AAB University

Faculty of Computer Sciences

Introduction to Digital Technologies and Circuits

Week 8:

Arithmetic Circuits

Asst. Prof. Dr. **Mentor Hamiti**

mentor.hamiti@universitetiaab.com

Last Time

- Combinational Logic Circuits
  - Decoders
  - Encoders
  - Multiplexers
  - De-multiplexers
  - Other Combinational Circuits

Today

- Combinational Arithmetic Circuits
  - Adders
  - Sub tractors
  - Multipliers
  - Dividers
  - Other Circuits

Combinational Arithmetic Circuits

- Arithmetic circuits are the ones which perform arithmetic operations like: addition, subtraction, multiplication, division, parity calculation, etc.

- Most of the time, designing these circuits is the same or similar as designing:
  - Encoders,
  - Decoders,
  - Multiplexers,
  - De-multiplexers,
  - Code converters, etc.

Adders

- Adders are the basic building blocks of all arithmetic circuits
- Adders add two binary numbers and give out sum and carry as output
  - Example: X + Y = S + c
- Basically we have two types of adders:
  - Half Adder.
  - Full Adder.

Half Adder

- Adding two single-bit binary values **X**, **Y** produces a sum **S** bit and a carry out **C**-out bit. This operation is called half addition and the circuit to realize it is called a half adder.

- Truth Table:

X	Y	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S(X,Y) = m^1(1,2) = X'Y + XY' = X \oplus Y$

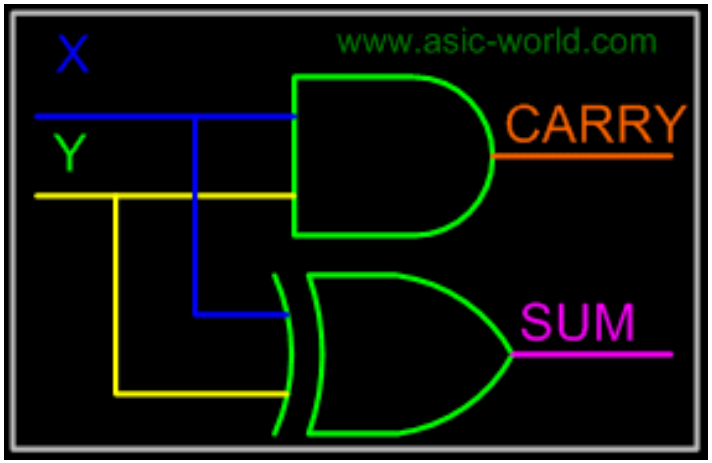
$CARRY(X,Y) = m^1(3) = X \cdot Y$

Half Adder

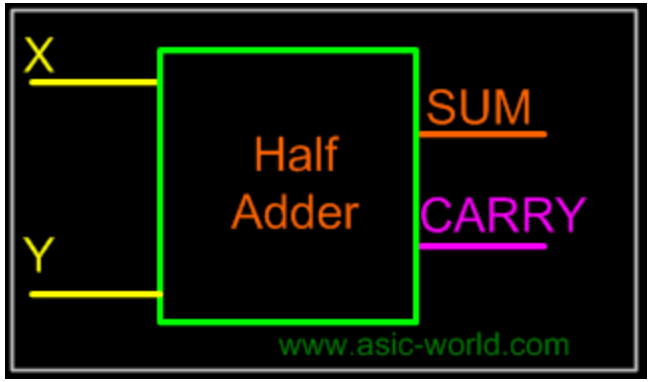
$S(X,Y) = m^1(1,2) = X'Y + XY' = X \oplus Y$

$CARRY(X,Y) = m^1(3) = X \cdot Y$

Circuit:



Symbol:



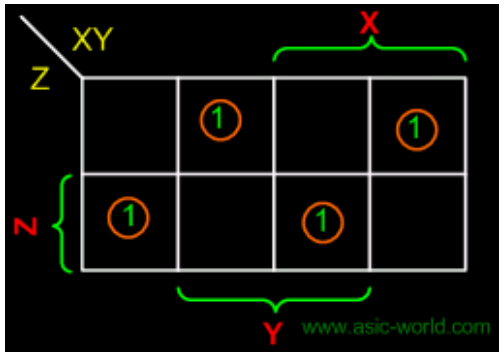
Full Adder

- Full adder takes a three-bits input. Adding two single-bit binary values X, Y with a carry input bit C-in produces a sum bit S and a carry out C-out bit.

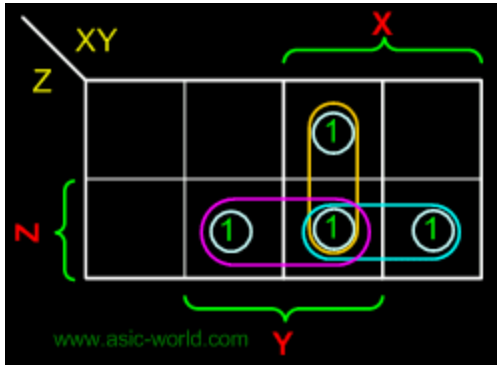
- Truth Table:

X	Y	Z	SUM	CARRY
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

K-Diagrams:



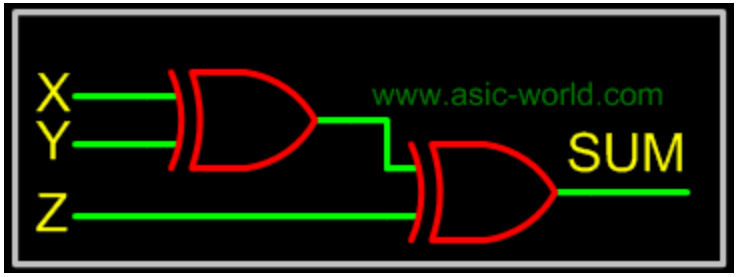
SUM=X⊕Y⊕Z



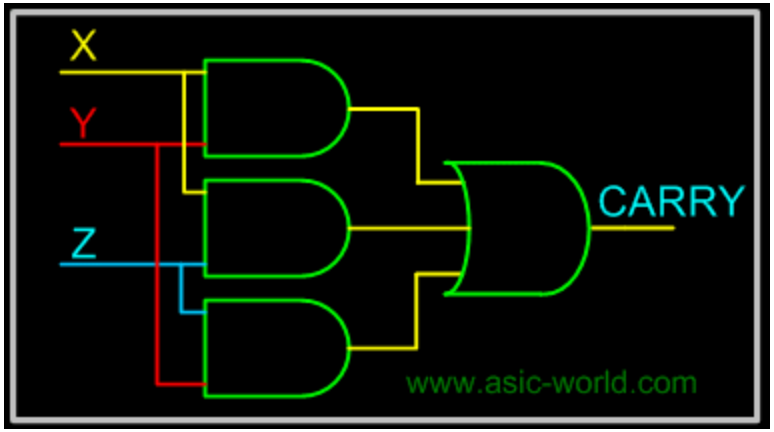
C=XY+YZ+XZ

Full Adder

SUM = X⊕ Y⊕ Z



CARRY = XY+YZ+XZ



Symbol:

n-bit (Carry Ripple) Adder

- An **n-bit adder** used to add two n-bit binary numbers can be built by connecting **n full adders in series**.
  Each full adder represents a bit position **j** (from **0** to **n-1**).
- Each carry out **C-out** from a full adder at position **j** is connected to the carry in **C-in** of the full adder at higher position **j+1**.
  The output of a full adder at position j is given by:

$$S_j = X_j \oplus Y_j \oplus C_j$$

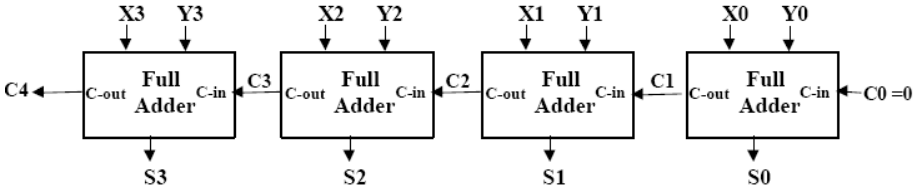
$$C_{j+1} = X_j \cdot Y_j + X_j \cdot C_j + Y_j \cdot C_j$$

- In the expression of the sum C_j must be generated by the full adder at lower position j. The propagation delay in each full adder to produce the carry is equal to two gate delays = 2 D. Since the generation of the sum requires the propagation of the carry from the lowest position to the highest position, the total propagation delay of the adder is approximately:
  - **Total Propagation delay = 2 nD**

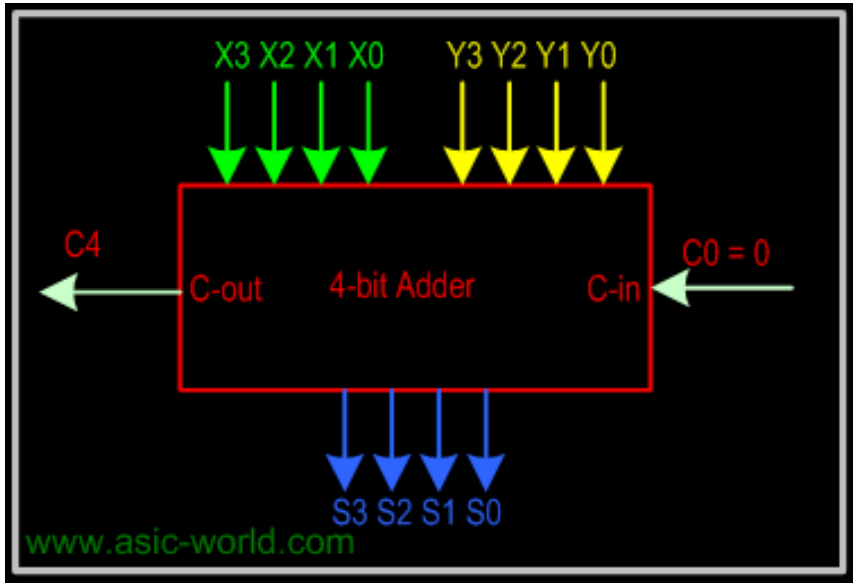
4-bit Carry Ripple Adder

- Adds two 4-bit numbers:

$$X = X_3\ X_2\ X_1\ X_0$$
$$Y = Y_3\ Y_2\ Y_1\ Y_0$$

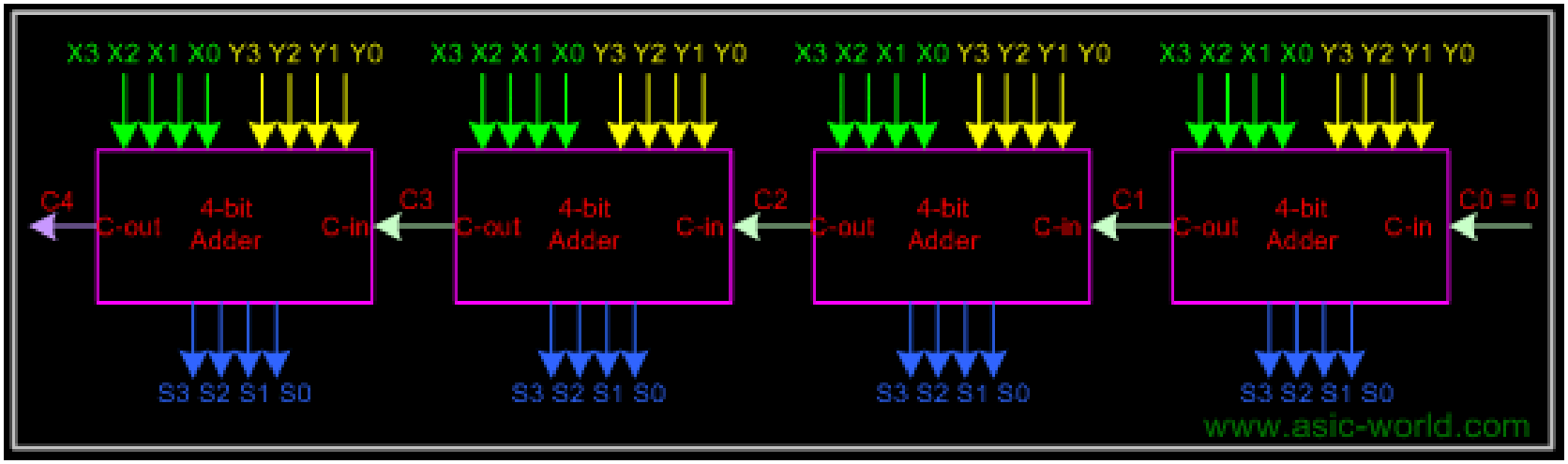


- Producing the sum: $S = S_3\ S_2\ S_1\ S_0$,
  $C_{out} = C_4$ from the most significant position j=3
  - Total Propagation delay = 2 nD = 8D or 8 gate delays!



Larger Adder

- Example: 16-bit adder using 4x4-bit adders.
  - Adds two 16-bit inputs **X** (bits X_0 to X_{15}), **Y** (bits Y_0 to Y_{15}) producing a 16-bit Sum **S** (bits S_0 to S_{15}) and a carry out **C_{16}** from the most significant position



- Propagation delay for 16-bit adder = 4 x propagation delay of 4-bit adder = 4 x 2 nD = 4 x 8D = 32 D or 32 gate delays

Carry Look-Ahead Adder

- The delay generated by an N-bit adder is proportional to the length N of the two numbers X and Y that are added because the carry signals have to propagate from one full-adder to the next.
- For large values of N, the delay becomes unacceptably large so that a special solution needs to be adopted to accelerate the calculation of the carry bits.
- This solution involves a "look-ahead carry generator" which is a block that simultaneously calculates all the carry bits involved.
- The design of the look-ahead carry generator involves two Boolean functions named Generate and Propagate. For each input bits pair these functions are defined as:

$$G_i = X_i \cdot Y_i$$
$$P_i = X_i + Y_i$$

Carry Look-Ahead Adder

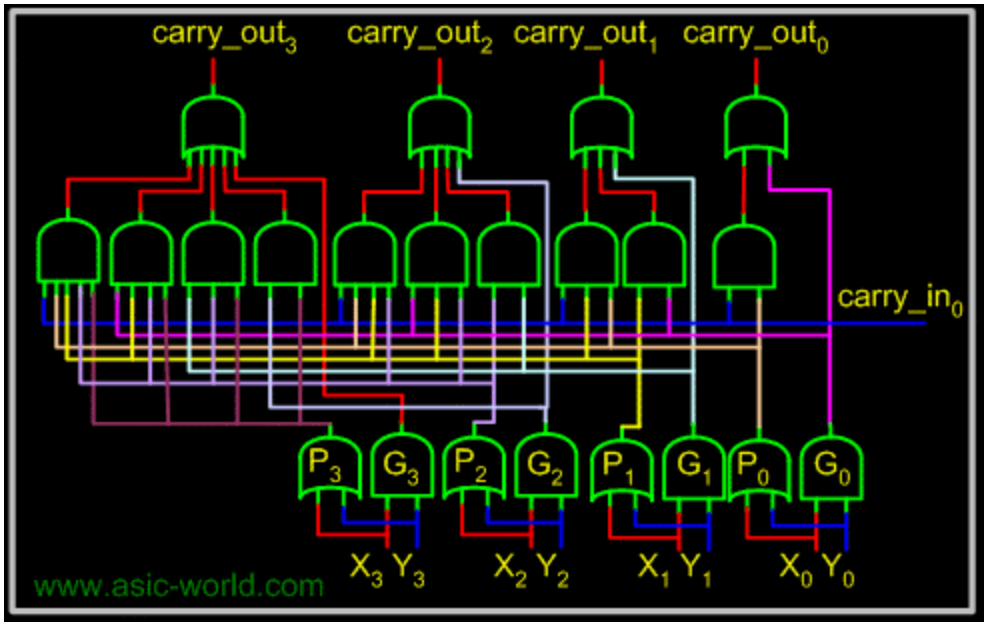
- For a four-bit adder the carry-outs are calculated as follows:

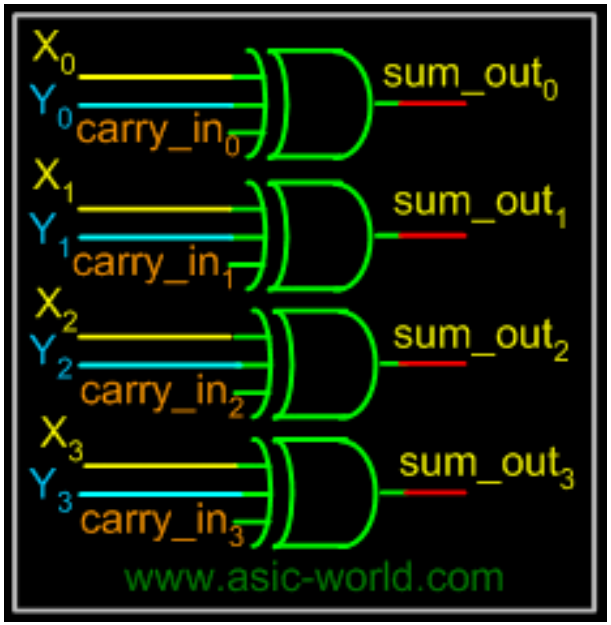
carry_out0 = G0 + P0 * carry_in0

carry_out1 = G1 + P1 * carry_out0 = G1 + P1G0 + P1P0 * carry_in0

carry_out2 = G2 + P2G1 + P2P1G0 + P2P1P0 * carry_in0

carry_out3 = G3 + P3G2 + P3P2G1 + P3P2P1G0 + P3P2P1 * carry_in0





Subtractor

- Subtractor circuits take two binary numbers as input and subtract one binary number input from the other binary number input.

- Similar to adders, it gives out two outputs, difference and borrow.

- There are two types of subtracters:
  - Half Subtracter.
  - Full Subtracter.

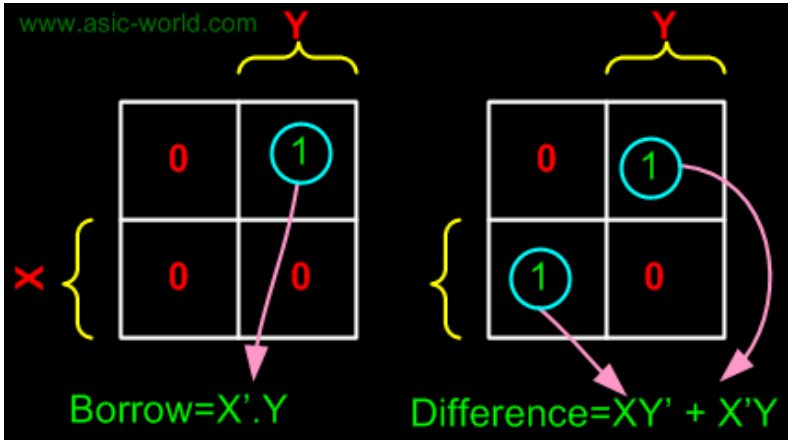
Half Subtracter

- The half-subtracter is a combinational circuit which is used to perform subtraction of two bits.
- It has two inputs, X (minuend) and Y (subtrahend) and two outputs D (difference) and B (borrow).

- Truth Table:

X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Boolean expressions:



Difference = XY'+X'Y

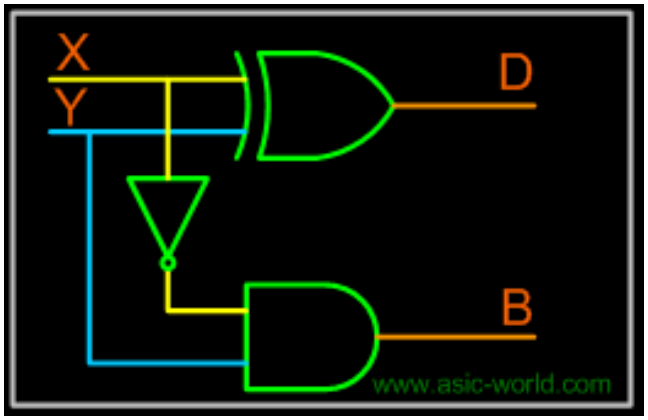
Borrow = X' Y

Half Subtracter

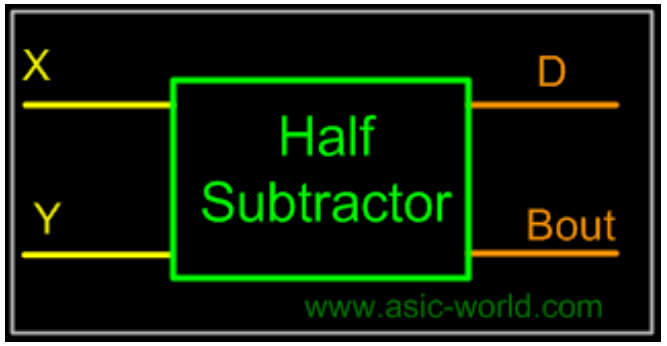
Difference = XY'+X'Y

Borrow = X' Y

Circuit:



Symbol:



Full Subtracter

- A full subtracter is a combinational circuit that performs subtraction involving three bits, namely minuend, subtrahend, and borrow-in.

- Truth Table:

X	Y	B_{in}	D	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = X \oplus Y \oplus B_{in}$$

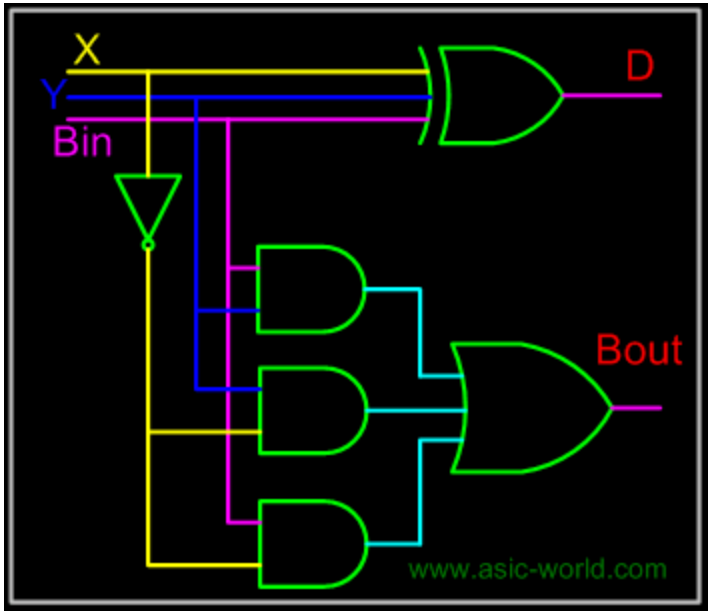
$$B = X'Y + X'B_{in} + YB_{in}$$

Full Subtracter

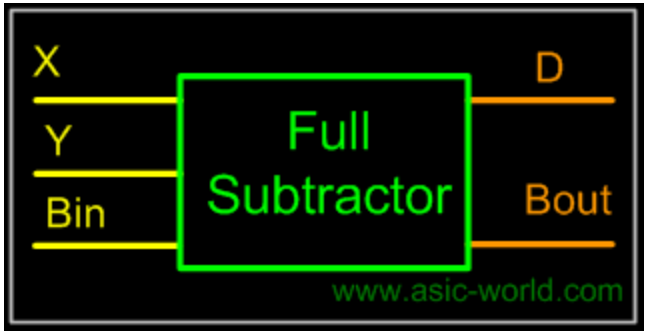
$D = X \oplus Y \oplus Bin$

$B = X'Y + X'B_{in} + YB_{in}$

- Circuit:

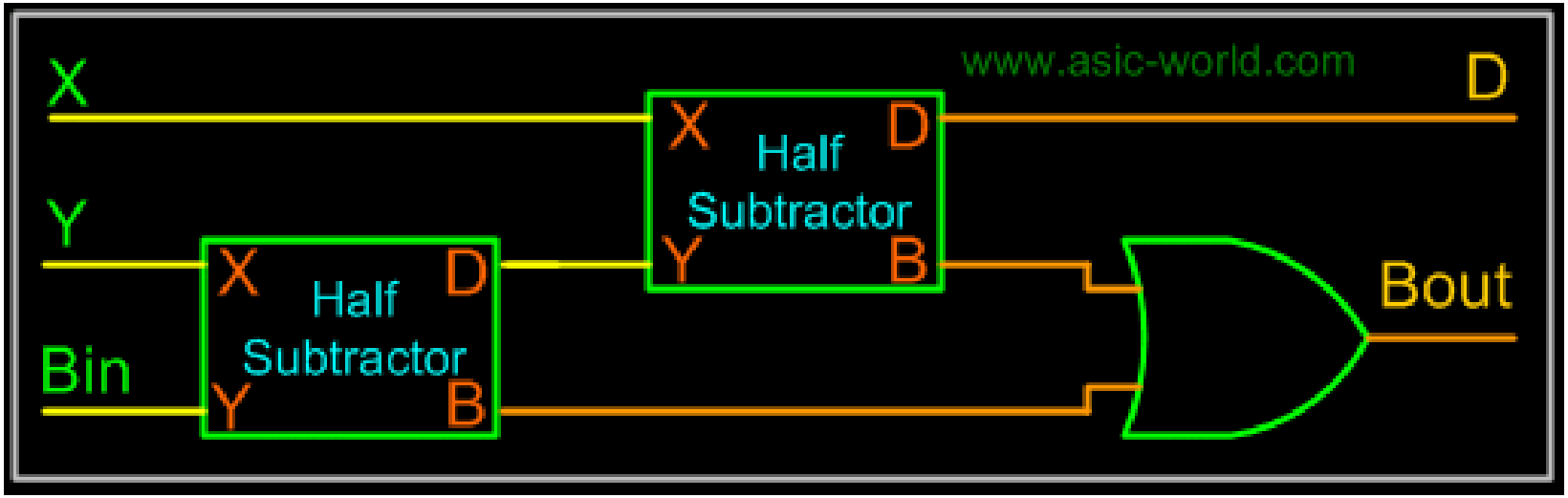


Symbol:



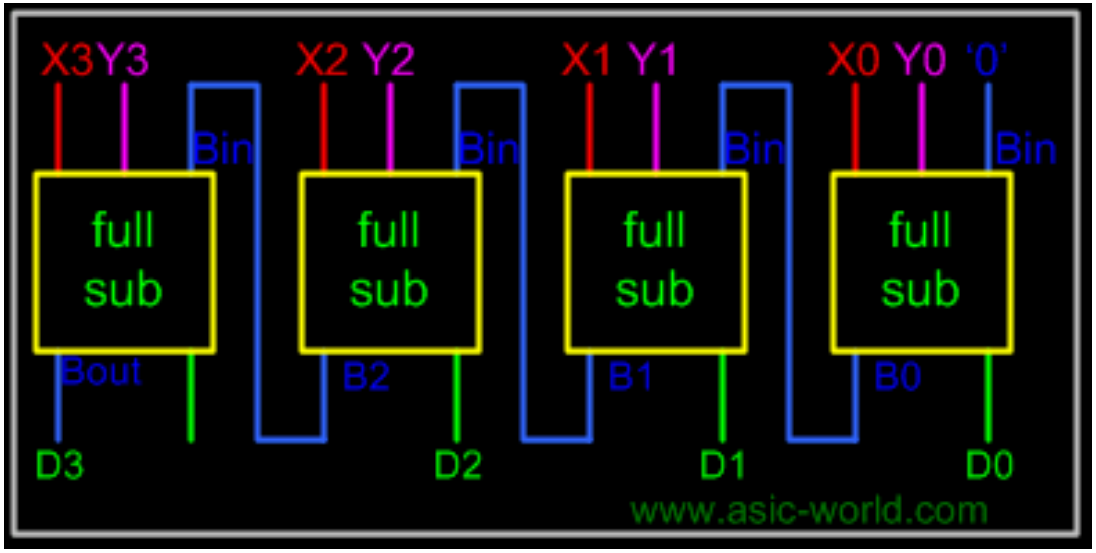
Full Subtracter

- Full subtractor composed by 2 half substractors:



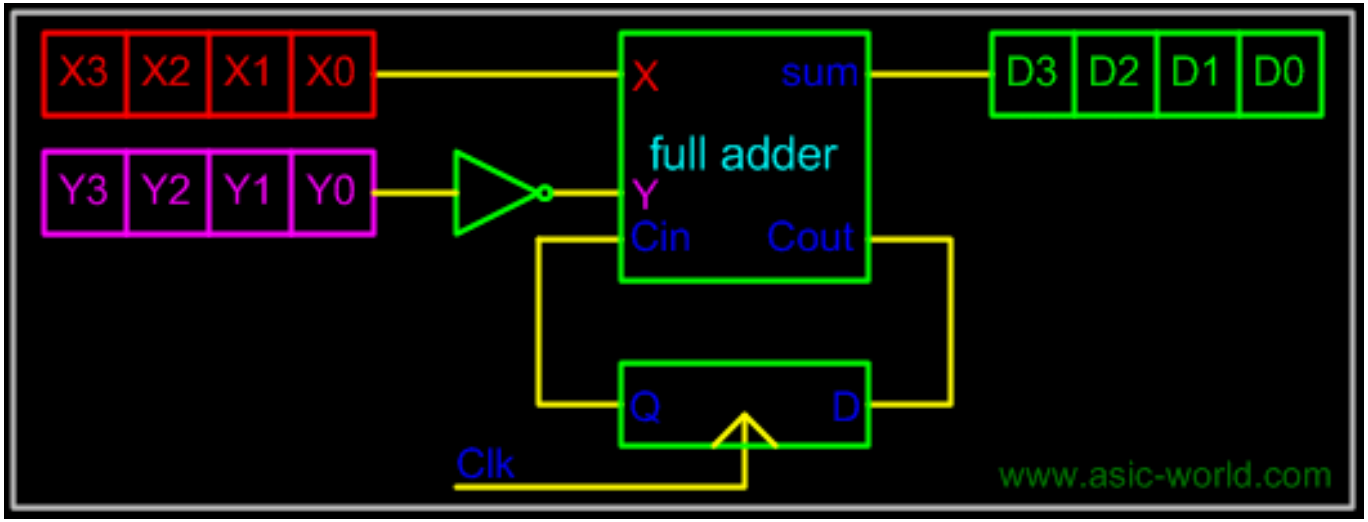
Parallel Binary Subtracter

- Parallel binary subtracter can be implemented by cascading several full-subtracters.
- Block level representation of a 4-bit parallel binary subtracter, which subtracts 4-bit **Y3Y2Y1Y0** from 4-bit **X3X2X1X0**. It has 4-bit difference output **D3D2D1D0** with borrow output B_{out}



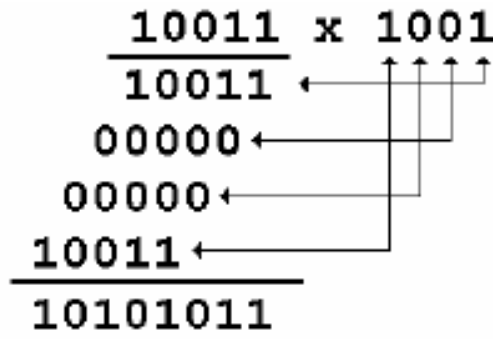
Serial Binary Subtracter

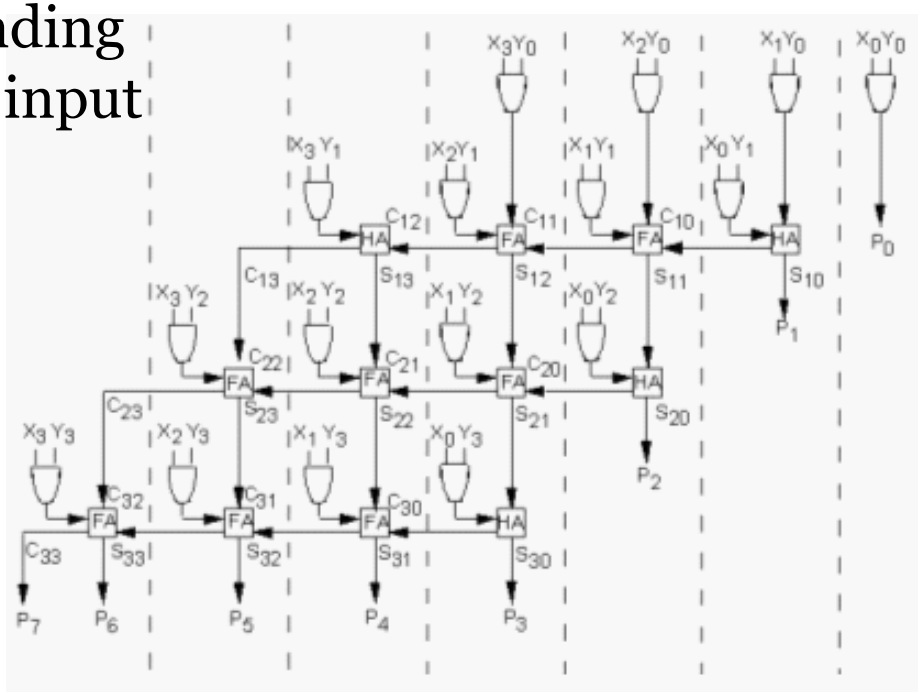
- A serial subtracter can be obtained by converting the serial adder using the 2's complement system.
- The subtrahend is stored in the Y register and must be 2's complemented before it is added to the minuend stored in the X register.
- The circuit for a 4-bit serial subtracter using full-adder:



Multipliers

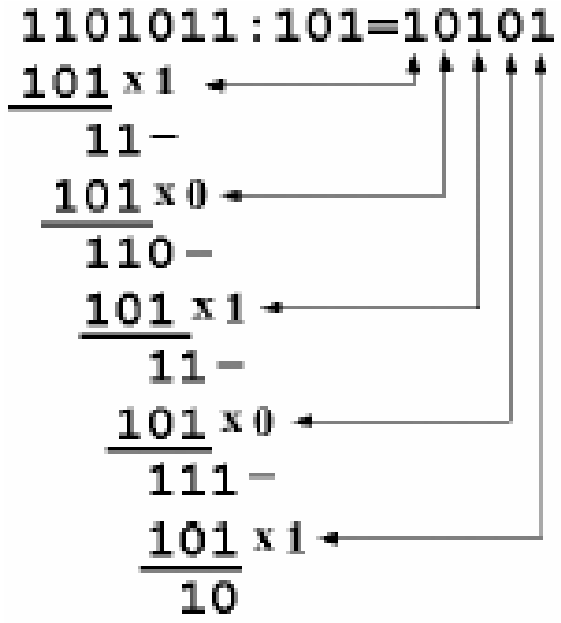
- Multiplication is achieved by adding a list of shifted multiplicands according to the digits of the multiplier.
- An n-bit X n-bit multiplier can be realized in combinational circuitry by using an array of n-1 n-bit adders where each adder is shifted by one position.
- For each adder one input is the shifted multiplicand multiplied by 0 or 1 (using AND gates) depending on the multiplier bit, the other input is n partial product bits.





Dividers

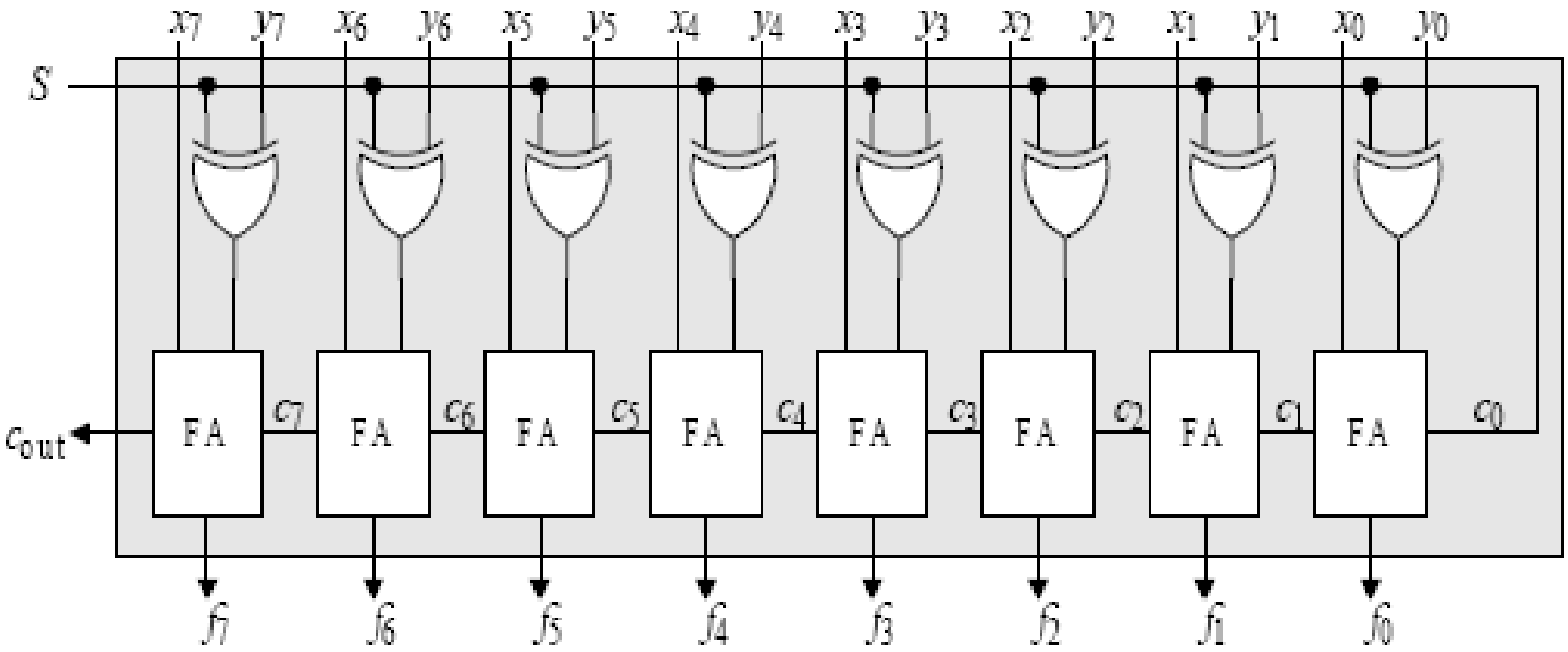
- The binary divisions are performed in a very similar manner to the decimal divisions.
- Thus, the second number is repeatedly subtracted from the figures of the first number after being multiplied either with '**1**' or with '**0**'.
- The multiplication bit ('1' or '0') is selected for each subtraction step in such a manner that the subtraction result is not negative.
- The division result is composed from all the successive multiplication bits while the remainder is the result of the last subtraction step.



Other Circuits

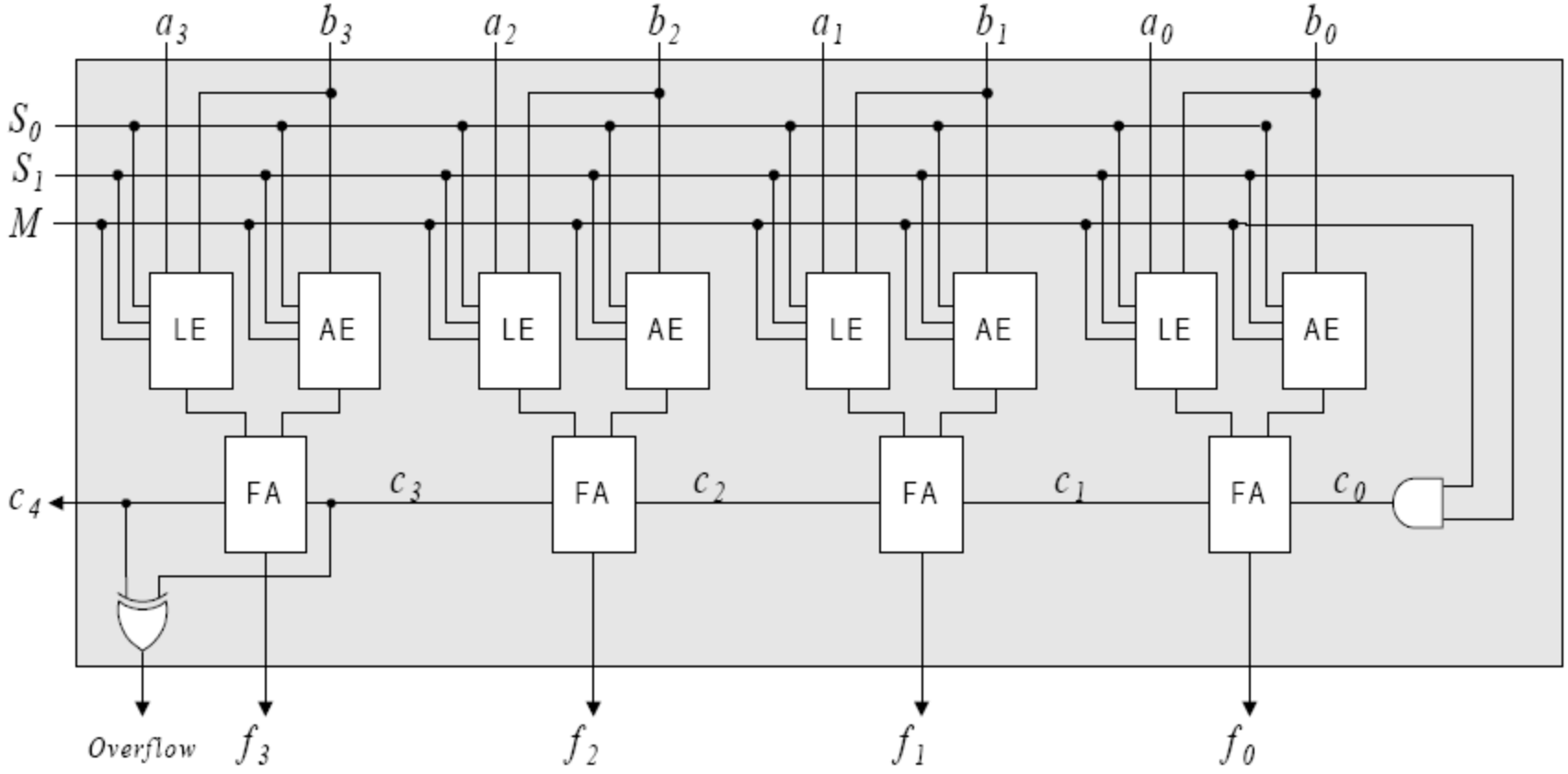
- Adder/Subtractor

S	Function	Comment
0	$X + Y$	Addition
1	$X + Y' + 1$	Subtraction



Other Circuits

- Arithmetic-Logic Unit (ALU)



Arithmetic Extender (AE)

Logic Extender (LE)

M	S_1	S_0	Function Name
1	0	0	Decrement
1	0	1	Add
1	1	0	Subtract
1	1	1	Increment

M	S_1	S_0	Function Name
0	0	0	Complement
0	0	1	AND
0	1	0	Identity
0	1	1	OR

- Questions?!

mentor.hamiti@universitetiaab.com