

**SHAID SMARAK COLLEGE**

Kirtipur, Kathmandu

*Lab no: 3 of Digital logics*

**Submitted by :-**

1st semester

Amir Maharjan

**Submitted to :-**

Himal Raj Gental

**LAB 3: HALF-ADDER AND FULL-ADDER**

**Objective:**

* To learn about half adders and full adders.
* To implement the half adders and full adders.

**Discussion:**

Digital computers perform a variety of information processing tasks. Among the basic functions encountered are the various arithmetic operations. The most basic arithmetic operation, no dough, is the addition of two binary digits.

* **Half-adders**

A half adder is a type of adder, an electronic circuit that performs the addition of number. The half adder is able to add two single binary digits and provide the output plus a carry value. It has two inputs, called A and B, and two outputs S (Sum) and C (Carry).

The Truth table formulation of half-adders given below:

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **S** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

*Truth Table of Half adder*

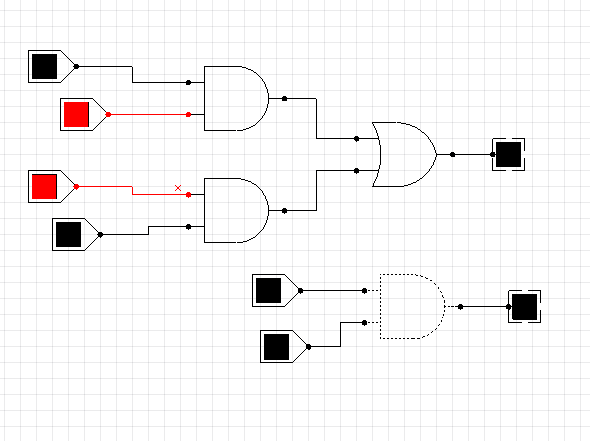
The simplified Boolean function for the two outputs can be obtained directly form the truth table. The simplified Sum of Products (SOP) expression are: -

**S = X’Y + XY’**

**C= XY**

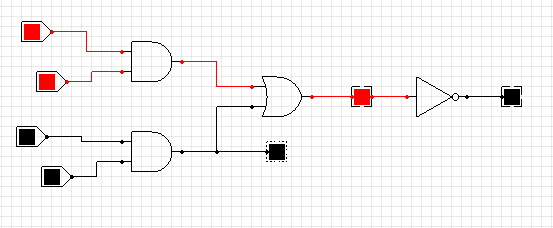
After obtaining the simplified Sum of Products (SOP) expression all we have to do is implement the simplified expression.

* **Implementation**

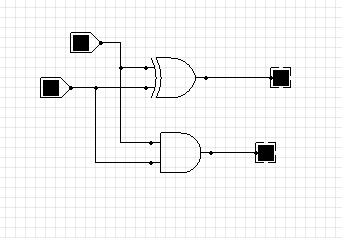


*Circuit diagram of S = X’Y + XY, C= XY*

*#* Other realizations and implementation of half-adders are: -



*Circuit diagram of S = (C + X’Y’), C = XY*

ࣙ

*Circuit diagram of S = XOR gate, C = XY*

* **Full adders**

Full adder is the adder which adds three inputs and produces two outputs. A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to another.

The three inputs are A, B and Z, and the two outputs are as in the half adder S (sum) and C (carry).

The truth table formulation of the full adder is given below: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **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 |

*Truth table of Full adder*

The input-output logical relationship of the full adder circuit may be expressed in tow Boolean function, each one output variable. Each output Boolean function requires a unique map for its simplification.

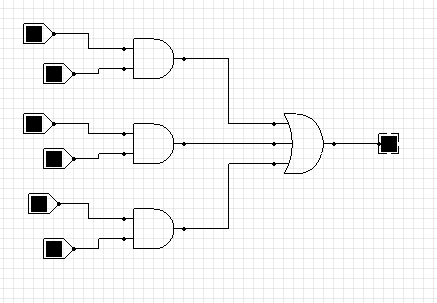
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A BZ** | **00** | **01** | **11** | **10** |
| **0** |  | 1 |  | 1 |
| **1** | 1 |  | 1 |  |

*S = X’Y’Z + X’YZ’ + XY’Z’ + XYZ*

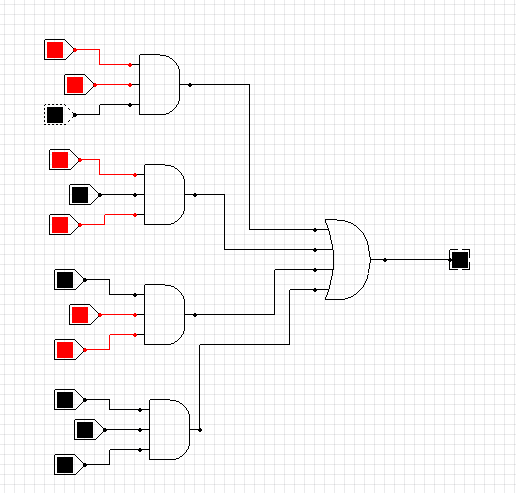
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A BZ** | **00** | **01** | **11** | **10** |
| **0** |  |  | 1 |  |
| **1** |  | 1 | 1 | 1 |

*C = XY + XZ + YZ*

* **Implementation**

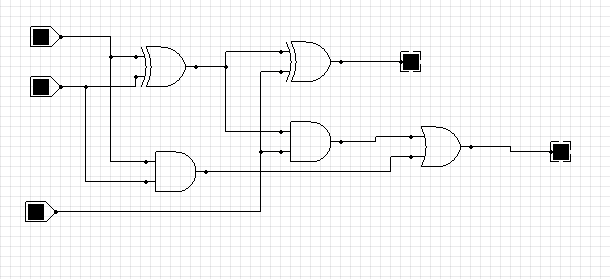


*C = XY + XZ + YZ*



*S = X’Y’Z + X’YZ’ + XY’Z’ + XYZ*

A full adder can be implemented with two half adders and one OR gate. An example is given below:



Here, the S (sum) output from the second half-adder is the exclusive-OR (XOR) of Z and the output of the first half adder giving;

S = XY’Z’ + X’YZ’ + XYZ + X’Y’Z

C = XY’Z + X’YZ + XY