Power and Area optimization of 8-bit GDI Multiplier in GPUs

[1]B.N. Manjunatha Reddy, [2]B.R.VijayaKumar, [3]Shanthala S.

[1,2]Dept. of ECE/CSE, Global Academy of Technology, Bengaluru, INDIA
[3]Dept. of TCE, Bangalore Institute of Technology, Bengaluru, INDIA

Abstract: In recent years, high performance computing technology has been developing rapidly. Graphical Processing Units (GPUs) have become an integral part of today's mainstream computing systems. GPU has several specialized floating-point (FP) units to achieve high throughput and better performance. In the current technology, floating-point units have large number of gates and consume more power. Hence, better architectural method is required to implement an effective design. Gate Diffusion Input (GDI) is one such technique of low-power digital circuit design. This technique will reduce power consumption, propagation delay and area of digital circuits. This paper emphasizes on designing 8-bit multiplier using GDI logic for the GPUs to achieve further reduction in power and area of the multiplier [1]. Also Multiplier designed in GDI logic requires lesser number of devices and dissipates low power as compared to CMOS logic [1][7].

Key Words: GPU, GDI, Low Power, Multiplier.

I. INTRODUCTION

As the processing technology continues to scale, more transistors are placed in the GPU chip resulting in higher computational capability with higher power consumption. This provides new opportunities for researchers to explore better power optimization techniques. The GPU architecture was originally introduced for graphics operations and later extended for general-purpose computations.

GPUs provide astonishing performance using hundreds of cores available in it. An overview of the NVIDIA GPU micro architecture is provided in the CUDA Programming Manual [2]. As higher dimensional graphics are becoming more vital in our lives, there is a need for fast graphics processors. The introduction of GPUs allows higher graphics processing speeds [3]. For fast-paced games and other signal processing tasks, the system has to go through the process millions of times per second. To achieve such huge computation speeds, special purpose architectures like GPUs are being used.

In modern GPU Streaming Multiprocessor (SM), the execution units consist of multiplier circuits required to perform FP operations. Because multiplication is the dominating component of many operations,a new architecture can be developed to achievehigh-performance, low-power and area efficiency.

The solution is to build low-power customized multiplier circuit that can provide superior performance in GPU compared to standard CMOS logic circuits.

In this paper, a multiplier circuit is designed using GDI technique which reduces power consumption and area of the GPU multiplier [1].

II. BASIC FUNCTIONS OF GDI LOGIC

The basic cell of GDI logic is shown in Fig.2.1 [4]. This is similar to the standard CMOS inverter with one input, but the GDI cell has three inputs viz., G (common gate input of nMOS and pMOS), P (source of pMOS), and N (source of nMOS). Some of the functions implemented in GDI logic are shown in Table1 [1].

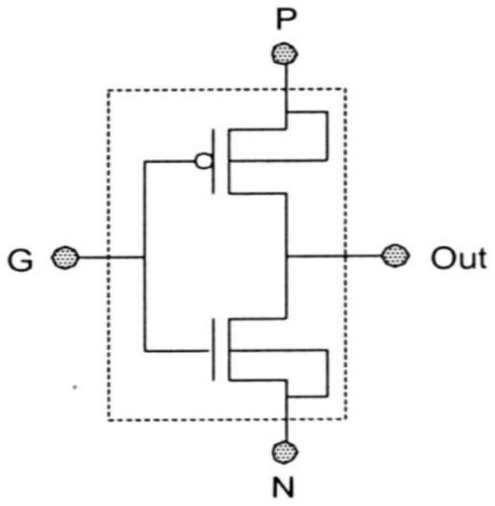

Figure 2.1: GDI basic cell.

Table1: Some functions implemented in GDI

N	P	G	Out
'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'

III. GPU ARCHITECTURE

The NVIDIA GPU card used for graphical and high speed Processing is shown in Figure 3.1. The GPU

Architecture consists of many “Streaming Multiprocessors“ (SM) [5]. Each SM comprises of 32 single-precision CUDA cores, 16 load/store units, 4 Special Function Units (SFUs) and a 64KB block of high speed on-chip memory. Also, each SM contains an on-chip programmer-controlled static memory called *shared memory*, since this memory is shared by all the threads running on the SM. NVIDIA has produced this specialized architecture to achieve exclusive high performance parallel systems that are being currently used.

Figure 3.1: NVIDIA GPU card

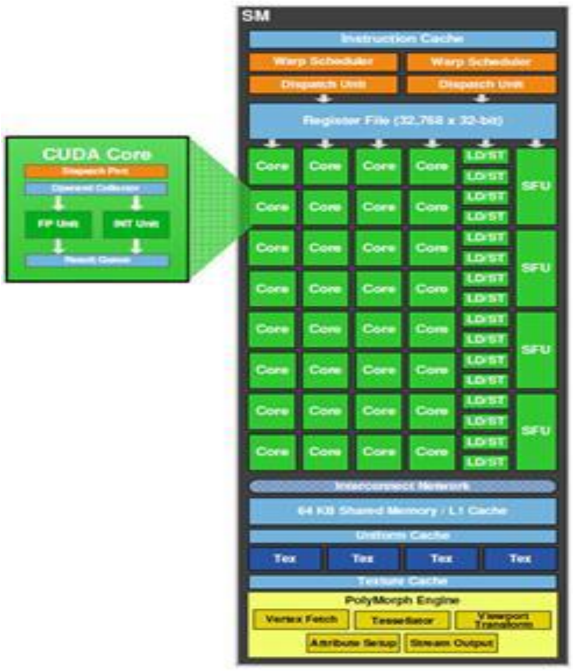

Figure 3.2: NVIDIA GPU Architecture

Fig.3.2 is a single multiprocessor showing hardware available in each SM. These processing units are the standard graphical processing hardware units that perform graphical operations and texturing. In this SM, each individual core has its own integer and floating point units [6]. These units contain multipliers to perform high-speed operations which are designed here using GDI logic.

IV. DESIGN METHODOLOGY FOR GPU 8-BIT MULTIPLIER

The GPU 8-bitMultiplier is implemented using Cadence tool of 180nm technology. The following methodology is employed to achieve the objective using Cadence design suite:

1. Sub module viz., Full Adder is redesigned and implemented using the logic gates which is shown as Schematic, Layout and simulation output in Figures 4.1, 4.2 and 4.3 respectively..

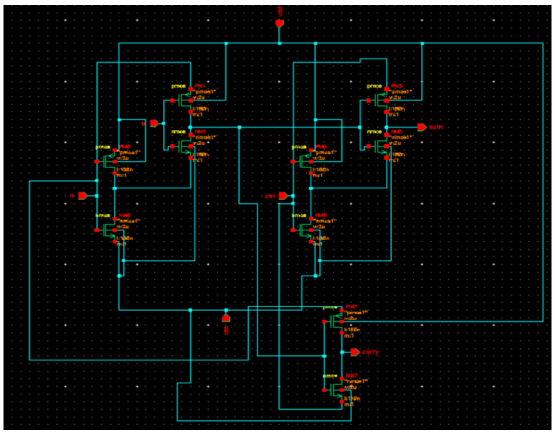

Figure 4.1: Full Adder Schematic

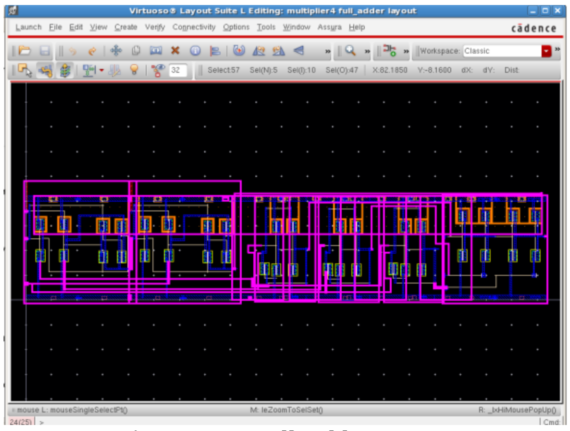
Virtuoso® Layout Suite L Editing: multiplier4 full_adder layout
Launch File Edit View Create Verify Connectivity Options Tools Window Assura Help
cadence

Figure 4.2: Full Adder Layout

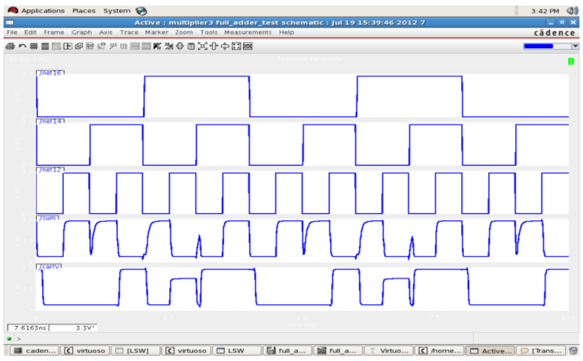

Figure 4.3: Full Adder simulation output

2. The main modules of the Multiplier are designed using sub modules as follows:

i. Booth Encoder - The input to Booth encoder is Multiplier MR and the outputs are x and y control signals of 8 bits each viz., x0, x1, x2, ….., x7 and z0, z1, z2, ….., z7 which are applied to Partial Product Generator.

ii. Partial product generator - The inputs to Partial Product Generator are multiplicand MD, complement of MD (-MD) and x & z control signals. The outputs from this module are 8 sets of partial products each of 8 bits. These partial product sets are MSB sign extended by 7 bits to make them 8 sets of 15 bits each. As such, the output of this module is a total of 120 partial products (15x8).

iii. Wallace tree adder- The input to Wallace tree adder is 120 partial products. The first partial product with positional weight 2^0 is passed through Buffer to get MSB bit of final product and the next two partial products of positional weight 2^1 are added using Half Adder. The sum of Half Adder gives second bit of the final product. Similarly, the remaining partial products are added using Half / Full Adders combining them into a group of same positional weight. The addition of all the partial products gives 22 bit output. The first 16 bits of the output gives the final product of 8-bit multiplier. These 16 bits are obtained by adding the first 92 partial products. The addition of the remaining 28 partial products results in generation of the last 6 bits of the output which don't contribute for the value of the product. Hence, these 28 partial products need not be considered for addition in Wallace tree adders.

3. Main modules are combined to implement 8-bit GPU Multiplier which is shown in Figure 4.4.

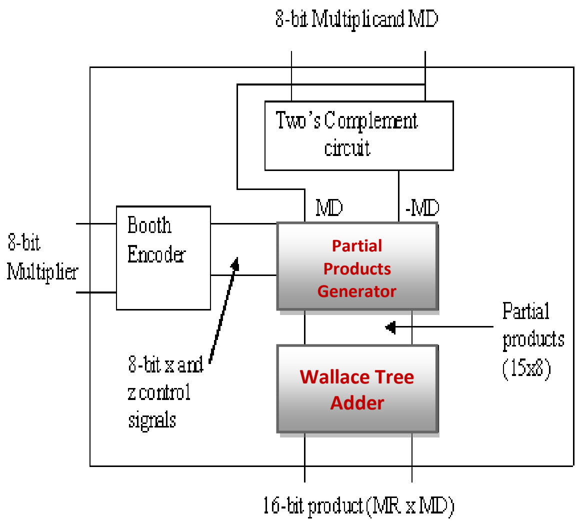

Figure 4.4: Block diagram of the Multiplier

The 8-bit GPU multiplier is the integration of the main and sub Modules. The inputs to the 8-bit multiplier are Multiplier MR and Multiplicand MD as shown in Figure 4.1. MD is applied as input to Two'sComplement generator. This module consists of Inverters and Half Adders and generates –MD as output. MR is applied to Booth encoder. This module consists of XOR gates, Inverters and AND gates. Booth Encoder encodes MR in terms of x and z control signals which are of 8 bits each. These signals along with MD and –MD are applied as inputs to Partial Product generator and its schematic is shown in Figure 4.5. This module generates 8 sets of partial products (each of 8 bits) viz., pp0, pp1, ….., pp7 which are either MD or –MD or all '0' depending on x and z control signals. These partial product sets are MSB sign extended by 7 bits to make each set of partial products of 15 bits. These partial products are applied as input to Wallace tree adder shown in Figure 4.6. This module adds all the partial products of same positional weight and produces 22 bits output. The first 16 bits of the product gives the 8-bit multiplier output. The final 8-bit GPU multiplier schematic is shown in Figure 4.7.

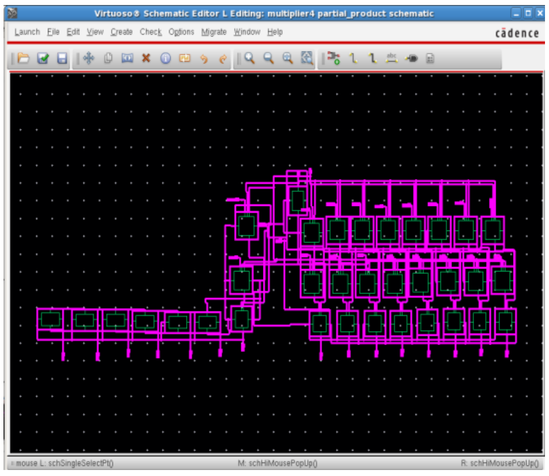
Virtuoso® Schematic Editor L Editing: multiplier4 partial_product schematic
Launch File Edit View Create Check Options Migrate Window Help
cadence
mouse L: schSingleSelectPt()
M: schHiMousePopUp()
R: schHiMousePopUp()

Figure 4.5: Partial product generator schematic (15 bits)

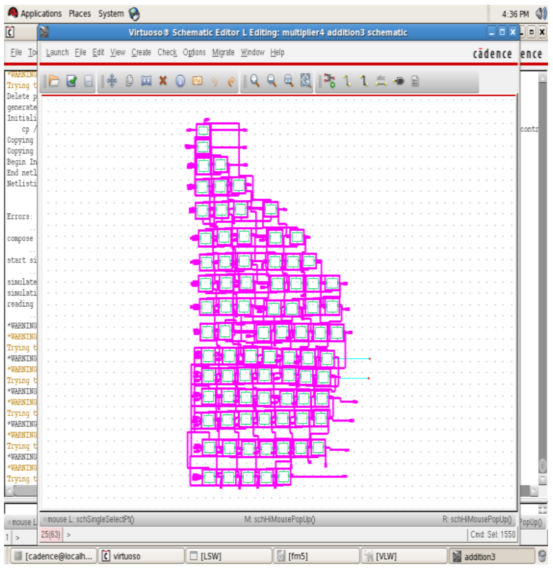
Applications Places System
4:36 PM
Virtuoso® Schematic Editor L Editing: multiplier4 addition3 schematic
Launch File Edit View Create Check Options Migrate Window Help
cadence
mouse L: schSingleSelectPt()
M: schHiMousePopUp()
R: schHiMousePopUp()
Cmd: Sel: 1550
[cadence@localh...
virtuoso
[LSW]
[fm5]
[VLW]
addition3

Figure 4.6: Wallace Tree Adder Schematic

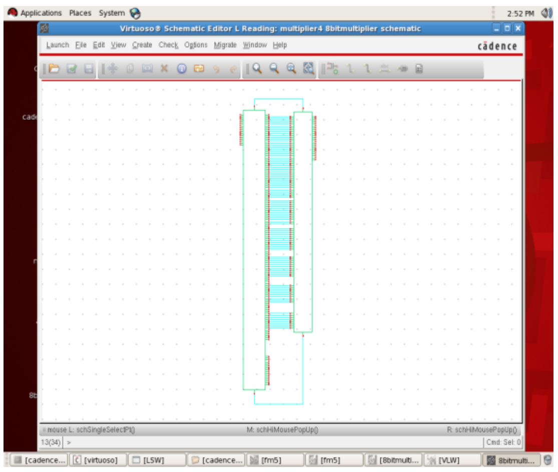

Figure 4.7: Final 8-bit Multiplier Schematic

V. SIMULATION RESULTS

The Power Consumption of proposed circuit is remarkable reduced than the other approach at 180nm technology with supply voltage of 1.8v. The designed 8-bit GPU multiplier circuits are tested successfully for different values of Multiplier MR and Multiplicand MD. The comparative study is made with reference to the following parameters:

1. Number of devices required for 8-bit multiplier in CMOS and two methods of GDI designs.

2. Power dissipation of the 8-bit multiplier in CMOS and both the GDI designs.

The comparative study of the multiplier design is made with reference to the number of devices required for 8-bit multiplier in CMOS and GDI designs as shown in Figure 5.1. Thus, the new GDI design results in 42.7**%** reduction in devices as compared to CMOS design**.**

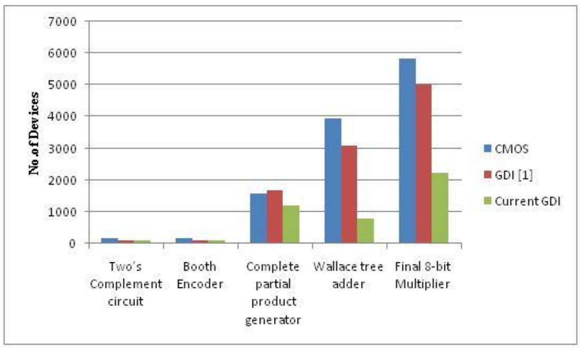

Figure 5.1: Devices required implementing main modules and 8-bit multiplier in CMOS and GDI Logics

The comparison is also made for power consumption of 8 bit multiplier in both GDI and CMOS designs. It is measured that the power consumed by the 8 bit multiplier design in GDI is 7.66μw and 19.48μw in CMOS design as shown in Figure 5.3 and Figure 5.4 respectively (Figure 5.2 shows the Power consumed by the 8 bit multiplier in GDI [1]).

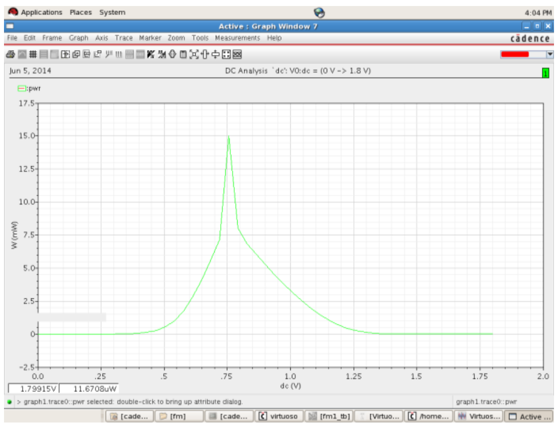

Figure 5.2: Power consumed by the 8 bit multiplier in GDI [1]

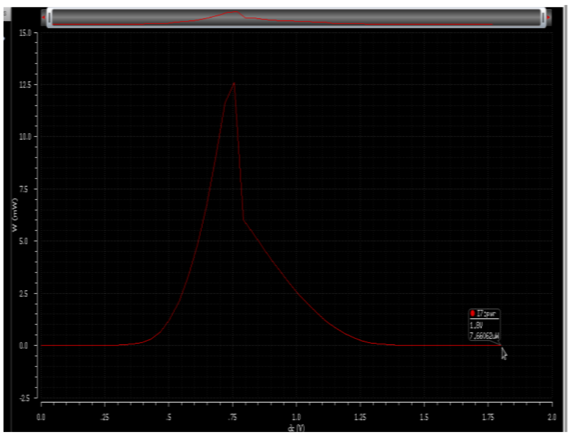

Figure 5.3: Power consumed by the 8 bit multiplier in GDI

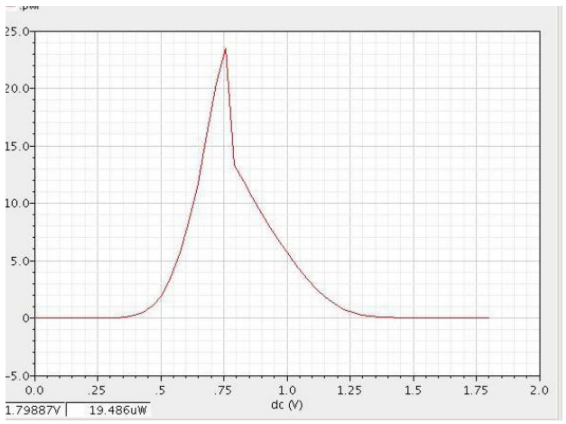

Figure 5.4: Power consumed by the 8 bit multiplier in CMOS

Simulation is also carried out for 9 transistor full adder & 11 transistor full adder circuits. The results show that these circuits consume more power than the 10 transistor full adder circuit when used in the multiplier circuit. The comparison of power consumption of these three full adder circuits is shown in table 2.

Table 2: **Comparison of Power of 9T & 11T Full adders with proposed 10T circuit** in GDI.

Parameters	Average Power
9T	827.19µw
10T	41.03µw
11T	42.23µw

VI. CONCLUSION

This paper proposes a novel approach to improve the GPU execution time by using 8-bit multiplier designed in GDI Logic with Booth encoding and Wallace tree addition techniques using Cadence design suite. The intention of the new multiplier design in GDI is to obtain reduction of power and area in the multiplier design as compared to CMOS design and the previous GDI design. A comparative study is made regarding the area and power dissipation of multiplier using CMOS and GDI logics. It is observed that multiplier designed in new method of GDI results in less number of devices and power reduction.

REFERENCES

[1] B.N.Manjunatha Reddy, Dr.B.R.Vijaya Kumar, Dr.Shanthala S. and H.N.Sheshagiri "Implementation of Low Power 8-Bit Multiplier using Gate Diffusion Input Logic", IEEE 17th International Conference on Computational Science and Engineering, Pages: 1868–1871, DOI:10.1109/CSE.2014. 342.

[2] NVIDIA CUDA Programming Guide 2.2, http://www.nvidia.com.

[3] Christopher Cullinan, Christopher Wyant, Timothy Frattesi, "Computing Performance Benchmarks among CPU, GPU, and FPGA", Mathworks.

[4] RajkumarSarma, Veerati Raju, "design and performance analysis of hybrid adders for high speed arithmetic circuit", International Journal of VLSI design & Communication Systems, Vol.3, No.3, June 2012.

[5] Michael Parker, Understanding Peak Floating-Point Performance Claims", Altera Corporation Technical White Paper, June 2014.

[6] NVIDIA, "NVIDIA's Next Generation CUDA Compute Architecture: Fermi v1.1," NVIDIA, 2009.

[7] PoojaVerma, RachnaManchanda, "Review Of Various GDI Techniques For Low Power Digital Circuits", International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 4, Issue 2, February 2014.

❖❖❖