Lecture 25

<u>Example</u> We construct a circuit for adding two four-bit binary numbers. What we did in the last lecture was proof of concept. Here we shall use another approach (modular) to solving the problem.

In the book you will find a discussion about representing numbers in binary. This you should read if you do not already know how to do this.

To avoid confusion with the Boolean operation '+' I will use the word "add" to mean binary addition

add	0	1
0	00	01
1	01	10

10 means

$1 \times 2^1 + 0 \times 2^0$

$= \underline{\underline{2}}$

[I have added zeros in front so that all outputs have two bits. In ordinary arithmetic they would of course be omitted.]

We shall first construct a circuit for adding two 1-bit binary numbers called a <u>half-adder</u>.

x	y	$Carry(x,y)$	$sum(x,y)$
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	0
inputs		outputs	

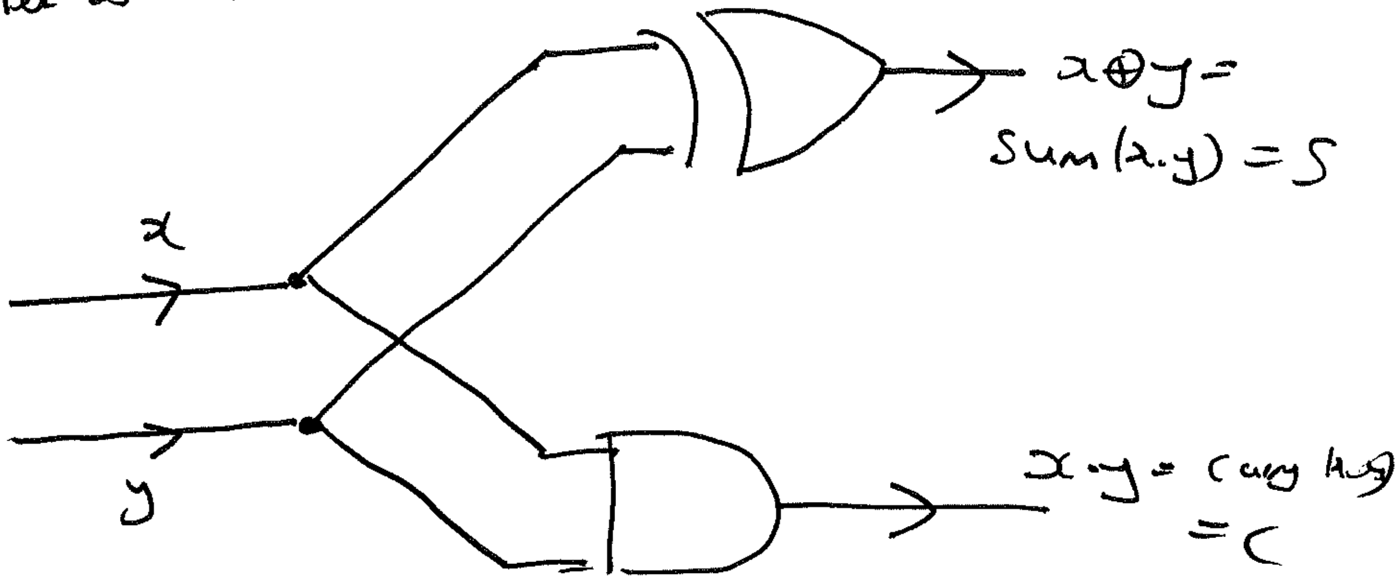
Where

x <u>add</u> y = $Carry(x,y)\ sum(x,y)$

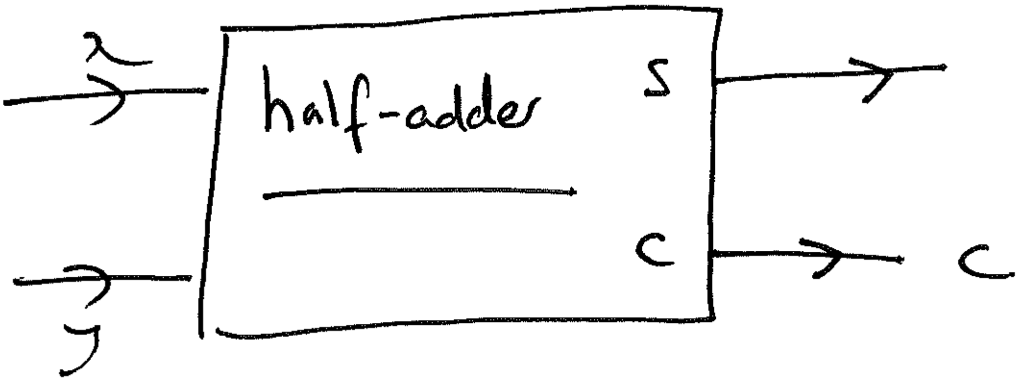
Observe that

$$\text{Carry}(x,y) = x \cdot y$$
$$\text{Sum}(x,y) = x \oplus y$$

Here is a circuit that will accomplish this



We represent the above circuit by the following block diagram



Our next circuit is called a <u>full-adder</u>

It adds three 1-bit binary numbers.

You will see why we need this table.

x	y	z	carry(x,y,z)	sum(x,y,z)
1	1	1	1	1
1	1	0	1	0
1	0	1	1	0
1	0	0	0	1
0	1	1	1	0
0	1	0	0	1
0	0	1	0	1
0	0	0	0	0

$$11 = 1 \times 2^1 + 1 \times 2^0 = 2+1 = 3$$

We shall construct a full-adder by using half-adders.

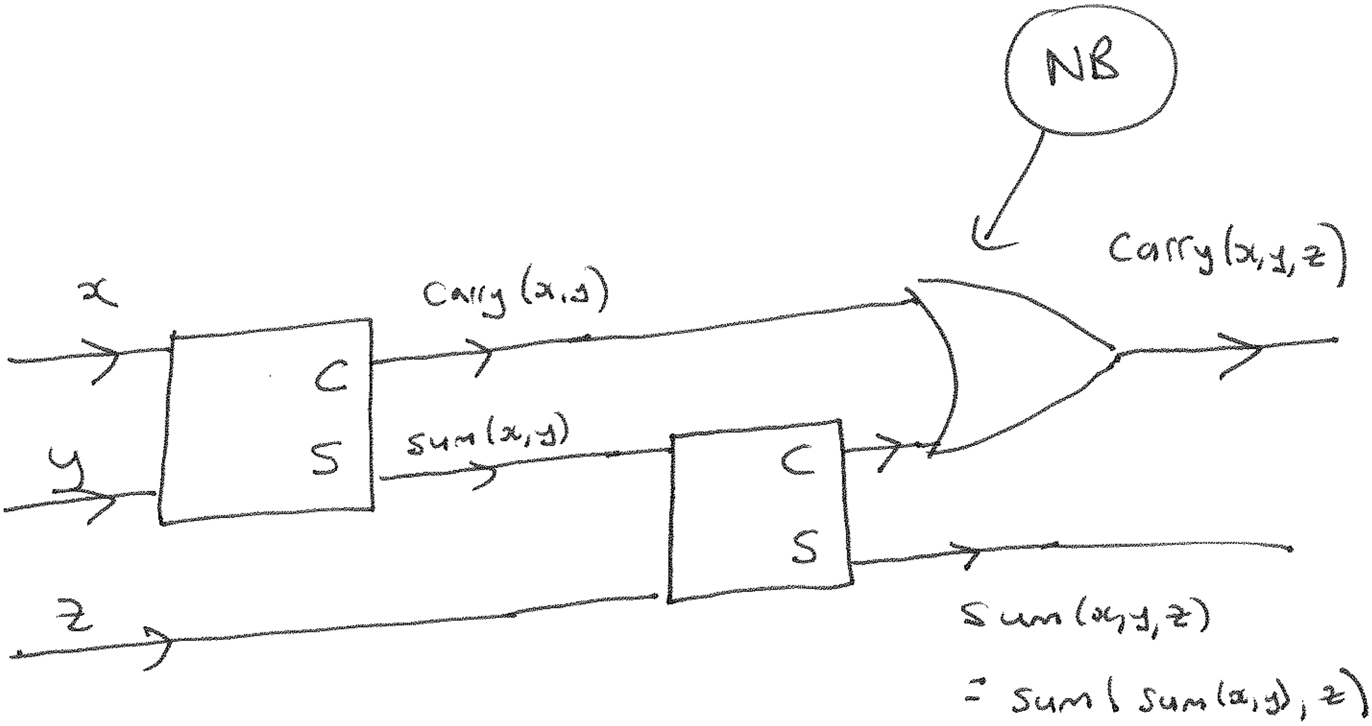
x add y add z = $(x$ add $y)$ add z

x add y = carry(x,y) sum(x,y)

So, $(x$ add $y)$ add z =

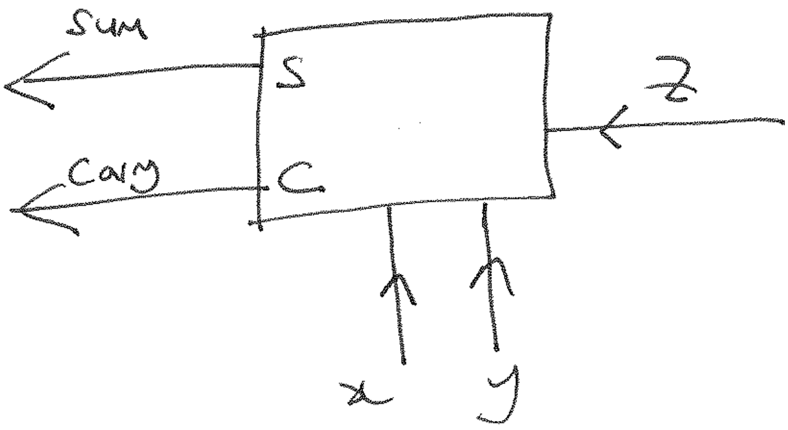
Carry(x,y) add	sum(x,y) z	
?	Sum(sum(x,y), z)	
carry(sum(x,y), z)		

You can check that carry(x,y) and carry(sum(x,y), z) can never both be 1 || NB



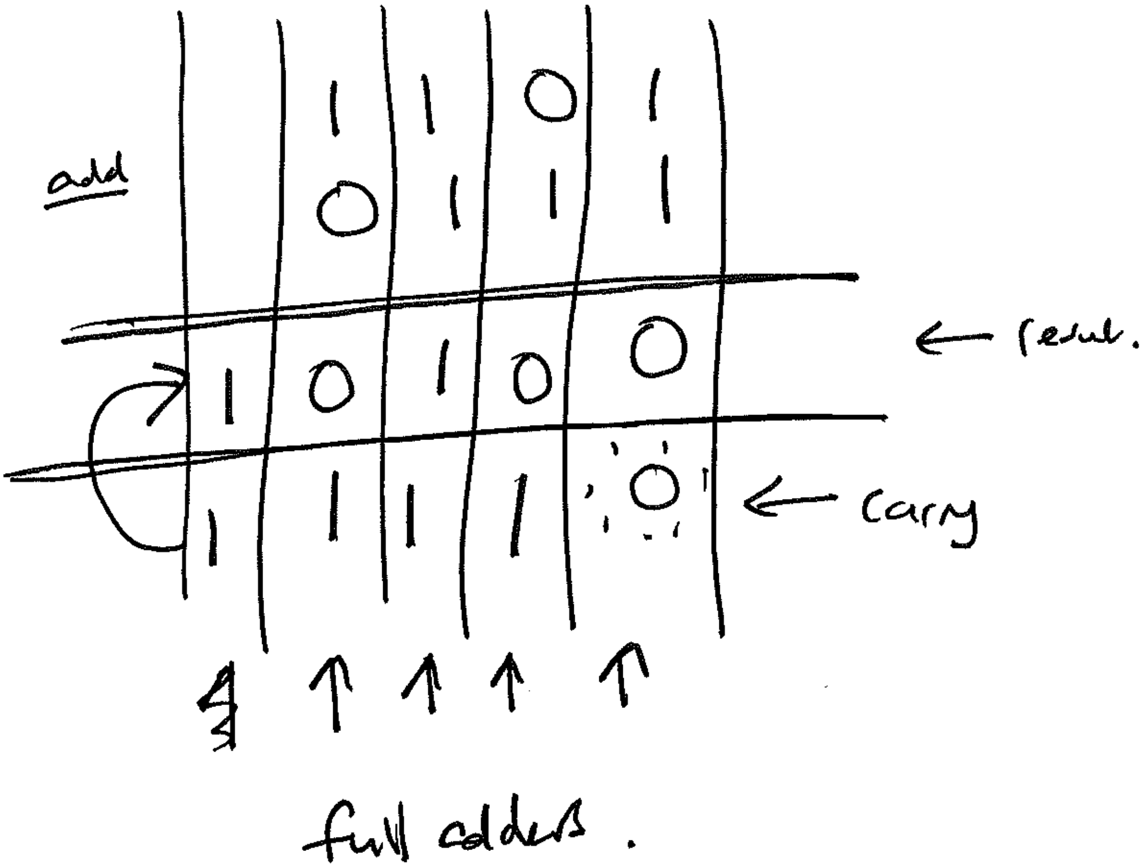
You can check this works
by constructing an input/output table

Block diagram of full-adder

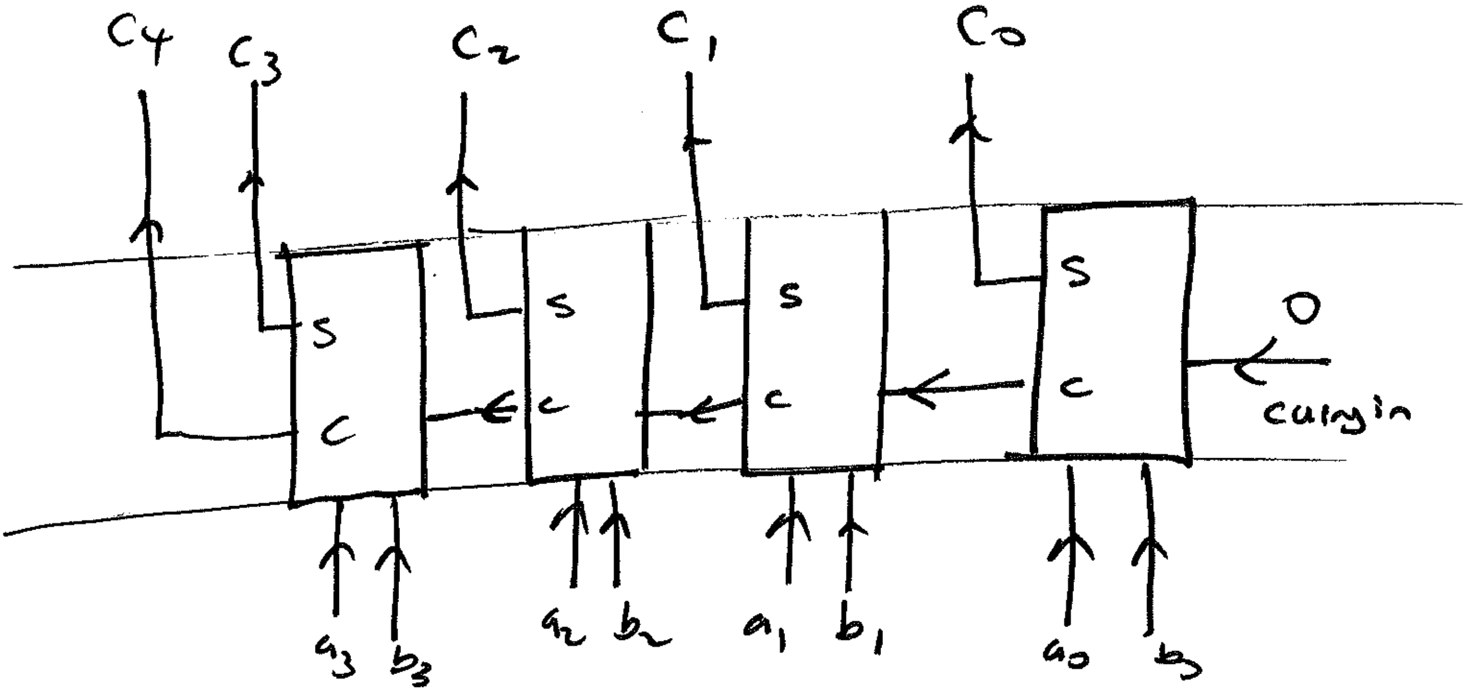


Now we can show how to build a circuit that will add two 4-bit binary numbers.

Example



		a_3	a_2	a_1	a_0
add		b_3	b_2	b_1	b_0
	c_4	c_3	c_2	c_1	c_0
					0	← carries



Need four full-adders in series.