Navdeep Goel, Sukhreet Singh / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 2,Mar-Apr 2012, pp.1488-1491

Comparative Analysis of 4-bit Multipliers Using Low Power Adder Cells

Navdeep Goel
Assistant Professor (ECE), Yadavindra College of Engineering, Punjabi University Kashi Campus, Talwandi Sabo (India)

Sukhreet Singh
Lecturer (ECE) Guru Ram Dass Institute of Engineering & Technology Bathinda (India)

Abstract
Multiplier is the most commonly used circuit in the digital devices. Multiplication is one of the basic functions used in digital signal processing. Most high performance DSP systems rely on hardware multiplication to achieve high data throughput. There are various types of multipliers available depending upon the application in which they are used. Full Adder is the main block of power dissipation in multiplier. So reducing the power dissipation of full adder ultimately reduces the power dissipation of multiplier. In this paper 4-bit multipliers based on Gate Diffusion Input (GDI) adder cells are compared using EDA Tanner (Evaluation version), simulations are based on 180nm CMOS technology**.**

Key Words: Multiplier, Gate Diffusion Input, CMOS, Full Adder, Low Power.

I. INTRODUCTION
With rapid development of portable digital applications, the demand for increasing speed, compact implementation, and low power dissipation triggers numerous research efforts [1]–[3]. To improve the performance of logic circuits, once based on traditional CMOS technology, resulted in the development of many logic design techniques during the last two decades, one form of logic that is popular in low-power digital circuits is pass-transistor logic (PTL). Many PTL circuit implementations have been proposed in the literature [1], [2], [3]–[6].
Some of the main advantages of PTL over standard CMOS design are 1) high speed, due to the small node capacitances; 2) low power dissipation, as a result of the reduced number of transistors; and 3) lower interconnection effects [7], [8], due to a small area.
However, most of the PTL implementations have two basic problems. First, the threshold drop across the single-channel pass transistors results in reduced current drive and hence slower operation at reduced supply voltages; this is particularly important for low-power design since it is desirable to operate at the lowest possible voltage level. Second, since the "high" input voltage level at the regenerative inverters is not, the PMOS device in the inverter is not fully turned off, and hence direct-path static power dissipation could be significant [2].
There are many sorts of PTL techniques that intend to solve the problems mentioned above [4].

1) Transmission gate CMOS (TG) uses transmission gate logic to realize complex logic functions using a small number of complementary transistors. It solves the problem of low logic level swing by using PMOS as well as NMOS [1].

2) Complementary pass-transistor logic (CPL) features complementary inputs/outputs using NMOS pass-transistor logic with CMOS output inverters. CPL's most important feature is the small stack height and the internal node low swing, which contribute to lowering the power consumption. The CPL suffers from static power consumption due to the low swing at the gates of the output inverters. To lower the power consumption of CPL circuits, LCPL and SRPL circuit styles are used. Those styles contain PMOS restoration transistors or cross-coupled inverters (respectively).

3) Double pass-transistor logic (DPL) uses complementary transistors to keep full swing operation and reduce the dc power consumption. This eliminates the need for restoration circuitry. One disadvantage of DPL is the large area used due to the presence of PMOS transistors. An additional problem of existing PTL is top-down logic design complexity, which prevents the pass transistors from capturing a major role in real logic LSIs [5]. One of the main reasons for this is that no simple and universal cell library is available for PTL-based design.
After that a gate diffusion input (GDI) technique [9] was presented which solves most of the problems discussed above. The GDI approach allows implementation of a wide range of complex logic functions using only two transistors. This method is suitable for design of fast, low-power circuits, using a reduced number of transistors (as compared to CMOS and existing PTL techniques), while improving logic level swing and static power characteristics and allowing simple top-down design by using small cell library.

This paper is organized as follows: basic GDI functions and their circuit principles in Section II. In section III some preliminaries and full adders based on GDI-XOR and GDI-XNOR gates are described [10]. In section IV, array and tree multiplication algorithms are discussed. In section V characteristics of multipliers are discussed. In section VI simulation results are compared. Some conclusion and references are finally drawn in VII and VIII respectively.

II. BASIC GDI FUNCTIONS

The GDI method is based on the use of a simple cell as shown in Fig. 1. At first glance, the basic cell reminds one of the standard CMOS inverter, but there are some important differences.

1) The GDI cell contains three inputs: (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 a CMOS inverter.

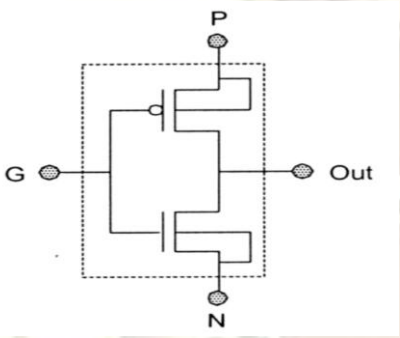

Fig 1: GDI basic cell. [9]

It must be remarked that not all of the functions are possible in standard p-well CMOS process but can be successfully implemented in twin-well CMOS or silicon on insulator (SOI) technologies.

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 two transistors per function) in the GDI design method.

In this paper, most of the designed circuits were based on the F1 and F2 functions. The reasons for this are as follows:

1) Both F1 and F2 are complete logic families (allows realization of any possible two-input logic function).

2) F1 is the only GDI function that can be realized in a standard p-well CMOS process, because the bulk of any NMOS is constantly and equally biased.

3) When N input is driven at high logic level and P input is at low logic level, the diodes between NMOS and PMOS bulks to out are directly polarized and there is a short between N and P, resulting in static power dissipation.

Table I Various Logic Functions Of GDI Cell for Different Input Configurations

N	P	G	Out	Function
0	B	A	$\overline{A}$ B	F1
B	1	A	$\overline{A}$ +B	F2
1	B	A	A+B	OR
B	0	A	AB	AND
C	B	A	$\overline{A}$ B+AC	MUX
0	1	A	$\overline{A}$	NOT

III. PRELIMINARIES

A. Logic Equations for the Proposed Full Adder

The full adder operation equations presented below can be stated as follows: given the three 1-bit inputs A, B and C_{in} which calculate two 1-bit outputs Sum, for sum and C_{out}, for carry out.

$$Sum = A \oplus B \oplus C_{in} \quad \text{.....(1)}$$

$$Sum = A \odot B \odot C_{in} \quad \text{.....(2)}$$

$$Sum = \bar{C}_{in}(A \oplus B) + C_{in}(A \odot B) \quad \text{.....(3)}$$

$$C_{out} = A(A \odot B) + \bar{C}_{in}(A \oplus B) \quad \text{.....(4)}$$

B. XOR and XNOR gates based on Gate-Diffusion-Input cell

The XOR and XNOR gates based on GDI cells are applications of the GDI technique. As can be seen in Fig. 2, each of them requires only four transistors. Obviously, the proposed GDI XOR and XNOR gates use less transistors compared with the conventional CMOS counterparts

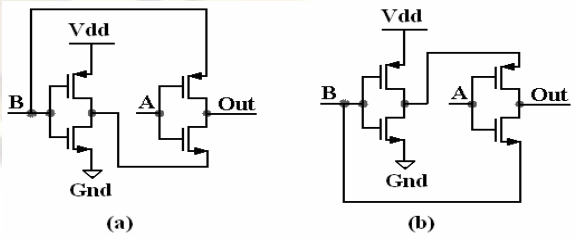

Fig 2: (a) GDI XOR gate (b) GDI XNOR gate [10]

C. Full Adders Based On Gate Diffusion Input XOR and XNOR Gates

According to the logic equations mentioned above and the GDI XOR and XNOR gates in Fig. 2, full adders can be designed in two patterns: GDI XOR full adder and GDI XNOR full adder. Each of the two full adders includes 10 transistors.

1. GDI XOR full adder

The transistor level implementation of GDI XOR full adder is shown in Fig. 3. This full adder consists of three modules—two GDI XOR gates and a multiplexer. The Sum and C_{out} can be calculated using (1) and (4). In the worst case, Sum has 4-T delay while C_{out} has 3-T delay. However, due to the advantages of GDI cell, this circuit still can achieve its benefit of low power consumption.

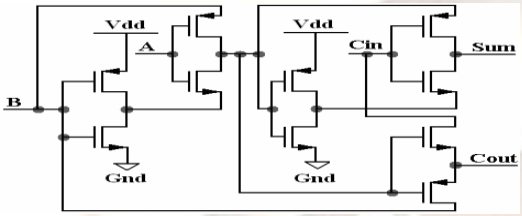

Fig 3: GDI XOR full adder [10]

2. GDI XNOR full adder

Fig. 4 is the GDI XNOR full adder which is another basic architecture of the application of GDI cells. This scheme also includes three modules. It contains two GDI XNOR gates and a multiplexer. In the worst route, Sum has 4-T delay and C_{out} has 3-T delay. The Sum and C_{out} can be calculated from (2) and (4) respectively.

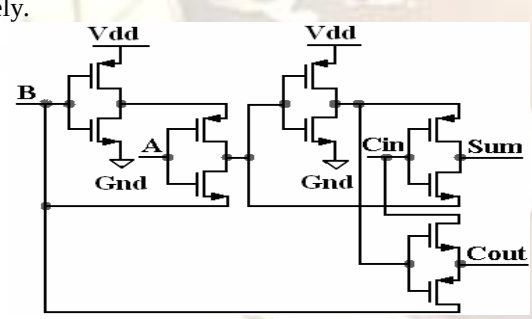

Fig 4: GDI XNOR full adder [10]

IV. MULTIPLICATION ALGORITHMS

At the most basic level, digital multiplication can be seen as a series of bit shifts and bit additions, where two numbers, the multiplier and the multiplicand are combined into the final result. There are numerous multiplier implementations are existing some of them are good in terms of power dissipation and some have better performance in terms of delay. In this section 4-bit array and tree multiplier algorithms are discussed.

1. Array Multiplier

The notable characteristic about the array architecture is its regular structure, shown in figure 5. This has the advantage that it is very easy to lay out, as a single adder block and associated connections are replicated the width and depth of the array. An n × n array of AND gates can compute all the a_ib_i terms simultaneously. The terms are summed by an array of 'n [n - 2]' full adders and 'n' half adders. The delay of this block is a function of the number of rows [11], O(n), which is a big improvement over the simple-minded scheme of using conventional adders for each row

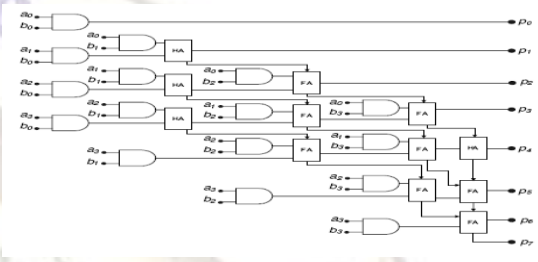

Fig 5: Block diagram of 4×4-bit array multiplier [12]

2. Tree multiplier

In 1964, C.S. Wallace [13] observed that the later stages of the array structure must always wait for all the earlier stages to complete before their final values will be established. When performing a series of independent add operations, it is possible to create a structure which has less delay by performing the addition operations in parallel, where possible. The advantage of Wallace tree is speed because the addition of partial products is O (log N) where N is the number of summands.

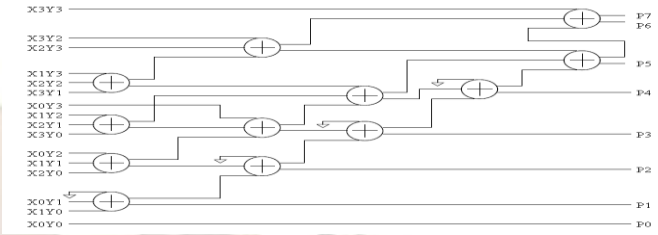

Fig. 6. 4 x 4 Tree Multiplier [14]

V. CHARACTERISTICS OF MULTIPLIERS

There are three main components of power consumption in digital CMOS VLSI circuits.

1. **Switching Power**: consumed in charging and discharging of the circuit capacitances during transistor switching.
2. **Short-Circuit Power**: consumed due to short-circuit current flowing from power supply to ground during transistor switching. This power more dominates in Deep Sub Micron (DSM) technology.
3. **Static Power**: consumed due to static and leakage currents flowing while the circuit is in a stable state. The first two components are referred to as dynamic power, since power is consumed dynamically while the circuit

is changing states. Dynamic power accounts for the majority of the total power consumption in digital CMOS VLSI circuits at micron technology [15], [16].

$P_{avg} = P_{dynamic} + P_{static}$

$= (P_{Switching}+P_{Short\text{-}Circuit})+P_{Leakage} = (\alpha_{0\rightarrow1} \times C_L \times V_{dd}^2 \times f_{clk}) + (I_{sc} \times V_{dd}) + (I_{leakage} \times V_{dd})$ .....(5)

Where α = switching activity

C_L = load capacitance

V_{dd}= supply voltage

f_{clk}= clock frequency

I_{sc}= short circuit current

$I_{leakage}$= leakage current

The first term and second term in (5) collectively represents the dynamic power. Under the circumstance of 180nm the static power loss is far less than its counterpart—dynamic power dissipation. Therefore, in most cases, the total power loss is approximate to dynamic power consumption, which is also considered to be related to the internal node capacitance and the probability of switching.

VI. SIMULATION AND COMPARISON

A. SIMULATION ENVIRONMENT: Array and tree multipliers based on GDI XOR and GDI XNOR adder cells are simulated in EDA Tanner (Evaluation version). All the results are obtained in 180nm CMOS process technology.

B. COMPARISON: 4-bit array and tree multipliers are compared based on the parameters like dynamic power consumption, delay and number of transistors. Comparative analysis of 4-bit multipliers using GDI based adder cells working at 400MHz is done with 4-bit CMOS multiplier [17] as shown in the table II. Please pay attention to the shaded area which indicates minimum value in each column.

Table II. Comparative analysis of 4-bit multipliers

Multiplier Name	Dynamic Power	Delay	Number of Transistors
GDI XOR BASED TREE	2.43×10^{-4}W	2.18ns	296
GDI XOR BASED ARRAY	2.52×10^{-4}W	2.19ns	296
GDI XNOR BASED TREE	7.868×10^{-5}W	1.39ns	288
GDI XNOR BASED ARRAY	8.32×10^{-5}W	1.50ns	288
CMOS BASED	2.12×10^{-4}W	1.08ns	320

VII. CONCLUSION

From the comparison table II, it is clear that GDI XNOR based tree multiplier has least power consumption at 400MHz and also it is most effective in terms of number of transistors required.

VIII. REFERENCES

[1] N. Weste and K. Eshraghian, Principles of CMOS digital design, Pearson Education, Addison-Wesley, 2002

[2] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low- power CMOS digital design," IEEE J. Solid-State Circuits, vol. 27, pp. 473–484, 1992.

[3] W. Al-Assadi, A. P. Jayasumana, and Y. K. Malaiya, "Pass-transistor logic design," Int. J. Electron., vol. 70, pp. 739–749, 1991.

[4] I. S. A. Khater, A. Bellaouar, and M. I. Elmastry, "Circuit techniques for CMOS low-power high-performance multipliers," IEEE J. Solid- State Circuits, vol. 31, pp. 1535–1546, 1996.

[5] K. Yano, Y. Sasaki, K. Rikino, and K. Seki, "Top-down pass-transistor logic design," IEEE J. Solid-State Circuits, vol. 31, pp. 792–803, 1996.

[6] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," IEEE J. Solid-State Circuits, vol. 32, pp. 1079–1090, 1997.

[7] T. Sakurai, "Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSI's," IEEE Trans. Electron Devices, vol. 40, pp. 118–124, 1993.

[8] V. Adler and E. G. Friedman, "Delay and power expressions for a CMOS inverter driving a resistive-capacitive load," Analog Integrated Circuits Signal Processing, vol. 14, pp. 29–39, 1997.

[9] A. Morgenshtein, A. Fish and I. A. Wagner, "Gate-Diffusion Input (GDI) – A power-efficient method for digital combinatorial circuits", IEEE Transactions on VLSI systems, vol.10, pp. 566-581, 2002.

[10] D. Wang, M. Yang, W. Cheng, X. Guan, Z. Zhu, Y. Yang, "Novel Low Power Full Adder Cells in 180nm CMOS Technology" Industrial Electronics and Applications, pp 430-433, 2009.

[11] P. C. H. Meier, "Analysis and Design of Low Power Digital Multipliers", Ph.D. Thesis, Carnegie Mellon University, Dept. of Electrical and Computer Engineering, Pittsburgh, Pennsylvania, 1999.

[12] N. Anuar, Y. Takahashi and T. Sekine, "4×4-Bit Array Two Phase Clocked Adiabatic Static CMOS Logic Multiplier with New XOR" 18th IEEE/IFIP International Conference on VLSI and System-on-Chip, pp. 364-368, 2010

[13] Wallace, C.S., "A Suggestion for a Fast Multiplier", IEEE Transactions on Electronic Computers, EC-13, pp. 14-17, 1964.

[14] K. Roy and K.Seng, "Low voltage, Low-power VLSI Subsystems", McGraw-Hill Publishers, 2005

[15] S. Devadas and S. Malik, "A survey of optimization techniques targeting low power VLSI circuits," in Proc. 32nd ACM/IEEE Design Automation Conf., pp. 242–247, 1995

[16] A. P. Chandrakasan, S. Sheng and R. W. Bordersen, "Lowpower CMOS digital design," IEEE J. Solid-State Circuits, vol. 27, pp. 473–484, 1992.

[17] N. Ravi, A. Satish, T. J. Prasad and T. S. Rao, "A New Design for Array Multiplier with Trade off in Power and Area", vol. 8, pp. 533-537, 2011.