Quaid-e-Awam University of Engineering, Science and Technology


بسم الله الرحمن الرحيم

*In the name of Allah, the Most Merciful, the Most Kind*

Date: 08-11-2021

# BCS 103
# Digital Logic & Computer Architecture

**Lecture 27 and 28**

**Department of Computer Science**

# IN THE LAST LECTURE

We have discussed

**Adders**

- **Half Adder**
- **Full Adder**

**n-bit Ripple Carry Adder**

# IN THE TODAY'S LECTURE

Today we will discuss about:

**Subtractor**

- **Half Subtractor**
- **Full Subtractor**

**Parallel Binary Subtractor**

# Half Subtractor

# Half Subtractor

**TYPES OF BINARY SUBTRACTOR**

- Half Subtractor
- Full Subtractor

**Half Subtractor:**

- It is a combinational circuit with two inputs and two outputs.
- Two inputs are A (minuend), B (subtrahend) and two outputs are D (difference) and $B_o$(borrow out).
- It is used to perform subtraction of two bits.

# Half Subtractor

## RULES FOR BINARY SUBTRACTION

0 - 0 = 0

**0 - 1 = 1** (with borrow 1)

1 - 0 = 1

1 - 1 = 0

**NOTE:** In the second case (0-1) it is necessary to borrow a 1.

# Half Subtractor

**TRUTH TABLE OF HALF-SUBTRACTOR**

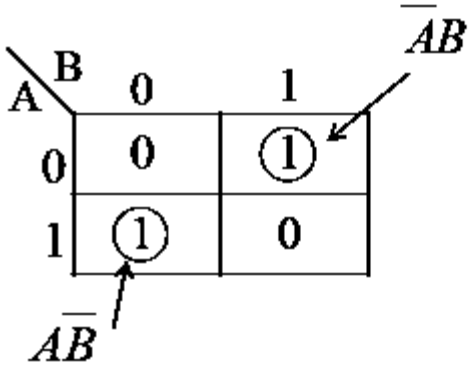
| INPUTS | | OUTPUTS | |
|---|---|---|---|
| Minuend (A) | Subtrahend (B) | Difference (D) | Borrow ($B_o$) |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

# Half Subtractor

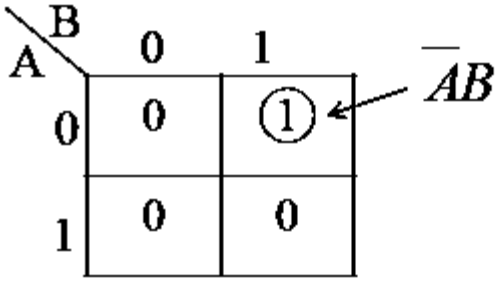
**TRUTH TABLE OF HALF-SUBTRACTOR**

| INPUTS | | OUTPUTS | |
|---|---|---|---|
| Minuend (A) | Subtrahend (B) | Difference (D) | Borrow ($B_o$) |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

K-Map for difference (D)




K-Map for Borrow Output ($B_o$)



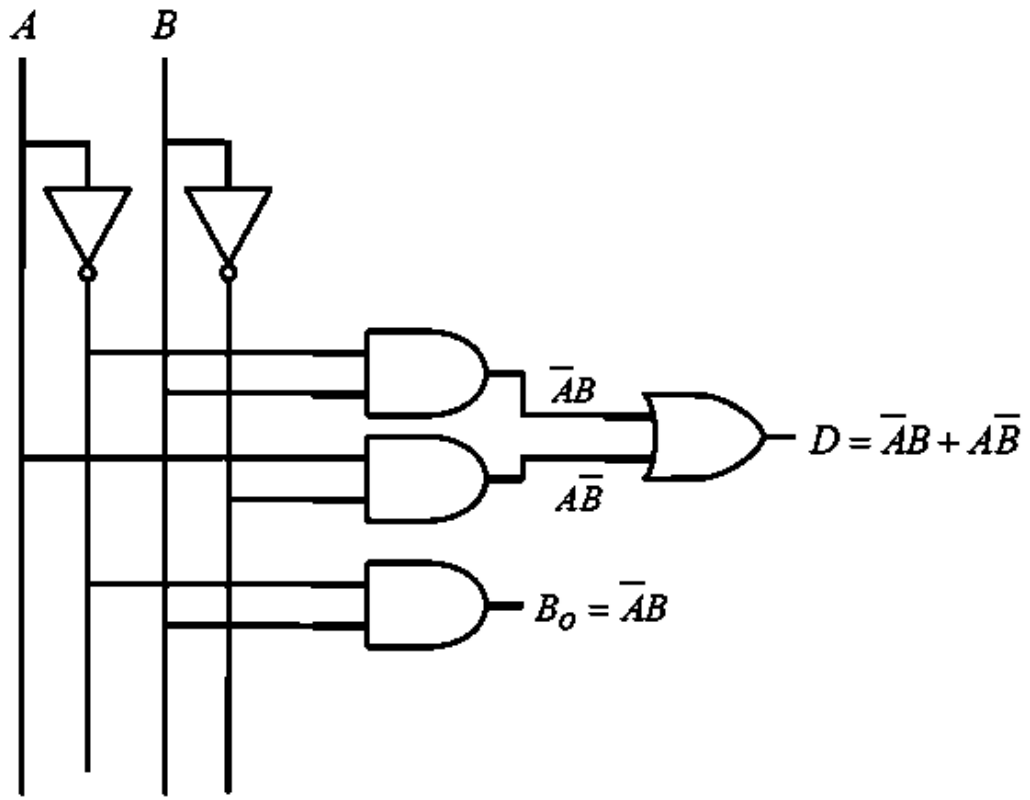
# Half Subtractor

From K-maps

$$D = \overline{A}B + A\overline{B}$$

$$B = \overline{A}B$$

**HALF SUBTRACTOR USING BASIC GATES:**



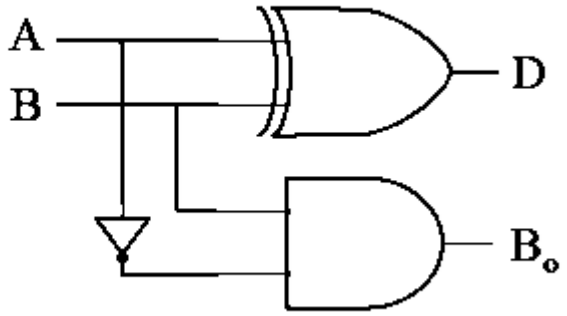
# Half Subtractor

From K-maps

$$D = \overline{A}B + A\overline{B}$$

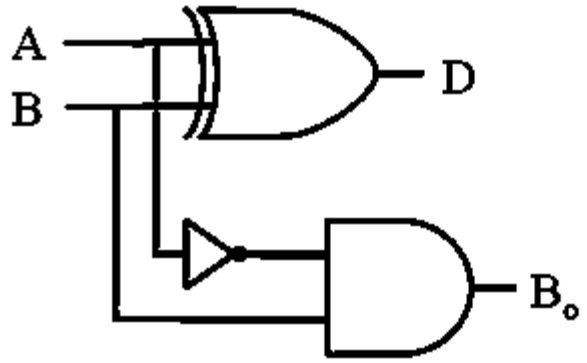
$$B = \overline{A}B$$

**LOGIC DIAGRAM**

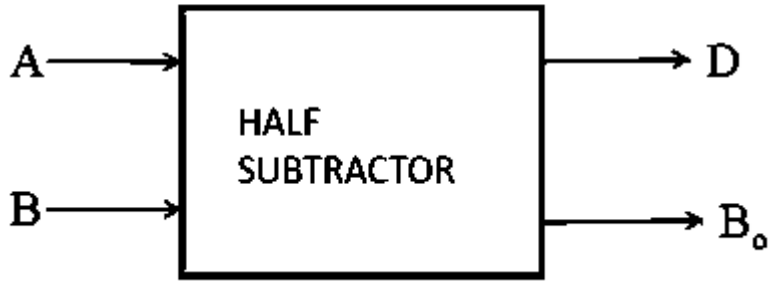


# Half Subtractor

**LOGIC DIAGRAM:**




**BLOCK DIAGRAM:**



# Disadvantage of Half Subtractor:

Half subtractor can only perform the subtraction of two binary bits. But while performing the subtraction, it does not take into account the borrow of the lower significant stage.

# Full Subtractor

# Full Subtractor

- Full subtractor is a combinational circuit.
- It performs subtraction involving three bits (inputs)
  1. Minuend bit (X)
  2. Subtrahend bit (Y)
  3. Borrow from the previous stage ($B_{in}$)

- It has two outputs
  1. Difference (D)
  2. Borrow out $B_{out}$.

# Full Subtractor

**TRUTH TABLE:**

| INPUTS | | | OUTPUTS | |
|---|---|---|---|---|
| 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 |

# Full Subtractor

**TRUTH TABLE:**

| INPUTS | | | OUTPUTS | |
|---|---|---|---|---|
| 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 |

**K-Map For Difference Output (D)**

| X \ $YB_{in}$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | (1) | 0 | (1) |
| 1 | (1) | 0 | (1) | 0 |

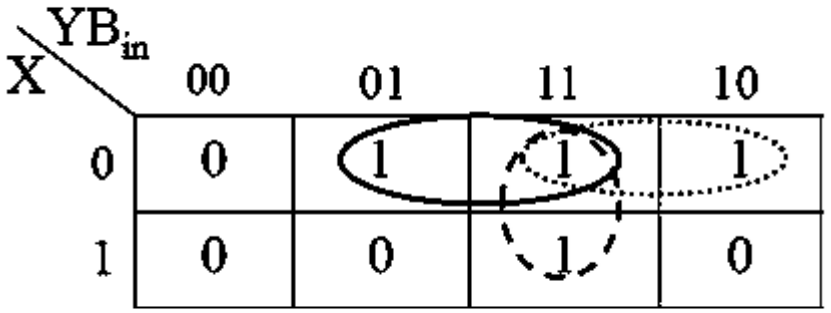
$$D = \overline{X}\,\overline{Y}B_{in} + \overline{X}Y\overline{B_{in}} + X\overline{Y}\,\overline{B_{in}} + XYB_{in}$$

# Full Subtractor

**TRUTH TABLE:**

| INPUTS | | | OUTPUTS | |
|---|---|---|---|---|
| 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 |

**K-Maps For Borrow Output ($B_{out}$)**




$$B_{out} = \overline{X}Y + \overline{X}B_{in} + YB_{in}$$

# Full Subtractor

$$D = \overline{X}\,\overline{Y}B_{in} + \overline{X}Y\overline{B_{in}} + X\overline{Y}\overline{B_{in}} + XYB_{in}$$

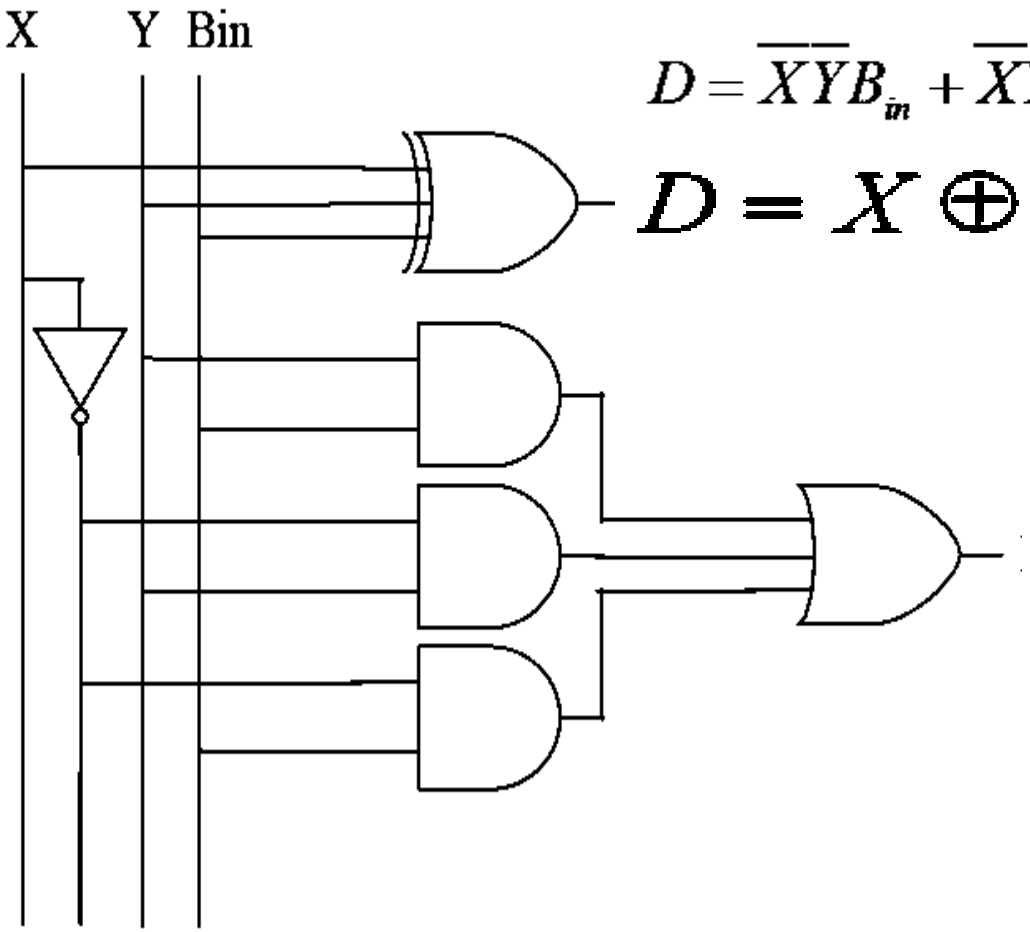
$$D = (\overline{X}\,\overline{Y} + XY)B_{in} + (\overline{X}Y + X\overline{Y})\overline{B_{in}}$$

$$D = (\overline{X \oplus Y})B_{in} + (X \oplus Y)\overline{B_{in}}$$

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

# Full Subtractor

**RELIZATION OF FULL SUBTRACTOR:**



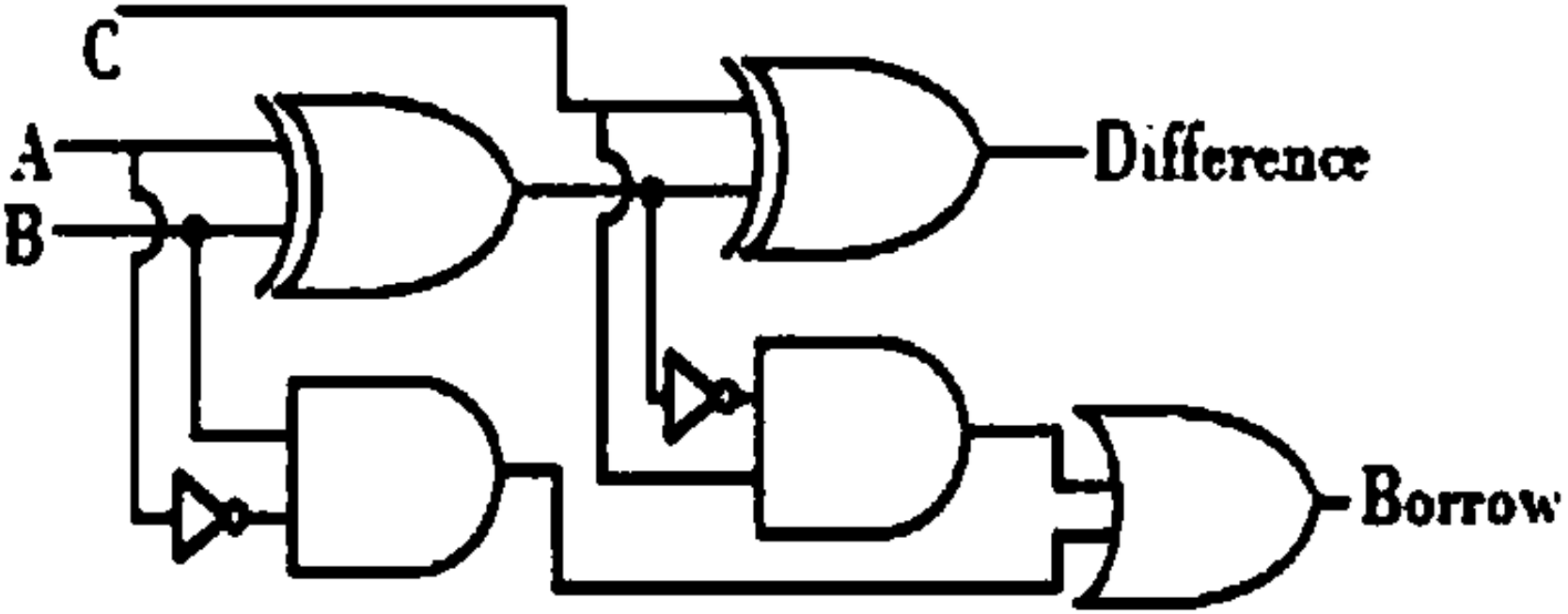

$$D = \overline{X}\,\overline{Y}B_{in} + \overline{X}Y\overline{B_{in}} + X\overline{Y}\,\overline{B_{in}} + XYB_{in}$$

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

$$B_{out} = \overline{X}Y + \overline{X}B_{in} + YB_{in}$$

# Full Subtractor

$$D = A \oplus B \oplus C$$



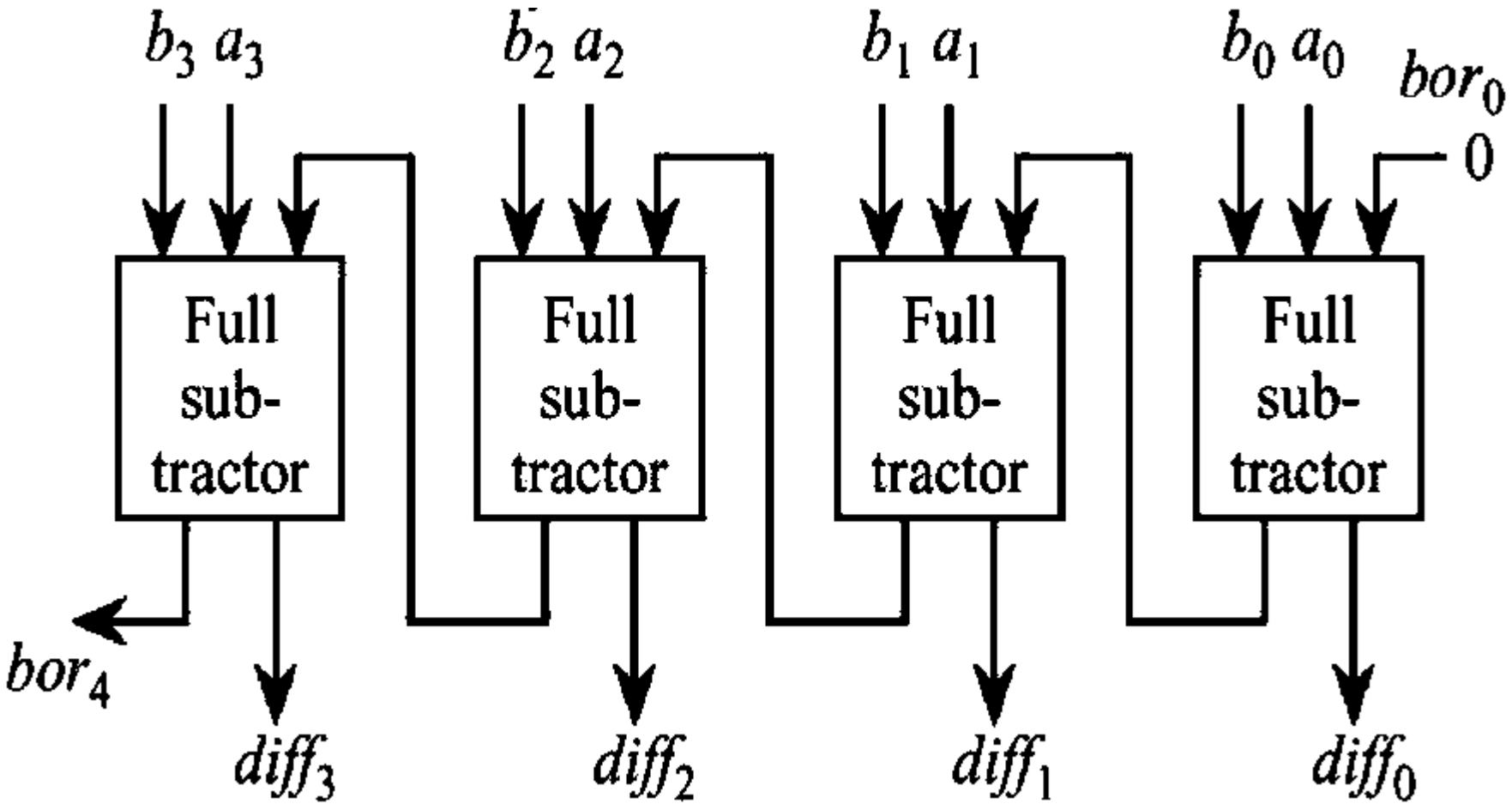

$$B_{(out)} = BC + (B \oplus C)A$$

# Parallel Binary Subtractor

Parallel binary subtractor can be implemented by cascading several full-subtractors.

Next slide shows the block level representation of a 4-bit parallel binary subtractor, which subtracts 4-bit $b_3b_2b_1b_0$ from 4-bit $a_3a_2a_1a_0$. It has 4-bit difference output $D_3D_2D_1D_0$ with borrow output B(out).

# Parallel Binary Subtractor



# Thanks