Experiment No 6

Implementation of Adder and Magnitude Comparator

6.1 Objectives:

After completing this experiment, student will be able to:

- Describe design and operational principle of Adders/Subtractors.
- Design logic circuits using half and Full Adders.
- Describe design and operational principle of Magnitude comparator.
- Design n-bit comparator and verify its operation.

6.2 Background Theory

Arithmetic Logic Unit (ALU) forms an integral part of digital circuits. It is the fundamental building block of the central processing unit of any computing devices. It carries out mathematical, logical, and decision making operations in a computer. Adders are the most essential component of computational circuits. Most of the complex arithmetic circuits are based on the addition. Adders play a major role in the functioning of an ALU.

6.2.1 Adder

There are four types of Adders:

- Half Adder
- Full Adder
- N-bit Adder
- N-bit Adder Subtractor

6.2.1.1 Half Adder

The half adder accepts two binary digits on its inputs and produce two binary digits outputs, a sum bit and a carry bit. The half adder is an example of a simple, functional digital circuit built from two logic gates. As shown in figure 6.1, the half adder adds to one-bit binary numbers (AB). The output is the sum of the two bits (S) and the carry (C).

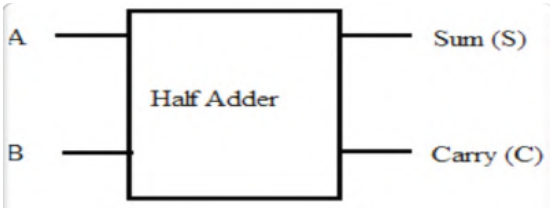

Figure 6.1: Half Adder Block Diagram

Table 6.1 Half Adder Truth Table

A	B	Sum	Carry-Out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

By using Table 6.1 below given Sum and Carry-out equations are obtained

$$Sum = A \oplus B$$

$$Carry - out = A.B$$

Note that how the same two inputs are directed to two different gates. The inputs to the XOR gate are also the inputs to the AND gate. The input "wires" to the XOR gate are tied to the input wires of the AND gate; thus, when voltage is applied to the A input of the XOR gate, the A input to the AND gate receives the same voltage.

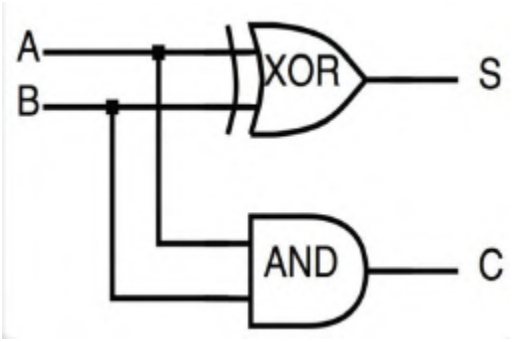

Figure 6.2 Half Adder Combinational Circuit Diagram

6.2.1.2 Full Adder

The full adder accepts two inputs bit and an input carry and generates a sum output and an output carry. The full-adder circuit adds three one-bit binary numbers (Cin, A, B) and outputs two one-bit binary numbers, a sum(S) and a carry (Cout). The full-adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. binary numbers.

Table 6.2 Full Adder Truth Table

A	B	Cin	Sum	Carry-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

By using Table 6.2 below given Sum and Carry-out equations are obtained

$$Sum = A \oplus B \oplus Cin$$

$$Carry - out = A \,.\, B + (\, A \oplus B)$$

Carry-in is a possible carry from a less significant digit, while a Carry-out represents a carry to a more significant digit. In many ways, the full adder can be thought of as two half adders connected together, with the first half adder passing its carry to the second half adder as shown.

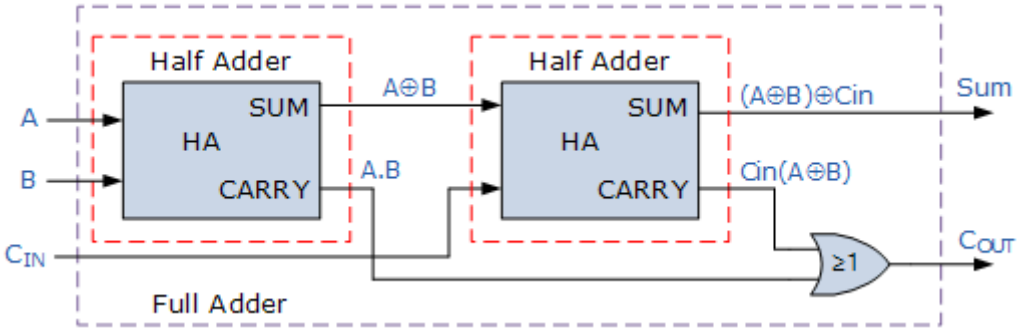

Figure 6.3 Full Adder Logic Diagram

6.2.1.3 N-Bit Adder

We have seen above that single 1-bit binary adders can be constructed from basic logic gates. But what if we wanted to add together two n-bit numbers, then n number of 1-bit full adders need to be connected or "cascaded" together to produce what is known as a **Ripple Carry Adder.**

By connecting ‘n’ number of full adders in parallel, an n-bit Parallel Adder can be constructed. From the below figure, it is to be noted that there is no carry at the least significant position, hence we can use either a half adder or make the carry input of full adder as zero at this position.

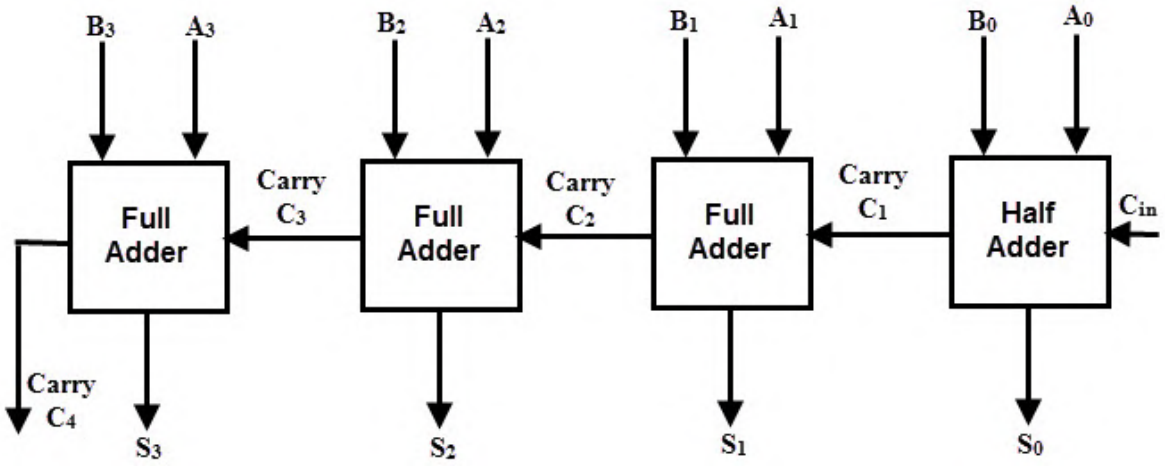

Figure 6.4: 4-Bit adder

It is called a ripple carry adder because the carry signal produce a “ripple” effect through the binary adder from right to left, (LSB to MSB) as shown in figure 6.4.

6.2.1.4 N-Bit Adder Subtractor

The operations of both addition and subtraction can be performed by a one common binary adder. Such binary circuit can be designed by adding an Ex-OR gate with each full adder as shown in below figure. The figure below shows the 4 bit parallel binary adder/subtractor which has two 4 bit inputs as ‘A3 A2 A1 A0‘ and ‘B3 B2 B1 B0‘ .

The mode input control line M is connected with carry input of the least significant bit of the full adder. This control line decides the type of operation, whether addition or subtraction.

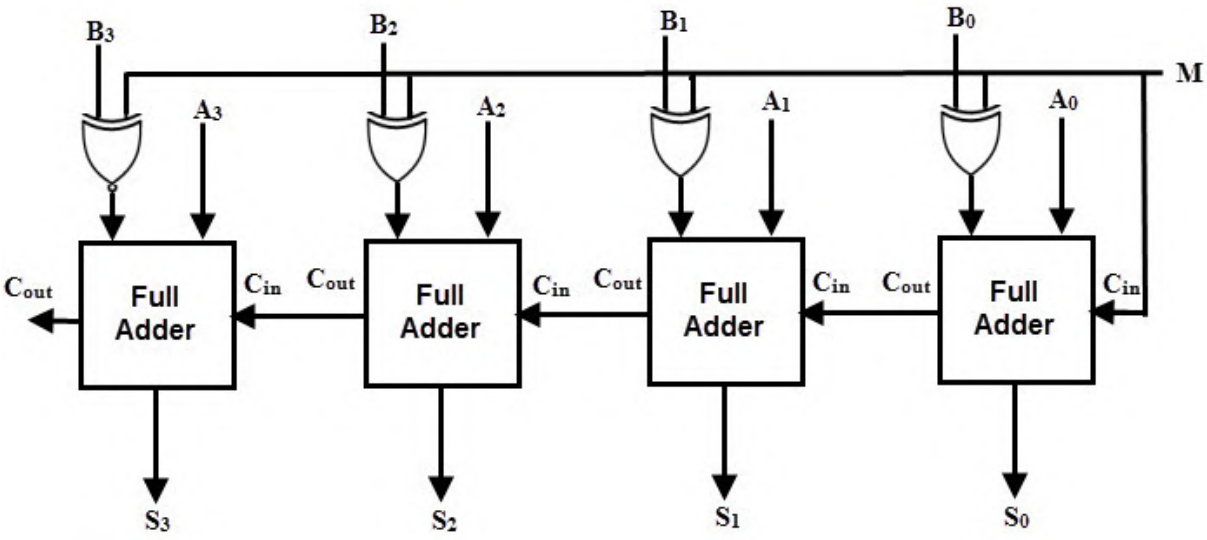

Figure 6.5 4-Bit adder Subtractor

6.2.2 Comparator

Digital or Binary Comparators are made up from standard AND, NOR and NOT gates that compare the digital signals present at their input terminals and produce an output depending upon the condition of those inputs.

For example, along with being able to add and subtract binary numbers we need to be able to compare them and determine whether the value of input A is greater than, smaller than or equal to the value at input B etc. The digital comparator accomplishes this using several logic gates that operate on the principles of Boolean algebra. There are two main types of Digital Comparator available and these are.

- **Identity Comparator**
  An Identity Comparator is a digital comparator with only one output terminal for when A = B, either A = B = 1(HIGH) or A = B = 0(LOW).
- **Magnitude Comparator**
  A Magnitude Comparator is a digital comparator which has three output terminals, one each for equality, A = B greater than A > B, and less than A < B.

6.2.2.1 Magnitude Comparator

The purpose of a Digital Comparator is to compare a set of variables or unknown numbers, for example A (A1, A2, A3, …. An, etc.) against that of a constant or unknown value such as B (B1, B2, B3, …. Bn, etc.) and produce an output condition or flag depending upon the result of the comparison. For example, a magnitude comparator of two 1-bits, (A and B) inputs would produce the following three output conditions when compared to each other.

$$A > B,\ A = B,\ A < B$$

Which means: A is greater than B, A is equal to B, or A is less than B

6.2.2.1.1 1-Bit Comparator

This is useful if we want to compare two variables and want to produce an output when any of the above three conditions are achieved. For example, produce an output from a counter when a certain count number is reached. Consider the simple 1-bit comparator below.

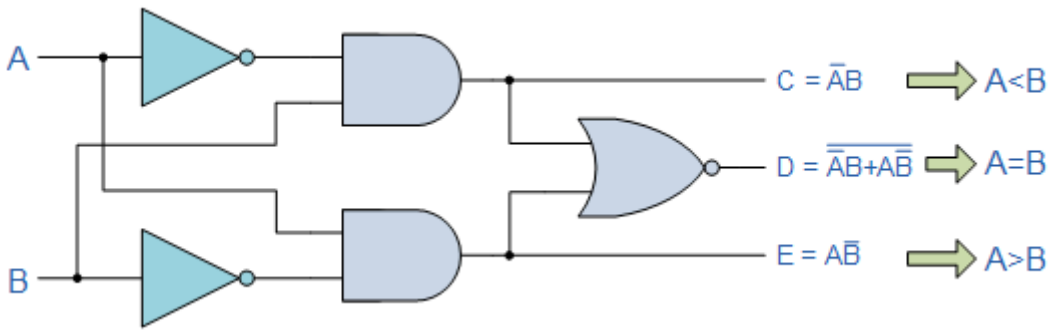

Figure 6.5 1-bit Digital Comparator circuit

Then the operation of a 1-bit digital comparator is given in the following Truth Table.

Table 6.3 1-bit Digital comparator Truth Table

Inputs		Outputs		
B	A	A > B	A = B	A < B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

You may notice two distinct features about the comparator from the above truth table. Firstly, the circuit does not distinguish between either two "0" or two "1"'s as an output A = B is produced when they are both equal, either A = B = "0" or A = B = "1". Secondly, the output condition for A = B resembles that of a commonly available logic gate, the Exclusive-NOR or Ex-NOR function (equivalence) on each of the n-bits giving: Q = A ⊕ B. Digital comparators actually use Exclusive-NOR gates within their design for comparing their respective pairs of bits. When we are comparing two binary or BCD values or variables against each other, we are comparing the "magnitude" of these values, a logic "0" against a logic "1" which is where the term Magnitude Comparator comes from. As well as comparing individual bits, we can design larger bit comparators by cascading together n of these and produce an n-bit comparator just as we did for the n-bit adder in the previous tutorial. Multi-bit comparators can be constructed to compare whole binary or BCD words to produce an output if one word is larger, equal to or less than the other.

6.2.2.1.2 2-Bit Comparator

A comparator used to compare two binary numbers each of two bits is called a 2-bit Magnitude comparator. It consists of four inputs and three outputs to generate less than, equal to and greater than between two binary numbers.

The truth table for a 2-bit comparator is given below:

Table 6.4 2-bit Digital comparator Truth Table

INPUT				OUTPUT		
A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

From the above truth table K-map for each output can be drawn as follows:

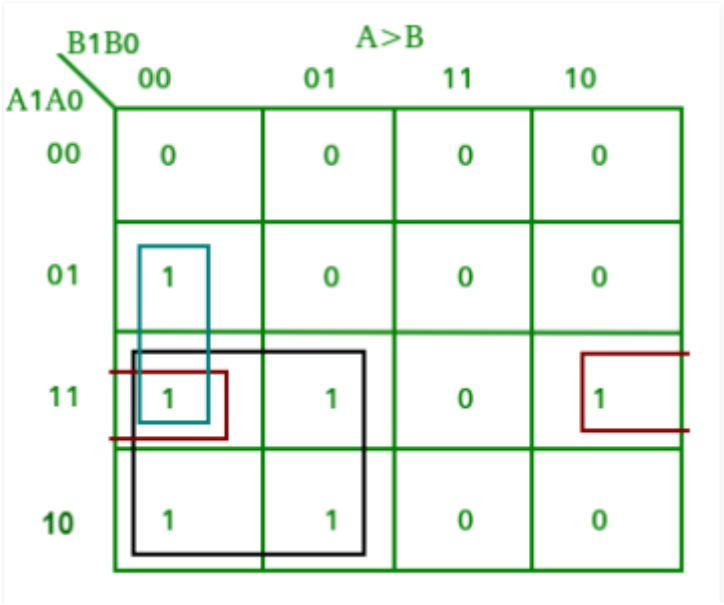

Figure 6.6 A > B K-Map with Grouping

Figure 6.7 A = B K-Map with Grouping

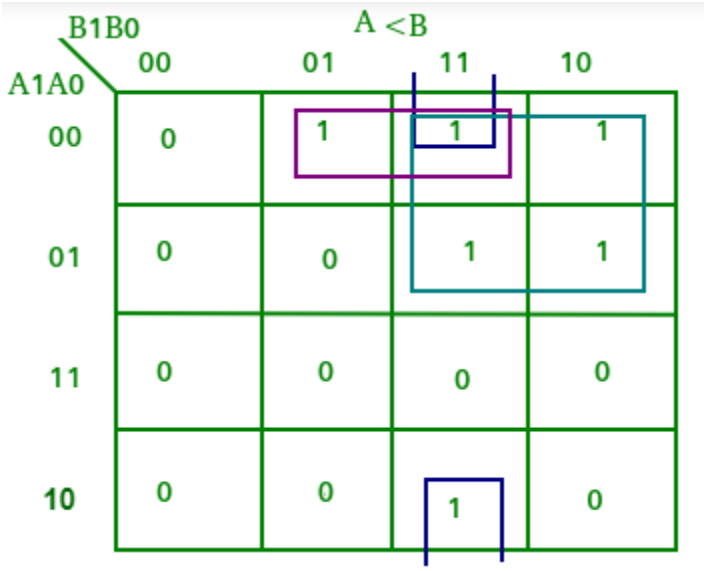

Figure 6.8 A < B K-Map with Grouping

From the above K-maps logical expressions for each output can be expressed as follows:

```
A>B:A1B1' + A0B1'B0' + A1A0B0'
A=B: A1'A0'B1'B0' + A1'A0B1'B0 + A1A0B1B0 + A1A0'B1B0'
   : A1'B1' (A0'B0' + A0B0) + A1B1 (A0B0 + A0'B0')
   : (A0B0 + A0'B0') (A1B1 + A1'B1')
   : (A0 Ex-Nor B0) (A1 Ex-Nor B1)
A<B:A1'B1 + A0'B1B0 + A1'A0'B0
```

By using these Boolean expressions, we can implement a logic circuit for this comparator as given below:

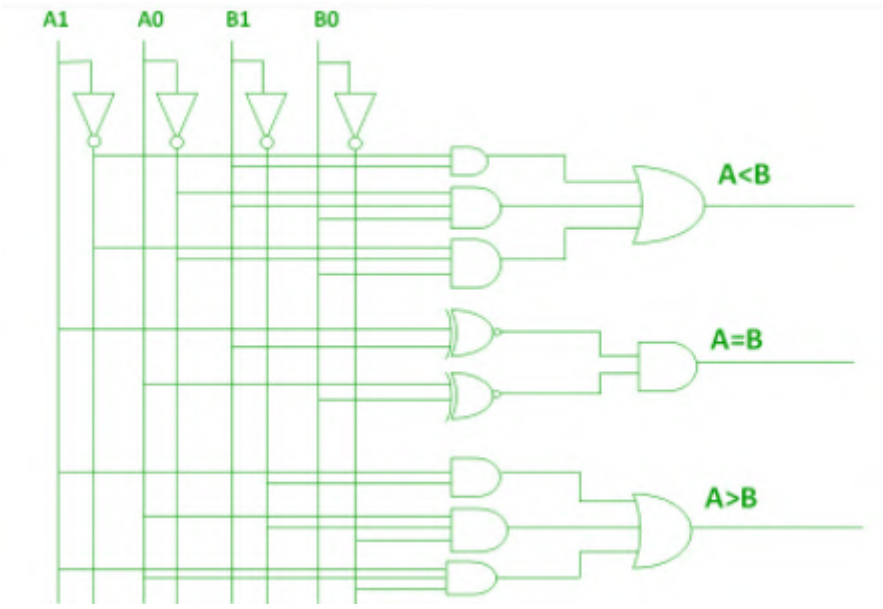

Figure 6.9 2-Bit Comparator logic circuit

6.2.2.1.3 4-Bit Comparator

Some commercially available digital comparators such as the TTL 74LS85 or CMOS 4063 4-bit magnitude comparator have additional input terminals that allow more individual comparators to be "cascaded" together to compare words larger than 4-bits with magnitude comparators of "n"-bits being produced. These cascading inputs are connected directly to the corresponding outputs of the previous comparator as shown to compare 8, 16 or even 32-bit words.

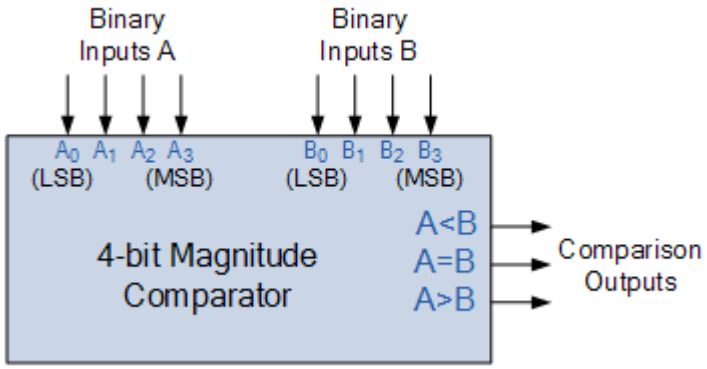

Figure 6.10 4-bit magnitude comparator

6.3 Lab Activities

6.3.1 Task-1:

Design a 4-bit Adder-Subtractor and implement it by cascading Full Adders. Also perform Truth Table based verification.

a) Four- bit Adder
b) Modify the four-bit Adder into a 4-bit Adder Subtractor

6.3.2 Task-2:

Design a 2-bit comparator and implement it. Also Perform Truth Table based verification.

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 Implementation of Adder and Magnitude Comparator	Makes few critical errors in applying procedural knowledge of Implementation of Adder and Magnitude Comparator	Makes some non-critical errors in applying procedural knowledge of Implementation of Adder and Magnitude Comparator	Applies the procedural knowledge of Implementation of Adder and Magnitude Comparator 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: _____**