Logical Design Lectures

Produced By Dr. Faris Llwaah

July 5, 2021

Lecture 6 – Half Adder & Full Adder

- An adder is a digital logic circuit in electronics that implements addition of numbers.
- The adders are used to calculate addresses, similar operations in the ALU in many computers and other types of processors.
- The adders can be built for many numerical representations like BCD or excess-3.
- There are two types: half adder and full adder.

■ **Half Adder**

- By using half adder, we can design simple addition with the help of logic gates. Let's see an addition of single bits.

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 10$

- Half adder truth table as follows:

Inputs		**Outputs**	
A	**B**	**Sum**	**Carry**
0	**0**	**0**	**0**
0	**1**	**1**	**0**
1	**0**	**1**	**0**
1	**1**	**0**	**1**

As it has been cleared in the above table, $1 - bit$ adder can be easily implemented with the help of the XOR Gate for the output 'SUM'. And an AND Gate for the 'Carry'. The half-adder is useful when you want to add one binary digit quantities. Figure 1 shows a logic circuit of Half Adder

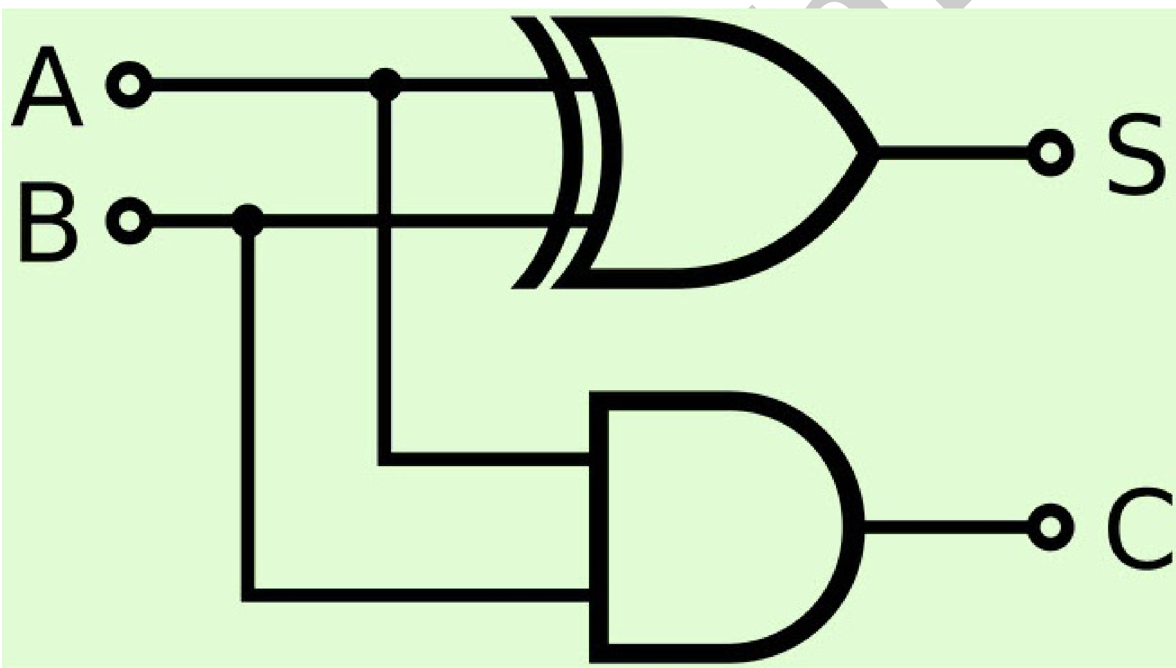


Figure 1: Half Adder Logic Circuit

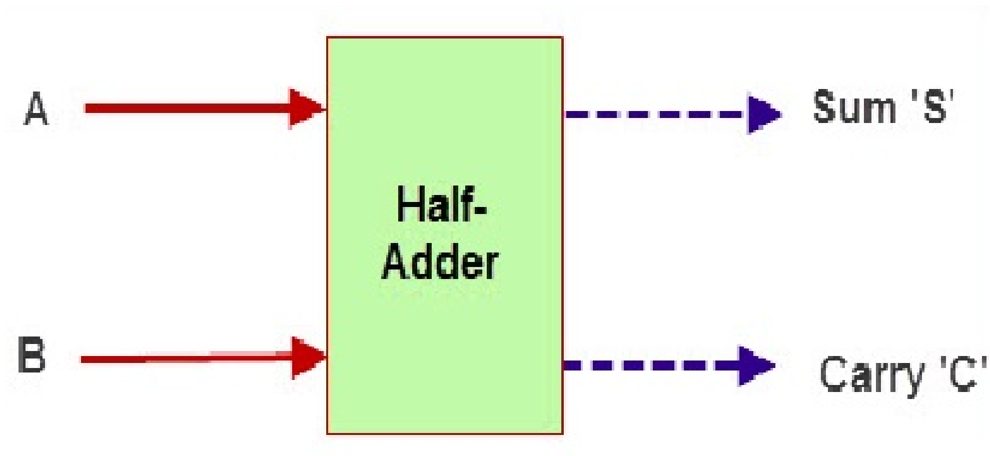


Figure 2: Half Adder Block Diagram

- The simplest expressions use the exclusive OR function for Sum and the And function for Carry:

  $Sum = A \oplus B$
  $Carry = A \bullet B$

■ Full Adder

This adder is difficult to implement than a Half Adder. The difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs, whereas half adder has only two inputs and two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. When a full-adder logic is designed, you string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.

- Full adder truth table as follows:

Inputs			Outputs	
Crn-1	**Bn**	**An**	**Sn**	**Crn**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- With the truth-table, the full adder logic can be implemented. we can see that the output S_n is an XOR between the input A_n and B_n and C_{rn-1}. The C_{rn} can be produced as OR between the first carry from half-adder and the second carry from half-adder.

- So, we can implement a full adder circuit with the help of two half adder circuits. At first, half adder will be used to add A_n andB_n to produce a partial Sum and a second half adder logic can be used to add C_{rn-1} to the Sum produced by the first half adder to get the final S_n output. Figure 3 shows a logic circuit of Full Adder.

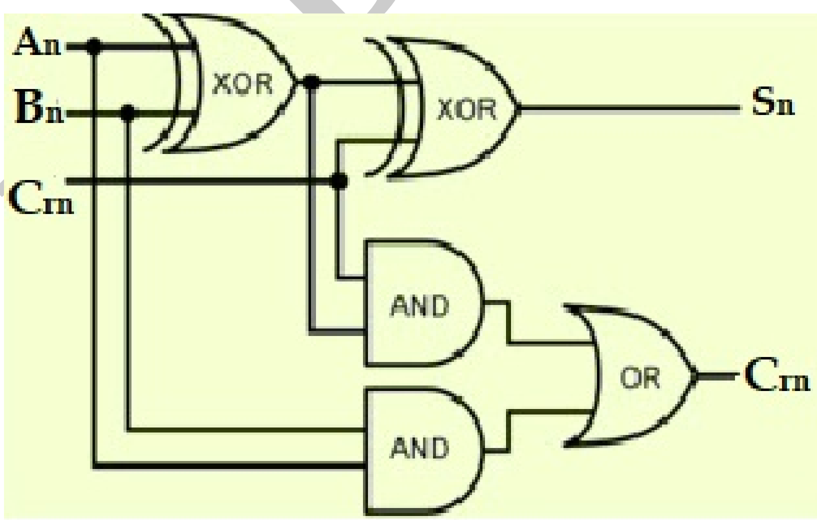


Figure 3: Full Adder Logic Circuit

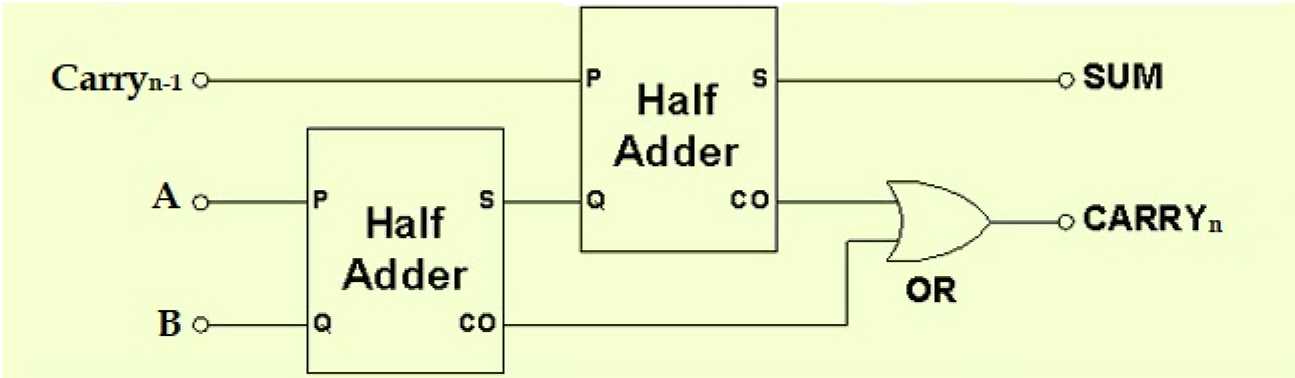
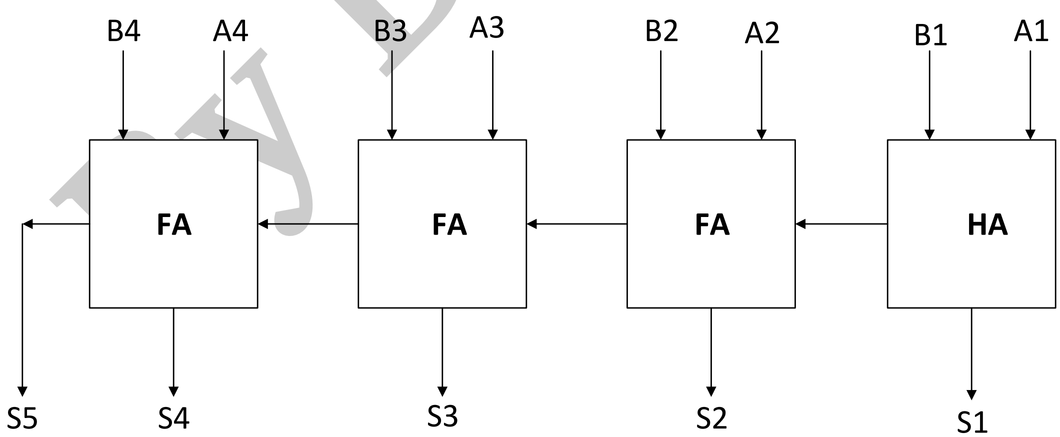


Figure 4: Full Adder Block Diagram

- The simplest expression use for the Full Adder as follows:
  $SUM = Carry_{rn-1} \oplus A \oplus B$

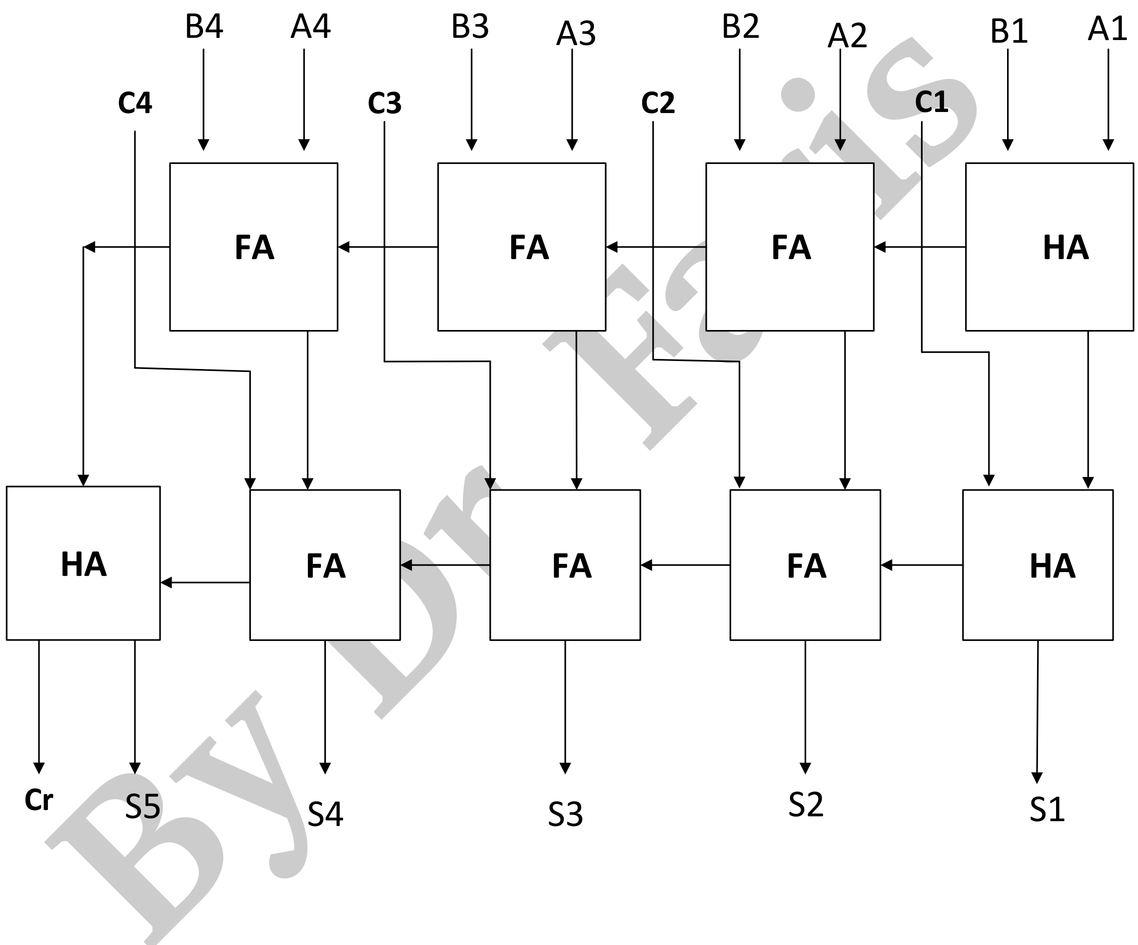
- With Figure 4, we can add two bits together, taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude. In a computer, for a multi-bit operation, each bit must be represented by a full adder and must be added simultaneously. Thus, to add two 4-bit numbers, we will need 3 full adders and 1 half adder which can be formed by cascading blocks as the following block diagram.



Example:

Design a logic circuit to add 3-binary numbers each of one has 4 bits.

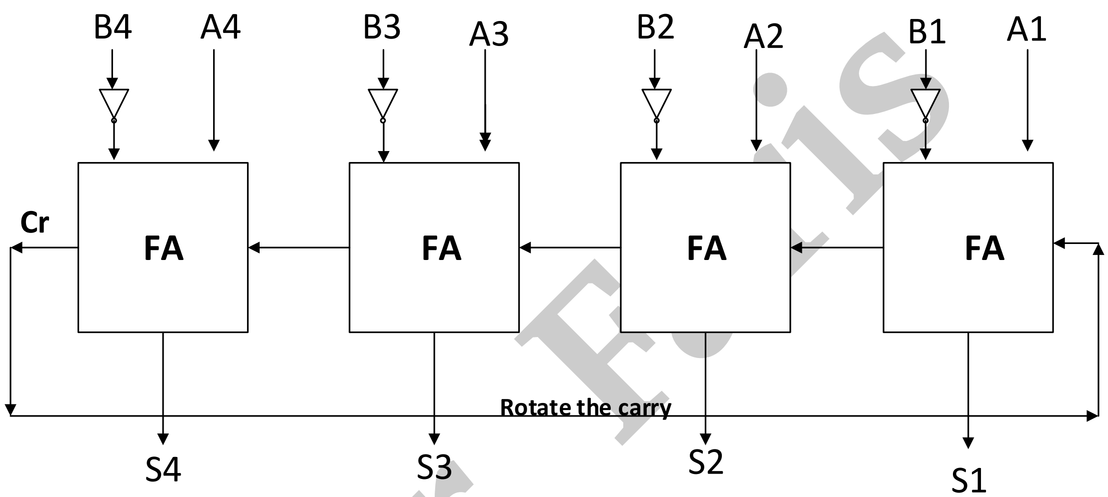
$A \rightarrow A_4A_3A_2A_1$
$B \rightarrow B_4B_3B_2B_1$
$C \rightarrow C_4C_3C_2C_1$



Example:

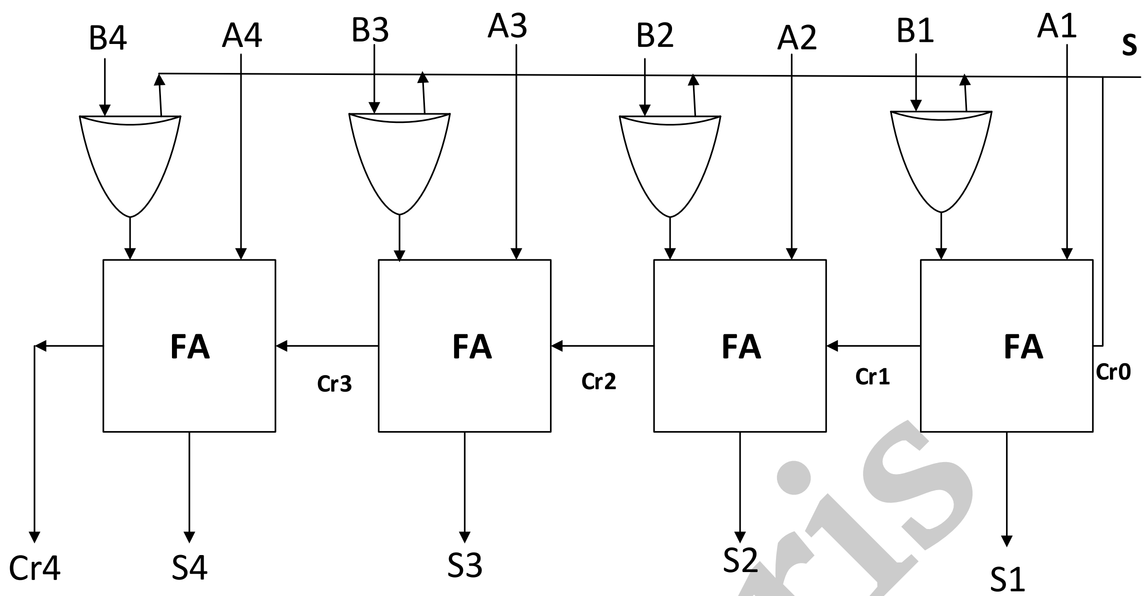
Design a 4-Bit subtract or using 4-Bit adders?
$A \rightarrow A_4A_3A_2A_1$
$B \rightarrow B_4B_3B_2B_1$
We can do it by adding A with the complement of B as follows:

$A_4A_3A_2A_1 + \overline{B_4}\,\overline{B_3}\,\overline{B_2}\,\overline{B_1}$



- The addition and subtraction can be combined into one circuit with common binary adder.
- This can be achieved by including an XOR gate with each Full-Adder.
- We have to add S control line operation to define the function of the circuit as follows:
  IF S = 0 " the circuit is working as Adder".
  Else S = 1 " the circuit is working as Subtractor".

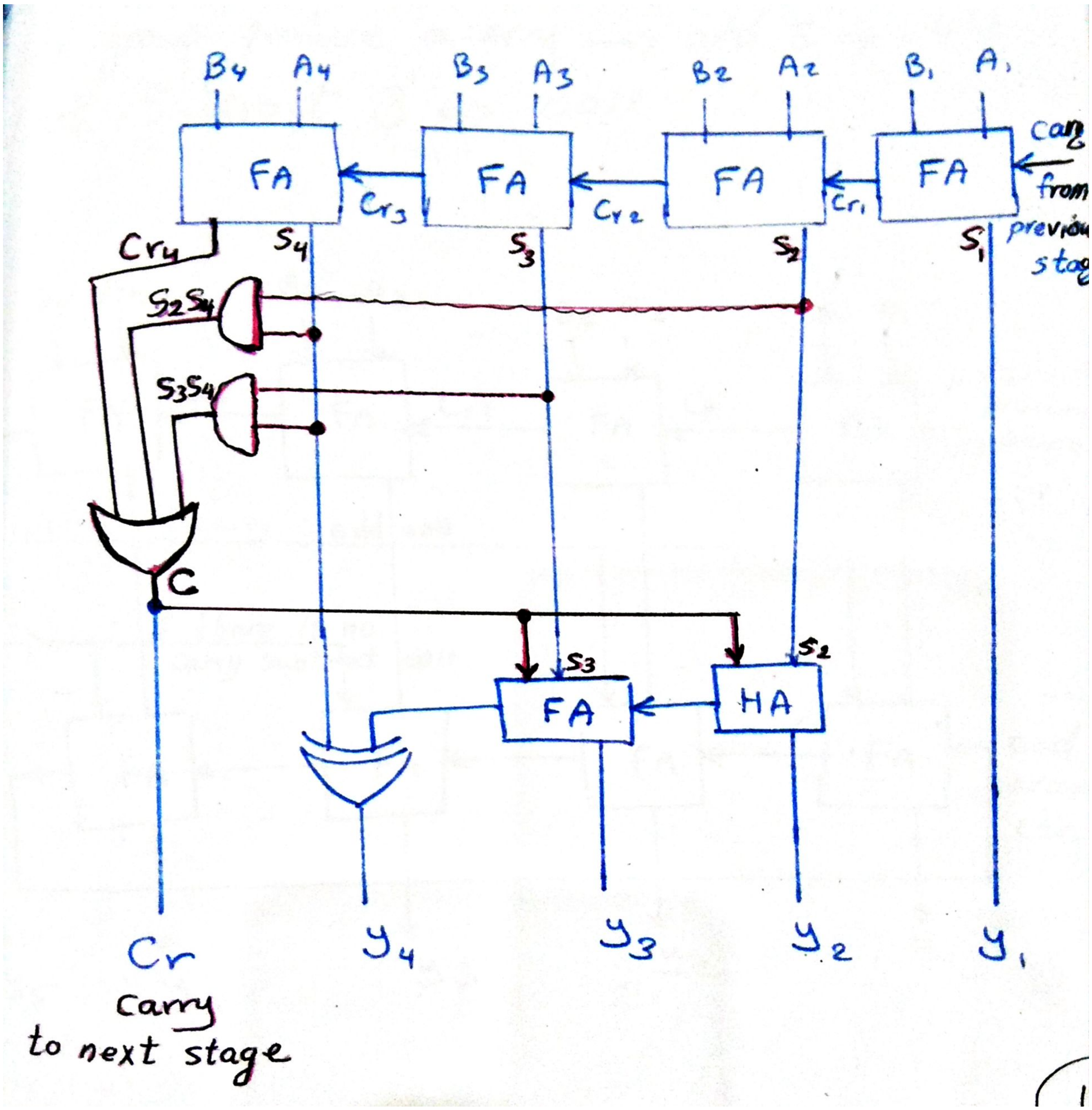
- When S = 0:
  $B_i \oplus 0 = B_i$
- When S = 1:
  $B_i \oplus 1 = \overline{B_i}$



Example:

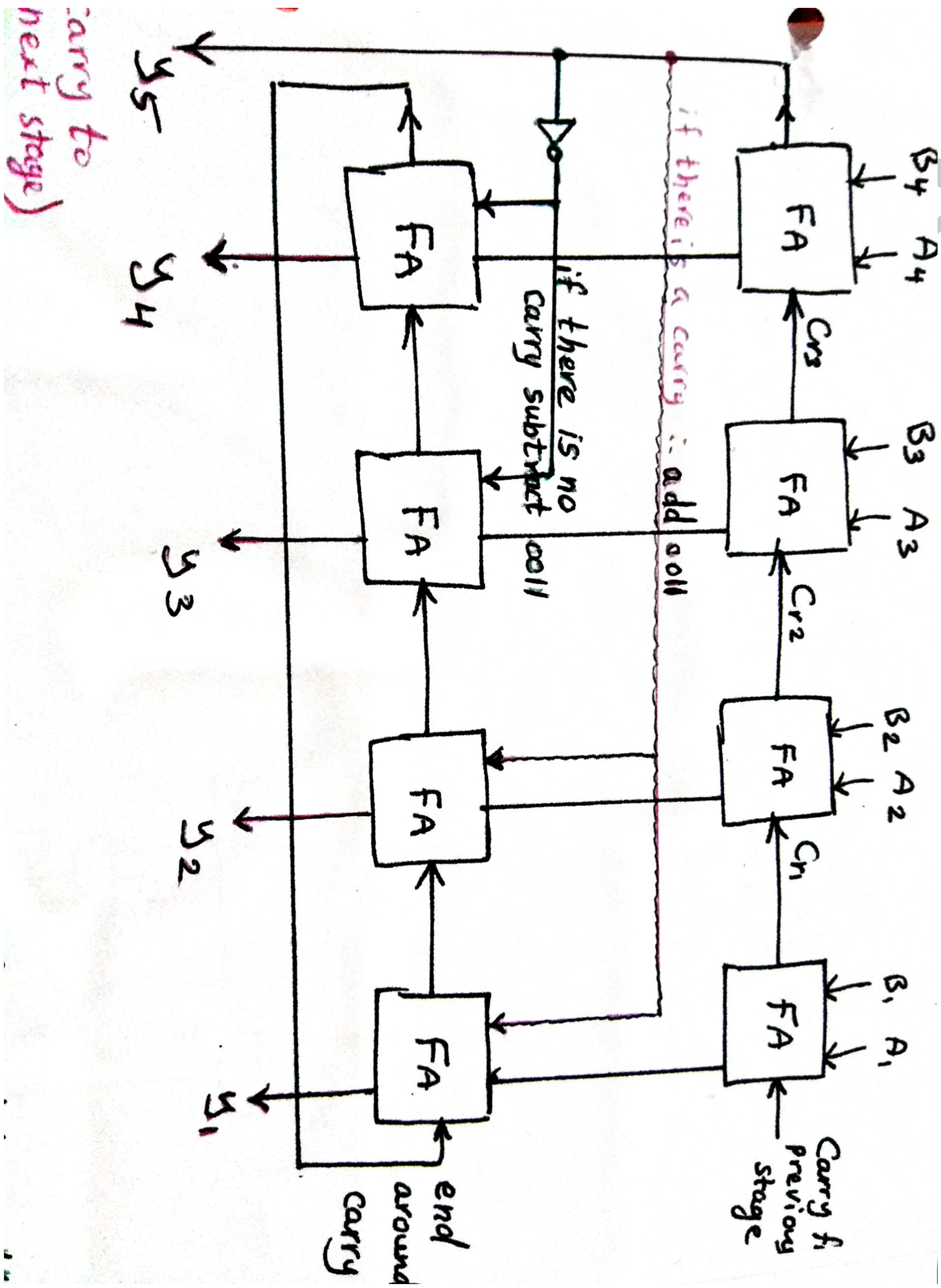
Design a logic circuit to add two numbers in BCD system?

- To design this circuit we should consider the following points:
- There are 6 invalid combinations in BCD such as 1010, 1011, 1100, 1101, 1110, 1111.
- The two and gates are used to detect if there is a sum invalid.
- we should add 0110 (6) to the sum if there is invalid case or there is a carry to the next stage.


B4
A4
B3
A3
B2
A2
B1
A1
FA
FA
FA
FA
Cr3
Cr2
Cr1
from
previous
Cr4
S4
S3
S2
S1
S2S4
S3S4
C
S3
S2
FA
HA
Cr
Y4
Y3
Y2
Y1
carry
to next stage

Example: Excess 3 adder

During an addition in Excess 3 code:
If the group produce a carry add 3 (0011)
if not, subtract 3 (0011), however the subtraction can be performed by adding complement of 0011 which is (1100).



Example:

Design a 2-Bit Binary Multiplier?
The multiplicand is multiplied by each bit of the multiplier, starting from LSB. Each such multiplication form a partial product. Successive partial product are shifted one bit to the left.

- The final product is obtained from the sum of the partial products. the following logic circuit: