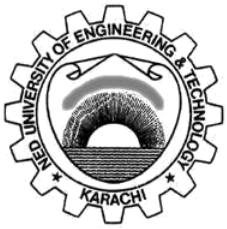


Department of Electronic Engineering
NED University of Engineering & Technology

PRACTICAL WORK BOOK

For the course

VLSI SYSTEM DESIGN

(EL-408) For B.E(EL)

Instructor's Name:

Student Name:

Roll no: **Batch:**

Semester: **Year:**

Department:

LABORATORY WORK BOOK
FOR THE COURSE

VLSI System Design (EL-408)

Prepared By:

Ms. Mariyum Jamshid (Lecturer)

Reviewed By:

Ms. Madiha Mazhar (Lecturer)

Approved By:

The Board of Studies of Department of Electronic Engineering

VLSI System Design Laboratory
CONTENTS

LAB SESSION NO 01

Objective:

To practice the following digital logic circuits:

a. **NMOS Inverter circuit with resistive load**
b. **NMOS AND, OR gates with resistive load**
c. **Use pass transistors to implement 2 inputs AND**

Apparatus:

- MOSFET transistors 2N7000
- Connecting Wires
- Digital Multimeter
- Proto-board

Theory:

Resistive Load Inverter

The basic structure of a resistive load inverter is shown in the figure given below. Here, enhancement type nMOS acts as the driver transistor. The load consists of a simple linear resistor R_L. The power supply of the circuit is $V_{DD.}$

Circuit Operation

When the input of the driver transistor is less than threshold voltage V_{TH} ($V_{in} < V_{TH}$), driver transistor is in the cut – off region and does not conduct any current. So, the voltage drop across the load resistor is ZERO and output voltage is equal to the V_{DD}.

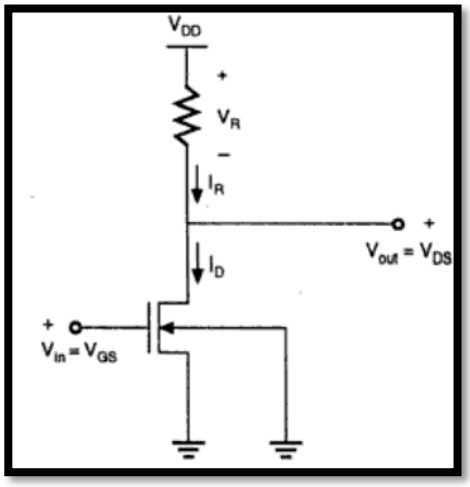


Figure 1.1: Structure of resistive load inverter

Pass Transistor

A transistor used as a switch to pass logic levels between nodes of a circuit, instead of as a switch connected directly to a supply voltage.

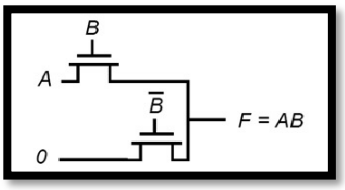


Figure 1.2: Design with Pass transistor logic

Procedure:

Part a:

Connect the circuit on breadboard according to the given figure for implementing NMOS inverter.

Observations:

Vin	Vout
0V	
5V	

Part b:

For implementing AND gate, connect two transistors in series and for OR gate, connect two transistors in parallel.

Observations:

AND GATE			OR GATE		
Vin		Vout	Vin		Vout
0V	0V		0V	0V	
5V	0V		5V	0V	
0V	5V		0V	5V	
5V	5V		5V	5V	

Part c:

Connect the circuit as shown in figure and implement AND operation.

Observations:

NED University of Engineering & Technology
Department of ____**ELECTRONIC**__ Engineering
Course Code and Title: ____**EL-408 VLSI System Design**_____

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to ***identify*** equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to ***demonstrate*** the use of the equipment for the lab work.	Doesn't demonstrate the use of equipment.	Slightly demonstrates the use of equipment.	Somewhat demonstrates the use of equipment.	Moderately demon strates the use of equipment.	Fully demonstrates the use of equipment.
Procedural Skills ***Displays*** skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to ***imitate*** the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to use the observations from lab work for experimental verifications and illustrations.	Not able to use the observations from lab work for experimental verifications and illustrations.	Slightly able to use the observations from lab work for experimental verifications and illustrations.	Somewhat able to use the observations from lab work for experimental verifications and illustrations.	Moderately able to use the observations from lab work for experimental verifications and illustrations.	Fully able to use the observations from lab work for experimental verifications and illustrations.
Safety Adherence Adherence to ***safety*** procedures.	Doesn't adhere to safety procedures.	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling ***Equipment care*** during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care.	Often handles equipment with required care.	Handles equipment with required care.
Group Work ***Contributes*** in a group based lab work.	Doesn't participate and contribute.	Slightly participates and contributes.	Somewhat participates and contributes.	Moderately participates and contributes.	Fully participates and contributes.

Laboratory Session No. _____01_____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

LAB SESSION NO 02

Objective:

To operate under supervision the operation of Digital-to-Analog and Analog-to-Digital Convertor.

Apparatus:

- 2-bit priority encoder IC
- Resistor sheet
- LEDS
- Connecting Wires
- Digital Multimeter
- Bread Board
- 7408 IC (AND)
- 7432 IC (OR)
- 7486 IC (XOR)
- 4049 IC (INVERTER)
- LF351 OPAMP
- Switches
- Protoboard

Theory:

ANALOG TO DIGITAL CONVERTER:

An analog-to-digital (A/D) conversion means quantizing the amplitude of a physical quantity (e.g., a voltage) into a discrete levels class. Thus, obtaining a series of digits, forming a number of a proper code. Generally, the binary code and, consequently, binary numbers are used. Analog data can be obtained again through digital-to-analog (D/A) conversion.

Due to the quantization, each value V of the analog signal included within the interval V_i to V_{i+1} is always quantized at the same level N_i.

The interval: V_{i+1} to V_1 = Q, is defined as "quantum level".

Another important parameter of A/D converters is the conversion time since it defines the capacity of the converter to operate the conversion of a variable signal; in fact, remember that the sequence of the quantized levels must allow the regeneration of the original analog signal.

A time-variable signal can be converted into a discrete values class carrying out the sampling and holding operations. The sampling and holding operations are carried out through proper circuits called "Sample and Hold".

Resolution

It defines the smallest standard incremental change in. the output voltage of a DAC or the amount of input voltage change required to increment the output of an ADC between a code change and the next adjacent code change. A converter with "n" switches can divide the input in 2^n parts: the

least significant increment is then 2^{-n}, or one least significant bit (LSB). On the contrary the Most Significant Bit carries a weight of 2. Resolution is applied to DACs and ADCs and may be expressed in percent of full scale or in binary bits.

Circuit Operation:

2-bit Analogue to Digital Converter Circuit

In general, 2^n-1 comparators would be required for conversion of an "n"-bit binary output, where "n" is typically in the range from 8 to 16. If we create a 2-bit ADC, then we will need 2^2– 1 which is "3" comparators as we need four different voltage levels corresponding to the 4 digital values required for a 4-to-2-bit encoder circuit as shown.

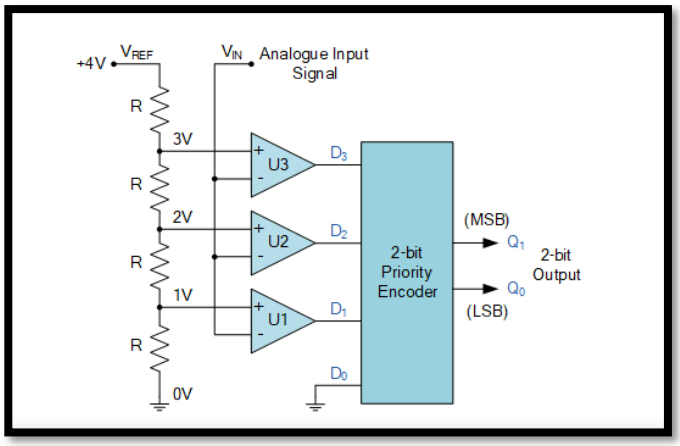


Figure 2.1: 2-bit analogue to digital converter circuit

Encoder alternate circuit:

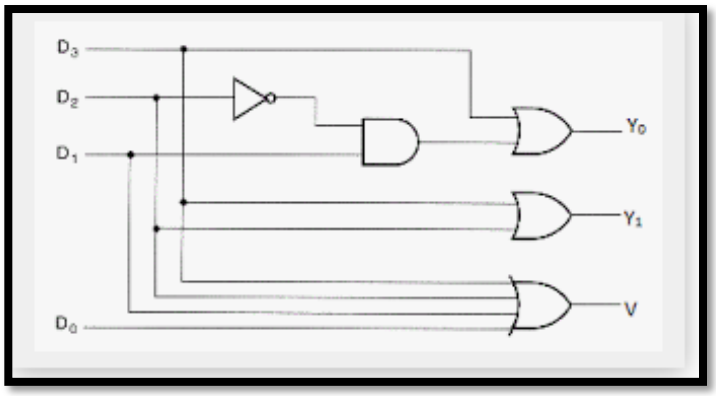


Figure 2.2: Encoder circuit

Step size= analog voltage/ no of bits

DIGITAL TO ANALOG CONVERTER:

It is often necessary to convert analog signal to an accurate digital number, and vice versa. For example, in applications where a microprocessor is controlling an experiment, the analog signal from a sensor needs to be converted into digital form so it can be communicated to the microprocessor. After the processing takes place in the digital form, the output from the microcontroller needs to be converted back to the analog form to communicate with the analog world. In this lab session we will consider the case of digital to analog conversion (DAC). A digital to analog converter (DAC) converts a digital signal to an analog voltage or current output. Many types of DACs are available and usually switches, resistors, and op-amps are used to implement the conversion.

Circuit Operation:

R-2R Ladder Digital to Analog Converter (DAC)

R-2R Ladder is another type of DAC based on the opamp summing amplifier It uses only two values of resistors which makes the fabrication of the circuit easier and more accurate. R-2R ladder can scalable to any number of bits desired and its output impedance will remain R, regardless the number of bits. Each bit corresponds to a switch.

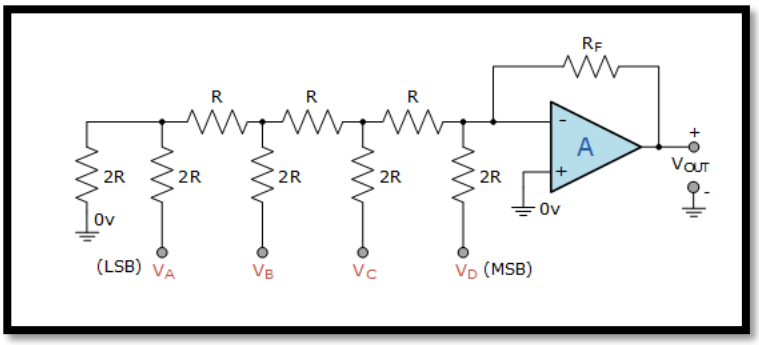


Figure 2.3: R-2R ladder digital to analogue converter circuit

If the bit is high, the corresponding switch is connected to the inverting input of the op-amp. If the bit is low, the corresponding switch is connected to ground.

bn means Bit n, hence;

If bit n is set, bn=1

If bit n is clear, bn=0

For a 4-Bit R-2R Ladder, output is equal to;

$$V_{out} = -V_{ref}\left(b_3 + b_2\frac{1}{2} + b_1\frac{1}{4} + b_0\frac{1}{8}\right)$$

In order to understand this, let's solve an example:

If we have the Vref = 2.4 V, and we need to convert the digital number 7 (which is 0111 in binary form) into analog then by applying the following equation, we get:

$$V_{out} = -V_{ref}\left(b_3 + b_2\frac{1}{2} + b_1\frac{1}{4} + b_0\frac{1}{8}\right)$$

V_{out} = - 2.4 [(0) + (1*1/2) + (1*1/4) + (1*1/8)]

V_{out} = -2.4 [0 + 1/2 + 1/4 + 1/8]

V_{out} = -2.4 * 7/8

V_{out} = -2.1 V

So, from the above example, we get that the voltage measured by the multimeter will be -2.1 V

Procedure:

a) **FOR ADC**
Connect the circuit on breadboard according to the given figure for implementing ADC converter

Observation:

The resulting truth table by implementing ADC converter circuit is as follows

ANALOG INPUT	COMPARATOR OUTPUTS				DIGITAL OUTPUTS	
VOLATGE (VIN)	D3	D2	D1	D0	Q1	Q0

b) FOR DAC

Connect the circuit on breadboard according to the given figure for implementing DAC converter.

Observation:

The resulting truth table by implementing DAC converter circuit is as follows

DECIMAL NUMERALS	DIGITAL INPUT				ANALOG OUTPUT
DECIMAL	D3	D2	D1	D0	VOUT
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

F/OBEM 01/05/00

NED University of Engineering & Technology
Department of ____**ELECTRONIC**__ Engineering
Course Code and Title: ____**EL-408 VLSI System Design**_____

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to ***identify*** equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to ***demonstrate*** the use of the equipment for the lab work.	Doesn't demonstrate the use of equipment.	Slightly demonstrates the use of equipment.	Somewhat demonstrates the use of equipment.	Moderately demon strates the use of equipment.	Fully demonstrates the use of equipment.
Procedural Skills ***Displays*** skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to ***imitate*** the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use ***Displays*** skills to use the observations from lab work for experimental verifications and illustrations.	Not able to use the observations from lab work for experimental verifications and illustrations.	Slightly able to use the observations from lab work for experimental verifications and illustrations.	Somewhat able to use the observations from lab work for experimental verifications and illustrations.	Moderately able to use the observations from lab work for experimental verifications and illustrations.	Fully able to use the observations from lab work for experimental verifications and illustrations.
Safety Adherence Adherence to ***safety*** procedures.	Doesn't adhere to safety procedures.	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling ***Equipment care*** during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care.	Often handles equipment with required care.	Handles equipment with required care.
Group Work ***Contributes*** in a group based lab work.	Doesn't participate and contribute.	Slightly participates and contributes.	Somewhat participates and contributes.	Moderately participates and contributes.	Fully participates and contributes.

Laboratory Session No. ___02_____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

LAB SESSION NO 03

Objective:

To manipulate with guidance of Full Adder circuit.

Apparatus:

- Digital multimeter
- Bread Board
- 7408 IC (AND)
- 7432 IC (OR)
- 7486 IC (XOR)
- LEDS
- Resistor sheet
- Connecting wires
- Supply or Battery

Theory:

Full- Adder

A full-adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs and two outputs. Two of the input variables, denoted by x and y, represent two significant bits to be added. The third input z, represents the carry from the previous lower significant position. Two outputs are necessary because the arithmetic sum of three binary digits ranges in value from 0 to3 and binary 2or 3 needs two digits. The binary variable S indicates the sum and C the carry. The binary variable S gives the value if the least significant bit of the sum. The binary variable C gives the output carry.

Procedure:

- Implement the circuit of full adder on bread board
- Supply the required power (5V) to ICs AND OR and XOR
- Check the output using logic probe or LEDs and fill in the truth table

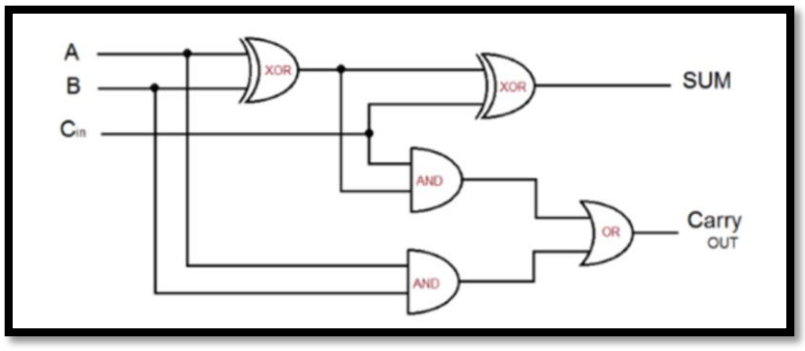


Figure 3.1: Full adder circuit

Observation:

The resulting truth table of the adder circuit is as follows:

A	B	CIN	COUT	S
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

NED University of Engineering & Technology
Department of ____**ELECTRONIC**__ Engineering
Course Code and Title: ____**EL-408 VLSI System Design**_____

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to ***identify*** equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to ***demonstrate*** the use of the equipment for the lab work.	Doesn't demonstrate the use of equipment.	Slightly demonstrates the use of equipment.	Somewhat demonstrates the use of equipment.	Moderately demon strates the use of equipment.	Fully demonstrates the use of equipment.
Procedural Skills ***Displays*** skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to ***imitate*** the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use ***Displays*** skills to use the observations from lab work for experimental verifications and illustrations.	Not able to use the observations from lab work for experimental verifications and illustrations.	Slightly able to use the observations from lab work for experimental verifications and illustrations.	Somewhat able to use the observations from lab work for experimental verifications and illustrations.	Moderately able to use the observations from lab work for experimental verifications and illustrations.	Fully able to use the observations from lab work for experimental verifications and illustrations.
Safety Adherence Adherence to ***safety*** procedures.	Doesn't adhere to safety procedures.	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling ***Equipment care*** during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care.	Often handles equipment with required care.	Handles equipment with required care.
Group Work ***Contributes*** in a group based lab work.	Doesn't participate and contribute.	Slightly participates and contributes.	Somewhat participates and contributes.	Moderately participates and contributes.	Fully participates and contributes.

Laboratory Session No. _____03_____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

LAB SESSION NO 04

Objective:

To operate under supervision all types of Flip-Flops circuits.

Apparatus:

- MC74HC73A (Dual JK flip-flop)
- HEF4013B (Dual D flip-flop)
- LM7805
- Switches
- Supply
- LEDS
- Resistor sheet
- Breadboard
- Connecting wires
- 7408 IC (AND)
- 7432 IC (OR)
- 7486 IC (XOR)
- 4049 IC (INVERTER)
- 7400 IC (NAND)

Theory:

The bistable multivibrator, commonly called flip- flops, are the most common form of digital memory elements. A memory element is generally a device which can store the logic state 0 or 1, called information "bit". The memory elements enable the storing of digital information for further uses. They permit to carry out complex sequential digital circuits, which took to the construction of modern calculators.

Types of Flip Flop

There are basically 4 types of flip-flops:

1. SR Flip Flop
2. JK Flip Flop
3. D Flip Flop
4. T Flip Flop

1)S-R Flip-flop (latch):

The **SR flip-flop**, also known as a *SR Latch*, can be considered as one of the most basic sequential logic circuit possible. This simple flip-flop is basically a one-bit memory bistable device that has two inputs, one which will "SET" the device (meaning the output = "1"), and is labelled **S** and one which will "RESET" the device (meaning the output = "0"), labelled **R**.

Then the SR description stands for "Set-Reset". The reset input resets the flip-flop back to its original state with an output Q that will be either at a logic level "1" or logic "0" depending upon this set/reset condition.

A basic NAND gate SR flip-flop circuit provides feedback from both of its outputs back to its opposing inputs and is commonly used in memory circuits to store a single data bit. Then the SR flip-flop actually has three inputs, Set, Reset and its current output Q relating to its current state or history.

The term "Flip-flop" relates to the actual operation of the device, as it can be "flipped" into one logic Set state or "flopped" back into the opposing logic Reset state.

The simplest way to make any basic single bit set-reset SR flip-flop is to connect together a pair of cross-coupled 2-input NAND gates as shown, to form a Set-Reset Bistable also known as an active LOW SR NAND Gate Latch, so that there is feedback from each output to one of the other NAND gate inputs.

This device consists of two inputs, one called the *Set*, S and the other called the *Reset*, R with two corresponding outputs Q and its inverse or complement Q (not-Q)

Basic RS Flip Flop circuit and symbol:

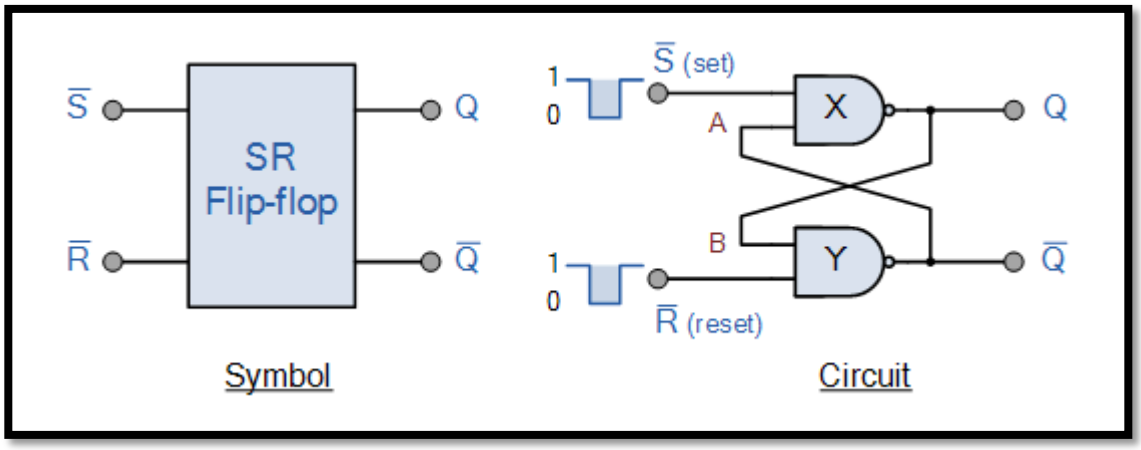


Figure 4.1: S-R Flip-flop symbol and circuit

The Set State

Consider the circuit shown above. If the input R is at logic level "0" (R = 0) and input S is at logic level "1" (S = 1), the NAND gate *Y* has at least one of its inputs at logic "0" therefore, its output Q must be at a logic level "1" (NAND Gate principles). Output Q is also fed back to input "A" and so both inputs to NAND gate *X* are at logic level "1", and therefore its output Q must be at logic level "0".

Again, NAND gate principals. If the reset input R changes state, and goes HIGH to logic "1" with S remaining HIGH also at logic level "1", NAND gate *Y* inputs are now R = "1" and B = "0". Since one of its inputs is still at logic level "0" the output at Q still remains HIGH at logic level "1" and there is no change of state. Therefore, the flip-flop circuit is said to be "Latched" or "Set" with Q = "1" and Q = "0".

The Reset State

In this second stable state, Q is at logic level "0", (not Q = "0") its inverse output at Q is at logic level "1", (Q = "1"), and is given by R = "1" and S = "0".

As gate *X* has one of its inputs at logic "0" its output Q must equal logic level "1" (again NAND gate principles). Output Q is fed back to input "B", so both inputs to NAND gate *Y* are at logic "1", therefore, Q = "0".

If the set input, S now changes state to logic "1" with input R remaining at logic "1", output Q still remains LOW at logic level "0" and there is no change of state. Therefore, the flip-flop circuits "Reset" state has also been latched and we can define this "set/reset" action in the following truth table.

State	S	R	Q	$\overline{Q}$	Description
Set	1	0	0	1	Set $\overline{Q}$ » 1
	1	1	0	1	no change
Reset	0	1	1	0	Reset $\overline{Q}$ » 0
	1	1	1	0	no change
Invalid	0	0	1	1	Invalid Condition

Figure 4.2: The Truth Table for the S-R Function

It can be seen that when both inputs S = "1" and R = "1" the outputs Q and Q can be at either logic level "1" or "0", depending upon the state of the inputs S or R BEFORE this input condition existed. Therefore, the condition of S = R = "1" does not change the state of the outputs Q and Q.

However, the input state of S = "0" and R = "0" is an undesirable or invalid condition and must be avoided. The condition of S = R = "0" causes both outputs Q and Q to be HIGH together at logic level "1" when we would normally want Q to be the inverse of Q.

The result is that the flip-flop loses control of Q and Q, and if the two inputs are now switched "HIGH" again after this condition to logic "1", the flip-flop becomes unstable and switches to an unknown data state based upon the unbalance as shown in the following switching diagram.

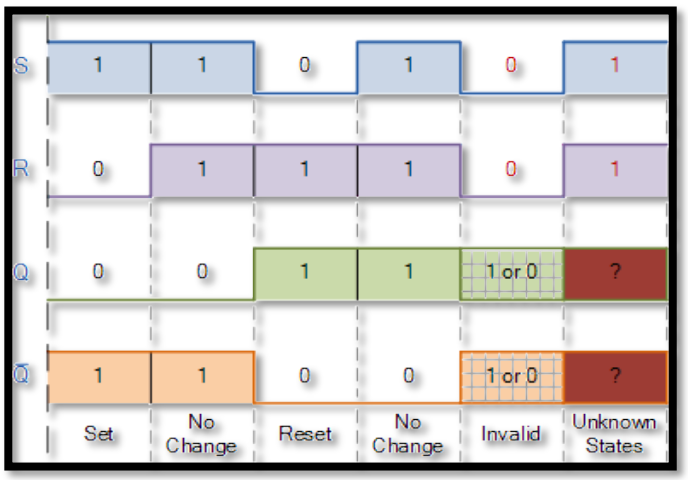


Figure 4.3: S-R Flip-flop Switching Diagram:

2) J-K flip-flop

The sequential operation of the JK flip flop is exactly the same as for the previous SR flip-flop with the same "Set" and "Reset" inputs. The difference this time is that the "JK flip flop" has no invalid or forbidden input states of the SR Latch even when S and R are both at logic "1".

The **JK flip flop** is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level "1". Due to this additional clocked input, a JK flip-flop has four possible input combinations, "logic 1", "logic 0", "no change" and "toggle".

Basic JK Flip Flop circuit and symbol:

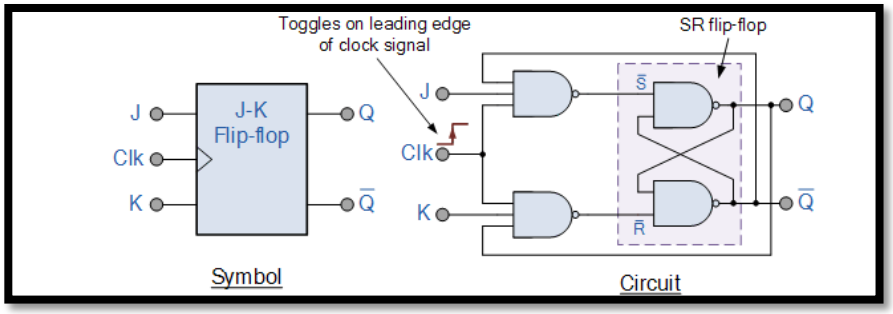


Figure 4.4: JK Flip-flop symbol and circuit

	Clock	Input		Output		Description
	Clk	J	K	Q	$\overline{Q}$	
same as for the SR Latch	X	0	0	1	0	Memory no change
	X	0	0	0	1	
	¯↓_	0	1	1	0	Reset Q » 0
	X	0	1	0	1	
	¯↓_	1	0	0	1	Set Q » 1
	X	1	0	1	0	
toggle action	¯↓_	1	1	0	1	Toggle
	¯↓_	1	1	1	0	

Figure 4.5: The Truth Table for the JK Function

3) <u>D Flip-flop:</u>

The **D Flip Flop** is by far the most important of all the clocked flip-flops. D Flip-flops are used as a part of memory storage elements and data processors as well. D flip-flop can be built using NAND gate or with NOR gate. Due to its versatility, they are available as IC packages. The major applications of D flip-flop are to introduce delay in timing circuit, as a buffer, sampling data at specific intervals. D flip-flop is simpler in terms of wiring connection compared to JK flip-flop. Here we are using **NAND gates** for demonstrating the D flip flop.

By adding an inverter (NOT gate) between the Set and Reset inputs, the S and R inputs become complements of each other ensuring that the two inputs S and R are never equal (0 or 1) to each other at the same time allowing us to control the toggle action of the flip-flop using one single D (Data) input. Then this Data input, labelled "D" and is used in place of the "Set" signal, and the inverter is used to generate the complementary "Reset" input thereby making a level-sensitive D-type flip-flop from a level-sensitive SR-latch as now S = D and R = not D as shown.

Basic D Flip Flop circuit and symbol:

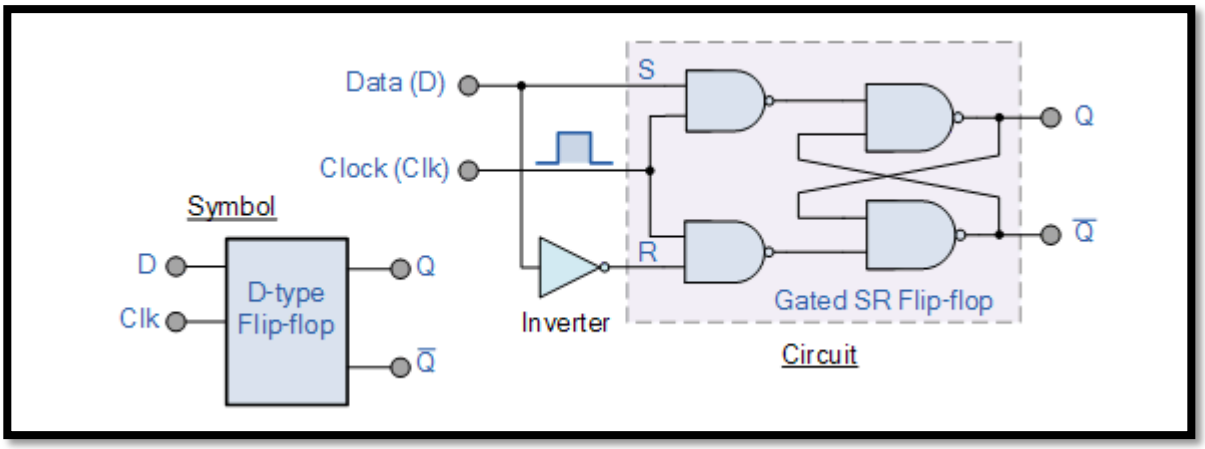


Figure 4.6: D Flip-flop symbol and circuit

A simple SR flip-flop requires two inputs, one to "SET" the output and one to "RESET" the output. By connecting an inverter (NOT gate) to the SR flip-flop we can "SET" and "RESET" the flip-flop using just one input as now the two input signals are complements of each other. This complement avoids the ambiguity inherent in the SR latch when both inputs are LOW, since that state is no longer possible.

Thus, this single input is called the "DATA" input. If this data input is held HIGH the flip flop would be "SET" and when it is LOW the flip flop would change and become "RESET". However, this would be rather pointless since the output of the flip flop would always change on every pulse applied to this data input.

To avoid this an additional input called the "CLOCK" or "ENABLE" input is used to isolate the data input from the flip flop's latching circuitry after the desired data has been stored. The effect is that D input condition is only copied to the output Q when the clock input is active. This then forms the basis of another sequential device called a **D Flip Flop**.

The "D flip flop" will store and output whatever logic level is applied to its data terminal so long as the clock input is HIGH. Once the clock input goes LOW the "set" and "reset" inputs of the flip-flop are both held at logic level "1" so it will not change state and store whatever data was present on its output before the clock transition occurred. In other words, the output is "latched" at either logic "0" or logic "1".

Clk	D	Q		Description
↓ » 0	X	Q	$\overline{Q}$	Memory no change
↑ » 1	0	0	1	Reset Q » 0
↑ » 1	1	1	0	Set Q » 1

Figure 4.7: The Truth Table for the D Function

Note that: ↓ and ↑ indicates direction of clock pulse as it is assumed D-type flip flops are edge triggered

4)T Flip-flop

While the *Data* (D) flip-flop is a variation of a clocked SR flip-flop constructed using either NAND or NOR gates, the *Toggle* (T) flip-flop is a variation of the clocked JK flip-flop. The toggle or T-type flip-flop gets its name from the fact that its two outputs Q and Q invert from their previous state as it toggles back and forth every time it is triggered (T = 1).

The Toggle schematic symbol has two inputs available; one represents the "toggle" (T) input and the other the "clock" (CLK) input.

This modified form of the JK flip-flop is obtained by connecting both inputs J and K together. It has only one input along with the clock input. It inverts the state of the outputs each time the input pulse applied to line T passes from the state 1 to the state.

Basic T Flip Flop circuit and symbol:

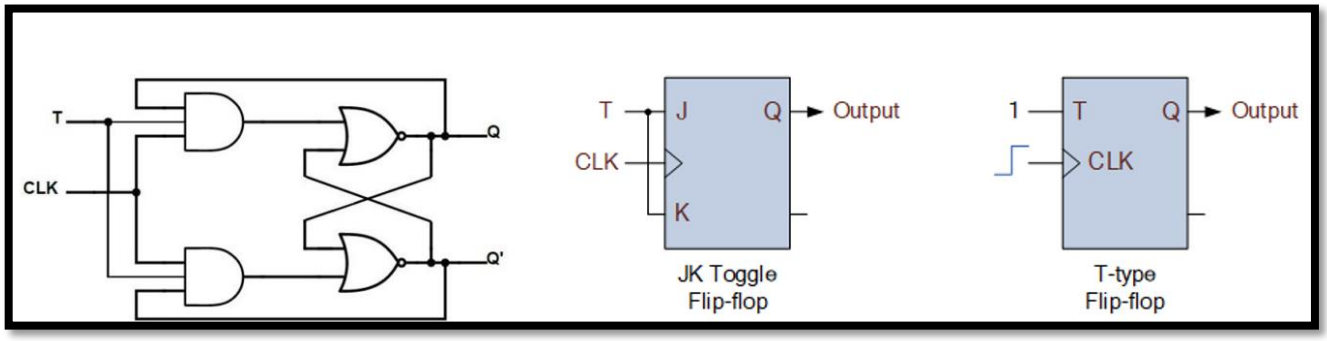


Figure 4.8: T Flip-flop symbol and circuit

Clock	INPUT		OUTPUT	
	RESET	T	Q	Q'
X	LOW	X	0	1
HIGH	HIGH	0	No Change	
HIGH	HIGH	1	Toggle	
LOW	HIGH	X	No Change	

Figure 4.9: The Truth Table for the T Function

Procedure:

1.Implementation and Analysis of a R-S Flip-flop

- Carry out a flip- flop type R-S either using NAND ports 7400 IC or **SN74HC00N** IC by looking into schematic provided.
- Connect the SET and RESET inputs to two switches.
- Connect the outputs Q and Q to two LEDs.
- Turn the SET input, with the switch, to 1 and then to 0.
- Analyze the behavior of the outputs.
- Set the RESET line to 1, and then to 0.
- Analyze the behavior of the outputs again.
- Repeat sometimes the operations with the switches and check the carried-out memorizations.

Now, try to set both inputs to 1 and explain what the reason of the uncertain state is.

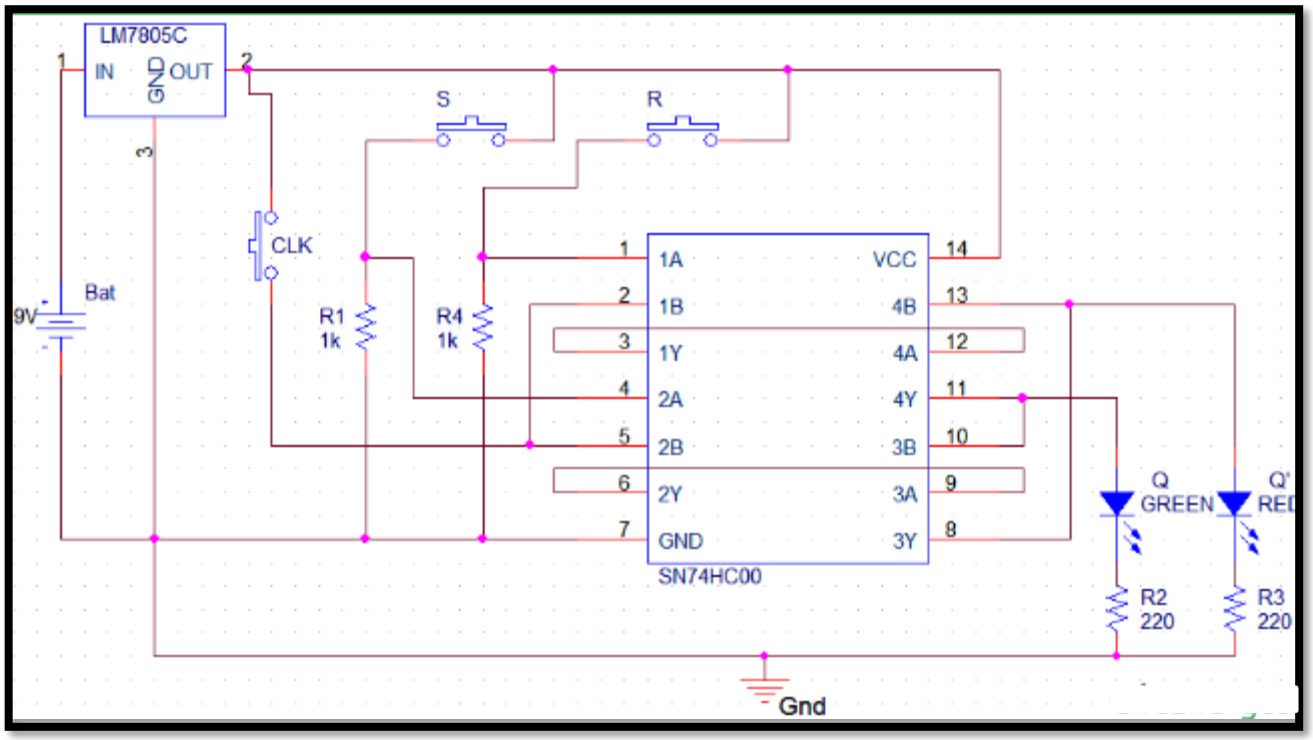


Figure 4.10: RS Flip-flop schematic diagram

Observation:

S	R	Q	Q`
0	0		
0	1		
1	0		
1	1		

2. Implementation of negative edge J-K Flip-flop

- Carry out the circuit of a flip-flop type by using MC74HC73A according to the given schematic diagram
- Connect the inputs J and K to two switches, and the outputs to two LEDs.
- Set the switches connected to the inputs alternatively high.
- Analyze the behavior of the LEDs.
- Now, set both switches to the logic level 1, and explain the behavior of the flip-flop.

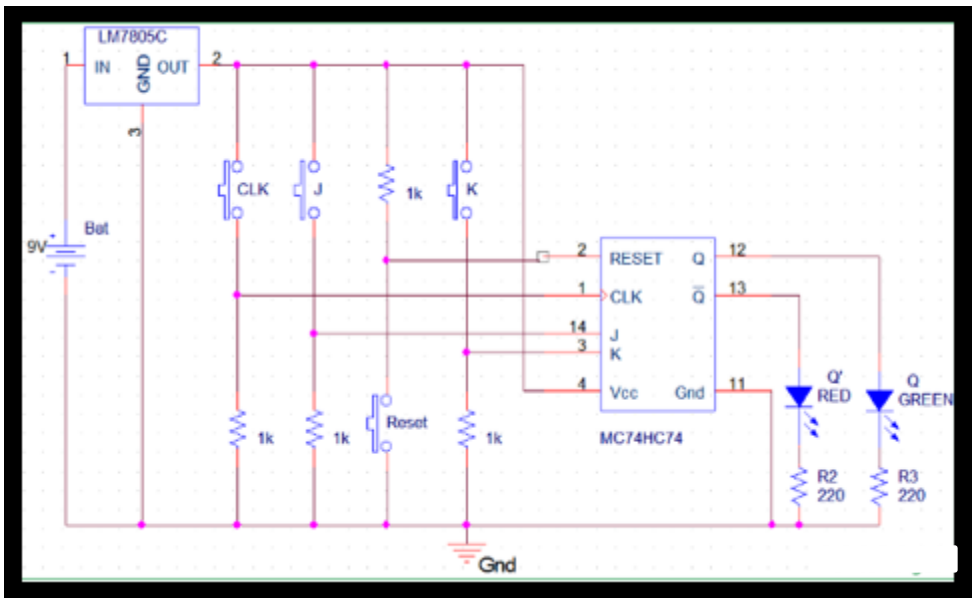


Figure 4.11: J-K Flip-flop schematic diagram

Observation:

J	K	Q	Q`
0	0		
0	1		
1	0		
1	1		

3. Implementation of D Flip-flop

- Carry out the circuit of a flip-flop type D by means IC **HEF4013B** by looking into the schematic
- Connect the inputs P and R to 1.
- Check the operation of the flip-flop D by means of switches 0-1 of the input D and the Clock.

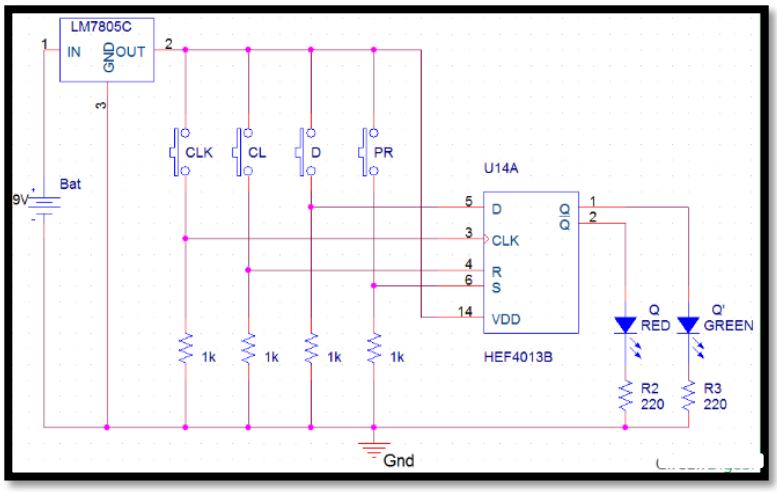


Figure 4.12: D Flip-flop schematic diagram

Observation:

D	Q

4. Implementation of T- Flip-flop

- Carry out the circuit of a flip-flop type T by means of J-K flip-flop according to the given schematic diagram
- Check the operation of the flip-flop T by means of switches 0-1 to the Clock input

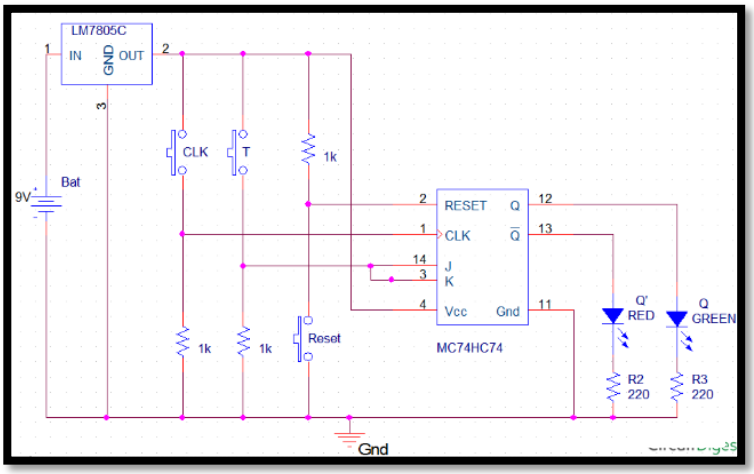


Figure 4.13: T Flip-flop schematic diagram

Observation:

T	Q

NED University of Engineering & Technology
Department of ____**ELECTRONIC**__ Engineering
Course Code and Title: ____**EL-408 VLSI System Design**_____

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to ***identify*** equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to ***demonstrate*** the use of the equipment for the lab work.	Doesn't demonstrate the use of equipment.	Slightly demonstrates the use of equipment.	Somewhat demonstrates the use of equipment.	Moderately demon strates the use of equipment.	Fully demonstrates the use of equipment.
Procedural Skills ***Displays*** skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to ***imitate*** the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use ***Displays*** skills to use the observations from lab work for experimental verifications and illustrations.	Not able to use the observations from lab work for experimental verifications and illustrations.	Slightly able to use the observations from lab work for experimental verifications and illustrations.	Somewhat able to use the observations from lab work for experimental verifications and illustrations.	Moderately able to use the observations from lab work for experimental verifications and illustrations.	Fully able to use the observations from lab work for experimental verifications and illustrations.
Safety Adherence Adherence to ***safety*** procedures.	Doesn't adhere to safety procedures.	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling ***Equipment care*** during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care.	Often handles equipment with required care.	Handles equipment with required care.
Group Work ***Contributes*** in a group based lab work.	Doesn't participate and contribute.	Slightly participates and contributes.	Somewhat participates and contributes.	Moderately participates and contributes.	Fully participates and contributes.

Laboratory Session No. ___04_____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

LAB SESSION NO 05

Objective:

To imitate 2-bit counter circuit.

Apparatus:

- MC74HC73A (Dual JK flip-flop)
- HEF4013B (Dual D flip-flop)
- LM7805
- Supply
- LEDS
- Resistor sheet
- Breadboard
- Connecting wires
- 7400 IC (NAND)
- Function generator
- Oscilloscope

Theory:

Counters:

Counters are digital integrated devices which can state in a well-defined sequence, applying a train pulse across the input. A counter is a circuit consisting of a number of Flip Flop and gates working together to count the number of clock pulses applied to its input. Such counters are used in digital clocks, frequency counters, digital voltmeters, digital computers, and numerous other applications. These counters are also called 'BINARY COUNTERS' and can be used apart as counters as frequency dividers, supplying the o/p with a pulse after 'n' input pulses.
If we apply a fixed frequency pulse train to a counter, rather than individual pulses coming at random intervals, we begin to notice some interesting characteristics and useful relationships between the input clock signal and the output signal.

Ripple Counter (Asynchronous):

A ripple counter is a serial counter. The clock input is applied to only the first of the series of the Flip Flop. Clock pulses for the other Flip Flop come from the preceding Flip Flop. Thus, the clock pulse "ripple" through the circuit in a series fashion. Such circuit is also called asynchronous since the only pulse required for the operation is the clock pulse. The JK Flip Flop have the J and K inputs both tied high, which allows them to toggle with each input pulse. The below figure shows a 2-bit ripple counter.

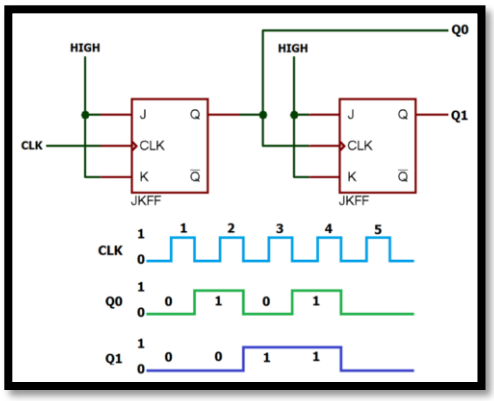


Figure 5.1: 2-bit ripple counter

Consider a single flip flop with a continuous succession of above. We note three useful facts about the output signals seen at Q and Q':

a) They are exactly inverted to each other
b) They are perfect square waves (50% duty cycle)
c) They have a frequency just half that of the clock pulse train.

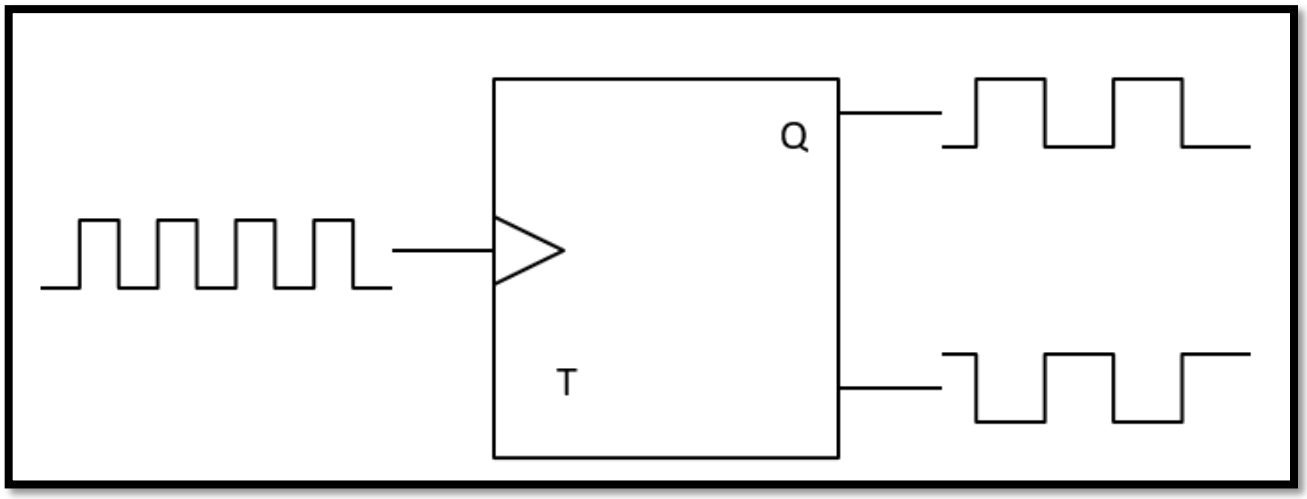


Figure 5.2: 2-bit ripple counter symbol

Procedure:

2-bit Asynchronous Counter:

- Carry out the circuit of a 2-stage asynchronous counter

- Connect all inputs, J and K to logic 1 (i.e., +5V)
- Connect 1Hz clock to input CK of first flip-flop
- Connect Q0 to CK of second flip-flop
- Connect the two outputs, Q0 and Q1, to oscilloscope
- Power the circuit and analyze the operation of complete system.

Outcome:

F/OBEM 01/05/00

NED University of Engineering & Technology
Department of ____**ELECTRONIC**__ Engineering
Course Code and Title: ____**EL-408 VLSI System Design**_____

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to ***identify*** equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to ***demonstrate*** the use of the equipment for the lab work.	Doesn't demonstrate the use of equipment.	Slightly demonstrates the use of equipment.	Somewhat demonstrates the use of equipment.	Moderately demon strates the use of equipment.	Fully demonstrates the use of equipment.
Procedural Skills ***Displays*** skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to ***imitate*** the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use ***Displays*** skills to use the observations from lab work for experimental verifications and illustrations.	Not able to use the observations from lab work for experimental verifications and illustrations.	Slightly able to use the observations from lab work for experimental verifications and illustrations.	Somewhat able to use the observations from lab work for experimental verifications and illustrations.	Moderately able to use the observations from lab work for experimental verifications and illustrations.	Fully able to use the observations from lab work for experimental verifications and illustrations.
Safety Adherence Adherence to ***safety*** procedures.	Doesn't adhere to safety procedures.	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling ***Equipment care*** during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care.	Often handles equipment with required care.	Handles equipment with required care.
Group Work ***Contributes*** in a group based lab work.	Doesn't participate and contribute.	Slightly participates and contributes.	Somewhat participates and contributes.	Moderately participates and contributes.	Fully participates and contributes.

Laboratory Session No. ___05_____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

LAB SESSION NO 06

Objective:

To practice and verify the layout of a CMOS inverter.

- **There must be a single rail of Vdd as well as for Vss.**
- **Width of PMOS is twice (or 2.5 times) as compared to NMOS transistor**.

Apparatus:

- Microwind/Mentor Graphics software installed PCs

Schematic:

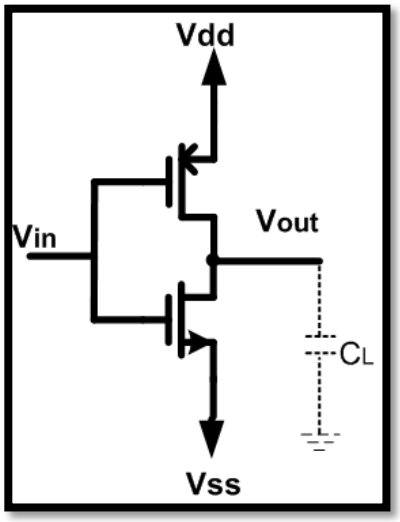


Figure 6.1: CMOS inverter schematic symbol

Theory:

CMOS logic-based function consists of a pair of NMOS and PMOS transistors. Two different transistors have to be placed in Layout so it will be a good exercise for learning structure of both transistors. The network consisting of all PMOS transistors, known as P-network, is responsible for determining rise time of output. While, the network consisting of all NMOS transistors, known as N-network, is responsible for determining fall time of output waveform.

CMOS logic-based function consists of a pair of NMOS and PMOS transistors. Two different transistors have to be placed in Layout so it will be a good exercise for learning structure of both transistors. The network consisting of all PMOS transistors, known as P-network, is responsible for determining rise time of output. While, the network consisting of all NMOS transistors, known as N-network, is responsible for determining fall time of output waveform.

There are two inputs involved i.e., A and B. So, there should be optimum gate ordering in layout. A simple method for finding the optimum gate ordering is the Euler-path method.

Procedure:

Follow the instructions given during Lab to complete task successfully.

Outcome:

a) What will be the effect on rise and fall time of output waveform, when:

1. $\left(\frac{W}{L}\right)n = \left(\frac{W}{L}\right)n$

2. $\left(\frac{W}{L}\right)n = 2\left(\frac{W}{L}\right)n$

b) Attached the Layout of CMOS inverter.

F/OBEM 01/05/00

NED University of Engineering & Technology

Department of Electronic Engineering

Course Code and Title: _____EL-408 VLSI System Design_____

Laboratory Session No. _____ 06_____ Date: _____

Software Use Rubric					
Criterion	**Level of Attainment**				
	Below Average (1)	**Average (2)**	**Good (3)**	**Very Good (4)**	**Excellent (5)**
Identification of software menu (syntax, components, commands, tools, layout etc.).	Can't identify software menus.	Rarely identifies software menus.	Occasionally identifies software menus.	Able to identify software menus.	Perfectly able to identify software menus.
Skills to use software (schematic, syntax, commands, tools, layout) efficiently.	Can't use software efficiently.	Rarely uses software efficiently.	Occasionally uses software efficiently.	Often uses software efficiently.	Efficiently uses software (syntax, commands, tools, layout)
Adherence to safety procedures and handling of equipment (computing unit, peripheral devices, and other equipment in lab).	Doesn't handle equipment with required care and safety.	Rarely handles equipment with required care and safety.	Occasionally handles equipment with required care and safety.	Often handles equipment with required care and safety.	Handles equipment with required care and safety.
Ability to troubleshoot software errors (detection and debugging).	Not able to troubleshoot the errors	Rarely able to troubleshoot the errors	Occasionally able to troubleshoot the errors	Often able to troubleshoot the errors	Fully able to troubleshoot the errors
Analysis and interpretation of results/outputs.	Not able to analyze and interpret results/outputs.	Rarely able to perform the analysis and interpretation.	Occasionally able to perform the analysis and interpretation.	Often able to perform the analysis and interpretation.	Perfectly able to perform the analysis and interpretation.

Weighted CLO (Score)	
Remarks	
Instructor's Signature with Date	

LAB SESSION NO 07

Objective:

To operate under supervision the layout of a CMOS NAND gate.

- **There should be continuous layer of n+ and p+ diffusion.**
- **There must be a single rail of Vdd as well as for Vss.**
- **Width of PMOS is twice (or 2.5 times) as compared to NMOS transistor and length of all transistors should be same.**

Apparatus:

Microwind/Mentor Graphics software installed PCs

Theory:

CMOS NAND gate is one of the important and simple realizations. CMOS is the combination of PMOS and NMOS. The circuit shows the realization of CMOS NAND gate which consists of two PMOS and two NMOS gates.

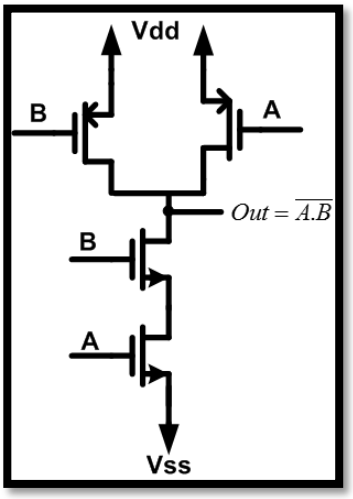


Figure 7.1: CMOS inverter NAND gate configuration

Procedure:

Follow the instructions given during Lab to complete task successfully.

Outcome:

Attached the Layout of CMOS logic NAND gate, mentioning all diffusion layer,

F/OBEM 01/05/00

NED University of Engineering & Technology
Department of Electronic Engineering

Course Code and Title: _____EL-408 VLSI System Design_____

Laboratory Session No. _____07_____ Date: _____

Software Use Rubric					
Criterion	**Level of Attainment**				
	Below Average (1)	**Average (2)**	**Good (3)**	**Very Good (4)**	**Excellent (5)**
Identification of software menu (syntax, components, commands, tools, layout etc.).	Can't identify software menus.	Rarely identifies software menus.	Occasionally identifies software menus.	Able to identify software menus.	Perfectly able to identify software menus.
Skills to use software (schematic, syntax, commands, tools, layout) efficiently.	Can't use software efficiently.	Rarely uses software efficiently.	Occasionally uses software efficiently.	Often uses software efficiently.	Efficiently uses software (syntax, commands, tools, layout)
Adherence to safety procedures and handling of equipment (computing unit, peripheral devices, and other equipment in lab).	Doesn't handle equipment with required care and safety.	Rarely handles equipment with required care and safety.	Occasionally handles equipment with required care and safety.	Often handles equipment with required care and safety.	Handles equipment with required care and safety.
Ability to troubleshoot software errors (detection and debugging).	Not able to troubleshoot the errors	Rarely able to troubleshoot the errors	Occasionally able to troubleshoot the errors	Often able to troubleshoot the errors	Fully able to troubleshoot the errors
Analysis and interpretation of results/outputs.	Not able to analyze and interpret results/outputs.	Rarely able to perform the analysis and interpretation.	Occasionally able to perform the analysis and interpretation.	Often able to perform the analysis and interpretation.	Perfectly able to perform the analysis and interpretation.

Weighted CLO (Score)	
Remarks	
Instructor's Signature with Date	

LAB SESSION NO 08

Objective:

To imitate and verify the layout of dynamic inverter logic using microwind

Apparatus:

Microwind/Mentor Graphics software installed PCs

Theory:

Dynamic logic is distinguished from so-called *static logic* in that dynamic logic uses a clock signal in its implementation of combinational logic circuits. The usual use of a clock signal is to synchronize transitions in sequential logic circuits.

The dynamic logic circuit requires two phases. The first phase, when *Clock* is low, is called the *setup phase* or the *precharge phase* and the second phase, when *Clock* is high, is called the *evaluation phase*. In the setup phase, the output is driven high unconditionally (no matter the values of the inputs *A* and *B*). The capacitor, which represents the load capacitance of this gate, becomes charged. Because the transistor at the bottom is turned off, it is impossible for the output to be driven low during this phase.

During the *evaluation phase*, *Clock* is high. If *A* and *B* are also high, the output will be pulled low. Otherwise, the output stays high (due to the load capacitance).

Dynamic logic has a few potential problems that static logic does not. For example, if the clock speed is too slow, the output will decay too quickly to be of use. Also, the output is only valid for part of each clock cycle, so the device connected to it must sample it synchronously during the time that it is valid.

Also, when both *A* and *B* are high, so that the output is low, the circuit will pump one capacitor-load of charge from Vdd to ground for each clock cycle, by first charging and then discharging the capacitor in each clock cycle. This makes the circuit (with its output connected to a high impedance) less efficient than the static version (which theoretically should not allow any current to flow except through the output), and when the *A* and *B* inputs are constant and both high, the dynamic NAND gate uses power in proportion to the clock rate, as long as it functions correctly

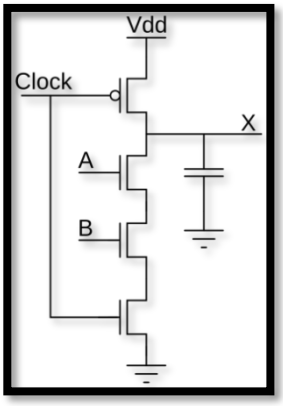


Figure 8.1: Dynamic NAND gate configuration

Procedure & Outcome:

a) Take frequency of clock signal in comparison to frequency of square wave input signal. Observe and attach the simulation results.
b) Take frequency of clock signal much smaller than frequency of square wave input signal. Observe and attach the simulation results.
c) Take frequency of clock signal much greater than frequency of square wave input signal. Observe and attach the simulation results.

NED University of Engineering & Technology
Department of Electronic Engineering

Course Code and Title: _____EL-408 VLSI System Design_____

Laboratory Session No. _____08_____ Date: _____

Software Use Rubric					
Criterion	**Level of Attainment**				
	Below Average (1)	**Average (2)**	**Good (3)**	**Very Good (4)**	**Excellent (5)**
Identification of software menu (syntax, components, commands, tools, layout etc.).	Can't identify software menus.	Rarely identifies software menus.	Occasionally identifies software menus.	Able to identify software menus.	Perfectly able to identify software menus.
Skills to use software (schematic, syntax, commands, tools, layout) efficiently.	Can't use software efficiently.	Rarely uses software efficiently.	Occasionally uses software efficiently.	Often uses software efficiently.	Efficiently uses software (syntax, commands, tools, layout)
Adherence to safety procedures and handling of equipment (computing unit, peripheral devices, and other equipment in lab).	Doesn't handle equipment with required care and safety.	Rarely handles equipment with required care and safety.	Occasionally handles equipment with required care and safety.	Often handles equipment with required care and safety.	Handles equipment with required care and safety.
Ability to troubleshoot software errors (detection and debugging).	Not able to troubleshoot the errors	Rarely able to troubleshoot the errors	Occasionally able to troubleshoot the errors	Often able to troubleshoot the errors	Fully able to troubleshoot the errors
Analysis and interpretation of results/outputs.	Not able to analyze and interpret results/outputs.	Rarely able to perform the analysis and interpretation.	Occasionally able to perform the analysis and interpretation.	Often able to perform the analysis and interpretation.	Perfectly able to perform the analysis and interpretation.

Weighted CLO (Score)	
Remarks	
Instructor's Signature with Date	

LAB SESSION NO 09

Objective:

To manipulate with guidance the layout of a logic function on CMOS logic.

- **There should be continuous layer of n+ and p+ diffusion.**
- **There must be a single rail of Vdd as well as for Vss.**
- **Width of PMOS is twice (or 2.5 times) as compared to NMOS transistor and length of all transistors should be same**.

Apparatus:

- Microwind/Mentor Graphics software installed PCs

Schematic for Logic Function:

Theory:

CMOS logic-based function consists of a pair of NMOS and PMOS transistors. Two different transistors have to be placed in Layout so it will be a good exercise for learning structure of both transistors. The network consisting of all PMOS transistors, known as P-network, is responsible for determining rise time of output. While, the network consisting of all NMOS transistors, known as N-network, is responsible for determining fall time of output waveform.

For multiple inputs, there should be optimum gate ordering in layout. A simple method for finding the optimum gate ordering is the Euler-path method.

Procedure:

Follow the instructions given during Lab to complete task successfully.

Outcome:

Attach the layout of a given function.

NED University of Engineering & Technology
Department of Electronic Engineering

Course Code and Title: _____EL-408 VLSI System Design_____
Laboratory Session No. _____09_____ Date: _____

Software Use Rubric					
Criterion	**Level of Attainment**				
	Below Average (1)	**Average (2)**	**Good (3)**	**Very Good (4)**	**Excellent (5)**
Identification of software menu (syntax, components, commands, tools, layout etc.).	Can't identify software menus.	Rarely identifies software menus.	Occasionally identifies software menus.	Able to identify software menus.	Perfectly able to identify software menus.
Skills to use software (schematic, syntax, commands, tools, layout) efficiently.	Can't use software efficiently.	Rarely uses software efficiently.	Occasionally uses software efficiently.	Often uses software efficiently.	Efficiently uses software (syntax, commands, tools, layout)
Adherence to safety procedures and handling of equipment (computing unit, peripheral devices, and other equipment in lab).	Doesn't handle equipment with required care and safety.	Rarely handles equipment with required care and safety.	Occasionally handles equipment with required care and safety.	Often handles equipment with required care and safety.	Handles equipment with required care and safety.
Ability to troubleshoot software errors (detection and debugging).	Not able to troubleshoot the errors	Rarely able to troubleshoot the errors	Occasionally able to troubleshoot the errors	Often able to troubleshoot the errors	Fully able to troubleshoot the errors
Analysis and interpretation of results/outputs.	Not able to analyze and interpret results/outputs.	Rarely able to perform the analysis and interpretation.	Occasionally able to perform the analysis and interpretation.	Often able to perform the analysis and interpretation.	Perfectly able to perform the analysis and interpretation.

Weighted CLO (Score)	
Remarks	
Instructor's Signature with Date	

LAB SESSION NO 10

Objective:

To practice all logic gates on Modelsim Software.

Apparatus:

Modelsim software installed PCs

Theory:

Traditionally, digital design was done by the schematic entry. This has been replaced today by the use of Hardware Description Language (HDL). In electronics, HDL is the language used for formal description of electronic circuits. HDL offers several advantages over traditional design techniques such as efficient and convenient way of designing, simulation and synthesis of large electronic circuits containing larger number of electronic components and devices, efficient verification of design in initial phase of development. Two popular HDLs are Verilog and VHDL. The HDL used in our lab will be Verilog.

Introduction to Verilog HDL:

Verilog is a hardware description language used to model electronic systems. Verilog HDL is one of the HDLs used by the integrated circuits (IC) designers. The other one is VHDL (VHSIC-Very High-Speed Integrated Circuit Hardware Description Language).

Verilog uses four levels of abstraction to describe the designs.

- The switch levels
- The gate or structural level
- The data flow level
- The behavioral or procedural level

The Switch Level:

It includes MOS transistors modeled as switches.

The Gate or Structural Level:

At this level, gates primitives are called to design the logic. It is not synthesizable.

The Data Flow Level:

At this level, continuous assignments are used by using the keyword ‘assign’. Synthesizer is required. It is also called Data Flow Models. This level allows using every type of operators.

The Behavioral or Procedural Level:

At this level, you just have to define the behavior. It is user friendly. It uses procedural blocks (Blocking and non-blocking) hence, called procedural level.

Procedure (A):

1. Open the ModelSim software.
2. Create a new project by **File => New => Project** from the Main window.
3. A "**Create Project**" window appears as shown in figure below. Select a suitable name for your project; leave the Default Library Name to work.

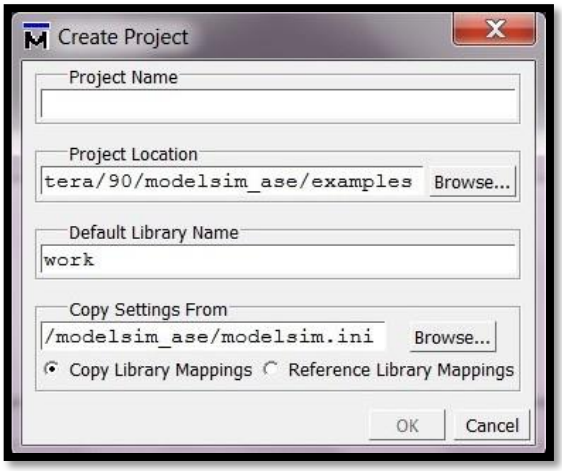


Figure 10.1: Create Project window snap

4. After project name, an **Add items to the Project** dialog pops out as shown in figure below.
5. From the "Add items to the Project" dialog click on **Create a new file**. If you have closed the "Add items to the Project" dialog, then select Project => **Add to Project => New File** from the main window.
6. A **Create Project File** dialog pops out. Select an appropriate file name for the file you want to add (the name of file must be same as you write in Step 4); choose **Verilog** as the add file as type option and **Top level** as the Folder option (see figure below) and then click on OK.
7. On the workspace section of the **Main Window**, double-click on the file you have just created (VLSI.v in our case).
8. Type Verilog code of the given task in the new window.
9. Save your code.
10. In workspace window do right click on project name (i.e., VLSI) select **Compile => Compile All**. A message "Compile of VLSI.v was successful" will appear in message window
11. For simulating the design click on **Simulation => Start Simulation** in main window, simulation environment will appears as shown in figure below.
12. Click on the (+) sign next to the **work** library. You should see the name of the entity of the code that we have just compiled **"VLSI"** select your desired file.
13. Locate the signals window and select the signals that you want to monitor for simulation. For this example of AND gate, select all signals as shown figure below.
14. Drag the above signals by selecting all then right click and select **Add => to Wave**=>

Selected items to the wave window.

15. Now we are ready to simulate our design. For this purpose, we will change the values of inputs (i.e., a and b in above example of AND gate) by right click on input and select **Force** and write either '0' or '1' in value box and repeat same step for changing the value of other inputs.
16. Click **Run** button in main window tool bar and can see the changes in the both the wave and objects windows.

Procedure (B)

1. Understand Gate level modeling.
2. Create new Modelsim project for writing the code.
3. Open new Verilog file and write code for Multiplexer in it.
4. Include Verilog file in your project and compile your project.
5. Simulate your project and verify results.
6. Understand Data flow modeling.
7. Repeat steps 3 to 5.
8. Understand Behavioral Modeling.
9. Repeat steps 3 to 5.

Outcome

a) Design the modules using all basic gates: AND, OR, XOR, NOR, NAND and XNOR gate and verify the results
b) Attach the verified results.

NED University of Engineering & Technology
Department of Electronic Engineering

Course Code and Title: _____EL-408 VLSI System Design_____
Laboratory Session No. _____10_____ Date: _____

Software Use Rubric					
Criterion	**Level of Attainment**				
	Below Average (1)	**Average (2)**	**Good (3)**	**Very Good (4)**	**Excellent (5)**
Identification of software menu (syntax, components, commands, tools, layout etc.).	Can't identify software menus.	Rarely identifies software menus.	Occasionally identifies software menus.	Able to identify software menus.	Perfectly able to identify software menus.
Skills to use software (schematic, syntax, commands, tools, layout) efficiently.	Can't use software efficiently.	Rarely uses software efficiently.	Occasionally uses software efficiently.	Often uses software efficiently.	Efficiently uses software (syntax, commands, tools, layout)
Adherence to safety procedures and handling of equipment (computing unit, peripheral devices, and other equipment in lab).	Doesn't handle equipment with required care and safety.	Rarely handles equipment with required care and safety.	Occasionally handles equipment with required care and safety.	Often handles equipment with required care and safety.	Handles equipment with required care and safety.
Ability to troubleshoot software errors (detection and debugging).	Not able to troubleshoot the errors	Rarely able to troubleshoot the errors	Occasionally able to troubleshoot the errors	Often able to troubleshoot the errors	Fully able to troubleshoot the errors
Analysis and interpretation of results/outputs.	Not able to analyze and interpret results/outputs.	Rarely able to perform the analysis and interpretation.	Occasionally able to perform the analysis and interpretation.	Often able to perform the analysis and interpretation.	Perfectly able to perform the analysis and interpretation.

Weighted CLO (Score)	
Remarks	
Instructor's Signature with Date	

LAB SESSION NO 11

Objective:

To imitate 4 to 1 MUX on Modelsim Software

Apparatus:

- Modelsim software installed PCs

Theory:

A multiplexer (MUX) is a digital switch which connects data from one of "n" inputs to a single output. A number of "Select Inputs" determine which data input is connected to the output. The Block Diagram of MUX with "n" data inputs and "s" select lines is shown in figure below:

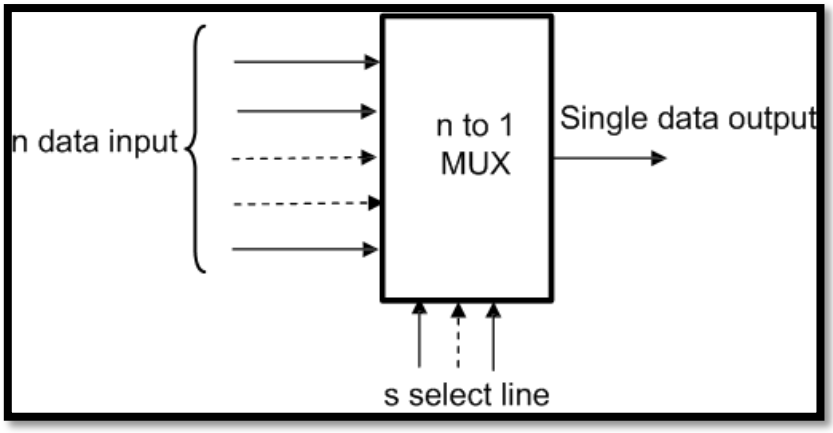


Figure 11.1: n to1 Multiplexer block diagram

MUX acts like a digitally controlled multi-position switch where the binary code applied to the select inputs controls the input source that will be switched on to the output. At any given point of time only one input gets selected and is connected to output, based on the select input signal. Input can be single bit or multi bits in nature.

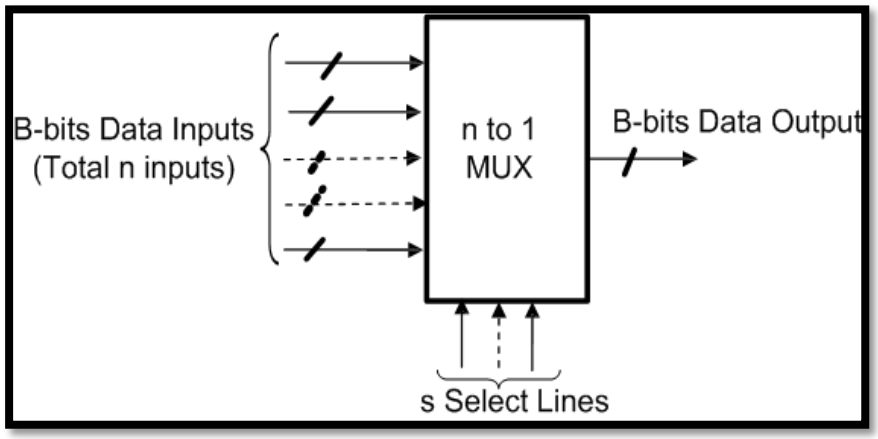


Figure 11.2: shows n to 1 MUX, handling "B" bits of each input and select them to "B" bits outpu

A 4 to 1 MUX is shown in figure below. There are four input lines, I0 to I3, and two selections

lines, S0 and S1, are decoded to select a particular input to appear at output.

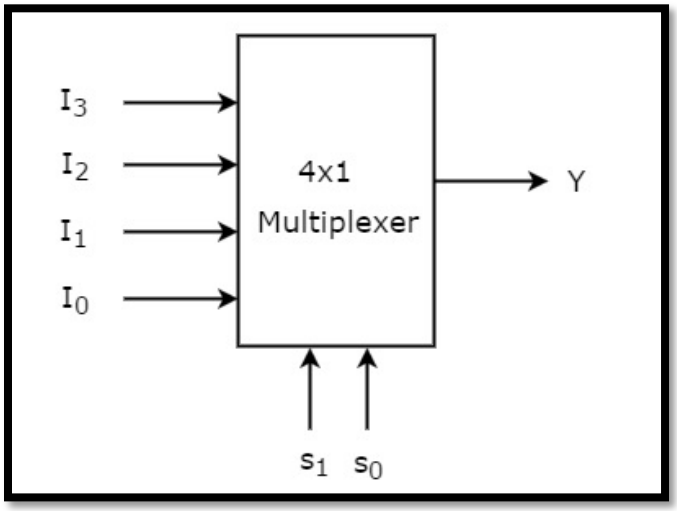


Figure 11.3: 4 to1 Multiplexer block diagram

The truth table for the 4:1 MUX is given as:

S1	S0	Output
0	0	I0
0	1	I1
1	0	I2
1	1	I3

Boolean function for output, Y as

$$Y=S1'S0'I0+S1'S0I1+S1S0'I2+S1S0I3$$

Procedure:

1. Open the ModelSim software.
2. Follow the procedure given in last lab.
3. Write Verilog code of 4 to 1 MUX.
4. Follow the steps described in previous lab to get output/simulation results.

Outcome:

Attach the printout of observed results.

NED University of Engineering & Technology
Department of Electronic Engineering

Course Code and Title: _____EL-408 VLSI System Design_____
Laboratory Session No. _____11_____ Date: _____

Software Use Rubric					
Criterion	**Level of Attainment**				
	Below Average (1)	**Average (2)**	**Good (3)**	**Very Good (4)**	**Excellent (5)**
Identification of software menu (syntax, components, commands, tools, layout etc.).	Can't identify software menus.	Rarely identifies software menus.	Occasionally identifies software menus.	Able to identify software menus.	Perfectly able to identify software menus.
Skills to use software (schematic, syntax, commands, tools, layout) efficiently.	Can't use software efficiently.	Rarely uses software efficiently.	Occasionally uses software efficiently.	Often uses software efficiently.	Efficiently uses software (syntax, commands, tools, layout)
Adherence to safety procedures and handling of equipment (computing unit, peripheral devices, and other equipment in lab).	Doesn't handle equipment with required care and safety.	Rarely handles equipment with required care and safety.	Occasionally handles equipment with required care and safety.	Often handles equipment with required care and safety.	Handles equipment with required care and safety.
Ability to troubleshoot software errors (detection and debugging).	Not able to troubleshoot the errors	Rarely able to troubleshoot the errors	Occasionally able to troubleshoot the errors	Often able to troubleshoot the errors	Fully able to troubleshoot the errors
Analysis and interpretation of results/outputs.	Not able to analyze and interpret results/outputs.	Rarely able to perform the analysis and interpretation.	Occasionally able to perform the analysis and interpretation.	Often able to perform the analysis and interpretation.	Perfectly able to perform the analysis and interpretation.

Weighted CLO (Score)	
Remarks	
Instructor's Signature with Date	

LAB SESSION NO 12

Objective:

To operate under supervision SR latch using Verilog HDL code on Quartus Software. Analyze the function of latch by testing code on ALTERA DE2 FPGA board

Apparatus:

- Altera cyclone II kit.
- Quartus II installed PCs.

Theory:

Latches are level sensitive storage elements; the action of data storage is dependent on the level (value) of the input clock (or enable) signal. Flip flops are edge sensitive storage elements; the action of data storage is synchronized to either a rising or falling edge of a signal.

SR latch (set reset) latches can be used by the implementation of NOR gates or NAND gates. We will program active low input SR latch cross coupled with NAND gates in the lab session.

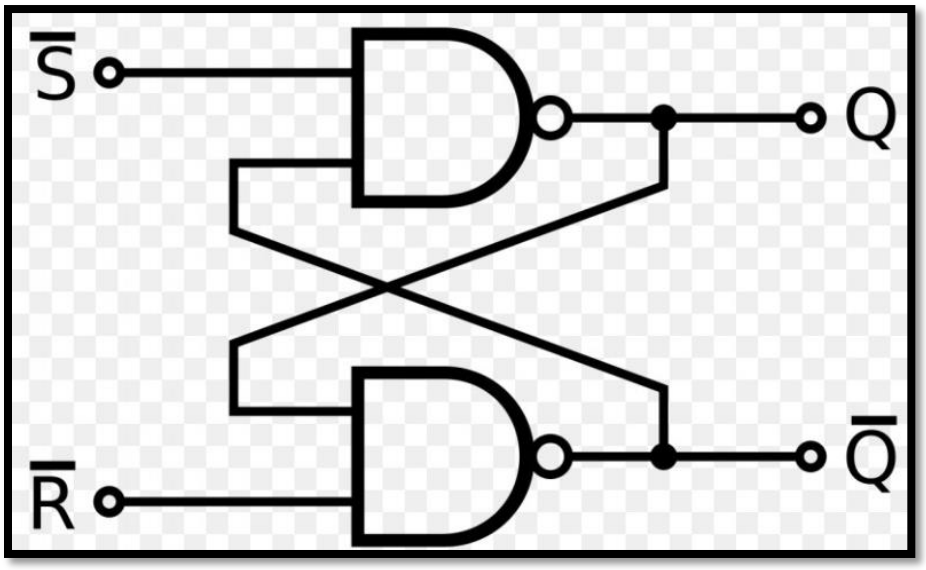


Figure 12.1: Schematic Diagram of SR latch

S’	R’	Q next	Q’next
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	Q’

Figure 12.2: Characteristic Table of active low input SR Latch with Cross Coupled Nand Gates

Modes of Programming

There are two modes of programming into FPGA.

- JTAG
- AS (active serial)

Modes can be selected manually by a two-way sliding button on the board by keeping its position on RUN or PROG respectively. The only advantage of AS mode is to keep your program saved into the flash memory of FPGA. In this mode your program will not be lost even after turning off the device and you may be able to access it again by powering up it again.
For JTAG, set up the switch on the board to RUN position.

Procedure:

Connect USB-blaster cable and power adaptor with ALTERA DE2 board.

1) Open the Quartus II software.
2) Create a new project by selecting “**Create a New Project (New Project Wizard)**” as shown in Figure below.
3) Select a suitable name for your new directory (or you can use the existing one) and also the name of the project and click on **next** option.

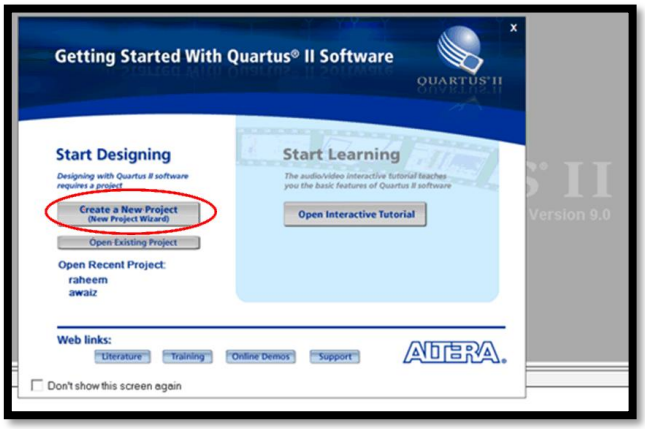


Figure 12.3

4) After creating new directory and project, create a new file by selecting **File** => **New** and select **Verilog HDL File** type as shown in Figure below

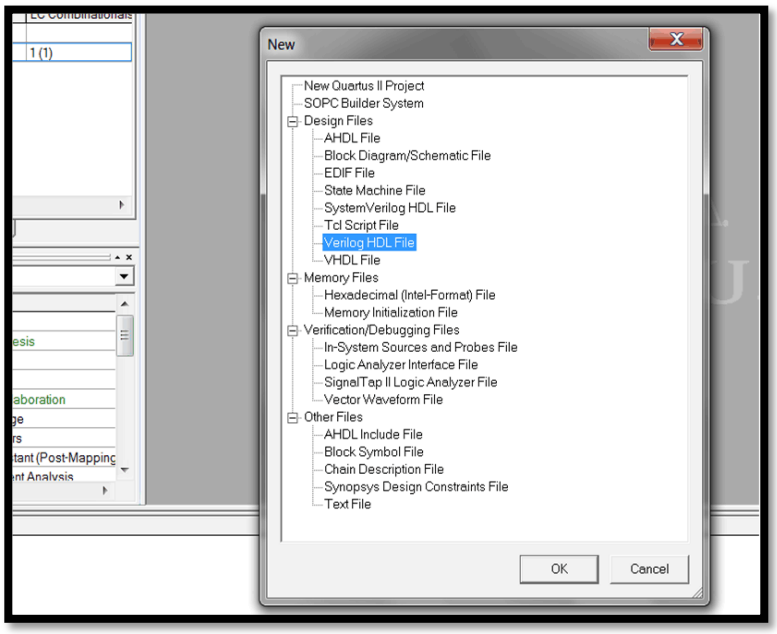


Figure 12.4

5) A command window will appear. Write your program and save it with the same name as given in module command. (Make sure that file should be saved in the same project directory mentioned in step **2**)
6) Now compile your program by selecting **Processing** => **start compilation.**
7) After completion of compilation a message will appear "**full compilation was successful**".
8) To verify your Verilog code on ALTERA board, assign suitable pins/switches/LEDs to your input/output terminals by **Assignments** => **Pin Planner.** A pin planner window will appear

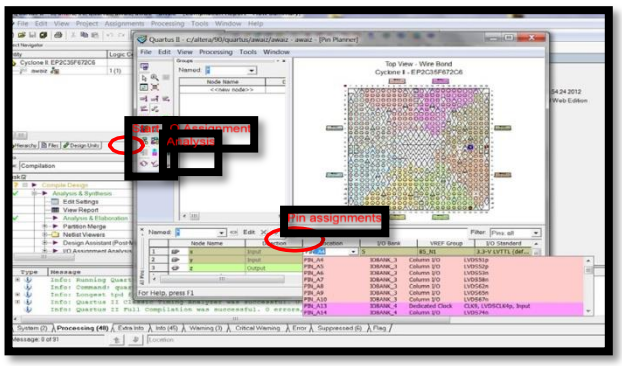


Figure 12.5

9) Assign switches and LEDs to all input and output terminals respectively and start I/O assignment analysis as shown in above figure. The location of ALTERA DE2 board can be

selected from "**ALTERA DE2 user manual**".

10) After I/O assignment analysis, now code is ready to be dumped in ALTERA DE2 board. Select **Tools** => **Programmer** and after selection of USB blaster option select **Start**. A 100% completion message will appear, when program is completely dump.
11) Now, you can test your program on ALTERA DE2 board.

Outcome:

Attach the printout of SR latch observed values.

F/OBEM 01/05/00

NED University of Engineering & Technology
Department of Electronic Engineering

Course Code and Title: _____EL-408 VLSI System Design_____
Laboratory Session No. _____12_____ Date: _____

Software Use Rubric					
Criterion	**Level of Attainment**				
	Below Average (1)	**Average (2)**	**Good (3)**	**Very Good (4)**	**Excellent (5)**
Identification of software menu (syntax, components, commands, tools, layout etc.).	Can't identify software menus.	Rarely identifies software menus.	Occasionally identifies software menus.	Able to identify software menus.	Perfectly able to identify software menus.
Skills to use software (schematic, syntax, commands, tools, layout) efficiently.	Can't use software efficiently.	Rarely uses software efficiently.	Occasionally uses software efficiently.	Often uses software efficiently.	Efficiently uses software (syntax, commands, tools, layout)
Adherence to safety procedures and handling of equipment (computing unit, peripheral devices, and other equipment in lab).	Doesn't handle equipment with required care and safety.	Rarely handles equipment with required care and safety.	Occasionally handles equipment with required care and safety.	Often handles equipment with required care and safety.	Handles equipment with required care and safety.
Ability to troubleshoot software errors (detection and debugging).	Not able to troubleshoot the errors	Rarely able to troubleshoot the errors	Occasionally able to troubleshoot the errors	Often able to troubleshoot the errors	Fully able to troubleshoot the errors
Analysis and interpretation of results/outputs.	Not able to analyze and interpret results/outputs.	Rarely able to perform the analysis and interpretation.	Occasionally able to perform the analysis and interpretation.	Often able to perform the analysis and interpretation.	Perfectly able to perform the analysis and interpretation.

Weighted CLO (Score)	
Remarks	
Instructor's Signature with Date	

LAB SESSION NO 13
OPEN ENDED LAB

Objective:

To practice a Package Sorter by using a Test bench on Verilog (Modelsim).

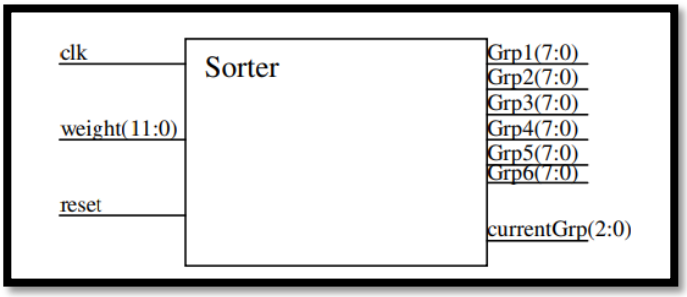
Project Description:

Design a package sorter to classify packages based on their weights and to keep track of packages of different categories. The sorter has an active high asynchronous reset and will keep track of packages since the last reset. Packages should be classified into 6 groups:

Group Configuration

i) between 1 and 200 grams
ii) between 201 and 500 grams
iii) between 501 and 800 grams
iv) between 801 and 1000 grams
v) between 1000 and 2000 grams
vi) greater than 2000

You need to decode weight measurements and classify them into various groups. The input to the circuit will be a 12-bit unsigned binary number (indicating the weight of the package), a clock signal, and a reset. One of the outputs will be currentGrp, a 3-bit unsigned number representing the current group number. There will also be six 8-bit unsigned outputs Grp1-Grp6 representing the number of items weighed in each category since the last reset. The reset line is provided as input to allow these counts to be cleared.



The output lines have the following functionality:

currentGrp[2:0]: Outputs the group number for the weight currently being applied to the sorter. When a weight of zero is applied, it should output a zero. This should update as soon as a package

weight changes and may not necessarily reflect the last group that a package was assigned to.

Grp1-Grp6[7:0]: Outputs the number of objects that have been weighed in each group since the last reset. These outputs should be zero when reset='1'.
Notice that the functionality of the two outputs is such that the description of currentGrp will be purely combinational since it does not depend on any previous inputs. But the description of Grp1-Grp6 will be sequential since it depends not just on the current input but also on the previous inputs.
Any sequential output should change on the falling edge of the clock. Notice that the clk signal will be significantly faster than the duration of the weight signal. As such, you must ensure that the count is only updated once for a given input weight. Secondly, new objects can only be detected and sorted if the weight is allowed to go to zero. This is to ensure that any fluctuations in the weight after it has been sampled are not considered new items. Only the first weight after 0 updates a group count.
Test your design by using a VHDL test bench. The test bench should use arrays (to set the inputs, to store the expected group counts and currentGrp values). Do not just use the example input. It is for illustrating the desired functionality. You are responsible for adequately testing your design, so make sure you test everything described for this problem.
Example input sequence For Configuration 1
Reset ⟶ Put 250grams on ⟶ Takeoff ⟶ Put on 300 grams ⟶ Take off ⟶ Put on 501grams ⟶ Put 512 grams more
[In your waveforms, this input sequence will look like this: reset → 250 → 0 →300 → 0 →501→1013]
At the end of this sequence, the outputs should be:
grp1 = grp4 = grp5 = 0x00
grp2 = 0x02
grp3 = 0x01
currentGrp = 0x5
Note that after 501 grams is sampled in grp3, adding 512 grams only updates the current group and not the grp5 count.53

Deliverables:

1. Name and save the file "your_First_name.extension".
2. You need to demonstrate that you performed the task through simulation.
3. Also, you will be asked questions about various aspects covered in the task.
4. Your ability to answer them and your demonstration will decide your score.
5. Submit a text/doc/pdf file containing the code in report.

F/OBEM 01/05/00

NED University of Engineering & Technology
Department of Electronic Engineering

Course Code and Title: _____EL-408 VLSI System Design_____
Laboratory Session No. _____13_____ Date: _____

Software Use Rubric					
Criterion	**Level of Attainment**				
	Below Average (1)	**Average (2)**	**Good (3)**	**Very Good (4)**	**Excellent (5)**
Identification of software menu (syntax, components, commands, tools, layout etc.).	Can't identify software menus.	Rarely identifies software menus.	Occasionally identifies software menus.	Able to identify software menus.	Perfectly able to identify software menus.
Skills to use software (schematic, syntax, commands, tools, layout) efficiently.	Can't use software efficiently.	Rarely uses software efficiently.	Occasionally uses software efficiently.	Often uses software efficiently.	Efficiently uses software (syntax, commands, tools, layout)
Adherence to safety procedures and handling of equipment (computing unit, peripheral devices, and other equipment in lab).	Doesn't handle equipment with required care and safety.	Rarely handles equipment with required care and safety.	Occasionally handles equipment with required care and safety.	Often handles equipment with required care and safety.	Handles equipment with required care and safety.
Ability to troubleshoot software errors (detection and debugging).	Not able to troubleshoot the errors	Rarely able to troubleshoot the errors	Occasionally able to troubleshoot the errors	Often able to troubleshoot the errors	Fully able to troubleshoot the errors
Analysis and interpretation of results/outputs.	Not able to analyze and interpret results/outputs.	Rarely able to perform the analysis and interpretation.	Occasionally able to perform the analysis and interpretation.	Often able to perform the analysis and interpretation.	Perfectly able to perform the analysis and interpretation.

Weighted CLO (Score)	
Remarks	
Instructor's Signature with Date	