CMOS VLSI Design

LAB BASED PROJECT REPORT

On

CMOS 3-Bit to Square of the Given Input

A Report Submitted in Partial Fulfilment of the Requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

by

Saurabh Mhaske

Enrollment Number: BT20ECE021

Semester: 6th

Submitted To

Dr. Paritosh Peshwe

Assistant Professor

Department of Electronics and Communication Engineering

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY NAGPUR

(An Institution of National Importance by Act of Parliament)

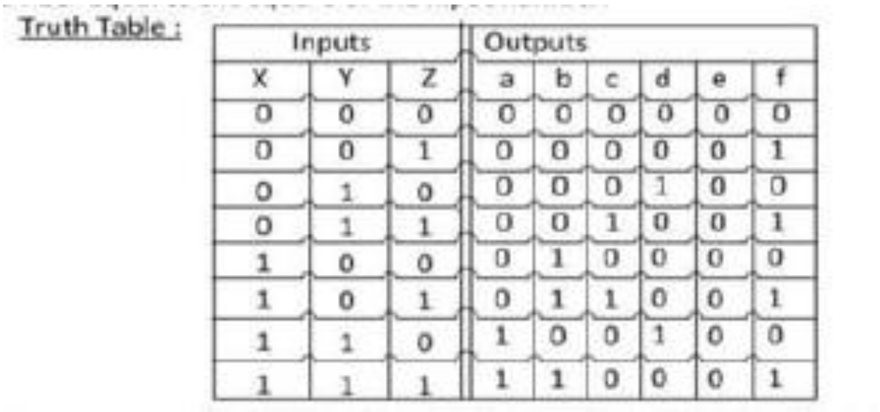
Contents

Project Title: CMOS 3-Bit Binary to Square of the given Input

ABSTRACT

Area and power minimization are the prime concerns in recent VLSI design. As chip size is shrinking and many other micro-electronics reliabilities are developing gradually, low power and small area design of any system has become priority. The performance is an important element to determine the efficiency of the whole circuit. In this Project, an area efficient layout design of has been proposed.

Design a combinational circuit that accepts a three-bit number and generates an output binary number equal to the square of the input number.



Truth Table :

Inputs			Outputs					
X	Y	Z	a	b	c	d	e	f
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

We see that, the all output of e is equal to zero and the all output of f is equal to the input z. so we can write e=0 and f=Z. Now we draw the map for a, b, c, d.

INTRODUCTION

CMOS TECHNOLOGY

The first working point contact transistor developed by John Bardeen, Walter Brattain and William Shockley at Bell laboratories in 1947 initiated the rapid growth of the information technology industry. In 1958, J Kilby invented the first integrated circuit flip flop at Texas and soon after this; Frank Wan lass at Fairchild described the first CMOS logic gate (NMOS and PMOS) in 1963.One of the most popular MOSFET technologies available today is the Complementary MOS or CMOS technology. CMOS technology is the dominant semiconductor technology for microprocessors, memories and application specific integrated circuits (ASICs).

The main advantage of CMOS over NMOS and BIPOLAR technology is the much smaller power dissipation. Unlike NMOS or BIPOLAR circuits, a CMOS circuit has almost no static power dissipation. Power is only dissipated in case the circuit actually switches. This allows to integrate many more CMOS gates on an IC than in NMOS or bipolar technology, resulting in much better performance.

In CMOS logic gates a collection of n-type MOSFETs is arranged in a pulldown network between the output and the lower voltage power supply rail (Vss or quite often ground). Instead of the load resistor of NMOS logic gates, CMOS logic gates have a collection of p-type MOSFETs in a pull-up network between the output and the higher voltage rail (often named Vdd). Thus, if both a p-type and n-type transistor have their gates connected to the same input, the p-type MOSFET will be on when the n-type MOSFET is off, and vice-versa. Area and power minimization are the prime concerns in recent VLSI design. As chip size is shrinking and many other micro-electronics reliabilities are developing gradually, low power and small area design of any system has become priority. The performance is an important element to determine the efficiency of the whole circuit. In this project, an area efficient layout design of has been proposed.

2. AIM OF THE PROJECT:

1. The Main Aim of the Project is To Understand how to design a schematic circuit using complimentary logic with respect to the inputs and outputs.

2. To learn how to design a schematic in Ngspice Software without errors.
3. To understand how to draw stick diagrams for a circuit and how to draw their layouts using Micro-Wind Software.

3. COMPONENTS REQUIRED:

A) Ngspice
B) Micro-Wind Software

4. Rules to Design a CMOS Schematic:

1. Before going to draw the CMOS logic circuits, we have to know the basic CMOS circuits for logic gates.
2. The given expression should contain whole bar.
3. No, two inputs have a common bar other than whole bar. But we can have bar for single input.
4. We should not remove any input during simplification of the Boolean expression.
Note: For NAND PMOS-Parallel; NMOS - Series
For NOR PMOS-Series; NMOS – Parallel

5. Stick Diagram and its Rules:

VLSI design aims to translate circuit concepts onto silicon. stick diagrams are a means of capturing topography and layer information using simple diagrams. Stick diagrams convey layer information through color codes (or monochrome encoding). Acts as an interface between symbolic circuit and the actual layout.
Does show all components. It shows relative placement of components. Goes one step closer to the layout. Helps plan the layout and routing. A stick diagram is a cartoon of a layout.

Notations

Metal-1 → Sky Blue
Metal-2 → Thick Blue
Poly Silicon → Red
Contact Cut → Black Dot
N- Diff → Green
P- Diff → Yellow

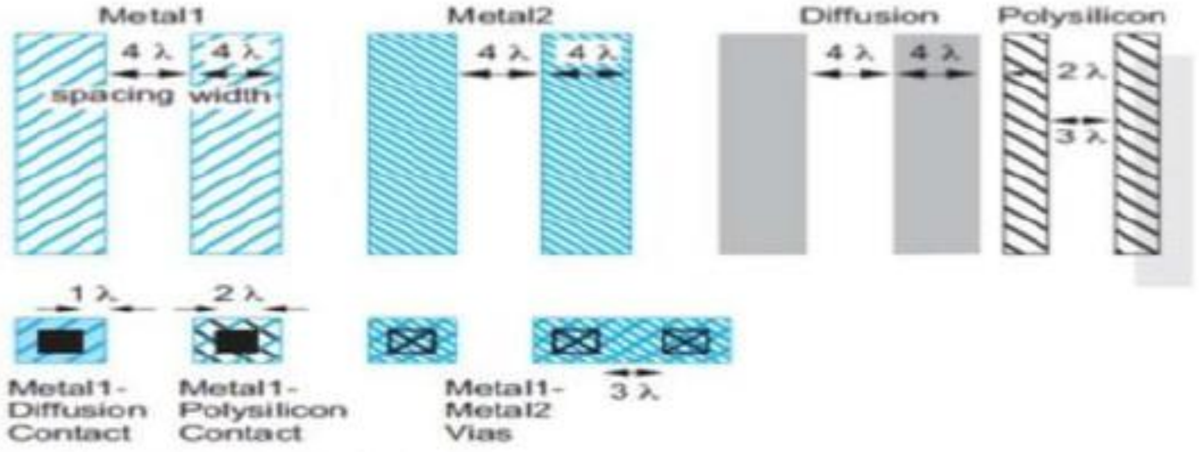
Rules need to be followed

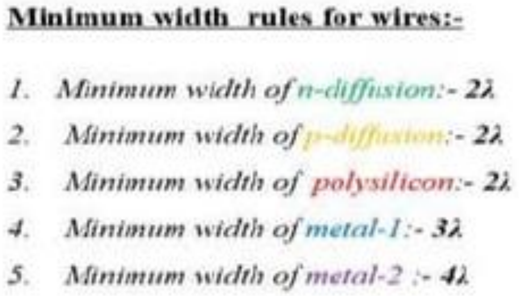
Rule 1. When two or more 'sticks' of the same type cross or touch each other that represents electrical contact.

Rule 2. When two or more 'sticks' of different type cross or touch each other there is no electrical contact. (If electrical contact is needed we have to show the connection explicitly).

Rule 3. When a poly crosses diffusion it represents a transistor. Rule 4. In CMOS a demarcation line is drawn to avoid touching of p-diff with n-diff: All pMOS must lie on one side of the line and all nMOS will have to be on the other side.

6. Lambda based Rules to draw a Layout:



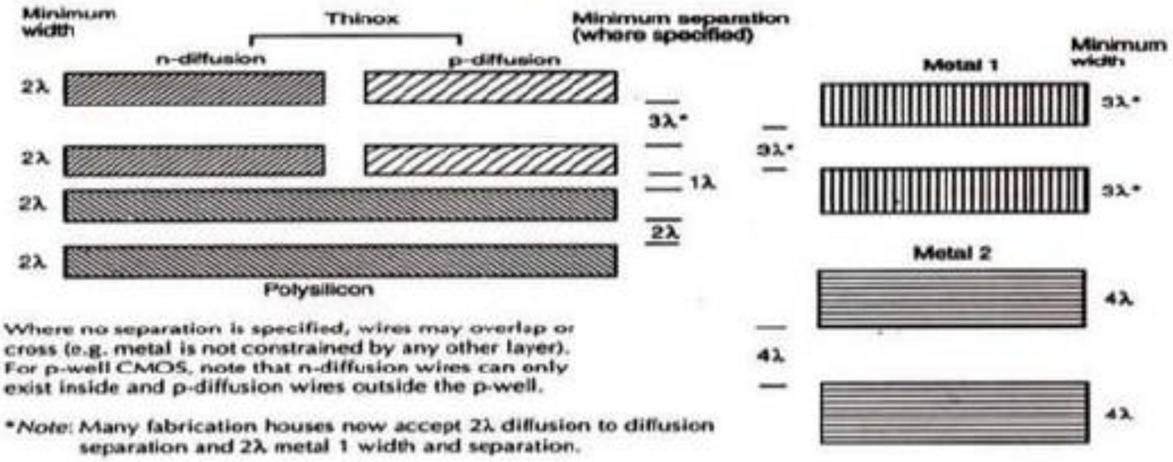


Minimum width rules for wires:-

1. *Minimum width of n-diffusion:- 2λ*
2. *Minimum width of p-diffusion:- 2λ*
3. *Minimum width of polysilicon:- 2λ*
4. *Minimum width of metal-1 :- 3λ*
5. *Minimum width of metal-2 :- 4λ*

Minimum Spacing rules for wires:-

1. *Minimum spacing b/w of* ***n-diff to n-diff****:- 3λ*
2. *Minimum spacing b/w of* ***p-diff to p-diff****:- 3λ*
3. *Minimum spacing b/w of* ***n-diff to p-diff****:- 3λ*
4. *Minimum spacing b/w of* ***diffusion(n or p) to polysilicon****:- 1λ*
5. *Minimum spacing b/w* ***poly to poly****:- 2λ*
6. *Minimum spacing b/w* ***metal-1 to metal-1****:- 3λ*
7. *Minimum spacing b/w* ***metal-2 to metal-2****:- 4λ*



Where no separation is specified, wires may overlap or cross (e.g. metal is not constrained by any other layer). For p-well CMOS, note that n-diffusion wires can only exist inside and p-diffusion wires outside the p-well.

*Note: Many fabrication houses now accept 2λ diffusion to diffusion separation and 2λ metal 1 width and separation.

7. About the Project:

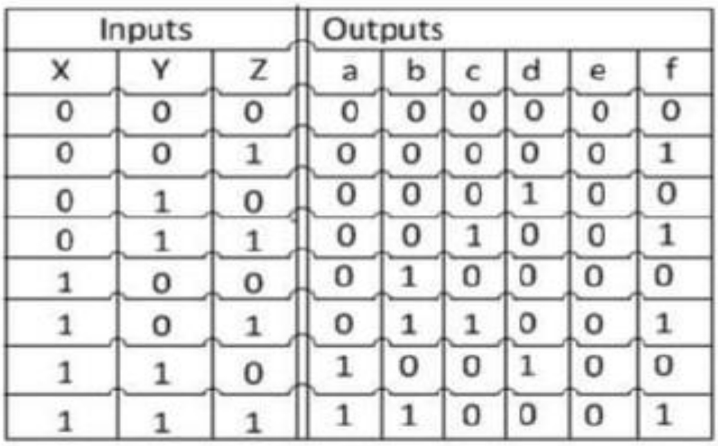
In these Project we are performing the operation of the square of the given 3-bit binary input. For these we are using complementary gate logic. Here, first of all we write the truth table of the required operation. After that we construct the K-maps for the outputs individually. After getting the required Boolean expressions for each output, we have to draw the schematic for those expressions by following rules mentioned above. For these Project we get the output "e" as zero. So, no need to apply k-map, we directly take It as zero. And for output "f" we get the values same as input 3 that is Z. so we no need to get the expression for these also. We can directly take these 2inputs.

After drawing the schematic, we have to check for the output. We are doing this by using Ngspice software. After the getting the exact output of the NETLIST in Ngspice, we have to draw the stick diagram for the

schematic by following proper rules and colors. Drawing the stick diagram doesn't complete your project. We have to do the layout for the schematic based on stick diagram in Micro-Wind. After the completion compare the waveforms with the expected output. So that we can go for
the extension of these project.

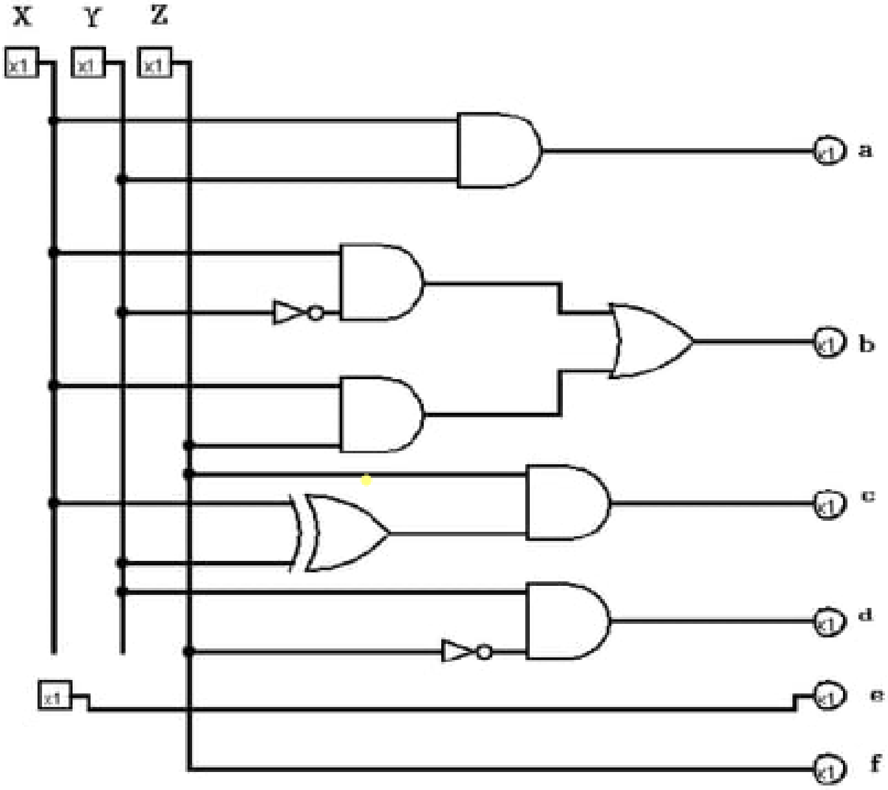
8. Truth Table:

Truth table

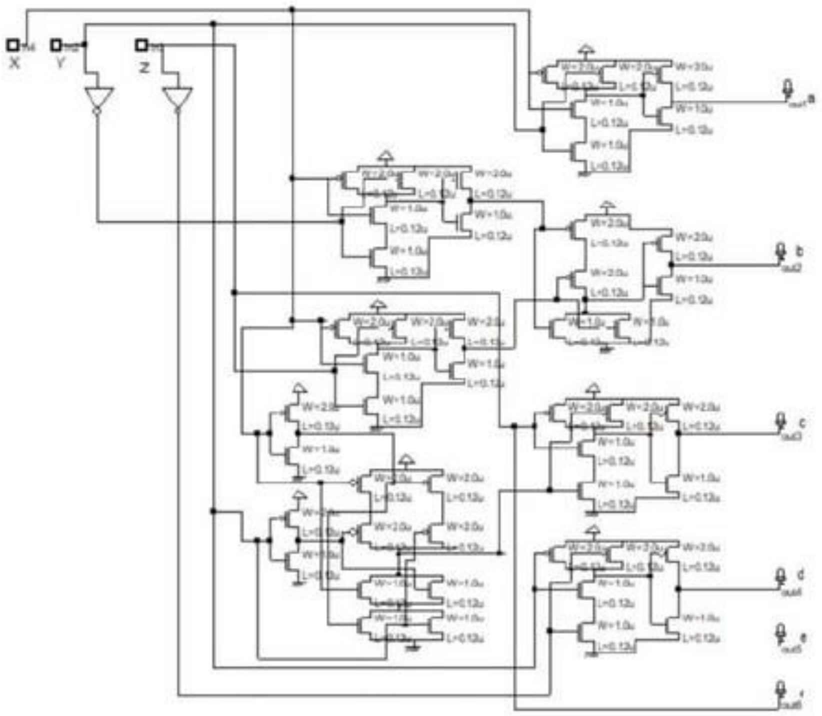


Inputs			Outputs					
X	Y	Z	a	b	c	d	e	f
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

Circuit Diagram



9. Schematic of the Project:



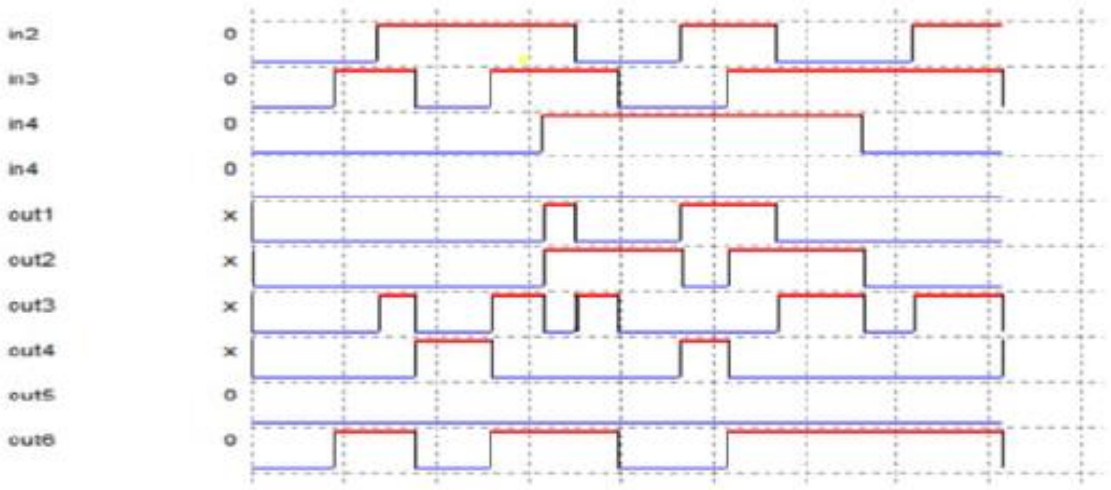
10. Netlist the Project:

```
*** 3-Bit Binary to square of the Given Input Using
Pass-Transistor Logic ***

.subckt inverter 1 2 3
M1n 2 1 0 0 nmod w = 40u l = 1u
M1p 2 1 3 3 pmod w = 40u l = 1u

.model nmod nmos level = 54 version = 4.7
.model pmod pmos level = 54 version = 4.7
.ends

.subckt pass_and 1 2 3 4
M1n 1 2 3 3 nmod w = 40u l = 1u
M2n 2 4 3 3 nmod w = 40u l = 1u

.model nmod nmos level = 54 version = 4.7
.ends
```

```
.subckt pass_or 1 2 3 4
M1n 1 4 3 3 nmod w = 40u l = 1u
M2n 2 2 3 3 nmod w = 40u l = 1u

.model nmod nmos level = 54 version = 4.7
.ends

.subckt pass_xor 1 2 3 4 5
M1n 1 5 3 3 nmod w = 40u l = 1u
M2n 4 2 3 3 nmod w = 40u l = 1u

.model nmod nmos level = 54 version = 4.7
.ends

Vdd 1 0 dc 5V
Va 10 0 pulse(0 5 0 0 0 20ns 40ns)
Vb 11 0 pulse(0 5 0 0 0 15ns 30ns)
Vc 12 0 pulse(0 5 0 0 0 10ns 20ns)
Vd 13 0 dc 0V

xa 10 14 1 inverter
xb 11 15 1 inverter
xc 12 16 1 inverter
xd 13 27 1 inverter

xa_and_b 10 11 17 15 pass_and

xa_and_b' 10 15 18 11 pass_and
xa_and_c 10 12 19 16 pass_and
xe 19 24 1 inverter
xa_and_b'_or_e 18 19 20 24 pass_or

xa_xor_b 10 11 21 14 15 pass_xor
xf_and_c 21 12 22 16 pass_and

xb_and_c' 11 16 23 12 pass_and

xd_and_d 13 13 25 27 pass_and
xc_and_c 12 12 26 16 pass_and

.tran 0.1ns 40ns
```

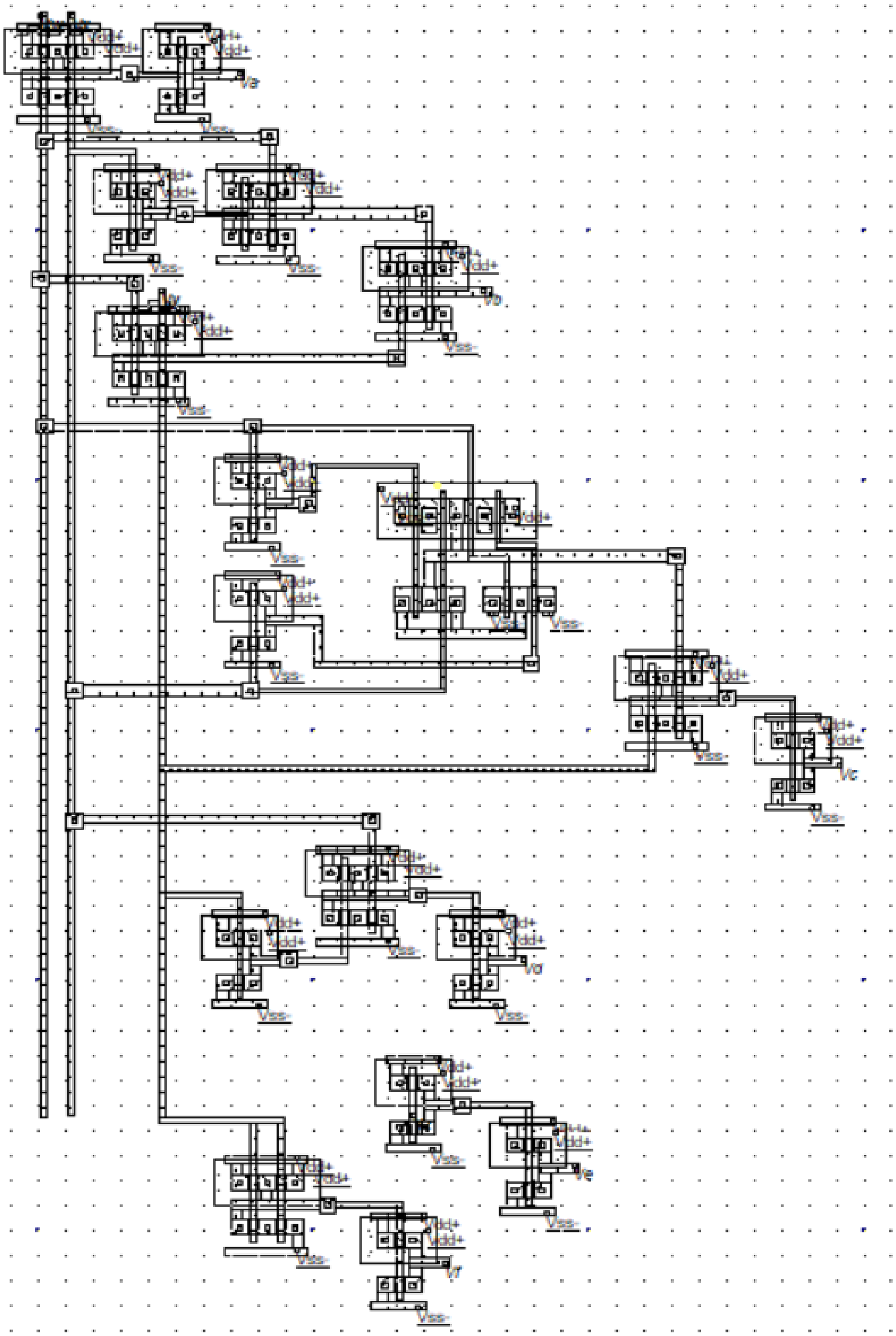
```
.control
run

plot W(10) title 'W 1st_input bit' xlabel 'Time' ylabel
'Output'
plot X(11) title 'X 2nd_input bit' xlabel 'Time' ylabel
'Output'
plot Y(12) title 'Y 3rd_input bit' xlabel 'Time' ylabel
'Output'
plot Z(13) title 'Z 4th_input bit' xlabel 'Time' ylabel
'Output'

plot A(17) title 'A 1st_output bit' xlabel 'Time' ylabel
'Output'
plot B(20) title 'B 2nd_output bit' xlabel 'Time' ylabel
'Output'
plot C(22) title 'C 3rd_output bit' xlabel 'Time' ylabel
'Output'
plot D(23) title 'D 4th_output bit' xlabel 'Time' ylabel
'Output'
plot E(25) title 'E 5th_output bit' xlabel 'Time' ylabel
'Output'
plot F(26) title 'F 6th_output bit' xlabel 'Time' ylabel
'Output'

.endc
.end
```

Output of the NETLIST:

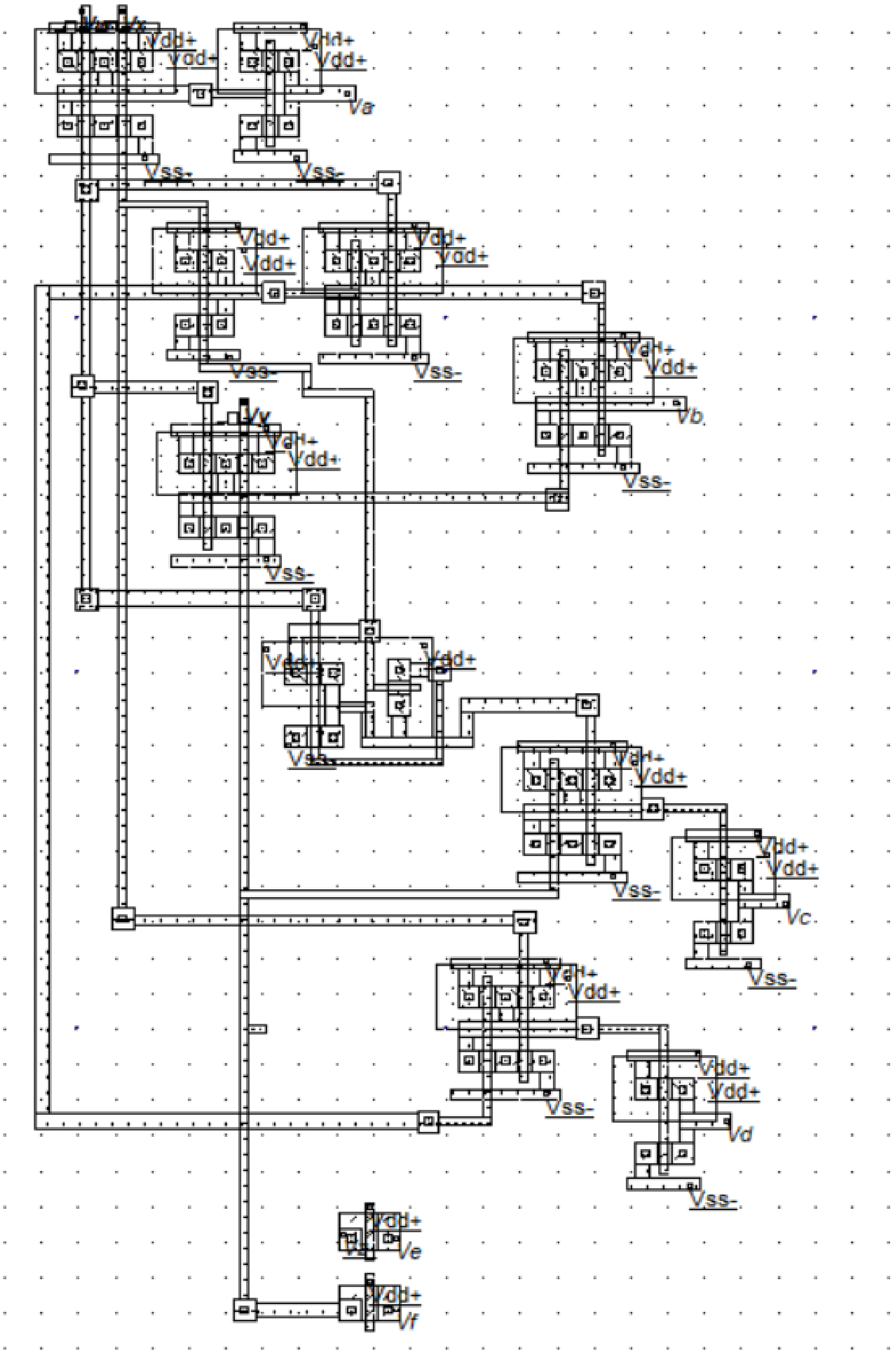


11. Layout for the Schematic:

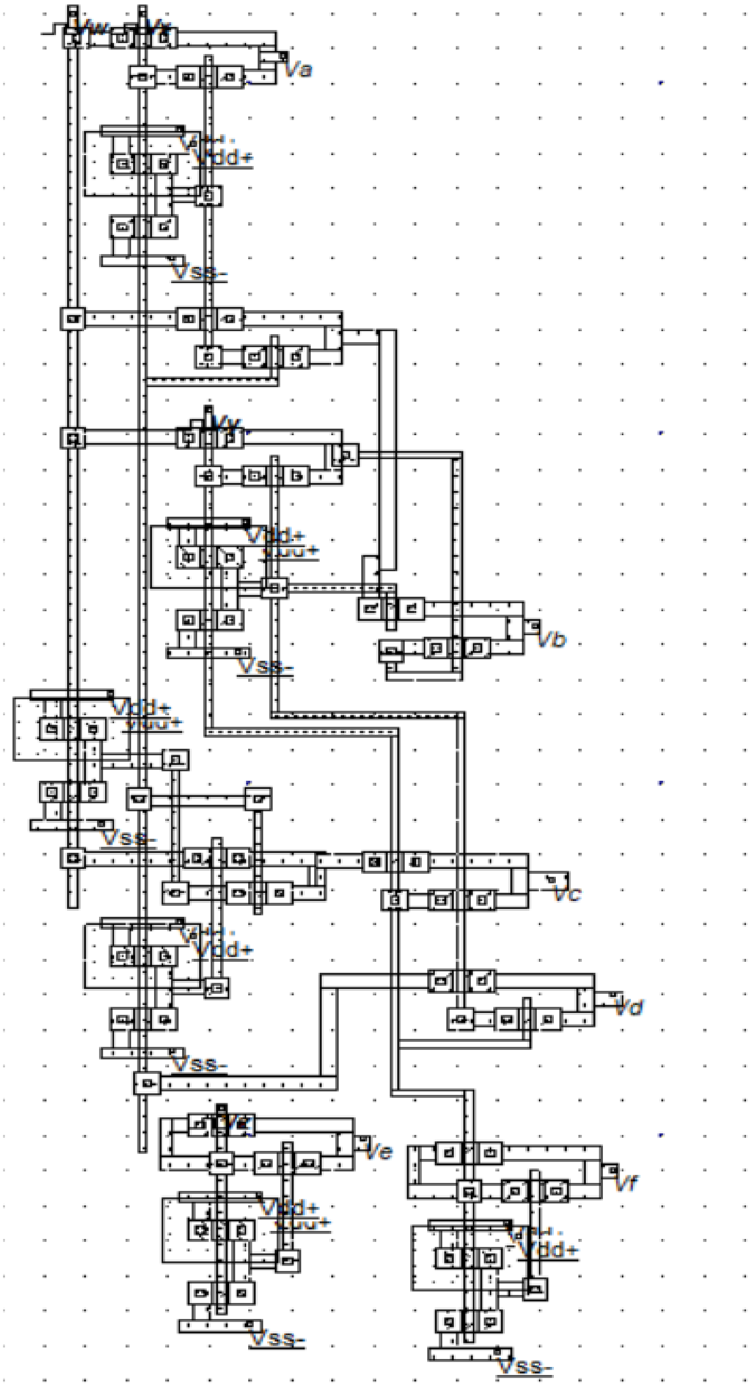
Stage 1 using CMOS Logic :



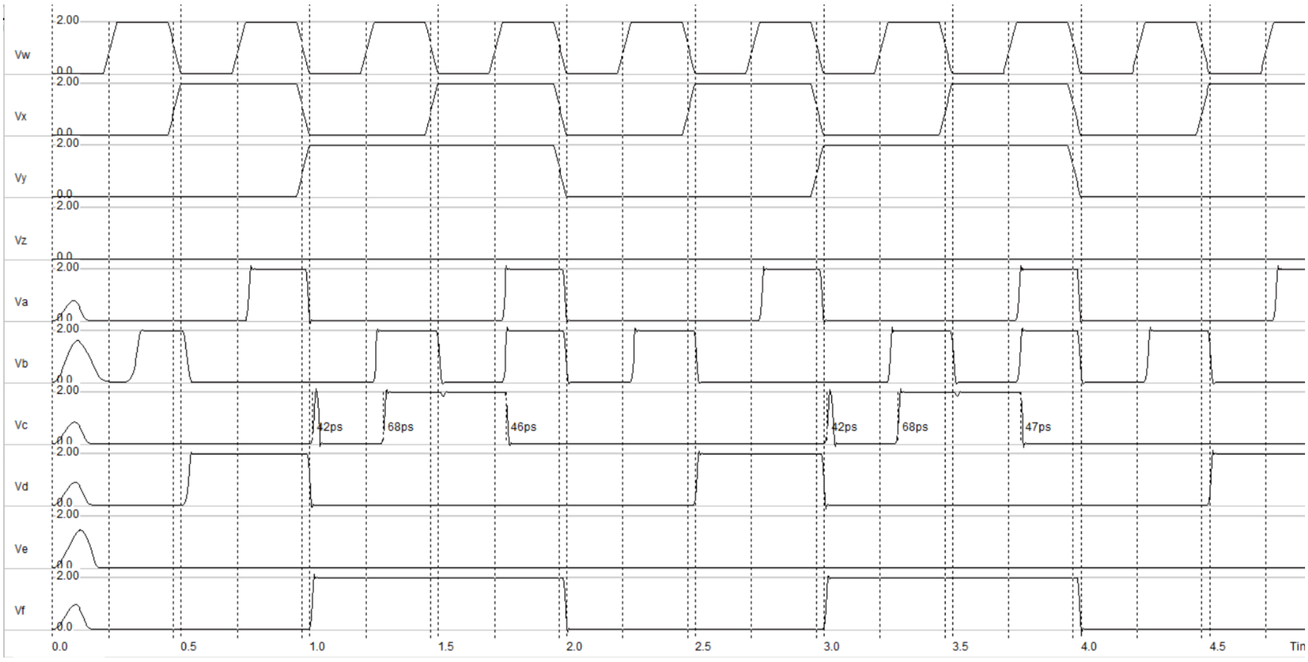
Stage 2 (Optimized) using CMOS Logic:



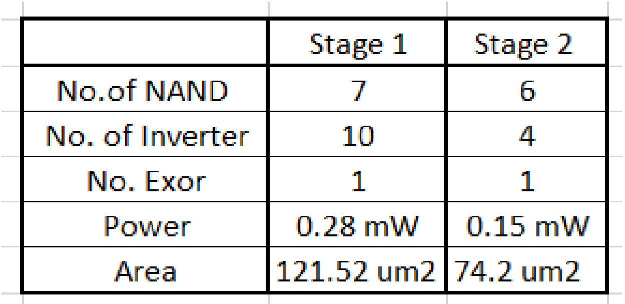
Stage 3 using Pass Transistor:



Output of the Layout:



Comparison Between stage 1 and Optimized Layout:



	Stage 1	Stage 2
No.of NAND	7	6
No. of Inverter	10	4
No. Exor	1	1
Power	0.28 mW	0.15 mW
Area	121.52 um2	74.2 um2

12. Application:

Used in Calculator, mobiles and some other electronic appliance.

13. Precautions:

a) Connections should be made carefully.
b) Make sure that Vdd, GND are given.
c) At the time of Stick diagram, no same colours should be Crossed if there is no connection at the point.
d) Layout should be done by checking DRC at step-to-step connection.
e) Observer the Waveforms Properly.

14. Results and Observations:

a) From this experiment we are able to design the schematic using CMOS Logic.
b) Developed the experience of how to observe the output Waveforms by comparing with truth table.
c) Learned to draw a stick diagram for complex circuits.
d) And also learned how to use NETLIST, Micro-wind software without getting errors.

15. Conclusion:

Area and power minimization are the prime concerns in recent VLSI design. As chip size is shrinking and many other micro-electronics reliabilities are developing gradually, low power and small area design of any system has become priority. The performance is an important element to determine the efficiency of the whole circuit. The main advantage of CMOS over NMOS and BIPOLAR technology is the much smaller power dissipation. Unlike NMOS or BIPOLAR circuits, a CMOS circuit has almost no static power dissipation. Power is only dissipated in case the circuit actually switches. This allows to integrate many more CMOS gates on an IC than in NMOS or bipolar technology, resulting in much better performance.

***