

Design Of Bypassing – Based Multipliers Using Ultra Low-Power Technique

Om Prakash Kuswaha

Pydah College of Engineering and Technology, Visakhapatnam, India

U.Pradeep Kumar

Assistant Professor, Dept. of ECE,
Pydah College of Engineering and Technology, Visakhapatnam, India.

Abstract

Recently building low-power VLSI systems has emerged as highly in demand because of the fast growing technologies in mobile communication and signal processing. The battery technology does not advance at the same rate as the microelectronics technology. So designers are faced with more constraints: high speed, high throughput, small silicon area, and at the same time, low-power consumption. 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. Hence reducing the power dissipation of multiplier ultimately reduces the power dissipation of the whole system. This paper uses a new CMOS logic style called GATE DIFFUSION INPUT is used to address the issue of dynamic power dissipation which enables us to design complex functions with fewer gates thereby, reducing the power consumption of the circuit. It also reduces the latency of the circuit.

I. INTRODUCTION

Multiplication is an essential arithmetic operation in DSP applications. For the multiplication of two unsigned n -bit numbers, the multiplicand, $A = a_{n-1} a_{n-2}, \dots, a_0$, and the multiplier, $B = b_{n-1} b_{n-2}, \dots, b_0$, the product, $P = P_{2n-1} P_{2n-2}, \dots, P_0$, can be represented as the following equation:

$$P = P_{2n-1} P_{2n-2} \dots P_0 = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (a_i b_j) 2^{i+j}$$

To achieve the high-performance demand in DSP applications, the structure of a parallel array multiplier is widely used and the typical implementation of such an array multiplier is Braun design. In a $n \times n$ Braun multiplier, the multiplier array consists of $(n-1)$ rows of carry-save adders (CSAs) and a $(n-1)$ -bit ripple-carry adder in the last row, in which each row contains $(n-1)$ full adders (FAs). The $(n-1)$ FAs in the first CSA row that have only two valid inputs can be replaced by $(n-1)$ half adders (HAs).

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. Flexibility is another critical requirement that mandates the use of programmable components like FPGAs in such devices. 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 PARALLEL MULTIPLIERS

This chapter reveals the design considerations of High Speed parallel multiplier and explains the development of source code (VHDL code) module wise.

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 by passing multiplier
3. 4x4 Column by passing multiplier

The detailed description of the above modules with respect to relevant schematic diagrams and necessary source code in VHDL has been explained. Here in High speed parallel multiplier all the codes are developed by using VHDL in behavioral and structural styles. All sub module codes are developed in data flow style.

For high speed parallel multiplier designs, precautions need to be taken at each abstraction level, from system level to technology process level. Higher in the abstraction level an appropriate decision is taken to increase speed, the higher the impact will be. In general design practices to reduce switching activity reduction can be controlled at various levels of the design flow.

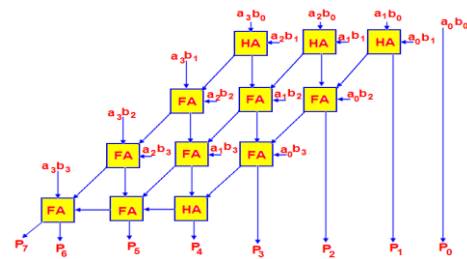
A. BRAUN Multiplier

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

A general Braun parallel multiplier operates by computing the partial products in parallel and by shifting and accumulating the partial products. Switching activity is poorly correlated with the input coefficient. In particular, reducing the switching activity of the component used in the design can minimize the power dissipation.

For Braun multiplier, precautions need to be taken at each abstraction level, from system level to technology process level. Higher in the abstraction level an appropriate decision is taken to reduce power, the higher the impact will be. In general design practices to reduce switching activity reduction can be controlled at various levels of the design flow.

For frequency signals, delay balancing and reduction of the number of logic levels are among the most efficient techniques to tackle power penalty. An obvious method to reduce the switching activity is to shut down the idle part of the circuit, which is not in operating condition. Further low power adder structure reduces the switching activity.



Schematic diagram of 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.

B. Row Bypassing Multiplier

The Row bypassing multiplier reduces the switching activity by bypassing the row in which the multiplier 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 multiplicand 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.

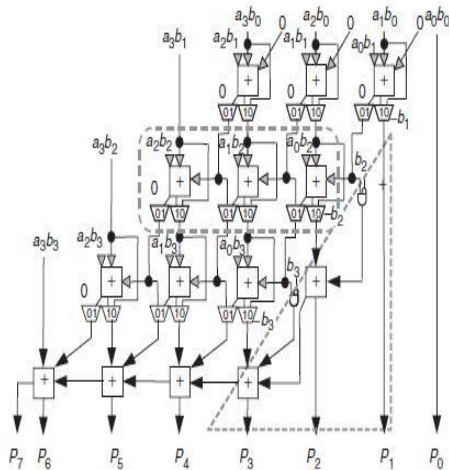
Here a special circuitry called adding cell is used instead of full adders. It consists of three state gates, full adder and multiplexers. The inputs i.e. the partial products to be summed up are given to the full adder through three state gates. The enable input to the three state gates and multiplexers 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.

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.

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

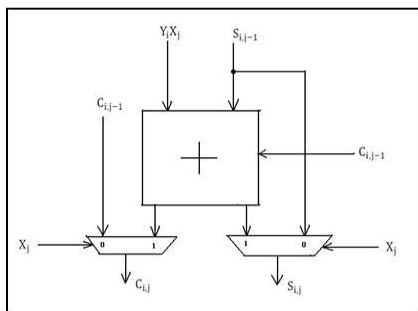
disabled and previous sum is taken as the present sum.

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.



Schematic diagram of Row by passing multiplier

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.



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.

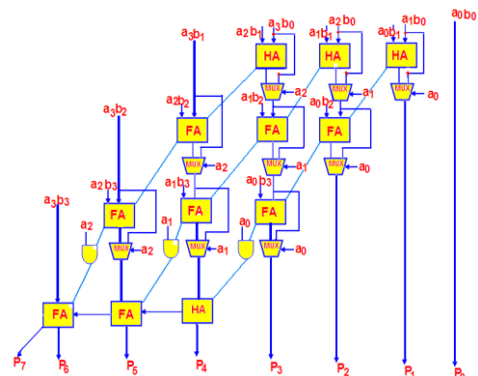
C. Column Bypassing Multiplier

The Column bypassing structure can stop the switching activity even if the bit coefficient is zero that ultimately reduces the power dissipation. This technique reduces the switching to fairly good extent. 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. 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. Higher power reduction can be achieved if the multiplicand contains more number of 0's than 1's

The Column parallel multiplier operates by computing the partial products in parallel and by shifting and accumulating the partial products. Switching activity is poorly correlated with the input coefficient. In particular, reducing the switching activity of the component used in the design can minimize the power dissipation.

For Column multiplier, precautions need to be taken at each abstraction level, from system level to technology process level. Higher in the abstraction level an appropriate decision is taken to reduce power, the higher the impact will be. In general design practices to reduce switching activity reduction can be controlled at various levels of the design flow.

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.



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. DESIGN OF HALF ADDER, FULL ADDER AND MULTIPLIER USING 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 im-portant differences: 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 suc-cessfully implemented in Twin-Well CMOS.

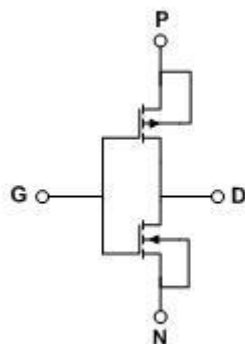


Figure 6: GDI Basic Cell of GDI Technique

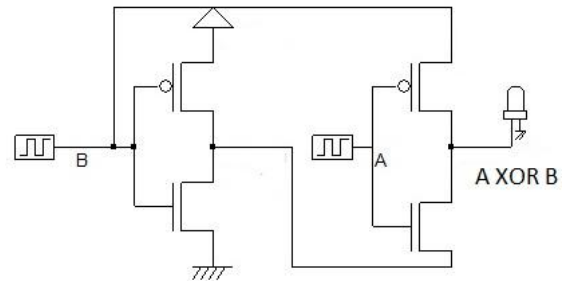
Table I shows how a simple change of the input configuration of the simple GDI cell corresponds to very different Boolean functions. Most of these functions are complex (6-12 transis-tors) in CMOS, as well as in standard PTL implementations, but very simple (only 2 transistors per function) in GDI design method.

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

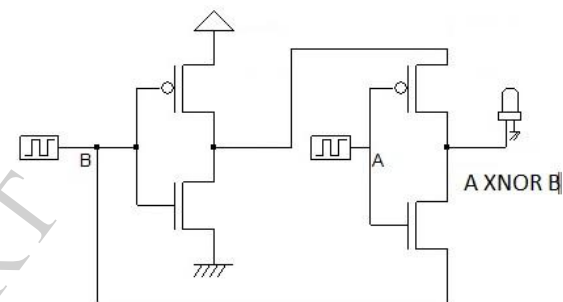
N	P	G	D
'0'	B	A	A`B
B	'1'	A	A`+B
'1'	B	A	A+B
B	'0'	A	AB
C	B	A	A`B+AC
'0'	'1'	A	A`

XOR and XNOR functions are the key variables in adder equations. If the generation of

them is optimized, this could greatly enhance the performance of the full adder cell. In this new cell, we have used the GDI technique for generating of XOR and XNOR functions. It uses only six transistors sepa-rately to generate the basic XOR and XNOR functions, as shown in figure below.



Basic XOR circuit with GDI technique



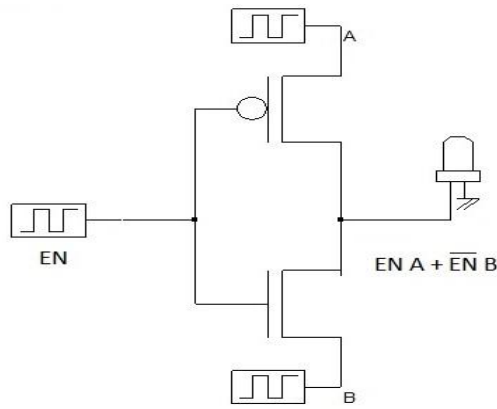
Basic XNOR circuit with GDI technique

A one-bit binary full adder takes three one-bit inputs: A, B and Cin and generate sum and carry.

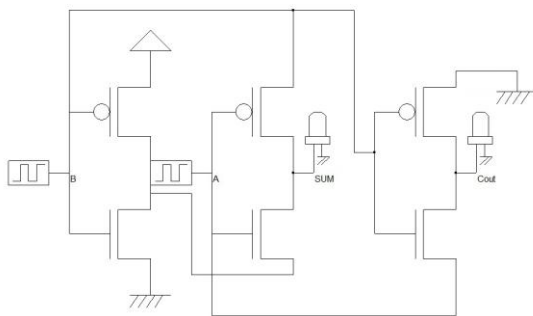
$$S = A \oplus (B \oplus C_{in})$$

$$C_{out} = A \cdot B + C_{in} (A \oplus B)$$

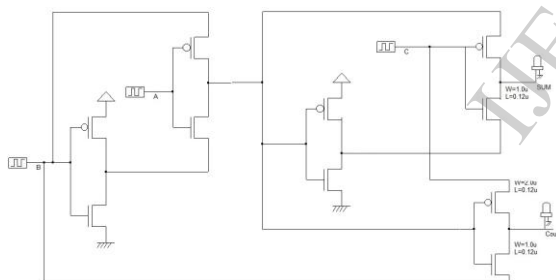
The goal of this paper is to design a high speed and low power full adder cell with the GDI technique. The full adder cell has the 12 transistors that is shown in figure [8] In case of this cell, the GDI technique is used for generating of XOR functions. This stage shows full swing with low vol-tage. The output of XOR, together with other inputs, will be fed to the other circuit, which has to design based on Gate-Diffition Input (GDI) technique. The Sum and Carry outputs are generated from the final stage.



Basic Multiplexer circuit with GDI technique



Basic half adder circuit with GDI technique



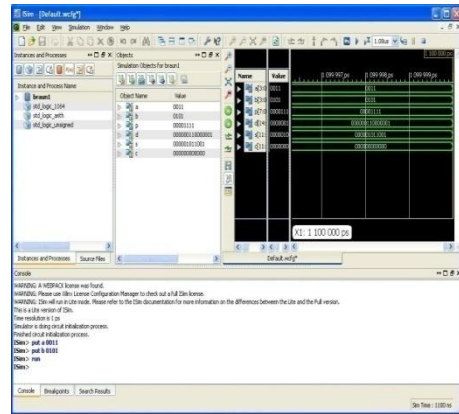
Basic full adder circuit with GDI technique

A. Simulation of Braun Multiplier using GDI Technique

The below Fig shows the simulation results of 4x4 Braun multiplier. Various signals involved in the simulation of 4x4 Braun multiplier are

Input signals : a[3:0] , b[3:0]

Output signals : p[7:0]



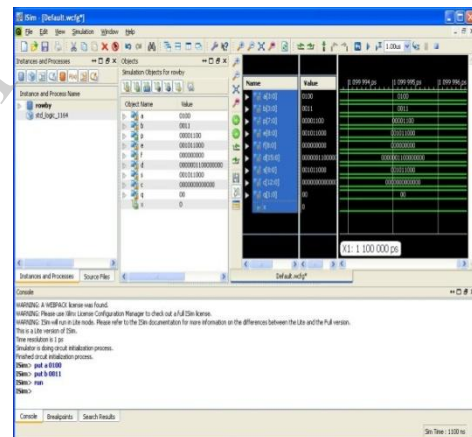
Simulation results of 4x4 Braun multiplier using GDI Technique

B. Simulation of Row by passing Multiplier using GDI Technique

The below Fig 4.2 shows the simulation results of 4x4 Row by passing multiplier. Various signals involved in the simulation of 4x4 Row by passing multiplier are

Input signals : a[3:0] , b[3:0]

Output signals : p[7:0]



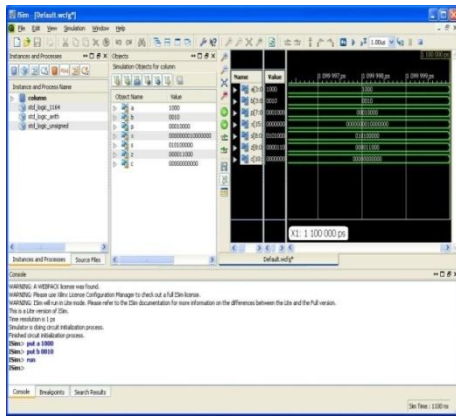
Simulation results of 4x4 Row by passing multiplier using GDI Technique

C. Simulation of column bypassing multiplier using GDI Technique

The above Fig 4.3 shows the simulation results of 4x4 Column multiplier. Various signals involved in the simulation of 4x4 Column multiplier are

Input signals : a[3:0] , b[3:0]

Output signals : p[7:0]



Simulation results of 4x4 column by passing multiplier using GDI Technique

Comparison of Time delays:

By comparing the time delays obtained in synthesis reports we can decide the high speed multiplier. The below table gives the comparison results.

Multiplier (4x4)	Number of LUTs	Maximum Combinational Path Delay(ns)
Braun Multiplier[1]	26	17.723
Row by passing Multiplier[9]	30	17.784
Column by passing Multiplier[10]	27	15.956

Table :Comparison of time delays

From the table it is understood that the time delay for the Column bypassing multiplier is less than any other multiplier.

Comparison of number of transistors used

By comparing the number of transistors used in the different designs we can decide the best possible multiplier which has the highest speed with lowest number of transistors used. The below table gives the comparison results.

Multiplier Design	No. of transistor required using CMOS techniques	No. of transistor required using GDI technique	Percentage of saving in transistors
Braun multiplier[11]	372	110	70.43%
Row bypassing[9]	616	192	61.75%
Column bypassing[10]	502	172	62.60%

Table Comparison of number of transistors used

From the table it is understood that number of transistors used in the design of the multiplier is least in the case of braun multiplier less than any other multiplier.

IV. CONCLUSIONS

By comparing the time obtained in synthesis reports delays and number of transistors required we can decide the high speed multiplier. The below tables 2 and table 3 gives the comparison results.

From the table it is understood that the time delay for the Column bypassing multiplier is less than any other multiplier and the number of transistors required has reduced. Hence giving a ultra low power high speed bypassing multiplier.

REFERENCES

- [1]. Jin- Tan Yan, Zhi- Wei Chen, "Low- Cost Low-Power Bypassing- Based Multiplier Design". IEEE International Symposium on Circuits and Systems (ISCAS), Proceedings of 2010, pp 2338 - 2341
- [2]. B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, 2000.
- [3]. T. Nishitani, "Micro-programmable DSP chip," *14th Workshop on Circuits and Systems*, pp.279-280, 2001.
- [4]. V. G. Moshnyaga and K. Tamaru, "A comparative study of switching activity reduction techniques for design of lowpower multipliers," *IEEE International Symposium on Circuits and Systems*, pp.1560-1563, 1995.
- [5]. A. Wu, "High performance adder cell for low power pipelined multiplier," *IEEE International Symposium on Circuits and Systems*, pp.57-60, 1996.
- [6]. Shahid Jaman, Nahian Chowdhury, Aasim Ullah, Muhammad Foyazur Rahman, "A New High Speed - Low Power 12 Transistor Full Adder Design with GDI Technique", *International Journal of Scientific &*

Engineering Research Volume 3, Issue 7, July-2012

- [7]. Arkadiy Morgenshtein, Alexander Fish, and Israel A. Wagner, "Gate-Diffusion Input (GDI): A Power-Efficient Method for Digital Combinatorial Circuits", *IEEE transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 5, october 2002.
- [8]. N. Weste and K. Eshraghian, *Principles of CMOS digital design*. Reading, MA: Addison-Wesley, pp. 304–307.
- [9]. J. Ohban, V. G. Moshnyaga, and K. Inoue, "Multiplier energy reduction through bypassing of partial products," *IEEE Asia-Pacific Conference on Circuits and Systems*, pp.13–17, 2002.
- [10]. M. C. Wen, S. J. Wang and Y. M. Lin, "Low power parallel multiplier with column bypassing," *IEEE International Symposium on Circuits and Systems*, pp.1638-1641, 2005.

IJERT