Ψηφιακή Σχεδίαση

Ενότητα 6: Δυαδικές Πράξεις, Συμπλήρωμα του 2, Δυαδικοί Αποκωδικοποιητές, Κωδικοποιητές, Πολυπλέκτες

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

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

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.

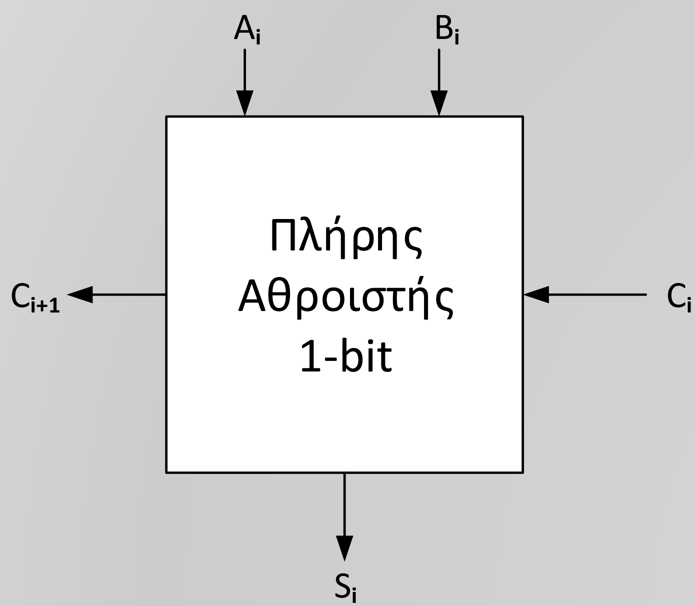
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και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

Σκοπός της ενότητας

- Να γίνει ανάλυση των δυαδικών πράξεων και τα συμπληρώματα του 2.
- Να γίνει ανάλυση των κωδικοποιητών , κωδικοποιητών προτεραιότητας, πολυπλέκτες.
- Να γίνει υλοποίηση συνάρτησης με πολυπλέκτη, αποπλέκτες, πύλη 3$^{\text{ών}}$ καταστάσεων.
- Εισαγωγή στα ακολουθιακά.

Πλήρης Αθροιστής 1-bit (Full Adder)

- Συνδυαστικό κύκλωμα που διεκπεραιώνει την προσθεση μεταξύ 3^ων^ bits (2 bits προσθετέων και 1 bit για κρατούμενο εισόδου - carry-in).



Δεκαδική Πρόσθεση

- Σχεδιάστε ένα κύκλωμα για την εκτέλεση πρόσθεσης, αφαίρεσης, ...
- Είσοδος σε κωδικοποιημένη δεκαδική μορφή, π.χ. BCD
- Δεκαδικός Αθροιστής BCD:
  – 8 είσοδοι (4 bits για τον κάθε δεκαδικό αριθμό).
  – 5 έξοδοι για το κάθε δεκαδικό άθροισμα και το κρατούμενο.
  – Θυμηθείτε τον κανόνα για BCD πρόσθεση:

  Προσθέτουμε το 0110 στο αθροισμα αν αυτό είναι μεγαλύτερο του 1010, για να διορθώσουμε την τιμή του κρατουμένου.

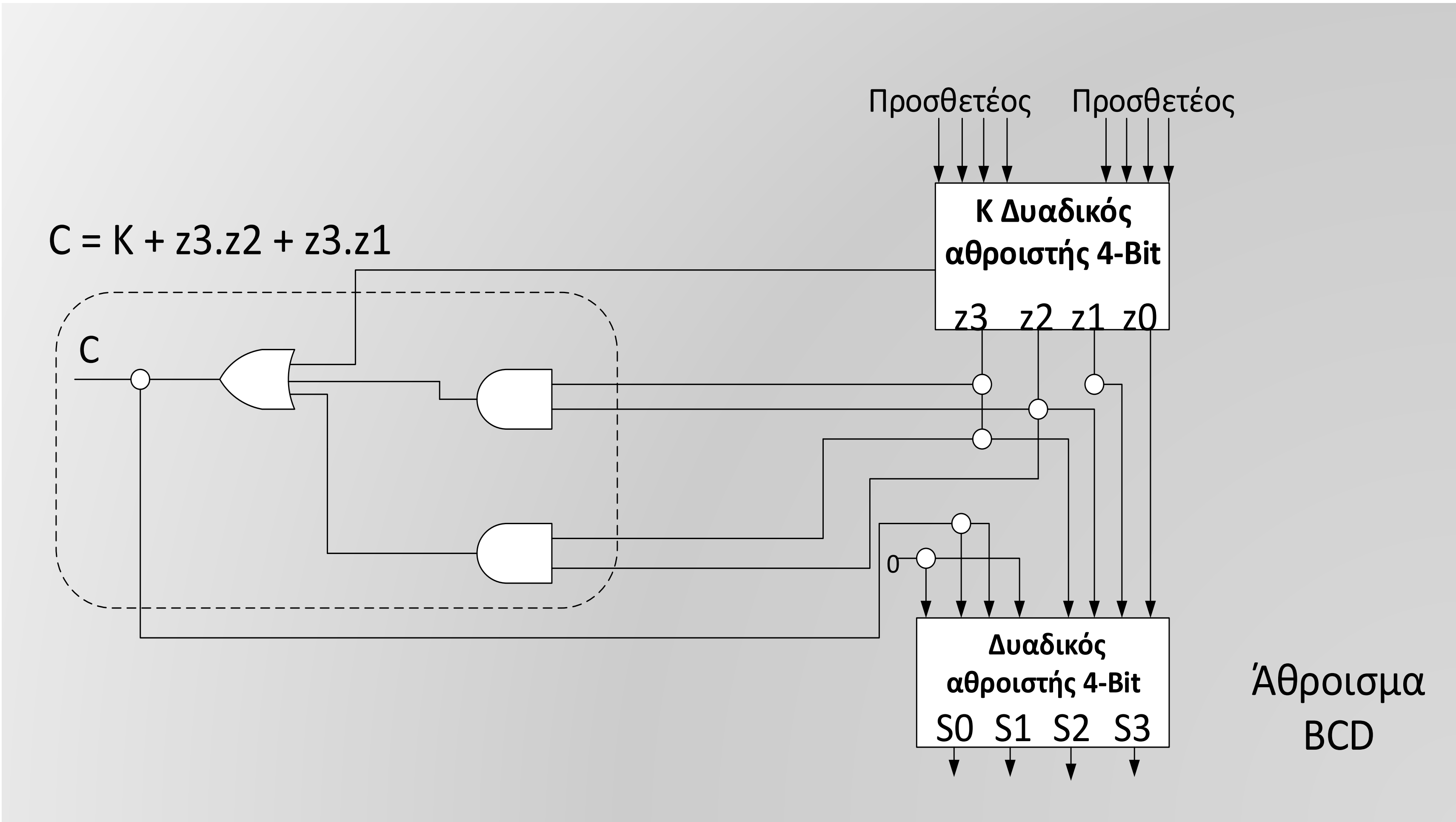
Αθροιστής BCD (1)

- Το αποτέλεσμα κυμαίνεται από 0 έως 19 (9 + 9 + 1).
- Χρησιμοποιούμε ένα δυαδικό αθροιστή των 4bit.
- Ο αθροιστής σχηματίζει το άθροισμα στο δυαδικό σύστημα.
- Το δυαδικό σύστημα θα πρέπει να μετατραπεί στο BCD. Από 0000 έως 1001 δεν απαιτείται μετατροπή. Από 1010 έως 0011 απαιτείται μετατροπή (προσθέτοντας το 6).

Αθροιστής BCD (2)

- Πως εντοπίζεται η ανάγκη διόρθωσης του αποτελέσματος;
  - Αν υπάρχει κρατούμενο Κ.
  - Αν $Z_8 Z_4 = 1$.
  - Αν $Z_8 Z_2 = 1$.

Αθροιστής BCD (3)



Δυαδική Αφαίρεση (1)

- Μη προσημασμένοι αριθμοί (Unsigned numbers) το πρόσημο δεν αναπαριστάται ρητά (εννοείται).
- Δεδομένων των δυαδικών αριθμών M και N, βρείτε M-N:
  - Περίπτωση I: M ≥ N, άρα το MSB του Borrow είναι το 0.

B	0 0 0 1 1 0				
M	1 1 1 1 0		30		
N	-1 0 0 1 1	→	-19	→	Το αποτέλεσμα είναι ορθό.
Diff	0 1 0 1 1		11		

  - Περίπτωση II: N > M, άρα το MSB του Borrow είναι το 1.

B	1 1 1 0 0 0				
M	1 0 0 1 1		19		
N	-1 1 1 1 0	→	-30	→	Το αποτέλεσμα είναι ορθό.
Diff	1 0 1 0 1		21		

Δυαδική Αφαίρεση (2)

- Γενικά, εάν N > M, Dif = M - N + 2^n, όπου το n = # bits.
- Στην περίπτωση II του προηγούμενου παραδείγματος, Dif = 19 - 30 + 2^5 = 21.
- Για να διορθωθεί η απόλυτη τιμή (magnitude) του Dif, που έπρεπε να ήταν N - M, υπολογίζεται το 2^n- (M - N + 2^n).
- Αυτό είναι γνωστό ως το συμπλήρωμα του 2 (2’s complement) του Dif.

Διαδικασία

- Για την αφαίρεση 2 n-bit αριθμών, M – N, στην βάση του 2:
  - Βρείτε M – N.
  - Εάν το MSB του Borrow είναι 0, τοτε M ≥ N. Το αποτέλεσμα είναι θετικό και ορθό.
  - Εάν το MSB του Borrow είναι 1, τοτε N > M. Το αποτέλεσμα είναι αρνητικό και ο βαθμός του πρέπει να διορθωθεί με την αφαίρεση του από το 2^n (βρείτε το συμπλήρωμα του 2).

Παράδειγμα

- M = 01100100 και N = 10010110, βρείτε M - N

B	1 0 0 1 1 1 1 0 0	
M	0 1 1 0 0 1 0 0	100
N	-1 0 0 1 0 1 1 0	-150
Diff	1 1 0 0 1 1 1 0	206

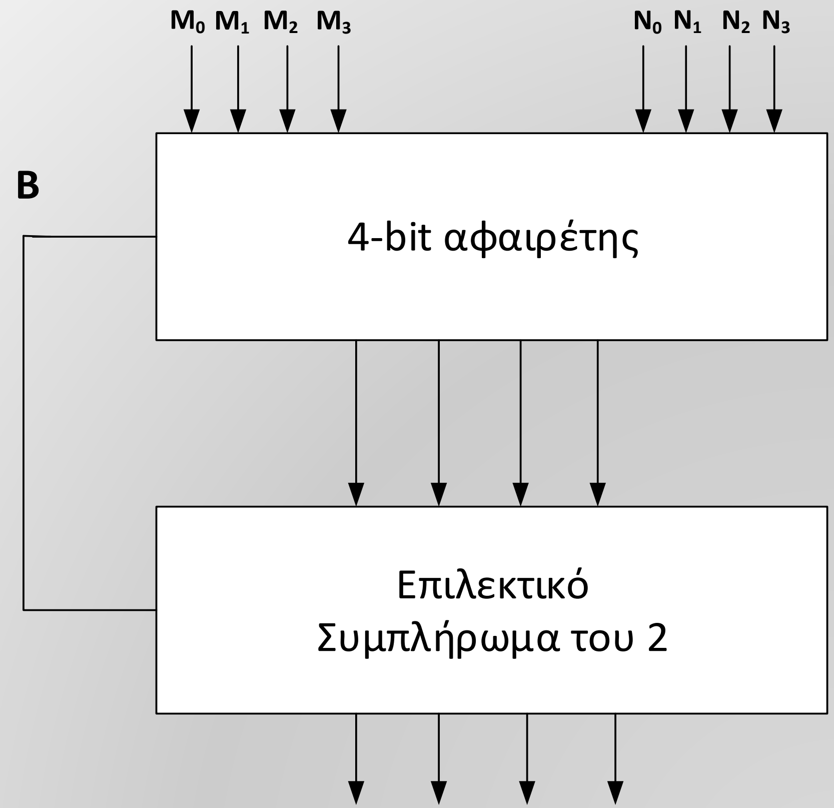
Το αποτέλεσμα είναι αρνητικό (αφού MSB(B) = 1) και Διορθώνεται με την αφαίρεση από το 2^n.

2^n	1 0 0 0 0 0 0 0 0	256
Dif	- 1 1 0 0 1 1 1 0	-206
	0 0 0 1 1 0 0 1 0	50

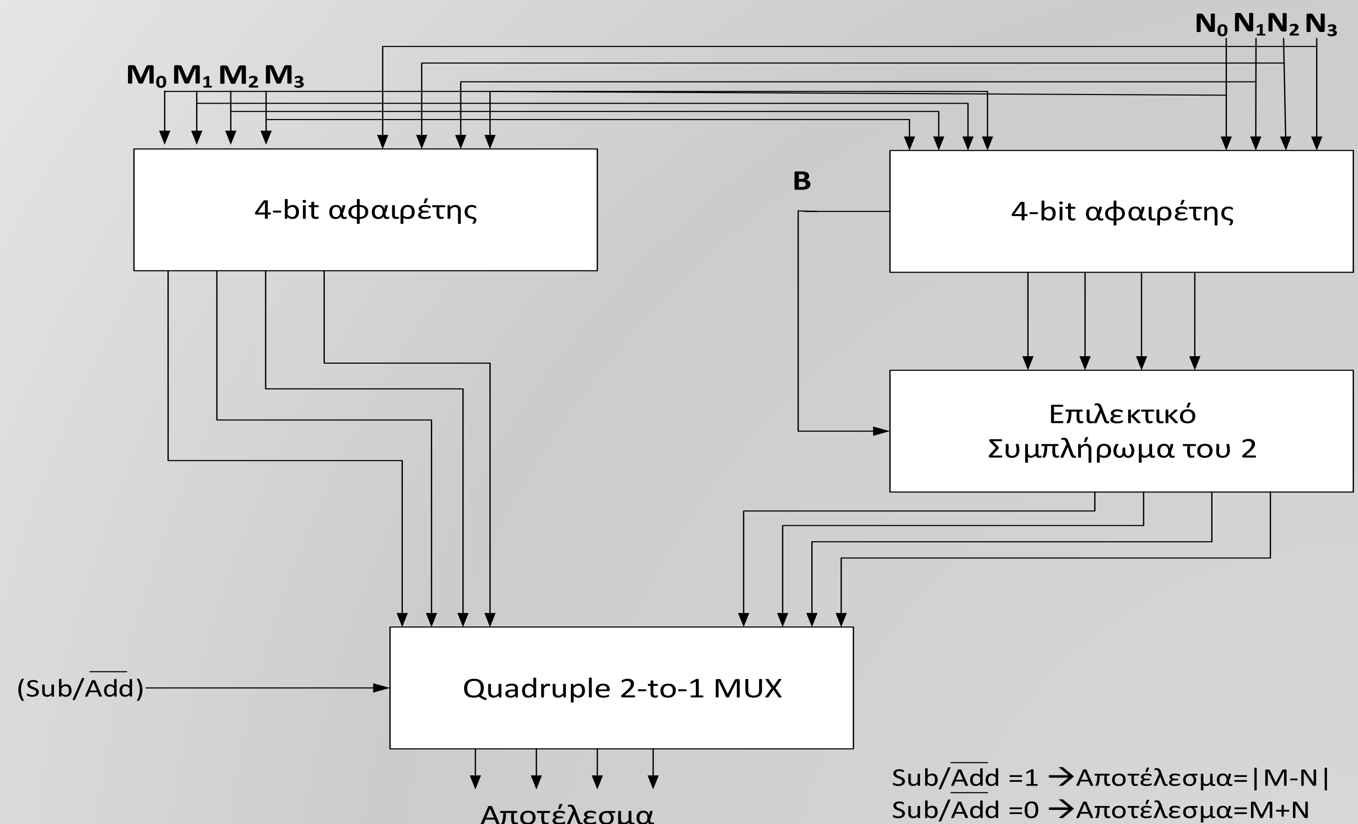
Ένα απλό διάγραμμα Αφαιρέτη

Διάγραμμα Αφαιρέτη →

- Το επιλεκτικό συμπλήρωμα του 2, ενεργοποιείται όταν B = 1; αλλιώς, το αποτέλεσμα από τον αφαιρέτη περνά.
- Δεν είναι ο καλύτερος τρόπος υλοποίησης κυκλώματος αφαιρέτη!



Διάγραμμα Δυαδικού Αθροιστή - Αφαιρέτη



Quadruple 2-to-1 MUX: Τετραπλό 2-σε-1 MUX

Sub: Αφαίρεση

Add: Πρόσθεση

Συμπλήρωμα του 2 (1)

- Για ένα θετικό δυαδικό αριθμό με n ψηφία N_2, το συμπλήρωμα του 2 , $2C(N_2)$, δίνεται από:
  - $2C(N_2) = \quad 2^n - N_2$, εάν $n > 0$
    0 , εάν $n = 0$

- Παράδειγμα 1:
  - $2C(N_2) = 2^4 - N_2 = 10000_2 - 1010_2 = 0110_2$
- Παράδειγμα 2:
  - $2C(N_2) = 2^5 - N_2 = 100000_2 - 11111_2 = 00001_2$

Συμπλήρωμα του 2 (2)

- Ένας πιο εύκολος τρόπος για να υπολογίσουμε το συμπλήρωμα του 2:

1. Αφήστε τα least significant 0 και πρώτο 1 χωρίς αλλαγές.
2. Αντικαταστήστε 0 με 1 και 1 με 0 στα υπόλοιπα higher significant bits.

Παραδείγματα:

Συμπλήρωμα χωρίς αλλαγές

N = 1010

0110

Συμπλήρωμα του 2

Συμπλήρωμα χωρίς αλλαγές

N = 01011000

10101000

Συμπλήρωμα του 2

Αφαίρεση με συμπληρώματα

- Για να βρούμε το M - N = M + (-N), μπορούμε να χρησιμοποιήσουμε μια συμπληρωματική μορφή για την αναπαράσταση ενός αρνητικού αριθμού –N, και να κάνουμε μια «απλή πρόσθεση».
- Πρέπει να μπορούμε να «μετατρέψουμε» το αποτέλεσμα.

Αφαίρεση με συμπλήρωμα του 2

- Εάν χρησιμοποιούμε συμλήρωμα του 2 για την αναπαράσταση αρνητικών αριθμών:

1. $R_I = M + 2C(N_2) = M + (2^n - N) = M - N + 2^n$
2. Εάν υπάρχει ένα μη-μηδενικό carry out στην πρόσθεση, τότε $M \geq N \rightarrow$ το carry out αγνοείται και τα υπόλοιπα ψηφία είναι ίσα με $R = M - N$.
3. Εάν $M < N$, τότε υπολογίζουμε το συμπλήρωμα του 2 του R_I ($= 2^n - R_I = 2^n - (M - N + 2^n) = N - M$) και προσθέτουμε ένα αρνητικό πρόσημο στην αρχή του αριθμού.

Δηλ. το αποτέλεσμα του R είναι $-2C([R_I]_2) = -(N - M)$.

Παράδειγμα (1)

- A = 1010100 (84_{10}), B= 1000011 (67_{10})
- Βρείτε R = A – B:
  - 2C(B) = 0111101 (61_{10})
  - A + 2C(B) = 1010100 + 0111101 = 10010001
  - Το carry out απορρίπτεται, R = 0010001 (17_{10})
- Βρείτε R = B – A:
  - 2C(A) = 0101100 (44_{10})
  - B+2C(A) = 1000011 + 0101100 = 01101111
  - R = -2C(B + 2C(A)) = -0010001 (-17_{10})

  (το bit του carry (κρατουμένου) δεν υπολογίζεται).

Δυαδικοί Αθροιστές – Αφαιρέτες (1)

- Εάν εκτελέσουμε αφαίρεση χρησιμοποιώντας συμπληρώματα, εξαλείφουμε την πράξη της αφαίρεσης, και επομένως μπορούμε να χρησιμοποιήσουμε έναν αθροιστή, με κατάλληλο κύκλωμα για συμπλήρωμα.
- Στην ακρίβεια, μπορούμε να χρησιμοποιήσουμε έναν αθροιστή για πρόσθεση <u>και</u> αφαίρεση:
  - Συμπλήρωμα αφαιρετέου για αφαίρεση.
  - Μη συμπλήρωση αφαιρετέου για πρόσθεση.
- Για να υλοποιήσουμε ένα κύκλωμα πρόσθεσης /αφαίρεσης, χρειαζόμαστε ένα αθροιστή (adder) και ένα κύκλωμα που να επιλέγει μεταξύ συμπληρώματος ή μη (selective complementer – επιλεκτικός συμπληρωτής).

Δυαδικοί Αθροιστές – Αφαιρέτες (2)

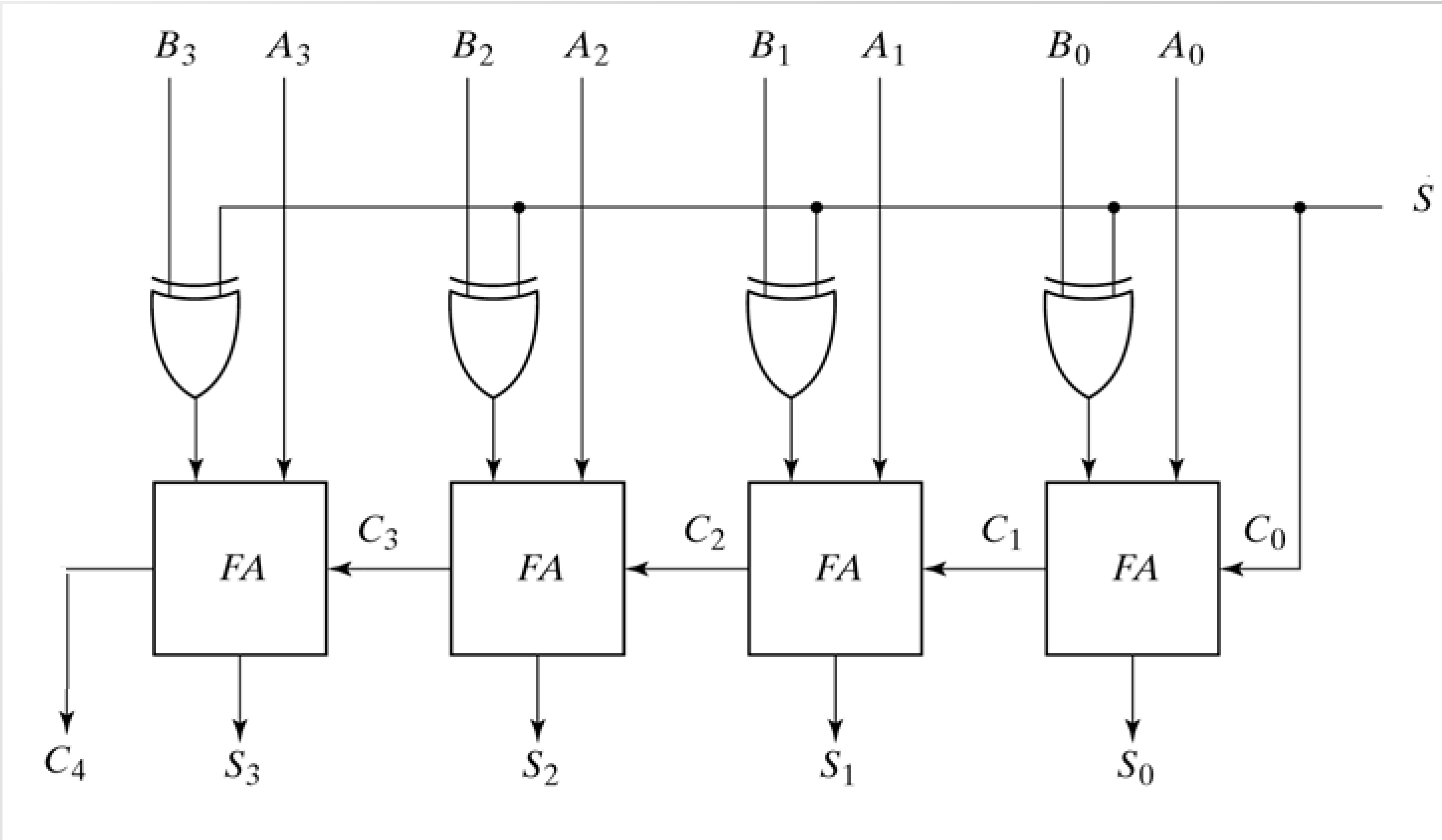
- Η αφαίρεση A – B μπορεί να γίνει υπολογίζοντας το συμπλήρωμα του 2 του B και προσθέτοντας το αποτέλεσμα στον A.
- Το συμπλήρωμα του 2 του B υπολογίζεται με

(i) Την συμπλήρωση του B και

(ii) προσθέτοντας 1 στο αποτέλεσμα του (i)

$$
\begin{aligned}
A - B &= A + 2C(B) \\
&= A + 1C(B) + 1 \\
&= A + B' + 1
\end{aligned}
$$

Δυαδικοί αθροιστές-αφαιρέτες 4ων bit (1)

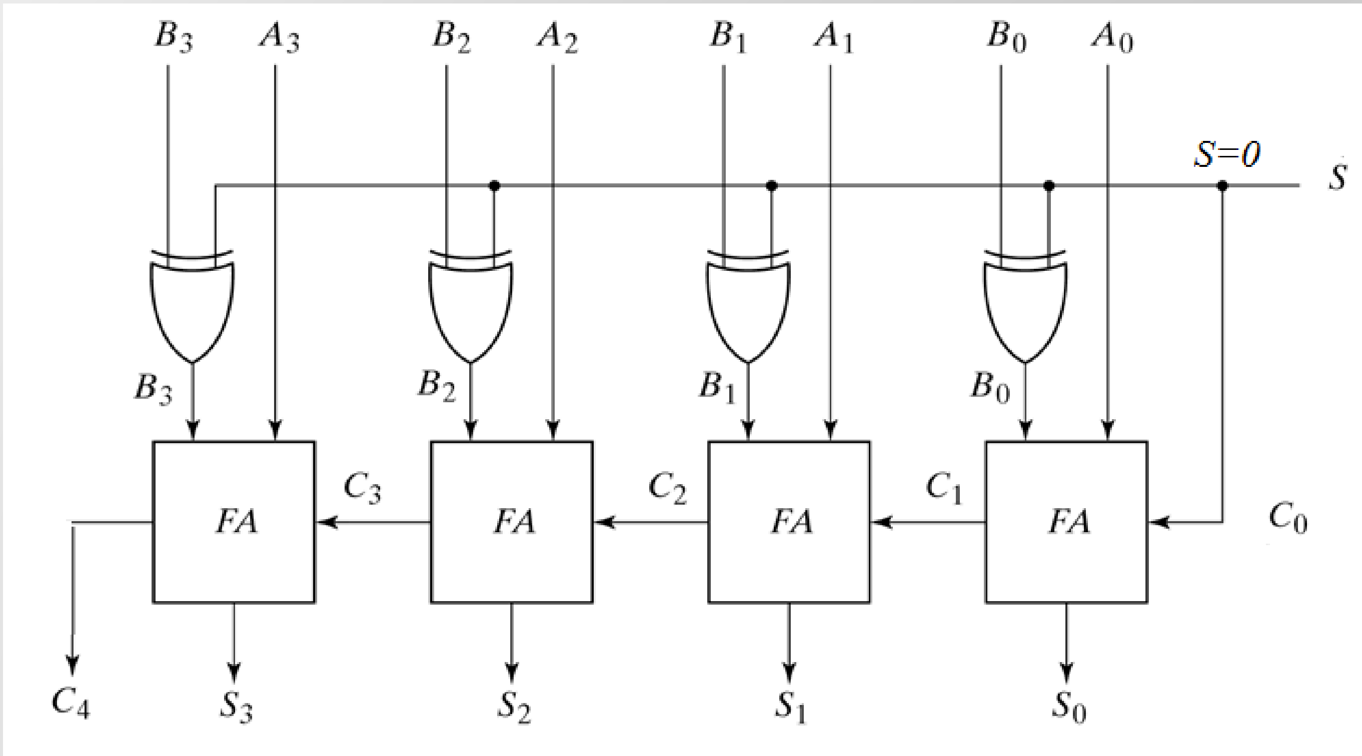


- Οι πύλες XOR λειτουργούν ως προγραμματιζόμενοι αντιστροφείς.

Δυαδικοί αθροιστές-αφαιρέτες $4^{ων}$ bit (2)

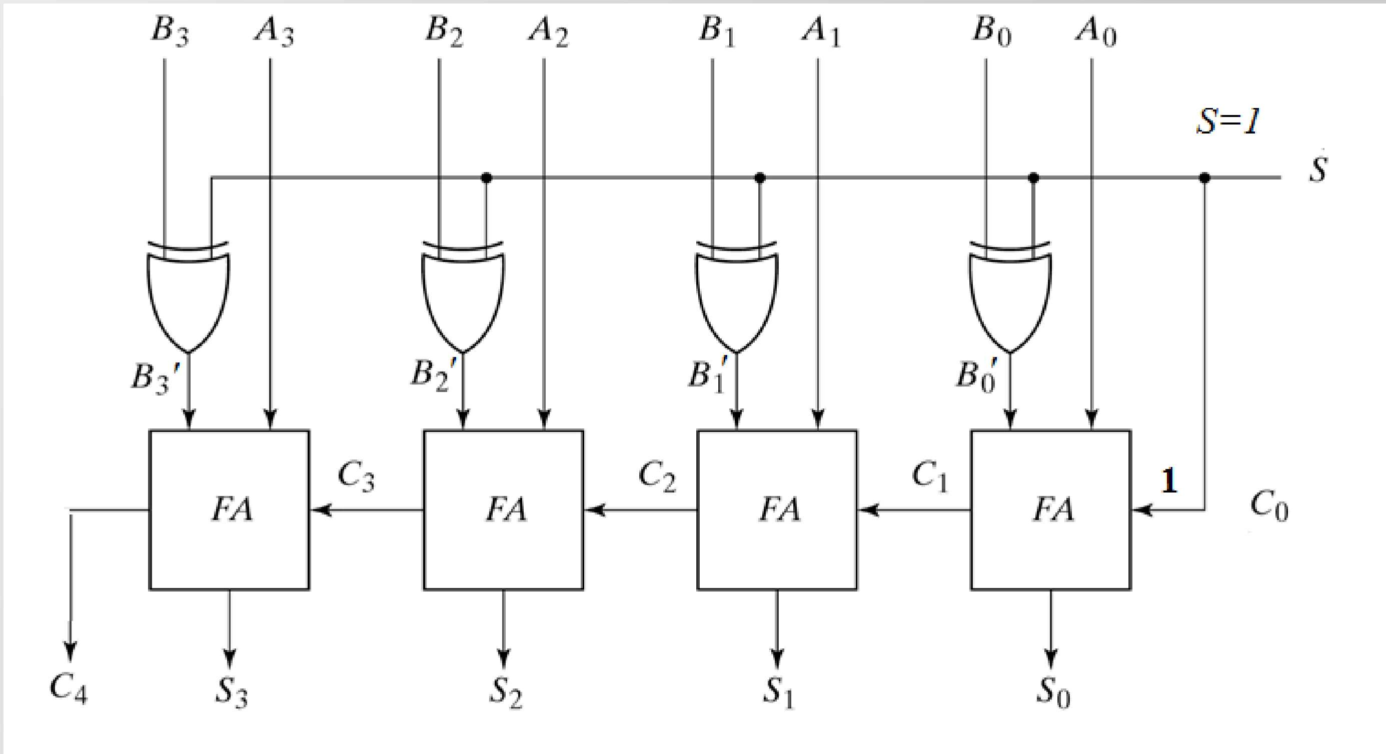
- Όταν S = 0, το κύκλωμα εκτελεί A + B, αφού το carry in στο LSB είναι 0 και οι έξοδοι των πυλών XOR δίνουν $B \oplus 0 = B$.
- Όταν S = 1, το κύκλωμα εκτελεί $A + B' + 1 = A - B$, αφού το carry στο LSB είναι 1 και οι έξοδοι των πυλών XOR δίνουν $B \oplus 1 = B'$. Άρα το κύκλωμα προσθέτει στον A το συμπλήρωμα του 1 του B <u>συν</u> 1 (από το carry στο LSB).

Δυαδικοί αθροιστές-αφαιρέτες $4^{ων}$ bit (3)



- Όταν S = 0, επιλέγει πρόσθεση.

Δυαδικοί αθροιστές-αφαιρέτες $4^{ων}$ bit (4)



- Όταν S = 1, επιλέγει αφαίρεση.

Δυαδικοί αθροιστές-αφαιρέτες 4^{ων} bit (5)

- Όταν $C_4 = 0$ και $S = 1$, τότε $A < B$ και πρέπει να διορθωθεί το αποτέλεσμα $R_3 \ldots R_0$.
- Άρα, πρέπει να υπολογιστεί το συμπλήρωμα του 2 του $R_3 \ldots R_0$:
  – Χρησιμοποιείται ένα ειδικό κύκλωμα για το συμπλήρωμα του 2 ή
  – Χρησιμοποιείται ο αθροιστής/αφαιρέτης ξανά, με $A_3 \ldots A_0 = 0000$, $B_3 \ldots B_0 = R_3 \ldots R_0$ και $S = 1$.

Προσημασμένοι Δυαδικοί Αριθμοί (1)

- Σύστημα Προσημασμένης- Απόλυτης Τιμής (Signed- Magnitude System):
  - Οι προσημασμένοι αριθμοί αναπαριστούνται χρησιμοποιώντας το MSB του δυαδικού αριθμού για τον καθορισμό του προσήμου του αριθμού:
    - Εάν MSB = 0 → θετικός αριθμός
    - Εάν MSB = 1 → αρνητικός αριθμός
  - Μην το συγχύσετε με **μη- προσημασμένους (unsigned)** αριθμούς!

Προσημασμένοι Δυαδικοί Αριθμοί (2)

- Για παράδειγμα:
  - -10_{10}
    - -1010_2 σε μη-προσημασμένο (το πρόσημο - δεν αποτελεί μέρος της δυαδικής τιμής).
    - 1010_2 σε προσημασμένο με signed-magnitude (το πρόσημο – αναπαριστάται με MSB = 1).
- Άλλο παράδειγμα:
  - 1011_2
    - 11_2 σε μη-προσημασμένο.
    - -3_{10} σε προσημασμένο με signed-magnitude.

Προσημασμένοι Δυαδικοί Αριθμοί (3)

- Για την υλοποίηση πρόσθεσης ή αφαίρεσης με signed- magnitude, χρειαζόμαστε:
  - Να ξεχωρίσουμε το **bit** του **προσήμου** από τα **magnitude bits**.
  - Να θεωήσουμε τα magnitude bits ως ένα μη-προσημασμένο αριθμό (η διόρθωση πρέπει να γίνεται όπου χρειάζεται).
- Για την αποφυγή της διόρθωσης, χρησιμοποιείται το σύστημα **Προσημασμένου-Συμπληρώματος (Signed-Complement)**.

Signed Complement(υπογεγραμμένο συμπλήρωμα) (1)

- Όταν διαβάζετε αριθμούς σε 2’s complement να θυμάστε ότι, όταν MSB = 1 ο αριθμός είναι αρνητικός και χρειάζεται να υπολογίσετε το 2’s complement της απόλυτης τιμής (magnitude).
- Παράδειγμα: Πιο είναι το δεκαδικό αντίστοιχο του 1001001_2;
  - Είναι αρνητικός αριθμός αφού οτ MSB = 1.
  - Magnitude – 001001 το συμπλήρωμα του 2 του magnitude = 110111.
  - Ο αριθμός είναι το -55_{10} .

Signed Complement (υπογεγραμμένο συμπλήρωμα) (2)

- Η αφαίρεση 2 προσημασμένων αριθμών, όπου οι αρνητικοί αριθμοί αναπαριστάνται σε signed 2's complement, παράγεται προσθέτοντας το 2's complement του αφαιρετέου με τον αφαιρέτη (συμπεριλαμβανόμενων των signed bits). Το carry out αγνοείται.
- Παραδείγματα: (5-bit αναπαραστάσεις)

01010	(+10)	01010	(+10)	10110	(-10)	10110	(-10)
00101	(-5)	-11011	-(-5)	-00101	-(+5)	-11011	-(-5)

01010	(+10)	01010	(+10)	10110	(-10)	10110	(-10)
+11011	+(-5)	+00101	+(+5)	+11011	+(-5)	+00101	+(+5)
00101	(+5)	01111	(+15)	10001	(-15)	11011	(-5)

Το πρόβλημα της υπερχείλισης

- Εάν η πρόσθεση των 2 n-bit αριθμών δίνει έναν αριθμό με n + 1 bots, τότε εμφανίζονται συνθήκες **υπερχείλισης.**
- Η εύρεση υπερχείλισης μπορέι να υλοποιηθέι είτε με υλικό (h / w) ή λογισμικό (s / w).
- Η εύρεση εξαρτάται από το αριθμιτικό σύστημα που χρησιμοποιείται: προσημασμένο ή μη-προσημασμένο.

Υπερχείλιση σε Μη προσημασμένους

- Πρόσθεση:
  - Όταν το Carry out == 1
- Αφαίρεση
  - Δεν μπορεί να γίνει ποτέ.

  Το magnitude (μέγεθος) του αποτελέσματος είναι πάντα ίσο ή μικρότερο από τον πιο μεγάλο των 2 αριθμών.
- → ΔΕΝ είναι πρόβλημα!

Υπερχείλιση σε προσημασμένους signed-2's complement

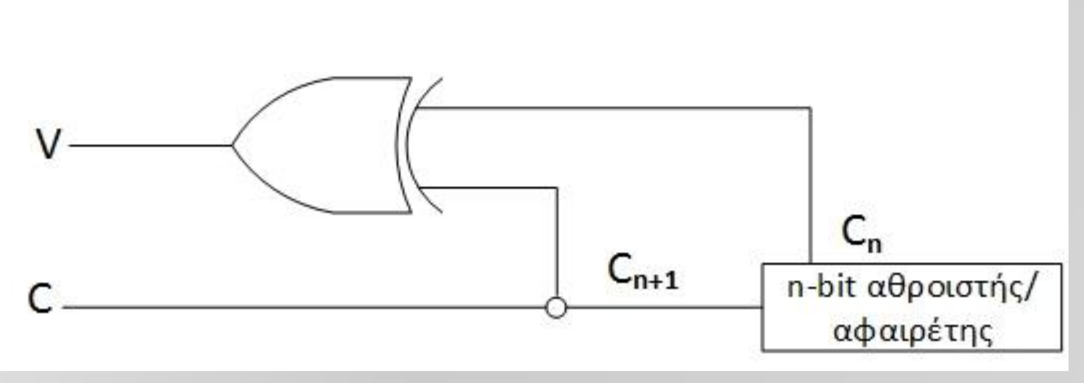
- Να θυμάστε ότι το MSB είναι το πρόσημο. Αλλά προστίθεται και το πρόσημο! Άρα, ένα carry out == 1 δεν σημαίνει πάντα υπερχείλιση!
- Υπερχείλιση παρατηρείται ΜΟΝΟ όταν και οι 2 αριθμοί έχουν το ίδιο πρόσημο. Αυτή η κατάσταση μπορεί να βρεθεί όταν το τελικό carry out (C_n) είναι <u>διαφορετικό</u> από το carry της προηγούμενης θέσης (C_{n-1}).

Παράδειγμα (2)

- Παράδειγμα 1: M = 65_{10} και N = 65_{10} σε ένα 8-bit σύστημα με signed 2's complement.
  - M = N = 01000001_2
  - M + N = 10000010 me C_n= 0. Αυτό είναι λάθος αφού δίνει αρνητικό αριθμό! Εάν το C_n οριστεί ως MSB, τότε έχουμε 010000010_2(130_{10}) που είναι ορθό, αλλά χρειάζεται 9-bits → υπερχείλιση.
- Παράδειγμα 2: M = -65_{10} και N = -65_{10} σε ένα 8-bit σύστημα με signed 2's complement.
  - M = N = 10111111_2
  - M + N = 01111110 me C_n= 1. Αυτό είναι πάλι λάθος αφού δίνει θετικό αριθμό! Εάν το C_n οριστεί ως MSB, τότε έχουμε 101111110_2(-130_{10}) που είναι ορθό, αλλά πάλι απαιτεί 9-bits → υπερχείλιση.

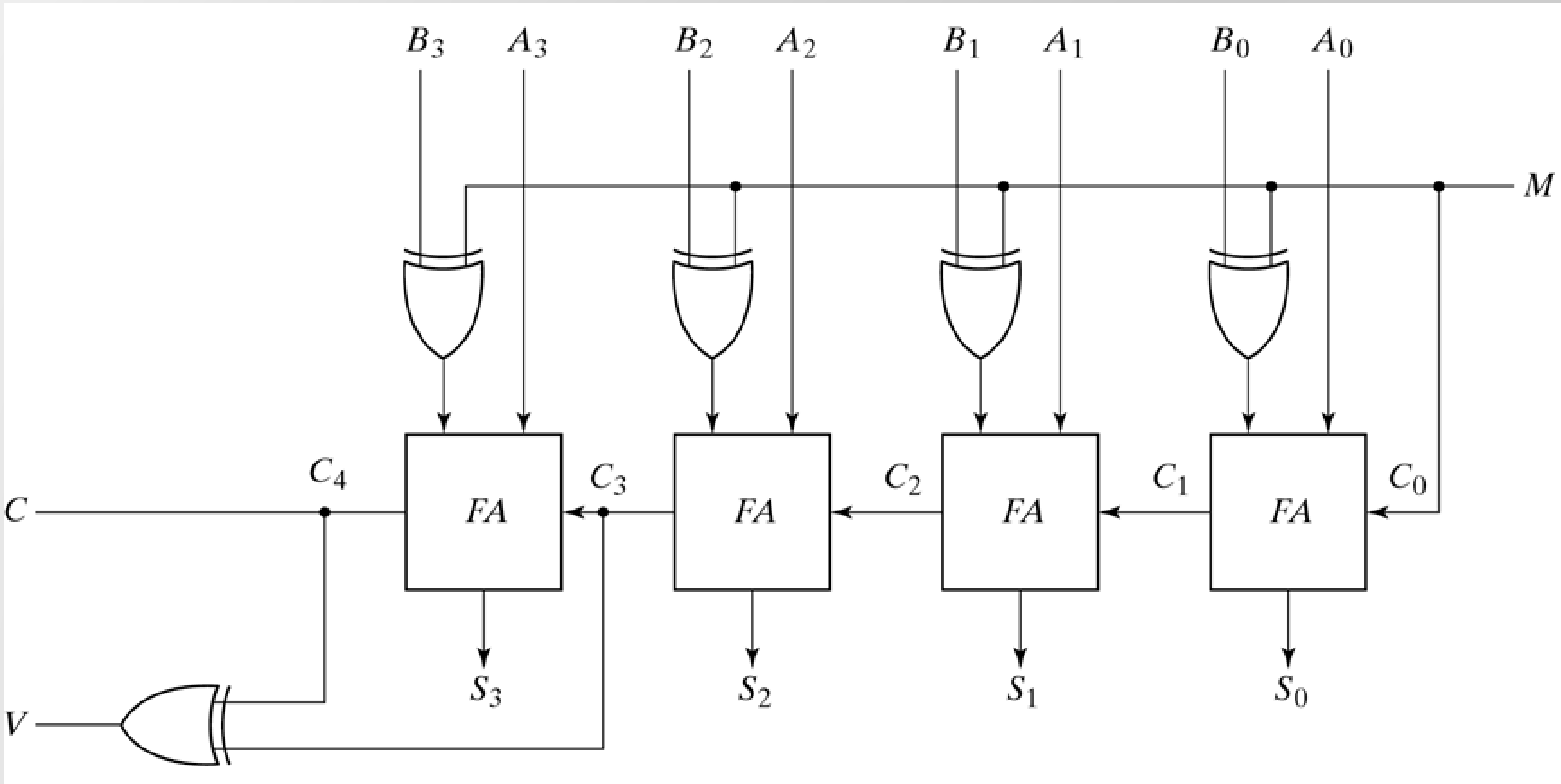
Εύρεση υπερχείλισης στο signed 2's complement

- Οι καταστάσει υπερχείλισης εντοπίζονται συγκρίνοντας τις τιμές στο carry out του sign bit (C_{n-1} και C_n).
  - Το C = 1 δείχνει υπερχείλιση όταν προσθέτουμε / αφαιρούμε unsigned αριθμούς.
  - Το V = 1 δείχνει υπερχείλιση όταν προσθέτουμε / αφαιρούμε αριθμούς σε signed = 2' s complement.



n-bit αθροιστής / αφαιρέτης με λογική εύρεσης υπερχείλισης.

Αθροιστής Αφαιρέτης 4bit με ανίχνευση υπερχείλισης



Δυαδικός Πολλαπλασιαστής (1)

- Ο δυαδικός πολ/σμός μοιάζει με τον δεκαδικό πολ/σμό:
  - Ο n-bit πολλαπλασιαστέος (multiplicand) πολ/ζεται με κάθε bit του m-bit πολλαπλασιαστή (multiplier), αρχίζοντας από το LSB, για την παρραγωγή n μερικών γινομένων.
  - Το κάθε διαδοχικό σύνολο των μερικών γινομένων μετατοπίζεται 1 bit προς τα αριστερά.
  - Το αποτέλεσμα παράγεται με την πρόσθεση των m γραμμών των μερικών γινομένων.

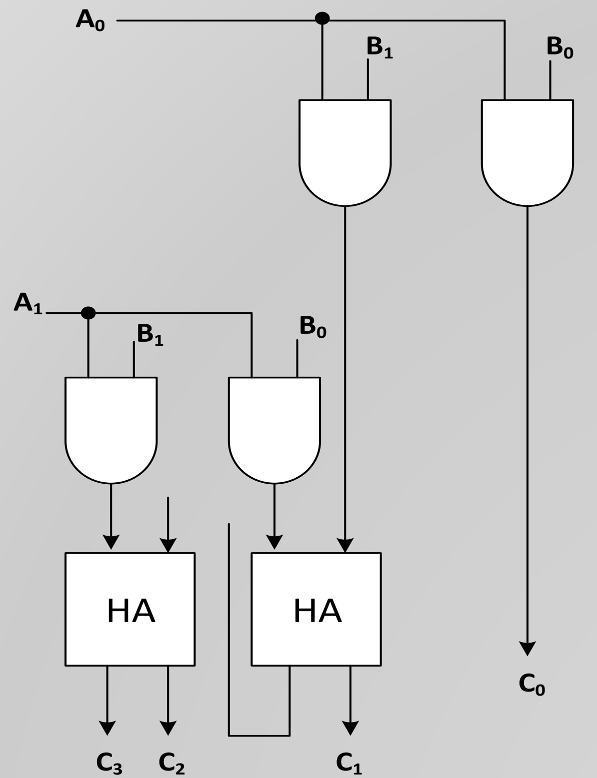
Δυαδικός Πολλαπλασιαστής (2)

- Παράδειγμα:
  - Πολ/στης A = A_1A_0 και πολλαπλασιαστέος B = B_1B_0
  - Βρείτε το C = A x B :

$$
\begin{array}{cccc}
 & & B_1 & B_0 \\
 & \times & \underline{A_1} & \underline{A_0} \\
 & & A_0B_1 & A_1B_0 \\
\underline{+} & \underline{A_1B_1} & \underline{A_1B_0} & \\
C_3 & C_2 & C_2 & C_0
\end{array}
$$

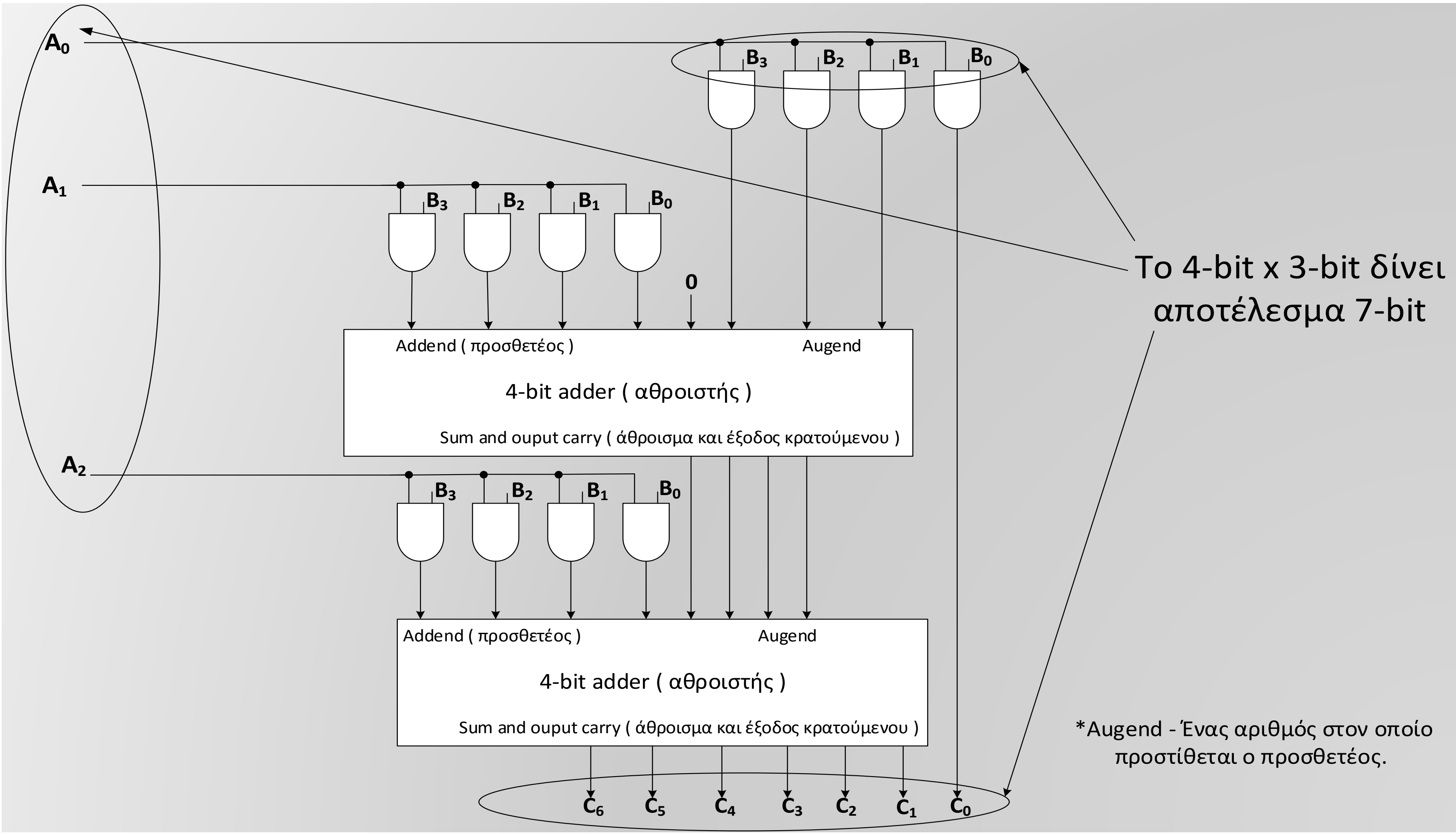
Κύκλωμα δυαδικού πολ/στη 2bit x 2bit

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



- Οι Half Adders (ημιαθροιστές) είναι αρκετοί αφού δεν υπάρχει Carry-in (κρατούμενο) μαζί με τις δύο εισόδους της πρόσθεσης.

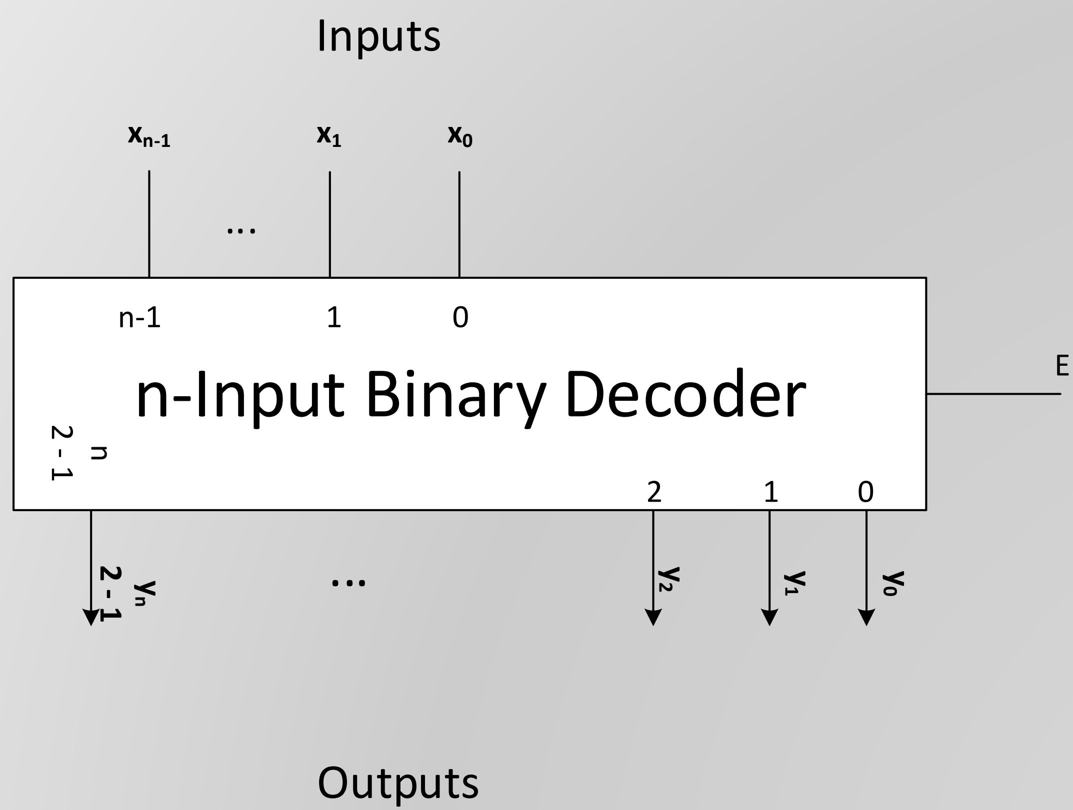
Κύκλωμα δυαδικού πολ/στη 4bit x 3bit



Δυαδικοί Αποκωδικοποιητές (1)

- Συνδυαστικό κύκλωμα για μετατροπή δυαδικών δεδομένων απο n κωδικοποιημένες εισόδους σε 2^n κωδικοποιημένες εξόδους. → Αποκωδικοποιητής (Binary Decoder) n-to-2^n.
- Αποκωδικοποιητής (Code Converter) n-σε-m, $2^n \geq m$.
  – Παραδείγματα: BSD-σε-7-segment και BCD-σε-Excess-3, όπου n = 4 και m = 10.

Δυαδικοί Αποκωδικοποιητές (2)

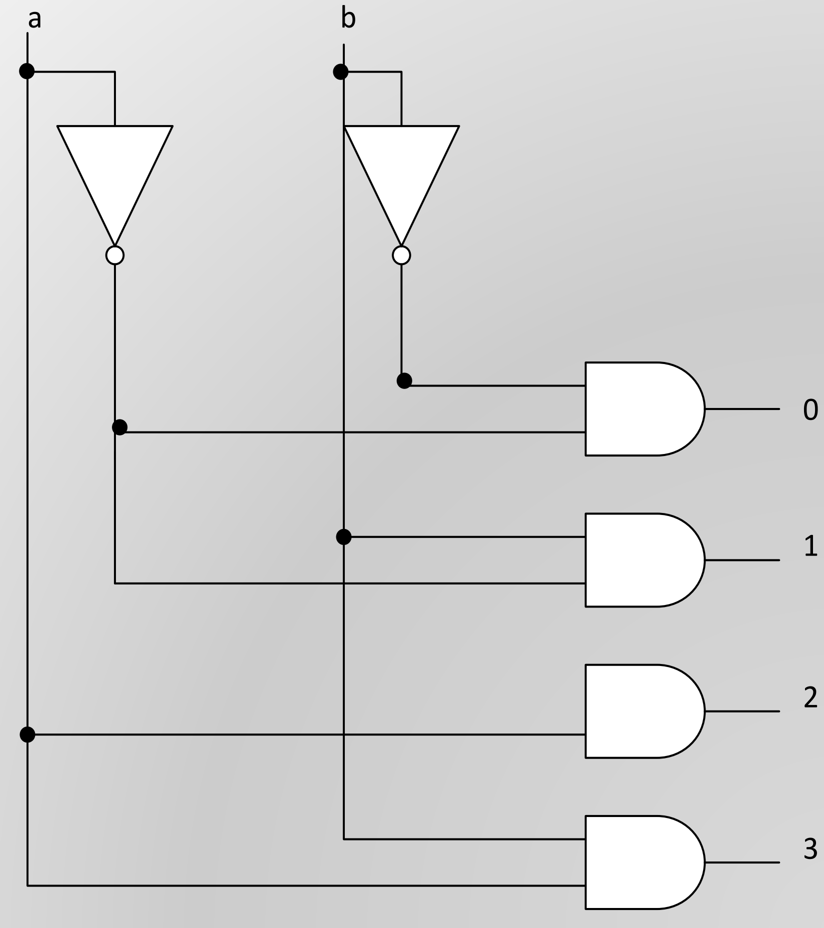


Inputs: Είσοδοι
Outputs: Έξοδοι
n-Input Binary Decoder: n Είσοδοι δυαδικού αποκωδικοποιητή

Αποκωδικοποιητής 2-σε-4 (1)

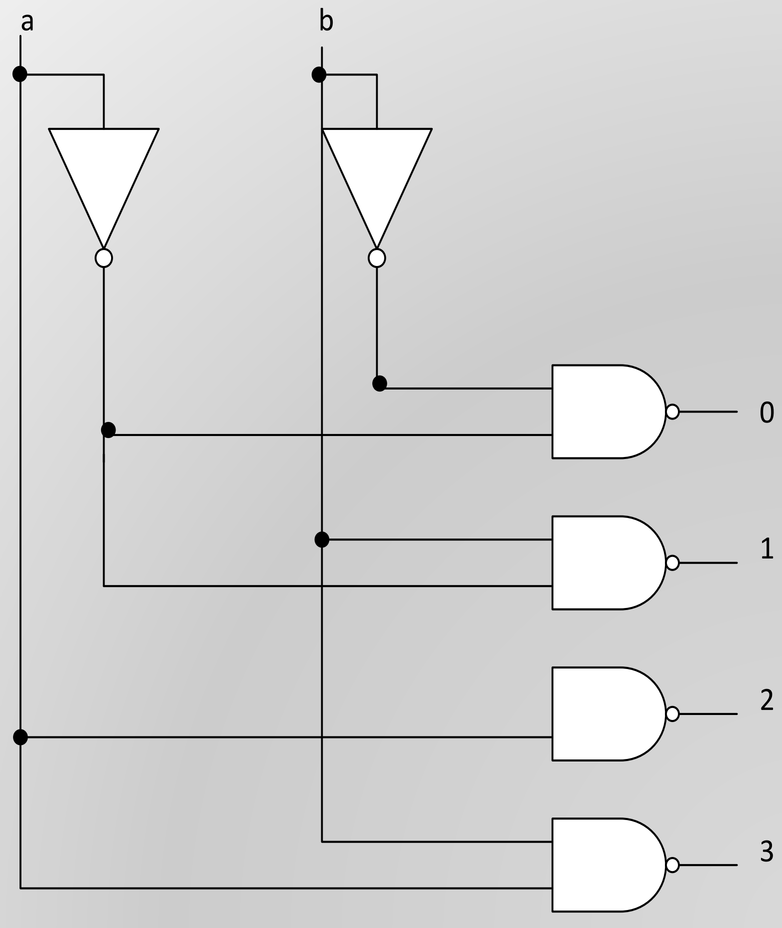
a	b	0	1	2	3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

- Σχεδιάστε ένα αποκωδικοποιητή 1-σε-2.

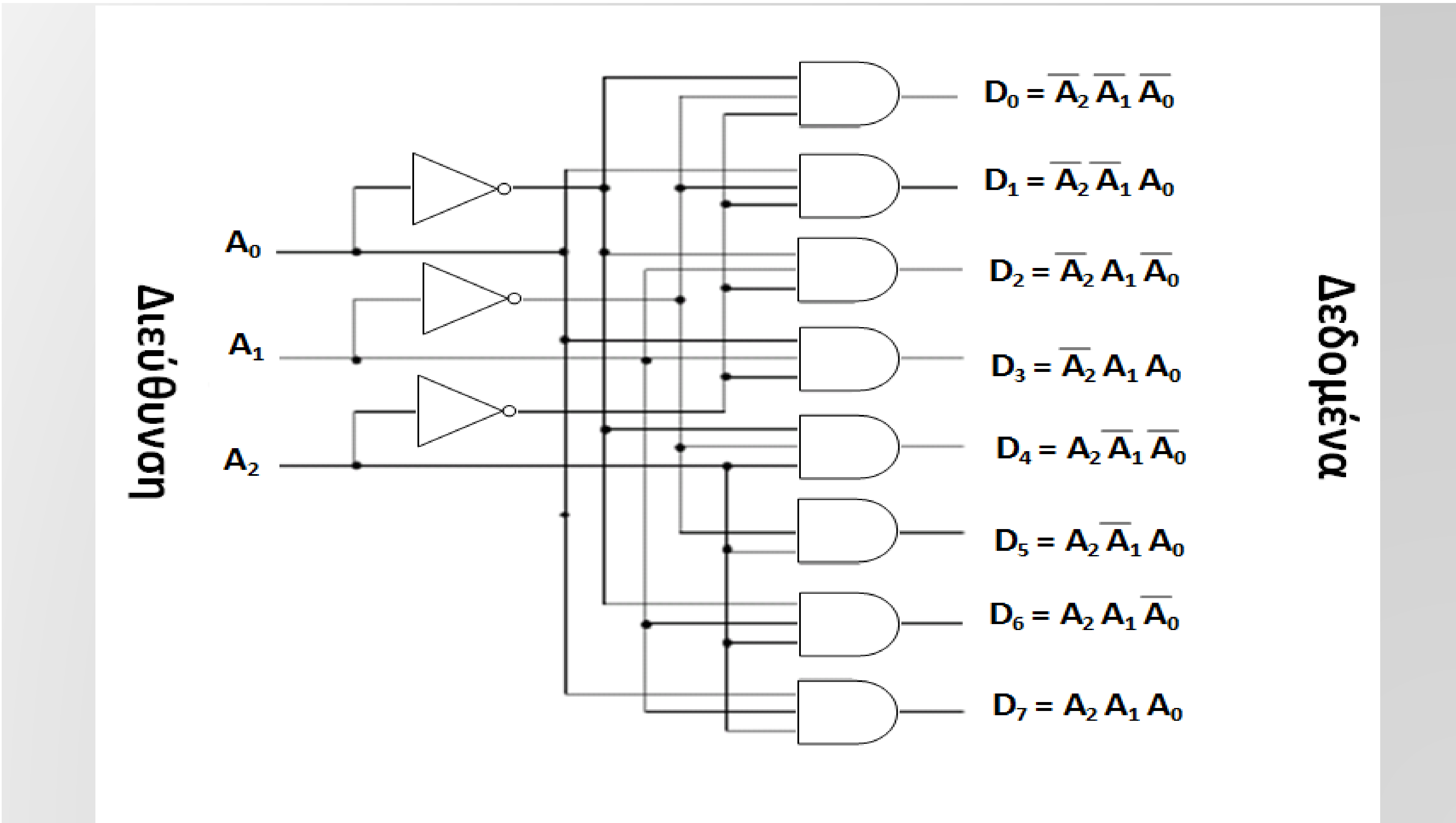


Αποκωδικοποιητής 2-σε-4 (2)

a	b	0	1	2	3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0



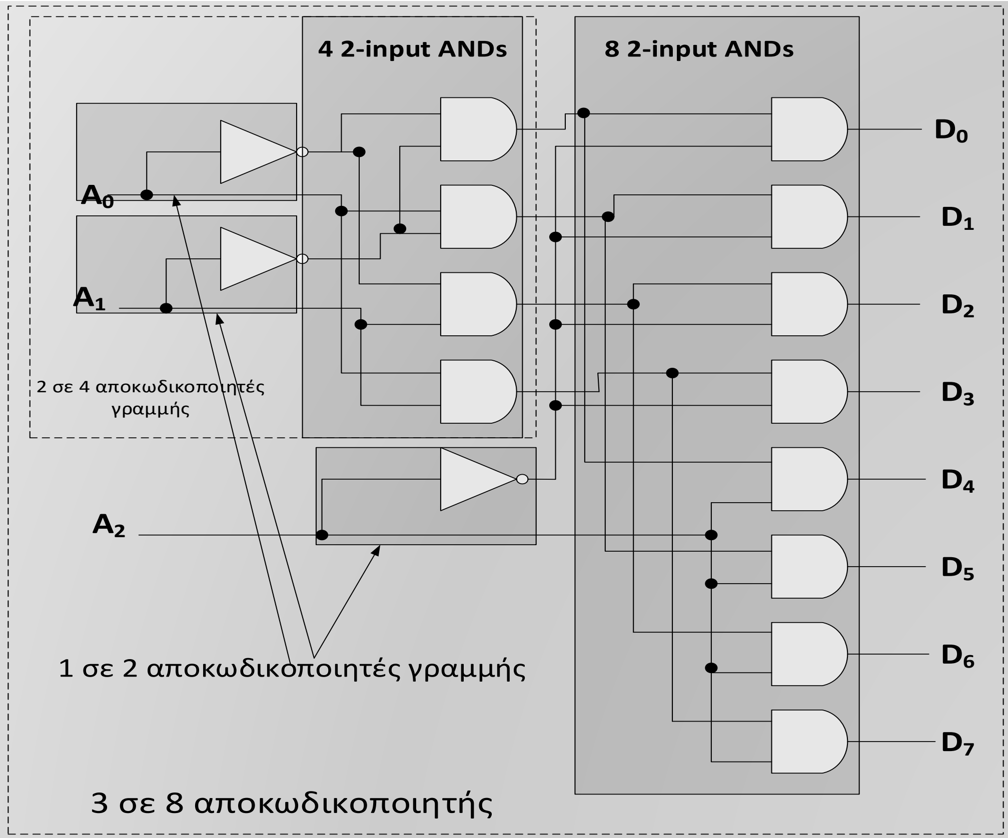
Αποκωδικοποιητής 3-σε-8 (1)



Αποκωδικοποιητής 3-σε-8 (2)

- Τρεις είσοδοι, $A_0A_1A_2$, αποκωδικοποιούνται σε οκτώ εξόδους. D_0 εώς D_7.
- Κάθε έξοδος D_i αντιπροσωπεύει έναν από τους ελαχιστόρους των $3^{ων}$ μεταβλητών εισόδου.
- $D_i = 1$ όταν ο δυαδικό αριθμός $A_0A_1A_2 = i$.
- Συντομογραφία $D_i = m_i$.
- Οι τιμές στις εξόδους έχουν αμοιβαία αποκλειστικότητα (mutually exclusive), δηλ. ΜΟΝΟ μία έξοδος μπορεί να έχει την τιμή 1 ανά πάσα στιγμή, και οι υπόλοιπες έχουν την τιμή 0.

Αποκωδικοποιητής 3-σε-8 (3)



4 2-input ANDs: 4 Διπλές είσοδοι με πύλες AND
8 2-input ANDs: 8 Διπλές είσοδοι με πύλες AND

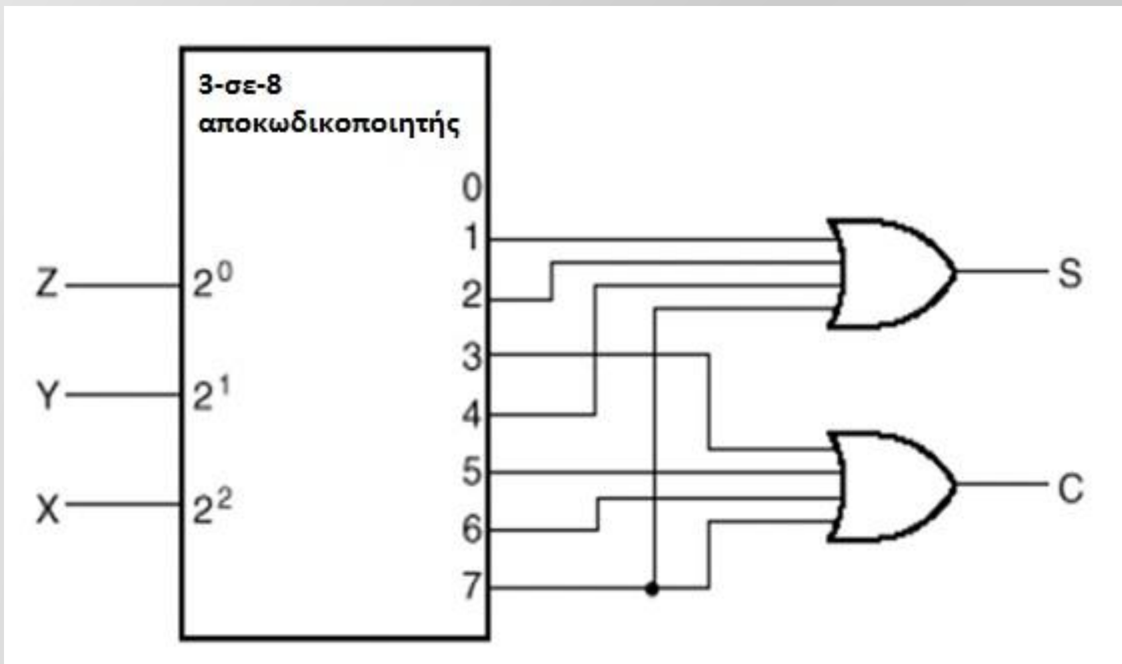
Υλοποίηση δυαδικών συναρτήσεων με τη χρήση αποκωδικοποιητών

- Οποιοδήποτε συνδυαστικό κύκλωμα μπορεί να υλοποιηθεί χρησιμοποιώντας μόνο έναν αποκωδικοποιητή και πύλες OR! Γιατι;
- Παράδειγμα:

  Υλοποιήστε έναν πλήρη αθροιστή με έναν αποκωδικοποιητή και 2 πύλες OR.
- Θεωρήστε X, Y και Z για εισόδους, S και C για εξόδους:
  - $S(X, Y, Z) = X + Y + Z = \sum m(1, 2, 4, 7)$
  - $C(X, Y, Z) = \sum m(3, 5, 6, 7)$
- Αφού υπάρχουν 3 είσοδοι και άρα 8 συνολικοί ελαχιστόροι, χρειαζόμαστε έναν αποκωδικοποιητή 3-σε-8.

Υλοποίηση Δυαδικού Αθροιστή με χρήση αποκωδικοποιητή

- S(X, Y, Z) = Σm(1, 2, 4, 7)   C(X, Y, Z) = Σm(3, 5, 6, 7)



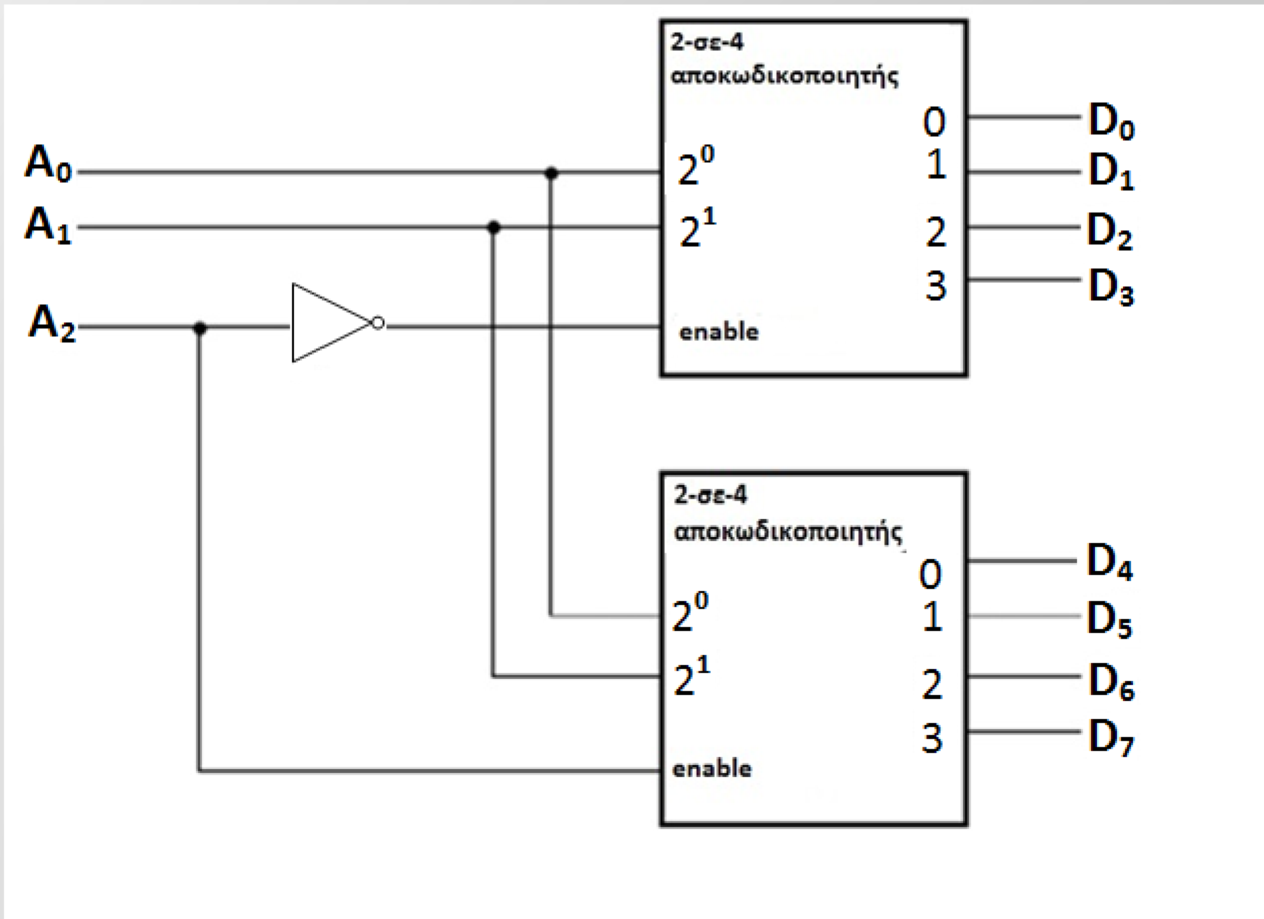
Επέκταση Αποκωδικοποιητή

- Μπορούμε να κατασκευάσουμε ένα μεγαλύτερο αποκωδικοποιητή χρησιμοποιώντας έναν αριθμό από μικρότερους.
- ΙΕΡΑΡΧΙΚΟΣ σχεδιασμός!
- Παράδειγμα:

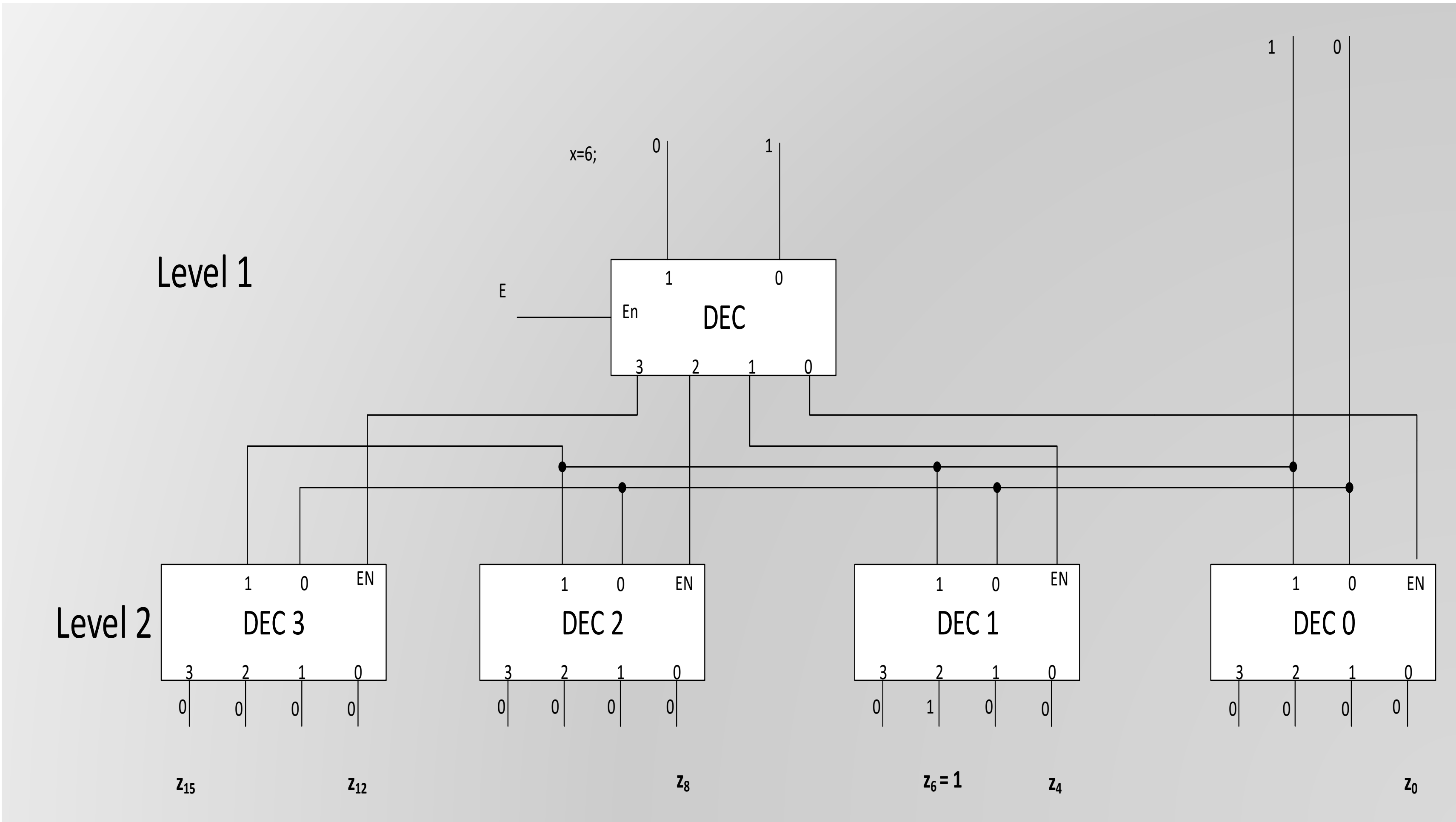
  Ένας αποκωδικοποιητής 6-σε-24 μπορεί να σχεδιαστεί με τέσσερις 4-σε-16 και έναν 2-σε-4. Πώς;

  (Υπόδειξη: Χρησιμοποιήστε τον 2-σε-4 για να παράγει το σήμα ενεργοποίησης των τεσσαρων 4-σε-16).

Αποκωδικοποιητής 3-σε-8 μέσα σε δύο 2-σε-4

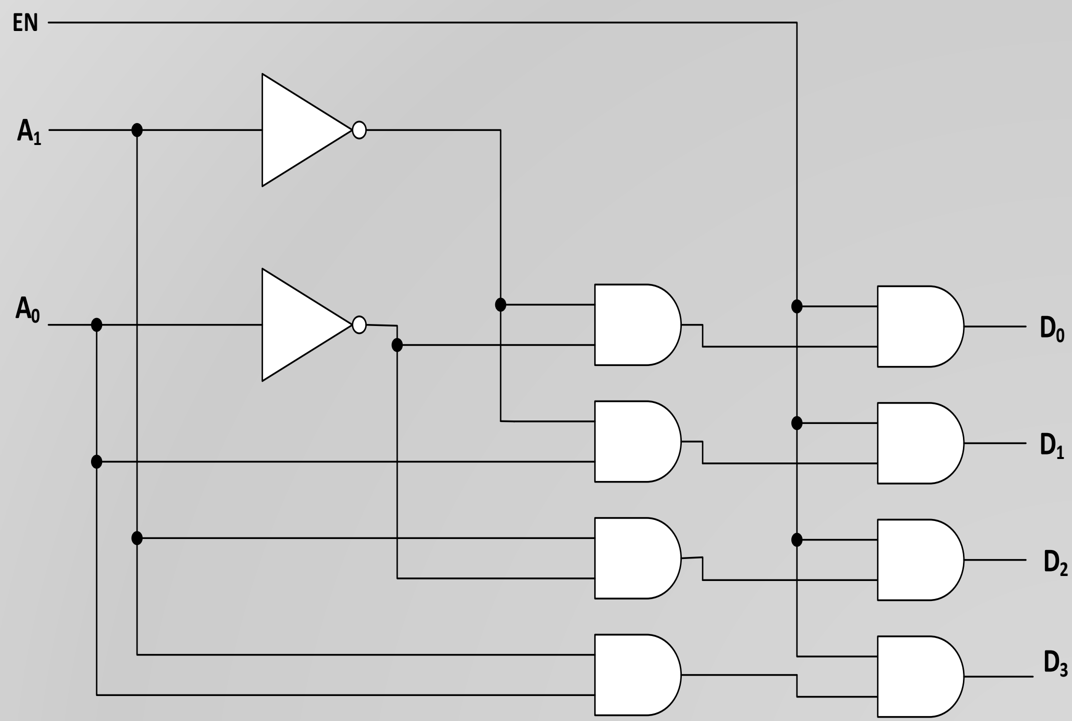


Δένδρο αποκωδικοποιητή με 4 εισόδους



Αποκωδικοποιητής με Enable

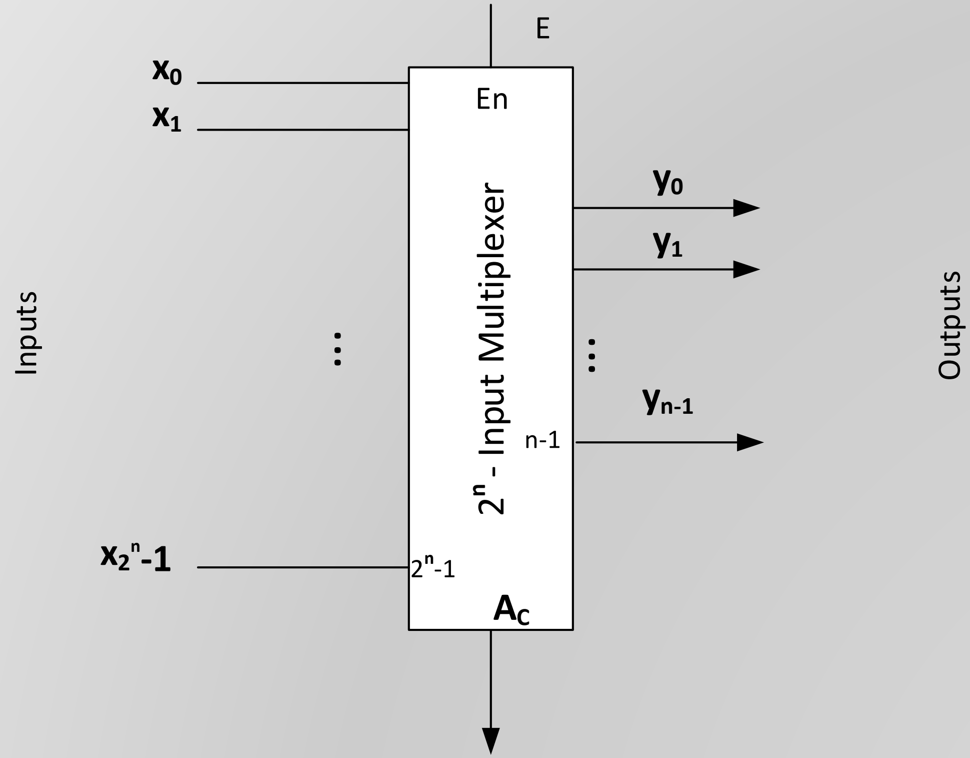
EN	A_1	A_0	D_0	D_1	D_2	D_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



Κωδικοποιητές (1)

- Συνδυαστικό κύκλωμα που διεκπεραιώνει την αντίστροφη λειτουργία από αυτή του αποκωδικοποιητή.
- Έχει 2^n εισόδους και n εξόδους.
- ΜΟΝΟ 1 είσοδος 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 (3)

- Παράδειγμα: δυαδικός κωδικοποιητής 8-σε-3

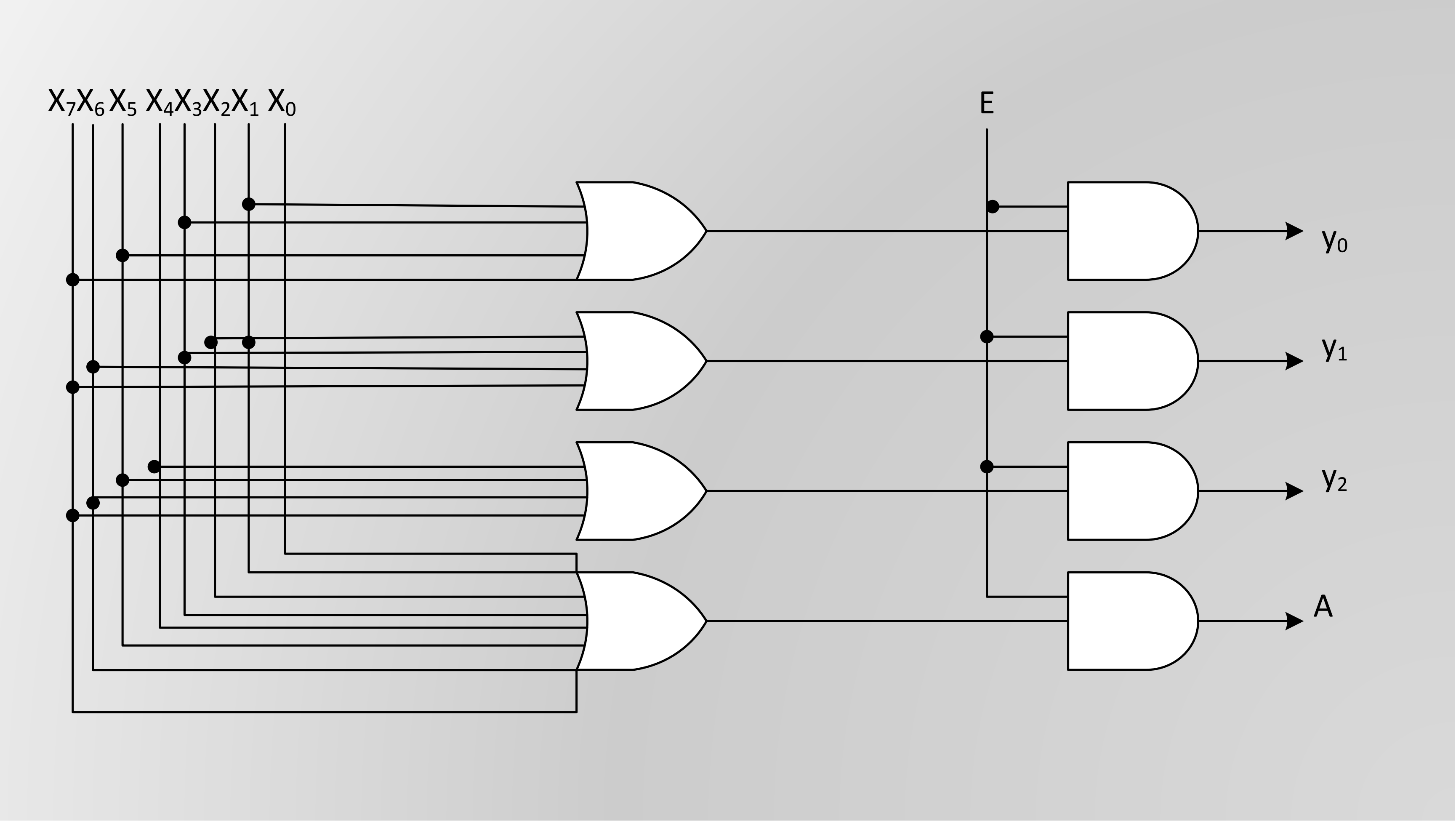
Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$A_0 = D_1 + D_2 + D_5 + D_7$

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

από 8-αδικό σε 2αδικό $A_2 = D_4 + D_5 + D_6 + D_7$

Κωδικοποιητές (4)



Θέματα Σχεδιασμού Κωδικοποιητών

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

1. ΜΟΝΟ μία 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. Αποτελέσματα 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 μία έχει προτεραιότητα από όλες τις άλλες.
  - Ρητή ένδειξη όταν καμία από τις εισόδους δεν είναι 1.

Κωδικοποιητής προτεραιότητας 4-σε-2 (1)

- Συμπυκνωμένως Πίνακας Αληθείας.

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

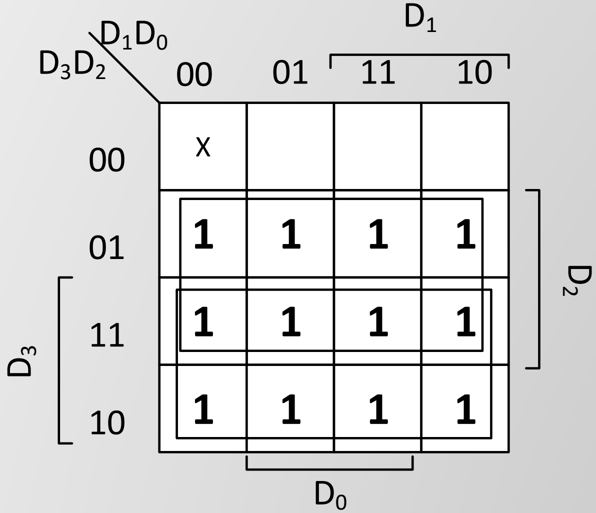
Κωδικοποιητής προτεραιότητας 4-σε-2 (2)

- 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 μία ή περισσότερες από τις ειδόδους έχουν την τιμή 1.

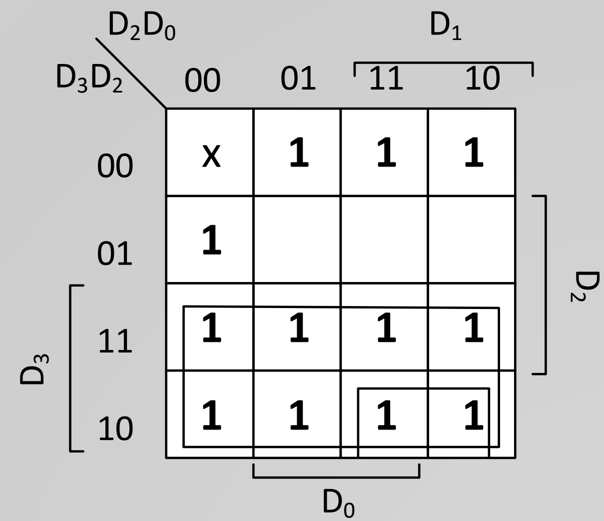
$V = D_3 + D_2 + D_1 + D_0$

Κωδικοποιητής προτεραιότητας 4-σε-2 (3)

K – Χάρτες



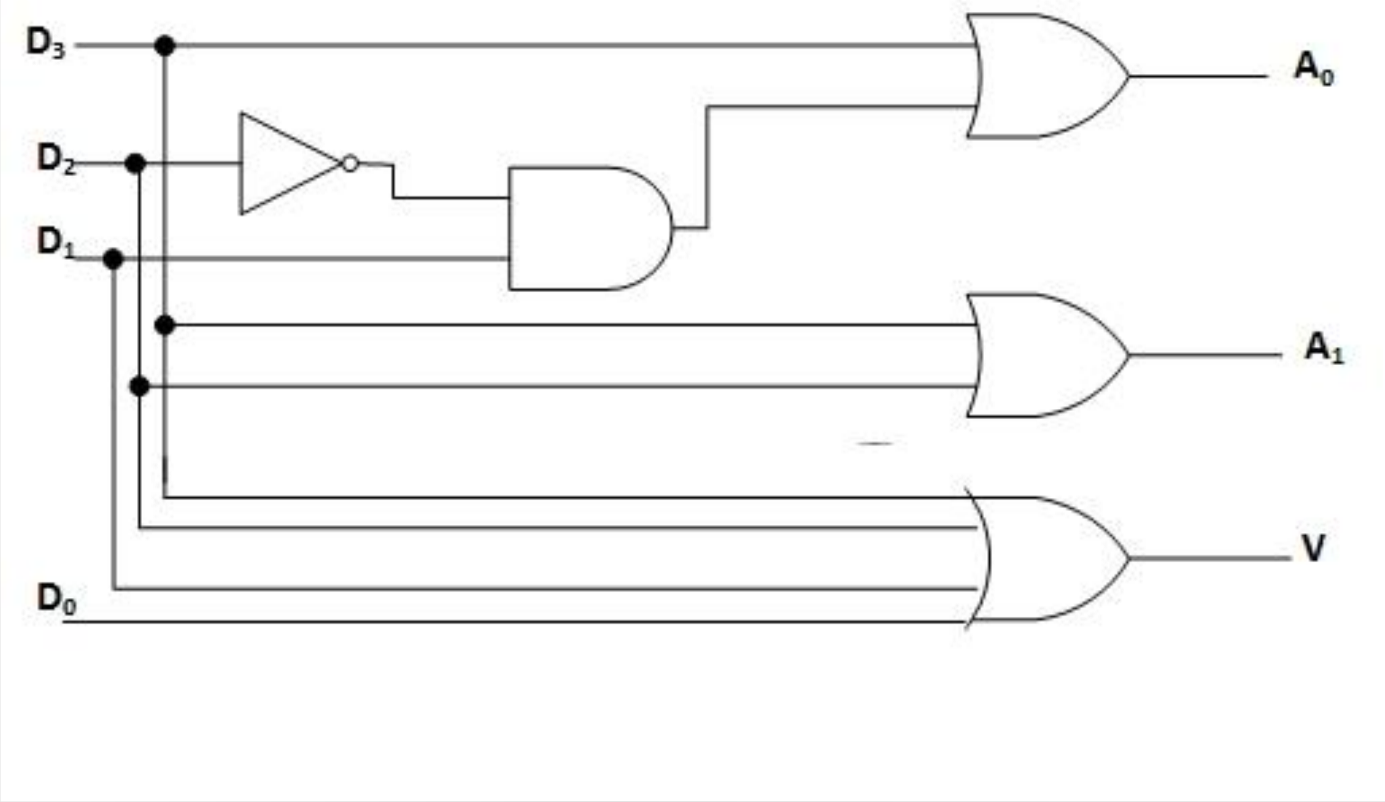
$A_1 = D_2 + D_3$



$A_0 = D_3 + D_1\overline{D_2}$

Κωδικοποιητής προτεραιότητας 4-σε-2 (4)

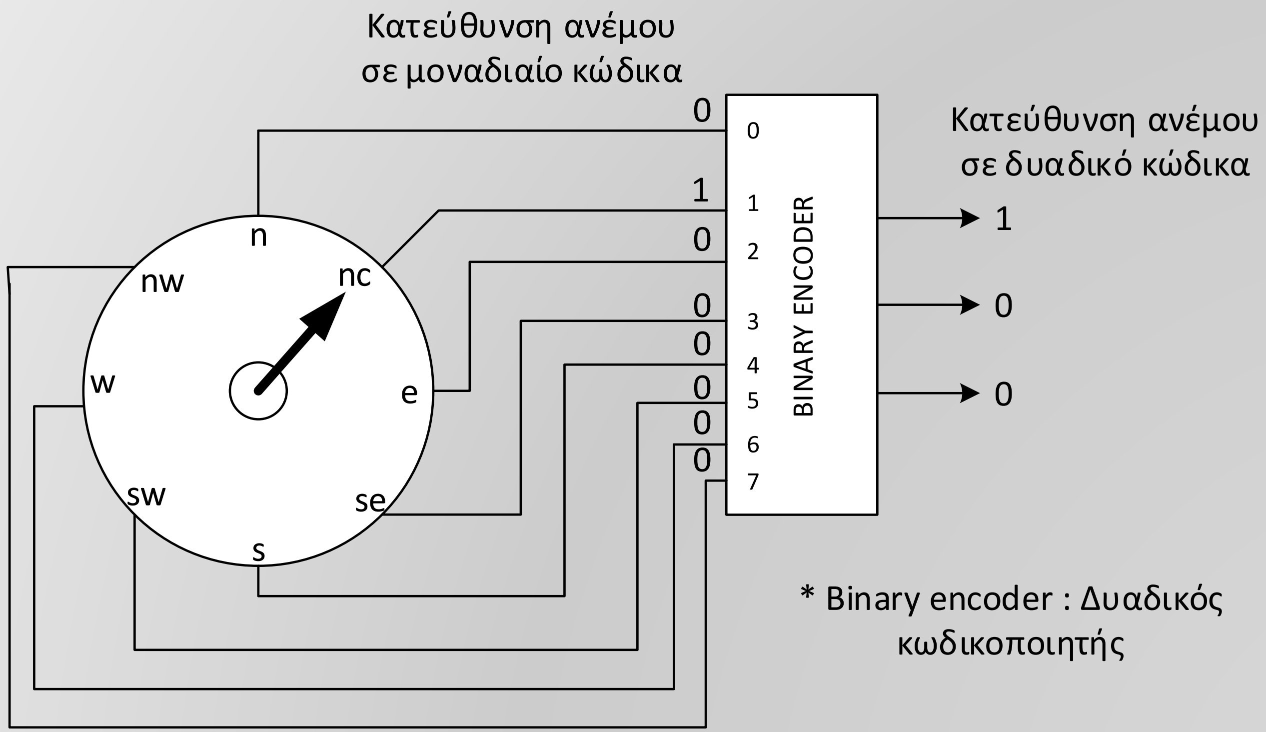
Λογικό Διάγραμμα



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

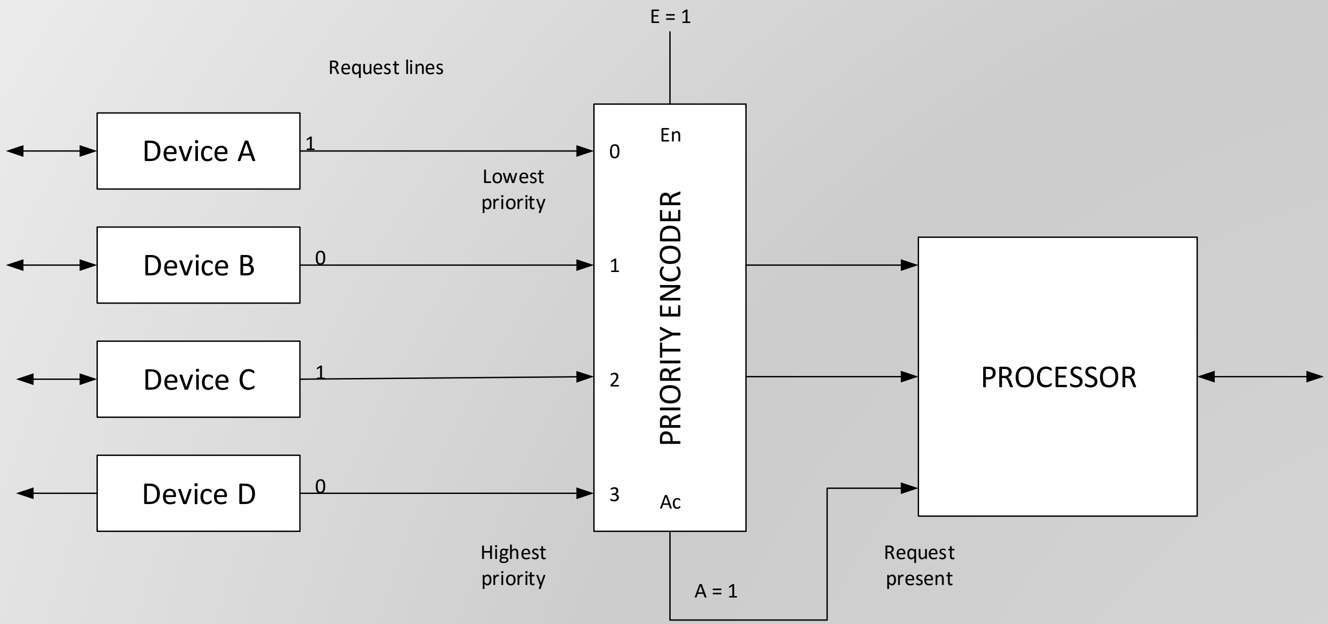
Χρήσεις Δυαδικού Κωδικοποιητή (1)



* Binary encoder : Δυαδικός κωδικοποιητής

- Δυαδική κωδικοποίηση κατεύθυνσης ανέμου

Χρήσεις Δυαδικού Κωδικοποιητή (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

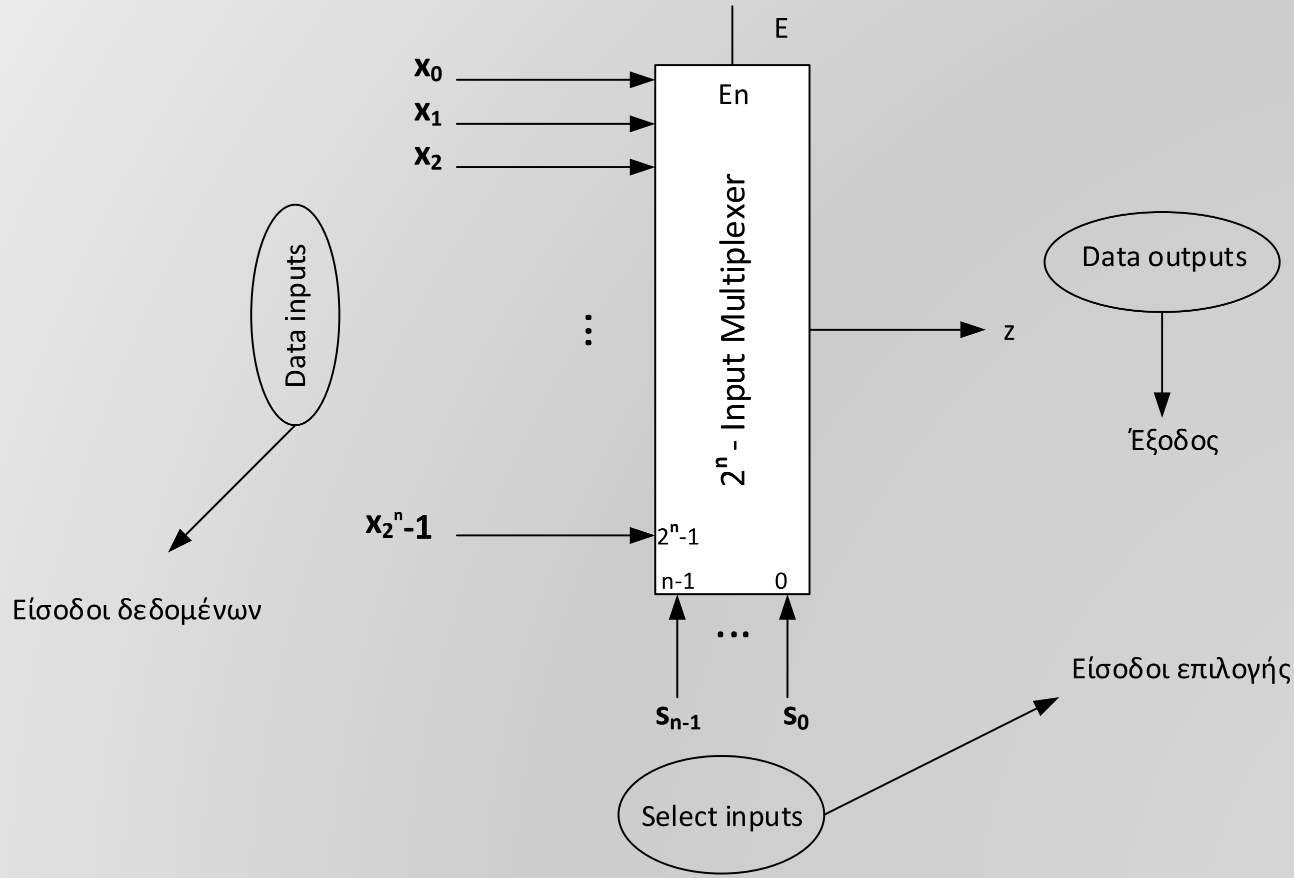
Device: Συσκευή

Πολυπλέκτες (1)

- Κύκλωμα που «επιλέγει» δυαδική πληροφορία από μία από τις εισόδους και την κατευθύνει στην μοναδική έξοδο.
- Επίσης γνωστό ως «επιλογέας» (selection circuit).

- Η επιλογή ελέγχετε από ένα σύνολο εισόδων, ο αριθμός των οποίων εξαρτάται από τον # των εισόδων δεδομένων.
- Για έναν 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 να καθορίζει την έισοδο δεδομένων που θα επιλεγεί.

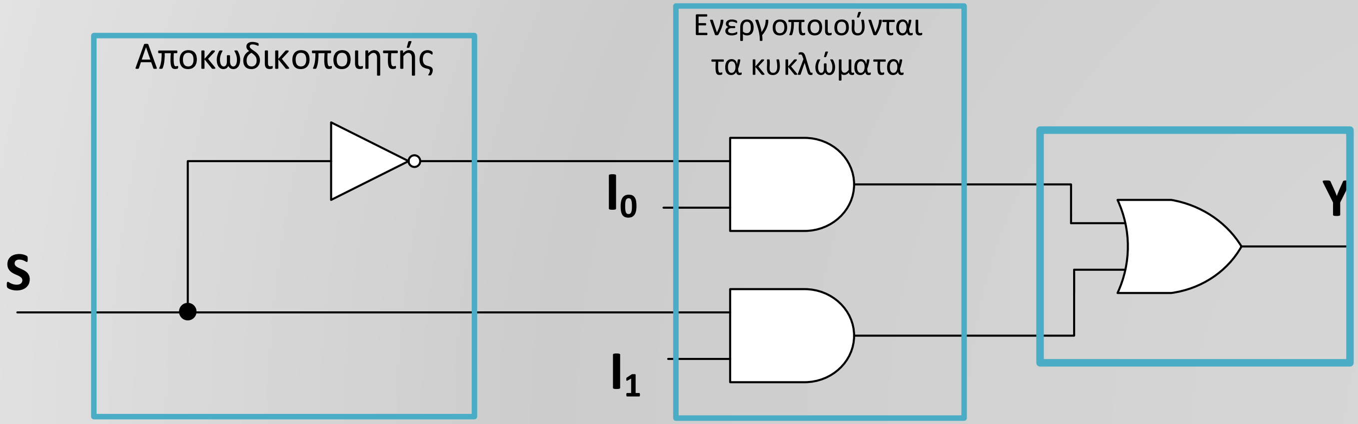
Πολυπλέκτες (2)



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

2-σε-1 MUX (1)

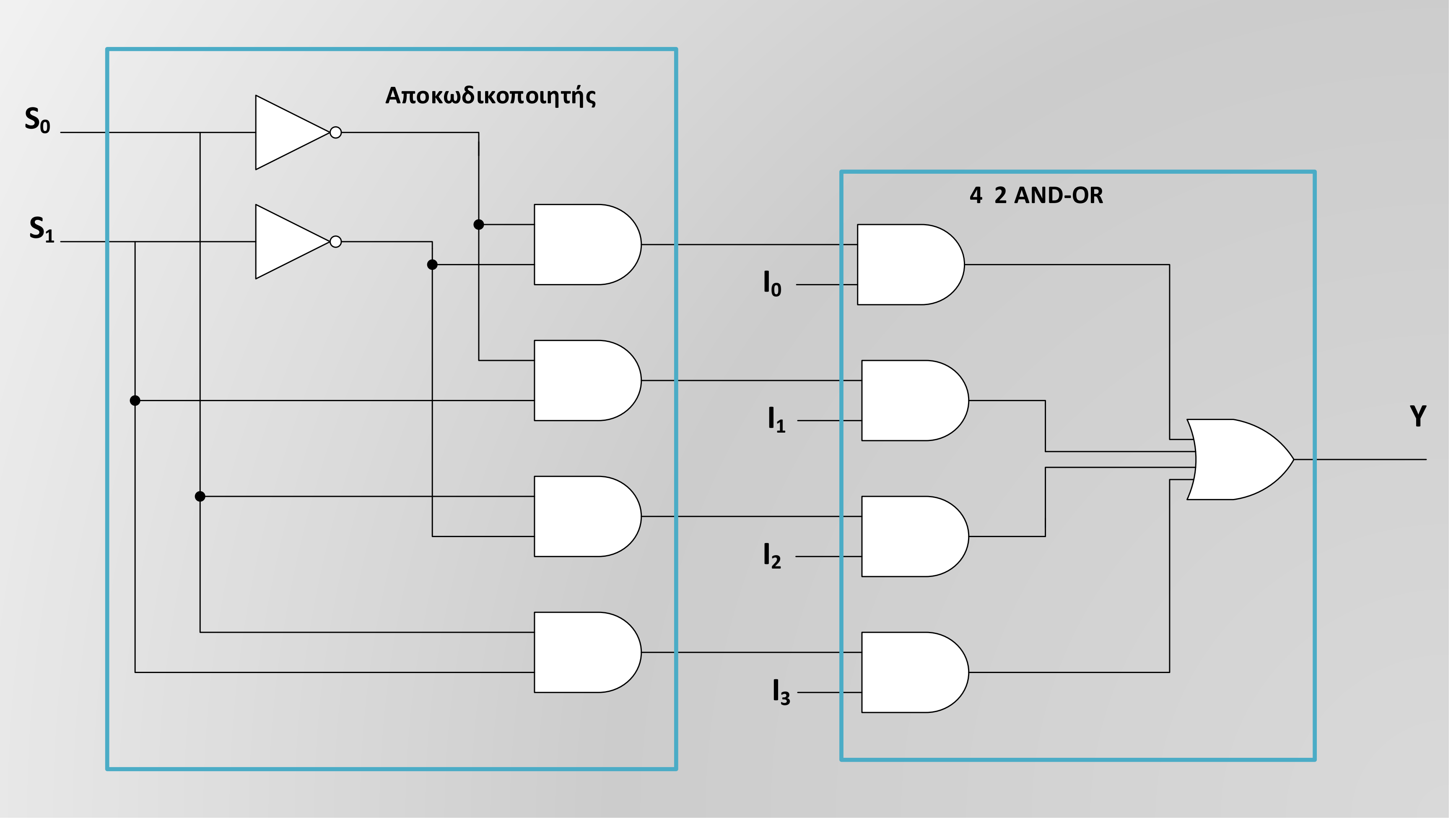
- Αφού υπάρχουν 2 είσοδοι δεδομένων, $2 = 2^1 \rightarrow n = 1$
- Υπάρχει μια είσοδος επιλογής S:
  - $S = 0$ επιλέγει την είσοδο I_0
  - $S = 1$ επιλέγει την είσοδο I_1
- Υλοποιεί την συνάρτηση: $Y = S'I_0 + SI_1$
- Το λογικό διάγραμμα:



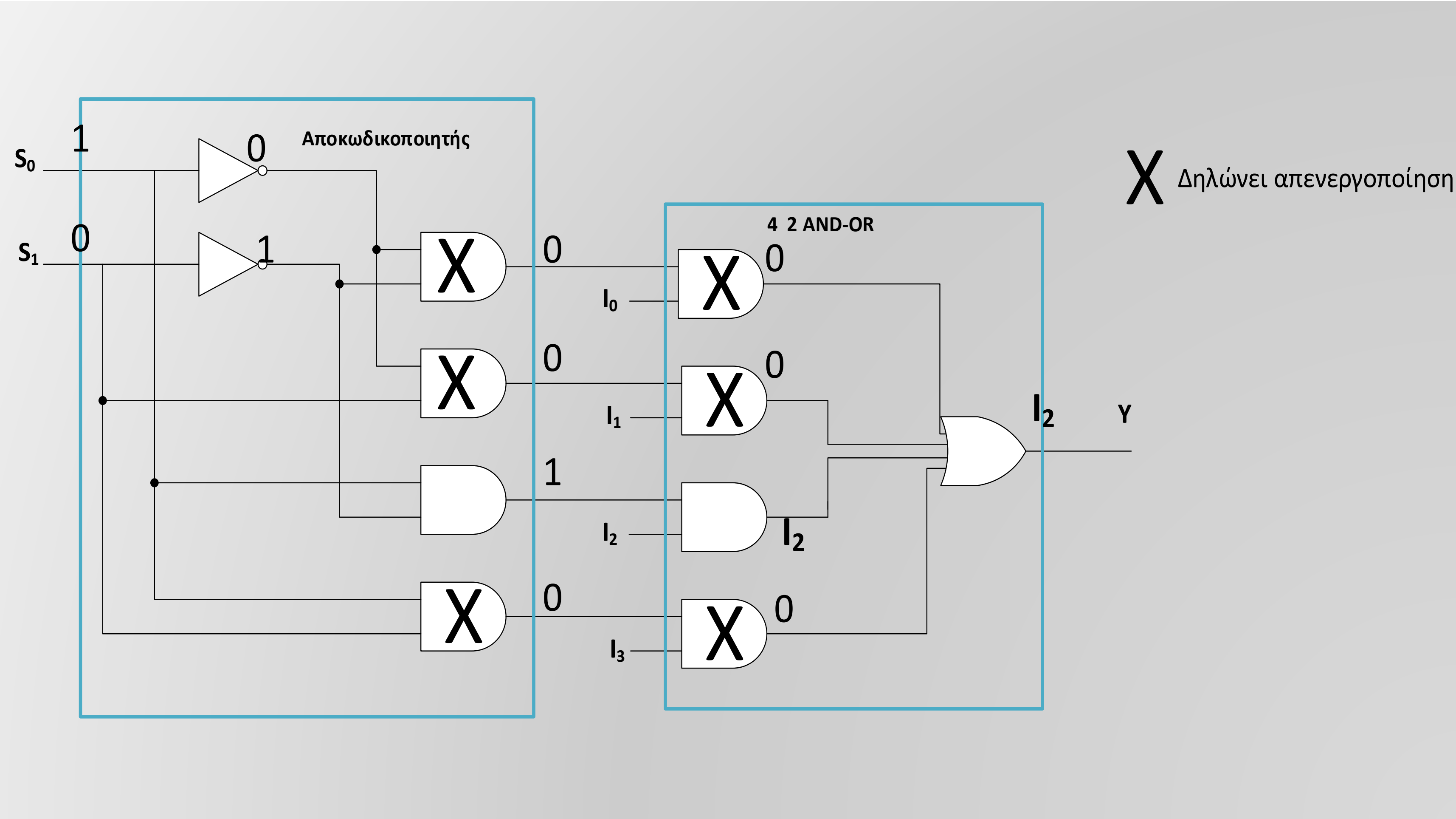
2-σε-1 MUX (2)

- Προσέξτε ότι τα διάφορα μέρη του πολυπλέκτη δείχνουν:
  - Ενα 1-σε-2 Αποκωδικοποιητή
  - Δύο κυκλώματα ενεργοποίησης (enable circuits)
  - Μια πύλη OR 2-εισόδων
- Τα πιο πάνω συνδυάζονται για να μας δώσουν τον πολυπλέκτη, τα κυκλώματα ενεργοποίησης, και η πύλη OR 2-εισόδων δίνουν ένα κύκλωμα 2x2 AND-OR, όπου οι 4 εισόδοι του προέρχονται από τις 2 εισόδους δεδομένων και τις 2 εισόδους του αποκωδικοποιητή.
  - 2 είσοδοι δεδομένων
  - 1-σε-2 αποκωδικοποιητή (παράγουν τους ελαχιστόρους)
  - 2x2 AND - OR
- Γενικά για έναν πολυπλέκτη 2^n-σε-1
  - 2^n είσοδοι δεδομένων
  - n-σε-2^n αποκωδικοποιητή
  - 2^nx2 AND - OR

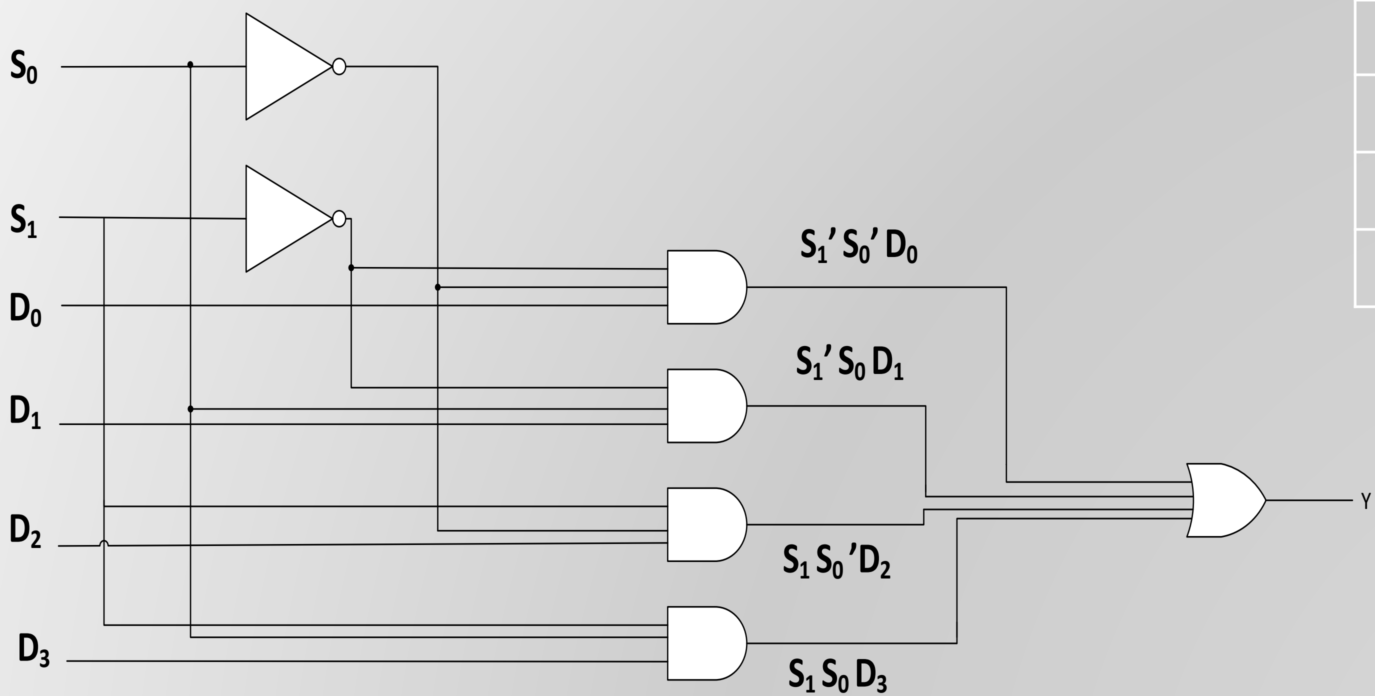
Παράδειγμα: 4-σε-1 MUX (1)



Παράδειγμα: 4-σε-1 MUX (2)

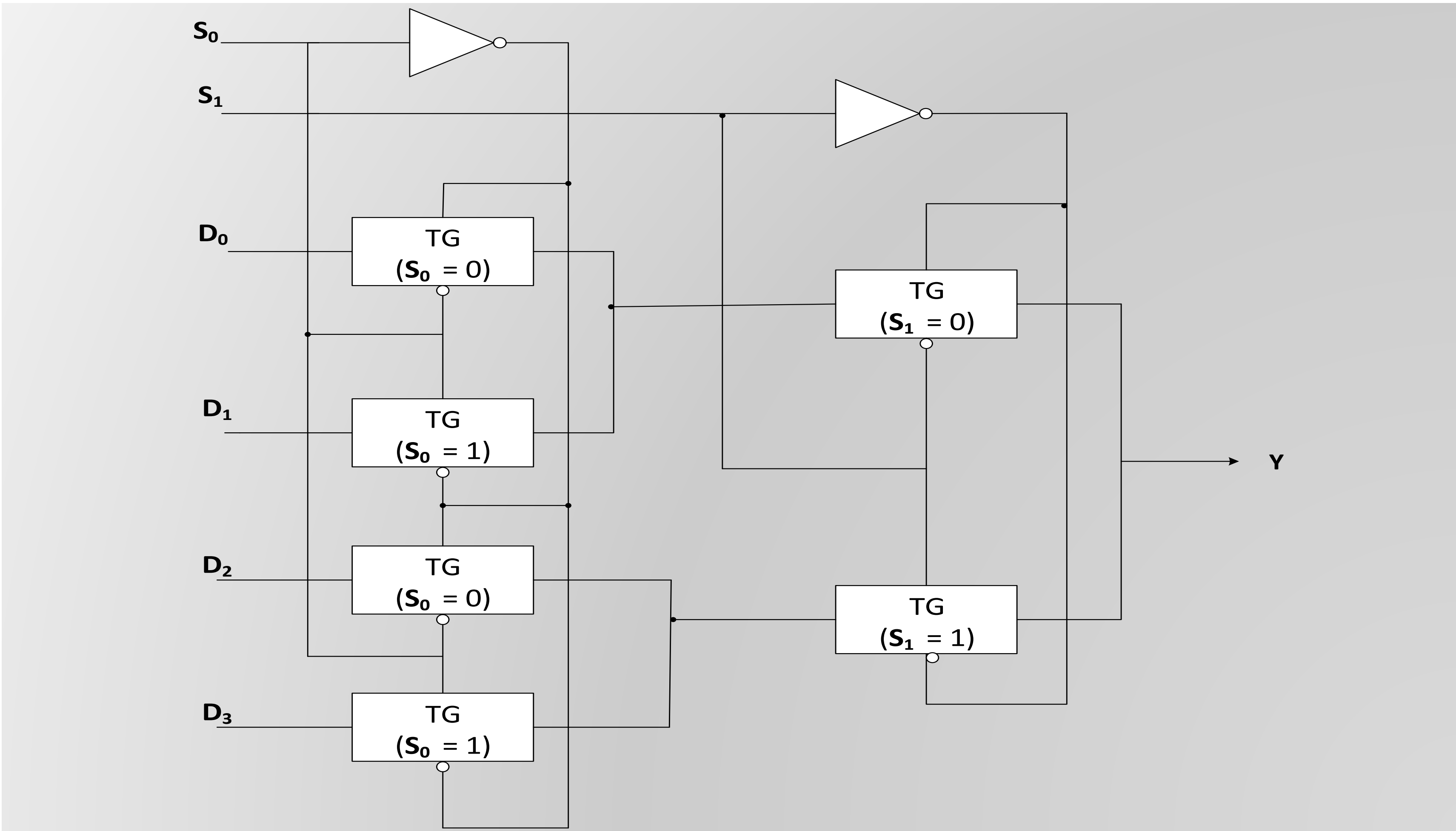


Παράδειγμα: 4-σε-1 MUX (3)



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 (4)



Πολυπλέκτες

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

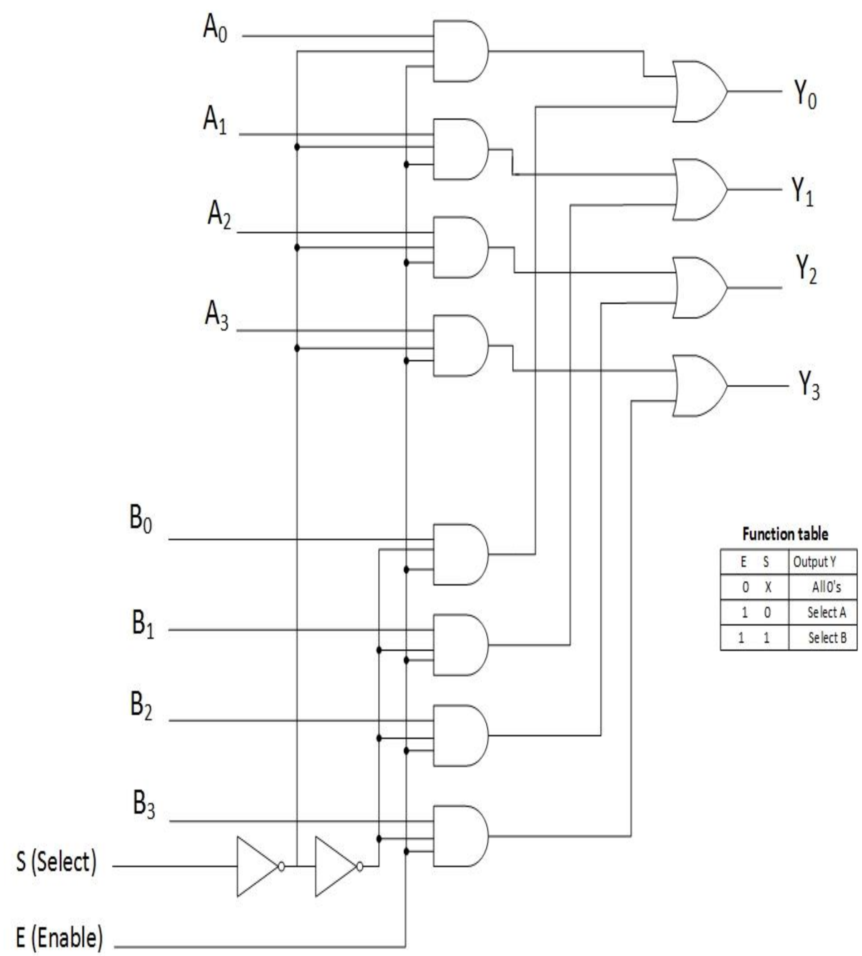
→ Συνδυάζουμε κυκλωματα MUX παράλληλα, με κοινές εισόδους επιλογής και ενεργοποίησης.

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

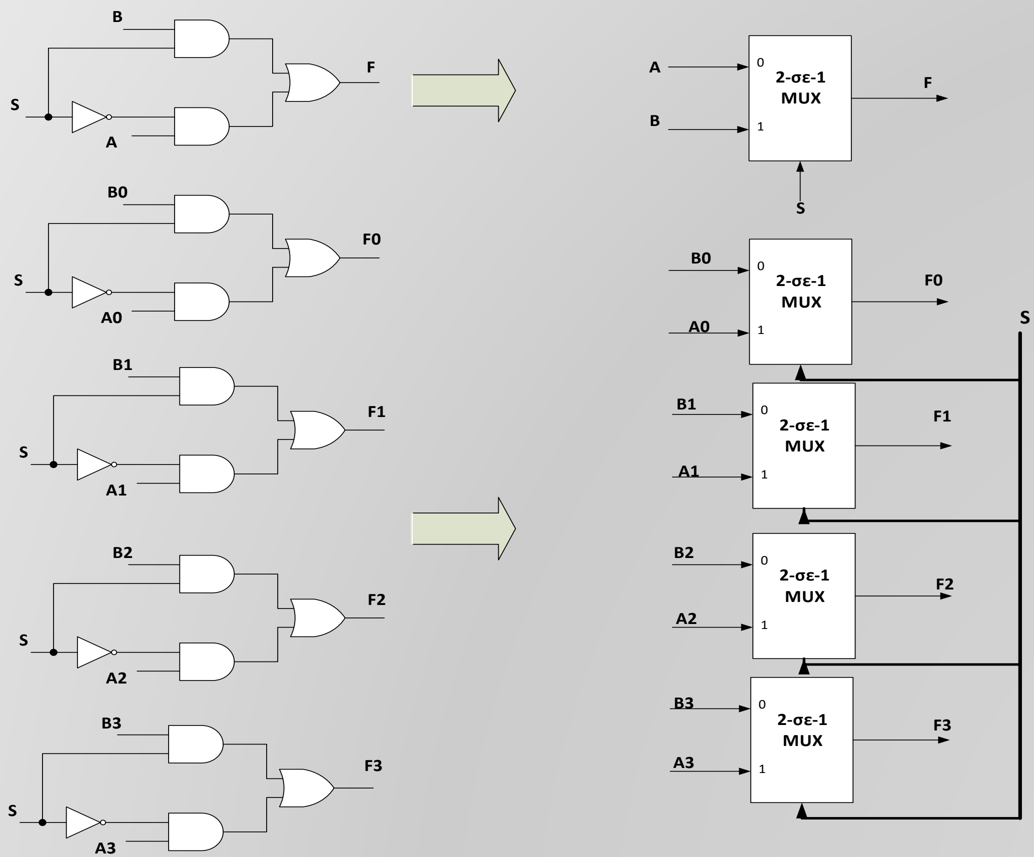
→ Τετραπλός 2-σε-1 πολυπλέκτης (Quad 2-to-1 MUX).

Τετραπλό (QUAD) 2-σε-1 MUX (1)

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

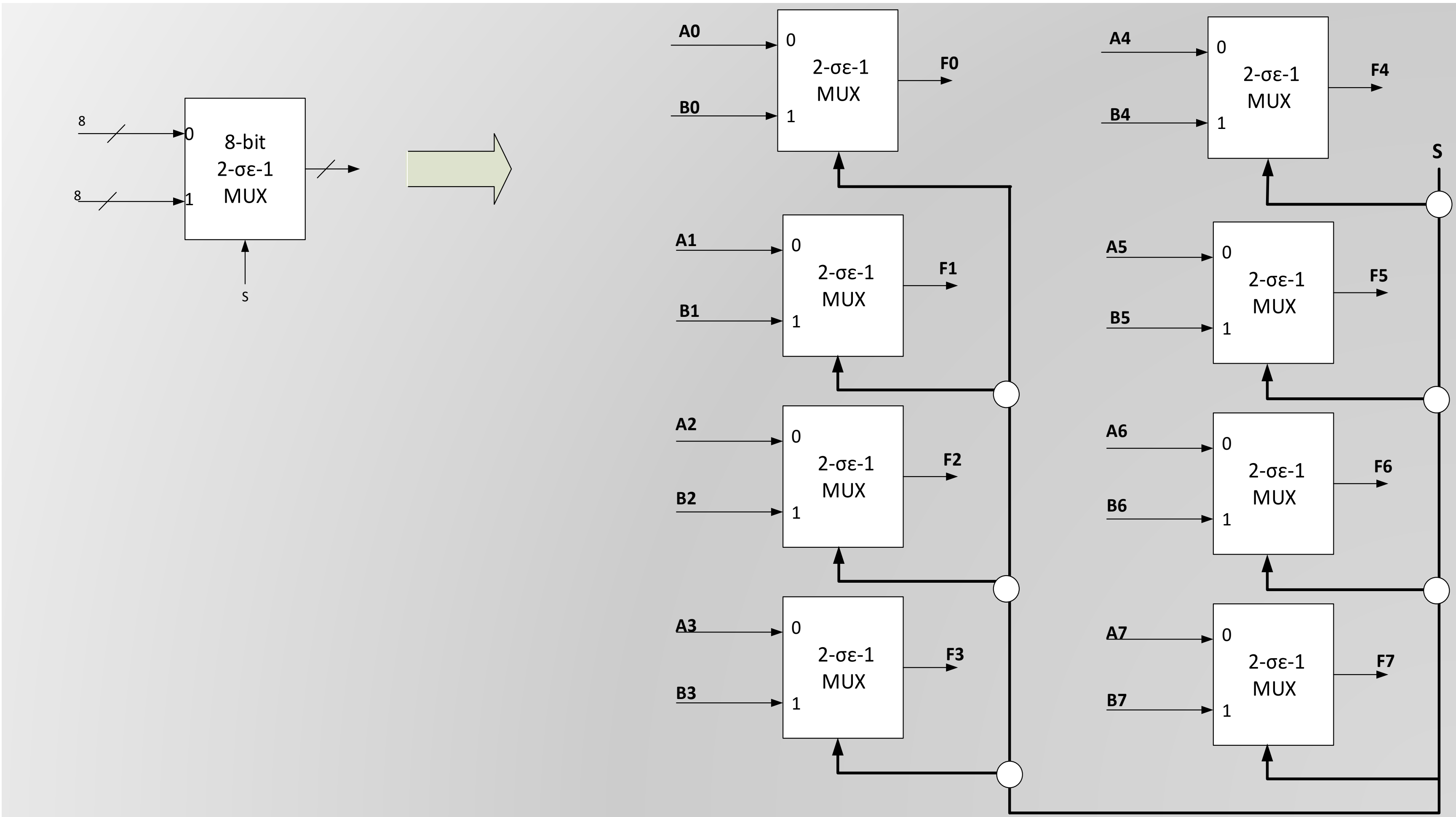


Τετραπλό (QUAD) 2-σε-1 MUX (2)

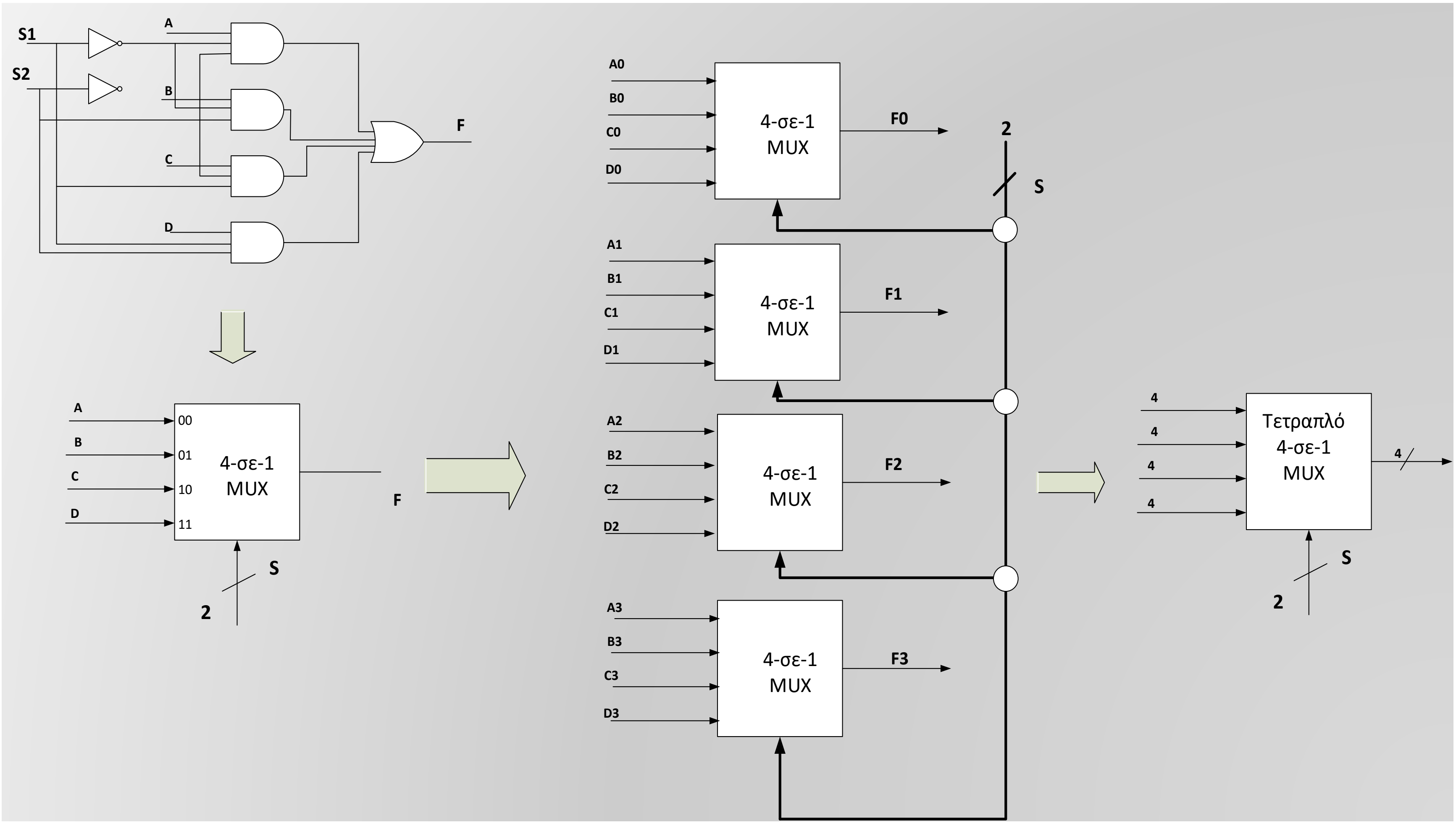


- Χρησιμοποιεί τέσσερις MUX 2-σε-1, με κοινή είσοδο επιλογής (S).
- Η είσοδος επιλογής S επιλέγει μεταξύ των A_i's και B_i's και στέλνει στα αντίστοιχα Y_i's

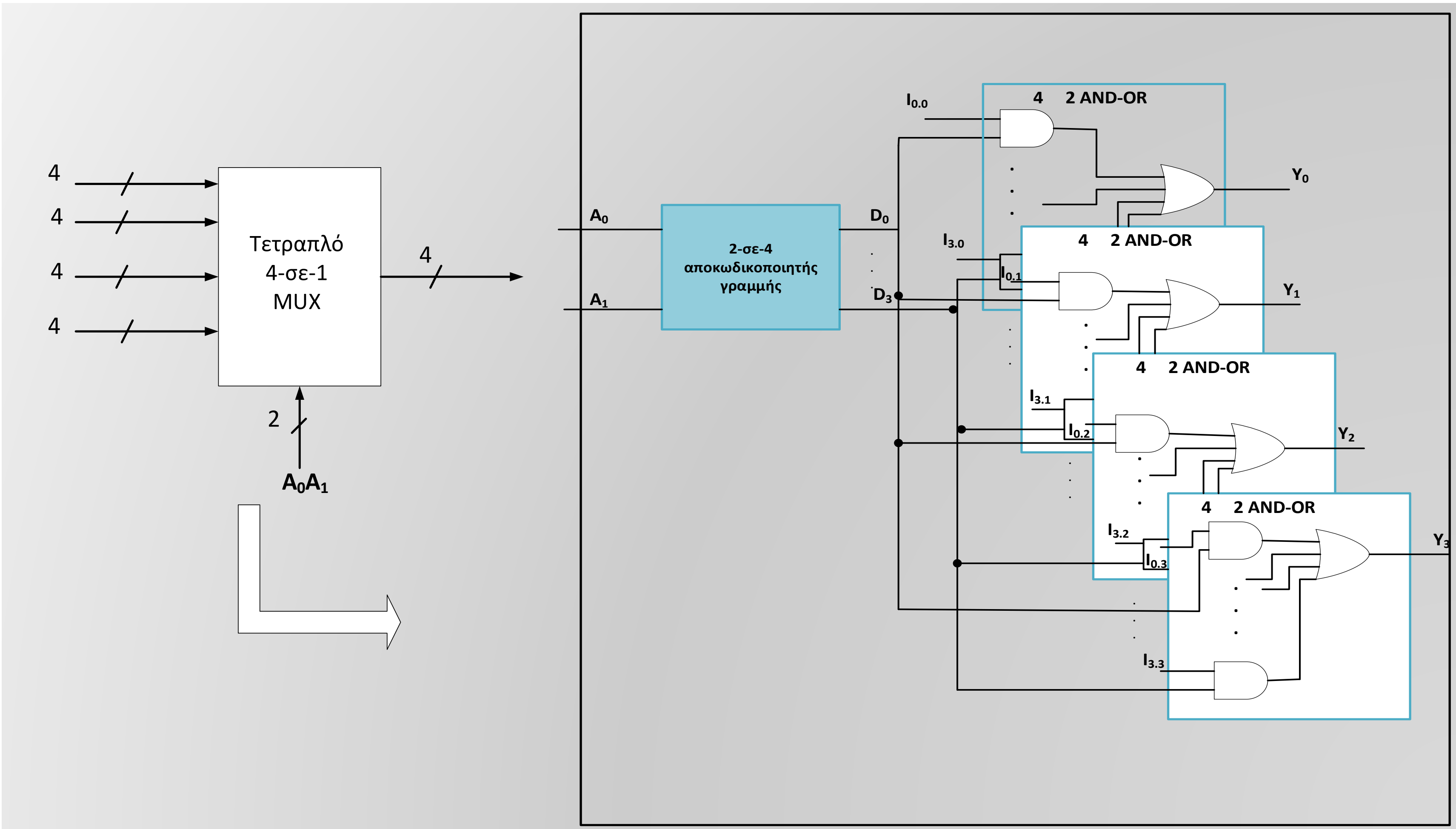
Παράδειγμα 8 bit 2-to-1 MUX



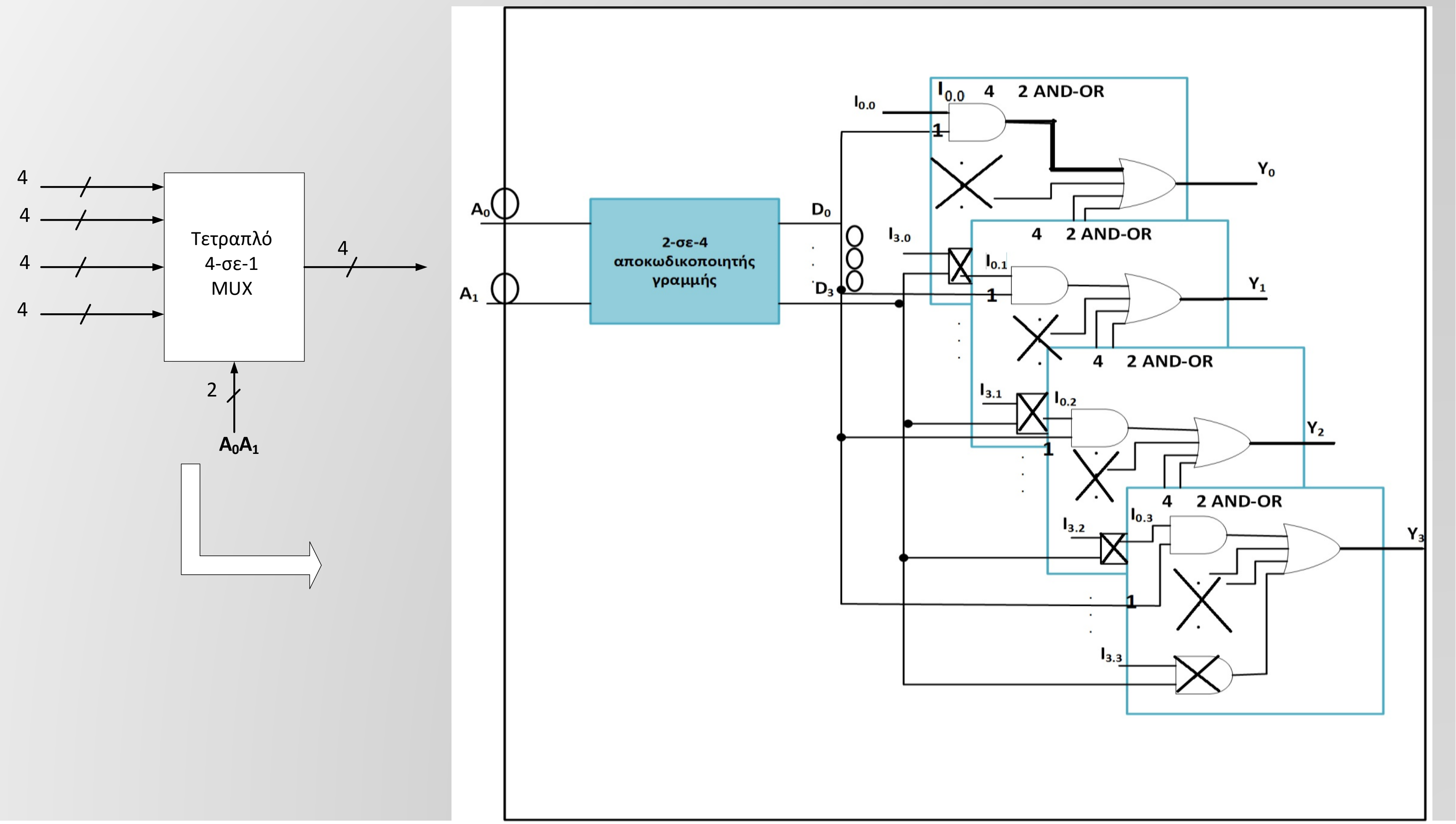
Παράδειγμα 4 bit 4-to-1 MUX (1)



Παράδειγμα 4 bit 4-to-1 MUX (2)



Παράδειγμα 4 bit 4-to-1 MUX (3)



Υλοποίηση συναρτήσεων με Boole πολυπλέκτες

- Οποιαδήποτε συνάρτηση Boole n μεταβλητών μπορεί να υλοποιηθεί χρησιμοποιώντας έναν πολυπλέκτη μεγέθους 2^{n-1}-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 ης συνάρτησης.
- Τα σήματα ΔΕΔΟΜΕΝΩΝ καθορίζουν τους ελαχιστόρους που οδηγούν στην πύλη OR.

Υλοποίηση συναρτήσεων με MUX

- Για μια συνάρτηση n-μεταβητών (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 τη 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 ξέυγη γειτονικών γραμμών στον πίνακα (μόνο το LSB διαφέρει, π.χ. D = 0 και 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).

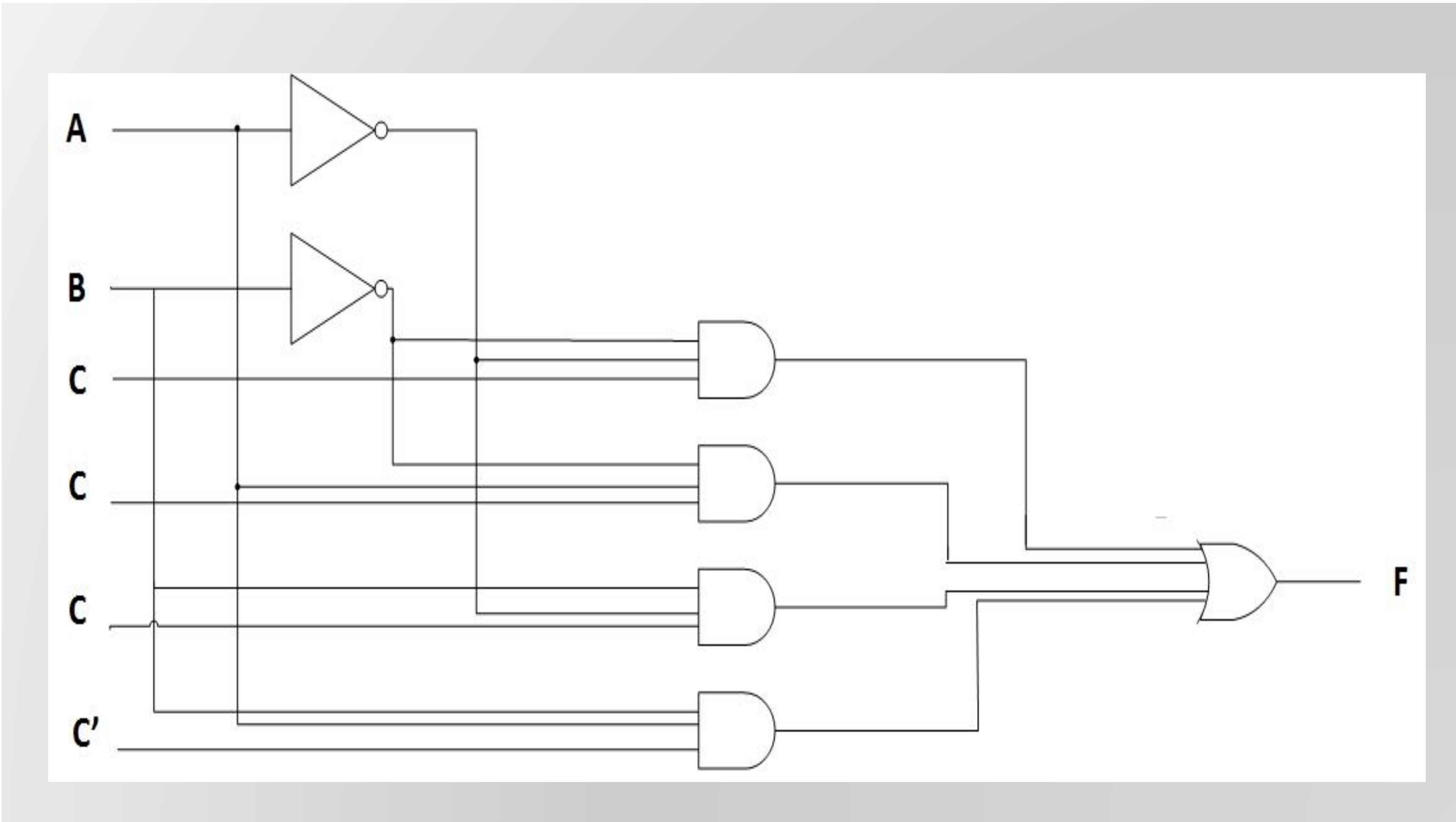
Παράδειγμα (3)

- Θεωρήστε $F(A, B, C) = \sum m(1, 3, 5, 6)$.
- 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 (4)

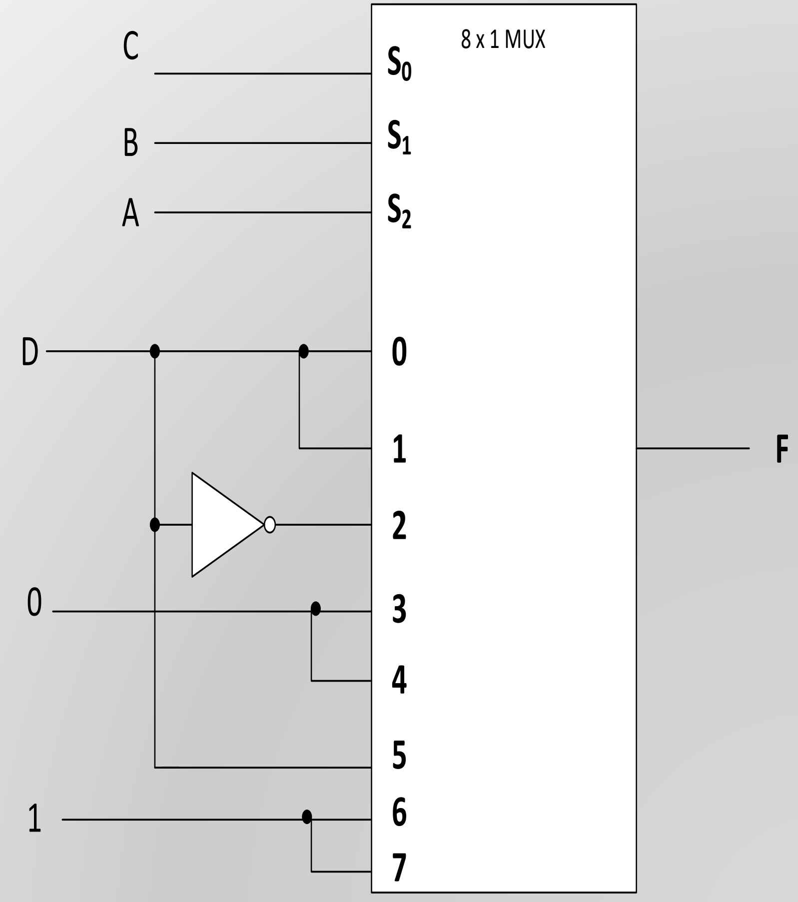
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Παράδειγμα (5)



Παράδειγμα (6)

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 Gray σε δυαδικό

- 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 πολθπλοκες.

Gray A B C	Binary A B C
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

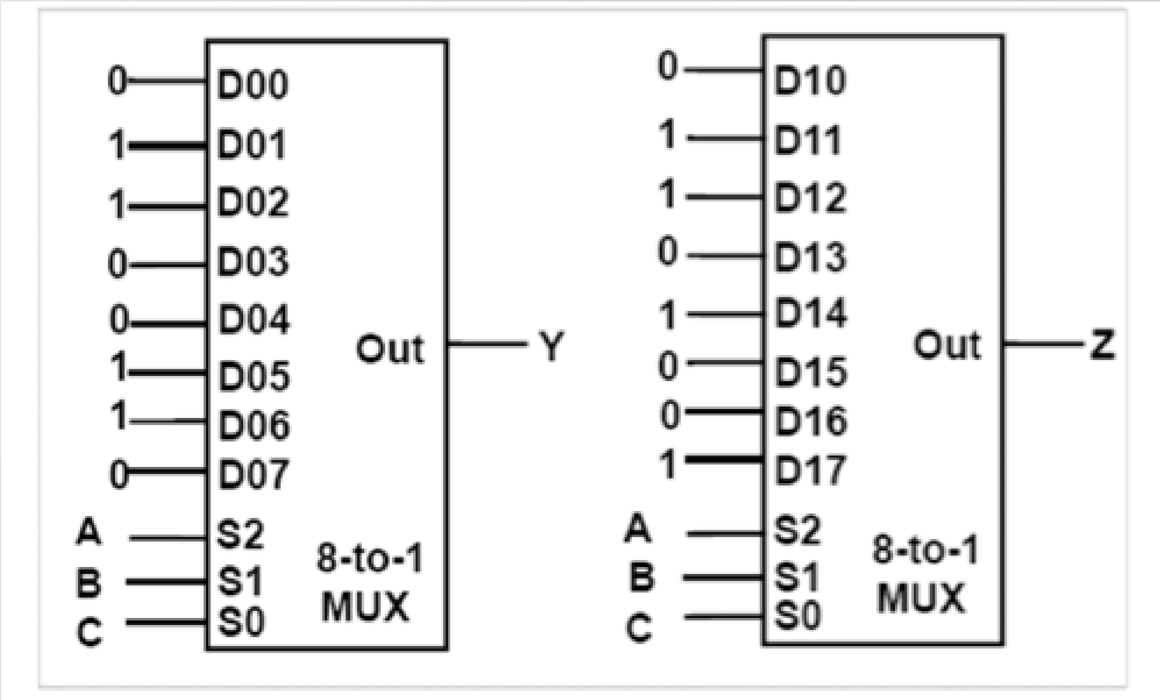
Gray σε δυαδικό (1η λύση) (1)

- Αναδιατάξτε τον πίνακα, έτσι ώστε οι διάφοροι συνδυασμοί εισόδων να είναι σε σειρά (000, 001, ..., 111).
- Οι συναρτήσεις y και z μπορούν να υλοποιηθούν με ένα διπλό (2-bit) 8-σε-1 MUX:
  - Οι A, B, 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 A B C
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) (2)

- 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! → 2^3 x 2 ROM.



Gray σε δυαδικό (2η λύση) (1)

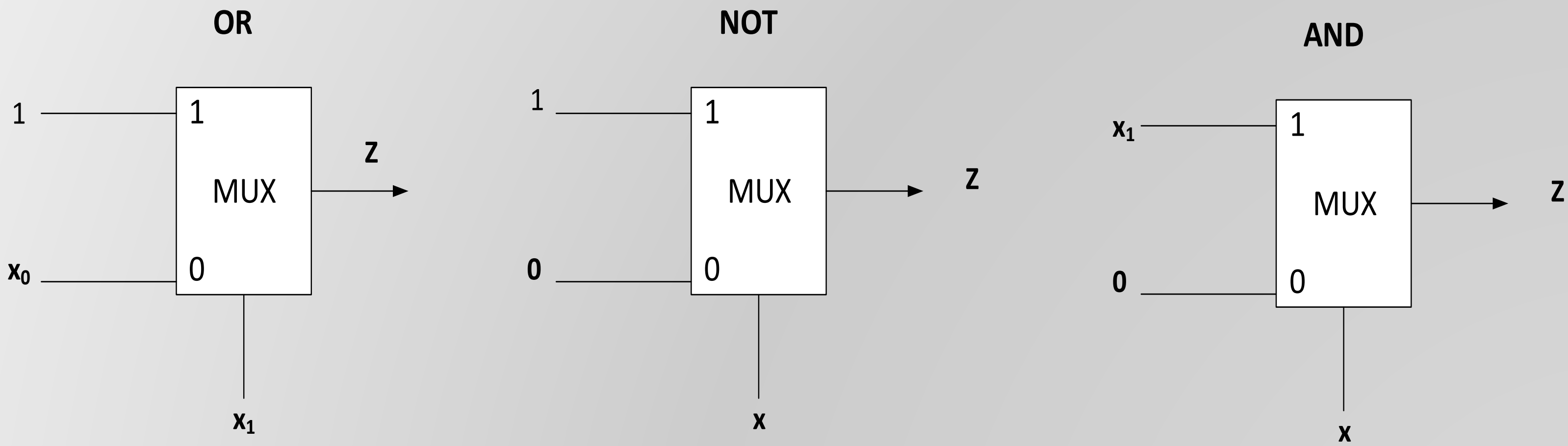
Gray A B C	Binary x y z	Στοιχειώδης συνάρτηση του C για y	Στοιχειώδης συνάρτηση του C για z
0 0 0	0 0 0	F = C	F = C
0 0 1	1 1 1	F = C	F = C
0 1 0	0 1 1	F = C	F = C
0 11 1	1 0 0	F = C	F = C
1 0 0	0 0 1	F = C	F = C
1 0 1	1 1 0	F = C	F = C
1 1 0	0 1 0	F = C	F = C
1 1 1	1 0 1	F = C	F = C

Gray σε δυαδικό (2η λύση) (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 λειτουρίες OR, AND και NOT μόνο με 2-σε-1 MUX. Άρα η 2-to-1 MUX είναι οικουμενική πύλη.



$z = x_1 + x_1\, x_0$
$= x_1x_0' + x_1x_0 + x_1'x_0$

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

$z = x_1\, x_0 + 0x_0' = x_1x_0$

Demultiplexers (DeMUX) - Αποπολυπλέκτες

- Εκτελεί το αντίστροφο της λειτουργίας του πολυπλέκτη:
  - Δέχεται δεδομένα από μία είσοδο και τα μεταβιβάζει σε συγκεκριμένη έξοδο, από τις δύο πιθανές που υπάρχουν.
  - Η επιλογή εξόδου γίνετε 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).

Τέλος Ενότητας