

Design of Area Efficient High Speed Parallel Multiplier Using Low Power Technique on 0.18um technology

Vamsi Krishna Rongali¹, B Srinivas²

Department Of ECE, MVGR College of Engineering, Vizianagaram, Andhrapradesh

ABSTRACT: Based on the simplification of the addition operations in a low-power bypassing-based multiplier, a low-cost low-power bypassing-based multiplier is proposed. Row-bypassing multiplier, column-bypassing multiplier and bruan multipliers are implemented in conventional method and GDI technique. By optimizing the transistor size in each stage the power and delay are minimized. The results of post-layout simulation compared to similar reported ones illustrate significant improvement. Simulation results show great improvement in terms of Power-Delay-Product (PDP). the experimental results show that our proposed low-cost low power multiplier saves hardware cost and reduces the power dissipation.

Keywords: Row bypassing multiplier, Column bypassing multiplier, GDI , and power dissipation.

I. INTRODUCTION

As we get closer to the limits of scaling in complementary metal oxide semiconductor (CMOS) circuits, speed issues are becoming more and more important. In recent years, the impact of pervasive computing and the internet have accelerated this trend. The applications for these domains are typically run on battery-powered embedded systems.

The resultant constraints on the speed require design for speed as well as design for performance at all layers of system design. Thus increasing speed is a key design goal for portable

computing and communication devices that employ increasingly sophisticated signal processing techniques.

However, there is a fundamental trade-off between efficiency and flexibility, and as a result, programmable designs incur significant performance and speed compared to application specific solutions. Consequently various digital signal processing chips are now designed with high speed performance. Signal processing applications typically exhibit high degrees of parallelism and are dominated by a few regular kernels of computation such as multiplication, that are responsible for a large fraction of execution time and energy. In such systems, multiplier is a fundamental arithmetic unit. Shrinking feature sizes are responsible for increasing delay-related problems as well.

II. DIFFERENT TYPES OF MULTIPLIERS

2.1 INTRODUCTION

This chapter reveals the design considerations of High Speed parallel multiplier . The design of efficient logic circuits is a fundamental problem in the design of high performance processors. The design of fast parallel multipliers is important, since multiplication is a commonly used and expensive operation.

This is particularly critical for specialized chips that support multiplication intensive operations, such as digital signal processing and graphics. It can also be useful for pipelined

CPUs, where faster multiplier components and multipliers can result in smaller clock cycles and/or shorter pipelines.

The various multipliers are:

1. 4x4 Braun multiplier
2. 4x4 Row bypassing multiplier
3. 4x4 Column bypassing multiplier

2.2 4x4 Braun multiplier

The Braun multiplier removes the extra correction circuitry needed. Also number of adders is less. But, the limitation of this technique is that it cannot stop the switching activity even if the bit coefficient is zero that ultimately results in unnecessary time delay.

Another high speed designs disable the operation in some rows, designed a technique that reduces the switching to fairly good extent.

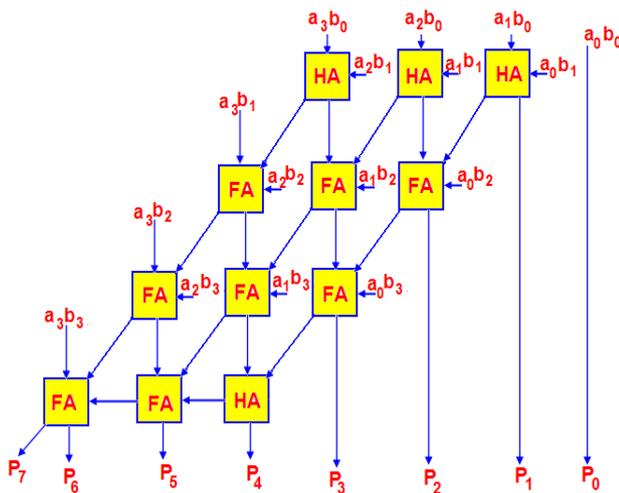


Fig 2.1: Schematic diagram of Braun multiplier

2.3 4x4 Row bypassing multiplier

The Row bypassing multiplier reduces the switching activity by bypassing the row in which the multiplicand bit is zero. That means in the multiplier if a bit is zero then that row of adders will get disabled. For example consider the multiplication of 1011 x 1010. Here the multiplier consists of zero in first and third positions. During multiplication the first and third row of adders get disabled and previous sum is taken as the present sum.

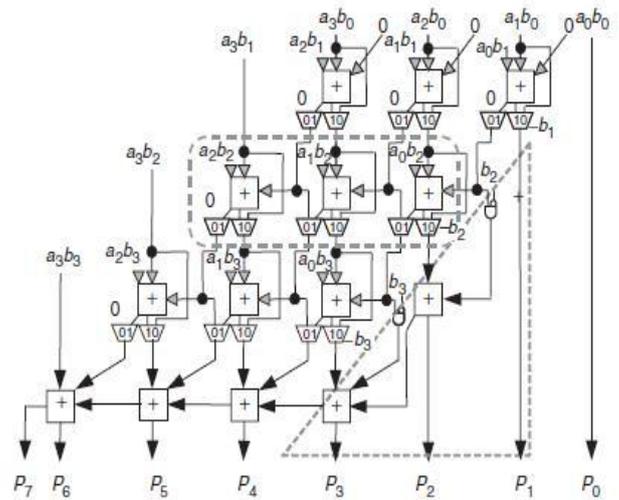


Fig 2.2: Schematic diagram of Row by passing multiplier

Here a special circuitry called adding cell is used instead of full adders. It consists of three state gates, full adder and multiplexers. The inputs i.e. the partial products to be summed up are given to the full adder through three state gates. The enable input to the three state gates and multiplexers is the corresponding multiplier bit. If this bit is zero then the three state gates goes into high impedance state and thus inputs are not given to the full adder. The previous sum is only taken as the present sum. If this bit is one then the three state gates gets enabled and the inputs are given to the full adder. Thus the sum is generated and this is taken as the present sum.

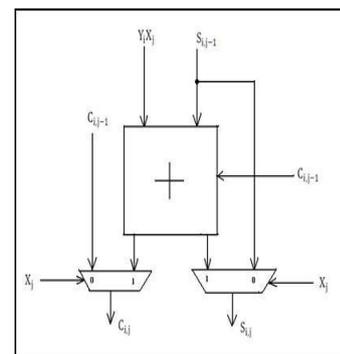


Fig 2.3: Internal structure of adding cell

In this adding cell the three state gate will enabled only when $X_j = 1$ and then the adder will get input. If $X_j = 0$ then the previous sum and carry only will be taken as the present

sum and carry. Thus row bypassing can be done by this adding cell (AC).

In this way the switching activity can be reduced if the multiplicand bit is zero. Thus switching activity in row bypassing multiplier is less than that of Braun multiplier. But the only disadvantage of this row bypassing multiplier is that it needs extra circuitry than Braun multiplier. This limitation can be overcome by the column bypass multiplier.

2.4 4x4 Column bypassing multiplier

Consider the multiplication of 1010 x 1000. Since the multiplicand contains two zeros, the corresponding columns i.e. first and third will get disabled. Now, consider another multiplication of 1111 x 1000. Since multiplicand contains no zero, all columns will get switched.

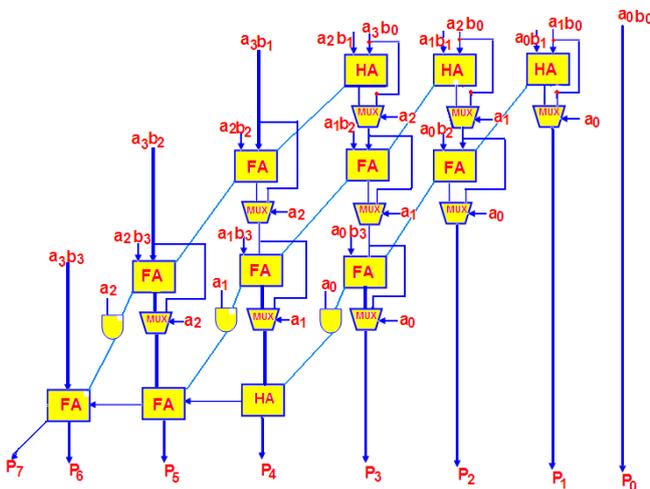


Fig 2.4: 4x4 Schematic Diagram of Column bypassing multiplier

The limitation of this technique is that number of columns switched depends on the number of ones in the multiplicand. For example if the multiplicand is 16 bit in length as 1111111111111111 then all the full adders in all the columns will get switched and consume more power. Less switching activity of the components can be achieved if the multiplicand contains more zeros than ones

III. IMPLEMENTING OF DIFFERENT TYPES OF MULTIPLIERS USING CONVENTIONAL METHOD

3.1 4x4 Braun multiplier

The 4x4 braun multiplier is designed in conventional method by using 456 transistors and power consumption of 4x4 braun multiplier in conventional method is 10.8mw.

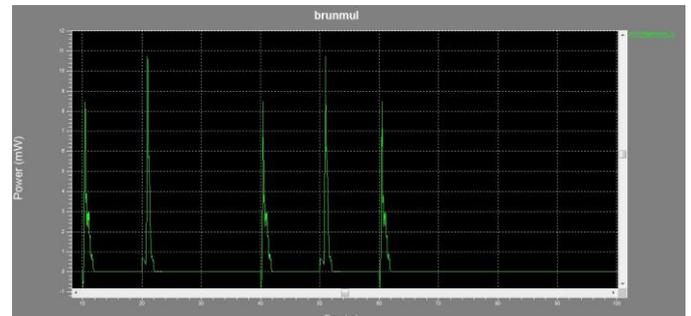


Fig 3.1: power consumption waveform of Braun multiplier in conventional method

3.2 4x4 Row bypassing multiplier

The 4x4 row bypassing multiplier is designed in conventional method by using 756 transistors and power consumption of 4x4 row bypassing multiplier in conventional method is 17.5mw.

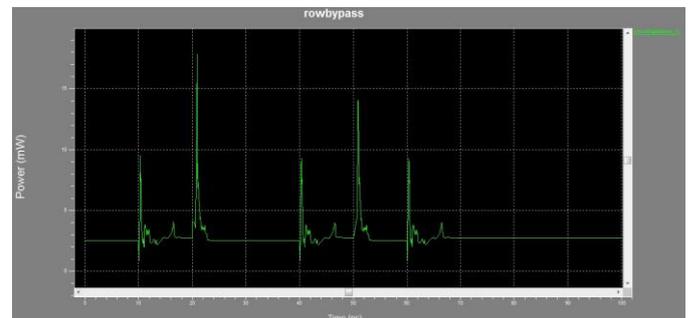


Fig 3.2: power consumption waveform of row bypassing multiplier in conventional method

3.3 4x4 Column bypassing multiplier

The 4x4 Column bypassing multiplier is designed in conventional method by using 582 transistors and power consumption of 4x4 Column bypassing multiplier in conventional method is 16mw.

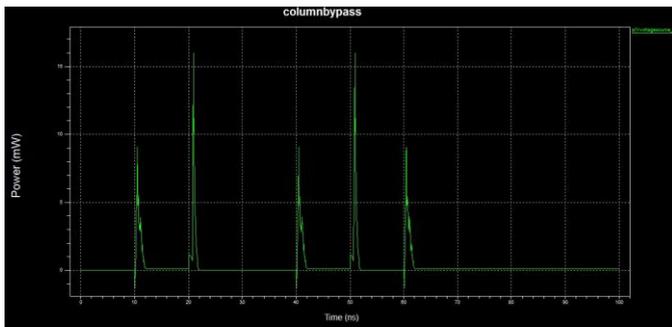


Fig 3.3: power consumption waveform of column bypassing multiplier in conventional method

IV. THE GDI TECHNIQUE

Gate-Diffusion-Input (GDI) method is based on the use of a simple cell as shown in figure .2. At a first glance the basic cell reminds the standard CMOS inverter, but there are some important differences:

- 1) GDI cell contains three inputs – G (common gate input of NMOS and PMOS), P (input to the source/drain of PMOS), and N (input to the source/drain of NMOS).
- 2) Bulks of both NMOS and PMOS are connected to N or P respectively, so it can be arbitrarily biased at contrast with CMOS inverter. It must be remarked, that not all the functions are possible in standard P-Well CMOS process, but can be successfully implemented in Twin-Well CMOS or SOI technologies .

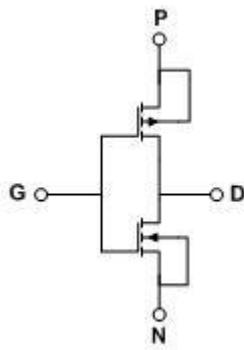


Fig 4.1: GDI basic cell

Table I shows how a simple change of the input configuration of the simple GDI cell corresponds to very different Boolean functions. Most of these functions are complex (6-12 transistors) in CMOS, as well as in standard

PTL implementations, but very simple (only 2 transistors per function) in GDI design method.

Table I: Some logic functions that can be implemented with a single GDI cell

N	P	G	D
0	B	A	AB
B	1	A	A+B
B	0	A	AB
C	B	A	AB+AC
0	1	A	A'

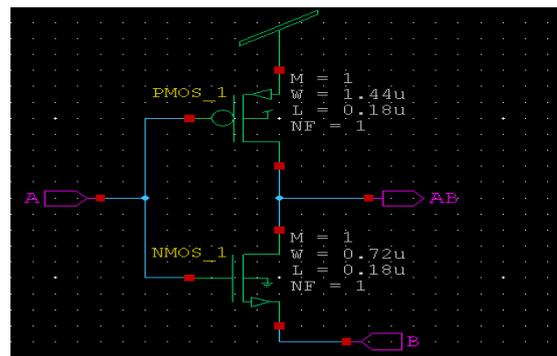


Fig 4.2: transistor level implementation of and gate using GDI technique

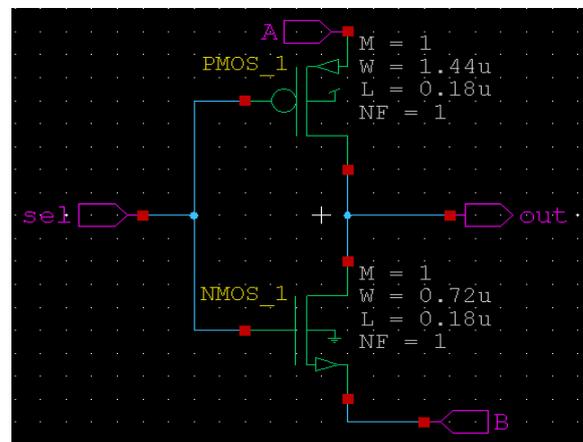


Fig 4.3: transistor level implementation of MUX using GDI technique

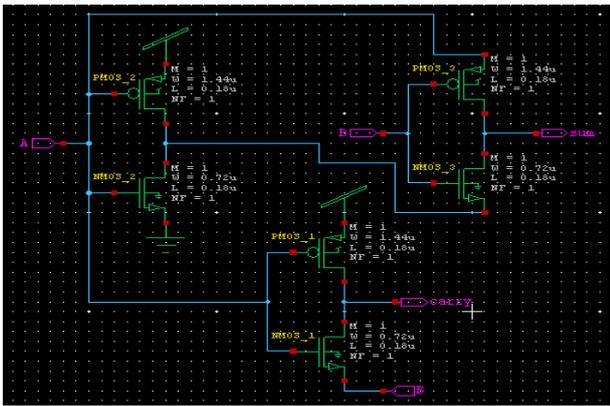


Fig 4.4: transistor level implementation of half adder using GDI technique

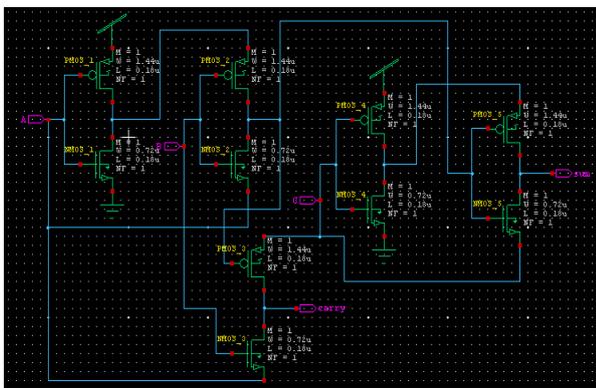


Fig 4.5: transistor level implementation of full adder using GDI technique

V. IMPLEMENTING OF DIFFERENT TYPES OF MULTIPLIERS USING GDI TECHNIQUE

5.1 4x4Braun multiplier

The 4x4 braun multiplier is designed in GDI technique by using 136 transistors and power consumption of 4x4 braun multiplier in GDI technique is 3mw.

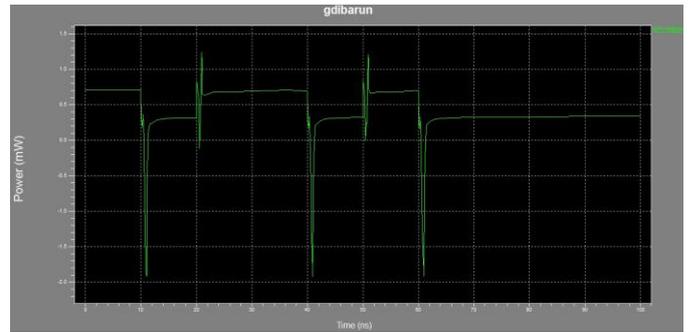


Fig 5.1: power consumption waveform of braun multiplier in GDI method

5.2 4x4 Row bypassing multiplier

The 4x4 row bypassing multiplier is designed in GDI technique by using 196 transistors and power consumption of 4x4 row bypassing multiplier in GDI technique is 3.7mw.



Fig 5.2: power consumption waveform of row bypass multiplier in GDI method

5.3 4x4 Column bypassing multiplier

The 4x4 Column bypassing multiplier is designed in GDI technique by using 160 transistors and power consumption of 4x4 Column bypassing multiplier in GDI technique is 3.4mw.

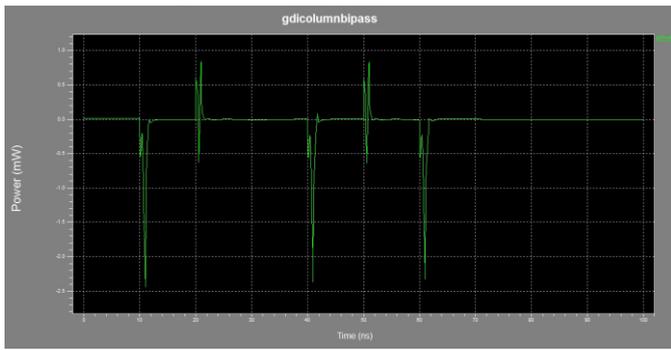


Fig 5.3: power consumption waveform of column bypass multiplier in GDI method

VI. COMPARISON RESULTS BETWEEN CONVENTIONAL METHOD AND GDI TECHNIQUE

6.1 area comparison between conventional method and GDI technique

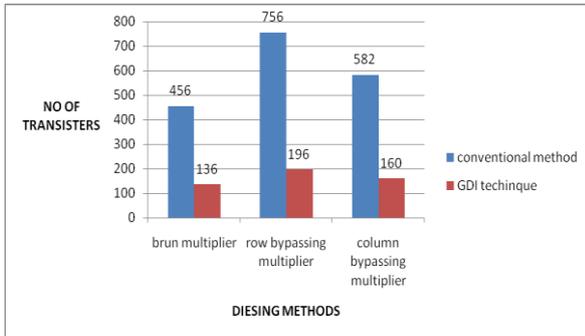


Fig 5.3: area comparison between conventional method and GDI technique

6.2 comparison of power consumption between conventional method and GDI technique

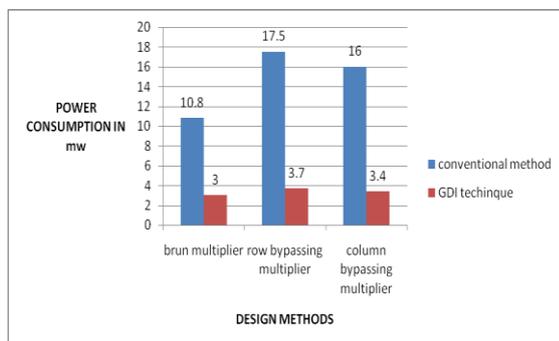


Fig 5.3: area comparison between conventional method and GDI technique

VII. CONCLUSION

Thus the hardware implementation of Braun, Row, column multiplier and 2-D multiplier gave the difference in time delays for each computation is compared in the previous slide through graphical representation.

By combining both row bypassing and column bypassing multipliers we have produced a 2-D bypassing multiplier with less delay and power consumption.

Based on the simplification of the MAC operations and by using low-power bypassing technique, we have designed a two dimensional multiplier which has less computations and less switching activities with low power consumption.

REFERENCES

- [1] Oscar T. -C. Chen, Sandy Wang, and Yi-Wen Wu, .Minimization of Switching Activities of Partial Products for Designing Low-Power Multipliers., IEEE Transactions on VLSI Systems, June 2003 vol. 11, no.3.
- [2] HichemBelhadj, BehroozZahiri, Albert Tai .Power-sensitive design techniques on FPGA devices. Proceedings of International conference on IC Taipei (2003).
- [3] S. Hong, S. Kim, M.C. Papaefthymiou, and W.E. Stark, .Low power parallel multiplier design for DSP applications through coefficient optimization. In Proc. of Twelfth Annual IEEE Int. ASIC/SOC onf., Sep. 1999, pp. 286-290.
- [4]I. S. Abu-Khater, A. Bellaouar, and M. Elmasry, .Circuit techniques for CMOS low-power high-performance multipliers., IEEE J. Solid-State Circuits, Oct. 1996, vol. 31, pp. 1535.1546.