

# *ECE216: Digital Electronics Laboratory*

## Table of Content

Task	Title	Page No.
✓ Experiment 1	Understanding the combinational logic by implementing the boolean function using basic logic gates	1
★ Experiment 2	To design and analyze the circuit for <u>Full adder</u> and Full subtractor using Logic Gates.	6
<b>Practical work evaluation 1</b>	Practical work evaluation based on Experiment 1 and Experiment 2.	9
Experiment 3	Understanding the combinational logic by implementing the boolean function using multiplexer	12
Experiment 4	Understanding the combinational logic by implementing the boolean function using decoder	16
<b>Practical work evaluation 2</b>	Practical work evaluation based on Experiment 3 and Experiment 4.	20
<b>Project evaluation 1</b>	Design and Implementation of application-based projects-1	23

Experiment 5	Understanding the sequential logic by implementing the flip flop with the help of logic gates	26
Experiment 6	Understanding the sequential logic by implementing the counter with flip flop.	28
<b>Practical work evaluation 3</b>	Practical work evaluation based on Experiment 5 and Experiment 6.	31
Experiment 7	To visualize the output of decade counter on seven segment display	34
Experiment 8	To implement and simulate combinational and sequential circuit using DSCH/Proteus.	37
<b>Practical work evaluation 4</b>	Practical work evaluation based on Experiment 7 and Experiment 8.	41
<b>Project evaluation 2</b>	Design and Implementation of application-based projects-2	44

## Experiment 2

**Aim:** To design a circuit for Full adder and full subtractor using X-OR and basic gates

1. **Apparatus Required:** - IC 7486, IC 7432, IC 7408, IC 7400, IC 7404 etc.

2. **Learning objective:** XOR OR AND NAND NOT

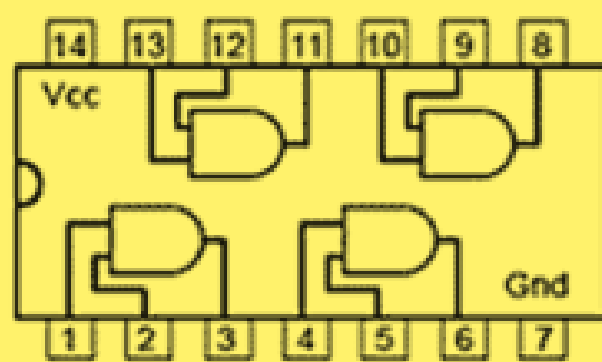
a) How to realize the functionality full adder.

3. **Theory:**

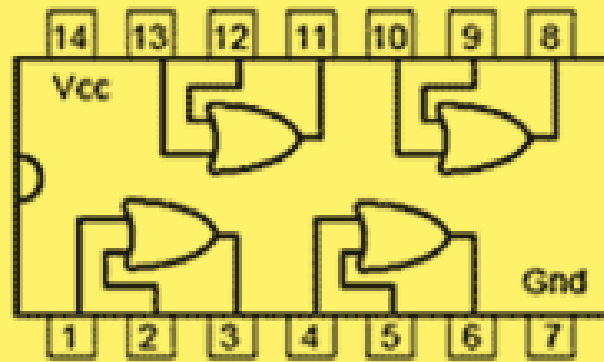
1) Using X – OR and Basic Gates to implement full Adder:

A full adder adds binary numbers and accounts for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as A, B, and Cin ; A and B are the operands, and Cin is a bit carried in from the next less significant stage. The full-adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. binary numbers. The circuit produces a two-bit output sum typically represented by the signals Count and S.

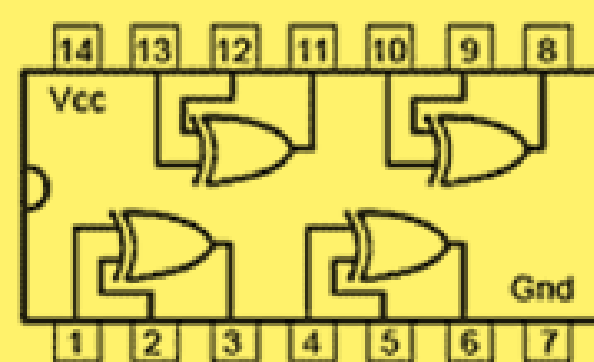
In this implementation, the final OR gate before the carry-out output may be replaced by an XOR gate without altering the resulting logic. Using only two types of gates is convenient if the circuit



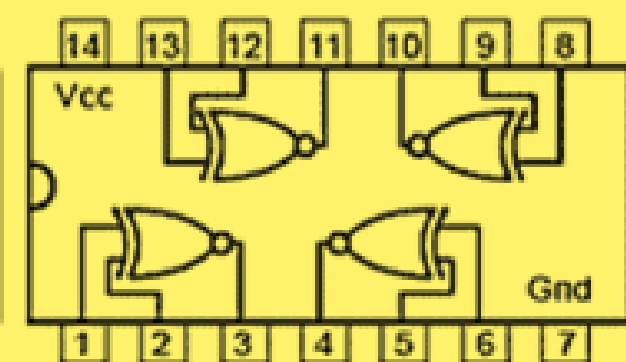
7408 Quad 2 input  
AND Gates



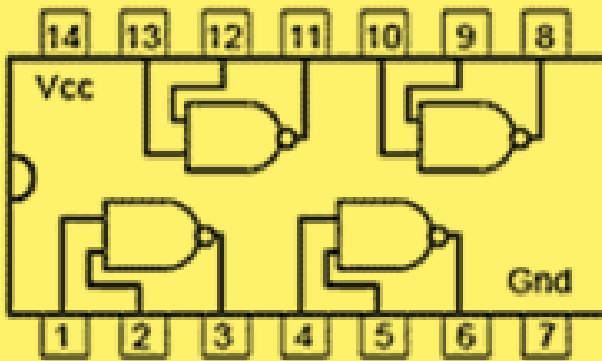
7432 Quad 2 input  
OR Gates



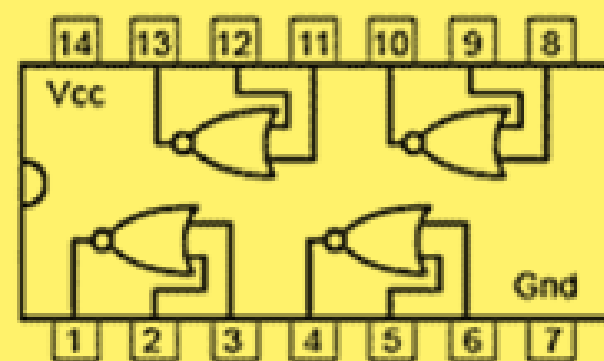
7486 Quad 2 input  
XOR Gates



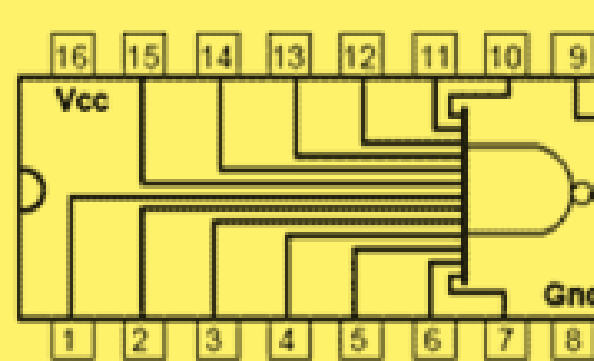
747266 Quad 2 input  
XNOR Gates



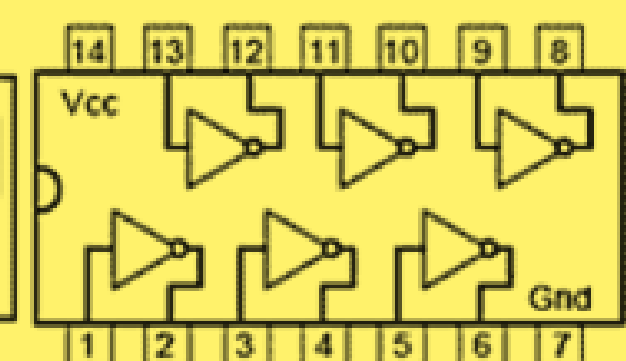
7400 Quad 2 input  
NAND Gates



7402 Quad 2 input  
NOR Gates



74133 Single 13 input  
NAND Gate



7404 Hex NOT Gates  
(Inverters)

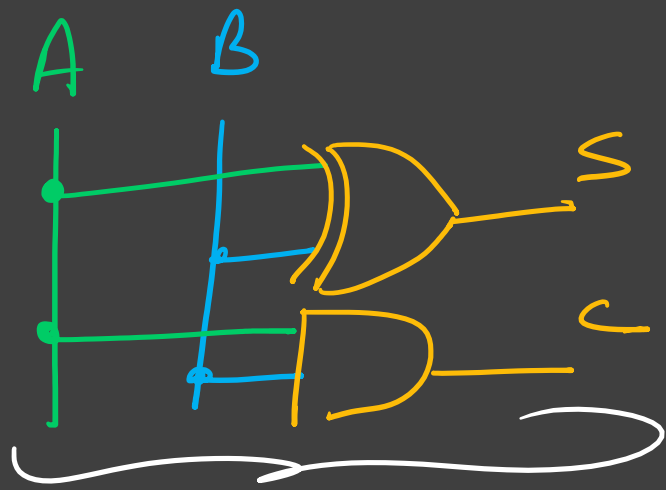
It can add 2, 1-bit binary number

# Half Adder

*g/p*

	A	B	S	C
0	0	0	0	0
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1

*o/p*



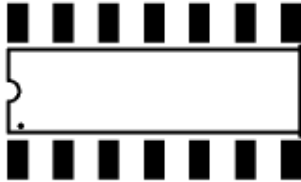
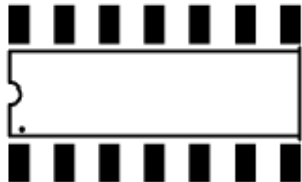
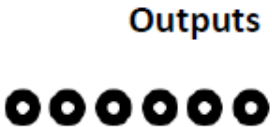
*sol*

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

$$C = AB$$

Half adder

Draw Bread Board Connection diagram:



GND



Inputs

# Full Adder

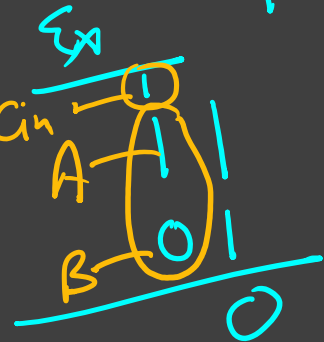
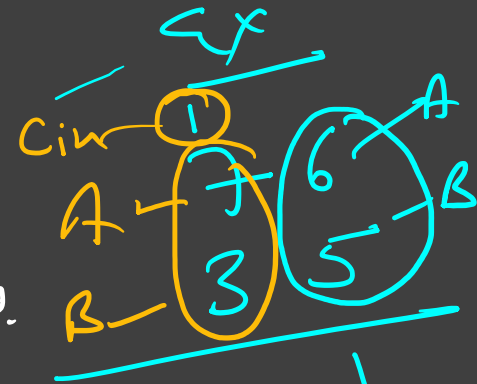
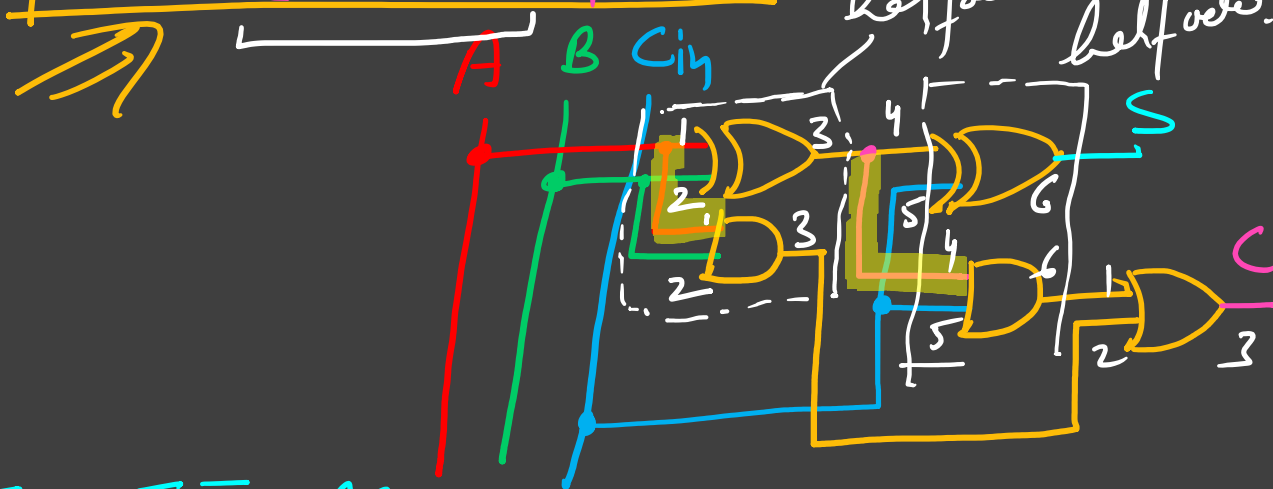
It can add 2, 1-bit Binary numbers, considering previous carry ( $A+B+C_{in}$ )

	A	B	$C_{in}$	$S$	$C_{out}$
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

$$C_{out} = \bar{A}B C_{in} + A\bar{B} C_{in} + AB\bar{C}_{in} + ABC_{in}$$

$$= (\bar{A}B + A\bar{B}) C_{in} + AB(\bar{C}_{in} + C_{in})$$

$$C_{out} = (A \oplus B) C_{in} + AB$$



$$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

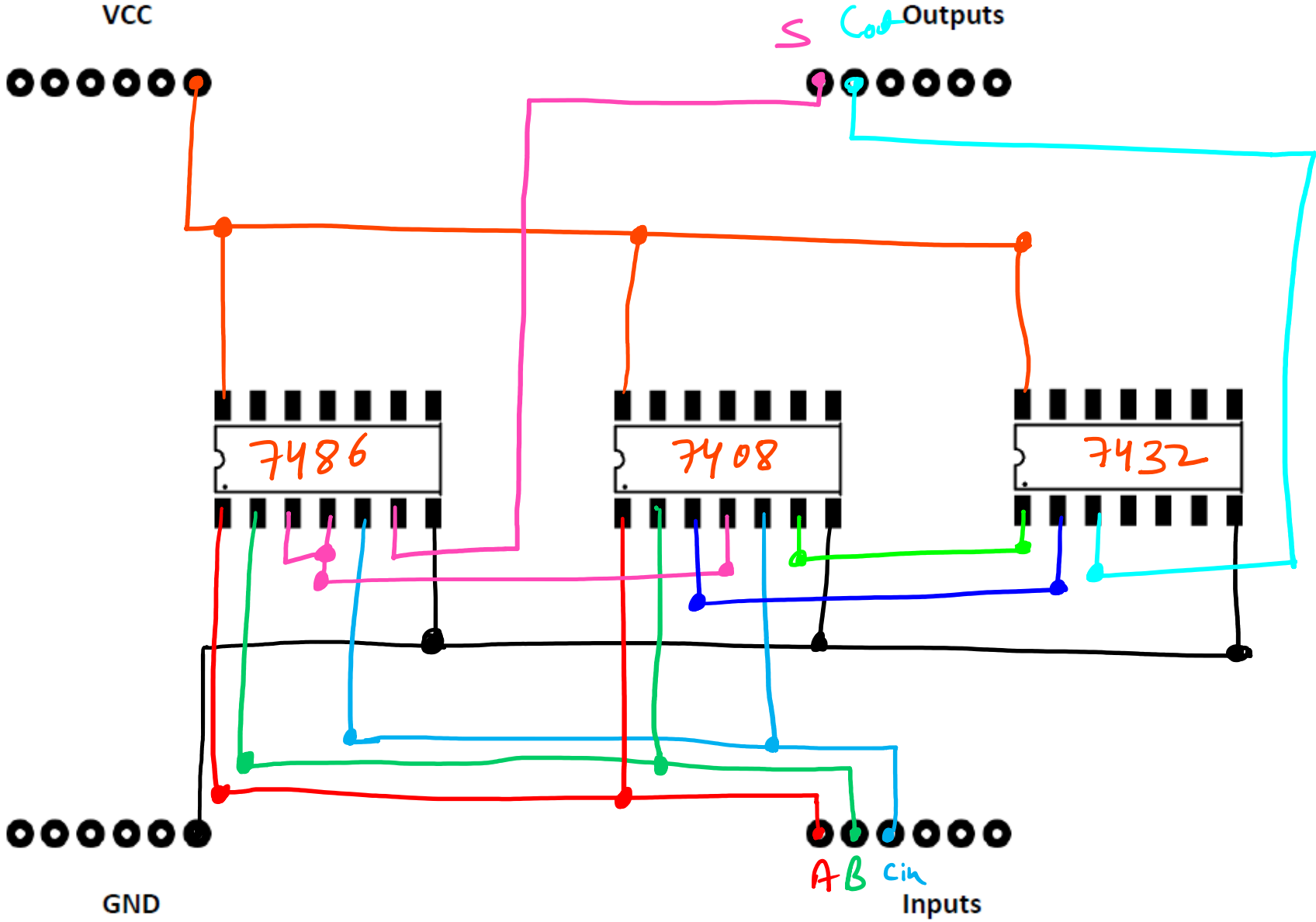
$$= (\bar{A}B + A\bar{B})\bar{C}_{in} + (\bar{A}\bar{B} + AB)C_{in}$$

$$= (A \oplus B)\bar{C}_{in} + \overline{A \oplus B} C_{in} = (A \oplus B) \oplus C_{in}$$

$$f_1 \bar{f}_2 + \bar{f}_1 f_2$$



Draw Bread Board Connection diagram:



# Half Subtractor

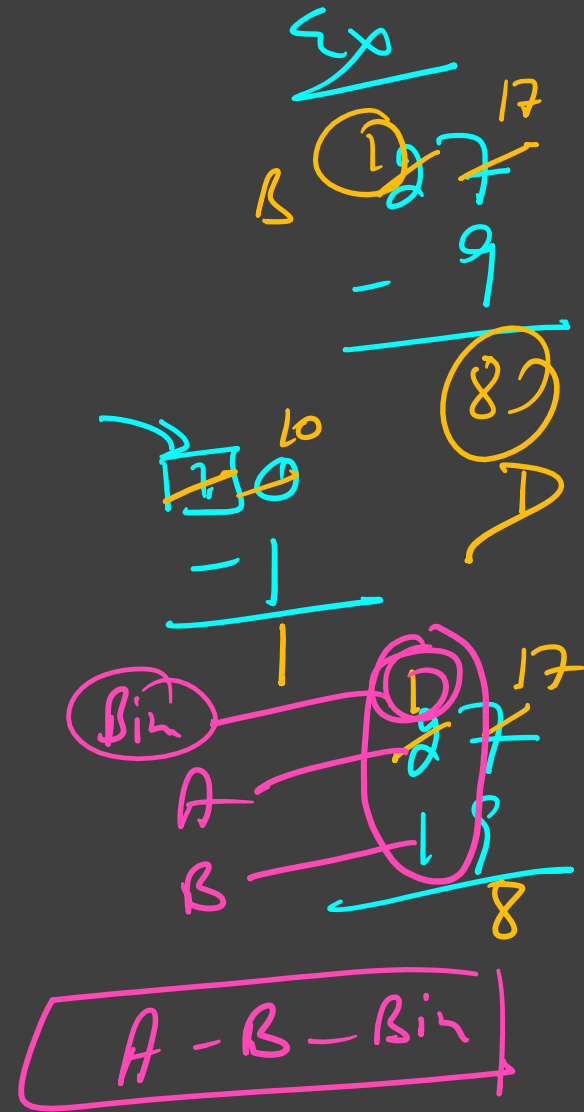
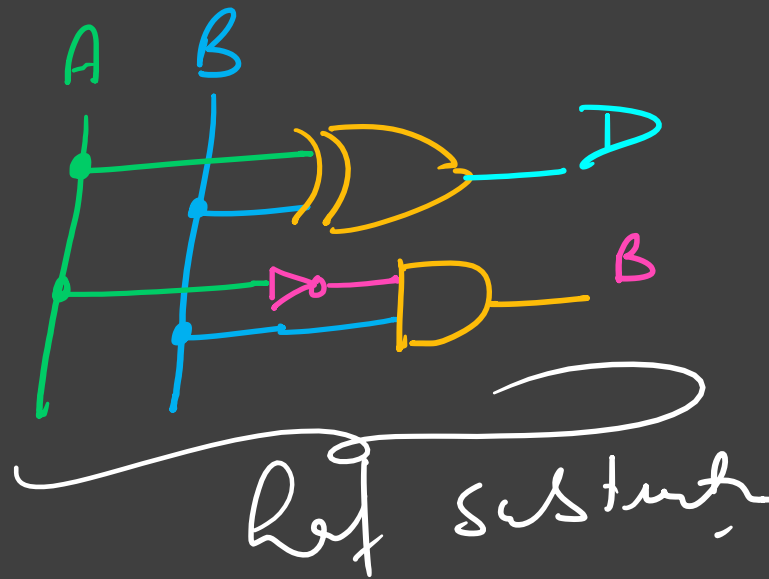
9. can subtract 2, 1-bit binary numbers  
 $A - B$

- 1
- 2
- 3

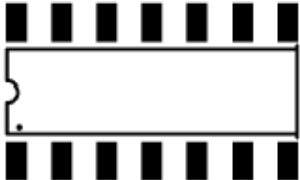
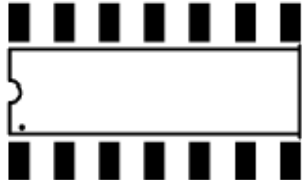
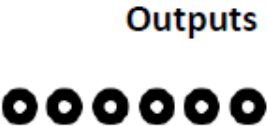
A	B	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D = \bar{A}B + A\bar{B} = A \oplus B$$

$$B = \bar{A}B$$



Draw Bread Board Connection diagram:



GND



Inputs

# Full Subtractor

	A	B	Bin	D	Bout
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	1	1

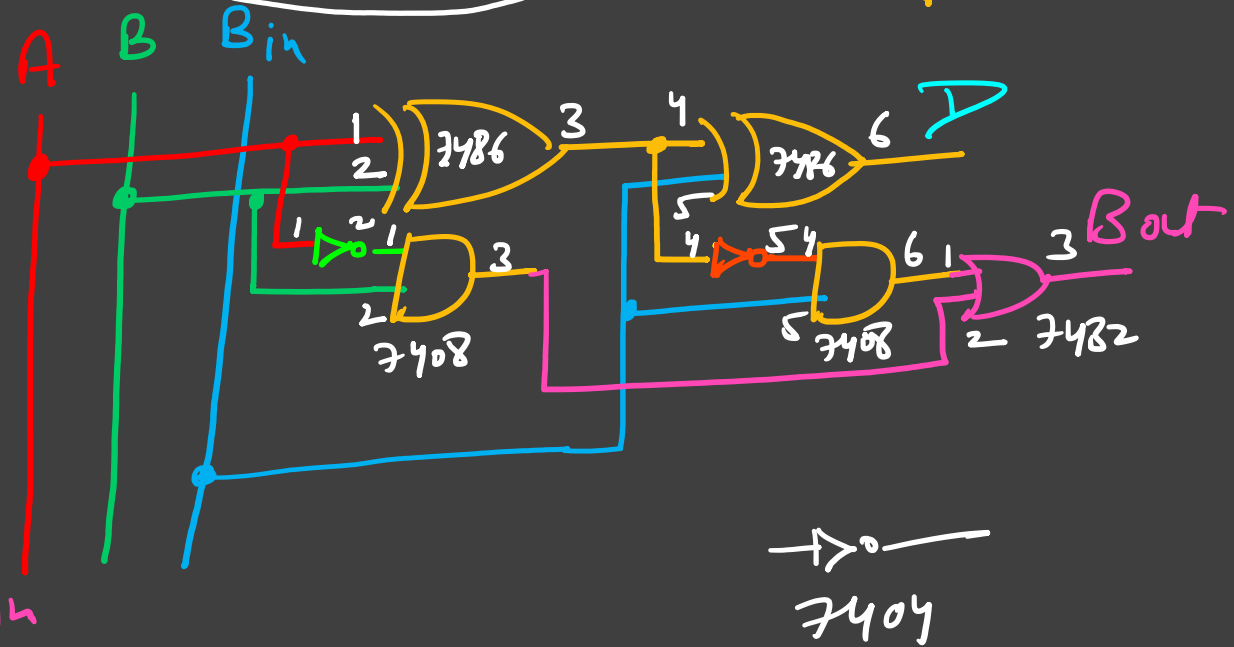
$$D = A \oplus B \oplus Bin$$

$$Bout = \bar{A} \bar{B} Bin + \bar{A} B \bar{Bin} + \bar{A} B Bin + A B Bin$$

It can add 2 1-bit binary number considering the previous sum  
 $(A - B) - Bin$

$$Bout = (\bar{A} \bar{B} + A B) Bin + \bar{A} B (\bar{Bin} + Bin)$$

$$Bout = (A \oplus B) Bin + \bar{A} B$$



Draw Bread Board Connection diagram:

for full subtractor (hw)

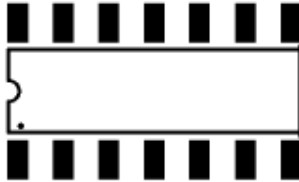
VCC



Outputs



7486  
7408  
7432  
7404



GND



Inputs

Test 2 — 8th Feb 2021

Google for

- MCQ / Then short answer question
- writeup / SA done
- Simulation

10 Marks

10 Marks

10 Marks