CSC 220: Computer Organization

Unit 5
Combinational Circuits

Prepared by:
Md Saiful Islam, PhD

Updated by:
Isra Al-Turaiki, PhD

Department of Computer Science
College of Computer and Information Sciences

Overview

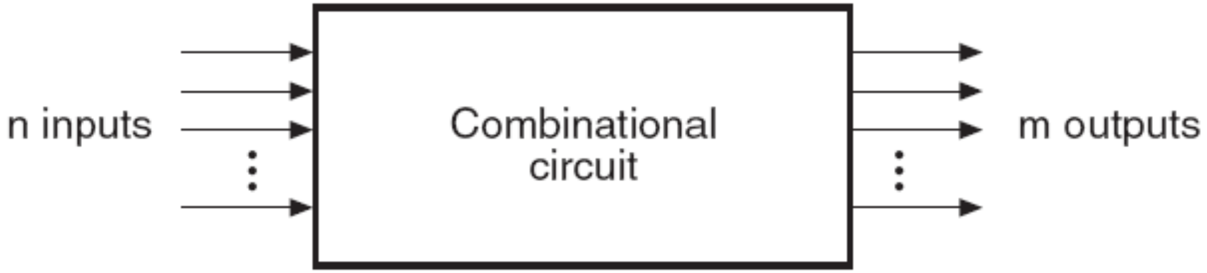
- Introduction to Combinational Circuits
- Adder
- Ripple Carry Adder
- Subtraction
- Adder/Subtractor

Chapter-3

M. Morris Mano, Charles R. Kime and Tom Martin, **Logic and Computer Design Fundamentals**, Global (5th) Edition, Pearson Education Limited, 2016. ISBN: 9781292096124

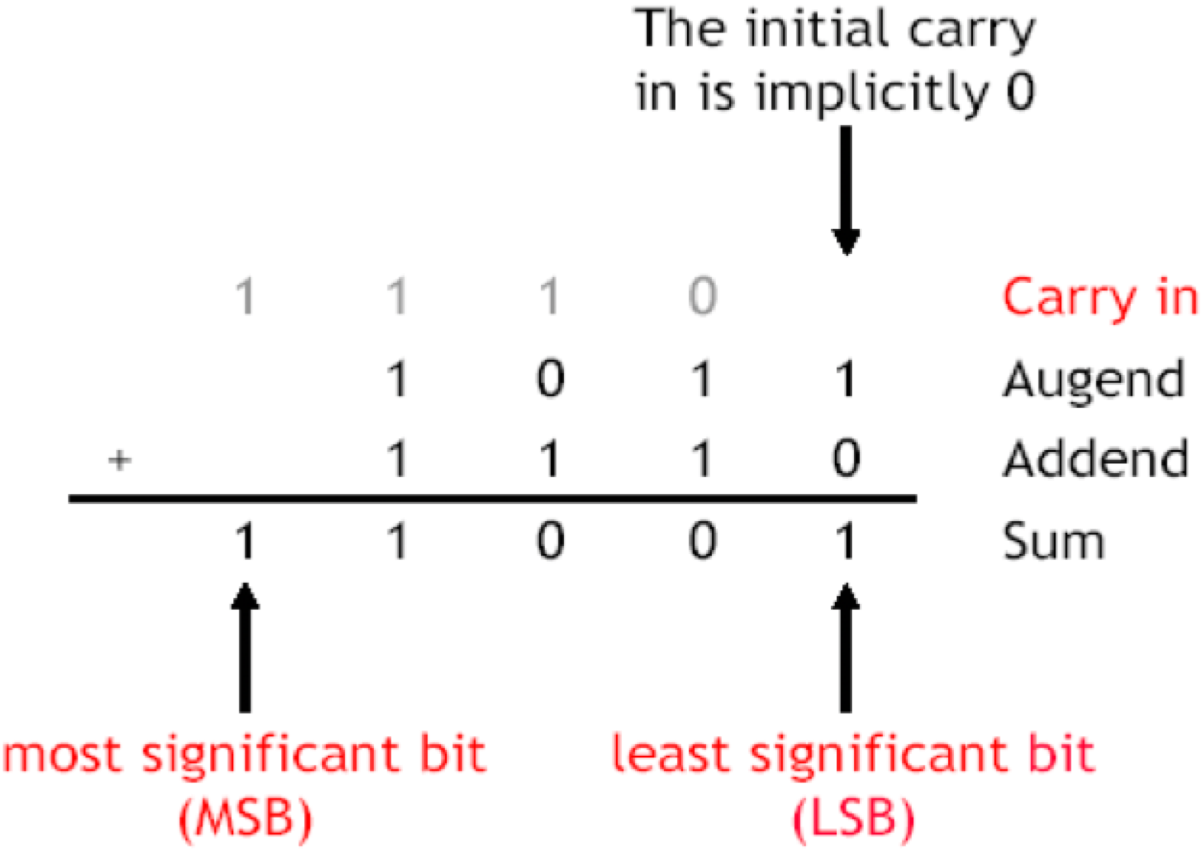
Combinational Circuit

- A combinational logic circuit is a connected arrangement of:
  - A set of n Boolean inputs,
  - A set of m Boolean outputs, and
  - Logic gates and interconnections



Block Diagram of Combinational Circuit

Binary Addition



The initial carry in is implicitly 0

	1	1	1	0		Carry in
		1	0	1	1	Augend
+		1	1	1	0	Addend
	1	1	0	0	1	Sum

most significant bit (MSB)

least significant bit (LSB)

Half 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	1		
1	0		
1	1		

		Y
X		

S=

		Y
X		

C=

Half 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:**

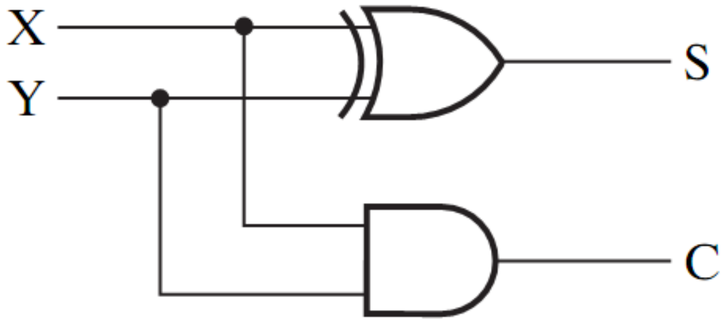
		Y
	$_{0}$	1_{1}
X	1_{2}	$_{3}$

		Y
	$_{0}$	$_{1}$
X	$_{2}$	1_{3}

$S = XY' + X'Y$

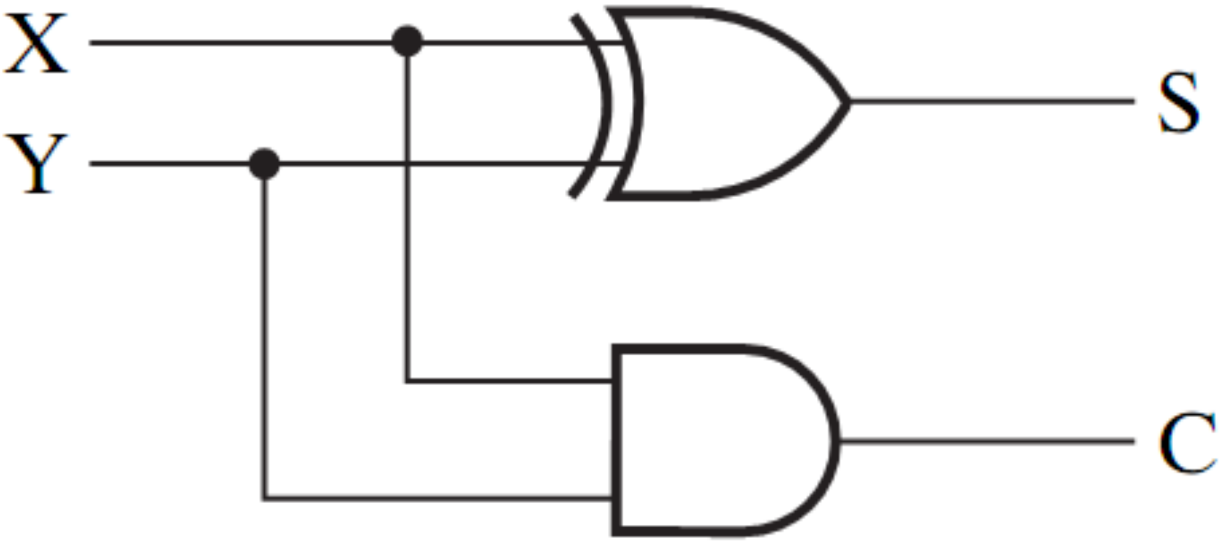
$= X \oplus Y$

$C = XY$



Half Adder

- A combinational circuit that performs the addition of two bits is called a *half adder*.
  - has two outputs: the sum (S) and carry (C)
  - can be implemented with one exclusive-OR gate and one AND gate,



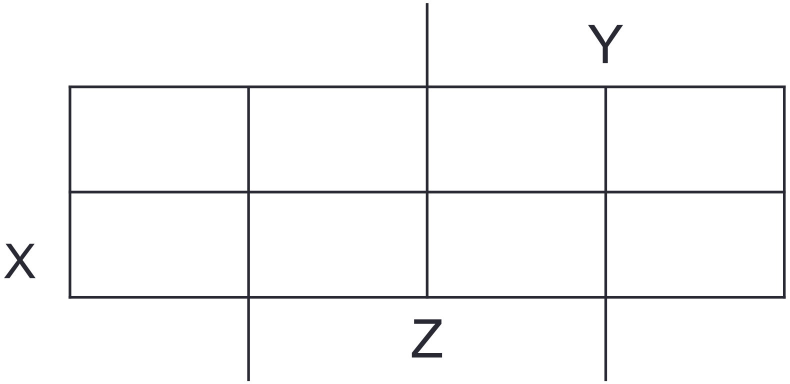
Full Adder

- A *full adder* is a combinational circuit that forms the arithmetic sum of three input bits.
    - has two outputs: the sum (S) and carry (C)
    - Z - carry in (to the current position)
    - C - carry out (to the next position)

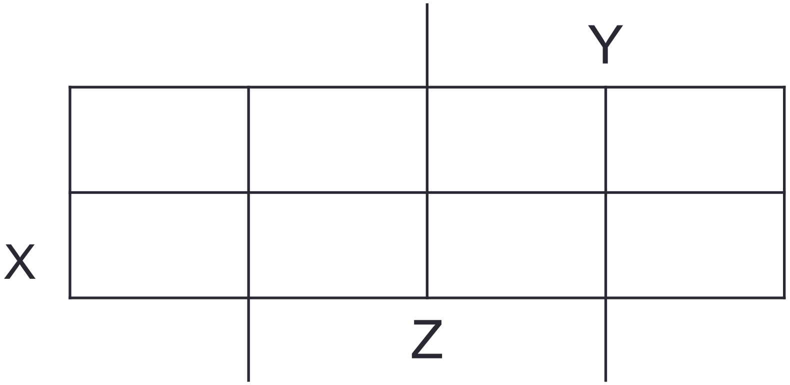
X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Full Adder

X	Y	Z	C	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



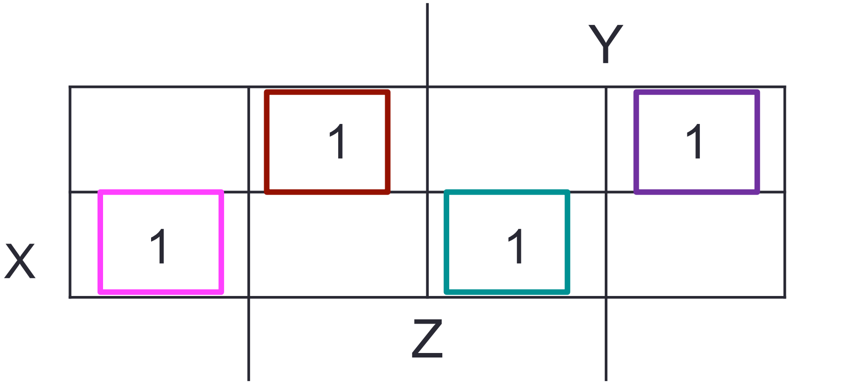
S



C

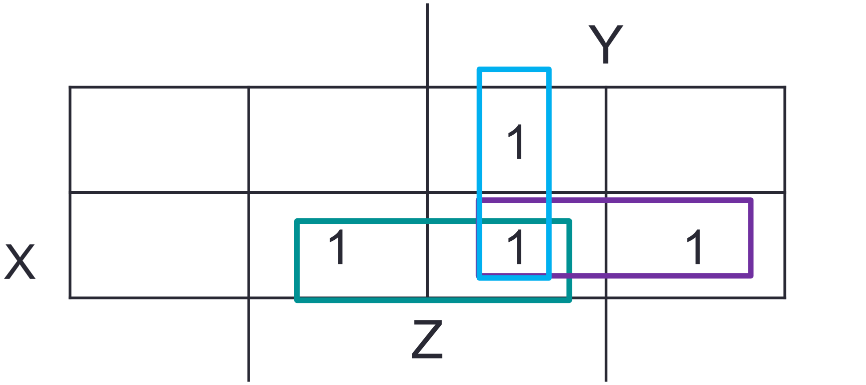
Full Adder

X	Y	Z	C	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



S = XY’Z’+ XYZ +X’Y’Z+X’YZ”

= (X ⊕ Y) ⊕ Z



C = XY + XZ + YZ

= XY + Z(X ⊕ Y)

Full Adder

$$S = XY'Z' + XYZ + X'Y'Z + X'YZ'$$

$$= X(Y'Z' + YZ) + X'(Y'Z + YZ')$$

$$= X(Y \oplus Z)' + X'(Y \oplus Z)$$

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

$$C = XY + XZ + YZ$$

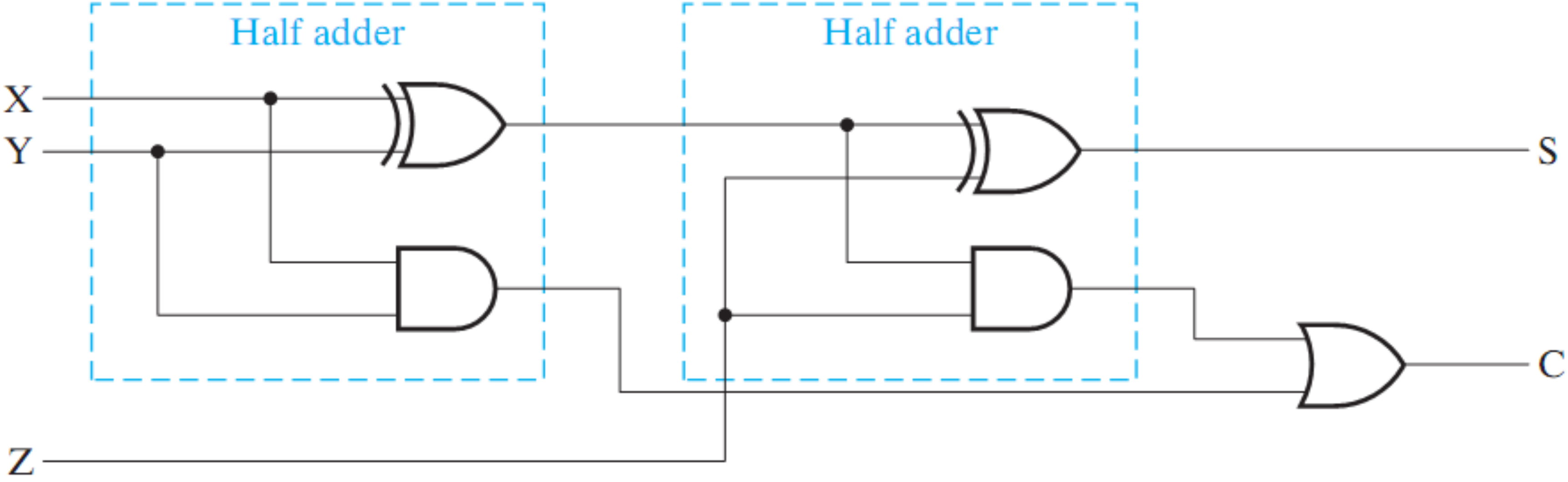
$$= XY + XZ(Y + Y') + YZ(X + X')$$

$$= XY + XYZ + XY'Z + XYZ + X'YZ$$

$$= XY(1 + Z) + Z(XY' + X'Y)$$

$$= XY + Z(X \oplus Y)$$

Full Adder



- It consists of two half adders and an OR gate.
- Think of Z as a carry in

n-bit Adder

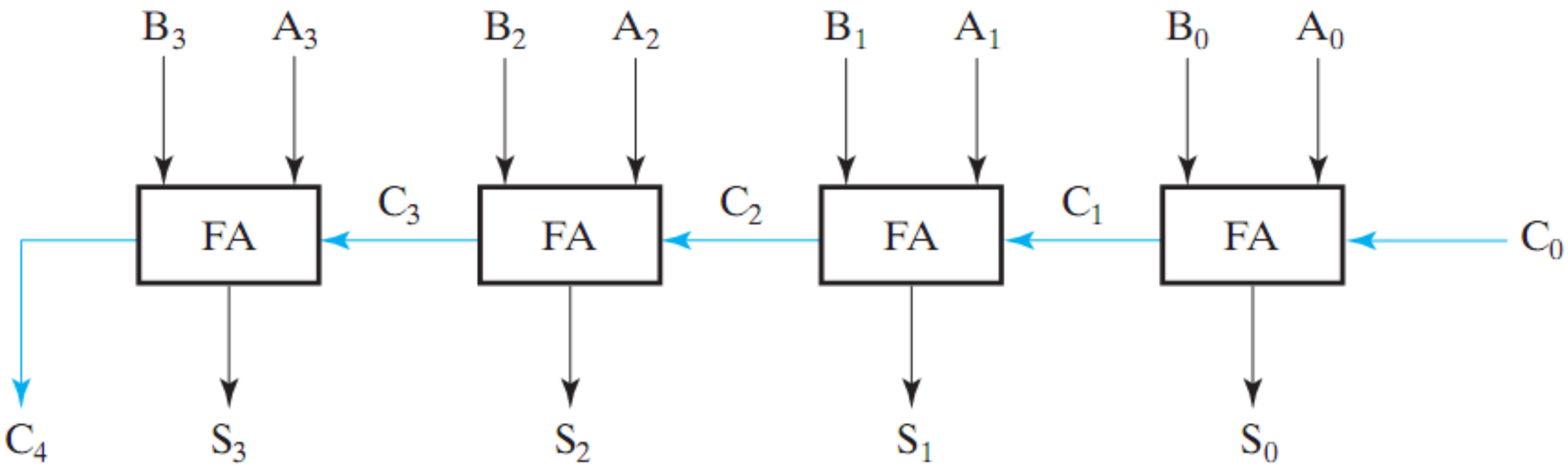
- How to build an adder for n-bit numbers?
  - Example: 4-Bit Adder
    - Inputs ?
    - What is the size of the truth table?
    - Outputs ?
    - How many functions to optimize?

n-bit Adder

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

Binary Parallel Adder

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

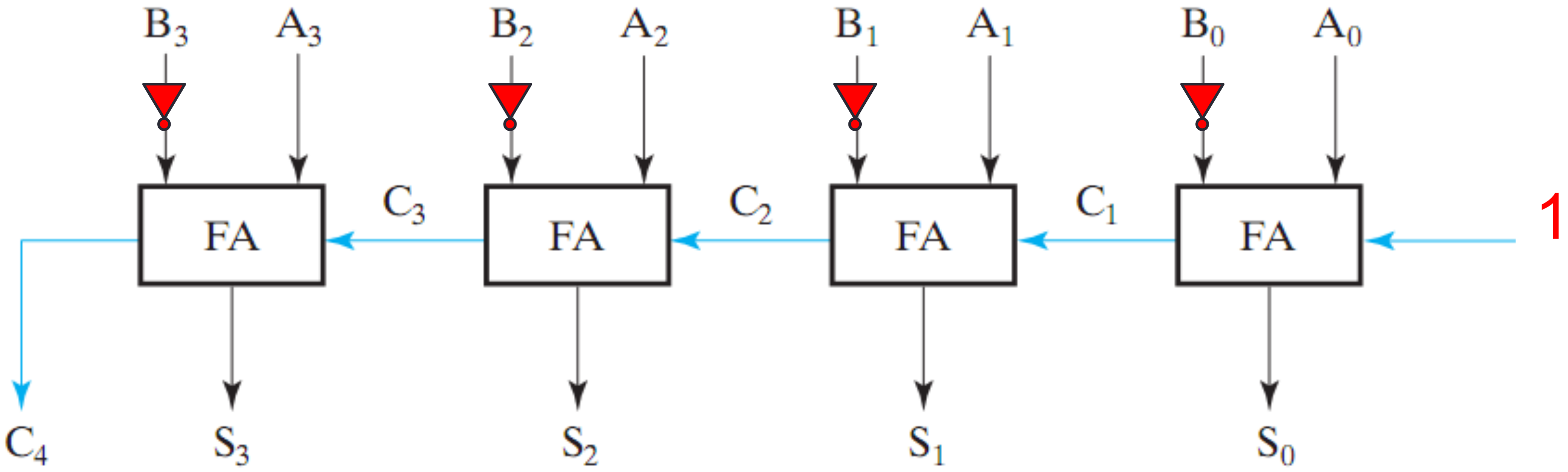


This adder is called ripple carry adder

Subtraction

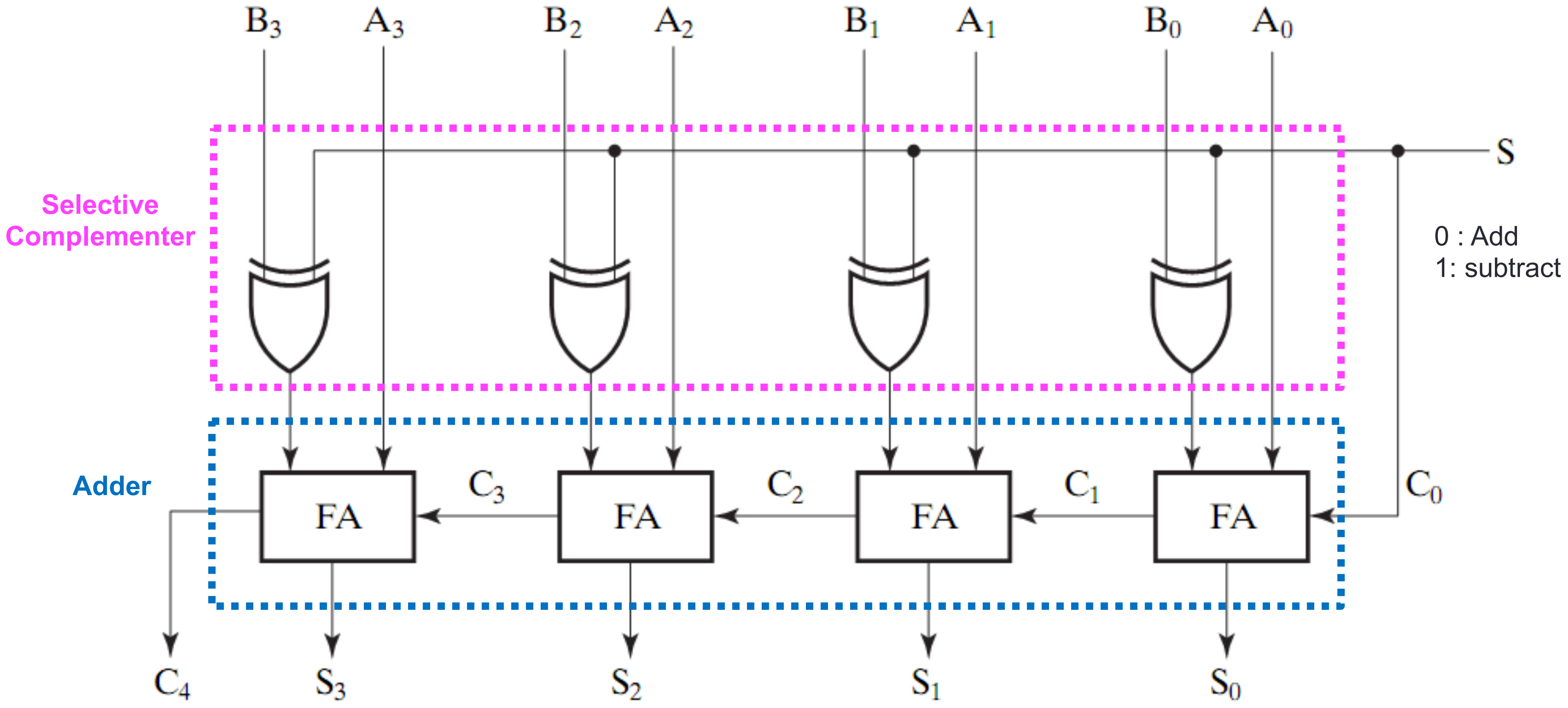
- How to build a subtractor using 2's complement?

$$
\begin{aligned}
S &= A - B \\
&= A + (-B) \\
&= A + 1\text{'s complement of } B + 1
\end{aligned}
$$



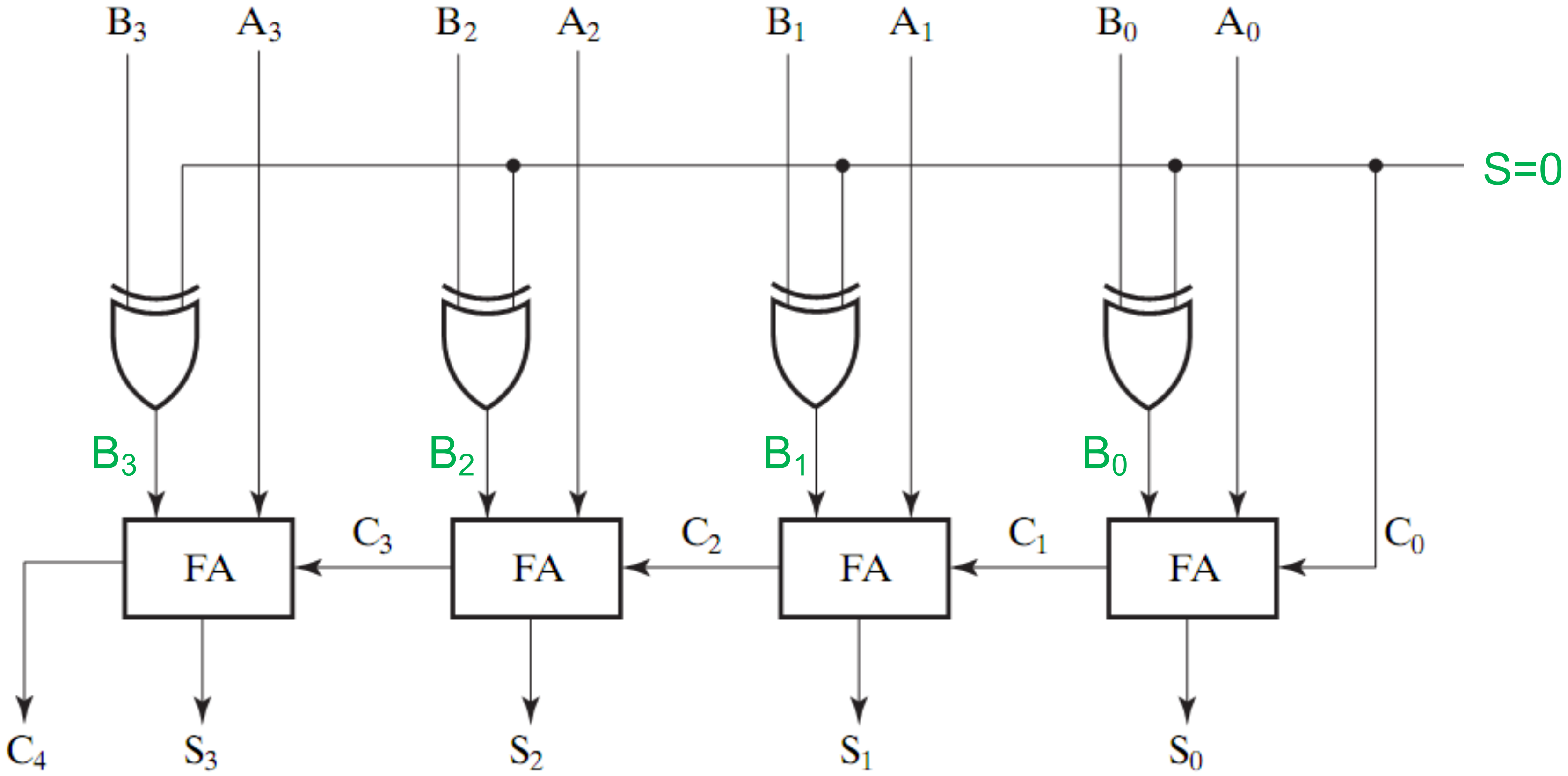
Adder-Subtractor

- How to build a circuit that performs both addition and subtraction?



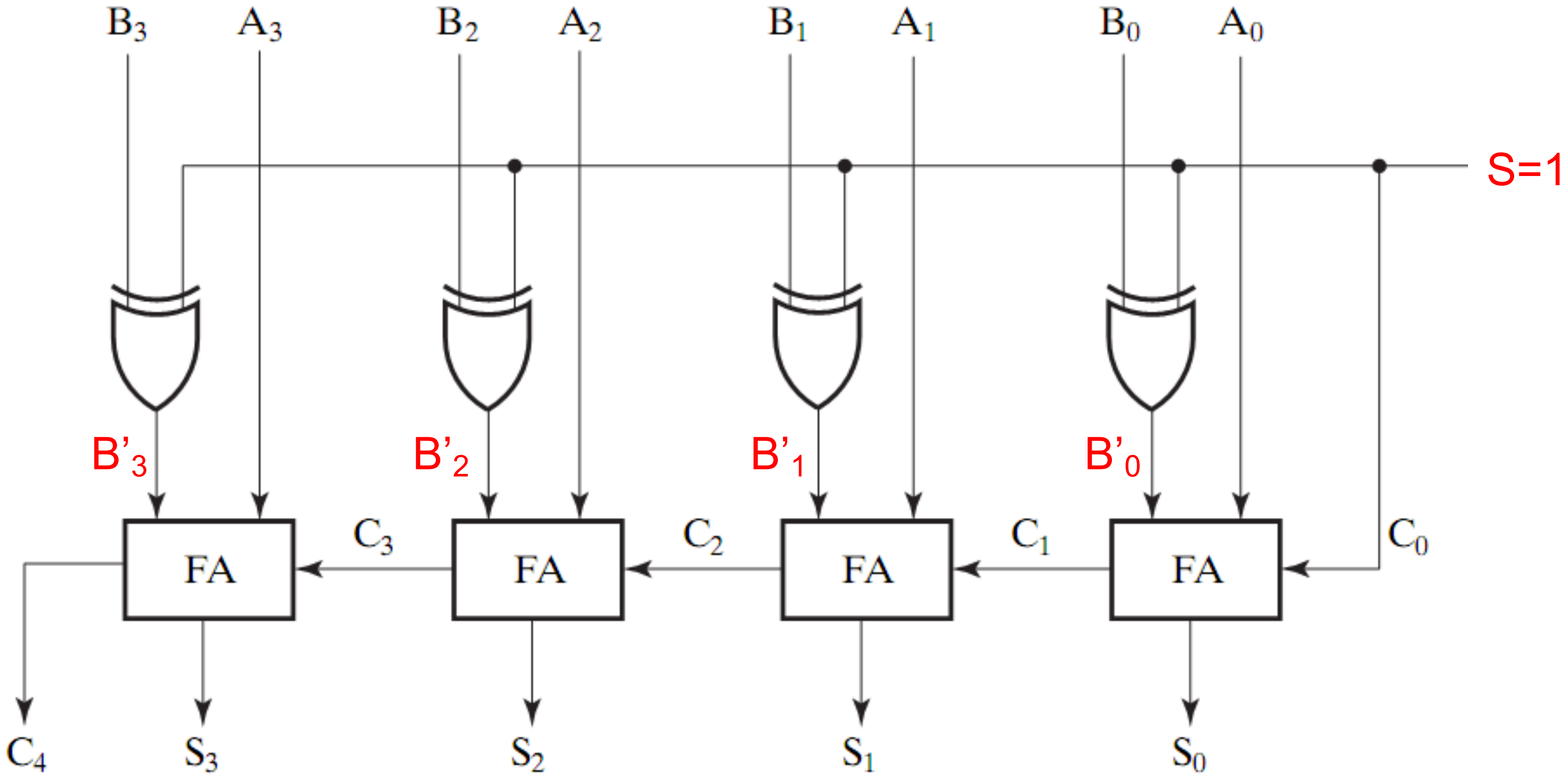
Using full adders and XOR we can build an Adder/Subtractor

Adder-Subtractor



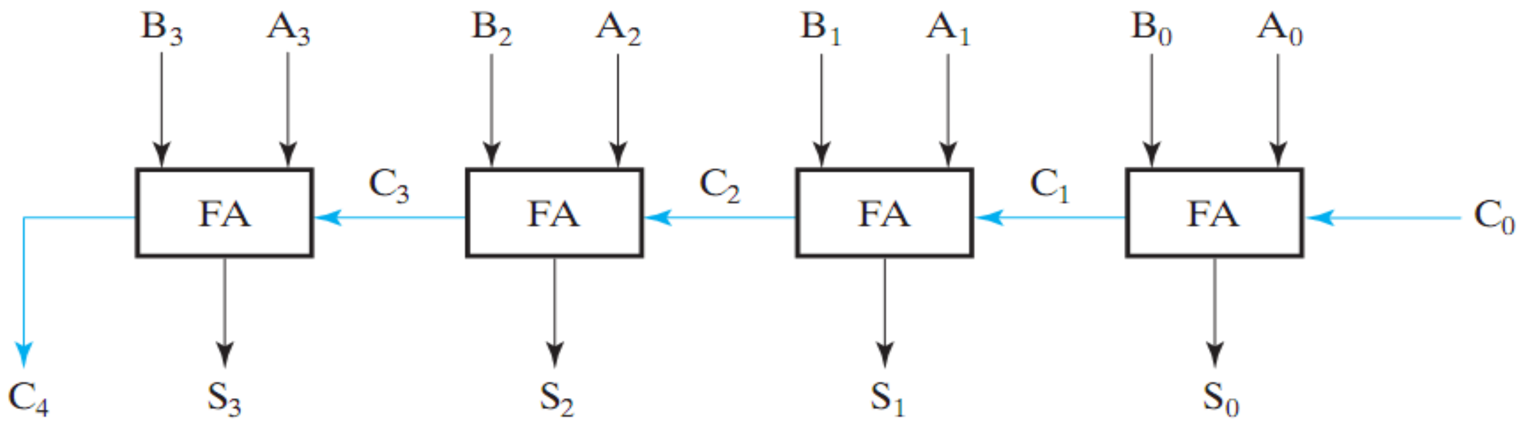
- When S = 0, the circuit performs A + B. The carry in is 0, and the XOR gates simply pass B untouched.

Adder-Subtractor



- When S = 1, the circuit performs A - B. The carry in is 1, and the XOR gates produces the1's complement of B.

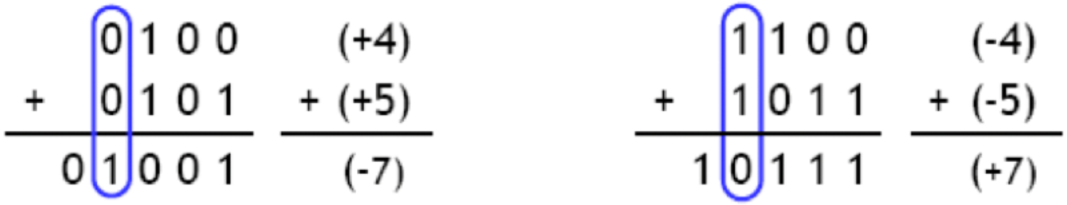
Carry Look Ahead Adder



- How to reduce propagation delay of ripple carry adders?
- Carry look ahead adder: All carries are computed as a function of C_0 (independent of n !)
- It works on the following standard principles:
  - A carry bit is generated when both input bits Ai and Bi are 1, or
  - When one of input bits is 1, and a carry in bit exists

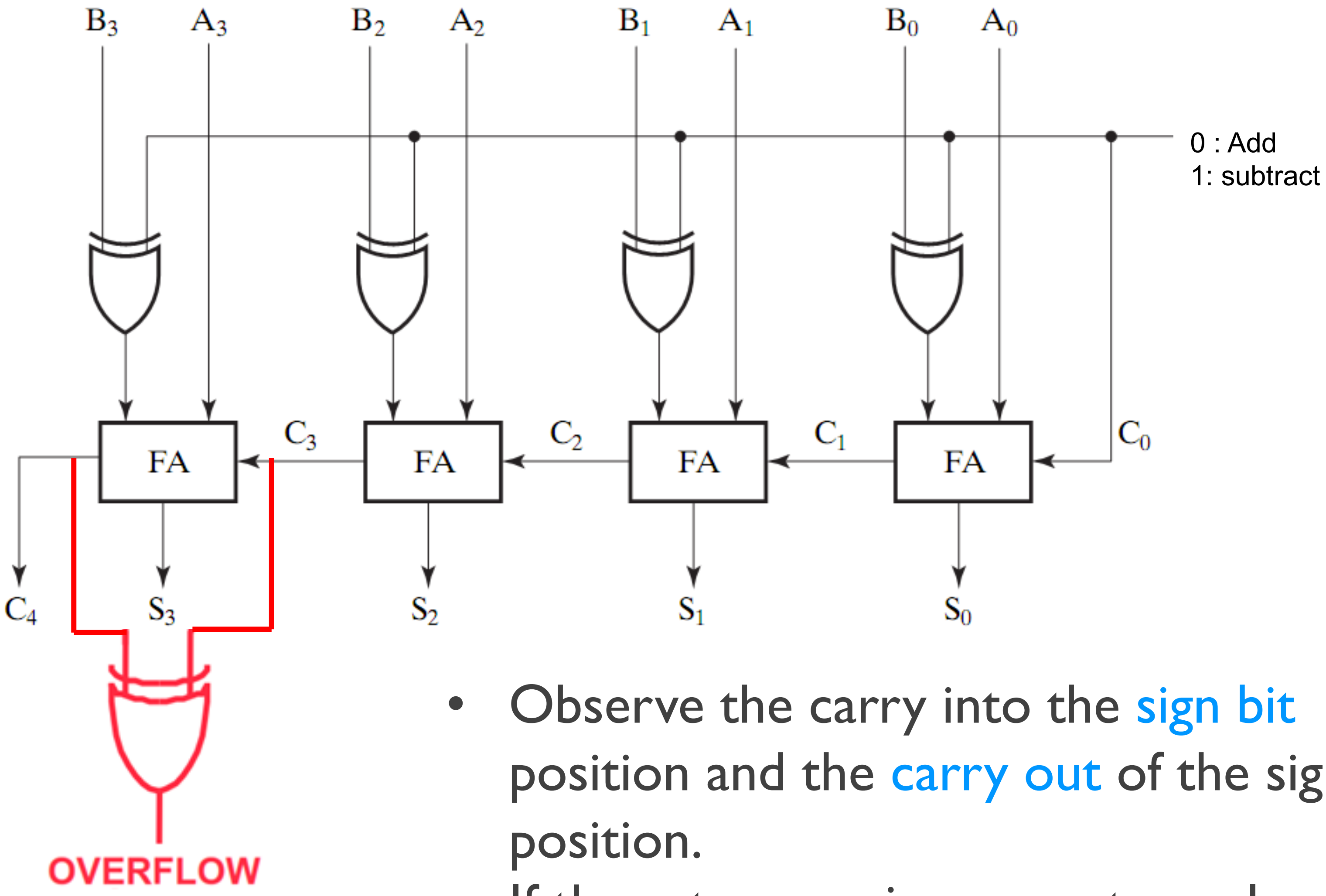
Detecting Overflow

- The easiest way to detect signed overflow is to look at the sign bit.



- An overflow may occur if the two numbers added are both positive or both negative.
  - ❑ If you add two positive numbers and get a negative result.
  - ❑ If you add two negative numbers and get a positive result.

Detecting Overflow



- Observe the carry into the sign bit position and the carry out of the sign bit position.
- If these two carries are not equal, an overflow has occurred