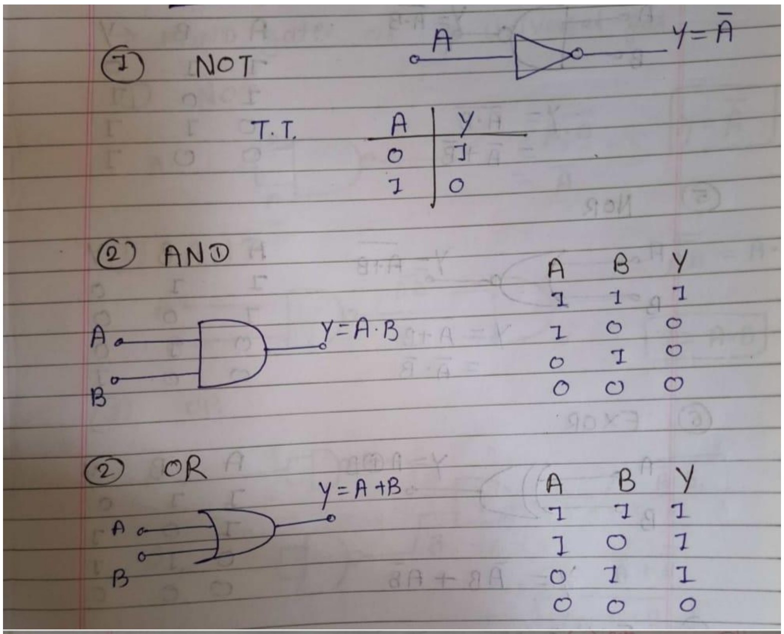
DAY 2 - 111 DAYS VERIFICATION CHALLENGE

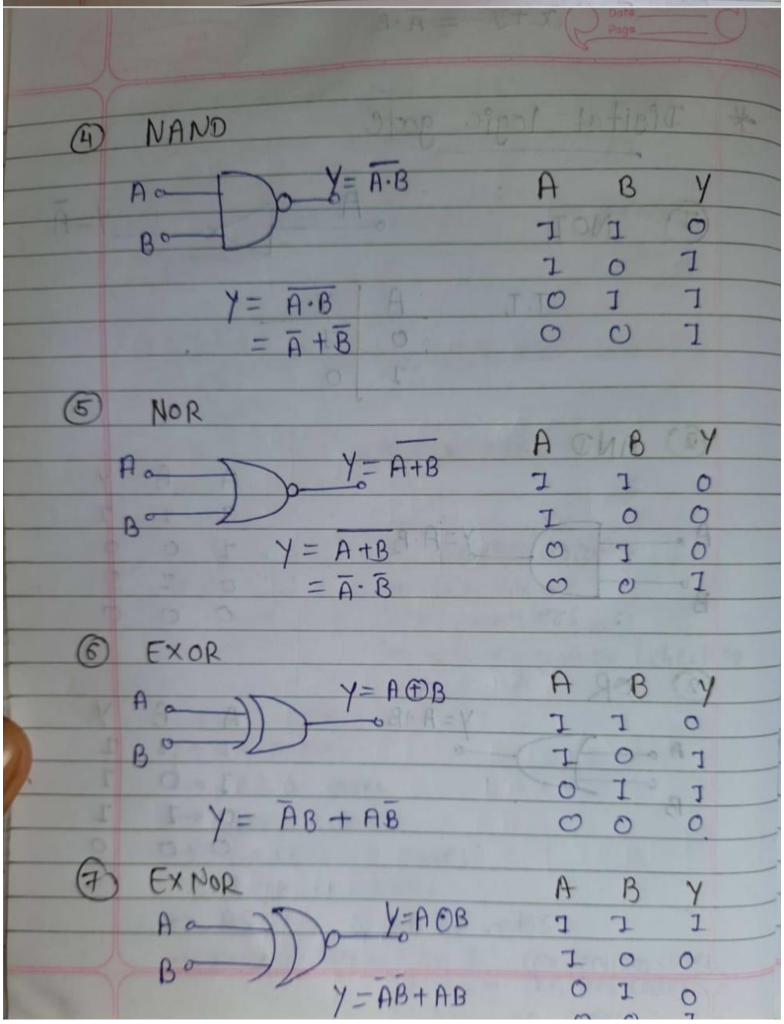
Topic: Logic Gates

Skill: Digital Electronics

DAY 2 CHALLENGE:

1. Which are basic logic gates. Explain working with truth table.



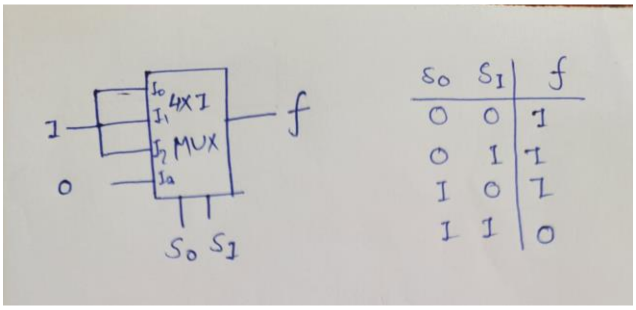


2. Why are NAND & NOR called the Universal Gates? What are advantages of universal gates.

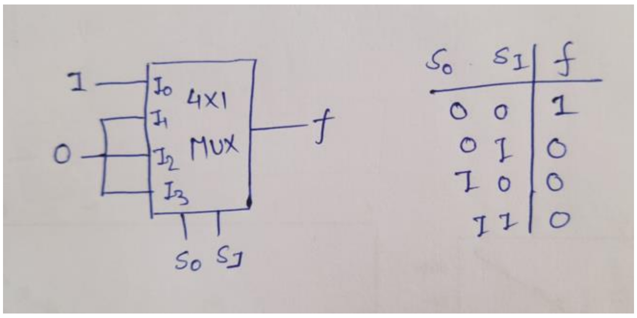
NAND and NOR gates are called Universal Gates because they can be used to implement any Boolean function, meaning they can create any other logic gate (AND, OR, NOT, etc.) through combinations of themselves. The main advantage of universal gates is that they reduce the complexity and cost of designing and manufacturing digital circuits, as designers can use just one type of gate to create any required logic function.

3. Design 4:1 MUX using universal gates:

a. NAND



b. NOR



4. What are applications of these gates:

a. NAND

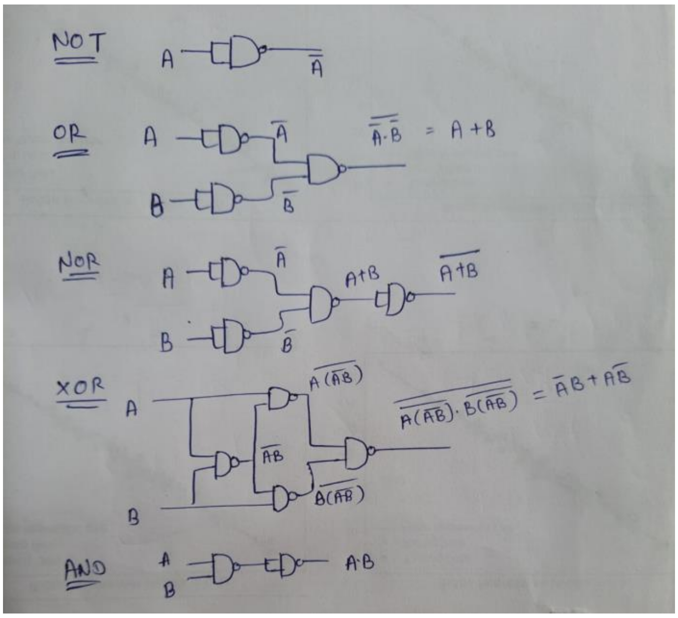
NAND gates are versatile and can be used to construct all other basic logic gates like AND, OR, and NOT. They are widely used in digital electronics for building complex circuits, such as in memory elements like SR flip-flops, microprocessors, and digital signal processing circuits. Their ability to perform all fundamental logic operations makes them essential in various applications.

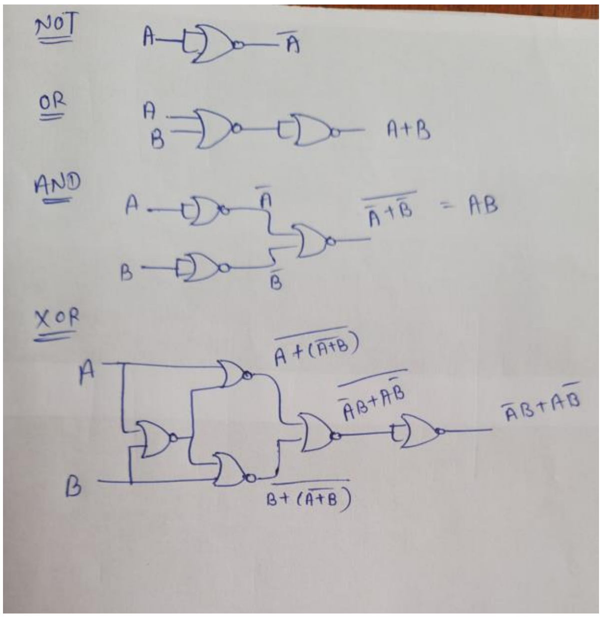
b. NOR

NOR gates are similarly versatile and can also be used to create all basic logic gates. They are crucial in memory and control circuits, forming the basis of SR flip-flops and latches. Additionally, NOR gates are used in microprocessors and digital systems, contributing to components like ALUs. They are also essential in generating clock pulses and timing signals in oscillators and timer circuits.

5.design OR, AND, XOR, NOT using

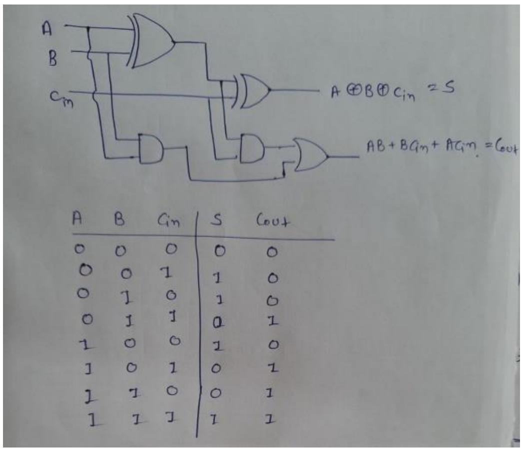
a. NAND **b. NOR**





6. Design below logic using Logic Gates

b. Full adder



A	B	Cin	S	Cout
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

7. What happens if more than one input is applied to a logic gate?

When more than two inputs are applied to a logic gate, the gate processes all inputs according to its logical function. Here's what happens for each type of logic gate with more than two inputs:

1. **AND Gate:** The output is high (1) only if all inputs are high. If any one of the inputs is low (0), the output will be low.
   - Example: For inputs A=1, B=1, C=0, and D=1, the output is 0.

2. **OR Gate:** The output is high if at least one input is high. The output is low only if all inputs are low.
    - Example: For inputs A=0, B=1, C=0, and D=0, the output is 1.
3. **NAND Gate:** The output is the inverse of the AND gate. The output is low only if all inputs are high; otherwise, it is high.
    - Example: For inputs A=1, B=1, C=0, and D=1, the output is 1.
4. **NOR Gate:** The output is the inverse of the OR gate. The output is high only if all inputs are low; otherwise, it is low.
    - Example: For inputs A=0, B=0, C=0, and D=1, the output is 0.
5. **XOR Gate:** The output is high if an odd number of inputs are high. For more than two inputs, it performs an extended XOR operation.
    - Example: For inputs A=1, B=0, C=1, and D=0, the output is 0.
6. **XNOR Gate:** The output is high if an even number of inputs are high. It performs an extended XNOR operation for more than two inputs.
    - Example: For inputs A=1, B=0, C=1, and D=0, the output is 1.