Brac University

EEE 412/ ECE 412/ CSE 460

VLSI Design Laboratory

Experiment 5

Part 1: Introduction to CMOS Technology and simulation of combinational circuits using DSCH2 Software

Theory:

Complementary metal–oxide–semiconductor (CMOS), also known as complementary-symmetry metal–oxide–semiconductor (COS-MOS), is a type of MOSFET (metal–oxide–semiconductor field-effect transistor) fabrication process that uses complementary and symmetrical pairs of p-type and n-type MOSFETs for logic functions.

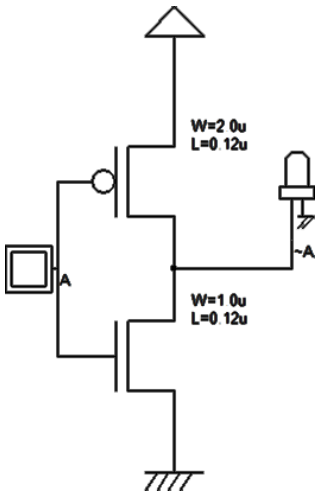
Pull-up Network: The network or combination of PMOS that is responsible for pulling the output up to logic state 1 or HIGH is called the pull-up network.

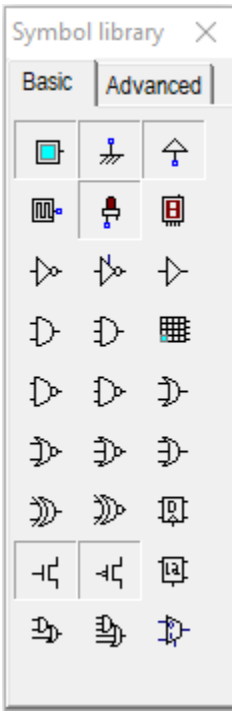
Pull-down Network: The network or combination of NMOS that is responsible for pulling the output down to logic state 0 or LOW is called the pull-up network.

Basic Inverter:

In an inverter circuit, the output is of the opposite logic state than the input. Here if the input A is HIGH, it turns the NMOS on and PMOS off. Thus the pull-down network becomes active and output is LOW. Again turning the input LOW turns on the PMOS and turns off the NMOS and thus the pull-up network is active and output is HIGH.

For simulating a basic inverter using DSCH2, drag down NMOS, PMOS, VCC, GND, Button and LED from symbol library and connect them as the figure. The LED shows the output state and button provides the HIGH/LOW input.





Converting a circuit to a block:

There are certain parts of a large circuit which may need to be repeated over and over. For those we can make blocks or sub-circuits from the CMOS Design that perform the same tasks. For doing so, we need to build the CMOS circuit and simulate it to check if the input output relations shown are correct. Then we choose file>schema to new symbol. We change the name and title according to the circuit operation and press ok.

NOTE: Using names that are already used by the software itself often cause erroneous results. So the name and title must be kept unique. For example, an unique title and name may be

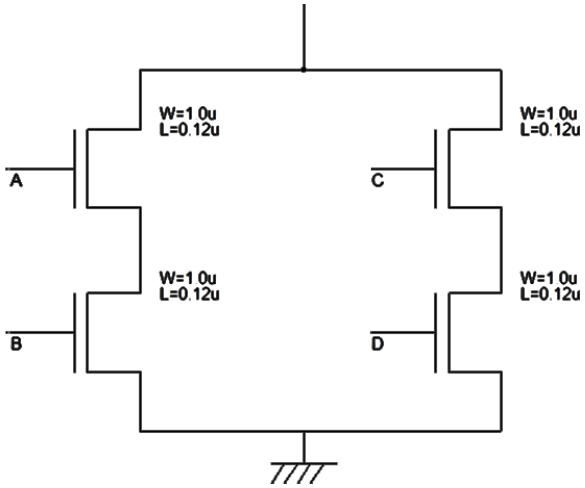
Inverter_(Student ID). Special characters and numbers at the beginning of name/title have to be avoided.

Designing CMOS circuit from equation:

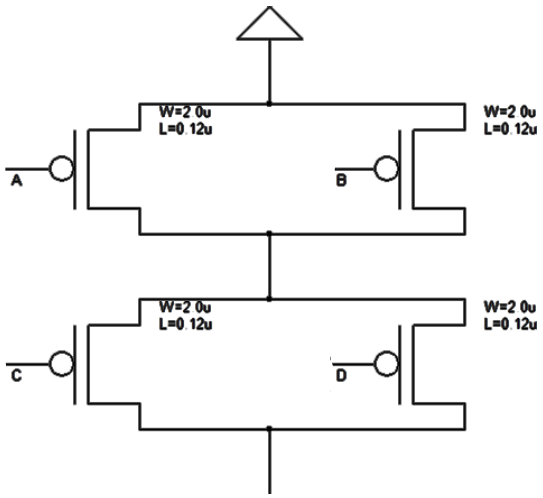
Steps:

1. Identify the AND, OR and NOT logic functions in the equation.

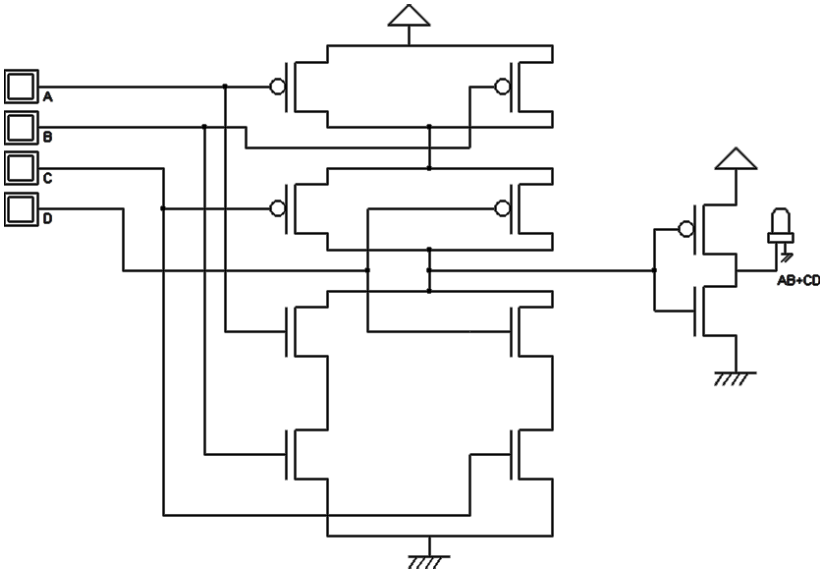
2. Design the pull-down network using NMOS. AND logics are placed in series and OR logics are placed in parallel. For example: say F=AB+CD will have the following pull-down network:



3. Above the pull down network, place the pull up network of PMOS. The series components of pull-down network becomes the parallel components of pull-up network and vice versa. For example: say F=AB+CD will have the following pull-up network:



4. Put an inverter with its input in the junction between the pull-up and pull-down networks and output as the output of the combinational circuit. For example: CMOS implementation of F=AB+CD will look like the following:



Labwork:

1. Design NAND and NOR gate using CMOS.

2. Implement a 4 to 2 priority encoder of priority 3>2>1>0 using CMOS.

Homework:

1. Design AND, OR and XOR gates using CMOS.

2. Design a 4 bit adder using CMOS technology. You may use blocks/sub-circuits made using CMOS technology but cannot use readily available logic gates.

3. Design a priority encoder of priority 2>3>0>1 using CMOS Technology.

Brac University

EEE 412/ ECE 412/ CSE 460

VLSI Design Laboratory

Experiment 5

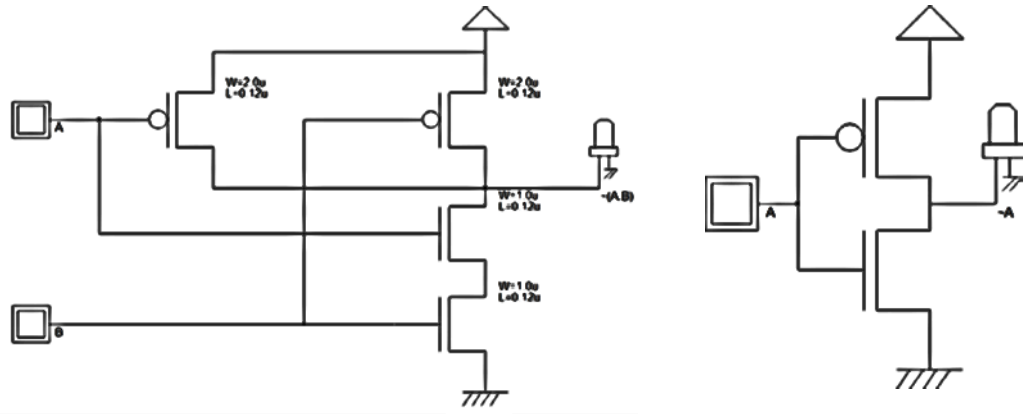
Part 2: Simulation of sequential circuits using DSCH2 Software

Theory:

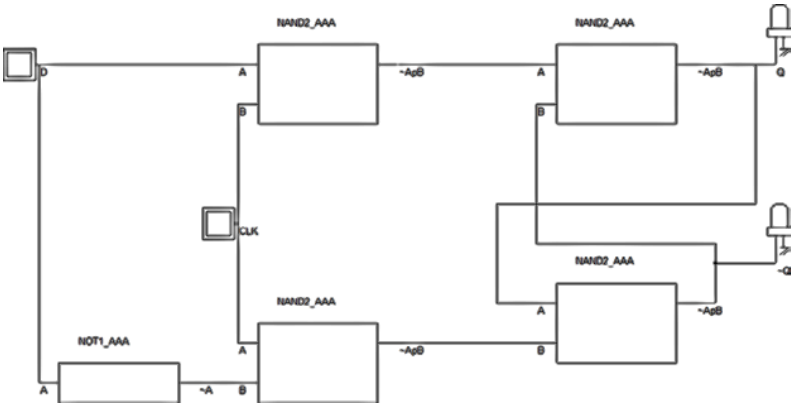
A sequential circuit is a logical circuit, where the output depends on the present value of the input signal as well as the sequence of past inputs. While a combinational circuit is a function of present input only. A sequential circuit is a combination of combinational circuit and a storage element.

The basic elements of sequential circuits are Latch and Flip-flops. They are the memory/storage elements. The basic difference between latch and flipflops are that latch are level triggered and flipflops are edge triggered. In case of D-latch, when the clock is high, the data in the input is sent to the output and when clock is low, the output holds its previous value. In case of D-Flipflop, the data from the input moves to the output only at the positive edge or negative edge of clock. For all other time, the output stores its previous value.

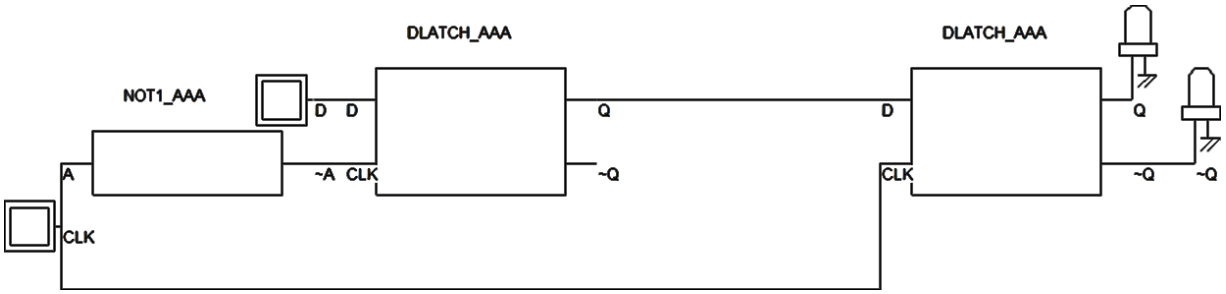
First task before we make a latch or flipflop is to make a NAND and NOT gate using CMOS technology.



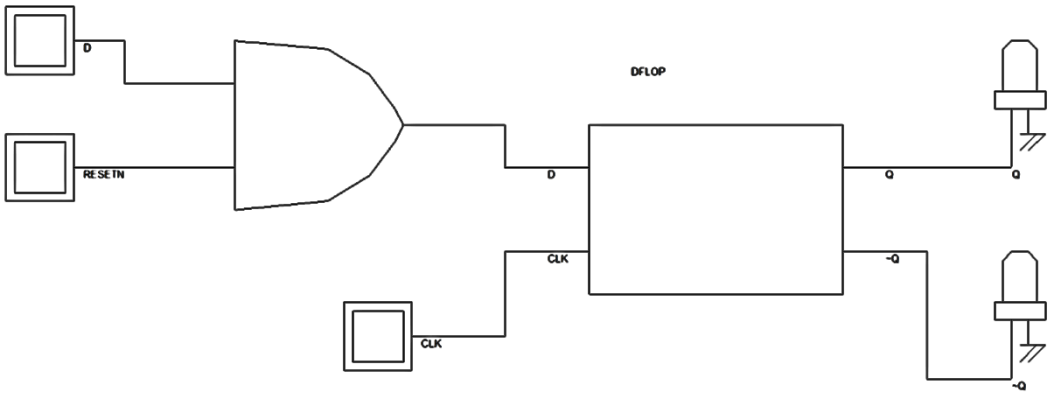
We then build a block of the gates and make the latch using the following circuit connection:



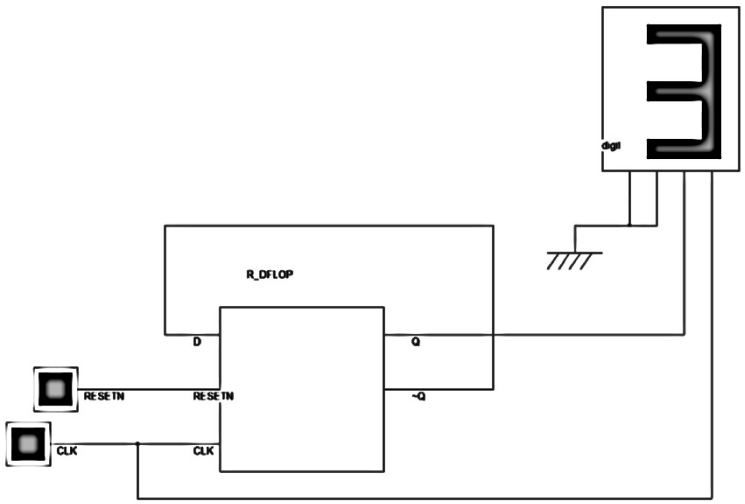
We can observe that the circuit is working as a level triggered latch. We then make a block out of the circuit. Then combining two latches, we build an edge-triggered D-Flipflop.



We now want to add reset functionality to our flipflop. We can do that by using an AND gate before the flipflop. Observing the following circuit, we can see that, when RESETN = 0, output of the AND gate will be 0, no matter what the value of input signal D is. So, during the next positive edge of the clock, value of output, Q will be reset to 0.



Finally, we want to build a 2-bit down counter using the flipflop with reset functionality. To do that, we make the flipflop with reset functionality into a new block and construct the following circuit to implement the counter:

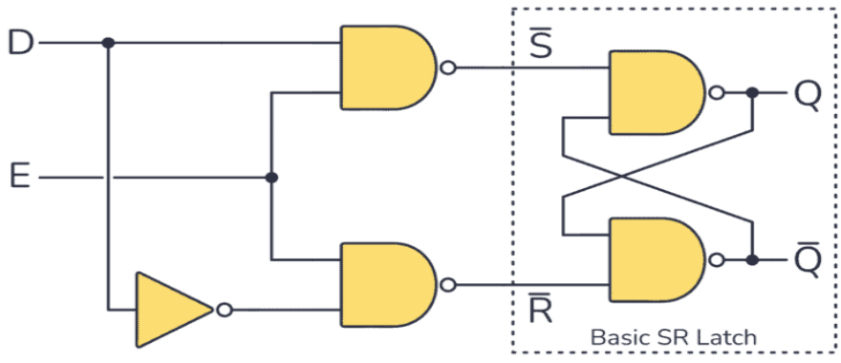


Labwork:

1. Design NAND and NOT gate using CMOS.
2. Design a DLatch.
3. Design a D Flip-flop with set and reset operations
4. Design a Johnson counter

Homework:

1. Design an end round shift register.
2. Design a 4 bit counter.
3. Design a 2 to 10 counter.



D Latch Circuit