

Κεφάλαιο 2. Ψηφιακή Σχεδίαση

Το Κεφάλαιο αυτό αποτελεί μια εισαγωγή στο αντικείμενο της ψηφιακής σχεδίασης. Τα θέματα στα οποία θα αναφερθούμε περιλαμβάνουν την άλγεβρα Boole, τις λογικές πύλες, τα ολοκληρωμένα κυκλώματα, τις ελαχιστόρους και τους μεγιστόρους, την ανάλυση και σχεδίαση συνδυαστικών κυκλωμάτων, καθώς και την απλοποίηση λογικών συναρτήσεων. Επίσης, θα αναφερθούμε σε βασικές ακολουθιακές μονάδες, καθώς και στην ανάλυση και σύνθεση σύγχρονων ακολουθιακών κυκλωμάτων.

2.1 Αναλογικά και ψηφιακά μεγέθη

Αναλογικό μέγεθος ονομάζεται ένα μέγεθος που μπορεί να πάρει οποιαδήποτε τιμή σε μία περιοχή τιμών. Για παράδειγμα, αναλογικά μεγέθη είναι: η ταχύτητα ενός αυτοκινήτου, η θερμοκρασία ενός δωματίου, το βάρος ενός ανθρώπου, το ύψος ενός δένδρου. Έτσι, κατά την επιτάχυνση ενός αυτοκινήτου από 0 χλμ/ώρα (αρχική ταχύτητα) έως 100 χλμ/ώρα (τελική ταχύτητα), η ταχύτητά του λαμβάνει όλες τις δυνατές τιμές στο διάστημα από 0 χλμ/ώρα έως 100 χλμ/ώρα (*άπειρο πλήθος τιμών*).

Ψηφιακό μέγεθος ονομάζεται το μέγεθος που μπορεί να πάρει συγκεκριμένες (διακριτές) τιμές σε μία περιοχή τιμών. Για παράδειγμα, ψηφιακά μεγέθη είναι: το πλήθος των «φάουλ» ενός παίκτη μπάσκετ κατά τη διάρκεια ενός αγώνα, οι βαθμοί μίας ομάδας ποδοσφαίρου κατά τη διάρκεια του πρωταθλήματος. Έτσι, κατά τη διάρκεια ενός αγώνα μπάσκετ, ένας παίκτης μπορεί να κάνει 1, 2, 3, 4 ή 5 «φάουλ» (*καθορισμένο πλήθος διακριτών τιμών*).

Δυαδικό μέγεθος είναι ένα ψηφιακό μέγεθος που μπορεί να πάρει μόνο δύο (2) διακριτές τιμές. Για παράδειγμα, δυαδικά μεγέθη είναι: η λογική πρόταση «σήμερα βρέχει» (η λογική πρόταση μπορεί να είναι αληθής (TRUE) αν πράγματι βρέχει ή ψευδής (FALSE) αν δεν βρέχει), η κατάσταση ενός λαμπτήρα (ο λαμπτήρας μπορεί να είναι αναμμένος (ON) ή σβηστός (OFF)), η κατάσταση ενός διακόπτη (ο διακόπτης μπορεί να είναι ανοικτός ή κλειστός όπως φαίνεται στην επόμενη Εικόνα).



Εικόνα: Δυαδικό μέγεθος

Τα ηλεκτρονικά κυκλώματα κατατάσσονται σε δύο βασικές κατηγορίες, ανάλογα με τα σήματα που επεξεργάζονται: αναλογικά κυκλώματα (analog circuits) και ψηφιακά κυκλώματα (digital circuits)

2.2 Άλγεβρα Boole

Η Άλγεβρα Boole (Boolean algebra) πήρε το όνομά της από τον G. Boole (1815-1864), ο οποίος ανέπτυξε ένα αλγεβρικό σύστημα (1854) για τη συστηματική αντιμετώπιση της λογικής. Τα αξιώματα της Άλγεβρας Boole διατυπώθηκαν από τον E. V. Huntington (1904). Οι μεταβλητές που χρησιμοποιούνται στην Άλγεβρα Boole ονομάζονται λογικές μεταβλητές γιατί μπορούν να πάρουν δύο (2) μόνο τιμές: 0 και 1. Αυτός είναι ο λόγος που η Άλγεβρα Boole αποτελεί τη βάση για τα ψηφιακά ηλεκτρονικά κυκλώματα. Στην Άλγεβρα Boole ορίζονται τρεις βασικές πράξεις:

⇒ η πράξη NOT (OXI) με σύμβολο –

⇒ η πράξη AND (ΚΑΙ) με σύμβολο \cdot

⇒ η πράξη OR (Η) με σύμβολο $+$

Η πράξη NOT

Στην πράξη NOT συμμετέχει μία μόνο λογική μεταβλητή και το αποτέλεσμα της πράξης είναι το συμπλήρωμα (αντίστροφο) της μεταβλητής αυτής, δηλαδή αν η μεταβλητή έχει την τιμή “0”, τότε το αποτέλεσμα είναι “1” και αντίστροφα αν η μεταβλητή έχει την τιμή “1”, τότε το αποτέλεσμα είναι “0”. Αν A είναι μία λογική μεταβλητή, τότε η πράξη NOT εκφράζεται με τη σχέση:

$$Y = \bar{A}$$

Ο πίνακας αληθείας της πράξης NOT παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας Αληθείας της πράξης NOT

A	$Y = \bar{A}$
0	1
1	0

Η πράξη AND

Στην πράξη AND συμμετέχουν δύο λογικές μεταβλητές και το αποτέλεσμα της πράξης είναι “1”, αν και οι δύο μεταβλητές είναι “1”. Αν A και B είναι δύο λογικές μεταβλητές, τότε η πράξη AND εκφράζεται με τη σχέση:

$$Y = A \cdot B$$

Το σύμβολο της πράξης AND (\cdot) μπορεί να παραλείπεται στις εκφράσεις της Άλγεβρας Boole ($A \cdot B = AB$). Ο πίνακας αληθείας της πράξης AND παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας Αληθείας της πράξης AND

A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Η πράξη OR

Στην πράξη OR συμμετέχουν δύο λογικές μεταβλητές και το αποτέλεσμα της πράξης είναι “1”, αν τουλάχιστον μία από τις δύο μεταβλητές είναι “1”. Αν A και B είναι δύο λογικές μεταβλητές, τότε η πράξη OR εκφράζεται με τη σχέση:

$$Y = A + B$$

Ο πίνακας αληθείας της πράξης OR παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας Αληθείας της πράξης OR

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Αξιώματα Huntington

Ουδέτερα στοιχεία των πράξεων AND και OR. Το ουδέτερο στοιχείο της πράξης AND είναι το 1 και το ουδέτερο στοιχείο της πράξης OR είναι το 0.

$$x \cdot 1 = 1 \cdot x = x$$

$$x + 0 = 0 + x = x$$

Το αξίωμα αυτό μπορεί να επαληθευτεί από τους πίνακες αληθείας των πράξεων AND και OR, από όπου φαίνεται ότι:

$$0 \cdot 1 = 1 \cdot 0 = 0 \text{ και } 1 \cdot 1 = 1$$

$$0 + 0 = 0 \text{ και } 1 + 0 = 0 + 1 = 1$$

2. Αντιμεταθετική ιδιότητα των πράξεων AND και OR

Οι πράξεις AND και OR έχουν την αντιμεταθετική ιδιότητα, δηλαδή ισχύει.

$$x \cdot y = y \cdot x$$

$$x + y = y + x$$

Επιμεριστική ιδιότητα των πράξεων AND και OR

Η πράξη AND έχει την επιμεριστική ιδιότητα ως προς την πράξη OR και η πράξη OR έχει την επιμεριστική ιδιότητα ως προς την πράξη AND.

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

Συμπλήρωμα (NOT)

Κάθε λογική μεταβλητή x έχει ένα συμπλήρωμα \bar{x} ή x' με τις ακόλουθες ιδιότητες:

$$x \cdot \bar{x} = 0$$

$$x + \bar{x} = 1$$

Το αξίωμα αυτό μπορεί να επαληθευτεί από τους πίνακες αληθείας της πράξης NOT, από όπου φαίνεται ότι:

$$0 \cdot \bar{0} = 0 \cdot 1 = 0 \text{ και } 1 \cdot \bar{1} = 1 \cdot 0 = 0$$

$$0 + \bar{0} = 0 + 1 = 1 \text{ και } 1 + \bar{1} = 1 + 0 = 1$$

Αρχή Διϊσμού

Η ισχύς των εκφράσεων της Άλγεβρας Boole εξακολουθεί να υφίσταται, αν γίνει αλλαγή των πράξεων AND και OR και των ουδέτερων στοιχείων μεταξύ τους ($\cdot \leftrightarrow +$ και $0 \leftrightarrow 1$). Για παράδειγμα, αν ισχύει η έκφραση $x+1=1$, τότε ισχύει και η έκφραση $x \cdot 0=0$ και η μία έκφραση ονομάζεται δυτική της άλλης.

Θεωρήματα Άλγεβρας Boole

Θεώρημα 1. Ισχύουν οι σχέσεις:

α. $x \cdot x = x$

β. $x+x=x$

Θεώρημα 2. Ισχύουν οι σχέσεις:

α. $x \cdot 0 = 0$

β. $x+1=1$

Θεώρημα 3. Ισχύουν οι σχέσεις:

$\bar{\bar{x}} = x$

Θεώρημα 4. Προσεταιριστική ιδιότητα

α. $x \cdot y \cdot z = x \cdot (y \cdot z) = (x \cdot y) \cdot z$

β. $x+y+z = x+(y+z) = (x+y)+z$

Θεώρημα 5. Θεώρημα απορρόφησης

α. $x+x \cdot y = x$

β. $x \cdot (x+y) = x$

Θεώρημα 6. Θεώρημα De Morgan

α. $\overline{x \cdot y} = \bar{x} + \bar{y}$

β. $\overline{x+y} = \bar{x} \cdot \bar{y}$

Αξίζει να σημειωθεί ότι το Θεώρημα De Morgan ισχύει και για περισσότερες από δύο μεταβλητές, για παράδειγμα:

α. $\overline{x \cdot y \cdot z} = \bar{x} + \bar{y} + \bar{z}$

β. $\overline{x+y+z} = \bar{x} \cdot \bar{y} \cdot \bar{z}$

Προτεραιότητα πράξεων

Για την εκτέλεση των πράξεων στις εκφράσεις της Άλγεβρας Boole είναι ανάγκη να καθορισθεί η προτεραιότητα της εκτέλεσής τους, όπως γίνεται στην γνωστή από τα μαθηματικά άλγεβρα. Η προτεραιότητα των πράξεων παρουσιάζεται στον ακόλουθο Πίνακα.

Πίνακας: Προτεραιότητα πράξεων

Προτεραιότητα	Πράξη
1	()
2	NOT
3	AND
4	OR

Από τον Πίνακα προτεραιότητας των πράξεων προκύπτει ότι σε μία έκφραση της Άλγεβρας Boole εκτελούνται πρώτα οι πράξεις μέσα σε παρενθέσεις, μετά υπολογίζονται τα συμπληρώματα, στη συνέχεια εκτελούνται οι πράξεις AND και τέλος εκτελούνται οι πράξεις OR.

2.3 Λογικές πύλες

2.3.1 Διαγράμματα των λογικών πυλών


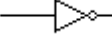
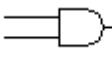
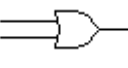
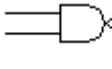
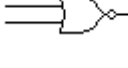
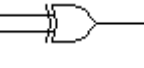
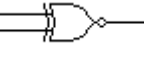
Οι λογικές πύλες είναι τα βασικά δομικά στοιχεία στα ψηφιακά κυκλώματα. Όπως έχουμε στις οικοδομές τα τούβλα και με αυτά κατασκευάζουμε τοίχους και σύνθετες κατασκευές χρησιμοποιώντας παρόμοια υλικά ξανά και ξανά, έτσι και στα ψηφιακά κυκλώματα χρησιμοποιούμε τις λογικές πύλες για να κατασκευάσουμε σύνθετα κυκλώματα. Οι λογικές πύλες μίας και δύο εισόδων παρουσιάζονται στον επόμενο Πίνακα όπου η έξοδος εκφράζεται ως συνάρτηση των εισόδων.

Πίνακας: Λογικές Πύλες - Συναρτήσεις

Λογική Πύλη	Είσοδοι	Έξοδος	Συνάρτηση
Απομονωτής Buffer	A	Y	$Y=A$
Αντιστροφέας NOT	A	Y	$Y=\bar{A}$
AND	A,B	Y	$Y=A \cdot B$
OR	A,B	Y	$Y=A+B$
NAND	A,B	Y	$Y=\overline{A \cdot B}$
NOR	A,B	Y	$Y=\overline{A+B}$
XOR	A,B	Y	$Y=A \cdot \bar{B} + \bar{A} \cdot B = A \oplus B$
XNOR	A,B	Y	$Y=A \cdot B + \bar{A} \cdot \bar{B} = \overline{A \oplus B} = A \odot B$

Τα λογικά διαγράμματα (οι συμβολισμοί) των πυλών αυτών παρουσιάζονται στον επόμενο Πίνακα.

Πίνακας: Λογικές Πύλες – Λογικά Διαγράμματα

Λογική Πύλη	Λογικό Διάγραμμα
Απομονωτής Buffer	A  Y=A
Αντιστροφέας NOT	A  Y= \bar{A}
AND	A B  Y=A·B
OR	A B  Y=A+B
NAND	A B  Y= $\overline{A \cdot B}$
NOR	A B  Y= $\overline{A + B}$
XOR	A B  Y=A⊕B
XNOR	A B  Y=A⊙B

2.3.2 Πίνακες αληθείας των λογικών πυλών

Ο απομονωτής (buffer) είναι μία πύλη με μία είσοδο και μία έξοδο που είναι ίση με την είσοδο. Η συνάρτηση του απομονωτή είναι:

$$Y=A$$

και ο πίνακας αληθείας του απομονωτή παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας αληθείας του απομονωτή

A	Y=A
0	0
1	1

Η πύλη NOT έχει μία είσοδο και μία έξοδο που είναι ίση με το συμπλήρωμα της εισόδου. Η συνάρτηση της πύλης NOT είναι $Y = \bar{A}$ και ο πίνακας αληθείας της παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας αληθείας της πύλης NOT

A	$Y = \bar{A}$
0	1
1	0

Η πύλη AND έχει δύο εισόδους και μία έξοδο που είναι “1”, αν και οι δύο εισοδοι είναι “1”. Η συνάρτηση της πύλης AND είναι: $Y=A \cdot B$ και ο πίνακας αληθείας της πύλης AND παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας Αληθείας της πύλης AND

A	B	$Y=A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Η πύλη OR έχει δύο εισόδους και μία έξοδο που είναι “1”, αν τουλάχιστον μία από τις δύο εισόδους είναι “1”. Η συνάρτηση της πύλης OR είναι $Y=A+B$ και ο πίνακας αληθείας της πύλης OR παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας Αληθείας της πύλης OR

A	B	$Y=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

Η πύλη NAND προκύπτει από μία πύλη AND ακολουθούμενη από μία πύλη NOT. Η πύλη NAND έχει δύο εισόδους και μία έξοδο που είναι “1”, αν τουλάχιστον μία από τις δύο εισόδους είναι “0”. Η συνάρτηση της πύλης NAND είναι $Y = \overline{A \cdot B}$ και ο πίνακας αληθείας της πύλης NAND παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας Αληθείας της πύλης NAND

A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

Η πύλη NOR προκύπτει από μία πύλη OR ακολουθούμενη από μία πύλη NOT. Η πύλη NOR έχει δύο εισόδους και μία έξοδο που είναι “1”, αν και οι δύο εισοδοι είναι “0”. Η συνάρτηση της πύλης NOR είναι $Y = \overline{A + B}$ και ο πίνακας αληθείας της παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας Αληθείας της πύλης NOR

A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

Η πύλη XOR (exclusive OR) έχει δύο εισόδους και μία έξοδο που είναι “1”, αν οι δύο εισοδοι είναι διαφορετικές μεταξύ τους (για αυτό ονομάζεται και πύλη διαφωνίας ή σύγκρισης). Η συνάρτηση της πύλης XOR είναι $Y=A\oplus B=A\cdot\overline{B}+\overline{A}\cdot B$ και ο πίνακας αληθείας της παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας Αληθείας της πύλης XOR

A	B	$Y=A\oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

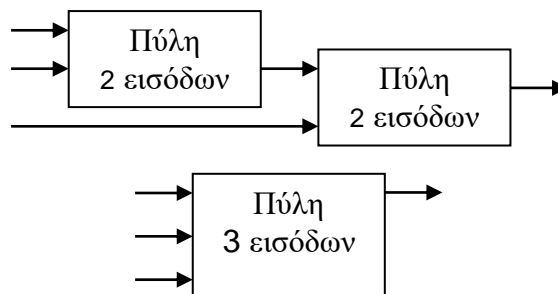
Η πύλη XNOR (exclusive NOR) έχει δύο εισόδους και μία έξοδο που είναι “1”, αν οι δύο εισοδοι είναι ίσες. Η συνάρτηση της πύλης XNOR είναι $Y=A\odot B=A\cdot B+\overline{A}\cdot\overline{B}$ και ο πίνακας αληθείας της πύλης XNOR παρουσιάζεται στον επόμενο Πίνακα.

Πίνακας: Πίνακας Αληθείας της πύλης XNOR

A	B	$Y=A\odot B$
0	0	1
0	1	0
1	0	0
1	1	1

2.3.3 Λογικές πύλες πολλών εισόδων

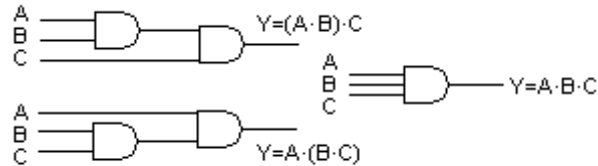
Οι πύλες δύο εισόδων μπορούν να επεκταθούν ώστε να έχουν περισσότερες από δύο εισόδους, εάν οι πράξεις τους έχουν την *αντιμεταθετική* και την *προσεταιριστική* ιδιότητα. Η υλοποίηση μίας τέτοιας πύλης τριών (3) εισόδων με χρήση ομοίων πυλών δύο (2) εισόδων παρουσιάζεται στην επόμενη Εικόνα.



Εικόνα: Τεχνική επέκτασης εισόδων πυλών

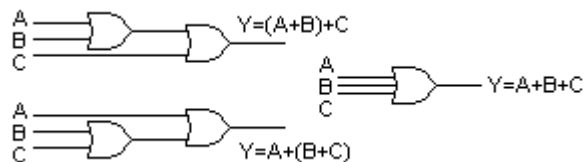
Για παράδειγμα, μία πύλη AND τριών εισόδων μπορεί να υλοποιηθεί χρησιμοποιώντας δύο πύλες AND δύο εισόδων όπως φαίνεται στην επόμενη Εικόνα, γιατί ισχύει:

- η αντιμεταθετική ιδιότητα $Y=A \cdot B=B \cdot A$ και
- η προσεταιριστική ιδιότητα $Y=A \cdot B \cdot C=(A \cdot B) \cdot C=A \cdot (B \cdot C)$



Εικόνα: Υλοποίηση πύλης AND τριών εισόδων με πύλες AND δύο εισόδων

Με την ίδια λογική, μία πύλη OR τριών εισόδων μπορεί να υλοποιηθεί χρησιμοποιώντας δύο πύλες OR δύο εισόδων όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Υλοποίηση πύλης OR τριών εισόδων με πύλες OR δύο εισόδων

Η πύλη NAND τριών εισόδων ορίζεται ως το συμπλήρωμα της πύλης AND τριών εισόδων. Επομένως, η έξοδος της πύλης NAND τριών εισόδων είναι “1”, αν τουλάχιστον μία από τις δύο εισόδους είναι “0”. Μία πύλη NAND τριών εισόδων **δεν** μπορεί να υλοποιηθεί χρησιμοποιώντας δύο πύλες NAND δύο εισόδων, γιατί ισχύει η αντιμεταθετική ιδιότητα, αλλά δεν ισχύει η προσεταιριστική ιδιότητα αφού:

$$\overline{A \cdot B \cdot C} \neq \overline{\overline{A \cdot B} \cdot C} = A \cdot B + \overline{C}$$

$$\overline{A \cdot B \cdot C} \neq \overline{\overline{A \cdot B} \cdot C} = \overline{A} + B \cdot C$$

όπως φαίνεται στον επόμενο Πίνακα.

Πίνακας: Στην πύλη NAND δεν ισχύει η προσεταιριστική ιδιότητα

A	B	C	$\overline{A \cdot B \cdot C}$	$\overline{\overline{A \cdot B} \cdot C}$	$\overline{\overline{A \cdot B} \cdot C}$
0	0	0	1	1	1
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	0	1
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	1	1

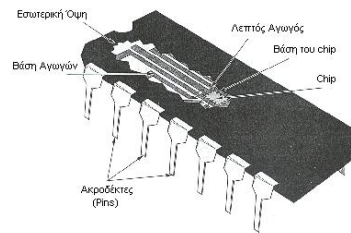
Με την ίδια λογική, μία πύλη NOR τριών εισόδων δεν μπορεί να υλοποιηθεί χρησιμοποιώντας δύο πύλες NOR δύο εισόδων.

Η λογική της επέκτασης του πλήθους των εισόδων των πυλών, μπορεί να εφαρμοστεί και για πύλες τεσσάρων εισόδων. Για παράδειγμα, μία πύλη AND τεσσάρων εισόδων μπορεί να υλοποιηθεί χρησιμοποιώντας τρεις πύλες AND δύο εισόδων και μία πύλη OR τεσσάρων εισόδων μπορεί να υλοποιηθεί χρησιμοποιώντας τρεις πύλες OR δύο εισόδων.

2.4 Ολοκληρωμένα κυκλώματα

2.4.1 Οικογένειες ολοκληρωμένων κυκλωμάτων

Τα ολοκληρωμένα κυκλώματα (integrated circuits) είναι συστατικά στοιχεία των ψηφιακών κυκλωμάτων. Ένα ολοκληρωμένο κύκλωμα είναι ένας ημιαγωγός κρύσταλλος από πυρίτιο (chip) που περιέχει ηλεκτρονικά στοιχεία με τα οποία κατασκευάζονται οι πύλες. Το chip τοποθετείται σε ένα πλαστικό περίβλημα και συγκολλούνται επαφές σε εξωτερικούς ακροδέκτες (pins) για να σχηματιστεί το ολοκληρωμένο κύκλωμα. Στην επόμενη Εικόνα φαίνεται η εσωτερική όψη ενός ολοκληρωμένου κυκλώματος σε συσκευασία ακροδεκτών διπλής σειράς (Dual In-line Package - DIP).



Εικόνα: Εσωτερική Όψη Ολοκληρωμένου Κυκλώματος

Τα ολοκληρωμένα κυκλώματα ανήκουν σε μία Κλίμακα Ολοκλήρωσης (Scale Integration) ανάλογα με το πλήθος των ισοδύναμων με πύλες κυκλωμάτων που περιέχουν. Έτσι, τα ολοκληρωμένα κυκλώματα ανήκουν σε μία από τις ακόλουθες κατηγορίες:

- ✓ **SSI** (Small Scale Integration) περιλαμβάνει λιγότερα από 12 ισοδύναμα με μία πύλη κυκλώματα
- ✓ **MSI** (Medium Scale Integration) περιλαμβάνει 12-100 ισοδύναμα με μία πύλη κυκλώματα
- ✓ **LSI** (Large Scale Integration) περιλαμβάνει 100-1000 ισοδύναμα με μία πύλη κυκλώματα
- ✓ **VLSI** (Very Large Scale Integration) περιλαμβάνει περισσότερα 1000-100000 ισοδύναμα με μία πύλη κυκλώματα
- ✓ **ULSI** (Ultra Large Scale Integration) περιλαμβάνει περισσότερα από 100000 ισοδύναμα με μία πύλη κυκλώματα

Οι λογικές πύλες ανήκουν σε μία οικογένεια (family) ολοκληρωμένων κυκλωμάτων, οι πιο γνωστές από τις οποίες είναι:

- BIPOLAR
- CMOS (Complementary Metal-Oxide Semiconductor)
- BICMOS (Bipolar CMOS)
- ECL (Emitter Coupled Logic)

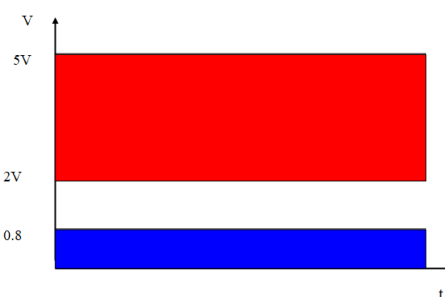
Τα χαρακτηριστικά των λογικών πυλών των ολοκληρωμένων κυκλωμάτων είναι τα ακόλουθα:

- ⇒ **Ικανότητα οδήγησης εξόδου** (Fan Out) είναι το πλήθος των εισόδων του ολοκληρωμένου κυκλώματος που μπορούν να οδηγηθούν από μία έξοδο του χωρίς να κινδυνεύσει η ομαλή λειτουργία.
- ⇒ **Απώλεια ισχύος** (Power Dissipation) είναι η ισχύς η οποία καταναλώνεται από τις πύλες κατά την λειτουργία τους με αποτέλεσμα την παραγωγή θερμότητας που διαχέεται στο περιβάλλον.
- ⇒ **Καθυστέρηση διάδοσης** (Propagation Delay) είναι ο χρόνος για να διαδοθεί η αλλαγή ενός σήματος από την είσοδο στην έξοδο.
- ⇒ **Περιθώριο θορύβου** (Noise Margin) είναι η ελάχιστη τάση εξωτερικού θορύβου που προκαλεί ανεπιθύμητη αλλαγή στην έξοδο.

2.4.2 Λογικές τιμές και περιοχές τάσης

Τα ολοκληρωμένα κυκλώματα αναγνωρίζουν στις εισόδους τους ηλεκτρικές τάσεις, στις οποίες αντιστοιχούν οι λογικές τιμές "0" ή "1". Επίσης, στις εξόδους τους δίνουν ηλεκτρικές τάσεις που αντιστοιχούν στις λογικές τιμές "0" ή "1".

Στην πράξη όμως δεν είναι δυνατόν να έχουμε απόλυτα ακριβείς τιμές τάσεων. Αυτό συμβαίνει για διάφορους λόγους, όπως διακυμάνσεις της τάσης τροφοδοσίας, επίδραση της θερμοκρασίας και των θορύβων στη λειτουργία των κυκλωμάτων και επίδραση του φορτίου στην τάση εξόδου. Για το λόγο αυτό ορίζονται δύο περιοχές τάσης, η μία που αντιστοιχεί στο λογικό "1" και η άλλη που αντιστοιχεί στο λογικό "0". Ανάμεσα τους υπάρχει μία περιοχή που τις ξεχωρίζει. Μία τιμή τάσης που βρίσκεται σε αυτή δεν μπορεί να θεωρηθεί από το κύκλωμα ούτε ως λογικό "0" ούτε ως λογικό "1" και έτσι η συμπεριφορά του ολοκληρωμένου είναι απρόβλεπτη. Οι περιοχές των τάσεων αναφέρονται στα φύλλα δεδομένων (Data Sheets) των κατασκευαστών. Για παράδειγμα, το ολοκληρωμένο κύκλωμα 7400 αναγνωρίζει στις **εισόδους** του ως λογικό "0" την περιοχή τάσεων από 0 Volts μέχρι 0.8 Volts και ως λογικό "1" την περιοχή τάσεων από 2 Volts μέχρι 5 Volts, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Αποδεκτές τιμές για τις τάσεις εισόδου

Οι αποδεκτές τιμές για τις τάσεις **εξόδου** του είναι από 0 Volts μέχρι 0.4 Volts για λογικό "0" και από 2.7 Volts μέχρι 5 Volts για λογικό "1", όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Αποδεκτές τιμές για τις τάσεις εξόδου

2.5 Συνδυαστικά κυκλώματα

Όπως γνωρίζουμε, ένα ψηφιακό κύκλωμα μπορεί να είναι είτε συνδυαστικό (combinational) είτε ακολουθιακό (sequential). Ένα συνδυαστικό κύκλωμα αποτελείται από εισόδους, λογικές πύλες που συνδέονται μεταξύ τους και εξόδους, όπως φαίνεται στην ακόλουθη Εικόνα.



Εικόνα: Συνδυαστικό Κύκλωμα

Σε ένα συνδυαστικό κύκλωμα με n εισόδους και m εξόδους, υπάρχει ένας συνδυασμός εξόδων για κάθε έναν από τους 2^n δυνατούς συνδυασμούς εισόδων. Κάθε χρονική στιγμή, κάθε μία από τις εξόδους εξαρτάται από τις τιμές των εισόδων την ίδια χρονική στιγμή.

Ένα συνδυαστικό κύκλωμα με n μεταβλητές εισόδου και m μεταβλητές εξόδου, έχει έναν πίνακα αληθείας. Ο πίνακας αυτός έχει στο αριστερό τμήμα n στήλες, όσες είναι και οι εισοδοί του κυκλώματος και στο δεξί τμήμα m στήλες, όσες είναι και οι έξοδοι του κυκλώματος. Το πλήθος των γραμμών του πίνακα είναι 2^n , όσοι είναι και οι δυνατοί συνδυασμοί των εισόδων. Οι συνδυασμοί παράγονται σύμφωνα με την ακολουθία δυαδικής αρίθμησης. Για κάθε συνδυασμό των τιμών των μεταβλητών εισόδου υπάρχει ένας συνδυασμός εξόδων, που εξαρτάται από τη λειτουργία του κυκλώματος. Για παράδειγμα, στον ακόλουθο Πίνακα παρουσιάζεται ο πίνακας αληθείας του συνδυαστικού κυκλώματος που εκτελεί την πρόσθεση δύο δυαδικών ψηφίων (bits). Το κύκλωμα έχει δύο εισόδους x (πρώτος προσθετέος) και y (δεύτερος προσθετέος) και δύο εξόδους S (άθροισμα-sum) και C (κρατούμενο-carry).

Είσοδοι		Έξοδοι	
x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Πίνακας: Πίνακας Αληθείας κυκλώματος με δύο εισόδους και δύο εξόδους

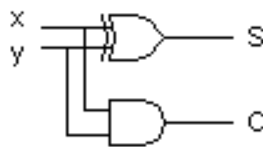
Σε ένα συνδυαστικό κύκλωμα, κάθε χρονική στιγμή, κάθε μία από τις εξόδους εξαρτάται από τις εισόδους την ίδια χρονική στιγμή και μπορεί να εκφραστεί ως λογική συνάρτηση των μεταβλητών εισόδου. Όταν μία μεταβλητή έχει τιμή «1» στον πίνακα αληθείας, τότε εμφανίζεται ως έχει στη συνάρτηση, ενώ όταν έχει τιμή «0», τότε εμφανίζεται με το συμπλήρωμά της. Κάθε μεταβλητή λαμβάνει την κανονική της μορφή. Οι συναρτήσεις των εξόδων του κυκλώματος προκύπτουν από τον πίνακα αληθείας του.

Για παράδειγμα, από τον ανωτέρω Πίνακα προκύπτει ότι η συνάρτηση εξόδου S έχει τιμή $S=1$ όταν $x=1$ και (AND) $y=0$ ($x \cdot y'$) ή (OR) όταν $x=0$ και (AND) $y=1$ ($x' \cdot y$), διαφορετικά έχει τιμή $S=0$. Επομένως, $S = x \cdot y' + x' \cdot y = x \oplus y$ (από τον πίνακα αληθείας της XOR).

Αντίστοιχα, η συνάρτηση εξόδου C έχει τιμή $C=1$ όταν $x=1$ και (AND) $y=1$ ($x \cdot y$), διαφορετικά έχει τιμή $C=0$. Επομένως, $C = x \cdot y$ (από τον πίνακα αληθείας της AND).

Λογικό κύκλωμα

Οι συναρτήσεις εξόδων του κυκλώματος μπορούν να υλοποιηθούν χρησιμοποιώντας λογικές πύλες. Για παράδειγμα, το κύκλωμα με τις συναρτήσεις εξόδου που προέκυψαν στην προηγούμενη παράγραφο μπορεί να υλοποιηθεί με μία πύλη XOR και μία πύλη AND, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Λογικό Κύκλωμα

2.6 Ελάχιστοι όροι και μέγιστοι όροι

Ελάχιστος όρος ή *ελαχιστόρος* (minterm) μίας συνάρτησης ονομάζεται ένα γινόμενο όλων των όρων της συνάρτησης, στο οποίο ο κάθε όρος (μεταβλητή) εμφανίζεται στην κανονική ή στη συμπληρωματική του μορφή.

Για παράδειγμα, ένας ελάχιστος όρος μιας συνάρτησης τριών μεταβλητών με εισόδους A, B, C είναι $A \cdot B' \cdot C$. Σε αυτό τον ελάχιστο όρο, οι μεταβλητές A, C εμφανίζονται με την κανονική τους μορφή, ενώ η μεταβλητή B εμφανίζεται στη συμπληρωματική της μορφή.

Αντίστοιχα, μέγιστος όρος ή *μεγιστόρος* (maxterm) μιας συνάρτησης ορίζεται ένα άθροισμα όλων των όρων της συνάρτησης, στο οποίο κάθε όρος εμφανίζεται στην κανονική ή τη συμπληρωματική της μορφή. Για παράδειγμα, ένας μέγιστος όρος μιας συνάρτησης τριών μεταβλητών είναι $A' + B' + C$.

Μία συνάρτηση n μεταβλητών έχει 2^n ελάχιστους όρους και 2^n μέγιστους όρους, οι οποίοι συμβολίζονται ως m_i και M_i αντίστοιχα ($0 \leq i < 2^n$). Κάθε συνάρτηση μπορεί να εκφραστεί ως άθροισμα ελάχιστων όρων ή ως γινόμενο μέγιστων όρων. Στον ακόλουθο Πίνακα παρουσιάζονται οι οκτώ ($=2^3$) ελάχιστοι όροι και οι 8 μέγιστοι όροι μίας συνάρτησης τριών μεταβλητών.

A	B	C	Ελάχιστοι όροι	Μέγιστοι όροι
0	0	0	$m_0 = A' \cdot B' \cdot C'$	$M_0 = A + B + C$
0	0	1	$m_1 = A' \cdot B' \cdot C$	$M_1 = A + B + C'$
0	1	0	$m_2 = A' \cdot B \cdot C'$	$M_2 = A + B' + C$
0	1	1	$m_3 = A' \cdot B \cdot C$	$M_3 = A + B' + C'$
1	0	0	$m_4 = A \cdot B' \cdot C'$	$M_4 = A' + B + C$
1	0	1	$m_5 = A \cdot B' \cdot C$	$M_5 = A' + B + C'$
1	1	0	$m_6 = A \cdot B \cdot C'$	$M_6 = A' + B' + C$
1	1	1	$m_7 = A \cdot B \cdot C$	$M_7 = A' + B' + C'$

Πίνακας: Ελάχιστοι όροι συνάρτησης τριών μεταβλητών

Παράδειγμα: Να γραφεί ως άθροισμα ελάχιστων όρων και ως γινόμενο μέγιστων όρων η συνάρτηση τριών μεταβλητών Y, ο πίνακας αληθείας της οποίας παρουσιάζεται στον ακόλουθο Πίνακα.

Απάντηση: Η συνάρτηση γράφεται ως άθροισμα ελάχιστων όρων ως εξής:

$$Y = m_1 + m_4 + m_7 = A' \cdot B' \cdot C + A \cdot B' \cdot C' + A \cdot B \cdot C.$$

Η συνάρτηση γράφεται επίσης ως γινόμενο μέγιστων όρων ως εξής:

$$Y = M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6 = (A + B + C) \cdot (A + B' + C) \cdot (A + B' + C') \cdot (A' + B + C') \cdot (A' + B' + C)$$

Είναι σαφές ότι η παράσταση με άθροισμα ελαχιστόρων προτιμάται όταν η συνάρτηση έχει πιο πολλά '0' στη στήλη του αποτελέσματος στον πίνακα αληθείας, ενώ η παράσταση με γινόμενο μεγιστόρων προτιμάται όταν η συνάρτηση έχει πιο πολλά '1' στη στήλη του αποτελέσματος του πίνακα αληθείας.

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Πίνακας: Πίνακας Αληθείας της συνάρτησης Y

Παράδειγμα: Να εκφραστεί ως άθροισμα ελαχιστόρων όρων η συνάρτηση τριών μεταβλητών $Y = A \cdot B + A' \cdot C$

Απάντηση: Η συνάρτηση δίνεται σε μορφή αθροίσματος γινομένων. Όμως σε κάθε γινόμενο δεν υπάρχουν όλοι οι όροι (μεταβλητές). Για τις μεταβλητές που λείπουν από κάθε γινόμενο του αθροίσματος, πολλαπλασιάζουμε το γινόμενο αυτό με το άθροισμα της μεταβλητής που λείπει και του συμπληρώματός της. Αν για παράδειγμα, λείπει η μεταβλητή A από ένα γινόμενο, τότε πολλαπλασιάζουμε το γινόμενο αυτό με το (A+A'). Έτσι, όλα τα γινόμενα μετατρέπονται σε ελάχιστους όρους, επομένως, η συνάρτηση εκφράζεται ως άθροισμα ελαχίστων όρων. Εφαρμόζοντας τη διαδικασία αυτή, έχουμε:

$$Y = A \cdot B + A' \cdot C = A \cdot B \cdot (C + C') + A' \cdot (B + B') \cdot C = A \cdot B \cdot C + A \cdot B \cdot C' + A' \cdot B \cdot C + A' \cdot B' \cdot C.$$

2.7 Σχεδίαση συνδυαστικών κυκλωμάτων

Το πρόβλημα της Σχεδίασης ενός συνδυαστικού κυκλώματος είναι η σχεδίαση του λογικού κυκλώματος, όταν δίνεται η περιγραφή της λειτουργίας του. Η σχεδίαση ενός Συνδυαστικού Κυκλώματος περιλαμβάνει τα ακόλουθα βήματα:

1. Κατασκευή του Πίνακα Αληθείας
2. Εύρεση των συναρτήσεων εξόδου
3. Σχεδίαση του λογικού κυκλώματος

Παράδειγμα: Να σχεδιαστεί ένα κύκλωμα που να αναγνωρίζει αν ένας 3-bits αριθμός είναι μικρότερος από 3, χρησιμοποιώντας πύλες NOT και πύλες AND και OR δύο εισόδων.

Απάντηση: Το κύκλωμα έχει τρεις εισόδους A, B και C, που αποτελούν τη δυαδική αναπαράσταση ενός δεκαδικού αριθμού από το 0 έως και το 7 και μία έξοδο Y. Η έξοδος του κυκλώματος είναι “1” όταν το δεκαδικό ισοδύναμο του 3-bits δυαδικού αριθμού των εισόδων του είναι μικρότερο από 3.

Βήμα 1. Κατασκευή του Πίνακα Αληθείας

Από την περιγραφή της λειτουργίας κατασκευάζεται ο Πίνακας Αληθείας που παρουσιάζεται στον επόμενο Πίνακα.

Δεκαδικός	A	B	C	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

Πίνακας: Πίνακας Αληθείας του Συνδυαστικού Κυκλώματος

Βήμα 2. Εύρεση των συναρτήσεων εξόδου του κυκλώματος

Από τον Πίνακα Αληθείας του κυκλώματος προκύπτει ότι η συνάρτηση εξόδου του είναι $Y=1$ όταν

$$A=0 \text{ και (AND) } B=0 \text{ και (AND) } C=0 \text{ ή (OR) } A=0 \text{ και (AND) } B=0 \text{ και (AND) } C=1 \text{ ή} \\ \text{(OR) } A=0 \text{ και (AND) } B=1 \text{ και (AND) } C=0$$

Επομένως, η συνάρτηση εξόδου βρίσκεται ως συνάρτηση των εισόδων του:
 $Y=A' \cdot B' \cdot C' + A' \cdot B' \cdot C + A' \cdot B \cdot C'$

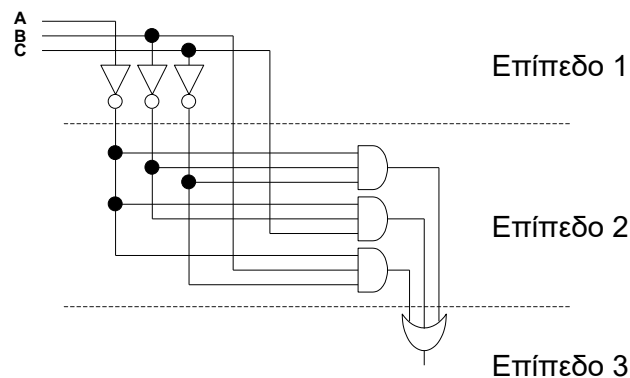
Βήμα 3. Σχεδίαση του λογικού κυκλώματος

Για τη σχεδίαση του λογικού κυκλώματος, ξεκινώντας από την έξοδο προς τις εισόδους του κυκλώματος, σχεδιάζονται οι πύλες του κυκλώματος λαμβάνοντας υπόψη τις λογικές πράξεις των συναρτήσεων εξόδου. Το λογικό κύκλωμα χωρίζεται σε επίπεδα που περιέχουν τις πύλες, με βάση την προτεραιότητα των πράξεων. Ξεκινώντας από την είσοδο του ΣΚ προς τις εισόδους του ΣΚ, το κύκλωμα χωρίζεται σε τρία επίπεδα πυλών:

Επίπεδο 1: Τρεις πύλες NOT που χρησιμοποιούνται για την παραγωγή των A' , B' , C' .

Επίπεδο 2: Τρεις πύλες AND τριών εισόδων που χρησιμοποιούνται για τον υπολογισμό των $A' \cdot B' \cdot C'$, $A' \cdot B' \cdot C$, $A' \cdot B \cdot C'$.

Επίπεδο 3: Μία πύλη OR τριών εισόδων, που χρησιμοποιείται για τον υπολογισμό της συνάρτησης όπως φαίνεται στην επόμενη Εικόνα.



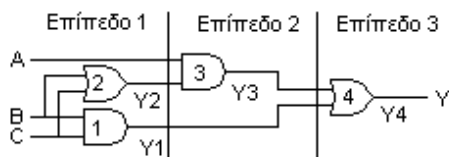
Εικόνα: Συνδυαστικό Κύκλωμα

2.8 Ανάλυση συνδυαστικών κυκλωμάτων

Κατά την ανάλυση ενός συνδυαστικού κυκλώματος επιδιώκεται η περιγραφή της λειτουργίας του κυκλώματος, όταν μας δίνεται το λογικό κύκλωμα. Η ανάλυση ενός συνδυαστικού κυκλώματος περιλαμβάνει τα ακόλουθα βήματα:

1. Εύρεση των συναρτήσεων εξόδου του κυκλώματος
2. Κατασκευή του Πίνακα Αληθείας του κυκλώματος

Παράδειγμα: Να προσδιοριστεί η λειτουργία του κυκλώματος της επόμενης Εικόνας.



Εικόνα: Συνδυαστικό Κύκλωμα

Βήμα 1. Εύρεση των συναρτήσεων εξόδου

Το κύκλωμα έχει τρεις εισόδους A, B και C και μία έξοδο Y. Αποτελείται από δύο πύλες AND δύο εισόδων (πύλες 1 και 3) και δύο πύλες OR (πύλες 2 και 4) δύο εισόδων. Για την εύρεση των συναρτήσεων εξόδου του κυκλώματος, ξεκινώντας από τις εισόδους προς τις εξόδους του κυκλώματος, το κύκλωμα χωρίζεται σε επίπεδα πυλών και καταγράφονται οι συναρτήσεις εξόδων των πυλών. Οι συναρτήσεις εξόδων των πυλών του πρώτου επιπέδου είναι συναρτήσεις των εισόδων του κυκλώματος. Οι συναρτήσεις εξόδων των πυλών των επόμενων επιπέδων είναι συναρτήσεις των εξόδων των πυλών των προηγούμενων επιπέδων. Οι συναρτήσεις εξόδων των πυλών του τελευταίου επιπέδου, που είναι οι συναρτήσεις εξόδων του κυκλώματος, θα είναι προφανώς συναρτήσεις των εισόδων του κυκλώματος.

Ξεκινώντας από τις εισόδους του κυκλώματος προς την έξοδό του, μπορούμε να χωρίσουμε το κύκλωμα σε τρία επίπεδα πυλών ως εξής.

Επίπεδο 1: Πύλες 1 και 2. Η έξοδος της πύλης 1 είναι: $Y1=B \cdot C$, και η έξοδος της πύλης 2 είναι: $Y2=B+C$

Επίπεδο 2: Πύλη 3. Η έξοδος της είναι: $Y3=A \cdot Y2=A \cdot (B+C)$

Επίπεδο 3: Πύλη 4. Η έξοδος της είναι: $Y4=Y1+Y3=B \cdot C+A \cdot (B+C)$

Η συνάρτηση εξόδου της πύλης 4 του τελευταίου επιπέδου είναι η συνάρτηση εξόδου του κυκλώματος: $Y=Y4$

Επομένως, η συνάρτηση εξόδου του κυκλώματος βρίσκεται ως συνάρτηση των εισόδων του κυκλώματος: $Y=Y4=A \cdot (B+C)+B \cdot C$

Βήμα 2. Κατασκευή του πίνακα αληθείας

Από τη συνάρτηση εξόδου κατασκευάζεται ο πίνακας αληθείας του κυκλώματος, που παρουσιάζεται στον επόμενο Πίνακα.

A	B	C	$Y1=B \cdot C$	$Y2=B+C$	$Y3=A \cdot (B+C)$	$Y=Y4=A \cdot (B+C)+B \cdot C$
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	1	0	0
0	1	1	1	1	0	1
1	0	0	0	0	0	0
1	0	1	0	1	1	1
1	1	0	0	1	1	1
1	1	1	1	1	1	1

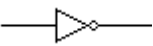


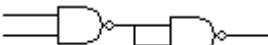

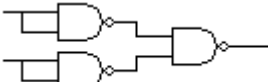

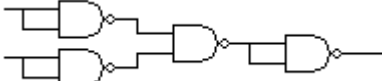

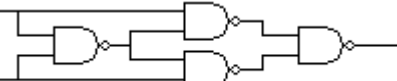
Πίνακας: Πίνακας Αληθείας του Συνδυαστικού Κυκλώματος

2.9 Οικουμενικές πύλες

Κάθε ψηφιακό κύκλωμα μπορεί να υλοποιηθεί με πύλες NOT, AND και OR, οι οποίες με τη σειρά τους μπορούν να υλοποιηθούν μόνο με πύλες NAND ή NOR. Επομένως, κάθε ψηφιακό κύκλωμα μπορεί να υλοποιηθεί μόνο με πύλες NAND ή NOR που ονομάζονται *οικουμενικές πύλες* (universal gates). Η υλοποίηση ψηφιακών κυκλωμάτων με οικουμενικές πύλες οδηγεί σε οικονομικές υλοποιήσεις.

2.9.1 Οικουμενικές πύλες NAND

Στη συνέχεια θα ασχοληθούμε με τις οικουμενικές πύλες NAND. Η υλοποίηση των πυλών NOT, AND και OR μόνο με πύλες NAND φαίνεται στην επόμενη Εικόνα.

NOT		
AND		
OR		
NOR		
XOR		

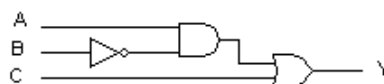
Εικόνα: Υλοποίηση των πυλών NOT, AND, OR, NOR και XOR με πύλες NAND

Η διαδικασία σχεδίασης συνδυαστικού κυκλώματος μόνο με πύλες NAND είναι η ακόλουθη:

- ✓ σχεδίαση του κυκλώματος με πύλες NOT, AND και OR
- ✓ αντικατάσταση των πυλών NOT, AND και OR από πύλες NAND
- ✓ διαγραφή (όπου αυτό είναι δυνατό) δύο συνεχόμενων πυλών NAND με βραχυκυκλωμένες εισόδους (οι οποίες λειτουργούν ως πύλες NOT)

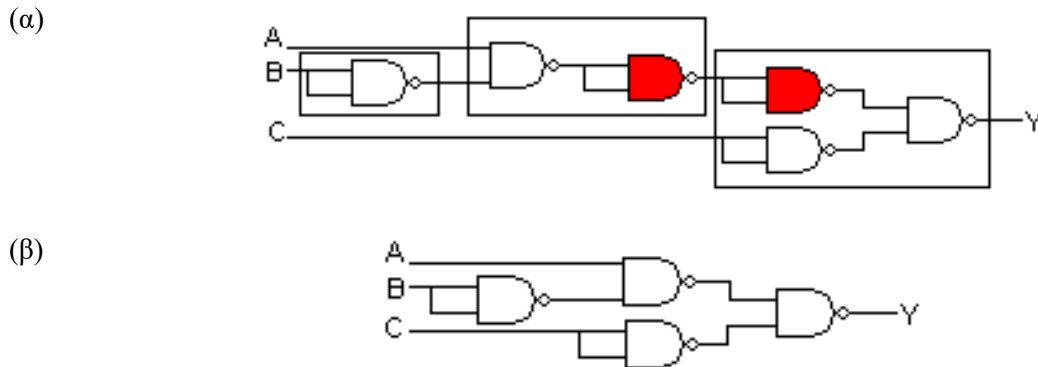
Παράδειγμα: Να σχεδιαστεί συνδυαστικό κύκλωμα έχει τρεις εισόδους A, B και C και μία έξοδο $Y=A \cdot B' + C$, μόνο με πύλες NAND.

Απάντηση: Η υλοποίηση του κυκλώματος με πύλες NOT, AND και OR απαιτεί μία πύλη NOT, μία πύλη AND και μία πύλη OR, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Υλοποίηση της συνάρτησης $Y=A \cdot B' + C$ με πύλες NOT, AND και OR

Η αντικατάσταση των πυλών NOT, AND και OR από πύλες NAND οδηγεί στην υλοποίηση του κυκλώματος με έξι πύλες NAND, όπως φαίνεται στην επόμενη Εικόνα (α). Η διαγραφή δύο συνεχόμενων πυλών NAND (απαλοιφή NOT-NOT) οδηγεί στην υλοποίηση του κυκλώματος με τέσσερις πύλες NAND 2 εισόδων, όπως φαίνεται στο (β).



Εικόνα: Αντικατάσταση των πυλών NOT, AND και OR από πύλες NAND
 (α) Υλοποίηση της $Y=A \cdot B'+C$ με πύλες NAND (β) απλοποίηση δύο συνεχόμενων NAND

2.9.2 Οικουμενικές πύλες NOR

Στη συνέχεια θα ασχοληθούμε με τις οικουμενικές πύλες NOR. Η υλοποίηση των πυλών NOT, AND και OR μόνο με πύλες NOR φαίνεται στην επόμενη Εικόνα.

NOT			$A' = (A+A)'$
AND			$(A \cdot B) = ((A \cdot B)')' = (A' + B')'$
OR			$(A+B) = ((A+B)')'$
NAND			
XOR			

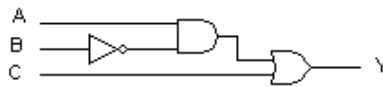
Εικόνα: Υλοποίηση των πυλών NOT, AND και OR με πύλες NOR

Η διαδικασία σχεδίασης συνδυαστικού κυκλώματος μόνο με πύλες NOR είναι η ακόλουθη:

- σχεδίαση του κυκλώματος με πύλες NOT, AND και OR
- αντικατάσταση των πυλών NOT, AND και OR από πύλες NOR
- διαγραφή δύο συνεχόμενων πυλών NOR με βραχυκυκλωμένες εισόδους, που λειτουργούν ως πύλες NOT

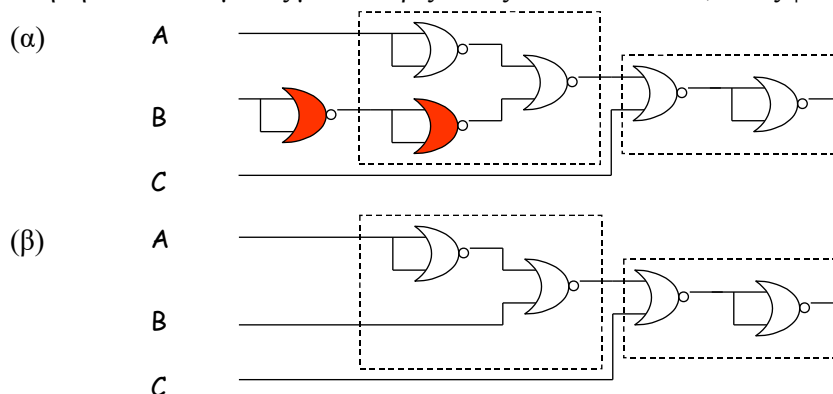
Παράδειγμα: Να υλοποιηθεί το συνδυαστικό κύκλωμα έχει τρεις εισόδους A, B και C και μία έξοδο $Y=A \cdot B' + C$ μόνο με πύλες NOR 2 εισόδων.

Απάντηση: Η υλοποίηση του κυκλώματος με πύλες NOT, AND και OR φαίνεται στην επόμενη Εικόνα.



Εικόνα: Υλοποίηση της $Y=A \cdot B' + C$ με πύλες NOT, AND και OR

Η αντικατάσταση των πυλών NOT, AND και OR από πύλες NOR οδηγεί στην υλοποίηση του κυκλώματος με έξι πύλες NOR. Η διαγραφή δύο συνεχόμενων πυλών NOR (απαλοιφή NOT-NOT) οδηγεί στην υλοποίηση του κυκλώματος με τέσσερις πύλες NOR 2 εισόδων, όπως φαίνεται στο (β).



Εικόνα: (α) Υλοποίηση της $Y=A \cdot B' + C$ μόνο με πύλες NOR

(β) Διαγραφή δύο συνεχόμενων πυλών NOR με βραχυκυκλωμένες εισόδους

2.10 Απλοποίηση λογικών συναρτήσεων

Η απλοποίηση των συναρτήσεων εξόδου ενός κυκλώματος οδηγεί σε απλούστερο και οικονομικότερο (από πλευράς υλοποίησης) κύκλωμα. Στην παράγραφο αυτή θα παρουσιαστούν δύο μέθοδοι απλοποίησης λογικών συναρτήσεων, με χρήση της Άλγεβρας Boole και με χρήση των χαρτών Karnaugh. Η απλοποίηση λογικών συναρτήσεων με χρήση της Άλγεβρας Boole, βασίζεται στη χρήση των Αξιωμάτων και των Θεωρημάτων της Άλγεβρας Boole, που έχουν αναφερθεί σε προηγούμενη παράγραφο.

Παράδειγμα: Να απλοποιηθεί η λογική συνάρτηση $Y=A' \cdot B \cdot C + A' \cdot BC'$

Απάντηση: Ακολουθούμε την εξής πορεία.

$$Y = A' \cdot B \cdot C + A' \cdot B \cdot C' = A' \cdot B \cdot (C + C') = A' \cdot B$$

Παράδειγμα: Να απλοποιηθεί η λογική συνάρτηση $Y = (A+B) \cdot (A+B')$

Απάντηση: Έχουμε:

$$Y = (A+B) \cdot (A+B') = A \cdot A + A \cdot B' + B \cdot A + B \cdot B' = A + A \cdot B' + A \cdot B + 0 = (A + A \cdot B) + A \cdot B' = A + A \cdot B' = A$$

(Θεώρημα Απορρόφησης)

Παράδειγμα: Να απλοποιηθεί η λογική συνάρτηση $Y = (A' \cdot B' + C)'$

Απάντηση: Έχουμε:

$$Y = (A' \cdot B' + C)' = (A' \cdot B')' \cdot (C)' = (A+B) \cdot C \text{ (Θεώρημα De Morgan)}$$

Η απλοποίηση με τη χρήση των αξιωμάτων της άλγεβρας Boole παρουσιάζει δυσκολίες, εφόσον δε μας προμηθεύει με μια συγκεκριμένη μεθοδολογία απλοποίησης των λογικών συναρτήσεων. Για το λόγο αυτό στην πράξη χρησιμοποιούμε μια μεθοδολογία με τη χρήση των χαρτών Karnaugh. Η μέθοδος απλοποίησης λογικών συναρτήσεων με χρήση των χαρτών Karnaugh είναι μία γραφική μέθοδος που βασίζεται σε μία διαφορετική αναπαράσταση των πινάκων αληθείας των λογικών συναρτήσεων και χρησιμοποιείται για απλοποίηση λογικών συναρτήσεων δύο, τριών, τεσσάρων έως και πέντε μεταβλητών. Στην πραγματικότητα, για συναρτήσεις με περισσότερες από πέντε μεταβλητές εισόδου, η τεχνική αυτή δε μπορεί να χρησιμοποιηθεί στην πράξη.

Αναπαράσταση λογικών συναρτήσεων με χάρτες Karnaugh

Ένας χάρτης Karnaugh αποτελείται από τετράγωνα, κάθε ένα από τα οποία αντιστοιχεί σε ένα ελάχιστο όρο της λογικής συνάρτησης που αναπαριστά. Οι χάρτες Karnaugh δύο, τριών και τεσσάρων μεταβλητών παρουσιάζονται στην επόμενη Εικόνα.

	B'	B
A'	m0	m1
A	m2	m3

(α)

	B'·C'	B'·C	B·C	B·C'
A'	m0	m1	m3	m2
A	m4	m5	m7	m6

(β)

	C'·D'	C'·D	C·D	C·D'
A'·B'	m0	m1	m3	m2
A'·B	m4	m5	m7	m6
A·B	m12	m13	m15	m14
A·B'	m8	m9	m11	m10

(γ)

Εικόνα: Χάρτης Karnaugh δύο, τριών, τεσσάρων μεταβλητών

Ο συμβολισμός των μεταβλητών είναι είτε όπως φαίνεται στην προηγούμενη Εικόνα είτε εναλλακτικά όπως φαίνεται στην επόμενη Εικόνα.

	B	
A	0	1
0	m0	m1
1	m2	m3

(α)

	B C			
A	00	01	11	10
0	m0	m1	m3	m2
1	m4	m5	m7	m6

(β)

	C D			
A B	00	01	11	10
00	m0	m1	m3	m2
01	m4	m5	m7	m6
11	m12	m13	m15	m14
10	m8	m9	m11	m10

(γ)

Εικόνα: Χάρτης Karnaugh δύο, τριών, τεσσάρων μεταβλητών

Για να αναπαραστήσουμε μία λογική συνάρτηση με χάρτη Karnaugh θέτουμε “1” σε κάθε τετράγωνο του χάρτη Karnaugh που αντιστοιχεί σε ελάχιστο όρο, όπου η συνάρτηση έχει τιμή “1” και “0” (ή τίποτα) σε κάθε τετράγωνο του χάρτη Karnaugh που αντιστοιχεί σε ελάχιστο όρο, όπου η συνάρτηση έχει τιμή “0”.

Παράδειγμα: Να αναπαρασταθεί με χάρτη Karnaugh η λογική συνάρτηση δύο μεταβλητών $Y(A,B)=A \cdot B' + A' \cdot B$.

Απάντηση: Η συνάρτηση γράφεται σε μορφή αθροίσματος ελάχιστων όρων, επομένως μπορεί να αναπαρασταθεί με το χάρτη Karnaugh της επόμενης Εικόνας.

	B'	B
A'		1
A	1	

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y(A,B)=A \cdot B' + A' \cdot B$

Παράδειγμα: Να αναπαρασταθεί χάρτη Karnaugh η λογική συνάρτηση τριών μεταβλητών $Y(A,B,C)=A' \cdot B \cdot C' + A' \cdot B \cdot C + A \cdot B' \cdot C' + A \cdot B' \cdot C$.

Απάντηση: Η συνάρτηση είναι σε μορφή αθροίσματος ελάχιστων όρων, επομένως μπορεί να αναπαρασταθεί με τον χάρτη Karnaugh της επόμενης Εικόνας.

	B'.C	B'.C	B.C	B.C'
A'			1	1
A	1	1		

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y(A,B,C)=A' \cdot B \cdot C' + A' \cdot B \cdot C + A \cdot B' \cdot C' + A \cdot B' \cdot C$

Παράδειγμα: Να αναπαρασταθεί με χάρτη Karnaugh η λογική συνάρτηση τριών μεταβλητών $Y(A,B,C)=A' \cdot B \cdot C + A \cdot C$.

Απάντηση: Η συνάρτηση γράφεται σε μορφή αθροίσματος ελάχιστων όρων: $Y(A,B,C)=A' \cdot B \cdot C + A \cdot C = A' \cdot B \cdot C + A \cdot (B+B') \cdot C = A' \cdot B \cdot C + A \cdot B \cdot C + A \cdot B' \cdot C$. Επομένως μπορεί να αναπαρασταθεί με το χάρτη Karnaugh της επόμενης Εικόνας.

	B'.C	B'.C	B.C	B.C'
A'			1	
A		1	1	

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y(A,B,C)=A' \cdot B \cdot C + A \cdot C$

Αξίζει να παρατηρήσει κανείς τα ακόλουθα:

- ο όρος $A' \cdot B \cdot C$ της συνάρτησης αντιστοιχεί στο **κόκκινο** τετράγωνο του χάρτη Karnaugh, που είναι η τομή των περιοχών $A=0$, $B=1$ και $C=1$
- ο όρος $A \cdot C$ συνάρτησης αντιστοιχεί στα **δύο μπλε** τετράγωνα του χάρτη Karnaugh, που είναι η τομή των περιοχών $A=1$ και $C=1$

Παράδειγμα: Να αναπαρασταθεί με χάρτη Karnaugh η λογική συνάρτηση τριών μεταβλητών $Y(A,B,C,D)=A \cdot B \cdot C \cdot D + A' \cdot B \cdot C + C' \cdot D'$.

Απάντηση: Η συνάρτηση γράφεται σε μορφή αθροίσματος ελάχιστων όρων:

$$Y(A,B,C,D) = A \cdot B \cdot C \cdot D + A' \cdot B \cdot C + C' \cdot D' = A \cdot B \cdot C \cdot D + A' \cdot B \cdot C \cdot (D + D') + (A + A') \cdot (B + B') \cdot C' \cdot D' = A' \cdot B' \cdot C' \cdot D' + A' \cdot B \cdot C' \cdot D' + A' \cdot B \cdot C \cdot D' + A' \cdot B \cdot C \cdot D + A \cdot B' \cdot C' \cdot D' + A \cdot B \cdot C' \cdot D' + A \cdot B \cdot C \cdot D$$

Επομένως, η συνάρτηση μπορεί να αναπαρασταθεί με τον χάρτη Karnaugh της επόμενης Εικόνας.

	C'.D'	C'.D	C.D	C.D'
A'.B'	1			
A'.B	1		1	1
A.B	1		1	
A.B'	1			

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y(A,B,C,D) = A \cdot B \cdot C \cdot D + A' \cdot B \cdot C + C' \cdot D'$

Αξίζει να παρατηρήσει κανείς τα ακόλουθα:

- ο όρος $A \cdot B \cdot C \cdot D$ της συνάρτησης αντιστοιχεί στο **μπλε** τετράγωνο του χάρτη Karnaugh, που είναι η τομή των περιοχών $A=1, B=1, C=1$ και $D=1$
- ο όρος $A' \cdot B \cdot C$ της συνάρτησης αντιστοιχεί στα δύο **κόκκινα** τετράγωνα του χάρτη Karnaugh, που είναι η τομή των περιοχών $A=0, B=1$ και $C=1$
- ο όρος $C' \cdot D'$ της συνάρτησης αντιστοιχεί στα τέσσερα **πράσινα** τετράγωνα του χάρτη Karnaugh, που είναι η τομή των περιοχών $C=0$ και $D=0$

Με βάση τις παρατηρήσεις αυτές θα μπορούσαμε να είχαμε αποφύγει τις πράξεις και να είχαμε τοποθετήσει τα "1" της συνάρτησης κατευθείαν στο χάρτη Karnaugh.

Απλοποίηση λογικών συναρτήσεων με χάρτες Karnaugh

Κεντρικό ρόλο στην απλοποίηση συναρτήσεων με χάρτη Karnaugh παίζει ή ιδέα των γειτονικών τετραγώνων. *Γειτονικά τετράγωνα* σε ένα χάρτη Karnaugh ονομάζονται τα τετράγωνα που είναι σε συνεχόμενες οριζόντιες ή κάθετες (αλλά όχι διαγώνιες) θέσεις.

Το πλήθος των γειτονικών τετραγώνων που μπορούμε να χρησιμοποιήσουμε για να εξαλείψουμε όρους από την παράσταση μιας συνάρτησης πρέπει να είναι δύναμη του 2, δηλαδή 2, 4, 8, 16 κ.λ.π..

Έτσι, στο χάρτη Karnaugh τα 2 τετράγωνα που περιέχουν τους ελάχιστους όρους m_0 και m_1 είναι γειτονικά. Επίσης, γειτονικά είναι τα 2 τετράγωνα που περιέχουν τους ελάχιστους όρους m_{11} και m_{15} , τα 4 τετράγωνα που περιέχουν τους ελάχιστους όρους m_4, m_5, m_6 και m_7 , καθώς και τα 8 τετράγωνα που περιέχουν τους ελάχιστους όρους $m_2, m_3, m_6, m_7, m_{10}, m_{11}, m_{14}$ και m_{15} . Δεν είναι γειτονικά τα 2 τετράγωνα που περιέχουν τους ελάχιστους όρους m_5 και m_{15} .

Ένα σημαντικό χαρακτηριστικό των χαρτών Karnaugh είναι ότι είναι *αναδιπλούμενοι*. Η αναδίπλωση μπορεί να γίνει γύρω από την περίμετρο (τις εξωτερικές γραμμές) του χάρτη Karnaugh.

Έτσι, στο χάρτη Karnaugh τα 2 τετράγωνα που περιέχουν τους ελάχιστους όρους m_0 και m_8 είναι γειτονικά. Επίσης, γειτονικά είναι τα 2 τετράγωνα που περιέχουν τους ελάχιστους όρους m_4 και m_6 , τα 4 τετράγωνα που περιέχουν τους ελάχιστους όρους m_2, m_3, m_{10} και m_{11} , καθώς και τα 4 τετράγωνα που περιέχουν τους ελάχιστους όρους m_4, m_6, m_{12} και m_{14} .

Δύο γειτονικά τετράγωνα σε ένα χάρτη Karnaugh αντιστοιχούν σε ελάχιστους όρους που διαφέρουν κατά μία μόνο μεταβλητή (η οποία εμφανίζεται με την πραγματική τιμή της στον έναν ελάχιστο όρο και με τη συμπληρωματική τιμή της στον άλλον ελάχιστο όρο). Αυτή η μεταβλητή μπορεί να απομακρυνθεί αν και στα δύο γειτονικά τετράγωνα έχει τεθεί “1”. Έτσι, αν ομαδοποιήσουμε 2 γειτονικά τετράγωνα στα οποία έχει τεθεί “1” τότε απομακρύνουμε μία μεταβλητή.

Με την ίδια λογική, αν ομαδοποιήσουμε 4 γειτονικά τετράγωνα στα οποία έχει τεθεί “1”, τότε απομακρύνουμε 2 μεταβλητές, αν ομαδοποιήσουμε 8 γειτονικά τετράγωνα στα οποία έχει τεθεί “1”, τότε απομακρύνουμε 3 μεταβλητές.

Αξίζει ακόμη να σημειωθεί ότι, σύμφωνα με τα αξιώματα της άλγεβρας Boole, επιτρέπεται να συμπεριληφθεί στην παράσταση μιας συνάρτησης ο ίδιος όρος περισσότερες από μια φορές. Επομένως, ένα τετράγωνο επιτρέπεται να συμπεριληφθεί σε πολλές ομάδες γειτονικών τετραγώνων.

Με βάση τα παραπάνω, αν ομαδοποιήσουμε γειτονικά τετράγωνα στα οποία έχει τεθεί “1” (), τότε απομακρύνουμε μεταβλητές. Επομένως, επιτυγχάνεται η απλοποίηση της λογικής συνάρτησης που αναπαρίσταται με το χάρτη Karnaugh.

Η απλοποίηση λογικών συναρτήσεων με χάρτες Karnaugh περιλαμβάνει τα ακόλουθα βήματα:

- Γράφουμε τη συνάρτηση με μορφή αθροίσματος γινομένων και τοποθετούμε τα “1” της συνάρτησης στο χάρτη Karnaugh.
- Δημιουργούμε ομάδες με “1” (δηλαδή όρους της συνάρτησης) των 2, 4, 8 μελών από γειτονικά τετράγωνα (οριζόντια, κάθετα και αναδιπλούμενα, αλλά όχι διαγώνια). Προσπαθούμε να δημιουργούμε όσο το δυνατόν μεγαλύτερες ομάδες. Κάθε “1” μπορεί να συμμετάσχει σε περισσότερες από μία ομάδες. Όταν όλα τα “1”, που μπορούν να ομαδοποιηθούν, έχουν συμπεριληφθεί σε κάποια ομάδα, τότε δε δημιουργούμε πλέον νέες ομάδες.
- Ξαναγράφουμε τη συνάρτηση με όρους που αντιστοιχούν στις ομάδες (παραλείποντας τις μεταβλητές που μέσα στην ομάδα αλλάζουν τιμή) και τους όρους που δεν έχουν ομαδοποιηθεί.

Παράδειγμα: Να απλοποιηθεί η λογική συνάρτηση δύο μεταβλητών $Y(A,B)=A \cdot B' + A \cdot B$

Απάντηση: Η συνάρτηση είναι σε μορφή αθροίσματος ελαχίστων όρων. Τοποθετούμε τα “1” της συνάρτησης στον χάρτη Karnaugh, όπως φαίνεται στην επόμενη Εικόνα.

	B'	B
A'		
A	1	1

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y=A \cdot B' + A \cdot B$

Οι ελάχιστοι όροι $A \cdot B'$ και $A \cdot B$ δικαιολογούν την απομάκρυνση της μεταβλητής B. Πράγματι, τα δύο “1” της συνάρτησης ομαδοποιούνται σε μία δυάδα: Τα δύο κόκκινα γειτονικά τετράγωνα του χάρτη Karnaugh, αντιστοιχούν στον όρο A. Η ομαδοποίηση των δύο τετραγώνων έχει ως αποτέλεσμα την απομάκρυνση μίας μεταβλητής (της μεταβλητής B) και την απλοποίηση της συνάρτησης ως εξής: $Y=A$

Παράδειγμα: Να απλοποιηθεί η λογική συνάρτηση τριών μεταβλητών

$$Y(A,B,C)=A' \cdot B \cdot C' + A' \cdot B \cdot C + A \cdot B' \cdot C' + A \cdot B' \cdot C$$

Απάντηση: Τοποθετούμε τους “1” της συνάρτησης στο χάρτη Karnaugh, όπως φαίνεται στην επόμενη Εικόνα.

	B'.C'	B'.C	B.C	B.C'
A'			1	1
A	1	1		

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y(A,B,C) = A'.B.C' + A'.B.C + A.B'.C' + A.B'.C$

Οι “1” της συνάρτησης ομαδοποιούνται σε δύο ομάδες:

- τα δύο κόκκινα γειτονικά τετράγωνα του χάρτη Karnaugh αντιστοιχούν στον όρο $A \cdot B'$
- τα δύο μπλε γειτονικά τετράγωνα του χάρτη Karnaugh αντιστοιχούν στον όρο $A' \cdot B$

Η ομαδοποίηση έχει ως αποτέλεσμα την απλοποίηση της συνάρτησης ως εξής: $Y = A \cdot B' + A' \cdot B$

Παράδειγμα. Να απλοποιηθεί η λογική συνάρτηση τριών μεταβλητών

$$Y(A,B,C) = A'.B.C + A.B'.C' + A.B.C' + A.B.C$$

Απάντηση: Η συνάρτηση μπορεί να απλοποιηθεί χρησιμοποιώντας το χάρτη Karnaugh της επόμενης Εικόνας.

	B'.C'	B'.C	B.C	B.C'
A'			1	
A	1		1	1

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y(A,B,C) = A'.B.C + A.B'.C' + A.B.C' + A.B.C$

Οι “1” της συνάρτησης ομαδοποιούνται ως εξής:

- τα δύο μπλε γειτονικά τετράγωνα του χάρτη Karnaugh αντιστοιχούν στον όρο $B \cdot C$
- τα δύο κόκκινα γειτονικά τετράγωνα του χάρτη Karnaugh, αντιστοιχούν στον όρο $A \cdot C'$

Επειδή όλα τα “1”, που μπορούν να ομαδοποιηθούν, έχουν συμπεριληφθεί σε κάποια ομάδα, δεν δημιουργούμε νέες ομάδες (οι “1” της συνάρτησης στα δύο γειτονικά τετράγωνα του χάρτη Karnaugh, που αντιστοιχούν στους όρους $A \cdot B \cdot C$ και $A \cdot B \cdot C'$ δεν ομαδοποιούνται).

Η ομαδοποίηση έχει ως αποτέλεσμα την απλοποίηση της συνάρτησης ως εξής: $Y = A \cdot C' + B \cdot C$

Παράδειγμα. Να απλοποιηθεί η λογική συνάρτηση τριών μεταβλητών $Y(A,B,C) = A \cdot B' \cdot C + B \cdot C' + B' \cdot C'$

Απάντηση: Η συνάρτηση γράφεται σε μορφή αθροίσματος ελάχιστων όρων ως εξής:

$$Y(A,B,C) = A \cdot B' \cdot C + B \cdot C' + B' \cdot C' = A \cdot B' \cdot C' + A \cdot B \cdot C' + A \cdot B' \cdot C' + A \cdot B' \cdot C + A \cdot B \cdot C'$$

Η συνάρτηση μπορεί να απλοποιηθεί χρησιμοποιώντας το χάρτη Karnaugh της επόμενης Εικόνας.

	B'.C'	B'.C	B.C	B.C'
A'	1			1
A	1	1		1

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y(A,B,C) = A \cdot B' \cdot C + B \cdot C' + B' \cdot C'$

Οι “1” της συνάρτησης ομαδοποιούνται ως εξής:

- τα δύο γειτονικά τετράγωνα του χάρτη Karnaugh, που βρίσκονται μέσα σε **κόκκινο** περίγραμμα, αντιστοιχούν στον όρο $A \cdot B'$
- τα τέσσερα μπλε γειτονικά τετράγωνα του χάρτη (αναδίπλωση), αντιστοιχούν στον όρο C' .

Η ομαδοποίηση έχει ως αποτέλεσμα την απλοποίηση της συνάρτησης ως εξής: $Y = A \cdot B' + C'$

Παράδειγμα. Να απλοποιηθεί η λογική συνάρτηση τριών μεταβλητών $Y(A,B,C) = A' \cdot C + A' \cdot B + A \cdot B' \cdot C + B \cdot C$

Απάντηση: Η συνάρτηση μπορεί να απλοποιηθεί χρησιμοποιώντας το χάρτη Karnaugh της επόμενης Εικόνας.

	$B' \cdot C'$	$B' \cdot C$	$B \cdot C$	$B \cdot C'$
A'		1	1	1
A		1	1	

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y(A,B,C) = A' \cdot C + A' \cdot B + A \cdot B' \cdot C + B \cdot C$

Οι “1” της συνάρτησης ομαδοποιούνται ως εξής:

- τα δύο γειτονικά τετράγωνα του χάρτη Karnaugh, που βρίσκονται στην πρώτη γραμμή μέσα σε περίγραμμα, αντιστοιχούν στον όρο $A' \cdot B$
- τα τέσσερα γειτονικά τετράγωνα των μεσαίων στηλών του χάρτη αντιστοιχούν στον όρο C

Η ομαδοποίηση έχει ως αποτέλεσμα την απλοποίηση της συνάρτησης ως εξής: $Y = A' \cdot B + C$

Παράδειγμα. Να απλοποιηθεί η λογική συνάρτηση τεσσάρων μεταβλητών

$$Y(A,B,C,D) = A' \cdot B' \cdot C' \cdot D' + A' \cdot B' \cdot C' \cdot D + A' \cdot B' \cdot C \cdot D' + A' \cdot B' \cdot C \cdot D + A' \cdot B \cdot C' \cdot D' + A' \cdot B \cdot C' \cdot D + A' \cdot B \cdot C \cdot D' + A' \cdot B \cdot C \cdot D + A \cdot B' \cdot C' \cdot D' + A \cdot B' \cdot C' \cdot D + A \cdot B' \cdot C \cdot D' + A \cdot B' \cdot C \cdot D + A \cdot B \cdot C' \cdot D' + A \cdot B \cdot C' \cdot D + A \cdot B \cdot C \cdot D'$$

Απάντηση: Η συνάρτηση μπορεί να απλοποιηθεί χρησιμοποιώντας το χάρτη Karnaugh της επόμενης Εικόνας.

	$C' \cdot D'$	$C' \cdot D$	$C \cdot D$	$C \cdot D'$
$A' \cdot B'$	1	1		1
$A' \cdot B$	1	1		1
A.B	1	1		1
$A \cdot B'$	1	1		

Εικόνα: Χάρτης Karnaugh της συνάρτησης $Y(A,B,C,D)$

Οι “1” της συνάρτησης μπορούν να ομαδοποιηθούν ως εξής:

- τα τέσσερα γειτονικά τετράγωνα του χάρτη Karnaugh, που βρίσκονται μέσα σε **κόκκινο** περίγραμμα (αναδίπλωση), αντιστοιχούν στον όρο $A' \cdot D'$

- τα τέσσερα γειτονικά τετράγωνα του χάρτη Karnaugh, με τα πράσινα γράμματα διαφορετικού χρώματος (αναδίπλωση), αντιστοιχούν στον όρο $B \cdot D'$
- τα οκτώ μπλε γειτονικά τετράγωνα του χάρτη αντιστοιχούν στον όρο C'

Η ομαδοποίηση έχει ως αποτέλεσμα την απλοποίηση της συνάρτησης ως εξής: $Y = A' \cdot D' + B \cdot D' + C'$

Απλοποίηση με αξιοποίηση αδιάφορων όρων

Σε πολλά πρακτικά προβλήματα, η τιμή των εξόδων για συγκεκριμένους συνδυασμούς των τιμών των εισόδων είτε δε μας ενδιαφέρει, είτε θεωρούμε ότι οι εισοδοί δεν προβλέπεται να πάρουν ποτέ το συγκεκριμένο συνδυασμό τιμών. Στην περίπτωση αυτή οι τιμές των εξόδων λέμε ότι έχουν αδιάφορη τιμή (don't care value) και συμβολίζονται είτε με 'X', είτε με '-', μπορούν δε να χρησιμοποιηθούν προκειμένου να απλοποιήσουμε πιο αποτελεσματικά τις λογικές συναρτήσεις των εξόδων. Ας θεωρήσουμε το παράδειγμα της ακόλουθης συνάρτησης.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	X
0	0	1	1	X
0	1	0	0	0
0	1	0	1	X
0	1	1	0	0
0	1	1	1	0
1	0	0	0	X
1	0	0	1	X
1	0	1	0	1
1	0	1	1	1
1	1	0	0	X
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Ο χάρτης Karnaugh της συνάρτησης αυτής, φαίνεται στον ακόλουθο πίνακα.

	$C' \cdot D'$	$C' \cdot D$	$C \cdot D$	$C \cdot D'$
$A' \cdot B'$			X	X
$A' \cdot B$		X		
$A \cdot B$	X	1	1	1
$A \cdot B'$	X	X	1	1

Στην περίπτωση αυτή, μπορούμε να χρησιμοποιήσουμε τους αδιάφορους όρους όπως κρίνουμε καλύτερα για την αποδοτική υλοποίησης της συνάρτησης, και μάλιστα σε κάθε 'X' μπορούμε να αποδώσουμε διαφορετική τιμή. Έτσι, μπορούμε να θεωρήσουμε τις ακόλουθες τιμές για τα 'X' και να έχουμε τον ακόλουθο χάρτη Karnaugh, ο οποίος αντιστοιχεί στη συνάρτηση $Y = A$.

	$C' \cdot D'$	$C' \cdot D$	$C \cdot D$	$C \cdot D'$
$A' \cdot B'$				
$A' \cdot B$				
$A \cdot B$	1	1	1	1
$A \cdot B'$	1	1	1	1

2.11. Σύνθετα συνδυαστικά κυκλώματα

2.11.1 Πολυπλέκτες

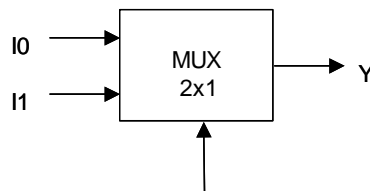
Ο Πολυπλέκτης (Multiplexer, MUX) 2^n εισόδων ($2^n \times 1$) είναι ένα συνδυαστικό κύκλωμα που έχει n γραμμές επιλογής (ελέγχου) και μία μοναδική γραμμή εξόδου. Το κύκλωμα επιλέγει δυαδικές πληροφορίες από 2^n γραμμές εισόδου, ανάλογα με τις τιμές των n γραμμών επιλογής και τις κατευθύνει στην γραμμή εξόδου. Ο συμβολισμός $2^n \times 1$ σημαίνει ότι ο πολυπλέκτης έχει 2^n εισόδους και μία έξοδο.

2.11.1.1 Υλοποίηση πολυπλεκτών

Η κύρια εφαρμογή του Πολυπλέκτη είναι η επιλογή μίας από τις πολλές πληροφορίες που εφαρμόζονται στις εισόδους του και η μεταφορά της στην έξοδό του.

Πολυπλέκτης 2 εισόδων

Ο Πολυπλέκτης 2 εισόδων (MUX 2×1) έχει δύο εισόδους I_0 και I_1 , μία είσοδο επιλογής S και μία έξοδο Y , όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Πολυπλέκτης 2 εισόδων

Ανάλογα με την τιμή της γραμμής επιλογής S , μία από τις εισόδους I_0 και I_1 μεταβιβάζεται στην έξοδο Y ως εξής: αν $S=0$, τότε $Y=I_0$, ενώ αν $S=1$, τότε $Y=I_1$. Στον επόμενο Πίνακα παρουσιάζεται ο πίνακας αληθείας, καθώς και ο συνοπτικός πίνακας αληθείας του πολυπλέκτη δύο εισόδων.

S	I1	I0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S	Y	S
0	I_0	0
1	I_1	1

Πίνακας: Πίνακας αληθείας και συνοπτικός πίνακας αληθείας πολυπλέκτη 2 εισόδων

Από τον παραπάνω πίνακα αληθείας προκύπτει ο χάρτης Karnaugh για την συνάρτηση εξόδου Y συναρτήσει της γραμμής επιλογής S και των εισόδων I_0 και I_1 που φαίνεται στην επόμενη Εικόνα.

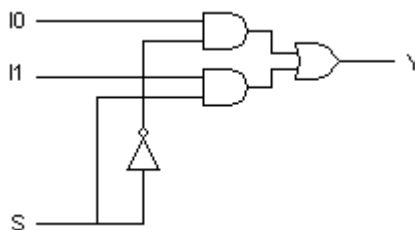
	$I1'.I0'$	$I1'.I0$	$I1.I0$	$I1.I0'$
S'		1	1	
S			1	1

Εικόνα: Χάρτης Karnaugh για την συνάρτηση εξόδου Y του Πολυπλέκτη 2 εισόδων

Από το χάρτη Karnaugh προκύπτει η ακόλουθη συνάρτηση εξόδου Y συναρτήσει της γραμμής επιλογής S και των εισόδων I0 και I1:

$$Y = I0.S' + I1.S$$

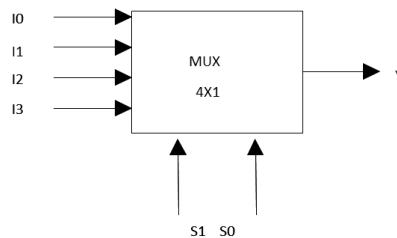
Επομένως, το κύκλωμα του Πολυπλέκτη 2 εισόδων μπορεί να υλοποιηθεί χρησιμοποιώντας μία πύλη NOT, δύο πύλες AND δύο εισόδων και μία πύλη OR δύο εισόδων, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Υλοποίηση Πολυπλέκτη 2 εισόδων με πύλες

Πολυπλέκτης 4 εισόδων

Ο Πολυπλέκτης 4 εισόδων (MUX 4x1) έχει τέσσερις εισόδους I0, I1, I2 και I3, δύο γραμμές επιλογής S0 και S1 και μία έξοδο Y, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Πολυπλέκτης 4 εισόδων

Ανάλογα με τις τιμές των γραμμών επιλογής S1 και S0, μία από τις εισόδους I0, I1, I2 και I3 μεταβιβάζεται στην έξοδο Y. Πιο συγκεκριμένα,

- αν $S1=0$ και $S0=0$, τότε $Y=I0$
- αν $S1=0$ και $S0=1$, τότε $Y=I1$
- αν $S1=1$ και $S0=0$, τότε $Y=I2$
- αν $S1=1$ και $S0=1$, τότε $Y=I3$

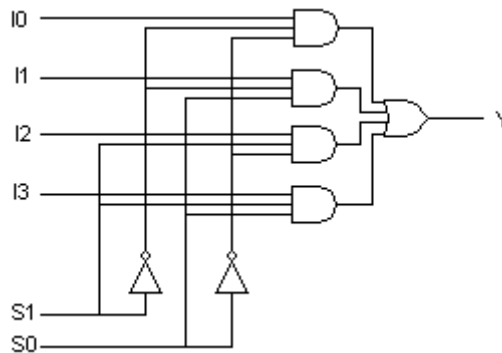
όπως φαίνεται στον επόμενο συνοπτικό πίνακα λειτουργίας.

S1	S0	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

Εικόνα: Συνοπτικός πίνακας λειτουργίας πολυπλέκτη 4 εισόδων

Η συνάρτηση εξόδου Y συναρτήσει των επιλογών S0 και S1 και των εισόδων I0, I1, I2 και I3 είναι η ακόλουθη: $Y = I_0 \cdot S_1' \cdot S_0 + I_1 \cdot S_1' \cdot S_0 + I_2 \cdot S_1 \cdot S_0' + I_3 \cdot S_1 \cdot S_0$

Με βάση τα παραπάνω, το κύκλωμα του πολυπλέκτη 4x1 μπορεί να υλοποιηθεί όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Υλοποίηση πολυπλέκτη 4 εισόδων με πύλες

Είναι εύκολο να παρατηρήσει κανείς ότι για την υλοποίηση ενός πολυπλέκτη 2^n σε 1 με n εισόδους ελέγχου, χρειάζονται n αντιστροφείς (πύλες NOT), 2^n πύλες AND (n+1)- εισόδων, και μία πύλη OR 2^n εισόδων.

2.11.1.2 Υλοποίηση λογικών συναρτήσεων με τη χρήση πολυπλεκτών

Μία ενδιαφέρουσα εφαρμογή των πολυπλεκτών είναι η υλοποίηση λογικών συναρτήσεων και συνδυαστικών κυκλωμάτων. Πιο συγκεκριμένα, είναι εύκολο να δειχθεί ότι οποιαδήποτε λογική συνάρτηση n μεταβλητών μπορεί να υλοποιηθεί με έναν πολυπλέκτη 2^{n-1} εισόδων. Η διαδικασία αντιστοίχισης των εισόδων του πολυπλέκτη στις μεταβλητές εισόδου της συνάρτησης είναι η ακόλουθη:

- Προσδιορίζουμε τον πίνακα αληθείας της συνάρτησης, αριθμώντας τις μεταβλητές εισόδου από I_n έως I_1 (I_n είναι το MSB και I_1 το LSB).
- Ορίζουμε τις n-1 μεταβλητές I_n έως I_2 στις n-1 εισόδους επιλογής.
- Εξετάζουμε ζεύγη γειτονικών γραμμών στον πίνακα σε καθένα από τα οποία διαφέρει μόνο η τιμή της μεταβλητής εισόδου I_1 .
- Για κάθε ένα από τα ζεύγη αυτά προσδιορίζουμε κατά πόσο η τιμή της συνάρτησης (εξόδου) για το συνδυασμό $(I_n, I_{n-1}, \dots, I_2, 0)$ και $(I_n, I_{n-1}, \dots, I_2, 1)$ είναι (0,0), (0,1), (1,0), or (1,1).
- Για κάθε συνδυασμό $(I_n, I_{n-1}, \dots, I_2)$, ορίζουμε 0, D, D', ή 1 στην είσοδο δεδομένων που αντιστοιχεί στο $(I_n, I_{n-1}, \dots, I_2)$.

Η διαδικασία μπορεί να φανεί καλύτερα με τη βοήθεια των ακόλουθων παραδειγμάτων.

Παράδειγμα: Έστω η λογική συνάρτηση Y τριών μεταβλητών A , B και C :

$Y(A, B, C) = A' \cdot B' \cdot C' + A' \cdot B \cdot C + A \cdot B \cdot C$, ο πίνακας αληθείας της οποίας παρουσιάζεται στον επόμενο Πίνακα.

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

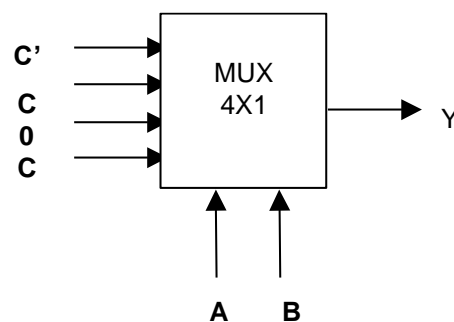
(α)

A	B	Y
0	0	C'
0	1	C
1	0	0
1	1	C

(β)

Πίνακας: Πίνακας Αληθείας της λογικής συνάρτησης $Y=A' \cdot B' \cdot C' + A' \cdot B \cdot C + A \cdot B \cdot C$ και πίνακας υλοποίησης με τη βοήθεια πολυπλέκτη

Απάντηση: Το πλήθος των μεταβλητών της λογικής συνάρτησης είναι: $n=3$. Επομένως, η συνάρτηση μπορεί να υλοποιηθεί χρησιμοποιώντας έναν Πολυπλέκτη 4 εισόδων ($2^2 \times 1$). Οι μεταβλητές A , B χρησιμοποιούνται ως γραμμές επιλογής του Πολυπλέκτη. Οι τέσσερις εισοδοί I_0 , I_1 , I_2 , I_3 του Πολυπλέκτη επιλέγονται κατάλληλα από τον πίνακα αληθείας της λογικής συνάρτησης με τη βοήθεια του πίνακα υλοποίησης της ροηγούμενης Εικόνας. Οι εισοδοί του πολυπλέκτη μπορούν να είναι είτε 0, 1, C , ή C' , όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Υλοποίηση της λογικής συνάρτησης $Y=A' \cdot B' \cdot C' + A' \cdot B \cdot C + A \cdot B \cdot C$ με Πολυπλέκτη 4 εισόδων

Με βάση τα παραπάνω, οποιοδήποτε συνδυαστικό κύκλωμα n εισόδων και m εξόδων μπορεί να υλοποιηθεί με m πολυπλέκτες 2^{n-1} εισόδων.

Παράδειγμα: Να υλοποιηθεί συνδυαστικό κύκλωμα που αναγνωρίζει το πλήθος των “1” ενός δυαδικού αριθμού 3-bits χρησιμοποιώντας πολυπλέκτες.

Απάντηση: Το συνδυαστικό κύκλωμα έχει τρεις εισόδους A , B και C που χρησιμοποιούνται για το δυαδικό αριθμό. Το πλήθος των “1” του 3-bits δυαδικού αριθμού είναι 0 ή 1 ή 2 ή 3. Επομένως, το συνδυαστικό κύκλωμα έχει δύο εξόδους Y_1 και Y_2 που χρησιμοποιούνται για το πλήθος των “1” του 3-

bits δυαδικού αριθμού. Ο πίνακας αληθείας του συνδυαστικού κυκλώματος καθώς και οι πίνακες υλοποίησης των εξόδων με τη βοήθεια πολυπλεκτών παρουσιάζονται στον επόμενο Πίνακα.

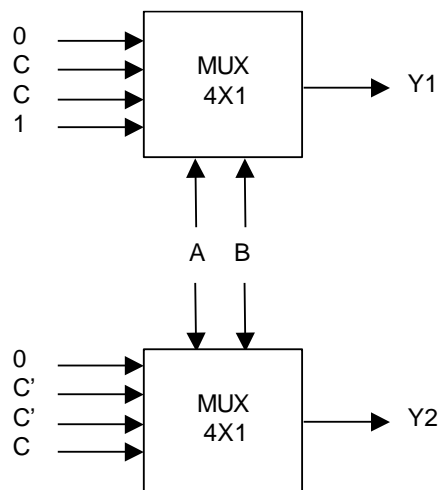
A	B	C	Πλήθος "1"	Y1	Y2
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	2	1	0
1	0	0	1	0	1
1	0	1	2	1	0
1	1	0	2	1	0
1	1	1	3	1	1

A	B	Y1
0	0	0
0	1	C
1	0	C
1	1	1

A	B	Y2
0	0	C
0	1	C'
1	0	C'
1	1	C

Πίνακας: Πίνακας Αληθείας του συνδυαστικού κυκλώματος που αναγνωρίζει το πλήθος των "1" ενός 3-bits δυαδικού αριθμού

Το συνδυαστικό κύκλωμα έχει $n=3$ εισόδους και $m=2$ εξόδους. Το πλήθος των μεταβλητών των δύο λογικών συναρτήσεων εξόδων είναι: $n=3$. Επομένως, κάθε συνάρτηση μπορεί να υλοποιηθεί χρησιμοποιώντας έναν Πολυπλέκτη $4 \times 1 (2^2 \times 1)$. Άρα, το συνδυαστικό κύκλωμα μπορεί να υλοποιηθεί χρησιμοποιώντας δύο ($m=2$) πολυπλέκτες 4×1 , όπως φαίνεται στην επόμενη Εικόνα. Οι μεταβλητές A, B και C χρησιμοποιούνται ως οι τρεις (κοινές) επιλογές των δύο πολυπλεκτών. Οι εισοδοί των δύο πολυπλεκτών επιλέγονται κατάλληλα από τους Πίνακες Αληθείας των λογικών συναρτήσεων. Οι εξοδοί των πολυπλεκτών αποτελούν τις εξόδους του συνδυαστικού κυκλώματος.



Εικόνα: Υλοποίηση του συνδυαστικού κυκλώματος που αναγνωρίζει το πλήθος των "1" ενός 3-bits δυαδικού αριθμού με δύο πολυπλέκτες 4×1

2.11.2 Αποπολυπλέκτες

Ο **Αποπολυπλέκτης** (Demultiplexer, DEMUX) 1×2^n είναι ένα συνδυαστικό κύκλωμα που έχει μία είσοδο, n γραμμές επιλογής και 2^n γραμμές εξόδου. Το κύκλωμα δέχεται πληροφορίες από την γραμμή εισόδου και τις μεταβιβάζει σε μία από τις 2^n γραμμές εξόδου, ανάλογα με την τιμή των n γραμμών επιλογής. Ο συμβολισμός 1×2^n σημαίνει ότι ο αποπολυπλέκτης έχει μία είσοδο και 2^n εξόδους.

Αποπολυπλέκτης 1×2

Ο αποπολυπλέκτης 1×2 έχει μία είσοδο E , μία γραμμή επιλογής I_0 και δύο εξόδους D_0 και D_1 . Η είσοδος E μεταβιβάζεται στην γραμμή εξόδου που υποδεικνύει το bit I_0 της γραμμής επιλογής. Ο πίνακας αληθείας του αποπολυπλέκτη 1×2 παρουσιάζεται στον επόμενο Πίνακα.

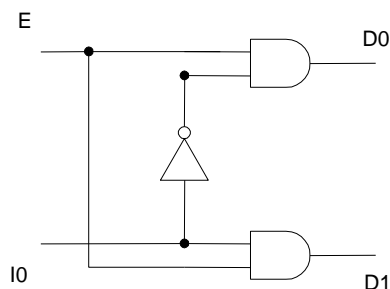
E	I ₀	D ₀	D ₁
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

Πίνακας: Πίνακας αληθείας του αποπολυπλέκτη 1×2

Στον πίνακα αληθείας του αποπολυπλέκτη 1×2 φαίνεται ότι όταν η γραμμή επιλογής είναι $I_0=0$, τότε η είσοδος E μεταβιβάζεται στην έξοδο D_0 (αν $E=0$ τότε $D_0=0$ στην πρώτη γραμμή και αν $E=1$ τότε $D_0=1$ στην τρίτη γραμμή) ενώ όταν η γραμμή επιλογής είναι $I_0=1$, τότε η είσοδος E μεταβιβάζεται στην έξοδο D_1 (αν $E=0$ τότε $D_1=0$ στην δεύτερη γραμμή και αν $E=1$ τότε $D_1=1$ στην τέταρτη γραμμή). Όλες οι υπόλοιπες έξοδοι (που δεν επιλέγονται) λαμβάνουν τιμές ίσες με “0” (ή ίσες με “1”, ανάλογα με τη σχεδίαση του αποπολυπλέκτη). Από τον πίνακα αληθείας του αποπολυπλέκτη 1×2 προκύπτουν οι ακόλουθες συναρτήσεις εξόδων D_0 και D_1 συναρτήσει εισόδου E και της γραμμής επιλογής I_0 . Επομένως, το κύκλωμα του Αποπολυπλέκτη 1×2 μπορεί να υλοποιηθεί, όπως φαίνεται στην επόμενη Εικόνα.

$$D_0 = E \cdot I_0'$$

$$D_1 = E \cdot I_0$$



Εικόνα: Υλοποίηση αποπολυπλέκτη 1×2 με πύλες

Αποπολυπλέκτης 1×4

Ο Αποπολυπλέκτης 1×4 έχει μία είσοδο E , δύο γραμμές επιλογής I_1 και I_0 και τέσσερις εξόδους D_0 , D_1 , D_2 και D_3 . Η είσοδος E μεταβιβάζεται στην γραμμή εξόδου που υποδεικνύουν οι είσοδοι επιλογής I_1 και I_0 . Ο πίνακας αληθείας του αποπολυπλέκτη 1×4 παρουσιάζεται στον επόμενο Πίνακα.

E	I1	I0	D0	D1	D2	D3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	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

Εικόνα: Πίνακας Αληθείας του αποπολυπλέκτη 1x4

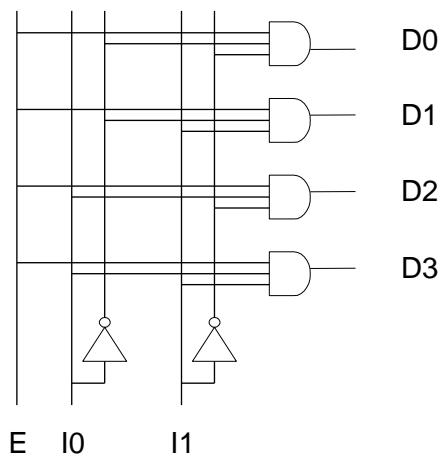
Από τον πίνακα αληθείας του αποπολυπλέκτη 1x4 προκύπτουν οι ακόλουθες συναρτήσεις εξόδων D0, D1, D2 και D3 συναρτήσει εισόδου E και των επιλογών I0 και I1. Επομένως το κύκλωμα του Αποπολυπλέκτη 1x4 μπορεί να υλοποιηθεί χρησιμοποιώντας δύο πύλες NOT και τέσσερις πύλες AND τριών εισόδων, όπως φαίνεται στην επόμενη Εικόνα.

$$D0 = E \cdot I1' \cdot I0'$$

$$D1 = E \cdot I1' \cdot I0$$

$$D2 = E \cdot I1 \cdot I0'$$

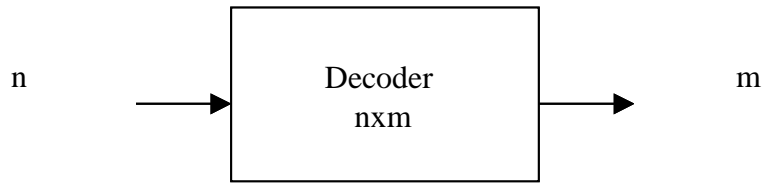
$$D3 = E \cdot I1 \cdot I0$$



Εικόνα: Υλοποίηση αποπολυπλέκτη 1x4

2.11.3 Αποκωδικοποιητές

Ο αποκωδικοποιητής (decoder) από n σε m (nxm) είναι ένα συνδυαστικό κύκλωμα με n γραμμές εισόδου και m γραμμές εξόδου ($m \leq 2^n$), όπως φαίνεται στην επόμενη Εικόνα. Κάθε μία από τις n εισόδους του αποκωδικοποιητή μπορεί να είναι “0” ή “1”, οπότε υπάρχουν 2^n διαφορετικοί συνδυασμοί. Για κάθε συνδυασμό εισόδου μόνο μία από τις εξόδους του αποκωδικοποιητή είναι “1” (είναι ενεργοποιημένη).



Εικόνα: Αποκωδικοποιητής nxm

Υπάρχουν αποκωδικοποιητές που χρησιμοποιούν όλους τους δυνατούς συνδυασμούς εισόδου ($m=2^n$), όπως είναι ο αποκωδικοποιητής 3x8, και αποκωδικοποιητές που χρησιμοποιούν λιγότερους συνδυασμούς εισόδου ($m<2^n$), όπως είναι ο αποκωδικοποιητής 4x10.

Αποκωδικοποιητής 3x8

Ο Αποκωδικοποιητής 3x8 χρησιμοποιεί όλους τους δυνατούς συνδυασμούς εισόδου. Έχει τρεις εισόδους C, B και A που αντιστοιχούν σε έναν δυαδικό αριθμό τριών ψηφίων και οκτώ εξόδους D0, D1, D2, D3, D4, D5, D6 και D7. Για κάθε συνδυασμό εισόδου μόνο μία από τις εξόδους του αποκωδικοποιητή είναι “1” (αυτή που αντιστοιχεί στον κωδικό εισόδου) και οι άλλες εξόδοι είναι “0”, όπως φαίνεται στον ακόλουθο Πίνακα.

C	B	A	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Πίνακας: Πίνακας Αληθείας του Αποκωδικοποιητή 3x8

Από τον πίνακα αληθείας του αποκωδικοποιητή 3x8 προκύπτουν οι ακόλουθες συναρτήσεις εξόδου:

$$D0=C' \cdot B' \cdot A'$$

$$D1=C' \cdot B' \cdot A$$

$$D2=C' \cdot B \cdot A'$$

$$D3=C' \cdot B \cdot A$$

$$D4=C \cdot B' \cdot A'$$

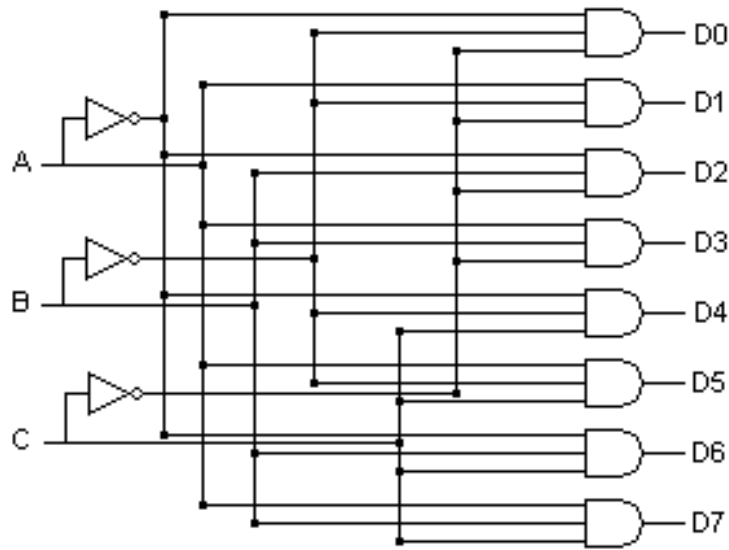
$$D5=C \cdot B' \cdot A$$

$$D6=C \cdot B \cdot A'$$

$$D7=C \cdot B \cdot A$$

Προφανώς, ο αποκωδικοποιητής 3x8 παράγει στις εξόδους του τους οκτώ ($2^3=8$) ελάχιστους όρους των τριών μεταβλητών εισόδου του. Γενικά, ο αποκωδικοποιητής nx 2^n παράγει στις εξόδους του τους 2^n ελάχιστους όρους των n μεταβλητών εισόδου του.

Ο αποκωδικοποιητής 3x8 μπορεί να υλοποιηθεί όπως φαίνεται στην ακόλουθη Εικόνα.



Εικόνα: Αποκωδικοποιητής 3x8

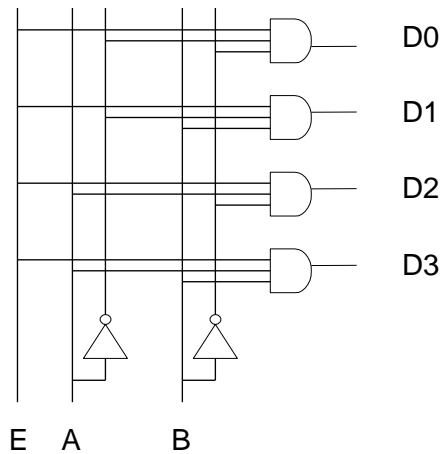
Υλοποίηση αποκωδικοποιητών με είσοδο επίτρεψης (enable)

Στην ψηφιακή σχεδίαση και στα υπολογιστικά συστήματα υπάρχει σε πολλές περιπτώσεις η ανάγκη καμία από τις εξόδους ενός αποκωδικοποιητή να μην είναι ενεργοποιημένη, ανάλογα με την τιμή ενός σήματος επίτρεψης (Enable). Στην περίπτωση αυτή, ο πίνακας αληθείας του κυκλώματος είναι όπως φαίνεται στον ακόλουθο πίνακα, για την περίπτωση ενός 2x4 αποκωδικοποιητή.

E	B	A	D0	D1	D2	D3
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
0	X	X	0	0	0	0

Πίνακας: Πίνακας αληθείας αποκωδικοποιητή 2x4 με είσοδο επίτρεψης

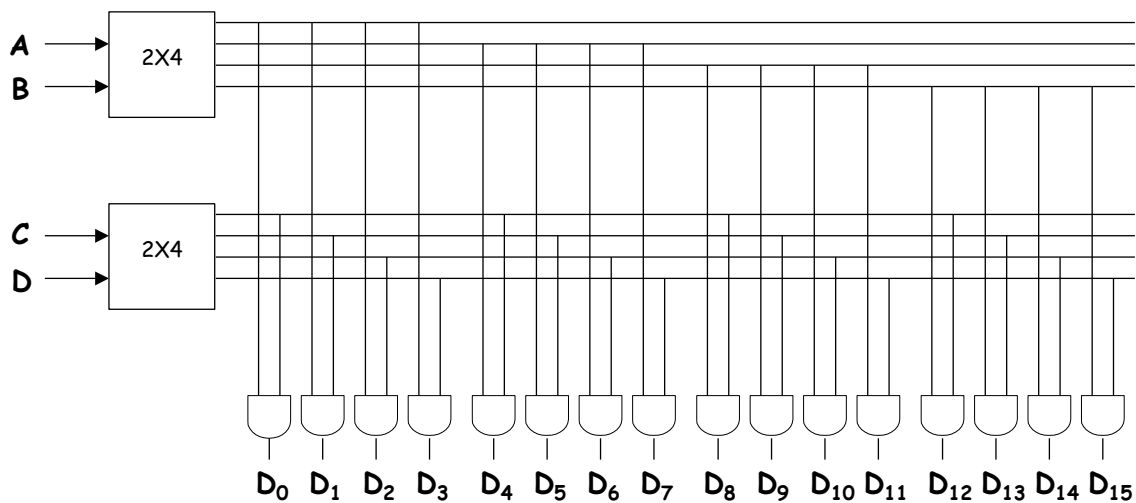
Η υλοποίηση ενός τέτοιου κυκλώματος φαίνεται στην ακόλουθη Εικόνα. Αξίζει να παρατηρήσει κανείς ότι το κύκλωμα στην περίπτωσης αυτή είναι όμοιο με το κύκλωμα του αποπολυπλέκτη 1x4.



Εικόνα: Αποκωδικοποιητής 2×4 με επίτρευση

Υλοποίηση μεγαλύτερων αποκωδικοποιητών με τη χρήση μικρότερων

Για μεγάλες τιμές του πλήθους των εξόδων του αποκωδικοποιητή, το πλήθος των πυλών και το πλήθος των εισόδων κάθε πύλης αυξάνεται ασύμφορα. Για παράδειγμα, είναι εύκολο να δει κανείς ότι για να υλοποιήσουμε ένα αποκωδικοποιητή 8×256, θα χρειαζόμαστε 256 πύλες των 8 εισόδων. Για το λόγο αυτό, και προκειμένου να μειώσουμε το κόστος υλοποίησης, σε τέτοιες περιπτώσεις χρησιμοποιούμε μια μεθοδολογία υλοποίησης μεγαλύτερων αποκωδικοποιητών με τη χρήση μικρότερων. Έτσι, ας υποθέσουμε ότι θέλουμε να υλοποιήσουμε ένα αποκωδικοποιητή 4×16. Η υλοποίηση με τη χρήση της τεχνικής που μόλις αναφέραμε, φαίνεται στην ακόλουθη Εικόνα.



Εικόνα: Αποκωδικοποιητής 4×16 υλοποιημένος χρησιμοποιώντας αποκωδικοποιητές 2×4 και πύλες AND

Αποκωδικοποιητής BCD σε δεκαδικό (4x10)

Ο Αποκωδικοποιητής BCD σε δεκαδικό (4x10) έχει τέσσερις εισόδους D, C, B και A και δέκα εξόδους D0, D1, D2, D3, D4, D5, D6, D7, D8 και D9 και δε χρησιμοποιεί όλους τους δυνατούς συνδυασμούς εισόδου. Υπάρχουν δεκαέξι ($2^4=16$) συνδυασμοί εισόδου: οι δέκα πρώτοι αντιστοιχούν στους (δέκα) BCD κωδικούς (0000 .. 1001) και οι υπόλοιποι έξι (6) είναι μη χρησιμοποιούμενοι. Για κάθε

χρησιμοποιούμενο συνδυασμό εισόδου μόνον μία από τις εξόδους του αποκωδικοποιητή είναι “1” (αυτή που αντιστοιχεί στον κωδικό εισόδου) και οι άλλες εξόδους είναι “0”. Για τους μη χρησιμοποιούμενους συνδυασμούς εισόδου όλες οι εξόδους είναι “0”. Ο πίνακας αληθείας του αποκωδικοποιητή 4x10 παρουσιάζεται στον ακόλουθο Πίνακα.

D	C	B	A	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
0	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	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0

Πίνακας: Πίνακας Αληθείας του Αποκωδικοποιητή 4x10

Υλοποίηση συνδυαστικών κυκλωμάτων με αποκωδικοποιητές

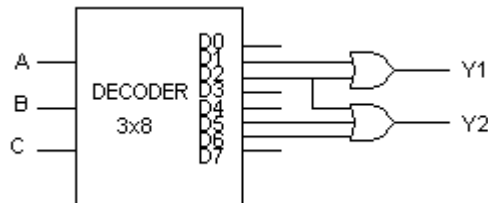
Όπως αναφέρθηκε, ο αποκωδικοποιητής $n \times 2^n$ παράγει στις εξόδους του τους 2^n ελάχιστους όρους των n μεταβλητών εισόδου του. Επίσης είναι γνωστό ότι κάθε λογική συνάρτηση μπορεί να εκφραστεί ως άθροισμα ελάχιστων όρων. Επομένως, κάθε λογική συνάρτηση n μεταβλητών μπορεί να υλοποιηθεί με έναν αποκωδικοποιητή $n \times 2^n$ και μία πύλη OR οι εισοδοί της οποίας τροφοδοτούνται από τις εξόδους του Αποκωδικοποιητή που αντιστοιχούν στους ελάχιστους όρους που η συνάρτηση έχει την τιμή "1". Επομένως, οποιοδήποτε συνδυαστικό κύκλωμα n εισόδων και m εξόδων μπορεί να υλοποιηθεί με έναν αποκωδικοποιητή $n \times 2^n$ και m πύλες OR, οι εισοδοί των οποίων τροφοδοτούνται κατάλληλα από τις εξόδους του Αποκωδικοποιητή.

Παράδειγμα: Ένα συνδυαστικό κύκλωμα έχει τρεις (3) εισόδους A, B και C και δύο (2) εξόδους:

$$Y1(A,B,C)=A \cdot B \cdot C + A \cdot B \cdot C'$$

$$Y2(A,B,C)= A \cdot B \cdot C' + A \cdot B' \cdot C + A \cdot B \cdot C'$$

Το πλήθος των εισόδων του κυκλώματος είναι $n=3$ και το πλήθος των εξόδων του κυκλώματος είναι $m=2$. Επομένως, το κύκλωμα μπορεί να υλοποιηθεί χρησιμοποιώντας έναν αποκωδικοποιητή 3×8 ($n \times 2^n$) και δύο (m) πύλες OR. Η μία πύλη OR δύο εισόδων υλοποιεί την συνάρτηση $Y1$ και οι εισοδοί της τροφοδοτούνται από τις εξόδους του Αποκωδικοποιητή που αντιστοιχούν στους ελάχιστους όρους που η συνάρτηση $Y1$ έχει την τιμή "1". Με την ίδια λογική τροφοδοτούνται οι εισοδοί της πύλης OR τριών εισόδων που υλοποιεί την συνάρτηση $Y2$. Το κύκλωμα φαίνεται στην ακόλουθη Εικόνα.



Εικόνα: Υλοποίηση συνδυαστικού κυκλώματος με έναν αποκωδικοποιητή και δύο πύλες OR

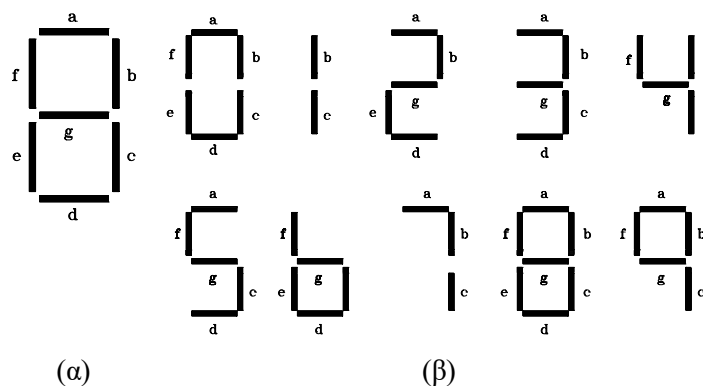
Αποκωδικοποιητές οδηγοί display 7 τμημάτων

Οι ενδείκτες (displays) δεκαδικών ψηφίων χρησιμοποιούν επτά (7) τμήματα (segments) για να αναπαραστήσουν τους δεκαδικούς αριθμούς 0-9.

Στα υπολογιστικά και ηλεκτρονικά συστήματα υπάρχουν ενδείκτες όπου χρησιμοποιούνται οι δίοδοι εκπομπής φωτός (Light Emitting Diodes, LEDs) για την κατασκευή των τμημάτων τους. Η λειτουργία τους βασίζεται στο γεγονός ότι κάθε τμήμα αποτελείται από υλικό το οποίο εκπέμπει φως όταν διαρρέεται από ρεύμα.

Επίσης, υπάρχουν ενδείκτες υγρού κρυστάλλου (Liquid Crystal Displays, LCDs), η λειτουργία των οποίων βασίζεται στην ιδιότητα ενός ειδικού υγρού κρυστάλλου να διαδίδει διαφορετικά το φως υπό την επίδραση εναλλασσόμενου ηλεκτρικού πεδίου. Τα LCDs έχουν ιδιαίτερα χαμηλή κατανάλωση ισχύος και είναι ιδανικά για φορητές συσκευές.

Στην ακόλουθη Εικόνα παρουσιάζεται ένας ενδείκτης δεκαδικών ψηφίων (display επτά τμημάτων). Στο (β) παρουσιάζεται ο τρόπος εμφάνισης των δεκαδικών ψηφίων 0-9 στον ενδείκτη δεκαδικών ψηφίων (display).



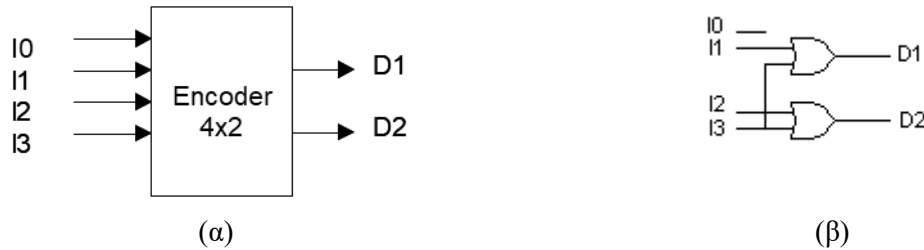
Εικόνα: (α) Ενδείκτης δεκαδικών ψηφίων επτά τμημάτων και (β) τρόπος εμφάνισης των δεκαδικών ψηφίων 0-9 στον ενδείκτη δεκαδικών ψηφίων

2.11.4 Κωδικοποιητές

Ο Κωδικοποιητής (encoder) από m σε n ($m \times n$) είναι ένα συνδυαστικό κύκλωμα με m γραμμές εισόδου και n γραμμές εξόδου ($m \leq 2^n$). Από τις m γραμμές εισόδου του κωδικοποιητή, μόνο μία επιτρέπεται να είναι ενεργοποιημένη. Στην έξοδο παράγεται ένας n -bits κωδικός που αντιστοιχεί στην ενεργοποιημένη είσοδο.

Κωδικοποιητής 4x2

Ο κωδικοποιητής 4x2 είναι ένα συνδυαστικό κύκλωμα που έχει τέσσερις ($m=4$) γραμμές εισόδου και δύο ($n=2$) γραμμές εξόδου ($m=2^n$), όπως φαίνεται στην ακόλουθη Εικόνα.



Εικόνα: Κωδικοποιητής 4x2 (α) και υλοποίηση με πύλες (β)

Ο κωδικοποιητής 4x2 παράγει στην έξοδό του το δυαδικό κώδικα που αντιστοιχεί στις εισόδους του και ο Πίνακας Αληθείας του παρουσιάζεται στον ακόλουθο Πίνακα.

I0	I1	I2	I3	D2	D1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Πίνακας: Πίνακας Αληθείας Κωδικοποιητή 4x2

Οι συναρτήσεις των εξόδων του κωδικοποιητή 4x2 είναι οι ακόλουθες:

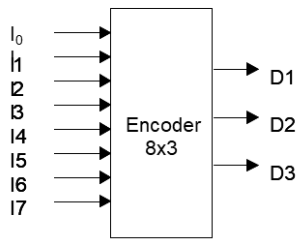
$$D2 = I2 + I3$$

$$D1 = I1 + I3$$

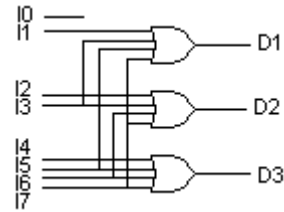
Το κύκλωμα που υλοποιεί τον Κωδικοποιητή 4x2 αποτελείται μόνο από πύλες OR και παρουσιάζεται στην επόμενη Εικόνα.

Κωδικοποιητής 8x3

Ο κωδικοποιητής 8x3 είναι ένα συνδυαστικό κύκλωμα με οκτώ ($m=8$) γραμμές εισόδου και τρεις ($n=3$) γραμμές εξόδου ($m=2^n$), όπως φαίνεται στην επόμενη Εικόνα.



(α)



(β)

Εικόνα: Κωδικοποιητής 8x3 (α) σχηματικό διάγραμμα (β) υλοποίηση

Ο κωδικοποιητής 8x3 παράγει στην έξοδό του το δυαδικό κώδικα που αντιστοιχεί στις εισόδους του και ο πίνακας αληθείας του παρουσιάζεται στον επόμενο Πίνακα.

I0	I1	I2	I3	I4	I5	I6	I7	D3	D2	D1
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Πίνακας: Πίνακας Αληθείας Κωδικοποιητή 8x3

Οι συναρτήσεις των εξόδων του κωδικοποιητή 8x3 είναι οι ακόλουθες:

$$D3 = I4 + I5 + I6 + I7$$

$$D2 = I2 + I3 + I6 + I7$$

$$D1 = I1 + I3 + I5 + I7$$

Κωδικοποιητές προτεραιότητας

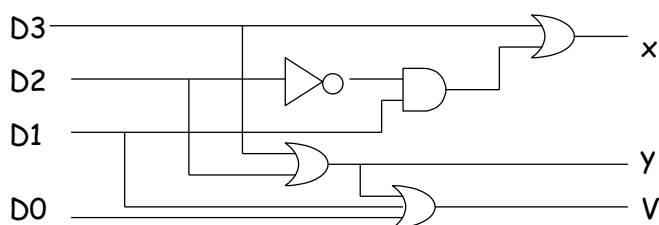
Όπως είδαμε, στην περίπτωση που μόνο μια είσοδος ενός κωδικοποιητή είναι ενεργοποιημένη, η έξοδος δείχνει ποια είναι η έξοδος αυτή. Στην περίπτωση όμως που περισσότερες από μια εισόδους του κωδικοποιητή μπορούν να είναι ταυτόχρονα ενεργές, τα πράγματα περιπλέκονται. Για το λόγο αυτό έχουν αναπτυχθεί και χρησιμοποιούνται στα υπολογιστικά συστήματα μονάδες που ονομάζονται κωδικοποιητές προτεραιότητας. Οι εισοδοί των μονάδων αυτών δεν είναι ισοδύναμες, αλλά κάθε μία έχει τη δική της προτεραιότητα. Αν μια είσοδος με υψηλότερη προτεραιότητα είναι ενεργοποιημένη, οι εισοδοί με χαμηλότερη προτεραιότητα από αυτή αγνοούνται, ακόμα κι αν έχουν ενεργοποιηθεί. Για παράδειγμα, ο πίνακας αληθείας ενός κωδικοποιητή προτεραιότητας με τέσσερις εισόδους φαίνεται στον επόμενο Πίνακα. Ο κωδικοποιητής αυτός έχει μια επιπλέον έξοδο V, η οποία έχει την τιμή 0, όταν όλες οι εισοδοί

του κωδικοποιητή έχουν την τιμή 0. Στην περίπτωση αυτή οι υπόλοιπες εξόδους του κωδικοποιητή είναι αδιάφορες.

D0	D1	D2	D3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

Πίνακας: Πίνακας αληθείας κωδικοποιητή προτεραιότητας με τέσσερις εισόδους

Εφαρμόζοντας απλοποίηση με χάρτες Karnaugh στις συναρτήσεις εξόδου του κωδικοποιητή προτεραιότητας προκύπτει η υλοποίηση της επόμενης Εικόνας.



Εικόνα: Κωδικοποιητής προτεραιότητας με τέσσερις εισόδους

2.11.5 Συγκριτές

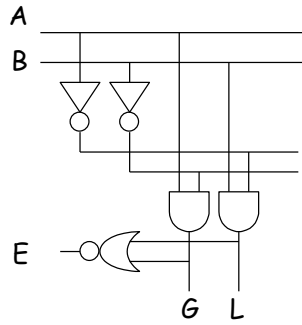
Ένας συγκριτής είναι ένα συνδυαστικό κύκλωμα το οποίο δέχεται ως είσοδο δύο αριθμούς (οι οποίοι μπορεί να αποτελούνται από ένα ή περισσότερα δυαδικά ψηφία) και έχει δύο εξόδους, τις οποίες ονομάζουμε συνήθως G (Greater) και L (Less), και ενεργοποιούνται ανάλογα με το αν ο πρώτος αριθμός είναι μεγαλύτερος ή μικρότερος από το δεύτερο. Μπορούμε ακόμη να δημιουργήσουμε και μια τρίτη έξοδο E (Equal), η οποία θα ενεργοποιείται όταν οι δύο αριθμοί είναι ίσοι.

Στην πιο απλή περίπτωση, οι προς σύγκριση αριθμοί είναι μονοψήφιοι. Στην περίπτωση αυτή, οι συναρτήσεις εξόδου δίνονται στον επόμενο Πίνακα.

A	B	G	L	E
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

Πίνακας: Πίνακας αληθείας του συγκριτή μεγέθους 1-ψηφίων δυαδικών αριθμών

Αξίζει να σημειωθεί ότι η έξοδος E μπορεί να ληφθεί ως το NOR των δύο άλλων εξόδων. Πράγματι, η E ενεργοποιείται όταν καμία από τις άλλες δύο δεν είναι ενεργοποιημένη. Το λογικό κύκλωμα του συγκριτή 1-bit δίνεται στην επόμενη Εικόνα.



Εικόνα: Συγκριτής 1-bit

Για τη σύγκριση αριθμών περισσότερων ψηφίων είναι εύκολο να υλοποιήσουμε, για κάθε περίπτωση, τον αντίστοιχο πίνακα αληθείας. Αξίζει όμως να σημειωθεί ότι ο πίνακας αληθείας του συγκριτή δυαδικών αριθμών 2 ψηφίων θα αποτελείται από 16 γραμμές, του συγκριτή αριθμών 3 ψηφίων από 64 γραμμές, ενώ για αριθμούς 4 ψηφίων, φτάνουμε στις 256 γραμμές. Για το λόγο αυτό, χρησιμοποιούμε ένα κύκλωμα ‘συγκριτή ριπής’, το οποίο βασίζεται στην απλή σκέψη ότι ο αριθμός A θα είναι μεγαλύτερος από το B αν το υψηλότερης τάξης ψηφίο του είναι μεγαλύτερο από το αντίστοιχο του B, μικρότερος αν το υψηλής τάξης ψηφίο του είναι μικρότερο από το αντίστοιχο του B, ενώ αν τα υψηλής τάξης ψηφία είναι ίσα, τότε η σύγκριση μεταφέρεται στο χαμηλότερης τάξης ψηφίο. Με βάση τη σκέψη αυτή, μπορούμε να ορίσουμε ένα συγκριτή όπως φαίνεται στους ακόλουθους πίνακες.

G_{i-1}	L_{i-1}	A_i	B_i	G_i	L_i
0	0	0	0		
0	0	0	1		1
0	0	1	0	1	
0	0	1	1		
0	1	0	0		1
0	1	0	1		1
0	1	1	0	1	
0	1	1	1		1
1	0	0	0	1	
1	0	0	1		1
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

G_i

	$G_{i-1}L_{i-1}$			
A_iB_i	00	01	11	10
00			X	1
01			X	
11			X	1
10	1	1	X	1

L_i

	$G_{i-1}L_{i-1}$			
A_iB_i	00	01	11	10
00		1	X	
01	1	1	X	1
11		1	X	
10			X	

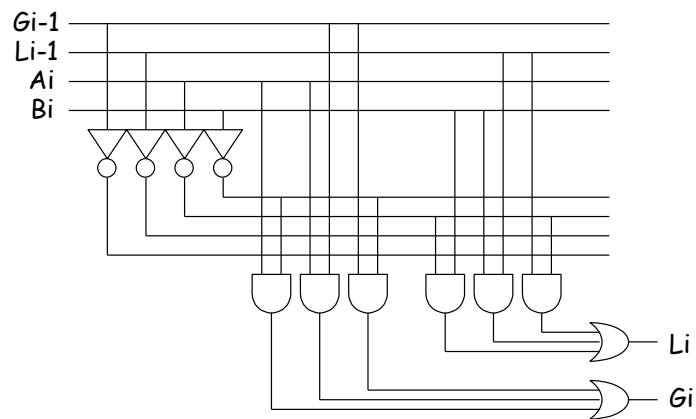
Εικόνα: Πίνακας αληθείας συγκριτή και χάρτες Karnaugh των συναρτήσεων εξόδου

Από τους χάρτες Karnaugh προκύπτουν οι ακόλουθες απλοποιημένες συναρτήσεις εξόδου του συγκριτή:

$$G_i = A_i \cdot B_i' + A_i \cdot G_{i-1} + B_i' \cdot G_{i-1}$$

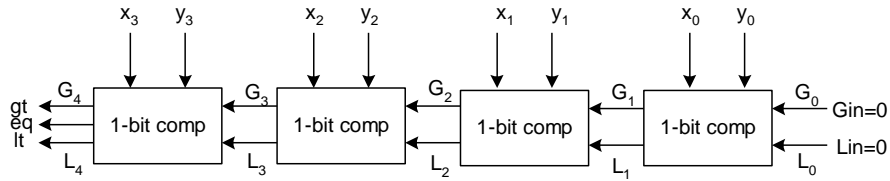
$$L_i = A_i' \cdot B_i + B_i \cdot L_{i-1} + A_i' \cdot L_{i-1}$$

Το αντίστοιχο κύκλωμα φαίνεται στην επόμενη Εικόνα.



Εικόνα: Συγκριτής ριθής

Χρησιμοποιώντας τη μονάδα συγκριτή της ανωτέρω Εικόνας μπορούμε να κατασκευάσουμε συγκριτή οσωνδήποτε ψηφίων, όπως φαίνεται στην επόμενη Εικόνα για την περίπτωση των τεσσάρων δυαδικών ψηφίων.



Εικόνα: Συγκριτής τεσσάρων ψηφίων

2.12 Αριθμητικές μονάδες

Όπως έχει αναφερθεί, μια από τις πιο σημαντικές λειτουργίες που εκτελούν τα υπολογιστικά συστήματα είναι οι αριθμητικές πράξεις. Οι πιο σημαντικές πράξεις, πάνω στις οποίες στηρίζονται οι υπόλοιπες, είναι η πρόσθεση και η αφαίρεση, ενώ σημαντικό ρόλο στην απόδοση ενός υπολογιστικού συστήματος παίζουν και οι μονάδες που εκτελούν πολλαπλασιασμό.

2.12.1 Αθροιστές

Η πιο βασική αριθμητική πράξη είναι η πρόσθεση δύο δυαδικών ψηφίων. Στην πρόσθεση δύο δυαδικών ψηφίων υπάρχουν τέσσερις δυνατές περιπτώσεις: $0+0=0$, $0+1=1$, $1+0=1$, $1+1=10$. Οι τρεις πρώτες πράξεις δημιουργούν ένα άθροισμα που το μήκος του είναι ένα ψηφίο. Όταν και οι δύο προσθετέοι είναι 1, το δυαδικό άθροισμα αποτελείται από δύο ψηφία. Το πιο σημαντικό από αυτά τα δύο ψηφία ονομάζεται 'κρατούμενο'. Όταν οι προσθετέοι περιέχουν και άλλα σημαντικά ψηφία, το κρατούμενο που βγαίνει από την πρόσθεση προστίθεται στο επόμενο μεγαλύτερης σημαντικότητας ζευγάρι δυαδικών ψηφίων. Ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση δυο δυαδικών ψηφίων λέγεται 'ημιαθροιστής'. Ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση τριών δυαδικών ψηφίων (δύο σημαντικών ψηφίων και ενός προηγούμενου κρατούμενου) λέγεται 'πλήρης αθροιστής'. Το όνομα του ημιαθροιστή προέρχεται από το γεγονός ότι δύο ημιαθροιστές μπορούν να χρησιμοποιηθούν για να υλοποιήσουν έναν πλήρη αθροιστή, όπως θα δούμε στη συνέχεια.

Ημιαθροιστής

Σύμφωνα με τα προηγούμενα, ο ημιαθροιστής είναι ένα συνδυαστικό κύκλωμα με δύο εξόδους, x και y , και δύο εξόδους s (sum, άθροισμα) και C (Carry, κρατούμενο), του οποίου ο πίνακας αληθείας δίνεται στη συνέχεια.

x	y	s	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

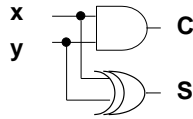
Πίνακας: Πίνακας αληθείας ημιαθροιστή

Είναι δυνατό να εξαγάγουμε τις συναρτήσεις Boole από τον πίνακα αληθείας οι οποίες είναι οι ακόλουθες.

$$S = x'y + xy' = x \oplus y$$

$$C = xy$$

Όπως έχουμε δει, η συνάρτηση $x'y + xy'$ υλοποιείται με την πύλη XOR. Το λογικό διάγραμμα της υλοποίησης αυτής με πύλες NAND και XOR φαίνεται στην επόμενη Εικόνα.



Εικόνα: Ημιαθροιστής

Πλήρης Αθροιστής

Ο πλήρης αθροιστής είναι ένα συνδυαστικό κύκλωμα που υπολογίζει το άθροισμα τριών δυαδικών ψηφίων εισόδου. Έχει τρεις εισόδους και δύο εξόδους. Οι δύο από τις μεταβλητές εισόδου αναπαριστούν τα δύο σημαντικά ψηφία που προστίθενται και η τρίτη είσοδος παριστάνει το κρατούμενο από τη λιγότερο σημαντική βαθμίδα. Ο πίνακας αληθείας του πλήρους αθροιστή δίνεται στον επόμενο Πίνακα.

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

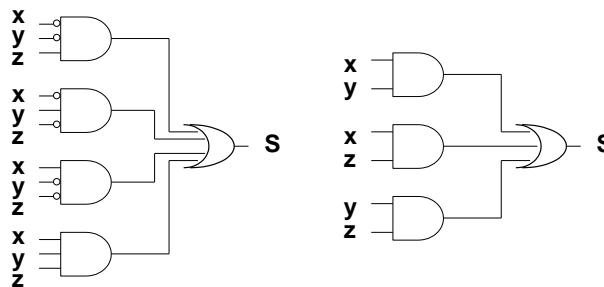
Πίνακας: Πίνακας αληθείας πλήρους αθροιστή

Μπορούμε, χρησιμοποιώντας τον πίνακα αληθείας, να εξάγουμε τις ακόλουθες συναρτήσεις για τις εξόδους S και C του πλήρους αθροιστή.

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

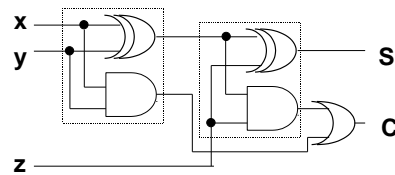
$$C = xy + xz + yz$$

Έτσι, μπορούμε να δώσουμε την υλοποίηση της επόμενης Εικόνας.



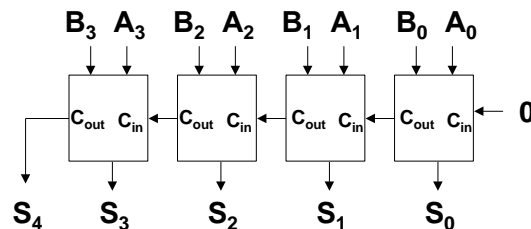
Εικόνα: Πλήρης αθροιστής

Είναι αρκετά ενδιαφέρον να παρατηρήσει κανείς ότι μια ισοδύναμη υλοποίηση του πλήρους αθροιστή μπορεί να δοθεί με τη βοήθεια ημιαθροιστών, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Πλήρους αθροιστής υλοποιημένος με ημιαθροιστές

Προκειμένου να προσθέσουμε δυαδικούς αριθμούς που αποτελούνται από περισσότερα του ενός ψηφία, χρησιμοποιούμε περισσότερους από έναν πλήρεις αθροιστές. Κάθε πλήρης αθροιστής αντιστοιχεί σε ένα ψηφίο αθροίσματος, και το κρατούμενο εξόδου κάθε αθροιστή συνδέεται στο κρατούμενο εισόδου του αθροιστή που αντιστοιχεί στην επόμενη βαθμίδα. Για παράδειγμα, στην επόμενη Εικόνα παρουσιάζεται ένα τέτοιος αθροιστής τεσσάρων βαθμίδων. Ο αθροιστής αυτός προσθέτει τους τετραψήφιους δυαδικούς αριθμούς $A_3A_2A_1A_0$ και $B_3B_2B_1B_0$ και δίνει σαν αποτέλεσμα τον πενταψήφιο δυαδικό αριθμό $S_4S_3S_2S_1S_0$.



Εικόνα: Αθροιστής 4 δυαδικών ψηφίων

2.12.2 Αφαιρέτες

Η αφαίρεση δύο δυαδικών αριθμών πραγματοποιείται παίρνοντας το συμπλήρωμα του αφαιρέτη και προσθέτοντάς το στον αφαιρετέο. Με αυτή την παρατήρηση, η πράξη της αφαίρεσης μετατρέπεται σε πρόσθεση και για την υλοποίησή της μπορούν να χρησιμοποιηθούν πλήρεις αθροιστές. Εναλλακτικά, μπορούμε να υλοποιήσουμε την αφαίρεση με άμεσο τρόπο (όπως την κάνουμε με χαρτί και μολύβι). Με αυτή τη μέθοδο, κάθε δυαδικό ψηφίο του αφαιρέτη αφαιρείται από το αντίστοιχης σημαντικότητας δυαδικό ψηφίο του αφαιρετέου και δίνει ένα δυαδικό ψηφίο διαφοράς. Αν το δυαδικό ψηφίο του αφαιρέτη είναι μεγαλύτερο από το ψηφίο του αφαιρετέου, δανειζόμαστε 1 από την επόμενη σημαντική θέση. Για να μεταβιβαστεί το γεγονός αυτό στην επόμενη βαθμίδα χρησιμοποιείται ένα σήμα δανεισμού (borrow). Όπως ακριβώς υπάρχουν ημιαθροιστές και πλήρεις αθροιστές, υπάρχουν ημιαφαιρέτες και πλήρεις αφαιρέτες.

Ημιαφαιρέτης

Ο ημιαφαιρέτης είναι ένα συνδυαστικό κύκλωμα το οποίο αφαιρεί δυο δυαδικά ψηφία και δίνει τη διαφορά τους. Έχει επίσης μία έξοδο που καθορίζει αν χρειάζεται να δανειστούμε (borrow) μια μονάδα. Αν συμβολίσουμε το ψηφίο του αφαιρετέου με x και του αφαιρέτη με y , για να κάνουμε την αφαίρεση $x - y$ πρέπει να ελέγξουμε τα σχετικά μεγέθη των x και y . Αν $x \geq y$ τότε υπάρχουν τρεις περιπτώσεις: $0 - 0 = 0$, $1 - 0 = 1$, $1 - 1 = 0$. Το αποτέλεσμα λέγεται δυαδικό ψηφίο διαφοράς. Αν $x < y$, τότε έχουμε $0 - 1$, και έτσι χρειάζεται να δανειστούμε ένα 1 από την επόμενη βαθμίδα. Το 1 που δανειζόμαστε από την επόμενη

θέση προσθέτει 2 στο δυαδικό ψηφίο του αφαιρέτη, όπως στο δεκαδικό σύστημα το κρατούμενο της αφαίρεσης προσθέτει 10 στον αφαιρετέο. Έτσι, με τον αφαιρετέο ίσο με 2, η διαφορά γίνεται $2-1=1$. Ο ημιαφαιρέτης χρειάζεται δύο εξόδους. Η μια έξοδος παράγει τη διαφορά και θα συμβολίζεται με D (Difference, διαφορά). Η δεύτερη έξοδος που θα συμβολίζεται με B (Borrow, δανεικό) παράγει το δυαδικό σήμα που πληροφορεί την επόμενη βαθμίδα ότι δανειστήκαμε μια μονάδα. Ο πίνακας αληθείας του ημιαφαιρέτη είναι ο ακόλουθος:

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Πίνακας: Πίνακας αληθείας ημιαφαιρέτη

Οι συναρτήσεις Boole για τις εξόδους του ημιαφαιρέτη βρίσκονται από τον πίνακα αληθείας και είναι:

$$D = x'y + xy'$$

$$B = xy'$$

Πλήρης αφαιρέτης

Ο πλήρης αφαιρέτης είναι ένα συνδυαστικό κύκλωμα που εκτελεί την αφαίρεση μεταξύ δύο δυαδικών ψηφίων παίρνοντας υπόψη ότι μπορεί η λιγότερο σημαντική βαθμίδα να έχει δανειστεί μια μονάδα. Αυτό το κύκλωμα έχει τρεις εισόδους και δύο εξόδους. Οι τρεις εισοδοί x,y,z συμβολίζουν το ψηφίο του αφαιρετέου, του αφαιρέτη και του κρατουμένου, αντίστοιχα. Οι εξοδοί B, D συμβολίζουν τη διαφορά και το κρατούμενο εξόδου, αντίστοιχα. Ο πίνακας αληθείας του κυκλώματος είναι ο εξής:

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Πίνακας: Πίνακας αληθείας πλήρους αφαιρέτη

Οι απλοποιημένες συναρτήσεις Boole των δύο εξόδων του πλήρους αφαιρέτη είναι οι ακόλουθες:

$$D = x'y'z + x'y'z' + xy'z' + xyz$$

$$B = x'y + x'z + yz$$

2.12.3 Αθροιστής με πρόβλεψη κρατουμένου (carry-look ahead adder)

Ο αθροιστής 4 ψηφίων που παρουσιάστηκε ονομάζεται αθροιστής ριπής (ripple carry adder), επειδή για να υπολογιστεί το κρατούμενο cout μιας βαθμίδας πρέπει να υπολογιστούν τα κρατούμενα των προηγούμενων (χαμηλότερης τάξης) βαθμίδων. Για παράδειγμα, για να υπολογιστεί το κρατούμενο εξόδου της 3^{ης} βαθμίδας, πρέπει να υπολογιστούν τα κρατούμενα εξόδου της βαθμίδας 0, της βαθμίδας 1, και της βαθμίδας 2. Αυτό έχει ως αποτέλεσμα αν θελήσουμε να υλοποιήσουμε αθροιστές 32 ή 64 βαθμίδων να είναι πολύ πιο αργοί σε λειτουργία από ένα αθροιστή 4 βαθμίδων. Επειδή η πρόσθεση είναι μια πολύ συχνή πράξη, ένας υπολογιστής των 32 ή των 64 bit θα καθυστερούσε υπερβολικά, ανεξάρτητα του πόσο γρήγορες είναι οι άλλες μονάδες του.

Στην πράξη, για να ελαττώσουμε την καθυστέρηση στην αλυσίδα διάδοσης του κρατουμένου χρησιμοποιείται μια τεχνική που ονομάζεται *πρόβλεψη κρατουμένου (carry-look ahead)*. Η τεχνική αυτή στηρίζεται στον υπολογισμό του κρατουμένου μιας βαθμίδας, χωρίς να υπολογιστεί το κρατούμενο της προηγούμενης βαθμίδας.

Έτσι για παράδειγμα, ας υποθέσουμε ότι έχουμε ένα αθροιστή 4 bit και θέλουμε να προσθέσουμε τους αριθμούς $X=x_3x_2x_1x_0$ και $Y=y_3y_2y_1y_0$. Στον πλήρη αθροιστή η πρόσθεση του ζεύγους x_i, y_i υλοποιείται από τις συναρτήσεις άθροισμα (s) και κρατούμενο (cout)

$$s_i = x_i \oplus y_i \oplus s_{i-1}$$

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

Από τις σχέσεις αυτές παρατηρούμε ότι, εάν ορίσουμε τις συναρτήσεις *γεννήτρια κρατουμένου (gi)* και *διαδότηρια κρατουμένου (pi)*

$$g_i = x_i \cdot y_i$$

$$p_i = x_i \oplus y_i$$

τότε τα διαδοχικά αθροίσματα s και κρατούμενα c στην πρόσθεση $X+Y$ μπορούν να υπολογισθούν από τις σχέσεις:

$$s_0 = p_0 \oplus c_{-1} \quad c_0 = g_0 + p_0 \cdot c_{-1}$$

$$s_1 = p_1 \oplus c_0 \quad c_1 = g_1 + p_1 \cdot c_0$$

$$s_2 = p_2 \oplus c_1 \quad c_2 = g_2 + p_2 \cdot c_1$$

$$s_3 = p_3 \oplus c_2 \quad c_3 = g_3 + p_3 \cdot c_2$$

Αν στις σχέσεις των κρατουμένων c αντικαταστήσουμε τα προηγούμενα κρατούμενα με τις συναρτήσεις τους, προκύπτουν οι σχέσεις:

$$c_0 = g_0 + p_0 \cdot c_{-1}$$

$$c_1 = g_1 + p_1 \cdot g_0 + p_1 \cdot p_0 \cdot c_{-1}$$

$$c_2 = g_2 + p_2 \cdot g_1 + p_2 \cdot p_1 \cdot g_0 + p_2 \cdot p_1 \cdot p_0 \cdot c_{-1}$$

$$c_3 = g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0 + p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot c_{-1}$$

Οι σχέσεις εξαρτώνται μόνο από το αρχικό κρατούμενο c_{-1} . Όλα τα κρατούμενα c_i μπορούν να υπολογισθούν κατευθείαν από τα bits x_i, y_i των προσθετέων αριθμών και το αρχικό κρατούμενο c_{-1} . Η καθυστέρηση είναι η ίδια για όλα τα c_i και ίση με *τρεις μονάδες χρόνου*, δηλ. μία μονάδα για κάθε συνάρτηση g_i, p_i , μία μονάδα για κάθε γινόμενο και μία για το άθροισμα, εάν υποθέσουμε ότι κάθε πύλη

καθυστερεί το σήμα μία μονάδα χρόνου. Συνεπώς και όλα τα αθροίσματα s_i υπολογίζονται σε τέσσερις μονάδες χρόνου το καθένα. Με τη μέθοδο αυτή, χρησιμοποιώντας το πρόσθετο κύκλωμα που απαιτείται, όλα τα bits s_i ($i=0, \dots, n-1$) του αθροίσματος υπολογίζονται σε σταθερό χρόνο, ανεξάρτητο του n .

Η μέθοδος της πρόβλεψης κρατουμένου, στην πράξη περιορίζεται συνήθως στα 4 ή το πολύ 8 bits λόγω του ότι απαιτεί πύλες με πολλές εισόδους. Από τις προηγούμενες σχέσεις βλέπουμε ότι κάθε συνάρτηση κρατουμένου ανωτέρας τάξεως c_{i+1} απαιτεί πύλες AND και OR με μία είσοδο επιπλέον από ό,τι στην προηγούμενη συνάρτηση c_i . Για ένα κύκλωμα πρόβλεψης n κρατουμένων απαιτούνται πύλες με n εισόδους. Το πλήθος των εισόδων όμως των πυλών που χρησιμοποιούνται στην πράξη συνήθως δεν ξεπερνά το 4. Για το λόγο αυτό τα κυκλώματα πρόβλεψης κρατουμένου δεν ξεπερνούν συνήθως τα 4 bit. Το c_3 μπορεί να γραφεί ως:

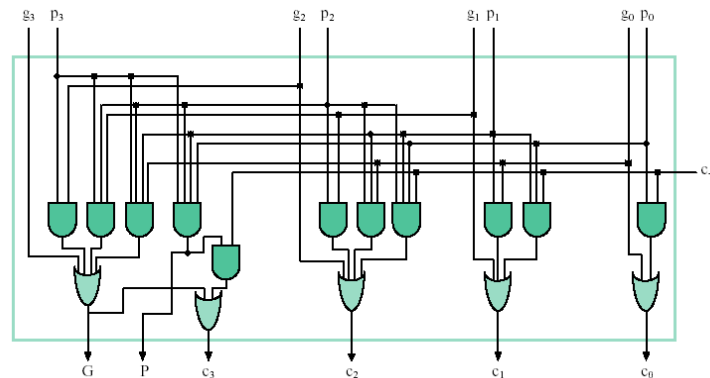
$$c_3 = G + P \cdot c_1$$

όπου:

$$G = g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0$$

$$P = p_3 \cdot p_2 \cdot p_1 \cdot p_0$$

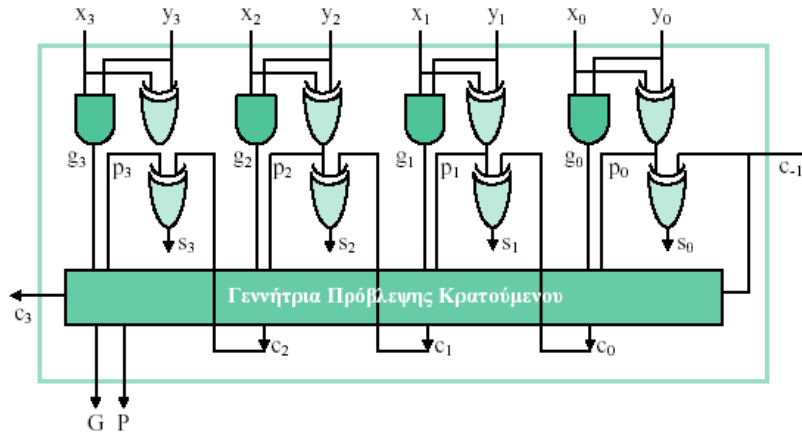
Οι σχέσεις αυτές ορίζουν τις λογικές συναρτήσεις ενός συνδυαστικού κυκλώματος που ονομάζεται *γεννήτρια προβλέψεως κρατουμένου* και φαίνεται στην επόμενη Εικόνα. Οι συναρτήσεις G και P ονομάζονται αντίστοιχα *γεννήτρια κρατουμένου* (Generate, G) και *διαδότρια κρατουμένου* (Propagate, P) για ένα πακέτο 4 bit και μπορούν να χρησιμοποιηθούν σε μία γεννήτρια κρατουμένου μεγαλύτερης τάξεως, όπως θα δούμε στη συνέχεια.



Εικόνα: Γεννήτρια πρόβλεψης κρατουμένου

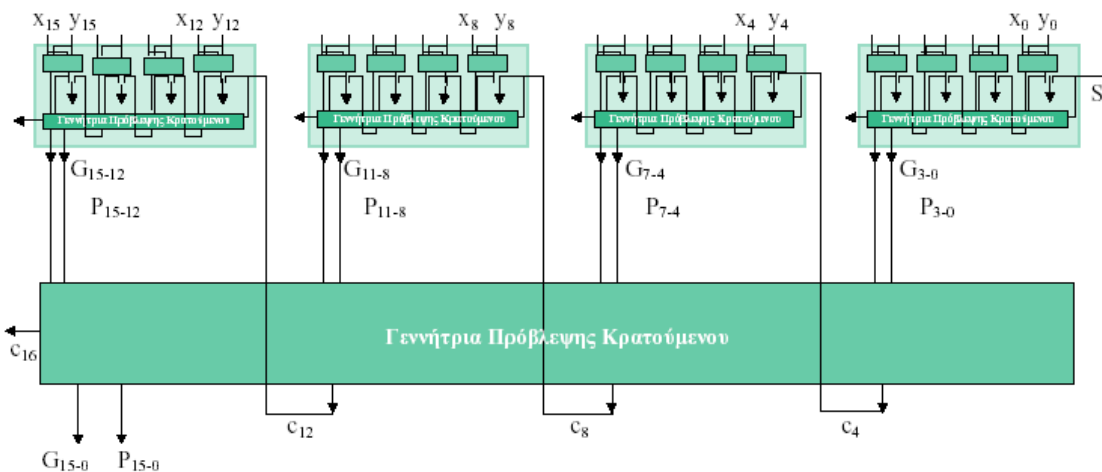
Κύκλωμα γεννήτριας

Η γεννήτρια κρατουμένου μπορεί να συμπληρώσει τον αθροιστή ριπής επιταχύνοντας έτσι την πράξη της άθροισης. Συνδυάζοντας ένα κύκλωμα γεννήτριας κρατουμένου 4 bit με τα αντίστοιχα κυκλώματα που υλοποιούν τα αθροίσματα s μπορεί να κατασκευασθεί ένα κύκλωμα ταχείας αθροίσεως, που ονομάζεται *αθροιστής με πρόβλεψη κρατουμένου* και φαίνεται στην επόμενη Εικόνα.



Εικόνα: Αθροιστής 4-bit με πρόβλεψη κρατούμενου

Ο περιορισμός των 4-bit στον αθροιστή με πρόβλεψη κρατούμενου μπορεί να παρακαμφθεί εάν τοποθετήσουμε περισσότερα επίπεδα πρόβλεψης κρατούμενου, όπως εξηγείται στη συνέχεια για την περίπτωση του αθροιστή 16-bit.



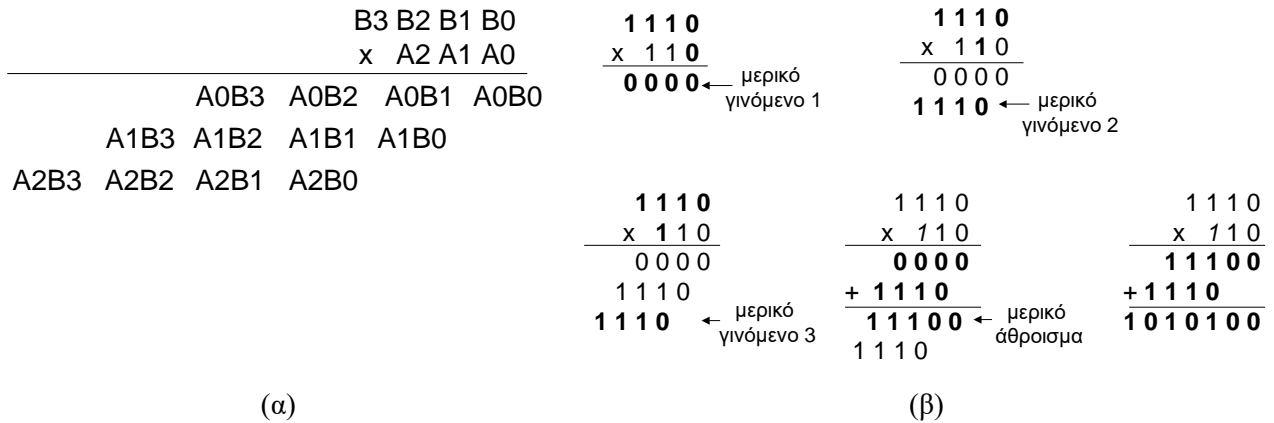
Εικόνα: Αθροιστής 16-bit με δύο επίπεδα πρόβλεψης κρατούμενου

Για να κατασκευάσουμε έναν αθροιστή των 16 bit με πρόβλεψη κρατούμενου μπορούμε να χρησιμοποιήσουμε πρόβλεψη δευτέρου επιπέδου, όπως φαίνεται στην προηγούμενη Εικόνα για την περίπτωση ενός αθροιστή των 16 bit. Ο αθροιστής αυτός αποτελείται από τέσσερις αθροιστές, των 4 bit ο καθένας με πρόβλεψη κρατούμενου, οι συναρτήσεις G_{3-0} , P_{3-0} , ..., G_{15-12} , P_{15-12} των οποίων τροφοδοτούν μία γεννήτρια πρόβλεψης κρατούμενου ανωτέρας τάξεως. Η γεννήτρια παράγει τα κρατούμενα c_4 , c_8 , c_{12} , c_{16} για τους επόμενους αθροιστές αντίστοιχα. Η καθυστέρηση για την άθροιση των 16-bit είναι $1 + 2 + 2 + 2 + 1 = 8$ μονάδες χρόνου, ενώ ένας κυματικός αθροιστής θα χρειαζόταν $2 \times 16 = 32$ μονάδες χρόνου.

Παρατηρούμε ότι σε 5 μονάδες χρόνου παράγονται και οι συναρτήσεις G_{15-0} και P_{15-0} που αντιπροσωπεύουν το σύνολο των 16 bit. Επομένως, συνδυάζοντας τέσσερις αθροιστές των 4 bit ο καθένας, με μία επιπλέον γεννήτρια μπορούμε να κατασκευάσουμε έναν αθροιστή των 64 bit που να έχει καθυστέρηση $5 + 2 + 2 + 2 + 1 = 12$ μονάδες χρόνου αντί των 128. Γενικά, ένας αθροιστής των n bit με πρόβλεψη κρατούμενου ανά 4 bit θα έχει περίπου $\log_4(n)$ επίπεδα και θα παρουσιάζει καθυστέρηση $4 \times \log_4(n)$ μονάδες χρόνου, ενώ η καθυστέρηση ενός αθροιστή ριπής είναι $2n$ μονάδες χρόνου.

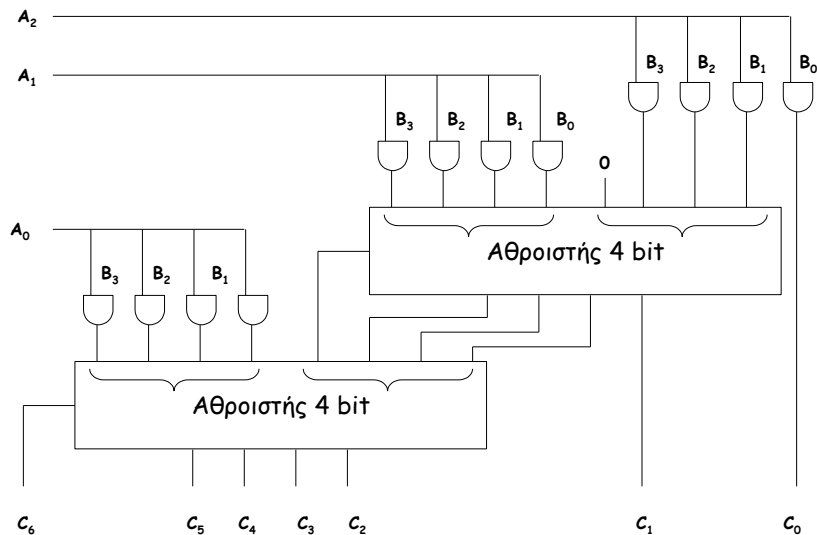
2.12.4 Πολλαπλασιαστής

Όπως έχει αναφερθεί, ο πολλαπλασιασμός στο δυαδικό σύστημα γίνεται όπως και στο δεκαδικό. Πιο συγκεκριμένα, πολλαπλασιάζουμε κάθε (δυαδικό) ψηφίο του πολλαπλασιαστή με κάθε ψηφίο του πολλαπλασιαστέου, και προσθέτουμε τα μερικά γινόμενα. Στην ακόλουθη Εικόνα (α) παρουσιάζεται σχηματικά η πράξη του πολλαπλασιασμού για δύο αριθμούς που αποτελούνται από 4 και 3 δυαδικά ψηφία αντίστοιχα, ενώ στο (β) παρουσιάζεται η διαδικασία υλοποίησης της πράξης για τους δυαδικούς αριθμούς 1110 και 110.



Εικόνα: (α) Εκτέλεση πολλαπλασιασμού 4×3 (β) Παράδειγμα εκτέλεσης

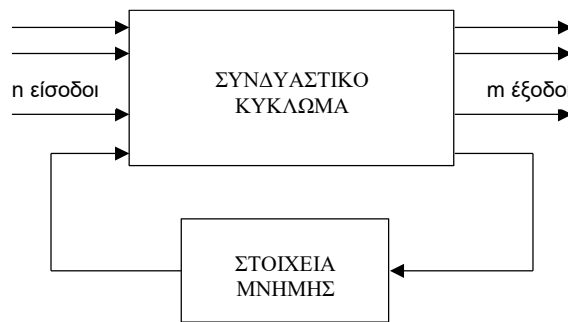
Ο τρόπος με τον οποίο υλοποιείται ο πολλαπλασιασμός οδηγεί σε μία προφανή υλοποίηση σε υλικό, σύμφωνα με την οποία σχηματίζουμε όλα τα μερικά γινόμενα, και στη συνέχεια προσθέτουμε τα μερικά αθροίσματα. Για την παραγωγή των μερικών γινομένων χρησιμοποιούνται ισάριθμες πύλες AND δύο εισόδων, ενώ για την παραγωγή των αθροισμάτων χρησιμοποιούνται πλήρεις αθροιστές. Στην επόμενη Εικόνα παρουσιάζεται το κύκλωμα που υλοποιεί την πράξη του πολλαπλασιασμού για αριθμούς 4×3 bits.



Εικόνα: Πολλαπλασιαστής 4×3 bit

2.13 Βασικές ακολουθιακές μονάδες

Τα ακολουθιακά κυκλώματα αποτελούνται από συνδυαστικά κυκλώματα και στοιχεία μνήμης, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Ακολουθιακό Κύκλωμα

Τα στοιχεία μνήμης μπορούν να αποθηκεύσουν δυαδικές πληροφορίες που αποτελούν την παρούσα κατάσταση του στοιχείου μνήμης (state) κάθε χρονική στιγμή. Οι έξοδοι και η επόμενη κατάσταση των στοιχείων μνήμης ενός Ακολουθιακού Κυκλώματος είναι συναρτήσεις των εισόδων και της παρούσας κατάστασης των στοιχείων μνήμης του ακολουθιακού κυκλώματος. Τα ακολουθιακά κυκλώματα διακρίνονται σε *σύγχρονα* (synchronous) και *ασύγχρονα* (asynchronous).

Σε ένα ασύγχρονο ακολουθιακό κύκλωμα τα στοιχεία μνήμης είναι κατάλληλα διασυνδεδεμένες λογικές πύλες που ονομάζονται *μανδαλωτές* (latches). Κάποιες από τις εξόδους του συνδυαστικού κυκλώματος που περιέχεται σε ένα ακολουθιακό κύκλωμα συνδέονται με τα στοιχεία μνήμης, οι εξόδους των οποίων τροφοδοτούν κάποιες εισόδους του συνδυαστικού κυκλώματος (βρόγχος ανάδρασης, feedback).

Σε ένα σύγχρονο ακολουθιακό κύκλωμα τα στοιχεία μνήμης είναι flip-flops. Το flip-flop είναι μια διάταξη που μπορεί να διατηρηθεί σε μία κατάσταση έως ότου κάποιο σήμα εισόδου (ρολόι) το κάνει να αλλάξει κατάσταση. Στη συνέχεια θα περιγράψουμε τους μανδαλωτές (latches) και τα flipflops.

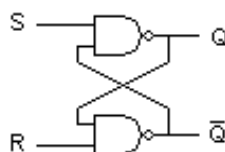
2.13.1 Μανδαλωτές

Ο μανταλωτής (latch) έχει δύο εισόδους: S (Set - θέση) και R (Reset - επαναφορά), καθώς και δύο εξόδους: Q (έξοδος), και Q' (συμπλήρωμα της εξόδου).

Το βασικό κύκλωμα ενός μανδαλωτή μπορεί να υλοποιηθεί με δύο πύλες NAND ή με δύο πύλες NOR. Η έξοδος κάθε πύλης συνδέεται χιαστί με την είσοδο της άλλης πύλης δημιουργώντας ένα βρόγχο ανάδρασης (feedback). Ο μανδαλωτής αυτός ονομάζεται μανταλωτής SR (SR latch).

Μανδαλωτής S-R με πύλες NAND

Το κύκλωμα του μανταλωτή (latch) μπορεί να υλοποιηθεί με δύο πύλες NAND όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Μανταλωτής (latch) με πύλες NAND

Ως κατάσταση του μανδαλωτή θεωρούμε την τιμή της εξόδου Q. Οι (χρήσιμες) καταστάσεις στις οποίες μπορεί να βρεθεί το flip-flop είναι η κατάσταση θέσης (set) όπου $Q=1$ και $Q'=0$ και κατάσταση επαναφοράς (reset) ή μηδενισμού (clear) όπου $Q=0$ και $Q'=1$.

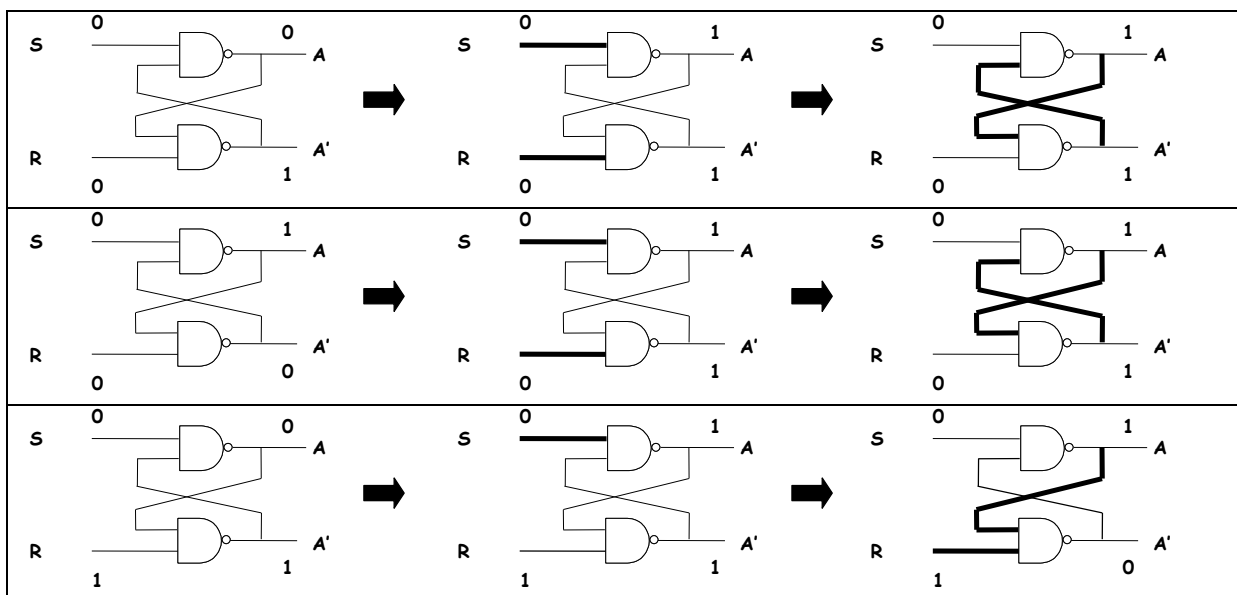
Ο μανδαλωτής με πύλες NAND λειτουργεί ως εξής:

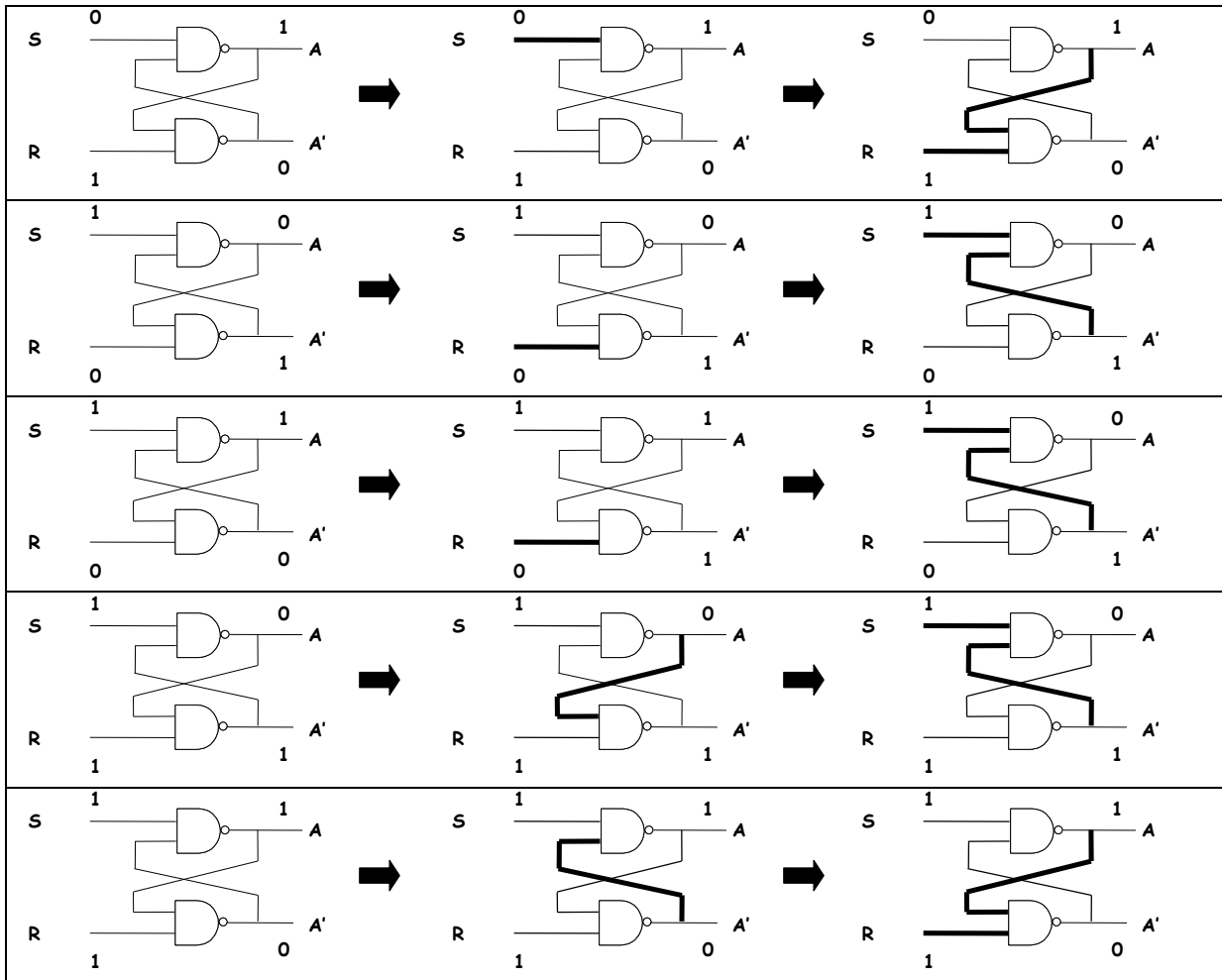
1. $S=1$ και $R=1$ (κατάσταση ηρεμίας). Στην κατάσταση αυτή η έξοδος παραμένει αμετάβλητη (οι έξοδοι διατηρούν τις τιμές που είχαν πριν τεθεί στις εισόδους $S=1$ και $R=1$).
2. $S=0$ και $R=1$ (ενεργοποίηση). Στην κατάσταση αυτή η έξοδος είναι $Q=1$ (θέση) και παραμένει $Q=1$.
3. $S=1$ και $R=0$ (μηδενισμός). Η έξοδος είναι $Q=0$ (μηδενισμός).
4. $S=0$ και $R=0$ (μη χρησιμοποιούμενη κατάσταση). Οι έξοδοι είναι $Q=1$ και $Q'=1$. Στην πράξη αυτή η κατάσταση δεν χρησιμοποιείται.

		Προηγούμενη κατάσταση		Νέα κατάσταση	
S	R	Q	Q'	Q	Q'
0	0	0	1	1	1
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	0	1
1	1	1	0	1	0

Πίνακας: Πίνακας μεταβάσεων του μανδαλωτή S-R με πύλες NAND

Στην επόμενη Εικόνα παρουσιάζεται η διαδικασία μεταβάσεων για κάθε περίπτωση του ανωτέρω Πίνακα.





Εικόνα: Μεταβάσεις των σημάτων για μανδαλωτή με πύλες NAND

Αν ο μανδαλωτής είναι σε κατάσταση θέσης ($S=0$ και $R=1$ με $Q=1$) και εφαρμοστεί $S=1$, τότε παραμένει σε κατάσταση θέσης ($Q=1$).

Αν το flip-flop είναι σε κατάσταση μηδενισμού ($S=1$ και $R=0$ με $Q=0$) και εφαρμοστεί $R=1$, τότε το flip-flop παραμένει σε κατάσταση μηδενισμού ($Q=0$).

Στις δύο παραπάνω περιπτώσεις οι έξοδοι παραμένουν *αμετάβλητες* (ο μανδαλωτής έχει μνήμη).

Αν πρέπει να αλλάξει η κατάσταση του μανταλωτή, τότε:

- Αν το flip-flop είναι σε κατάσταση θέσης ($S=0$ και $R=1$ με $Q=1$), τότε πρώτα $S=1$, οπότε το flip-flop πάει σε κατάσταση ηρεμίας, και μετά $R=0$, οπότε το flip-flop πάει σε κατάσταση μηδενισμού ($Q=0$).

- Αν το flip-flop είναι σε κατάσταση μηδενισμού ($S=1$ και $R=0$ με $Q=0$), τότε πρώτα $R=1$ οπότε το flip-flop πάει σε κατάσταση ηρεμίας και μετά $S=0$, οπότε το flip-flop πάει σε κατάσταση θέσης ($Q=1$).

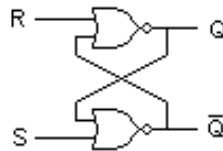
Στον ερ;ομενο Πίνακα παρουσιάζεται ο πίνακας αληθείας του μανδαλωτή με πύλες NAND, όπου συνοψίζεται η λειτουργία του.

S	R	Q	Q'		
0	0	1	1	Μη χρησιμοποιούμενη	Μη χρησιμοποιούμενη
0	1	1	0	Q=1	Θέση
1	0	0	1	Q=0	Μηδενισμός
1	1	0	1	Μετά από S=1 και R=0	Αμετάβλητη
1	1	1	0	Μετά από S=0 και R=1	Αμετάβλητη

Πίνακας: Πίνακας αληθείας μανδαλωτή με πύλες NAND (Η έξοδος Q ακολουθεί την είσοδο R όταν $S \neq R$)

Μανδαλωτής με πύλες NOR

Εναλλακτικά με την προηγούμενη υλοποίηση, το κύκλωμα του μανδαλωτή μπορεί να υλοποιηθεί με δύο πύλες NOR όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Μανδαλωτής (latch) με πύλες NOR

Η κατάσταση του flip-flop είναι η τιμή της εξόδου Q. Οι (χρήσιμες) καταστάσεις, στις οποίες μπορεί να βρεθεί το flip-flop είναι η κατάσταση θέσης (set) όπου $Q=1$ και $Q'=0$ και η κατάσταση επαναφοράς (reset) ή μηδενισμού (clear) όπου $Q=0$ και $Q'=1$.

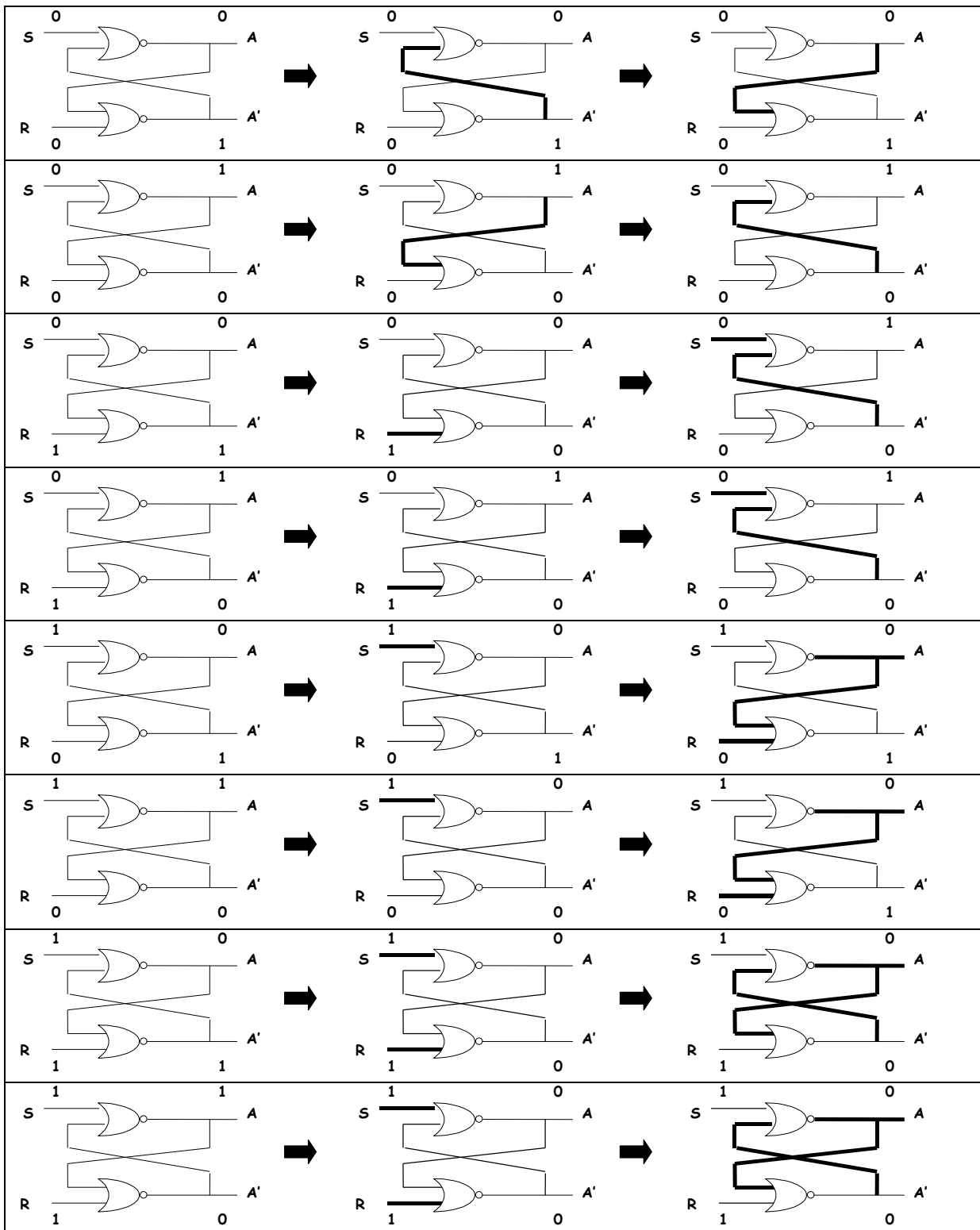
Η λειτουργία του μανταλωτή με πύλες NOR μπορεί να περιγραφεί ως εξής:

1. $S=0$ και $R=0$ (κατάσταση ηρεμίας). Στην κατάσταση αυτή η έξοδος παραμένει αμετάβλητη (οι εξοδοί διατηρούν τις τιμές που είχαν πριν τεθεί στις εισόδους $S=0$ και $R=0$).
2. $S=0$ και $R=1$ (μηδενισμός). Η έξοδος είναι $Q=0$ και η έξοδος $Q'=1$.
3. $S=1$ και $R=0$ (ενεργοποίηση). Η έξοδος είναι $Q=1$ και η έξοδος $Q'=0$.
4. $S=1$ και $R=1$ (μη χρησιμοποιούμενη κατάσταση). Αμφότερες οι εξοδοί είναι $Q=0$ και $Q'=0$. Η κατάσταση αυτή δε χρησιμοποιείται στην πράξη.

		Προηγούμενη κατάσταση		Νέα κατάσταση	
S	R	Q	Q'	Q	Q'
0	0	0	1	0	1
0	0	1	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	0	0	0

Πίνακας: Πίνακας μεταβάσεων του μανδαλωτή S-R με πύλες NOR

Στην επόμενη Εικόνα παρουσιάζεται η διαδικασία μεταβάσεων για κάθε περίπτωση του ανωτέρω Πίνακα.



Εικόνα: Μεταβάσεις για μανδαλωτή με πύλες NOR

Αν ο μανδαλωτής είναι σε κατάσταση θέσης (S=1 και R=0 με Q=1) και εφαρμοστεί S=0, τότε το flip-flop παραμένει σε κατάσταση θέσης (Q=1).

Αν ο μανδαλωτής είναι σε κατάσταση μηδενισμού ($S=0$ και $R=1$ με $Q=0$) και εφαρμοστεί $R=0$, τότε το flip-flop παραμένει σε κατάσταση μηδενισμού ($Q=0$).

Στις δύο παραπάνω περιπτώσεις οι εξόδοι παραμένουν *αμετάβλητες* (ο μανδαλωτής έχει μνήμη).

Αν επιθυμούμε να *αλλάξει* η κατάσταση του μανδαλωτή, τότε:

- Αν είναι σε κατάσταση θέσης ($S=1$ και $R=0$ με $Q=1$), τότε πρώτα $S=0$ οπότε πάει σε κατάσταση ηρεμίας και μετά $R=1$, οπότε πάει σε κατάσταση μηδενισμού ($Q=0$).

- Αν είναι σε κατάσταση μηδενισμού ($S=0$ και $R=1$ με $Q=0$), τότε πρώτα $R=0$ οπότε πάει σε κατάσταση ηρεμίας και μετά $S=1$, οπότε πάει σε κατάσταση θέσης ($Q=1$).

Στη συνέχεια παρουσιάζεται ο Πίνακας Αληθείας του μανδαλωτή με πύλες NOR, όπου συνοψίζεται η λειτουργία του.

S	R	Q	Q'		
0	0	0	1	Μετά από $S=0$ και $R=1$	Αμετάβλητη
0	0	1	0	Μετά από $S=1$ και $R=0$	Αμετάβλητη
0	1	0	1	$Q=0$	Μηδενισμός
1	0	1	0	$Q=1$	Θέση
1	1	0	0	Μη χρησιμοποιούμενη	Μη χρησιμοποιούμενη

Πίνακας: Πίνακας Αληθείας Μανταλωτή με πύλες NOR (Η έξοδος Q ακολουθεί την είσοδο S όταν $S \neq R$)

2.13.2 Μανδαλωτές με είσοδο ελέγχου

Στους μανδαλωτές που είδαμε μέχρι τώρα, αν οποιαδήποτε από τις δύο εισόδους τους αλλάξει τιμή, η μεταβολή μεταφέρεται αμέσως στις εξόδους. Οι μονάδες που θα παρουσιάσουμε στην παράγραφο αυτή, διαθέτουν μία επιπλέον είσοδο, στην οποία αναφερόμαστε με το όνομα CP (Control Pulse). Όταν η είσοδος αυτή βρίσκεται στο λογικό '0', οποιαδήποτε αλλαγή στις άλλες εισόδους δεν επηρεάζει την έξοδο. Αντιθέτως, όταν η είσοδος CP έχει τιμή '1', οι αλλαγές στις εισόδους επηρεάζουν τις εξόδους. Οι πιο συχνά χρησιμοποιούμενοι μανδαλωτές με είσοδο ελέγχου είναι οι R-S, D, J-K, T, και περιγράφονται στη συνέχεια.

Μανδαλωτής R-S με είσοδο ελέγχου

Ο Μανδαλωτής R-S με είσοδο ελέγχου μπορεί να υλοποιηθεί με τέσσερις πύλες NAND, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Μανδαλωτής R-S με είσοδο ελέγχου και γραφικό σύμβολο

Η λειτουργία του μανδαλωτή R-S με είσοδο ελέγχου όταν η είσοδος CP είναι στην τιμή '1', περιγράφεται ως εξής:

- ♦ Όταν $S=0$ και $R=0$, τότε η επόμενη κατάσταση (έξοδος Q) είναι ίδια με την προηγούμενη κατάσταση.

- ♦ Όταν $S=0$ και $R=1$, τότε η επόμενη κατάσταση είναι $Q=0$.
- ♦ Όταν $S=1$ και $R=0$, τότε η επόμενη κατάσταση είναι $Q=1$.
- ♦ Όταν $S=1$ και $R=1$, τότε η επόμενη κατάσταση είναι απροσδιόριστη. Αυτή είναι μη χρησιμοποιούμενη κατάσταση.

Όταν η είσοδος CP έχει τιμή 0, τότε οι έξοδοι των αριστερών πυλών NAND (τις οποίες οδηγεί η είσοδος CP) είναι αμφότερες '1' επομένως ο μανδαλωτής RS που αποτελείται από τις δεξιές πύλες NAND βρίσκεται σε κατάσταση ηρεμίας, επομένως οποιαδήποτε αλλαγή στις εισόδους δεν επηρεάζει τις εξόδους.

Ο χαρακτηριστικός πίνακας του μανδαλωτή S-R με είσοδο ελέγχου δείχνει την επόμενη κατάσταση $Q(n+1)$ όταν είναι γνωστή η παρούσα κατάσταση $Q(n)$ και οι εισόδους και παρουσιάζεται στον επόμενο Πίνακα.

Q(n)	S	R	Q(n+1)		S	R	Q(n+1)
0	0	0	0				
0	0	1	0				
0	1	0	1		0	0	Q(n)
0	1	1	X	ή	0	1	0
1	0	0	1		1	0	1
1	0	1	0		1	1	X
1	1	0	1				
1	1	1	X				

Πίνακας: Χαρακτηριστικός πίνακας μανδαλωτή S-R με είσοδο ελέγχου

Ο πίνακας διέγερσης του μανδαλωτή S-R με είσοδο ελέγχου δείχνει τον τρόπο μετάβασης από την παρούσα κατάσταση στην επόμενη κατάσταση και εξάγεται από το χαρακτηριστικό πίνακα. Ο πίνακας διέγερσης παρουσιάζεται στον επόμενο Πίνακα.

S	R	Q(n)	Q(n+1)
0	X	0	0
1	0	0	1
0	1	1	0
X	0	1	1

Πίνακας: Πίνακας Διέγερσης μανδαλωτή S-R

Μανδαλωτής D με είσοδο ελέγχου

Όπως είδαμε, ένα από τα ανεπιθύμητα χαρακτηριστικά του μανδαλωτή τύπου S-R (με ή χωρίς είσοδο ελέγχου) είναι η ανεπιθύμητη κατάσταση η οποία προκύπτει όταν και οι δύο εισόδους S, R έχουν την τιμή 1 (ή 0, ανάλογα με το αν η υλοποίηση πραγματοποιείται με πύλες NAND ή NOR). Ένας τρόπος που μπορεί να σκεφτεί κανείς προκειμένου να εξαλείψει την ανεπιθύμητη αυτή συμπεριφορά είναι να συνδέσει τις δύο εισόδους του μανδαλωτή με μία πύλη NOT. Ο τρόπος αυτός αποτελεί τη βασική ιδέα για την υλοποίηση του μανδαλωτή τύπου D.

Ο μανδαλωτής D με είσοδο ελέγχου μπορεί να υλοποιηθεί χρησιμοποιώντας ένα μανδαλωτή R-S με είσοδο ελέγχου και μία πύλη NOT, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Μανδαλωτής D με είσοδο ελέγχου και γραφικό σύμβολο

Ο μανδαλωτής D με είσοδο ελέγχου λειτουργεί ως εξής:

- ♦ Αν ο παλμός εισόδου είναι $CP=0$, τότε δε μπορεί να αλλάξει κατάσταση, ανεξάρτητα από την τιμή της εισόδου D (πράγματι, αν $CP=0$, τότε οι εισοδοί του μανταλωτή με πύλες NAND είναι "1")
- ♦ Αν ο παλμός εισόδου είναι $CP=1$, τότε αν $D=0$, τότε $Q=0$ (μηδενισμός), ενώ αν $D=1$, τότε $Q=1$ (θέση)

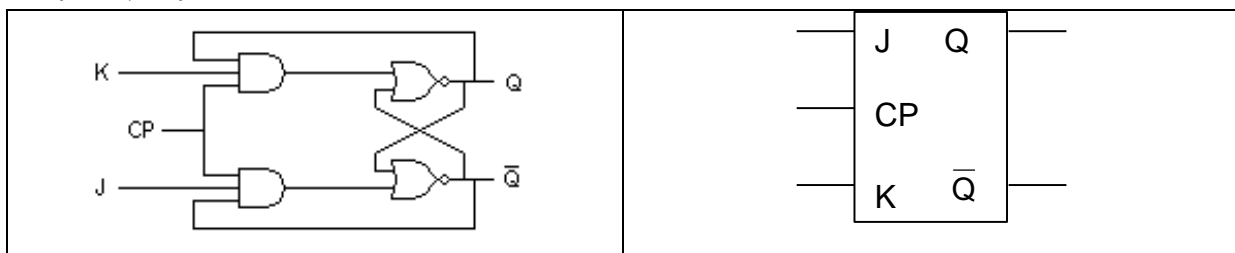
Το όνομα του μανδαλωτή D με είσοδο ελέγχου προέρχεται από την δυνατότητά του να αποθηκεύει δεδομένα (Data) και να καθυστερεί τη διάδοσή τους (Delay). Οι δυαδικές πληροφορίες της εισόδου δεδομένων D του μεταφέρονται στην έξοδο Q του, όταν $CP=1$ (η έξοδος ακολουθεί τα δεδομένα εισόδου όσο $CP=1$). Όταν τεθεί $CP=0$, τότε τα δεδομένα της εισόδου D δεν μεταφέρονται στην έξοδο Q μέχρι να τεθεί $CP=1$. Ο χαρακτηριστικός πίνακας και ο πίνακας διέγερσης του D παρουσιάζεται στον επόμενο Πίνακα.

Q(n)	D	Q(n+1)		D	Q(n+1)	D	Q(n)	Q(n+1)	
0	0	0	H	0	0	0	0	0	
0	1	1		0	1	1	1	0	1
1	0	0		1	0	0	0	1	0
1	1	1		1	1	1	1	1	1

Πίνακας: Χαρακτηριστικός πίνακας μανδαλωτή D με είσοδο ελέγχου και πίνακας διέγερσης

Μανδαλωτής J-K με είσοδο ελέγχου

Μία άλλη τεχνική για να αποφευχθεί η μη επιτρεπτή κατάσταση του μανδαλωτή R-S με είσοδο ελέγχου είναι ο μανδαλωτής τύπου J-K. Ο μανδαλωτής τύπου J-K flip-flop διαθέτει δύο εισόδους (J και K), οι οποίες όταν τεθούν στις τιμές $J=1$ και $K=1$, τότε ο μανδαλωτής αλλάζει κατάσταση (δηλαδή αν η έξοδος ήταν $Q=0$ τότε θα γίνει $Q=1$ και αντίστροφα). Ο μανδαλωτής J-K μπορεί να υλοποιηθεί όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Μανδαλωτής J-K με είσοδο ελέγχου

Η λειτουργία του είναι η ακόλουθη:

1. Όταν $J=0$ και $K=0$, τότε η επόμενη κατάσταση είναι ίδια με την προηγούμενη.
2. Όταν $J=0$ και $K=1$, τότε η επόμενη κατάσταση είναι $Q=0$.
3. Όταν $J=1$ και $K=0$, τότε η επόμενη κατάσταση είναι $Q=1$.
4. Όταν $J=1$ και $K=1$, τότε η κατάσταση του flip-flop αντιστρέφεται, δηλαδή η επόμενη κατάσταση είναι η συμπληρωματική της προηγούμενης κατάστασης.

Αξίζει να σημειωθεί ότι όταν $J=1$ και $K=1$ και $CP=1$, τότε η κατάσταση του μανδαλωτή αντιστρέφεται συνεχώς μέχρι να γίνει $CP=0$. Αυτός ο ανεπιθύμητος τρόπος λειτουργίας αποφεύγεται χρησιμοποιώντας flip flop όπως θα περιγραφεί σε επόμενη παράγραφο.

Ο χαρακτηριστικός πίνακας του μανδαλωτή τύπου J-K και ο πίνακας διέγερσης παρουσιάζονται στον πίνακα παρουσιάζεται στον επόμενο Πίνακα.

Q(n)	J	K	Q(n+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

J	K	Q(n+1)
0	0	Q(n)
0	1	0
1	0	1
1	1	Q(n)'

J	K	Q(n)	Q(n+1)
0	X	0	0
1	X	0	1
X	1	1	0
X	0	1	1

Πίνακας: Χαρακτηριστικός Πίνακας και πίνακας διέγερσης μανδαλωτή J-K

Μανδαλωτής τύπου T

Αν οι είσοδοι του μανδαλωτή τύπου J-K συνδεθούν μεταξύ τους, τότε προκύπτει ο μανδαλωτής τύπου T, ο οποίος παρουσιάζεται στην επόμενη Εικόνα.



Εικόνα: Μανδαλωτής τύπου T

Το όνομα του μανδαλωτή τύπου T προέρχεται από τη δυνατότητά του να αντιστρέφει (Toggle) την κατάστασή του και λειτουργεί ως εξής:

- ◆ Όταν $T=0$, τότε η επόμενη κατάσταση είναι ίδια με την προηγούμενη.
- ◆ Όταν $T=1$, τότε η κατάσταση του flip-flop αντιστρέφεται, δηλαδή η επόμενη κατάσταση είναι η συμπληρωματική της προηγούμενης.

Ο χαρακτηριστικός πίνακας καθώς και ο πίνακας διέγερσης παρουσιάζονται στον επόμενο Πίνακα.

Q(n)	T	Q(n+1)		T	Q(n)	Q(n+1)	
0	0	0		T	Q(n+1)	0	0
0	1	1		0	Q(n)	1	0
1	0	1	ή	1	Q(n)'	1	1
1	1	0				0	1

Πίνακας: Χαρακτηριστικός πίνακας και πίνακας διέγερσης του μανδαλωτή T

2.13.3 Flip-flops

Στα ακολουθιακά στοιχεία που είδαμε μέχρι τώρα, οι έξοδοι μεταβάλλονται άμεσα (οποιαδήποτε στιγμή μεταβληθούν οι είσοδοι). Στην περίπτωση των μανδαλωτών με είσοδο επίτρεψης, οι έξοδοι επηρεάζονται όταν η είσοδος ελέγχου έχει τιμή '1', σε κάθε περίπτωση όμως, όσο CP=1, οποιαδήποτε αλλαγή στις εισόδους μεταφέρεται στις εξόδους. Αυτό μπορεί να δημιουργήσει διάφορα προβλήματα. Για παράδειγμα, στο μανδαλωτή J-K είδαμε ότι όταν η είσοδος επίτρεψης CP=1, ο συνδυασμός J=1, K=1 έχει ως αποτέλεσμα η τιμή της εξόδου να αντιστρέφεται συνέχεια. Στην πραγματικότητα, στα σύγχρονα ψηφιακά κυκλώματα, τα οποία χρησιμοποιούνται κατά κόρον στα υπολογιστικά συστήματα, και τα οποία θα μελετήσουμε αναλυτικά στη συνέχεια, οι καταστάσεις των στοιχείων μνήμης μεταβάλλονται ταυτόχρονα σε όλα τα στοιχεία μνήμης μέσω ενός κοινού σήματος ρολογιού (clock signal, clk).

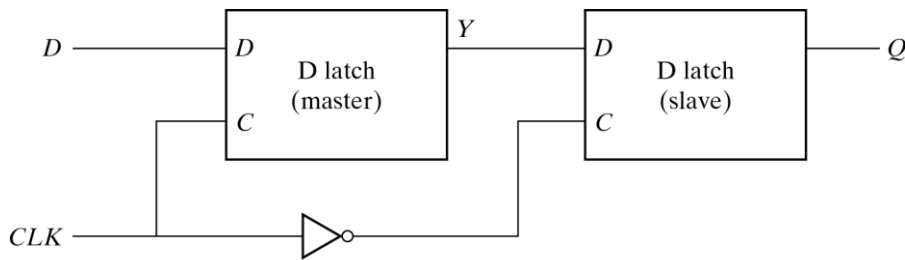
Στην παράγραφο αυτή θα μελετήσουμε μονάδες flip flops. Τα flip-flops διεγείρονται με τους παλμούς του ρολογιού (clock) τους. Οι παλμοί του ρολογιού μπορεί να είναι θετικοί ή αρνητικοί. Μία πηγή θετικών παλμών ρολογιού παραμένει στο "0" κατά το διάστημα μεταξύ παλμών και πάει στο "1" κατά τη διάρκεια του παλμού. Μία πηγή αρνητικών παλμών ρολογιού παραμένει στο "1" κατά το διάστημα μεταξύ παλμών και πάει στο "0" κατά τη διάρκεια του παλμού. Επομένως, και στις δύο περιπτώσεις, υπάρχουν δύο μεταβάσεις του σήματος για κάθε παλμό: η μετάβαση από το "0" στο "1" ονομάζεται θετική μετάβαση ή μετάβαση ανόδου ή θετική ακμή (positive edge) ή θετικό μέτωπο και η μετάβαση από το "1" στο "0" ονομάζεται αρνητική μετάβαση ή μετάβαση καθόδου ή αρνητική ακμή (negative edge) ή αρνητικό μέτωπο. Τα δύο είδη μεταβάσεων φαίνονται στην επόμενη Εικόνα.



Εικόνα: Μεταβάσεις των παλμών του ρολογιού θετική και αρνητική μετάβαση

Οι είσοδοι του flip-flop προετοιμάζουν την αλλαγή κατάστασής του, η οποία πραγματοποιείται με τη θετική ή αρνητική ακμή του παλμού του ρολογιού.

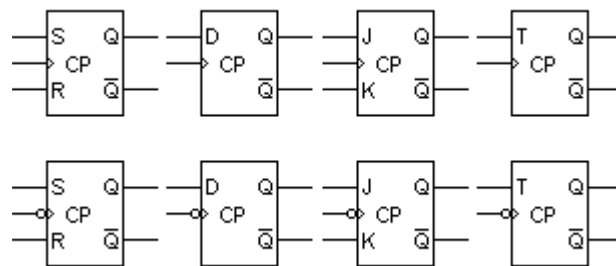
Για να υλοποιήσουμε ένα flip flop χρησιμοποιούμε κατάλληλα συνδεδεμένους μανδαλωτές. Η πιο απλή σκέψη που έχει εφαρμοστεί είναι να χρησιμοποιήσουμε δύο μανδαλωτές, στους οποίους η έξοδος του ενός θα είναι είσοδος στον άλλο, οι δε είσοδοι χρονισμού οδηγούνται από αντίθετα σήματα ρολογιού. Το flip flop αυτό ονομάζεται αφέντη-σκλάβου (master-slave). Για παράδειγμα, το flip flop αφέντη-σκλάβου τύπου D φαίνεται στην επόμενη Εικόνα.



Εικόνα: Flip flop αφέντη-σκλάβου

Στο flip flop της προηγούμενης Εικόνας, όταν το σήμα clk έχει λογικό '1', η τιμή της εισόδου περνάει στη γραμμή Y, ενώ η έξοδος Q δεν επηρεάζεται, αφού η είσοδος C του slave latch είναι στο λογικό '0'. Όταν η είσοδος clk πάρει την τιμή 0, οι αλλαγές στην είσοδο D παύουν να επηρεάζουν τη γραμμή Y η οποία μένει σταθερή. Η τιμή που είχε η Y κατά τη στιγμή της μετάβασης μεταφέρεται στην έξοδο Q. Επομένως, στο flip flop της προηγούμενης Εικόνας, η έξοδος Q αλλάζει τιμή κατά την αρνητική (από το 1 στο 0) μετάβαση του ρολογιού clk. Μπορούμε εύκολα (τοποθετώντας τον αντιστροφέα στην είσοδο C του master latch) να δημιουργήσουμε ένα flip flop που πυροδοτείται κατά τη θετική μετάβαση του παλμού clk.

Στην επόμενη Εικόνα παρουσιάζονται τα γραφικά σύμβολα των flip-flops. Το τριγωνάκι στην είσοδο του ρολογιού δείχνει ότι τα flip-flops διεγείρονται με την θετική ακμή του παλμού του ρολογιού ενώ κύκλος πριν το τριγωνάκι δείχνει ότι τα flip-flops διεγείρονται με την αρνητική ακμή του παλμού του ρολογιού.



Εικόνα: Γραφικά σύμβολα των flip-flops

Αξίζει να σημειωθεί ότι τα γραφικά σύμβολα των flip flops μοιάζουν με εκείνα των μανδαλωτών με είσοδο επίτρεψης. Για το λόγο αυτό, όταν συναντάμε ένα τέτοιο σύμβολο πρέπει να μας προσδιορίζουν σαφώς σε τι ακολουθιακή μονάδα αναφέρεται.

2.13.4 Ασύγχρονες εισοδοί flip flop

Όπως είδαμε στην προηγούμενη παράγραφο, τα flip flops έχουν το πλεονέκτημα (έναντι των απλών μανδαλωτών) ότι οι έξοδοι μεταβάλλονται σύμφωνα με τις μεταβάσεις ενός σήματος ρολογιού (clock). Οι εισοδοί S, R, J, K, D και T των flip-flops που αναλύθηκαν ονομάζονται σύγχρονες εισοδοί, γιατί η επίδρασή τους στις εξόδους των flip-flops συγχρονίζεται με την είσοδο του παλμού του ρολογιού. Πολλές φορές όμως, είναι δυνατό να χρειαστεί να μεταβάλλουμε τις εξόδους των flip flops ασύγχρονα. Για το σκοπό αυτό, πολλά flip-flops διαθέτουν δύο επιπλέον εισόδους που ονομάζονται *ασύγχρονες εισοδοί*, γιατί η επίδρασή τους στις εξόδους των flip-flops δεν εξαρτάται από τους παλμούς του ρολογιού.

Οι ασύγχρονες εισόδους καθορίζουν την κατάσταση του flip-flop ανεξάρτητα από τις τιμές των σύγχρονων εισόδων του και χρησιμοποιούνται συνήθως για να τεθούν τα flip-flops σε μία ορισμένη αρχική κατάσταση (θέση ή μηδενισμός) πριν αρχίσει η λειτουργία τους με το ρολόι. Οι ασύγχρονες εισόδους είναι η *προτοποθέτηση* (preset) που χρησιμοποιείται για να τίθεται το flip-flop σε κατάσταση θέσης ($Q=1$) και ο *μηδενισμός* (clear) που χρησιμοποιείται για να τίθεται το flip-flop σε κατάσταση μηδενισμού ($Q=0$).

Η λειτουργία του flip-flop με ασύγχρονες εισόδους είναι ως εξής:

1. Όταν $preset=1$ και $clear=1$ τότε οι ασύγχρονες εισόδους είναι απενεργοποιημένες (τα κυκλάκια στις ασύγχρονες εισόδους σημαίνουν ότι αυτές ενεργοποιούνται με λογική κατάσταση "0") και οι έξοδοι του flip-flop ανταποκρίνονται στις σύγχρονες εισόδους καθώς και στους παλμούς του ρολογιού CP, δηλαδή πραγματοποιείται η λειτουργία χρονισμού.
2. Όταν $preset=0$ και $clear=1$, τότε το flip-flop τίθεται σε κατάσταση θέσης ($Q=1$).
3. Όταν $preset=1$ και $clear=0$, τότε το flip-flop τίθεται σε κατάσταση μηδενισμού ($Q=0$).
4. Δεν πρέπει να είναι ταυτόχρονα $preset=0$ και $clear=0$.

Για παράδειγμα, στον επόμενο Πίνακα παρουσιάζεται ο πίνακας λειτουργίας του J-K flip-flop με ασύγχρονες εισόδους όπου συνοψίζεται η λειτουργία του.

Preset	Clear	λειτουργία flip-flop
0	0	Μη χρησιμοποιούμενη
0	1	Θέση ($Q=1$)
1	0	Μηδενισμός ($Q=0$)
1	1	Λειτουργία Χρονισμού

Πίνακας: Πίνακας Λειτουργίας του J-K flip-flop με ασύγχρονες εισόδους

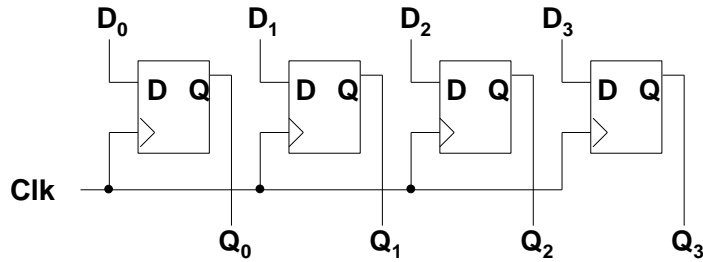
2.14 Καταχωρητές και απαριθμητές

Με τον όρο καταχώρηση εννοούμε τη λειτουργία με την οποία (δυαδικές) πληροφορίες φυλάσσονται για μετέπειτα επεξεργασία και χρήση. Στη σχεδίαση και υλοποίηση υπολογιστικών συστημάτων χρησιμοποιούνται κυρίως δύο είδη καταχωρητών, οι παράλληλοι καταχωρητές και οι καταχωρητές ολίσθησης.

2.14.1 Παράλληλοι Καταχωρητές

Ένας παράλληλος καταχωρητής χρησιμοποιείται για τη φύλαξη δεδομένων και αποτελείται από στοιχεία μνήμης (flip flops). Τα στοιχεία μνήμης ενός παράλληλου καταχωρητή οδηγούνται από μια κοινή είσοδο ρολογιού. Αυτή η είσοδος ρολογιού πυροδοτεί όλα τα flip flops, ώστε οι πληροφορίες που βρίσκονται εκείνη τη στιγμή στις εισόδους του καταχωρητή να μεταφερθούν στις εξόδους του.

Στην επόμενη Εικόνα φαίνεται το σχηματικό διάγραμμα ενός παράλληλου καταχωρητή 4 βαθμίδων (συνήθως αναφερόμαστε σε ένα τέτοιο καταχωρητή με τον όρο 4-μπιτο (4-bit), επειδή μπορεί να αποθηκεύσει πληροφορία 4 δυαδικών ψηφίων.

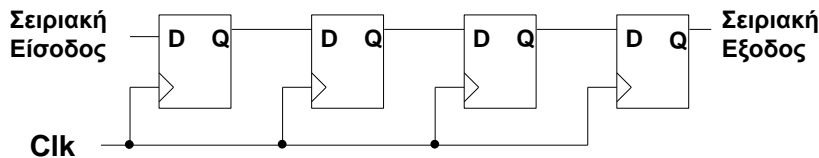


Εικόνα: Παράλληλος καταχωρητής 4 βαθμίδων

2.14.2 Καταχωρητές Ολίσθησης

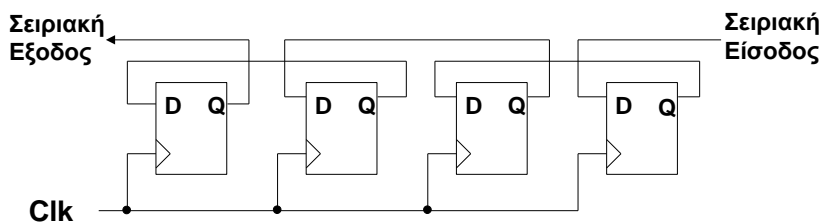
Ένας καταχωρητής που μπορεί να ‘ολισθαίνει’ τις πληροφορίες που περιέχει προς τη μια ή προς την άλλη κατεύθυνση ονομάζεται ‘καταχωρητής ολίσθησης’ (shift register). Ένας τέτοιος καταχωρητής αποτελείται από μια αλυσίδα από flip flops συνδεδεμένα στη σειρά, στα οποία η έξοδος του ενός τροφοδοτεί την είσοδο του γειτονικού του. Όλα τα flip flops παίρνουν ένα κοινό ρολόι, η ενεργοποίηση του οποίου προκαλεί την ολίσθηση από τη μια βαθμίδα στην επόμενη.

Σε ένα καταχωρητή δεξιάς ολίσθησης, σε κάθε παλμό του ρολογιού, το περιεχόμενο του καταχωρητή ολισθαίνει κατά μια θέση προς τα δεξιά. Η σειριακή είσοδος ρυθμίζει τι θα μπει στην είσοδο του πιο αριστερού flip flop σε κάθε ολίσθηση. Τη σειριακή έξοδο την παίρνουμε από την έξοδο του ακραίου δεξιού flip flop με την εφαρμογή του παλμού του ρολογιού.



Εικόνα: Καταχωρητής δεξιάς ολίσθησης τεσσάρων βαθμίδων

Αντίστοιχα, σε έναν καταχωρητή αριστερής ολίσθησης, σε κάθε παλμό του ρολογιού, το περιεχόμενο του καταχωρητή ολισθαίνει κατά μια θέση προς τα αριστερά.



Εικόνα: Καταχωρητής αριστερής ολίσθησης τεσσάρων βαθμίδων

2.14.3 Καταχωρητές ολίσθησης με παράλληλη φόρτιση

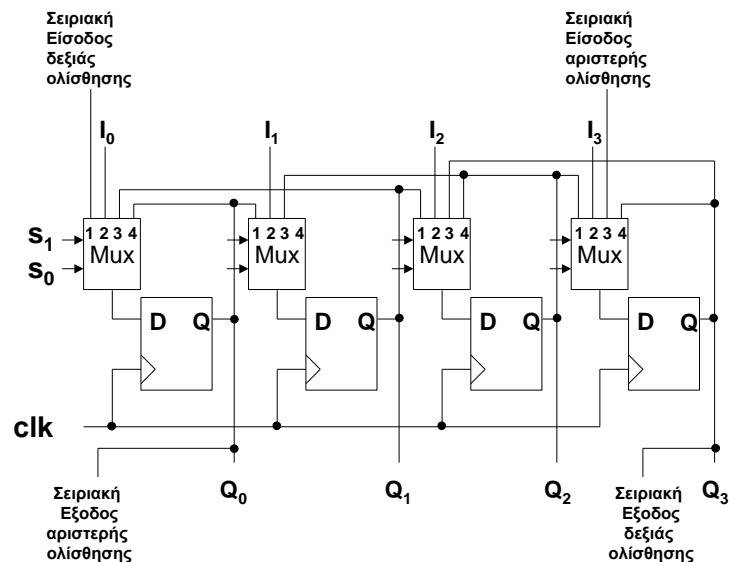
Αν έχουμε στη διάθεσή μας τις εξόδους όλων των flip flops ενός καταχωρητή ολίσθησης ώστε να μπορούμε να κάνουμε τις κατάλληλες συνδέσεις, τότε τις πληροφορίες που εισάγουμε σειριακά στον καταχωρητή μπορούμε να τις πάρουμε παράλληλα από τις εξόδους των flip flops. Αν προσθέσουμε στον καταχωρητή ολίσθησης και τη δυνατότητα παράλληλης φόρτισης, τότε μπορούμε επιπλέον να παίρνουμε σειριακά από την έξοδο τα δεδομένα εκείνα τα οποία βάλαμε παράλληλα στην είσοδο, ολισθαίνοντάς τα

μέσα από τον καταχωρητή. Έτσι, οι καταχωρητές ολίσθησης μπορούν να χρησιμοποιηθούν για τη μετατροπή σειριακών δεδομένων σε παράλληλα και αντίστροφα. Ένας καταχωρητής ολίσθησης με παράλληλη φόρτιση έχει τις ακόλουθες δυνατότητες.

- Παράλληλη φόρτωση δεδομένων στον καταχωρητή
- δεξιά ολίσθηση
- αριστερή ολίσθηση

Ένας καταχωρητής ικανός για ολισθήσεις τόσο προς τα δεξιά όσο και προς τα αριστερά λέγεται *αμφίδρομος καταχωρητής ολίσθησης*. Αν ακόμη μπορεί να φορτωθεί παράλληλα, ονομάζεται *καταχωρητής ολίσθησης με παράλληλη φόρτωση*.

Ένας καταχωρητής που έχει τις δυνατότητες αυτές φαίνεται στην επόμενη Εικόνα. Αποτελείται από 4 flip flops, η είσοδος δεδομένων (D) κάθε ενός από τα οποία τροφοδοτείται από την έξοδο ενός πολυπλέκτη (multiplexer, MUX). Οι πολυπλέκτες έχουν δύο κοινές μεταβλητές επιλογής, s_1 και s_0 , οι οποίες ρυθμίζουν τη λειτουργία του καταχωρητή.



Εικόνα: Αμφίδρομος καταχωρητής ολίσθησης με παράλληλη φόρτωση 4 ψηφίων

Στον επόμενο Πίνακα παρουσιάζεται η λειτουργία του καταχωρητή για τις διαφορετικές τιμές των s_1 και s_0 .

s_1 s_0	Είσοδος	Λειτουργία
0 0	1	δεξιά ολίσθηση
0 1	2	παράλληλη φόρτιση
1 0	3	αριστερή ολίσθηση
1 1	4	προηγούμενη τιμή

Πίνακας: Λειτουργία καταχωρητή ολίσθησης με παράλληλη φόρτιση

Η στήλη 'Είσοδος' του πίνακα δείχνει ποια από τις τέσσερις εισόδους του πολυπλέκτη μεταφέρεται στην είσοδο D του flip flop. Η λειτουργία 'προηγούμενη τιμή' σημαίνει ότι οι έξοδοι του καταχωρητή διατηρούνται αμετάβλητες.

Χρήση καταχωρητών ολίσθησης

Γνωρίζουμε ότι μέσα στον υπολογιστή τα δεδομένα μεταφέρονται παράλληλα, μέσω του διαδρόμου. Κάποιες όμως από τις περιφερειακές μονάδες εισόδου ή εξόδου (π.χ. πληκτρολόγιο, ποντίκι) επικοινωνούν με τον υπολογιστή σειριακά, στέλνοντας και λαμβάνοντας ένα-ένα τα δυαδικά ψηφία.

Οι καταχωρητές ολίσθησης μπορούν να χρησιμοποιηθούν για τη μετάδοση και λήψη των δεδομένων από τις μονάδες αυτές.

Για παράδειγμα, ας θεωρήσουμε ότι το υπολογιστικό σύστημα θέλει να αποστείλει δεδομένα (π.χ. 1 byte, 8 δυαδικά ψηφία) σειριακά. Για το σκοπό αυτό μπορεί να χρησιμοποιηθεί ένας καταχωρητής ολίσθησης 8 βαθμίδων. Ο καταχωρητής αυτός φορτώνεται με παράλληλη φόρτιση ($s_{1s0}=01$). Στη συνέχεια, λειτουργεί με δεξιά ολίσθηση για 8 κύκλους και μεταφέρει τα δεδομένα, bit-προς-bit, στη μονάδα μέσω της σειριακής εξόδου δεξιάς ολίσθησης.

Αντίστροφα, αν το υπολογιστικό σύστημα θέλει να λάβει δεδομένα, (π.χ. 8 δυαδικά ψηφία) από τη μονάδα, ο καταχωρητής λειτουργεί με αριστερή ολίσθηση για 8 κύκλους λαμβάνοντας ένα-ένα τα δυαδικά ψηφία από τη σειριακή είσοδο αριστερής ολίσθησης. Μόλις φορτωθεί ένα ολόκληρο byte στον καταχωρητή, στέλνεται (παράλληλα) στη μονάδα του υπολογιστικού συστήματος για την οποία προορίζεται (ΚΜΕ, μνήμη κ.λπ.) μέσω του διαδρόμου του συστήματος.

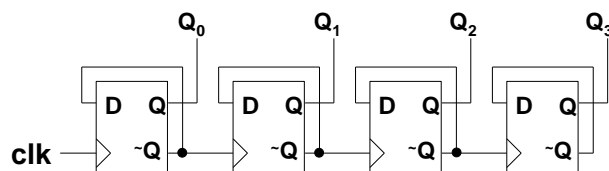
2.14.4 Μετρητές ή απαριθμητές

Ένας δυαδικός μετρητής ή απαριθμητής (counter) είναι μια μονάδα που αποτελείται από στοιχεία μνήμης (flip flops) και έχει μια είσοδο χρονισμού (ρολόι). Σε κάθε μεταβολή της εισόδου χρονισμού από 0 σε 1, οι έξοδοι του μετρητή μεταβάλλονται με τέτοιο τρόπο ώστε η δυαδική τιμή τους να είναι μεγαλύτερη κατά 1 από την δυαδική τιμή που είχαν στην προηγούμενη κατάσταση. Ένας μετρητής χαρακτηρίζεται από το πλήθος των βαθμίδων (στοιχείων μνήμης) από τα οποία αποτελείται. Οι έξοδοι ενός μετρητή που αποτελείται από n βαθμίδες, μεταβάλλονται με τέτοιο τρόπο ώστε η αριθμητική τους τιμή να αυξάνεται από 0 έως 2^n-1 . Για παράδειγμα, ένας μετρητής τριών βαθμίδων μπορεί να παίρνει τις τιμές 000, 001, 010, 011, 100, 101, 110, 111, 000, ...

Οι μετρητές διακρίνονται, ανάλογα με τη σχεδίαση, σε *ασύγχρονους* μετρητές (μετρητές ριπής) και σε *σύγχρονους* μετρητές. Στη συνέχεια θα αναφερθούμε στη σχεδίαση των δύο αυτών ειδών μετρητών.

Ασύγχρονοι μετρητές

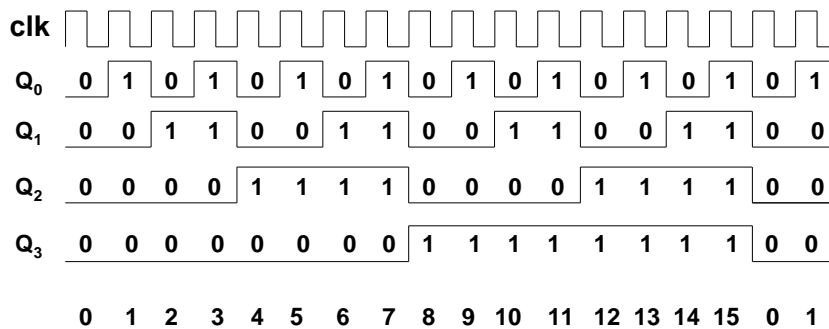
Ένας ασύγχρονος μετρητής αποτελείται από flip flops συνδεδεμένα έτσι ώστε η έξοδος κάθε flip flop να συνδέεται στην είσοδο χρονισμού του flip flop της επόμενης βαθμίδας. Το flip flop της πρώτης βαθμίδας δέχεται τους εισερχόμενους παλμούς μέτρησης. Ένας ασύγχρονος απαριθμητής 4 βαθμίδων αποτελούμενος από flip flops τύπου D φαίνεται στην επόμενη Εικόνα.



Εικόνα: Ασύγχρονος απαριθμητής 4 βαθμίδων

Ο απαριθμητής αυτός λειτουργεί ως εξής. Αρχικά οι έξοδοι όλων των βαθμίδων είναι 0. Όταν το σήμα χρονισμού γίνει 1 από 0, η τιμή του flip flop Q_0 θα αλλάξει από 0 σε 1. Σε κάθε μεταβολή του σήματος

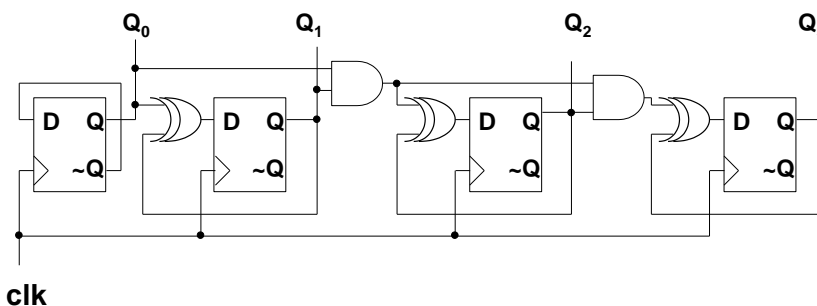
χρονισμού από '0' σε '1', η τιμή του flip flop Q_0 θα μεταβάλλεται. Επιπλέον, κάθε φορά που η τιμή του flip flop Q_0 αλλάζει από 0 σε 1, η τιμή του flip flop Q_1 θα μεταβάλλεται. Γενικά, όταν η τιμή μιας βαθμίδας μεταβάλλεται από '0' σε '1', η τιμή της επόμενης βαθμίδας θα αλλάξει. Στην επόμενη Εικόνα φαίνεται η ακολουθία που παράγεται από το δυαδικό απαριθμητή της προηγούμενης Εικόνα ξεκινώντας από την τιμή 0000.



Εικόνα: Ακολουθία που παράγεται από τον απαριθμητή 4-βαθμίδων

Σύγχρονοι Μετρητές

Οι σύγχρονοι μετρητές διαφέρουν από τους μετρητές ριπής στο ότι οι παλμοί του ρολογιού εφαρμόζονται στις εισόδους χρονισμού όλων των flip flops. Ο κοινός παλμός πυροδοτεί όλα τα flip flops συγχρόνως, και όχι το ένα μετά το άλλο, όπως στους μετρητές ριπής. Ένας σύγχρονος δυαδικός μετρητής σχεδιασμένος με flip flops τύπου D φαίνεται στην επόμενη Εικόνα.



Εικόνα: Σύγχρονος δυαδικός απαριθμητής τεσσάρων βαθμίδων

Η είσοδος δεδομένων κάθε flip flop οδηγείται από μια πύλη XOR της οποίας η μια είσοδος είναι η έξοδος του flip flop. Η απόφαση σχετικά με το αν η έξοδος ενός flip flop πρέπει να αντιστραφεί ή όχι στηρίζεται στην τιμή της εξόδου της πύλης AND που τροφοδοτεί την άλλη έξοδο της πύλης XOR. Η έξοδος του πρώτου flip flop αντιστρέφεται σε κάθε κύκλο ρολογιού. Η έξοδος του δεύτερου flip flop αντιστρέφεται όταν η έξοδος του πρώτου flip flop είναι 1. Η έξοδος των επόμενων βαθμίδων αντιστρέφεται όταν οι εξοδοί όλων των προηγούμενων βαθμίδων είναι '1'. Έτσι, το Q_2 αντιστρέφεται όταν $Q_0=Q_1=1$, ενώ η Q_3 αντιστρέφεται όταν $Q_0=Q_1=Q_2=1$.

2.15 Σύγχρονα ακολουθιακά κυκλώματα

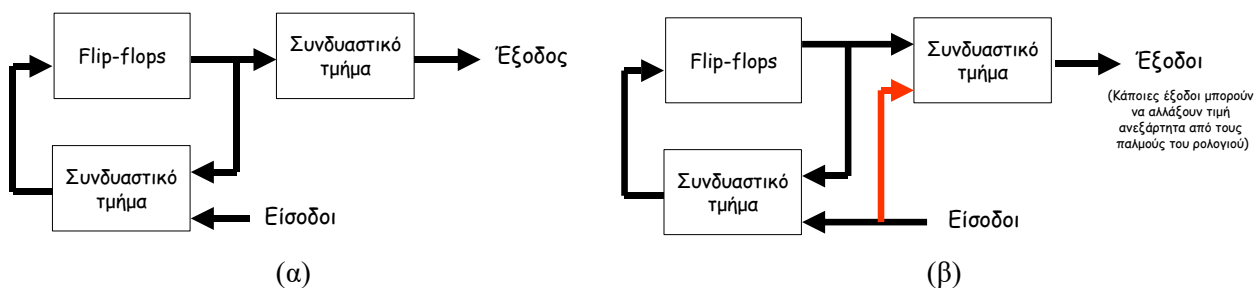
Σε προηγούμενη παράγραφο μελετήσαμε τους τύπους των flip flops (SR, D, JK, T). Στην πράξη, μας ενδιαφέρει η περίπτωση όπου περισσότερα από ένα flip flops συνδέονται μεταξύ τους και λειτουργούν συγχρονισμένα με βάση τους παλμούς κάποιου ρολογιού. Αυτό που ζητείται συνήθως είναι είτε να

αναλύσουμε τη λειτουργία ενός κυκλώματος που μας δίνεται, είτε να σχεδιάσουμε ένα ακολουθιακό κύκλωμα το οποίο να λειτουργεί σύμφωνα με κάποιες προδιαγραφές.

Αν σε ένα ακολουθιακό κύκλωμα οι εισόδους χρονισμού όλων των flip flops οδηγούνται από το ίδιο σήμα χρονισμού, τότε το κύκλωμα ονομάζεται *σύγχρονο*. Στη συνέχεια θα περιγράψουμε την ανάλυση και τη σχεδίαση τέτοιων κυκλωμάτων. Κατά την *ανάλυση* ενός κυκλώματος δίνεται το κύκλωμα και ζητείται να προσδιοριστεί η λειτουργία του. Στη *σχεδίαση* μας δίνεται η λειτουργία του κυκλώματος και μας ζητείται να βρούμε το κύκλωμα που συμπεριφέρεται σύμφωνα με αυτή.

Όπως έχει αναφερθεί, η έξοδος ενός συνδυαστικού κυκλώματος είναι συνάρτηση της εισόδου του καθώς και της λογικής που υλοποιούν οι πύλες του κυκλώματος. Σε ένα ακολουθιακό κύκλωμα, η έξοδος είναι συνάρτηση της παρούσας κατάστασης των flip flops, καθώς και της εξωτερικής εισόδου. Με τον όρο κατάσταση (state) εννοούμε τις εξόδους των flip flops στη συγκεκριμένη χρονική στιγμή. Είναι γνωστό ότι αν ένα κύκλωμα αποτελείται από n flip flops, μπορεί να έχει το πολύ 2^n διαφορετικές καταστάσεις.

Πρέπει να σημειωθεί ότι η κατάσταση δεν είναι το ίδιο πράγμα με την έξοδο του κυκλώματος. Μερικές ή και όλες οι εξόδους των flip flops μπορεί να οδηγούνται κατευθείαν στην έξοδο, αλλά γενικά η έξοδος ενός ακολουθιακού κυκλώματος είναι ένας συνδυασμός των εξόδων (καταστάσεων) των flip flops, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: (α) Κύκλωμα Moore και (β) Κύκλωμα Mealy

Αξίζει να σημειωθεί ότι στο κύκλωμα της Εικόνας (α) η έξοδος του κυκλώματος αλλάζει μόνο στην ενεργό ακμή του ρολογιού. Ένα τέτοιο κύκλωμα (στο οποίο η έξοδος αλλάζει σε συμφωνία με τους ωρολογιακούς παλμούς) ονομάζεται κύκλωμα Moore.

Αν, αντιθέτως, η εξωτερική είσοδος μπορεί να επηρεάσει την έξοδο του κυκλώματος σε οποιαδήποτε χρονική στιγμή, χωρίς να είναι απαραίτητο να συγχρονιστεί με τους ωρολογιακούς παλμούς, όπως συμβαίνει στην περίπτωση της Εικόνας β), το κύκλωμα ονομάζεται κύκλωμα Mealy.

2.15.1 Ανάλυση σύγχρονων ακολουθιακών κυκλωμάτων

Με τον όρο *ανάλυση* ενός ακολουθιακού κυκλώματος εννοούμε τον προσδιορισμό των καταστάσεών του, καθώς και των εξόδων του, ως συνάρτηση των τιμών των εισόδων του κυκλώματος και των ωρολογιακών παλμών. Η ανάλυση ενός σύγχρονου ακολουθιακού κυκλώματος περιλαμβάνει τρία βήματα: σχηματισμό των συναρτήσεων εξόδου, δημιουργία του πίνακα καταστάσεων και δημιουργία του διαγράμματος καταστάσεων. Επομένως, εντοπίζουμε στη διαδικασία τρία διακριτά βήματα, όπως περιγράφεται στη συνέχεια.

Βήμα A1: Γράφουμε τις λογικές συναρτήσεις των εισόδων των flip flops.

Βήμα A2: Καταstrώνουμε τον πίνακα καταστάσεων.

Ο πίνακας καταστάσεων περιέχει την παρούσα και την επόμενη κατάσταση του ακολουθιακού κυκλώματος. Ο πίνακας καταστάσεων αποτελείται από τρία τμήματα:

- ♦ Το πρώτο τμήμα περιέχει την παρούσα κατάσταση. Σ' αυτό καταγράφονται όλες οι δυνατές τιμές των καταστάσεων Q. Στη γενική περίπτωση κυκλώματος, που αποτελείται από n flip flops, έχουμε συνολικά 2^n δυνατές καταστάσεις.
- ♦ Το δεύτερο τμήμα περιέχει τις τιμές των εισόδων των flip flops, όπως αυτές προκύπτουν από τις συναρτήσεις εισόδου του βήματος A1 και τις τιμές της παρούσας κατάστασης.
- ♦ Το τρίτο τμήμα περιέχει την επόμενη κατάσταση του κυκλώματος. Κάθε τιμή του τμήματος αυτού προκύπτει από τις αντίστοιχες τιμές της παρούσας κατάστασης και των εισόδων, λαμβάνοντας υπόψη το χαρακτηριστικό πίνακα του flip flop.

Οι χαρακτηριστικοί πίνακες των τεσσάρων ειδών flip flops (S-R, J-K, D, T) είναι ίδιοι με τους πίνακες των αντίστοιχων μανδαλωτών που έχουν δοθεί προηγουμένως, ωστόσο επαναλαμβάνονται εδώ για λόγους πληρότητας.

S-R flip flop			
Q(n)	S	R	Q(n+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

J-K flip flop			
Q(n)	J	K	Q(n+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

D flip flop		
Q(n)	D	Q(n+1)
0	0	0
0	1	1
1	0	0
1	1	1

T flip flop		
Q(n)	T	Q(n+1)
0	0	0
0	1	1
1	0	1
1	1	0

Πίνακας: Χαρακτηριστικοί πίνακες flip flops

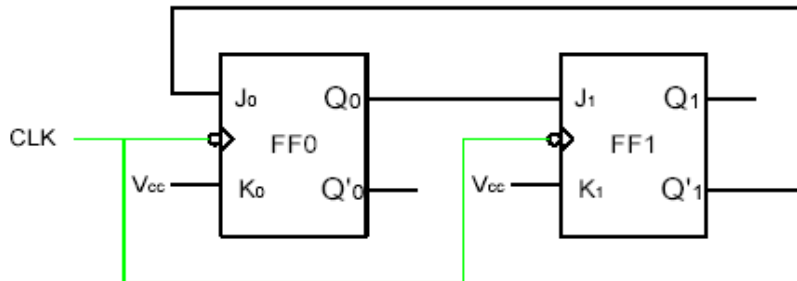
Βήμα A3: Σχεδιάζουμε το διάγραμμα καταστάσεων.

Το διάγραμμα καταστάσεων (state diagram) είναι η γραφική απεικόνιση του πίνακα καταστάσεων. Κάθε κατάσταση απεικονίζεται με ένα κύκλο και κάθε μετάβαση αναπαρίσταται με ένα βέλος.

Στη συνέχεια θα περιγράψουμε τα διαδικασία αυτή, ξεκινώντας από απλά ακολουθιακά κυκλώματα χωρίς εισόδους και εξόδους και καταλήγοντας σε κυκλώματα με εισόδους και εξόδους.

2.15.1.1 Σύγχρονα ακολουθιακά κυκλώματα χωρίς εξωτερικές εισόδους και εξόδους

Παράδειγμα 1: Θα εφαρμόσουμε τη διαδικασία που περιγράψαμε στο κύκλωμα της επόμενης Εικόνας. Πρόκειται για ένα κύκλωμα με δύο flip flops τύπου JK αρνητικής ακμής πυροδότησης. Στο κύκλωμα δεν εφαρμόζεται καμιά εξωτερική είσοδος, και οι εξοδοί του είναι αυτές των flip flops (κατάσταση των flip flops).



Εικόνα: Σύγχρονο ακολουθιακό κύκλωμα

Για να διευκολυνθούμε στην ανάλυση του κυκλώματος, απαριθμούμε τα flip flops αρχίζοντας από το 0. Βάζουμε τον αύξοντα αριθμό του κάθε flip flop ως δείκτη σε κάθε σύμβολο εισόδου και εξόδου των flip-flops. Έτσι έχουμε το FF0 με εισόδους J0, K0 και εξόδους Q0, Q0' και το FF1 με εισόδους J1, K1 και εξόδους Q1, Q1'. Η ανάλυση πραγματοποιείται ως εξής:

Βήμα A1: Γράφουμε τις λογικές συναρτήσεις των εισόδων των flip flops.

$$K0 = 1 \quad K1 = 1$$

Παρούσα κατάσταση		Είσοδοι				Επόμενη κατάσταση	
Q1	Q0	J1	K1	J0	K0	Q1	Q0
0	0	0	1	1	1	0	1
0	1	1	1	1	1	1	0
1	0	0	1	0	1	0	0
1	1	1	1	1	1	0	0

Πίνακας: Πίνακας καταστάσεων του κυκλώματος του Παραδείγματος 1

Βήμα A2: Καταστρώνουμε τον πίνακα καταστάσεων.

Ο πίνακας καταστάσεων αποτελείται από τρία τμήματα:

- ♦ Το πρώτο τμήμα περιέχει την παρούσα κατάσταση. Σ' αυτό καταγράφονται όλες οι δυνατές τιμές των καταστάσεων Q. Στην προκειμένη περίπτωση έχουμε 2 flip flops και άρα 4 δυνατές τιμές για το ζεύγος Q1, Q0. Στη γενική περίπτωση κυκλώματος, που αποτελείται από n flip flops, έχουμε συνολικά 2^n δυνατές καταστάσεις.
- ♦ Το δεύτερο τμήμα περιέχει τις τιμές των εισόδων των flip flops, όπως αυτές προκύπτουν από τις συναρτήσεις εισόδου του βήματος A1 και τις τιμές της παρούσας κατάστασης. Έτσι, παρατηρούμε ότι η στήλη J1 είναι ίδια με την στήλη Q0 της παρούσας κατάστασης, αφού $J1 = Q0$. Επίσης, η στήλη J0 είναι συμπληρωματική της στήλης Q1, αφού $J0 = Q1'$. Τέλος, οι τιμές των K1 και K0 είναι όλες ίσες με 1.

- ♦ Το τρίτο τμήμα περιέχει την επόμενη κατάσταση του κυκλώματος. Κάθε τιμή του τμήματος αυτού προκύπτει από τις αντίστοιχες τιμές της παρούσας κατάστασης και των εισόδων, λαμβάνοντας υπόψη τον τρόπο λειτουργίας του flip flop.

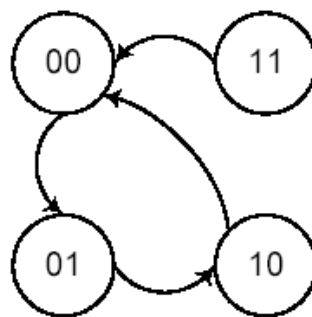
Έτσι, οι τιμές της στήλης Q1 της επόμενης κατάστασης προκύπτουν από την στήλη Q1 της παρούσας κατάστασης και τις τιμές των στηλών εισόδου J1, K1. Για παράδειγμα, η δεύτερη τιμή του Q1 της επόμενης κατάστασης είναι 1, επειδή η αντίστοιχη τιμή του Q1 της παρούσας κατάστασης είναι 0 και οι εισοδοί $J1 = 1, K1 = 1$, που σημαίνει ότι το flip-flop αλλάζει κατάσταση με την έλευση του ενεργού παλμού του ρολογιού (toggle).

Βήμα A3: Σχεδιάζουμε το διάγραμμα καταστάσεων.

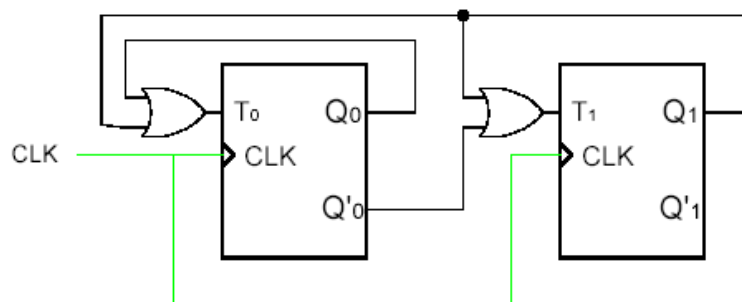
Το διάγραμμα καταστάσεων του κυκλώματος που εξετάζουμε παρουσιάζεται στην επόμενη Εικόνα.

Το διάγραμμα προέκυψε από τις αντίστοιχες τιμές της παρούσας και επόμενης κατάστασης του Πίνακα. Βλέπουμε, δηλαδή, πως, αν η παρούσα κατάσταση (Q1Q0) είναι 00, η επόμενη κατάσταση (τρίτο τμήμα του πίνακα καταστάσεων) είναι 01. Άρα, με την έλευση του ενεργού παλμού το κύκλωμα μεταβαίνει από το 00 στο 01. Όμοια βρίσκουμε ότι με τον επόμενο παλμό του ρολογιού το κύκλωμα αλλάζει κατάσταση και από το 01 μεταβαίνει στο 10, ενώ από το 10 μεταβαίνει στο 00.

Επομένως, το συγκεκριμένο κύκλωμα περνά διαδοχικά από τις καταστάσεις 00, 01, 10 ή 0, 1, 2 στο δεκαδικό σύστημα αρίθμησης. Πρόκειται, επομένως, για ένα κύκλωμα απαρίθμησης παλμών (counter) με υπόλοιπο ως προς 3. Αν το κύκλωμα για κάποιο λόγο (π.χ. κατά την έναρξη της τροφοδοσίας του) βρεθεί στην κατάσταση 11 (δηλαδή $Q1 = 1, Q0 = 1$), τότε με τον πρώτο ενεργό ωρολογιακό παλμό μεταβαίνει στην κατάσταση 00 και από εκεί συνεχίζει.



Εικόνα: Διάγραμμα καταστάσεων για το κύκλωμα του Παραδείγματος 1



Εικόνα: Σύγχρονο ακολουθιακό κύκλωμα Παραδείγματος 2

Παράδειγμα 2. Για να αναλύσουμε τη λειτουργία του σύγχρονου ακολουθιακού κυκλώματος της προηγούμενης Εικόνας, το οποίο αποτελείται από δύο flip flops τύπου T θετικής ακμής πυροδότησης, εργαζόμαστε ως εξής.

Βήμα Α1: Γράφουμε τις συναρτήσεις εισόδου.

$$T0 = Q0 + Q1$$

$$T1 = Q0' + Q1$$

Βήμα Α2: Καταστρώνουμε τον πίνακα καταστάσεων.

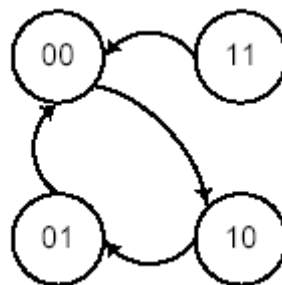
Το τμήμα της παρούσας κατάστασης περιέχει όλες τις δυνατές τιμές των εξόδων των flip flops. Με βάση αυτές και τις συναρτήσεις εισόδου συμπληρώνουμε το μεσαίο τμήμα του πίνακα καταστάσεων, όπου φαίνονται οι τιμές των εισόδων T0 και T1. Η επόμενη κατάσταση προσδιορίζεται με τη βοήθεια του πίνακα λειτουργίας του flip flop τύπου T και τις αντίστοιχες τιμές της παρούσας κατάστασης και των εισόδων αυτού.

Παρούσα κατάσταση		Είσοδοι		Επόμενη κατάσταση	
Q1	Q0	T1	T0	Q1	Q0
0	0	1	0	1	0
0	1	0	1	0	0
1	0	1	1	0	1
1	1	1	1	0	0

Πίνακας: Πίνακας καταστάσεων του κυκλώματος του Παραδείγματος 2

Βήμα Α3: Σχεδιάζουμε το διάγραμμα καταστάσεων.

Με βάση τις τιμές της παρούσας και επόμενης κατάστασης σχεδιάζουμε το διάγραμμα καταστάσεων της επόμενης Εικόνας. Παρατηρούμε ότι το κύκλωμα αυτό με κάθε ωρολογιακό παλμό αλλάζει κατάσταση διατρέχοντας τις τιμές του δεκαδικού συστήματος: 2, 1, 0, 2, 1, 0, 2, Πρόκειται, επομένως, για ένα κύκλωμα απαρίθμησης. Σε αντίθεση με την περίπτωση του κυκλώματος του προηγούμενου παραδείγματος, το κύκλωμα αυτό απαριθμεί ανάστροφα. Αν το κύκλωμα βρεθεί στην κατάσταση Q1Q0 = 11, τότε με τον επόμενο ωρολογιακό παλμό μεταβαίνει στο 00 και από εκεί και έπειτα με κάθε νέο παλμό συνεχίζει τον κύκλο του.

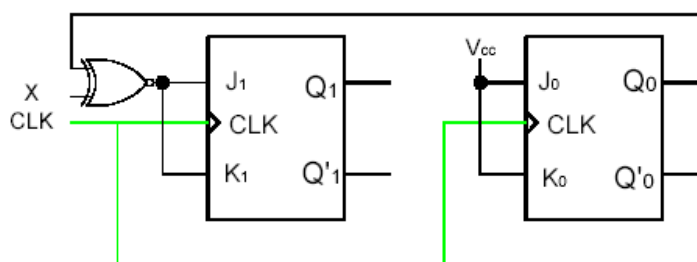


Εικόνα: Διάγραμμα καταστάσεων για το κύκλωμα του Παραδείγματος 2

2.15.1.2 Σύγχρονα ακολουθιακά κυκλώματα με εξωτερικές εισόδους

Αν το κύκλωμα που θέλουμε να αναλύσουμε έχει εξωτερικές εισόδους, τότε πρέπει να τις λάβουμε υπόψη μας στη διαδικασία της ανάλυσης. Η διαδικασία της ανάλυσης του κυκλώματος περιλαμβάνει τρία βήματα, όπως φαίνεται στο ακόλουθο παράδειγμα.

Παράδειγμα 3. Έστω το κύκλωμα της επόμενης Εικόνας. Η διαδικασία ανάλυσης είναι η ακόλουθη.



Εικόνα: Σύγχρονο ακολουθιακό κύκλωμα παραδείγματος 3

Βήμα A1: Γράφουμε τις συναρτήσεις εισόδου.

$$J_1 = K_1 = X \otimes Q_0$$

$$J_0 = K_0 = 1$$

Βήμα A2: Καταστρώνουμε τον πίνακα καταστάσεων.

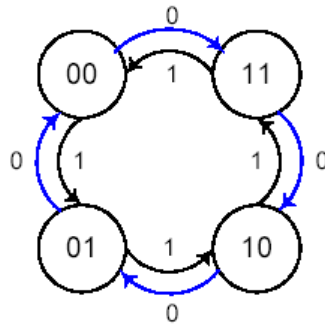
Παρατηρούμε ότι οι εισόδους J, K των flip flops είναι συναρτήσεις όχι μόνο της παρούσας κατάστασης Q, αλλά και της τιμής της εισόδου X. Αυτό σημαίνει πως, για να προσδιορίσουμε την επόμενη κατάσταση, πρέπει να ξέρουμε την παρούσα κατάσταση και την είσοδο X. Έτσι, στο πρώτο τμήμα του πίνακα καταστάσεων περιλαμβάνονται και οι τιμές του X. Οι δυνατοί συνδυασμοί των τριών μεταβλητών X, Q1, Q0 είναι 8, όπως φαίνεται στον πίνακα.

Παρούσα κατάσταση			Είσοδοι				Επόμενη κατάσταση	
X	Q1	Q0	J1	K1	J0	K0	Q1	Q0
0	0	0	1	1	1	1	1	1
0	0	1	0	0	1	1	0	0
0	1	0	1	1	1	1	0	1
0	1	1	0	0	1	1	1	0
1	0	0	0	0	1	1	0	1
1	0	1	1	1	1	1	1	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	0	0

Πίνακας: Πίνακας καταστάσεων του κυκλώματος του Παραδείγματος 3

Βήμα A3: Σχεδιάζουμε το διάγραμμα καταστάσεων.

Σε κάθε βέλος (μετάβαση) του διαγράμματος καταστάσεων σημειώνουμε και την αντίστοιχη τιμή της εισόδου X. Έτσι παρατηρούμε ότι όταν $X = 1$, το κύκλωμα απαριθμεί ‘προς τα πάνω’, δηλαδή 0, 1, 2, 3, 0, 1, ..., ενώ, όταν $X = 0$, το κύκλωμα απαριθμεί ‘προς τα κάτω’, δηλαδή 3, 2, 1, 0, 3, 2, ... Συνεπώς, πρόκειται για ένα κύκλωμα απαρίθμησης 4 παλμών (modulo-4), η λειτουργία του οποίου ρυθμίζεται από την εξωτερική είσοδο X.

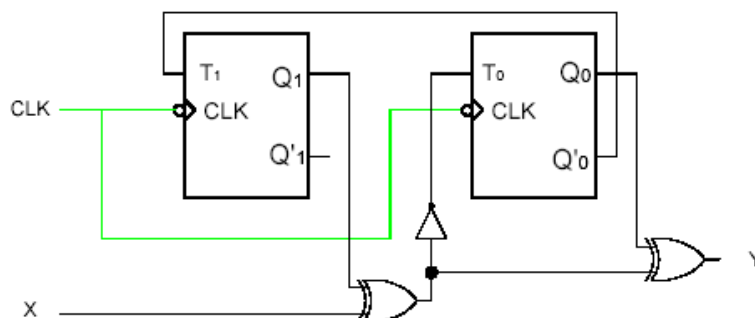


Εικόνα: Διάγραμμα μεταβάσεων για το κύκλωμα του Παραδείγματος 3

2.15.1.3 Σύγχρονα ακολουθιακά κύκλωμα με εξωτερικές εισόδους και εξόδους

Στη γενική περίπτωση, ένα σύγχρονο ακολουθιακό κύκλωμα μπορεί να έχει και εισόδους και εξόδους. Και στην περίπτωση αυτή η ανάλυση της λειτουργίας γίνεται σε τρία βήματα. Ο πίνακας καταστάσεων θα αποτελείται από ένα επιπλέον τμήμα, στο οποίο θα καταγράφονται οι τιμές των εξωτερικών εξόδων για κάθε δυνατό συνδυασμό των εισόδων και της παρούσας κατάστασης. Η διαδικασία φαίνεται στο ακόλουθο παράδειγμα.

Παράδειγμα 4: Για να αναλύσουμε τη λειτουργία του κυκλώματος της επόμενης Εικόνας, εργαζόμαστε ως εξής:



Εικόνα: Σύγχρονο ακολουθιακό κύκλωμα με T flip flops και εξωτερική είσοδο και έξοδο

Βήμα A1: Γράφουμε τις εξισώσεις εισόδου των flip flops.

$$T1 = Q0'$$

$$T0 = X \otimes Q1$$

Αφού το κύκλωμα διαθέτει και μια εξωτερική έξοδο Y, γράφουμε τη λογική της συνάρτησης, ώστε στη συνέχεια να είμαστε έτοιμοι να προσδιορίσουμε την τιμή της για κάθε δυνατό συνδυασμό της παρούσας κατάστασης και της εισόδου. Έτσι έχουμε:

$$Y = X \oplus Q1 \oplus Q0$$

Βήμα Α2: Καταστρώνουμε τον πίνακα καταστάσεων.

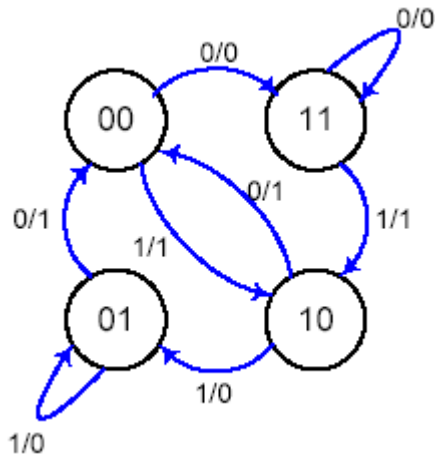
Έχουμε μία εξωτερική είσοδο και δύο flip flops, επομένως 8 περιπτώσεις. Με βάση αυτές τις τιμές και τις συναρτήσεις εισόδου, συμπληρώνουμε το δεύτερο τμήμα του πίνακα καταστάσεων, που περιέχει όλες τις δυνατές περιπτώσεις στις οποίες μπορούν να βρεθούν οι είσοδοι T0, T1 των δύο flip flops. Στο σημείο αυτό, και πριν συνεχίσουμε στον προσδιορισμό της επόμενης κατάστασης (δεξιό τμήμα του Πίνακα), καταγράφουμε τις τιμές εξόδου της Y. Οι τιμές της προκύπτουν από τις αντίστοιχες τιμές της εξωτερικής εισόδου και της παρούσας κατάστασης. Τέλος, προσδιορίζουμε την επόμενη κατάσταση από την παρούσα κατάσταση και τις τιμές των εισόδων T0, T1 των flip flops με βάση τον πίνακα λειτουργίας τους.

Είσοδος και Παρούσα κατάσταση			Είσοδοι		Έξοδος	Επόμενη κατάσταση	
X	Q1	Q0	T1	T0	Y	Q1	Q0
0	0	0	1	1	0	1	1
0	0	1	0	1	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
1	0	0	1	0	1	1	0
1	0	1	0	0	0	0	1
1	1	0	1	1	0	0	1
1	1	1	0	1	1	1	0

Πίνακας: Πίνακας καταστάσεων του κυκλώματος του Παραδείγματος 4

Βήμα Α3: Σχεδιάζουμε το διάγραμμα καταστάσεων.

Το διάγραμμα καταστάσεων σχεδιάζεται κατά τα γνωστά από την παρούσα και επόμενη κατάσταση του προηγούμενου Πίνακα. Σε κάθε μετάβαση σημειώνουμε τις αντίστοιχες τιμές των εξωτερικών εισόδων και εξόδων με τη μορφή X/Y. Έτσι, αν το κύκλωμα βρεθεί στην κατάσταση 10 και είσοδος $X = 0$, τότε ο παλμός θα το οδηγήσει στην κατάσταση 00 και η έξοδος θα είναι $Y = 1$. Για το λόγο αυτό στο βέλος από το 10 στο 00 σημειώνουμε 0/1. Αν όμως η είσοδος $X = 1$, τότε η έλευση του ενεργού παλμού θα το οδηγήσει στην κατάσταση 01 και η έξοδος Y θα ισούται με 0 ($Y=0$). Παρατηρούμε ότι, αν το κύκλωμα βρεθεί στην κατάσταση 01 και η είσοδος $X = 1$, τότε αυτό εγκλωβίζεται στην ίδια κατάσταση. Αυτό δείχνεται στο διάγραμμα καταστάσεων με ένα βρόχο. Όμοια, αν το κύκλωμα βρεθεί στην κατάσταση 11 και η είσοδος είναι $X = 0$, τότε παραμένει στην ίδια κατάσταση μέχρις ότου γίνει $X = 1$, οπότε και με την έλευση του ενεργού παλμού μεταβαίνει στην κατάσταση 10.



Εικόνα: Διάγραμμα μεταβάσεων για το κύκλωμα του Παραδείγματος 4

2.15.2 Σχεδίαση σύγχρονων ακολουθιακών κυκλωμάτων

Η σχεδίαση σύγχρονων ακολουθιακών κυκλωμάτων είναι αντίστροφη από την ανάλυση που περιγράψαμε στην προηγούμενη παράγραφο. Στη σχεδίαση δίνονται οι προδιαγραφές του κυκλώματος και ζητείται να βρεθεί το κύκλωμα το οποίο λειτουργεί σύμφωνα με τις προδιαγραφές αυτές. Συνήθως μας δίνεται το διάγραμμα καταστάσεων του κυκλώματος και κατά συνέπεια και ο πίνακας καταστάσεων αυτού. Αυτό σημαίνει ότι γνωρίζουμε την παρούσα και την επόμενη κατάσταση καθενός από τα flip flops, και μένει να προσδιορίσουμε τις εισόδους των flip flops, ώστε με τον ερχομό κάθε ωρολογιακού παλμού το κάθε flip flop να μεταβαίνει από την παρούσα κατάσταση στην επόμενη. Τα βήματα για τη σχεδίαση ενός σύγχρονου ακολουθιακού κυκλώματος μπορούν να κωδικοποιηθούν ως εξής:

Βήμα Σ1: Σχεδιάζουμε το διάγραμμα καταστάσεων του κυκλώματος και από αυτό καταστρώνουμε τον πίνακα καταστάσεων.

Βήμα Σ2: Προσδιορίζουμε τις απλοποιημένες συναρτήσεις εισόδου των flip flops.

Βήμα Σ3: Σχεδιάζουμε το λογικό κύκλωμα.

Πριν συνεχίσουμε στην αναλυτική περιγραφή των βημάτων αυτών, είναι απαραίτητο να υπενθυμίσουμε την έννοια του πίνακα διέγερσης (excitation table) των flip flops. Ο πίνακας διέγερσης ενός flip flop μας δίνει τις επιθυμητές τιμές των εισόδων του, όταν γνωρίζουμε τις τιμές της παρούσας και της επόμενης κατάστασης αυτού.

Οι πίνακες διέγερσης των flip-flops δίνονται στην επόμενη Εικόνα. Από τους πίνακες αυτούς παρατηρούμε ότι το flip flop τύπου JK έχει τις περισσότερες περιπτώσεις συνθηκών αδιαφορίας στις εισόδους του έναντι των άλλων. Επίσης, παρατηρούμε ότι η είσοδος D του flip flop τύπου D είναι ίδια με την επόμενη κατάσταση $Q(t+1)$ αυτού, όπως άλλωστε ήταν αναμενόμενο, αφού για να γίνει η επόμενη κατάσταση 0 ή 1, πρέπει η είσοδος να είναι 0 ή 1 αντίστοιχα, ανεξάρτητα από το ποια είναι η παρούσα κατάσταση. Οι πίνακες διέγερσης είναι απαραίτητοι για τη σχεδίαση σύγχρονων ακολουθιακών κυκλωμάτων κατ' αναλογία με τους πίνακες λειτουργίας των flip flops, οι οποίοι χρησιμοποιούνται για την ανάλυση των κυκλωμάτων.

S-R flip-flop			
Q(t)	Q(t-1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

D flip-flop		
Q(t)	Q(t-1)	D
0	0	0
0	1	1
1	0	0
1	1	1

J-K flip-flop			
Q(t)	Q(t-1)	J	K
0	0	0	X
0	1	1	X
1	0	X	0
1	1	X	1

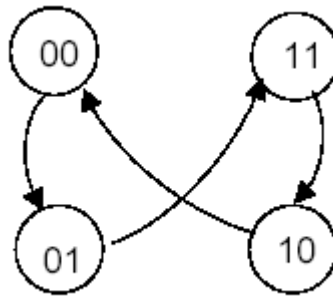
T flip-flop		
Q(t)	Q(t-1)	T
0	0	0
0	1	1
1	0	1
1	1	0

Εικόνα: Πίνακες διέγερσης των flip flops

Παράδειγμα 5: Για να σχεδιάσουμε σύγχρονο ακολουθιακό κύκλωμα το οποίο να περνά διαδοχικά από τις καταστάσεις 0, 1, 3, 2 (δηλαδή να απαριθμεί σύμφωνα με τον κώδικα Gray), με flip flops τύπου JK αρνητικής ακμής πυροδότησης εργαζόμαστε ως εξής.

Υπολογίζουμε πόσα flip flops χρειαζόμαστε. Εφόσον έχουμε 4 διαφορετικές καταστάσεις από το 0 μέχρι το 3, σημαίνει πως θα χρειαστούμε 2 flip flops. Επομένως, ακολουθούμε τα τρία βήματα σχεδίασης και έχουμε:

Βήμα Σ1: Σχεδιάζουμε το διάγραμμα καταστάσεων και με βάση αυτό τον αντίστοιχο πίνακα καταστάσεων.



Εικόνα: Διάγραμμα καταστάσεων για το κύκλωμα του Παραδείγματος 5

Τα τμήματα της παρούσας και επόμενης κατάστασης του πίνακα καταστάσεων εξάγονται απ' ευθείας από το διάγραμμα καταστάσεων της Εικόνας. Το τρίτο τμήμα όμως του πίνακα καταστάσεων, το οποίο περιέχει τις αντίστοιχες τιμές των εισόδων J, K των flip flops, συμπληρώνεται με τη βοήθεια του πίνακα διέγερσης για το flip flop τύπου JK. Έτσι, αν το FF1 βρίσκεται κατά την παρούσα χρονική στιγμή στην κατάσταση 0 και θέλουμε να παραμείνει σ' αυτή (επόμενη κατάσταση Q1=0), τότε οι εισοδοί J, K αυτού πρέπει να είναι 0, X αντίστοιχα. Αν το FF1 βρίσκεται στο 0 και θέλουμε να μεταβεί στο 1, τότε οι εισοδοί J, K πρέπει να είναι 1, X αντίστοιχα.

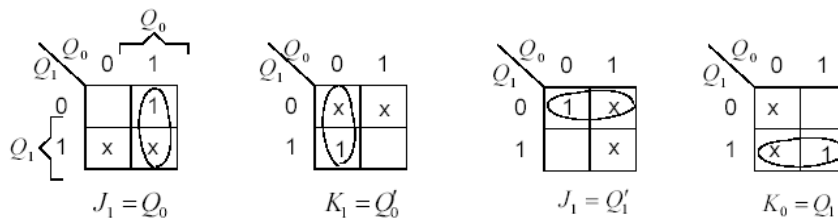
Συνεχίζοντας με τον ίδιο τρόπο συμπληρώνουμε τις υπόλοιπες γραμμές και στήλες του τμήματος του πίνακα καταστάσεων.

Παρούσα κατάσταση		Επόμενη κατάσταση		Είσοδοι			
Q1	Q0	Q1	Q0	J1	K1	J0	K0
0	0	0	1	0	X	1	X
0	1	1	1	1	X	X	0
1	0	0	0	X	1	0	X
1	1	1	0	X	0	X	1

Πίνακας: Πίνακας καταστάσεων για το διάγραμμα καταστάσεων του Παραδείγματος 5

Βήμα Σ2: Προσδιορίζουμε τις απλοποιημένες συναρτήσεις εισόδου των flip flops.

Εφόσον οι συναρτήσεις εισόδου των flip flops εμπεριέχουν αδιάφορους όρους, μπορούμε να εξάγουμε απλοποιημένες μορφές χρησιμοποιώντας χάρτες Karnaugh, όπως φαίνεται στην εποόμενη Εικόνα για καθεμία είσοδο των flip flops, με βάση τον πίνακα καταστάσεων.

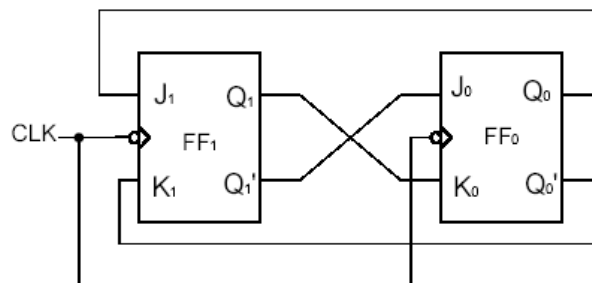


Εικόνα: Απλοποίηση των συναρτήσεων εισόδου με χάρτες Karnaugh

Μπορούμε να παρατηρήσουμε ότι οι απλοποιημένες συναρτήσεις είναι $J1=Q0$, $K1=Q0'$, $J0=Q1'$ και $K0=Q1$.

Βήμα Σ3: Σχεδιάζουμε το λογικό κύκλωμα.

Έχοντας προσδιορίσει τις συναρτήσεις εισόδου των flip flops, προχωρούμε στη σχεδίαση του ζητούμενου λογικού κυκλώματος, όπως φαίνεται στην εποόμενη Εικόνα.



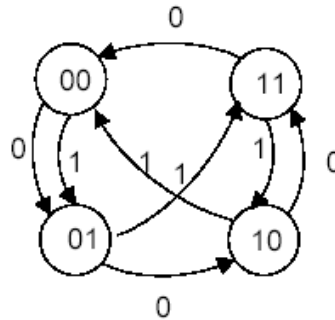
Εικόνα: Σύγχρονο ακολουθιακό κύκλωμα για το Παράδειγμα 5

Στη συνέχεια θα δούμε παραδείγματα στα οποία το ζητούμενο κύκλωμα ελέγχεται εξωτερικά από κάποια είσοδο.

Παράδειγμα 6: Έστω ότι ζητείται να σχεδιαστεί σύγχρονο ακολουθιακό κύκλωμα, το οποίο να περνά από τις καταστάσεις 0, 1, 2, 3 (κανονική απαρίθμηση) ή 0, 1, 3, 2 (απαρίθμηση σύμφωνα με τον κώδικα Gray) ανάλογα με το αν η είσοδος X είναι 0 ή 1, με τη χρήση flip flops τύπου JK θετικής ακμής πυροδότησης.

Το ζητούμενο κύκλωμα θα πρέπει να αποτελείται από δύο flip flops τύπου JK, αφού και στις δύο περιπτώσεις αυτό θα πρέπει να διατρέχει τις καταστάσεις 0, 1, 2, 3 αλλά με διαφορετική σειρά. Επομένως, αρχίζουμε τη σχεδίαση ακολουθώντας τη γνωστή διαδικασία.

Βήμα Σ1: Σχεδιάζουμε το διάγραμμα καταστάσεων και από αυτό τον πίνακα καταστάσεων.



Εικόνα: Διάγραμμα καταστάσεων για το Παράδειγμα 6

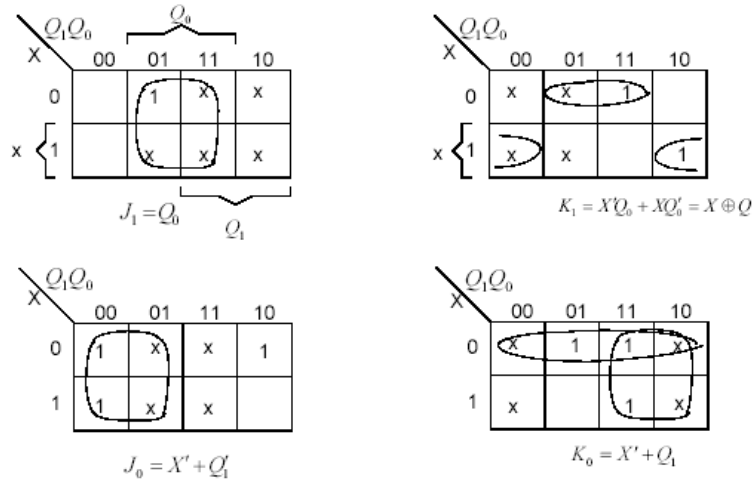
Είσοδοι & Παρούσα Κατάσταση			Επόμενη Κατάσταση		Είσοδοι			
X	Q1	Q0	Q1	Q0	J1	K1	J0	K0
0	0	0	0	1	0	X	1	X
0	0	1	1	0	1	X	X	1
0	1	0	1	1	X	0	1	X
0	1	1	0	0	X	1	X	1
1	0	0	0	1	0	X	Q	X
1	0	1	1	1	1	X	X	0
1	1	0	0	0	X	1	0	X
1	1	1	1	0	X	0	X	1

Πίνακας: Πίνακας καταστάσεων για το Παράδειγμα 6

Οι αριθμοί στα βέλη του διαγράμματος καταστάσεων υποδηλώνουν τις τιμές της εισόδου X. Οι τιμές του X συμπεριλαμβάνονται στο πρώτο τμήμα του πίνακα καταστάσεων, όπου καταγράφονται όλοι οι δυνατοί συνδυασμοί εξωτερικής εισόδου και παρούσας κατάστασης. Η επόμενη κατάσταση (δεύτερο τμήμα του πίνακα καταστάσεων) προκύπτει από το διάγραμμα καταστάσεων, ενώ οι τιμές των εισόδων των flip flops (τρίτο τμήμα του πίνακα καταστάσεων) προκύπτουν από τις τιμές της παρούσας και επόμενης κατάστασης κάθε flip flop σε συνδυασμό με τον αντίστοιχο πίνακα διέγερσης.

Βήμα Σ2: Προσδιορίζουμε τις απλοποιημένες συναρτήσεις εισόδου των flip flops.

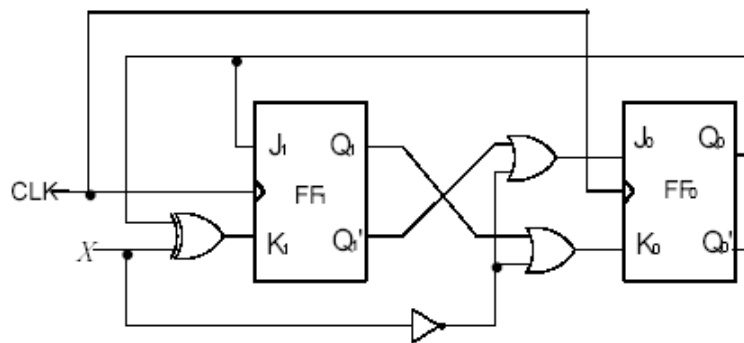
Για να απλοποιήσουμε τις συναρτήσεις των εισόδων J1, K1, J0, K0 των flip flops, χρησιμοποιούμε χάρτες Karnaugh, όπως φαίνεται στην επόμενη Εικόνα. Σ' αυτούς απεικονίζονται οι τιμές της εξωτερικής εισόδου και της παρούσας κατάστασης (πρώτο τμήμα του πίνακα καταστάσεων) και η αντίστοιχη στήλη τιμών κάθε εισόδου (τρίτο τμήμα του πίνακα καταστάσεων).



Εικόνα: Χάρτες Karnaugh για την απλοποίηση των συναρτήσεων εισόδου του Παραδείγματος 6

Βήμα Σ3: Σχεδιάζουμε το λογικό κύκλωμα.

Γνωρίζοντας τις συναρτήσεις της κάθε εισόδου των flip flops, προχωρούμε στη σχεδίαση του κυκλώματος.



Εικόνα: Κύκλωμα που υλοποιεί το διάγραμμα καταστάσεων του Παραδείγματος 6

10.3 Σχεδίαση με flip flop τύπου D

Η ανάλυση και σχεδίαση σύγχρονων ακολουθιακών κυκλωμάτων με flip flops τύπου D είναι εξαιρετικά απλή, αφού στους πίνακες καταστάσεων αυτών το τμήμα των εισόδων θα μπορούσε να παραληφθεί, γιατί αυτό είναι ίδιο με εκείνο της επόμενης κατάστασης. Από την άλλη μεριά όμως, τα κυκλώματα που προκύπτουν είναι συνήθως πιο πολύπλοκα, αφού διαθέτουμε μία μόνο είσοδο για την υλοποίηση όλων των λειτουργιών. Στη συνέχεια θα σχεδιάσουμε το κύκλωμα του Παραδείγματος 6 με flip flops τύπου D.

Βήμα Σ1: Σχεδιάζουμε το διάγραμμα καταστάσεων και με βάση αυτό καταστρώνουμε τον αντίστοιχο επόμενο Πίνακα καταστάσεων.

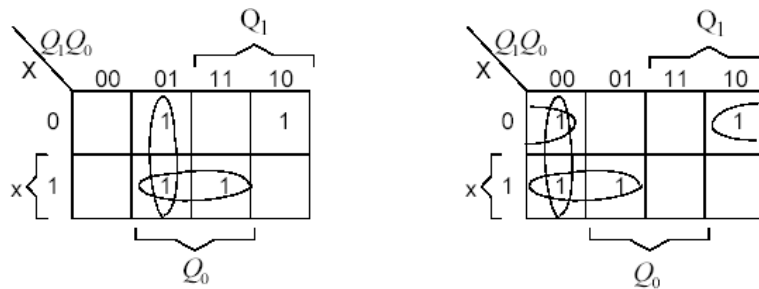
Είσοδοι & Παρούσα Κατάσταση			Επόμενη Κατάσταση		Είσοδοι	
X	Q1	Q0	Q1	Q0	D1	D0
0	0	0	0	1	0	1
0	0	1	1	0	1	0
0	1	0	1	1	1	1
0	1	1	0	0	0	0
1	0	0	0	1	0	1
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	1	1	0	1	0

Πίνακας: Πίνακας καταστάσεων για το Παράδειγμα 6 με D flip flops

Τα δύο πρώτα τμήματα του πίνακα καταστάσεων είναι ίδια με εκείνα του Προηγούμενου Πίνακα. Το τρίτο τμήμα του Πίνακα δίνει τις εισόδους των D flip flops.

Βήμα Σ2: Προσδιορίζουμε τις απλοποιημένες συναρτήσεις των εισόδων.

Για την καλύτερη απλοποίηση των συναρτήσεων εισόδου D1, D0, θα χρησιμοποιήσουμε χάρτες Karnaugh, όπως φαίνεται στην επόμενη Εικόνα.

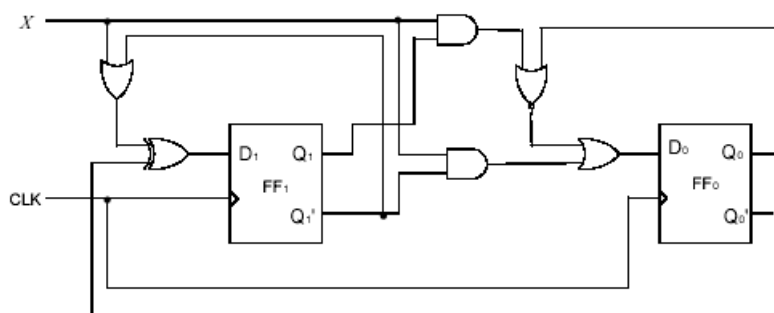


$$\begin{aligned}
 D_1 &= Q_0 Q_1 + Q_0 X + Q_0 \bar{Q}_1 \bar{X} = \\
 &= Q_0(Q_1 + X) + Q_0(\bar{Q}_1 + \bar{X}) = \\
 &= Q_0 \oplus (Q_1 + X)
 \end{aligned}$$

$$\begin{aligned}
 D_0 &= Q_0 \bar{Q}_1 + Q_0 \bar{X} + Q_1 X = \\
 &= Q_0(\bar{Q}_1 + \bar{X}) + Q_1 X = \\
 &= Q_0(\bar{Q}_1 X) + Q_1 X = \\
 &= (Q_0 + Q_1 X) + Q_1 \bar{X}
 \end{aligned}$$

Εικόνα: Πίνακες Karnaugh για τις συναρτήσεις εισόδου του προηγούμενου Πίνακα

Βήμα Σ3: Σχεδιάζουμε το λογικό κύκλωμα.



Εικόνα: Λογικό κύκλωμα του Παραδείγματος 6 με flip flops τύπου D

Παρατηρούμε ότι το λογικό κύκλωμα που προέκυψε χρειάζεται περισσότερες πύλες σε σχέση με το αντίστοιχο με flip flop τύπου JK.

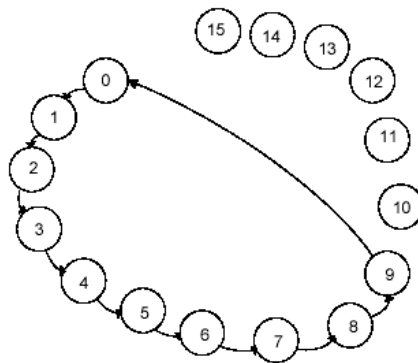
2.15.3 Σύγχρονα ακολουθιακά κυκλώματα με αδιάφορες καταστάσεις

Μέχρι τώρα ασχοληθήκαμε με τη σχεδίαση σύγχρονων ακολουθιακών κυκλωμάτων, τα οποία διατρέχουν όλες τις δυνατές καταστάσεις. Στη συνέχεια θα δούμε παραδείγματα περιπτώσεων στις οποίες κάποιες από τις δυνατές καταστάσεις του κυκλώματος δε χρησιμοποιούνται.

Παράδειγμα 7: Έστω ότι μας ζητείται να σχεδιαστεί σύγχρονο ακολουθιακό κύκλωμα το οποίο να διατρέχει τις τιμές 0 μέχρι και 9 χρησιμοποιώντας flip flops τύπου JK αρνητικής ακμής πυροδότησης.

Για τη δυαδική αναπαράσταση των δέκα τιμών απαιτούνται 4 δυαδικά ψηφία, δηλαδή 4 flip flops. Ακολουθώντας τη διαδικασία σχεδίασης, έχουμε:

Βήμα Σ1: Σχεδιάζουμε το διάγραμμα καταστάσεων και καταστρώνουμε τον αντίστοιχο πίνακα καταστάσεων.



Εικόνα: Διάγραμμα καταστάσεων του Παραδείγματος 7

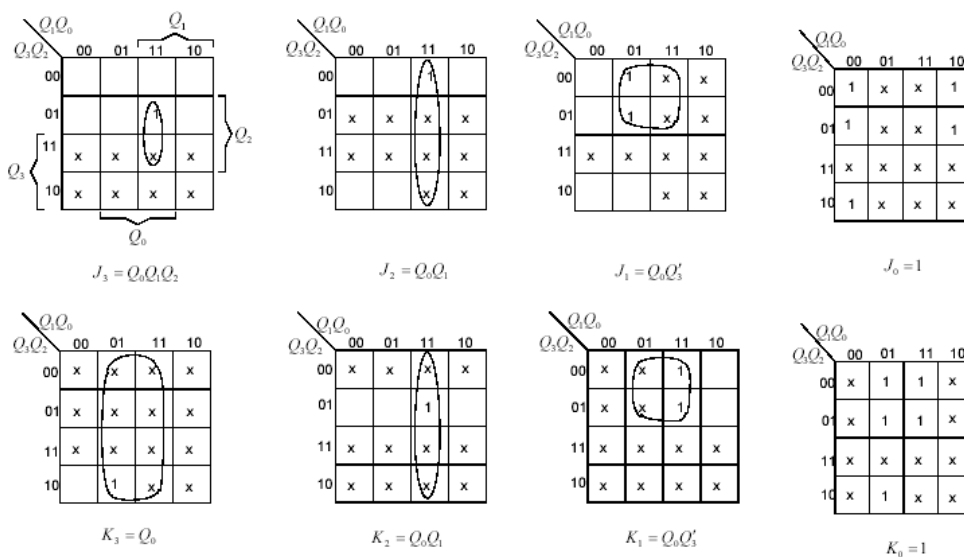
Το διάγραμμα καταστάσεων περιλαμβάνει $2^4 = 16$ καταστάσεις, από τις οποίες, όμως, μόνο οι δέκα χρησιμοποιούνται όπως φαίνεται στην προηγούμενη Εικόνα. Οι υπόλοιπες έξι (οι καταστάσεις 10, 11, 12, 13, 14, 15) είναι αδιάφορες. Το γεγονός αυτό αντικατοπτρίζεται στον επόμενο Πίνακα (πίνακα καταστάσεων) με το σύμβολο "X" στο τμήμα της επόμενης κατάστασης και στο τμήμα των εισόδων. Υπενθυμίζουμε ότι η ύπαρξη αυτών των συνθηκών αδιαφορίας θα βοηθήσει στην απλοποίηση των συναρτήσεων εισόδου. Όσον αφορά στη συμπλήρωση του πίνακα για τις καταστάσεις 0 μέχρι και 9, αυτή γίνεται κατά τα γνωστά με τη βοήθεια του πίνακα διέγερσης του flip flop τύπου JK.

Παρούσα κατάσταση				Επόμενη κατάσταση				Είσοδοι							
Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0	J3	K3	J2	K2	J1	K1	J0	K0
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	0	0	0	0	X	1	0	X	0	X	X	1
1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X

Πίνακας: Πίνακας καταστάσεων για το Παράδειγμα 7

Βήμα Σ2: Γράφουμε τις απλοποιημένες συναρτήσεις εισόδου.

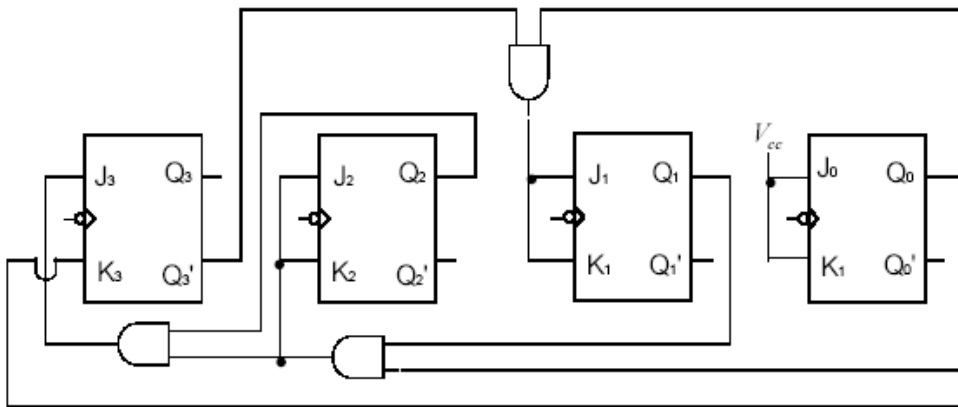
Η απλοποίηση των συναρτήσεων εισόδου των flip flops γίνεται με τη βοήθεια χαρτών Karnaugh όπως φαίνεται στην επόμενη εικόνα.



Εικόνα: Χάρτες Karnaugh για τις συναρτήσεις εισόδου του Παραδείγματος 7

Βήμα Σ3: Σχεδιάζουμε το λογικό κύκλωμα.

Έχοντας προσδιορίσει τις συναρτήσεις εισόδου των flip flops, προχωρούμε στη σχεδίαση του ζητούμενου κυκλώματος που φαίνεται στην επόμενη Εικόνα.



Εικόνα: Κύκλωμα για το Παράδειγμα 7

Στο σημείο αυτό η διαδικασία της σχεδίασης έχει ολοκληρωθεί. Υπάρχει, όμως, ένα θέμα το οποίο πρέπει να τεθεί και το οποίο σχετίζεται με το γεγονός ότι η αυθαίρετη αξιοποίηση των αδιάφορων καταστάσεων μπορεί να δημιουργήσει κάποια προβλήματα στο κύκλωμα. Για παράδειγμα, αν το κύκλωμα βρεθεί σε κάποια από τις αδιάφορες για εμάς καταστάσεις (11, 12, ..., 15) να παράγει ένα μη επιτρεπτό κύκλο απαρίθμησης, π.χ. 11, 12, 13, 14, 11, 12, 13, 14, 11, ...

Σε μια τέτοια περίπτωση το κύκλωμα δε θα φτάνει ποτέ σε μια από τις ζητούμενες καταστάσεις και θα εγκλωβίζεται σε ένα μη επιθυμητό βρόχο απαρίθμησης. Για να εξετάσουμε την περίπτωση αυτή, θα πρέπει να ελέγξουμε τη λειτουργία του κυκλώματος, δηλαδή να αναλύσουμε το κύκλωμα, προκειμένου να επιβεβαιώσουμε την ορθότητα της λειτουργίας του. Στην περίπτωση που έχουμε μη-επιθυμητό βρόχο απαρίθμησης, θα πρέπει να επαναλάβουμε τη διαδικασία της σχεδίασης αυτού, αρχίζοντας και πάλι από το βήμα Σ1, αλλά πιο αυστηροί στις προδιαγραφές μας.

Βήμα Σ4: Έλεγχος ορθής λειτουργίας του κυκλώματος.

Ο έλεγχος αυτός γίνεται, όπως αναφέραμε, με την ανάλυση του κυκλώματος που σχεδιάσαμε. Ακολουθούμε, επομένως, τα βήματα της ανάλυσης.

Βήμα Α1: Γράφουμε τις συναρτήσεις εισόδου των flip flops.

$$J_3 = Q_2 \cdot Q_1 \cdot Q_0$$

$$J_2 = K_2 = Q_1 \cdot Q_0$$

$$J_1 = K_1 = Q_3' \cdot Q_0 \quad J_0 = K_0 = 1$$

$$K_3 = Q_0$$

Βήμα Α2: Καταστρώνουμε τον πίνακα καταστάσεων.

Βασιζόμενοι στις συναρτήσεις εισόδου και στις τιμές της παρούσας κατάστασης προσδιορίζουμε την επόμενη κατάσταση του κυκλώματος, όπως φαίνεται στον επόμενο Πίνακα.

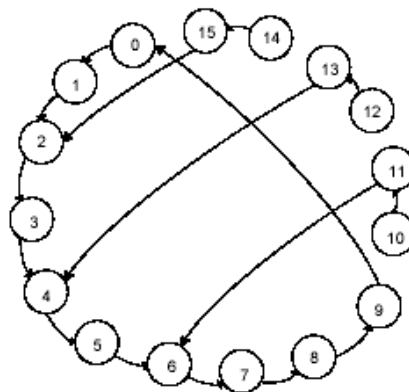
Παρούσα				Επόμενη				Είσοδοι							
Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0	J3	K3	J2	K2	J1	K1	J0	K0
...
1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	1
1	0	1	1	0	1	1	0	0	1	1	1	0	0	1	1
1	1	0	0	1	1	0	1	0	0	0	0	0	0	1	1
1	1	0	1	0	1	0	0	0	1	0	0	0	0	1	1
1	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1
1	1	1	1	0	0	1	0	1	1	1	1	0	0	1	1

Πίνακας: Πίνακας καταστάσεων για το κύκλωμα του Παραδείγματος 7

Για τις καταστάσεις 0 μέχρι και 9 ο πίνακας αυτός είναι ίδιος με τον προηγούμενο Πίνακα. Η διαφοροποίηση αρχίζει από την κατάσταση 10 και μετά. Το αντίστοιχο διάγραμμα καταστάσεων θα περιγράψει τη λειτουργία του κυκλώματος.

Βήμα A3: Σχεδιάζουμε το διάγραμμα καταστάσεων.

Το διάγραμμα καταστάσεων, που προκύπτει, φαίνεται στην επόμενη Εικόνα. Παρατηρούμε ότι το κύκλωμα, που σχεδιάσαμε, διατρέχει όντως τον κύκλο των καταστάσεων 0 μέχρι και 9.



Εικόνα: Διάγραμμα καταστάσεων του κυκλώματος του Παραδείγματος

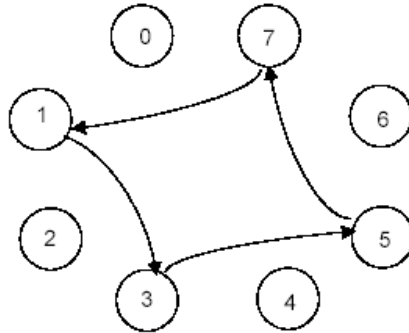
Αν το κύκλωμα βρεθεί σε μία από τις αδιάφορες καταστάσεις, τότε με τον πρώτο ωρολογιακό παλμό μεταβαίνει στην αμέσως επόμενη κατάσταση, ενώ με το δεύτερο παλμό μεταβαίνει σε μια από τις επιθυμητές (έγκυρες) καταστάσεις. Για παράδειγμα, αν βρεθεί στη κατάσταση 10, μεταβαίνει στην 11 και ακολούθως στην 6, οπότε και συνεχίζει.

Παράδειγμα 8: Σχεδιασμός κυκλώματος που να διατρέχει διαδοχικά τους αριθμούς 1, 3, 5, 7 χρησιμοποιώντας flip flops τύπου T.

Για την αναπαράσταση των καταστάσεων σε δυαδική μορφή χρειαζόμαστε 3 flip flops, για την αναπαράσταση των καταστάσεων 5 και 7. Με 3 flip flops έχουμε 8 δυνατές καταστάσεις, από τις οποίες όμως οι 4 μόνο μας χρειάζονται. Ακολουθώντας τη διαδικασία σχεδίασης έχουμε:

Βήμα Σ1: Σχεδιάζουμε το διάγραμμα καταστάσεων και τον αντίστοιχο πίνακα καταστάσεων.

Το διάγραμμα καταστάσεων φαίνεται στην επόμενη Εικόνα και ο αντίστοιχος πίνακας καταστάσεων στον επόμενο Πίνακα.



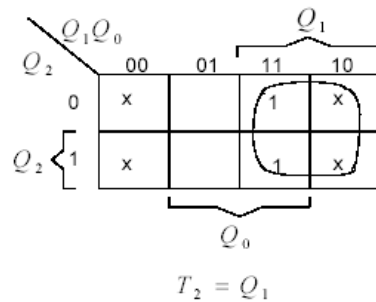
Εικόνα: Διάγραμμα καταστάσεων για το Παράδειγμα 6

Παρούσα κατάσταση			Επόμενη κατάσταση			Είσοδοι		
Q2	Q1	Q0	Q2	Q1	Q0	T2	T1	T0
0	0	0	X	X	X	X	X	X
0	0	1	0	1	1	0	1	0
0	1	0	X	X	X	X	X	X
0	1	1	1	0	1	1	1	0
1	0	0	X	X	X	X	X	X
1	0	1	1	1	1	0	1	0
1	1	0	X	X	X	X	X	X
1	1	1	0	0	1	1	1	0

Πίνακας: Πίνακας καταστάσεων για το Παράδειγμα 7

Βήμα Σ2: Προσδιορίζουμε τις απλοποιημένες συναρτήσεις εισόδου

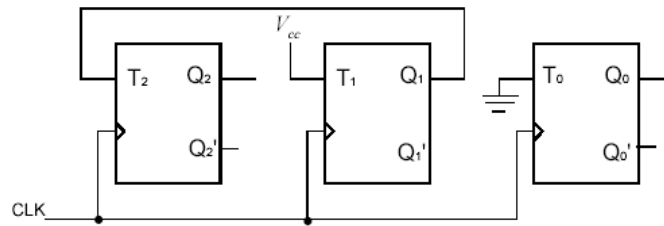
Από το τμήμα των εισόδων του πίνακα καταστάσεων και με αξιοποίηση των συνθηκών αδιαφορίας, εξάγεται ότι $T_0=0$, $T_1=1$. Η είσοδος T_2 προσδιορίζεται με βάση το χάρτη Karnaugh της επόμενης Εικόνας ίση με Q_1 , δηλαδή $T_2=Q_1$.



Εικόνα: Χάρτης Karnaugh για τη συνάρτηση εισόδου T_2

Βήμα Σ3: Σχεδιάζουμε το λογικό κύκλωμα.

Γνωρίζοντας τις συναρτήσεις εισόδου, σχεδιάζουμε το λογικό κύκλωμα (επόμενη Εικόνα).



Εικόνα: Κύκλωμα για το Παράδειγμα 7

Προκειμένου να ελέγξουμε τι θα συμβεί αν το κύκλωμα βρεθεί αρχικά σε μία από τις μη έγκυρες καταστάσεις 0, 2, 4, 6, συνεχίζουμε με την επαλήθευση της ορθής (επιθυμητής) λειτουργίας του κυκλώματος, που σχεδιάσαμε.

Βήμα Σ4: Έλεγχος ορθής λειτουργίας του κυκλώματος.

Ακολουθούμε τη διαδικασία ανάλυσης για το κύκλωμα που σχεδιάσαμε.

Βήμα Α1: Γράφουμε τις συναρτήσεις εισόδου.

$$T_2 = Q_1 \quad T_1 = 1 \quad T_0 = 0$$

Βήμα Α2: Καταστρώνουμε τον πίνακα καταστάσεων.

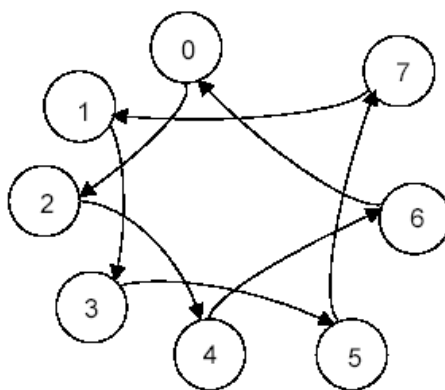
Ο πίνακας καταστάσεων προκύπτει εύκολα από την παρούσα κατάσταση και τις εισόδους σε συνδυασμό με τον πίνακα λειτουργίας του T FF.

Παρούσα κατάσταση			Είσοδοι			Επόμενη κατάσταση		
Q2	Q1	Q0	T2	T1	T0	Q2	Q1	Q0
0	0	0	0	1	0	0	1	0
0	0	1	0	1	0	0	1	1
0	1	0	1	1	0	1	0	0
0	1	1	1	1	0	1	0	1
1	0	0	0	1	0	1	1	0
1	0	1	0	1	0	1	1	1
1	1	0	1	1	0	0	0	0
1	1	1	1	1	0	0	0	1

Πίνακας: Πίνακας καταστάσεων για το κύκλωμα της προηγούμενης Εικόνα

Βήμα Α3: Σχεδιάζουμε το διάγραμμα καταστάσεων.

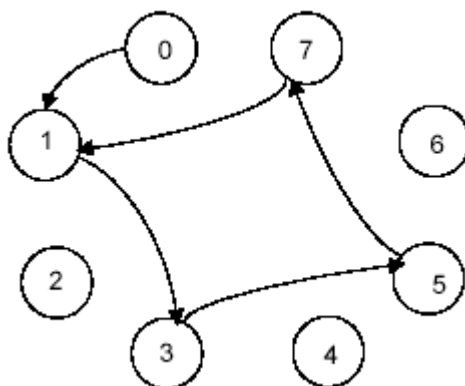
Από τον ανωτέρω Πίνακα προκύπτει το διάγραμμα καταστάσεων της επόμενης Εικόνας.



Εικόνα: Διάγραμμα καταστάσεων για τον προηγούμενο Πίνακα

Είναι σαφές ότι αν το κύκλωμα βρεθεί σε μία από τις μη έγκυρες καταστάσεις 0, 2, 4 ή 6, τότε εγκλωβίζεται στον κύκλο 0, 2, 4, 6, 0, 2, 4, 6, 0, 2, ... Για να λυθεί το πρόβλημα αυτό, θα πρέπει να παρέμβουμε στο βήμα Σ1 της σχεδίασης, αναγκάζοντας το κύκλωμα να μεταβεί σε μία από τις έγκυρες καταστάσεις. Θα μπορούσαμε, για παράδειγμα, να αναγκάσουμε το κύκλωμα μετά την κατάσταση 0 να μεταβεί στην κατάσταση 1. Με αυτό τον τρόπο "σπάμε" το μη έγκυρο κύκλο 0, 2, 4, 6, 0, 2, Έτσι έχουμε τα ακόλουθα βήματα:

Βήμα Σ1': Σχεδιάζουμε το διάγραμμα και τον πίνακα καταστάσεων, όπως φαίνεται στην επόμενη Εικόνα.

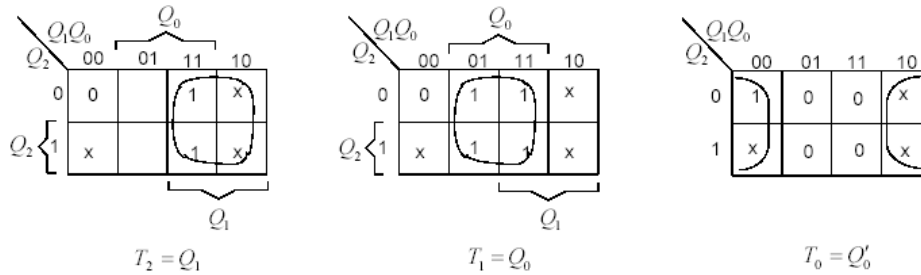


Εικόνα: Διορθωμένο διάγραμμα καταστάσεων

Παρούσα κατάσταση			Επόμενη κατάσταση			Είσοδοι		
Q2	Q1	Q0	Q2	Q1	Q0	T2	T1	T0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	1	0	0	1	1	0
0	1	1	1	0	1	1	1	0
1	0	0	1	1	0	0	1	0
1	0	1	1	1	1	0	1	0
1	1	0	0	0	0	1	1	0
1	1	1	0	0	1	1	1	0

Πίνακας: Διορθωμένος πίνακας καταστάσεων

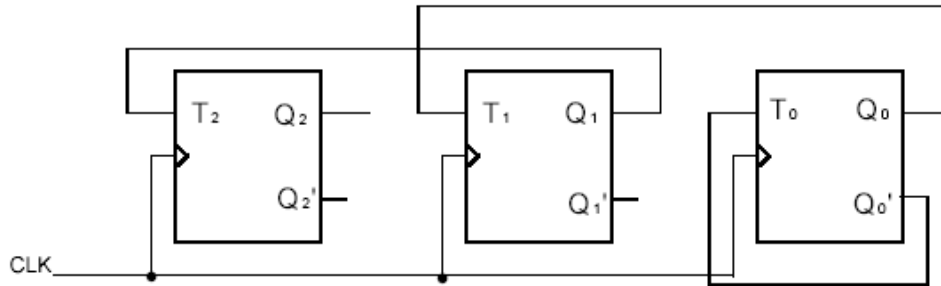
Βήμα Σ2': Προσδιορίζουμε τις απλοποιημένες συναρτήσεις εισόδου με τη βοήθεια χαρτών Karnaugh, όπως φαίνεται στην επόμενη Εικόνα.



Εικόνα: Χάρτες Karnaugh για τον Πίνακα καταστάσεων

Βήμα Σ3': Σχεδιάζουμε το λογικό κύκλωμα.

Στην επόμενη Εικόνα φαίνεται το λογικό κύκλωμα, το οποίο αντιστοιχεί στο διορθωμένο διάγραμμα καταστάσεων. Έτσι ολοκληρώνεται η σχεδίαση του ζητούμενου κυκλώματος.



Εικόνα: Κύκλωμα για το διάγραμμα καταστάσεων 10.27

Αξίζει να σημειωθεί ότι στην περίπτωση αυτή χρειάστηκε να επαναλάβουμε τη σχεδίαση δύο φορές. Αυτό συνέβη, γιατί οι συνθήκες αδιαφορίας, αν και βοήθησαν στη γρήγορη και εύκολη απλοποίηση των συναρτήσεων εισόδου, μας οδήγησαν σε ένα κλειστό κύκλο μη έγκυρων καταστάσεων, από τον οποίο δε μπορούσε να βγει το κύκλωμα.

Αξίζει ακόμη να σημειωθεί ότι αν τα flip flops που χρησιμοποιούμε διαθέτουν ασύγχρονες εισόδους clear ή preset, μπορούμε να αξιοποιήσουμε τις εισόδους αυτές, για να πετύχουμε την εκκίνηση του κυκλώματος από μια επιθυμητή κατάσταση.