Ψηφιακή Σχεδίαση

Ενότητα 7: κωδικοποιητές, κωδικοποιητές προτεραιότητας, πολυπλέκτες, υλοποίηση συνάρτησης με πολυπλέκτη, αποπλέκτες, πύλη 3ιών καταστάσεων, εισαγωγή στα ακολουθιακά

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org
Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών
http://arch.icte.uowm.gr/mdasyg

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

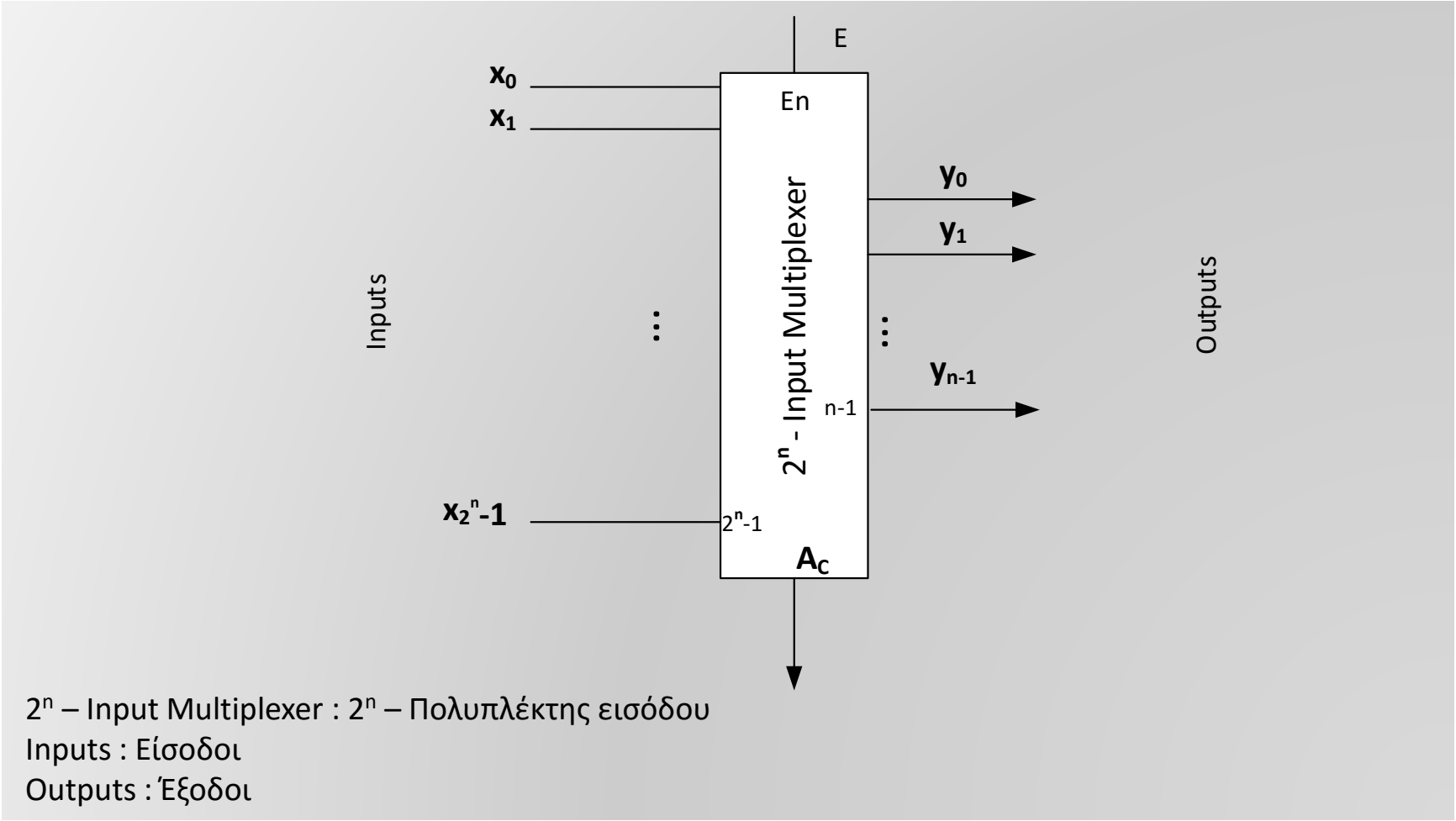
Σκοπός της ενότητας

- Ανάλυση των κωδικοποιητών.
- Ανάλυση των πολυπλεκτών και των αποπλεκτών.
- Ανάλυση των πυλών $3^{ων}$ καταστάσεων.
- Εισαγωγή στα ακολουθιακά κυκλώματα.

Κωδικοποιητές (1)

- Συνδυαστικό κύκλωμα που διεκπεραιώνει την αντίστροφη λειτουργία από αυτή του αποκωδικοποιητή.
- →Έχει 2^n εισόδους και n εξόδους.
- → ΜΟΝΟ 1 εισόδους μπορεί να έχει την τιμή 1 ανά πάσα στιγμή (αντιστοιχεί σε 1 από τους 2^n ελαχιστόρους).
- Οι έξοδοι παράγουν το δυαδικό ισοδύναμο της εισόδου με τιμή 1.

Κωδικοποιητές (2)



2^n – Input Multiplexer : 2^n – Πολυπλέκτης εισόδου
Inputs : Είσοδοι
Outputs : Έξοδοι

Κωδικοποιητές - Παράδειγμα

- Παράδειγμα: δυαδικός κωδικοποιητής 8-σε-3.

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_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

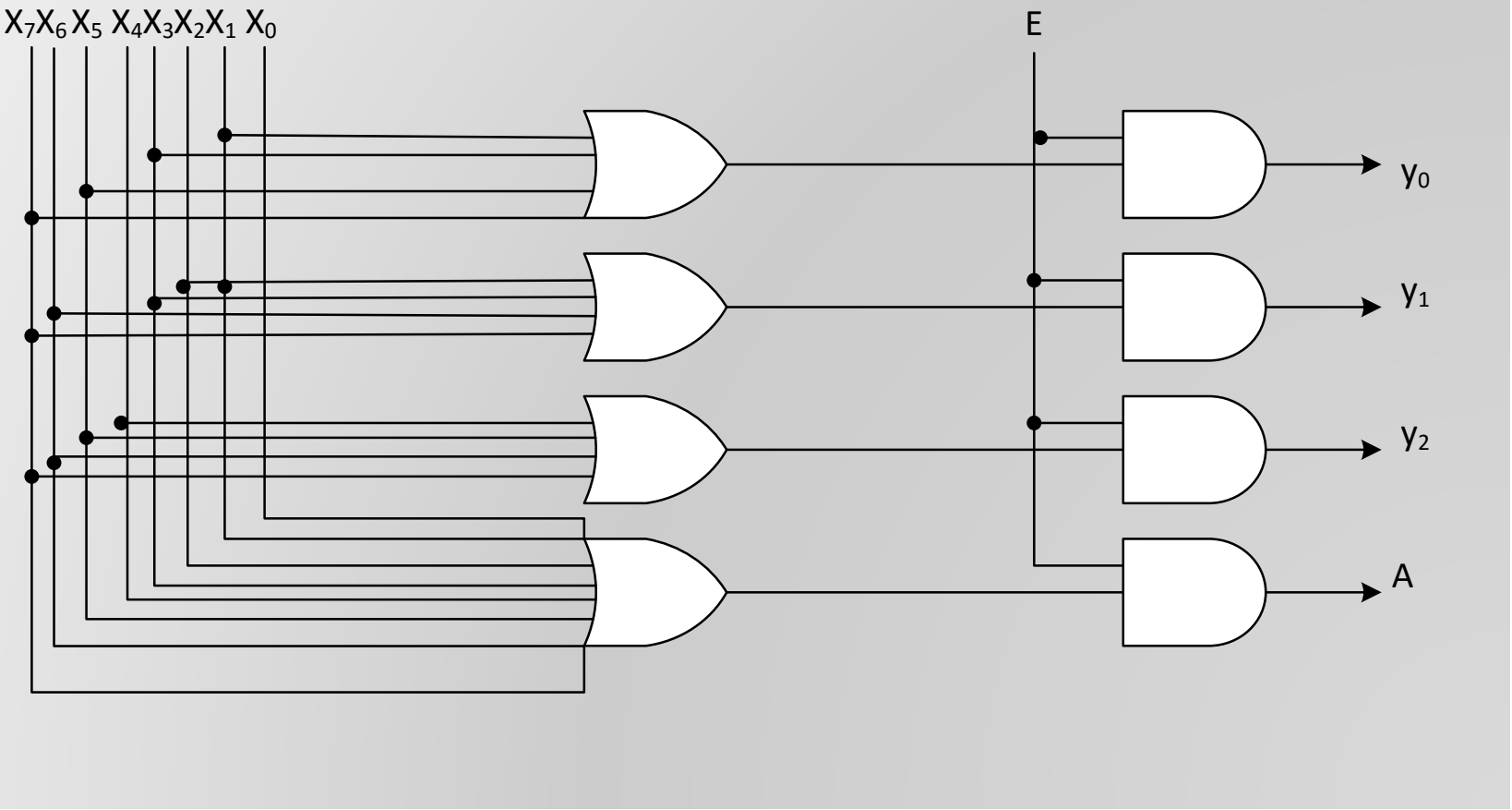
$$A_0 = D_1 + D_2 + D_5 + D_7$$

Είναι κωδικοποιητής από 8-αδικό σε 2αδικό.

$$A_1 = D_2 + D_3 + D_6 + D_7$$

$$A_2 = D_4 + D_5 + D_6 + D_7$$

Παράδειγμα Υλοποίησης



Θέματα Σχεδιασμού Κωδικοποιητών

- Υπάρχουν 2 αοριστίες που συσχετίζονται με τον σχεδιασμό ενός απλού κωδικοποιητή:

1. ΜΟΝΟ μια είσοδος μπορεί να είναι ενεργή (active ή High), ανά πάσα στιγμή. Αν ενεργοποιηθούν δύο μαζί, οι τιμές στις εξόδους είναι ακαθόριστες (π.χ. αν D_3 και D_6 είναι 1 μαζί, το αποτέλεσμα στις εξόδους είναι 111).
2. Αποτέλεσμα με όλο 0 μπορεί να παραχθεί όταν όλες οι είσοδοι είναι 0, ή όταν το D_0 είναι 1.

Κωδικοποιητές Προτεραιότητας

Επιλύουν τις αοριστίες που προαναφέρθηκαν.

- Περισσότερες από μια είσοδοι μπορούν να πάρουν την τιμή 1. Όμως, μια έχει προτεραιότητα από όλες τις άλλες.
- Ρητή ένδειξη όταν καμία από τις εισόδους δεν είναι 1. Bit εγκυρότητας.

Κωδικοποιητής Προτεραιότητας 4-σε-2 (1)

Πίνακας Αληθείας (συμπυκνωμένος)

Inputs				Outputs		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	X	X	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

Ποια είναι η σειρά προτεραιότητας;

Η σειρά προτεραιότητας είναι $D_3 > D_2 > D_1 > D_0$

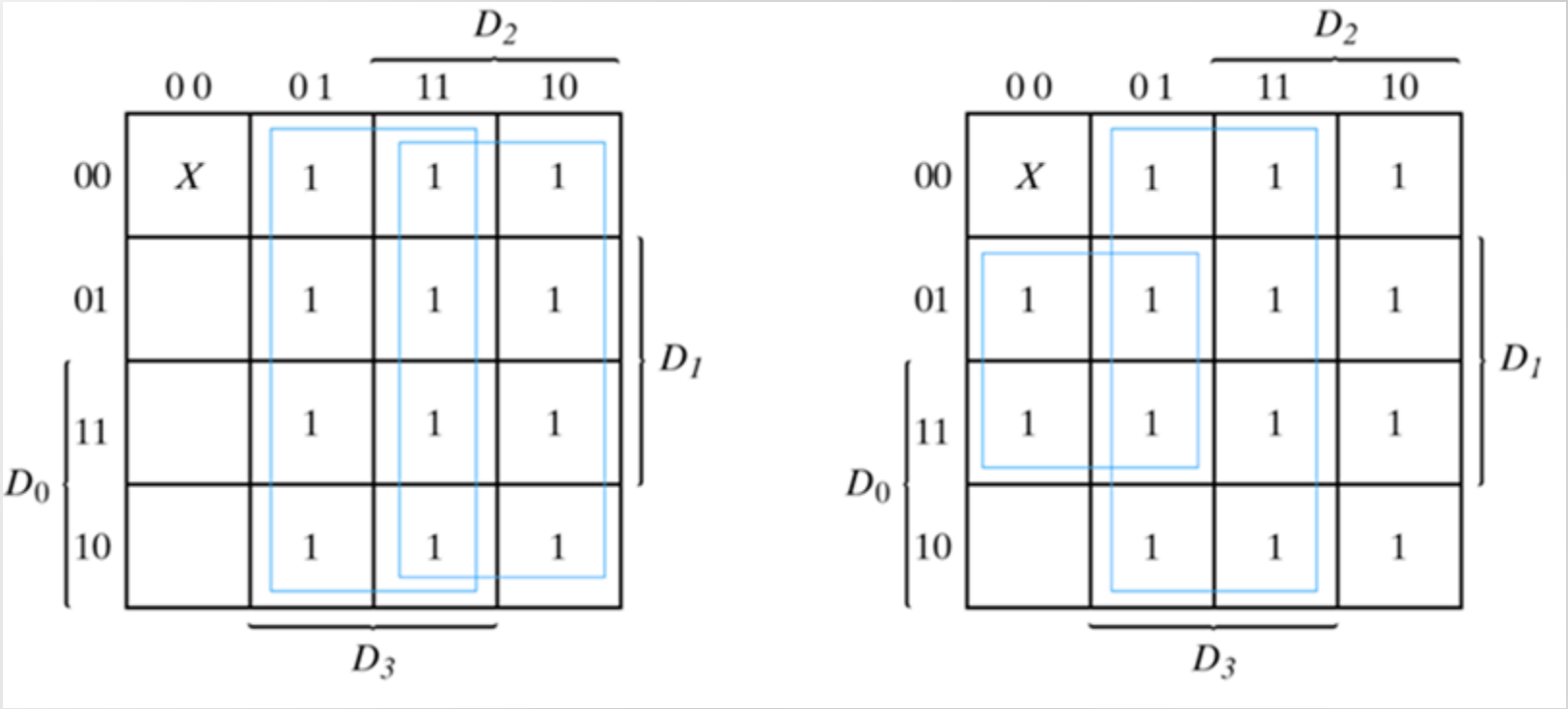
Κωδικοποιητής Προτεραιότητας 4-σε-2 (2)

Λειτουργία:

- Εάν δύο ή περισσότερες είσοδοι είναι 1 συγχρόνως, η είσοδος με τον πιο ψηλό αριθμοδείκτη παίρνει προτεραιότητα.
- Ο **έγκυρος δείκτης εξόδου (valid output indicator,** ορισμένος ως V στην προηγούμενη διαφάνεια), παίρνει την τιμή 1 μόνο όταν μια ή περισσότερες από τις εισόδους έχουν την τιμή 1.

→ $V = D_3 + D_2 + D_1 + D_0$

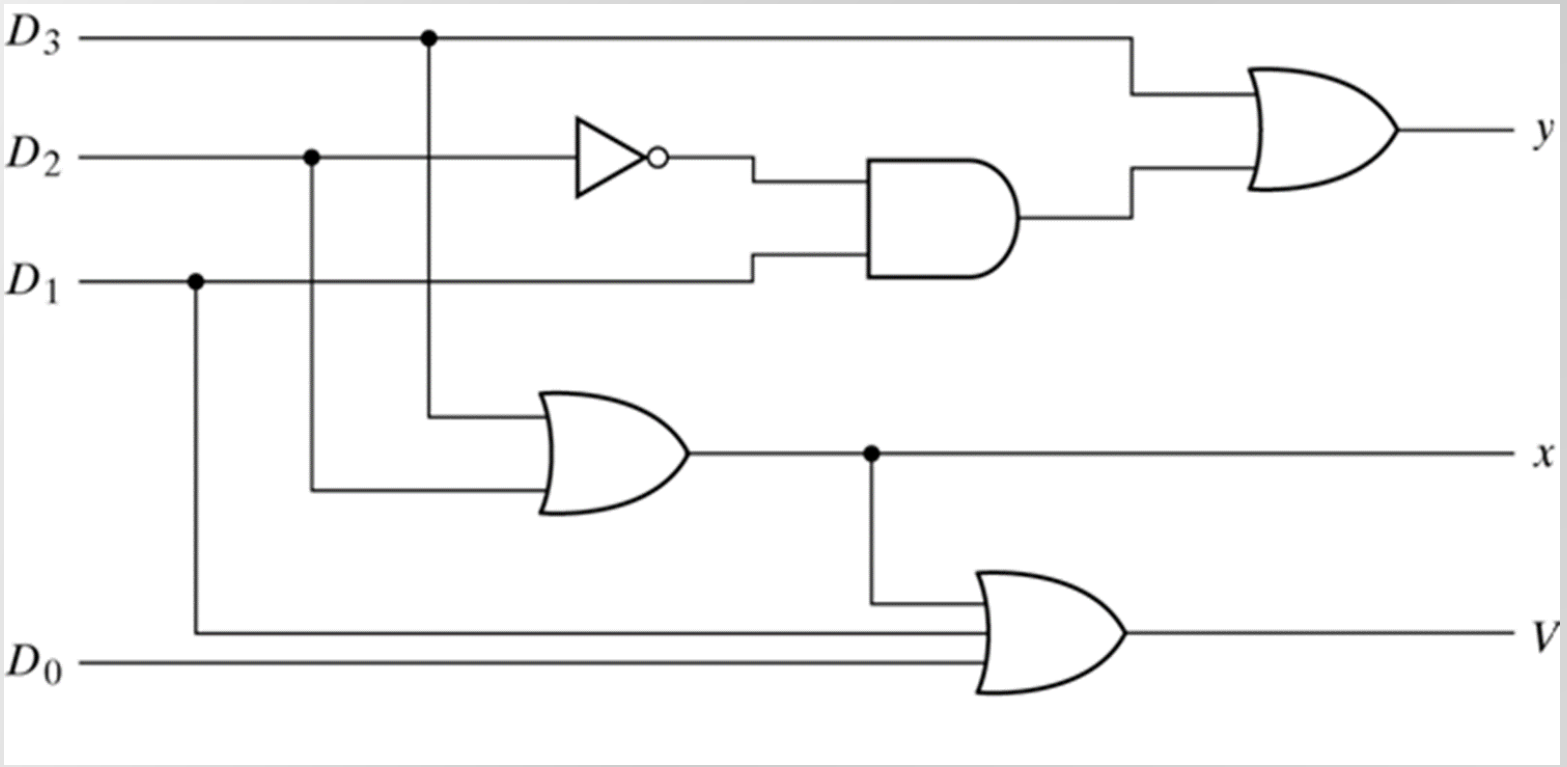
Κωδικοποιητής Προτεραιότητας 4-σε-2: Πίνακες αλήθειας



$x = D_2 + D_3$

$y = D_3 + D_1D_2'$

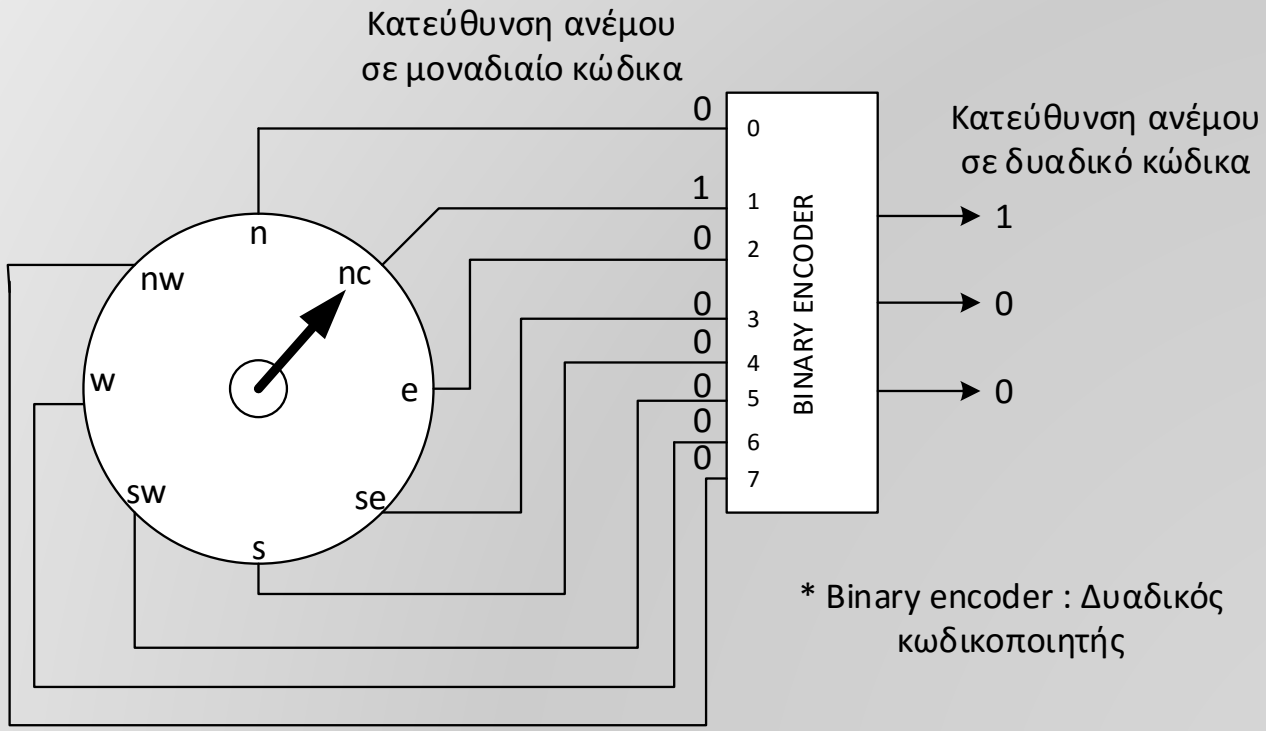
Κωδικοποιητής Προτεραιότητας 4-σε-2: Υλοποίηση



Κωδικοποιητής Προτεραιότητας 8-σε-3

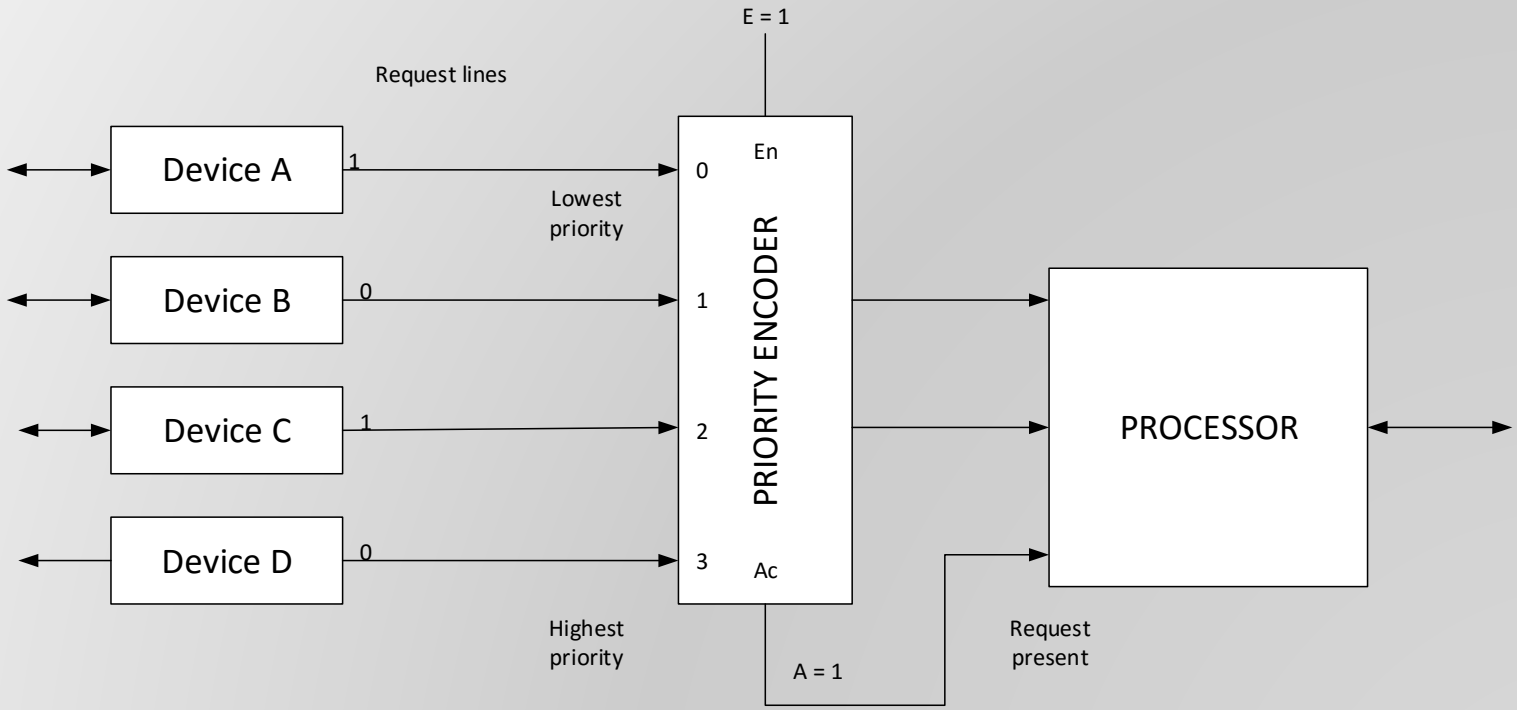
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Z_0	Z_1	Z_2	NR
0	0	0	0	0	0	0	0	X	X	X	1
X	X	X	X	X	X	X	1	1	1	1	0
X	X	X	X	X	X	1	0	1	1	0	0
X	X	X	X	X	1	0	0	1	0	1	0
X	X	X	X	1	0	0	0	1	0	0	0
X	X	X	1	0	0	0	0	0	1	1	0
X	X	1	0	0	0	0	0	0	1	0	0
X	1	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0

Χρήσεις Δυαδικού Κωδικοποιητή: Ένα παράδειγμα



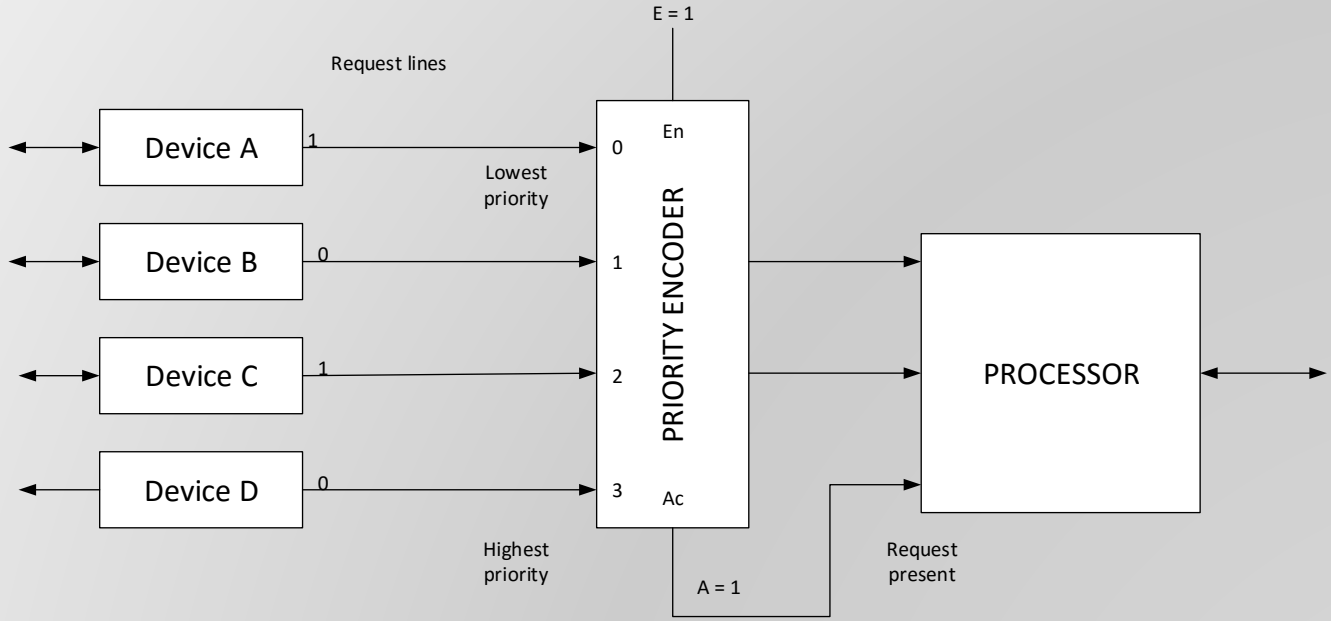
- Δυαδική κωδικοποίηση κατεύθυνσης ανέμου

Χρήσεις Δυαδικού Κωδικοποιητή στην αρχιτεκτονική υπολογιστών (1)



Επίλυση αιτημάτων διακοπών (interrupt requests) με χρήση κωδικοποιητή

Χρήσεις Δυαδικού Κωδικοποιητή στην αρχιτεκτονική υπολογιστών (2)



Επίλυση αιτημάτων διακοπών (interrupt request) με χρήση κωδικοποιητή
Lowest priority: Χαμηλότερης προτεραιότητας
Highest priority: Υψηλότερης προτεραιότητας
Request lines: Απαιτούμενες γραμμές
Request present: Αίτημα του παρόντος
Priority Encoder: Κωδικοποιητής προτεραιότητας
Processor: Επεξεργαστής
Device: Συσκευή

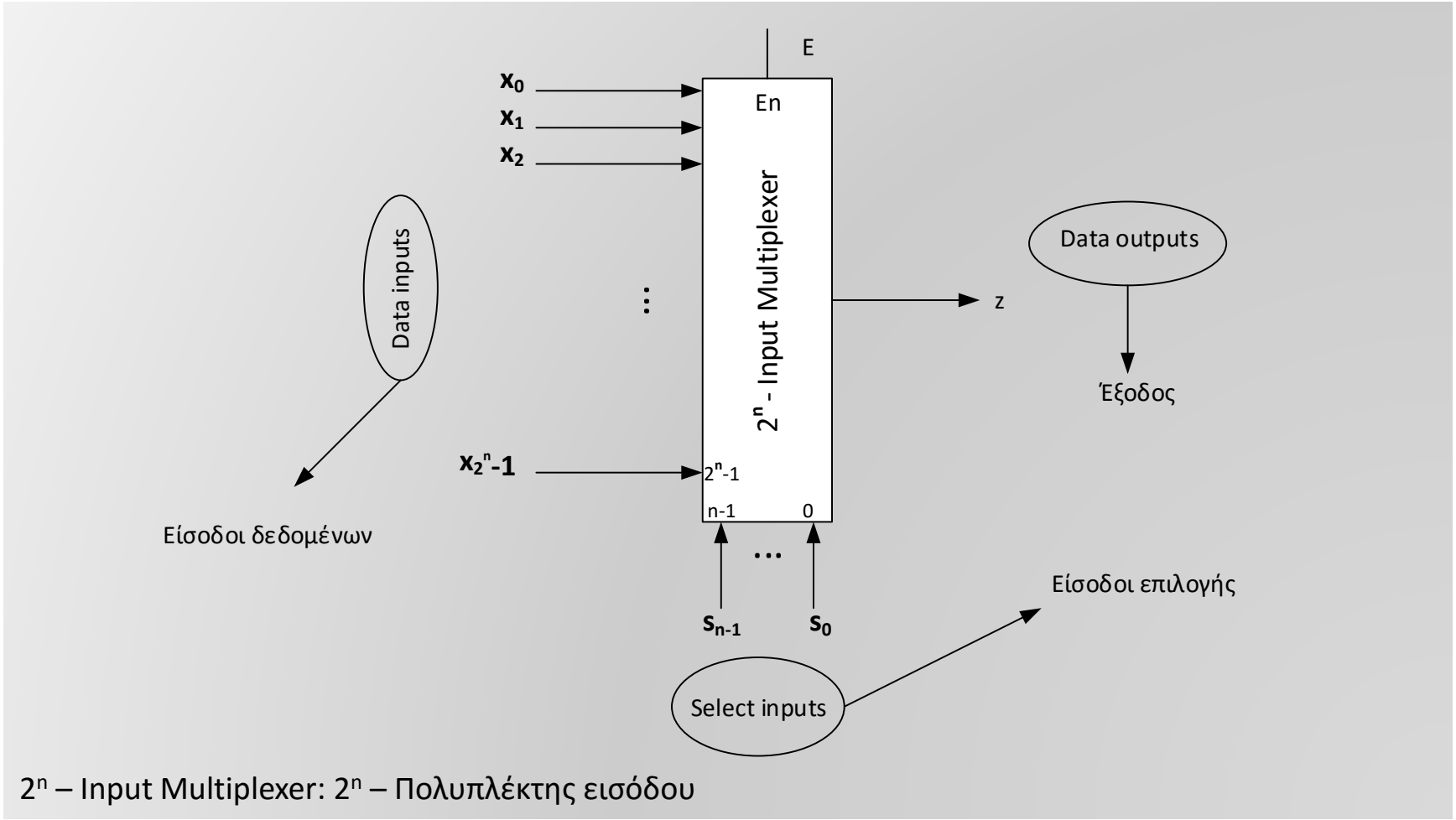
Παράδειγμα προτεραιοτήτων στον επεξεργαστή PIC

Bit	Disable IRQ	Function
7	IRQ7	Parallel Port
6	IRQ6	Floppy Disk Controller
5	IRQ5	Reserved / Sound Card
4	IRQ4	Serial Port
3	IRQ3	Serial Port
2	IRQ2	PIC2
1	IRQ1	Keyboard
0	IRQ0	System Timer

Πολυπλέκτες (multiplexers) (1)

- Κύκλωμα που «επιλέγει» δυαδική πληροφορία από μια τις εισόδους και την κατευθύνει στη μοναδική έξοδο.
- Επίσης γνωστό ως «επιλογέας» (selection circuit).
- Η επιλογή ελέγχει από ένα σύνολο εισόδων, ο αριθμός των οποίων εξαρτάται από το # των εισόδων δεδομένων.
- Για ένα πολυπλέκτη 2^n-σε-1 , υπάρχουν $2^n + n$ είσοδοι:
  - 2^n είσοδοι δεδομένων και
  - n είσοδοι επιλογής, έτσι ώστε ο συνδυασμός των bit τους καθορίζει την είσοδο δεδομένων που θα επιλέγει.

Πολυπλέκτες (multiplexers) (2)

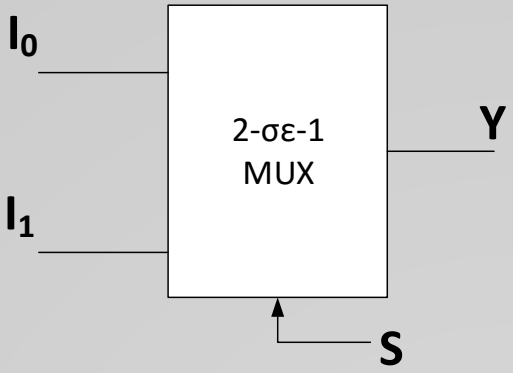


2^n – Input Multiplexer: 2^n – Πολυπλέκτης εισόδου

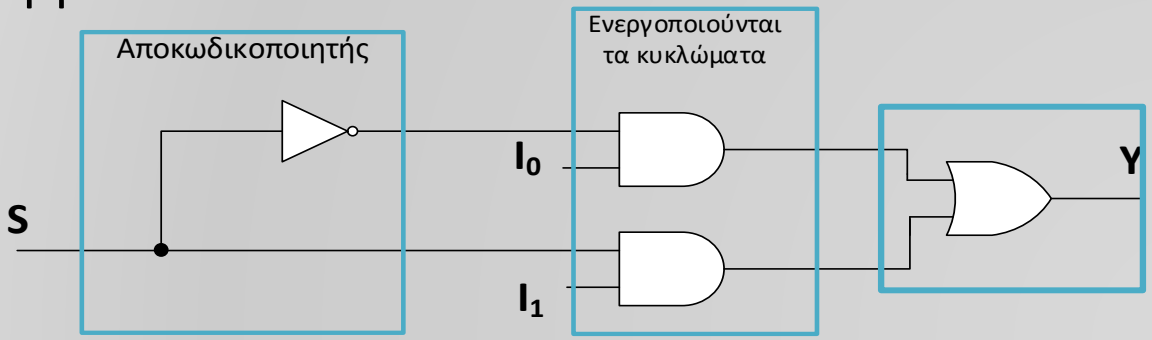
2-σε-1 MUX (1)

- Αφού υπάρχουν 2 είσοδοι δεδομένων, $2 = 2^1 \rightarrow n = 1$
- Υπάρχει μια είσοδος επιλογής S:
  - S = 0 επιλέγει την είσοδο I_0
  - S = 1 επιλέγει την είσοδο I_1
- Υλοποιεί την συνάρτηση:

$Y = S'I_0 + SI_1$



- Το λογικό διάγραμμα:



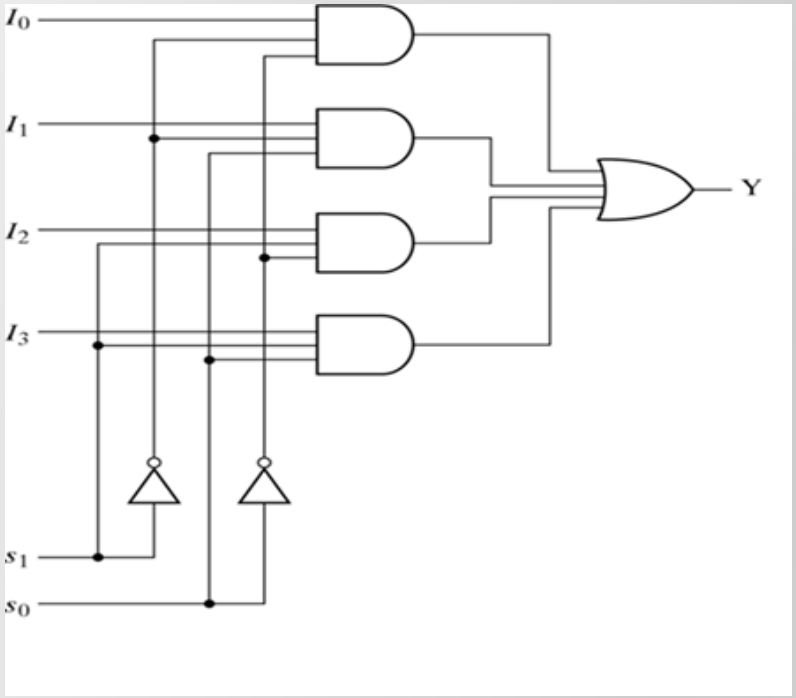
2-σε-1 MUX (2)

- Προσέξετε ότι τα διάφορα μέρη του πολυπλέκτη δείχνουν:
  - Ένα 1-σε-2 Αποκωδικοποιητή.
  - Δυο κυκλώματα ενεργοποίησης (enable circuits).
  - Μια πύλη OR 2-εισόδων.
- Τα πιο πάνω συδυάζονται για να μας δώσουν τον πολυπλέκτη, τα κυκλώματα ενεργοποιήσης και η πύλη OR 2-εισόδων δίνουν ένα κύκλωμα 2 x 2 AND-OR, όπου οι 4 είσοδοι του προέρχονται από τις 2 εισόδους δεδομένων και τις 2 εισόδους του αποκωδικοποιητή:
  - 2 είσοδοι δεδομένων.
  - 1-σε-2 αποκωδικοποιητή (παράγουν τους ελαχιστόρους).
  - 2 x 2 AND-OR.
- Γενικά, για έναν πολυπλέκτη 2^n-σε-1:
  - 2^n είσοδοι δεδομένων.
  - n-σε-2^n αποκωδικοποιητή .
  - 2^n x 2 AND-OR.

Παράδειγμα: 4-σε-1 MUX (1)

- Επιλέγεται κάθε φορά ένα bit από τα I_0 έως I_3 .

  Είναι λοιπόν 4-σε-1 MUX 1 bit.

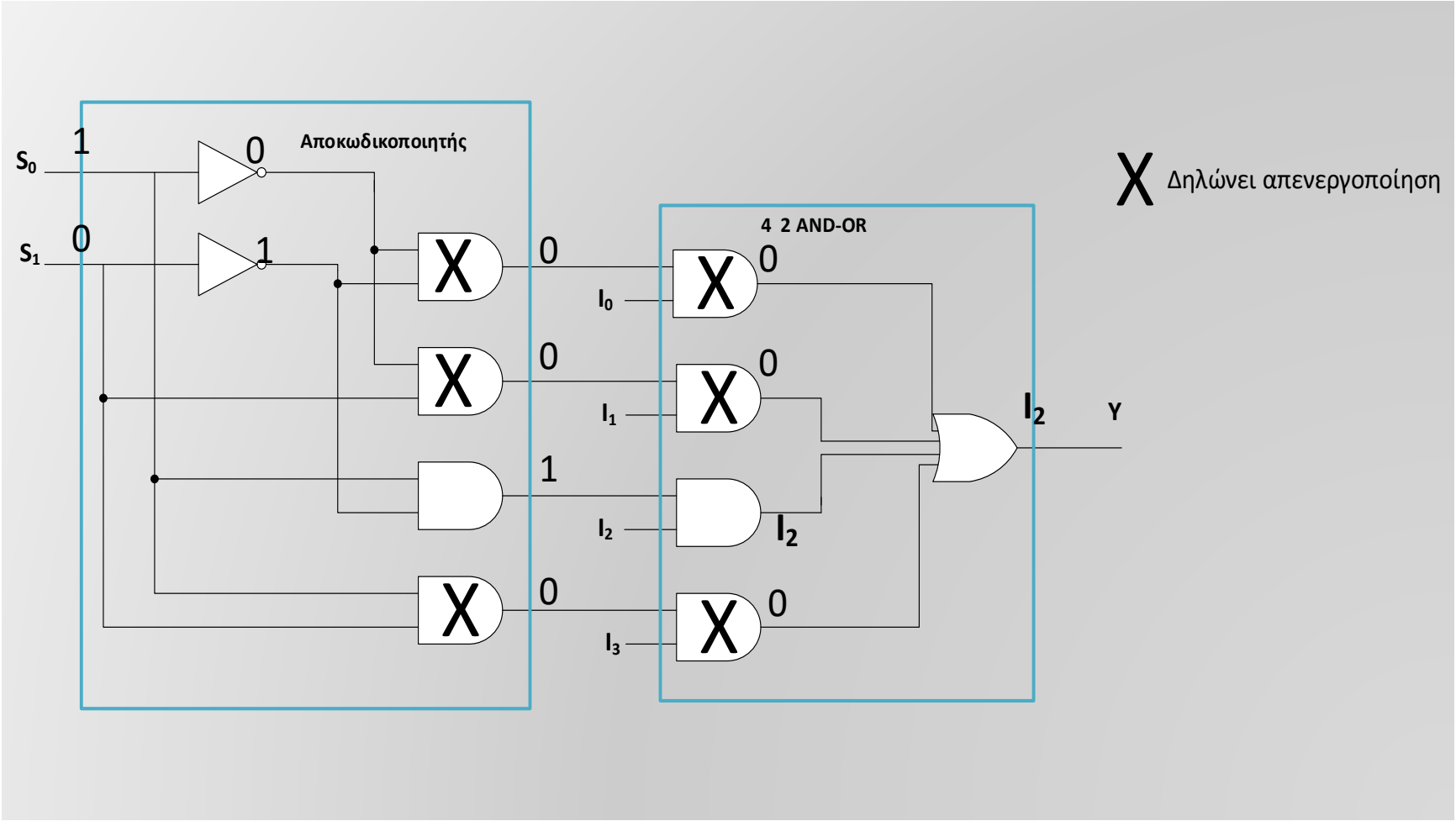


(α) Λογικό διάγραμμα

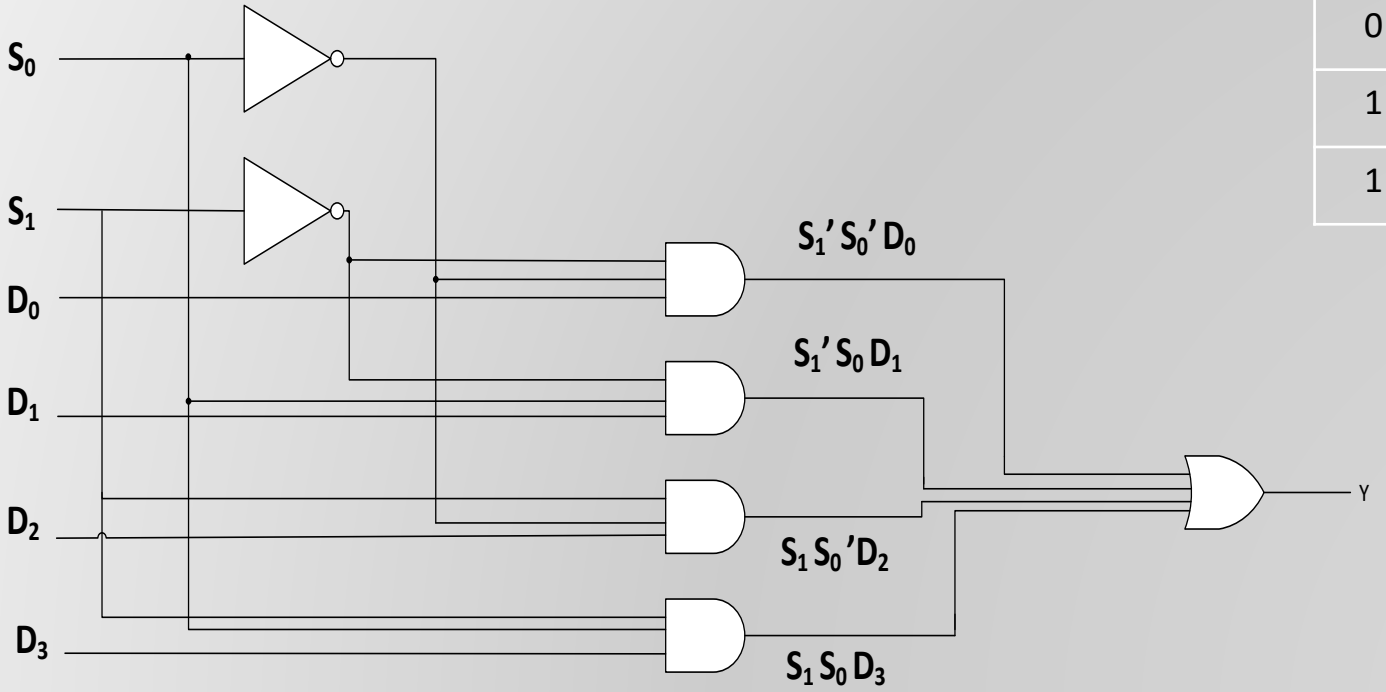
s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(β) Πίνακας συνάρτησης

Παράδειγμα: 4-σε-1 MUX (2)

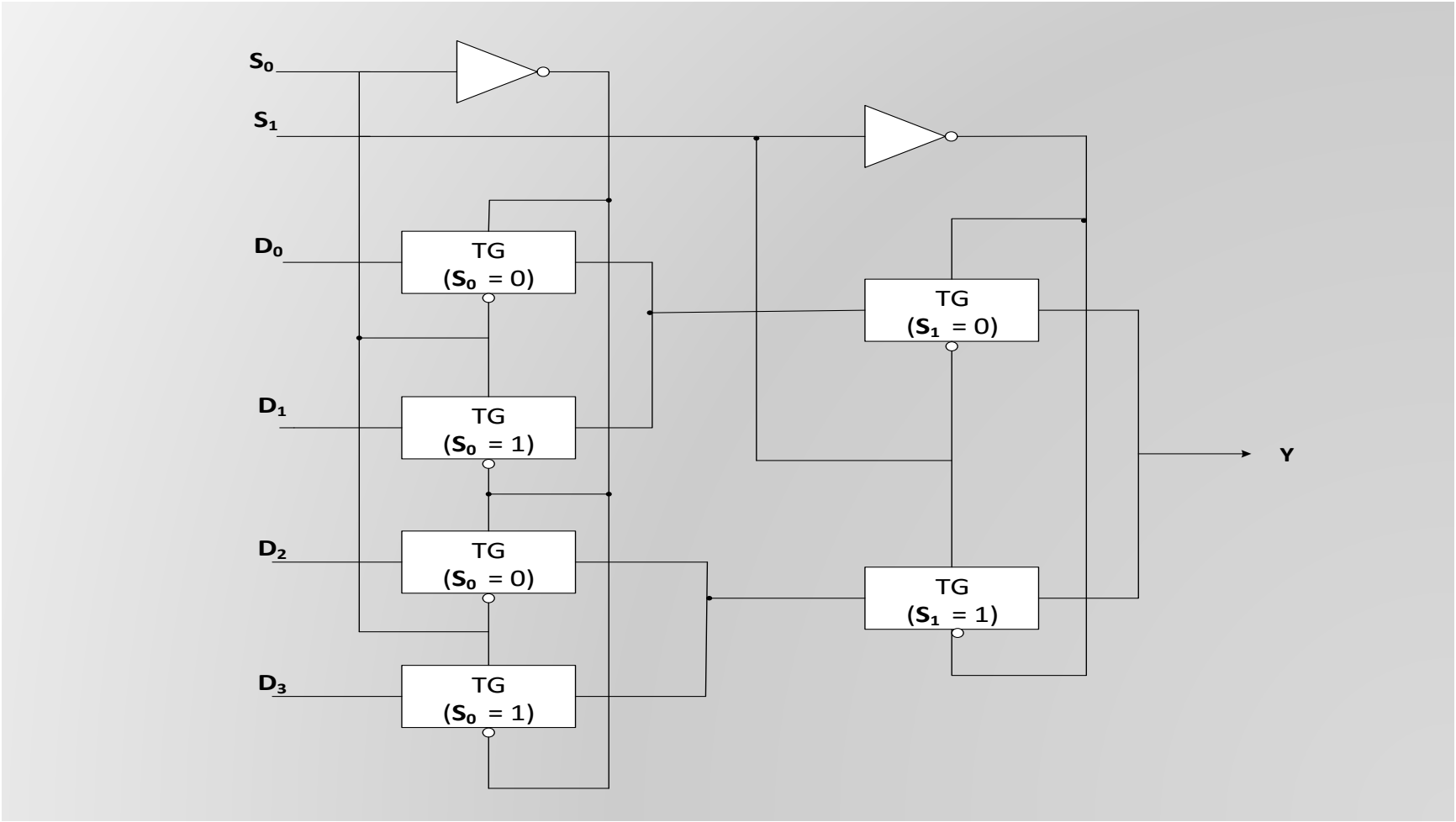


Παράδειγμα: 4-σε-1 MUX (βελτιστοποίηση με πύλες 3 εισόδων)



S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

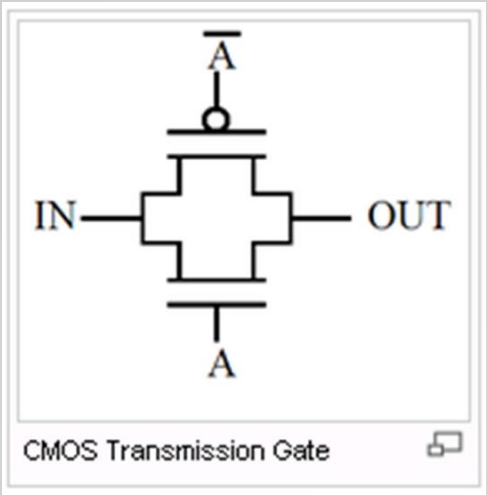
Παράδειγμα: 4-σε-1 MUX (με πύλες Μετάβασης)



Πύλη μετάδοσης

- Μια πύλη μετάδοσης είναι ένα ηλεκτρικό στοιχείο. Είναι ένας καλός μη-μηχανικός αναμεταδότης , βασισμένος στην τεχνολογία CMOS.
- Συχνά είναι γνωστό ως αναλογική πύλη ή αναλογικός διακόπτης ή ηλεκτρικός αναμεταδότης , ανάλογα την χρήση.

Επιτρέπει ορισμένες λογικές συναρτήσεις που πρέπει να εφαρμοστούν με λιγότερα τρανζίστορ χρησιμοποιώντας άλλα λογικά CMOS . Με αυτή την λογική σχεδιάζονται οι πολυπλέκτες.



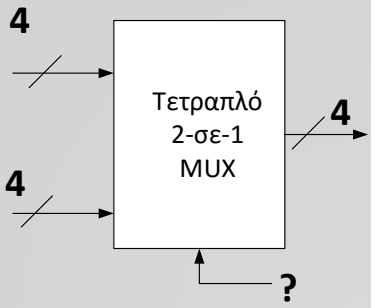
Πολυπλέκτες

- Μέχρι στιγμής, έχουμε εξετάσει επιλογή δυαδικής πληροφορίας ενός-bit από MUX. Τι γίνεται αν θέλουμε να επιλέξομε πληροφορία των m-bit (data / words);

→ Συνδυάζουμε κυκλώματα MUX παράλληλα, με κοινές εισόδους επιλογής και ενεργοποίησης.

- Παράδειγμα: Βρείτε το λογικό διάγραμμα ενός πολυπλέκτη που επιλέγει μεταξύ 2 συνόλων από εισόδους 4-bit …

→ Τετραπλός 2-σε-1 πολυπλέκτης (Quad 2-to-1 MUX).

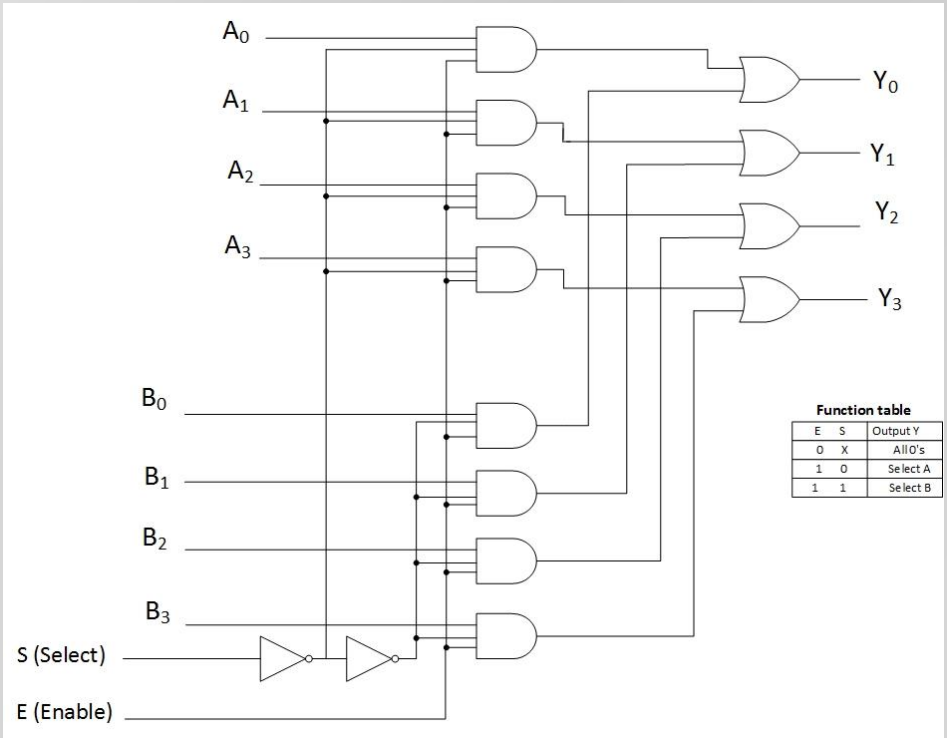


Παράδειγμα: Τετραπλό (QUAD) 2-σε-1 MUX (4-bits)

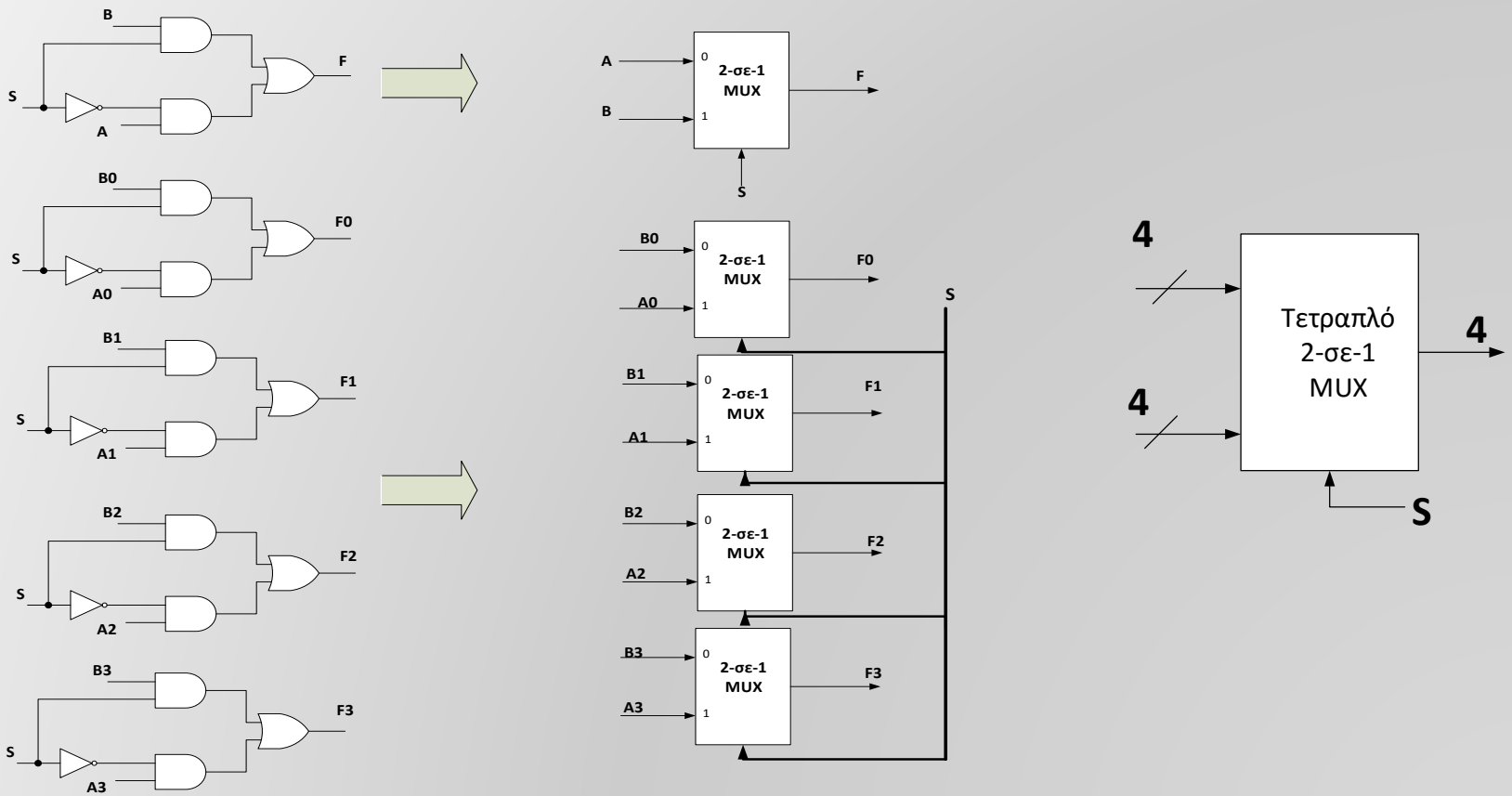
- Χρησιμοποιεί τέσσερις MUX 2-σε-1, με κοινή είσοδο επιλογής (S) και κοινή είσοδο ενεργοποίησης (E).
- Η είσοδος επιλογής S επιλέγει μεταξύ των A_i 's και B_i 's και στέλνει στα αντίστοιχα Y_i 's.
- Το σήμα ενεργοποίησης E αφήνει τα επιλεγμένα δεδομένα εισόδου να φτάσουν στις εξόδους

(E = 1 για ενεργή λειτουργία) ή όλοι οι έξοδοι μένουν σταθεροί σε 0

(E = 0 για απενεργοποίηση) .

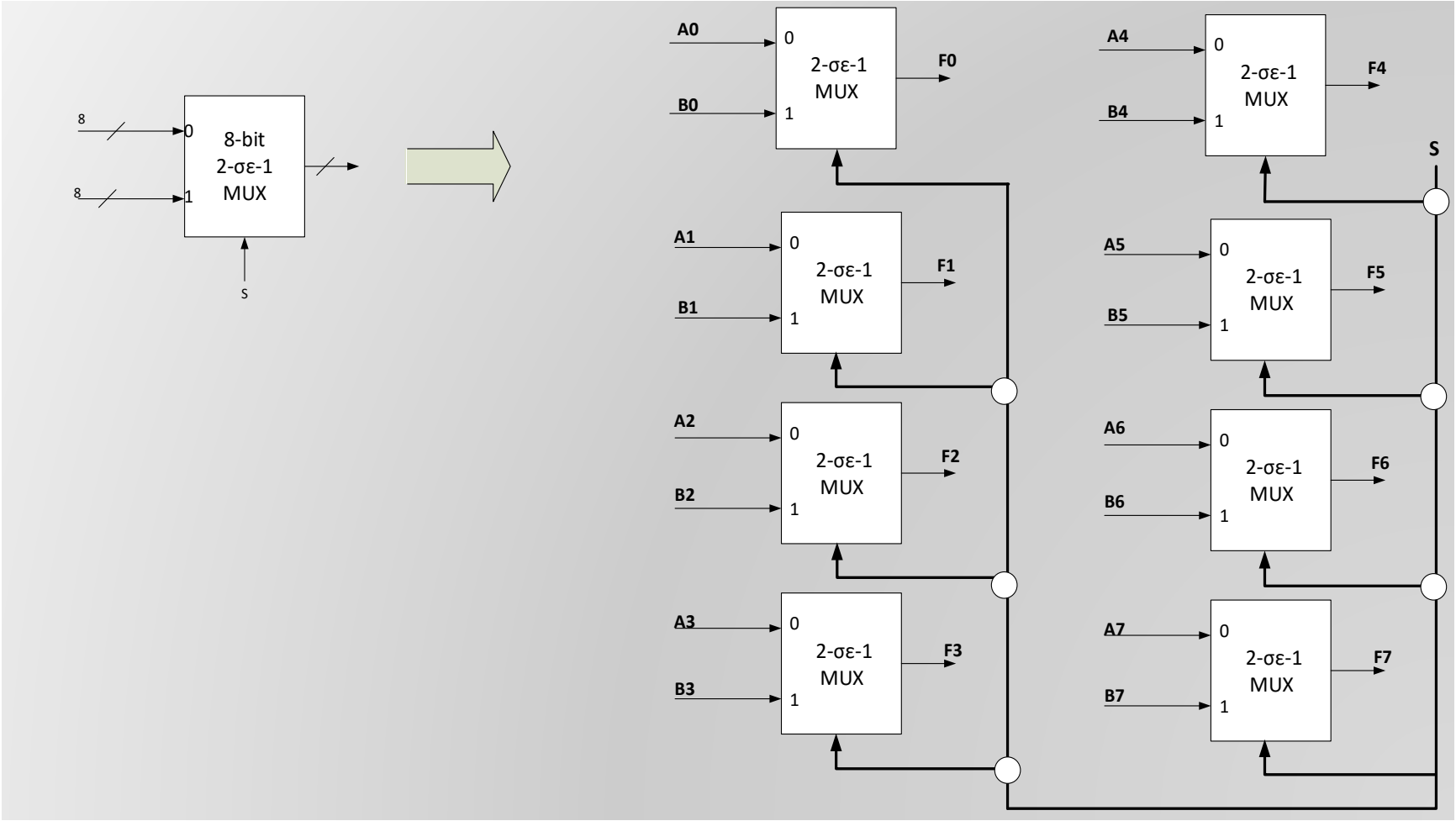


Παράδειγμα: Τετραπλό (QUAD) 2-σε-1 MUX (διαφορετική υλοποίηση)

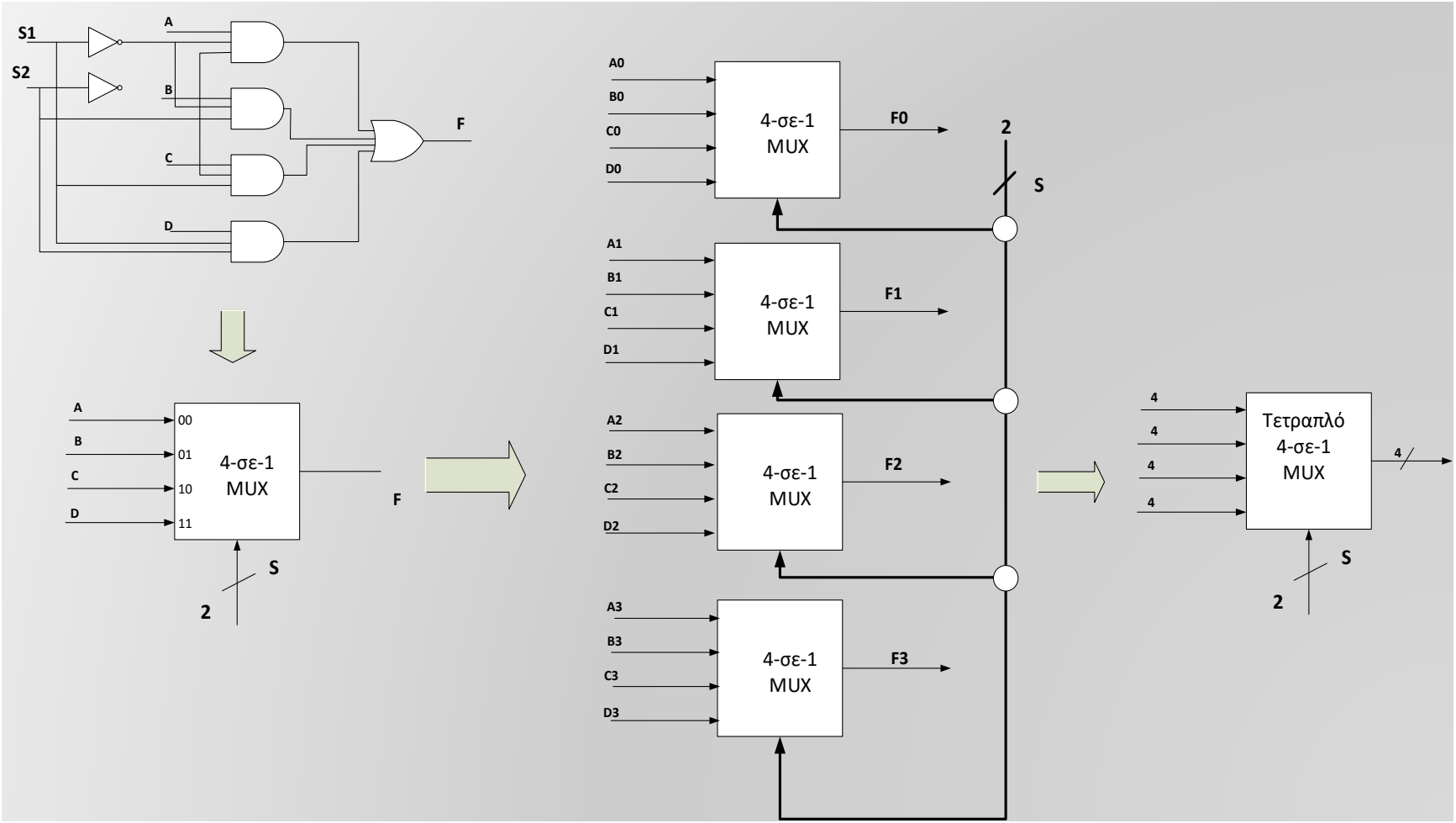


- Χρησιμοποιεί τέσσερις MUX 2-σε-1, με κοινή είσοδο επιλογής (S).
- Η είσοδος επιλογής S επιλέγει μεταξύ των A_i 's και B_i 's και στέλνει στα αντίστοιχα Y_i 's.

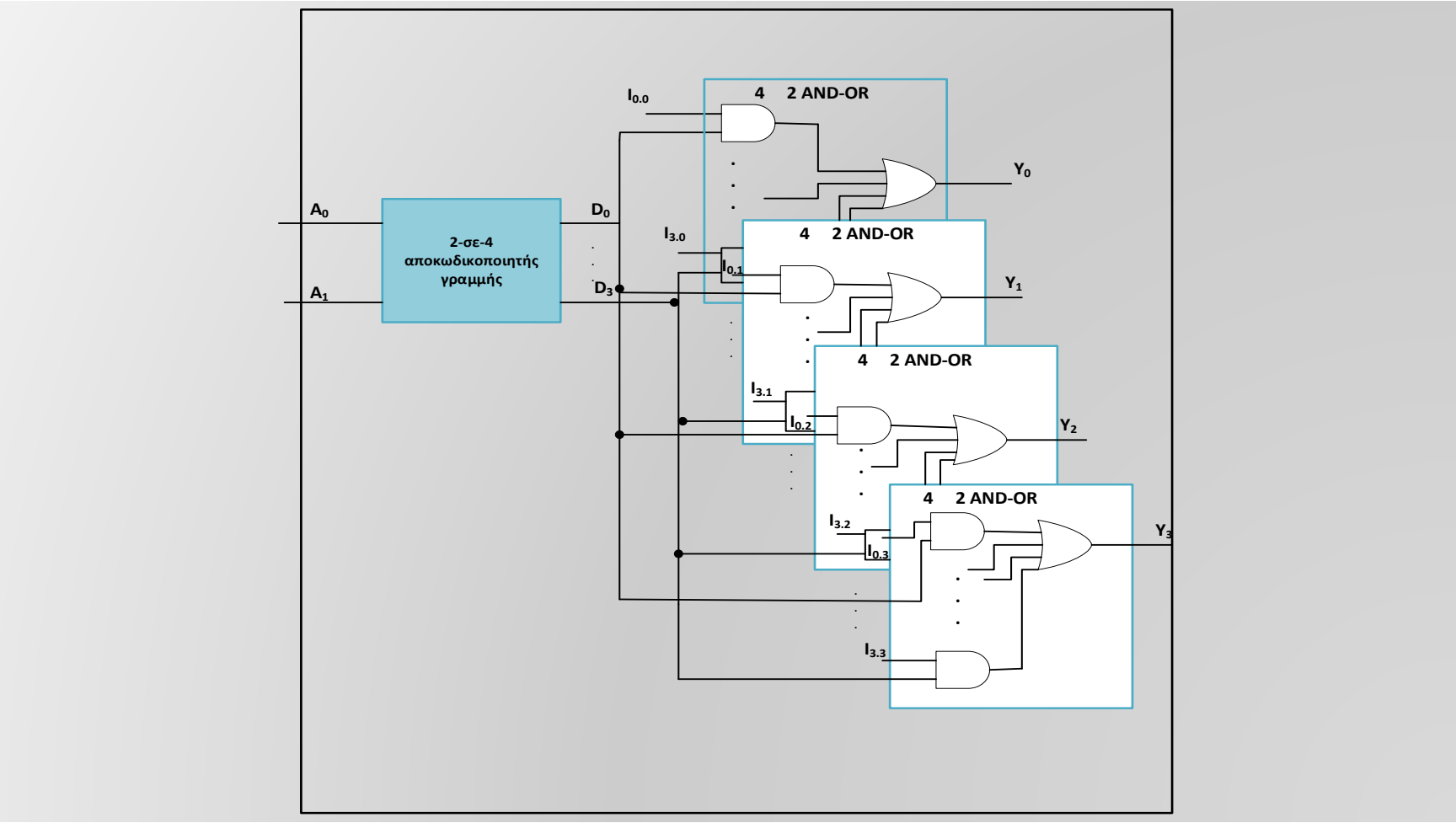
Άλλο παράδειγμα: 2-to-1 MUX (8 bit)



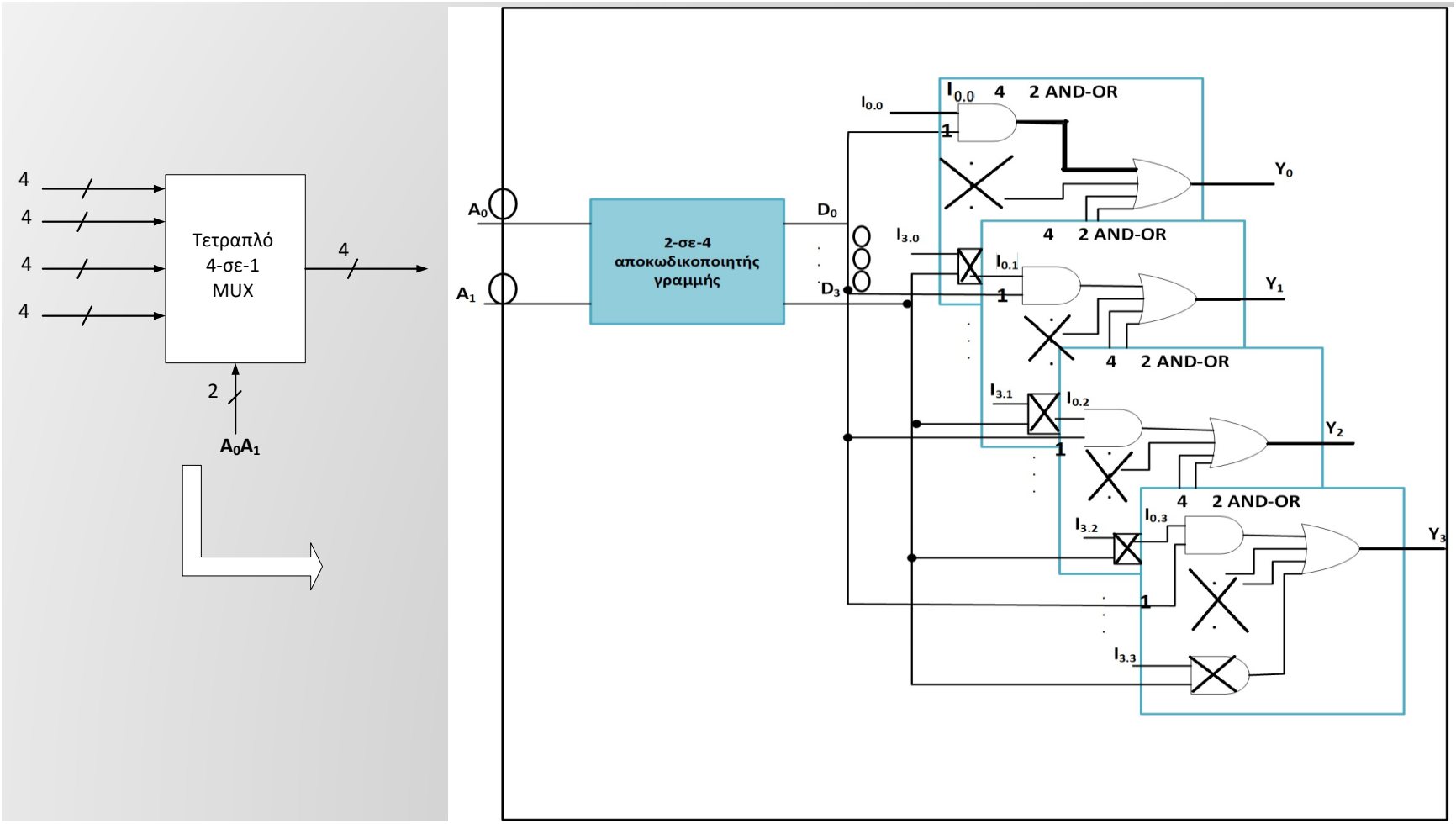
Άλλο παράδειγμα: 4-to-1 MUX (Quad) 4 bit



Άλλο παράδειγμα: (Quad) 4 bit 4-to-1 MUX (1)



Άλλο παράδειγμα: (Quad) 4 bit 4-to-1 MUX (2)



Μπορούμε να υλοποιήσουμε τις συναρτήσεις Boole με πολυπλέκτες

- Οποιαδήποτε συνάρτηση Boole n μεταβλητών μπορεί να υλοποιηθεί χρησιμοποιώντας ένα πολυπλέκτη μεγέθους 2^n -σε-1 και μια πύλη NOT.
- Αναμενόμενο, αφού ένας πολυπλέκτης αποτελείται από έναν αποκωδικοποιητή, με τις εξόδους του να καταλήγουν σε μια πύλη OR.
- Τα σήματα ΕΠΙΛΟΓΗΣ παράγουν τους ελαχιστόρους της συνάρτησης.
- Τα σήματα ΔΕΔΟΜΕΝΩΝ καθορίζουν τους ελαχιστόρους που οδηγούν στην πύλη OR.

Υλοποίηση συναρτήσεων Boole με πολυπλέκτες

- Μια λογική συνάρτηση n μεταβλητών υλοποιείται με έναν πολυπλέκτη που έχει n-1 εισόδους επιλογής.
- Οι πρώτες n-1 μεταβλητές της συνάρτησης συνδέονται με τις εισόδους επιλογής.
- Η εναπομένουσα μεταβλητή της συνάρτησης χρησιμοποιείται για τις εισόδους δεδομένων.
- Αν αυτή η μοναδική μεταβλητή συμβολίζεται με z τότε κάθε είσοδος δεδομένων πρέπει να έχει τιμή ίση με z, z’, 1, 0.

Ακολουθεί παράδειγμα…

Παράδειγμα (1)

- Έστω F(x, y, z) = Σ(1, 2, 6, 7)
- **n = 3** άρα θέλουμε έναν πολυπλέκτη που έχει **n - 1** εισόδους επιλογής.
- 2 είσοδοι επιλογής άρα ο πολυπλέκτης είναι **4-σε-1**.
- Οι 2 πρώτες μεταβλητές x, y τροφοδοτούν τις γραμμές επιλογής, το **x** πηγαίνει στην **S1** το **y** πηγαίνει στην είσοδο **S0**.

Παράδειγμα (2)

- Οι τιμές για τις γραμμές εισόδου καθορίζονται από τον πίνακα αληθείας.
- Στις γραμμές εισόδου τοποθετούμε **z, z', 1, 0**
- Εξετάζουμε τον πίνακα αληθείας και τοποθετούμε ένα από τα παραπάνω στην αντίστοιχη είσοδο.

Παράδειγμα (3)

- F(x, y, z) = Σ(1, 2, 6, 7)
- Ομαδοποιούμε ανά 2 τις γραμμές του πίνακα.
- Οι 2 πρώτες γραμμές του πίνακα αληθείας.

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

F = z

Παράδειγμα (4)

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

F=z

Δηλαδή στην είσοδο D0 που αντιστοιχεί στην επιλογή x = 0 , y = 0 πρέπει να τοποθετηθεί η z.

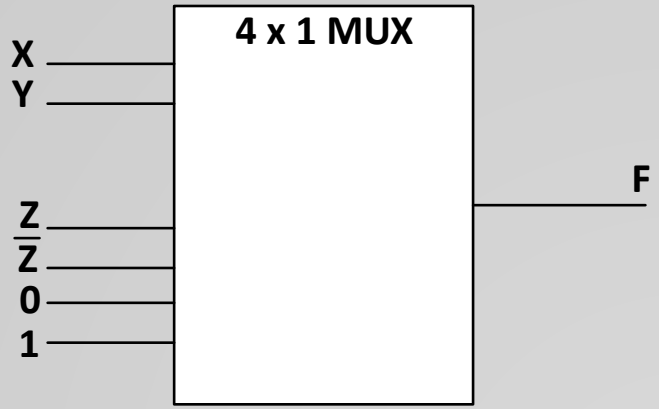
Ομοίως για τις επόμενες εισόδους, δηλαδή για τις
D1 (x = 0, y = 1) , D2 (x = 1, y = 0) , D3 (x = 1, y = 1).

Παράδειγμα (5)

- F(X, Y, Z) = X'Y'Z + X'YZ' + XYZ' + XYZ = Σm(1, 2, 6, 7)
- Υπάρχουν n = 3 είσοδοι, άρα , χρειαζόμαστε ένα 2^2-to-1 MUX.
- Οι πρώτες n - 1 (=2) είσοδοι υπηρετούν ως είσοδοι επιλογής.

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

(α) Πίνακας αληθείας



(β) Εφαρμογή πολυπλέκτη

Συστηματική μέθοδος για υλοποίηση συναρτήσεων με MUX

Για μία συνάρτηση n-μεταβλητών (π.χ., f(A, B, C, D)):

1. Χρειάζεται ένας 2^{n-1}-to-1 MUX, με n-1 εισόδους επιλογής.
2. Υπολογίζουμε τον πίνακα αληθείας της συνάρτησης, με την σειρά μεταβλητών A > B > C > D (A είναι το MSB και D το LSB).
3. Ορίζουμε τις πιο σημαντικές n - 1 μεταβλητές στις n - 1 εισόδους επιλογής (π.χ., A, B, C).
4. Εξετάζουμε ζεύγη γειτονικών γραμμών στον πίνακα (μόνο το LSB διαφέρει, π.χ. D = 0 and D = 1).
5. Καθορίζουμε κατά πόσο η τιμή της συνάρτησης (έξοδος) για το συνδυασμό (A, B, C, 0) και (A, B, C, 1) είναι (0, 0), (0, 1), (1, 0) ή (1, 1).
6. Για κάθε συνδυασμό (A, B, C), ορίζουμε 0, D, D’, ή 1 στην είσοδο δεδομένων που αντιστοιχεί στο (A, B, C).

Άλλο παράδειγμα (1)

- Θεωρήστε F(A, B, C) = Σm(1, 3, 5, 6).
- Μπορούμε να υλοποιήσουμε τη συνάρτηση με ένα 4-σε-1 MUX.
- Η σειρά μεταβλητών είναι A > B > C.
- Τότε, τα σήματα επιλογής ορίζονται ως S_1 = A και S_0 = B.
- Βρείτε τον πίνακα αληθείας…

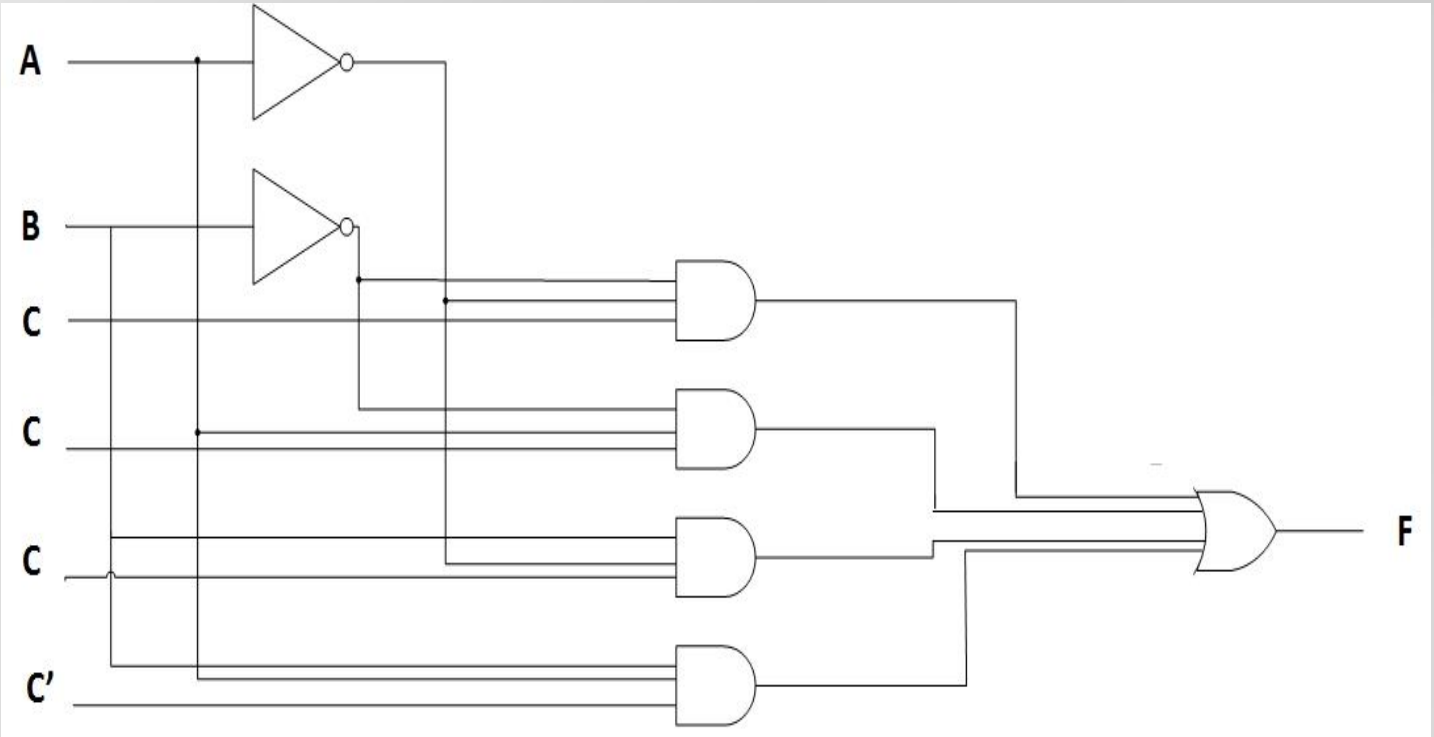
Παράδειγμα (6)

- Όταν A = B = 0, F = C
- Όταν A = 0, B = 1, F = C
- Όταν A = 1, B = 0, F = C
- Όταν A = B = 1, F = C’

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

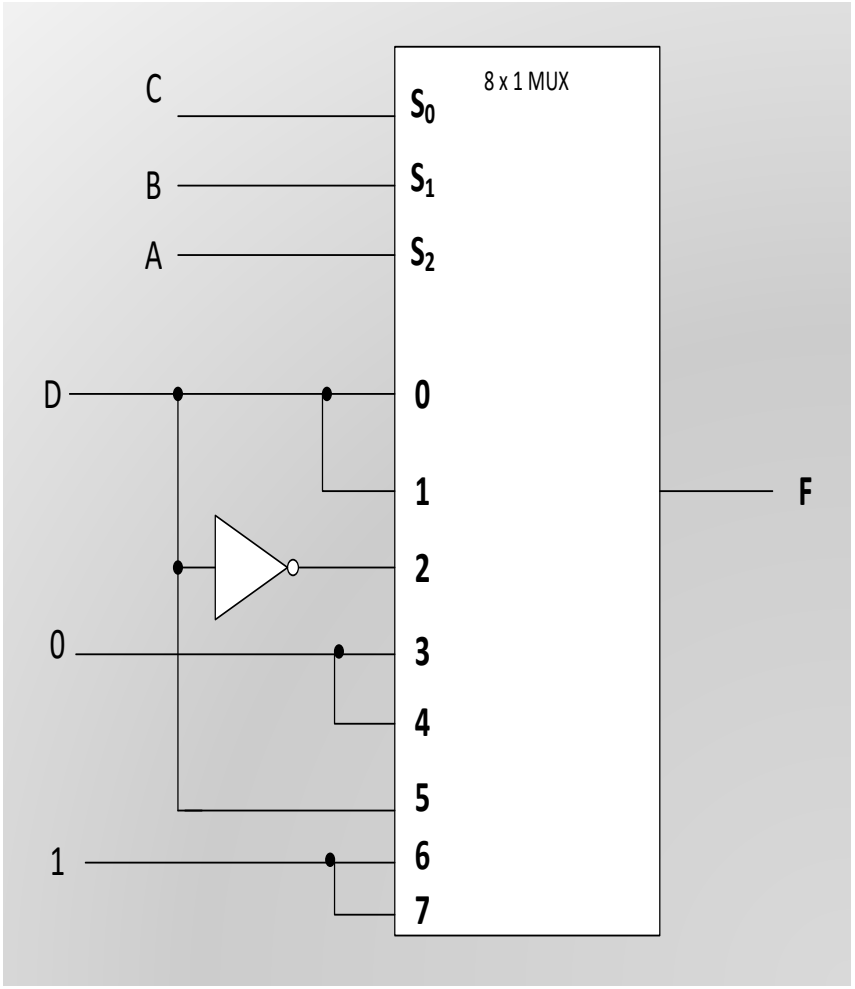
Άλλο παράδειγμα (2)

- Υλοποίηση F(A, B, C) = Σm(1, 3, 5, 6) με MUX.



Παράδειγμα (7)

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 =
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	



Παράδειγμα με πολλαπλές εισόδους: Gray σε binary

- Σχεδιάστε το κύκλωμα που μετατρέπει από 3-bit Gray στο δυαδικό κώδικα.
- Ο πίνακας αληθείας δίνεται στα δεξιά.
- Είναι φανερό ότι, X = C ενώ οι συναρτήσεις Y και Z είναι πιο πολύπλοκες.

Gray A B C	Binary x y z
0 0 0	0 0 0
1 0 0	0 0 1
1 1 0	0 1 0
0 1 0	0 1 1
0 1 1	1 0 0
1 1 1	1 0 1
1 0 1	1 1 0
0 0 1	1 1 1

Gray σε binary (1η Λύση – χρήση ROM) (1)

- Αναδιατάξτε τον πίνακα, έτσι ώστε οι διάφοροι συνδυασμοί εισόδων να είναι σε σειρά (000, 001, …,111).
- Οι συναρτήσεις y και z να υλοποιηθούν με ένα διπλό (2-bit) 8-σε-1 MUX:
  - Οι A,B και C ενώνονται στις εισόδους επιλογής.
  - Οι έξοδοι του MUX ορίζονται ως η y και η z.
  - Οι είσοδοι δεδομένων παίρνουν τις αντίστοιχες σταθερές τιμές από τον πίνακα αληθείας(value fixing).

Gray A B C	Binary x y z
0 0 0	0 0 0
0 0 1	1 1 1
0 1 0	0 1 1
0 1 1	1 0 0
1 0 0	0 0 1
1 0 1	1 1 0
1 1 0	0 1 0
1 1 1	1 0 1

Gray σε binary (1η Λύση – χρήση ROM) (2)

Υπολογισμός Z

- Για ABC = 000 (δηλαδή η D0) έχουμε z = 0.

→Άρα για D0 θα τοποθετήσουμε την τιμή 0.

→Ομοίως D1 = 1 , D2 = 1.

Υπολογισμός y

Για ABC = 000 (δηλαδή η D0) έχουμε y = 0.

→ Άρα για D0 θα τοποθετήσουμε την τιμή 0.

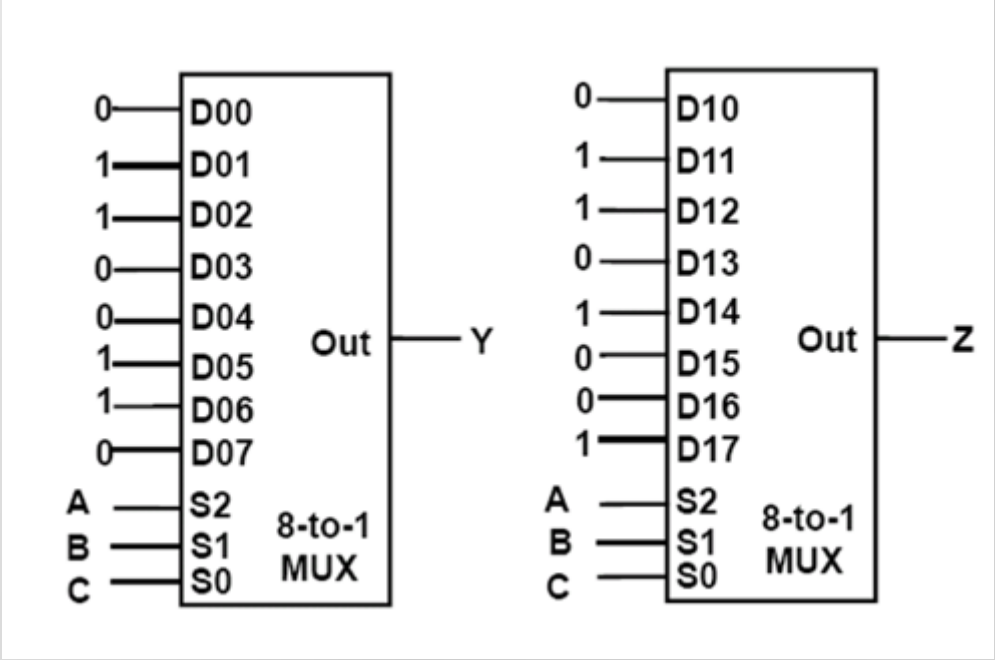
→Ομοίως D1 = 1 , D2 = 1

→…, D7 = 1

Για να μη μπερδεύονται οι είσοδοι θεωρούμε ότι ο υπολογισμός του y χρησιμοποιεί αντί για D0-D7 , D10-D17.

Gray A B C	Binary x y z
0 0 0	0 0 0
0 0 1	1 1 1
0 1 0	0 1 1
0 1 1	1 0 0
1 0 0	0 0 1
1 0 1	1 1 0
1 1 0	0 1 0
1 1 1	1 0 1

Gray σε δυαδικό (1η Λύση – χρήση ROM) (3)



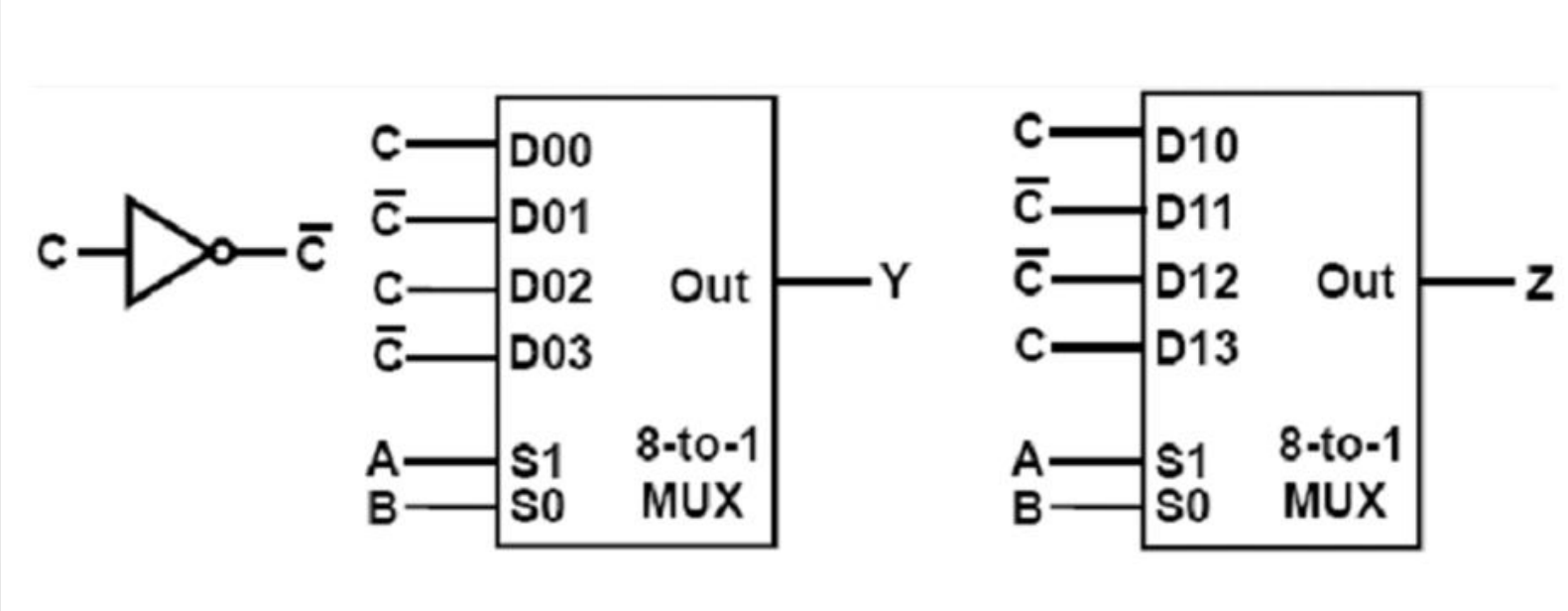
- Βασικά, ένας 2-bit 8-to-1 MUX με σταθερές τιμές είναι πανομοιότυπος με μια ROM με διευθύνσεις $3^{ων}$ –bit (είσοδοι) και δεδομένα εξόδου 2-bit ! → 2^2 x 2 ROM.

Gray σε δυαδικό (2η Λύση – χρήση MUX) (1)

- Αναδιατάξτε τον πίνακα, έτσι ώστε οι διάφοροι συνδυασμοί εισόδων να είναι σε σειρά (000, 001, …, 111).

Gray A B C	Binary x y z	Στοιχειώδης συνάρτηση του C για y	Στοιχειώδης συνάρτηση του C για z
0 0 0	0 0 0	F = C	F = C
0 0 1	1 1 1		
0 1 0	0 1 1	F =	F =
0 1 1	1 0 0		
1 0 0	0 0 1	F = C	F =
1 0 1	1 1 0		
1 1 0	0 1 0	F =	F = C
1 1 1	1 0 1		

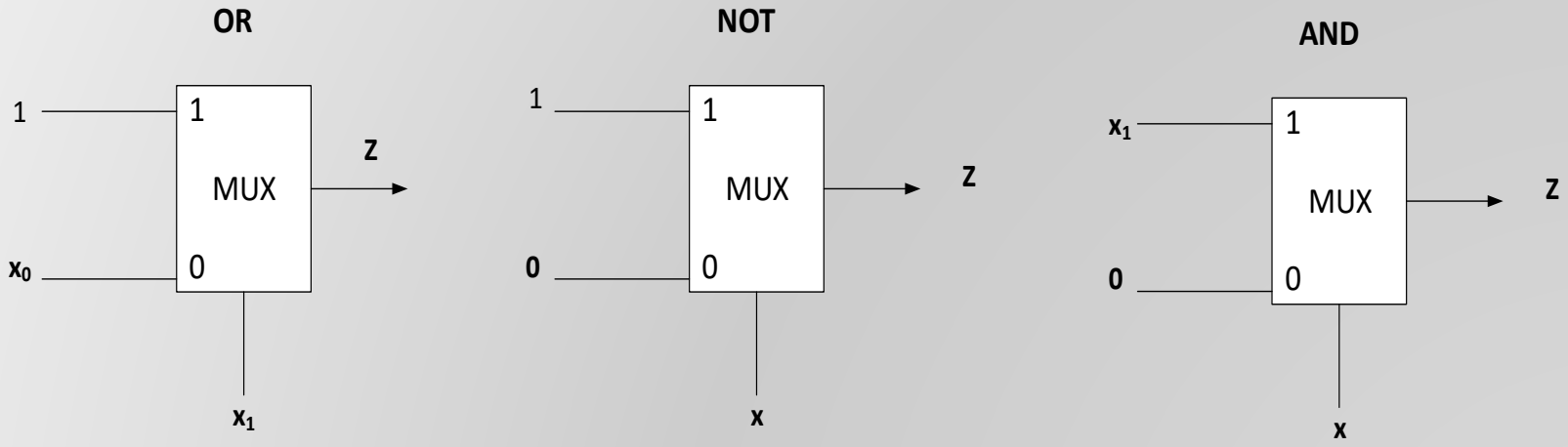
Gray σε δυαδικό (2η Λύση – χρήση MUX) (2)



- Η 2η λύση μειώνει το κόστος σχεδόν στο μισό της 1ης .
- Η 2η λύση δεν μοιάζει με ROM.

MUX ως οικουμενική πύλη

- Μπορούμε να παράγουμε τις λειτουργίες OR, AND και NOT μόνο με 2-σε-1 MUX. Άρα η 2-to-1 MUX είναι οικουμενική πύλη.



$z = x_1 + x_1 x_0$

$= x_1 x_0' + x_1 x_0 + x_1' x_0$

$z = 0x + 1x' = x'$

$z = x_1 x_0 + 0x_0' = x_1 x_0$

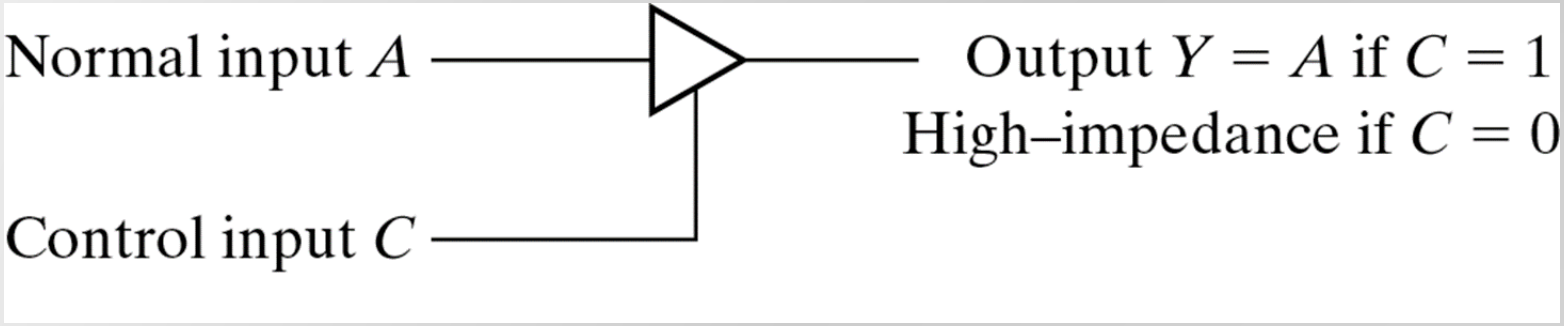
Αποπολυπλέκτες (Demultiplexers - DeMUX)

- Εκτελεί το αντίστροφο της λειτουργίας του πολυπλέκτη:
  - Δέχεται δεδομένα από μια είσοδο και τα μεταβιβάζει σε συγκεκριμένη έξοδο, από τις 2^n πιθανές που υπάρχουν.
  - Η επιλογή εξόδου γίνεται από τις n εισόδους επιλογής.
  - Βασικά, είναι ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΕΣ! Για παράδειγμα, ένας 2-σε-4 DeMUX είναι ένας αποκωδικοποιητής 2-σε-4, με είσοδο ενεργοποίησης (ενώνεται στην είσοδο δεδομένων).

Πύλη 3 καταστάσεων (1)

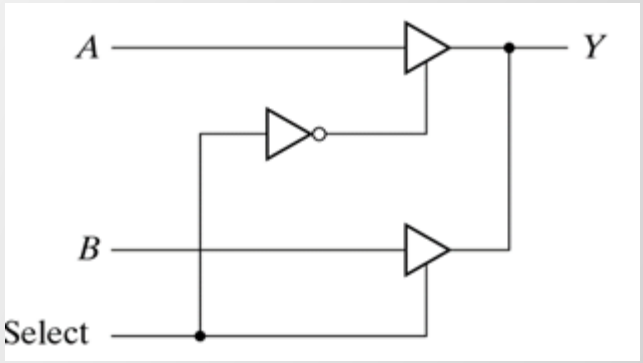
- 3 ευσταθείς καταστάσεις εξόδου:
  - λογικό 0
  - λογικό 1
  - υψηλής αντίστασης εξόδου (αποσύνδεση από το κύκλωμα).
- Χρησιμοποιείται συνήθως ως buffer.
- Επιτρέπεται η γαλβανική σύνδεση εξόδων πολλών πυλών $3^{ων}$ καταστάσεων.

Πύλη 3 καταστάσεων (2)

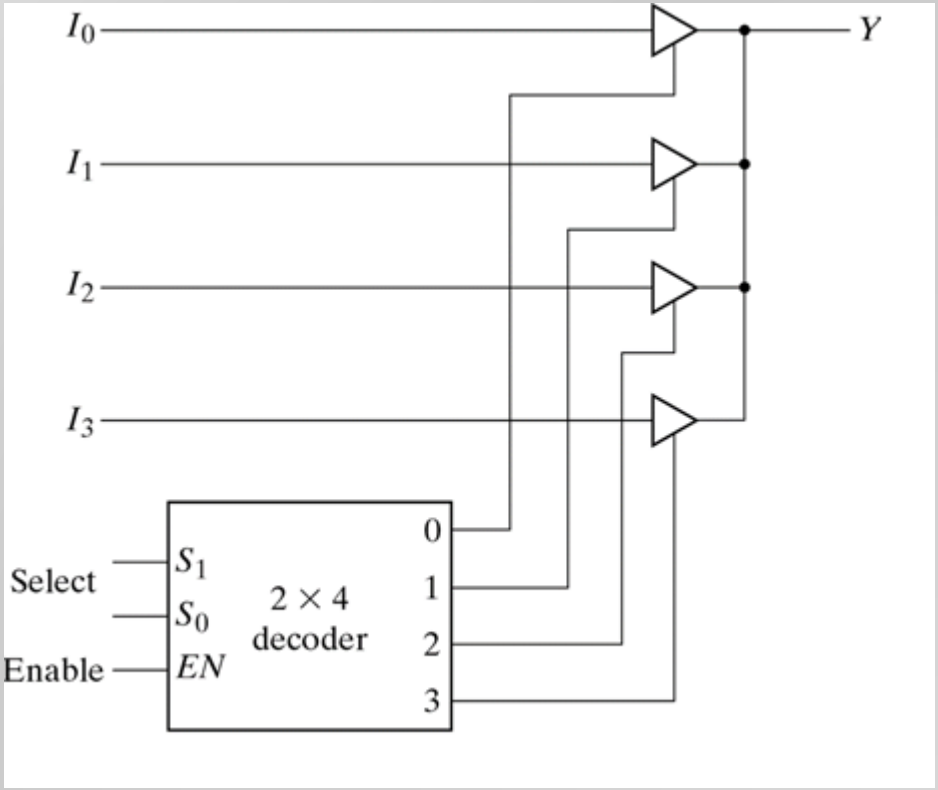


Normal input A: Κανονική είσοδος A
Control input C: Ελεγχόμενη είσοδος C
Output Y = A if C=1: Έξοδος Y = A, αν C = 1
High-impedance if C =0: Υψηλής αντίστασης, αν C = 0

Υλοποίηση Πολυπλεκτών με 3state gates (τριών καταστάσεων πυλών)

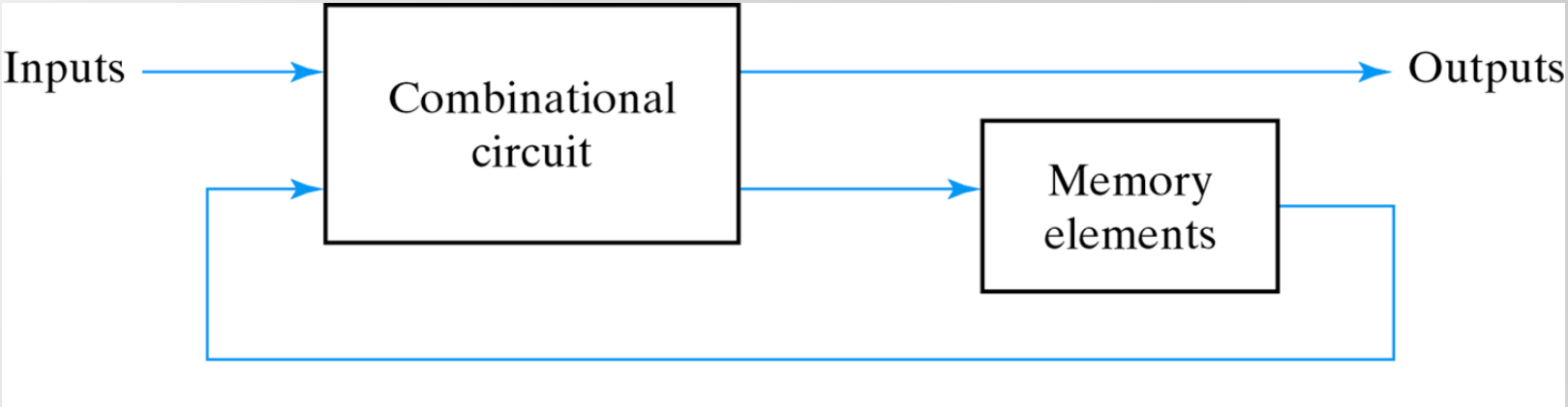


(α) 2-σε-1 γραμμή πολυπλέκτη



(b) 4-σε-1 γραμμή πολυπλέκτη

Τα ακολουθιακά κυκλώματα έχουν ανάδραση και στοιχεία μνήμης

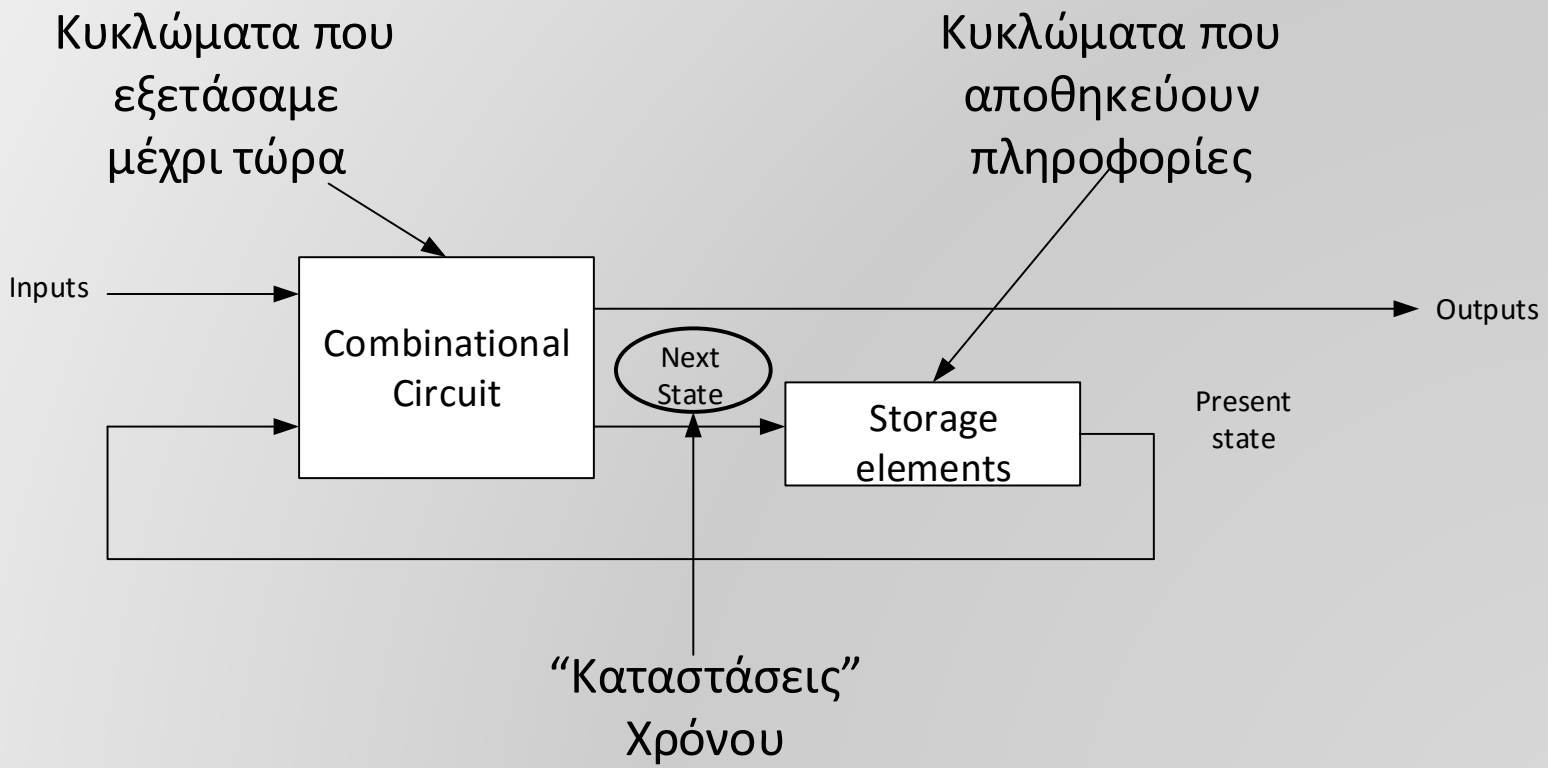


Inputs: Είσοδοι
Combinational circuit: Συνδυαστικό κύκλωμα
Memory elements: Στοιχεία μνήμης
Outputs: Έξοδοι

Ακολουθιακά Κυκλώματα (1)

- Ακολουθιακή Λογική:
  - Η τιμή σε μία έξοδο δεν εξαρτάται μόνο από τις τρέχουσες τιμές των εισόδων, αλλά και από τις προηγούμενες τιμές των εισόδων.
  - Αποθηκεύει πληροφορίες μεταξύ λειτουργιών (δεν απαιτεί διαδοχή).
  - Χρειάζεται κάποιου είδους μνήμη για να μπορεί να «θυμάται» τις προηγούμενες τιμές των εισόδων.

Ακολουθιακά Κυκλώματα (2)



Next state: Επόμενη κατάσταση
Storage elements: Στοιχεία αποθήκευσης
Present state: Παρούσα κατάσταση

Βασικές Έννοιες Ακολουθιακής Λογικής

- Τα κυκλώματα ακολουθιακής λογικής έχουν την ικανότητα να «θυμούνται» προηγούμενες καταστάσεις του κυκλώματος και προηγούμενες τιμές στις εισόδους.
- Έξοδοι του κυκλώματος μπορούν να χρησιμοποιηθούν ως νέες τιμές εισόδου στο κύκλωμα (κυκλώματα ανάδρασης = feedback circuits).
- Τα στοιχεία αποθήκευσης είναι κυκλώματα που μπορούν να αποθηκεύουν δυαδική πληροφορία: μνήμη.

Ακολουθιακά Κυκλώματα:Σύγχρονα vs Ασύγχρονα

Υπάρχουν 2 τύποι ακολουθιακών κυκλωμάτων:

- Σύγχρονο (latch mode) ακολουθιακό κύκλωμα:
  - Η συμπεριφορά του ορίζεται βάσει των τιμών στις εξόδους και στα στοιχεία μνήμης, σε διακριτές στιγμές του χρόνου.
  - Αυτού του είδους τα κυκλώματα πετυχαίνουν συγχρονισμό χρησιμοποιώντας ένα σήμα χρονισμού, το γνωστό ως ρολόι.
- Ασύγχρονο (fundamental mode) ακολουθιακό κύκλωμα:
  - Η συμπεριφορά του ορίζεται από την σειρά των αλλαγών των τιμών στις εισόδους σε συνεχή χρόνο.
  - Οι τιμές των εξόδων μπορούν να αλλάξουν ανά πάσα στιγμή, χωρίς κανένα συγκεκριμένο συγχρονισμό (μπορούν να γίνουν ασταθή).

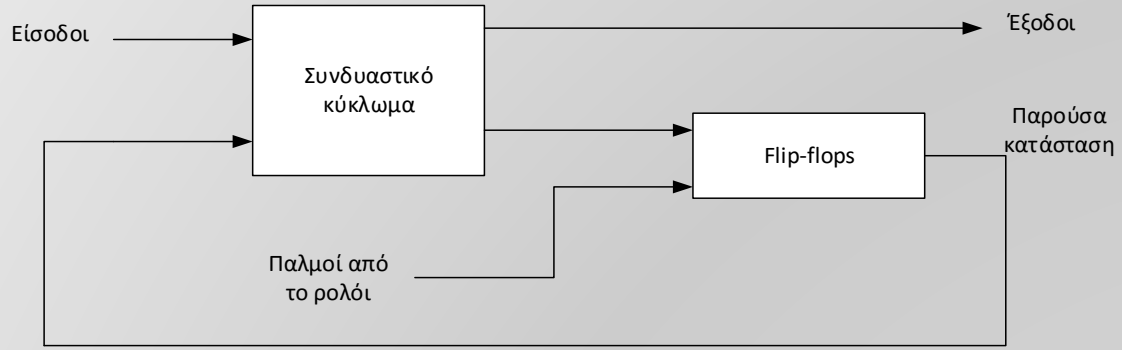
Μνήμη…

- Τα στοιχεία μνήμης που χρησιμοποιούνται ονομάζονται

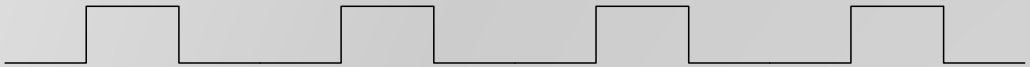
flip-flop

- Αποθηκεύουν 1 bit πληροφορίας.
- Η κατάσταση των flip-flop αλλάζει μόνο κατά την αλλαγή επιπέδου παλμού ρολογιού.

Σύγχρονο Ακολουθιακό Κύκλωμα με ρολόι



(α) Σχηματικό διάγραμμα



(β) Χρονικό διάγραμμα από τους παλμούς του ρολογιού

- Τα flip-flops έχουν ως εισόδους σήματα από το συνδυαστικό κομμάτι του κυκλώματος καθώς και σήμα από ένα ρολόι με περιοδικούς παλμούς μεταξύ αμετάβλητων περιοδικών διαστημάτων.

Τέλος Ενότητας