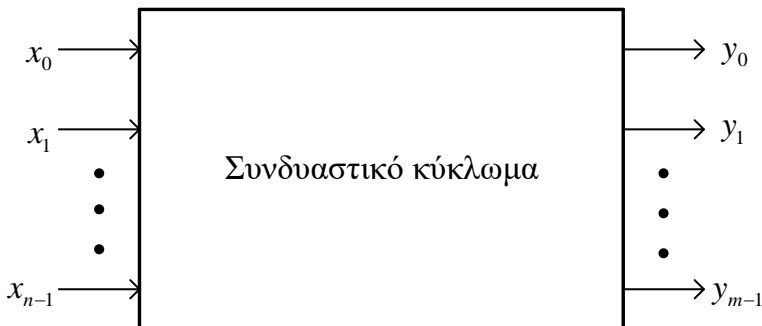


## ΣΥΝΔΥΑΣΤΙΚΑ ΚΥΚΛΩΜΑΤΑ

### 5.1 Εισαγωγή

Τα λογικά κυκλώματα διακρίνονται σε συνδυαστικά ή ακολουθιακά. Ένα λογικό κύκλωμα με  $n$  γραμμές εισόδου  $x_0, x_1, \dots, x_{n-1}$  και  $m$  γραμμές εξόδου  $y_0, y_1, \dots, y_{m-1}$ , όπως αυτό του σχήματος 5.1, ονομάζεται *συνδυαστικό (combinational)* εάν οι τιμές των εξόδων του  $y_i$  εξαρτώνται αποκλειστικά από τις παρούσες τιμές των εισόδων του  $x_j$  και εκφράζονται με λογικές παραστάσεις του τύπου  $y_i = f_i(x_0, x_1, \dots, x_{n-1})$  όπως αυτές που εξετάστηκαν σε προηγούμενο κεφάλαιο. Τα ακολουθιακά κυκλώματα θα μελετηθούν σε επόμενα κεφάλαια.

Για τα συνδυαστικά κυκλώματα έχουν αναπτυχθεί μέθοδοι ανάλυσης και σύνθεσης οι οποίες περιγράφονται στη συνέχεια. Αρκετά συνδυαστικά κυκλώματα χρησιμοποιούνται σαν δομικά στοιχεία για τη σχεδίαση σύνθετων ψηφιακών συστημάτων. Στο κεφάλαιο αυτό θα γίνει επίσης παρουσίαση των πιο διαδεδομένων από αυτά.



Σχήμα 5.1. Συνδυαστικό κύκλωμα

## 5.2 Ανάλυση Συνδυαστικών Κυκλωμάτων

*Ανάλυση* ενός συνδυαστικού κυκλώματος είναι ο προσδιορισμός των λογικών παραστάσεων που υλοποιούνται στις εξόδους του, καθώς και των αντίστοιχων πινάκων αληθείας. Η ανάλυση των συνδυαστικών κυκλωμάτων είναι χρήσιμη στην κατανόηση και την επαλήθευση της λειτουργίας τους και γίνεται με τα βήματα που περιγράφονται στη συνέχεια.

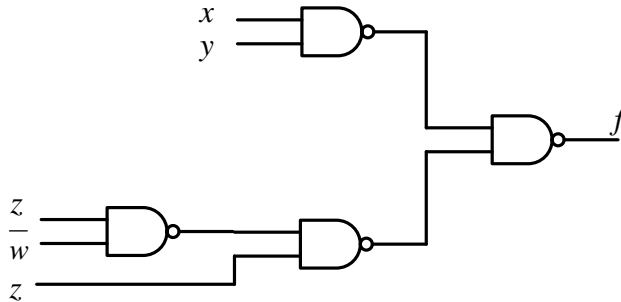
1. Δίδονται συμβολικά ονόματα σε όλες τις εξόδους των πυλών.
2. Γράφονται λογικές εκφράσεις για τις εξόδους των λογικών πυλών.
3. Αντικαθίστανται οι εσωτερικές ονομασίες των εξόδων των πυλών μέχρι να προκύψουν λογικές παραστάσεις που να περιλαμβάνουν μόνο τις εισόδους.

Αυτό το βήμα μπορεί να γίνει με δύο τρόπους. Αρχίζοντας από τις πύλες που είναι κοντά στις εισόδους ή αρχίζοντας από τις πύλες που είναι κοντά στις εξόδους. Συνήθως χρησιμοποιείται ο δεύτερος τρόπος.

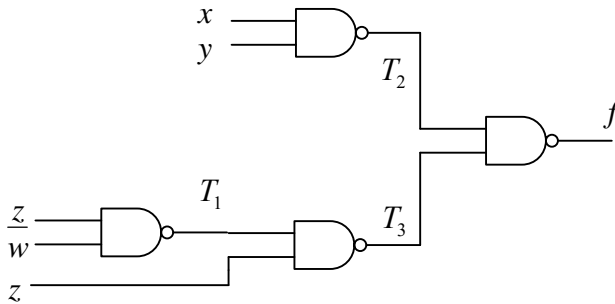
4. Μετατρέπεται κάθε λογική παράσταση που προέκυψε σε ισοδύναμη κανονική και προσδιορίζεται ο αντίστοιχος πίνακας αληθείας.

Στη συνέχεια δίδεται ένα παράδειγμα ανάλυσης ενός απλού συνδυαστικού κυκλώματος.

**Παράδειγμα 5.1.** Να αναλυθεί το λογικό κύκλωμα που δίδεται στη συνέχεια



Αρχικά δίδονται συμβολικά ονόματα στις εξόδους των λογικών πυλών όπως στο επόμενο σχήμα.



Για την έξοδο  $f$  του δοσμένου κυκλώματος ισχύει

$$\begin{aligned}
 f &= \overline{T_2 T_3} = \overline{T_2} + \overline{T_3} = \overline{xy} + \overline{T_1 z} = \\
 &= xy + T_1 z = xy + (\overline{z\bar{w}})z = xy + (\bar{z} + w)z .
 \end{aligned}$$

Τελικά,

$$f = xy + zw$$

Στη συνέχεια η λογική παράσταση που προέκυψε μετατρέπεται στην κανονική της μορφή σαν άθροισμα ελαχιστόρων.

$$\begin{aligned}
 f &= xy(z + \bar{z})(w + \bar{w}) + (x + \bar{x})(y + \bar{y})zw \\
 &= xyzw + xy\bar{z}w + xyz\bar{w} + xy\bar{z}\bar{w} + x\bar{y}zw + \bar{x}yzw + x\bar{y}\bar{z}w + \bar{x}\bar{y}zw \\
 &= \underbrace{xyzw}_{13} + \underbrace{xy\bar{z}w}_{14} + \underbrace{xyz\bar{w}}_{12} + \underbrace{xy\bar{z}\bar{w}}_{15} + \underbrace{x\bar{y}zw}_{7} + \underbrace{\bar{x}yzw}_{11} + \underbrace{x\bar{y}\bar{z}w}_{3}
 \end{aligned}$$

Επομένως η κανονική παράσταση της λογικής συνάρτησης  $f$  είναι

$$f = \sum m(3, 7, 11, 12, 13, 14, 15)$$

Από την κανονική παράσταση προκύπτει ο πίνακας αληθείας της λογικής συνάρτησης  $f$  που δίδεται στην συνέχεια.

$x$	$y$	$z$	$w$	$f$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

### 5.3 Σύνθεση Συνδυαστικών Κυκλωμάτων

Η *σύνθεση (synthesis)* ή *σχεδίαση* συνδυαστικών κυκλωμάτων γίνεται με τα πιο κάτω βήματα:

1. Αρχικά γίνεται λειτουργική περιγραφή του κυκλώματος και προσδιορίζονται οι μεταβλητές εισόδου και εξόδου.
2. Κατασκευάζεται ο πίνακας αληθείας ή προσδιορίζεται μία λογική παράσταση για κάθε έξοδο.
3. Προσδιορίζονται οι απλούστερες λογικές παραστάσεις για κάθε έξοδο.
4. Σχεδιάζονται τα λογικά κυκλώματα που αντιστοιχούν στις απλοποιημένες λογικές παραστάσεις που αντιστοιχούν στις εξόδους.

**Παράδειγμα 5.2.** Να σχεδιασθεί ψηφιακό κύκλωμα ένδειξης μηδενικής τιμής για προσημασμένους αριθμούς των τριών bit. Ένδειξη μηδενικής τιμής να θεωρηθεί το λογικό 1.

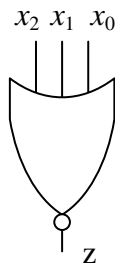
Έστω  $x_2, x_1, x_0$  οι είσοδοι του κυκλώματος και  $z$  η ένδειξη μηδενικής τιμής. Η έξοδος  $z$  θα γίνεται 1 όταν όλες οι είσοδοι έχουν την τιμή 0. Ο πίνακας αληθείας του κυκλώματος δίδεται στην συνέχεια.

$x_2$	$x_1$	$x_0$	$z$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Από τον πίνακα αληθείας προκύπτει ότι

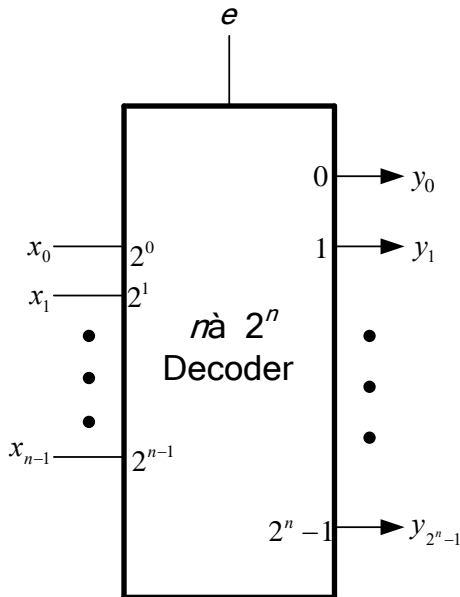
$$z = \overline{x_2} \overline{x_1} \overline{x_0} = x_2 + x_1 + x_0$$

Η υλοποίηση του  $z$  δίδεται στην συνέχεια



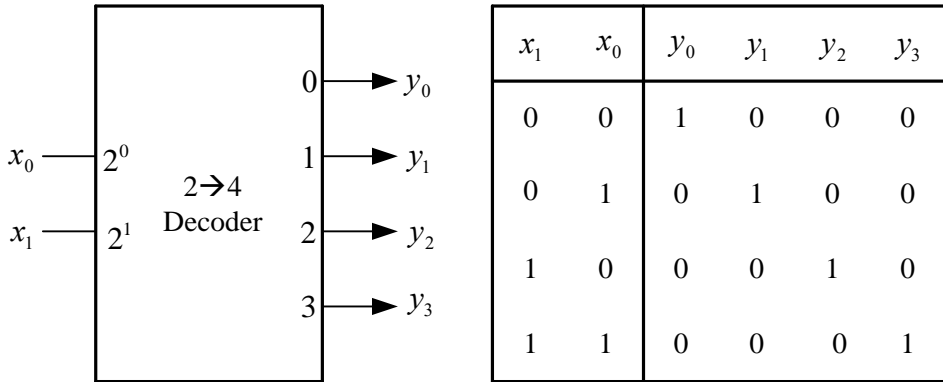
### 5.4 Αποκωδικοποιητές

Αποκωδικοποιητής (*decoder*)  $n$  εισόδων, ή  $n \rightarrow 2^n$  αποκωδικοποιητής είναι ένα συνδυαστικό κύκλωμα που έχει  $n$  εισόδους  $x_{n-1}, \dots, x_1, x_0$  και  $2^n$  εξόδους  $y_{2^n-1}, \dots, y_1, y_0$ . Η έξοδος  $y_i$  γίνεται 1 ( $y_i=1$ ) όταν για τις εισόδους ισχύει  $x_{n-1}2^{n-1} + \dots + x_12 + x_0 = i$ . Σε δεδομένη χρονική στιγμή μία και μόνο έξοδος  $y_i$  θα έχει την τιμή 1 και όλες οι άλλες εξοδοί έχουν την τιμή 0. Συνήθως υπάρχει και μία επιπλέον είσοδος ενεργοποίησης (*enable*)  $e$ . Όταν  $e=0$  όλες οι εξοδοί του αποκωδικοποιητή έχουν την τιμή 0, ενώ όταν  $e=1$  ο αποκωδικοποιητής λειτουργεί κανονικά. Στο σχήμα 5.2 δίδεται το λογικό σύμβολο ενός αποκωδικοποιητή  $n$  εισόδων με είσοδο ενεργοποίησης  $e$ . Κάθε έξοδος ενός  $n \rightarrow 2^n$  αποκωδικοποιητή αντιστοιχεί σε μία λογική συνάρτηση που έχει την τιμή 1 για συγκεκριμένο συνδυασμό των εισόδων, δηλαδή κάθε έξοδος υλοποιεί έναν ελαχιστόρο των μεταβλητών εισόδου. Επομένως  $y_i = m_i(x_{n-1}, \dots, x_1, x_0)$  ή  $y_i = e \cdot m_i(x_{n-1}, \dots, x_1, x_0)$  για αποκωδικοποιητή με είσοδο ενεργοποίησης.



**Σχήμα 5.2.** Λογικό σύμβολο αποκωδικοποιητή  $n$  εισόδων με είσοδο ενεργοποίησης

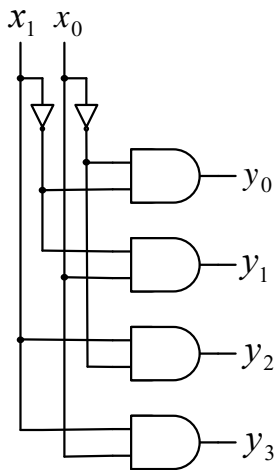
**Παράδειγμα 5.3.** Στη συνέχεια δίδεται το λογικό σύμβολο και ο πίνακας αληθείας ενός  $2 \rightarrow 4$  αποκωδικοποιητή.



Οι λογικές παραστάσεις των εξόδων του  $2 \rightarrow 4$  αποκωδικοποιητή που προκύπτουν από τον πίνακα αληθείας είναι οι εξής:

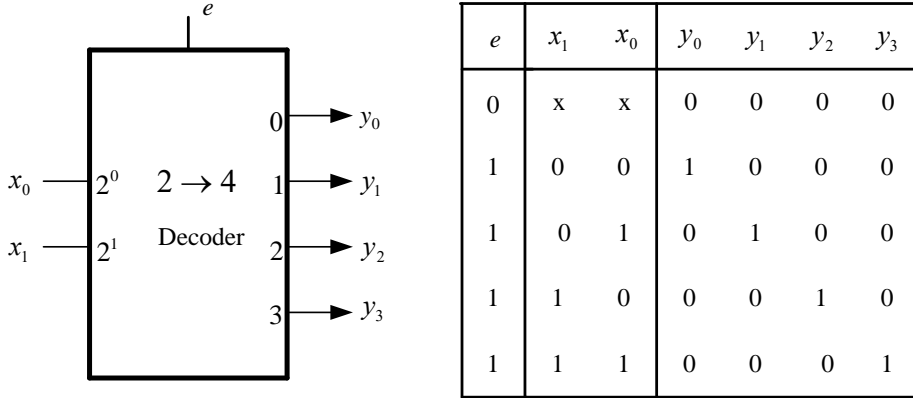
$$y_0 = \bar{x}_1 \cdot \bar{x}_0, y_1 = \bar{x}_1 \cdot x_0, y_2 = x_1 \cdot \bar{x}_0, y_3 = x_1 \cdot x_0$$

Η υλοποίηση του  $2 \rightarrow 4$  αποκωδικοποιητή σύμφωνα με τις πιο πάνω λογικές παραστάσεις δίδεται στο σχήμα 5.3.



**Σχήμα 5.3.** Σχεδίαση  $2 \rightarrow 4$  αποκωδικοποιητή

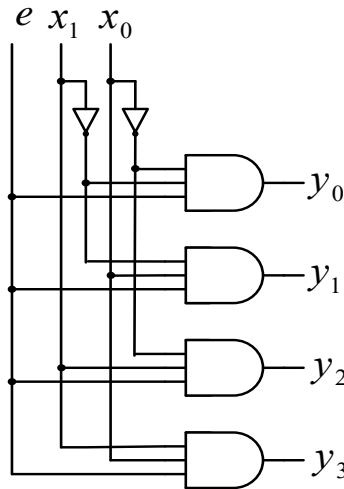
**Παράδειγμα 5.4.** Στη συνέχεια δίδεται το λογικό σύμβολο και ο πίνακας αληθείας του 2→4 αποκωδικοποιητή με είσοδο ενεργοποίησης  $e$ .



Οι λογικές παραστάσεις των εξόδων του 2→4 αποκωδικοποιητή με είσοδο που προκύπτουν από τον πίνακα αληθείας είναι οι εξής

$$y_0 = \bar{x}_1 \cdot \bar{x}_0 \cdot e, \quad y_1 = \bar{x}_1 \cdot x_0 \cdot e, \quad y_2 = x_1 \cdot \bar{x}_0 \cdot e, \quad y_3 = x_1 \cdot x_0 \cdot e$$

Η υλοποίηση του 2→4 αποκωδικοποιητή με είσοδο ενεργοποίησης σύμφωνα με τις πιο πάνω παραστάσεις δίνεται στο σχήμα 5.4.



**Σχήμα 5.4.** Σχεδίαση 2→4 αποκωδικοποιητή με είσοδο ενεργοποίησης



### 5.4.1 Δίκτυα αποκωδικοποιητών

Λόγω των περιορισμών που εισάγει η τεχνολογία υλοποίησης στον μέγιστο αριθμό των εισόδων που μπορούν να έχουν οι λογικές πύλες, αρχικά υλοποιούνται αποκωδικοποιητές με μικρό αριθμό εισόδων. Μεγαλύτεροι αποκωδικοποιητές υλοποιούνται συνδέοντας αποκωδικοποιητές με μικρότερο αριθμό εισόδων. Οι υλοποιήσεις αυτές ονομάζονται *δίκτυα αποκωδικοποιητών* και διακρίνονται σε δέντρα αποκωδικοποιητών και δισδιάστατες υλοποιήσεις.

#### Δέντρα αποκωδικοποιητών

Μια προσέγγιση στο να υλοποιηθούν μεγάλοι αποκωδικοποιητές είναι να σχηματισθούν δένδρα μικρότερων αποκωδικοποιητών. Η μεθοδολογία αυτή επεξηγείται με τα επόμενα παραδείγματα.

**Παράδειγμα 5.5.** Να σχεδιασθεί ένας  $3 \rightarrow 8$  αποκωδικοποιητής με είσοδο επίτρεψης χρησιμοποιώντας σαν στοιχείο σχεδίασης τον  $2 \rightarrow 4$  αποκωδικοποιητή με είσοδο επίτρεψης.

Το ζητούμενο κύκλωμα του  $3 \rightarrow 8$  αποκωδικοποιητή με είσοδο επίτρεψης  $e$ , εισόδους επιλογής  $x_2, x_1, x_0$  και εξόδους  $z_0, z_1, \dots, z_7$  θα έχει τον πίνακα αλήθειας που δίδεται στο σχήμα 5.5.

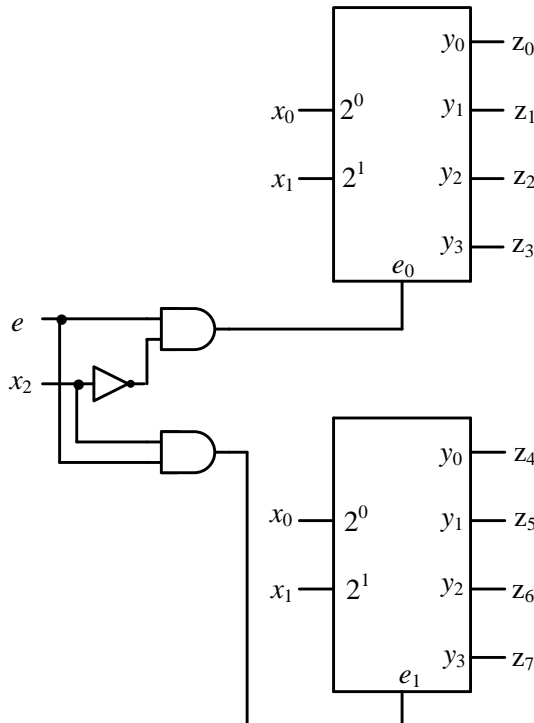
$e$	$x_2$	$x_1$	$x_0$	$z_0$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$	$z_7$
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

Σχήμα 5.5. Πίνακας αλήθειας του  $3 \rightarrow 8$  αποκωδικοποιητή με είσοδο επίτρεψης

Σύμφωνα με τον πιο πάνω πίνακα αληθείας οι έξοδοι που αντιστοιχούν σε κάθε τετράγωνο υλοποιούνται με έναν  $2 \rightarrow 4$  αποκωδικοποιητή εισόδους τις  $x_1, x_0$ . Η είσοδος επίτρεψης  $e_i$  του κάθε  $2 \rightarrow 4$  αποκωδικοποιητή ενεργοποιείται από τον κατάλληλο συνδυασμό των εισόδων  $e, x_2$ . Η λειτουργία αυτή περιγράφεται στον επόμενο πίνακα.

$e$	$x_2$	$e_0$	$e_1$
0	x	0	0
1	0	1	0
1	1	0	1

Επομένως οι είσοδοι επίτρεψης των  $2 \rightarrow 4$  αποκωδικοποιητών είναι  $e_0 = e \cdot \bar{x}_2$ ,  $e_1 = e \cdot x_2$ . Το κύκλωμα που προκύπτει δίδεται στο σχήμα 5.6.



**Σχήμα 5.6.** Σχεδίαση  $3 \rightarrow 8$  αποκωδικοποιητή με  $2 \rightarrow 4$  αποκωδικοποιητές

**Παράδειγμα 5.6.** Να σχεδιασθεί ένας  $4 \rightarrow 16$  αποκωδικοποιητής με είσοδο επίτρεψης χρησιμοποιώντας σαν στοιχείο σχεδίασης τον  $2 \rightarrow 4$  αποκωδικοποιητή με είσοδο επίτρεψης.

Το ζητούμενο κύκλωμα του  $4 \rightarrow 16$  αποκωδικοποιητή με είσοδο επίτρεψης  $e$ , εισόδους επιλογής  $x_3, x_2, x_1, x_0$  και εξόδους  $z_0, z_1, \dots, z_{15}$  θα έχει τον πίνακα αλήθειας που δίδεται στο σχήμα 5.7.

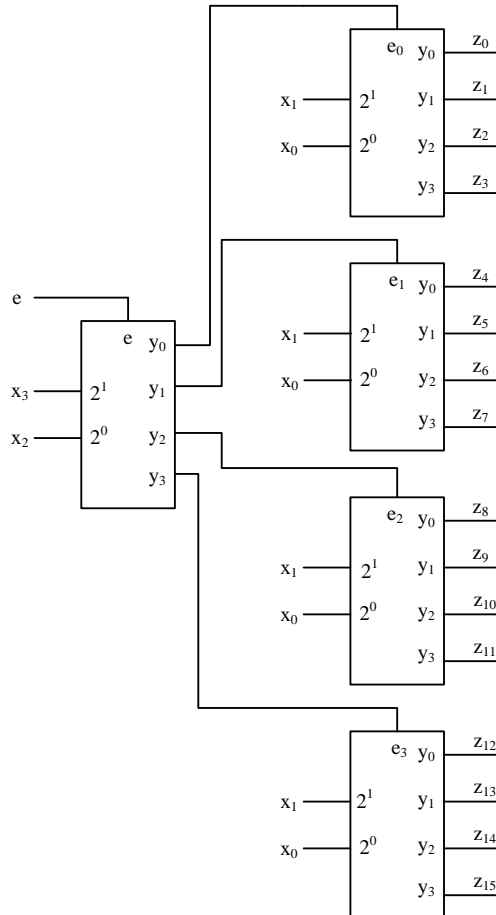
e	$x_3$	$x_2$	$x_1$	$x_0$	$z_0$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$	$z_7$	$z_8$	$z_9$	$z_{10}$	$z_{11}$	$z_{12}$	$z_{13}$	$z_{14}$	$z_{15}$
0	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

**Σχήμα 5.7.** Πίνακας αληθείας  $4 \rightarrow 16$  αποκωδικοποιητή με είσοδο επίτρεψης

Σύμφωνα με τον πιο πάνω πίνακα αληθείας οι έξοδοι που αντιστοιχούν σε κάθε τετράγωνο υλοποιούνται με έναν  $2 \rightarrow 4$  αποκωδικοποιητή με εισόδους τις  $x_1, x_0$ . Η είσοδος επίτρεψης  $e_i$  του κάθε  $2 \rightarrow 4$  αποκωδικοποιητή ενεργοποιείται από τον κατάλληλο συνδυασμό των εισόδων  $e, x_3, x_2$ . Η λειτουργία αυτή περιγράφεται στον πίνακα που δίδεται στη συνέχεια:

$e$	$x_3$	$x_2$	$e_0$	$e_1$	$e_2$	$e_3$
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

Επομένως για την οδήγηση των εισόδων επίτρεψης των αποκωδικοποιητών με εισόδους τις  $x_1, x_0$  μπορεί να χρησιμοποιηθεί ένας αποκωδικοποιητής  $2 \rightarrow 4$  με εισόδους επιλογής τις  $x_3, x_2$ . Το ζητούμενο κύκλωμα δίδεται στο σχήμα 5.8.



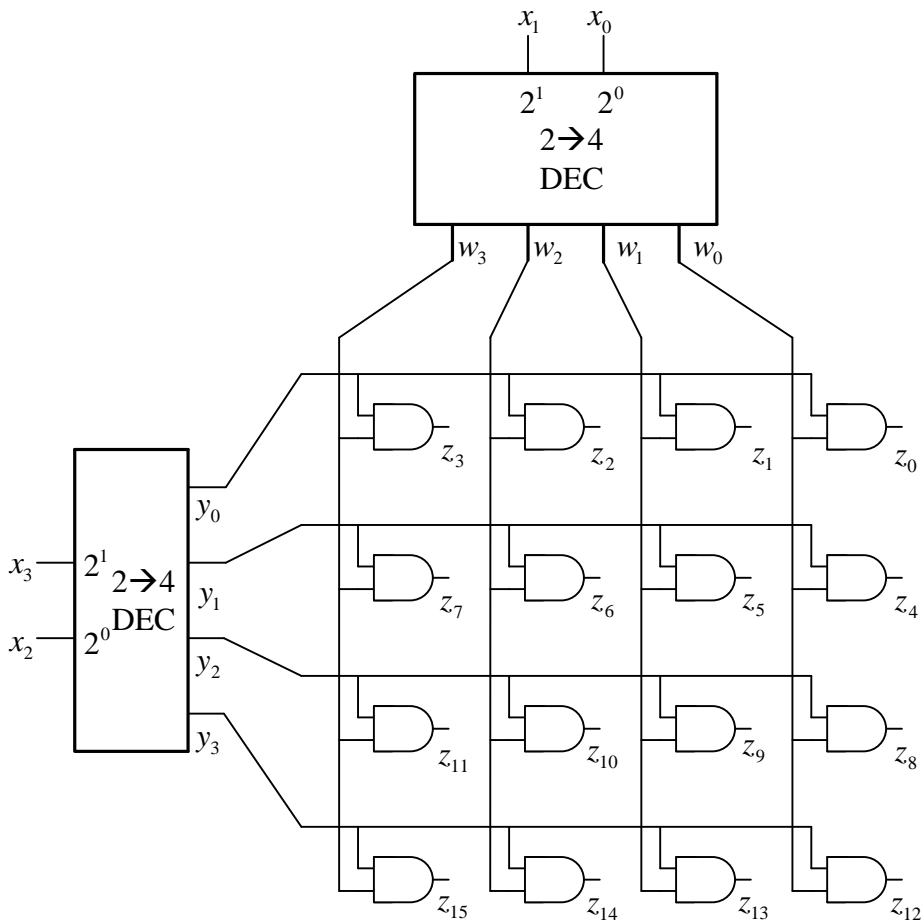
Σχήμα 5.8. Σχεδίαση  $4 \rightarrow 16$  αποκωδικοποιητή με  $2 \rightarrow 4$  αποκωδικοποιητές

### Δυσδιάστατοι Αποκωδικοποιητές

Με την δυσδιάστατη αποκωδικοποίηση ένας  $n \rightarrow 2^n$  αποκωδικοποιητής σχεδιάζεται χρησιμοποιώντας δύο  $n/2 \rightarrow 2^{n/2}$  αποκωδικοποιητές και  $2^n$  πύλες AND. Η μέθοδος αυτή επεξηγείται με το επόμενο παράδειγμα.

**Παράδειγμα 5.7.** Να γίνει δυσδιάστατη υλοποίηση ενός  $4 \rightarrow 16$  αποκωδικοποιητή με  $2 \rightarrow 4$  αποκωδικοποιητές και πύλες AND.

Έστω  $x_0, x_1, x_2, x_3$  οι είσοδοι του  $4 \rightarrow 16$  αποκωδικοποιητή και  $z_0, z_1, \dots, z_{15}$  οι έξοδοί του. Για τη δυσδιάστατη σχεδίαση θα χρησιμοποιηθούν δύο  $2 \rightarrow 4$  αποκωδικοποιητές και 16 πύλες AND. Η δυσδιάστατη υλοποίηση του  $4 \rightarrow 16$  αποκωδικοποιητή δίδεται στο σχήμα 5.9.



Σχήμα 5.9. Δυσδιάστατη υλοποίηση του  $4 \rightarrow 16$  αποκωδικοποιητή

Ο ένας  $2 \rightarrow 4$  αποκωδικοποιητής αποκωδικοποιεί τις εισόδους  $x_0, x_1$  και έχει εξόδους τις  $w_0, w_1, w_2, w_3$ , ενώ ο άλλος τις εισόδους  $x_2, x_3$  και έχει εξόδους τις  $y_0, y_1, y_2, y_3$ . Τα ζεύγη που συνίστανται από τις εξόδους των αποκωδικοποιητών γίνονται εισοδοί σε πύλες AND. Οι έξοδοι των πυλών AND αντιστοιχούν στις εξόδους του  $4 \rightarrow 16$  αποκωδικοποιητή. Οι δείκτες των μεταβλητών των εξόδων προκύπτουν όπως στη συνέχεια. Έστω  $z_i$  η έξοδος της πύλης που συνδέει τις εξόδους  $w_j, y_k$  των  $2 \rightarrow 4$  αποκωδικοποιητών. Ο δείκτης  $i$  προκύπτει σύμφωνα με τον τύπο  $i = k \cdot 2^2 + j$ . Έστω ότι μία πύλη AND συνδέει τις εξόδους  $w_1$  και  $y_2$ . Ισχύει  $i = 2 \cdot 2^2 + 1 = 9$ . Δηλαδή η έξοδος της πύλης θα αντιστοιχεί στην έξοδο  $z_9$  του  $4 \rightarrow 16$  αποκωδικοποιητή.

#### 5.4.2 Σχεδίαση συνδυαστικών κυκλωμάτων με αποκωδικοποιητές και πύλες OR

Όπως προαναφέραμε κάθε έξοδος του  $n \rightarrow 2^n$  αποκωδικοποιητή υλοποιεί έναν ελαχιστόρο για τις μεταβλητές που εφαρμόζονται στις εισόδους του. Επομένως κάθε λογική συνάρτηση  $n$  μεταβλητών μπορεί να υλοποιηθεί με έναν  $n \rightarrow 2^n$  αποκωδικοποιητή και μία πύλη OR. Η υλοποίηση της συνάρτησης προκύπτει εφαρμόζοντας σαν εισόδους στον αποκωδικοποιητή τις μεταβλητές της συνάρτησης και κάνοντας OR στις εξόδους του αποκωδικοποιητή που αντιστοιχούν στους ελαχιστόρους που περιλαμβάνονται στην κανονική μορφή της συνάρτησης.

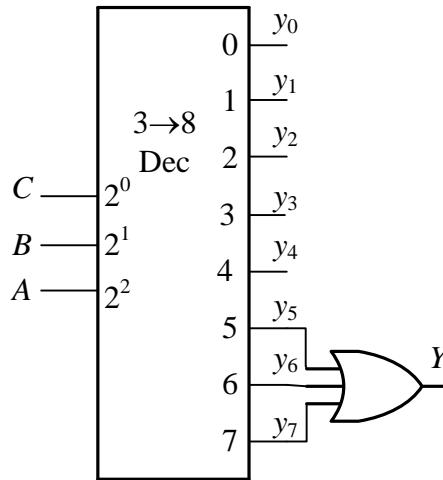
**Παράδειγμα 5.8.** Να υλοποιηθεί με χρήση αποκωδικοποιητή και μίας πύλης OR η λογική συνάρτηση που δίδεται στη συνέχεια.

$$Y = AB + AC$$

Αρχικά η παράσταση της λογικής συνάρτησης μετατρέπεται στην κανονική της μορφή σαν άθροισμα ελαχιστόρων.

$$\begin{aligned} Y &= AB + AC = AB(C + \bar{C}) + AC(B + \bar{B}) = ABC + AB\bar{C} + ABC + A\bar{B}C = \\ &= A\bar{B}\bar{C} + AB\bar{C} + ABC \quad \text{ή} \quad Y = \sum m(5,6,7) \end{aligned}$$

Επειδή η συνάρτηση είναι τριών μεταβλητών θα χρησιμοποιηθεί ένας  $3 \rightarrow 8$  αποκωδικοποιητής. Η υλοποίηση της συνάρτησης δίδεται στο σχήμα 5.10. Σύμφωνα με τα προηγούμενα, εάν ο αποκωδικοποιητής έχει σαν εισόδους τις μεταβλητές  $A, B, C$ , σαν εισόδους της πύλης OR θα χρησιμοποιηθούν οι εξόδοι  $y_5, y_6, y_7$  του αποκωδικοποιητή που υλοποιούν τους ελαχιστόρους  $m_5, m_6, m_7$ .

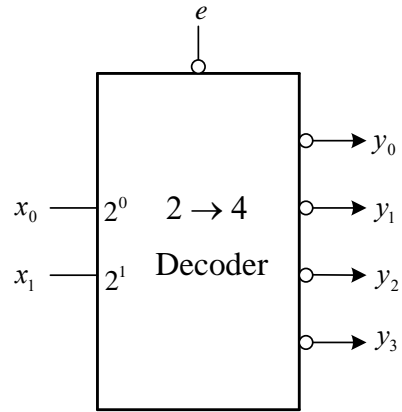


**Σχήμα 5.10.** Υλοποίηση της κανονικής παράστασης  $Y = \sum m(5,6,7)$  με  $3 \rightarrow 8$  αποκωδικοποιητή

### 5.4.3 Αποκωδικοποιητές με συμπληρωματικές εξόδους

Στην πράξη είναι πιο αποδοτικό να κατασκευάζονται οι αποκωδικοποιητές με πύλες NAND αντί για AND. Η επιλογή αυτή διαφοροποιεί τον πίνακα αληθείας του αποκωδικοποιητή στον οποίο οι τιμές των εξόδων αντιστρέφονται. Στο σχήμα 5.11 δίδεται ο πίνακας αληθείας και το λογικό σύμβολο ενός  $2 \rightarrow 4$  αποκωδικοποιητή που παράγει συμπληρωματικές εξόδους και λειτουργεί με συμπληρωματική είσοδο επίτρεψης. Στο ίδιο σχήμα δίδονται οι λογικές συναρτήσεις που υλοποιούνται στις εξόδους του και η υλοποίησή του με πύλες NAND και NOT.

$e$	$x_1$	$x_0$	$y_0$	$y_1$	$y_2$	$y_3$
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

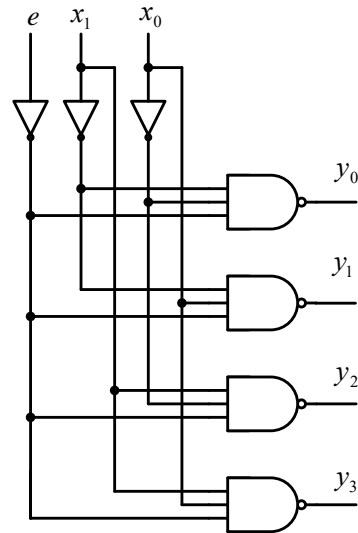


$$y_0 = e + x_1 + x_0 = \overline{\overline{e + x_1 + x_0}} = \overline{\overline{e} \cdot \overline{x_1} \cdot \overline{x_0}}$$

$$y_1 = e + x_1 + \overline{x_0} = \overline{\overline{e + x_1 + \overline{x_0}}} = \overline{\overline{e} \cdot \overline{x_1} \cdot x_0}$$

$$y_2 = e + \overline{x_1} + x_0 = \overline{\overline{e + \overline{x_1} + x_0}} = \overline{\overline{e} \cdot x_1 \cdot \overline{x_0}}$$

$$y_3 = e + \overline{x_1} + \overline{x_0} = \overline{\overline{e + \overline{x_1} + \overline{x_0}}} = \overline{\overline{e} \cdot x_1 \cdot x_0}$$



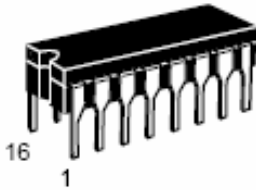
Σχήμα 5.11. Σχεδίαση αποκωδικοποιητή με συμπληρωματικές εξόδους

#### 5.4.4 Ολοκληρωμένα κυκλώματα αποκωδικοποιητών

Στον σχήμα 5.12 περιγράφεται σύντομα το ολοκληρωμένο κύκλωμα 74139 που περιέχει δύο 2→4 αποκωδικοποιητές με συμπληρωματικές εξόδους και είσοδο επίτρεψης.



ΟΛΟΚΛΗΡΩΜΕΝΑ  
ΚΥΚΛΩΜΑΤΑ



DIP

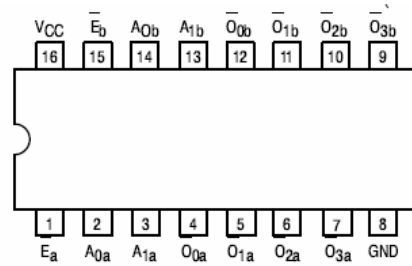


SMT

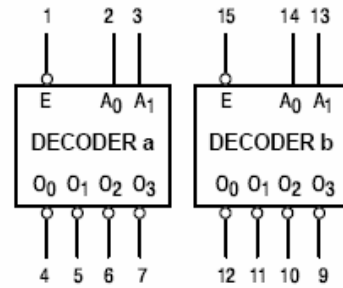
ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

INPUTS			OUTPUTS			
E	A <sub>0</sub>	A <sub>1</sub>	O <sub>0</sub>	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

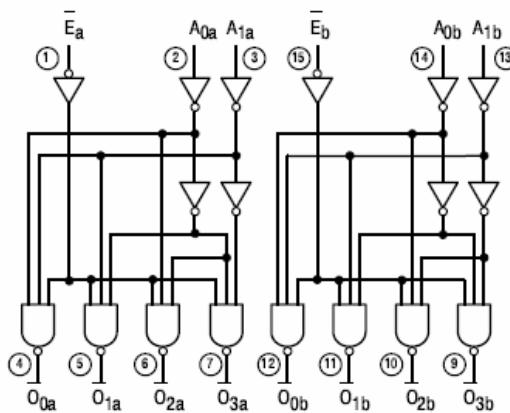
ΔΙΑΓΡΑΜΜΑ ΑΚΙΔΩΝ



ΛΕΙΤΟΥΡΓΙΚΟ ΔΙΑΓΡΑΜΜΑ  
ΑΚΙΔΩΝ



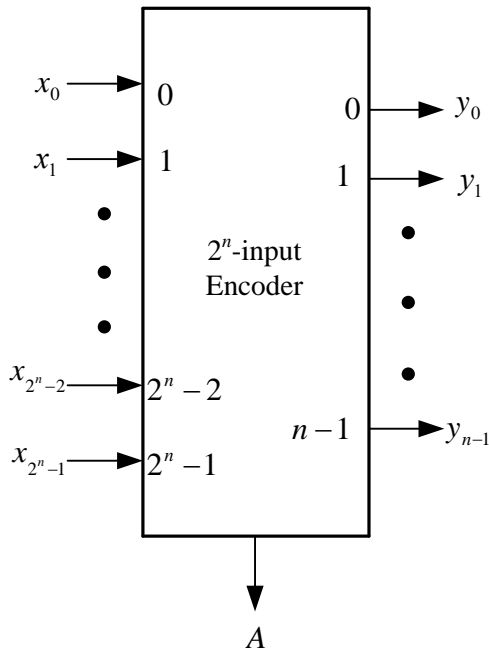
ΛΟΓΙΚΟ ΚΥΚΛΩΜΑ



Σχήμα 5.12. Ολοκληρωμένο κύκλωμα 74LS139

### 5.5 Κωδικοποιητές

*Κωδικοποιητής (encoder)* είναι ένα συνδυαστικό κύκλωμα με  $2^n$  δυαδικές εισόδους  $x = x_0, x_1, \dots, x_{2^n-1}$  και  $n$  εξόδους  $y = y_0, y_1, \dots, y_{n-1}$ . Εκτελεί την αντίστροφη λειτουργία από αυτή του αποκωδικοποιητή. Μετατρέπει τον 1-από- $2^n$  κώδικα στον δυαδικό κώδικα. Από τις εισόδους του κωδικοποιητή μία μόνο μπορεί να έχει την τιμή 1. Οι εξοδοί του κωδικοποιητή δίδουν στο δυαδικό σύστημα την τιμή του δείκτη της εισόδου που έχει την τιμή 1. Συνήθως υπάρχει και μία έξοδος ένδειξης λειτουργίας  $A$  που χρησιμοποιείται για τη δημιουργία δικτύων κωδικοποιητών. Η έξοδος  $A$  έχει την τιμή 1 εάν μία τουλάχιστον είσοδος έχει την τιμή 1. Στο σχήμα 5.13 δίδεται το λογικό σύμβολο ενός κωδικοποιητή  $2^n$  εισόδων.



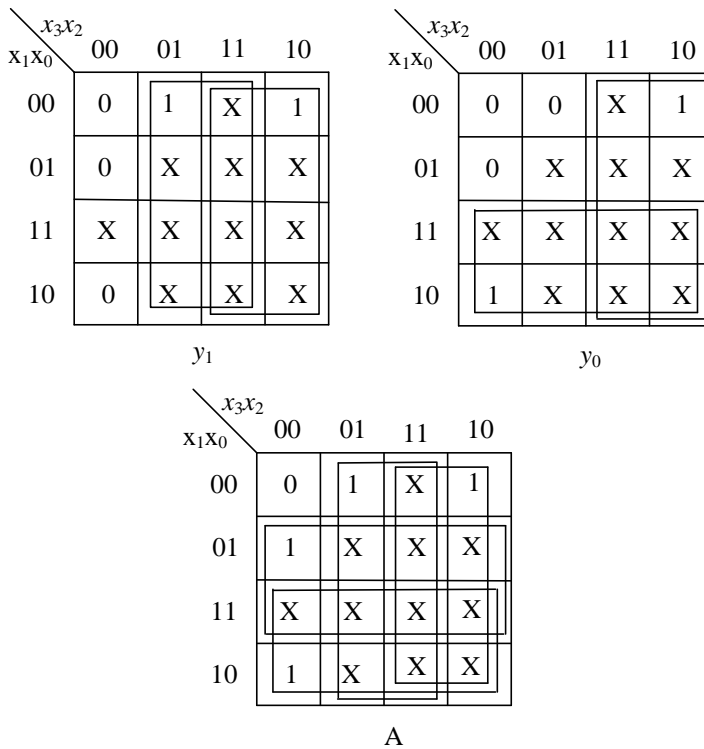
**Σχήμα 5.13.** Λογικό σύμβολο κωδικοποιητή  $2^n$  εισόδων

**Παράδειγμα 5.9.** Να σχεδιασθεί ένας κωδικοποιητής 4 εισόδων, δηλαδή ένας  $4 \rightarrow 2$  κωδικοποιητής.

Η λειτουργία του  $4 \rightarrow 2$  κωδικοποιητή περιγράφεται στον πίνακα αληθείας που δίδεται στη συνέχεια.

$x_3$	$x_2$	$x_1$	$x_0$	$y_1$	$y_0$	$A$
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1
0	0	0	0	0	0	0

Με βάση τον πιο πάνω πίνακα απεικονίζουμε τις τιμές των εξόδων  $y_1$ ,  $y_0$ ,  $A$  σε χάρτες Karnaugh όπως δείχνεται στο σχήμα 5.14. Επειδή μόνο μία είσοδος έχει την τιμή 1 για τους υπόλοιπους συνδυασμούς υποθέτουμε ότι μπορεί να είναι 0 ή 1 (X).



Σχήμα 5.14. Εξαγωγή απλοποιημένων παραστάσεων των εξόδων του 4→2 κωδικοποιητή

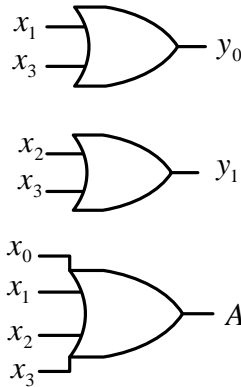
Από τους χάρτες Karnaugh προκύπτουν οι επόμενες απλοποιημένες παραστάσεις για τα  $y_0$ ,  $y_1$ ,  $A$ :

$$y_0 = x_1 + x_3$$

$$y_1 = x_2 + x_3$$

$$A = x_0 + x_1 + x_2 + x_3$$

Με βάση τις πιο πάνω λογικές παραστάσεις προκύπτει το κύκλωμα του αποκωδικοποιητή 4 εισόδων που δίδεται στο σχήμα 5.15.



**Σχήμα 5.15.** Σχεδίαση του 4→2 κωδικοποιητή

**Παράδειγμα 5.10.** Να περιγραφεί με πίνακα η λειτουργία του 8→3 κωδικοποιητή και να δοθούν οι λογικές παραστάσεις των εξόδων του.

Ο πίνακας που περιγράφει τη λειτουργία του 8→3 κωδικοποιητή δίδεται στο σχήμα 5.16. Από τον πίνακα αυτό προκύπτουν οι λογικές παραστάσεις των εξόδων του κωδικοποιητή 8→3 που δίδονται στη συνέχεια.

$$y_0 = x_1 + x_3 + x_5 + x_7$$

$$y_1 = x_2 + x_3 + x_6 + x_7$$

$$y_2 = x_4 + x_5 + x_6 + x_7$$

$$A = x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

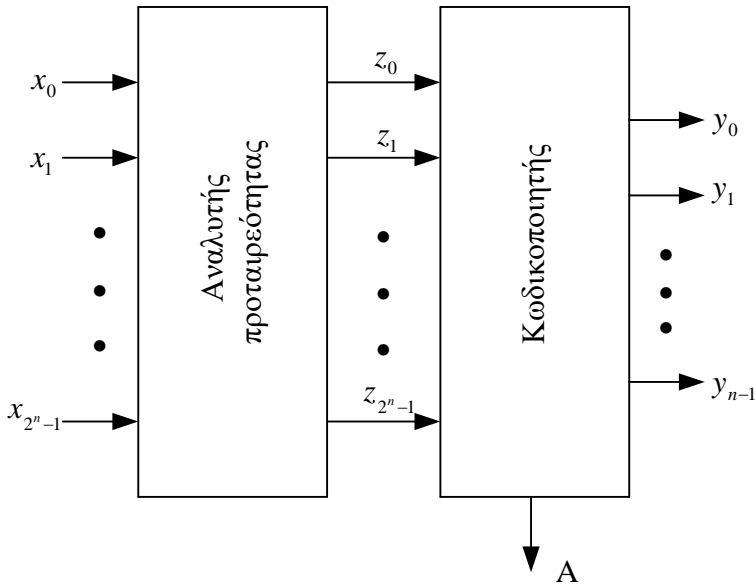
$x_7$	$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	$y_2$	$y_1$	$y_0$	A
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0	1
0	0	0	0	1	0	0	0	0	1	1	1
0	0	0	1	0	0	0	0	1	0	0	1
0	0	1	0	0	0	0	0	1	0	1	1
0	1	0	0	0	0	0	0	1	1	0	1
1	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0

Σχήμα 5.16. Πίνακας λειτουργίας του  $8 \rightarrow 3$  κωδικοποιητή

### 5.6 Κωδικοποιητές προτεραιότητας

Οι κωδικοποιητές που εξετάστηκαν προηγουμένως έχουν τον περιορισμό ότι σε μία δεδομένη χρονική στιγμή μία μόνο είσοδος μπορεί να έχει τη λογική τιμή 1. Οι κωδικοποιητές προτεραιότητας (*priority encoders*) μπορούν να έχουν περισσότερες από μία εισόδους με την τιμή 1. Οι έξοδοι του κωδικοποιητή προτεραιότητας δίδουν στον δυαδικό κώδικα την τιμή του δείκτη της εισόδου με την μεγαλύτερη προτεραιότητα που έχει την τιμή 1. Για έναν κωδικοποιητή προτεραιότητας  $2^n \rightarrow n$  συνήθως η είσοδος  $x_{2^n-1}$  έχει την μεγαλύτερη προτεραιότητα και η είσοδος  $x_0$  τη μικρότερη.

Ο κωδικοποιητής προτεραιότητας μπορεί να υλοποιηθεί με δύο υποκυκλώματα, όπως δείχνεται στο σχήμα 5.17. Το πρώτο είναι ένας αναλυτής προτεραιότητας (*priority analyzer*) που αλλάζει σε 0 όλες τις εισόδους που έχουν τιμή 1 εκτός από αυτή με την μεγαλύτερη προτεραιότητα. Το δεύτερο κύκλωμα είναι ένας κωδικοποιητής όπως αυτός που εξετάσαμε προηγουμένως.



Σχήμα 5.17. Δομή κωδικοποιητή προτεραιότητας

**Παράδειγμα 5.11.** Να σχεδιασθεί ένας  $4 \rightarrow 2$  κωδικοποιητής προτεραιότητας.

Το κύκλωμα θα αποτελείται από έναν αναλυτή προτεραιότητας των 4 bit και έναν κωδικοποιητή. Στο σχήμα 5.18 δίδεται ο πίνακας αληθείας του αναλυτή προτεραιότητας των 4 bit.

$x_3$	$x_2$	$x_1$	$x_0$	$z_3$	$z_2$	$z_1$	$z_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	X	0	0	1	0
0	1	X	X	0	1	0	0
1	X	X	X	1	0	0	0

Σχήμα 5.18. Πίνακας αληθείας του αναλυτή προτεραιότητας των 4 bit.

Από τον πίνακα αληθείας του σχήματος 5.18 προκύπτουν οι λογικές παραστάσεις για τις εξόδους του αναλυτή προτεραιότητας που δίδονται στη συνέχεια.

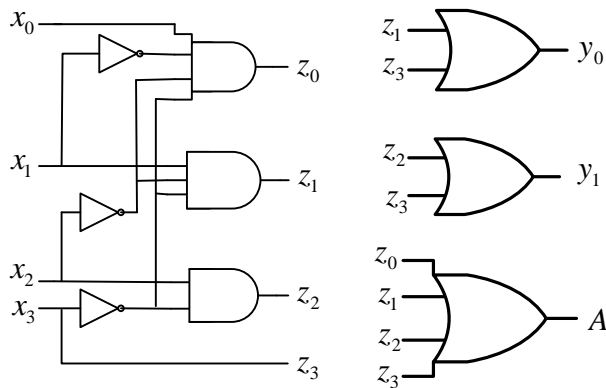
$$z_0 = \bar{x}_3 \cdot \bar{x}_2 \cdot \bar{x}_1 \cdot x_0$$

$$z_1 = \bar{x}_3 \cdot \bar{x}_2 \cdot x_1$$

$$z_2 = \bar{x}_3 \cdot x_2$$

$$z_3 = x_3$$

Το κύκλωμα του  $4 \rightarrow 2$  κωδικοποιητή είναι το ίδιο με αυτό του παραδείγματος 5.9 και δίδεται στη συνέχεια. Η σχεδίαση με λογικές πύλες του κωδικοποιητή προτεραιότητας των 4 bit δίδεται σχήμα 5.19.



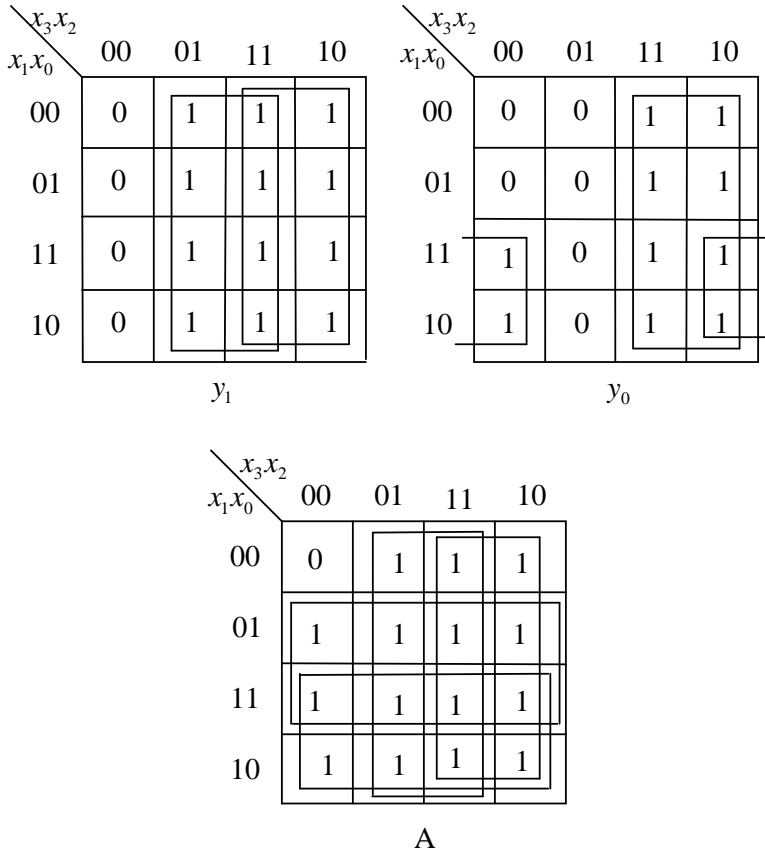
Σχήμα 5.19. Κωδικοποιητής προτεραιότητας

**Παρατήρηση.** Λόγω του μικρού αριθμού των εισόδων ο κωδικοποιητής προτεραιότητας  $4 \rightarrow 2$  μπορεί να σχεδιασθεί και χωρίς το σχεδιασμό αναλυτή προτεραιότητας.

Ο πίνακας αληθείας του  $4 \rightarrow 2$  κωδικοποιητή προτεραιότητας είναι αυτός που δίδεται στη συνέχεια.

$x_3$	$x_2$	$x_1$	$x_0$	$y_1$	$y_0$	A
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Απεικονίζουμε τις τιμές των  $y_1$ ,  $y_0$ ,  $A$  σε χάρτες Karnaugh και απλοποιούμε ό-  
πως στο σχήμα 5.20.



**Σχήμα 5.20.** Εξαγωγή απλοποιημένων παραστάσεων για τα  $y_1$ ,  $y_0$ ,  $A$

Από τους χάρτες Karnaugh του σχήματος 5.20 προκύπτουν οι επόμενες απλοποι-  
ημένες παραστάσεις για τις συναρτήσεις  $y_0$ ,  $y_1$ ,  $A$ .

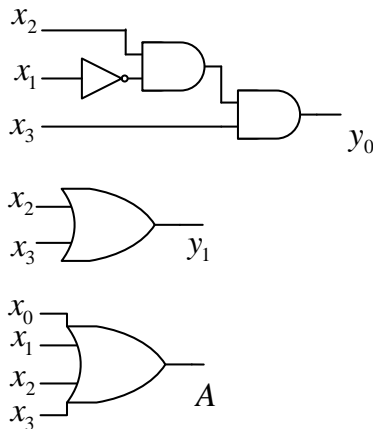
$$y_1 = x_2 + x_3$$

$$y_0 = x_3 + \bar{x}_2 \cdot x_1$$

$$A = x_0 + x_1 + x_2 + x_3$$

Από τις πιο πάνω παραστάσεις προκύπτει η σχεδίαση με λογικές πύλες του απο-  
κωδικοποιητή προτεραιότητας τεσσάρων εισόδων που δίδεται στο σχήμα 5.21.

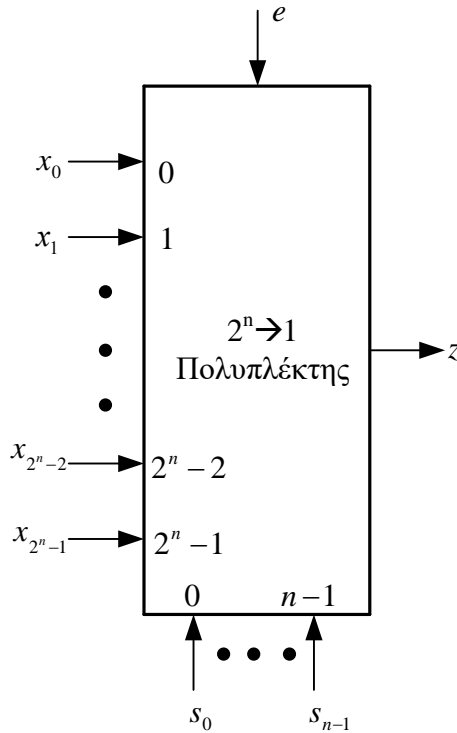




Σχήμα 5.21. Σχεδίαση του  $4 \rightarrow 2$  κωδικοποιητή προτεραιότητας

### 5.7 Πολυπλέκτες

Ο  $2^n \rightarrow 1$  πολυπλέκτης (*multiplexer* ή *selector*) έχει  $2^n$  εισόδους δεδομένων  $x_{2^n-1}, \dots, x_1, x_0$ ,  $n$  εισόδους ελέγχου (ή επιλογής)  $s_{n-1}, \dots, s_1, s_0$  και μία έξοδο  $z$ . Συνήθως υπάρχει και μία είσοδος ενεργοποίησης ή επίτρεψης  $e$ . Στο σχήμα 5.22 δίδεται το λογικό σύμβολο του  $2^n \rightarrow 1$  πολυπλέκτη. Η έξοδος του πολυπλέκτη είναι ίση με την είσοδο που καθορίζεται από τις εισόδους ελέγχου. Έστω  $s = s_{n-1}2^{n-1} + \dots + s_12 + s_0$ , τότε η έξοδος  $z = x_s$ . Για πολυπλέκτες με είσοδο ενεργοποίησης ισχύει  $z = x_s \cdot e$ .

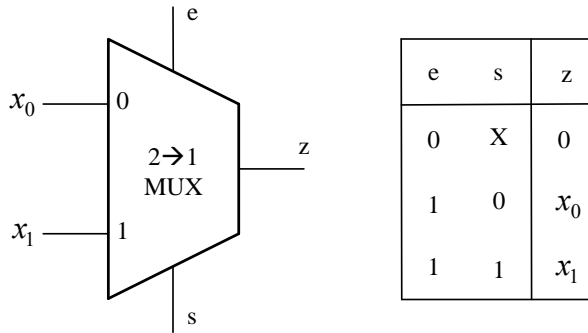


Σχήμα 5.22. Λογικό σύμβολο  $2^n \rightarrow 1$  πολυπλέκτη

Η σχεδίαση του  $2 \rightarrow 1$  πολυπλέκτη εξετάστηκε σε προηγούμενο κεφάλαιο. Στη συνέχεια αναλύεται η σχεδίαση του  $2 \rightarrow 1$  πολυπλέκτη με είσοδο ενεργοποίησης.

**Παράδειγμα 5.12.** Να δοθεί το λογικό σύμβολο, ο πίνακας λειτουργίας και ο συνοπτικός πίνακας αληθείας ενός  $2 \rightarrow 1$  πολυπλέκτη με είσοδο ενεργοποίησης. Στη συνέχεια να σχεδιασθεί με λογικές πύλες.

Το λογικό σύμβολο και ο πίνακας λειτουργία του  $2 \rightarrow 1$  πολυπλέκτη δίδονται στο σχήμα 5.23. Ο πίνακας αληθείας ο συνοπτικός πίνακας αληθείας και η σχεδίασή του με λογικές πύλες δίδεται στο σχήμα 5.24.



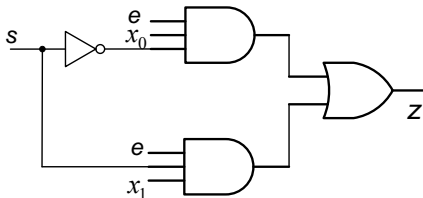
Σχήμα 5.23 Λογικό σύμβολο και πίνακας λειτουργίας του 2→1 πολυπλέκτη με είσοδο επίτρεψης

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

e	s	$x_0$	$x_1$	z
0	x	x	x	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

ΣΥΝΟΠΤΙΚΟΣ ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

e	s	$x_0$	$x_1$	z
0	x	x	x	0
1	0	0	x	0
1	0	1	x	1
1	1	x	0	0
1	1	x	1	1



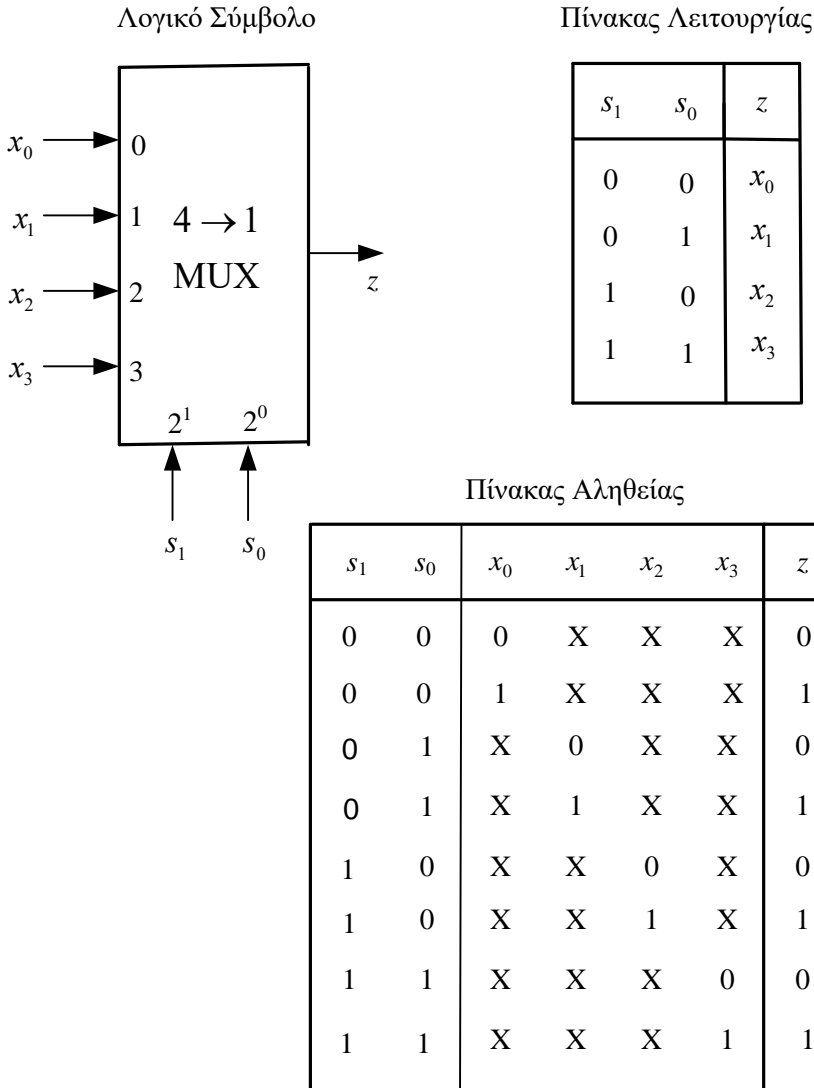
$x_0x_1$ \ es	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	1
10	0	0	0	1

$$z = \overline{e}sx_0 + esx_1$$

Σχήμα 5.24. Πίνακας αληθείας, συνοπτικός πίνακας αληθείας και λογική σχεδίαση του 2→1 πολυπλέκτη με είσοδο επίτρεψης

**Παράδειγμα 5.13.** Να δοθεί το λογικό σύμβολο, ο πίνακας λειτουργίας και ο συνοπτικός πίνακας αληθείας ενός  $4 \rightarrow 1$  πολυπλέκτη χωρίς είσοδο επίτρεψης. Στη συνέχεια να σχεδιασθεί με λογικές πύλες ο  $4 \rightarrow 1$  πολυπλέκτης.

Ο  $4 \rightarrow 1$  πολυπλέκτης έχει 4 γραμμές εισόδου δεδομένων και 2 γραμμές ελέγχου (επιλογής). Το λογικό του σύμβολο, ο πίνακας λειτουργίας και ο συνοπτικός πίνακας αληθείας δίδονται στο σχήμα 5.25.

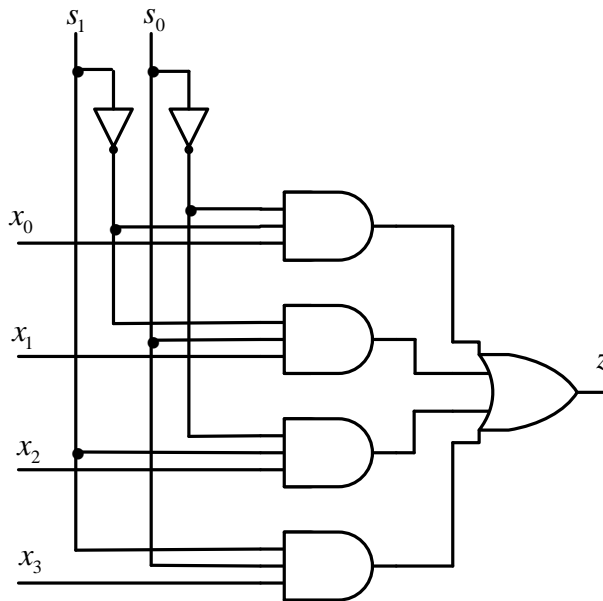


**Σχήμα 5.25** Λογικό σύμβολο, πίνακας λειτουργίας και πίνακας αληθείας του  $4 \rightarrow 1$  πολυπλέκτη

Η λογική εξίσωση για την έξοδο του  $4 \rightarrow 1$  πολυπλέκτη που εξάγεται από τον συνοπτικό πίνακα αληθείας είναι αυτή που δίδεται στη συνέχεια.

$$z = \bar{s}_1 \bar{s}_0 x_0 + \bar{s}_1 s_0 x_1 + s_1 \bar{s}_0 x_2 + s_1 s_0 x_3$$

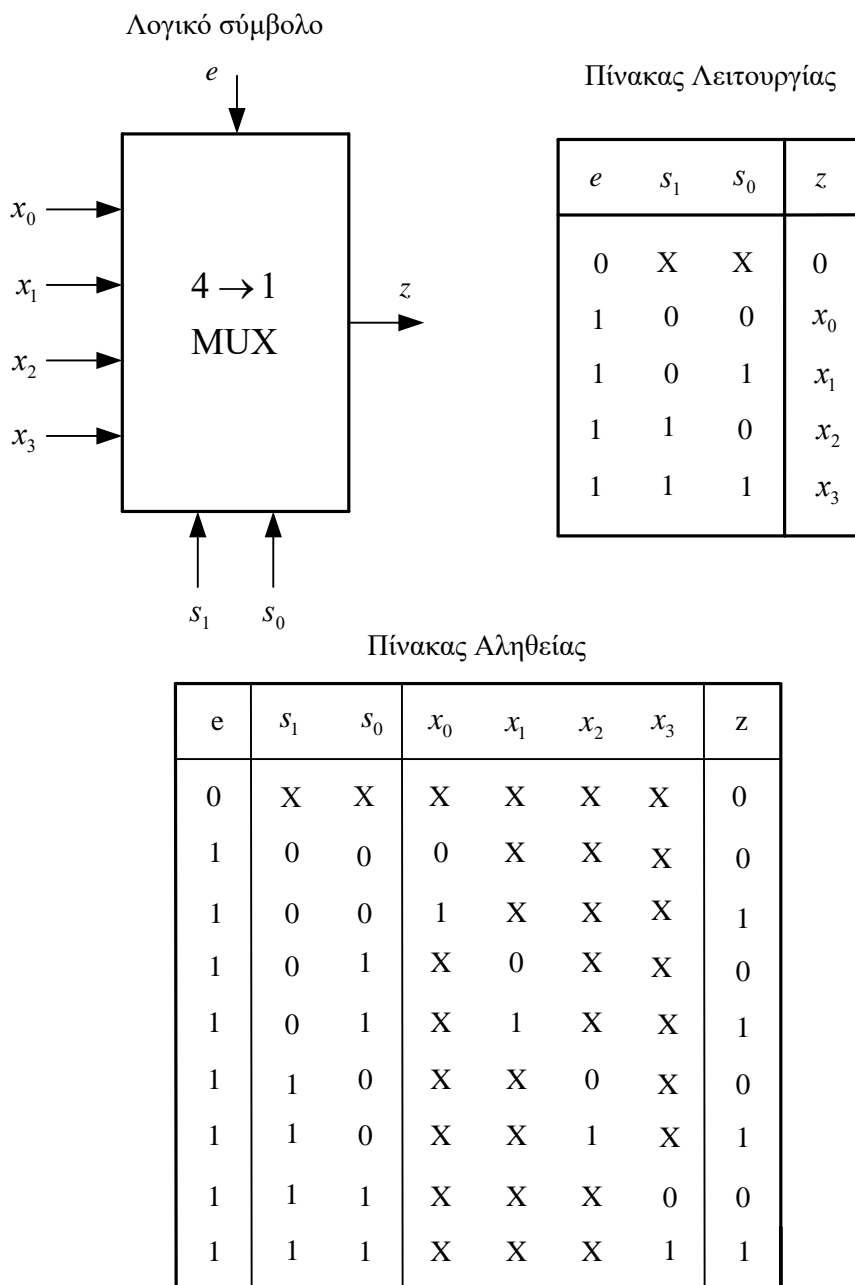
Από την πιο πάνω εξίσωση προκύπτει η υλοποίηση με λογικές πύλες του  $4 \rightarrow 1$  πολυπλέκτη που δίδεται στο σχήμα 5.26.



Σχήμα 5.26. Σχεδίαση του  $4 \rightarrow 1$  πολυπλέκτη

**Παράδειγμα 5.14.** Να δοθεί το λογικό σύμβολο, ο πίνακας λειτουργίας και ο συνοπτικός πίνακας αληθείας ενός  $4 \rightarrow 1$  πολυπλέκτη με είσοδο επίτρεψης. Στη συνέχεια να σχεδιασθεί με λογικές πύλες.

Στο σχήμα 5.27 δίδεται το λογικό σύμβολο, ο πίνακας λειτουργίας και ο πίνακας αληθείας ενός  $4 \rightarrow 1$  πολυπλέκτη με είσοδο επίτρεψης.

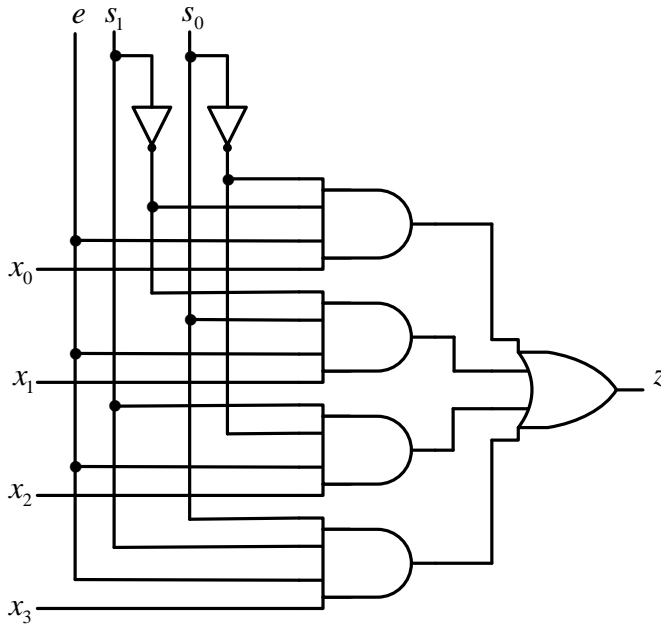


**Σχήμα 5.27.** Λογικό σύμβολο, πίνακας λειτουργίας και πίνακας αληθείας του  $4 \rightarrow 1$  πολυπλέκτη με είσοδο επίτρεψης.

Η λογική παράσταση για την έξοδο του  $4 \rightarrow 1$  πολυπλέκτη που εξάγεται από τον συνοπτικό πίνακα αληθείας είναι η εξής

$$z = e\bar{s}_1\bar{s}_0x_0 + e\bar{s}_1s_0x_1 + es_1\bar{s}_0x_2 + es_1s_0x_3$$

Από την πιο πάνω λογική παράσταση προκύπτει η υλοποίηση του  $4 \rightarrow 1$  πολυπλέκτη που δίδεται στο σχήμα 5.28.



Σχήμα 5.28. Υλοποίηση του  $4 \rightarrow 1$  πολυπλέκτη με είσοδο επίτρεψης

### 5.7.1 Δίκτυα πολυπλεκτών

Ένας πολυπλέκτης με πολλές εισόδους μπορεί να υλοποιηθεί σαν δικτύωμα πολλών επιπέδων από μικρότερους πολυπλέκτες. Η υλοποίηση πολυπλεκτών με αυτό τον τρόπο εξηγείται με το επόμενο παράδειγμα.

**Παράδειγμα 5.15.** Να σχεδιασθεί ένας  $4 \rightarrow 1$  πολυπλέκτης χρησιμοποιώντας σαν στοιχείο σχεδίασης τον  $2 \rightarrow 1$  πολυπλέκτη.

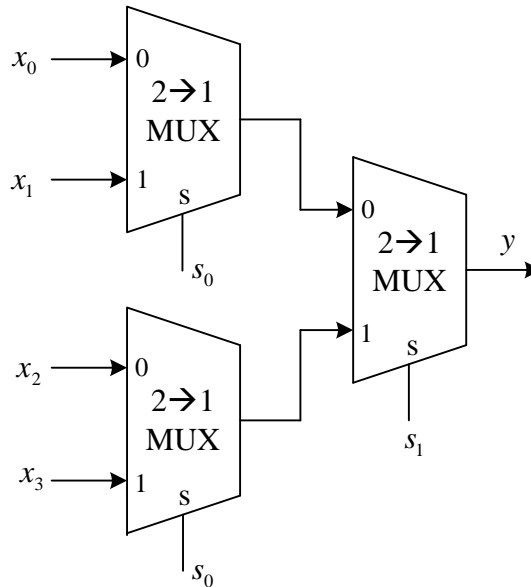
Η λογική συνάρτηση που υλοποιεί ο  $4 \rightarrow 1$  πολυπλέκτης είναι όπως προαναφέραμε η εξής

$$y = \bar{s}_1\bar{s}_0x_0 + \bar{s}_1s_0x_1 + s_1\bar{s}_0x_2 + s_1s_0x_3$$

Αυτή η παράσταση μπορεί να γραφεί όπως στη συνέχεια:

$$y = \bar{s}_1(\bar{s}_0x_0 + s_0x_1) + s_1(\bar{s}_1x_2 + s_0x_3)$$

Με βάση την πιο πάνω σχέση και έχοντας υπ' όψιν ότι ο 2→1 πολυπλέκτης με εισόδους δεδομένων  $x$ ,  $y$  και είσοδο ελέγχου  $s$ , υλοποιεί την λογική παράσταση  $\bar{s} \cdot x + s \cdot y$  προκύπτει η σχεδίαση του 4→1 πολυπλέκτη με χρήση 2→1 πολυπλεκτών που δίδεται στο σχήμα 5.29.



**Σχήμα 5.29.** Σχεδίαση 4→1 πολυπλέκτη με χρήση 2→1 πολυπλεκτών

### 5.7.2 Υλοποίηση λογικών συναρτήσεων με πολυπλέκτες

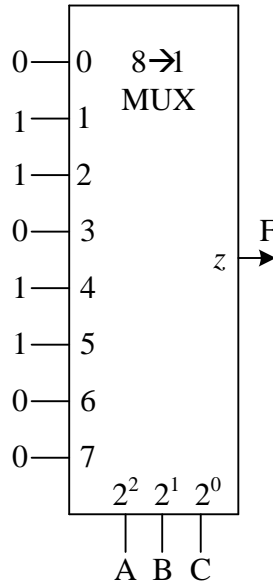
Ένας  $2^n \rightarrow 1$  πολυπλέκτης όπως αυτός του σχήματος 5.22 μπορεί να χρησιμοποιηθεί για την υλοποίηση λογικών συναρτήσεων  $n$  μεταβλητών. Οι λογικές μεταβλητές της συνάρτησης συνδέονται στις εισόδους ελέγχου του πολυπλέκτη. Οι εισόδοι δεδομένων του πολυπλέκτη τίθενται στο 0 ή στο 1 ανάλογα με την τιμή της συνάρτησης είναι 0 ή 1 για τον αντίστοιχο συνδυασμό τιμών των μεταβλητών ελέγχου.

**Παράδειγμα 5.16.** Να υλοποιηθεί με πολυπλέκτη η λογική συνάρτηση της οποίας η κανονική λογική παράσταση δίδεται στην συνέχεια

$$F = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C = \sum(1,2,4,5)$$



Εφόσον η λογική συνάρτηση έχει 3 μεταβλητές θα χρησιμοποιηθεί ένας  $2^3 \rightarrow 1$  πολυπλέκτης. Οι μεταβλητές εισόδου συνδέονται στις εισόδους ελέγχου του πολυπλέκτη. Συνδέονται στο 1 οι εισόδοι 1, 2, 4, 5. Οι άλλες εισόδοι συνδέονται στο 0. Η υλοποίηση με τον  $8 \rightarrow 1$  πολυπλέκτη της δοσμένης λογικής συνάρτησης δίδεται στη συνέχεια:



Το μέγεθος του πολυπλέκτη που χρησιμοποιείται για την υλοποίηση μιας λογικής συνάρτησης μπορεί να μειωθεί με τη χρήση μιας μεταβλητής της συνάρτησης σαν είσοδο δεδομένων. Σε αυτή την περίπτωση η υλοποίηση λογικής συνάρτησης με  $n$  μεταβλητές είναι δυνατή με τη χρήση  $2^{n-1} \rightarrow 1$  πολυπλέκτη. Η μέθοδος αυτή εξηγείται με το επόμενο παράδειγμα.

**Παράδειγμα 5.17.** Να υλοποιηθεί με πολυπλέκτη η λογική συνάρτηση της οποίας η κανονική λογική παράσταση δίδεται στην συνέχεια.

$$F = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C$$

Εφ' όσον η λογική συνάρτηση έχει 3 μεταβλητές η υλοποίησή της θα βασισθεί σε πολυπλέκτη με  $3-1=2$  γραμμές ελέγχου, δηλαδή σε έναν  $2^2 \rightarrow 1$  πολυπλέκτη. Η λογική συνάρτηση που υλοποιεί ο  $4 \rightarrow 1$  πολυπλέκτης είναι αυτή που δίδεται στη συνέχεια

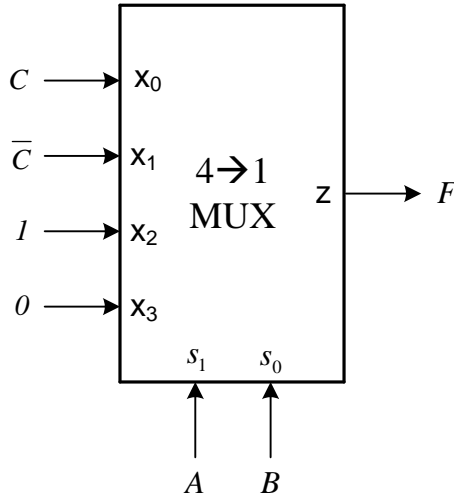
$$z = \bar{s}_1 \bar{s}_0 x_0 + \bar{s}_1 s_0 x_1 + s_1 \bar{s}_0 x_2 + s_1 s_0 x_3$$

Η λογική συνάρτηση  $F$  μπορεί να ξαναγραφεί σαν

$$F = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot (\bar{C} + C)$$

$$\text{ή } F = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot 1 + A \cdot B \cdot 0$$

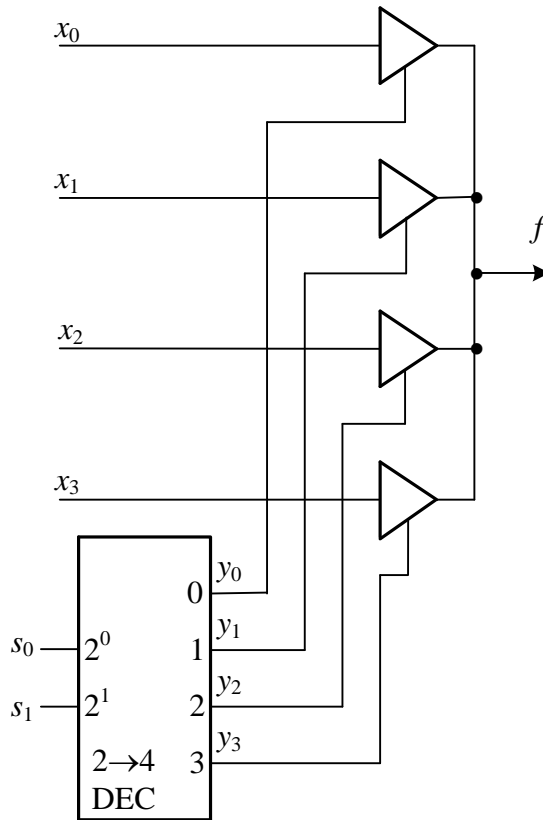
Συγκρίνοντας τις λογικές παραστάσεις των  $z$ ,  $F$  παρατηρούμε ότι η λογική συνάρτηση  $F$  μπορεί να υλοποιηθεί από τον  $2^2 \rightarrow 1$  πολυπλέκτη θέτοντας  $s_1=A$ ,  $s_0=B$ ,  $x_0=C$ ,  $x_1=\bar{C}$ ,  $x_2=1$ ,  $x_3=0$ . Η υλοποίηση αυτή δίδεται στη συνέχεια.



### 5.7.3 Υλοποίηση πολυπλεκτών με απομονωτές τριών καταστάσεων

Οι πολυπλέκτες μπορεί να σχεδιασθούν με απομονωτές τριών καταστάσεων. Η σχεδίαση του  $2 \rightarrow 1$  πολυπλέκτη με απόμονωτές τριών καταστάσεων περιγράφηκε σε προηγούμενο κεφάλαιο.

Στο σχήμα 5.30 δίδεται η σχεδίαση ενός  $4 \rightarrow 1$  πολυπλέκτη με βάση κυκλώματα απομονωτή τριών καταστάσεων. Οι έξοδοι των 4 απομονωτών τριών καταστάσεων συνδέονται ώστε να υπάρχει μία κοινή έξοδος. Για να είναι μόνο μία μόνο είσοδος συνδεδεμένη στην έξοδο, χρησιμοποιείται ένας  $2 \rightarrow 4$  αποκωδικοποιητής οι έξοδοι του οποίου συνδέονται στις εισόδους ελέγχου των απομονωτών. Με τις εισόδους  $s_1$ ,  $s_0$  καθορίζουμε ποιες από τις εισόδους  $x_i$  θα συνδέεται στην έξοδο  $y$ . Για παράδειγμα με  $(s_1, s_0) = (0, 1)$  ενεργοποιείται μόνο η έξοδος  $y_1$  ( $y_1=1$ ). Η έξοδος αυτή ενεργοποιεί τον απομονωτή στον οποίο είναι συνδεδεμένη και η έξοδος  $f$  γίνεται ίση με την είσοδο  $x_1$ .

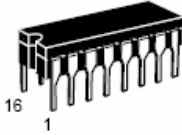


Σχήμα 5.30. Σχεδίαση  $4 \rightarrow 1$  πολυπλέκτη με απομονωτές τριών καταστάσεων

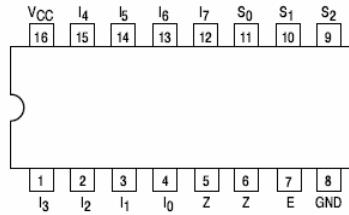
#### 5.7.4 Ολοκληρωμένα κυκλώματα με πολυπλέκτες

Στο σχήμα 5.31 περιγράφεται το ολοκληρωμένο κύκλωμα 74LS151 που περιέχει έναν  $8 \rightarrow 1$  πολυπλέκτη με συμπληρωματική (ενεργοποιούμενη με 0) είσοδο επί-τρησης.

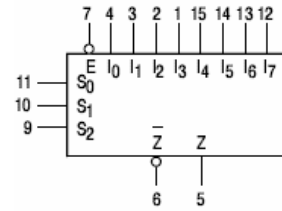
ΟΛΟΚΛΗΡΩΜΕΝΟ  
ΚΥΚΛΩΜΑ



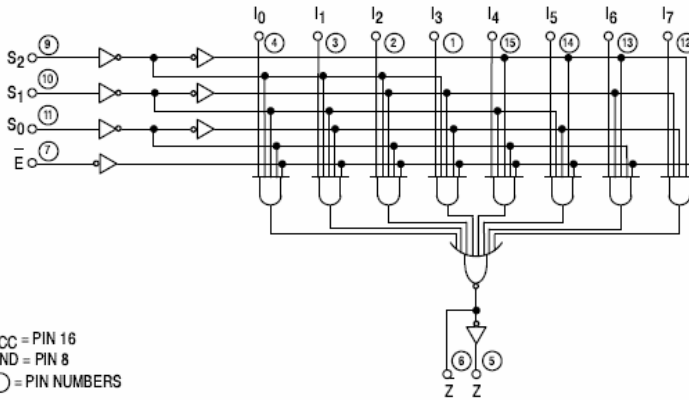
ΔΙΑΓΡΑΜΜΑ ΑΚΙΔΩΝ



ΛΟΓΙΚΟ ΣΥΜΒΟΛΟ



ΛΟΓΙΚΟ ΚΥΚΛΩΜΑ



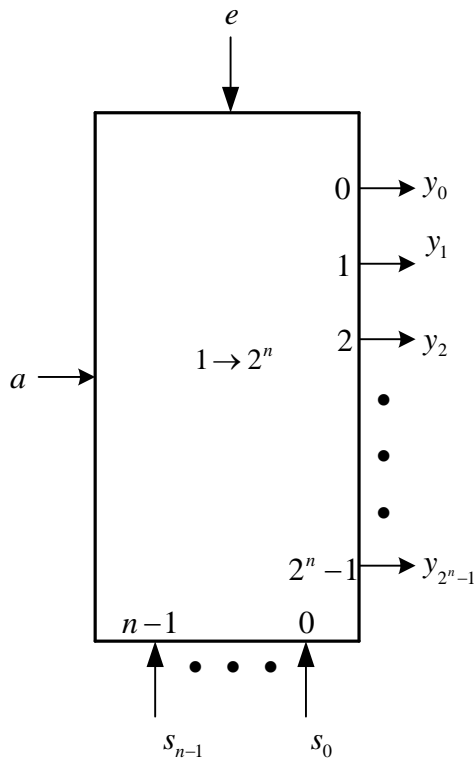
ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

E	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	I <sub>7</sub>	Z	Z
H	X	X	X	X	X	X	X	X	X	X	X	H	L
L	L	L	L	L	X	X	X	X	X	X	X	H	L
L	L	L	L	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	H	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	H	L
L	L	H	L	X	X	H	X	X	X	X	X	L	H
L	L	H	H	X	X	X	L	X	X	X	X	H	L
L	L	H	H	X	X	X	H	X	X	X	X	L	H
L	H	L	L	X	X	X	X	L	X	X	X	H	L
L	H	L	L	X	X	X	X	H	X	X	X	L	H
L	H	L	H	X	X	X	X	X	L	X	X	H	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	H	L
L	H	H	L	X	X	X	X	X	X	H	X	L	H
L	H	H	H	X	X	X	X	X	X	X	H	L	H

Σχήμα 5.31. Ολοκληρωμένο κύκλωμα 8 → Πολυπλέκτη 74LS151

### 5.8 Αποπολυπλέκτες

Ένας αποπολυπλέκτης (*demultiplexer*)  $2^n$  εξόδων, (ή  $1 \rightarrow 2^n$  αποπολυπλέκτης) είναι ένα κύκλωμα με  $n$  γραμμές ελέγχου ή επιλογής  $s_{n-1}, \dots, s_1, s_0$ , μία γραμμή εισόδου δεδομένων  $a$  και  $2^n$  γραμμές εξόδου δεδομένων  $y_{2^n-1}, \dots, y_1, y_0$ . Οι αποπολυπλέκτες επιτελούν την αντίστροφη εργασία από τους πολυπλέκτες. Δηλαδή με  $n$  εισόδους επιλογής καθορίζεται σε ποια από τις  $2^n$  εξόδους θα οδηγηθεί η είσοδος. Συνήθως στους αποπολυπλέκτες υπάρχει και μία είσοδος ενεργοποίησης  $e$  για τη δημιουργία δικτύων αποπολυπλεκτών. Στο σχήμα 5.32 δίδεται το λογικό σύμβολο του  $1 \rightarrow 2^n$  αποπολυπλέκτη.



Σχήμα 5.32. Λογικό σύμβολο  $1 \rightarrow 2^n$  αποπολυπλέκτη

Στο σχήμα 5.33 δίδεται ο πίνακας λειτουργίας και ο πίνακας αληθείας του  $1 \rightarrow 4$  αποπολυπλέκτη.

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ							
$s_1$	$s_0$	$a$	$y_0$	$y_1$	$y_2$	$y_3$	
0	0	0	0	0	0	0	
0	0	1	1	0	0	0	
0	1	0	0	0	0	0	
0	1	1	0	1	0	0	
1	0	0	0	0	0	0	
1	0	0	0	0	1	0	
1	1	0	0	0	0	0	
1	1	1	0	0	0	1	

ΠΙΝΑΚΑΣ ΛΕΙΤΟΥΡΓΙΑΣ							
$s_1$	$s_0$	$y_0$	$y_1$	$y_2$	$y_3$		
0	0	$a$	0	0	0		
0	1	0	$a$	0	0		
1	0	0	0	$a$	0		
1	1	0	0	0	$a$		

*Σχήμα 5.33. Πίνακας λειτουργίας και πίνακας αληθείας του  $1 \rightarrow 4$  αποπολυπλέκτη*

Από τον πίνακα αληθείας προκύπτουν οι επόμενες λογικές συναρτήσεις.

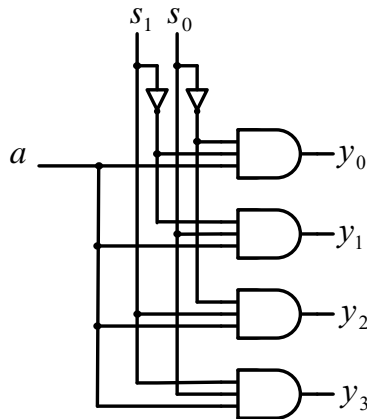
$$y_0 = \bar{s}_1 \cdot \bar{s}_0 \cdot a$$

$$y_1 = \bar{s}_1 \cdot s_0 \cdot a$$

$$y_2 = s_1 \cdot \bar{s}_0 \cdot a$$

$$y_3 = s_1 \cdot s_0 \cdot a$$

Η υλοποίηση του  $1 \rightarrow 4$  αποπολυπλέκτη δίδεται στο σχήμα 5.34. Παρατηρούμε ότι η υλοποίηση του  $4 \rightarrow 1$  πολυπλέκτη είναι η ίδια με αυτή του  $2 \rightarrow 4$  αποκωδικοποιητή με είσοδο επίτρεψης.



Σχήμα 5.34. Υλοποίηση του  $1 \rightarrow 4$  αποπολυπλέκτη

### 5.9 Κυκλώματα μετατροπής από και προς τον κώδικα Gray

#### Κυκλώματα μετατροπής από τον δυαδικό κώδικα στον κώδικα Gray

Στο παράγραφο 2.8 έγινε μία σύντομη εισαγωγή στους κώδικες Gray. Ένας άλλος τρόπος παραγωγής των κωδικών λέξεων του κώδικα Gray από τις αντίστοιχες λέξεις του δυαδικού κώδικα περιγράφεται στη συνέχεια.

1. Τα ψηφία  $g_i$  λέξεων του κώδικα Gray αριθμούνται από δεξιά προς τα αριστερά από 0 έως  $n-1$ .
2. Το ψηφίο  $g_{n-1}$  ισούται με το ψηφίο  $b_{n-1}$  της αντίστοιχης δυαδικής λέξης.
3. Το ψηφίο  $g_i$ , όπου  $n-2 \leq i \leq 0$ , μιας λέξης του κώδικα Gray είναι 0, εάν τα ψηφία  $b_i, b_{i+1}$  της αντίστοιχης δυαδικής λέξης είναι ίδια, διαφορετικά είναι 1.

Έστω  $b_{n-1}b_{n-2} \dots b_1b_0$  η παράσταση ενός αριθμού στο δυαδικό σύστημα. Η μετατροπή του αριθμού αυτού στον κώδικα Gray γίνεται, σύμφωνα με τα πιο πάνω, με τις επόμενες σχέσεις.

$$\begin{aligned}
 g_{n-1} &= b_{n-1} \\
 g_{n-2} &= b_{n-2} \oplus b_{n-1} \\
 &\dots \dots \\
 g_i &= b_i \oplus b_{i+1} \\
 &\dots \dots \\
 g_0 &= b_0 \oplus b_1
 \end{aligned}$$

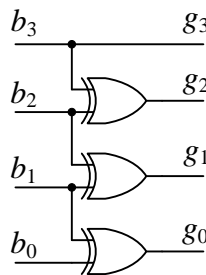
όπου  $g_{n-1}g_{n-2} \dots g_1g_0$  είναι η παράσταση του αριθμού στον κώδικα Gray.

**Παράδειγμα 5.18.** Να σχεδιασθεί κύκλωμα μετατροπής από τον δυαδικό κώδικα των 4 bit στον αντίστοιχο κώδικα Gray.

Έστω  $b_3b_2b_1b_0$  η αναπαράσταση ενός αριθμού στον δυαδικό κώδικα και  $g_3g_2g_1g_0$  στον κώδικα Gray. Σύμφωνα με τις προηγούμενες σχέσεις

$$\begin{aligned} g_3 &= b_3 \\ g_2 &= b_2 \oplus b_3 \\ g_1 &= b_1 \oplus b_2 \\ g_0 &= b_0 \oplus b_1 \end{aligned}$$

Η σχεδίαση του κυκλώματος μετατροπής δίδεται στη συνέχεια



### Κυκλώματα μετατροπής από τον κώδικα Gray στον δυαδικό κώδικα

Για τον κώδικα Gray ισχύει  $b_{n-1} = g_{n-1}$  και  $g_i = b_i \oplus b_{i+1}$  για  $0 \leq i \leq n-2$ . Ισχύει επίσης η σχέση  $g_i \oplus b_{i+1} = (b_i \oplus b_{i+1}) \oplus b_{i+1} = b_i \oplus (b_{i+1} \oplus b_{i+1}) = b_i \oplus 0$  ή  $b_i = g_i \oplus b_{i+1}$ . Επομένως η μετατροπή μιας λέξης του κώδικα Gray στην αντίστοιχη του δυαδικού κώδικα γίνεται σύμφωνα με τις σχέσεις που δίδονται στη συνέχεια

$$\begin{aligned} b_{n-1} &= g_{n-1} \\ b_{n-2} &= g_{n-2} \oplus b_{n-1} \\ b_{n-3} &= g_{n-3} \oplus b_{n-2} \\ &\dots \dots \\ b_i &= g_i \oplus b_{i+1} \\ &\dots \dots \\ b_0 &= g_0 \oplus b_1 \end{aligned}$$

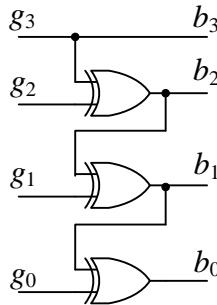
**Παράδειγμα 5.19.** Να σχεδιασθεί κύκλωμα μετατροπής από τον κώδικα Gray των 4 bit στον αντίστοιχο δυαδικό κώδικα.

Έστω  $b_3b_2b_1b_0$  η αναπαράσταση ενός αριθμού στον δυαδικό κώδικα και  $g_3g_2g_1g_0$  στον κώδικα Gray. Σύμφωνα με τις προηγούμενες σχέσεις



$$\begin{aligned}
 b_3 &= g_3 \\
 b_2 &= g_2 \oplus b_3 \\
 b_1 &= g_1 \oplus b_2 \\
 b_0 &= g_0 \oplus b_1
 \end{aligned}$$

Η σχεδίαση του κυκλώματος δίδεται στη συνέχεια



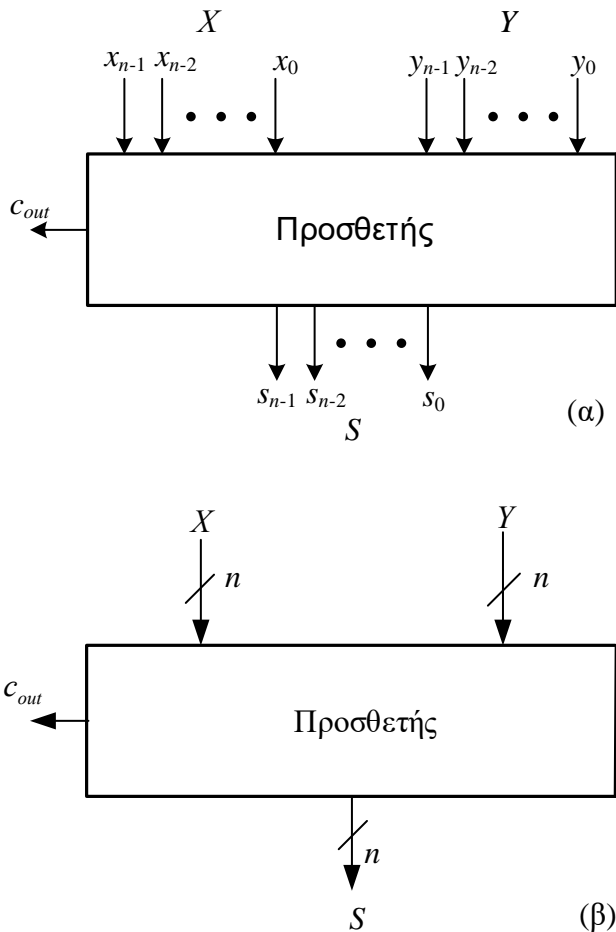
## 5.10 Κυκλώματα δυαδικής πρόσθεσης

### 5.10.1 Προσθετές $n$ -ψήφων μη προσημασμένων δυαδικών αριθμών

Ο παράλληλος δυαδικός προσθετής  $n$ -ψήφων μη προσημασμένων δυαδικών αριθμών είναι συνδυαστικό κύκλωμα που έχει σαν εισόδους δύο  $n$ -ψήφιους αριθμούς  $X = x_{n-1}x_{n-2} \dots x_1x_0$ ,  $Y = y_{n-1}y_{n-2} \dots y_1y_0$  και παράγει σαν έξοδο έναν δυαδικό αριθμό  $S = s_{n-1}s_{n-2} \dots s_1s_0$  που αναπαριστά το αριθμητικό άθροισμα των  $X$  και  $Y$  και ένα κρατούμενο εξόδο  $c_{out}$ . Στην συνέχεια δίδεται η αριθμητική πράξη που υλοποιεί ο προσθετής

$$X + Y = S + 2^n c_{out}$$

Στο σχήμα 5.35 δίδεται το λογικό σύμβολο (α) και το συνοπτικό λογικό σύμβολο (β) του παράλληλου προσθετή  $n$ -ψηφίων αριθμών.



Σχήμα 5.35. Προσθετής των  $n$ -bit

Υπάρχουν πολλές υλοποιήσεις των παράλληλων προσθετών που διαφέρουν στη δομή, στο κυκλωματικό κόστος και στην ταχύτητα λειτουργίας. Η υλοποίηση με τη μέθοδο σχεδίασης η οποία καταλήγει σε δύο επίπεδα πυλών ακόμα και για σχετικά μικρό αριθμό εισόδων απαιτεί μεγάλο αριθμό πυλών με μεγάλο αριθμό εισόδων η κάθε μία. Κατά συνέπεια η διερεύνηση άλλων πιο αποδοτικών υλοποιήσεων είναι απαραίτητη. Στη συνέχεια περιγράφεται μια υλοποίηση που απαιτεί μικρό αριθμό πυλών και είναι γνωστή σαν προσθετής με διάδοση κρατουμένου.

### Προσθετές με διάδοση κρατουμένου

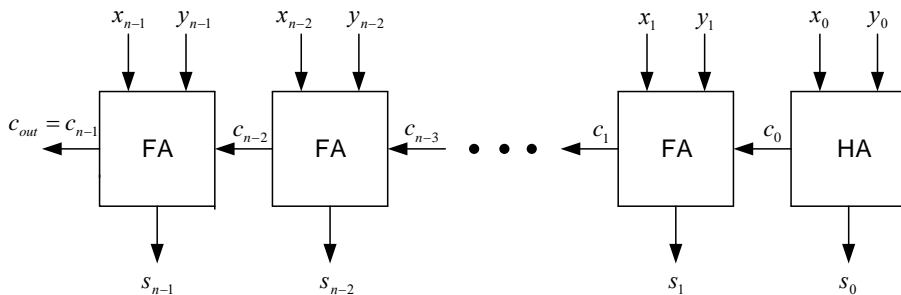
Όπως αναφέραμε σε προηγούμενο κεφάλαιο για να υπολογίσουμε το άθροισμά των  $X, Y$  προσθέτουμε τα λιγότερο σημαντικά ψηφία  $x_0, y_0$  τα οποία παράγουν το

ψηφίο αθροίσματος  $s_0$  και ένα ψηφίο κρατούμενου εξόδου  $c_0$ . Στην συνέχεια προσθέτουμε τα ψηφία  $x_1, y_1$  και το κρατούμενο  $c_0$  από την προηγούμενη πρόσθεση και παράγεται το ψηφίο αθροίσματος  $s_1$  και το κρατούμενο  $c_1$ . Η διαδικασία επαναλαμβάνεται από αριστερά προς τα δεξιά για τα υπόλοιπα ψηφία. Το  $c_{n-1}$  είναι το κρατούμενο εξόδου ( $c_{out}$ ) της πρόσθεσης των  $X$  και  $Y$ . Στη συνέχεια συνοψίζεται η πρόσθεση των αριθμών  $X$  και  $Y$ .

$$\begin{array}{r}
 c_{n-2} c_{n-3} \dots c_1 c_0 \\
 x_{n-1} x_{n-2} \dots x_2 x_1 x_0 \\
 + y_{n-1} y_{n-2} \dots y_2 y_1 y_0 \\
 \hline
 c_{n-1} s_{n-1} s_{n-2} \dots s_2 s_1 s_0
 \end{array}$$

$c_{out}$  ↗

Με βάση αυτή την μεθοδολογία πρόσθεσης οι προσθετές με διάδοση κρατούμενου (*ripple carry adders*) υλοποιούνται με την διασύνδεση προσθετών του ενός bit που ονομάζονται ημιαθροιστές (HA) και πλήρεις αθροιστές (full adders ή FA) όπως στο σχήμα 5.36. Οι πλήρεις προσθετές εκτελούν πρόσθεση αντιστοίχων ζευγών ψηφίων εισόδου και ενός κρατούμενου εισόδου και παράγουν ένα ψηφίο αθροίσματος και ένα ψηφίο κρατούμενου εξόδου.



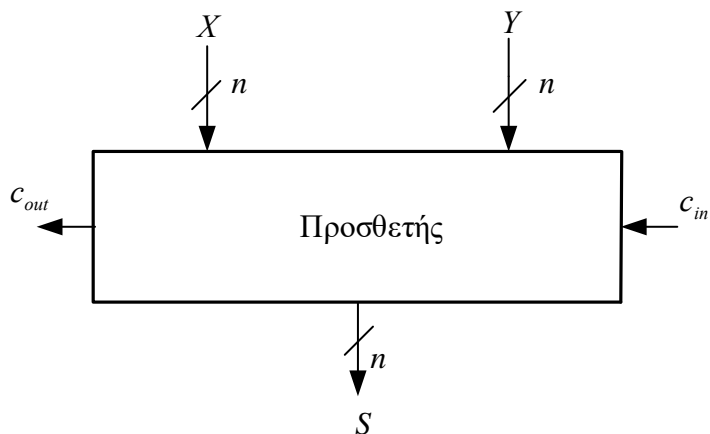
Σχήμα 5.36. Προσθετής των  $n$ -bit με διάδοση κρατούμενου

**Παράλληλοι προσθετές με κρατούμενο εισόδου**

Για λόγους επεκτασιμότητας οι παράλληλοι προσθετές μπορεί να διαθέτουν κρατούμενο εξόδου  $c_{in}$ . Στην συνέχεια δίδεται η αριθμητική πράξη που υλοποιεί ο προσθετής των  $n$ -bit με κρατούμενο εισόδου

$$X + Y + c_{in} = S + 2^n c_{out}$$

Στο σχήμα 5.37 δίδεται το λογικό σύμβολο ενός παράλληλου προσθετή  $n$ -ψηφίων αριθμών με κρατούμενο εισόδου.



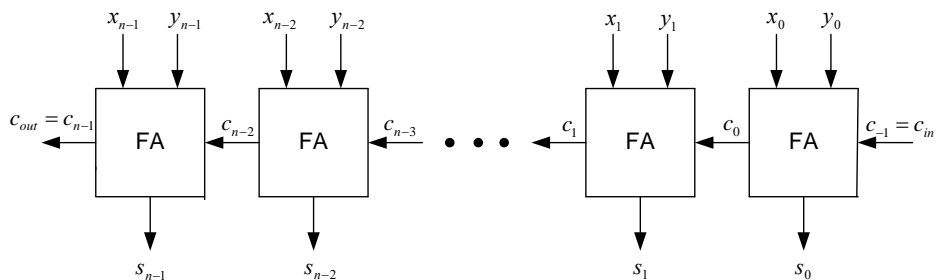
Σχήμα 5.37. Προσθετής των  $n$ -bit με κρατούμενο εισόδου

Στη συνέχεια συνοψίζεται η πρόσθεση των αριθμών  $X$  και  $Y$  με κρατούμενο εισόδου.

$$\begin{array}{r}
 c_{n-2} \ c_{n-3} \ \dots \ c_1 \ c_0 \ c_{in} \\
 x_{n-1} \ x_{n-2} \ \dots \ x_2 \ x_1 \ x_0 \\
 + \ y_{n-1} \ y_{n-2} \ \dots \ y_2 \ y_1 \ y_0 \\
 \hline
 c_{n-1} \ s_{n-1} \ s_{n-2} \ \dots \ s_2 \ s_1 \ s_0
 \end{array}$$

$c_{out}$  ←

Με βάση αυτή την μεθοδολογία πρόσθεσης οι προσθετές με διάδοση κρατουμένου με κρατούμενο εισόδου υλοποιούνται με την διασύνδεση πλήρων αθροιστών όπως στο σχήμα 5.38.



Σχήμα 5.38. Προσθετής των  $n$ -bit με διάδοση κρατουμένου και κρατούμενο εισόδου

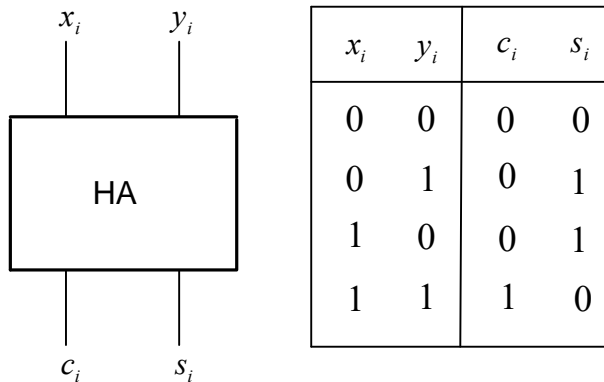
### 5.10.2 Προσθετές του ενός bit

#### Ημιαθροιστής

Για τη δημιουργία προσθετών με διάδοση κρατούμενου, χωρίς κρατούμενο εισόδου χρησιμοποιείται για την πρόσθεση των λιγότερο σημαντικών ψηφίων ένας ημιαθροιστής. Ο ημιαθροιστής (*Half-Adder* ή *HA*) είναι ένα λογικό κύκλωμα που προσθέτει δύο δυαδικά ψηφία  $x_i, y_i$  και δίνει στην έξοδο το άθροισμα  $s_i$  και το κρατούμενο εξόδου  $c_i$  της πρόσθεσης. Ο ημιαθροιστής εκτελεί την πράξη

$$2c_i + s_i = x_i + y_i.$$

Στο σχήμα 5.39 δίδεται το λογικό σύμβολο και ο πίνακας αληθείας του ημιαθροιστή.



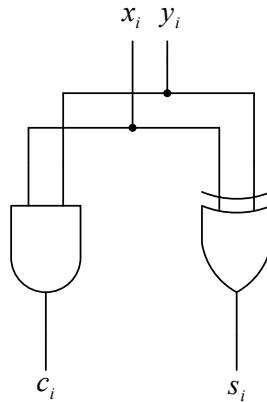
Σχήμα 5. 39. Λογικό σύμβολο και πίνακας αληθείας του ημιαθροιστή

Από τον πίνακα αληθείας προκύπτουν οι επόμενες λογικές παραστάσεις

$$c_i = x_i y_i$$

$$s_i = x_i \bar{y}_i + \bar{x}_i y_i = x_i \oplus y_i$$

Η υλοποίηση με λογικές πύλες του ημιαθροιστή που αντιστοιχεί στις πιο πάνω λογικές παραστάσεις δίδεται στο σχήμα 5.40.



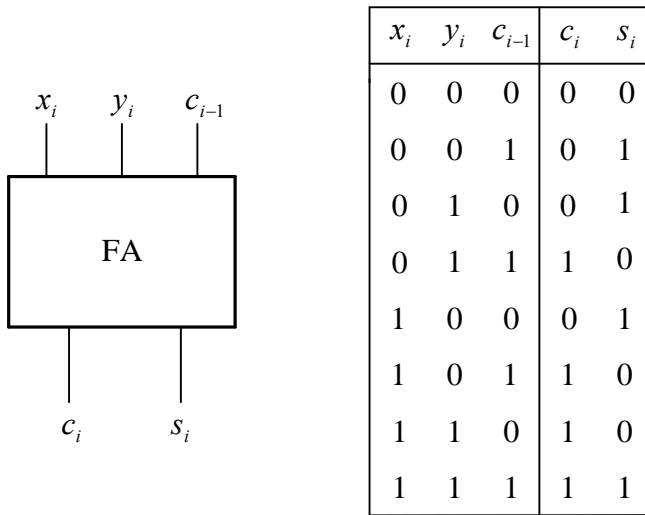
**Σχήμα 5.40.** Σχεδίαση ημιαθροιστή

Ο ημιαθροιστής χρησιμοποιείται όπως και ο πλήρης αθροιστής για τη σχεδίαση εκτός από τους παράλληλους αθροιστές και στην σχεδίαση άλλων αριθμητικών κυκλωμάτων.

### Πλήρης αθροιστής

Ο *πλήρης αθροιστής* (*full adder*, ή FA) είναι κύκλωμα που προσθέτει δύο δυαδικά ψηφία  $x_i$ ,  $y_i$ , το κρατούμενο εισόδου  $c_{i-1}$  και δίνει στην έξοδό του το άθροισμα  $s_i$  και το κρατούμενο εξόδου  $c_i$ . Στο σχήμα 5.41 δίδεται το λογικό σύμβολο και ο πίνακας αληθείας του πλήρη αθροιστή. Η αριθμητική πράξη που εκτελεί ο πλήρης αθροιστής είναι η

$$2c_i + s_i = x_i + y_i + c_{i-1}$$

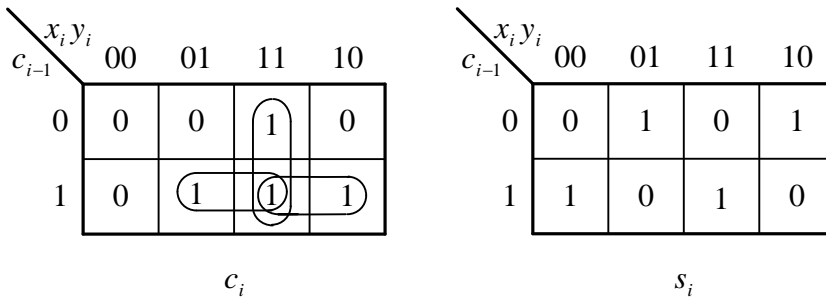


**Σχήμα 5.41.** Λογικό σύμβολο και πίνακας αληθείας του πλήρη αθροιστή

Στη συνέχεια δίδονται τρεις διαφορετικές υλοποιήσεις του πλήρη αθροιστή. Οι δύο πρώτες βασίζονται μόνο σε λογικές πύλες, ενώ η τρίτη βασίζεται σε 2→1 πολυπλέκτη.

**Υλοποίηση Α**

Η υλοποίηση αυτή προκύπτει ως εξής. Απεικονίζονται σε χάρτες Karnaugh τα  $c_i$  και  $s_i$  σύμφωνα με τον πίνακα αληθείας του πλήρη αθροιστή.



Από το χάρτη πρώτο χάρτη Karnaugh προκύπτει η απλοποιημένη έκφραση για το κρατούμενο εξόδου  $c_i$ :

$$c_i = x_i y_i + x_i c_{i-1} + y_i c_{i-1}$$

ή

$$c_i = x_i y_i + (x_i + y_i) c_{i-1}$$

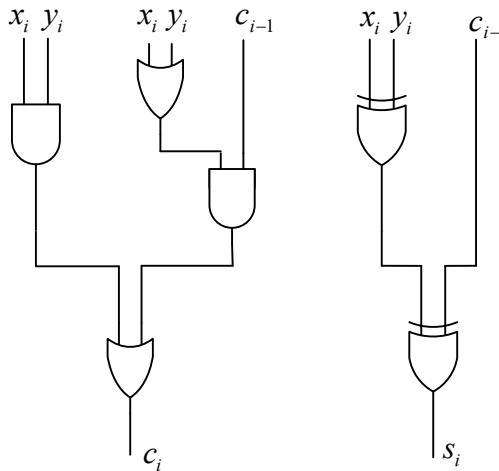
Για την λογική συνάρτηση του ψηφίου αθροίσματος  $s_i$  παρατηρούμε ότι δεν μπορεί να γίνει απλοποίηση με το χάρτη Karnaugh. Μπορεί όμως να υλοποιηθεί με πύλες XOR όπως δείχνεται στη συνέχεια.

$$\begin{aligned} s_i &= \bar{x}_i \bar{y}_i c_{i-1} + \bar{x}_i y_i \bar{c}_{i-1} + x_i \bar{y}_i \bar{c}_{i-1} + x_i y_i c_{i-1} \\ &= (\bar{x}_i \bar{y}_i + x_i y_i) c_{i-1} + (\bar{x}_i y_i + x_i \bar{y}_i) \bar{c}_{i-1} \\ &= (x_i \oplus y_i) c_{i-1} + (x_i \oplus y_i) \bar{c}_{i-1} \end{aligned}$$

Δηλαδή,

$$s_i = (x_i \oplus y_i) \oplus c_{i-1}$$

Από τις πιο πάνω λογικές παραστάσεις των  $c_i, s_i$  προκύπτει η σχεδίαση του πλήρη αθροιστή που δίδεται στο σχήμα 5.42.



Σχήμα 5.42. Σχεδίαση πλήρη αθροιστή

### Υλοποίηση Β

Στη συνέχεια δίδεται μία διαφορετική σχεδίαση του πλήρη αθροιστή. Η λογική παράσταση για το  $s_i$  παραμένει αυτή που προέκυψε για την υλοποίηση Α, δηλαδή

$$s_i = (x_i \oplus y_i) \oplus c_{i-1}$$

Για το  $c_i$  ισχύει



$$c_i = \bar{x}_i y_i c_{i-1} + x_i \bar{y}_i c_{i-1} + x_i y_i \bar{c}_{i-1} + x_i y_i c_{i-1} \dots$$

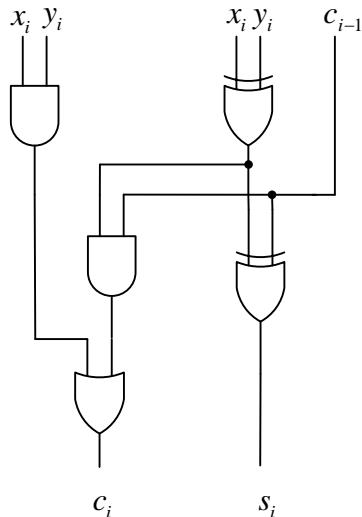
ή

$$c_i = (\bar{x}_i y_i + x_i \bar{y}_i) c_{i-1} + x_i y_i (\bar{c}_{i-1} + c_{i-1})$$

Δηλαδή,

$$c_i = (x_i \oplus y_i) c_{i-1} + x_i y_i$$

Η υλοποίηση του πλήρη προσθετή που αντιστοιχεί στις πιο πάνω εξισώσεις για τα  $s_i$ ,  $c_i$  δίδεται στο σχήμα 5.43. Η πύλη XOR χρησιμοποιείται και για τον υπολογισμό του κρατουμένου και για υπολογισμό του αθροίσματος.



**Σχήμα 5.43.** Εναλλακτική σχεδίαση του πλήρη αθροιστή

### Υλοποίηση Γ

Στη συνέχεια περιγράφεται μία διαφορετική υλοποίηση που για τον υπολογισμό του κρατουμένου εξόδου χρησιμοποιείται  $2 \rightarrow 1$  πολυπλέκτης. Η λογική παράσταση για το  $s_i$  παραμένει αυτή που προέκυψε για την υλοποίηση Α, δηλαδή

$$s_i = (x_i \oplus y_i) \oplus c_{i-1}$$

Σύμφωνα με την υλοποίηση Β για το  $c_i$  ισχύει

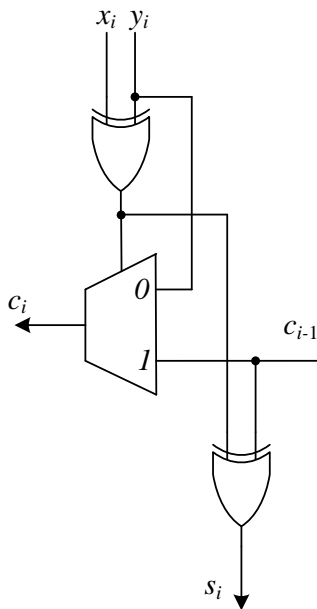
$$\begin{aligned} c_i &= x_i y_i + (x_i \oplus y_i) c_{i-1} = \\ &= 0 + x_i y_i + (x_i \oplus y_i) c_{i-1} \\ &= \bar{x}_i \bar{y}_i y_i + x_i y_i y_i + (x_i \oplus y_i) c_{i-1} = \end{aligned}$$

$$= (\bar{x}_i \bar{y}_i + x_i y_i) y_i + (x_i \oplus y_i) c_{i-1},$$

ή

$$c_i = \overline{(x_i \oplus y_i) y_i} + (x_i \oplus y_i) c_{i-1}.$$

Η συνάρτηση αυτή μπορεί να υλοποιηθεί με τη χρήση ενός  $2 \rightarrow 1$  πολυπλέκτη του οποίου η είσοδος ελέγχου είναι η έξοδος της πύλης που υλοποιεί τη συνάρτηση  $x_i \oplus y_i$  και είσοδοι δεδομένων οι  $y_i, c_{i-1}$ . Η υλοποίηση του πλήρη αθροιστή που αντιστοιχεί στις πιο πάνω εξισώσεις δίδεται στο σχήμα 5.42. Με δεδομένο ότι με τις σύγχρονες τεχνολογίες υλοποίησης μπορούν να κατασκευασθούν πολύ γρήγοροι  $2 \rightarrow 1$  πολυπλέκτες, με χρήση των προσθετών του σχήματος 5.42 μπορεί να κατασκευασθούν γρήγοροι προσθετές με διάδοση κρατουμένου.

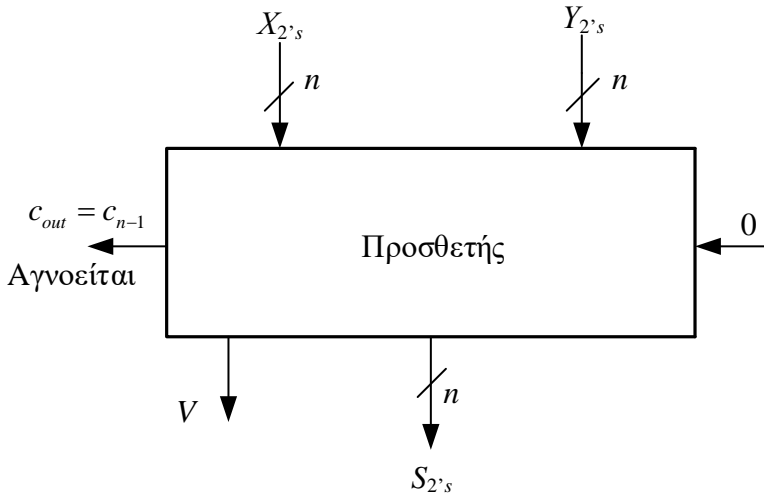


Σχήμα 5.44. Σχεδίαση πλήρη αθροιστή με χρήση πολυπλέκτη

### 5.10.2 Κύκλωμα πρόσθεσης αριθμών σε σύστημα συμπληρώματος του 2

Έστω  $X_{2's} = x_{n-1}x_{n-2} \dots x_1x_0$ ,  $Y_{2's} = y_{n-1}y_{n-2} \dots y_1y_0$  οι παραστάσεις με  $n$  bit δύο προσημασμένων αριθμών  $X, Y$  σε σύστημα συμπληρώματος του 2. Σύμφωνα με το θεώρημα 2.6 η παράσταση του αθροίσματός τους  $S_{2's} = s_{n-1}s_{n-2} \dots s_1s_0$  σε σύστημα συμπληρώματος του 2 μπορεί να υπολογισθεί με τη χρήση του προσθετή μη προσημασμένων αριθμών του σχήματος 5.35 ή του σχήματος 5.37 με κρατούμενο εισόδου 0, αγνοώντας το κρατούμενο εξόδου, όπως δείχνεται στο σχήμα 5.45.

Στην πρόσθεση σε συμπλήρωμα του 2 για ένδειξη ότι τα  $n$  bit του αθροίσματος δεν επαρκούν για να αναπαρασταθεί το αποτέλεσμα χρησιμοποιείται αντί του κρατουμένου εξόδου η υπερχείλιση  $V$  η οποία αναλύεται στην συνέχεια.



*Σχήμα 5.45. Προσθετής προσημασμένων αριθμών σε σύστημα συμπληρώματος του 2*

### Κυκλώματα ανίχνευσης υπερχείλισης

Κατά την πρόσθεση  $n$ -ψηφίων προσημασμένων αριθμών σε σύστημα συμπληρώματος του 2 υπερχείλιση συμβαίνει, όπως προαναφέραμε, όταν προστίθενται δύο θετικοί αριθμοί και προκύπτει αρνητικό αποτέλεσμα ή όταν προστίθενται αρνητικοί αριθμοί και προκύπτει θετικό αποτέλεσμα. Έστω  $V$  η ένδειξη υπερχείλισης. Ισχύει ο επόμενος πίνακας αληθείας

$x_{n-1}$	$y_{n-1}$	$s_{n-1}$	$V$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Από τον πίνακα αληθείας προκύπτει ότι το σήμα υπερχείλισης  $V$  υπολογίζεται από τα περισσότερα σημαντικά ψηφία των προσθετέων και του αποτελέσματος σύμφωνα με τη σχέση

$$V = \bar{x}_{n-1}\bar{y}_{n-1}s_{n-1} + x_{n-1}y_{n-1}\bar{s}_{n-1}$$

Στην περίπτωση που είναι διαθέσιμα τα εσωτερικά κρατούμενα του προσθετή, ισχύει ο πίνακα αληθείας που δίδεται στην συνέχεια

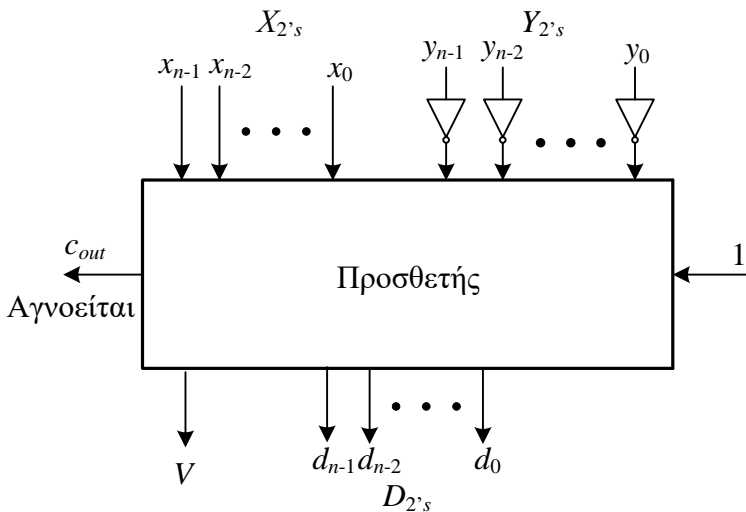
$x_{n-1}$	$y_{n-1}$	$c_{n-2}$	$c_{n-1}$	$s_{n-1}$	$V$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

Από τον προηγούμενο πίνακα αληθείας προκύπτει εύκολα ότι η ένδειξη υπερχείλισης είναι

$$V = c_{n-1} \oplus c_{n-2}$$

### 5.11 Κύκλωμα αφαίρεσης αριθμών σε σύστημα συμπληρώματος του 2

Έστω  $X_{2^s} = x_{n-1}x_{n-2} \dots x_0$ ,  $Y_{2^s} = y_{n-1}y_{n-2} \dots y_0$ , οι αναπαραστάσεις με  $n$  bit δύο προσημασμένων αριθμών  $X, Y$  σε σύστημα συμπληρώματος του 2. Σύμφωνα με τα όσα αναφέρθηκαν στην παράγραφο 2.6 ισχύει  $D_{2^s} = (X - Y)_{2^s} = (X + (-Y)_{2^s})$ , ή  $D_{2^s} = X_{2^s} + \bar{Y}_{2^s} + 1$ , όπου  $\bar{Y}_{2^s}$  είναι το συμπλήρωμα ως προς 1 του  $Y_{2^s}$ , ενώ το κρατούμενο εξόδου της πρόσθεσης αγνοείται. Επομένως η αφαίρεση μπορεί να γίνει με τη χρήση του προσθετή μη προσημασμένων αριθμών του σχήματος 5.36 που μπορεί εύκολα να προκύψει αντιστέφοντας τα ψηφία του  $Y_{2^s}$ , θέτοντας το κρατούμενο εισόδου 1 και αγνοώντας το κρατούμενο εξόδου, όπως δείχνεται στο σχήμα 5.47.



Σχήμα 5.47. Αφαιρέτης αριθμών σε σύστημα συμπληρώματος του 2

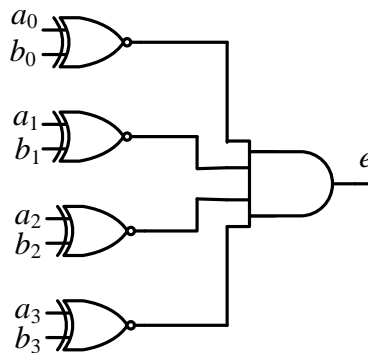
### 5.12 Συγκριτές δυαδικών αριθμών

#### Συγκριτές ισότητας μη προσημασμένων δυαδικών αριθμών

Οι συγκριτές ισότητας δύο μη προσημασμένων δυαδικών αριθμών των  $n$ -bit θα δίνει ένδειξη ισότητας όταν οι αριθμοί αυτοί είναι ίσοι ψηφίο προς ψηφίο.

**Παράδειγμα 5.20.** Να σχεδιασθεί συγκριτής ισότητας τετραψηφίων μη προσημασμένων δυαδικών αριθμών.

Έστω  $A = a_3a_2a_1a_0$ ,  $B = b_3b_2b_1b_0$  οι δύο τετραψήφιοι μη προσημασμένοι δυαδικοί αριθμοί. Οι αριθμοί αυτοί θα είναι ίσοι εάν κάθε ψηφίο του ενός είναι ίσο με το αντίστοιχο ψηφίο του άλλου. Η ανίχνευση ισότητας ψηφίο προς ψηφίο επιτυγχάνεται με πύλες XNOR, των οποίων η έξοδος γίνεται 1 εάν και οι δύο εισοδοί είναι ίσες. Επομένως η ισότητα των  $A, B$  διαπιστώνεται με το λογικό AND των εξόδων των πυλών XNOR. Η σχεδίαση του κυκλώματος δίδεται στην συνέχεια. Η έξοδος  $e$  γίνεται 1 εάν  $A=B$ .

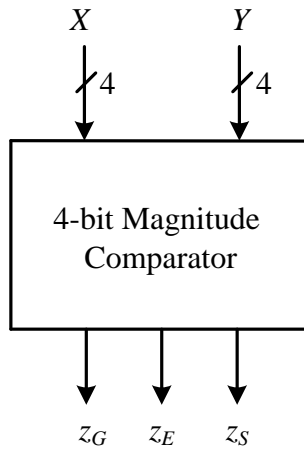


### Συγκριτές μεγέθους μη προσημασμένων αριθμών

Οι *συγκριτές μεγέθους* (*magnitude comparators*) είναι συνδυαστικά κυκλώματα που έχουν σαν εισόδους δύο μη προσημασμένους δυαδικούς αριθμούς των  $n$ -bit,  $X = x_{n-1}x_{n-2} \dots x_1x_0$  και  $Y = y_{n-1}y_{n-2} \dots y_1y_0$  και μία έξοδο  $Z = z_Gz_Ez_S$  που δείχνει τη σχέση μεγέθους μεταξύ των αριθμών  $X, Y$ . Για την έξοδο  $Z$  οι τυπικές τιμές είναι

$$Z = \begin{cases} 100, & \text{εάν } X > Y \\ 010, & \text{εάν } X = Y \\ 001, & \text{εάν } X < Y \end{cases}$$

Στο σχήμα 5.48 δίδεται το λογικό σύμβολο συγκριτή μεγέθους μη προσημασμένων δυαδικών αριθμών των 4-bit.



**Σχήμα 5.48.** Λογικό σύμβολο για τον συγκριτή μεγέθους των 4-bit

Για τη σχεδίαση των συγκριτών μεγέθους χρησιμοποιούνται οι επόμενες συναρτήσεις σύγκρισης δυαδικών ψηφίων:

$$G_i = x_i \bar{y}_i, \quad (x_i > y_i)$$

$$E_i = x_i \oplus y_i, \quad (x_i = y_i)$$

$$S_i = \bar{x}_i \cdot y_i, \quad (x_i < y_i)$$

**Παράδειγμα 5.21.** Να σχεδιασθεί συγκριτής μεγέθους δύο μονοψήφιων δυαδικών αριθμών.

Έστω οι αριθμοί  $X = x_0$ ,  $Y = y_0$ , και έστω ότι η έξοδος  $z_G$  γίνεται 1 όταν  $X > Y$ , η  $z_E$  όταν  $X = Y$  και η  $z_S$  όταν  $X < Y$ . Στη συνέχεια δίδεται ο πίνακας αληθείας του συγκριτή μονοψήφιων αριθμών.

$x_0$	$y_0$	$z_G$	$z_E$	$z_S$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

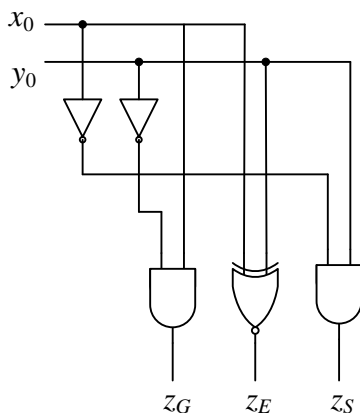
Από τον πίνακα αληθείας προκύπτουν οι επόμενες εξισώσεις για τα  $z_E$ ,  $z_G$ ,  $z_S$

$$z_G = x_0 \bar{y}_0 \quad (x_0 > y_0)$$

$$z_E = x_0 \oplus y_0 \quad (x_0 = y_0)$$

$$z_S = \bar{x}_0 y_0 \quad (x_0 < y_0)$$

Η λογική σχεδίαση του συγκριτή μεγέθους μονοψηφίων δυαδικών αριθμών δίδεται στο σχήμα 5.49.



**Σχήμα 5.49.** Συγκριτής μεγέθους μονοψηφίων δυαδικών αριθμών

**Παράδειγμα 5.22.** Να σχεδιασθεί συγκριτής μεγέθους δύο διψήφιων δυαδικών αριθμών.

Έστω οι δυαδικοί αριθμοί  $X = x_1 x_0$  και  $Y = y_1 y_0$ , και έστω ότι η έξοδος  $z_G$  γίνεται 1 όταν  $X > Y$ , η έξοδος  $z_E$  όταν  $X = Y$  και η έξοδος  $z_S$  όταν  $X < Y$ .

Για τα bit εξόδου του συγκριτή ισχύουν

$$z_G = 1 \text{ εάν } x_1 > y_1 \text{ ή } (x_1 = y_1 \text{ και } x_0 > y_0).$$

$$z_E = 1 \text{ εάν } x_1 = y_1 \text{ και } x_0 = y_0.$$

$$z_S = 1 \text{ εάν } x_1 < y_1 \text{ ή } (x_1 = y_1 \text{ και } x_0 < y_0).$$

Η αντίστοιχα

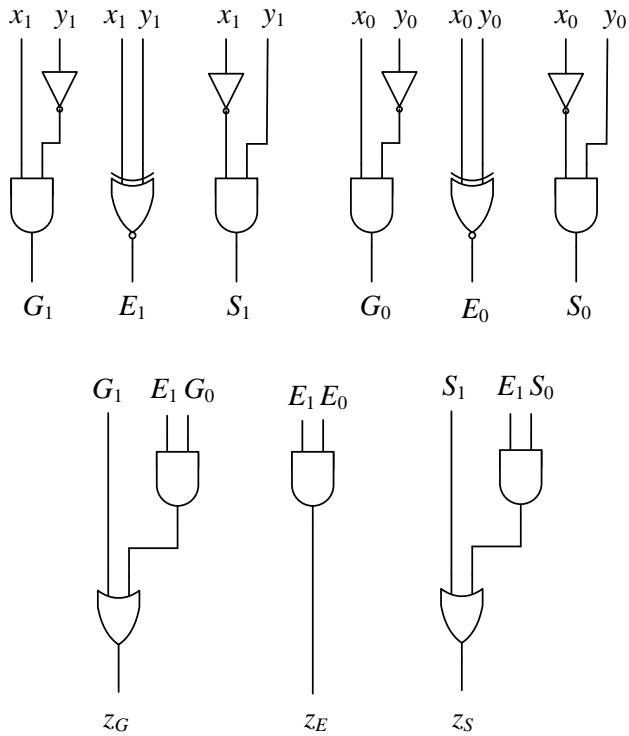
$$z_G = x_1 \bar{y}_1 + \overline{(x_1 \oplus y_1)} x_0 \bar{y}_0 = G_1 + E_1 G_0$$

$$z_E = \overline{(x_1 \oplus y_1)} (x_0 \oplus y_0) = E_1 E_0$$

$$z_S = \bar{x}_1 y_1 + \overline{(x_1 \oplus y_1)} \bar{x}_0 y_0 = S_1 + E_1 S_0$$

Το κύκλωμα του συγκριτή διψήφιων αριθμών δίδεται στο σχήμα 5.50.

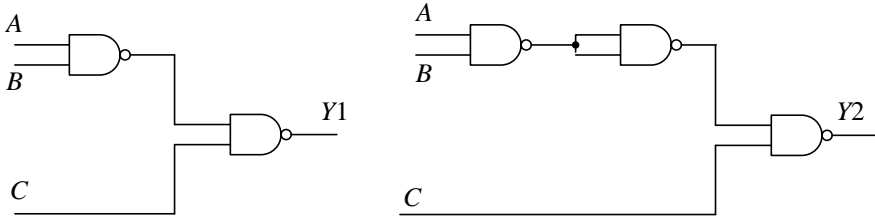




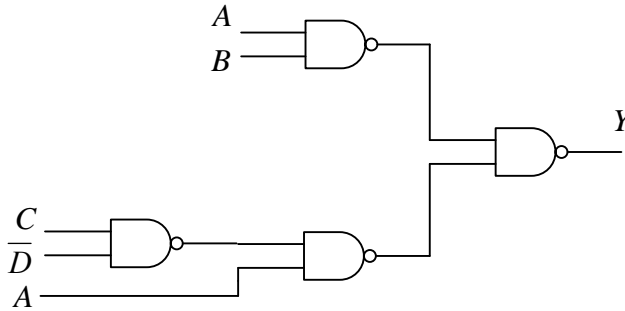
Σχήμα 5.50. Συγκριτής μεγέθους διψήφιων δυαδικών αριθμών

## Ασκήσεις

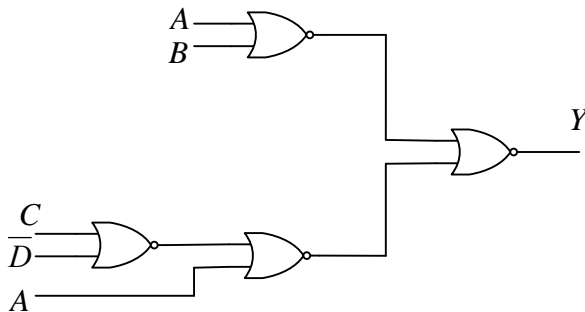
5.1 Να αναλυθούν τα λογικά κυκλώματα που δίδονται στην συνέχεια.



5.2 Να αναλυθεί το λογικό κύκλωμα που δίδεται στη συνέχεια. Δηλαδή να βρεθεί στην κανονική της μορφή η λογική συνάρτηση που υλοποιεί και ο αντίστοιχος πίνακας αληθείας.

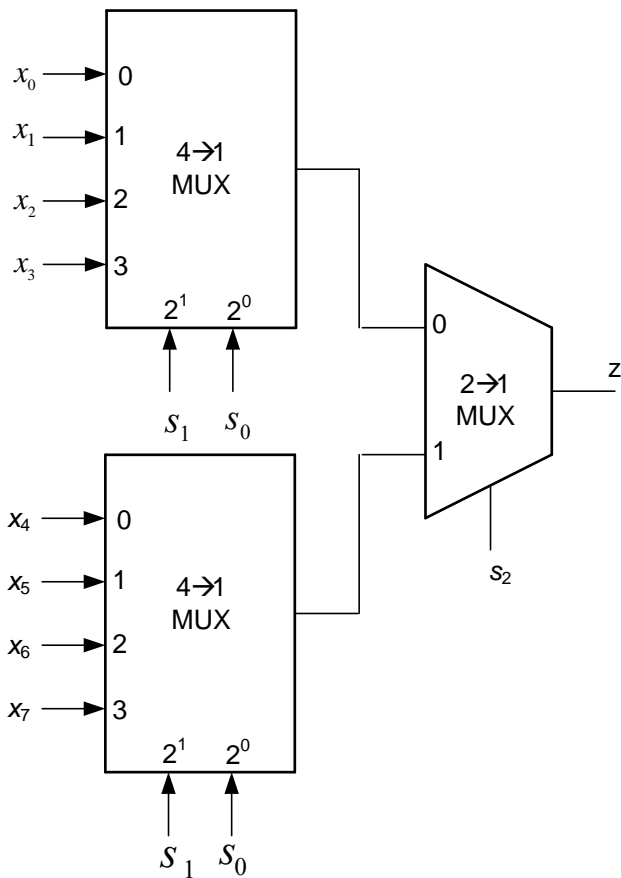


5.3 Να αναλυθεί το λογικό κύκλωμα που δίδεται στη συνέχεια. Δηλαδή να βρεθεί στην κανονική της μορφή η λογική συνάρτηση που υλοποιεί και ο αντίστοιχος πίνακας αληθείας.

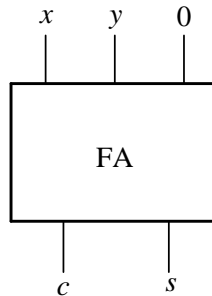


5.4 Να σχεδιασθεί ένα κύκλωμα ανίχνευσης μηδενικής τιμής για μη προσημασμένους αριθμούς των 4 bit. Ένδειξη του μηδενός να θεωρηθεί το λογικό 1.

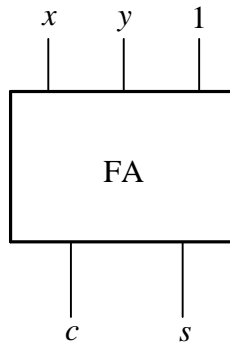
- 5.5 Να σχεδιασθεί ένα κύκλωμα ανίχνευσης μηδενικής τιμής για προσημασμένους αριθμούς των 4 bits. Ένδειξη του μηδενός να θεωρηθεί το λογικό 1. Θεωρήστε ότι οι αριθμοί είναι σε σύστημα προσημασμένου συμπληρώματος του 2.
- 5.6 Να σχεδιασθεί ένα κύκλωμα ανίχνευσης μηδενικής τιμής για προσημασμένους αριθμούς των 4 bits. Ένδειξη του μηδενός να θεωρηθεί το λογικό 1. Θεωρήστε ότι οι αριθμοί είναι σε σύστημα προσημασμένου μέτρου.
- 5.7 Να σχεδιασθεί ένα κύκλωμα ανίχνευσης μηδενικής τιμής για προσημασμένους αριθμούς των 4 bits. Ένδειξη του μηδενός να θεωρηθεί το λογικό 1. Θεωρήστε ότι οι αριθμοί είναι σε σύστημα προσημασμένου συμπληρώματος του 2.
- 5.8 Να δοθεί ο πίνακας αληθείας και στη συνέχεια να σχεδιασθεί με λογικές πύλες το κύκλωμα ενός  $2 \rightarrow 4$  αποκωδικοποιητή.
- 5.9 Να δοθεί ο πίνακας αληθείας και στη συνέχεια να σχεδιασθεί το κύκλωμα ενός  $2 \rightarrow 4$  αποκωδικοποιητή με είσοδο επίτρησης.
- 5.10 Να δοθεί ο πίνακας αληθείας ενός  $3 \rightarrow 8$  αποκωδικοποιητή. Στη συνέχεια να σχεδιασθεί το κύκλωμα του αποκωδικοποιητή χρησιμοποιώντας σαν στοιχείο σχεδίασης τον  $2 \rightarrow 4$  αποκωδικοποιητή με είσοδο επίτρησης. Έχετε στη διάθεσή σας και λογικές πύλες.
- 5.11 Να δοθεί ο πίνακας αληθείας ενός  $3 \rightarrow 8$  αποκωδικοποιητή με είσοδο επίτρησης. Στη συνέχεια να σχεδιασθεί ισοδύναμο κύκλωμα χρησιμοποιώντας σαν στοιχείο σχεδίασης τον  $2 \rightarrow 4$  αποκωδικοποιητή με είσοδο επίτρησης. Έχετε στη διάθεσή σας και λογικές πύλες.
- 5.12 Να δοθεί ο πίνακας αληθείας του  $4 \rightarrow 1$  κωδικοποιητή προτεραιότητας με είσοδο επίτρησης.
- 5.13 Να δοθεί ο πίνακας αληθείας και στη συνέχεια να σχεδιασθεί ένας  $2 \rightarrow 1$  πολυπλέκτης με είσοδο επίτρησης.
- 5.14 Να δοθεί ο πίνακας αληθείας και στη συνέχεια να σχεδιασθεί ένας  $4 \rightarrow 1$  πολυπλέκτης.
- 5.15 Να δοθεί ο πίνακας αληθείας και στη συνέχεια να σχεδιασθεί ένας  $4 \rightarrow 1$  πολυπλέκτης με είσοδο επίτρησης.
- 5.16 Να σχεδιασθεί ένας  $4 \rightarrow 1$  πολυπλέκτης χρησιμοποιώντας σαν στοιχείο σχεδίασης τον  $2 \rightarrow 1$  πολυπλέκτη.
- 5.17 Να δοθεί ο πίνακας λειτουργίας του κυκλώματος που δίδεται στην συνέχεια.



- 5.18 Να υλοποιηθεί η λογική συνάρτηση  $Y = ABC\bar{C} + A\bar{B}C$  χρησιμοποιώντας σαν στοιχείο σχεδίασης τον 4→1 πολυπλέκτη.
- 5.19 Να υλοποιηθεί η λογική συνάρτηση  $Y = AB + A\bar{B}C$  χρησιμοποιώντας σαν στοιχείο σχεδίασης τον 4→1 πολυπλέκτη.
- 5.20 Να σχεδιασθεί κύκλωμα ισοδύναμο με έναν ημιαθροιστή. Έχετε στη διάθεσή σας έναν 2→4 αποκωδικοποιητή και πύλες OR.
- 5.21 Να σχεδιασθεί κύκλωμα ισοδύναμο ενός πλήρη αθροιστή. Έχετε στη διάθεσή σας έναν 3→8 αποκωδικοποιητή και πύλες OR.
- 5.22 Να δοθεί ο πίνακας αληθείας του κυκλώματος που δίδεται στη συνέχεια. Ακολουθώς να σχεδιασθεί ισοδύναμο κύκλωμα χρησιμοποιώντας απλές λογικές πύλες. Σχολιάστε.



5.23 Να δοθεί ο πίνακας αληθείας του κυκλώματος που δίδεται στη συνέχεια. Ακολουθώς να σχεδιαστεί ισοδύναμο κύκλωμα χρησιμοποιώντας απλές λογικές πύλες.

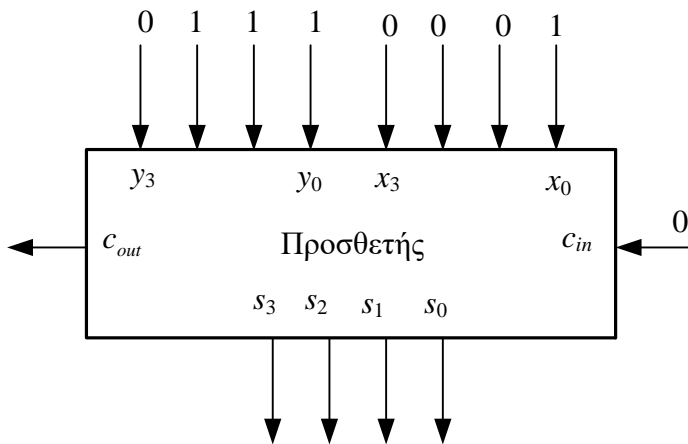


5.24 Να σχεδιασθεί ένας προσθετής των 4 bit χωρίς κρατούμενο εισόδου χρησιμοποιώντας σαν στοιχεία σχεδίασης τον πλήρη αθροιστή και τον ημιαθροιστή.

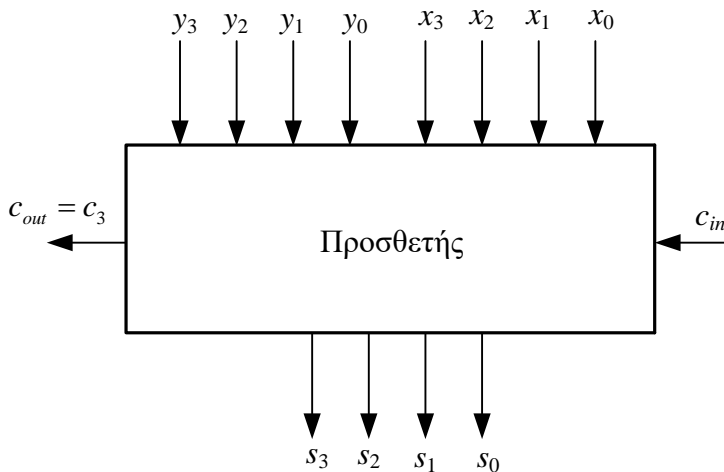
5.25 Να σχεδιασθεί κύκλωμα αύξησης ενός τετράμπιτου δυαδικού αριθμού κατά 1. Έχετε στη διάθεσή σας ημιαθροιστές.

5.26 Να σχεδιασθεί κύκλωμα αύξησης ενός οκτάμπιτου δυαδικού αριθμού κατά 4. Έχετε στη διάθεσή σας ημιαθροιστές.

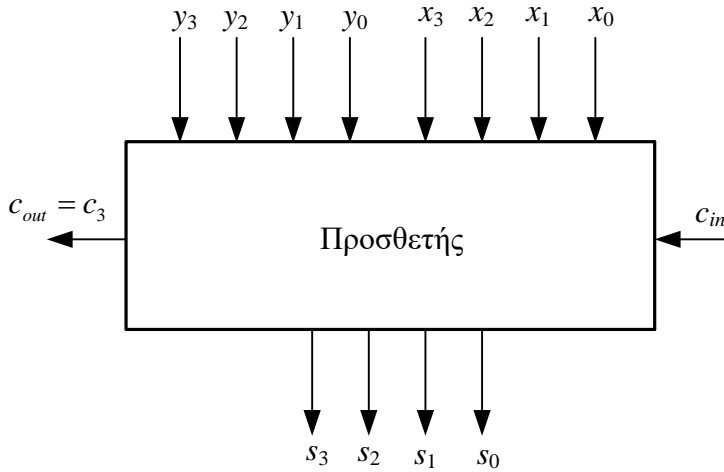
5.27 Να υπολογισθούν οι έξοδοι του προσθετή των 4 bit που δίδεται στην συνέχεια για τις δοσμένες εισόδους.



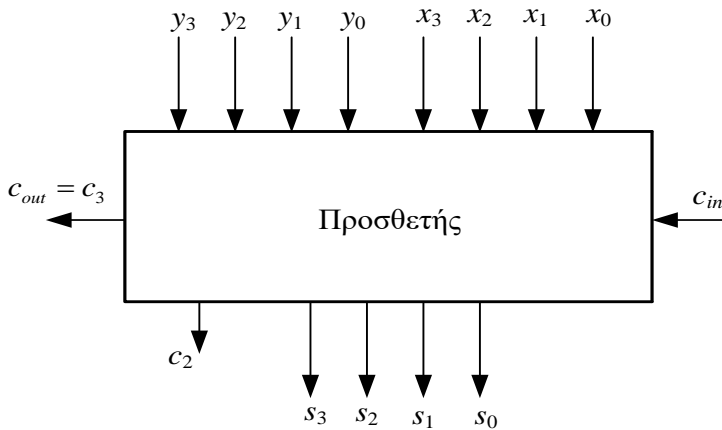
5.28 Στον προσθετή των 4 bit που δίδεται στη συνέχεια να γίνουν οι κατάλληλες προσθήκες ώστε να έχει ένδειξη μηδενικής τιμής στην περίπτωση που θα χρησιμοποιηθεί για πρόσθεση μη προσημασμένων αριθμών ή αριθμών σε σύστημα συμπληρώματος του 2.



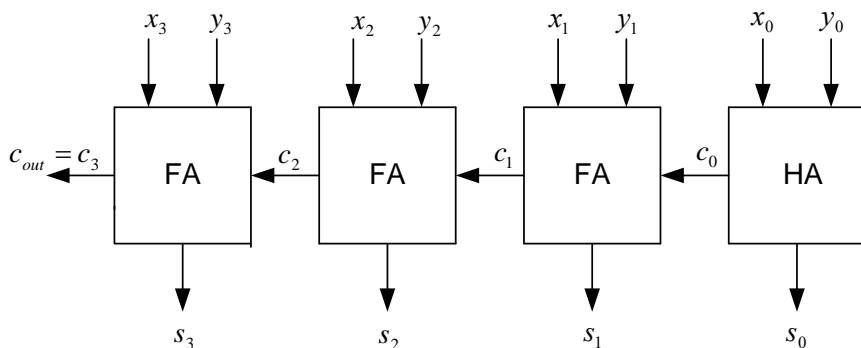
5.29 Στον προσθετή των 4 bit που δίδεται στη συνέχεια να γίνουν οι κατάλληλες προσθήκες ώστε να έχει ένδειξη μηδενικής τιμής στην περίπτωση που θα χρησιμοποιηθεί για πρόσθεση μη προσημασμένων αριθμών ή αριθμών σε σύστημα συμπληρώματος του 2.



5.30 Στον προσθετή των 4 bit που δίδεται στη συνέχεια να γίνουν οι κατάλληλες προσθήκες ώστε να έχει ένδειξη υπερχείλισης στην περίπτωση που θα χρησιμοποιηθεί για πρόσθεση αριθμών σε σύστημα συμπληρώματος του 2.

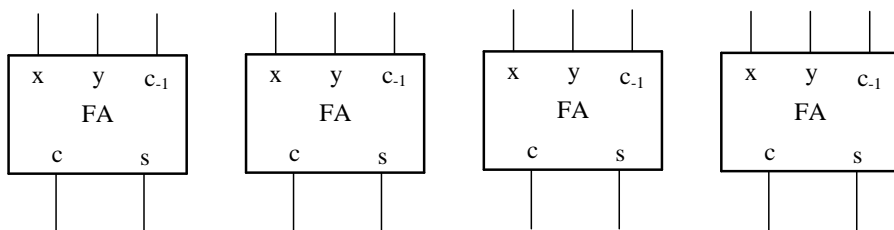


5.31 Στον προσθετή των 4 bit που δίδεται στην συνέχεια να προστεθεί το κατάλληλο κύκλωμα ώστε να έχει ένδειξη υπερχείλισης για να χρησιμοποιείται και για την πρόσθεση αριθμών σε παράσταση συμπληρώματος του 2.

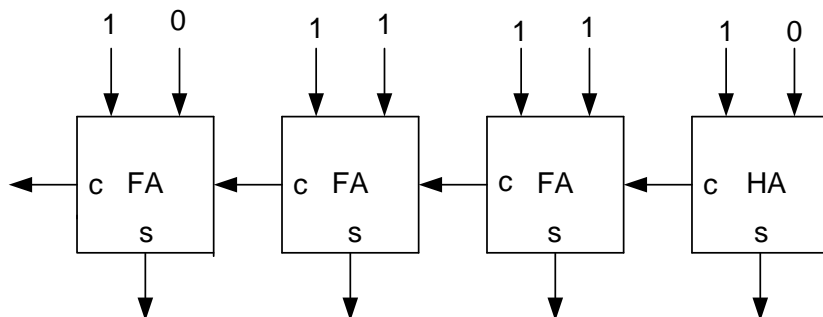


5.32 Να σχεδιασθεί ένα κύκλωμα πρόσθεσης/αφαίρεσης για αριθμούς των 4 bit σε σύστημα συμπληρώματος του 2.

5.33 Στο σχήμα που δίδεται στην συνέχεια να γίνουν οι κατάλληλες συνδέσεις ώστε να υλοποιηθεί ένας προσθετής των τεσσάρων bit. Σημειώστε το κρατούμενο εισόδου και το κρατούμενο εξόδου.



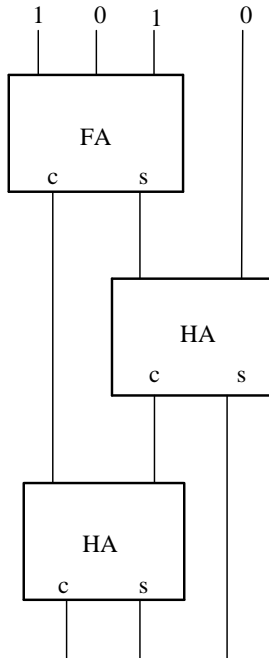
5.34 Να υπολογισθούν οι έξοδοι των FA και του HA για τις δοσμένες εισόδους.



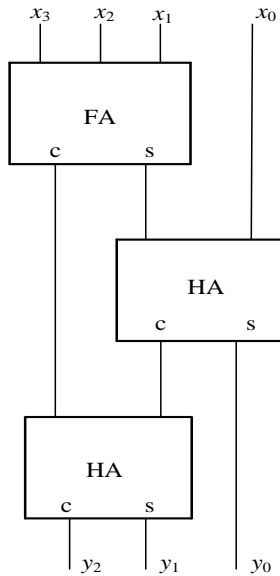
5.35 Να σχεδιασθεί κύκλωμα με 3 εισόδους το οποίο να υπολογίζει τον αριθμό των 1 στις εισόδους του.

5.36 Να δοθούν οι έξοδοι των FA και των HA του κυκλώματος που δίδεται στην συνέχεια για την δοσμένη είσοδο.





5.37 Να δοθεί ο πίνακας αληθείας του κυκλώματος που δίδεται στη συνέχεια. Σχολιάστε.



5.38 Να σχεδιασθεί ένα κύκλωμα το οποίο να δείχνει πότε ένας διψήφιος δυαδικός αριθμός  $X=x_1x_0$  είναι μικρότερος από έναν αριθμό  $Y=y_1y_0$ .

