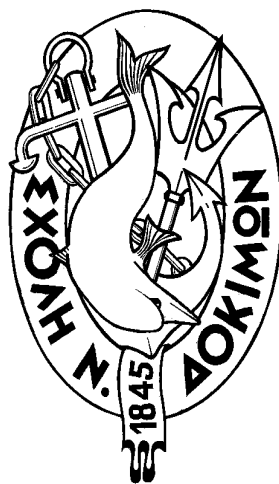


ΨΗΦΙΑΚΑ ΚΥΚΛΩΜΑΤΑ

ΑΝΔΡΕΑΣ Δ. ΤΣΙΓΚΟΠΟΥΛΟΣ
Δρ. ΗΛΕΚΤΡΟΛΟΓΟΣ ΜΗΧΑΝΙΚΟΣ
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ ΣΝΔ



ΣΧΟΛΗ ΝΑΥΤΙΚΩΝ ΔΟΚΙΜΩΝ
ΠΕΙΡΑΙΑΣ 2014

ΠΕΡΙΕΧΟΜΕΝΑ

ΨΗΦΙΑΚΑ ΚΥΚΛΩΜΑΤΑ.....	5
ΠΡΟΛΟΓΟΣ.....	5
1. ΕΙΣΑΓΩΓΗ.....	7
2. ΑΡΙΘΜΗΤΙΚΑ ΣΥΣΤΗΜΑΤΑ - ΚΩΔΙΚΕΣ.....	9
2.1. Εισαγωγή στα διάφορα συστήματα αρίθμησης.....	9
2.2. Μετατροπή αριθμού από οποιοδήποτε σύστημα στο δεκαδικό.....	10
2.3. Μετατροπή δεκαδικού σε οποιοδήποτε σύστημα.....	10
2.4. Μετατροπή αριθμού από ένα σύστημα σε άλλο.....	11
2.4.1. Μετατροπή οκταδικού αριθμού σε δυαδικό και αντίστροφα.....	11
2.4.2. Μετατροπή δεκαεξαδικού αριθμού σε δυαδικό και αντίστροφα.....	12
2.5. Αριθμητικές πράξεις στο δυαδικό σύστημα.....	13
2.6. Προσημασμένοι αριθμοί. Συμπληρώματα αριθμών.....	13
2.7. Αφαίρεση δύο αριθμών με τη βοήθεια συμπληρώματος.....	16
2.8. Κώδικες.....	18
2.8.1. Δυαδικοί Κώδικες.....	18
2.8.2. Δυαδικοί κώδικες με βάρη.....	19
2.8.2.1. Ο κώδικας BCD.....	19
2.8.2.2. Μετατροπή από BCD σε δεκαδικό.....	20
2.8.2.3. Μετατροπή από δεκαδικό σε BCD.....	20
2.8.2.4. Αριθμοί του κώδικα BCD και δυαδικοί αριθμοί.....	20
2.8.3. Δυαδικοί κώδικες χωρίς βάρη.....	21
2.8.3.1. Ορισμοί.....	21
2.8.3.2. Ο Κώδικας GRAY.....	22
2.8.4. Αλφαριθμητικοί κώδικες.....	23
2.8.4.1. Ορισμοί.....	23
2.8.4.2. Ο Κώδικας ASCII.....	24
3. ΑΛΓΕΒΡΑ BOOLE.....	27
3.1. Θεωρήματα και Αξιώματα της Άλγεβρας Boole.....	27
3.2. Λογικές συναρτήσεις.....	30
3.2.1. Απλοποίηση λογικών συναρτήσεων.....	30
3.2.2. Λογικές συναρτήσεις ως άθροισμα γινομένων (SOP : Sum Of Products).....	32
3.2.3. Λογικές συναρτήσεις ως γινόμενο αθροισμάτων (POS : Product Of Sums).....	34
3.3. Ψηφιακές Λογικές Πύλες.....	36
3.4. Υλοποίηση Συναρτήσεων με πύλες NAND.....	40
3.5. Υλοποίηση Συναρτήσεων με πύλες NOR.....	41
4. ΑΠΛΟΠΟΙΗΣΗ ΛΟΓΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΜΕ ΤΗ ΜΕΘΟΔΟ KARNAUGH.....	45
4.1. Χάραξη του χάρτη Karnaugh για δύο μεταβλητές.....	45
4.2. Χάραξη του χάρτη Karnaugh για τρεις μεταβλητές.....	46
4.3. Χάραξη του χάρτη Karnaugh για τέσσερις μεταβλητές.....	46
4.4. Χάραξη του χάρτη Karnaugh για πέντε μεταβλητές.....	47
4.5. Απλοποίηση με τη μέθοδο Karnaugh.....	47
4.6. Απλοποίηση συνάρτησης δύο μεταβλητών.....	48
4.7. Απλοποίηση συνάρτησης τριών μεταβλητών.....	49
4.8. Απλοποίηση συνάρτησης τεσσάρων μεταβλητών.....	50
4.9. Αδιάφοροι όροι (Don't care).....	51
4.10. Παραδείγματα.....	52
5. ΣΥΝΔΥΑΣΤΙΚΑ ΚΥΚΛΩΜΑΤΑ.....	57
5.1. Αθροιστές (Adders).....	58
5.1.1. Ημιαθροιστής (Half-Adder).....	58
5.1.2. Πλήρης Αθροιστής (Full-Adder).....	59
5.2. Αφαιρέτες (Subtractors).....	63
5.2.1. Ημιαφαιρέτης (Half-Subtractor).....	63

5.2.2. Πλήρης Αφαιρέτης (Full Subtractor).....	64
5.3. Αποκωδικοποιητές (Decoders).....	65
5.4. Κωδικοποιητές (Coders).....	69
5.5. Πολυπλέκτες (Multiplexers).....	70
5.6. Αποπολυπλέκτες (Demultiplexers).....	73
5.7. Μνήμη ROM.....	75
6. ΑΚΟΛΟΥΘΙΑΚΑ ΚΥΚΛΩΜΑΤΑ.....	87
6.1. Flip-Flops.....	88
6.1.1. Flip-Flop S-R.....	88
6.1.2. Flip-Flop S-R με ρολοί (Clock).....	90
6.1.3. Flip-Flop J-K.....	91
6.1.4. Flip-Flop D.....	92
6.1.5. Flip-Flop T.....	94
6.1.6. Πυροδότηση των Flip-Flops.....	94
6.1.7. Master - Slave Flip-Flop.....	95
6.1.8. Ακμοπυροδοτούμενα Flip-Flop.....	97
6.1.9. Ακμοπυροδοτούμενα Flip-Flop.....	100
6.2. Καταχωρητές (Registers).....	104
6.2.1. Καταχωρητές με παράλληλη φόρτωση.....	104
6.2.2. Καταχωρητές με σειριακή φόρτωση ή Καταχωρητές ολίσθησης.....	107
6.3. Η μνήμη τυχαίας προσπέλασης (RAM).....	112
6.4. Απαριθμητές (Counters).....	114
6.4.1. Ασύγχρονοι απαριθμητές.....	114
6.4.2. Ασύγχρονος δεκαδικός απαριθμητής.....	116
6.4.3. Σύγχρονοι απαριθμητές.....	125
7. ΜΕΘΟΔΟΙ ΣΧΕΔΙΑΣΗΣ ΣΥΓΧΡΟΝΩΝ ΑΚΟΛΟΥΘΙΑΚΩΝ ΚΥΚΛΩΜΑΤΩΝ.....	133
7.1. Εισαγωγή.....	133
7.2. Διάγραμμα καταστάσεων (state diagram).....	133
7.2.1. Πίνακας καταστάσεων (state table).....	134
7.2.2. Εξισώσεις καταστάσεων (state equations).....	134
7.3. Παραδείγματα Σχεδίασης Σύγχρονων Ακολουθιακών Κυκλωμάτων.....	136
Παράδειγμα 7.1.....	136
Παράδειγμα 7.2.....	138
Παράδειγμα 7.3.....	140
Παράδειγμα 7.4.....	145
8. ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ.....	151

ΨΗΦΙΑΚΑ ΚΥΚΛΩΜΑΤΑ

ΠΡΟΛΟΓΟΣ

Τα ψηφιακά ηλεκτρονικά κυκλώματα είναι απαραίτητα για την κατανόηση του σχεδιασμού και του τρόπου λειτουργίας ενός ολοένα αυξανόμενου εύρους εφαρμογών, από ηλεκτρονικές διατάξεις εμπορικής και βιομηχανικής χρήσης μέχρι τις σύγχρονες συσκευές επικοινωνίας και από υπολογιστές και ολοκληρωμένα πληροφοριακά συστήματα μέχρι στρατιωτικά συστήματα και εφαρμογές ασφαλείας. Όσο οι διατάξεις αυτές σμικρύνονται σε μέγεθος και χρησιμοποιούν πιο περίπλοκες τεχνολογίες, είναι απαραίτητο να γίνουν κατανοητές από τους Ν. Δοκίμους, οι οποίοι θα τις χρησιμοποιούν σε επιχειρησιακή βάση, οι βασικές αρχές λειτουργίας, σχεδιασμού και υλοποίησης των ψηφιακών κυκλωμάτων.

Η ανάπτυξη των ψηφιακών ηλεκτρονικών, που πραγματοποιήθηκε τα τελευταία 30 χρόνια, αποτελεί γεγονός, για το οποίο δεν υπάρχει κάτι ανάλογο σε οποιοδήποτε άλλο κλάδο της επιστήμης του μηχανικού. Με βάση τα παραπάνω, τούτο το σύγγραμμα έχει ως σκοπό να παρέχει στους Ν. Δοκίμους τις αναγκαίες θεμελιώδεις έννοιες της ψηφιακής λογικής και ταυτόχρονα να τους εξοικειώσει με μεθόδους σχεδιασμού και τεχνικές στο επίπεδο του συστήματος. Επίσης, περιλαμβάνονται λυμένα παραδείγματα, ώστε να αναδεικνύεται η πρακτική χρησιμότητα των διαφόρων εννοιών που εισάγονται στη θεωρία, αλλά και ο σωστός τρόπος εφαρμογής τους. Η λίστα βιβλιογραφικών αναφορών περιέχει όλες εκείνες τις πηγές που χρησιμοποιήθηκαν και εμπεριέχονται στο παρόν σύγγραμμα υπό τη μορφή περικοπών, αυτούσιων ή επεξεργασμένων τμημάτων κειμένων και σχημάτων.

1. ΕΙΣΑΓΩΓΗ

Τα ψηφιακά κυκλώματα χρησιμοποιούνται στη σχεδίαση συστημάτων όπως: ψηφιακών υπολογιστών, συστημάτων ελέγχου, μετάδοσης δεδομένων και πολλών άλλων εφαρμογών που απαιτούν ψηφιακό ηλεκτρονικό υλικό. Οι ψηφιακοί υπολογιστές ειδικότερα, με τη δυνατότητα ταχύτατων υπολογισμών και επεξεργασίας δεδομένων που διαθέτουν, έκαναν δυνατά πολλά επιστημονικά, βιομηχανικά και εμπορικά επιτεύγματα τα οποία θα ήταν ανέφικτα χωρίς αυτούς. Το πιο σημαντικό χαρακτηριστικό ενός ψηφιακού υπολογιστή είναι η δυνατότητα επεξεργασίας *διακριτών στοιχείων* πληροφορίας. Τέτοια διακριτά στοιχεία μπορεί να είναι λ.χ. οι ηλεκτρικοί παλμοί που μεταβιβάζονται διαμέσου των ψηφιακών κυκλωμάτων του υπολογιστή.

Τα διακριτά στοιχεία πληροφορίας παριστάνονται σε ένα ψηφιακό σύστημα από φυσικές ποσότητες που λέγονται *σήματα*. Τα πιο συνηθισμένα σήματα είναι τα ηλεκτρικά, όπως οι τάσεις και οι εντάσεις. Τα σήματα σε όλα τα σημερινά ψηφιακά ηλεκτρονικά συστήματα έχουν δύο μόνο διακριτές τιμές, και γι' αυτό τα λέμε *δυναδικά* (binary). Οι σχεδιαστές ψηφιακών συστημάτων περιορίζονται στη χρήση δυναδικών μόνο σημάτων εξαιτίας της χαμηλής αξιοπιστίας των ηλεκτρονικών κυκλωμάτων, όπου τα σήματα μπορούν να παίρνουν περισσότερες από δύο καταστάσεις. Ο διακριτός χαρακτήρας της πληροφορίας προέρχεται είτε απ' αυτή καθαυτή τη φύση μιας κάποιας διαδικασίας ή από κάποιας μορφής “κβαντισμό” (quantization) που τον κάνουμε επίτηδες σε μια διαδικασία με συνεχή χαρακτήρα καταγράφοντας συγκεκριμένες μόνο τιμές και κωδικοποιώντας τις σε δυναδική μορφή.

Η δυναδική λογική ασχολείται με μεταβλητές που μπορούν να έχουν δύο μόνο διακριτές τιμές και συνδέονται μεταξύ τους με “λογικές” (δυναδικές) πράξεις. Οι δύο τιμές που μπορούν να παίρνουν οι μεταβλητές μπορούν να έχουν διάφορα ονόματα (π.χ. αληθής και ψευδής, ή ναι και όχι, κλπ.), αλλά για τους δικούς μας σκοπούς είναι βολικό να τις σκεφτόμαστε ως bits και να δίνουμε τις τιμές 1 και 0. Η δυναδική λογική χρησιμοποιείται για να περιγράψει με μαθηματικό τρόπο την επεξεργασία των δυναδικών πληροφοριών, κι έτσι ταιριάζει εξαιρετικά για την ανάλυση και το σχεδιασμό ψηφιακών συστημάτων. Η δυναδική λογική την οποία θα εξετάσουμε ευθύς αμέσως περιγράφεται με τη βοήθεια της μαθηματικής μεθοδολογίας της “Άλγεβρας Boole”. Η πρακτική εφαρμογή της άλγεβρας Boole έγινε για πρώτη φορά από τον C.E. Shannon το 1938 με τη βοήθεια ηλεκτρικών κυκλωμάτων διακοπών δύο καταστάσεων.

2. ΑΡΙΘΜΗΤΙΚΑ ΣΥΣΤΗΜΑΤΑ - ΚΩΔΙΚΕΣ

2.1. Εισαγωγή στα διάφορα συστήματα αρίθμησης

Το αριθμητικό σύστημα που χρησιμοποιούμε καθημερινά είναι το **δεκαδικό σύστημα**. Δεν είναι όμως το μοναδικό. Είναι ένα από τα άπειρα δυνατά αριθμητικά συστήματα, που μπορούμε να σχηματίσουμε με βάση τον οποιονδήποτε αριθμό, λ.χ. το επταδικό με βάση το 7, το οκταδικό με βάση το 8, το δεκαεξαδικό με βάση το 16, κ.ο.κ.

Το απλούστερο σύστημα αρίθμησης είναι το **δυναδικό σύστημα** το οποίο χρησιμοποιείται στους ηλεκτρονικούς υπολογιστές και σε όλα τα ψηφιακά συστήματα.

Ένας οποιοσδήποτε αριθμός N μπορεί να εκφραστεί σ' ένα αριθμητικό σύστημα με βάση r ως εξής:

$$(N)_r = a_n r^n + a_{n-1} r^{n-1} + \dots + a_1 r^1 + a_0 r^0 \quad \text{εάν } N > 1$$

$$(N)_r = a_{-1} r^{-1} + a_{-2} r^{-2} + \dots + a_{-n} r^{-n} \quad \text{εάν } 0 < N < 1$$

όπου r : η βάση του συστήματος.

n : 1,2,3,... ανάλογα με την περίπτωση.

a : ακέραιος αριθμός από 0 έως $r-1$.

Τα παραπάνω γίνονται εύκολα κατανοητά με τα ακόλουθα παραδείγματα:

$$(956)_{10} = 9 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$$

$$(123,456)_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} + 6 \times 10^{-3}$$

$$(10111)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (23)_{10}$$

$$(110,101)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = (6,625)_{10}$$

$$(765)_8 = 7 \times 8^2 + 6 \times 8^1 + 5 \times 8^0 = (501)_{10}$$

$$(A14,B6)_{16} = 10 \times 16^2 + 1 \times 16^1 + 4 \times 16^0 + 11 \times 16^{-1} + 6 \times 16^{-2} = (2580,7)_{10}$$

Μερικά ενδιαφέροντα χαρακτηριστικά των συστημάτων αρίθμησης είναι τα ακόλουθα:

- 1) Ο αριθμός των ψηφίων που χρησιμοποιούνται σ' ένα σύστημα ισούται με την βάση του. Μάλιστα το μεγαλύτερο ψηφίο είναι: βάση-1.

Π.χ. Στο δυναδικό σύστημα τα ψηφία είναι 2. Το 0 και το 1. Επομένως κάθε αριθμός στο δυναδικό σύστημα θα γραφτεί με συνδυασμό των 0 και 1.

Τα ψηφία των σπουδαιότερων αριθμητικών συστημάτων

Δυναδικό	0	1														
Οκταδικό	0	1	2	3	4	5	6	7								
Δεκαδικό	0	1	2	3	4	5	6	7	8	9						
Δεκαεξαδικό	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
											(10)	(11)	(12)	(13)	(14)	(15)

- 2) Στο άθροισμα που εκφράζεται ένας αριθμός, κάθε ψηφίο του αριθμού πολλαπλασιάζεται με την βάση στην κατάλληλη δύναμη που εξαρτάται από τη θέση του ψηφίου.

Π.χ. $(1\ 1\ 0)_2$

$$1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

- 3) Στο άθροισμα που παριστάνει έναν αριθμό, το μέρος με τις αρνητικές δυνάμεις της βάσης εκφράζει το κλασματικό μέρος του αριθμού.

Π.χ. $(1230,41)_5$

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

ακέραιο μέρος

κλασματικό μέρος

2.2. Μετατροπή αριθμού από οποιοδήποτε σύστημα στο δεκαδικό

Σύμφωνα με όσα ειπώθηκαν στην προηγούμενη παράγραφο, μπορούμε εκφράζοντας έναν αριθμό σε οποιοδήποτε σύστημα σαν ανάπτυγμα δυνάμεων της βάσης του συστήματος, να τον μετατρέψουμε στο δεκαδικό σύστημα.

Π.χ.

$$(1011,1101)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\ = 8 + 0 + 2 + 1 + \frac{1}{2} + \frac{1}{4} + 0 + \frac{1}{16} = (11,8125)_{10}$$

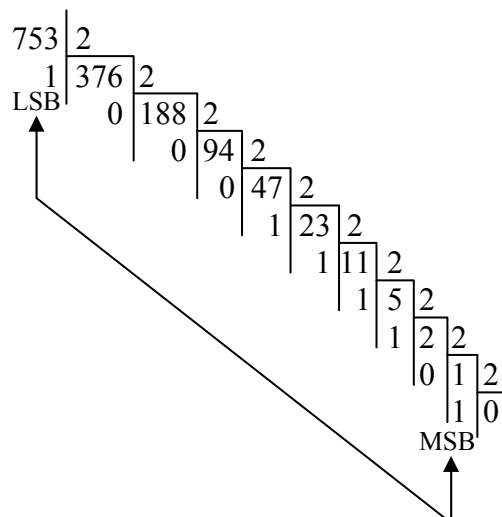
$$(351,72)_8 = 3 \times 8^2 + 5 \times 8^1 + 1 \times 8^0 + 7 \times 8^{-1} + 2 \times 8^{-2} = (233,90625)_{10}$$

$$(3B7C,1A)_{16} = 3 \times 16^3 + 11 \times 16^2 + 7 \times 16^1 + 12 \times 16^0 + 1 \times 16^{-1} + 10 \times 16^{-2} = (15228,1015625)_{10}$$

2.3. Μετατροπή δεκαδικού σε οποιοδήποτε σύστημα

Μια συνηθισμένη μέθοδος είναι να εκτελούμε διαδοχικές διαρέσεις του ακέραιου μέρους με τη βάση του συστήματος στο οποίο θέλουμε να κάνουμε τη μετατροπή. Δηλαδή, κάθε πηλίκου που προκύπτει, διαιρείται ξανά για να πάρουμε ένα νέο πηλίκου, και η διαδικασία συνεχίζεται μέχρι να πάρουμε μηδενικό πηλίκου. Τα υπόλοιπα σε κάθε στάδιο είναι τα ψηφία που σχηματίζουν τον αριθμό στο νέο σύστημα, με σημαντικότερο ψηφίο (MSB) το τελευταίο υπόλοιπο.

Παράδειγμα: $(753)_{10} = (1011110001)_2$



$$(165)_{10} = (A5)_{16}$$

$$\begin{array}{r|l} 165 & 16 \\ \hline 5 & 10 \\ \hline \text{LSB} & 10 \\ \hline & 0 \\ \hline & \text{MSB} \end{array}$$

$$(320)_{10} = (500)_8$$

$$\begin{array}{r|l} 320 & 8 \\ \hline 0 & 40 \\ \hline \text{LSB} & 0 \\ \hline & 5 \\ \hline & 5 \\ \hline & 0 \\ \hline & \text{MSB} \end{array}$$

Για να μετατρέψουμε ένα κλασματικό αριθμό από το δεκαδικό σύστημα σε ένα άλλο σύστημα, πραγματοποιούμε διαδοχικούς πολλαπλασιασμούς με την βάση του συστήματος. Εδώ, αντί για υπόλοιπα, έχουμε πλεονάσματα. Αυτά αποτελούν τα ψηφία του αριθμού στο νέο σύστημα. Η διαδικασία φαίνεται καθαρά στο παρακάτω παράδειγμα, όπου θέλουμε να μετατρέψουμε τον αριθμό $(0,8125)_{10}$ στο δυαδικό σύστημα.

$$\begin{array}{l} 0,8125 \times 2 = 1,6250 = 0,6250 \text{ και πλεόνασμα } 1 \\ 0,6250 \times 2 = 1,2500 = 0,2500 \text{ και πλεόνασμα } 1 \\ 0,2500 \times 2 = 0,5000 = 0,5000 \text{ και πλεόνασμα } 0 \\ 0,5000 \times 2 = 1,0000 = 0,0000 \text{ και πλεόνασμα } 1 \end{array} \begin{array}{l} \left| \begin{array}{l} \leftarrow \text{MSB} \\ \downarrow \\ \leftarrow \text{LSB} \end{array} \right. \end{array}$$

Άρα $(0,8125)_{10} = (0,1101)_2$

$$\text{Έτσι λοιπόν } (753,8125)_{10} = (1011110001,1101)_2$$

Από το παραπάνω παράδειγμα παρατηρούμε:

(α) Όταν κάποιο γινόμενο υπερβεί το 1, τότε στην επόμενη γραμμή γράφουμε το κλασματικό του μέρος, ενώ το ακέραιο μέρος 1 το κρατάμε ως πλεόνασμα και αποτελεί ψηφίο του αριθμού του νέου συστήματος.

(β) Οι διαδοχικοί πολλαπλασιασμοί συνεχίζονται είτε μέχρι να καλυφθεί το ίδιο μήκος λέξης, είτε μέχρι να πετύχουμε την ίδια ακρίβεια δεκαδικού και δυαδικού.

(γ) Αν το πλήθος των bits των αριθμών είναι n_2 και n_{10} αντίστοιχα, τότε για να έχουμε την ίδια ακρίβεια πρέπει:

$$2^{n_2} \cong 10^{n_{10}} \Rightarrow n_2 = \frac{n_{10}}{\log_{10} 2} \Rightarrow n_2 = 3,3 \times n_{10}$$

Έτσι, στο προηγούμενο παράδειγμα, που ήταν $n_{10} = 4 \Rightarrow n_2 = 13$ για να έχουν οι αριθμοί την ίδια ακρίβεια.

2.4. Μετατροπή αριθμού από ένα σύστημα σε άλλο

Γενικά ακολουθείται η πορεία:

$$(\text{αριθμός στο } \Sigma_1) \rightarrow (\text{αριθμός στο δεκαδικό}) \rightarrow (\text{αριθμός στο } \Sigma_2)$$

Αν όμως το Σ_1 είναι το οκταδικό ή το δεκαεξαδικό και Σ_2 το δυαδικό η διαδικασία είναι απλούστερη.

2.4.1. Μετατροπή οκταδικού αριθμού σε δυαδικό και αντίστροφα

Μετατρέπουμε κάθε οκταδικό ψηφίο στον αντίστοιχο του τριψήφιο δυαδικό αριθμό και βάζουμε τον πλήρη τριψήφιο δυαδικό αριθμό στην αντίστοιχη θέση του οκταδικού ψηφίου (διότι $2^3 = 8$).

Π.χ.

$$\begin{array}{ccc} & 3 & 6 & 2 \\ & \downarrow & \downarrow & \downarrow \\ (362)_8 & = & 011 & 110 & 010 \end{array}$$

Άρα $(362)_8 = (011110010)_2$

Η μετατροπή δυαδικού σε οκταδικό είναι ακριβώς η αντίστροφη διαδικασία.

Π.χ.

$$\begin{array}{cccc} (1010,01001)_2 & = & 001 & 010 & 010 & 010 \\ & & \downarrow & \downarrow & \downarrow & \downarrow \\ & & 1 & 2 & 2 & 2 \end{array}$$

Άρα $(1010,01001)_2 = (12,22)_8$

Δηλαδή χωρίζουμε τα δυαδικά ψηφία σε τριάδες αρχίζοντας από την υποδιαστολή, προς τα αριστερά για τα ακέραια ψηφία και προς τα δεξιά για τα κλασματικά. Αν τα τελευταία δεξιά και αριστερά ψηφία δεν συμπληρώνουν τριάδες, προσθέτουμε όσα 0 χρειάζονται για να συμπληρωθούν. Στο τέλος γίνεται η μετατροπή κάθε τριάδας στο αντίστοιχο οκταδικό ψηφίο.

2.4.2. Μετατροπή δεκαεξαδικού αριθμού σε δυαδικό και αντίστροφα

Εδώ κάθε δεκαεξαδικό ψηφίο έχει το αντίστοιχό του σε μία δυαδική τετράδα (διότι $2^4 = 16$). Οι διαδικασίες είναι οι ίδιες όπως και στην περίπτωση των οκταδικών αριθμών.

Π.χ.

$$\begin{array}{ccccc} & A & 2 & C & D & 1 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ (A2C,D1)_{16} & = & 1010 & 0010 & 1100 & 1101 & 0001 \end{array}$$

Άρα $(A2C,D1)_{16} = (101000101100,11010001)_2$

Π.χ.

$$\begin{array}{cccc} (1011101,11101)_2 & = & 0101 & 1101 & 1110 & 1000 \\ & & \downarrow & \downarrow & \downarrow & \downarrow \\ & & 5 & D & E & 8 \end{array}$$

Άρα $(1011101,11101)_2 = (5D,E8)_{16}$

Ο χωρισμός και η συμπλήρωση με μηδενικά γίνεται ανάλογα με την περίπτωση μετατροπής δυαδικού σε οκταδικό.

Πίνακας των 16 πρώτων αριθμών στα διάφορα συστήματα αρίθμησης			
Δεκαδικός	Δυαδικός	Οκταδικός	Δεκαεξαδικός
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

2.5. Αριθμητικές πράξεις στο δυαδικό σύστημα

Πρόσθεση: Εκτελείται κατά τον ίδιο τρόπο όπως και η δεκαδική, ισχύουν δε οι ακόλουθοι νόμοι:

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	0 και 1 κρατούμενο

$$\begin{array}{r} \text{Π.χ.} \quad 110,01 \\ + \quad 111,01 \\ \hline 1101,10 \end{array}$$

Αφαίρεση: Εκτελείται κατά τον ίδιο τρόπο όπως και η δεκαδική, ισχύουν δε οι ακόλουθοι νόμοι:

A	B	A - B
0	0	0
0	1	1 και 1 κρατούμενο
1	0	1
1	1	0

$$\begin{array}{r} \text{Π.χ.} \quad 11000 \\ - \quad 1111 \\ \hline 1001 \end{array}$$

Πολλαπλασιασμός: Εκτελείται κατά τον ίδιο τρόπο όπως και ο δεκαδικός, ισχύουν δε οι ακόλουθοι νόμοι:

A	B	A x B
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{array}{r} \text{Π.χ.} \quad 110,1 \\ \times \quad 10,1 \\ \hline 1101 \\ 0000 \\ \hline 1101 \\ \hline 10000,01 \end{array}$$

Διαίρεση: Εκτελείται κατά τον ίδιο τρόπο όπως και η δεκαδική, ισχύουν δε οι ακόλουθοι νόμοι:

A	B	A : B
0	1	0
1	1	1

$$\begin{array}{r} \text{Π.χ.} \quad 101011111 \mid 1101 \\ \underline{1101} \quad \mid 11011 \\ 010001 \\ \underline{1101} \\ 0010011 \\ \underline{1101} \\ 001101 \\ \underline{1101} \\ 0000 \end{array}$$

2.6. Προσημασμένοι αριθμοί. Συμπληρώματα αριθμών

Οι αριθμοί (θετικοί και αρνητικοί) μπορούν να παρασταθούν στο δυαδικό σύστημα με δύο τρόπους:

1) Οι θετικοί αριθμοί μπορούν να παρασταθούν ως αριθμοί χωρίς πρόσημο. Το ίδιο ισχύει και για το μηδέν. Αντίθετα οι αρνητικοί αριθμοί χρειάζονται ένα συμβολισμό

για να δηλώσουμε την αρνητική τους τιμή. Ενώ στη συνηθισμένη αριθμητική ένας αρνητικός αριθμός δηλώνεται με το πρόσημο μείον και ένας θετικός με το πρόσημο συν, στους υπολογιστές όπου τα πάντα συμβολίζονται με δυαδικά ψηφία, τα δύο αυτά πρόσημα απεικονίζονται με δυαδικά ψηφία που λέγονται **bits**. Έχει επικρατήσει να απεικονίζεται το πρόσημο ενός αριθμού με την τοποθέτηση ενός επιπλέον bit (**bit προσήμου**) στη αριστερότερη θέση του αριθμού. Εάν το bit προσήμου είναι 0 τότε ο αριθμός που ακολουθεί είναι θετικός, εάν είναι 1 τότε ο αριθμός που ακολουθεί είναι αρνητικός. Π.χ. για αριθμό ο οποίος απεικονίζεται με οκτώ ψηφία, το 9 μπορεί να γραφεί:

$$\begin{array}{rcl} 00001001 & = & + (9)_{10} \\ 10001001 & = & - (9)_{10} \end{array}$$

Στο σημείο αυτό πρέπει να τονίσουμε ότι και οι προσημασμένοι δυαδικοί αριθμοί και αυτοί χωρίς πρόσημο αποτελούνται από ένα πλήθος bits. Ο χρήστης καθορίζει εάν ο αριθμός που έχει μπροστά του έχει πρόσημο ή όχι. Αν ο δυαδικός αριθμός έχει πρόσημο, τότε το αριστερότερο bit παριστάνει το πρόσημό του και τα υπόλοιπα bits αποτελούν την απόλυτη τιμή του αριθμού. Εάν ο δυαδικός αριθμός θεωρείται χωρίς πρόσημο τότε το αριστερότερο bit είναι το MSB (Most Significant Bit) του αριθμού.

Π.χ. ο 00100101 μπορεί να θεωρηθεί ως $(37)_{10}$ (χωρίς πρόσημο) ή ως $(+37)_{10}$ αφού το αριστερότερο bit είναι το μηδέν.

Ομοίως ο 10100101 μπορεί να θεωρηθεί ως $(165)_{10}$ όταν δηλώνει αριθμό χωρίς πρόσημο ή ως $(-37)_{10}$ όταν θεωρείται προσημασμένος αριθμός λόγω του bit 1 στην αριστερότερη θέση, που δηλώνει αρνητικό πρόσημο και των υπολοίπων 7 bits που παριστάνουν το δυαδικό αριθμό $(37)_{10}$. Με τον τρόπο αυτό γίνεται η απεικόνιση των προσημασμένων αριθμών που χρησιμοποιούνται στη συνηθισμένη αριθμητική.

2) Όταν εκτελούνται αριθμητικές πράξεις σε έναν υπολογιστή, είναι πιο βολικό να χρησιμοποιείται για την παράσταση των αρνητικών αριθμών, ένα διαφορετικό σύστημα που ονομάζεται **μέθοδος του συμπληρώματος**. Με τη μέθοδο αυτή κάθε αρνητικός αριθμός δηλώνεται με το συμπλήρωμά του.

Το συμπλήρωμα ως προς 1 (1's complement)

Στο σύστημα αυτό οι μεν θετικοί αριθμοί γράφονται με την γνωστή δυαδική τους μορφή, οι δε αρνητικοί αριθμοί με το συμπλήρωμα ως προς 1. Πρακτικά, για να βρούμε το συμπλήρωμα ως προς 1 ενός αρνητικού δυαδικού αριθμού αντικαθιστούμε όλα τα 1 του αριθμού με αντίστοιχα 0 και όλα τα 0 με αντίστοιχα 1.

Π.χ. το 1's complement του δυαδικού αριθμού 1011011001 είναι:

$$(1's) 1011011001 = 0100100110$$

Το συμπλήρωμα ως προς 2 (2's complement)

Για να βρούμε το συμπλήρωμα ως προς 2 ενός δυαδικού αριθμού (αρνητικού), βρίσκουμε πρώτα το συμπλήρωμα ως προς 1 του δυαδικού αριθμού και σε αυτό προσθέτουμε τη μονάδα.

Π.χ. το 2's complement του δυαδικού αριθμού 1100 είναι:

$$\begin{array}{r} (1's) 1100 = 0011 \\ + \quad \quad 1 \\ \hline 0100 \end{array}$$

Με τη βοήθεια του συμπληρώματος οι πράξεις μεταξύ των δυαδικών αριθμών γίνονται γρηγορότερα και ευκολότερα, η δε αφαίρεση ανάγεται σε πρόσθεση.

Παράδειγμα

ΔΕΚΑΔΙΚΟΙ	ΔΥΑΔΙΚΟΙ	ΣΥΜΠΛΗΡΩΜΑΤΑ	
		1's	2's
19	10011	10011	10011
-9	- 01001	+ 10110	+ 10111
10	01010	1 01001	1 01010
		└─ + 1	└─
		01010	αννοείται

Στο συμπλήρωμα ως προς 1 (1's) η υπερχειλίση, εφόσον υπάρχει προστίθεται ενώ στο συμπλήρωμα ως προς 2 (2's) αγνοείται. Στις αφαιρέσεις ο μειωτέος και ο αφαιρετέος πρέπει να παριστάνονται με τον ίδιο αριθμό ψηφίων (bits). Οι υπολογιστές λειτουργούν με τη μέθοδο του συμπληρώματος ως προς 2 (το γεγονός ότι το τελικό κρατούμενο αγνοείται είναι πλεονέκτημα).

Σημείωση: Σε κάθε αριθμητικό σύστημα με βάση r υπάρχουν δύο συμπληρώματα: (α) το συμπλήρωμα ως προς r και (β) το συμπλήρωμα ως προς $(r-1)$. Αν τώρα αντικατασταθεί η τιμή της βάσης με το 10 τότε έχουμε τα συμπληρώματα ως προς 10 και ως προς 9, για τους δεκαδικούς αριθμούς.

Συμπλήρωμα ως προς Βάση μείον 1

Ένας αριθμός N με ψηφία σε βάση r , έχει συμπλήρωμα ως προς $(r-1)$ που ορίζεται από τη σχέση $(r^n - 1) - N$. Για δεκαδικούς αριθμούς, $r=10$ και $r-1=9$ το συμπλήρωμα ως προς 9 του N είναι $(10^n - 1) - N$. Το 10^n είναι ένας αριθμός που αποτελείται από ένα μόνο 1 ακολουθούμενο από n μηδέν. Το 10^n-1 είναι ένας αριθμός που αποτελείται από n 9. Αν $n=5$ έχουμε: $10^5 = 100000$ και $10^5-1=99999$.

Από τα παραπάνω συμπεραίνουμε ότι το συμπλήρωμα ως προς 9, ενός δεκαδικού αριθμού, προκύπτει με αφαίρεση κάθε ψηφίου από το 9. Π.χ. το συμπλήρωμα ως προς 9 του 465320 είναι: $999999 - 465320 = 534679$.

Το συμπλήρωμα ως προς 9 του 002179 είναι: $999999 - 002179 = 997820$.

Σύμφωνα με όσα αναφέραμε το συμπλήρωμα ως προς $(r-1)$ των οκταδικών ή δεκαεξαδικών αριθμών προκύπτει με αφαίρεση κάθε ψηφίου τους από το 7 ή το F, αντίστοιχα.

Συμπλήρωμα ως προς Βάση

Ένας αριθμός N με ψηφία σε βάση r , έχει συμπλήρωμα ως προς r που ορίζεται από τη σχέση:

$$r^n - N \text{ για } N \neq 0 \text{ και } 0 \text{ για } N=0.$$

Συγκρίνοντας το συμπλήρωμα ως προς βάση με το συμπλήρωμα ως προς βάση μείον 1, παρατηρούμε ότι το συμπλήρωμα ως προς βάση προκύπτει αν προσθέσουμε ένα (1) στο συμπλήρωμα ως προς $(r-1)$ αφού ισχύει: $r^n - N = [(r^n - 1) - N] + 1$. Έτσι, το συμπλήρωμα ως προς 10 του δεκαδικού 4328 είναι $5671 + 1 = 5672$ που προκύπτει αν προσθέσουμε 1 στο συμπλήρωμα ως προς 9.

Εάν ο αριθμός N , του οποίου ζητάμε το συμπλήρωμα περιέχει υποδιαστολή, αυτή αφαιρείται προσωρινά, για να σχηματίσουμε το συμπλήρωμα ως προς r ή $r-1$. Μετά η υποδιαστολή ξαναπαίρνει στο νέο αριθμό (συμπλήρωμα) και στην ίδια σχετικά θέση.

Επίσης, πρέπει να τονίσουμε ότι το συμπλήρωμα του συμπληρώματος δίνει τον αρχικό αριθμό. Αυτό διότι: $r^n - (r^n - N) = N$.

2.7. Αφαίρεση δύο αριθμών με τη βοήθεια συμπληρώματος

Για να αφαιρέσουμε έναν αριθμό (αφαιρετέο) N , από έναν άλλο αριθμό (μειωτέο) M , είναι γρηγορότερο και ευκολότερο να προσθέσουμε στο μειωτέο το συμπλήρωμα του αφαιρετέου. Έτσι, εξάλλου πραγματοποιούν την πράξη της αφαίρεσης και οι ηλεκτρονικοί υπολογιστές. Η αφαίρεση δύο θετικών αριθμών $(M-N)$ που ανήκουν και οι δύο στην ίδια βάση r , και έχουν n ψηφία, γίνεται ως εξής:

- (1) Προσθέτουμε τον μειωτέο M στο συμπλήρωμα ως προς r του αφαιρετέου N . Δηλαδή έχουμε: $M + (r^n - N) = M - N + r^n$ και διακρίνουμε 2 περιπτώσεις:
- (2) Εάν $M \geq N$ τότε η παραπάνω σχέση γράφεται: $M + (r^n - N) = r^n + M - N$. Στην περίπτωση αυτή, ο όρος r^n είναι το τελικό κρατούμενο της πρόσθεσης το οποίο αγνοείται στο τελικό αποτέλεσμα.
- (3) Εάν $M < N$ τότε η παραπάνω σχέση γράφεται: $M + (r^n - N) = r^n - (N - M)$. Δηλαδή, το αποτέλεσμα της πρόσθεσης δεν έχει τελικό κρατούμενο και είναι το συμπλήρωμα ως προς r του $(N-M)$. Άρα, για να βρούμε το τελικό αποτέλεσμα θα πρέπει να πάρουμε το συμπλήρωμα του αποτελέσματος της πρόσθεσης και να βάλουμε το πρόσημο μείον μπροστά.

Παραδείγματα

(α) $(72532)_{10} - (3250)_{10}$

Επειδή οι δύο αριθμοί (M, N) πρέπει να έχουν τον ίδιο αριθμό ψηφίων, προσθέτουμε ως περισσότερο σημαντικά ψηφία του μικρότερου σε μήκος αριθμού, μηδενικά (ο αριθμός 3250 γράφεται 03250).

Το συμπλήρωμα ως προς 10 του αφαιρετέου $N=03250$ είναι: $100000-03250 = 96750$.

Άρα:
$$\begin{array}{r} M: \quad 72532 \\ N: \quad \underline{96750} \\ \hline 1)69282 \end{array}$$

Το αποτέλεσμα είναι 69282 και το τελικό κρατούμενο 1 παραλείπεται.

(β) $(3250)_{10} - (72532)_{10}$

Το συμπλήρωμα ως προς 10 του αφαιρετέου $N=72532$ είναι: $100000-72532 = 27468$.

Άρα:
$$\begin{array}{r} M: \quad 03250 \\ N: \quad \underline{27468} \\ \hline 30718 \end{array}$$

Επειδή δεν προκύπτει τελικό κρατούμενο, το αποτέλεσμα της αφαίρεσης είναι αρνητικό (δηλ. $M < N$). Άρα για να βρούμε το τελικό αποτέλεσμα θα πρέπει να πάρουμε το συμπλήρωμα του παραπάνω αποτελέσματος και να βάλουμε το πρόσημο μείον μπροστά. Δηλαδή:

$$-(10's) 30718 = -69282.$$

(γ) $(6324)_8 - (3565)_8$

Το συμπλήρωμα ως προς 8 του αφαιρετέου $N=3565$ προκύπτει αν στο συμπλήρωμα ως προς 7 προσθέσουμε ένα 1. Δηλαδή: $(7's) 3565 = 4212$.

Έχουμε: $(8's) 3565 = 4212 + 1 = 4213$.

$$\begin{array}{r} \text{Άρα: } M: \quad 6324 \\ N: \quad \underline{4213} \\ 1)2537 \end{array}$$

Το αποτέλεσμα είναι 2537 και το τελικό κρατούμενο 1 παραλείπεται.

$$(\delta) (3565)_8 - (6324)_8$$

Το συμπλήρωμα ως προς 7 του αφαιρετέου $N=6324$ είναι: $7777 - 6324 = 1453$.

Έχουμε ότι το συμπλήρωμα ως προς 8 είναι: $1453 + 1 = 1454$.

$$\begin{array}{r} \text{Άρα: } M: \quad 3565 \\ N: \quad \underline{1454} \\ 5241 \end{array}$$

Επειδή δεν προκύπτει τελικό κρατούμενο, το αποτέλεσμα της αφαίρεσης είναι αρνητικό (δηλ. $M < N$). Άρα για να βρούμε το τελικό αποτέλεσμα θα πρέπει να πάρουμε το συμπλήρωμα του παραπάνω αποτελέσματος και να βάλουμε το πρόσημο μείον μπροστά. Δηλαδή:

$$-(8's) 5241 = -2537.$$

$$(\epsilon) (18B)_{16} - (7A)_{16}$$

Το συμπλήρωμα ως προς 16 του αφαιρετέου $N=7A$ προκύπτει αν στο συμπλήρωμα ως προς 15 προσθέσουμε ένα 1. Δηλαδή: $(15's) 07A = FFF - 07A = F85$.

Έχουμε: $(16's) 07A = F85 + 1 = F86$.

$$\begin{array}{r} \text{Άρα: } M: \quad 18B \\ N: \quad \underline{F86} \\ 1)111 \end{array}$$

Το αποτέλεσμα είναι 111 και το τελικό κρατούμενο 1 παραλείπεται.

$$(\sigma\tau) (7A)_{16} - (18B)_{16}$$

Το συμπλήρωμα ως προς 15 του αφαιρετέου $N=18B$ είναι: $FFF - 18B = E74$.

Έχουμε ότι το συμπλήρωμα ως προς 16 είναι: $E74 + 1 = E75$.

$$\begin{array}{r} \text{Άρα: } M: \quad 07A \\ N: \quad \underline{E75} \\ EEF \end{array}$$

Επειδή δεν προκύπτει τελικό κρατούμενο, το αποτέλεσμα της αφαίρεσης είναι αρνητικό (δηλ. $M < N$). Άρα για να βρούμε το τελικό αποτέλεσμα θα πρέπει να πάρουμε το συμπλήρωμα του παραπάνω αποτελέσματος και να βάλουμε το πρόσημο μείον μπροστά. Δηλαδή:

$$-(16's) EEF = -111.$$

ζ) Εάν $X = 1010101$ και $Y = 1000100$, να εκτελεστούν οι αφαιρέσεις $X - Y$ και $Y - X$.

$$\begin{array}{r} \text{(i)} \quad X = 1010101 \\ Y = 0111100 \\ \hline \text{άθροισμα} \quad = 1)0010001 \end{array}$$

Με απόρριψη του τελικού κρατουμένου προκύπτει: $X - Y = 00010001$.

$$\begin{array}{r} \text{(ii)} \quad Y = 1000100 \\ X = 0101011 \\ \hline \text{άθροισμα} \quad = 1101111 \end{array}$$

Επειδή δεν προκύπτει τελικό κρατούμενο το τελικό αποτέλεσμα θα είναι:

$$Y - X = -(2's) 1101111 = -0010001.$$

η) Να γίνουν οι παρακάτω αφαιρέσεις, με συμπλήρωμα ως προς 1 (1's) και συμπλήρωμα ως προς 2 (2's).

(i) $110111 - 101110$ (ii) $1111001 - 0110110$ (iii) $11100011 - 11110100$

ΣΥΜΠΛΗΡΩΜΑ ΩΣ ΠΡΟΣ 1

ΣΥΜΠΛΗΡΩΜΑ ΩΣ ΠΡΟΣ 2

(i)

$$\begin{array}{r} 110111 \\ \underline{010001} \\ 1\ 001000 \\ \underline{ + 1} \\ 001001 \end{array}$$

(i)

$$\begin{array}{r} 110111 \\ \underline{010010} \\ 1\ 001001 \\ \leftarrow \text{αννοείται} \end{array}$$

(ii)

$$\begin{array}{r} 1111001 \\ \underline{1001001} \\ 1\ 1000010 \\ \underline{ + 1} \\ 1000011 \end{array}$$

(ii)

$$\begin{array}{r} 1111001 \\ \underline{1001010} \\ 1\ 1000011 \\ \leftarrow \text{αννοείται} \end{array}$$

(iii)

$$\begin{array}{r} 11100011 \\ \underline{00001011} \\ 11101110 \end{array}$$

(iii)

$$\begin{array}{r} 11100011 \\ \underline{00001100} \\ 11101111 \end{array}$$

$-(1's) 11101110 = -00010001 = (-17)_{10}$ $-(2's) 11101111 = -00010001 = (-17)_{10}$

2.8. Κώδικες

2.8.1. Δυαδικοί Κώδικες

Ο άνθρωπος χρησιμοποιεί τη δεκαδική λογική. Αντίθετα, οι ηλεκτρονικοί υπολογιστές λειτουργούν με βάση τη δυαδική λογική. Είναι προφανές ότι υπάρχει πρόβλημα επικοινωνίας του χρήστη με τον ηλεκτρονικό υπολογιστή. Ο χρήστης εισάγει δεδομένα σε δεκαδική μορφή. Ο ηλεκτρονικός υπολογιστής επεξεργάζεται τα δεδομένα σε δυαδική μορφή. Επομένως, απαιτείται η κατάλληλη μετατροπή των πληροφοριών που ονομάζεται **κωδικοποίηση**. **Κώδικας** είναι ένας συστηματικός τρόπος παράστασης πληροφοριών.

Τα ηλεκτρονικά ψηφιακά συστήματα χρησιμοποιούν σήματα που έχουν δύο διακριτές τιμές. Όμως, τα ψηφιακά συστήματα αναπαριστούν και χειρίζονται πολλά διακριτά στοιχεία πληροφορίας και όχι μόνο δυαδικές πληροφορίες. Κάθε διακριτό στοιχείο πληροφορίας μπορεί να παρασταθεί με έναν δυαδικό κώδικα. **Δυαδικός κώδικας** είναι ένας συστηματικός τρόπος παράστασης πληροφοριών σε δυαδική μορφή.

Οι δυαδικοί κώδικες χρησιμοποιούν το δυαδικό ψηφίο (binary digit - bit) με δύο πιθανές τιμές "0" και "1". Με έναν δυαδικό κώδικα που χρησιμοποιεί n bits μπορούν να παρασταθούν το πολύ 2^n διακεκριμένα στοιχεία πληροφορίας, αφού τα n bits μπορούν να τοποθετηθούν στη σειρά με 2^n διαφορετικούς τρόπους (συνδυασμοί).

Τέσσερα στοιχεία μπορούν να παρασταθούν με έναν δυαδικό κώδικα των 2 bits. Κάθε στοιχείο παριστάνεται με έναν από τους τέσσερις τρόπους που μπορούν να τοποθετηθούν στη σειρά αυτά τα 2 bits: 00, 01, 10 και 11. Για παράδειγμα, οι τέσσερις εποχές του χρόνου θα μπορούσαν να παρασταθούν ως εξής:

Άνοιξη	↔	00
Καλοκαίρι	↔	01
Φθινόπωρο	↔	10
Χειμώνας	↔	11

Η παραπάνω αντιστοιχία των εποχών με δυαδικούς αριθμούς είναι ένας δυαδικός κώδικας. Η αντιστοιχία αυτή δεν είναι μοναδική και επιλέγεται ανάλογα με την εφαρμογή.

Αν το πλήθος των στοιχείων που πρόκειται να κωδικοποιηθούν δεν είναι δύναμη του 2, τότε μερικοί από τους συνδυασμούς των bits δεν χρησιμοποιούνται. Για παράδειγμα, τα 10 ψηφία του δεκαδικού συστήματος μπορούν να παρασταθούν με έναν δυαδικό κώδικα των 4 bits. Με 4 bits, όμως, μπορούν να αναπτυχθούν 16 συνδυασμοί. Επομένως, δεν χρησιμοποιούνται 6 συνδυασμοί.

Οι δυαδικοί κώδικες ανήκουν στις δύο ακόλουθες κατηγορίες ανάλογα με τον τρόπο κατασκευής τους:

- **δυαδικοί κώδικες με βάρη**
- **δυαδικοί κώδικες χωρίς βάρη**

2.8.2. Δυαδικοί κώδικες με βάρη

Οι δυαδικοί κώδικες με βάρη κατασκευάζονται με τέτοιον τρόπο ώστε στη θέση κάθε bit του κώδικα να αντιστοιχεί ένα βάρος (κάθε θέση έχει μία αξία).

Οι ακόλουθοι δυαδικοί κώδικες με βάρη στα bits ανάλογα με τη θέση τους, χρησιμοποιούνται για την κωδικοποίηση των 10 ψηφίων του δεκαδικού συστήματος:

- ο BCD κώδικας που χρησιμοποιεί 4 bits με βάρη 8 4 2 1.
- ο κώδικας με βάρη 2 4 2 1 που χρησιμοποιεί 4 bits με βάρη 2 4 2 1.
- ο Biquinary κώδικας που χρησιμοποιεί 7 bits με βάρη 5 0 4 3 2 1 0.

2.8.2.1. Ο κώδικας BCD

Ο κώδικας BCD είναι δυαδικός κώδικας με βάρη, που χρησιμοποιείται για την κωδικοποίηση των 10 ψηφίων του δεκαδικού συστήματος, όπως δηλώνει άλλωστε το όνομά του: Binary Coded Decimal (δυαδικά κωδικοποιημένο δεκαδικό).

Ο κώδικας BCD χρησιμοποιεί 4 bits με βάρη 8 4 2 1 και παρουσιάζεται στον Πίνακα 2.8.1. Ονομάζεται επίσης κώδικας 8421.

Ο κώδικας BCD είναι ένας τρόπος παράστασης των 10 ψηφίων του δεκαδικού συστήματος, το κάθε ένα από τα οποία αντιστοιχεί σε μία τετράδα bits.

Για παράδειγμα, ο δεκαδικός αριθμός 5 αντιστοιχεί στην τετράδα 0101 ($0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 5$).

Πίνακας 2.8.1
Κώδικες

Δεκαδικό ψηφίο	BCD 8421	2421	5211	Excess-3	Biquinary 5043210
0	0000	0000	0000	0011	0100001
1	0001	0001	0001	0100	0100010
2	0010	0010	0011	0101	0100100
3	0011	0011	0101	0110	0101000
4	0100	0100	0111	0111	0110000
5	0101	1011	1000	1000	1000001
6	0110	1100	1010	1001	1000010
7	0111	1101	1100	1010	1000100
8	1000	1110	1110	1011	1001000
9	1001	1111	1111	1100	1010000

2.8.2.2. Μετατροπή από BCD σε δεκαδικό

Για τη μετατροπή ενός BCD αριθμού σε δεκαδικό αριθμό χωρίζεται ο BCD αριθμός σε ομάδες **τεσσάρων** (4) bits και κάθε ομάδα μετατρέπεται στο ισοδύναμο δεκαδικό ψηφίο, σύμφωνα με τον Πίνακα 2.8.1.

Για παράδειγμα, ο BCD αριθμός 1000011000101001 αντιστοιχεί στο δεκαδικό αριθμό 8629 αφού:

1000	0110	0010	1001
8	6	2	9

Παρατήρηση. Ο κώδικας BCD χρησιμοποιεί τους 10 από τους 16 δυνατούς συνδυασμούς των 4 bits. Οι 6 συνδυασμοί 1010, 1011, 1100, 1101, 1110 και 1111 δεν χρησιμοποιούνται.

2.8.2.3. Μετατροπή από δεκαδικό σε BCD

Για τη μετατροπή ενός δεκαδικού αριθμού σε BCD αριθμό, μετατρέπεται κάθε ψηφίο του δεκαδικού αριθμού σε μία ομάδα **τεσσάρων** (4) bits που αποτελούν τον ισοδύναμο BCD αριθμό του κάθε δεκαδικού ψηφίου, σύμφωνα με τον Πίνακα 2.8.1.

Για παράδειγμα, ο δεκαδικός αριθμός 4738 αντιστοιχεί στον BCD αριθμό 0100011000111000 αφού:

4	7	3	8
0100	0110	0011	1000

2.8.2.4. Αριθμοί του κώδικα BCD και δυαδικοί αριθμοί

Ο κώδικας BCD δεν είναι ένα άλλο αριθμητικό σύστημα (όπως το δεκαδικό, το δυαδικό, το οκταδικό, το δεκαεξαδικό), αλλά είναι ένας τρόπος παράστασης των 10 ψηφίων του δεκαδικού συστήματος, το κάθε ένα από τα οποία αντιστοιχεί σε μία τετράδα bits.

Επομένως, είναι σημαντική η διαφορά ανάμεσα στη δυαδική κωδικοποίηση ενός δεκαδικού αριθμού και στη μετατροπή ενός δεκαδικού αριθμού στο δυαδικό σύστημα.

Ο κώδικας BCD είναι ένας άμεσος δυαδικός μετατροπέας μόνο για τους δεκαδικούς αριθμούς 0-9. Για τους δεκαδικούς αριθμούς που είναι μεγαλύτεροι από 9, η κωδικοποίηση και η μετατροπή είναι διαφορετικές.

Για παράδειγμα, ο δεκαδικός αριθμός 253 αντιστοιχεί:

- στον 12-bits BCD αριθμό 001001010011
- στον 8-bits δυαδικό αριθμό 11111101

2.8.3. Δυαδικοί κώδικες χωρίς βάρη

2.8.3.1. Ορισμοί

Στους δυαδικούς κώδικες χωρίς βάρη η θέση κάθε bit του κώδικα δεν αντιστοιχεί σε κάποιο βάρος, όπως γίνεται στους δυαδικούς κώδικες με βάρη. Αυτοί οι κώδικες προκύπτουν από κάποιον κανόνα.

Τέτοιοι δυαδικοί κώδικες χωρίς βάρη είναι οι ακόλουθοι:

- ο κώδικας Gray
- ο κώδικας υπερβολής κατά 3 (Excess-3)

Ο κώδικας Excess-3 χρησιμοποιεί επίσης 4 bits για να παραστήσει τους δεκαδικούς αριθμούς 0 έως 9, όπως φαίνεται στον Πίνακα 2.8.1. Το όνομά του προκύπτει από το ότι η παράσταση κάθε δεκαδικού ψηφίου 0 έως 9 στον κώδικα αυτόν, είναι μεγαλύτερη κατά 3 από την αντίστοιχη του κώδικα BCD, δηλαδή ο κώδικας Excess-3 προκύπτει από τον κώδικα BCD προσθέτοντας 3. Το πλεονέκτημά του σε σχέση με τον κώδικα BCD είναι ότι είναι αυτό-συμπληρούμενος κώδικας όπως θα δούμε ευθύς αμέσως.

Από τον πίνακα 2.8.1 βλέπουμε ότι

$$(1)_{10} = (0100)_{\text{Excess-3}}$$

$$(8)_{10} = (1011)_{\text{Excess-3}}$$

Δηλαδή, τα αριστερά μέλη των ισοτήτων είναι το συμπλήρωμα ως προς 9 το ένα του άλλου, ενώ τα δεξιά μέλη είναι το συμπλήρωμα ως προς 1 το ένα του άλλου. Ομοίως,

$$(4)_{10} = (0111)_{\text{Excess-3}}$$

$$(5)_{10} = (1000)_{\text{Excess-3}}$$

Άρα, ο κώδικας Excess-3 μπορεί να χρησιμοποιηθεί για να εκτελέσουμε την αφαίρεση δεκαδικών αριθμών με τη μέθοδο συμπληρώματος ως προς 9. Όμως, ο κώδικας Excess-3 δεν είναι κώδικας με βάρη όπως ο BCD. Για παράδειγμα,

$$\left(\frac{0111}{4} + \frac{0101}{2} \neq \frac{1001}{6} \right)_{\text{Excess-3}}$$

Ενώ

$$\left(\frac{0100}{4} + \frac{0101}{5} = \frac{1001}{9} \right)_{\text{BCD}}$$

Ο κώδικας 2421 συνδυάζει τα πλεονεκτήματα τόσο του BCD όσο και του Excess-3, δηλαδή, είναι αυτό-συμπληρούμενος και κώδικας με βάρη. Για παράδειγμα,

$$(2)_{10} = (0010)_{2421}$$

$$(7)_{10} = (1101)_{2421}$$

Δηλαδή, τα αριστερά μέλη των ισοτήτων είναι το συμπλήρωμα ως προς 9 το ένα του άλλου, ενώ τα δεξιά μέλη είναι το συμπλήρωμα ως προς 1 το ένα του άλλου. Επίσης,

$$\left(\frac{0100}{4} + \frac{1011}{5} = \frac{1111}{9} \right)_{2421}$$

2.8.3.2. Ο Κώδικας GRAY

Ο κώδικας Gray είναι δυαδικός κώδικας χωρίς βάρη, μη αυτό-συμπληρούμενος, που χρησιμοποιείται για την κωδικοποίηση των δεκαδικών αριθμών (όχι μόνο των 10 ψηφίων του δεκαδικού συστήματος, όπως γίνεται στον κώδικα BCD).

Ο κώδικας Gray που χρησιμοποιεί 4 bits (κωδικοποίηση των 16 πρώτων δεκαδικών αριθμών 0-15) παρουσιάζεται στον Πίνακα 2.8.2.

Ο κώδικας Gray ονομάζεται κατοπτρικός κώδικας, λόγω του τρόπου κατασκευής του.

Στον Πίνακα 2.8.2 φαίνεται ότι:

Η πρώτη στήλη από δεξιά (LSB) ξεκινάει πρώτα με ένα "0" και μετά με ένα "1". Αυτά είναι τα 2 πρώτα κατακόρυφα bits. Τα επόμενα 2 κατακόρυφα bits είναι κατοπτρικά των 2 πρώτων bits (υπάρχει συμμετρία ως προς τη μέση τους). Έτσι, δημιουργούνται 4 bits. Τα επόμενα 4 κατακόρυφα bits είναι κατοπτρικά των 4 πρώτων bits. Έτσι, δημιουργούνται 8 bits. Τα επόμενα 8 bits είναι κατοπτρικά των 8 πρώτων bits.

Η δεύτερη στήλη από δεξιά ξεκινάει πρώτα με δύο "0" και μετά με δύο "1". Τα επόμενα 4 bits είναι κατοπτρικά των 4 πρώτων bits. Έτσι, δημιουργούνται 8 bits. Τα επόμενα 8 bits είναι κατοπτρικά των 8 πρώτων bits.

Πίνακας 2.8.2
Κώδικας Gray

Δεκαδικός Αριθμός	Gray
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

Αυτή η διαδικασία επαναλαμβάνεται και στις επόμενες στήλες. Η τρίτη στήλη από δεξιά ξεκινάει πρώτα με τέσσερα "0" και μετά με τέσσερα "1" και είναι κατοπτρική ως προς το μέσον της. Η τέταρτη στήλη από δεξιά ξεκινάει πρώτα με οκτώ "0" και μετά με οκτώ "1".

Ο κώδικας Gray έχει το εξής σημαντικό χαρακτηριστικό: **στον κώδικα Gray αλλάζει ένα μόνο bit μεταξύ δύο διαδοχικών αριθμών**. Για παράδειγμα, οι διαδοχικοί αριθμοί 5 και 6 του κώδικα Gray είναι 0111 και 0101, αντίστοιχα, δηλαδή αλλάζει μόνο το δεύτερο bit από δεξιά. Επίσης, οι διαδοχικοί αριθμοί 7 και 8 του κώδικα Gray είναι 0100 και 1100, αντίστοιχα, δηλαδή αλλάζει μόνο το τέταρτο bit από αριστερά. Αυτό δεν συμβαίνει στο δυαδικό σύστημα. Οι διαδοχικοί αριθμοί 5 και 6 στο δυαδικό σύστημα είναι 0101 και 0110, αντίστοιχα, δηλαδή αλλάζουν τα 2 bits από δεξιά. Επίσης, οι διαδοχικοί αριθμοί 7 και 8 στο δυαδικό σύστημα είναι 0111 και 1000, αντίστοιχα, δηλαδή αλλάζουν και τα 4 bits.

Αν χρησιμοποιούνται δυαδικοί αριθμοί για τη μετάβαση από έναν αριθμό στον επόμενο, τότε υπάρχει η πιθανότητα σφάλματος: η μετάβαση από το 0111 (7) στο 1000 (8) μπορεί να οδηγήσει (για μικρό χρονικό διάστημα) στο 0110 (4) αν το LSB αλλάζει γρηγορότερα από τα άλλα bits, με αποτέλεσμα να γίνει λάθος στη μετατροπή. Αν χρησιμοποιείται ο κώδικας Gray για τη μετάβαση από έναν αριθμό στον επόμενο, τότε η πιθανότητα σφάλματος εξαλείφεται: η μετάβαση από το 0100 (7) στο 1100 (8) επιτυγχάνεται με την αλλαγή ενός (1) μόνο bit.

2.8.4. Αλφαριθμητικοί κώδικες

2.8.4.1. Ορισμοί

Πολλές εφαρμογές των ηλεκτρονικών υπολογιστών απαιτούν τη χρήση δεδομένων που αποτελούνται από αριθμούς αλλά και από γράμματα και από ειδικούς χαρακτήρες. Για παράδειγμα, το λογιστήριο μιας εταιρείας χρησιμοποιεί ηλεκτρονικό υπολογιστή για να επεξεργάζεται τα αρχεία της μισθοδοσίας της εταιρείας. Για να παρασταθούν τα ονόματα των εργαζομένων σε δυαδική μορφή, πρέπει να υπάρχει ένας δυαδικός κώδικας για το αλφάβητο. Για να παρασταθούν οι μισθοί των εργαζομένων σε δυαδική μορφή πρέπει να υπάρχει ένας δυαδικός κώδικας για τους δεκαδικούς αριθμούς και για κάποιους ειδικούς χαρακτήρες, όπως είναι ο χαρακτήρας "\$".

Οι **αλφαριθμητικοί χαρακτήρες** περιλαμβάνουν:

- τα 26 κεφαλαία γράμματα του αγγλικού αλφαβήτου A-Z
- τα 26 μικρά γράμματα του αγγλικού αλφαβήτου a-z
- τα 10 δεκαδικά ψηφία 0-9
- τους ειδικούς χαρακτήρες (τα σημεία στίξης όπως ! , , ? και άλλοι χαρακτήρες όπως @ # \$ % & * + /).

Ένας **αλφαριθμητικός κώδικας** είναι ένας συστηματικός τρόπος παράστασης των αλφαριθμητικών χαρακτήρων σε δυαδική μορφή. Κάθε αλφαριθμητικός χαρακτήρας παριστάνεται με μία ομάδα bits, το μέγεθος της οποίας εξαρτάται από το πλήθος των αλφαριθμητικών χαρακτήρων που παριστάνει ο κώδικας.

Τέτοιοι δυαδικοί αλφαριθμητικοί κώδικες είναι οι ακόλουθοι:

- ο κώδικας ASCII που χρησιμοποιεί 7 bits
- ο κώδικας Baudot που χρησιμοποιεί 5 bits

2.8.4.2. Ο Κώδικας ASCII

Ο πλέον συχνά χρησιμοποιούμενος δυαδικός αλφαριθμητικός κώδικας είναι ο κώδικας ASCII (American Standard Code for Information Interchange) ο οποίος χρησιμοποιεί 7 bits για την κωδικοποίηση 128 χαρακτήρων.

Πίνακας 2.8.3
Κώδικας ASCII

b ₇ b ₆ b ₅ \ b ₄ b ₃ b ₂ b ₁	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	'	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	-	o	DEL

Ο κώδικας ASCII περιλαμβάνει 94 εκτυπώσιμους γραφικούς χαρακτήρες και 34 μη εκτυπώσιμους χαρακτήρες ελέγχου (control characters), δηλαδή συνολικά 128 χαρακτήρες που παρουσιάζονται στον Πίνακα 2.8.3.

Οι εκτυπώσιμοι χαρακτήρες είναι:

- τα 26 κεφαλαία γράμματα του αγγλικού αλφαβήτου A-Z
- τα 26 μικρά γράμματα του αγγλικού αλφαβήτου a-z
- οι 10 αριθμοί 0-9
- οι 32 ειδικοί χαρακτήρες.

Οι χαρακτήρες ελέγχου χωρίζονται σε:

διαμορφωτές μορφής
 διαχωριστές πληροφορίας
 χαρακτήρες ελέγχου-επικοινωνίας.

Οι ηλεκτρονικοί υπολογιστές συνήθως χρησιμοποιούν δυαδικές λέξεις των 8 bits (1 byte), ενώ ο κώδικας ASCII χρησιμοποιεί 7 bits. Έτσι, κάθε χαρακτήρας του κώδικα ASCII συνήθως αναπαρίσταται με 1 byte των 8 bits, οπότε μπορεί να γίνει κωδικοποίηση 256 χαρακτήρων. Για την κωδικοποίηση των 128 χαρακτήρων του κώδικα ASCII χρησιμοποιείται το MSB με τιμή "0" (και τα υπόλοιπα 7 bits είναι τα 7 bits του κώδικα ASCII).

Παράδειγμα: Η λέξη **bit** στον κώδικα ASCII είναι:

b	i	t
1100010	1101001	1110100

Για την κωδικοποίηση άλλων χαρακτήρων (για παράδειγμα τα γράμματα του ελληνικού αλφαβήτου) χρησιμοποιείται το MSB με τιμή “1”. Με τον τρόπο αυτόν έχει προκύψει το Πρότυπο ΕΛΟΤ-928 του Ελληνικού Οργανισμού Τυποποίησης που είναι εγκεκριμένο από την ISO (International Standards Organisation).

3. ΑΛΓΕΒΡΑ BOOLE

3.1. Θεωρήματα και Αξιώματα της Άλγεβρας Boole

Όπως προαναφέρθηκε, η δυαδική λογική αποτελείται από δυαδικές μεταβλητές και λογικές πράξεις. Οι μεταβλητές συμβολίζονται με γράμματα του αλφαβήτου, όπως A, B, C, X, Y, Z, κλπ. και κάθε μεταβλητή έχει δύο μόνο δυνατές διακριτές τιμές: 1 και 0. Ορίζονται τρεις βασικές λογικές πράξεις: “ΚΑΙ” (AND), “Ή” (OR) και “ΟΧΙ” (NOT).

1. ΚΑΙ (AND): Αυτή η πράξη παριστάνεται με μία τελεία ή και χωρίς (όπως και ο πολλαπλασιασμός στην Άλγεβρα των πραγματικών αριθμών). Π.χ., $x \cdot y = z$ ή $xy = z$ διαβάζεται “x ΚΑΙ y ίσον z”. Η λογική πράξη ΚΑΙ (AND) σημαίνει ότι το $z = 1$ όταν και μόνον όταν το $x = 1$ και το $y = 1$, διαφορετικά το $z = 0$.
2. Ή (OR): Αυτή η πράξη παριστάνεται με το σύμβολο συν. Π.χ., $x + y = z$ διαβάζεται “x Ή y ίσον z”, που σημαίνει ότι το $z = 1$ αν το $x = 1$ ή αν το $y = 1$ ή αν και τα δύο είναι $x = 1$ και $y = 1$. Αν και τα δύο είναι $x = 0$ και $y = 0$, τότε $z = 0$.
3. ΟΧΙ (NOT): Αυτή η πράξη συμβολίζεται με μια παύλα πάνω από το γράμμα. Π.χ. $\bar{x} = z$ διαβάζεται “όχι x ίσον z”, που σημαίνει ότι το z είναι το “αντίθετο” ή “αντίστροφο” του x: αν το $x = 1$, τότε το $z = 0$ και αν το $x = 0$, τότε το $z = 1$.

Δίνουμε επίσης τους πίνακες αληθείας των τριών βασικών λογικών πράξεων. Για κάθε συνδυασμό τιμών των x και y υπάρχει μια τιμή του z που προσδιορίζεται από τον ορισμό της λογικής πράξης. Αυτοί οι ορισμοί μπορούν να συγκεντρωθούν σε αυτό που λέμε “πίνακα αληθείας” (truth table). Ο πίνακας αληθείας είναι ένας πίνακας όλων των δυνατών συνδυασμών των τιμών μεταβλητών, όπου φαίνεται η σχέση μεταξύ των τιμών που μπορούν να πάρουν οι μεταβλητές και του αποτελέσματος της πράξης. Στη συνέχεια, δίνονται οι πίνακες αληθείας των τριών βασικών λογικών πράξεων.

ΚΑΙ		
x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Ή (OR)		
x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

ΟΧΙ (NOT)	
x	\bar{x}
0	1
1	0

Η άλγεβρα Boole είναι μια αλγεβρική δομή ορισμένη πάνω σ’ ένα σύνολο στοιχείων B, μαζί με τους δύο δυαδικούς τελεστές (πράξεις) + και \cdot , αρκεί να ικανοποιούνται τα παρακάτω αξιώματα:

1. (α) Κλειστή ως προς τον τελεστή + .
(β) Κλειστή ως προς τον τελεστή \cdot .
2. (α) Ένα ουδέτερο στοιχείο ως προς +, που συμβολίζεται με 0: $x + 0 = 0 + x = x$.
(β) Ένα ουδέτερο στοιχείο ως προς \cdot , που συμβολίζεται με 1: $x \cdot 1 = 1 \cdot x = x$.

3. (α) Αντιμεταθετική ως προς +: $x + y = y + x$.
 (β) Αντιμεταθετική ως προς \cdot : $x \cdot y = y \cdot x$.
4. (α) $O \cdot$ είναι επιμεριστικός ως προς +: $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.
 (β) $O +$ είναι επιμεριστικός ως προς \cdot : $x + (y \cdot z) = (x + y) \cdot (x + z)$.
5. Για κάθε στοιχείο $x \in B$, υπάρχει ένα στοιχείο $\bar{x} \in B$ (που ονομάζεται “συμπλήρωμα” του x) τέτοιο ώστε:
 (α) $x + \bar{x} = 1$ και (β) $x \cdot \bar{x} = 0$.
6. Υπάρχουν τουλάχιστον δύο στοιχεία $x, y \in B$ που να είναι $x \neq y$.

Τα ανωτέρω αξιώματα δόθηκαν κατά ζεύγη, καθένα με ένα μέρος (α) και ένα μέρος (β). Το ένα μέρος μπορεί να προκύψει από το άλλο, αν αλλάξουμε τους δυαδικούς τελεστές (δηλ. την πράξη OR σε AND και αντίστροφα) και τα ουδέτερα στοιχεία μεταξύ τους (δηλ. βάζοντας 1 όπου υπάρχει 0 και αντίστροφα). Αυτή η σημαντική ιδιότητα της άλγεβρας Boole ονομάζεται “αρχή του δυϊσμού” (duality principle). Με βάση τα ανωτέρω αξιώματα αποδεικνύονται τα πιο κάτω βασικότερα θεωρήματα της άλγεβρας Boole.

ΘΕΩΡΗΜΑ 1 (α): $x + x = x$.

$$\begin{aligned} x + x &= (x + x) \cdot 1 && \text{από το αξίωμα : 2(β)} \\ &= (x + x) (x + \bar{x}) && 5(\alpha) \\ &= x + x\bar{x} && 4(\beta) \\ &= x + 0 && 5(\beta) \\ &= x && 2(\alpha) \end{aligned}$$

ΘΕΩΡΗΜΑ 1 (β): $x \cdot x = x$.

$$\begin{aligned} x \cdot x &= xx + 0 && \text{από το αξίωμα : 2(α)} \\ &= xx + x\bar{x} && 5(\beta) \\ &= x (x + \bar{x}) && 4(\alpha) \\ &= x \cdot 1 && 5(\alpha) \\ &= x && 2(\beta) \end{aligned}$$

ΘΕΩΡΗΜΑ 2 (α): $x + 1 = 1$.

$$\begin{aligned} x + 1 &= 1 \cdot (x + 1) && \text{από το αξίωμα : 2(β)} \\ &= (x + \bar{x}) (x + 1) && 5(\alpha) \\ &= x + \bar{x} \cdot 1 && 4(\beta) \\ &= x + \bar{x} && 2(\beta) \\ &= 1 && 5(\alpha) \end{aligned}$$

ΘΕΩΡΗΜΑ 2 (β): $x \cdot 0 = 0$ από τον δυϊσμό.

ΘΕΩΡΗΜΑ 3 : $\bar{\bar{x}} = x$. Από το αξίωμα 5, έχουμε ότι $x + \bar{x} = 1$ και $x\bar{x} = 0$, που ορίζουν το συμπλήρωμα του x . Άρα το συμπλήρωμα του \bar{x} είναι το x και επίσης το $\bar{\bar{x}}$. Επομένως, αφού το συμπλήρωμα είναι μοναδικό, έχουμε ότι $\bar{\bar{x}} = x$.

ΘΕΩΡΗΜΑ 4 (α): $x + xy = x$.

$$\begin{aligned} x + xy &= x \cdot 1 + xy && \text{από το αξίωμα : 2(β)} \\ &= x(1 + y) && 5(α) \\ &= x(y + 1) && 4(α) \\ &= x \cdot 1 && 5(α) \\ &= x && 2(β) \end{aligned}$$

ΘΕΩΡΗΜΑ 4 (β): $x(x + y) = x$ από τον δυϊσμό.

Τα θεωρήματα της άλγεβρας Boole μπορούν να αποδειχτούν και με τη βοήθεια των πινάκων αληθείας.

ΘΕΩΡΗΜΑ 5 (α): $x + (y + z) = (x + y) + z$.

ΘΕΩΡΗΜΑ 5 (β): $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ από τον δυϊσμό.

x	y	z	(y + z)	(x + y)	x + (y + z)	(x + y) + z
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

ΘΕΩΡΗΜΑ 6 (α): $\overline{(x + y)} = \bar{x} \bar{y}$ (De Morgan).

ΘΕΩΡΗΜΑ 6 (β): $\overline{(xy)} = \bar{x} + \bar{y}$ από τον δυϊσμό (De Morgan).

x	y	x + y	$\overline{(x + y)}$	\bar{x}	\bar{y}	$\bar{x} \bar{y}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Η γενικευμένη μορφή του ανωτέρω θεωρήματος De Morgan λέει ότι το συμπλήρωμα μιας συνάρτησης μπορεί να βρεθεί εναλλάσσοντας τους τελεστές AND με OR και συμπληρώνοντας κάθε όρο. Π.χ. το συμπλήρωμα της λογικής συνάρτησης $F = \bar{x}yz + \bar{x} \bar{y}z$ θα δίνεται από

$$\bar{F} = \overline{(\bar{x}yz + \bar{x} \bar{y}z)} = \overline{(\bar{x}yz)} \overline{(\bar{x} \bar{y}z)} = (x + \bar{y} + z)(x + y + \bar{z})$$

Στον πιο κάτω πίνακα δίνονται συνοπτικά τα αξιώματα και τα κυριότερα θεωρήματα της άλγεβρας Boole.

Αξίωμα 2	(α) $x + 0 = x$	(β) $x \cdot 1 = x$
Αξίωμα 5	(α) $x + \bar{x} = 1$	(β) $x \cdot \bar{x} = 0$
Θεώρημα 1	(α) $x + x = x$	(β) $x \cdot x = x$
Θεώρημα 2	(α) $x + 1 = 1$	(β) $x \cdot 0 = 0$
Θεώρημα 3, (δύο αρνήσεις)	$\bar{\bar{x}} = x$	
Αξίωμα 3, αντιμεταθετική	(α) $x + y = y + x$	(β) $x \cdot y = y \cdot x$
Θεώρημα 4, απορρόφηση	(α) $x + xy = x$	(β) $x(x + y) = x$
Αξίωμα 4, επιμεριστική	(α) $x(y + z) = xy + xz$	(β) $x + yz = (x + y)(x + z)$
Θεώρημα 5, προσεταιριστική	(α) $x + (y + z) = (x + y) + z$	(β) $x(yz) = (xy)z$
Θεώρημα 6, De Morgan	(α) $\overline{(x + y)} = \bar{x} \bar{y}$	(β) $\overline{(xy)} = \bar{x} + \bar{y}$

3.2. Λογικές συναρτήσεις

Μία δυαδική μεταβλητή μπορεί να πάρει την τιμή 0 ή 1. Μία λογική συνάρτηση ή συνάρτηση Boole είναι μία έκφραση που σχηματίζεται από δυαδικές μεταβλητές, τους δύο δυαδικούς τελεστές OR και AND, τον τελεστή NOT, παρενθέσεις και ένα “ίσον”. Για μία δεδομένη τιμή των μεταβλητών, η συνάρτηση μπορεί να είναι είτε 0 είτε 1. Έτσι, υπάρχουν τρεις τρόποι συνδυασμού λογικών μεταβλητών για να φτιάξουμε λογικές συναρτήσεις:

- 1) Οι λογικές μεταβλητές να συνδέονται με την πράξη OR
π.χ. $F = A + B + C + D$
- 2) Οι λογικές μεταβλητές να συνδέονται με την πράξη AND
π.χ. $F = A \cdot B \cdot C$
- 3) Οι λογικές μεταβλητές να συνδέονται και με τις δύο πράξεις με τους εξής τρόπους
 $F = AB + BC + AD$ (άθροισμα γινομένων)
 $F = (A+B)(C+D)$ (γινόμενο αθροισμάτων)

Σε οποιαδήποτε από τις παραπάνω μορφές μπορεί να υπάρχει και η πράξη της αντιστροφής. Ας σημειωθεί ότι οι αλγεβρικές εκφράσεις Boole δεν είναι μοναδικές. Άρα, μπορεί να υπάρχουν δύο διαφορετικές αλγεβρικές εκφράσεις που να περιγράφουν την ίδια συνάρτηση.

3.2.1. Απλοποίηση λογικών συναρτήσεων

Με τη βοήθεια των αξιωμάτων και των θεωρημάτων της άλγεβρας Boole μπορούμε πολλές φορές να απλοποιήσουμε τη μορφή κάποιας λογικής συνάρτησης.

Παράδειγμα 3.1:

Δίνεται η λογική συνάρτηση 4 λογικών μεταβλητών A, B, C, και D

$$F = \bar{A}\bar{B}\bar{C}D + \bar{A}BCD + A\bar{B}\bar{C}D + ABCD$$

Να απλοποιηθεί.

Λύση:

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

$$\begin{aligned}
 &= \overline{A}BD \cdot 1 + ABD \cdot 1 \\
 &= \overline{A}BD + ABD \\
 &= BD(\overline{A} + A) = BD \cdot 1 = BD
 \end{aligned}$$

Παράδειγμα 3.2:

Δίνεται η λογική συνάρτηση 3 λογικών μεταβλητών A, B, και C

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

Να απλοποιηθεί.

Λύση:

$$\begin{aligned}
 F &= \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC \\
 &= \overline{A}BC + A\overline{B}C + AB\overline{C} + \overline{A}BC + ABC && [\text{Θεώρημα 1}(\alpha)] \\
 &= (\overline{A} + A)BC + A\overline{B}C + AB\overline{C} + ABC && [\text{Αξίωμα 5}(\alpha)] \\
 &= BC + A\overline{B}C + AB\overline{C} + ABC \\
 &= BC + A\overline{B}C + AB\overline{C} + ABC + ABC && [\text{Θεώρημα 1}(\alpha)] \\
 &= BC + AC(\overline{B} + B) + AB(C + \overline{C}) \\
 &= BC + AC + AB
 \end{aligned}$$

Παράδειγμα 3.3:

Δίνεται η λογική συνάρτηση 4 λογικών μεταβλητών A, B, C και D.

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

Να απλοποιηθεί.

Λύση:

$$\begin{aligned}
 F &= [A\overline{B}(C + BD) + \overline{A}\overline{B}]C = (A\overline{B}C + A\overline{B}BD + \overline{A}\overline{B})C \\
 &= A\overline{B}CC + A\overline{B}BDC + \overline{A}\overline{B}C && [\text{Αξίωμα 5}(\beta)] \\
 &= A\overline{B}CC + \overline{A}\overline{B}C && [\text{Θεώρημα 1}(\beta)] \\
 &= A\overline{B}C + \overline{A}\overline{B}C \\
 &= (A + \overline{A})\overline{B}C && [\text{Αξίωμα 5}(\alpha)] \\
 &= \overline{B}C
 \end{aligned}$$

Μία λογική συνάρτηση μπορεί επίσης να παριστάνεται με τη βοήθεια του πίνακα αληθείας της. Για να παραστήσουμε μια λογική συνάρτηση κατ' αυτόν τον τρόπο, χρειαζόμαστε ένα κατάλογο των 2^n συνδυασμών 1 και 0 των n ανεξάρτητων μεταβλητών της συνάρτησης και μία στήλη που να δείχνει για κάθε συνδυασμό εάν η συνάρτηση ισούται με 1 ή με 0. Έστω λ.χ. για παράδειγμα η συνάρτηση

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

Οι δυνατοί συνδυασμοί τριών μεταβλητών είναι 8. Ο πίνακας αληθείας λοιπόν της συνάρτησης F θα είναι

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

Υπάρχουν βέβαια και άλλες πιο συστηματικές μέθοδοι απλοποίησης λογικών συναρτήσεων όπως λ.χ. η μέθοδος των πινάκων Karnaugh που θα μελετήσουμε αργότερα.

Θα τελειώσουμε δίνοντας δύο ακόμη παραδείγματα απλοποίησης λογικών συναρτήσεων.

Παράδειγμα 3.4:

Να απλοποιηθούν οι πιο κάτω λογικές συναρτήσεις

$$F_1 = \bar{A}\bar{B} + A\bar{B} + \bar{A}B \quad , \quad F_2 = A\bar{B} + AB\bar{C} + ABCD + ABC\bar{D}$$

Λύση:

$$F_1 = \bar{A}\bar{B} + A\bar{B} + \bar{A}B = \bar{B}(A + \bar{A}) + \bar{A}B = \bar{B} \cdot 1 + \bar{A}B = \bar{B} + \bar{A}B.$$

$$F_2 = A\bar{B} + AB\bar{C} + ABCD + ABC\bar{D} = A\bar{B} + AB\bar{C} + ABC(D + \bar{D}) = A(\bar{B} + B\bar{C} + BC) = A[\bar{B} + B(\bar{C} + C)] = A(\bar{B} + B) = A.$$

Παράδειγμα 3.5:

Με χρήση του θεωρήματος De Morgan να απλοποιηθεί η συνάρτηση

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

Λύση :

$$\begin{aligned} F &= \overline{(AC + \bar{A}BC + \bar{B}C)} + ABC\bar{C} = \overline{AC} \cdot \overline{\bar{A}BC} \cdot \overline{\bar{B}C} + ABC\bar{C} = (\bar{A} + \bar{C})(\bar{A} + B + \bar{C})(\bar{B} + \bar{C}) + ABC\bar{C} \\ &= [\bar{A}(A + B + \bar{C}) + \bar{C}(A + B + \bar{C})] \cdot (\bar{B} + \bar{C}) + ABC\bar{C} \\ &= (A\bar{A} + \bar{A}B + \bar{A}\bar{C} + A\bar{C} + \bar{B}\bar{C} + \bar{C}\bar{C})(\bar{B} + \bar{C}) + ABC\bar{C} \\ &= (\bar{A}B + \bar{C})(\bar{B} + \bar{C}) + ABC\bar{C} = \bar{A}B\bar{B} + B\bar{C} + \bar{A}B\bar{C} + \bar{C}\bar{C} + ABC\bar{C} \\ &= \bar{C}(B + \bar{A}B + 1 + AB) = \bar{C}. \end{aligned}$$

3.2.2. Λογικές συναρτήσεις ως άθροισμα γινομένων (SOP : Sum Of Products)

Θα δούμε στη συνέχεια, πώς μπορούμε να βρούμε τον πίνακα αληθείας μιας συνάρτησης αν γνωρίζουμε τη μορφή της.

Έστω η συνάρτηση

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

Παρατηρούμε τα εξής:

- α) Αν κάποιος όρος της συνάρτησης γίνει 1, όλοι οι υπόλοιποι όροι θα είναι μηδενικοί (διότι διαφέρουν τουλάχιστον κατά την τιμή μιας μεταβλητής) και τελικά η τιμή της συνάρτησης θα είναι 1.
- β) Αν όλοι οι όροι είναι μηδενικοί τότε η συνάρτηση θα πάρει τιμή 0.

A	B	C	Γινόμενα
0	0	0	\overline{ABC}
0	0	1	$\overline{AB}C$
0	1	0	$\overline{A}B\overline{C}$
0	1	1	$\overline{A}BC$
1	0	0	$A\overline{B}\overline{C}$
1	0	1	$A\overline{B}C$
1	1	0	$AB\overline{C}$
1	1	1	ABC

Αυτές οι δύο παρατηρήσεις μας οδηγούν στην κατάσταση του πιο πάνω πίνακα. Φτιάχνουμε κατ' αρχήν όλους τους δυνατούς συνδυασμούς των τιμών που μπορούν να πάρουν οι μεταβλητές A, B, C (είναι $2^3 = 8$. Για n μεταβλητές θα έχουμε 2^n δυνατούς συνδυασμούς). Σε κάθε συνδυασμό αντιστοιχεί ένα γινόμενο. Σε κάθε γινόμενο οι μεταβλητές εμφανίζονται κανονικά αν οι τιμές τους είναι 1, ή συμπληρωμένες (δηλ. με αντιστροφή) αν οι τιμές τους στο συνδυασμό είναι 0. Τα γινόμενα αυτά ονομάζονται *βασικά γινόμενα* (fundamental products) ή *ελαχιστόροι* (minterms).

Αν θέλουμε να βρούμε τον πίνακα αληθείας μιας γνωστής συνάρτησης π.χ. της $F = \overline{ABC} + \overline{A}B\overline{C} + ABC$, τότε θέτουμε 1 για κάθε συνδυασμό των A, B, C που το αντίστοιχο βασικό γινόμενο υπάρχει σαν όρος στη δοθείσα συνάρτηση και 0 στις υπόλοιπες περιπτώσεις. Τα αποτελέσματα αυτής της διαδικασίας φαίνονται στον πιο κάτω πίνακα.

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

Πολλές φορές μπορεί σε μια λογική συνάρτηση F να λείπουν από ένα ή και περισσότερους όρους μία ή και περισσότερες μεταβλητές. Π.χ.

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

Σ' αυτήν την περίπτωση με χρήση των θεωρημάτων της άλγεβρας Boole καταλήγουμε σε μορφή όπου θα υπάρχουν όλες οι μεταβλητές. Π.χ.

$$F = A \cdot 1 + BC \cdot 1 + \overline{A}B\overline{C}$$

$$\begin{aligned}
 &= A(B + \bar{B}) + BC(A + \bar{A}) + \bar{A}\bar{B}C \\
 &= AB + A\bar{B} + BCA + BC\bar{A} + \bar{A}\bar{B}C \\
 &= AB \cdot 1 + A\bar{B} \cdot 1 + ABC + \bar{A}BC + \bar{A}\bar{B}C \\
 &= AB(C + \bar{C}) + A\bar{B}(C + \bar{C}) + ABC + \bar{A}BC + \bar{A}\bar{B}C \\
 &= ABC + ABC\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + ABC + \bar{A}BC + \bar{A}\bar{B}C \\
 &= ABC + ABC\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC + \bar{A}\bar{B}C
 \end{aligned}$$

οπότε ο πίνακας αληθείας κατασκευάζεται κατά τα γνωστά.

Εύκολα μπορούμε να λύσουμε και το αντίστροφο πρόβλημα. Δηλαδή, από τον πίνακα αληθείας, να βρούμε τη μορφή της συνάρτησης. Σ' αυτήν την περίπτωση, η συνάρτηση θα είναι ένα άθροισμα βασικών γινομένων που αντιστοιχούν σε συνδυασμούς των A, B, C για τους οποίους η τιμή της F είναι 1. Δίνεται παρακάτω ένα παράδειγμα όπου είναι γνωστός ο πίνακας αληθείας μιας συνάρτησης τριών μεταβλητών.

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

Η λογική συνάρτηση είναι: $F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C} + ABC$.

3.2.3. Λογικές συναρτήσεις ως γινόμενο αθροισμάτων (POS : Product Of Sums)

Ας δούμε κατ' αρχήν πώς μπορούμε να βρούμε τον πίνακα αληθείας αν μας δίνεται η μορφή της συνάρτησης

Έστω π.χ. η συνάρτηση

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

Παρατηρούμε τα εξής:

α) Για να είναι η F μηδενική θα πρέπει κάποιο από τα αθροίσματα να είναι μηδέν (ένα ή και περισσότερα).

β) Αν κάποιο άθροισμα είναι μηδέν τότε όλοι οι προσθετέοι είναι μηδενικοί, πράγμα που αντιστοιχεί σε ένα συνδυασμό τιμών των A, B, C στον πίνακα αληθείας.

Με βάση τα παραπάνω καταστρώνουμε τον παρακάτω πίνακα. Στα βασικά αθροίσματα (fundamental sums) ή μεγιστόρους (maxterms) οι μεταβλητές βρίσκονται στην κανονική μορφή τους αν οι τιμές στους αντίστοιχους συνδυασμούς είναι 0. Ο πίνακας αληθείας της F έχει μηδενικές τιμές αν κάποιο βασικό άθροισμα εμφανίζεται στη συνάρτηση.

A	B	C	Αθροίσματα	F
0	0	0	$A + B + C$ →	0
0	0	1	$A + B + \bar{C}$	1
0	1	0	$A + \bar{B} + C$ →	0
0	1	1	$A + \bar{B} + \bar{C}$	1
1	0	0	$\bar{A} + B + C$ →	0
1	0	1	$\bar{A} + B + \bar{C}$	1
1	1	0	$\bar{A} + \bar{B} + C$	1
1	1	1	$\bar{A} + \bar{B} + \bar{C}$	1

Στο αντίστροφο πρόβλημα, δηλαδή στην εύρεση της συνάρτησης αν είναι δεδομένος ο πίνακας αληθείας, σχηματίζουμε τη συνάρτηση από τα βασικά αθροίσματα που αντιστοιχούν σε 0 τιμές της F. Π.χ.

A	B	C	F	Αθροίσματα
0	0	0	1	$A + B + C$
0	0	1	0 →	$A + B + \bar{C}$
0	1	0	1	$A + \bar{B} + C$
0	1	1	1	$A + \bar{B} + \bar{C}$
1	0	0	1	$\bar{A} + B + C$
1	0	1	0 →	$\bar{A} + B + \bar{C}$
1	1	0	0 →	$\bar{A} + \bar{B} + C$
1	1	1	1	$\bar{A} + \bar{B} + \bar{C}$

Η λογική συνάρτηση είναι: $F = (A + B + \bar{C})(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + C)$.

Πολλές φορές μπορεί σε μια λογική συνάρτηση F να λείπουν από ένα ή και περισσότερους όρους μία ή και περισσότερες μεταβλητές. Π.χ.

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

Σ' αυτήν την περίπτωση με χρήση των θεωρημάτων της άλγεβρας Boole καταλήγουμε σε μορφή όπου θα υπάρχουν όλες οι μεταβλητές. Π.χ.

$$\begin{aligned} F &= AB + \bar{A}C \\ &= (AB + \bar{A})(AB + C) && \text{από το αξίωμα : 4(β)} \\ &= (A + \bar{A})(B + \bar{A})(A + C)(B + C) \\ &= (\bar{A} + B)(A + C)(B + C) \end{aligned}$$

Στη συνέχεια, εισάγουμε σε κάθε όρο την τρίτη μεταβλητή

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

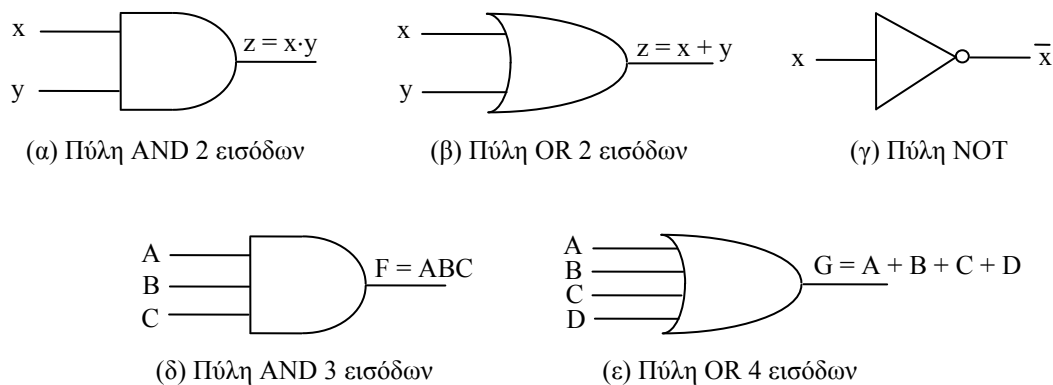
και τελικά

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

οπότε ο πίνακας αληθείας κατασκευάζεται κατά τα γνωστά.

3.3. Ψηφιακές Λογικές Πύλες

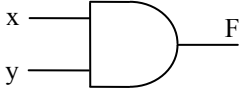
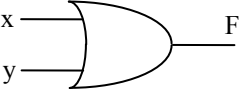
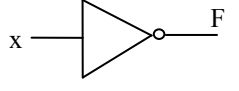
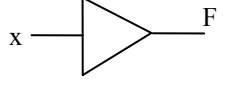
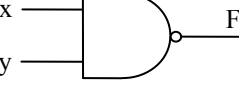
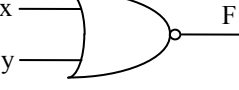


Τα ηλεκτρονικά ψηφιακά κυκλώματα ονομάζονται επίσης “λογικά κυκλώματα” διότι με τις κατάλληλες εισόδους παράγουν εξόδους που είναι λογικές συναρτήσεις. Μπορούμε να επεξεργαστούμε οποιαδήποτε πληροφορία υπολογισμού ή ελέγχου περνώντας δυαδικά σήματα από διάφορους συνδυασμούς λογικών κυκλωμάτων, όπου κάθε σήμα παριστάνει μια μεταβλητή και μεταφέρει ένα bit πληροφορίας. Στο Σχ. 3.1 φαίνονται τα σύμβολα των λογικών κυκλωμάτων που πραγματοποιούν τις λογικές πράξεις AND, OR, NOT. Αυτά τα κυκλώματα ονομάζονται “πύλες” (gates) και παράγουν ένα λογικό “1” ή ένα λογικό “0” ως σήμα εξόδου αν ικανοποιούνται οι κατάλληλες λογικές συνθήκες εισόδου. Οι πύλες AND και OR μπορούν να έχουν περισσότερες από δύο εισόδους. Η πύλη AND με τρεις εισόδους λ.χ., δίνει λογικό “1” στην έξοδο αν και τα τρία σήματα εισόδου είναι λογικό “1” (άρα δίνει λογικό “0” αν οποιαδήποτε είσοδος είναι λογικό “0”). Η πύλη OR με τέσσερις εισόδους δίνει λογικό “1” στην έξοδο, όταν οποιαδήποτε είσοδος είναι λογικό “1” (άρα, η έξοδος γίνεται λογικό “0” αν όλα τα σήματα εισόδου είναι λογικό “0”).



Σχ. 3.1. Σύμβολα ψηφιακών λογικών κυκλωμάτων.

Αφού τις συναρτήσεις Boole τις εκφράζουμε χρησιμοποιώντας τις πράξεις AND, OR, και NOT, είναι απλό το να υλοποιούμε τις συναρτήσεις Boole με αυτούς τους τύπους πυλών. Στο Σχ. 3.2 δίνονται οι τύποι πυλών που χρησιμοποιούνται ως πρότυπες πύλες στο σχεδιασμό ψηφιακών λογικών κυκλωμάτων. Στο ίδιο σχήμα φαίνονται και οι πίνακες αληθείας των πυλών αυτών. Ειδικότερα, η πύλη αντιστροφέα NOT, συμβολίζεται με ένα μικρό κύκλο, στην έξοδο του σχηματικού συμβόλου τριγώνου, ο οποίος συμβολίζει το λογικό συμπλήρωμα. Το τρίγωνο αυτό καθαυτό συμβολίζει ένα κύκλωμα απομονωτή (buffer). Ο απομονωτής υλοποιεί τη συνάρτηση μεταφοράς, χωρίς να κάνει οποιαδήποτε άλλη λογική λειτουργία, αφού η δυαδική τιμή της εξόδου ισούται με τη δυαδική τιμή της εισόδου. Αυτό το κύκλωμα χρησιμοποιείται μόνον για ενίσχυση της ισχύος του σήματος και είναι ισοδύναμο με δύο αντιστροφείς συνδεδεμένους σε σειρά.

Η συνάρτηση NAND είναι το συμπλήρωμα της συνάρτησης AND, γι’ αυτό και συμβολίζεται σαν την AND, αλλά με ένα κυκλάκι στην έξοδο. Οι πύλες NAND και NOR χρησιμοποιούνται ευρύτατα ως πρότυπες λογικές πύλες και στην πραγματικότητα είναι πολύ πιο συνηθισμένες από τις πύλες AND και OR. Αυτό συμβαίνει

Όνομα	Γραφικό Σύμβολο	Αλγεβρική Συνάρτηση	Πίνακας Αληθείας															
AND		$F = xy$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT		$F = \bar{x}$	<table border="1"> <thead> <tr> <th>x</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
Απομονωτής		$F = x$	<table border="1"> <thead> <tr> <th>x</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
NAND		$F = \overline{xy}$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = \overline{x + y}$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR		$F = x\bar{y} + \bar{x}y$ $= x \oplus y$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
XNOR		$F = xy + \bar{x}\bar{y}$ $= x \odot y$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Σχ. 3.2. Ψηφιακές λογικές πύλες.

διότι οι πύλες NAND και NOR κατασκευάζονται ευκολότερα με κυκλώματα transistor και διότι οι συναρτήσεις Boole μπορούν εύκολα να υλοποιηθούν με αυτές.

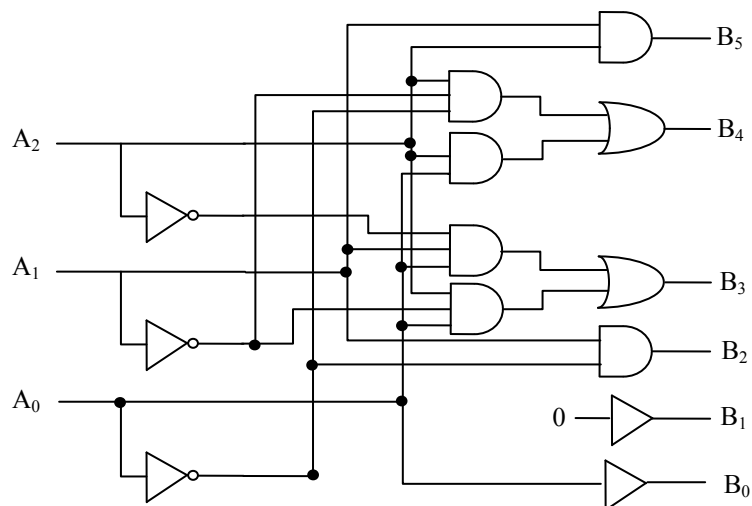
Για την πύλη XOR χρησιμοποιούμε ένα σύμβολο ίδιο με αυτό της πύλης OR, μόνο που έχει μια πρόσθετη καμπύλη στην πλευρά εισόδου. Η πύλη XNOR είναι το συμπλήρωμα της XOR και έτσι χρησιμοποιούμε το ίδιο σύμβολο αλλά με ένα κυκλάκι στην έξοδο. Φυσικά, όλες οι ανωτέρω πύλες μπορούν να επεκταθούν ώστε να έχουν περισσότερες από δύο εισόδους.

Παράδειγμα 3.6: Να σχεδιαστεί συνδυαστικό κύκλωμα που να δέχεται είσοδο τριψήφιο αριθμό (μη προσημασμένο) και να δίνει στην έξοδο το τετράγωνό του.

Λύση:

Δεδομένου ότι ο αριθμός είναι τριψήφιος (δηλ. έχει 3 bits), ο μέγιστος δεκαδικός αριθμός που δέχεται ως είσοδο το κύκλωμα είναι το 7. Το τετράγωνο του αριθμού 7 είναι 49 του οποίου η δυαδική αναπαράσταση είναι 110001. Έχει δηλαδή 6 ψηφία (bits), και επομένως το συνδυαστικό κύκλωμα πρέπει να έχει 6 εξόδους. Δίνουμε τον πίνακα αληθείας του κυκλώματος.

Είσοδοι			Έξοδοι						Δεκαδικός
A ₂	A ₁	A ₀	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀	
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49



Σχ. 3.3. Συνδυαστικό κύκλωμα ύψωσης στο τετράγωνο ενός τριψήφιου αριθμού.

Από τον πίνακα αληθείας προκύπτουν οι εκφράσεις των μεταβλητών εξόδου B_i συναρτήσεις των μεταβλητών εισόδου A_i.

$$B_0 = A_0$$

$$B_1 = 0$$

$$B_2 = \bar{A}_2 A_1 \bar{A}_0 + A_2 A_1 \bar{A}_0 = A_1 \bar{A}_0$$

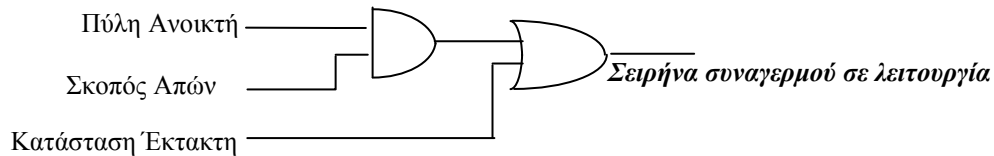
$$B_3 = \bar{A}_2 A_1 A_0 + A_2 \bar{A}_1 A_0$$

$$B_4 = A_2 \bar{A}_1 \bar{A}_0 + A_2 \bar{A}_1 A_0 + A_2 A_1 A_0 = A_2 \bar{A}_1 \bar{A}_0 + A_2 A_0$$

$$B_5 = A_2 A_1 \bar{A}_0 + A_2 A_1 A_0 = A_2 A_1$$

Η κυκλωματική υλοποίηση του συνδυαστικού κυκλώματος που δίνει το τετράγωνο ενός τριγώνιου αριθμού δίνεται στο Σχ. 3.3.

Παράδειγμα 3.7



Στο παραπάνω σχήμα, δίνεται το λογικό διάγραμμα που δείχνει πότε ενεργοποιείται η σειρήνα του συναγερμού σε ένα φυλασσόμενο χώρο. Έκτακτη κατάσταση εννοείται η κατάσταση ύπαρξης εισβολέα στην είσοδο της πύλης.

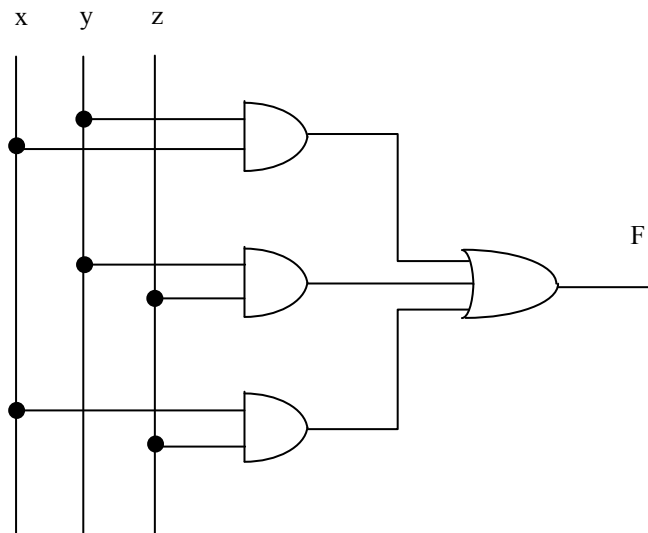
- A) Περιγράψτε τη λειτουργία του κυκλώματος.
- B) Θεωρώντας τις μεταβλητές x, y και z (πύλη ανοικτή, σκοπός απών, κατάσταση έκτακτη, αντίστοιχα), να δημιουργήσετε τον πίνακα αλήθειας ο οποίος να ενεργοποιεί το συναγερμό (συνάρτηση F) όταν:
 - (α) η κατάσταση είναι έκτακτη και συμβαίνει ένα από τα ακόλουθα γεγονότα (ή και τα δύο): Η πύλη είναι ανοικτή ή ο σκοπός είναι απών.
 - (β) Όταν συμβαίνουν ταυτόχρονα τα ακόλουθα δύο γεγονότα: Ο σκοπός είναι απών και η πύλη είναι ανοικτή.
- Γ) Από τον πίνακα αλήθειας να γραφεί η συνάρτηση σαν άθροισμα γινομένων. Να απλοποιηθεί με τη βοήθεια αξιωμάτων και θεωρημάτων της άλγεβρας Boole. Να υλοποιηθεί το λογικό διάγραμμα της νέας συνάρτησης.

Λύση

Η σειρήνα συναγερμού τίθεται σε λειτουργία όταν: (α) η πύλη είναι ανοικτή και ο σκοπός απών ή (β) όταν η κατάσταση είναι έκτακτη.

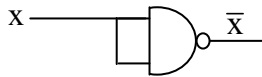
$$F = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz + xyz + xyz = xz(y + \bar{y}) + yz(\bar{x} + x) + xy(z + \bar{z}) = xz + yz + xy$$

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



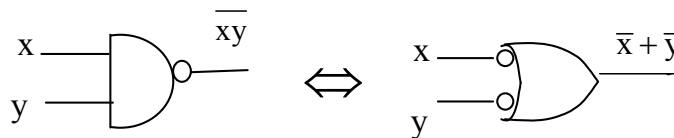
3.4. Υλοποίηση Συναρτήσεων με πύλες NAND

Ένα πρόβλημα που πολύ συχνά αντιμετωπίζουμε στην πράξη είναι η υλοποίηση μίας συνάρτησης με ένα και μόνο είδος πυλών. Η πύλη που μπορεί να υλοποιήσει όλες τις πράξεις είναι η NOR καθώς και η NAND. Ας δούμε πώς μπορούν να υλοποιηθούν όλες οι πύλες με τη βοήθεια της NAND.



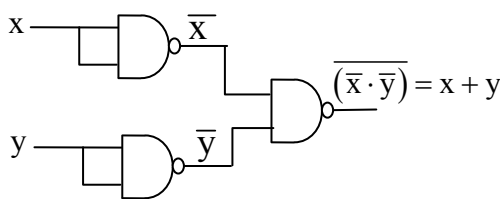
Σχήμα 3.4α. Υλοποίηση της πύλης NOT με NAND.

Από τον πίνακα αληθείας της πύλης NAND παρατηρούμε ότι αν οι δύο εισοδοι της πύλης NAND είναι σε κατάσταση 0, τότε η έξοδος είναι 1. Αν και οι δύο εισοδοι είναι σε κατάσταση 1 τότε η έξοδος είναι 0. Αυτή η λογική πράξη είναι η πράξη της αντιστροφής. Επομένως, το λογικό NOT υλοποιείται αν βραχυκυκλώσουμε τις δύο εισόδους της πύλης NAND.

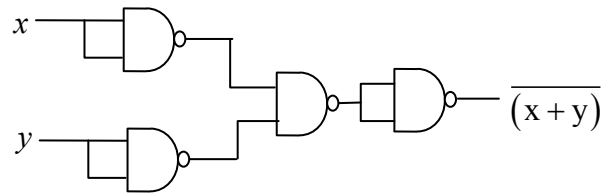


Σχήμα 3.4β. Ισοδυναμία πύλης NAND.

Ο κανόνας De Morgan μας ενημερώνει ότι $\overline{(x \cdot y)} = \bar{x} + \bar{y}$. Αυτό σημαίνει ότι αν στις εισόδους είχαμε \bar{x} και \bar{y} , τότε η έξοδος θα ήταν $x+y$. Η λογική αυτή φαίνεται στην παραπλεύρως ισοδυναμία.

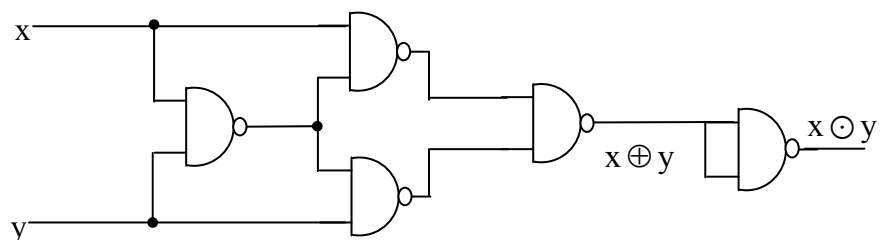
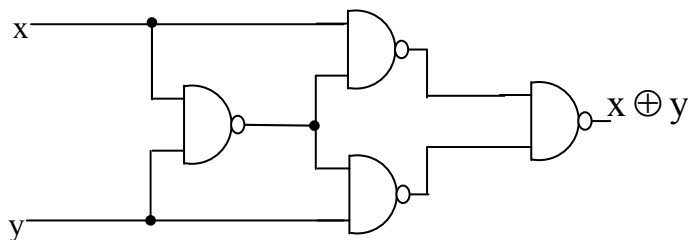


Σχήμα 3.4γ. Υλοποίηση της πύλης OR με πύλες NAND.



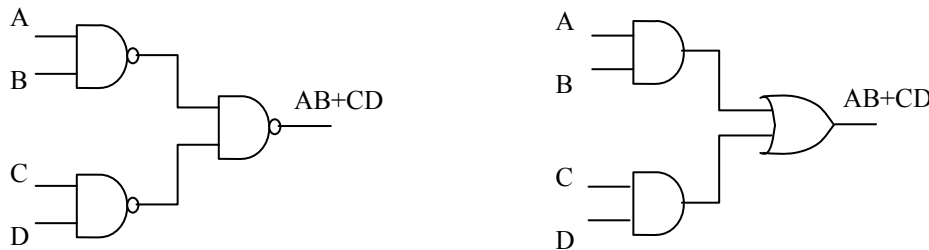
Σχήμα 3.4δ. Υλοποίηση πύλης NOR με πύλες NAND.

Η υλοποίηση της πύλης NOR είναι απλή υπόθεση, εφόσον έχουμε την υλοποίηση της NAND και της NOT. Τέλος, όσον αφορά στην υλοποίηση των 2 πυλών XAND και XOR, με πύλες NAND, αυτές φαίνονται στα σχηματα 3.4ε και 3.4στ αντίστοιχα.



Σχήμα 3.4ε. Υλοποίηση πύλης XOR και (στ) XNOR με πύλες NAND.

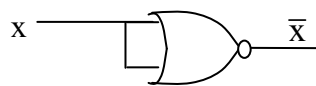
Από το παραπάνω ισοδύναμο της πύλης NAND συμπεραίνουμε ότι ένα λογικό κύκλωμα δύο βαθμίδων με πύλες NAND – NAND μπορεί εύκολα να αντικατασταθεί από ένα κύκλωμα δύο βαθμίδων με πύλες AND – OR.



Σχήμα 3.4ζ. Ισοδύναμα κυκλώματα.

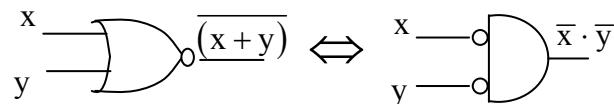
3.5. Υλοποίηση Συναρτήσεων με πύλες NOR

Η υλοποίηση μίας συνάρτησης μπορεί επίσης να γίνει χρησιμοποιώντας μόνο πύλες NOR. Ας δοκιμάσουμε να δούμε πώς μπορούν να υλοποιηθούν όλες οι πύλες με τη βοήθεια της NOR.



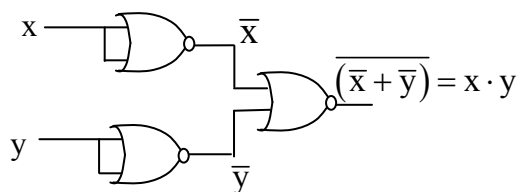
Σχήμα 3.5α. Υλοποίηση της πύλης NOT με NOR.

Από τον πίνακα αλήθειας της πύλης NOR παρατηρούμε ότι αν οι δύο εισοδοί της πύλης NOR είναι σε κατάσταση 0, τότε η έξοδος είναι 1. Αν και οι δύο εισοδοί είναι σε κατάσταση 1 τότε η έξοδος είναι 0. Αυτή η λογική πράξη είναι η πράξη της αντιστροφής. Επομένως, το λογικό NOT υλοποιείται αν βραχυκυκλώσουμε τις δύο εισόδους της πύλης NOR.

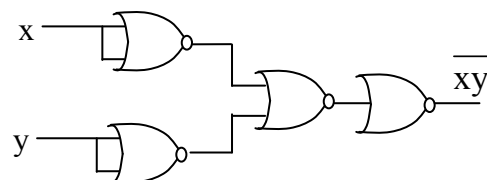


Σχήμα 3.5β. Ισοδυναμία πύλης NOR.

Ο κανόνας De Morgan μας ενημερώνει ότι $\overline{(x + y)} = \bar{x} \cdot \bar{y}$. Αυτό σημαίνει ότι αν στις εισόδους είχαμε \bar{x} και \bar{y} , τότε η έξοδος θα ήταν $x \cdot y$. Η λογική αυτή φαίνεται στην παραπλεύρως ισοδυναμία.



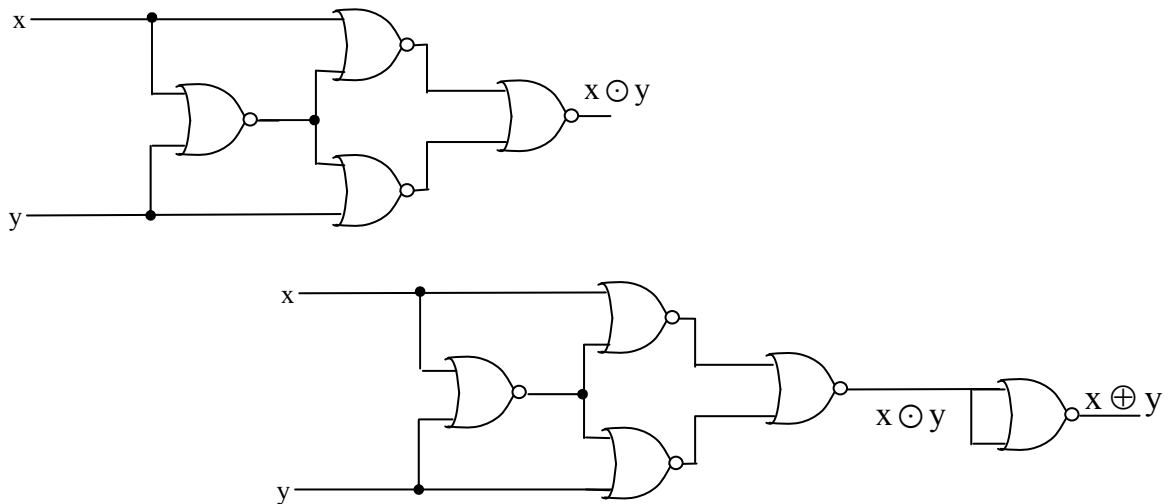
Σχήμα 3.5γ. Υλοποίηση της πύλης AND με πύλες NOR.



Σχήμα 3.5δ. Υλοποίηση πύλης NAND με πύλες NOR.

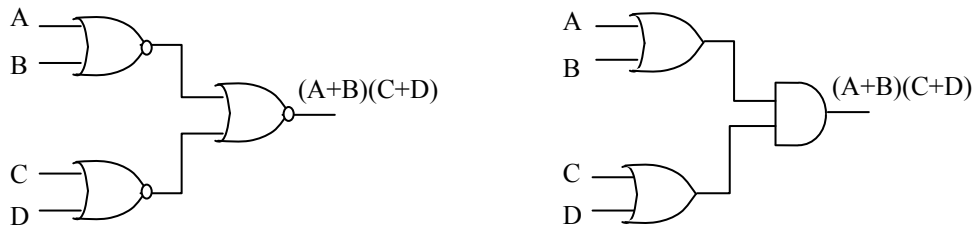
Η υλοποίηση της πύλης NAND είναι απλή υπόθεση, εφόσον έχουμε την υλοποίηση της AND και της NOT.

Τέλος, όσον αφορά στην υλοποίηση των 2 πυλών XNOR και XOR, με πύλες NOR, αυτές φαίνονται στα σχήματα 3.5ε και 3.5στ αντίστοιχα.



Σχήμα 3.5ε. Υλοποίηση πύλης XNOR και (στ) XOR με πύλες NOR.

Από το παραπάνω ισοδύναμο της πύλης NOR συμπεραίνουμε ότι ένα λογικό κύκλωμα δύο βαθμίδων με πύλες NOR – NOR μπορεί εύκολα να αντικατασταθεί από ένα κύκλωμα δύο βαθμίδων με πύλες OR - AND.

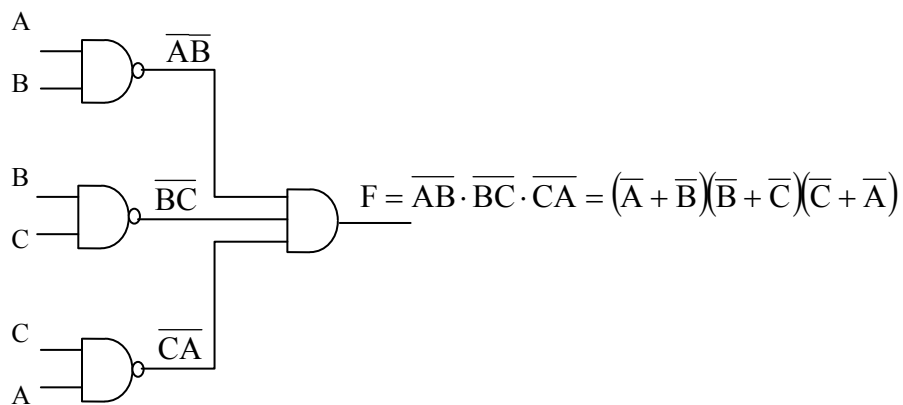


Σχήμα 3.5ζ. Ισοδύναμα κυκλώματα.

Παράδειγμα 3.8:

Δίνεται η λογική συνάρτηση $F = (\overline{A + B})(\overline{B + C})(\overline{C + A})$. Να σχεδιαστεί το λογικό κύκλωμα με πύλες NAND-AND.

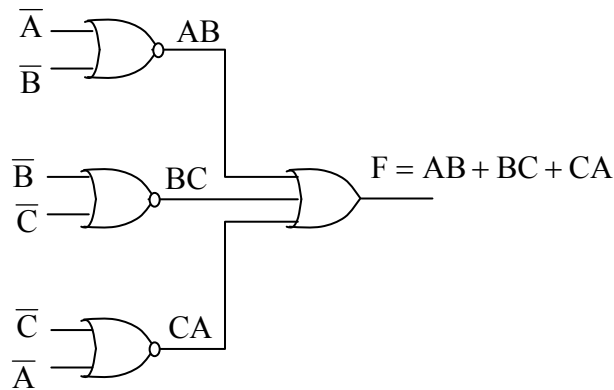
Λύση



Παράδειγμα 3.9:

Δίνεται η λογική συνάρτηση $F = AB + BC + CA$. Να σχεδιαστεί το λογικό κύκλωμα με πύλες NOR-OR.

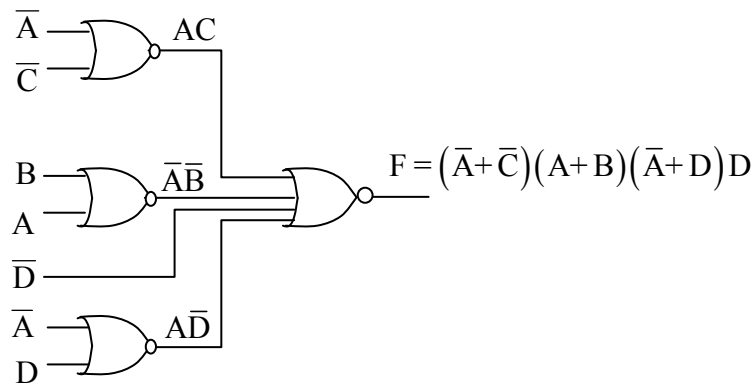
Λύση



Παράδειγμα 3.10:

Δίνεται η λογική συνάρτηση $F = (\bar{A} + \bar{C})(A + B)(\bar{A} + D) \cdot D$. Να σχεδιαστεί το λογικό κύκλωμα μόνο με πύλες NOR.

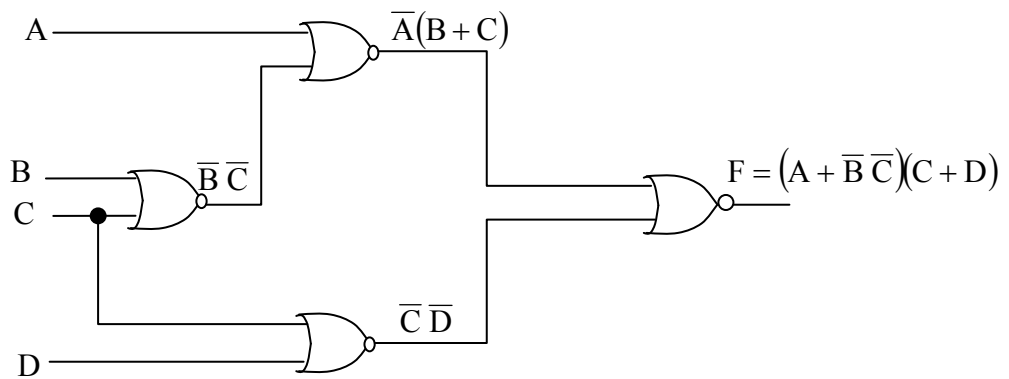
Λύση



Παράδειγμα 3.11:

Δίνεται η λογική συνάρτηση $F = (A + \bar{B}\bar{C})(C + D)$. Να σχεδιαστεί το λογικό κύκλωμα μόνο με πύλες NOR.

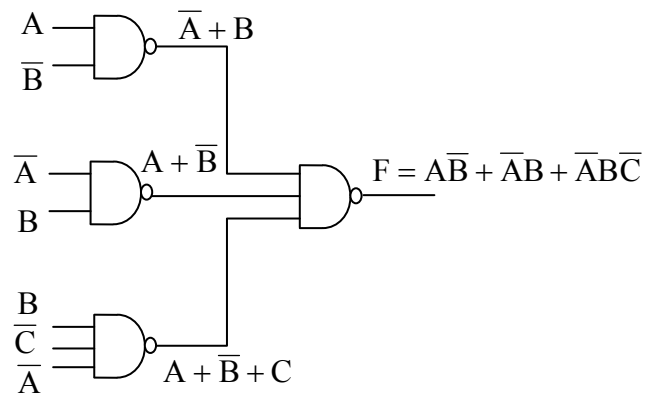
Λύση



Παράδειγμα 3.12:

Δίνεται η λογική συνάρτηση $F = \overline{A}B\overline{C} + A\overline{B} + \overline{A}B$. Να σχεδιαστεί το λογικό κύκλωμα μόνο με πύλες NAND.

Λύση



4. ΑΠΛΟΠΟΙΗΣΗ ΛΟΓΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΜΕ ΤΗ ΜΕΘΟΔΟ KARNAUGH

Η πολυπλοκότητα ψηφιακών πυλών που υλοποιούν μια συνάρτηση Boole σχετίζεται άμεσα με την πολυπλοκότητα της αλγεβρικής έκφρασης από την οποία υλοποιείται η συνάρτηση. Αν και η αναπαράσταση μιας συνάρτησης με πίνακα αληθείας είναι μοναδική, όπως ήδη είπαμε, η αλγεβρική της αναπαράσταση μπορεί να πάρει πολλές μορφές.

Οι συναρτήσεις Boole μπορούν να απλοποιηθούν με αλγεβρικούς τρόπους όπως είδαμε στο Κεφάλαιο 3. Όμως αυτός ο τρόπος απλοποίησης είναι δύσχρηστος, διότι δεν έχει συγκεκριμένους κανόνες που να μας καθοδηγούν κάθε φορά. Σε τούτη την παράγραφο θα μελετήσουμε μια απλή και συστηματική μέθοδο απλοποίησης λογικών συναρτήσεων η οποία ονομάζεται μέθοδος “χάρτη Karnaugh”.

Ο χάρτης Karnaugh είναι ένα διάγραμμα αποτελούμενο από τετράγωνα. Κάθε τετράγωνο παριστάνει έναν ελαχιστόρο (minterm). Έτσι, η μέθοδος ουσιαστικά συνίσταται στη χάραξη ενός ορθογωνικού διαγράμματος το οποίο περιέχει τόσα τετράγωνα όσα και οι δυνατοί συνδυασμοί των μεταβλητών (δηλαδή για n μεταβλητές θα περιέχει 2^n τετράγωνα). Επειδή κάθε συνάρτηση Boole μπορεί να εκφραστεί σαν άθροισμα ελαχιστόρων, έπεται ότι μια συνάρτηση Boole αναγνωρίζεται γραφικά στο χάρτη από την περιοχή που καλύπτουν τα τετράγωνα των ελαχιστόρων που περιέχονται στη συνάρτηση. Δηλαδή, ο χάρτης είναι ένα σχηματικό διάγραμμα όλων των δυνατών τρόπων με τους οποίους η συνάρτηση μπορεί να εκφραστεί σε πρότυπη μορφή. Αναγνωρίζοντας διάφορες σχηματικές μορφές, ο χρήστης μπορεί να δημιουργήσει εναλλακτικές αλγεβρικές παραστάσεις για την ίδια συνάρτηση, από τις οποίες μπορεί να διαλέξει την απλούστερη. Θα θεωρούμε στο εξής ότι η απλούστερη αλγεβρική έκφραση είναι οποιοδήποτε άθροισμα γινομένων ή γινόμενο αθροισμάτων έχει ελάχιστο αριθμό παραγόντων.

x y	0	1
0	0	1
1	2	3

Σχ. 4.1. Χάρτης Karnaugh για δύο μεταβλητές.

4.1. Χάραξη του χάρτη Karnaugh για δύο μεταβλητές

Στο Σχ. 4.1 φαίνεται το διάγραμμα χάρτη Karnaugh για δύο μεταβλητές x και y . Το διάγραμμα έχει 4 ($=2^2$) τετράγωνα. Η μεταβλητή x αποδίδεται στις στήλες και η μεταβλητή y στις γραμμές. Επικεφαλής κάθε στήλης και κάθε γραμμής υπάρχουν οι δυαδικές τιμές 0 και 1 και είναι οι τιμές που μπορούν να πάρουν οι μεταβλητές x και y . Στα κάτω δεξιό μέρος κάθε τετραγώνου γράφεται ένας αριθμός που είναι το

δεκαδικό αντίστοιχο του συνδυασμού των μεταβλητών ο οποίος αντιστοιχεί σε αυτό το τετράγωνο. Π.χ. τα $y = 1, x = 0$ αντιστοιχούν στον αριθμό $2 = (10)_2$. Μέσα σε κάθε τετράγωνο γράφουμε το 1 ή το 0 ανάλογα με την τιμή που παίρνει η συνάρτηση δηλαδή 1 ή 0 για τον αντίστοιχο συνδυασμό των δυαδικών μεταβλητών. Γενικά, συνηθίζεται να γράφονται μέσα στα τετράγωνα μόνον οι τιμές 1 αφήνοντας τα τετράγωνα όπου πρόκειται να γραφεί το 0 κενά.

yx \ z	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Σχ. 4.2. Χάρτης Karnaugh για τρεις μεταβλητές.

4.2. Χάραξη του χάρτη Karnaugh για τρεις μεταβλητές

Στο Σχ. 4.2 φαίνεται ο χάρτης Karnaugh για τρεις μεταβλητές, ο οποίος αποτελείται από $2^3 = 8$ τετράγωνα (4 στήλες και 2 γραμμές). Δύο μεταβλητές (οι x και y) αποδίδονται στις στήλες και η τρίτη μεταβλητή (η z) στις γραμμές. Οι τέσσερις επικεφαλίδες των στηλών είναι οι τέσσερις δυνατοί συνδυασμοί δύο μεταβλητών, δηλ. οι συνδυασμοί 00, 01, 10, και 11.

Η διάταξη των συνδυασμών είναι κατά τρόπον ώστε όταν περνάμε από τη μία στήλη στην άλλη να μεταβάλλεται μόνο μία δυαδική μεταβλητή δηλαδή $00 \rightarrow 01 \rightarrow 11 \rightarrow 10$. Δύο στήλες αυτού του είδους ονομάζονται *γειτονικές στήλες* και είναι αυτές που προκύπτουν δια μιας μόνο μεταβολής δυαδικής μεταβλητής. Κατ' αυτόν τον τρόπο μπορούν να θεωρηθούν ότι είναι γειτονικές και η πρώτη με την τελευταία στήλη, διότι γίνεται μία μόνο μεταβολή δυαδικής μεταβλητής όταν περνάμε από τη μία στήλη στην άλλη, εν προκειμένω μάλιστα μεταβάλλεται η μεταβλητή y (από 0 σε 1).

Σε ό,τι αφορά την τοποθέτηση των αριθμών στο κάτω δεξιά μέρος κάθε τετραγώνου καθώς επίσης και για τη συμπλήρωση των τετραγώνων με 1 ή 0 ισχύουν όσα είπαμε στην προηγούμενη παράγραφο.

4.3. Χάραξη του χάρτη Karnaugh για τέσσερις μεταβλητές

Στο Σχ. 4.3 φαίνεται ο χάρτης Karnaugh για τέσσερις μεταβλητές, ο οποίος αποτελείται από $2^4 = 16$ τετράγωνα (4 στήλες και 4 γραμμές). Δύο μεταβλητές (οι x και y) αποδίδονται στις στήλες και δύο μεταβλητές (οι z και w) στις γραμμές. Οι τέσσερις επικεφαλίδες των στηλών και των γραμμών είναι οι τέσσερις δυνατοί συνδυασμοί δύο μεταβλητών, δηλ. οι συνδυασμοί 00, 01, 10, και 11.

Η διάταξη των στηλών και των γραμμών είναι και εδώ όπως η του Σχ. 4.2. Ισχύουν λοιπόν όσα προαναφέραμε για τις γειτονικές στήλες. Στις στήλες αυτές η διάταξη των διαφόρων συνδυασμών των μεταβλητών ακολουθεί τον κανόνα της μεταβολής μιας

μόνο δυαδικής μεταβλητής όταν περνάμε από τη μία στήλη στην άλλη. Εφόσον η διάταξη των συνδυασμών των μεταβλητών στις γραμμές είναι η ίδια με τη διάταξη των στηλών ισχύει και εδώ ότι κάθε γραμμή είναι γειτονική με τη διπλανή της όπως και η πρώτη με την τελευταία.

y x w z	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Σχ. 4.3. Χάρτης Karnaugh για τέσσερις μεταβλητές.

4.4. Χάραξη του χάρτη Karnaugh για πέντε μεταβλητές

Το αντίστοιχο διάγραμμα περιέχει $2^5 = 32$ τετράγωνα και ως εκ τούτου χρησιμοποιείται πολύ δύσκολα. Πρακτικά είναι δυνατό να διασπασθεί σε δύο διαγράμματα των 16 τετραγώνων. Στην περίπτωση αυτή ένα από τα διαγράμματα αντιστοιχεί στην τιμή 0 και το άλλο στην τιμή 1 της πέμπτης δυαδικής μεταβλητής.

4.5. Απλοποίηση με τη μέθοδο Karnaugh

Η μέθοδος Karnaugh συνίσταται στη χρήση του Αξιώματος 5(α) $x + \bar{x} = 1$, προς απλοποίηση όρων ανά δύο σύμφωνα με το ακόλουθο σκεπτικό:

α) Δύο όροι που διαφέρουν μεταξύ τους μόνο κατά την τιμή μιας μεταβλητής, οι οποίοι δηλαδή είναι γειτονικοί στο χάρτη Karnaugh, απλοποιούνται διότι

$$F \cdot x + F \cdot \bar{x} = F$$

Παράδειγμα: $A \cdot B \cdot C + A \cdot B \cdot \bar{C} = A \cdot B$ όπου $(F = A \cdot B \text{ και } C = x)$

β) Τέσσερις όροι στους οποίους οι δύο μεταβλητές παίρνουν και τους τέσσερις δυνατούς συνδυασμούς, ενώ οι υπόλοιπες μεταβλητές διατηρούν την ίδια πάντοτε τιμή απλοποιούνται εξαφανίζοντας τις δύο πρώτες μεταβλητές.

Παράδειγμα: $\bar{A} \cdot B \cdot C + \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot \bar{B} \cdot \bar{C} = \bar{A}$

Παρατηρούμε ότι οι τέσσερις αυτοί όροι αντιστοιχούν στο χάρτη Karnaugh σε τέσσερα τετράγωνα τα οποία σχηματίζουν ένα ευρύτερο τετράγωνο ή μία γραμμή, ανάλογα με τη διάταξη των μεταβλητών A, B, C, πάνω στο χάρτη.

Είναι πολλές φορές δυνατό να παρουσιαστούν όροι συγκεντρωμένοι ανά 8, δηλαδή τα 8 τετράγωνα που αντιστοιχούν σ' αυτούς τους όρους να σχηματίζουν ορθογώνιο. Και στην περίπτωση αυτή επιτρέπεται να απλοποιηθούν τρεις μεταβλητές. Σκοπός μας είναι να απλοποιηθούν όσο το δυνατόν περισσότερες μεταβλητές. Η γενική μέθοδος για να το πετύχουμε συνίσταται στα εξής:

- 1) Αναζητούνται πάνω στο χάρτη Karnaugh όλα τα 1 τα οποία δεν γειτονεύουν με κανένα άλλο, απομονώνονται και γράφεται η έκφραση των αντίστοιχων όρων της συνάρτησης.
- 2) Αναζητούνται στη συνέχεια ομάδες τετραγώνων τα οποία είναι γειτονικά μεταξύ τους. Ο αριθμός των τετραγώνων τα οποία αποτελούν την κάθε ομάδα πρέπει να είναι 2^n όπου n ακέραιος, το δε σχήμα το οποίο σχηματίζουν τα τετράγωνα της ομάδας πάνω στο χάρτη Karnaugh είναι ευθεία γραμμή ή τετράγωνο ή ορθογώνιο. Από 2^n τα οποία έχουν την ιδιότητα αυτή απλοποιούνται n μεταβλητές. Κατ' αυτόν τον τρόπο από 2 ή 4 ή 8 τετράγωνα απλοποιούνται αντίστοιχα 1 ή 2 ή 3 μεταβλητές. Προκειμένου να αποφασιστεί ποια μεταβλητή απλοποιείται, αναζητείται η μεταβλητή που μεταβάλλεται όταν περνάμε, κάθετα ή οριζόντια, από το ένα τετράγωνο της ομάδας στο άλλο. Οι υπόλοιπες μεταβλητές αποτελούν την έκφραση της ομάδας των 2 ή 4 ή 8 όρων. Ας σημειωθεί ότι κάθε τετράγωνο μπορεί να ανήκει συγχρόνως σε πολλές ομάδες, διότι ισχύει το Θεώρημα 1(α)

$$A + A + A + \dots + A = A$$

Είναι συμφέρον εξάλλου, να αναζητούνται ομάδες όσο το δυνατό περισσότερων τετραγώνων διότι τότε επιτυγχάνεται απλοποίηση περισσότερων μεταβλητών. Για την αναγραφή της αλγεβρικής έκφρασης του κάθε όρου της συνάρτησης ακολουθείται ο εξής κανόνας: Εφόσον η μεταβλητή x έχει την τιμή 1 γράφουμε x , εφόσον δε έχει την τιμή 0 γράφουμε \bar{x} .

Στη συνέχεια, θα μελετήσουμε τη μέθοδο απλοποίησης με χάρτη Karnaugh χρησιμοποιώντας όσα προαναφέραμε σε μερικά χαρακτηριστικά παραδείγματα.

4.6. Απλοποίηση συνάρτησης δύο μεταβλητών

Δίνεται η συνάρτηση

$$F = \bar{x}\bar{y} + \bar{y}x + y\bar{x}$$

Να απλοποιηθεί.

Ο αντίστοιχος χάρτης Karnaugh δίνεται στο Σχ. 4.4. Γράφουμε από ένα 1 στα τετράγωνα 0, 1, 2, καθόσον από την αλγεβρική έκφραση της συνάρτησης προκύπτει ότι οι συνδυασμοί των δυαδικών μεταβλητών x, y για τους οποίους αληθεύει (δηλ. παίρνει την τιμή 1) η συνάρτηση, αντιστοιχούν σε αυτά ακριβώς τα τετράγωνα.

Το ίδιο είναι δυνατό να γίνει αυτόματα, με σύγκριση των δεκαδικών αντιστοιχών εκείνων των συνδυασμών για τους οποίους $F = 1$, αφενός μεν προς αυτά που προκύπτουν από την εικόνα της συνάρτησης στο χάρτη, αφετέρου δε προς αυτά τα οποία αναγράφονται στο κάτω δεξιό μέρος του κάθε τετραγώνου του χάρτη.

Γράφεται έτσι ένα 1 σε κάθε τετράγωνο που έχει δεκαδικό αντίστοιχο όμοιο προς κάθε δεκαδικό αντίστοιχο της εικόνας της συνάρτησης. Παρατηρούμε έτσι δύο ομάδες που η καθεμιά έχει δύο τετράγωνα. Η αλγεβρική έκφραση της ομάδας των δύο οριζόντιων τετραγώνων είναι \bar{y} και των δύο κατακόρυφων είναι \bar{x} , άρα είναι

$$F = \bar{x} + \bar{y}$$

x y	0	1
0	1	1
1	1	0

0 1
2 3

Σχ. 4.4. Χάρτης Karnaugh συνάρτησης F δύο μεταβλητών.

Παρατηρούμε ότι το τετράγωνο με δεκαδικό αντίστοιχο το 0 ανήκει και χρησιμοποιήθηκε και στις δύο ομάδες. Αυτό έγινε για να απλοποιηθεί όσο το δυνατόν περισσότερο η έκφραση της συνάρτησης. Εάν τα τετράγωνα με δεκαδικό αντίστοιχο το 2 ή το 1 είχαν παρθεί χωριστά θα προέκυπταν αντίστοιχα οι εξής εκφράσεις της συνάρτησης

$$F = \bar{y} + y\bar{x}$$

$$F = \bar{x} + \bar{y}x$$

4.7. Απλοποίηση συνάρτησης τριών μεταβλητών

Δίνεται η συνάρτηση

$$F = \bar{z} \bar{y} \bar{x} + \bar{z} y \bar{x} + \bar{z} \bar{y} x + z \bar{y} \bar{x} + z \bar{y} x$$

Να απλοποιηθεί.

Ο χάρτης Karnaugh που αντιστοιχεί στη συνάρτηση F τριών μεταβλητών δίνεται στο Σχ. 4.5.

y x z	00	01	11	10
0	1	1		1
1	1	1		

0 1 3 2
4 5 7 6

Σχ. 4.5. Χάρτης Karnaugh συνάρτησης F τριών μεταβλητών.

Στα τετράγωνα του χάρτη Karnaugh γράφονται τα 1 όπως και στην προηγούμενη περίπτωση δηλαδή είτε από την αλγεβρική έκφραση της συνάρτησης είτε από την εικόνα της συνάρτησης στο χάρτη. Στην παρούσα περίπτωση τα πέντε τετράγωνα μέσα στα οποία υπάρχει το 1 σχηματίζουν δύο ομάδες. Μία ομάδα των τεσσάρων τετραγώνων και μία ομάδα των δύο τετραγώνων. Από την ομάδα των τεσσάρων

απλοποιούνται δύο μεταβλητές ενώ από την ομάδα των δύο απλοποιείται μία μεταβλητή.

Η ομάδα των τεσσάρων τετραγώνων αποτελείται από τα τετράγωνα με δεκαδικό αντίστοιχο 0, 1, 4, 5. Για να αναγνωρισθούν οι μεταβλητές που απλοποιούνται, παρατηρούνται οι μεταβλητές που αλλάζουν τιμή κατά την οριζόντια μετακίνηση από το τετράγωνο 0 προς το 1 ή από το 4 προς το 5. Η μεταβλητή αυτή είναι η x ($0 \rightarrow 1$) ενώ η y διατηρεί την τιμή $y = 0$. Άρα για την έκφραση της συνάρτησης αρκεί το \bar{y} . Η ίδια παρατήρηση γίνεται και κατά την κατακόρυφη μετακίνηση δηλαδή από το 0 στο 4 ή από το 1 στο 5. Τότε μεταβάλλεται η z ($0 \rightarrow 1$). Άρα η z απλοποιείται και η έκφραση της ομάδας των τεσσάρων τετραγώνων είναι \bar{y} . Κατ' αυτόν τον τρόπο απλοποιήθηκαν δύο μεταβλητές οι x, z .

Για την ομάδα των δύο τετραγώνων με τα δεκαδικά αντίστοιχα 0 και 2 είναι γνωστό ότι απλοποιείται μόνο μία μεταβλητή. Παρατηρούνται και πάλι οι μεταβλητές που αλλάζουν κατά την οριζόντια μετακίνηση από το 0 στο 2. Η μεταβλητή αυτή είναι μία, δηλαδή η y ($0 \rightarrow 1$). Άρα απλοποιείται η y . Η αλγεβρική έκφραση της ομάδας θα περιέχει τη μεταβλητή x η οποία διατηρεί την ίδια τιμή για τα τετράγωνα 0 και 2 και την z η οποία διατηρεί επίσης την ίδια τιμή για τα ίδια τετράγωνα. Επειδή είναι $x = 0$ και $z = 0$, η έκφραση της ομάδας θα είναι $\bar{x}\bar{z}$.

Άρα η τελική έκφραση της συνάρτησης θα είναι

$$F = \bar{y} + \bar{z}\bar{x}$$

4.8. Απλοποίηση συνάρτησης τεσσάρων μεταβλητών

Δίνεται η συνάρτηση

$$F = w\bar{y}\bar{x} + \bar{z}y\bar{x} + zy\bar{x} + w\bar{z}\bar{y}x + \bar{w}z\bar{y}x$$

Να απλοποιηθεί.

Ο χάρτης Karnaugh που αντιστοιχεί στη συνάρτηση F τεσσάρων μεταβλητών δίνεται στο Σχ. 4.6.

Στα τετράγωνα του χάρτη Karnaugh γράφονται τα 1 όπως και στην προηγούμενη περίπτωση δηλαδή είτε από την αλγεβρική έκφραση της συνάρτησης είτε από την εικόνα της συνάρτησης στο χάρτη. Στην παρούσα περίπτωση τα οκτώ τετράγωνα μέσα στα οποία υπάρχει το 1 σχηματίζουν τέσσερις ομάδες. Δύο ομάδες των τεσσάρων τετραγώνων, μία ομάδα των δύο τετραγώνων, και μία ομάδα ενός μεμονωμένου τετραγώνου. Από τις ομάδες των τεσσάρων απλοποιούνται δύο μεταβλητές ενώ από την ομάδα των δύο απλοποιείται μία μεταβλητή. Το απομονωμένο τετράγωνο δεν συνδυάζεται με κάποιο γειτονικό του και επομένως θα δώσει έναν όρο τεσσάρων μεταβλητών, τον $\bar{w}z\bar{y}x$.

Η πρώτη ομάδα των τεσσάρων τετραγώνων αποτελείται από τα τετράγωνα με δεκαδικό αντίστοιχο 14, 12, 8, 10. Για να αναγνωρισθούν οι μεταβλητές που απλοποιούνται, παρατηρούνται οι μεταβλητές που αλλάζουν τιμή κατά την οριζόντια μετακίνηση από το τετράγωνο 12 προς το 14 ή από το 8 προς το 10. Η μεταβλητή αυτή είναι η y ($0 \rightarrow 1$) ενώ η x διατηρεί την τιμή $x = 0$. Άρα για την έκφραση της συνάρτησης αρκεί το \bar{x} . Η ίδια παρατήρηση γίνεται και κατά την κατακόρυφη μετακίνηση δηλαδή από το 12 στο 8 ή από το 14 στο 10. Τότε μεταβάλλεται η z ($1 \rightarrow 0$). Άρα η z απλοποιείται και η έκφραση της ομάδας των τεσσάρων τετραγώνων είναι $w\bar{x}$. Κατ' αυτόν τον τρόπο απλοποιήθηκαν δύο μεταβλητές οι y, z .

$y x$ $w z$	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Σχ. 4.6. Χάρτης Karnaugh συνάρτησης F τεσσάρων μεταβλητών.

Η δεύτερη ομάδα των τεσσάρων τετραγώνων αποτελείται από τα τετράγωνα με δεκαδικό αντίστοιχο 2, 6, 14, 10. Στην ομάδα αυτή παρατηρείται μεταβολή μόνο κατά την κατακόρυφη διεύθυνση και μάλιστα και των δύο μεταβλητών w και z οι οποίες συνεπώς απλοποιούνται. Στην έκφραση λοιπόν της ομάδας των τεσσάρων τετραγώνων απομένουν οι μεταβλητές $y\bar{x}$.

Για την ομάδα των δύο τετραγώνων με τα δεκαδικά αντίστοιχα 8 και 9 είναι γνωστό ότι απλοποιείται μόνο μία μεταβλητή. Παρατηρούνται και πάλι οι μεταβλητές που αλλάζουν κατά την οριζόντια μετακίνηση από το 8 στο 9. Η μεταβλητή αυτή είναι μία, δηλαδή η x ($0 \rightarrow 1$). Άρα απλοποιείται η x. Η αλγεβρική έκφραση της ομάδας θα περιέχει τη μεταβλητή y η οποία διατηρεί την ίδια τιμή για τα τετράγωνα 8 και 9 και τις z, w οι οποίες διατηρούν επίσης την ίδια τιμή για τα ίδια τετράγωνα. Επειδή είναι $y = 0$, $z = 0$, και $w = 1$, η έκφραση της ομάδας θα είναι $wz\bar{y}$.

Άρα η τελική έκφραση της συνάρτησης θα είναι

$$F = w\bar{x} + y\bar{x} + w\bar{z}\bar{y} + \bar{w}z\bar{y}x$$

4.9. Αδιάφοροι όροι (Don't care)

Πολλές φορές υπάρχουν σε μια συνάρτηση πλεονάζοντες όροι, των οποίων η τιμή δεν μας ενδιαφέρει αν θα είναι 1 ή 0. Οι όροι αυτοί ονομάζονται αδιάφοροι όροι. Οι αδιάφοροι όροι παριστάνονται με το σύμβολο X. Παράδειγμα αποτελεί η περίπτωση ενός δεκαδικού απαριθμητή με 4 bits, ο οποίος μπορεί να βρεθεί σε $2^4=16$ διαφορετικές καταστάσεις, από τις οποίες μόνο οι 10 μας ενδιαφέρουν. Οι άλλες 6 καταστάσεις είναι αδιάφοροι όροι.

Παρ' όλο που η τιμή των αδιάφορων όρων δεν έχει σημασία για τη λογική συνάρτηση, ωστόσο μπορούν να χρησιμοποιηθούν στην απλοποίηση της λογικής συνάρτησης. Γι' αυτό τους δίνουμε άλλοτε την τιμή 0 και άλλοτε την τιμή 1, έτσι ώστε να σχηματίζουμε όσο το δυνατόν λιγότερες και συγχρόνως μεγαλύτερες ομάδες γειτονικών τετραγωνιδίων στον πίνακα Karnaugh.

Παράδειγμα: Να απλοποιηθεί η συνάρτηση της οποίας δίνεται παρακάτω ο πίνακας αληθείας.

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

C \ AB	00	01	11	10
0	X	0	1	0
1	1	1	X	1

Από τον πίνακα Karnaugh προκύπτει η απλοποιημένη συνάρτηση: $F = C + AB$.

Παρατηρήσεις:

- (1) Σε κάθε ομάδα που σχηματίζουμε πρέπει να υπάρχει τουλάχιστον ένας μη αδιάφορος όρος (1 ή 0) ανάλογα με το αν ομαδοποιώ 1 ή 0. Δηλαδή δεν επιτρέπεται να σχηματίσουμε ομάδα που να περιέχει μόνο αδιάφορους όρους.
- (2) Δίνω την τιμή 1 ή 0 στους αδιάφορους όρους που ομαδοποιώ και μόνο σε αυτούς.

4.10. Παραδείγματα

Πρέπει να σημειωθεί ότι η απλοποίηση με τη μέθοδο του χάρτη Karnaugh δεν οδηγεί υποχρεωτικά σε μία και μόνη λύση. Είναι δυνατό να επιτευχθούν πολλές λύσεις ισοδύναμες μεταξύ τους.

Παράδειγμα 4.1:

z \ y x	00	01	11	10
0	1	1		1
1		1		1

z \ y x	00	01	11	10
0	1	1		1
1		1		1

Σχ. 4.7. Χάρτης Karnaugh συνάρτησης F τριών μεταβλητών και δύο διαφορετικές απλοποιήσεις της.

Στη συνάρτηση του πιο πάνω σχήματος μπορούν να απομονωθούν τρεις ομάδες των δύο τετραγώνων αλλά κατά δύο διαφορετικούς τρόπους όπως φαίνεται. Ο κάθε ένας τρόπος συγκέντρωσης των τετραγώνων σε ομάδες δίνει και διαφορετική λύση.

α) Το Σχ. 4.7α οδηγεί στην έκφραση

$$F = \bar{z} \bar{y} + \bar{y}x + y\bar{x}$$

β) Το Σχ. 4.7β οδηγεί στην έκφραση

$$F = \bar{z} \bar{x} + \bar{y}x + y\bar{x}$$

Οι δύο αυτές εκφράσεις είναι ισοδύναμες.

2) Αντί να θεωρούμε τα τετράγωνα για τα οποία είναι $F = 1$, μπορούμε να θεωρούμε εκείνα για τα οποία είναι $F = 0$ και να υπολογίζεται η έκφραση της \bar{F} .

Παράδειγμα 4.2:

Από το Σχ. 4.7 προκύπτει

$$\bar{F} = z \bar{y} \bar{x} + yx$$

Άρα είναι και

$$F = \bar{\bar{F}} = \overline{z \bar{y} \bar{x} + yx} = \overline{(yx)} \cdot \overline{(z \bar{y} \bar{x})} = (\bar{y} + \bar{x}) \cdot (\bar{z} + y + x)$$

3) Με υπολογισμό της έκφρασης της συνάρτησης από το χάρτη Karnaugh με την ομαδοποίηση των 1 διευκολύνεται πολύ η κατασκευή του αντίστοιχου κυκλώματος αποκλειστικά με πύλες NAND.

Με υπολογισμό της έκφρασης της συνάρτησης από το χάρτη Karnaugh με την ομαδοποίηση των 0 διευκολύνεται πολύ η κατασκευή του αντίστοιχου κυκλώματος αποκλειστικά με πύλες NOR.

Παράδειγμα 4.3:

Δίνεται ο πίνακας αληθείας μιας συνάρτησης. Να βρεθεί η απλοποιημένη συνάρτηση με τη βοήθεια του χάρτη Karnaugh.

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

Από τον πίνακα αληθείας εξάγεται η λογική συνάρτηση:

$$F = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} \bar{B} \bar{C} D + \bar{A} \bar{B} C \bar{D} + \bar{A} B C D + A \bar{B} \bar{C} \bar{D} + A \bar{B} \bar{C} D + A \bar{B} C \bar{D} + A B C D = \sum(0,1,2,7,8,9,10,15)$$

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

Άρα η απλοποιημένη συνάρτηση είναι: $F = BCD + \bar{B}\bar{C} + \bar{B}\bar{D}$.

Παράδειγμα 4.4:

Δίνεται η ακόλουθη συνάρτηση ως γινόμενο μεγιστόρων $F = \prod(0,3,4,5,6,7,11,13,14,15)$. Να βρεθεί η απλοποιημένη συνάρτηση με τη βοήθεια του χάρτη Karnaugh.

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

Ομαδοποιώντας τους μηδενικούς όρους βρίσκουμε την απλοποιημένη συνάρτηση $F = (\bar{C} + \bar{D})(\bar{B} + \bar{D})(\bar{B} + \bar{C})(A + C + D)$.

Παράδειγμα 4.5:

Δίνεται η ακόλουθη συνάρτηση ως άθροισμα ελαχιστόρων $F = \sum(0,1,3,6,9,11,12,13,15)$. Να βρεθεί η απλοποιημένη συνάρτηση με τη βοήθεια του χάρτη Karnaugh.

Λύση

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

Ομαδοποιώντας τους όρους με 1, βρίσκουμε την απλοποιημένη συνάρτηση $F = \overline{A} \overline{B} \overline{C} + \overline{B} D + AD + ABC\overline{C} + \overline{A} B C \overline{D}$.

Παράδειγμα 4.6:

Από τον παρακάτω πίνακα Karnaugh να εξαχθούν δύο συναρτήσεις: Η F_1 ως άθροισμα γινομένων και η F_2 ως γινόμενο αθροισμάτων.

Λύση

AB \ CD	00	01	11	10
00	0	1	1	X
01	0	0	0	0
11	X	X	X	0
10	0	1	X	1

$$F_1 = B\overline{D} + A\overline{D}$$

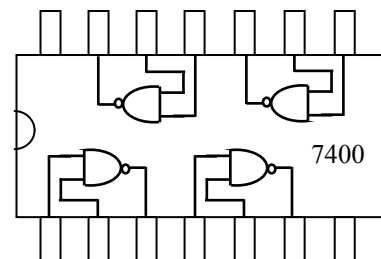
$$F_2 = \overline{D}(A + B)$$

Παράδειγμα 4.7:

Μία μικρή εταιρεία έχει 14 μετοχές και για κάθε μετοχή δίνει μία ψήφο σ' αυτόν που την έχει. Αυτές τις μετοχές τις έχουν 4 άτομα ως εξής:

- Ο κ. x έχει 1 μετοχή
- Ο κ. y έχει 6 μετοχές
- Η κ. z έχει 5 μετοχές και
- Ο κ. w έχει 2 μετοχές

Καθένας από αυτούς τους τέσσερις έχει έναν διακόπτη που τον κλείνει (βραχυκύκλωμα) για να ψηφίσει ΝΑΙ ή τον ανοίγει για να ψηφίσει ΟΧΙ. Θέλουμε να σχεδιάσουμε κύκλωμα που να δείχνει αν το αποτέλεσμα μιας ψηφοφορίας είναι ΝΑΙ ή ΟΧΙ. Στην περίπτωση ισοψηφίας το αποτέλεσμα να είναι ΝΑΙ.



Έχετε στη διάθεσή σας δύο ολοκληρωμένα 7400 των οποίων η διάταξη φαίνεται στο παραπάνω σχήμα και μία πύλη NAND.

ΥΠΟΔΕΙΞΗ: Να γίνει απλοποίηση με χρήση του χάρτη Karnaugh.

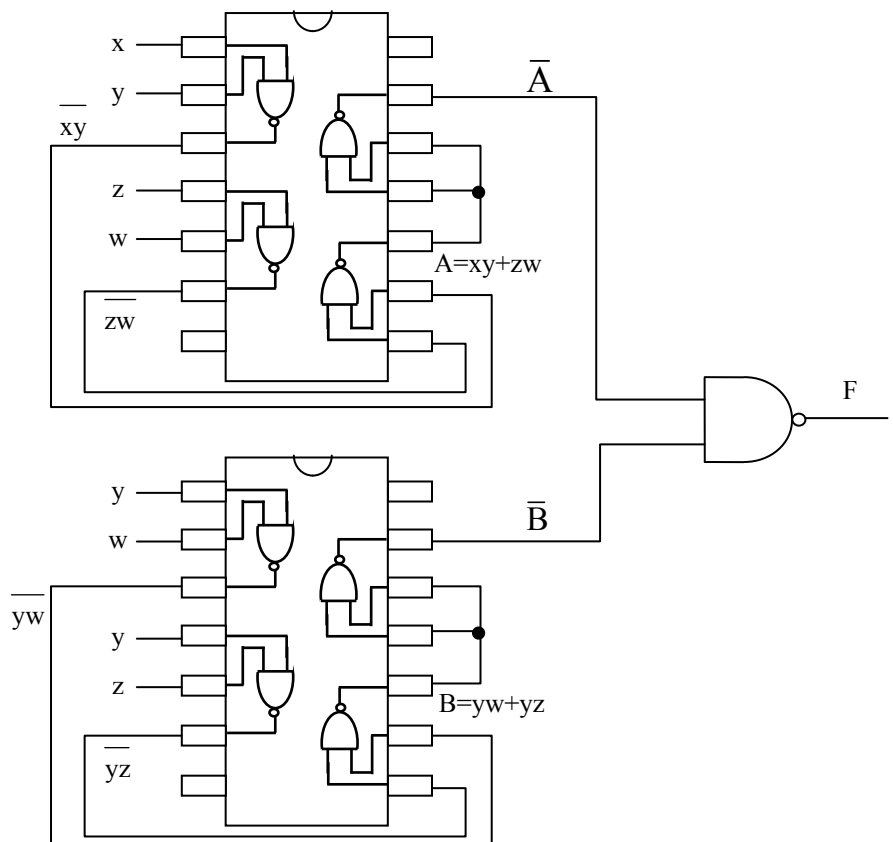
Λύση

Καταστρώνουμε τον πίνακα αληθείας με βάση τα δεδομένα του παραδείγματος. Στη συνέχεια απλοποιούμε με τη βοήθεια του χάρτη Karnaugh.

1	6	5	2	
x	y	z	w	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

xy \ zw	00	01	11	10
00			1	
01		1	1	1
11	1	1	1	1
10			1	

Η απλοποιημένη συνάρτηση είναι: $F = xy + zw + yw + yz$.



5. ΣΥΝΔΥΑΣΤΙΚΑ ΚΥΚΛΩΜΑΤΑ

Τα λογικά κυκλώματα των ψηφιακών συστημάτων μπορούν να είναι “συνδυαστικά” (combinational) ή “ακολουθιακά” (sequential). Ένα συνδυαστικό κύκλωμα αποτελείται από λογικές πύλες των οποίων οι έξοδοι, ανά πάσα στιγμή, καθορίζονται κατευθείαν από τις εισόδους εκείνης της στιγμής και δεν εξαρτώνται καθόλου από προηγούμενες τιμές των εισόδων. Ένα συνδυαστικό κύκλωμα εκτελεί μία συγκεκριμένη επεξεργασία πληροφοριών, που καθορίζεται επακριβώς από κάποιες συναρτήσεις Boole. Τα ακολουθιακά κυκλώματα χρησιμοποιούν και στοιχεία μνήμης (δυναμικά κύτταρα) επιπλέον των λογικών πυλών. Οι έξοδοί τους είναι συναρτήσεις τόσο των εισόδων όσο και της κατάστασης των στοιχείων μνήμης. Η κατάσταση των στοιχείων μνήμης, με τη σειρά της, είναι συνάρτηση των προηγούμενων εισόδων. Σαν αποτέλεσμα, οι έξοδοι ενός ακολουθιακού κυκλώματος εξαρτώνται όχι μόνον από τις τωρινές εισόδους, αλλά και από προηγούμενες εισόδους και η συμπεριφορά του κυκλώματος πρέπει να καθοριστεί σε σχέση με μια χρονική ακολουθία εισόδων και εσωτερικών καταστάσεων. Τα ακολουθιακά κυκλώματα θα τα μελετήσουμε αργότερα.

Ο σχεδιασμός συνδυαστικών κυκλωμάτων αρχίζει με μια προφορική διατύπωση του προβλήματος και τελειώνει με ένα λογικό διάγραμμα του κυκλώματος ή με κάποιες συναρτήσεις Boole από τις οποίες μπορεί να βγει το λογικό διάγραμμα. Η διαδικασία περιλαμβάνει τα εξής βήματα:

1. Καθορίζεται το πρόβλημα.
2. Καθορίζεται ο αριθμός των διαθέσιμων μεταβλητών εισόδου και των απαιτούμενων μεταβλητών εξόδου.
3. Διαλέγουμε συμβολικά ονόματα για τις μεταβλητές εισόδου και εξόδου.
4. Βρίσκουμε τον πίνακα αληθείας που καθορίζει τις απαιτούμενες σχέσεις μεταξύ εισόδων και εξόδων.
5. Απλοποιούμε τη συνάρτηση Boole για κάθε έξοδο.
6. Σχεδιάζουμε το λογικό διάγραμμα.

Ένας πίνακας αληθείς για ένα συνδυαστικό κύκλωμα αποτελείται από στήλες εισόδου και στήλες εξόδου. Τα 1 και τα 0 στις στήλες εισόδου τα παίρνουμε από τους 2^n δυαδικούς συνδυασμούς των n μεταβλητών εισόδου. Οι δυαδικές τιμές των εξόδων καθορίζονται από το συγκεκριμένο πρόβλημα. Μία έξοδος μπορεί να είναι είτε 0 είτε 1 για κάθε αποδεκτό συνδυασμό εισόδων. Οι συναρτήσεις εξόδου, που προσδιορίζονται από τον πίνακα αληθείας, δίνουν τον ακριβή ορισμό του συνδυαστικού κυκλώματος. Οι συναρτήσεις Boole των εξόδων απλοποιούνται με οποιαδήποτε διαθέσιμη μέθοδο από τον πίνακα αληθείας π.χ. με αλγεβρικές πράξεις, με τη μέθοδο του χάρτη ή με τη μέθοδο κατάταξης σε πίνακα. Σε κάθε συγκεκριμένη εφαρμογή, διάφοροι περιορισμοί και κριτήρια χρησιμεύουν σαν οδηγός σ’ αυτή την επιλογή της συγκεκριμένης αλγεβρικής έκφρασης. Στην πράξη, οι σχεδιαστές τείνουν να πηγαίνουν από τις συναρτήσεις Boole σ’ έναν κατάλογο συνδεσμολογίας που δείχνει τις συνδέσεις ανάμεσα σε διάφορες πύλες κάποιας λογικής οικογένειας. Σ’ αυτή την περίπτωση, ο σχεδιασμός δεν χρειάζεται να προχωρήσει πιο πέρα από τις απαραίτητες απλοποιημένες συναρτήσεις Boole των εξόδων. Πάντως, ένα λογικό διάγραμμα είναι χρήσιμο για να σχηματίσει κανείς μια καθαρή εικόνα στο μυαλό του για την υλοποίηση των εκφράσεων με πύλες.

Στη συνέχεια τούτης της παραγράφου, θα μελετήσουμε τα σημαντικότερα συνδυαστικά κυκλώματα που χρησιμοποιούνται στην πράξη για την υλοποίηση των λογικών συναρτήσεων.

5.1. Αθροιστές (Adders)

Οι ψηφιακοί υπολογιστές εκτελούν μια ποικιλία από επεξεργασίες πληροφοριών. Ανάμεσα στις βασικές συναρτήσεις που απαντώνται, είναι οι διάφορες αριθμητικές πράξεις. Στην ουσία δύο μόνον πράξεις αρκούν, η πρόσθεση και η αφαίρεση, επειδή ο μεν πολλαπλασιασμός εκτελείται με επανειλημμένες προσθέσεις η δε διαίρεση με επανειλημμένες αφαιρέσεις. Η πιο βασική αριθμητική πράξη είναι η πρόσθεση δύο δυαδικών ψηφίων. Αυτή η απλή πρόσθεση έχει τέσσερις δυνατές περιπτώσεις: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = 10$. Οι τρεις πρώτες δημιουργούν ένα άθροισμα που το μήκος του είναι ένα ψηφίο, αλλά όταν και οι δύο προσθετέοι είναι 1, το δυαδικό άθροισμα αποτελείται από δύο ψηφία. Το πιο σημαντικό από αυτά τα δύο bits το λέμε “κρατούμενο” (carry). Όταν οι προσθετέοι περιέχουν κι άλλα σημαντικά ψηφία, το κρατούμενο που προκύπτει από την πρόσθεση δύο bits προστίθεται στο επόμενο μεγαλύτερης σημαντικότητας ζευγάρι bits. Ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση δύο bits λέγεται “ημιαθροιστής” (half-adder). Ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση τριών bits (δύο σημαντικών bits κι ενός προηγούμενου κρατούμενου) λέγεται “πλήρης αθροιστής” (full-adder). Το όνομα του πρώτου προέρχεται από το γεγονός ότι δύο ημιαθροιστές μπορούν να χρησιμοποιηθούν για να υλοποιήσουν έναν πλήρη αθροιστή. Τα δύο αθροιστικά κυκλώματα είναι τα πρώτα που θα μελετήσουμε.

5.1.1. Ημιαθροιστής (Half-Adder)

Το συνδυαστικό κύκλωμα του ημιαθροιστή απαιτεί δύο δυαδικές εισόδους και δύο δυαδικές εξόδους. Οι μεταβλητές εισόδου είναι οι προσθετέοι, ενώ οι μεταβλητές εξόδου είναι το άθροισμα και το κρατούμενο. Συμβολίζουμε αυθαίρετα με x και y τις δύο μεταβλητές εισόδου και με S (Sum - άθροισμα) και C (Carry - κρατούμενο) τις δύο μεταβλητές εξόδου. Δίνουμε πιο κάτω τον πίνακα αληθείας της δυαδικής πρόσθεσης ο οποίος θα προσδιορίσει και τη συνάρτηση του ημιαθροιστή.

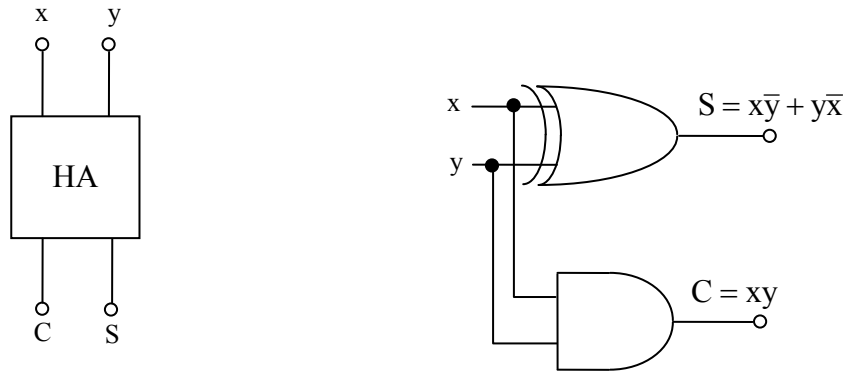
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Το κρατούμενο είναι 0, εκτός από την περίπτωση που και οι δύο εισοδοι είναι 1. Η έξοδος S είναι το λιγότερο σημαντικό bit του αθροίσματος. Οι απλοποιημένες συναρτήσεις Boole των εξόδων μπορούν να εξαχθούν κατευθείαν από τον πίνακα αληθείας και είναι - σε μορφή αθροίσματος γινομένων - οι εξής:

$$S = \bar{x}y + x\bar{y} = x \oplus y$$

$$C = xy$$

Το λογικό διάγραμμα της υλοποίησης και το συμβολικό σχήμα του ημιαθροιστή φαίνεται στο Σχ. 5.1. Η υλοποίηση που δείχνουμε είναι η απλούστερη δυνατή διότι χρησιμοποιεί μόνο μια πύλη XOR που υλοποιεί το άθροισμα S και μια πύλη AND που υλοποιεί το κρατούμενο C. Αυτή η υλοποίηση θα χρησιμοποιηθεί αργότερα για να δείξει ότι δύο κυκλώματα ημιαθροιστών είναι απαραίτητα για την κατασκευή ενός πλήρη αθροιστή.



Σχ. 5.1. α) Συμβολικό διάγραμμα, και β) λογικό κύκλωμα ημιαθροιστή.

5.1.2. Πλήρης Αθροιστής (Full-Adder)

Ο πλήρης αθροιστής είναι ένα συνδυαστικό κύκλωμα που σχηματίζει το αριθμητικό άθροισμα τριών bits εισόδου. Έχει τρεις εισόδους και δύο εξόδους. Δύο από τις μεταβλητές εισόδου, που συμβολίζουμε με x και y, παριστάνουν τα δύο σημαντικά bits που προστίθενται. Η τρίτη είσοδος, η z, παριστάνει το κρατούμενο από την αμέσως προηγούμενη λιγότερο σημαντική θέση (δηλ. αν η θέση του πλήρη αθροιστή είναι σημαντικότητας n, τότε η είσοδος z είναι ουσιαστικά το κρατούμενο C_{n-1} της προηγούμενης σε σημαντικότητα βαθμίδας). Χρειάζονται δύο εξόδοι, διότι το αριθμητικό άθροισμα τριών δυαδικών ψηφίων κυμαίνεται από 0 μέχρι 3, και οι δυαδικοί 2 και 3 είναι διψήφιοι. Οι δύο εξόδοι συμβολίζονται με S για το άθροισμα και C για το κρατούμενο. Η δυαδική μεταβλητή S δίνει την τιμή του λιγότερο σημαντικού bit του αθροίσματος. Η δυαδική μεταβλητή C δίνει το κρατούμενο εξόδου. Ο πίνακας αληθείας του πλήρη-αθροιστή είναι ο εξής

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

Τα 1 και 0 των μεταβλητών εξόδου καθορίζονται από το αριθμητικό άθροισμα των bits εισόδου. Όταν όλα τα bits εισόδου είναι 0, η έξοδος είναι 0. Η έξοδος S ισούται με 1, όταν μόνο μία μεταβλητή εισόδου ισούται με ένα ή όταν και οι τρεις μεταβλητές εισόδου ισούνται με 1. Η έξοδος C δίνει κρατούμενο 1, όταν δύο ή τρεις εισόδοι ισούνται με 1.

Τα bits εισόδου και εξόδου του συνδυαστικού κυκλώματος έχουν διαφορετικές ερμηνείες σε διαφορετικές φάσεις του προβλήματος. Από τη μια μεριά, τα δυαδικά σήματα των εισόδων θεωρούνται σαν δυαδικά ψηφία και προστίθενται αριθμητικά για να σχηματίσουν ένα διψήφιο άθροισμα στην έξοδο. Από την άλλη μεριά, οι ίδιες δυαδικές τιμές θεωρούνται σαν μεταβλητές συναρτήσεων Boole και σαν τέτοιες χρησιμοποιούνται στον πίνακα αληθείας και όταν το κύκλωμα υλοποιείται με λογικές πύλες. Είναι σημαντικό να συνειδητοποιήσουμε ότι δίνονται δύο διαφορετικές ερμηνείες στις τιμές των bits που συναντώνται σ' αυτό το κύκλωμα.

x \ yz	00	01	11	10
0		1		1
1	1		1	

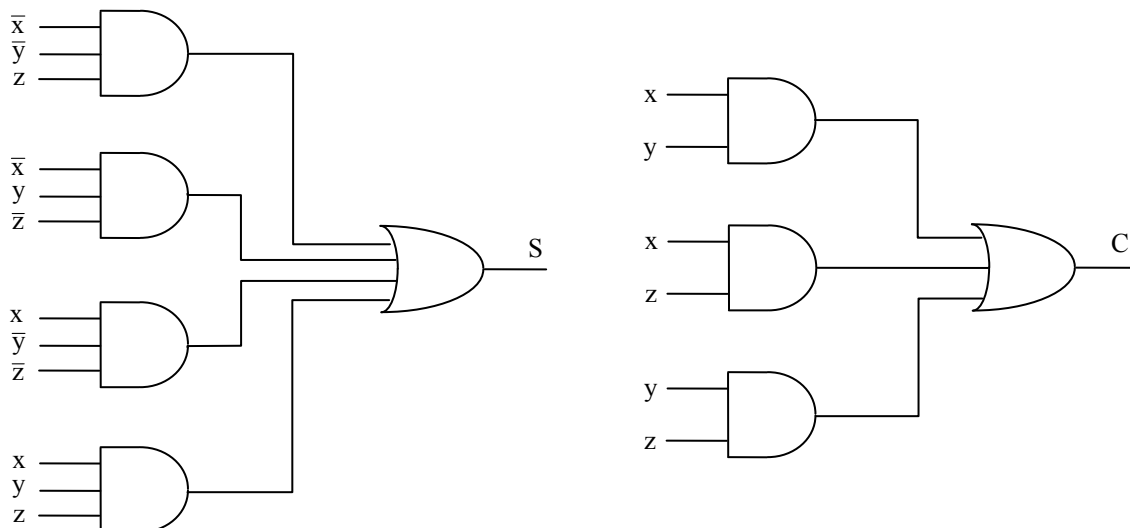
$$S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$$

x \ yz	00	01	11	10
0			1	
1		1	1	1

$$C = xy + xz + yz$$

Σχ. 5.2. Χάρτες Karnaugh για τον πλήρη αθροιστή.

Η λογική σχέση εισόδου - εξόδου του κυκλώματος πλήρους αθροιστή μπορεί να εκφραστεί με δύο συναρτήσεις Boole, μία για κάθε μεταβλητή εξόδου. Καθεμιά τους χρειάζεται έναν ξεχωριστό χάρτη για να απλοποιηθεί. Ο κάθε χάρτης πρέπει να έχει οκτώ τετράγωνα, αφού κάθε έξοδος είναι συνάρτηση τριών μεταβλητών εισόδου. Χρησιμοποιούμε τους χάρτες του Σχ. 5.2. Οι μονάδες στα τετράγωνα των χαρτών των S και C καθορίζονται κατευθείαν από τον πίνακα αληθείας. Για το S, δεν υπάρχουν γειτονικά τετράγωνα που να μπορούν να συνδυαστούν για να δώσουν μία απλοποιημένη έκφραση σε άθροισμα γινομένων. Η έξοδος C μπορεί να απλοποιηθεί σε μια έκφραση έξι παραγόντων.



Σχ. 5.3. Υλοποίηση πλήρη αθροιστή σε άθροισμα γινομένων.

Το λογικό διάγραμμα για τον πλήρη αθροιστή υλοποιημένο σε άθροισμα γινομένων, φαίνεται στο Σχ. 5.3. Αυτή η υλοποίηση χρησιμοποιεί τις εκφράσεις Boole που προκύπτουν από τους χάρτες Karnaugh

$$S = \bar{x} \bar{y} z + \bar{x} y \bar{z} + x \bar{y} \bar{z} + x y z$$

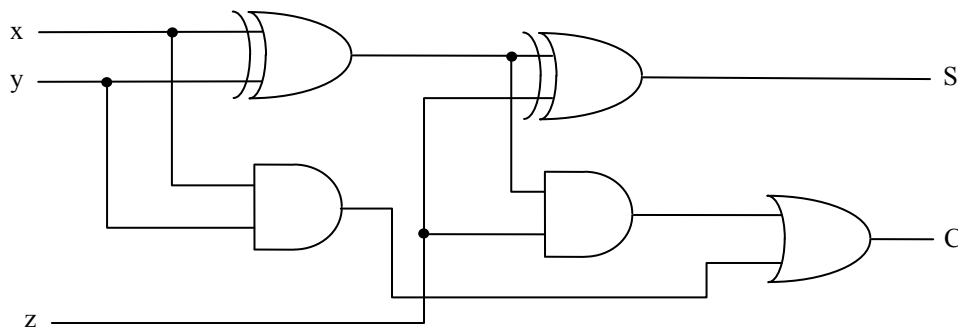
$$C = xy + xz + yz$$

Μπορούν να βρεθούν και άλλες μορφές ενός πλήρη αθροιστή. Ένας πλήρης αθροιστής μπορεί να υλοποιηθεί με δύο ημιαθροιστές και μία πύλη OR όπως φαίνεται στο Σχ. 5.4. Η έξοδος S του δεύτερου ημιαθροιστή είναι το XOR του z με την έξοδο του πρώτου ημιαθροιστή

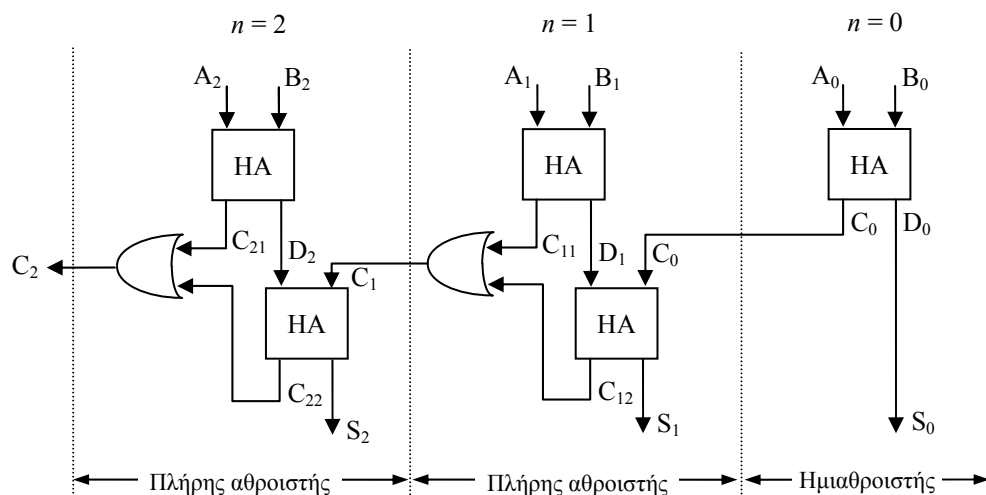
$$\begin{aligned} S &= z \oplus (x \oplus y) \\ &= \bar{z}(x\bar{y} + \bar{x}y) + z(\overline{x\bar{y} + \bar{x}y}) \\ &= \bar{z}(x\bar{y} + \bar{x}y) + z(xy + \bar{x}\bar{y}) \\ &= x\bar{y}\bar{z} + \bar{x}y\bar{z} + xyz + \bar{x}\bar{y}z \end{aligned}$$

και η έξοδος του κρατουμένου είναι

$$C = z(x\bar{y} + \bar{x}y) + xy = x\bar{y}z + \bar{x}yz + xy = z(x \oplus y) + xy$$



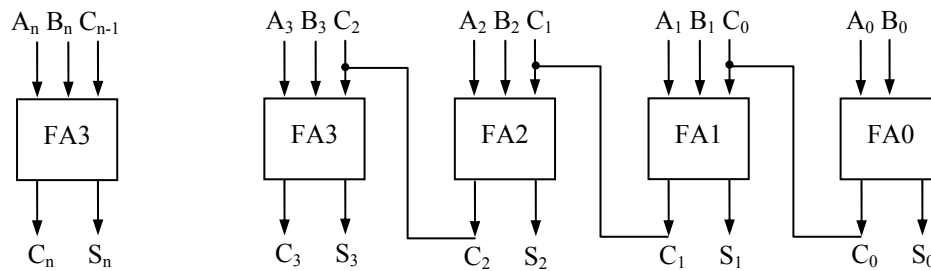
Σχ. 5.4. Υλοποίηση πλήρη αθροιστή με δύο ημιαθροιστές και μία πύλη OR.



Σχ. 5.5. Συνδεσμολογία πλήρη αθροιστή για την πρόσθεση δύο δυαδικών αριθμών 3 ψηφίων.

Στο Σχ. 5.5 φαίνεται η συνδεσμολογία του πλήρη αθροιστή προκειμένου να εκτελεστεί η πρόσθεση δύο δυαδικών αριθμών 3 ψηφίων. Δείχνεται το πλήρες κύκλωμα κάθε πλήρη αθροιστή αποτελούμενου από δύο ημιαθροιστές και μία πύλη

OR. Στο Σχ. 5.6 φαίνεται η χονδρική διάταξη ενός πλήρη αθροιστή τεσσάρων βαθμίδων για την πρόσθεση δύο τετραψήφιων δυαδικών αριθμών.



Σχ. 5.6. Χονδρική διάταξη πλήρη αθροιστή τεσσάρων βαθμίδων για την πρόσθεση δύο τετραψήφιων δυαδικών αριθμών.

Παράδειγμα 5.1: Να σχεδιαστεί αθροιστής με σειριακό τρόπο λειτουργίας.

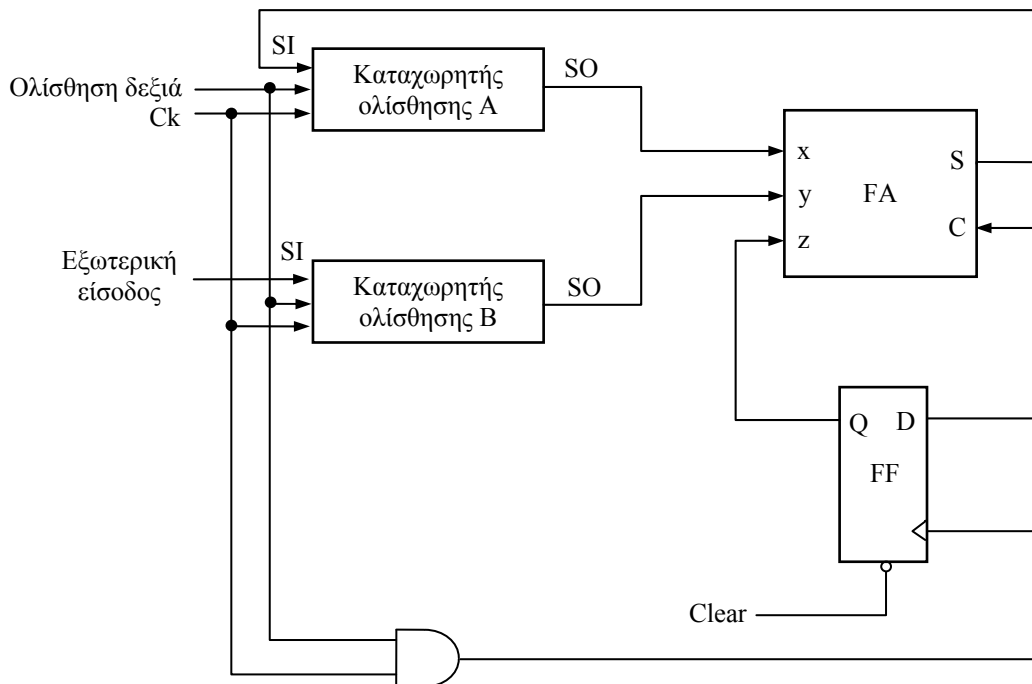
Λύση:

Στο παράδειγμα αυτό θα μελετήσουμε το σχεδιασμό ενός δυαδικού αθροιστή ο οποίος όμως λειτουργεί σειριακά και όχι παράλληλα όπως είδαμε στην παράγραφο 5.1.2. Στη σειριακή πρόσθεση, οι δύο προσθετέοι βρίσκονται αποθηκευμένοι σε δύο καταχωρητές ολίσθησης. Τα bits τους προστίθενται ζευγάρι-ζευγάρι, το ένα μετά το άλλο, με τη βοήθεια ενός και μόνο πλήρη αθροιστή (FA), όπως δείχνει το Σχ. 5.7. Το κρατούμενο που παράγεται από τον πλήρη αθροιστή κρατιέται σε ένα flip-flop τύπου D. Η έξοδος αυτού του flip-flop, μετά, χρησιμοποιείται σαν κρατούμενο εισόδου για το επόμενο σε σημαντικότητα ζευγάρι των bits. Τους δύο καταχωρητές τους ολισθαίνουμε προς τα δεξιά επί μία περίοδο ίση με το χρόνο λέξης. Τα bits του αθροίσματος, από την έξοδο S του πλήρη αθροιστή, θα μπορούσαν να μεταφερθούν σε ένα τρίτο καταχωρητή ολίσθησης. Μία άλλη δυνατότητα, όμως, που την παρουσιάζουμε στο Σχ. 5.7, είναι το να ολισθαίνουμε τα bits του αθροίσματος μέσα στον καταχωρητή A, την ίδια ώρα που παίρνουμε με ολίσθηση τα αρχικά bits του A από αυτόν. Κατ' αυτόν τον τρόπο χρησιμοποιούμε τον ίδιο καταχωρητή για τον ένα προσθετέο και για το άθροισμα. Επίσης, η σειριακή είσοδος (SI) του καταχωρητή B μπορεί να δεχτεί έναν καινούριο δυαδικό αριθμό, καθώς τα bits του άλλου προσθετέου ολισθαίνουν προς τα έξω για να γίνει η πρόσθεση.

Η λειτουργία του σειριακού αθροιστή έχει ως εξής: Αρχικά, οι καταχωρητές A και B περιέχουν τους προσθετέους, το δε flip-flop του κρατουμένου το μηδενίζουμε. Οι σειριακές εξοδοί (SO) των A και B δίνουν ένα ζευγάρι bits στα x και y του πλήρη αθροιστή. Η έξοδος Q του flip-flop δίνει το κρατούμενο εισόδου στο z. Ο έλεγχος δεξιά-ολίσθησης ενεργοποιεί και τους δύο καταχωρητές και το flip-flop του κρατουμένου, κι έτσι με τον επόμενο παλμό του ρολογιού, οι δύο καταχωρητές ολισθαίνουν προς τα δεξιά κατά μία θέση, το bit του αθροίσματος από το S μπαίνει στο αριστερότερο flip-flop του A και το κρατούμενο εξόδου μεταφέρεται στο flip-flop Q.

Ο έλεγχος δεξιάς-ολίσθησης συνεχίζει να ενεργοποιεί τους καταχωρητές, για ένα συνολικό αριθμό παλμών ρολογιού ίσο προς τον αριθμό των bits που περιέχει ο κάθε καταχωρητής. Με καθέναν από αυτούς τους παλμούς του ρολογιού κι από ένα καινούριο bit του αθροίσματος μεταφέρεται στον A, κι από ένα καινούριο κρατούμενο μεταφέρεται στο Q, οι δε καταχωρητές ολισθαίνουν μία θέση δεξιά.

Αυτά συνεχίζονται έως ότου σταματήσει ο έλεγχος δεξιάς-ολίσθησης. Έτσι, η πρόσθεση γίνεται περνώντας κάθε ζευγάρι bits μαζί με το προηγούμενο κρατούμενο μέσα από τον ένα και μοναδικό πλήρη αθροιστή και μεταφέροντας το άθροισμα, bit-προς-bit, στον καταχωρητή A.



Σχ. 5.7. Σειριακός αθροιστής.

Εάν θέλουμε να προσθέσουμε κι έναν άλλον αριθμό στο περιεχόμενο του καταχωρητή A, τότε θα πρέπει πρώτα να τον μεταφέρουμε σειριακά μέσα στον καταχωρητή B. Μετά, επαναλαμβάνοντας τη διαδικασία άλλη μια φορά, θα πάρουμε στον A το άθροισμά του και με το δεύτερο αριθμό.

5.2. Αφαιρέτες (Subtractors)

Η αφαίρεση δύο δυαδικών αριθμών μπορεί να υλοποιηθεί με λογικά κυκλώματα κατά “άμεσο” τρόπο - όπως την κάνουμε με χαρτί και μολύβι. Με αυτή τη μέθοδο, κάθε bit αφαιρετέου αφαιρείται από το αντίστοιχης σημαντικότητας bit του μειωτέου και δίνει ένα bit της διαφοράς. Εάν το bit του μειωτέου είναι μικρότερο από το bit του αφαιρετέου, δανειζόμαστε ένα 1 από την επόμενη σημαντική θέση. Το γεγονός ότι δανειστήκαμε μια μονάδα, πρέπει κάπως να μεταβιβαστεί στο επόμενο προς τα πάνω ζευγάρι bits, μέσω κάποιου δυαδικού σήματος, που βγαίνει (έξοδος) από τη μια βαθμίδα και μπαίνει (είσοδος) στην επόμενη προς τα πάνω σημαντικότητας βαθμίδα. Όπως ακριβώς υπάρχουν ημιαθροιστές και πλήρεις αθροιστές, υπάρχουν και “ημιαφαιρέτες” (half-subtractors) και “πλήρεις αφαιρέτες” (full-subtractors).

5.2.1. Ημιαφαιρέτης (Half-Subtractor)

Ο ημιαφαιρέτης είναι ένα συνδυαστικό κύκλωμα που αφαιρεί δύο bits και δίνει τη διαφορά τους. Έχει επίσης μία έξοδο που λέει αν χρειάστηκε να δανειστεί (borrow) μια μονάδα. Ας συμβολίσουμε το μειωτέο bit με x και τον αφαιρετέο με y. Για να

κάνουμε την αφαίρεση $x - y$ πρέπει να ελέγξουμε τα σχετικά μεγέθη των x και y . Εάν $x \geq y$, τότε έχουμε τρεις πιθανότητες: $0 - 0 = 0$, $1 - 0 = 1$ και $1 - 1 = 0$. Το αποτέλεσμα λέγεται “bit της διαφοράς” (difference bit). Εάν $x < y$, τότε έχουμε $0 - 1$, κι έτσι χρειάζεται να δανειστούμε ένα 1 από την επόμενη προς τα πάνω βαθμίδα. Το 1 που δανειζόμαστε από την επόμενη θέση προσθέτει 2 στο bit του μειωτέου, όπως στο δεκαδικό σύστημα το “κρατούμενο” (“δανεικό”) της αφαίρεσης προσθέτει 10 στο μειωτέο. Με το μειωτέο ίσο με 2, η διαφορά γίνεται $2 - 1 = 1$. Ο ημιαφαιρέτης χρειάζεται δύο εξόδους. Η μία έξοδος παράγει τη διαφορά και θα συμβολίζεται με D . Η δεύτερη έξοδος, που θα συμβολίζεται με B (borrow - δανεικό - κρατούμενο), παράγει το δυαδικό σήμα που πληροφορεί την επόμενη βαθμίδα ότι δανειστήκαμε μια μονάδα. Ο πίνακας αληθείας για τις σχέσεις εισόδου - εξόδου ενός ημιαφαιρέτη έχει επομένως ως εξής

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

Το κρατούμενο εξόδου B είναι 0 υπό τον όρο ότι $x \geq y$. Είναι 1 για $x = 0$ και $y = 1$. Το D είναι το αποτέλεσμα της αριθμητικής πράξης $2B + x - y$. Οι συναρτήσεις Boole για τις δύο εξόδους του ημιαφαιρέτη βρίσκονται κατευθείαν από τον πίνακα αληθείας

$$D = \bar{x}y + x\bar{y}$$

$$B = \bar{x}y$$

Είναι ενδιαφέρον να σημειωθεί ότι η συνάρτηση του D είναι ακριβώς η ίδια όπως η λογική συνάρτηση του S του ημιαθροιστή.

5.2.2. Πλήρης Αφαιρέτης (Full Subtractor)

Ο πλήρης αφαιρέτης είναι ένα συνδυαστικό κύκλωμα που εκτελεί την αφαίρεση μεταξύ δύο bits, παίρνοντας υπόψη ότι μπορεί η λιγότερο σημαντική βαθμίδα να μας έχει δανειστεί μια μονάδα. Αυτό το κύκλωμα έχει τρεις εισόδους και δύο εξόδους. Οι τρεις εισόδοι x , y και z , συμβολίζουν το μειωτέο, τον αφαιρετέο, και το προηγούμενο κρατούμενο, αντίστοιχα. Οι δύο εξόδοι, D και B , συμβολίζουν τη διαφορά και το κρατούμενο εξόδου, αντίστοιχα. Ο πίνακας αληθείας του κυκλώματος είναι ο εξής:

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

Τα 1 και 0 των μεταβλητών εξόδου υπολογίζονται από την αφαίρεση $2B + x - y - z$. Οι συνδυασμοί που έχουν κρατούμενο εισόδου $z = 0$ δίνουν ίδιες εξόδους, όπως και ο ημιαθροιστής. Για $x = 0, y = 0$ και $z = 1$, πρέπει να δανειστούμε ένα 1 από την επόμενη βαθμίδα, που κάνει το $B = 1$ και προσθέτει 2 στο x . Αφού $2 - 0 - 1 = 1$, άρα $D = 1$. Για $x = 0$ και $yz = 11$, χρειάζεται να ξαναδανειστούμε, κάνοντας το $B = 1$ και το $x = 2$. Αφού $2 - 1 - 1 = 0$, άρα $D = 0$. Για $x = 1$ και $yz = 01$, έχουμε ότι $x - y - z = 0$, που κάνει το $B = 0$ και το $D = 0$. Τελικά για $x = 1, y = 1, z = 1$, πρέπει να δανειστούμε 1, κάνοντας το $B = 1$ και το $x = 3$, οπότε $3 - 1 - 1 = 1$, κάνοντας το $D = 1$.

x \ yz	00	01	11	10
0		1		1
1	1		1	

$$D = \bar{x} \bar{y} z + \bar{x} y \bar{z} + x \bar{y} \bar{z} + xyz$$

x \ yz	00	01	11	10
0		1	1	1
1			1	

$$B = \bar{x} y + \bar{x} z + yz$$

Σχ. 5.8. Χάρτες Karnaugh για τον πλήρη αφαιρέτη.

Οι απλοποιημένες συναρτήσεις Boole των δύο εξόδων του πλήρη αφαιρέτη βρίσκονται από τους χάρτες του Σχ. 5.8. Σε μορφή αθροίσματος γινομένων είναι

$$D = \bar{x} \bar{y} z + \bar{x} y \bar{z} + x \bar{y} \bar{z} + xyz$$

$$B = \bar{x} y + \bar{x} z + yz$$

Ας σημειωθεί τέλος, ότι ο πλήρης αφαιρέτης μπορεί να χρησιμοποιηθεί ως συγκριτής δύο αριθμών. Εάν το κρατούμενο z της τελευταίας βαθμίδας (δηλ. της βαθμίδας υψηλότερης σημαντικότητας) είναι 0 τότε ο μειωτέος x είναι μεγαλύτερος από τον αφαιρετέο y . Αν το κρατούμενο z της τελευταίας βαθμίδας είναι 1 τότε ο μειωτέος x είναι μικρότερος από τον αφαιρετέο y .

5.3. Αποκωδικοποιητές (Decoders)

Οι διακριτές ποσότητες πληροφοριών στα ψηφιακά συστήματα συμβολίζονται με δυαδικούς κώδικες. Ένας δυαδικός κώδικας των n bits είναι ικανός να παραστήσει έως 2^n διαφορετικά στοιχεία της κωδικοποιημένης πληροφορίας. Ένας αποκωδικοποιητής (decoder) είναι ένα συνδυαστικό κύκλωμα που μετατρέπει τη δυαδική πληροφορία n γραμμών εισόδου σε έως 2^n μοναδικές γραμμές εξόδου. Οι αποκωδικοποιητές που θα μας απασχολήσουν ονομάζονται αποκωδικοποιητές από n σε m γραμμές, όπου $m \leq 2^n$. Σκοπός τους είναι να παράγουν τους 2^n (ή λιγότερους) ελαχιστόρους των n μεταβλητών εισόδου.

Σαν παράδειγμα, δίνεται το κύκλωμα αποκωδικοποιητή τεσσάρων - σε δέκα γραμμών του Σχ. 5.9. Οι τέσσερις εισοδοί αποκωδικοποιούνται σε δέκα εξόδους, όπου κάθε έξοδος συμβολίζει έναν από τους ελαχιστόρους των τεσσάρων μεταβλητών εισόδου. Οι τέσσερις αντιστροφείς παράγουν το συμπλήρωμα των εισόδων και καθεμιά από τις δέκα πύλες AND παράγει κι έναν από τους ελαχιστόρους. Μια συγκεκριμένη εφαρμογή αυτού του αποκωδικοποιητή θα μπορούσε να είναι η μετατροπή από δυαδικό σε δεκαδικό αριθμό. Οι μεταβλητές εισόδου μπορεί να συμβολίζουν ένα δυαδικό αριθμό, οπότε οι εξοδοί θα παριστάνουν

Ένας αποκωδικοποιητής μας δίνει τους 2^n ελαχιστόρους των n μεταβλητών εισόδου. Αφού κάθε συνάρτηση Boole μπορεί να εκφραστεί σε κανονική μορφή αθροίσματος ελαχιστόρων, μπορεί κανείς να χρησιμοποιήσει έναν αποκωδικοποιητή για να παράγει τους ελαχιστόρους και μία εξωτερική πύλη OR για να σχηματίσει το άθροισμα. Μ' αυτό τον τρόπο, κάθε συνδυαστικό κύκλωμα με n εισόδους και m εξόδους μπορεί να υλοποιηθεί με έναν αποκωδικοποιητή n - σε - 2^n γραμμών και με m πύλες OR.

Παράδειγμα 5.2:

Να υλοποιηθεί ένα κύκλωμα πλήρη αθροιστή με έναν αποκωδικοποιητή και δύο πύλες OR.

Λύση:

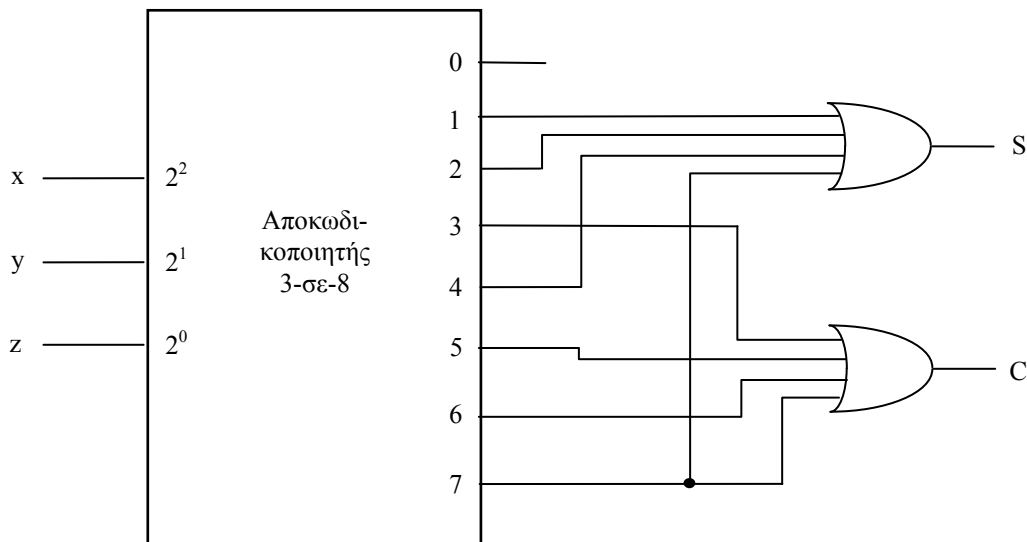
Από τον πίνακα αληθείας του πλήρη αθροιστή, παίρνουμε τις συναρτήσεις αυτού του συνδυαστικού κυκλώματος σαν άθροισμα ελαχιστόρων

$$S(x, y, z) = \Sigma(1, 2, 4, 7)$$

$$C(x, y, z) = \Sigma(3, 5, 6, 7)$$

όπου οι αριθμοί στις παρενθέσεις είναι το δεκαδικό αντίστοιχο των ελαχιστόρων με το συγκεκριμένο συνδυασμό εισόδων [π.χ. $(3)_{10} = (011)_2$].

Αφού υπάρχουν τρεις εισοδοί και συνολικά οκτώ ελαχιστόροι, χρειαζόμαστε έναν αποκωδικοποιητή 3 - σε - 8 γραμμών. Η υλοποίηση φαίνεται στο Σχ. 5.10. Ο αποκωδικοποιητής παράγει τους οκτώ ελαχιστόρους των x, y, z . Η πύλη OR για την έξοδο S σχηματίζει το άθροισμα των ελαχιστόρων 1, 2, 4 και 7. Η πύλη OR για την έξοδο C σχηματίζει το άθροισμα των ελαχιστόρων 3, 5, 6 και 7.

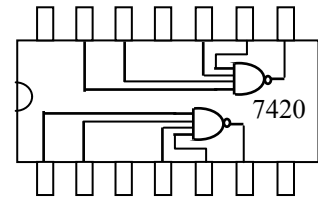


Σχ. 5.10. Υλοποίηση ενός πλήρη αθροιστή με αποκωδικοποιητή.

Παράδειγμα 5.3:

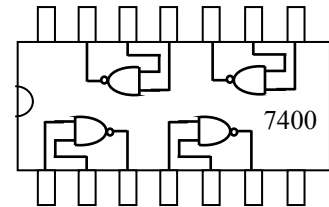
Σε ένα συμβούλιο αξιολόγησης μίας προμήθειας υπάρχουν 3 κριτές. Οι ψήφοι τους έχουν διαφορετικά βάρη:

Του κριτή Α έχει βάρος 3
 Του κριτή Β έχει βάρος 2
 Του κριτή Γ έχει βάρος 1



Καθένας από τους κριτές έχει έναν διακόπτη που τον κλείνει για να ψηφίσει ΝΑΙ και τον ανοίγει για να ψηφίσει ΟΧΙ. Σε περίπτωση ισοψηφίας το αποτέλεσμα είναι ΟΧΙ.

1) Να δώσετε τον πίνακα αληθείας της συνάρτησης που υλοποιεί το αποτέλεσμα της ψηφοφορίας.



2) Από τον παραπάνω πίνακα αληθείας και χωρίς να απλοποιήσετε τη συνάρτηση που προκύπτει να σχεδιάσετε το κύκλωμα που δείχνει αν το αποτέλεσμα της ψηφοφορίας είναι ΝΑΙ ή ΟΧΙ.

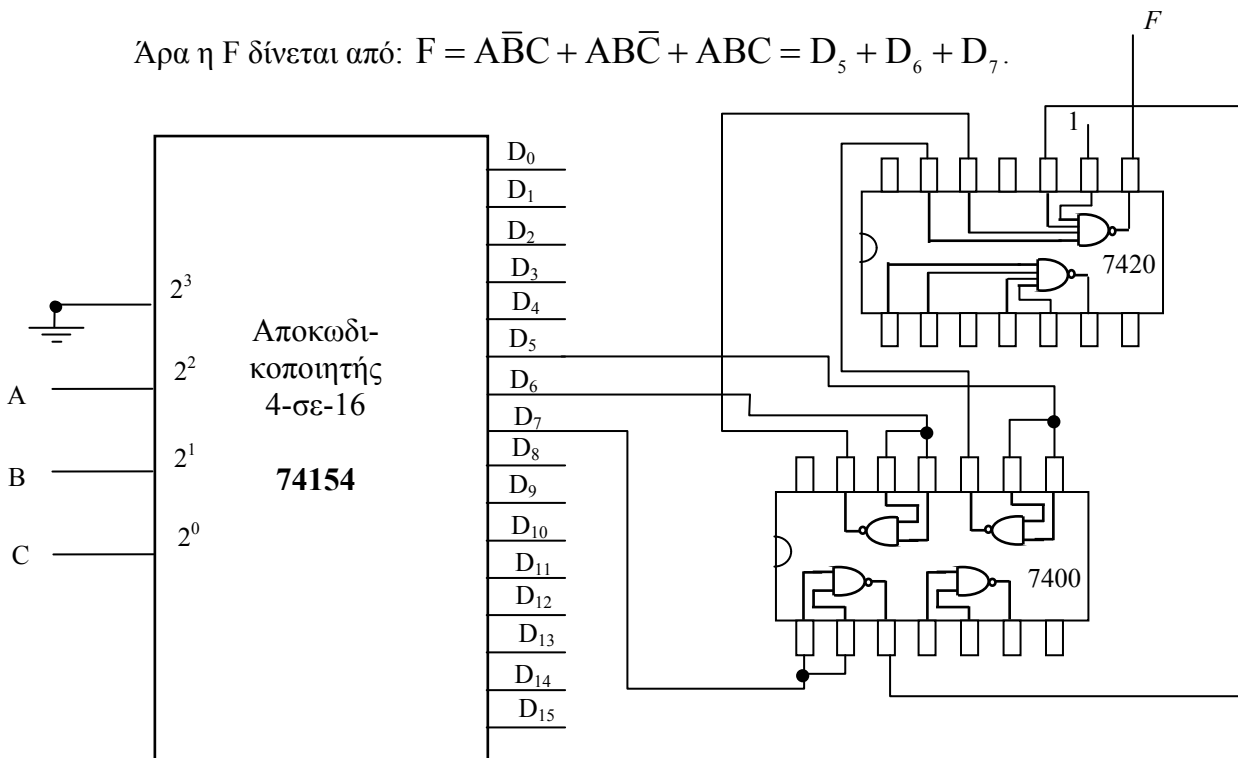
Έχετε στη διάθεσή σας τα ολοκληρωμένα 74154 (αποκωδικοποιητής 4-σε-16), 7420 και 7400 (των οποίων η χωροδιάταξη παρουσιάζεται παραπλεύρως).

Λύση

Με βάση τα δεδομένα, ο πίνακας αληθείας της προς υλοποίηση συνάρτησης είναι:

3	2	1	
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$\text{Άρα η F δίνεται από: } F = \overline{A}BC + A\overline{B}C + ABC = D_5 + D_6 + D_7.$$



5.4. Κωδικοποιητές (Coders)

Ένας κωδικοποιητής είναι ένα ψηφιακό κύκλωμα που εκτελεί την αντίστροφη λειτουργία από ό,τι ο αποκωδικοποιητής. Ένας κωδικοποιητής έχει 2^n (ή λιγότερες) γραμμές εισόδου και n γραμμές εξόδου. Οι γραμμές εξόδου παράγουν το δυαδικό κώδικα που αντιστοιχεί στις μεταβλητές εισόδου. Παράδειγμα κωδικοποιητή είναι ο οκταδικός - σε - δυαδικό, του οποίου ο πίνακας αληθείας δίνεται στον πιο κάτω πίνακα. Έχει οκτώ εισόδους, μία για κάθε οκταδικό ψηφίο, και τρεις εξόδους που παράγουν τον αντίστοιχο δυαδικό αριθμό. Υποτίθεται ότι μόνο μία είσοδος έχει την τιμή 1 σε μια δοσμένη χρονική στιγμή, αλλιώς το κύκλωμα δεν έχει νόημα.

Ο κωδικοποιητής μπορεί να υλοποιηθεί με πύλες OR, των οποίων οι εισοδοί καθορίζονται κατευθείαν από τον πίνακα αληθείας. Η έξοδος z είναι ίση με 1, αν το οκταδικό ψηφίο εισόδου είναι 1 ή 3 ή 5 ή 7. Η έξοδος y είναι 1 για τα οκταδικά ψηφία 2, 3, 6, ή 7 και η έξοδος x είναι 1 για τα ψηφία 4, 5, 6, ή 7. Αυτές οι συνθήκες μπορούν να εκφραστούν από τις επόμενες συναρτήσεις Boole εξόδου:

Είσοδοι								Έξοδοι		
D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$z = D_1 + D_3 + D_5 + D_7$$

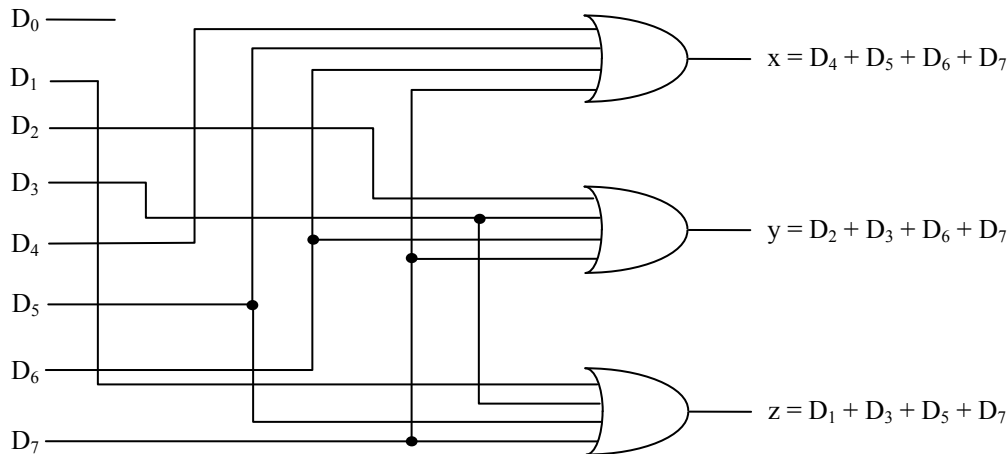
$$y = D_2 + D_3 + D_6 + D_7$$

$$x = D_4 + D_5 + D_6 + D_7$$

Ο κωδικοποιητής υλοποιείται με τρεις πύλες OR, όπως φαίνεται στο Σχ. 5.11.

Ο κωδικοποιητής που ορίστηκε με τον πιο πάνω πίνακα έχει τον περιορισμό ότι μόνο μία είσοδος μπορεί να είναι ενεργός σε κάθε χρονική στιγμή. Αν δύο εισοδοί είναι ενεργές ταυτόχρονα, η έξοδος παράγει έναν απροσδιόριστο συνδυασμό. Για παράδειγμα, αν D_3 και D_6 είναι 1 ταυτόχρονα, η έξοδος του κωδικοποιητή θα είναι 111, διότι και οι τρεις έξοδοι είναι ίσες με 1. Αυτό όμως δεν παριστάνει το δυαδικό 3 ούτε το δυαδικό 6. Για την άρση αυτής της αβεβαιότητας, τα κυκλώματα του κωδικοποιητή θα πρέπει να καθορίζουν μια προτεραιότητα που να εξασφαλίζει ότι μόνο μία είσοδος κωδικοποιείται. Αν δώσουμε μεγαλύτερη προτεραιότητα στις εισόδους με μεγαλύτερο δείκτη, και αν και η D_3 και η D_6 είναι 1 ταυτόχρονα, η έξοδος θα είναι 110, διότι η D_6 έχει μεγαλύτερη προτεραιότητα από την D_3 .

Μια άλλη αβεβαιότητα στον κωδικοποιητή οκταδικό - σε δυαδικό είναι ότι όταν όλες οι εισοδοί είναι 0, δημιουργείται μια έξοδος όλο μηδενικά. Το πρόβλημα είναι ότι και όταν η D_0 ισούται με 1, τότε πάλι η έξοδος έχει όλο μηδενικά. Για να άρουμε και αυτή την αβεβαιότητα, θα πρέπει να δημιουργήσουμε μία επιπλέον έξοδο, η οποία καθορίζει την κατάσταση κατά την οποία καμιά από τις εισόδους δεν είναι ενεργός.



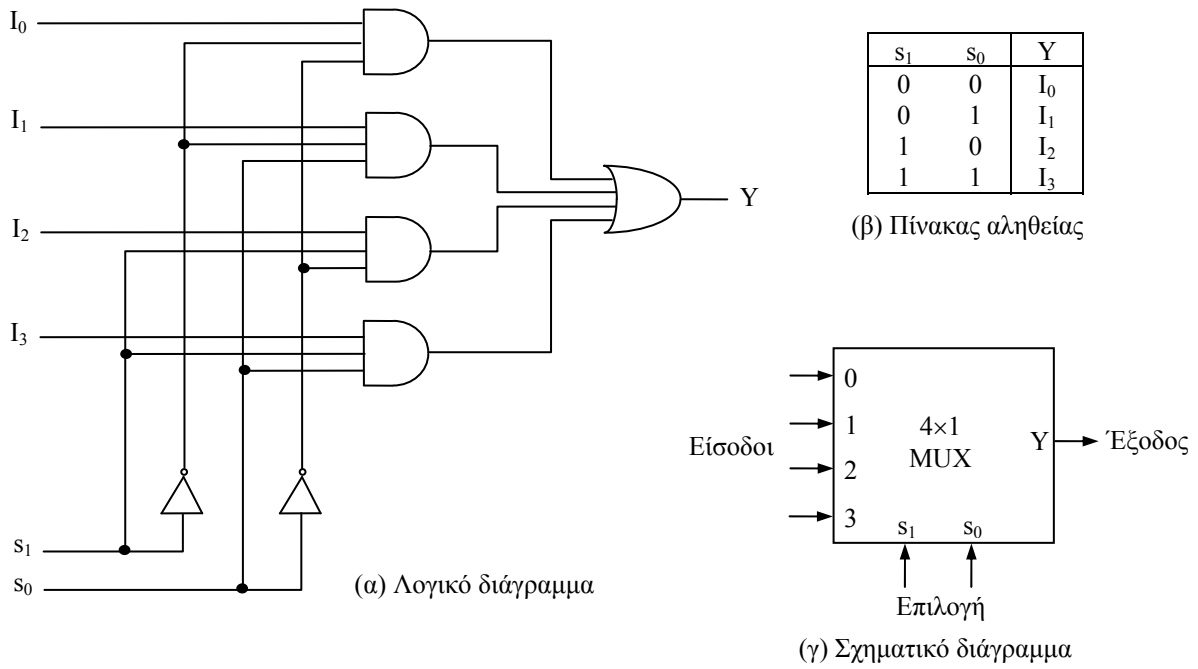
Σχ. 5.11. Κωδικοποιητής από οκταδικό σε δυαδικό.

5.5. Πολυπλέκτες (Multiplexers)

Πολυπλεξία (multiplexing) σημαίνει τη μεταβίβαση ενός μεγάλου αριθμού πληροφοριών μέσα από ένα μικρότερο αριθμό καναλιών ή γραμμών. Ένας ψηφιακός πολυπλέκτης είναι ένα συνδυαστικό κύκλωμα που επιλέγει δυαδικές πληροφορίες ανάμεσα σε πολλές γραμμές εισόδου και τις κατευθύνει στη μία μοναδική γραμμή εξόδου. Η επιλογή της μίας συγκεκριμένης γραμμής εισόδου γίνεται μέσω μερικών γραμμών επιλογής. Κανονικά, υπάρχουν 2^n γραμμές εισόδου και n γραμμές επιλογής που οι συνδυασμοί των bit τους καθορίζουν ποια είσοδος επιλέγεται.

Στο Σχ. 5.12 φαίνεται ένας πολυπλέκτης από 4 σε 1 γραμμή. Καθεμιά από τις τέσσερις γραμμές εισόδου, I_0 ως I_3 , εφαρμόζεται σε μία είσοδο μιας πύλης AND. Οι γραμμές επιλογής s_1 και s_0 αποκωδικοποιούνται και επιλέγουν μία συγκεκριμένη πύλη AND. Ο πίνακας της συνάρτησης, στο σχήμα, δείχνει το δρόμο που αποκαθίσταται από την είσοδο προς την έξοδο για κάθε πιθανό συνδυασμό των γραμμών επιλογής. Όταν ένας τέτοιος πολυπλέκτης χρησιμοποιείται στο σχεδιασμό ενός ψηφιακού συστήματος, συμβολίζεται σχηματικά όπως στο Σχ. 5.12γ. Για να περιγράψουμε τη λειτουργία του κυκλώματος, θεωρήστε την περίπτωση όπου $s_1 s_0 = 10$. Η πύλη AND που συνδέεται με την είσοδο I_2 έχει δύο από τις εισόδους της ίσες με 1 και την τρίτη είσοδο συνδεδεμένη με το I_2 . Οι άλλες τρεις πύλες AND έχουν τουλάχιστον μία είσοδό τους ίση με 0, οπότε οι έξοδοι είναι αναγκαστικά όλες 0. Η έξοδος της πύλης OR ισούται τώρα με την τιμή της I_2 , παρέχοντας έτσι ένα δίαυλο από την είσοδο που διαλέγουμε ως την έξοδο. Ο πολυπλέκτης ονομάζεται επίσης “επιλογέας δεδομένων” (data selector), εφόσον επιλέγει μία είσοδο μεταξύ πολλών και οδηγεί τη δυαδική πληροφορία της στη γραμμή εξόδου.

Οι πύλες AND και οι αντιστροφείς στον πολυπλέκτη μοιάζουν με το κύκλωμα του αποκωδικοποιητή και όντως αποκωδικοποιούν τις γραμμές επιλογής. Γενικά, ένας πολυπλέκτης από 2^n - σε - 1 γραμμή κατασκευάζεται από έναν αποκωδικοποιητή από n - σε - 2^n γραμμές, προσθέτοντας σ' αυτόν 2^n γραμμές εισόδου - μία σε κάθε πύλη AND. Οι έξοδοι των πύλων AND εφαρμόζονται σε μία μοναδική πύλη OR για να δώσουν τη μία γραμμή εξόδου. Σαν μέγεθος ενός πολυπλέκτη συνήθως δίνεται ο αριθμός 2^n των γραμμών εισόδου και η μία, γραμμή εξόδου. Αυτό συνεπάγεται ότι περιέχει επίσης n γραμμές επιλογής. Ένας πολυπλέκτης (multiplexer) συχνά ονομάζεται συντετμημένα “MUX”.

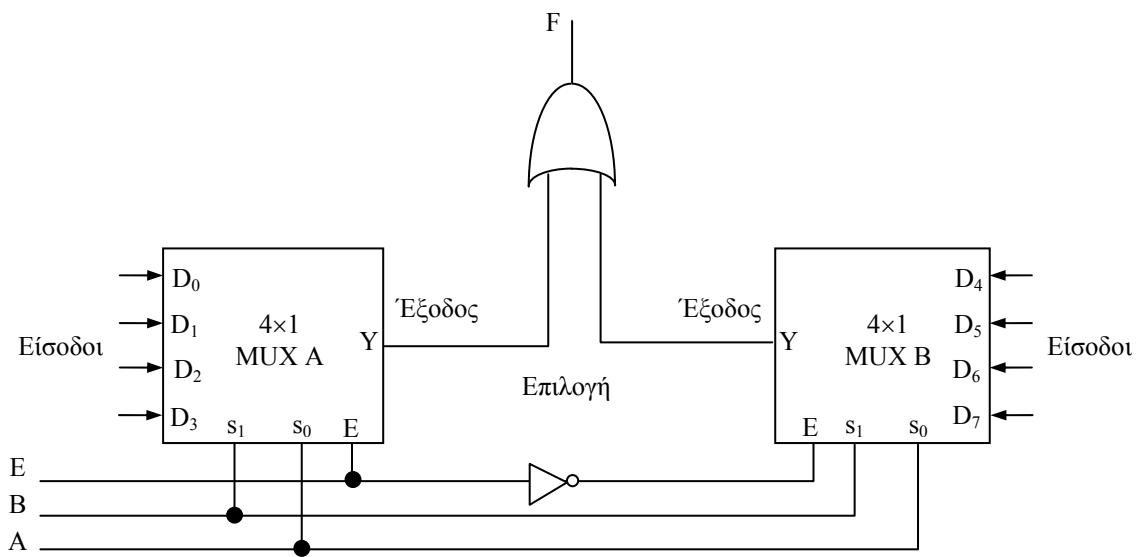


Σχ. 5.12. Πολυπλέκτης 4 - σε - 1 γραμμών.

Η συνάρτηση εξόδου γράφεται ως:

$$Y = \bar{s}_1 \bar{s}_0 I_0 + \bar{s}_1 s_0 I_1 + s_1 \bar{s}_0 I_2 + s_1 s_0 I_3$$

Όπως και οι αποκωδικοποιητές, έτσι και οι πολυπλέκτες που είναι διαθέσιμοι σαν chips, συχνά έχουν μία είσοδο επίτρησης για να ελέγχουμε τη λειτουργία τους. Όταν η είσοδος επίτρησης E είναι σε μία ορισμένη δυαδική κατάσταση, τότε οι έξοδοι “αποτρέπονται” (disabled - δηλαδή αδρανοποιούνται), ενώ, όταν είναι στην άλλη κατάσταση (της επίτρησης), το κύκλωμα λειτουργεί σαν κανονικός πολυπλέκτης. Η είσοδος επίτρησης μπορεί να χρησιμοποιηθεί για σκοπούς επέκτασης - για να σχηματίσουμε έναν πολυπλέκτη με περισσότερες εισόδους από δύο ή περισσότερα chips μικρότερων πολυπλεκτών όπως φαίνεται στο Σχ. 5.13.



Σχ. 5.13. Πολυπλέκτης 8 - σε - 1 γραμμών ο οποίος σχηματίζεται από δύο πολυπλέκτες 4 - σε - 1 γραμμών.

Παράδειγμα 5.4: Να υλοποιηθεί με τη βοήθεια πολυπλέκτη η συνάρτηση Boole τριών μεταβλητών:

$$F = \sum (1, 3, 5, 6)$$

(όπου οι αριθμοί στην παρένθεση αντιστοιχούν στο δεκαδικό ισοδύναμο του σχετικού ελαχιστόρου).

Λύση:

Στο παράδειγμα αυτό θα μελετήσουμε τη δυνατότητα υλοποίησης οποιασδήποτε συνάρτησης Boole με τη βοήθεια πολυπλέκτη. Είδαμε στην παράγραφο 5.3 ότι ένας αποκωδικοποιητής μπορεί να χρησιμοποιηθεί για την υλοποίηση μιας συνάρτησης Boole αν μεταχειριστούμε μια εξωτερική πύλη OR. Μια ματιά στον πολυπλέκτη του Σχ. 5.12 δείχνει ότι πρόκειται ουσιαστικά για έναν αποκωδικοποιητή με την πύλη OR ήδη διαθέσιμη. Εκείνους από τους ελαχιστόρους του αποκωδικοποιητή που χρειαζόμαστε μπορούμε να τους διαλέξουμε με τις γραμμές εισόδου. Τους ελαχιστόρους της συνάρτησης που θέλουμε να υλοποιήσουμε, τους επιλέγουμε κάνοντας τις αντίστοιχες εισόδους ίσες με 1, ενώ τους υπόλοιπους τους αδρανοποιούμε κάνοντας τις αντίστοιχες τους εισόδους 0. Αυτό οδηγεί σε μια μέθοδο υλοποίησης κάθε συνάρτησης Boole n μεταβλητών με έναν πολυπλέκτη 2^{n-1} -σε-1 γραμμών.

Η γενική διαδικασία για τη σύνθεση του κυκλώματος υλοποίησης μιας δοσμένης συνάρτησης Boole n μεταβλητών με χρήση πολυπλέκτη 2^{n-1} -σε-1 γραμμών έχει ως εξής:

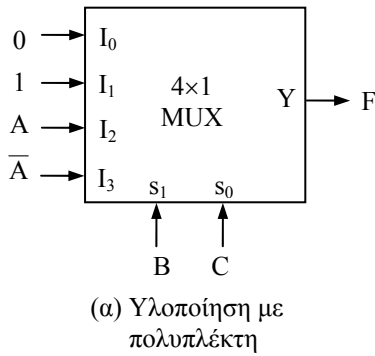
Πρώτα εκφράζουμε τη συνάρτηση ως άθροισμα ελαχιστόρων. Έστω ότι η σειρά των μεταβλητών είναι ABCD..., όπου A είναι η αριστερότερη μεταβλητή και BCD... είναι οι υπόλοιπες $n - 1$ μεταβλητές. Συνδέουμε τις $n - 1$ μεταβλητές στις γραμμές επιλογής του πολυπλέκτη, συνδέοντας το B στην περισσότερο σημαντική γραμμή επιλογής, το C στην επόμενη, κ.ο.κ. Ας θεωρήσουμε τώρα τη μεταβλητή A. Αφού αυτή η μεταβλητή είναι στην υψηλότερη θέση σημαντικότητας, θα εμφανίζεται με το συμπλήρωμά της στους ελαχιστόρους 0 ως $(2^n/2) - 1$, που είναι οι πρώτοι μισοί. Οι υπόλοιποι μισοί θα έχουν ασυμπλήρωτη τη μεταβλητή τους A. Για μια συνάρτηση λ.χ. τριών μεταβλητών A, B, C έχουμε οκτώ ελαχιστόρους. Η μεταβλητή A εμφανίζεται συμπληρωμένη στους 0 ως 3 και ασυμπλήρωτη στους 4 ως 7. Καταγράφουμε τις εισόδους του πολυπλέκτη και κάτω από αυτές όλους τους ελαχιστόρους σε δύο σειρές. Στην πρώτη σειρά γράφουμε όλους εκείνους, όπου το A είναι συμπληρωμένο, και στη δεύτερη αυτούς με το A ασυμπλήρωτο. Μετά σημειώνουμε με έναν κύκλο τους ελαχιστόρους που περιέχονται στη δοσμένη συνάρτηση και κοιτάμε τις στήλες μία-μία:

Εάν οι δύο ελαχιστόροι σε μία στήλη δεν είναι μέσα σε κύκλο, βάζουμε 0 στην αντίστοιχη είσοδο του πολυπλέκτη.

Εάν οι δύο ελαχιστόροι είναι μέσα σε κύκλο, βάζουμε 1 στην αντίστοιχη είσοδο του πολυπλέκτη.

Εάν ο κάτω ελαχιστόρος έχει σημειωθεί, αλλά όχι και ο επάνω, τότε βάζουμε A στην αντίστοιχη είσοδο του πολυπλέκτη.

Εάν είναι σημειωμένος ο επάνω ελαχιστόρος, αλλά όχι και ο κάτω, τότε τροφοδοτούμε το \bar{A} στην αντίστοιχη είσοδο του πολυπλέκτη.



Ελαχιστόρος	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

(β) Πίνακας αληθείας

	I ₀	I ₁	I ₂	I ₃
\bar{A}	0	1	2	3
A	4	5	6	7
	0	1	A	\bar{A}

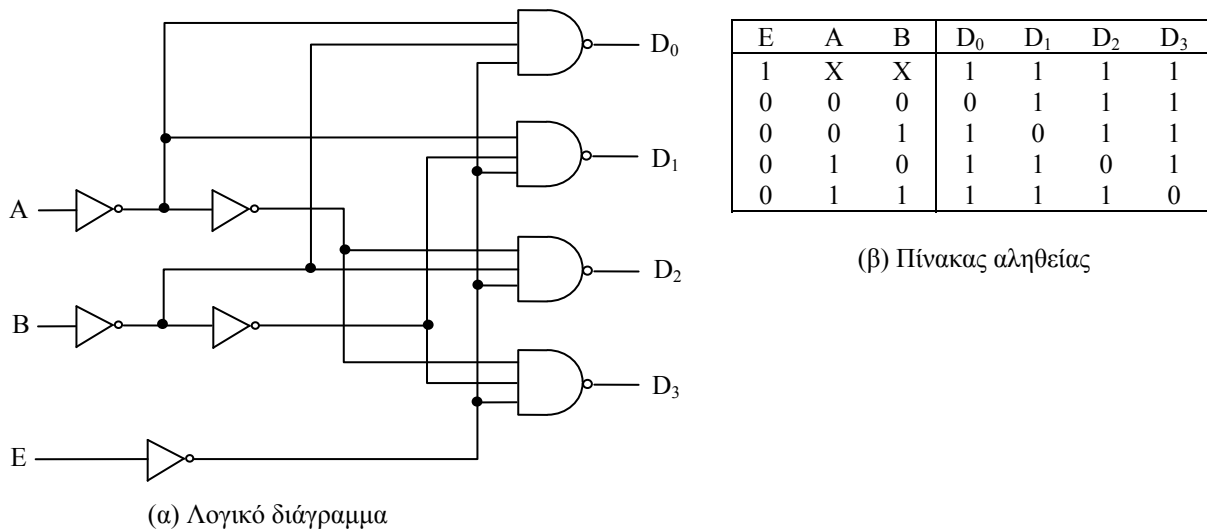
(γ) Πίνακας υλοποίησης

Σχ. 5.14. Υλοποίηση της $F = \sum(1,3,5,6)$ με πολυπλέκτη.

Σύμφωνα με την πιο πάνω διαδικασία και για το δοσμένο παράδειγμα, έχουμε κατ' αρχήν ότι η συνάρτηση F μπορεί να υλοποιηθεί με έναν πολυπλέκτη 4-σε-1, όπως φαίνεται στο Σχ. 5.14. Δύο από τις μεταβλητές, B και C, εφαρμόζονται στις γραμμές επιλογής με αυτή τη σειρά, δηλαδή, η B συνδέεται στο s_1 και η C στο s_0 . Οι εισοδοί του πολυπλέκτη είναι 0, 1, A και \bar{A} . Όταν $BC = 00$, η έξοδος $F = 0$ αφού $I_0 = 0$. Επομένως, και οι δύο ελαχιστόροι $\bar{A}\bar{B}\bar{C}$ και $A\bar{B}\bar{C}$ δίνουν έξοδο 0, αφού η έξοδος είναι 0 όταν $BC = 00$, ανεξάρτητα από την τιμή του A. Όταν $BC = 01$ η έξοδος $F = 1$, αφού $I_1 = 1$. Επομένως, και οι δύο ελαχιστόροι $\bar{A}\bar{B}C$ και $A\bar{B}C$ παράγουν έξοδο 1, αφού η έξοδος είναι 1 όταν $BC = 01$, ανεξάρτητα από την τιμή του A. Όταν $BC = 10$, επιλέγεται η είσοδος I_2 . Αφού το A συνδέεται σ' αυτή την είσοδο, η έξοδος θα είναι ίση με 1 μόνο για τον ελαχιστόρο $A\bar{B}\bar{C}$ αλλά όχι για τον $\bar{A}\bar{B}\bar{C}$, διότι όταν $\bar{A} = 1$, τότε $A = 0$, και αφού $I_2 = 0$, έχουμε ότι $F = 0$. Τελικά, όταν $BC = 11$, επιλέγεται η είσοδος I_3 . Αφού σ' αυτή την είσοδο συνδέεται το \bar{A} , η έξοδος θα είναι ίση με 1 μόνο για τον ελαχιστόρο $\bar{A}BC$, αλλά όχι και για τον ABC . Όλα αυτά συνοψίζονται στο Σχ. 5.14β που είναι ο πίνακας αληθείας της συνάρτησης που θέλουμε να υλοποιήσουμε.

5.6. Αποπολυπλέκτες (Demultiplexers)

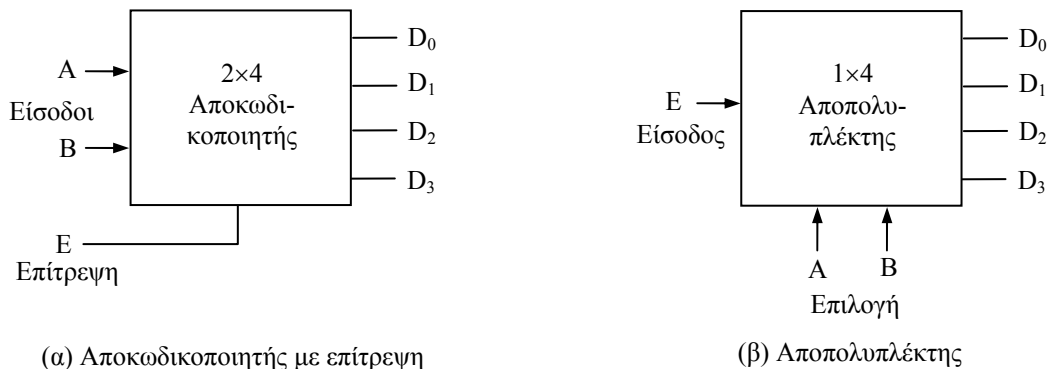
Ο αποπολυπλέκτης είναι ένα συνδυαστικό κύκλωμα που δέχεται πληροφορίες από μία απλή γραμμή και τις μεταβιβάζει σε μία από τις 2^n δυνατές γραμμές εξόδου. Η επιλογή μιας συγκεκριμένης γραμμής εξόδου γίνεται ανάλογα με τις τιμές των n γραμμών επιλογής. Ο αποκωδικοποιητής του Σχ. 5.15 μπορεί να λειτουργήσει και ως αποπολυπλέκτης, εάν θεωρήσουμε τη γραμμή E σαν είσοδο δεδομένων (πληροφοριών) και τις γραμμές A και B σαν γραμμές επιλογής. Αυτό φαίνεται στο Σχ. 5.16β. Η μοναδική μεταβλητή εισόδου E έχει στη διάθεσή της ένα δρόμο προς όλες (και τις τέσσερις) τις εξόδους, αλλά οι πληροφορίες που βρίσκονται στην είσοδο κατευθύνονται προς μόνο μία από τις γραμμές εξόδου, όπως καθορίζεται με τη δυαδική τιμή των δύο γραμμών επιλογής A και B.



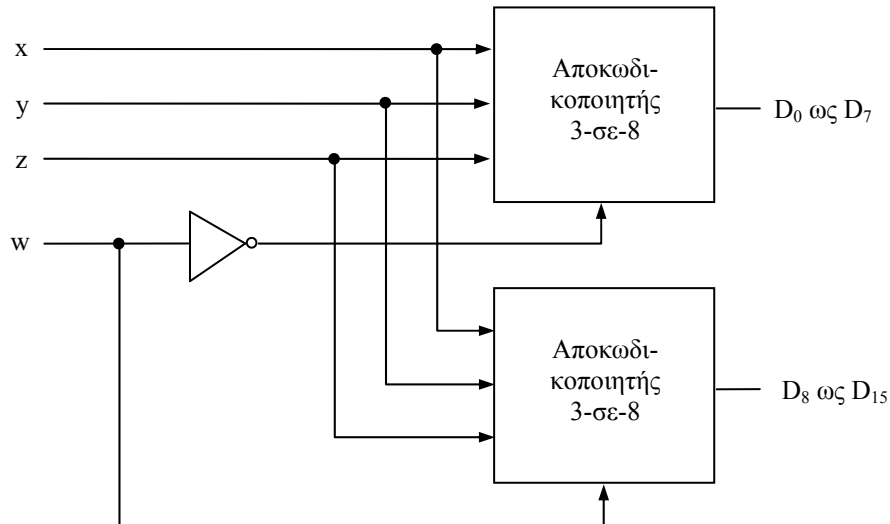
Σχ. 5.15. Αποκωδικοποιητής από 2 - σε - 4 γραμμές, με είσοδο επίτρεψης (E).

Αυτό μπορεί να επιβεβαιωθεί από τον πίνακα αληθείας του κυκλώματος, που φαίνεται στο Σχ 5.15β. Για παράδειγμα, εάν οι γραμμές επιλογής είναι $AB = 10$, η έξοδος D_2 θα είναι η ίδια όπως η είσοδος E, ενώ όλες οι άλλες εξόδους διατηρούνται στο 1. Επειδή οι λειτουργίες του αποκωδικοποιητή και του αποπολυπλέκτη λαμβάνονται από το ίδιο κύκλωμα, ένας αποκωδικοποιητής με είσοδο επίτρεψης ονομάζεται συνήθως αποκωδικοποιητής / αποπολυπλέκτης. Το κύκλωμα αυτό γίνεται αποπολυπλέκτης χάρη στην είσοδο επίτρεψης. Ο αποκωδικοποιητής αυτός καθαυτός μπορεί να χρησιμοποιεί πύλες AND, NAND ή NOT.

Τα chips αποκωδικοποιητή/αποπολυπλέκτη μπορούν να συνδεθούν μεταξύ τους και να σχηματίσουν ένα μεγαλύτερο κύκλωμα αποκωδικοποιητή. Το Σχ. 5.17 δείχνει δύο αποκωδικοποιητές 3 - σε - 8, με εισόδους επίτρεψης που συνδέονται για να σχηματίσουν έναν αποκωδικοποιητή 4 - σε - 16. Όταν το $w = 0$, “επιτρέπεται” η λειτουργία του επάνω αποκωδικοποιητή και “αποτρέπεται” η λειτουργία του κάτω. Οι εξόδους του κάτω αποκωδικοποιητή είναι όλες 0 και οι οκτώ επάνω εξόδους παράγουν τους ελαχιστόρους από 0000 μέχρι 0111. Όταν το $w = 1$, οι συνθήκες επίτρεψης αντιστρέφονται: οι εξόδους του κάτω αποκωδικοποιητή παράγουν τους ελαχιστόρους 1000 ως 1111, ενώ οι εξόδους του πάνω αποκωδικοποιητή είναι όλες 0. Αυτό το παράδειγμα δείχνει τη χρησιμότητα των εισόδων επίτρεψης: δίνουν ένα βολικό τρόπο για τη σύνδεση δύο ή περισσότερων chips, με σκοπό την επέκταση της λειτουργίας τους σε περισσότερες εισόδους και εξόδους.



Σχ. 5.16. Σχηματικά διαγράμματα για το κύκλωμα του Σχ. 5.15.

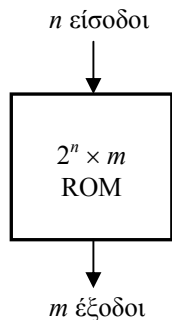


Σχ. 5.17. Ένας αποκωδικοποιητής 4-σε-16, υλοποιημένος από δύο αποκωδικοποιητές 3-σε-8.

5.7. Μνήμη ROM

Όπως είδαμε, ένας αποκωδικοποιητής παράγει τους 2^n ελαχιστόρους των n μεταβλητών εισόδου. Με τη βοήθεια πυλών OR αθροίζουμε τους ελαχιστόρους της συνάρτησης που θέλουμε να υλοποιήσουμε. Η μνήμη-ανάγνωσης-μόνο (Read Only Memory – ROM) είναι μια διάταξη που περιλαμβάνει τον αποκωδικοποιητή και τις πύλες OR μέσα στο ίδιο chip. Οι συνδέσεις μεταξύ των εξόδων του αποκωδικοποιητή και των εισόδων των πυλών OR μπορούν να καθοριστούν με “προγραμματισμό” της ROM. Η ROM χρησιμοποιείται πολύ συχνά για την υλοποίηση περίπλοκων συνδυαστικών κυκλωμάτων με ένα μόνο chip, καταργώντας έτσι την ανάγκη για όλα τα καλώδια διασύνδεσης.

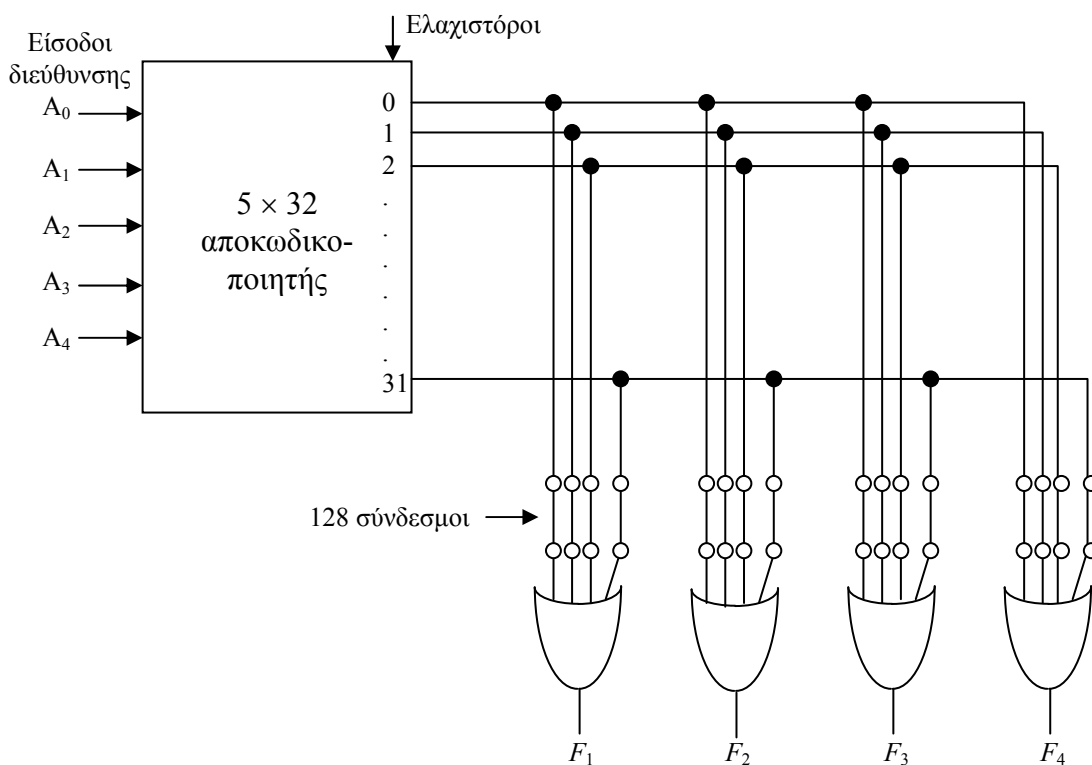
Η ROM είναι βασικά μια συσκευή μνήμης στην οποία αποθηκεύεται ένα σταθερό σύνολο δυαδικών πληροφοριών οι οποίες καθορίζονται από το χρήστη και μετά ενσωματώνονται στο chip ώστε να σχηματιστούν οι απαραίτητες διασυνδέσεις. Οι ROM κατασκευάζονται με ειδικούς εσωτερικούς συνδέσμους τους οποίους μπορούμε αργότερα να κάψουμε ή να σπάσουμε. Μόλις οριστικοποιηθεί ένα τέτοιο σύνολο διασυνδέσεων για μια ROM, αυτό παραμένει σταθερό ακόμα και όταν “κοπεί το ρεύμα”, δηλαδή διακοπεί η τροφοδοσία.



Σχ. 5.18. Σχηματικό διάγραμμα ROM.

Το Σχ. 5.18 δείχνει το συμβολικό διάγραμμα μιας ROM. Έχει n γραμμές εισόδου και m γραμμές εξόδου. Κάθε συνδυασμός των bits εισόδου λέγεται “διεύθυνση” (address). Κάθε συνδυασμός των bits εξόδου λέγεται “λέξη” (word). Ο αριθμός των bits κάθε λέξης ισούται με τον αριθμό των γραμμών εξόδου, δηλαδή m . Μία διεύθυνση είναι βασικά ένας δυαδικός αριθμός που δείχνει έναν από τους ελαχιστόρους των n μεταβλητών. Ο αριθμός των δυνατών διαφορετικών διευθύνσεων για n μεταβλητές εισόδου είναι 2^n . Μία λέξη εξόδου μπορεί να επιλεγεί με μία μοναδική διεύθυνση και αφού υπάρχουν 2^n διαφορετικές διευθύνσεις σε μία ROM, υπάρχουν 2^n διαφορετικές λέξεις αποθηκευμένες στο chip. Η λέξη που βγαίνει στις γραμμές εξόδου κάθε δοσμένη στιγμή εξαρτάται από τη διεύθυνση που εφαρμόζεται στις γραμμές εισόδου. Μια ROM χαρακτηρίζεται από τον αριθμό των λέξεων 2^n και από τον αριθμό των bits σε κάθε λέξη, m .

Έστω μία ROM 32×8 . Η μονάδα αποτελείται από 32 λέξεις των 8 bits η καθεμία. Αυτό σημαίνει ότι υπάρχουν 8 γραμμές εξόδου και ότι υπάρχουν 32 διαφορετικές λέξεις αποθηκευμένες σ’ αυτή τη ROM, η καθεμία από τις οποίες μπορεί να εμφανιστεί στις γραμμές εξόδου. Η συγκεκριμένη λέξη, που θα εμφανιστεί στις γραμμές εξόδου, καθορίζεται από τις 5 γραμμές εισόδου. Υπάρχουν μόνο 5 εισοδοί σε μία ROM 32×8 , διότι $2^5=32$ και με 5 μεταβλητές μπορούμε να ορίσουμε 32 διευθύνσεις ή ελαχιστόρους.

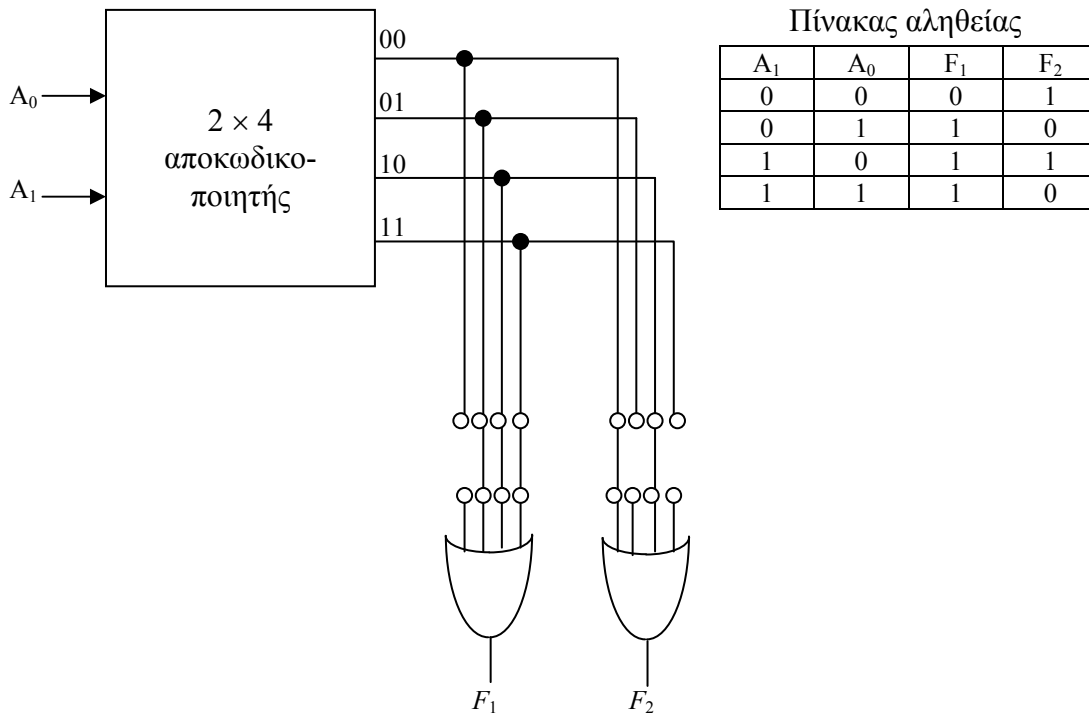


Σχ. 5.19. Εσωτερική οργάνωση μιας ROM.

Ο αριθμός των λέξεων σε μία ROM καθορίζεται από το γεγονός ότι χρειάζονται n γραμμές εισόδου για να ξεχωρίσουν 2^n λέξεις. Οι ROM περιγράφονται συχνά με το συνολικό αριθμό bits που περιέχουν, τα οποία είναι $2^n \times m$. Για παράδειγμα, μία ROM των 2048 bits μπορεί να είναι οργανωμένη σε 512 λέξεις των 4 bits η καθεμία. Αυτό σημαίνει ότι το chip αυτό έχει 4 γραμμές εξόδου και 9 γραμμές εισόδου για να ορίσει

$2^9=512$ λέξεις. Ο συνολικός αριθμός των bits που είναι αποθηκευμένα στη μονάδα είναι $512 \times 4=2048$.

Εσωτερικά, η ROM είναι ένα συνδυαστικό κύκλωμα με πύλες AND συνδεδεμένες σαν αποκωδικοποιητής και με τόσες πύλες OR, όσες και οι εξοδοί της ROM. Το Σχ. 5.19 δείχνει την εσωτερική κατασκευή μιας ROM 32×4 . Οι 5 μεταβλητές εισόδου αποκωδικοποιούνται σε 32 γραμμές με τη βοήθεια 32 πυλών AND και 5 αντιστροφών. Κάθε μία από τις 32 διευθύνσεις επιλέγει μία και μόνο έξοδο του αποκωδικοποιητή. Οι 32 έξοδοι του αποκωδικοποιητή συνδέονται με “συνδέσμους” σε κάθε πύλη OR.



Σχ. 5.20. Υλοποίηση συνδυαστικού κυκλώματος με ROM.

Ας δούμε ένα συγκεκριμένο παράδειγμα. Ο πίνακας αληθείας του Σχ. 5.20 καθορίζει ένα συνδυαστικό κύκλωμα με 2 εισόδους και 2 εξόδους. Οι συναρτήσεις Boole εκφράζονται ως άθροισμα ελαχιστόρων:

$$F_1(A_1, A_0) = \Sigma(1, 2, 3)$$

$$F_2(A_1, A_0) = \Sigma(0, 2)$$

Όταν ένα συνδυαστικό κύκλωμα υλοποιείται με ROM, οι συναρτήσεις πρέπει να εκφράζονται ως άθροισμα ελαχιστόρων. Εάν απλοποιούσαμε τις συναρτήσεις εξόδου θα βρίσκαμε ότι το κύκλωμα χρειάζεται μόνο μια πύλη OR κι έναν αντιστροφέα. Φυσικά, το παρόν κύκλωμα είναι πολύ απλό. Το πλεονέκτημα της ROM είναι στην υλοποίηση πολύπλοκων κυκλωμάτων.

Η ROM που θα υλοποιήσει τις παραπάνω συναρτήσεις πρέπει να έχει 2 εισόδους και 2 εξόδους, άρα το μέγεθός της θα είναι 4×2 . Αφού γνωρίζουμε τους ελαχιστόρους των συναρτήσεων μπορούμε εύκολα να προσδιορίσουμε ποιοι από τους 8 συνδέσμους πρέπει να κοπούν και ποιοι να παραμείνουν.

Το προηγούμενο παράδειγμα περιγράφει τη γενική διαδικασία υλοποίησης οποιουδήποτε συνδυαστικού κυκλώματος με ROM. Πρώτα βρίσκουμε το μέγεθος της

απαιτούμενης ROM από τον αριθμό των εισόδων και εξόδων, μετά παίρνουμε τον πίνακα αληθείας της ROM ο οποίος χρησιμοποιείται κατευθείαν για τον προγραμματισμό, χωρίς άλλη αλλαγή ή απλοποίηση. Στην πράξη, όταν σχεδιάζει κανείς ένα κύκλωμα με τη βοήθεια της ROM, δεν χρειάζεται να σχεδιάσει τις εσωτερικές συνδεσμολογίες. Το μόνο που χρειάζεται να κάνει είναι να καθορίσει το συγκεκριμένο τύπο ROM (π.χ. τον κωδικό της) και να δώσει τον πίνακα αληθείας της ο οποίος περιέχει όλες τις πληροφορίες προγραμματισμού της ROM, χωρίς να χρειάζεται εσωτερικό λογικό διάγραμμα.

Δυστυχώς, τα δεδομένα που αποθηκεύονται σε μια ROM εισάγονται κατά τη διαδικασία κατασκευής και δεν μπορούν να μεταβληθούν στη συνέχεια. Ένας πολύ βολικός τύπος μνημών ROM είναι οι “σβηνώμενες προγραμματιζόμενες ROM” (Erasable Programmable Read-Only Memory, EPROM) το περιεχόμενο των οποίων μπορεί εύκολα να προγραμματιστεί και αποθηκευθεί και μπορεί επίσης να αλλάξει εάν χρειαστεί.

Παράδειγμα 5.5

Να σχεδιαστούν οι ακόλουθες λογικές συναρτήσεις χρησιμοποιώντας μόνο ένα ολοκληρωμένο κύκλωμα 74155 και ένα ολοκληρωμένο 7400:

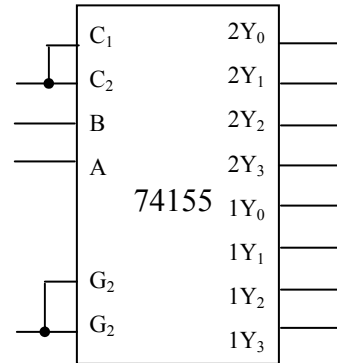
$$F_1 = \bar{x} \cdot \bar{z}$$

$$F_2 = x \cdot \bar{y} + x \cdot \bar{y} \cdot \bar{z}$$

$$F_3 = x \cdot y \cdot \bar{z} + \bar{x} \cdot \bar{y} \cdot z$$

Το 74155 μπορεί να λειτουργήσει σαν διπλός αποκωδικοποιητής 2-σε-4 ή σαν απλός αποκωδικοποιητής 3-σε-8, οπότε οι εισοδοί C₁ και C₂, καθώς επίσης και οι εισοδοί επέτρησης G₁ και G₂ πρέπει να είναι βραχυκυκλωμένες μεταξύ τους όπως φαίνεται στο σχήμα.

Ο πίνακας αλήθειας του σχήματος περιγράφει τη λειτουργία του 74155. Επειδή εσωτερικά το ολοκληρωμένο αυτό χρησιμοποιεί πύλες NAND, η έξοδος που επιλέγεται γίνεται 0, ενώ οι υπόλοιπες μένουν στο 1. Λάβετε υπ' όψη σας αυτή την ιδιοτυπία όταν κάνετε τη σχεδίασή σας.



Είσοδοι				Έξοδοι							
G	C	B	A	2Y ₀	2Y ₁	2Y ₂	2Y ₃	1Y ₀	1Y ₁	1Y ₂	1Y ₃
1	X	X	X	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	0

Λύση

Γράφουμε τις 3 συναρτήσεις σαν άθροισμα γινομένων (ελαχιστόρων). Θα έχουμε:

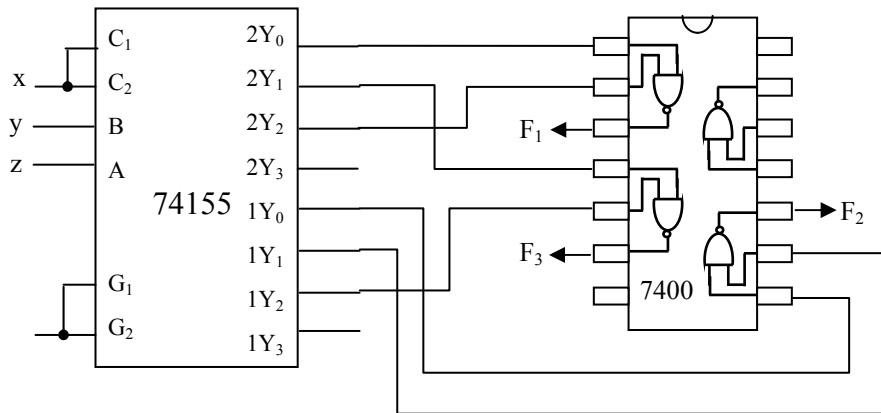
$$F_1 = \bar{x} \bar{z} = \bar{x} \bar{z} (y + \bar{y}) = \bar{x} y \bar{z} + \bar{x} \bar{y} \bar{z}.$$

$$F_2 = x \bar{y} + x \bar{y} \bar{z} = x \bar{y} (z + \bar{z}) + x \bar{y} \bar{z} = x \bar{y} z + x \bar{y} \bar{z}.$$

$$F_3 = x y \bar{z} + \bar{x} \bar{y} z.$$

Η F_1 αληθεύει στα $2Y_2$ και $2Y_0$.
 Η F_2 αληθεύει στα $1Y_1$ και $1Y_0$.
 Η F_3 αληθεύει στα $1Y_2$ και $2Y_1$.

Άρα τελικά το λογικό κύκλωμα θα είναι:

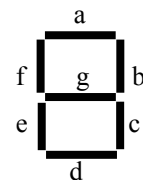


Παράδειγμα 5.6

Ένας αποκωδικοποιητής σε 7 τμήματα είναι ένα συνδυαστικό κύκλωμα (το ολοκληρωμένο 7447) που μετατρέπει την πληροφορία 4 γραμμών εισόδου που χρειάζονται για τη δυαδική αναπαράσταση ενός μονοψήφιου δεκαδικού αριθμού σε 7 γραμμές εξόδου οι οποίες εμφανίζουν τον αντίστοιχο δεκαδικό αριθμό. Ο πίνακας αλήθειας αυτού του αποκωδικοποιητή φαίνεται στο ακόλουθο σχήμα. Αν κάποια έξοδος (a-g) είναι 0 τότε το αντίστοιχο τμήμα του ενδείκτη 7 τμημάτων είναι σβηστό, ενώ αν είναι 1 ανάβει.

- A) Να δοθεί η συνάρτηση που παρουσιάζει την έξοδο a.
- B) Να υλοποιηθεί με τη βοήθεια αποκωδικοποιητή 4-σε-16 και ό,τι πύλες θέλετε.
- Γ) Για την απεικόνιση του αριθμού 2 ποιος συνδυασμός πρέπει να συμβεί στην είσοδο του αποκωδικοποιητή και ποιες έξοδοι είναι ενεργοποιημένες;
- Δ) Να υλοποιηθεί η συνάρτηση της εξόδου a με τη βοήθεια πολυπλέκτη.

D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1
1	0	1	0	0	0	0	1	1	0	1
1	0	1	1	0	0	1	1	0	0	1
1	1	0	0	0	1	0	0	0	1	1
1	1	0	1	1	0	0	1	0	1	1
1	1	1	0	0	0	0	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0

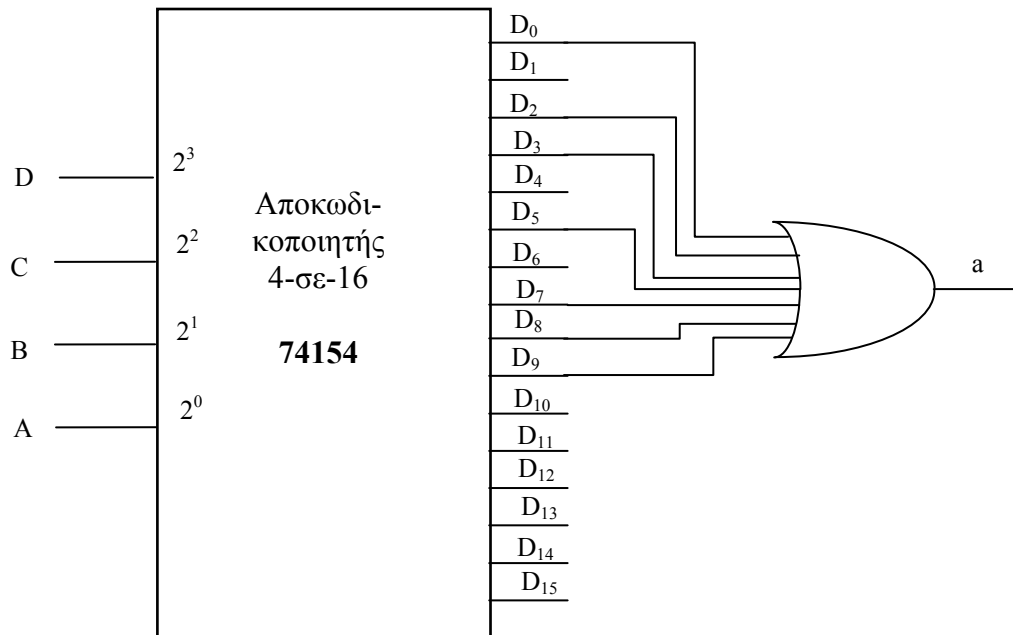


Λύση

A) Από τον πίνακα αληθείας της εξόδου a και λαμβάνοντας υπόψη μόνο τις 10 πρώτες γραμμές (συνδυασμούς) εισόδων θα έχουμε:

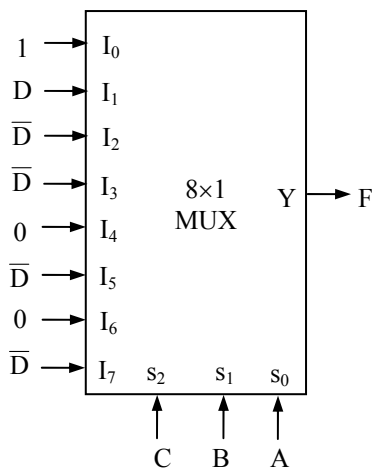
$$a = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} B \bar{C} \bar{D} + A \bar{B} \bar{C} \bar{D} + A \bar{B} C \bar{D} + A B C \bar{D} + \bar{A} \bar{B} \bar{C} D + A \bar{B} \bar{C} D.$$

B) Ως γνωστόν, ο αποκωδικοποιητής υλοποιεί σε κάθε μία έξοδό του έναν ελαχιστόρο των γραμμών εισόδου. Κάθε δε συνάρτηση Boole γράφεται ως άθροισμα των ελαχιστόρων της. Άρα, οδηγώντας σε μια πύλη OR τους ελαχιστόρους της συνάρτησης α υλοποιούμε το λογικό κύκλωμά της.



Γ) Για την απεικόνιση του αριθμού 2 πρέπει να είναι ενεργοποιημένες στην έξοδο οι γραμμές: a, b, g, e, d. Ο συνδυασμός που υλοποιεί την ενεργοποίηση αυτών των γραμμών είναι ο συνδυασμός που αντιστοιχεί στον ελαχιστόρο D₂.

Δ) Σύμφωνα με τη διαδικασία υλοποίησης μιας συνάρτησης με πολυπλέκτη που παρουσιάστηκε στο Παράδειγμα 5.4 θα χρησιμοποιήσουμε πολυπλέκτη 2³×1=8×1 δεδομένου ότι η προς υλοποίηση συνάρτηση είναι συνάρτηση 4 μεταβλητών.



(α) Υλοποίηση με πολυπλέκτη

	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇
\bar{D}	0	1	2	3	4	5	6	7
D	8	9	10	11	12	13	14	15
	1	D	\bar{D}	\bar{D}	0	\bar{D}	0	\bar{D}

(β) Πίνακας υλοποίησης

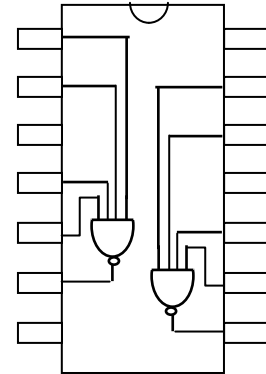
Παράδειγμα 5.7

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

Δίνεται ο πίνακας αλήθειας που φαίνεται αριστερά.

A) Να δοθεί η συνάρτηση F σαν άθροισμα γινομένων και σαν γινόμενο αθροισμάτων.

B) Να σχεδιάσετε το λογικό διάγραμμα της απλοποιημένης συνάρτησης με χρήση ενός 7420 του οποίου η διάταξη των πυλών παρουσιάζεται παραπλεύρως, καθώς και ενός 7400 του οποίου η περιγραφή έχει προηγηθεί.



Γ) Να σχεδιαστεί η συνάρτηση με χρήση πολυπλέκτη.

Λύση

A) Με βάση τον πίνακα αληθείας γράφουμε τη συνάρτηση ως άθροισμα ελαχιστόρων και ως γινόμενο μεγιστόρων.

$$F = \bar{x} \bar{y} \bar{z} \bar{w} + \bar{x} \bar{y} z \bar{w} + \bar{x} y \bar{z} \bar{w} + \bar{x} y \bar{z} w + \bar{x} y z \bar{w} + \bar{x} y z w + x y \bar{z} w + x y z w .$$

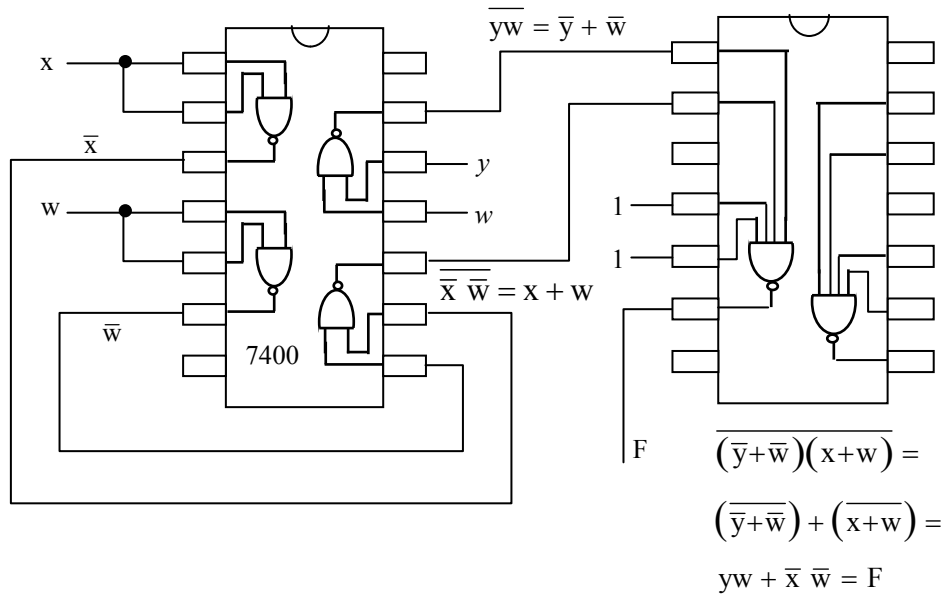
$$F = (x + y + z + \bar{w})(x + y + \bar{z} + \bar{w})(\bar{x} + y + z + w)(\bar{x} + y + z + \bar{w})(\bar{x} + y + \bar{z} + w)(\bar{x} + y + \bar{z} + \bar{w})(\bar{x} + \bar{y} + z + w)(\bar{x} + \bar{y} + \bar{z} + w).$$

B) Απλοποιούμε τη δοσμένη συνάρτηση με τη βοήθεια του χάρτη Karnaugh.

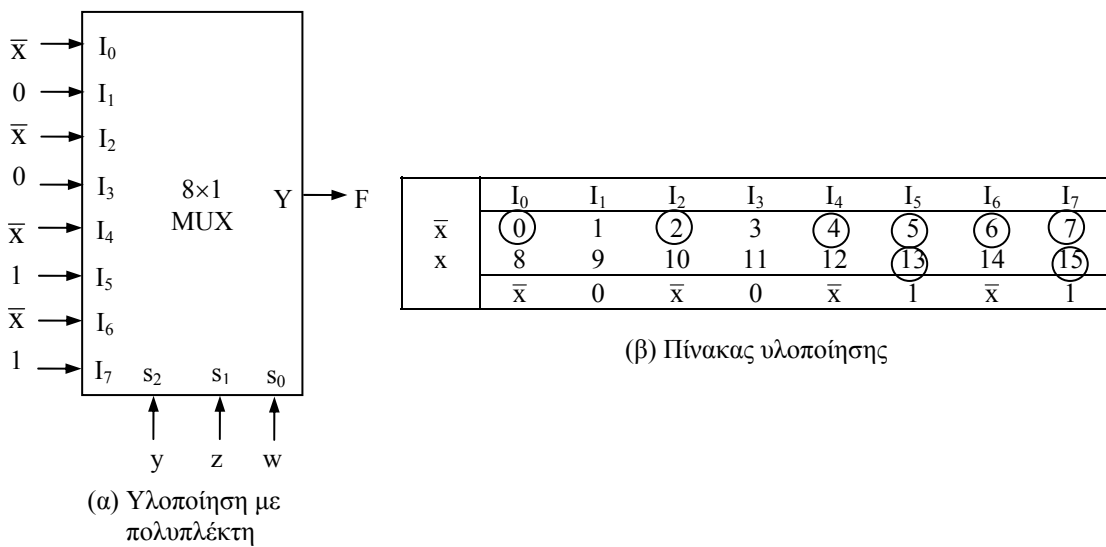
xy \ zw	00	01	11	10
00	1			1
01	1	1	1	1
11		1	1	
10				

Η απλοποιημένη συνάρτηση είναι:

$$F = yw + \bar{x} \bar{w} .$$



Γ) Σύμφωνα με τη διαδικασία υλοποίησης μιας συνάρτησης με πολυπλέκτη που παρουσιάστηκε στο Παράδειγμα 5.4 θα χρησιμοποιήσουμε πολυπλέκτη $2^3 \times 1 = 8 \times 1$ δεδομένου ότι η προς υλοποίηση συνάρτηση είναι συνάρτηση 4 μεταβλητών.



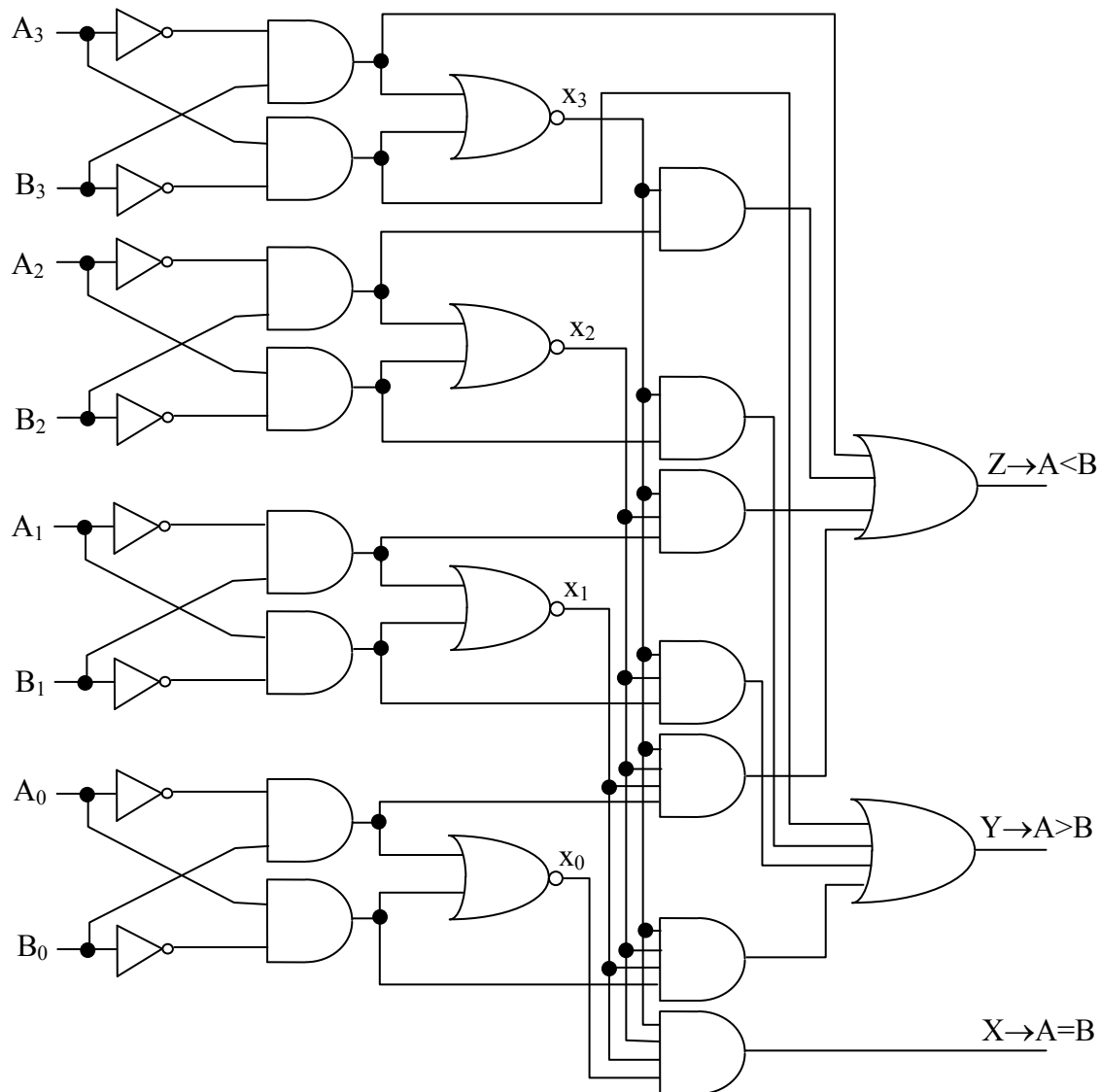
Παράδειγμα 5.8

Σχεδιάστε ένα συνδυαστικό κύκλωμα που συγκρίνει 2 αριθμούς 4-bit και αποφασίζει αν οι αριθμοί είναι ίσοι μεταξύ τους, ή είναι μικρότερος ή μεγαλύτερος ο ένας σε σχέση με τον άλλον.

Λύση

Έστω οι 2 αριθμοί 4-bit, A και B. Ο συγκριτής θα έχει 3 δυαδικές εξόδους X, Y και Z που θα παριστάνουν τα αποτελέσματα της σύγκρισης $A=B$, $A>B$ και $A<B$,

αντίστοιχα. Ανάλογα με το σχετικό μέγεθος των αριθμών, η αντίστοιχη έξοδος X, Y ή Z θα δείχνει λογικό 1. Για παράδειγμα, οι δύο αριθμοί A και B που τους παριστάνουμε ως $(A_3A_2A_1A_0)$ και $(B_3B_2B_1B_0)$, θα είναι ίσοι (δηλαδή $X=1$) εάν, $A_3=B_3, A_2=B_2, A_1=B_1, A_0=B_0$. Για να προσδιορίσουμε αν ο A είναι μεγαλύτερος ή μικρότερος από τον B, εξετάζουμε το σχετικό μέγεθος των ζευγών bit αντίστοιχης βαρύτητας, ξεκινώντας από το πλέον σημαντικό bit. Η σύγκριση συνεχίζεται διαδοχικά στο αμέσως επόμενο ζεύγος bit χαμηλότερης σημαντικότητας εφόσον το ζεύγος bit της σημαντικότητας που εξετάσαμε είναι ίσα μεταξύ τους. Η σύγκριση συνεχίζεται μέχρι να βρεθεί ένα ζεύγος άνισων μεταξύ τους bit. Στο ζεύγος αυτό, εάν $A_i=1$ και $B_i=0$, τότε $A>B$, ενώ εάν $A_i=0$ και $B_i=1$, τότε $A<B$.



Σχ. 5.21. Συγκριτής δύο αριθμών 4-bit.

Για παράδειγμα, ο έλεγχος της κατάστασης $A>B$ μέσω της συνάρτησης Y, δίνει 1 μόνο όταν $A_i=1$ και $B_i=0$ με $A_j=B_j$ για $j>i$, δηλαδή ελέγχει από αριστερά προς τα δεξιά ένα προς ένα τα bits. Αντίστοιχα γίνεται ο έλεγχος της κατάστασης $A<B$, μέσω της συνάρτησης Z, η οποία δίνει 1 μόνο όταν $A_i=0$ και $B_i=1$ με $A_j=B_j$ για $j>i$. Οι εκφράσεις Boole που υλοποιούν τις συναρτήσεις X, Y και Z δίνονται από τις παρακάτω εξισώσεις:

$$X = x_3 \cdot x_2 \cdot x_1 \cdot x_0 \quad \text{όπου} \quad x_i = A_i \cdot B_i + \bar{A}_i \cdot \bar{B}_i$$

$$Y = A_3 \cdot \bar{B}_3 + x_3 \cdot A_2 \cdot \bar{B}_2 + x_3 \cdot x_2 \cdot A_1 \cdot \bar{B}_1 + x_3 \cdot x_2 \cdot x_1 \cdot A_0 \cdot \bar{B}_0$$

$$Z = \bar{A}_3 \cdot B_3 + x_3 \cdot \bar{A}_2 \cdot B_2 + x_3 \cdot x_2 \cdot \bar{A}_1 \cdot B_1 + x_3 \cdot x_2 \cdot x_1 \cdot \bar{A}_0 \cdot B_0$$

Ας εξετάσουμε την εξίσωση για το X. Η μεταβλητή x_3 θα είναι 1 μόνο όταν η A_3 και η B_3 είναι ίσες. Ομοίως, οι συνθήκες για να είναι ίσες με 1 οι μεταβλητές x_2 , x_1 , και x_0 , είναι να είναι ίσες οι A_2 και η B_2 , οι A_1 και η B_1 , και οι A_0 και η B_0 , αντίστοιχα. Πολλαπλασιάζοντας μέσω πύλης AND τις x_3 , x_2 , x_1 , και x_0 , διασφαλίζουμε ότι η X θα είναι 1 όταν και οι τέσσερις μεταβλητές x_3 , x_2 , x_1 , και x_0 είναι 1. Άρα $X=1$, σημαίνει ότι $A=B$.

Παρόμοια, μπορούμε να δούμε ότι οι εξισώσεις για τις συναρτήσεις Y και Z, ελέγχουν τις περιπτώσεις $A>B$ και $A<B$ αντίστοιχα. Στο Σχ. 5.21 φαίνεται το λογικό διάγραμμα του συγκριτή.

Παράδειγμα 5.9

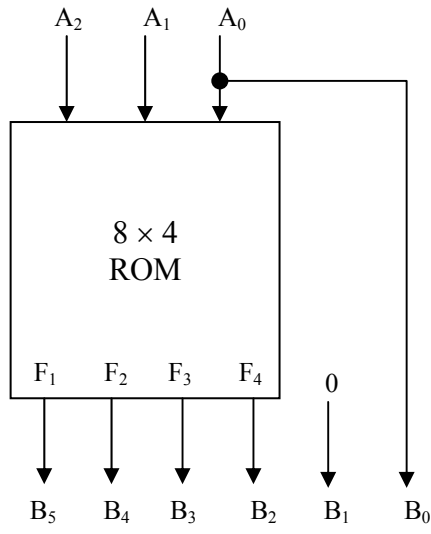
Σχεδιάστε ένα συνδυαστικό κύκλωμα με ROM. Το κύκλωμα πρέπει να δέχεται έναν αριθμό των 3 bits και να παράγει ένα δυαδικό αριθμό στην έξοδο, ίσο με το τετράγωνο του αριθμού εισόδου.

Λύση

Στο παρακάτω σχήμα δίνεται ο πίνακας αληθείας του συνδυαστικού κυκλώματος.

Είσοδοι			Έξοδοι						Δεκαδικός
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

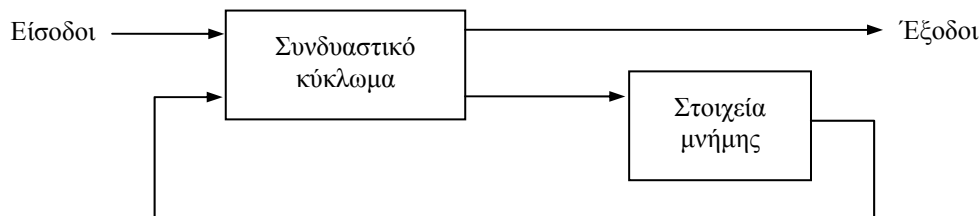
Χρειαζόμαστε 3 εισόδους και 6 εξόδους. Παρατηρούμε ότι η έξοδος B_0 είναι πάντα ίση με την είσοδο A_0 , κι έτσι δεν χρειάζεται ROM για να παράγουμε την B_0 . Επίσης, η έξοδος B_1 είναι πάντα 0. Άρα τελικά η ROM χρειάζεται να παράγει μόνο τις 4 εξόδους. Έτσι, η ROM που χρειαζόμαστε πρέπει να έχει 3 εισόδους και 4 εξόδους. Οι 3 είσοδοι ορίζουν 8 λέξεις και άρα το μέγεθος της ROM πρέπει να είναι 8×4 . Η υλοποίηση της ROM και ο πίνακας αληθείας της φαίνονται στο παρακάτω σχήμα.



A ₂	A ₁	A ₀	F ₁	F ₂	F ₃	F ₄
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

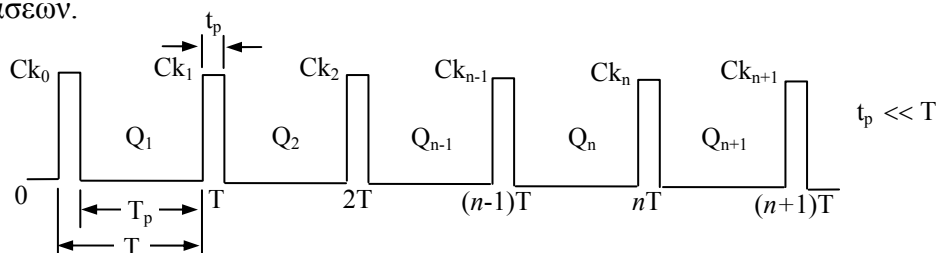
6. ΑΚΟΛΟΥΘΙΑΚΑ ΚΥΚΛΩΜΑΤΑ

Τα ψηφιακά κυκλώματα που εξετάσαμε μέχρι τώρα ήταν όλα συνδυαστικά, δηλαδή οι εξοδοί τους την κάθε στιγμή εξαρτώνται αποκλειστικά και μόνο από τις τιμές των εισόδων τους εκείνης της ίδιας στιγμής. Κάθε ψηφιακό σύστημα περιέχει - συνήθως - συνδυαστικά κυκλώματα, όμως τα περισσότερά τους επίσης περιέχουν στοιχεία μνήμης, τα οποία και κάνουν το όλο σύστημα να είναι “ακολουθιακό” (sequential).



Σχ. 6.1. Σχηματικό διάγραμμα ακολουθιακού κυκλώματος.

Στο Σχ. 6.1 βλέπουμε ένα σχηματικό διάγραμμα ενός ακολουθιακού κυκλώματος. Αποτελείται από ένα συνδυαστικό κύκλωμα συνδεδεμένο με στοιχεία μνήμης σε ένα σχηματισμό βρόχου ανατροφοδότησης. Τα στοιχεία μνήμης είναι διατάξεις που μπορούν να αποθηκεύουν δυαδικές πληροφορίες. Στην κάθε χρονική στιγμή, οι δυαδικές πληροφορίες που είναι αποθηκευμένες στα στοιχεία μνήμης ενός ακολουθιακού κυκλώματος, αποτελούν την “κατάστασή” του (state). Το ακολουθιακό κύκλωμα δέχεται πληροφορίες από τις εξωτερικές του εισόδους. Αυτές οι εισοδοί, μαζί με την παρούσα κατάσταση των στοιχείων μνήμης, καθορίζουν τις τιμές των εξόδων. Ακόμη, αυτές καθορίζουν το πώς θα αλλάξει η κατάσταση των στοιχείων μνήμης. Έτσι, απ’ αυτό το διάγραμμα βλέπουμε ότι οι εξοδοί ενός ακολουθιακού κυκλώματος είναι συναρτήσεις όχι μόνο των εισόδων του, αλλά και της παρούσας κατάστασης των στοιχείων μνήμης του. Η επόμενη κατάσταση αυτών των στοιχείων μνήμης είναι κι αυτή συνάρτηση τόσο των εισόδων όσο και της παρούσας κατάστασης. Έτσι, για την περιγραφή των ακολουθιακών κυκλωμάτων δεν φτάνει να κοιτάξει κανείς μόνο τις παρούσες τιμές των εισόδων και εξόδων - πρέπει να μελετήσει μια ολόκληρη χρονική ακολουθία (sequence) εισόδων, εξόδων και καταστάσεων.



Σχ. 6.2. Παλμοσειρά ρολογιού.

Τα περισσότερα λογικά κυκλώματα λειτουργούν συγχρονισμένα με μια παλμοσειρά περιόδου T (Clock) όπως φαίνεται στο Σχ. 6.2. Η διάρκεια κάθε παλμού t_p θεωρείται πολύ μικρή σε σχέση με την περίοδο T της παλμοσειράς. Το κύκλωμα παραγωγής των παλμών αυτών είναι ένας ελεύθερος πολυδονητής και ονομάζεται ρολόι ή Clock (Ck), υπάρχει δε στο εμπόριο ως ολοκληρωμένο κύκλωμα.

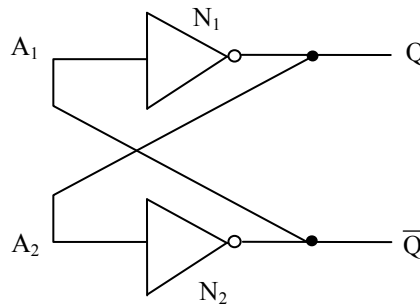
Στα ακολουθιακά κυκλώματα η αλλαγή καταστάσεως είναι δυνατή μόνο μετά την εφαρμογή σε αυτά του παλμού ρολογιού. Το χρονικό διάστημα μεταξύ δύο παλμών T

- $t_p = T_p$ διατίθεται επομένως για να εκφρασθεί στην έξοδο της πύλης η κατάσταση Q_n μετά τον $(n - 1)$ παλμό. Το σύστημα αυτό στο οποίο οι διαδοχικές καταστάσεις Q_1, \dots, Q_n εμφανίζονται υποχρεωτικά κατά σειρά και μέσα στα χρονικά διαστήματα T_1, \dots, T_n αντίστοιχα, και επίσης η κατάσταση Q_{n+1} εξαρτάται από την κατάσταση Q_n η οποία προηγήθηκε, ονομάζονται ακολουθιακά λογικά κυκλώματα.

Τα στοιχεία μνήμης που χρησιμοποιούνται στα ακολουθιακά κυκλώματα με ρολόι λέγονται flip-flops. Πρόκειται για δυαδικά κύτταρα που μπορούν ν' αποθηκεύσουν ένα bit πληροφορίας. Τα flip-flops έχουν συνήθως δύο εξόδους - μία για την τιμή του bit που είναι αποθηκευμένο μέσα τους και μία για το συμπλήρωμά της. Δυαδικές πληροφορίες μπορούν να μπουν στο flip-flop με διάφορους τρόπους, κι έτσι έχουμε διάφορους τύπους flip-flop. Ένα κύκλωμα flip-flop μπορεί να διατηρηθεί σε μια σταθερή κατάσταση επ' αόριστον (εφόσον το τροφοδοτούμε με ισχύ), έως ότου κάποιο σήμα εισόδου το κάνει να αλλάξει κατάσταση. Στη συνέχεια, θα εξετάσουμε τους διάφορους τύπους flip-flop και θα ορίσουμε τις λογικές τους ιδιότητες.

6.1. Flip-Flops

Το βασικό ψηφιακό κύκλωμα μνήμης flip-flop επιτυγχάνεται με την ένωση δύο πυλών NOT όπως φαίνεται στο Σχ. 6.3.



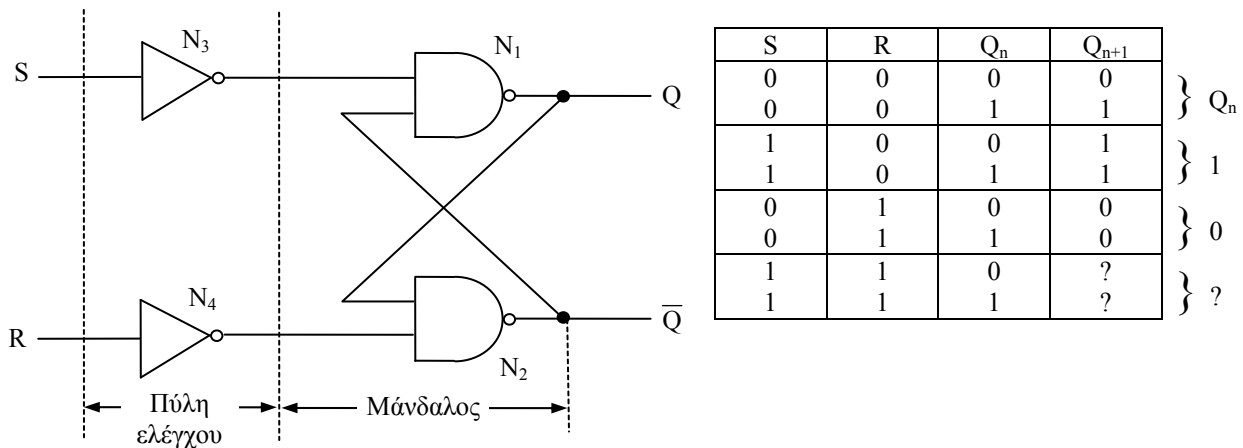
Σχ. 6.3. Βασικό κύκλωμα flip-flop.

Στο κύκλωμα αυτό η έξοδος της μιας πύλης N_1 συνδέεται στην είσοδο της άλλης και αντιστρόφως. Η κυριότερη ιδιότητα ενός κυκλώματος αυτού του τύπου είναι, όταν η μία έξοδος Q βρίσκεται στην κατάσταση "0" η άλλη βρίσκεται υποχρεωτικά στην κατάσταση "1" και αντιστρόφως. Τούτο διότι όταν είναι $Q = 1$ είναι και $A_2 = 1$ άρα η έξοδος της N_2 θα είναι 0. Ομοίως, εάν είναι $A_1 = 0$ τότε η έξοδος της N_1 δεν είναι δυνατόν να έχει άλλη τιμή από $Q = 1$ όπως ήταν και η αρχική μας υπόθεση. Παρόμοια απόδειξη γίνεται και εάν θεωρηθεί αρχικά το $Q = 0$.

Στο κύκλωμα του Σχ. 6.3 δεν υπάρχει η δυνατότητα να επιβληθεί κατά βούληση μία κατάσταση εξόδου. Στη συνέχεια, ως κατάσταση εξόδου ενός flip-flop θα εννοείται η κατάσταση στην οποία βρίσκεται η έξοδος της πύλης N_1 ή η κατάσταση της εξόδου Q .

6.1.1. Flip-Flop S-R

Προκειμένου να επιτύχουμε τον έλεγχο της κατάστασης εξόδου ενός κυκλώματος flip-flop κατασκευάζουμε το κύκλωμα του Σχ. 6.4. Στο κύκλωμα αυτό αλλάξαμε τις δύο πύλες NOT του προηγούμενου flip-flop με δύο πύλες NAND (N_1, N_2) και προσθέσαμε δύο πύλες NOT τις N_3 και N_4 με εισόδους S και R αντίστοιχα, που σημαίνουν $S = SET$ και $R = RESET$.



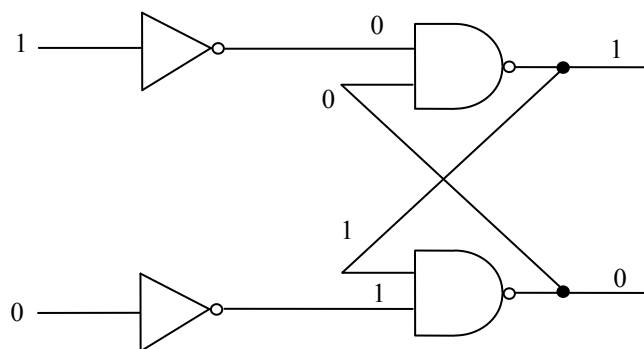
Σχ. 6.4. Βασικό κύκλωμα και πίνακας αληθείας flip-flop S-R .

Στον πίνακα αληθείας του κυκλώματος αυτού το σύμβολο Q_n σημαίνει την κατάσταση της εξόδου Q κατά την περίοδο n και Q_{n+1} την κατάσταση της εξόδου Q κατά την περίοδο $n + 1$. Παρατηρούμε ότι ο πίνακας αληθείας του Σχ. 6.4 απλοποιείται στον πίνακα αληθείας του Σχ. 6.5.

Για να εξηγήσουμε τον τρόπο με τον οποίο επιτυγχάνεται η κατάσταση του πίνακα αληθείας του Σχ. 6.5 θεωρούμε την κατάσταση $Q_n = 1$ και τις εισόδους $S = 1, R = 0$. Εάν στο κύκλωμα του Σχ. 6.4 σημειωθούν οι τιμές αυτές τότε παίρνουμε το Σχ. 6.5. Με τον ίδιο τρόπο αποδεικνύεται η ορθότητα των γραμμών του πίνακα αληθείας μία προς μία. Σε ό,τι αφορά τις γραμμές με το ερωτηματικό, στις περιπτώσεις δηλαδή που αντιστοιχούν στις εισόδους $S = R = 1$, προκύπτει απροσδιόριστη κατάσταση για το κύκλωμα. Στην περίπτωση $S = R = 1$, έχουμε 0 στις εξόδους των N_3 και N_4 και 1 στις εξόδους των N_1 και N_2 . Αυτό αντιφάσκει προς ό,τι είπαμε, ότι οι εξόδοι N_1 και N_2 είναι η μία το συμπλήρωμα της άλλης.

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

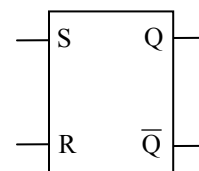
(α)



$Q_n \backslash SR$	00	01	11	10
0			X	1
1	1		X	1

(β)

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

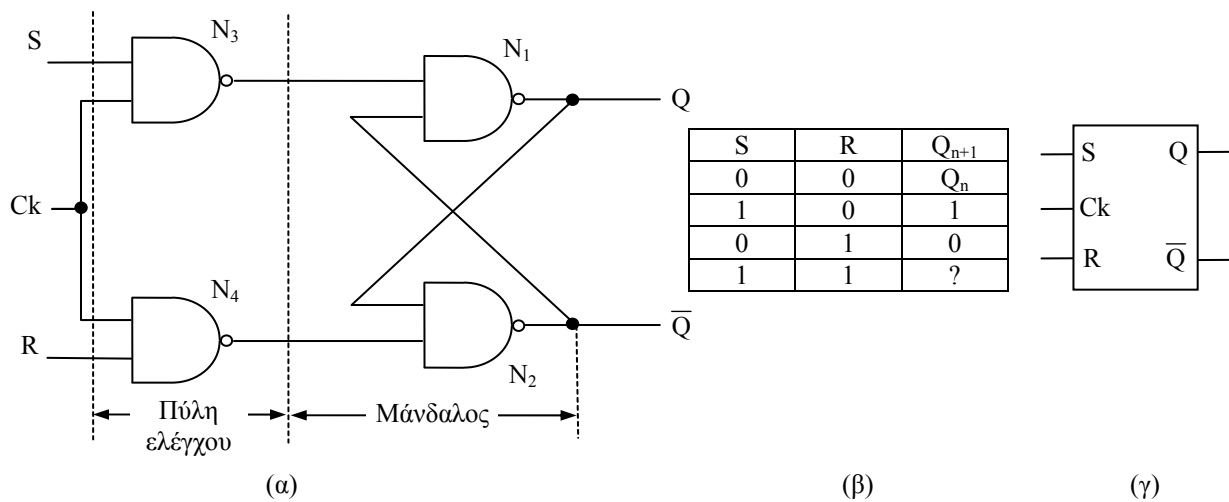


Σχ. 6.5. α) Πίνακας αληθείας flip-flop S-R και παράδειγμα λειτουργίας, β) χάρτης Karnaugh και γραφικό σύμβολο flip-flop S-R.

Η κατάσταση αυτή θεωρείται απροσδιόριστη για το κύκλωμα και υπό κανονική λειτουργία πρέπει να αποφεύγεται. Εάν μετά γίνουν και οι δύο είσοδοι 0, τότε το flip-flop πηγαίνει επίσης σε απροσδιόριστη κατάσταση, η οποία εξαρτάται από το ποια από τις δύο εισόδους αλλάζει τελευταία από το 1 στο 0.

6.1.2. Flip-Flop S-R με ρολοϊ (Clock)

Σε ένα ακολουθιακό ψηφιακό σύστημα απαιτείται πολλές φορές να συγχρονίζεται η λειτουργία ενός flip-flop με τους παλμούς ενός ρολογιού (Ck). Αυτό επιτυγχάνεται με την αντικατάσταση των πυλών N_3 και N_4 του κυκλώματος του Σχ. 6.4 με δύο πύλες NAND και ταυτοχρόνως με εισαγωγή των παλμών Ck στην κοινή τους είσοδο. Το κύκλωμα αυτό φαίνεται στο Σχ. 6.6.



Σχ. 6.6. Βασικό κύκλωμα, πίνακας αληθείας, και γραφικό σύμβολο flip-flop S-R με ρολοϊ.

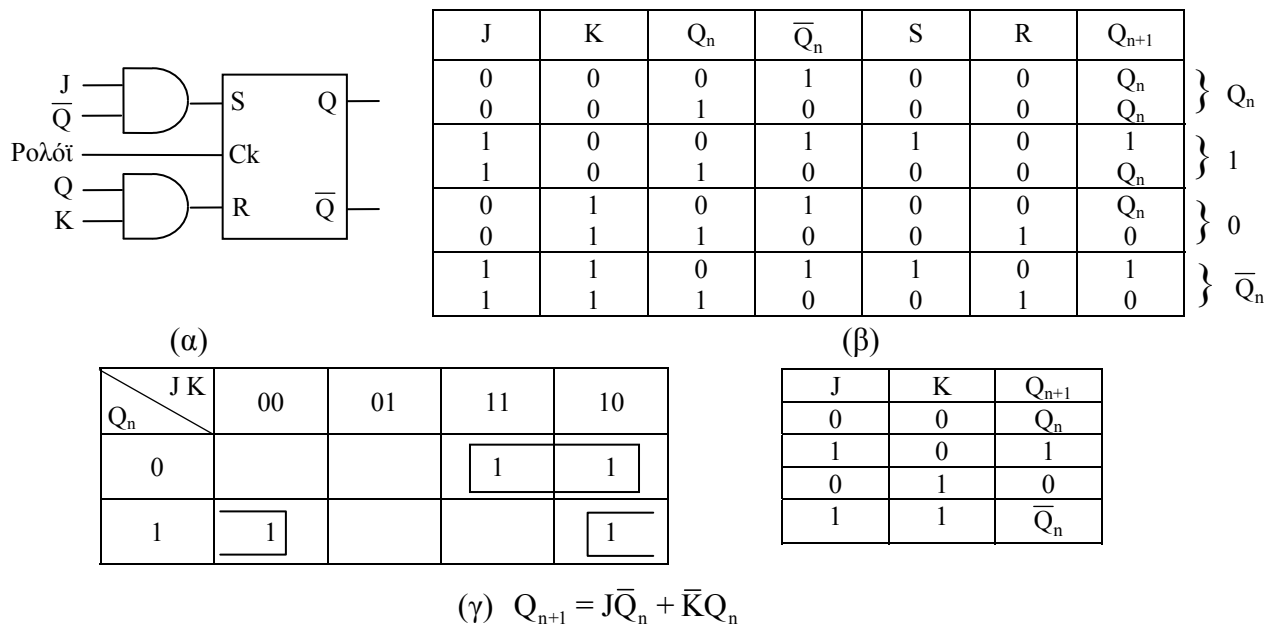
Το κύκλωμα του Σχ. 6.6α ονομάζεται flip-flop S-R με είσοδο ρολογιού. Σε αυτό το flip-flop οι πύλες N_1 και N_2 είναι ο μάνδαλος ή μανταλωτής (latch), ενώ οι πύλες N_3 και N_4 είναι οι πύλες ελέγχου (steering gates) οι οποίες προγραμματίζουν την κατάσταση εξόδου που θα επικρατήσει στο flip-flop αμέσως μετά τον πρώτο παλμό του ρολογιού $Ck = 1$. Στο χρονικό διάστημα μεταξύ δύο παλμών κατά τη διάρκεια του οποίου θα είναι $Ck = 0$ δεν είναι δυνατό να γίνει καμία αλλαγή. Στο Σχ. 6.6β φαίνεται ο πίνακας αληθείας του flip-flop S-R και στο Σχ. 6.6γ το γραφικό σύμβολο του flip-flop S-R. Από αυτά προκύπτει η κατάσταση δύο διαδοχικών περιόδων T_n και T_{n+1} , Q_n και Q_{n+1} , αντιστοίχως ως εξής:

- α) Εάν είναι $S_n = 0$ και $R_n = 0$, είναι δε Q_n είτε “0” είτε “1”, τότε μετά τον επόμενο παλμό δηλαδή κατά την περίοδο T_{n+1} η κατάσταση Q_{n+1} θα παραμένει όμοια με την προηγούμενη Q_n .
- β) Εάν είναι $S_n = 1$ και $R_n = 0$, τότε οποιαδήποτε και αν είναι η κατάσταση Q_n , μετά τον επόμενο παλμό δηλαδή κατά την περίοδο T_{n+1} θα είναι $Q_{n+1} = 1$.
- γ) Εάν είναι $S_n = 0$ και $R_n = 1$, τότε οποιαδήποτε και αν είναι η κατάσταση Q_n , μετά τον επόμενο παλμό δηλαδή κατά την περίοδο T_{n+1} θα είναι $Q_{n+1} = 0$.
- δ) Εάν είναι $S_n = 1$ και $R_n = 1$, τότε προκύπτει μία ασαφής (απροσδιόριστη) κατάσταση που μπορεί να δώσει είτε $Q_{n+1} = 0$ είτε $Q_{n+1} = 1$.

Από τα ανωτέρω δικαιολογείται ο συμβολισμός με τα γράμματα “S” (Set : Θέση), και “R” (Reset : Επαναφορά), διότι όταν είναι $S = 1$ και $R = 0$ το flip-flop **τίθεται** (Set) στην κατάσταση “1” δηλαδή $Q = 1$ και αντιθέτως, όταν είναι $S = 0$ και $R = 1$ το flip-flop **επαναφέρεται** (Reset) στην κατάσταση “0” δηλαδή $Q = 0$.

6.1.3. Flip-Flop J-K

Η ασάφεια καταστάσεως του flip-flop S-R η οποία φαίνεται στον πίνακα αληθείας του Σχ. 6.6β, αναιρείται με την προσθήκη δύο πυλών AND στις εισόδους του flip-flop S-R όπως φαίνεται στο Σχ. 6.7α, οπότε ο πίνακας αληθείας δεν περιλαμβάνει πλέον απροσδιοριστία, Σχ. 6.7β. Εάν συγκρίνουμε το κύκλωμα αυτό με εκείνο του Σχ. 6.6α, θα διαπιστώσουμε ότι είναι δυνατό να διαμορφωθεί η ίδια κατάσταση εάν προστεθεί απλώς μία τρίτη είσοδος σε κάθε μία από τις πύλες N_3 και N_4 όπως φαίνεται στο Σχ. 6.8α. Στο κύκλωμα αυτό οι εξόδοι Q και \bar{Q} είναι διασυνδεδεμένες με ανατροφοδότηση (Feedback) στις αντίστοιχες εισόδους όπως φαίνεται από τις έντονες γραμμές του κυκλώματος του Σχ. 6.8α. Ο πίνακας αληθείας του Σχ. 6.7β δίνει την κατάσταση των εξόδων του flip-flop J-K, Q_{n+1} μετά από τον n-οστό παλμό του ρολογιού, συναρτήσει της κατάστασης των εισόδων J_n και K_n πριν από τον παλμό αυτόν. Εννοείται ότι η κατάσταση των εξόδων αυτών πριν από τον παλμό είναι τυχούσα.



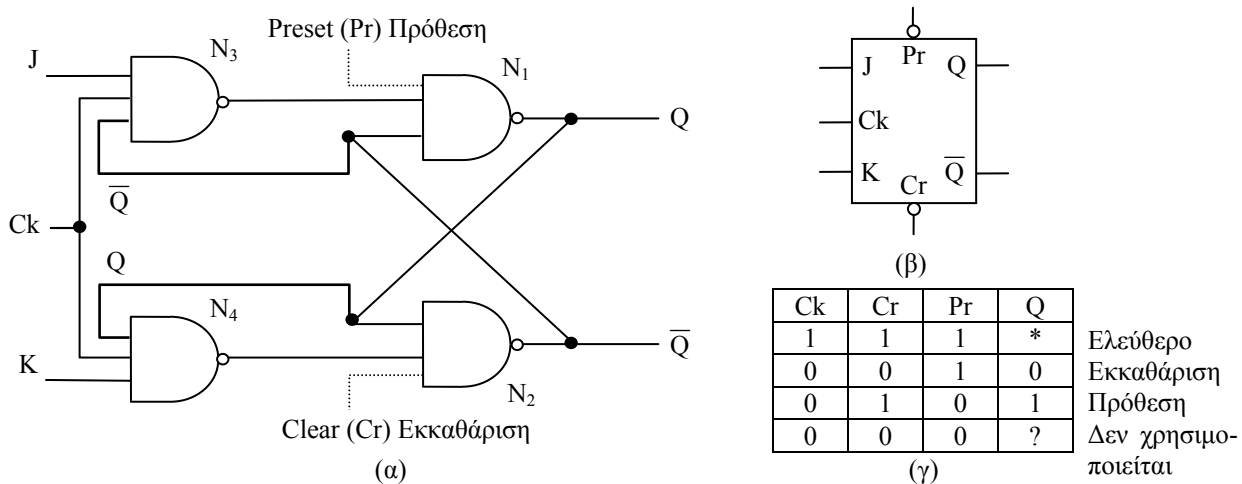
Σχ. 6.7. Βασικό κύκλωμα, πίνακας αληθείας και χάρτης Karnaugh του flip-flop J-K.

Στο κύκλωμα του Σχ. 6.8α έχουν προστεθεί από μία είσοδος σε κάθε πύλη N_1 και N_2 . Οι εισόδους αυτές φαίνονται στο κύκλωμα με διακεκομμένη γραμμή και με αυτές γίνεται ο προσδιορισμός της αρχικής κατάστασης του flip-flop J-K.

Εάν επιθυμούμε την εκκαθάριση της μνήμης (Clear - Cr), πριν να επιβληθεί ο παλμός, δηλαδή την κατάσταση $Q = 0$ όταν $Ck = 0$, τότε επιβάλλουμε τον συνδυασμό $Cr = 0$, $Pr = 1$ και $Ck = 0$ οπότε η έξοδος του N_2 θα είναι οπωσδήποτε $\bar{Q} = 1$ και επομένως $Q = 0$.

Εάν εξάλλου επιθυμούμε να θέσουμε τη μνήμη (Preset - Pr) πριν να επιβληθεί ο παλμός, δηλαδή να θέσουμε την κατάσταση $Q = 1$ όταν είναι $Ck = 0$, τότε

επιβάλλουμε τον συνδυασμό $Cr = 1$, $Pr = 0$ και $Ck = 0$ οπότε η έξοδος του N_1 θα είναι οπωσδήποτε $Q = 1$.



Σχ. 6.8. Flip-flop J-K με εκκαθάριση και πρόθεση και ο αντίστοιχος πίνακας αληθείας.

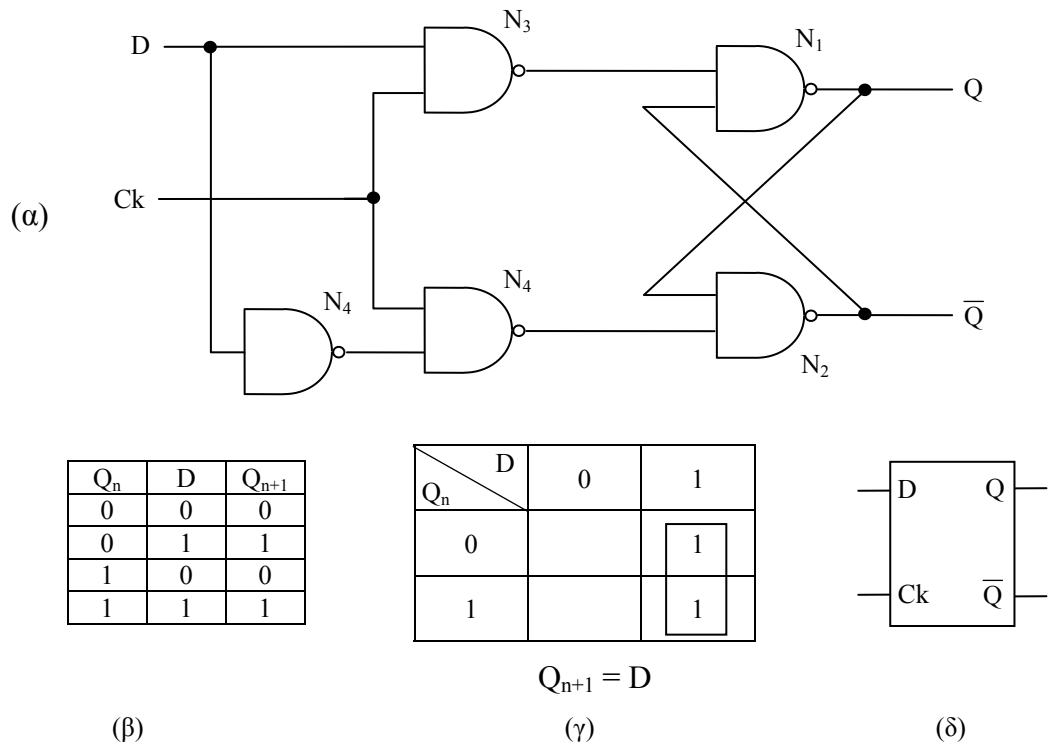
Οι καταστάσεις εκκαθάρισης και πρόθεσης είναι ασύγχρονες, δηλαδή είναι δυνατό να επιτευχθούν ανεξάρτητα από τον παλμό του ρολογιού. Τούτο σημαίνει ότι για να γίνει η πρόθεση ή εκκαθάριση του flip-flop, η αλλαγή της κατάστασης του μανδάλου γίνεται ακόμη και όταν είναι $Ck = 0$. Για να φέρουμε το flip-flop J-K στην κατάσταση συγχρονιζόμενης αλλαγής συνδέουμε και τα δύο άκρα εκκαθάρισης και πρόθεσης στην κατάσταση “1” δηλαδή $Cr = 1$ και $Pr = 1$, τότε ο επόμενος παλμός $Ck = 1$ θα φέρει το flip-flop σε κατάσταση η οποία να αντιστοιχεί στις τιμές των J_n και K_n σύμφωνα με τον πίνακα αληθείας του Σχ. 6.7β. Από τον πίνακα αυτό παρατηρούμε ότι εάν τεθεί $J_n = 1$ και $K_n = 1$, τότε κάθε νέος παλμός θα προκαλεί αλλαγή της κατάστασης του flip-flop.

6.1.4. Flip-Flop D

Ένας άλλος τρόπος εξάλειψης της ανεπιθύμητης συμπεριφοράς στην απροσδιόριστη κατάσταση ενός flip-flop S-R είναι να εξασφαλιστεί ότι οι εισόδους S και R δεν είναι ποτέ ταυτόχρονα 1. Αυτό γίνεται στο flip-flop D που φαίνεται στο Σχ. 6.9α. Το flip-flop D έχει δύο μόνον εισόδους: D και Ck. Η είσοδος D πηγαίνει κατευθείαν στην είσοδο S και το συμπλήρωμά της στην είσοδο R. Εφόσον ο παλμός εισόδου είναι $Ck = 0$, οι έξοδοι των πυλών N_3 και N_4 είναι 1 και το κύκλωμα δεν μπορεί να αλλάξει κατάσταση ανεξάρτητα από την τιμή της D. Η είσοδος D “δειγματοληπτείται” όταν $Ck = 1$. Αν η D είναι 1, η έξοδος Q γίνεται 1 και το κύκλωμα πηγαίνει στην κατάσταση θέσης (Set). Αν η D είναι 0, η έξοδος Q γίνεται 0 και το κύκλωμα μεταβαίνει στην κατάσταση μηδενισμού.

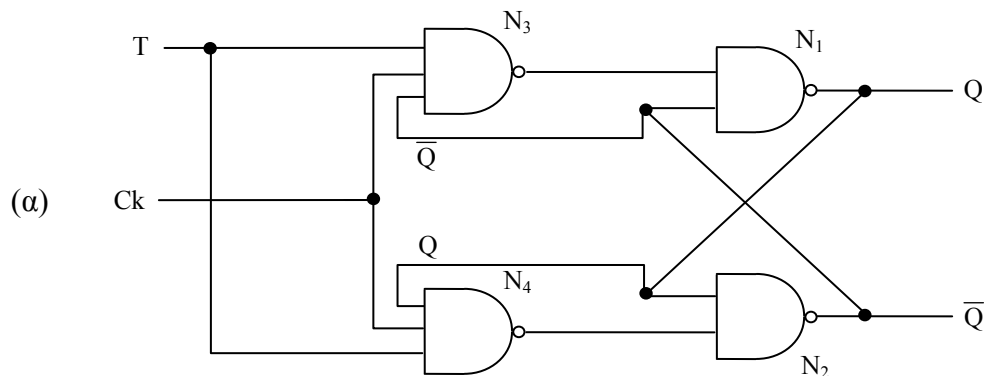
Το flip-flop τύπου D παίρνει το όνομά του από τη δυνατότητα να κρατά μέσα του δεδομένα (Data). Αυτόν τον τύπο flip-flop τον λέμε μερικές φορές “φυλασσόμενο μανταλωτή τύπου D” (gated D-latch). Οι δυαδικές πληροφορίες που βρίσκονται στην είσοδο δεδομένων του flip-flop τύπου D, μεταφέρονται στην έξοδο Q, όταν ενεργοποιείται η είσοδος Ck. Η έξοδος ακολουθεί τα δεδομένα εισόδου όσο ο παλμός παραμένει στην κατάσταση 1. Δηλαδή ο μανταλωτής D είναι ευαίσθητος στη διάρκεια του παλμού ρολογιού, και η έξοδος μπορεί να αλλάζει ακολουθώντας τις

μεταβολές της εισόδου, όσο ο παλμός ρολογιού είναι ενεργοποιημένος (HIGH). Όταν ο παλμός πάει στο 0, οι δυαδικές πληροφορίες που βρίσκονταν στην είσοδο δεδομένων την ώρα που συνέβη η μετάβαση του παλμού, παραμένουν στην έξοδο Q μέχρις ότου η παλμική είσοδος ενεργοποιηθεί ξανά. Η ευαισθησία αυτή των μανταλωτών στη διάρκεια του παλμού ενεργοποίησης ισχύει όχι μόνο για τα flip-flop D αλλά και για όλα τα άλλα flip-flop π.χ. S-R, J-K, T, τα οποία επίσης ονομάζονται μανταλωτές (latches).



Σχ. 6.9. α) Βασικό κύκλωμα, β) πίνακας αληθείας, γ) χάρτης Karnaugh, και δ) γραφικό σύμβολο flip-flop τύπου D.

Ο πίνακας αληθείας του flip-flop D φαίνεται στο Σχ. 6.9β. Από αυτόν φαίνεται ότι η επόμενη κατάσταση του flip-flop είναι ανεξάρτητη από την παρούσα, αφού η Q_{n+1} ισούται με την είσοδο D είτε η Q_n είναι 1 είτε είναι 0. Αυτό σημαίνει ότι ένας παλμός εισόδου θα μεταφέρει την τιμή της εισόδου D στην έξοδο του flip-flop, ανεξάρτητα από την τιμή της εξόδου πριν από την εφαρμογή του παλμού. Η χαρακτηριστική εξίσωση δείχνει καθαρά ότι η Q_{n+1} είναι ίση με την D.



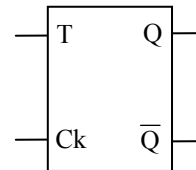
Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

(β)

$Q_n \backslash T$	0	1
0		1
1	1	

$$Q_{n+1} = T\bar{Q}_n + \bar{T}Q_n$$

(γ)



(δ)

Σχ. 6.10. α) Βασικό κύκλωμα, β) πίνακας αληθείας, γ) χάρτης Karnaugh, και δ) γραφικό σύμβολο flip-flop τύπου T.

6.1.5. Flip-Flop T

Το flip-flop τύπου T είναι μια παραλλαγή του flip-flop τύπου J-K που έχει μία μόνο είσοδο. Όπως φαίνεται στο Σχ. 6.10α, το flip-flop τύπου T το παίρνουμε από το τύπου J-K, εάν συνδέσουμε μεταξύ τους τις δύο εισόδους. Η ονομασία T προέρχεται από τη δυνατότητα του flip-flop να *αντιστρέφεται* (toggle), δηλαδή να αλλάζει κατάσταση. Σε όποια κατάσταση και να βρίσκεται το flip-flop, όταν έλθει ο παλμός του ρολογιού ενώ $T = 1$, πηγαίνει στη συμπληρωματική κατάσταση. Ο χαρακτηριστικός πίνακας και η χαρακτηριστική εξίσωση δείχνουν ότι όταν $T = 0$, $Q_{n+1} = Q_n$, δηλαδή η επόμενη κατάσταση είναι ίδια με την παρούσα και καμιά αλλαγή δεν συμβαίνει. Όταν $T = 1$, τότε $Q_{n+1} = \bar{Q}_n$ και η κατάσταση του flip-flop συμπληρώνεται.

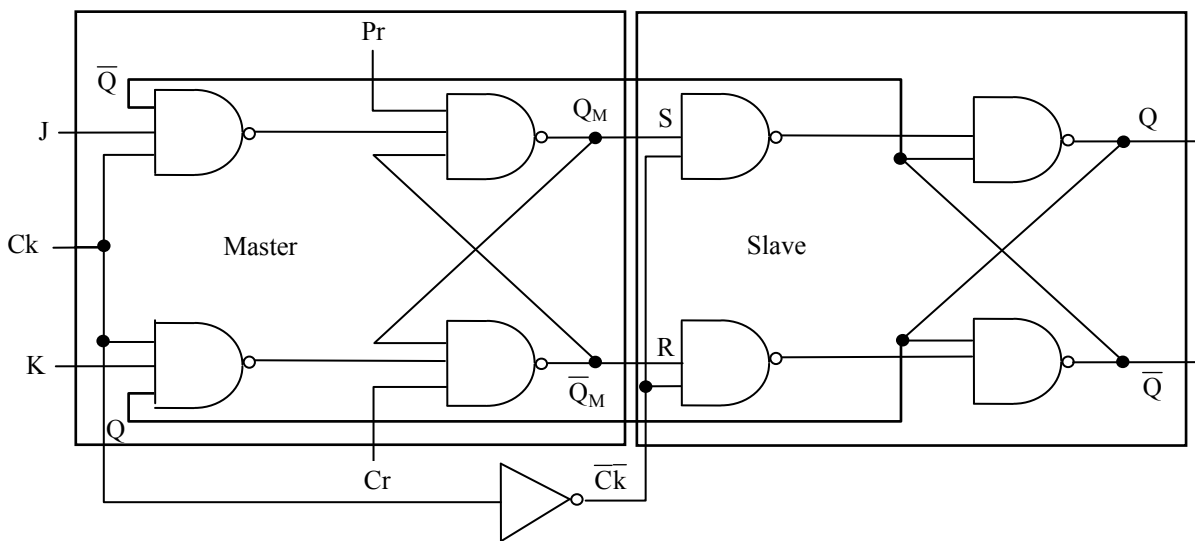
6.1.6. Πυροδότηση των Flip-Flops

Το flip-flop J-K του Σχ. 6.8, που αναλύσαμε παραπάνω, παρουσιάζει την εξής αδυναμία. Όταν $Ck = 1$, και $J = K = 1$, τότε αλλάζει κατάσταση. Λόγω της ανατροφοδότησης που υπάρχει, το flip-flop θα συνεχίζει να αλλάζει καταστάσεις για όσο χρονικό διάστημα ο παλμός του ρολογιού βρίσκεται στο 1. Πόσο συχνά θα αλλάξει καταστάσεις; Όση είναι η καθυστέρηση διάδοσης του σήματος μέσα από το flip-flop. Στην προκειμένη περίπτωση αυτή είναι ίση με την καθυστέρηση 3 πυλών. Αυτό σημαίνει πως για τα συνήθη ολοκληρωμένα κυκλώματα στα οποία κάθε πύλη εισάγει καθυστέρηση 10 nsec, η συνολική καθυστέρηση του flip-flop θα είναι 30 nsec. Άρα, μια πρώτη αντιμετώπιση του προβλήματος αυτού θα ήταν οι παλμοί του ρολογιού να έχουν εύρος στο λογικό "1", μικρότερο από την καθυστέρηση διάδοσης του σήματος μέσα από το flip-flop, δηλαδή μικρότερο των 30 nsec.

Αυτή η απαίτηση είναι εξαιρετικά δεσμευτική και δεν χρησιμοποιείται στην πράξη. Το πρόβλημα αυτό μπορεί να αντιμετωπιστεί με δύο τρόπους: (α) με flip-flop J-K τύπου Master-Slave και (β) με flip-flops τα οποία ενεργοποιούνται κατά την άνοδο ή καθοδο του παλμού ρολογιού (ακμοπυροδοτούμενα, edge-triggered) και όχι κατά τη διάρκεια του παλμού. Αυτές τις δύο τεχνικές θα εξετάσουμε αμέσως παρακάτω.

6.1.7. Master - Slave Flip-Flop

Το flip-flop Master – Slave (Κύριο – Εξαρτημένο) αποτελείται από δύο flip-flop, ένα **Κύριο (Master)** και ένα **Εξαρτημένο (Slave)**, το πρώτο είναι τύπου J-K και το δεύτερο τύπου S-R. Σε τούτο το flip-flop, η ανατροφοδότηση γίνεται από την έξοδο του δεύτερου flip-flop προς την είσοδο του πρώτου. Οι παλμοί ρολογιού επιβάλλονται στην είσοδο Ck του πρώτου flip-flop και μετά από αντιστροφή επιβάλλονται και στην είσοδο του δεύτερου. Όταν $Ck = 1$, ενεργοποιείται το πρώτο flip-flop, ενώ το δεύτερο είναι απενεργοποιημένο. Το αντίθετο συμβαίνει όταν ο παλμός ρολογιού γίνει $Ck = 0$. Έτσι, σε όλη τη διάρκεια που ο παλμός $Ck = 1$, το Slave flip-flop δεν μπορεί να αλλάξει κατάσταση και κατά συνέπεια οι εισόδους του πρώτου flip-flop από ανατροφοδότηση μένουν σταθερές στην κατάσταση που ήταν. Ας δούμε τη λειτουργία του λίγο πιο αναλυτικά, Σχ. 6.11.



Σχ. 6.11. Βασικό κύκλωμα flip-flop J-K Master-Slave.

Η κατάσταση η οποία επικρατεί στις εισόδους J και K ενεργεί στο Master flip-flop κατά την ύψωση της τάσης του παλμού $Ck = 1$ και συγκρατείται εκεί έως ότου έλθει η πτώση της τάσης του παλμού μετά το χρόνο t_p , και τότε επιτρέπεται να περάσει η πληροφορία αυτή από το Master προς το Slave flip-flop. Εάν οι εισόδους είναι σε κατάσταση $J = 1$ και $K = 0$ πριν από την επιβολή ενός παλμού ρολογιού, το flip-flop έρχεται στην κατάσταση $Q = 1$ αμέσως μετά από τον παλμό. Εάν οι εισόδους είναι σε κατάσταση $J = 0$ και $K = 1$ πριν από την επιβολή ενός παλμού ρολογιού, το flip-flop έρχεται στην κατάσταση $Q = 0$ αμέσως μετά από τον παλμό. Εάν οι εισόδους είναι σε κατάσταση $J = 0$ και $K = 0$ πριν από την επιβολή ενός παλμού ρολογιού, το flip-flop παραμένει μετά τον παλμό στην οποιαδήποτε κατάσταση βρισκόταν πριν από αυτόν τον παλμό. Εάν οι εισόδους είναι σε κατάσταση $J = 1$ και $K = 1$ πριν από την επιβολή ενός παλμού ρολογιού, το flip-flop αλλάζει κατάσταση κατά την πτώση της τάσης του παλμού και έρχεται σε κατάσταση αντίθετη από εκείνη στην οποία βρισκόταν πριν από την επιβολή του παλμού. Παρατηρούμε δηλαδή ότι το flip-flop Master-Slave λειτουργεί ακριβώς όπως το flip-flop J-K του Σχ. 6.8α και δεν παρουσιάζει το πρόβλημα των συνεχών αλλαγών καταστάσεων για $J = K = Ck = 1$. Από την όλη ανάλυση θα μπορούσαμε να πούμε ότι το flip-flop Master-Slave εισάγει τα δεδομένα στο Master flip-flop όταν το ρολοί είναι "1" και μεταφέρει το αποτέλεσμα στο Slave flip-flop όταν το ρολοί πηγαίνει στο "0". Παρατηρήστε ότι οι εξωτερικές εισόδους J-K

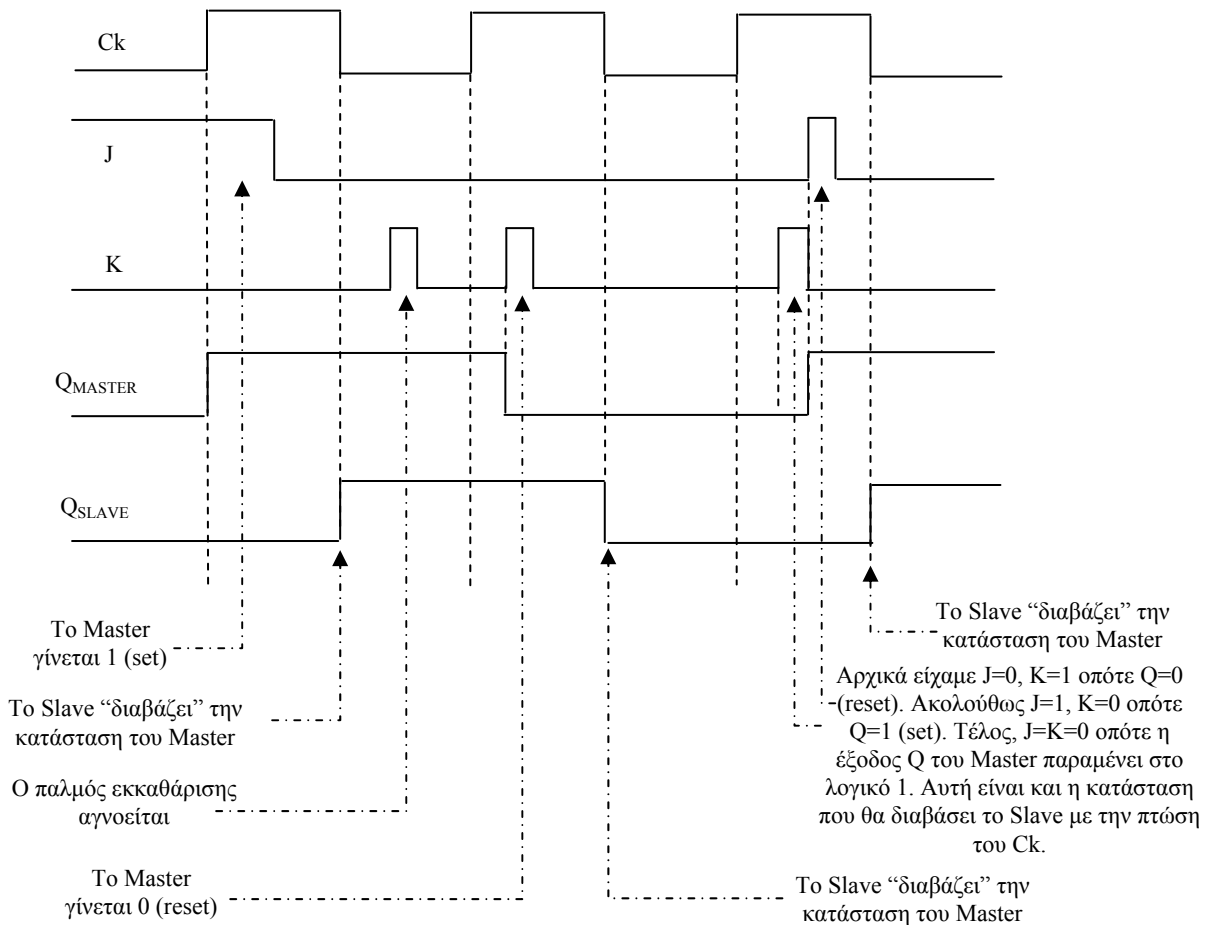
επιτρέπεται να αλλάζουν αμέσως μόλις ο παλμός Ck πέσει από το 1 στο 0 (δηλαδή στην αρνητική του ακμή), διότι μόλις $Ck = 0$, ο Master απομονώνεται και οι εισόδους του J και K δεν έχουν πια καμιά επίδραση μέχρις ότου έλθει ο επόμενος παλμός του ρολογιού. Έτσι, στα flip-flop Master-Slave μπορούμε να χρησιμοποιούμε για όλα τα ίδιο ρολοί και να έχουμε και τις εισόδους και τις εξόδους να αλλάζουν την ίδια στιγμή. Θα μπορούσε δηλαδή η είσοδος J να έρχεται από την έξοδο Q ενός άλλου ίδιου flip-flop που αλλάζει με το ίδιο ρολοί.

Όπως περιγράψαμε, στα flip-flop Master-Slave οι αλλαγές της κατάστασής τους γίνονται με την αρνητική ακμή του ρολογιού. Υπάρχουν flip-flop που οι αλλαγές αυτές γίνονται στη θετική ακμή του ρολογιού. Σε αυτά τα flip-flop, υπάρχει ένας επιπλέον αντιστροφέας μεταξύ του ακροδέκτη Ck και της εισόδου ρολογιού του Master. Αυτά τα flip-flop πυροδοτούνται με αρνητικούς παλμούς, έτσι ώστε η αρνητική ακμή του παλμού να επιδρά στο Master και μετά η θετική στο Slave, και άρα και στις εξόδους.

Ο συνδυασμός Master-Slave μπορεί να κατασκευαστεί για κάθε τύπο flip-flop προσθέτοντάς του ένα flip-flop S-R με αντεστραμμένο ρολοί που να αποτελεί το Slave.

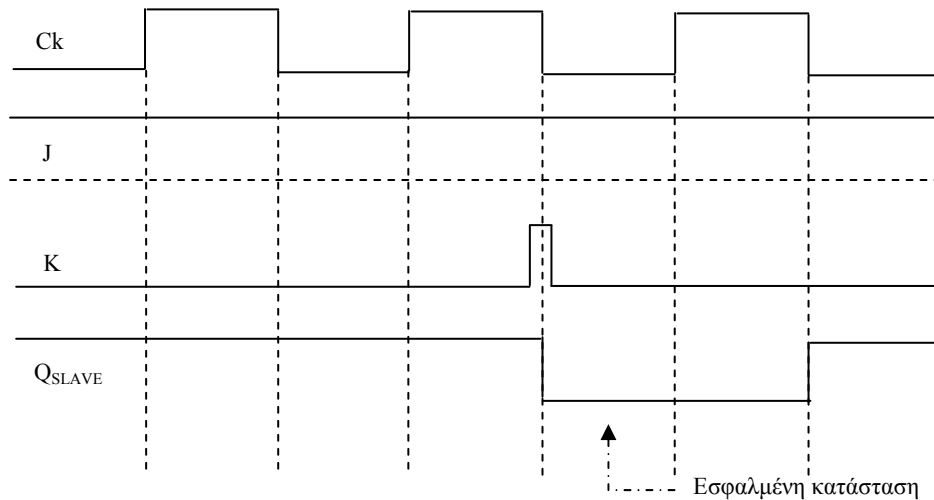
Παράδειγμα 6.1

Σχεδιάστε την κυματομορφή εξόδου Q ενός Flip-Flop J-K Master-Slave του οποίου οι κυματομορφές εισόδου φαίνονται παρακάτω. Θεωρήστε ότι αρχικά έχουμε $Q = 0$.



Σχ. 6.12. Κυματομορφές εισόδου-εξόδου ενός J-K flip-flop τύπου Master-Slave.

Τα Master-Slave flip-flop ονομάζονται παλμο-πυροδοτούμενα ή επιπεδο-πυροδοτούμενα (pulse-triggered ή level-triggered) διότι ενεργοποιούνται στη διάρκεια του παλμού ρολογιού (Π) και όχι κατά τη στιγμή που ο παλμός αλλάζει κατάσταση από HIGH (“1”) σε LOW (“0”) (↓) ή από LOW (“0”) σε HIGH (↑) (“1”). Έτσι τα δεδομένα εισόδου διαβάζονται σε όλη τη διάρκεια που ο παλμός ρολογιού είναι HIGH. Το γεγονός αυτό εξακολουθεί να μας δημιουργεί κάποιο πρόβλημα. Έστω, για παράδειγμα, ότι $J = 1$ και $K = 0$, οπότε η έξοδος του Master-Slave JK γίνεται $Q = 1$.



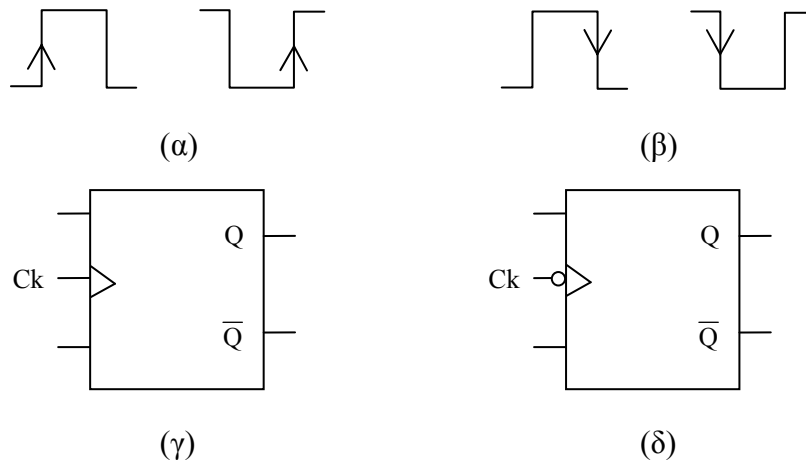
Σχ. 6.13. Ένας ανεπιθύμητος παλμός στην είσοδο K ενός Master-Slave JK flip-flop οδηγεί εσφαλμένα στην αλλαγή κατάστασής του.

Ένας ανεπιθύμητος θετικός παλμός μικρού εύρους (σπινθήρας, spike) στην είσοδο K κατά τη διάρκεια που ο παλμός ρολογιού είναι HIGH, οδηγεί σε εκκαθάριση του Master ($Q_M = 0$). Αυτή η κατάσταση μεταφέρεται με την πτώση του παλμού ρολογιού στην έξοδο του Slave, οπότε τελικά $Q = 0$. Το αποτέλεσμα είναι το Master-Slave flip-flop εσφαλμένα να αλλάξει κατάσταση, Σχ. 6.13. Το ίδιο θα συνέβαινε εάν $J=0$ και $K=1$, οπότε $Q = 0$. Η εμφάνιση ενός θετικού σπινθήρα στην είσοδο J θα οδηγούσε εσφαλμένα στην αλλαγή κατάστασης του flip-flop. Τη λύση σε όλα αυτά τα προβλήματα, μας την δίνει η σχεδίαση flip-flop που να είναι ευαίσθητα κατά τη στιγμή που αλλάζει κατάσταση ο παλμός ρολογιού και όχι όταν αυτός βρίσκεται σε μια συγκεκριμένη λογική στάθμη. Αυτά ονομάζονται ακμοπυροδοτούμενα (edge-triggered) και τα μελετάμε στη συνέχεια.

6.1.8. Ακμοπυροδοτούμενα Flip-Flop

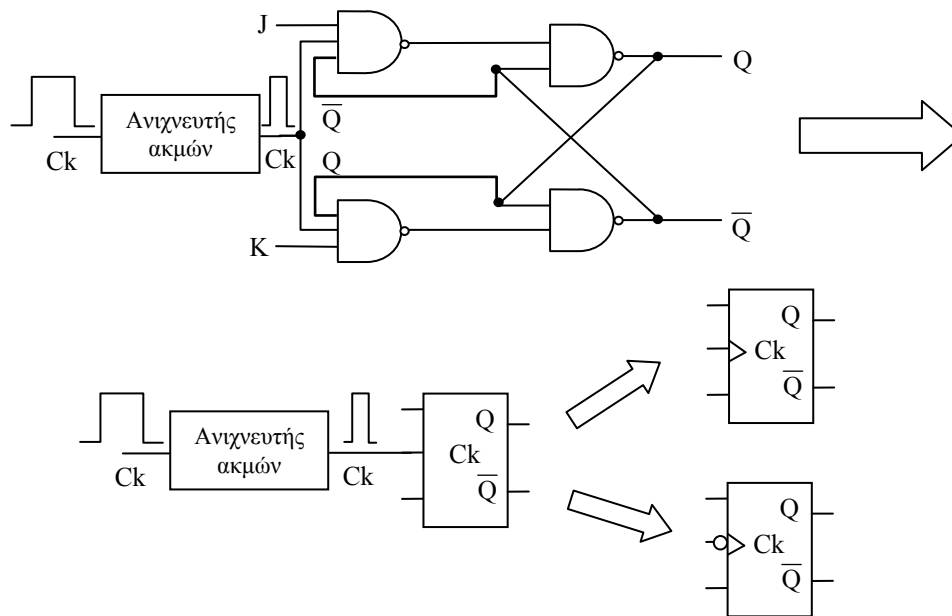
Ακμοπυροδοτούμενα flip-flop είναι αυτά τα οποία είναι έτσι σχεδιασμένα ώστε να είναι ευαίσθητα κατά την αλλαγή της λογικής στάθμης του παλμού ρολογιού. Όταν ο παλμός ρολογιού μεταβαίνει από το “0” στο “1”, τότε λέμε ότι έχουμε μία θετική μετάβαση (Σχ. 6.14), και τα flip-flop που είναι ευαίσθητα κατά τη μετάβαση αυτή ονομάζονται **θετικά ακμοπυροδοτούμενα** (positive edge-triggered flip-flops). Στην αντίθετη περίπτωση, δηλαδή για μετάβαση από “1” σε “0” έχουμε τα **αρνητικά ακμοπυροδοτούμενα** flip-flop (negative edge-triggered flip-flops). Η είσοδος του παλμού ρολογιού σε ένα flip-flop συμβολίζεται με ένα τρίγωνο. Το τριγωνικό αυτό σύμβολο, Σχ. 6.14γ, δηλώνει ότι το flip-flop πυροδοτείται στη θετική ακμή του παλμού. Η παρουσία ενός μικρού κύκλου έξω από το ορθογώνιο, και αμέσως πριν

από το τριγωνικό σύμβολο, δηλώνει ότι το flip-flop πυροδοτείται στην αρνητική ακμή του παλμού, Σχ. 6.14δ.



Σχ. 6.14. Ακμοπυροδότηση: (α) θετική ακμή παλμού, (β) αρνητική ακμή παλμού, (γ) σύμβολο θετικά ακμοπυροδοτούμενου flip-flop, (δ) σύμβολο αρνητικά ακμοπυροδοτούμενου flip-flop.

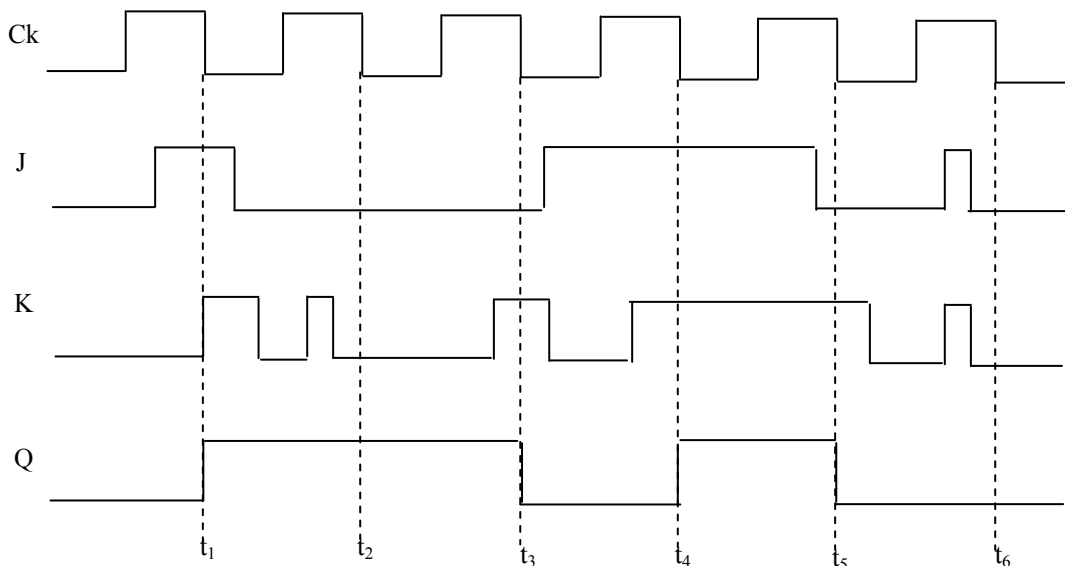
Μπορούμε να έχουμε ακμοπυροδοτούμενα flip-flop όλων των τύπων, δηλ. S-R, D, J-K, T, γι' αυτό και στα Σχ. 6.14γ και 6.14δ δεν προσδιορίζεται ο τύπος του flip-flop αλλά μόνον ο τύπος της ακμοπυροδότησης. Έτσι, ένα ακμοπυροδοτούμενο flip-flop J-K είναι το ίδιο με το flip-flop J-K του Σχ. 6.8, με μόνη διαφορά στο κύκλωμα εισόδου των παλμών, όπου έχει προστεθεί ένα κύκλωμα ανίχνευσης των ακμών (edge detector, όπως φαίνεται στο Σχ. 6.15). Ο πίνακας αληθείας του flip-flop παραμένει ακριβώς ο ίδιος.



Σχ. 6.15. Λογικό κύκλωμα και συμβολισμός ενός ακμοπυροδοτούμενου flip-flop.

Παράδειγμα 6.2

Σχεδιάστε την κυματομορφή εξόδου Q ενός αρνητικά ακμοπυροδοτούμενου Flip-Flop J-K του οποίου οι κυματομορφές εισόδου φαίνονται παρακάτω. Θεωρήστε ότι αρχικά έχουμε $Q = 0$.



Σχ. 6.16. Κυματομορφές εισόδου – εξόδου αρνητικά ακμοπυροδοτούμενου flip-flop J-K.

Το flip-flop είναι αρνητικά ακμοπυροδοτούμενο, άρα μας ενδιαφέρει το τι βλέπει στις εισόδους του εκείνη τη χρονική στιγμή. Οι αρνητικές μεταβάσεις των παλμών ρολογιού σημειώνονται ως t_1, t_2, \dots, t_6 . Έτσι έχουμε:

t_1 : $J = 1, K = 0$, άρα $Q = 1$.

t_2 : $J = 0, K = 0$ και συνεπώς αυτό παραμένει στην προηγούμενη κατάσταση του $Q = 1$. Παρατηρείστε ότι οι μεταβάσεις της εισόδου K πριν την αρνητική ακμή αγνοήθηκαν.

t_3 : $J = 0, K = 1$ άρα $Q = 0$.

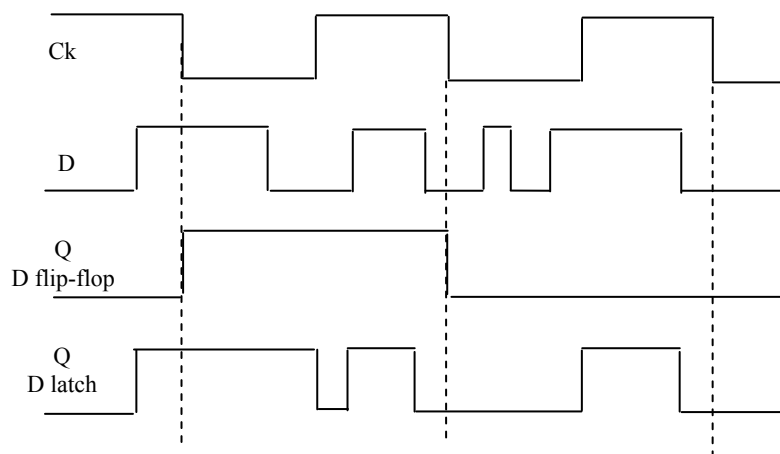
t_4 : $J = 1, K = 1$ άρα $Q = 1$.

t_5 : $J = 0, K = 1$ άρα $Q = 0$.

t_6 : $J = 0, K = 0$ άρα $Q = 0$, όπως και στην αμέσως προηγούμενη κατάσταση.

Παράδειγμα 6.3

Σχεδιάστε την κυματομορφή εξόδου Q ενός αρνητικά ακμοπυροδοτούμενου flip-flop D του οποίου οι κυματομορφές εισόδου φαίνονται παρακάτω. Συγκρίνετε την κυματομορφή που προκύπτει με την αντίστοιχη του μανταλωτή D (D-latch). Θεωρήστε ότι αρχικά έχουμε $Q = 0$.



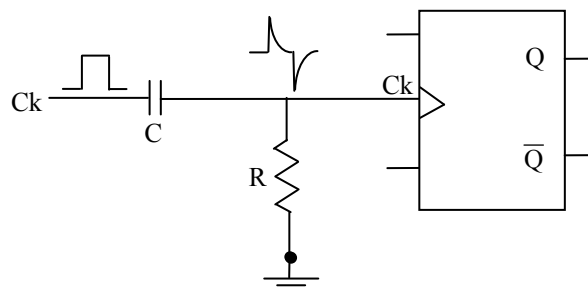
Σχ. 6.17. Κυματομορφές εισόδου – εξόδου αρνητικά ακμοπυροδοτούμενου flip-flop D και μανταλωτή D (latch).

Εφόσον το D flip-flop είναι αρνητικά ακροπυροδοτούμενο, θα ενεργοποιείται κάθε φορά που παρουσιάζεται μια αρνητική ακμή στους παλμούς ρολογιού, μεταφέροντας στην έξοδο Q την τιμή που έχει η είσοδος D τη στιγμή εκείνη. Οι οποιοσδήποτε εναλλαγές της εισόδου πριν την πτώση (αρνητική ακμή) των παλμών ρολογιού, δεν γίνονται αντιληπτές από το D flip-flop. Αντίθετα, οι οποιοσδήποτε τέτοιες εναλλαγές της εισόδου του D μανταλωτή (D latch) κατά τη διάρκεια που ο παλμός ρολογιού είναι “1”, μεταφέρονται στην έξοδό του, αφού $Q = D$, όταν αυτό είναι ενεργοποιημένο. Γίνεται δηλαδή και πάλι φανερό, ότι ο D μανταλωτής είναι ευαίσθητος στη διάρκεια του παλμού ρολογιού, ενώ το D flip-flop στην αλλαγή στάθμης του παλμού ρολογιού.

6.1.9. Ακροπυροδοτούμενα Flip-Flop

Είδαμε παραπάνω ότι όλα τα προβλήματα που παρουσίασαν τα flip-flops, αντιμετωπίστηκαν με την μετατροπή τους από παλμο-πυροδοτούμενα σε ακροπυροδοτούμενα. Αυτό έγινε με τη βοήθεια ενός κυκλώματος ανίχνευσης των ακμών, δηλαδή ανίχνευσης των αλλαγών στάθμης του ρολογιού.

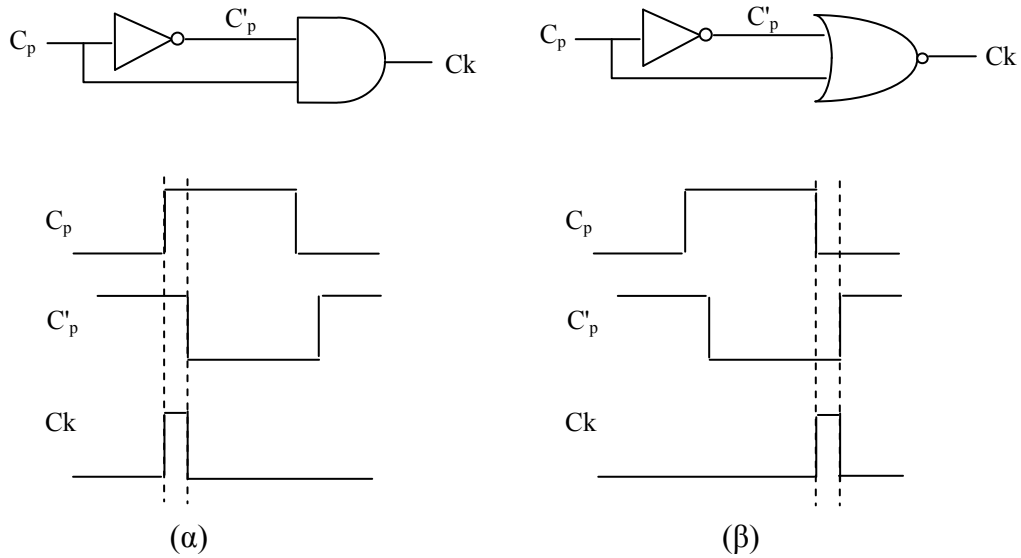
Το απλούστερο κύκλωμα για την ανίχνευση των ακμών είναι ένα RC κύκλωμα, το οποίο δρα ως κύκλωμα διαφορίσης των παλμών ρολογιού, Σχ. 6.18. Η θετική μετάβαση του παλμού δημιουργεί ένα θετικό σπινθήρα, ενώ η αρνητική μετάβαση δημιουργεί ένα αρνητικό σπινθήρα. Οι τιμές των R και C καθορίζουν τη σταθερά χρόνου εκφόρτισης του πυκνωτή και κατά συνέπεια το πόσο “λεπτός” είναι ο σπινθήρας. Ένα τέτοιο κύκλωμα ανίχνευσης ακμών, αν και απλό στη σύλληψή του, δεν είναι καλό από άποψη κατασκευής σε ολοκληρωμένη μορφή, διότι παρουσιάζει μεγάλη ανομοιομορφία σε σχέση με τα υπόλοιπα ψηφιακά κυκλώματα ενός ολοκληρωμένου κυκλώματος που είναι απλές πύλες.



Σχ. 6.18. Ανίχνευση ακμών μέσω ενός απλού RC κυκλώματος διαφορίσης.

Έτσι, στην πράξη χρησιμοποιούνται ως ανιχνευτές ακμών άλλα κυκλώματα κατασκευασμένα από πύλες NAND ή NOR, όπως αυτά του Σχ. 6.19. Η όλη λειτουργία τους βασίζεται στο γεγονός της καθυστέρησης διάδοσης ενός σήματος μέσω μιας πύλης. Η καθυστέρηση αυτή είναι της τάξης των nsec. Έτσι, αναφερόμενοι στο Σχ. 6.19α, όταν η είσοδος C_p γίνει “1” η έξοδος C_k της πύλης AND θα γίνει επίσης “1” για ένα πολύ μικρό χρονικό διάστημα. Αυτό οφείλεται στο ότι και οι δύο εισοδοί της πύλης AND βρίσκονται σε κατάσταση “1”. Πολύ σύντομα όμως (σε χρόνο $<10\text{nsec}$) η έξοδος του αντιστροφέα, ανταποκρινόμενη στο λογικό “1” της εισόδου, γίνεται “0” με αποτέλεσμα η πύλη AND να επιστρέψει στο λογικό “0”. Με το κύκλωμα αυτό επιτυγχάνουμε να ανιχνεύσουμε τη θετική μετάβαση του παλμού ρολογιού. Με ένα παρόμοιο απλό κύκλωμα μπορούμε να ανιχνεύσουμε την αρνητική μετάβαση του παλμού ρολογιού, όπως φαίνεται στο Σχ. 6.19β. Οι παραγόμενοι

παλμοί ρολογιού έχουν εύρος παλμών μικρότερο των 10 nsec, ανάλογα με την τεχνολογία κατασκευής των πυλών.

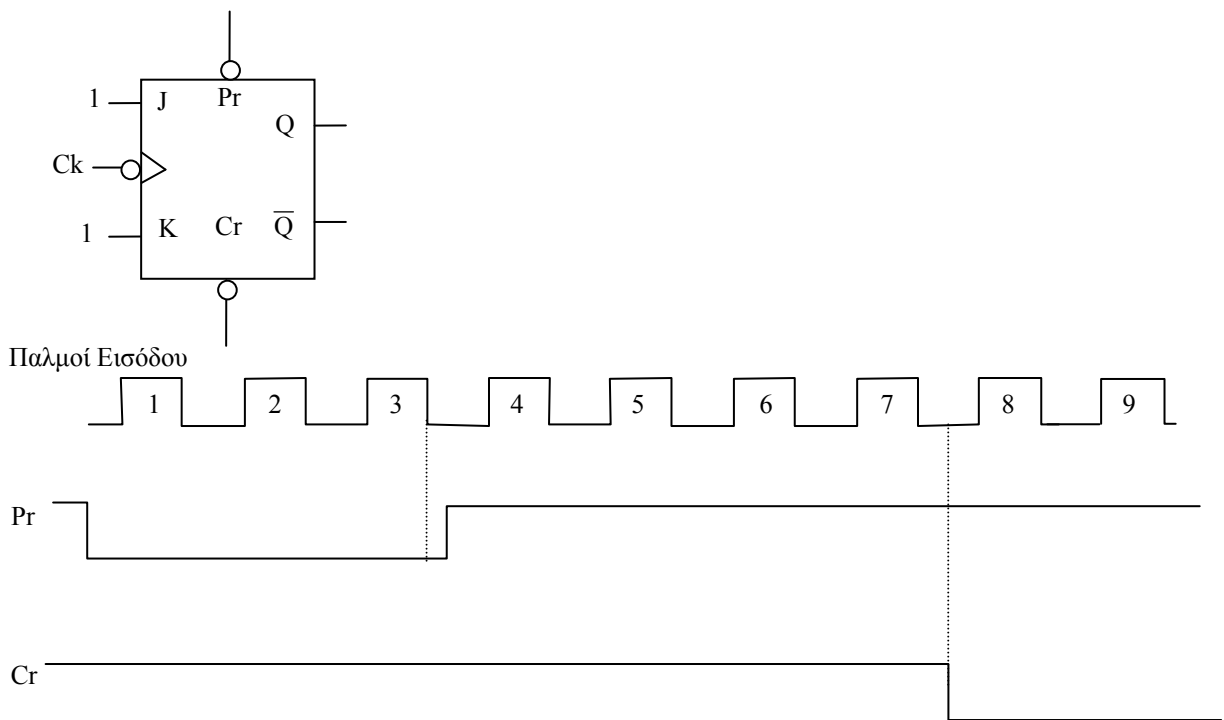


Σχ. 6.19. Κυκλώματα αντίληψης (α) των θετικών και (β) των αρνητικών ακμών για την πυροδότηση των flip-flops.

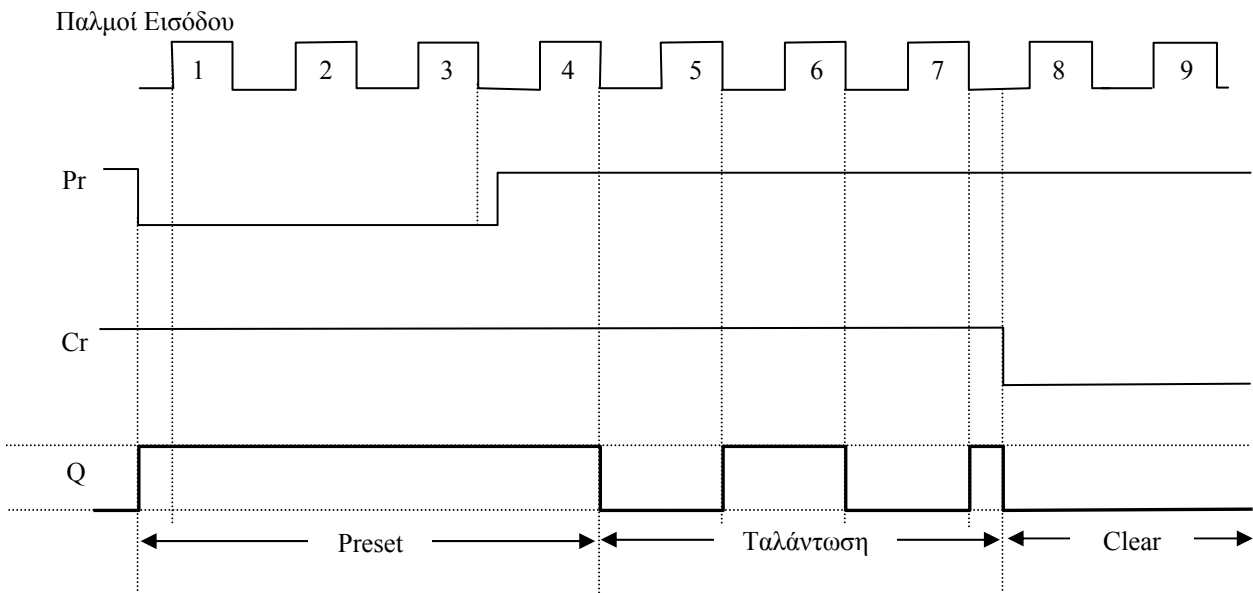
Από εδώ και στο εξής, θα υποθέτουμε ότι όλα τα flip-flop στα ακολουθιακά κυκλώματα με ρολόι είναι ακμοπυροδοτούμενα.

Παράδειγμα 6.4

Σχεδιάστε την κυματομορφή εξόδου Q του Flip-Flop J-K του οποίου οι κυματομορφές Clock, Preset και Clear φαίνονται παρακάτω.



Λύση



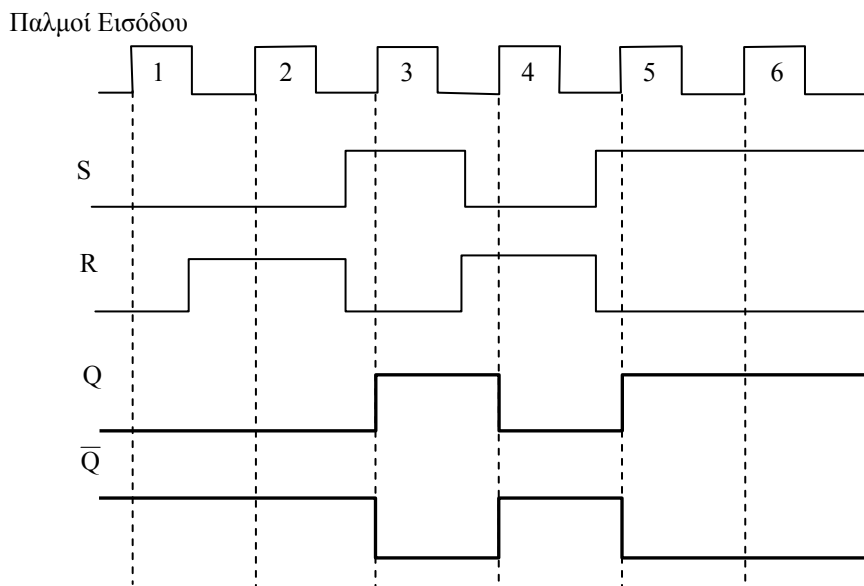
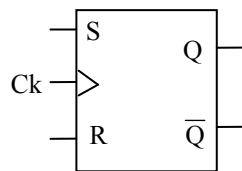
Παρατηρούμε ότι οι ασύγχρονες εισόδους Pr και Cr επηρεάζουν τη λειτουργία του flip-flop ανεξάρτητα από τις άλλες εισόδους ή την κατάσταση του flip-flop.

Παράδειγμα 6.5

Σχεδιάστε τις κυματομορφές εξόδου Q και \bar{Q} του Flip-Flop S-R του οποίου οι κυματομορφές Clock, S και R φαίνονται παρακάτω. Θεωρήστε ότι οι αλλαγές κατάστασης του Flip-Flop συμβαίνουν στη θετική ακμή κάθε παλμού.

Λύση

Με βάση τον πίνακα αληθείας του Flip-Flop S-R θα έχουμε:



Παράδειγμα 6.6

Να σχεδιαστεί κύκλωμα που υλοποιεί ένα Flip-Flop T με τη βοήθεια Flip-Flop S-R.

Λύση

Κατ' αρχήν θα καταστρώσουμε τον πίνακα συσχέτισης των εισόδων ενός Flip-Flop T και των αντίστοιχων εισόδων του Flip-Flop S-R, που έχουν το ίδιο αποτέλεσμα στην κατάσταση Q_{n+1} . Δηλαδή, θα καταγράψουμε όλους τους δυνατούς συνδυασμούς των εισόδων S, R και της προηγούμενης κατάστασης Q_n που παράγουν συγκεκριμένο αποτέλεσμα για την επόμενη κατάσταση Q_{n+1} , μαζί με τους αντίστοιχους συνδυασμούς της εισόδου T που παράγει το ίδιο αποτέλεσμα Q_{n+1} .

Από τους πίνακες αληθείας των Flip-Flop T και S-R, έχουμε:

Πίνακας Συσχέτισης

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

Από τον πίνακα συσχέτισης, εξάγουμε τις εξισώσεις των S, R συναρτήσεων των T και Q_n με τη βοήθεια του χάρτη Karnaugh.

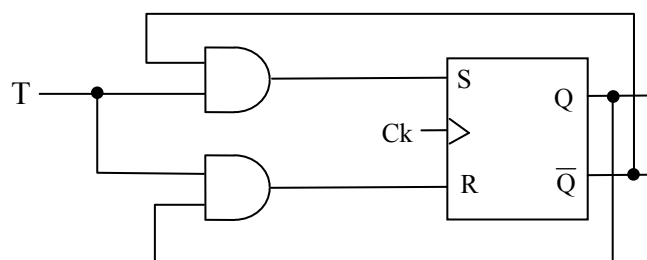
$Q_n \backslash T$	0	1
0	0	1
1	X	0

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

$Q_n \backslash T$	0	1
0	X	0
1	0	1

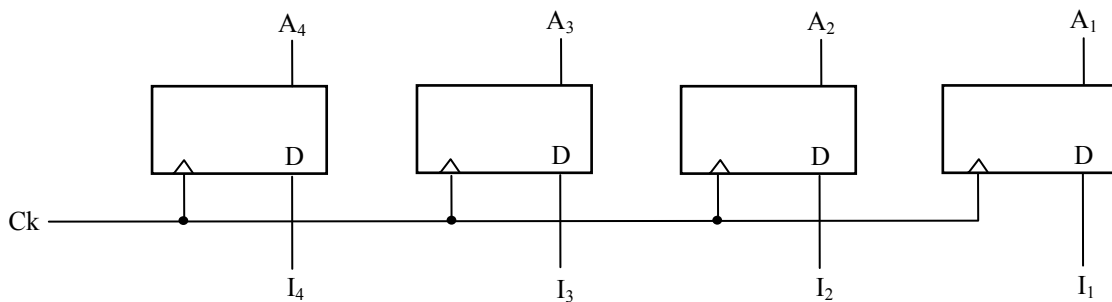
$$R = TQ_n$$

Με βάση τα παραπάνω, σχεδιάζουμε το τελικό κύκλωμα:



6.2. Καταχωρητές (Registers)

Ένας καταχωρητής (register) είναι μια ομάδα από δυαδικά κύτταρα αποθήκευσης, που είναι κατάλληλα για να κρατάνε δυαδικές πληροφορίες. Μια ομάδα flip-flops αποτελεί έναν καταχωρητή, αφού κάθε flip-flop μπορεί να αποθηκεύσει ένα bit πληροφορίας. Ένας καταχωρητής των n bits περιέχει n flip-flops και, άρα, είναι σε θέση να αποθηκεύσει οποιαδήποτε δυαδική πληροφορία περιέχει n bits. Στο Σχ. 6.20 φαίνεται ένας καταχωρητής 4 flip-flops τύπου D με μια κοινή είσοδο ρολογιού. Αυτή η είσοδος (Ck) πυροδοτεί όλα τα flip-flops, έτσι ώστε οι πληροφορίες που βρίσκονται εκείνη τη στιγμή στις 4 εισόδους τους, να μεταφερθούν μέσα στον καταχωρητή 4 bits. Από τις 4 εξόδους μπορούμε, όποτε το θελήσουμε, να πάρουμε τις πληροφορίες που βρίσκονται εκείνη τη στιγμή αποθηκευμένες στον καταχωρητή.



Σχ. 6.20. Ένας καταχωρητής 4 bits (το τρίγωνο συμβολίζει την είσοδο Ck στο flip-flop).

Ο τρόπος που πυροδοτούνται τα flip-flops είναι πρωταρχικής σημασίας. Εάν τα flip-flops είναι τύπου D, τότε οι πληροφορίες που βρίσκονται στις εισόδους D μεταφέρονται στις εξόδους Q, όταν η είσοδος επίτρεψης Ck είναι 1, και οι εξοδοί Q ακολουθούν τις εισόδους D, όση ώρα το Ck παραμένει στο 1. Όταν το Ck πέσει στο 0, οι πληροφορίες που βρίσκονταν στις εισόδους δεδομένων D αμέσως πριν τη μετάβαση αυτή κρατιούνται στις εξόδους Q.

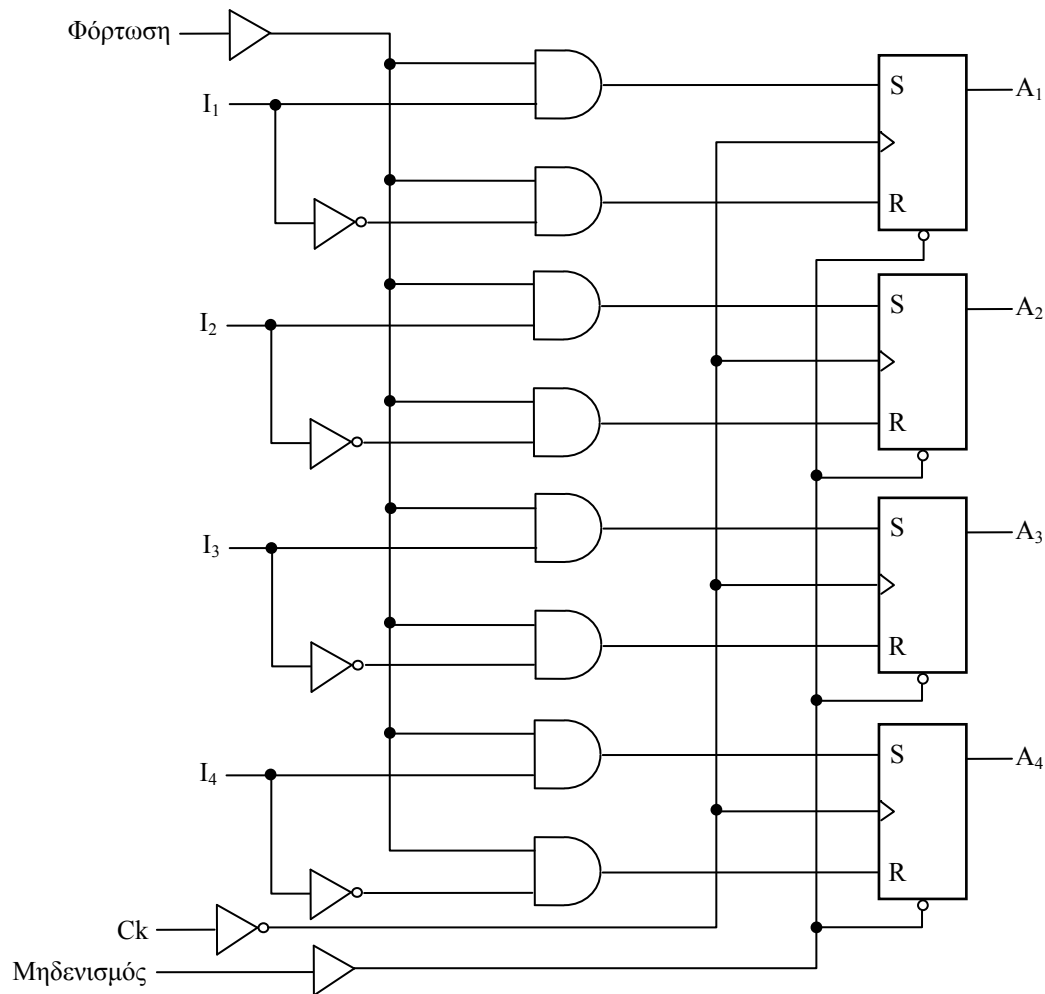
6.2.1. Καταχωρητές με παράλληλη φόρτωση

Η μεταφορά των πληροφοριών μέσα σε έναν καταχωρητή λέγεται “φόρτωση” (loading) του καταχωρητή. Αν όλα τα bits ενός καταχωρητή φορτώνονται ταυτόχρονα, μ’ έναν μόνο παλμό του ρολογιού, λέμε ότι η φόρτωση γίνεται “παράλληλα”. Εάν εφαρμόσουμε έναν παλμό στην είσοδο Ck του καταχωρητή του Σχ. 6.20, θα φορτωθούν παράλληλα και οι 4 εισόδοι. Με αυτή την κατασκευή, αν θέλουμε να αφήσουμε το περιεχόμενο του καταχωρητή αναλλοίωτο, πρέπει να απαγορέψουμε στον παλμό του ρολογιού να φθάσει στον ακροδέκτη Ck. Με άλλα λόγια, η είσοδος Ck δρα ως σήμα επίτρεψης που ελέγχει τη φόρτωση των πληροφοριών στον καταχωρητή. Όταν το Ck παίρνει την τιμή 1, οι πληροφορίες εισόδου φορτώνονται. Εάν το Ck μείνει στο 0, τότε το περιεχόμενο του καταχωρητή δεν αλλάζει. Θεωρούμε ότι η αλλαγή κατάστασης των εξόδων γίνεται στη θετική ακμή του παλμού. Αν ένα flip-flop αλλάζει κατάσταση στην αρνητική ακμή, τότε βάζουμε ένα μικρό κύκλο κάτω από το τριγωνικό σύμβολο της εισόδου Ck.

Τα περισσότερα ψηφιακά συστήματα έχουν μια γεννήτρια για το κύριο ρολόι τους, η οποία τα τροφοδοτεί με μια συνεχή σειρά παλμών ρολογιού. Όλοι αυτοί οι παλμοί εφαρμόζονται σε όλα τα flip-flops και σε όλους τους καταχωρητές του συστήματος. Υπ’ αυτές τις συνθήκες, μετά, υπάρχει ένα ξεχωριστό σήμα ελέγχου που κανονίζει

ποιοι συγκεκριμένοι παλμοί του ρολογιού θα έχουν κάποια επίδραση σε έναν συγκεκριμένο καταχωρητή. Σε ένα τέτοιο σύστημα, πρέπει να περάσουμε πρώτα τους παλμούς του ρολογιού μέσα από μια πύλη AND με το σήμα ελέγχου και μετά την έξοδο της πύλης να την εφαρμόσουμε στον ακροδέκτη Ck του καταχωρητή του Σχ. 6.20. Όταν το σήμα ελέγχου είναι 0, η έξοδος της πύλης AND μένει στο 0, και οι αποθηκευμένες πληροφορίες παραμένουν αναλλοίωτες. Όταν το σήμα ελέγχου είναι 1, τότε ο παλμός του ρολογιού περνάει μέσα από την πύλη AND, φθάνει στον ακροδέκτη Ck, και προκαλεί τη φόρτωση νέων πληροφοριών. Μια τέτοια μεταβλητή ελέγχου ονομάζεται είσοδος “ελέγχου φόρτωσης” (load control).

Το Σχ. 6.21 δείχνει έναν καταχωρητή 4 bits, με είσοδο ελέγχου φόρτωσης, αποτελούμενο από flip-flop τύπου S-R. Η είσοδος Ck του καταχωρητή παίρνει συνεχείς, συγχρονισμένους παλμούς, οι οποίοι εφαρμόζονται σε όλα τα flip-flops. Ο αντιστροφέας πάνω στο Ck κάνει όλα τα flip-flops να πυροδοτούνται από την αρνητική ακμή των παλμών εισόδου. Ο σκοπός του αντιστροφέα είναι να μειώσει το φορτίο (ρεύμα και χωρητικότητα) στη γεννήτρια του κύριου ρολογιού. Αυτό το επιτυγχάνουμε, αφού τώρα ο ακροδέκτης Ck τροφοδοτεί μία πύλη μόνο (τον αντιστροφέα), ενώ αλλιώς θα τροφοδοτούσε 4 (ή 8...) εισόδους πυλών, αν συνδέονταν σε όλα τα flip-flops.

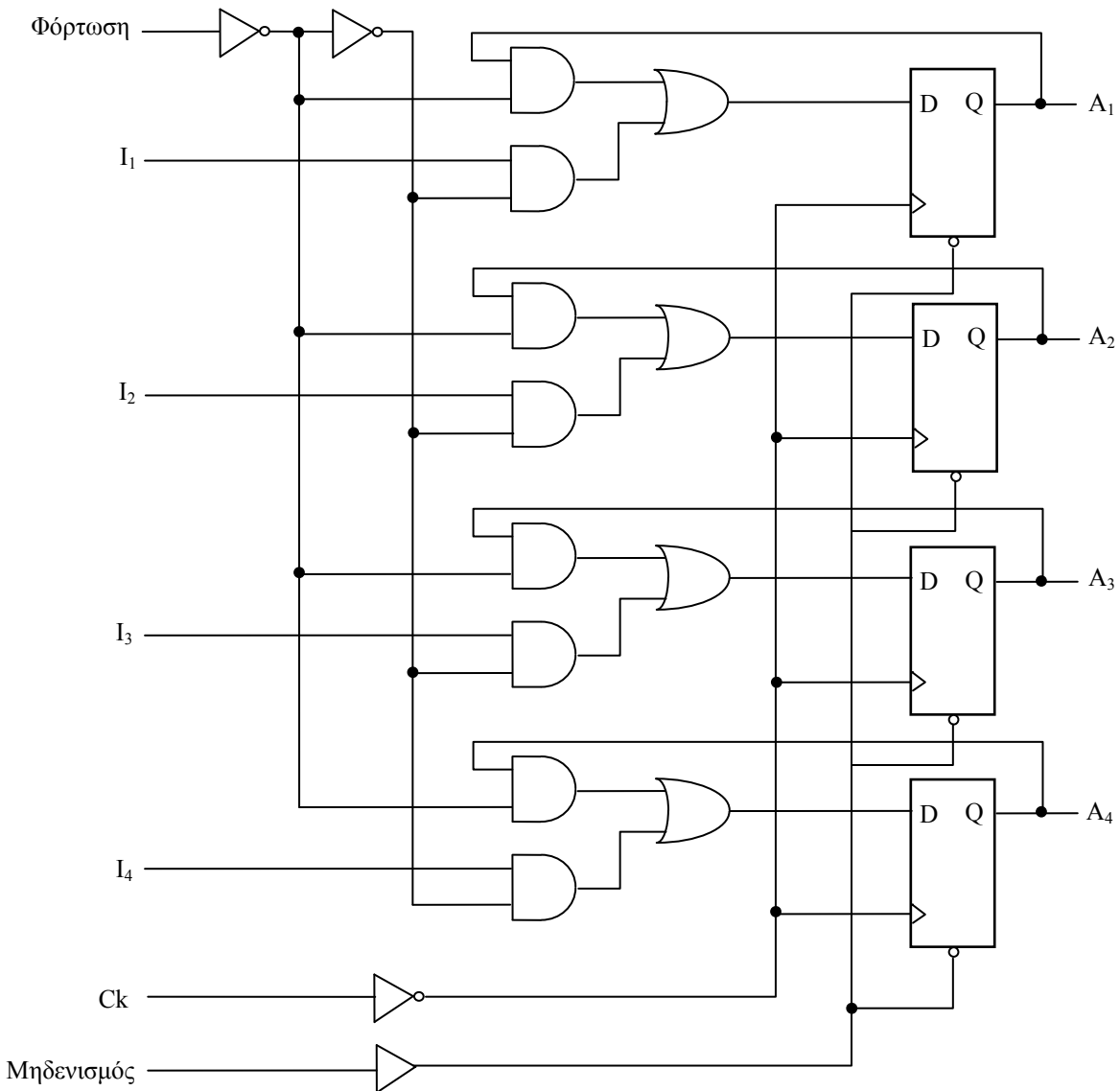


Σχ. 6.21. Καταχωρητής παράλληλης φόρτωσης 4 bits.

Η είσοδος εκκαθάρισης (Cr) πηγαίνει σε έναν ειδικό ακροδέκτη κάθε flip-flop, μέσω ενός μη-αντιστρέφοντος απομονωτή. Όταν αυτός ο ακροδέκτης γίνεται 0, το

flip-flop μηδενίζεται ασύγχρονα. Η είσοδος αυτή είναι χρήσιμη για το μηδενισμό (εκκαθάριση) του καταχωρητή, πριν από τη λειτουργία με το ρολοί. Η είσοδος εκκαθάρισης πρέπει να κρατιέται στο 1 κατά τη διάρκεια της κανονικής λειτουργίας με το ρολοί.

Η είσοδος φόρτωσης (load) συνδέεται μέσω ενός απομονωτή (για μείωση φορτίου) και μέσω μιας σειράς πυλών AND με τις εισόδους S και R των flip-flops. Αν και οι παλμοί του ρολογιού έρχονται συνεχώς, εκείνη που τελικά ελέγχει τη λειτουργία του καταχωρητή είναι η είσοδος φόρτωσης. Οι τιμές των R και S για κάθε flip-flop καθορίζονται από τις δύο πύλες AND και τον αντιστροφέα που συνδέονται σε κάθε είσοδο I. Εάν η είσοδος φόρτωσης είναι 0, τόσο τα R όσο και τα S είναι 0, και η κατάσταση δεν μπορεί να αλλάξει με κανέναν παλμό του ρολογιού. Δηλαδή, η είσοδος φόρτωσης είναι μια μεταβλητή ελέγχου η οποία μπορεί να εμποδίσει οποιαδήποτε μεταβολή πληροφοριών στον καταχωρητή όσο μένει στο 0.



Σχ. 6.22. Καταχωρητής παράλληλης φόρτωσης με flip-flops τύπου D.

Όταν η είσοδος φόρτωσης γίνει 1, οι εισόδους I₁ έως I₄ καθορίζουν τις πληροφορίες που θα φορτωθούν στον καταχωρητή με τον επόμενο παλμό του ρολογιού. Για κάθε

είσοδο I που είναι 1, το αντίστοιχο flip-flop δέχεται εισόδους $S = 1$ και $R = 0$. Για κάθε I που είναι 0, έχουμε $S = 0$ και $R = 1$.

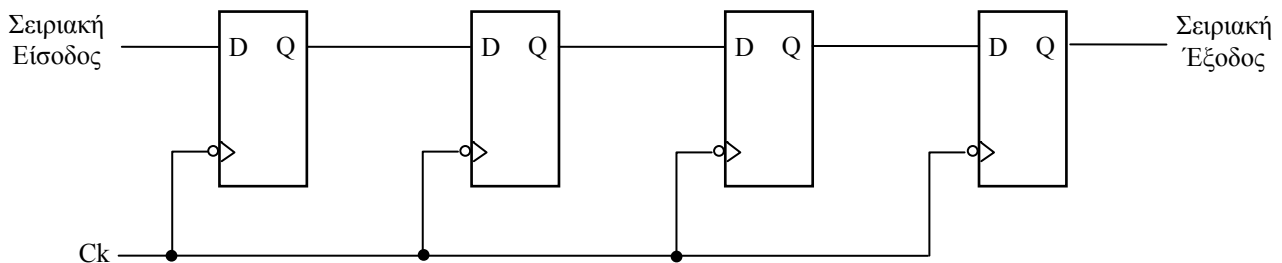
Άρα, η τιμή της εισόδου μεταφέρεται μέσα στον καταχωρητή, όταν η είσοδος φόρτωσης είναι 1, η είσοδος εκκαθάρισης είναι 1 και ο παλμός του ρολογιού πέφτει από το 1 στο 0. Αυτή η μεταφορά λέγεται “παράλληλη φόρτωση” (parallel load), γιατί όλα τα bits του καταχωρητή φορτώνονται ταυτόχρονα. Εάν ο απομονωτής που συνδέεται στην είσοδο φόρτωσης αντικατασταθεί με έναν αντιστροφέα, τότε ο καταχωρητής θα φορτώνεται, όταν η είσοδος αυτή είναι 0, και θα εμποδίζεται, όταν αυτή είναι 1.

Ένας καταχωρητής παράλληλης φόρτωσης μπορεί να υλοποιηθεί και με flip-flop τύπου D, όπως δείχνει το Σχ. 6.22. Οι εισόδοι ρολογιού και εκκαθάρισης είναι οι ίδιες όπως πριν. Όταν η είσοδος φόρτωσης είναι 1, οι εισόδοι I φορτώνονται στον καταχωρητή με τον επόμενο παλμό του ρολογιού. Όταν η είσοδος φόρτωσης είναι 0, οι εισόδοι του κυκλώματος εμποδίζονται, και τα flip-flops D ξαναφορτώνονται με τις παρούσες τιμές τους, διατηρώντας συνεπώς το περιεχόμενο του καταχωρητή. Η σύνδεση ανατροφοδότησης σε κάθε flip-flop είναι απαραίτητη, γιατί τα flip-flops τύπου D δεν έχουν συνθήκη εισόδου “μη αλλαγής” - με τον κάθε παλμό του ρολογιού, η είσοδος D καθορίζει την επόμενη έξοδο. Για να μείνει η έξοδος αμετάβλητη πρέπει να κάνουμε την είσοδο D ίση με την παρούσα έξοδο Q για κάθε flip-flop.

6.2.2. Καταχωρητές με σειριακή φόρτωση ή Καταχωρητές ολίσθησης

Ένας καταχωρητής που μπορεί να “ολισθαίνει” τις πληροφορίες που περιέχει προς τη μία ή προς την άλλη κατεύθυνση λέγεται *καταχωρητής ολίσθησης* (shift register). Ένας τέτοιος καταχωρητής αποτελείται από μια αλυσίδα από flip-flops συνδεδεμένα στη σειρά, με την έξοδο του ενός να τροφοδοτεί την είσοδο του γειτονικού του. Όλα τα flip-flops δέχονται ένα κοινό ρολόϊ, το οποίο προκαλεί την ολίσθηση από τη μια βαθμίδα στην επόμενη.

Ο απλούστερος δυνατός καταχωρητής ολίσθησης είναι εκείνος που έχει μόνο flip-flops, όπως φαίνεται στο Σχ. 6.23. Η έξοδος Q του κάθε flip-flop συνδέεται στην είσοδο D του δεξιού του γειτονικού flip-flop. Με κάθε παλμό του ρολογιού, το περιεχόμενο του καταχωρητή ολισθαίνει προς τα δεξιά, κατά μία θέση bit. Η σειριακή είσοδος κανονίζει το τί θα μπει στο ακραίο αριστερό flip-flop στη διάρκεια της ολίσθησης.

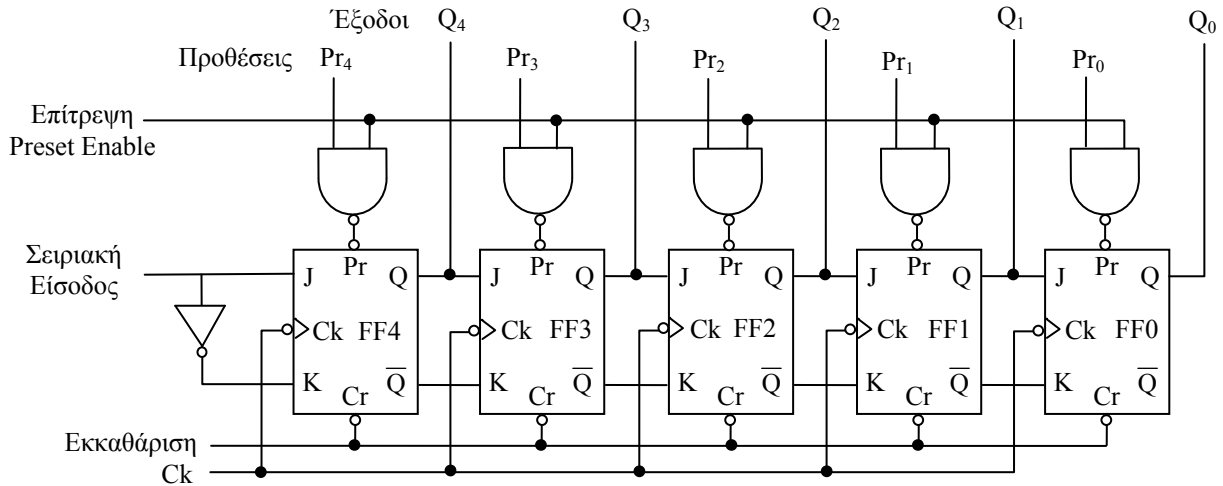


Σχ. 6.23. Καταχωρητής ολίσθησης.

Τη σειριακή έξοδο την παίρνουμε από την έξοδο του ακραίου δεξιού flip-flop, πριν από την εφαρμογή του παλμού του ρολογιού. Παρόλο που αυτός ο καταχωρητής ολισθαίνει το περιεχόμενό του προς τα δεξιά, αν τον αναποδογυρίσουμε θα το ολισθαίνει προς τα αριστερά. Άρα ένας μονοκατευθυντικός καταχωρητής ολίσθησης

μπορεί να λειτουργήσει ή σαν ολισθητής προς τα δεξιά ή σαν ολισθητής προς τα αριστερά.

Ο καταχωρητής του Σχ. 6.23 ολισθαίνει το περιεχόμενό του με τον κάθε παλμό του ρολογιού, στην αρνητική του ακμή. Εάν θέλουμε να ελέγξουμε την ολίσθηση ώστε να συμβαίνει με μερικούς μόνο παλμούς και όχι με άλλους, πρέπει να ελέγξουμε την είσοδο Ck του καταχωρητή.



Σχ. 6.24. Καταχωρητής σειριακής και παράλληλης φόρτωσης.

Στο Σχ. 6.24 φαίνεται ένας καταχωρητής 5 bits ο οποίος μπορεί να χρησιμοποιηθεί τόσο για σειριακή όσο και για παράλληλη φόρτωση. Οι ακροδέκτες εξόδου του κάθε flip-flop συνδέονται κατευθείαν με τους ακροδέκτες εισόδου του επόμενου flip-flop. Το πρώτο από αριστερά flip-flop 4 το οποίο αντιστοιχεί στο μέγιστο σημαντικό ψηφίο του αριθμού έχει στην είσοδό του μεταξύ J_4 και K_4 μία πύλη NOT. Κατ' αυτόν τον τρόπο όταν στην είσοδο J_4 θέσουμε το ψηφίο 1 ή 0 τότε αμέσως η K_4 θα δεχθεί το αντίθετό του.

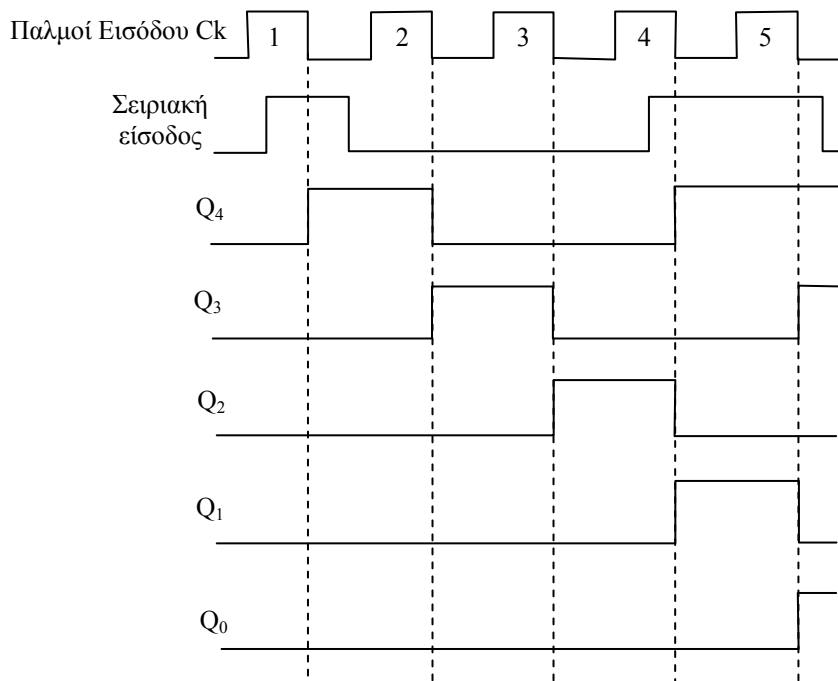
Ας εξετάσουμε τώρα τη διαδικασία εισαγωγής ενός δυαδικού αριθμού π.χ. του 11001 στον καταχωρητή. Κατ' αρχήν γίνεται εκκαθάριση του καταχωρητή με την επιβολή ενός 0 στην είσοδο Cr. Έτσι, από όλες τις εξόδους Q_0 ως Q_4 θα ληφθεί 0. Στη συνέχεια στον ακροδέκτη Cr επιβάλλεται ένα 1.

Όπως ήδη είπαμε, με το κύκλωμα του Σχ. 6.24 είναι δυνατή η καταχώρηση ενός δυαδικού αριθμού είτε παράλληλα είτε σειριακά. Η παράλληλη καταχώρηση γίνεται με τη βοήθεια του κυκλώματος πρόθεσης στο οποίο χρησιμοποιούνται πέντε πύλες NAND. Η μία είσοδος όλων των πυλών αυτών συνδέεται με τον κοινό κόμβο που παίρνει τον χαρακτηρισμό “επιτρέπεται η πρόθεση” (Preset Enable). Όταν στον ακροδέκτη αυτόν επιβάλλεται ένα 0 τότε κλείνουν όλες οι πύλες NAND ανεξάρτητα από την κατάσταση της δεύτερης εισόδου κάθε μιας από τις πύλες αυτές, δηλαδή των εισόδων Pr_0 έως Pr_4 . Στην περίπτωση αυτή στις εισόδους Pr όλων των flip-flops επιβάλλεται το 1.

Η παράλληλη εισαγωγή ενός δυαδικού αριθμού στον καταχωρητή γίνεται με την εφαρμογή του αριθμού στους ακροδέκτες πρόθεσης Pr_4 έως Pr_0 . Όταν ο ακροδέκτης Preset-Enable τεθεί στην κατάσταση 1, εκεί όπου ο αριθμός Pr_4 Pr_3 Pr_2 Pr_1 Pr_0 έχει ψηφίο 1 μεταφέρεται στον ακροδέκτη Pr του αντίστοιχου flip-flop το 0 οπότε σύμφωνα με τον πίνακα αληθείας του Σχ. 6.8 το αντίστοιχο flip-flop τίθεται στην κατάσταση 1. Το αντίστροφο συμβαίνει εκεί όπου ο αριθμός ο οποίος εισάγεται

παράλληλα έχει ψηφίο 0. Η παράλληλη καταχώρηση είναι ασύγχρονη δηλαδή γίνεται ανεξάρτητα από τους παλμούς του ρολογιού Ck της εισόδου.

Παλμοί ρολογιού	Ψηφία αριθμού	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀
1	1	1	0	0	0	0
2	0	0	1	0	0	0
3	0	0	0	1	0	0
4	1	1	0	0	1	0
5	1	1	1	0	0	1



Σχ. 6.25. Σειριακή καταχώρηση δυαδικού αριθμού και παλμοσειρές σειριακής καταχώρησής του.

Η εισαγωγή σε σειρά ενός δυαδικού αριθμού γίνεται με την εφαρμογή των ψηφίων του αριθμού στην είσοδο σειράς διαδοχικά, αρχίζοντας από το ελάχιστο σημαντικό ψηφίο (LSB: Least Significant Bit). Για να γίνει αυτό εφαρμόζεται το LSB του αριθμού στην είσοδο J₄, στην περίπτωση μας του παραδείγματος το 1 του αριθμού 11001. Το ψηφίο αυτό με τον πρώτο παλμό του ρολογιού, κατά τη διάρκεια του οποίου το Ck αλλάζει από 1 σε 0, καταχωρείται στο flip-flop 4, δηλαδή το Q₄ αλλάζει από 0 σε 1 όπως φαίνεται και στο Σχ. 6.25. Στη συνέχεια, εφαρμόζεται το δεύτερο σημαντικό ψηφίο στην είσοδο J₄, στην περίπτωση μας το 0 του αριθμού 11001, το οποίο με το δεύτερο παλμό του ρολογιού καταχωρείται στο flip-flop 4, δηλαδή το Q₄ αλλάζει σε 0, συγχρόνως δε προωθείται το 1 από το flip-flop 4 στο flip-flop 3, εφόσον η έξοδος του Q₄ του πρώτου συνδέεται με την είσοδο J₃ του δεύτερου.

Κατ' αυτόν τον τρόπο προωθείται ολόκληρος ο αριθμός ο οποίος πρόκειται να καταχωρηθεί ψηφίο προς ψηφίο μέχρι και το μέγιστο σημαντικό ψηφίο (MSB: Most Significant Bit) και καταλαμβάνει τις πέντε θέσεις του καταχωρητή.

Η διάταξη αυτή ονομάζεται καταχωρητής ολίσθησης (shift register), επειδή η είσοδος του αριθμού γίνεται σ' αυτόν με ολίσθηση των ψηφίων από αριστερά προς τα δεξιά. Το κύκλωμα αυτό μπορεί να χρησιμοποιηθεί και για τις εξής λειτουργίες:

- 1) Μετατροπή πληροφοριών από μορφή παράλληλη σε σειριακή και αντιστρόφως (parallel to serial converter, serial to parallel converter).
- 2) Καταχωρητής δεξιάς ή αριστερής ολίσθησης (right-shift, left-shift register).

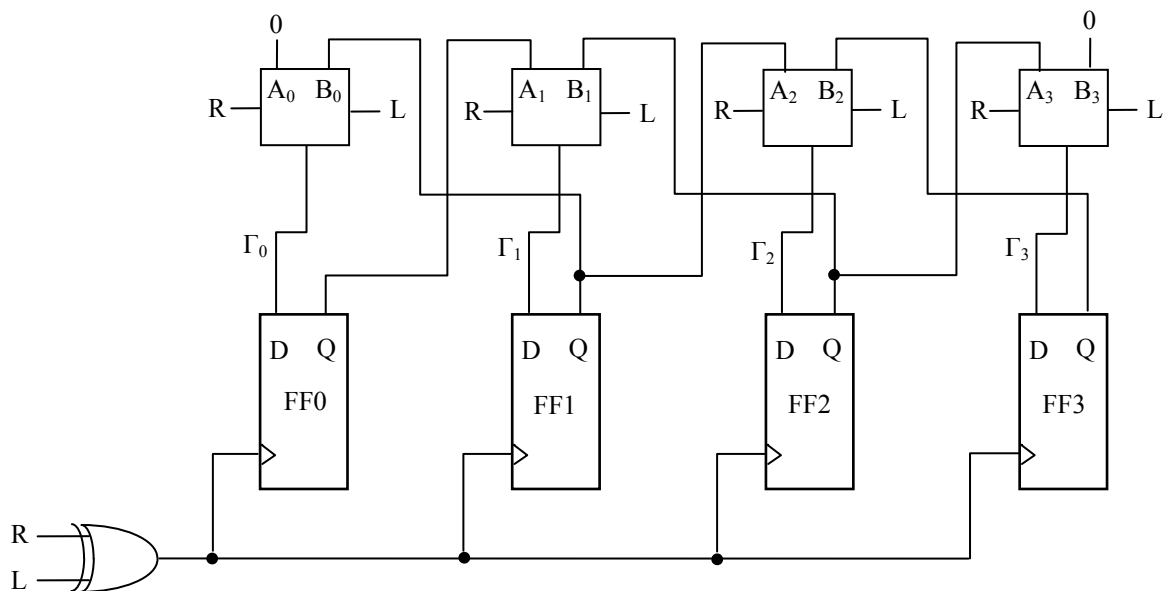
Παράδειγμα 6.7

Να σχεδιαστεί καταχωρητής τεσσάρων bits ο οποίος όταν το σήμα ελέγχου $L = 1$ ολισθαίνει μία θέση προς τα αριστερά και αντιστρόφως ολισθαίνει μία θέση προς τα δεξιά όταν $R = 1$.

Πώς τροποποιείται το κύκλωμα όταν ο καταχωρητής είναι n bits και έχουμε n σήματα ελέγχου L_i και n σήματα ελέγχου R_i για τα οποία ισχύει: $L_i = 1 \rightarrow$ ολίσθηση κατά i θέσεις αριστερά, $R_i = 1 \rightarrow$ ολίσθηση κατά i θέσεις δεξιά.

Τροποποιήστε το αρχικό κύκλωμα έτσι ώστε ο καταχωρητής εκτός από τα σήματα ελέγχου L και R να ελέγχεται και από τρίτο σήμα W το οποίο όταν $W = 1$ επιτρέπει την παράλληλη καταχώρηση από 4 γραμμές δεδομένων. Επίσης, θα πρέπει όταν $R = L = W = 0$ να μην επηρεάζεται ο καταχωρητής, και όταν εφαρμόζονται ταυτόχρονα περισσότερα από ένα σήματα ελέγχου να μην εφαρμόζεται καμία είσοδος στον καταχωρητή.

Λύση:



Σχ. 6.26. Κυκλωματική υλοποίηση του καταχωρητή.

Η κυκλωματική υλοποίηση του καταχωρητή φαίνεται στο Σχ. 6.26. Τα συνδυαστικά κυκλώματα στο επάνω μέρος του σχήματος υλοποιούν τις συναρτήσεις

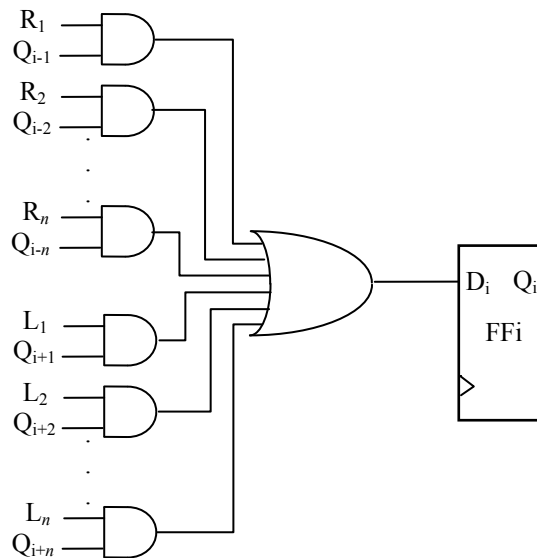
$$\Gamma_i = RA_i + LB_i \quad i = 0, 1, 2, 3$$

Εύκολα διαπιστώνουμε ότι όταν $R = 1$, το περιεχόμενο της εξόδου Q κάθε flip-flop τύπου D μετατοπίζεται μία θέση προς τα δεξιά. Οι είσοδοι R και L συνδέονται στην είσοδο ρολογιού κάθε flip-flop μέσω μιας πύλης XOR ώστε να είναι αμοιβαία αποκλειόμενες μεταξύ τους. Ομοίως, όταν $L = 1$, το περιεχόμενο της εξόδου Q κάθε flip-flop τύπου D μετατοπίζεται μία θέση προς τα αριστερά. Σε κάθε μετατόπιση προς τα αριστερά ή προς τα δεξιά, στη θέση του bit μικρότερης σημαντικότητας τίθεται μέσω του συνδυαστικού κυκλώματος η τιμή 0.

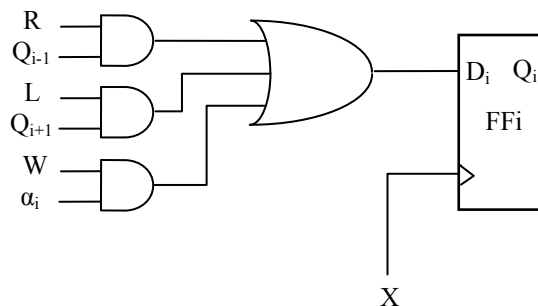
Στην περίπτωση n σημάτων ελέγχου η κυκλωματική υλοποίηση τροποποιείται για κάθε flip-flop ως ακολούθως:

Το Σχ. 6.27 αποτελεί ουσιαστικά επέκταση του Σχ. 6.26 για καταχωρητή n bits και για λόγους σχεδιαστικής ευκολίας δίνουμε μόνο την κυκλωματική υλοποίηση ενός flip-flop.

Τέλος, δίνεται στο Σχ. 6.28 η τροποποίηση του αρχικού κυκλώματος προκειμένου να μπορούμε να επιβάλλουμε και τρίτο σήμα ελέγχου W . Για να ικανοποιούνται δε, οι δύο τελευταίες απαιτήσεις της εκφώνησης, επιβάλλεται το σήμα ελέγχου καταχώρησης X , το οποίο συνδέεται στην είσοδο Ck κάθε flip-flop, όπως φαίνεται στο Σχ. 6.28, και προκύπτει από τον ακόλουθο πίνακα αληθείας.



Σχ. 6.27. Κυκλωματική τροποποίηση του καταχωρητή για περισσότερα σήματα ελέγχου.



Σχ. 6.28. Τροποποίηση του αρχικού κυκλώματος για σήματα ελέγχου W και X .

R	L	W	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$$X = \bar{R} \bar{L} W + \bar{R} L \bar{W} + R \bar{L} \bar{W}$$

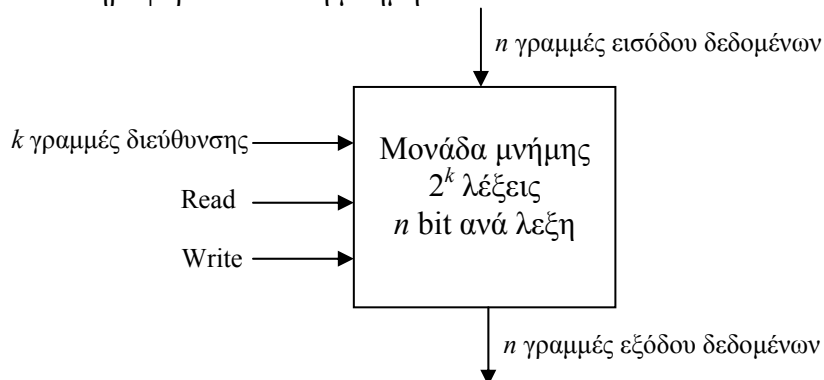
Σύμφωνα με τον πίνακα αυτό, όταν $R = L = W = 0$ δεν επηρεάζεται ο καταχωρητής διότι $X = 0$, και όταν εφαρμόζονται ταυτόχρονα περισσότερα από ένα σήματα ελέγχου, δεν εφαρμόζεται καμία είσοδος στον καταχωρητή διότι και πάλι $X = 0$. Για οποιοδήποτε άλλο συνδυασμό το σήμα ελέγχου X έχει τιμή 1, επιτρέποντας την ομαλή λειτουργία των σημάτων ελέγχου R , L , και W .

Τα δεδομένα παράλληλης καταχώρησης μέσω του σήματος ελέγχου W , συμβολίζονται στο Σχ. 6.28 με a_i . Επίσης, στο Σχ. 6.28 για λόγους σχεδιαστικής ευκολίας παραλείπουμε το συνδυαστικό κύκλωμα υλοποίησης της συνάρτησης X .

6.3. Η μνήμη τυχαίας προσπέλασης (RAM)

Μία μονάδα μνήμης είναι ένα σύνολο από κύτταρα αποθήκευσης, μαζί με τα απαραίτητα κυκλώματα για τη μεταφορά των πληροφοριών μέσα και έξω από τη συσκευή. Επειδή μπορεί να γίνει προσπέλαση στα κύτταρα μνήμης, για μεταφορά πληροφορίας από και προς οποιαδήποτε επιθυμητή τυχαία θέση, προέκυψε το όνομα “μνήμη τυχαίας προσπέλασης” (Random Access Memory, RAM).

Οι λέξεις μνήμης αποτελούνται από bits. Μία ομάδα από 8 bits ονομάζεται byte. Οι περισσότερες μνήμες υπολογιστών χρησιμοποιούν λέξεις που έχουν μήκος πολλαπλάσιο των 8 bits. Η χωρητικότητα μιας μνήμης ορίζεται από το συνολικό αριθμό των bytes που μπορεί να αποθηκεύσει. Η επικοινωνία μεταξύ της μονάδας μνήμης και του περιβάλλοντός της επιτυγχάνεται με γραμμές εισόδου και εξόδου δεδομένων, γραμμές επιλογής διεύθυνσης και γραμμές ελέγχου, που καθορίζουν τη διεύθυνση μεταφοράς. Στο Σχ. 6.29 φαίνεται ένα σχηματικό διάγραμμα μνήμης RAM. Οι n γραμμές εισόδου δεδομένων παρέχουν την πληροφορία που θα αποθηκευθεί στη μνήμη και οι n γραμμές εξόδου δεδομένων την πληροφορία που εξέρχεται από τη μνήμη. Οι k γραμμές διεύθυνσης προσδιορίζουν τη συγκεκριμένη λέξη που επιλέχθηκε ανάμεσα στις πολλές διαθέσιμες. Οι 2 είσοδοι ελέγχου καθορίζουν την επιθυμητή κατεύθυνση μεταφοράς: η είσοδος write προκαλεί μεταφορά δυαδικών πληροφοριών προς τη μνήμη, ενώ η είσοδος read μεταφορά δυαδικών πληροφοριών από τη μνήμη.



Σχ. 6.29. Σχηματικό διάγραμμα μνήμης RAM.

Μια μονάδα μνήμης περιγράφεται από τον αριθμό των λέξεων που περιέχει και τον αριθμό των bits ανά λέξη. Οι γραμμές διεύθυνσης επιλέγουν μια συγκεκριμένη λέξη. Κάθε λέξη της μνήμης εφοδιάζεται με έναν αριθμό αναγνώρισης, που καλείται διεύθυνση και ο οποίος ξεκινά από το 0 και συνεχίζει με 1, 2, 3 μέχρι το 2^k-1 , όπου k είναι ο αριθμός των γραμμών διεύθυνσης. Η επιλογή μιας συγκεκριμένης λέξης στη μνήμη γίνεται με την εφαρμογή της δυαδικής διεύθυνσης των k bits στις γραμμές διεύθυνσης. Ένας αποκωδικοποιητής στο εσωτερικό της μνήμης δέχεται τη διεύθυνση και ανοίγει τους διαύλους που χρειάζονται για την επιλογή της συγκεκριμένης λέξης. Οι μνήμες των υπολογιστών κυμαίνονται από 1024 λέξεις – που απαιτούν διεύθυνση των 10 bits – μέχρι 2^{32} λέξεις – που απαιτούν 32 bits διεύθυνσης. Συνηθίζεται να αναφερόμαστε στον αριθμό των λέξεων (ή bytes) στη μνήμη με ένα από τα γράμματα K (kilo), M (mega), ή G (giga). Το K είναι ισοδύναμο με 2^{10} , το M με 2^{20} και το G με 2^{30} . Έτσι, έχουμε $64K = 2^{16}$, $2M = 2^{21}$ και $4G = 2^{32}$.

Διεύθυνση μνήμης		Περιεχόμενα μνήμης
Δυαδικό	Δεκαδικό	
0000000000	0	1011010101011100
0000000001	1	1010101110001001
0000000010	2	0000110101000110
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
1111111101	1021	1001110100010101
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

Σχ. 6.30. Περιεχόμενα μιας μνήμης RAM 1024×16.

Έστω για παράδειγμα, μια RAM με 1K λέξεων των 16 bits η καθεμία. Αφού $1K = 1024 = 2^{10}$ και 16 bits συνιστούν 2 bytes, μπορούμε να πούμε ότι η μνήμη χωράει $2048 = 2K$ bytes. Το Σχ. 6.30 δείχνει το πιθανό περιεχόμενο των 3 πρώτων και των 3 τελευταίων λέξεων της μνήμης. Κάθε λέξη περιέχει 16 bits, που μπορούν να χωριστούν σε 2 bytes. Οι λέξεις αναγνωρίζονται από τη δεκαδική τους διεύθυνση, που κυμαίνεται από 0 έως 1023. Η ισοδύναμη δυαδική διεύθυνση αποτελείται από 10 bits. Η πρώτη διεύθυνση δηλώνεται με δέκα 0, ενώ η τελευταία διεύθυνση με 10 1. Αυτό συμβαίνει, γιατί στο δυαδικό σύστημα το 1023 είναι ισοδύναμο με 1111111111. Μία λέξη της μνήμης επιλέγεται με βάση τη δυαδική της διεύθυνση. Όταν μια λέξη διαβάζεται ή γράφεται, η μνήμη συμπεριφέρεται στα 16 bits σαν να αποτελούν μια ενιαία μονάδα.

Η μνήμη των 1K×16 του Σχ. 6.30 έχει 10 bits στη διεύθυνση και 16 bits σε κάθε λέξη. Αντίστοιχα, μια μνήμη των 64K×10 θα έχει 16 bits στη διεύθυνση (αφού $64K = 2^{16}$) και κάθε λέξη θα αποτελείται από 10 bits. Ο αριθμός των bits της διεύθυνσης, που απαιτούνται σε μια μνήμη, εξαρτάται από το συνολικό αριθμό λέξεων που μπορούν να αποθηκευτούν στη μνήμη και είναι ανεξάρτητος από τον αριθμό των bits

της κάθε λέξης. Ο αριθμός των bits στη διεύθυνση καθορίζεται από τη σχέση $2^k = m$, όπου m είναι ο συνολικός αριθμός των λέξεων και k ο αριθμός των bits της διεύθυνσης.

6.4. Απαριθμητές (Counters)

Ένα ακολουθιακό κύκλωμα που περνάει από μια προδιαγεγραμμένη ακολουθία καταστάσεων, όταν του εφαρμόζουμε παλμούς στην είσοδο, ονομάζεται “απαριθμητής” (counter). Οι παλμοί εισόδου, που τους ονομάζουμε “παλμούς μέτρησης” (count pulses), μπορεί να είναι παλμοί ρολογιού ή να προέρχονται από κάποια εξωτερική πηγή και μπορεί να έρχονται σε κανονικά ή σε ακανόνιστα διαστήματα. Σε έναν απαριθμητή η ακολουθία των καταστάσεων μπορεί να είναι η δυαδική σειρά μέτρησης ή μια οποιαδήποτε άλλη σειρά. Απαριθμητές βρίσκουμε σχεδόν μέσα σε όλα τα ψηφιακά συστήματα. Τους χρησιμοποιούμε για να μετράμε πόσες φορές συμβαίνει κάποιο γεγονός ή για τη δημιουργία ακολουθιών χρονισμού για τον έλεγχο των λειτουργιών ενός ψηφιακού συστήματος.

Οι απαριθμητές απαρτίζονται από flip-flops κατάλληλα συνδεδεμένα μεταξύ τους, και ανήκουν σε δύο κατηγορίες: τους *ασύγχρονους* απαριθμητές και τους *σύγχρονους* απαριθμητές. Στους ασύγχρονους απαριθμητές, οι ακμές των εξόδων των flip-flops χρησιμεύουν ως πηγές πυροδότησης άλλων flip-flops. Με άλλα λόγια, οι εισοδοί Ck όλων των flip-flops (εκτός του πρώτου) πυροδοτούνται όχι από τους παλμούς του ρολογιού, αλλά από τις ακμές (μεταβάσεις) των κυματομορφών που βγαίνουν από άλλα flip-flops. Στους σύγχρονους απαριθμητές, όλες οι εισοδοί Ck όλων των flip-flops τροφοδοτούνται από τους ίδιους παλμούς του ρολογιού. Σ’ αυτούς, η αλλαγή κατάστασης κάθε συγκεκριμένου flip-flop εξαρτάται από την παρούσα κατάσταση των άλλων flip-flops.

Στη συνέχεια, θα μελετήσουμε αναλυτικότερα και τους δύο αυτούς τύπους απαριθμητών.

6.4.1. Ασύγχρονοι απαριθμητές

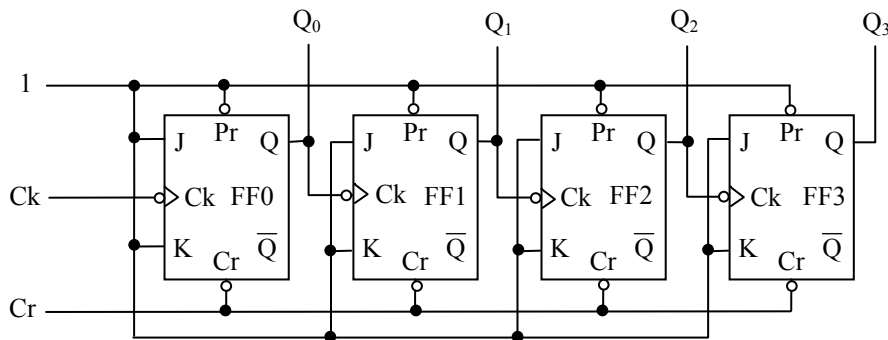
Στο Σχ. 6.31 φαίνεται η διάταξη ενός ασύγχρονου απαριθμητή με τον οποίο είναι δυνατή η μέτρηση του πλήθους των παλμών οι οποίοι εφαρμόζονται κατά τη διάρκεια ενός χρονικού διαστήματος πάνω στον ακροδέκτη ρολογιού Ck ενός flip-flop J-K.

Στη διάταξη του Σχ. 6.31 έχουν συνδεθεί σε σειρά τέσσερα flip-flops J-K στα οποία όλοι οι ακροδέκτες εισόδου J και K είναι συνδεδεμένοι μονίμως στην κατάσταση 1 όπως επίσης και οι ακροδέκτες πρόθεσης Pr. Ο ακροδέκτης εκκαθάρισης Cr είναι στην κατάσταση 1 και τίθεται στην κατάσταση 0 κάθε φορά που επιθυμούμε να εκκαθαρίσουμε, δηλαδή να μηδενίσουμε, τον απαριθμητή. Τέλος, ο ακροδέκτης Q κάθε flip-flop συνδέεται με τον ακροδέκτη Ck του επόμενου.

Η είσοδος των παλμών οι οποίοι πρόκειται να μετρηθούν γίνεται από τον ακροδέκτη Ck του πρώτου flip-flop-0, στις δε εξόδους Q₀, Q₁, Q₂, Q₃, εμφανίζεται ο δυαδικός αριθμός ο οποίος εκφράζει το πλήθος των παλμών οι οποίοι παρουσιάστηκαν στην είσοδο από την αρχή της μέτρησης μέχρι τη στιγμή της ανάγνωσης. Πριν από την έναρξη της μέτρησης συνδέεται προς στιγμής στον ακροδέκτη εκκαθάρισης ο παλμός Cr = 0 για να μηδενιστούν όλες οι έξοδοι Q₀ ως Q₃. Στη συνέχεια, συνδέεται και πάλι ο ακροδέκτης Cr στο 1 και εισάγεται η προς μέτρηση παλμοσειρά στον ακροδέκτη Ck του flip-flop-0. Ο πίνακας αληθείας του Σχ. 6.31β δίνει τις καταστάσεις των flip-flops μετά από κάθε παλμό εισόδου.

Με τα τέσσερα αυτά flip-flops είναι δυνατή η καταμέτρηση μέχρι και του δέκατου έκτου παλμού, επειδή αυτός είναι ο μεγαλύτερος δεκαδικός αριθμός ο οποίος εκφράζεται με τέσσερα μόνον δυαδικά ψηφία. Το πλέον σημαντικό ψηφίο (MSB) αναγράφεται στο τελευταίο flip-flop-3 και το ελάχιστο (LSB) στο flip-flop-0.

Όπως είναι γνωστό με ένα παλμό ρολογιού $Ck = 1$ ανατρέπεται η κατάσταση ενός flip-flop J-K, και τούτο συντελείται κατά την πτώση (ακμή) της τάσης του παλμού αυτού, ενώ κατά την ύψωση της τάσης απλώς αποθηκεύεται η πληροφορία στο Master και δεν αλλάζει η κατάσταση της εξόδου Q. Στο κύκλωμα του Σχ. 6.31 λοιπόν, κατά την πτώση της τάσης του πρώτου παλμού η έξοδος του flip-flop-0, Q_0 αλλάζει κατάσταση και γίνεται $Q_0 = 1$. Τούτο δεν επιδρά όμως στην έξοδο Q_1 του flip-flop-1 επειδή έγινε μόνον ύψωση τάσης και όχι πτώση τάσης στον ακροδέκτη Ck του flip-flop-1. Στη συνέχεια, κατά την πτώση τάσης του δεύτερου παλμού ανατρέπεται πάλι η κατάσταση του flip-flop-0, δηλαδή γίνεται το $Q_0 = 0$, τώρα όμως τούτο επιδρά στο flip-flop-1 και ανατρέπει την κατάστασή του επειδή έγινε πτώση τάσης στον ακροδέκτη Ck του flip-flop-1. Τα ίδια συμβαίνουν και με τους επόμενους παλμούς και καταγράφεται το πλήθος τους στις εξόδους του κυκλώματος.



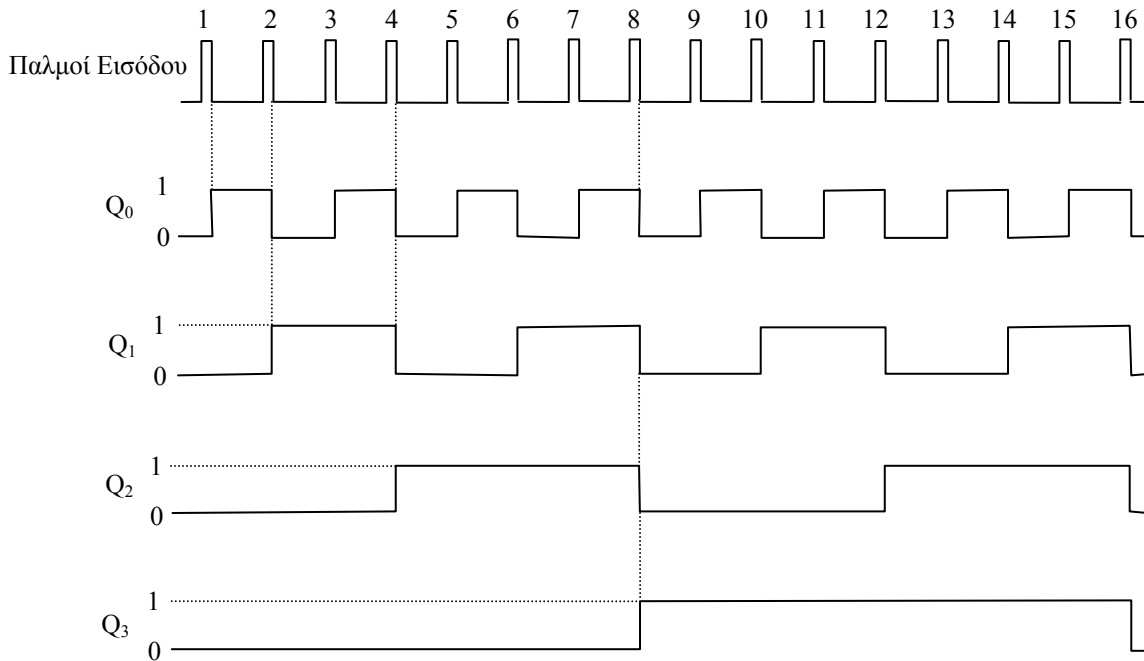
Σχ. 6.31α. Ασύγχρονος απαριθμητής.

Αριθμός παλμών εισόδου	Έξοδοι Flip-Flop			
	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

Σχ. 6.31β. Πίνακας αληθείας ασύγχρονου απαριθμητή.

Το Σχ. 6.32 παριστάνει τη μορφή των σημάτων εξόδου του ρολογιού και κάθε μιας εξόδου των flip-flops, Q_0, Q_1, Q_2, Q_3 .

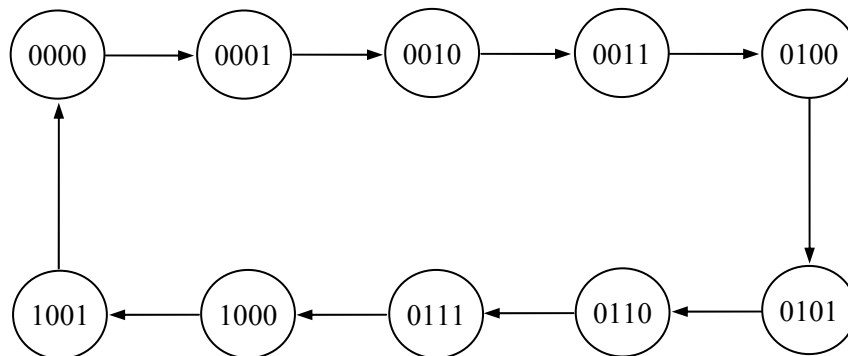
Με την πιο πάνω μέθοδο είναι δυνατό με τα τέσσερα flip-flops J-K του Σχ. 6.31 να γίνει καταμέτρηση έως και 16 παλμών όπως φαίνεται και από τον πίνακα αληθείας. Με αυτόν τον απαριθμητή είναι δυνατή η καταμέτρηση της συχνότητας ενός εναλλασσομένου ρεύματος με καταμέτρηση του πλήθους των περιόδων μέσα σε ένα ορισμένο χρονικό διάστημα, π.χ. ένα δευτερόλεπτο.



Σχ. 6.32. Διάγραμμα χρονισμού του ασύγχρονου απαριθμητή του Σχ. 6.31.

6.4.2. Ασύγχρονος δεκαδικός απαριθμητής

Ένας δεκαδικός απαριθμητής παίρνει μια ακολουθία δέκα καταστάσεων και ξανααγυρίζει στο 0 μετά το 9. Ένας τέτοιος απαριθμητής πρέπει να έχει τουλάχιστον 4 flip-flops, αφού τόσα bits χρειάζονται για την αναπαράσταση των δέκα δεκαδικών ψηφίων. Η ακολουθία των καταστάσεων ενός δεκαδικού απαριθμητή καθορίζεται από το δυαδικό κώδικα που χρησιμοποιούμε για την αναπαράσταση των ψηφίων του δεκαδικού συστήματος. Εάν χρησιμοποιούμε τον BCD (Binary-Coded-Decimal) κώδικα, η ακολουθία των καταστάσεων είναι όπως φαίνεται στο Σχ. 6.33. Αυτή η ακολουθία είναι όμοια με εκείνη των δυαδικών απαριθμητών, μόνο που μετά την κατάσταση 1001 (ο κώδικας του 9) πάει στην 0000 (ο κώδικας του 0).

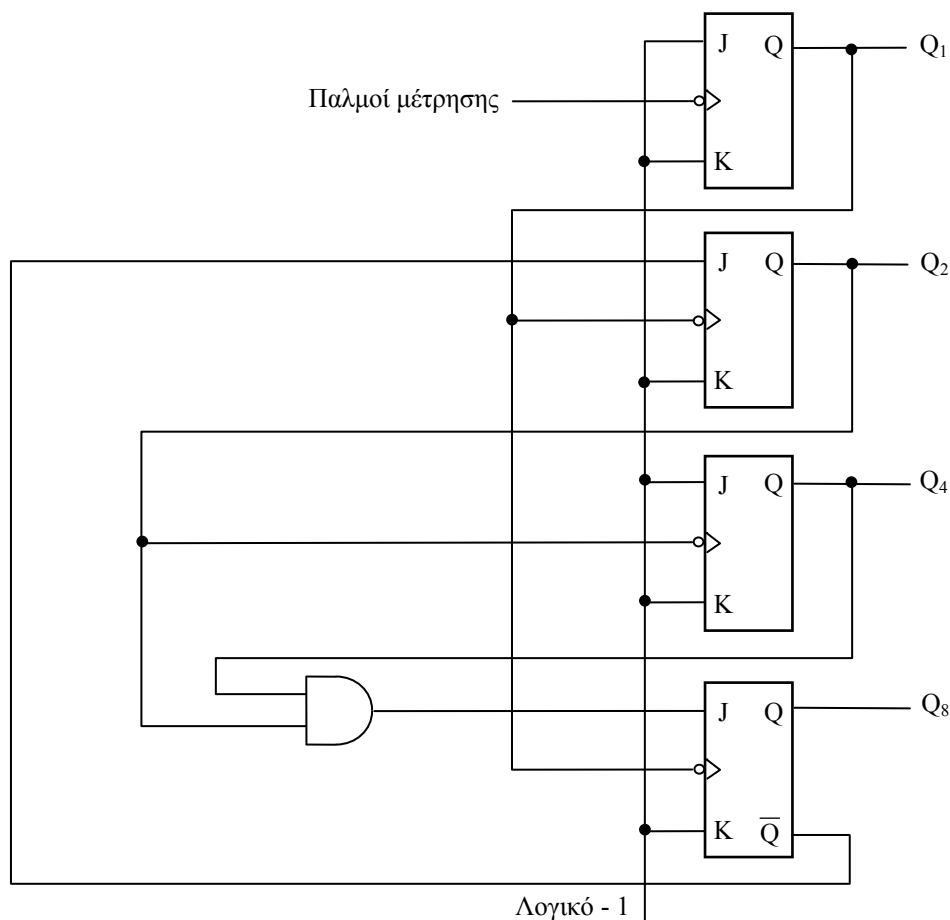


Σχ. 6.33. Ακολουθία καταστάσεων ενός δεκαδικού (BCD) απαριθμητή.

Η σχεδίαση ενός δεκαδικού απαριθμητή, και γενικότερα κάθε ασύγχρονου απαριθμητή που δεν ακολουθεί τη δυαδική σειρά, δεν είναι και τόσο εύκολη δουλειά. Οι τυπικές μέθοδοι λογικού σχεδιασμού μπορούν να χρησιμεύσουν μόνο σαν γενικός οδηγός. Για ικανοποιητικό τελικό αποτέλεσμα χρειάζεται βασικά εφευρετικότητα και φαντασία από την πλευρά του σχεδιαστή.

Το λογικό διάγραμμα ενός ασύγχρονου απαριθμητή BCD φαίνεται στο Σχ. 6.34. Οι 4 έξοδοι φαίνονται με το σύμβολο Q και με το βάρος τους στον κώδικα BCD για δείκτη. Τα flip-flops πυροδοτούνται με την αρνητική ακμή, δηλαδή, όταν το Ck πέφτει από το 1 στο 0. Παρατηρείστε ότι η έξοδος του Q₁ εφαρμόζεται στην είσοδο Ck του Q₄. Οι είσοδοι J και K των flip-flops συνδέονται είτε σε σταθερό 1 είτε στις εξόδους άλλων flip-flops, όπως βλέπουμε στο διάγραμμα.

Οι ασύγχρονοι απαριθμητές είναι ακολουθιακά κυκλώματα που δεν μπορούν να περιγραφούν με τις εξισώσεις Boole και τις μεθόδους περιγραφής των ακολουθιακών κυκλωμάτων με ρολόι. Οι μεταβολές των flip-flops εξαρτώνται από τη σειρά με την οποία αλλάζουν μεταξύ 0 και 1 τα σήματα που ελέγχουν το κάθε flip-flop. Η λειτουργία του απαριθμητή μπορεί να εξηγηθεί με έναν κατάλογο των συνθηκών για τις αλλαγές των flip-flops. Αυτές οι συνθήκες βγαίνουν από το λογικό διάγραμμα και από το πώς δουλεύουν τα flip-flops τύπου J-K. Θυμηθείτε ότι όταν το Ck πέφτει από 1 σε 0, το flip-flop γίνεται 1 αν J = 1, μηδενίζεται αν K = 1, συμπληρώνεται αν J = K = 1, και μένει αμετάβλητο αν J = K = 0. Οι συνθήκες για τις μεταβολές καταστάσεων των flip-flops έχουν ως εξής:



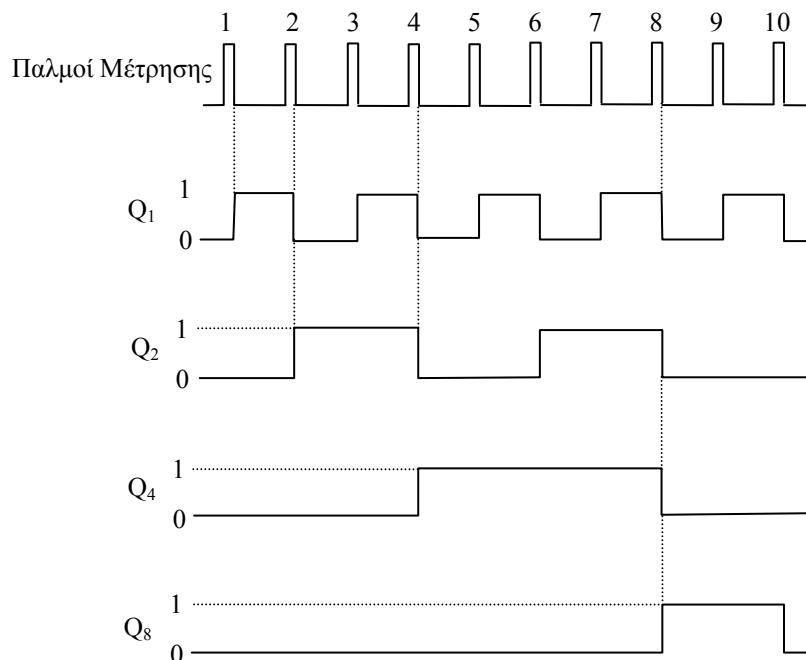
Σχ. 6.34. Λογικό διάγραμμα ασύγχρονου δεκαδικού απαριθμητή (BCD).

1. Το Q₁ αντιστρέφεται στην αρνητική ακμή του κάθε παλμού μέτρησης.

2. Το Q_2 αντιστρέφεται αν το $Q_8 = 0$ και το Q_1 πέφτει από 1 σε 0. Το Q_2 μηδενίζεται αν το $Q_8 = 1$ και το Q_1 πέφτει από 1 σε 0.
3. Το Q_4 αντιστρέφεται όταν το Q_2 πέφτει από 1 σε 0.
4. Το Q_8 αντιστρέφεται όταν $Q_4Q_2 = 11$ και το Q_1 πέφτει από 1 σε 0. Το Q_8 μηδενίζεται εάν το Q_4 ή το Q_2 είναι 0 και το Q_1 πέφτει από 1 σε 0.

Τώρα πρέπει να επιβεβαιώσουμε αν όντως αυτές οι συνθήκες κάνουν τα flip-flops να ακολουθούν το διάγραμμα καταστάσεων του Σχ. 6.33, ούτως ώστε το σύστημα να λειτουργεί ως απαριθμητής BCD. Ένας άλλος τρόπος να επιβεβαιώσουμε τη λειτουργία του απαριθμητή είναι να εξάγουμε το διάγραμμα χρονισμού του κάθε flip-flop από τις συνθήκες που αναφέραμε παραπάνω.

Το διάγραμμα αυτό φαίνεται στο Σχ. 6.35, όπου και καταγράφονται οι καταστάσεις των flip-flops μετά από κάθε παλμό μέτρησης.

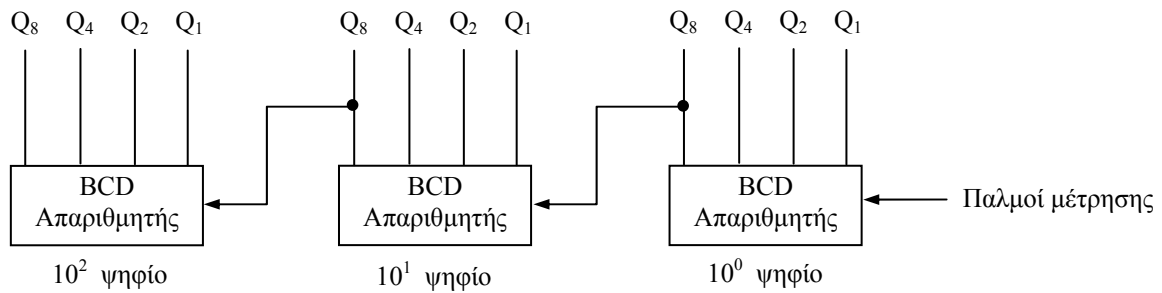


Σχ. 6.35. Διάγραμμα χρονισμού του δεκαδικού απαριθμητή του Σχ. 6.33.

Το Q_2 αντιστρέφεται κάθε φορά που το Q_1 πέφτει από το 1 στο 0, εφόσον το $Q_8 = 0$. Όταν το Q_8 γίνει 1, το Q_2 μένει μηδενισμένο. Το Q_4 αντιστρέφεται κάθε φορά που το Q_2 πάει από το 1 στο 0. Το Q_8 παραμένει μηδενισμένο όση ώρα το Q_2 ή το Q_4 είναι 0. Όταν και τα δύο τους γίνουν 1, το Q_8 αντιστρέφεται με το που το Q_1 πέφτει από το 1 στο 0. Το Q_8 μηδενίζεται με την επόμενη παρόμοια μετάβαση του Q_1 .

Ο απαριθμητής BCD του Σχ. 6.34 είναι απαριθμητής μιας “δεκάδας”, αφού μετράει από το 0 ως το 9. Για να μετρήσουμε στο δεκαδικό από το 0 ως το 99, χρειαζόμαστε ένα μετρητή δύο δεκάδων, από το 0 ως το 999 τριών δεκάδων, κ.ο.κ. Μπορούμε να φτιάξουμε απαριθμητές πολλαπλών δεκάδων συνδέοντας απαριθμητές BCD στη σειρά, έναν για κάθε δεκάδα. Στο Σχ. 6.36 βλέπουμε έναν απαριθμητή τριών δεκάδων. Οι είσοδοι της δεύτερης και της τρίτης δεκάδας έρχονται από το Q_8 της προηγούμενης δεκάδας. Όταν το Q_8 της μιας βαθμίδας πέφτει από το 1 στο 0, δηλαδή, όταν η δεκάδα αυτή πάει από το 9 στο 0, τότε πυροδοτείται η μέτρηση της

επόμενης σε σημαντικότητα βαθμίδας. Έτσι, για παράδειγμα, μετά το 399 ο απαριθμητής πάει στο 400.

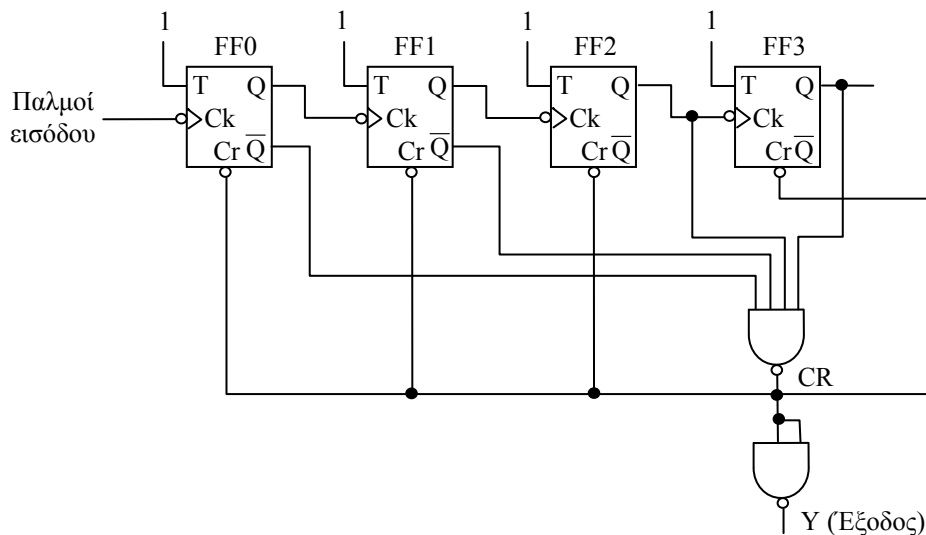


Σχ. 6.36. Σχηματικό διάγραμμα δεκαδικού απαριθμητή BCD 3 δεκάδων.

Παράδειγμα 6.8: Να σχεδιαστεί κύκλωμα που να διαιρεί τη συχνότητα του σήματος ρολογιού δια 12.

Λύση:

Το κύκλωμα που πραγματοποιεί τη ζητούμενη διαίρεση δίνεται στο Σχ. 6.37.



Σχ. 6.37. Κύκλωμα διαίρεσης συχνότητας ρολογιού.

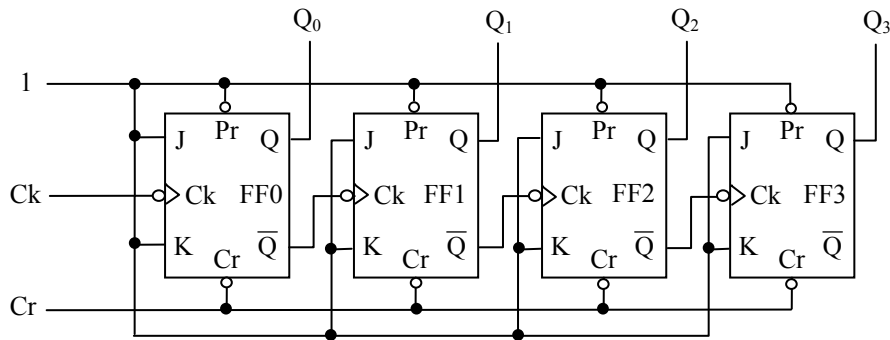
Το κύκλωμα λειτουργεί ως εξής: Κατ' αρχήν χρησιμοποιούμε ένα ασύγχρονο απαριθμητή 4 bits ($2^4 = 16 > 12$) ο οποίος μόλις μετρήσει τον 12ο παλμό θα έχει στις εξόδους του την ακολουθία 1100. Δηλαδή, οι εξοδοί των flip-flops μετά το 12ο παλμό θα είναι $Q_3Q_2Q_1Q_0 = 1100$. Έτσι, με τη συνδεσμολογία του Σχ. 6.37 μόνο μετά το 12ο παλμό η έξοδος CR της πρώτης πύλης NAND θα είναι 0, διότι $CR = Q_3 Q_2 \overline{Q_1} \overline{Q_0} = 0$, οπότε και θα γίνει εκκαθάριση όλων των flip-flops και θα αρχίσει ξανά η διαδικασία μέτρησης. Η έξοδος του κυκλώματος είναι η έξοδος Y της δεύτερης πύλης NAND, η οποία θα γίνει 1 μόλις η έξοδος CR γίνει 0, και στην οποία επομένως θα εμφανιστεί παλμός κάθε φορά που ο απαριθμητής θα έχει πρώτα μετρήσει 12 παλμούς ρολογιού. Κατ' αυτόν τον τρόπο επιτυγχάνουμε διαίρεση της συχνότητας ρολογιού διά 12.

Παράδειγμα 6.9

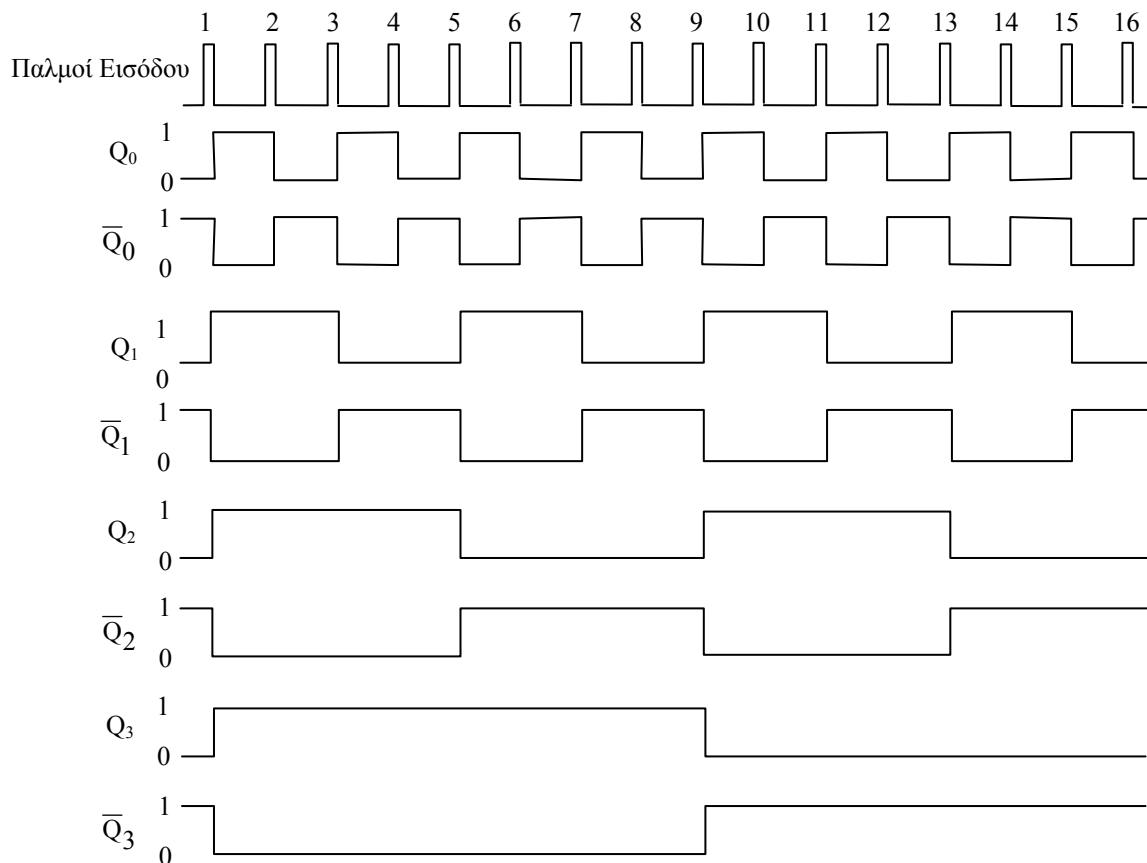
Να δοθεί το διάγραμμα χρονισμού ενός ασύγχρονου απαριθμητή 4 bits προς τα κάτω.

Λύση

Ένας απαριθμητής προς τα κάτω σχεδιάζεται όπως και ο απαριθμητής προς τα πάνω, μόνο που στην είσοδο ρολογιού κάθε flip-flop συνδέεται το συμπλήρωμα της εξόδου του flip-flop μικρότερης σημαντικότητας. Έτσι, το κυκλωματικό διάγραμμα ενός απαριθμητή προς τα κάτω είναι της μορφής του Σχ. 6.38.



Σχ. 6.38. Κυκλωματικό διάγραμμα ασύγχρονου απαριθμητή προς τα κάτω.



Σχ. 6.39. Διάγραμμα χρονισμού ασύγχρονου απαριθμητή προς τα κάτω.

Το διάγραμμα χρονισμού του ασύγχρονου απαριθμητή προς τα κάτω προκύπτει εύκολα από τους πίνακες αληθείας των flip-flops τύπου T και από όσα είπαμε για τους ασύγχρονους απαριθμητές στην παράγραφο 6.4.1.

Παράδειγμα 6.10: Να σχεδιαστεί δυαδικός απαριθμητής προς τα άνω με τις εξής προδιαγραφές:

- 1) Όταν δέχεται σήμα εισόδου $R = 1$ να μηδενίζεται.
- 2) Όταν δέχεται σήμα εισόδου $F = 1$ να “παγώνει” (δηλ. να σταματάει η λειτουργία του).
- 3) Όταν δέχεται σήματα εισόδου $R = F = 1$ να μην συμβαίνει τίποτα (δηλ. να αποκλείεται η ταυτόχρονη επενέργεια των δύο σημάτων ελέγχου F και R).
- 4) Όταν το δεκαδικό ισοδύναμο του δυαδικού απαριθμητή είναι $17 < N \leq 49$ να υπάρχει σήμα εξόδου $S = 1$.
- 5) Όταν $N = 49$ να μηδενίζεται ο απαριθμητής.

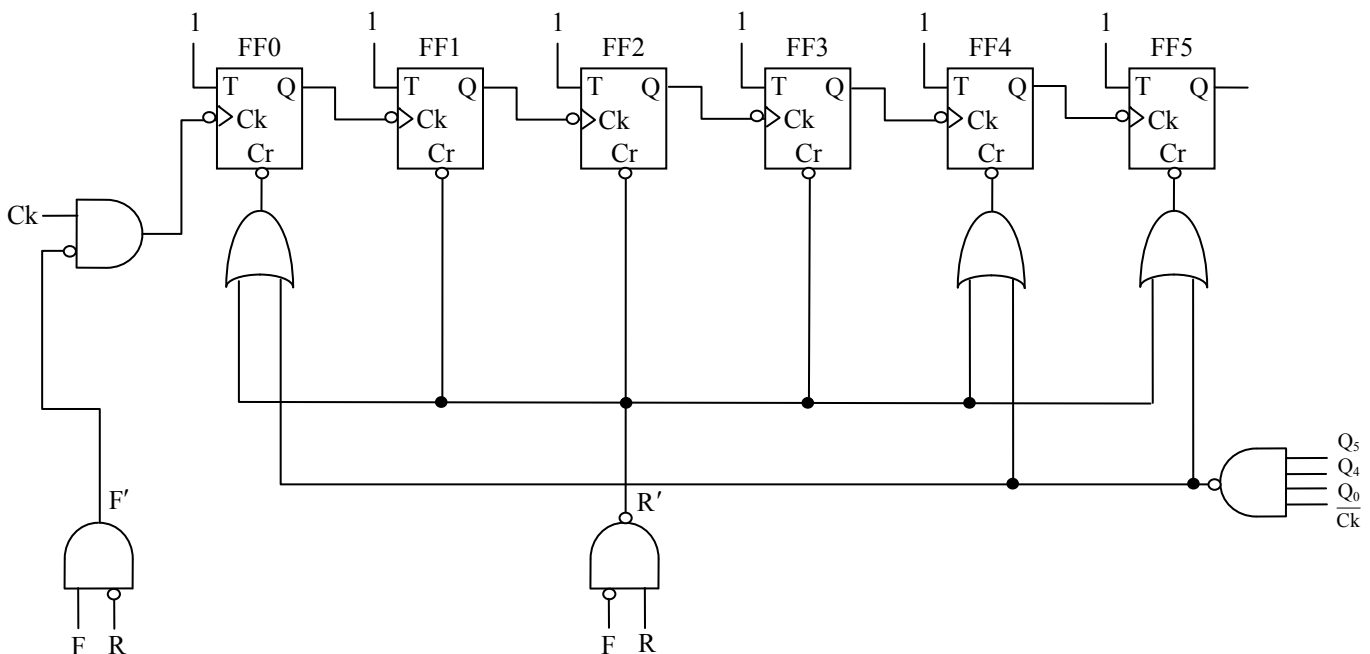
Λύση

Η λύση του παραδείγματος αυτού φαίνεται στο Σχ. 6.40. Από τα δεδομένα του παραδείγματος ο απαριθμητής, τον οποίο σχεδιάζουμε με flip-flops τύπου T, θα πρέπει να μπορεί να μετράει 49 παλμούς, να μηδενίζεται μόλις μετρηθεί και ο 49ος παλμός, και να αρχίζει ξανά τη διαδικασία μέτρησης. Άρα, ο ελάχιστος αριθμός flip-flops που πρέπει να διαθέτει είναι $n = 6$ (διότι $2^5 = 32 < 49$, $2^6 = 64 > 49$). Εφόσον επίσης τα σήματα ελέγχου F και R πρέπει να είναι αμοιβαία αποκλειόμενα θα πρέπει να ισχύει:

$$F' = (F \oplus R)F = (F\bar{R} + \bar{F}R)F = F\bar{R}$$

$$R' = \overline{(F \oplus R)R} = \overline{(F\bar{R} + \bar{F}R)R} = \bar{F}\bar{R} = F + \bar{R}$$

που υλοποιείται με τις πύλες AND και NAND που φαίνονται στο σχήμα. Έτσι, όταν $F = 1$ και $R = 0$, έχουμε $F' = 1$ και αποτρέπονται οι παλμοί εισόδου Ck λόγω της πύλης AND στο αριστερό άκρο του σχήματος και επομένως ο απαριθμητής “παγώνει”.

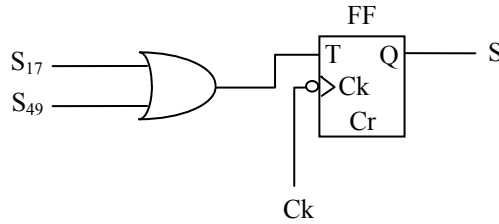


Σχ. 6.40. Υλοποίηση του κυκλώματος του Παραδείγματος 6.10.

Όταν $R = 1$ και $F = 0$, έχουμε $R' = 0$ και ενεργοποιούνται οι εισοδοί Clear (Cr) των flip-flops με αποτέλεσμα να μηδενίζεται ο απαριθμητής. Όταν $F = R = 1$, έχουμε $\bar{F} = \bar{R} = 0$, και επομένως δεν συμβαίνει τίποτα διότι ακυρώνεται η ταυτόχρονη επενέργεια

των δύο σημάτων ελέγχου και ο απαριθμητής συνεχίζει να απαριθμεί κανονικά με τους παλμούς εισόδου.

Η λύση του ερωτήματος (4) υλοποιείται πολύ εύκολα με το ακόλουθο σχήμα:



Σχ. 6.41. Υλοποίηση του κυκλώματος του ερωτήματος 6.9 (4).

Για το σχεδιασμό του πιο πάνω κυκλώματος του Σχ. 6.41 λαμβάνουμε υπόψη ότι η δυαδική απεικόνιση του 17 είναι $(17)_{10} = (010001)_2$ και του 49 είναι $(49)_{10} = (110001)_2$. Έτσι, εάν οι εισοδοί σε ένα flip-flop τύπου T μέσω της πύλης OR είναι οι S₁₇ και S₄₉ που ορίζονται ως εξής:

$$S_{17} = \overline{Q}_5 Q_4 \overline{Q}_3 \overline{Q}_2 \overline{Q}_1 Q_0 \quad S_{49} = Q_5 Q_4 \overline{Q}_3 \overline{Q}_2 \overline{Q}_1 Q_0$$

τότε, μόλις ο απαριθμητής μετρήσει τον 17ο παλμό, και σύμφωνα με τον πίνακα αληθείας ενός flip-flop τύπου T, η είσοδος S₁₇ θα γίνει 1 οπότε η έξοδος του flip-flop θα συμπληρωθεί και από S = 0 (θεωρούμε εδώ ότι στο flip-flop έχουμε κάνει αρχικά εκκαθάριση Clear) θα γίνει S = 1, θα παραμείνει δε ίση με 1 διότι μέχρι τον 49ο παλμό και οι δύο εισοδοί S₁₇ και S₄₉ είναι 0 (οπότε το flip-flop παραμένει στην προηγούμενη κατάσταση 1). Μόλις μετρηθεί ο 49ος παλμός η είσοδος S₄₉ γίνεται 1, οπότε η έξοδος του flip-flop συμπληρώνεται και από 1 γίνεται 0, όπως ακριβώς ήταν και το ζητούμενο της άσκησης. Σημειώνουμε ότι για λόγους ευκρίνειας του Σχ. 6.40 δεν περιλάβαμε το Σχ. 6.41 σε αυτό.

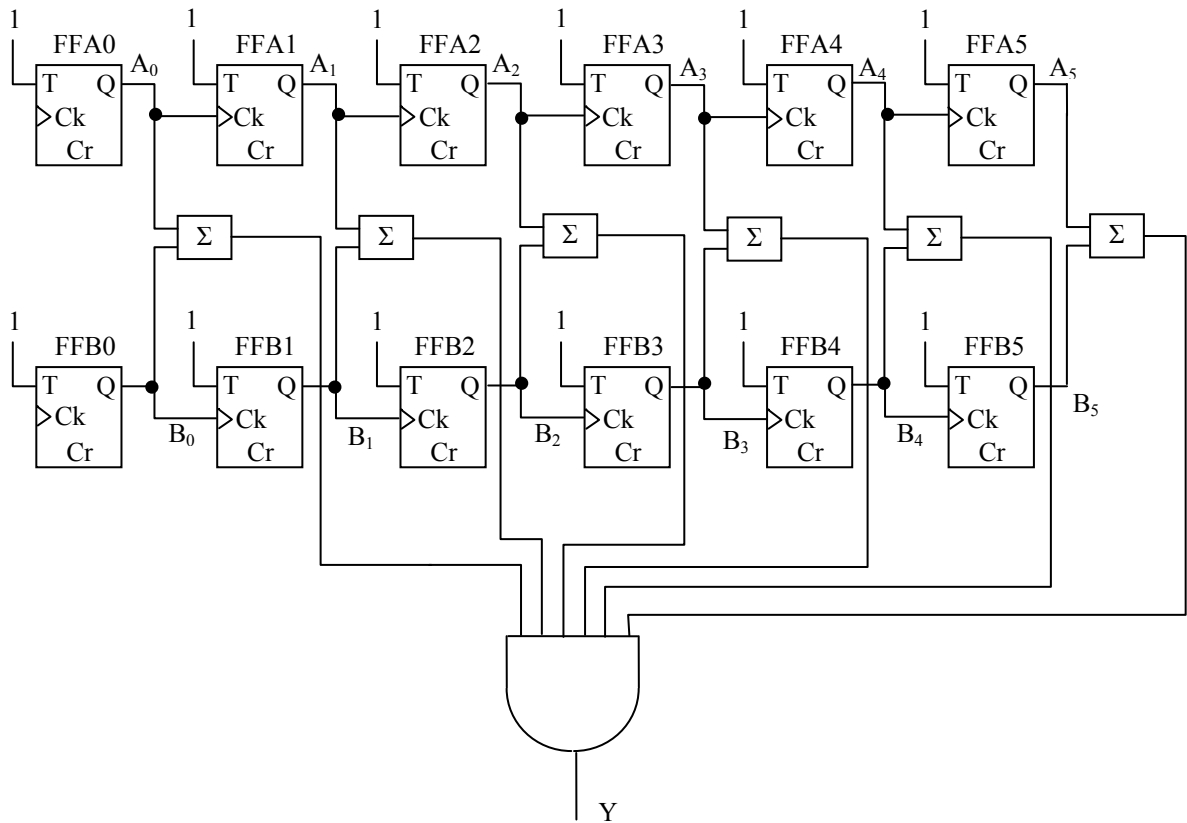
Τέλος, στο Σχ. 6.40 δίνουμε την υλοποίηση του κυκλώματος του ερωτήματος (5) που αποτελείται ουσιαστικά από την πύλη NAND στο κάτω δεξιό μέρος του σχήματος με εισόδους $Q_5 Q_4 Q_0 \overline{Ck}$ (θυμηθείτε ότι τα flip-flop ενεργοποιούνται με την αρνητική ακμή ρολογιού, άρα όταν ο παλμός ρολογιού έχει πέσει στο 0). Μόλις μετρηθεί ο 49ος παλμός και οι τέσσερις εισοδοί της πύλης NAND είναι 1, οπότε η έξοδος της που γίνεται 0 οδηγείται στην είσοδο Clear των flip-flops FF0, FF4, FF5, τα οποία και μηδενίζει ώστε να αρχίσει ξανά τη διαδικασία μέτρησης ο απαριθμητής.

Παράδειγμα 6.11

Δίνονται δύο εξαψήφιοι απαριθμητές A και B, με α και β το περιεχόμενό τους. Να σχεδιαστεί κύκλωμα το οποίο όταν $a = \beta$ να παράγει σήμα $Y = 1$. Ομοίως για $a = 2\beta + 3$. (Διατίθεται μόνο ένας παράλληλος αθροιστής).

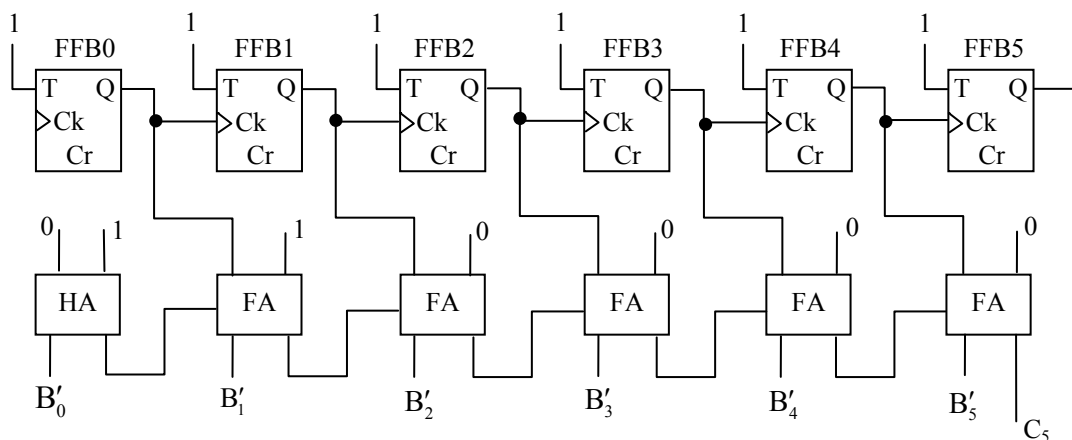
Λύση

Η πρώτη συνθήκη η οποία πρέπει να ικανοποιείται, είναι το περιεχόμενο του απαριθμητή β να μην είναι πολύ μεγάλο, δηλαδή να μην είναι το μέγιστο σημαντικό του ψηφίο 1 (MSB = 1), διότι δεν θα μπορούμε να παραστήσουμε τον αριθμό 2β με τον ίδιο απαριθμητή (γιατί;). Στο επόμενο σχήμα δίνουμε την κυκλωματική υλοποίηση της συνδεσμολογίας των δύο απαριθμητών προκειμένου να παράγεται το σήμα $Y = 1$ όταν $a = \beta$.



Σχ. 6.42. Κυκλωματική υλοποίηση συγκριτή δύο απαριθμητών.

Τα κυκλώματα που συμβολίζονται με το σύμβολο Σ είναι κυκλώματα συγκριτή ενός bit. Τέτοιο κύκλωμα είναι η πύλη XNOR η οποία σύμφωνα με τον πίνακα αληθείας του Σχ. 3.2 δίνει στην έξοδό της 1 όταν και οι δύο εισοδοί της είναι ίδιες (0 ή 1), ενώ δίνει στην έξοδό της 0 όταν οι εισοδοί της είναι διαφορετικές (δηλαδή η μία 1 και η άλλη 0). Έτσι, στο κύκλωμα του Σχ. 6.42, η έξοδος Y θα είναι ίση με 1 όταν και μόνο όταν και οι έξι εισοδοί της είναι 1. Αυτό θα συμβεί όταν οι έξοδοι και των έξι συγκριτών είναι 1, δηλαδή όταν το περιεχόμενο και των δύο απαριθμητών είναι ίδιο (η σύγκριση γίνεται ξεχωριστά σε κάθε bit). Για οποιοδήποτε άλλο συνδυασμό η έξοδος Y είναι 0.



Σχ. 6.43. Κυκλωματική υλοποίηση της πράξης $2\beta+3$.

Σε ό,τι αφορά την κυκλωματική υλοποίηση για την έξοδο σήματος $Y = 1$ όταν $a = 2\beta+3$, αρκεί κατ' αρχήν να ολισθήσουμε το περιεχόμενο του καταχωρητή B κατά μία θέση δεξιά (ώστε το δυαδικό του περιεχόμενο να διπλασιασθεί) και να προσθέσουμε το δυαδικό ισοδύναμο του δεκαδικού αριθμού 3, και στη συνέχεια να συγκρίνουμε κατά bit το νέο περιεχόμενο που θα προκύψει με το περιεχόμενο του απαριθμητή a. Η κυκλωματική υλοποίηση αυτών των πράξεων φαίνεται στο Σχ. 6.43.

Η ζητούμενη έξοδος Y θα προκύψει από μία πύλη AND της οποίας οι είσοδοι θα είναι τα σήματα: $\overline{B_5}$, $(A_0 \Sigma B'_0)$, $(A_1 \Sigma B'_1)$, $(A_2 \Sigma B'_2)$, $(A_3 \Sigma B'_3)$, $(A_4 \Sigma B'_4)$, $(A_5 \Sigma B'_5)$, $\overline{C_5}$. Με το σύμβολο $(A_i \Sigma B'_i)$ συμβολίζουμε το αποτέλεσμα της σύγκρισης του bit A_i του απαριθμητή A με το αντίστοιχο bit B'_i του Σχ. 6.43. Η πύλη AND πρέπει να έχει ως είσοδο το σήμα $\overline{B_5}$ για να ελέγχουμε εάν το αρχικό περιεχόμενο του απαριθμητή B είχε $MSB = 1$ όπως είπαμε στην αρχή. Επίσης, η πύλη AND πρέπει να έχει ως είσοδο το σήμα $\overline{C_5}$ ώστε να ελέγχουμε εάν τυχόν προκύψει κρατούμενο από την πράξη $2\beta+3$. Έτσι, η έξοδος Y θα δίνεται από το γινόμενο, με χρήση μιας πύλης AND

$$Y = \overline{B_5} \cdot (A_0 \Sigma B'_0) \cdot (A_1 \Sigma B'_1) \cdot (A_2 \Sigma B'_2) \cdot (A_3 \Sigma B'_3) \cdot (A_4 \Sigma B'_4) \cdot (A_5 \Sigma B'_5) \cdot \overline{C_5}$$

και θα έχουμε

$$Y = 1 \text{ όταν } a = 2\beta+3$$

$$Y = 0 \text{ όταν } a \neq 2\beta+3$$

Έτσι, η κυκλωματική υλοποίηση θα είναι ίδια με αυτήν του Σχ. 6.42 όπου όμως οι έξοδοι $B_0 - B_5$ θα αντικατασταθούν με τις εξόδους $B'_0 - B'_5$, και έχουν προστεθεί στην πύλη AND οι έξοδοι $\overline{B_5}$ και $\overline{C_5}$. Για λόγους σχεδιαστικής πολυπλοκότητας παραλείπουμε την πλήρη κυκλωματική υλοποίηση.

Παράδειγμα 6.12

Να σχεδιαστεί BCD απαριθμητής με δυνατότητα μέτρησης 99 παλμών. Ο απαριθμητής θα πρέπει:

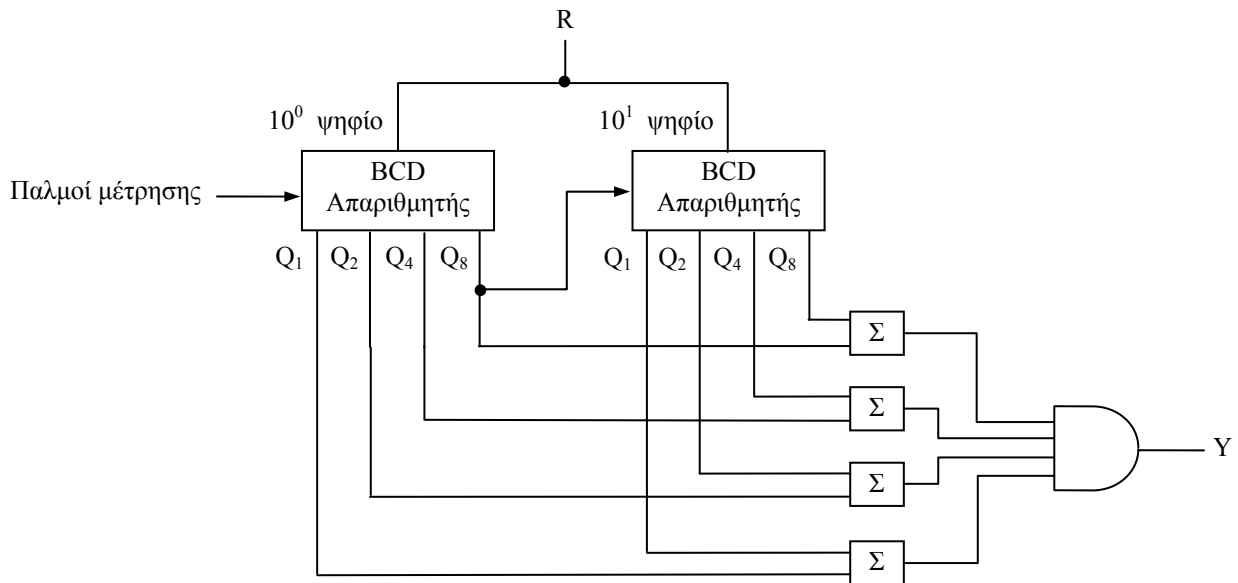
- 1) Όταν δέχεται σήμα εισόδου $R = 1$ να μηδενίζεται.
- 2) Να παράγει σήμα εξόδου $Y = 1$ όταν το αποτέλεσμα της μέτρησης είναι διαιρετό με το $(11)_{10}$.

Λύση

Η κυκλωματική υλοποίηση του απαριθμητή φαίνεται στο Σχ. 6.44.

Από τα δεδομένα του παραδείγματος είναι φανερό ότι θα πρέπει να χρησιμοποιήσουμε δύο δεκαδικούς BCD απαριθμητές, έναν για κάθε δεκάδα, δεδομένου ότι πρέπει να μπορούμε να μετρήσουμε 99 παλμούς. Η είσοδος R συνδέεται στο Reset των δεκαδικών απαριθμητών και όταν ενεργοποιείται (δηλ. $R = 1$) μηδενίζει τους απαριθμητές. Για να είναι διαιρετή η έξοδος του απαριθμητή με το 11 θα πρέπει τα δύο δεκαδικά ψηφία να είναι ίσα, δηλαδή να είναι 11, 22, 33, ... 99.

Έτσι, συνδέουμε ανά δύο τα ψηφία των εξόδων ίδιας σημαντικότητας των απαριθμητών BCD ως εισόδους σε συγκριτές ενός bit, και τα αποτελέσματα των συγκρίσεων διοχετεύονται σε μια πύλη AND. Η έξοδος της πύλης αυτής θα είναι 1 όταν και μόνο όταν τα όλα τα ψηφία ίδιας σημαντικότητας είναι ίσα μεταξύ τους, δηλαδή όταν οι δεκαδικοί αριθμοί στην έξοδο των απαριθμητών είναι ίσοι.

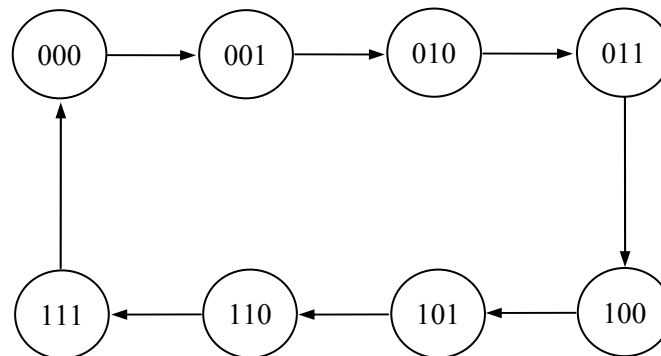


Σχ. 6.44. Κυκλωματική υλοποίηση του BCD απαριθμητή.

6.4.3. Σύγχρονοι απαριθμητές

Οι σύγχρονοι απαριθμητές διαφέρουν από τους ασύγχρονους κατά το ότι οι παλμοί του ρολογιού εφαρμόζονται στις εισόδους Ck όλων των flip-flops. Η απόφαση, του κατά πόσον ένα flip-flop είναι να αντιστραφεί ή όχι, στηρίζεται στις τιμές των J και K την ώρα του παλμού. Με $J = K = 0$, μένει αμετάβλητο. Με $J = K = 1$, αντιστρέφεται. Το επόμενο παράδειγμα θα βοηθήσει ιδιαίτερα στην κατανόηση της σχεδίασης ενός σύγχρονου δυαδικού απαριθμητή.

Έστω λοιπόν ένας δυαδικός απαριθμητής των n bits ο οποίος αποτελείται από n flip-flops και μπορεί να μετράει στο δυαδικό από το 0 ως το $2^n - 1$. Στο Σχ. 6.45 φαίνεται το διάγραμμα καταστάσεων ενός τέτοιου απαριθμητή 3 bits. Όπως βλέπουμε από τους κώδικες των καταστάσεων μέσα στους κύκλους, οι έξοδοι των flip-flops μετράνε στο δυαδικό, από το 000 ως το 111, και μετά ξανά από την αρχή. Δίνεται ο πίνακας διέγερσης του δυαδικού απαριθμητή. Τα τρία flip-flops τα ονομάζουμε A_2 , A_1 και A_0 . Ο καταλληλότερος τύπος flip-flop για τους δυαδικούς απαριθμητές είναι ο T (δηλ. ο J-K με τις εισόδους J και K βραχυκυκλωμένες). Τις διεγέρσεις των εισόδων T τις βρίσκουμε από τον πίνακα διέγερσης του flip-flop τύπου T και από την παρούσα και την επόμενη κατάσταση, δηλαδή από τις μετρήσεις σε μια γραμμή και στην επόμενη της.



Σχ. 6.45. Διάγραμμα καταστάσεων ενός σύγχρονου δυαδικού απαριθμητή 3 bits.

Για παράδειγμα, θεωρήστε τη γραμμή 001. Η παρούσα κατάσταση είναι η 001 και η επόμενη είναι η 010 (στην από κάτω γραμμή). Συγκρίνοντας αυτές τις δύο καταστάσεις, βλέπουμε ότι το A_2 πηγαίνει από κατάσταση 0 σε 0 - άρα $TA_2 = 0$, αφού το flip-flop αυτό πρέπει να μείνει αμετάβλητο. Το A_1 πάει από το 0 στο 1 - άρα $TA_1 = 1$, αφού αυτό το flip-flop πρέπει να αντιστραφεί με τον παλμό μέτρησης. Ομοίως, το A_0 αλλάζει από 1 σε 0, άρα $TA_0 = 1$, για να γίνει αυτή η αντιστροφή. Την τελευταία γραμμή (111) τη συγκρίνουμε με την πρώτη (000) που είναι η επόμενη της κατάσταση. Αφού και τα τρία flip-flops πρέπει να αντιστραφούν (από το 1 στο 0), έχουμε $TA_2 = TA_1 = TA_0 = 1$.

Πίνακας διέγερσης σύγχρονου δυαδικού απαριθμητή 3 bits

Παρούσα κατάσταση			Επόμενη κατάσταση			Είσοδος flip-flop		
A_2	A_1	A_0	A_2	A_1	A_0	TA_2	TA_1	TA_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Τις συναρτήσεις εισόδου των flip-flops, που τις παίρνουμε από τον πίνακα διέγερσης, τις απλοποιούμε με τους χάρτες Karnaugh του Σχ. 6.46. Κάτω από κάθε χάρτη έχουμε γράψει την αντίστοιχη απλοποιημένη συνάρτηση Boole. Αυτές οι συναρτήσεις περιγράφουν το συνδυαστικό κομμάτι του κυκλώματος και βάζοντας αυτό μαζί με τα τρία flip-flops, παίρνουμε το λογικό διάγραμμα του απαριθμητή που φαίνεται στο Σχ. 6.47.

$A_2 \backslash A_1A_0$	00	01	11	10
0			1	
1			1	

$TA_2 = A_1A_0$

$A_2 \backslash A_1A_0$	00	01	11	10
0		1	1	
1		1	1	

$TA_1 = A_0$

$A_2 \backslash A_1A_0$	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$TA_0 = 1$

Σχ. 6.46. Χάρτες για τον δυαδικό απαριθμητή 3 bits.

Κατά παρόμοιο τρόπο μπορούμε να βρούμε το λογικό διάγραμμα σύγχρονου δυαδικού απαριθμητή 5 bits, το οποίο δίνεται στο Σχ. 6.48. Από το διάγραμμα χρονισμού του Σχ. 6.35 προκύπτει ότι θα πρέπει να έχουμε:

Εναλλαγή του : Q_0 με κάθε παλμό	$\rightarrow T_0 = 1$
Q_1 μόνον όταν $Q_0 = 1$	$\rightarrow T_1 = Q_0$
Q_2 μόνον όταν $Q_0 = Q_1 = 1$	$\rightarrow T_2 = Q_0Q_1$
Q_3 μόνον όταν $Q_0 = Q_1 = Q_2 = 1$	$\rightarrow T_3 = Q_0Q_1Q_2$
Q_4 μόνον όταν $Q_0 = Q_1 = Q_2 = Q_3 = 1$	$\rightarrow T_4 = Q_0Q_1Q_2Q_3$

Άρα συμπεραίνουμε ότι:

$$T_0 = 1$$

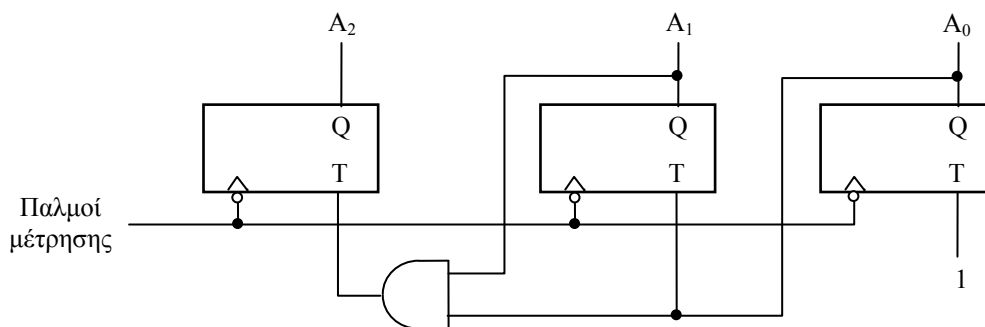
$$T_1 = Q_0$$

$$T_2 = Q_1T_1$$

$$T_3 = Q_2T_2$$

$$T_4 = Q_3T_3$$

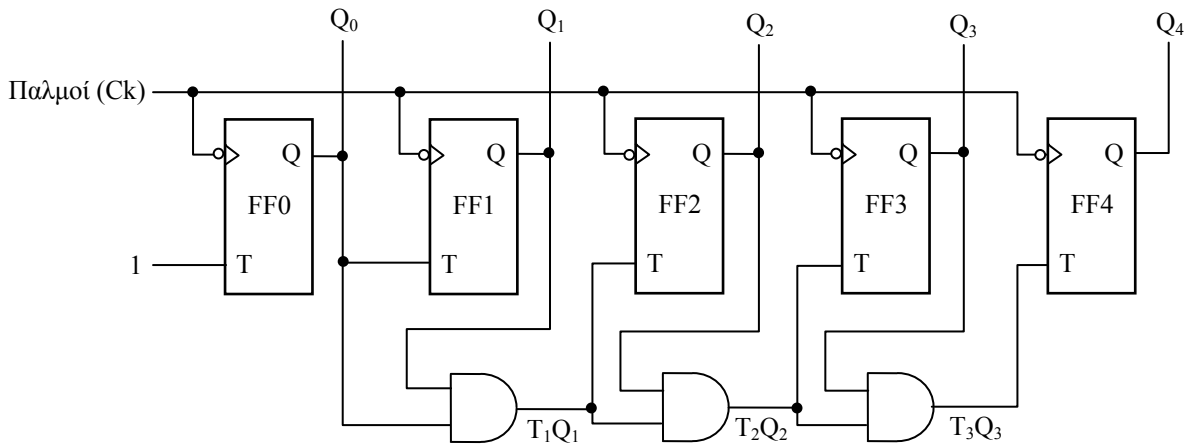
Αυτές ακριβώς τις σχέσεις υλοποιεί το κύκλωμα του Σχ. 6.48.



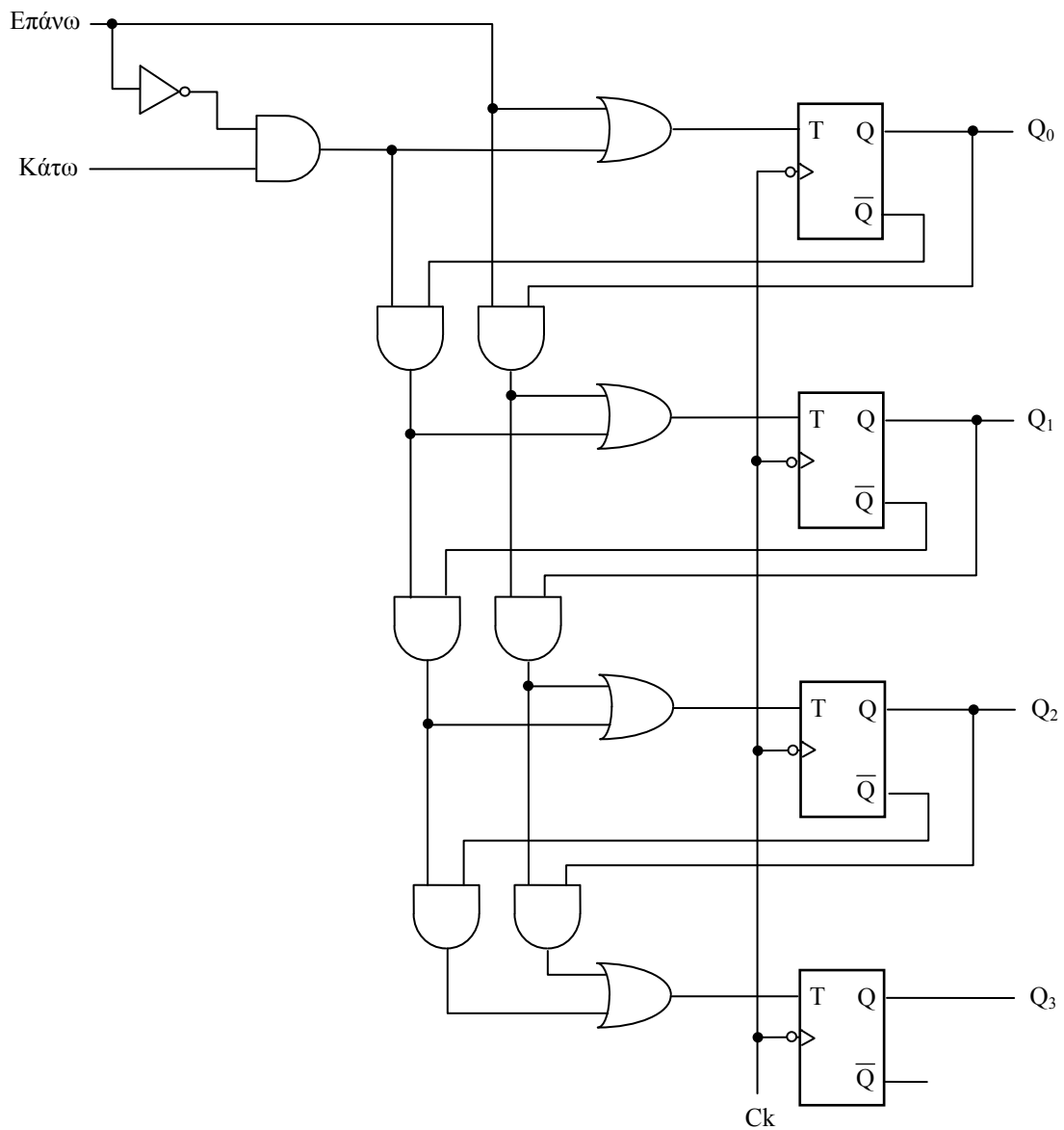
Σχ. 6.47. Το λογικό διάγραμμα του δυαδικού απαριθμητή 3 bits.

Η σχεδίαση σύγχρονων δυαδικών απαριθμητών είναι τόσο απλή, ώστε περιττεύει η χρήση της αυστηρής μεθόδου σχεδίασης ακολουθιακής λογικής. Σε έναν τέτοιο μετρητή, το λιγότερο σημαντικό flip-flop αντιστρέφεται με τον κάθε παλμό. Αυτό σημαίνει ότι οι εισοδοί J και K πρέπει να κρατούνται στο λογικό 1. Σε κάθε άλλη θέση σημαντικότητας, όταν έρχεται ο παλμός, το flip-flop πρέπει να αντιστραφεί εάν, και μόνο εάν, όλα τα bits των λιγότερο σημαντικών θέσεων είναι 1, αφού αν όλα αυτά είναι 1, τότε θα αλλάξουν όλα τους σε 0 με τον επόμενο παλμό μέτρησης. Η μέτρηση στο δυαδικό σύστημα βασικά συνίσταται στην αντιστροφή των bits, αρχίζοντας από δεξιά και προχωρώντας προς τα αριστερά, έως ότου βρεθεί ένα bit που η αντιστροφή του να το μετατρέψει από 0 σε 1.

Για παράδειγμα, αν η παρούσα κατάσταση του απαριθμητή 5 bits είναι $Q_4Q_3Q_2Q_1Q_0 = 00011$, τότε η επόμενη μέτρηση θα είναι 00100. Το Q_0 αντιστρέφεται πάντα. Το Q_1 αντιστρέφεται επειδή η παρούσα κατάστασή του Q_1 είναι 1 και η είσοδος του $T_1 = Q_0 = 1$. Το Q_2 αντιστρέφεται επειδή $Q_1Q_0 = 11$. Αλλά, το Q_3 δεν αντιστρέφεται διότι $Q_2Q_1Q_0 = 011$, που δεν είναι όλα τους 1. Οι σύγχρονοι δυαδικοί απαριθμητές έχουν ομοιόμορφη δομή και μπορούν να κατασκευαστούν εύκολα με πύλες AND και flip-flops τύπου T. Αυτή η ομοιόμορφη δομή φαίνεται καθαρά στον απαριθμητή του Σχ. 6.48.



Σχ. 6.48. Το λογικό διάγραμμα του δυαδικού απαριθμητή 5 bits.



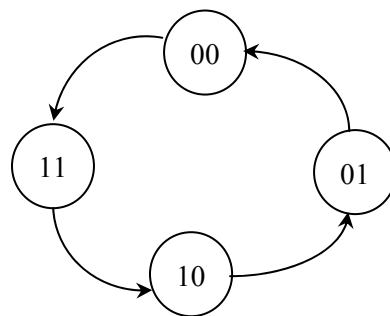
Σχ. 6.49. Το λογικό διάγραμμα του δυαδικού απαριθμητή 4 bits “επάνω/κάτω”.

Είναι δυνατό ο δυαδικός απαριθμητής να μετράει προς-τα-κάτω, δηλαδή να μειώνεται το δεκαδικό αντίστοιχο της εξόδου του με κάθε παλμό μέτρησης. Σε ένα σύγχρονο δυαδικό απαριθμητή προς-τα-κάτω, το λιγότερο σημαντικό flip-flop αντιστρέφεται με κάθε παλμό. Στις υπόλοιπες θέσεις, όταν έρχεται ο παλμός, τα flip-flops αντιστρέφονται αν τα λιγότερο σημαντικά bits είναι όλα τους 0. Για παράδειγμα, αν η παρούσα κατάσταση ενός απαριθμητή 4 bits προς-τα-κάτω είναι $Q_3Q_2Q_1Q_0 = 1100$, η επόμενη μέτρηση θα είναι 1011. Το Q_0 αντιστρέφεται πάντα. Το Q_1 αντιστρέφεται επειδή η παρούσα κατάσταση του Q_0 είναι το 0. Το Q_2 αντιστρέφεται επειδή $Q_1Q_0 = 00$. Αλλά, το Q_3 δεν αντιστρέφεται, αφού $Q_2Q_1Q_0 = 100$, δηλαδή, δεν είναι όλα τους μηδενικά.

Ένας δυαδικός απαριθμητής προς-τα-κάτω μπορεί να κατασκευαστεί όπως και ο απαριθμητής του Σχ. 6.48, με μόνη τη διαφορά ότι οι είσοδοι των πυλών AND θα πρέπει να έρχονται από τις συμπληρωματικές εξόδους \bar{Q} , αντί από τις κανονικές Q , των προηγούμενων flip-flops. Οι δύο τρόποι λειτουργίας μπορούν να ενσωματωθούν σε ένα κύκλωμα. Το Σχ. 6.49 δείχνει ένα δυαδικό απαριθμητή 4 bits που μπορεί να μετράει είτε προς τα πάνω είτε προς τα κάτω. Όταν η είσοδος ελέγχου “επάνω” είναι 1, το κύκλωμα μετράει προς τα πάνω, αφού οι είσοδοι T καθορίζονται από τις προηγούμενες τιμές των κανονικών εξόδων Q . Όταν η είσοδος ελέγχου “κάτω” είναι 1, το κύκλωμα μετράει προς τα κάτω, αφού τα T καθορίζονται από τις συμπληρωματικές εξόδους \bar{Q} . Όταν και το “επάνω” και το “κάτω” είναι 0, τότε ο απαριθμητής μένει αμετάβλητος στην ίδια μέτρηση. Όταν οι είσοδοι “επάνω” και “κάτω” είναι ταυτόχρονα 1, το κύκλωμα μετράει προς τα επάνω. Αυτό εξασφαλίζει ότι μία μόνο λειτουργία εκτελείται κάθε φορά.

Παράδειγμα 6.13

Να σχεδιαστεί απαριθμητής με μη δυαδική ακολουθία μέτρησης η οποία περιγράφεται στο ακόλουθο διάγραμμα καταστάσεων. Να χρησιμοποιηθούν flip-flop τύπου T.



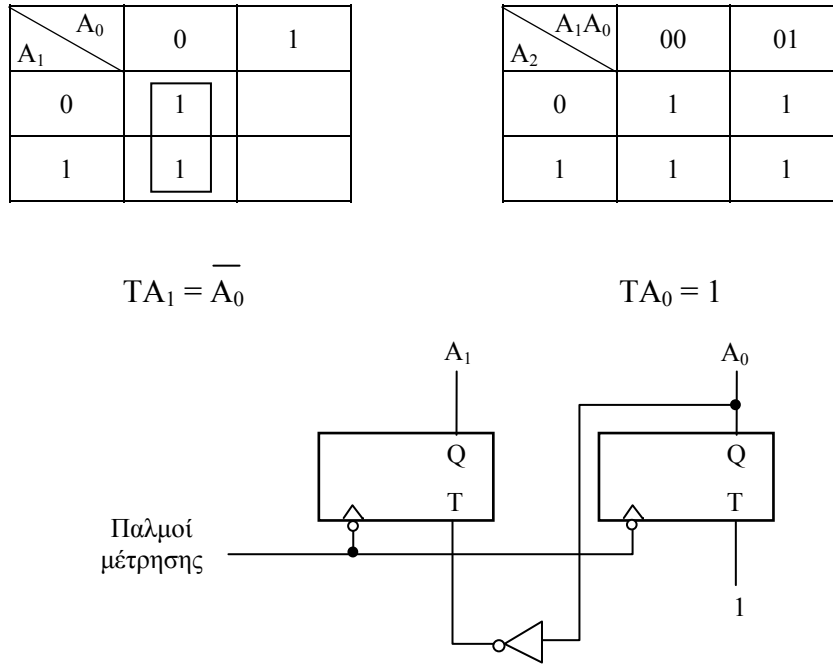
Λύση

Κατασκευάζουμε τον πίνακα διέγερσης του σύγχρονου απαριθμητή 2 bits τον οποίο θα χρησιμοποιήσουμε.

Πίνακας διέγερσης σύγχρονου δυαδικού απαριθμητή 2 bits

Παρούσα κατάσταση		Επόμενη κατάσταση		Είσοδος flip-flop	
A_1	A_0	A_1	A_0	TA_1	TA_0
0	0	1	1	1	1
1	1	1	0	0	1
1	0	0	1	1	1
0	1	0	0	0	1

Τις συναρτήσεις εισόδου των flip-flops, που τις παίρνουμε από τον πίνακα διέγερσης, τις απλοποιούμε με τους παρακάτω χάρτες Karnaugh. Κάτω από κάθε χάρτη έχουμε γράψει την αντίστοιχη απλοποιημένη συνάρτηση Boole. Αυτές οι συναρτήσεις περιγράφουν το συνδυαστικό κομμάτι του κυκλώματος και βάζοντας αυτό μαζί με τα δύο flip-flops, παίρνουμε το λογικό διάγραμμα του απαριθμητή που φαίνεται παρακάτω.



Σχ. 6.50. Το λογικό διάγραμμα του δυαδικού απαριθμητή 3 bits.

Παράδειγμα 6.14

Να σχεδιαστεί απαριθμητής με μη δυαδική ακολουθία μέτρησης η οποία περιγράφεται στον πιο κάτω πίνακα.

Παρούσα κατάσταση			Επόμενη κατάσταση		
A	B	C	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0

Λύση

Ένας απαριθμητής με n flip-flops μπορεί να έχει και ακολουθία μέτρησης με λιγότερες από 2^n καταστάσεις. Έτσι, ένας απαριθμητής BCD ακολουθεί τη δυαδική σειρά από 0000 ως 1001 και μετά επιστρέφει στην 0000 και ξαναρχίζει. Άλλοι απαριθμητές μπορεί να ακολουθήσουν κάποια αυθαίρετη σειρά η οποία πιθανόν να

μην είναι η απλή δυαδική σειρά. Σε τούτο το παράδειγμα θα μελετήσουμε το σχεδιασμό ενός τέτοιου απαριθμητή. Σε όλες τις περιπτώσεις η διαδικασία σχεδιασμού είναι η ίδια: Ο πίνακας καταστάσεων εξάγεται από την ακολουθία μέτρησης και ο απαριθμητής σχεδιάζεται με τη βοήθεια των πινάκων αληθείας των flip-flops. Στο δοσμένο παράδειγμα, ο απαριθμητής έχει μία επαναλαμβανόμενη ακολουθία έξι καταστάσεων. Σ' αυτή την ακολουθία, τα flip-flops B και C επαναλαμβάνουν τη μέτρηση 00, 01, 10, ενώ το flip-flop A εναλλάσσεται μεταξύ του 0 και του 1 κάθε τρεις μετρήσεις. Η σειρά μέτρησης των ABC δεν είναι η απλή δυαδική, και υπάρχουν δύο αχρησιμοποίητες καταστάσεις - οι 011 και 111. Επιλέγουμε flip-flops τύπου J-K και στη συνέχεια φτιάχνουμε τον πίνακα διέγερσης που φαίνεται στον επόμενο πίνακα.

Παρούσα κατάσταση			Επόμενη κατάσταση			Είσοδοι flip-flop					
A	B	C	A	B	C	JA	KA	JB	KB	JC	KC
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

Ο πίνακας διέγερσης ενός flip-flop βρίσκεται από τον πίνακα αληθείας. Ως γνωστόν, ο πίνακας αληθείας καθορίζει την επόμενη κατάσταση, όταν ξέρουμε την παρούσα και τις εισόδους. Κατά το σχεδιασμό κυκλωμάτων, όμως, συνήθως ξέρουμε τη μετάβαση από την παρούσα στην επόμενη κατάσταση και θέλουμε να βρούμε τις συνθήκες εισόδου του flip-flop που θα την πραγματοποιήσουν. Για το σκοπό αυτό χρειαζόμαστε έναν πίνακα που να δίνει τις απαιτούμενες εισόδους για ορισμένη αλλαγή της κατάστασης. Ο πίνακας αυτός λέγεται “πίνακας διέγερσης” (excitation table). Στον πίνακα διέγερσης, το σύμβολο X παριστάνει έναν αδιάφορο όρο, δηλαδή δεν πειράζει αν η αντίστοιχη είσοδος είναι 1 ή 0. Ο πίνακας διέγερσης του flip-flop J-K είναι ως ακολούθως:

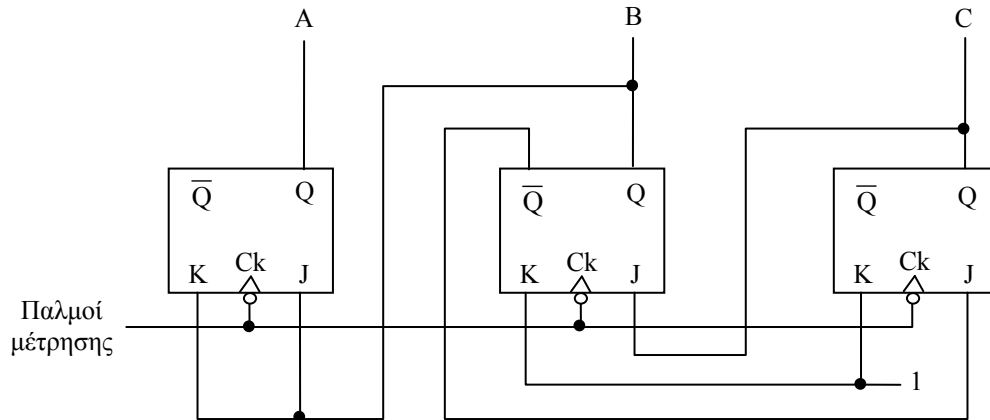
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Στο παράδειγμά μας, οι είσοδοι KB και KC έχουν μόνο 1 και X στις στήλες τους, κι έτσι αυτές θα τις κρατούμε συνεχώς στο 1. Τις υπόλοιπες συναρτήσεις εισόδου των flip-flops μπορούμε να τις απλοποιήσουμε, μη ξεχνώντας και ότι οι ελαχιστόροι 3 και 7 είναι αδιάφοροι όροι. Οι απλοποιημένες συναρτήσεις είναι:

$$\begin{aligned}
 JA &= B & KA &= B \\
 JB &= C & KB &= 1 \\
 JC &= \bar{B} & KC &= 1
 \end{aligned}$$

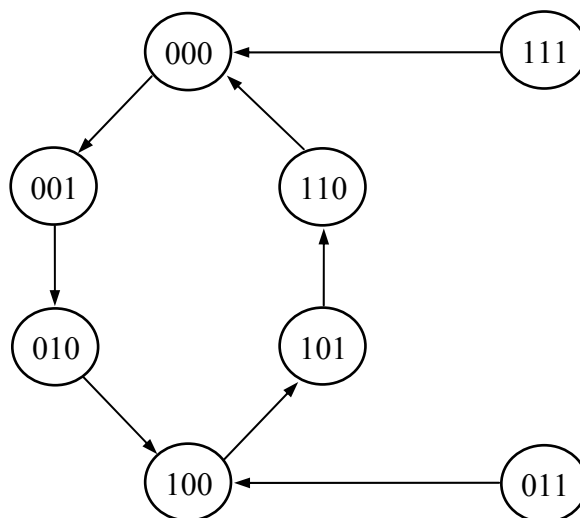
Κατά συνέπεια, το λογικό διάγραμμα του απαριθμητή είναι όπως φαίνεται στο Σχ. 6.51. Αφού υπάρχουν δύο αχρησιμοποίητες καταστάσεις, αναλύουμε το κύκλωμα για

να βρούμε τις συνέπειές τους. Αν το κύκλωμα βρεθεί από ένα λάθος σήμα στην κατάσταση 011, τότε πηγαίνει στην κατάσταση 100 μετά την εφαρμογή ενός παλμού ρολογιού. Αυτό προκύπτει από την παρατήρηση ότι, ενώ το κύκλωμα είναι στην παρούσα κατάσταση 011, οι έξοδοι των flip-flops είναι $A = 0$, $B = 1$ και $C = 1$. Από τις συναρτήσεις εισόδου των flip-flops έχουμε: $J_A = K_A = 1$, $J_B = K_B = 1$, $J_C = 0$ και $K_C = 1$. Έτσι, το flip-flop A συμπληρώνεται και γίνεται 1. Ομοίως, το B συμπληρώνεται και γίνεται 0. Το flip-flop C επαναφέρεται στο 0, διότι $K_C = 1$. Αυτά έχουν ως αποτέλεσμα η επόμενη κατάσταση να είναι 100. Με όμοιο τρόπο, βρίσκουμε ότι η επόμενη κατάσταση της 111 είναι η 000.



Σχ. 6.51. Κυκλωματικό διάγραμμα απαριθμητή με μη δυαδική ακολουθία.

Το Σχ. 6.52 δείχνει το πλήρες διάγραμμα καταστάσεων. Αν το κύκλωμα βρεθεί ποτέ σε μία από τις αχρησιμοποίητες καταστάσεις, ο αμέσως επόμενος παλμός μέτρησης θα το φέρει σε μια από τις έγκυρες καταστάσεις του, από όπου και θα συνεχίσει κανονικά να μετράει. Άρα, αυτός ο απαριθμητής έχει αυτόματη διόρθωση, δηλαδή, από οποιαδήποτε κατάσταση κι αν ξεκινήσει, έστω και αν αυτή είναι αχρησιμοποίητη, θα φτάσει τελικά στη σωστή ακολουθία μετρήσεων.



Σχ. 6.52. Διάγραμμα καταστάσεων απαριθμητή με μη δυαδική ακολουθία.

7. ΜΕΘΟΔΟΙ ΣΧΕΔΙΑΣΗΣ ΣΥΓΧΡΟΝΩΝ ΑΚΟΛΟΥΘΙΑΚΩΝ ΚΥΚΛΩΜΑΤΩΝ

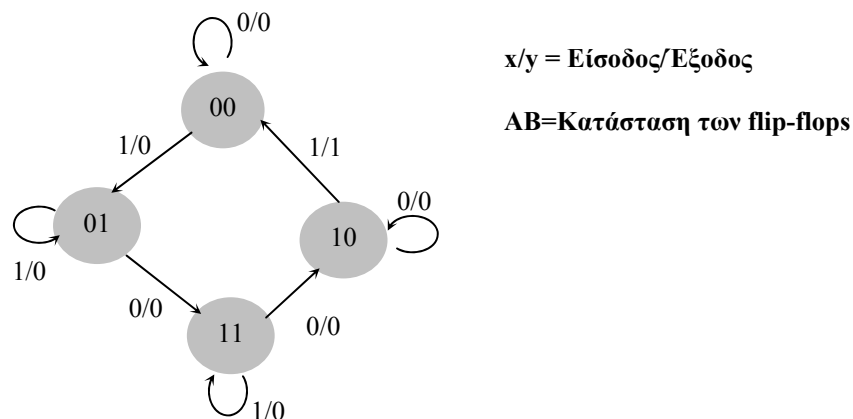
7.1. Εισαγωγή

Η σχεδίαση ενός ακολουθιακού κυκλώματος με ρολόι αρχίζει με ένα σύνολο προδιαγραφών και τελειώνει με ένα λογικό διάγραμμα ή με μερικές συναρτήσεις Boole από τις οποίες μπορεί να προκύψει το λογικό διάγραμμα. Ενώ ένα συνδυαστικό κύκλωμα περιγράφεται πλήρως με τον πίνακα αληθείας του, ένα ακολουθιακό κύκλωμα χρειάζεται τον πίνακα καταστάσεων του για να περιγραφεί ολοκληρωτικά. Το πρώτο βήμα στη σχεδίαση ενός ακολουθιακού κυκλώματος είναι να βρούμε τον πίνακα καταστάσεων του ή κάποια άλλη ισοδύναμη αναπαράσταση, όπως το διάγραμμα καταστάσεων ή οι εξισώσεις καταστάσεων.

7.2. Διάγραμμα καταστάσεων (state diagram)

Στα διαγράμματα καταστάσεων τις διαφορετικές καταστάσεις των flip-flops τις παριστάνουμε με κύκλους. Στην προκειμένη περίπτωση του Σχ. 7.1, έχουμε δύο flip-flops με εξόδους Q. Για το πρώτο flip-flop $Q=A$ και για το δεύτερο flip-flop $Q=B$. Τις μεταβάσεις από μία κατάσταση σε άλλη τις παριστάνουμε με βέλη που συνδέουν τους κύκλους. Μέσα σε κάθε κύκλο γράφουμε το δυαδικό αριθμό της κατάστασης την οποία παριστάνει αυτός (AB). Τα βέλη δείχνουν την επόμενη κατάσταση και έχουν πάνω τους δύο αριθμούς (x/y) που χωρίζονται μεταξύ τους με μία κάθετη. Ο πρώτος αριθμός (x) είναι η τιμή των εισόδων που προκαλεί αυτή τη μετάβαση καταστάσεων. Ο δεύτερος αριθμός (y) δίνει την τιμή των εξόδων κατά τη διάρκεια της παρούσας κατάστασης και για την τιμή των εισόδων που αντιστοιχεί στο ίδιο βέλος.

Για παράδειγμα, έστω το διάγραμμα καταστάσεων του σχήματος 7.1. Όταν το ακολουθιακό κύκλωμα βρίσκεται στην κατάσταση 01 [A(t)B(t)] και όταν η είσοδός του είναι 0 (x), τότε η επόμενη κατάσταση στην οποία θα μεταβεί το κύκλωμα με τον επόμενο παλμό ρολογιού, θα είναι η κατάσταση 11 [A(t+1)B(t+1)] και η έξοδός του θα είναι 0 (y). Όταν η είσοδός του είναι 1 (x) τότε η επόμενη κατάσταση [A(t+1)B(t+1)] στην οποία θα μεταβεί το κύκλωμα με τον επόμενο παλμό ρολογιού, θα είναι η κατάσταση 01 και η έξοδός του θα είναι 0 (y).



Σχ. 7.1. Διάγραμμα καταστάσεων σύγχρονου ακολουθιακού κυκλώματος προς σχεδίαση.

7.2.1. Πίνακας καταστάσεων (state table)

Τις ίδιες ακριβώς πληροφορίες μας δίνει και ο πίνακας καταστάσεων (state table). Ένας πίνακας καταστάσεων αποτελείται από 3 μέρη: Παρούσα κατάσταση, Επόμενη κατάσταση και Έξοδος. Η Παρούσα κατάσταση είναι η κατάσταση των flip-flops πριν από έναν ορισμένο παλμό ρολογιού. Η επόμενη κατάσταση είναι οι καταστάσεις των flip-flops που θα προκύψουν αμέσως μετά την πυροδότηση. Στο μέρος Έξοδος, δίνονται οι τιμές των μεταβλητών εξόδου στη διάρκεια της παρούσας κατάστασης. Τόσο η Επόμενη κατάσταση όσο και η Έξοδος έχουν δύο στήλες, μία για $x=0$ και μία για $x=1$, αφού και οι δύο τους είναι συναρτήσεις και της τιμής της εισόδου κατά τη διάρκεια της παρούσας κατάστασης.

Ο πίνακας καταστάσεων οποιουδήποτε ακολουθιακού κυκλώματος με m flip-flops και n εισόδους θα έχει 2^m γραμμές – μία για κάθε κατάσταση. Τα μέρη της επόμενης κατάστασης και των εξόδων θα έχουν 2^n στήλες το καθένα – μία για κάθε συνδυασμό εισόδων.

Παρούσα Κατάσταση	Επόμενη Κατάσταση A(t+1)B(t+1)		Έξοδος Y	
	x=0	x=1	x=0	x=1
A(t)B(t)	A(t+1)B(t+1)	A(t+1)B(t+1)	y	Y
0 0	0 0	0 1	0	0
0 1	1 1	0 1	0	0
1 0	1 0	0 0	0	1
1 1	1 0	1 1	0	0

Σχ. 7.2. Πίνακας καταστάσεων του σύγχρονου ακολουθιακού κυκλώματος προς σχεδίαση.

Παρατηρούμε ότι όταν $x=0$ από την κατάσταση $AB=01$ μεταβαίνουμε στην κατάσταση $AB=11$ και τότε η έξοδος είναι 0. Αυτό ακριβώς φαίνεται και από το διάγραμμα καταστάσεων.

7.2.2. Εξισώσεις καταστάσεων (state equations)

Η εξίσωση καταστάσεων είναι μία αλγεβρική έκφραση που καθορίζει τις συνθήκες μεταβολής κατάστασης ενός flip-flop. Το αριστερό μέρος της εξίσωσης είναι η επόμενη κατάσταση του flip-flop και το δεξιό μέρος δίνει αυτή την κατάσταση σαν συνάρτηση της παρούσας κατάστασης και των εισόδων. Οι εξισώσεις καταστάσεων προκύπτουν από τον πίνακα καταστάσεων. Για παράδειγμα για την επόμενη κατάσταση του A, παρατηρούμε ότι το A γίνεται 1 στις ακόλουθες 4 περιπτώσεις. Αν $x=0$ και η παρούσα κατάσταση των flip-flops είναι 01 ή 10 ή 11 και αν $x=1$ και η παρούσα κατάσταση των flip-flops είναι 11. Επομένως

$$A(t+1) = \bar{A} \cdot B \cdot \bar{x} + A \cdot \bar{B} \cdot \bar{x} + A \cdot B \cdot \bar{x} + A \cdot B \cdot x$$

Χρησιμοποιώντας το χάρτη Karnaugh

Bx	00	01	11	10
A				
0				1
1	1		1	1

Σχ. 7.3. Χάρτης Karnaugh για απλοποίηση της συνάρτησης A(t+1) που φαίνεται από την τρίτη και πέμπτη στήλη του πίνακα καταστάσεων του σχήματος 7.2.

$$A(t+1) = B \cdot \bar{x} + A \cdot B + A \cdot \bar{x} = B \cdot \bar{x} + A \cdot (B + \bar{x}) = B \cdot \bar{x} + A \cdot (\overline{\bar{B} \cdot x})$$

Ορίζουμε

$$S = B \cdot \bar{x}$$

και

$$R = \bar{B} \cdot x,$$

οπότε έχουμε:

$$A(t+1) = S + \bar{R} \cdot A$$

η οποία είναι η χαρακτηριστική εξίσωση του flip-flop A τύπου S-R.

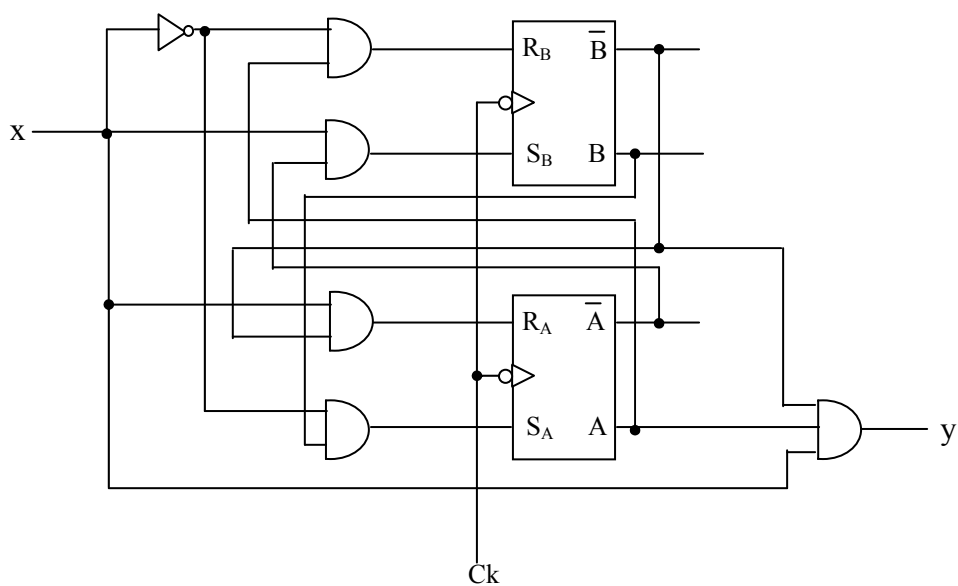
Ομοίως για την επόμενη κατάσταση B του δεύτερου flip-flop, μετά την εξαγωγή της εξίσωσης και την απλοποίησή της με τη βοήθεια του χάρτη Karnaugh, θα ισχύει

$$B(t+1) = \bar{A} \cdot x + B \cdot (\overline{A \cdot \bar{x}})$$

και

$$y(t+1) = A \cdot \bar{B} \cdot x$$

Το λογικό διάγραμμα θα είναι όπως στο σχήμα



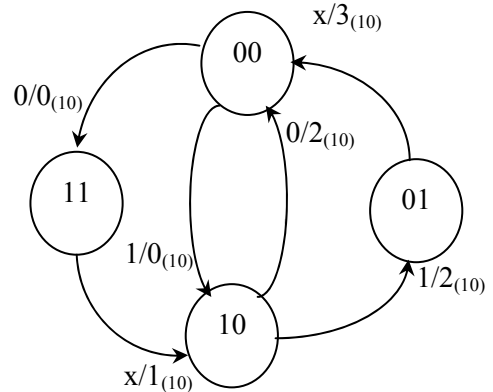
Σχ. 7.4. Σύγχρονο ακολουθιακό κύκλωμα.

7.3. Παραδείγματα Σχεδίασης Σύγχρονων Ακολουθιακών Κυκλωμάτων

Παράδειγμα 7.1

Έστω ότι θέλουμε να σχεδιάσουμε ένα ακολουθιακό κύκλωμα το οποίο:

- A) Από την κατάσταση 00 να μεταβαίνει
 - α) Στην κατάσταση 10 όταν η είσοδος είναι 1 και τότε η έξοδος του να είναι ο δεκαδικός αριθμός 0.
 - β) Στην κατάσταση 11 όταν η είσοδος είναι 0 και τότε η έξοδος να είναι ο δεκαδικός αριθμός 0.
- B) Από την κατάσταση 11 να μεταβαίνει στην κατάσταση 10 ανεξάρτητα της εισόδου και τότε η έξοδος του να είναι ο δεκαδικός αριθμός 1.
- Γ) Από την κατάσταση 10 να μεταβαίνει
 - α) Στην κατάσταση 00 όταν η είσοδος είναι 0 και τότε η έξοδος του να είναι ο δεκαδικός αριθμός 2.
 - β) Στην κατάσταση 01 όταν η είσοδος είναι 1 και τότε η έξοδος να είναι ο δεκαδικός αριθμός 2.
- Δ) Από την κατάσταση 01 να μεταβαίνει στην κατάσταση 00 ανεξάρτητα της εισόδου και τότε η έξοδος του να είναι ο δεκαδικός αριθμός 3.



Σχ. 7.5. Διάγραμμα καταστάσεων.

Το διάγραμμα καταστάσεων του ακολουθιακού κυκλώματος παρουσιάζεται στο Σχ. 7.5.

Πρώτος τρόπος επίλυσης

Σύμφωνα με τα προηγούμενα, ο πίνακας καταστάσεων του ακολουθιακού κυκλώματος είναι αυτός που παρουσιάζεται στο Σχ. 7.6.

Παρούσα Κατάσταση	Επόμενη Κατάσταση $A(t+1)B(t+1)$		Έξοδος Y	
	$x=0$	$x=1$	$x=0$	$x=1$
$A(t)B(t)$	$A(t+1)B(t+1)$	$A(t+1)B(t+1)$	y_1y_0	y_1y_0
0 0	1 1	1 0	0 0	0 0
0 1	0 0	0 0	1 1	1 1
1 0	0 0	0 1	1 0	1 0
1 1	1 0	1 0	0 1	0 1

Σχ. 7.6. Πίνακας Καταστάσεων.

Από τον πίνακα καταστάσεων προκύπτουν οι ακόλουθες εξισώσεις καταστάσεων. Όπου A εννοείται $A(t)$ και για λόγους απλότητας παραλείπεται η χρονική στιγμή. Ομοίως για το B.

$$\begin{aligned} A(t+1) &= \bar{A}\bar{B}\bar{x} + A\bar{B}\bar{x} + \bar{A}Bx + ABx \\ B(t+1) &= \bar{A}\bar{B}\bar{x} + A\bar{B}\bar{x} \\ y_1 &= \bar{A}B\bar{x} + A\bar{B}\bar{x} + \bar{A}Bx + A\bar{B}x = \bar{A}B + A\bar{B} = A \oplus B \\ y_0 &= \bar{A}B\bar{x} + A\bar{B}\bar{x} + \bar{A}Bx + ABx = \bar{A}B + AB = B \end{aligned} \quad \text{Εξισώσεις (1)}$$

Έστω ότι θέλουμε να υλοποιήσουμε το κύκλωμα με J-K flip-flops των οποίων η εξίσωση είναι της μορφής

$$Q(t+1) = J\bar{Q}(t) + \bar{K}Q(t) \quad (2)$$

Αυτό σημαίνει ότι πρέπει να φέρουμε τις εξισώσεις (1) για τα A(t+1) και B(t+1) στη μορφή (2).

$$\begin{aligned} A(t+1) &= (\bar{B}x + \bar{B}\bar{x})\bar{A} + (B\bar{x} + Bx)A = \bar{B}\bar{A} + BA \\ B(t+1) &= \bar{B}(\bar{x}\bar{A} + xA) = (\bar{x}\bar{A} + xA)\bar{B} + 0B = (\bar{x} \oplus A)\bar{B} + 0B \end{aligned} \quad \text{Εξισώσεις (3)}$$

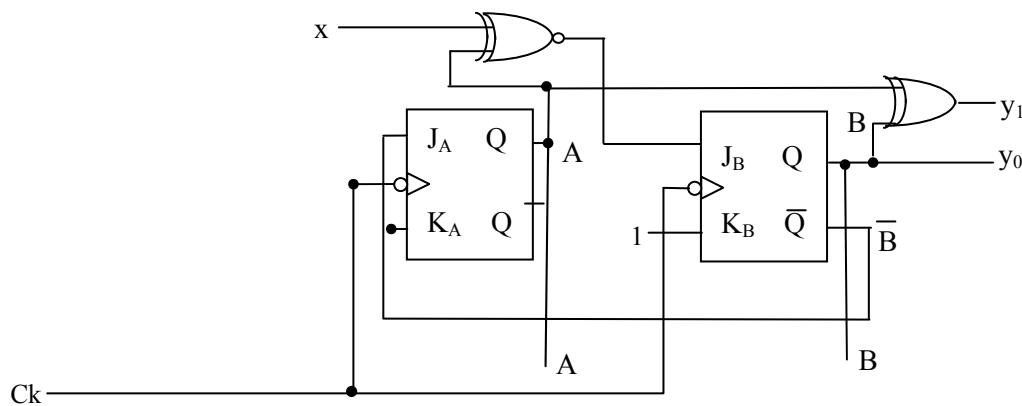
Από τις εξισώσεις (3) με τη βοήθεια της (2) οδηγούμαστε στο συμπέρασμα ότι για το πρώτο flip-flop (A) ισχύει:

$$\begin{aligned} J_A &= \bar{B} \\ K_A &= \bar{B} \end{aligned}$$

ενώ για το δεύτερο flip-flop (B) ισχύει:

$$\begin{aligned} J_B &= \bar{x}\bar{A} + xA = \overline{(A \oplus x)} \\ K_B &= 1 \end{aligned}$$

Μπορούμε να σχεδιάσουμε το κύκλωμά μας όπως φαίνεται στο Σχ. 7.7.



Σχ. 7.7. Το σύγχρονο ακολουθιακό κύκλωμα.

Δεύτερος τρόπος επίλυσης

Από τον πίνακα του Σχ. 7.6, για το J-K flip-flop μπορούμε να σχηματίσουμε τον πίνακα του Σχ. 7.9. Αυτός προκύπτει εύκολα αν σκεφτούμε ότι όταν το flip-flop μεταβαίνει από την κατάσταση Q(t)=0 στην κατάσταση Q(t+1)=0 τότε το J=0 και το K είναι αδιάφορο. Επίσης, για παράδειγμα, όταν το flip-flop μεταβαίνει από την κατάσταση Q(t)=1 στην κατάσταση Q(t+1)=0 τότε το K=1 ενώ το J είναι αδιάφορο.

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

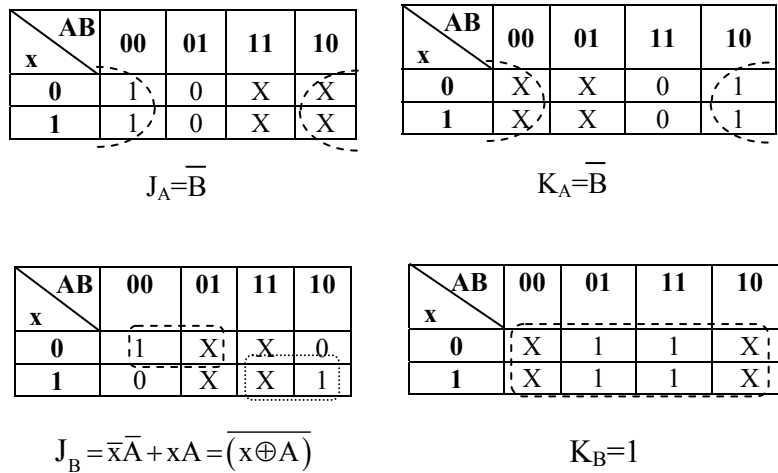
Σχ. 7.8. Πίνακας διέγερσης J-K flip-flop.

Ο πίνακας καταστάσεων του Σχ. 7.6 όσον αφορά στις επόμενες καταστάσεις των flip-flops, τροποποιείται όπως παρουσιάζεται στο Σχ. 7.9.

Παρούσα Κατάσταση	Επόμενη Κατάσταση $A(t+1)B(t+1)$		Είσοδοι των flip-flops			
			Είσοδοι του A		Είσοδοι του B	
	$x=0$	$x=1$	$x=0$	$x=1$	$x=0$	$x=1$
$A(t)B(t)$	$A(t+1)B(t+1)$	$A(t+1)B(t+1)$	$J_A K_A$	$J_A K_A$	$J_B K_B$	$J_B K_B$
0 0	1 1	1 0	1 X	1 X	1 X	0 X
0 1	0 0	0 0	0 X	0 X	X 1	X 1
1 0	0 0	0 1	X 1	X 1	0 X	1 X
1 1	1 0	1 0	X 0	X 0	X 1	X 1

Σχ. 7.9. Πίνακας καταστάσεων με τις εισόδους των flip-flops.

Από τον πίνακα του Σχ. 7.9, μπορούμε να απλοποιήσουμε τις συναρτήσεις για τα J_A, K_A, J_B, K_B με τη βοήθεια του χάρτη Karnaugh, όπως φαίνεται στο Σχ. 7.10.

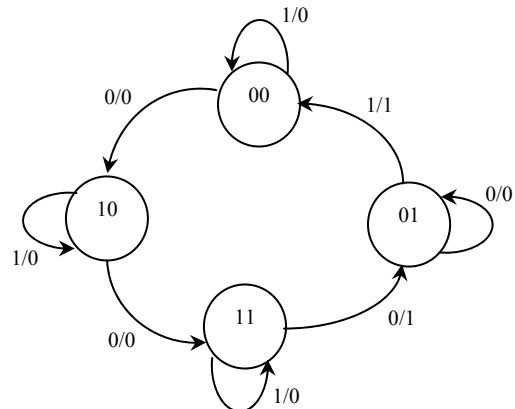


Σχ. 7.10. Χάρτες Karnaugh για απλοποίηση των εισόδων των flip-flops

Όπως παρατηρούμε, οδηγηθήκαμε στο ίδιο αποτέλεσμα με το προηγούμενο. Επομένως, η σχεδίαση του σύγχρονου ακολουθιακού κυκλώματος είναι αυτή του Σχ. 7.7.

Παράδειγμα 7.2

Να σχεδιαστεί ένα σύγχρονο ακολουθιακό κύκλωμα του οποίου το διάγραμμα κατάστασης φαίνεται παραπλεύρως. Το κύκλωμα έχει μία είσοδο και μία έξοδο. Η σχεδίαση να γίνει με χρήση S-R flip-flops.



Λύση

Από το διάγραμμα καταστάσεων συνάγεται ο ακόλουθος πίνακας καταστάσεων του ακολουθιακού κυκλώματος.

Παρούσα Κατάσταση	Επόμενη Κατάσταση A(t+1)B(t+1)		Έξοδος Y	
	x=0	x=1	x=0	x=1
A(t)B(t)	A(t+1)B(t+1)	A(t+1)B(t+1)	y₀	y₁
0 0	1 0	0 0	0	0
0 1	0 1	0 0	0	1
1 0	1 1	1 0	0	0
1 1	0 1	1 1	1	0

Από τον πίνακα καταστάσεων και τον πίνακα διέγερσης του S-R flip-flop μπορούμε να σχηματίσουμε τον παρακάτω πίνακα.

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

Σχ. 7.11. Πίνακας διέγερσης S-R flip-flop.

Παρούσα Κατάσταση	Επόμενη Κατάσταση A(t+1)B(t+1)		Είσοδοι των flip-flops			
			Είσοδοι του A		Είσοδοι του B	
	x=0	x=1	x=0	x=1	x=0	x=1
A(t)B(t)	A(t+1)B(t+1)	A(t+1)B(t+1)	S_AR_A	S_AR_A	S_BR_B	S_BR_B
0 0	1 0	0 0	1 0	0 X	0 X	0 X
0 1	0 1	0 0	0 X	0 X	X 0	0 1
1 0	1 1	1 0	X 0	X 0	1 0	0 X
1 1	0 1	1 1	0 1	X 0	X 0	X 0

Σχ. 7.12. Πίνακας καταστάσεων με τις εισόδους των flip-flops.

Από τον παραπάνω πίνακα, μπορούμε να απλοποιήσουμε τις συναρτήσεις για τα S_A, R_A, S_B, R_B με τη βοήθεια του χάρτη Karnaugh, όπως φαίνεται στο παρακάτω σχήμα.

x \ AB	00	01	11	10
0	1			X
1			X	X

$$S_A = \bar{x} \bar{B}$$

x \ AB	00	01	11	10
0		X	1	
1	X	X		

$$R_A = \bar{x}B$$

	AB	00	01	11	10
x					
0			X	X	X
1				X	

$$S_B = \bar{x}A$$

	AB	00	01	11	10
x					
0		X			
1		X	1		X

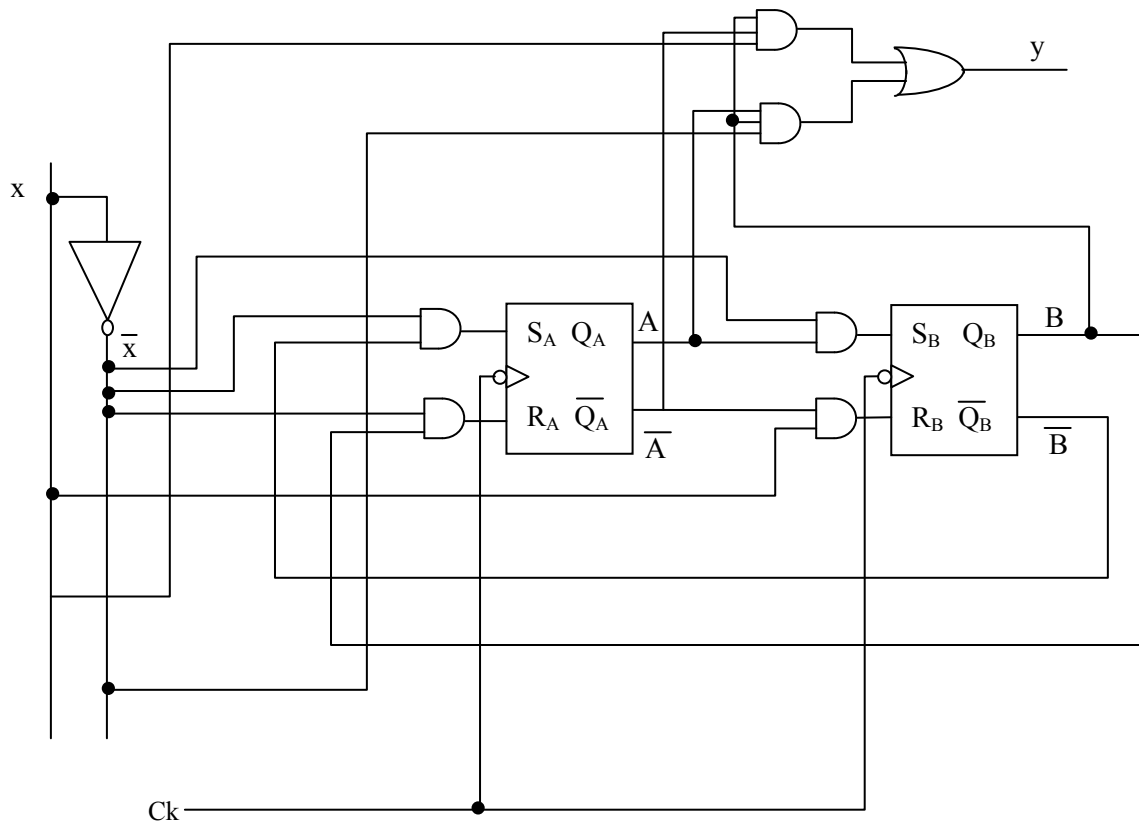
$$R_B = x\bar{A}$$

	AB	00	01	11	10
x					
0				1	
1			1		

$$y = \bar{x}AB + x\bar{A}B$$

Σχ. 7.13. Χάρτες Karnaugh για απλοποίηση των εισόδων των flip-flops.

Το τελικό κύκλωμα θα είναι:



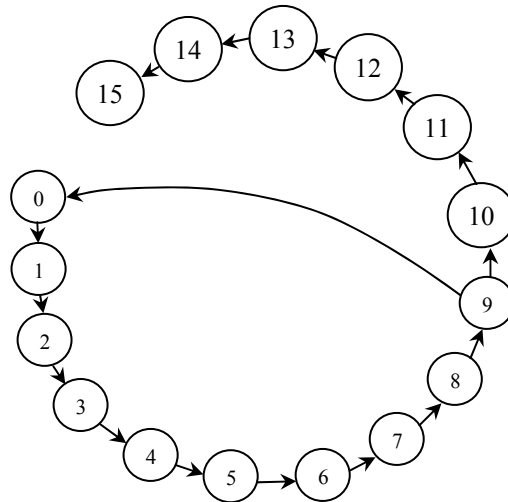
Σχ. 7.14. Το σύγχρονο ακολουθιακό κύκλωμα.

Παράδειγμα 7.3

Να σχεδιαστεί ένα σύγχρονο ακολουθιακό κύκλωμα το οποίο να διατρέχει διαδοχικά τις τιμές 0 μέχρι και 9. Η σχεδίαση να γίνει με χρήση J-K flip-flops.

Λύση

Πρόκειται για έναν δεκαδικό απαριθμητή. Για την δυαδική αναπαράσταση των 10 τιμών απαιτούνται 4 δυαδικά ψηφία, δηλαδή 4 flip-flops. Σχεδιάζουμε αρχικά το διάγραμμα καταστάσεων.



Σχ. 7.15. Διάγραμμα καταστάσεων.

Παρούσα Κατάσταση	Επόμενη Κατάσταση	Είσοδοι			
		$Q_3Q_2Q_1Q_0$	$Q_3Q_2Q_1Q_0$	J_3K_3	J_2K_2
0 0 0 0	0 0 0 1	0 X	0 X	0 X	1 X
0 0 0 1	0 0 1 0	0 X	0 X	1 X	X 1
0 0 1 0	0 0 1 1	0 X	0 X	X 0	1 X
0 0 1 1	0 1 0 0	0 X	1 X	X 1	X 1
0 1 0 0	0 1 0 1	0 X	X 0	0 X	1 X
0 1 0 1	0 1 1 0	0 X	X 0	1 X	X 1
0 1 1 0	0 1 1 1	0 X	X 0	X 0	1 X
0 1 1 1	1 0 0 0	1 X	X 1	X 1	X 1
1 0 0 0	1 0 0 1	X 0	0 X	0 X	1 X
1 0 0 1	0 0 0 0	X 1	0 X	0 X	X 1
1 0 1 0	X X X X	X X	X X	X X	X X
1 0 1 1	X X X X	X X	X X	X X	X X
1 1 0 0	X X X X	X X	X X	X X	X X
1 1 0 1	X X X X	X X	X X	X X	X X
1 1 1 0	X X X X	X X	X X	X X	X X
1 1 1 1	X X X X	X X	X X	X X	X X

Σχ. 7.16. Πίνακας καταστάσεων με τις εισόδους των flip-flops.

Το διάγραμμα καταστάσεων περιλαμβάνει $2^4 = 16$ καταστάσεις, από τις οποίες όμως, μόνο οι δέκα χρησιμοποιούνται. Οι υπόλοιπες έξι (στην προκειμένη περίπτωση οι καταστάσεις 10, 11, 12, 13, 14, 15) μας είναι προς το παρόν αδιάφορες. Το γεγονός αυτό αντικατοπτρίζεται στον παραπάνω πίνακα καταστάσεων με το σύμβολο X στο τμήμα της επόμενης κατάστασης και κατ' επέκταση στο αντίστοιχο τμήμα των

εισόδων. Γίνεται φανερό ότι η ύπαρξη αυτών των συνθηκών αδιαφορίας θα βοηθήσει πάρα πολύ στην απλοποίηση των συναρτήσεων εισόδου, όπως θα δούμε αμέσως μετά. Όσον αφορά στη συμπλήρωση του πίνακα για τις καταστάσεις 0 μέχρι και 9, αυτή γίνεται κατά τα γνωστά με τη βοήθεια του πίνακα διέγερσης για τα J-K flip-flop.

Στη συνέχεια, θα απλοποιήσουμε με τη βοήθεια του χάρτη Karnaugh τις συναρτήσεις εισόδου. Η ύπαρξη πολλών αδιάφορων όρων (X), λόγω των αδιάφορων καταστάσεων, οδηγεί εύκολα σε απλοποιημένες συναρτήσεις εισόδου.

$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00				
01			1	
11	X	X	X	X
10	X	X	X	X

$$J_3 = Q_0Q_1Q_2$$

$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00			1	
01	X	X	X	X
11	X	X	X	X
10			X	X

$$J_2 = Q_0Q_1$$

$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00		1	X	X
01		1	X	X
11	X	X	X	X
10			X	X

$$J_1 = Q_0\overline{Q_3}$$

$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	1	X	X	1
01	1	X	X	1
11	X	X	X	X
10	1	X	X	X

$$J_0 = 1$$

$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	X	X	X	X
10		1	X	X

$$K_3 = Q_0$$

$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	X	X	X	X
01			1	
11	X	X	X	X
10	X	X	X	X

$$K_2 = Q_0Q_1$$

$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	X	X	1	
01	X	X	1	
11	X	X	X	X
10	X	X	X	X

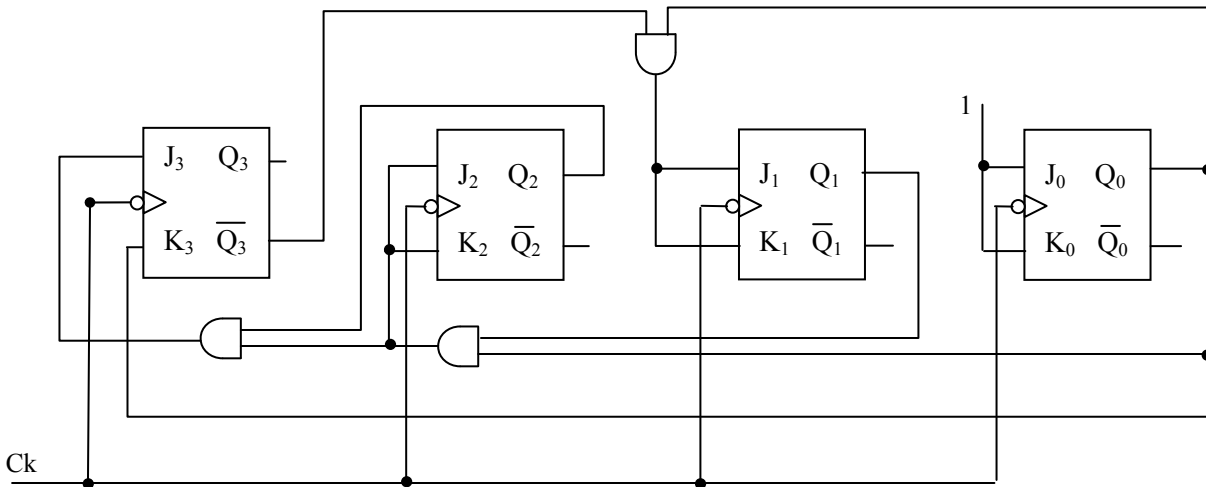
$$K_1 = Q_0\overline{Q_3}$$

$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	X	1	1	X
01	X	1	1	X
11	X	X	X	X
10	X	1	X	X

$$K_0 = 1$$

Σχ. 7.17. Χάρτες Karnaugh για απλοποίηση των εισόδων των flip-flops.

Έχοντας προσδιορίσει τις συναρτήσεις εισόδου των flip-flops, προχωρούμε στη σχεδίαση του κυκλώματος.



Σχ. 7.18. Το σύγχρονο ακολουθιακό κύκλωμα.

Στο σημείο αυτό η διαδικασία της σχεδίασης έχει ολοκληρωθεί. Υπάρχει όμως ένα ερώτημα το οποίο πρέπει να απαντηθεί. Μήπως η ύπαρξη των αδιάφορων καταστάσεων, και η επακόλουθη αυθαίρετη αξιοποίησή τους κατά τη διαδικασία της απλοποίησης, δημιούργησε κάποια προβλήματα στο κύκλωμα αυτό και δεν λειτουργεί όπως θα θέλαμε; Δηλαδή, μήπως οι αποφάσεις που πήραμε αυθαίρετα για κάποια X ίσα με 1, οδήγησαν το κύκλωμα σε μη επιθυμητή λειτουργία, όπως για παράδειγμα σ' ένα κύκλο απαρίθμησης 11, 12, 13, 14, 11, 12, 13, 14, 11, ... ; Σε μια τέτοια περίπτωση το κύκλωμα ποτέ δεν φτάνει σε μια από τις επιθυμητές καταστάσεις και εγκλωβίζεται σε ένα μη επιθυμητό βρόχο απαρίθμησης.

Για να απαντήσουμε στο ερώτημα αυτό θα πρέπει να ελέγξουμε τη λειτουργία του κυκλώματος. Άρα, απαιτείται να αναλύσουμε το κύκλωμα που μόλις σχεδιάσαμε και έτσι να επιβεβαιώσουμε την ορθότητα της λειτουργίας του. Στην περίπτωση που αυτό δεν συμβαίνει, τότε θα πρέπει να επαναλάβουμε τη διαδικασία της σχεδίασής του, αρχίζοντας και πάλι από το σχεδιασμό του διαγράμματος καταστάσεων και την κατάστρωση του σχετικού πίνακα καταστάσεων, αλλά όντας πιο αυστηροί και σαφείς στις προδιαγραφές μας. Συνεπώς κατά τη σχεδίαση σύγχρονων ακολουθιακών κυκλωμάτων τα οποία έχουν αδιάφορες καταστάσεις, επιβάλλεται ένα επιπλέον βήμα, αυτό του ελέγχου της ορθής λειτουργίας τους.

Ο έλεγχος αυτός γίνεται, όπως μόλις αναφέραμε, με την ανάλυση του κυκλώματος του Σχ. 7.18. Ακολουθούμε επομένως τα εξής βήματα.

1) Γράφουμε τις συναρτήσεις εισόδου των flip-flops.

$$J_3 = Q_0 Q_1 Q_2 \quad J_2 = K_2 = Q_0 Q_1 \quad J_1 = K_1 = Q_0 \overline{Q_3} \quad J_0 = K_0 = 1$$

$$K_3 = Q_0$$

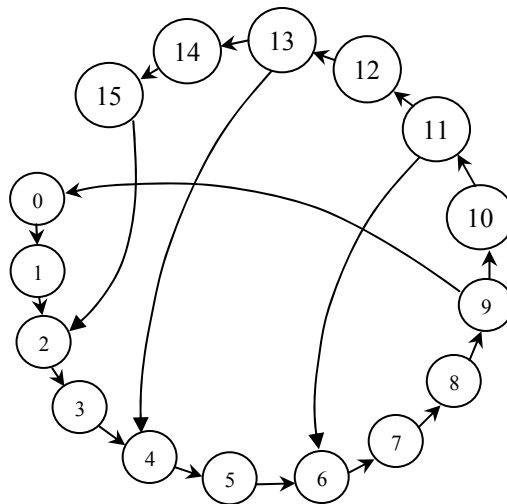
2) Καταστρώνουμε τον πίνακα καταστάσεων. Βασιζόμενοι στις συναρτήσεις εισόδου και στις τιμές της παρούσας κατάστασης, προσδιορίζουμε την επόμενη κατάσταση του κυκλώματος, όπως φαίνεται στον παρακάτω πίνακα, Σχ. 7.19.

Παρούσα Κατάσταση	Επόμενη Κατάσταση	Είσοδοι			
		$Q_3Q_2Q_1Q_0$	$Q_3Q_2Q_1Q_0$	J_3K_3	J_2K_2
0 0 0 0	0 0 0 1	0 X	0 X	0 X	1 X
0 0 0 1	0 0 1 0	0 X	0 X	1 X	X 1
0 0 1 0	0 0 1 1	0 X	0 X	X 0	1 X
0 0 1 1	0 1 0 0	0 X	1 X	X 1	X 1
0 1 0 0	0 1 0 1	0 X	X 0	0 X	1 X
0 1 0 1	0 1 1 0	0 X	X 0	1 X	X 1
0 1 1 0	0 1 1 1	0 X	X 0	X 0	1 X
0 1 1 1	1 0 0 0	1 X	X 1	X 1	X 1
1 0 0 0	1 0 0 1	X 0	0 X	0 X	1 X
1 0 0 1	0 0 0 0	X 1	0 X	0 X	X 1
1 0 1 0	1 0 1 1	0 0	0 0	0 0	1 1
1 0 1 1	0 1 1 0	0 1	1 1	0 0	1 1
1 1 0 0	1 1 0 1	0 0	0 0	0 0	1 1
1 1 0 1	0 1 0 0	0 1	0 0	0 0	1 1
1 1 1 0	1 1 1 1	0 0	0 0	0 0	1 1
1 1 1 1	0 0 1 0	1 1	1 1	0 0	1 1

Σχ. 7.19. Πίνακας καταστάσεων του κυκλώματος του Σχ. 7.18.

Για τις καταστάσεις 0 μέχρι και 9 ο πίνακας αυτός είναι ίδιος με τον προηγούμενο πίνακα καταστάσεων του Σχ. 7.16, όπως είναι αναμενόμενο. Η διαφοροποίηση αρχίζει από την κατάσταση 10 και μετά. Το αντίστοιχο διάγραμμα καταστάσεων θα μας δείξει τον ακριβή τρόπο λειτουργίας του κυκλώματος και γι' αυτό προχωρούμε άμεσα στη σχεδίασή του.

3) Σχεδιάζουμε το διάγραμμα καταστάσεων.



Σχ. 7.20. Διάγραμμα καταστάσεων του κυκλώματος του Σχ. 7.18.

Παρατηρούμε ότι το κύκλωμα που σχεδιάσαμε διατρέχει όντως τον κύκλο των καταστάσεων 0 μέχρι και 9. Αν βρεθεί σε μία από τις αδιάφορες καταστάσεις, τότε με

τον πρώτο παλμό ρολογιού μεταβαίνει στην αμέσως επόμενη κατάσταση, και με τον δεύτερο παλμό μεταβαίνει σε μια από τις επιθυμητές (έγκυρες) καταστάσεις και συνεχίζει πλέον τον κύκλο κανονικά. Για παράδειγμα, αν βρεθεί στην κατάσταση 10, μεταβαίνει στην 11 και ακολούθως στην 6, οπότε και συνεχίζει. Δηλαδή αρχίζοντας από την κατάσταση 10 θα έχουμε την αλληλουχία καταστάσεων: 10, 11, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, ... Παρόμοια συμβαίνουν και όταν το κύκλωμα βρεθεί σε μία από τις υπόλοιπες μη επιτρεπτές καταστάσεις. Το αργότερο μετά από δύο παλμούς ρολογιού βρίσκεται σε μια από τις επιτρεπτές (έγκυρες) καταστάσεις οπότε και συνεχίζει κανονικά. Αυτού του είδους τα κυκλώματα λέμε ότι έχουν αυτόματη εκκίνηση (self-starting) και αυτόματη διόρθωση (self-correcting). Υπάρχουν κυκλώματα στα οποία δεν συμβαίνει αυτό. Σε μια τέτοια περίπτωση θα πρέπει να παρέμβουμε στη λειτουργία του κυκλώματος και να την διορθώσουμε, επαναλαμβάνοντας τμήμα ή και όλη τη διαδικασία της σχεδίασής του. Στην περίπτωση που μόλις εξετάσαμε, θεωρήσαμε ότι τα flip-flops που χρησιμοποιούμε δεν διαθέτουν ασύγχρονες (άμεσες) εισόδους εκκαθάρισης Cr ή πρόθεσης Pr. Αν διαθέτουν, τότε μπορούμε να τις αξιοποιήσουμε για να πετύχουμε την επιθυμητή εκκίνηση του κυκλώματος.

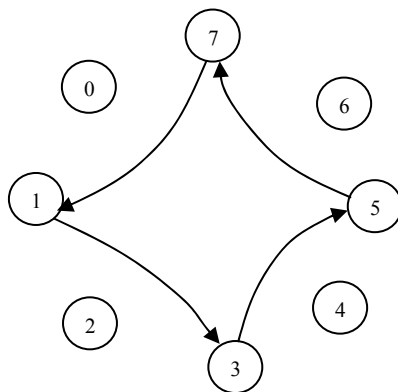
Παράδειγμα 7.4

Να σχεδιαστεί ένα σύγχρονο ακολουθιακό κύκλωμα το οποίο να διατρέχει διαδοχικά τους αριθμούς 1, 3, 5, 7. Η σχεδίαση να γίνει με χρήση T flip-flops.

Λύση

Θέλουμε να σχεδιάσουμε ένα κύκλωμα το οποίο μετράει μόνο τους περιττούς αριθμούς μέχρι το 7, και μετά να ξεκινάει πάλι από την αρχή. Δηλαδή να διατρέχει μόνο αυτές τις 4 καταστάσεις. Όμως, αν και οι καταστάσεις είναι τέσσερις, για την αναπαράστασή τους σε δυαδική μορφή χρειαζόμαστε 3 flip-flops προκειμένου να παραστήσουμε το 5 ή το 7. Με 3 flip-flops έχουμε 8 δυνατές καταστάσεις από τις οποίες μόνο οι 4 μας χρειάζονται. Ας θεωρήσουμε αρχικά ότι μας είναι αδιάφορο το πού θα πάει το κύκλωμα αν βρεθεί σε μία από αυτές και ας προσπαθήσουμε να το σχεδιάσουμε χρησιμοποιώντας flip-flops τύπου T.

Σχεδιάζουμε το διάγραμμα καταστάσεων και καταστρώνουμε τον πίνακα καταστάσεων. Μας είναι αδιάφορο το πού θα βρεθεί το κύκλωμα μετά την κατάσταση 0 ή 2 ή 4 ή 6, γεγονός που αντικατοπτρίζεται και στο τμήμα των εισόδων του πίνακα καταστάσεων με τις συνθήκες αδιαφορίας X.



Σχ. 7.21. Διάγραμμα καταστάσεων.

Παρούσα Κατάσταση $Q_2Q_1Q_0$	Επόμενη Κατάσταση $Q_2Q_1Q_0$	Είσοδοι		
		T_2	T_1	T_0
0 0 0	X X X	X	X	X
0 0 1	0 1 1	0	1	0
0 1 0	X X X	X	X	X
0 1 1	1 0 1	1	1	0
1 0 0	X X X	X	X	X
1 0 1	1 1 1	0	1	0
1 1 0	X X X	X	X	X
1 1 1	0 0 1	1	1	0

Σχ. 7.22. Πίνακας καταστάσεων με τις εισόδους των flip-flops.

Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

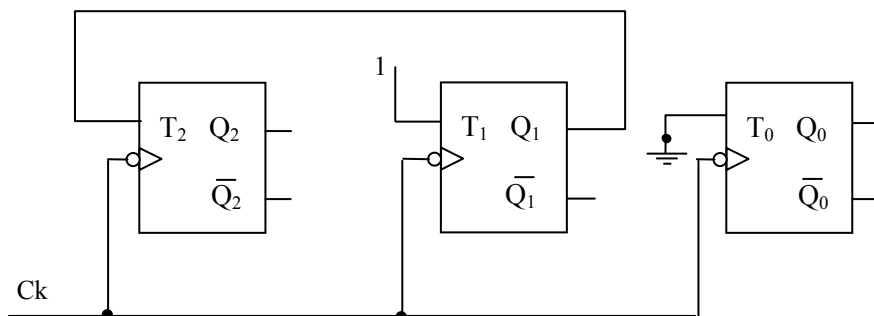
Σχ. 7.23. Πίνακας διέγερσης T flip-flop.

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

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	X		1	X
1	X		1	X

$T_2 = Q_1$

Σχ. 7.24. Χάρτης Karnaugh για απλοποίηση της εισόδου του flip-flop T_2 .



Σχ. 7.25. Το σύγχρονο ακολουθιακό κύκλωμα.

Σχεδιάζουμε το λογικό κύκλωμα στο Σχ. 7.25. Εφόσον δεν έχουμε κάνει κάποιο λάθος στη διαδικασία της σχεδίασης, είμαστε σίγουροι ότι το κύκλωμα θα διατρέχει τους αριθμούς 1, 3, 5, 7, όταν βρεθεί αρχικά σε κάποιον από αυτούς. Τι θα συμβεί όμως αν αρχικά βρεθεί σε μία από τις μη έγκυρες καταστάσεις 0, 2, 4, 6; Αυτό πρέπει να το ελέγξουμε. Έτσι, συνεχίζουμε με το επόμενο βήμα, της επαλήθευσης της ορθής (επιθυμητής) λειτουργίας του κυκλώματος που σχεδιάσαμε.

Αν και θα μπορούσαμε να ελέγξουμε τη λειτουργία του κυκλώματος αποσπασματικά και μόνο για τις μη επιτρεπτές καταστάσεις, εμείς επαναλαμβάνουμε όλη τη διαδικασία ανάλυσης.

Γράφουμε τις συναρτήσεις εισόδου.

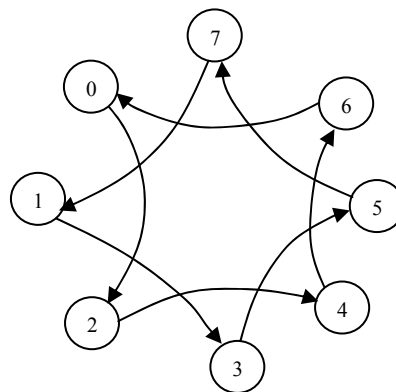
$$T_2 = Q_1 \quad T_1 = 1 \quad T_0 = 0$$

Ο πίνακας καταστάσεων προκύπτει εύκολα από την παρούσα κατάσταση και τις εισόδους σε συνδυασμό με τον πίνακα αληθείας του flip-flop.

Παρούσα Κατάσταση	Επόμενη Κατάσταση	Είσοδοι		
		$Q_2Q_1Q_0$	T_2	T_1
0 0 0	0 1 0	0	1	0
0 0 1	0 1 1	0	1	0
0 1 0	1 0 0	1	1	0
0 1 1	1 0 1	1	1	0
1 0 0	1 1 0	0	1	0
1 0 1	1 1 1	0	1	0
1 1 0	0 0 0	1	1	0
1 1 1	0 0 1	1	1	0

Σχ. 7.26. Πίνακας καταστάσεων του κυκλώματος του Σχ. 7.25.

Σχεδιάζουμε το νέο διάγραμμα καταστάσεων.

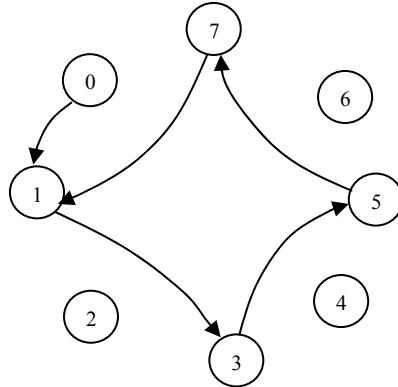


Σχ. 7.27. Διάγραμμα καταστάσεων του κυκλώματος του Σχ. 7.25.

Είναι φανερό ότι το κύκλωμα παρουσιάζει πρόβλημα! Αν βρεθεί σε μία από τις μη έγκυρες καταστάσεις 0 ή 2 ή 4 ή 6, τότε εγκλωβίζεται στον κύκλο 0, 2, 4, 6, 0, 2, 4, 6, 0, 2...

Για να λυθεί το πρόβλημα αυτό θα πρέπει να παρέμβουμε στο αρχικό στάδιο σχεδιασμού του διαγράμματος καταστάσεων και κατάστρωσης του σχετικού πίνακα καταστάσεων, αναγκάζοντας το κύκλωμα να μεταβεί σε μία από τις έγκυρες

καταστάσεις. Θα μπορούσαμε για παράδειγμα, να αναγκάσουμε το κύκλωμα μετά την κατάσταση 0 να μεταβεί στην κατάσταση 1. Με αυτό τον τρόπο σπάμε τον μη έγκυρο κύκλο 0, 2, 4, 6, 0, 2, ... Με βάση το νέο αυτό δεδομένο, επαναλαμβάνουμε τη διαδικασία σχεδίασης από την αρχή. Έτσι, σχεδιάζουμε το νέο διάγραμμα και πίνακα καταστάσεων και έχουμε:



Σχ. 7.28. Διορθωμένο διάγραμμα καταστάσεων του κυκλώματος του Σχ. 7.25.

Παρούσα Κατάσταση	Επόμενη Κατάσταση	Είσοδοι		
		Q ₂ Q ₁ Q ₀	Q ₂ Q ₁ Q ₀	T ₂
0 0 0	0 0 1	0	0	1
0 0 1	0 1 1	0	1	0
0 1 0	X X X	X	X	X
0 1 1	1 0 1	1	1	0
1 0 0	X X X	X	X	X
1 0 1	1 1 1	0	1	0
1 1 0	X X X	X	X	X
1 1 1	0 0 1	1	1	0

Σχ. 7.29. Πίνακας καταστάσεων του κυκλώματος του Σχ. 7.25.

Στη συνέχεια, προσδιορίζουμε τις απλοποιημένες συναρτήσεις εισόδου, με τη βοήθεια των χαρτών Karnaugh.

Q ₂ \ Q ₁ Q ₀	00	01	11	10
0	0		1	X
1	X		1	X

$$T_2 = Q_1$$

Q ₂ \ Q ₁ Q ₀	00	01	11	10
0	0	1	1	X
1	X	1	X	X

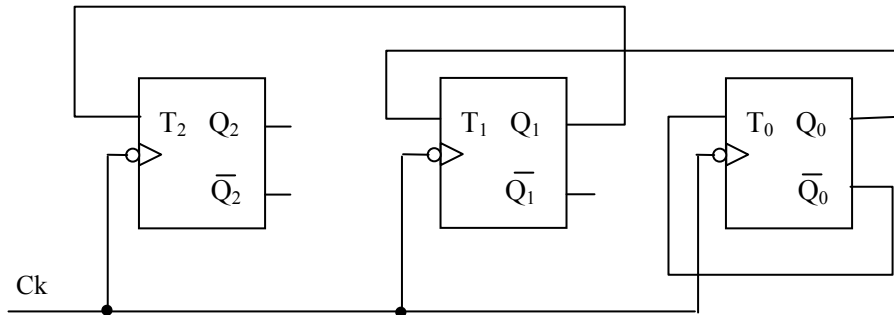
$$T_1 = Q_0$$

Q ₂ \ Q ₁ Q ₀	00	01	11	10
0	1	0	0	X
1	X	0	0	X

$$T_0 = Q_0$$

Σχ. 7.30. Χάρτες Karnaugh για απλοποίηση των εισόδων των flip-flops του πίνακα καταστάσεων του Σχ. 7.29.

Τέλος, σχεδιάζουμε το λογικό κύκλωμα το οποίο αντιστοιχεί στο διορθωμένο διάγραμμα καταστάσεων.



Σχ. 7.31. Το διορθωμένο σύγχρονο ακολουθιακό κύκλωμα που αντιστοιχεί στο διάγραμμα καταστάσεων του Σχ. 7.28.

Φυσικά, η λύση που επιλέξαμε για το σπάσιμο του κύκλου των μη έγκυρων καταστάσεων δεν είναι μοναδική. Μπορεί κανείς να επιλέξει άλλο τρόπο και επομένως να καταλήξει σε διαφορετικό τελικό κύκλωμα.

8. ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

- [1] Α. Λυριωτάκη, “Ψηφιακά Κυκλώματα”, Τ.Ε.Ι. Αθήνας, 1997.
- [2] M. Morris Mano, “Ψηφιακή Σχεδίαση”, Prentice-Hall International, Εκδόσεις Παπασωτηρίου, 1992.
- [3] A.K. Maini, “Digital Electronics – Principles, Devices and Applications”, J. Wiley & Sons, 2007.
- [4] Α.Ν. Μπαλουκτσής, “Ψηφιακά Κυκλώματα”, ΑΤΕΙ Σερρών, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Πληροφορικής & Επικοινωνιών, 2004.
- [5] Γ. Πρωτοπαπαδάκη, Α. Ζερβού, “Ψηφιακά Κυκλώματα και Ηλεκτρονικοί Υπολογισταί”, Σχολή Ναυτικών Δοκίμων, 1976.
- [6] Α.Ν. Σκόδρας, “Ψηφιακή Λογική: Ακολουθιακά Κυκλώματα”, Ελληνικό Ανοικτό Πανεπιστήμιο Πατρών.
- [7] Η. Taub, “Digital Circuits and Microprocessors”, McGraw-Hill Book Company, 1982.
- [8] Καραγιάννη Ευαγγελία, “Μαθήματα Ηλεκτρονικής - Τηλεπικοινωνιών”, Στρατ. Σχολή Ευελπίδων, 2003.
- [9] Κ. Τσαγκάρης, “Αρχιτεκτονικές Υπολογιστών”, Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς, 2007.