ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.2, Pages : 57-61 (2013)
Special Issue of NCRTECE 2013 - Held during 8-9 February, 2013 in SMK Fomra Institute of Technology, OMR, Thaiyur, Kelambakkam, Chennai

AREA OPTIMIZATION OF 8-BIT MULTIPLIER USING GATE DIFFUSION INPUT LOGIC

[1]B.N. Manjunatha Reddy, [2]H. N. Sheshagiri, [3]Dr. Shanthala S.
[1,2]Global Academy of Technology, Bangalore. [3] Bangalore Institute of Technology, Bangalore.
manjunatha_reddy@gat.ac.in, hnsg1957@gmail.com, shanthala_wg@yahoo.com

Abstract:* *Multipliers are used in all modern digital systems and DSP applications. They are used in hardware multiplication to achieve high data throughput. Multipliers are major sources of area consumption and power dissipation in such systems. Reduction in area can be achieved using Booth encoding and Wallace tree technique since they generate partial products efficiently and are most suited for multiplication of signed numbers. Multiplier designed in Gate Diffusion Logic (GDI) logic requires lesser number of devices as compared to CMOS logic. Hence, the proposed multiplier design will substantially reduce the number of devices as compared to CMOS design. This results in area optimization with the consequent reduction in power and delay of multiplier.

Key Words:* *GDI, Booth Encoder, Partial Products Generator, Wallace Tree Adder.

I. INTRODUCTION

CMOS multipliers occupy substantial area. To further optimize area, it is required to reduce the number of devices in design of multipliers. This can be implemented using a revised CMOS logic viz., Gate Diffusion Input (GDI) technique. Circuits designed in GDI are based on GDI basic cell as shown in Fig.1 which allows us to apply more than one input (G,P&N) to a device unlike in CMOS circuits. Some of the functions implemented in GDI logic are as in Table1 [1].

It is possible to design and implement complex functions in GDI with fewer devices as compared to CMOS. This results in reduction of the area and also power consumption & delay of GDI circuits [1].

II. DESIGN OF 8-BIT MULTIPLIER

The block diagram of the proposed multiplier is as shown in Fig.2. Booth encoder is employed for efficient generation of partial products [2]. The partial products generated are either Multiplicand (MD) or complement of MD (–MD) or ‘0’ depending on the output of Booth Encoder [3]. Wallace tree technique is implemented using Full Adders with 3:2 compression.

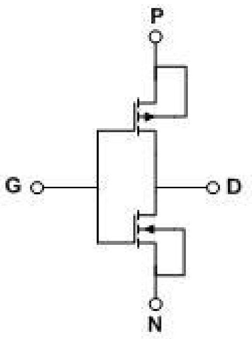


Fig.1: GDI basic cell [1]

Table1: Some functions implemented in GDI [1]

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'

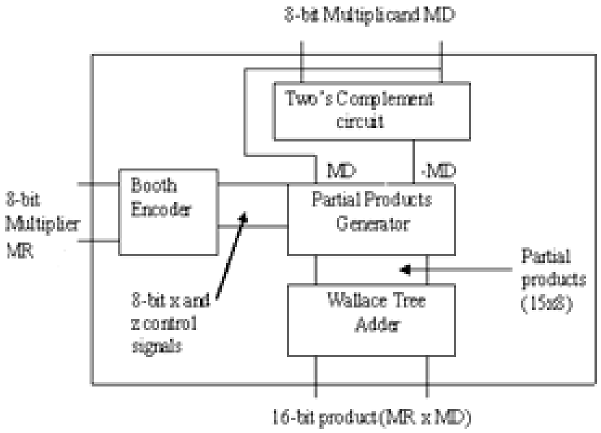


Fig.2: Block diagram of 8-bit GDI Multiplier

The input to Booth Encoder is Multiplier (MR) and this module consists of 8 XOR gates, 7 Inverters and 8 AND gates. The output of Booth encoder consists of two sets of signals called x and z control signals. A set of 8 ‘x’ control signals are generated by performing the inversion of the previous bit and ANDing it with the current bit. Another set of 8 ‘z’ control signals are generated by performing XORing of the current and previous bit of MR as shown in Fig.3. The previous bit for LSB is taken as ‘0’ for generation of both x and z control signals.

The partial products generator for 1 bit of MR is shown Fig.4. It has inputs as MD & -MD (from 2’s complement circuit) and x & z control signals from Booth encoder. It gives 8-bit output which is sign extended to make it 15-bit partial product (pp). The design uses eight such modules for 8 bits of MR.

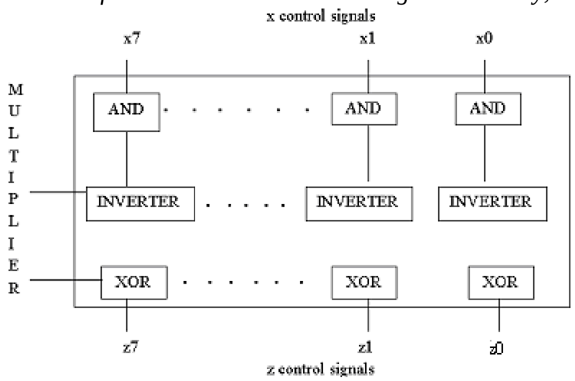


Fig.3: Block diagram of Booth encoder

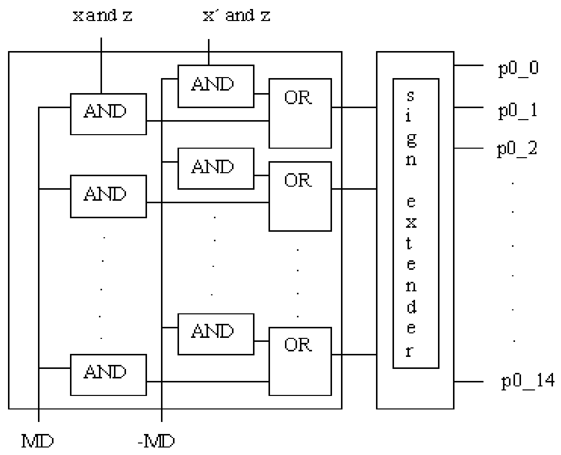


Fig.4: Block diagram of Partial product generator for 1-bit of MR

The final module of the multiplier is Wallace tree adder. This consists mainly of Full adders, half adders and a buffer. The partial products of similar weight are added using either Half or Full adders. The block diagram is shown in Fig.5. The output from Wallace tree adder is 22 bits. But, the first 16 bits from LSB will give the desired product of MR and MD.

III. IMPLEMENTATION OF MULTIPLIER IN GDI

The 8-bit Multiplier shown in fig.2 is now designed using GDI technique. The basic gates, sub and main modules are implemented in GDI using Cadence proprietary general purpose design kit viz., gpdk 180. The design is carried out in Cadence design suite which has Virtuoso schematic & layout editor and Assura DRC, ERC & LVS. In the design, Vdd is taken as 3.3 volts and simulation is carried for 10 nano seconds.

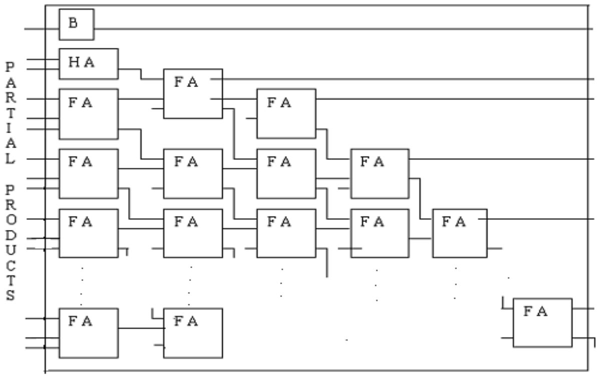


Fig.5: Block diagram of Wallace tree adder

In the first step, the design of the basic gates, sub modules like Sign Extender, Half and Full adders are carried out in GDI with reference to Table1. These sub modules are combined in the main modules of multiplier design.

Design of basic gates (AND, OR) and XOR gates in GDI logic: The two input AND & OR gates are implemented using A'B and A'+B options respectively (instead of AB and A+B) as in Table 1, to avoid deterioration of output levels. The schematic diagrams for these gates are shown in Fig. 6 & Fig.7 respectively. Inverters are used to get complements of inputs. Additional role of these Inverters is to buffer the internal signals to restore the swing and to improve driving ability of output [4]. Also, XOR gate is realized as shown in Fig.8.

The sub modules are designed using the above gates. For example, Half adder is implemented as Sum= $A \oplus B$ and Carry = AB.

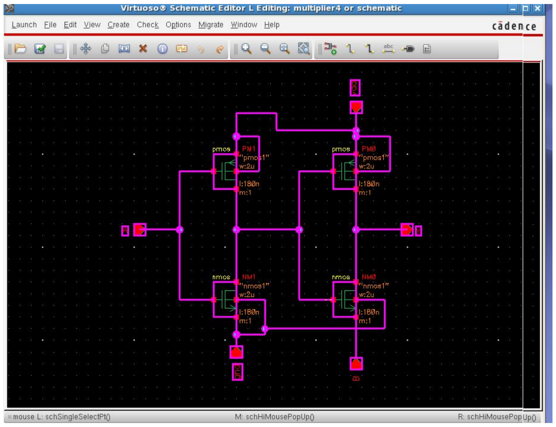

Virtuoso® Schematic Editor L Editing: multiplier4 or schematic
Launch File Edit View Create Check Options Migrate Window Help
cādence
pmos
PM1
"pmos1"
w:2u
l:180n
m:1
pmos
PM0
nmos
NM1
"nmos1"
nmos
NM0
mouse L: schSingleSelectPt()
M: schHiMousePopUp()
R: schHiMousePopUp()

Fig.6: Two Input OR gate schematic

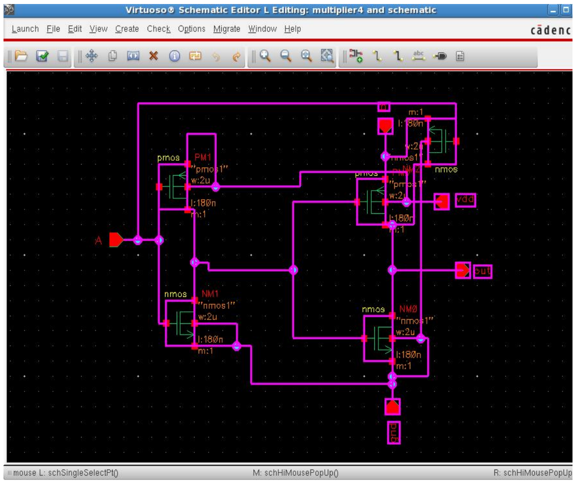
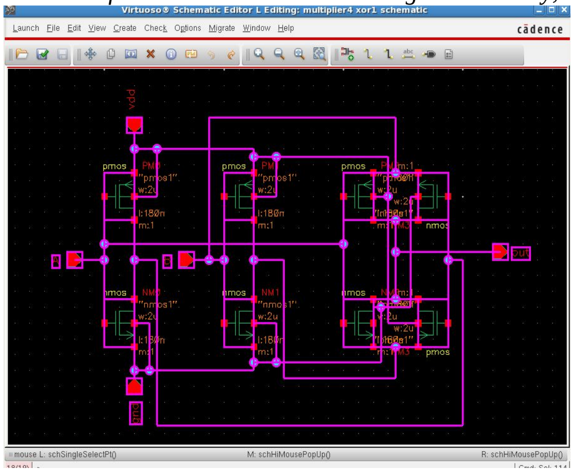

Virtuoso® Schematic Editor L Editing: multiplier4 and schematic
Launch File Edit View Create Check Options Migrate Window Help
cādence
pmos
PM1
"pmos1"
w:2u
l:180n
m:1
nmos
NM1
"nmos1"
NM0
A
out
mouse L: schSingleSelectPt()
M: schHiMousePopUp()
R: schHiMousePopUp()

Fig.7: Two input AND gate schematic


Virtuoso® Schematic Editor L Editing: multiplier4 xor1 schematic

mouse L: schSingleSelectPt() M: schHiMousePopUp() R: schHiMousePopUp()

Fig.8: Two input XOR gate schematic

The design of multiplier is carried out by combining these sub modules to implement the main modules viz., i) Booth Encoder, 2's complement circuit & Partial products generator and ii) Wallace tree adder. The schematics of the above modules are as shown in Fig.9 & Fig.10 respectively.

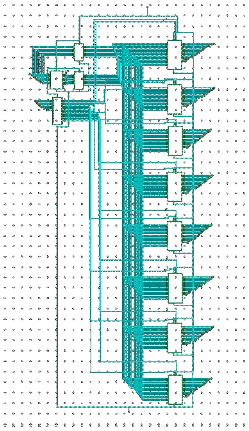


Fig.9: Complete schematic of partial product generator (8x15 bits) along with Booth encoder and 2's complement circuit.

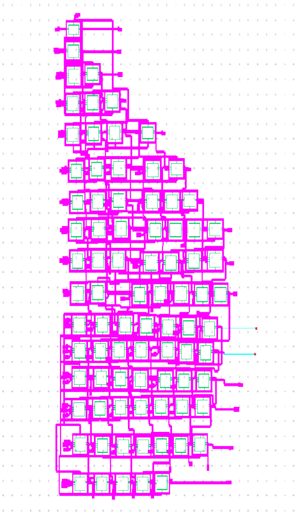


Fig10: Wallace tree adder schematic

The final multiplier is implemented by combining the main modules as indicated in Fig.2. The final schematic and layout of 8-bit multiplier are shown in Fig.11 & Fig.12 respectively.

IV. RESULTS AND COMPARATIVE STUDY OF THE MULTIPLIER DESIGN

The designed multiplier circuit is tested successfully for different values of Multiplier MR and Multiplicand MD.

1. Example1: MD=11000100 and MR=00100000
2. Example2 : MD=01001010 and MR=01100100
3. Example3 : MD=11001110 and MR=10100000

The results are verified and found to be matching with the desired product.

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.

Table 2 indicates number of devices required to implement basic gates and adders in CMOS and GDI logics.

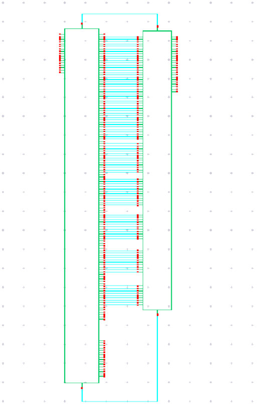
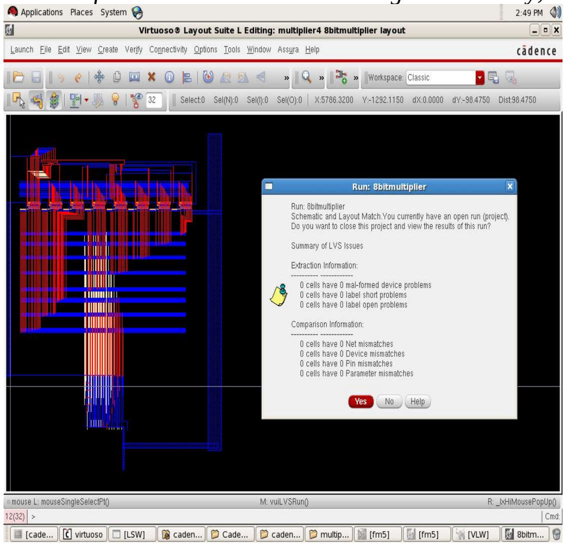


Fig.11: Final 8-bit Multiplier schematic


Applications
Places
System
2:49 PM
Virtuoso® Layout Suite L Editing: multiplier4 8bitmultiplier layout
Launch File Edit View Create Verify Connectivity Options Tools Window Assura Help
cādence
Workspace: Classic
32
Select:0 Sel(N):0 Sel(I):0 Sel(O):0 X:5786.3200 Y:-1292.1150 dX:0.0000 dY:-98.4750 Dist:98.4750
Run: 8bitmultiplier
Yes
No
Help
mouse L: mouseSingleSelectPt()
M: vuiLVSRun()
R: _lxHiMousePopUp()
Cmd:

Run: 8bitmultiplier
Schematic and Layout Match.You currently have an open run (project).
Do you want to close this project and view the results of this run?

Summary of LVS Issues

Extraction Information:
---------- -----------
0 cells have 0 mal-formed device problems
0 cells have 0 label short problems
0 cells have 0 label open problems

Comparison Information:
---------- -----------
0 cells have 0 Net mismatches
0 cells have 0 Device mismatches
0 cells have 0 Pin mismatches
0 cells have 0 Parameter mismatches

Fig..12: Final 8-bit Multiplier layout

Table 3 shows total number of gates and adders needed to implement the main modules in the multiplier circuit like Two's complement circuit, Booth Encoder, Partial product generator and Wallace tree adder in both CMOS and GDI logics.

The requirement of nMOS and pMOS devices for the main modules and the final 8-bit Multiplier is arrived by combining the data of Table 2 & 3 which is shown in Table 4. This table shows that GDI requires 5015 devices and CMOS design requires 5844 devices to design 8-bit multiplier. Thus, 8-bit multiplier implemented in GDI results in a reduction of 14.2 % of devices as compared to CMOS design. Thus, GDI logic reduces the area of multiplier.

V. CONCLUSION and FUTURE SCOPE

8-bit multiplier is designed in GDI Logic with Booth encoding and Wallace tree addition techniques using Cadence design suite (gpdk 180).

The Layouts are drawn for the basic gates, sub modules and main modules separately. Layout is also drawn for Final 8-bit multiplier and it matches with the corresponding schematic diagram. Layout Versus Schematic is matched for all devices.

Table 2: Number of devices required to implement basic gates and adders in CMOS and GDI logic styles.

Sl. No.	Gates/Adders	Devices in CMOS	Devices in GDI
1	Inverter	2	2
2	Two input AND Gate	6	5
3	Two input XOR Gate	12	8
4	Two input OR Gate	6	4
5	Half Adder	18	13
6	Full Adder	50	39

Table 3: Requirement of gates and adders to implement main modules of 8-bit multiplier in CMOS or GDI logic

Sl. No.	Main Modules	I	HA	XOR	AND	OR	SE/B	FA
1	Two's comp. Circuit	8	8	-	-	-	-	-
2	Booth encoder	7	-	8	8	-	-	-
3	Partial product generator (1set)	1	-	-	16*	8	-	-
4	Complete partial product generator (8 sets)	8	-	-	128*	64	35	-
5	Wallace tree adder	-	6	-	-	-	1	77

I – Inverter; HA- Half Adder; SE/B – Sign Extender / Buffer; FA – Full Adder * Three input AND gates

Table 4: Requirement of nMOS and pMOS devices to implement main modules of 8-bit multiplier in CMOS and GDI logics

Sl. No.	Main Modules	Number of devices in CMOS design	Number of devices in GDI design
1	Two's Complement circuit	160	120
2	Booth Encoder	158	118
3	Complete partial product generator (8 bit)	1564	1692
4	Wallace tree adder	3962	3085
5	Final 8-bit Multiplier (1+2+3+4)	5844	5015

The objective of the multiplier design in GDI is to show reduction of area in the multiplier design compared to CMOS design. A comparative study is made regarding the total number of devices required for the multiplier design using CMOS and GDI logics. It is observed that multiplier designed in GDI results in 14.2 % reduction of devices, thus minimizing the area of multiplier.

The future work that may be carried out on the current multiplier design is as follows:

1. Carry Save Adders (CSA) and 4:2 compression may be adapted in the Wallace tree adder to further reduce the multiplier delay.
2. The performance of the multiplier may be enhanced by improving the current design. For example, the speed performance of long full adder chain is improved by

going in for mixed full adder topology in which CMOS inverter gates are inserted in the chain to optimize delay and power [5]. The multiplier design in GDI may be implemented in SOI or twin well CMOS process to yield a better power and delay efficient design.

REFERENCES

[1]. 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, DOI: 10.1109 / TVLSI.2002.801578.

[2]. David Villeger and Vogin G. Oklobdzija, “Evaluation of Booth encoding techniques for parallel Multiplier implementation”, Electronics Letters, IET Journals & Magazines, 1993.

[3]. http://www.eecs.tufts.edu/ ~ryun01 / vlsi/index.htm.

[4]. Arkadiy Morgenshtein, Alexander Fish, and Israel A.Wagner, “An efficient implementation of D Flip Flop using GDI technique”, 2004 IEEE, DOI: 10.1109/ISCAS.2004.1329361.

[5]. Adarsh Kumar Agrawal, S. Wairya, R.K.Nagraria and S. Tiwari, “A new Mixed Gate Diffusion Input Full Adder Topology for High Speed Low Power Digital Circuits”, World Applied Sciences Journal 7(Special Isue of Computer & IT), IDOSI Publications, 2009.