EC 705 Integrated Circuit Design Lab

M Tech VLSI Design

Design a 4-bit gray counter using static CMOS logic

Mini Project

By

JAYPAL RATHOD (212VL016)

DIVYESH SAWANT (212VL031)

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

ABSTRACT

A Gray code counter which has an iterative and relatively simple structure is described. The code is shown to be the reflected binary Gray code, implying simple conversion of the count into binary code. Gray Codes have many practical applications that go beyond research interests.

Counter is a digital device and the output of the counter includes a predefined state based on the clock pulse applications. The output of the counter can be used to count the number of pulses. Generally, counters consist of a flip-flop arrangement which can be synchronous counter or asynchronous counter. Counters have a primary function of producing a specified output sequence and are thus sometimes referred to as pattern generators.

INTRODUCTION

Gray code was invented by Frank Gray of Bell Labs in 1947. He called his code as reflected binary code. Later others started calling it as Gray code. This code has also come to be known as mirror code. Both the names, viz., reflected binary code and mirror code, stem from its mirrorreflection property.

The main features of the Gray code are:

• It is a binary code just like to the conventional binary code.

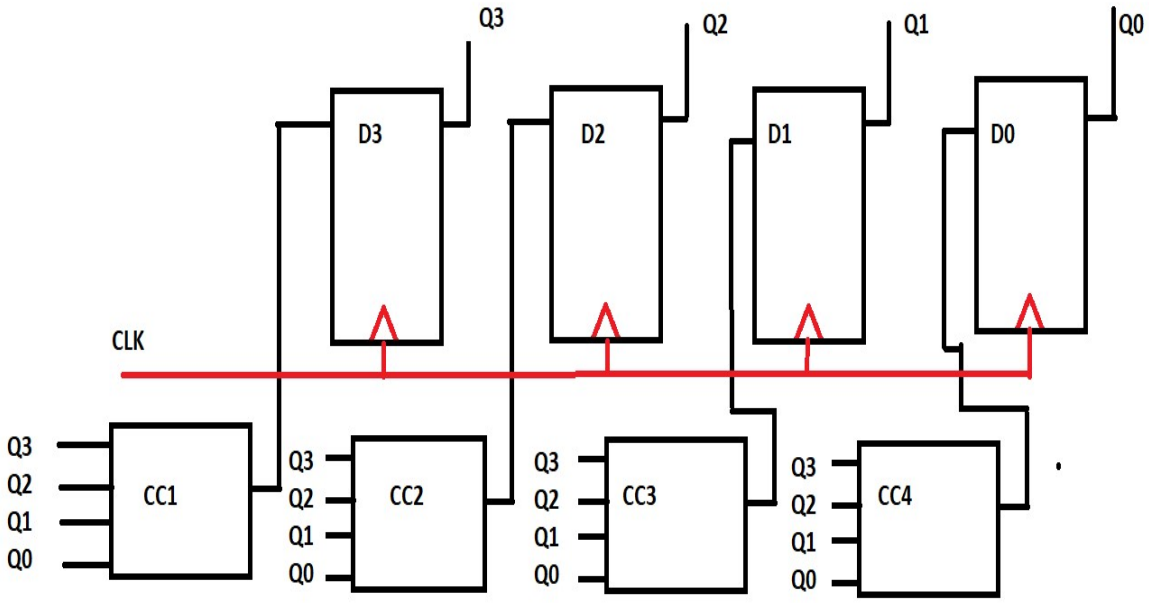
• The code can be constructed using its mirror-reflecting property.

• Adjacent numbers of the code differ in one bit-position only.

The reflected Binary Code (RBC) is the unfamiliar and original name of Gray Code. As the original name implies, the values of the second half of a GREY CODE set are equivalent to the first half but in reverse order. This with the exception for the most significant bit (MSB) which is inverted i.e. 0 changes to 1. GREY CODEs were originated when digital logic circuits were built around vacuum tubes and electromechanical relays. And thus, counters were generating enormous power demands and noise spikes when many bits changed at the same time. A GREY CODE is a binary code with code-words having unitary Hamming distance between each other of the codes. A GREY CODE of length n bits is represented by all the integers from 0 to 2n -1 in the forms of bit patterns (called “words”) of length n. Consecutive words in a GREY CODE set includes the adjacency

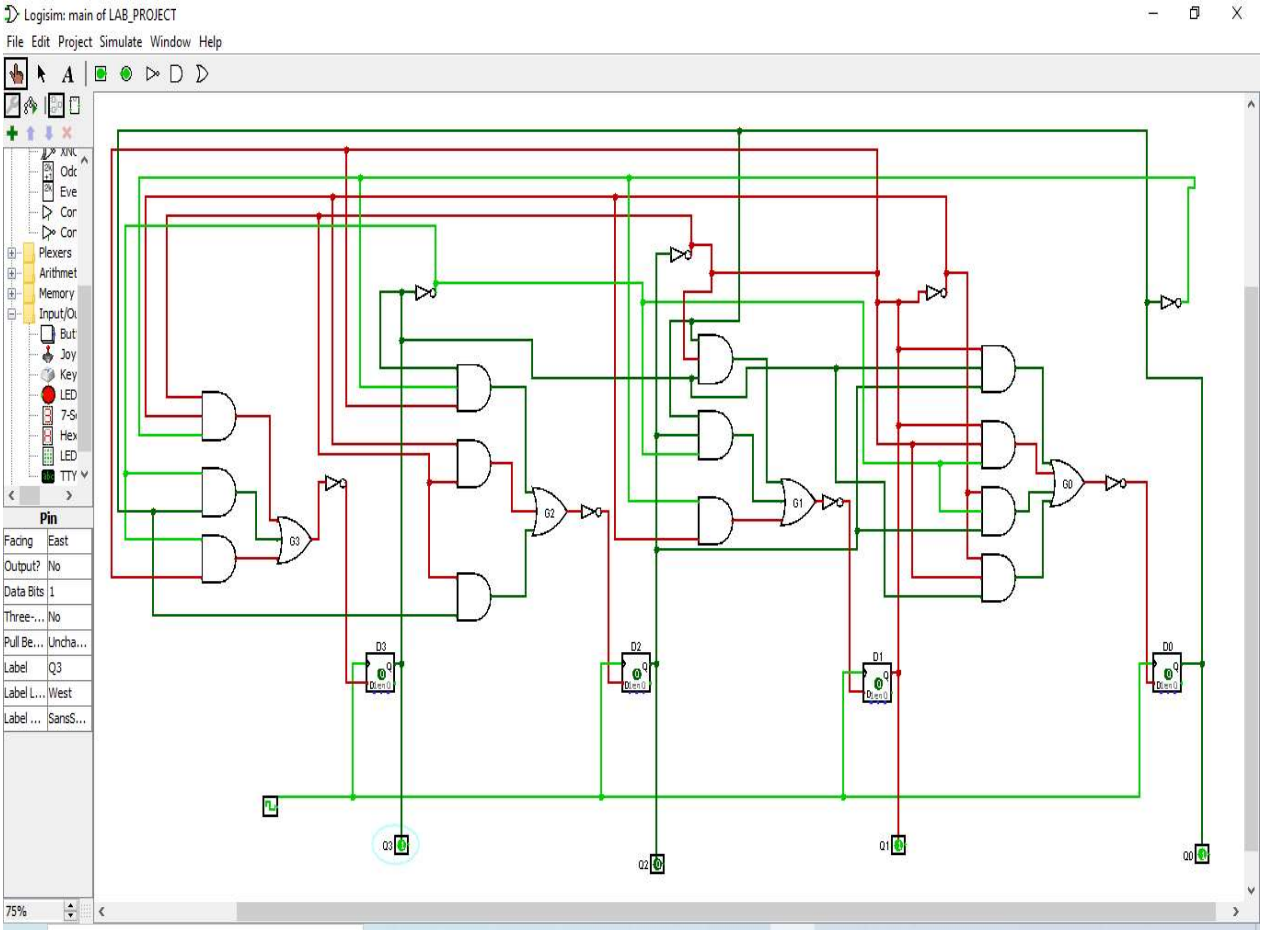
Circuit Diagram :

Gray Code Counter

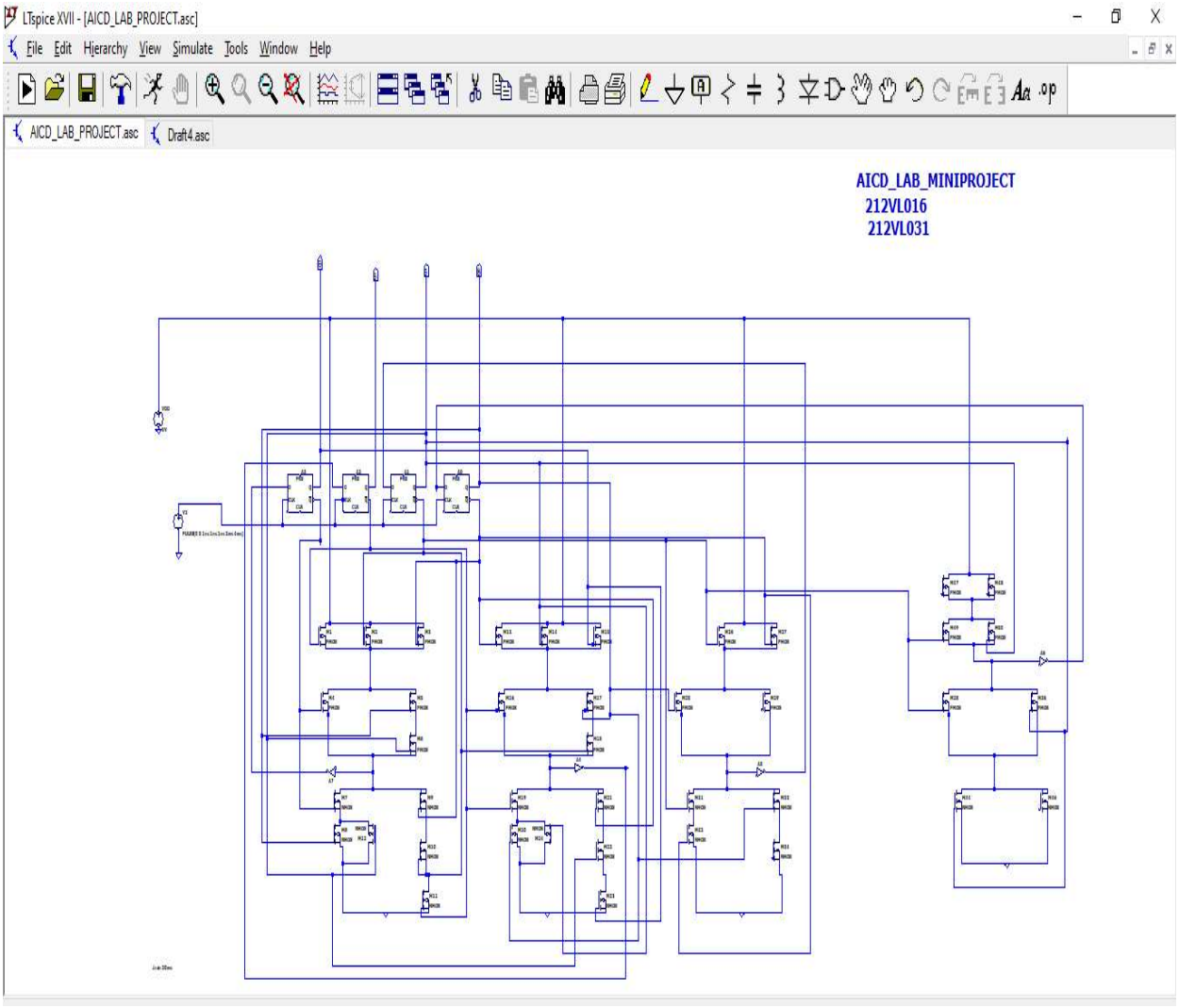


Here we have used D Flips Flops for clock mechanism and combinational circuits to implement the Boolean expression for various outputs of the flip flop.

Circuit is Verified Using LOGISIM



Circuit Diagram of 4 bit gray counter using static CMOS logic in LTSPICE



TRUTH TABLE

Present State				Next State				D3	D2	D1	D0
Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0				
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	0	1	0
0	0	1	0	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	0	0	1	0	0
0	1	0	0	1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	1
1	0	1	1	1	0	0	1	1	0	0	1
1	0	0	1	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0

Combinational Circuit Blocks Design using K- Map:

A)

Implementing D3 compliment so that we will get D3 by using static CMOS logic implementation **COMBINATIONA CIRCUIT 1 (CC1)**

$$D3' = Q3'(Q1 + Q0) + Q2'Q1'Q0'$$

$D3'$ implementation using K-map

Q1Q0 / Q3Q2	00	01	11	10
00	1	1	1	1
01		1	1	1
11				
10	1			

B) Implementing D2 compliment so that we will get D2 by using static CMOS logic implementation **COMBINATIONA CIRCUIT 2 (CC2)**

$$D2' = Q2'(Q0 + Q1') + Q3Q1Q0'$$

$D2'$ Implementation using K-map

Q1Q0 / Q3Q2	00	01	11	10
00	1	1	1	
01				
11				1
10	1	1	1	1

C) Implementing D1 compliment so that we will get D1 by using static CMOS logic implementation **COMBINATIONA CIRCUIT 3 (CC3)**

$$D1' = Q1'Q0' + Q3'Q2Q0 + Q3Q2'Q0$$
$$= Q1'Q0' + Q0X$$

Here X is Q3 XOR Q2.

$D1'$ implementation using K-map

Q1Q0 / Q3Q2	00	01	11	10
00	1			
01	1	1	1	
11	1			
10	1	1	1	

D) Implementing D0 compliment so that we will get D0 by using static CMOS logic implementation **COMBINATIONA CIRCUIT 4 (CC4)**

$$D0' = Q1(Q3Q2 + Q3'Q2') + Q1'(Q3'Q2 + Q3Q2')$$
$$= Q1X' + Q1'X$$

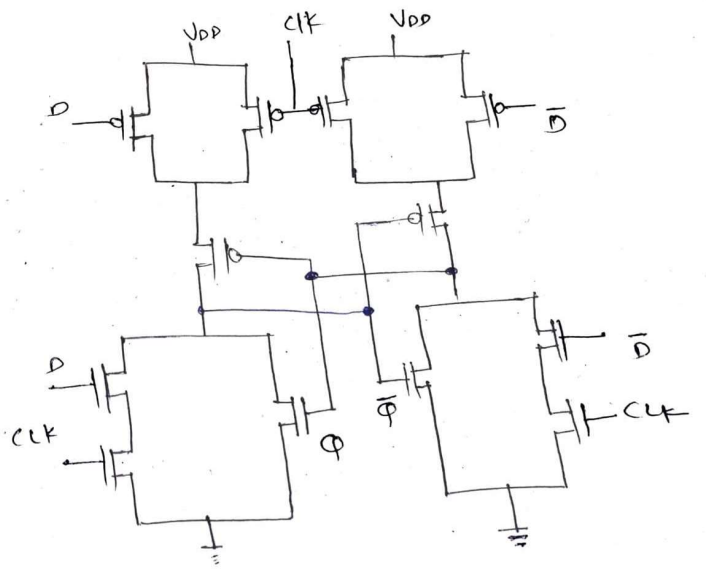
Here we have taken an XOR gate as X = Q3 XOR Q2

$D0'$ implementation using K-map

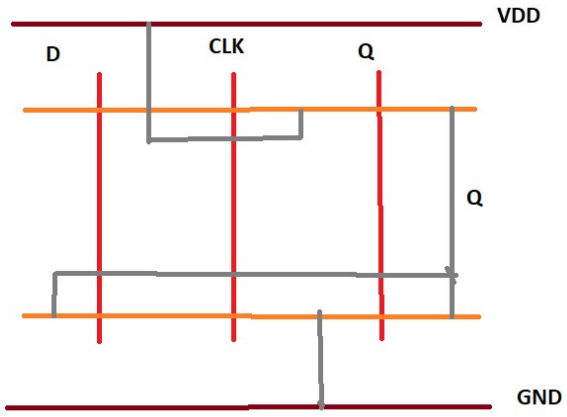
Q1Q0 / Q3Q2	00	01	11	10
00			1	1
01	1	1		
11			1	1
10	1	1		

FF And Combinational ckt used for static CMOS implementation :

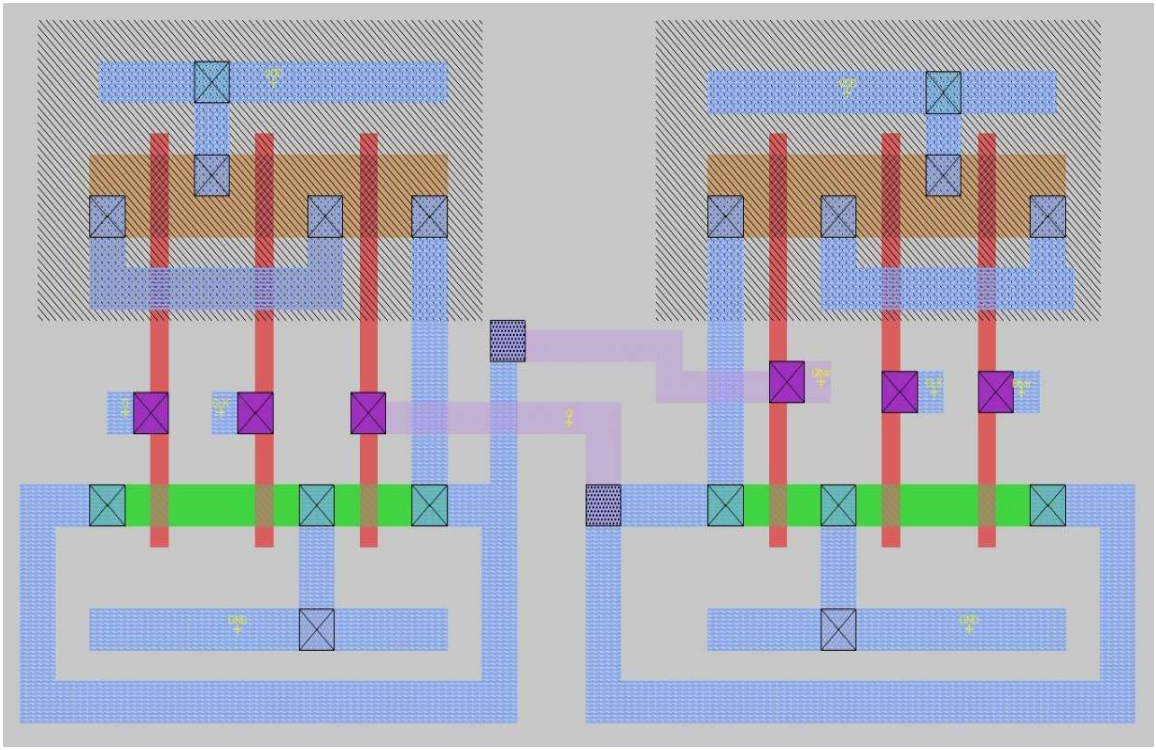
Static CMOS logic implementation of D-FF



Stick diagram of D-FF

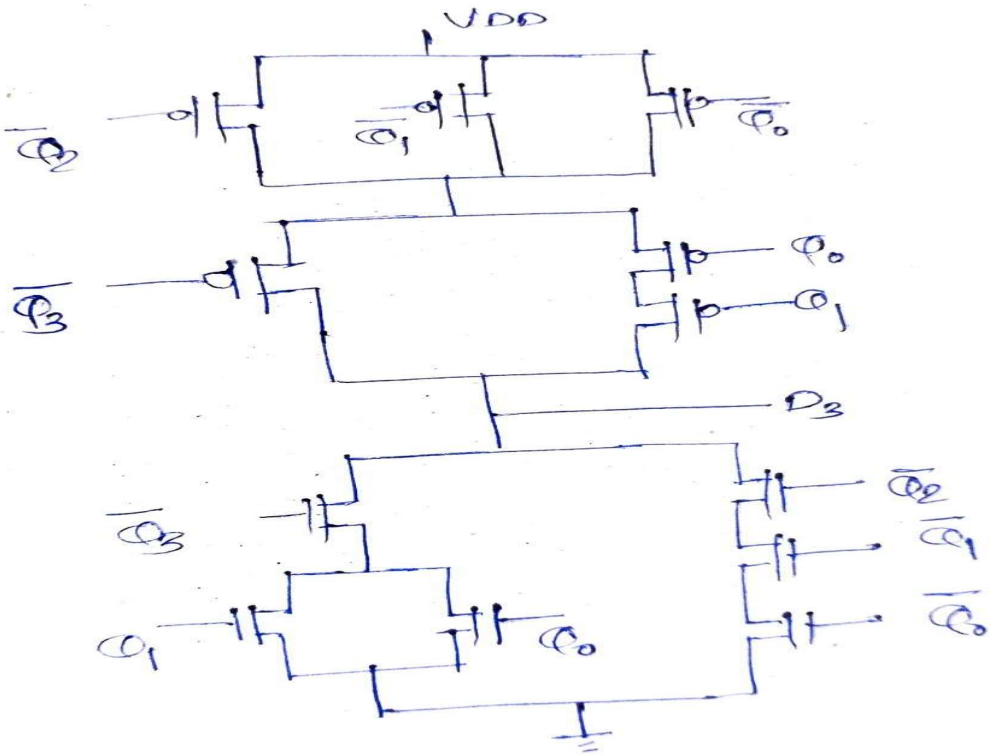


Magic Layout for D Flip Fop:

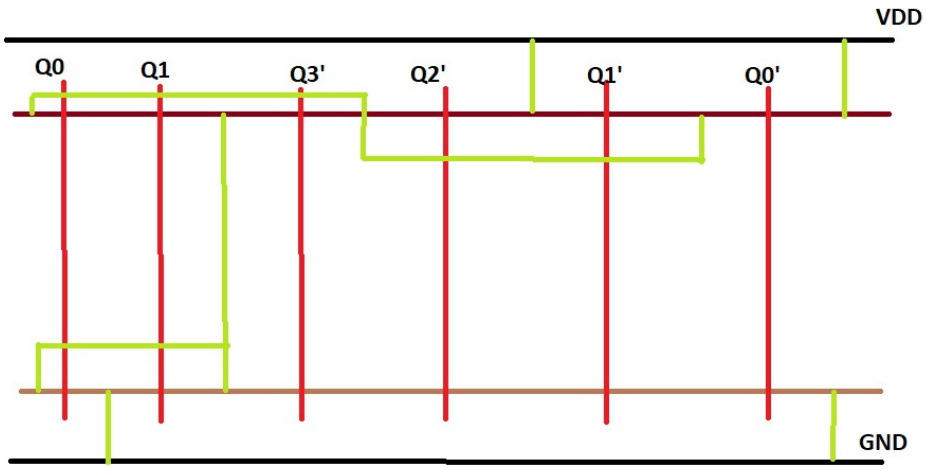


Combinational Circuit 1 (CC1) or D3:

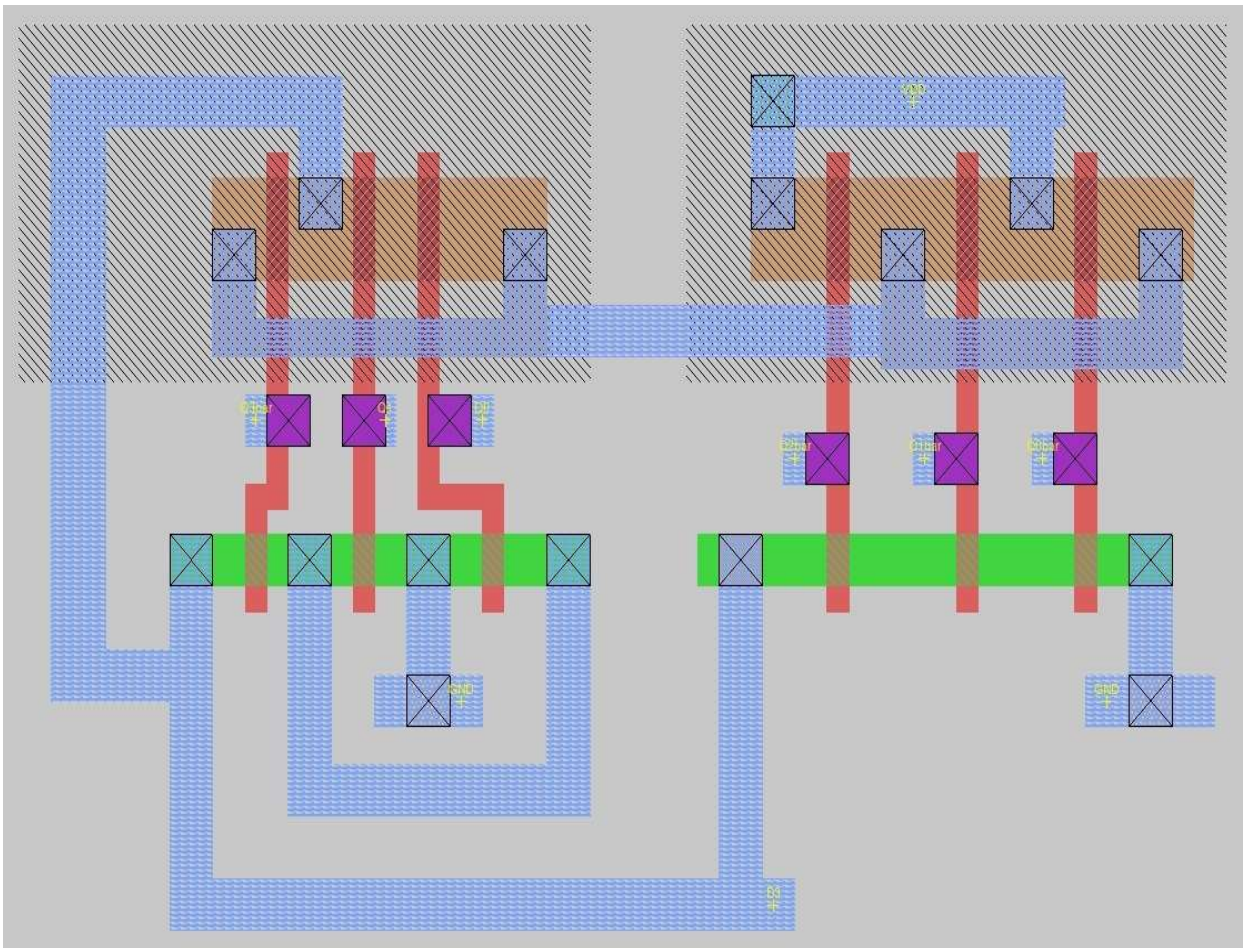
Static CMOS logic implementation of Combinational Circuit 1



Stick diagram of Combinational Circuit 1

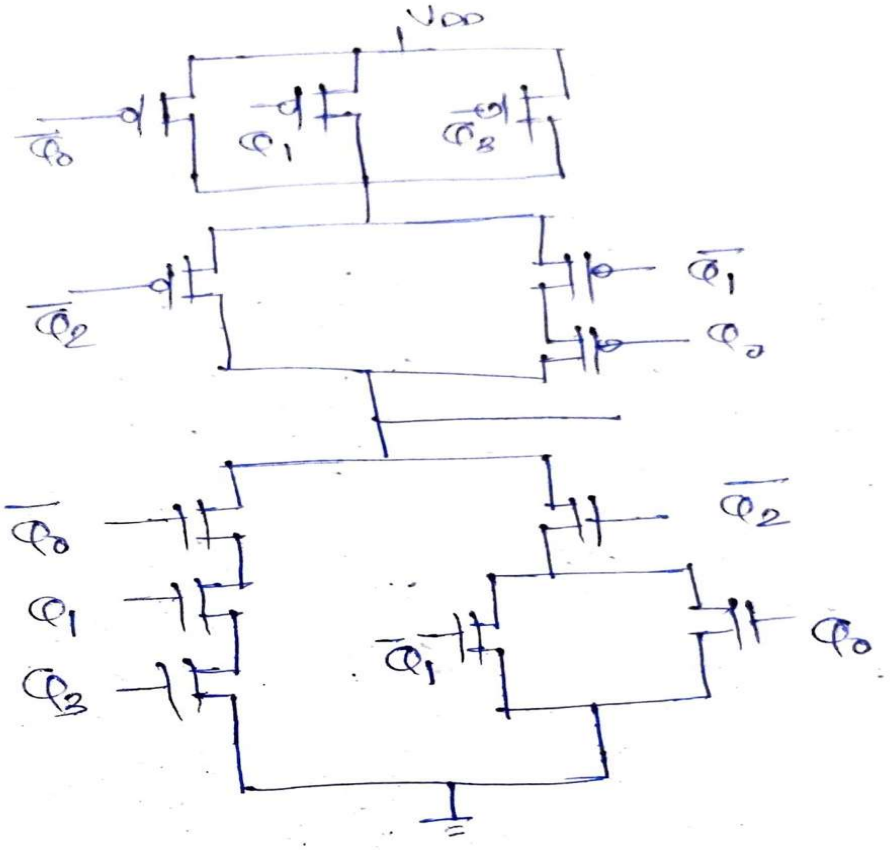


Magic Layout for D3:

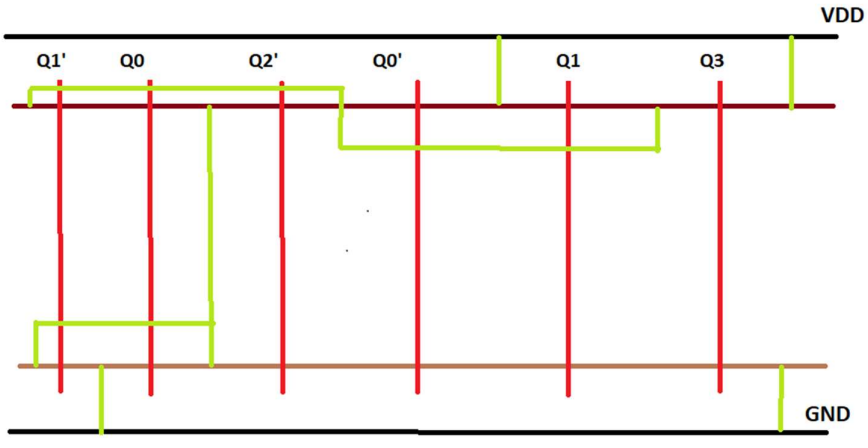


Combinational Circuit 2 (CC2):

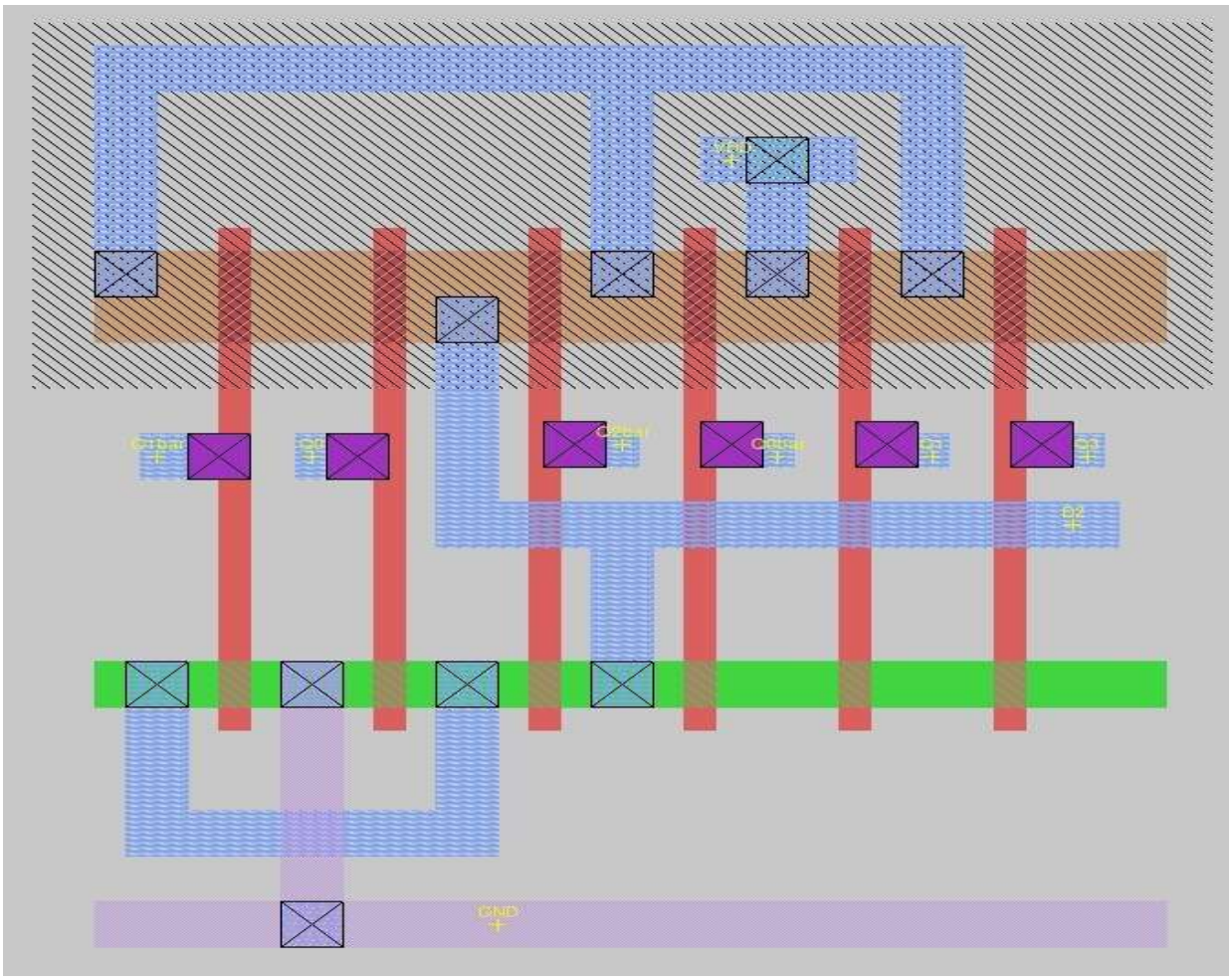
Static CMOS logic implementation of Combinational Circuit 2



Stick diagram of Combinational Circuit 2

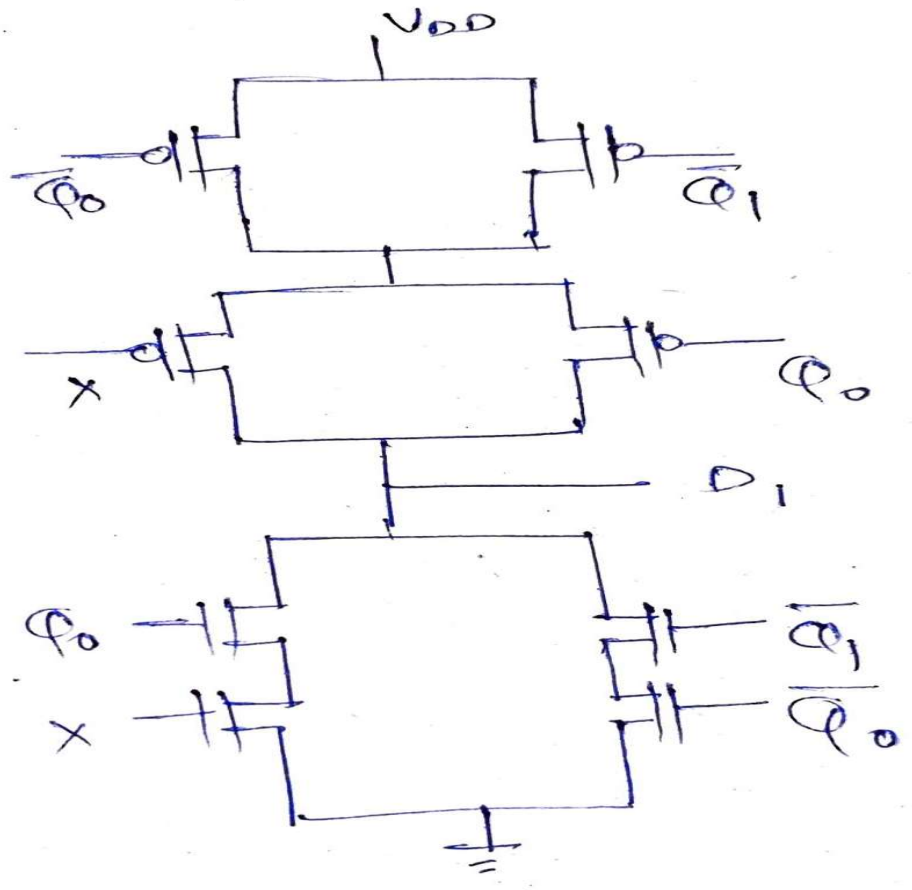


Magic Layout for D2:

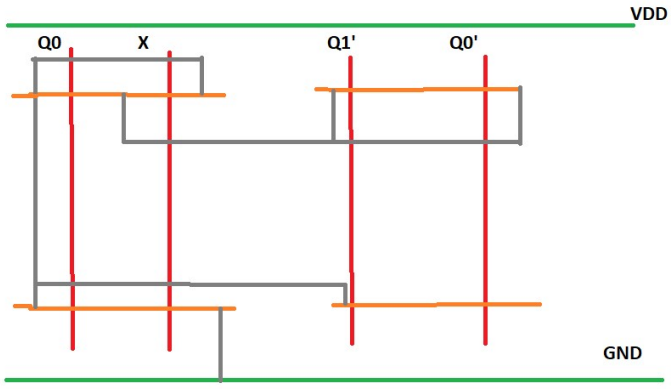


Combinational Circuit 3 (CC3):

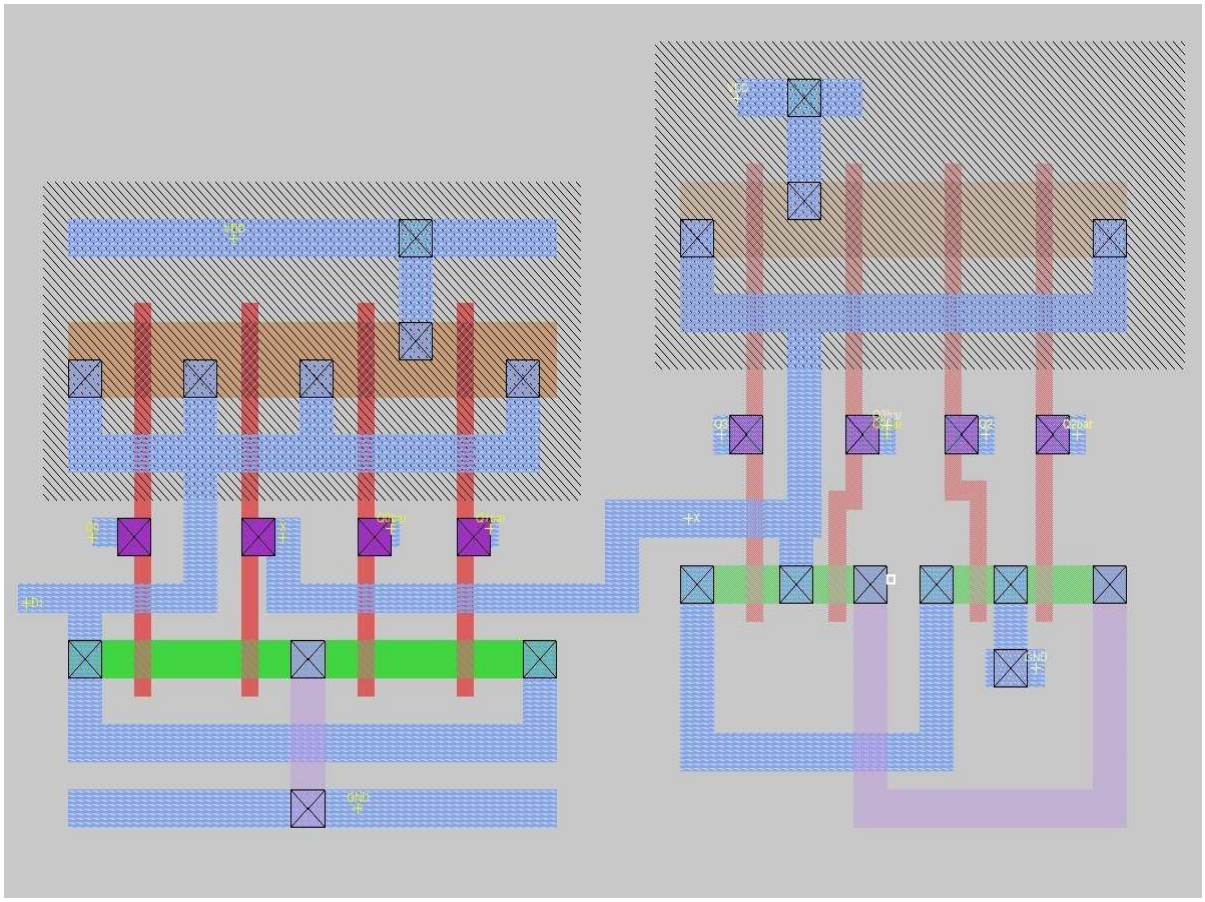
Static CMOS logic implementation of Combinational Circuit 3



Stick diagram of Combinational Circuit 3

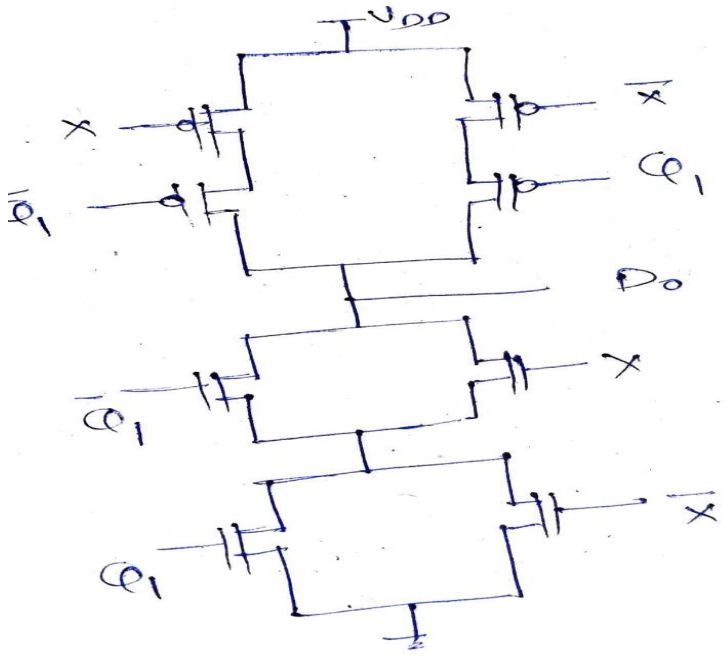


Magic Layout for D1:

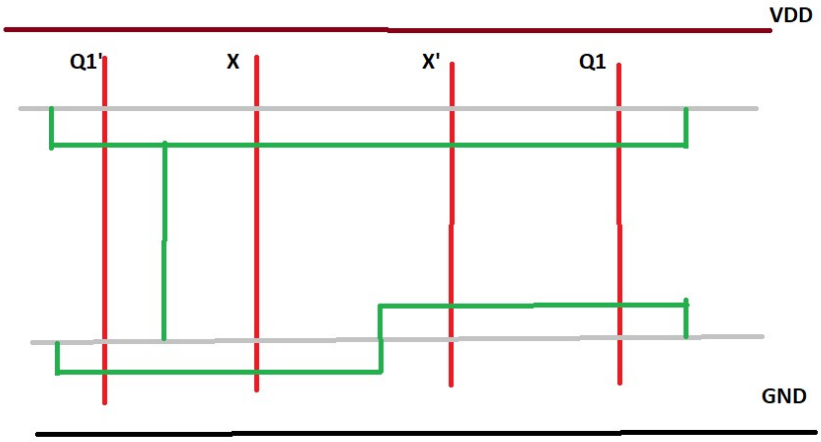


Combinational Circuit 4 (CC4):

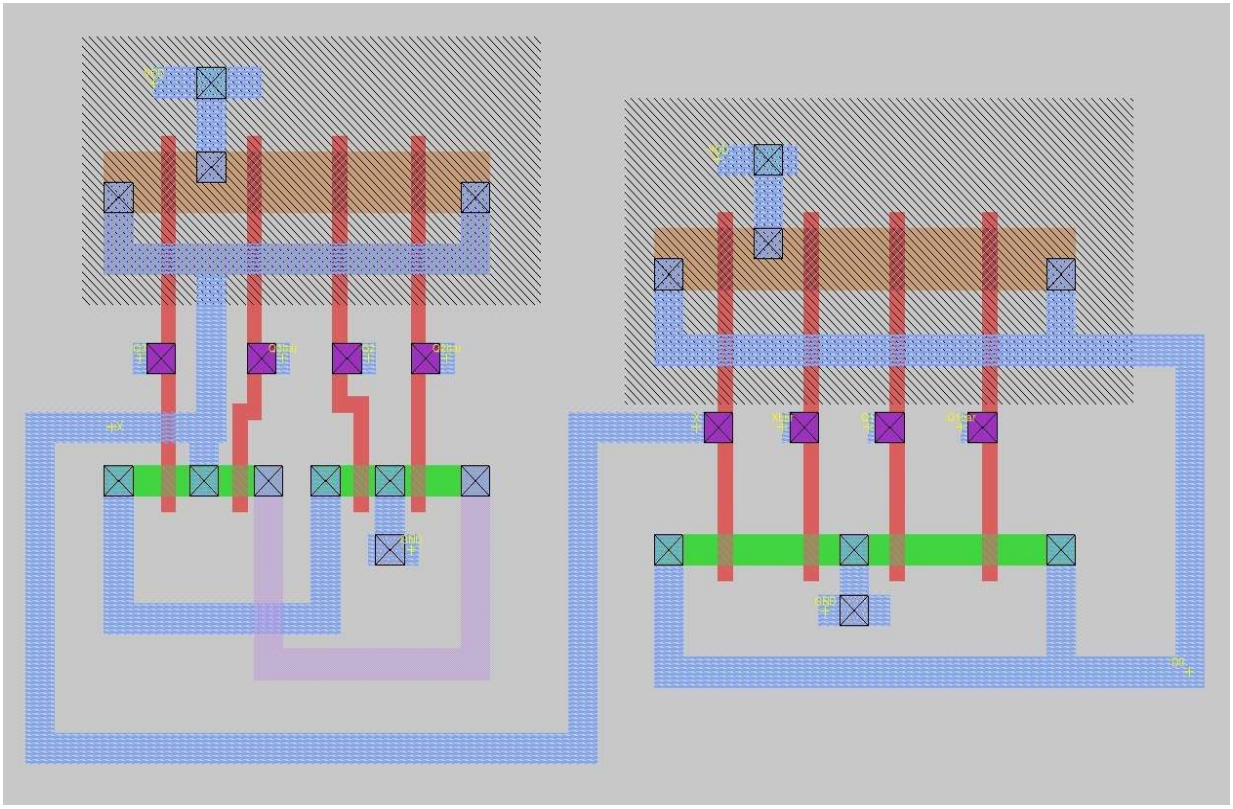
Static CMOS logic implementation of Combinational Circuit 4



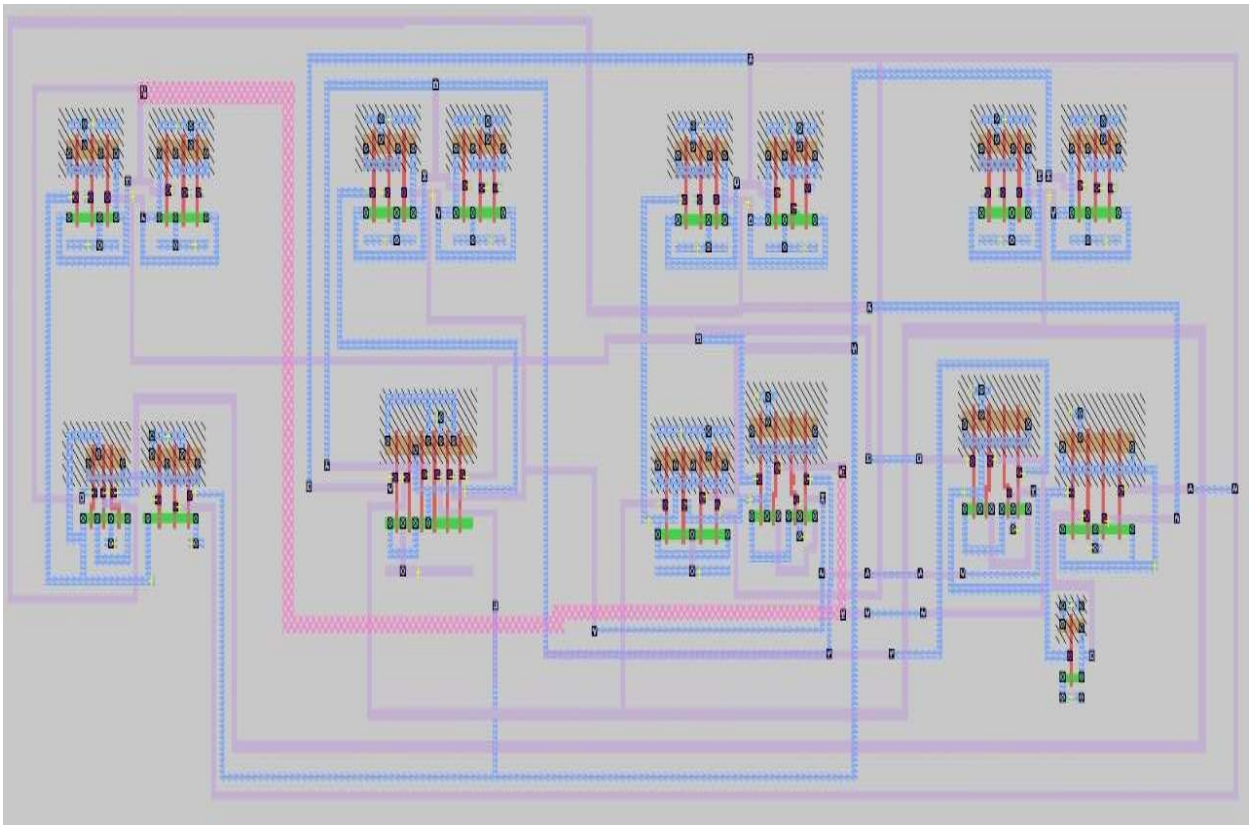
Stick diagram of Combinational Circuit 4



Magic Layout for D0:

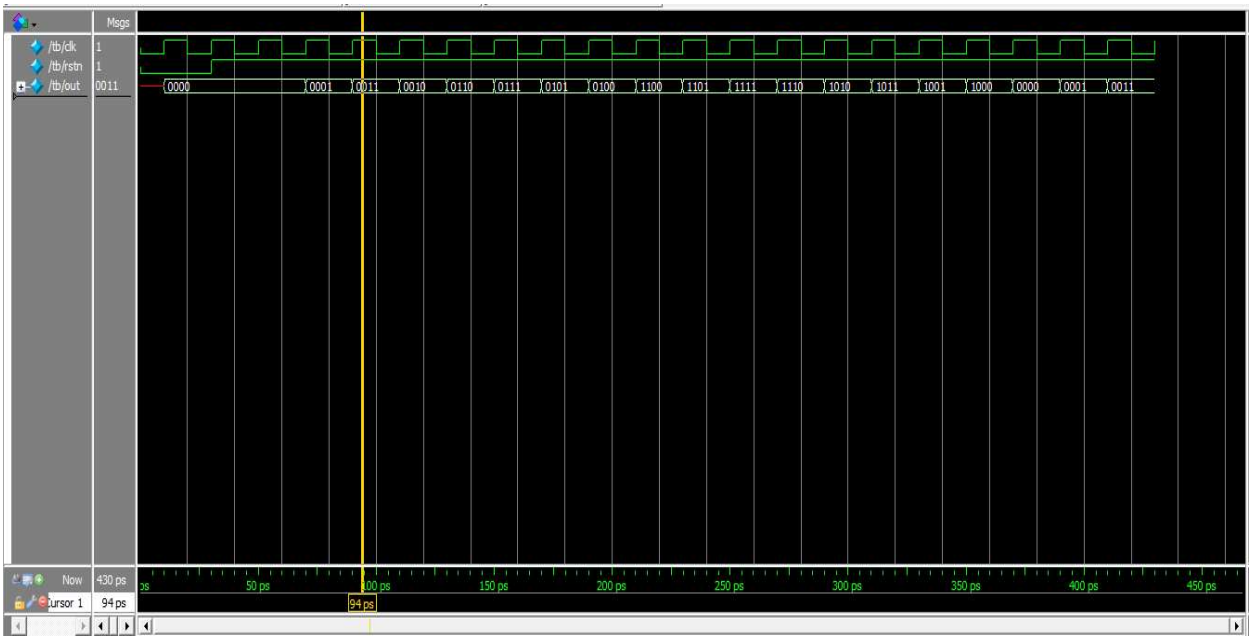


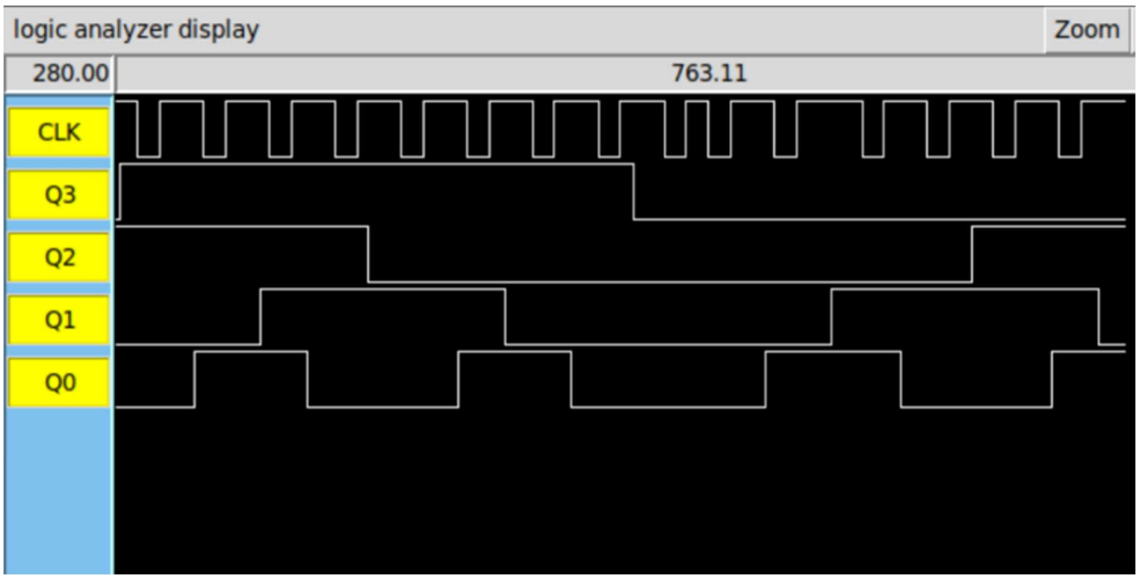
Complete Layout After Interconnections:



COMPLETE LAYOUT OF 4-BIT GRAY COUNTER USING STATIC CMOS LOGIC IMPLEMENTATION

LAYOUT SIMULATION OUTPUT :





The initial state was given as 1100 with Q3 as MSB then the sequence is followed as:

1100, 1101,1111,1110,1010,1011,1001,1000,0000,0001,0011,0010,0100,0110,0111

Applications of Gray Code:

- Boolean circuit minimization
- Communication between clock domains
- Error correction
- Genetic algorithms
- Mathematical puzzles

Advantages of Gray Code:

- Better for error minimization in converting analog signals to digital signals
- Reduces the occurrence of “Hamming Walls” (an undesirable state) when used in genetic algorithms
- Can be used to in to minimize a logic circuit
- Useful in clock domain crossing

Disadvantages of Gray Code:

- Not suitable for arithmetic operations
- Limited practical use outside of a few specific applications.

REFERENCES:

1. Solved 1. Design the 3-bit synchronous Gray code counter | Chegg.com

2. flipflop - 3bit Gray counter using D Flip Flops and logic gates - Electrical Engineering Stack Exchange

.