Name : Ritesh Jha
Roll No : 16010423076
Batch : SY-IT(B3)
Subject : Digital Systems IA 1 : Mini Project
Virtual Lab : https://da-iitb.vlabs.ac.in/exp/eight-bit-ones-complement/simulation.html

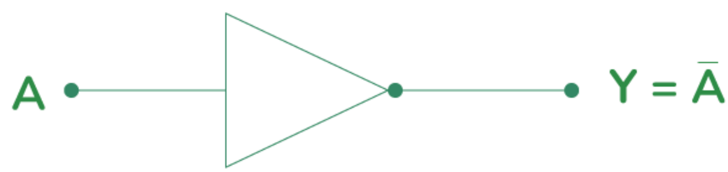
Basics of NOT gate and its application in an 8-bit one's complement circuit

Aim:
To apply a basic NOT gate logic in a 8 -bit one's complement circuit. The user will be able to build, simulate and verify the 8-bit one's complementing circuit using the generalized simulator (a blank canvas with click & place facility for selected gates).

Theory :
The NOT gate, also called an inverter, performs an operation known as inversion or complementation. In Boolean algebra, the opposite of a number is called its complement. For example, the complement of 1 is 0 and vice versa. This operation is represented by a bar over a variable. The NOT gate changes one logic level to another—turning a 1 into a 0 and a 0 into a 1. Essentially, if the input is LOW (0), the output will be HIGH (1) and if the input is HIGH (1), the output will be LOW (0).

NOT Gate Function
A NOT gate has one input and one output. It produces a HIGH output if the input is LOW and a LOW output if the input is HIGH. The gate is used to invert the input signal, making it a key component in various logic circuits.

NOT Gate



Truth Table

A (Input)	$Y = \bar{A}$ (Output)
0	1
1	0

Application: 1's Complement of an 8-Bit Number

To obtain the 1’s complement of an 8-bit number, we use eight NOT gates. Since each IC 74LS04 (TTL Hex Inverter) or IC 74HCT04 (CMOS Hex Inverter) contains six NOT gates, two such ICs are required for this operation. The 2's complement of an 8-bit number can be derived by adding 1 to its 1's complement. This process is often used in binary subtraction operations involving 8-bit numbers.

Concept :

A NOT gate flips the input signal, producing a LOW output for a HIGH input and a HIGH output for a LOW input. This inversion property makes the NOT gate useful in generating the 1’s complement of a number. When a bubble appears on the output of a logic gate, it means the active output state is 0, also known as an active-low output. This concept is applied in designing circuits like the 8-bit 1's complement circuit.

Pretest :

Virtual Labs
An MoE Govt of India Initiative

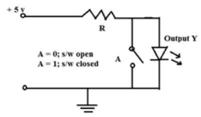
HOME PARTNERS CONTACT

Electronics and Communication Engineering > Digital Applications > Experiments

- Aim
- Theory
- Pretest
- Procedure
- Simulation
- Posttest
- References
- Feedback

Basics of NOT gate and its application in an 8-bit one's complement circuit

The output of the circuit shown below represents which of the following operations?



- a : ANDing
- b : ORing
- c : Inverting
- d : Buffering

A byte 10101100 is applied as input to a 8-bit complementing logic circuit. The output will be:

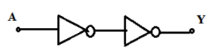
- a : 00001111
- b : 10100011
- c : 11111100
- d : 01010011

The output of the combinational logic shown in figure below is given by the Boolean expression:

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

The output of the combinational logic shown in figure below is given by the Boolean expression:



- a : Y = 1
- b : Y = 0
- c : Y = A
- d : Y = A'

The output of the expression AB.A'= ______.

- a : A
- b : A'
- c : 0
- d : 1

The output of the expression A'(A + A') = _______.

- a : A
- b : A'
- c : 1
- d : 0

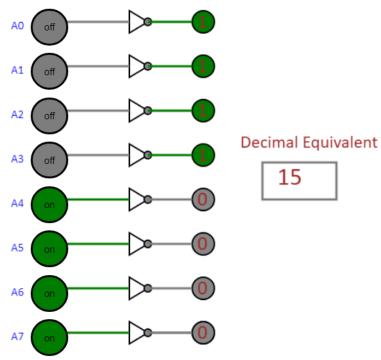
Submit Quiz

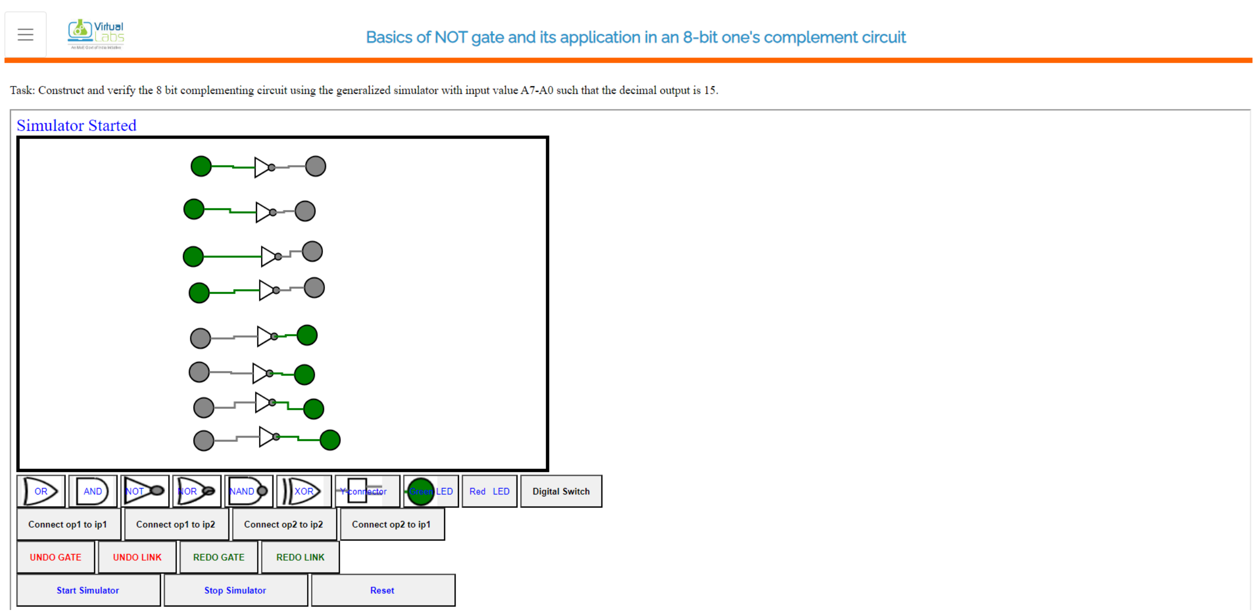
5 out of 5

Procedure :

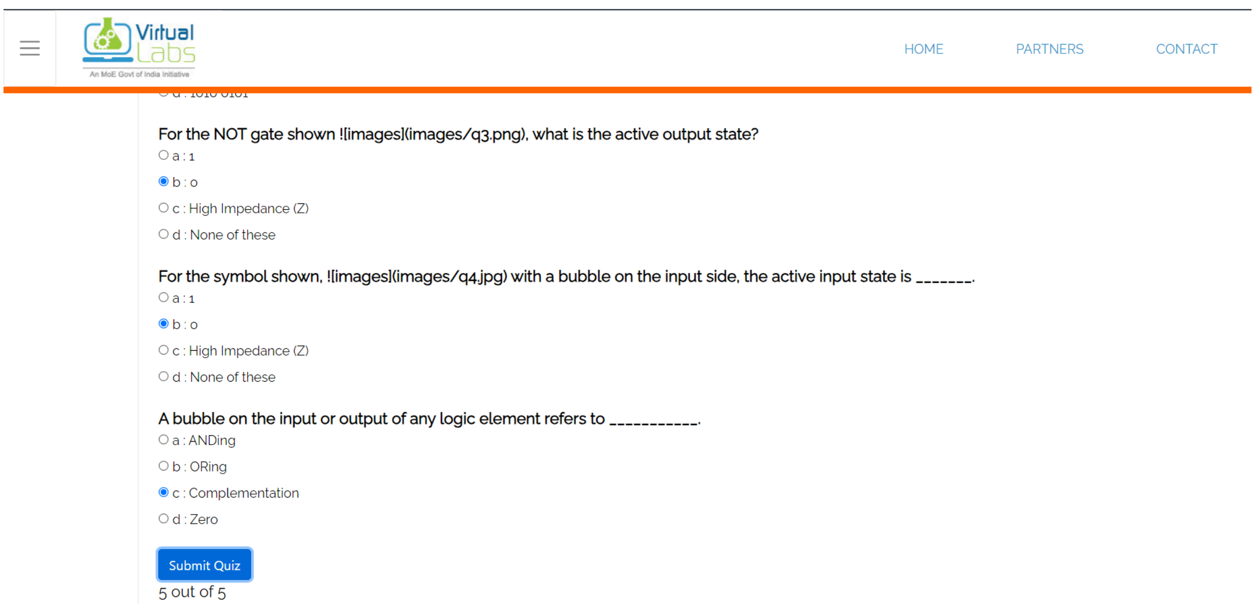
1. Set the 8-bit input to some value, say 1010 0101.
2. Observe the output on the LEDs.
3. Also observe the decimal equivalent of the output on the right hand side box.
4. Using a simulator, construct the 8-bit one's complement circuit.
5. Verify the results.

Simulation :





Post test :



Conclusion :

Through this simulation, I learned how to construct an 8-bit one's complement circuit using NOT gates. By inputting a binary value and inverting each bit, I was able to successfully generate the one's complement of the input. Specifically, I saw how an input of 1111 0000 was complemented to 0000 1111, which corresponds to 15 in decimal. This simulation helped me understand the practical application of NOT gates in digital circuits, and how they can be used to perform bitwise negation in a straightforward and visual way. Additionally, I gained hands-on experience with circuit design, component connections, and output verification using LEDs, enhancing my understanding of basic digital logic.