Experiment No 7

Design of Combinational Logic Using Multiplexers and Decoders

7.1 Objectives:

After completing this experiment, student will be able to:

- Describe construction and operational principle of digital decoder.
- Design logic circuits using decoder.
- Describe construction and operational principle of digital multiplexers.
- Design logic circuits using multiplexers.

7.2 Background Theory

A decoder is a circuit that changes a code into a set of signals. It is called a decoder because it does the reverse of encoding, we will begin our study with decoders because they are simpler to design. Multiplexing is the generic term used to describe the operation of sending one or more analogue or digital signals over a common transmission line at different times or speeds and as such, the device we use to do just that is called a **Multiplexer**.

7.2.1 Decoders

A decoder is a logic circuit that will detect the presence of a specific binary number or word. The input to the decoder is a parallel binary number and the output is a binary signal that indicates the presence or absence of that specific number. It is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.

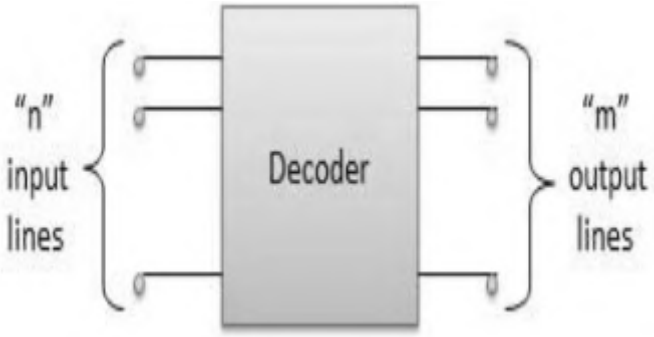


Figure 7.6: Logic Diagram of Decoder

7.2.1.1 2-to-4 Decoder

The 2-to-4 line binary decoder consists of an array of four AND gates. The 2 binary inputs labeled A and B are decoded into one of 4 outputs, hence the description of 2-to-4 binary decoder. Each output represents one of the minterms of the 2 input variables, (each output = a minterm).

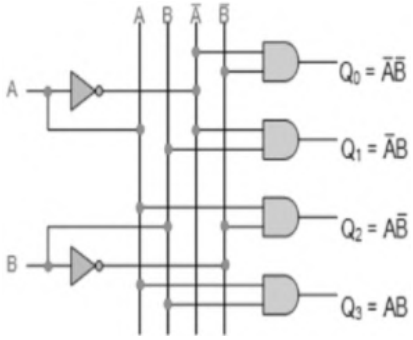
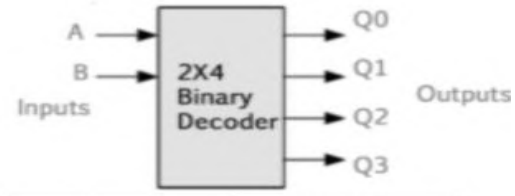


Figure 7.7: Circuit Diagram of 2-to-4 Decoder

The binary inputs A and B determine which output line from Q0 to Q3 is "HIGH" at logic level "1" while the remaining outputs are held "LOW" at logic "0" so only one output can be active (HIGH) at any one time.



A	B	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Figure 7.8: Logic Diagram and Truth table of 2-to-4 Decoder

Therefore, whichever output line is "HIGH" identifies the binary code present at the input, in other words it "decodes" the binary input. Some binary decoders have an additional input pin labeled "Enable" that controls the outputs from the device.

This extra input allows the decoders outputs to be turned "ON" or "OFF" as required. Output is only generated when the Enable input has value 1; otherwise, all outputs are 0. Only a small change in the implementation is required: the Enable input is fed into the AND gates which produce the outputs.

If Enable is 0, all AND gates are supplied with one of the inputs as 0 and hence no output is produced. When Enable is 1, the AND gates get one of the inputs as 1, and now the output depends upon the remaining inputs. Hence the output of the decoder is dependent on whether the Enable is high or low.

7.2.1.2 3-to-8 Decoder Implementation

In 3 to 8 line decoder, it includes three inputs and eight outputs. Here the inputs are represented through A, B & C whereas the outputs are represented through D0, D1, D2…D7.

The selection of 8 outputs can be done based on the three inputs. So, the truth table of this 3 line to 8 line decoder is shown below. From the following truth table, we can observe that simply one of 8 outputs from DO – D7 can be selected depending on 3 select inputs.

Table 7.2: Truth Table of 3 to 8 Decoder

Sr No	**Inputs**			**Outputs**							
1	**A**	**B**	**C**	**D0**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
2	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	0	1	0	0	0	0	0	0
4	0	1	0	0	0	1	0	0	0	0	0
5	0	1	1	0	0	0	1	0	0	0	0
6	1	0	0	0	0	0	0	1	0	0	0
7	1	0	1	0	0	0	0	0	1	0	0
8	1	1	0	0	0	0	0	0	0	1	0
9	1	1	1	0	0	0	0	0	0	0	1

From the above truth table of 3 lines to 8 line decoder, the logic expression can be defined as

D0 = A’B’C’

D1= A’B’C

D2 = A’BC’

D3 = A’BC

D4 = AB’C’

D5= AB’C

D6 = ABC’

D7 = ABC

From the above Boolean expressions, the implementation of 3 to 8 decoder circuit can be done with the help of three NOT gates & 8-three input AND gates.

In the above circuit, the three inputs can be decoded into 8 outputs, where every output represents one of the midterms of the three input variables.

The 3 inverters in the above logic circuit will provide the complement of the inputs & each one of the AND gates will generate one of the midterms.

This kind of decoder mainly used to decode any 3-bit code & generates eight outputs, equivalent to 8 different combinations for the input code.

This decoder is also known as a binary to octal decoder because the inputs of this decoder represent three-bit binary numbers whereas the outputs represent the 8 digits within the octal number system.

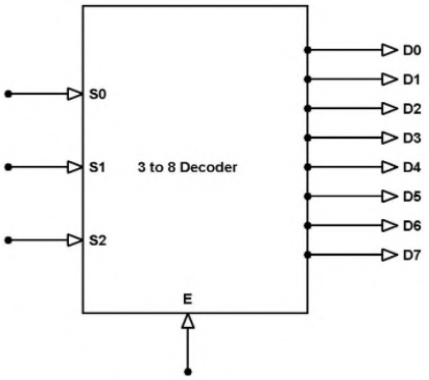


Figure 7.9: 3-to-8 Decoder Block Diagram

This decoder circuit gives 8 logic outputs for 3 inputs and has a enable pin. The circuit is designed with AND and NAND logic gates. It takes 3 binary inputs and activates one of the eight outputs. 3 to 8 line decoder circuit is also called a binary to an octal decoder.

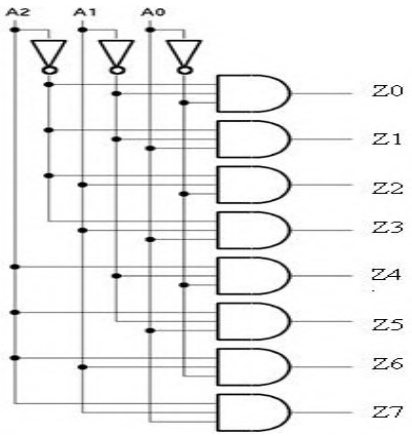


Figure 7.10: 3-to-8 Decoder Circuit

The decoder circuit works only when the Enable pin (E) is high. S0, S1 and S2 are three different inputs and D0, D1, D2, D3. D4. D5. D6. D7 are the eight outputs. The logic diagram of the 3 to 8 line decoder is shown above.

The implementation of this 3 line to 8 line decoder can be done using two 2 lines to 4 line decoders. The block diagram is shown below by using two 2 to 4 decoders.

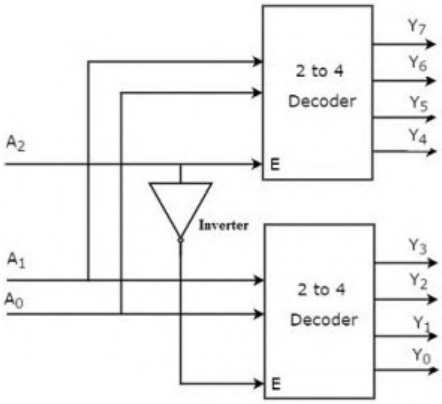


Figure 7.11: 3-to-8 Decoder using 2-to-4 Decoder

7.2.1.3 Boolean Function Implementation with Decoder

Boolean function in sum of minterms form can be represented using decoders and external OR gates. Implement the following Boolean function using 3 to 8 decoders.

$$S(x, y, z) = \sum (1, 2, 4, 7)$$

$$C(x, y, z) = \sum (3, 5, 6, 7)$$

Three bit adder Truth table is shown below.

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure 7.12: Truth Table of 3 bit Adder

Implementation of three bit adder with 3 to 8 decoder is shown below.

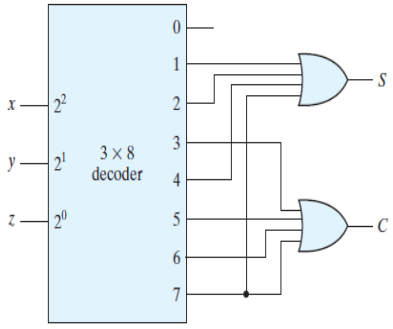


Figure 7.13: Circuit of 3 bit Adder

7.2.2 Multiplexers

In electronics, a multiplexer (or mux) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2^n inputs has n select lines, which are used to select which input line to send to the output.

- A 2^n-to-1 multiplexer sends one of 2^n input lines to a single output line.
- A multiplexer has two sets of inputs:
  - 2^n data input lines
  - n select lines, to pick one of the 2^n data inputs
- The mux output is a single bit, which is one of the 2^n data inputs.

7.2.2.1 4-to-1 Multiplexer

A 4-to-1 multiplexer consists four data input lines as I0 to I3, two select lines as S0 and S1 and a single output line Y. The select lines S0 and S1 select one of the four input lines to connect the output line. The figure below shows the block diagram of a 4-to-1 multiplexer in which, the multiplexer decodes the input through select line.

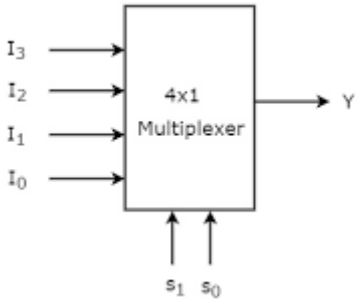


Figure 7.14: 4-to-1 Multiplexer

The truth table of a 4-to-1 multiplexer is shown below in which four input combinations 00, 10, 01 and 11 on the select lines respectively switches the inputs I0, I2, I1 and I3 to the output. That means when S0=0 and S1 =0, the output at Y is D0, similarly Y is I1 if the select inputs S0=0 and S1= 1 and so on.

Table 7.2: Truth Table of 4-to-1 Multiplexer

Sr. No	S0	S1	I0	I1	I2	I3	Y
1	0	0	0	X	X	X	0
2	0	0	1	X	X	X	1
3	0	1	X	0	X	X	0
4	0	1	X	1	X	X	1
5	1	0	X	X	0	X	0
6	1	0	X	X	1	X	1
7	1	1	X	X	X	0	0
8	1	1	X	X	X	1	1

From the above truth table, we can write the output expressions as follows:

$$Y = S0\ S1\ I0 + S0\ S1\ I1 + S0\ S1\ I2 + S0\ S1\ I3$$

From the above expression of the output, a 4-to-1 multiplexer can be implemented by using basic logic gates. The below figure shows the logic circuit of 4:1 MUX which is implemented by four 3-inputs AND gates, two 1-input NOT gates, and one 4-inputs OR gate.

In this circuit, each data input line is connected as input to an AND gate and two select lines are connected as other two inputs to it. Additionally, there is also an Enable Signal. The output of all the AND gates are connected to inputs of OR gate in order to produce the output Y.

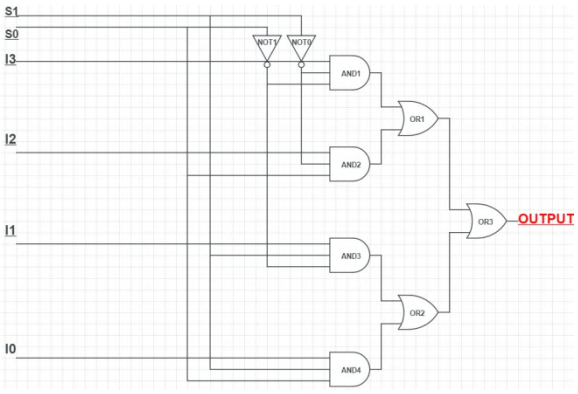


Figure 7.15: Logic circuit of 4-to-1 Multiplexer

7.2.2.2 8-to-1 Multiplexer Implementation

An 8-to-1 multiplexer consists of eight data inputs D0 through D7, three input select lines S0 through S2 and a single output line Y. Depending on the select lines combinations, multiplexer selects the inputs.

The below figure shows the block diagram of an 8-to-1 multiplexer with enable input that can enable or disable the multiplexer. Since the number data bits given to the MUX are eight, then 3 bits (23 = 8) are needed to select one of the eight data bits.

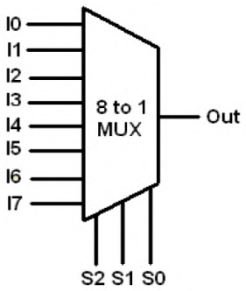


Figure 7.16: 8-to-1 Multiplexer

The truth table for an 8-to1 multiplexer is given below with eight combinations of inputs so as to generate each output corresponds to input.

For example, if S2= 0, S1=1 and S0=0 then the data output Y is equal to D2. Similarly the data outputs D0 to D7 will be selected through the combinations of S2, S1 and S0 as shown in below figure.

Table 7.3: Truth Table of 8-to-1 Multiplexer

Sr. No	S0	S1	S2	I0	I1	I2	I3	I4	I5	I6	I7	Y
1	0	0	0	0	X	X	X	X	X	X	X	0
2	0	0	0	1	X	X	X	X	X	X	X	1
3	0	0	1	X	0	X	X	X	X	X	X	0
4	0	0	1	X	1	X	X	X	X	X	X	1
5	0	1	0	X	X	0	X	X	X	X	X	0
6	0	1	0	X	X	1	X	X	X	X	X	1
7	0	1	1	X	X	X	0	X	0	0	0	0

8	0	1	1	X	X	X	1	X	1	1	1	1
9	1	0	0	X	X	X	X	0	X	X	X	0
10	1	0	0	X	X	X	X	1	X	X	X	1
11	1	0	1	X	X	X	X	X	0	X	X	0
12	1	0	1	X	X	X	X	X	1	X	X	1
13	1	1	0	X	X	X	X	X	X	0	X	0
14	1	1	0	X	X	X	X	X	X	1	X	1
15	1	1	1	X	X	X	X	X	X	X	0	0
16	1	1	1	X	X	X	X	X	X	X	1	1

From the above truth table, the Boolean equation for the output is given as:

Y = S0 S1 S2 I0 + S0 S1 S2 I1 + S0 S1 S2 I2 + S0 S1 S2 I3 + S0 S1 S2 I4 + S0 S1 S2 I5 + S0 S1 S2 I6 + S0 S1 S2 I7

From the above Boolean equation, the logic circuit diagram of an 8-to-1 multiplexer can be implemented by using 8 AND gates, 1 OR gate and 7 NOT gates as shown in below figure. In the circuit, when enable pin is set to one, the multiplexer will be disabled and if it is zero, then select lines will select the corresponding data input to pass through the output.

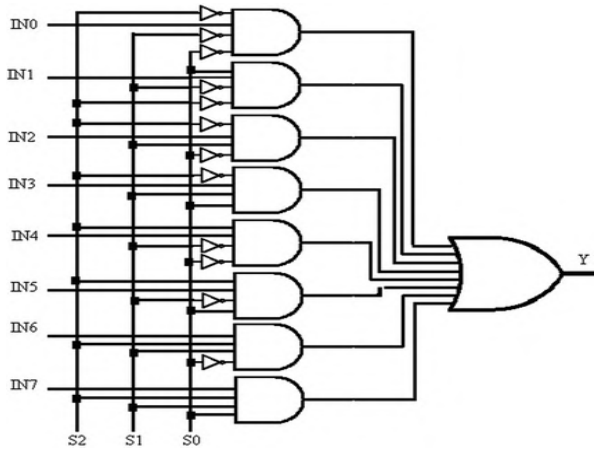


Figure 7.17: Logic circuit of 8-to-1 Multiplexer

If you observe the Boolean Expression of 8-to-1 Multiplexer shown above, we can re-write it as follows:

Y = S0 S1 S2 I0 + S0 S1 S2 I1 + S0 S1 S2 I2 + S0 S1 S2 I3 + S0 S1 S2 I4 + S0 S1 S2 I5 + S0 S1 S2 I6 + S0 S1 S2 I7

Y = S0 (S1 S2 I0 + S1 S2 I1 + S1 S2 I2 + S1 S2 I3) + S0(S1 S2 I4 + S1 S2 I5 + S1 S2 I6 + S1 S2 I7)

The expression in the first bracket i.e., S1 S2 I0 + S1 S2 I1 + S1 S2 I2 + S1 S2 I3 is similar to the Boolean Expression of a 4-to-1 Multiplexer with I0, I1, I2 and I3 as inputs and S1 and S2 as Select Lines. Let this expression be P1.

Similarly, the expression in the second bracket i.e., S1 S2 I4 + S1 S2 I5 + S1 S2 I6 + S1 S2 I7 is similar to the Boolean Expression of another 4-to-1 Multiplexer with D4, I5, I6 and I7 as inputs and S1 and S2 as Select Lines. Let this expression be P2.

Now, replacing the above expressions with P1 and P2, we get,

S0 P1 + S0 P2

This expression is similar to a 2-to-1 Multiplexer with P1 and P2 (where, P1 and P2 are outputs of respective 4-to-1 Multiplexers) as Inputs and S0 as Select Signal. So, finally, we can deduce that an 8-to-1 Multiplexer can be implemented using two 4-to-1 Multiplexers and one 2-to-1 Multiplexer. The block diagram of the same is shown below:

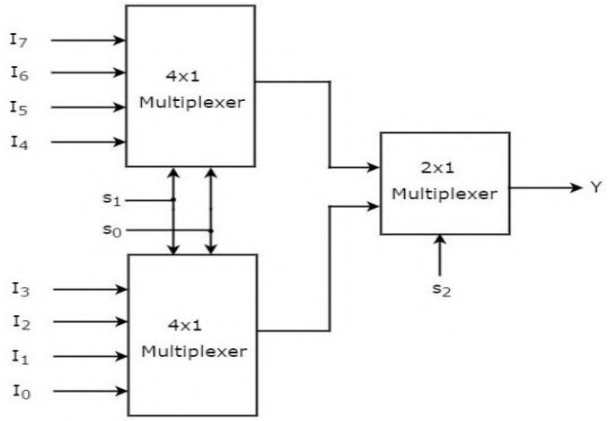


Figure 7.18: 8-to-1 Multiplexer using 4-to-1 Multiplexer

7.2.2.3 Boolean Function Implementation with Multiplexer

A multiplexer with (n-1) selection lines can be used to implement any Boolean Function with n variables.

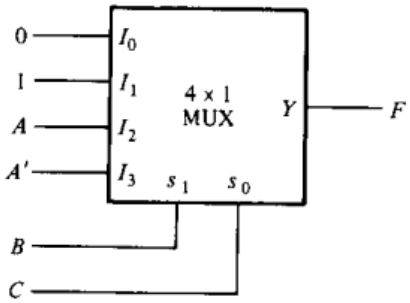
Procedure:

1. Express the Boolean Function in sum of minterms
2. Connect (n-1) input variables to selection lines of the Mux
3. Make an implementation table that contains Mux inputs as the columns
4. List all the possible minterms in two rows. First row contains minterm with nth input variable as 0. Second row contains minterms with nth variable as 1.
5. Circle all the minterms of the function in two rows of the implementation table
6. If both minterms in a column are circled, apply 1 to that input of the Mux
7. If none of the minterms in a colum is circled, apply 0 to that input of the Mux
8. If the bottom minterm in a column is circled and top is not, apply nth variable to that input of Mux
9. If the top minterm in a column is circled and the bottom is not, apply complement of nth variable to that input of Mux

Implement the following Boolean function using a 4 to 1 Mux :

$$F(A, B, C) = \Sigma (1, 3, 5, 6)$$

	I_0	I_1	I_2	I_3
A'	0	①	2	③
A	4	⑤	⑥	7
	0	1	A	A'



Minterm	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

7.3 Lab Activities

7.3.1 Task-1: Boolean function Implementation using 3-to-8 Decoder:

Design a logic circuit for the Boolean function below and Implement using a suitable decoder. Also perform truth table based verification.

F (A, B, C) = Σ (1, 2, 4, 7)

7.3.2 Task-2: Design a 3-to-8 line Decoder using 2-to-4 line Decoder:

Implement a 3-to-8 decoder using 2-to-4 line decoder and perform truth table based verification.

7.3.3 Task-3: Boolean function Implementation using 8-to-1 Multiplexers:

Design a logic circuit for the Boolean function below and Implement using a suitable multiplexer. Also perform truth table based verification.

F (A, B, C, D) = Σ m (0, 1, 3, 4, 8, 9, 15)

Lab Exercise and Summary

Summary should cover Introduction, Procedure, Data Analysis and Evaluation.

Student's Signature: _____ **Date: _____**

LABORATORY SKILLS ASSESMENT (Psychomotor)

Total Marks: 100

Criteria (Max Marks)	Level 1 0% ≤ S < 50%	Level 2 50% ≤ S< 70%	Level 3 70% ≤ S< 90%	Level 4 90%≤ S ≤100%	Score (S)
Procedural Awareness (20)	Selects inappropriate skills and/or strategies required by the task	Selects and applies appropriate skills and/or strategies required by the task with some errors	Selects and applies the appropriate strategies and/or skills specific to the task without significant errors	Selects and applies appropriate strategies and/or skills specific to the task without any error	
Practical Implementation (30)	Makes several critical errors in applying procedural knowledge of using multiplexers and decoders	Makes few critical errors in applying procedural knowledge of using multiplexers and decoders	Makes some non-critical errors in applying procedural knowledge of using multiplexers and decoders	Applies the procedural knowledge of using multiplexers and decoders in perfect ways	
Safety (10)	Requires constant reminders to follow safety procedures	Requires some reminders to follow safety procedures	Follows safety procedures with only minimal reminders	Routinely follows safety procedures	
Use of Tool/Equipment (20)	Uses tools, equipment and materials with limited competence	Uses tools, equipment and materials with some competence	Uses tools, equipment and materials with considerable competence	Uses tools, equipment and materials with a high degree of competence	
Participation to Achieve Group Goals (10)	Shows little commitment to group goals and fails to perform assigned roles	Demonstrates commitment to group goals, but has difficulty performing assigned roles	Demonstrates commitment to group goals and carries out assigned roles effectively	Actively helps to identify group goals and works effectively to meet them in all roles assumed	
Interpersonal Skills in Group Work (10)	Rarely interacts positively within a group, even with prompting	Interacts with other group members if prompted	Interacts with all group members spontaneously	Interacts positively with all group members and encourages such interaction in others	
				Marks Obtained	

Instructor's Signature: _____ **Date: _____**

LABORATORY SKILLS ASSESMENT (Affective)

Total Marks: 40

Criteria (Max. Marks)	Level 1 0% ≤ S < 50%	Level 2 50% ≤ S < 70%	Level 3 70% ≤ S < 90%	Level 4 90% ≤ S ≤ 100%	Score (S)
Introduction (5)	Very little background information provided or information is incorrect	Introduction is brief with some minor mistakes	Introduction is nearly complete, missing some minor points	Introduction complete and well-written; provides all necessary background principles for the experiment	
Procedure (5)	Many stages of the procedure are not entered on the lab report.	Many stages of the procedure are entered on the lab report.	The procedure could be more efficiently designed but most stages of the procedure are entered on the lab report.	The procedure is well designed and all stages of the procedure are entered on the lab report.	
Data Record (10)	Data is brief and missing significant pieces of information.	Data provides some significant information and has few critical mistakes.	Data is almost complete but has some minor mistakes.	Data is complete and relevant. Tables with units are provided. Graphs are labeled. All questions are answered correctly.	
Data Analysis (10)	Data is presented in very unclear manner. Error analysis is not included.	Data is presented in ways (charts, tables, graphs) that are not clear enough. Error analysis is included.	Data is presented in ways (charts, tables, graphs) that can be understood and interpreted. Error analysis is included.	Data are presented in ways (charts, tables, graphs) that best facilitate understanding and interpretation. Error analysis is included.	
Report Quality (10)	Report contains many errors.	Report is somewhat organized with some spelling or grammatical errors.	Report is well organized and cohesive but contains some grammatical errors.	Report is well organized and cohesive and contains no grammatical errors. Presentation seems polished.	
				Marks Obtained	

LABORATORY SKILLS ASSESSMENT (Cognitive)

Total Marks: 10

(If any) **Marks Obtained**	

Instructor's Signature: _____ **Date: _____**