Τμήμα Πληροφορικής
Πανεπιστήμιο Πειραιώς

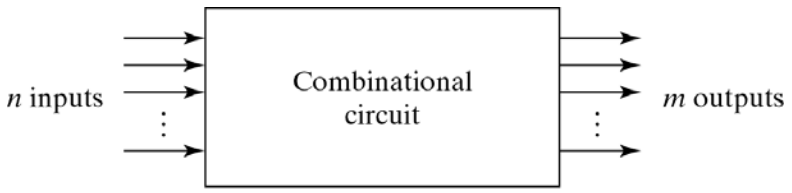
Λογική Σχεδίαση Ψηφιακών Συστημάτων

Συνδυαστική Λογική (Combinational Logic)

Μιχάλης Ψαράκης

Γενικό Συνδυαστικό Κύκλωμα

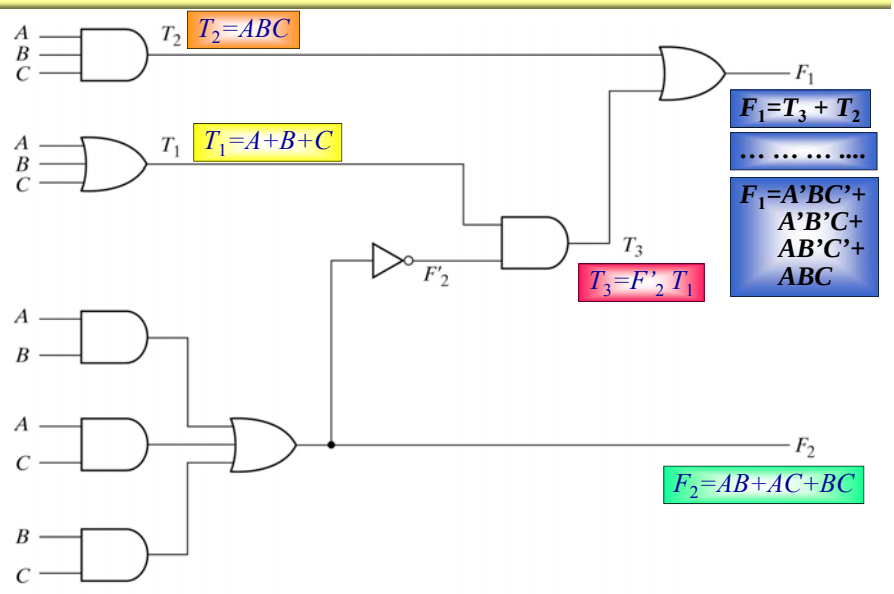
- Συνδυαστικό κύκλωμα (Combinational circuit)
  - Η τιμή των εξόδων εξαρτάται μόνο από την τιμή των εισόδων
  - Ονομάζεται συνδυαστικό, διότι ο συνδυασμός των τιμών των εισόδων περιγράφει μόνος του τη λειτουργία του κυκλώματος
- Αντίθετα, η λειτουργία των ακολουθιακών κυκλωμάτων περιγράφεται από μια ακολουθία καταστάσεων
  - Τα μελετούμε στα επόμενα κεφάλαια



Διαδικασία ανάλυσης

- Ανάλυση συνδυαστικού κυκλώματος:
  - Ο προσδιορισμός των λογικών συναρτήσεων που υλοποιούν
- Βήματα διαδικασίας ανάλυσης
  - Ονομάζουμε τις εξόδους των πυλών (ό,τι σύμβολο ή όνομα θέλουμε) που εξαρτώνται μόνο από τις εισόδους
    - Βρίσκουμε τις συναρτήσεις αυτών των πυλών
  - Στη συνέχεια, ονομάζουμε τις εξόδους των πυλών που είναι συναρτήσεις των προηγουμένων πυλών και των εισόδων με νέα σύμβολα/ονόματα
    - Βρίσκουμε τις συναρτήσεις και αυτών των πυλών
  - Επαναλαμβάνουμε το προηγούμενο βήμα μέχρι να οδηγηθούμε στις εξόδους του κυκλώματος
  - Αντικαθιστώντας διαδοχικά τις λογικές συναρτήσεις, υπολογίζουμε τις λογικές συναρτήσεις των εξόδων σαν συνάρτηση των μεταβλητών εισόδου

Παράδειγμα ανάλυσης



Πίνακας αληθείας παραδείγματος

A	B	C	F_2	F_2'	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0
1	1	1	1	0	1	1	0	1

Διαδικασία σχεδίασης

- Φυσικά, είναι το αντίθετο της ανάλυσης:
  - Δεδομένης μιας προδιαγραφής (specification) να παραχθεί λογικό κύκλωμα που να την ικανοποιεί
- Βήματα σχεδίασης
  - Από τις προδιαγραφές καθορίζουμε το πλήθος εισόδων και εξόδων και τους δίνουμε ονόματα (αυθαίρετα ή περιγραφικά)
  - Εξάγουμε τον πίνακα αληθείας που καθορίζει τη σχέση εισόδου/εξόδου
    - Μιλούμε για συνδυαστικά κυκλώματα
  - Βρίσκουμε τις εξόδους ως λογικές συναρτήσεις των εισόδων
    - Και πιθανόν τις απλοποιούμε
  - Κατασκευάζουμε το λογικό διάγραμμα
    - Και επαληθεύουμε την ορθότητα της σχεδίασης του διαγράμματος

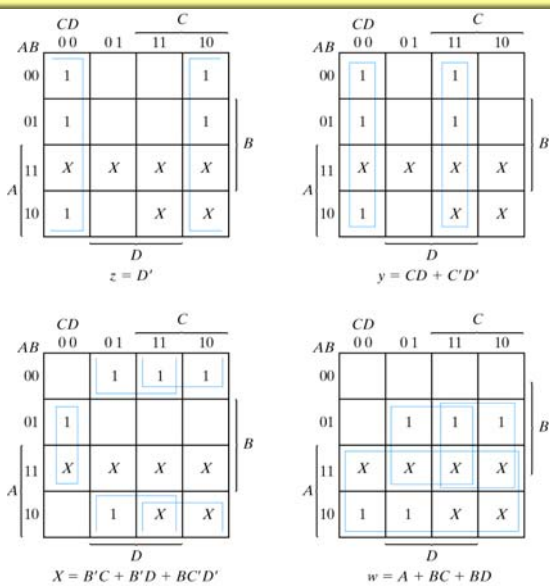
Παράδειγμα σχεδίασης

- Μετατροπή από έναν κώδικα σε έναν άλλο
  - Να σχεδιαστεί κύκλωμα που να μετατρέπει ένα ψηφίο κώδικα BCD (4 bit) σε κώδικα Excess-3 (υπέρβαση κατά 3)
- Από τον ορισμό των δύο κωδίκων παράγουμε τον πίνακα αληθείας
- «Λείπουν» 6 γραμμές – είναι οι γνωστοί αδιάφοροι όροι του BCD

Είσοδος (Κώδικας BCD) A B C D	Έξοδος (Κώδικας Excess-3) w x y z
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

Χάρτες Karnaugh για τις 4 συναρτήσεις

- Υπάρχουν έξι αδιάφοροι όροι
  - Λόγω των αχρησιμοποίητων συνδυασμών του κώδικα BCD
  - Στα ίδια σημεία των τεσσάρων χαρτών



Οι λογικές συναρτήσεις

Μετά την απλοποίηση με το χάρτη Karnaugh προκύπτουν τέσσερις συναρτήσεις για την έξοδο: w,x,y,z

$z=D'$

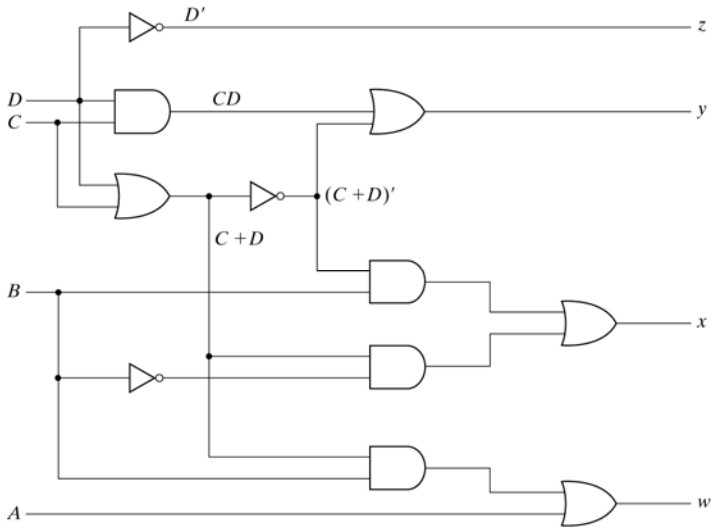
$y=CD+C'D'=CD+(C+D)'$

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

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

Και το λογικό διάγραμμα/κύκλωμα

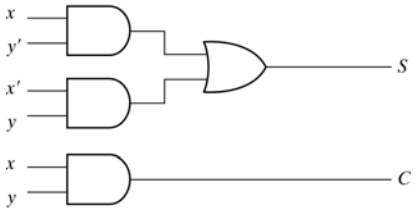
Τέσσερις είσοδοι και τέσσερις έξοδοι



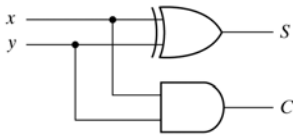
Αθροιστές – Ημιαθροιστής (Half-Adder)

- Ημιαθροιστής: προσθέτει 2 bit
- Πίνακας αληθείας
- Δύο υλοποιήσεις
  - 2 επίπεδα AND-OR
  - Με χρήση XOR

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



(a) $S = xy' + x'y$
$C = xy$



(b) $S = x \oplus y$
$C = xy$

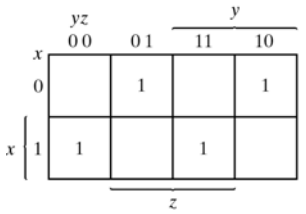
Πλήρης Αθροιστής – Full Adder

- Πλήρης Αθροιστής: προσθέτει 3 bit
  - Το τρίτο είναι το κρατούμενο
- Πίνακας αληθείας

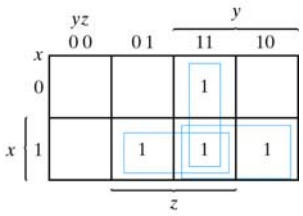
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Χάρτες Karnaugh για Πλήρη Αθροιστή

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



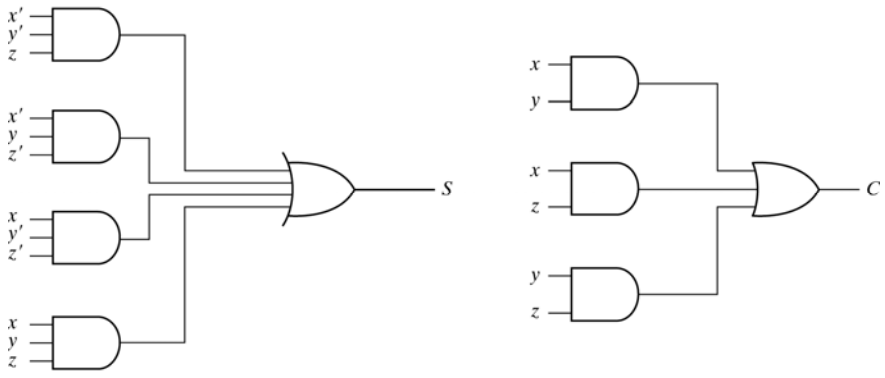
$S = x'y'z + x'yz' + xy'z' + xyz$



$S = xy + xz + yz$
$= xy + xy'z + x'yz$

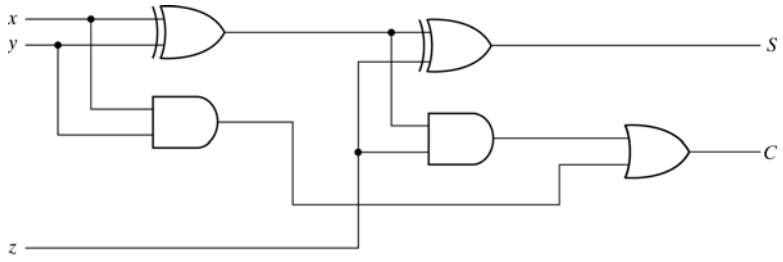
Υλοποίηση Πλήρους Αθροιστή

- Σε μορφή αθροίσματος γινομένων (AND-OR)



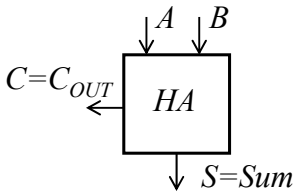
Εναλλακτική υλοποίηση

- Ένας πλήρης αθροιστής υλοποιείται και με
  - 2 ημιαθροιστές και μια πύλη OR

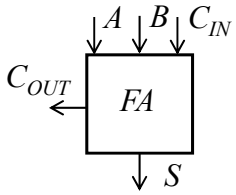


Σύμβολα Ημι- και Πλήρους Αθροιστή

- Ημιαθροιστής: 2 είσοδοι, 2 έξοδοι



- Πλήρης αθροιστής: 3 είσοδοι 2 έξοδοι



Δυαδική πρόσθεση των k bit

Δείκτης i:	3	2	1	0	
Κρατούμενο εισόδου	0	1	1	0	C_i
1ος προσθετέος	1	0	1	1	A_i
2ος προσθετέος	0	0	1	1	B_i
Άθροισμα	1	1	1	0	S_i
Κρατούμενο εξόδου	0	0	1	1	C_{i+1}

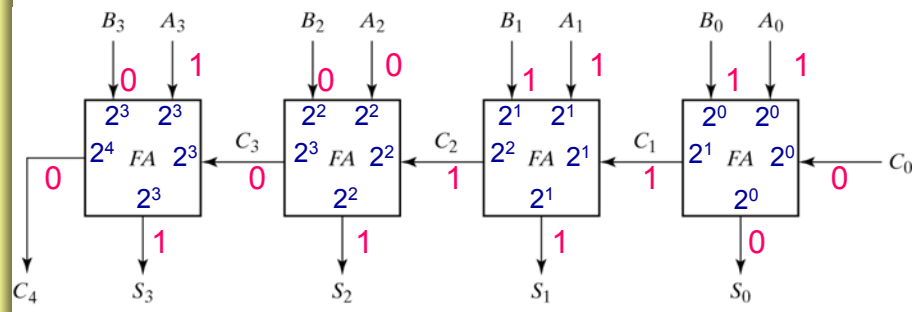
Πώς θα σχεδιάζατε έναν δυαδικό αθροιστή;

- Αν k=4 (όπως στο προηγούμενο παράδειγμα), τότε ένας δυαδικός αθροιστής είναι ένα κύκλωμα με 9 εισόδους:
  - 4 εισόδους για τον πρώτο προσθετέο, 4 εισόδους για το δεύτερο και 1 κρατούμενο εισόδου
- και 5 εξόδους:
  - 4 εξόδους για το άθροισμα και το κρατούμενο εξόδου

- Για να το υλοποιήσουμε πρέπει να απλοποιήσουμε (με χάρτη Karnaugh;; ή αλλιώς) 5 συναρτήσεις των 9 εισόδων
- Κάθε συνάρτηση έχει 2^9=512 ελαχιστόρους (!!!)
  - Τι κάνουμε;

Επαναληπτική χρήση κυκλώματος

- Ιεραρχική σχεδίαση με χρήση ενός βασικού κυκλώματος
  - Πλήρης αθροιστής ή
  - Ημιαθροιστής
- Δυαδικός αθροιστής (ριπής κρατουμένου – ripple carry) των 4 bit

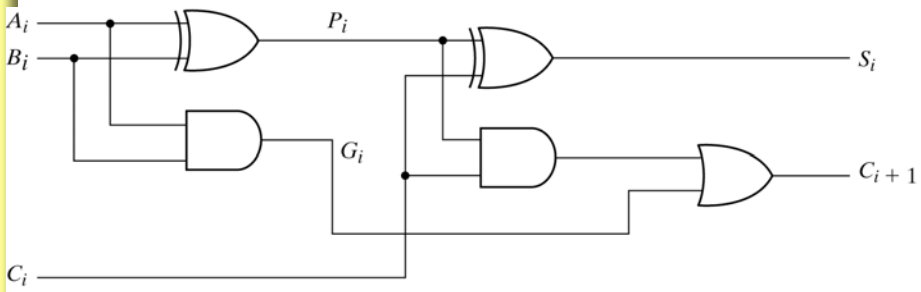


Καθυστέρηση διάδοσης κρατουμένου

- Απλή σχεδίαση αλλά...
- Στον αθροιστή ριπής που είδαμε,
  για να υπολογιστεί κάθε bit του αθροίσματος πρέπει
  να είναι γνωστό το κρατούμενο της προηγούμενης βαθμίδας
  - Μεγάλη καθυστέρηση διάδοσης (propagation delay)
- Θα μπορούσαμε να έχουμε διαθέσιμα όλα τα bit κρατουμένου ταυτόχρονα;
  - Αυτό προσπαθεί να πετύχει η επόμενη σχεδίαση
  - «Πρόβλεψη κρατουμένου»

Δύο νέα σήματα: Διάδοση και Γέννηση

- Σήματα διάδοσης (propagation) και γέννησης (generation) κρατουμένου
  - $P_i = A_i \oplus B_i$ $G_i = A_i \ B_i$
- Τότε:
  - $S_i = P_i \oplus C_i$ $C_{i+1} = G_i + P_i \ C_i$



Γιατί ονομάζονται έτσι;

- Το σήμα γέννησης (G_i - generation) είναι ίσο με 1 όταν μόνα τους τα bit A_i, B_i μιας βαθμίδας γεννούν κρατούμενο εξόδου ότι τιμή κι αν έχει το κρατούμενο εισόδου
  - Αυτό ισχύει μόνο όταν τα A_i, B_i είναι και τα δύο ίσα με 1
  - Δηλαδή ότι κι αν έρθει σαν κρατούμενο εισόδου, το κρατούμενο εξόδου θα είναι ίσο με 1
- Το σήμα διάδοσης (P_i - propagation) είναι ίσο με 1 όταν τα bit A_i, B_i μιας βαθμίδας διαδίδουν το κρατούμενο εισόδου σαν κρατούμενο εξόδου
  - Αυτό ισχύει όταν ένα από τα A_i, B_i είναι ίσο με 1
  - Οπότε ότι κρατούμενο έρθει στην είσοδο θα διαδοθεί το ίδιο στην έξοδο
- Όταν και τα δύο A_i, B_i είναι ίσα με 0, τότε δεν έχουμε ούτε γέννηση ούτε διάδοση
  - Και υποχρεωτικά το κρατούμενο εξόδου είναι ίσο με 0

Οι συναρτήσεις κρατουμένων με G και P

- Το κρατούμενο εισόδου σε έναν αθροιστή είναι το C_0

$C_1 = G_0 + P_0 C_0$

$C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$

$C_3 = G_2 + P_2 C_2 = \ldots = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$

$C_4 = \ldots = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$

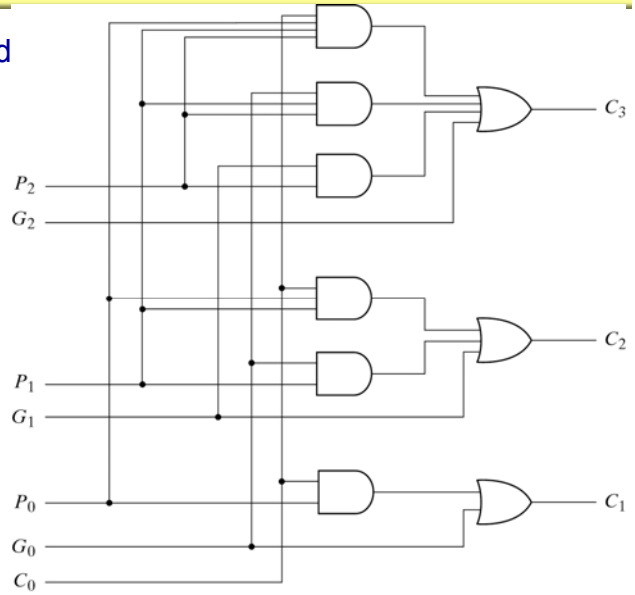
$C_5 = \ldots = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 G_0 + P_4 P_3 P_2 P_1 P_0 C_0$

$\ldots \ldots \ldots \ldots$

- Παρατηρήσεις:
  - Όλα τα G και P παράγονται μόνο από τα αντίστοιχα A και B
  - Όλα τα C επάνω παράγονται από δύο επίπεδα (AND + OR)
- Άρα: όλα τα κρατούμενα μπορούν να παραχθούν παράλληλα
  - Δεν υπάρχει σειριακή εξάρτηση

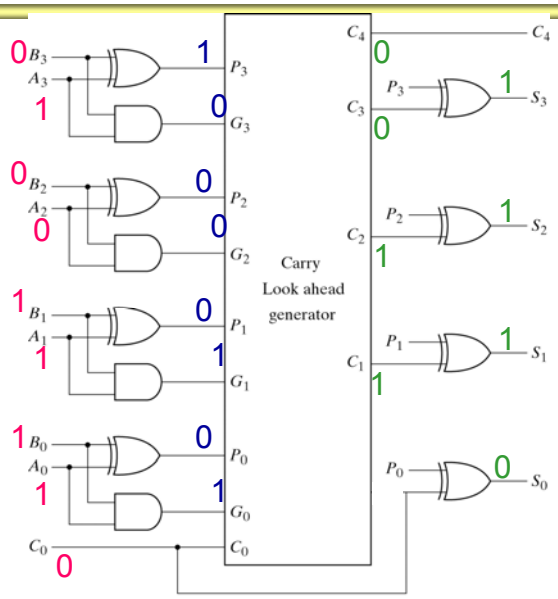
Γεννήτρια πρόβλεψης κρατουμένου

- Carry Lookahead Generator
- Σε δύο μόνο επίπεδα παράγονται **όλα** τα κρατούμενα ταυτόχρονα



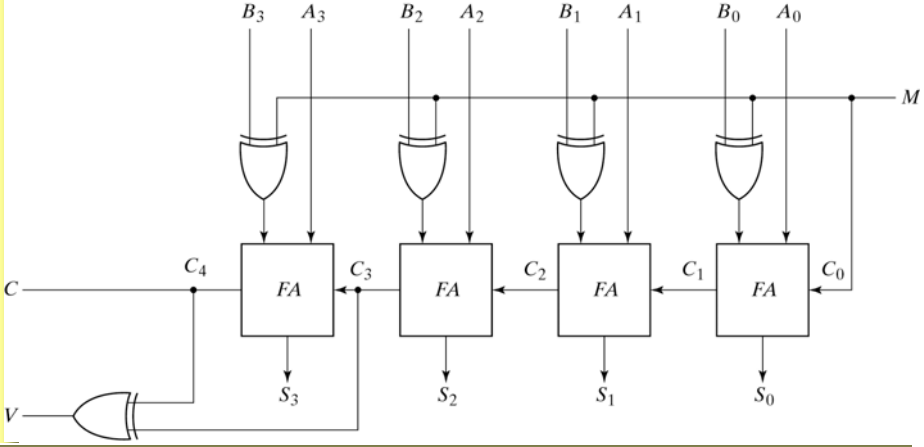
Αθροιστής πρόβλεψης κρατουμένου

- Των 4 bit
- Η γεννήτρια πρόβλεψης κρατουμένου είναι από το προηγούμενο σχήμα
- Προσέξτε την παράλληλη λειτουργία



Αθροιστής-Αφαιρέτης των 4 bit

- M=0, Πρόσθεση
- M=1, Αφαίρεση
  - Η αφαίρεση είναι σε συμπλήρωμα ως προς 2 (A+B'+1)



Δεκαδικός ή BCD αθροιστής

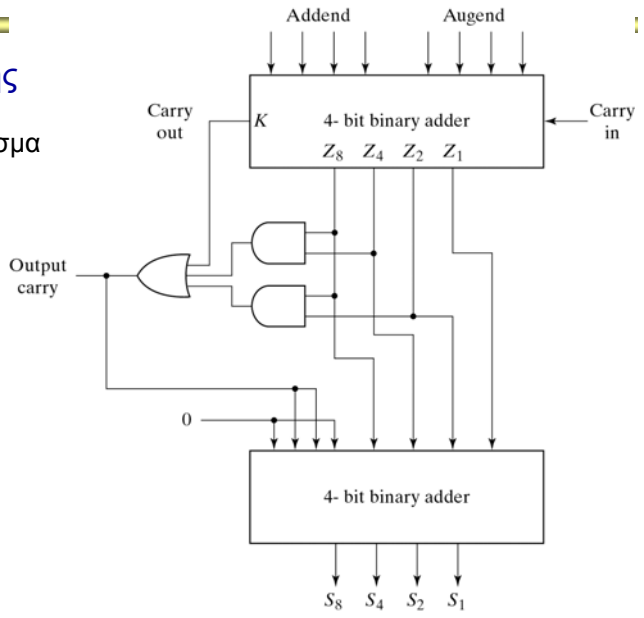
Αν προσθέσουμε 2 BCD ψηφία σε έναν απλό δυαδικό αθροιστή των 4 bit θα πάρουμε το αποτέλεσμα σε 4 bit και ένα κρατούμενο εξόδου

Δυαδικό Άθροισμα					Σωστό Άθροισμα BCD					Δεκαδικός
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

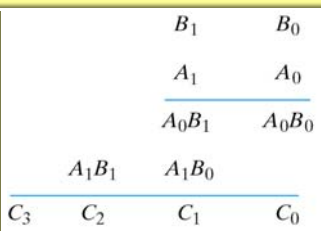
Αθροιστής BCD

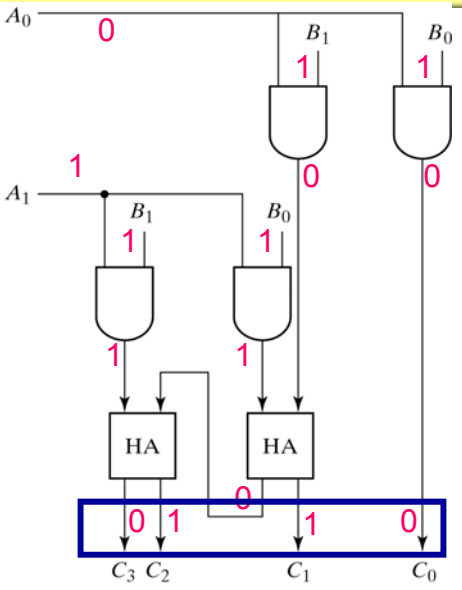
Βήμα Διόρθωσης

- +6 (0110) στο δυαδικό άθροισμα



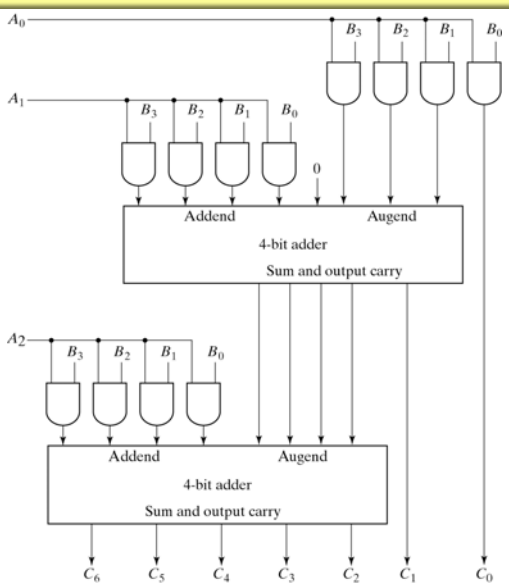
Δυαδικός Πολλαπλασιαστής 2x2





- Ας κάνουμε το 2x3
- A_1A_0=10
- B_1B_0=11

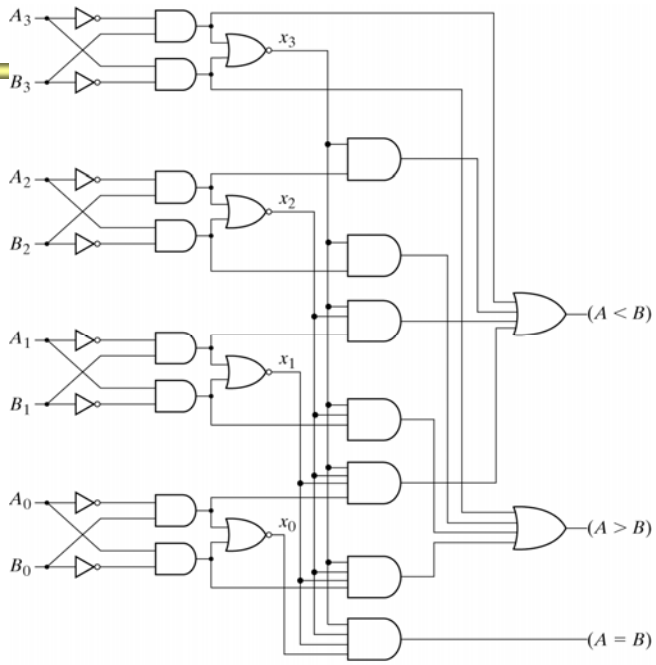
Δυαδικός Πολλαπλασιαστής 4 x 3



Συγκριτής μεγέθους (Comparator)

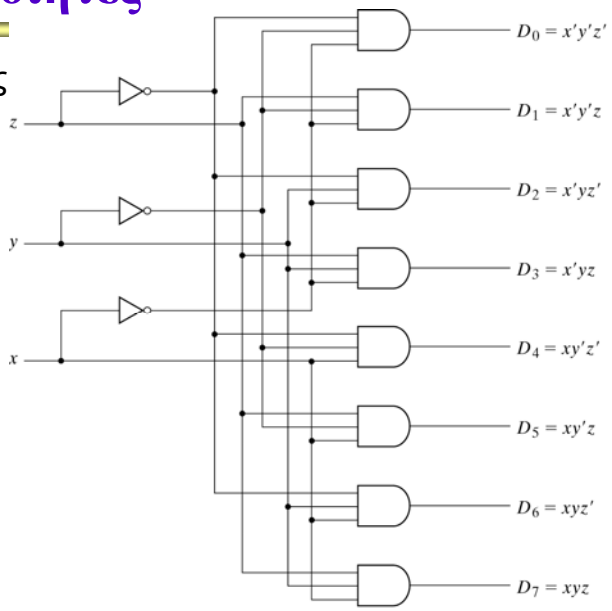
- Μη προσημασμένοι αριθμοί
  $A = A_3A_2A_1A_0$
  $B = B_3B_2B_1B_0$
- Ισότητα
  - Αποκλειστικό-ΟΥΤΕ (Exclusive NOR) σε κάθε ζεύγος bit
  - $x_i = A_iB_i + A_i'B_i'$
  - Άρα $(A=B) = x_3x_2x_1x_0$
- Ανισότητα
  - $(A>B) = A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0B_0'$
  - $(A<B) = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1 + x_3x_2x_1A_0'B_0$

Συγκριτής



Αποκωδικοποιητές

- Αποκωδικοποιητής 3-σε-8
- Γενικά
  - k-σε-2^k



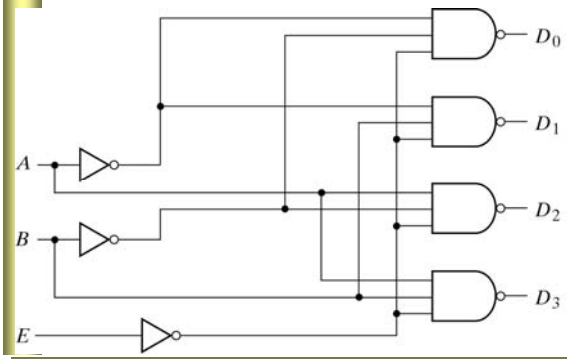
Πίνακας αληθείας του 3-σε-8

- «Θετική» λογική:
  η έξοδος που επιλέγεται έχει την τιμή 1 και όλες οι άλλες 0
- «Αρνητική» λογική:
  η έξοδος που επιλέγεται έχει την τιμή 0 και όλες οι άλλες 1

Είσοδοι			Έξοδοι							
x	*y*	*z*	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Αποκωδικοποιητής 2-σε-4 με επίτρεψη

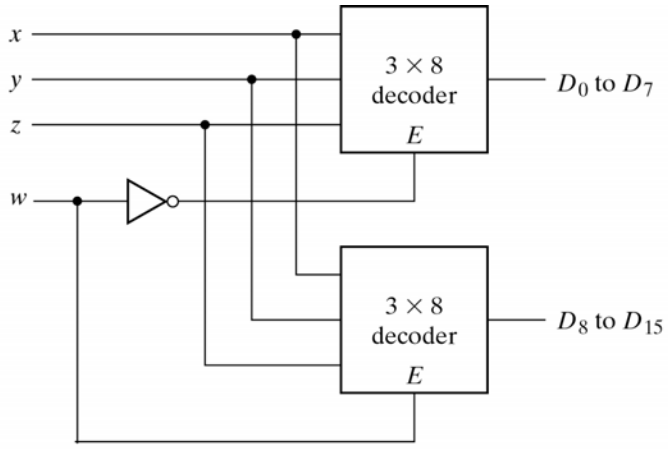
- Όταν η είσοδος επίτρεψης E είναι ίση με 0, τότε λειτουργεί σαν αποκωδικοποιητής
  - Εδώ με «αρνητική» λογική
- Όταν η είσοδος επίτρεψης είναι ίση με 1, όλες οι έξοδοι είναι στο 1 ανεξάρτητα των A και B



E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

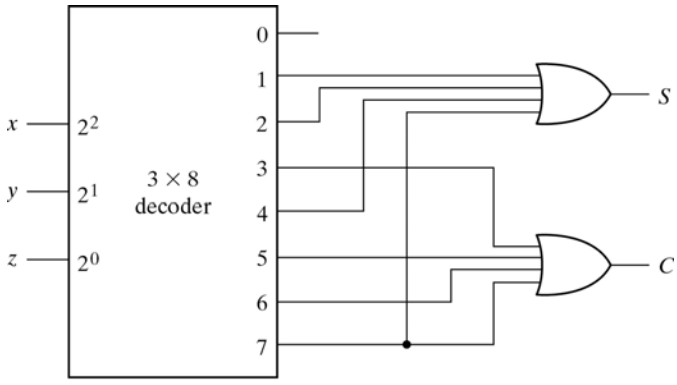
Ιεραρχική σχεδίαση αποκωδικοποιητών

- Σχεδίαση αποκωδικοποιητή 4 σε 16 από δύο αποκωδικοποιητές 3 σε 8



Υλοποίηση συνδυαστικής λογικής

- Ένας πλήρης αθροιστής υλοποιημένος με αποκωδικοποιητή 3-σε-8
  - $S(x,y,z) = \Sigma(1,2,4,7)$
  - $C(x,y,z) = \Sigma(3,5,6,7)$



Κωδικοποιητές

- Πίνακας αληθείας κωδικοποιητή οκταδικού σε δυαδικό
- 8-σε-3 (γενικά 2^k-σε-k)

Είσοδοι								Έξοδοι		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Στις άλλες περιπτώσεις τι γίνεται;

- Τί έξοδο θα δίνει το κύκλωμα του κωδικοποιητή όταν υπάρχουν στις εισόδους περισσότεροι από έναν άσσοι;
- Χρήση «Κωδικοποιητή προτεραιότητας»
  - Εδώ έχουν «προτεραιότητα» οι είσοδοι με μεγαλύτερο δείκτη

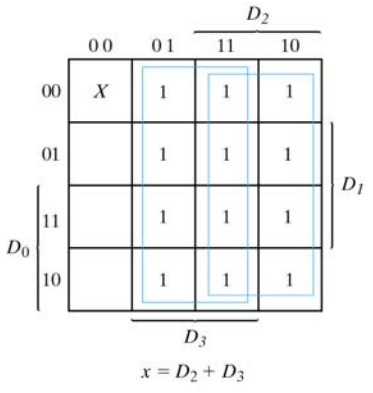
Είσοδοι				Έξοδοι		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

Χάρτες Κωδικοποιητή προτεραιότητας

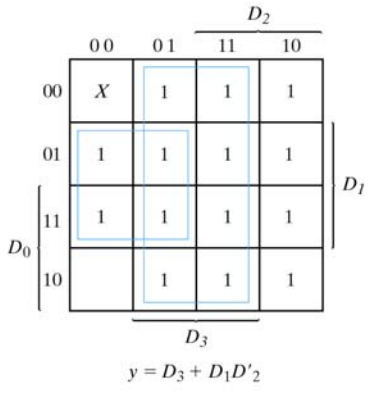
$$x = D_2 + D_3$$

$$y = D_3 + D_1 D_2'$$

$$V = D_0 + D_1 + D_2 + D_3$$



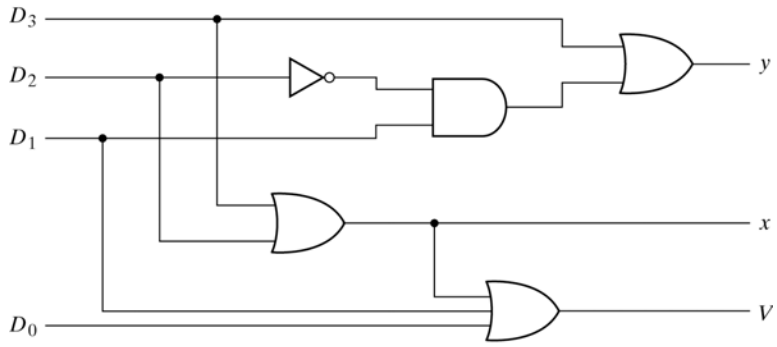
$x = D_2 + D_3$



$y = D_3 + D_1 D'_2$

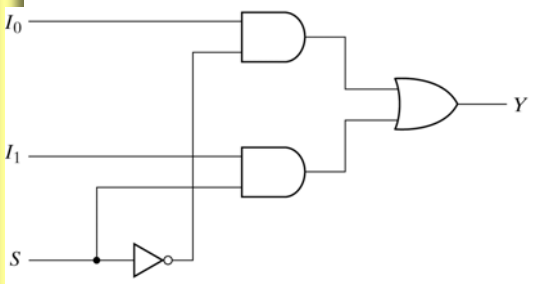
Κωδικοποιητής προτεραιότητας

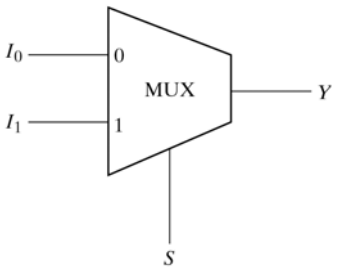
- Των 4 εισόδων



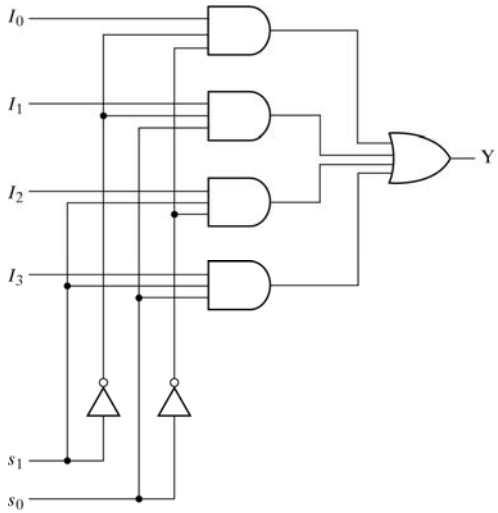
Πολυπλέκτες

- Πολυπλέκτης 2-σε-1
- Γενικά 2^n-σε-1
  - 2^n είσοδοι δεδομένων
  - 1 έξοδος δεδομένων
  - n είσοδοι επιλογής





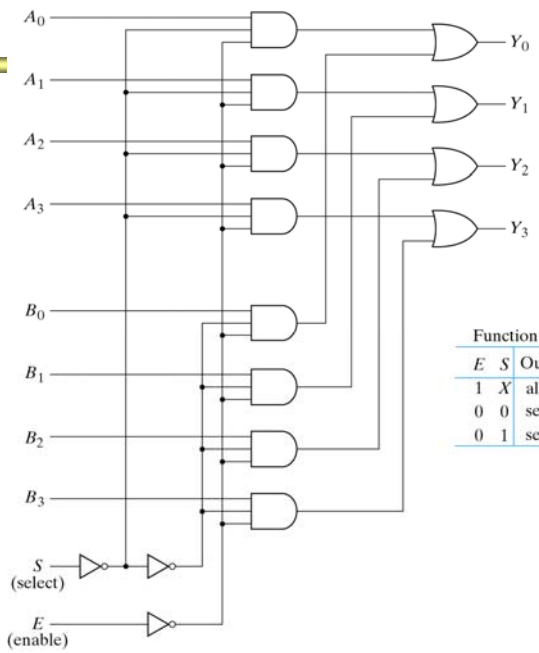
Πολυπλέκτης 4-σε-1



Είσοδοι		Έξοδοι
s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Πολλά bit

- Τετραπλός (4 bit) πολυπλέκτης 2-σε-1
  - 2 λέξεις των 4 bit



Function table

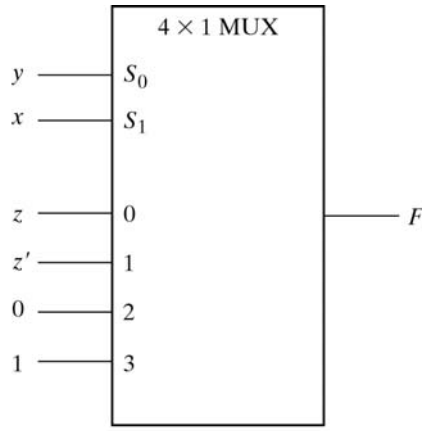
E	S	Output Y
1	X	all 0's
0	0	select A
0	1	select B

Συναρτήσεις με Πολυπλέκτες

- Συνάρτηση 3 μεταβλητών

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

x	y	z	F	
0	0	0	0	$F = z$
0	0	1	1	
0	1	0	1	$F = z'$
0	1	1	0	
1	0	0	0	$F = 0$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	

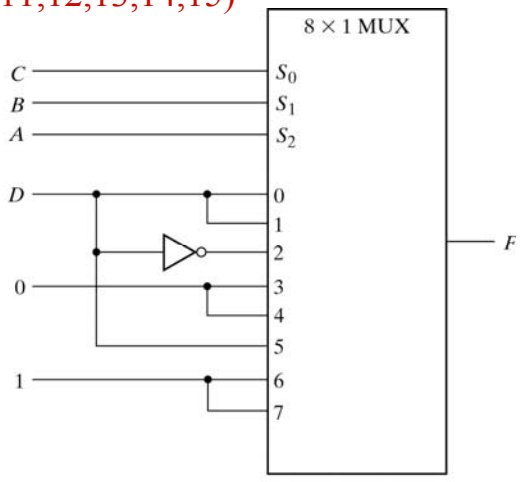


Συναρτήσεις με Πολυπλέκτες (συνέχ.)

- Συνάρτηση 4 μεταβλητών

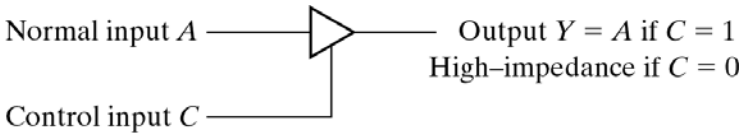
$F(A,B,C,D) = \Sigma(1,3,4,11,12,13,14,15)$

A	B	C	D	F	
0	0	0	0	0	$F = D$
0	0	0	1	1	
0	0	1	0	0	$F = D$
0	0	1	1	1	
0	1	0	0	1	$F = D'$
0	1	0	1	0	
0	1	1	0	0	$F = 0$
0	1	1	1	0	
1	0	0	0	0	$F = 0$
1	0	0	1	0	
1	0	1	0	0	$F = D$
1	0	1	1	1	
1	1	0	0	1	$F = 1$
1	1	0	1	1	
1	1	1	0	1	$F = 1$
1	1	1	1	1	

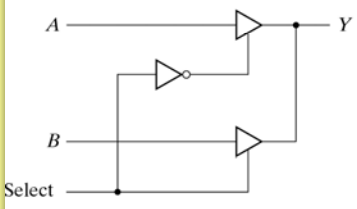


Πύλες Τριών Καταστάσεων

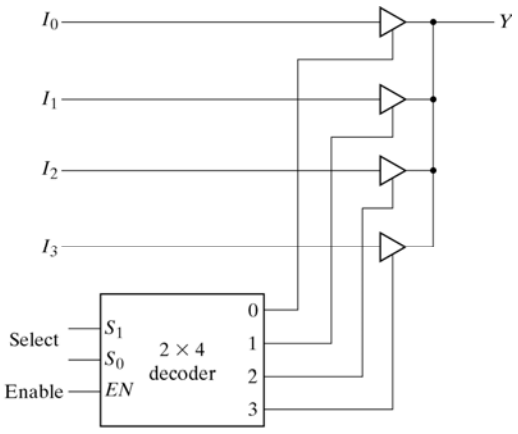
- Απομονωτής Τριών Καταστάσεων (Three-state buffer)



Πολυπλέκτες με πύλες 3 καταστάσεων



(a) 2-to-1- line mux



(b) 4 - to - 1 line mux