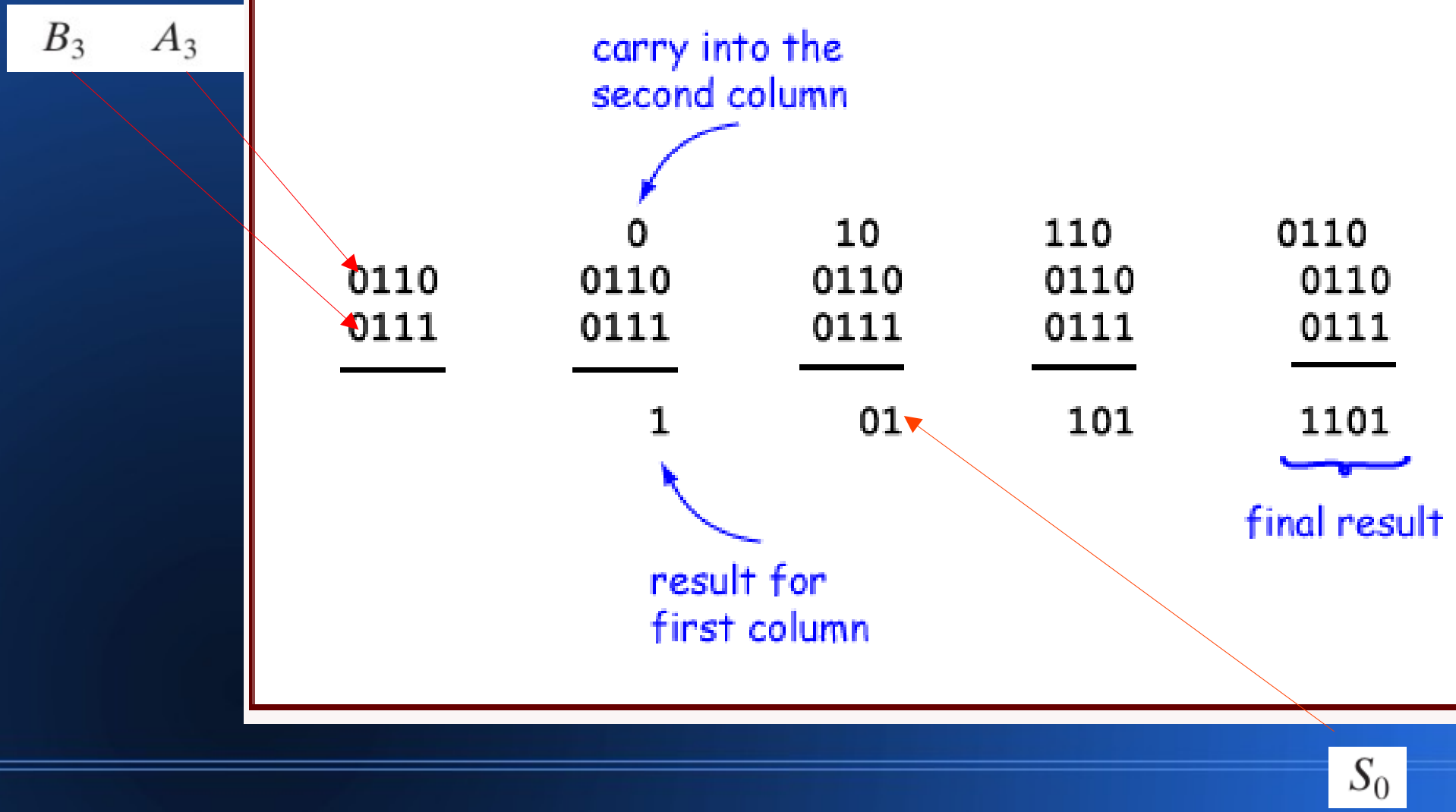
The most basic arithmetic operation is the addition of two binary digits.

This simple addition consists of four possible elementary operations: 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, and 1 + 1 = 10.

The first three operations produce a sum of one digit, but when both augend and addend bits are equal to 1, the binary sum consists of two digits.

The higher significant bit of this result is called a carry.

Binary Addition Example

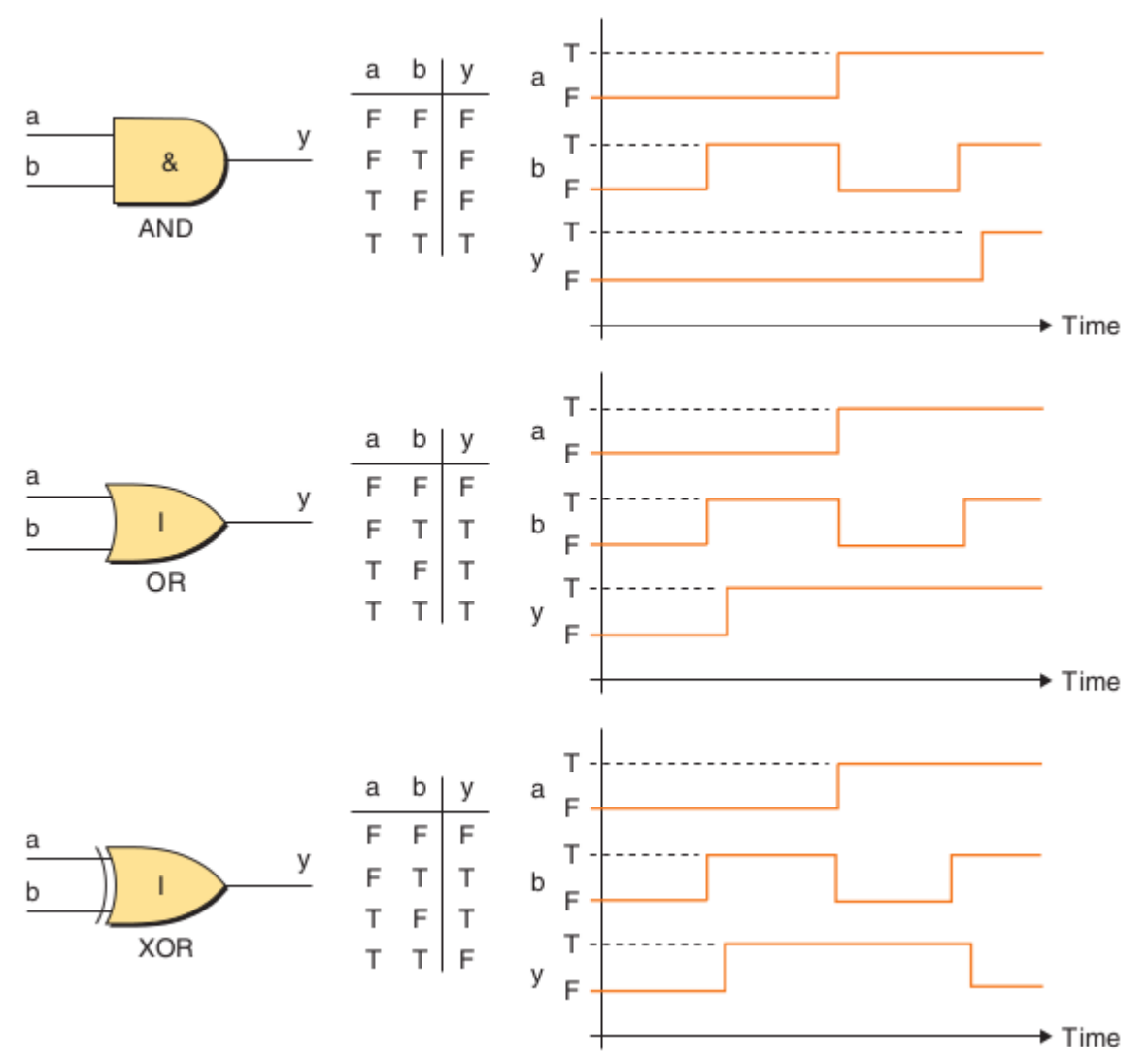


You'll rarely hear anyone except a maths teacher use either of these terms.

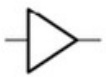
Addend is a math term for any number that's added to another.
If you add 10 to 15, then 10 is the addend. You can also call an addend a summand.

If you're adding 7 to 112, the 7 is an addend and the second number is officially called an augend

AND, OR, and XOR functions.

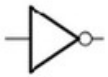


YES



INPUT	OUTPUT
A	
0	0
1	1

NOT



INPUT	OUTPUT
A	
0	1
1	0

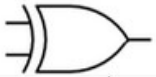
AND

INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

XOR

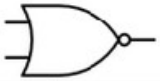


INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

NAND

INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

NOR



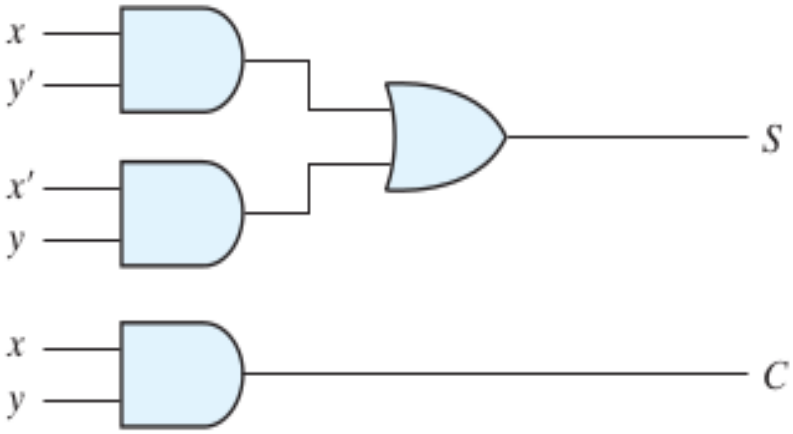
INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR

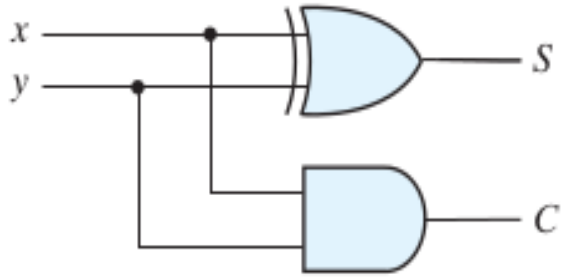
INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	1

Implementation of half adder

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



(a) $S = xy' + x'y$
$C = xy$

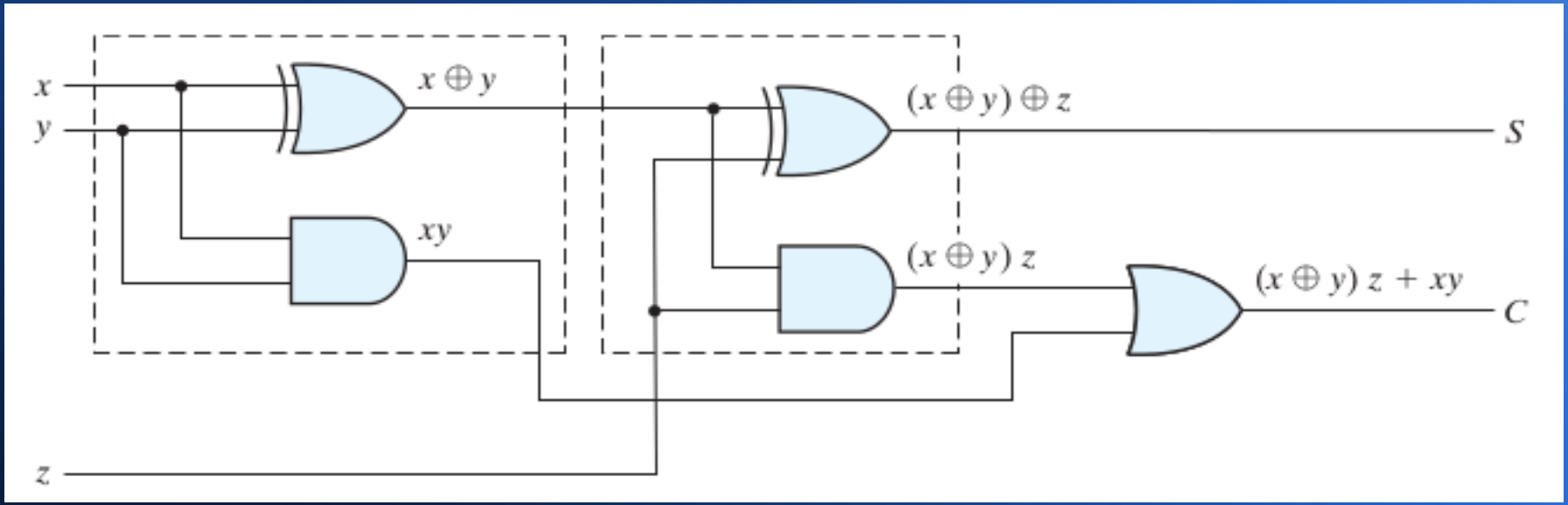
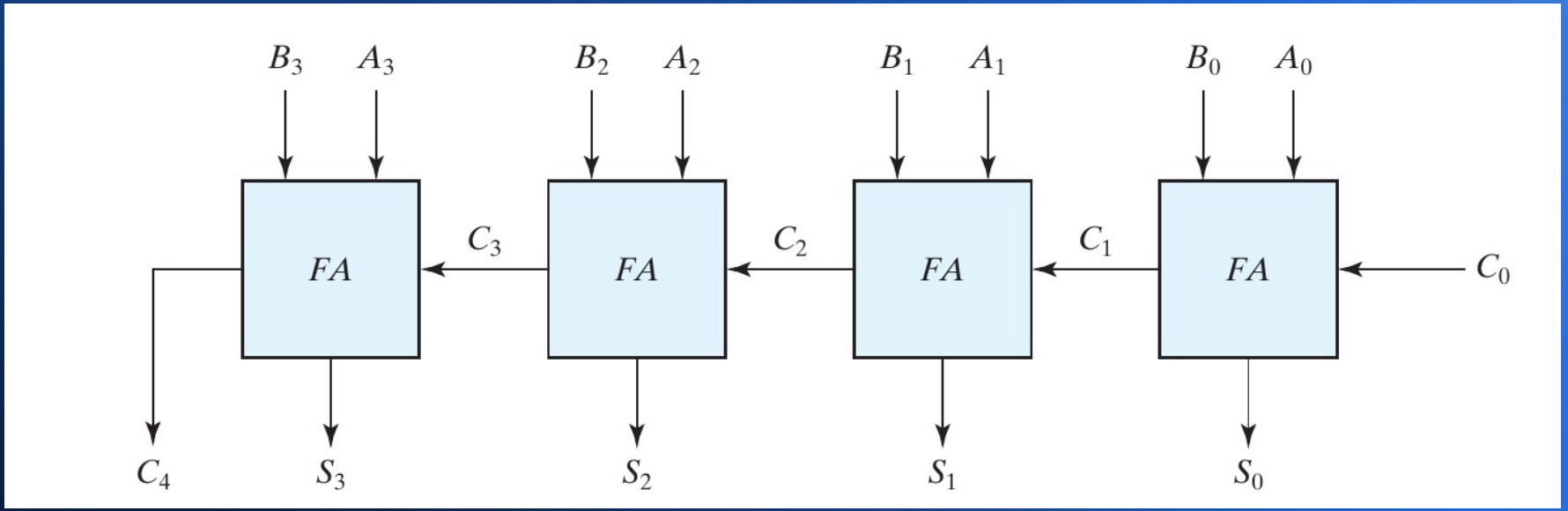


(b) $S = x \oplus y$
$C = xy$

Full Adder

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

Implementation of full adder with two half adders and an OR gate



Four-bit adder