
EE1001 4 bit add theory

By Dr. K. F. Tsang

Binary Addition

■ *Review of Binary addition*

■ Use binary addition to compute 5+7

Write 5 in binary as: 101

Write 7 in binary as: 111

We have:

Add column by column!

		1	0	1	
+	1	1	1	1	
	1	1	0	0	$1 + 1 = 2 = 10$
					$0 + 1 + 1 = 2 = 10$
					$1 + 1 + 1 = 3 = 11$

0 is the sum, 1 is the carry

0 is the sum, 1 is the carry

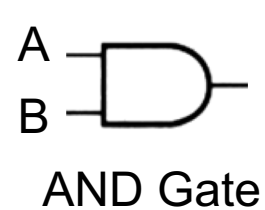
1 is the sum, 1 is the carry

Therefore, the answer is 1100(12)

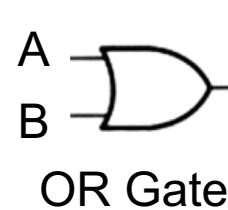
In summary: 1 + 1 creates a carry.

Recall the Truth Tables in Engineering

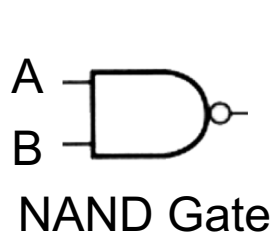
Convert **Truth/False (T/F)** to **binary number (1/0)**



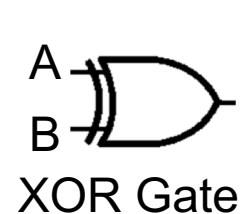
Input		Output
A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0



Input		Output
A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0



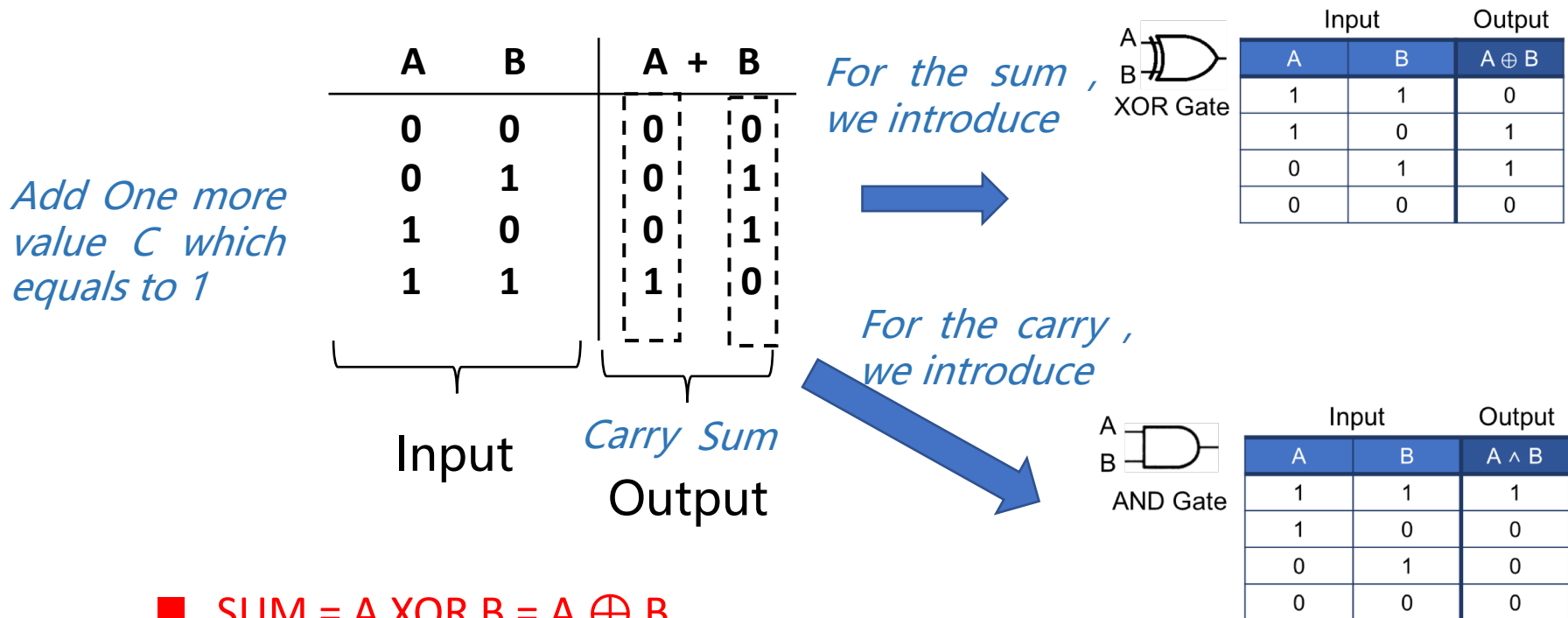
Input		Output
A	B	$\sim(A \wedge B)$
1	1	0
1	0	1
0	1	1
0	0	1



Input		Output
A	B	$A \oplus B$
1	1	0
1	0	1
0	1	1
0	0	0

logic circuits

- This problem can be solved with *the logic circuits*
- Consider we have two terms A and B, and compute A+B. Both of A and B can only take the value 1 or 0.

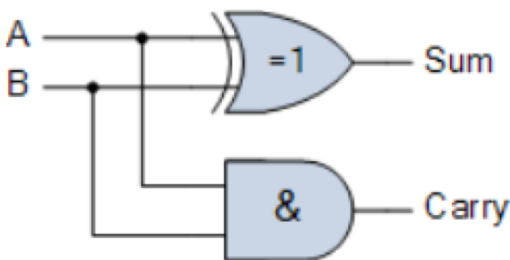


■ $SUM = A \text{ XOR } B = A \oplus B$

■ $CARRY = A \text{ AND } B = A.B$

Half Adder Circuit

- A **half adder** is a logical circuit that performs an addition operation on two binary digits. The half adder produces a **sum** and a **carry** value which are both binary digits.

Symbol	Truth Table			
	B	A	SUM	CARRY
	0	0	0	0
	0	1	1	0
	1	0	1	0
	1	1	0	1

Full Adder Circuit

- Apart from A and B, let's consider another C which equals to 1.
- C also can be regarded as a carrier from the previous stage.

C	A	B	A + B + C
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Carry-in	Input	<i>Carry-out</i>	<i>Sum</i>
----------	-------	------------------	------------

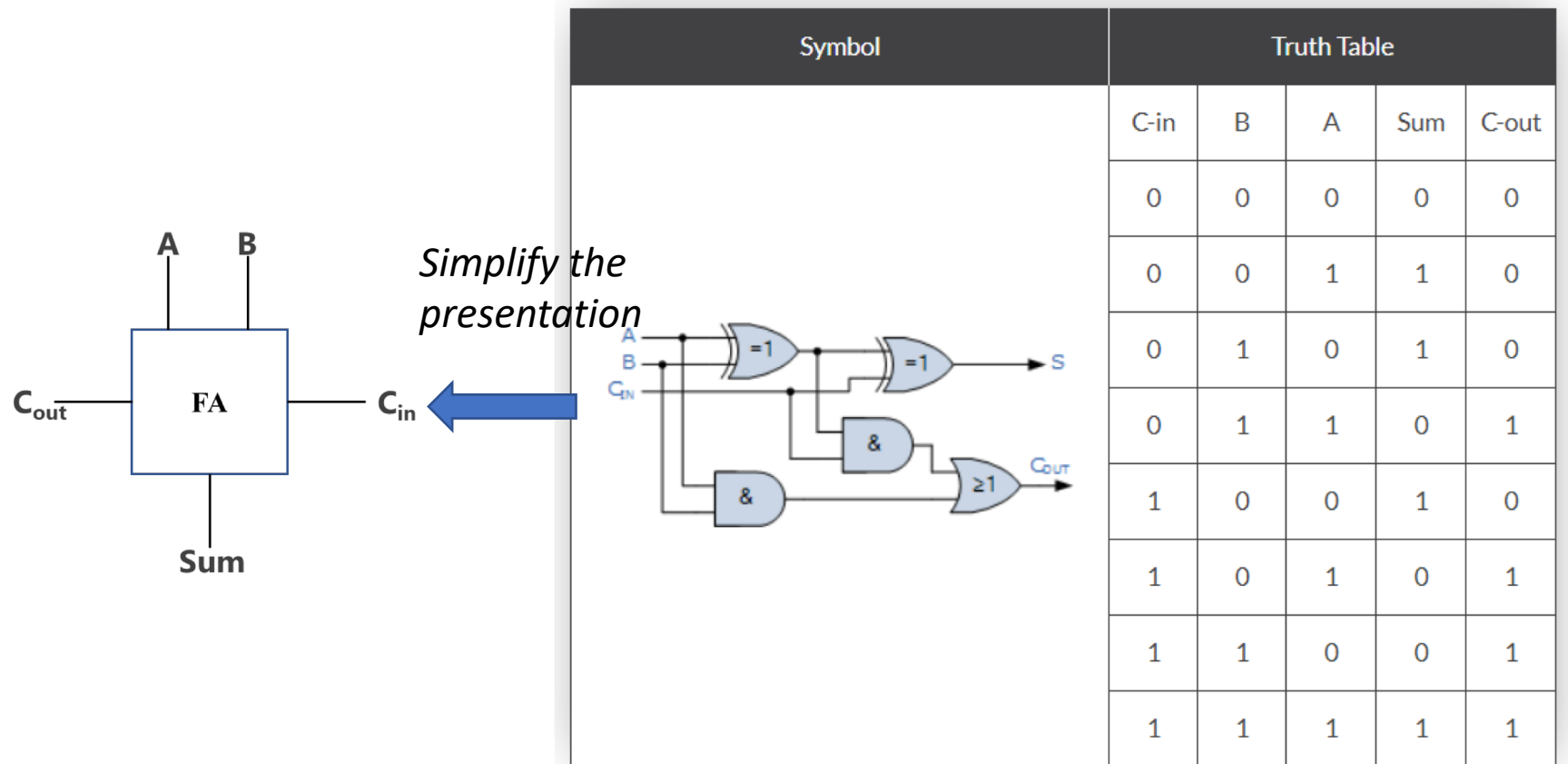
- Compared with the result of $A+B+C$ with $A+B$, it can be concluded:

■ $SUM = (A \text{ XOR } B) \text{ XOR } C = (A \oplus B) \oplus C = A \oplus B \oplus C$

■ CARRY-OUT = A AND B OR $C_{in}(A \text{ XOR } B) = A.B + C_{in}(A \oplus B)$

Full Adder Circuit

- The **Full Adder** is an adder has *three* inputs. The same two single bit data inputs A and B as before plus an additional Carry-in (C_{in}) input to *receive the carry from a previous stage*.

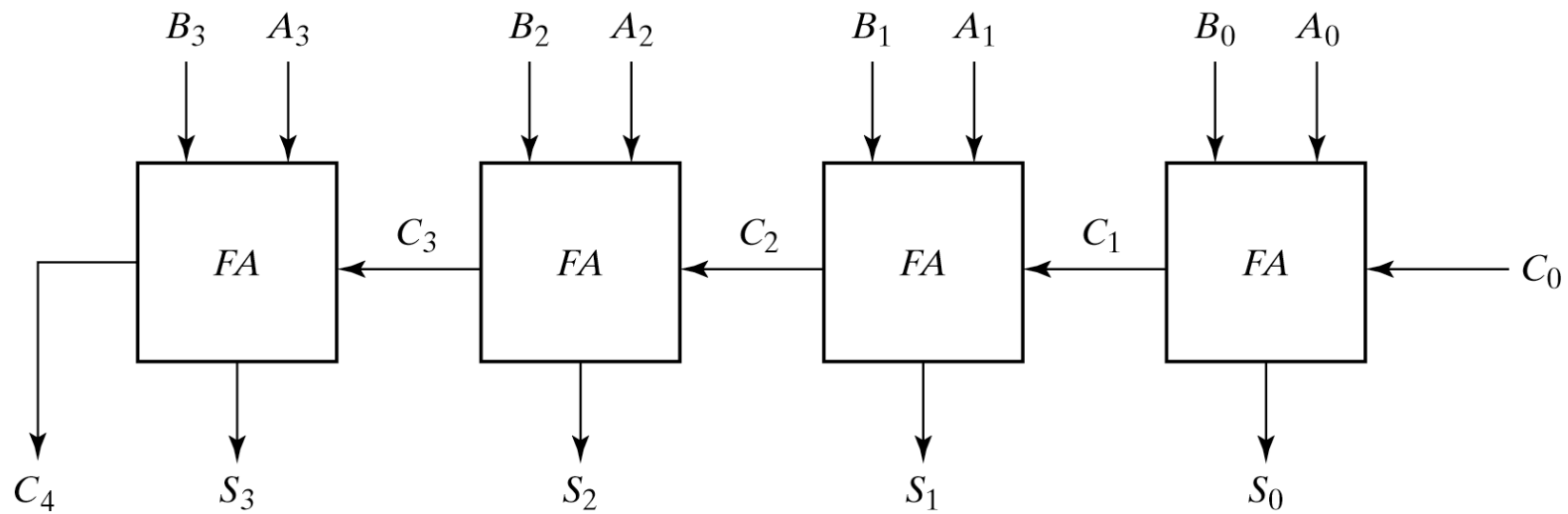


4-bit Adder Implementation

- Let's consider two 4-bit binary numbers A and B as inputs to the Digital Circuit for the operation with digits:

A_0, A_1, A_2, A_3

B_0, B_1, B_2, B_3



- n-bit binary adder requires n full adders.
- The sum of each stage will be the output of the stage;
- The carry-out in this stage will be the carry-in in the next stage;
- The carry-out of the final stage will also be the output;

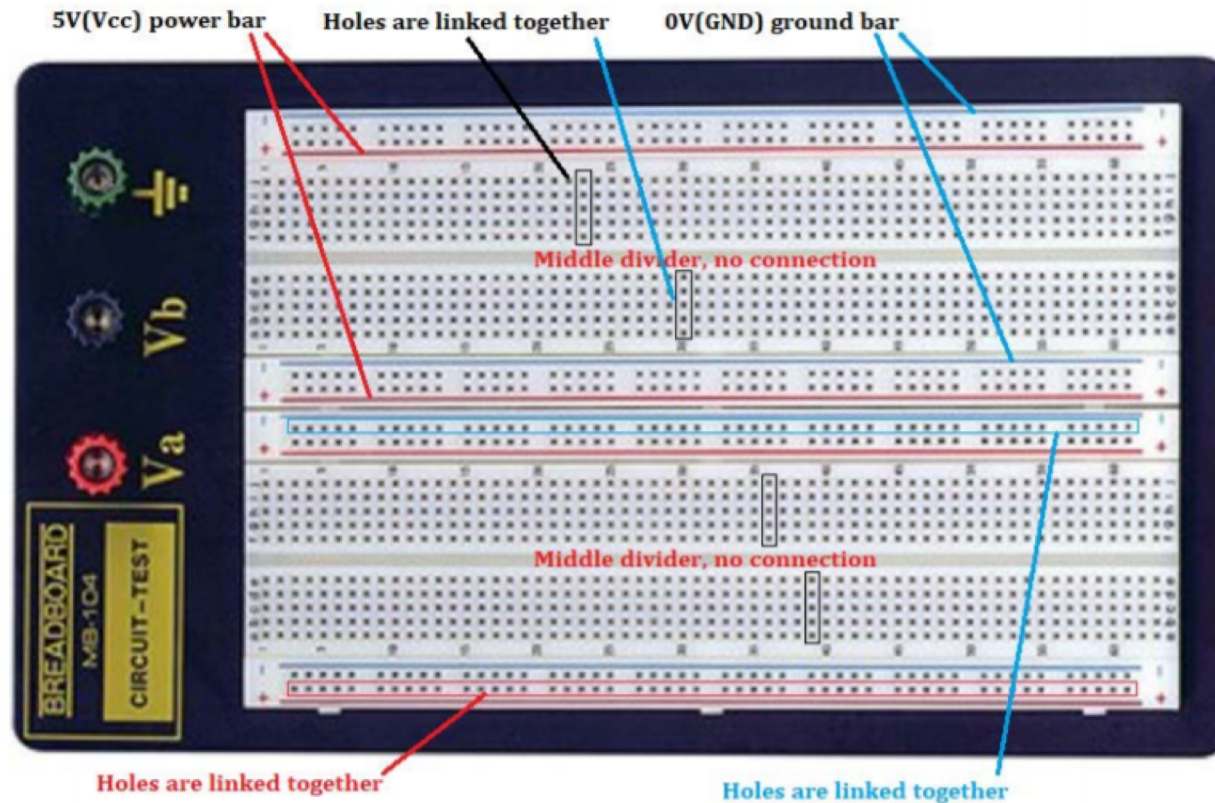
Implementation

■ Components provided:

Breadboard		1
2-input AND gate	74LS08	2
3-input AND gate	74LS11	1
2-input OR gate	74LS32	1
2-input XOR gate	74LS86	2
LEDs (Red x 4, white x4)		8
4-ways DIP switch		2
470 Ω resistor (color ring: yellow, purple, black, black)		10
1k Ω resistor (color ring: brown, black, black, brown)		10
5V USB wire (option: lab at home)		1
Breadboard red wire (option: lab at home)		0.5M
Breadboard black wire (option: lab at home)		0.5M
Breadboard yellow wire (option: lab at home)		2M
Breadboard green wire (option: lab at home)		2M

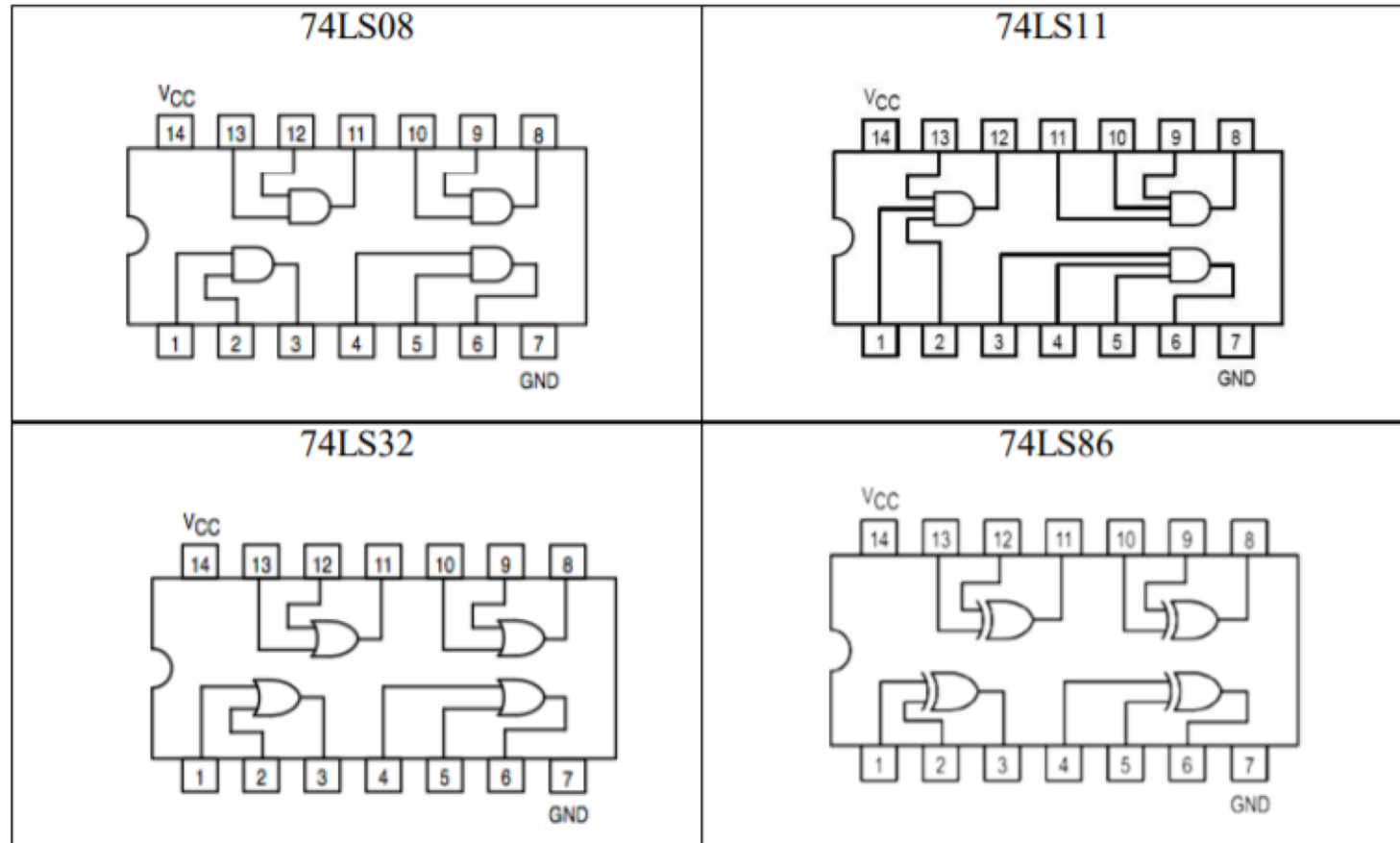
Introduction to bread board

■ Equipment provided:



Introduction to Chipset

- Each Chipset contains 4 Gates



Before implementing on the bread board, please read carefully the Laboratory Manual to make sure you are safe and hopefully also avoid damaging equipments.

Tutorial

■ Compute $6 + 7$ using logic circuits.

- (1) How many adders are needed?
- (2) Which adder should be adopted? Half or full?
- (3) What is the input and output of each adder?
- (4) Draw the logic circuits.

Solution:

- (1) The first step is writing the number 6 and 7 in binary:

$$6 = 1 \ 1 \ 0$$

$$7 = 1 \ 1 \ 1$$

There are three digital numbers for each term, therefore, 3 adders will be needed.

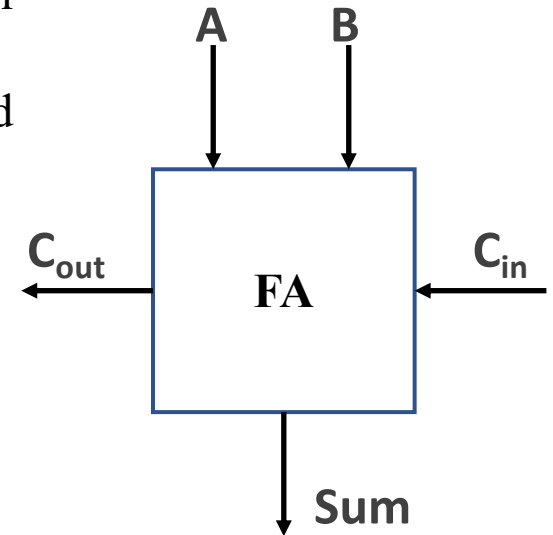
- (2) Use A0, A1, A2 to represent each number in 6, we have

$$\begin{array}{rcl} 6 = & \boxed{1} & \boxed{1} & \boxed{0} \\ & A2 & A1 & A0 \\ 7 = & \boxed{1} & \boxed{1} & \boxed{1} \\ & B2 & B1 & B0 \end{array}$$

There is existing $1+1$, where a carry will be generated, therefore, full adder should be used.

(3) For a full adder, it has three inputs: two digital binary input numbers and one carry from the previous stage:

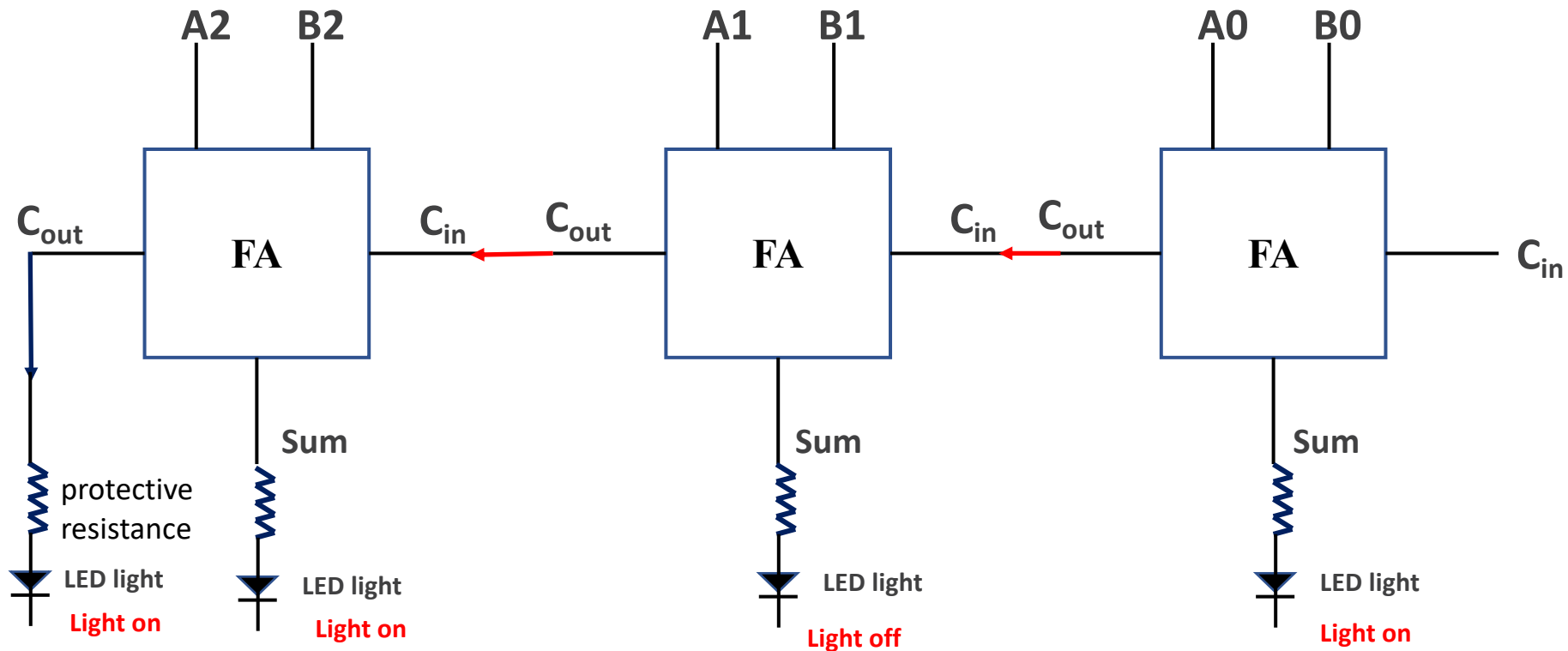
- The input of the first adder is $A_0=0$, $B_0=1$, and the initial carry-in $C_{in}0=0$;
Since $1+0=1$, the output of the first adder is the $sum_0=1$; and carry-out $C_{out}0=0$;
- For the input of the second adder is $A_1=1$, $B_1=1$, and the carry-in $C_{in}1$.
Noted that the carry-in $C_{in}1$ is the carry-out in the previous stage $C_{out}0$, thus, $C_{in}1=0$;
Since $0+1+1=10$, the output of the second adder is the $sum_1=0$; and carry-out $C_{out}1=1$;
- For the input of the third adder is $A_2=1$, $B_2=1$, and the carry-in $C_{in}2$.
Noted that the carry-in $C_{in}2$ is the carry-out in the previous stage $C_{out}1$, thus, $C_{in}2=1$;
Since $1+1+1=11$, the output of the second adder is the $sum_1=2$; and carry-out $C_{out}2=1$;
- The global output of all the last carry-out $C_{out}2$ and all the sums: sum_0 , sum_1 , sum_2 .
- Thus, the global output is : 1,1,0,1, which is also 13(6+7).



(3) To draw circuit, we first put all the adders and identify all the inputs and outputs:

We connect the C_{in} and C_{out} in adjacent adders

We connect LED light and protective resistance to each final output



If the output is 1, the light will on;

If the output is 0, the light will off, consider output is 1101

• - END -