



Συνδυαστική Λογική

Βαρτζιώτης Φώτιος

Συνδυαστικά Κυκλώματα

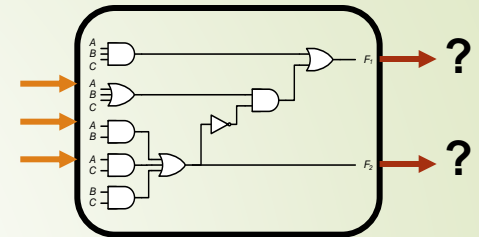
- Η έξοδος είναι συνάρτηση μόνο της εισόδου
 - Δεν υπάρχει βρόχος ανάδρασης



- Όταν αλλάζει η είσοδος, η έξοδος μπορεί να αλλάξει επίσης (με μια χρονική καθυστέρηση)

➤ Ανάλυση

- Έστω ένα κύκλωμα, βρείτε τη **λειτουργία** του
- Η λειτουργία μπορεί να εκφράζεται ως:
 - Boolean Συνάρτηση
 - Πίνακας Αληθείας



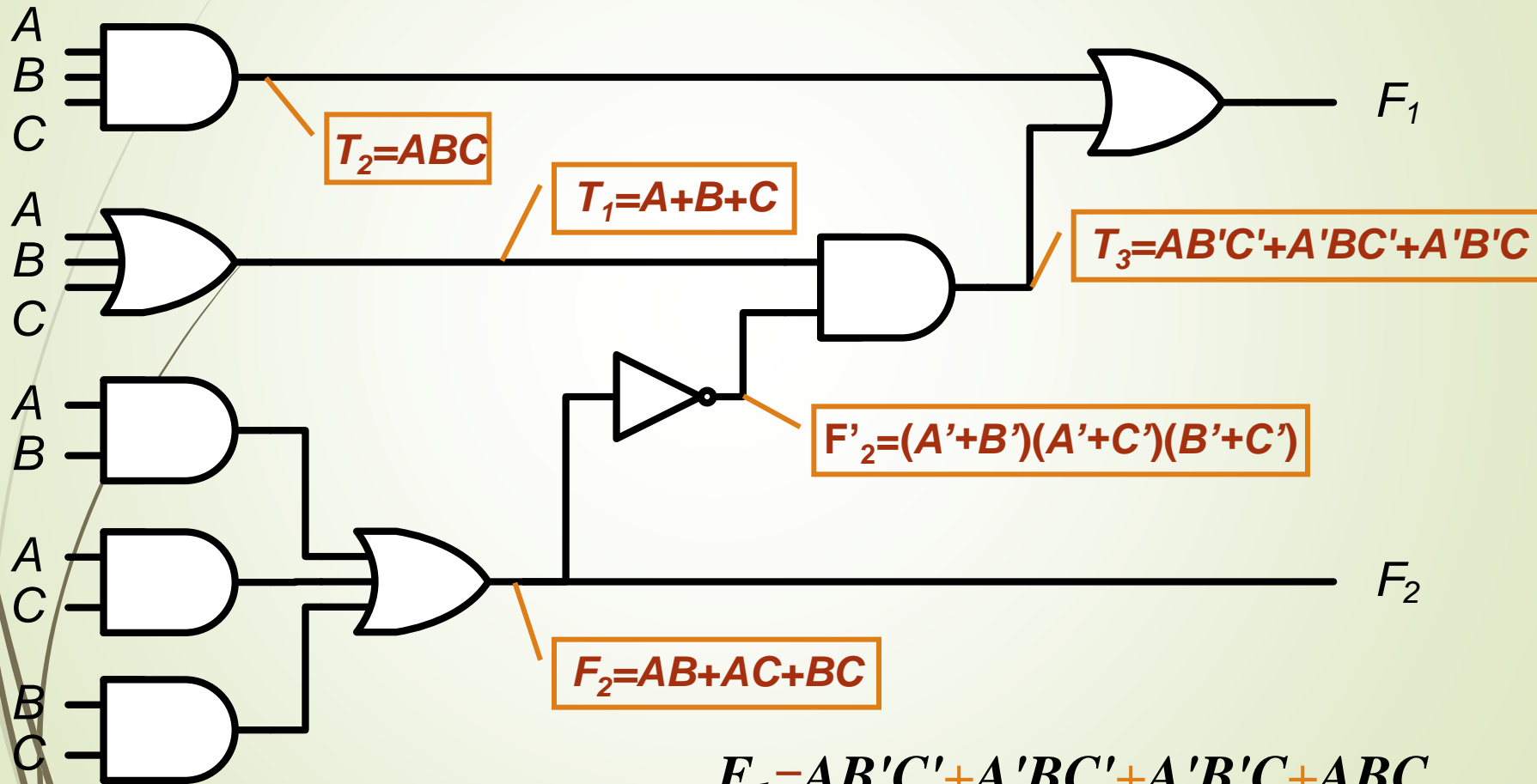
➤ Σχεδίαση

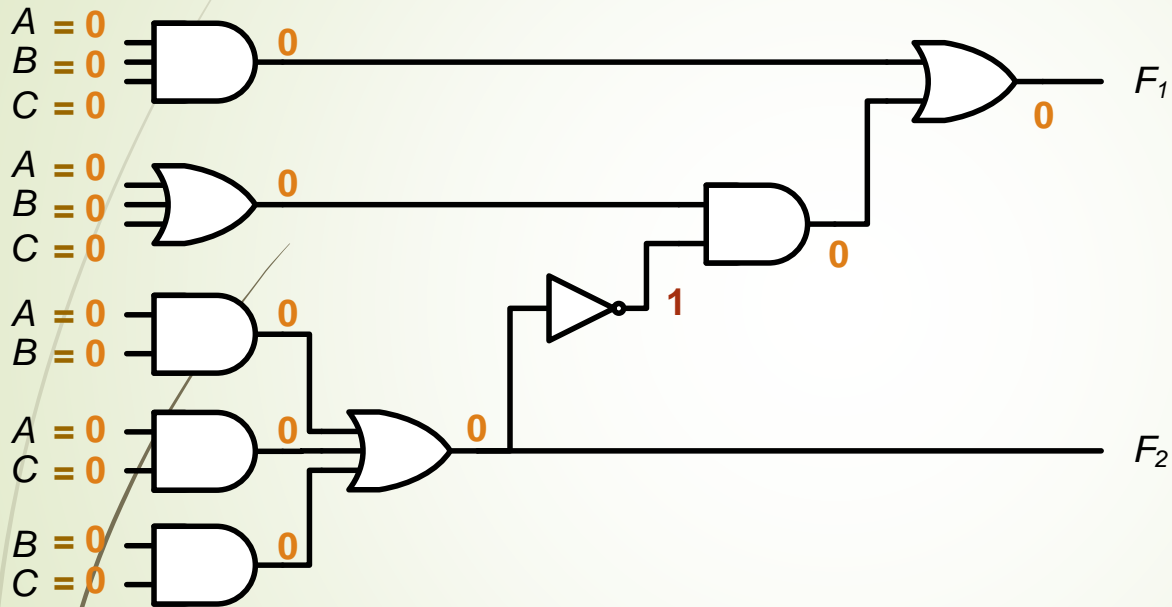
- Έστω μια επιθυμητή λειτουργία, βρείτε το **κύκλωμα** που την υλοποιεί.
- Η λειτουργία μπορεί να εκφράζεται ως:
 - Boolean Συνάρτηση
 - Πίνακας Αληθείας

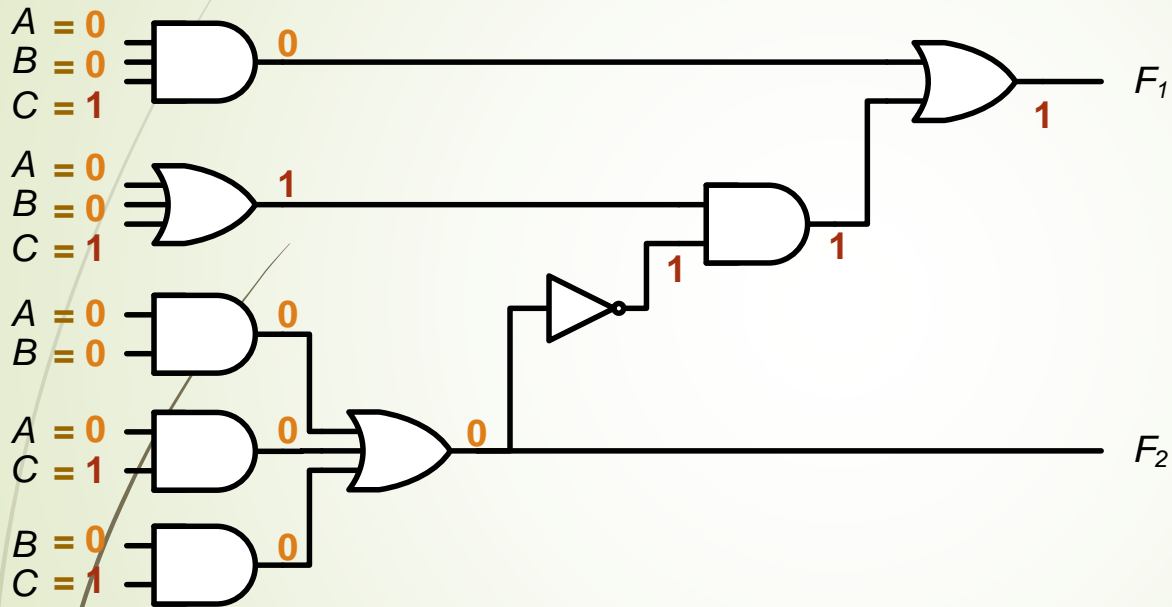


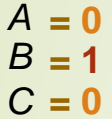
Διαδικασία Ανάλυσης

► Προσέγγιση υπολογισμού της Boolean συνάρτησης



[illegible]

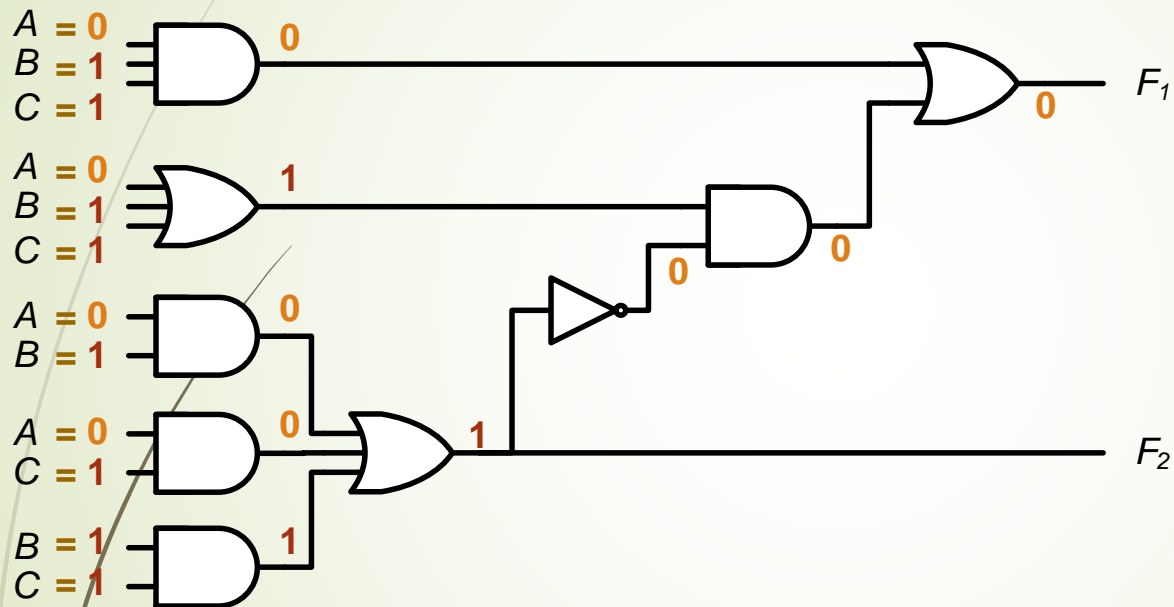
[illegible]



A	B	C	F_1	F_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0

Διαδικασία Ανάλυσης

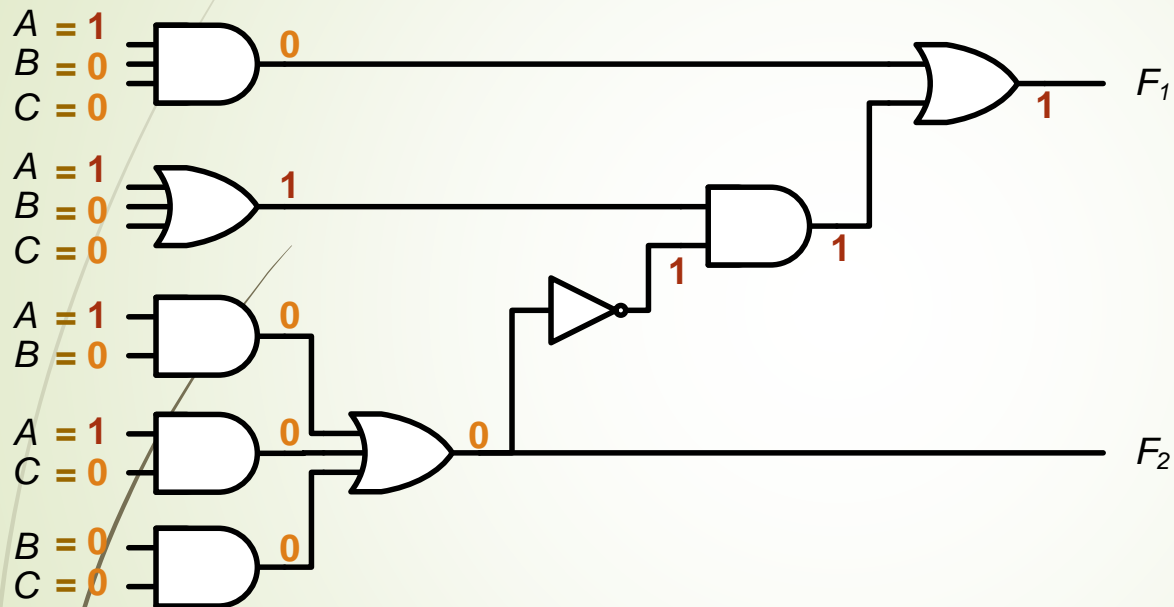
➤ Προσέγγιση με πίνακα αληθείας



<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i> ₁	<i>F</i> ₂
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1

Διαδικασία Ανάλυσης

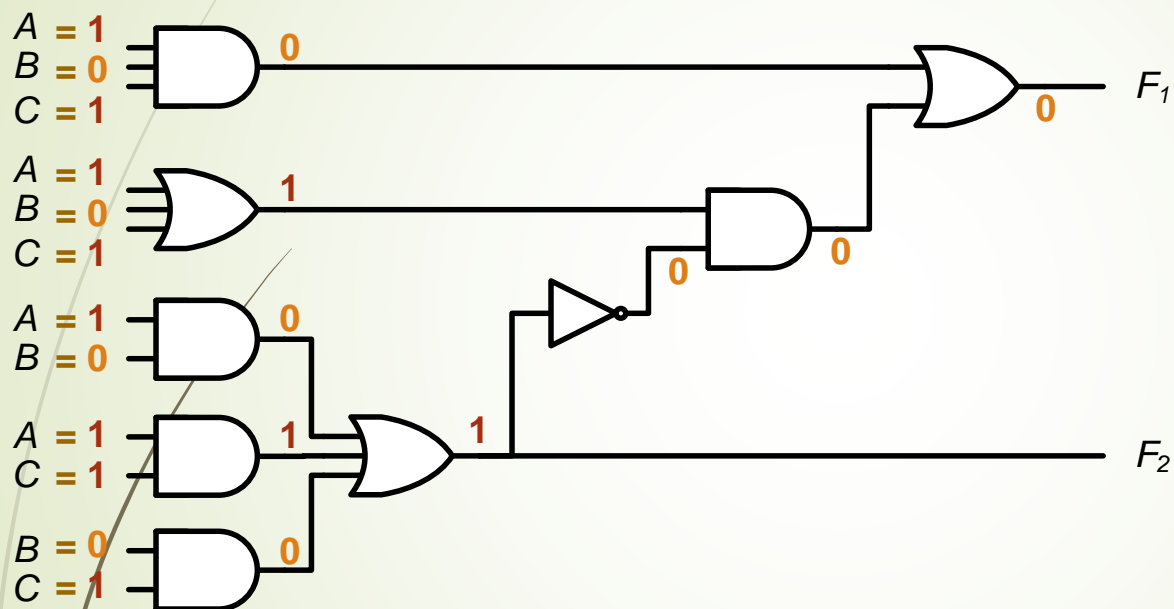
➔ Προσέγγιση με πίνακα αληθείας



A	B	C	F ₁	F ₂
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

Διαδικασία Ανάλυσης

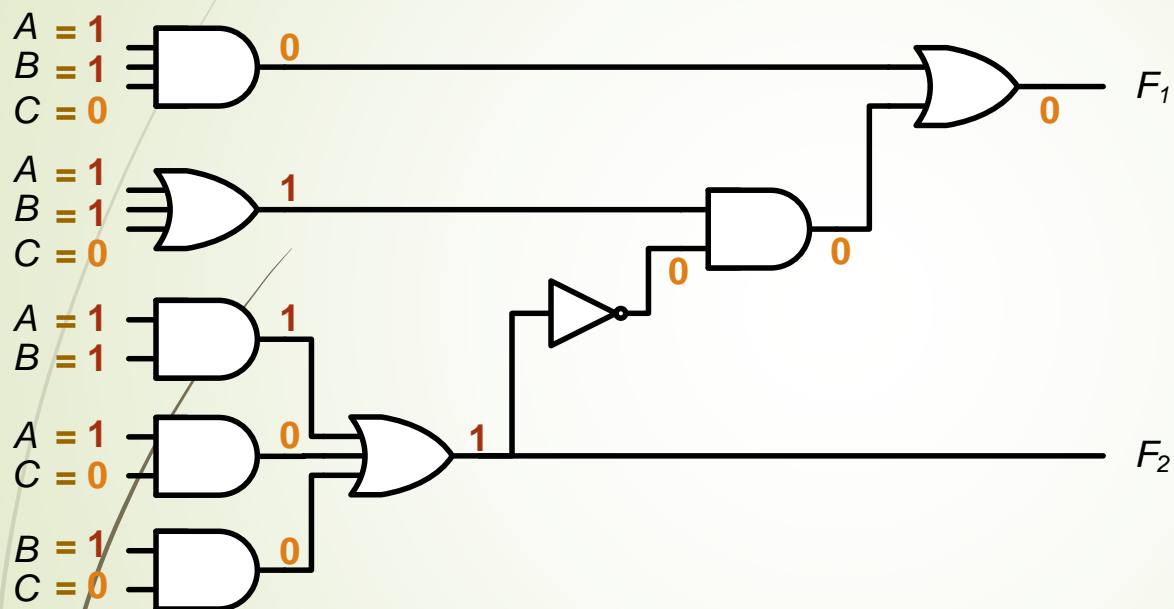
➤ Προσέγγιση με πίνακα αληθείας



<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i> ₁	<i>F</i> ₂
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

Διαδικασία Ανάλυσης

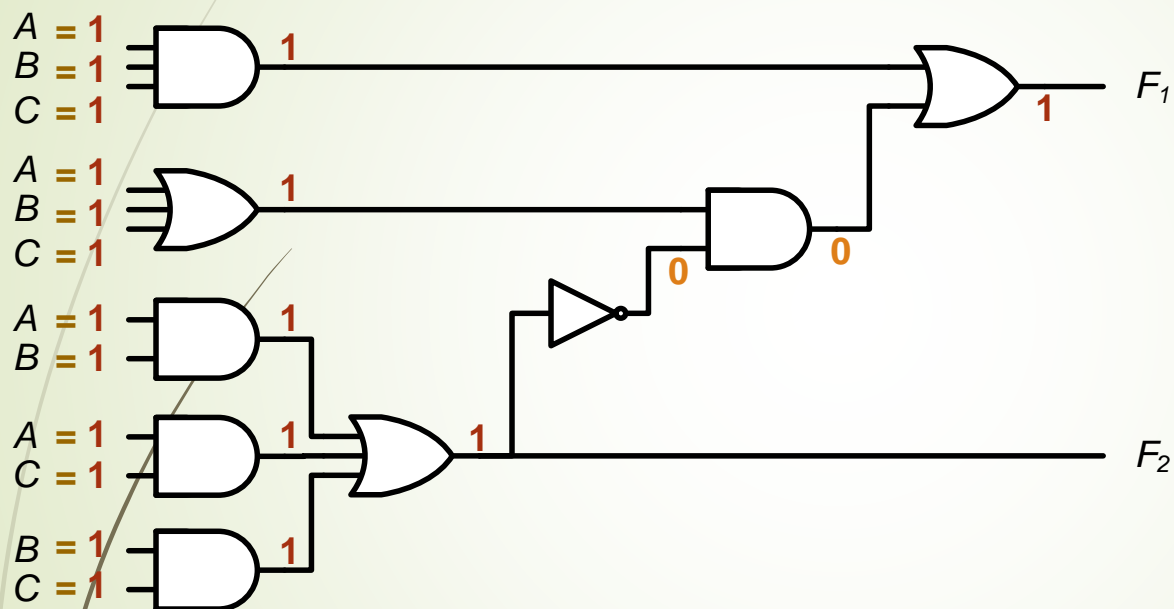
➤ Προσέγγιση με πίνακα αληθείας



A	B	C	F_1	F_2
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

Διαδικασία Ανάλυσης

➤ Προσέγγιση με πίνακα αληθείας



A	B	C	F_1	F_2
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

		B			
		0	1	0	1
A	0	0	1	0	1
	1	1	0	1	0
		C			

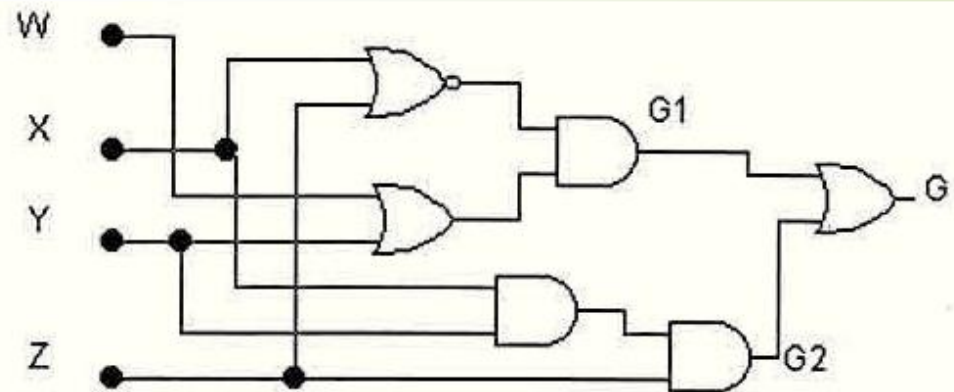
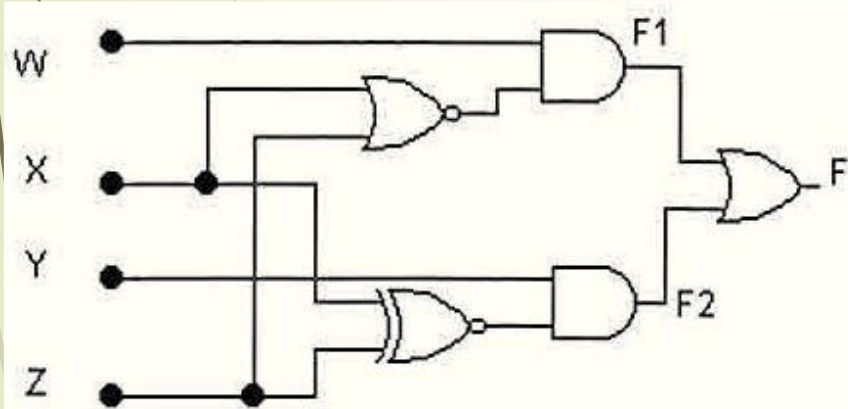
		B			
		0	0	1	0
A	0	0	0	1	0
	1	0	1	1	1
		C			

$$F_1 = AB'C' + A'BC' + A'B'C + ABC$$

$$F_2 = AB + AC + BC$$

Διαδικασία Ανάλυσης

Αναλύστε τα παρακάτω κυκλώματα

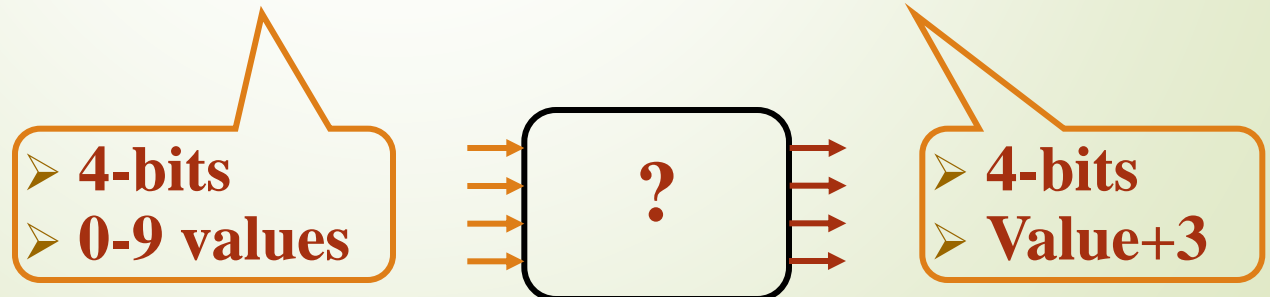


Διαδικασία Σχεδίασης

- Έστω οι προδιαγραφές ενός κυκλώματος:
 - Καθορίστε τον αριθμό των **εισόδων** και **εξόδων**
 - Εξάγεται τον πίνακα αληθείας
 - Απλοποιήστε την Boolean έκφραση **για κάθε έξοδο**
 - Σχεδιάστε το Λογικό Διάγραμμα

Παράδειγμα:

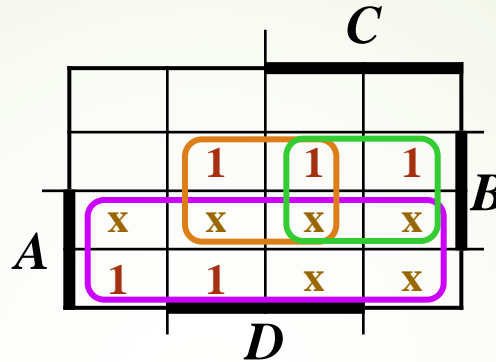
Σχεδιάστε ένα κύκλωμα μετατροπής του κώδικα "BCD" σε κώδικα "Excess 3"



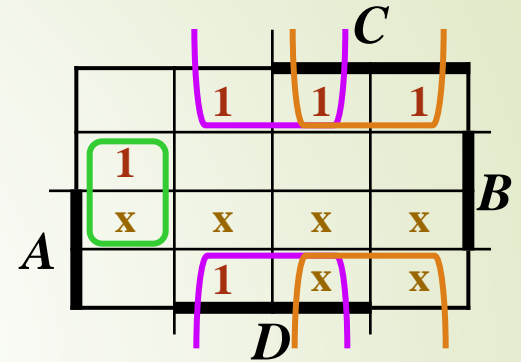
Διαδικασία Σχεδίασης

➔ Μετατροπέας BCD-to-Excess 3

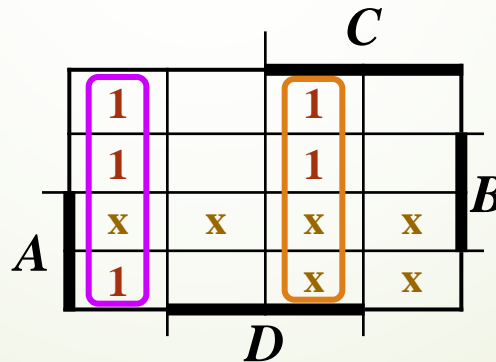
<i>A B C D</i>	<i>w x y z</i>
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 1 0 0
1 0 1 0	x x x x
1 0 1 1	x x x x
1 1 0 0	x x x x
1 1 0 1	x x x x
1 1 1 0	x x x x
1 1 1 1	x x x x



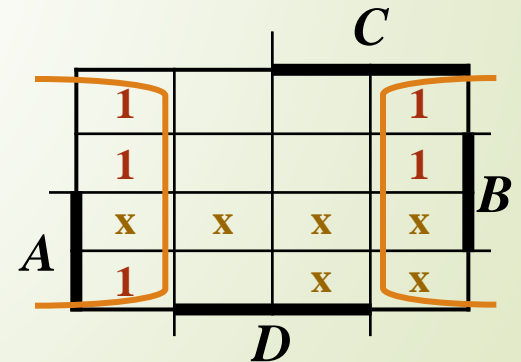
$$w = A + BC + BD$$



$$x = B'C + B'D + BC'D'$$



$$y = C'D' + CD$$

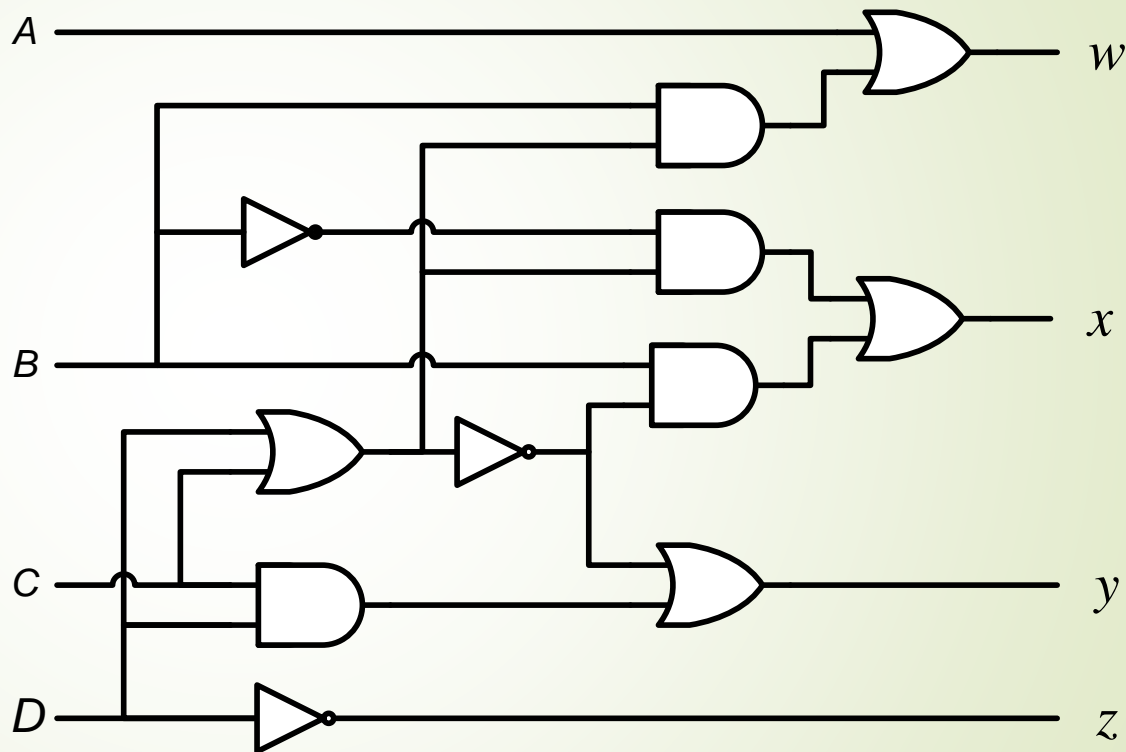


$$z = D'$$

Διαδικασία Σχεδίασης

➔ Μετατροπέας BCD-to-Excess 3

<i>A B C D</i>	<i>w x y z</i>
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 1 0 0
1 0 1 0	x x x x
1 0 1 1	x x x x
1 1 0 0	x x x x
1 1 0 1	x x x x
1 1 1 0	x x x x
1 1 1 1	x x x x



$$w = A + B(C + D)$$

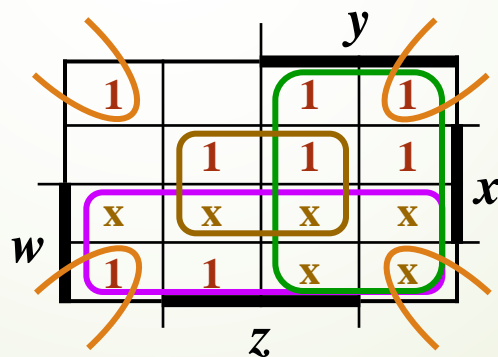
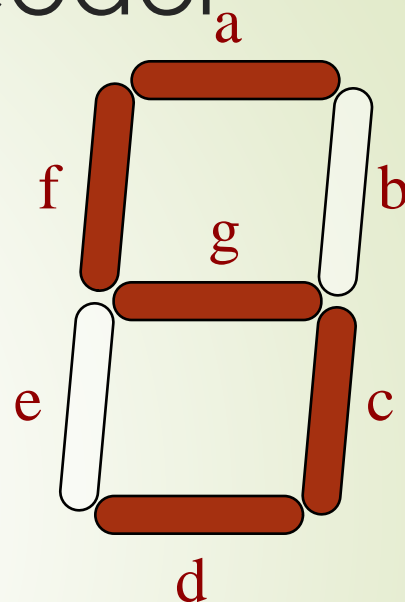
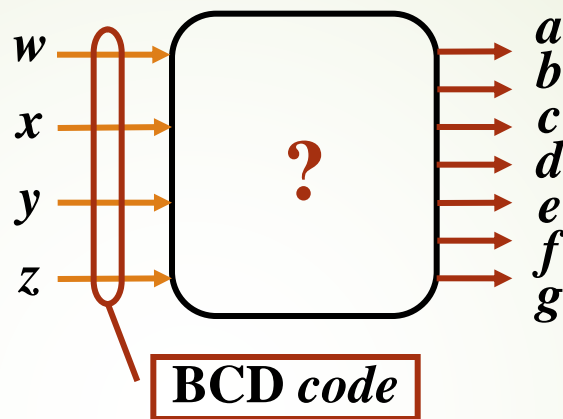
$$x = B'(C + D) + B(C + D)'$$

$$y = (C + D)' + CD$$

$$z = D'$$

Seven-Segment Decoder

<i>w x y z</i>	<i>a b c d e f g</i>
0 0 0 0	1 1 1 1 1 1 0
0 0 0 1	0 1 1 0 0 0 0
0 0 1 0	1 1 0 1 1 0 1
0 0 1 1	1 1 1 1 0 0 1
0 1 0 0	0 1 1 0 0 1 1
0 1 0 1	1 0 1 1 0 1 1
0 1 1 0	1 0 1 1 1 1 1
0 1 1 1	1 1 1 0 0 0 0
1 0 0 0	1 1 1 1 1 1 1
1 0 0 1	1 1 1 1 0 1 1
1 0 1 0	x x x x x x x
1 0 1 1	x x x x x x x
1 1 0 0	x x x x x x x
1 1 0 1	x x x x x x x
1 1 1 0	x x x x x x x
1 1 1 1	x x x x x x x



$$a = w + y + xz + x'z'$$

$$b = \dots$$

$$c = \dots$$

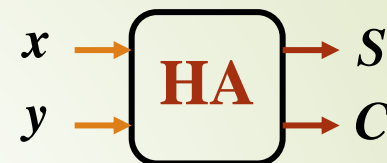
$$d = \dots$$



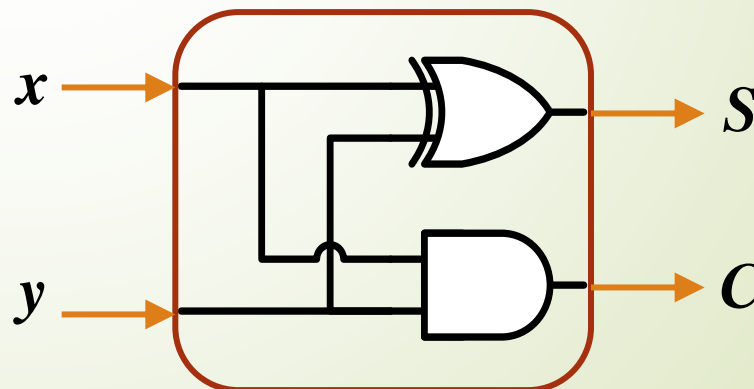
Δυαδικός Αθροιστής

- Ημιαθροιστής
 - Προσθέτει 1-bit συν 1-bit
 - Παράγει τις εξόδους Sum και Carry

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



$$\begin{array}{r} x \\ + y \\ \hline C \quad S \end{array}$$



Δυαδικός Αθροιστής

► Πλήρης Αθροιστής

- Προσθέτει 1-bit συν 1-bit συν 1-bit
- Παράγει τις εξόδους Sum και Carry



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

		y	
		0	1
x		1	0
	z	1	0

$$S = xy'z' + x'yz' + x'y'z + xyz = x \oplus y \oplus z$$

		y	
		0	0
x		0	1
	z	1	1

$$C = xy + xz + yz$$

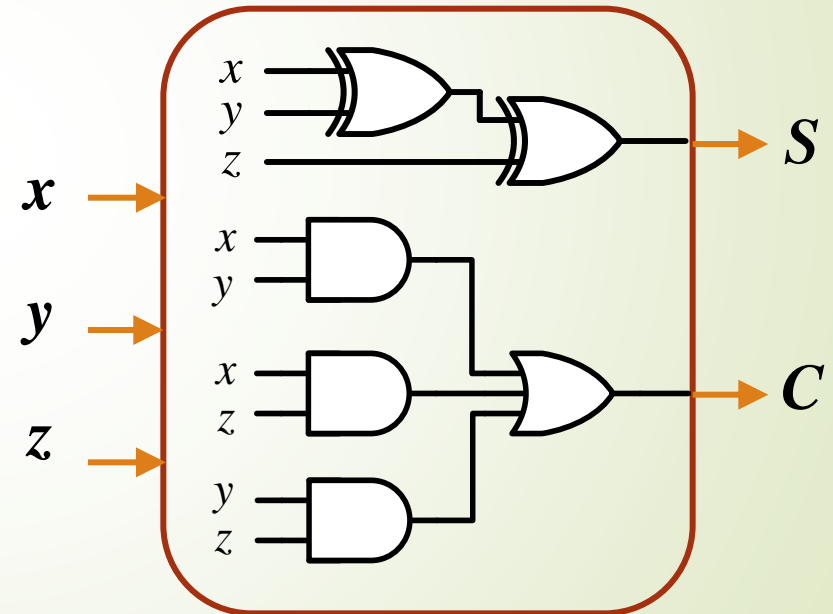
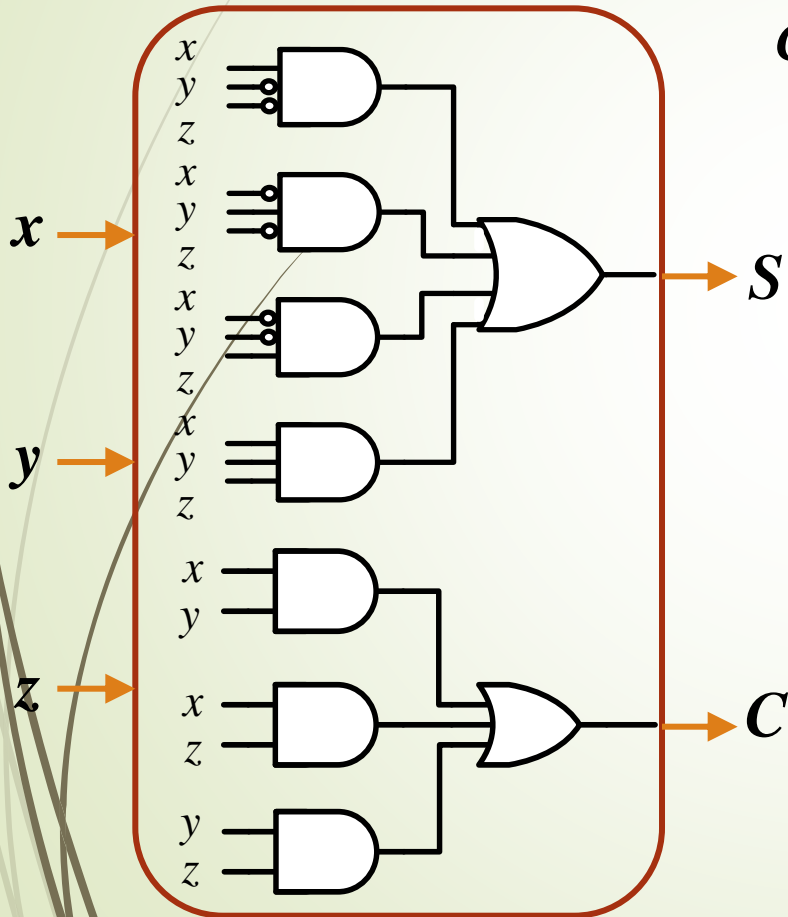
$$\begin{array}{r} x \\ + y \\ + z \\ \hline C \quad S \end{array}$$

Δυαδικός Αθροιστής

► Πλήρης Αθροιστής

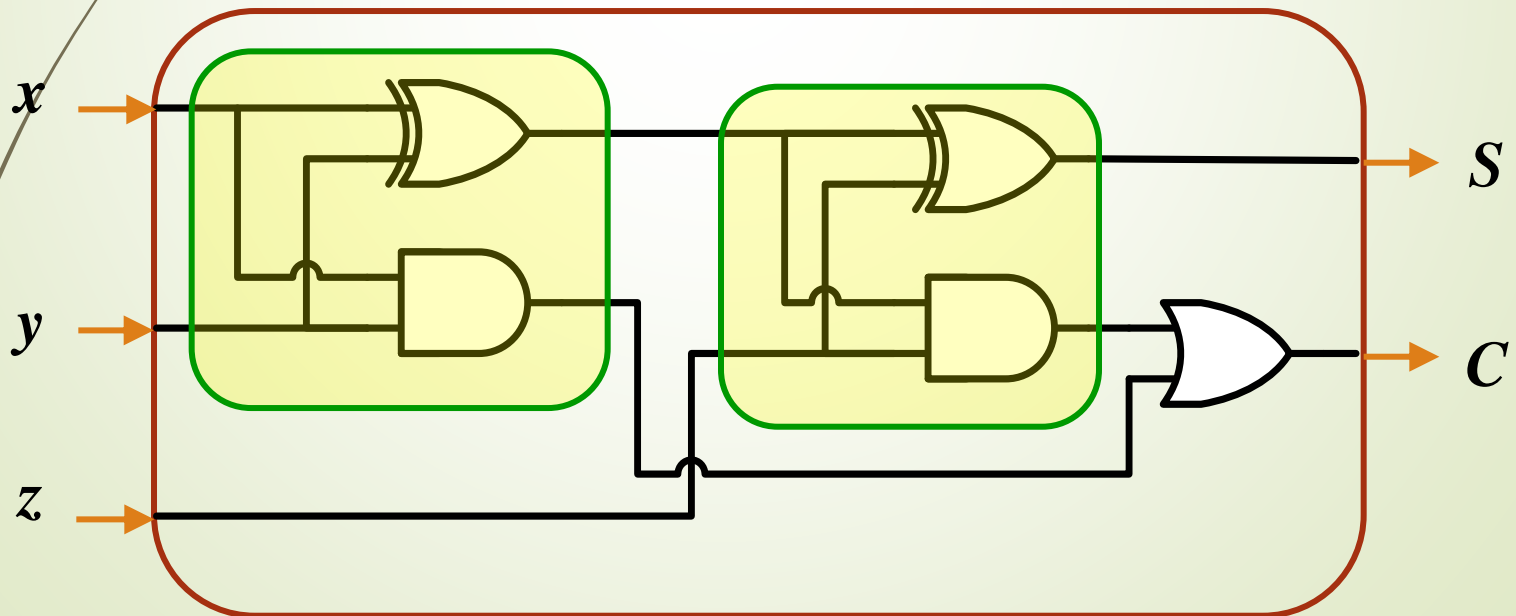
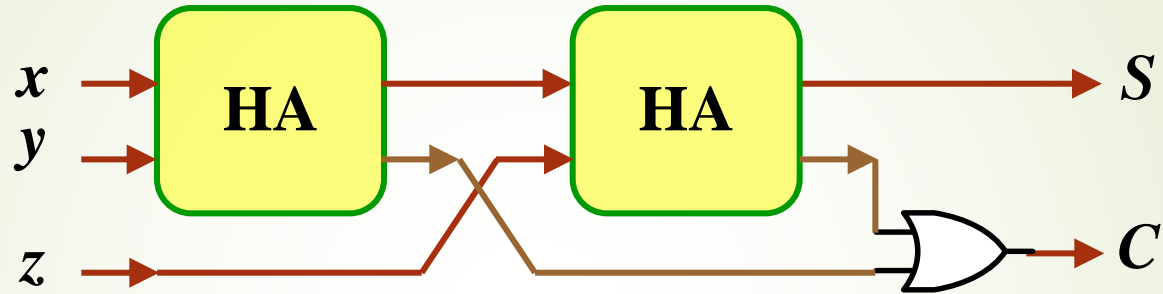
$$S = xy'z' + x'yz' + x'y'z + xyz = x \oplus y \oplus z$$

$$C = xy + xz + yz$$

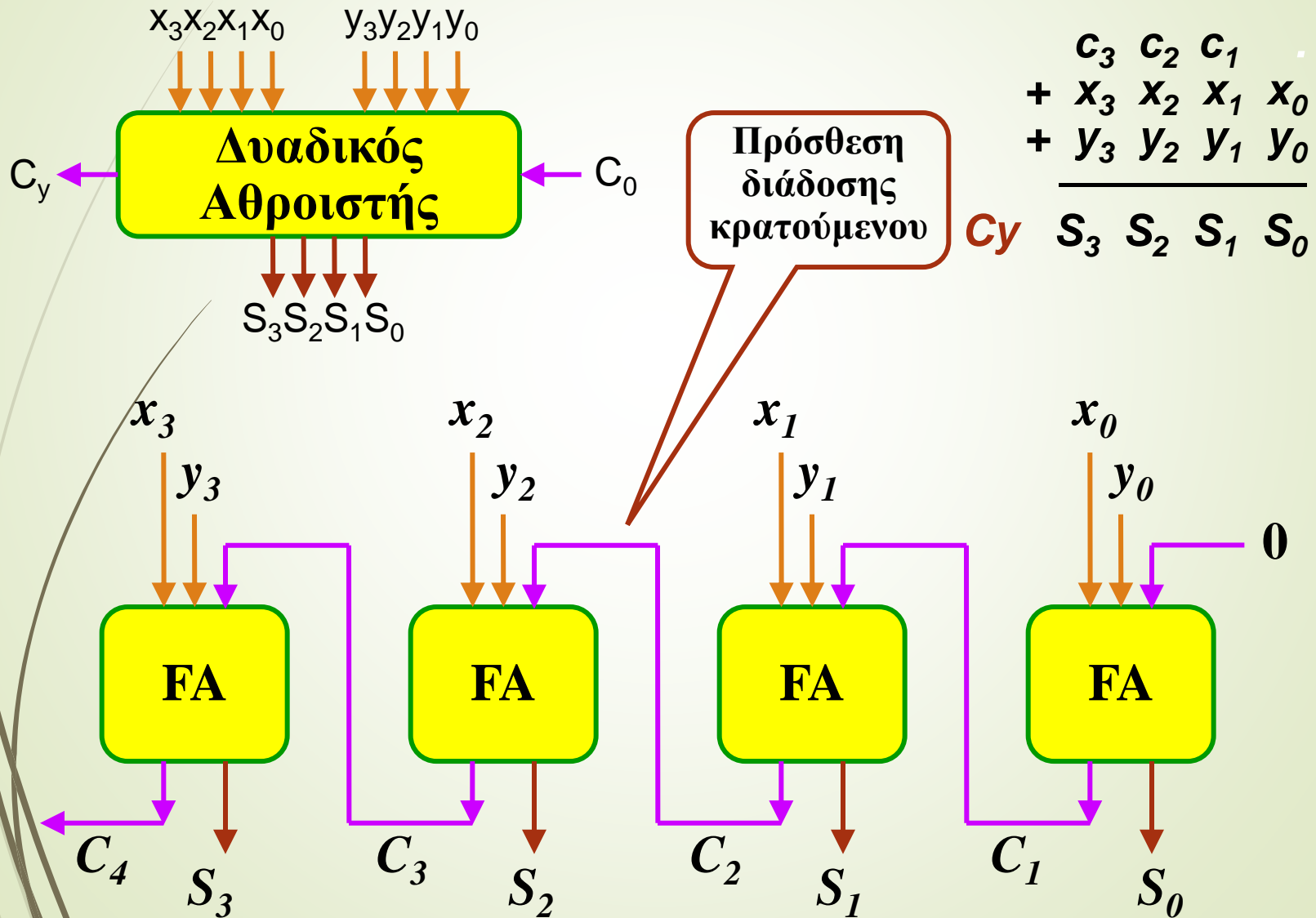


Δυαδικός Αθροιστής

► Πλήρης Αθροιστής

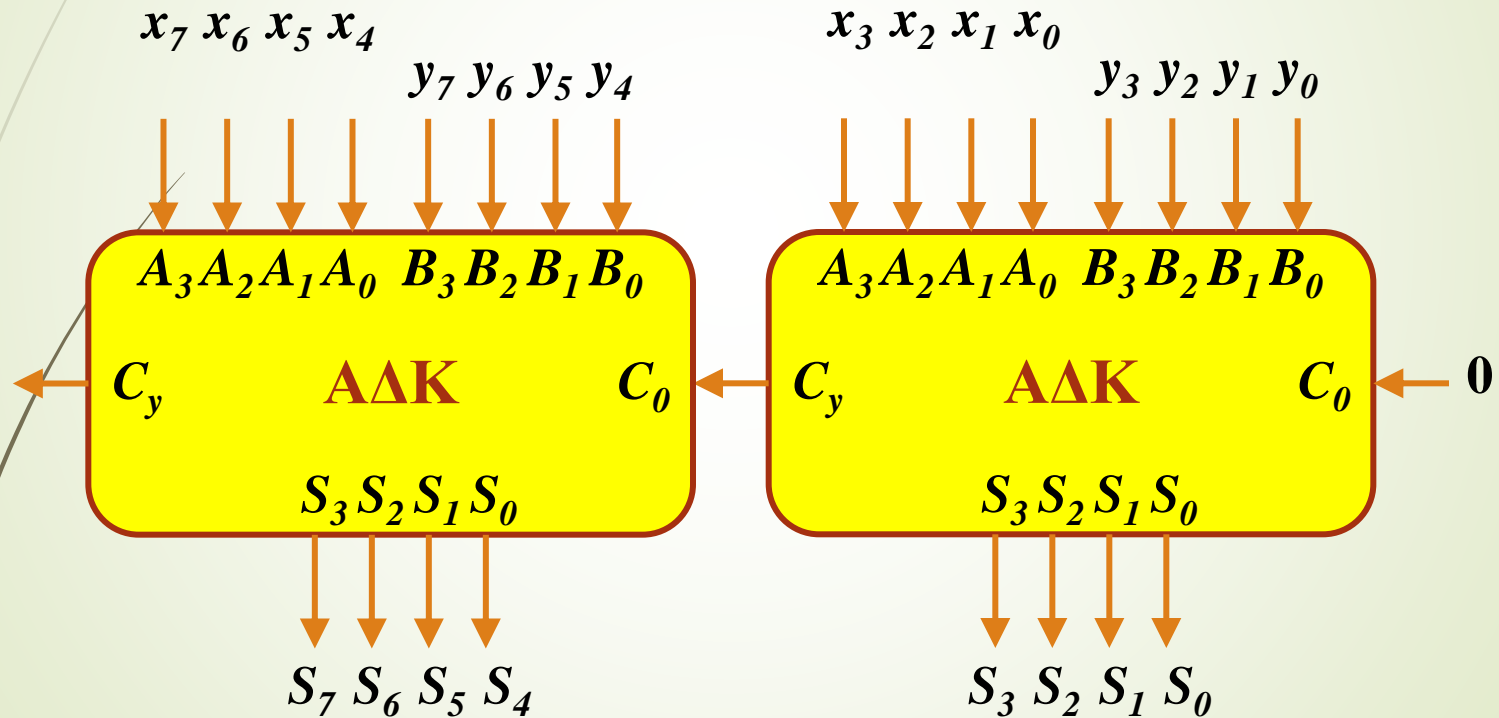


Δυαδικός Αθροιστής



Δυαδικός Αθροιστής

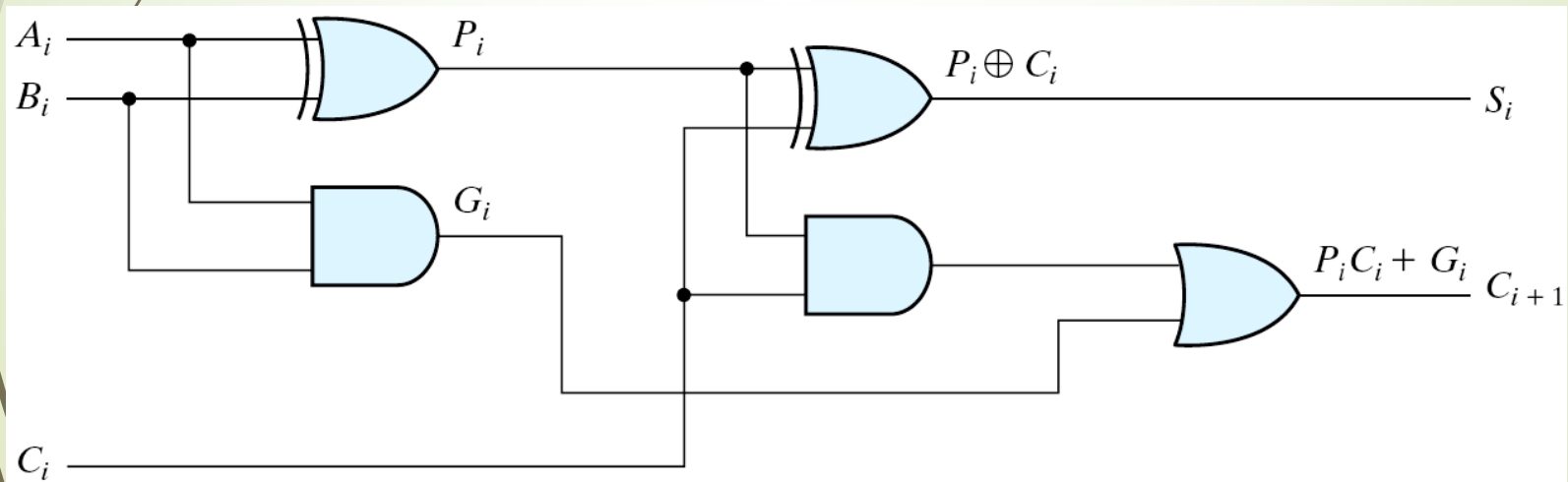
- Αθροιστής Διάδοσης Κρατούμενου (ΑΔΚ)



Δυαδικός Αθροιστής

➤ Διάδοση κρατούμενων

- Σκεφτείτε πότε είναι διαθέσιμες οι έγκυρες έξοδοι
- Το κρίσιμο μονοπάτι του κυκλώματος μετράει (η μεγαλύτερη καθυστέρηση)
- $(A_1, B_1, C_1) \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow (C_5, S_4)$
- Για ένα πλήρη αθροιστή 4-bits \rightarrow 8 στάδια (ή επίπεδα) πυλών (n -bits: $2n$ στάδια πυλών)

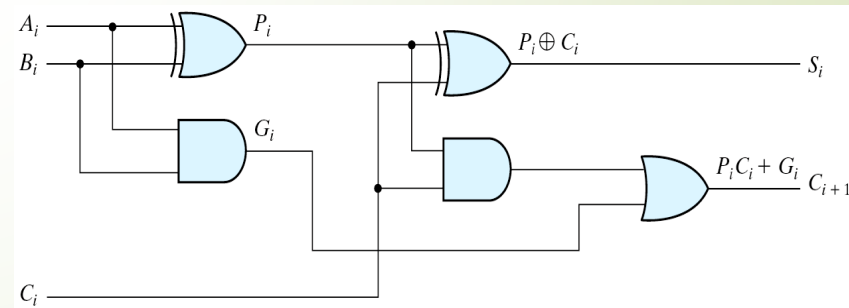


Πλήρης αθροιστής με τις P και G συναρτήσεις

Παράλληλοι Αθροιστές

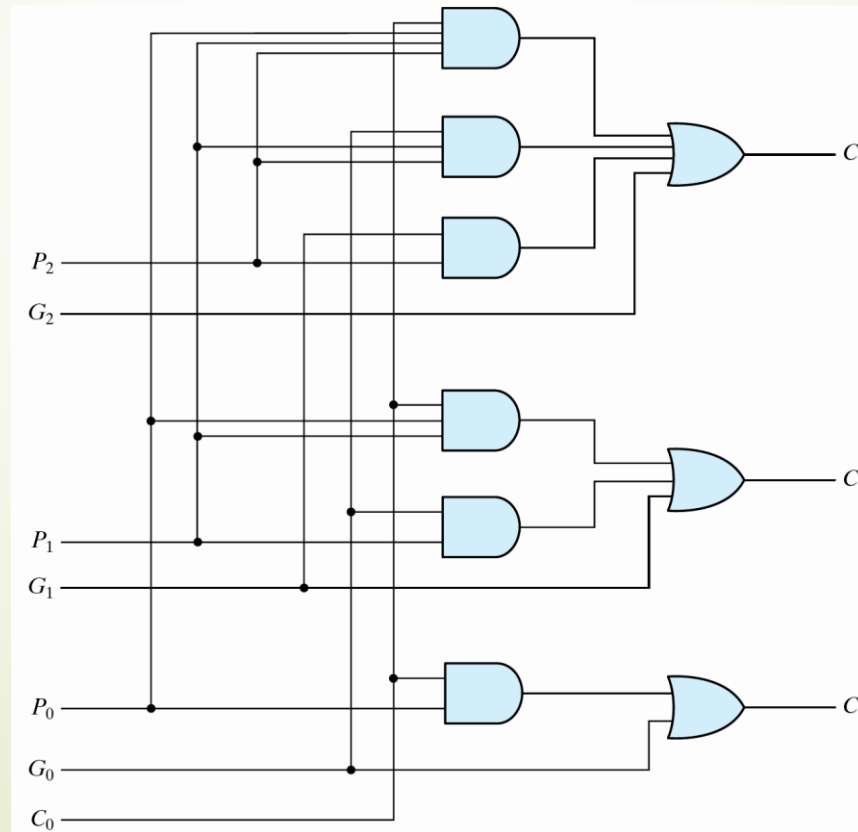
- Μείωση της καθυστέρησης που προκαλεί η διάδοση κρατούμενων

- Χρησιμοποιούν γρηγορότερες πύλες
- Πρόβλεψη κρατούμενου (περίπλοκος μηχανισμός, αλλά γρηγορότερος)
- Σήμα διάδοσης κρατούμενου: $P_i = A_i \oplus B_i$
- Σήμα παραγωγής κρατούμενου: $G_i = A_i B_i$
- Άθροισμα: $S_i = P_i \oplus C_i$
- Κρατούμενο: $C_{i+1} = G_i + P_i C_i$
- $C_0 = \text{Κρατούμενο εισόδου}$
- $C_1 = G_0 + P_0 C_0$
- $C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$
- $C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$



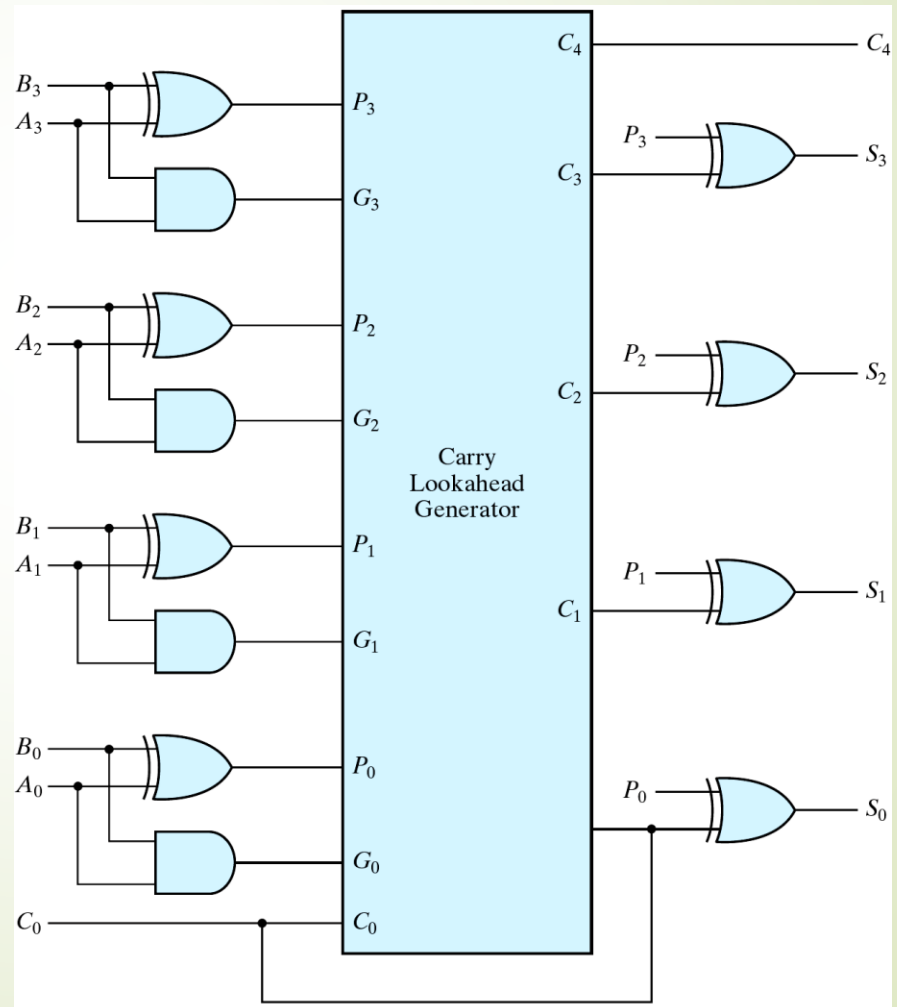
4-bit Αθροιστής με Πρόβλεψη Κρατούμενου

- Λογικό διάγραμμα γεννήτριας πρόβλεψης κρατούμενου



4-bit Αθροιστής με Πρόβλεψη Κρατούμενου

- Η καθυστέρηση διάδοσης των C_3 , C_2 και C_1 είναι ίσες,



Αθροιστής BCD

- 4-bits συν 4-bits
- Οι παράγοντες και το αποτέλεσμα: 0 ως 9

$$\begin{array}{r}
 + \quad x_3 \ x_2 \ x_1 \ x_0 \\
 + \quad y_3 \ y_2 \ y_1 \ y_0 \\
 \hline
 \end{array}$$

Cy s_3 s_2 s_1 s_0

$X+Y$	$x_3 \ x_2 \ x_1 \ x_0$	$y_3 \ y_2 \ y_1 \ y_0$	Sum	Cy	$s_3 \ s_2 \ s_1 \ s_0$
0 + 0	0 0 0 0	0 0 0 0	= 0	0	0 0 0 0
0 + 1	0 0 0 0	0 0 0 1	= 1	0	0 0 0 1
0 + 2	0 0 0 0	0 0 1 0	= 2	0	0 0 1 0
...
0 + 9	0 0 0 0	1 0 0 1	= 9	0	1 0 0 1
1 + 0	0 0 0 1	0 0 0 0	= 1	0	0 0 0 1
1 + 1	0 0 0 1	0 0 0 1	= 2	0	0 0 1 0
...
1 + 8	0 0 0 1	1 0 0 0	= 9	0	1 0 0 1
1 + 9	0 0 0 1	1 0 0 1	= A	0	1 0 1 0
2 + 0	0 0 1 0	0 0 0 0	= 2	0	0 0 1 0
...
9 + 9	1 0 0 1	1 0 0 1	= 18	1	0 0 1 0

Λάθος κώδικας

Λάθος BCD τιμή

0001 1000

Αθροιστής BCD

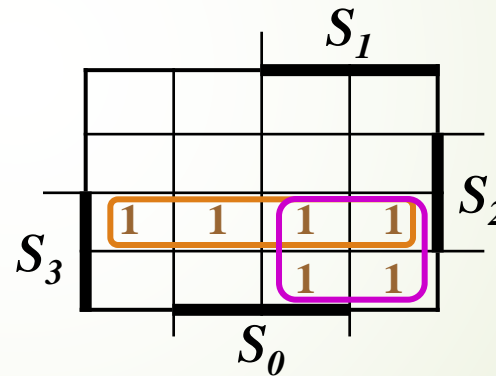
$X + Y$	$x_3 x_2 x_1 x_0$	$y_3 y_2 y_1 y_0$	<i>Sum</i>	<i>Cy</i>	$S_3 S_2 S_1 S_0$	<i>Required BCD Output</i>	<i>Value</i>
9 + 0	1 0 0 1	0 0 0 0	= 9	0	1 0 0 1	0 0 0 0 1 0 0 1	= 9
9 + 1	1 0 0 1	0 0 0 1	= 10	0	1 0 1 0	0 0 0 1 0 0 0 0	= 16 ✕
9 + 2	1 0 0 1	0 0 1 0	= 11	0	1 0 1 1	0 0 0 1 0 0 0 1	= 17 ✕
9 + 3	1 0 0 1	0 0 1 1	= 12	0	1 1 0 0	0 0 0 1 0 0 1 0	= 18 ✕
9 + 4	1 0 0 1	0 1 0 0	= 13	0	1 1 0 1	0 0 0 1 0 0 1 1	= 19 ✕
9 + 5	1 0 0 1	0 1 0 1	= 14	0	1 1 1 0	0 0 0 1 0 1 0 0	= 20 ✕
9 + 6	1 0 0 1	0 1 1 0	= 15	0	1 1 1 1	0 0 0 1 0 1 0 1	= 21 ✕
9 + 7	1 0 0 1	0 1 1 1	= 16	1	0 0 0 0	0 0 0 1 0 1 1 0	= 22 ✕
9 + 8	1 0 0 1	1 0 0 0	= 17	1	0 0 0 1	0 0 0 1 0 1 1 1	= 23 ✕
9 + 9	1 0 0 1	1 0 0 1	= 18	1	0 0 1 0	0 0 0 1 1 0 0 0	= 24 ✕

+ 6

Αθροιστής BCD

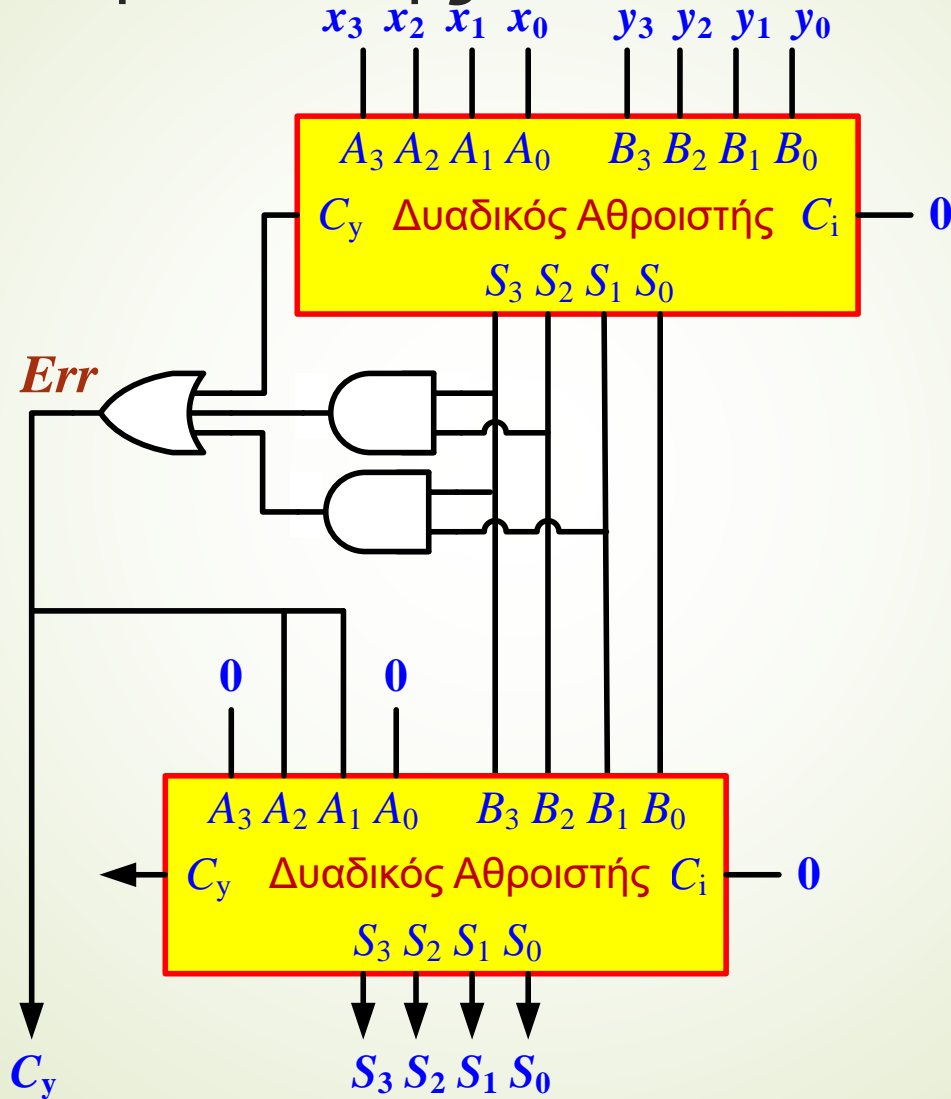
- Διορθώστε την έξοδο του δυαδικού αθροιστή (+6)
 - Αν το αποτέλεσμα είναι μεταξύ των 'A' και 'F'
 - Αν $Cy = 1$

$S_3 S_2 S_1 S_0$	<i>Err</i>
0 0 0 0	0
...	
1 0 0 0	0
1 0 0 1	0
1 0 1 0	1
1 0 1 1	1
1 1 0 0	1
1 1 0 1	1
1 1 1 0	1
1 1 1 1	1



$$Err = S_3 S_2 + S_3 S_1$$

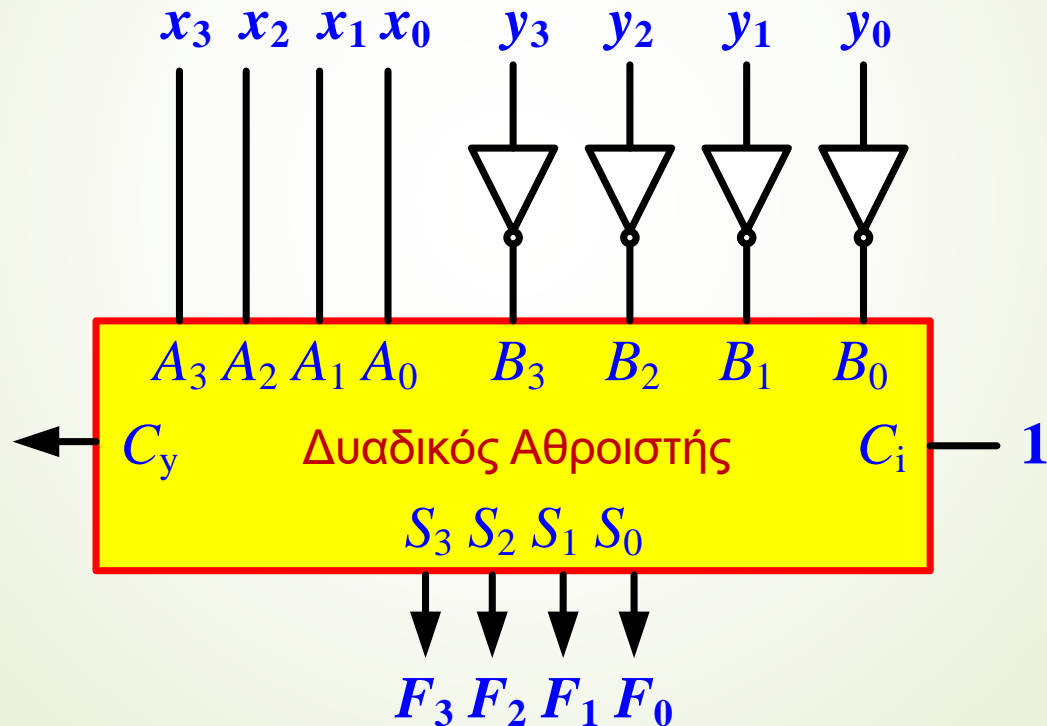
Αθροιστής BCD



Binary Subtractor

- Χρησιμοποιήστε το συμπλήρωμα ως προς 2 με δυαδικό αθροιστή

$$x - y = x + (-y) = x + y' + 1$$

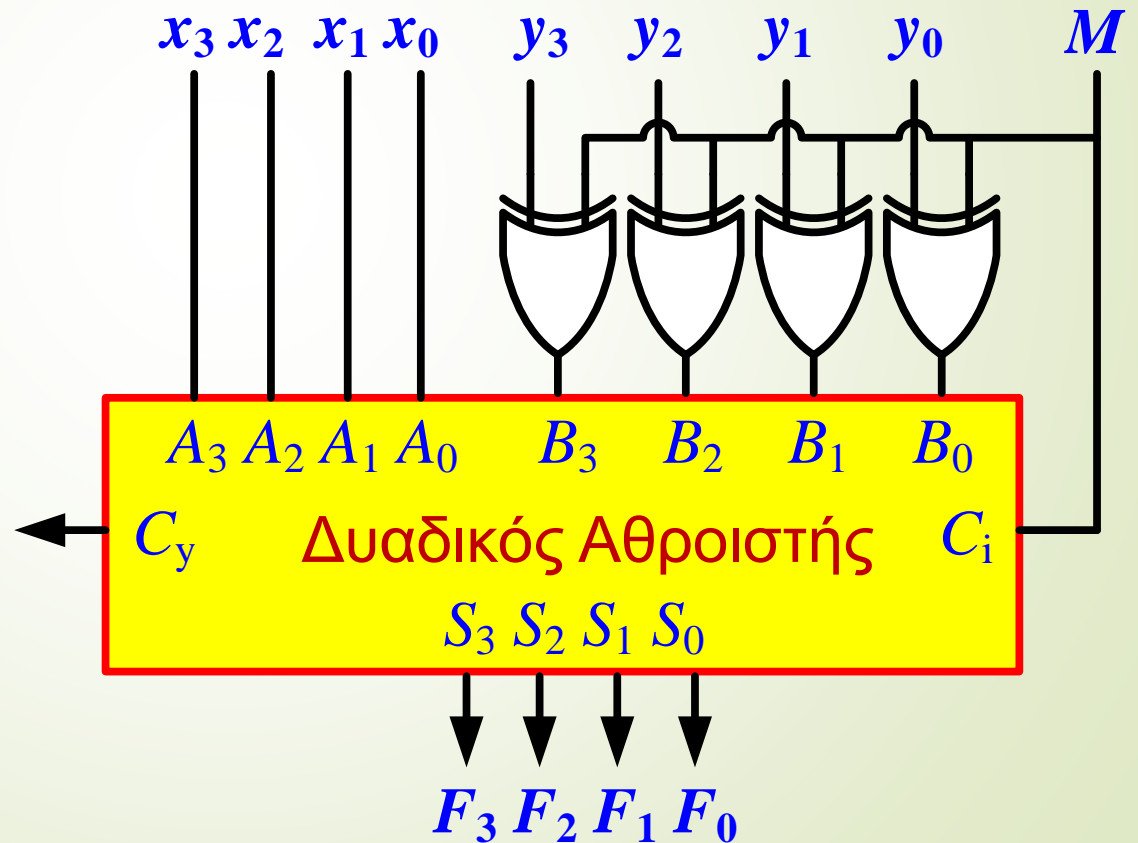


Binary Adder/Subtractor

➤ M : Σήμα Ελέγχου (Λειτουργία)

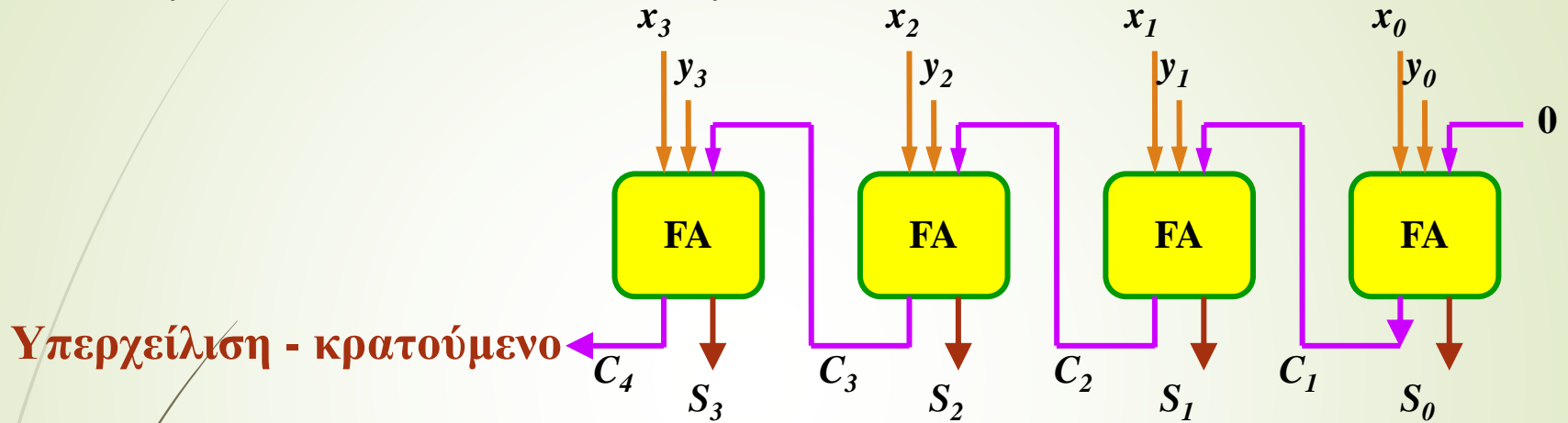
➤ $M=0 \rightarrow F = x + y$

➤ $M=1 \rightarrow F = x - y$

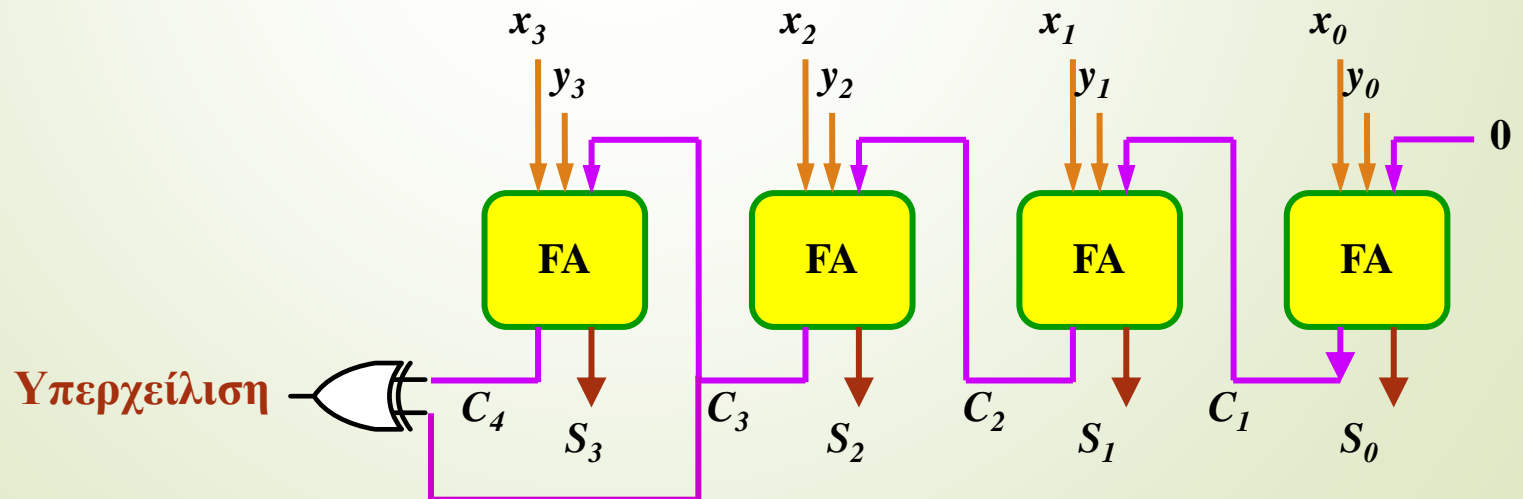


Υπερχείλιση

- Μη προσημασμένοι δυαδικοί αριθμοί



- Προσημασμένοι δυαδικοί αριθμοί (Συμπλήρωμα ως προς 2)



Συγκριτής Μεγέθους

- Συγκρίνετε δύο 4-bit αριθμούς
 - 3 Εξόδους: $<$, $=$, $>$
 - Με δυνατότητα επέκτασης σε αριθμούς με περισσότερα bits

$$x_3 = \bar{A}_3 \bar{B}_3 + A_3 B_3$$

$$x_2 = \bar{A}_2 \bar{B}_2 + A_2 B_2$$

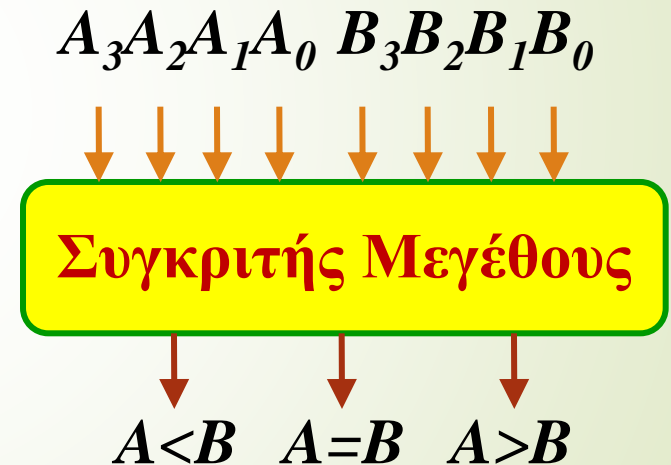
$$x_1 = \bar{A}_1 \bar{B}_1 + A_1 B_1$$

$$x_0 = \bar{A}_0 \bar{B}_0 + A_0 B_0$$

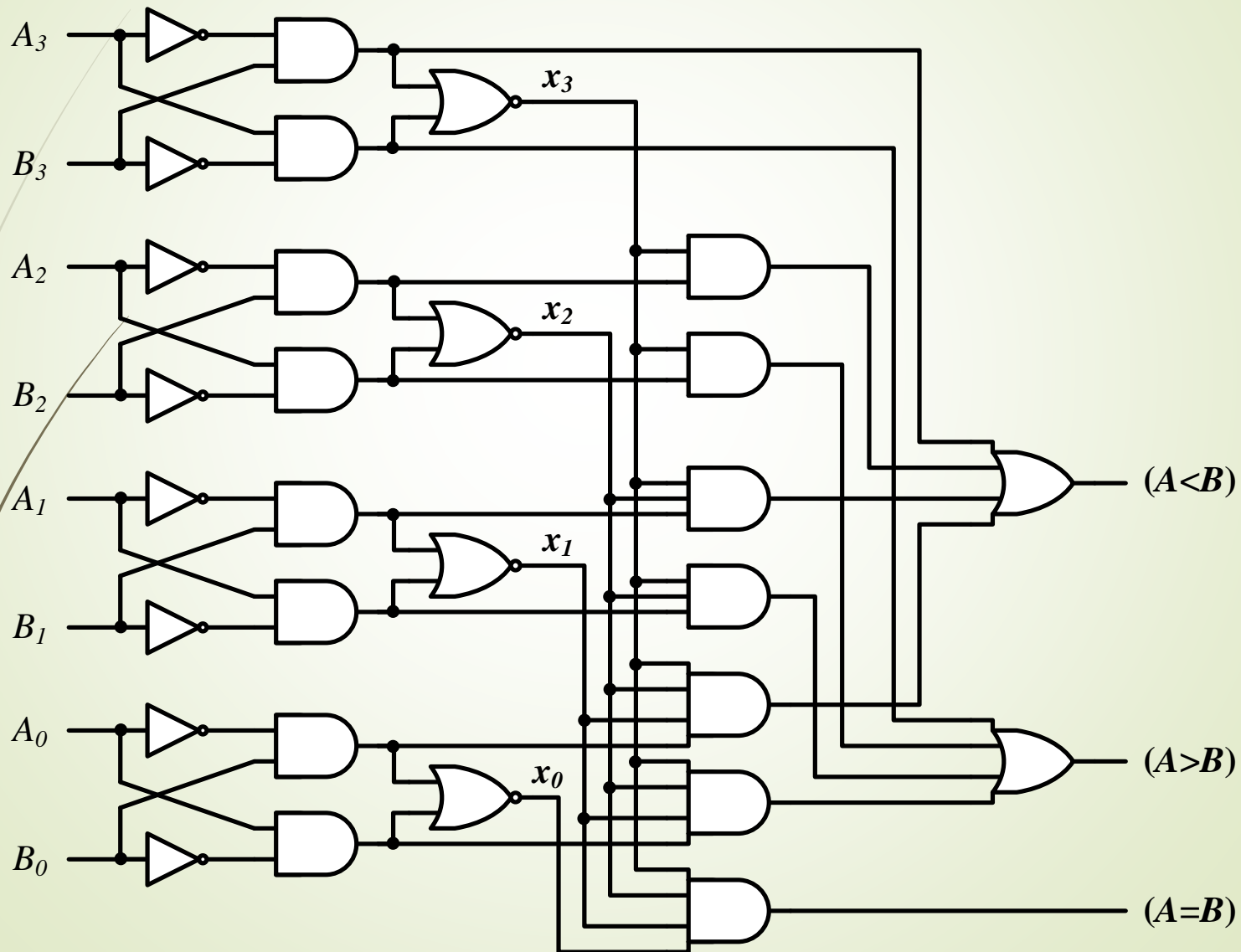
$$(A = B) = x_3 x_2 x_1 x_0$$

$$(A > B) = A_3 \bar{B}_3 + x_3 A_2 \bar{B}_2 + x_3 x_2 A_1 \bar{B}_1 + x_3 x_2 x_1 A_0 \bar{B}_0$$

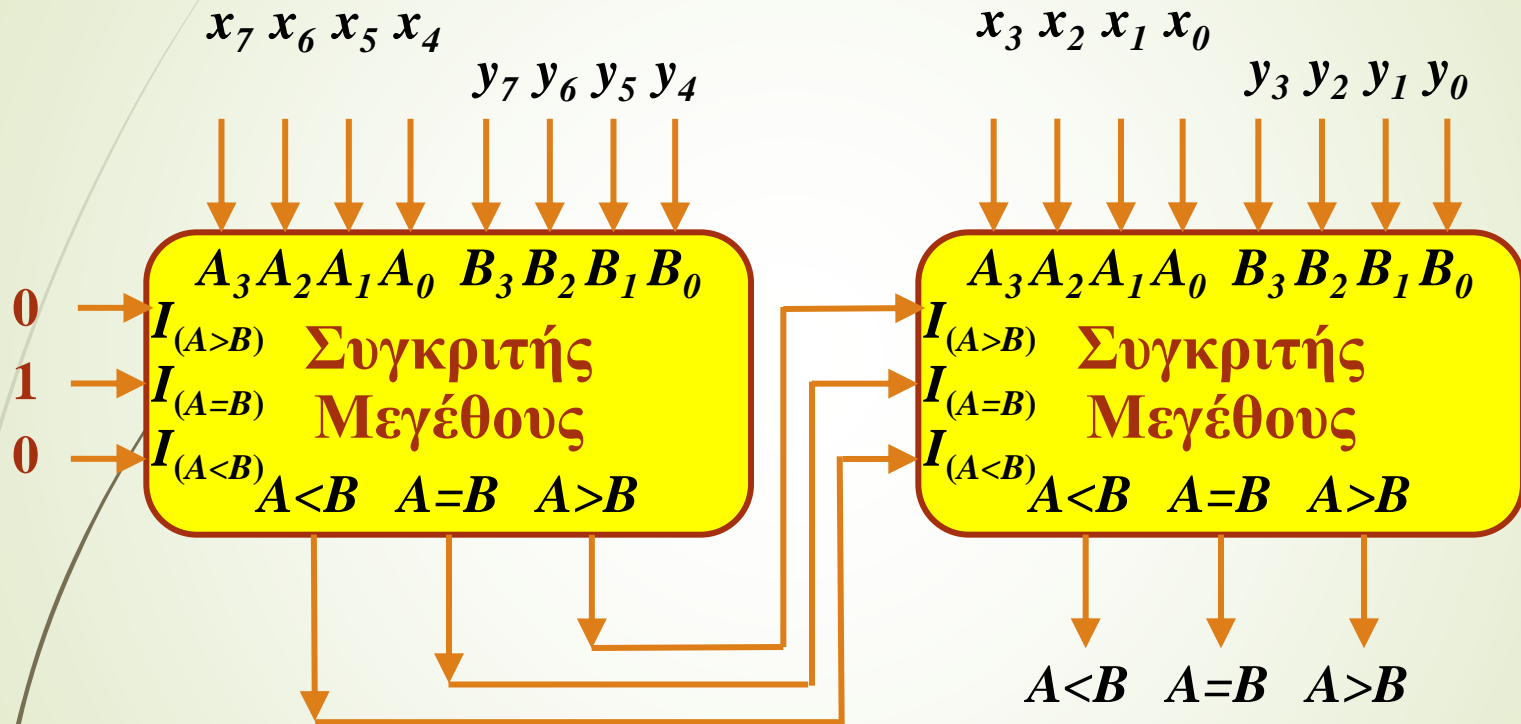
$$(A < B) = \bar{A}_3 B_3 + x_3 \bar{A}_2 B_2 + x_3 x_2 \bar{A}_1 B_1 + x_3 x_2 x_1 \bar{A}_0 B_0$$



Συγκριτής Μεγέθους



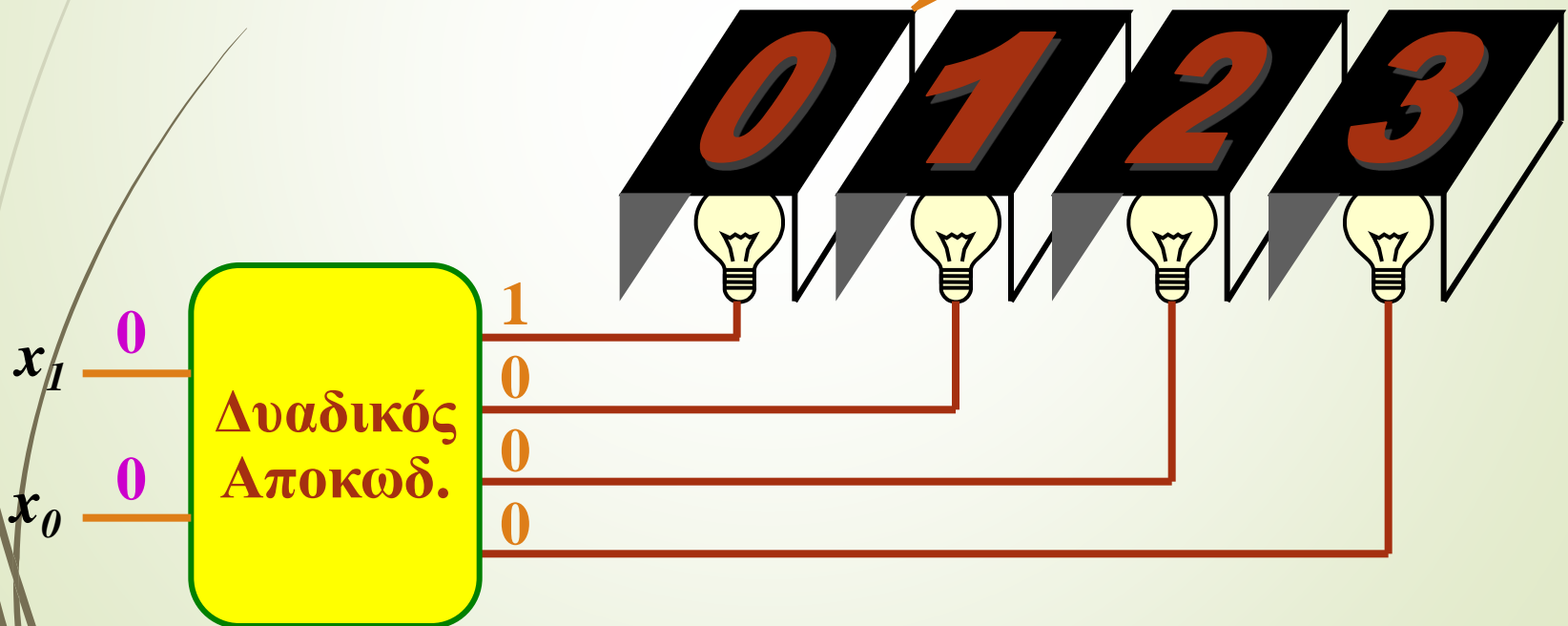
Συγκριτής Μεγέθους



Αποκωδικοποιητές

- Εξαγωγή “**Πληροφορίας**” από τον κώδικα
- Δυαδικός Αποκωδικοποιητής
 - Παράδειγμα: Δυαδικός αριθμός 2-bit

Μόνο *ένα*
led θα
ανάψει

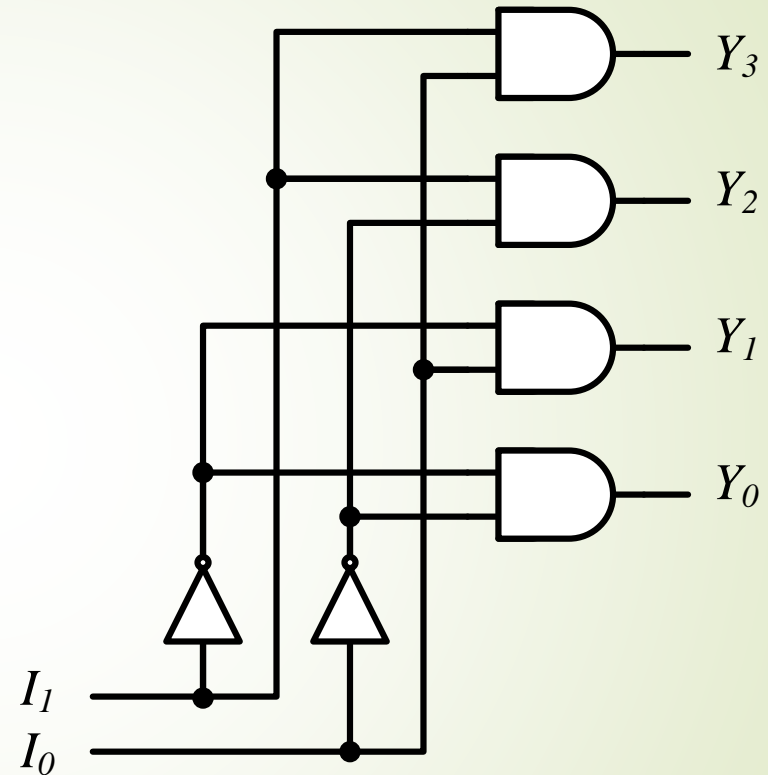


ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΕΣ

➤ ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗΣ 2-σε-4



I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



$$Y_3 = I_1 I_0$$

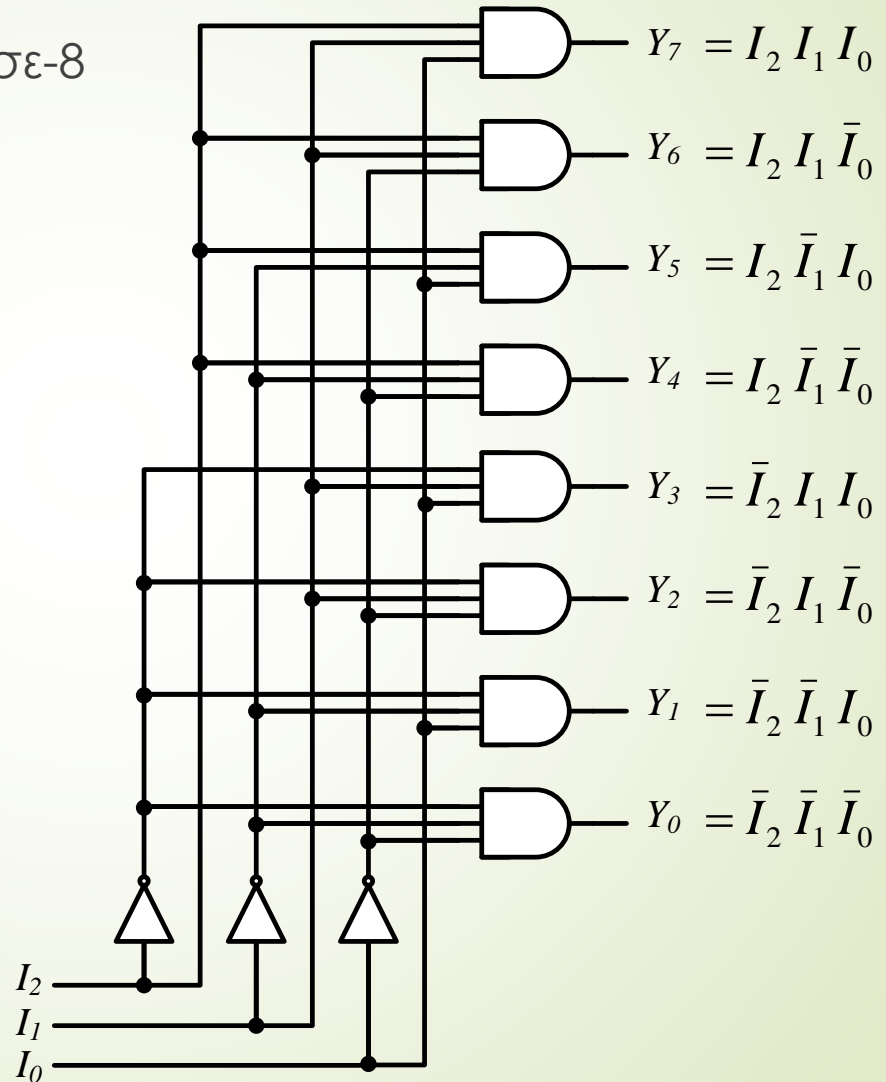
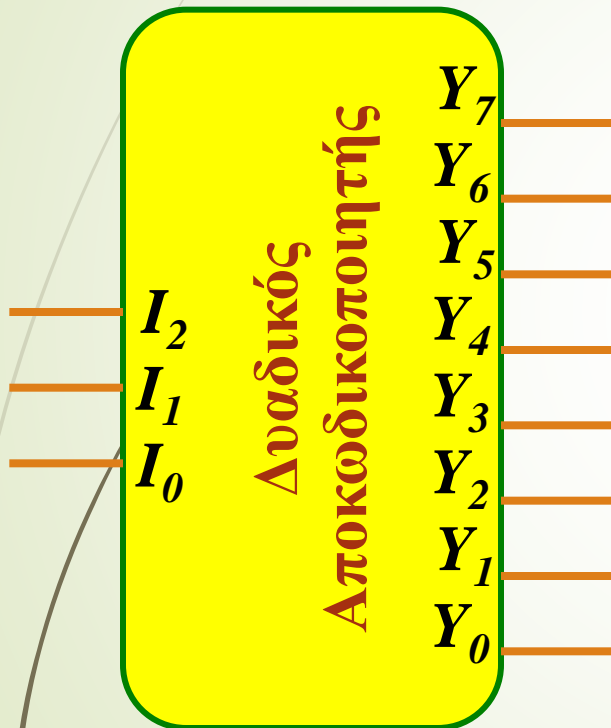
$$Y_2 = I_1 \bar{I}_0$$

$$Y_1 = \bar{I}_1 I_0$$

$$Y_0 = \bar{I}_1 \bar{I}_0$$

ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΕΣ

➤ Αποκωδικοποιητής 3-σε-8

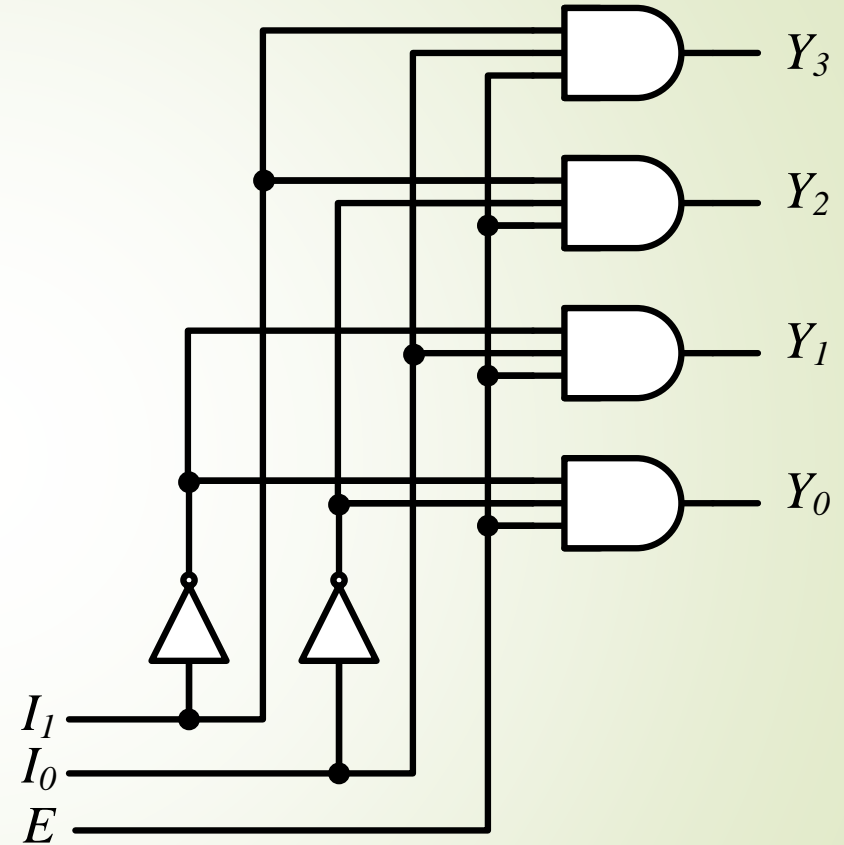


ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΕΣ

➤ Έλεγχος Επίτρεψης (*Enable*)



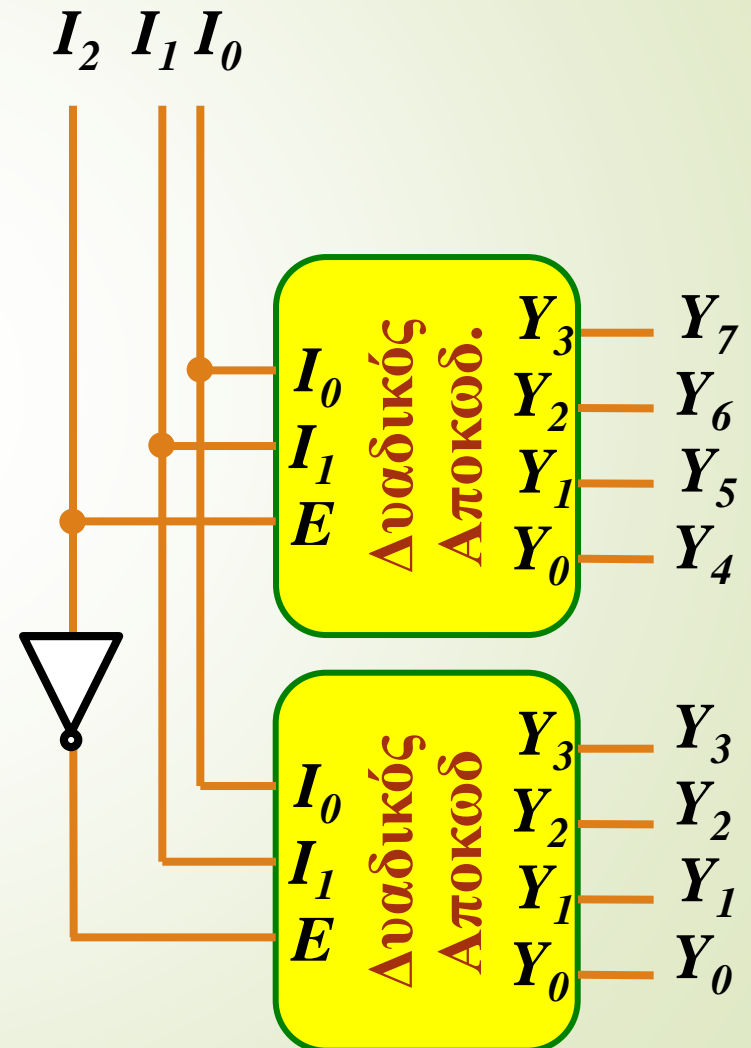
E	I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΕΣ

➤ Επέκταση

$I_2 I_1 I_0$	$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0$
0 0 0	0 0 0 0 0 0 0 1
0 0 1	0 0 0 0 0 0 1 0
0 1 0	0 0 0 0 0 1 0 0
0 1 1	0 0 0 0 1 0 0 0
1 0 0	0 0 0 1 0 0 0 0
1 0 1	0 0 1 0 0 0 0 0
1 1 0	0 1 0 0 0 0 0 0
1 1 1	1 0 0 0 0 0 0 0

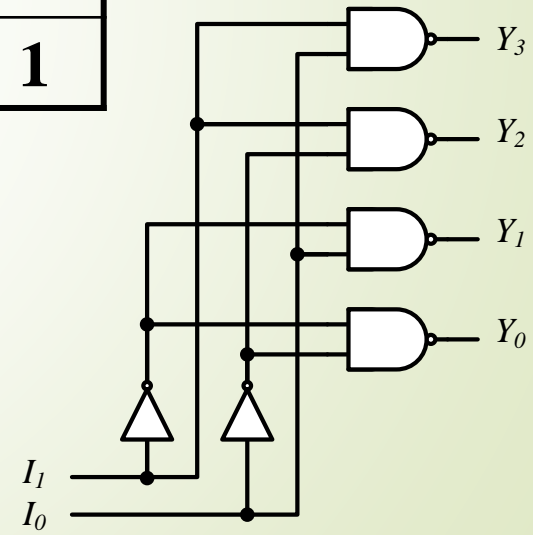
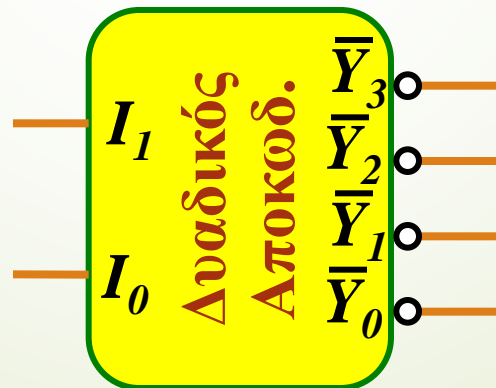
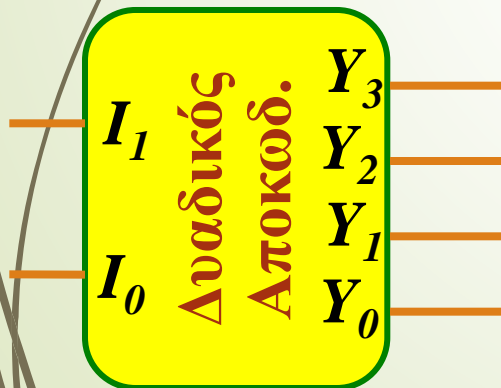


ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΕΣ

► Active-High / Active-Low

I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1



Υλοποίηση Συνδυαστικής Λογικής με Αποκωδικοποιητές

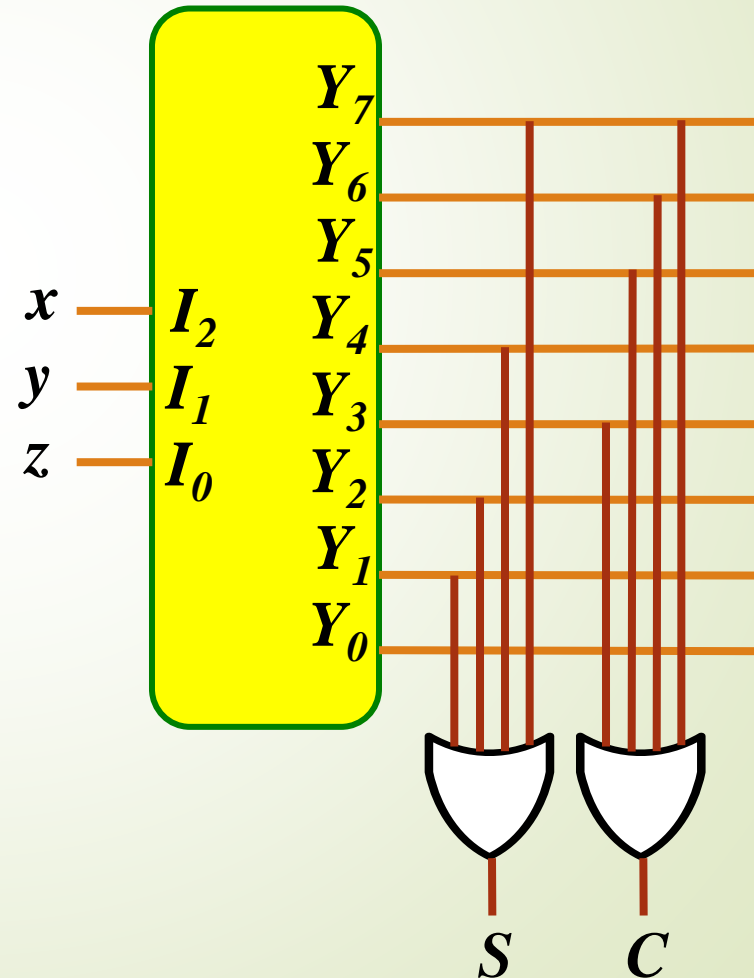
Δυαδικός Αποκωδικοποιητής

- Κάθε έξοδος είναι ένας ελαχιστόρος
- Παράγονται όλοι οι ελαχιστόροι
- Χρησιμοποιούμε τους ελαχιστόρους που χρειαζόμαστε

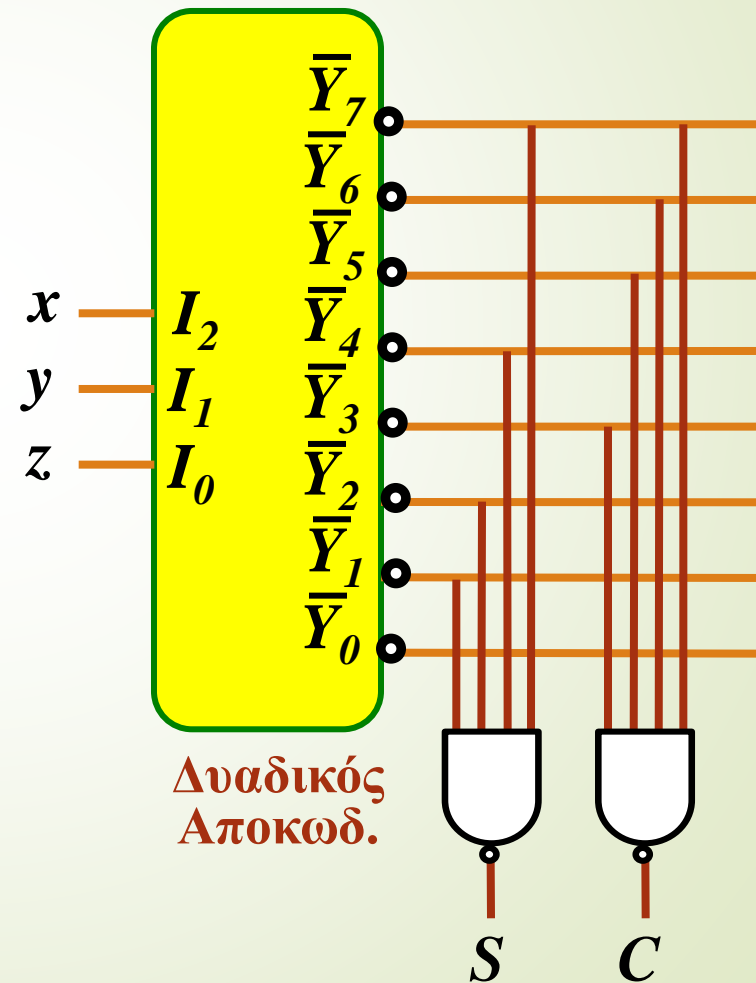
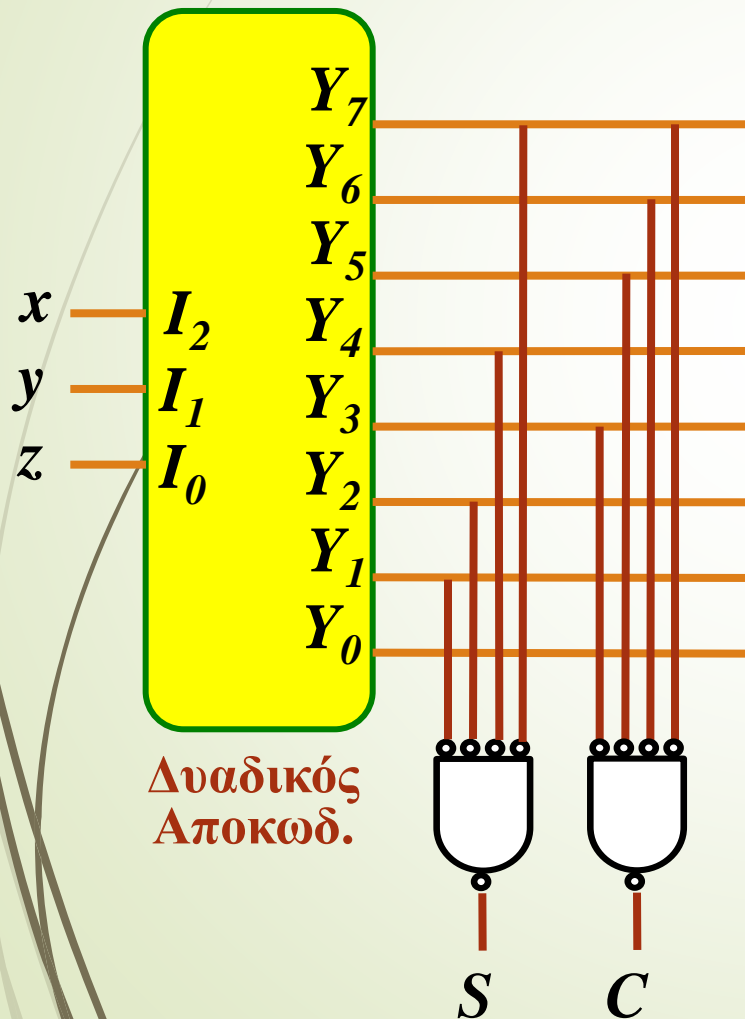
Παράδειγμα: Πλήρης Αθροιστής

$$S(x, y, z) = \sum(1, 2, 4, 7)$$

$$C(x, y, z) = \sum(3, 5, 6, 7)$$

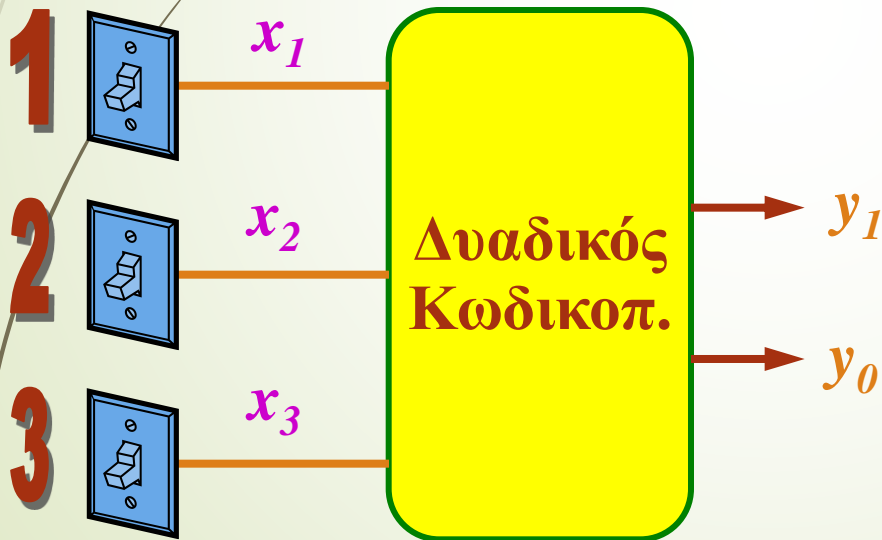


Υλοποίηση Συνδυαστικής Λογικής με Αποκωδικοποιητές



Κωδικοποιητές

- Εισάγουμε “Πληροφορία” σε κώδικα
- Δυαδικός Κωδικοποιητής
 - Παράδειγμα: Δυαδικός Κωδικοποιητής 4-σε-2



Μόνο **ένας**
διακόπτης
πρέπει να
ενεργοποιηθεί
κάθε φορά

x_3	x_2	x_1	y_1	y_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
1	0	0	1	1

Κωδικοποιητές

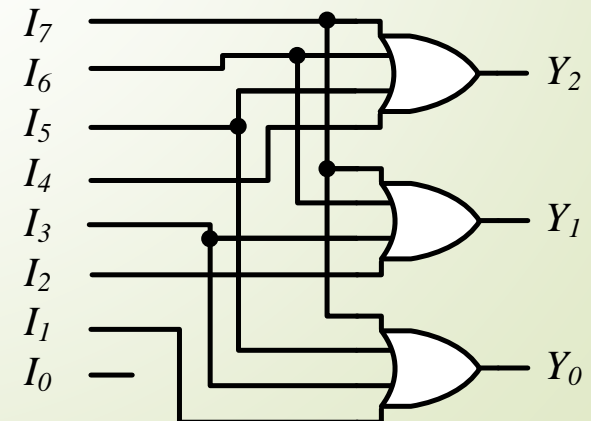
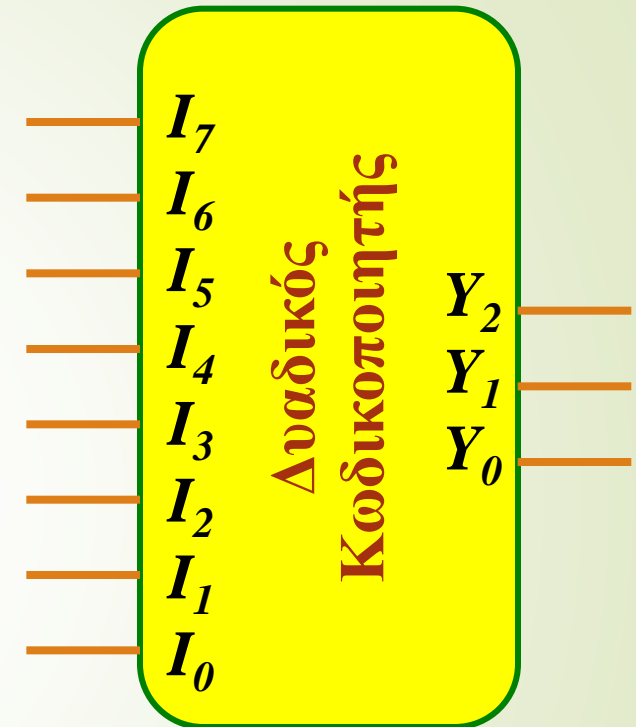
Κωδικοποιητής 8-σε-3

I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$$Y_2 = I_7 + I_6 + I_5 + I_4$$

$$Y_1 = I_7 + I_6 + I_3 + I_2$$

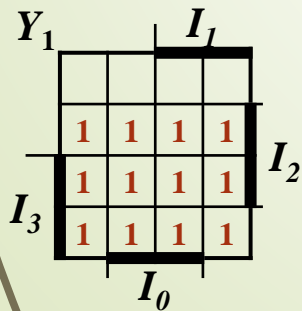
$$Y_0 = I_7 + I_5 + I_3 + I_1$$



Κωδικοποιητές Προτεραιότητας

➤ Κωδικοποιητής προτεραιότητας 4-εισόδων

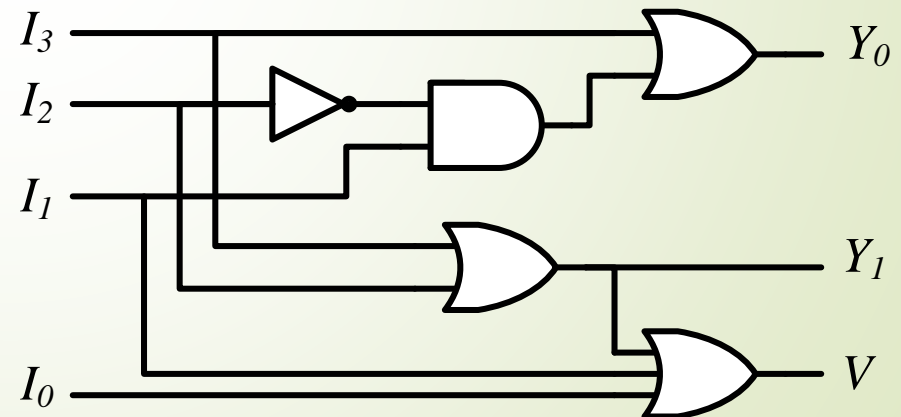
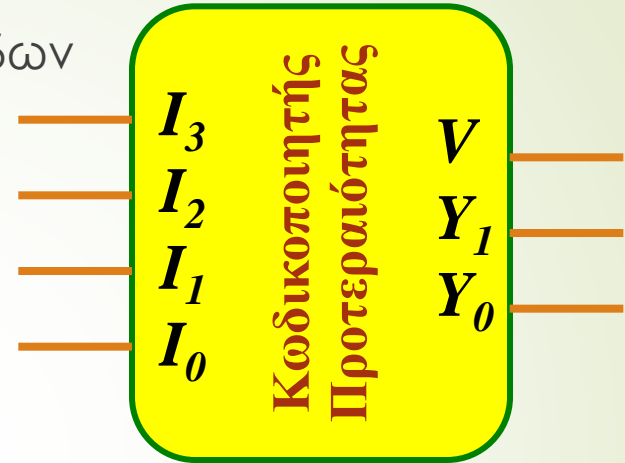
I_3	I_2	I_1	I_0	Y_1	Y_0	V
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1



$$Y_1 = I_3 + I_2$$

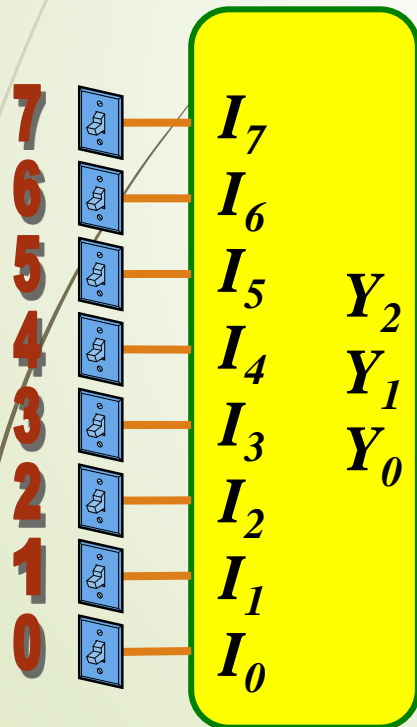
$$Y_0 = I_3 + \bar{I}_2 I_1$$

$$V = I_3 + I_2 + I_1 + I_0$$

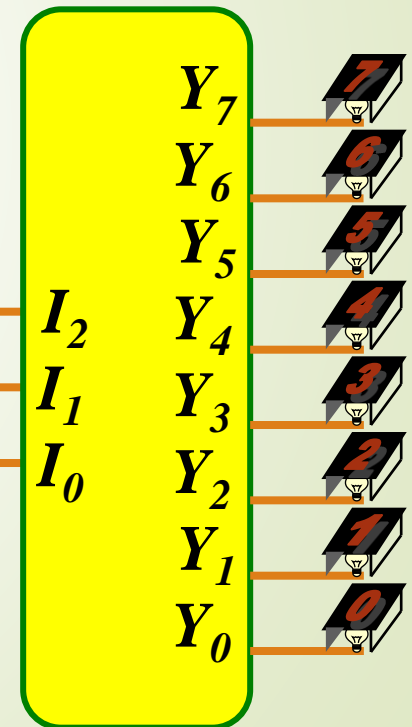


Ζεύγη Κωδικοποιητών / Αποκωδικοποιητών

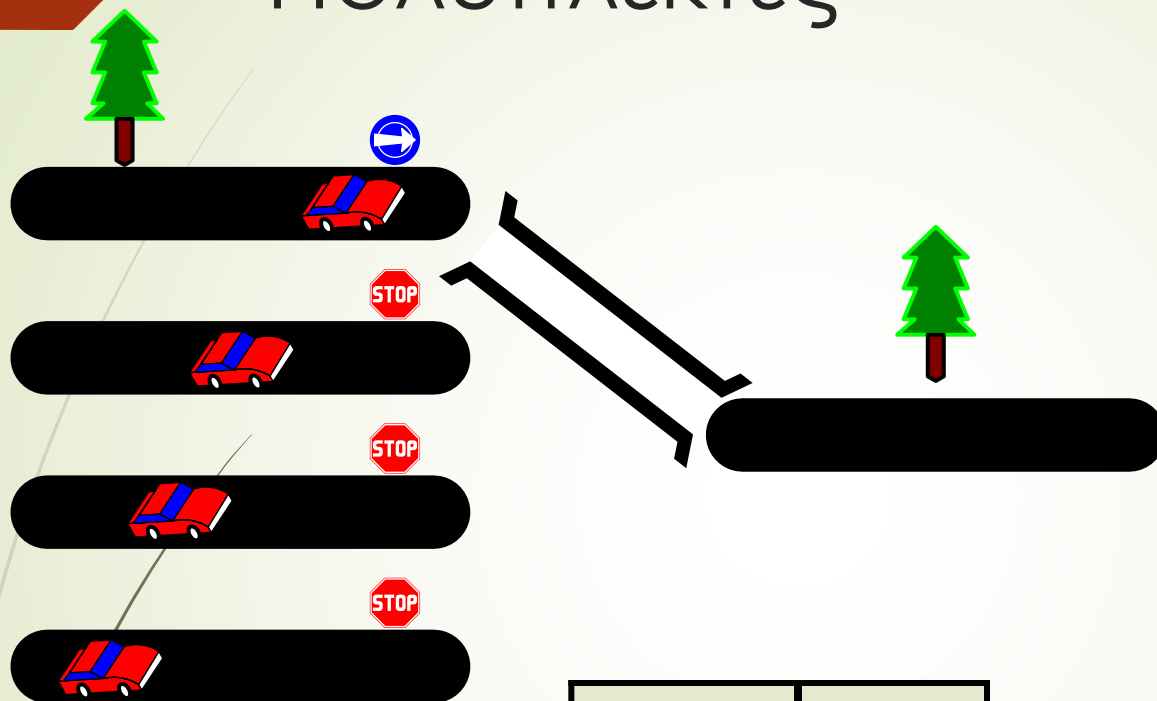
Δυαδικός
Κωδικοποιητής



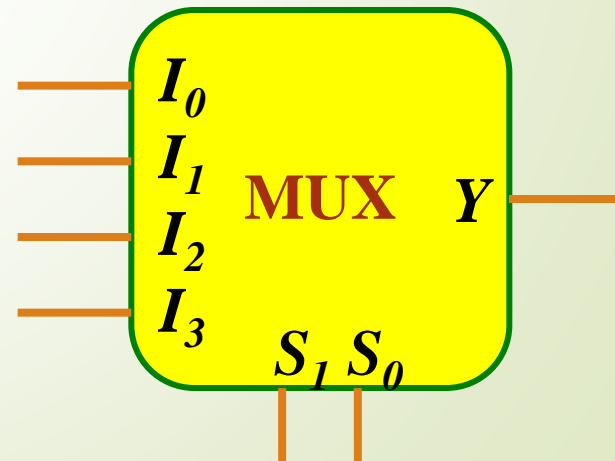
Δυαδικός
Αποκωδικοποιητής



Πολυπλέκτες

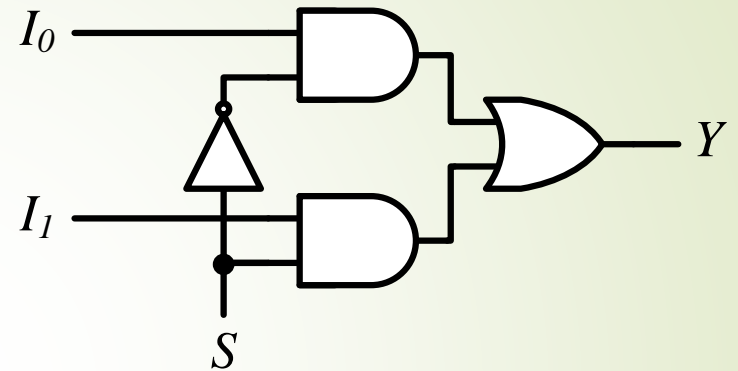
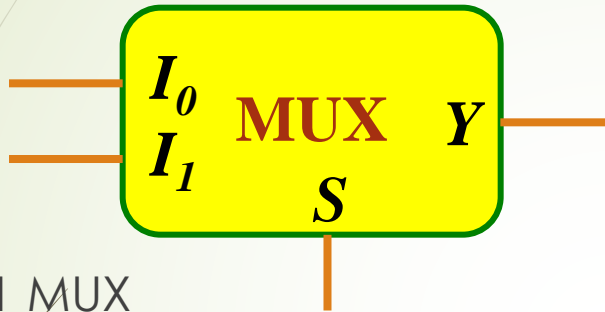


S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

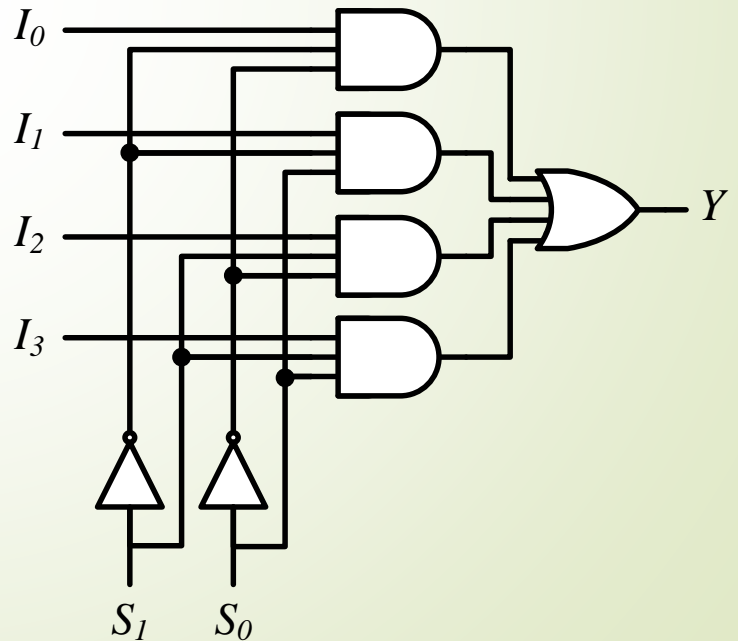
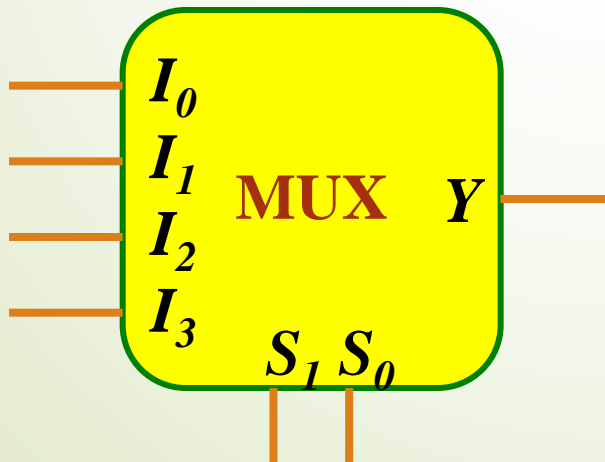


ΠΟΛΥΠΛΕΚΤΕΣ

➤ 2-σε-1 MUX

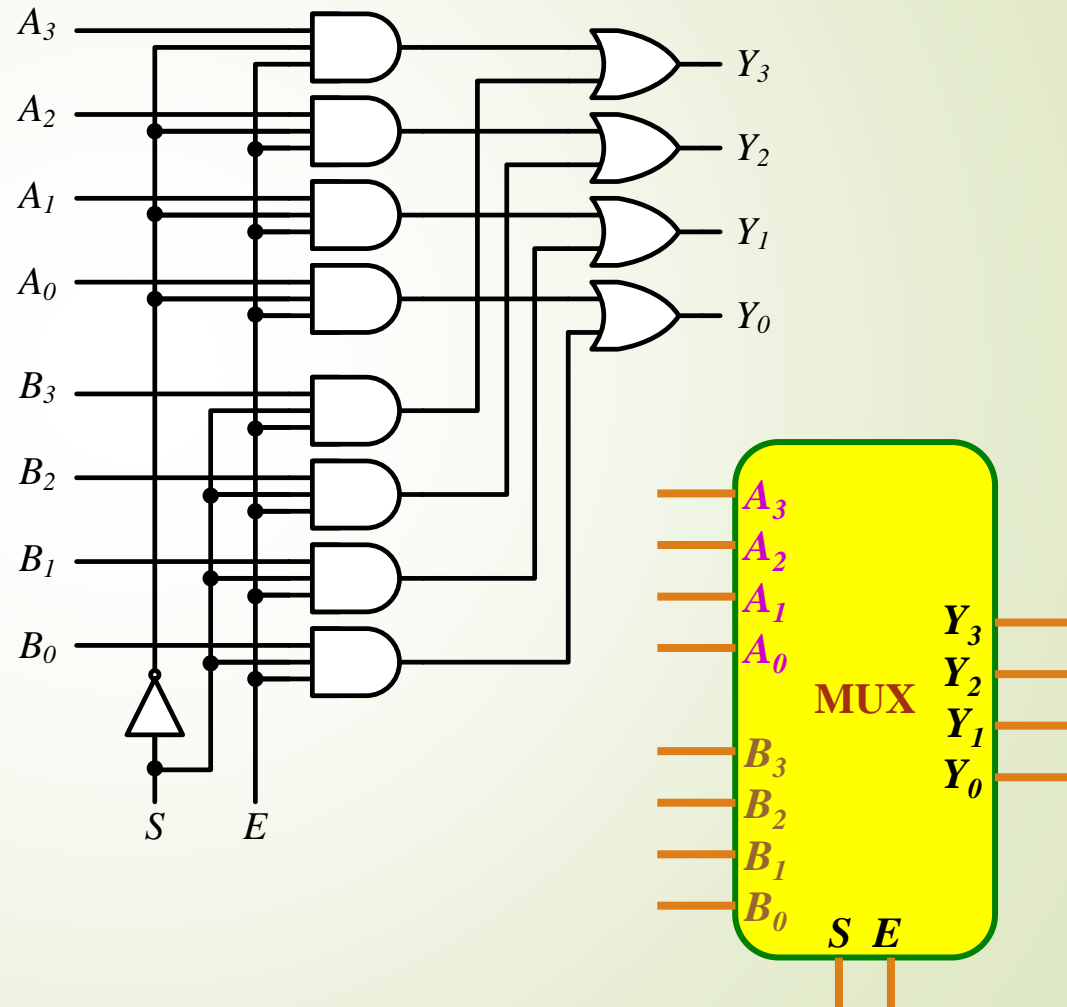
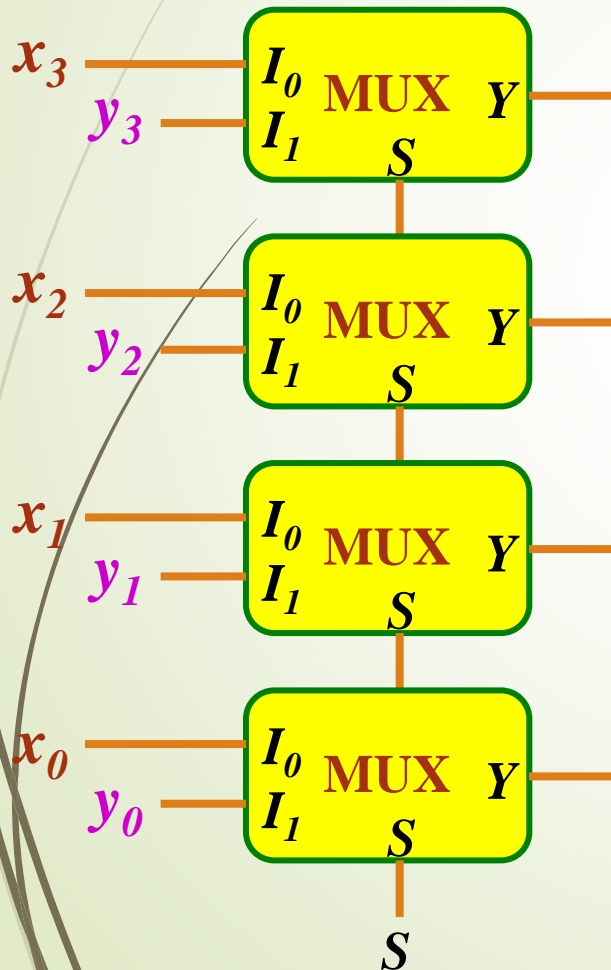


➤ 4-σε-1 MUX



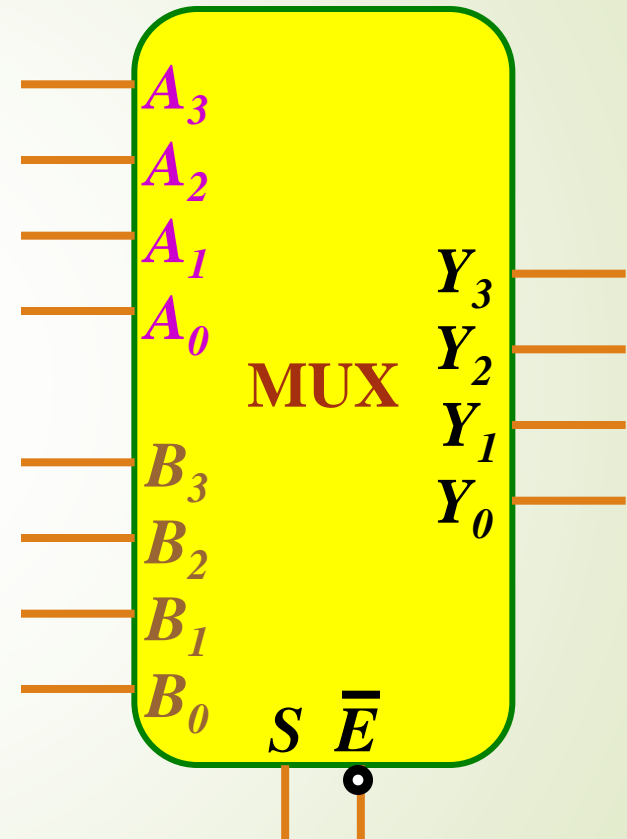
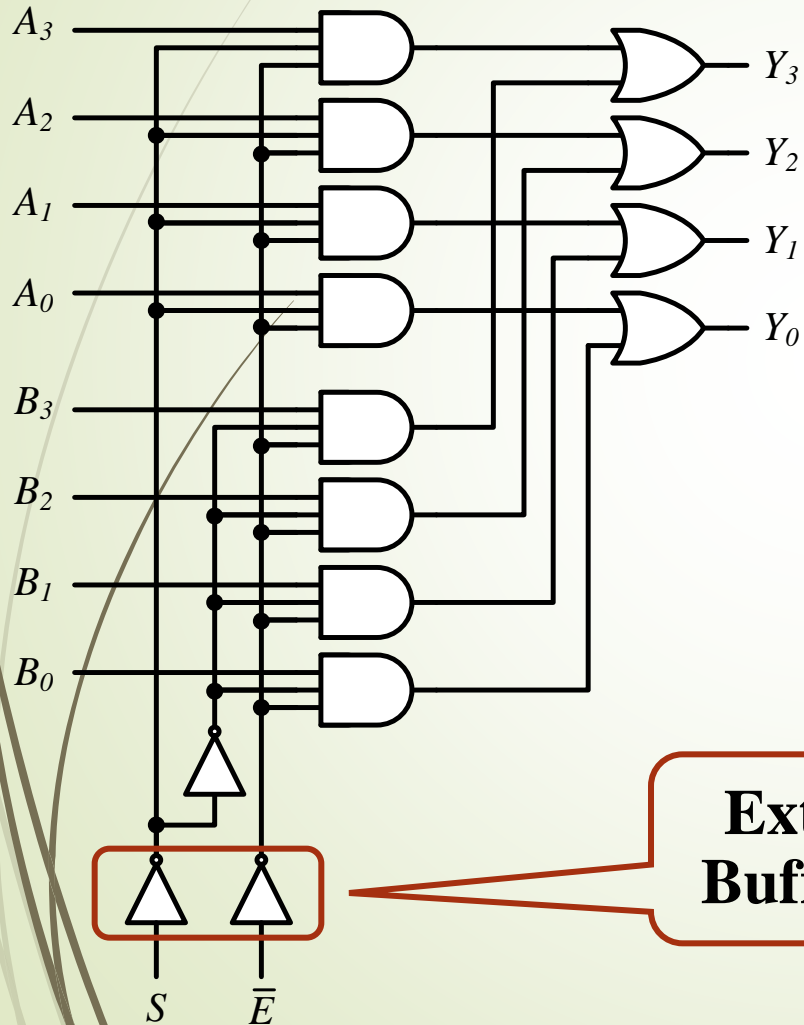
Πολυπλέκτες

- Τετραπλός πολυπλέκτης 2-σε-1 (MUX)



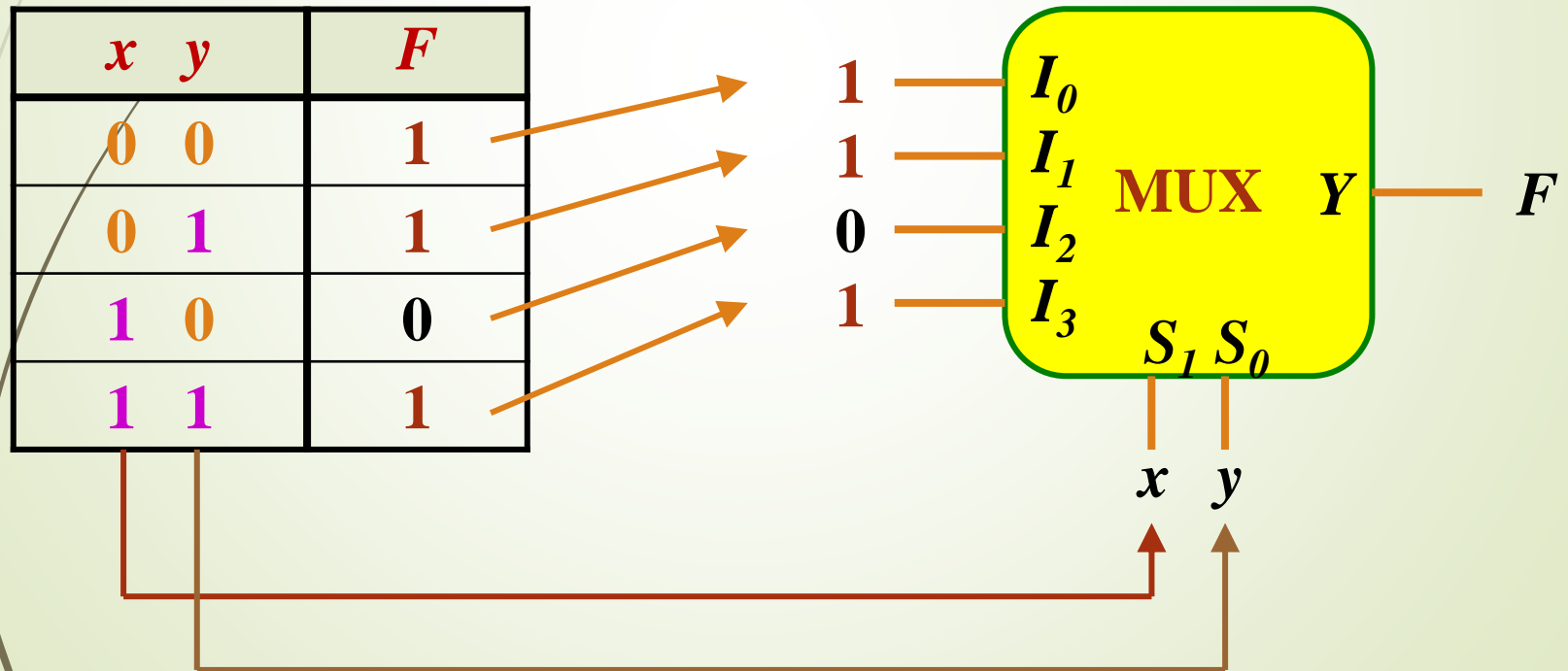
ΠΟΛΥΠΛΕΚΤΕΣ

➤ Τετραπλός πολυπλέκτης 2-σε-1 (MUX)



Υλοποίηση Συνδυαστικών Κυκλωμάτων με Πολυπλέκτες

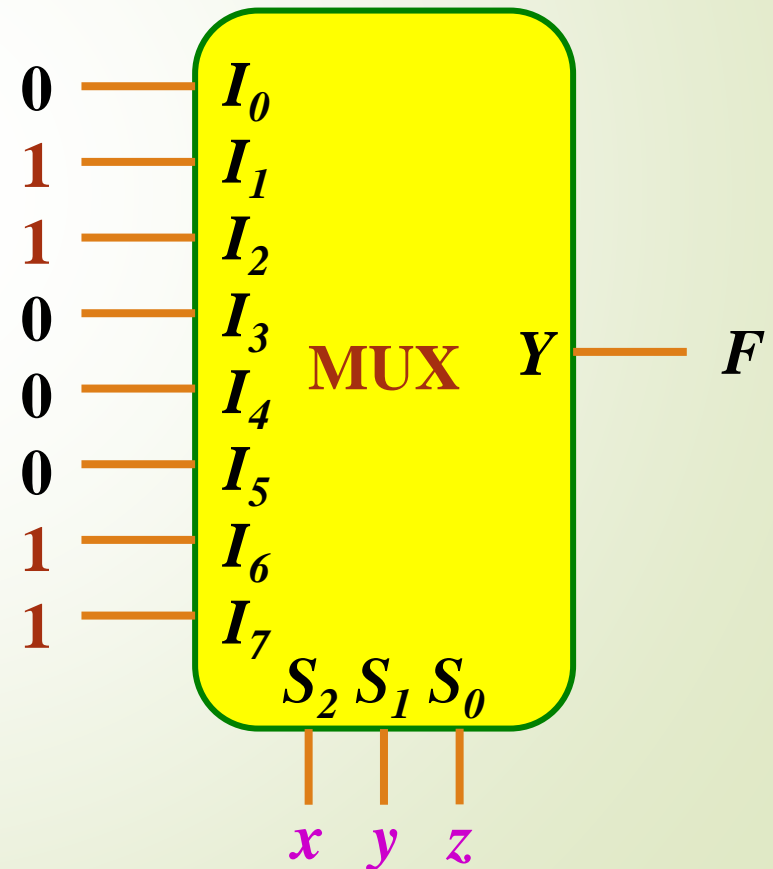
- Παράδειγμα
 $F(x, y) = \sum(0, 1, 3)$



Υλοποίηση Συνδυαστικών Κυκλωμάτων με Πολυπλέκτες

- Παράδειγμα
 $F(x, y, z) = \sum(1, 2, 6, 7)$

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



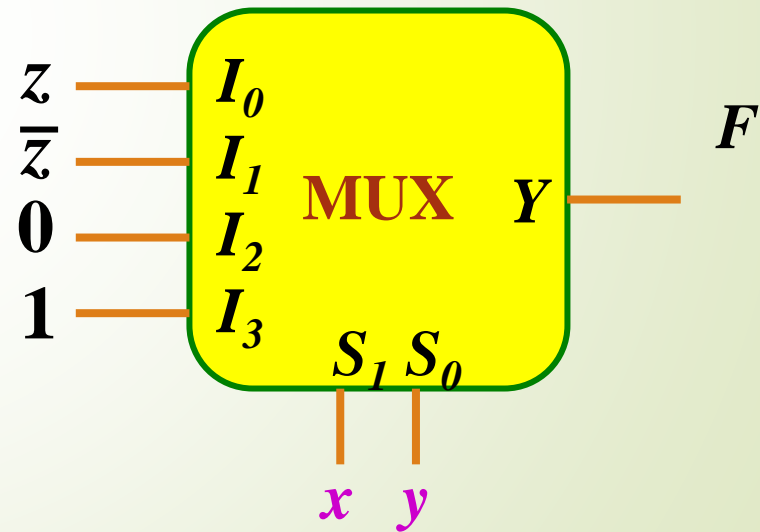
Υλοποίηση Συνδυαστικών Κυκλωμάτων με Πολυπλέκτες

► Παράδειγμα

$$F(x, y, z) = \sum(1, 2, 6, 7)$$

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$\left. \begin{array}{l} F = z \\ F = \bar{z} \end{array} \right\}$
 $\left. \begin{array}{l} F = 0 \\ F = 1 \end{array} \right\}$



Υλοποίηση Συνδυαστικών Κυκλωμάτων με Πολυπλέκτες

► Παράδειγμα

$$F(A, B, C, D) = \sum(1, 3, 4, 11, 12, 13, 14, 15)$$

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

} $F = D$

} $F = D$

} $F = \bar{D}$

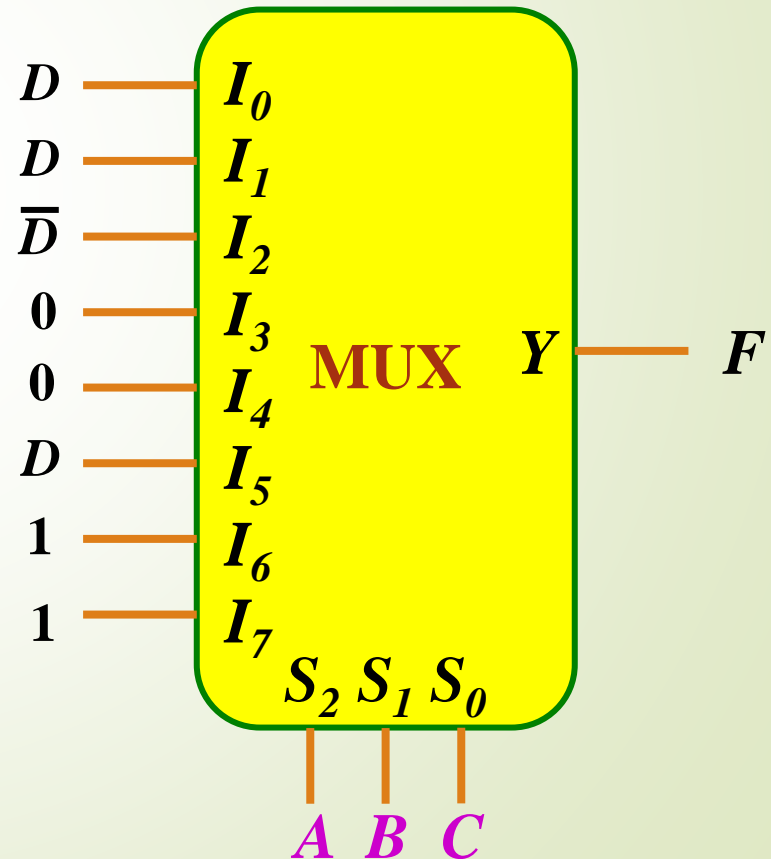
} $F = 0$

} $F = 0$

} $F = D$

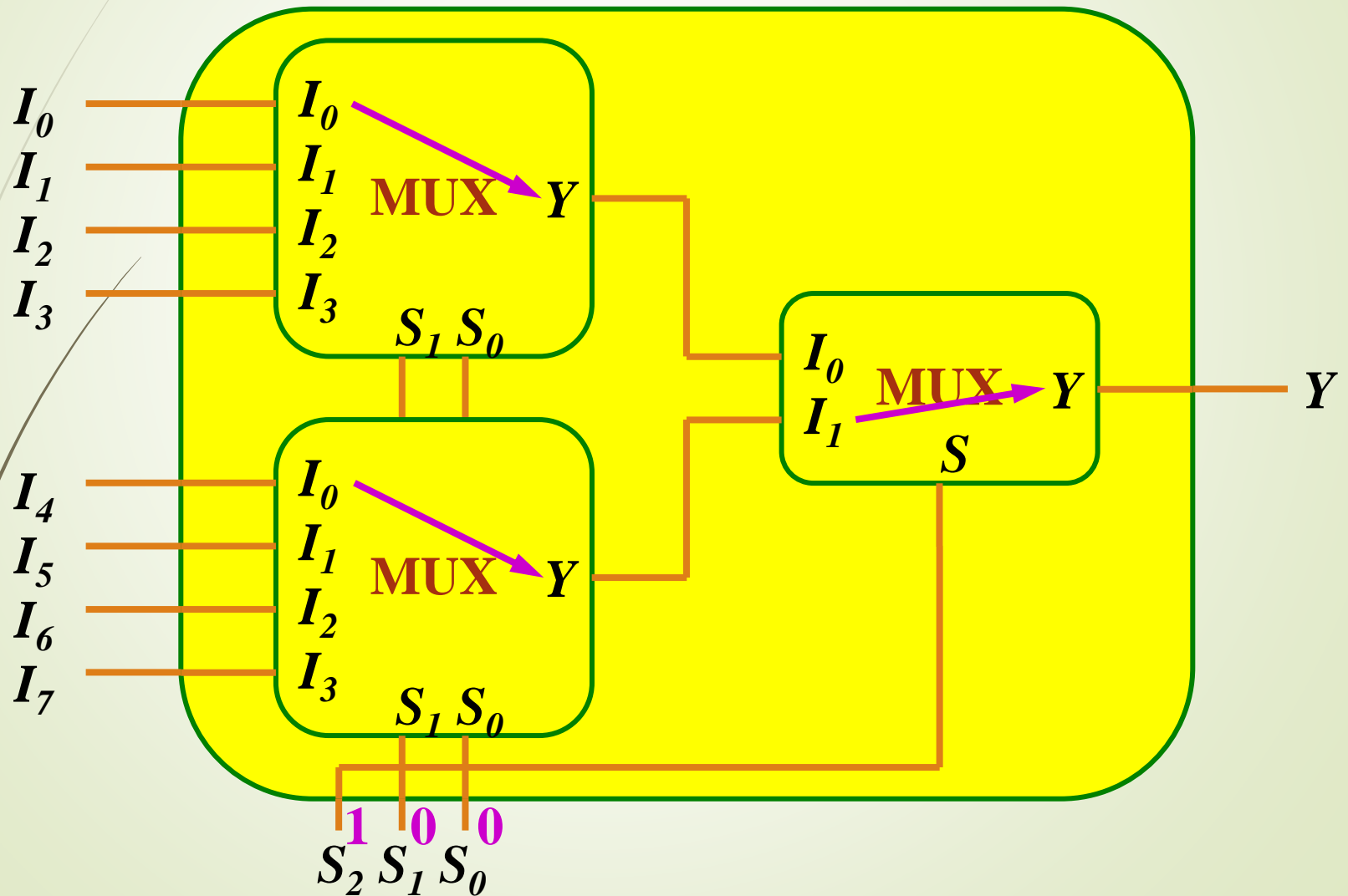
} $F = 1$

} $F = 1$

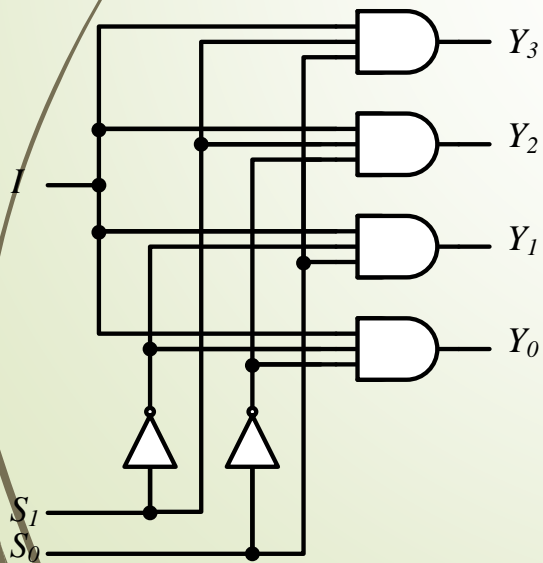
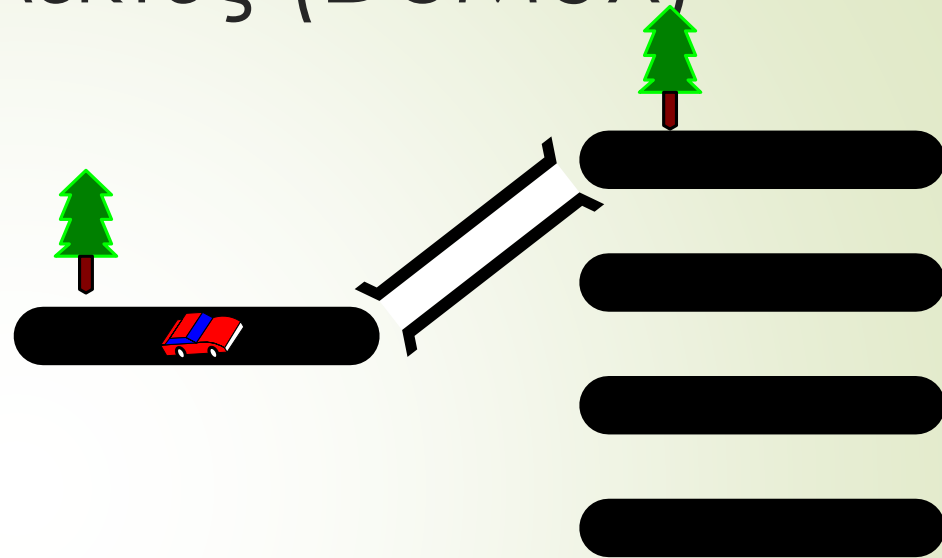
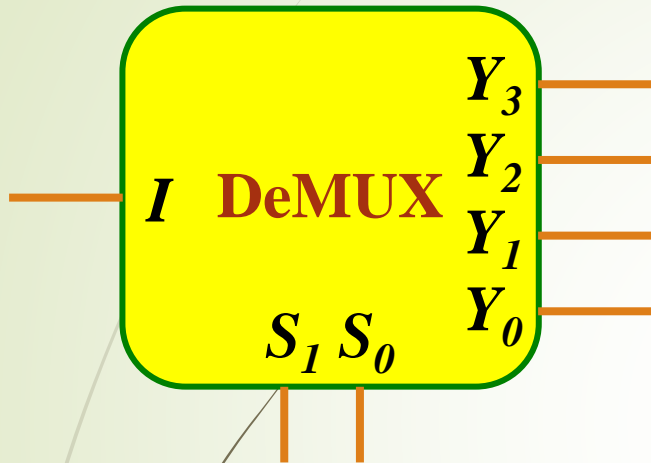


Multiplexer Expansion

- 8-to-1 MUX με χρήση δύο 4-σε-1 MUX

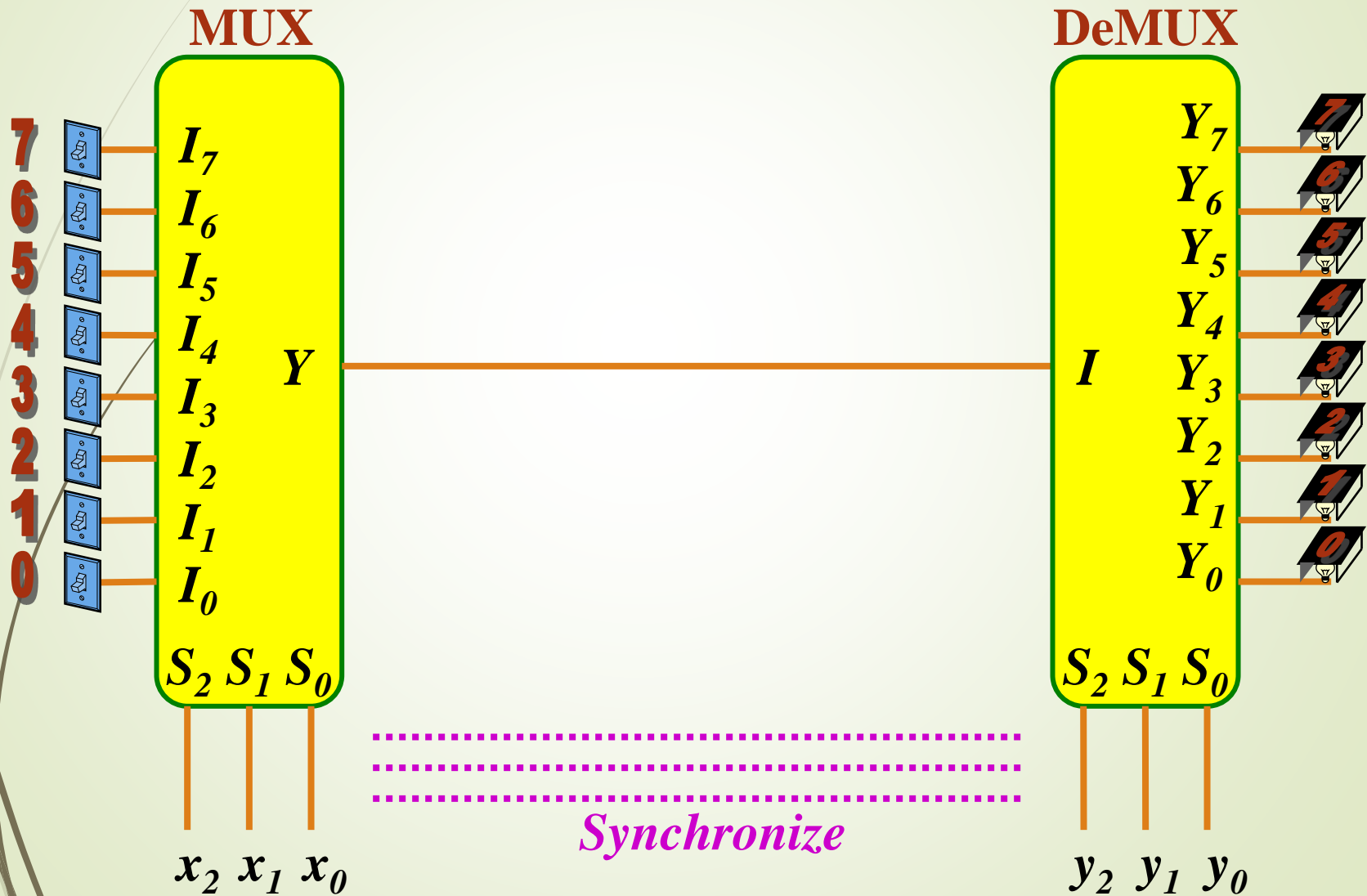


ΑΠΟΠΟΛΥΠΛΕΚΤΕΣ (DeMUX)

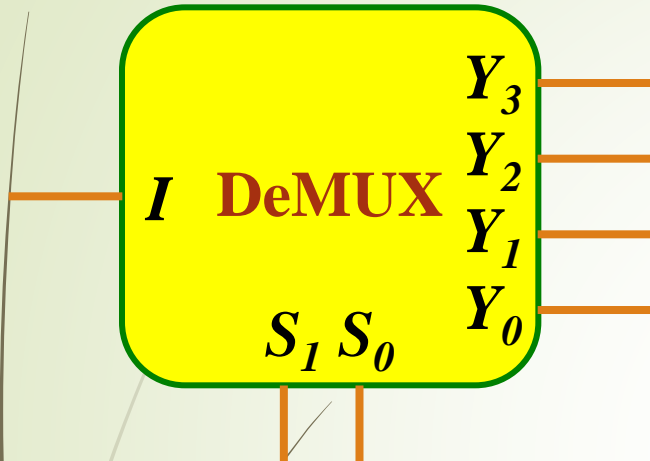


S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

Ζεύγη MUX / DEMUX



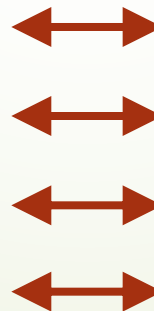
DeMux Αποκωδικοποιητές



S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

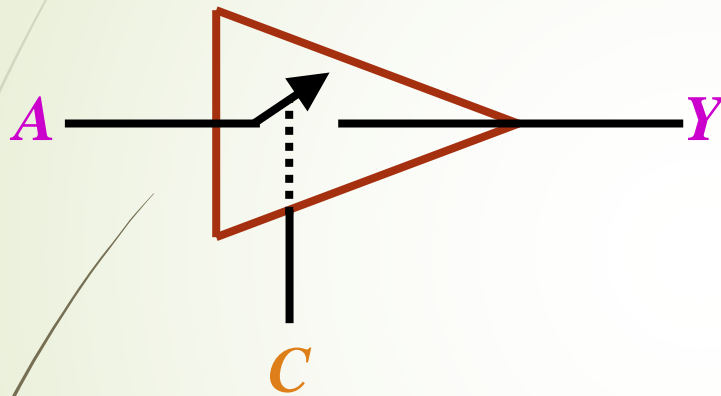


E	I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

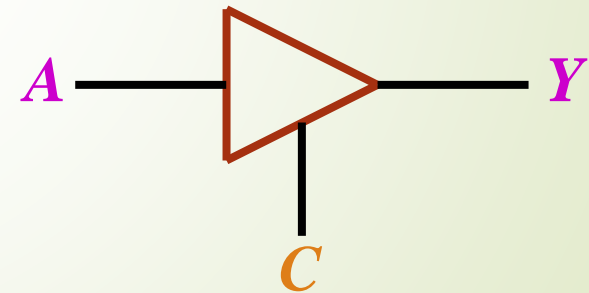


Τρισταθείς Πύλες

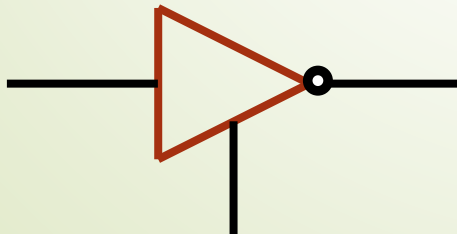
- Τρισταθής Απομονωτής (Buffer)



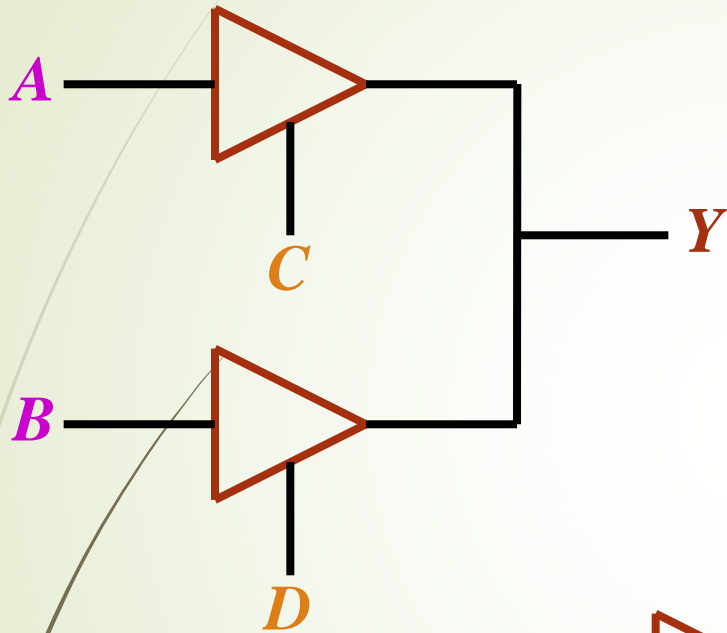
<i>C</i>	<i>A</i>	<i>Y</i>
0	x	Hi-Z
1	0	0
1	1	1



- Τρισταθής Αντιστροφέας (Inverter)

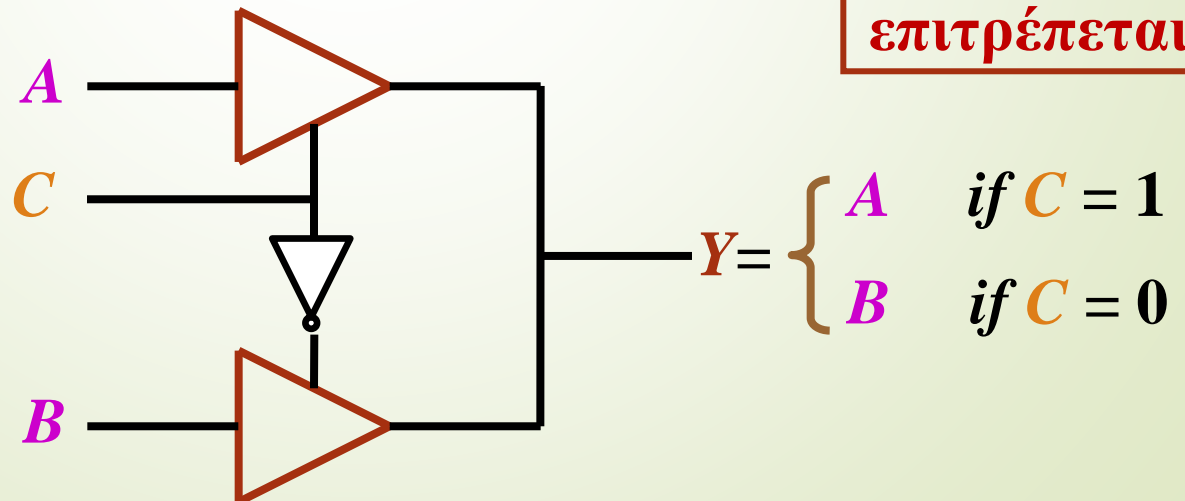


Τρισταθείς Πύλες



C	D	Y
0	0	Hi-Z
0	1	B
1	0	A
1	1	?

Δεν
επιτρέπεται



Τρισταθείς Πύλες

