

Computer Organization and Architecture

Prepared by **Dr. Eng. Gamal Selim**

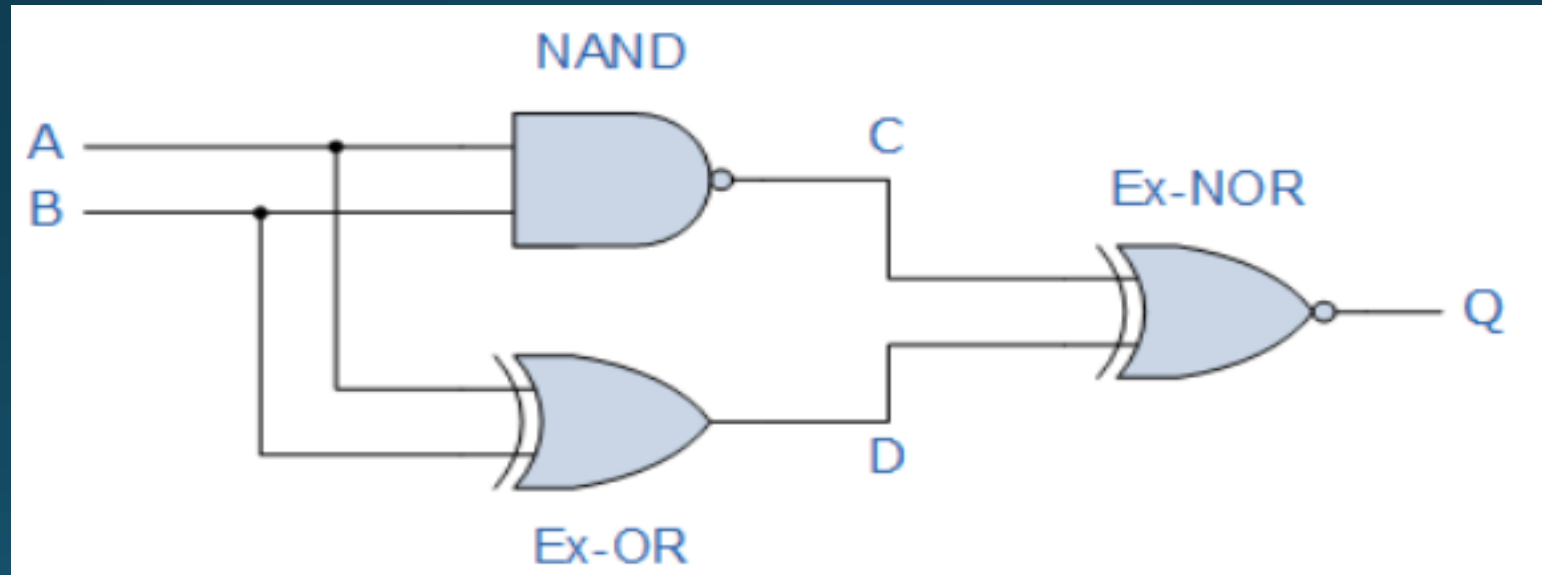
Lecture 06



Revision

Boolean Algebra Examples No1

Construct a Truth Table for the logical functions at points C, D and Q in the following circuit and identify a single logic gate that can be used to replace the whole circuit.



Boolean Algebra Examples No1

First observations tell us that the circuit consists of a 2-input NAND gate, a 2-input EX-OR gate and finally a 2-input EX-NOR gate at the output. As there are only 2 inputs to the circuit labelled A and B, there can only be 4 possible combinations of the input (2^2) and these are: 0-0, 0-1, 1-0 and finally 1-1. Plotting the logical functions from each gate in tabular form will give us the following truth table for the whole of the logic circuit below.

Inputs		Output at		
A	B	C	D	Q
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	0	1

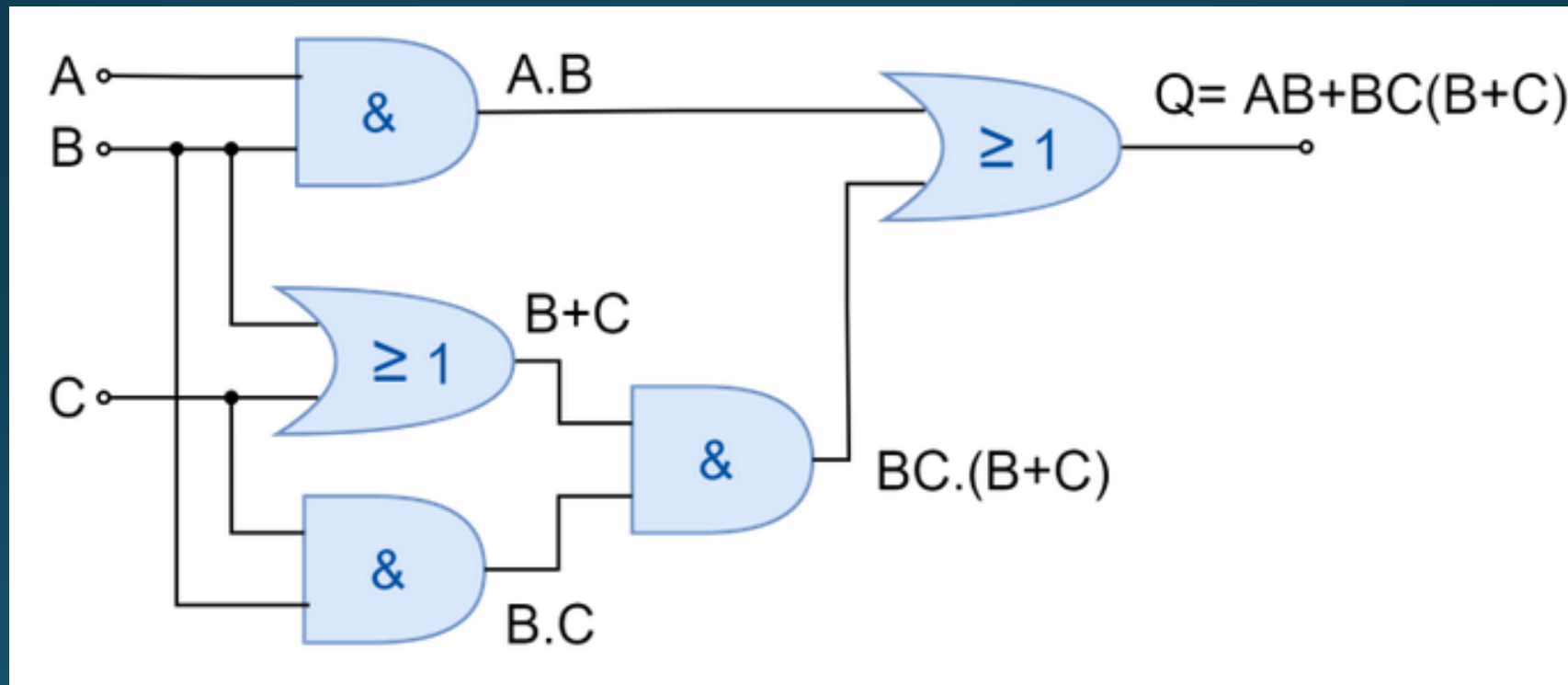
Boolean Algebra Examples No1

From the previous truth table, column C represents the output function generated by the NAND gate, while column D represents the output function from the Ex-OR gate. Both of these two output expressions then become the input condition for the Ex-NOR gate at the output.

It can be seen from the truth table that an output at Q is present when any of the two inputs A or B are at logic 1. The only truth table that satisfies this condition is that of an OR Gate. Therefore, the whole of the above circuit can be replaced by just one single **2-input OR Gate**.

Boolean Algebra Examples No2

In this example, the desired logic function is reduced by applying Boolean Algebra Laws and Theorems.



Boolean Algebra Examples No2

The steps involved in the reduction of Boolean expression are as follow:

$$AB + BC(B+C)$$

Applying Distributive Law

$$AB + BBC + BCC$$

Applying Identity Law ($AA=A$) to 2nd and 3rd terms

$$AB + BC + BC$$

Applying Identity Law ($A + A = A$) to 2nd and 3rd terms

$$AB + BC$$

Taking out common B

$$B(A+C)$$

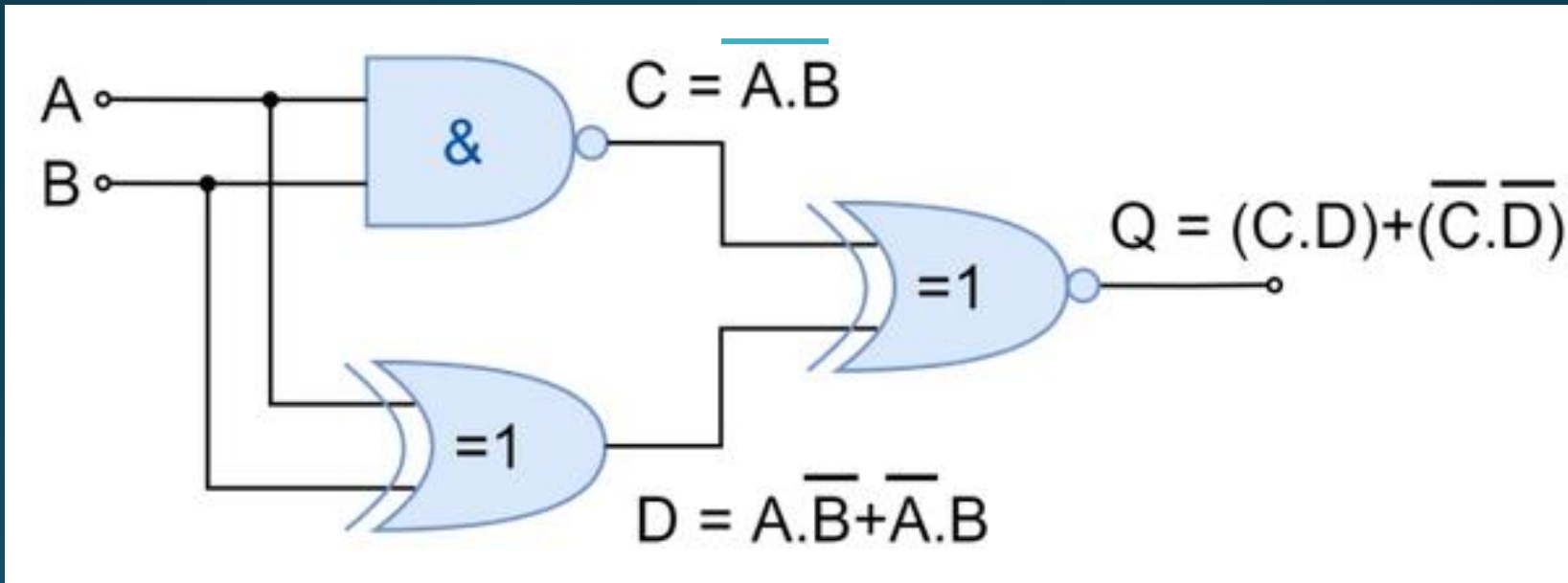
Boolean Algebra Examples No2

The reduced expression $B(A+C)$ produces the same logical output as of original expression. The truth table of the original Boolean expression as well as of reduced expression (red) is shown below:

Truth Table of Example No. 1								
A	B	C	A.B	B+C	B.C	BC.(B+C)	Q=AB+BC(B+C)	B(A+C)
0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0
0	1	0	0	1	0	0	0	0
0	1	1	0	1	1	1	1	1
1	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0
1	1	0	1	1	0	0	1	1
1	1	1	1	1	1	1	1	1

Boolean Algebra Examples No3

In this example, the desired logic function is reduced by applying Boolean Algebra Laws and Theorems.



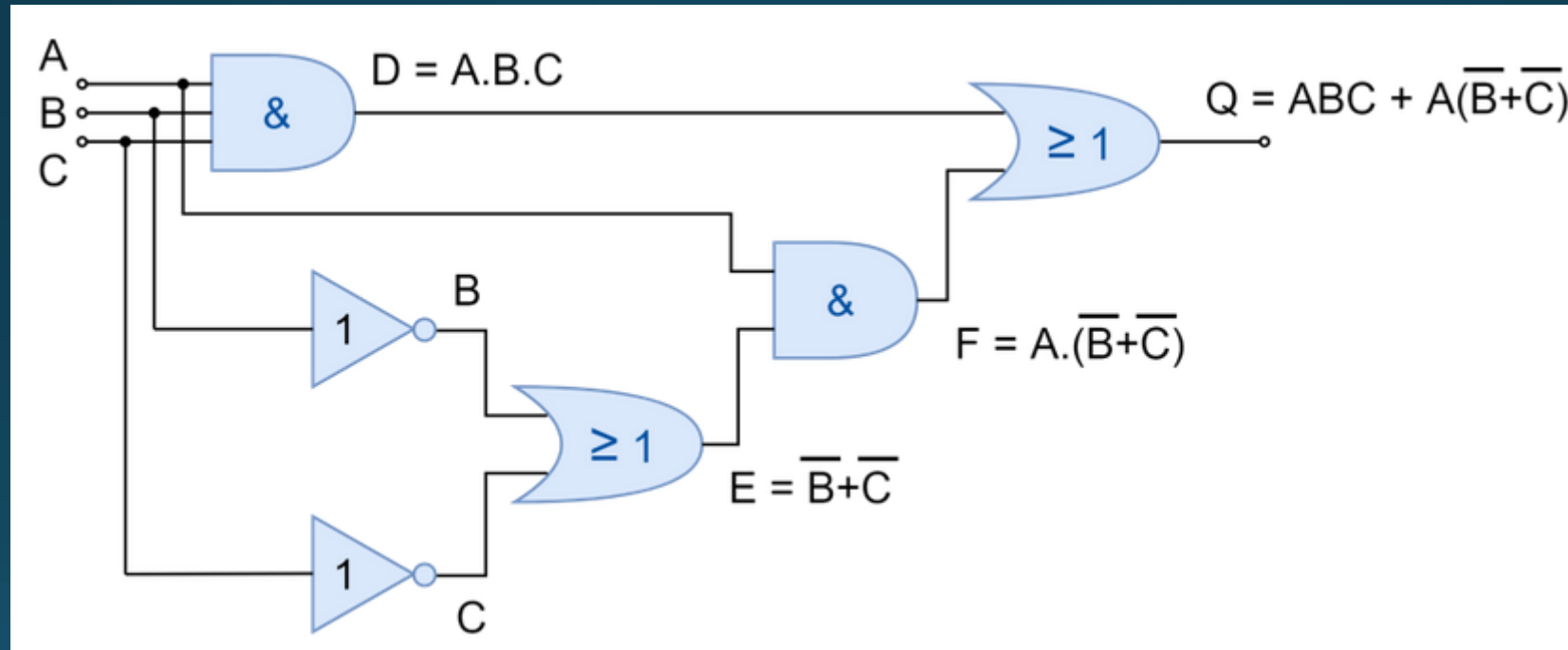
Boolean Algebra Examples No3

The digital logic circuit comprises three logic gates i.e. AND, NOR, and OR. The output of the NOR is converted to the product of inversed individual input. The final output of the OR gate which is the function of the whole logic circuit comes to be equivalent to the function of XNOR logic. Hence, the given three gates digital logic circuit can be replaced by a single gate of Exclusive-NOR gate.

A	B	C	D	Q
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	0	1

Boolean Algebra Examples No4

In the final example, a more complex digital logic circuit has simplified which consists of six logic gates. The logic circuit to be reduced has been shown below.



Boolean Algebra Examples No4

Truth Table of Example No. 4								
A	B	C	A.B.C	$\overline{B}$	$\overline{C}$	$\overline{B} + \overline{C}$	$A.(\overline{B} + \overline{C})$	$Q = ABC + A.(\overline{B} + \overline{C})$
0	0	0	0	1	1	1	0	0
0	0	1	0	1	0	1	0	0
0	1	0	0	0	1	1	0	0
0	1	1	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1
1	0	1	0	1	0	1	1	1
1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	0	0	1

Boolean Algebra Examples No4

The Boolean expression of the given logic circuit has also been reduced below using Boolean algebra laws and theorems.

Equation to be reduced:

$$= ABC + A(\bar{B} + \bar{C})$$

Applying De Morgan's Law: $\bar{B} + \bar{C} = \overline{BC}$

$$= ABC + A(\overline{BC})$$

Factoring (A) out:

$$= A(BC + \overline{BC})$$

Applying Complement Law: $BC + \overline{BC} = 1$

$$= A.1$$

Applying Identity Law: $A.1 = A$

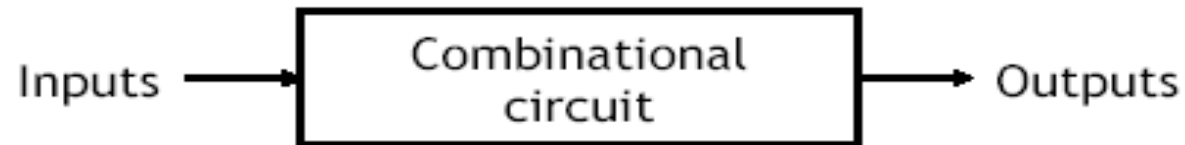
$$= A$$

Combinational Circuits

Introduction

- Introduction to Combinational Circuits
- Adder
- Ripple Carry Adder
- Subtraction
- Adder/Subtractor

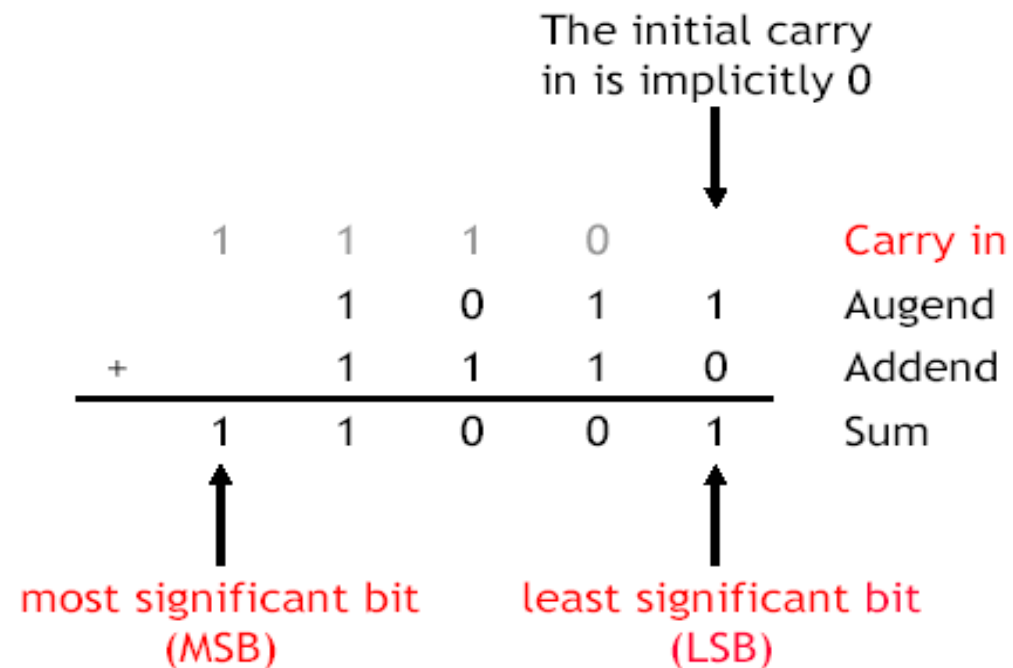
Combinational circuits



- So far we've only worked with **combinational circuits**, where applying the same inputs always produces the same outputs.
 - This corresponds to a mathematical function, where every input has a single, unique output.
 - In programming terminology, combinational circuits are similar to “functional programs” that do not contain variables and assignments.
- Such circuits are comparatively easy to design and analyze.

Binary addition by hand

- You can add two binary numbers one column at a time starting from the right, just like you add two decimal numbers.
- But remember it's binary. For example, $1 + 1 = 10$ and you have to carry!



Adders

Adder

- **Design an Adder for 1-bit numbers?**

- **Specification:**

2 inputs (X,Y)

2 outputs (C,S)

Adder

- Design an Adder for 1-bit numbers?

- **1. Specification:**

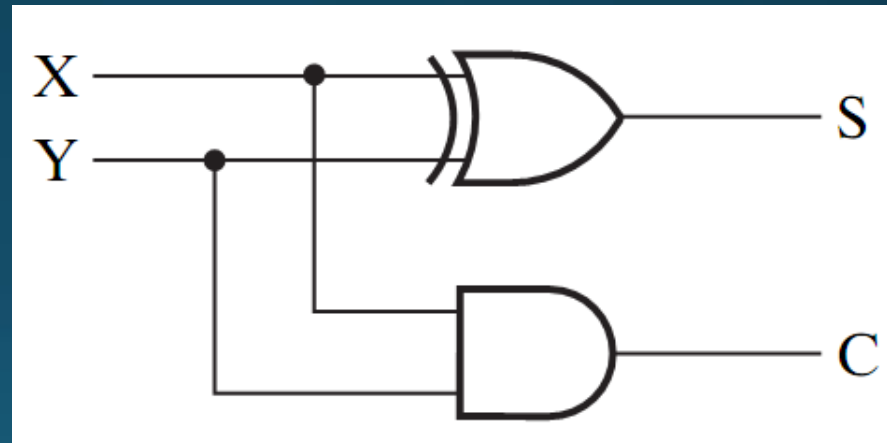
2 inputs (X,Y)

2 outputs (C,S)

- **2. Formulation:**

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

- **3. Optimization/Circuit**



Half Adders

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.

Half Adder Truth Table

Inputs		Outputs	
X	Y	S	C-out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



$$S(X, Y) = \Sigma (1, 2)$$

$$S = X'Y + XY'$$

$$S = X \oplus Y$$

$$C\text{-out}(x, y, C\text{-in}) = \Sigma (3)$$

$$C\text{-out} = XY$$



Full Adders

Full Adder

- A combinational circuit that adds 3 input bits to generate a Sum bit and a Carry bit

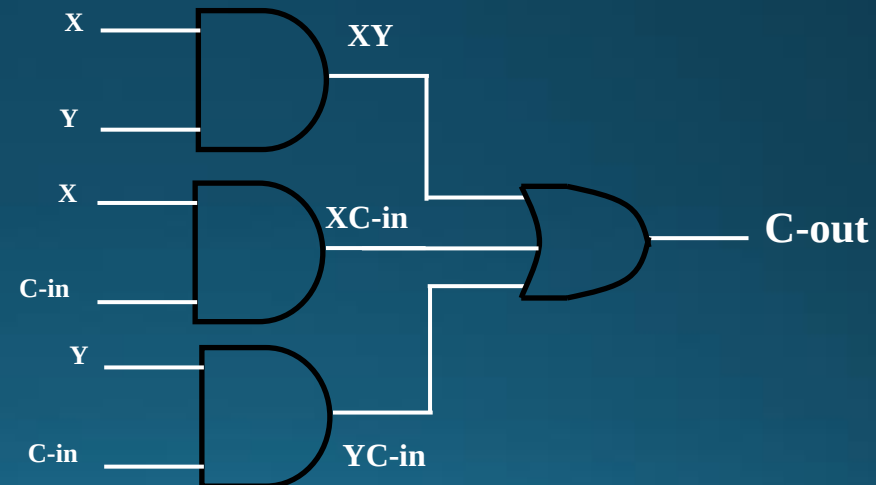
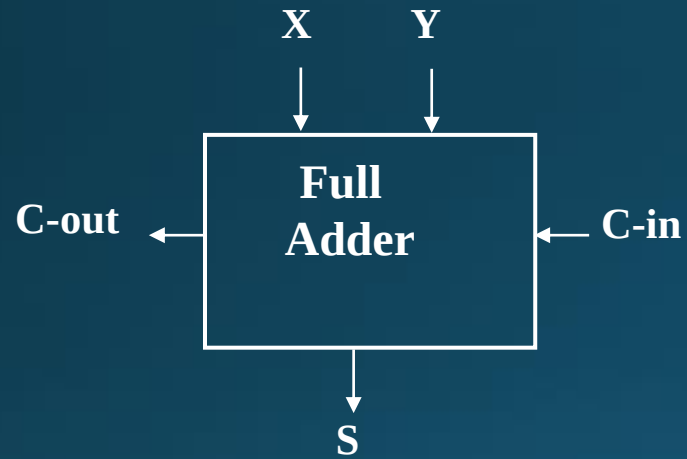
- A truth table and sum of minterm equations for C and S are shown below.

	X	Y	Z	C	S
	0	0	0	0	0
	0	0	1	0	1
	0	1	0	0	1
$0 + 1 + 1 = 10 \rightarrow$	0	1	1	1	0
	1	0	0	0	1
	1	0	1	1	0
	1	1	0	1	0
$1 + 1 + 1 = 11 \rightarrow$	1	1	1	1	1

$$C(X,Y,Z) = \sum m(3,5,6,7)$$

$$S(X,Y,Z) = \sum m(1,2,4,7)$$

Full Adder Circuit Using XOR



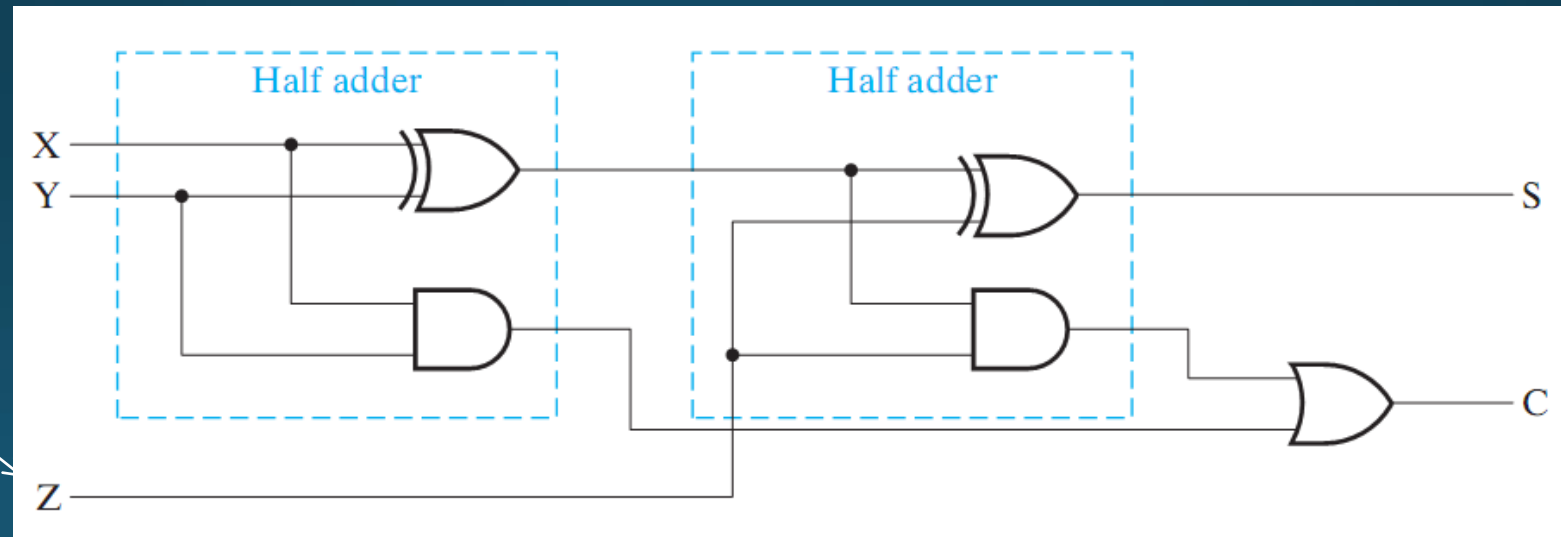
Full Adder = 2 Half Adders

Manipulating the Equations:

$$S = (X \oplus Y) \oplus Z$$

$$C = XY + XZ + YZ = XY + Z(X \oplus Y)$$

Think of
Z as a
carry in



SRC: Mano's BOOK

Bigger Adders

Bigger Adders

- **How to build an adder for n-bit numbers?**
 - Example: 4-Bit Adder
 - Inputs ?
 - Outputs ?
 - What is the size of the truth table?
 - How many functions to optimize?

Bigger Adders

- **How to build an adder for n-bit numbers?**
 - Example: 4-Bit Adder
 - Inputs ? **9 inputs**
 - Outputs ? **5 outputs**
 - What is the size of the truth table? **512 rows!**
 - How many functions to optimize? **5 functions**

Ripple carry Adders

Ripple Carry Adder

- To add n-bit numbers:
- Use n Full-Adders in parallel
- The carries propagates as in addition by hand
- Use Z in the circuit as a C_{in}

$$\begin{array}{r} 1\ 0\ 0\ 0 \\ 0\ 1\ 0\ 1 \\ 0\ 1\ 1\ 0 \\ \hline 1\ 0\ 1\ 1 \end{array}$$

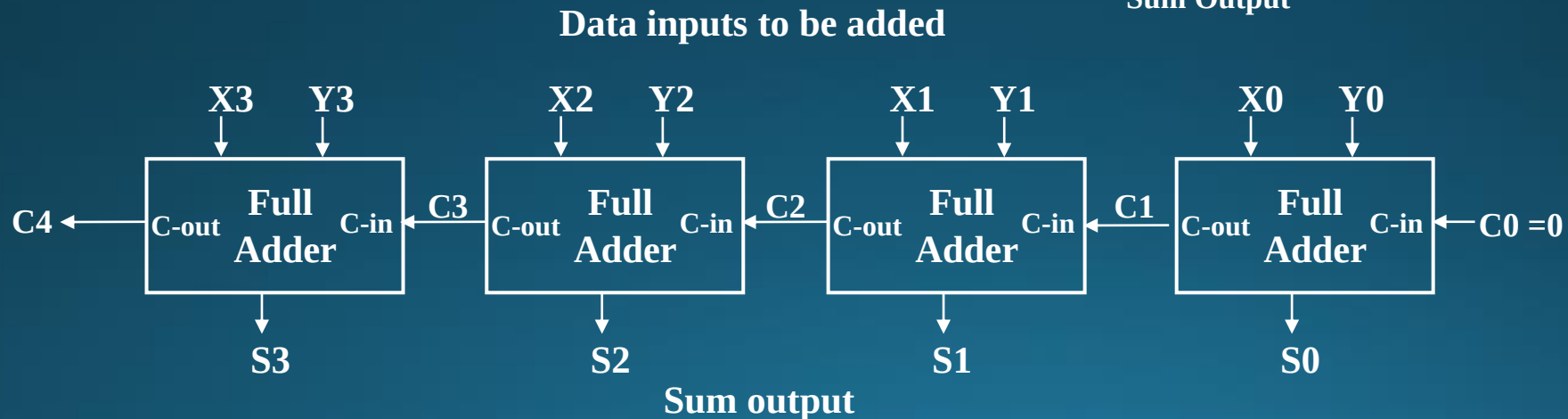
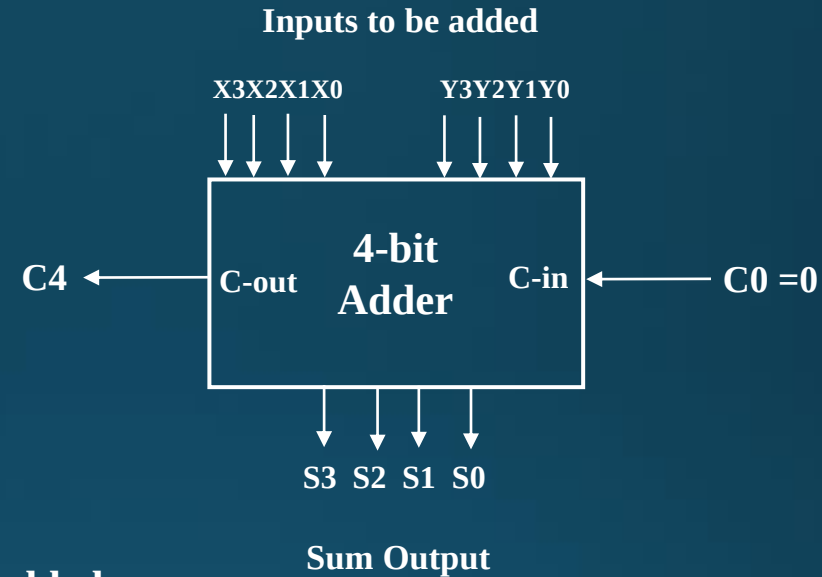
4-bit Carry Ripple Adder

Adds two 4-bit numbers:

$X = X_3 \ X_2 \ X_1 \ X_0$

$Y = Y_3 \ Y_2 \ Y_1 \ Y_0$

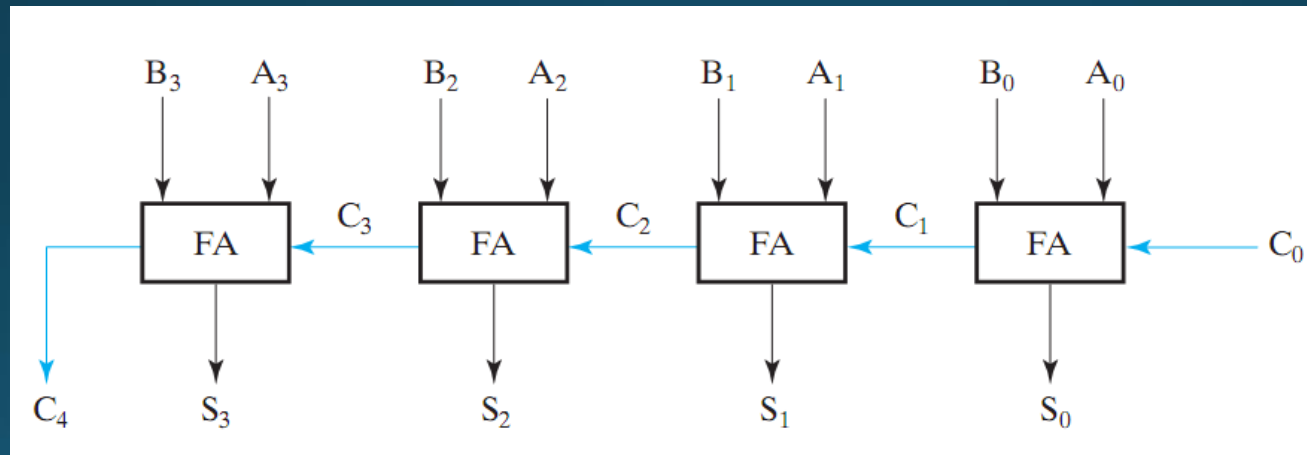
producing the sum $S = S_3 \ S_2 \ S_1 \ S_0$,
 $C\text{-out} = C_4$ from the most significant position $j=3$



Binary Parallel Adders

Binary Parallel Adder

- To add n-bit numbers:
- Use n Full-Adders in parallel
- The carries propagates as in addition by hand



o's Book

This adder is called *ripple carry adder*

Subtractors

Half Subtractor

- Subtracting a single-bit binary value Y from another X (i.e. $X - Y$) produces a difference bit D and a borrow out bit $B\text{-out}$.
- This operation is called half subtraction and the circuit to realize it is called a half subtractor.

Half Subtractor Truth Table

Inputs		Outputs	
X	Y	D	B-out
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



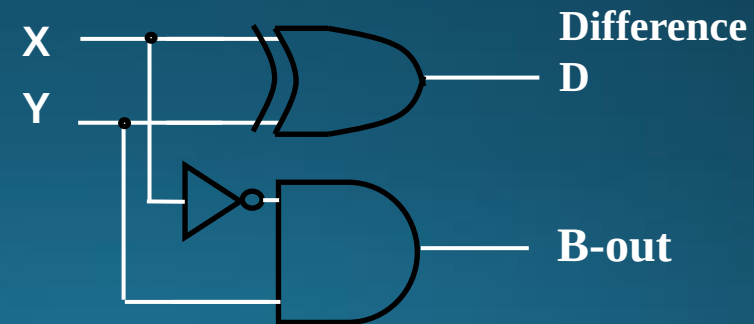
$$D(X, Y) = \Sigma (1, 2)$$

$$D = X'Y + XY'$$

$$D = X \oplus Y$$

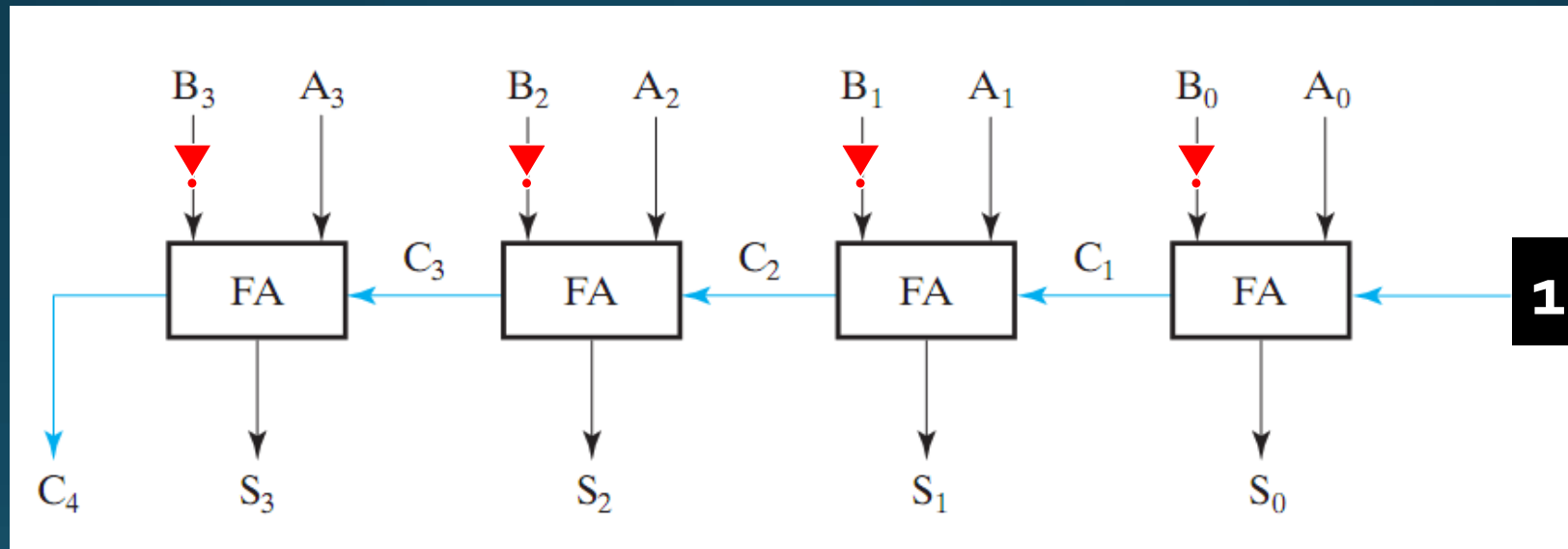
$$B\text{-out}(x, y, C\text{-in}) = \Sigma (1)$$

$$B\text{-out} = X'Y$$



Subtraction (2's Complement)

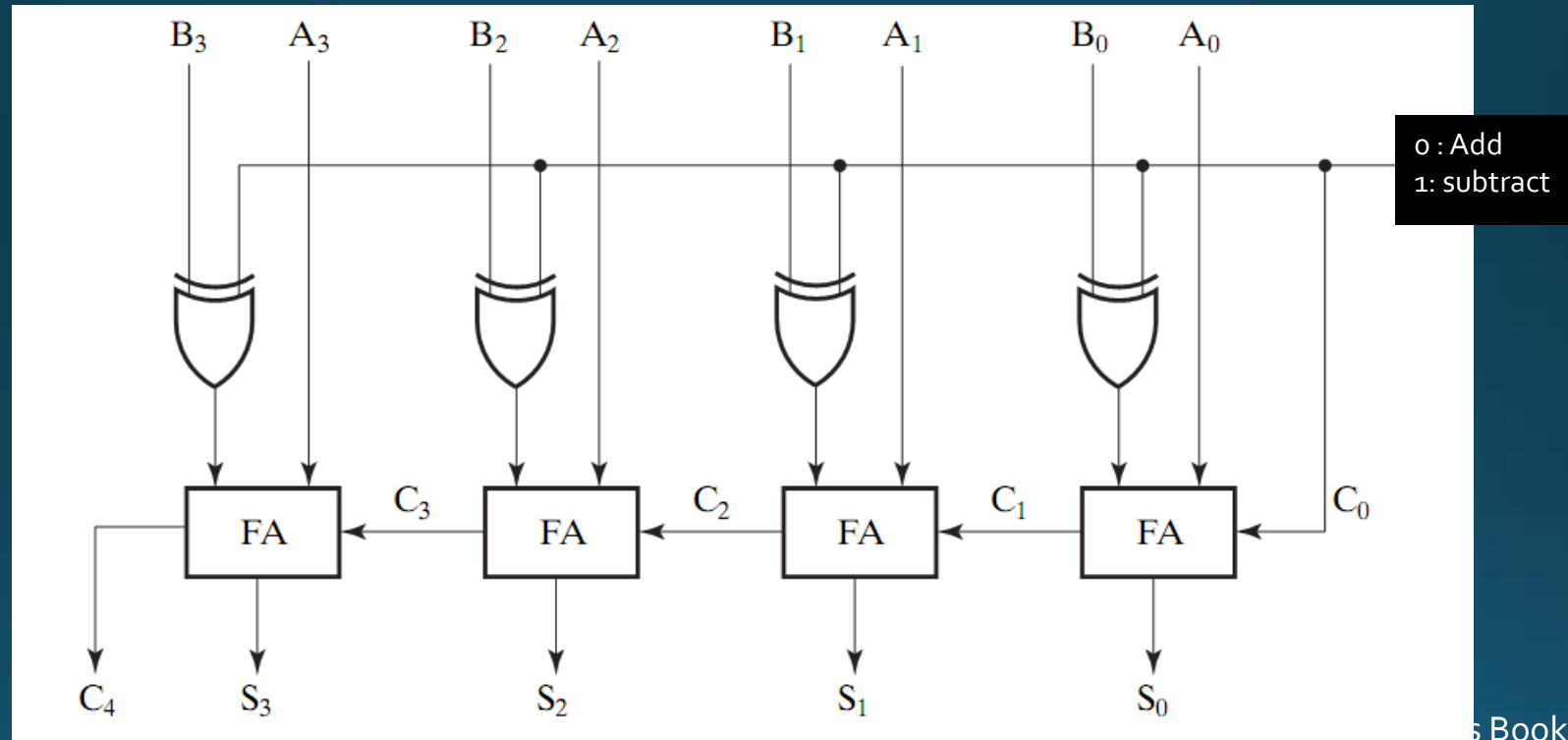
- How to build a subtractor using 2's complement?



Src: Mano's Book

$$S = A + (-B)$$

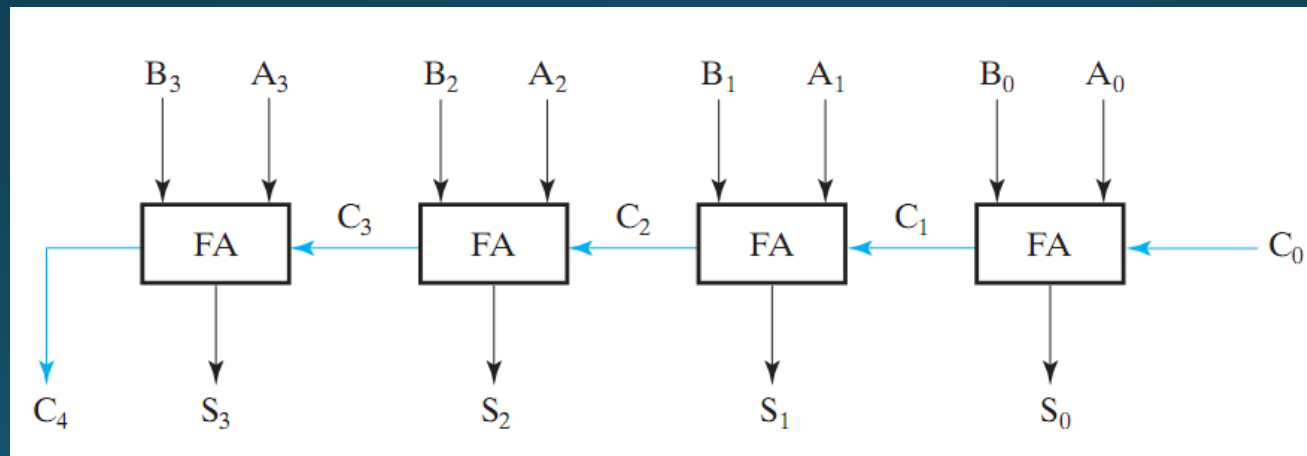
Adder/Subtractor



Using full adders and XOR we can build an Adder/Subtractor!

Binary Parallel Adder (Again)

- To add n-bit numbers:
- Use n Full-Adders in parallel
- The carries propagate as in addition by hand



o's Book

This adder is called *ripple carry adder*



مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY
BY EVERYONE'S SMART UNIVERSITY



Thank You

