Department of Computer and Software Engineering – ITU

SE201L: Digital Logic Design Lab

Course Instructor: **Ms Aqsa Khalid**	**Dated:** **09-10-2024**
Lab Engineer: **Muhammad Kashif**	**Semester:** **Fall 2024**
Batch: **BSSE 23A**	

LAB 8 Sequential Circuits: Construction and Function of Latch

Name	**Roll Number**	**Lab Marks**
Zunaira Abdul Aziz	BSSE23058	

Checked on: ____________________________

Signature: ____________________________

Sequential Circuits: Construction and Function of Latch

8.1. Introduction

This lab exercise is the first in a series of lab exercises on sequential circuits. It gives students an insight into the principle of working behind a sequential circuit by studying its most basic element: the latch. SR NAND latches with control inputs are studied and constructed using Verilog. A D latch is also constructed using the NAND gate IC.

8.2. Objectives

This lab will enable the students to achieve the following:

- Understand the construction and working of S’R’ latch
- Complete characteristic table by building S’R’ latch using NAND gates
- Derive characteristic equation from the characteristic table of S’R’ latch
- Understand the role of control input in an SR latch and demonstrate its importance
- Construct a D-latch, complete its characteristic table and derive its characteristic equation

8.3. Conduct of Lab

1. Task 1 – 2 of this lab have to be performed using ModelSim and Task 3 using the logic trainer in Embedded Lab.
2. Bring printout of this lab manual when you come to perform the lab.
3. You can work and get evaluated in groups of two. However, manual submission has to be separate.
4. If there is difficulty in understanding any aspect of the lab, please seek help from the lab engineer or the TA.
5. If a lab task contains an instruction to show the work to lab engineer, make sure that the lab engineer evaluates and marks on your manual for that task. If your manual is unmarked for this task, it can result in mark deduction.
6. All tasks must be evaluated within the lab time. Print and attach screenshots of code and timing waveforms with your manual and submit it before the next lab.

8.4. Theory and Background

8.4.1. Sequential circuits

In digital Logic, circuits are of two types: *combinational* and *sequential*. Combinational circuits are digital circuits whose outputs depend only on the present state of inputs. In contrast, sequential circuits are digital circuits that have outputs dependent on both the present state of inputs and previous state of the sequential circuit. This dependence on previous state of the circuit implies that there is a feedback path in sequential circuits. Storage elements are used to store information about the state of sequential circuit at a particular time to be used later. Have a look at the following block diagram.

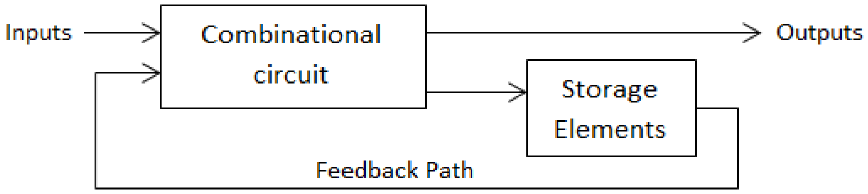
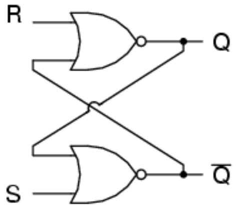


Figure 8.1: Block diagram illustrating the general structure of sequential circuits

8.4.1.1. Latch

A latch is a fundamental storage element of a sequential circuit. It can store one bit of information. An SR latch is constructed with two cross-coupled NOR gates. Circuit diagram and truth table, or the characteristic table as it is called in case of sequential circuits, is as follows:



S	R	Q	$\overline{Q}$
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	0	0

Figure 8.2: SR latch and its truth table

The equation for next state of output Q is:

$Q(t + 1) = S + \overline{R}Q(t)$.

The last operation of the truth table i.e. S = 1, R = 1, is not allowed because it violates the rule that must be the complement of Q.

8.4.2. Implementation of SR latch in Verilog

The following code implements the SR latch of Figure 8.2 using structural modeling style:

```
module srlatch(S, R, Q, Qnot);
input S, R;
output Q, Qnot;
nor #1(Qnot, S, Q);
nor #1(Q, R, Qnot);
endmodule
```

The “#1” written after the use of *nor* primitive indicates the gate delay for a particular operation. This in our case is equal to 1 nanosecond (or whatever time unit has been set in your simulation). Introducing this gate delay is important in sequential circuits as the operation of sequential circuits is ensured by this delay.

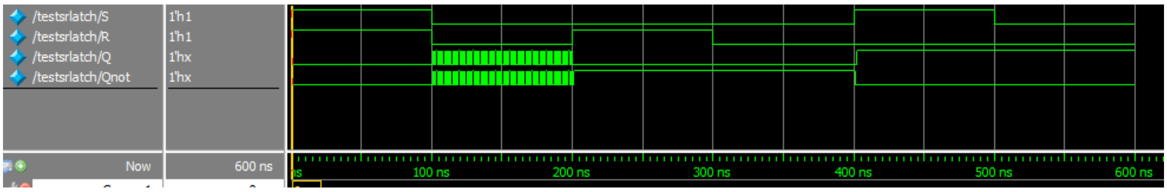
The following code replaces the structural style statements with equivalent dataflow statements:

```
module srlatch(S, R, Q, Qnot);
input S, R;
output Q, Qnot;
assign #1 Qnot = ~(S | Q);
assign #1 Q = ~(R | Qnot);
endmodule
```

Here again, “#1” specifies the delay in the operation.

Simulating both of these codes with the following test bench gives identical timing diagram results which are shown in Figure 8.3.

```
module testsrlatch ();
reg S, R;
wire Q, Qnot;
srlatch mylatch (S, R, Q, Qnot);
initial begin
S = 1; R = 1; #100
S = 0; R = 0; #100
S = 0; R = 1; #100
S = 0; R = 0; #100
S = 1; R = 0; #100
S = 0; R = 0;
end
endmodule
```



/testsrlatch/S 1'h1
/testsrlatch/R 1'h1
/testsrlatch/Q 1'hx
/testsrlatch/Qnot 1'hx

Now 600 ns

Figure 8.3: Timing diagrams for SR latch

Let's look at the various sections of the timing diagram of Figure 8.3.

1. From 0 to 100 ns, we have input S = 1 and R = 1, for which the output is Q = 0, and Qnot = 0. This is the "not allowed" state of SR latch as it violates the condition of Qnot being the negation of Q.
2. From 100 to 200 ns, we have input S = 0 and R = 0, for which the output is not constant, rather oscillates between 1 and 0 after every 1 ns time interval. For S = 0 and R = 0, the output should ideally "latch", which means that the previous output should stay. However, as the previous output is of the "not allowed" state, thus invalid, it cannot be latched. Figure 8.4 shows a magnified snippet of the output in this case.

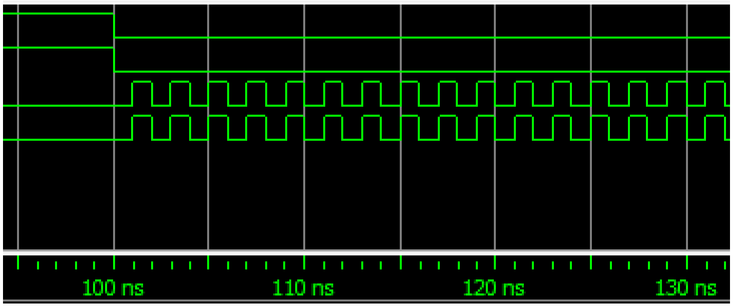


Figure 8.4: Magnified view of invalid output of SR latch due to "not allowed" state

3. From 200 to 300 ns, we have input S = 0 and R = 1. The output in this case is Q = 0 and Qnot = 1. This is the "reset" state.
4. From 300 to 400 ns, we have input S = 0 and R = 0. The previous state is latched in this case.
5. From 400 to 500 ns, we have input S = 1 and R = 0. The output in the case is Q = 1 and Qnot = 0. This is the "set" state.
6. From 500 to 600 ns, we have again input S = 0 and R = 0, for which the previous state is again latched.

8.5. Parts and Equipment

1. Digital logic trainer
2. 7400: Quad 2-input NAND gate IC
3. Connecting wires
4. Software: Modelsim SE

8.6. Lab Tasks

8.6.1. Task 1: S'R' latch [Marks: 8]

1. Build the circuit of NAND latch given in Figure 8.5 using structural or dataflow modeling. [3]

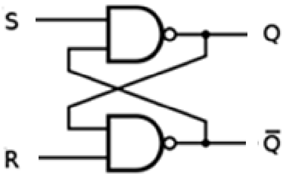


Figure 8.5: S'R' latch

2. Write test bench for your latch module with combinations given in Table 8.1. Do not change the order of these combinations.

```
module sr_latch(S, R, Q, Qnot);

input S, R;

output Q, Qnot;

nand G1(Qnot, S, Q);

nand G2(Q, R, Qnot);

endmodule


module sr_latch_tb;
reg S, R;
wire Q, Qnot;
sr_latch z (.S(S), .R(R), .Q(Q), .Qnot(Qnot));
initial begin
S=0; R=0; #100;
S=0; R=1; #100;
S=1; R=0; #100;
S=1; R=1; #100;
end
initial begin
end
endmodule
```

Fill in the correct values in the table from the timing diagram: [2]

Table 8.1: Test bench combinations for the S'R' latch

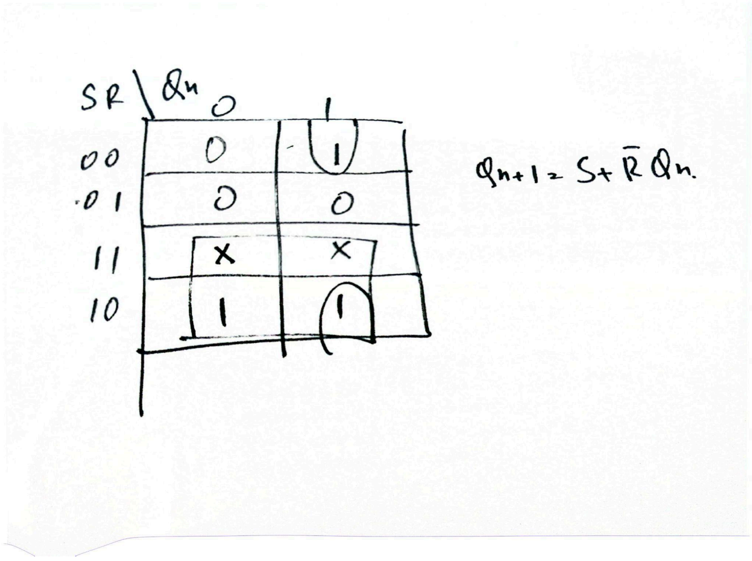
S	R	Q(t+1)	Q'(t+1)
0	0	1	1
1	1	1	0
0	1	1	0
1	1	0	1
1	0	0	1
1	1	1	0

3. From the output values noted in Table 8.1, populate the characteristic table of this latch in Table 8.2: [1]

Table 8.2: Characteristic table of S'R' latch

S	R	Q(t+1)	Q'(t+1)	State
0	0	1	1	Invalid
0	1	1	0	0
1	0	0	1	1
1	1	1	0	Hold

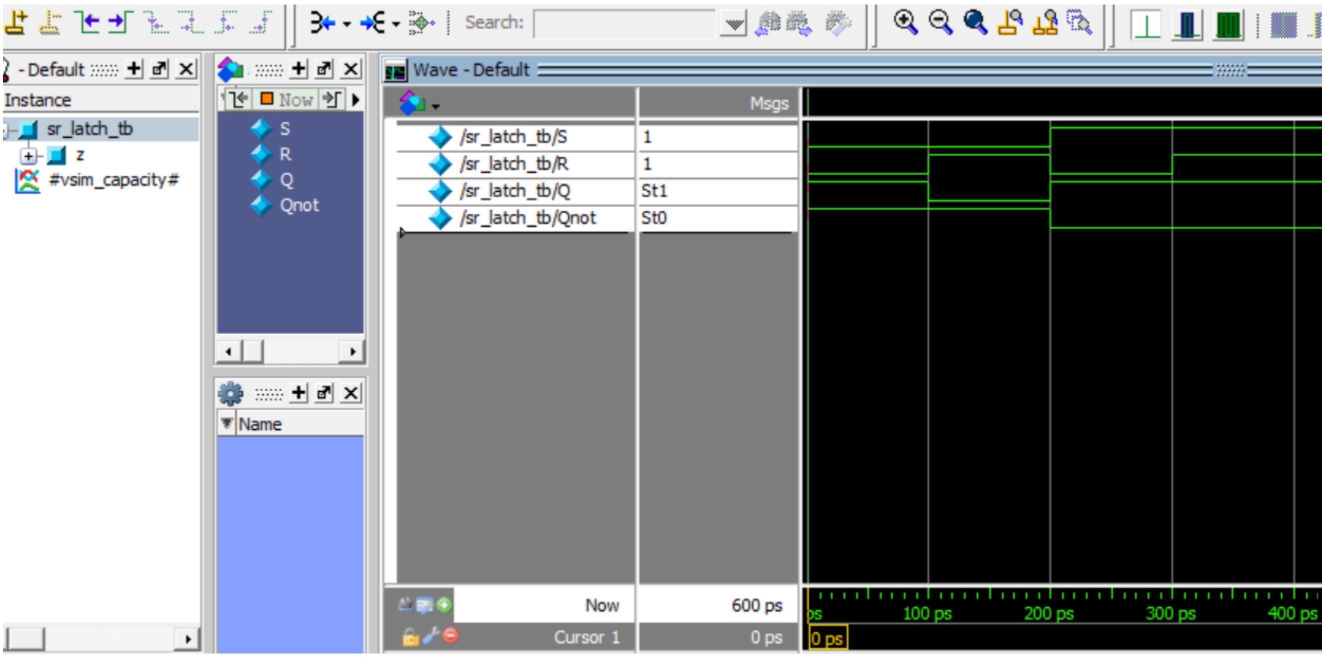
4. Derive the characteristic equation from the above characteristic table. Show derivation here: [2]



SR \ Q_n	0	1
00	0	1
01	0	0
11	X	X
10	1	1

$$Q_{n+1} = S + \bar{R} Q_n$$

5. Give answer to question 1 of Analysis.
6. Take clear screenshots of the code of all modules of this task and the simulation timing waveforms, and paste them in a WORD file such that all fit on no more than two pages. Make sure that screenshots are legible.



	Msgs
/sr_latch_tb/S	1
/sr_latch_tb/R	1
/sr_latch_tb/Q	St1
/sr_latch_tb/Qnot	St0
Now	600 ps
Cursor 1	0 ps

8.6.2. Task 2: NAND latch with enable input [Marks: 6]

1. The output of the latch constructed in last task is affected whenever there is a change in its inputs. This behavior can sometimes be undesirable and it is modified by introduction of a *control/enable* input as shown in Figure 8.6 such that the control input determines when the state of the latch is to be affected. Modify your code to implement the latch of Figure 8.6 with control input E. You may choose any modeling style; structural or dataflow. [2]

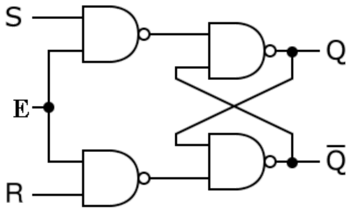


Figure 8.6: Latch with enable input

2. Now, write a test bench and fill its characteristic table in Table 8.3 according to the simulation results of test bench. Paste code and screenshot of simulation results in the textbox given below: [2]

```
module sr_latch_with_enable(E, S, R, Q, Qnot);
   input E, S, R;
   output Q, Qnot;
   assign Qnot = ~(E & (S & Q));
   assign Q = ~(E & (R & Qnot));
endmodule

module sr_latch_with_enable_tb;
reg E, S, R;
wire Q, Qnot;
sr_latch_with_enable z(.E(E), .S(S), .R(R), .Q(Q), .Qnot(Qnot));
initial begin
   E=0; S=0; R=0; #100;
   E=1; S=0; R=0; #100;
   E=1; S=0; R=1; #100;
   E=1; S=1; R=0; #100;
   E=1; S=1; R=1; #100;
end
endmodule
```

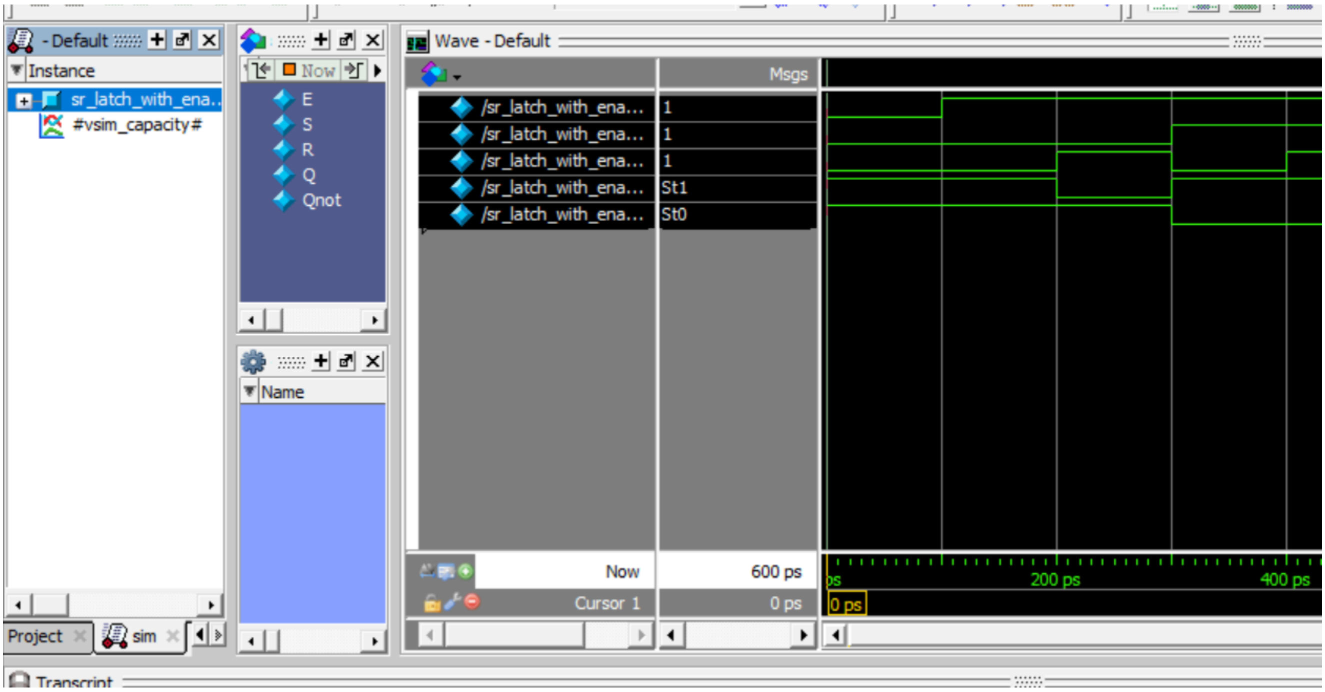
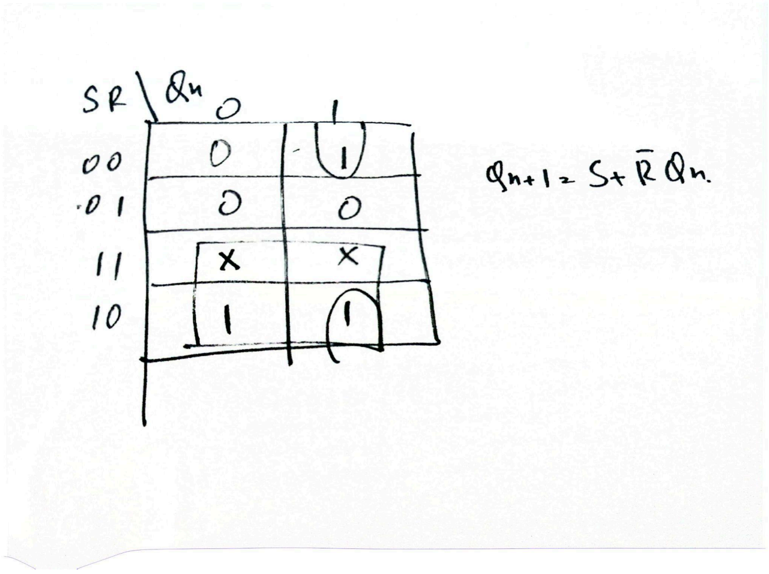


Table 8.3: Characteristic table of latch with enable E

E	S	R	Q(t+1)	Q'(t+1)	State
0	0	0	**0**	**1**	**no change**

0	0	1	**0**	**1**	**no change**
0	1	0	**0**	**1**	**hold**
0	1	1	**0**	**1**	**hold**
1	0	0	**1**	**0**	**hold**
1	0	1	**0**	**1**	**0**
1	1	0	**1**	**0**	**1**
1	1	1	**1**	**1**	**invalid**

3. Derive the characteristic equation including E as an input from the above characteristic table here: [2]



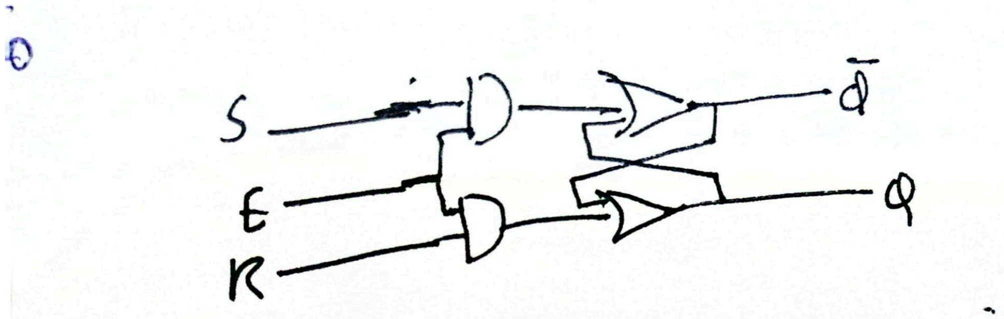
SR \ Q_n	0	1
00	0	1
01	0	0
11	X	X
10	1	1

$Q_{n+1} = S + \bar{R} Q_n.$

4. Give answers to questions 2 and 3 of Analysis.

8.6.3. Task 3: D-latch [Marks: 6]

1. To avoid the “not allowed” combination of S and R in an SR latch, only one data input “D” is used at S, and its complement is fed in place of R. Such a latch is called a D-latch. Modify the circuit of Figure 8.6 to implement a D-latch according to this description and draw here: [1]



2. Build your proposed D-latch circuit on breadboard using the NAND gate IC.
3. Show working circuit to the lab engineer to obtain credit. Fill in its characteristic table in Table 8.4. [4]

Table 8.4: Characteristic table of D-latch

E	D	Q(t+1)	Q’(t+1)
0	X	Q(t)	Q’(t)
1	0	0	1
1	1	1	0

4. Derive the characteristic equation of D-latch including E from the above characteristic table here: [1]

$$\overset{11}{ED} + \overset{0x}{\bar{E}Q(t)}.$$

$$Q(t+1) = ED + (E' \cdot Q(t))$$

8.6.4. Analysis [Marks: 5]

1. Can you guess why this latch has been called S'R' latch in the title of this task (Task 1)? Give reason according to the characteristic table. [2]

The latch is called S'R' latch because D replaces **S** and D replaces R, ensuring no invalid state.

2. What will the equation be if E is always 1? Compare this equation with the Task 1 equation. [2]

If E=1, the equation simplifies to Q(t+1)=D which is the same as the simplified version of the Task 1 equation.

3. Why can this latch be now called an SR latch? [1]

This latch can be called an SR latch because it mimics the behavior of an SR latch, where D and D' replace S and R.

Assessment Rubric

Method:
Lab report evaluation and instructor observation during lab sessions.

Performance	CLO	Able to complete the tasks over 80% (4-5)	Able to complete the tasks 50 – 80% (2-3)	Tasks completion below 50% (0-1)	Marks
1. Teamwork	1	Actively engages and cooperates with other group members in an effective manner	Cooperates with other group members in a reasonable manner	Distracts or discourages other group members from conducting the experiments	
2. Laboratory safety and disciplinary rules	1	Observes lab safety rules; handles the development board and other components with care and adheres to the lab disciplinary guidelines aptly	Observes safety rules and disciplinary guidelines with minor deviations	Disregards lab safety and disciplinary rules	
3. Realization of experiment	3	Conceptually understands the topic under study and develops the experimental setup accordingly	Needs guidance to understand the purpose of the experiment and to develop the required setup	Incapable of understanding the purpose of the experiment and consequently fails to develop the required setup	
4. Conducting experiment	3	Sets up hardware/software properly according to the requirement of experiment and examines the output carefully	Makes minor errors in hardware/software setup and observation of output	Unable to set up experimental setup, and perform the procedure of experiment	
5. Data collection	3	Completes data collection from the experiment setup by giving proper inputs and observing the outputs, complies with the instructions regarding data entry in manual	Completes data collection with minor errors and enters data in lab report with slight deviations from provided guidelines	Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab report properly	
6. Data analysis	3	Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy in data from theory with sound explanation, where asked	Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any discrepancy in data from theory	Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data	
7. Computer use	3	Successfully uses lab PC and internet to look for relevant datasheets, carry out calculations, or verify results using simulation	Requires assistance in looking for IC datasheets and carrying out calculation and simulation tasks	Does not know how to use computer to look up datasheets or carry out calculation and simulation tasks	
				Total (out of 35)	