

**Department of Computer Science and Engineering**  
**BRAC University**  
**CSE 260: Digital Logic Design**

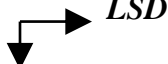
**Experiment # 5**

***Design and Implementation of 4-bit Parallel Binary Adder***

**Theory:** The addition of two binary numbers is performed in exactly the same manner as the addition of decimal numbers.

Let us first review the decimal addition

$$\begin{array}{r} \phantom{0}3 \phantom{0}7 \phantom{0}6 \\ \phantom{0}4 \phantom{0}6 \phantom{0}1 \\ \hline \phantom{0}8 \phantom{0}3 \phantom{0}7 \end{array}$$

 **LSD**

The least significant digit position is operated on first, producing a sum of 7. The digits in the second position are then added to produce a sum of 13, which produces a **carry** of 1 into the third position. This produces a sum of 8 in the third position.

The same general steps are followed in binary addition. However only four cases can occur in adding the two binary digits (bits) in any position. They are

$$0+0=0$$

$$1+0=1$$

$$1+1=10=0+\text{carry of } 1 \text{ into the next position}$$

$$1+1+1=11=1+\text{carry of } 1 \text{ into the next position}$$

Here are several examples of the addition of two binary numbers:

$$\begin{array}{r} 1001 \\ 1111 \\ \hline 11000 \end{array}$$

**Full adder:** A full adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs and two outputs. Two of the input variables, denoted by  $x$  and  $y$  represent the two significant bits to be added. The third input  $z$  represents the carry from the previous lower significant position. The two outputs are designed by the symbols  $S$  and  $C$ . The binary  $S$  gives the value of the least significant bit of the sum. The binary variable  $C$  gives the output carry.

The truth table of the full adder is as follows:

$x$	$y$	$z$	$C$	$S$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0

1 1 1 1 1

### A four bit full adder:

A binary parallel adder is a digital function that produces the arithmetic sum of two binary numbers in parallel. It consists of full adders connected in cascade, with the output carry from one full adder connected to the input of the next full adder.

### IC: 7486(XOR) 7483(4bit parallel adder):

To implement addition and subtraction together:

1.  $B_1 \text{ xor } C_0$ ,  $B_2 \text{ xor } C_0$ ,  $B_3 \text{ xor } C_0$  and  $B_4 \text{ xor } C_0$
2. Connect output from step 1 to the input of 7483 IC's B inputs.
3. Keep  $C_0$  common for all steps
4. give  $C_0=0$  to perform addition,  $C_0=1$  to perform subtraction

We use XOR gate as it produces invert output of one operand when the other operand is equal to 1.

A	B	Output
1	0	1 (invert of B)
1	1	0 (invert of B)
0	1	1
0	0	0

