Πανεπιστήμιο Πατρών
Τμήμα Φυσικής
Εργαστήριο Ηλεκτρονικής

Ψηφιακά Ηλεκτρονικά

Συνδυαστική Λογική

Επιμέλεια Διαφανειών: Δ. Μπακάλης

Πάτρα, Φεβρουάριος 2009

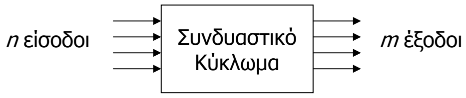
Ψηφιακά Κυκλώματα

- Τα ψηφιακά κυκλώματα διακρίνονται σε συνδυαστικά (combinational) και ακολουθιακά (sequential).

- Συνδυαστικά κυκλώματα: Οι έξοδοι είναι συνάρτηση των τρέχουσων εισόδων και μόνο.

- Ακολουθιακά κυκλώματα: Οι έξοδοι είναι συνάρτηση των τρεχουσών εισόδων και της κατάστασης των στοιχείων μνήμης η οποία είναι συνάρτηση των προηγούμενων εισόδων.

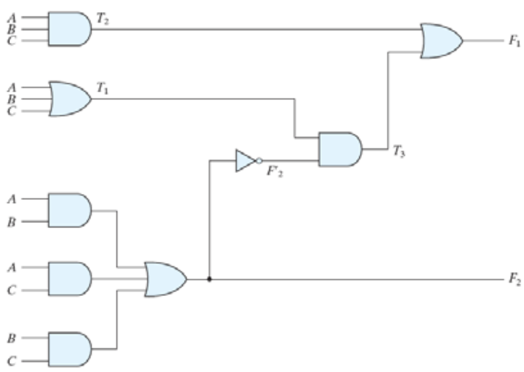
Συνδυαστικά Κυκλώματα



- Ένα συνδυαστικό κύκλωμα αποτελείται από μεταβλητές εισόδου, λογικές πύλες και μεταβλητές εξόδου.
- Για κάθε έναν από τους 2^n πιθανούς συνδυασμούς δυαδικών τιμών στις εισόδους υπάρχει ένας και μόνο ένας συνδυασμός των εξόδων.
- Ένα συνδυαστικό κύκλωμα περιγράφεται με *m* συναρτήσεις Boole *n* μεταβλητών, μία για κάθε μεταβλητή εξόδου.
- Κάθε μεταβλητή εισόδου μπορεί να είναι διαθέσιμη με την κανονική ή/και με τη συμπληρωματική μορφή της.

Ανάλυση Συνδυαστικού Κυκλώματος

Από το κύκλωμα βρίσκουμε τις συναρτήσεις Boole:
1. Ονομάζουμε τις εξόδους των πυλών του κυκλώματος.
2. Βρίσκουμε τις συναρτήσεις σε κάθε επίπεδο μέχρι το τελευταίο.



Κύκλωμα πλήρη αθροιστή

$T_1 = A+B+C$

$T_2 = ABC$

$F_2 = AB+AC+BC = \Sigma(3,5,6,7)$

$T_3 = F_2'T_1$

$F_1 = T_3 + T_2 = \dots =$
$= A'BC'+A'B'C+AB'C'+ABC$
$= \Sigma(1,2,4,7)$

Ανάλυση Συνδυαστικού Κυκλώματος

Μπορούμε να κατασκευάσουμε τον πίνακα αλήθειας του κυκλώματος κατευθείαν από το λογικό διάγραμμα χωρίς να εξάγουμε συναρτήσεις Boole.

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

$T_1 = A+B+C$

$T_2 = ABC$

$F_2 = AB+AC+BC$

$= \Sigma(3,5,6,7)$

$T_3 = F_2'T_1$

$F_1 = T_3 + T_2$

$= \Sigma(1,2,4,7)$

Σχεδιασμός Συνδυαστικών Κυκλωμάτων

Διαδικασία Σχεδιασμού

1. Καθορισμός πλήθους εισόδων/εξόδων και αντιστοίχηση ονομάτων.
2. Εξαγωγή του πίνακα αλήθειας.
3. Απλοποίηση συναρτήσεων των εξόδων του κυκλώματος.
4. Κατασκευάζουμε το λογικό διάγραμμα του κυκλώματος.
5. Επαληθεύουμε την ορθότητα της σχεδίασης.

Σχεδιασμός Συνδυαστικών Κυκλωμάτων

Επιλογή Απλοποιημένης Έκφρασης

1. Ελάχιστος αριθμός πυλών.
2. Ελάχιστος αριθμός εισόδων πύλης.
3. Ελάχιστο χρόνο διάδοσης σήματος.
4. Ελάχιστος αριθμός διασυνδέσεων.
5. Περιορισμοί οδήγησης.

Παράδειγμα: Κύκλωμα Μετατροπής Κωδίκων

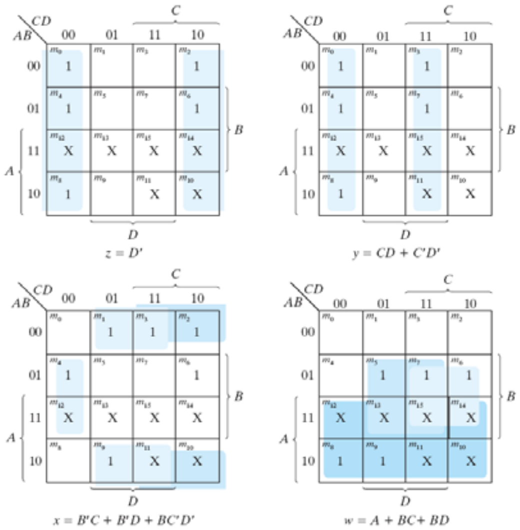
- Η ύπαρξη πολλών κωδίκων οδηγεί στην ανάγκη μετατροπών ανάλογα με τη λειτουργία του κάθε συστήματος.
- Ένας μετατροπέας κωδίκων είναι ένα κύκλωμα που μετατρέπει πληροφορία από ένα κώδικα Α σε ένα κώδικα Β.
- Οι γραμμές εισόδου τροφοδοτούν το κύκλωμα με συνδυασμούς δυαδικών ψηφίων που ανήκουν στον κώδικα Α.
- Οι γραμμές εξόδου παράγουν συνδυασμούς δυαδικούς ψηφίων που ανήκουν στον κώδικα Β.
- Οι αχρησιμοποίητες καταστάσεις αποτελούν αδιάφορους όρους.

Παράδειγμα: Μετατροπή BCD σε Excess-3

Είσοδος Κώδικας BCD				Έξοδος Κώδικας excess-3			
A	*B*	*C*	*D*	*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

Παράδειγμα: Μετατροπή BCD σε Excess-3

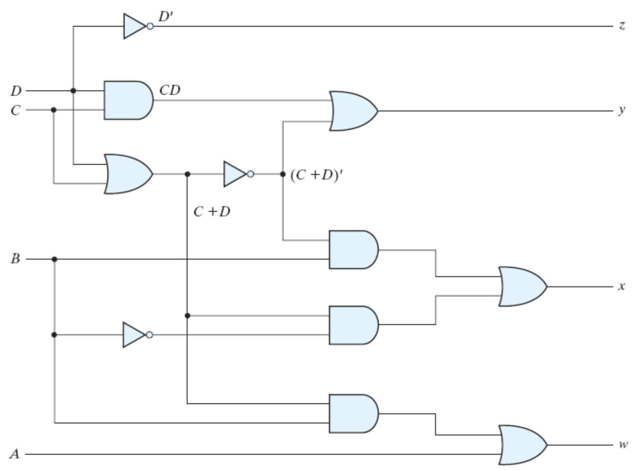
Χάρτες μετατροπέα κώδικα BCD σε Excess-3.



Παράδειγμα: Μετατροπή BCD σε Excess-3

Μπορούμε να αλλάξουμε αλγεβρικά τις απλοποιημένες εκφράσεις με σκοπό να χρησιμοποιήσουμε κοινές πύλες για 2 ή περισσότερες εξόδους.

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)



Ημι-Αθροιστής (Half Adder)

1.Καθορισμός προβλήματος: κύκλωμα που να προσθέτει δύο δυαδικά ψηφία.

2.Πλήθος εισόδων/εξόδων: 2 είσοδοι – 2 έξοδοι.

3.Ονομασία εισόδων/εξόδων: έστω x, y οι δύο είσοδοι (προσθετέοι) και C (κρατούμενο), S (άθροισμα) οι δύο έξοδοι.

4.Πίνακας αλήθειας:

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

Ημι-Αθροιστής

(α)

S = x'y + xy'

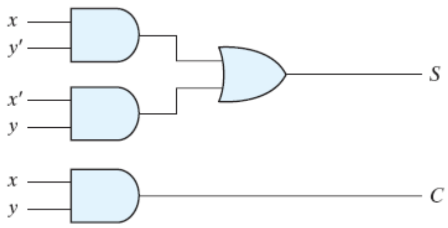
C = xy

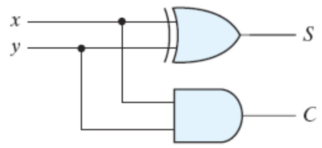
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(β)

S = x ⊕ y

C = xy





Πλήρης Αθροιστής (Full Adder)

1.<u>Καθορισμός προβλήματος</u>: κύκλωμα που να προσθέτει τρία δυαδικά ψηφία.

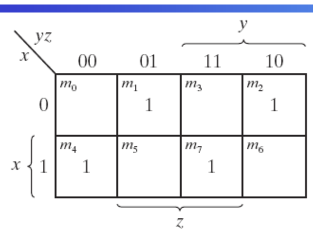
2.<u>Πλήθος εισόδων/εξόδων</u>: 3 είσοδοι – 2 έξοδοι.

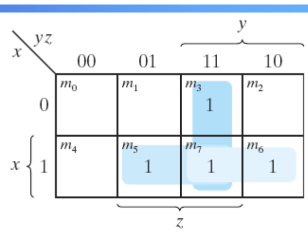
3.<u>Ονομασία εισόδων/εξόδων</u>: έστω x, y οι δύο είσοδοι (προσθετέοι), z το κρατούμενο της προηγούμενης βαθμίδας και C (κρατούμενο), S (άθροισμα) οι δύο έξοδοι.

4.<u>Πίνακας αλήθειας</u>:

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

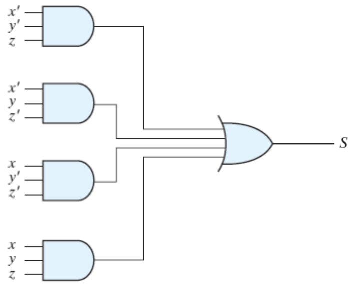
Πλήρης-Αθροιστής

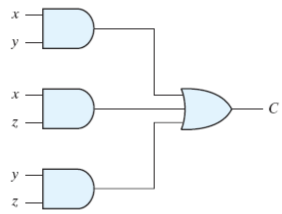




S = x'y'z+x'yz'+xy'z'+xyz C = xy + yz + xz

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

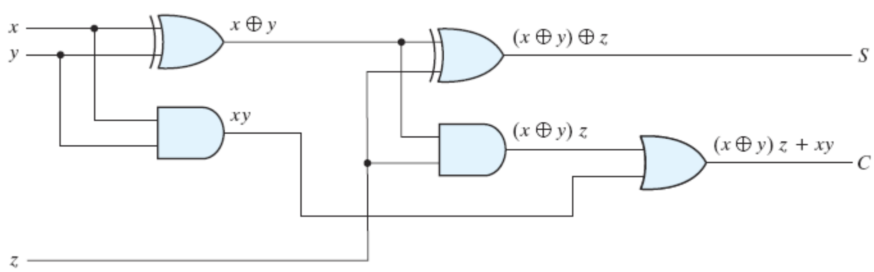




Πλήρης-Αθροιστής

Ένας πλήρης αθροιστής μπορεί να υλοποιηθεί με δύο ημιαθροιστές και μία πύλη OR.

$S = x \oplus y \oplus z$ $C = xy + yz + xz$

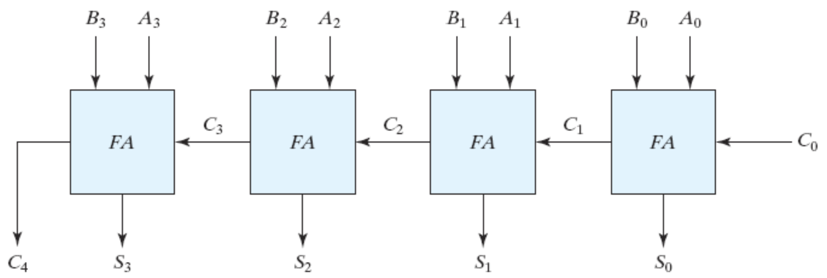


Δυαδικός Αθροιστής

Δείκτης i	4	3	2	1	
Κρατούμενο εισόδου	0	1	1	0	C_i
Προσθετέος	1	0	1	1	A_i
Προσθετέος	0	0	1	1	B_i
Άθροισμα	1	1	1	0	S_i
Κρατούμενο εξόδου	0	0	1	1	C_{i+1}

Το άθροισμα δύο δυαδικών αριθμών των *n* bits μπορεί να παραχθεί είτε σειριακά είτε παράλληλα.

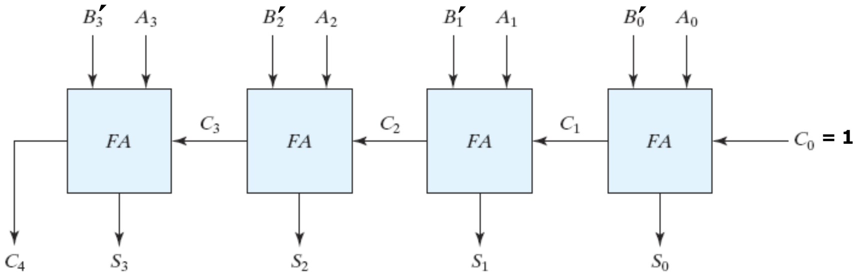
Παράλληλος Δυαδικός Αθροιστής



Ο Παράλληλος Αθροιστής αποτελείται από *n* πλήρεις αθροιστές και παράγει το αποτέλεσμα σε 1 κύκλο ρολογιού.

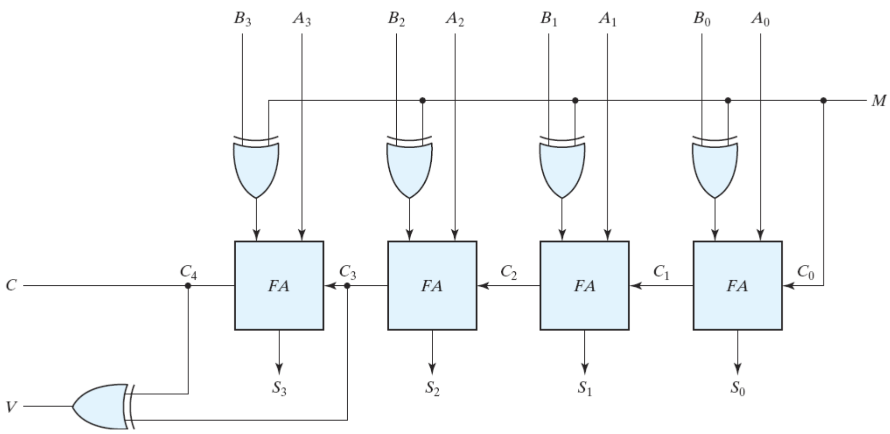
Υλοποίηση με συναρτήσεις:
Πίνακας αλήθειας με 9 εισόδους και 2^9=512 καταστάσεις.

Παράλληλος Δυαδικός Αφαιρέτης



A – B = A + συμπλήρωμα(B) = A + B' + 1

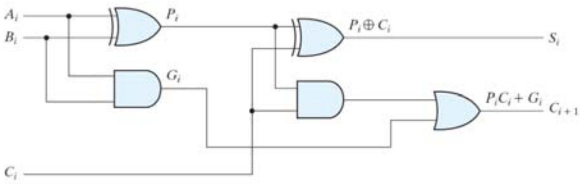
Παράλληλος Δυαδικός Αθροιστής/Αφαιρέτης



Διάδοση Κρατουμένου Δυαδικού Αθροιστή

Χρόνος Διάδοσης (Καθυστέρηση): Επίπεδα Πυλών × Καθυστέρηση Πύλης

Παράλληλος Αθροιστής: Η μεγαλύτερη καθυστέρηση οφείλεται στη διάδοση του κρατουμένου.

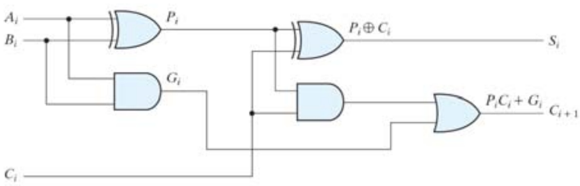


- 2 επίπεδα πυλών για κάθε διάδοση κρατουμένου.
- 2 × *n* επίπεδα για τον παράλληλο αθροιστή *n* bits.

Ο χρόνος διάδοσης κρατουμένου είναι πολύ μεγάλος. ⟺ Η πρόσθεση είναι η συχνότερη πράξη.

Μείωση χρόνου διάδοσης κρατουμένου.
- Χρήση γρηγορότερων πυλών (άνω όριο).
- Αύξηση πολυπλοκότητας κυκλώματος (πρόβλεψη κρατουμένου).

Πρόβλεψη Κρατουμένου (Carry Look-Ahead)



$P_i = A_i \oplus B_i$	$S_i = P_i \oplus C_i$
$G_i = A_iB_i$	$C_{i+1} = G_i + P_iC_i$

Διαδοτής Κρατουμένου

Γεννητής Κρατουμένου

C_0 = κρατούμενο εισόδου

$C_1 = G_0 + P_0C_0$

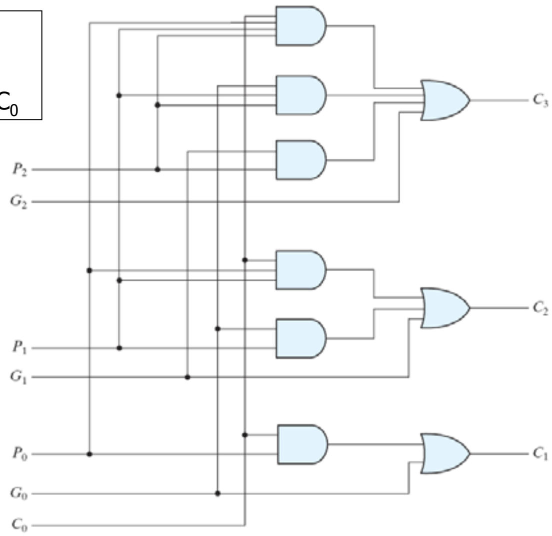
$C_2 = G_1 + P_1C_1 = G_1 + P_1(G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$

$C_3 = G_2 + P_2C_2 = \ldots = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$

Κάθε κρατούμενο μπορεί να υλοποιηθεί με ένα επίπεδο πυλών AND ακολουθούμενο από μία πύλη OR.

Γεννήτρια Πρόβλεψης Κρατουμένου

$C_1 = G_0 + P_0C_0$
$C_2 = G_1 + P_1G_0 + P_1P_0C_0$
$C_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$

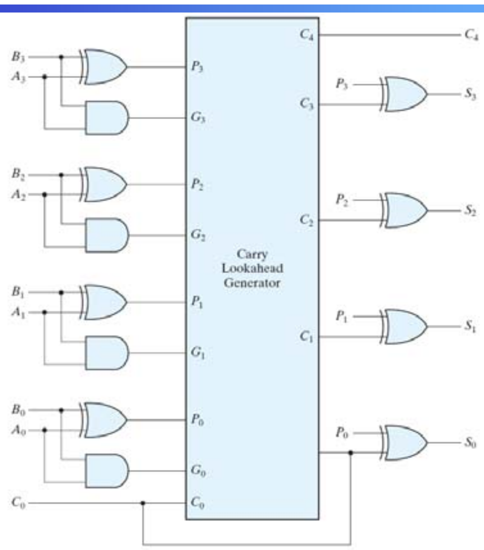


Αθροιστής Πρόβλεψης Κρατουμένου

$P_i = A_i \oplus B_i$

$G_i = A_iB_i$

$S_i = P_i \oplus C_i$



Κύκλωμα Σύγκρισης (Comparator)

Συγκριτής: συγκρίνει δύο αριθμούς και βρίσκει τη σχέση τους (<,>,=).

- Για δύο αριθμούς των n bits έχουμε 2^{2n} συνδυασμούς.
- Το κύκλωμα του συγκριτή έχει αρκετή κανονικότητα.

➢ Έστω $A = A_3A_2A_1A_0$ και $B = B_3B_2B_1B_0$ οι δύο αριθμοί.

➢ Ισχύει A = B όταν όλα τα ζευγάρια (A_i, B_i) είναι ίσα, δηλαδή $A_3=B_3$ και $A_2=B_2$ και $A_1=B_1$ και $A_0=B_0$.

$$\boxed{(A=B) = x_3x_2x_1x_0 \qquad x_i = A_iB_i + A_i'B_i'}$$

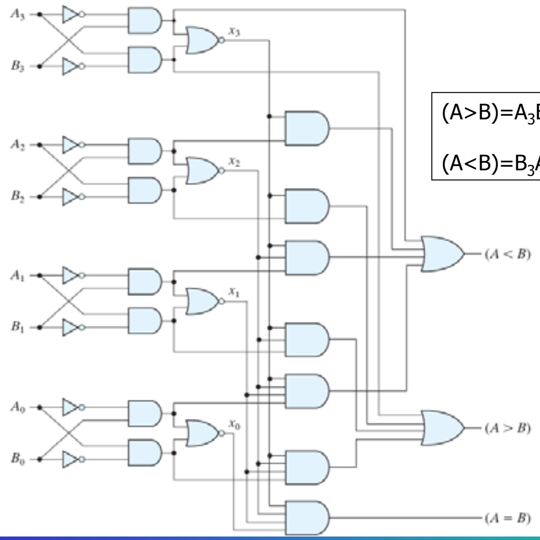
Κύκλωμα Σύγκρισης

➢ Για να βρούμε εάν A<B ή A>B εξετάζουμε τα σχετικά μεγέθη των ζευγαριών ψηφίων ξεκινώντας από την πιο σημαντική θέση. Εάν τα δύο ψηφία είναι ίσα τότε συγκρίνουμε το επόμενο λιγότερο σημαντικό ζευγάρι ψηφίων.

➢ Εάν A_i = 1 και B_i = 0 τότε A > B, εάν A_i = 0 και B_i = 1 τότε A < B.

$$\boxed{\begin{aligned}(A>B) &= A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0B_0' \\ (A<B) &= B_3A_3' + x_3B_2A_2' + x_3x_2B_1A_1' + x_3x_2x_1B_0A_0'\end{aligned}}$$

Κύκλωμα Σύγκρισης



$(A=B) = x_3x_2x_1x_0$
$x_i = A_iB_i + A_i'B_i'$

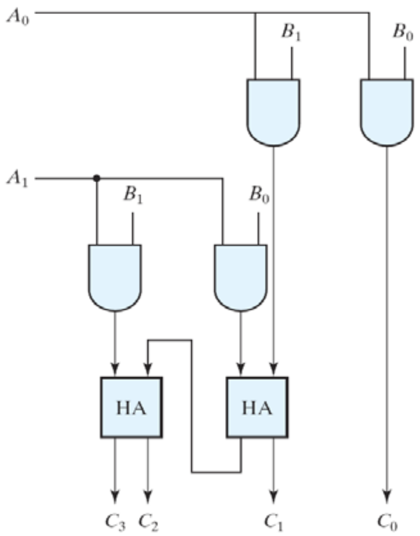
$(A>B)=A_3B_3'+x_3A_2B_2'+x_3x_2A_1B_1'+x_3x_2x_1A_0B_0'$

$(A<B)=B_3A_3'+x_3B_2A_2'+x_3x_2B_1A_1'+x_3x_2x_1B_0A_0'$

Δυαδικός πολλαπλασιαστής

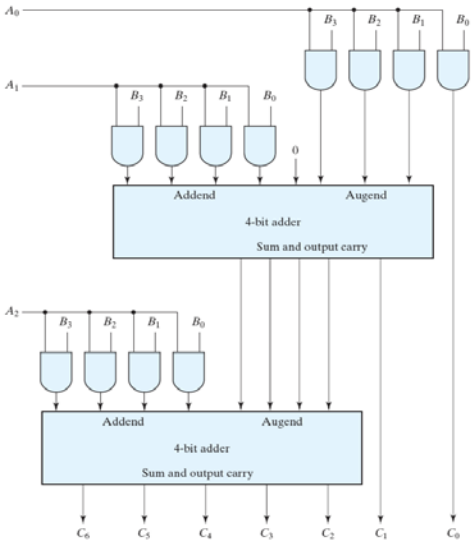
		B_1	B_0
		A_1	A_0
		A_0B_1	A_0B_0
	A_1B_1	A_1B_0	
C_3	C_2	C_1	C_0

Κύκλωμα πολ/στη 2x2 bit



Δυαδικός πολλαπλασιαστής

Κύκλωμα πολ/στη 4x3 bit

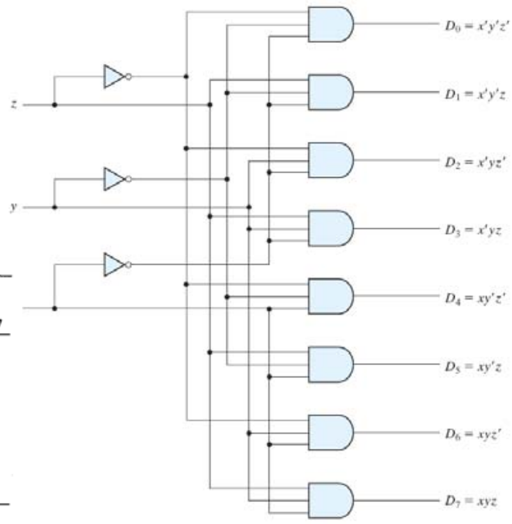


Αποκωδικοποιητής (Decoder)

<u>Αποκωδικοποιητής</u>: κύκλωμα που μετατρέπει τη δυαδική πληροφορία των *n* γραμμών εισόδου σε εως 2^n μοναδικές γραμμές εξόδου (ελαχιστόροι *n* μεταβλητών).

<u>Παράδειγμα</u>: Αποκωδικοποιητής 3-σε-8

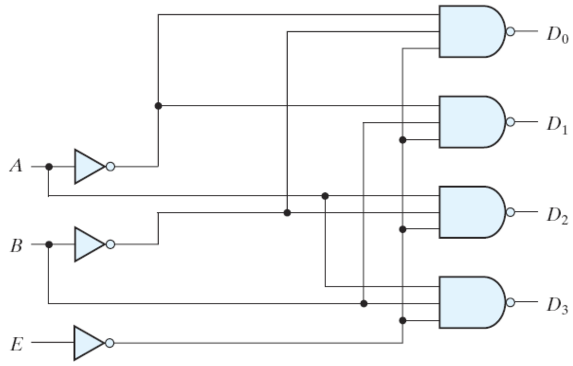
Είσοδοι			Έξοδοι							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Αποκωδικοποιητής με Είσοδο Επίτρεψης

Ο αποκωδικοποιητής μπορεί να παράγει συμπληρωματικές εξόδους.

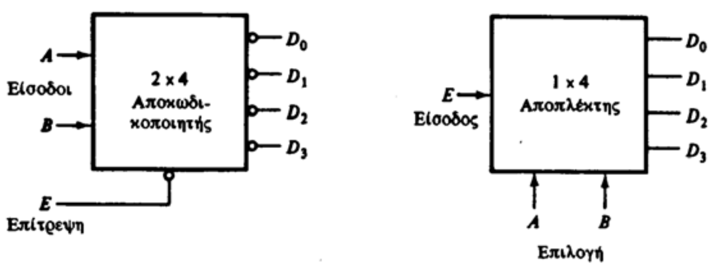
Ο αποκωδικοποιητής μπορεί να έχει είσοδο επίτρεψης.



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

Αποπλέκτης (Demultiplexer)

Ο αποπλέκτης δέχεται πληροφορίες από μία απλή γραμμή και τις μεταβιβάζει σε μία από τις 2^n δυνατές γραμμές εξόδου ανάλογα με τις τιμές των n γραμμών επιλογής.

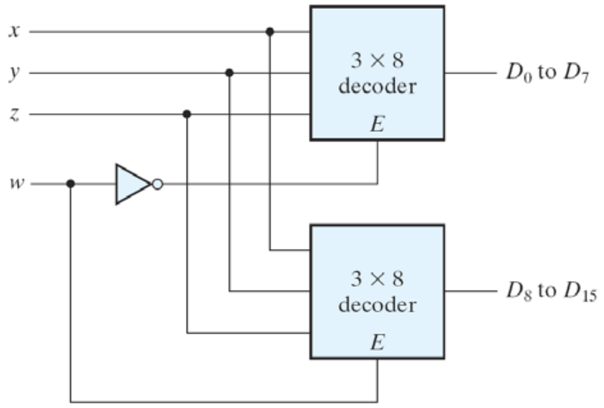


(α) Αποκωδικοποιητής με επίτρεψη

(β) Αποπλέκτης

Αποκωδικοποιητής/Αποπλέκτης

Επέκταση αποκωδικοποιητή με χρήση πολλών αποκωδικοποιητών ⇨ 2 αποκωδικοποιητές 3 σε 8 δίνουν 1 αποκωδικοποιητή 4 σε 16



Υλοποίηση Συνάρτησης με Αποκωδικοποιητή

- Αφού ο αποκωδικοποιητής παράγει τους 2^n ελαχιστόρους μπορεί να χρησιμοποιηθεί για να υλοποιήσει οποιαδήποτε συνάρτηση.

- Κάθε συνδυαστικό κύκλωμα με *n* εισόδους και *m* εξόδους μπορεί να υλοποιηθεί με έναν αποκωδικοποιητή *n*-σε-2^n γραμμών και *m* πύλες OR.

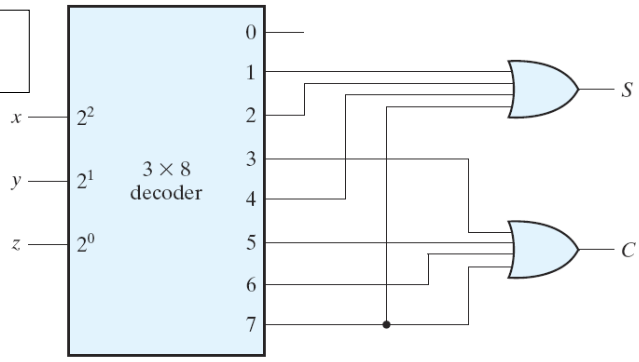
- Εάν ο αριθμός των ελαχιστόρων μιας συνάρτησης είναι μεγαλύτερος από $2^n/2$, τότε μπορούμε να χρησιμοποιήσουμε μία πύλη NOR για να αθροίσουμε τους ελαχιστόρους της F'. Η έξοδος της πύλης NOR δίνει τη συνάρτηση F.

Παράδειγμα

Υλοποίηση πλήρους αθροιστή με αποκωδικοποιητή.

S(x,y,z) = Σ(1,2,4,7)

C(x,y,z) = Σ(3,5,6,7)



Κωδικοποιητής (Encoder)

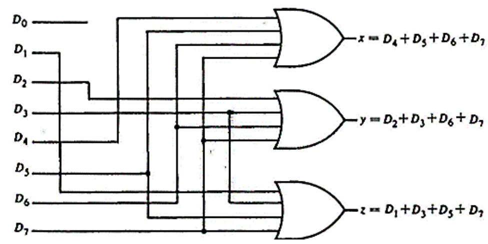
Ο Κωδικοποιητής εκτελεί την αντίστροφη λειτουργία από τον Αποκωδικοποιητή: Έχει 2^n γραμμές εισόδου και n γραμμές εξόδου και δίνει στην έξοδο τον δυαδικό κώδικα που αντιστοιχεί στις γραμμές εισόδου.

Προβλήματα:

- Όταν περισσότερες της μίας είσοδοι είναι 1 τότε η έξοδος είναι απροσδιόριστη. (Λύση: προτεραιότητα)
- Όταν όλες οι είσοδοι είναι 0 τότε η έξοδος είναι 0 που δεν είναι σωστό αφού η $D_0 \neq 1$. (Λύση: διάκριση της κατάστασης)

Κωδικοποιητής από Οκταδικό σε Δυαδικό

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

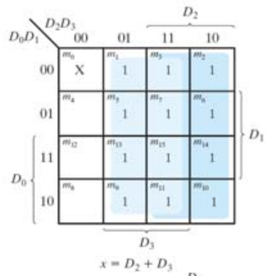


Κωδικοποιητής Προτεραιότητας

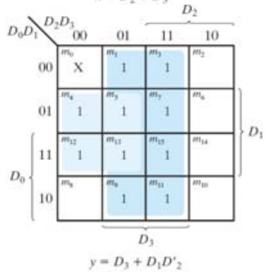
Ο κωδικοποιητής προτεραιότητας είναι ένα κύκλωμα κωδικοποιητή που περιλαμβάνει συνάρτηση προτεραιότητας.

<u>Παράδειγμα</u>: Κωδικοποιητής προτεραιότητας 4 εισόδων

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



$x = D_2 + D_3$



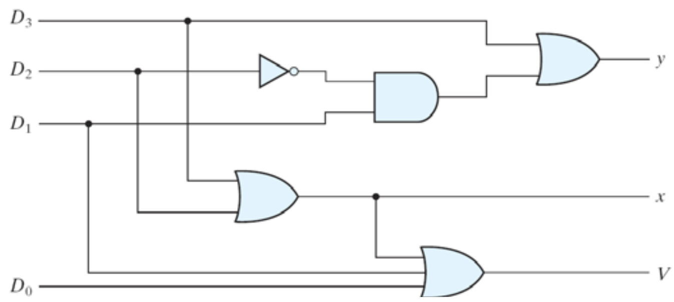
$y = D_3 + D_1D'_2$

Κωδικοποιητής Προτεραιότητας 4 Εισόδων

$x = D_2+D_3$

$y = D_3+D_1D_2'$

$V = D_0+D_1+D_2+D_3$



Λύνει το πρόβλημα της επιλογής όταν περισσότερες της μίας εισόδων είναι 1 επιλέγοντας αυτή με τη μεγαλύτερη προτεραιότητα.

Πολυπλέκτης (Multiplexer)

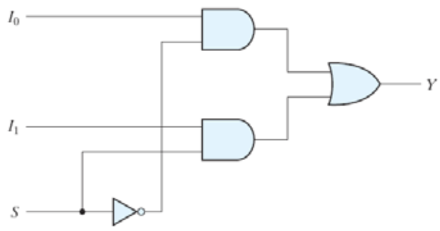
- Ο πολυπλέκτης είναι ένα συνδυαστικό κύκλωμα που επιλέγει δυαδικές πληροφορίες ανάμεσα σε πολλές γραμμές εισόδου και τις κατευθύνει σε μία γραμμή εξόδου.

- Η επιλογή της μιας συγκεκριμένης γραμμής εισόδου γίνεται μέσω μερικών γραμμών επιλογής.

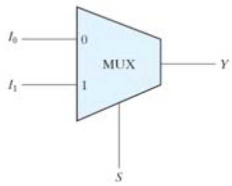
- Ένας πολυπλέκτης 2^n-σε-1 γραμμή κατασκευάζεται από έναν αποκωδικοποιητή n-σε-2^n προσθέτοντας σε αυτόν 2^n εισόδους μια για κάθε πύλη AND. Οι έξοδοι των πυλών AND εφαρμόζονται σε μια μοναδική πύλη OR για να δώσουν τη μία γραμμή εξόδου.

Πολυπλέκτης 2-σε-1

s	Y
0	I_0
1	I_1

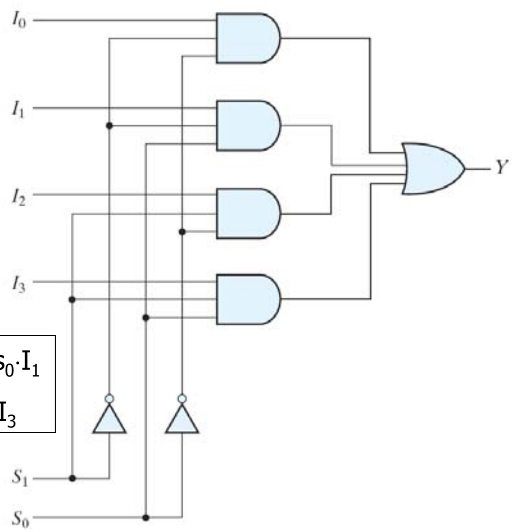


$$Y(s,I_0,I_1) = s'\cdot I_0 + s\cdot I_1$$



Πολυπλέκτης 4-σε-1

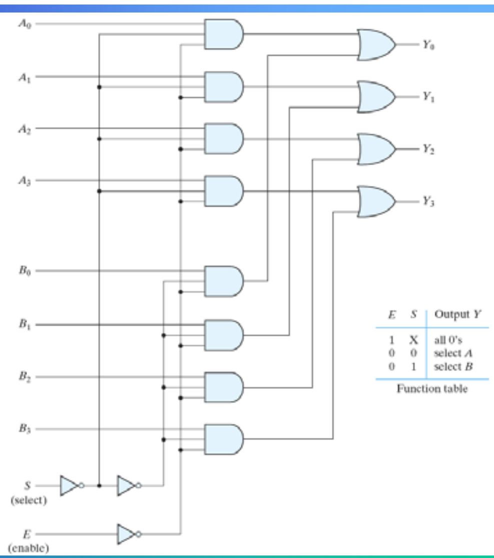
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



$$Y(s_1,s_0,I_3,I_2,I_1,I_0) = s_1'\cdot s_0'\cdot I_0 + s_1'\cdot s_0\cdot I_1 + s_1\cdot s_0'\cdot I_2 + s_1\cdot s_0\cdot I_3$$

Πολυπλέκτες με Κοινή Είσοδο Επίτρεψης

Η είσοδος Επίτρεψης (Ε) τοποθετείται για λόγους επέκτασης.



Υλοποίηση Συνάρτησης με Πολυπλέκτη

Κάθε πολυπλέκτης 2^n σε 1 μπορεί να υλοποιήσει οποιαδήποτε συνάρτηση n+1 μεταβλητών ως εξής:

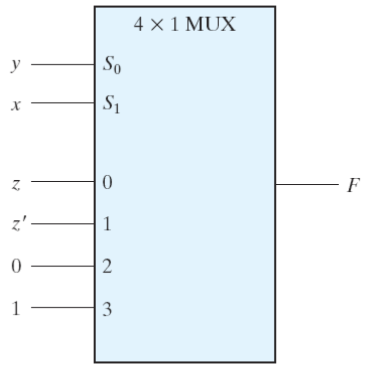
1. Βάζουμε τις n μεταβλητές στις εισόδους επιλογής.

2. Χρησιμοποιούμε την τελευταία μεταβλητή για τις εισόδους.

Παράδειγμα

F(x,y,z) = Σ(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	



Υλοποίηση Συναρτήσεων Boole

Αλγόριθμος υλοποίησης συνάρτησης n μεταβλητών με χρήση πολυπλέκτη:

1. Χρησιμοποιούμε πολυπλέκτη 2^{n-1}-σε-1 με n-1 γραμμές επιλογής.

1. Δημιουργούμε τον πίνακα αλήθειας της συνάρτησης.

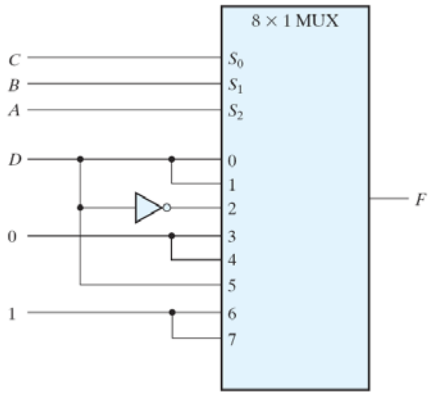
2. Συνδέουμε τις *n* - 1 περισσότερο σημαντικές μεταβλητές στις γραμμές επιλογής και κρατάμε τη δεξιότερη (λιγότερο σημαντική).

3. Για κάθε συνδυασμό των μεταβλητών των γραμμών επιλογής, εκφράζουμε την έξοδο ως συνάρτηση της τελευταίας μεταβλητής.

Παράδειγμα

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

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	



Βιβλιογραφία

1. Ψηφιακή Σχεδίαση (3η έκδοση), M. Morris Mano, Εκδόσεις Παπασωτηρίου, 2005
2. Ψηφιακή Σχεδίαση Αρχές και Πρακτικές, J. Wakerly, Εκδόσεις Κλειδάριθμος, 2002
3. Digital Design (4th edition), M. Morris Mano & M. Ciletti, Pearson Prentice Hall, 2007