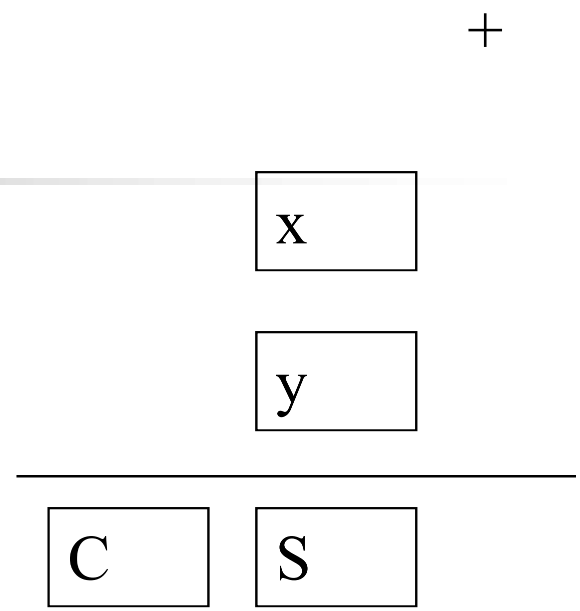
Unit 2 Part I
Combinational Circuits

HALF-ADDER



• **Half-adder**: Performs the most basic digital arithmetic operation, that is, the addition of two binary numbers.

•The half-adder requires two outputs because the sum 1 + 1 is binary 10. The two inputs are:

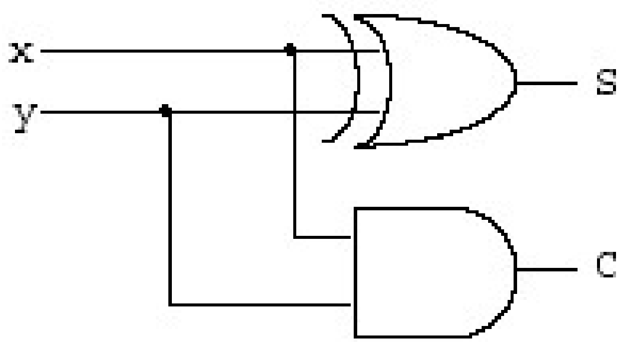
called S (for sum) and C (for carry out).

From the truth table write the Boolean function outputs for the sum S and the carry out C:

S=x`y +xy` (Exclusive OR)

C=xy (AND)

(Logic diagram)



(Truth table for half-adder)			
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

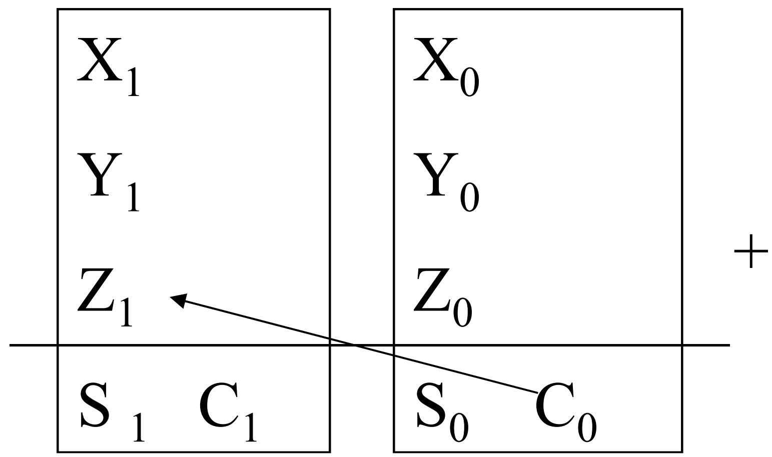
Here is a proof of the Exclusive OR identity using truth table.

(Proof by truth table)

x	y	x'y	xy'	x'y+xy'
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

FULL-ADDER

- To implement an arithmetic adder for multiple-bit inputs, we need to treat the carry out from the lower bit as **a third input** (it becomes carry in for the current bit) in addition to the two input bits at the current bit position.



Full- Adder

It adds 3-bits, it has 3-inputs and 2-outputs

We will use x, y and z for inputs and s for sum and c for carry are the tw
outputs.

The truth table

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

Full Adder

$$S = x`y`z + x`yz` + xy`z` + xyz$$

<u>k-map for s</u>

x \ y z	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Full Adder

Inputs			Outputs	
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

$$S = x \oplus y \oplus z$$

K-map on C

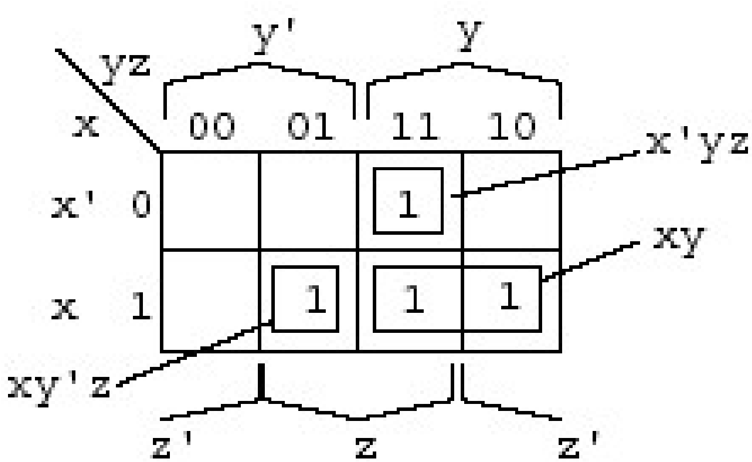
C=yz+xz+xy

But we would like to use the previous logic gate XOR

x \ y z	00	01	11	10
0	**0**	**0**	**1**	**0**
1	**0**	**1**	**1**	**1**

K-map on C

We can write C = x\`yz + xy\`z + xyz + xyz\`
= z(x\`y + xy\`) + xy(z+z\`)
= z (x ⊕ y) + xy (x ⊕ y) is already used for the sum S

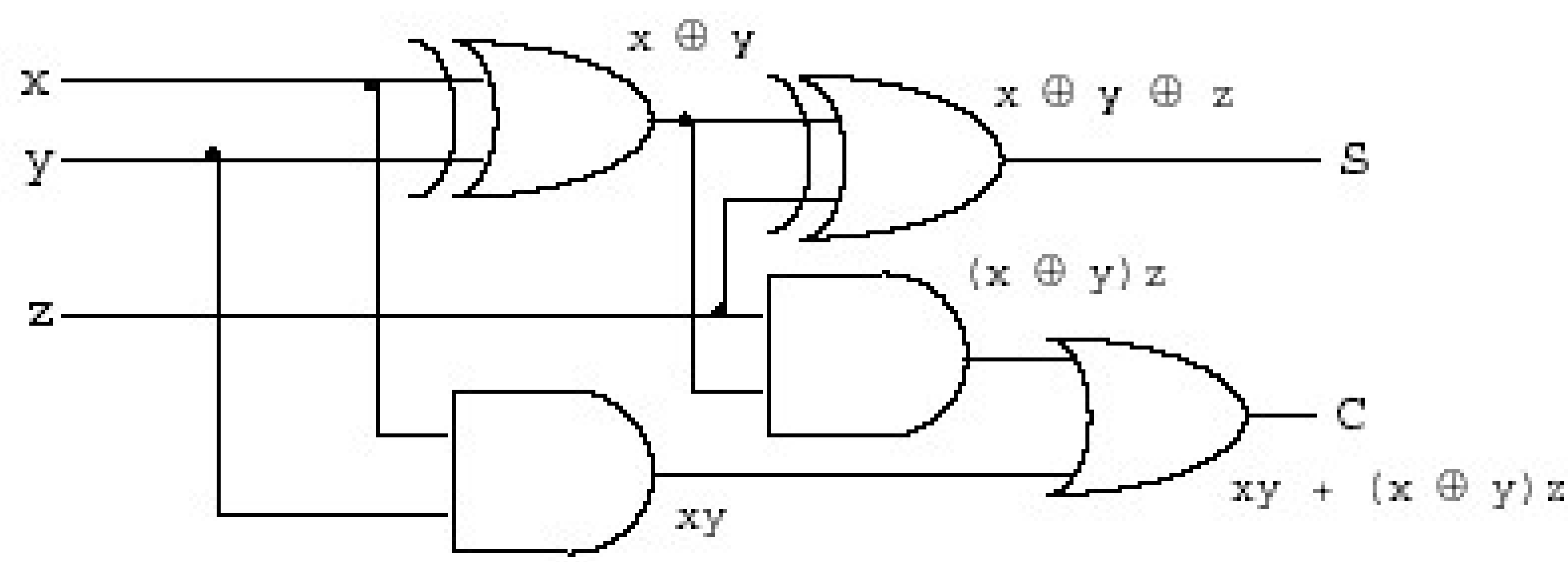


Full Adder

Putting them together we get:

$S = x \oplus y \oplus z$

$C = z\,(x \oplus y) + xy$



The logic diagram for the full adder