ITS 62704
Computer Organization & Architecture
Rajesh Maharjan

Auto Desk Tinker CAD – Digital Logic Circuits

Practical Test Bachelor of Computer Science (Hons)

Date (Day): 31st March, 2023 (Friday) Time: 12:30p – 2:30p

January 2023; Section D Group: D5 (Room G004)

Student ID	University ID	Full Name
22013224	0362230	Bikrant Shilpakar
22013225	0362261	Pragyan Tamakhu
22013237	0362286	Rijan Khyaju
22013230	0362483	Sujal Ratna Tuladhar
22013243	0362485	Sujen Dangol

ITS 62704
Computer Organization & Architecture
Rajesh Maharjan

Table of Contents

Part 1: Basic Logic Gates

Part A: Truth Tables

Q. a. Output is 0 when inputs at same level

XOR: exclusive-OR act as either/or; output is “true” if either, but not both, the input is “true” and output is “false” if both inputs are “false” or if both inputs are “true”

INPUTS		OUTPUT
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

$$X = A\ XOR\ B = A \oplus B = \bar{A}B + A\bar{B}$$

Q. b. The output is not 1 until both inputs are 1.

AND: output is “true” when both inputs are “true” otherwise “false” aka Conjunction

INPUTS		OUTPUT
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

$$X = A\ AND\ B = A.B = AB$$

Q. c. The Output is 0 when both inputs are 1, otherwise output is 1.
NAND: AND gate followed by NOT gate; output is “false” if both inputs are “true” else “true”

INPUTS		OUTPUT
A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

$$X = A\ NOT\ AND\ B = A\ NAND\ B = \overline{AB}$$

Q. d. Output is 0 for same inputs.
XOR: exclusive-OR act as either/or; output is “true” if either, but not both, the input is “true” and output is “false” if both inputs are “false” or if both inputs are “true”

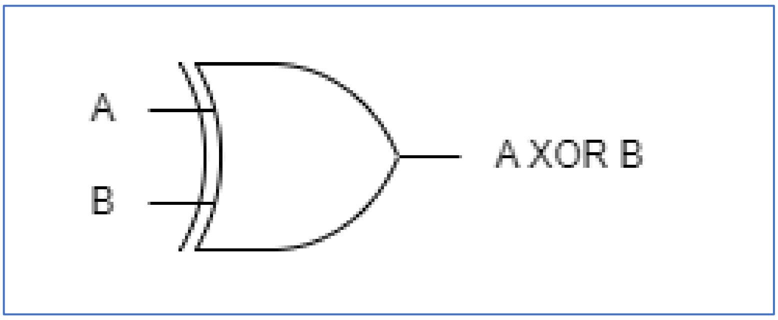
INPUTS		OUTPUT
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

$$X = A\ XOR\ B = A \oplus B = \bar{A}B + A\bar{B}$$

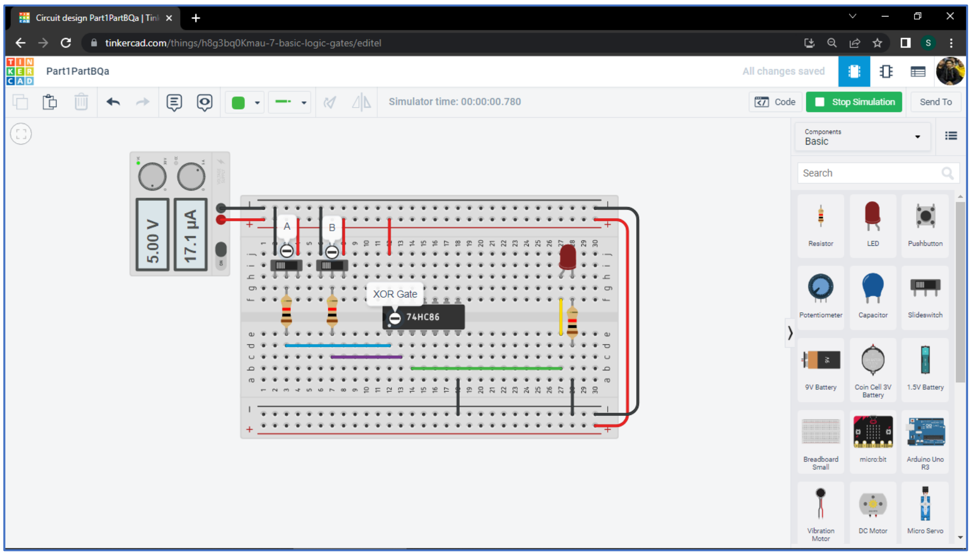
ITS 62704
Computer Organization & Architecture
Rajesh Maharjan

Part B: Demonstrate in Tinker CAD

Q. a. The output is 0 when inputs are at same level.

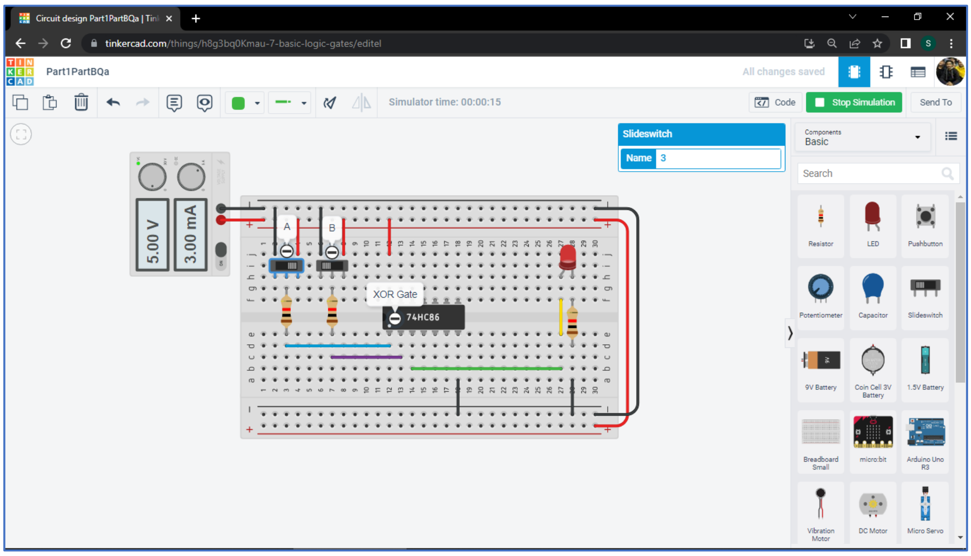


00



Simulator time: 00:00:00.780

10

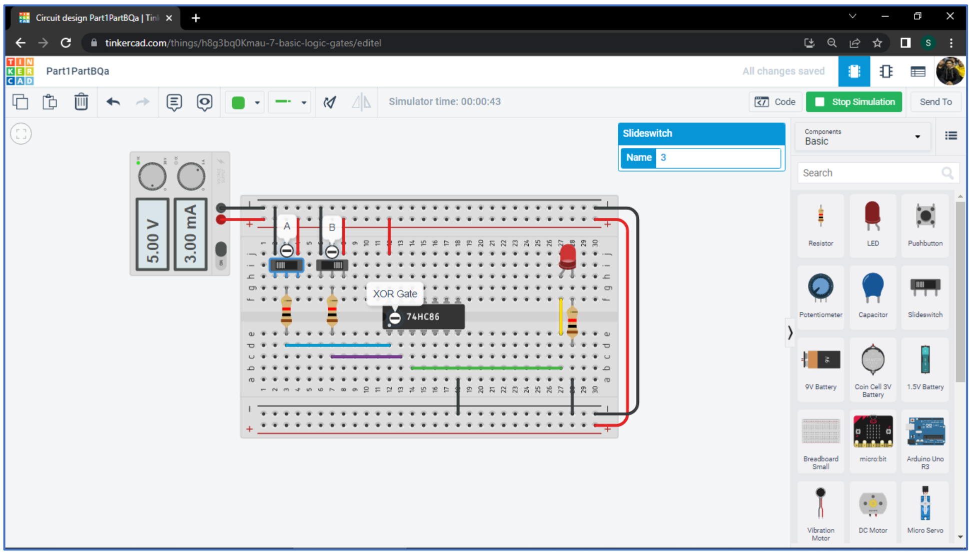


Simulator time: 00:00:15

Name 3

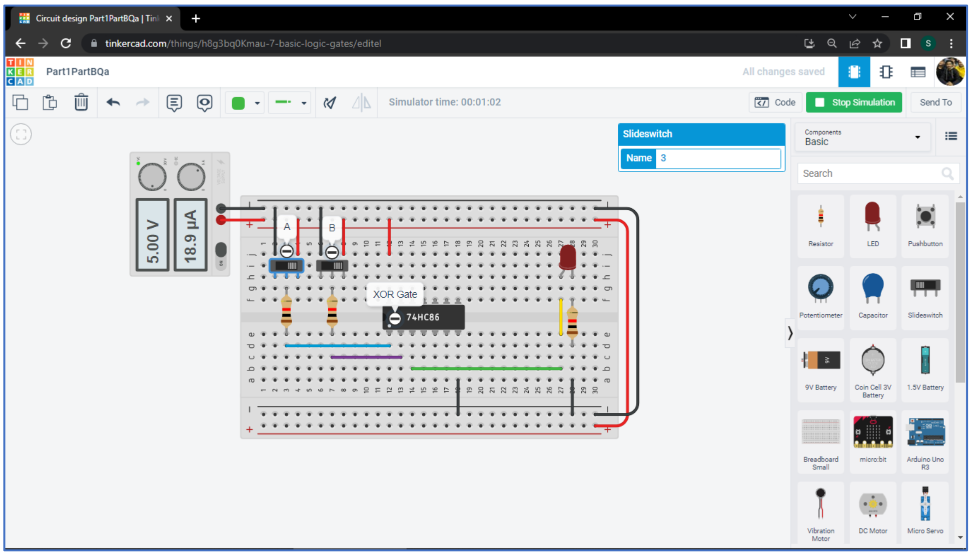
ITS 62704
Computer Organization & Architecture
Rajesh Maharjan

01



Slideswitch
Name 3

11



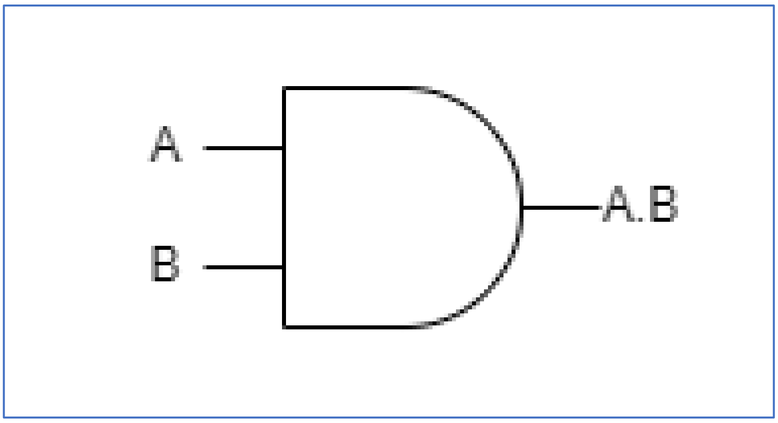
Slideswitch
Name 3

ITS 62704

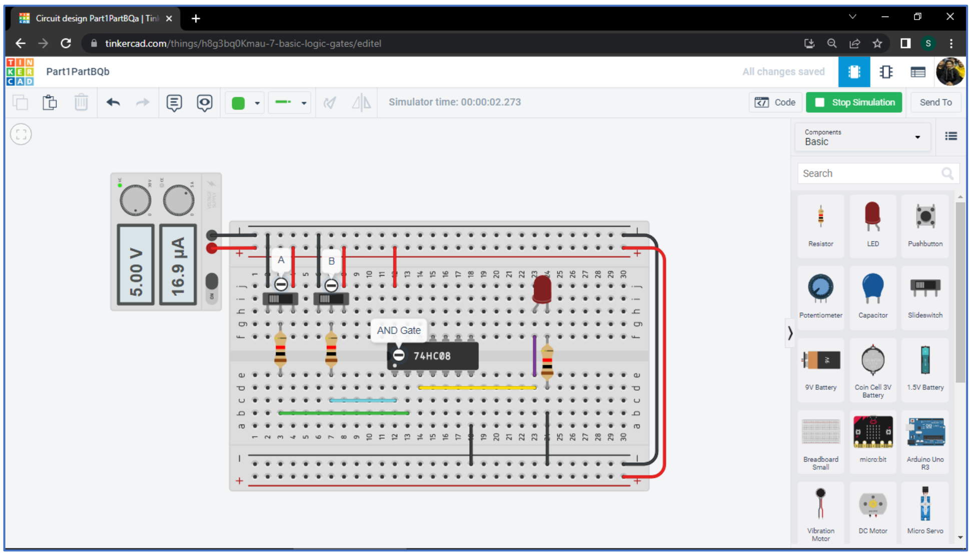
Computer Organization & Architecture

Rajesh Maharjan

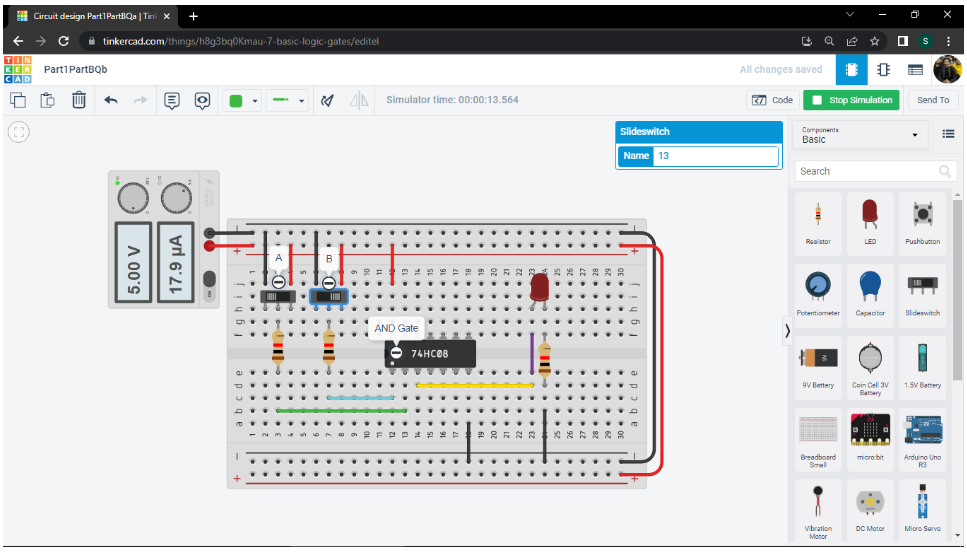
Q. b. The output is not 1 until both inputs are 1.



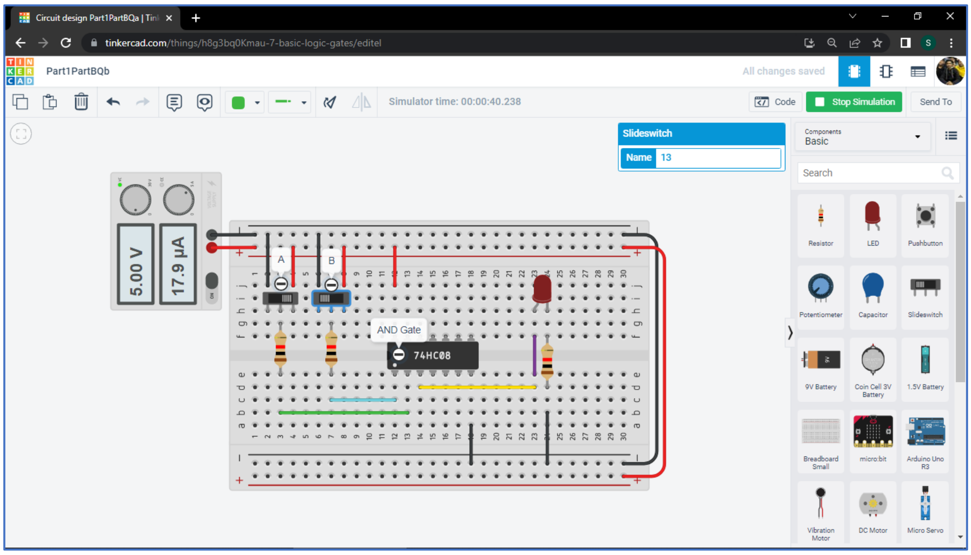
00



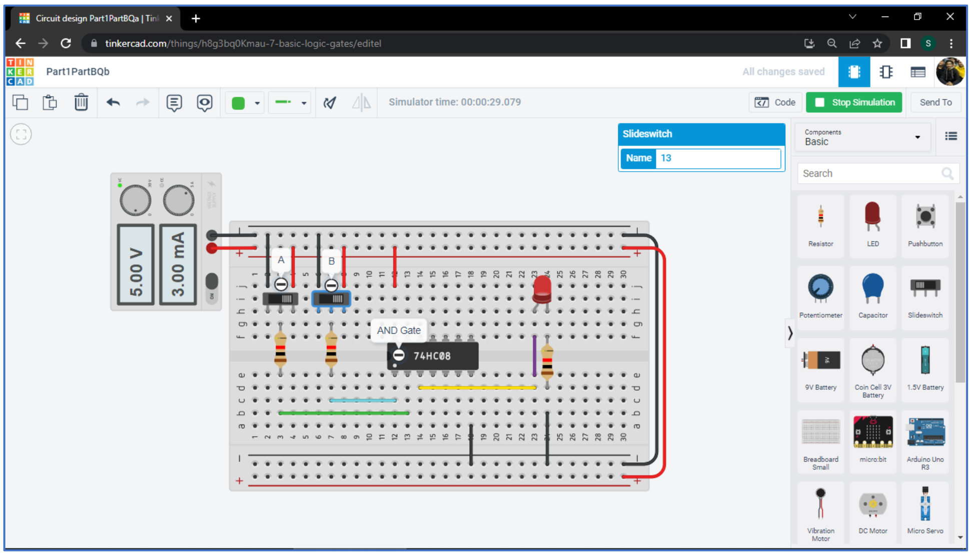
01



10

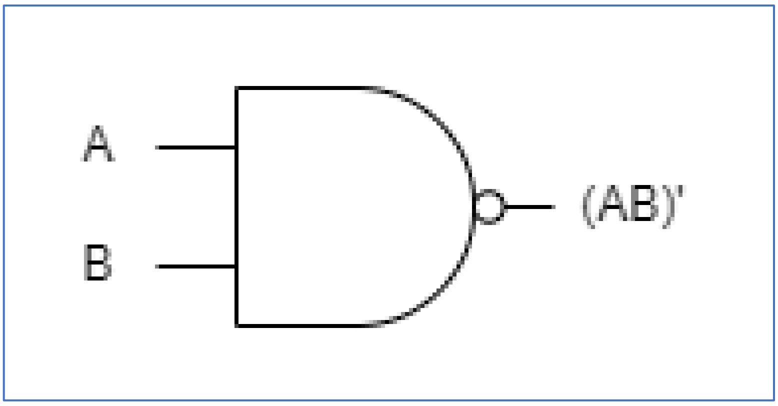

Circuit design Part1PartBQa | Tin
tinkercad.com/things/h8g3bq0Kmau-7-basic-logic-gates/editel
Part1PartBQb
All changes saved
Simulator time: 00:00:40.238
Code
Stop Simulation
Send To
Slideswitch
Name 13
Components
Basic
Search
5.00 V
17.9 µA
A
B
AND Gate
74HC08
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
9V Battery
Coin Cell 3V Battery
1.5V Battery
Breadboard Small
micro:bit
Arduino Uno R3
Vibration Motor
DC Motor
Micro Servo

11

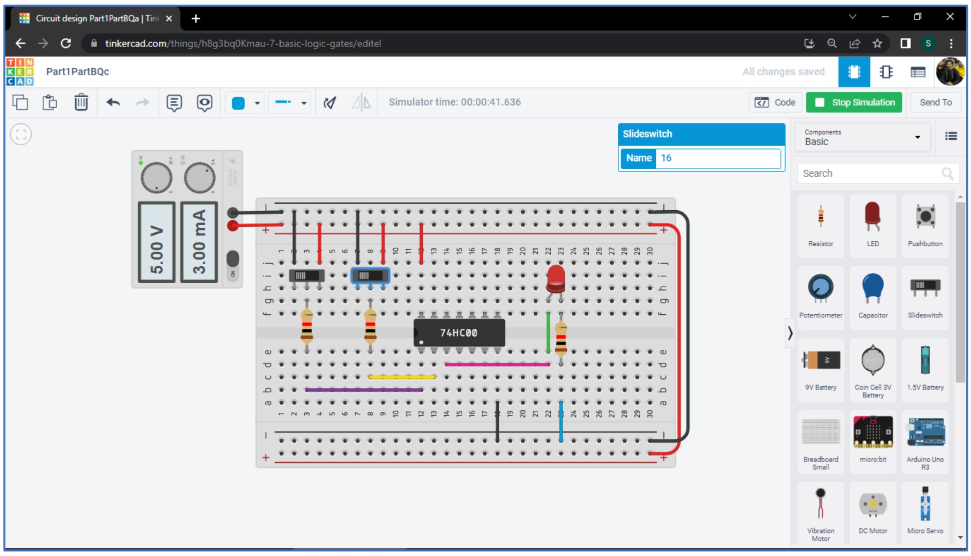

Circuit design Part1PartBQa | Tin
tinkercad.com/things/h8g3bq0Kmau-7-basic-logic-gates/editel
Part1PartBQb
All changes saved
Simulator time: 00:00:29.079
Code
Stop Simulation
Send To
Slideswitch
Name 13
Components
Basic
Search
5.00 V
3.00 mA
A
B
AND Gate
74HC08
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
9V Battery
Coin Cell 3V Battery
1.5V Battery
Breadboard Small
micro:bit
Arduino Uno R3
Vibration Motor
DC Motor
Micro Servo

ITS 62704
Computer Organization & Architecture
Rajesh Maharjan

Q. c. The input is 0 when both inputs are 1, otherwise output is 1.



00

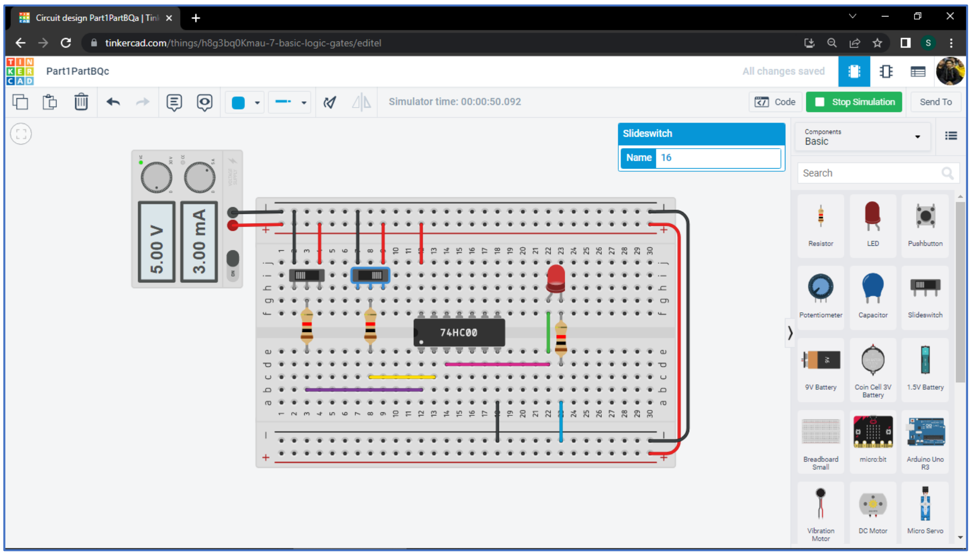


Simulator time: 00:00:41.636

Slideswitch

Name 16

01

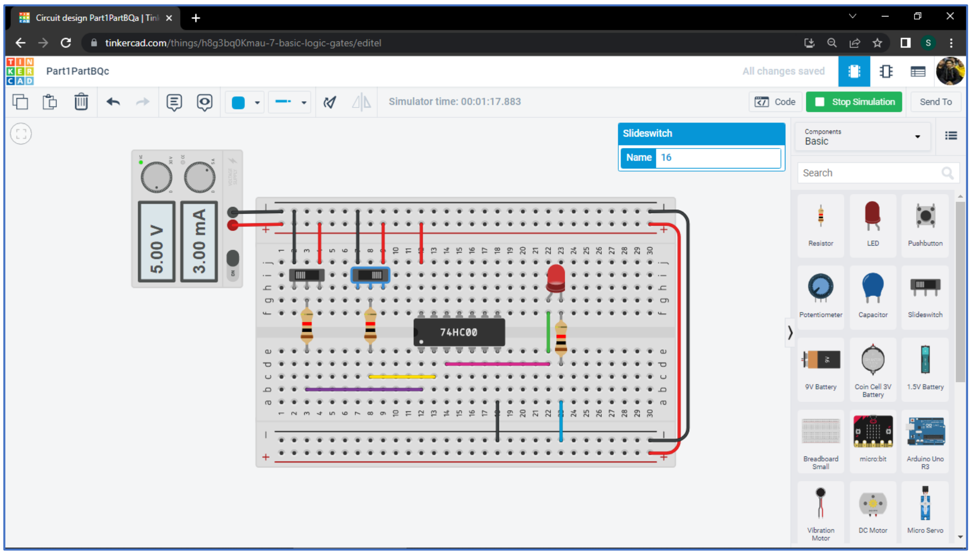


Simulator time: 00:00:50.092

Slideswitch

Name 16

10



Circuit design Part1PartBQa | Tink

tinkercad.com/things/h8g3bq0Kmau-7-basic-logic-gates/editel

Part1PartBQc

All changes saved

Simulator time: 00:01:17.883

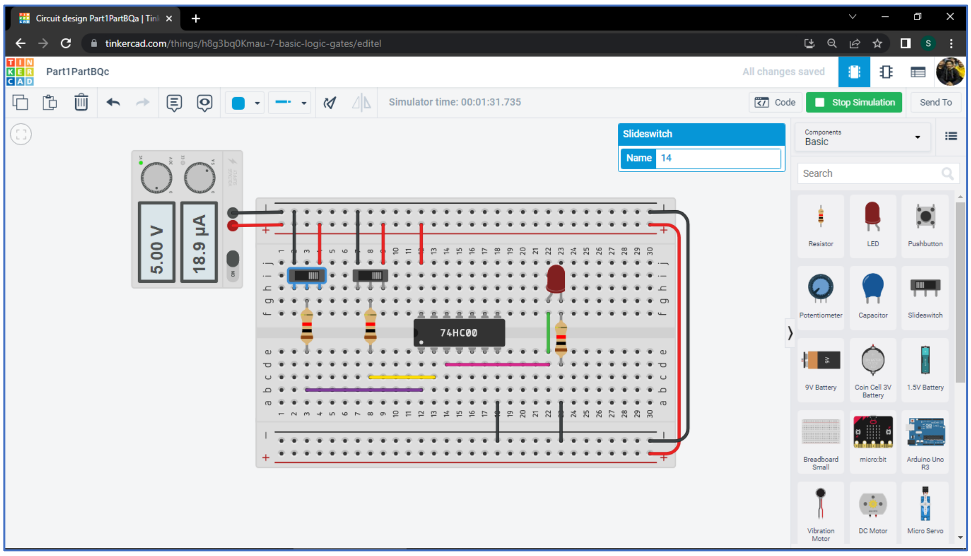
Slideswitch

Name: 16

5.00 V

3.00 mA

11



Circuit design Part1PartBQa | Tink

tinkercad.com/things/h8g3bq0Kmau-7-basic-logic-gates/editel

Part1PartBQc

All changes saved

Simulator time: 00:01:31.735

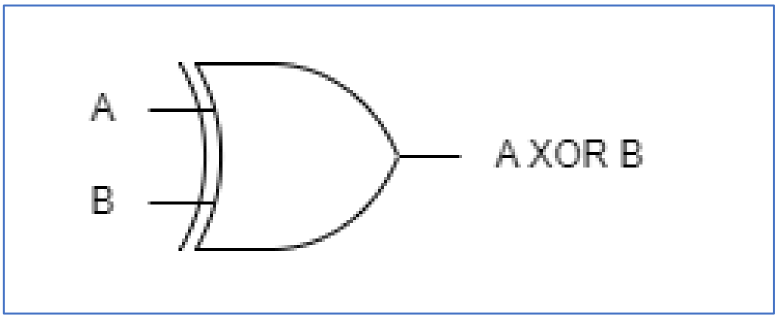
Slideswitch

Name: 14

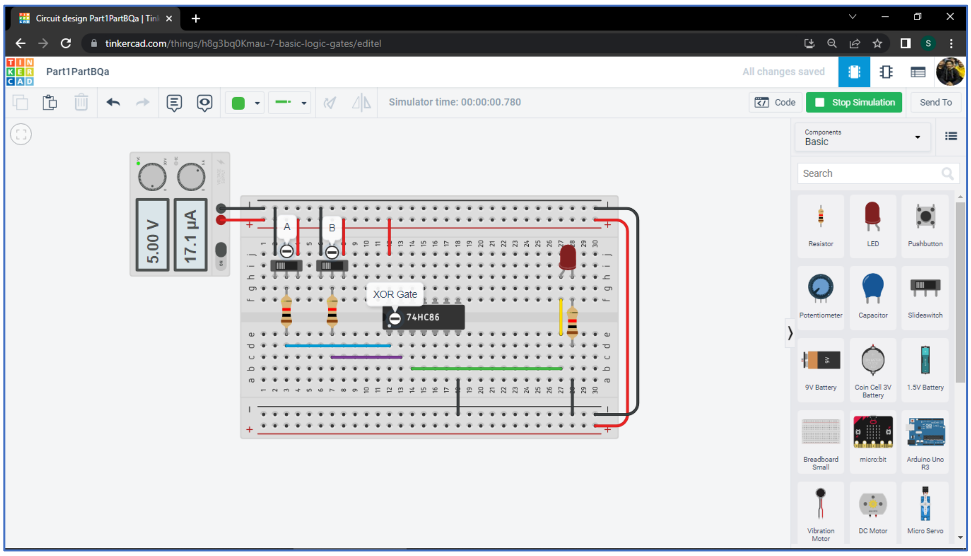
5.00 V

18.9 µA

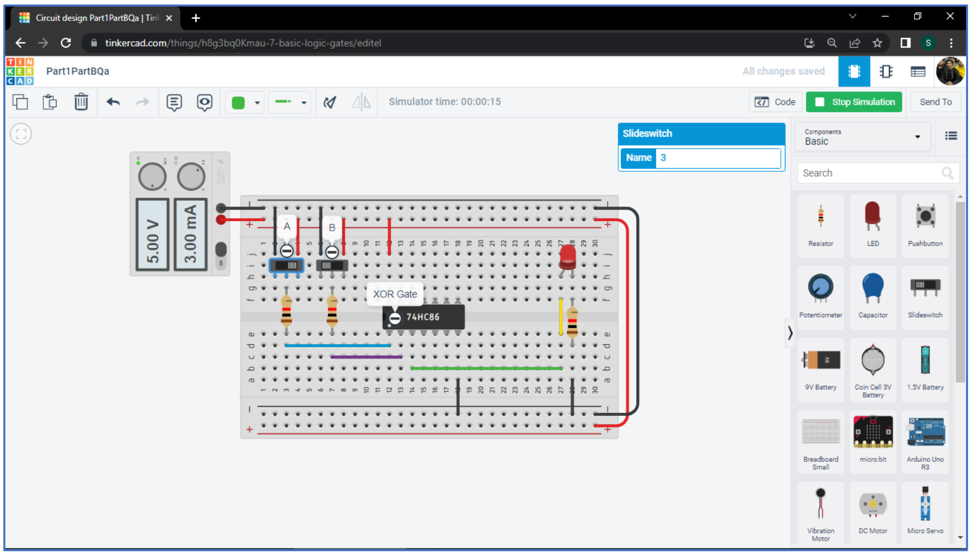
Q. d. Output is 0 for same input.



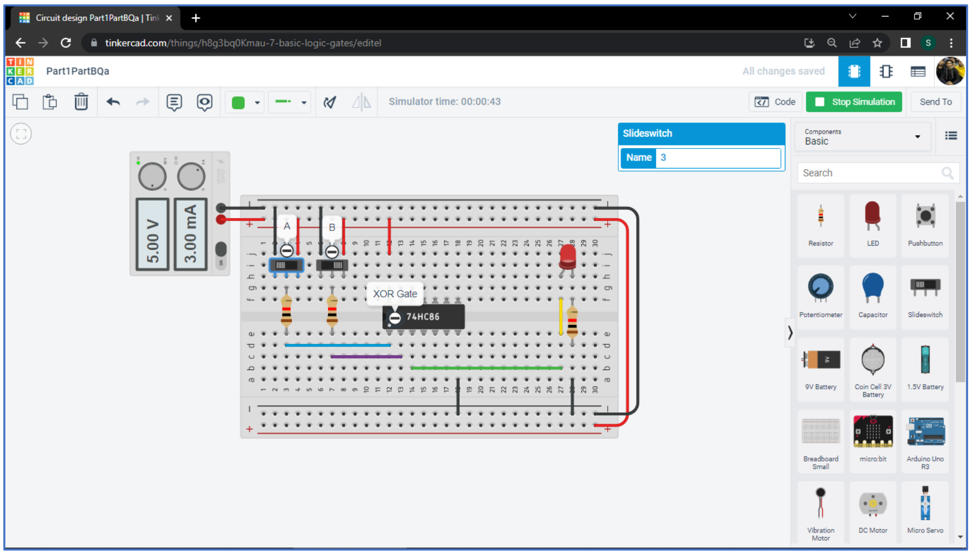
00



10



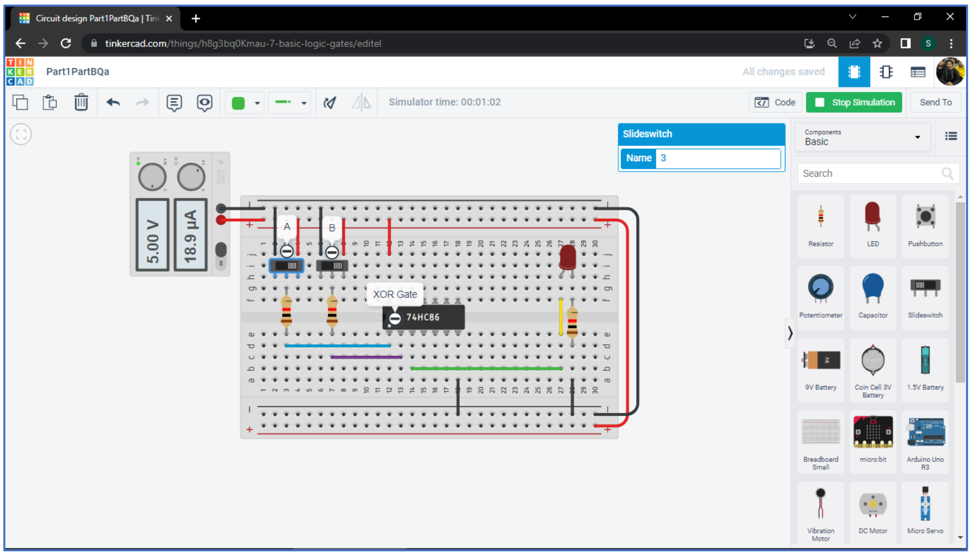
01



Simulator time: 00:00:43

Name 3

11



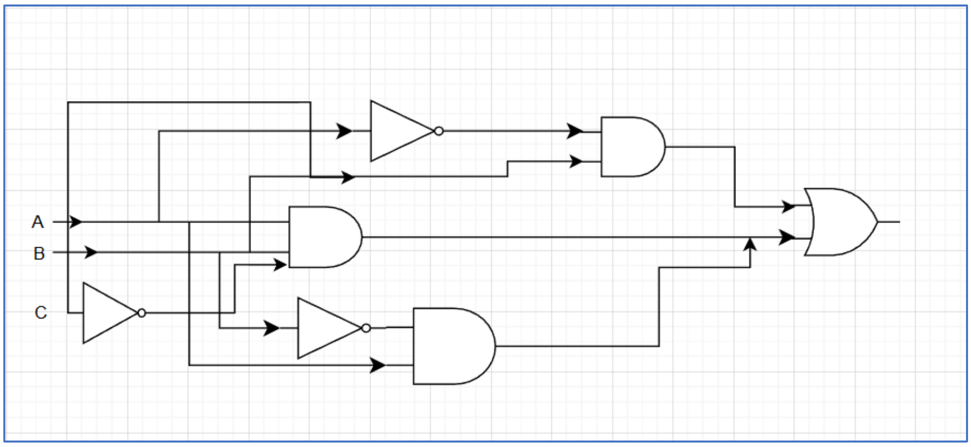
Simulator time: 00:01:02

Name 3

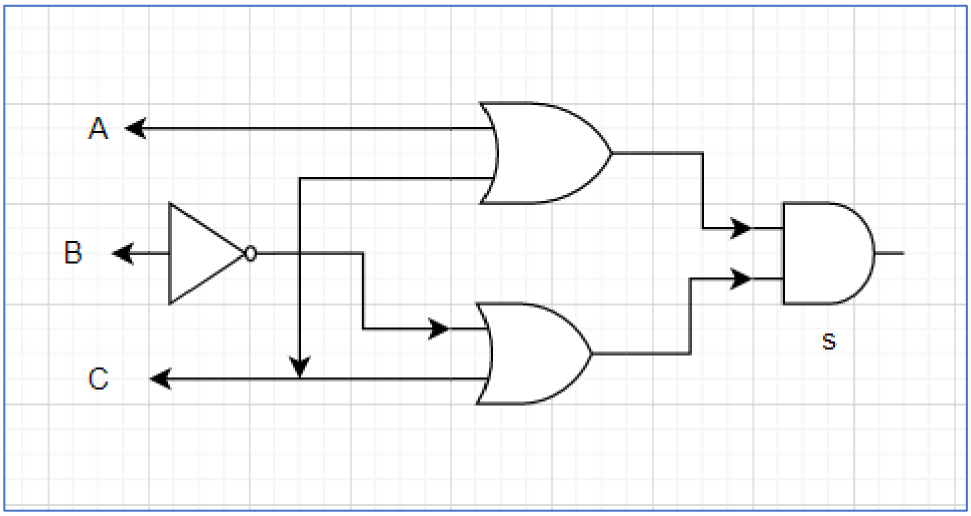
ITS 62704
Computer Organization & Architecture
Rajesh Maharjan

Part C: Circuit Diagram

Q. a. $AB\bar{C} + A\bar{B}C + \bar{A}BC = AB + BC + CA$



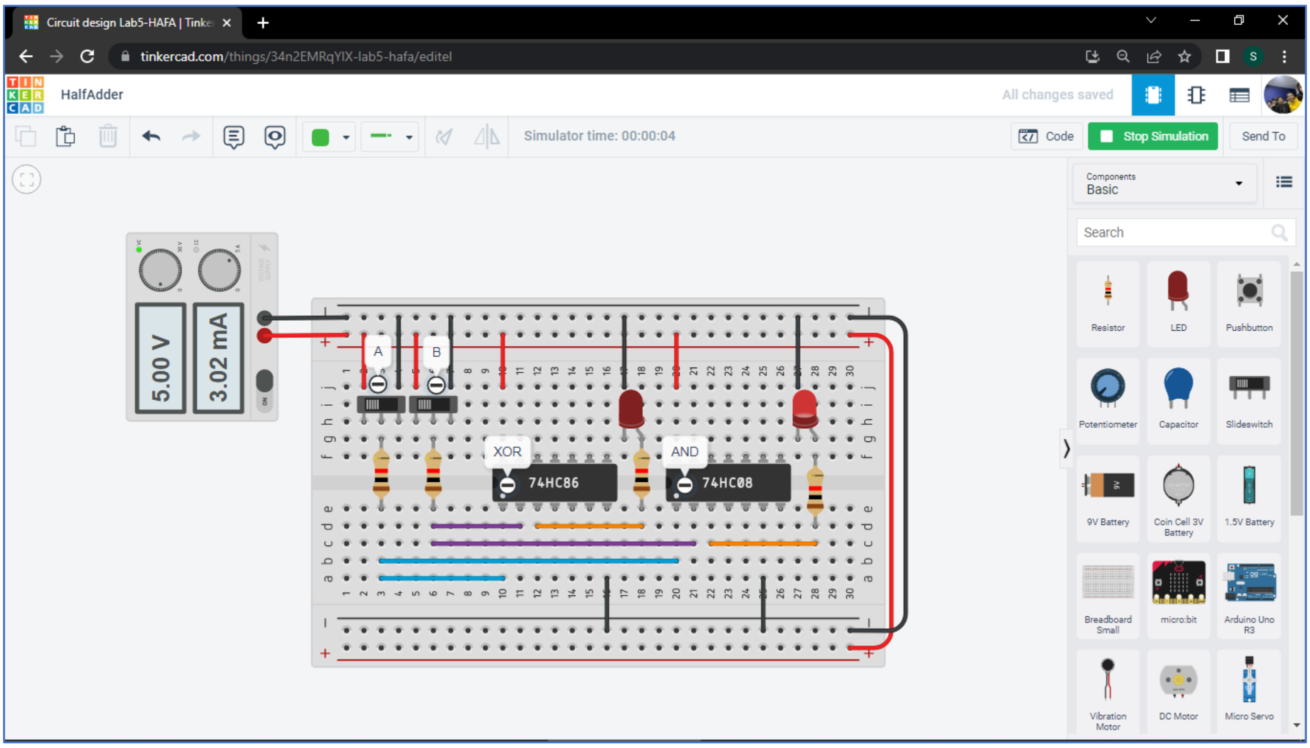
Q. b. $(A + C)(\bar{B} + C)$

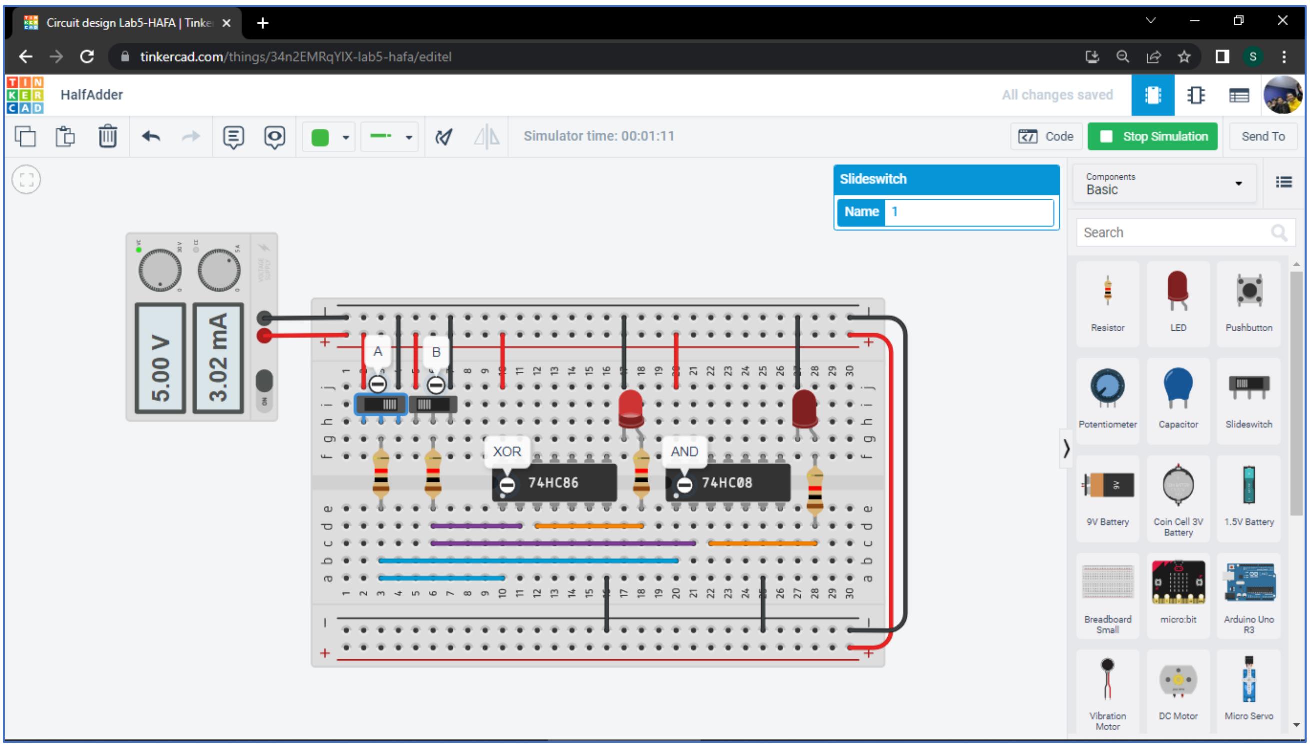


ITS 62704
Computer Organization & Architecture
Rajesh Maharjan

Part 2: Half Adder

Part A: Circuit Diagram



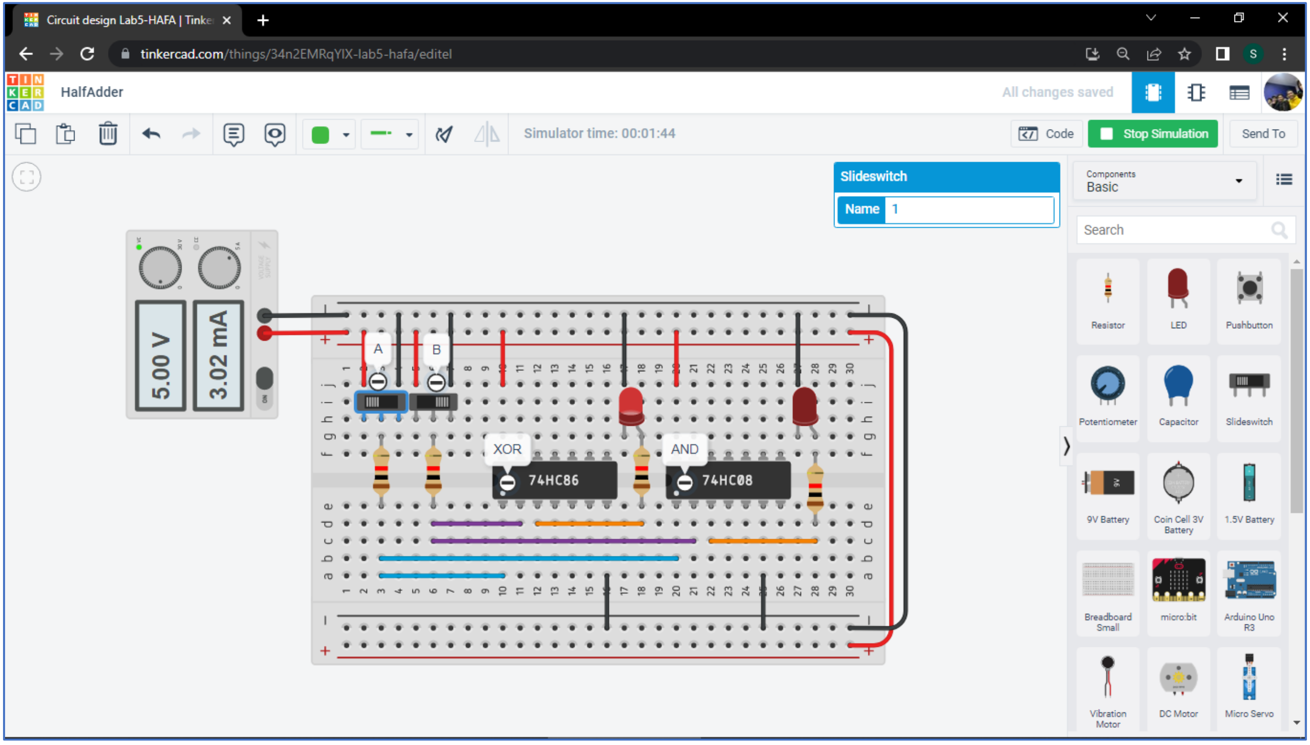


Slideswitch
Name 1

ITS 62704

Computer Organization & Architecture

Rajesh Maharjan



Circuit design Lab5-HAFA | Tinke

tinkercad.com/things/34n2EMRqYIX-lab5-hafa/editel

HalfAdder

All changes saved

Simulator time: 00:01:44

Code

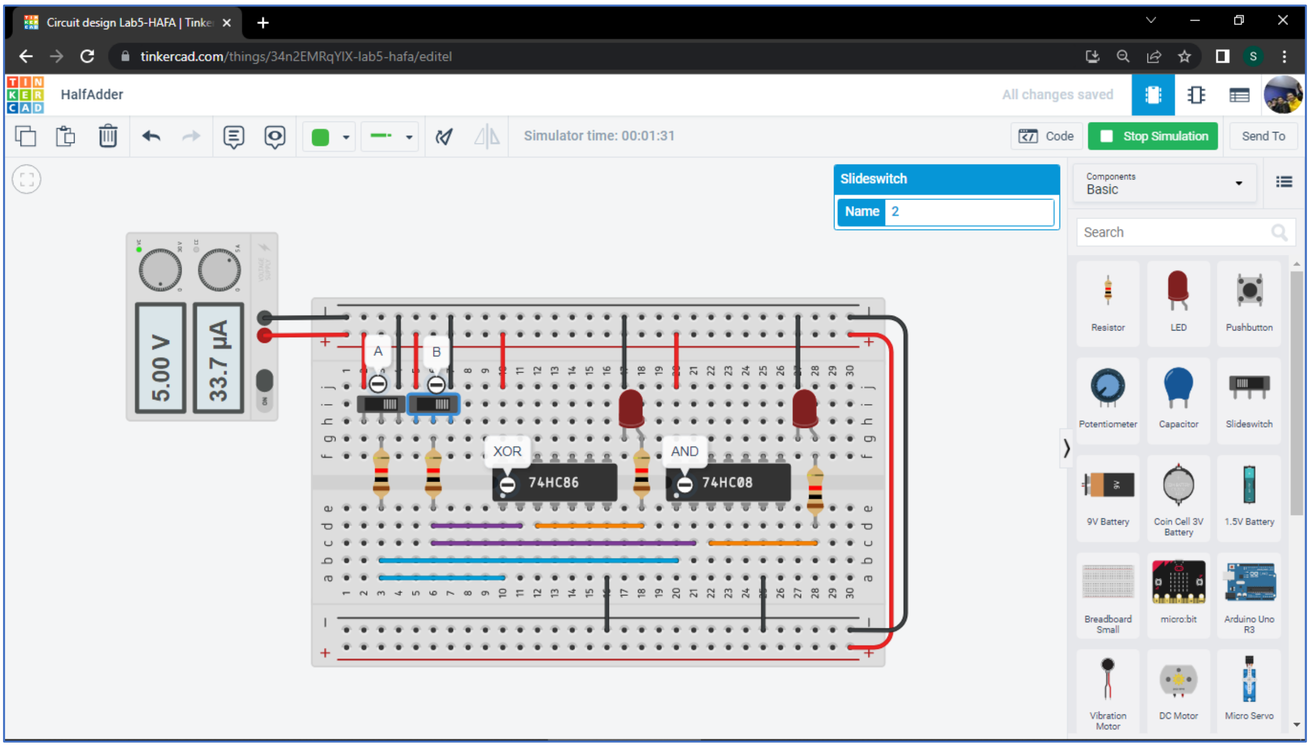
Stop Simulation

Send To

Slideswitch

Name 1

Components Basic



Circuit design Lab5-HAFA | Tinke

tinkercad.com/things/34n2EMRqYIX-lab5-hafa/editel

HalfAdder

All changes saved

Simulator time: 00:01:31

Code

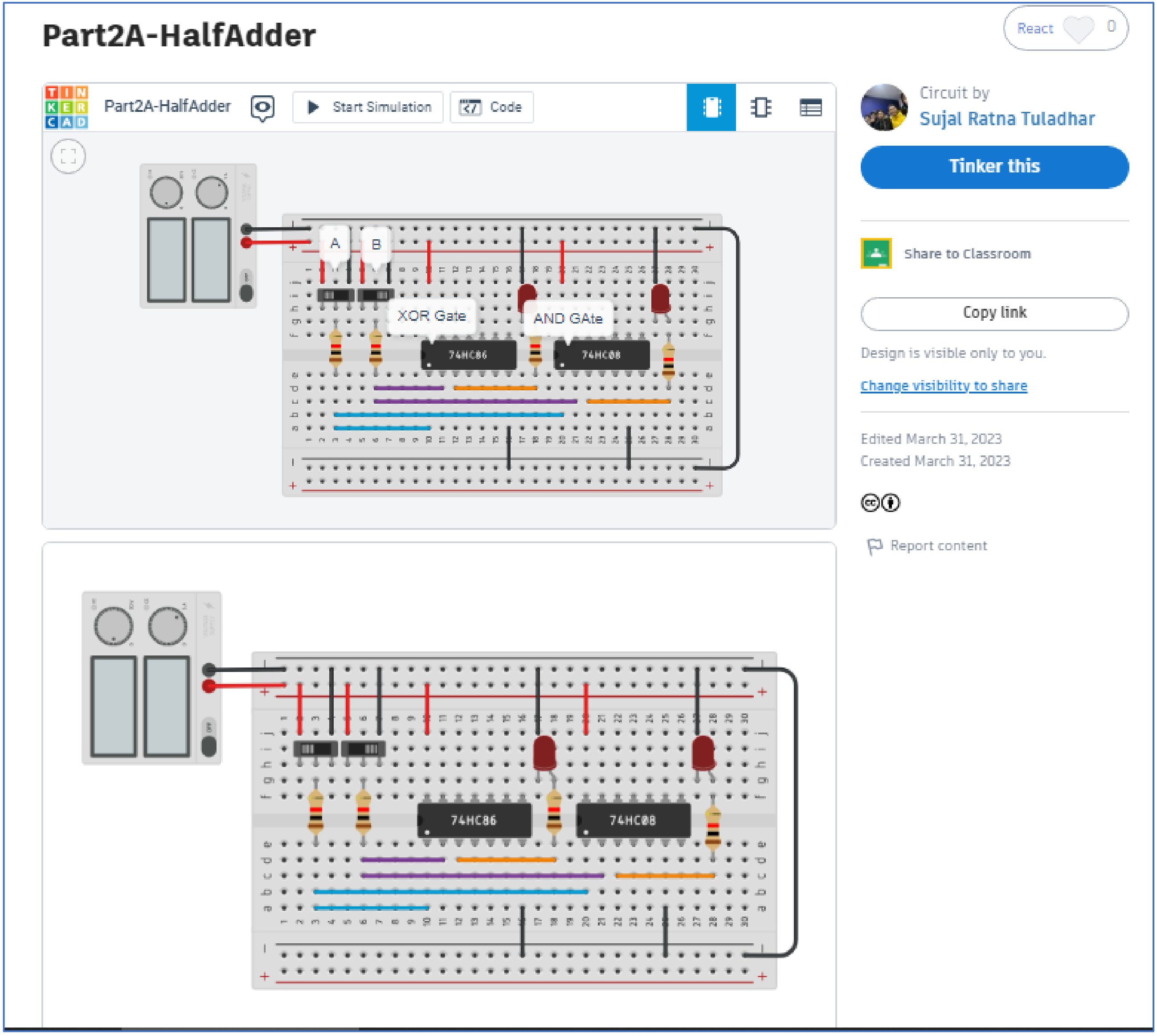
Stop Simulation

Send To

Slideswitch

Name 2

Components Basic



Part2A-HalfAdder

Circuit by
Sujal Ratna Tuladhar

Design is visible only to you.

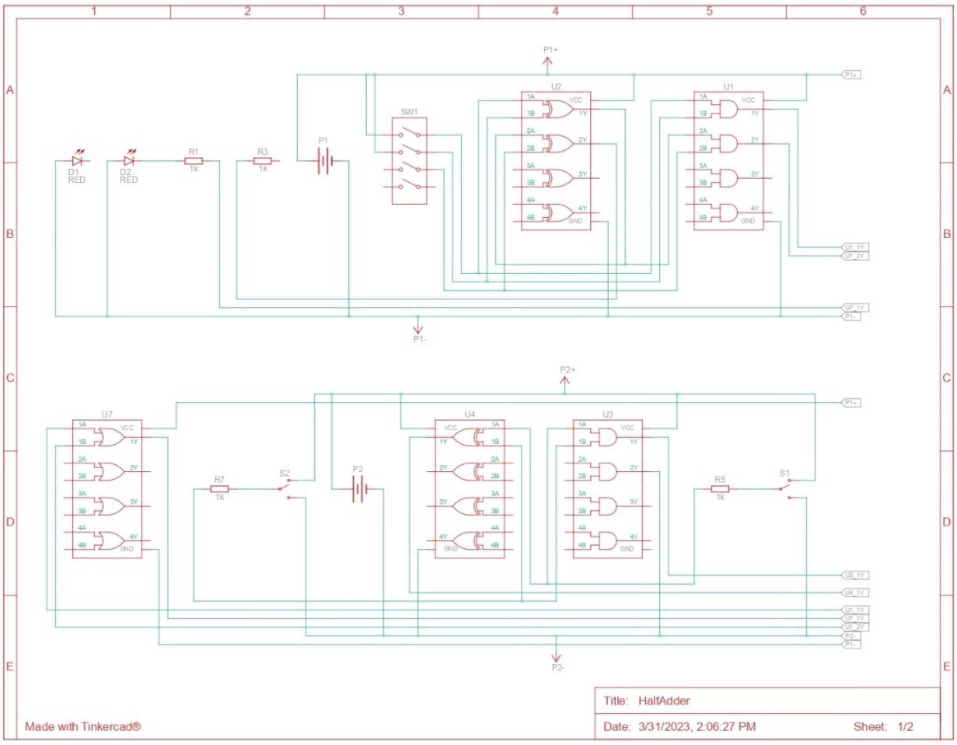
Change visibility to share

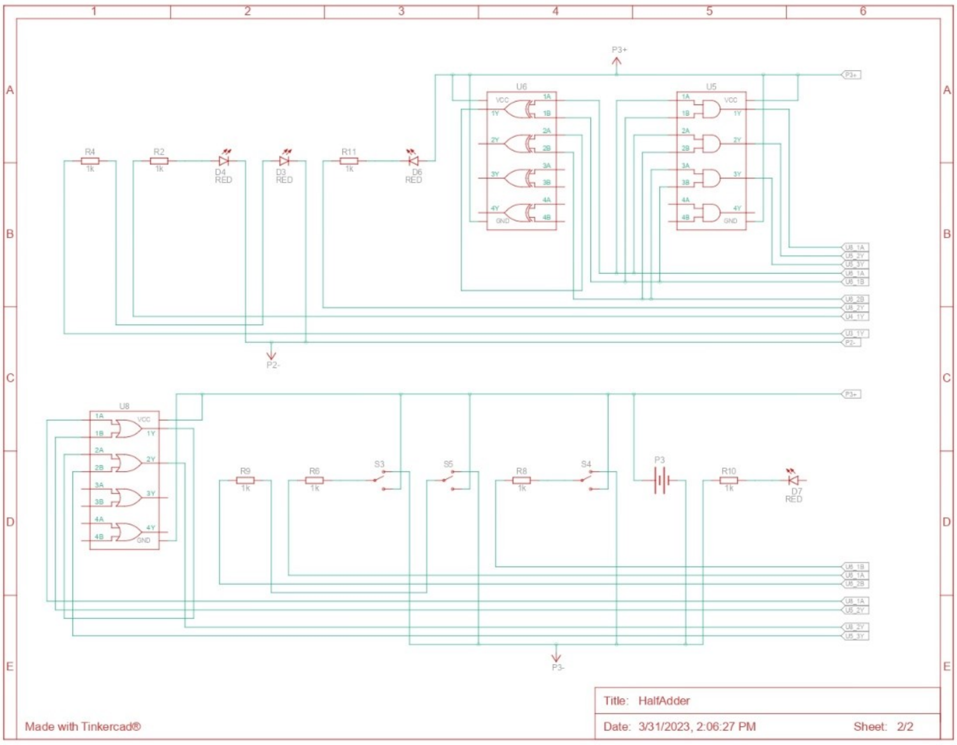
Edited March 31, 2023
Created March 31, 2023

ITS 62704

Computer Organization & Architecture

Rajesh Maharjan





Part B: Truth Table

INPUTS		OUTPUTS	
A	B	SUM	CARRY OUT
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

K MAP

SUM

A B	0	1
0	0	1
1	1	0

CARRY OUT

A B	0	1
0	0	0
1	0	1

Part C: Review Questions

Q. 1. What is the use of a half adder?

Half adder is a type combinational logic circuit which is used to add two 1 bit-binary numbers and results in two-bit output and is used in calculators, computers, digital measuring devices and so on.

Q. 2. Why is full adder better than half adder?

Full adder is better than half adder because half adder only adds the current inputs and does not focus on previous inputs while full adder can easily carry the current inputs as well as the output of previous addition.

Q. 3. In the output of half adder, what can be observed?

We have two outputs where we can see in sum the output is 1(true) when only one output is (1) i.e., two of inputs are true and two are false, and in carry out only 1 output is true when both input is true.

Inputs		outputs	
a	b	sum	Carry out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Q. 4. How many AND gate required to make a Half adder?

1 AND gate is required to make a half adder.

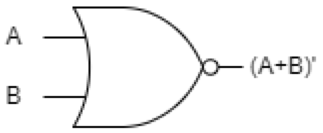
Q. 5. What is the use of a Full adder?

Full adder is a type of combinational logic circuit that takes three of 1-bit binary inputs and is used in multiple bit addition and digital operations.

Introduction

Logic Gates: building blocks for digital circuits, perform basic logical functions based on Boolean algebra in binary condition

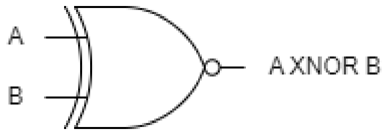
NOR: OR gate followed by NOT gate; output is “true” if both inputs are “false” else “false



INPUTS		OUTPUT
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

$$X = A\ NOT\ OR\ B = A\ NOR\ B = \overline{A+B}$$

XNOR: exclusive-NOR, combination of XOR gate followed by NOT gate. Output is “true if the inputs are the same, and “false” if inputs are different.



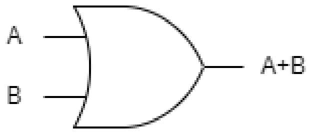
INPUTS		OUTPUT
A	B	X
0	0	1

0	1	0
1	0	0
1	1	1

$$X = A\ XNOR\ B = A \ominus B = \overline{AB} + AB$$

The output is 1 when at least one of the two inputs is 1.

OR: output is “true id ither or both inputs are “true” otherwise “false” aka Disjunction

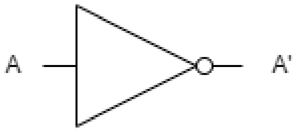


INPUTS		OUTPUT
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

$$X = A\ OR\ B = A + B$$

The input is opposite of output.

NOT: logical inverter, reverses the input and output aka Negation/ Complement



INPUTS	OUTPUT
A	$\bar{A}$
0	1
1	0

$$X = NOT\ A = \bar{A}$$

Boolean Algebra: branch category of algebra where it deals with truth values, true (1) and false (0) to simplify digital circuits and gates. or Binary or Logical Algebra. Fundamental component of digital electronics

Boolean Variable: Alphabetical letters used to define a symbol to represent logical quantities

Truth Table: table that gives all possible values of logical combinations. Number of rows should be equal to 2^n, where "n" is the number of variables/inputs.

Boolean Algebra Theorems: de Morgan's First and Second Law which reduce expression in simplified form by changing its form

1st: Complement of Product is equal to Sum of Individual Complements

2nd: Complement of Sum is equal to Product of Individual Complements

ITS 62704
Computer Organization & Architecture
Rajesh Maharjan

Boolean Theorem/Laws:

1a. $X \cdot 0 = 0$	1b. $X + 1 = 1$	Annulment Law
2a. $X \cdot 1 = X$	2b. $X + 0 = 0$	Identity Law
3a. $X \cdot X = X$	3b. $X + X = X$	Idempotent Law
4a. $\overline{X} \cdot X = 0$	4b. $\overline{X} + X = 0$	Complement Law
5a. $\overline{\overline{X}} = X$		Double Negation Law
6a. $X \cdot Y = Y \cdot X$	6b. $X + Y = Y + X$	Commutative Law
7a. $X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z = (X \cdot Z) = Y = XYZ$		Associative Law
7b. $X + (Y + Z) = (X + Y) + Z = (X + Z) + Y = X + Y + Z$		
8a. $X \cdot (Y + Z) = XY + XZ$		Distributive Law
8b. $X + YZ = (X + Y) \cdot (X + Z)$		
9a. $\overline{X \cdot Y} = \overline{X} + \overline{Y}$	9b. $\overline{X + Y} = \overline{X} \cdot \overline{Y}$	de Morgan's Theorem
10a. $X \cdot (X + Y) = X$	10b. $X + XY = X$	Absorption Law
11a.$(X + Y) \cdot (X + \bar{Y}) = X$	11b. $XY + X\bar{Y} = X$	Redundancy Law
12a. $(X + \overline{Y}) \cdot Y = XY$	12b. $X\overline{Y} + Y = X + Y$	
13a. $(X + Y) \cdot (\bar{X} + Z) \cdot (Y + Z) = (X + Y) \cdot (\bar{X} + Z)$		Consensus Law
13b. $XY + \bar{X}Z + YZ = XY + \bar{X}Z$		
14a. $X \oplus Y = (X + \overline{Y}) \cdot (\bar{X} + Y)$		XOR Gate
14b. $X \oplus Y = XY + XY$		
15a. $X \odot Y = (X + Y) \cdot (\bar{X} \cdot \bar{Y})$		XNOR Gate
15b. $X \odot Y = \overline{X\,Y} + X\,Y$		
15c. $X \odot Y = (X + Y) \cdot (\overline{X} + \overline{S})$		

Binary adder: digital circuit that performs the arithmetic binary addition of two numbers with respect to the logic operations and laws of Boolean Algebra.

Half adder:

- digital logic circuit that performs binary addition of two single bit binary numbers.
- Two inputs and 2 outputs (sum and carry out)
- Sum output is least significant bit (lsb); xor $\text{Sum}, S=A\oplus B=AB'+A'B$
- Carry out is most significant bit(msb) indicating carryover from additional of two inputs; and$\text{Carry}, C=A\cdot B$
- Simplest and Building block of more complex adder circuits as full adder and multiple bit adder
- Combinational arithmetic circuit

Tuth table

$0+0=0$

$0+1=1$

$1+0=1$

$1+1=10$ (Sum = 0 & Carry = 1)

Inputs		outputs	
a	b	sum	Carry out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

K map

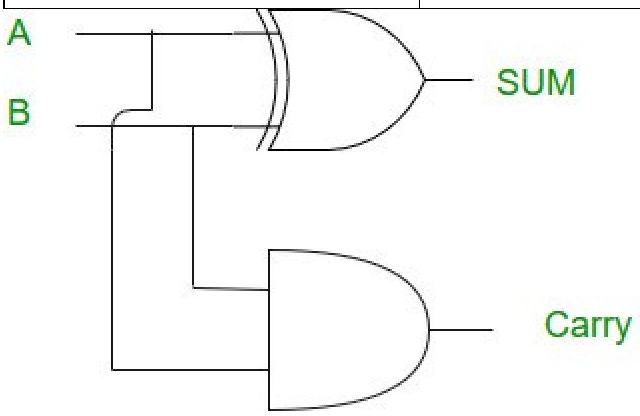
sum

a b	0	1
0	0	1
1	1	0

Carry out

a	0	1

b		
0	0	0
1	0	1



Applications:

Half adder is used in ALU (Arithmetic Logic Unit) of computer processors to add binary bits.

Half adder is used to realize full adder circuit.

Half adder is used in calculators.

Half adder is used to calculate addresses and tables.

Full adder: ripple carry

Adds three inputs and produce two outputs

First two inputs are a and b third is cin

Cin has cout, other have sum

It can take 8 inputs to create a byte-wide adder, to cascade carry bit from one adder to another

Truth

inputs			Outputs	
a	b	C in	sum	C 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

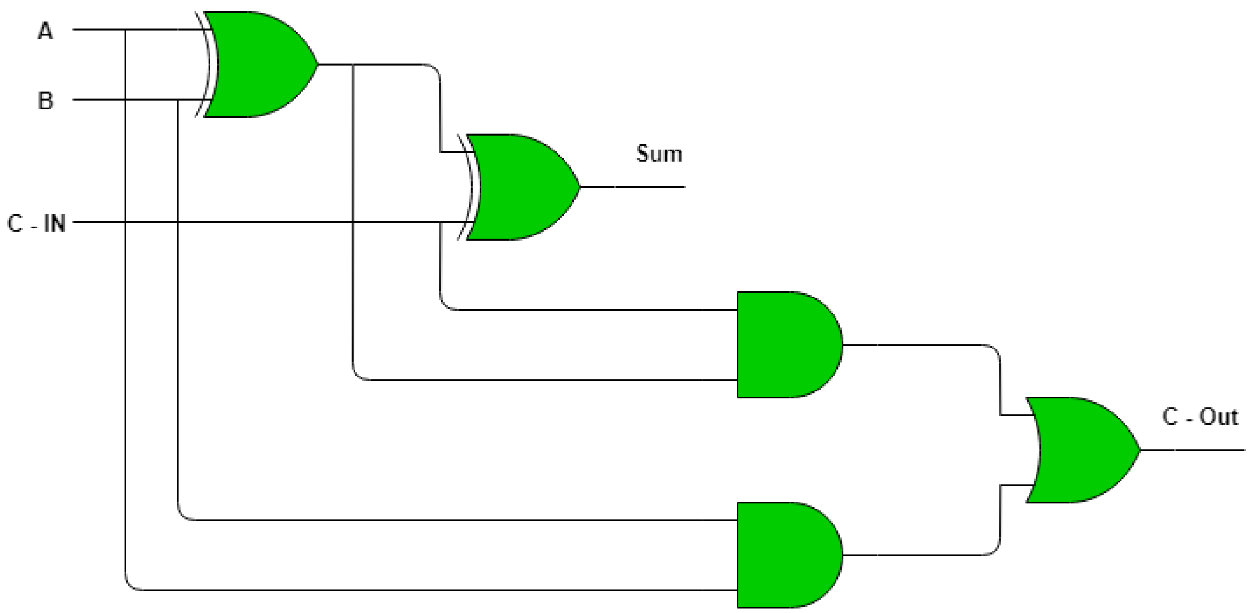
sum $S=A\oplus B\oplus Cin$

c Ab	0	1
00		1
01	1	
11		1
10	1	

Carry out C=AB+BCin+ACin

c Ab	0	1
00		
01		1
11	1	1

10		1



Advantages of Full Adder

The following are the important advantages of full adder over half adder −

Full adder provides facility to add the carry from the previous stage.

The power consumed by the full adder is relatively less as compared to half adder.

Full adder can be easily converted into a half subtractor just by adding a NOT gate in the circuit.

Full adder produces higher output that half adder.

Full adder is one of the essential part of critic digital circuits like multiplexers.

Full adder performs operation at higher speed.

Applications of Full Adder

The following are the important applications of full adder −

Full adders are used in ALUs (arithmetic logic units) of CPUs of computers.

Full adders are used in calculators.

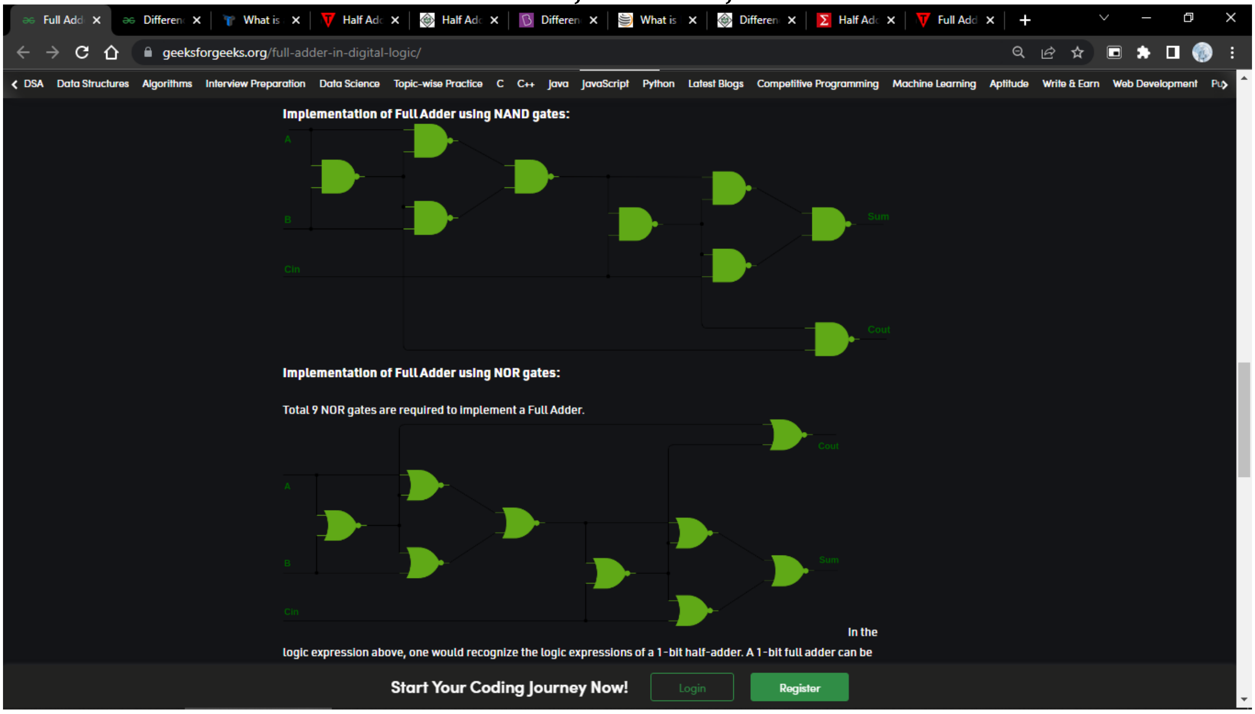
Full adders also help in carrying out multiplication of binary numbers.

Full adders are also used to realize critic digital circuits like multiplexers.

Full adders are used to generate memory addresses.

Full adders are also used in generation of program counterpoints.

Full adders are also used in GPU (Graphical Processing Unit)



Implementation of Full Adder using NAND gates:

Implementation of Full Adder using NOR gates:

Total 9 NOR gates are required to implement a Full Adder.

In the logic expression above, one would recognize the logic expressions of a 1-bit half-adder. A 1-bit full adder can be

AND: True if A and B are both true

OR: True if A or B are true

XOR: True if either A or B are true, false if both are true

NOT: Inverted; false if input is true, true if input is false

NAND: AND followed by NOT; false if both A and B are true

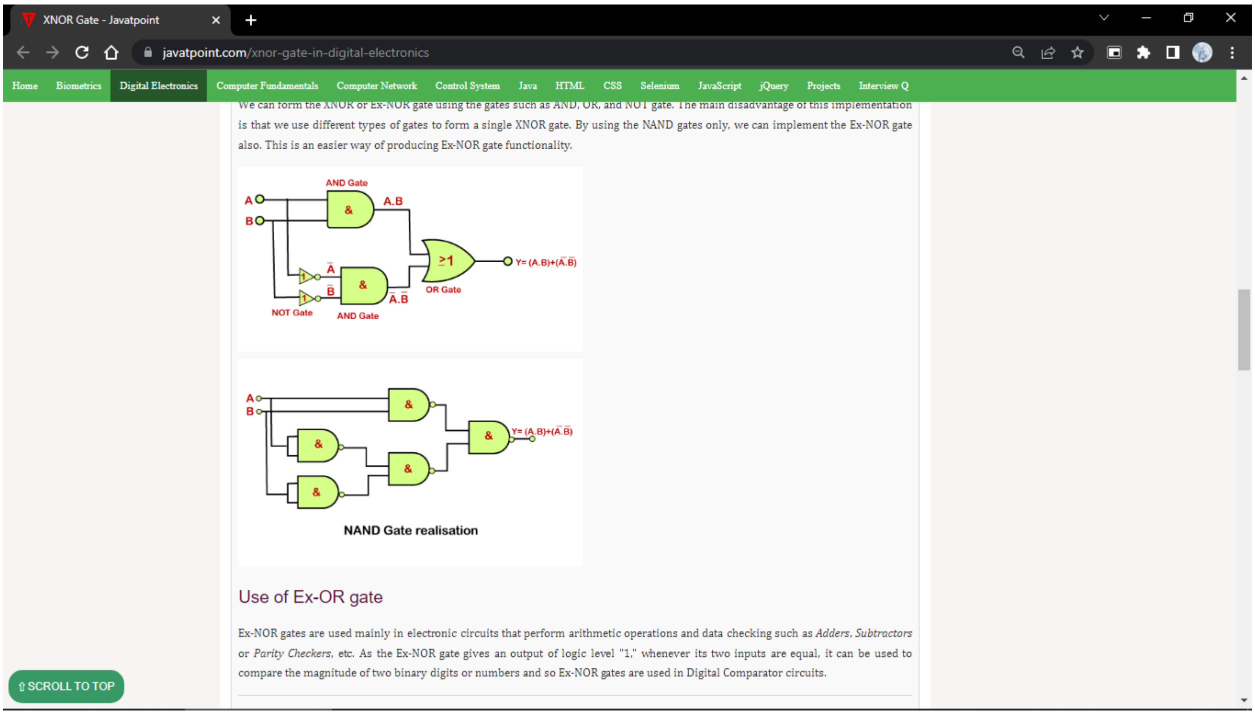
NOR: OR followed by NOT; true if both A and B are false

XNOR: XOR followed by NOT; true if A and B are both true or both false

F(A,B,C,D)=A+BC'+D

The output will be high when A=1 or BC'=1 or D=1 or all are set to 1. The truth table of the above example is given below. The 2^n is the number of rows in the truth table. The n defines the number of input variables. So the possible input combinations are 2^3=8.

Inputs				Output
A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



We can form the XNOR or Ex-NOR gate using the gates such as AND, OR, and NOT gate. The main disadvantage of this implementation is that we use different types of gates to form a single XNOR gate. By using the NAND gates only, we can implement the Ex-NOR gate also. This is an easier way of producing Ex-NOR gate functionality.

Use of Ex-OR gate

Ex-NOR gates are used mainly in electronic circuits that perform arithmetic operations and data checking such as *Adders, Subtractors* or *Parity Checkers*, etc. As the Ex-NOR gate gives an output of logic level "1," whenever its two inputs are equal, it can be used to compare the magnitude of two binary digits or numbers and so Ex-NOR gates are used in Digital Comparator circuits.