①

Combinational logic : Binary Adder - Subtractor - I

objective :

1- Half adder
2- Full adder
3- Binary Adder
4- Binary subtractor
5- Binary Adder - subtractor.

1) Half adder : is a combinational circuit that performs the addition of two bits, this circuit needs two binary inputs and two binary outputs.

truth table

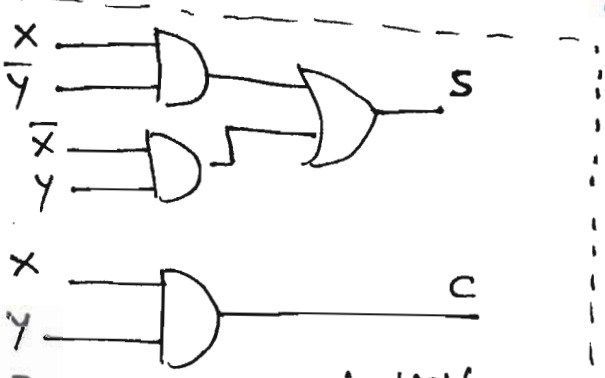
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

⇒

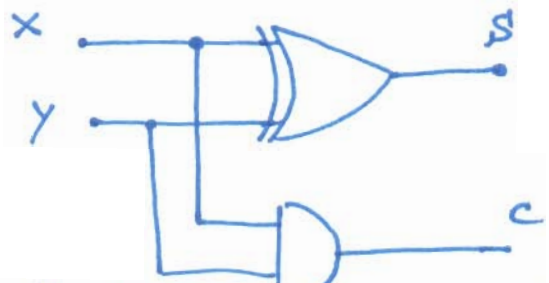
The simplified Boolean function from the table :

$$\begin{cases} S = \bar{X}Y + X\bar{Y} \leftarrow \text{sum} \\ C = XY \rightarrow \text{carry.} \end{cases} \rightarrow ① \text{("sum of products")}$$

$$\begin{cases} S = X \oplus Y \\ C = XY \end{cases} \rightarrow ② \text{ using exclusive-OR and AND gates}$$



Implementation of Half adder using equation (1)



Implementation of Half adder using equation ②

* The implementation of Half adder using exclusive-OR and an AND gate is used to show that two Half adders can be used to construct a full adder.

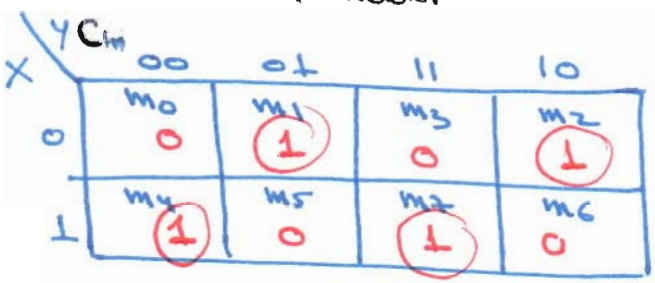
2) Full Adder : is a combinational circuit that performs the addition of three bits (two significant bits and previous carry).

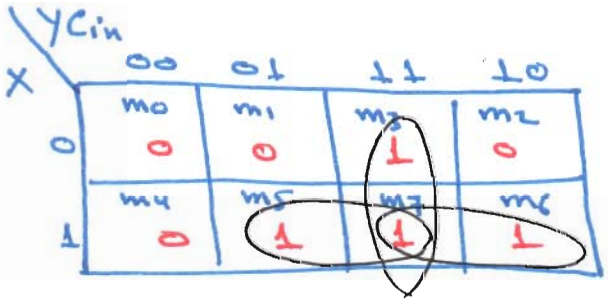
it consists of three inputs and two outputs, two inputs are the bits to be added, the third input represents the carry from the previous position.

Inputs			outputs	
X	Y	Cin	Cout	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 for the full adder

* The S output is equal to 1 when only one input is equal to 1 or when all three inputs are equal to 1.
* The C_{out} output has a carry 1 if two or three inputs are equal to 1.
* The Karnaugh maps and the simplified expression are shown in figures.



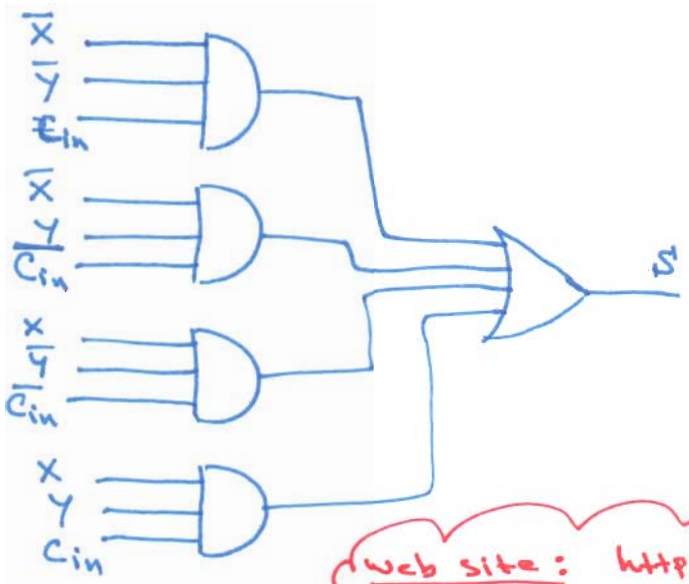


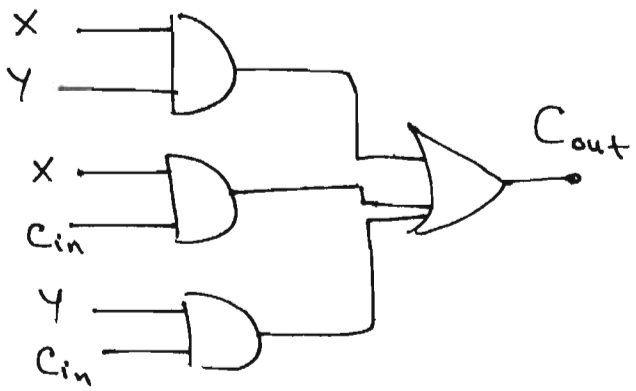
$$S = \bar{X}\bar{Y}C_{in} + \bar{X}Y\bar{C}_{in} + X\bar{Y}\bar{C}_{in} + XYC_{in}$$

$$C_{out} = XY + XC_{in} + YC_{in}$$

① sum of products.

The logic diagrams for the full adder implemented in sum-of-products form are the following:





web site: http://isweb.redwoods.edu/instruct/Calderwood/diglogic/full.htm

Reference

It can also be implemented using two half adders and one OR gate. (using XOR gates).

$$S = C_{in} \oplus (X \oplus Y)$$

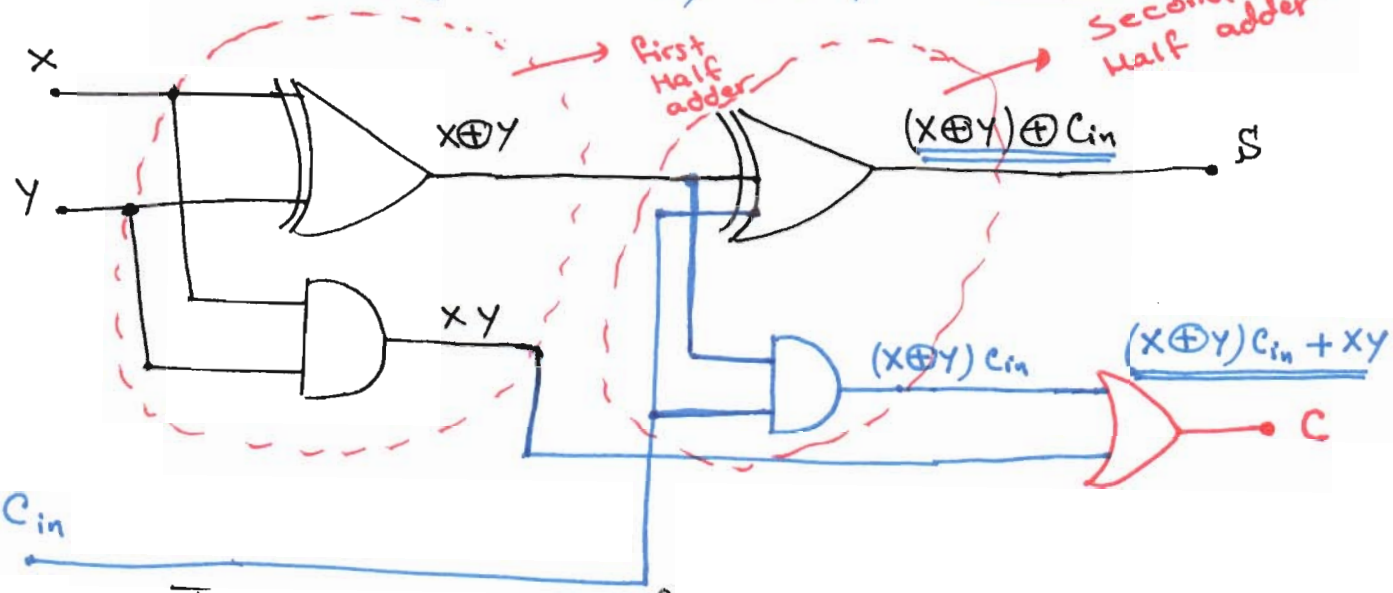
$$= \overline{C_{in}}(X\bar{Y} + \bar{X}Y) + C_{in}(X\bar{Y} + \bar{X}Y)$$

$$= \overline{C_{in}}(X\bar{Y} + \bar{X}Y) + C_{in}(XY + \bar{X}\bar{Y})$$

$$= X\bar{Y}\overline{C_{in}} + \bar{X}Y\overline{C_{in}} + XYC_{in} + \bar{X}\bar{Y}C_{in}$$

The Carry output is

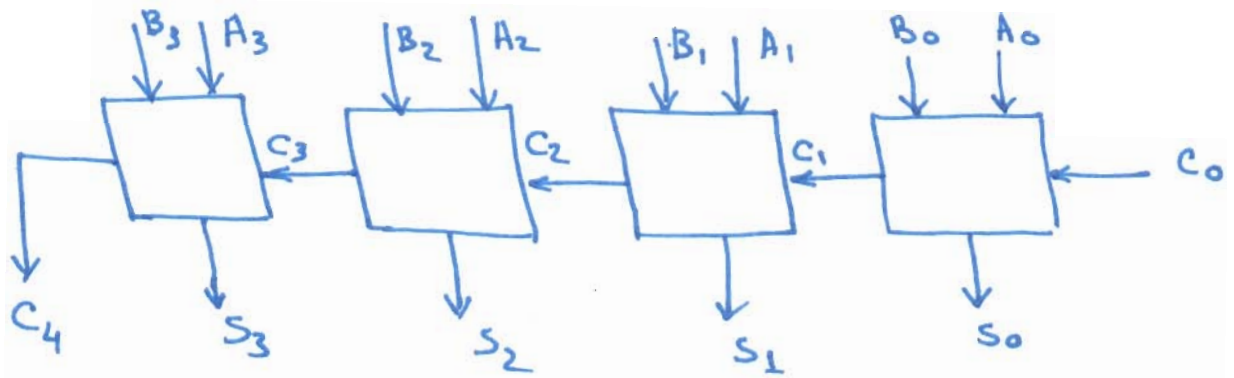
$$C = C_{in}(X\bar{Y} + \bar{X}Y) + XY = C_{in}\cdot(X \oplus Y) + XY$$



Implementation of full adder with two Half Adders and an OR gate.

3. Binary Adder (asynchronous ripple-carry adder)

A binary adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascade, with the output carry from each full adder connected to the input carry of the next full adder in the chain.



Four-bit adder (ripple carry adder)

* The input carry to the adder is C_0 and its ripples through the full adders to the output carry C_4.
* An n-bit adder requires n full adders.

example: A + B (A = 1011) and (B = 0011).

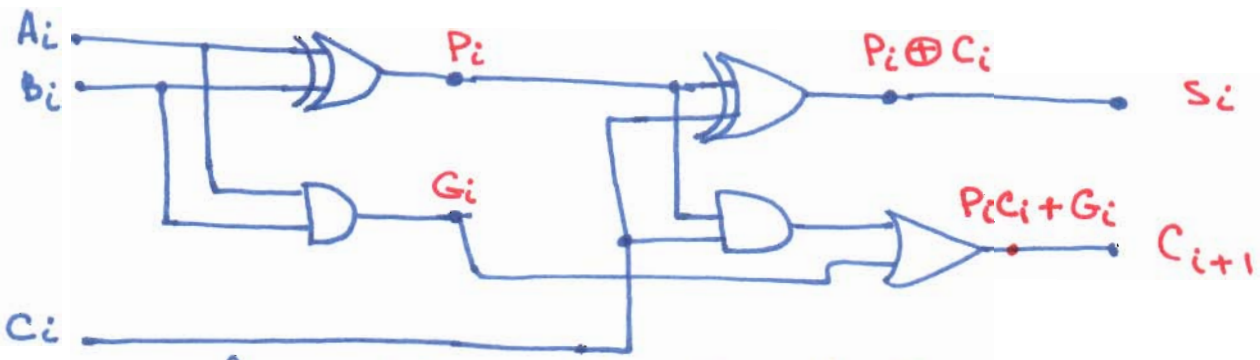
Subscript i	3	2	1	0	
Input Carry	0	1	1	0	C_i
A	1	0	1	1	A_i
B	0	0	1	1	B_i
Sum	1	1	1	0	S_i
Output Carry	0	0	1	1	C_{i+1}

$C_0 = 0$

* The four-bit adder is a typical example of a standard component. It can be used in many application involving arithmetic operations.

* Carry Propagation :

The addition of (A + B) binary numbers in parallel implies that all the bits of A and B are avaiable for computation at the same time.
As in any combinational circuit, the signal must propagate through the gates befor the correct output sum is avaiable.
The longest propagation delay time in an adder is the time it takes the carry to propagate through the full adders.



full adder with P and G

* The signal from the carry input C_i to the output carry C_{i+1} propagates through an AND gate and an OR gate, which equals 2 gate levels.

if there are 4 full adders in the adder, the output carry C_4 would have $2 \times 4 = 8$ gate levels, from C_0 to C_4, For an n-bit adder, $\Rightarrow$ **2n gate levels** for the carry to propagate from input to output.

* The **carry propagation time** is an important attribute of the adder because it limits the speed with which two numbers are added.

To reduce the carry propagation delay time:

1- employ faster gates with reduced delays.

2- employ the principle of **Carry lookahead logic**.

Proof: (using carry lookahead logic)

$$P_i = A_i \oplus B_i$$
$$G_i = A_i B_i$$

The output sum and carry are:

$$S_i = P_i \oplus C_i$$
$$C_{i+1} = G_i + P_i C_i$$

G_i - called a carry generate, and it produce a carry of 1 when both A_i and B_i are 1.

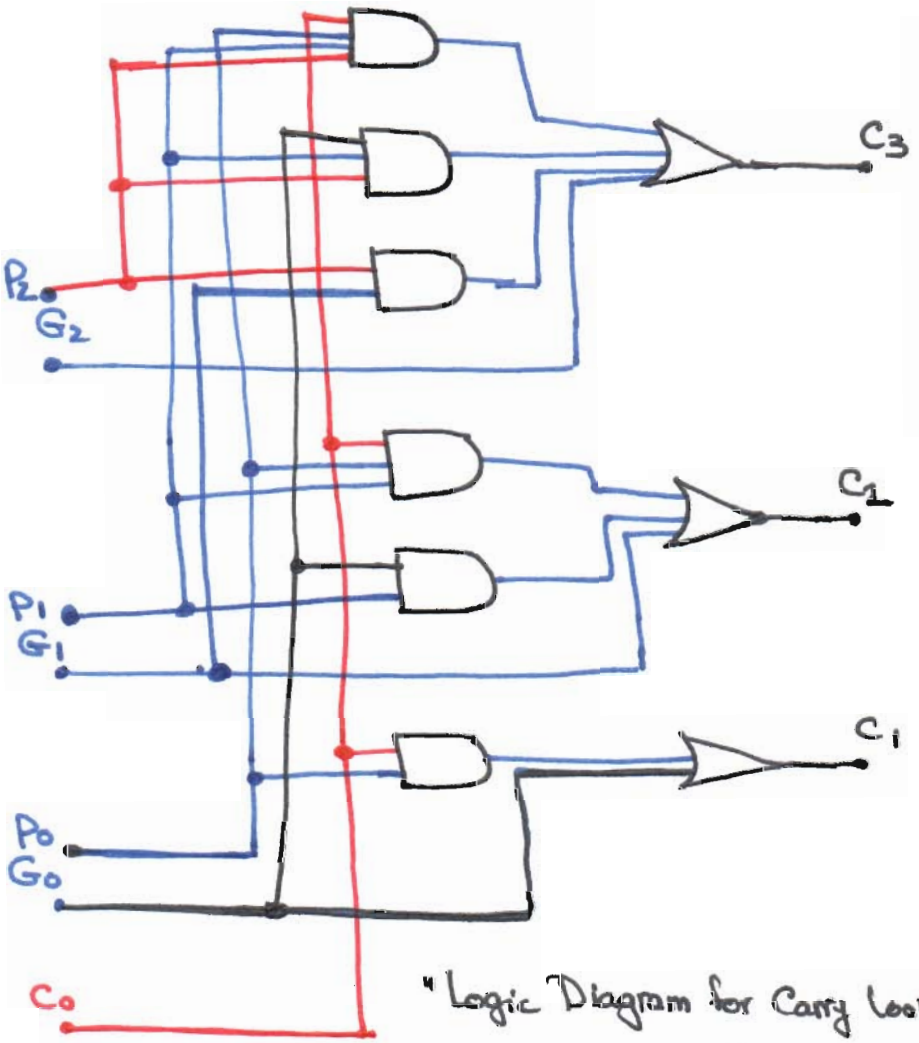
P_i - called a carry propagate → it determine whether a carry into stage i will propagate into stage i+1.

The Boolean function for the carry outputs of each stage and substitue the value of each C_i from the previous equations:

$$C_0 = \text{input carry}$$
$$C_1 = G_0 + P_0 C_0$$
$$C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0)$$
$$= G_1 + P_1 G_0 + P_1 P_0 C_0$$
$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0$$
$$= P_2 P_1 P_0 C_0$$

→ Sum-of-product form.

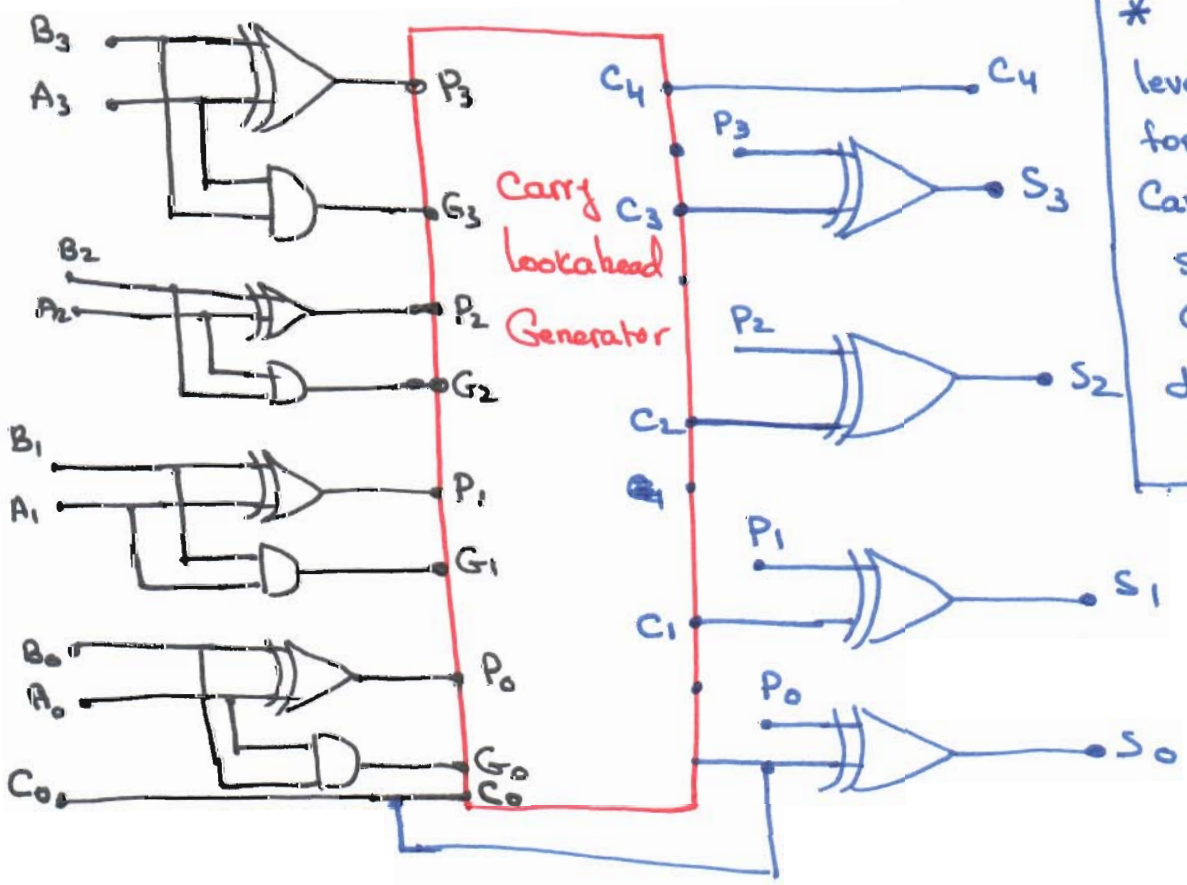
The three Boolean functions C_1, C_2, and C_3 are implemented in the **Carry lookahead generator**.



* C_3 does not have to wait for C_2 and C_1 to propagate, in fact C_3 is propagated at the same time as C_1 and C_2.

"Logic Diagram for Carry lookahead generator".

The construction of a four-bit adder with a carry lookahead scheme is the following:



* The two level-circuit for the output Carry C_4 is not shown, it can be easily derived by the equation

(4.) Binary subtractor:

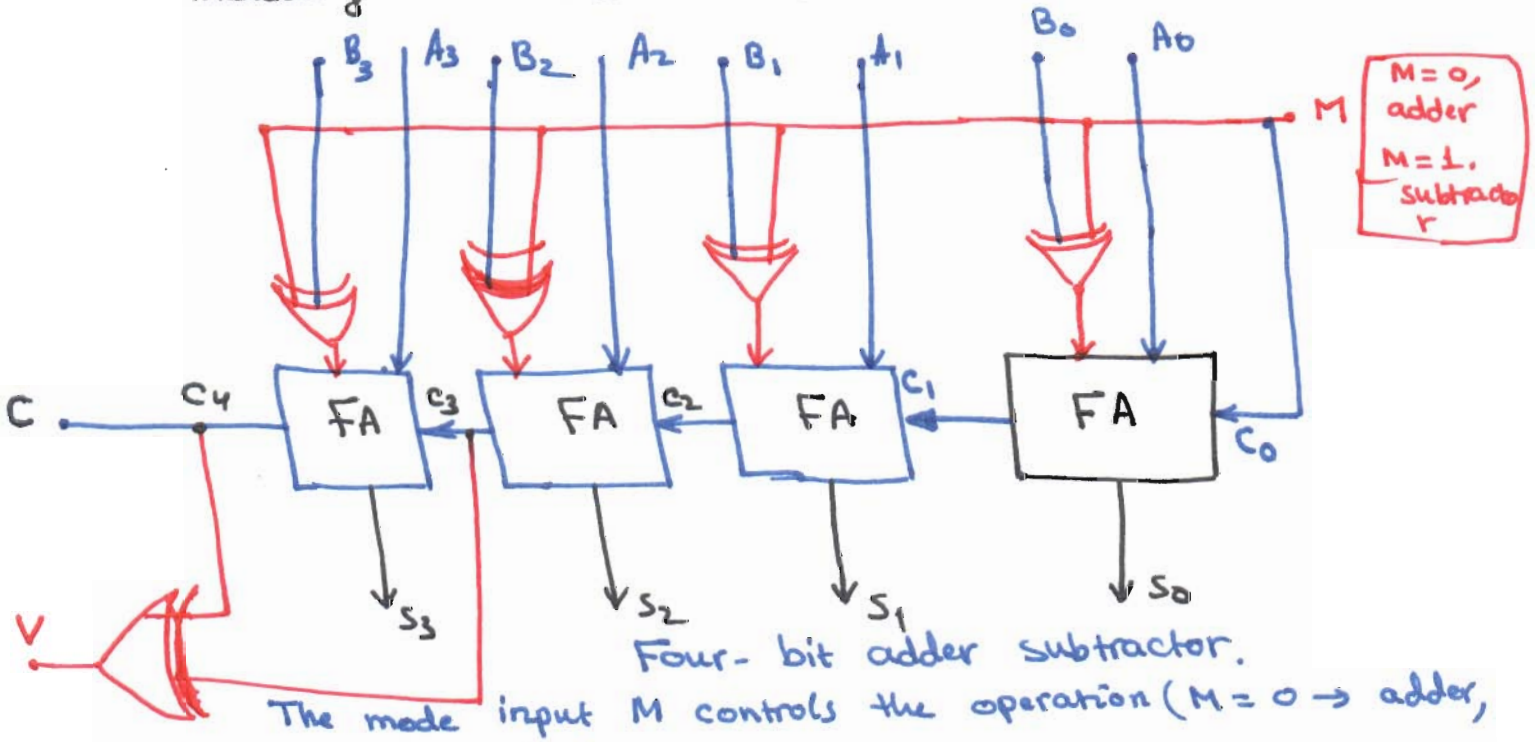
* A - B ⇒ using the 2's complements, the subtraction can be converted to addition:
* 2's complements can be obtained by talking the 1's complement and adding 1 to the lsd bit.

1) 1's complement can be implemented with inverters.
2) 1 can be added to the sum through the input carry.

The circuit for subtracting A - B consists of an adder with inverters placed between each data input B and the corresponding input of the full adder. The input carry C_0 must be equal to 1.

(5) Binary Adder- Subtractor:

The addition and subtraction operations can be combined into one circuit with one common binary adder by including an exculsive-OR gate with each full-adder.



Four- bit adder subtractor.

The mode input M controls the operation (M = 0 → adder, M = 1 → subtractor).

each XOR gate receives M signal ⇒ and B ⇒

when M = 0 ⇒ $B \oplus 0 = B \rightarrow$, carry = 0, The circuit A + B

when M = 1 ⇒ $B \oplus 1 = \overline{B}$, carry = 1, The circuit performs the operation A plus the 2's complement of B.

* The exclusive-OR with output V is for detecting an overflow.