



AROR UNIVERSITY  
OF ART, ARCHITECTURE,  
DESIGN & HERITAGE,  
SUKKUR, SINDH

## **Faculty of Artificial Intelligence & Multimedia Gamming**

BS – Artificial Intelligence (Section A)

Digital Logic Design Lab

### **Lab # 03:**

Instructor: Mr. Abdul Ghafoor

Name:

Marks obtained:

Date: 19/02/2024

### **Lab Learning Objectives:**

Upon successful completion of this experiment, the student will be able:

- To implement half and full adders with logic gates and create truth tables from them
- To implement multiple full adders with each other to explore the ripple effect
- To verify the truth table for a full adder

### **Background Theory:**

## Binary-Coded Decimal

Value	8	4	2	1
Bit	0	0	1	1
Result	0	0	2	1

$$\text{Sum} = 0 + 0 + 2 + 1 = 3$$

Each “bit” is assigned a value

Multiply the value by the bit

Add the results together

The sum is the **binary-coded decimal** value

Figure 1-1 Video. View the video here: <https://youtu.be/YYGAPRracLY>



### Video Summary

- In larger circuits it is not practical to make truth tables so binary is converted into a binary-coded decimal (BDC)
- Half adders can be represented in Multisim by a single component or by creating them using an AND and an XOR gate
- Half adders have two inputs and two outputs
- Full adders can be represented in Multisim by a single component or two AND gates and three XOR gates
- Full adders have three inputs and two outputs

- **Binary-Coded Decimals**

Let's look at the following example of a 4-bit binary number (four binary numbers code to one decimal number):

0101

The number furthest to the right is given the value of 1 (similar to the ones column in regular addition). The position immediately to its left is given the value of 2 (similar to a tens column). The value to the left of this is 4 (similar to the hundreds column). The pattern you will notice is

that the values assigned to the number on the left increase by twice the value of the number before it.

In our example, from left to right, the BCD value for the four-bit binary code above, adding up the values is:

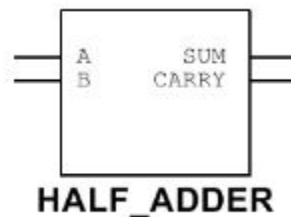
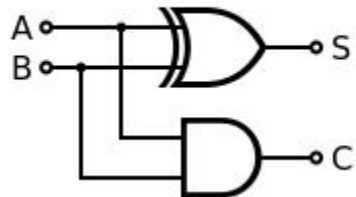
$$0 + 4 + 0 + 1 = 5$$

Conversions that result in a single digit number (0-9) are already in BCD format. In order to output the decimal value of a binary number greater than 9 in circuitry, a Binary-to-BCD converter such as a Shift-Add-3 algorithm must be used.

## Adders

### Half-Adders:

- A half-adder does binary addition on two inputs (A and B).
- The two outputs are labeled sum (S) and carry (C).
- Half adders can be built with:
  - an XOR gate (7486) and an AND gate (7408) (shown on the left).
  - a component in Multisim (shown on the right).



*Figure 1-2 Half Adder from Gates*

*Figure 1-3 Half Adder Multisim*

### Full-Adders:

- A full-adder does binary addition on three inputs: A, B, and  $C_{in}$ .
- Full adders usually work in a cascade fashion where they are used to add binary numbers with an increasing number of bits.
- The two outputs are sum (S) and carry ( $C_{out}$ ).
- You will notice that full adders can also use logic gates or a component.

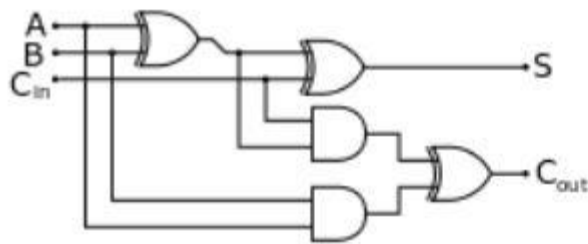


Figure 1-4 Full Adder from Gates

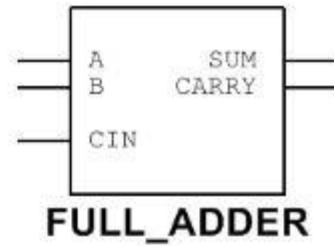


Figure 1-5 Full Adder Multisim

1-1 Why is it hard to use truth tables as the sequence of binary digits increases?

---



---



---



---

1-2 What are the similarities and differences between half adders and full adders?

---



---



---

1-3 When full adders are connected to each other such that  $C_{out}$  of one lead to the  $C_{in}$  of the other, this is called rippling. What do you think this means?

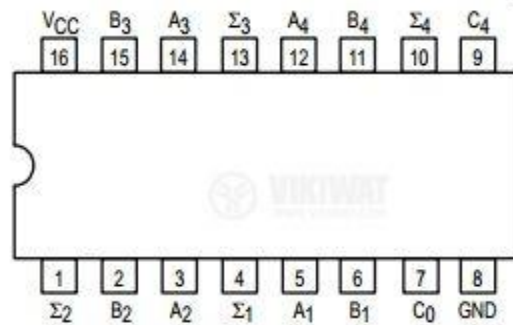
---



---



---



Full Adder 74LS283

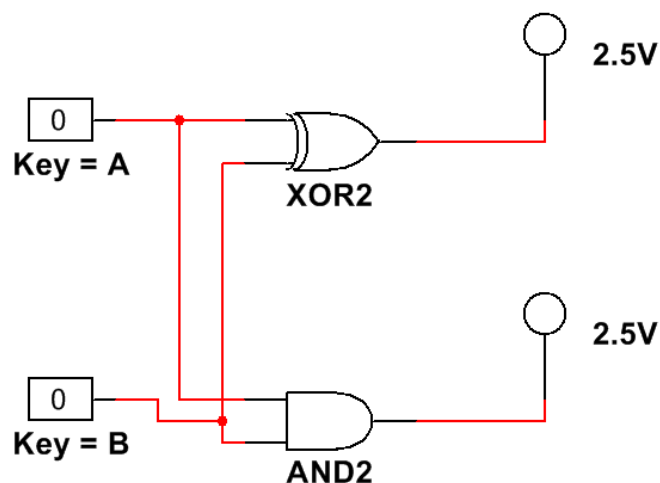
### Lab Example:

Build the following Half-Adder circuit using multiple Gates in Multisim:

### Half-Adder Circuits

Half-adder circuits can be built using a combination of logic gates.

- Launch Multisim.
- Open a new circuit.
  - Select **File>>New**.
  - In the menu that appears, select **Blank** and click **Create**.
- Connect the following circuit:
  - Place an **XOR** gate and an **AND** gate from the **Misc Digital** group.
  - Place two **INTERACTIVE\_DIGITAL\_CONSTANTS** from the **Sources** group.
  - Place two **PROBE\_DIG\_REDS** from the **Indicators** group.
  - Wire them as shown:



*Figure 1-6 A Half Adder Circuit*

- Click the **Run** button to begin simulating the circuit.



*Figure 1-7 Run Button*

- Using the **A** and **B** keys, vary the inputs into the circuit.

1-4 Fill out the truth table below.

A	B	XOR (SUM)	AND (CARRY)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Stop the simulation by clicking the **Stop** button.



*Figure 1-8 Stop Button*

*Can you see how the XOR and AND gates represent the sum and carry of the numbers A and B added together?*

- $0 + 0 = 0$
- $1 + 0$  or  $0 + 1 = 1$ , with no carry.
- $1 + 1 = 2$ , but 2 is not a binary number. In binary, 2 is represented as 10. The 1 is the carry and the 0 is the sum.

### **Lab Activity:**

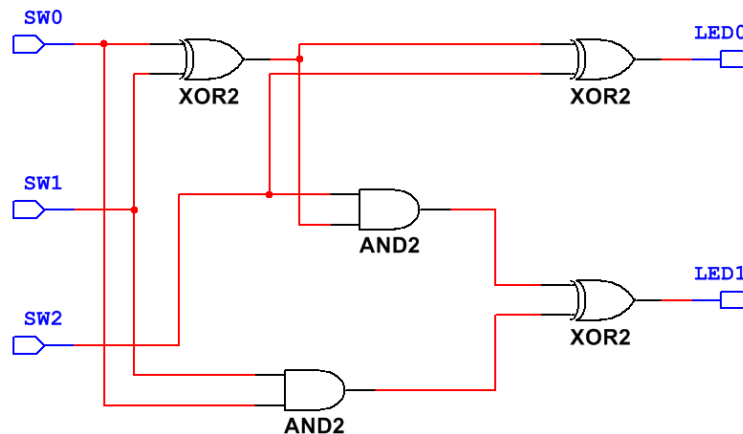
Build the following Full-Adder circuit using multiple Gates in Multisim:

### **Full-Adder Circuits**

Open a new Multisim File

Connect the following circuit:

- Place three **XOR** gates and two **AND** gates from the **Misc Digital** group.
- Place the input connectors **SW0**, **SW1**, and **SW2**.
- Place the output connectors **LED0** and **LED1**.
- Wire them as shown:



*Figure 1-9 A Full Adder Circuit*

1-5 Fill out the truth table below.

SW0	SW1	SW2	SUM	C <sub>out</sub>
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

*Again, can you see how the truth table represents the sum and carry of the numbers A, B, and C<sub>IN</sub> added together?*

- $0 + 0 + 0 = 0$
- $0 + 0 + 1 = 1$ , with no carry.
- $0 + 1 + 1 = 2$ , but 2 is not a binary number. In binary, 2 is represented as 10. The 1 is the carry and the 0 is the sum.
- $1 + 1 + 1 = 3$ . In binary, 3 is represented as 11. The first 1 is the carry and the second 1 is the sum.

### Lab Exercise:

- Build the Half-Adder circuit using XOR and AND gate on NI ELVIS II/Multisim, verify the truth table and attach the circuit Picture
- Build the 4-bit Adder circuit using **Full Adder IC 74LS283** on NI ELVIS II/Multisim, verify the truth table and attach the circuit Picture
- Design the Half-Adder, Full-Adder and 4-bit Adder using PLD (Programmable Logic Design)

### Conclusion:

1-8 Under what conditions would you use a half adder? A full adder?

---

---

---

---

1-9 What is the use of the Full Adder component within Multisim? In other words, why wouldn't you just place XOR and AND gates for each adder?

---

---

---

---

1-10 When converting binary numbers to BCD:

- A. Numbers are given specific values depending on their position in the sequence
- B. The value of the number on the left increases by twice the value of the number on the right
- C. Single digit numbers are already in BCD format
- D. All of the above

1-11 Half-adders do binary addition on two inputs using two logic gates. What is the correct combination of these logic gates?

- A. XOR and OR



- B. XOR and AND
- C. XNOR and AND
- D. NOT and OR

1-12 The outputs of a half adder are:

- A. Carry and  $C_{in}$
- B. Sum and Carry
- C. Sum and  $C_{in}$
- D. Carry and  $C_{out}$

1-13 Which is a feature of full-adders?

- A. They have two inputs
- B. They have three outputs
- C. They work in a cascading fashion
- D. All of the above

1-14 When connecting full-adders together in Multisim, the Carry of one is connect to \_\_\_\_\_ input of the next.

- A. A
- B. B
- C.  $C_{in}$
- D. None of the above