. Multiplexer-based design enables swift decision-making whether to do the addition or the multiplication, and therefore, it is perfect for time-critical applications where speed becomes the necessity. This mechanism assists the system to perform better without being forced to withstand unnecessary multiplication cycles.

VII. ACKNOWLEDGEMENT

We Acknowledge Dr.P.Gayathri for their patience checking in Manuscript and Result Validations.

REFERENCES

[1] D.Padmashri, V.Santosh kumar “High Performance of Booth Multiplier For DSP “International Journal of Electronics, Electrical and Computation System IJEECS Issn 2348-117x Vol.6 Issue 9 Sept 2017.
[2] N.N Gopal m Papa Rao,V Shiva “VLSI Design of High Performance Complex Multipliers” International Journal of Engineering Inventions Issn : 2278-7461, Volume 5Issue 1 2016.
[3] Manchal Abuja,Sakshi “Design of Bypassing Multiplier with Different Adders Universal journal of Electrical and Electronic Engineering 217-221,2014.
[4] B. Shao and P. Li, Array-Based Approximate Arithmetic Computing: A General Model and Applications to Multiplier and Squarer Design. IEEE Transactions on Circuits and Systems I: Regular Papers.vol.62, pp. 1081-1090 (2015)
[5] C. Senthilpari, A.K. Singh, and K. Diwakar. Design of a low-power, high performance, 8×8 bit multiplier using a Shannon-based adder cell. Microelectronics Journal. vol. 39, pp. 812-821 (2008)
[6] M. Jhamb and H. Lohani. Design, implementation and performance comparison of multiplier topologies in power-delay space. Engineering Science and Technology, an International Journal. (2015)