

ECE119 – Ψηφιακή Σχεδίαση

Διδάσκοντες Εργαστηρίου: Δ. Καραμπερόπουλος
Α. Φεύγας

- Lab 5: Binary Conversion and Adders

Περιεχόμενα Εργαστηριακού Μαθήματος



- Εισαγωγή
- Lab 1: Multisim Circuit Simulation and Basic Gates
- Lab 2: Truth Tables and Basic Logic Gates
- Lab 3: Logic Gates Explored and Boolean Algebra
- Lab 4: Karnaugh Maps
- **Lab 5: Binary Conversion and Adders**
- Lab 6: Encoders and Decoders
- Lab 7: Multiplexers and Demultiplexers
- Lab 8: Latches and Sequential Logic Circuits
- Lab 9: Flip-Flops
- Lab 10: Sequential Circuits – FSM (1ο)
- Lab 11: Sequential Circuits – FSM (2ο)

Binary Conversion and Adders



In the first lab, you explored simple logic circuits with two binary inputs, using truth tables to describe their behavior.

As digital circuits become more complex, the number of possible input combinations increases rapidly, making truth-table-based design less practical.

In this lab, you will explore two fundamental concepts:

Binary and BCD Representation

How numbers are represented in digital systems and how binary values can be converted into Binary-Coded Decimal (BCD).

Binary Adders

How combinational logic circuits perform binary addition.

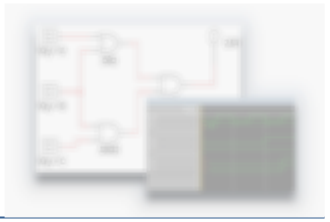
Learning Objectives



In this lab, students will:

- Construct half and full adders with logic gates and create truth tables from them
- Confirm the truth table for a full adder.
- Design and simulate a 4-bit adder-subtractor in Verilog, and interpret its results for signed and unsigned arithmetic using the carry-out and overflow flags.

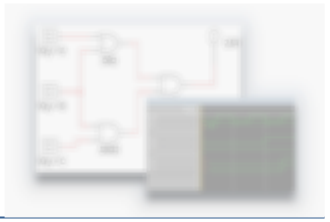
Expected Deliverables



In this lab, you will submit the following:

- Answers to the theory and conclusion questions.
- Completed truth tables.
- Multisim circuit files.
- Verilog HDL files for the Half Adder, Full Adder, and 4-bit Adder-Subtractor.
- Verilog testbench for the 4-bit Adder-Subtractor.
- Completed simulation results table (S, C, V).
- Answers to the simulation evaluation questions.

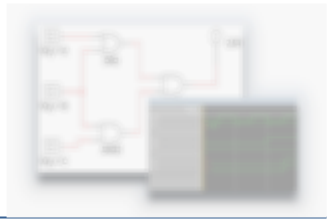
Binary-Coded Decimals



- Δυαδικά κωδικοποιημένοι δεκαδικοί - BCD
- Η κάθε ομάδα των 4 bit αντιστοιχεί σε ένα δεκαδικό ψηφίο
- π.χ. Ο δεκαδικός $(396)_{10}$ αναπαρίσταται σε BCD με 12 bit ως $(0011\ 1001\ 0110)_{BCD}$
- Οι δυαδικοί συνδυασμοί 1010 ως 1111 δεν χρησιμοποιούνται στο σύστημα BCD και επομένως, δεν έχουν νόημα στον κώδικα αυτό.

- $(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (10111001)_2$

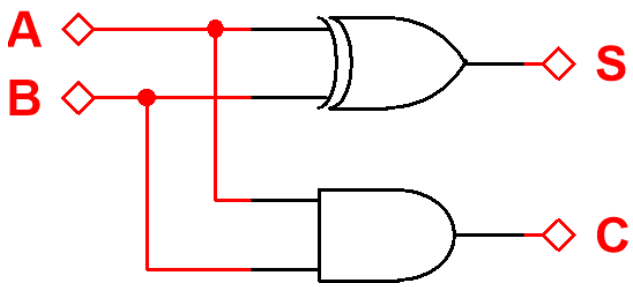
Half-Adder

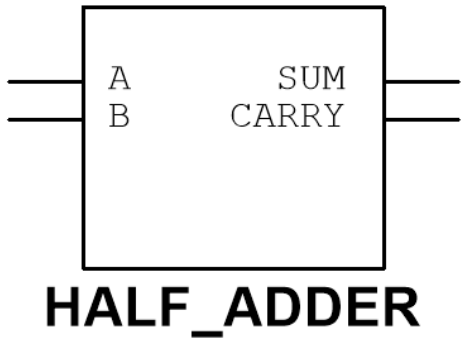


- A half-adder does binary addition on two inputs (A and B).
- The two outputs are labeled sum (S) and carry (C).
- Half adders can be built with:
  - an XOR gate and an AND gate (shown on the left).
  - a component in Multisim (shown on the right).

Half Adder

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



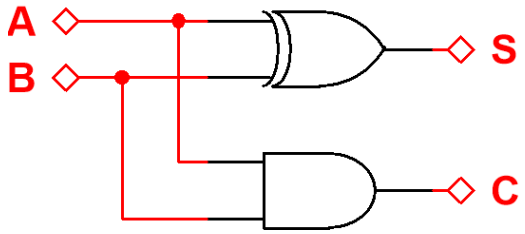


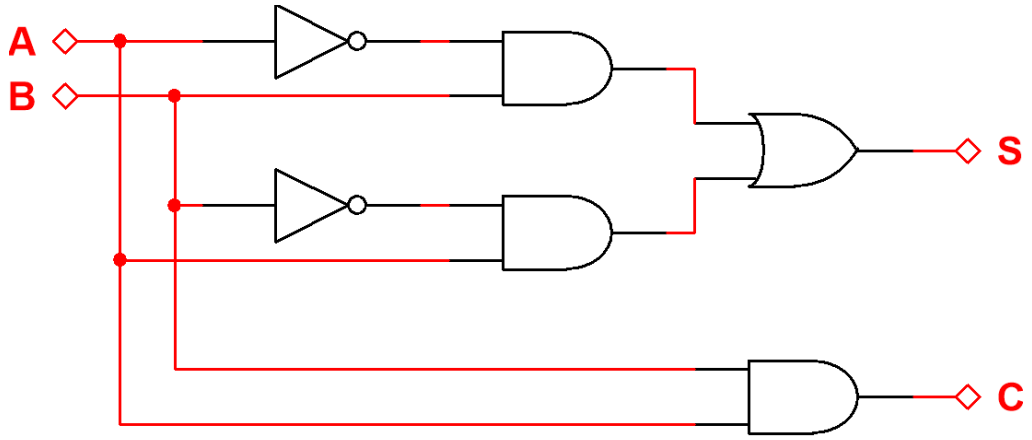
Half-Adder

- Το κύκλωμα που πραγματοποιεί την πρόσθεση δυο ψηφίων χωρίς να λαμβάνει υπόψη τυχόν προηγούμενο κρατούμενο ονομάζεται **Ημιαθροιστής**.
- Ο Ημιαθροιστής έχει δυο εισόδους x και y (τα bit που προστίθενται) και δυο εξόδους C (κρατούμενο-carry) και S (άθροισμα-sum).

Half Adder

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0





$$S = A \oplus B$$
$$C = A \cdot B$$

$$S = \overline{A} \cdot B + A \cdot \overline{B}$$
$$C = A \cdot B$$

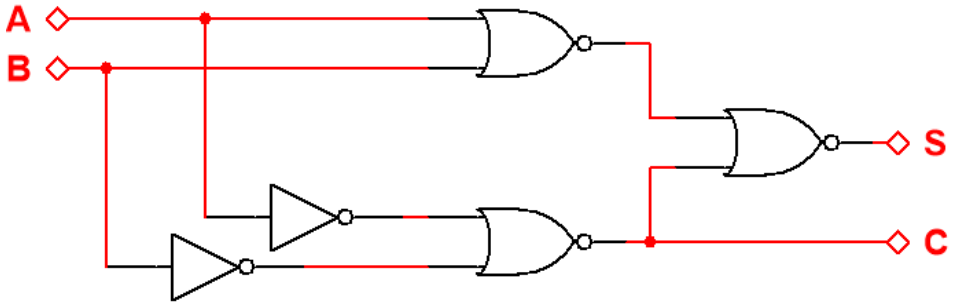
Half-Adder – Με πύλες NOR και NAND

- Με πύλες NOR.

$$S = (A+B)(\overline{A}+\overline{B}) = \overline{\overline{A+B} + \overline{\overline{A}+\overline{B}}} \quad (De\ Morgan)$$

$$C = A \cdot B = \overline{\overline{A}+\overline{B}}$$

Ημιαθροιστής με **NOR** πύλες



A	B	S	C
0	0	**0**	**0**
0	1	**1**	**0**
1	0	**1**	**0**
1	1	**0**	**1**

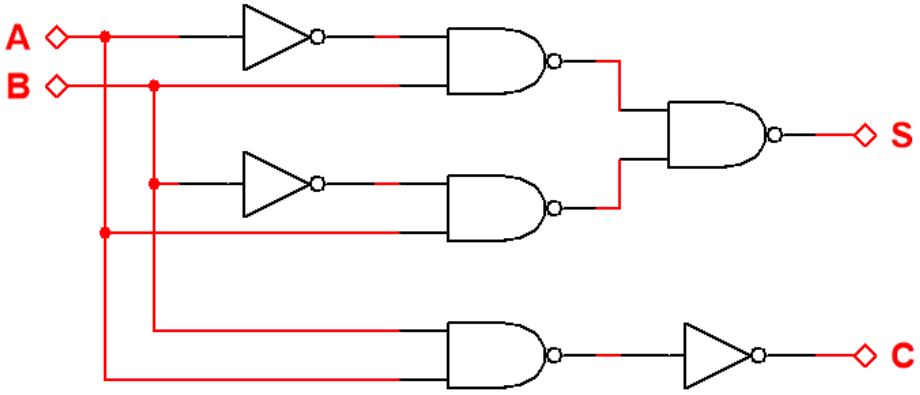
  - Συνεπώς η **S** μπορεί να δημιουργηθεί χρησιμοποιώντας μόνο πύλες NOR, η δε **C** προκύπτει από μία εξ αυτών των πυλών.

- Με πύλες NAND.

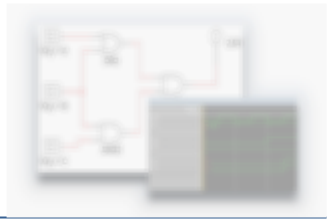
$$S = \overline{A}\cdot B + A\cdot\overline{B} = \overline{(\overline{\overline{A}\cdot B}) \cdot (\overline{A\cdot\overline{B}})} \quad (De\ Morgan)$$

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

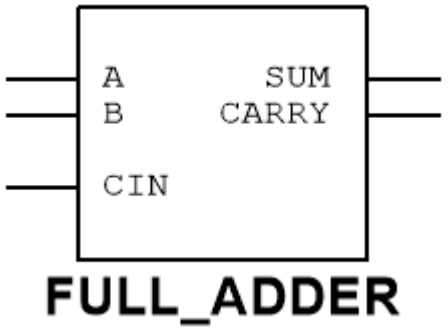
Ημιαθροιστής με **NAND** πύλες



Full-Adder



- A full-adder does binary addition on three inputs: A, B, and Cin.
- Full adders usually work in a cascade fashion where they are used to add binary numbers with an increasing number of bits.
- The two outputs are sum (S) and carry (Cout).
- You will notice that full adders can also use logic gates or a component.



Full Adder

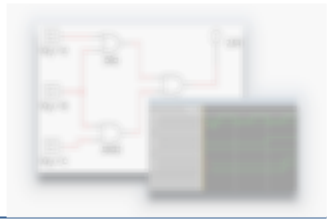
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Full-Adder



- Όταν προσθέτουμε δύο bits με τον ημιαθροιστή, παίρνουμε δυο εξόδους: ένα άθροισμα και ένα κρατούμενο
- Αυτό το κρατούμενο πρέπει να προστεθεί στην επόμενη στήλη από τα αριστερά (όταν υλοποιούμε πρόσθεση αριθμών με πολλά δυαδικά ψηφία)
- Συνεπώς χρειαζόμαστε ένα ψηφιακό κύκλωμα που να προσθέτει τρία bits μαζί (δυο ψηφία και τυχόν προηγούμενο κρατούμενο)
- Άρα, **ο πλήρης αθροιστής έχει τρεις εισόδους και δυο εξόδους.**

Full-Adder



- Το κύκλωμα που πραγματοποιεί την πρόσθεση δυο ψηφίων λαμβάνοντας υπόψη τυχόν προηγούμενο κρατούμενο, ονομάζεται **Πλήρης Αθροιστής**. Ο Πλήρης Αθροιστής έχει τρεις εισόδους A, B (τα bit που προστίθενται) και Ci (κρατούμενο εισόδου) και δυο εξόδους C (κρατούμενο εξόδου - carryout) και S (άθροισμα-sum).

A	B	C_i	S	C_0
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**

$$\begin{aligned} S &= \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + ABC \\ &= \overline{C}(\overline{A}B + A\overline{B}) + C(\overline{A}\,\overline{B} + AB) \\ &= \overline{C}(A \oplus B) + C(\overline{A \oplus B}) \\ &= \boldsymbol{(A \oplus B) \oplus C} \end{aligned}$$

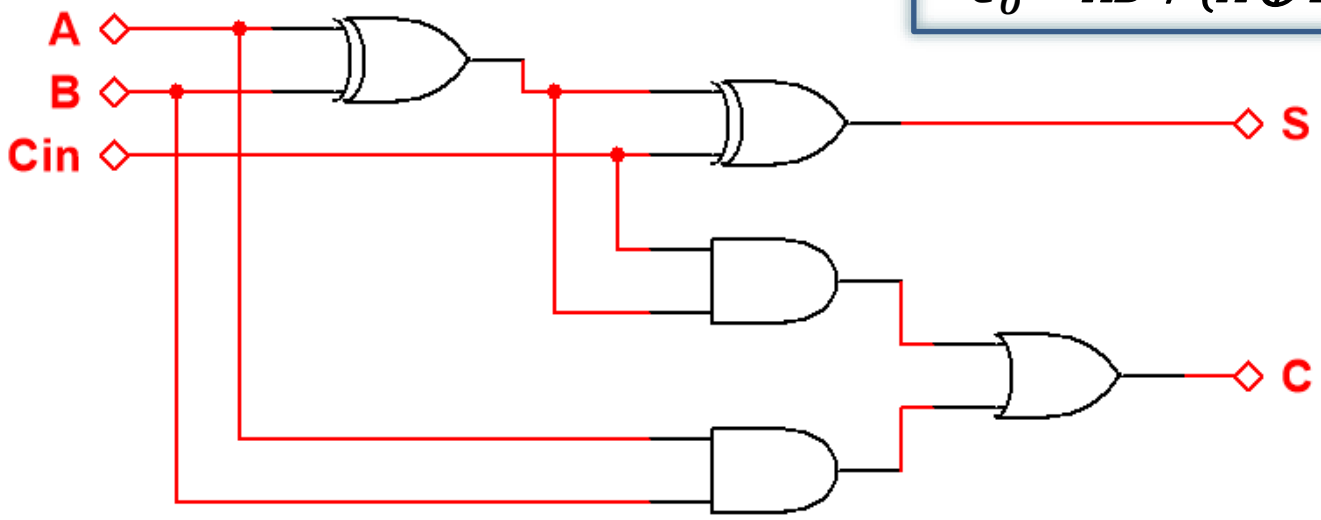
$$\begin{aligned} C_o &= \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC \\ &= C(\overline{A}B + A\overline{B}) + AB(\overline{C} + C) \\ &= \boldsymbol{AB + (A \oplus B)C} \end{aligned}$$

Full-Adder

- Το κύκλωμα που πραγματοποιεί την πρόσθεση δυο ψηφίων λαμβάνοντας υπόψη τυχόν προηγούμενο κρατούμενο, ονομάζεται **Πλήρης Αθροιστής**. Ο Πλήρης Αθροιστής έχει τρεις εισόδους A, B (τα bit που προστίθενται) και Ci (κρατούμενο εισόδου) και δυο εξόδους C (κρατούμενο εξόδου - carryout) και S (άθροισμα-sum).

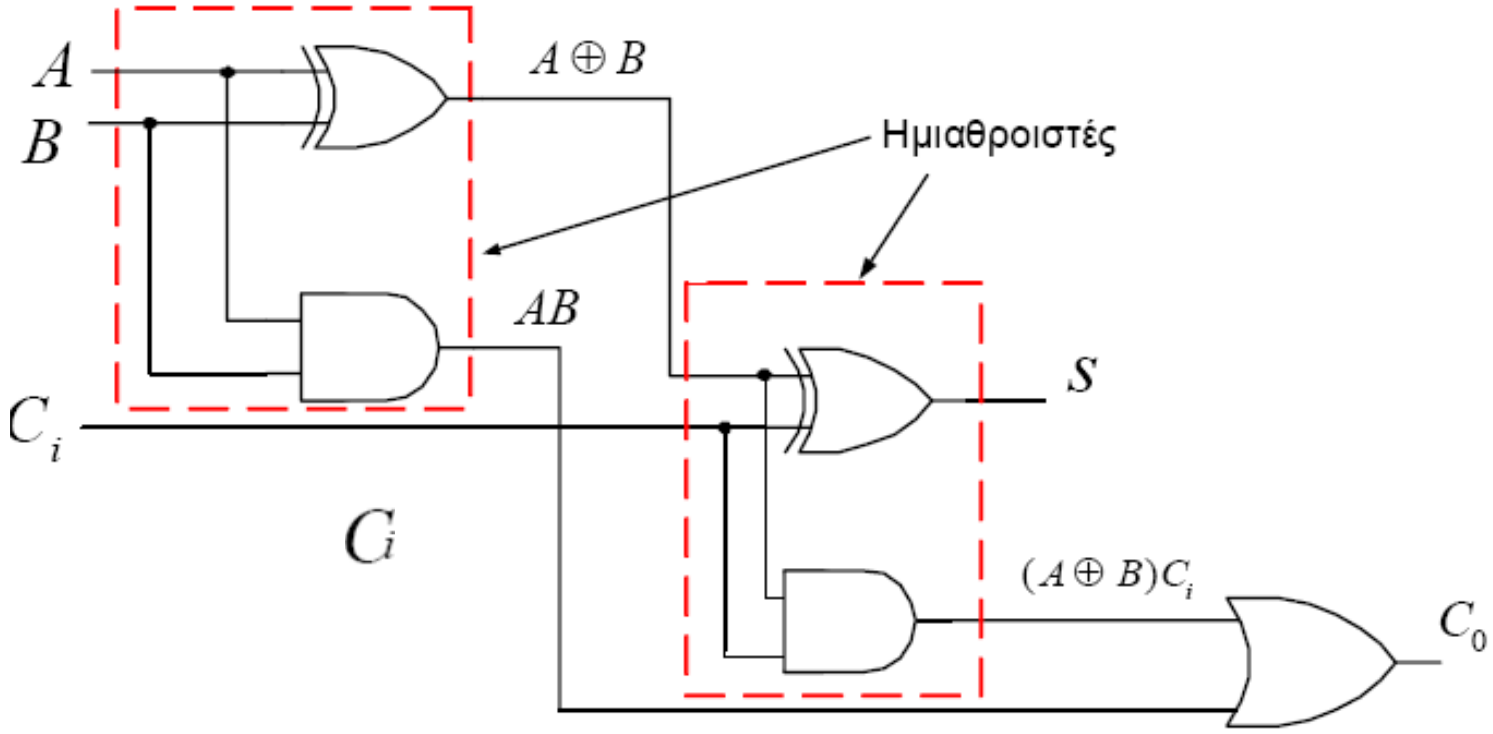
$$S = (A \oplus B) \oplus C$$
$$C_0 = AB + (A \oplus B)C$$

A	B	C_i	S	C_0
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**



Full-Adder – Υλοποίηση με Half-Adder

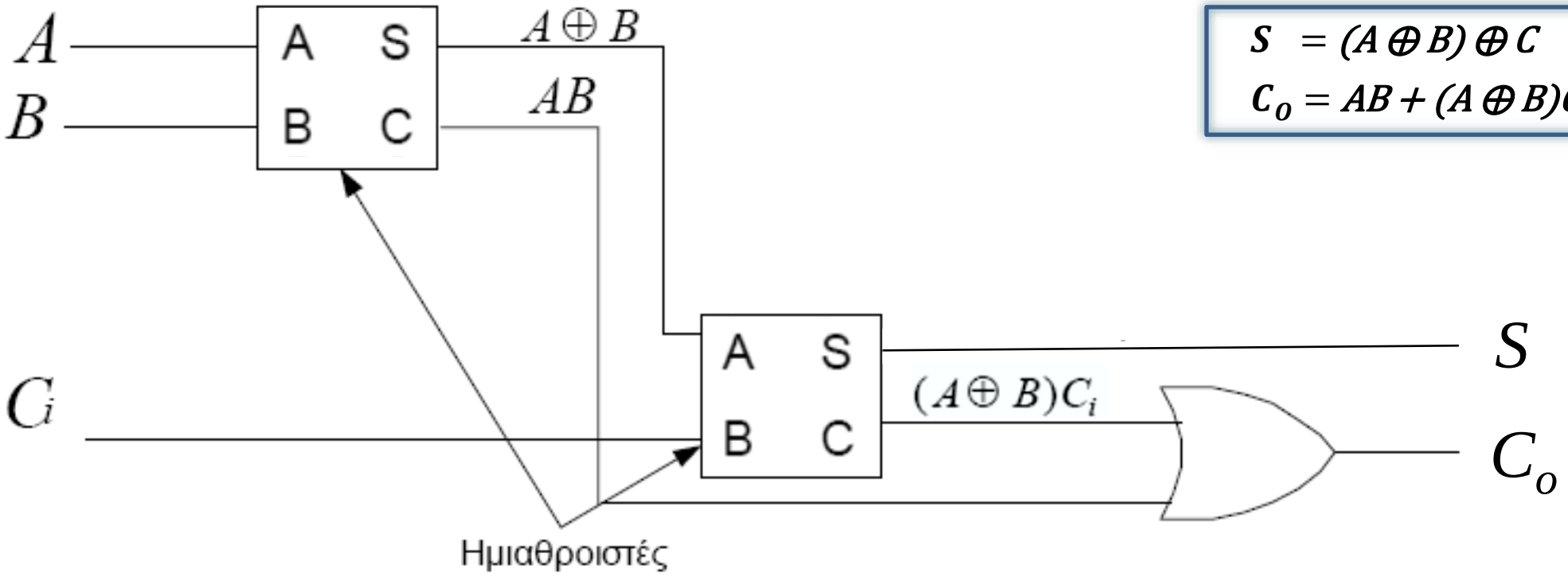




$$S = (A \oplus B) \oplus C$$
$$C_0 = AB + (A \oplus B)C$$

Full-Adder – Υλοποίηση με Half-Adder

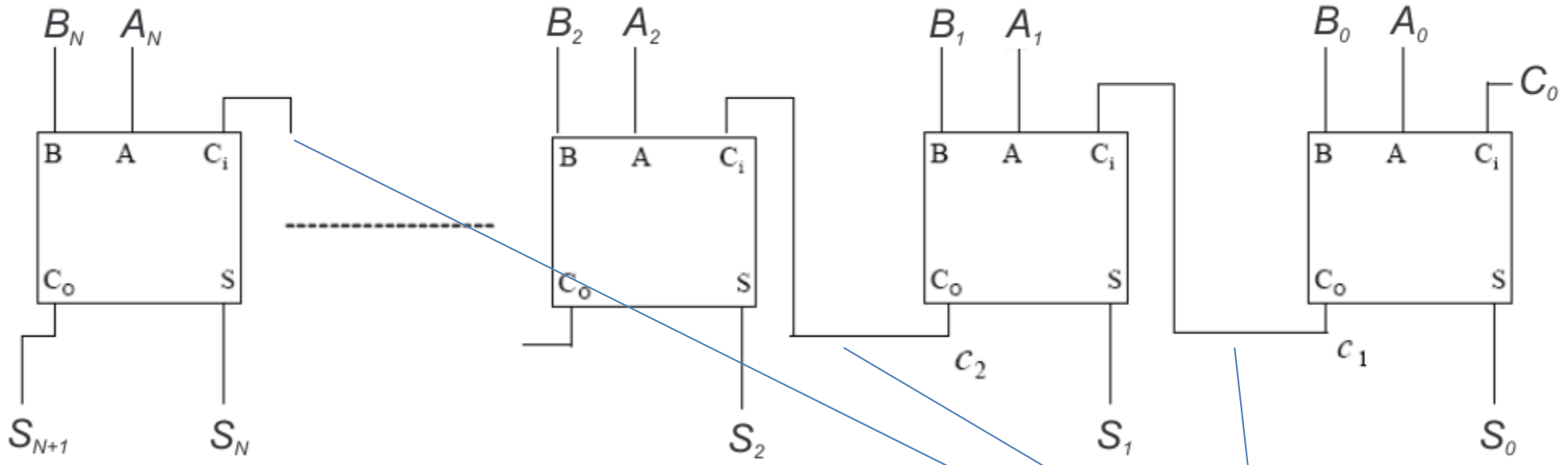
- Χρησιμοποιώντας το συμβολικό κύκλωμα του ημιαθροιστή:

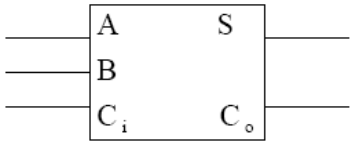


$$S = (A \oplus B) \oplus C$$
$$C_o = AB + (A \oplus B)C$$

Full-Adder

- Η άθροιση αριθμών με περισσότερα του ενός δυαδικά ψηφία γίνεται με το κύκλωμα του **παράλληλου αθροιστή** ως εξής:



όπου  το σύμβολο του πλήρους αθροιστή.

4-bit adder

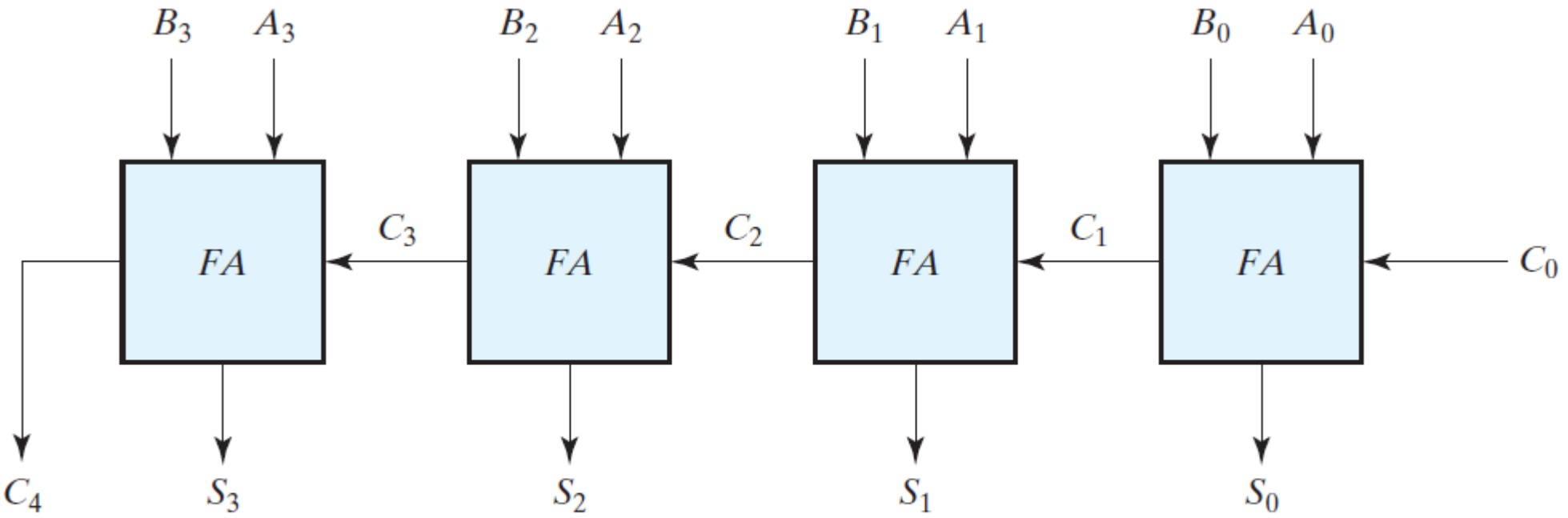
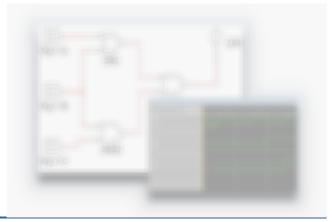


FIGURE 4.9
Four-bit adder

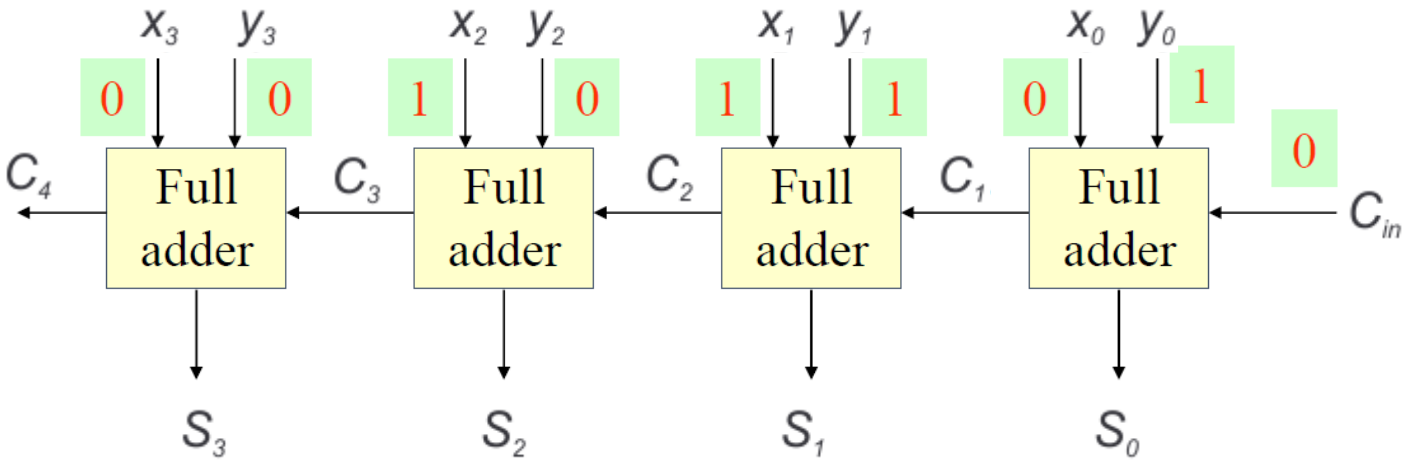
Full-Adder, Παράδειγμα πρόσθεσης 4 bit



- Ο **παράλληλος δυαδικός αθροιστής** παράγει το αριθμητικό άθροισμα δυο δυαδικών αριθμών παράλληλα.
- Αποτελείται από πλήρεις αθροιστές που συνδέονται σε σειρά και όπου το κρατούμενο εξόδου του ενός τροφοδοτεί το κρατούμενο εισόδου του επόμενου.
- **Παράδειγμα πρόσθεσης**
- Προσθέτουμε 6 + 3 σε υπολογιστή με μήκος λέξης 4 bit:
  - input x = 6 = 0110
  - input y = 3 = 0011
  - (αρχικό) κρατούμενο cin = 0

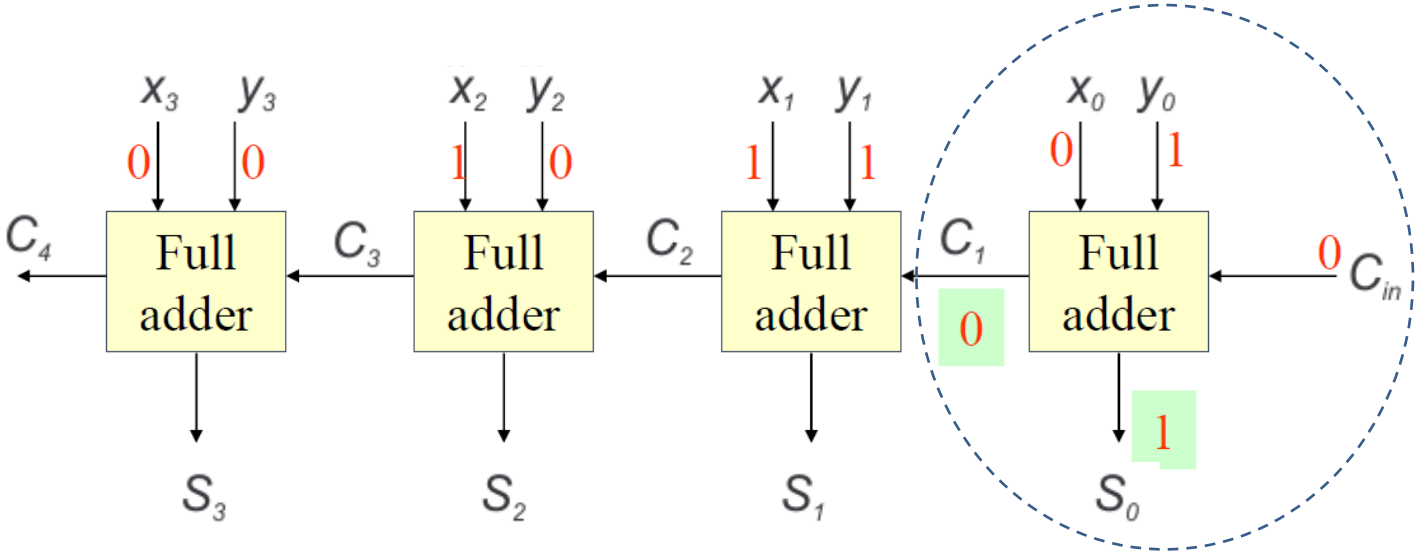
Full-Adder, Παράδειγμα πρόσθεσης 4 bit

x=0110 και y=0011



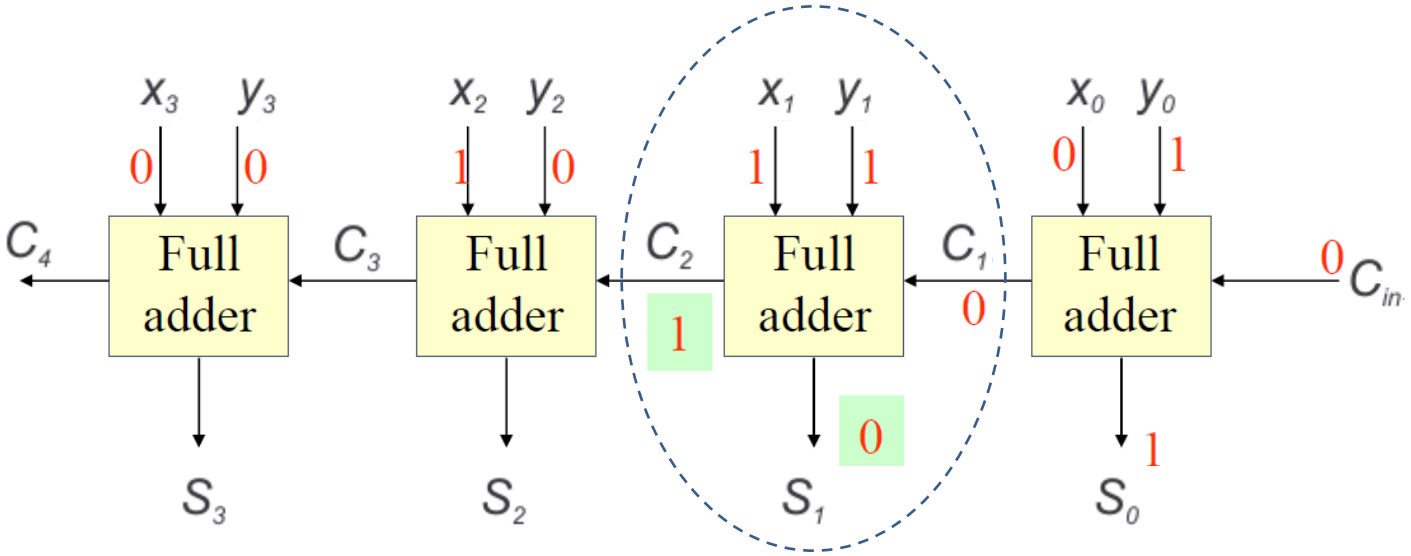
Full-Adder, Παράδειγμα πρόσθεσης 4 bit

Από τον πίνακα αληθείας βλέπουμε ότι **S_0=1** και **C_1=0**



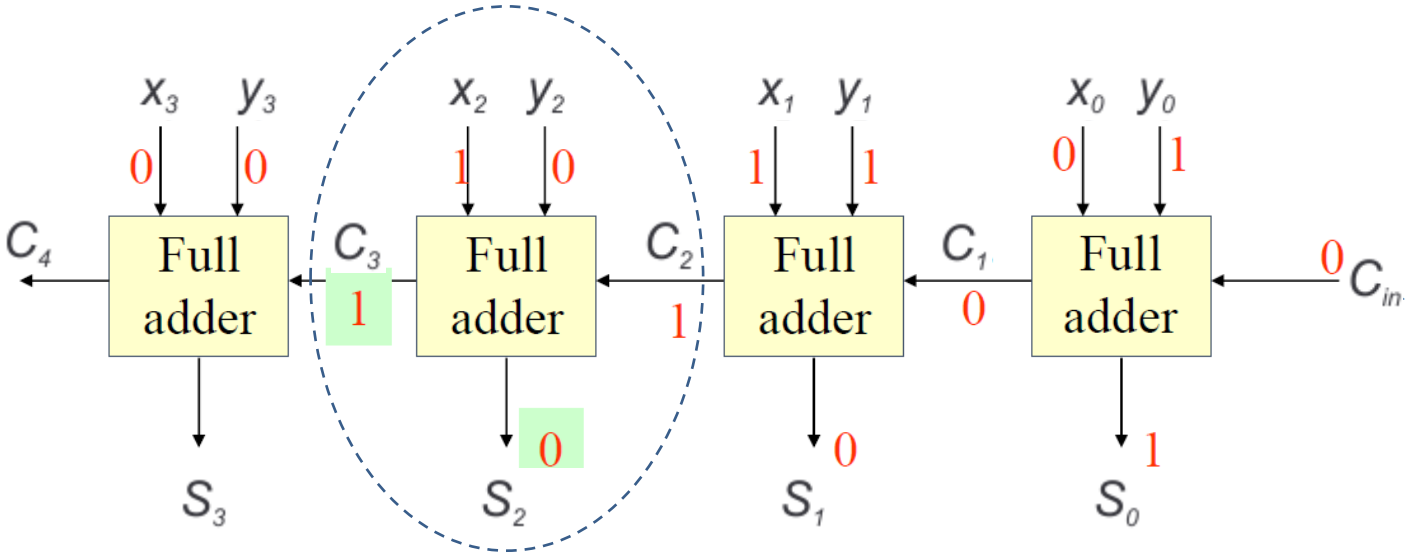
Full-Adder, Παράδειγμα πρόσθεσης 4 bit

Από τον πίνακα αληθείας βλέπουμε ότι S_1=0 και C_2=1 .



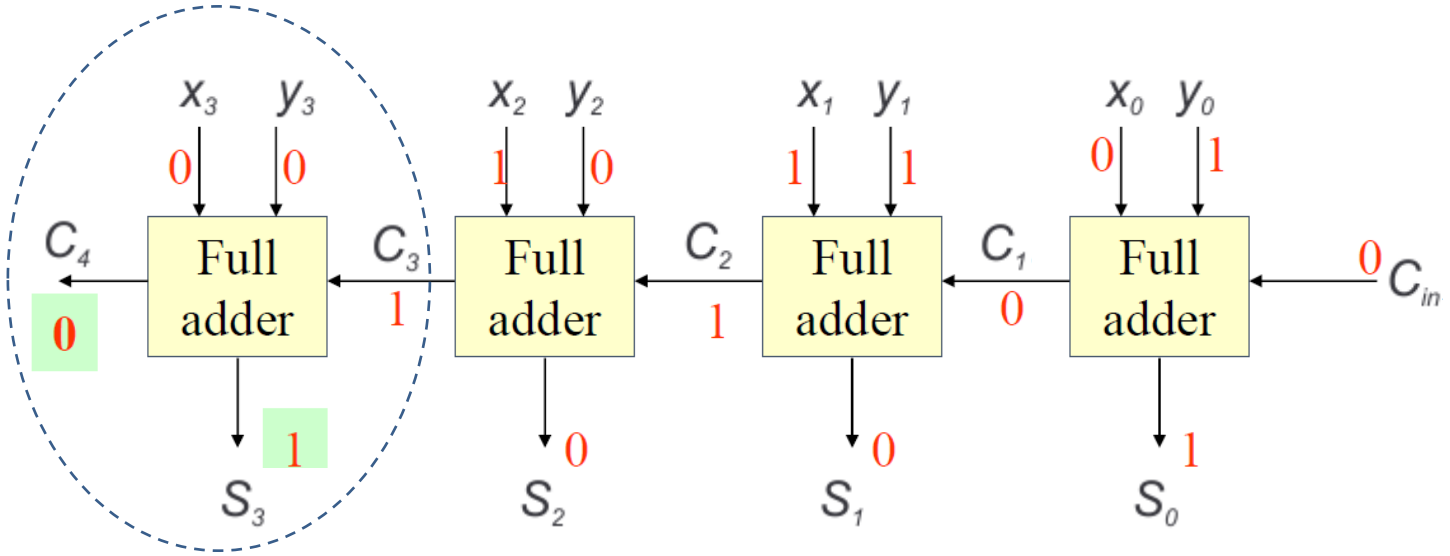
Full-Adder, Παράδειγμα πρόσθεσης 4 bit

Από τον πίνακα αληθείας βλέπουμε ότι **S_2=0** και **C_3=1** .



Full-Adder, Παράδειγμα πρόσθεσης 4 bit

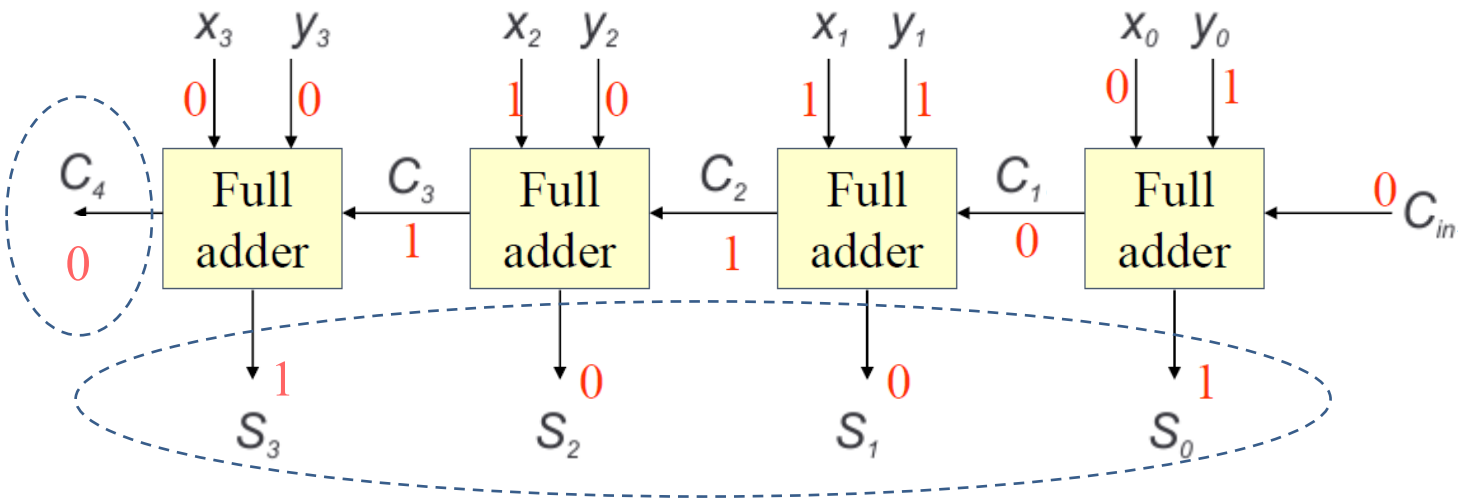
Από τον πίνακα αληθείας βλέπουμε ότι **S_3=1** και **C_4=0**



Full-Adder, Παράδειγμα πρόσθεσης 4 bit

Το αποτέλεσμα είναι: **Άθροισμα = [1,0,0,1] = 9**

Κρατούμενο $C_4 = 0$



4-bit adder-subtractor - Δυαδικός αθροιστής και αφαιρέτης

- Η αφαίρεση δυαδικών αριθμών μπορεί να γίνει εύκολα με τη χρήση των συμπληρωμάτων.
  Η αφαίρεση A-B υλοποιείται ως πρόσθεση του A με το συμπλήρωμα ως προς 2 του B:
  A−B = A + NOT(B) + 1.

- Οι πράξεις της πρόσθεσης και της αφαίρεσης μπορούν να συνδυαστούν σε ένα κοινό κύκλωμα

- Το κύκλωμα δεν γνωρίζει αν οι αριθμοί είναι **signed** ή **unsigned**. Εκτελεί τις ίδιες δυαδικές πράξεις. Εμείς ερμηνεύουμε το αποτέλεσμα και τις ενδείξεις **C** και **V** ανάλογα με την αναπαράσταση των αριθμών.

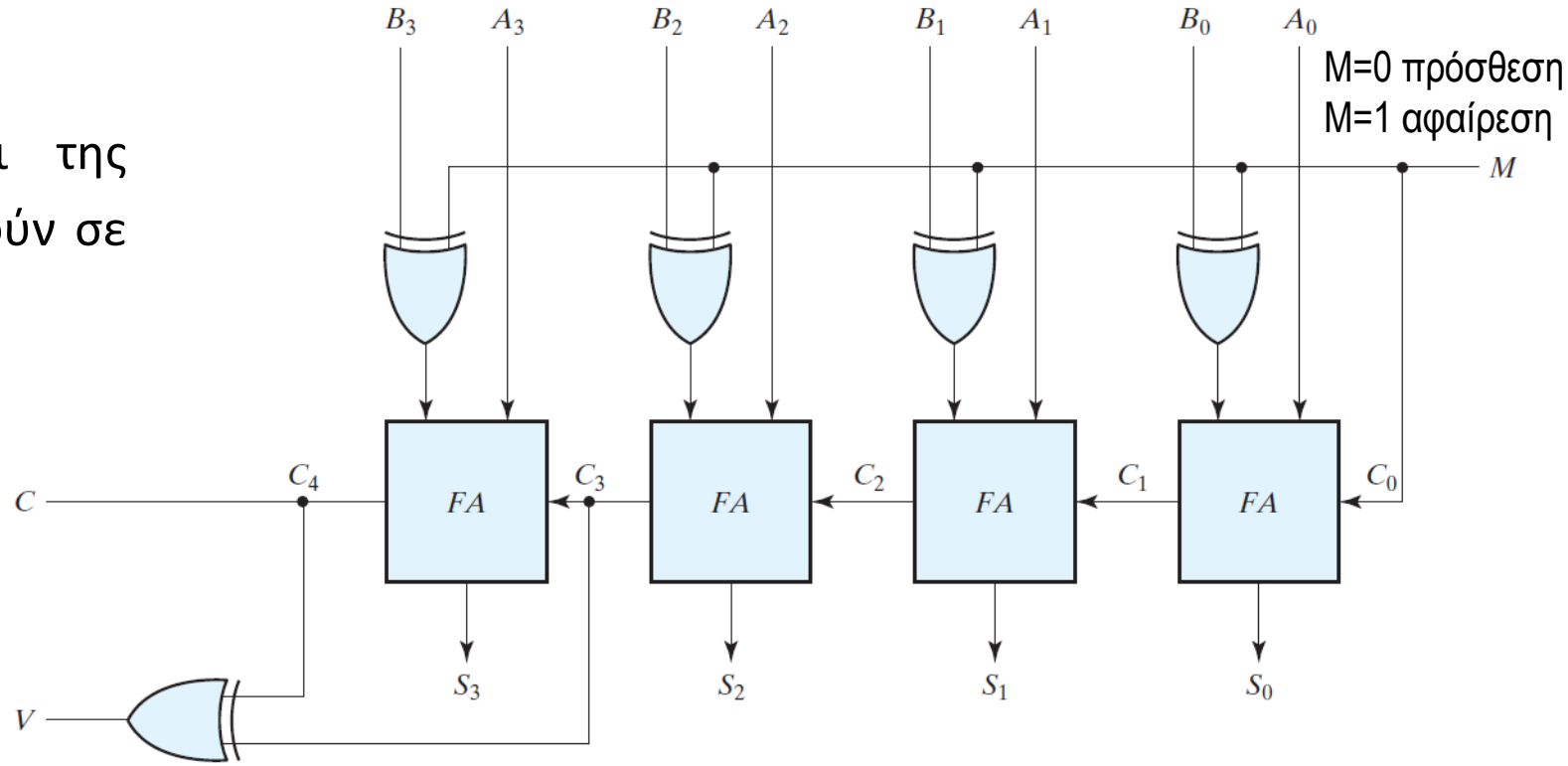


FIGURE 4.13
Four-bit adder–subtractor (with overflow detection)

4-bit adder-subtractor - Δυαδικός αθροιστής και αφαιρέτης

Έξοδοι του 4-bit Adder-Subtractor

- $S_3S_2S_1S_0$ (Result): Τα τέσσερα bits του αποτελέσματος της πρόσθεσης ή αφαίρεσης.
- $C = C_4$ (Carry-out): Το τελικό κρατούμενο. Στην unsigned πρόσθεση δείχνει υπέρβαση των 4 bits, ενώ στην unsigned αφαίρεση C=0 δηλώνει δανεισμό.
- $V = C_3 \oplus C_4$ (Overflow): Δηλώνει ότι το μαθηματικό αποτέλεσμα δεν μπορεί να αναπαρασταθεί ως signed 4-bit αριθμός.

Unsigned 4-bit: 0 έως 15

Signed 4-bit (two's complement): −8 έως +7

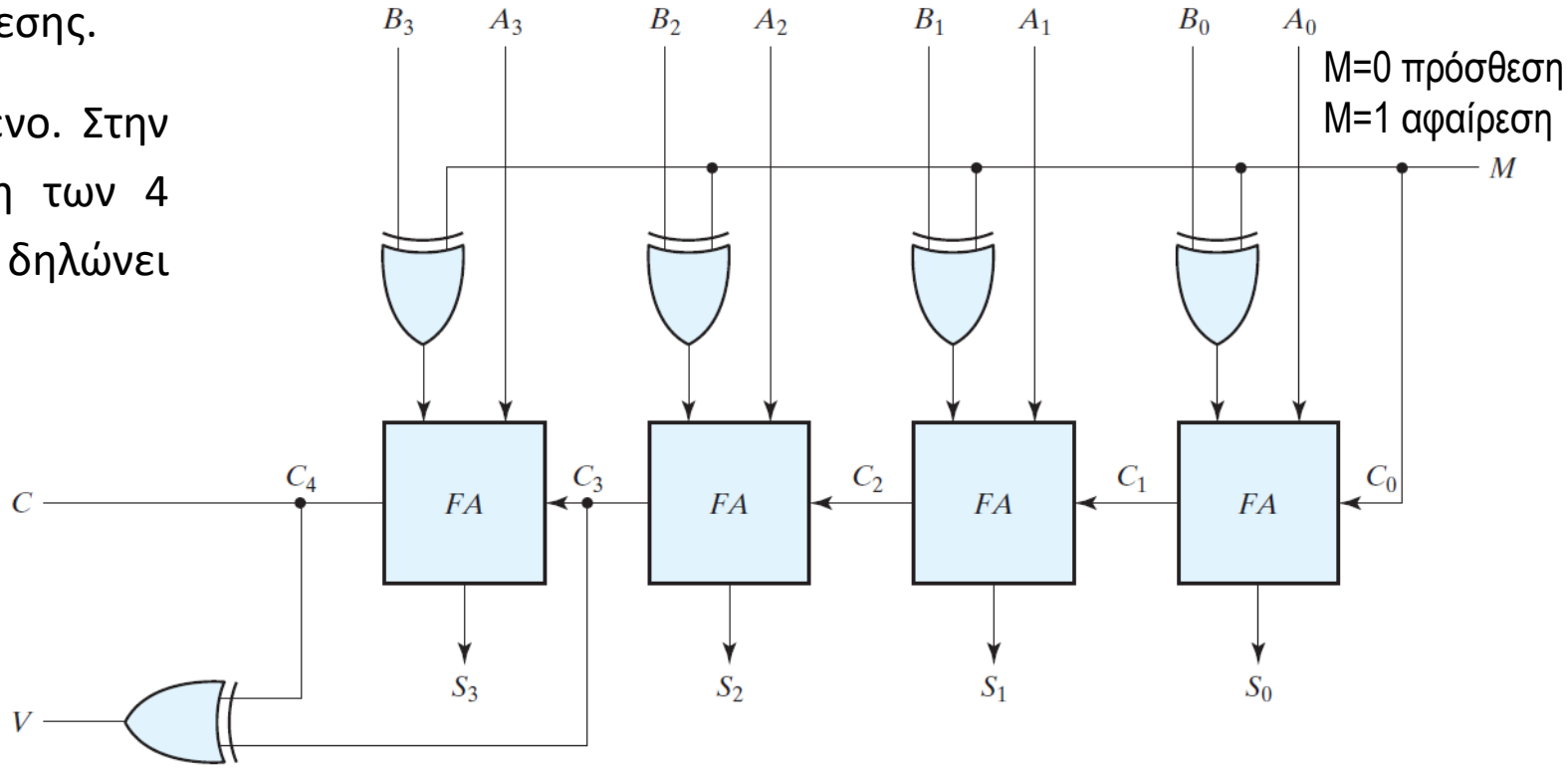
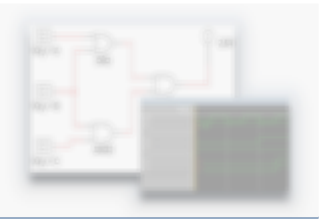


FIGURE 4.13
Four-bit adder–subtractor (with overflow detection)

4-bit adder-subtractor - Δυαδικός αθροιστής και αφαιρέτης



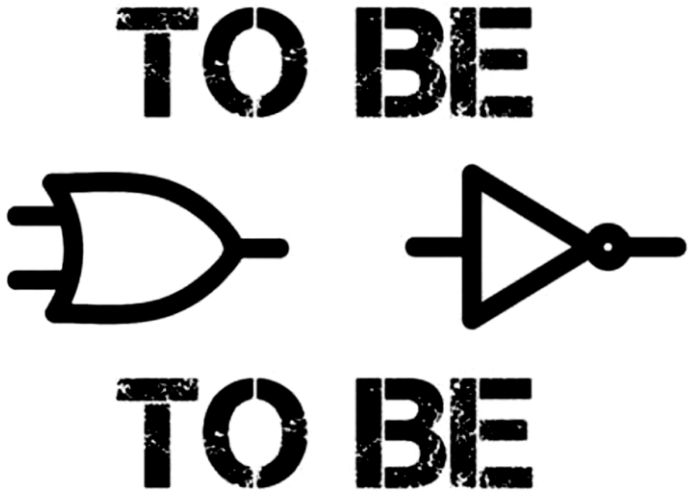
Σύντομος πίνακας αποφάσεων:

Περίπτωση (πράξη A-B)	Τι κάνουμε με το S
Unsigned πρόσθεση, C=0	Το S είναι το πλήρες αποτέλεσμα
Unsigned πρόσθεση, C=1	Χρησιμοποιούμε και το C για πλήρες αποτέλεσμα 5 bits
Unsigned αφαίρεση, C=1	Το S είναι η σωστή μη αρνητική διαφορά
Unsigned αφαίρεση, C=0	Έχει γίνει δανεισμός (borrow), δηλαδή A μικρότερο από B. Το S δεν δίνει απευθείας την αρνητική διαφορά. Παίρνουμε το συμπλήρωμα ως προς 2 του S για να βρούμε το μέτρο της και προσθέτουμε αρνητικό πρόσημο.
Signed πράξη, V=0	Το S είναι σωστό σε two's complement
Signed πράξη, V=1	Το αποτέλεσμα δεν χωρά σε 4 signed bits. Χρειάζονται περισσότερα bits για την αναπαράσταση του μαθηματικού αποτελέσματος.

Σημείωση:
Στους **signed αριθμούς**, όταν **V=0, το S είναι ήδη το σωστό αποτέλεσμα σε two's complement**. Δεν χρειάζεται μετατροπή. Αν το S αναπαριστά αρνητικό αριθμό (MSB=1), μπορούμε να πάρουμε το συμπλήρωμά του μόνο για να υπολογίσουμε το μέτρο του.

Ευχαριστώ για την προσοχή σας!

- **Ερωτήσεις / Απορίες ;**



Επικοινωνία: ece119.uth@gmail.com