|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Expt. No:** | | **4** | **Full Adder and Full Subtractor** |
|  | | |
| **Date:** |  | |
|  | | | |

**Aim:**  To design and implement Full Adder and Full Subtractor Circuits.

**SOFTWARE TOOLS / OTHER REQUIREMENTS:**

1. Multisim Simulator
2. Logic Gates (AND, NOT and EX-OR)

# Theory:

# full ADDER:

Adders/Subtractors are important in computers and also in other types of digital systems in which numerical data are processed. An understanding of the basic adder/subtractor operation is fundamental to the study of digital systems.

Figure (a) below shows the logic symbol of a full-adder. As seen from this figure, we find that the full-adder accepts three binary digits on its inputs (two new bits and one carry from the previous stage) and produces two digits on its outputs: a sum bit (S) and a carry bit (Cout). Fig (b) shows the truth table for the full-adder.



The full –adder also follows the same basic rules of binary addition as half-adder:



The Boolean expression for the sum output (S) can be obtained from the above truth table by summing and then simplifying the terms for which S=1. Thus, the sum is



Simillarly, adding up all the terms for which carry output () is 1 and simplifying will lead us to expression for output carry as



The equations for Sum and Carry can be easily implemented by using logic gates. From equation of Sum we find that to implement to full-adder’s sum output function, two 2-input Exclusive–OR gates can be used. The first Exclusive–OR gate generates the term , and the second has its inputs the output of the first Exclusive–OR gate and the input carry as shown in the Fig below. Similarly from equation of Carry we find that to implement the full- adder’s carry output function, three 2-input AND gates followed by a 3-input OR gate can be used. The complete circuit of a full adder is shown below.

****

**full subtractor:**

Fig. (*a*) below shows the logic symbol of a full-subtractor. As seen from this figure, we find that the full-subtractor accepts three inputs. Two input bits *A* and *B* and a borrow bit (*B*in). It has two outputs : (1) a difference output (D) and a borrow output (*B*out). Fig. (*b*) shows the truth table for the full-subtractor.



We observe that the full-subtractor also follows the basic rules of binary subtraction as half-subtractor:



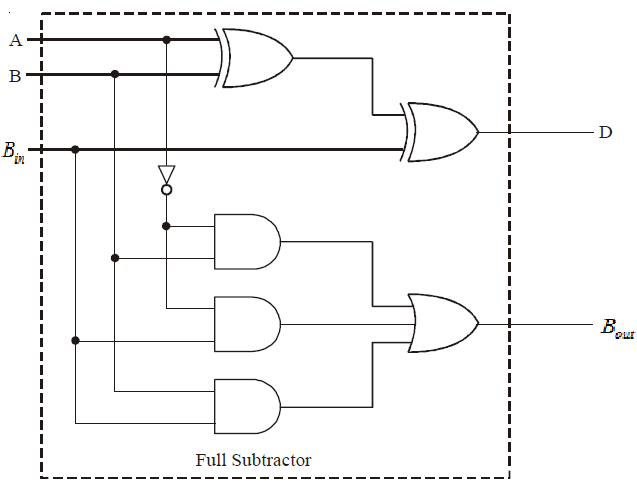
The Boolean expression for the difference bit (D) can be obtained by summing and simplifying all the input combinations from the truth table which have 1 in the corresponding difference column. The final simplified expression for difference is given by



and, the Boolean expression for the borrow bit,



From the above two expressions we find that to implement full-subtractor’s difference output function, two-2 input Exclusive–OR gates can be used. Similarly from equation of borrow we find that to implement the full-subtractor’s borrow output function, a NOT gate, three 2-input AND gates followed by a 3-input OR gate can be used. The complete circuit of a full-subtractor is shown below.



**FULL Adder: Circuit/connection diagrams (fROM mULTISIM)**

**FULL Adder: oUTPUT pLOTs/WAVEFORMS (from MULTISIM):**

**FULL subtractor: Circuit/connection diagrams (MULTISIM)**

**FULL subtractor: oUTPUT pLOTs/WAVEFORMS (MULTISIM)**

**Conclusions**