



**ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΕΠΙΚΟΙΝΩΝΙΩΝ**

Ψηφιακά κυκλώματα

Σημειώσεις

**Αναστάσιος Ι. Μπαλουκτσής (Μηχανολόγος/Ηλεκτρολόγος Μηχανικός,
Μαθηματικός, PhD)**
Καθηγητής

Τομέας Αρχιτεκτονικής Η/Υ & Βιομηχανικών Εφαρμογών

Σέρρες 2004

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Πρόλογος	5
Συστήματα αριθμών	6
Δεκαδικό σύστημα	6
Παράδειγμα 1.1	6
Δυαδικό σύστημα	6
Μετατροπή δεκαδικού σε δυαδικό	8
Παράδειγμα 1.2	8
Παράδειγμα 1.3	9
Άλυτα προβλήματα	10
Βασικές λογικές πράξεις – λογικές πύλες	10
Ψηφιακή πύλη OR	11
Ψηφιακή πύλη AND	11
Ψηφιακή πύλη NOT	12
Ψηφιακή πύλη NAND (NOT AND)	12
Ψηφιακή πύλη NOR (NOT OR)	12
Ψηφιακή πύλη XOR	13
Ψηφιακή πύλη XNOR (NOT XOR)	13
Συνοπτικός πίνακας λογικών πυλών	14
Δυνατοί πίνακες αληθείας στο δυαδικό σύστημα	15
Άλλοι τρόποι δυαδικής κωδικοποίησης	15
Κωδικοποίηση BCD (Binary Coded Decimal)	16
Παράδειγμα 1.4	16
Μετατροπή από BCD σε δεκαδικό	16
Παράδειγμα 1.5	16
Κώδικας Gray	17
Κώδικες με ανίχνευση σφάλματος	18
Κώδικας περιττής ισοτιμίας	18
Κώδικας άρτιας ισοτιμίας	19
Ορισμός των λογικών επιπέδων	20
Άλγεβρα Boole	21
Ιδιότητες και κανόνες της άλγεβρας Boole	21
Λογικές πράξεις με σταθερές	21
Λογικές πράξεις με μια μεταβλητή	21
Παράδειγμα 1.6	22
Λογικές πράξεις-ιδιότητες με δυο ή περισσότερες μεταβλητές	22
Παράδειγμα 1.7	23
Παράδειγμα 1.8	23
Διαδικασία σχεδίασης ψηφιακής λογικής συνάρτησης	24
Κανονικές μορφές λογικών συναρτήσεων	25
Κανονική μορφή αθροίσματος	25
Παράδειγμα 1.9	25
Παράδειγμα 1.10	26
Σύντομη γραφή για την κανονική μορφή αθροίσματος	26
Παράδειγμα 1.11	27
Παράδειγμα 1.12	27
Ημιαθροιστής	29
Κανονική μορφή γινομένου	30
Σύντομη γραφή για την κανονική μορφή γινομένου	31
Παράδειγμα 1.13	32

Σύνθεση ψηφιακού κυκλώματος	33
Σύνθεση ψηφιακών κυκλωμάτων με πύλες NAND	34
Παράδειγμα 1.14	34
Παράδειγμα 1.15	35
Παράδειγμα 1.16	35
Αντικατάσταση πυλών με πύλες NAND	36
Σύνθεση ψηφιακών κυκλωμάτων με πύλες NOR	37
Αντικατάσταση πυλών με πύλες NOR	37
Ελαχιστοποίηση λογικών συναρτήσεων με τη χρήση των πινάκων Karnaugh	38
Πίνακες Karnaugh	38
Παράδειγμα 1.17	40
Παράδειγμα 1.18	40
Παράδειγμα 1.19	41
Παράδειγμα 1.20	41
Ύπαρξη αδιάφορων περιπτώσεων	42
Πλήρης Αθροιστής	43
Σπινθήρες	45
Παράδειγμα 1.21	47
Ασκήσεις επανάληψης 1	49
Άσκηση 1.1	49
Άσκηση 1.2	49
Άσκηση 1.3	50
Άσκηση 1.4	50
Άσκηση 1.5	50
Άσκηση 1.6	51
Άσκηση 1.7	52
Άσκηση 1.8	52
Άσκηση 1.9	53
Άσκηση 1.10	54
Κυκλώματα ακολουθιακής λογικής	57
Γενικές μορφές κυκλωμάτων	57
Flip – Flops	58
To SR (Set-Reset) flip-flop (ff)	59
To Clocked SR - ff	61
Flip – Flop τύπου D	61
Flip – Flop τύπου T (Toggle ff)	62
Flip – Flop τύπου JK	62
Υλοποίηση σύγχρονων flip-flops με όρους SR-ff	62
Υλοποίηση σύγχρονων flip-flops με όρους SR-ff	63
Παράδειγμα 2.1:	63
Προβλήματα που σχετίζονται με απλά σύγχρονα ff	64
1. Αναπήδηση εισόδου	64
2. Κακή λειτουργία κυκλωμάτων που χρησιμοποιούν διαδοχικά ff	65
3. Ταλαντώσεις σε ff λόγω ανάδρασης	65
Μέθοδοι επίλυσης των προβλημάτων	66
Διάταξη “Αφέντη – Σκλάβου” ff	66
Δημιουργία εσωτερικού ρολογιού για παραγωγή παλμού βραχείας διάρκειας	66
Εφαρμογές ff	67
Απλοί καταχωρητές	68
Καταχωρητές ολίσθησης	68
Κυκλώματα μετρητών	69
Ασύγχρονοι μετρητές	70
Σύγχρονοι μετρητές	72

Βασικοί ορισμοί για τους μετρητές	73
Τροποποίηση του βασικού σύγχρονου μετρητή για τη δημιουργία ενός MOD-M μετρητή	73
Εισαγωγή στο σχεδιασμό ψηφιακών κυκλωμάτων με διαγράμματα καταστάσεων (state diagrams)	74
Διαγράμματα καταστάσεων	74
Παράδειγμα 2.2	75
Πίνακες Καταστάσεων	76
Παράδειγμα 2.3	76
Πίνακας διέγερσης και εξισώσεις διέγερσης	77
Παράδειγμα 2.4	77
Εξισώσεις διέγερσης	78
Παράδειγμα 2.5	79
Πρόβλημα 2.1	82
Προβλήματα από καταστάσεις που δεν χρησιμοποιούνται	85
Παράδειγμα 2.6	86
Ασκήσεις επανάληψης 2	88
Άσκηση 2.1	88
Άσκηση 2.2	89
Άσκηση 2.3	91
Άσκηση 2.4	92
Άσκηση 2.5	95

Πρόλογος

Οι σημειώσεις αυτές αποτελούν βασικό διδακτικό βοήθημα για το μάθημα «Ψηφιακά Κυκλώματα» που διδάσκεται στο Τμήμα Πληροφορικής & Επικοινωνιών του ΤΕΙ Σερρών στο 3^ο εξάμηνο σπουδών.

Η ανάπτυξη των ψηφιακών ηλεκτρονικών, που πραγματοποιήθηκε τα τελευταία 30 χρόνια, αποτελεί γεγονός, για το οποίο δεν υπάρχει κάτι ανάλογο σε κανέναν άλλο κλάδο της επιστήμης του μηχανικού. Μάλιστα κατά τη ραγδαία αυτή ανάπτυξη, ενώ το πραγματικό κόστος των ψηφιακών ολοκληρωμένων κυκλωμάτων (hardware) μειώνονταν στο μισό κάθε έτος, η σύνθεση και η πολυπλοκότητά τους τετραπλασιάζονταν κάθε τρία έτη.

Έχοντας υπόψη το παραπάνω σκηνικό της ταχύτατης ανάπτυξης, οι σημειώσεις δομήθηκαν με κατεύθυνση να παρέχουν στους φοιτητές τις αναγκαίες θεμελιώδεις έννοιες της ψηφιακής λογικής και ταυτόχρονα να τους εξοικειώνουν με μεθόδους σχεδιασμού και τεχνικές στο επίπεδο του συστήματος. Επίσης, στις σημειώσεις περιλαμβάνονται ασκήσεις για τους φοιτητές με τις λύσεις τους, ώστε να αναδεικνύεται η πρακτική χρησιμότητα των διαφόρων εννοιών που εισάγονται στη θεωρία, αλλά και ο σωστός τρόπος εφαρμογής των.

Οπωσδήποτε, για περισσότερη εμβάθυνση, απαιτείται η χρήση πρόσθετης βιβλιογραφίας, η οποία διατίθεται στη βιβλιοθήκη του ΤΕΙ Σερρών.

Αναστάσιος Μπαλουκτσής

Συστήματα αριθμών

Ένα σύστημα αριθμών χρησιμοποιεί ένα σύνολο συμβόλων γνωστό ως ψηφία. Υπάρχουν διάφορα συστήματα αριθμών όπως το δεκαδικό, το δυαδικό κ.λ.π.

Δεκαδικό σύστημα

Στο δεκαδικό σύστημα χρησιμοποιούνται δέκα ψηφία 0, 1, 2, 3, 4, 5, 6, 7, 8, & 9, ενώ το 10 ορίζεται ως βάση του συστήματος.

Παράδειγμα 1.1:

$$503,14 =$$

$$500 + 0 + 3 + \frac{1}{10} + \frac{4}{100} =$$

$$5 \cdot 10^2 + 0 \cdot 10^1 + 3 \cdot 10^0 + 1 \cdot 10^{-1} + 4 \cdot 10^{-2}$$

Η γενική μορφή της απεικόνισης στο δεκαδικό σύστημα είναι:

$$D_{10} = (d_n 10^n + d_{n-1} 10^{n-1} + \dots d_1 10^1 + d_0 10^0 + d_{-1} 10^{-1} + \dots d_{-n} 10^{-n})$$

ή ο αριθμός παριστάνεται ως

$$(d_n d_{n-1} \dots d_1 d_0 d_{-1} d_{-2} \dots d_{-n})$$

όπου $d_i \in (0, 1, \dots, 9)$ είναι οι συντελεστές των αντίστοιχων δυνάμεων του 10.

Δυαδικό σύστημα

Γενικά οι αριθμοί μπορεί να έχουν βάσεις διάφορες του 10, για παράδειγμα:

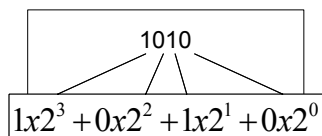
βάση 16, δεκαεξαδικό σύστημα,

βάση 8, οκταδικό σύστημα, ή

βάση 2, δυαδικό σύστημα.

Στο δυαδικό σύστημα που έχει βάση το 2 υπάρχουν δύο ψηφία, το 0 και το 1.

Παράδειγμα δυαδικού αριθμού:



ο αντίστοιχος δεκαδικός του 1010 είναι ο $8+0+2+0=10$

Η μορφή της γενικής παράστασης στο δυαδικό σύστημα είναι:

$$B_2 = b_n 2^n + b_{n-1} 2^{n-1} + \dots b_1 2^1 + b_0 2^0 + b_{-1} 2^{-1} + \dots b_{-m} 2^{-m}$$

ή ο αριθμός παριστάνεται ως

$$(b_n b_{n-1} \dots b_1 b_0 b_{-1} b_{-2} \dots b_{-m})_b$$

όπου b_i (0 ή 1) είναι δυαδικά ψηφία (bits) που παριστάνουν τους συντελεστές των αντίστοιχων δυνάμεων του 2.

Για παράδειγμα οι ακέραιοι δυαδικοί αριθμοί με 4 ψηφία είναι της μορφής:

$$b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0$$

Ο μεγαλύτερος αριθμός με 4 ψηφία είναι ο 1111 ο οποίος είναι ισοδύναμος με τον δεκαδικό αριθμό 15.

$$\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 2^3 & + & 2^2 & + & 2^1 & + & 2^0 \\ 8 & + & 4 & + & 2 & + & 1 = 15 \end{array}$$

Γενικά ένας δυαδικός αριθμός με n ψηφία μπορεί να παραστήσει ένα εύρος από 2^n δεκαδικούς αριθμούς:

1 ψηφίο 0 και 1

2 ψηφία 0 - 3

3 ψηφία 0 - 7

4 ψηφία 0 - 15

5 ψηφία 0 - 31 κ.λ.π.

Μετατροπή δεκαδικού σε δυαδικό

Για τη μετατροπή ενός ακέραιου δεκαδικού σε δυαδικό χρησιμοποιείται η διαδικασία της διαδοχικής διαίρεσης ως εξής:

Γενική μορφή ενός ακεραίου δυαδικού είναι: $(b_n b_{n-1} \dots b_1 b_0)_2$

και ο αντίστοιχος δεκαδικός του: $D = b_n 2^n + b_{n-1} 2^{n-1} + \dots b_1 2^1 + b_0 2^0$

συνεπώς παρατηρούμε ότι διαιρώντας τον D με το 2 προκύπτει ως πηλίκο το

$$b_n 2^{n-1} + b_{n-1} 2^{n-2} + \dots b_1 2^0$$

και ως υπόλοιπο το b_0

Κατόπιν διαιρώντας το πηλίκο με 2 θα προκύψει ως νέο πηλίκο το

$$b_n 2^{n-2} + b_{n-1} 2^{n-3} + \dots b_2 2^0$$

και υπόλοιπο b_1

Επαναλαμβάνεται η διαδικασία μέχρι να προκύψει πηλίκο μηδέν. Τα υπόλοιπα των διαιρέσεων είναι ουσιαστικά τα ψηφία του δυαδικού αριθμού.

Παράδειγμα 1.2:

Να μετατραπεί ο ακέραιος 19_{10} στον αντίστοιχο δυαδικό B_2

Απάντηση

1^η διαίρεση με το 2;

$$D/2=19/2=\text{πηλίκο } 9 \text{ και υπόλοιπο } 1 \quad \text{άρα} \quad b_0=1$$

2^η διαίρεση με το 2;

$$9/2=\text{πηλίκο } 4 \text{ και υπόλοιπο } 1 \quad \text{άρα} \quad b_1=1$$

3^η διαίρεση με το 2;

$$4/2=\text{πηλίκο } 2 \text{ και υπόλοιπο } 0 \quad \text{άρα} \quad b_2=0$$

4^η διαίρεση με το 2;

$$2/2=\text{πηλίκο } 1 \text{ και υπόλοιπο } 0 \quad \text{άρα} \quad b_3=0$$

5^η διαίρεση με το 2;

$$1/2=\text{πηλίκο } 0 \text{ και υπόλοιπο } 1 \quad \text{άρα} \quad b_4=1$$

συνεπώς ο αντίστοιχος δυαδικός αριθμός είναι:

$$B_2=10011=19_{10}$$

Για τη μετατροπή του κλασματικού μέρους ενός δεκαδικού αριθμού στο αντίστοιχο κλασματικό μέρος του δυαδικού χρησιμοποιείται η διαδικασία των διαδοχικών πολλαπλασιασμών ως εξής:

Γενική μορφή του κλασματικού μέρους ενός δυαδικού είναι: $(b_{-1}b_{-2}...b_{-m})_b$ και το αντίστοιχο κλασματικό μέρος του δεκαδικού είναι:

$$D = b_{-1}2^{-1} + b_{-2}2^{-2} + ...b_{-m}2^{-m}$$

συνεπώς παρατηρούμε ότι πολλαπλασιάζοντας τον D με το 2 προκύπτει;

$$b_{-1} + b_{-2}2^{-1} + ...b_{-m}2^{-m+1}$$

άρα το ακέραιο μέρος του νέου αριθμού (0 ή 1) είναι το ψηφίο b_{-1} και το κλασματικό του μέρος το

$$b_{-2}2^{-1} + ...b_{-m}2^{-m+1}$$

Κατόπιν πολλαπλασιάζουμε το νέο κλασματικό μέρος με 2 οπότε προκύπτει το ψηφίο b_{-2} . Επαναλαμβάνεται η διαδικασία μέχρι να προκύψει κλασματικό μέρος μηδέν ή να επιτευχθεί η επιθυμητή ακρίβεια.

Παράδειγμα 1.3:

Να μετατραπεί ο δεκαδικός 28.375 στον αντίστοιχο δυαδικό

Απάντηση

Πρώτα υπολογίζεται το ακέραιο μέρος:

1^η διαίρεση με το 2:

$$D/2=28/2= \text{πηλίκο } 14 \text{ και υπόλοιπο } 0 \quad \text{άρα} \quad b_0=0$$

2^η διαίρεση με το 2:

$$14/2= \text{πηλίκο } 7 \text{ και υπόλοιπο } 0 \quad \text{άρα} \quad b_1=0$$

3^η διαίρεση με το 2:

$$7/2= \text{πηλίκο } 3 \text{ και υπόλοιπο } 1 \quad \text{άρα} \quad b_2=1$$

4^η διαίρεση με το 2:

$$3/2 = \text{πηλίκο } 1 \text{ και υπόλοιπο } 1 \quad \text{άρα} \quad b_3 = 1$$

5^η διαίρεση με το 2:

$$1/2 = \text{πηλίκο } 0 \text{ και υπόλοιπο } 1 \quad \text{άρα} \quad b_4 = 1$$

Συνεπώς το ακέραιο μέρος του δυαδικού είναι: 11100_2

Κατόπιν υπολογίζεται το κλασματικό μέρος του δυαδικού:

$$0,375 \times 2 = 0,75 \text{ με ακέραιο μέρος } 0 \text{ και κλασματικό } 0,75$$

$$0,75 \times 2 = 1,50 \text{ με ακέραιο μέρος } 1 \text{ και κλασματικό } 0,50$$

$$0,50 \times 2 = 1,00 \text{ με ακέραιο μέρος } 1 \text{ και κλασματικό } 0$$

Συνεπώς το κλασματικό μέρος του δυαδικού είναι: $,011_2$

Ο αντίστοιχος δυαδικός είναι: $11100,011_2$

Άλυτα προβλήματα:

(1) Να μετατραπούν οι δεκαδικοί αριθμοί στους αντίστοιχους δυαδικούς:

(α) 7_{10}

(β) 39_{10}

(γ) 75_{10}

(δ) 130_{10}

(ε) $0,875_{10}$

(2) Να μετατραπούν οι δυαδικοί αριθμοί στους αντίστοιχους δεκαδικούς:

(α) 00101_2

(β) 101011_2

(γ) $0,1101_2$

(δ) $111101,101111_2$

Βασικές λογικές πράξεις – λογικές πύλες

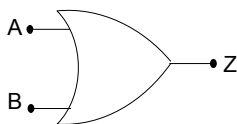
Μία λογική πράξη μεταξύ μεταβλητών είναι μία συνάρτηση που ορίζεται από έναν πίνακα αληθείας (truth table). Το ηλεκτρικό κύκλωμα που εκτελεί μία λογική πράξη ονομάζεται **λογική ή ψηφιακή πύλη** και παριστάνεται από ένα σύμβολο. Τα δυαδικά ψηφία 1 και 0, που ουσιαστικά παριστάνουν τις δύο καταστάσεις αληθής (true), ψευδής (false), στη

φυσική τους υπόσταση είναι δυο διακριτά επίπεδα ηλεκτρικής τάσης (συνήθως στην ιδανική περίπτωση 5V και 0V).

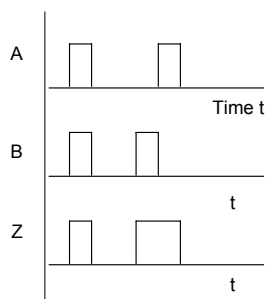
Ψηφιακή πύλη OR

Η έξοδος είναι αληθής (true) (1), εάν μια από τις εισόδους ή και οι δυο είναι αληθείς (1)

Σύμβολο



$$Z = A + B$$



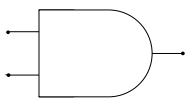
Πίνακας αληθείας

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

Ψηφιακή πύλη AND

Η έξοδος είναι αληθής (1), όταν και οι δυο εισοδοι είναι αληθείς (1)

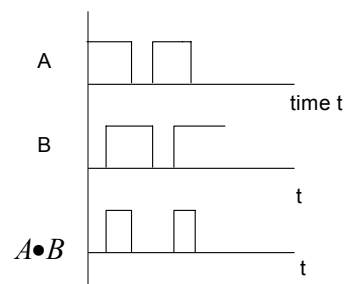
Σύμβολο



$$Z = A \cdot B$$

Πίνακας αληθείας

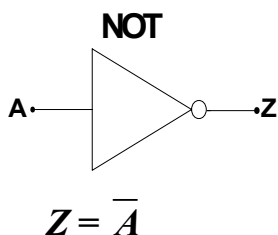
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1



Ψηφιακή πύλη NOT

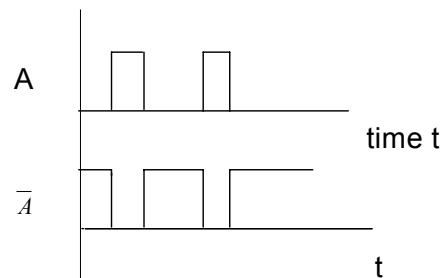
Δημιουργεί αντιστροφή του σήματος εισόδου

Σύμβολο



Πίνακας αληθείας

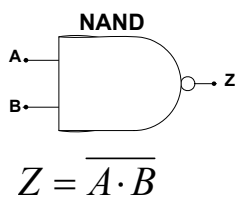
A	$\overline{A}$
0	1
1	0



Ψηφιακή πύλη NAND (NOT AND)

Η έξοδος είναι ψευδής (0) μόνο όταν A και B είναι αληθείς (1)

Σύμβολο



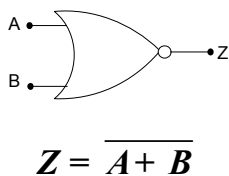
Πίνακας αληθείας

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

Ψηφιακή πύλη NOR (NOT OR)

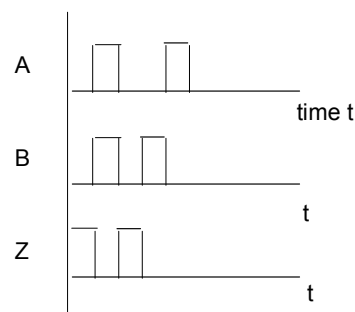
Η έξοδος είναι αληθής (1), όταν και οι δύο είσοδοι είναι ψευδείς (0)

Σύμβολο



Πίνακας αληθείας

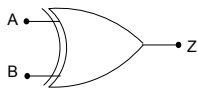
A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0



Ψηφιακή πύλη XOR

Η έξοδος είναι αληθής (1), όταν ή μία εκ των δύο εισόδων είναι αληθής (1), αλλά όχι και οι δύο ταυτόχρονα:

Σύμβολο



$$Z = A \oplus B$$

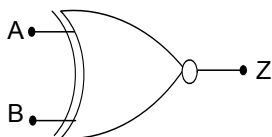
Πίνακας αληθείας

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

Ψηφιακή πύλη XNOR (NOT XOR)

Η έξοδος είναι αληθής (1) όταν και οι δυο είσοδοι είναι ψευδείς (0), ή και οι δυο είναι αληθείς (1)

Σύμβολο

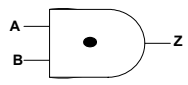
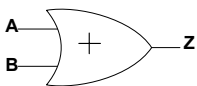
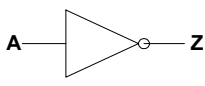
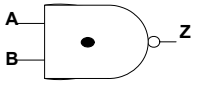
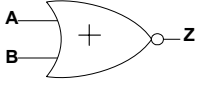
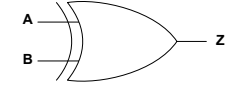
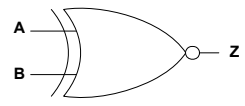


$$Z = \overline{A \oplus B}$$

Πίνακας αληθείας

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

Συνοπτικός πίνακας λογικών πυλών

Ονομασία	Σύμβολο	Σχέση	Πίνακας αληθείας		
			A	B	Z
AND		$Z = A \bullet B$	0	0	0
			0	1	0
			1	0	0
			1	1	1
OR		$Z = A + B$	0	0	0
			0	1	1
			1	0	1
			1	1	1
NOT		$Z = \bar{A}$	0		1
			1		0
NAND		$Z = \overline{A \bullet B}$	0	0	1
			0	1	1
			1	0	1
			1	1	0
NOR		$Z = \overline{A + B}$	0	0	1
			0	1	0
			1	0	0
			1	1	0
XOR		$Z = A \oplus B$	0	0	0
			0	1	1
			1	0	1
			1	1	0
XNOR		$Z = \overline{A \oplus B}$	0	0	1
			0	1	0
			1	0	0
			1	1	1

Δυνατοί πίνακες αληθείας στο δυαδικό σύστημα

Ένας πίνακας αληθείας παριστάνει τη συνάρτηση μεταξύ των εισόδων και της εξόδου ενός λογικού συστήματος. Για δυο εισόδους (*F*, *T*) υπάρχουν τέσσερις πιθανοί συνδυασμοί πραγματικών τιμών:

FF, FT, TF, TT

Επειδή κάθε δυνατή είσοδος μπορεί να δώσει δύο διαφορετικές εξόδους (0, 1) συνεπάγεται ότι οι δυνατοί πίνακες αληθείας για ένα λογικό σύστημα δύο εισόδων είναι:

$$2^4 = 16$$

Όλοι οι πίνακες αληθείας για δύο εισόδους A, B και μία έξοδο Z

A	τιμές εισόδου				Συνάρτηση (έξοδος Z)	Σύμβολο
	F	F	T	T		
B	F	T	F	T		
0	F	F	F	F	πάντοτε 0	0
1	F	F	F	T	AND	$A \cdot B$
2	F	F	T	F	-	-
3	F	F	T	T	είσοδος A	A
4	F	T	F	F	-	-
5	F	T	F	T	είσοδος B	B
6	F	T	T	F	XOR	$A \oplus B$
7	F	T	T	T	OR	$A + B$
8	T	F	F	F	NOR	$\overline{A + B}$
9	T	F	F	T	XNOR	$\overline{A \oplus B}$
10	T	F	T	F	Not B	$\overline{B}$
11	T	F	T	T	-	-
12	T	T	F	F	Not A	$\overline{A}$
13	T	T	F	T	-	-
14	T	T	T	F	NAND	$\overline{A \cdot B}$
15	T	T	T	T	πάντοτε 1	1

Άλλοι τρόποι δυαδικής κωδικοποίησης

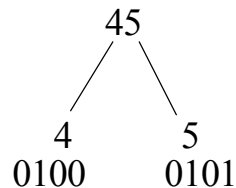
Εκτός από την κανονική δυαδική κωδικοποίηση υπάρχουν κι' άλλοι τρόποι δυαδικής κωδικοποίησης οι οποίοι χρησιμοποιούνται σε διάφορες περιπτώσεις.

Κωδικοποίηση BCD (Binary Coded Decimal)

Η κωδικοποίηση καθιστά δυνατή την απλή μετατροπή μεταξύ δυαδικού και δεκαδικού αριθμού. Κάθε ψηφίο ενός δεκαδικού αριθμού αντικαθίσταται από 4 bits του αντίστοιχου δυαδικού του.

Παράδειγμα 1.4:

Μετατροπή του 45_{10} σε BCD



Επομένως $45_{10} = 01000101$

Μετατροπή από BCD σε δεκαδικό

Η δυαδική λέξη χωρίζεται σε ομάδες των 4bits ξεκινώντας από το λιγότερο σημαντικό ψηφίο. Κατόπιν η κάθε ομάδα μετατρέπεται στον αντίστοιχο δεκαδικό.

Παράδειγμα 1.5:

Μετατροπή 1010011 σε δεκαδικό

Απάντηση:

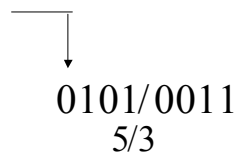
Πρόσθεση μηδενικού

Χωρισμός σε ομάδες των 4

και μετατροπή της κάθε

ομάδας στον αντίστοιχο δεκαδικό

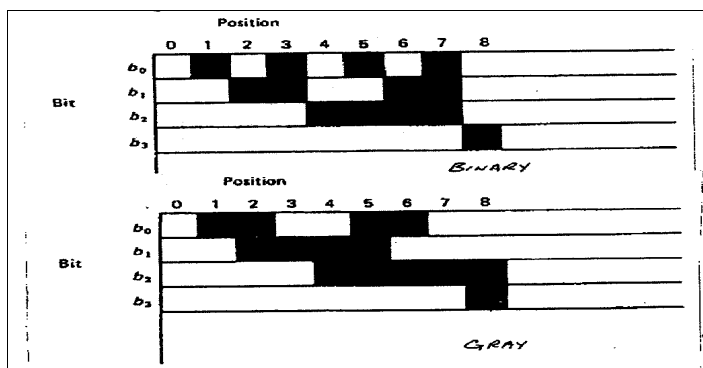
τελικό αποτέλεσμα $01010011_{BCD} = 53_{10}$



Κώδικας Gray

Συχνά χρησιμοποιείται σε ηλεκτρονικά κυκλώματα για την αποφυγή προβλημάτων που θα μπορούσαν να προκύψουν εάν χρησιμοποιούνταν η απευθείας δυαδική κωδικοποίηση.

Για παράδειγμα, σε μετρήσεις της θέσης ενός αντικειμένου, θα μπορούσε να φαίνεται ότι γειτονικές θέσεις του αντικειμένου διαφέρουν περισσότερο από ένα bit, εάν χρησιμοποιηθεί η απευθείας δυαδική κωδικοποίηση.



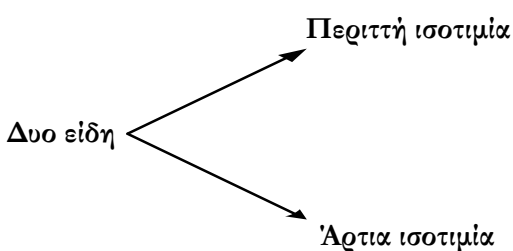
Decimal	Gray Code	
0	0	
1	1	
2	11	Set bit 1. Reflect bit 0
3	10	Set bit 2. Reflect bits 1 and 0
4	110	
5	111	
6	101	
7	100	
8	1100	Set bit 3. Reflect bits 2, 1 and 0
9	1101	
10	1111	
11	1110	
12	1010	
13	1011	
14	1001	
15	1000	
16	11000	Set bit 4. Reflect bits 3, 2, 1 and 0

Imagine a mirror is placed under the reflected bits.

Κώδικες με ανίχνευση σφάλματος

Στα ψηφιακά συστήματα, υπάρχουν περιπτώσεις όπου κατά την παραγωγή δεδομένων και την επεξεργασία αυτών, εμφανίζονται σφάλματα. Για παράδειγμα κάποιο ψηφίο 1, ενός συνόλου δυαδικών ψηφίων, μπορεί να μετατραπεί σε ψηφίο 0, είτε κατά το στάδιο της μετάδοσης, είτε γιατί το ψηφιακό σύστημα δεν λειτούργησε σωστά. Μία απλή μέθοδος, ανίχνευσης του σφάλματος, είναι η χρήση του κώδικα ανίχνευσης λάθους, η οποία χρησιμοποιεί ένα επιπλέον ψηφίο ισοτιμίας (parity bit).

Κώδικες ισοτιμίας



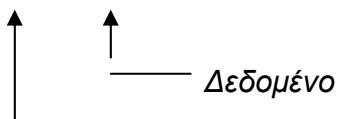
Δημιουργία επιπλέον ψηφίου ισοτιμίας πριν τη μετάδοση της λέξης δεδομένων

Κώδικας περιττής ισοτιμίας

Το ψηφίο ισοτιμίας είναι 0 αν το σύνολο των ψηφίων 1 είναι περιττό. Το ψηφίο ισοτιμίας είναι 1 αν το σύνολο των ψηφίων 1 είναι άρτιο.

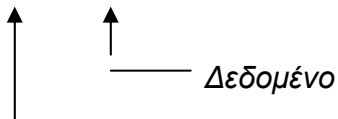
Για παράδειγμα η δυαδική λέξη **010001** έχει αριθμό ψηφίων '1' άρτιο, συνεπώς θα μεταδοθεί με ψηφίο ισοτιμίας '1', είτε:

1 010001



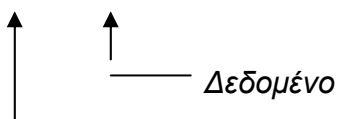
ψηφίο ισοτιμίας

παρομοίως η δυαδική λέξη **01000** έχει αριθμό ψηφίων '1' περιττό, συνεπώς θα μεταδοθεί με ψηφίο ισοτιμίας '0', είτε:

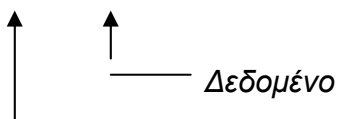
0 010000*ψηφίο ισοτιμίας****Κώδικας άρτιας ισοτιμίας***

Αντίστροφος της περιττής ισοτιμίας. Το ψηφίο ισοτιμίας είναι 1 αν το σύνολο των '1' είναι περιττό. Το ψηφίο ισοτιμίας είναι 0 αν το σύνολο των '1' είναι άρτιο

Για παράδειγμα η δυαδική λέξη **10110** έχει αριθμό ψηφίων '1' περιττό, συνεπώς θα μεταδοθεί με ψηφίο ισοτιμίας '1', είτε:

1 10110*ψηφίο ισοτιμίας*

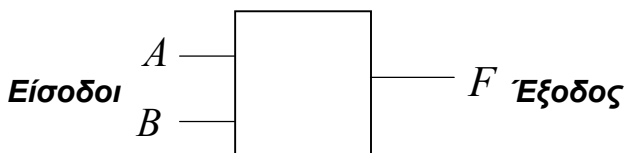
Υποθέστε ότι κατά τη μετάδοση δημιουργείται σφάλμα και η λέξη αλλάζει στην **11110**, οπότε αυτό που λαμβάνεται είναι το

1 11110*ψηφίο ισοτιμίας*

Ένας έλεγχος ισοτιμίας θα έδινε ψηφίο ισοτιμίας '0', αλλά το ψηφίο ισοτιμίας που λαμβάνεται είναι '1'. Η ασυμφωνία δηλώνει ότι υπάρχει σφάλμα, αλλά όχι τη θέση του.

Ορισμός των λογικών επιπέδων

Για ένα ψηφιακό κύκλωμα θεωρούμε ότι οι επιτρεπτές καταστάσεις τάσεως είναι 0V & 5V.



Υπάρχουν δύο είδη ορισμού των λογικών επιπέδων. Ο ένας είναι να ορίσουμε 0 Volts= 0 και 5Volts= 1, οπότε έχουμε τη θετική κωδικοποίηση και ο δεύτερος τα 0Volts= 1 και τα 5Volts= 0, οπότε έχουμε την αρνητική κωδικοποίηση.

Έστω ότι ο πίνακας αληθείας για τη θετική κωδικοποίηση του παραπάνω κυκλώματος είναι:

A	B	F
0V	0V	0V
0V	5V	0V
5V	0V	0V
5V	5V	5V

Δηλαδή εκτελεί τη λογική συνάρτηση **AND**.

Εάν θεωρήσουμε την αρνητική κωδικοποίηση τότε θα έχουμε:

A	B	F
0V	0V	0V
0V	5V	0V
5V	0V	0V
5V	5V	5V

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

Δηλαδή εκτελεί τη λογική συνάρτηση **OR**

Συνεπώς το ίδιο ηλεκτρονικό σύστημα μπορεί να παρουσιάσει την **AND** ή **OR**

συμπεριφορά ανάλογα με την πολικότητα της κωδικοποίησης.

Είναι αξιοπρόσεκτο ότι ένα κύκλωμα το οποίο παρουσιάζει την **AND** συμπεριφορά στην θετική κωδικοποίηση γίνεται πύλη **OR** στην αρνητική κωδικοποίηση και όχι πύλη **NAND**. Οι περισσότεροι κατασκευαστές χρησιμοποιούν τη θετική κωδικοποίηση.

Άλγεβρα Boole

Οι αρχές της λογικής αναπτύχθηκαν από τον George Boole (1815-1884) και τον Augustus De Morgan. Εκατό χρόνια αργότερα ο Claude Shannon (ως μεταπτυχιακός φοιτητής στο MIT) έδειξε ότι η άλγεβρα Boole ήταν σχετική με την ανάλυση διακοπτικών (switching) κυκλωμάτων. Η άλγεβρα Boole αποτελεί τη μαθηματική βάση για την ηλεκτρονική επεξεργασία της δυαδικής πληροφορίας.

Ιδιότητες και κανόνες της άλγεβρας Boole

Οι ιδιότητες και οι κανόνες της Άλγεβρας Boole εφαρμόζονται και ισχύουν σε τρεις κύριες ομάδες πράξεων.

1. Λογικές πράξεις με σταθερές.
2. Λογικές πράξεις με μια μεταβλητή.
3. Λογικές πράξεις με δυο ή περισσότερες μεταβλητές.

Λογικές πράξεις με σταθερές

AND	OR	NOT
$0 \bullet 0 = 0$	$0 + 0 = 0$	
$0 \bullet 1 = 0$	$0 + 1 = 1$	$\overline{0} = 1$
$1 \bullet 0 = 0$	$1 + 0 = 1$	$\overline{1} = 0$
$1 \bullet 1 = 1$	$1 + 1 = 1$	

Λογικές πράξεις με μια μεταβλητή

AND	OR	NOT
$A \bullet 0 = 0$	$A + 0 = A$	
$A \bullet 1 = A$	$A + 1 = 1$	$\overline{\overline{A}} = A$
$A \bullet \overline{A} = 0$	$A + \overline{A} = 1$	

Παράδειγμα 1.6:

Να αποδειχθούν οι σχέσεις:

$$A + \bar{A} = 1 \quad \text{και} \quad A \bullet 1 = A$$

χρησιμοποιώντας τον πίνακα αληθείας

Απάντηση:

A	$\bar{A}$	$A + \bar{A}$	A	$A \bullet 1$
0	1	1	0	0
1	0	1	1	1

Λογικές πράξεις-ιδιότητες με δυο ή περισσότερες μεταβλητές

Αντιμεταθετική ιδιότητα

$$A + B = B + A$$

$$A \bullet B = B \bullet A$$

Απορροφητική ιδιότητα

$$A + (A \bullet B) = A$$

$$A \bullet (A + B) = A$$

Προσεταιριστική ιδιότητα

$$A + (B + C) = (A + B) + C$$

$$A \bullet (B \bullet C) = (A \bullet B) \bullet C$$

Επιμεριστική ιδιότητα

$$A \bullet (B + C) = (A \bullet B) + (A \bullet C)$$

$$A + (B \bullet C) = (A + B) \bullet (A + C)$$

Κανόνες De Morgan

$$\overline{A + B} = \bar{A} \bullet \bar{B}$$

$$\overline{A \bullet B} = \bar{A} + \bar{B}$$

Σημείωση:

$$\overline{A + B} \text{ διαβάζεται } A \text{ NOR } B$$

$$\overline{A \bullet B} \text{ διαβάζεται } A \text{ NAND } B$$

Κανόνες ελαχιστοποίησης

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

$$(A + B) \cdot (A + \overline{B}) = A$$

Παράδειγμα 1.7:

Να αποδειχθεί ότι: $(A + B) \cdot (A + \overline{B}) = A$

Απάντηση:

$$\begin{aligned} (A + B)(A + \overline{B}) &= AA + A\overline{B} + AB + B\overline{B} \\ &= A + A\overline{B} + AB + 0 \\ &= A + A(\overline{B} + B) \\ &= A + A \\ &= A \end{aligned}$$

Παράδειγμα 1.8:

Να αποδειχθεί ότι:

$$AB + A\overline{B} = A$$

$$A + (AB) = A(A + B) = A$$

Απάντηση:

$$1) AB + A\overline{B} = A(B + \overline{B}) = A \cdot 1 = A$$

$$A + (AB) = AA + AB = A(A + B)$$

$$2) A + (AB) = A(1 + B) = A \cdot 1 = A$$

Επίσης η απόδειξη μπορεί να γίνει και με τη χρήση του πίνακα αληθείας:

A	A•B	A+(A•B)
0	0	0
1	B	1

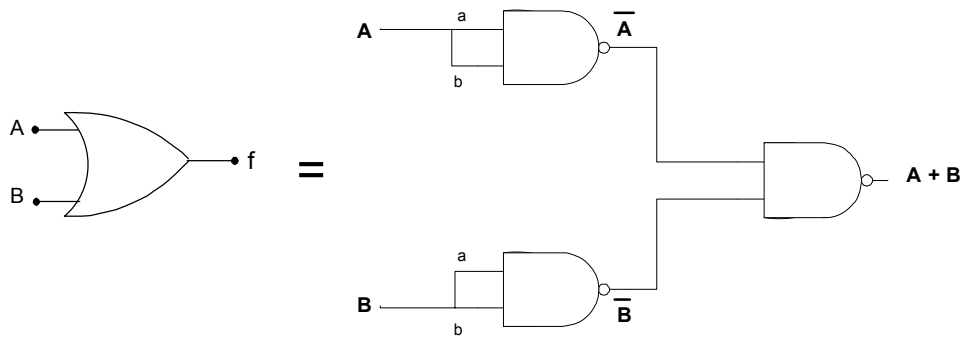
Τα θεωρήματα De Morgan είναι πιο σημαντικά στην λογική σχεδίαση όπου συσχετίζονται **AND** και **NOR** πύλες, ή **OR** και **NAND** πύλες. Για παράδειγμα χρησιμοποιούμε τα θεωρήματα De Morgan για να σχεδιάσουμε ένα συνδυασμό πυλών **NAND** που είναι ισοδύναμος με μια πύλη **OR** δύο εισόδων. Για μία πύλη **OR** ισχύει:

$$f = A + B$$

χρησιμοποιώντας τα θεωρήματα De Morgan:

$$f = A + B = \overline{\overline{A} \cdot \overline{B}}$$

Η παραπάνω σχέση υποδηλώνει μια πύλη **NAND** με **NOT** εισόδους. Επίσης, επειδή $\overline{\overline{A}} = A$, μια πύλη **NAND** με ίδιες εισόδους παρουσιάζει την λειτουργία της πύλης **NOT**. Συνεπώς προκύπτει το παρακάτω κύκλωμα:



Διαδικασία σχεδίασης ψηφιακής λογικής συνάρτησης

Με τον όρο σχεδιασμός ψηφιακής λογικής συνάρτησης, εννοείται ένας συνδυασμός λογικών πυλών για την πραγματοποίηση της επιθυμητής συνάρτησης, η συμπεριφοράς. Η διαδικασία σχεδίασης περιλαμβάνει τα παρακάτω βήματα:

1. Σαφής διατύπωση της επιθυμητής συνάρτησης-συμπεριφοράς
2. Πίνακας αληθείας
3. Έκφραση της συνάρτησης υπό μορφή μεταβλητών (άλγεβρα Boole)
4. Κατάλληλη επεξεργασία της συνάρτησης για την εξαγωγή μιας απλούστερης μορφής

5. Υλοποίηση του ψηφιακού κυκλώματος με πύλες **AND**, **OR** και **NOT**. Σε πολλές περιπτώσεις η υλοποίηση του κυκλώματος μπορεί να γίνει μόνο με πύλες **NAND**, η μόνο με πύλες **NOR**.

Κανονικές μορφές λογικών συναρτήσεων

Υπάρχουν δύο κανονικές μορφές λογικών συναρτήσεων, η κανονική μορφή αθροίσματος και η κανονική μορφή γινομένου

Κανονική μορφή αθροίσματος

Δημιουργείται από τον πίνακα αληθείας και είναι το λογικό άθροισμα (δηλαδή συνδυάζονται υπό μορφή **OR**) όρων που είναι εκφράσεις **AND** των μεταβλητών εισόδου στην κανονική, ή συμπληρωματική τους μορφή ανάλογα με την τιμή που έχουν (1 ή 0). Οι όροι που συμπεριλαμβάνονται στο λογικό άθροισμα είναι οι όροι για τους οποίους η τελική συνάρτηση έχει τιμή 1

Παράδειγμα 1.9:

Θεωρούμε τον πίνακα αληθείας

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

$\overline{ABC}$

$A\overline{B}\overline{C}$

$A\overline{B}C$

F=1

A=0, B=1 & C=1

A=1, B=0 & C=0

A=1, B=0 & C=1

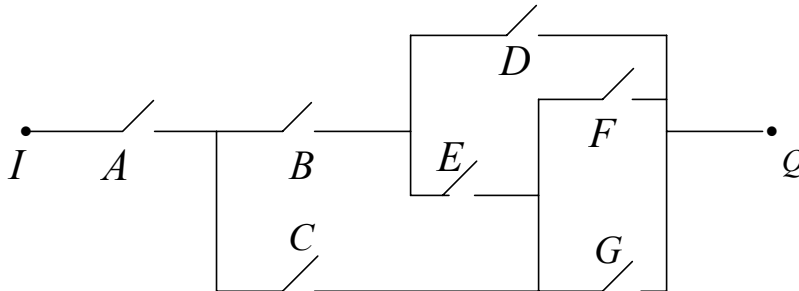
Δηλαδή οι δεκαδικοί αριθμοί 3, 4 & 5

Για κάθε έξοδο $F=1$ δημιουργούνται οι όροι **AND** όπου οι μεταβλητές είναι στη κανονική τους μορφή εάν είναι 1 και στην αντιστροφή τους μορφή εάν είναι 0. Κατόπιν οι παραπάνω όροι αθροίζονται λογικά, οπότε η κανονική μορφή αθροίσματος που προκύπτει είναι:

$$F = \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C$$

Παράδειγμα 1.10:

Γράψτε υπό κανονική μορφή αθροίσματος τη λογική έκφραση, σύμφωνα με την άλγεβρα Boole, για το παρακάτω κύκλωμα:



Απάντηση

Ακολουθούμε κάθε διαδρομή από το I στο Q και προσδιορίζουμε τους διακόπτες (μεταβλητές) που χρειάζεται να κλείσουν (ση) προκειμένου να παραχθεί Q = 1.

Επομένως το άθροισμα των όρων AND είναι:

$$Q = ACF + ACG + ACED + ABD + ABEF + ABEG$$

Σύντομη γραφή για την κανονική μορφή αθροίσματος

Στη σύντομη γραφή για την κανονική μορφή αθροίσματος, αντικαθίστανται κατ' αρχήν οι μεταβλητές των όρων από τις δυαδικές τους τιμές και κατόπιν ο κάθε όρος εκφράζεται με την αντίστοιχη δεκαδική του μορφή. Για την όλη γραφή χρησιμοποιείται το σύμβολο **Σ**. Για την περίπτωση μιας λογικής συνάρτησης τριών μεταβλητών, όπως του παραπάνω παραδείγματος, έχουμε:

$$F(ABC) = \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C$$

οπότε αντικαθιστώντας τις μεταβλητές με τις δυαδικές τους τιμές προκύπτει:

$$F(ABC) = 011 + 100 + 101$$

και λαμβάνοντας τη δεκαδική μορφή του κάθε όρου:

$$F(ABC) = 3 + 4 + 5$$

Συνεπώς χρησιμοποιώντας το σύμβολο Σ έχουμε τη γραφή:

$$F(ABC) = \Sigma(3,4,5)$$

Παράδειγμα 1.11:

Να εκφράσετε τη συνάρτηση $F(ABCD) = \Sigma(3,4,9,10)$ με όρους μεταβλητών (υπό μορφή άλγεβρας Boole).

Απάντηση

Ο όρος 3 είναι το 0011 και αντιστοιχεί στο

$$\overline{A} \overline{B} C D$$

Ο όρος 4 είναι το 0100 και αντιστοιχεί στο

$$\overline{A} B \overline{C} \overline{D}$$

Ο όρος 9 είναι το 1001 και αντιστοιχεί στο

$$A \overline{B} \overline{C} D$$

Ο όρος 10 είναι το 1010 και αντιστοιχεί στο

$$A \overline{B} C \overline{D}$$

Οπότε σε κανονική μορφή αθροίσματος είναι:

$$F = \overline{A} \overline{B} C D + \overline{A} B \overline{C} \overline{D} + A \overline{B} \overline{C} D + A \overline{B} C \overline{D}$$

Παράδειγμα 1.12:

Δίνεται η λογική συνάρτηση $Q = (A + B + C)(A + \overline{B} + C)(A + \overline{B} + \overline{C})$

Να γίνει ο πίνακας αληθείας, να γραφεί η κανονική μορφή αθροίσματος, να απλοποιηθεί η σχέση χρησιμοποιώντας την άλγεβρα Boole και να σχεδιαστεί το ψηφιακό κύκλωμα που την υλοποιεί.

Απάντηση:

A	B	C	$(A+B+C)$	$(A+\bar{B}+C)$	$(A+\bar{B}+\bar{C})$	Q
0	0	0	0	1	1	0
0	0	1	1	1	1	$\bar{A}\bar{B}C$ 1
0	1	0	1	0	1	0
0	1	1	1	1	0	0
1	0	0	1	1	1	$A\bar{B}\bar{C}$ 1
1	0	1	1	1	1	$A\bar{B}C$ 1
1	1	0	1	1	1	$AB\bar{C}$ 1
1	1	1	1	1	1	ABC 1

Χρησιμοποιώντας τους όρους για τους οποίους $Q=1$ προκύπτει:

$$Q = \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

και απλοποιώντας:

$$Q = \bar{A}\bar{B}C + A\bar{B}(\bar{C} + C) + AB(\bar{C} + C)$$

$$= \bar{A}\bar{B}C + A\bar{B} + AB$$

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

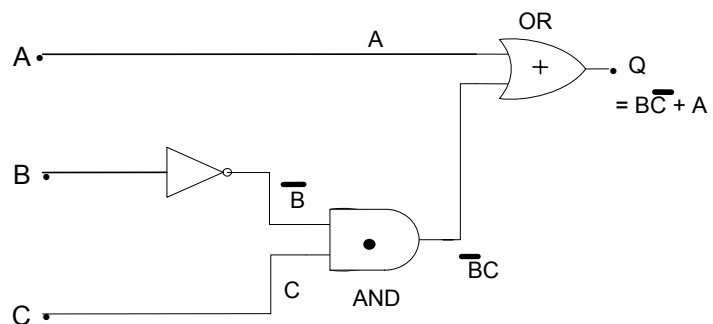
$$Q = \bar{A}\bar{B}C + A$$

για περαιτέρω απλοποίηση:

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

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

Ο σχεδιασμός του ψηφιακού κυκλώματος είναι ο εξής:



Ημιαθροιστής

Ο ημιαθροιστής είναι ένα ψηφιακό κύκλωμα που πραγματοποιεί την αλγεβρική άθροιση δύο δυαδικών ψηφίων:

Αλγεβρική άθροιση				
A	1	1	0	0
B	1	0	1	0
A+B	10	1	1	0

ο αντίστοιχος πίνακας αληθείας είναι:

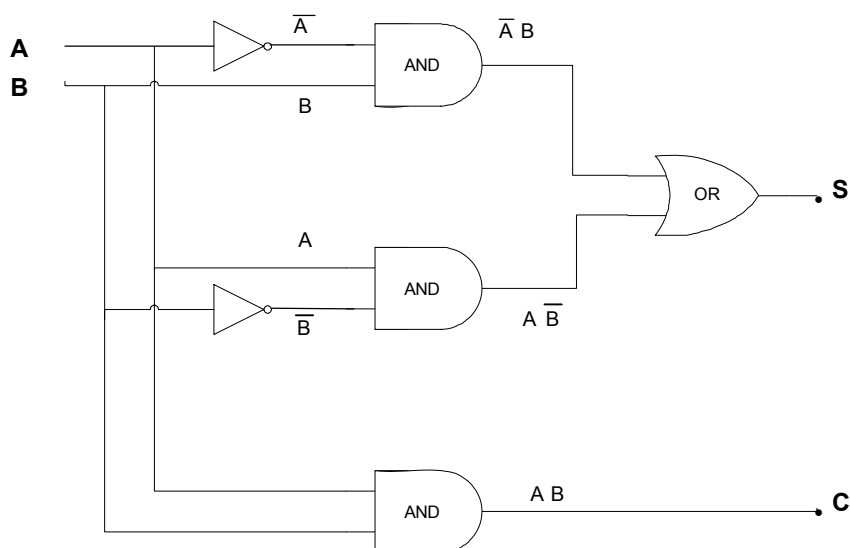
Πίνακας Αληθείας

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Συνεπώς

$$S = \bar{A}B + A\bar{B} \text{ και } C = AB$$

Το ψηφιακό του κύκλωμα είναι:



Κανονική μορφή γινομένου

Αυτή είναι μια εναλλακτική μορφή υλοποίησης της πρώτης μορφής. Οι όροι είναι αθροίσματα (δηλαδή τύπου **OR**) και πολλαπλασιάζονται μεταξύ τους προκειμένου να σχηματίσουν την έξοδο. Η κατανόηση της διατύπωσης του κανόνα που θα χρησιμοποιούμε στο σχηματισμό της κανονικής μορφής γινομένου γίνεται με το παρακάτω παράδειγμα:

Έστω ότι δίνεται ο πίνακας αληθείας:

A	B	C	F
0	0	0	0 $\bar{A} \bar{B} \bar{C}$
0	0	1	0 $\bar{A} \bar{B} C$
0	1	0	0 $\bar{A} B \bar{C}$
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0 $A B \bar{C}$
1	1	1	0 $A B C$

οπότε προκύπτει:

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

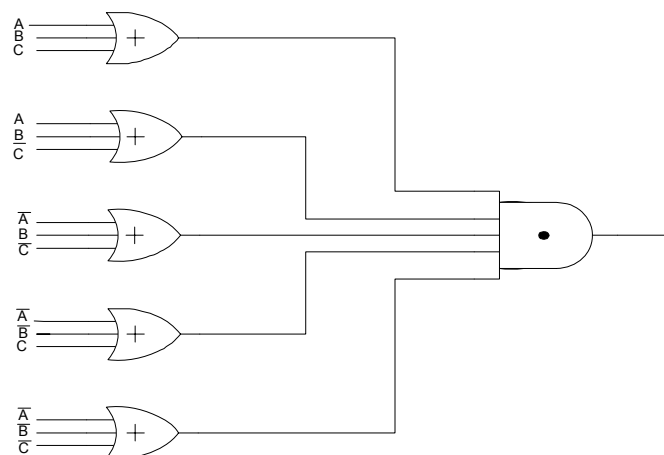
$$F = \overline{\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC}$$

$$F = \overline{\bar{A}\bar{B}\bar{C}} \cdot \overline{\bar{A}\bar{B}C} \cdot \overline{\bar{A}B\bar{C}} \cdot \overline{A\bar{B}\bar{C}} \cdot \overline{ABC}$$

και τελικά

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

Η υλοποίηση της τελικής συνάρτησης γίνεται με το κύκλωμα:



Τελικά η κανονική μορφή γινομένου μπορεί να αποκτηθεί κατευθείαν από τον πίνακα αληθείας χωρίς τη χρήση κάποιων πράξεων ως εξής:

1. Εντοπίζουμε τους όρους που δίνουν $F=0$.
2. Δημιουργούμε τα αθροίσματα των μεταβλητών, όπου εάν η μεταβλητή έχει τιμή 0 γράφεται στην κανονική της μορφή, ενώ εάν έχει τιμή 1, γράφεται στην αντίστροφη μορφή της.
3. Παίρνουμε το γινόμενο των παραπάνω αθροισμάτων.

Παράδειγμα:

A	B	C	F	
0	0	0	0	$A+B+C$
0	0	1	0	$A+B+\overline{C}$
0	1	0	0	$A+\overline{B}+C$
0	1	1	1	
1	0	0	1	
1	0	1	1	
1	1	0	0	$\overline{A}+\overline{B}+C$
1	1	1	0	$\overline{A}+\overline{B}+\overline{C}$

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

Σύντομη γραφή για την κανονική μορφή γινομένου

Στην περίπτωση αυτή η κανονική μορφή των μεταβλητών παριστάνει το **0**, ενώ η αντίστροφη το **1**. Συνεπώς αντικαθιστώντας τις μεταβλητές με τη δυαδική τους μορφή, χρησιμοποιώντας το παραπάνω παράδειγμα, προκύπτει:

$$F = (000)(001)(101)(110)(111)$$

0 1 5 6 7

και σε σύντομη γραφή:

$$F = \prod(0,1,5,6,7)$$

Παράδειγμα 1.13:

Να γράψετε τις δύο κανονικές μορφές της συνάρτησης XOR

Απάντηση:

Ο πίνακας αληθείας για την πύλη XOR είναι:

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

Για την κανονική μορφή αθροίσματος παίρνουμε τους όρους για $F=1$:

$F=1$ $A=0 \text{ \& } B=1$ δίνει $\overline{A}B$
 $A=1 \text{ \& } B=0$ δίνει $A\overline{B}$

Οπότε προκύπτει:

$$F = \overline{A}B + A\overline{B}$$

Για τη δεύτερη κανονική μορφή γινομένου παίρνουμε τους όρους για $F=0$:

$F=0$ $A=0 \text{ \& } B=0$ δίνει $(\overline{A} + \overline{B})$
 $A=1 \text{ \& } B=1$ δίνει $(A + B)$

οπότε προκύπτει:

$$F = (A + B) \bullet (\overline{A} + \overline{B})$$

Σημειώνεται ότι ισχύει:

$$F = (A + B) \bullet (\overline{A} + \overline{B}) = \overline{A}B + A\overline{B}$$

Σύνθεση ψηφιακού κυκλώματος

Κατ' αρχήν απλοποιείται η λογική συνάρτηση, η οποία πρόκειται να υλοποιηθεί. Κατόπιν σχεδιάζεται το ψηφιακό κύκλωμα που αντιστοιχεί στην λογική συνάρτηση ξεκινώντας από την έξοδο του κυκλώματος και πηγαίνοντας προς την είσοδό του.

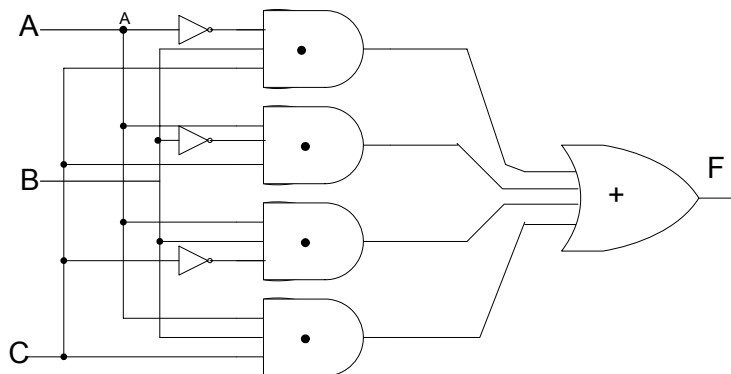
Παράδειγμα:

Να υλοποιηθεί η συνάρτηση που έχει τον παρακάτω πίνακα αληθείας

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 $\bar{A}BC$
1	0	0	0
1	0	1	1 $A\bar{B}C$
1	1	0	1 $AB\bar{C}$
1	1	1	1 ABC

$$F = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

Το ψηφιακό κύκλωμα χωρίς απλοποίηση είναι:



Παρατηρείται ότι χωρίς απλοποίηση χρειάζονται 8 πύλες.

Γίνεται απλοποίηση της λογικής συνάρτησης:

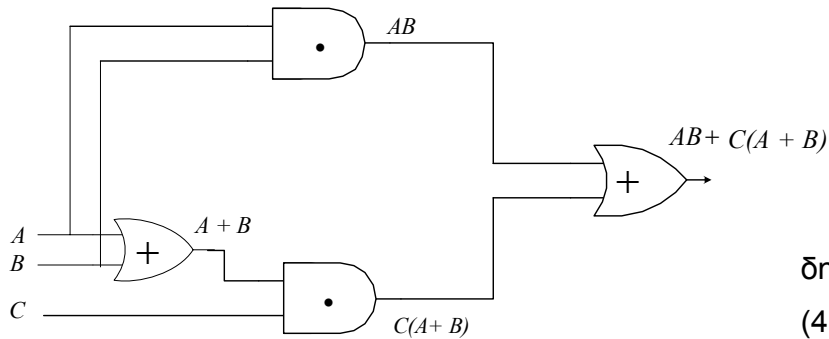
$$F = A\bar{B}\bar{C} + A\bar{B}C + \bar{A}B\bar{C} + A\bar{B}C + \bar{A}B\bar{C} + ABC + \bar{A}B\bar{C} + ABC$$

δηλαδή προστέθηκε ο όρος ABC αφού είναι γνωστό ότι $ABC+ABC=ABC$!

Συνεπώς:

$$F = AB(\bar{C} + C) + C(\bar{A}\bar{B} + A\bar{B} + AB) = AB + C(A + B) \quad \text{αφού} \quad \bar{A}\bar{B} + A\bar{B} + AB = AB$$

Υλοποίηση:



δηλαδή χρειάζονται τέσσερις
(4) πύλες

Σύνθεση ψηφιακών κυκλωμάτων με πύλες NAND

Επειδή τα τρανζίστορ είναι ουσιαστικά αντιστροφείς, οι πύλες NAND αποτελούν δομικά στοιχεία των ολοκληρωμένων κυκλωμάτων τεχνολογίας DTL & TTL. Τα βήματα που χρησιμοποιούνται για τη σχεδίαση ενός κυκλώματος μόνο με πύλες **NAND** είναι τα εξής:

1. Χρησιμοποιείται ο πίνακας αληθείας για να εκφρασθεί η λογική συνάρτηση υπό μορφή αθροίσματος γινομένων:

$$F = P_1 + P_2 + \dots + P_n$$

(P είναι το γινόμενο των μεταβλητών εισόδου σε μια γραμμή στην οποία η έξοδος είναι 1)

2. Στο γινόμενο που αντιστοιχεί σε μια δεδομένη γραμμή, οι μεταβλητές των οποίων οι τιμές είναι 0, λαμβάνονται με την αντίστροφη μορφή τους (δηλαδή εάν η μεταβλητή A σε κάποιον όρο έχει τιμή 0, στο γινόμενο θα εμφανιστεί ως $\overline{A}$)

3. Χρησιμοποιώντας το θεώρημα του De Morgan γράφεται η σχέση υπό τη μορφή:

$$F = \overline{\overline{P_1} \cdot \overline{P_2} \dots \overline{P_n}}$$

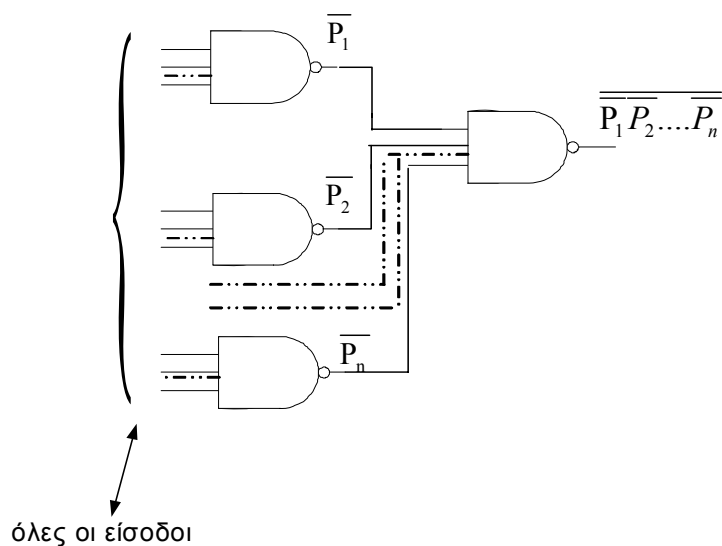
4. Συνθέτουμε το κύκλωμα με πύλες NAND.

Παράδειγμα 1.14:

Ο πίνακας αληθείας μιας συνάρτησης F είναι:

A	B	C	D	f
0	0	0	0	P_1
0	0	0	1	
0	0	1	0	
0	1	0	1	P_k
1	1	1	0	
1	1	1	1	P_n
1	1	1	1	

οπότε ο σχεδιασμός του κυκλώματος με πύλες **NAND** είναι:

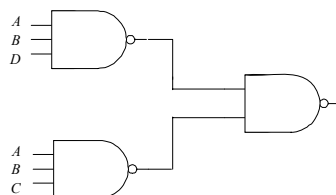


Παράδειγμα 1.15:

Η συνάρτηση $F = ABC + ABD$ να υλοποιηθεί με πύλες **NAND:**

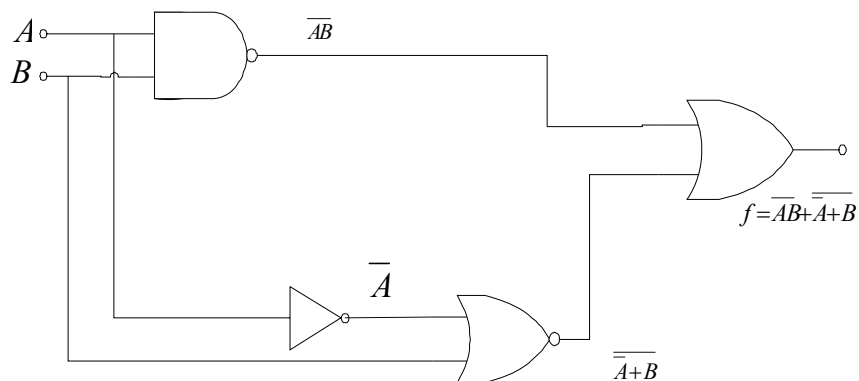
Λύση:

$$F = \overline{\overline{ABC} \cdot \overline{ABD}}$$



Παράδειγμα 1.16:

Να αναλυθεί το κύκλωμα του σχήματος, να γίνει ο πίνακας αληθείας και να αποδειχθεί, χρησιμοποιώντας είτε την άλγεβρα *Boole* είτε πίνακα αληθείας, ότι μπορεί να αντικατασταθεί από μια πύλη **NAND.**



Λύση:

$$\begin{aligned}
 f &= \overline{AB} + \overline{\overline{A} + B} \\
 &= (\overline{A} + \overline{B}) + A\overline{B} \\
 &= \overline{A} + \overline{B}(1 + A) \\
 &= \overline{A} + \overline{B}
 \end{aligned}$$

$\overline{AB}$



πύλη NAND

A	B	$\overline{AB}$	$\overline{\overline{A} + B}$	f
0	0	1	0	1
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0

πύλη NAND

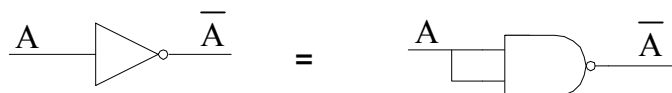
Αντικατάσταση πυλών με πύλες NAND

Οι πύλες **AND**, **OR** και **NOT** μπορούν να εξαχθούν από πύλες **NAND**.

NOT

ισχύει $F = \overline{A}$ ή $F = \overline{A \cdot A}$

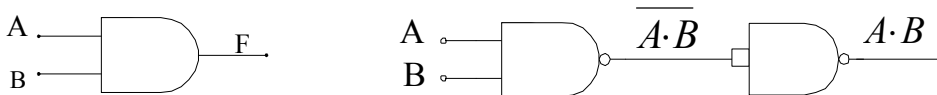
συνεπώς



AND

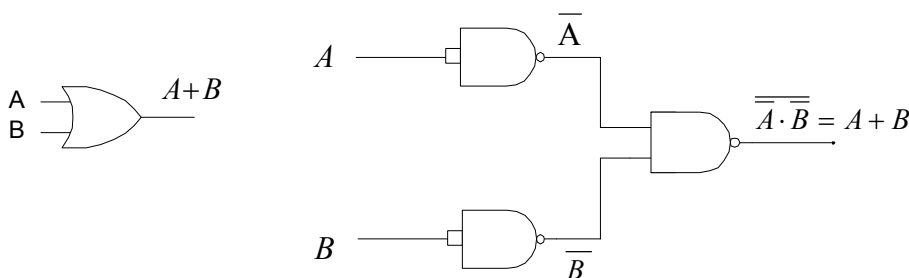
$$A \cdot A = \overline{\overline{A \cdot B}}$$

Δηλαδή η **AND** πύλη μπορεί να αντικατασταθεί από μια **NAND**, η έξοδος της οποίας αντιστρέφεται από μια δεύτερη **NAND**

**OR**

Θεώρημα De Morgan

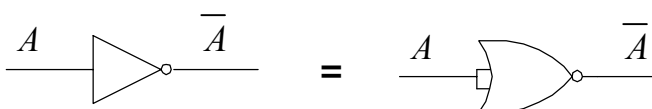
$$\overline{A \cdot B} = \overline{A} + \overline{B} \quad \text{οπότε} \quad A + B = \overline{\overline{A \cdot B}}$$

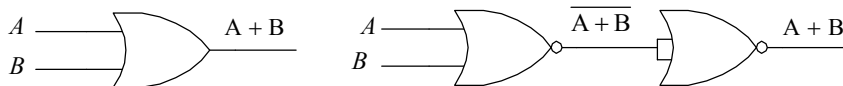
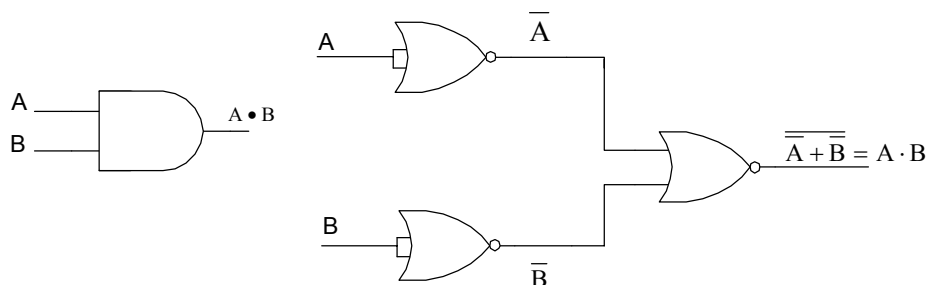
**Σύνθεση ψηφιακών κυκλωμάτων με πύλες NOR**

Η σύνθεση των ψηφιακών κυκλωμάτων μόνο με πύλες **NOR** γίνεται με παρόμοιο τρόπο όπως με τις πύλες **NAND**, μόνο που σ' αυτή την περίπτωση χρησιμοποιείται η κανονική μορφή γινομένου.

Αντικατάσταση πυλών με πύλες NOR

Η λογική **NOR** είναι η δυαδική της λογικής **NAND**. Οι πύλες **AND**, **OR** και **NOT** μπορούν να δημιουργηθούν με πύλες **NOR** ως εξής:

NOT

OR**AND**

Ελαχιστοποίηση λογικών συναρτήσεων με τη χρήση των πινάκων Karnaugh

Στο σχεδιασμό λογικών κυκλωμάτων επιζητείται το βέλτιστο, προκειμένου να υλοποιηθεί μια συγκεκριμένη λογική συνάρτηση. Κριτήρια του βέλτιστου μπορεί να είναι

- η ταχύτητα (λιγότερα λογικά επίπεδα)
- το κόστος (λιγότερες λογικές πύλες)

Ήδη έχει επιδειχθεί ο τρόπος ελαχιστοποίησης με τη χρήση της άλγεβρας Boole.

Εναλλακτικά μπορούν να χρησιμοποιηθούν οι πίνακες Karnaugh, εάν η συνάρτηση είναι γραμμένη με μια από τις δυο κανονικές μορφές.

Πίνακες Karnaugh

Αν θεωρηθεί μια συνάρτηση τριών μεταβλητών ABC, τότε η συνάρτηση μπορεί να απεικονισθεί στον πίνακα Karnaugh με τον εξής τρόπο:

F		AB	$\overline{A}\overline{B}$	$\overline{A}B$	$A\overline{B}$
		00	01	11	10
C	$\overline{C}$ 0	$\overline{A}\overline{B}\overline{C}$ 0	$\overline{A}\overline{B}C$ 2	$A\overline{B}\overline{C}$ 6	$A\overline{B}C$ 4
	C 1	$\overline{A}B\overline{C}$ 1	$\overline{A}BC$ 3	ABC 7	ABC 5

Παρατηρήσεις:

- Κάθε τετράγωνο αντιστοιχεί σ' έναν από τους οκτώ (8) δυνατούς συνδυασμούς των τριών μεταβλητών.
- Τα τετράγωνα του πίνακα είναι κατά αυτόν τον τρόπο διατεταγμένα ώστε σε γειτονικά τετράγωνα να αλλάζει μόνο μια μεταβλητή (κώδικας Gray).

Για κάθε ζεύγος τετραγώνων γίνεται η παρακάτω απλοποίηση:

$\overline{A}\overline{B}\overline{C}$
$\overline{A}\overline{B}C$

$$\begin{aligned}
 F &= \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C \\
 &= \overline{A}\overline{B}(\overline{C} + C) = \\
 &= \overline{A}\overline{B}
 \end{aligned}$$

ομοίως

ABC	$A\overline{B}C$
-------	------------------

$$\begin{aligned}
 F &= ABC + A\overline{B}C \\
 &= AC(\overline{B} + B) \\
 &= AC
 \end{aligned}$$

Για τέσσερα γειτονικά τετράγωνα ισχύει:

$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$
$A\overline{B}\overline{C}$	$A\overline{B}C$

$$F = A$$

Συμπέρασμα:

- Δύο (2) γειτονικά τετράγωνα (δηλαδή δυο όροι) δημιουργούν έναν όρο με μια μεταβλητή λιγότερη.

- Τέσσερα (4) γειτονικά τετράγωνα δημιουργούν έναν όρο με δυο μεταβλητές λιγότερες.
- Ομάδες των τριών τετραγώνων πρέπει να χωρίζονται σε ομάδες των δυο.

Παράδειγμα 1.17:

Να γίνει πίνακας-K για τη συνάρτηση $F = \Sigma(1,2,5,6)$

		00	01	11	10
F					
0		0	1	1	0
1		1	0	0	1

πλήρης συνάρτηση

$$F = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}C + A\overline{B}\overline{C}$$

απλοποιημένη συνάρτηση

$$F = \overline{B}C + B\overline{C}$$

Σημείωση: Ο αριθμός των μεταβλητών είναι ίσος με από τον εκθέτη του 2 για τον οποίο η δύναμη του 2 μας δίνει αριθμό μεγαλύτερο ή ίσο με το μέγιστο αριθμό που έχουμε στη συνάρτηση. Συνεπώς στο παράδειγμα $2^3 > 6$, άρα 3 μεταβλητές.

Παράδειγμα 1.18:

Να γίνει ο πίνακας-K για τη συνάρτηση $F = \Sigma(0,2,4,9,11)$, καθώς επίσης απλοποίηση αυτής

Λύση:

$11 < 2^4$ άρα 4 μεταβλητές

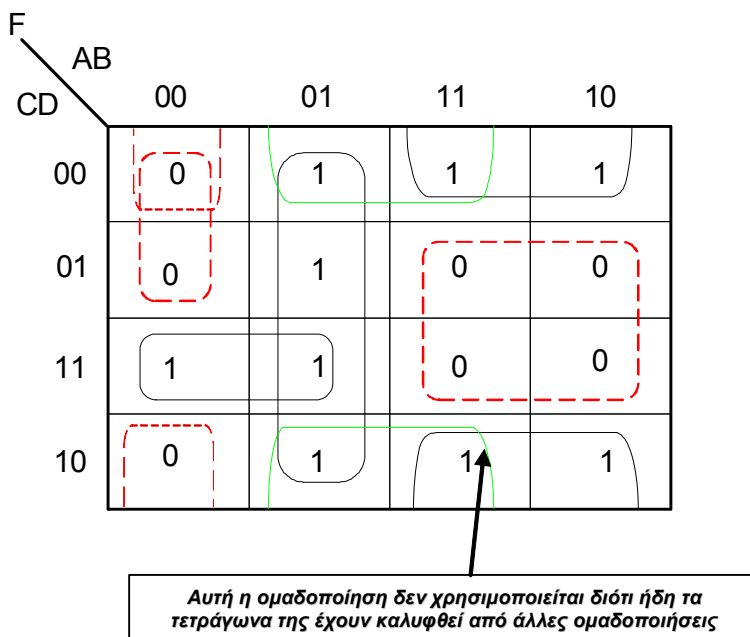
		AB	00	01	11	10
F						
CD						
00			1	1	0	0
01			0	0	0	1
11			0	0	0	1
10			1	0	0	0

$$F = \overline{A}\overline{B}C + \overline{A}BD + A\overline{B}D$$

Παράδειγμα 1.19:

Να βρεθεί η ελαχιστοποιημένη μορφή αθροίσματος και η ελαχιστοποιημένη μορφή γινομένου της συνάρτησης $F = \Sigma (3,4,5,6,7,8,10,12,14)$

Λύση:



Για την κανονική μορφή αθροίσματος προκύπτει:

$$F = \overline{A}\overline{B} + A\overline{D} + \overline{A}CD$$

Για τη μορφή γινομένου ομαδοποιούμε τα μηδενικά, όπως φαίνεται στον πίνακα-K,

Οπότε προκύπτει:

$$F = (\overline{A} + \overline{D})(A + B + C)(A + B + D)$$

Παράδειγμα 1.20:

Να ελαχιστοποιηθεί η συνάρτηση

$$F = \overline{A}\overline{B}(C + D) + \overline{A}C\overline{D} + ABC\overline{D} + ABCD$$

Απάντηση:

F	AB			
	00	01	11	10
CD	00	01	11	10
00	1	1	1	1
01				1
11				
10	1	1		

$$F = \overline{A}\overline{D} + \overline{C}\overline{D} + \overline{A}BC$$

Ύπαρξη αδιάφορων περιπτώσεων

Σε λογικά κυκλώματα υπάρχουν πολλές φορές ορισμένοι συνδυασμοί των μεταβλητών εισόδου που μας είναι αδιάφοροι. Για παράδειγμα έστω ότι έχουμε ένα ηλεκτρονικό ψηφιακό κύκλωμα που θέτει εκτός ένα σήμα (alarm), εάν στην είσοδο του έχει τους αριθμούς 0,4,6,8,9. Εάν έχει σχεδιαστεί κατά τέτοιον τρόπο ώστε να δέχεται αριθμούς μόνο από το 0 έως το 9 να α) προσδιοριστεί το πρόβλημα υπό μορφή πίνακα β) βρεθεί η ελαχιστοποιημένη συνάρτηση με τη χρήση του πίνακα-K.

Πίνακας αληθείας

	A	B	C	D	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1

10 -----
 11 -----

αδιάφορες
περιπτώσεις

Πίνακας-K

F	AB			
	00	01	11	10
CD	00	01	11	10
00	1	1	D	1
01	0	0	D	1
11	0	0	D	D
10	0	1	D	D

αδιάφορες
περιπτώσεις

Εάν κατά την απλοποίηση δεν ληφθούν υπόψη οι αδιάφορες περιπτώσεις προκύπτει η σχέση:

$$F = \overline{A}CD + \overline{A}BD + \overline{A}BC$$

Λαμβάνοντας υπόψη και τις αδιάφορες περιπτώσεις η σχέση στην οποία καταλήγουμε είναι απλούστερη:

$$F = A + \overline{C}\overline{D} + \overline{B}\overline{C}$$

Πλήρης Αθροιστής

Κατ' αρχήν εξετάζεται ο ημιαθροιστής δημιουργώντας το ψηφιακό του κύκλωμα χρησιμοποιώντας την κανονική μορφή γινομένου:

A	B	S	C
0	0	0	0
0	1	1 ($\overline{A}B$)	0
1	0	1 ($A\overline{B}$)	0
1	1	0	1 (AB)

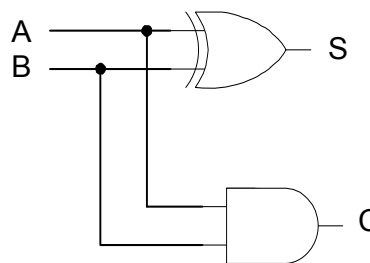
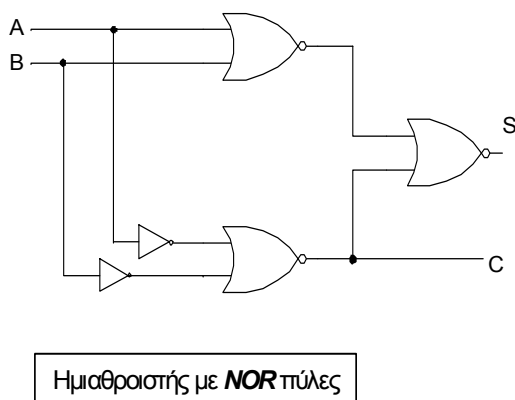
$$S = \overline{A}B + A\overline{B}, \quad C = A \cdot B$$

Χρησιμοποιώντας για το **S** τη δεύτερη μορφή γινομένου προκύπτει:

$$S = (A + B)(\overline{A} + \overline{B}) = \overline{\overline{A+B} + \overline{\overline{A+B}}} \quad (\text{De Morgan})$$

$$\text{επίσης: } C = A \cdot B = \overline{\overline{A+B}}$$

Συνεπώς η S μπορεί να δημιουργηθεί χρησιμοποιώντας μόνο πύλες NOR, η δε C προκύπτει από μια εξ αυτών των πυλών:



Ημιαθροιστής χρησιμοποιώντας **XOR** και **AND** πύλες

Ο πλήρης αθροιστής έχει τον παρακάτω πίνακα αληθείας:

A	B	C _i	S	C ₀
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

άρα

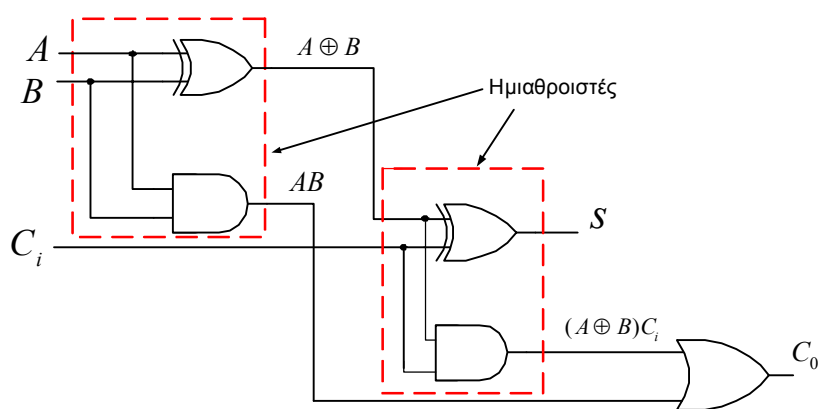
$$S = \overline{A}B\overline{C}_i + \overline{A}BC_i + A\overline{B}\overline{C}_i + ABC_i$$

$$\begin{aligned} S &= \overline{C}_i(\overline{A}B + A\overline{B}) + C_i(\overline{A}B + AB) \\ &= \overline{C}_i(A \oplus B) + C_i(\overline{A \oplus B}) = (A \oplus B) \oplus C_i \end{aligned}$$

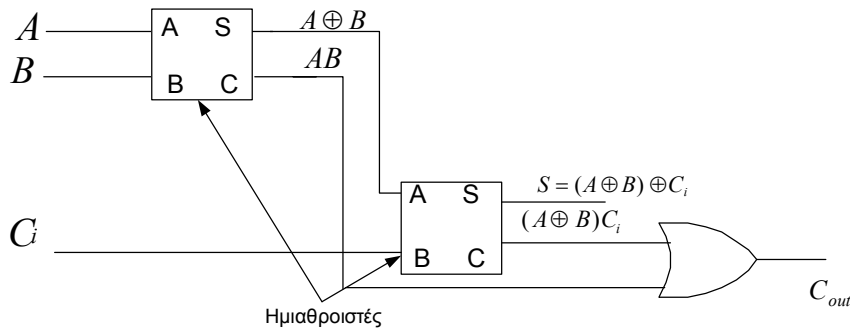
και

$$\begin{aligned} C_0 &= ABC_i + \overline{A}BC_i + A\overline{B}C_i + ABC_i \\ &= C_i(\overline{A}B + A\overline{B}) + AB = C_i(A \oplus B) + AB \end{aligned}$$

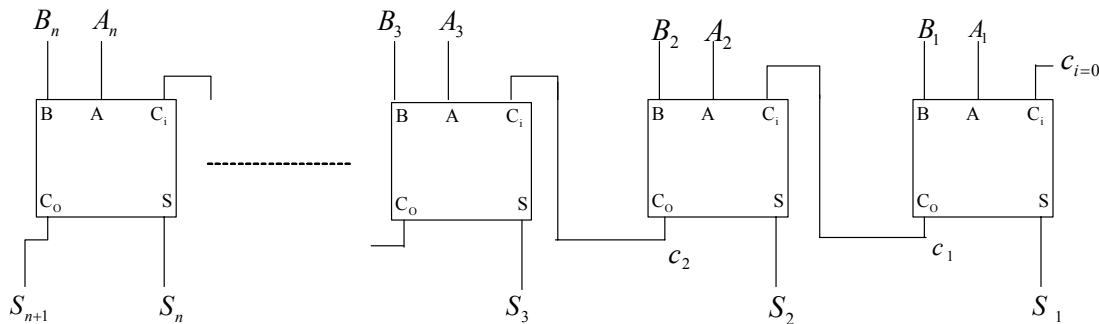
Το κύκλωμα που υλοποιεί τις παραπάνω σχέσεις είναι:

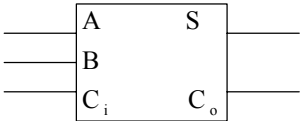


ή χρησιμοποιώντας το συμβολικό κύκλωμα του ημιαθροιστή:



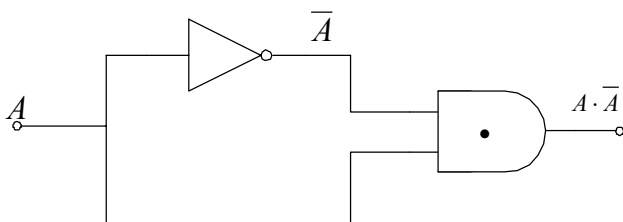
Η άθροιση αριθμών με περισσότερα του ενός δυαδικά ψηφία γίνεται με το κύκλωμα του παράλληλου αθροιστή ως εξής:



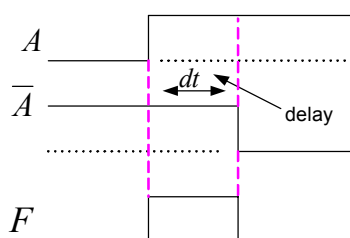
όπου  το σύμβολο του πλήρους αθροιστή.

Σπινθήρες

Οι πραγματικές ηλεκτρονικές πύλες απαιτούν κάποιο χρόνο για τη λειτουργία τους. Δηλαδή παρουσιάζουν καθυστέρηση (delay) της τάξης των λίγων μs . Οι καθυστερήσεις αυτές δημιουργούν καταστάσεις εξόδου, όπως είναι οι σπινθήρες (**hazards**), που είναι πολλές φορές ανεπιθύμητες. Για παράδειγμα στο κύκλωμα:



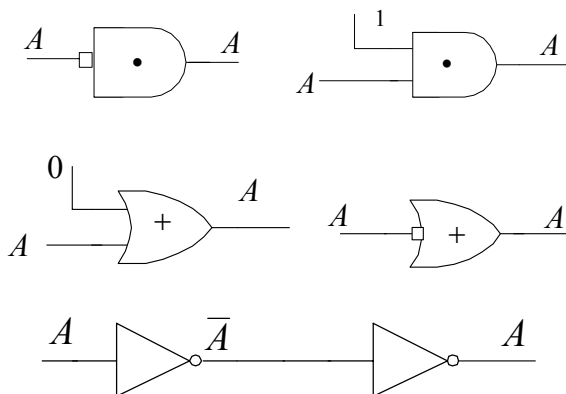
η έξοδος του, στην ιδανική περίπτωση, θα πρέπει να είναι ίση με μηδέν ανεξάρτητα από την τιμή της εισόδου. Στην πραγματικότητα η έξοδος είναι όπως φαίνεται στο σχήμα:



Δηλαδή παρατηρείται ότι η έξοδος παίρνει την τιμή 1 κατά το χρονικό διάστημα της καθυστέρησης (hazard).

Υπάρχουν τρεις τρόποι περιορισμού των σπινθηρισμών:

- Αναμονή μέχρι ωστόσο να εμφανιστεί η σωστή έξοδος. Η μέθοδος αυτή δεν συνίσταται κυρίως για ψηφιακά συνδυαστικά κυκλώματα που χρησιμοποιούνται ως οδηγοί ακολουθιακών κυκλωμάτων.
- Εξισορρόπηση της καθυστέρησης χρησιμοποιώντας διατάξεις πυλών όπως:



Χρήση πινάκων Karnaugh. Ο σπινθηρισμός συμβαίνει συνήθως κατά τη μετάβαση καταστάσεων που είναι γειτονικές στον πίνακα-K και ομαδοποιούνται μεταξύ τους. Η επίλυση γίνεται ως εξής: ομαδοποιούνται γειτονικά τετράγωνα (cells) ακόμα κι' αν αυτά εμπλέκονται σ' άλλες ομάδες εισάγοντας έναν άλλο εφεδρικό όρο στην συνάρτηση.

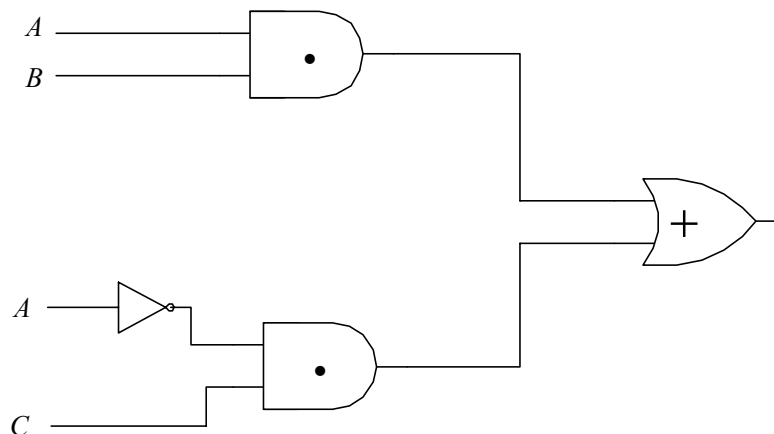
Παράδειγμα 1.21:

Βρείτε και απαλείψτε το σπινθηρισμό στη συνάρτηση:

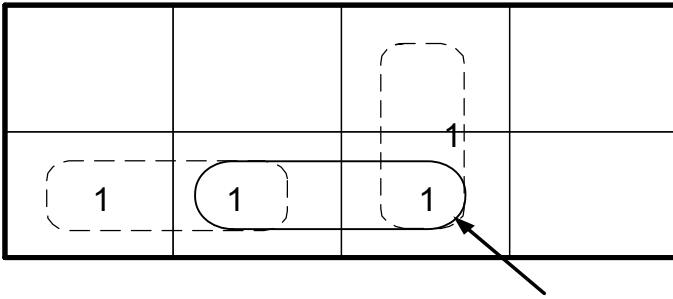
$$F = \sum (1, 3, 6, 7)$$

		AB	
C			1
	1	1	1

$$F = AB + \bar{A}C$$



Ο σπινθηρισμός συμβαίνει κατά τη μετάβαση από το τετράγωνο **7** στο τετράγωνο **3**, ή από τη κατάσταση **ABC=111** στην **ABC=011**. Από το κύκλωμα εξάγεται ότι αλγεβρικά η **F** πρέπει να έχει την τιμή 1, λόγω όμως της καθυστέρησης θα παρουσιάσει στιγμιαία τιμή 0. Ο σπινθηρισμός μπορεί να αποφευχθεί χρησιμοποιώντας εφεδρικό βρόχο μεταξύ του **7** και **3**.



οπότε η συνάρτηση γίνεται:

$$F = AB + \overline{A}C + BC$$

Ασκήσεις επανάληψης 1

Άσκηση 1.1

Να εκφράσετε τον δυαδικό αριθμό **1100111.1101** στον αντίστοιχο δεκαδικό (βάση το 10)

Λύση:

Ακέραιο μέρος: $1100111 = 1 \cdot 2^6 + 1 \cdot 2^5 + \dots + 1 = 103_{10}$

Δεκαδικό μέρος: $.1101 = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = 0,8125_{10}$

Άρα ο αριθμός είναι : 103.8125_{10}

Άσκηση 1.2

Να μετατραπεί ο δεκαδικός αριθμός **278,632** στον ισοδύναμο δυαδικό. (βάση το 2)

Λύση:

Ακέραιο μέρος	διαίρεση με 2	
	2) 278	
	2) 139 (πηλίκο)	0 (υπόλοιπο)
	2) 69	1
	2) 34	1
	2) 17	0
	2) 8	1
	2) 4	0
	2) 2	0
	2) 1	0
	0	1

100010110

δεκαδικό μέρος		
	0,632 * 2 = 1,264	1
	0,264 * 2 = 0,528	0
	0,528 * 2 = 1,056	1
	0,056 * 2 = 0,112	0
	0,112 * 2 = 0,226	0
	0,224 * 2 = 0,448	0
	0,448 * 2 = 0,896	0
	0,896 * 2 = 1,792	1
	0,792 * 2 = 1,584	1
		

0,101000011

δυαδικός αριθμός: 100010110.101000011

Άσκηση 1.3

Να μετατραπεί ο δεκαδικός αριθμός **123,456** σε ισοδύναμο οκταδικό

Λύση:

ακέραιο	8) 123		
μέρος	8) 15	3	
	8) 1	7	
	0	1	→ 173 ₈

δεκαδικό	0,456 * 8 = 3,648	3	
μέρος	0,648 * 8 = 5,184	5	
	0,184 * 8 = 1,472	1	
	0,472 * 8 = 3,776	3	
			
			

→ 0,3513₈

άρα ο αριθμός είναι 173,3513₈

Άσκηση 1.4

Να βρεθεί ο δυαδικός αριθμός που αντιστοιχεί στον δεκαδικό **27810** χρησιμοποιώντας κωδικοποίηση **BCD**

Λύση:

$$278_{10} = \underbrace{0010}_{2} \underbrace{0111}_{7} \underbrace{1000}_{8}$$

Άσκηση 1.5

Να μετατραπεί α) ο δυαδικός **011010011012** στον αντίστοιχο δυαδικό του κώδικα Gray και β) ο δυαδικός **1000110101010** του κώδικα Gray στον αντίστοιχο κανονικό δυαδικό αριθμό.

Λύση:

α)

0	1	1	0	1	0	0	1	1	0	1
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	1	0	1	1	1	0	1	0	1	1

Ψηφιακά Κυκλώματα 50 / 95 Τμήμα Πληροφορικής & Επικοινωνιών

Κανόνας μετατροπής: α) Το πρώτο ψηφίο από αριστερά είναι ίδιο, β) Όταν τα διαδοχικά ψηφία του κανονικού δυαδικού αλλάζουν, τότε το αντίστοιχο ψηφίο του δυαδικού σε κώδικα Gray γίνεται 1 διαφορετικά 0.

β) 1 0 0 0 1 1 0 1 0 1 0 1 0
 1 1 1 1 0 1 1 0 0 1 1 0 0

Κανόνας μετατροπής: α) Το πρώτο ψηφίο από αριστερά είναι ίδιο β) για τα υπόλοιπα, όταν υπάρχει 1 στον κώδικα Gray σημαίνει ότι το αντίστοιχο ψηφίο του κανονικού δυαδικού πρέπει να αλλάξει, ενώ όταν υπάρχει μηδέν παραμένει ίδιο με το προηγούμενό του

Άσκηση 1.6

Να αποδειχθούν οι ισότητες:

$$A + \overline{AB} = A + B, \quad \overline{ACD} + \overline{ACD} + ABD = \overline{AD} + BD$$

$$XY + XZ + YZ = \overline{\overline{XY}} + \overline{\overline{XZ}} + \overline{\overline{YZ}}$$

Λύση:

$$\alpha) A + \overline{AB} = \overline{\overline{A} \cdot \overline{AB}} = \overline{\overline{A} \cdot (A + \overline{B})} = \overline{\overline{A}A + \overline{A}\overline{B}} = \overline{\overline{A}\overline{B}} = A + B$$

$$\beta) \overline{ACD} + \overline{ACD} + ABD = \overline{AD}(\overline{C} + C) + ABD = \overline{AD} + ABD =$$

$$= (\overline{A} + AB)D = (\overline{A} + B)D = \overline{AD} + BD$$

γίνεται χρήση της (α)

$$\gamma) XY + XZ + YZ = (\overline{\overline{X} + \overline{Y}}) + (\overline{\overline{X} + \overline{Z}})(\overline{\overline{Y} + \overline{Z}}) =$$

$$\overline{(\overline{X} + \overline{Y})(\overline{X} + \overline{Z})(\overline{Y} + \overline{Z})} = \overline{\overline{XY} + \overline{YX} + \overline{XYZ} + \overline{YZ} + \overline{XZ} + \overline{XYZ} +$$

$$\overline{\overline{XY}} + \overline{\overline{YZ}} = \overline{\overline{XY}} + \overline{\overline{XZ}} + \overline{\overline{YZ}}$$

Άσκηση 1.7

Να απλοποιηθούν οι σχέσεις:

$$(1) \quad \overline{AB} + \overline{BC} + A$$

$$(2) \quad \overline{ABC} + \overline{ABC} \quad (3) \quad A(BC + CD + C + B + \overline{C}) + \overline{A}$$

Λύση:

$$(1) \quad \overline{AB} + \overline{BC} + A = \underbrace{\overline{AB} + A}_{A+B} + \overline{BC} = A + \underbrace{B + \overline{BC}}_{A+B+C} = A + B + C$$

$$(2) \quad \overline{ABC} + \overline{ABC} = (\overline{ABC}) \cdot (\overline{ABC}) = (A + B + \overline{C}) \cdot ((\overline{A} + \overline{B}) + \overline{C}) = \\ (A + B + \overline{C})(\overline{A} + \overline{B} + \overline{C}) = \overline{A}B + \overline{A}\overline{C} + \overline{B}\overline{C} + \overline{C} = \\ \overline{A}B + \overline{C}(A + B + 1) = \overline{A}B + \overline{C}$$

$$(3) \quad A(BC + CD + C + B + \overline{C}) + \overline{A} = A + \overline{A} = 1$$

Άσκηση 1.8

Να απλοποιηθούν οι παρακάτω εκφράσεις χρησιμοποιώντας τον πίνακα Karnaugh:

$$(1) \quad F = \overline{AD} + \overline{ABC} + \overline{ABD} + \overline{ABC} + \overline{ABCD}$$

$$(2) \quad F = \sum(1,2,3,5,10,11) - 4,9,13 \text{ (Don't carres)}$$

$$(3) \quad F = \prod(2,3,7,12,13,14,15)$$

Λύση:

(1)

		A B			
		0 0	0 1	1 1	1 0
C D	0 0	1	1	1	1
	0 1	1	1	1	
	1 1	1	1		
	1 0				1

$$F = \overline{AC} + \overline{BC} + \overline{AD} + \overline{ABD}$$

(2)

F C D \ A B		F			
		0 0	0 1	1 1	1 0
0 0	0 (0)	D (4)	0 ₍₁₂₎	0 (8)	
0 1	1 (1)	1 (5)	D (13)	D (9)	
1 1	1 (3)	0 (7)	0 (15)	1 (11)	
1 0	1 (2)	0 (6)	0 (14)	1 (10)	

$$F = \overline{C}D + \overline{B}C$$

(3)

C D \ A B		F			
		0 0	0 1	1 1	1 0
0 0	1	1	<u>0</u>	1	
0 1	1	1	<u>0</u>	1	
1 1	<u>0</u>	<u>0</u>	<u>0</u>	1	
1 0	<u>0</u>	1	<u>0</u>	1	

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

$$= \overline{A}\overline{C} + \overline{A}B\overline{D} + AB\overline{C}$$

$$F = (A + \overline{B})(A + \overline{C} + \overline{D})(A + B + \overline{C}) =$$

$$= \overline{A}\overline{C} + \overline{A}B\overline{D} + AB$$

Άσκηση 1.9

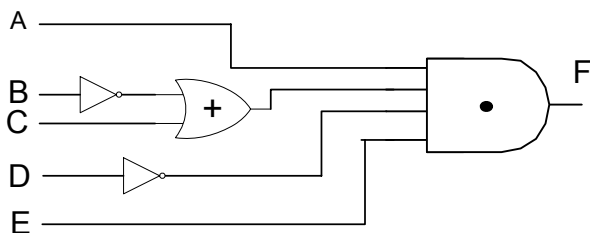
Να σχεδιαστεί το ψηφιακό κύκλωμα που υλοποιεί την έκφραση:

$$F = A \cdot (\overline{B} + C) \cdot \overline{D} \cdot E$$

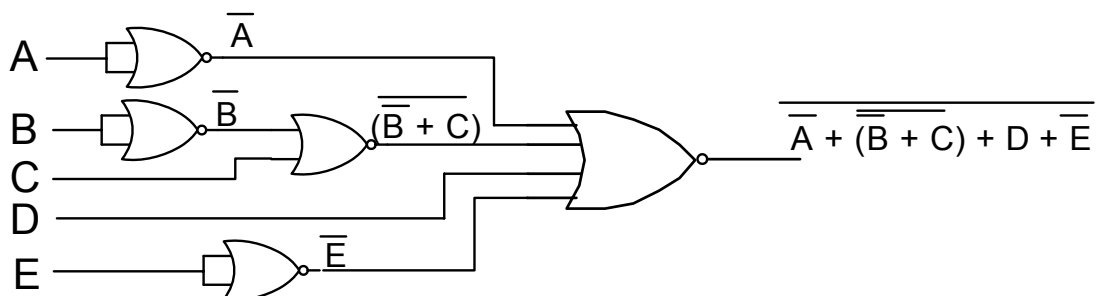
Κατόπιν να σχεδιαστεί το ίδιο κύκλωμα χρησιμοποιώντας μόνο πύλες **NOR**.

Λύση:

α)



$$\beta) A \cdot (\bar{B} + C) \bar{D} \cdot E = \overline{\overline{A(\bar{B} + C) \bar{D} \cdot E}} = \overline{\bar{A} + (\bar{\bar{B} + C}) + D + \bar{E}}$$



Άσκηση 1.10

Δίνεται ο πίνακας αληθείας:

A	B	C	F	
0	0	0	0	
0	0	1	1	$\bar{\bar{A}}\bar{B}\bar{C}$
0	1	0	0	
0	1	1	0	
1	0	0	1	$\bar{A}\bar{B}\bar{C}$
1	0	1	1	$\bar{A}\bar{B}C$
1	1	0	1	$\bar{A}B\bar{C}$
1	1	1	1	$\bar{A}BC$

1. Χρησιμοποιώντας τον πίνακα Karnaugh να απλοποιηθεί η λογική συνάρτηση χρησιμοποιώντας και τις δύο κανονικές μορφές (αθροίσματος και γινομένου)
2. Να υλοποιηθούν τα κυκλώματα και στις δύο μορφές
3. Να σχεδιαστούν τα ισοδύναμα των παραπάνω κυκλωμάτων χρησιμοποιώντας μόνο πύλες **NAND** και **NOR** αντίστοιχα

Λύση:

C \ AB	0 0		0 1		1 1		1 0	
	0	0	0	1	1	1	1	0
0	0	0	0	0	1	1	1	1
1	1	1	0	0	1	1	1	1

α)

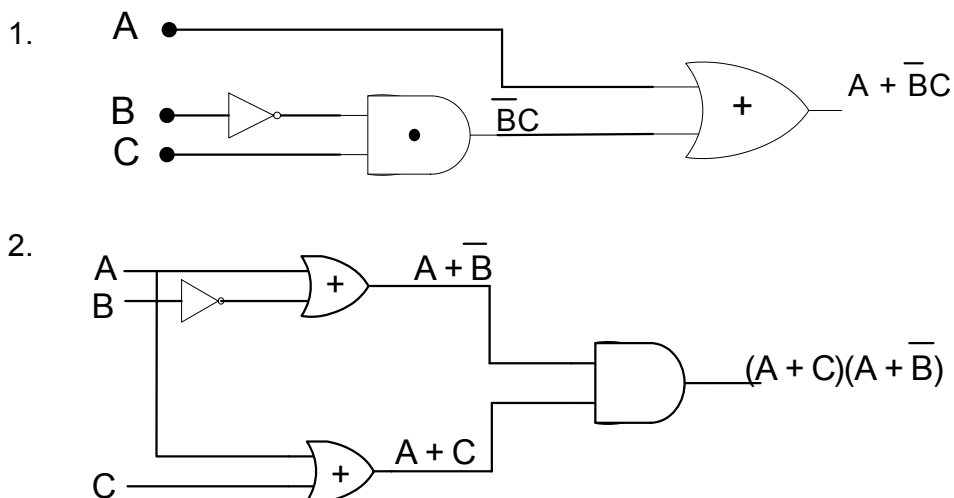
- Κανονική μορφή αθροίσματος:

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

- Κανονική μορφή γινομένου:

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

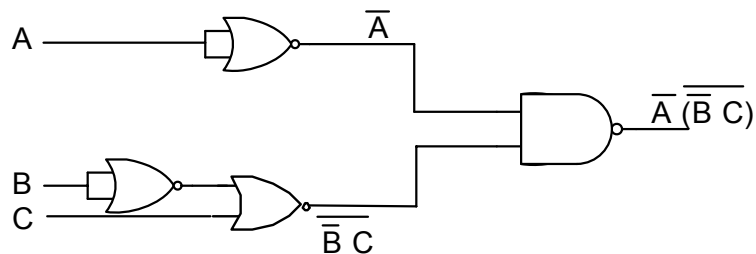
β)



γ)

1. NAND

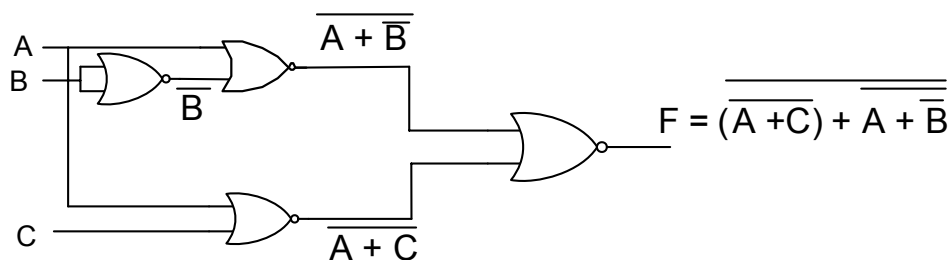
$$A + \overline{B}C = \overline{\overline{A + \overline{B}C}} = \overline{\overline{A} \overline{\overline{B}C}} = \overline{\overline{A} (\overline{\overline{B}C})}$$



2.

NOR

$$\overline{(A + C)(A + \overline{B})} = \overline{(A + C)} + \overline{(A + \overline{B})}$$



Κυκλώματα ακολουθιακής λογικής

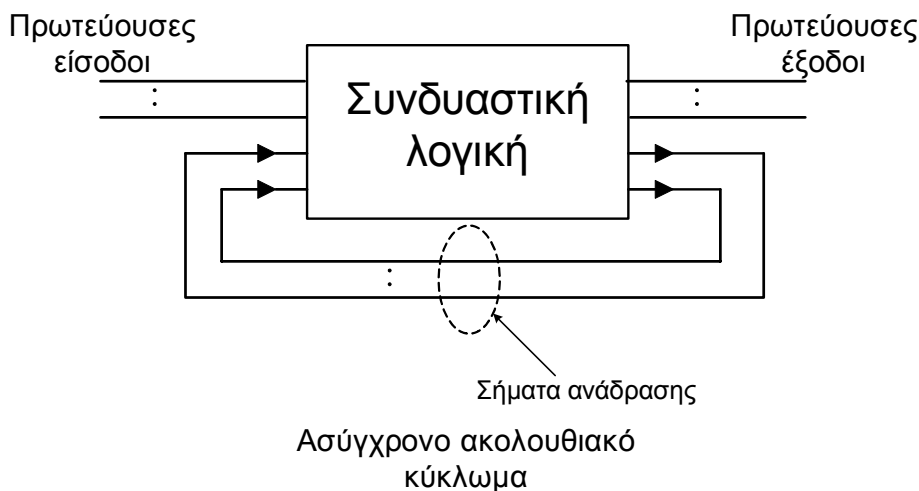
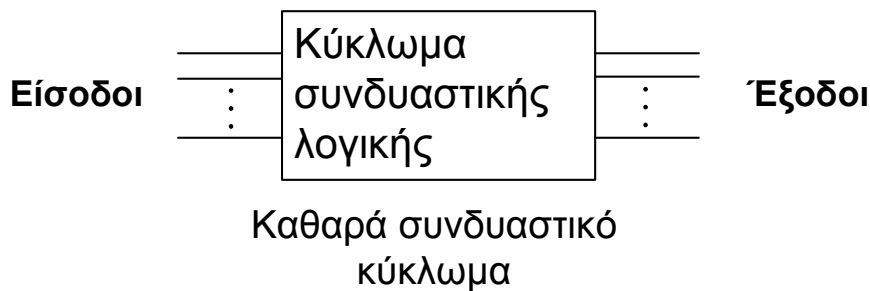
Στα κυκλώματα συνδυαστικής λογικής, τα οποία εξετάσθηκαν στα προηγούμενα κεφάλαια, οι τιμές της εξόδου σ' οποιαδήποτε χρονική στιγμή είναι συνάρτηση μόνο των τιμών της εισόδου της ίδιας χρονικής στιγμής.

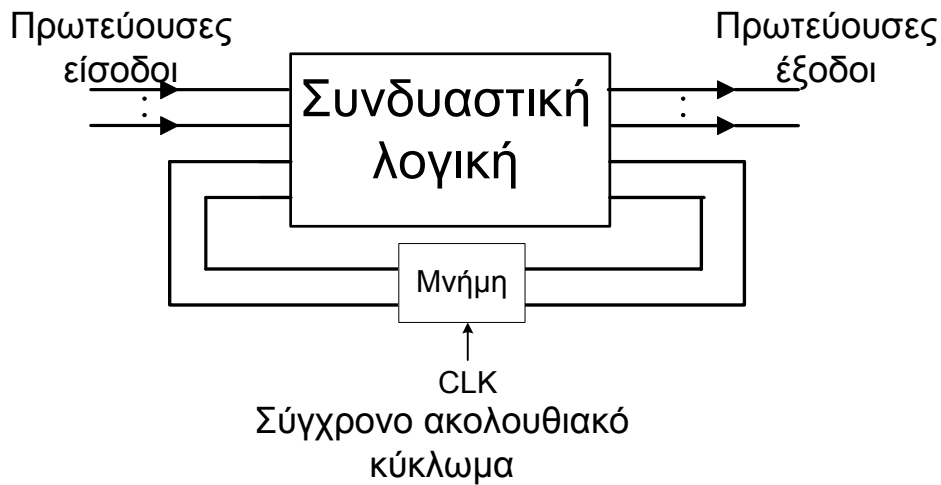
Συνδυαστική λογική: τιμές εξόδου = f (παρούσες τιμές εισόδου)

Στην ακολουθιακή λογική οι τιμές της εξόδου των κυκλωμάτων επηρεάζονται από τις παρούσες αλλά και τις προηγούμενες τιμές της εισόδου:

Ακολουθιακή λογική: τιμές εξόδου = f (παρούσες + προηγούμενες τιμές εισόδου)

Γενικές μορφές κυκλωμάτων





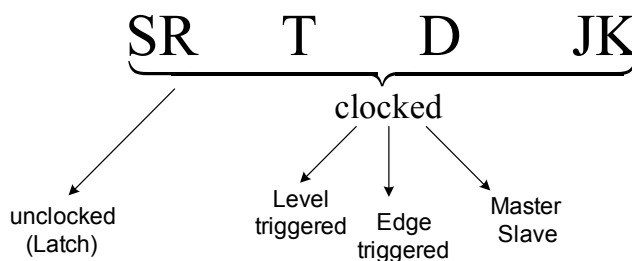
Τα ακολουθιακά κυκλώματα «θυμούνται» μέσω της σύνδεσης της ανάδρασης. Δύο είναι οι κύριες κατηγορίες των ακολουθιακών κυκλωμάτων:

Ασύγχρονα: Αλλάζουν κατάσταση σύμφωνα με τις αλλαγές των εισόδων τους. Απαιτούνται ειδικές τεχνικές σχεδιασμού.

Σύγχρονα: Τα σήματα ανάδρασης διακόπτονται από καταχωρητές που σκανδαλίζονται από παλμούς ρολογιού. Συνεπώς η κατάστασή του κυκλώματος αλλάζει σύμφωνα με τους παλμούς του ρολογιού. Η κατάσταση του κυκλώματος ορίζεται από το περιεχόμενο των στοιχείων της μνήμης.

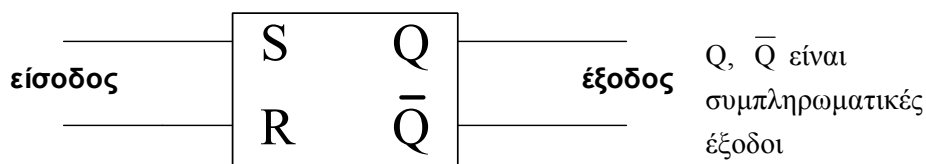
Flip – Flops

Τα flip-flops διαθέτουν δύο σταθερές καταστάσεις (1 και 0), και παρέχουν μνήμη που αποθηκεύει πληροφορία ενός (1) bit. Υπάρχουν διάφοροι τύποι flip-flops, οι οποίοι ταξινομούνται σύμφωνα με τον τρόπο λειτουργίας τους.



Τα flip-flops αποτελούν τα βασικά δομικά στοιχεία για το σχεδιασμό των ακολουθιακών κυκλωμάτων.

To SR (Set-Reset) flip-flop (ff)



Πίνακας αληθείας του SR-ff

S	R	Q_{n+1}	
0	0	Q_n	επόμενη κατάσταση παρούσα κατάσταση
0	1	0	
1	0	1	
1	1	-	ακαθόριστη έξοδος (μη επιτρεπτή περίπτωση)

Για την υλοποίηση του SR-ff δημιουργούνται ο εκτεταμένος πίνακας αληθείας και οι πίνακες Karnaugh, όπου το Q_n (παρούσα κατάσταση εξόδου) χρησιμοποιείται ως μεταβλητή εισόδου:

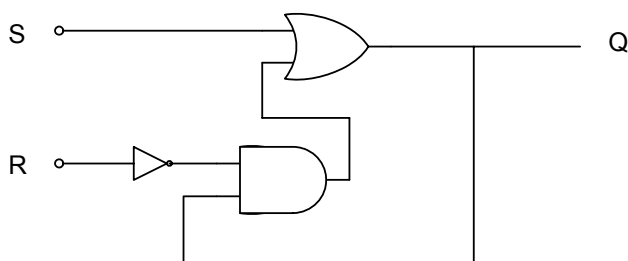
R	S	Q_n	Q_{n+1}	
0	0	0	0	Μνήμη
0	0	1	1	Μνήμη
0	1	0	1	Set
0	1	1	1	Set
1	0	0	0	Reset
1	0	1	0	Reset
1	1	0	x	}
1	1	1	x	

Πίνακας-K

		Q_n			
		00	01	11	10
R	0	0	1	1	1
	1	0	0	X	X

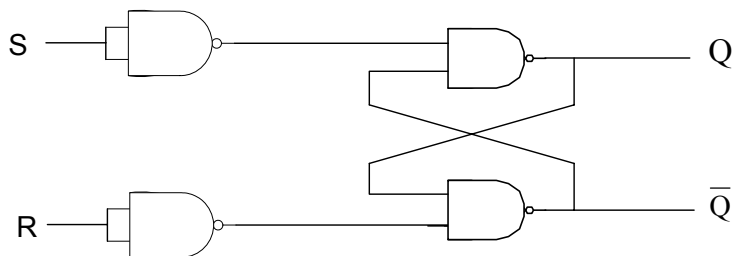
$Q_{n+1} = S + \bar{R}Q_n$
 ↗
 Χαρακτηριστική εξίσωση

Το κύκλωμα που υλοποιεί την παραπάνω σχέση είναι:



Χρησιμοποιώντας το θεώρημα De Morgan, η σχέση για σχεδιασμό με πύλες NAND έχει ως εξής:

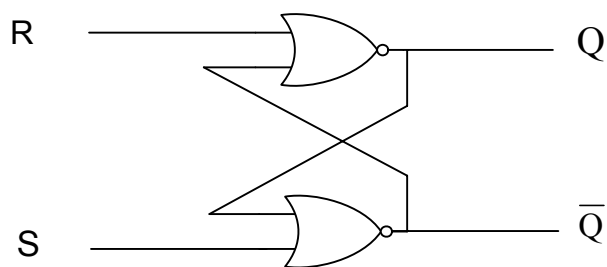
$$Q_{n+1} = \overline{\bar{S} \cdot \bar{R}Q_n}$$



Χρησιμοποιώντας έναν εκτεταμένο πίνακα αληθείας και τους πίνακες-K που συσχετίζουν τα $\bar{Q}_{n+1}$, $\bar{Q}_n$, R και S προκύπτει μια άλλη χαρακτηριστική εξίσωση η οποία δίνεται από τη σχέση:

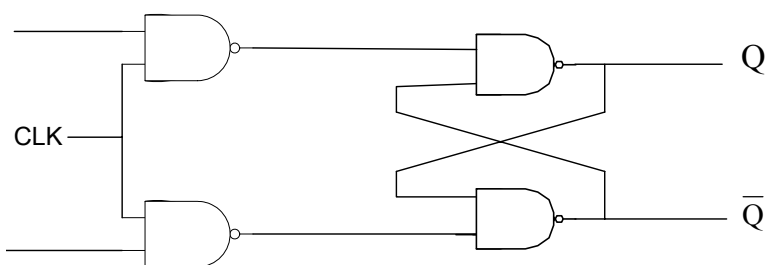
$$\bar{Q}_{n+1} = R + \bar{S}\bar{Q}_n$$

από την οποία εξάγεται η σχέση για την υλοποίηση του ff με πύλες NOR



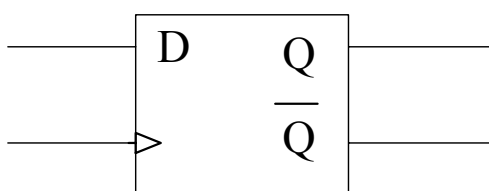
$$Q_{n+1} = \overline{R + S + Q_n}$$

To Clocked SR - ff



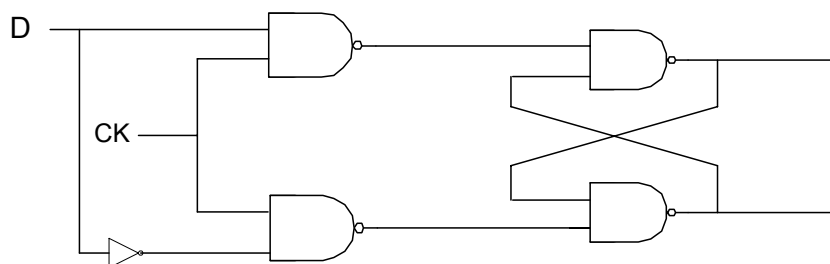
Το σήμα του ρολογιού δρα ως ένα σήμα που ενεργοποιεί το SR-ff και επομένως οι έξοδοι του μπορεί να αλλάζουν μόνο όταν ο παλμός του ρολογιού είναι 1.

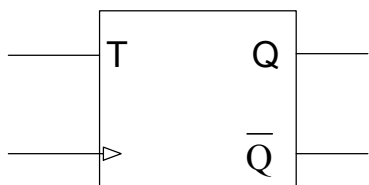
Flip – Flop τύπου D



Ck	D	Q_{n+1}
0	0	Q_n
0	1	Q_n
1	0	0
1	1	1

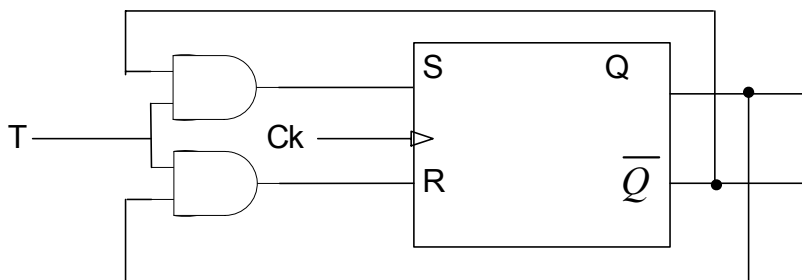
Υλοποίηση:



Flip – Flop τύπου T (Toggle ff)

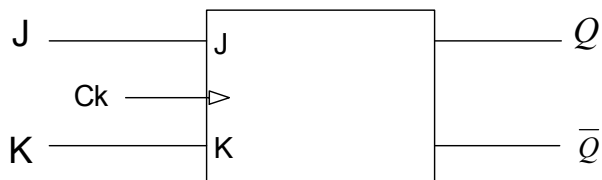
Ck	T	Q_{n+1}
0	0	Q_n
0	1	Q_n
1	0	Q_n
1	1	$\overline{Q_n}$

Υλοποίηση:



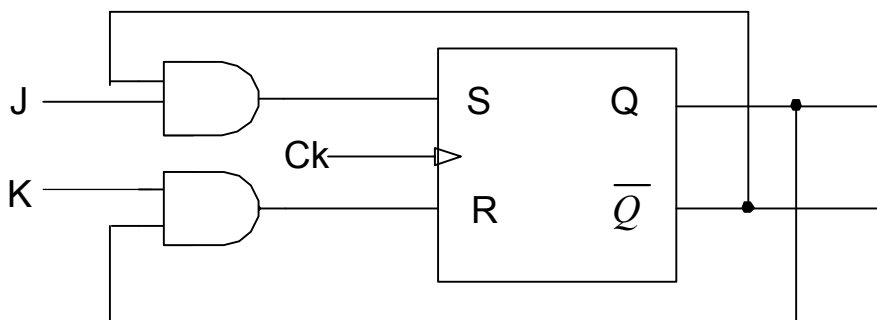
$$S = T\overline{Q}$$

$$R = TQ$$

Flip – Flop τύπου JK

Ck	J	k	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	$\overline{Q_n}$

Υλοποίηση:



$$S = J\overline{Q}$$

$$R = KQ$$

Υλοποίηση σύγχρονων flip-flops με όρους SR-ff

Παράδειγμα 2.1:

Να γίνει η υλοποίηση ενός T-ff σε όρους ενός SR-ff.

Απάντηση:

Κατ' αρχήν δημιουργείται ένας πίνακας συσχέτισης των εισόδων ενός T-ff (C_k , T , Q_n) και των αντίστοιχων εισόδων του SR-ff που έχουν το ίδιο αποτέλεσμα στην κατάσταση Q_{n+1}

Πίνακας συσχέτισης:

C_k	T	Q_n	Q_{n+1}	S	R
0	0	0	0	x	x
0	0	1	1	x	x
0	1	0	0	x	x
0	1	1	1	x	x
1	0	0	0	0	x
1	0	1	1	x	0
1	1	0	1	1	0
1	1	1	0	0	1

Είσοδοι που απαιτούνται για να έχουμε την ίδια αλληλουχία $Q_n \rightarrow Q_{n+1}$

Καταστάσεις μνήμης ($C_k = 0 \Rightarrow Q_{n+1} = Q_n$)

Κατόπιν εξάγονται οι εξισώσεις των S , R με όρους C_k , T και Q_n χρησιμοποιώντας τους πίνακες – K

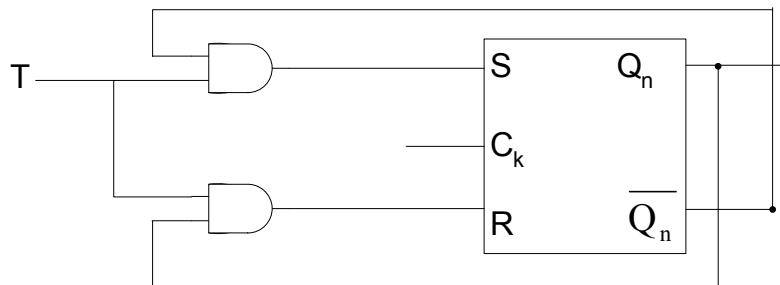
		$T Q_n$			
		00	01	11	10
C_k	0	x	x	x	x
	1	0	x	0	1

$$S = T \overline{Q_n}$$

		$T Q_n$			
		00	01	11	10
C_k	0	x	x	x	x
	1	x	0	1	0

$$R = T Q_n$$

Τελικό κύκλωμα:



Σημείωση: Επειδή ουσιαστικά μας ενδιαφέρουν μόνο οι περιπτώσεις που το $C_k=1$, μπορεί να αγνοηθεί η παράμετρος C_k , ώστε να προκύπτουν πιο απλοί πίνακες

Το παραπάνω παράδειγμα μπορεί να λυθεί απλούστερα ως εξής:

T	Q_n	Q_{n+1}	S	R
0	0	0	0	X
0	1	1	X	0
1	0	1	1	0
1	1	0	0	1

		T	
		0	1
Q_n	0	0	1
	1	X	0

$S = T\overline{Q_n}$

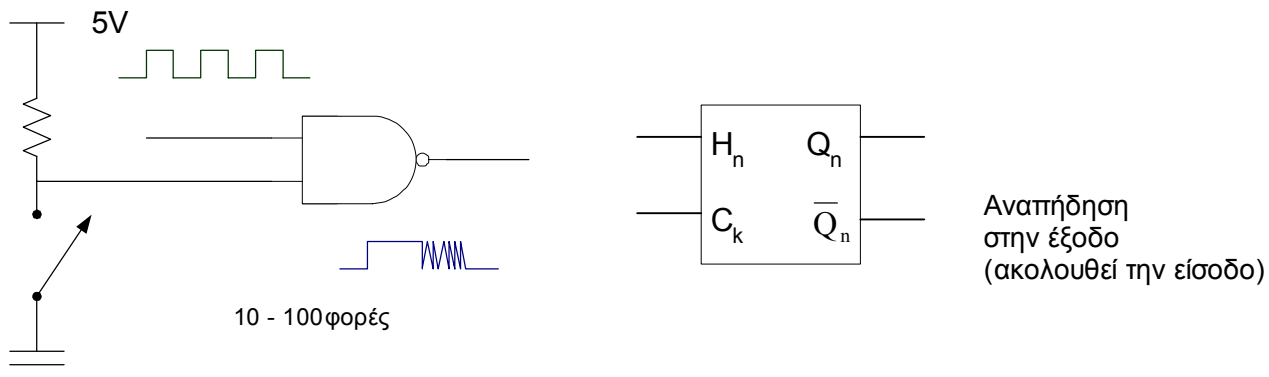
		T	
		0	1
Q_n	0	X	0
	1	0	1

$R = TQ_n$

Κατά παρόμοιο τρόπο υλοποιούνται τα D-ff και JK-ff από το SR-ff, καθώς επίσης και η υλοποίηση οποιουδήποτε σύγχρονου ff με όρους κάποιου άλλου ff.

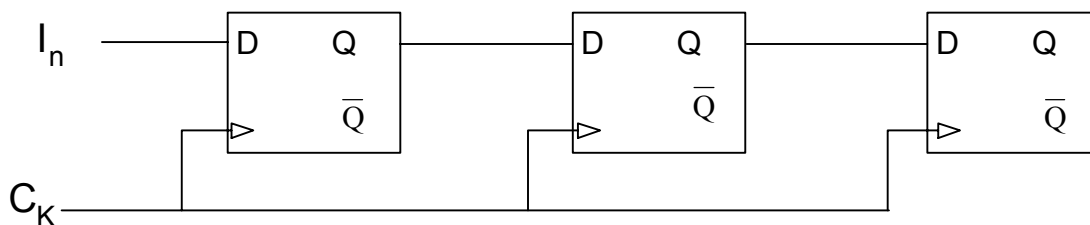
Προβλήματα που σχετίζονται με απλά σύγχρονα ff

1. Αναπήδηση εισόδου



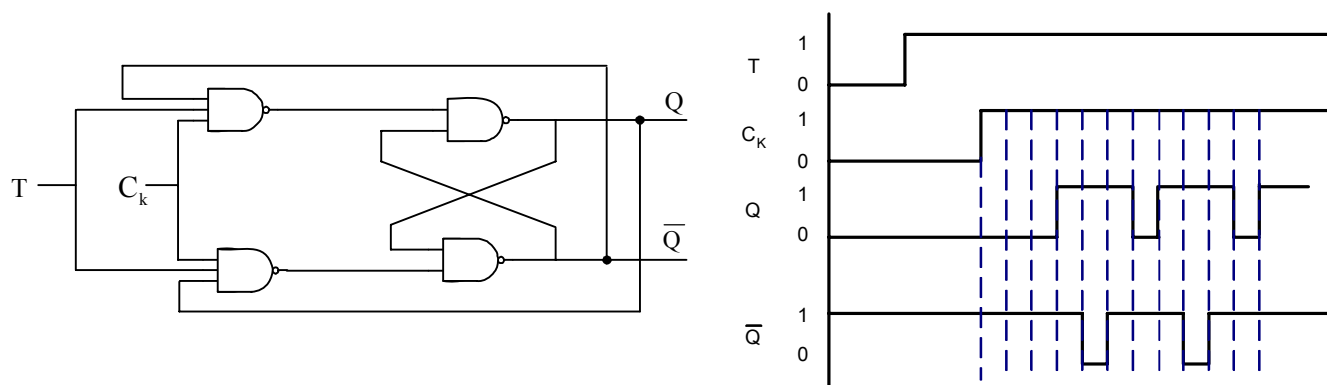
2. Κακή λειτουργία κυκλωμάτων που χρησιμοποιούν διαδοχικά ff

π.χ. οι καταχωρητές μετατόπισης



3. Ταλαντώσεις σε ff λόγω ανάδρασης

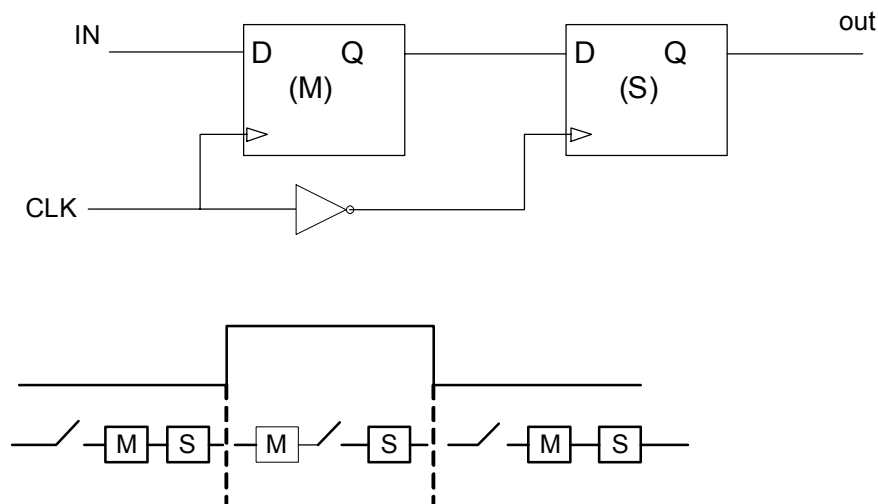
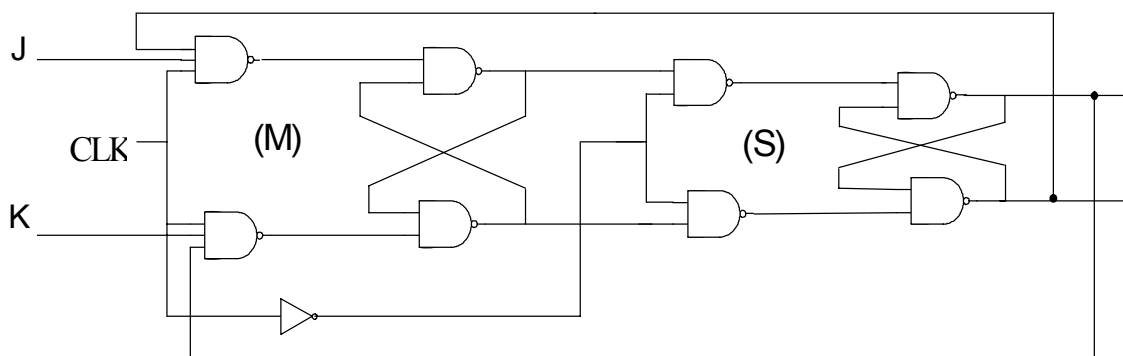
π.χ. στο T-ff



Μέθοδοι επίλυσης των προβλημάτων

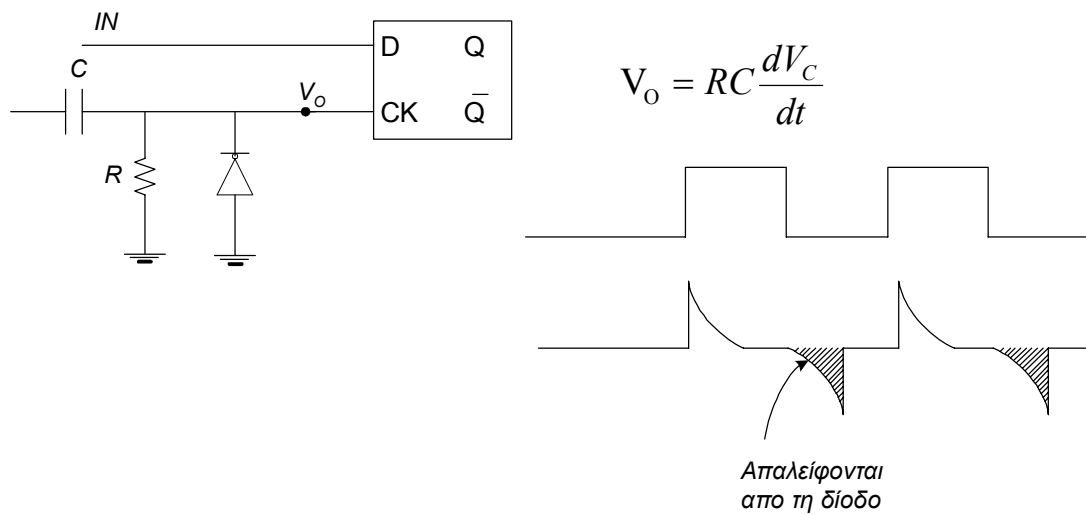
Χρήση "Αφέντη – Σκλάβου" ff (Master – Slave ff)

Ρύθμιση του παλμού σκανδαλισμού (edge triggered devices)

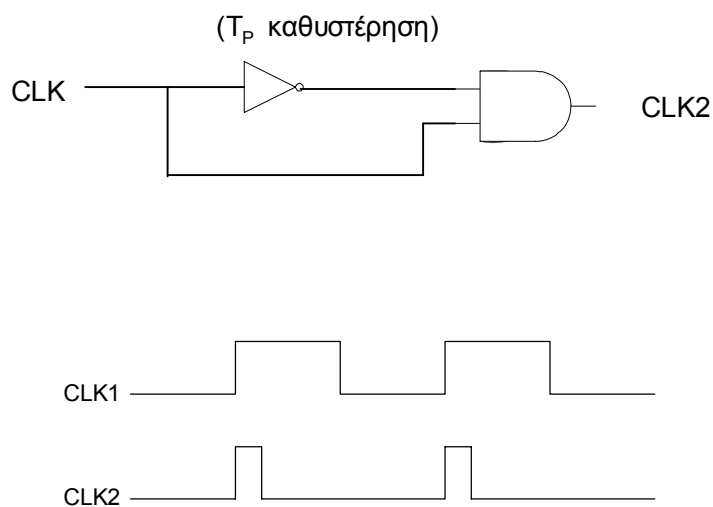
*Διάταξη "Αφέντη – Σκλάβου" ff***π.χ. χρησιμοποιώντας D ff***διάταξη του "Αφέντη – Σκλάβου" ff χρησιμοποιώντας JK-ff**Δημιουργία εσωτερικού ρολογιού για παραγωγή παλμού βραχείας διάρκειας*

Αναφέρονται δύο μέθοδοι:

α)



β)



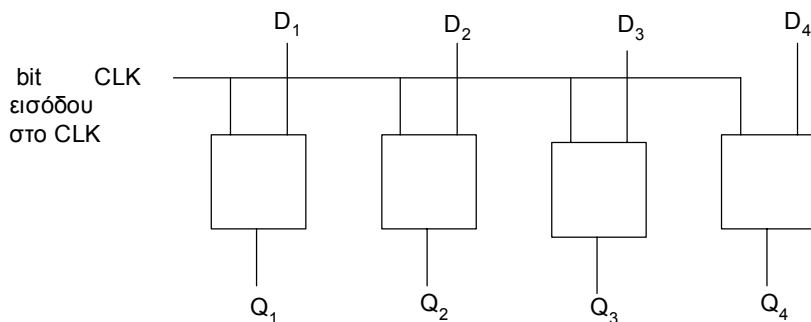
Εφαρμογές ff

Τυπικές εφαρμογές των ffs είναι:

1. Απλοί καταχωρητές
2. Κυκλώματα καταχωρητών ολίσθησης
3. Μετρητές

Απλοί καταχωρητές

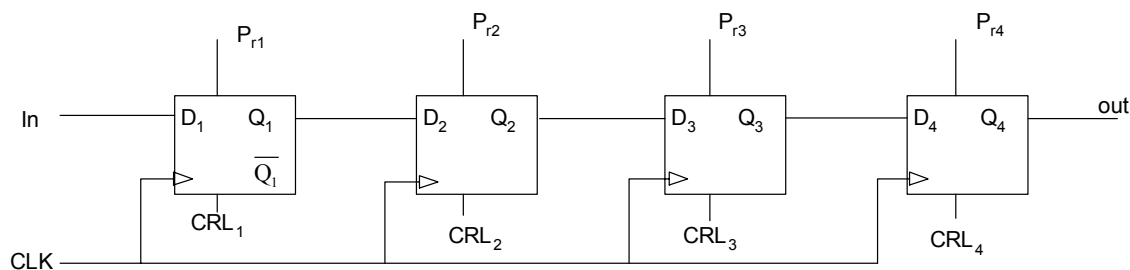
Π.χ. Καταχωρητής 4 bit



Μια ομάδα από m ff's αποθηκεύει m bits πληροφορίας. Τέτοιες ομάδες συχνά χρησιμοποιούνται για αποθήκευση πληροφορίας σε συστήματα υπολογιστών. (Προσωρινή αποθήκευση δεδομένων)

Καταχωρητές ολίσθησης

Για παράδειγμα καταχωρητής ολίσθησης 4 bit χρησιμοποιώντας D-ff



DATA IN	Q_1	Q_2	Q_3	Q_4
1	0	0	0	0
1	1	0	0	0
0	1	1	0	0
0	0	1	1	0
1	0	0	1	1
1	1	0	0	1

Ο παραπάνω καταχωρητής είναι γνωστός και ως καταχωρητής SISO (Serial In Serial Out). Εάν σ' ένα SISO καταχωρητή το Q_1 είναι το πιο σημαντικό ψηφίο και το Q_4 το πιο χαμηλής σημαντικότητας ψηφίο (MSB και LSB αντίστοιχα), τότε η μετατόπιση γίνεται προς τα δεξιά.

Στην αντίθετη περίπτωση, δηλαδή το $Q_4 \rightarrow \text{MSB}$ και το $Q_1 \rightarrow \text{LSB}$, τότε η μετατόπιση γίνεται προς τα αριστερά.

Σημείωση: Κάθε είσοδος 0 στον καταχωρητή μετατόπισης έχει ως αποτέλεσμα

- Τη διαίρεση με το 2 εάν είναι ο καταχωρητής μετατόπισης προς τα δεξιά και
- πολλαπλασιασμό με το 2 εάν είναι καταχωρητής μετατόπισης προς τα αριστερά

Π.χ.

$$0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 = 9$$

0 καταχωρητής μετατόπισης προς τα δεξιά $\left(\frac{9}{2} = 4 \right)$ ακέραιο μέρος

$$\rightarrow 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 = 4$$

καταχωρητής μετατόπισης προς τα αριστερά

$$0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 = 18 \leftarrow 0 \ (9 \cdot 2 = 18)$$

Κυκλώματα μετρητών

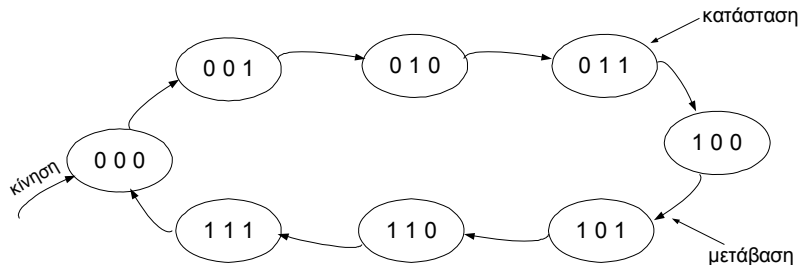
Οι μετρητές είναι κυκλώματα τα οποία στην έξοδό τους επιτυγχάνουν μια καθορισμένη ακολουθία σκανδαλιζόμενα από μεταβολές σημάτων (edges), ή παλμούς που παράγονται από κύκλωμα χρονισμού στην είσοδό τους.

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0
...	...	...

↓
Ακολουθία
αρίθμησης (up counter)

- (1) Κάθε bit του μετρητή χρειάζεται ένα ff
- (2) Κατά την εφαρμογή ενός παλμού στην είσοδο του μετρητή, ο μετρητής αλλάζει κατάσταση. (Στο παράδειγμα η κατάσταση ορίζεται από τις εξόδους A, B και C)

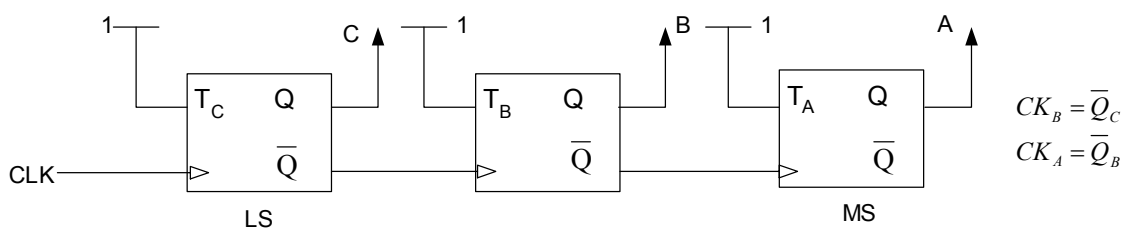
Η λειτουργία των μετρητών μπορεί να ορισθεί πλήρως με ένα διάγραμμα κατάστασης



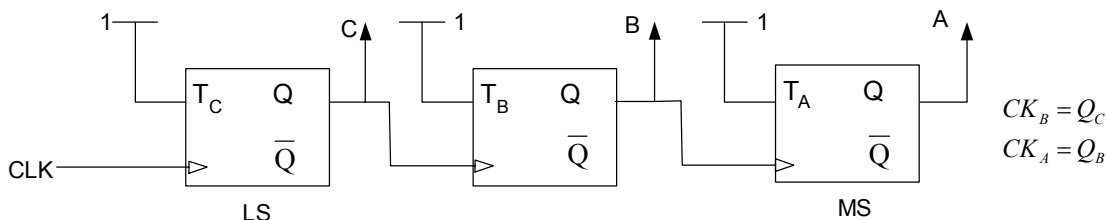
Δύο είναι οι κύριες κατηγορίες των μετρητών

- ο Οι ασύγχρονοι μετρητές (ή Ripple Counters)
- ο και οι σύγχρονοι μετρητές

Ασύγχρονοι μετρητές



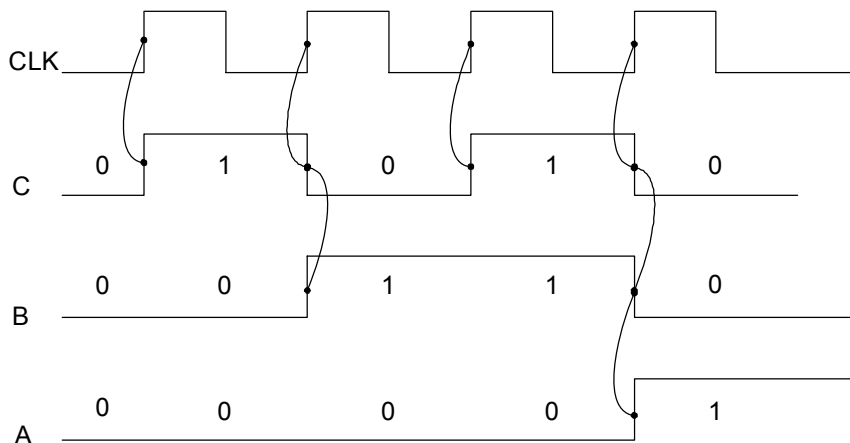
Up counter



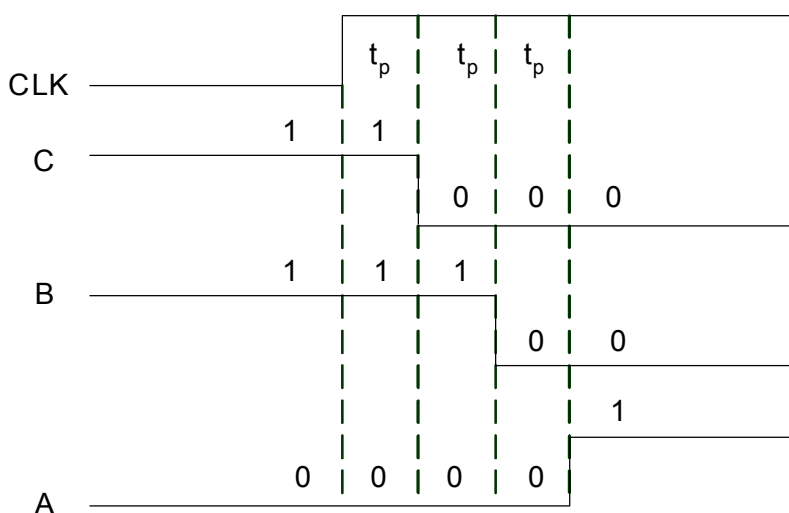
Down counter

Σημείωση : Στους ασύγχρονους μετρητές μόνο το LSD ff δέχεται παλμό από το εξωτερικό ρολοί, ενώ όλα τα υπόλοιπα ff's στην αλυσίδα σκανδαλίζονται από την έξοδο του ff της προηγούμενης βαθμίδας.

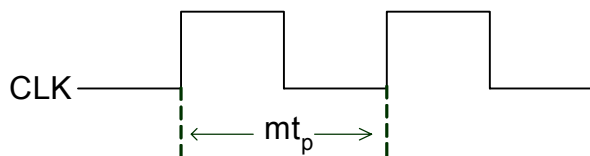
Εάν αγνοηθούν οι καθυστερήσεις στις πύλες, τα διαγράμματα εξόδου του CLK, A, B και C έχουν ως εξής:



Λόγω καθυστέρησης στα ff, κατά τη μετάβαση μιας κατάστασης σε μια άλλη, απαιτείται κάποιο χρονικό διάστημα. Στα ασύγχρονα κυκλώματα των μετρητών, κατά τη μετάβαση μιας κατάστασης σε μια άλλη, παρουσιάζονται ενδιάμεσες ανεπιθύμητες καταστάσεις. Π.χ. θεωρήστε μια μετάβαση από τον αριθμό 3 (0 1 1) στον αριθμό 4 (1 0 0) σ' ένα ασύγχρονο μετρητή 3 bits



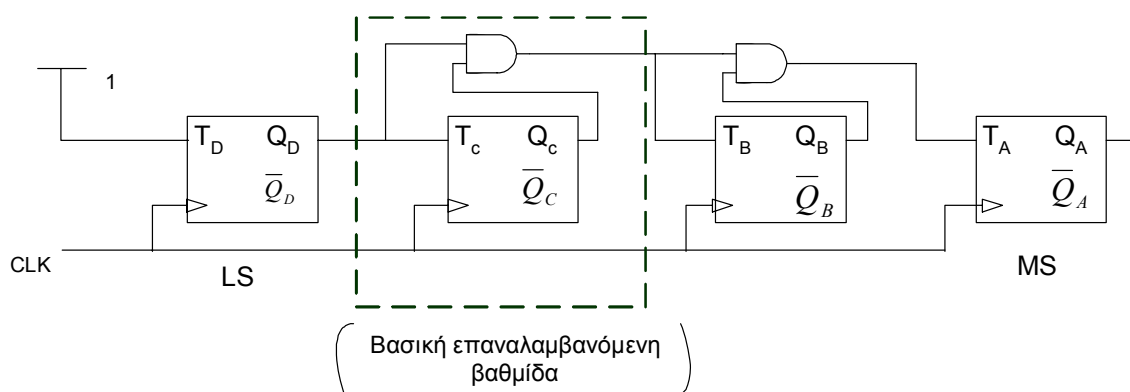
Παρατηρείστε ότι για να μεταβείτε από την 0 1 1 στην 1 0 0 δημιουργούνται οι καταστάσεις 0 1 1 $\rightarrow$ 0 1 0 $\rightarrow$ 0 0 0 . Γενικά εάν υπ'άρχουν m βαθμίδες ff θα υπάρχει μια συνολική καθυστέρηση mt_p . Συνεπώς θα πρέπει να εξασφαλιστεί ώστε η περίοδος του παλμού του ρολογιού να είναι μεγαλύτερη του mt_p .



Σύγχρονοι μετρητές

Για να αποφύγουμε τα προβλήματα των ασύγχρονων μετρητών χρησιμοποιούνται σύγχρονοι μετρητές. Στους σύγχρονους μετρητές, όλα τα ff's σκανδαλίζονται ταυτόχρονα με τον ίδιο παλμό.

Η βασική υλοποίηση του σύγχρονου μετρητή δίδεται από το κύκλωμα:



$$T_D = 1, T_C = Q_D \quad T_B = Q_C Q_D \quad T_A = Q_B * Q_C * Q_D$$

Το κύκλωμα θα λειτουργεί ως ένας (up-counter) εάν οι καταστάσεις A B C D ληφθούν από τα Q_A Q_B Q_C Q_D αντίστοιχα, και ως ένας (Down Counter) εάν οι καταστάσεις A B C D ληφθούν από τα

$$\overline{Q_A}, \overline{Q_B}, \overline{Q_C}, \text{ και } \overline{Q_D}$$

Πίνακας εξόδου του (up-counter)

Παρούσα κατάσταση έξοδοι				Είσοδοι ff				Επόμενη κατάσταση			
A	B	C	D	T _A	T _B	T _C	T	Q _A ⁺	Q _B ⁺	Q _C ⁺	Q _B ⁺
Q _A	Q _B	Q _C	Q _D	D							
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	1	0	0	1	0
0	0	1	0	0	0	0	1	0	0	1	1
0	0	1	1	0	0	0	1	0	1	0	0
0	1	0	0	0	1	1	1	0	1	0	1
				0	0	0	1				
.						.				.	
.						.				.	

Βασικοί ορισμοί για τους μετρητές

MODULUS Ο αριθμός των καταστάσεων που διατρέχει ο μετρητής έως ότου αρχίσουν να επαναλαμβάνονται

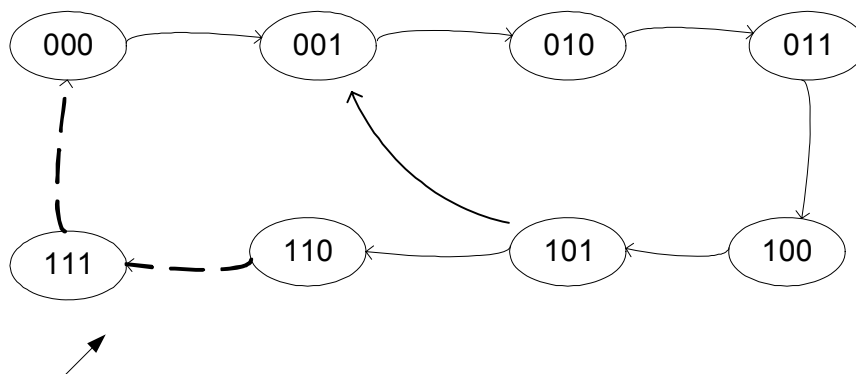
MODULUS M ή MOD- M Ένας μετρητής με M καταστάσεις

MAXIMUM MODULUS ($M_{\max}$) Ο μέγιστος αριθμός των καταστάσεων που μπορεί να δημιουργηθούν από τα n bits του μετρητή ($M_{\max} = 2^n$)

Μετρητής πλήρους ακολουθίας : → ένας μετρητής του οποίου το MODULUS είναι ίδιο με το $M_{\max}$

Ο μετρητής του παραδείγματος παραπάνω είναι πλήρους ακολουθίας με $n=4$ και άρα MOD-16

Μετρητής τμηματικής ακολουθίας: → Ένας μετρητής του οποίου το MODULUS είναι μικρότερο από το $M_{\max}$ ($M < 2^n$ για έναν n bits μετρητή). Δηλαδή σ' ένα MOD-5 τριών bits μετρητή χρησιμοποιούνται μόνο 5 από τις 8 δυνατές καταστάσεις.



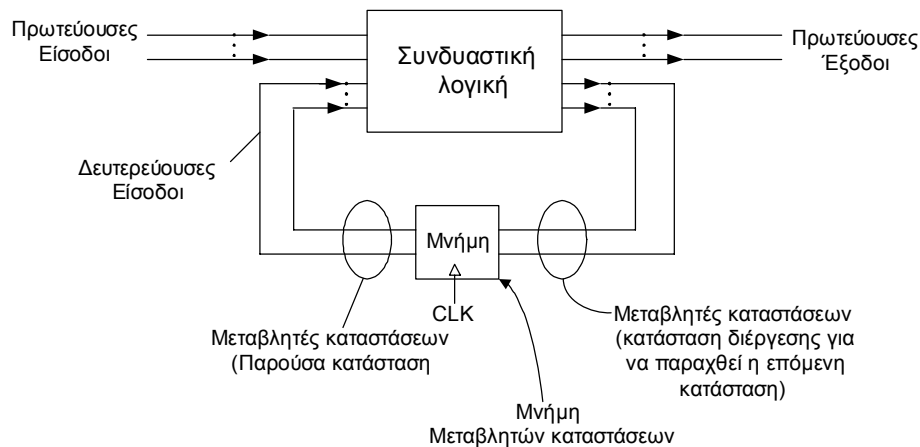
Καταστάσεις που
δεν χρησιμοποιούνται

Τροποποίηση του βασικού σύγχρονου μετρητή για τη δημιουργία ενός MOD- M μετρητή

Θα μελετηθεί πολύ καλά η σχετική άσκηση από τη 2^η ενότητα των ασκήσεων.

Εισαγωγή στο σχεδιασμό ψηφιακών κυκλωμάτων με διαγράμματα καταστάσεων (state diagrams)

- Μπορούν να χρησιμοποιηθούν για το σχεδιασμό σύγχρονων ακολουθιακών κυκλωμάτων
- Γενική μορφή σύγχρονου ακολουθιακού κυκλώματος που χρησιμοποιείται σε διαγράμματα καταστάσεων (Δ.Κ.)

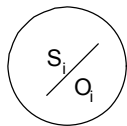


Ένα σύστημα με 'N' μεταβλητές καταστάσεων χρειάζεται N flip-flops για μνήμη.

Διαγράμματα καταστάσεων

- Είναι η γραφική παράσταση της λειτουργίας ενός ακολουθιακού κυκλώματος

Στοιχείο κατάστασης:



Όπου S_i είναι η περιγραφή της κατάστασης i και O_i είναι οι έξοδοι που σχετίζονται με την κατάσταση S_i

Στοιχείο μετάβασης καταστάσεων:

Η παραπάνω γραφική αναπαράσταση των διαγραμμάτων κατάστασης είναι γνωστή και ως παράσταση Moore.

Παράδειγμα 2.2

Να σχεδιαστεί το Δ.Κ. για ένα ακολουθιακό κύκλωμα MOD-3 με εξωτερική γραμμή ελέγχου για reset ή clear των flip-flops.

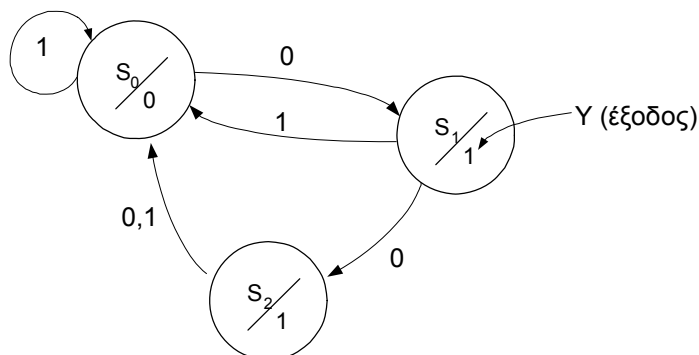
Απάντηση:

Όταν η γραμμή ελέγχου “clear” ενεργοποιείται τότε τα flip-flops αποκτούν τη μηδενική κατάσταση (cleared). Συνεπώς εάν θεωρηθεί το σήμα στην είσοδο της γραμμής ελέγχου ως μεταβλητή εισόδου τότε

Είσοδος “CLR” = 1 τα flip-flops επανέρχονται στην κατάσταση μηδέν

Είσοδος “CLR” = 0 το κύκλωμα λειτουργεί κανονικά μεταβαίνοντας από κατάσταση σε κατάσταση

Στο παράδειγμά υπάρχουν τρεις καταστάσεις τις οποίες συμβολίζουμε με S_1, S_2, S_3 και για τις οποίες απαιτείται να χρησιμοποιηθούν (τουλάχιστον) δύο (2) μεταβλητές καταστάσεων ή 2 flip-flops. Επειδή για $CLR = 1$ τα flip-flops μηδενίζονται, μπορεί (πιθανότατα) να θεωρηθεί ότι $S_0 = 00$ (Τα Q_1 και Q_2 των δύο flip-flops). Βέβαια θα πρέπει να ορισθούν και οι S_1 και S_2 , οι οποίες, εάν δεν υπάρχουν άλλα στοιχεία, θα μπορούσαν να παρασταθούν με $S_1=01$ και $S_2=10$ (δηλαδή η κανονική δυαδική ακολουθία)

Δ.Κ.**Πίνακες Καταστάσεων**

Είναι η παράσταση του διαγράμματος καταστάσεων υπό μορφή πίνακα.

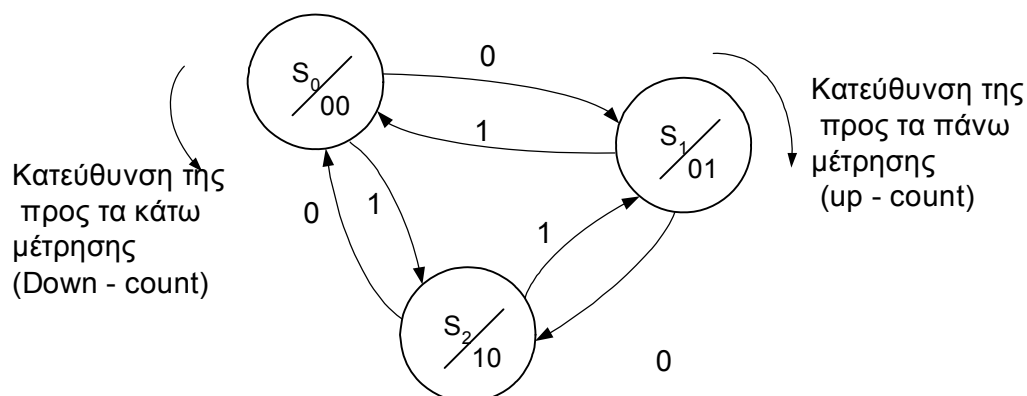
Σ' αυτόν περιλαμβάνονται η παρούσα κατάσταση (PS) η επόμενη κατάσταση (NS), οι συνθήκες εισόδου που προκαλούν τη μετάβαση και οι αντίστοιχες τιμές εξόδου.

Παράδειγμα 2.3

Μετρητής MOD-3 (UP/DOWN)

είσοδοι : DIR = 1 : Μετράει μπρος τα κάτω (Down)
DIR = 0 : Μετράει μπρος τα επάνω (Up)

έξοδοι : A,B



Είσοδοι (DIR)	PS	MS	Έξοδοι A B	
0	S ₀	S ₁	0	0
1	S ₀	S ₂	0	0
0	S ₁	S ₂	0	1
1	S ₁	S ₀	0	1
0	S ₂	S ₀	1	0
1	S ₂	S ₁	1	0

1η Μορφή

PS	NS DIR=0	NS DIR=1	Έξοδοι A B	
S ₀	S ₁	S ₂	0	0
S ₁	S ₂	S ₀	0	1
S ₂	S ₀	S ₁	1	0

2η Μορφή

Πίνακας διέγερσης και εξισώσεις διέγερσης

Ο πίνακας διέγερσης περιλαμβάνει τις τιμές της εισόδου που απαιτούνται για κάθε μετάβαση μεταξύ των καταστάσεων, για επιλεγμένο τύπο flip-flop. Ο προηγούμενος πίνακας καταστάσεων δημιουργήθηκε ανεξάρτητα από τον τύπο του flip-flop, ή τον τύπο της συνδυαστικής λογικής που επρόκειτο να χρησιμοποιηθεί. Άρα και αποφασισθεί ο τύπος του flip-flop που πρόκειται να χρησιμοποιηθεί, προσδιορίζονται οι μεταβλητές καταστάσεων και κατασκευάζεται ο πίνακας διέγερσης.

Σ' ένα τυπικό ακολουθιακό κύκλωμα θα μπορούσαν να χρησιμοποιηθούν είτε D-ffs είτε JK-ffs.

D-ffs

Απλούστερη επεξεργασία
Κατά το σχεδιασμό

JK-ffs

Λιγότερο σύνθετο
τελικό κύκλωμα

Παράδειγμα 2.4

Να γίνει ο σχεδιασμός του Μετρητή MOD-3 (UP/DOWN) χρησιμοποιώντας flip-flops τύπου D.

Απάντηση

Κατ' αρχήν υπάρχουν τρεις καταστάσεις S₀, S₁ και S₂, και των οποίων ως μεταβλητές ορίζονται οι έξοδοι Q₁ και Q₂ των (2) ffs τύπου D.

Κατάσ.	Q1	Q2
S ₀	0	0
S ₁	0	1
S ₂	1	0

Πίνακας ορισμού των καταστάσεων

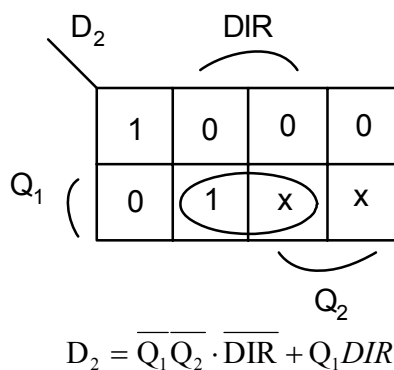
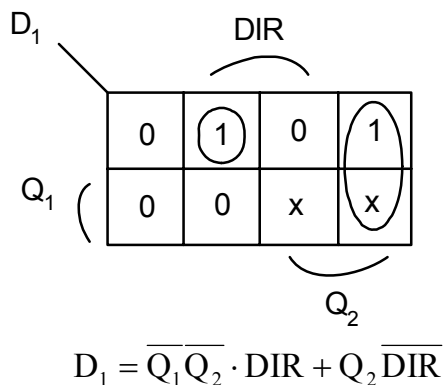
Χρησιμοποιώντας τον πίνακα ορισμού των καταστάσεων ως καταστάσεις εισόδου (PS) στον πίνακα καταστάσεων, και τις εισόδους D₁, D₂ των D-ffs ως επόμενες καταστάσεις (NS), δημιουργείται ο πίνακας καταστάσεων διέγερσης ως εξής:

Είσοδος DIR	PS			NS	καταστάσεις διέγερσης		Έξοδοι	
	Q1	Q2	D ₁		D ₂	A	B	
0	S ₀	0	0	S ₁	0	1	0	0
1	S ₀	0	0	S ₂	1	0	0	0
0	S ₁	0	1	S ₂	1	0	0	1
1	S ₁	0	1	S ₀	0	0	0	1
0	S ₂	1	0	S ₀	0	0	1	1
1	S ₂	1	0	S ₁	0	1	1	0

Στο παράδειγμα
οι έξοδοι
λήφθηκαν όπως οι
(PS)

Εξισώσεις διέγερσης

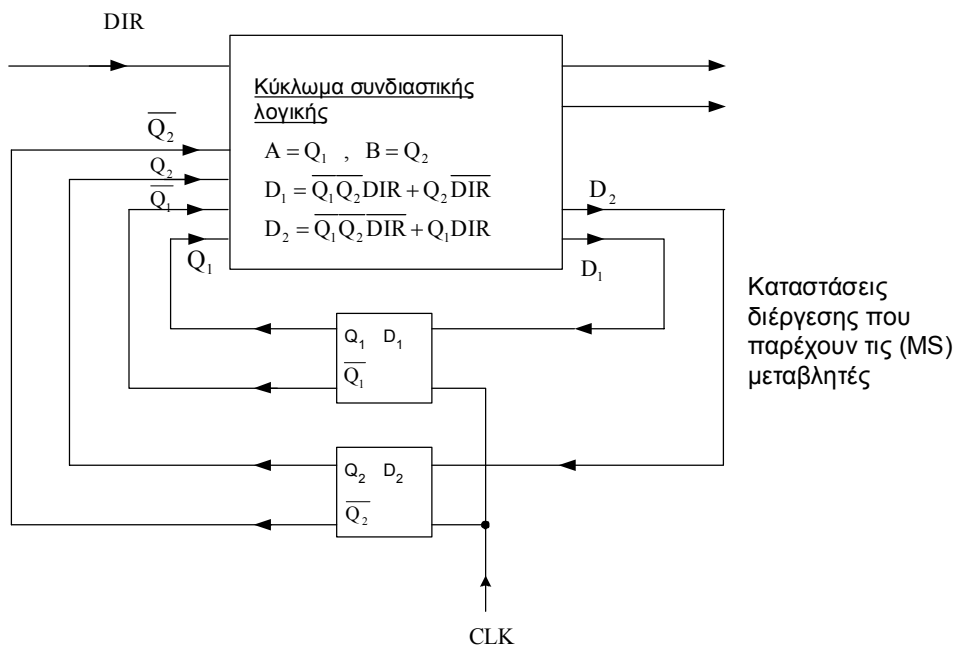
Χρησιμοποιώντας τους πίνακες-K, δημιουργείται μια εξίσωση διέγερσης για κάθε είσοδο των ffs σε σχέση με τις εισόδους του συστήματος και τις μεταβλητές κατάστασης (στη προκειμένη περίπτωση DIR, Q1, Q2)



Κατά τον ίδιο τρόπο προκύπτει

$$A = Q_1 \quad \text{και} \quad B = Q_2$$

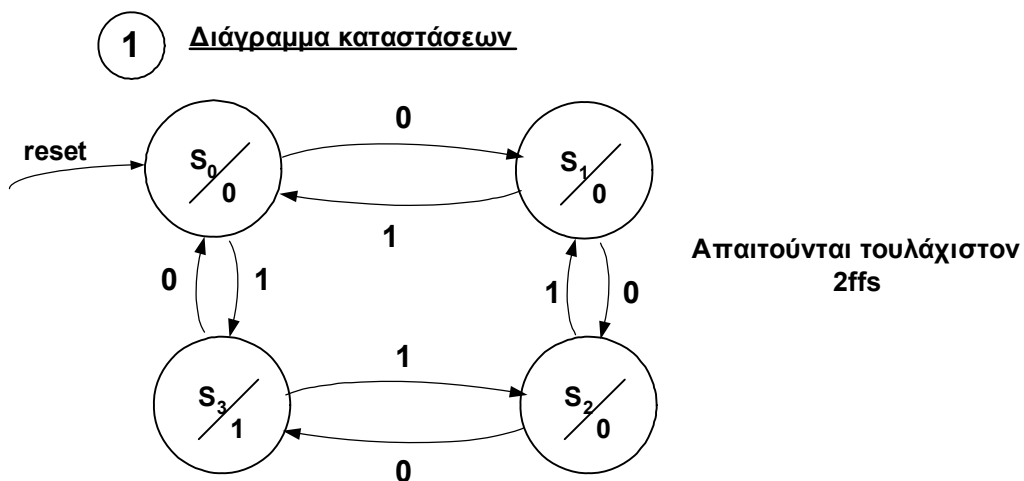
Το διάγραμμα του κυκλώματος είναι:



Παράδειγμα 2.5

Υλοποίηση ενός MOD-4 μετρητή με έλεγχο κατεύθυνσης (UP/DOWN) και έξοδο A.

Απάντηση



2 Πίνακας καταστάσεων

FS	NS		Έξοδος A
	DIR=0	DIR=1	
S ₀	S ₁	S ₃	0
S ₁	S ₂	S ₀	0
S ₂	S ₃	S ₁	0
S ₃	S ₀	S ₂	1

3 Ορισμός μεταβλητών κατάστασης

Κατάσταση	Q1	Q2
S ₀	0	0
S ₁	0	1
S ₂	1	0
S ₃	1	1

4 πίνακας διέγερσης

PS		DIR=0				DIR=1				Εξοδος Y				
		NS		EX		NS		EX						
Q ₁	Q ₂	Q ₁	Q ₂	J ₁	K ₁	J ₂	K ₂	Q ₁	Q ₂	J ₁	K ₁	J ₂	K ₂	
0	0	S ₁ :	0	1	0	X	1	X	S ₃ :	1	1	1	X	0
0	1	S ₂ :	1	0	1	X	X	1	S ₀ :	0	0	0	X	0
1	0	S ₃ :	1	1	X	0	1	X	S ₁ :	0	1	X	1	0
1	1	S ₀ :	0	0	X	1	X	1	S ₂ :	1	0	X	1	1

Οι παραπάνω στήλες
δημιουργήθηκαν χρησιμοποιώντας
Τον πίνακα για το Jk-ff

T

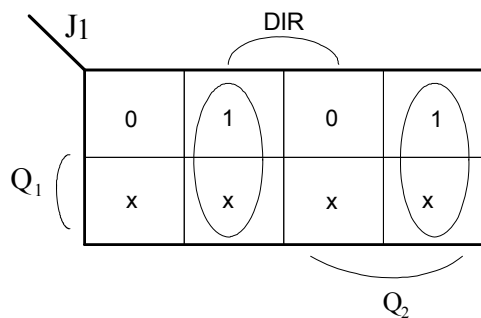
Q

K

$\overline{Q}$

$Q_n \rightarrow Q_{n+1}$	J	K
$0 \rightarrow 0$	0	x
$0 \rightarrow 1$	1	x
$1 \rightarrow 0$	x	1
$1 \rightarrow 1$	1	0

5 Εξισώσεις διέγερσης:

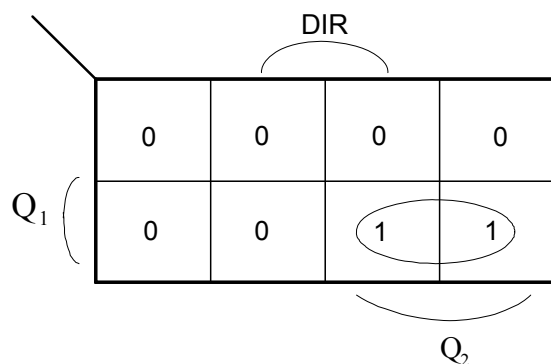


$$J_1 = \text{DIR} \cdot \overline{Q_2} + \overline{\text{DIR}} \cdot Q_2 = \text{DIR} \oplus Q_2$$

Ομοίως $K_1 = \text{DIR} \cdot \overline{Q_2} + \overline{\text{DIR}} \cdot Q_2 = \text{DIR} \oplus Q_2$

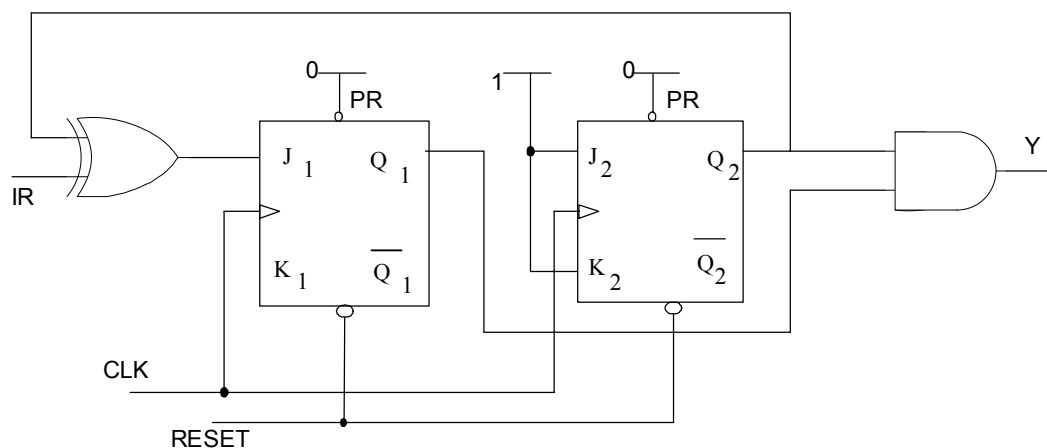
και $J_2 = K_2 = 1$

6 Εξισώσεις Εξόδου



$$Y = Q_1 Q_2$$

7 Διάγραμμα κυκλώματος



Πρόβλημα 2.1

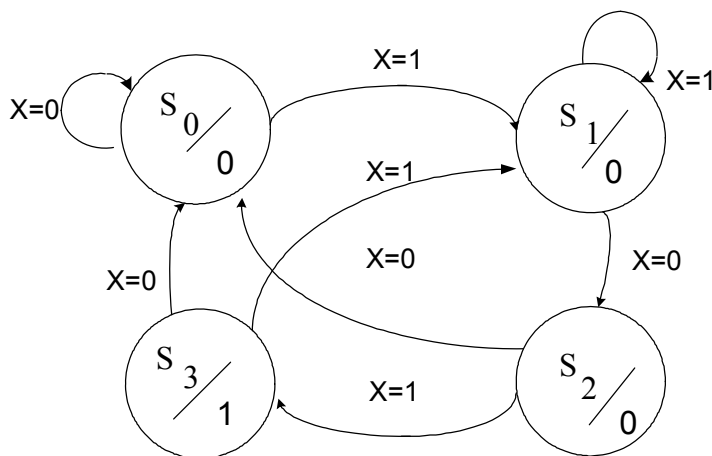
Να σχεδιαστεί ένα ακολουθιακό κύκλωμα που ανιχνεύει την ακολουθία 1 0 1 από ένα σύνολο δυαδικών στοιχείων που εισάγονται σειριακά με ρυθμό 1 bit ανά παλμό ρολογιού (Να χρησιμοποιηθούν JK-ffs)

Λύση:

Το κύκλωμα έχει μια μεταβλητή εισόδου X που παίρνει τις τιμές 0 και 1 και μια μεταβλητή εξόδου Z η οποία γίνεται 1 όταν ληφθεί μια ακολουθία 1 0 1. Κατ' αρχήν δεν είναι φανερό πόσες είναι οι δυνατές καταστάσεις, οι οποίες όμως μπορούν να προσδιοριστούν κάνοντας το διάγραμμα καταστάσεων.

Έστω S_0 η αρχική κατάσταση κατά την οποία κανένα από τα bit εισόδου δεν έχουν φτάσει στη σωστή ακολουθία. Στην κατάσταση S_0 , εάν $X = 0$ παραμένουμε στην κατάσταση S_0 , ενώ εάν $X = 1$ τότε έχουμε το πρώτο bit όπως ακριβώς το αναμένουμε στην επιθυμητή ακολουθία, συνεπώς γίνεται μετάβαση στην κατάσταση S_1 . Η έξοδος και στις δυο περιπτώσεις είναι μηδέν αφού δεν έχει ανιχνευθεί ακόμη η πλήρης ακολουθία.

Σκεπτόμενοι κατά τον ίδιο τρόπο εάν είμαστε στη κατάσταση S_1 και $X = 0$ μεταβαίνουμε στην κατάσταση S_2 , ενώ εάν $X = 1$ παραμένουμε στην S_1 θεωρώντας ότι το 1 είναι το πρώτο bit της αναμενόμενης ακολουθίας. Εάν είμαστε στην κατάσταση S_2 και $X = 1$ τότε μεταβαίνουμε στην κατάσταση S_3 όπου η έξοδος $Z = 1$, ενώ εάν $X = 0$ επιστρέφουμε στην κατάσταση S_0 . Τέλος από το S_3 για $X = 0$ επιστρέφουμε στην κατάσταση S_0 , ενώ για $X = 1$ στην S_1 . Σύμφωνα με τα παραπάνω σχεδιάζεται το παρακάτω διάγραμμα καταστάσεων.



Από το διάγραμμα καταστάσεων παρατηρούμε ότι έχουμε τέσσερις καταστάσεις, άρα απαιτούνται 2 μεταβλητές καταστάσεων, η δυο ffs.

Για τα σχεδιασμό του κυκλώματος γίνονται τα παρακάτω βήματα:

1. Πίνακας καταστάσεων

PS	NS		Εξοδος Z
	x = 0	x = 1	
S ₀	S ₀	S ₁	0
S ₁	S ₂	S ₁	0
S ₂	S ₀	S ₃	0
S ₃	S ₀	S ₁	1

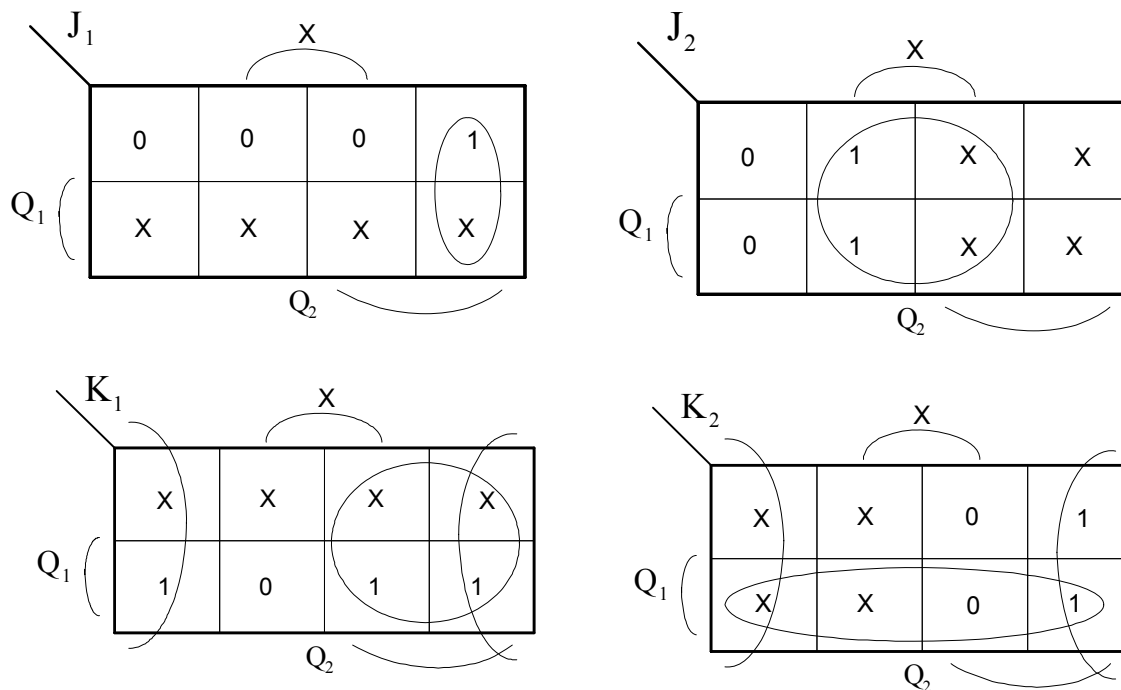
2. Ορισμός μεταβλητών καταστάσεων

	Q ₁	Q ₂
S ₀	0	0
S ₁	0	1
S ₂	1	0
S ₃	1	1

1. Πίνακας διέγερσης

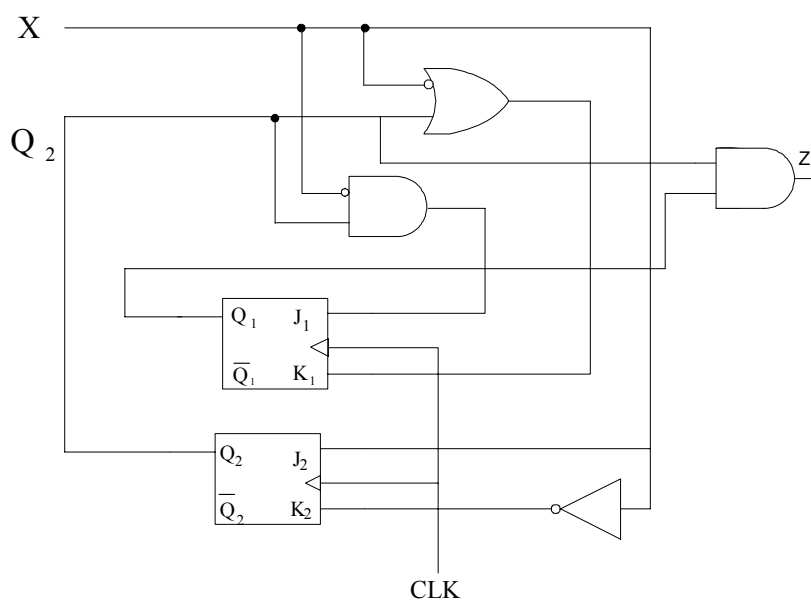
	PS		NS X=0 EX				NS X=0 EX				Z				
	Q ₁	Q ₂	Q ₁	Q ₂	J ₁	K ₁	J ₂	K ₂	Q ₁	Q ₂		J ₁	K ₁	J ₂	K ₂
S ₀	0	0	S ₀ : 0	0	0	X	0	X	S ₁ : 0	1	0	X	1	X	0
S ₁	0	1	S ₂ : 1	0	X	1	X	1	S ₁ : 0	1	0	X	1	0	0
S ₂	1	0	S ₀ : 0	0	X	1	0	X	S ₃ : 1	1	X	0	1	X	0
S ₃	1	1	S ₀ : 0	0	X	1	X	1	S ₄ : 0	1	X	1	X	0	1

4. Εξισώσεις διέγερσης



$$J_1 = \bar{X} \bar{Q}_2, \quad K_1 = \bar{X} + Q_2, \quad J_2 = X, \quad K_2 = QX$$

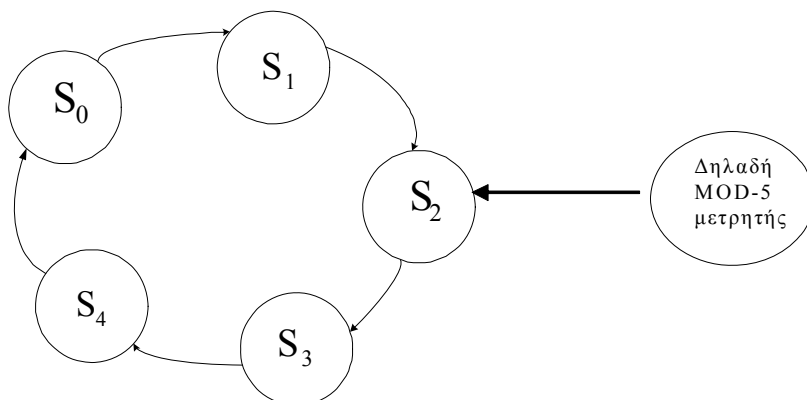
και $Z = Q_1 Q_2$



Προβλήματα από καταστάσεις που δεν χρησιμοποιούνται

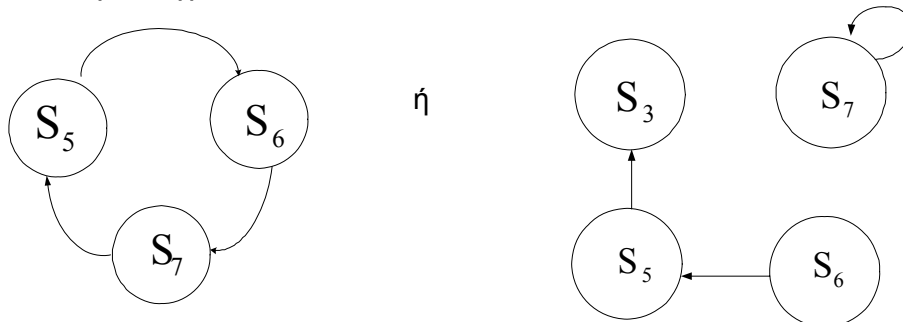
Ένα ακολουθιακό κύκλωμα μπορεί να ξεκινήσει από μια κατάσταση η οποία δεν χρησιμοποιείται, ή να εμφανίσει μια τέτοια κατάσταση λόγω θορύβου. Σ' αυτή τη περίπτωση η συμπεριφορά του κυκλώματος είναι μη προβλέψιμη.

Δηλαδή ένα σύστημα μπορεί να έχει το παρακάτω διάγραμμα καταστάσεων



Υπάρχουν τρεις καταστάσεις οι οποίες δεν χρησιμοποιούνται και οι οποίες μπορεί να εμφανίσουν διάφορες αλληλεπιδράσεις που εξαρτώνται από τον τρόπο υλοποίησης του κυκλώματος.

Για παράδειγμα:



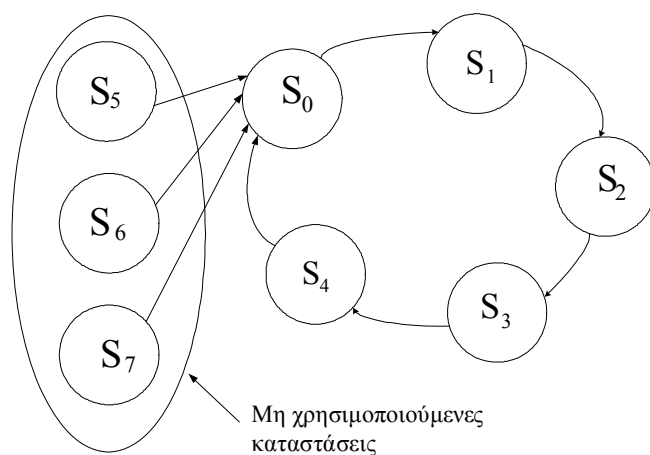
κλπ.

Λύση:

Περιλαμβάνονται στο διάγραμμα καταστάσεων και οι καταστάσεις που δεν χρησιμοποιούνται, σε μια λογική επαναφοράς στην αρχική κατάσταση (Reset circuitry). Δηλαδή κατασκευάζεται ένα διάγραμμα καταστάσεων το οποίο δεν επιτρέπει την εμφάνιση

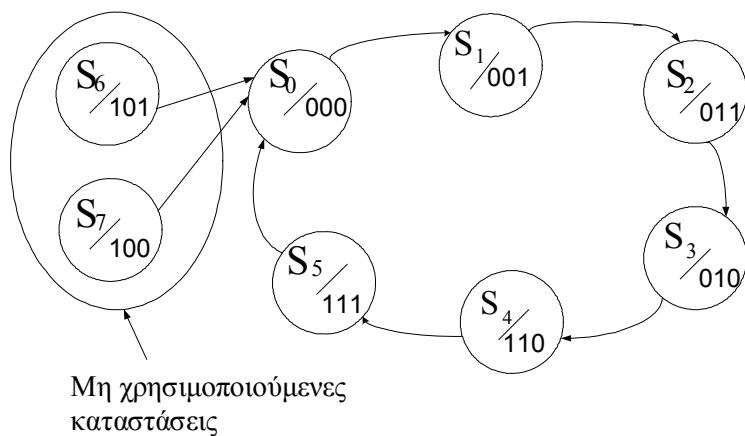
καταστάσεων 'παγίδα', ή εάν υπάρχουν, η εμφάνισή τους να έγινε πριν την έναρξη του ρολογιού.

Π.χ



Παράδειγμα 2.6

Να σχεδιαστεί ο μετρητής MOD-6 Gray Code χρησιμοποιώντας D-ffs με διάγραμμα καταστάσεων

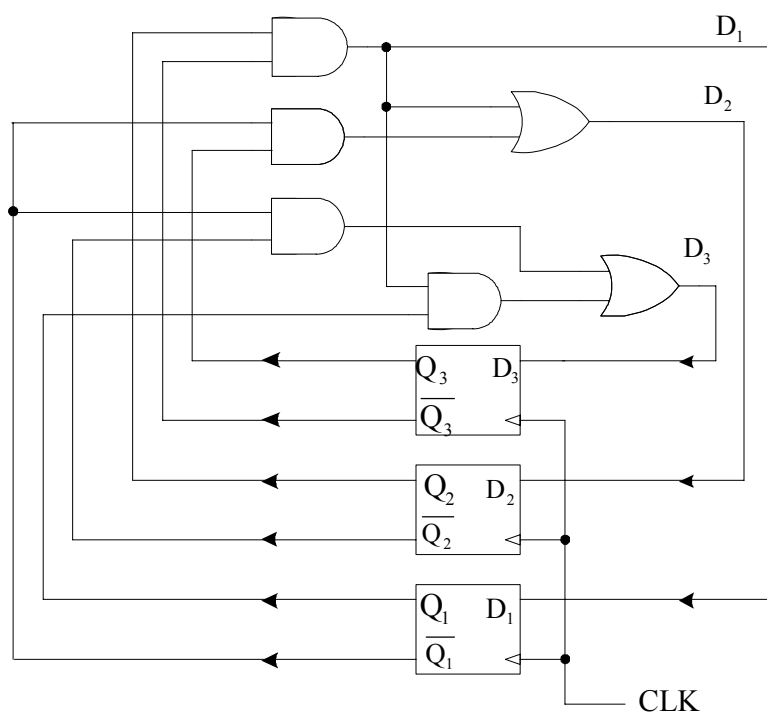


Θεωρούμε στο παράδειγμα μας ότι οι καταστάσεις PS είναι ίδιες με τις καταστάσεις εξόδου.

Απάντηση

PS+Εξοδοι $Q_1 \ Q_2 \ Q_3$	NS+Καταστάσεις διέγερσης $D_1 \ D_2 \ D_3$
$S_0 : 0 \ 0 \ 0$	$S_1 : 0 \ 0 \ 1$
$S_1 : 0 \ 0 \ 1$	$S_2 : 0 \ 1 \ 1$
$S_2 : 0 \ 1 \ 1$	$S_3 : 0 \ 1 \ 0$
$S_3 : 0 \ 1 \ 0$	$S_4 : 1 \ 1 \ 0$
$S_4 : 1 \ 1 \ 0$	$S_5 : 1 \ 1 \ 1$
$S_5 : 1 \ 1 \ 1$	$S_0 : 0 \ 0 \ 0$
$S_6 : 1 \ 0 \ 1$	$S_0 : 0 \ 0 \ 0$
$S_7 : 1 \ 0 \ 0$	$S_0 : 0 \ 0 \ 0$

Τελικό κύκλωμα



Ασκήσεις επανάληψης 2

Άσκηση 2.1

Να σχεδιαστεί το κύκλωμα που μετατρέπει ένα SR-ff σ' ένα JK-ff.

Λύση:

Πίνακες αληθείας των 2 ff's

J	K	Q_{n+1}	S	R	Q_{n+1}
0	0	Q_n	0	0	Q_n
0	1	0	0	1	0
1	0	1	1	0	1
1	1	$\overline{Q_n}$	1	1	-

Πίνακες συσχέτισης των εισόδων των 2 ffs

J	K	Q_n	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

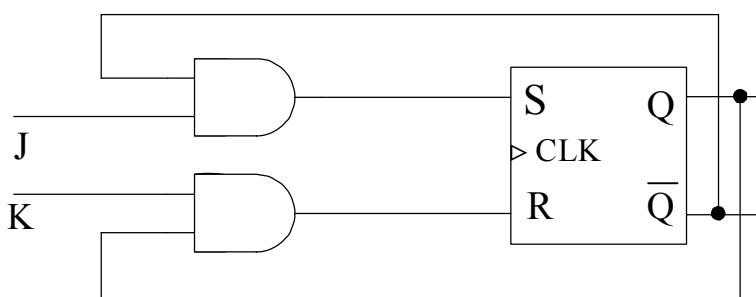
Εξαγωγή εκφράσεων των R,S σε συνάρτηση με τα J,k,

S	JK			
	00	01	11	10
0	0	0	1	1
1	X	0	0	X

R				
	00	01	11	10
0	X	X	0	0
1	0	1	1	0

$$S = J\bar{Q}_n$$

$$R = Q_n K$$

Τελικό κύκλωμα:**Άσκηση 2.2**

Επιθυμούμε να σχεδιάσουμε ένα ff νέου τύπου, το οποίο ονομάζουμε 'X-Y' flip-flop και του οποίου ο πίνακας αληθείας δίνεται ως εξής:

X	Y	Q_{n+1}
0	0	0
0	1	1
1	0	$\bar{Q}_n$
1	1	$\bar{Q}_n$

Να δείξετε ότι το X-Y ff μπορεί να σχεδιαστεί με βάση το S-R ff.

Λύση:

Ο πίνακας αληθείας του S-R ff είναι:

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	-

Πίνακας συσχέτισης των 2 ff's

X	Y	Q_n	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	1	X	0
1	0	0	1	1	0
1	0	1	0	0	1
1	1	0	1	1	0
1	1	1	0	0	1

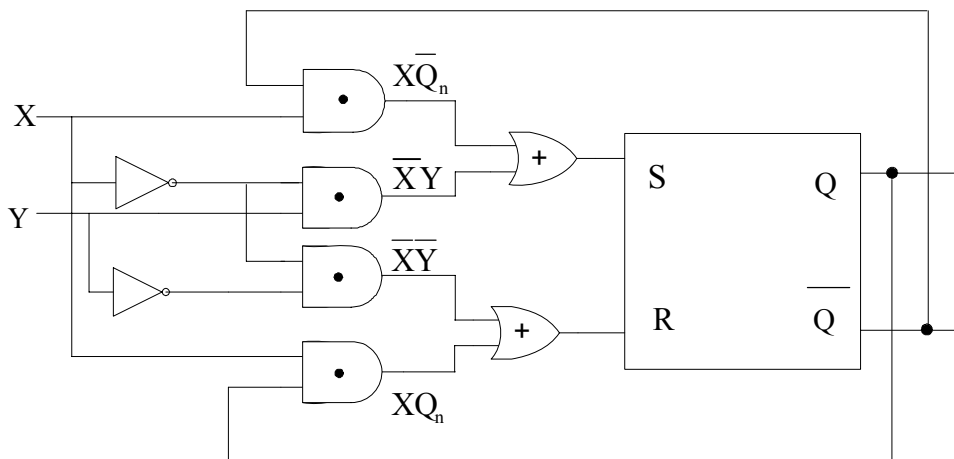
S, R συναρτήσει των X, Y, Q_n

		XY			
		00	01	11	10
Q_n	0	0	1	1	1
	1	0	X	0	0

$$S = \overline{X}Y + X\overline{Q}_n$$

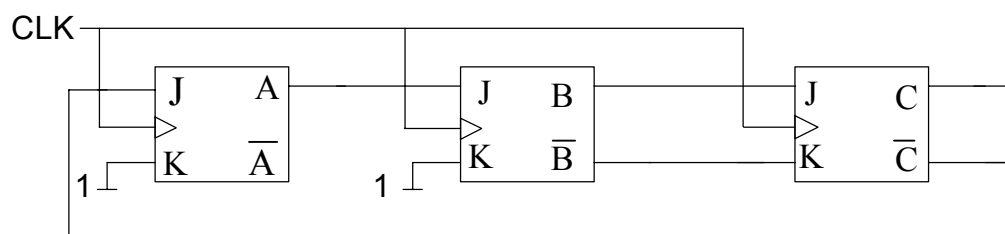
		XY			
		00	01	11	10
Q_n	0	X	0	0	0
	1	1	0	1	1

$$R = \overline{\overline{X}Y} + XQ_n$$

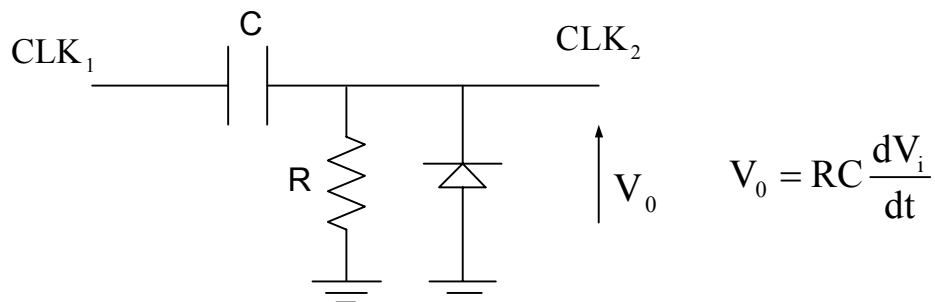
Τελικό κύκλωμα:**Άσκηση 2.3**

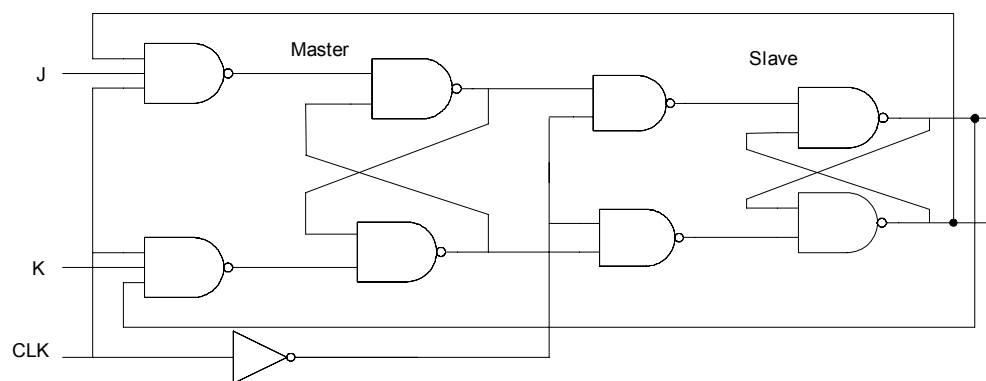
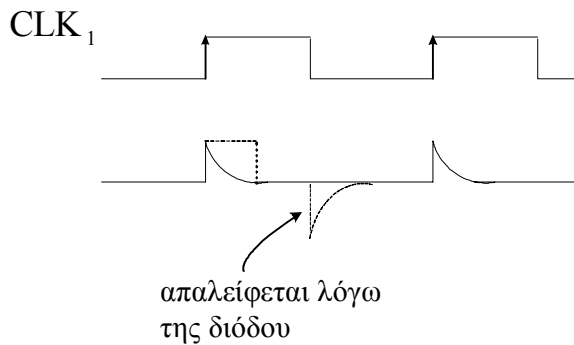
Για το μετρητή του σχήματος θεωρούμε ότι όλα τα ff είναι τύπου (Master-slave). Εάν όλα τα ff σκανδαλίζονται (trigger) κατά τη θετική έναυση του παλμού του ρολογιού και κατά την εκκίνηση είναι όλα σε κατάσταση 0, να:

- α) σχεδιαστεί ένα κύκλωμα (R,C, διόδου) που δίνει τις εναύσεις
- β) σχεδιαστεί το κύκλωμα του M-S του JK-ff
- γ) βρεθούν οι έξοδοι A, B, C του μετρητή για τους τέσσερις πρώτους παλμούς του ρολογιού.

**Λύση:**

α)





γ)

Παλμός	A	B	C	JA(= $\bar{C}$)	KA	JB(= A)	KB	JC(= B)	KC(= $\bar{B}$)
0	0	0	0	1	1	0	1	0	1
1	1	0	0	1	1	1	1	0	1
2	0	1	0	1	1	0	1	1	0
3	1	0	1	0	1	1	1	0	1
4	0	1	0						

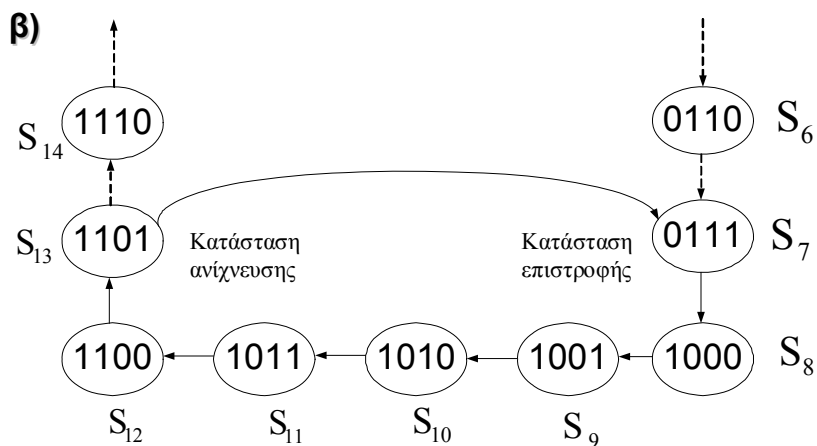
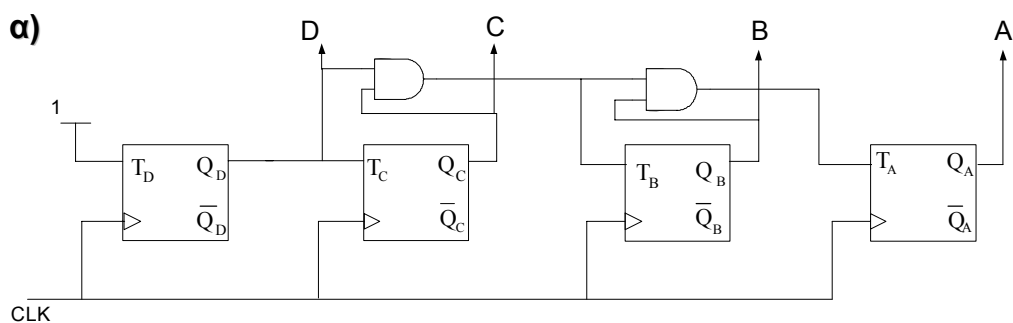
Άσκηση 2.4

Ένας σύγχρονος μετρητής 4 βαθμίδων έχει 16 καταστάσεις S_0 έως S_{15}), όπου $S_0 = 0\ 0\ 0\ 0$, $S_1 = 0\ 0\ 0\ 1$, $S_2 = 0\ 0\ 1\ 0$ κ.τ.λ.

α) Να σχεδιαστεί ο πλήρης σύγχρονος μετρητής που θα παράγει και τις 16 καταστάσεις διαδοχικά και αυξητικά (up-counter).

β) Να τροποποιηθεί το αρχικό κύκλωμα ώστε να παράγει μια περιορισμένη ακολουθία καταστάσεων μεταξύ της S_7 και S_{13}

Λύση:



γ)

ABCD

Η κατάσταση ανίχνευσης $X = S_{13} = 1\ 1\ 0\ 1 = A\bar{B}\bar{C}D$

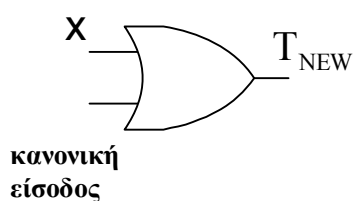
Κανονική διαδοχή $1101 \rightarrow 1110$

Επιθυμητή διαδοχή $1101 \rightarrow 0111$

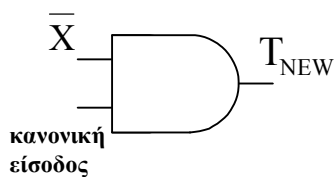
Όταν το $X = 1$ δηλαδή $A = 1$, $B = 1$, $C = 1$ και $D = 1$ τότε θα πρέπει:

το A , το οποίο σε κανονική διαδοχή δεν θα άλλαζε, στην προκειμένη περίπτωση θα πρέπει να αλλάξει και από 1 να γίνει 0. Συνεπώς το T_A θα πρέπει να κρατηθεί στο 1. Το B και C δεν απαιτούν αλλαγή διότι έχουν τις ίδιες τιμές που θα είχαν σε κανονική διαδοχή, ενώ το C , το οποίο υπό κανονική διαδοχή θα άλλαζε από 1 σε 0, ενώ εμείς επιθυμούμε να το κρατήσουμε στο 1, θα πρέπει να αναγκάσουμε το T_D να είναι 0.

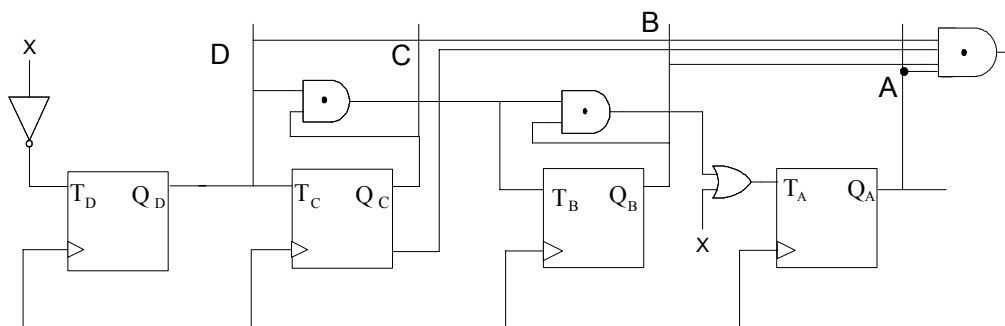
Για να έχουμε στην είσοδο του T το 1 χρησιμοποιούμε το κύκλωμα



και για T στο 0



Άρα το τροποποιημένο κύκλωμα είναι:



Άσκηση 2.5

Είναι το παράδειγμα 5 από τις σημειώσεις του μαθήματος 9.

Λύση:

α) Διάγραμμα όπως τις σημειώσεις

β) Πίνακας όπως τις σημειώσεις

γ) Εξισώσεις διέγερσης

	Q_2		
D_1	0	1	0
Q_1	0	1	0
			Q_3

	Q_2		
D_2	0	1	1
Q_1	0	1	0
			Q_3

	Q_2		
D_3	1	0	0
Q_1	0	1	0
			Q_3

$$D_2 = \overline{Q_1}Q_3 + Q_2\overline{Q_3} = \overline{Q_1}Q_3 + D_1$$

$$D_3 = \overline{Q_1}\overline{Q_2} + Q_1Q_2\overline{Q_3} = \overline{Q_1}\overline{Q_2} + Q_1D_1$$

$$D_1 = Q_2\overline{Q_3}$$

Το τελικό κύκλωμα είναι όπως σχεδιάστηκε στο παράδειγμα 2.6 των σημειώσεων.