16-Bit GDI Multiplier Design for Low Power Applications

B. N. Manjunatha Reddy[1*], S. Shanthala[2], B. R. Vijaya Kumar[3]
[1]Research Scholar, Dept. of TCE, Bangalore Institute of Technology, Bengaluru, Karnataka, India.
Email: manjunatha_reddy@gat.ac.in
[2]Professor and Head, Dept. of TCE, Bangalore Institute of Technology, Bengaluru, Karnataka, India.
Email: Shanthala_wg@yahoo.com
[3]Professor and Dean, Global Academy of Technology, Bengaluru, Karnataka, India.
Email: vbelaguli@gat.ac.in
*Corresponding Author

Abstract: Today, more and more, high speed mobile computational devices and equipments are being introduced in the market. These computational devices strain and drain the battery very quickly. Researchers are making efforts to find ways and means to conserve the battery power for longer period. The core key components in these computational devices are the Multipliers to support high speed computational intensive applications in real time. Thus it becomes more important to reduce power dissipation and area in these multiplier modules as they affect the performance of the device. Several VLSI design techniques have been attempted to optimize the power and area occupied by the multiplier module, but there are very few design techniques that gives the required extensibility both in terms of power and area. In this paper a high speed, reliable and efficient 16 bit multiplier VLSI module design is presented using GDI (Gate Diffusion Input) technique, addressing both power consumption and area complexity. Further, comparative study results of the proposed design over the traditional CMOS design are also presented. Detailed design steps and comparative study using Cadence Virtuoso simulation tool at 180nm CMOS technology is discussed. The simulation results presented show reduction in both power and area of the proposed design at 1.8 V supply voltage.

Keywords: Booth encoder, GDI, Partial product generator, Wallace tree adder.

I. Introduction

Increase in demand for high speed data processing inspired researchers to seek faster processors. Arithmetic circuits like adders and multipliers, which are fast, reliable and efficient, are the essential components in the design of many digital systems. The total power dissipated by the circuit depends on Power dissipation in multiplier and hence it affects the performance of the device. High speed multipliers contain three sections: first section is partial product generation and a common approach for performing it is using Booth Encoding approach. In the Second section, partial product reduction schemes, such as Wallace tree is used for adding partial products vertically, until rows that were generated in part one decrease to two rows. Finally, the third section is a final adder that adds two rows generated in section two, where it uses an advanced adder approach viz., carry-select adder. This paper focuses on the design and simulation of high-speed 16-bit Multiplier, for low-power applications. The existing systems use the CMOS Technology where transistor count & Power consumption are high. A low power digital combinational circuit is described by using a new technique viz., Gate Diffusion Input (GDI) logic. This technique reduces the area and hence power consumption of digital circuits while maintaining low complexity of the design. A wide range of complex logic functions are implemented in GDI by using only two transistors.

II. Architecture of Multiplier

The 16-bit GDI multiplier architecture has the main modules like 2's Complement Generator, partial product generator, Booth encoder Wallace tree adder and final adders respectively. Booth encoder and Wallace tree adders are used to reduce area. Also, Wallace Tree Adder is used to accumulate partial products and Booth encoder is used for partial products generation. Fig. 1 shows the architecture of a multiplier [1].

A. Booth Encoder

Multiplication of negative/signed numbers is complex to implement. The method used for unsigned numbers cannot be applied for negative numbers as it gives wrong result. So, a different approach viz., Booth encoding is used for multiplication of signed and unsigned numbers Multiplier (MR) and Multiplicand (MD). Booth's multiplication algorithm multiplies two signed binary numbers in two's compliment notation. In Booth encoding technique,

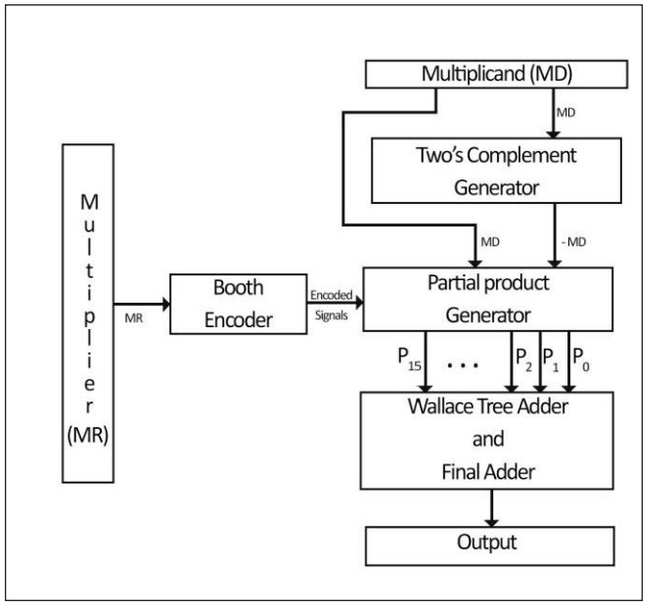

Fig. 1: Architecture of Multiplier

- If i^{th} bit b_i is 0 and $(i-1)^{th}$ bit $b_{(i-1)}$ is 1, then encoded MR's i^{th}bit is +1.
- If i^{th} bit b_i is 1 and $(i-1)^{th}$ bit $b_{(i-1)}$ is 0, then encoded MR's i^{th} bit is -1.
- If i^{th} bit b_i is 0 and $(i-1)^{th}$ bit $b_{(i-1)}$ is 0, then encoded MR's i^{th} bit is 0.
- If i^{th} bit b_i is 1 and $(i-1)^{th}$ bit $b_{(i-1)}$ is 1, then encoded MR's i^{th} bit is 0.

B. Partial Products Generator

In Array Multiplier, the Partial Products (PPs) are generated by performing logical ANDing with each bit of multiplicand with each bit of multiplier. In current design, the following partial products are generated depending on the Booth Encoder output control signals:

1. –MD expressed as 16 bits +15 MSB/Sign Extension bits …Total 15 bits (1).
2. MD expressed as 16 bits +15 MSB/Sign Extension bits …Total 15 bits (2).
3. '0' expressed as 16 bits +15 MSB/Sign Extension bits … Total 15 bits (3).

For example, if LSB bits of x and z signals i.e. x0 and z0 are both '1', then set of partial products (PPs) p0 generated are as indicated in (1). If x0 and z0 are '0' and '1' respectively, PPs p0 generated are as in (2) and if x0 and z0 are both '0', PPs generated are p0 as shown in (3). Likewise, all 15 other control signal pairs (x1, z1; x2, z2; …; x15, z15) are compared to get a total of 16 sets (each of 16 bits) of PPs (p0, p1, ……, p15) yielding 128 (16x16) partial products. Each set of PPs are sign extended by 15 bits to make them 31 bits each thus resulting in 120 (15x8) PPs.

C. Two's Complement Generator

The partial product generator requires 2's complement of the Multiplicand (MD). This is required because –MD is a set of partial products when the Booth Encoder output is "i" i.e. when its control signals x and z are both '1'. The block diagram of 2's Complement Generator is as shown in Fig. 2.

Each bit of MD is applied to an Inverter to get its 1's complement. Logic '1' is added to the LSB bit by using half adder. The carry generated is added with the next bit in another half adder and this is repeated for all the 16 bits of MD. Thus, 2's complement circuit contains16 Inverter and 16half adders. The expression is –MD = 1's complement (MD) $\oplus$ '1'. The output of this Module is complement of MD i.e. –MD.

D. Wallace Tree Adder

In high speed designs, Wallace Tree Adder is used in order to generate two rows of partial products which are added in the last row. Buffers, full adders and half adders are used in Wallace Tree adders to reduce partial products. Partial Products generated from the Booth Encoder are summing using Wallace tree adder. The partial products are added together by using full adders and half adders. Finally the output of the Wallace tree adder is 32 bits of sum and carry.

E. Final Adder

This stage is also important for any multiplier because addition of operands is performed for fast carry propagation. Here the Wallace Tree Adder generates the 32 bits of sum and carry. Finally sum and carry get summed up with a final adder. The final adder consists of Full Adders and buffers. The output of the final adder is 32 bits multiplied output.

III. Implementation of 16-BIT Multiplier

The 16-bit Multiplier is designed using GDI Logic as shown in Fig. 3. It is implemented using Cadence virtuoso tool at 180nm CMOS technology at 1.8v supply voltage. First the basic gates like AND, EXOR, OR functions are designed in GDI logic. Using these gates, sub modules like full adders, half adders are implemented. Finally, main modules like Booth Encoder, Partial Product Generator, 2's complement generator, Wallace Tree Adder are carried out in GDI. The 16-bit Multiplier is designed by combining the sub modules with the above main modulesand finally added with the Final adder. The layout of 16-bit Multiplier is as shown in Fig. 4.

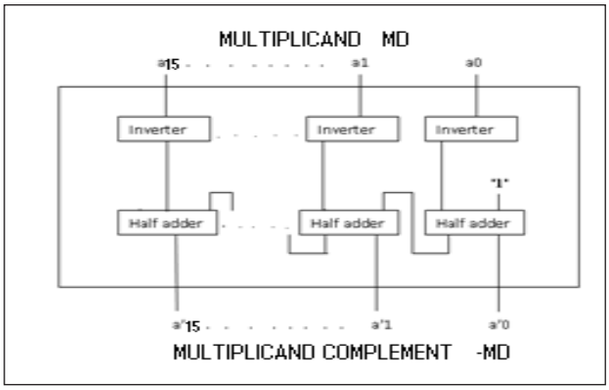

Fig. 2: Block Diagram of Two's Complement Generator

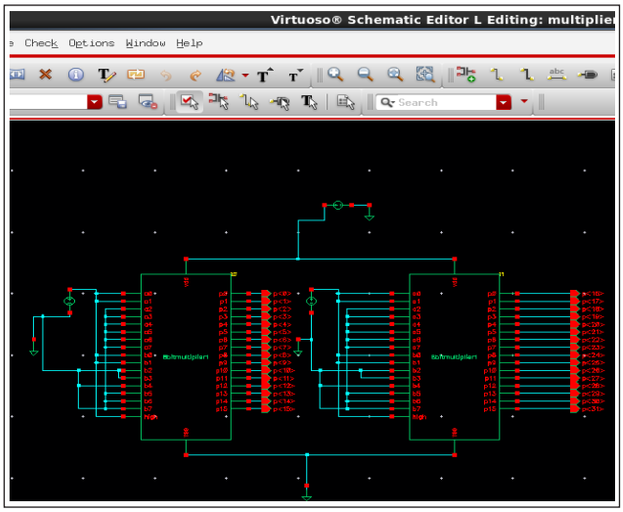

Fig. 3: Schematic of Multiplier

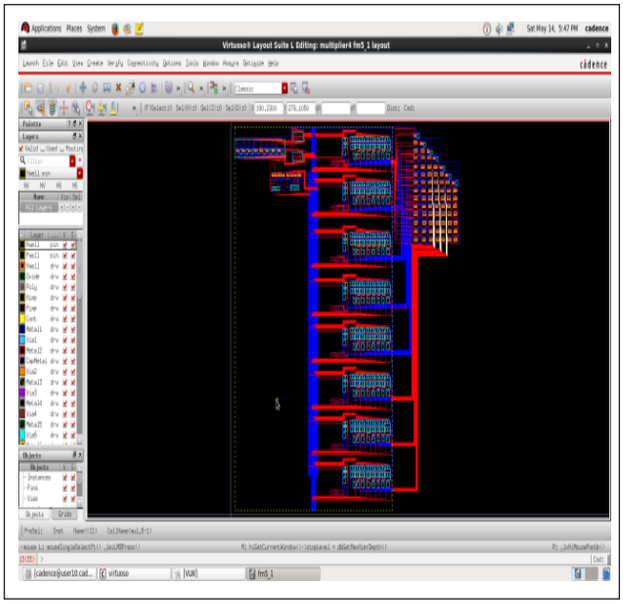

Fig. 4: Layout of Multiplier

IV. Results and Discussions

Here, the operational functionalities of the proposed designs are examined in Radix-2 algorithm. The analysis is carried out in Cadence Virtuoso tool. The performance of the proposed circuit design is carried out in GDI and determined in regard of transistors count and power consumption. The comparison of power and transistors count between the proposed GDI, CMOS and earlier results of GDI based gates, adders and multipliers are shown in Table I. It is seen from the Simulation results that the performance of proposed GDI based circuits are much better than the CMOS based cells [2] and the earlier GDI results [3] with respect to power and area consumption. Simulation output waveform of 16-bit multiplier is as shown in Fig. 5. The comparison of the results obtained in 8-bit and 16-bit multipliers is as shown in Fig. 6. It is observed that the power consumption and the reduction of transistors in designing of 16-bit multiplier have linearly increased with respect to 8 bit multiplier [4] in GDI.

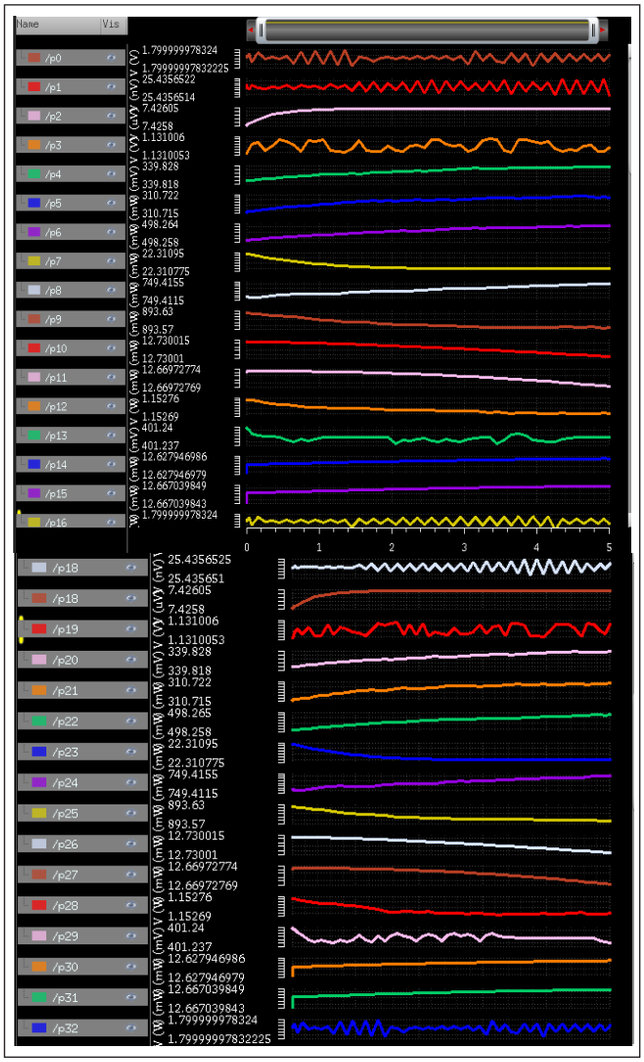

Fig. 5: Output Waveforms of 16 Bit Multiplier

TABLE I: COMPARISON OF NUMBER OF DEVICES IN CMOS, PROPOSED GDI AND THE RESULTS FROM PREVIOUS GDI DESIGN IN RADIX-2 ALGORITHM

Sl. No.	Gates/Modules	Devices in CMOS	Devices in previous designs using GDI [3]	Devices in Proposed GDI Logic
1.	Inverter	2	2	2
2.	2 i/p AND gate	6	5	5
3.	2 i/p XOR gate	12	8	8
4.	2 i/p OR gate	6	4	4
5.	Half Adder	18	13	13
6.	Full Adder	42	32	10
7.	Two's Complement	320	240	240
8.	Booth Encoder	318	238	236
9.	Complete Partial Product generator	9888	8864	3384
10.	Wallace tree adder	14740	11220	6170
11.	16 bit Multiplier	25266	20562	10024

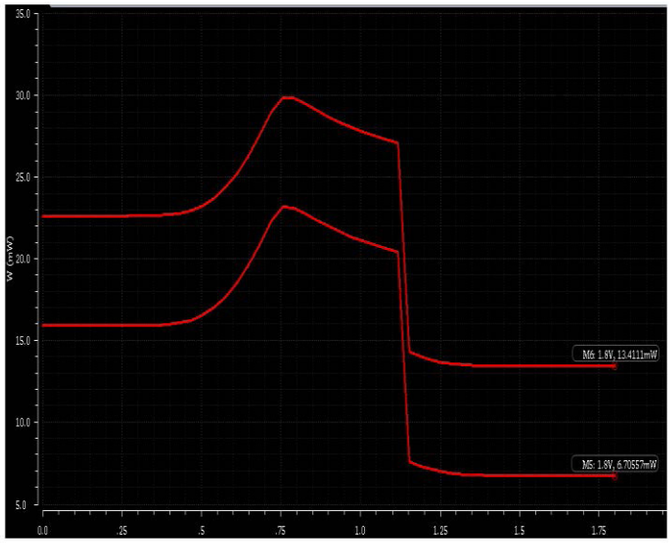

Fig. 6: Comparison of Power in 8-Bit and 16-Bit Multipliers

V. CONCLUSION

The 16-bit Multiplier is designed in GDI by using Booth Encoding and Wallace tree addition techniques in Cadence tool. It gives a clear concept of designing a 16-bit multiplier using a technology called as GDI. The aim of the Multiplier design in GDI logic is to show the reduction of area as compared to CMOS and the previous results of GDI designs. A comparative study is carried out with regard to total number of devices required for the multiplier design in CMOS logic and GDI logic. It is found that the multiplier designed in proposed GDI, results in reduction of devices used and thus minimizing area of multiplier and optimizing the power. It is also observed that the same design can be extended to 32 and 64 bit multiplier designs which are currently being used in the systems.

REFERENCES

[1] M. V. Babu, M. D. Rao, and V. V. K. D. V. Prasad, "A novel architecture of 16-bit multiplier using modified gate diffusion input logic," *International Journal of Computer Science and Information Technology and Security*, vol. 6, no. 4, pp. 69-74, Jul-Aug. 2016.

[2] L. Kalpesh, and B. H. Nagpara, "A 16-bit booth multiplier design using GDI logic style," *International Journal of Advance Research in Engineering, Science and Technology*, vol. 2, no. 5, pp. 1-4, May 2015.

[3] B. K. S. Kumar, H. J. Vidyarani, and S. Pushpalatha, "16-bit low power booth encoded multiplier design using GDI CMOS logic," *International Journal of Electrical, Electronics and Computer Systems*, vol. 3, no. 1, pp. 49-52, 2015.

[4] B. N. M. Reddy, H. N. Sheshagiri, B. R. V. Kumar, and S. Shanthala, "Implementation of low power 8-bit multiplier using gate diffusion input logic," *IEEE 17th International Conference on Computational Science and Engineering*, pp. 1868-1871, 2014.

[5] S. Kaur, B. Singh, and D. K. Jain, "Design and performance analysis of various adders and multipliers using GDI technique," *International Journal of VLSI Design and Communication Systems,* vol. 6, no. 5, pp. 45-56, Oct. 2015.

[6] R. P. Rajput, and M. N. S. Swamy, "High speed modified booth encoder multiplier for signed and unsigned numbers," *IEEE 14th International Conference on Modelling and Simulation*, pp. 649-654, 2012.

[7] S. Abraham, S. Kaur, and S. Singh, "Study of various high speed multipliers," *International Conference on Computer Communication and Informatics (ICCCI-2015),* Coimbatore, India, 08-10 Jan. 2015.

[8] K. Kaur, P. Singh, and G. Joshi, "Analysis of ternary multiplier using booth encoding technique," *2nd International Conference on Signal Processing and Integrated Network (SPIN),* 2015.