

Παναγιώτης Ηρ.Γιαννακόπουλος

Λογικά Κυκλώματα



Ελληνικά Ακαδημαϊκά Ηλεκτρονικά
Συγγράμματα και Βοηθήματα
www.kallipos.gr

HEALLINK
Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΠΑΝΑΓΙΩΤΗΣ ΗΡ. ΓΙΑΝΝΑΚΟΠΟΥΛΟΣ
ΚΑΘΗΓΗΤΗΣ

Λογικά Κυκλώματα

Αριθμητικά συστήματα, Δυαδικοί κώδικες, Κώδικας IEEE-754, Άλγεβρα Boole, Ελαχιστοποίηση λογικών συναρτήσεων, Λογικές πύλες, Πύλες με διακριτά στοιχεία, Κυκλώματα πράξεων, Δισταθείς πολυδομητές(*ff*), Απαριθμητές, Κωδικοποιητές, Αποκωδικοποιητές, Πολυπλέκτες, Μηχανές Καταστάσεων, Κυκλώματα Προσαρμογής, Προγραμματιζόμενες λογικές διατάξεις, Καταχωρητές, Μνήμες



Ελληνικά Ακαδημαϊκά Ηλεκτρονικά
Συγγράμματα και Βοηθήματα
www.kallipos.gr

Λογικά Κυκλώματα

Συγγραφή

Παναγιώτης Ηρ. Γιαννακόπουλος

Κριτικός αναγνώστης

Ιωάννης Καλλιακάτσος

Συντελεστές έκδοσης

ΓΛΩΣΣΙΚΗ ΕΠΙΜΕΛΕΙΑ: **Ελένη Νίκα**

ΓΡΑΦΙΣΤΙΚΗ ΕΠΙΜΕΛΕΙΑ: **TEKMON GEOMATICS**

ΤΕΧΝΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ: **TEKMON GEOMATICS**

Copyright © Σ.Ε.Α.Β, 2015



Το παρόν έργο αδειοδοτείται υπό τους όρους της άδειας Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 3.0.

ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Εθνικό Μετσόβιο Πολυτεχνείο

Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου

www.kallipos.gr

ISBN: 978-960-603-228-8

Αφιερώνεται στον Ηρακλή και την Κατερίνα

Περιεχόμενα

Πρόλογος.....	10
Εισαγωγή	11
Πίνακας συντομεύσεων-ακρωνύμια	13
Κεφάλαιο 1.....	15
1. Αριθμητικά Συστήματα.....	15
1.1. Εισαγωγή.....	15
1.2. Κύρια αριθμητικά συστήματα.....	16
1.3. Μετατροπή αριθμών μεταξύ των αριθμητικών συστημάτων.....	16
1.3.1. Μετατροπή ακέραιων αριθμών από το δεκαδικό σύστημα	16
1.3.2. Μετατροπή αριθμών μικρότερων της μονάδας από το δεκαδικό σύστημα αρίθμησης.....	18
1.3.3. Μετατροπή αριθμού από οποιοδήποτε σύστημα αρίθμησης στο δεκαδικό	20
1.3.4. Μετατροπή από και προς οποιαδήποτε βάση αρίθμησης.....	20
1.4. Συμπληρώματα αριθμού βάσης β ως προς βάσεις $\beta-1$ και β	21
1.5. Αναπαράσταση των θετικών και αρνητικών αριθμών στο δυαδικό σύστημα αρίθμησης	22
1.5.1. Προσημασμένη παράσταση θετικών και αρνητικών δυαδικών αριθμών	22
1.5.2. Παράσταση θετικών και αρνητικών δυαδικών αριθμών με συμπλήρωμα ως προς 1	23
1.5.3. Παράσταση θετικών και αρνητικών δυαδικών αριθμών με συμπλήρωμα ως προς 2	24
1.5.4. Σύγκριση των συστημάτων προσημασμένων αριθμών.....	25
1.5.5. Λειτουργικότητα της μεθόδου.....	26
1.6. Πράξεις στο δυαδικό σύστημα (με τα ψηφία 0 και 1).....	27
1.6.1. Πρόσθεση δυαδικών αριθμών.....	27
1.6.2. Αφαίρεση δυαδικών αριθμών	28
1.6.3. Πολλαπλασιασμός δυαδικών αριθμών	28
1.6.3.1 Πολλαπλασιασμός θετικών δυαδικών αριθμών.....	29
1.6.3.2 Πολλαπλασιασμός προσημασμένων δυαδικών αριθμών	30
1.6.3.3 Πολλαπλασιασμός προσημασμένων δυαδικών (αρνητικός πολλαπλασιαστής)	30
1.6.3.4 Πολλαπλασιασμός προσημασμένων δυαδικών (αρνητικός πολλαπλασιαστής)	31
1.6.4. Διαίρεση δυαδικών αριθμών.....	31
1.6.4.1 Αλγόριθμοι Διαίρεσης	32
1.6.4.2 Προσημασμένη Διαίρεση.....	32
1.7. Πράξεις με προσημασμένους δυαδικούς αριθμούς.....	33
1.7.1. Πρόσθεση αριθμών με συμπλήρωμα ως προς ένα.....	33
1.7.2. Δυαδική Πρόσθεση αριθμών με συμπλήρωμα ως προς δύο.....	34
1.7.3. Πρόσθεση τριών ψηφίων	36
1.8. Ασκήσεις – Ερωτήσεις	38
Αναφορές-Βιβλιογραφία.....	39
Κεφάλαιο 2.....	40
2. Κωδικοποίηση των ψηφίων του δεκαδικού συστήματος.....	40
2.1. Κωδικοποίηση των ψηφίων του δεκαδικού μέσω του δυαδικού συστήματος.....	40
2.2. Οκταδικός κωδικοποιημένος στο δυαδικό	40
2.3. Δεκαεξαδικός κωδικοποιημένος στο δυαδικό	41
2.4. Δεκαδικός κωδικοποιημένος στο δυαδικό	41
2.4.1. Δυαδικός κώδικας (Binary code).....	42
2.4.2. Οι κώδικες 5421 και 5311.....	42
2.4.3. Κώδικας Aiken ή κώδικας 2421 ή κώδικας $84-2-1$	42
2.4.4. Κώδικας Υπερβολής κατά τρία.....	43
2.4.5. Δυπενταδικός και Δυδεκαδικός κώδικας.....	43
2.4.6. Κώδικας 2-από-5.....	44

2.4.7.	<i>Κώδικας Gray (Κυκλικός κώδικας)</i>	44
2.5.	Αλφαριθμητικοί κώδικες–Κωδικοποίηση χαρακτήρων μέσω του δυαδικού συστήματος αρίθμησης.....	45
2.5.1.	<i>Χρήση της ισοτιμίας στην ασφαλή κωδικοποίηση των χαρακτήρων</i>	45
2.5.2.	<i>Κώδικες διόρθωσης σφαλμάτων</i>	46
2.5.3.	<i>Κώδικας Hamming</i>	48
2.6.	Αριθμητική κινητής υποδιαστολής και το πρότυπο IEEE-754.....	55
2.6.1.	<i>Η ιστορία των πράξεων κινητής υποδιαστολής</i>	55
2.6.2.	<i>Υλικό Κινητής Υποδιαστολής</i>	55
2.6.3.	<i>Αριθμοί κινητής υποδιαστολής -Εισαγωγή</i>	55
2.6.4.	<i>Το εύρος των αριθμών</i>	56
2.6.5.	<i>Αναπαράσταση αριθμών κινητής υποδιαστολής</i>	56
2.6.6.	<i>Παράσταση αριθμών απλής / διπλής ακριβείας</i>	56
2.6.7.	<i>Αριθμητική κινητής υποδιαστολής</i>	57
2.6.8.	<i>Μετατροπή αριθμών μεταξύ της IEEE-754 και του δεκαδικού</i>	58
2.6.9.	<i>Όρια της αναπαράστασης IEEE-754</i>	59
2.6.10.	<i>Πράξεις αριθμών κινητής υποδιαστολής (πρόσθεση και πολλαπλασιασμός)</i>	59
2.7.	Ασκήσεις – Ερωτήσεις	61
Αναφορές-Βιβλιογραφία.....		63
Κεφάλαιο 3.....		64
3.	Άλγεβρα Boole και ελαχιστοποίηση λογικών συναρτήσεων.....	64
3.1.	Εισαγωγή.....	64
3.2.	Παράσταση των μεταβλητών της Άλγεβρας Boole και πράξεις με αυτές.....	64
3.3.	Αξιώματα της Άλγεβρας Boole.....	65
3.4.	Θεωρήματα της Άλγεβρας Boole.....	65
3.4.1.	<i>Θεώρημα 1ο</i>	66
3.4.2.	<i>Θεώρημα 2ο</i>	66
3.4.3.	<i>Θεώρημα 3ο</i>	66
3.4.4.	<i>Θεώρημα 4ο</i>	67
3.4.5.	<i>Θεώρημα 5ο- (Θεωρήματα De Morgan)</i>	67
3.4.6.	<i>Θεώρημα 6ο</i>	67
3.4.7.	<i>Θεώρημα 7ο– Γενίκευση του θεωρήματος De Morgan</i>	68
3.4.8.	<i>Θεώρημα 8ο</i>	68
3.5.	Μελέτη της Άλγεβρας με λογικά κυκλώματα.....	69
3.6.	Λογικά κυκλώματα διακοπών.....	69
3.7.	Ανάλυση ενός λογικού κυκλώματος με διακόπτες.....	71
3.8.	Σύνθεση ενός λογικού κυκλώματος με διακόπτες.....	73
3.9.	Μέθοδοι ελαχιστοποίησης λογικών συναρτήσεων.....	74
3.9.1.	<i>Ελαχιστοβάθμιοι και μεγιστοβάθμιοι όροι</i>	74
3.9.1.1	Ελαχιστοβάθμιος όρος (Minterm) - ΕΛΒΟ	74
3.9.1.2	Μεγιστοβάθμιος όρος (Maxterm) - ΜΕΒΟ	75
3.9.2.	<i>Άλγεβρική ελαχιστοποίηση λογικών συναρτήσεων</i>	77
3.9.3.	<i>Μέθοδος ελαχιστοποίησης με διαγράμματα Venn</i>	77
3.9.4.	<i>Μέθοδοι ελαχιστοποίησης με τους χάρτες Karnaugh</i>	79
3.9.5.	<i>Χάρτης Karnaugh με περισσότερες από τέσσερις μεταβλητές</i>	83
3.9.6.	<i>Μέθοδος ελαχιστοποίησης με τη μέθοδο Quine-Mc Cluskey</i>	84
3.9.7.	<i>Αδιάφοροι όροι</i>	86
3.10.	Ασκήσεις – Ερωτήσεις	88
Αναφορές-Βιβλιογραφία.....		93
Κεφάλαιο 4.....		94
4.	Λογικές Πύλες- Κυκλώματα βασικών πράξεων.....	94
4.1.	Λογικές Πύλες	94

4.2	Μετασηματισμός των πυλών AND, OR, NOT με πύλες NAND, NOR.....	96
4.3.	Πύλη αποκλειστικής διάζευξης (XOR).....	98
4.3.1.	Πύλη XNOR	100
4.4.	Σύνθετες πύλες δύο επιπέδων (AOI και OAI).....	100
4.5.	Βασική Ανάλυση και Σχεδίαση Κυκλωμάτων	101
4.6.	Δυαδικός ημιαθροιστής.....	104
4.7.	Δυαδικός αθροιστής τριών εισόδων ή πλήρης αθροιστής (Full adder).....	105
4.8.	Σειριακός πλήρης αθροιστής (Serial full adder).....	107
4.9.	Παράλληλος πλήρης αθροιστής (Parallel full adder)	108
4.9.1.	Αθροιστής τετραψήφιων δυαδικών αριθμών με χρήση Ο.Κ.	110
4.10.	Σειριακός πλήρης αφαιρέτης (Serial full subtractor).....	111
4.10.1.	Κυκλώματα αθροιστών και αφαιρετών.....	113
4.10.2.	Κύκλωμα δυαδικού πολλαπλασιαστή	114
4.11.	Καθυστερήσεις πυλών.....	115
4.12.	Κυκλώματα σύγκρισης.....	116
4.12.1.	Σειριακός δυαδικός συγκριτής	116
4.12.2.	Παράλληλος δυαδικός συγκριτής	119
4.12.3.	Συγκριτής δύο δυαδικών ψηφίων.....	120
4.13.	Κυκλώματα μετατροπής κωδίκων	123
4.14.	Έλεγχος των τιμών δυαδικών ψηφίων.....	124
4.15.	Ασκήσεις – Ερωτήσεις	126
	Αναφορές-Βιβλιογραφία.....	131
	Κεφάλαιο 5.....	132
5.	Κατασκευή πυλών με διακριτά στοιχεία	132
5.1.	Εισαγωγή.....	132
5.2.	Πρακτικά κυκλώματα πυλών OR και AND με τη χρήση διακριτών στοιχείων	132
5.3.	Πρακτικά κυκλώματα πυλών NOT.....	136
5.4.	Πρακτικά κυκλώματα πυλών NAND, NOR.....	138
5.4.1.	Πύλες NOR και NAND με διόδους και τρανζίστορ DTL	140
5.4.2.	Η πύλη XOR με διακριτά στοιχεία	141
5.5.	Ολοκληρωμένα κυκλώματα.....	142
5.5.1.	Οικογένεια ολοκληρωμένων κυκλωμάτων TTL	144
5.5.2.	Οικογένεια ολοκληρωμένων κυκλωμάτων CMOS	146
5.5.3.	Οικογένεια ολοκληρωμένων κυκλωμάτων ECL.....	148
5.6.	Γενικευμένες πύλες NOR και NAND	149
5.6.1.	Υβριδικά λογικά CMOS memristor.....	150
5.7.	Κωδικοποιητές – Αποκωδικοποιητές με διακριτά στοιχεία	150
5.8.	Κυκλώματα Πολυδονητών.....	153
5.8.1.	Ασταθής πολυδονητής.....	154
5.8.2.	Ασταθής πολυδονητής με πύλες.....	155
5.8.3.	Μονοσταθής πολυδονητής	157
5.8.4.	Μονοσταθής πολυδονητής με πύλες.....	159
5.8.5.	Δισταθής πολυδονητής.....	160
5.9.	Ασκήσεις–Ερωτήσεις	163
	Αναφορές-Βιβλιογραφία.....	166
	Κεφάλαιο 6.....	167
6.	Λογικά Ακολουθιακά Κυκλώματα	167

6.1.	Δισταθείς πολυδονητές (f-f) με πύλες λογικής.....	167
6.2.	Στοιχεία μνήμης (flip-flop)	168
6.2.1.	<i>R-S flip-flop</i>	168
6.2.2.	<i>J-K flip-flop</i>	169
6.2.3.	<i>Ωρολογιακό J-K flip-flop</i>	170
6.2.4.	<i>T flip-flop</i>	172
6.2.5.	<i>D flip-flop</i>	173
6.2.6.	<i>R-S-T flip-flop</i>	174
6.2.7.	<i>Master slave flip-flop</i>	175
6.3.	Καταχωρητές	176
6.3.1.	<i>Καταχωρητές ολίσθησης</i>	176
6.3.2.	<i>Ολισθητής με R-S flip-flop</i>	177
6.4.	Απαριθμητές.....	180
6.4.1.	<i>Δυαδικοί Απαριθμητές</i>	180
6.4.2.	<i>Σύγχρονος δυαδικός απαριθμητής</i>	182
6.4.3.	<i>Απαριθμητές ανάδρασης</i>	187
6.4.4.	<i>Απαριθμητές ως προς μέτρο m</i>	188
6.4.5.	<i>Απαριθμητής δακτυλίου</i>	191
6.4.6.	<i>Απαριθμητής Johnson</i>	191
6.5.	Γεννήτρια ψευδοτυχαίων δυαδικών αριθμών	193
6.6.	Διαγράμματα καταστάσεων στα ακολουθιακά κυκλώματα.....	193
6.6.1.	<i>Περιγραφή ακολουθιακών κυκλωμάτων με διαγράμματα καταστάσεων</i>	194
6.6.2.	<i>Συνδυαστικά και ακολουθιακά κυκλώματα</i>	194
6.7.	Υποδειγματικά λυμένες ασκήσεις.....	197
6.7.1.	<i>Άσκηση 1^η Απαριθμητής ως προς μέτρο M</i>	197
6.7.2.	<i>Άσκηση 2^η Δεκαδικός απαριθμητής με χρήση του κώδικα XS3</i>	199
6.7.3.	<i>Άσκηση 3^η Μη σειριακός απαριθμητής</i>	202
6.7.4.	<i>Άσκηση 4^η Αμφίδρομος απαριθμητής οκτώ καταστάσεων</i>	205
6.8.	Μηχανές καταστάσεων.....	207
6.8.1.	<i>Μηχανές Mealy και Moore</i>	208
6.8.1.1	<i>Μηχανή Καταστάσεων Moore</i>	208
6.8.1.2	<i>Μηχανή Καταστάσεων Mealy</i>	212
6.8.2.	<i>Υποδειγματικά λυμένες ασκήσεις</i>	214
6.8.2.1	<i>Άσκηση 5^η Σχεδίαση ακολουθιακού κυκλώματος</i>	214
6.8.2.2	<i>Άσκηση 6^η Αναγνώριση μορφών</i>	217
6.9.	Μετρητές με μηχανές καταστάσεων	223
6.9.1.	<i>Μετρητής δύο δυαδικών ψηφίων</i>	223
6.9.2.	<i>Μετρητής αυτόματης εκκίνησης</i>	225
6.9.3.	<i>Μετρητής οκτώ δυαδικών ψηφίων</i>	226
6.9.4.	<i>Άνω / κάτω περιορισμένος μετρητής των 4-bit</i>	227
6.10.	Άσκησης – Ερωτήσεις	229
Αναφορές-Βιβλιογραφία.....		233
Κεφάλαιο 7.....		234
7.	Λογικά Κυκλώματα Προσαρμογής	234
7.1.	Μετατροπείς ψηφιακών σημάτων σε αναλογικά	234
7.1.1	<i>Χρήση των μετατροπέων D/A για την περιστροφή αξόνων</i>	236
7.2.	Μετατροπείς αναλογικών σημάτων σε ψηφιακά σήματα	237
7.2.1.	<i>Μετατροπή της πληροφορίας γωνίας άξονα σε ψηφιακή πληροφορία</i>	241
7.3.	Αποκωδικοποιητές.....	243
7.3.1.	<i>Λειτουργία του αποκωδικοποιητή</i>	243
7.3.2.	<i>Αποκωδικοποιητής 2-σε-4</i>	244
7.3.3.	<i>Είσοδοι ενεργοποίησης (Enable)</i>	244
7.3.4.	<i>Δομικά διαγράμματα</i>	245
7.3.5.	<i>Αποκωδικοποιητής 3-σε-8</i>	245
7.3.6.	<i>Πλεονεκτήματα αποκωδικοποιητή</i>	246

7.4.	Χρήση αποκωδικοποιητών	246
7.4.1.	Πρόσθεση με τη χρήση αποκωδικοποιητών	246
7.4.2.	Υλοποίηση με τη χρήση ενός αποκωδικοποιητή	247
7.4.3.	Επέκταση των αποκωδικοποιητών	247
7.4.4.	Τροποποίηση του συνήθους αποκωδικοποιητή	248
7.4.5.	Παράδειγμα Active-low αποκωδικοποιητή	248
7.4.6.	Παράγοντες σχεδίασης και η προέλευση τους	249
7.5.	Πολυπλέκτης	249
7.5.1.	Σύντηξη του πίνακα αλήθειας	250
7.5.2.	Πολυπλέκτης 4-σε-1	250
7.5.3.	Υλοποίηση συναρτήσεων με πολυπλέκτης	251
7.5.3.1	Υλοποίηση συνάρτησης με πολυπλέκτη 8 σε 1	251
7.5.3.2	Αθροιστής με χρήση πολυπλέκτη	252
7.5.3.3	Υλοποίηση πλήρους αθροιστή με διπλό πολυπλέκτη	253
7.6.	Απομονωτές (Buffers)	254
7.7.	MODEM	255
7.7.1.	Διαμόρφωση	256
7.8.	Ασκήσεις – Ερωτήσεις	258
Αναφορές-Βιβλιογραφία		260
Κεφάλαιο 8		261
8.	Προγραμματιζόμενες λογικές διατάξεις	261
8.1.	Εισαγωγή	261
8.2.	Αναπαράσταση κυκλωμάτων προγραμματιζόμενης λογικής	261
8.2.1.	Πλεονεκτήματα της προγραμματιζόμενης λογικής	263
8.3.	Στοιχεία προγραμματιζόμενης λογικής	264
8.4.	Μνήμη PROM	265
8.4.1.	Παραδείγματα χρήσης μνημών PROM	266
8.4.1.1	Σχεδίαση πυλών OR, XNOR	266
8.4.1.2	Σχεδίαση πλήρους αθροιστή με PROM	267
8.4.1.3	Αποκωδικοποιητής BCD σε δεκαδικό	269
8.5.	Εισαγωγή στα PLDs	271
8.5.1.	Προγραμματιζόμενες λογικές διατάξεις	271
8.5.1.1	Παράδειγμα με PLA	273
8.5.2.	Προγραμματιζόμενες λογικές διατάξεις(PAL)	274
8.5.2.1	Παράδειγμα με PAL	276
8.5.3.	Σύγκριση των PROM - PLA -PAL	277
8.6.	Σύγκριση GAL και PAL	279
8.7.	Ασκήσεις – Ερωτήσεις	280
Αναφορές-Βιβλιογραφία		283
Κεφάλαιο 9		284
9.	Καταχωρητές, Μνήμες, Αριθμητική και Λογική Μονάδα	284
9.1	Εισαγωγή στους καταχωρητές	284
9.1.1.	Καταχωρητές Ολίσθησης	286
9.1.2.	Καταχωρητές ολίσθησης με παράλληλη φόρτωση	287
9.1.2.1	Εφαρμογή του καταχωρητή ολίσθησης	287
9.1.3.	Οι καταχωρητές στο σύγχρονο υλικό	288
9.2.	Μνήμες	288
9.2.1	Η μνήμη RAM	289
9.2.1.1	Δομικό διάγραμμα της RAM	289
9.2.1.2	Απεικόνιση της μνήμης	289
9.2.1.3	Τα μεγέθη της μνήμης	290
9.2.1.4	Ανάγνωση- Εγγραφή στη μνήμη RAM	290

9.2.1.5	Τύποι στατικής μνήμη	290
9.2.1.6	RAM 4x4.....	294
9.2.1.7	Κατασκευή μεγαλύτερων μνημών RAM.....	294
9.2.1.8	Ανάλυση της 256K x 8 RAM.....	295
9.2.1.9	Εύρος διευθύνσεων	295
9.2.1.10	Κατασκευή «ευρύτερης» μνήμης	296
9.2.2.	Μνήμες RAM ειδικού τύπου	296
9.2.2.1	Δυναμική Μνήμη RAM (DRAM)	297
9.2.3.	Σύγκριση της δυναμικής και της στατικής μνήμης	297
9.2.4.	Η μνήμη μόνο ανάγνωσης (ROM)	298
9.2.4.1	Οργάνωση της ROM	299
9.2.5.	Σύγκριση της ROM και της RAM	300
9.3.	Βασική περιγραφή της Αριθμητικής μονάδας	301
9.3.1.	Αθροιστής 4-bit	301
9.3.2.	Ο Αθροιστής – Αφαιρέτης	301
9.3.3.	Αριθμητικές συναρτήσεις	303
9.4.	Βασική περιγραφή της Λογικής Μονάδας.....	305
9.5.	Συνδυάζοντας την Αριθμητική και τη Λογική Μονάδα	306
9.6.	Πράξεις και δικτυακές λειτουργίες με ένα δυαδικό ψηφίο	308
9.7.	Αρχιτεκτονική των υπολογιστών.....	309
9.8.	Ασκήσεις – Ερωτήσεις	311
	Αναφορές-Βιβλιογραφία.....	313
	Κεφάλαιο 10.....	314
10.	Παραδείγματα εφαρμογής λογικών κυκλωμάτων.....	314
10.1.	Παράδειγμα 1 ^ο (Οικιακή προστασία)	314
10.2.	Παράδειγμα 2 ^ο (Βιομηχανική ασφάλεια).....	315
10.3.	Παράδειγμα 3 ^ο (Τάισμα κατοικίδιου).....	315
10.4.	Παράδειγμα 4 ^ο (Έλεγχος Φαναριού (Κοινωνική εφαρμογή).....	316
	Βιβλιογραφία-Αναφορές.....	318
	ΠΑΡΑΡΤΗΜΑ	319
	Αντιστοίχιση Βασικών Ελληνικών Όρων στην Αγγλική.....	324
	Αλφαβητάρι Βασικών Όρων.....	326

Πρόλογος

Τα λογικά κυκλώματα είναι ο θεμέλιος λίθος των ηλεκτρονικών συστημάτων. Η ύλη των λογικών κυκλωμάτων ή των κυκλωμάτων ψηφιακής σχεδίασης ή ψηφιακών κυκλωμάτων, διδάσκεται σε όλα τα τμήματα των μηχανικών και σε πολλά τμήματα οικονομικών επιστημών, στο πλαίσιο των ψηφιακών τους μαθημάτων.

Σκοπός της παρούσας προσπάθειας στο χώρο των «Λογικών Κυκλωμάτων» είναι να καλύψει πλήρως τις ανάγκες των φοιτητών της Τριτοβάθμιας Εκπαίδευσης, διαφορετικών τμημάτων, που δεν έχουν όλα τις ίδιες απαιτήσεις καθώς και οποιουδήποτε ασχολείται με τα λογικά - ψηφιακά κυκλώματα. Η προσπάθεια αυτή αποτελεί το αποτέλεσμα της διδακτικής διαδικασίας και της αλληλεπίδρασης με τους φοιτητές για περισσότερο από μια δεκαπενταετία. Η μελέτη του βιβλίου αυτού απαιτεί συγκέντρωση κατά τη μελέτη του και, κυρίως, ενδιαφέρον και αγάπη για το σχεδιασμό λογικών κυκλωμάτων.

Δόθηκε ιδιαίτερη προσοχή στην ορολογία που χρησιμοποιήθηκε, ώστε να είναι η ευρύτερα αποδεκτή. Στην ύλη που περιλαμβάνεται δεν συμπεριελήφθη κανένα εργαλείο λογισμικού που χρησιμοποιείται στο λογικό σχεδιασμό κυκλωμάτων, καθώς η συγκεκριμένη ύλη αποτελεί διδακτικό περιεχόμενο, κυρίως, πρωτοετών φοιτητών και καθώς στο προσομοιωτικό σχεδιασμό με υπολογιστή το λογισμικό εξαρτάται από τον εκάστοτε διδάσκοντα, είναι αναλώσιμο και συνεχώς ανανεώνεται.

Έγινε διαχωρισμός της ύλης σε κεφάλαια με τέτοιο τρόπο, ώστε το περιεχόμενο των κεφαλαίων να είναι συμπληρωματικό αλλά και αυτοτελές. Συνεπώς η ύλη μπορεί να χρησιμοποιηθεί τμηματικά ή στο σύνολό της και από τους αναγνώστες και από τους διδάσκοντες. Έτσι, για παράδειγμα, το πέμπτο κεφάλαιο μπορεί να παραληφθεί από τα τμήματα των οικονομικών επιστημών, καθώς το πρώτο αναφέρεται κυρίως στη χρήση διακριτών στοιχείων, και το ένατο κεφάλαιο αντίστοιχα μπορεί να παραληφθεί από τα τμήματα μηχανικών, καθώς υπάρχει μάθημα σε μεγαλύτερο εξάμηνο σχετικό με την υλοποίηση μνημών και της αριθμητικής και λογικής μονάδας.

Οι στόχοι του βιβλίου είναι να δώσει με εύκολο και εύληπτο τρόπο τις λειτουργίες εκείνες που είναι βασικές για την κατανόηση των λογικών κυκλωμάτων. Επίσης βασικό στόχο αποτελεί η εμπάθυνση στη θεωρία και στο σχεδιασμό βασικών κυκλωμάτων. Για το σκοπό αυτό οι γνώσεις που παρέχονται στο σύγγραμμα αυτό συνδυάζονται με βασικές γνώσεις. Έγινε προσπάθεια να αποτελεί και αυτοδύναμο εγχειρίδιο για κάποιον που πλέον δεν έχει τη δυνατότητα να φοιτήσει σε κάποια σχολή.

Η συνύπαρξη θεωρίας και ασκήσεων αποσκοπεί στο να διευκολύνει τον αναγνώστη και, ταυτόχρονα, να του παρέχει τη βασική γνώση, και, τέλος, να επιβεβαιώσει την απόκτησή της μέσα από την επίλυση των προτεινόμενων ασκήσεων και ερωτημάτων πολλαπλής επιλογής.

Η βιβλιογραφία, που αναφέρεται στο τέλος κάθε κεφαλαίου, αποτελεί ερέθισμα για περαιτέρω μελέτη, και κυρίως είναι εγχειρίδια για την περαιτέρω εμπάθυνση στην ύλη από τους αναγνώστες του παρόντος συγγράμματος. Σε μερικά σημεία του κειμένου παρέχεται συγκεκριμένη αναφορά στη βιβλιογραφία και αυτό γίνεται, καθώς δεν υπάρχει εκτενής αναφορά στα θέματα στα οποία αναφέρεται.

Ελπίδα είναι αυτό το βιβλίο να εμπλουτίσει την ήδη υπάρχουσα ύλη του διαδικτύου και να βοηθήσει στην εξεύρεση νέων τρόπων παρουσίασης, όπως είναι το μοντέλο pyjan, του συγγραφέα.

Τελειώνοντας, οφείλω να ευχαριστήσω τον ομότιμο καθηγητή Απόστολο Κοσσίδα για την πολύτιμη βοήθειά του, τόσο κατά τη διάρθρωση, όσο και κατά τη διόρθωση της ύλης. Η συνεργασία με τον κριτικό αναγνώστη και τους άλλους συντελεστές (τη φιλόλογο και την εταιρεία που ανέλαβε την τεχνική επεξεργασία –σχήματα και τη δημιουργία του ηλεκτρονικού βιβλίου) ήταν εξαιρετική και φυσικά το τελικό αποτέλεσμα είναι ομαδικό.

Παρατηρήσεις από τους αναγνώστες είναι ευπρόσδεκτες ώστε να είναι εφικτή η περαιτέρω βελτίωση αυτού του συγγράμματος. Για τους διδάσκοντες είναι δυνατή η αποστολή των λύσεων των ασκήσεων που περιλαμβάνονται στο βιβλίο αυτό.

Εισαγωγή

Ακολουθεί μια γενική περιγραφή της διάρθρωσης της ύλης του παρόντος συγγράμματος για την ενημέρωση του αναγνώστη. Έτσι, στο πρώτο κεφάλαιο μελετώνται τα κύρια αριθμητικά συστήματα, οι αλγόριθμοι μετατροπής των ψηφίων του δεκαδικού συστήματος στα άλλα αριθμητικά συστήματα, που χρησιμοποιούν οι υπολογιστές, ο τρόπος εκτέλεσης των τεσσάρων βασικών πράξεων στο δυαδικό σύστημα καθώς και οι βασικοί αλγόριθμοι που τις διέπουν. Ακολουθεί η μετατροπή από και προς οποιαδήποτε βάση αρίθμησης, και τα συμπληρώματα των αριθμών βάσης β ως προς βάσεις $\beta-1$ και β , η αναπαράσταση των θετικών και αρνητικών αριθμών στο δυαδικό σύστημα αρίθμησης, η προσημασμένη παράσταση θετικών και αρνητικών δυαδικών αριθμών, και οι πράξεις με προσημασμένους δυαδικούς αριθμούς. Οι αλγόριθμοι πρόσθεσης, αφαίρεσης, πολλαπλασιασμού, διαίρεσης δυαδικών αριθμών, που βασίζονται στην πρόσθεση, με δεδομένο ότι οι πράξεις που εκτελεί ο υπολογιστής είναι η πρόσθεση, η σύγκριση και η μεταφορά δεδομένων.

Στο δεύτερο κεφάλαιο παρουσιάζεται αναλυτικά η κωδικοποίηση των αριθμών του δεκαδικού μέσω του δυαδικού συστήματος αρίθμησης. Αρχικά περιλαμβάνονται τα ψηφία του δεκαδικού συστήματος, κωδικοποιημένα στο δυαδικό. Ακολουθούν τα ψηφία του οκταδικού και του δεκαεξαδικού συστήματος κωδικοποιημένα, στο δυαδικό. Στη συνέχεια παρέχεται στον αναγνώστη οι κώδικες που χρησιμοποιούνται συχνά. Έτσι αναλύονται οι: δυαδικός κώδικας (Binary code), οι κώδικες βαρών, ο κώδικας Aiken, ο XS3 και άλλοι. Η κωδικοποίηση των χαρακτήρων στο δυαδικό σύστημα γίνεται με τους αλφαριθμητικούς κώδικες, που παρουσιάζονται στο παράρτημα. Ακολουθεί η ασφάλεια κωδικοποίησης των χαρακτήρων. Οι κώδικες διόρθωσης απλών σφαλμάτων, με κύρια αναφορά στον κώδικα Hamming. Βασικός για τους υπολογιστές είναι ο τρόπος αναπαράστασης μεγάλων αριθμών, η αριθμητική κινητής υποδιαστολής κατά το πρότυπο IEEE-754. Το κεφάλαιο ολοκληρώνεται με τις αριθμητικές πράξεις κατά το πρότυπο αυτό.

Στο επόμενο κεφάλαιο παρουσιάζονται τα αξιώματα και τα θεωρήματα της άλγεβρας Boole. Κατόπιν ακολουθεί η παρουσίαση λογικών κυκλωμάτων με διακόπτες. Βασική είναι η παρουσίαση των ελαχιστοβάθμιων και μεγιστοβάθμιων όρων και η χρήση τους. Επίσης, περιγράφονται αναλυτικά οι μέθοδοι ελαχιστοποίησης μίας συνάρτησης με τις κλασικές μεθόδους (Χάρτης Karnaugh, μέθοδος Quine Mc Cluskey).

Ακολουθεί η περιγραφή των βασικών λογικών πυλών AND, OR, NOT, NAND και NOR καθώς και οι πύλες XOR και XNOR. Ο μετασχηματισμός των βασικών πυλών AND, OR, NOT με πύλες NAND, NOR είναι βασικός στο σχεδιασμό και παρουσιάζεται αναλυτικά. Τέλος γίνεται μια περιγραφή των πυλών δύο επιπέδων τύπου AOI και OAI. Ακολουθεί η βασική ανάλυση και σχεδίαση κυκλωμάτων. Τα κυκλώματα των βασικών κυκλωμάτων αριθμητικών πράξεων της άλγεβρας (κύκλωμα ημιαθροιστή, κύκλωμα αθροιστή σειριακού και παράλληλου, κύκλωμα αφαιρέτη, κύκλωμα δυαδικού πολλαπλασιαστή). Το κεφάλαιο ολοκληρώνεται με τους δυαδικούς συγκριτές.

Το πέμπτο κεφάλαιο περιλαμβάνει το σχεδιασμό των λογικών πυλών με διακριτά στοιχεία και τρανζίστορ. Τα ολοκληρωμένα κυκλώματα βασικών λογικών κυκλωμάτων και ο σχεδιασμός τους με τεχνολογίες TTL, ECL, CMOS. Τέλος γίνεται αναφορά στην υλοποίηση των πυλών με τη νέα τεχνολογία των καρβονικών νανωσωλήνων και των memristors. Ακολουθούν οι κωδικοποιητές- αποκωδικοποιητές και το κεφάλαιο ολοκληρώνεται με την παρουσίαση των πολυδονητών, που είναι το βασικό στοιχείο για την ανάπτυξη των f-f των οποίων η μελέτη ακολουθεί.

Ακολουθεί η μελέτη του πλέον σημαντικού στοιχείου των λογικών κυκλωμάτων, του πολυδονητή δύο καταστάσεων (f-f), ο οποίος είναι βασικό στοιχείο στη μελέτη των ακολουθιακών κυκλωμάτων. Γίνεται περιγραφή όλων των τύπων των δισταθών πολυδονητών. Το σύγγραμμα στη συνέχεια παρουσιάζει τη βασική περιγραφή των καταχωρητών, των απαριθμητών, της γεννήτριας ψευδοτυχαίων αριθμών, και δίνονται τα διαγράμματα καταστάσεων των ακολουθιακών κυκλωμάτων. Μελετώνται τα ακολουθιακά κυκλώματα (σύγχρονα και ασύγχρονα) και ο τρόπος λειτουργίας τους.

Το κεφάλαιο αυτό περιλαμβάνει ένα σύνολο, αναλυτικά λυμένων, ασκήσεων είτε ακολουθιακών κυκλωμάτων είτε κυκλωμάτων με τη χρήση μηχανών καταστάσεων, και ολοκληρώνεται με την περιγραφή και τη σχεδίαση μετρητών με μηχανές καταστάσεων.

Ακολουθεί ο πλέον σύγχρονος τρόπος μελέτης και σχεδίασης λογικών συναρτήσεων, με τις προγραμματιζόμενες λογικές διατάξεις. Συγκεκριμένα παρουσιάζεται αναλυτικά ο τρόπος μελέτης και σχεδιασμού κάθε λογικής συνάρτησης, χρησιμοποιώντας τεχνικές PLA, PROM, PAL.

Στο επόμενο κεφάλαιο εξετάζουμε ορισμένα λογικά κυκλώματα προσαρμογής, τα οποία χρησιμοποιούνται συχνά στους ηλεκτρονικούς υπολογιστές. Αρχικά γίνεται αναφορά στον τρόπο μετατροπής ψηφιακών σημάτων σε αναλογικά και αναλογικών σε ψηφιακά και δίνεται λεπτομερής ανάλυση χρήσιμων κυκλωμάτων

προσαρμογής. Στη συνέχεια, γίνεται εκτενής περιγραφή των αποκωδικοποιητών, των κυκλωμάτων πολυπλεξίας, αναφέροντας και τις εφαρμογές τους, και το κεφάλαιο ολοκληρώνεται με τον απομονωτή και το modem.

Η κεντρική μονάδα επεξεργασίας του υπολογιστή αποτελείται από τους καταχωρητές, τη μονάδα ελέγχου, την αριθμητική και λογική μονάδα, το ρολόι και, φυσικά, τους διαύλους δεδομένων. Θα γίνει μια επισκόπηση των βασικών αυτών μονάδων, ολοκληρώνοντας τις θεμελιώδεις γνώσεις που απαιτούνται για έναν υπολογιστή. Αρχικά γίνεται μια εισαγωγή στους καταχωρητές με τη χρήση των ήδη μελετημένων $f\text{-}f$. Ακολουθεί εκτενής περιγραφή των μνημών RAM με πλήρη ανάλυση του τρόπου κατασκευής τους, η μνήμη ROM και άλλες χρησιμοποιούμενες δυναμικές μνήμες. Ακολουθεί η περιγραφή των βασικών πράξεων της αριθμητικής και λογικής μονάδας. Ολοκληρώνοντας, αναφέρονται οι δικτυακές λειτουργίες και, τέλος, δίνεται η περιγραφή της αρχιτεκτονικής των υπολογιστών.

Τέλος περιλαμβάνονται κλασικές εφαρμογές της καθημερινότητας, με χρήση λογικών κυκλωμάτων, για την καλύτερη κατανόηση και εμπέδωση της ύλης από τους αναγνώστες.

Λογικά Κυκλώματα

Πίνακας συντομεύσεων-ακρωνύμια

ΑΛΜ	Αριθμητική και Λογική Μονάδα
ΕΛΒΟ	Ελαχιστοβάθμιοι όροι
ΗΥ	Ηλεκτρονικός Υπολογιστής
ΚΜΕ	Κεντρικής Μονάδας Επεξεργασίας
ΚΥ	Κινητή Υποδιαστολή
ΜΕΒΟ	Μεγιστοβάθμιοι όροι
ΟΚ	Ολοκληρωμένο Κύκλωμα
ΑΔΡΣ	Address
Α/Δ	Analog to Digital
ΑΛΥ	Arithmetic Logic Unit
ΑΟΙ	And Or Invert
ΑSΠΙ	American Standards Code for Information Interchange
CS	Chip Select
CMOS	Complementary Metal Oxide Semiconductor
CNTFET	Carbon Nano Tube Field Effect Transistor
CPLDs	Complex programmable Logic Devices
CRC	Cyclic redundancy check
DDR-RAM	Double Data Rate Random Access Memory
DTL	Diode Transistor Logic
Δ/Α	Digital to Analog
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECL	Emitter Coupled Logic
EN	Enable
FET	Field Effect Transistor
F-F (<i>f-f</i>)	Flip- flop
FIFO	First In First Out
FPGA	Field Programmable Gate Arrays
GAL	Generic array Logic
IC	Integrated Circuit
LCA	Logic Cell Array
LSB	Least Significant bit
LSI	Large Scale Integration

MODEM	Modulator-Demodulator
MSB	Most Significant bit
PC	Program Counter
PE	Parity Error
PLD	Programmable Logic Devices
POS	Product of Sum
PROM	Programmable Read Only Memory
RAM	Random Access Memory
RDRAM	Rambus Dynamic Random Access Memory
ROM	Read Only Memory
RTL	Resistor Transistor Logic
SAR	Successive Approximation Register
SDRAM	Synchronous Dynamic Random Access Memory
SOP	Sum of Products
SPLD	Simple programmable Logic Devices
SSI	Small Scale Integration
TTL	Transistor Transistor Logic
VLSI	Very Large Scale Integration
WR	Write

Κεφάλαιο 1

Σύνοψη

Στο κεφάλαιο αυτό θα παρουσιαστούν τα κύρια αριθμητικά συστήματα, οι αλγόριθμοι μετατροπής μεταξύ των συστημάτων για την κάθε μια περίπτωση, ο τρόπος εκτέλεσης των τεσσάρων βασικών πράξεων στο δυαδικό σύστημα καθώς και οι βασικοί αλγόριθμοι που τις διέπουν. Θα γίνει διάκριση σε πράξεις με υπερχειλίση, μετατροπή ακέραιων αριθμών από το δεκαδικό σύστημα, μετατροπή αριθμών μικρότερων της μονάδας από το δεκαδικό σύστημα αρίθμησης, μετατροπή αριθμού από οποιοδήποτε σύστημα αρίθμησης στο δεκαδικό. Επίσης θα γίνει μετατροπή από και προς οποιαδήποτε βάση αρίθμησης, συμπληρώματα αριθμού βάσης β ως προς βάσεις $\beta-1$ και β , αναπαράσταση των θετικών και αρνητικών αριθμών στο δυαδικό σύστημα αρίθμησης, προσημασμένη παράσταση θετικών και αρνητικών δυαδικών αριθμών, παράσταση θετικών και αρνητικών δυαδικών αριθμών με συμπλήρωμα ως προς 1 και ως προς 2, πράξεις με προσημασμένους δυαδικούς αριθμούς. Ακόμα θα παρουσιαστεί δυαδική πρόσθεση αριθμών με συμπλήρωμα ως προς 1 και ως προς 2, πρόσθεση τριών ή περισσότερων δυαδικών ψηφίων, πρόσθεση, αφαίρεση, πολλαπλασιασμός δυαδικών αριθμών με (α) αρνητικό πολλαπλασιαστέο και (β) αρνητικό πολλαπλασιαστή, αλγόριθμοι διαίρεσης δυαδικών αριθμών και τέλος η προσημασμένη διαίρεση.

Προαπαιτούμενη γνώση

Οι βασικές γνώσεις των αριθμητικών συστημάτων και ο τρόπος εκτέλεσης των πράξεων μεταξύ των αριθμών του δεκαδικού συστήματος.

1. Αριθμητικά Συστήματα

1.1. Εισαγωγή

Ένα αριθμητικό σύστημα ορίζεται από ένα σύνολο τιμών, που χρησιμοποιούνται για την αναπαράσταση μίας ποσότητας. Η μελέτη των αριθμητικών συστημάτων δεν περιορίζεται μόνο στους ηλεκτρονικούς υπολογιστές. Κάθε μέρα χρησιμοποιούμε αριθμούς, και γνωρίζοντας τον τρόπο λειτουργίας τους, μας παρέχεται η πληροφορία για το πώς ένας υπολογιστής χειρίζεται και αποθηκεύει τους αριθμούς.

Η ανθρωπότητα, κατά τη διάρκεια της πορείας της, χρησιμοποίησε σχήματα ή σύμβολα για την αναπαράσταση των αριθμών, αρχικά με τη χρήση γραμμών. Η γραφική απεικόνιση αριθμών ήταν όμως δύσκολη. Οι Βαβυλώνιοι χρησιμοποιούσαν το εξηναδικό σύστημα (sexagesimal) οι δε Μάγια το εικοσαδικό (vigesimal). Οι Ρωμαίοι επινόησαν ένα αριθμητικό σύστημα για τους αριθμούς από το 1 έως το 1 εκατομμύριο με τη χρήση 7 συμβόλων (τα γράμματα I, V, X, L, C, D, M). Το γνωστό μας αραβικό σύστημα χρησιμοποιήθηκε για πρώτη φορά πριν μόλις 2300 χρόνια. Η εισαγωγή του μηδενός, ώστε να προσδιορίσει την αξία ενός ψηφίου, ήταν πολύ σημαντική. (Maxfield, 2009; Αριθμητικά Συστήματα, 2015)

Οι ηλεκτρονικοί υπολογιστές εκτελούν τις πράξεις, χρησιμοποιώντας όχι το δεκαδικό σύστημα αρίθμησης αλλά το δυαδικό, που περιλαμβάνει τους δυαδικούς αριθμούς «1» και «0». Τα ψηφιακά ηλεκτρονικά είναι ο κόσμος των υπολογιστών, των ολοκληρωμένων κυκλωμάτων, των αριθμομηχανών. Η χρήση των ψηφιακών ηλεκτρονικών επεκτείνεται σε πολλαπλές εφαρμογές της καθημερινής μας ζωής και, για αυτό το λόγο, θα πρέπει να γνωρίζουμε τις βασικές αρχές τους.

Ένας οποιοσδήποτε αριθμός, σε οποιοδήποτε σύστημα αρίθμησης, μπορεί να παρασταθεί από την παρακάτω σχέση: (Κοσσιδάς, 1996)

$$N = \sum_{i=-m}^{i=n-1} \Psi_i B^i \text{ για κάθε } 0 \leq \Psi_i \leq B-1 \quad (1-1)$$

Στη σχέση 1-1 έχουμε:

n = το πλήθος των ακεραίων ψηφίων του αριθμού

m = το πλήθος των δεξιά της υποδιαστολής ψηφίων του αριθμού

Ψ_i = τα ψηφία του αριθμού

B = η βάση του συστήματος αρίθμησης

i = η τάξη του ψηφίου Ψ_i

Στη συνέχεια θα αναφερθούμε στα κύρια αριθμητικά συστήματα καθώς και στον τρόπο μετατροπής των αριθμών από ένα σύστημα σε οποιοδήποτε άλλο.

1.2. Κύρια αριθμητικά συστήματα

Το πλέον χρησιμοποιούμενο σύστημα αρίθμησης, στην καθημερινή μας ζωή, είναι το δεκαδικό. Αυτό περιλαμβάνει τα ψηφία από 0 έως 9 και έχει σαν βάση του (base, radix) τον αριθμό 10. Παρατηρούμε ότι η βάση του συστήματος δεν ανήκει στα ψηφία του συστήματος, αλλά υπερβαίνει κατά μία μονάδα το μεγαλύτερο ψηφίο του. Το ίδιο ισχύει για όλα τα αριθμητικά συστήματα, ανεξάρτητα από τη βάση, σύμφωνα με την οποία αυτά έχουν ορισθεί. (Κοσσιδάς, 1996; Givone, 2002; Predko, 2005; Maxfield, 2009; Pritchard, 2015)

Παράδειγμα 1ο:

Ο αριθμός 1974 μπορεί να γραφτεί στο δεκαδικό σύστημα σύμφωνα με τον προηγούμενο τύπο ως εξής (παρατηρούμε ότι πρόκειται για ακέραιο αριθμό, άρα στον προηγούμενο τύπο (1-1) το $m=0$): Το μοντέλο είναι

$$N = \sum_{i=0}^{i=3} \Psi_i B_i = \Psi_0 B^0 + \Psi_1 B^1 + \Psi_2 B^2 + \Psi_3 B^3 \quad (1-2)$$

$$1974 = 4 \cdot 10^0 + 7 \cdot 10^1 + 9 \cdot 10^2 + 1 \cdot 10^3$$

Εκτός από το δεκαδικό αριθμητικό σύστημα, τα κύρια αριθμητικά συστήματα με τα οποία θα ασχοληθούμε είναι το δυαδικό, το οκταδικό και το δεκαεξαδικό.

Όπως αναφέραμε, το δυαδικό σύστημα είναι αυτό που χρησιμοποιείται στους ηλεκτρονικούς υπολογιστές. Τα δύο ψηφία, που το περιγράφουν, μπορούν να παρασταθούν εύκολα με τη βοήθεια ηλεκτρονικών εξαρτημάτων, όπως οι διακόπτες ή τα τρανζίστορ, των οποίων οι έξοδοι μπορούν να οδηγηθούν σε δύο καταστάσεις. Η μία από αυτές παριστάνει το «0» και η άλλη το «1» (με τη χρήση διακοπών η πρώτη είναι ανοικτός διακόπτης και η δεύτερη κλειστός διακόπτης).

Τα ψηφία που χρησιμοποιεί το κάθε σύστημα αρίθμησης είναι (για τα κύρια συστήματα):

ΔΥΑΔΙΚΟ: 0, 1

ΟΚΤΑΔΙΚΟ: 0, 1, 2, 3, 4, 5, 6, 7

ΔΕΚΑΔΙΚΟ: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

ΔΕΚΑΕΞΑΔΙΚΟ: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Συμπληρωματικά αναφέρουμε και τα στοιχεία ενός άλλου συστήματος αρίθμησης, του δωδεκαδικού (Duo-decimal), το οποίο χρησιμοποιήθηκε για τις διάτρητες κάρτες των υπολογιστών. Στο σύστημα αυτό μετά τα δέκα γνωστά ψηφία χρησιμοποιούνται τα γράμματα **T**(Ten) ή **X** και **E**(Eleven).

Η βάση του δυαδικού συστήματος είναι το **2**, του οκταδικού το **8**, και του δεκαεξαδικού το **16**. Το δεκαεξαδικό σύστημα, δεδομένου ότι περιέχει και ψηφία μεγαλύτερα του 9, χρησιμοποιούνται και τα έξι πρώτα γράμματα του λατινικού αλφαβήτου (A, B, C, D, E, F) για να αναπαρασταθούν τα υπόλοιπα ψηφία του.

Στον πίνακα 1.1, που ακολουθεί, δίνεται η παράσταση των αριθμών από το 1 έως το 20 στα τέσσερα κύρια αριθμητικά συστήματα που σχετίζονται αμέσως με το δυαδικό.

Κάθε αριθμός, που θα χρησιμοποιούμε, θα παρίσταται με το γράμμα **N**(number) και δείκτη τη βάση του αριθμητικού συστήματός του π.χ. ο $N_{10}=3569$ υποδηλώνει αριθμό του δεκαδικού συστήματος, ο $N_8 = 13467$ του οκταδικού συστήματος και ο $N_2=10101101$ για το δυαδικό σύστημα.

1.3. Μετατροπή αριθμών μεταξύ των αριθμητικών συστημάτων

Θα μελετήσουμε τις περιπτώσεις μετατροπής ενός αριθμού που ανήκει σ' ένα σύστημα αρίθμησης με βάση **a**, σ' ένα ισοδύναμο αριθμό στο σύστημα με βάση **β**. (Κοσσιδάς, 1996; Balabanian, & Carlson, 2007; Katz, 2005; Parhami, 2010; Floyd, 2013)

1.3.1. Μετατροπή ακέραιων αριθμών από το δεκαδικό σύστημα

Έστω ότι ο αριθμός **N** παριστάνεται στα δύο συστήματα ως εξής:

$$N_{\beta} = \Psi_n, \Psi_{n-1}, \dots, \Psi_0$$

$$N_{\alpha} = \Phi_v, \Phi_{v-1}, \dots, \Phi_0$$

(1-3)

όπου Ψ_i ψηφία $\{0, 1, 2, \dots, \beta-1\}$ και Φ_i ψηφία $\{0, 1, 2, \dots, \alpha-1\}$. Επειδή οι δύο αυτές παραστάσεις αντιπροσωπεύουν τον ίδιο αριθμό, προφανώς θα ισχύει ότι:

$$N_{\beta} = \sum_{i=0}^{n-1} \Psi_i B^i = N_{\alpha} = \sum_{i=0}^{v-1} \Phi_i \alpha^i \quad (1-4)$$

Υποθέτουμε ότι τα ψηφία Ψ_0, Ψ_1, Ψ_n είναι γνωστά, δηλαδή είναι γνωστή η παράσταση του αριθμού με βάση β , και ζητάμε την ισοδύναμη παράστασή του με βάση α . Θα πρέπει δηλαδή να υπολογίσουμε τα νέα ψηφία $\Phi_0, \Phi_1, \dots, \Phi_v$.

Είναι γνωστό από την αριθμητική ότι για κάθε σύστημα αρίθμησης ισχύει:

$$\frac{\Lambda}{\delta} = \pi + \frac{Y}{\delta} \quad (1-5)$$

όπου Λ ο διαιρέτης, δ ο διαιρετέος, π το ηλίκο και Y το υπόλοιπο.

Για τη βάση $\beta = 10$ έχουμε:

$$N_{10} = \Psi_{n-1}\beta^{n-1} + \Psi_{n-2}\beta^{n-2} + \Psi_{n-3}\beta^{n-3} + \dots + \Psi_1\beta^1 + \Psi_0\beta^0 \quad (1-6)$$

$$N_{10} = \Phi_{v-1}a^{v-1} + \Phi_{v-2}a^{v-2} + \Phi_{v-3}a^{v-3} + \dots + \Phi_1a^1 + \Phi_0a^0 \quad (1-7)$$

$$\frac{N_{10}}{a} = \Phi_{v-1}a^{v-2} + \Phi_{v-2}a^{v-3} + \Phi_{v-3}a^{v-4} + \dots + \Phi_1a^0 + \frac{\Phi_0}{a} \quad (1-8)$$

Συγκρίνοντας τις εξισώσεις (1-5) και (1-8) έχουμε:

$$\Pi_0 = \Phi_{v-1}a^{v-2} + \Phi_{v-2}a^{v-3} + \dots + \Phi_1a^0 \quad (1-9)$$

$$Y_0 = \Phi_0 \quad (1-10)$$

όπου Π_0 είναι το ηλίκο και Y_0 το υπόλοιπο της πρώτης διαίρεσης των αριθμών N_{10}/a . Δηλαδή το υπόλοιπο της πρώτης διαίρεσης του αριθμού N_{10} με το a μας δίνει το ψηφίο Φ_0 της κατώτερης τάξης του αριθμού N_{10} στο σύστημα αρίθμησης με βάση το a . Αν συνεχίσουμε τις διαιρέσεις, διαιρώντας το Π_0 με το a , βρίσκουμε τα Π_1 και Y_1 . Συνεχίζοντας με αυτό τον τρόπο, βρίσκουμε διαδοχικά όλα τα ψηφία του αριθμού στο σύστημα αρίθμησης με βάση a , δηλαδή τα $\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_{v-1}$. Κατά συνέπεια, η μετατροπή ενός ακεραίου αριθμού του δεκαδικού συστήματος σε άλλα συστήματα επιτυγχάνεται με συνεχείς διαιρέσεις του αριθμού με τη βάση του νέου συστήματος στο οποίο μετατρέπεται ο αριθμός, από τις οποίες αυτό που μας ενδιαφέρει είναι το υπόλοιπο, το οποίο είναι ακεραίος μικρότερος από τη βάση του νέου συστήματος (ουσιαστικά προκύπτει από την πράξη MOD- ακεραίο υπόλοιπο-).

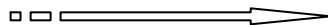
Παράδειγμα 2ο:

Βασική αρχή στην απεικόνιση αυτή (*Μοντέλο μετατροπής και απεικόνισης ργίαν*, το οποίο χρησιμοποιήθηκε για πρώτη φορά από τον συγγραφέα) είναι ότι γράφουμε τον αριθμό μας δεξιά και κατόπιν εκτελούμε συνεχείς διαιρέσεις με τη βάση του συστήματος, στο οποίο θέλουμε να μετατρέψουμε τον αριθμό μας. Στο ίδιο επίπεδο με τον αρχικό αριθμό γράφουμε το ακεραίο ηλίκο (DIV) και ακριβώς από κάτω το ακεραίο υπόλοιπο (MOD). Σταματάμε τη διαδικασία όταν το ηλίκο γίνει μικρότερο από τον αριθμό με τον οποίο διαιρούμε, αποτέλεσμα το οποίο καταβάζουμε στη θέση του υπολοίπου. Η ανάγνωση του αριθμού που προκύπτει, από αριστερά προς τα δεξιά, δίνει τον αριθμό στο νέο αριθμητικό σύστημα. (Εννοείται ότι δε γνωρίζουμε εξαρχής το τελικό πλήθος των ψηφίων του αριθμού στο νέο σύστημα, ώστε να δημιουργήσουμε το κατάλληλο πλήθος θέσεων).

A) Μετατροπή από το δεκαδικό στο δυαδικό σύστημα.

Εστω ότι θέλουμε να μετατρέψουμε τον δεκαδικό $N_{10} = 53_{10}$ στο δυαδικό σύστημα.

ΠΗΛΙΚΟ		1	3	6	13	26	53
ΥΠΟΛΟΙΠΟ	1	1	0	1	0	1	X (διαίρεση δια 2)

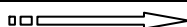


Συνεπώς ο αριθμός $N_{10} = 53_{10}$ στο δυαδικό σύστημα αρίθμησης είναι ο $N_2 = 110101_2$.

B) Μετατροπή από το δεκαδικό στο οκταδικό σύστημα.

Να μετατραπεί ο αριθμός $N_{10} = 53_{10}$ στο οκταδικό σύστημα.

ΠΗΛΙΚΟ		6	53
ΥΠΟΛΟΙΠΟ	6	5	X (διαίρεση δια 8)



Ο αριθμός $N_{10} = 53_{10}$ στο οκταδικό σύστημα είναι ο $N_8 = 65_8$.

Γ) Μετατροπή από το δεκαδικό στο δεκαεξαδικό σύστημα.

Να μετατραπεί ο αριθμός $N_{10} = 53_{10}$ στο δεκαεξαδικό σύστημα.

ΠΗΛΙΚΟ
ΥΠΟΛΟΙΠΟ

	3	53
3	5	X (διαίρεση δια 16)

Ο αριθμός στο δεκαεξαδικό σύστημα είναι ο $N_{16} = 35_{16}$.

Επομένως ο αριθμός **53** του δεκαδικού συστήματος στα άλλα τρία συστήματα, που σχετίζονται με το δυαδικό σύστημα, έχει τους εξής ισοδύναμους αριθμούς: $N_2 = 110101_2$, $N_8 = 65_8$ και $N_{16} = 35_{16}$. Η επαλήθευση των αποτελεσμάτων γίνεται εύκολα μέσω της σχέσης 1-1:

$$110101_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 16 + 0 + 4 + 0 + 1 = 53_{10}$$

$$65_8 = 6 \cdot 8^1 + 5 \cdot 8^0 = 48 + 5 = 53$$

$$35_{16} = 3 \cdot 16^1 + 5 \cdot 16^0 = 48 + 5 = 53$$

Δ) Έστω ότι θέλουμε να μετατρέψουμε τον αριθμό $N_{10} = 1973_{10}$ σε όλα τα γνωστά αριθμητικά συστήματα. Θα ξεκινήσουμε με τον υπολογισμό του ισοδύναμου δυαδικού αριθμού.

	1	3	7	15	30	61	123	246	493	986	1973
1	1	1	1	0	1	1	0	1	0	1	X (διαίρεση δια 2)

Ο αριθμός 1973 του δεκαδικού συστήματος στο δυαδικό σύστημα είναι ο $N_2 = 11110110101_2$. Για το οκταδικό σύστημα θα έχουμε:

	3	30	246	1973
3	6	6	5	X (διαίρεση δια 8)

ή ισοδύναμα έχουμε:(κλασσική απεικόνιση)

	ΠΗΛΙΚΟ	ΥΠΟΛΟΙΠΟ
1973 : 8	246	5
246 : 8	30	6
30 : 8	3	6
3 : 8	0	3

Όπως διαπιστώνετε, με τον κλασικό τρόπο παρουσίασης που έχουν υιοθετήσει τα περισσότερα βιβλία, πρέπει να αντιστρέψουμε το αποτέλεσμα δηλαδή να γράψουμε πρώτο το τελευταίο ψηφίο κ.λπ. Άρα ο αριθμός 1973 του δεκαδικού στο οκταδικό σύστημα είναι ο $N_8 = 3665_8$.

Τέλος στο δεκαεξαδικό σύστημα θα έχουμε:

ΠΗΛΙΚΟ	ΥΠΟΛΟΙΠΟ
7	123
7	B
5	X (διαίρεση δια 16)

Συνεπώς στο δεκαεξαδικό σύστημα αντιστοιχεί στον αριθμό $N_{16} = 7B5_{16}$.

Άρα ο αριθμός $1973_{10} = 7B5_{16} = 3665_8 = 11110110101_2$

1.3.2. Μετατροπή αριθμών μικρότερων της μονάδας από το δεκαδικό σύστημα αρίθμησης

Στην περίπτωση αυτή κάνουμε ακριβώς το αντίθετο. Δηλαδή, αντί να διαιρούμε τον αριθμό με τη βάση του νέου συστήματος αρίθμησης, εκτελούμε συνεχείς πολλαπλασιασμούς με τη βάση του νέου συστήματος αρίθμησης. Η διαδικασία ολοκληρώνεται, όταν το υπόλοιπο είναι μηδέν. (Nelson, Nagle, Carroll, & Irwin, 1995; Κοσσίδης, & Γιαννακόπουλος, 2006; Wakerly, 2006; Balabanian, & Carlson, 2007; Roth, & Kinney, 2014)

Έστω ο κλασματικός δεκαδικός αριθμός:

$$N_{10} = \sum_{i=-m}^{-1} \Psi_i \beta^i = \Psi_{-1} \beta^{-1} + \Psi_{-2} \beta^{-2} + \Psi_{-3} \beta^{-3} + \dots + \Psi_{-m} \beta^{-m} \quad (1-11)$$

$$N_{10} = \sum_{i=-\mu}^{-1} \Phi_i \alpha^i = \Phi_{-1} \alpha^{-1} + \Phi_{-2} \alpha^{-2} + \Phi_{-3} \alpha^{-3} + \dots + \Phi_{-\mu} \alpha^{-\mu} \quad (1-12)$$

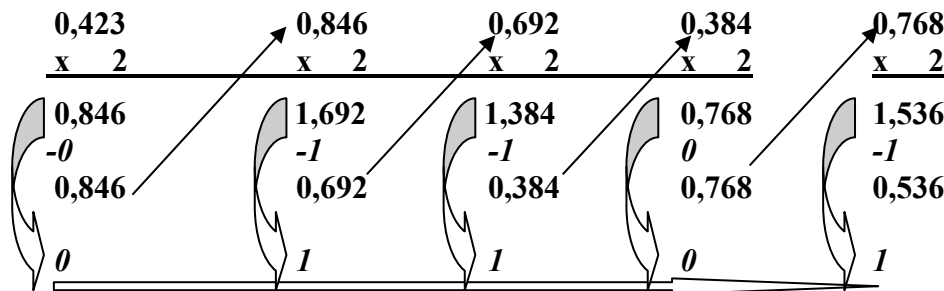
Έχοντας γνωστά τα Ψ_i , θέλουμε να υπολογίσουμε τα Φ_i . Πολλαπλασιάζοντας και τα δύο μέλη της εξίσωσης (1-12) με α έχουμε:

$$\alpha N_{10} = \Phi_{-1}\alpha^0 + \Phi_{-2}\alpha^{-1} + \Phi_{-3}\alpha^{-2} + \dots + \Phi_{-\mu}\alpha^{-\mu+1} \quad (1-13)$$

Βλέπουμε ότι το Φ_{-1} είναι το πρώτο ψηφίο του αριθμού N_{10} με βάση το α . Με συνεχείς πολλαπλασιασμούς υπολογίζουμε και τα υπόλοιπα ψηφία $\Phi_{-2}, \dots, \Phi_{-\mu}$ μέχρι να πετύχουμε μια ικανοποιητική προσέγγιση.

Παράδειγμα 3ο:

A) Να μετατραπεί ο κλασματικός δεκαδικός αριθμός $N_{10} = 0,423_{10}$ στο δυαδικό σύστημα αρίθμησης.



Αρα 01101 είναι τα ακέραια μέρη μετά τους πολλαπλασιασμούς επί 2

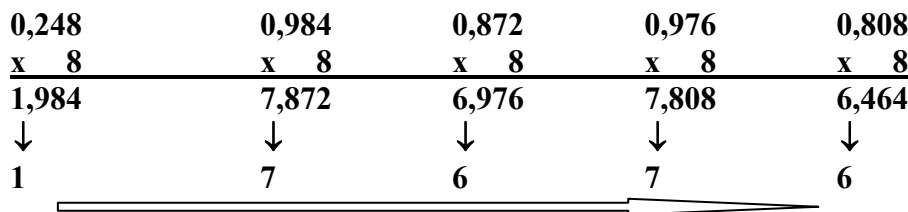
Διαδικασία μετατροπής δεκαδικού κλασματικού αριθμού:

Σύμφωνα με το παραπάνω παράδειγμα, κρατάμε το ακέραιο ψηφίο που προκύπτει μετά τον πολλαπλασιασμό του αριθμού με τη βάση στην οποία θέλουμε να μετατρέψουμε τον αριθμό και, ταυτόχρονα, αφαιρούμε αυτό το ακέραιο ψηφίο από τον αριθμό ο οποίος έχει προκύψει από τον πολλαπλασιασμό. Ο νέος αριθμός, που προκύπτει, μετά την αφαίρεση του ακεραίου τμήματος, πολλαπλασιάζεται εκ νέου με τη βάση κοκ. Λαμβάνοντας όλα τα προκύπτοντα ακέραια μέρη (0 ή 1 για το δυαδικό), έχουμε τον αριθμό στο νέο σύστημα, ο οποίος αριθμός είναι και κλασματικός!). Ο νέος πολλαπλασιασμός με τη βάση γίνεται μόνο με το νέο κλασματικό μέρος. (Η διαδικασία αποδίδεται από αριστερά προς τα δεξιά για λόγους οικονομίας χώρου).

Συνεπώς ο δεκαδικός αριθμός $N_{10} = 0,423_{10}$ είναι ο αριθμός $N_2 = 0,01101_2$ του δυαδικού. Είναι προφανές ότι ο νέος αριθμός δεν είναι ακριβώς ίσος με τον αρχικό αλλά κατά προσέγγιση. Αυτό προκύπτει, διότι η διαδικασία ολοκληρώνεται χωρίς να έχει προκύψει τελικό υπόλοιπο ίσο με το μηδέν (εδώ 0,536).

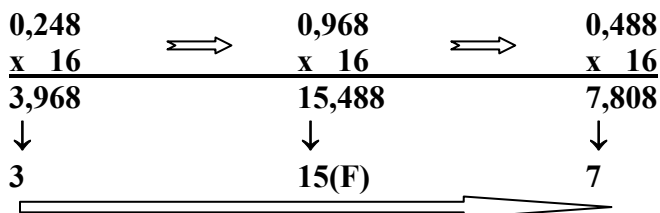
B) Να μετατραπεί ο δεκαδικός κλασματικός αριθμός $N_{10} = 0,248$ στο οκταδικό σύστημα αρίθμησης.

Εργαζόμενοι όπως προηγουμένως, έχουμε:



Άρα στο οκταδικό σύστημα ο κλασματικός αριθμός είναι ο $N_8 = 0,17676_8$.

Γ) Να μετατραπεί ο δεκαδικός κλασματικός αριθμός $N_{10} = 0,248$ στο δεκαεξαδικό σύστημα αρίθμησης.



Στο δεκαεξαδικό σύστημα ο κλασματικός αριθμός είναι ο $N_{16} = 0,3F7_{16}$.

Σε περίπτωση που ο δεκαδικός έχει και ακέραιο και κλασματικό μέρος, τότε η μετατροπή γίνεται, λαμβάνοντας χωριστά το ακέραιο μέρος και χωριστά το κλασματικό μέρος. Εκτελούμε τις αντίστοιχες διαδικασίες, όπως αναφέρθηκε παραπάνω, και, τέλος, γράφουμε τον τελικό αριθμό, ο οποίος προκύπτει από τη σύνθεση του ακεραίου και του κλασματικού μέρους.

1.3.3. Μετατροπή αριθμού από οποιοδήποτε σύστημα αρίθμησης στο δεκαδικό

Υπολογίζεται η τιμή του αριθμού στο δεκαδικό σύστημα αρίθμησης, χρησιμοποιώντας τις δυνάμεις της βάσης του συστήματος στην οποία βρίσκεται ο αριθμός. Η μετατροπή σ' αυτή την περίπτωση είναι πολύ απλή:

A) Μετατροπή αριθμού από το δυαδικό στο δεκαδικό.

Να μετατραπεί ο δυαδικός $N_2 = 101011$ στον αντίστοιχο δεκαδικό N_{10} .

Ξεκινάμε πάντα από το λιγότερο σημαντικό ψηφίο -με τη μηδενική δύναμη- μέχρι να πάρουμε όλους τους όρους, αυξάνοντας τη δύναμη της βάσης συνεχώς κατά ένα, ή μετρούμε το πλήθος των ψηφίων και η δύναμη του μεγιστοβάθμιου όρου προκύπτει, αν αφαιρέσουμε ένα (1) από τον αριθμό που αντιστοιχεί στο πλήθος των ψηφίων, δηλαδή αν το πλήθος των ψηφίων είναι 6, τότε η δύναμη του μεγιστοβάθμιου όρου είναι το 5)

$$N_2 = 101011 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 32 + 0 + 8 + 0 + 2 + 1 = 43_{10}$$

B) Από οκταδικό σε δεκαδικό. Να μετατραπεί ο $N_8 = 3765_8$ στον αντίστοιχο N_{10}

$$N_8 = 3765 = 3 \cdot 8^3 + 7 \cdot 8^2 + 6 \cdot 8^1 + 5 \cdot 8^0 = 1536 + 448 + 48 + 5 = 2037_{10}$$

Γ) Από δεκαεξαδικό σε δεκαδικό. Να μετατραπεί ο $N_{16} = 1F$ στον αντίστοιχο N_{10}

$$N_{16} = 1F = 1 \cdot 16^1 + F \cdot 16^0 = 1 \cdot 16^1 + 15 \cdot 16^0 = 16 + 15 = 31_{10}$$

1.3.4. Μετατροπή από και προς οποιαδήποτε βάση αρίθμησης

Σημαντική παρατήρηση, πριν προχωρήσουμε σε οποιαδήποτε νέα μετατροπή, είναι να γνωρίζουμε ότι κάθε δεκαεξαδικός αριθμός ισοδυναμεί με ένα τετραψηφίο στο δυαδικό σύστημα και κάθε αριθμός του οκταδικού συστήματος ισοδυναμεί με ένα τριψηφίο στο δυαδικό σύστημα (Πίνακας 1.1).

ΔΕΚΑΔΙΚΟ	ΔΥΑΔΙΚΟ	ΟΚΤΑΔΙΚΟ	ΔΕΚΑΕΞΑΔΙΚΟ
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	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
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Πίνακας 1.1. Παράσταση αριθμών στα κύρια αριθμητικά συστήματα

Ένας συνηθισμένος τρόπος, για να μετατρέψουμε έναν αριθμό που ανήκει σε οποιοδήποτε σύστημα αρίθμησης σ' έναν ισοδύναμο αριθμό διαφορετικού συστήματος, είναι να τον μετατρέψουμε αρχικά στον ισοδύναμο δεκαδικό και μετά στο επιθυμητό σύστημα.

Έστω λοιπόν ότι θέλουμε να κάνουμε μια μετατροπή από το δεκαεξαδικό στο οκταδικό σύστημα. Θα βρούμε πρώτα τον αντίστοιχο αριθμό στο δεκαδικό σύστημα και, στη συνέχεια, από το δεκαδικό θα βρούμε τον αντίστοιχο οκταδικό αριθμό. (Wakerly, 2006; Parhami, 2010; Mano, & Ciletti, 2014)

Παράδειγμα 4ο:

Α' τρόπος(εφαρμογή του μοντέλου ryian)

Να μετατραπεί ο $N_{16} = ABC$ στο οκταδικό σύστημα $N_8 =$;

$$N_{16} = ABC = A \cdot 16^2 + B \cdot 16^1 + C \cdot 16^0 = 10 \cdot 16^2 + 11 \cdot 16^1 + 12 \cdot 16^0 = 2560 + 176 + 12 = 2748$$

$$N_{16} = ABC \rightarrow N_{10} = 2748_{10}$$

	5	42	343	2748
5	2	7	4	X (διαίρεση δια 8)

Στο οκταδικό ο αριθμός είναι ο $N_8 = 5274_8$

Ακολουθήσαμε τα βήματα, που έχουμε ήδη περιγράψει, δηλαδή τη μετατροπή στο δεκαδικό σύστημα και από εκεί στο οκταδικό: $N_{16} = ABC_{16} \rightarrow N_{10} = 2748_{10} \rightarrow N_8 = 5274_8$

Β' τρόπος

Ένας άλλος τρόπος μετατροπής, είναι να μετατρέψουμε αρχικά το δεκαεξαδικό αριθμό στον αντίστοιχο δυαδικό, αντικαθιστώντας κάθε δεκαεξαδικό ψηφίο με τα αντίστοιχά του 4 δυαδικά. Κατόπιν, ξεκινώντας από τα δεξιά, χωρίζουμε εκ νέου τον αριθμό μας, αυτή τη φορά σε τριάδες ψηφίων. Κάθε τριάδα αντιστοιχεί σε ένα αριθμό του οκταδικού συστήματος.

Για το παραπάνω παράδειγμα έχουμε:

A	B	C
1010	1011	1100

Ο αντίστοιχος δυαδικός αριθμός είναι λοιπόν 1010 1011 1100, τον οποίο χωρίζουμε πλέον σε τριάδες, ξεκινώντας πάντα από τα δεξιά. (αναφερόμαστε σε ακέραιους αριθμούς)

101	010	111	100
5	2	7	4

Άρα ο δεκαεξαδικός αριθμός ABC_{16} , στο οκταδικό σύστημα, είναι ο αριθμός 5274_8 .

Επαφίεται στον χρήστη να αποφασίσει ποιός από τους δύο τρόπους είναι πιο γρήγορος στην υλοποίηση.

1.4. Συμπληρώματα αριθμού βάσης β ως προς βάσεις β-1 και β

Έστω N_β ο αριθμός και $\bar{N}_{\beta-1}$ το συμπλήρωμά του ως προς β-1. Τότε θα έχουμε: $N_\beta = \sum_{i=0}^{n-1} \Psi_i \beta^i$

$$\bar{N}_{\beta-1} = \sum_{i=0}^{n-1} \bar{\Psi}_i \beta^i \quad N_\beta + \bar{N}_{\beta-1} = \sum_{i=0}^{n-1} (\Psi_i + \bar{\Psi}_i) \beta^i = \sum_{i=0}^{n-1} (\beta - 1) \beta^i$$

$$\text{Το συμπλήρωμα ενός ψηφίου ορίζεται ως: } \Psi_i + \bar{\Psi}_i = \beta - 1 \quad (1-14)$$

Επομένως θα έχουμε:

$$N_\beta + \bar{N}_{\beta-1} = (\beta - 1)\beta^0 + (\beta - 1)\beta^1 + \dots + (\beta - 1)\beta^{n-1}$$

$$N_\beta + \bar{N}_{\beta-1} = (\beta^n - 1) \quad \text{και} \quad \bar{N}_{\beta-1} = \beta^n - N_\beta - 1 \quad (1-15)$$

όπου το $\bar{N}_{\beta-1}$ λέγεται το βάσης β-1 συμπλήρωμα του αριθμού N_β .

$$\text{Ο αριθμός λέγεται συμπλήρωμα του αριθμού } N_\beta \text{ ως προς βάση } \beta. \quad \bar{N}_\beta = \bar{N}_{\beta-1} + 1 \quad (1-16)$$

Παράδειγμα 5ο:

Στον πίνακα 1.2 που ακολουθεί παραθέτουμε τυχαίους δεκαδικούς αριθμούς και τα αντίστοιχα συμπληρώματά τους ως προς 9 και 10. Τα συμπληρώματα υπολογίζονται, εφαρμόζοντας τους τύπους (1-15) και (1-16).

Δεκαδικοί αριθμοί N_{10}	Συμπλήρωμα ως προς 9 \bar{N}_9	Συμπλήρωμα ως προς 10 \bar{N}_{10}
135	864	865
23	76	77
118,16	881,83	881,84

Πίνακας 1.2. Συμπληρώματα δεκαδικών αριθμών ως προς 9 και 10 (βάση και βάση-1)

Κατά τον ίδιο τρόπο, στον πίνακα 1.3, που ακολουθεί παραθέτουμε τυχαίους δυαδικούς αριθμούς και τα αντίστοιχα συμπληρώματά τους ως προς 1 και 2. Τα συμπληρώματα υπολογίζονται εφαρμόζοντας και πάλι από τους τύπους (1-15) και (1-16).

Είναι σκόπιμο να αναφέρουμε ότι στο δυαδικό σύστημα, όπως εύκολα αποδεικνύεται, το συμπλήρωμα του ψηφίου 0 είναι το 1 και αντίστροφα. Στη συνέχεια, δίνονται πρακτικοί κανόνες για τον υπολογισμό των συμπληρωμάτων στους δυαδικούς αριθμούς.

Το **συμπλήρωμα ως προς 1** ενός δυαδικού αριθμού με ή χωρίς υποδιαστολή, υπολογίζεται, αλλάζοντας ένα προς ένα όλα τα δυαδικά ψηφία με το αντίστοιχο συμπλήρωμα: τα 0 σε 1 και τα 1 σε 0.

Δυαδικοί αριθμοί N_2	Συμπλήρωμα ως προς 1 \bar{N}_1	Συμπλήρωμα ως προς 2 \bar{N}_2
0101101	1010010	1010011
1111010	0000101	0000110
010,101	101,010	101,011
11010,0011	00101,1100	00101,1101

Πίνακας 1.3. Συμπληρώματα δυαδικών αριθμών ως προς 1 και 2

Το **συμπλήρωμα ως προς 2** ενός δυαδικού αριθμού υπολογίζεται αρχίζοντας από το τέλος του αριθμού και επαναλαμβάνοντας τα δυαδικά ψηφία, όπως έχουν, μέχρις ότου βρεθεί η πρώτη μονάδα, η οποία και αυτή επαναλαμβάνεται. Στη συνέχεια αλλάζουμε όλα τα δυαδικά ψηφία, δηλαδή τα 0 σε 1 και τα 1 σε 0.

Ένας άλλος, ευκολότερος ίσως, τρόπος είναι, προσθέτοντας τη μονάδα στο λιγότερο σημαντικό ψηφίο του αριθμού του συμπληρώματος ως προς ένα που έχουμε υπολογίσει προηγουμένως.

1.5. Αναπαράσταση των θετικών και αρνητικών αριθμών στο δυαδικό σύστημα αρίθμησης

Θα εξετάσουμε τρεις διαφορετικούς τρόπους αναπαράστασης προσημασμένων αριθμών

Την προσημασμένη παράσταση (Sign representation)

Το συμπλήρωμα ως προς 1 (1's complement)

Το συμπλήρωμα ως προς 2 (2's complement)

Πώς μπορούμε να αποφασίσουμε για το ποια αναπαράσταση είναι καλύτερη;

Είναι προφανές ότι η καλύτερη αναπαράσταση θα πρέπει να καταλήγει σε απλούστερες και γρηγορότερες λειτουργίες. Μας ενδιαφέρουν δύο συγκεκριμένες λειτουργίες: Η μετατροπή του x σε $-x$ και η πρόσθεση δύο προσημασμένων αριθμών, ή ο υπολογισμός της γενικής μορφής $x + y$. (Nashelsky, 1994; Κοσσίδης, 1996, Holdsworth, & Woods, 2002; Givone, 2002; Katz, 2005; Wakerly, 2006; Maxfield, 2009; Parhami, 2010; Roth, & Kinney, 2014; Mano, & Ciletti, 2014; Rajaraman, & Adabala, 2015)

1.5.1. Προσημασμένη παράσταση θετικών και αρνητικών δυαδικών αριθμών

Μέχρι στιγμής δεν εξετάσαμε τον τρόπο παράστασης των δυαδικών αριθμών, λαμβάνοντας υπόψη το πρόσημό τους, αλλά υποθέσαμε ότι όλες οι πράξεις αφορούσαν θετικούς αριθμούς. Διάφοροι τρόποι έχουν χρησιμοποιηθεί για την προσημάνση των δυαδικών αριθμών.

Συνήθως χρησιμοποιούμε ένα σύστημα προσημασμένων μεγεθών: Προσθέτουμε το + ή το - μπροστά από ένα μέγεθος για να προσδιορίσουμε το πρόσημό του.

Μπορούμε να το κάνουμε ίδιο και στο δυαδικό σύστημα, προσθέτοντας ένα επιπλέον δυαδικό ψηφίο πρόσημου μπροστά από τους αριθμούς. Συμβατικά λοιπόν το σύμβολο **0** αναπαριστά ένα θετικό αριθμό, και το σύμβολο **1** αναπαριστά έναν αρνητικό αριθμό.

Προσημασμένη παράσταση μεγέθους

Παράδειγμα 6ο:

- $1101_2 = 13_{10}$ (4-bit μη προσημασμένος αριθμός)
- $01101 = +13_{10}$ (5-bit θετικός αριθμός προσημασμένου μεγέθους)
- $11101 = -13_{10}$ (5-bit αρνητικός αριθμός προσημασμένου μεγέθους)
- $0100_2 = 4_{10}$ (4-bit μη προσημασμένος αριθμός)
- $00100 = +4_{10}$ (5-bit θετικός αριθμός προσημασμένου μεγέθους)
- $10100 = -4_{10}$ (5-bit αρνητικός αριθμός προσημασμένου μεγέθους)

Με αυτό τον τρόπο π.χ. το **-5** γράφεται σαν **10101** και το **+5** σαν **00101**. (Υποθέτουμε ότι χρησιμοποιούνται τέσσερα bit για την παράσταση της απόλυτης τιμής του αριθμού).

Παρόλο όμως ότι η παράσταση των προσημασμένων δυαδικών αριθμών με αυτό τον τρόπο ήταν πολύ εύκολη, δεν επικράτησε για δύο λόγους κυρίως α) γιατί έτσι γίνεται σπατάλη δυαδικών ψηφίων και β) για λειτουργικούς λόγους της Αριθμητικής Μονάδας (π.χ. η αφαίρεση δεν ήταν δυνατό να γίνει με τη βοήθεια της πρόσθεσης, αλλά απαιτούσε ξεχωριστές βαθμίδες). Οι περισσότεροι χρησιμοποιούμενες μέθοδοι για την προσημασμένη των δυαδικών αριθμών αναπτύσσονται παρακάτω.

Λειτουργίες προσημασμένων μεγεθών

Η μετατροπή ενός προσημασμένου αριθμού από θετικό σε αρνητικό είναι τριτομμένη: απλά αλλάζουμε το bit πρόσημου από 0 σε 1 και αντίστροφα.

Η πρόσθεση των αριθμών, όμως, είναι δύσκολη. Τα προσημασμένα μεγέθη είναι αυτά που κυρίως χρησιμοποιούμε και βασίζονται στη σύγκριση των προσήμων των δύο προσθετέων:

Αν έχουν το ίδιο πρόσημο, προσθέτουμε τα μεγέθη και κρατάμε το ίδιο πρόσημο. Αν έχουν διαφορετικά πρόσημα, αφαιρούμε το μικρότερο μέγεθος από το μεγαλύτερο. Το πρόσημο του μεγαλύτερου μεγέθους είναι και το πρόσημο του αποτελέσματος.

Αυτή η μέθοδος της αφαίρεσης, μάλλον θα μας οδηγήσει σε πολύπλοκο σχεδιασμό

Αριθμός μετά την πρόσθεση και του δανεικού

	+3	7	9			5	13	17
+	6	4	7			6	4	7
	2	6	8			3	7	9
						2	6	8

διότι
ισχύει →

Στη δεξιά αναπαράσταση και πάνω από τους αριθμούς σημειώνονται οι αριθμοί που προκύπτουν μετά την πρόσθεση και της δανεικής μονάδας (δεκάδας). Έτσι σημειώνεται ο αριθμός 17, ο οποίος εμπεριέχει και μία δανεική δεκάδα από την προηγούμενη βαθμίδα, και αντίστοιχα το 13.

Προσοχή: Το 17 της δεξιάς στήλης, προκύπτει λόγω του δανεισμού από την προηγούμενη δεκάδα, ενώ αρχικά ήταν 7. Στη μεσαία στήλη το αρχικό 14(4+10 δανεικό από την προηγούμενη δεκάδα επίσης) γίνεται 14-1=13, καθώς δάνεισε δεξιάτερα μια δεκάδα. Ομοίως το 5, που προκύπτει, πάνω από το 6 του περισσότερο σημαντικού ψηφίου, είναι απόρροια της πράξης 6-1 (πάλι με το ίδιο σκεπτικό με το προηγούμενο) λόγω του δανεισμού μιας δεκάδας, δεξιάτερα. Άρα οι αριθμοί της πρώτης γραμμής, δεξιά, είναι οι αριθμοί που προκύπτουν, αν αφαιρέσουμε και το τυχόν δανεικό που μας χρειάζεται, για να υλοποιηθεί η αφαίρεση.

1.5.2. Παράσταση θετικών και αρνητικών δυαδικών αριθμών με συμπλήρωμα ως προς 1

Μια διαφορετική προσέγγιση είναι το συμπλήρωμα ως προς ένα κατά την οποία μετατρέπουμε έναν αριθμό από θετικό σε αρνητικό, συμπληρώνοντας κάθε δυαδικό ψηφίο του αριθμού (αλλάζοντας το 0 σε 1 ή το 1 σε 0 αντίστοιχα).

Σε ότι αφορά στο πρόσημο, κρατάμε τα δυαδικά ψηφία με την ίδια αναπαράσταση, όπως και προηγουμένως. Έτσι το 0 αντιστοιχεί σε θετικούς, και το 1 σε αρνητικούς αριθμούς. Το δυαδικό ψηφίο προσήμου συμπληρώνεται μαζί με τα υπόλοιπα δυαδικά ψηφία. Το ερώτημα που τίθεται είναι γιατί ονομάζεται «συμπλήρωμα ως προς ένα».

Το συμπλήρωμα ενός bit είναι ισοδύναμο, με την αφαίρεσή του από το 1. Έτσι $\bar{0} = 1$ δηλαδή $1 - 0 = 1$ και $\bar{1} = 0$ δηλαδή $1 - 1 = 0$

Ομοίως, συμπληρώνοντας κάθε bit ενός n-bit αριθμού, είναι σαν να αφαιρούμε τον αριθμό αυτόν από τον αριθμό $2^n - 1$. Για παράδειγμα, έστω ότι θέλουμε να βρούμε τον αρνητικό του πενταψηφίου αριθμού 01101. Εδώ $n=5$, και $2^5 - 1 = 31_{10} = 11111_2$.

Αφαιρώντας το 01101 από το 11111 έχουμε σαν αποτέλεσμα 10010:

$$\begin{array}{r} \\ - \\ \hline \end{array}$$

Παράδειγμα 7ο:

A)

- 1101₂ = 13₁₀ (4-bit μη προσημασμένος αριθμός)
- 01101 = +13₁₀ (5-bit θετικός αριθμός με συμπλήρωμα ως προς ένα)
- 10010 = -13₁₀ (5-bit αρνητικός αριθμός με συμπλήρωμα ως προς ένα)
- 0100₂ = 4₁₀ (4-bit μη προσημασμένος αριθμός)
- 00100 = +4₁₀ (5-bit θετικός αριθμός με συμπλήρωμα ως προς ένα)
- 11011 = -4₁₀ (5-bit αρνητικός αριθμός με συμπλήρωμα ως προς ένα)

B)

Στον πίνακα 1.4 παραθέτουμε μερικούς προσημασμένους δυαδικούς αριθμούς (πρώτη στήλη) μαζί με τους αντίστοιχους δεκαδικούς αριθμούς που αντιπροσωπεύουν (δεύτερη στήλη), σύμφωνα με τους κανόνες συμπλήρωσης ως προς ένα.

Εάν υποθέσουμε ότι έχουμε δυαδικές παραστάσεις με 6 δυαδικά ψηφία, ο μεγαλύτερος θετικός αριθμός που δύναται να γραφεί είναι ο αριθμός 011111 (+31) και ο μικρότερος αρνητικός είναι ο 100000 (-31). Δηλαδή μπορούμε να παραστήσουμε τους αριθμούς από -31 έως +31 με 6 δυαδικά ψηφία. Εύκολα συνάγεται ότι, αν έχουμε αριθμούς αποτελούμενους από «n» δυαδικά ψηφία, ο μεγαλύτερος θετικός που μπορεί να περιγραφεί μέσω αυτών είναι ο $2^{n-1} - 1$ ενώ ο μικρότερος αρνητικός είναι ο $-(2^{n-1} - 1)$.

Προσημασμένες δυαδικές παραστάσεις με συμπλήρωμα ως προς 1	Αντίστοιχοι δεκαδικοί αριθμοί
01010111	+87
00000000	+0
11111111	-0
0111111111	+511
1000000000	-511
00010001	+17

Πίνακας 1.4. Προσημασμένοι δυαδικοί αριθμοί με συμπλήρωμα ως προς 1

1.5.3. Παράσταση θετικών και αρνητικών δυαδικών αριθμών με συμπλήρωμα ως προς 2

Συχνά μιλάμε για το «συμπλήρωμα ως προς δύο» ενός αριθμού. Αυτή είναι μια «μπερδεμένη φράση», αλλά συνήθως συσχετίζεται με την αλλαγή του προσήμου κάποιου αριθμού.

Όπως και στην παράσταση με συμπλήρωμα ως προς 1 έτσι και εδώ, εάν το πρώτο δυαδικό ψηφίο του αριθμού είναι «1» ο αριθμός είναι αρνητικός, ενώ εάν είναι «0» ο αριθμός είναι θετικός, με τη διαφορά ότι στην αναπαράσταση αυτή και το πρόσημο παίζει ρόλο στην αξία του αριθμού.

Για να μετατρέψουμε έναν αριθμό σε αρνητικό, συμπληρώνουμε κάθε δυαδικό ψηφίο (όπως στο συμπλήρωμα ως προς ένα) κι έπειτα προσθέτουμε 1 στο λιγότερο σημαντικό ψηφίο.

Στον πίνακα 1.5 παραθέτουμε ορισμένα παραδείγματα δυαδικών αριθμών (πρώτη στήλη) μαζί τον αντίστοιχο δεκαδικό αριθμό που παριστάνουν (δεύτερη στήλη).

Δύο άλλοι, ισοδύναμοι τρόποι για να μετατρέψουμε αριθμούς, στο συμπλήρωμα ως προς δύο είναι:

α) Να αφαιρέσουμε έναν αριθμό των n-bit από τον αριθμό 2^n , π.χ.

$$\begin{array}{r}
 1\ 0\ 0\ 0\ 0\ 0 \\
 -\ 0\ 1\ 1\ 0\ 1\ (+13_{10}) \\
 \hline
 1\ 0\ 0\ 1\ 1\ (-13_{10})
 \end{array}
 \qquad
 \begin{array}{r}
 1\ 0\ 0\ 0\ 0\ 0 \\
 -\ 0\ 0\ 1\ 0\ 0\ (+4_{10}) \\
 \hline
 1\ 1\ 1\ 0\ 0\ (-4_{10})
 \end{array}$$

β) Να συμπληρώσουμε όλα τα bits που βρίσκονται αριστερά από τον δεξιότερο 1.

01101 = +13₁₀ (ένας θετικός αριθμός σε συμπλήρωμα ως προς δύο)

10011 = - 13₁₀ (ένας αρνητικός αριθμός σε συμπλήρωμα ως προς δύο)

00100 = + 4₁₀ (ένας θετικός αριθμός σε συμπλήρωμα ως προς δύο)

11100 = - 4₁₀ (ένας αρνητικός αριθμός σε συμπλήρωμα ως προς δύο)

Παράδειγμα 8ο:

A)

1101₂ = 13₁₀ (4-bit μη προσημασμένος αριθμός)

01101 = +13₁₀ (5-bit θετικός αριθμός με συμπλήρωμα ως προς δύο)

10010 = - 13₁₀ (5-bit αρνητικός αριθμός με συμπλήρωμα ως προς ένα)

10011 = - 13₁₀ (5-bit αρνητικός αριθμός με συμπλήρωμα ως προς δύο)

0100 = 4₁₀ (4-bit μη προσημασμένος αριθμός)

00100 = + 4₁₀ (5-bit θετικός αριθμός με συμπλήρωμα ως προς δύο)

11011 = - 4₁₀ (5-bit αρνητικός αριθμός με συμπλήρωμα ως προς ένα)

11100 = - 4₁₀ (5-bit αρνητικός αριθμός με συμπλήρωμα ως προς δύο)

B)

Προσημασμένες δυαδικές παραστάσεις με συμπλήρωμα ως προς 2	Αντίστοιχοι δεκαδικοί αριθμοί
01001100	+76
01111111	+127
10000000	-128
11111111	-1
00000000	+0
10101010	-86

Πίνακας 1.5. Προσημασμένοι δυαδικοί αριθμοί με συμπλήρωμα ως προς 2

Εάν υποθέσουμε και εδώ ότι έχουμε δυαδικές παραστάσεις με 6 δυαδικά ψηφία, ο μεγαλύτερος θετικός αριθμός που δύναται να γραφεί είναι ο 011111 (+31) ενώ ο μικρότερος αρνητικός είναι ο 100000 (-32). Δηλαδή με 6 δυαδικά ψηφία μπορούμε να παραστήσουμε τους αριθμούς από -32 έως +31.

Για «n» δυαδικά ψηφία ο μεγαλύτερος θετικός θα είναι ο $2^{n-1}-1$ και ο μικρότερος αρνητικός θα είναι ο -2^{n-1} .

1.5.4. Σύγκριση των συστημάτων προσημασμένων αριθμών

Στον πίνακα 1.6 παρατίθενται όλοι οι αριθμοί(θετικοί και αρνητικοί) των 4-bit στα διαφορετικά συστήματα αναπαράστασης. Παρατηρούμε ότι οι θετικοί αριθμοί είναι οι ίδιοι και στις τρεις αναπαράστασεις.

Στο προσημασμένο μέγεθος και στο συμπλήρωμα ως προς ένα υπάρχουν δύο τρόποι αναπαράστασης του 0. Αυτό κάνει τα πράγματα πιο πολύπλοκα. Στο συμπλήρωμα ως προς δύο έχουμε ασύμμετρο εύρος αριθμών, καθώς υπάρχει ένας παραπάνω αρνητικός αριθμός από ότι θετικός. Μπορούμε να αναπαράστησουμε το -8 αλλά όχι το +8. Εντούτοις, το συμπλήρωμα ως προς δύο προτιμάται διότι έχει μόνο ένα τρόπο αναπαράστασης του 0, και ο αλγόριθμος της πρόσθεσης είναι απλούστερος.

Δεκαδικός	Παράσταση προσήμου	Συμπλήρωμα ως προς 1	Συμπλήρωμα ως προς 2
7	0111	0111	0111
6	0110	0110	0110
5	0101	0101	0101
4	0100	0100	0100
3	0011	0011	0011
2	0010	0010	0010
1	0001	0001	0001
0	0000	0000	0000
-0	1000	1111	—
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	—	—	1000

Πίνακας 1.6. Αντιστοίχιση δεκαδικών αριθμών στις δυνατές αναπαραστάσεις δυαδικών αριθμών

Εύρος των προσημασμένων συστημάτων των αριθμών

Πόσοι αρνητικοί και θετικοί αριθμοί μπορούν να αναπαρασταθούν σε καθένα από τα συστήματα αναπαράστασης του πίνακα 1.6;

	Χωρίς Πρόσημο	Μεγέθη με πρόσημο	Συμπλήρωμα ως προς 1	Συμπλήρωμα ως προς 2
Ελάχιστος	0000 (0)	1111 (-7)	1000 (-7)	1000 (-8)
Μέγιστος	1111 (15)	0111 (+7)	0111 (+7)	0111 (+7)

Έτσι και με βάση τα αποτελέσματα των παραγράφων 1.5.2, και 1.5.3 που ήδη μελετήσαμε, σε αριθμούς των n-bit συμπεριλαμβανομένου και του προσήμου, το πλήθος είναι:

	Χωρίς Πρόσημο	Μεγέθη με πρόσημο	Συμπλήρωμα ως προς 1	Συμπλήρωμα ως προς 2
Ελάχιστος	0	$-(2^{n-1}-1)$	$-(2^{n-1}-1)$	-2^{n-1}
Μέγιστος	2^n-1	$+(2^{n-1}-1)$	$+(2^{n-1}-1)$	$+(2^{n-1}-1)$

1.5.5. Λειτουργικότητα της μεθόδου

Για αριθμούς των n-bit, η μετατροπή π.χ. του B σε συμπλήρωμα ως προς δύο δίνει τον αριθμό $2^n - B$ (αυτός είναι ένας από τους εναλλακτικούς τρόπους μετατροπής σε συμπλήρωμα ως προς δύο).

$$A - B = A + (-B) = A + (2^n - B) = (A - B) + 2^n$$

Αν $A \geq B$, τότε ο $(A - B)$ είναι θετικός αριθμός, και το 2^n αναπαριστά το κρατούμενο (1). Αγνοώντας το κρατούμενο αυτό, είναι ισοδύναμο με το αν αφαιρέσουμε το 2^n και μας δίνει το επιθυμητό αποτέλεσμα $(A - B)$.

Αν $A < B$, τότε ο $(A - B)$ είναι αρνητικός αριθμός και έχουμε $2^n - (A - B)$. Αυτό αντιστοιχεί στο επιθυμητό αποτέλεσμα, $-(A - B)$, σε μορφή συμπληρώματος ως προς δύο.

Παράδειγμα 9ο:

Α) Μετατροπή προσημασμένου αριθμού στο δεκαδικό

Να μετατρέψετε τον δυαδικό 110101 στο δεκαδικό, θεωρώντας ότι ο αριθμός βρίσκεται σε μια από τις παρακάτω μορφές: (με τη χρήση 6 δυαδικών ψηφίων).

(a) Μορφή προσημασμένων μεγεθών

- (b) Συμπλήρωμα ως προς ένα
- (c) Συμπλήρωμα ως προς δύο

Λύση

Καθώς το bit πρόσημου είναι 1, έχουμε αρνητικό αριθμό. Η μετατροπή μπορεί να γίνει με τους τρεις διαφορετικούς τρόπους, που προφανώς θα μας δώσουν διαφορετικά αποτελέσματα!!

(a) Μορφή προσημασμένων μεγεθών

Ο αρχικός δυαδικός αριθμός, 110101, αφαιρώντας το πρώτο δυαδικό ψηφίο που παριστά το πρόσημο, μάς δίνει τον αριθμό 10101, ο οποίος είναι ο δεκαδικός +21. Άρα ο αριθμός 110101 αντιπροσωπεύει το -2110.

(b) Συμπλήρωμα ως προς ένα

Η μετατροπή του 110101 σε συμπλήρωμα ως προς ένα δίνει 001010 = +1010. Έτσι ο αρχικός αριθμός είναι το -1010.

(c) Συμπλήρωμα ως προς δύο

Η μετατροπή του 110101 σε συμπλήρωμα ως προς δύο δίνει 001011 = 1110, και συνεπώς ο αρχικός αριθμός είναι το -1110.

Το πιο σημαντικό στοιχείο είναι πως ο δυαδικός αριθμός έχει διαφορετικό νόημα ανάλογα με τον τρόπο αναπαράστασής του.

1.6. Πράξεις στο δυαδικό σύστημα (με τα ψηφία 0 και 1)

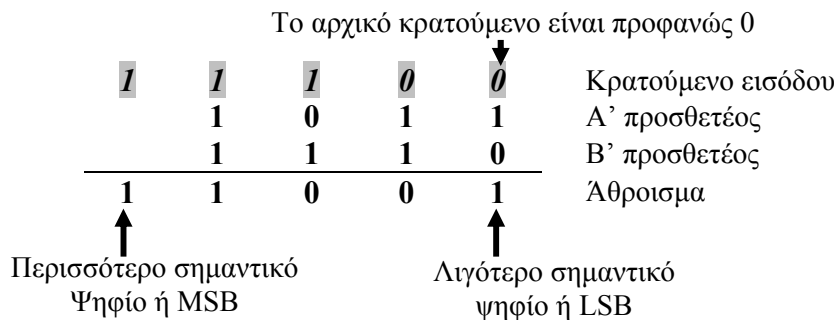
Οι αριθμητικές πράξεις, και συνεπώς η αριθμητική, είναι οι πιο βασικές λειτουργίες που εκτελεί ένας υπολογιστής, αλλά δεν είναι τόσο εύκολες, όσο πιστεύετε. Οι πράξεις στο δυαδικό σύστημα αρίθμησης εκτελούνται ακριβώς όπως και στο δεκαδικό. Είναι γνωστό βέβαια ότι ο υπολογιστής εκτελεί μόνο μία πράξη, την πρόσθεση, και ανάγει όλες τις άλλες σε αυτή.

Θα αναλύσουμε τις τέσσερις πράξεις, δηλαδή την πρόσθεση, την αφαίρεση, τον πολλαπλασιασμό και τη διαίρεση, με βάση την εμπειρία μας. Όλες οι πράξεις στον υπολογιστή εκτελούνται στην αριθμητική και λογική μονάδα (A.L.M./A.L.U.), που είναι η καρδιά της Κεντρικής Μονάδας Επεξεργασίας (K.M.E./C.P.U.), μόνο όσο αφορά στην αριθμητική αυτών και όχι στον τρόπο πραγματοποίησης των πράξεων με λογικά κυκλώματα, θέμα το οποίο θα εξετάσουμε σε επόμενο κεφάλαιο. (Φραγκάκης, 1975; Nashelsky, 1994; Κοσσιδάς, 1996; Balch, 2003; Givone, 2002; Wakerly, 2006; Balabanian, & Carlson, 2007; Tocci, Widmer, & Moss, 2010; Floyd, 2013)

1.6.1. Πρόσθεση δυαδικών αριθμών

Στο δεκαδικό σύστημα αρίθμησης, εάν από την πρόσθεση δύο ψηφίων ίδιας τάξης φθάσουμε ή υπερβούμε τη βάση του συστήματος, μεταφέρουμε την πλεονάζουσα δεκάδα στο αμέσως μεγαλύτερης τάξης ψηφίο κ.ο.κ.

Το ίδιο συμβαίνει και με το δυαδικό σύστημα όπου, όταν το άθροισμα φθάσει ή υπερβεί τον αριθμό 2 (βάση του συστήματος), μεταφέρουμε μία δυάδα στο αμέσως υψηλότερης τάξης ψηφίο.



Προσθετέος X	Προσθετέος Y	Άθροισμα Σ	Κρατούμενο K
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Πίνακας 1.7. Πρόσθεση δυαδικών ψηφίων

Μπορούμε να προσθέσουμε δύο δυαδικούς αριθμούς, κατά στήλες, αρχίζοντας από τα δεξιά, όπως προσθέτουμε δύο δεκαδικούς αριθμούς. Πρέπει όμως να θυμόμαστε πως είναι δυαδικοί αριθμοί. Για παράδειγμα, $1+1 = 10$ γράφουμε δηλαδή το μηδέν και έχουμε ένα κρατούμενο! (προφανώς ισχύει $0 + 1 = 1$)

Στον πίνακα 1.7 φαίνονται τα αθροίσματα (Σ) και τα κρατούμενα (Κ) όλων των δυνατών συνδυασμών δύο μονοψηφίων δυαδικών αριθμών X και Y.

Το αποτέλεσμα (το 1) που προκύπτει σαν κρατούμενο στην τελευταία πρόσθεση του πίνακα 1.7 προέρχεται από το ότι η πρόσθεση δύο άσων(1) στο δυαδικό μας δίνει 10_2 (2_{10}).

Καλό είναι να λάβουμε υπόψη μας ότι σε περίπτωση που προσθέτουμε περισσότερους από τρεις δυαδικούς αριθμούς, χρειάζεται ιδιαίτερη προσοχή, γιατί η προς μεταφορά δυάδα είναι δυνατόν να μετατοπισθεί περισσότερες από μία θέσεις προς τα αριστερά. Αυτό μπορεί να συμβεί και στον πολλαπλασιασμό όπου εκεί προσθέτουμε πολλά μερικά γινόμενα.

1.6.2. Αφαίρεση δυαδικών αριθμών

Θα μελετήσουμε τώρα την αφαίρεση και τους αρνητικούς αριθμούς. Η αριθμητική που υλοποιήσαμε μέχρι τώρα περιορίστηκε κυρίως σε μη προσημασμένους (θετικούς) ακεραίους.

Το κύριο πρόβλημα είναι η αναπαράσταση των αρνητικών αριθμών στο δυαδικό σύστημα. Θα εισαγάγουμε τρεις μεθόδους, και θα δείξουμε γιατί μια από αυτές είναι η καλύτερη.

Με τη χρήση αρνητικών αριθμών, έχουμε τη δυνατότητα να εκτελέσουμε την αφαίρεση χρησιμοποιώντας τους αθροιστές, καθώς $A - B = A + (-B)$.

Και η αφαίρεση δυαδικών αριθμών γίνεται όπως ακριβώς και στους δεκαδικούς. Βασική προϋπόθεση της αφαίρεσης είναι η λήψη μιας δανεικής (βοήθω) μονάδας (με αξία δύο καθώς έχουμε δυαδικό σύστημα) από το μεγαλύτερης τάξης ψηφίο, στην περίπτωση όπου το ψηφίο του μειωτέου είναι μικρότερο του ψηφίου του αφαιρετέου. Στον πίνακα 1.8 φαίνονται η διαφορά (Δ) και το δανεικό (B) όλων των δυνατών συνδυασμών δύο μονοψηφίων δυαδικών αριθμών X και Y.

Μειωτέος X	Αφαιρετέος Y	Διαφορά Δ	Δανεικό B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Πίνακας 1.8. Αφαίρεση δυαδικών ψηφίων

Παράδειγμα 10ο:

1010101	Μειωτέος	→	85
<u>-1000111</u>	Αφαιρετέος	→	<u>-71</u>
0001110	Διαφορά	→	14

Παρατήρηση: Αντί για αφαίρεση θα μπορούσαμε να κάνουμε πρόσθεση του συμπληρώματος του αρνητικού αριθμού.

1.6.3. Πολλαπλασιασμός δυαδικών αριθμών

Ο πολλαπλασιασμός δεν είναι τόσο δύσκολος σαν αλγοριθμική διαδικασία. Γίνεται μάλιστα πολύ απλός αν τον δούμε σαν να είναι απλά επαναλαμβανόμενες προσθέσεις. Αν έχουμε αθροιστές, μπορούμε εύκολα να εκτελέσουμε και τον πολλαπλασιασμό.

Υπενθυμίζουμε ότι η λειτουργία της λογικής πράξης AND, είναι ισοδύναμη με τον πολλαπλασιασμό δύο ψηφίων. Στο δεκαδικό, ένας εύκολος τρόπος να πολλαπλασιάσουμε με το 10 είναι να ολισθήσουμε όλα τα ψηφία προς τα αριστερά κατά μια θέση και να βάλουμε ένα 0 στο δεξιό άκρο, Έτσι $128 \times 10 = 1280$

Μπορούμε να κάνουμε το ίδιο και στο δυαδικό. Η ολίσθηση προς τα αριστερά ισοδυναμεί με πολλαπλασιασμό επί 2: $11 \times 10 = 110$ (στο δεκαδικό, $3 \times 2 = 6$)

Η ολίσθηση προς τα αριστερά δύο φορές ισοδυναμεί με πολλαπλασιασμό επί 4:

$$11 \times 100 = 1100 \text{ (στο δεκαδικό, } 3 \times 4 = 12)$$

Έτσι, ένας δυαδικός αριθμός που πολλαπλασιάζεται κατά 2^n ολισθαίνει n θέσεις προς τα αριστερά. Άρα, η ολίσθηση προς τα δεξιά ισοδυναμεί με διαίρεση δια του 2. (θα μελετηθεί παρακάτω)

$$110 \div 10 = 11 \text{ (στο δεκαδικό, } 6 \div 2 = 3)$$

a	b	a*b
0	0	0
0	1	0
1	0	0
1	1	1

Πίνακας 1.9. Η πράξη του πολλαπλασιασμού

Πολλαπλασιασμός μη προσημασμένων θετικών δυαδικών

Θα αναλύσουμε όλες τις δυνατές περιπτώσεις πολλαπλασιασμών δηλαδή μεταξύ θετικών αριθμών, αρνητικού πολλαπλασιαστή και αρνητικού πολλαπλασιαστέου. Θα ξεκινήσουμε με την πιο απλή περίπτωση του πολλαπλασιασμού μη προσημασμένων αριθμών. Ακολουθεί ένα παράδειγμα πολλαπλασιασμού θετικών μη προσημασμένων αριθμών.

				1	1	0	1	Πολλαπλασιαστέος
		x		0	1	1	0	Πολλαπλασιαστής
				0	0	0	0	Μερικό γινόμενο
			1	1	0	1		Μερικό γινόμενο
			1	0	1			Μερικό γινόμενο
+	0	0	0	0				Μερικό γινόμενο
	1	0	0	1	1	1	0	Γινόμενο

Δεδομένου ότι στο δυαδικό πολλαπλασιάζουμε πάντα με 0 ή 1, τα μερικά γινόμενα είναι πάντα είτε 0000 είτε ο πολλαπλασιαστέος (1101 στο παράδειγμα μας).

Ο πολλαπλασιασμός δύο ακεραίων αριθμών των n -ψηφίων δίνει αποτέλεσμα το οποίο μπορεί να έχει μήκος ψηφίων μέχρι $2n$. (το άθροισμα των ψηφίων των αρχικών αριθμών)

Υπάρχουν τέσσερα μερικά γινόμενα τα οποία προστίθενται για να μας δώσουν το αποτέλεσμα. Ακόμα κι αν το αποτέλεσμα έχει μέχρι 8 ψηφία, μπορούμε να χρησιμοποιήσουμε αθροιστές των 4-ψηφίων. Το δομικό διάγραμμα του σειριακού κυκλώματος δυαδικού πολλαπλασιασμού θα δοθεί σε επόμενο κεφάλαιο.

1.6.3.1 Πολλαπλασιασμός θετικών δυαδικών αριθμών

Η εκτέλεση του πολλαπλασιασμού στηρίζεται βασικά σε προσθέσεις μερικών γινομένων, που προκύπτουν από το γινόμενο κάθε ψηφίου του πολλαπλασιαστή με τον πολλαπλασιαστέο. Το αποτέλεσμα αυτών των μερικών γινομένων θα είναι το μηδέν ή ο ίδιος ο πολλαπλασιαστέος, ανάλογα με την τιμή των ψηφίων του πολλαπλασιαστή (0 ή 1). (Holdsworth, & Woods, 2002; Balch, 2003; Predko, 2005; Balabanian, & Carlson, 2007; Tocci, et al., 2010; Parhami, 2010; Roth, & Kinney, 2014)

Παράδειγμα 11ο:

Ακολουθεί η διαδικασία ενός πολλαπλασιασμού στο δυαδικό και στο δεκαδικό σύστημα

1010111	Πολλαπλασιαστέος	87
x 1011	Πολλαπλασιαστής	x11
1010111		87
1010111	Μερικά γινόμενα	87
0000000		957
1010111		
1110111101	← Γινόμενο	

Συνεπώς ο πολλαπλασιασμός στηρίζεται σε συνεχείς ολισθήσεις μέχρι να εξαντληθούν όλα τα ψηφία του πολλαπλασιαστή.

1.6.3.2 Πολλαπλασιασμός προσημασμένων δυαδικών αριθμών

Επαναλαμβάνουμε την πράξη του πολλαπλασιασμού, θεωρώντας ότι ο πρώτος αριθμός είναι αρνητικός (το πρώτο ψηφίο του είναι 1). Εκτελούμε τις πράξεις όπως και προηγουμένως. Φυσικά το αποτέλεσμα είναι το ίδιο.

				1	1	0	1	<i>Πολλαπλασιαστέος</i>
	x			0	1	1	0	<i>Πολλαπλασιαστής</i>
				0	0	0	0	<i>Μερικά γινόμενα</i>
		1		1	0	1		
		1	1	0	1			
+	0	0	0	0				
	1	0	0	1	1	1	0	<i>Γινόμενο</i>

Αν λάβουμε υπόψη μας ότι οι αριθμοί μας είναι προσημασμένοι, τότε ο πρώτος αριθμός (αρνητικός) είναι ο -3 και ο δεύτερος (θετικός) είναι ο 6. Φυσικά το αποτέλεσμα είναι λάθος, καθώς το γινόμενό τους δεν είναι -50! (Το συμπλήρωμα ως προς 2 του αποτελέσματος). Η παρατήρηση αυτή μας οδηγεί στο συμπέρασμα ότι πρέπει να ακολουθήσουμε διαφορετικό αλγόριθμο από τον τετριμμένο, στην περίπτωση στην οποία κάποιος από τους αριθμούς μας (πολλαπλασιαστής ή πολλαπλασιαστέος) είναι αρνητικός.

Είναι γνωστό ότι κάθε ψηφίο του πολλαπλασιαστή παράγει μια μετατοπισμένη έκφραση του πολλαπλασιαστέου, και αυτό αντιστοιχεί στο μερικό γινόμενο για αυτό το ψηφίο.

Στο δυαδικό πολλαπλασιασμό, ένας πολλαπλασιαστέος των 4-bit 1101 αποθηκεύεται σε μια λέξη των 8-bit σαν 00001101. Κάθε μερικό γινόμενο είναι ο αριθμός αυτός, μετατοπισμένος προς τα αριστερά, και συμπληρωμένος με τα αντίστοιχα μηδενικά από δεξιά, π.χ. με διπλή μετατόπιση ο 00110100.

Η παραπάνω διαδικασία δεν ισχύει, εάν ο πολλαπλασιαστέος/στής είναι αρνητικός. Το πρόβλημα είναι ότι κάθε μερικό γινόμενο του αρνητικού πολλαπλασιαστέου πρέπει να είναι αρνητικό σε ένα πεδίο 2n-bit. Έτσι δεν υπάρχει χώρος για το ψηφίο του προσήμου. (Parhami, 2010; Rajaraman, & Adabla, 2015)

1.6.3.3 Πολλαπλασιασμός προσημασμένων δυαδικών (αρνητικός πολλαπλασιαστέος)

Έστω λοιπόν ότι οι αριθμοί μας είναι μη προσημασμένοι και θέλουμε να πολλαπλασιάσουμε το θετικό 9 με το θετικό 3. Τότε προφανώς το αποτέλεσμα θα είναι 27.

				1	0	0	1	(9) Πολλαπλασιαστέος
			x	0	0	1	1	(3) Πολλαπλασιαστής
				0	1	0	0	<i>Μερικά γινόμενα</i>
	0	0	0	0	1	0	0	0
	0	0	0	1	0	0	1	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	1	1	0	1	(27) Γινόμενο

Τα δύο τελευταία μερικά γινόμενα είναι σαφώς πλεονάζοντα, καθώς το αποτέλεσμα είναι μηδέν. Αρχικά θα εξετάσουμε την περίπτωση που ο πολλαπλασιαστέος είναι αρνητικός αριθμός. Έστω τώρα ότι οι ίδιοι αριθμοί είναι ακέραιοι προσημασμένοι αριθμοί με συμπλήρωμα ως προς 2. Τότε ο πρώτος όρος αντιστοιχεί στον αρνητικό -7 και ο δεύτερος αντιστοιχεί στο θετικό αριθμό 3.

				1	0	0	1	(-7) Πολλαπλασιαστέος
			x	0	0	1	1	(3) Πολλαπλασιαστής
				1	1	0	0	<i>Μερικά γινόμενα</i>
	1	1	1	1	0	0	1	1
	1	1	1	0	0	1	0	0
	1	1	0	1	0	1	1	(-21) Γινόμενο

Στο υπόδειγμα παρατηρούμε ότι αντί να προσθέσουμε αριστερά του αποτελέσματος μηδενικά προσθέτουμε άσσους, μέχρι το συνολικό να γίνει ίσο με το άθροισμα των μηκών των δύο όρων του πολλαπλασιασμού (εδώ 2n). Είναι προφανές ότι παίρνουμε το συμπλήρωμα ως προς 2 του αποτελέσματος για να βρούμε

τον αντίστοιχο θετικό δεκαδικό αριθμό στον οποίο απλά προσθέτουμε το αρνητικό πρόσημο. (Θα αλλάζε το αποτέλεσμα αν βάζαμε εμπρός αντί τρεις, μόνο δύο άσους στο πρώτο μερικό γινόμενο και αντίστοιχα ένα στο δεύτερο.)

1.6.3.4 Πολλαπλασιασμός προσημασμένων δυαδικών (αρνητικός πολλαπλασιαστής)

Στην περίπτωση στην οποία ο πολλαπλασιαστής είναι αρνητικός, ο γνωστός αλγόριθμος του πολλαπλασιασμού δε μας δίνει σωστά αποτελέσματα, καθώς τα ψηφία του πολλαπλασιαστή δεν αντιστοιχούν πλέον στις ολισθήσεις ή στους πολλαπλασιασμούς που πρέπει να εκτελεστούν.

Υπάρχουν διάφοροι τρόποι επίλυσης του προβλήματος αυτού:

α) Να μετατρέψουμε τόσο τον πολλαπλασιαστή όσο και τον πολλαπλασιαστέο σε θετικούς αριθμούς, να εκτελέσουμε τον πολλαπλασιασμό και κατόπιν να πάρουμε το συμπλήρωμα ως προς δύο του αποτελέσματος, μόνο εάν διαφέρει το πρόσημο των δύο αρχικών αριθμών.

β) Να χρησιμοποιήσουμε τον αλγόριθμο του Booth, ο οποίος είναι βασικός αλγόριθμος της αρχιτεκτονικής υπολογιστών. (Βλέπε παράρτημα) (Parhami, 2010; Roth, & Kinney, 2014)

Ακολουθεί ένα παράδειγμα πολλαπλασιασμού με τον πολλαπλασιαστή αρνητικό, όπου παρουσιάζεται ένας άλλος τρόπος υλοποίησης. Οι αριθμοί μας είναι προσημασμένοι με το δεύτερο αρνητικό (η αναπαράστασή τους γίνεται με πενταψηφίους αριθμούς).

$$\begin{array}{r}
 00110 \quad (6) \\
 \times 11010 \quad (-6) \\
 \hline
 00110 \quad 0 \\
 00110 \quad 0 \\
 00110 \quad 0 \\
 00110 \quad 0 \\
 \hline
 111011100 \quad (-36) \\
 000100100 \quad (+36)
 \end{array}$$

Είναι προφανές ότι όταν κάνουμε την πράξη με το τελευταίο ψηφίο του πολλαπλασιαστή δε γράφουμε στα μερικά γινόμενα τον πολλαπλασιαστέο, αλλά το συμπλήρωμά του ως προς 2 και κατόπιν εκτελούμε κανονικά την πρόσθεση. Παίρνουμε το συμπλήρωμα του αριθμού, που προκύπτει, και ελέγχουμε την ορθότητα του αποτελέσματος.

1.6.4. Διαίρεση δυαδικών αριθμών

Η διαίρεση δυαδικών αριθμών είναι πολύπλοκότερη του πολλαπλασιασμού, αλλά βασίζεται στις ίδιες αρχές. Περιλαμβάνει επαναλαμβανόμενες ολισθήσεις και πρόσθεση ή αφαίρεση. Παρουσιάζουμε ένα παράδειγμα, το οποίο είναι ουσιαστικά αυτό που γνωρίζουμε από τη διαίρεση με δεκαδικούς αριθμούς.

$$\begin{array}{r}
 10010011 \quad | \quad 1011 \quad \leftarrow \text{Διαιρέτης} \\
 \underline{1011} \quad \downarrow \\
 001110 \quad \\
 \underline{1011} \\
 001111 \\
 \underline{1011} \\
 1000 \quad \leftarrow \text{Υπόλοιπο}
 \end{array}$$

Η περιγραφή είναι εύκολη, καθώς ακολουθούμε ακριβώς ό,τι ξέρουμε για τους δεκαδικούς αριθμούς. Έτσι αρχικά διαπιστώνουμε ότι ο 1011 > 1001 και, συνεπώς, κατεβάζουμε και το επόμενο δυαδικό ψηφίο, οπότε έχουμε 10010, και αφαιρούμε το διαιρέτη(1011). Μετά τη γραμμή γράφουμε το αποτέλεσμα της αφαίρεσης κ.ο.κ. (Predko, 2005; Parhami, 2010; Roth, & Kinney, 2014; Rajaraman, & Adabala, 2015)

1.6.4.1 Αλγόριθμοι Διαίρεσης

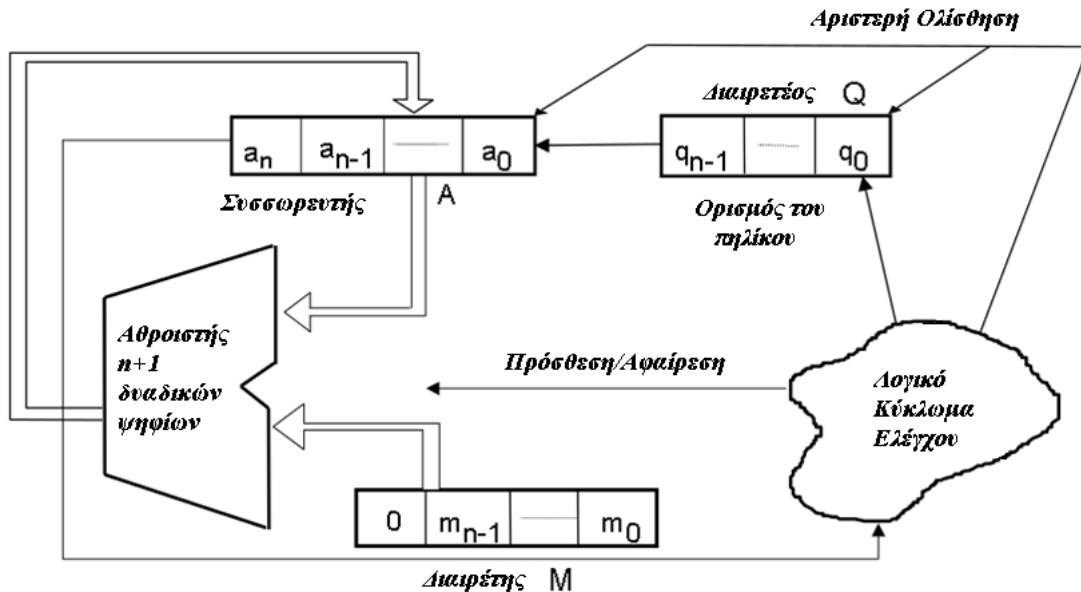
Η συνήθης διαδικασία που ακολουθούμε για τον προσδιορισμό των ψηφίων του πηλίκου στα παραδείγματα των διαιρέσεων, που κάνουμε με το χέρι, είναι πολύ δύσκολο να την αυτοματοποιήσουμε σε ένα λογικό κύκλωμα.

Για θετικούς αριθμούς μπορούμε να χρησιμοποιήσουμε δύο τρόπους:

- Τη διαίρεση με ανάκτηση (*Restoring*) και
- Τη διαίρεση χωρίς ανάκτηση (*Non-restoring*)

Διαίρεση με ανάκτηση

Αν και θεωρείται δύσκολη η κατανόηση του διαγράμματος που ακολουθεί αυτό παρατίθεται αρχικά μόνο για λόγους πληρότητας.



Σχήμα 1.1. Σχηματικό διάγραμμα του αλγόριθμου διαίρεσης με ανάκτηση

Ο διαιρέτης τοποθετείται κάτω από το διαιρετέο (η τετριμμένη διαδικασία). Κατόπιν κάνουμε την αφαίρεση. Εάν το υπόλοιπο είναι μηδέν ή θετικό, τότε θέτουμε 1 στο ψηφίο του πηλίκου. Κατόπιν, κατεβάζοντας το επόμενο ψηφίο του διαιρετέου και επανατοποθετώντας τον διαιρέτη, εκτελούμε νέα αφαίρεση κ.ο.κ.

Εάν το υπόλοιπο είναι αρνητικό, τότε θέτουμε στα ψηφία του πηλίκου το ψηφίο 0, και ανακατούμε το διαιρετέο, προσθέτοντας πίσω το διαιρέτη, που ήδη αφαιρέσαμε, και επανατοποθετούμε το διαιρέτη για την επόμενη αφαίρεση. Η διαδικασία επαναλαμβάνεται n φορές, μέχρι να ολοκληρωθεί η διαίρεση.

Διαίρεση χωρίς ανάκτηση

Είναι δυνατό να βελτιώσουμε τη διαδικασία του αλγορίθμου με ανάκτηση αποφεύγοντας την ανάγκη ανάκτησης του μετά από μία αποτυχημένη αφαίρεση, όταν δηλαδή το αποτέλεσμα είναι αρνητικό.

Ας λάβουμε υπόψη τι συμβαίνει μετά την πράξη της αφαίρεσης:

Εάν το περιεχόμενο του καταχωρητή είναι θετικό, ολισθαίνουμε το περιεχόμενό του προς τα αριστερά μία θέση (πολλαπλασιασμός επί δύο) και αφαιρούμε τον πολλαπλασιαστέο (δηλαδή από το διπλάσιο του περιεχομένου του καταχωρητή αφαιρούμε τον πολλαπλασιαστέο).

Εάν το περιεχόμενο του καταχωρητή είναι αρνητικός, κάνουμε την ανάκτηση, εκτελώντας αρχικά πρόσθεση του περιεχομένου του καταχωρητή και του πολλαπλασιαστέου και, κατόπιν, ολισθαίνουμε προς τα αριστερά μια θέση και αφαιρούμε τον πολλαπλασιαστέο (δηλαδή είναι ισοδύναμο με το άθροισμα του διπλάσιου γινομένου του καταχωρητή και του πολλαπλασιαστέου). (Ο αλγόριθμος παρατίθεται στο παράρτημα)

1.6.4.2 Προσημασμένη Διαίρεση

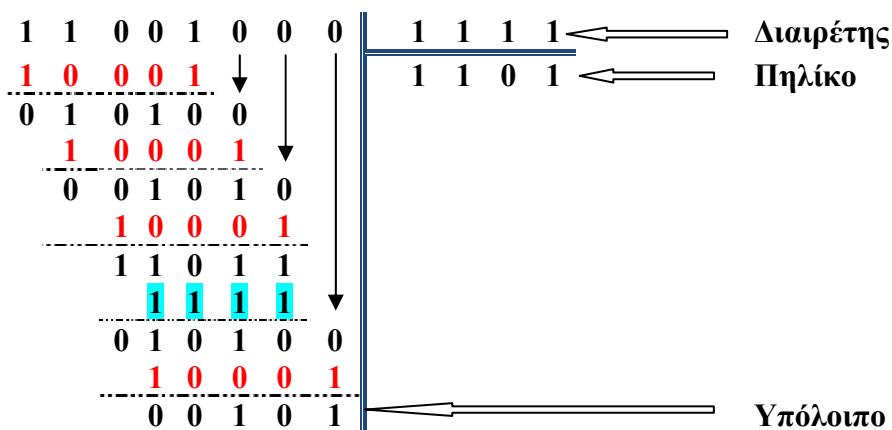
Δεν υπάρχουν απλοί αλγόριθμοι για την εκτέλεση προσημασμένης διαίρεσης, που να είναι συγκρίσιμοι με αυτούς του πολλαπλασιασμού.

Στη διαίρεση, συχνά απαιτείται προ-επεξεργασία των ποσοτήτων και / ή μετέπειτα επεξεργασία των αποτελεσμάτων. Αυτές οι επιπλέον λειτουργίες εξαρτώνται από το πρόσημο των ποσοτήτων σύμφωνα με το επιθυμητό πρόσημο των αποτελεσμάτων.

Παραδείγματα των αλγόριθμων διαίρεσης

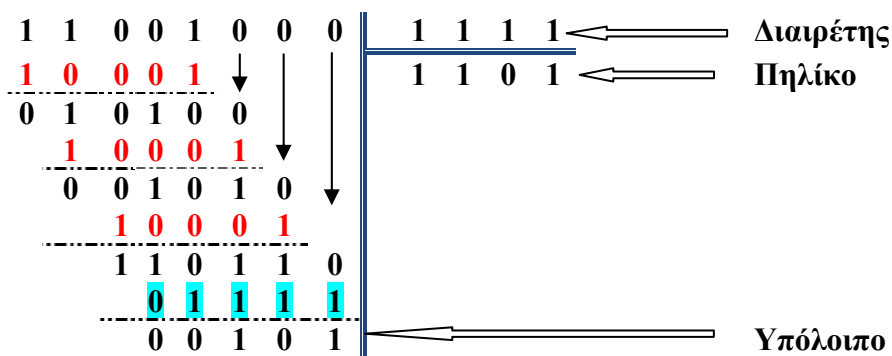
Θα εκτελέσουμε τη διαίρεση 200/15 στο δυαδικό σύστημα. Το $200 = 1001000_{(2)}$ και το $15 = 1111_{(2)}$ στο δυαδικό σύστημα. Έχουμε λοιπόν:

Διαίρεση με ανάκτηση



Σημείωση: Το συμπλήρωμα ως προς δύο του δυαδικού 1111 με πέντε δυαδικά ψηφία είναι: $01111 \rightarrow 10000 \rightarrow 10001$ και εκτελώ πράξεις, κάνοντας πρόσθεση και όχι αφαίρεση. Παρατηρήστε ότι, όταν στο περισσότερο σημαντικό ψηφίο έχω άθροισμα 2, τότε το κρατούμενο(1) αγνοείται.

Διαίρεση χωρίς ανάκτηση



1.7. Πράξεις με προσημασμένους δυαδικούς αριθμούς

Θα αναφερθούμε αρχικά σε πράξεις με προσημασμένους δυαδικούς αριθμούς δίδοντας ιδιαίτερη σημασία στην πρόσθεση, αφού τόσο οι τρεις υπόλοιπες πράξεις (αφαίρεση, πολλαπλασιασμός, διαίρεση) όσο και άλλες συνθετότερες, π.χ. εύρεση λογαρίθμων, καταλήγουν σε αυτήν. Ακολουθούν ορισμένα παραδείγματα πρόσθεσης δυαδικών προσημασμένων αριθμών. (Nashelsky, 1994; Κοσσιδάς, 1996; Holdsworth, & Woods, 2002; Balch, 2003; Givone, 2002; Balabanian, & Carlson, 2007; Maxfield, 2009; Parhami, 2010; Mano, & Ciletti, 2014)

1.7.1. Πρόσθεση αριθμών με συμπλήρωμα ως προς ένα

Για να προσθέσουμε αριθμούς με συμπλήρωμα ως προς ένα: Πρώτα κάνουμε μη προσημασμένη πρόσθεση των αριθμών, συμπεριλαμβάνοντας και τα bits προσήμων. Κατόπιν παίρνουμε το κρατούμενο και το προσθέτουμε στο άθροισμα.

Παράδειγμα 12ο:

Η διαδικασία της πρόσθεσης του κρατουμένου του παραδείγματος που ακολουθεί ονομάζεται **κυκλική μεταφορά κρατουμένου** (end around carry). Αυτός ο τρόπος είναι πιο απλός και πιο ομοιογενής από την πρόσθεση προσημασμένων αριθμών.

α) Οι αριθμοί παριστάνονται με συμπλήρωμα ως προς 1

$\begin{array}{r} 0\ 1\ 1\ 1\ (+7) \\ +\ 1\ 0\ 1\ 1\ (-4) \\ \hline 1\ 0\ 0\ 1\ 0 \\ \xrightarrow{\hspace{2cm}} 1 \\ \hline 0\ 0\ 1\ 1\ (+3) \end{array}$		$\begin{array}{r} 0\ 0\ 1\ 1\ (+3) \\ 0\ 0\ 1\ 0\ (+2) \\ \hline 0\ 1\ 0\ 1 \\ \xrightarrow{\hspace{1cm}} 0 \\ \hline 0\ 1\ 0\ 1\ (+5) \end{array}$
---	--	---

Στα παραδείγματα που ακολουθούν το μήκος κάθε προσημασμένης δυαδικής παράστασης θεωρείται ίσο με 6 δυαδικά ψηφία έχει δε ληφθεί μέριμνα, ώστε το τελικό προσημασμένο άθροισμα να μην υπερβαίνει τα 6 δυαδικά ψηφία. Έτσι το μήκος της δυαδικής προσημασμένης παράστασης είναι αρκετό για την παράσταση του αποτελέσματος.

$\begin{array}{r} 0\ 1\ 0\ 0\ 1\ 0 \\ +\ 1\ 1\ 1\ 0\ 0\ 0 \\ \hline 1\ 0\ 0\ 1\ 0\ 1\ 0 \\ \xrightarrow{\hspace{2cm}} 1 \\ \hline 0\ 0\ 1\ 0\ 1\ 1 \end{array}$	$\begin{array}{r} +\ 18 \\ -\ 7 \\ \hline +\ 11 \end{array}$
$\begin{array}{r} 0\ 1\ 0\ 0\ 1\ 0 \\ +\ 0\ 0\ 0\ 1\ 1\ 1 \\ \hline 0\ 1\ 1\ 0\ 0\ 1 \end{array}$	$\begin{array}{r} +\ 18 \\ +\ 7 \\ \hline +\ 25 \end{array}$
$\begin{array}{r} 0\ 0\ 1\ 0\ 0\ 0 \\ +\ 1\ 0\ 1\ 0\ 0\ 0 \\ \hline 1\ 1\ 0\ 0\ 0\ 0 \end{array}$	$\begin{array}{r} +\ 8 \\ -\ 23 \\ \hline -\ 15 \end{array}$
$\begin{array}{r} 1\ 0\ 1\ 0\ 0\ 1 \\ +\ 1\ 1\ 1\ 1\ 0\ 0 \\ \hline 1\ 1\ 0\ 0\ 1\ 0\ 1 \\ \xrightarrow{\hspace{2cm}} 1 \\ \hline 1\ 0\ 0\ 1\ 1\ 0 \end{array}$	$\begin{array}{r} -\ 22 \\ -\ 3 \\ \hline -\ 25 \end{array}$

1.7.2. Δυαδική Πρόσθεση αριθμών με συμπλήρωμα ως προς δύο

Η μετατροπή ενός αριθμού σε συμπλήρωμα ως προς δύο απαιτεί περισσότερη δουλειά, αλλά η πρόσθεση είναι πολύ πιο εύκολη, από ό,τι με τις άλλες δύο μεθόδους.

Για να βρούμε το $A + B$, πρέπει να:

- Να κάνουμε μη προσημασμένη πρόσθεση των A και B , συμπεριλαμβάνοντας και τα bits προσήμων.
- Να αγνοήσουμε οποιοδήποτε κρατούμενο.

Για παράδειγμα, για να βρούμε το αποτέλεσμα της πράξης $0111 + 1100$ ή $(+7) + (-4)$, έχουμε: Αρχικά προσθέτουμε τους $0111 + 1100$ σαν μη προσημασμένους αριθμούς:

$$\begin{array}{r}
 0 \quad 1 \quad 1 \quad 1 \\
 + 1 \quad 1 \quad 0 \quad 0 \\
 \hline
 1 \quad 0 \quad 0 \quad 1 \quad 1
 \end{array}$$

Αγνοούμε το bit κρατούμενου (1), το δε τελικό αποτέλεσμα είναι 0011 (+3).

Για να πεισθούμε ότι λειτουργεί σωστά, θα προσπαθήσουμε να προσθέσουμε δύο αρνητικούς αριθμούς τους 1101 + 1110, ή (-3) + (-2) στο δεκαδικό.

Προσθέτοντας τους αριθμούς, έχουμε σαν αποτέλεσμα 11011:

$$\begin{array}{r}
 1 \quad 1 \quad 0 \quad 1 \\
 + 1 \quad 1 \quad 1 \quad 0 \\
 \hline
 1 \quad 0 \quad 1 \quad 1
 \end{array}$$

Αγνοώντας το bit κρατούμενου (1), έχουμε το αποτέλεσμα 1011 (-5).

Παρατηρούμε λοιπόν ότι πραγματικά «πετάμε» τα ψηφία που είναι επιπλέον από το μήκος των αρχικών λέξεων.

Το αρχικό κρατούμενο είναι προφανώς 0

	0	1	1	0	↓	
	1	0	0	1		Κρατούμενο εισόδου
+	0	0	1	1		Α' προσθετέος
	1	1	0	0		Β' προσθετέος
	1	1	0	0		Άθροισμα
	↑		↑			
	↑		↑			

↑ Περισσότερο σημαντικό ψηφίο ↑ Λιγότερο σημαντικό ψηφίο

Αν θεωρήσουμε ότι θέλουμε να προσθέσουμε τον αριθμό 9 (1001)(Α προσθετέος) με τον αριθμό 3 (0011)(Β προσθετέος), το αποτέλεσμα είναι προφανώς 12 (1100). Οι ακέραιοι αριθμοί θεωρούνται μη προσημασμένοι.

Στην περίπτωση όπου έχουμε ακεραίους αριθμούς με συμπλήρωμα ως προς 2, και έστω ότι προσθέτουμε τον -7 (1001) με τον 3 (0011), τότε το αποτέλεσμα είναι -4 (1100)! (Για να το αποδείξουμε πρέπει να πάρουμε το συμπλήρωμα του αριθμού και να προσθέσουμε τη μονάδα στο λιγότερο σημαντικό του ψηφίο). *(Χρησιμοποιήσαμε αυτή την παρατήρηση και στον πολλαπλασιασμό ετεροσήμων αριθμών)*

Στα παραδείγματα που ακολουθούν το μήκος κάθε προσημασμένης δυαδικής παράστασης θεωρείται ίσο με 6 δυαδικά ψηφία, έχει δε ληφθεί μέριμνα, ώστε το τελικό προσημασμένο άθροισμα να μην υπερβαίνει τα 6 δυαδικά ψηφία. Έτσι το μήκος της δυαδικής προσημασμένης παράστασης είναι αρκετό για την παράσταση του αποτελέσματος.

Παράδειγμα 13ο:

$$\begin{array}{r}
 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad + \quad 18 \\
 + 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad - \quad 8 \\
 \hline
 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad + \quad 10
 \end{array}$$

Ο πρώτος άσσος δε λαμβάνεται υπόψη. Έτσι το αποτέλεσμα είναι 001010, που πράγματι αντιστοιχεί στο δεκαδικό 10.

$$\begin{array}{r}
 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad + \quad 18 \\
 + 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad + \quad 7 \\
 \hline
 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad + \quad 25
 \end{array}$$

Δεν παρατηρούμε καμία διαφορά μεταξύ των δύο παραστάσεων (συμπλήρωμα ως προς ένα και συμπλήρωμα ως προς 2 στους θετικούς).

1.8. Ασκήσεις – Ερωτήσεις

1. Να αναφέρετε τα κύρια αριθμητικά συστήματα καθώς και τα ψηφία του καθενός.
2. Δίνονται οι δεκαεξαδικοί αριθμοί: α) 7AE,C3 β) AB2,7E γ) 1FF,88. Να μετατραπούν στους αντίστοιχους οκταδικούς, δεκαδικούς και δυαδικούς αριθμούς.
3. Στον πίνακα που ακολουθεί να συμπληρώσετε τους αριθμούς στα άλλα συστήματα

Δεκαδικό	Οκταδικό	Δυαδικό	Δεκαεξαδικό	Πενταδικό
				1234
			2C,F3	
		101101		
	453,12			
29,675				

4. Να μετατρέψετε τους παρακάτω αριθμούς του δεκαδικού συστήματος στους αντίστοιχους δυαδικούς: α) 148 β) 0,48679 γ) 0,572 δ) 14,725
5. Να μετατραπεί ο δυαδικός αριθμός $N = 101010011100$ στον ισοδύναμο δεκαεξαδικό και στον ισοδύναμο οκταδικό αριθμό.
6. Οι αριθμοί 110010100100 και 001101010101 είναι δυαδικοί με παράσταση συμπληρώματος ως προς 1. Να μετατραπούν σε παράσταση συμπληρώματος ως προς 2 και να βρεθούν οι ισοδύναμοι αριθμοί του δεκαδικού συστήματος αρίθμησης.
7. Να γίνει η πρόσθεση των αριθμών -19 και -40 στη δυαδική αναπαράσταση των αριθμών με χρήση οκτώ δυαδικών ψηφίων (bits).
8. Οι αριθμοί του δεκαδικού συστήματος αρίθμησης +375 και -127 πρόκειται να προστεθούν από υπολογιστή μήκους λέξης 11 δυαδικών ψηφίων και με παράσταση συμπληρώματος ως προς 1. Να δοθεί η μορφή με την οποία θα προστεθούν οι αριθμοί, καθώς και η μορφή του αποτελέσματος. Να ελέγξετε το αποτέλεσμα μέσω του δυαδικού συστήματος αρίθμησης.
9. Δίδονται οι παρακάτω προσημασμένοι δυαδικοί αριθμοί:
α) -10101 β) 0,1101 γ) ± 0 δ) -10,10001
Να μετατραπούν στους δυαδικούς ισοδύναμους θετικούς και αρνητικούς αριθμούς με προσημασμένη παράσταση, καθώς και στους ισοδύναμους με προσημασμένη παράσταση συμπληρώματος ως προς 1 και 2.
10. Ποιος είναι ο μεγαλύτερος και ποιος ο μικρότερος αριθμός, που μπορεί να εισαχθεί σε υπολογιστή μήκους λέξης 18 δυαδικών ψηφίων με συμπλήρωμα ως προς 1 και 2;
11. Να εκτελέσετε τον πολλαπλασιασμό των δεκαδικών αριθμών -7 επί 10. Με πόσα δυαδικά ψηφία πρέπει να γίνει η αναπαράσταση των αριθμών; Να επιβεβαιώσετε το αποτέλεσμα ως προς την ορθότητά του.

Αναφορές-Βιβλιογραφία

- Κοσσίδης Α.Θ. (1996). *Σχεδίαση Ψηφιακών Κυκλωμάτων*, Εκδόσεις Μπένος
- Κοσσίδης Α.Θ., Γιαννακόπουλος Π. (2006). *Αριθμητικά Συστήματα και Ψηφιακά Κυκλώματα*, Εκδόσεις Νέες Τεχνολογίες, Αθήνα
- Φραγκάκης Γ. (1975). *Λογικά Κυκλώματα*, Αθήνα
- Balch M. (2003). *Complete Digital Design*, Mc Graw Hill, New York
- Balabanian N., Carlson B. (2007). *Digital Logic Design Principles*, John Wiley
- Floyd Thomas L. (2013). *Digital Fundamentals: A systems Approach*, Pearson International Edition
- Givone D. (2002). *Digital Principles and Design*, Mc Graw Hill
- Holdsworth Brian, Woods Clive (2002). *Digital Logic Design*, 4th Edition, Newnes
- Katz R., (2005). *Contemporary Logic Design*, 2/e, Prentice Hall
- Mano M., Ciletti M., (2014). *Ψηφιακή Σχεδίαση*, 5^η έκδοση, Παπασωτηρίου
- Maxfield Clive (2009). *Bebop to the Boolean Boogie, An Unconventional Guide to Electronics*, 3rd, Newnes, Elsevier (Online έκδοση: <http://english.360elib.com/datu/T/EM140371.pdf>)
- Nashelsky Louis (1994). *Introduction to Digital Technology*, 4th Ed., Prentice Hall
- Nelson V., Nagle H., Carroll B., Irwin J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall
- Parhami B. (2010). *Computer Arithmetic: Algorithms and Hardware Designs*, 2nd edition, Oxford University Press, New York
- Predko Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill
- Pritchard N. (2015). *Fundamentals of Digital Electronics*, CreateSpace Independent Publishing Platform
- Rajaraman V., Adabala N. (2015). *Fundamental of Computers*, 6th, PHI Learning
- Roth Charles Jr., Kinney L.L. (2014). *Fundamentals of Logic Design*, 7th, Cengage Learning
- Tocci R. J., Widmer N. S, and Moss Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley
- Wakerly J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Διαδικτυακοί τόποι

Αριθμητικά Συστήματα, Ανακτήθηκε 17/2/2015 από https://en.wikipedia.org/wiki/List_of_numeral_systems

Διαδικτυακοί τόποι περαιτέρω μελέτης:

http://www.cs.virginia.edu/~cs216/Fall2005/notes/number_lect.pdf (Αριθμητικά Συστήματα)

<http://www.allaboutcircuits.com/textbook/digital/chpt-1/numbers-and-symbols/> (Αριθμητικά Συστήματα)

<http://www.allaboutcircuits.com/textbook/digital/chpt-2/numbers-versus-numeration/> (Δυαδική Αριθμητική)

http://www.electronics-tutorials.ws/binary/bin_1.html (Δυαδική αριθμητική)

<http://www.swarthmore.edu/NatSci/echeeve1/Ref/BinaryMath/BinaryMath.html> (Δυαδική αριθμητική)

Κεφάλαιο 2

Σύνοψη

Στο κεφάλαιο αυτό θα παρουσιαστεί αναλυτικά η κωδικοποίηση των αριθμών μέσω του δυαδικού συστήματος αρίθμησης. Αρχικά περιλαμβάνονται τα ψηφία του δεκαδικού συστήματος κωδικοποιημένα στο δυαδικό. Ακολουθούν τα ψηφία του οκταδικού και του δεκαεξαδικού συστήματος κωδικοποιημένα επίσης στο δυαδικό. Κατόπιν ακολουθεί η παρουσίαση των κωδίκων που χρησιμοποιούνται συχνά. Έτσι αναλύονται οι: δυαδικός κώδικας (Binary code), οι κώδικες βαρών 5421, 5311, ο κώδικας Aiken ή 2421, ο XS3 (Υπερβολής κατά τρία), ο δυεπταδικός, ο δυδεκαδικός κώδικας. Τέλος αναφορά γίνεται στους κώδικες 2-από-5 και στον κώδικα Gray. Η κωδικοποίηση των χαρακτήρων στο δυαδικό σύστημα γίνεται με τους αλφαριθμητικοί κώδικες. Ακολουθεί η ασφάλεια κωδικοποίησης των χαρακτήρων. Οι κώδικες διόρθωσης σφαλμάτων, με κύρια αναφορά στον κώδικα Hamming. Βασική για τους υπολογιστές είναι και η αριθμητική κινητής υποδιαστολής και το πρότυπο IEEE-754. Δίνεται η αναπαράσταση των αριθμών κινητής υποδιαστολής απλής/διπλής ακριβείας. Η αριθμητική κινητής υποδιαστολής καθώς και η μετατροπή αριθμών μεταξύ της IEEE-754 και του δεκαδικού συστήματος με τις βασικές πράξεις αριθμών κινητής υποδιαστολής ολοκληρώνουν το κεφάλαιο.

Προαπαιτούμενη γνώση

Βασικές γνώσεις των αριθμητικών συστημάτων καθώς και οι αριθμητικές πράξεις.

2. Κωδικοποίηση των ψηφίων του δεκαδικού συστήματος

2.1. Κωδικοποίηση των ψηφίων του δεκαδικού μέσω του δυαδικού συστήματος

Όπως αναφέραμε, η λειτουργία των υπολογιστών στηρίζεται στο δυαδικό σύστημα αρίθμησης. Οι οποιοσδήποτε πράξεις πρέπει να εκτελούνται στο σύστημα αυτό. Κατά συνέπεια, τα προς εισαγωγή δεδομένα, ανεξάρτητα από τον τρόπο εγγραφής τους, πρέπει να κωδικοποιούνται αυτόματα από τον υπολογιστή στο δυαδικό σύστημα. Το ίδιο ισχύει και για την εξαγωγή τους. Τα αποτελέσματα μετά την επιτυχή, ή όχι, εκτέλεση των πράξεων πρέπει να αποκωδικοποιούνται σε σύστημα αρίθμησης κατανοητό από τον άνθρωπο. Για την κωδικοποίηση των ψηφίων των διαφόρων συστημάτων αρίθμησης μέσω του δυαδικού συστήματος απαιτούνται τόσα δυαδικά ψηφία έτσι, ώστε ο συνδυασμός αυτών να μπορεί να καλύψει τα κωδικοποιούμενα ψηφία. Στις περιπτώσεις στις οποίες το μεγαλύτερο ψηφίο του συστήματος που κωδικοποιείται είναι δύναμη του 2, τότε κατά την κωδικοποίηση έχουμε μονοσήμαντο κώδικα. Σε αντίθετη περίπτωση, όταν το μεγαλύτερο ψηφίο του προς κωδικοποίηση συστήματος δεν είναι δύναμη του 2, τότε η κωδικοποίηση είναι πλεονάζουσα, και μπορούν να δημιουργηθούν περισσότεροι του ενός τρόποι με τους οποίους μπορούμε να παραστήσουμε τα δεκαδικά ψηφία, και, συνεπώς, έχουμε περισσότερους κώδικες.

2.2. Οκταδικός κωδικοποιημένος στο δυαδικό

Τα οκτώ ψηφία του οκταδικού συστήματος αρίθμησης για την κωδικοποίησή τους απαιτούν 3 δυαδικά ψηφία. Η κωδικοποίηση είναι μονοσήμαντη, αφού $2^3 = 8$. Στον πίνακα 2.1, που ακολουθεί, δίνεται η αντιστοιχία μεταξύ οκταδικών ψηφίων και των αντίστοιχων τριάδων δυαδικών ψηφίων.

Ακολουθούν μερικά παραδείγματα για την καλύτερη κατανόηση της μετατροπής: Ο αριθμός $N_8 = 254$ κωδικοποιημένος στο δυαδικό είναι, επομένως, ίσος με τον **010-101-100** δηλαδή τον **010101100**.

Οκταδικό ψηφίο	Δυαδική κωδικοποίηση	Οκταδικό ψηφίο	Δυαδική κωδικοποίηση
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

Πίνακας 2.1. Δυαδική παράσταση των οκταδικών ψηφίων

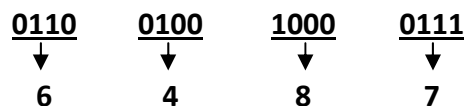
Εάν έχουμε έναν οκταδικό αριθμό κωδικοποιημένο στο δυαδικό σύστημα, για να βρούμε τα ψηφία του στο οκταδικό, το χωρίζουμε σε τριάδες δυαδικών ψηφίων, ξεκινώντας από το δεξιότερο ψηφίο του. π.χ. ο οκταδικός κωδικοποιημένος στο δυαδικό αριθμός **001100110**, στο οκταδικό είναι ίσος με τον $N_8 = 146$

2.3. Δεκαεξαδικός κωδικοποιημένος στο δυαδικό

Και στο δεκαεξαδικό σύστημα αρίθμησης έχουμε μονοσήμαντη αντιστοιχία των ψηφίων. Πράγματι η βάση του δυαδικού υψωμένη στην τέταρτη δύναμη δίνει τη βάση του δεκαεξαδικού, δηλαδή $2^4 = 16$. Απαιτούνται, επομένως, τέσσερα δυαδικά ψηφία για τον καθορισμό ενός δεκαεξαδικού ψηφίου. (Wakerly, 2006; Mano, & Ciletti, 2014)

Ο αριθμός $N_{16} = 359_{16}$ κωδικοποιημένος στο δυαδικό γράφεται: **001101011001**.

Ο δεκαεξαδικός κωδικοποιημένος στο δυαδικό **0110010010000111** παριστάνει το δεκαεξαδικό αριθμό: $N_{16} = 6487$ (ο διαχωρισμός γίνεται πάντα από δεξιά).



Στον πίνακα 2.2 που ακολουθεί αποδίδεται η αντιστοιχία των δεκαεξαδικών αριθμών με τους αντίστοιχους τετραψήφιους δυαδικούς αριθμούς.

Δεκαεξαδικό ψηφίο	Δυαδική αναπαράσταση	Δεκαεξαδικό ψηφίο	Δυαδική αναπαράσταση
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Πίνακας 2.2. Δυαδική παράσταση δεκαεξαδικού ψηφίου

2.4. Δεκαδικός κωδικοποιημένος στο δυαδικό

Δεδομένου ότι ο αριθμός των ψηφίων του δεκαδικού συστήματος αρίθμησης δεν είναι δύναμη του 2, έχουμε πλεονασμό κωδικοποίησης με αποτέλεσμα τη δημιουργία πάρα πολλών κωδικών. Πράγματι για την κωδικοποίηση των δέκα ψηφίων απαιτούνται 4 δυαδικά ψηφία. (Φραγκάκης, 1975; Nashelsky, 1994; Κοσσίδης, 1996; Blahut, 2003; Givone, 2002; Balch, 2003; Predko, 2005; Balabanian, & Carlson, 2007; Floyd, 2013; Mano, & Ciletti, 2014; Pritchard, 2015; Rajaraman, & Adabala, 2015)

Δεκαδικό ψηφίο	Δυαδικός Κώδικας BCD 8421	Κώδικας Aiken 2421	Κώδικας Aiken 84-2-1	Κώδικας Υπερβολής κατά 3 (Excess -3)
0	0000	0000	0000	0011
1	0001	0001	0111	0100
2	0010	0010	0110	0101
3	0011	0011	0101	0110
4	0100	0100	0100	0111
5	0101	1011	1011	1000
6	0110	1100	1010	1001
7	0111	1101	1001	1010
8	1000	1110	1000	1011
9	1001	1111	1111	1100

Πίνακας 2.3α. Αντιστοιχία των δεκαδικών ψηφίων και των κυρίων κωδικών

Όλοι οι κώδικες χωρίζονται σε τρεις κατηγορίες α) Στους κώδικες βαρών, που είναι και η πλειονότητα, β) στους δυαδικούς κώδικες των ψηφίων του δεκαδικού, που είναι χωρίς βάρη και γ) στο κώδικα Gray. Στους πίνακες 2.3α, 2.3β και 2.3γ παραθέτουμε τους κυριότερους κώδικες του δεκαδικού συστήματος αρίθμησης.

Στον πίνακα 2.3α εκτός από τους κώδικες με τέσσερα δυαδικά ψηφία δίνονται και άλλοι που αποτελούνται από περισσότερα από τέσσερα δυαδικά ψηφία για κάθε δεκαδικό. Οι λόγοι δημιουργίας δεκαδικών κωδίκων με περισσότερα από τέσσερα δυαδικά ψηφία σχετίζονται κυρίως με την απλοποίηση των κυκλωμάτων κωδικοποίησης- αποκωδικοποίησης. Το τελευταίο γίνεται εμφανές στις δύο τελευταίες στήλες του πίνακα 2.3β, όπου άξιο παρατήρησης είναι το γεγονός ότι στο δυεπταδικό κώδικα (Biquinary) κάθε δεκαδικό ψηφίο έχει δύο δυαδικές μονάδες, ενώ στο δυδεκαδικό (Bin-decimal) έχει μόνο μια μονάδα, καθώς περιλαμβάνονται όλα τα ψηφία από το 0 μέχρι και το 9.

Άλλοι κώδικες με βάρη, που μπορούμε να χρησιμοποιήσουμε για την αναπαράσταση των δεκαδικών ψηφίων, είναι ο κώδικας 7421, ο κώδικας 753 $\bar{6}$, ο οποίος χρησιμοποιεί αρνητικά βάρη, ο 5421, ο κώδικας 5211 (κατοπτρικός κώδικας- αυτοσυμπληρούμενος κώδικας), και τέλος όλοι η σειρά των κωδίκων που ονομάζεται p από q με πιο κοινό τον κώδικα 2-από-5 κ.λπ. (να σχηματίσετε τα ψηφία των ανωτέρω κωδίκων στο δυαδικό σύστημα αρίθμησης).

Αναφέρεται επικουρικά ότι οι κώδικες, των οποίων τα βάρη έχουν άθροισμα 9, ονομάζονται αυτοσυμπληρούμενοι κώδικες, και είναι οι πλέον κατάλληλοι για την εκτέλεση αριθμητικών πράξεων. (Ποιοι από τους κώδικες των πινάκων 2.3α, β, γ, δ είναι αυτοσυμπληρούμενοι;)

Βασικοί δυαδικοί κώδικες κωδικοποίησης αριθμών του δεκαδικού συστήματος (και συνεπώς τετραδικοί κώδικες είναι οι: 8421(BCD κώδικας), 5421, 2421 Κώδικας Aiken (αυτοσυμπληρούμενος), 4221 Κώδικας Aiken, 5311, 7421(κώδικας ελαχιστοποίησης των άσπων), 74-2-1.

2.4.1. Δυαδικός κώδικας (Binary code)

Ο δυαδικός κώδικας BCD χρησιμοποιεί τα βάρη του δυαδικού συστήματος αρίθμησης για την κωδικοποίηση των δεκαδικών ψηφίων. Στον πίνακα 1.12α φαίνεται ο τρόπος κωδικοποίησης. Π.χ. ο αριθμός 23 κωδικοποιείται σαν 0010 0011. Είναι προφανές ότι κάθε ψηφίο του δεκαδικού κωδικοποιείται ξεχωριστά στο δυαδικό σύστημα (γράφουμε την αντίστοιχη ισοδύναμη τετράδα).

2.4.2. Οι κώδικες 5421 και 5311

Στον κώδικα 5421 εάν ένας αριθμός έχει περισσότερες από μια αναπαράστασεις, χρησιμοποιούμε αυτή η οποία έχει το λιγότερο σημαντικό ψηφίο βάρους, πχ. για την τιμή 5 δε θα χρησιμοποιήσουμε την 1000 αλλά την αναπαράσταση 0101. (Pritchard, 2015)

Αντίστοιχα στον κώδικα 5311 στην αντίστοιχη περίπτωση χρησιμοποιούμε την κωδικοποίηση με τον μικρότερο αριθμό άσπων, χρησιμοποιώντας, επίσης, τα λιγότερο σημαντικά ψηφία.

Δεκαδικό ψηφίο	Κώδικας 5421	Κώδικας 5311	Κώδικας 4221	Κώδικας 7421
0	0000	0000	0000	0000
1	0001	0001	0001	0111
2	0010	0011	0010	0010
3	0011	0100	0011	0011
4	0100	0101	0110	0100
5	0101	0111	1001	0101
6	0110	1001	1100	0110
7	0111	1011	1101	1000
8	1011	1100	1110	1001
9	1100	1101	1111	1010

Πίνακας 2.3β. Αντιστοιχία των δεκαδικών ψηφίων και κυρίων κωδίκων

2.4.3. Κώδικας Aiken ή κώδικας 2421 ή κώδικας 8 4 -2-1

Ο κώδικας Aiken, όπως και ο δυαδικός, μας δίνει τη δυνατότητα να προσδιορίσουμε αν ο κωδικοποιούμενος αριθμός είναι άρτιος ή περιττός. Η πληροφορία αυτή αναδεικνύεται από την τιμή του δεξιότερου ψηφίου του. Είναι ένας κώδικας με βάρη και μέγιστο αριθμό τον μέγιστο δεκαδικό. Παρέχει τη δυνατότητα εύκολης συ-

μπλήρωσης, σε περίπτωση που χρειαστεί να εκτελεστούν πράξεις αφαίρεσης, και διατηρεί την ιδιότητα των κωδικών που δίνουν βάρος στα δυαδικά ψηφία, από τα οποία δημιουργούνται τα ψηφία του κώδικα. Παρατηρήστε στον κώδικα αυτό ότι μετά την αναπαράσταση του αριθμού 4 του δεκαδικού συστήματος (πέμπτο στοιχείο του κώδικα) τα υπόλοιπα στοιχεία του προκύπτουν ως συμπληρωματικά κατοπτρικά των προηγούμενων. (Ψηφία 5 έως και 9).

2.4.4. Κώδικας Υπερβολής κατά τρία

Ο κώδικας υπερβολής κατά τρία (Excess three, XS3), όπως και ο προηγούμενος, διαθέτει ευκολία συμπλήρωσης και αποτελεί μια παραλλαγή του Δυαδικού κώδικα. Ονομάζεται και κώδικας Stibitz. Προκύπτει από το δυαδικό κώδικα, αν προσθέτουμε το τρία στη δυαδική μορφή του καθενός κωδικοποιημένου ψηφίου. Και αυτός ο κώδικας είναι κατοπτρικός συμπληρωματικός κώδικας. Τέλος αναφέρουμε ότι είναι ένας κώδικας ο οποίος δεν έχει βάρη.

Είναι προφανές ότι, αν θέλουμε μπορούμε να «φτιάξουμε» οποιονδήποτε κώδικα υπερβολής, αρκεί η τελευταία δυαδική αναπαράσταση του κώδικα να μην υπερβαίνει τον αριθμό 1111. (Αυτοσυμπληρούμενοι κώδικες είναι οι XS3, 5211, 4221, 84-2-1). (Ποιος είναι ο μέγιστος δυνατός κώδικας υπερβολής για την αναπαράσταση των ψηφίων του δεκαδικού συστήματος;).

2.4.5. Δυπενταδικός και Δυδεκαδικός κώδικας

Ο Δυπενταδικός κώδικας (Biquinary code) κωδικοποιεί κάθε δεκαδικό ψηφίο με τη βοήθεια 7 bit χωρισμένων σε δύο ομάδες. Η πρώτη περιλαμβάνει 2 bit και η δεύτερη 5. Κάθε κωδικοποιούμενο ψηφίο περιλαμβάνει πάντοτε δύο δυαδικές μονάδες, μία σε κάθε τμήμα. Το μεγάλο πλεονέκτημα του Δυπενταδικού κώδικα είναι ο αυτοέλεγχος όσο αφορά στην ασφάλεια μεταβίβασης πληροφοριών. Χρησιμοποιείται συνήθως για την ανίχνευση λαθών και η ύπαρξη λιγότερων ή περισσότερων άσων των δύο προσδιορίζει την ύπαρξη λάθους.

Ο Δυδεκαδικός κώδικας (Bindecimal code) περιλαμβάνει 10 bit για την κωδικοποίηση κάθε ψηφίου. Τα πλεονεκτήματά του, όπως και του δυπενταδικού, είναι η απλότητα κωδικοποίησης και η ασφάλεια μεταφοράς πληροφοριών. Μειονεκτήματα των κωδικών αυτών είναι το μεγάλο μήκος του κώδικα και, κατά συνέπεια, η μικρότερη ταχύτητα μετάδοσης πληροφοριών. (Πίνακας 2.3γ).

Δεκαδικό ψηφίο	Κώδικας Gray	Κώδικας Biquinary 50,43210	Κώδικας Bindecimal 9876543210
0	0000	01,00001	000000001
1	0001	01,00010	000000010
2	0011	01,00100	000000100
3	0010	01,01000	000001000
4	0110	01,10000	000010000
5	0111	10,00001	000010000
6	0101	10,00010	000100000
7	0100	10,00100	001000000
8	1100	10,01000	010000000
9	1101	10,10000	100000000

Πίνακας 2.3γ. Αντιστοιχία κωδικών στο δεκαδικό σύστημα

Δεκαδικό ψηφίο	Κώδικας 74210	Κώδικας 5211	Κώδικας 7536
0	11000	0000	0000
1	00011	0001	1001
2	00101	0100	0111
3	00110	0101	0010
4	01001	0111	1011
5	01010	1000	0100
6	01100	1001	1101
7	10001	1100	1000
8	10010	1101	0110
9	10100	1111	1111

Πίνακας 2.3δ. Κώδικες αντιστοίχισης ψηφίων του δεκαδικού συστήματος

2.4.6. Κώδικας 2-από-5

Ο κώδικας 2 -από -5 (2-out of-5) Είναι και αυτός, όπως και ο κώδικας υπερβολής κατά 3, ένας κώδικας ο οποίος δεν έχει βάρη, έχει όμως την ιδιότητα, όπως και ο προηγούμενος κώδικας, να περιέχει μόνο δύο άσσους σε κάθε κωδικοποιημένη του λέξη.

Παρόλο που, όπως αναφέραμε, δεν έχει βάρη μπορούμε να πούμε ότι είναι κώδικας με βάρη 74210 εκτός της τιμής που αντιστοιχεί στο μηδέν (11000). Χρησιμοποιείται συνήθως για την ανίχνευση ενός λάθους και η ύπαρξη άσπων λιγότερων ή περισσότερων των δύο προσδιορίζει την ύπαρξη λάθους. (πίνακας 2.3δ).

2.4.7. Κώδικας Gray (Κυκλικός κώδικας)

Ο κώδικας Gray βρίσκει μεγάλη χρησιμότητα στις διατάξεις εισόδου- εξόδου πληροφοριών, αφού βελτιώνει τη δυνατότητα ανάγνωσης- εκτύπωσης πληροφοριών και απλουστεύει την κατασκευή των αντιστοιχών διατάξεων. Βασική ιδιότητα του κώδικα αυτού είναι ότι, μεταξύ διαδοχικών αριθμών του δεκαδικού συστήματος αρίθμησης, ο αντίστοιχος κώδικας Gray διαφέρει μόνο κατά ένα δυαδικό ψηφίο, όπως φαίνεται και στον πίνακα 2.3γ. (Nashelsky, 1994; Predko, 2005; Balabanian, & Carlson, 2007)

Για την κωδικοποίηση αριθμών με περισσότερα από ένα δεκαδικά ψηφία δεν κωδικοποιείται κάθε ψηφίο χωριστά, για να τοποθετηθούν κατόπιν οι αντίστοιχες τετράδες ή μία δίπλα στην άλλη, αλλά κωδικοποιείται όλος ο αριθμός μαζί, αφού υπολογισθεί η δυαδική τιμή του.

Παρακάτω, με τη βοήθεια μερικών παραδειγμάτων, θα εξηγήσουμε τον τρόπο κωδικοποίησης του συγκεκριμένου κώδικα. Έστω ο αριθμός $N_{10} = 135$ για τον οποίο θέλουμε να βρούμε τον αντίστοιχό του σε κώδικα Gray. Έχουμε ότι:

$$N_{10} = 135 \longrightarrow \begin{array}{l} 1000111 \\ 1100100 \end{array} \quad \begin{array}{l} \text{Δυαδικός} \\ \text{Gray} \end{array}$$

Αρχικά βρίσκουμε τον αντίστοιχο δυαδικό του δεκαδικού και στη συνέχεια εφαρμόζουμε τον παρακάτω κανόνα:

Αρχίζουμε από το ψηφίο μεγαλύτερης τάξης του δυαδικού και το συγκρίνουμε με το αμέσως προηγούμενο (για το πρώτο ψηφίο το προηγούμενο είναι πάντοτε το μηδέν). Εάν κατά τη σύγκριση έχουμε τα ίδια ψηφία, θέτουμε «0». Εάν έχουμε διαφορετικά θέτουμε «1» κ.ο.κ., έως ότου εξαντλήσουμε όλα τα δυαδικά ψηφία. Η μετατροπή από τον δυαδικό αριθμό στον αντίστοιχο του κώδικα Gray μπορεί να φανεί και από το παρακάτω παράδειγμα:

$$\begin{array}{ll} \text{Δυαδικός} & 0100101011001 \\ \text{Gray} & 011011110101 \end{array}$$

Ένας δεύτερος τρόπος που μπορούμε να χρησιμοποιήσουμε είναι η πρόσθεση των ψηφίων του αρχικού αριθμού ανά δύο (στη θέση που θέλουμε να τοποθετήσουμε το αποτέλεσμα και στην αμέσως προηγούμενη). (Να ελέγξετε το αποτέλεσμα αυτής της μεθόδου).

Για τη μετατροπή ενός αριθμού από μορφή Gray σε δυαδική μορφή, η διαδικασία είναι περισσότερο πολύπλοκη.

Δεκαδικός	Gray	Δεκαδικός	Gray
0	0000	10	1111
1	0001	11	1110
2	0011	12	1010
3	0010	13	1011
4	0110	14	1001
5	0111	15	1000
6	0101	16	11000
7	0100	17	11001
8	1100	18	11011
9	1101	19	11010

Πίνακας 2.4. Κώδικας Gray

Σ' αυτήν την περίπτωση αρχίζουμε από τα αριστερά του αριθμού και επαναλαμβάνουμε τα δυαδικά ψηφία, μέχρις ότου βρούμε ένα ψηφίο ίσο με «1». Μετά την πρώτη δυαδική μονάδα του Gray, εάν ακολουθούν μηδενικά, ένα ή περισσότερα, επαναλαμβάνουμε στο δυαδικό αριθμό την τιμή του προηγούμενου δυαδικού του ψηφίου. Στη συνέχεια, εάν στον Gray έχουμε «1», βάζουμε στο δυαδικό την αντίθετη τιμή από αυ-

τή που είχε το προηγούμενο δυαδικό του ψηφίο, ενώ, εάν έχουμε «0», επαναλαμβάνουμε την τιμή που είχε το προηγούμενο δυαδικό του ψηφίο κ.ο.κ. Στον πίνακα 2.4 δίδονται οι αριθμοί από το 0 έως το 19 μαζί με τους ισοδύναμους τους σε κώδικα Gray. Ακολουθούν δύο παραδείγματα μετατροπής από τον κώδικα Gray στο δυαδικό, όπου εφαρμόζουμε τον παραπάνω αλγόριθμο.

Gray	010100001001	Gray	11001010100101
Δυαδικός	011000001110	Δυαδικός	10001100111001

2.5. Αλφαριθμητικοί κώδικες–Κωδικοποίηση χαρακτήρων μέσω του δυαδικού συστήματος αρίθμησης

Με τον όρο «χαρακτήρα» εννοούμε αριθμούς, γράμματα της αλφαβήτου, σημεία στίξης και γενικά κάθε σύμβολο που χρησιμοποιείται κατά την εγγραφή ενός προγράμματος ή την εκτύπωση ενός αποτελέσματος.

Εάν λάβουμε υπόψη μας ότι τα ψηφία του δεκαδικού συστήματος είναι 10 και τα γράμματα του Λατινικού αλφαβήτου 26, απαιτούνται τουλάχιστον 6 δυαδικά ψηφία, για να κωδικοποιηθούν. Με 6 δυαδικά ψηφία έχουμε όμως 64 συνδυασμούς και, κατά συνέπεια, μας περισσεύουν 28 για τα ειδικά σύμβολα και τα σημεία στίξης.

Για την κωδικοποίηση, επομένως όλων των χαρακτήρων, απαιτούνται τουλάχιστον 6 δυαδικά ψηφία. Με βάση το γεγονός αυτό έχουν αναπτυχθεί αρκετοί κώδικες, οι κυριότεροι των οποίων είναι ο Αλφαβητικός Εξαψήφιος Κώδικας που παρατίθεται στον πίνακα 1.1 του Παραρτήματος, ο ASCII (American Standards Code for Information Interchange) που παρατίθεται στους πίνακες του παραρτήματος και ο EBCDIC (Extended Binary Coded Decimal Interchange Code) που παρατίθεται στο παράρτημα.

Ο κώδικας ASCII έχει 8 δυαδικά ψηφία και χρησιμοποιείται διεθνώς σε εφαρμογές με μικροεπεξεργαστές. Οι πρώτοι 128 χαρακτήρες του περιλαμβάνουν τα αλφαβητικά και αριθμητικά σύμβολα, ενώ οι δεύτεροι 128 περιλαμβάνουν, κατά κανόνα, σύμβολα που έχουν σχέση με την τοπική γλώσσα του κράτους που χρησιμοποιείται. Για το λόγο αυτό τον έχουμε χωρίσει σε δύο μέρη. Το πρώτο από αυτά (bit7=0) είναι αυτό με το διεθνές σετ συμβόλων, ενώ το δεύτερο (bit7=1) περιλαμβάνει τους χαρακτήρες της Ελληνικής γλώσσας. (Wakerly, 2006; Balabanian, & Carlson, 2007; Tocci, et al., 2010; Mano, & Ciletti, 2014)

Οι πρώτοι 32 χαρακτήρες μεταφέρουν μη εκτυπώσιμα σύμβολα, αυτά που είναι περισσότερο γνωστά σαν χαρακτήρες ελέγχου. Οι χαρακτήρες αυτοί έχουν ειδικές σημασίες και αποτείνονται προς τη μηχανή. Μιλώντας πιο συγκεκριμένα, μπορούμε να πούμε πως οι χαρακτήρες αυτοί αναγκάζουν, π.χ. την κεφαλή του εκτυπωτή να μεταφερθεί στο αριστερό μέρος της σελίδας και να αλλάξει γραμμή, να προχωρήσει στην αρχή της επόμενης σελίδας κ.λπ. Ακόμα ένας από αυτούς τους χαρακτήρες προκαλεί την εκπομπή ενός σύντομου ακουστικού σήματος, χρήσιμου σαν σήματος ειδοποίησης προς το χειριστή της μηχανής.

Στον πίνακα αυτόν γίνεται εμφανής η παρουσία των Ελληνικών συμβόλων, τα οποία, όπως και στην περίπτωση του διεθνούς κώδικα, εμφανίζονται με διαφορετικό κωδικό ανάλογα με τον αν είναι πεζά ή κεφαλαία. Αξιοσημείωτο είναι ότι τα φωνήεντα εμφανίζονται και αυτά με δύο κωδικούς ανάλογα με το αν είναι τονισμένα ή όχι.

Ο κώδικας EBCDIC, αποτελεί μία τροποποίηση του ASCII, και χρησιμοποιείται ευρέως από την IBM. Δε χρησιμοποιείται σε εφαρμογές με μικροεπεξεργαστές.

2.5.1. Χρήση της ισοτιμίας στην ασφαλή κωδικοποίηση των χαρακτήρων

Κατά τη μεταφορά κωδικοποιημένων χαρακτήρων τόσο μέσα σε ένα υπολογιστικό σύστημα όσο και έξω από αυτό θα πρέπει να είμαστε βέβαιοι για το κατά πόσο οι μεταφερόμενες πληροφορίες είναι σωστές, δηλαδή για το κατά πόσο, στη διάρκεια της μεταφοράς των δυαδικών ψηφίων «0» ή «1», δε συνέβησαν μεταβολές ικανές να αλλοιώσουν τους κωδικοποιούμενους χαρακτήρες.

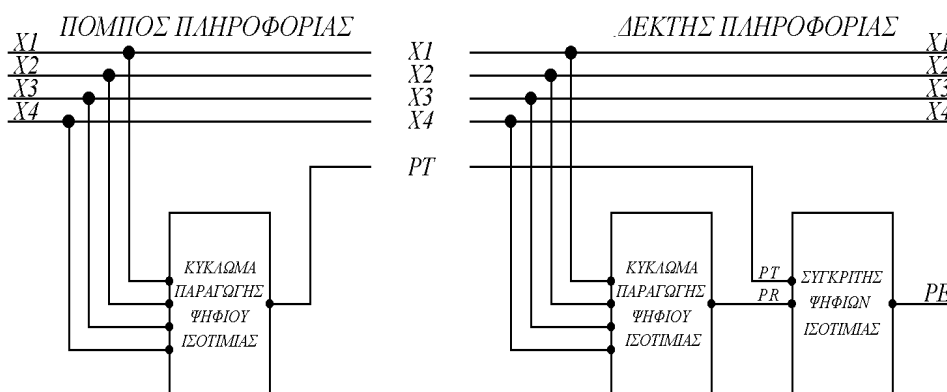
Για να είναι εύκολη η αναγνώριση ενός πιθανού σφάλματος, έχει προστεθεί στους ήδη κωδικοποιημένους χαρακτήρες ένα επιπλέον δυαδικό ψηφίο, το λεγόμενο *ψηφίο ισοτιμίας* (parity bit), με το οποίο αναγνωρίζεται εύκολα η παρουσία του πιθανού σφάλματος. Το ψηφίο ισοτιμίας τίθεται στην αρχή ή στο τέλος του κωδικοποιημένου χαρακτήρα και η τιμή του («0» ή «1») καθορίζεται από τον αριθμό των δυαδικών μονάδων του κωδικοποιημένου χαρακτήρα και από τον τρόπο ισοτιμίας που θέλουμε να εφαρμόσουμε (περιττή ή άρτια ισοτιμία). Εάν, για παράδειγμα, ο υπό μεταφορά κωδικοποιημένος χαρακτήρας έχει τέσσερις μονάδες και έχουμε περιττή ισοτιμία, το parity bit που προστίθεται έχει τιμή «1» έτσι, ώστε το σύνολο των μονάδων να είναι περιττό. Αντίθετα, εάν ο υπό μεταφορά κωδικοποιημένος χαρακτήρας έχει τέσσερις μονάδες και έχουμε άρτια ισοτιμία, τότε το parity bit που προστίθεται έχει τιμή «0» έτσι, ώστε το σύνολο των μονάδων να

παραμένει άρτιο. Είναι λογικό ότι τυχόν λάθος ανακαλύπτεται εύκολα, γιατί θα έχει σαν αποτέλεσμα την αλλαγή του αριθμού των μονάδων από περιττό σε άρτιο αριθμό. Στον πίνακα 2.5 δίνονται οι αριθμοί από 1 έως 10 κωδικοποιημένοι στο δυαδικό με ψηφία άρτιας και περιττής ισοτιμίας.

Δεκαδικός	Δυαδικός με άρτια ισοτιμία	Δυαδικός με περιττή ισοτιμία
0	0 0000	1 0000
1	1 0001	0 0001
2	1 0010	0 0010
3	0 0011	1 0011
4	1 0100	0 0100
5	0 0101	1 0101
6	0 0110	1 0110
7	1 0111	0 0111
8	1 1000	0 1000
9	0 1001	1 1001

Πίνακας 2.5. Παράσταση δυαδικών αριθμών με ψηφία ισοτιμίας

Στο Σχήμα 2.1 αποδίδεται ένα τυπικό διάγραμμα που δείχνει τον τρόπο ελέγχου της ύπαρξης πιθανών σφαλμάτων σε μια μεταδιδόμενη λέξη.



Σχήμα 2.1. Τυπικά διαγράμματα πομπού και δέκτη πληροφορίας 4 bit, κωδικοποιημένης σύμφωνα με τους κανόνες περιττής ή άρτιας ισοτιμίας

Τα ψηφία $X_1X_2X_3X_4$ της λέξης μεταδίδονται μαζί με ένα ακόμα ψηφίο ισοτιμίας, το PT (Parity Transmitter), το οποίο παράγεται από την επεξεργασία των αρχικών ψηφίων. Στη μεριά του δέκτη της πληροφορίας παράγεται πάλι με τον ίδιο τρόπο το ψηφίο ισοτιμίας PR (Parity Receiver), το οποίο συγκρίνεται με το PT. Αν η βαθμίδα της σύγκρισης βρει τα ψηφία αυτά διαφορετικά, ενεργοποιεί το σήμα PE (Parity Error) σαν ενδεικτικό της λανθασμένης λήψης.

Αξίζει να σημειώσουμε ότι η παραπάνω μέθοδος ανίχνευσης σφαλμάτων είναι σε θέση να σηματοδοτήσει εσφαλμένη λήψη, μόνο όταν το σύνολο των ψηφίων που έχουν υποστεί αλλοίωση εκφράζεται από έναν περιττό αριθμό.

2.5.2. Κώδικες διόρθωσης σφαλμάτων

Η κωδικοποίηση για τη διόρθωση σφαλμάτων είναι πιο ενδιαφέρουσα από την κωδικοποίηση για την ανίχνευση σφαλμάτων. Ο στόχος της διόρθωσης σφαλμάτων είναι η ανίχνευση και ο εντοπισμός σφαλμάτων στη μετάδοση της πληροφορίας. Όταν γίνει ο εντοπισμός η διόρθωση είναι τετριμμένη διαδικασία, καθώς απλά γίνεται αντιστροφή του ψηφίου.

Στις επικοινωνίες συχνά προτιμάται η ανίχνευση σφάλματος και η επανεκπομπή. Στις τηλεπικοινωνίες εντούτοις, λόγω της καθυστέρησης της μετάδοσης της πληροφορίας, αποστέλλονται πολλά πακέτα από τον εκπομπό, πριν φθάσει η πληροφορία για επανεκπομπή. Αυτό κάνει το χειρισμό των δεδομένων αρκετά πολύπλοκο. Επίσης η μετάδοση σε πραγματικό χρόνο αποκλείει την επανεκπομπή. Είναι, συνεπώς, απαραίτητο να μην υπάρχει σφάλμα. Έτσι απαιτείται ένα μεγαλύτερο εύρος ζώνης το κόστος του οποίου δεν είναι υψηλό για

τα πλεονάζοντα ψηφία ελέγχου (redundant check-bits). Υπάρχουν δύο κατηγορίες κωδίκων: Οι κώδικες Μπλοκ (π.χ. κώδικας Hamming) και οι συγκεραστικοί κώδικες (convolutional codes). (Nelson, et al., 1995; Klove, & Korzhik, 1995; Blahut, 2003; Lee, 2006; Wakerly, 2006; Rajaraman, & Adabala, 2015)

Κυκλικός Πλεονάζων Κώδικας Ελέγχου (CRC)

Οι *Κυκλικοί Πλεονάζοντες Κώδικες Ελέγχου* (cyclic redundancy check) περιλαμβάνουν τη διαίρεση της ακολουθίας των bytes των δεδομένων με μια σταθερά. Κάθε υπόλοιπο από αυτή τη διαίρεση είναι γραμμένο σε μορφή 2 CRC bytes, ή χαρακτήρων. Όταν τα δεδομένα διαβάζονται, αυτά τα 2 bytes αφαιρούνται από την ακολουθία των δεδομένων. Το αποτέλεσμα διαιρείται με την αρχική σταθερά. Αυτή η διαίρεση δίνει μηδενικό υπόλοιπο, αν τα δεδομένα δεν περιέχουν λάθη. Η μέθοδος CRC χρησιμοποιείται συνήθως για έλεγχο λαθών του δίσκου αποθήκευσης και των δικτύων δεδομένων. Σε ένα οδηγό δίσκου, κάθε μπλοκ δεδομένων (τυπικά 512 bytes) προστατεύεται από ένα CRC κώδικα, έτσι, ώστε σφάλμα σε ένα μπλοκ να μπορεί να ανιχνευθεί και, σε ορισμένους οδηγούς, να μπορεί να διορθωθεί. Σε ένα δίκτυο δεδομένων κάθε πακέτο δεδομένων τελειώνει με ένα CRC byte. Οι κώδικες CRC επιλέγονται και για τις δύο εφαρμογές λόγω των ιδιοτήτων ανίχνευσης καταγιστικών σφαλμάτων. Επιπλέον, από το να μπορούν να ανιχνεύσουν ένα σφάλμα, μπορούν να ανιχνεύσουν σφάλματα πολλαπλών bits, τα οποία είναι ομαδοποιημένα στο μπλοκ του δίσκου ή του πακέτου.

Κώδικες δύο διαστάσεων

Όταν μερικές δυαδικές λέξεις μεταδίδονται ή λαμβάνονται διαδοχικά, η προκύπτουσα συλλογή των bits μπορεί να θεωρηθεί ως ένα μπλοκ δεδομένων, το οποίο έχει σειρές και στήλες. Για παράδειγμα, έστω τέσσερις διαδοχικές οκτάμπιτες λέξεις δημιουργούν μια μορφή ενός μπλοκ 4x8. Έτσι μπορούμε να δημιουργήσουμε τα bits ισοτιμίας και ανά γραμμή και ανά στήλη. Αυτός ο κώδικας αναφέρεται σαν δισδιάστατος κώδικας ή κώδικας του μπλόκ ισοτιμίας ή κώδικας γινομένου. Είναι δυνατόν να ανιχνευθεί οποιοδήποτε απλό σφάλμα ανά λέξη δεδομένων ή διπλό σφάλμα ανά λέξη. (Σχήμα 2.2α.)

1	0	1	1	0	1	0	1	1
0	1	0	1	0	1	0	1	0
1	1	0	0	1	1	0	1	1
1	0	0	1	1	0	1	1	1
1	0	1	1	0	1	1	0	1

Σχήμα 2.2α. Διόρθωση απλού σφάλματος

Από τον παραπάνω πίνακα διαπιστώνουμε σφάλμα στη δεύτερη γραμμή και στην τρίτη κολώνα που δείχνει ότι το τρίτο bit της δεύτερης στήλης είναι εσφαλμένο, οπότε το διορθώνουμε σε 1.

1	0	0	0	0	1	0	1	1
0	1	0	1	0	1	0	1	0
1	1	0	0	1	1	0	1	1
1	0	0	1	1	0	1	1	1
1	0	0	0	0	1	1	0	1
1	0	1	1	0	1	1	0	1

Σχήμα 2.2β. Αδυναμία διόρθωσης διπλού σφάλματος

Στο σχήμα 2.2β απεικονίζεται η περίπτωση, όπου έχουμε δύο σφάλματα, π.χ. στην πρώτη γραμμή. Σε αυτή την περίπτωση το ψηφίο ισοτιμίας της πρώτης γραμμής δεν αναγνωρίζει κάποιο λάθος. Παράλληλα τα ψηφία ισοτιμίας των στηλών 3 και 4 αναγνωρίζουν λάθος. Σε αυτή την περίπτωση δεν μπορούμε να διορθώσουμε κάποιο λάθος, καθώς δε γνωρίζουμε τη γραμμή από την οποία προέρχεται το σφάλμα.

Κώδικες Αθροίσματος Ελέγχου (Check sum codes)

Οι κώδικες αθροίσματος ελέγχου χρησιμοποιούνται, για να ανιχνεύσουν λάθος σε μπλοκ δεδομένων. Το άθροισμα ελέγχου δημιουργείται από την άθροιση όλων των bytes των δεδομένων σε ένα μπλόκ, **αγνοώντας οποιοδήποτε κρατούμενο**. Είναι το αριθμητικό άθροισμα όλων των bytes δεδομένων. Το νούμερο που προκύπτει αποθηκεύεται μετά το μπλοκ δεδομένων. Όταν τα δεδομένα διαβάζονται, αυτό προστίθεται εκ νέου, και το άθροισμα συγκρίνεται με την αποθηκευμένη τιμή του ελέγχου αθροίσματος. Αν δεν υπάρχει σφάλμα, τότε τα δύο αθροίσματα είναι ταυτόσημα. Σε κάθε άλλη περίπτωση, στην περίπτωση λάθους, τα υπό μετάδοση δεδομένα αναμεταδίδονται.

Παράδειγμα: Να υπολογίσετε το άθροισμα ελέγχου των χαρακτήρων

1	1	0	1	1	0	0	1
0	0	1	1	1	0	1	0
1	1	1	0	0	0	1	1

Αθροίζοντας προκύπτει ο αριθμός 11100011 (αντιστοιχεί στον δεκαεξαδικό E3).

Κώδικες m από n

Σε αυτούς τους κώδικες η κάθε κωδική λέξη έχει m άσσους και όλα τα υπόλοιπα είναι μηδέν. Αυτός ο κωδικός έχει ελάχιστη απόσταση το 2, γιατί οποιαδήποτε αλλαγή σε ένα bit αλλάζει τον αριθμό των άσσων και, κατά συνέπεια, προκύπτει μια νέα λέξη που δεν είναι κωδική λέξη. Αυτός ο κώδικας είναι χρήσιμος για την ανίχνευση πολλαπλών σφαλμάτων μονής κατεύθυνσης. Σε αυτή την περίπτωση όλα τα σφάλματα αλλάζουν με την ίδια κατεύθυνση (τα 0 σε 1 και το 1 σε 0). Ως εκ τούτου, ο κώδικας χρησιμοποιείται για την ανίχνευση σφαλμάτων σε συστήματα στα οποία ο κυρίαρχος μηχανισμός σφαλμάτων τείνει να αλλάξει όλα τα bits προς την ίδια διεύθυνση.

2.5.3. Κώδικας Hamming

Δημιουργήθηκε το 1949 από τον Richard Hamming. Είναι ένας κώδικας ο οποίος μπορεί να διορθώσει ένα απλό λάθος και μπορεί να ανιχνεύσει οποιοδήποτε διπλό λάθος (δηλ. δύο απλά λάθη). Προσδιορίζεται με δύο αριθμούς N, K και αναφερόμαστε σε αυτόν ως *κώδικα Hamming (N,K)*. Οι *κώδικες Hamming* χρησιμοποιούν αριθμητική modulo-2. Η πρόσθεση σε αριθμητική modulo-2 είναι η λογική λειτουργία της πύλης exclusive OR (XOR) ο πίνακας της οποίας παρατίθεται τον *πίνακα 2.6*. (Shanmugam, 1979; Nashelsky, 1994; Klove, & Korzhik, 1995; Nelson, et al., 1995; Κοσσίδας, 1996; Holdsworth, & Woods, 2002; Givone, 2002; Blahut, 2003; Wakerly, 2006; Balabanian, & Carlson, 2007; Tervo, 2014; Tocci, et al., 2010)

Ο κώδικας Hamming χρησιμοποιείται στη μνήμη RAM του υπολογιστή, και είναι μια καλή επιλογή για τυχαία εμφανιζόμενα σφάλματα. Στην περίπτωση που έχουμε καταιγισμό λαθών, χρησιμοποιούνται άλλοι κώδικες. Η απλότητα του κώδικα Hamming είναι αυτή που τον κάνει εκπαιδευτικά χρήσιμο.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Πίνακας 2.6. Πίνακας αλήθειας της πύλης XOR

Ο κώδικας χρησιμοποιεί επιπλέον ψηφία για τον έλεγχο σφαλμάτων και εκτελεί τους ελέγχους με ειδικές εξισώσεις ελέγχου. Η εξίσωση ελέγχου της ισοτιμίας μιας ακολουθίας δυαδικών ψηφίων απλά προσθέτει τα ψηφία της ακολουθίας και ορίζει ότι το άθροισμα είναι άρτιο (για άρτια ισοτιμία) ή περιττό (για περιττή ισοτιμία). Εναλλακτικά λέμε ότι το άθροισμα είναι MOD 2, ή ότι το άθροισμα λαμβάνεται από την πράξη MOD 2 επί των ακεραίων. Η περαιτέρω θεωρητική ανάλυση του κώδικα ξεφεύγει από τα όρια του μαθήματος.

Ο κώδικας Hamming διαφέρει από όλους τους άλλους κώδικες αναγνώρισης σφαλμάτων, επειδή είναι σε θέση όχι μόνο να παρέχει ένδειξη για τη παρουσία ενός εσφαλμένου δυαδικού ψηφίου αλλά και να προσδιορίζει με ακρίβεια τη θέση του μέσα στη μεταδιδόμενη λέξη. Για να πετύχει τον ακριβή προσδιορισμό, επαυξάνει το σύνολο των ψηφίων m της προς μετάδοση λέξης, με ένα συγκεκριμένο αριθμό ψηφίων ισοτιμίας P_i που το πλήθος τους είναι ίσο με k .

Το μέγεθος του κώδικα Hamming

Ο κώδικας Hamming μπορεί να εξυπηρετήσει οποιονδήποτε αριθμό ψηφίων δεδομένων, αλλά είναι ενδιαφέρον να παραθέσουμε το μέγιστο αριθμό των απαιτούμενων ψηφίων ελέγχου. Ο πίνακας 2.7, που ακολουθεί, περιλαμβάνει το συνολικό αριθμό ψηφίων ελέγχου έτσι, ώστε να δίνει το πλήθος των ψηφίων του κώδικα Hamming, ο οποίος περιλαμβάνει και την ανίχνευση διπλού σφάλματος, καθώς επίσης, και την απόδοση του κώδικα Hamming, η οποία προκύπτει από τη διαίρεση του αριθμού των ψηφίων της λέξεως προς την ποσότητα (ολικό μέγεθος -1).

Έτσι για 64 bits ή 8 bytes, απαιτούνται 7 bytes δεδομένων (συν 1 bit) και χρησιμοποιείται 1 byte για τον έλεγχο των ψηφίων (πραγματικά μόνο 7 bits). Συνεπώς σ' ένα σύστημα που είναι επιρρεπές στα σφάλματα απαιτείται μόνο 1 επιπλέον byte στα συνολικά 8, ήτοι 12.5% επιπλέον για την ανίχνευση και διόρθωση σφαλμάτων).

Η χρήση του κώδικα αυτού αφορά κυρίως στη μετάδοση πληροφορίας από έναν υπολογιστή-πομπό σ' έναν άλλο, στις περιπτώσεις εκείνες που ο υπολογιστής-δέκτης δεν είναι σε θέση να ζητήσει επανάληψη της πληροφορίας που έχει αποσταλεί, αλλά οφείλει μόνος του να τη διορθώσει. Ακόμα χρησιμοποιείται κατά την εγγραφή/ανάγνωση δεδομένων σε μνήμες RAM.

Απαιτούμενα Ψηφία ελέγχου	Μέγιστος αριθμός Ψηφίων λέξεως	Ολικό Μέγεθος	Απόδοση (E)
3	1	4	0,33
4	4	8	0,57
5	11	16	0,73
6	26	32	0,84
7	57	64	0,90
8	120	128	0,94

Πίνακας 2.7. Απόδοση του κώδικα Hamming

Συνδυασμός ψηφίων C_i	Εσφαλμένο ψηφίο
000	κανένα
001	1 ^ο
010	2 ^ο
011	3 ^ο
100	4 ^ο
101	5 ^ο
110	6 ^ο
111	7 ^ο

Πίνακας 2.8. Θέση του εσφαλμένου ψηφίου στη μεταδιδόμενη λέξη

Κατά την εκπομπή της πληροφορίας, συγκεκριμένες ομάδες δυαδικών ψηφίων που αποτελούν την προς μετάδοση λέξη, οδηγούνται σε κυκλώματα παραγωγής ψηφίων ισοτιμίας. Τα παραγόμενα ψηφία αποστέλλονται, κατόπιν, μαζί με τα χρήσιμα, στον υπολογιστή - δέκτη. Ο δέκτης, έχοντας λάβει όλα τα ψηφία, παράγει και αυτός ψηφία ισοτιμίας, συγκρίνοντας τα ψηφία των ίδιων ομάδων που ελήφθησαν υπόψη κατά

την εκπομπή. Κατόπιν συγκρίνει τα ψηφία ισοτιμίας που ο ίδιος παρήγαγε με εκείνα που ήδη έχει λάβει. Αν το αποτέλεσμα της σύγκρισης οδηγήσει σε μια ακολουθία δυαδικών ψηφίων που αποτελείται από μηδενικά, τότε η λήψη έχει γίνει με επιτυχία. Σε αντίθετη περίπτωση, ο δυαδικός αριθμός που εκφράζει η ακολουθία είναι ενδεικτικός της θέσης του ψηφίου που έχει αλλοιωθεί. Στον πίνακα 2.8 φαίνεται η σημασία των ψηφίων C_i , που αποτελούν τα ψηφία εξόδου του συγκριτή. (Απόδοση του κώδικα Hamming, 2015)

Στο Σχήμα 2.3 φαίνονται, σε απλοποιημένη μορφή, τα διαγράμματα ενός πομπού πληροφορίας κωδικοποιημένης σύμφωνα με τον κώδικα Hamming και ενός δέκτη, που την αποκωδικοποιεί και τη διορθώνει. Τα εισερχόμενα ψηφία οδηγούνται κατά ομάδες στα κυκλώματα παραγωγής ψηφίων ισοτιμίας, των οποίων οι έξοδοι αναδεικνύουν τα ομώνυμα bit. Η επαυξημένη σε πλήθος ψηφίων πληροφορία φθάνει στο δέκτη, όπου οι έξοδοι C_i του συγκριτή ψηφίων ισοτιμίας διορθώνουν, εφόσον χρειάζεται, τη δυαδική πληροφορία.

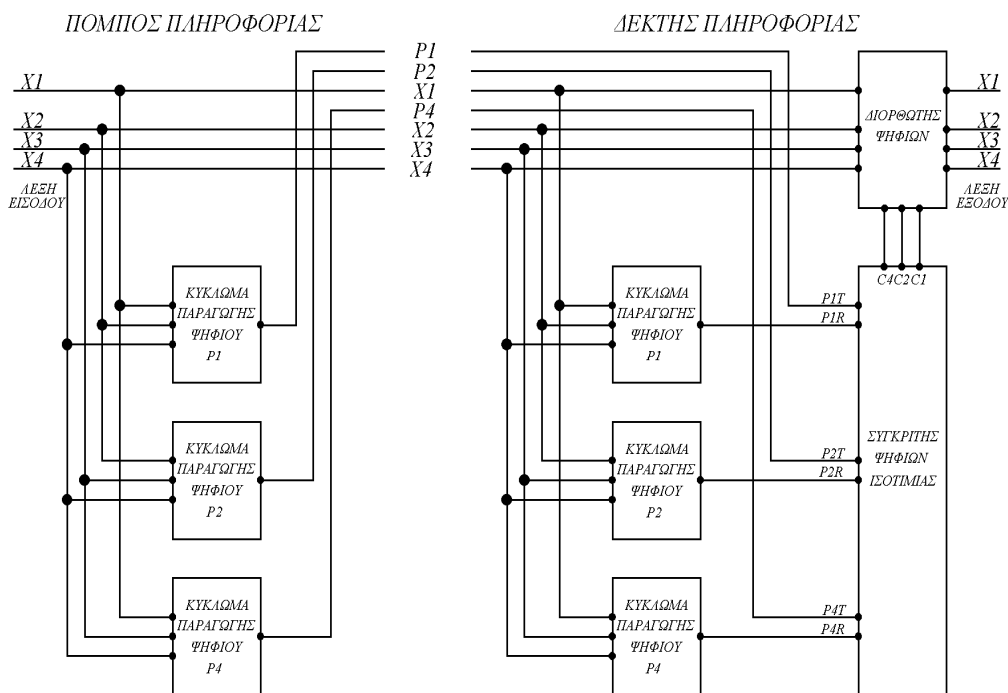
Επειδή το σύνολο k των ψηφίων ισοτιμίας P_i πρέπει να είναι σε θέση να προσδιορίζει με ακρίβεια τη θέση του λανθασμένου ψηφίου μέσα στη μεταδιδόμενη λέξη, που τώρα πια θα έχει μήκος $m+k$, θα πρέπει οι συνδυασμοί των k bit να σχηματίζουν αριθμούς που η τιμή τους θα μπορεί να προσδιορίζει με ακρίβεια τη θέση του λανθασμένου ψηφίου. Αν μάλιστα λάβουμε υπόψη πως ο συνδυασμός των k ψηφίων ισοτιμίας, που είναι ίσος με το 0, υποδηλώνει μια λέξη που μεταδόθηκε σωστά, τότε μπορούμε εύκολα να καταλήξουμε στη σχέση: $2^k - 1 \geq m + k$.

Τροποποιώντας τη σχέση αυτή, λαμβάνουμε την τελική που μας δίνει το πλήθος των k ψηφίων ισοτιμίας P_i που συνοδεύουν μια μεταδιδόμενη λέξη. Θα έχουμε επομένως: $2^k \geq m + k + 1$

m	k (ελάχιστο)
1	2
2 - 4	3
5 - 11	4
12 - 26	5

Πίνακας 2.9. Ελάχιστα ψηφία ισοτιμίας για λέξεις μήκους m

Στον παραπάνω πίνακα 2.9 σημειώνεται ο ελάχιστος αριθμός των ψηφίων ισοτιμίας k , για μεταδιδόμενες λέξεις μήκους m . (εύκολα προκύπτει από τον γενικό πίνακα 2.7).



Σχήμα 2.3. Τυπικά διαγράμματα πομπού και δέκτη πληροφορίας 4 bit, κωδικοποιημένη σύμφωνα με τον κώδικα Hamming

(Η ανάλυση των επιμέρους χρησιμοποιούμενων κυκλωμάτων θα γίνει σε επόμενο κεφάλαιο).

Τα ψηφία ισοτιμίας P_i παρεμβάλλονται στις θέσεις 1, 2, 4, 8, ..., 2^{k-1} , δηλαδή στις θέσεις εκείνες που προσδιορίζονται από τις δυνάμεις του 2. Η τιμή τους εξαρτάται από το αν έχουμε επιλέξει άρτια ή περιττή ισοτιμία. Τα ψηφία της χρήσιμης πληροφορίας $X_1, X_2, X_3, \dots, X_m$ καταλαμβάνουν τις υπόλοιπες θέσεις της μεταδιδόμενης πληροφορίας.

Στην περίπτωση όπου η πληροφορία που προορίζεται να μεταδοθεί έχει μήκος π.χ. $m=8$ bit, η μεταδιδόμενη λέξη θα έχει μήκος $m+k=12$ bit, αφού θα έχουν συμπεριληφθεί, σύμφωνα με τον πίνακα 1.18, τέσσερα ψηφία ισοτιμίας τα P_1, P_2, P_4 και P_8 , θα έχει δηλαδή τη μορφή:

Θέση ψηφίων	1	2	3	4	5	6	7	8	9	10	11	12
Σημασία ψηφίων	P_1	P_2	X_1	P_4	X_2	X_3	X_4	P_8	X_5	X_6	X_7	X_8

Τρόπος προσδιορισμού των bits του κώδικα Hamming

Στον κώδικα Hamming χρησιμοποιούνται επιπλέον ψηφία ισοτιμίας, που μας δίνουν τη δυνατότητα προσδιορισμού ενός απλού λάθους.

Για τη δημιουργία της κωδικής λέξης εργαζόμαστε ως εξής: Σημειώνουμε όλες τις θέσεις που αντιστοιχούν σε δυνάμεις του 2, οι οποίες αποτελούν και τα ψηφία ισοτιμίας (parity bits). Τα ψηφία αυτά θα γεμίσουν με δυαδικά ψηφία, τα οποία είναι αποτέλεσμα υπολογισμών με τη χρήση των δυαδικών ψηφίων της αρχικής λέξης. (θέσεις 1, 2, 4, 8, 16, 32, 64, κ.λπ.)

Όλες οι υπόλοιπες θέσεις θα χρησιμοποιηθούν για τα δεδομένα που πρόκειται να κωδικοποιηθούν. (θέσεις 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, κ.λπ.)

Θέση	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Ψηφίο 1	x		x		x		x		x		x		x		x		x
Ψηφίο 2		x	x			x	x			x	x			x	x		
Ψηφίο 4				x	x	x	x					x	x	x	x		
Ψηφίο 8								x	x	x	x	x	x	x	x		
Ψηφίο 16																x	x

Κάθε ψηφίο ισοτιμίας υπολογίζει την ισοτιμία κάποιων συγκεκριμένων δυαδικών ψηφίων της αρχικής, υπό κωδικοποίηση, λέξης. Η θέση του ψηφίου ισοτιμίας προσδιορίζει και την ακολουθία των υπό χρήση bits, ακολουθώντας τον παρακάτω αλγοριθμική διαδικασία. (για τις θέσεις 1 – 32).

Θέση 1: έλεγξε 1 bit, παράλειψε 1 bit, έλεγξε 1 bit, παράλειψε 1 bit κ.λπ. (1, 3, 5, 7, 9, 11, 13, 15,...)

Θέση 2: έλεγξε 2 bits, παράλειψε 2 bits, έλεγξε 2 bits, παράλειψε 2 bits, κ.λπ. (2, 3, 6, 7, 10, 11, 14, 15, ...)

Θέση 4: έλεγξε 4 bits, παράλειψε 4 bits, έλεγξε 4 bits, παράλειψε 4 bits, κ.λπ. (4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23,...)

Θέση 8: έλεγξε 8 bits, παράλειψε 8 bits, έλεγξε 8 bits, παράλειψε 8 bits, κ.λπ. (8-15, 24-31, 40-47,...)

Θέση 16: έλεγξε 16 bits, παράλειψε 16 bits, έλεγξε 16 bits, παράλειψε 16 bits, κ.λπ. (16-31, 48-63, 80-95,...)

Θέση 32: έλεγξε 32 bits, παράλειψε 32 bits, έλεγξε 32 bits, παράλειψε 32 bits, κ.λπ. (32-63, 96-127, 160-191, ...)

Θέστε το parity bit σε 1, εάν το συνολικό άθροισμα των άσσων στις υπό έλεγχο θέσεις είναι περιττό.

Θέστε το parity bit σε 0, εάν το συνολικό άθροισμα των άσσων στις υπό έλεγχο θέσεις είναι άρτιο.

Παρατηρήστε λοιπόν ότι ξεκινάμε πάντα από το ψηφίο ισοτιμίας του οποίου θέλουμε να προσδιορίσουμε την τιμή. Η ομάδα ψηφίων που παράγει το ψηφίο ισοτιμίας P_1 αποτελείται από τα bit που βρίσκονται στις θέσεις 3, 5, 7, 9, 11 της λέξης. (Το ψηφίο ή τα ψηφία τα οποία ανήκουν στις θέσεις των ψηφίων ισοτιμίας δεν λαμβάνονται υπόψη). Η αντίστοιχη ομάδα που δίνει το P_2 αποτελείται από τα bit των θέσεων 3, 6, 7, 10, 11, ενώ οι ομάδες που δίνουν τα P_4 και P_8 είναι αυτές που περιλαμβάνουν τα bit των θέσεων 5, 6, 7, 12 και 9, 10, 11, 12 αντίστοιχα.

Παράδειγμα 1ο:

Ένα εύλογο ερώτημα είναι το γιατί επιλέγουμε τα ψηφία αυτά - Τα τρία πρώτα ψηφία ισοτιμίας (1, 2, 4) σχετίζονται με τα ψηφία δεδομένων (3,5,6,7). Στα σχήματα 2.3α και 2.3β κάθε αλληλεπικαλυπτόμενος κύκλος

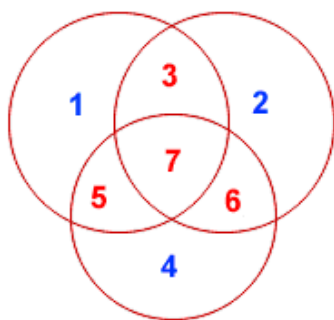
αντιστοιχεί σε ένα ψηφίο ισοτιμίας και ορίζει τα τέσσερα ψηφία, που συνεισφέρουν στον υπολογισμό της ισοτιμίας. Για παράδειγμα, το ψηφίο δεδομένων 3 συνεισφέρει στα ψηφία ισοτιμίας 1 και 2. Κάθε κύκλος (ψηφίο ισοτιμίας-parity bit) συμπεριλαμβάνει ένα σύνολο από τέσσερα ψηφία, και κάθε κύκλος πρέπει να έχει άρτια ισοτιμία. Λαμβάνοντας υπόψη μας τα τέσσερα ψηφία των δεδομένων, μπορούμε πολύ εύκολα να επιλέξουμε τα τρία ψηφία ισοτιμίας ώστε να εξασφαλισθεί η άρτια ισοτιμία.

Εύκολα διαπιστώνουμε ότι η αλλαγή ενός οποιουδήποτε δυαδικού ψηφίου από τα 7 επηρεάζει τα τρία ψηφία ισοτιμίας. Η τροποποίηση του 7^{ου} δυαδικού ψηφίου επηρεάζει όλα τα parity bits, ενώ σφάλμα στο 6^ο δυαδικό ψηφίο επηρεάζει μόνο τα ψηφία ισοτιμίας 2 και 4, και ένα σφάλμα ενός parity bit επηρεάζει μόνο αυτό το δυαδικό ψηφίο. Η θέση οποιουδήποτε απλού σφάλματος προσδιορίζεται από τον έλεγχο των τριών κύκλων ισοτιμίας.

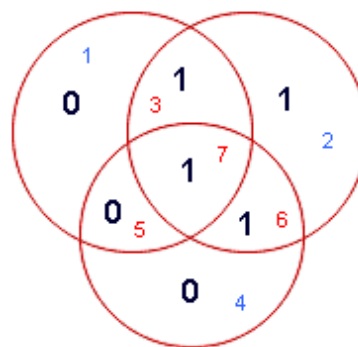
Το μήνυμα 1011 θα μπορούσε να σταλεί ως 0110011, καθώς:

1	2	3	4	5	6	7	
0	1	1	0	0	1	1	7-ψήφια κωδική λέξη
0	-	1	-	0	-	1	Άρτια ισοτιμία
-	1	1	-	-	1	1	Άρτια ισοτιμία
-	-	-	0	0	1	1	Άρτια ισοτιμία

Όταν τα τέσσερα αυτά ψηφία εισαχθούν στους κύκλους ισοτιμίας, επιβεβαιώνεται ότι η επιλογή των τριών αυτών ψηφίων ισοτιμίας βεβαιώνει ότι η ισοτιμία σε κάθε κύκλο είναι άρτια, όπως φαίνεται από τα σχήματα 2.4α και 2.4β.



Σχήμα 2.4α.



Σχήμα 2.4β.

Παράδειγμα 2ο:

Εστω η ακολουθία δεδομένων: 10011010. Να βρεθεί η προς αποστολή λέξη η οποία κωδικοποιείται με τον κώδικα Hamming.

Θέση	1	2	3	4	5	6	7	8	9	10	11	12	Ισοτιμία
Ψηφίο / (δεδομένα)	P1	P2	1	P4	0	0	1	P8	1	0	1	0	
Έλεγχος: 1	x		x		x		x		x		x		0
Έλεγχος: 2		x	x			x	x			x	x		1
Έλεγχος: 4				x	x	x	x					x	1
Έλεγχος: 8								x	x	x	x	x	0
Τελική Κωδική Λέξη	0	1	1	1	0	0	1	0	1	0	1	0	

Άρα η τελική κωδική λέξη που θα αποσταλεί είναι : **0111 0010 1010**

Παράδειγμα 3ο:

Ας θεωρήσουμε ότι θέλουμε να μεταδώσουμε μια πληροφορία 4 bit, έστω τον δυαδικό αριθμό **1010**. Θα θεωρήσουμε ακόμα ότι τα εμπλεκόμενα ψηφία ισοτιμίας αποκτούν τιμή σύμφωνα με τους κανόνες της άρτιας ισοτιμίας.

Τα ψηφία της πληροφορίας είναι 4, δηλαδή $m=4$, οπότε σύμφωνα με τον πίνακα 1.18 έχουμε $k=3$. Η μεταδιδόμενη λέξη θα έχει, επομένως, τη μορφή:

Θέση ψηφίων	1	2	3	4	5	6	7
Σημασία ψηφίων	P_1	P_2	X_1	P_4	X_2	X_3	X_4
Τιμή ψηφίων	-	-	1	-	0	1	0

Η τιμή των ψηφίων P_i υπολογίζεται αμέσως παρακάτω σύμφωνα με τους κανόνες της άρτιας ισοτιμίας. Για το ψηφίο ισοτιμίας P_1 λαμβάνουμε υπόψη τις τιμές των ψηφίων 3, 5 και 7. Έχουμε επομένως:

Θέση ψηφίων	3	5	7
Σημασία ψηφίων	X_1	X_2	X_4
Τιμή ψηφίων	1	0	0

από όπου προκύπτει ότι $P_1 = 1$.

Για το ψηφίο ισοτιμίας P_2 λαμβάνουμε υπόψη τις τιμές των ψηφίων 3, 6 και 7. Έχουμε επομένως:

Θέση ψηφίων	3	6	7
Σημασία ψηφίων	X_1	X_3	X_4
Τιμή ψηφίων	1	1	0

από όπου προκύπτει ότι $P_2 = 0$.

Για το ψηφίο ισοτιμίας P_4 λαμβάνουμε υπόψη τις τιμές των ψηφίων 5, 6 και 7. Έχουμε επομένως:

Θέση ψηφίων	5	6	7
Σημασία ψηφίων	X_2	X_3	X_4
Τιμή ψηφίων	0	1	0

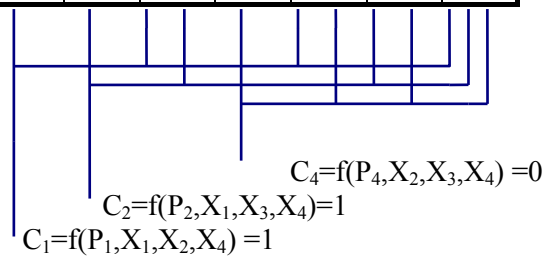
από όπου προκύπτει ότι $P_4 = 1$.

Παρεμβάλλοντας στη λέξη τα ψηφία ισοτιμίας, που υπολογίστηκαν με τον παραπάνω τρόπο, καταλήγουμε στο συμπέρασμα ότι η προς μετάδοση λέξη θα είναι η **1011010**.

Προσδιορισμός και διόρθωση λάθους

Αν, κατά τη μετάδοση, συμβεί σφάλμα στη θέση 3, η λέξη θα αποκτήσει τιμή ίση με 1001010. Η έξοδος του κυκλώματος λήψης θα αναδείξει την πληροφορία:

Θέση ψηφίων	1	2	3	4	5	6	7
Σημασία ψηφίων	P_1	P_2	X_1	P_4	X_2	X_3	X_4
Τιμή ψηφίων	1	0	0	1	0	1	0



Υποβάλλοντας τη ληφθείσα ακολουθία δυαδικών ψηφίων σε διαδικασίες παραγωγής ψηφίων ισοτιμίας, σύμφωνα με τους ίδιους κανόνες που χρησιμοποιήθηκαν για την παραγωγή των ίδιων ψηφίων κατά την εκπομπή, και συγκρίνοντας κατόπιν τα ψηφία αυτά με τα ψηφία ισοτιμίας που έχουν ήδη ληφθεί, διαπιστώνουμε ότι οι έξοδοι του κυκλώματος σύγκρισης παράγουν τον συνδυασμό $C_4C_2C_1 = 011$ που είναι διαφορετι-

κός από τον $C_4C_2C_1 = 000$, ο οποίος εκφράζει τη λήψη χωρίς σφάλματα. Η τιμή του συνδυασμού υποδηλώνει και τη θέση του λανθασμένου ψηφίου, που, στην περίπτωση μας, είναι το τρίτο ($011_2 = 3_{10}$).

Παράδειγμα 4ο:

Ας θεωρήσουμε ότι θέλουμε να μεταδώσουμε μια πληροφορία 8 bit, έστω το δυαδικό αριθμό **10101101**, χρησιμοποιώντας ψηφία άρτιας ισοτιμίας.

Τα ψηφία της πληροφορίας είναι 8, οπότε $m=8$ και σύμφωνα με τον πίνακα 1.18, $k=4$. Η μεταδιδόμενη λέξη θα έχει, επομένως, τη μορφή:

Θέση ψηφίων	1	2	3	4	5	6	7	8	9	10	11	12
Σημασία ψηφίων	P_1	P_2	X_1	P_4	X_2	X_3	X_4	P_8	X_5	X_6	X_7	X_8
Τιμή ψηφίων			1		0	1	0		1	1	0	1

Για το ψηφίο ισοτιμίας P_1 λαμβάνουμε υπόψη τις τιμές των ψηφίων 3, 5, 7, 9 και 11. Έχουμε επομένως:

Θέση ψηφίων	3	5	7	9	11
Σημασία ψηφίων	X_1	X_2	X_4	X_5	X_7
Τιμή ψηφίων	1	0	0	1	0

από όπου προκύπτει ότι $P_1 = 0$.

Για το ψηφίο ισοτιμίας P_2 λαμβάνουμε υπόψη τις τιμές των ψηφίων 3, 6, 7, 10 και 11. Έχουμε επομένως:

Θέση ψηφίων	3	6	7	10	11
Σημασία ψηφίων	X_1	X_3	X_4	X_6	X_7
Τιμή ψηφίων	1	1	0	1	0

από όπου προκύπτει ότι $P_2 = 1$.

Για το ψηφίο ισοτιμίας P_4 λαμβάνουμε υπόψη τις τιμές των ψηφίων 5, 6, 7 και 12. Έχουμε επομένως:

Θέση ψηφίων	5	6	7	12
Σημασία ψηφίων	X_2	X_3	X_4	X_8
Τιμή ψηφίων	0	1	0	1

από όπου προκύπτει ότι $P_4 = 0$.

Για το ψηφίο ισοτιμίας P_8 λαμβάνουμε υπόψη τις τιμές των ψηφίων 9, 10, 11 και 12. Έχουμε επομένως:

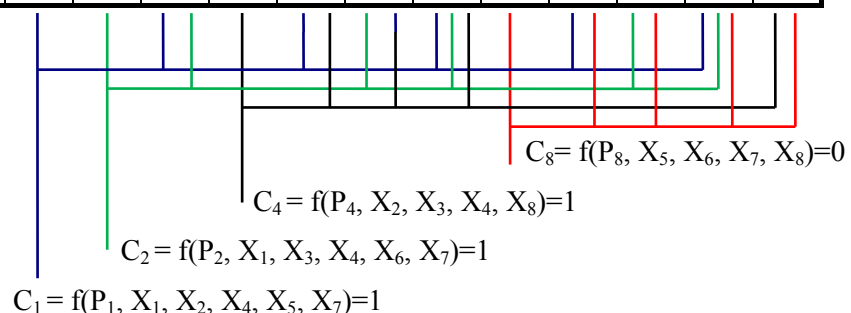
Θέση ψηφίων	9	10	11	12
Σημασία ψηφίων	X_5	X_6	X_7	X_8
Τιμή ψηφίων	1	1	0	1

από όπου προκύπτει ότι $P_8 = 1$.

Παρεμβάλλοντας στη λέξη τα ψηφία ισοτιμίας που υπολογίσθηκαν με τον παραπάνω τρόπο, καταλήγουμε στο συμπέρασμα ότι η προς μετάδοση λέξη θα είναι η **011001011101**.

Προσδιορισμός και διόρθωση λάθους

Θέση ψηφίων	1	2	3	4	5	6	7	8	9	10	11	12
Σημασία ψηφίων	P_1	P_2	X_1	P_4	X_2	X_3	X_4	P_8	X_5	X_6	X_7	X_8
Τιμή ψηφίων	0	1	1	0	0	1	1	1	1	1	0	1



Αν κατά τη μετάδοση συμβεί σφάλμα στη θέση 7, η λέξη θα αποκτήσει τιμή ίση με: 011001111101. Η έξοδος του κυκλώματος λήψης θα αναδείξει τότε την πληροφορία.

Υποβάλλοντας τα ληφθέντα δυαδικά ψηφία σε επεξεργασίες ίδιες μ' αυτές που αναφέρθηκαν στο προηγούμενο παράδειγμα, το κύκλωμα ελέγχου εγκυρότητας της πληροφορίας θα δώσει στις εξόδους του το συνδυασμό $C_8C_4C_2C_1 = 0111$, ο οποίος είναι διαφορετικός από το συνδυασμό $C_8C_4C_2C_1 = 0000$, που εκφράζει την ορθά ληφθείσα πληροφορία. Η τιμή του συνδυασμού υποδηλώνει και τη θέση του λανθασμένου ψηφίου, που στη συγκεκριμένη περίπτωση είναι το τρίτο ($0111_2 = 7_{10}$).

Ο κώδικας Hamming μας παρέχει: Διόρθωση ενός ψηφίου, ανίχνευση δύο εσφαλμένων ψηφίων, κόστος 3 ψηφίων, που πρέπει να προστεθούν στα 4 ψηφία του μηνύματος.

Η δυνατότητα της διόρθωσης ενός σφάλματος έχει κόστος, το οποίο είναι μικρότερο από το κόστος της επανεκπομπής του μηνύματος.

2.6. Αριθμητική κινητής υποδιαστολής και το πρότυπο IEEE-754

Κύριος στόχος μας είναι να μελετήσουμε τον τρόπο με τον οποίο γίνονται οι πράξεις κινητής υποδιαστολής. Αρχικά θα μελετήσουμε τους αριθμούς αυτούς και, στη συνέχεια, θα δούμε τα μοντέλα εκτέλεσης πράξεων. (Predko, 2005; Kahan, 2008; IEEE, 2008; Pritchard, 2015; Rajaraman, & Adabala, 2015)

2.6.1. Η ιστορία των πράξεων κινητής υποδιαστολής

Υπάρχουν διάφοροι τρόποι, για να αναπαρασταθούν οι πραγματικοί αριθμοί στους υπολογιστές. Ένας τρόπος είναι η κινητή υποδιαστολή. Η κινητή υποδιαστολή δεν είναι προκαθορισμένη, αλλά μεταβάλλεται και προσαρμόζεται αυτόματα στις μεταβαλλόμενες απαιτήσεις των υπολογισμών. Έτσι η θέση της υποδιαστολής μετακινείται (floats) και οι αριθμοί αυτοί ονομάζονται *αριθμοί κινητής υποδιαστολής*. Αυτό τους ξεχωρίζει από τους αριθμούς σταθερής υποδιαστολής, στους οποίους η υποδιαστολή βρίσκεται πάντα στην ίδια θέση.

Στις πρώτες μηχανές οι προγραμματιστές έπρεπε να γράψουν λογισμικό για την εκτέλεση των λειτουργιών της κινητής υποδιαστολής. Έτσι κάθε μηχανή είχε τον δικό της τρόπο υλοποίησης της αριθμητικής Κινητής Υποδιαστολής (Κ.Υ.), τόσο στο υλικό όσο και στο λογισμικό. Ήταν λοιπόν αδύνατο να γραφούν μεταβιβάσιμα προγράμματα, τα οποία θα οδηγούσαν στα ίδια αποτελέσματα σε διαφορετικά συστήματα.

Το πρότυπο IEEE-754 υιοθετήθηκε το 1985 και καθόριζε στους κατασκευαστές των Κ.Μ.Ε. τον τρόπο υλοποίησης των υπολογισμών της κινητής υποδιαστολής.

Έχοντας ένα πρότυπο, τουλάχιστον έχουμε τη διαβεβαίωση ότι όλες οι συμβατές μηχανές για το ίδιο πρόγραμμα θα παράξουν την ίδια έξοδο.

2.6.2. Υλικό Κινητής Υποδιαστολής

Όταν εισήχθη στους μικροεπεξεργαστές η κινητή υποδιαστολή, δεν υπήρχαν αρκετά τρανζίστορ στη ψηφίδα (on chip), ώστε να είναι δυνατή η υλοποίηση της. Έτσι οι κατασκευαστές των ΚΜΕ παρείχαν ξεχωριστούς συνεπεξεργαστές για τις απαραίτητες λειτουργίες της Κινητής Υποδιαστολής, π.χ., ο 8087 της Intel. Αποτέλεσμα ήταν πολλές ISA να χρησιμοποιούν ξεχωριστούς καταχωρητές για την Κινητή Υποδιαστολή.

Ο προϋπολογισμός του κόστους των νέων τρανζίστορ έδωσε τη δυνατότητα στη μονάδα Κ.Υ. να είναι πλέον στη ψηφίδα. Έτσι σήμερα οι ΚΜΕ έχουν ενσωματωμένες μονάδες κινητής υποδιαστολής. Ο 486 της Intel ήταν ο πρώτος x86 με ενσωματωμένη τη μονάδα Κ.Υ. (1989)

Ακόμα και οι νεότερες ISA έχουν ξεχωριστά αρχεία καταχωρητών για την κινητή υποδιαστολή.

2.6.3. Αριθμοί κινητής υποδιαστολής -Εισαγωγή

Μέχρι τώρα έχουμε ασχοληθεί, κατά κανόνα, με ακέραιες τιμές. Για τους πραγματικούς αριθμούς, δηλαδή για αριθμούς που περιέχουν και ακέραιο μέρος και μέρος μετά την υποδιαστολή, χρειαζόμαστε την αναπαράσταση της κινητής υποδιαστολής π.χ. οι αριθμοί: 3.14159 , 1.563×10^{28}

[Σημείωση: Για να παραστήσουμε έναν αριθμό σε δεκαεξαδική μορφή, χρησιμοποιούμε συνήθως είτε το H είτε το x: Έτσι, π.χ., ο αριθμός 0 γράφεται είτε ως 00H είτε ως 0x00.

Για τους δυαδικούς αντίστοιχα χρησιμοποιούμε την παράσταση b '001' (binary), και για τους δεκαδικούς την d '17' (decimal)].

Έστω ο αριθμός 0.5467×2^{28} . Στον αριθμό αυτό η ποσότητα 0.5467 αντιστοιχεί στα ψηφία μετά την υποδιαστολή (Mantissa), ο αριθμός 28 είναι ο εκθέτης και τέλος το 2 αντιστοιχεί στη βάση. Οι αριθμοί κινητής υποδιαστολής ορίζονται συναρτήσει:

1. των ψηφίων μετά την υποδιαστολή (Mantissa),

- Πώς αναπαρίστανται οι αριθμοί Κ.Υ.
- Ποια είναι τα όρια των αριθμών Κ.Υ.
- Την πρόσθεση και τον πολλαπλασιασμό αριθμών Κ.Υ.

2.6.8. Μετατροπή αριθμών μεταξύ της IEEE-754 και του δεκαδικού

s	e	f
-----	-----	-----

Η δεκαδική τιμή ενός IEEE αριθμού δίνεται από την εξίσωση: $(1 - 2s) * (1 + f) * 2^{e-bias}$

Τα πεδία s , f και e θεωρούμε ότι είναι στο δεκαδικό σύστημα. Η ποσότητα $(1 - 2s)$ είναι 1 ή -1, εξαρτώμενη από το εάν το ψηφίο του προσήμου είναι 0 ή 1. Τέλος στο κλασματικό πεδίο του f , προσθέτουμε έναν υπονοούμενο άσσο «1». Η μετατόπιση-offset(bias) είναι είτε 127 για αριθμούς απλής ακριβείας είτε 1023 για διπλής ακριβείας.

Παράδειγμα 5ο:

1. Μετατροπή από IEEE-754 σε δεκαδικό αριθμό

a) (Πρώτη μέθοδος επίλυσης)

Έστω ότι θέλουμε να βρούμε τη δεκαδική τιμή του IEEE-754 αριθμού που ακολουθεί:

1 01111100 1100000000000000000000

Πρώτα μετατρέπουμε κάθε πεδίο στον αντίστοιχο δεκαδικό αριθμό.

Το ψηφίο προσήμου s είναι 1.

Το πεδίο e περιέχει την τιμή $01111100 = 124_{10}$.

Η mantissa είναι $0.11000... = 0.75_{10} \cdot (\frac{1}{2} + \frac{1}{4})$

Κατόπιν απλά τοποθετούμε αυτές τις δεκαδικές τιμές των s , e και f στην εξίσωσή μας:

$$(1 - 2s) * (1 + f) * 2^{e-bias}$$

$$\text{Αυτό μας δίνει } (1 - 2) * (1 + 0.75) * 2^{124-127} = (-1) * (1.75 * 2^{-3}) = -0.21875.$$

b) (Δεύτερη μέθοδος επίλυσης)

Έστω ότι θέλουμε να βρούμε τη δεκαδική τιμή του ακόλουθου IEEE-754 αριθμού απλής ακριβείας.

0 10001000 100100100000000000000000

Θα χρησιμοποιήσουμε την εξίσωση $X = (-1)^S * 2^{E-127} * (1.M)$

(Το .M συμβολίζει το μέρος μετά την υποδιαστολή και όχι κάποιο γινόμενο) όπου: $S = 0$

$$E = 10001000_2 = 136_{10}$$

$$1.M = 1.100100100000000000000000 = 1 + 2^{-1} + 2^{-4} + 2^{-7} = 1.5703125$$

$$\text{Έτσι τελικά: } X = (-1)^0 * 2^{136-127} * 1.5703125 = 804 = 8.04 * 10^2.$$

2. Μετατροπή δεκαδικού αριθμού σε IEEE-754

Ποια είναι η αναπαράσταση του αριθμού 347.625 σε απλής ακριβείας;

Αρχικά μετατρέπουμε το δεκαδικό αριθμό σε δυαδικό: $347.625 = 101011011.101_2$.

Κανονικοποιούμε το δυαδικό αριθμό, ολισθαίνοντας τη δεκαδική τελεία, μέχρι να υπάρχει αριστερά της μόνο ένας 1: $101011011.101 * 2^0 = 1.01011011101 * 2^8$

Τα ψηφία στα δεξιά της υποδιαστολής αποτελούν το κλασματικό πεδίο f .

Ο αριθμός των ολισθήσεων που εκτελέσαμε μας δίνει τον εκθέτη. Το πεδίο e θα πρέπει να περιέχει την τιμή: εκθέτης + 127.

Ψηφίο προσήμου: 0, εάν είναι θετικός, και 1 εάν είναι αρνητικός.

(Να επαληθευθεί το αποτέλεσμα μετατρέποντάς το πάλι στο δεκαδικό)

Ειδικές περιπτώσεις

Ο μικρότερος και ο μεγαλύτερος εκθέτης είναι αντίστοιχα ο $e=00000000$ και ο $e=11111111$ (και τα διπλής ακριβείας αντίγραφα τους) και είναι δεσμευμένοι για ειδικές περιπτώσεις.

Εάν η mantissa είναι πάντα $(1 + f)$, τότε πώς μπορεί να αναπαρασταθεί το 0;

Το κλασματικό πεδίο f θα πρέπει να είναι $0000...0000$.

Το εκθετικό πεδίο e περιέχει την τιμή 00000000 .

Με προσημασμένα μεγέθη, υπάρχουν δύο μηδενικά: +0.0 και -0.0.

Όσον αφορά στο άπειρο υπάρχουν αναπαραστάσεις του θετικού και του αρνητικού απείρου, οι οποίες θα μπορούσαν να βοηθήσουν μερικές φορές στην περίπτωση της υπερχειλίσισης.

Το τμήμα f είναι τότε 0000...0000.

Το εκθετικό τμήμα e ορίζεται σαν 11111111.

Τέλος υπάρχει μία ειδική τιμή «όχι ένας αριθμός», η οποία μπορεί να χειρισθεί μερικές περιπτώσεις λαθών ή μη επιτρεπτών λειτουργιών όπως η πράξη 0.0/0.0.

Το κλασματικό πεδίο f τίθεται τότε σε οποιαδήποτε μη μηδενική τιμή.

Ο εκθέτης e θα περιέχει την τιμή 11111111.

2.6.9. Όρια της αναπαράστασης IEEE-754

Υπάρχουν επίσης μερικοί απλοί δεκαδικοί αριθμοί οι οποίοι δεν μπορούν να αναπαρασταθούν ακριβώς στο δυαδικό όπως π.χ. το 0.1010, όπου $0.1010 = 0.0001100110011\dots_{(2)}$

Προβλήματα της αναπαράστασης IEEE-754 - Ο δεκαδικός αριθμός 0.10

Μια μελέτη προσδιόρισε την ακριβή αναπαράσταση, του 0.10 σε δυαδική μορφή. Η 24-bit δυαδική αναπαράσταση του 0.10 στην πραγματικότητα αντιστοιχεί στον αριθμό 0.099999904632568359375, ο οποίος απέχει του 0.10 κατά 0.000000095367431640625. Αυτή η διαφορά δε φαίνεται μεγάλη, αλλά μετά από 100 ώρες ο χρόνος τελειώνει, και είναι μικρότερος κατά 0.34 sec. Η χρήση του λοιπόν σε ηλεκτρονικά συστήματα, π.χ. αναχαίτησης, δίνει σφάλμα στον προσδιορισμό μεγεθών που σχετίζονται με το χρόνο κίνησης του συστήματος. (GAO report 1992)

2.6.10. Πράξεις αριθμών κινητής υποδιαστολής (πρόσθεση και πολλαπλασιασμός)

A. Πρόσθεση αριθμών κινητής υποδιαστολής

1. Εξισώνουμε τους εκθέτες.

Ο τελεστέος με το μικρότερο εκθέτη πρέπει να ξαναγραφεί, αυξάνοντας τον εκθέτη του και ολισθαίνοντας την υποδιαστολή προς τα αριστερά:

$2.710 * 10^{-1} = 0.0271x 10^1$ Με τέσσερα σημαντικά ψηφία, ο αριθμός στρογγυλοποιείται στον: $0.027x 10^1$.

Αυτό έχει σαν αποτέλεσμα την απώλεια των λιγότερο σημαντικών ψηφίων, στην περίπτωσή μας το τελευταίο 1. Αλλά, ξαναγράφοντας τον αριθμό με μεγαλύτερο εκθέτη, αυτό μπορεί να έχει σαν αποτέλεσμα την απώλεια των περισσότερων σημαντικών ψηφίων, το οποίο είναι χειρότερο.

2. Προσθέτουμε τις mantissa.

$$\begin{array}{r} 9.989 * 10^1 \\ + 0.027 * 10^1 \\ \hline 10.016 * 10^1 \end{array}$$

Αναπαράσταση του αποτελέσματος (Βήματα 3-5)

3. Κανονικοποιούμε το αποτέλεσμα, εάν είναι απαραίτητο.

$$10.016 * 10^1 = 1.0016 * 10^2$$

Αυτό το βήμα μπορεί να έχει σαν αποτέλεσμα η υποδιαστολή να ολισθήσει είτε προς τα αριστερά είτε προς τα δεξιά, ο δε εκθέτης είτε να αυξηθεί είτε να ελαττωθεί.

4. Στρογγυλοποιούμε τον αριθμό, εάν είναι απαραίτητο.

$$\text{Ο αριθμός } 1.0016 * 10^2 \text{ στρογγυλοποιείται στον } 1.002 * 10^2$$

5. Επανάληψη του βήματος 3 εάν το αποτέλεσμα δεν είναι κανονικοποιημένο.

Στο παράδειγμά μας, το βήμα αυτό δεν απαιτείται, αλλά είναι πιθανόν να χρειάζεται στρογγυλοποίηση σε όλα τα ψηφία π.χ., η στρογγυλοποίηση του 9.9995 οδηγεί στον αριθμό 10.000.

Το αποτέλεσμα μας είναι $1.002 * 10^2$ ή 100.2. Το σωστό αποτέλεσμα είναι 100.16, και έτσι έχουμε την απάντηση με τέσσερα σημαντικά ψηφία, αλλά υπάρχει ένα μικρό λάθος.

Παράδειγμα 6ο:

A. Πρόσθεση

Για να αντιληφθούμε τις λειτουργίες της κινητής υποδιαστολής, θα παραθέσουμε ένα παράδειγμα πρόσθεσης. Για λόγους απλότητας, θα χρησιμοποιήσουμε την αναπαράσταση με βάση το 10.

Θα κάνουμε την υπόθεση ότι η mantissa έχει τέσσερα ψηφία, και ο εκθέτης ένα ψηφίο. Έστω λοιπόν η πρόσθεση στο δεκαδικό σύστημα: $99.89 + 0.272 = 100.162$

Οι τελεστέοι σαν κανονικοποιημένοι αριθμοί γράφονται ως εξής: $9.989 * 10^1$ και $2.720 * 10^{-1}$

B. Πολλαπλασιασμός αριθμών κινητής υποδιαστολής

Έστω ότι θέλουμε να πολλαπλασιάσουμε τους δεκαδικούς αριθμούς: 99.89 και 0.272, δηλαδή τους κανονικοποιημένους αριθμούς, $9.989 * 10^1$ και $2.720 * 10^{-1}$.

Για να πολλαπλασιάσουμε δύο τιμές κινητής υποδιαστολής, πρώτα πολλαπλασιάζουμε τα μεγέθη και κατόπιν προσθέτουμε τους εκθέτες.

$$\begin{array}{r} 9.989 * 10^1 \\ * 2.720 * 10^{-1} \\ \hline 27.170 * 10^0 \end{array}$$

Κατόπιν μπορούμε να στρογγυλοποιήσουμε και να κανονικοποιήσουμε το αποτέλεσμα, παίρνοντας $2.717 * 10^1$. Το πρόσημο του γινομένου είναι η πράξη XOR των προσήμων των τελεστέων.

Εάν οι δύο αριθμοί έχουν το ίδιο πρόσημο, τότε το γινόμενό τους είναι θετικό, ενώ, εάν οι δύο αριθμοί έχουν διαφορετικό πρόσημο, τότε το γινόμενό τους είναι αρνητικό.

$$0 \oplus 0 = 0 \quad 0 \oplus 1 = 1 \quad 1 \oplus 0 = 1 \quad 1 \oplus 1 = 0$$

Αυτό είναι ένα από τα κύρια πλεονεκτήματα της χρήσης προσημασμένων μεγεθών.

2.7. Ασκήσεις – Ερωτήσεις

ΕΡΩΤΗΣΕΙΣ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

1. Στον κώδικα Gray, κάθε αριθμός είναι 3 φορές μεγαλύτερος από τη δυαδική αναπαράσταση του αριθμού.
Α. Σωστό
B. Λάθος
2. Χρησιμοποιήστε τους συντελεστές βαρύτητας, για να μετατρέψετε τους ακόλουθους αριθμούς του BCD σε δυαδικούς α) 0101 0011 και β) 0010 0110 1000
Α. 01010011 001001101000
B. 11010100 100001100000
Γ. 110101 100001100
Δ. 101011 001100001
3. Ένας δυαδικός κώδικας που εξελίσσεται έτσι ώστε να αλλάζει μόνο ένα bit μεταξύ δύο διαδοχικών κωδίκων είναι ο κώδικας:
Α. Συμπληρώματος ως προς 9
B. 8421
Γ. Excess-3 (XS3)
Δ. Gray
4. Πόσες εξόδους έχει ένας αποκωδικοποιητής BCD;
Α. 4
B. 16
Γ. 8
Δ. 10

ΑΣΚΗΣΕΙΣ

1. Ο αριθμός $N_2 = 101001100101$ είναι στον κώδικα BCD 2421. Να μετατραπεί στον ισοδύναμο δεκαεξαδικό, δεκαδικό και οκταδικό αριθμό.
2. Να γράψετε έναν BCD κώδικα. Πως είναι δυνατόν να εφοδιασθεί αυτός με την ιδιότητα διαπίστωσης ενός σφάλματος.
3. Καθεμία από τις ακόλουθες δυαδικές παραστάσεις είναι ένα BCD ψηφίο εφοδιασμένο με bit περιττής ισοτιμίας. Σε ποιες παραστάσεις υπάρχει σφάλμα;
α) 00010 β) 00011 γ) 00010 δ) 0111 ε) 00101 στ) 11010
4. Να μετατρέψετε τους παρακάτω δυαδικούς αριθμούς στον κώδικα Gray.
α) 0111010010011101011 β) 100101110101011
γ) 1111100001010 δ) 0010101110001 ε) 1101101101
5. Να μετατρέψετε τους παρακάτω αριθμούς που είναι κωδικοποιημένοι με τον κώδικα Gray, στους αντίστοιχους δυαδικούς.
α) 10011100110101 β) 10101010101010011 γ) 1111111111
δ) 000111000111010011 ε) 10000001000111111
6. Να αναπαρασταθεί ο δεκαδικός αριθμός 468.625 κατά IEEE-754.
7. Να γίνουν οι πράξεις των δυαδικών αριθμών που ακολουθούν κατά IEEE-754: α) $53,97 + 0,15$ β) $7,93 * 2,51$
8. Δίνεται ο δεκαδικός αριθμός 192,21875. Να αναπαρασταθεί κατά IEEE-754.
9. Δίνεται ο κατά IEEE-754 αριθμός 1 01010111 1100.....000. Να αναπαρασταθεί στο δεκαδικό σύστημα.
10. Να εκτελέσετε τον πολλαπλασιασμό των δεκαδικών αριθμών 10 επί -7. Να επιβεβαιώσετε το αποτέλεσμα ως προς την ορθότητά του. Τι παρατηρείτε συγκρίνοντας το αποτέλεσμα που παίρνετε με το αποτέλεσμα της προηγούμενης άσκησης;
11. Να εκτελέσετε στο δυαδικό σύστημα τη διαίρεση των δεκαδικών αριθμών 189 δια 7 με ανάκτηση και χωρίς ανάκτηση. Να σχολιάσετε τα τελικά αποτελέσματα καθώς και τον τρόπο υλοποίησης των διαιρέσεων.
12. Έστω ότι θέλουμε να μεταδώσουμε πληροφορία αποτελούμενη από 4 bit, έστω το δυαδικό αριθμό 1101, χρησιμοποιώντας περιττή ισοτιμία. Να προσδιορίσετε την υπό μετάδοση λέξη χρησιμοποιώντας τον κώδικα Hamming.
13. Έστω ότι θέλουμε να μεταδώσουμε πληροφορία αποτελούμενη από 8 bit, έστω το δυαδικό αριθμό 10110011, χρησιμοποιώντας άρτια ισοτιμία. Να προσδιορίσετε την υπό μετάδοση λέξη, χρησιμοποιώντας τον κώδικα Hamming.

14. Έστω ότι θέλουμε να μεταδώσουμε πληροφορία αποτελούμενη από 8 bit, έστω το δυαδικό αριθμό 10101101. Να προσδιορίσετε την υπό μετάδοση λέξη, χρησιμοποιώντας τον κώδικα Hamming, είτε με άρτια είτε με περιττή ισοτιμία.
15. Να αντιστοιχίσετε τα ψηφία του δεκαδικού συστήματος στο δυεπενταδικό κώδικα και στον κώδικα 2 από 5.
16. Αν η καμινάδα ενός σπιτιού λειτουργεί κανονικά, το σπίτι είναι κρύο και η ενδεικτική λυχνία είναι αναμμένη, να γράψετε τη λογική συνάρτηση που επιτρέπει να ανοίξει η κεντρική βαλβίδα καυσίμου, για να λειτουργήσει η θέρμανση. Δίνονται οι λογικές εκφράσεις b = καμινάδα σε κανονική λειτουργία, c = το σπίτι είναι κρύο, p = η ενδεικτική λυχνία είναι αναμμένη, v = άνοιξε την κεντρική βαλβίδα καυσίμου
17. Να συμπληρώσετε στον παρακάτω πίνακα στα κενά τον αριθμό που αντιστοιχεί στον κάθε κώδικα.

Δεκαδικό	BCD	Κώδικας 84-2-1	Κώδικας 2421	Κώδικας 5421	Κώδικας 5311	Κώδικας XS3	Κώδικας Gray
					1001		
	0111						
			1111				
						1001	
				0010			

Αναφορές-Βιβλιογραφία

- Κοσσίδης Α.Θ. (1996). *Σχεδίαση Ψηφιακών Κυκλωμάτων*, Εκδόσεις Μπένοσ
- Φραγκάκης Γ. (1975). *Λογικά Κυκλώματα*, Αθήνα
- Balabanian N., Carlson B. (2007). *Digital Logic Design Principles*, John Wiley
- Blahut R. E. (2003). *Algebraic Codes for Data Transmission*, Cambridge University Press
- Givone D, (2003). *Digital Principles and Design*,. Mc Graw Hill
- Holdsworth Brian, Woods Clive (2002). *Digital Logic Design*, 4th Edition, Newnes
- IEEE, (2008). *IEEE-754 Standard for Floating Point Arithmetic*
- Klove T., Korzhik V. (1995). *Error detecting Codes*, Kluwer Academic Publishers, Boston
- Lee S., (2006). *Advanced Digital Logic Design*, Thomson-Nelson
- Mano M., Ciletti M., (2014). *Ψηφιακή Σχεδίαση*, 5^η έκδοση, Παπασωτηρίου
- Nashelsky Louis (1994). *Introduction to Digital Technology*, 4th Ed., Prentice Hall
- Nelson V., Nagle H., Carroll B., Irwin J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall
- Predko Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill
- Pritchard N. (2015). *Fundamentals of Digital Electronics*, CreateSpace Independent Publishing Platform
- Rajaraman V., Adabala N. (2015). *Fundamental of Computers*, 6th, PHI Learning
- Shanmugam K. (1979). *Digital and Analog Communication Systems*, John Wiley and Sons
- Tocci R. J., Widmer N. S, and Moss Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley
- Wakerly J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Διαδικτυακοί τόποι

- Απόδοση του κώδικα Hamming, Ανακτήθηκε 10/3/2015 από το <http://cnx.org/contents/uVzho1JX@27/Error-Correcting-Codes-Hamming>
- GAO Report (1992). GAO/IMTEC 92-26 Report <http://www.fas.org/spp/starwars/gao/im92026.htm> Patriot Missile Defense -Software Problem Led to System Failure at Dhahran, Saudi Arabia, USA General Accounting Office, B-247094.
- Kahan W. (2008). <http://http.cs.berkeley.edu/~demmel/cs267/lecture21/lecture21.html>
- Tervo R. (2014). <http://www.ece.unb.ca/tervo/ee4253/hamming.shtml>

Κεφάλαιο 3

Σύνοψη

Αρχικά θα παρουσιαστούν τα αξιώματα και τα θεωρήματα της άλγεβρας Boole. Κατόπιν ακολουθεί η παρουσίαση λογικών κυκλωμάτων με διακόπτες. Βασική είναι η παρουσίαση των ελαχιστοβάθμιων και μεγιστοβάθμιων όρων και η χρήση του. Επίσης περιγράφονται αναλυτικά οι μέθοδοι ελαχιστοποίησης μίας συνάρτησης με τις κλασσικές μεθόδους (διαγράμματα Venn, Χάρτης Karnaugh, μέθοδος Quine Mc Cluskey).

Προαπαιτούμενη γνώση

Βασικές γνώσεις των αριθμητικής, χρήση των διαγραμμάτων Venn. Παράλληλη και σε σειρά χρήση διακοπών (γνώσεις από την θεωρία κυκλωμάτων).

3. Άλγεβρα Boole και ελαχιστοποίηση λογικών συναρτήσεων

3.1. Εισαγωγή

Τα λογικά κυκλώματα, από τα οποία αποτελούνται οι ηλεκτρονικοί υπολογιστές, ξεχωρίζουν από όλα τα άλλα, επειδή οι τάσεις που χρησιμοποιούν έχουν μόνο δύο τιμές. Οι τάσεις αυτές θεωρούνται σαν αριθμητικές μεταβλητές, και ονομάζονται «λογικές» ή «δυναδικές».

Η άλγεβρα που ασχολείται με αυτές τις μεταβλητές καλείται *άλγεβρα λογικής* ή *άλγεβρα Boole*, από το όνομα του Άγγλου μαθηματικού George Boole, που ήταν ο πρώτος που ασχολήθηκε με αυτή.

Η άλγεβρα Boole, όπως θα δούμε παρακάτω, βρίσκει πολλές εφαρμογές στην ηλεκτρονική και ιδιαίτερα στη σχεδίαση λογικών κυκλωμάτων ηλεκτρονικών υπολογιστών, -καθώς και διαφόρων άλλων συστημάτων αυτοματισμού.

Μια μεταβλητή στην άλγεβρα Boole γενικά παριστάνεται με κεφαλαία γράμματα, π.χ. «V», μπορεί μάλιστα, επειδή παίρνει δύο τιμές, να γράφεται ακόμα και σαν V_1 , V_2 , ή σαν V , \bar{V} ή αριθμητικά σαν 1 , 0 .

Καλό είναι να ληφθεί υπόψη ότι τα σύμβολα «0» και «1» χάνουν την αριθμητική τους έννοια που αποδίδουμε σε αυτά και παριστάνουν πλέον τις δύο λογικές καταστάσεις. Επομένως μια μεταβλητή της άλγεβρας Boole είναι μία δίτιμος μεταβλητή που μπορεί να παραστήσει δύο καταστάσεις ανεξάρτητες μεταξύ τους αλλά με δυνατότητα μετάβασης από τη μία κατάσταση στην άλλη.

3.2. Παράσταση των μεταβλητών της Άλγεβρας Boole και πράξεις με αυτές

Οι λογικές μεταβλητές της άλγεβρας Boole, όπως αναφέραμε και προηγουμένως, μπορούν να λάβουν δύο τιμές. Οι τιμές αυτές παριστάνουν δύο συγκεκριμένες καταστάσεις, ανάλογα με το προς επεξεργασία πρόβλημα. Οι λογικές τιμές «1» ή «0» μπορούν να παριστάνουν έναν κλειστό, ή ένα ανοικτό, διακόπτη (Σχήμα 3.1) ή αντίστοιχα το αληθές, ή το εσφαλμένο, μιας απάντησης. Επαναλαμβάνουμε ότι οι μεταβλητές της άλγεβρας Boole γράφονται με κεφαλαία γράμματα του Λατινικού αλφαβήτου π.χ. A, B, C, D, . . .



Σχήμα 3.1. Παράσταση λογικών τιμών στην άλγεβρα Boole με διακόπτες

Εάν έχουμε μια μεταβλητή, π.χ. την A και η αρχική τιμή της είναι ίση με «1», τότε μπορούμε να γράψουμε ότι $A = 1$ και συνεπώς $\bar{A} = 0$. Η παύλα πάνω από το χαρακτήρα, που συμβολίζει τη μεταβλητή, δηλώνει την άρνηση της λογικής μεταβλητής, ή αλλιώς το *συμπλήρωμα* αυτής.

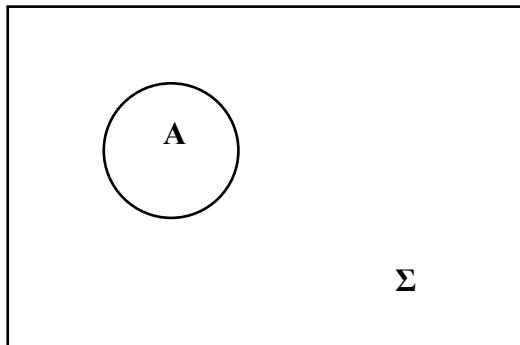
Ένας άλλος τρόπος παράστασης των μεταβλητών του Boole είναι με τη βοήθεια της άλγεβρας των συνόλων και τη χρησιμοποίηση των διαγραμμάτων Venn. Το κυρίως σύνολο Σ παριστάνεται σαν το σύνολο όλων των σημείων στο εσωτερικό ενός ορθογωνίου. Τα υποσύνολα του Σ παριστάνονται σαν σύνολα σημείων του Σ , που βρίσκονται μέσα σε κύκλους.

Στο σχήμα 3.2 με το υποσύνολο A παριστάνεται η λογική μεταβλητή A μέσα στο σύνολο Σ . Οι μεταβλητές του Boole, παριστάνονται, επίσης, και με ευθύγραμμα τμήματα, όπως θα δούμε παρακάτω.

Λογική συνάρτηση είναι μία συνάρτηση που αποτελείται από συνδυασμούς λογικών μεταβλητών, όπως π.χ. $f = A + B$, και θα συμβολίζεται με το γράμμα «T» ή με μικρά λατινικά γράμματα όπως f, g κ.λπ.

Η τελική τιμή μιας λογικής συνάρτησης θα είναι το «0» ή το «1», ανάλογα με τις τιμές των μεταβλητών που συμμετέχουν σ' αυτήν.

Οι πράξεις στην άλγεβρα Boole βασικά είναι δύο και συμβολίζονται με το «+» (λογική πρόσθεση) και με το «·» (λογικό γινόμενο).



Σχήμα 3.2. Παράσταση λογικής μεταβλητής με τη βοήθεια συνόλων

Σε περίπτωση που, για την κατανόηση της άλγεβρας Boole χρησιμοποιούμε σύνολα, οι λογικές πράξεις της πρόσθεσης και του γινομένου εκφράζονται με τα σύμβολα της ένωσης (« \cup ») ή της τομής (« \cap ») δύο συνόλων αντίστοιχα.

Τέλος, σε περίπτωση παράστασης των μεταβλητών του **Boole** με διακόπτες, οι λογικές πράξεις της πρόσθεσης και του γινομένου εκφράζονται με τη λογική πράξη «OR» και τη λογική πράξη «AND». Το λογικό άθροισμα παριστάνεται με διακόπτες συνδεσμολογημένους παράλληλα, ενώ το λογικό γινόμενο με διακόπτες συνδεσμολογημένους σε σειρά. (Nashelsky, 1994; Κοσσιδάς, 1996; Givone, 2002; Katz, 2005; Wakerly, 2006; Balabanian, & Carlson, 2007; Maxfield, 2009; Tocci, et al., 2010; Mano, & Ciletti, 2014)

3.3. Αξιώματα της Άλγεβρας Boole

Η *άλγεβρα Boole* στηρίζεται σε μια σειρά αξιωμάτων, τα οποία παραθέτουμε παρακάτω. Τα αξιώματα αυτά δεν αποτελούν, βεβαίως, το μοναδικό τρόπο ορισμού της. Μας επιτρέπουν όμως να την κατανοήσουμε μ' έναν απλό και εύκολο τρόπο. Για την ανάπτυξη της άλγεβρας Boole θεωρούμε μία κλάση Σ , που περιλαμβάνει τα στοιχεία (λογικές μεταβλητές) A, B, \dots, X κ.λπ., όπως, επίσης, και τα στοιχεία που προκύπτουν από τις πράξεις μεταξύ των στοιχείων που συμπεριλαμβάνονται στην κλάση K . Θεωρούμε ακόμα ότι η κλάση K περιέχει πάντοτε περισσότερα του ενός στοιχεία. Τόσο τα αξιώματα της άλγεβρας Boole, όσο και τα θεωρήματα που θα δούμε στη συνέχεια χαρακτηρίζονται από μία ιδιότητα που ονομάζεται *δυσκότητα*. Εξακολουθούν, δηλαδή, να ισχύουν και στην περίπτωση που τα σύμβολα «+» και «·», όπως, επίσης και οι ποσότητες «1» και «0», αντιμετωπίζονται.

1^ο Αξίωμα: Εάν $X \neq 1$ θα είναι $X=0$ ή το δυικό εάν $X \neq 0$ θα είναι $X=1$

2^ο Αξίωμα: Οι πράξεις λογικής πρόσθεσης και γινομένου είναι αντιμεταθετικές: $X+Y=Y+X$ ή η δυική της $X \cdot Y=Y \cdot X$

3^ο Αξίωμα: Για κάθε διμελή πράξη «+» ή «·» υπάρχει ένα στοιχείο «0» ή «1» τέτοιο ώστε: $X+0=X$ και $X \cdot 1=X$

4^ο Αξίωμα: Για κάθε διμελή πράξη «+» ή «·» υπάρχει ένα στοιχείο «1» ή «0» τέτοιο ώστε: $X+1=1$ ή η δυική της $X \cdot 0=0$

5^ο Αξίωμα: Το λογικό άθροισμα (γινόμενο) μιας μεταβλητής με τη συμπληρωματική της ισούται με 1 (0). $X+\bar{X}=1$ και $X \cdot \bar{X}=0$

6^ο Αξίωμα: Κάθε πράξη μεταξύ λογικών μεταβλητών είναι επιμεριστική.

$X \cdot (Y+Z) = X \cdot Y + X \cdot Z$ ή η δυική της $X+(Y \cdot Z) = (X+Y) \cdot (X+Z)$

3.4. Θεωρήματα της Άλγεβρας Boole

Για την απόδειξη των θεωρημάτων της άλγεβρας Boole μπορούμε να χρησιμοποιήσουμε διαφορετικούς τρόπους. Ο κλασικός τρόπος απόδειξης στηρίζεται στα αξιώματα της άλγεβρας Boole, που αναφέραμε και στην εξίσωση των δύο μερών του θεωρήματος, με κατάλληλους μετασχηματισμούς.

Ένας άλλος τρόπος απόδειξης είναι να θέσουμε σε κάθε μεταβλητή όλες τις πιθανές τιμές που μπορεί να λάβει και να εξακριβώσουμε ότι το θεώρημα ισχύει για όλες αυτές (επαγωγικός τρόπος). Κατά τον επαγωγικό τρόπο απόδειξης χρησιμοποιούνται πολλές φορές κατάλληλοι συνδυαστικοί πίνακες, οι οποίοι είναι γνωστοί με το όνομα «πίνακες αλήθειας».

Ένας τελευταίος τρόπος απόδειξης χρησιμοποιεί τα διαγράμματα Venn, ή ακόμα απλά ηλεκτρικά κυκλώματα με διακόπτες. (Nelson, et al., 1995; Holdsworth, & Woods, 2002; Predko, 2007; Godse, & Godse, 2011; Mano, & Ciletti, 2014; Pritchard, 2015)

3.4.1. Θεώρημα 1ο

$$\mathbf{X + X = X} \quad (3-1) \quad \text{ή το δυικό του} \quad \mathbf{X \cdot X = X} \quad (3-2)$$

Απόδειξη

$$\begin{aligned} \mathbf{X + X} &= (\mathbf{X + X}) \cdot \mathbf{1} = && \text{αξίωμα } 3^{\circ} \\ &= (\mathbf{X + X}) \cdot (\mathbf{X + \bar{X}}) = && \text{αξίωμα } 5^{\circ} \\ &= \mathbf{X \cdot X + X \cdot \bar{X} + X \cdot X + X \cdot \bar{X}} = && \text{αξίωμα } 3^{\circ} \\ &= \mathbf{X \cdot X + X \cdot \bar{X} + X \cdot X + 0} = \\ &= \mathbf{X \cdot (X + \bar{X}) + X \cdot X} = && \text{αξίωμα } 5^{\circ} \\ &= \mathbf{X + X \cdot X} = \\ &= \mathbf{X \cdot (1 + X)} = && \text{αξίωμα } 4^{\circ} \\ &= \mathbf{X \cdot 1 = X} && \text{αξίωμα } 3^{\circ} \end{aligned}$$

Το παραπάνω θεώρημα αποδεικνύεται ευκολότερα με επαγωγικό τρόπο, δίνοντας, δηλαδή, όλες τις δυνατές τιμές στις μεταβλητές που συσχετίζει.

3.4.2. Θεώρημα 2ο

$$\mathbf{\bar{\bar{X}} = X} \quad (3-3)$$

Απόδειξη

$$\mathbf{X + \bar{X} = 1} \quad \text{και} \quad \mathbf{X \cdot \bar{X} = 0} \quad \text{αξίωμα } 5^{\circ}$$

Με βάση το 5^ο αξίωμα είναι προφανές ότι το \mathbf{X} είναι συμπληρωματικό του $\bar{\mathbf{X}}$. Το θεώρημα επίσης, αποδεικνύεται επαγωγικά συμπληρώνοντας ένα πίνακα αλήθειας. (Βλέπε πίνακα 3.1).

\mathbf{X}	$\bar{\mathbf{X}}$	$\bar{\bar{\mathbf{X}}}$
0	1	0
1	0	1

Πίνακας 3.1. Απόδειξη του θεωρήματος $\bar{\bar{X}} = X$

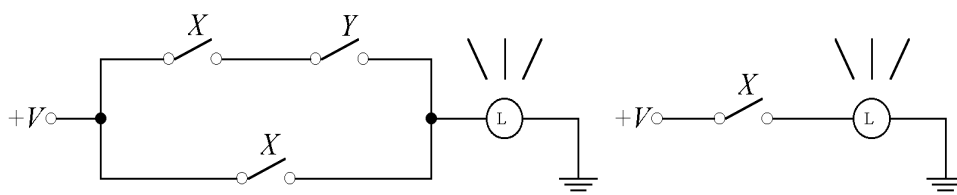
3.4.3. Θεώρημα 3ο

$$\mathbf{X + X \cdot Y = X} \quad (3-4) \quad \text{ή το δυικό του} \quad \mathbf{X \cdot (X + Y) = X} \quad (3-5)$$

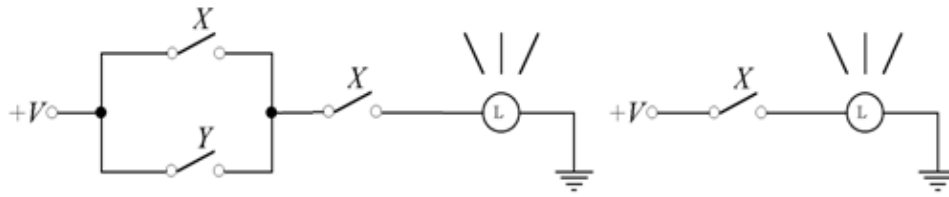
Απόδειξη

$$\begin{aligned} \mathbf{X + X \cdot Y} &= \mathbf{X \cdot (1 + Y)} = \mathbf{X} && \text{και το δυικό του} \\ \mathbf{X \cdot (X + Y)} &= \mathbf{X \cdot X + X \cdot Y} = \mathbf{X + X \cdot Y} = \mathbf{X \cdot (1 + Y)} = \mathbf{X} \end{aligned}$$

Η απόδειξη των ανωτέρω μπορεί να γίνει και με τους διακόπτες που απεικονίζονται στα σχήματα 3.3 και 3.4.



Σχήμα 3.3. Απόδειξη του 3^{ου} θεωρήματος με διακόπτες



Σχήμα 3.4 Απόδειξη της διτικής μορφής 3^{ου} θεωρήματος με διακόπτες

Πράγματι τα αντίστοιχα ζεύγη κυκλωμάτων είναι ισοδύναμα.

3.4.4. Θεώρημα 4ο

$$Y \cdot (X + \bar{Y}) = X \cdot Y \quad (3-6) \text{ ή το διτικό του } Y + (X \cdot \bar{Y}) = X + Y \quad (3-7)$$

Απόδειξη

$$Y \cdot (X + \bar{Y}) = X \cdot Y + \bar{Y} \cdot Y = X \cdot Y + 0 = X \cdot Y$$

Η απόδειξη της διτικής μορφής του θεωρήματος είναι λίγο περίπλοκη. Το πρώτο μέλος του θα ισούται με:

$$Y + X \cdot \bar{Y} = Y \cdot (X + \bar{X}) + X \cdot \bar{Y} = X \cdot Y + \bar{X} \cdot Y + X \cdot \bar{Y} = X \cdot (Y + \bar{Y}) + \bar{X} \cdot Y = X + \bar{X} \cdot Y$$

Το δεύτερο μέλος του θα ισούται με:

$$X + Y = X + Y \cdot (X + \bar{X}) = X + X \cdot Y + \bar{X} \cdot Y = X \cdot (1 + Y) + \bar{X} \cdot Y = X + \bar{X} \cdot Y$$

Επειδή, όπως προκύπτει το πρώτο μέλος ισούται με το δεύτερο, το θεώρημα ισχύει.

3.4.5. Θεώρημα 5ο- (Θεωρήματα De Morgan)

$$\overline{X + Y} = \bar{X} \cdot \bar{Y} \quad (3-8) \text{ ή το διτικό του } \overline{X \cdot Y} = \bar{X} + \bar{Y} \quad (3-9)$$

Απόδειξη

Η πρώτη ισότητα λέει ότι το $X + Y$ είναι το συμπλήρωμα του $\bar{X} \cdot \bar{Y}$. Επομένως βάσει του 5^{ου} αξιώματος πρέπει να ισχύει:

$$(X + Y) \cdot (\bar{X} \cdot \bar{Y}) = 0$$

$$(X + Y) + (\bar{X} \cdot \bar{Y}) = 1$$

Η πρώτη ισότητα αποδεικνύεται εύκολα. Πράγματι:

$$(X + Y) \cdot (\bar{X} \cdot \bar{Y}) = X \cdot \bar{X} \cdot \bar{Y} + Y \cdot \bar{X} \cdot \bar{Y} = 0 + 0 = 0$$

Για τη δεύτερη ισότητα έχουμε:

$$(X + Y) + (\bar{X} \cdot \bar{Y}) = (X \cdot \bar{Y} + Y) + (\bar{X} \cdot \bar{Y}) = X \cdot \bar{Y} + Y + \bar{X} \cdot \bar{Y} = \bar{Y} \cdot (X + \bar{X}) + Y$$

$$= \bar{Y} \cdot 1 + Y = \bar{Y} + Y = 1$$

Το παραπάνω θεώρημα ονομάζεται θεώρημα του De Morgan. Η απόδειξή του είναι ευκολότερη με τα διαγράμματα του Venn. Τα σχήματα 3.5γ, 3.5δ που ακολουθούν προκύπτουν ισοδύναμα. Όπως θα δούμε στη συνέχεια, με το θεώρημα του De Morgan είναι πάντοτε δυνατόν, όταν δοθεί ένα κύκλωμα, να βρεθεί ένα άλλο, το οποίο να έχει συμπληρωματικά χαρακτηριστικά. Το γεγονός αυτό είναι πολύ ενδιαφέρον, γιατί, σε πολλές περιπτώσεις το συμπληρωματικό κύκλωμα είναι κατά πολύ απλούστερο, με συνέπεια την ευκολότερη υλοποίησή του.

3.4.6. Θεώρημα 6ο

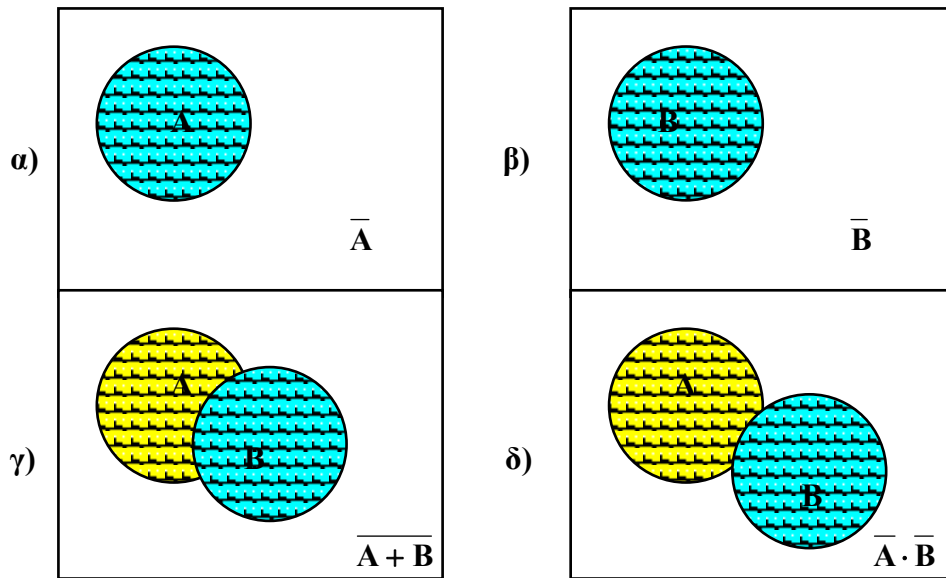
$$(X + Y) \cdot (\bar{X} + Z) = X \cdot Z + \bar{X} \cdot Y \quad (3-10)$$

Απόδειξη

$$(X + Y) \cdot (\bar{X} + Z) = X \cdot \bar{X} + X \cdot Z + \bar{X} \cdot Y + Y \cdot Z =$$

$$= 0 + X \cdot Z + \bar{X} \cdot Y + Y \cdot Z \cdot (X + \bar{X}) = X \cdot Z \cdot (1 + Y) + \bar{X} \cdot Y \cdot (1 + Z) = X \cdot Z + \bar{X} \cdot Y$$

Άρα αρχικά κανονικοποιώ τον τρίτο όρο και χωρίζω σε νέα ζεύγη για την τελική απλοποίηση.



Σχήμα 3.5. Απόδειξη του θεωρήματος De Morgan

3.4.7. Θεώρημα 7ο– Γενίκευση του θεωρήματος De Morgan

Γενικεύοντας το θεώρημα του De Morgan σε n μεταβλητές, έχουμε:

$$\overline{(X + Y + Z + \dots)} = \bar{X} \cdot \bar{Y} \cdot \bar{Z} \cdot \dots \quad (3-11)$$

$$\overline{(X \cdot Y \cdot Z \cdot \dots)} = \bar{X} + \bar{Y} + \bar{Z} + \dots \quad (3-12)$$

Γενικεύοντας, φθάνουμε στο θεώρημα του Shannon, το οποίο δίνεται από την παρακάτω παράσταση:

$$f(\overline{X_1, X_2, \dots, X_n, +, \dots}) = f(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n, \dots, +)$$

όπου με $f()$ παρίσταται μια λογική συνάρτηση, δηλαδή ένας συνδυασμός μεταβλητών X_1, X_2, \dots και σημείων « \cdot » « $+$ ».

3.4.8. Θεώρημα 8ο

Μια συνάρτηση n λογικών μεταβλητών μπορεί να εκφραστεί με διάφορους τρόπους όπως φαίνεται παρακάτω:

$$\begin{aligned} f(X_n) &= f(X_1, X_2, \dots, X_n) = \\ &= X_1 \cdot f(1, X_2, \dots, X_n) + \bar{X}_1 \cdot f(0, X_2, \dots, X_n) = \\ &= [X_1 + f(0, X_2, \dots, X_n)] \cdot [\bar{X}_1 + f(1, X_2, \dots, X_n)] \end{aligned} \quad (3-13)$$

Οι δύο τελευταίες ισότητες αποδεικνύονται ισοδύναμες με την επαγωγική μέθοδο. Αντικαθιστούμε στο X τιμές 0 και 1. Αυτό ονομάζεται *θεώρημα της επέκτασης* (expansion theorem).

Στηριζόμενοι σε αυτό το θεώρημα, μπορούμε να συνάγουμε εκφράσεις χρήσιμες για την άλγεβρα Boole.

Επαγωγικά αποδεικνύεται ότι ισχύει η παρακάτω έκφραση:

$$X_1 + f(X_1, X_n) = X_1 + f(0, X_n) \quad (3-14)$$

Εφαρμόζοντας τώρα την *ισότητα 3-14*, αποδεικνύουμε ότι ισχύει η παρακάτω σχέση:

$X + Y + (X + \bar{Y}) \cdot (\bar{X} + Y) = 1$, πράγματι κάνοντας πράξεις έχουμε:

$$X + Y + (X + \bar{Y}) \cdot (\bar{X} + Y) = X + Y + (0 + \bar{Y}) \cdot (\bar{X} + 0) = X + Y + \bar{X} \cdot \bar{Y} = (X + Y) + \overline{X \cdot Y} = 1$$

3.5. Μελέτη της Άλγεβρας με λογικά κυκλώματα

Ονομάζουμε *λογικά κυκλώματα* εκείνα τα κυκλώματα που αποτελούνται από έναν ή περισσότερους ακροδέκτες εισόδου και έναν ή περισσότερους ακροδέκτες εξόδου, και χαρακτηρίζονται από το γεγονός ότι σε κάθε ακροδέκτη μπορούν να εφαρμοσθούν δύο προκαθορισμένες τιμές τάσεων π.χ. «0» ή «V». Ο αριθμός των ακροδεκτών εισόδου και εξόδου μπορεί να είναι διαφορετικός.

Τυχόν τάσεις διαφορετικής τιμής στο κύκλωμα δηλώνουν ότι δεν εργάζεται σωστά, παρουσιάζει δηλαδή κατασκευαστικές ανωμαλίες.

Η κατασκευή των λογικών κυκλωμάτων μπορεί να γίνει τόσο με διακόπτες όσο και με διόδους, τρανζίστορ, ολοκληρωμένα κυκλώματα, PLA κ.λπ.

Στο κεφάλαιο αυτό θα αναφερθούμε περισσότερο στα κυκλώματα διακοπών. Με βάση αυτά θα μελετήσουμε διάφορες λογικές συναρτήσεις. Θα αφήσουμε για το επόμενο κεφάλαιο την ανάλυση των πρακτικών κυκλωμάτων των υπολογιστών, που κατασκευάζονται με ηλεκτρονικά εξαρτήματα.

Τα λογικά κυκλώματα, των οποίων οι τάσεις των εξόδων τους είναι συναρτήσεις τόσο της κατασκευαστικής δομής του κυκλώματος τους όσο και των τιμών εισόδου που επιβάλλονται την ίδια χρονική στιγμή, ονομάζονται «**συνδυαστικά κυκλώματα**» (combinational circuits). Εάν όμως οι τιμές των τάσεων εξόδου είναι συναρτήσεις και των προηγούμενων τιμών των τάσεων εισόδου ή εξόδου, τότε λέμε ότι είναι λογικά «**ακολουθιακά κυκλώματα**» (sequential circuits). Τα ακολουθιακά κυκλώματα θα αναφερθούν σε επόμενο κεφάλαιο.

Για τη μελέτη ενός κυκλώματος, πρέπει να ληφθούν υπόψη ορισμένες προϋποθέσεις έτσι, ώστε να ακολουθηθεί η κατάλληλη διαδικασία και να επιτευχθεί το βέλτιστο κύκλωμα. Η διαδικασία μελέτης αναφέρεται παρακάτω, χωρίς όμως αυτό να προδικάζει την πιστή της εφαρμογή. Έχει αποδειχθεί ότι η πείρα του μελετητή μπορεί να παρακάμψει ορισμένες λεπτομέρειες και να προσθέσει ορισμένα απαραίτητα στοιχεία στο όλο κύκλωμα κατά τη διάρκεια της μελέτης του. Οι ενέργειες που πρέπει να γίνουν συνοψίζονται παρακάτω:

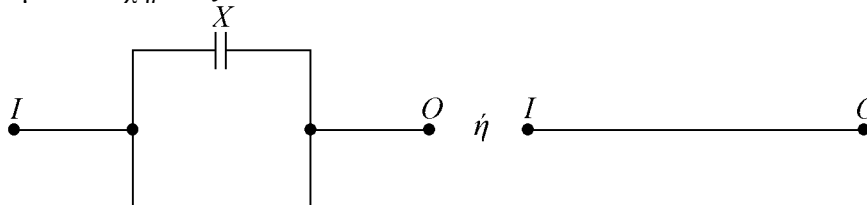
- Καθορισμός της συνάρτησης μεταφοράς (συνάρτηση που συνδέει την είσοδο με την έξοδο) ή λογικής συνάρτησης του υπό μελέτη κυκλώματος.
- Ελαχιστοποίηση και απλοποίηση της συνάρτησης μεταφοράς.
- Σύνθεση του λογικού κυκλώματος με βάση τη συνάρτηση μεταφοράς.
- Μελέτη και κατασκευή του πραγματικού ηλεκτρονικού κυκλώματος από το λογικό κύκλωμα που φτιάξαμε.
- Προσθήκες - συμπληρώσεις στο ηλεκτρικό κύκλωμα έτσι, ώστε να αντισταθμιστούν απώλειες, παραμορφώσεις και λοιπά προβλήματα των ηλεκτρονικών εξαρτημάτων.

3.6. Λογικά κυκλώματα διακοπών

Παραθέτουμε ορισμένα κυκλώματα διακοπών μαζί με τις αντίστοιχες συναρτήσεις που ικανοποιούν. Σε κάθε επαφή διακόπτη αντιστοιχούμε και μια λογική μεταβλητή. Ορίζουμε π.χ. με X μια επαφή κλειστή και με \bar{X} μια ανοικτή. (Φραγκάκης, 1975; Κοσσίδα, 1996; Floyd, 2013; Mano, & Ciletti, 2014)

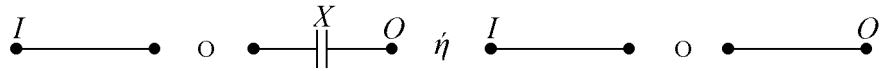
Θα δώσουμε ορισμένα παραδείγματα με διακόπτες από τα ήδη αναφερθέντα αξιώματα και θεωρήματα της άλγεβρας Boole. Καλό είναι να υπενθυμίσουμε ότι το λογικό άθροισμα δύο μεταβλητών αντιστοιχεί στον παράλληλο συνδυασμό δύο διακοπών ενώ το γινόμενο στην τοποθέτηση δύο διακοπών σε σειρά.

Η παράσταση των διακοπών γίνεται με το σύμβολο « $-|$ » χάριν απλότητας. Η έκφραση $1+X=1$ αντιστοιχεί το κύκλωμα του Σχήματος 3.6.



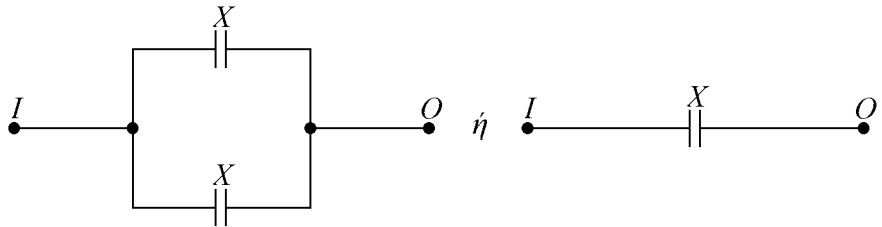
Σχήμα 3.6. Απεικόνιση της παράστασης $1+X$ με διακόπτες

Η έκφραση $0 \cdot X = 0$ αντιστοιχεί το κύκλωμα του Σχήματος 3.7.



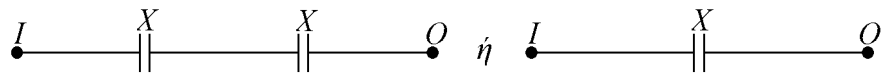
Σχήμα 3.7. Απεικόνιση της παράστασης $0 \cdot X$ με διακόπτες

Η έκφραση $X+X=X$ αντιστοιχεί το κύκλωμα του Σχήματος 3.8.



Σχήμα 3.8. Απεικόνιση της παράστασης $X+X$ με διακόπτες

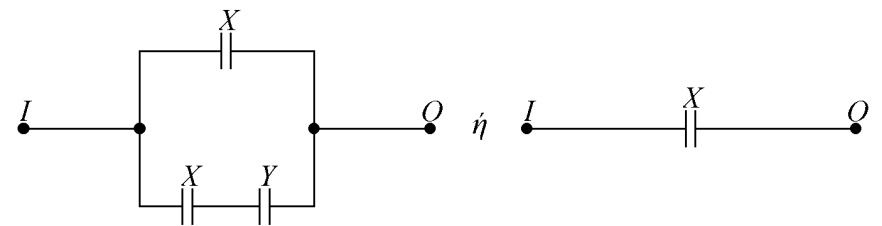
Η έκφραση $X \cdot X = X$ αντιστοιχεί το κύκλωμα του Σχήματος 3.9.



Σχήμα 3.9. Απεικόνιση της παράστασης $X \cdot X$ με διακόπτες

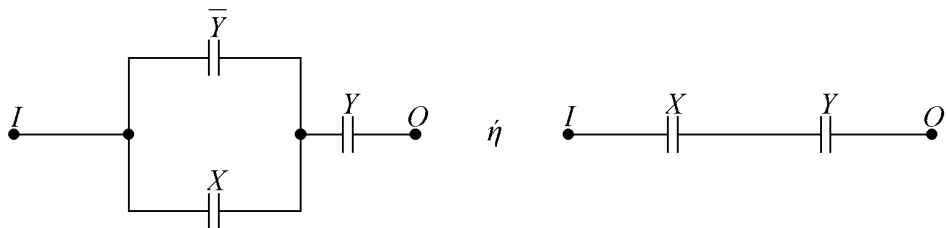
Κατά συνέπεια για τις παρακάτω λογικές εκφράσεις έχουμε τα αντίστοιχα κυκλώματα μαζί με τα ισοδύναμά τους.

α) $X + X \cdot Y = X$



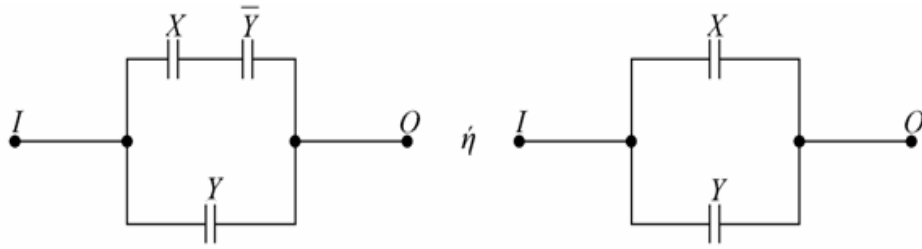
Σχήμα 3.10. Απεικόνιση της παράστασης $X + X \cdot Y$ με διακόπτες

β) $(X + \bar{Y}) \cdot Y = X \cdot Y$



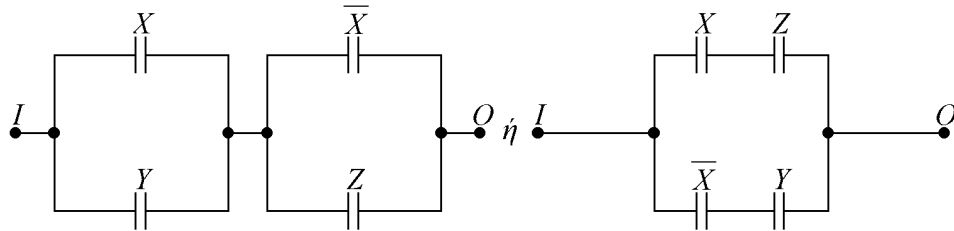
Σχήμα 3.11. Απεικόνιση της παράστασης $(X + \bar{Y}) \cdot Y$ με διακόπτες

γ) $X \cdot \bar{Y} + Y = X + Y$



Σχήμα 3.12. Απεικόνιση της παράστασης $X \cdot \bar{Y} + Y$ με διακόπτες

δ) $(X + Y) \cdot (\bar{X} + Z) = X \cdot Z + \bar{X} \cdot Y$



Σχήμα 3.13. Απεικόνιση της παράστασης $(X + Y) \cdot (\bar{X} + Z)$ με διακόπτες

3.7. Ανάλυση ενός λογικού κυκλώματος με διακόπτες

Ανάλυση ενός κυκλώματος ονομάζουμε την εργασία, όπου έχοντας δοσμένη τη γραφική απεικόνιση ενός κυκλώματος, υπολογίζουμε τη λογική συνάρτηση μεταφοράς που το περιγράφει.

Η ανάλυση του κυκλώματος στηρίζεται στους παρακάτω κανόνες:

- Η ανάλυση του κυκλώματος γίνεται τεμαχίζοντας το κύκλωμα σε επί μέρους στοιχειώδη κυκλώματα σειράς από τα οποία αποτελείται ή εν παραλλήλω.
- Σε κάθε επαφή αντιστοιχεί μία μόνο λογική μεταβλητή.
- Η λογική μεταβλητή X δεν είναι υπεργραμμισμένη (δεν έχει παύλα επάνω της), εάν η επαφή είναι κλειστή, ενώ είναι υπεργραμμισμένη \bar{X} (έχει παύλα), εάν αυτή είναι ανοικτή.
- Σε κάθε κλειστή επαφή η συνάρτηση μεταφοράς έχει τιμή «1», ενώ σε κάθε ανοικτή έχει τιμή «0».
- Σε κάθε παράλληλη διάταξη δύο επαφών αντιστοιχεί μια συνάρτηση αθροίσματος των λογικών μεταβλητών. Σε κάθε διάταξη σειράς επαφών αντιστοιχεί μία συνάρτηση γινόμενου των δύο λογικών μεταβλητών.

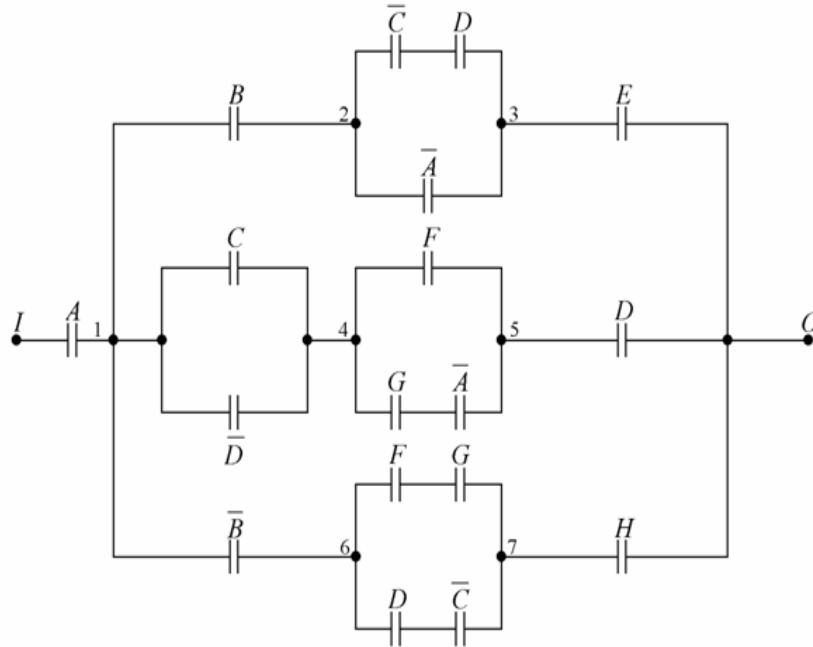
Παράδειγμα 1ο:

Έστω ότι δίνεται το κύκλωμα του σχήματος 3.14. Αυτό χωρίζεται στα εξής τμήματα με τις αντίστοιχες συναρτήσεις μεταφοράς.

I	-	1	A
1	-	2	B
2	-	3	$\bar{C} \cdot D + \bar{A}$
3	-	O	E
1	-	4	$C + \bar{D}$
4	-	5	$F + G \cdot \bar{A}$
5	-	O	D
1	-	6	\bar{B}
6	-	7	$F \cdot G + D \cdot \bar{C}$
7	-	O	H

Συνενώνοντας τα παραπάνω τμήματα κατά κλάδους, έχουμε τις συναρτήσεις:

1 - 2 - 3 - O	$B \cdot (\bar{C} \cdot D + \bar{A}) \cdot E$
1 - 4 - 5 - O	$(C + \bar{D}) \cdot (F + G \cdot \bar{A}) \cdot D$
1 - 6 - 7 - O	$\bar{B} \cdot (F \cdot G + D \cdot \bar{C}) \cdot H$



Σχήμα 3.14. Κύκλωμα σειράς – παραλλήλου προς ανάλυση

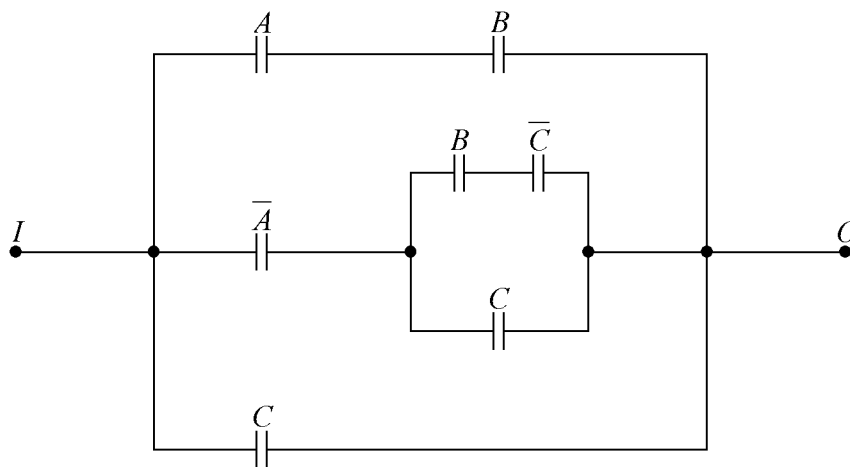
Ο παράλληλος συνδυασμός των τριών κλάδων 1 - O θα είναι:

$$B \cdot (\bar{C} \cdot D + \bar{A}) \cdot E + (C + \bar{D}) \cdot (F + G \cdot \bar{A}) \cdot D + \bar{B} \cdot (F \cdot G + D \cdot \bar{C}) \cdot H$$

Και η ολική συνάρτηση μεταφοράς θα έχει την παρακάτω μορφή:

$$T = I - O = A \cdot [B \cdot (\bar{C} \cdot D + \bar{A}) \cdot E + (C + \bar{D}) \cdot (F + G \cdot \bar{A}) \cdot D + \bar{B} \cdot (F \cdot G + D \cdot \bar{C}) \cdot H] \quad (3-15)$$

Για τον υπολογισμό της τιμής της συνάρτησης μεταφοράς **T**, ανάλογα με την εκάστοτε τιμή των λογικών μεταβλητών, συνηθίζεται η κατασκευή ενός πίνακα αλήθειας ο οποίος αποτελείται από τόσες στήλες όσες είναι οι υπάρχουσες μεταβλητές στη συνάρτηση συν μια επιπλέον για την τιμή της υπό μελέτη συνάρτησης **T**. Ο πίνακας αποτελείται από τόσες γραμμές, όσοι είναι οι δυνατοί συνδυασμοί των μεταβλητών λαμβανομένων όλων μαζί.



Σχήμα 3.15. Κύκλωμα για τον υπολογισμό της συνάρτησης T

Θα υπολογίσουμε τώρα την τιμή της συνάρτησης μεταφοράς για το κύκλωμα του σχήματος 3.15. Θα εργαστούμε ανάλογα με το προηγούμενο παράδειγμα.

Εφαρμόζοντας τους κανόνες που αναφέραμε παραπάνω, προκύπτει ότι:

$$T = A \cdot B + \overline{A} \cdot (B \cdot \overline{C} + C) + C \quad (3-16)$$

Ο πίνακας αλήθειας της συνάρτησης **T** (πίνακας 3.2) βρίσκεται, υπολογίζοντας, για όλες τις δυνατές τιμές των **A, B, C**, την τιμή της **T**. π.χ. για **A=B=C=0**, το **T=0** κ.ο.κ.

A	B	C	T
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Πίνακας 3.2. Πίνακας αλήθειας της λογικής συνάρτησης της σχέσης 3-16

3.8. Σύνθεση ενός λογικού κυκλώματος με διακόπτες

Σύνθεση ενός κυκλώματος είναι η εργασία κατά την οποία μελετάται και κατασκευάζεται ένα κύκλωμα βάσει συγκεκριμένων απαιτήσεων και προδιαγραφών έτσι, ώστε η έξοδος του να δίνει τις επιθυμητές τιμές.

Γενικά οι προδιαγραφές του κυκλώματος δίνονται υπό μορφή πίνακα αλήθειας.

Έστω, π.χ., ότι θέλουμε να κατασκευάσουμε ένα κύκλωμα που θα αποτελείται από δύο επαφές (μεταβλητές) (**X, Y**) τέτοιο, ώστε η συνάρτηση **T** να έχει τιμή «1», μόνο όταν και οι δύο επαφές έχουν την ίδια κατάσταση. Ο πίνακας αλήθειας 3.3, που ακολουθεί, περιγράφει το κύκλωμα.

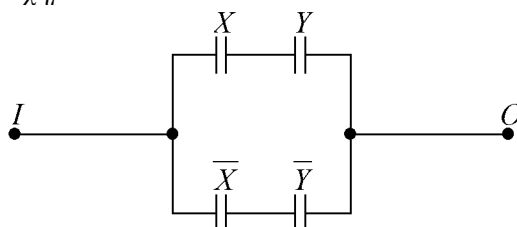
X	Y	T
0	0	1
0	1	0
1	0	0
1	1	1

Πίνακας 3.3. Πίνακας αλήθειας της συνάρτησης **T**

Το ζητούμενο κύκλωμα μπορεί να γίνει με δύο τρόπους: είτε εκφράζοντας τις περιπτώσεις, όπου η **T=1**, είτε εκφράζοντας τις περιπτώσεις, όπου η **T=0**. (Συνήθως θα χρησιμοποιούμε τον πρώτο τρόπο). Στο παράδειγμά μας θέλουμε **T=1**, αν το **X** και το **Y** είναι 0 ή εάν το **X** και το **Y** είναι 1. Αυτό συνεπάγεται μια παράλληλη σύνδεση μεταξύ του $\overline{X} \cdot \overline{Y}$ και του **X Y**. Η συνάρτηση **T** θα δίνεται από τη σχέση 3-17:

$$T = \overline{X} \cdot \overline{Y} + X \cdot Y \quad (3-17)$$

το δε αντίστοιχο κύκλωμα από το σχήμα 3.16.



Σχήμα 3.16. Παράσταση του πίνακα αλήθειας 3.3, με κύκλωμα διακοπών

Κατ' αναλογία, λαμβάνοντας υπόψη τις περιπτώσεις όπου **T=0**, έχουμε τη σχέση 3-18:

$$T = (X + \overline{Y}) \cdot (\overline{X} + Y) \quad (3-18)$$

Βλέπουμε ότι αυτή η έκφραση της **T** είναι ισοδύναμη της προηγούμενης. Πράγματι, εκτελώντας τις πράξεις, επανερχόμαστε στην προηγούμενη έκφραση. Η επιλογή των μονάδων ή των μηδενικών από τον πίνακα αλήθειας για την έκφραση της **T** εξαρτάται από τον αριθμό τους έτσι, ώστε να οδηγούμαστε στην απλούστερη δυνατή συνάρτηση. Θα πρέπει να σημειώσουμε ότι πολλές φορές οι συναρτήσεις που παράγονται δεν παριστάνουν πάντοτε την απλούστερη μορφή του κυκλώματος. Σε πολλές περιπτώσεις μπορούν, με κατάλληλες μεθόδους, να απλοποιηθούν ακόμα περισσότερο.

3.9. Μέθοδοι ελαχιστοποίησης λογικών συναρτήσεων

Οι λογικές συναρτήσεις, οι οποίες δίνονται, όπως και αυτές που προκύπτουν από πίνακες αλήθειας, πολλές φορές ελαχιστοποιούνται, με σκοπό το προς μελέτη και κατασκευή κύκλωμα να γίνει μικρότερο και απλούστερο. Το γεγονός αυτό έχει σαν αποτέλεσμα την καλύτερη και απλούστερη λειτουργία του.

Οι μέθοδοι που ακολουθούνται για την ελαχιστοποίηση των ψηφιακών κυκλωμάτων είναι πολλές. Θα αναφέρουμε μόνο αυτές που χρησιμοποιούνται περισσότερο. (Givone, 2002; Maxfield, 2009; Godse, 2011; Mano, & Ciletti, 2014; Pritchard, 2015)

- α. Αλγεβρική μέθοδος,
- β. Γραφική μέθοδος,
- γ. Διαγράμματα του Venn,
- δ. Χάρτης Karnaugh,
- ε. Μέθοδος Quine -Mc Cluskey

Πριν προχωρήσουμε στην ανάλυση των μεθόδων ελαχιστοποίησης, θα μελετήσουμε τους ελαχιστοβάθμιους και μεγιστοβάθμιους όρους μίας συνάρτησης καθώς και την κανονικοποίηση.

3.9.1. Ελαχιστοβάθμιοι και μεγιστοβάθμιοι όροι

3.9.1.1 Ελαχιστοβάθμιος όρος (Minterm) - ΕΛΒΟ

Ένας ελαχιστοβάθμιος όρος (ΕΛΒΟ) είναι ένα γινόμενο παραγόντων, στο οποίο κάθε μεταβλητή εισόδου εμφανίζεται μία μόνο φορά. Στις συναρτήσεις που ακολουθούν, για λόγους λειτουργικότητας, η παρουσίαση των ΕΛΒΟ γίνεται με τη χρήση του λογικού γινομένου (\cdot) μεταξύ των λογικών μεταβλητών, ή, όπου δε χρησιμοποιείται, θεωρείται αυτονόητη η ύπαρξή του.

Μία συνάρτηση με n μεταβλητές έχει 2^n ελαχιστοβάθμιους όρους (καθώς κάθε μεταβλητή μπορεί να εμφανιστεί με την κανονική ή τη συμπληρωματική της μορφή).

Έτσι μία συνάρτηση με τρεις μεταβλητές, όπως π.χ. η $f(x,y,z)$, έχει $2^3 = 8$ ελαχιστοβάθμιους όρους:

1 ^{ος}	$\overline{x} \overline{y} \overline{z}$	2 ^{ος}	$\overline{x} y \overline{z}$	3 ^{ος}	$\overline{x} y z$	4 ^{ος}	$x \overline{y} \overline{z}$
5 ^{ος}	$\overline{x} y z$	6 ^{ος}	$x \overline{y} \overline{z}$	7 ^{ος}	$x y \overline{z}$	8 ^{ος}	$x y z$

Κάθε ελαχιστοβάθμιος όρος είναι αληθής για ακριβώς ένα συνδυασμό εισόδων:

ΕΛΒΟ (Μεταβλητές x,y,z)	Είναι αληθής όταν..	Συντομογραφία
$\overline{x} \overline{y} \overline{z}$	$x=0, y=0, z=0$	m_0
$\overline{x} y \overline{z}$	$x=0, y=0, z=1$	m_1
$\overline{x} y z$	$x=0, y=1, z=0$	m_2
$\overline{x} y z$	$x=0, y=1, z=1$	m_3
$x \overline{y} \overline{z}$	$x=1, y=0, z=0$	m_4
$x \overline{y} z$	$x=1, y=0, z=1$	m_5
$x y \overline{z}$	$x=1, y=1, z=0$	m_6
$x y z$	$x=1, y=1, z=1$	m_7

Κάθε συνάρτηση μπορεί να γραφτεί σαν ένα άθροισμα ελαχιστοβάθμιων όρων (ΕΛΒΟ), το οποίο είναι μία ειδική μορφή αθροίσματος γινομένων.

Η μορφή αθροίσματος των ΕΛΒΟ για κάθε συνάρτηση είναι μοναδική. Αν έχουμε τον πίνακα αληθείας μιας συνάρτησης, μπορούμε να γράψουμε το άθροισμα των ΕΛΒΟ, απλώς αθροίζοντας τις σειρές του πίνακα, όπου η έξοδος της συνάρτησης είναι 1.

x	y	z	$f(x,y,z)$	$\bar{f}(x,y,z)$
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

Έστω λοιπόν ο παραπάνω πίνακας αλήθειας, στον οποίο θεωρούμε δεδομένη τη συνάρτηση f και συνεπώς και τη συμπληρωματική της:

Έτσι εύκολα προκύπτουν οι συναρτήσεις για την f και την \bar{f} αντίστοιχα :

$$\begin{aligned}
 f &= \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz + x\bar{y}\bar{z} & \bar{f} &= x\bar{y}\bar{z} + x\bar{y}z + xyz \\
 &= m_0 + m_1 + m_2 + m_3 + m_6 & &= m_4 + m_5 + m_7 \\
 &= \sum m(0,1,2,3,6) & &= \sum m(4,5,7)
 \end{aligned}$$

Η \bar{f} περιέχει όλους τους ΕΛΒΟ που δεν περιέχονται στην f .

3.9.1.2 Μεγιστοβάθμιος όρος (Maxterm) - MEBO

Το συμπλήρωμα των ΕΛΒΟ (η δυική ιδέα) είναι το γινόμενο αθροισμάτων. Στο γινόμενο αθροισμάτων (Product Of Sum-POS) πρέπει να γνωρίζουμε ότι η έκφραση αυτή περιέχει μόνο πράξεις AND (γινόμενο). Κάθε παράγοντάς της αποτελείται από ένα άθροισμα όρων.

Ένας μεγιστοβάθμιος όρος (MEBO) είναι το άθροισμα όρων, στους οποίους κάθε μεταβλητή είσοδος εμφανίζεται μόνο μια φορά με την κανονική ή τη συμπληρωματική της μορφή.

Μια συνάρτηση με n μεταβλητές έχει 2^n μεγιστοβάθμιους όρους. Οι MEBO για μια συνάρτηση τριών μεταβλητών $f(x,y,z)$ είναι:

1 ^{ος}	$\bar{x} + \bar{y} + \bar{z}$	2 ^{ος}	$\bar{x} + \bar{y} + z$	3 ^{ος}	$\bar{x} + y + \bar{z}$	4 ^{ος}	$\bar{x} + y + z$
5 ^{ος}	$x + \bar{y} + \bar{z}$	6 ^{ος}	$x + \bar{y} + z$	7 ^{ος}	$x + y + \bar{z}$	8 ^{ος}	$x + y + z$

Κάθε MEBO είναι ψευδής για έναν ακριβώς συνδυασμό των εισόδων:

MEBO (Μεταβλητές x,y,z)	Είναι ψευδής όταν...	Συντομογραφία
$x + y + z$	$x=0, y=0, z=0$	M_0
$x + y + \bar{z}$	$x=0, y=0, z=1$	M_1
$x + \bar{y} + z$	$x=0, y=1, z=0$	M_2
$x + \bar{y} + \bar{z}$	$x=0, y=1, z=1$	M_3
$\bar{x} + y + z$	$x=1, y=0, z=0$	M_4
$\bar{x} + y + \bar{z}$	$x=1, y=0, z=1$	M_5
$\bar{x} + \bar{y} + z$	$x=1, y=1, z=0$	M_6
$\bar{x} + \bar{y} + \bar{z}$	$x=1, y=1, z=1$	M_7

Κάθε συνάρτηση μπορεί να γραφτεί σαν ένα μοναδικό γινόμενο MEBO. Αν έχουμε για μία συνάρτηση τον πίνακα αληθείας, μπορούμε να την εκφράσουμε σαν ένα γινόμενο MEBO, επιλέγοντας τις σειρές του πίνακα όπου η έξοδος είναι 0. (Προσοχή! Γράφουμε τους πραγματικούς όρους)

x	y	z	$f(x,y,z)$	$\bar{f}(x,y,z)$
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

$$\begin{aligned}
 f &= (\bar{x} + y + z)(\bar{x} + y + \bar{z})(\bar{x} + \bar{y} + \bar{z}) & \bar{f} &= (x + y + z)(x + y + \bar{z})(x + \bar{y} + z)(x + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z) \\
 &= M_4 M_5 M_7 & &= M_0 M_1 M_2 M_3 M_6 \\
 &= \prod M(4,5,7) & &= \prod M(0,1,2,3,6)
 \end{aligned}$$

Και εδώ η \bar{f} περιέχει όλους τους MEBO που δεν περιέχονται στην f .

Εύκολα παρατηρούμε ότι οι ΕΛΒΟ και οι MEBO αλληλοσυσχετίζονται. Κάθε ΕΛΒΟ m_i είναι το συμπλήρωμα του αντιστοίχου MEBO M_i .

Για παράδειγμα: $\bar{m}_4 = M_4$ καθώς $\overline{x\bar{y}\bar{z}} = \bar{x} + y + z$

Με βάση τον πίνακα 3.4, που ακολουθεί, μπορούμε εύκολα να μετατρέψουμε ένα άθροισμα ΕΛΒΟ σε ένα γινόμενο MEBO.

Από τα προηγούμενα: $f = \sum m(0,1,2,3,6)$ και $\bar{f} = \sum m(4,5,7) = m_4 + m_5 + m_7$

Η συμπληρωματική συνάρτηση $\bar{f} = \overline{m_4 + m_5 + m_7}$ έτσι $f = \bar{m}_4 \bar{m}_5 \bar{m}_7$ [θεώρημα De Morgan]
 $= M_4 M_5 M_7$ [βάσει του προηγούμενου πίνακα] = $\prod M(4,5,7)$

Γενικότερα απλώς αντικαθιστούμε τους ΕΛΒΟ με τους MEBO, χρησιμοποιώντας τους MEBO που δεν εμφανίζονται στο άθροισμα των ΕΛΒΟ:

$$f = \sum m(0,1,2,3,6) = \prod M(4,5,7)$$

Η ίδια μεθοδολογία χρησιμοποιείται και για την αντίστροφη μετατροπή από γινόμενο MEBO σε άθροισμα ΕΛΒΟ.

MEBO	Συντομογραφία	ΕΛΒΟ	Συντομογραφία
$x + y + z$	M_0	$\overline{x y z}$	m_0
$x + y + \bar{z}$	M_1	$\overline{x y \bar{z}}$	m_1
$x + \bar{y} + z$	M_2	$\overline{x \bar{y} z}$	m_2
$x + \bar{y} + \bar{z}$	M_3	$\overline{x \bar{y} \bar{z}}$	m_3
$\bar{x} + y + z$	M_4	$\overline{\bar{x} y z}$	m_4
$\bar{x} + y + \bar{z}$	M_5	$\overline{\bar{x} y \bar{z}}$	m_5
$\bar{x} + \bar{y} + z$	M_6	$\overline{\bar{x} \bar{y} z}$	m_6
$\bar{x} + \bar{y} + \bar{z}$	M_7	$\overline{\bar{x} \bar{y} \bar{z}}$	m_7

Πίνακας 3.4. Αντιστοίχιση μεταξύ των ΕΛΒΟ και των MEBO

Κανονική μορφή (Κανονικοποίηση)

Κανονική λέγεται η συνάρτηση στην οποία όλες οι μεταβλητές, ή το συμπλήρωμά τους, εμφανίζονται μόνο μία φορά σε κάθε όρο της υπό μορφή αθροίσματος ελαχιστοβάθμιων όρων ή υπό μορφή γινομένου μεγιστοβάθμιων όρων.

Χρησιμοποιούμε την επιμεριστική ιδιότητα με ανάστροφο τρόπο, για να προσθέσουμε τις μεταβλητές που δε συμπεριλαμβάνονται στην αρχική συνάρτηση. Αυτή η διαδικασία γίνεται σπάνια, αλλά είναι αρκετά χρήσιμη.

Έστω λοιπόν η συνάρτηση:

$$\begin{aligned} xy + \bar{y}z + xz &= (xy \cdot 1) + (\bar{y}z \cdot 1) + (xz \cdot 1) = (xy \cdot (z + \bar{z})) + (\bar{y}z \cdot (x + \bar{x})) + (xz \cdot (y + \bar{y})) \\ &= (xyz + xy\bar{z}) + (\bar{y}zx + \bar{y}z\bar{x}) + (xzy + xz\bar{y}) = xyz + xy\bar{z} + \bar{y}zx + \bar{y}z\bar{x} + xzy + xz\bar{y} \end{aligned}$$

Και στις δύο περιπτώσεις κάνουμε πιο πολύπλοκη την αρχική μας έκφραση. Η έκφραση που προκύπτει είναι μεγαλύτερη από την αρχική, αλλά, έχοντας όλους τους μεμονωμένους ΕΛΒΟ, είναι πιο εύκολο να τους συνδυάσουμε σε ένα χάρτη Karnaugh που χρησιμοποιείται για την απλοποίηση, και τον οποίο θα αναλύσουμε λεπτομερώς παρακάτω.

3.9.2. Άλγεβρική ελαχιστοποίηση λογικών συναρτήσεων

Η ελαχιστοποίηση επιτυγχάνεται με τη βοήθεια των θεωρημάτων και αξιωμάτων της άλγεβρας Boole και μέσω καταλλήλου μετασχηματισμού των παραστάσεων.

Όπως ήδη έχουμε δει, το λογικό γινόμενο δύο ή περισσότερων λογικών μεταβλητών ονομάζεται *ελαχιστοβάθμιος όρος*, ενώ το λογικό άθροισμα δύο ή περισσότερων λογικών μεταβλητών ονομάζεται *μεγιστοβάθμιος όρος*.

Ακολουθούν ορισμένα παραδείγματα ελαχιστοποίησης λογικών συναρτήσεων.

A) Έστω η συνάρτηση:

$$T = X \cdot Y \cdot Z \cdot W + X \cdot \bar{Y} \cdot Z \cdot W + X \cdot Y \cdot \bar{Z} \cdot W + X \cdot \bar{Y} \cdot \bar{Z} \cdot W + \bar{X} \cdot Y \cdot Z \cdot W + X \cdot \bar{Y} \cdot \bar{Z} \cdot \bar{W}$$

Μια πρώτη απλοποίηση επιτυγχάνεται ως εξής:

$$T = X \cdot W \cdot (Y \cdot Z + \bar{Y} \cdot Z + Y \cdot \bar{Z} + \bar{Y} \cdot \bar{Z}) + \bar{X} \cdot Y \cdot Z \cdot W + X \cdot \bar{Y} \cdot \bar{Z} \cdot \bar{W} \Rightarrow$$

$$T = X \cdot W + \bar{X} \cdot Y \cdot Z \cdot W + X \cdot \bar{Y} \cdot \bar{Z} \cdot \bar{W}$$

και αυτό, γιατί η παράσταση μέσα στην παρένθεση ισούται με τη μονάδα, αφού περιέχει το άθροισμα όλων των δυνατών συνδυασμών δύο λογικών μεταβλητών.

Η τελευταία έκφραση μπορεί να απλοποιηθεί περαιτέρω, θέτοντας στη θέση του $X \cdot W$ την έκφραση $X \cdot W + X \cdot \bar{W}$. Θα έχουμε επομένως:

$$T = X \cdot W + X \cdot \bar{W} + \bar{X} \cdot Y \cdot Z \cdot W + X \cdot \bar{Y} \cdot \bar{Z} \cdot \bar{W} \Rightarrow$$

$$T = X \cdot (\bar{Y} \cdot \bar{Z} \cdot \bar{W} + W) + W \cdot (\bar{X} \cdot Y \cdot Z + X) \Rightarrow \quad (\text{Θεώρημα } 4^0)$$

$$T = X \cdot (\bar{Y} \cdot \bar{Z} + W) + W \cdot (Y \cdot Z + X) \Rightarrow$$

$$T = X \cdot \bar{Y} \cdot \bar{Z} + X \cdot W + Y \cdot Z \cdot W + X \cdot W \Rightarrow$$

$$T = X \cdot \bar{Y} \cdot \bar{Z} + X \cdot W + Y \cdot Z \cdot W = X \cdot (\bar{Y} \cdot \bar{Z} + W) + Y \cdot Z \cdot W \quad \mathbf{(3-19)}$$

έκφραση η οποία είναι απλοποιημένη σχετικά με την αρχική.

B) Να απλοποιήσετε τη συνάρτηση: $T = A \cdot \bar{B} + \bar{A} \cdot B$

Εφαρμόζοντας το θεώρημα De Morgan, θα έχουμε:

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

3.9.3. Μέθοδος ελαχιστοποίησης με διαγράμματα Venn

Η γραφική απεικόνιση των λογικών συναρτήσεων Boole με τη βοήθεια των διαγραμμάτων του Venn έχει σαν αποτέλεσμα να εμφανίζει τις «καταστάσεις γειτνιάσεως» (γειτονικές καταστάσεις), δηλαδή τους όρους της λογικής συνάρτησης οι οποίοι διαφέρουν μεταξύ τους ως προς την τιμή μιας μόνο μεταβλητής.

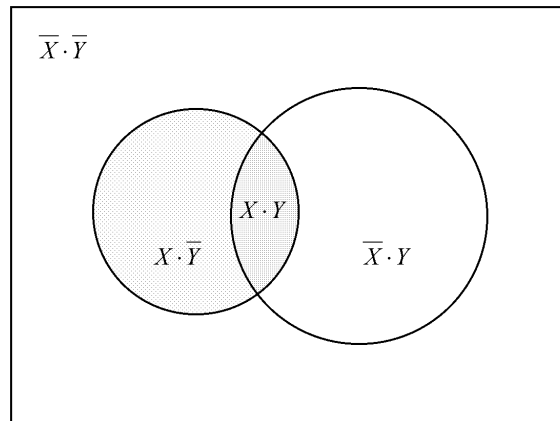
Στο Σχήμα 3.17 παριστάνεται μια συνάρτηση δύο μεταβλητών. Στο σχήμα εμφανίζονται οι τέσσερις περιοχές με τις αντίστοιχες μεταβλητές που παριστάνουν, δηλαδή: $\mathbf{XY, X\bar{Y}, \bar{X}Y, \bar{X}\bar{Y}}$. Η ζώνη που περικλείει το εμβαδόν των δύο κύκλων είναι η ζώνη $\mathbf{X + Y = X \cdot \bar{Y} + X \cdot Y + \bar{X} \cdot Y}$.

Πράγματι: $X \cdot \bar{Y} + X \cdot Y + \bar{X} \cdot Y = X \cdot (\bar{Y} + Y) + \bar{X} \cdot Y = X + \bar{X} \cdot Y = X + Y$ Η ζώνη αυτή είναι συμπληρωματική της $\bar{X} \cdot \bar{Y}$, όπως φαίνεται και στο σχήμα 3.17, που ακολουθεί.

Χρησιμοποιώντας το σχήμα 3.17, εύκολα συνάγουμε το συμπέρασμα ότι σε άθροισμα γειτονικών ζωνών απαλείφεται η μεταβλητή, η οποία εμφανίζεται με διαφορετική τιμή στις δύο ζώνες.

Στο σχήμα 3.18 απεικονίζεται μια συνάρτηση τριών μεταβλητών X, Y, Z . Διακρίνουμε εύκολα τις 8 ζώνες που αντιστοιχούν στους 8 δυνατούς συνδυασμούς. Και εδώ εύκολα αποδεικνύεται ότι η ζώνη, η περικλειόμενη από το εμβαδόν των τριών κύκλων, ισούται με $X+Y+Z$, πράγματι

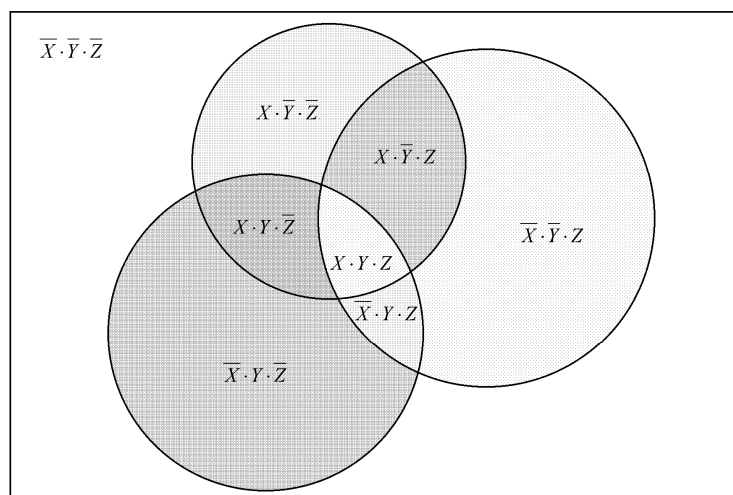
$$\begin{aligned} T &= X \cdot \bar{Y} \cdot \bar{Z} + \bar{X} \cdot Y \cdot \bar{Z} + \bar{X} \cdot \bar{Y} \cdot Z + X \cdot \bar{Y} \cdot Z + X \cdot Y \cdot \bar{Z} + \bar{X} \cdot Y \cdot Z + X \cdot Y \cdot Z = \\ &= X \cdot \bar{Z} \cdot (Y + \bar{Y}) + \bar{X} \cdot Z \cdot (Y + \bar{Y}) + \bar{X} \cdot Y \cdot (Z + \bar{Z}) + \bar{X} \cdot \bar{Y} \cdot Z = \end{aligned}$$



Σχήμα 3.17. Οι τέσσερις περιοχές μιας συνάρτησης δύο μεταβλητών

$$\begin{aligned} T &= X \cdot \bar{Z} + X \cdot Z + \bar{X} \cdot Y + \bar{X} \cdot \bar{Y} \cdot Z = X + \bar{X} \cdot Y + \bar{X} \cdot \bar{Y} \cdot Z = X + \bar{X} \cdot (Y + \bar{Y} \cdot Z) \\ &= X + \bar{X} \cdot (Z + Y) = X + \bar{X} \cdot Z + \bar{X} \cdot Y = X + Y + \bar{X} \cdot Z = X + Y + Z \quad (3-23) \end{aligned}$$

Κατ' αναλογία, οι ζώνες που περικλείονται από το εμβαδόν δύο μόνο κύκλων, ισούται αντίστοιχα με $X+Y, X+Z, Y+Z$.



Σχήμα 3.18. Οι οκτώ περιοχές του διαγράμματος Venn των συναρτήσεων τριών μεταβλητών

Για συναρτήσεις με περισσότερες των τριών μεταβλητών, η απεικόνιση με κύκλους παρουσιάζει δυσκολίες. Στην περίπτωση αυτή χρησιμοποιούνται κυρίως πίνακες. Οι πίνακες πρέπει να δημιουργηθούν έτσι, ώστε κάθε συνδυασμός να γειτνιάζει με τέσσερις άλλους συνδυασμούς, οι οποίοι θα διαφέρουν κατά την τιμή μιας μόνο μεταβλητής.

3.9.4. Μέθοδοι ελαχιστοποίησης με τους χάρτες Karnaugh

Οι χάρτες Karnaugh είναι πίνακες Veitch ελάχιστα τροποποιημένοι, ώστε να μπορούν να παραστήσουν συνδυασμούς n μεταβλητών. Αν και η μέθοδος που βασίζεται σ' αυτούς τους πίνακες μπορεί να χρησιμοποιηθεί για n μεταβλητές, χρησιμοποιείται για απλοποιήσεις συναρτήσεων το πολύ έξι μεταβλητών.

Ένας πρώτος τρόπος απεικόνισης των μεταβλητών του χάρτη Karnaugh, με βάση τους πίνακες Veitch για τρεις μεταβλητές, είναι ο ακόλουθος: (εννοείται ότι όπου δεν υπάρχει η μεταβλητή χρησιμοποιούμε το συμπλήρωμά της). Παρατηρούμε λοιπόν στον πίνακα 3.5α ότι π.χ. αριστερά και δεξιά της μεταβλητής Z οι όροι εμφανίζονται με το συμπλήρωμα της μεταβλητής Z. Το ίδιο ισχύει και για τις άλλες μεταβλητές του πίνακα. Συνεπώς στην πρώτη γραμμή οι όροι περιέχουν το συμπλήρωμα της μεταβλητής X και οι δύο αριστεροί κάθετοι όροι περιέχουν το συμπλήρωμα της μεταβλητής Y.

		Y	
	$\bar{x}y\bar{z}$	$\bar{x}yz$	
X	$x\bar{y}\bar{z}$	$x\bar{y}z$	xyz
		Z	

Με τη χρήση ελαχιστοβάθμιων όρων θα έχουμε την ακόλουθη απεικόνιση του χάρτη:

		Y	
	m_0	m_1	m_3
X	m_4	m_5	m_7
		Z	

Πίνακας 3.5α. Χάρτης Karnaugh – 1η μορφή

Είναι προφανές ότι η αναπαράσταση εξασφαλίζει ότι γειτονικά τετράγωνα έχουν κοινές μεταβλητές και ο κάθε γειτονικός όρος διαφέρει από τον άλλο σε μία μόνο μεταβλητή.

Χάρτης Karnaugh τεσσάρων μεταβλητών

Θα ξεκινήσουμε με την απεικόνιση με τις τέσσερις μεταβλητές (έστω w, x, y, z) λίγο διαφορετική από την συνηθισμένη:

		Y		
	$\bar{w}\bar{x}\bar{y}\bar{z}$	$\bar{w}\bar{x}y\bar{z}$	$\bar{w}x\bar{y}z$	$\bar{w}xy\bar{z}$
	$\bar{w}x\bar{y}\bar{z}$	$\bar{w}x\bar{y}z$	$\bar{w}xy\bar{z}$	$\bar{w}xyz$
W	$wx\bar{y}\bar{z}$	$wx\bar{y}z$	$wxy\bar{z}$	$wxyz$
	$wx\bar{y}\bar{z}$	$wx\bar{y}z$	$wx\bar{y}z$	$wxyz$
		Z		

		Y		
	m_0	m_1	m_3	m_2
	m_4	m_5	m_7	m_6
W	m_{12}	m_{13}	m_{15}	m_{14}
	m_8	m_9	m_{11}	m_{10}
		Z		

Ένας άλλος τρόπος αναπαράστασης (ο πλέον συνηθισμένος) του χάρτη Karnaugh τεσσάρων μεταβλητών δίνεται στον πίνακα 3.5β που ακολουθεί:

Εύκολα διαπιστώνουμε ότι ο κάθε γειτονικός όρος διαφέρει από τον άλλο σε ένα μόνο δυαδικό ψηφίο (μία μεταβλητή). Έτσι, για τον παρακάτω χάρτη των 16 θέσεων, έχουμε τους συνδυασμούς 00, 01, 11, 10. Ισοδύναμα θα μπορούσαμε να χρησιμοποιήσουμε την ακολουθία 11,10, 00, 01, ή οποιαδήποτε άλλη που πληροί την προϋπόθεση μεταξύ των γειτονικών όρων.

Σε καθεμία από τις 16 θέσεις του πίνακα αντιστοιχούμε ένα ΕΛΒΟ όρο.

Παρατηρούμε ότι οι ΕΛΒΟ στην τρίτη και την τέταρτη στήλη και στην τρίτη και τέταρτη γραμμή αλλάζουν μεταξύ τους. Η ομαδοποίηση των ΕΛΒΟ γίνεται με παρόμοιο τρόπο, ανεξάρτητα του αριθμού των μεταβλητών, έχοντας υπόψη μας ότι μπορούμε να έχουμε ομάδες από 1, 2, 4, 8 ή 16 ΕΛΒΟ.

Επίσης δεν ξεχνάμε ότι μπορούμε να «διπλώσουμε» (switch around) και τις τέσσερις πλευρές του χάρτη. (Η απεικόνιση που θα επιλέξει ο κάθε χρήστης στην υλοποίησή του είναι αυθαίρετη, εφόσον ικανοποιούνται οι προϋποθέσεις των γειτονικών όρων).

	XY	00	01	11	10
WZ					
00					
01					
11					
10					

Πίνακας 3.5β. Χάρτης Karnaugh – Συνήθης μορφή αναπαράστασης

Στο χάρτη Karnaugh και στις αντίστοιχες θέσεις του θέτουμε 1 ή 0, ανάλογα με το αν ο όρος της συνάρτησης που προορίζεται να απλοποιηθεί υπάρχει ή όχι. Έτσι η συνάρτηση:

$T = A \cdot B \cdot \bar{C} \cdot D + A \cdot \bar{B} \cdot C \cdot \bar{D} + \bar{A} \cdot B \cdot C \cdot D + \bar{A} \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot \bar{B} \cdot C \cdot \bar{D} + \bar{A} \cdot B \cdot \bar{C} \cdot D$ απεικονίζεται στο χάρτη Karnaugh του πίνακα 3.6:

Στις κενές θέσεις τοποθετούνται τα μηδενικά, τα οποία όμως παραλείπονται για την καλύτερη ευκρίνεια του χάρτη. Σε περίπτωση όπου η προς απλοποίηση συνάρτηση έχει και όρους με λιγότερες από τέσσερις μεταβλητές, τότε θα σημειώνονται άσσοι σε όλες τις θέσεις που ορίζουν οι συνδυασμοί των απόντων όρων. Εάν, για παράδειγμα, είχαμε τον όρο ABC ή τον όρο AD, ο αντίστοιχος χάρτης δίνεται στους πίνακες 3.7α και 3.7β.

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

Πίνακας 3.6. Η συνάρτηση $T = A \cdot B \cdot \bar{C} \cdot D + A \cdot \bar{B} \cdot C \cdot \bar{D} + \bar{A} \cdot B \cdot C \cdot D + \bar{A} \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot \bar{B} \cdot C \cdot \bar{D} + \bar{A} \cdot B \cdot \bar{C} \cdot D$

Από τους πίνακες που ακολουθούν φαίνεται καθαρά ότι οι ομάδες των γειτονικών άσπων παίζουν σημαντικό ρόλο στην απλοποίηση μιας συνάρτησης. Επαναλαμβάνουμε ότι, αν μια συνάρτηση τεσσάρων μεταβλητών έχει κάποιον όρο αποτελούμενο από τρεις μεταβλητές, αντιστοιχούν σε αυτόν δύο γειτνιαζόντες άσσοι. Κατά ανάλογο τρόπο, σε κάποιον όρο αποτελούμενο από δύο μεταβλητές, αντιστοιχούν τέσσερις γειτνιαζόντες άσσοι, ενώ σ' έναν όρο αποτελούμενο από μια μεταβλητή αντιστοιχούν οκτώ άσσοι.

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

Πίνακας 3.7α. Παράσταση του όρου ABC

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

Πίνακας 3.7β. Παράσταση του όρου AD

Η διαδικασία που ακολουθείται για την απλοποίηση μιας λογικής συνάρτησης περιγράφεται παρακάτω:

- Παριστάνεται η συνάρτηση στο χάρτη Karnaugh.
- Βρίσκουμε τους υπάρχοντες υποχάρτες με το μεγαλύτερο δυνατό αριθμό γειτνιαζόντων άσπων.
- Τυχόν επικαλύψεις υποχαρτών επιτρέπονται.

Συνιστάται στον αναγνώστη μια καλή εξάσκηση στο χάρτη Karnaugh έτσι, ώστε η εύρεση και επιλογή των υποχαρτών να είναι η καλύτερη δυνατή για την απλοποίηση των αντιστοιχών συναρτήσεων.

Για την καλύτερη κατανόηση της χρήσης του, θα δώσουμε ορισμένα παραδείγματα απλοποίησης με τη βοήθεια του χάρτη Karnaugh.

Παράδειγμα 1ο:

Έστω η παρακάτω συνάρτηση προς απλοποίηση:

$$T = \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}BCD + \overline{A}BC\overline{D} + \overline{A}BCD + \overline{A}BC\overline{D} \quad (3-24)$$

της οποίας ο αντίστοιχος χάρτης Karnaugh αποδίδεται στον πίνακα 3.8.

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

Πίνακας 3.8. Χάρτης Karnaugh της σχέσης 3-24

Η αντίστοιχη απλοποιημένη συνάρτηση, όπως προκύπτει με βάση τις παραπάνω ομαδοποιήσεις, έχει ως εξής: $T = \overline{A} \cdot \overline{D} + \overline{A} \cdot B \cdot C + A \cdot B \cdot D + \overline{B} \cdot \overline{C} \cdot \overline{D}$

Εάν η υπό μελέτη συνάρτηση έχει πολλούς άσπους, αυτή θα μπορούσε να γραφεί με βάση τα μηδενικά, δηλαδή γινόμενα αθροισμάτων (maxterm), αρκεί να δημιουργηθούν οι κατάλληλοι υποχάρτες μηδενικών.

Παράδειγμα 2ο: (Παράδειγμα όπου γίνεται χρήση των μηδενικών).

Έστω η συνάρτηση: $T = \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot B \cdot \overline{C} \cdot D + A \cdot \overline{B} \cdot \overline{C} \cdot D + A \cdot B \cdot \overline{C} \cdot D + A \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot \overline{B} \cdot C \cdot D + \overline{A} \cdot B \cdot C \cdot D + \overline{A} \cdot B \cdot C \cdot \overline{D}$ (3-25)

από την οποία προκύπτει ο πίνακας 3.9 που ακολουθεί:

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

Πίνακας 3.9. Χάρτης Karnaugh της σχέσης 3-25

Η απλοποιημένη συνάρτηση που προκύπτει είναι η: $\bar{T} = A \cdot C + \bar{B} \cdot \bar{D}$ από την οποία, χρησιμοποιώντας το θεώρημα του De Morgan, έχουμε: $T = (\bar{A} + \bar{C}) \cdot (B + D)$

Παράδειγμα 3ο:

Δίνεται η συνάρτηση: $f = \sum 0,2,4,6,7,10,13,14,15$. Χρησιμοποιώντας χάρτη Karnaugh, να βρείτε την ελάχιστη έκφραση του αθροίσματος γινομένων (SOP) και του γινομένου αθροισμάτων (POS).

Λύση

Αρχικά τοποθετούμε τους όρους που δόθηκαν στον πίνακα 3.10. Το άθροισμα των γινομένων δεν είναι τίποτε άλλο από τους γνωστούς μας ελαχιστοβάθμιους όρους, τα δε γινόμενα αθροισμάτων είναι οι μεγιστοβάθμιοι όροι.

Είναι προφανές ότι, για να υπολογίσουμε το άθροισμα των γινομένων, κάνουμε τις ομαδοποιήσεις των άσων της συνάρτησης σε υποχάρτες. Η τελική απλοποιημένη συνάρτηση προκύπτει από την απλοποίηση των όρων. Έτσι έχουμε: $f_1 = A \cdot B \cdot D + B \cdot C + \bar{C}\bar{D} + \bar{A}\bar{D}$

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

Πίνακας 3.10. Η συνάρτηση $f = \sum 0,2,4,6,7,10,13,14,15$ σε χάρτη Karnaugh

Για να προσδιορίσουμε τώρα τα γινόμενα αθροισμάτων, δουλεύουμε με τα μηδενικά, και κατόπιν, χρησιμοποιώντας τα θεωρήματα της Άλγεβρας Boole, παίρνουμε το συμπλήρωμα της συνάρτησης. Τελικά καταλήγουμε στην: $f_2 = (\bar{A} + C + D) \cdot (A + C + \bar{D}) \cdot (B + \bar{D})$

(Η απόδειξη και των δύο παραπάνω σχέσεων αφήνεται σαν άσκηση για τον αναγνώστη).

3.9.5. Χάρτης Karnaugh με περισσότερες από τέσσερις μεταβλητές

Αν ένας χάρτης Karnaugh περιλαμβάνει περισσότερες από τέσσερις μεταβλητές είναι προτιμότερο, αντί για ένα ενιαίο χάρτη, όπως αυτός του πίνακα 3.11 που ακολουθεί, να σχεδιάσουμε δύο ή περισσότερους χάρτες από τους οποίους ο καθένας θα έχει το πολύ τέσσερις μεταβλητές. Π.χ. αν έχουμε μια συνάρτηση με πέντε μεταβλητές, έστω οι **A, B, C, D, E**, τότε είναι καλύτερο να δημιουργήσουμε δύο χάρτες, όπως φαίνεται στον πίνακα 3.11, όπου στον αριστερό χάρτη έχουμε τη μεταβλητή \bar{E} και στο δεξιό τη μεταβλητή E . (Nashelsky, 1994; Κοσσίδας, 1996; Wakerly, 2006; Pritchard, 2015)

Οι ελάχιστοι όροι από 0 - 15 θα ανήκουν τότε στον αριστερό χάρτη, όπου θεωρούμε ότι $E=0$ και οι ελάχιστοι όροι από 16 - 31 θα ανήκουν στο δεξιό χάρτη, όπου έχουμε τη μεταβλητή $E=1$.

Για κάθε ένα χάρτη ισχύουν οι κανόνες που έχουν αναφερθεί προηγουμένως. Επιπλέον κάθε τετράγωνο στο χάρτη $E=0$ είναι γειτονικό με το αντίστοιχο στο χάρτη $E=1$, εφόσον βρίσκεται στην ίδια θέση υπερθέτοντας υποθετικά τον ένα χάρτη επάνω στον άλλο. Π.χ. το 2 είναι γειτονικό με το 18, το 13 με το 29, το 11 με το 27 κ.λπ. Στην περίπτωση των έξι μεταβλητών, οι αντίστοιχοι χάρτες που πρέπει να δημιουργηθούν είναι τέσσερις (βλέπε πίνακα 3.12α). Και γι' αυτούς θα ισχύουν τα ίδια, όπως και στους χάρτες των πέντε μεταβλητών.

	AB				
CD		00	01	11	10
00	0	1	3	2	
01	4	5	7	6	
11	12	13	15	14	
10	8	9	11	10	

	AB				
CD		00	01	11	10
00	16	17	19	18	
01	20	21	23	22	
11	28	29	31	30	
10	24	25	27	26	

Πίνακας 3.11. Διπλός χάρτης Karnaugh για συναρτήσεις με περισσότερες από τέσσερις μεταβλητές

Παράδειγμα 4ο:

Θα αναλύσουμε την περίπτωση όπου έχουμε μία συνάρτηση με έξι μεταβλητές. Έστω η συνάρτηση:

$$\begin{aligned}
 T = & \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \cdot F + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot E \cdot F + A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \cdot F + \\
 & + A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot E \cdot F + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D \cdot \bar{E} \cdot F + A \cdot \bar{B} \cdot \bar{C} \cdot D \cdot E \cdot F + \\
 & + \bar{A} \cdot \bar{B} \cdot C \cdot \bar{D} \cdot \bar{E} \cdot F + \bar{A} \cdot \bar{B} \cdot C \cdot D \cdot \bar{E} \cdot F + A \cdot \bar{B} \cdot C \cdot \bar{D} \cdot \bar{E} \cdot F + \\
 & + A \cdot \bar{B} \cdot C \cdot D \cdot \bar{E} \cdot F + \bar{A} \cdot \bar{B} \cdot C \cdot D \cdot E \cdot F + \bar{A} \cdot B \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \cdot F + \\
 & + \bar{A} \cdot B \cdot \bar{C} \cdot D \cdot \bar{E} \cdot F + \bar{A} \cdot B \cdot C \cdot \bar{D} \cdot \bar{E} \cdot F + \bar{A} \cdot B \cdot C \cdot D \cdot \bar{E} \cdot F + \\
 & + \bar{A} \cdot B \cdot C \cdot D \cdot E \cdot F + A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \cdot F + A \cdot \bar{B} \cdot \bar{C} \cdot D \cdot \bar{E} \cdot F + \\
 & + A \cdot \bar{B} \cdot C \cdot \bar{D} \cdot \bar{E} \cdot F + A \cdot \bar{B} \cdot C \cdot D \cdot E \cdot F
 \end{aligned}
 \tag{3-26}$$

Και στον ανωτέρω χάρτη για την ομαδοποίηση των όρων του εφαρμόζουμε τις προϋποθέσεις που χρησιμοποιήσαμε και στο χάρτη των τεσσάρων μεταβλητών. Έτσι ομαδοποιούμε σε ζεύγη ή τετράδες γειτονικούς όρους. Η απλοποίηση αφήνεται σαν άσκηση στον αναγνώστη.

Ας εξετάσουμε τώρα την παραπάνω περίπτωση, χρησιμοποιώντας τέσσερις χάρτες των τεσσάρων μεταβλητών (Πίνακας 3.12). Για την απλοποίηση χρησιμοποιούμε τις παρακάτω συνθήκες:

Δύο όροι είναι γειτονικοί, όχι μόνο αν πληρούν τις προϋποθέσεις που εφαρμόσαμε στους χάρτες των τεσσάρων μεταβλητών, αλλά και στην περίπτωση που ο ένας όρος, μετά την υπέρθεση γειτονικών χαρτών, βρίσκεται υπεράνω ή κάτω του άλλου αντίστοιχα. Έτσι, για παράδειγμα, ο όρος 5 είναι γειτονικός του 31 και αντίστοιχα γειτονικός του 37, αφού καταλαμβάνουν την ίδια θέση σε γειτονικό χάρτη των τεσσάρων μεταβλητών.

		\bar{E}						E			
\bar{F}	AB										
	CD	00	01	11	10	00	01	11	10		
	00	0	1	3	2	16	17	19	18		
	01	4	5	7	6	20	21	23	22		
11	12	13	15	14	28	29	31	30			
10	8	9	11	10	24	25	27	26			

F	AB								
	CD	00	01	11	10	00	01	11	10
	00	32	33	35	34	48	49	51	50
	01	36	37	39	38	52	53	55	54
11	44	45	47	46	60	61	63	62	
10	40	41	43	42	56	57	59	58	

Πίνακας 3.12. Χάρτης Karnaugh της σχέσης 3-26 με τέσσερις χάρτες (έξι μεταβλητές)

3.9.6. Μέθοδος ελαχιστοποίησης με τη μέθοδο Quine-Mc Cluskey

Στην περίπτωση στην οποία πρέπει να επιλύσουμε συναρτήσεις με περισσότερες από 6 μεταβλητές, τότε χρησιμοποιούμε τη μέθοδο Quine-Mc Cluskey. (Φραγκάκης, 1975; Nashelsky, 1994; Givone, 2002; Katz, 2005; Wakerly, 2006; Godse, & Godse, 2010; Mano, & Ciletti, 2014; Roth, 2014). Η μέθοδος αυτή δεν είναι γραφική μέθοδος, όπως ο χάρτης Karnaugh, και μπορεί εύκολα να προγραμματισθεί σε ένα ηλεκτρονικό υπολογιστή. Βασίζεται στη συστηματική και εξαντλητική διερεύνηση της σύγκρισης όλων των όρων της συνάρτησης, την οποία καλούμαστε να απλοποιήσουμε. Θα αναλύσουμε τη μεθοδολογία, δίνοντας ένα παράδειγμα με τέσσερις μεταβλητές:

Παράδειγμα 5ο:

Να απλοποιήσετε τη συνάρτηση: $f(A, B, C, D) = \bar{A}\bar{B}\bar{C} + \bar{A}BD + \bar{A}BC + \bar{A}BD + \bar{A}BC$

Λύση

Βήμα 1^ο

Αρχικά εκφράζουμε τη συνάρτηση στην κανονικοποιημένη της μορφή και, κατόπιν, αντιστοιχούμε κάθε όρο με τον ισοδύναμο δεκαδικό αριθμό.

$$f(A, B, C, D) = \bar{A}\bar{B}\bar{C} + \bar{A}BD + \bar{A}BC + \bar{A}BD + \bar{A}BC \Rightarrow$$

$$f(A, B, C, D) = \bar{A}\bar{B}\bar{C}(D + \bar{D}) + \bar{A}B(C + \bar{C})D + \bar{A}\bar{B}\bar{C}(D + \bar{D}) + \bar{A}B(\bar{C} + C)D + \bar{A}BC(D + \bar{D})$$

$$f(A, B, C, D) = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}BDC + \bar{A}B\bar{D}C + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}$$

$$+ \bar{A}BDC + \bar{A}B\bar{D}C + \bar{A}BCD + \bar{A}BC\bar{D}$$

Ξαναγράφοντας τη συνάρτηση αλλά τοποθετώντας τις μεταβλητές στην αντίστοιχη θέση έχουμε:

$$f(A, B, C, D) = \overline{A}BCD + A\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD$$

από την οποία απαλείφοντας τους όμοιους όρους και κάνοντας απλοποίηση έχουμε:

$$f(A, B, C, D) = \overline{A}BCD + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD + \overline{A}BCD$$

Οι όροι είναι λοιπόν, από τον μικρότερο προς τον μεγαλύτερο, 4, 5, 6, 7, 8, 9, 11.

Βήμα 2°

Ομαδοποιούμε τους όρους ανάλογα με τον αριθμό των άσσων τους. Έτσι οι όροι της πρώτης ομάδας περιέχουν έναν άσσο, της δεύτερης δύο, κ.ο.κ. Προκύπτει ο πίνακας 3.13α που ακολουθεί.

Βήμα 3°

Συγκρίνουμε κάθε όρο της πρώτης ομάδας με όλους τους όρους της επόμενης ομάδας. Σε κάθε σύγκριση ελέγχουμε αν οι δύο όροι διαφέρουν σε μία μόνο μεταβλητή. Σε αυτή την περίπτωση δημιουργούμε ένα νέο όρο με τις κοινές μεταβλητές των δύο αρχικών όρων. Στη θέση της διαφορετικής μεταβλητής θέτουμε μια παύλα (-).

Το βήμα αυτό εκτελείται συνεχώς, μέχρι ότου να συγκρίνουμε όλους τους όρους της κάθε ομάδας με τους όρους της επόμενης ομάδας και να μην υπάρχει δυνατότητα περαιτέρω συγκρίσεων. Ο αρχικός πίνακας, που προκύπτει, είναι ο πίνακας 3.13β:

Όροι της Συνάρτησης	Δεκαδικός αριθμός	Δυαδική μορφή	Χρήση
$\overline{A}BCD$	4	0100	ναι
$A\overline{B}CD$	8	1000	ναι
$\overline{A}B\overline{C}D$	5	0101	ναι
$\overline{A}BC\overline{D}$	6	0110	ναι
$\overline{A}BCD$	9	1001	ναι
$\overline{A}BCD$	7	0111	ναι
$\overline{A}BCD$	11	1011	ναι

Πίνακας 3.13α. Ομαδοποίηση των όρων ανάλογα με το πλήθος των άσσων τους

Συνδυασμοί	Δυαδική μορφή	Χρήση
4,5	010-	ναι
4,6	01-0	ναι
8,9	100-	Κ
5,7	01-1	ναι
6,7	011-	ναι
9,11	10-1	Λ

Πίνακας 3.13β. Συγχώνευση γειτονικών όρων

Με βάση τον παραπάνω πίνακα συνεχίζουμε τις συγκρίσεις και έχουμε:

Συνδυασμοί	Δυαδική μορφή	Χρήση
4,5,6,7	01--	Μ

Βήμα 4°

Αφού εξαντλήσουμε όλους τους όρους, σε όσους από αυτούς δεν έχουμε χρησιμοποιήσει τοποθετούμε ένα γράμμα, το οποίο θα χρησιμοποιήσουμε στο επόμενο βήμα μας. Οι όροι αυτοί ονομάζονται *πρώτοι συνεπάγοντες (prime implicants)*:

Βήμα 5°

Κατασκευάζουμε τον πίνακα των πρώτων συνεπαγόντων όρων, του οποίου οι γραμμές αποτελούνται από τους πρώτους συνεπάγοντες, και οι στήλες από τους αρχικούς όρους της συνάρτησης. Στα σημεία τομής των συνεπαγόντων όρων με τους αντίστοιχους όρους της αρχικής συνάρτησης θέτουμε ένα X.

		4	5	6	7	8	9	11
K(8,9)						*	*	
Λ(9,11)							*	*
M(4,5,6,7)		*	*	*	*			

Βήμα 6°

Χρησιμοποιώντας τον τελευταίο πίνακα, βρίσκουμε τους *ουσιώδεις όρους (essential terms)*, οι οποίοι ορίζονται ως οι πρώτοι συνεπάγοντες όροι στους οποίους, και μόνο σε αυτούς, περιέχεται ένας από τους όρους της προς απλοποίηση συνάρτησης. Ουσιώδεις λοιπόν είναι οι όροι οι οποίοι έχουν στην κατακόρυφο μόνο ένα X. Στην παραπάνω συνάρτηση διαπιστώνουμε ότι όλοι οι όροι είναι ουσιώδεις, καθώς όλοι έχουν από ένα όρο που δε συμπεριλαμβάνεται σε άλλο πρώτο συνεπάγοντα. Ο ουσιώδεις όροι πρέπει να καλύπτουν όλους τους όρους της αρχικής συνάρτησης.

Η τελική συνάρτηση είναι: $f(A, B, C, D) = K + \Lambda + M$

Όπου $K = \overline{ABC}$, $\Lambda = \overline{ABD}$ και τέλος $M = \overline{AB}$ οπότε $f(A, B, C, D) = \overline{ABC} + \overline{ABD} + \overline{AB}$. Εύκολα μπορούμε να επαληθεύσουμε το τελικό αποτέλεσμα, χρησιμοποιώντας χάρτη Karnaugh.

3.9.7. Αδιάφοροι όροι

Στα λογικά κυκλώματα που εξετάσαμε μέχρι τώρα, για καθέναν από τους συνδυασμούς των μεταβλητών εισόδου είχαμε πάντοτε μια συγκεκριμένη τιμή εξόδου, 0 ή 1. Υπάρχουν, όμως, περιπτώσεις που για μερικούς συνδυασμούς δε μας ενδιαφέρει το ποια θα είναι η λογική κατάσταση της εξόδου του κυκλώματος. Οι συνδυασμοί αυτοί, στους οποίους η τιμή τους δεν είναι καθορισμένη, και, συνεπώς, δεν επηρεάζει τη συνάρτηση εξόδου, λέγονται *αδιάφοροι όροι (don't care)* και παριστάνονται στους χάρτες του Karnaugh με το σύμβολο X.

Ο τρόπος με τον οποίο αντιμετωπίζουμε τους όρους αυτούς σ' ένα χάρτη Karnaugh είναι πολύ απλός. Έχοντας τοποθετήσει τους άσους στα τετράγωνα εκείνα που προβλέπονται από τον πίνακα αλήθειας, τοποθετούμε αμέσως μετά τους αδιάφορους όρους με το X. Ελέγχουμε αν μερικά από τα X γειτονεύουν με τους άσους (1). Αν πράγματι συμβαίνει αυτό, θεωρούμε ότι τα συγκεκριμένα X έχουν τιμή 1 και κάνουμε τις απλοποιήσεις, όπως ξέρουμε. Εκείνα που δε γειτονεύουν με 1, τα αγνοούμε.

Παράδειγμα 6°:

Ο πίνακας 3.14 περιγράφει παράδειγμα στο οποίο γίνεται χρήση των αδιάφορων όρων: Έστω ότι έχουμε τα δέκα ψηφία του δεκαδικού συστήματος αρίθμησης ($0_{10} - 9_{10}$) και θέλουμε να σχεδιάσουμε ένα λογικό κύκλωμα που στην έξοδό του θα παράγει λογικό 1 μόνο, όταν στην είσοδο του κυκλώματος υπάρχει ένας δυαδικός αριθμός που αντιπροσωπεύει έναν περιττό δεκαδικό. Δεδομένου ότι τα δέκα δεκαδικά ψηφία περιγράφονται με δέκα συνδυασμούς των τεσσάρων δυαδικών ψηφίων, ο πίνακας αλήθειας, που θα περιγράφει τη συνάρτηση, θα περιλαμβάνει $2^4 = 16$ γραμμές, εκ των οποίων μόνο οι 10 θα είναι δυνατόν να επιβληθούν στην είσοδο του κυκλώματος. Οι υπόλοιπες 6 είσοδοι αναφέρονται σε συνδυασμούς οι οποίοι δεν περιγράφουν αριθμούς του δεκαδικού συστήματος και, συνεπώς, είναι αδιάφοροι όροι.

ΔΕΚΑΔΙΚΟ	ΔΥΑΔΙΚΟ ABCD	ΕΞΟΔΟΣ T
0	0000	0
1	0001	1
2	0010	0
3	0011	1
4	0100	0
5	0101	1
6	0110	0
7	0111	1
8	1000	0
9	1001	1
10	1010	X
11	1011	X
12	1100	X
13	1101	X
14	1110	X
15	1111	X

Πίνακας 3.14. Όροι με αδιάφορες καταστάσεις

Στη συνέχεια σχεδιάζουμε το χάρτη Karnaugh που αντιστοιχεί στον πίνακα αλήθειας.

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

Από το χάρτη Karnaugh, αν δε λάβουμε υπόψη τους αδιάφορους όρους, προκύπτει η λογική συνάρτηση: $T = \overline{A}D + \overline{B}CD$ (άλλος τρόπος απεικόνισης που απαντάται στην βιβλιογραφία) ή αλλιώς $T = \overline{A} \cdot D + \overline{B} \cdot C \cdot D$, (έχουμε μία τετράδα όρων και ένα ζευγάρι).

Αν θεωρήσουμε τους αδιάφορους όρους που γειτονεύουν με 1 ίσους με 1, θα έχουμε μια απλούστερη μορφή: (έχουμε μία οκτάδα γειτονικών όρων) $T = \overline{A} \cdot D + A \cdot D = D$.

Άρα η χρήση των αδιάφορων όρων βοηθά στην ελαχιστοποίηση της συνάρτησης.

3.10. Ασκήσεις – Ερωτήσεις

ΕΡΩΤΗΣΕΙΣ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

- Κάθε «1» σε ένα χάρτη Karnaugh αντιπροσωπεύει:
 - A. HIGH για κάθε είσοδο κατάσταση του πίνακα αληθείας που παράγει έξοδο HIGH
 - B. έξοδο HIGH στο πίνακα αληθείας για όλους τους LOW συνδυασμούς εισόδων
 - Γ. έξοδο LOW για όλες τις HIGH πιθανές καταστάσεις εισόδου
 - Δ. κατάσταση ΑΔΙΑΦΟΡΗ για όλους τους πιθανούς συνδυασμούς εισόδων του πίνακα αληθείας

- Η αντιμεταθετική ιδιότητα της άλγεβρας Boole αναφέρει ότι: $A + B = A \times B$
 - A. Σωστό
 - B. Λάθος

- Να μετατρέψετε την παρακάτω έκφραση του SOP σε μια ισοδύναμη με POS.

$$X = ABC + \overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC}$$

- A. $X = (\overline{A} + \overline{B} + \overline{C})(\overline{A} + B + \overline{C})(\overline{A} + B + C)$
- B. $X = (A + B + C)(A + \overline{B} + \overline{C})(A + \overline{B} + C)$
- Γ. $X = (\overline{A} + \overline{B} + \overline{C})(A + \overline{B} + C)(\overline{A} + \overline{B} + C)$
- Δ. $X = (A + B + C)(\overline{A} + B + \overline{C})(A + \overline{B} + C)$

- Όταν ομαδοποιούμε κελιά σε ένα χάρτη Karnaugh, τα κελιά πρέπει να συνδυάζονται σε ομάδες των:
 - A. 2
 - B. 1, 2, 4, 8 κ.τ.λ.
 - Γ. 4
 - Δ. 3

- Στη συνάρτηση $X = \overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC}$ αντιστοιχεί ο χάρτης Karnaugh

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

(A)

(B)

(C)

(D)

- A. (A)
- B. (B)
- Γ. (C)
- Δ. (D)

- Ποια πρόταση περιγράφει καλύτερα το χάρτη Karnaugh;
 - A. Ο χάρτης Karnaugh χρησιμοποιείται, για να αντικαταστήσει τους κανόνες της Άλγεβρας Boole
 - B. Ο χάρτης Karnaugh απαλείφει την ανάγκη για πύλες NAND και NOR
 - Γ. Τα συμπληρώματα των μεταβλητών θα απαλειφθούν με το χάρτη Karnaugh
 - Δ. Ο χάρτης Karnaugh αποτελεί ένα εύκολο τρόπο απλοποίησης των Boolean εκφράσεων

- Ποιος από τους παρακάτω συνδυασμούς δεν μπορεί να συμπεριληφθεί σε χάρτη Karnaugh;
 - A. Γωνίες στην ίδια γραμμή
 - B. Γωνίες στην ίδια στήλη
 - Γ. Διαγώνιος
 - Δ. Επικαλυπτόμενοι συνδυασμοί

- Ποιο από τα παρακάτω παραδείγματα εκφράζει ην επιμεριστική ιδιότητα της άλγεβρας Boole;
 - A. $(A + B) + C = A + (B + C)$
 - B. $A(B + C) = AB + AC$
 - Γ. $A + (B + C) = AB + AC$
 - Δ. $A(BC) = (AB) + C$

- Να καθορίσετε τις δυαδικές τιμές των μεταβλητών για τις οποίες η έκφραση των POS

$$X = (A + \overline{B} + C)(\overline{A} + B + \overline{C}) = 0.$$

- A. $(0 + 1 + 0)(1 + 0 + 1)$
- B. $(1 + 1 + 1)(0 + 0 + 0)$
- Γ. $(0 + 0 + 0)(1 + 0 + 1)$
- Δ. $(1 + 1 + 0)(1 + 0 + 0)$

10. Εφαρμόζοντας το θεώρημα De Morgan στην έκφραση $X = \overline{ABC}$ έχουμε ως αποτέλεσμα:
 Α. $X = \overline{A} + \overline{B} + \overline{C}$ Β. $X = \overline{(A + B + C)}$
 Γ. $X = A + \overline{B} + \overline{CC}$ Δ. $X = A(B + C)$
11. Συστηματική μείωση των λογικών κυκλωμάτων επιτυγχάνεται με:
 Α. Χρήση άλγεβρας Boolean Β. Συμβολική μείωση
 Γ. Λογική TTL Δ. Χρήση πινάκων αληθείας
12. Πόσες πύλες απαιτούνται για να υλοποιηθεί η παρακάτω Boolean έκφραση, πριν την απλοποίηση;
 $XY + X(X + Z) + Y(X + Z)$
 Α. 1 Β. 2 Γ. 4 Δ. 5
13. Η έκφραση $AC + ABC = AC$ είναι
 Α. Σωστή Β. Λάθος
14. Ο χάρτης Karnaugh είναι ένας συστηματικός τρόπος απλοποίησης της έκφρασης:
 Α. POS Β. XNOR Γ. SOP Δ. Των συζυγών παραστάσεων
15. Ένα από τα θεωρήματα De Morgan εκφράζεται από τη σχέση $\overline{X + Y} = \overline{X} \overline{Y}$. Αυτό σημαίνει ότι δεν υπάρχει διαφορά μεταξύ:
 Α. Μιας NOR και μιας AND πύλης με ανεστραμμένες εισόδους
 Β. Μιας NAND και μιας OR πύλης με ανεστραμμένες εισόδους
 Γ. Μιας AND και μιας NOR πύλης με ανεστραμμένες εισόδους
 Δ. Μιας NOR και μιας NAND πύλης με ανεστραμμένες εισόδους
16. Ποιο από τα παρακάτω είναι σημαντικό χαρακτηριστικό της έκφρασης του SOP;
 Α. Όλα τα λογικά κυκλώματα είναι απλά AND και OR πύλες
 Β. Μειώνεται σημαντικά ο χρόνος καθυστέρησης (Delay)
 Γ. Δεν υπάρχει σήμα που να πρέπει να περάσει μέσα από περισσότερες από δύο πύλες, μη συμπεριλαμβάνοντας τους μετατροπείς
 Δ. Ο μέγιστος αριθμός πυλών που κάθε σήμα μπορεί να περάσει μέσα τους μειώνεται κατά έναν παράγοντα του δύο.
17. Για τον παρακάτω πίνακα αληθείας να καθοριστεί η έκφραση SOP

ΕΙΣΟΔΟΙ			ΕΞΟΔΟΣ
A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

- Α. $X = A\overline{B}C + ABC + \overline{A}B\overline{C}$ Β. $X = A\overline{B}C + ABC + \overline{A}B\overline{C}$
 Γ. $X = A\overline{B}C + ABC + \overline{A}B\overline{C}$ Δ. $X = A\overline{B}C + ABC + \overline{A}B\overline{C}$
18. Σε έναν πίνακα αληθείας για την έκφραση SOP $X = \overline{A}B\overline{C} + \overline{A}B\overline{C} + \overline{A}B\overline{C}$ πόσους συνδυασμούς εισόδων μπορούμε να έχουμε; $X = \overline{A}B\overline{C} + \overline{A}B\overline{C} + \overline{A}B\overline{C}$
 Α. 1 Β. 2 Γ. 4 Δ. 8
19. Για την έκφραση του SOP $X = \overline{A}B\overline{C} + \overline{A}B\overline{C} + \overline{A}B\overline{C}$ πόσοι άσσοι βρίσκονται στη στήλη της εξόδου του πίνακα αληθείας;
 Α. 1 Β. 2 Γ. 3 Δ. 5

20. Πριν την εφαρμογή αθροίσματος γινομένων (SOP), πόσες πύλες θα απαιτούσε η εξίσωση $X = AB(\overline{CD} + EF)$;
 A. 1 B. 2 Γ. 4 Δ. 5
21. Η έκφραση του SOP που προκύπτει από τον παρακάτω πίνακα αλήθειας είναι:

ΕΙΣΟΔΟΙ			ΕΞΟΔΟΣ
A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

A. $X = A\overline{B}C + ABC$
 Γ. $X = \overline{A}\overline{B}C + ABC$

B. $X = ABC + ABC$
 Δ. Καμία από τις ανωτέρω

ΑΣΚΗΣΕΙΣ

- Να δειχθεί με τη βοήθεια πινάκων αλήθειας ποια σχέση υπάρχει μεταξύ των λογικών συναρτήσεων: $T_1 = X \cdot \overline{Y} + \overline{X} \cdot Y$ και $T_2 = X \cdot Y + \overline{X} \cdot \overline{Y}$
 Στη συνέχεια να σχεδιαστούν τα λογικά κυκλώματα που αντιστοιχούν σ' αυτές.
- Να αποδειχθεί η παρακάτω λογική ισότητα $(A + B) \cdot (\overline{A} + C) = A \cdot C + \overline{A} \cdot B$:
 α) αλγεβρικά και β) με τη βοήθεια του πίνακα αλήθειας.
 Να σχεδιασθεί το λογικό κύκλωμα της συνάρτησης με διακόπτες.
- Να απλοποιήσετε αλγεβρικά την παρακάτω λογική συνάρτηση:
 $T = A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot C + A \cdot B \cdot C$
- Να απλοποιήσετε με χάρτη Karnaugh τις παρακάτω λογικές συναρτήσεις
 α) $T_1 = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot C \cdot D +$
 $+ \overline{A} \cdot B \cdot \overline{C} \cdot D + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot B \cdot C \cdot \overline{D}$
 β) $T_2 = \overline{A} \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot C$
- Να απλοποιήσετε, με χάρτη Karnaugh, τη συνάρτηση:
 $T = (A + B + C + D) \cdot (\overline{A} + B + C + \overline{D}) \cdot (A + B + C + \overline{D}) \cdot (A + B + \overline{C} + D) \cdot$
 $(A + \overline{B} + C + D) \cdot (\overline{A} + \overline{B} + C + \overline{D}) \cdot (A + \overline{B} + \overline{C} + \overline{D}) \cdot (\overline{A} + \overline{B} + \overline{C} + \overline{D})$

Σημείωση: Για τη λύση της άσκησης καλό είναι να υπολογιστεί πρώτα αναλυτικά η \overline{T} , να γίνει ο χάρτης Karnaugh, που αντιστοιχεί σ' αυτή, και, στη συνέχεια, για τον υπολογισμό της T, να ληφθούν υπόψη οι κενές θέσεις (μηδενικά της \overline{T}).
- Να γράψετε την έκφραση ελαχίστων όρων της συνάρτησης: $f(A, B, C) = A + BC$
- Να απλοποιήσετε την λογική συνάρτηση: α) αλγεβρικά και β) με τη βοήθεια του χάρτη Karnaugh. $T = \overline{A} + A \cdot \overline{B} + B \cdot D + B \cdot \overline{C} \cdot \overline{D} + B \cdot C \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot C \cdot D$
- Να εκφράσετε σε κανονική μορφή τις λογικές συναρτήσεις:
 $T(A, B, C, D, E) = ABDE + ABE + CE + BE$
 $T(A, B, C) = ABC + \overline{ABC} + BC$
- Δίνεται ο παρακάτω χάρτης Karnaugh:

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

Να βρεθεί η λογική συνάρτηση που παριστάνει ο χάρτης και να σχεδιαστεί το αντίστοιχο λογικό κύκλωμα με διακόπτες.

10. Για τους παρακάτω χάρτες Karnaugh να γραφούν οι αντίστοιχες λογικές συναρτήσεις.

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

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

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

11. Να απλοποιηθεί με τη μέθοδο Quine-Mc-Cluskey η λογική συνάρτηση:

$$f(A, B, C, D, E) = \sum (0, 1, 2, 6, 7, 8, 10, 13, 17, 18, 21, 22, 29, 30)$$

12. Να σχεδιάσετε το χάρτη Karnaugh των συναρτήσεων:

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

$$f(A, B, C, D) = \bar{A}\bar{B} + C(\bar{D} \oplus (A \oplus B)) + \bar{A}\bar{C}D$$

13. Να χρησιμοποιήσετε χάρτη Karnaugh και να βρείτε την ελάχιστη συνάρτηση αθροίσματος γινομένων καθώς και την ελάχιστη συνάρτηση γινομένου αθροισμάτων της συνάρτησης:

$$T_1(A, B, C, D) = \sum (0, 2, 4, 6, 7, 10, 13, 14, 15)$$

14. Να απλοποιήσετε με χάρτη Karnaugh τις λογικές συναρτήσεις:

$$T_1(A, B, C, D) = \sum (0, 2, 5, 6, 7, 8, 10) \text{ και } T_2(X, Y, Z, W) = \sum (5, 6, 9, 10)$$

15. Τι ονομάζουμε δυϊκότητα; Τι ονομάζουμε μεγιστοβάθμιους και τι ελαχιστοβάθμιους όρους;

16. Να εκφραστεί η λογική συνάρτηση $T(A, B, C, D) = \sum (0, 2, 5, 7, 10, 13)$ με τρεις διαφορετικούς τρόπους.

17. Να απλοποιήσετε τις παρακάτω λογικές συναρτήσεις με το χάρτη Karnaugh.

$$\alpha) T_1 = \bar{A} \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot C$$

$$\beta) T_2 = \bar{A} \cdot B \cdot \bar{C} \cdot \bar{D} + A \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot B \cdot C \cdot \bar{D} + A \cdot B \cdot C \cdot \bar{D}$$

$$\gamma) T_3 = \bar{A} \cdot B \cdot \bar{C} \cdot D + \bar{A} \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot \bar{C} \cdot D + A \cdot \bar{B} \cdot \bar{C} \cdot D + A \cdot \bar{B} \cdot C \cdot D$$

$$\delta) T_4 = \bar{A} \cdot B \cdot \bar{C} \cdot \bar{D} + A \cdot B \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot B \cdot \bar{C} \cdot D + A \cdot B \cdot \bar{C} \cdot D + \bar{A} \cdot B \cdot C \cdot D$$

$$+ A \cdot B \cdot C \cdot D + \bar{A} \cdot B \cdot C \cdot \bar{D} + A \cdot B \cdot C \cdot \bar{D}$$

Αναφορές-Βιβλιογραφία

- Κοσσίδης Α.Θ. (1996). *Σχεδίαση Ψηφιακών Κυκλωμάτων*, Εκδόσεις Μπένος
- Φραγκάκης Γ. (1975). *Λογικά Κυκλώματα*, Αθήνα
- Balabanian N., Carlson B. (2007). *Digital Logic Design Principles*, John Wiley
- Floyd Thomas L. (2013). *Digital Fundamentals: A systems Approach*, Pearson International Edition
- Givone D, (2002). *Digital Principles and Design*,. Mc Graw Hill
- Godse A.P., Godse D.A. (2011). *Digital Logic Design and Application*, 2nd, Technical Publications Pune
- Holdsworth Brian, Woods Clive (2002). *Digital Logic Design*, 4th Edition, Newnes
- Katz R., (2005). *Contemporary Logic Design*, 2/e, Prentice Hall
- Mano M., Ciletti M., (2014). *Ψηφιακή Σχεδίαση*, 5^η έκδοση, Παπασωτηρίου
- Maxfield Clive (2009). *Bebop to the Boolean Boogie, An Unconventional Guide to Electronics*, 3rd, Newnes, Elsevier (Online έκδοση: <http://english.360elib.com/datu/T/EM140371.pdf>)
- Nashelsky Louis (1994). *Introduction to Digital Technology*, 4th Ed., Prentice Hall
- Nelson V., Nagle H., Carroll B., Irwin J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall
- Predko Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill
- Pritchard N. (2015). *Fundamentals of Digital Electronics*, CreateSpace Independent Publishing Platform
- Roth Charles Jr., Kinney L.L., (2014). *Fundamentals of Logic Design*, 7th, Cengage Learning
- Tocci R. J., Widmer N. S, and Moss Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley
- Wakerly J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Κεφάλαιο 4

Σύνοψη

Αρχικά, στο κεφάλαιο αυτό, θα περιγραφούν οι βασικές λογικές πύλες AND, OR, NOT και θα δοθούν οι πίνακες αληθείας τους. Στη συνέχεια, ακολουθούν οι πύλες NAND και NOR καθώς και οι πύλες αποκλειστικής διάζευξης XOR και XNOR. Ο μετασχηματισμός των βασικών πυλών AND, OR, NOT με πύλες NAND, NOR είναι βασικός στο σχεδιασμό. Τέλος γίνεται μια περιγραφή των πυλών δύο επιπέδων τύπου AOI και OAI. Ακολουθεί η βασική ανάλυση και σχεδίαση κυκλωμάτων. Τα κυκλώματα των βασικών κυκλωμάτων αριθμητικών πράξεων της άλγεβρας (κύκλωμα ημιαθροιστή, κύκλωμα αθροιστή σειριακού και παράλληλου, κύκλωμα αφαιρέτη, κύκλωμα δυαδικού πολλαπλασιασμού). Το κεφάλαιο ολοκληρώνεται με το κύκλωμα του δυαδικού συγκριτή και το κύκλωμα μετατροπής κωδίκων με τη χρήση πυλών.

Προαπαιτούμενη γνώση

Βασικές γνώσεις της Άλγεβρας Boole και των θεωρημάτων της.

4. Λογικές Πύλες- Κυκλώματα βασικών πράξεων

4.1. Λογικές Πύλες

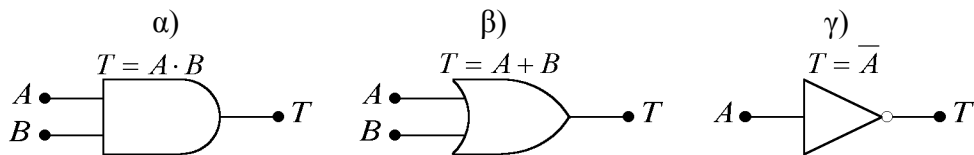
Τα θεμελιώδη λογικά κυκλώματα (λογικές πύλες), με τη βοήθεια των οποίων είναι δυνατόν να κατασκευαστούν άλλα συνθετότερα, είναι τρία:

- α. Η πύλη λογικού γινομένου **AND** (ΚΑΙ). Σύμβολο πράξης « \cdot » (γινόμενο) ή τομή δύο συνόλων.
- β. Η πύλη λογικής πρόσθεσης **OR** (Η). Σύμβολο πράξης « $+$ » (πρόσθεση) ή ένωση δύο συνόλων.
- γ. Η πύλη λογικής άρνησης **NOT** (ΟΧΙ). Σύμβολο πράξης « \neg ».

Στον πίνακα 4.1 που ακολουθεί παρατίθενται όλοι οι γνωστοί τρόποι γραφικής απεικόνισης των λογικών πυλών. Η συνήθης πρακτική είναι η χρήση τους κατά IEC 60617-12. (Nelson et al., 1995)

	IEC 60617-12	ANSI/IEEE 91-1984 ANSI/IEEE Std 91a-1991	DIN 40700
BUFFER			
NOT			
OR			
NOR			
AND			
NAND			
XOR			
XNOR			

Πίνακας 4.1. Γραφικά σύμβολα πυλών λογικών συναρτήσεων



Σχήμα 4.1. Συμβολισμός και συναρτήσεις των βασικών πυλών

Στο σχήμα 4.1 σημειώνονται τα σύμβολα των τριών παραπάνω πυλών, μαζί με τα ονόματα των λογικών μεταβλητών εισόδου και το αντίστοιχο λογικό αποτέλεσμα στην έξοδο T. (σχήματα.4.1 α, β, γ). (Nashelsky, 1994; Κοσσίδης, 1996; Holdsworth, & Woods, 2002; Givone, 2002; Balch, 2003; Katz, 2005; Κοσσίδης, & Γιαννακόπουλος, 2006; Wakerly, 2006; Balabanian, & Carlson, 2007; Maxfield, 2009; Floyd, 2013; Mano, & Ciletti, 2014; Roth, & Kinney, 2014; Pritchard, 2015)

Για το συμβολισμό των βασικών πυλών του σχήματος 4.1 χρησιμοποιήθηκαν πύλες δύο εισόδων, αριθμός που μπορεί να αυξηθεί, ώστε να ανταποκρίνεται στις ανάγκες των εκάστοτε εφαρμογών.

Στον πίνακα 4.2 δίνονται οι πίνακες αλήθειας των τριών βασικών λογικών πυλών για όλους τους δυνατούς συνδυασμούς των μεταβλητών εισόδου.

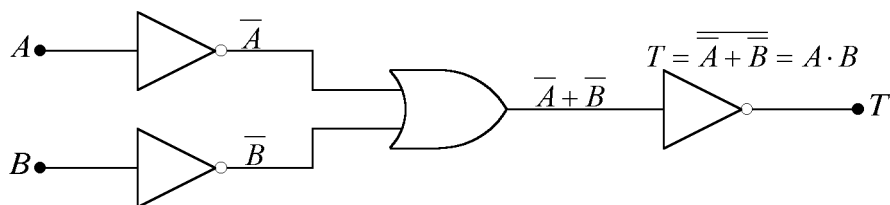
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	A · B
0	0	0
0	1	0
1	0	0
1	1	1

A	A-bar
0	1
1	0

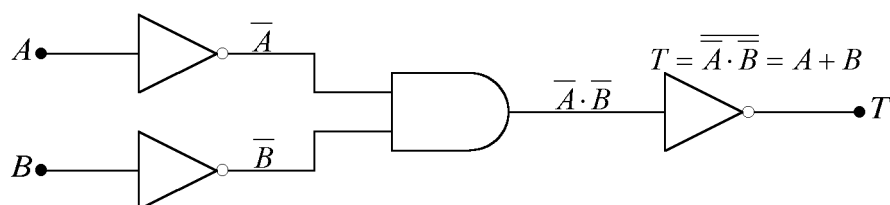
Πίνακας 4.2. Πίνακες αλήθειας των πυλών OR, AND και NOT

Στη πράξη όλα τα λογικά κυκλώματα μπορούν να κατασκευαστούν με τη βοήθεια δύο ζευγών πυλών: NOT - OR ή NOT - AND. Το παραπάνω εύκολα αποδεικνύεται, χρησιμοποιώντας το θεώρημα του De Morgan. Έτσι το κύκλωμα λογικού γινομένου $A \cdot B$ μπορεί να αντικατασταθεί από το κύκλωμα του σχήματος 4.2, που περιλαμβάνει μόνο πύλες OR και NOT.



Σχήμα 4.2. Πύλη AND με τη χρήση των πυλών NOT και OR

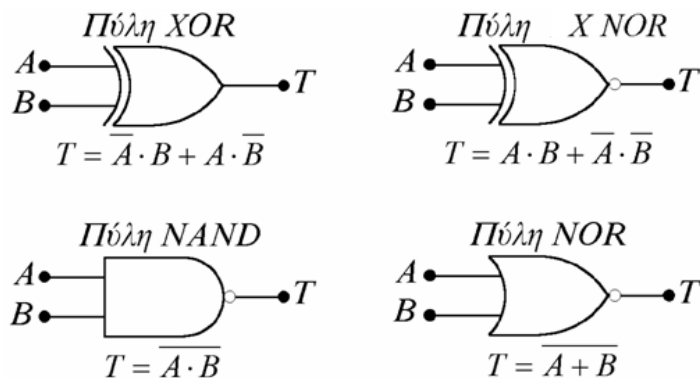
Αντίστοιχα το κύκλωμα λογικού αθροίσματος OR μπορεί να αντικατασταθεί από πύλες AND και NOT, όπως φαίνεται στο σχήμα 4.3.



Σχήμα 4.3. Πύλη OR με τη χρήση των πυλών NOT και AND

Εκτός από τις παραπάνω λογικές πύλες, υπάρχουν και άλλες εξίσου σημαντικές, με τις οποίες επιτυγχάνουμε απλούστευση των υπό μελέτη και κατασκευή λογικών κυκλωμάτων. Οι πύλες αυτές είναι οι:

- α. Η πύλη **NAND** (OXI-KAI). Σύμβολο πράξης « $\bar{\cdot}$ ».
- β. Η πύλη **NOR** (OXI-H). Σύμβολο πράξης « $\bar{\vee}$ ».
- γ. Η πύλη **XOR** ή Αποκλειστικού Ή (Exclusive-OR ή modulo-2). Σύμβολο πράξης « \oplus ».
- δ. Η πύλη **XNOR** ή Σύμπτωσης ή Αποκλειστικού NOR (Coincidence). Σύμβολο πράξης « \odot ».



Σχήμα 4.4. Συμβολισμός και συναρτήσεις των πυλών NAND, NOR, XOR, XNOR

Στο σχήμα 4.4 δίνονται τα αντίστοιχα σύμβολα των πυλών. Οι αντίστοιχοι πίνακες αλήθειας των λογικών αυτών πυλών δίνονται στον πίνακα 4.3.

A	B	T
0	0	1
0	1	0
1	0	0
1	1	1

A	B	T
0	0	0
0	1	1
1	0	1
1	1	0

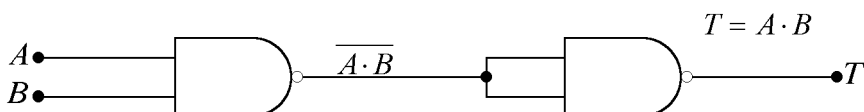
A	B	T
0	0	1
0	1	0
1	0	0
1	1	0

A	B	T
0	0	1
0	1	1
1	0	1
1	1	0

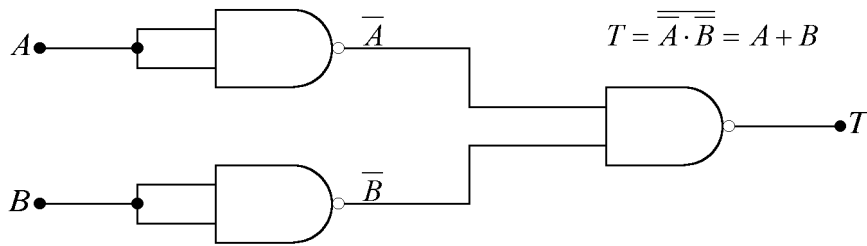
Πίνακας 4.3. Πίνακες αλήθειας των πυλών XOR, XNOR, NOR, NAND

4.2 Μετασχηματισμός των πυλών AND, OR, NOT με πύλες NAND, NOR

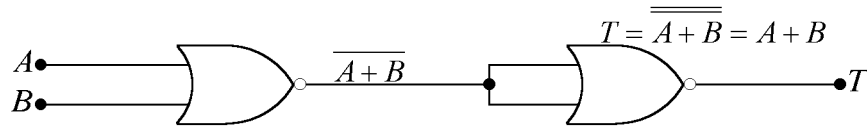
Ο μετασχηματισμός ενός κυκλώματος που έχει μελετηθεί με πύλες AND, OR, NOT σε ισοδύναμο κύκλωμα με πύλες NAND, NOR μπορεί να γίνει, χρησιμοποιώντας τις ισοδύναμες πύλες του σχήματος 4.5. Το κύκλωμα που θα προκύψει, θα πρέπει να απλοποιηθεί αμέσως μετά.



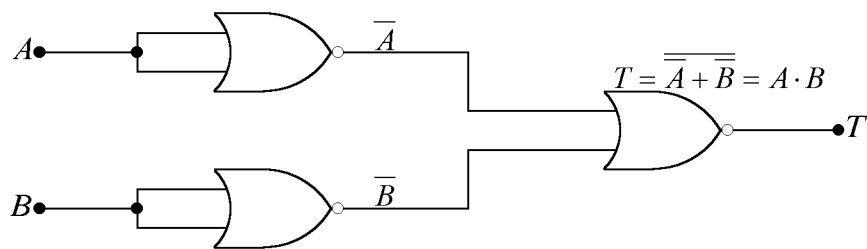
Σχήμα 4.5α. Υλοποίηση της πύλης AND με πύλες NAND.



Σχήμα 4.5β. Υλοποίηση της πύλης OR με πύλες NAND



Σχήμα 4.5γ. Υλοποίηση της πύλης OR με πύλες NOR



Σχήμα 4.5δ. Υλοποίηση της πύλης AND με πύλες NOR

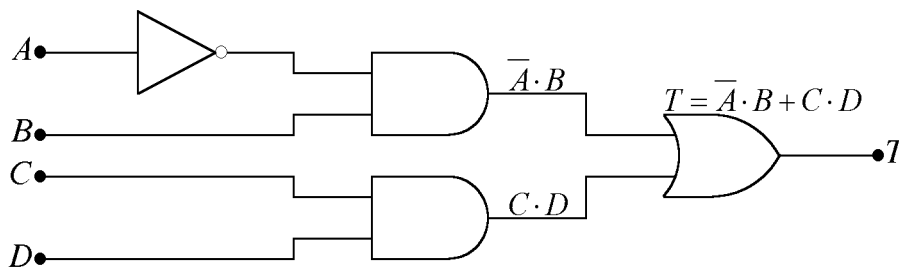
Στα σχήματα 4.5 α,β,γ,δ δίνουμε μερικά απλά κυκλώματα υλοποίησης των βασικών πυλών AND, OR, NOT με πύλες NAND ή πύλες NOR. (Στο σχήμα 4.5α το δεύτερο τμήμα του κυκλώματος είναι μια πύλη NOT). Ένας άλλος τρόπος απεικόνισης του τελικού σχήματος είναι να απλοποιήσουμε πρώτα τη συνάρτηση χρησιμοποιώντας θεωρήματα και τέλος να σχεδιάσουμε το κύκλωμα με τις πύλες NAND ή NOR:

Παράδειγμα 1ο:

Έστω ότι πρέπει να κατασκευάσουμε το λογικό κύκλωμα της συνάρτησης που ακολουθεί με πύλες NAND ή πύλες NOR.

$$T = \bar{A} \cdot B + C \cdot D \tag{4-1}$$

Το κύκλωμα με πύλες τις βασικές πύλες NOT, AND, OR έχει την παρακάτω μορφή:

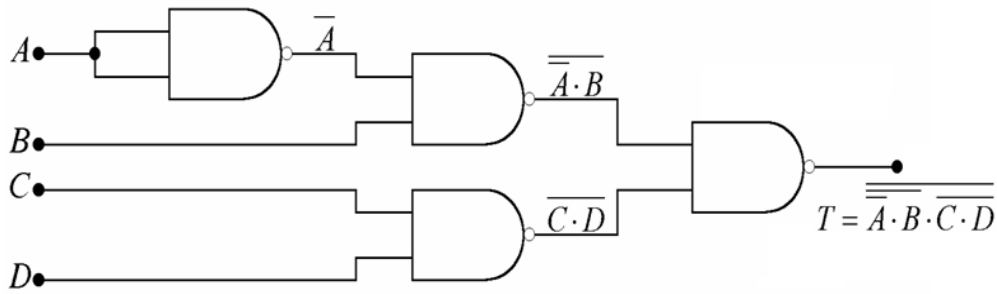


Σχήμα 4.6α. Υλοποίηση της συνάρτησης με κλασσικές πύλες

Η συνάρτηση T, εφαρμόζοντας το θεώρημα του De Morgan, γράφεται:

$$T = \overline{\overline{\bar{A} \cdot B + C \cdot D}} = \overline{\overline{\bar{A} \cdot B} \cdot \overline{C \cdot D}} \tag{4-2}$$

από όπου προκύπτει το παρακάτω κύκλωμα το οποίο είναι κατασκευασμένο εξ' ολοκλήρου με πύλες NAND.

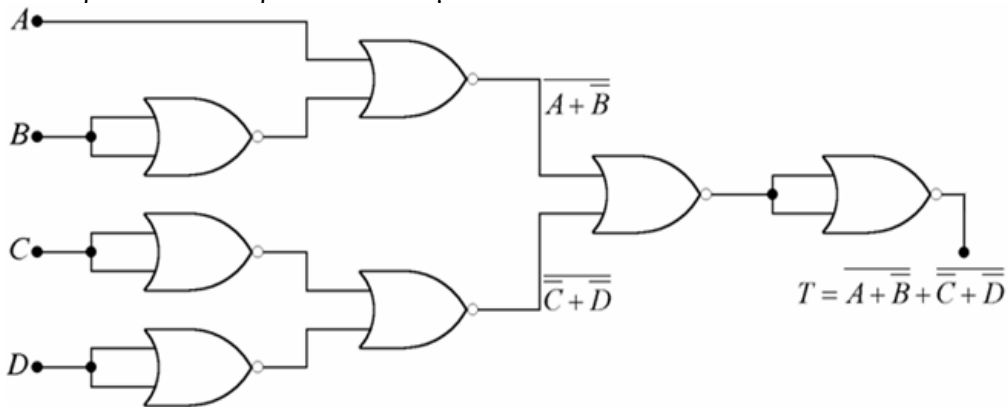


Σχήμα 4.6β. Υλοποίηση της συνάρτησης με πύλες NAND

Αντίστοιχα για το παραπάνω κύκλωμα με πύλες NOR έχουμε:

$$T = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}} = \overline{\overline{A} \cdot \overline{B} + \overline{C} \cdot \overline{D}} = \overline{\overline{A+B} + \overline{C+D}} \quad (4-3)$$

απ' όπου προκύπτει το παρακάτω κύκλωμα:



Σχήμα 4.6γ. Υλοποίηση της συνάρτησης με πύλες NOR

4.3. Πύλη αποκλειστικής διάζευξης (XOR)

Μια πύλη αποκλειστικής διάζευξης (Exclusive OR-XOR) δύο εισόδων έχει έξοδο αληθή (1), όταν μόνο μια από τις δύο εισόδους είναι αληθής. Η XOR είναι ιδιαίτερα χρήσιμη στο σχεδιασμό αθροιστών και στα κυκλώματα ανίχνευσης / διόρθωσης σφαλμάτων.

Μερικές ενδιαφέρουσες ιδιότητες της XOR είναι οι:

$$\begin{aligned} x \oplus 0 &= x & x \oplus 1 &= x' \\ x \oplus x &= 0 & x \oplus x' &= 1 \\ x \oplus (y \oplus z) &= (x \oplus y) \oplus z & & \text{[Προσεταιριστική]} \\ x \oplus y &= y \oplus x & & \text{[Επιμεριστική]} \end{aligned}$$

Η γενική συνάρτηση της XOR είναι αληθής, όταν έχουμε περιττό αριθμό ορισμάτων της αληθής (1) (δηλαδή περιττό αριθμό άσων).

Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε την άλγεβρα Boole, για να απλοποιήσουμε μια πύλη XOR τριών εισόδων στην παρακάτω έκφραση, με δεδομένο τον πίνακα αληθείας της XOR. (Πίνακας 4.4)

$$\begin{aligned} x \oplus (y \oplus z) &= x \oplus (y'z + yz') & & \text{[Ορισμός της XOR]} \\ &= x'(y'z + yz') + x(y'z + yz')' & & \text{[Ορισμός της XOR]} \\ &= x'y'z + x'yz' + x(y'z + yz')' & & \text{[Επιμεριστική]} \\ &= x'y'z + x'yz' + x((y'z)' (yz')') & & \text{[Θεώρημα De Morgan]} \\ &= x'y'z + x'yz' + x((y + z')(y' + z)) & & \text{[Θεώρημα De Morgan]} \end{aligned}$$

$$\begin{aligned}
 &= x'y'z + x'yz' + x(yz + y'z') && \text{[Επιμεριστική ιδιότητα]} \\
 &= x'y'z + x'yz' + xyz + xy'z' && \text{[Επιμεριστική ιδιότητα]}
 \end{aligned}$$

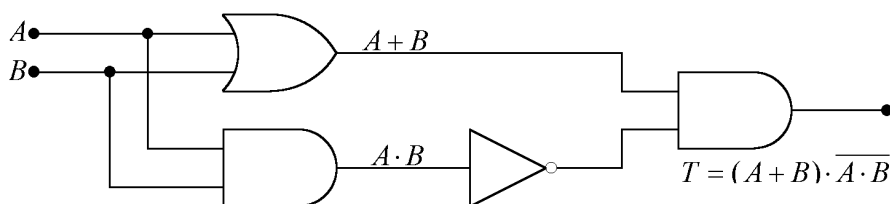
x	y	z	$x \oplus y \oplus z$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Πίνακας 4.4. Γενική συνάρτηση της πύλης XOR

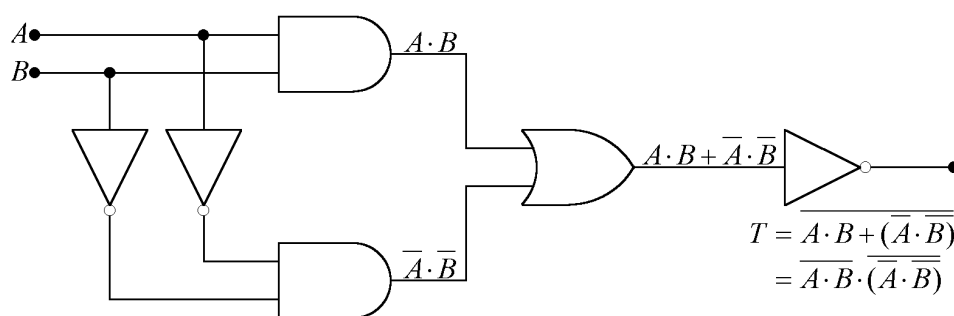
(Υπενθυμίζουμε ότι η αναπαράσταση της άρνησης γίνεται στο παρόν κεφάλαιο είτε με τονούμενη είτε με υπεργραμμισμένη μεταβλητή).

Με βάση τον πίνακα 4.4 της πύλης αποκλειστικής διάζευξης, XOR, η πύλη θα μπορούσε να κατασκευασθεί με διάφορους τρόπους με τη βοήθεια των γνωστών βασικών πυλών, που έχουμε ήδη γνωρίσει. Στο σχήμα 4.6α βλέπουμε ένα τέτοιο κύκλωμα όπου:

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



Σχήμα 4.7α. Πύλη XOR



Σχήμα 4.7β. Ισοδύναμη υλοποίηση πύλης XOR

Στο κύκλωμα του σχήματος 4.6β αποδίδεται το κύκλωμα της πύλης XOR, κατασκευασμένο, όμως, από διαφορετικές πύλες, όπως προκύπτει από μια ακόμα επεξεργασία της λογικής συνάρτησης που περιγράφει τις πύλες αυτού του είδους. Η σχεδιάσή του είναι απλή και εύκολο να κατανοηθεί, αν παρατηρήσουμε τις λογικές ποσότητες που εμφανίζονται στις εξόδους των πυλών και λάβουμε υπόψη μας τα θεωρήματα της άλγεβρας Boole.

Στη συνέχεια αποδεικνύουμε, για εκπαιδευτικούς λόγους, ότι η συνάρτηση $T = \overline{\bar{A} \cdot B \cdot \bar{A} \cdot B}$ περιγράφει πράξη XOR.

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

$$T = \bar{A} \cdot A + \bar{A} \cdot B + A \cdot \bar{B} + B \cdot B \text{ και επειδή } \bar{A} \cdot A = B \cdot B = 0 \text{ έχουμε τελικά } T = A \cdot \bar{B} + \bar{A} \cdot B$$

Αποδείξαμε δηλαδή ότι: $T = A \oplus B = \overline{A \cdot B} + \overline{\overline{A} \cdot \overline{B}} = \overline{A \cdot B} + \overline{\overline{A} \cdot \overline{B}}$

Με ανάλογο τρόπο μπορεί να αποδειχθεί ότι: $\overline{A \cdot B} + \overline{\overline{A} \cdot \overline{B}} = A \oplus B$

4.3.1. Πύλη XNOR

Η πύλη XNOR ή Σύμπτωσης ή Αποκλειστικού NOR (Coincidence), είναι η δυική πράξη της XOR. Η πύλη αναφέρεται και ως συγκριτής, διότι η έξοδός της είναι «1» όταν και δύο είσοδοι έχουν την ίδια τιμή. Ανάλογα, όταν οι είσοδοι έχουν διαφορετικές τιμές η έξοδος της πύλης είναι «0». Η πράξη απεικονίζεται με το σύμβολο «⊕».

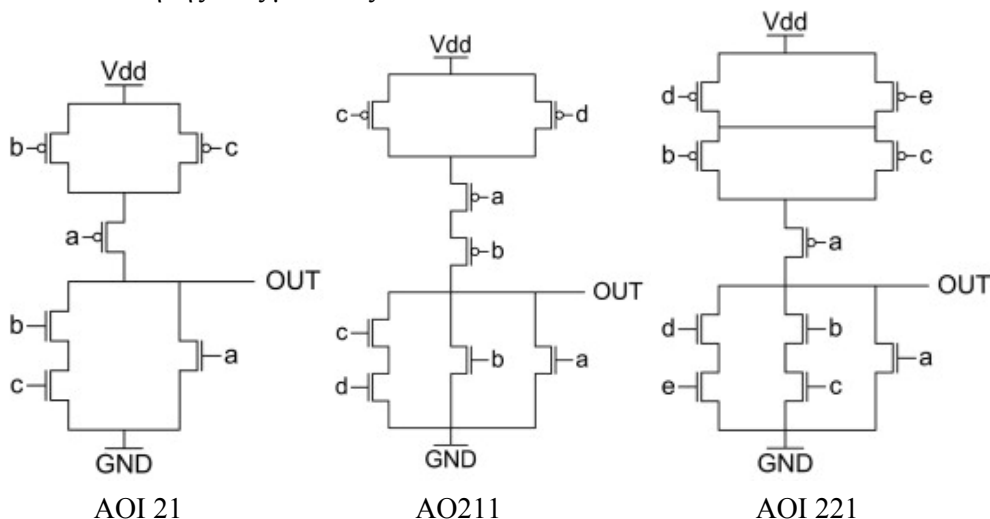
4.4. Σύνθετες πύλες δύο επιπέδων (AOI και OAI)

Με δεδομένη η ιεραρχία των λογικών πράξεων που είναι: NOT, AND, OR κρίνεται σκόπιμο να αναφέρουμε και την ύπαρξη σύνθετων πυλών δύο επιπέδων όπως, οι πύλες AOI(AND-OR-Invert) ή οι πύλες OAI(OR-AND-Invert) οι οποίες υλοποιούν όλες τις εκφράσεις SOP ή POS, που έχουν ήδη αναφερθεί. (Katz, 2005; Wakerly, 2006)

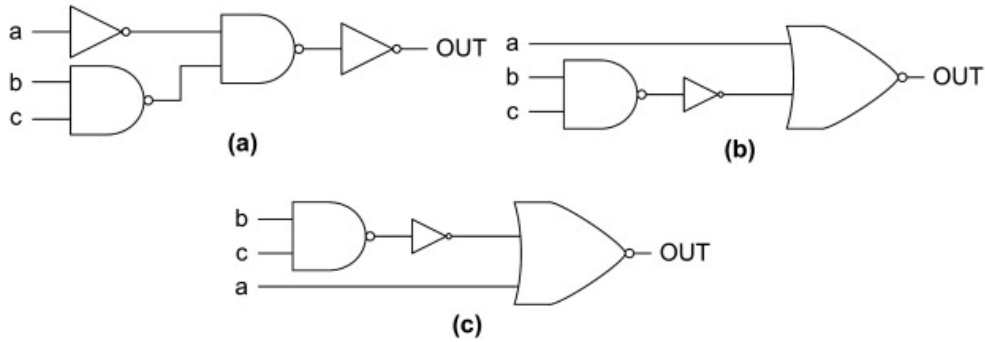
Οι δύο αυτοί τύποι πυλών είναι συμπληρωματικοί μεταξύ τους και κατασκευάζονται οι πύλες AOI από πύλες AND, που ακολουθούνται από πύλες NOR, και οι πύλες OAI από πύλες OR, που ακολουθούνται από πύλες NAND. Τα νούμερα που ακολουθούν τη σύνθετη πύλη αντιστοιχούν στον αριθμό των πυλών του πρώτου επιπέδου και στον αριθμό των εισόδων, που η καθεμία έχει. Συνήθως απεικονίζονται με τη μορφή π.χ. AND-OR-Invert-2-1 ή AOI-21 που αποδίδεται από την έκφραση: $X = \overline{(A \cdot B)} + C$. Αντίστοιχα η πύλη AOI-33 (έχει στο πρώτο επίπεδο δύο πύλες AND που η καθεμία έχει τρεις εισόδους) αντιστοιχεί στην έκφραση $X = \overline{(A \cdot B \cdot C)} + \overline{(D \cdot E \cdot F)}$. Για τις πύλες OAI αντίστοιχα παραδείγματα είναι τα π.χ. OAI και μια πύλη OAI222 έχει τρεις πύλες OR με την κάθε μία να έχει δύο εισόδους. (Η αρίθμηση γίνεται πάντα με φθίνουσα τάξη, καθώς είναι ισοδύναμη η απεικόνιση OAI222 ή OAI221).

Οι σύνθετες πύλες συχνά χρησιμοποιούνται κατά το σχεδιασμό των VLSI chip με CMOS, επειδή μπορεί να υλοποιηθούν αποτελεσματικά, βασισμένες σε έξυπνους συνδυασμούς σε σειρά και παράλληλα συνδεδεμένων τρανζίστορ. Για παράδειγμα, η πύλη AOI33 μπορεί να κατασκευαστεί, χρησιμοποιώντας μόνο 12 τρανζίστορ, ενώ η αντίστοιχη διακριτή υλοποίηση θα απαιτήσει 4 τρανζίστορ για την πύλη NOR, και 8 τρανζίστορ για κάθε πύλη AND τριών εισόδων, απαιτώντας έτσι ένα σύνολο από 20 τρανζίστορ – υλοποίηση σχεδόν δύο φορές πιο ακριβή και δύο φορές πιο αργή. (Hendrich 2006)

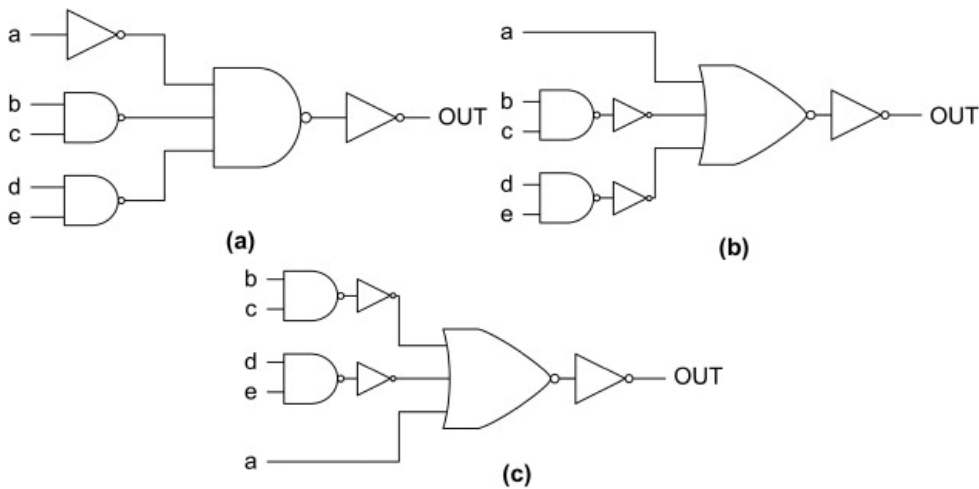
Στο διάγραμμα που ακολουθεί αποδίδονται ενδεικτικά τα δομικά διαγράμματα των AOI 21, AOI 211, AOI 221 με CMOS. (Butzen, Dal Bem, Reis & Ribas 2010). Ακολουθούν οι γραφικές απεικονίσεις των κυρίων διαγραμμάτων υλοποίησής τους με πύλες.



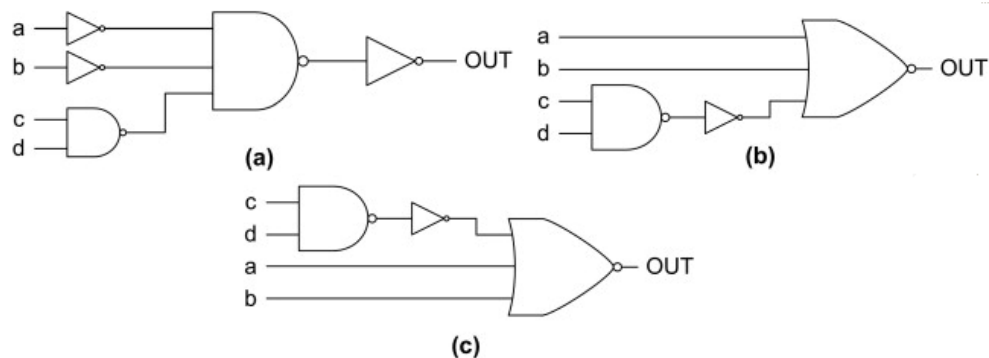
Σχήμα 4.8α. Απεικονίσεις με CMOS των πυλών AOI21, AOI211, AOI221



Σχήμα 4.8β. Διαγράμματα πλόν της AOI-21



Σχήμα 4.8γ. Διαγράμματα πλόν της AOI-221



Σχήμα 4.8δ. Διαγράμματα πλόν της AOI-211

4.5. Βασική Ανάλυση και Σχεδίαση Κυκλωμάτων

Μάθαμε ήδη ότι είναι απαραίτητο για να προχωρήσουμε στη σχεδίαση κυκλώματος:

- Οι πίνακες αληθείας και οι εκφράσεις της άλγεβρας Boole περιγράφουν συναρτήσεις.
- Οι εκφράσεις μπορούν να μετατραπούν σε κυκλώματα.
- Η άλγεβρα Boole και οι χάρτες Karnaugh μας βοηθούν να απλοποιήσουμε τις εκφράσεις και τα κυκλώματα.

Τώρα μπορούμε να χρησιμοποιήσουμε αυτές τις γνώσεις, για να αναλύσουμε και να σχεδιάσουμε μεγαλύτερα κυκλώματα.

Σχεδίαση κυκλώματος

Ο στόχος της σχεδίασης κυκλωμάτων είναι η κατασκευή υλικού το οποίο να υλοποιεί κάποια συγκεκριμένη συνάρτηση. Η βασική ιδέα είναι να γράψουμε τη συνάρτηση σαν μία Boolean έκφραση και κατόπιν να τη μετατρέψουμε σε ένα κύκλωμα. Τα βήματα, που ακολουθούμε κατά την υλοποίηση, είναι:

- Βήμα 1:** Προσδιορισμός του αριθμού των εισόδων και των εξόδων.
- Βήμα 2:** Πρέπει να είμαστε βέβαιοι ότι έχουμε τη συνάρτηση περιγραφής, είτε με τη μορφή του πίνακα αλήθειας είτε σαν μία Boolean έκφραση.
- Βήμα 3:** Μετατροπή αυτής σε μία απλοποιημένη Boolean έκφραση. (Βασικός στόχος είναι η εύρεση του ΕΑΓ, εκτός αν ζητείται κάτι διαφορετικό).
- Βήμα 4:** Κατασκευή του κυκλώματος, βασισμένη στην απλοποιημένη έκφραση.

Ανάλυση Κυκλωμάτων

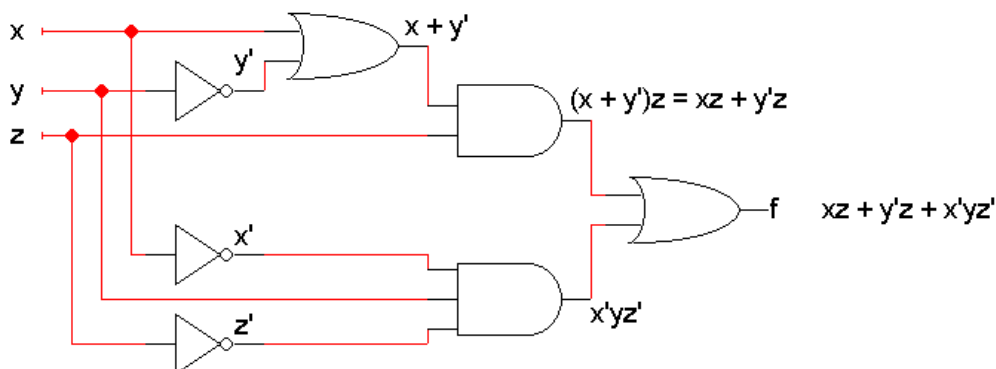
Η ανάλυση κυκλωμάτων περιλαμβάνει την κατανόηση τις λειτουργίας κάποιων κυκλωμάτων. Κάθε κύκλωμα υπολογίζει μία συνάρτηση, η οποία μπορεί να περιγραφεί με «Boolean» εκφράσεις ή με πίνακες αληθείας.

Έτσι ο στόχος μας είναι να βρούμε μια έκφραση ή ένα πίνακα αληθείας για το κύκλωμα.

Αρχικά πρέπει να προσδιορίσουμε όλες τις εισόδους και τις εξόδους του κυκλώματος. Το κύκλωμα της συνάρτησης $f(x, y, z) = xz + \bar{y}z + \bar{x}y\bar{z}$ που ακολουθεί (σχήμα 4.9α) έχει, όπως εύκολα διαπιστώνουμε, τρεις εισόδους τις x, y, z και μία έξοδο την f .

Στη συνέχεια, γράφουμε εκφράσεις για τις εξόδους κάθε πύλης, οι οποίες βασίζονται στις εισόδους των πυλών. Ξεκινάμε από τις εισόδους και εργαζόμαστε κατευθυνόμενοι προς τις εξόδους. Αν είναι δυνατόν κάνουμε και τυχόν απλοποιήσεις που θα μας βοηθήσουν στην περαιτέρω πορεία μας.

Στο διάγραμμα του σχήματος 4.9α παραθέτουμε, επίσης, τις συναρτήσεις (αλγεβρικές εκφράσεις) της καθεμιάς πύλης ξεχωριστά καθώς και της τελικής εξόδου.



Σχήμα 4.9α. Υλοποίηση της συνάρτησης $f(x, y, z) = xz + \bar{y}z + \bar{x}y\bar{z}$

Πίνακας αλήθειας

x	y	z	xz	$\bar{y}z$	$\bar{x}y\bar{z}$	f
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	0	1	1
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	1	1	0	1
1	1	0	0	0	0	0
1	1	1	1	0	0	1

Πίνακας 4.5. Πίνακας αλήθειας της συνάρτησης $f(x, y, z) = xz + \bar{y}z + \bar{x}y\bar{z}$

Τον πίνακα αληθείας τον φτιάχνουμε απευθείας από το κύκλωμα. Όταν γνωρίζουμε τον αριθμό των εισόδων και το εξόδων, καταγράφουμε όλους τους πιθανούς συνδυασμούς εισόδων στον πίνακα αληθείας.

Ένα κύκλωμα με n εισόδους πρέπει να έχει ένα πίνακα αληθείας με 2^n σειρές. Το παράδειγμά μας έχει τρεις εισόδους. Άρα ο πίνακας αληθείας θα έχει $2^3 = 8$ σειρές.

Δίνοντας τις τιμές στις μεταβλητές x, y, z και δημιουργώντας τους όρους της συνάρτησης, παίρνουμε τον πίνακα 4.5.

Εκφράσεις και πίνακες αλήθειας

Είδαμε ήδη ότι το κύκλωμα του σχήματος 4.9α υπολογίζει τη συνάρτηση $f(x, y, z) = xz + \bar{y}z + \bar{x}y\bar{z}$. Μπορούμε να χρησιμοποιήσουμε τη συνάρτηση αυτή, για να συμπληρώσουμε τον πίνακα αλήθειας.

Δημιουργούμε ξεχωριστές στήλες για καθένα από τους όρους xz , $\bar{y}z$ και $\bar{x}y\bar{z}$. Τελικά παίρνουμε τη συνάρτηση f , κάνοντας την πράξη OR μεταξύ των ενδιαμέσων στηλών. (Πίνακας 4.5).

Μπορεί να γίνει και η αντίθετη μετατροπή. Είναι εύκολο να φτιάξουμε μια αλγεβρική έκφραση, αν ήδη έχουμε τον πίνακα αληθείας.

Είδαμε πόσο γρήγορα μετατρέψαμε έναν πίνακα αληθείας σε μία έκφραση αθροίσματος ΕΛΒΟ. Οι ΕΛΒΟ αντιστοιχούν στις γραμμές του πίνακα που έχουν την τιμή 1.

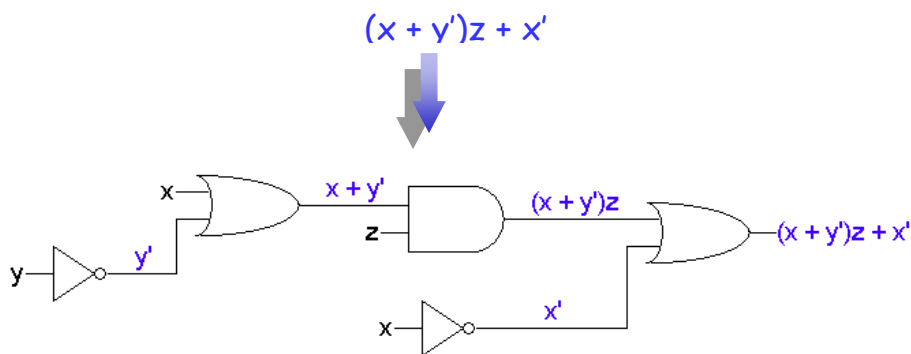
x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Η συνάρτηση είναι η: $f(x,y,z) = x'y'z + x'yz' + xy'z + xyz = m_1 + m_2 + m_5 + m_7$

Αν θέλουμε, μπορούμε να απλοποιήσουμε αυτό το άθροισμα των ΕΛΒΟ, χρησιμοποιώντας π.χ. ένα χάρτη Karnaugh. (Ποιο θα είναι το αποτέλεσμα;)

Παράδειγμα υλοποίησης απλών κυκλωμάτων με τη χρήση πυλών

Οποιαδήποτε Boolean έκφραση μπορεί να μετατραπεί σε κύκλωμα, συνδυάζοντας τις βασικές πύλες με έναν σχετικά εύκολο τρόπο.



Σχήμα 4.9β. Υλοποίηση της συνάρτησης $f = (x + \bar{y})z + \bar{x}$

Έστω η συνάρτηση: $f = (x + \bar{y})z + \bar{x}$. Η προτεραιότητα είναι ρητά καθορισμένη σε ένα κύκλωμα. Πρέπει να είμαστε σίγουροι ότι το κύκλωμα εκτελεί τις λειτουργίες με τον σωστό τρόπο, δηλαδή με τη σωστή σειρά.

Στο σχήμα 4.9 δίνεται η υλοποίησή της με λογικές πύλες και απεικονίζονται οι εισοδοί και οι έξοδοι κάθε πύλης.

4.6. Δυαδικός ημιαθροιστής

Πρόσθεση δύο δυαδικών ψηφίων

Ο δυαδικός ημιαθροιστής (Half adder) είναι ένα συνδυαστικό λογικό κύκλωμα ικανό να εκτελεί την πράξη της πρόσθεσης δυαδικών πληροφοριών μήκους ενός δυαδικού ψηφίου (bit) η κάθε μία. Το αποτέλεσμα λαμβάνεται από δύο εξόδους, εκ των οποίων η πρώτη παρέχει το άθροισμα και η δεύτερη το τυχόν κρατούμενο. *Nashelsky, 1994; Κοσσίδης, 1996; Holdsworth, & Woods, 2002; Givone, 2002; Predko, 2005; Κοσσίδης, & Γιαννακόπουλος, 2006; Wakerly, 2006; Godse, & Godse, 2011; Roth, & Kinney, 2014)*

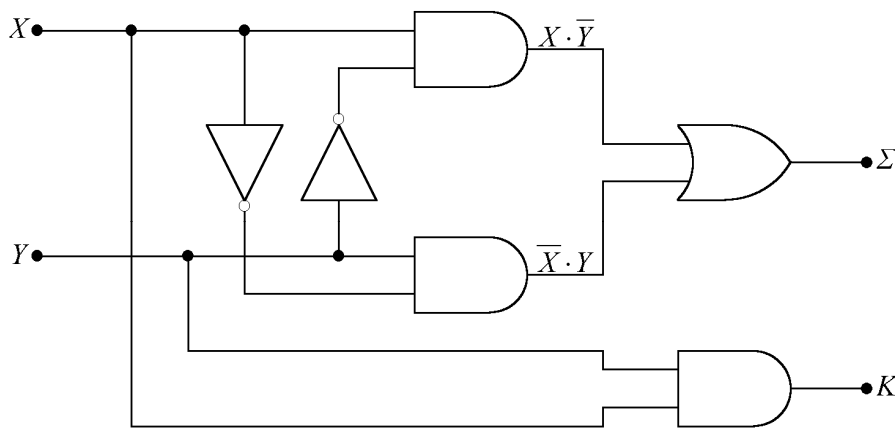
Από τον πίνακα πρόσθεσης δυαδικών ψηφίων υπολογίζουμε τις λογικές εξισώσεις που προσδιορίζουν τη λειτουργία του ημιαθροιστή:

$$\Sigma = \overline{X}Y + X\overline{Y} = X \oplus Y \quad (4-4) \quad \text{και} \quad K = X \cdot Y \quad (4-5)$$

Από αυτές προκύπτει το κύκλωμα του σχήματος 4.10α.

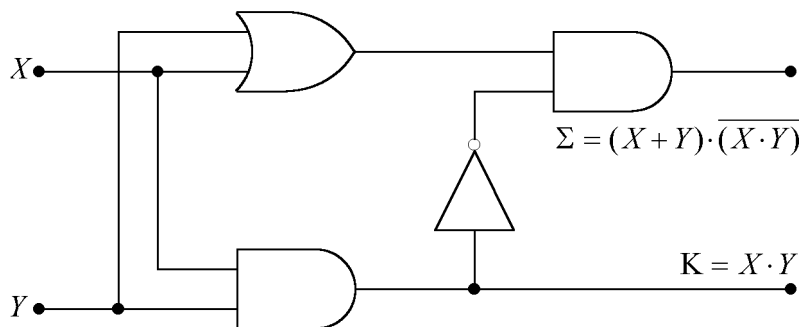
Το κύκλωμα του σχήματος 4.10α μπορεί να απλοποιηθεί, λαμβάνοντας υπόψη ότι το Σ έχει τιμή «1» μόνο όταν η μεταβλητή X ή η μεταβλητή Y είναι «1», και όχι και οι δύο ταυτόχρονα. Κατά συνέπεια, η παραπάνω σχέση που προσδιορίζει το άθροισμα Σ μπορεί να εκφραστεί και ως εξής:

$$\Sigma = (X + Y) \cdot \overline{X \cdot Y} \quad (4-6) \quad \text{και} \quad K = X \cdot Y \quad (4-7)$$

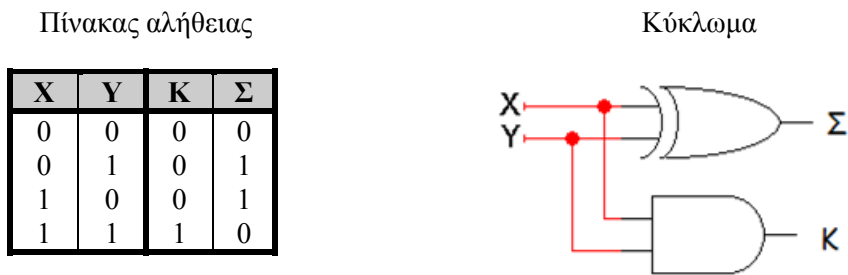


Σχήμα 4.10α. Αναλυτικό διάγραμμα του δυαδικού ημιαθροιστή

Κάνοντας πράξεις, προκύπτει ότι οι δύο σχέσεις 4-4 και 4-6 που προσδιορίζουν το Σ είναι ισοδύναμες. Το κύκλωμα που αντιστοιχεί σε αυτή την περίπτωση είναι το Σχήμα 4.11α, το οποίο με τη χρήση της πύλης της αποκλειστικής διάζευξης αντιστοιχεί στο κύκλωμα του σχήματος 4.11β που ακολουθεί.



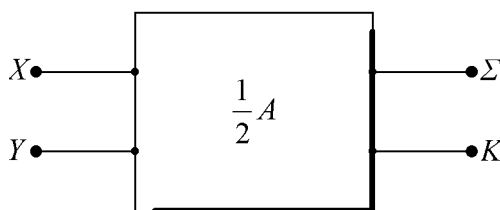
Σχήμα 4.11α. Δυαδικός ημιαθροιστής



Σχήμα 4.11β. Πίνακας αλήθειας και διάγραμμα του ημιαθροιστή

Στο ανωτέρω, Σχήμα 4.11β, δίνεται ο πίνακας αληθείας και το κύκλωμα του ημιαθροιστή, με τη χρήση της πύλης της αποκλειστικής διάζευξης (XOR).

Σχηματικά το δομικό διάγραμμα του ημιαθροιστή δίνεται στο σχήμα 4.12.



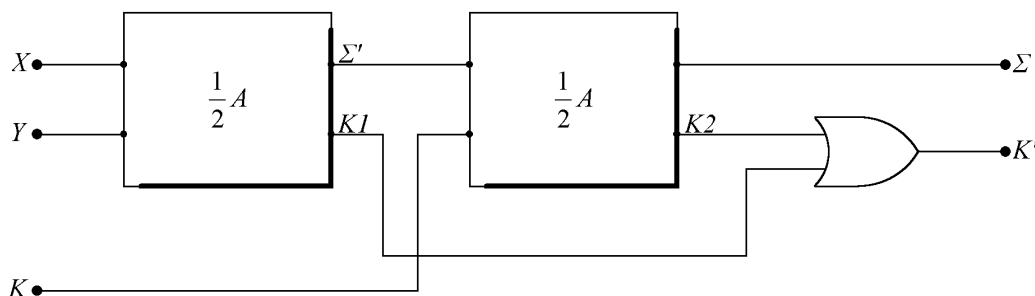
Σχήμα 4.12. Δομικό διάγραμμα ημιαθροιστή

4.7. Δυαδικός αθροιστής τριών εισόδων ή πλήρης αθροιστής (Full adder)

Ο ημιαθροιστής είναι σε θέση να προσθέσει δύο πληροφορίες του ενός bit η καθεμία, χωρίς να λαμβάνει υπόψη τυχόν κρατούμενο που μπορεί να έχει παραχθεί από ένα άλλο κύκλωμα πρόσθεσης. Με το δυαδικό αθροιστή τριών εισόδων, ή πλήρη αθροιστή, λαμβάνεται υπόψη και το πιθανό κρατούμενο. (Nashelsky, 1994; Holdsworth, & Woods, 2002; Givone, 2002; Predko, 2005; Κοσσίδης, & Γιαννακόπουλος, 2006; Godse, & Godse, 2011)

Η πρόσθεση των X, Y και K δίνει σαν αποτέλεσμα το άθροισμα Σ και το κρατούμενο K'. Η πρόσθεση αυτή εύκολα μπορεί να χωριστεί σε δύο επιμέρους προσθέσεις: στην πρώτη X+Y που δίνει άθροισμα Σ' και κρατούμενο K₁ και στη δεύτερη Σ'+K που δίνει άθροισμα Σ και κρατούμενο K₂. Πρέπει να ληφθεί υπόψη ότι, όταν το άθροισμα X+Y δίνει κρατούμενο K₁=1, το Σ' θα ισούται με 0, και, επομένως, το άθροισμα Σ'+K δεν μπορεί να δώσει κρατούμενο. Κατά συνέπεια, εάν τοποθετήσουμε δύο ημιαθροιστές (Σχήμα 4.12) σε σειρά, επιτυγχάνουμε το επιθυμητό κύκλωμα που φαίνεται στα Σχήματα 4.13α και 4.13β. Οι εξισώσεις που προκύπτουν είναι οι:

$$\begin{aligned} \Sigma' &= (X + Y) \cdot (\overline{X \cdot Y}) & K_1 &= X \cdot Y \\ \Sigma &= (\Sigma' + K) \cdot (\overline{\Sigma' \cdot K}) & K_2 &= \Sigma' \cdot K \end{aligned}$$

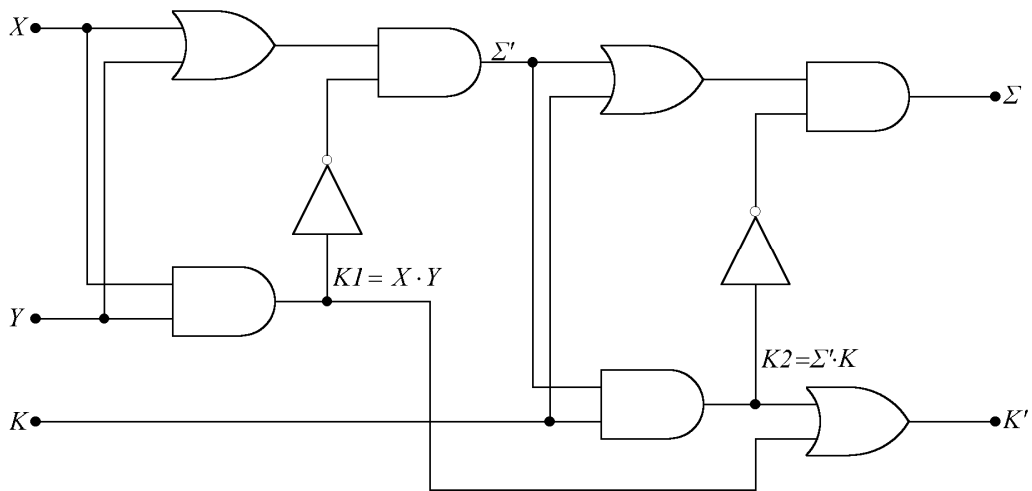


Σχήμα 4.13α. Πλήρης αθροιστής με χρήση του δομικού διαγράμματος του ημιαθροιστή

Αντικαθιστώντας, θα έχουμε:

$$\Sigma = [(X + Y) \cdot (\overline{X \cdot Y}) + K] \cdot \overline{[(X + Y) \cdot (\overline{X \cdot Y}) \cdot K]} \quad (4-8)$$

$$K' = K_1 + K_2 = X \cdot Y + (X + Y) \cdot (\overline{X \cdot Y}) \cdot K \quad (4-9)$$



Σχήμα 4.13β. Αναλυτικό διάγραμμα πλήρους αθροιστή

Πρόσθεση τριών δυαδικών ψηφίων

Με ανάλογο τρόπο μπορεί να μελετηθεί και ο αθροιστής τριών εισόδων. Τι πρέπει να κάνουμε, για να προσθέσουμε 3 bits; Απλά να αθροίσουμε τους δύο προσθετέους και το τυχόν κρατούμενο, που μεταφέρεται από τα δεξιά.

Ο πίνακας 4.6 συσχετίζει τις τρεις μεταβλητές εισόδου (**X**, **Y**, **K**) και δίνει σαν αποτέλεσμα δύο στήλες: μια για το άθροισμα (**Σ**) και μια για το κρατούμενο (**K'**).

Χρησιμοποιώντας την άλγεβρα Boole, παίρνουμε τις εξισώσεις που ακολουθούν. Η χρήση της πύλης XOR απλοποιεί περαιτέρω τη μορφή των εξισώσεων. (Χρησιμοποιούμε την άλγεβρα διότι δε μπορούμε εύκολα, να εξάγουμε τις εξισώσεις της XOR από τους χάρτες Karnaugh).

X	Y	K	Σ	K'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Πίνακας 4.6: Πίνακας αλήθειας πλήρους αθροιστή

$$\begin{aligned} \Sigma = \Sigma m(1,2,4,7) &= \overline{X} \overline{Y} K + \overline{X} Y \overline{K} + X \overline{Y} \overline{K} + XYK = \overline{X} (\overline{Y} K + Y \overline{K}) + X (\overline{Y} \overline{K} + YK) \\ &= \overline{X} (K \oplus Y) + X (\overline{Y \oplus K}) = X \oplus Y \oplus K \end{aligned} \quad (4-10)$$

και

$$\begin{aligned} K' = \Sigma m(3,5,6,7) &= \overline{X} Y K + X \overline{Y} K + X Y \overline{K} + XYK = (\overline{X} Y + X \overline{Y}) K + XY(\overline{K} + K) \\ &= (Y \oplus X) K + XY \end{aligned} \quad (4-11)$$

Συνεπώς οι εξισώσεις του πλήρους αθροιστή δίνονται από τις σχέσεις:

$$\Sigma = X \oplus Y \oplus K \text{ και } K' = (X \oplus Y) K + XY$$

Εναλλακτικά από τον πίνακα αλήθειας του αθροιστή προκύπτουν οι παρακάτω εξισώσεις:

$$\Sigma = \bar{X} \cdot \bar{Y} \cdot K + \bar{X} \cdot Y \cdot \bar{K} + X \cdot \bar{Y} \cdot \bar{K} + X \cdot Y \cdot K$$

$$K' = \bar{X} \cdot Y \cdot K + X \cdot \bar{Y} \cdot K + X \cdot Y \cdot \bar{K} + X \cdot Y \cdot K$$

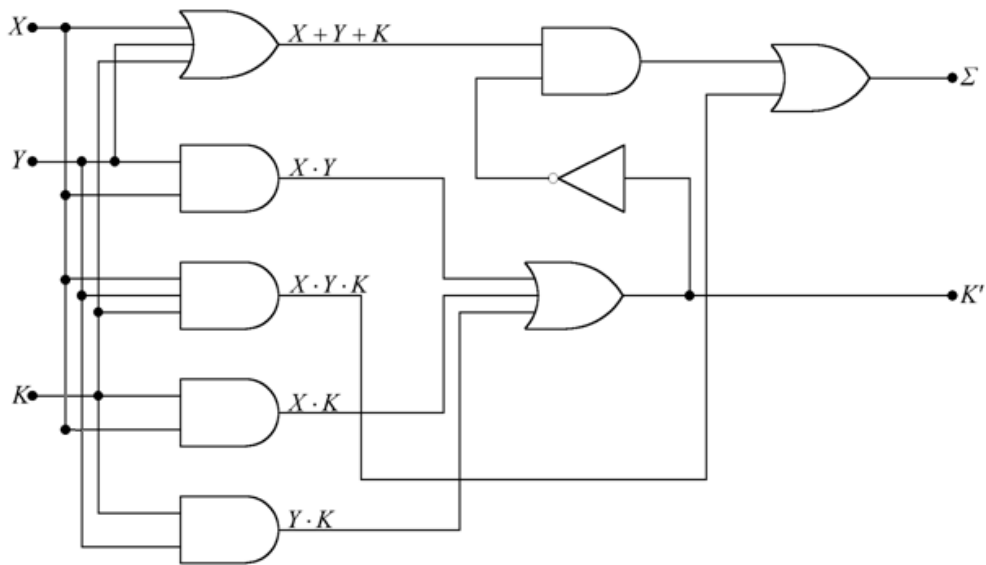
Η απλοποίηση των παραπάνω σχέσεων μπορεί να γίνει με τους χάρτες Karnaugh. Εύκολα επιτυγχάνουμε μια άλλη έκφραση του Σ και K' , αν ληφθεί υπόψη ότι το Σ έχει τιμή «1», όταν, όπως ήδη αναφέραμε περιττός αριθμός εισόδων έχει τιμή «1». Δηλαδή έχουμε:

$$\Sigma = (X + Y + K) \cdot (X \cdot Y + X \cdot K + Y \cdot K) + X \cdot Y \cdot K \quad (4-12)$$

Αντίστοιχα το κρατούμενο έχει τιμή «1», όταν δύο από τις τρεις εισόδους είναι «1» ή και οι τρεις έχουν τιμή «1».

$$K' = X \cdot Y + X \cdot K + Y \cdot K \quad (4-13)$$

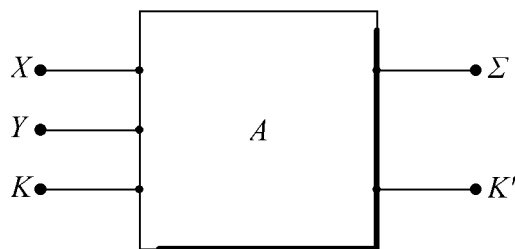
Σύμφωνα με τις παραπάνω σχέσεις προκύπτει το κύκλωμα του Σχήματος 4.14.



Σχήμα 4.14. Αναλυτικό διάγραμμα πλήρους αθροιστή με βασικές πύλες

(Να συγκρίνετε τις σχέσεις 4-10 και 4-12 καθώς και τις 4-11 και 4-13 και να σχεδιάσετε το διάγραμμα του πλήρους αθροιστή χρησιμοποιώντας την πύλη της αποκλειστικής διάζευξης).

Το δομικό διάγραμμα του πλήρους αθροιστή δίνεται στο Σχήμα 4.15.

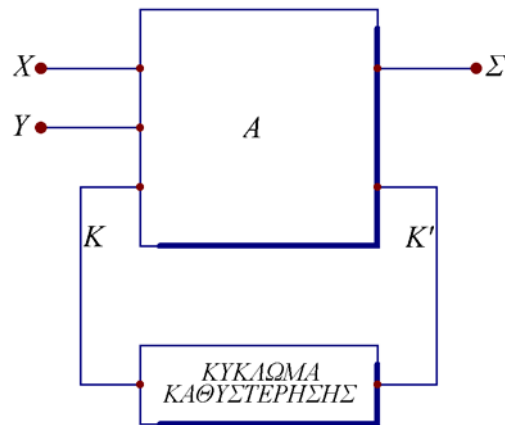


Σχήμα 4.15. Δομικό διάγραμμα πλήρους αθροιστή

4.8. Σειριακός πλήρης αθροιστής (Serial full adder)

Εάν έχουμε να αθροίσουμε δύο αριθμούς, που αποτελούνται από περισσότερα του ενός bit ο κάθε ένας, ο αθροιστής που αναφέραμε προηγουμένως (Σχήμα 4.15) είναι ικανός να εκτελέσει την πρόσθεση υπό την προϋπόθεση ότι τα δυαδικά ψηφία ακολουθούν το ένα το άλλο, ξεκινώντας από το ψηφίο μικρότερης τάξης και καταλήγοντας στο ψηφίο της μεγαλύτερης τάξης. Επίσης το κρατούμενο K_i που προκύπτει από την πρόσθεση των X_i και Y_i δυαδικών ψηφίων, επανεμφανίζεται στην είσοδο σαν K_{i+1} και συνυπολογίζεται κατά την πρόσθεση των ψηφίων X_{i+1} και Y_{i+1} των δυαδικών αριθμών.

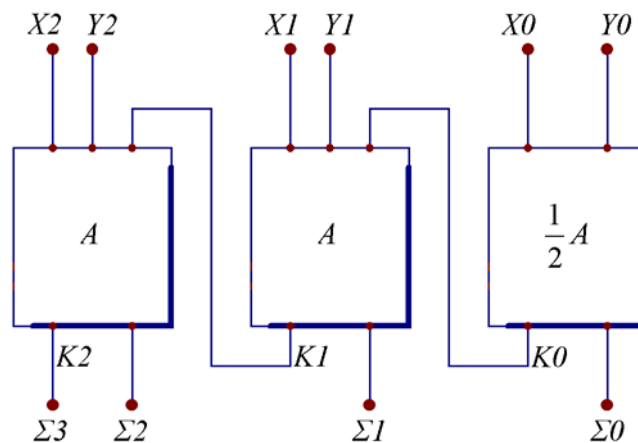
Αυτό επιτυγχάνεται συνδέοντας την έξοδο K' του αθροιστή στην είσοδο K του ίδιου αθροιστή δια μέσου ενός κυκλώματος καθυστέρησης (Delay f-f, $(D f-f)$), το οποίο συγκρατεί την πληροφορία για το απαιτούμενο χρονικό διάστημα που μεσολαβεί κατά την εφαρμογή δύο ψηφίων στην είσοδο του αθροιστή. Στο Σχήμα 4.16 δίνεται σχηματικά ο σειριακός πλήρης αθροιστής. Αποτελείται από δύο μόνον εισόδους X , Y , στις οποίες εισάγονται με σταθερή συχνότητα εξόδου τα δυαδικά ψηφία (bit) που προορίζονται να προστεθούν και από μία έξοδο Σ από την οποία εξέρχονται ένα - ένα τα ψηφία του αποτελέσματος. Τα ψηφία αυτά μπορούν στη συνέχεια να αποθηκευτούν σε έναν καταχωρητή / ολισθητή.



Σχήμα 4.16. Σειριακός δυαδικός αθροιστής

4.9. Παράλληλος πλήρης αθροιστής (Parallel full adder)

Ένας παράλληλος πλήρης αθροιστής επιτυγχάνεται, συνδέοντας κατάλληλα δυαδικούς πλήρεις αθροιστές. Κατά τη σύνδεση, πρέπει να ληφθεί υπόψη ότι το κρατούμενο της προηγούμενης βαθμίδας προστίθεται στην επόμενη βαθμίδα κ.ο.κ. Επειδή το άθροισμα δύο αριθμών n bit είναι δυνατόν να δώσει αποτέλεσμα με $n+1$ bit, θα πρέπει το κρατούμενο της νιοστής βαθμίδας, αντίθετα από τα υπόλοιπα κρατούμενα, να ληφθεί υπόψη κατά την παρουσίαση του αποτελέσματος. Στο σχήμα 4.17 δίνεται ένας αθροιστής δύο αριθμών που ο καθένας αποτελείται από τρία δυαδικά ψηφία. (Κοσσιδάς, 1996; Holdsworth, & Woods, 2002; Givone, 2002; Predko, 2005; Κοσσιδάς, & Γιαννακόπουλος, 2006; Godse, & Godse, 2011)



Σχήμα 4.17. Παράλληλος πλήρης αθροιστής

Η συνδεσμολογία αυτή παρουσιάζει ορισμένα προβλήματα. Ο αθροιστής, που προσθέτει τα δύο bit υψηλότερης τάξης, προκειμένου να εκτελέσει την πράξη της πρόσθεσης χρειάζεται να έχει και το κρατούμενο που προέκυψε από την προηγούμενη βαθμίδα. Είναι λοιπόν προφανές ότι θα πρέπει να περιμένει, για να πραγματοποιηθεί η πρόσθεση των δύο bit της προηγούμενης τάξης. Το ίδιο θα ισχύει και για τον αθροιστή των bit X_1 και Y_1 , ο οποίος θα πρέπει να περιμένει το κρατούμενο από τον αθροιστή των X_0 και Y_0 . Αν κάθε επιμέρους

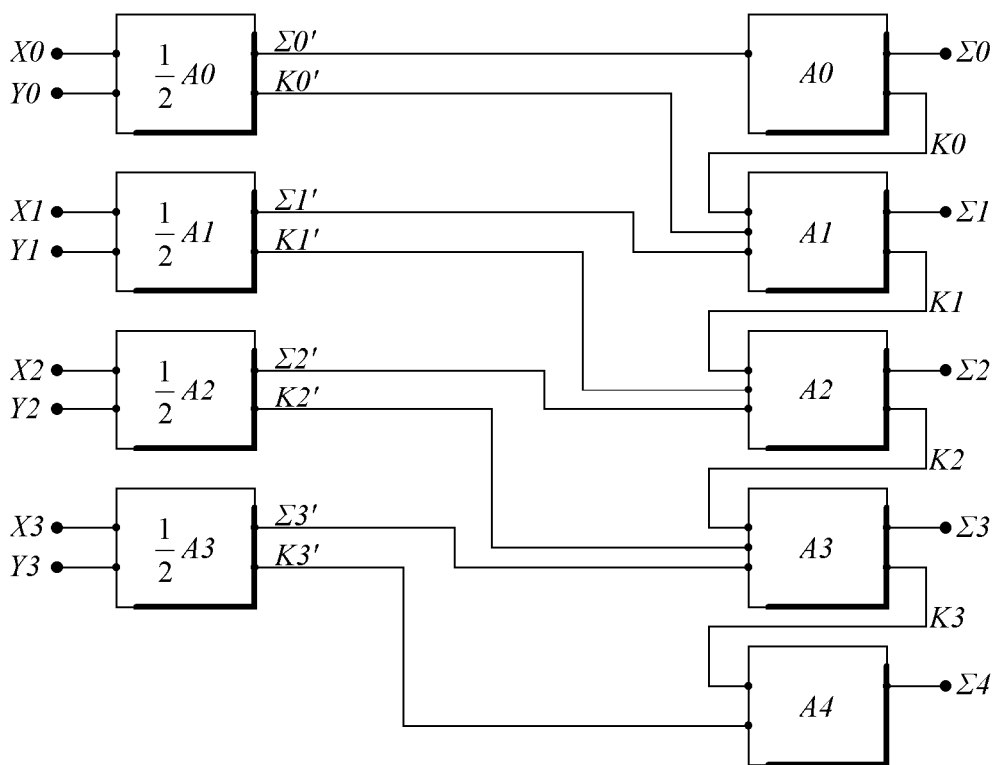
αθροιστής απαιτεί ένα χρόνο t , για να πραγματοποιήσει πρόσθεση δύο bit, τότε για να έχουμε το σωστό αποτέλεσμα στην έξοδο του κυκλώματος θα πρέπει να περιμένουμε ένα χρονικό διάστημα ίσο με $3 \cdot t$. Το πρόβλημα αυτό λύνεται με τη χρήση ενός συνθετότερου τύπου αθροιστή, με πρόβλεψη κρατούμενου, ο οποίος βάσει των τιμών των προηγούμενων bit, υπολογίζει πολύ γρήγορα τα επιμέρους κρατούμενα. Βεβαίως η πρόβλεψη κρατούμενου είναι αρκετά πολύπλοκη να πραγματοποιηθεί, γιατί απαιτεί μεγάλο αριθμό πυλών. Το τελευταίο κάνει δύσκολη την εφαρμογή αυτής της λύσης, ιδιαίτερα όταν οι προστιθέμενοι αριθμοί έχουν μεγάλο μήκος.

Μια λύση στο πρόβλημα αυτό δίνεται, αν υπολογισθεί το άθροισμα σε δύο φάσεις. Στην πρώτη φάση, αθροίζονται οι αριθμοί ψηφίο προς ψηφίο, παράγοντας τις ποσότητες Σ' και K' , όπου Σ' είναι το άθροισμα που προκύπτει χωρίς να ληφθούν υπόψη τα κρατούμενα και K' τα αποτελέσματα των επί μέρους κρατούμενων. Στη δεύτερη φάση αθροίζονται κατάλληλα τα Σ' και K' , παράγοντας το Σ . Το Σχήμα 4.18 παριστάνει ένα κύκλωμα αθροιστή τετρανήφικων δυαδικών αριθμών που εργάζεται με αυτή τη μέθοδο. Αρχικά οι δύο αριθμοί εφαρμόζονται ψηφίο προς ψηφίο σε ισάριθμους ημιαθροιστές σε παράλληλη μορφή. Στη συνέχεια, οι έξοδοι κάθε ημιαθροιστή στέλνονται στις εισόδους των πλήρων αθροιστών.

Τα κρατούμενα των ημιαθροιστών στέλνονται στις επόμενες βαθμίδες των αθροιστών μαζί με τα κρατούμενα των αθροιστών των προηγούμενων βαθμίδων. Οι εξισώσεις που περιγράφουν το κύκλωμα είναι οι:

$$\Sigma_i = (X_i + Y_i) \cdot \overline{(X_i \cdot Y_i)} \quad (4-14) \quad \text{και} \quad K_i = X_i \cdot Y_i \quad (4-15)$$

Οι εξισώσεις των πλήρων αθροιστών τριών εισόδων είναι οι ήδη γνωστές, με τη μόνη διαφορά ότι ο πίνακας αλήθειας, που τις καθορίζει, είναι πλέον απλοποιημένος, λόγω του ότι ορισμένοι συνδυασμοί αποκλείονται εκ των προτέρων. Πράγματι στις εξόδους των ημιαθροιστών είναι προφανές ότι είναι αδύνατη η συνύπαρξη παλμών Σ'_i και K'_i .



Σχήμα 4.18. Δομικό διάγραμμα δυαδικού αθροιστή τεσσάρων ψηφίων

Εξετάζοντας τον αθροιστή της πρώτης θέσης, προκύπτει ότι $K_0 = 0$ γιατί ο A_0 έχει μόνο μία είσοδο Σ_0 . Κατά συνέπεια, για να έχουμε από τον αθροιστή A_1 , $K_1 = 1$, πρέπει να έχουμε συγχρόνως $\Sigma'_1 = 1$ και $K'_0 = 1$. Επίσης, για να είναι το $K_2 = 1$, πρέπει τουλάχιστον σε δύο εισόδους του A_2 να εφαρμόζονται παλμοί. Δεδομένου ότι $\Sigma'_1 = 1$, θα είναι $K'_1 = 0$. Επομένως θα είναι $K_2 = 1$, μόνο εφόσον $\Sigma'_2 = 1$ και $K'_1 = 1$.

Συνεχίζοντας κατ' αυτόν τον τρόπο, συμπεραίνουμε ότι, εάν ο παλμός K_i είναι ίσος με «1», πρέπει υποχρεωτικά να είναι $\Sigma'_i = 1$ και επομένως $K'_i = 0$.

Δηλαδή είναι αδύνατο να είναι ταυτόχρονα «1» τα K_i και K'_i . Ο πίνακας αλήθειας, κατά συνέπεια, απλοποιείται και καταλήγει να έχει τη μορφή του πίνακα 4.13.

Σ'_i	K'_{i-1}	K_{i-1}	Σ_i	K_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1

Πίνακας 4.13. Άθροισμα και κρατούμενο δυαδικού αθροιστή τεσσάρων ψηφίων

Από τον πίνακα 4.13 προκύπτουν οι εξισώσεις:

$$K_i = \Sigma'_i \cdot (K'_{i-1} + K_{i-1}) \quad (4-16)$$

$$\Sigma_i = (\Sigma'_i + K'_{i-1} + K_{i-1}) \cdot [\Sigma'_i \cdot (K'_{i-1} + K_{i-1})] \quad (4-17)$$

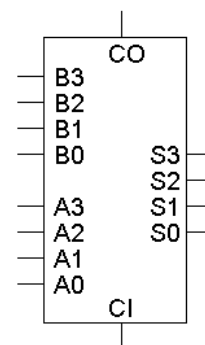
οι οποίες είναι αρκετά απλούστερες σε σύγκριση μ' αυτές που εξετάσαμε προηγουμένως.

4.9.1. Αθροιστής τετραψήφιων δυαδικών αριθμών με χρήση Ο.Κ.

Τέσσερις πλήρεις αθροιστές μαζί δίνουν έναν αθροιστή τετραψήφιων δυαδικών αριθμών. Σε έναν τέτοιο αθροιστή υπάρχουν συνολικά εννέα εισόδοι: Οι δυο αριθμοί των τεσσάρων δυαδικών ψηφίων, έστω A3A2A1A0 και B3B2B1B0 και το αρχικό κρατούμενο εισόδου, CI (Carry In).

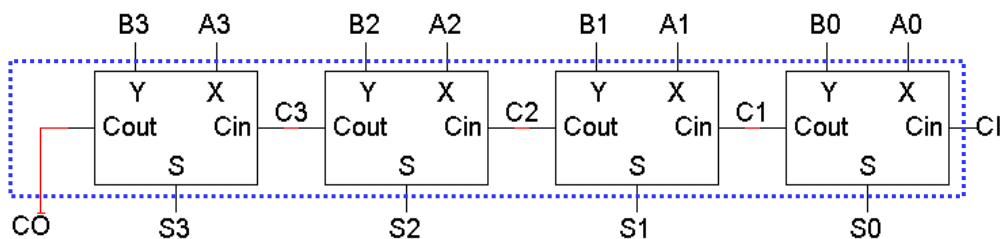
Οι πέντε εξόδοι είναι: Το άθροισμα, το οποίο είναι, επίσης, τετραψήφιο, έστω S3S2S1S0 και ένα κρατούμενο εξόδου, CO.

Φανταστείτε την περίπτωση όπου θα έπρεπε να σχεδιάσουμε έναν αθροιστή εννέα εισόδων χωρίς αυτή την ιεραρχική δομή - θα είχαμε έναν πίνακα αλήθειας με 512 γραμμές (2^9) και πέντε εξόδους!



Σχήμα 4.19. Δομικό διάγραμμα Ο.Κ. αθροιστή

Στο σχήμα 4.19 αποδίδεται το δομικό διάγραμμα του Ο.Κ. του αθροιστή δύο τετραψήφιων αριθμών. Στο Σχήμα 4.20 δίδεται το κύκλωμα του αθροιστή των 4- bits. (μη προσημασμένου)



Σχήμα 4.20. Αναλυτικό διάγραμμα αθροιστή τετραψήφιων αριθμών

Παράδειγμα – (πρόσθεση δύο τετραψήφιων δυαδικών αριθμών)

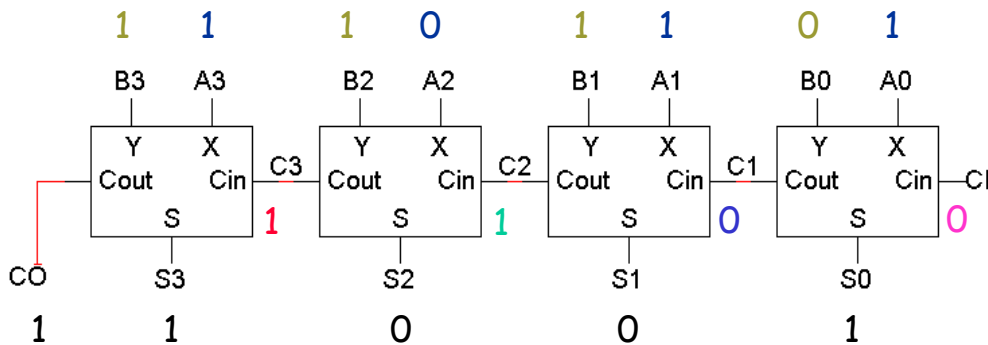
Ας τοποθετήσουμε στο κύκλωμα του προηγούμενου σχήματος 4.18: Έστω A=1011, και B=1110. (Σχήμα 4.21)

Βήμα 1: Τοποθετούμε σε όλες τις εισόδους τα δεδομένα και στο CI=0 (Αρχικοποίηση)

Βήμα 2: Το κύκλωμα παράγει το C1 και S0 ($1 + 0 + 0 = 01$)

- Βήμα 3:** Χρησιμοποιούμε το C1, για να βρούμε τα C2 και S1 $(1 + 1 + 0 = 10)$
Βήμα 4: Χρησιμοποιούμε το C2, για να υπολογίσουμε τα C3 και S2 $(0 + 1 + 1 = 10)$
Βήμα 5: Χρησιμοποιούμε το C3, για να υπολογίσουμε τα CO και S3 $(1 + 1 + 1 = 11)$

Το τελικό αποτέλεσμα είναι 11001.



Σχήμα 4.21. Παράδειγμα αθροιστή δύο τετραψήφιων αριθμών

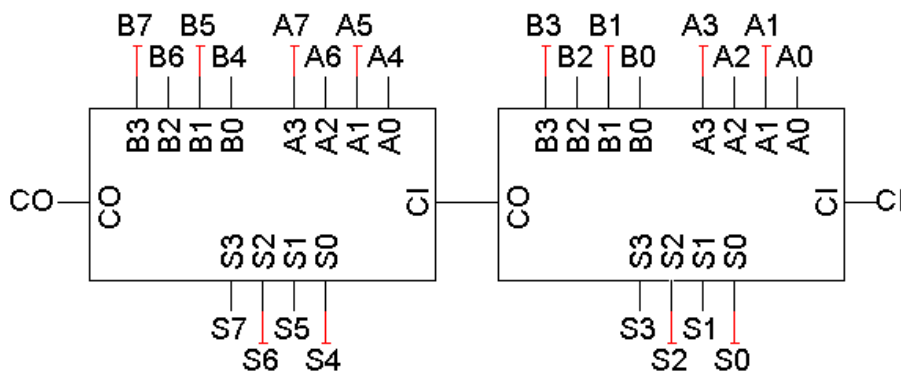
Υπερχείλιση

Στην περίπτωση αυτή, σημειώστε ότι η απάντηση (11001) έχει μήκος πέντε ψηφίων, αν και οι είσοδοί μας είχαν μόνο 4 ψηφία (1011 και 1110). Αυτό ονομάζεται *υπερχείλιση*.

Παρόλο που η απάντηση 11001 είναι σωστή, δεν μπορούμε να χρησιμοποιήσουμε το αποτέλεσμα αυτό σε κανένα επόμενο υπολογισμό με αθροιστή 4-bit. Στη μη προσημασμένη πρόσθεση, έχουμε υπερχείλιση, όταν το κρατούμενο εξόδου είναι 1.

Σχεδίαση Ιεραρχικού αθροιστή

Όταν προσθέτουμε δύο τετραψήφιους αριθμούς, το κρατούμενο εισόδου είναι πάντα 0. Αλλά τίθεται το ερώτημα γιατί ο αθροιστής των 4-bit πρέπει να έχει σαν είσοδο την CI (Είσοδος κρατουμένου). Ένας λόγος είναι, για να μπορέσουμε να τοποθετήσουμε δύο αθροιστές των 4-bit, ώστε να δημιουργήσουμε μεγαλύτερους αθροιστές. Στο σχήμα 4.22 απεικονίζεται ένας αθροιστής των 8-bit. Η είσοδος CI είναι, επίσης, χρήσιμη και στην αφαίρεση, όπως θα δούμε στη συνέχεια.



Σχήμα 4.22. Αθροιστής 8 δυαδικών ψηφίων

4.10. Σειριακός πλήρης αφαιρέτης (Serial full subtractor)

Για τη μελέτη του αφαιρέτη χρησιμοποιείται πίνακας αλήθειας, όπως και στον αθροιστή, μόνο που εδώ έχουμε έλλειμμα, και όχι περίσσειμα, σε περίπτωση ύπαρξης κρατουμένου. Ο πίνακας αλήθειας ενός σειριακού πλήρους αφαιρέτη είναι αυτός που φαίνεται στον πίνακα 4.14.

Από τον πίνακα 4.14 προκύπτουν οι παρακάτω εξισώσεις:

$$\Delta = \bar{X} \cdot \bar{Y} \cdot R + \bar{X} \cdot Y \cdot \bar{R} + X \cdot \bar{Y} \cdot \bar{R} + X \cdot Y \cdot R = \Delta = \bar{X} \cdot (\bar{Y} \cdot R + Y \cdot \bar{R}) + X \cdot (\bar{Y} \cdot \bar{R} + Y \cdot R) = \bar{X} \cdot (Y + R) \cdot (\bar{Y} \cdot \bar{R}) + X \cdot [(\bar{Y} + R) + Y \cdot R] \quad (4-18)$$

X	Y	Κρατούμενο R	Δ	R'
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Πίνακας 4.14. Πίνακας αλήθειας σειριακού δυαδικού αφαιρέτη

$$\begin{aligned}
 R' &= \bar{X} \cdot \bar{Y} \cdot R + \bar{X} \cdot Y \cdot \bar{R} + \bar{X} \cdot Y \cdot R + X \cdot Y \cdot R = \bar{X} \cdot (\bar{Y} \cdot R + Y \cdot \bar{R} + Y \cdot R) + X \cdot Y \cdot R = \\
 &= \bar{X} \cdot (\bar{Y} \cdot R + Y) + X \cdot Y \cdot R = \bar{X} \cdot (Y + R) + X \cdot Y \cdot R = \quad (4-19) \\
 &= \bar{X} \cdot Y + \bar{X} \cdot R + X \cdot Y \cdot R = \bar{X} \cdot Y + R \cdot (\bar{X} + Y) = \\
 &= \bar{X} \cdot Y + R \cdot \bar{X} + R \cdot Y = \bar{X} \cdot (Y + R) + R \cdot Y
 \end{aligned}$$

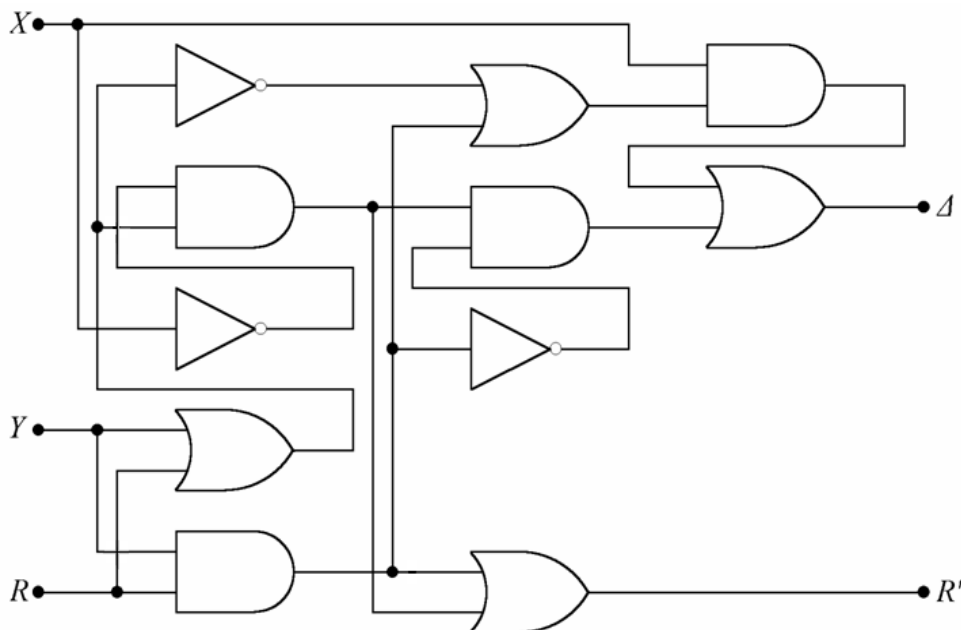
οι οποίες οδηγούν στο κύκλωμα του σχήματος 4.23. Συγκρίνοντας τους πίνακες αλήθειας του πλήρους αθροιστή και του πλήρους αφαιρέτη, προκύπτει ότι οι στήλες του αθροίσματος (Σ) και της διαφοράς (Δ) είναι ίδιες, ενώ υπάρχουν ορισμένες διαφορές στις στήλες των κρατούμενων (K') και (R'). Κατά συνέπεια, συνδέοντας έναν αθροιστή και έναν αφαιρέτη, το κύκλωμα του Σ και Δ θα παραμείνει το ίδιο, ενώ τα κυκλώματα παραγωγής του K' και του R' θα είναι διαφορετικά.

Συνενώνοντας μεταξύ τους τις εξισώσεις έχουμε:

$$\Sigma = \Delta = (X + Y + K) \cdot (\bar{X} \cdot Y + X \cdot K + Y \cdot K) + X \cdot Y \cdot K \quad (4-20)$$

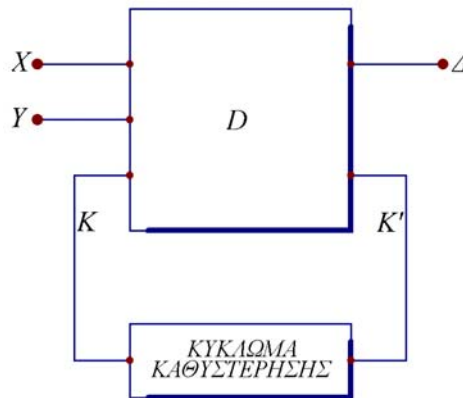
$$\begin{aligned}
 K' &= S \cdot (X \cdot Y + X \cdot K + Y \cdot K) + \bar{S} \cdot (\bar{X} \cdot Y + \bar{X} \cdot K + Y \cdot K) = \\
 &= S \cdot X \cdot (Y + K) + \bar{S} \cdot \bar{X} \cdot (Y + K) + Y \cdot K = (Y + K) \cdot (S \cdot X + \bar{S} \cdot \bar{X}) + Y \cdot K \quad (4-21)
 \end{aligned}$$

όπου με το S παριστάνουμε την πρόσθεση και με το \bar{S} την αφαίρεση. Σύμφωνα με τις προηγούμενες εξισώσεις μπορούμε να κατασκευάσουμε το αντίστοιχο λογικό κύκλωμα με λογικές πύλες.



Σχήμα 4.23 Σειριακός πλήρης αφαιρέτης

Η πραγματοποίηση του κυκλώματος του σειριακού αφαιρέτη βασίζεται στην αρχή της επαναφοράς του κρατούμενου στην είσοδο του αφαιρέτη και στη διαδοχική λήψη του αποτελέσματος από την έξοδο. Στο Σχήμα 4.24 αποδίδεται το δομικό διάγραμμα σειριακού δυαδικού αφαιρέτη.



Σχήμα 4.24. Δομικό διάγραμμα σειριακού δυαδικού αφαιρέτη

4.10.1.Κυκλώματα αθροιστών και αφαιρέτών

Κύκλωμα αφαίρεσης

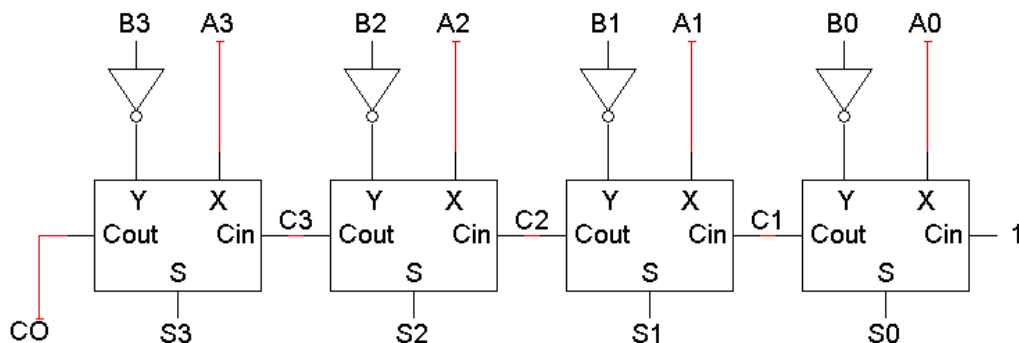
Θα μπορούσαμε να κατασκευάσουμε κατευθείαν ένα κύκλωμα αφαίρεσης με παρόμοιο τρόπο με αυτόν με τον οποίο κατασκευάσαμε μη προσημασμένους αθροιστές. Μας είναι ήδη γνωστή η διαδικασία μετατροπής της αφαίρεσης σε πρόσθεση, χρησιμοποιώντας το συμπλήρωμα ως προς δύο. Αλγεβρικά έχουμε, $A - B = A + (-B)$.

Έτσι, για να αφαιρέσουμε το B από το A, μπορούμε να προσθέσουμε τον αντίθετο του B στον A. Με αυτόν τον τρόπο μπορούμε να ξαναχρησιμοποιήσουμε το κύκλωμα του μη προσημασμένου αθροιστή, που έχουμε ήδη μελετήσει. (Σχήμα 4.33)

Κύκλωμα αφαίρεσης τετραψήφιων αριθμών με συμπλήρωμα ως προς δύο

Για να βρούμε το $A - B$ με έναν αθροιστή, πρέπει να συμπληρώσουμε κάθε bit του B. Κατόπιν θέτουμε στο κρατούμενο εισόδου του πρώτου αθροιστή το 1. Το τελικό αποτέλεσμα είναι $A + B' + 1$, όπου το $B' + 1$ είναι το συμπλήρωμα ως προς δύο, του B. (Σχήμα 4.25)

Λάβετε υπόψη σας πως τα A_3, B_3 και S_3 εδώ, είναι στην πραγματικότητα τα bits προσήμων.



Σχήμα 4.25. Κύκλωμα αφαίρεσης 4-bit με συμπλήρωμα ως προς δύο

Σύγκριση των κυκλωμάτων του αθροιστή και του αφαιρέτη

Οι μόνες διαφορές μεταξύ των κυκλωμάτων αθροιστή και αφαιρέτη είναι ότι:

- Ο αφαιρέτης πρέπει να αντιστρέψει τα $B_3B_2B_1B_0$.
- Ο αφαιρέτης θέτει το αρχικό κρατούμενο σε 1, αντί για 0.

Συνεπώς δεν είναι δύσκολο να φτιάξουμε ένα κύκλωμα που θα εκτελεί συγχρόνως και πρόσθεση και αφαίρεση.

Κύκλωμα αθροιστή-αφαιρέτη

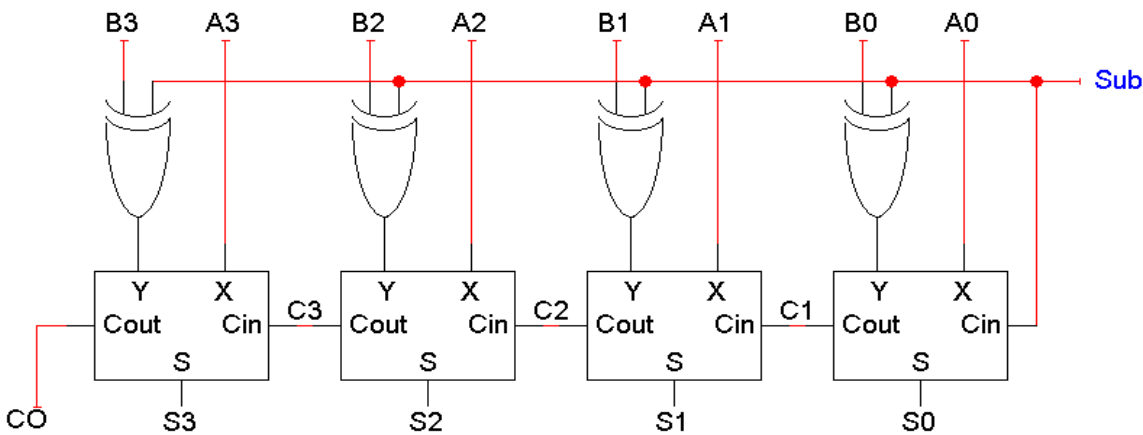
Οι πύλες XOR μας επιτρέπουν να συμπληρώσουμε επιλεκτικά την είσοδο B με βάση τις σχέσεις:

$$X \oplus 0 = X \quad X \oplus 1 = X'$$

Όταν η Sub = 0, οι έξοδοι των πυλών XOR B3' B2' B1' B0' και το κρατούμενο εισόδου είναι 0. Η έξοδος του αθροιστή θα είναι A + B + 0, ή A + B.

Όταν η Sub = 1, οι έξοδοι των πυλών XOR B3' B2' B1' B0' και το κρατούμενο εισόδου είναι 1. Έτσι η έξοδος του αθροιστή θα είναι μια αφαίρεση με συμπλήρωμα ως προς δύο, A - B.

Στο Σχήμα 4.26 που ακολουθεί απεικονίζεται το κύκλωμα του αθροιστή-αφαιρέτη.

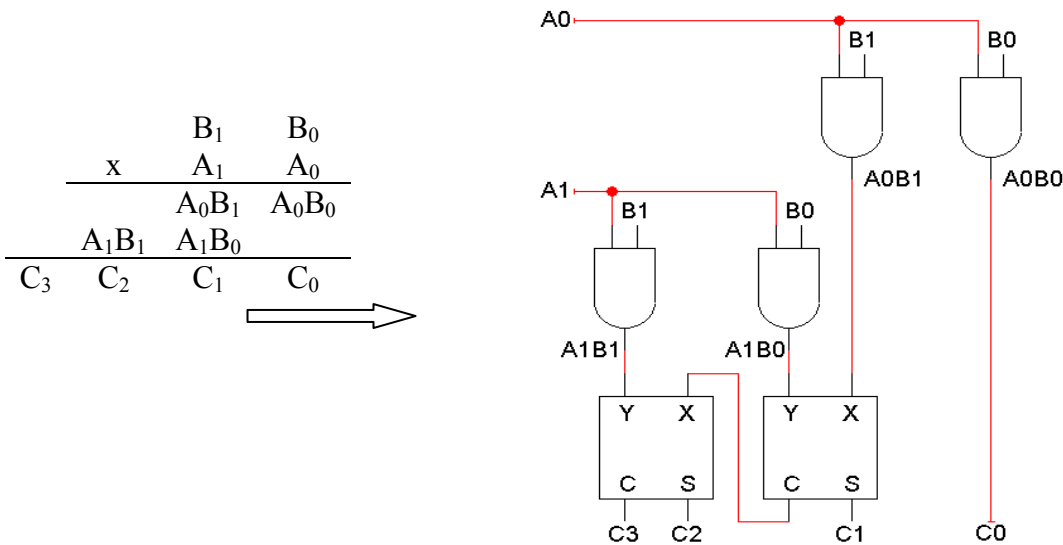


Σχήμα 4.26. Κύκλωμα αθροιστή- αφαιρέτη 4-bit

4.10.2.Κύκλωμα δυαδικού πολλαπλασιαστή

Οι πύλες AND παράγουν τα μερικά γινόμενα. Για ένα διψήφιο επί διψήφιο πολλαπλασιαστή, μπορούμε απλά να χρησιμοποιήσουμε δύο ημιαθροιστές, για να αθροίσουμε τα μερικά γινόμενα. Γενικά όμως θα χρησιμοποιήσουμε πλήρεις αθροιστές.

Τα C₃-C₀ του πίνακα είναι τα γινόμενα ή/και τα αθροίσματα των μερικών γινομένων, και όχι τα κρατούμενα! (Στο Σχήμα 4.27 απεικονίζεται ένας 2x2 δυαδικός πολλαπλασιαστής) και στο κύκλωμα του σχήματος 4.28 αποδίδεται το πλήρες διάγραμμα ενός δυαδικού πολλαπλασιαστή 4x4.

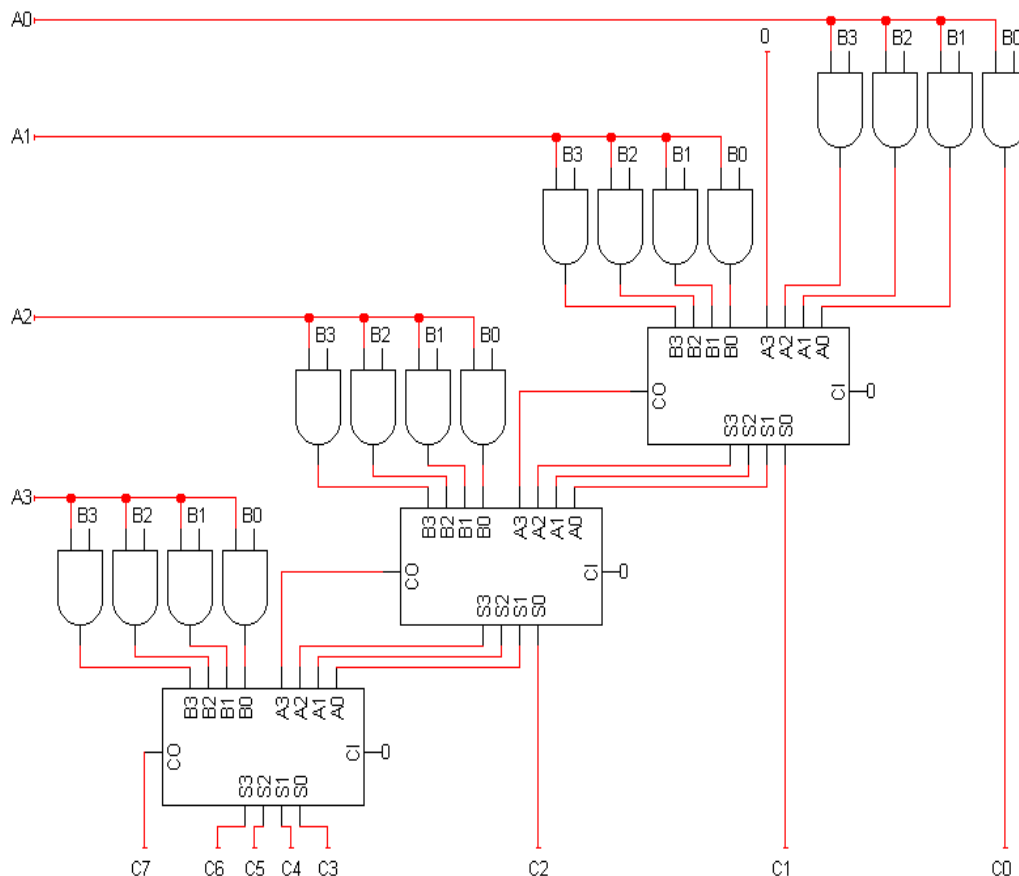


Σχήμα 4.27. Δυαδικός πολλαπλασιαστής 2x2

Ο πολλαπλασιαστής των τεσσάρων ψηφίων παράγει ένα αποτέλεσμα οκτώ ψηφίων. Θα μπορούσαμε να κρατήσουμε και τα οκτώ ψηφία του αποτελέσματος ή, αν θέλαμε αποτέλεσμα τεσσάρων ψηφίων, θα μπορούσαμε να αγνοήσουμε τα ψηφία C4-C7, και στην περίπτωση που το αποτέλεσμα είναι μεγαλύτερο από 4 ψηφία, να το θεωρήσουμε σαν κατάσταση υπερχείλισης.

Οι πολλαπλασιαστές είναι γενικά πολύπλοκα κυκλώματα. Έτσι, όταν πολλαπλασιάζουμε έναν αριθμό m -ψηφίων με έναν αριθμό n -ψηφίων:

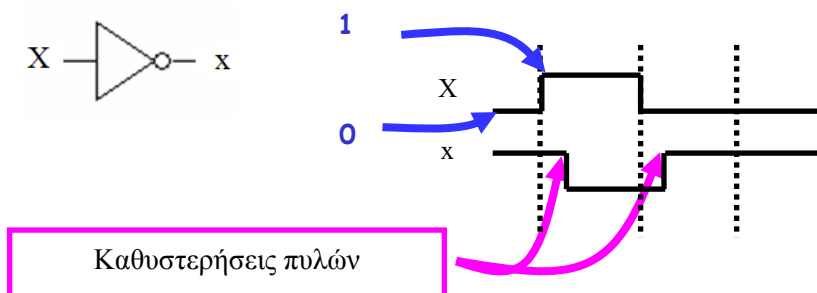
Υπάρχουν n μερικά γινόμενα, ένα για κάθε ψηφίο του πολλαπλασιαστή. Αυτό απαιτεί $n-1$ αθροιστές, ο καθένας των οποίων θα μπορεί να προσθέσει m ψηφία (το μέγεθος του πολλαπλασιαστέου). Το κύκλωμα για πολλαπλασιασμό 32-bit ή 64-bit θα ήταν τεράστιο!



Σχήμα 4.28. Αναλυτικό διάγραμμα δυαδικού πολλαπλασιαστή 4x4

4.11. Καθυστερήσεις πυλών

Κάθε πύλη χρειάζεται κάποιο χρόνο από τη στιγμή που θα δοθεί το σήμα στην είσοδο μέχρι να εμφανιστούν στην έξοδο τα σωστά αποτελέσματα. Αυτό το χρονικό διάστημα ονομάζεται *καθυστερήση πύλης*.



Σχήμα 4.29. Γραφική απεικόνιση των καθυστερήσεων των πυλών

Υπάρχουν πολλοί λεπτομερείς τρόποι υπολογισμού των καθυστερήσεων κάθε πύλης, που είναι όμως αρκετά πολύπλοκοι. Ας θεωρήσουμε πως αυτή η καθυστέρηση είναι σταθερή και ότι είναι η ίδια για όλες τις πύλες. Μπορούμε να χρησιμοποιήσουμε ένα διάγραμμα χρονισμού, για να δείξουμε γραφικά τις καθυστερήσεις των πυλών (Σχήμα 4.29).

Υπολογισμός των κρατούμενων εξόδου

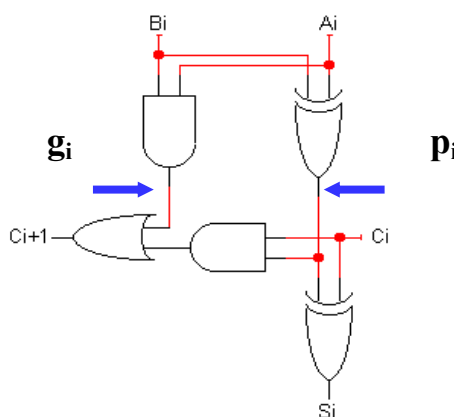
Μπορούμε, αντί να περιμένουμε τα κρατούμενα των εξόδων από όλες τις προηγούμενες βαθμίδες, να τα υπολογίσουμε απευθείας με ένα κύκλωμα δύο επιπέδων (Σχήμα 4.30), μειώνοντας έτσι την καθυστέρηση. Πρώτα ορίζουμε δύο συναρτήσεις:

Την «παράγουσα συνάρτηση» g_i που θα παράγει 1, όταν πρέπει να υπάρχει κρατούμενο εξόδου από τη θέση i (δηλ. όταν το A_i και το B_i είναι 1). $g_i = A_i B_i$

Τη συνάρτηση «διάδοσης» p_i που είναι αληθής, όταν, εάν υπάρχει ένα εισερχόμενο κρατούμενο, αυτό «διαδίδεται». (δηλ. όταν ή $A_i=1$ ή $B_i=1$, αλλά ποτέ και τα δύο μαζί) $p_i = A_i \oplus B_i$ Είναι προφανές ότι το αρχικό κρατούμενο είναι μηδέν.

Μπορούμε λοιπόν να ξαναγράψουμε τη συνάρτηση του κρατουμένου εξόδου ως: $c_{i+1} = g_i + p_i c_i$

A_i	B_i	C_i	C_{i+1}
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



Σχήμα 4.30. Βασικό κύκλωμα δύο επιπέδων

Ας δούμε τις εξισώσεις των κρατουμένων εξόδου για συγκεκριμένα ψηφία, χρησιμοποιώντας την προηγούμενη γενική εξίσωση: $c_{i+1} = g_i + p_i c_i$. Έτσι έχουμε:

$$c_1 = g_0 + p_0 c_0$$

$$c_2 = g_1 + p_1 c_1 = g_1 + p_1 (g_0 + p_0 c_0) = g_1 + p_1 g_0 + p_1 p_0 c_0$$

$$c_3 = g_2 + p_2 c_2 = g_2 + p_2 (g_1 + p_1 g_0 + p_1 p_0 c_0) = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0$$

$$c_4 = g_3 + p_3 c_3 = g_3 + p_3 (g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0) = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_0$$

4.12. Κυκλώματα σύγκρισης

Τα κυκλώματα σύγκρισης (Comparators) χρησιμεύουν για τη σύγκριση δυαδικών αριθμών ή αριθμών άλλου συστήματος αρίθμησης, που είναι κωδικοποιημένοι στο δυαδικό. Οι προς σύγκριση αριθμοί μπορούν να θεωρηθούν ότι μεταφέρουν, ή όχι, πληροφορία προσήμου, λαμβάνοντας υπόψη ότι το ψηφίο μεγαλύτερης τάξης παριστάνει το πρόσημο του αριθμού. Κατά συνέπεια, για τη σύγκριση μεταξύ ετερόσημων αριθμών είναι αρκετό να συγκριθούν μόνο τα πρόσημα, ενώ μεταξύ των ομόσημων να ληφθεί υπόψη η απόλυτη τιμή τους. Στη δεύτερη περίπτωση, εάν είναι και οι δύο θετικοί ο μεγαλύτερος βρίσκεται κανονικά, ενώ, εάν είναι και οι δύο αρνητικοί, ο μεγαλύτερος θα είναι αυτός που θα έχει τη μικρότερη απόλυτη τιμή. Οι προς σύγκριση αριθμοί έχουν πάντοτε τα ψηφία μικρότερης τάξης στο δεξιότερο μέρος τους. (Κοσσίδης, & Γιαννακόπουλος, 2006; Balabanian, & Carlson, 2007; Tocci, et al., 2010)

4.12.1. Σειριακός δυαδικός συγκριτής

Ο σειριακός δυαδικός συγκριτής (serial binary comparator) εκτελεί τη σε σειρά σύγκριση δύο αριθμών συγκρίνοντας τα δυαδικά ψηφία του ενός αριθμού με τα αντίστοιχα του άλλου, έως ότου φθάσουμε στο μεγαλύτερης τάξης ψηφίο. Κατά συνέπεια, η σύγκριση δύο ψηφίων θα έχει ένα από τα αποτελέσματα που φαίνονται στον πίνακα 4.15.

Ένας ηλεκτρονικός υπολογιστής είναι σε θέση να χρησιμοποιήσει κατά διάφορους τρόπους το αποτέλεσμα μιας σύγκρισης. Υπάρχουν υπολογιστές που, συγκρίνοντας δύο δυαδικά ψηφία, είναι ικανοί να δίνουν όλα τα δυνατά αποτελέσματα $>$, \geq , $=$, \leq , $<$, \neq , άλλοι που δίνουν μερικά από αυτά και τέλος άλλοι που είναι σε θέση να δώσουν σαν αποτέλεσμα μόνο την ένδειξη $=$ ή την ένδειξη \neq . Το βασικό κύκλωμα σύγκρισης σε κάθε συγκριτή είναι ο δυαδικός συγκριτής, που αποδίδεται στο Σχήμα 4.33. Το κύκλωμα αυτό προκύπτει εύκολα από τον πίνακα 4.16. (Με «1» παριστάνεται η ισότητα και με «0» η ανισότητα).

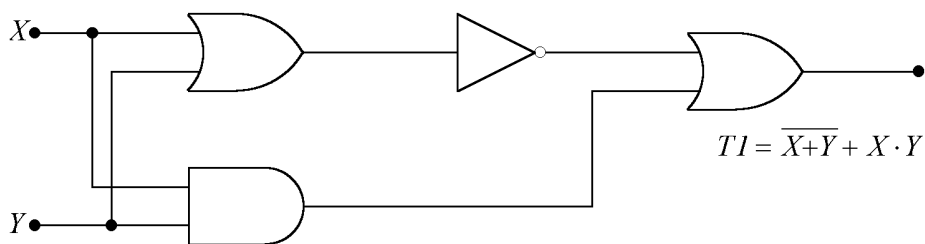
X	Y	Αποτέλεσμα
0	0	=
0	1	<
1	0	>
1	1	=

Πίνακας 4.15. Αποτελέσματα σύγκρισης δύο ψηφίων

X	Y	T ₁
0	0	1
0	1	0
1	0	0
1	1	1

Πίνακας 4.16. Πίνακας αλήθειας δυαδικού συγκριτή ισότητας

Από τον πίνακα προκύπτει ότι: $T_1 = \overline{X} \cdot \overline{Y} + X \cdot Y = \overline{X + Y} + X \cdot Y$ (XNOR)



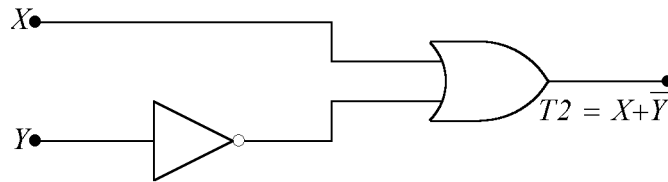
Σχήμα 4.33. Υλοποίηση δυαδικού συγκριτή ισότητας

Εάν ο δυαδικός συγκριτής έχει σχεδιαστεί, ώστε να ανταποκρίνεται στις περιπτώσεις **μεγαλύτερο ή ίσο (\geq) και μικρότερο ($<$)**, δίνοντας αποτέλεσμα «1» και «0» αντίστοιχα, θα έχουμε τον παρακάτω πίνακα 4.17.

X	Y	T ₂
0	0	1
0	1	0
1	0	1
1	1	1

Πίνακας 4.17. Πίνακας αλήθειας δυαδικού συγκριτή

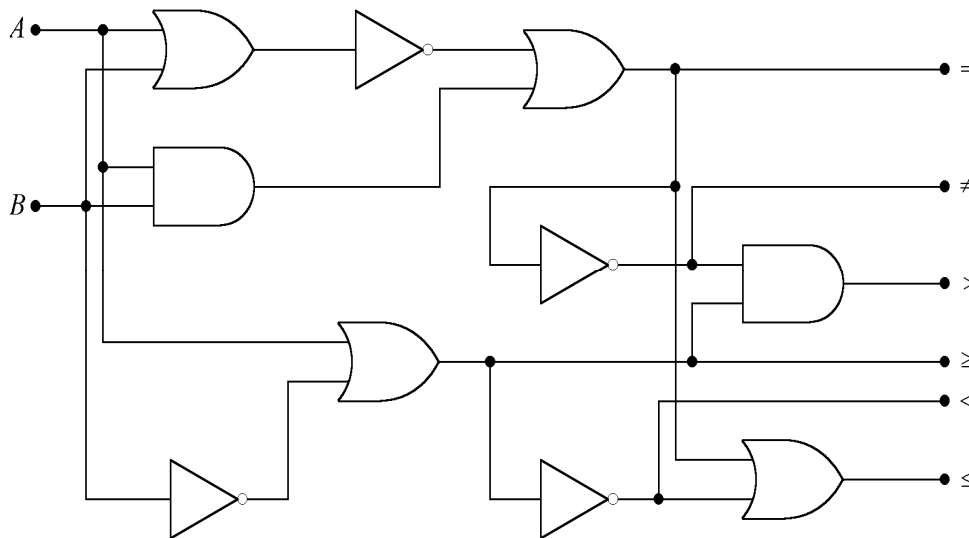
Η λογική συνάρτηση, που προκύπτει από τον ανωτέρω πίνακα αλήθειας, είναι η: $T_2 = \overline{X} \cdot \overline{Y} + X \cdot \overline{Y} + X \cdot Y = X + \overline{Y}$ και το αντίστοιχο κύκλωμα σύγκρισης δίνεται στο Σχήμα 4.34.



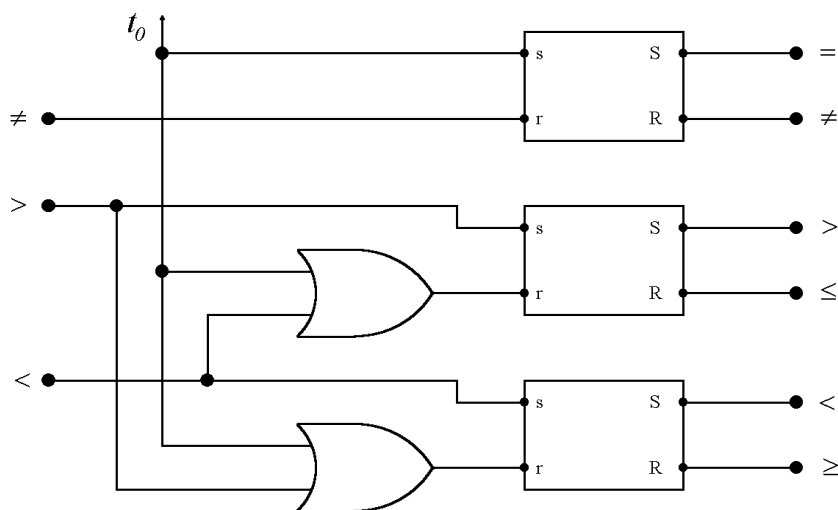
Σχήμα 4.34. Δυαδικός συγκριτής με πύλες

Συνενώνοντας τα δύο κυκλώματα, που υπολογίσαμε προηγουμένως, επιτυγχάνουμε ένα ενιαίο κύκλωμα (Σχήμα 4.35α) που είναι σε θέση να αναδείξει όλα τα δυνατά αποτελέσματα μιας σύγκρισης. Ο συγκριτής αυτός ανταποκρίνεται σε συγκρίσεις αριθμών του ενός ψηφίου. Σε περίπτωση αριθμών με περισσότερα ψηφία συγκρίνονται διαδοχικά όλα τα ψηφία, αρχίζοντας από αυτά της μικρότερης τάξης. Συνέπεια αυτού του τρόπου σύγκρισης είναι να έχουμε, πολλές φορές, λανθασμένο αποτέλεσμα, αφού οι έξοδοι του κυκλώματος θα διατηρούν το αποτέλεσμα της σύγκρισης των δύο τελευταίων ψηφίων, έχοντας «ξεχάσει» τα αποτελέσματα των προηγουμένων συγκρίσεων. Πράγματι, συγκρίνοντας τους δεκαδικούς αριθμούς 5240 και 6421, η ένδειξη που προκύπτει από τη σύγκριση των ψηφίων με μεγαλύτερο βάρος θα είναι το ίσον (=), αντίθετα προς την πραγματική ένδειξη που είναι το μικρότερο (<). Αυτό το σφάλμα παρακάμπτεται, καταχωρώντας την τελευταία ένδειξη ανισότητας, η οποία δεν πρέπει να μεταβάλλεται από τυχόν υπάρχουσες στη συνέχεια ισότητες. Η ανισότητα καταγράφεται με τη βοήθεια R-S *f-f*, όπως αυτών που φαίνονται στο Σχήμα 4.35β. (Τη λειτουργία των *f-f* θα μελετήσουμε στο έκτο κεφάλαιο).

Τα *f-f* καταγραφής αρχικά βρίσκονται όλα σε κατάσταση που να περιλαμβάνει την ισότητα (\geq , $=$, \leq). Αυτό επιτυγχάνεται εφαρμόζοντας έναν παλμό t_0 στην είσοδο των *f-f*, προτού αρχίσει η εφαρμογή των ψηφίων σύγκρισης στο κύκλωμα. Επομένως το *f-f* που παριστάνει το « \neq » ή το « \neq » παραμένει στην κατάσταση του « \neq », εφόσον εφαρμοσθεί παλμός ανισότητας. Κατ' αναλογία, για τις περιπτώσεις των επομένων *f-f* οι καταστάσεις « \leq » και « \geq » παραμένουν, έως ότου εφαρμοσθούν παλμοί « $>$ » ή « $<$ ».



Σχήμα 4.35α. Αναλυτικό διάγραμμα πλήρους σειριακού δυαδικού συγκριτή με πύλες

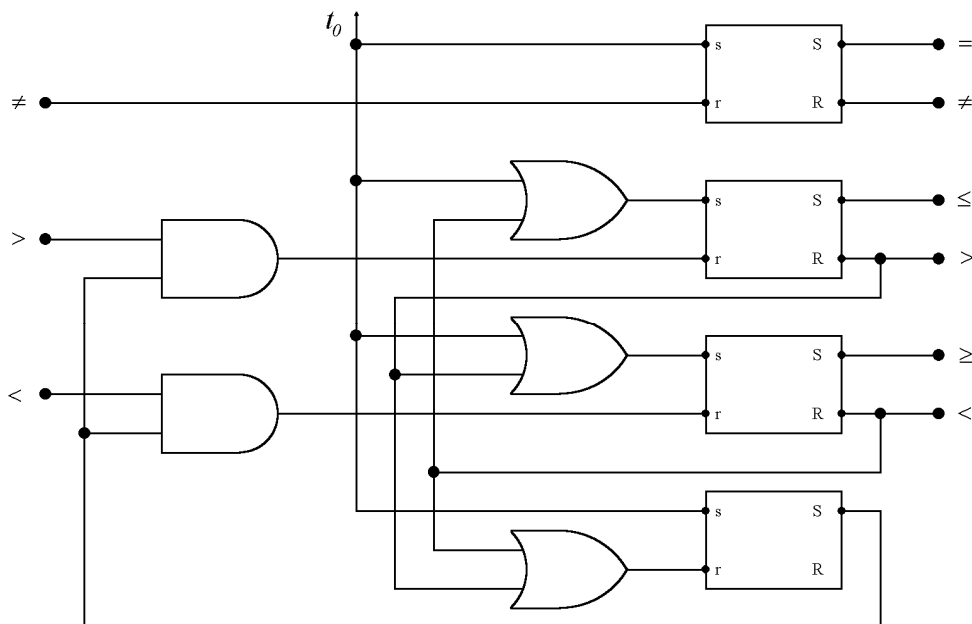


Σχήμα 4.35β. Αναλυτικό διάγραμμα πλήρους σειριακού δυαδικού συγκριτή

4.12.2. Παράλληλος δυαδικός συγκριτής

Μία απευθείας σύγκριση δύο αριθμών καταχωρημένων παράλληλα θα οδηγούσε σε πολύπλοκα κυκλώματα στην περίπτωση της σειριακής σύγκρισης δεδομένου ότι, εάν οι αριθμοί, που συγκρίνονται διαφέρουν ως προς το ψηφίο μικρότερης τάξης, το αποτέλεσμα εξαρτάται από τον διαδοχικό έλεγχο όλων των προηγούμενων ψηφίων, και απαιτούνται κυκλώματα που να συνδέουν ταυτόχρονα όλα τα υπάρχοντα ψηφία των δύο προς σύγκριση αριθμών. Λύση στο πρόβλημα αποτελεί η ανά ζεύγη σύγκριση των ψηφίων της ίδιας τάξης, με τη χρήση πυλών XNOR, όσων και το πλήθος των ψηφίων των αριθμών. Το αποτέλεσμα των εξόδων των πυλών οδηγείται σε μια πύλη AND της οποίας το αποτέλεσμα είναι «1» μόνο εάν όλα τα ψηφία των αριθμών έχουν την ίδια τιμή.

Μία άλλη λύση στο παραπάνω πρόβλημα επιτυγχάνεται, με τη σειριακή σύγκριση των ψηφίων των αριθμών, αρχίζοντας από τα ψηφία μεγαλύτερης τάξης. Ο συγκριτής αυτός ονομάζεται παράλληλος δυαδικός συγκριτής (Parallel binary comparator). Το κύκλωμά του αποδίδεται στο Σχήμα 4.36.



Σχήμα 4.36. Σειριακός δυαδικός συγκριτής αρχίζοντας από τα ψηφία μεγαλύτερης τάξης

4.12.3. Συγκριτής δύο δυαδικών ψηφίων

Έστω ότι θέλουμε να σχεδιάσουμε ένα κύκλωμα που θα συγκρίνει δύο αριθμούς των 2-bit, τον A και τον B. Το κύκλωμα πρέπει να έχει τρεις εξόδους:

- **G** («Greater») : θα είναι 1 μόνο όταν $A > B$ (μεγαλύτερο).
- **E** («Equal») : θα είναι 1 μόνο όταν $A = B$ (ίσο).
- **L** («Lesser») : θα είναι 1 μόνο όταν $A < B$ (μικρότερο).

Οι εισόδοι A και B θα είναι 00, 01, 10, ή 11 (0, 1, 2 ή 3 στο δεκαδικό). Για οποιαδήποτε είσοδο A και B, μόνο μια από τις τρεις εξόδους θα είναι 1.

Βήμα 1: Πόσες εισόδους και πόσες εξόδους θα έχουμε;

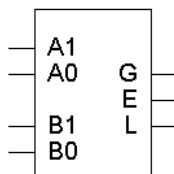
Δύο αριθμοί των 2-bit μας δίνουν συνολικά τέσσερις εισόδους.

- Αρχικά θα πρέπει να ονομάσουμε καθεμία από τις εισόδους.

Έστω ότι ο πρώτος αριθμός αποτελείται από τα ψηφία A1 και A0 από τα αριστερά προς τα δεξιά και ο δεύτερος αριθμός αποτελείται από τα ψηφία B1 και B0.

- Το πρόβλημα απαιτεί τρεις εξόδους: G, E και L.

Στο σχήμα 4.37 αποδίδεται το δομικό διάγραμμα στο οποίο απεικονίζονται σαφώς οι εισόδοι και οι εξόδοι. Αφού προσδιορίσουμε τις εξισώσεις, θα μπορέσουμε να σχεδιάσουμε τα στοιχεία του κυκλώματος, που περιέχονται μέσα στο δομικό στοιχείο.



Σχήμα 4.37. Δομικό διάγραμμα συγκριτή δύο δυαδικών ψηφίων

Βήμα 2: Προσδιορισμός της λειτουργίας

Για το πρόβλημα αυτό είναι ευκολότερο να αρχίσουμε με τον πίνακα αληθείας (πίνακας 4.18).

Με αυτόν τον τρόπο μπορούμε να δείξουμε σαφώς τις σχέσεις (>, =, <) μεταξύ των εισόδων.

Μια συνάρτηση τεσσάρων εισόδων έχει έναν πίνακα αληθείας δεκαέξι γραμμών. Συνήθως είναι πιο εύκολο να τοποθετήσουμε τις γραμμές του πίνακα αληθείας με δυαδική σειρά. Σε αυτή την περίπτωση, από 0000 ως 1111 για τα A1, A0, B1 και B0.

A1	A0	B1	B0	G	E	L
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Πίνακας 4.18. Πίνακας αληθείας συγκριτή δύο δυαδικών ψηφίων

Για παράδειγμα, έστω ότι έχουμε να συγκρίνουμε το 01 με το 10. Το $01 < 10$ και έτσι η έκτη σειρά του πίνακα αληθείας, (που αντιστοιχεί στις εισόδους $A=01$ και $B=10$) μας δίνει την έξοδο $L=1$, και, συνεπώς, οι έξοδοι G και E είναι και οι δύο 0. Σκεπτόμενοι με το ίδιο τρόπο, συμπληρώνουμε τις εξόδους του πίνακα 3.20.

Βήμα 3: Απλουστευμένες Boolean εκφράσεις

Για την απλοποίηση θα χρησιμοποιήσουμε το χάρτη Karnaugh. Υπάρχουν τρεις συναρτήσεις (και για τις τρεις οι εισόδοι είναι κοινές $\overline{A1} A0 B1 B0$). Έτσι χρειαζόμαστε τρεις χάρτες Karnaugh.

		B1			
		0	0	0	0
		1	0	0	0
A1		1	1	0	1
		1	1	0	0
		B0			

$$G(A1, A0, B1, B0) = A1 A0 \overline{B0} + A0 \overline{B1} \overline{B0} + A1 \overline{B1}$$

		B1			
		1	0	0	0
		0	1	0	0
A1		0	0	1	0
		0	0	0	1
		B0			

$$E(A1, A0, B1, B0) = \overline{A1} \overline{A0} \overline{B1} \overline{B0} + \overline{A1} A0 \overline{B1} B0 + A1 A0 B1 B0 + A1 \overline{A0} B1 \overline{B0}$$

		B1			
		0	1	1	1
		0	0	1	1
A1		0	0	0	0
		0	0	1	0
		B0			

$$L(A1, A0, B1, B0) = \overline{A1} \overline{A0} B0 + \overline{A0} B1 B0 + \overline{A1} B1$$

Βήμα 4: Σχεδιασμός του κυκλώματος

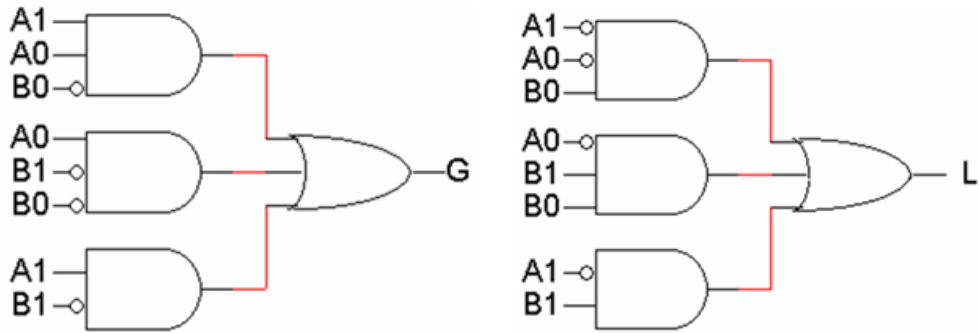
Με δεδομένες τις απλοποιημένες συναρτήσεις, όπως προέκυψαν μετά την απλοποίηση:

$$G = A1 A0 \overline{B0} + A0 \overline{B1} \overline{B0} + A1 \overline{B1}$$

$$E = \overline{A1} \overline{A0} \overline{B1} \overline{B0} + \overline{A1} A0 \overline{B1} B0 + A1 A0 B1 B0 + A1 \overline{A0} B1 \overline{B0}$$

$$L = \overline{A1} \overline{A0} B0 + \overline{A0} B1 B0 + \overline{A1} B1$$

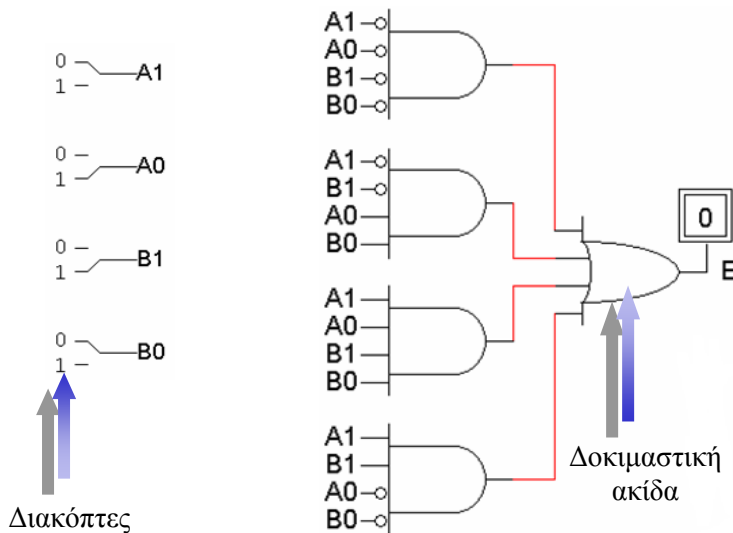
προχωρούμε στην υλοποίηση της κάθε συνάρτησης ξεχωριστά: (Σχήμα 4.38). Το κύκλωμα της συνάρτησης E αποδίδεται στο Σχήμα 4.39 με τη χρήση της δοκιμαστικής ακίδας καθώς και των διακοπών εισόδου.



Σχήμα 4.38. Υλοποίηση των συναρτήσεων G και L του συγκριτή

Έλεγχος του κυκλώματος με λογισμικό

Οι είσοδοι προφανώς προέρχονται από δυαδικούς διακόπτες. Για να δούμε και να ελέγξουμε τις εξόδους, χρησιμοποιούμε δυαδικές δοκιμαστικές ακίδες. Στο σχήμα που ακολουθεί αποδίδεται ο έλεγχος του κυκλώματος E σε περίπτωση στην οποία δεν ισχύει η ισότητα ($E=0$) (Σχήμα 4.39). Ο ίδιος έλεγχος στα κυκλώματα των G, L θα έδινε 1 στο κύκλωμα του L .



Σχήμα 4.39. Έλεγχος του κυκλώματος ισοδυναμίας « E » του συγκριτή

Για την αναπαράσταση των δεδομένων:

Χρησιμοποιήσαμε τρεις εξόδους, μία για κάθε πιθανό αποτέλεσμα της σύγκρισης των αριθμών, να είναι δηλαδή ο ένας αριθμός μεγαλύτερος, ίσος ή μικρότερος από τον άλλον. Αυτό ονομάζεται και κώδικας 1-από-3.

Πλεονεκτήματα και περιορισμοί των χαρτών Karnaugh

Τα κυκλώματά μας είναι υλοποιήσεις δύο επιπέδων, και είναι σχετικά εύκολο να τα σχεδιάσουμε και να τα παρακολουθήσουμε. Η προηγούμενη συνάρτηση $E(A1, A0, B1, B0)$ δεν μπορεί να απλοποιηθεί περαιτέρω με το χάρτη Karnaugh. Μπορούσαμε άραγε να καταφέρουμε κάτι καλύτερο, αν χρησιμοποιήσουμε την άλγεβρα Boole;

Επεκτασιμότητα

Χρησιμοποιήσαμε μια δυναμική προσέγγιση, καταγράφοντας όλες τις πιθανές εισόδους και εξόδους. Αυτό κάνει πιο δύσκολη την επέκταση του κυκλώματός μας για σύγκριση αριθμών με 3-bit, για παράδειγμα.

4.13. Κυκλώματα μετατροπής κωδίκων

Έστω ότι θέλουμε να μετατρέψουμε κώδικα BCD 8421 σε κώδικα BCD Excess 3 (XS3). (πίνακας 4.19) Για τη μετατροπή, εφαρμόζουμε τις αρχές που έχουμε αναφέρει. Αρχικά σχηματίζουμε τον πίνακα αλήθειας των προς μετατροπή κωδικών και, στη συνέχεια, απλοποιούμε με τους χάρτες Karnaugh. Τέλος σχεδιάζουμε το αντίστοιχο λογικό κύκλωμα. (Nashelsky, 1994; Κοσσιδάς, 1996)

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

Πίνακας 4.19. Μετατροπή του BCD 8421 σε BCD XS3

Οι χάρτες Karnaugh και οι εξισώσεις που προκύπτουν είναι:

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

$$X = A + B \cdot D + B \cdot C$$

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

$$Y = \bar{B} \cdot C + \bar{B} \cdot D + B \cdot \bar{C} \cdot \bar{D}$$

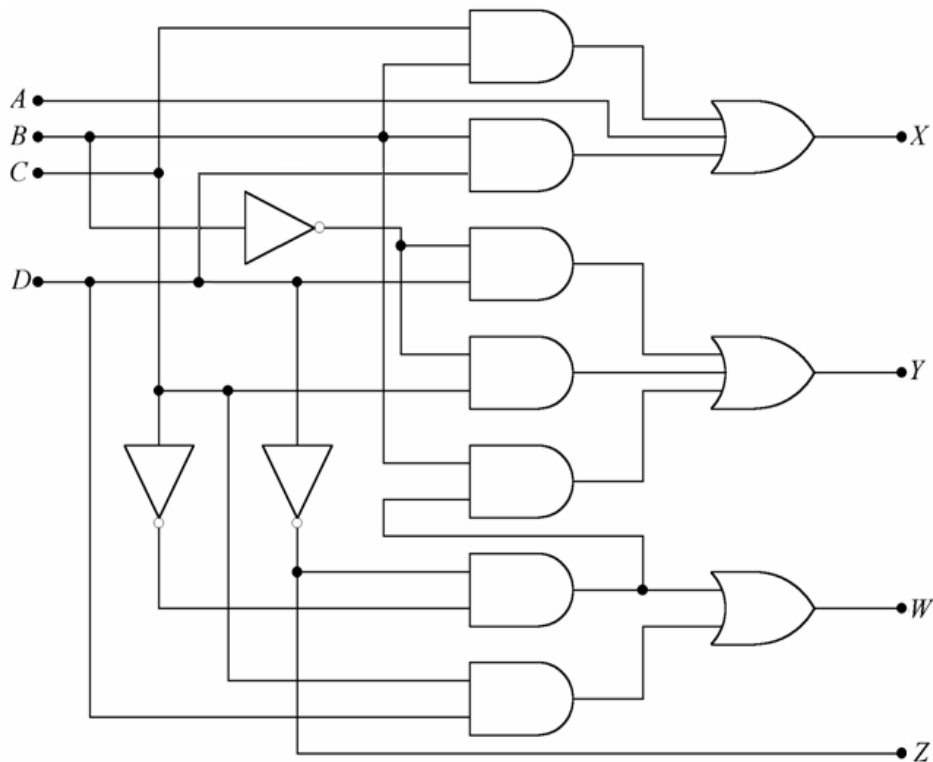
AB \ CD	00	01	11	10
00	1	1	X	1
01	0	0	X	0
11	1	1	X	X
10	0	0	X	X

$$W = \bar{C} \cdot \bar{D} + C \cdot D$$

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

$$Z = \bar{D}$$

Από τους παραπάνω χάρτες προκύπτει το κύκλωμα του σχήματος 4.40.



Σχήμα 4.40. Κύκλωμα μετατροπής του BCD 8421 σε BCD XS3

4.14. Έλεγχος των τιμών δυαδικών ψηφίων

A) Χειρισμός δυαδικών ψηφίων με την πύλη «OR»

Αλλαγή ενός bit (έστω του 4ου) από 0 σε «1». (Κοσσίδης, & Γιαννακόπουλος, 2006)

	0	1	1	0	1	1	1	0
OR	0	0	0	1	0	0	0	0
	0	1	1	1	1	1	1	0

Εκτελούμε την πράξη OR με λέξη που περιέχει στη θέση αυτή «1». Έτσι είμαστε βέβαιοι για την έξοδο του συγκεκριμένου ψηφίου.

B) Χειρισμός δυαδικών ψηφίων με την πύλη «AND»

Αλλαγή ενός bit (έστω του 4ου) από 1 σε «0»

	0	1	1	1	1	1	1	0
AND	1	1	1	0	1	1	1	1
	0	1	1	0	1	1	1	0

Εκτελούμε την πράξη AND με λέξη που περιέχει στη θέση αυτή «0». Έτσι, αντίστοιχα, ήμαστε βέβαιοι για την έξοδο του συγκεκριμένου ψηφίου.

Γ) Χειρισμός των bit με την πύλη «XOR»

Συμπλήρωμα (αντιστροφή) των bits (4 και 0) (μετρώ από δεξιά προς τα αριστερά)

	0	1	1	1	1	1	1	0
XOR	0	0	0	1	0	0	0	1
	0	1	1	0	1	1	1	1

Για να «συμπληρώσουμε» κάποιο δυαδικό ψηφίο, εκτελούμε την πράξη XOR με λέξη που περιέχει στις θέσεις που μας ενδιαφέρει «1». Έτσι είμαστε βέβαιοι ότι το τελικό αποτέλεσμα θα είναι το συμπληρωματικό του αρχικού, για τα συγκεκριμένα δυαδικά ψηφία.

Δ) Έλεγχος ισοδύναμων bytes με την πύλη «XOR»

	0	1	1	0	1	0	1	0
XOR	0	1	1	0	1	0	1	0
	0	0	0	0	0	0	0	0

Για να ελέγξουμε αν κάποια λέξη είναι η ζητούμενη, εκτελούμε την πράξη XOR με την ίδια τη λέξη, που είναι η ζητούμενη. Σε αυτή την περίπτωση, αν η λέξη είναι σωστή, το τελικό αποτέλεσμα θα είναι μηδέν.

Ε) Έλεγχος για μηδενικό Byte

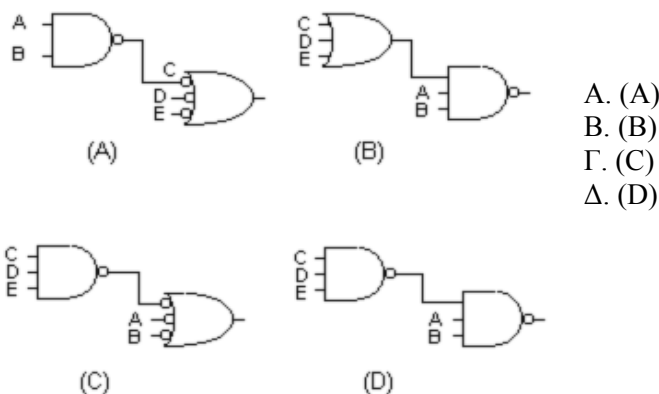
Όμοια, για να ελέγξουμε αν κάποια λέξη είναι μηδενική, εκτελούμε την πράξη OR με το μηδέν. Σε αυτή την περίπτωση το τελικό αποτέλεσμα πρέπει να είναι μηδέν, αν η αρχική λέξη είναι επίσης μηδέν.

	0	0	0	0	0	0	0	0
OR	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

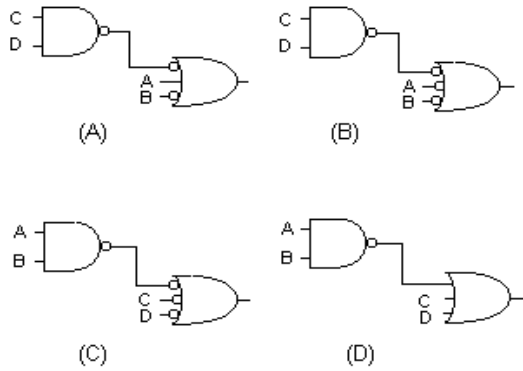
4.15. Ασκήσεις – Ερωτήσεις

ΕΡΩΤΗΣΕΙΣ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

- Για την υλοποίηση της έκφρασης $X = ABC\bar{D} + \bar{A}\bar{B}CD + ABC\bar{D}$ χρειάζεται μια πύλη OR και
 - τρεις πύλες AND και τρεις inverter
 - τρεις πύλες AND και τέσσερις inverter
 - τρεις πύλες AND
 - μια πύλη AND
- Ένα κύκλωμα(AND-OR) 4 μεταβλητών παράγει στην έξοδο $Y=1$. Ποιος συνδυασμός εισόδων είναι ο σωστός;
 - $A = 0, B = 0, C = 0, D = 0$
 - $A = 0, B = 1, C = 1, D = 0$
 - $A = 1, B = 1, C = 0, D = 0$
 - $A = 1, B = 0, C = 0, D = 0$
- Πόσες πύλες AND απαιτούνται για την εφαρμογή της Boolean εξίσωσης $X = ABC\bar{C} + \bar{A}\bar{B}C + \bar{A}BC$;
 - 1
 - 2
 - 3
 - 4
- Πόσες πύλες NOT απαιτούνται για την εφαρμογή της Boolean εξίσωσης $X = \bar{A}\bar{B}C + \bar{A}BC$;
 - 1
 - 2
 - 4
 - 5
- Με πόσες πύλες NAND μπορεί να παραχθεί ένας inverter;
 - 1
 - 2
 - 3
 - 4
- Υλοποιώντας την έκφραση $X = \overline{(A+B)}(CDE)$ και χρησιμοποιώντας λογική NAND, έχουμε:

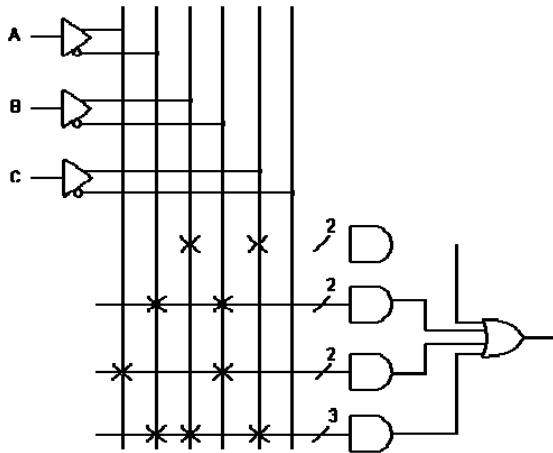


- Μια πιθανή έξοδος για ένα κύκλωμα AND-OR-Invert έχοντας μία πύλη AND με εισόδους A, B και C και μία πύλη AND με εισόδους D και E είναι:
 - $X = \overline{ABC + DE}$
 - $X = \overline{\overline{A} + \overline{B} + \overline{C} + \overline{D} + \overline{E}}$
 - $X = \overline{(A + B + C)}(\overline{D} + \overline{E})$
 - $X = (A + B + C)(D + E)$
- Η εφαρμογή της Boolean εξίσωσης $X = ABC + AB + \bar{A}\bar{C}$ καταλήγει σε:
 - τρεις πύλες AND, μια πύλη OR
 - τρεις πύλες AND, μια πύλη NOT, μια πύλη OR
 - τρεις πύλες AND, μια πύλη NOT, τρεις πύλες OR
 - τρεις πύλες AND, τρεις πύλες OR
- Πόσες πύλες NOR δυο εισόδων χρειάζονται, για να παραχθεί μια πύλη NAND δυο εισόδων;
 - 1
 - 2
 - 3
 - 4
- Πόσες πύλες XNOR απαιτούνται για κύκλωμα συγκριτή των 8-bit;
 - 4
 - 6
 - 8
 - 0
- Υλοποιώντας την έκφραση $X = \bar{A} + \bar{B} + CD$ χρησιμοποιώντας λογική NAND, έχουμε:



- A. (A)
- B. (B)
- Γ. (C)
- Δ. (D)

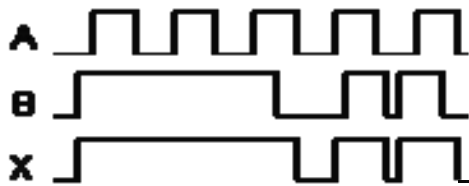
12. Ένα λογικό κύκλωμα με μια έξοδο $X = \overline{ABC} + \overline{AB}$ αποτελείται από:
- A. δυο πύλες AND, δυο πύλες OR, δυο inverter
 - B. τρεις πύλες AND, δυο πύλες OR, έναν inverter
 - Γ. δυο πύλες AND, μια πύλη OR, δυο inverter
 - Δ. δυο πύλες AND, μία πύλη OR
13. Η έξοδος μιας πύλης exclusive-NOR(XNOR) είναι 1. Ποιος από τους παρακάτω συνδυασμούς εισόδων, είναι ο σωστός;
- | | |
|-------------|---------------------------|
| A. A=1, B=0 | B. A=0, B=1 |
| Γ. A=0, B=0 | Δ. Κανένα από τα παραπάνω |
14. Σύμφωνα με το διάγραμμα GAL, η σωστή λογική συνάρτηση είναι:



- A. $X = BC + \overline{AB} + \overline{AB} + \overline{ABC}$
- B. $X = BC + \overline{AB} + \overline{AB} + \overline{ABC}$
- Γ. $X = BC + \overline{AB} + \overline{AB} + \overline{ABC}$
- Δ. $X = BC + \overline{AB} + \overline{AB} + \overline{ABC}$

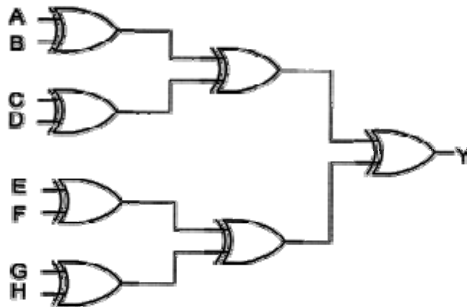
15. Ένα κύκλωμα 4 μεταβλητών AOI(AND-OR-Inverter) δίνει στην έξοδο $Y=0$. Ποιος συνδυασμός εισόδων είναι σωστός;
- | | |
|--|--|
| A. $X = \overline{AB} + \overline{CD}$ | B. $X = \overline{AB} + \overline{CD}$ |
| Γ. $X = \overline{AB} + \overline{CD}$ | Δ. Κανένας από τα παραπάνω |
16. Ποια πύλη μπορεί να χρησιμοποιηθεί σαν βασικός συγκριτής;
- | | | | |
|--------|-------|--------|---------|
| A. NOR | B. OR | Γ. XOR | Δ. NAND |
|--------|-------|--------|---------|
17. Δύο 4-bit δυαδικό αριθμοί (1011 και 1111) εφαρμόζονται σε ένα 4-bit παράλληλο αθροιστή. Η είσοδος κρατουμένου(carry input) είναι 1. Ποιες είναι οι τιμές για την έξοδο του sum και του carry;
- | | |
|---|---|
| A. $\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 0111, C_{out} = 0$ | B. $\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 1111, C_{out} = 1$ |
| Γ. $\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 1011, C_{out} = 1$ | Δ. $\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 1100, C_{out} = 1$ |
18. Αν μια πύλη NOR τριών εισόδων έχει 8 πιθανούς συνδυασμούς για τα δεδομένα στις εισόδους τις, πόσοι από αυτούς τους συνδυασμούς έχουν σαν αποτέλεσμα στην έξοδο της πύλης το λογικό '1';
- | | | | |
|------|------|------|------|
| A. 1 | B. 2 | Γ. 7 | Δ. 8 |
|------|------|------|------|

19. Οι ακόλουθες κυματομορφές αναπαριστούν μια:



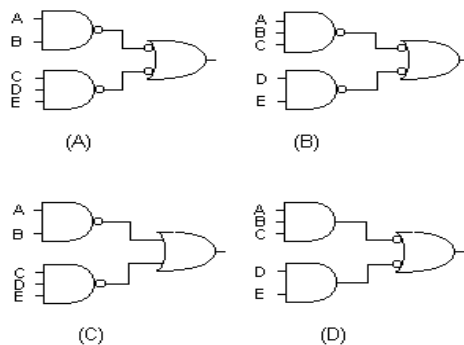
- A. πύλη AND δυο εισόδων
- B. πύλη OR δυο εισόδων
- Γ. πύλη XOR
- Δ. τίποτα από τα παραπάνω

20. Τα παρακάτω κύκλωμα αποτελείται από πύλες XOR 8 εισόδων και έχει έξοδο $Y=1$. Ποιος συνδυασμός εισόδων (με σειρά A-H) είναι σωστός;



- A. 10111100
- B. 10111000
- Γ. 11100111
- Δ. 00011101

21. Υλοποιώντας τη συνάρτηση $AB + CDE$ με τη χρήση λογικής NAND, έχουμε:



- A. (A)
- B. (B)
- Γ. (C)
- Δ. (D)

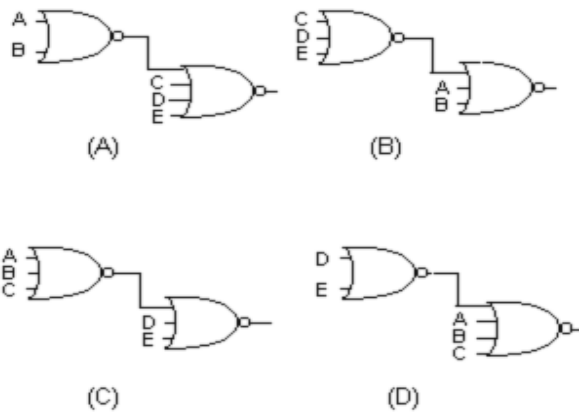
22. Ένας θετικός παλμός με $t_w=75\mu s$ εφαρμόζεται σε μία από τις εισόδους ενός κυκλώματος XOR. Ένας δεύτερος θετικός παλμός $t_w=15\mu s$ εφαρμόζεται στην άλλη είσοδο μετά από $20\mu s$ μετά τη προπορευόμενη ακμή του πρώτου παλμού. Ποια δήλωση περιγράφει την έξοδο σε σχέση με τις εισόδους;

- A. Η έξοδος της XOR είναι ένας παλμός $20\mu s$ ακολουθούμενος από έναν παλμό $40\mu s$, με ένα διαχωρισμό $15\mu s$ μεταξύ των παλμών.
- B. Η έξοδος της XOR είναι ένας παλμός $20\mu s$ ακολουθούμενος από έναν παλμό $15\mu s$, με ένα διαχωρισμό $40\mu s$ μεταξύ των παλμών.
- Γ. Η έξοδος της XOR είναι ένας παλμός $15\mu s$, ακολουθούμενος από έναν παλμό $40\mu s$
- Δ. Η έξοδος της XOR είναι ένας παλμός $20\mu s$ ακολουθούμενος από έναν παλμό $15\mu s$, ο οποίος ακολουθείται από έναν παλμό $40\mu s$.

23. Οι δυαδικοί αριθμοί $A = 1100$ και $B = 1001$ εφαρμόζονται στις εισόδους ενός συγκριτή. Ποια τα επίπεδα εξόδου (G, L, E);

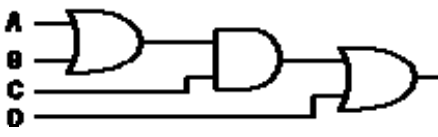
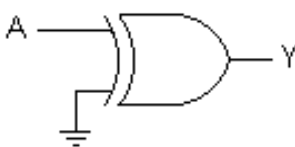
- A. $A > B = 1, A < B = 0, A = B = 1$
- B. $A > B = 0, A < B = 1, A = B = 0$
- Γ. $A > B = 1, A < B = 0, A = B = 0$
- Δ. $A > B = 0, A < B = 1, A = B = 1$

24. Υλοποιώντας την έκφραση $X = \overline{AB} + (C + D + E)$ με λογική NOR, έχουμε:



- A. (A)
- B. (B)
- Γ. (C)
- Δ. (D)

25. Να προσδιορίσετε τις τιμές των A, B, C και D που ικανοποιούν την σχέση: $X = \bar{A} + B + \bar{C} + D = 0$.
- | | |
|--------------------|--------------------|
| A. A=1 B=0 C=0 D=0 | B. A=1 B=0 C=1 D=0 |
| Γ. A=0 B=1 C=0 D=0 | Δ. A=1 B=0 C=1 D=1 |
26. Ένας SIPO 4-bit καταχωρητής ολίσθησης αρχικά περιέχει άσους. Το 0111 είναι σε αναμονή για να εισέλθει. Μετά από τέσσερις παλμούς ρολογιού περιέχει
- | | | | |
|---------|---------|---------|---------|
| A. 0000 | B. 1111 | Γ. 0111 | Δ. 1000 |
|---------|---------|---------|---------|
27. Με συχνότητα 200 KHz, οκτώ bits μπορούν να εγγραφούν σειριακά σε έναν καταχωρητή ολίσθησης σε:
- | | | | |
|--------------|---------------|----------------|----------|
| A. 4 μ s | B. 40 μ s | Γ. 400 μ s | Δ. 40 ms |
|--------------|---------------|----------------|----------|
28. Ένας πλήρης αθροιστής έχει $C_{in} = 0$. Ποιο είναι το άθροισμα sum (Σ) και το κρατούμενο carry (C_{out}) όταν το A = 1 και το B = 1;
- | | |
|------------------------------|------------------------------|
| A. $\Sigma = 0, C_{out} = 0$ | B. $\Sigma = 0, C_{out} = 1$ |
| Γ. $\Sigma = 1, C_{out} = 0$ | Δ. $\Sigma = 1, C_{out} = 1$ |
29. Εάν ένα σήμα διέρχεται μέσω μιας πύλης αναστέλλεται από την αποστολή ενός λογικού '0' σε μια από τις εισόδους της, και η έξοδος είναι λογικό '1' τότε η πύλη είναι:
- | | | | |
|--------|---------|--------|-------|
| A. AND | B. NAND | Γ. NOR | Δ. OR |
|--------|---------|--------|-------|
30. Η έξοδος μιας NOR πύλης είναι λογικό '1' όταν:
- A. Όλοι οι εισοδοί της είναι ίσες με λογικό '1'
 - B. Κάποιες από τις εισόδους της είναι ίσες με λογικό '1'
 - Γ. Κάποιες από τις εισόδους της είναι ίσες με λογικό '0'
 - Δ. Όλοι οι εισοδοί της είναι ίσες με λογικό '0'
31. Αν μια πύλη AND τριών εισόδων έχει 8 πιθανούς συνδυασμούς για τα δεδομένα στις εισόδους της, πόσοι από αυτούς τους συνδυασμούς έχουν σαν αποτέλεσμα στην έξοδο της πύλης το λογικό '1';
- | | | | |
|------|------|------|------|
| A. 1 | B. 2 | Γ. 7 | Δ. 8 |
|------|------|------|------|
32. Η λογική έκφραση μιας πύλης AND τριών εισόδων είναι:
- | | | | |
|-------------|--------------|----------------|---------------|
| A. $X = AB$ | B. $X = ABC$ | Γ. $X = A+B+C$ | Δ. $X = AB+C$ |
|-------------|--------------|----------------|---------------|
33. Οι πύλες NAND και NOR αναφέρονται ως «καθολικές» πύλες, επειδή και οι δύο:
- A. Μπορούν να βρεθούν σε όλα τα ψηφιακά κυκλώματα
 - B. Μπορούν να χρησιμοποιηθούν, ώστε να κατασκευαστούν όλοι οι άλλοι τύποι πυλών
 - Γ. Χρησιμοποιούνται σε όλες τις χώρες του κόσμου
 - Δ. Ήταν οι πρώτες πύλες που υλοποιήθηκαν
34. Μια πύλη NAND με άρνηση στις εισόδους της υλοποιεί την ίδια λειτουργία με μια _____ πύλη.
- | | | | |
|--------|-------|--------|---------|
| A. NOT | B. OR | Γ. NOR | Δ. NAND |
|--------|-------|--------|---------|
35. Μια πύλη NOR 2 εισόδων είναι ισοδύναμη με
- | | |
|-----------------------|---------------------------|
| A. Αρνητική πύλη OR | B. Αρνητική πύλη AND |
| Γ. Αρνητική πύλη NAND | Δ. Τίποτα από τα παραπάνω |

36. Ο όρος «fan-out» αναφέρεται
 Α. στην τάση
 Β. στο ρεύμα
 Γ. στα watts
 Δ. στις μονάδες φορτίου
37. Η λογική έκφραση μιας πύλης NOR είναι:
 Α. $X = A + B$
 Β. $X = \overline{A + B}$
 Γ. $X = A + B$
 Δ. $X = \overline{A} + B$
38. Δεδομένης μιας πύλης AND, ποιό από τα παρακάτω είναι αληθές;
 Α. Μια πύλη AND έχει 2 εισόδους και 1 έξοδο
 Β. Μια πύλη AND έχει 2 ή παραπάνω εισόδους και 2 εξόδους
 Γ. Εάν μια είσοδος μιας πύλης AND 2 εισόδων είναι λογικό «1», τότε η έξοδος αντιστοιχεί στην τιμή της άλλης εισόδου
 Δ. Σε μια πύλη AND 2 εισόδων υπάρχουν 8 πιθανοί συνδυασμοί στις εισόδους της
39. Να προσδιορίσετε την Boolean έκφραση για το λογικό κύκλωμα που ακολουθεί:

 Α. $X = CB + AC + CD$
 Β. $X = CB + AC + D$
 Γ. $X = C(B + A) + D$
 Δ. $X = C(B + A)\overline{D}$
40. Η έξοδος μιας πύλης AND με εισόδους A, B, C είναι το λογικό «1» όταν:
 Α. A=1, B=1, C=0
 Β. A=0, B=0, C=0
 Γ. A=1, B=1, C=1
 Δ. A=1, B=0, C=1
41. Η έξοδος του κυκλώματος που ακολουθεί θα είναι πάντα

 Α. 1
 Β. 0
 Γ. A
 Δ. \overline{A}
42. Η έξοδος μιας πύλης OR με εισόδους τα A, B, C είναι το λογικό «0» όταν:
 Α. A=0, B=0, C=0
 Β. A=0, B=1, C=1
 Γ. A=0, B=0, C=1
 Δ. Όλα τα παραπάνω
43. Εάν ένας 8-bit κυκλικός απαριθμητής έχει σαν αρχική κατάσταση την 10111110. Ποιά είναι η κατάσταση μετά τον τέταρτο παλμό ρολογιού;
 Α. 11101011 Β. 00010111 Γ. 11110000 Δ. 00000000
44. Ένας 4-bit καταχωρητής ολίσθησης που δέχεται 4 bit δεδομένων παράλληλα θα μετατοπιστεί προς τα _____ κατά _____ θέσεις για κάθε παλμό ρολογιού.
 Α. δεξιά, μία Β. δεξιά, δύο
45. Η έξοδος μιας XNOR πύλης είναι λογικό «1» όταν
 Α. οι εισοδοί είναι ίσες
 Β. μία είσοδος είναι λογικό '1' και η άλλη είσοδος είναι λογικό «0»
 Γ. οι εισοδοί είναι άνισοι
 Δ. τίποτα από τα παραπάνω
46. Ποιά από τις παρακάτω εξισώσεις περιγράφει καλύτερα τη λειτουργία μιας πύλης OR 4 εισόδων, όπου A=1, B=1, C=0, D=0
 Α. $1 + 1 + 0 + 0 = 01$
 Β. $1 + 1 + 0 + 0 = 1$
 Γ. $1 + 1 + 0 + 0 = 0$
 Δ. $1 + 1 + 0 + 0 = 00$

Αναφορές-Βιβλιογραφία

- Κοσσίδης, Α.Θ. (1996). *Σχεδίαση Ψηφιακών Κυκλωμάτων*, Εκδόσεις Μπένος
- Κοσσίδης, Α.Θ., Γιαννακόπουλος, Π. (2006). *Αριθμητικά Συστήματα και Ψηφιακά Κυκλώματα*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα
- Balabanian, N., Carlson, B. (2007). *Digital Logic Design Principles*, John Wiley
- Balch, M. (2003). *Complete Digital Design*, Mc Graw Hill
- Butzen, P.F., Dal Bem, V., Reis, A. I., Ribas, R. P. (2010). *Transistor network restructuring against NBTI degradation*, *Microelectronics Reliability* 50(9):1298-1303. DOI: 10.1016/j.microrel.2010.07.140
- Floyd, Thomas L., (2006). *Digital Fundamentals*, 9th, Pearson International Edition
- Givone, D. (2002). *Digital Principles and Design*,. Mc Graw Hill
- Godse, A.P., Godse, D.A. (2010). *Digital Logic Design and Application*, Technical Publications Pune
- Hendrich, Norman (2006). *HADES Tutorial*, University of Hamburg, v.0.92.
- Holdsworth, Brian, Woods, Clive (2002). *Digital Logic Design*, 4th Edition, Newnes
- Katz, R. (2005). *Contemporary Logic Design*, 2/e, Prentice Hall
- Mano, M., Ciletti, M. (2014). *Ψηφιακή Σχεδίαση*, 5^η έκδοση, Παπασωτηρίου
- Maxfield, Clive, (2008). *Bebop to the Boolean Boogie, An Unconventional Guide to Electronics*, 3rd , Newnes
- Nashelsky, Louis (1994). *Introduction to Digital Technology*, 4th Ed., Prentice Hall
- Nelson, V., Nagle, H., Carroll, B., Irwin, J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall
- Predko, Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill
- Pritchard, N. (2015). *Fundamentals of Digital Electronics*, CreateSpace Independent Publishing Platform
- Roth, Charles, (2006). *Fundamentals of Logic Design*, Thomson-Nelson
- Tocci, R. J., Widmer, N. S, and Moss, Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley
- Wakerly, J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Κεφάλαιο 5

Σύνοψη

Στο κεφάλαιο αυτό θα παρουσιαστεί ο σχεδιασμός των βασικών λογικών πυλών με διακριτά στοιχεία. Κατόπιν ακολουθούν πρακτικά κυκλώματα των λογικών πυλών NAND και NOR με τη χρήση διακριτών στοιχείων, η πύλη αποκλειστικής διάζευξης XOR, και η υλοποίηση των βασικών πυλών με DTL. Τα ολοκληρωμένα κυκλώματα βασικών λογικών κυκλωμάτων και ο σχεδιασμός τους με τεχνολογίες TTL, ECL, CMOS. Τέλος γίνεται αναφορά στην υλοποίηση των πυλών με τη νέα τεχνολογία των καρβονικών νανοσωληνών (CNTFET) και των memristors. Ακολουθούν οι κωδικοποιητές- αποκωδικοποιητές με διακριτά στοιχεία, και το κεφάλαιο ολοκληρώνεται με την παρουσίαση των πολυδονητών που είναι το βασικό στοιχείο για τη μελέτη των f-f.

Προαπαιτούμενη γνώση

Βασικές γνώσεις από τα βασικά ηλεκτρονικά, όπως η λειτουργία των διόδων, η γνώση των διπολικών τρανζίστορ και των τρανζίστορ επίδρασης πεδίου.

5. Κατασκευή πυλών με διακριτά στοιχεία

5.1. Εισαγωγή

Όλα τα λογικά κυκλώματα που εξετάσαμε μέχρι τώρα μελετήθηκαν με τη βοήθεια διακοπών, και χρησίμευσαν για την καλύτερη εμπέδωση της Άλγεβρας Boole. Τα σύγχρονα υπολογιστικά συστήματα κατασκευάζονται με ηλεκτρονικά λογικά κυκλώματα και διαθέτουν πολλά πλεονεκτήματα ως προς την ταχύτητα λειτουργίας, την αξιοπιστία και γενικότερα την απόδοσή τους. Τα ηλεκτρονικά κυκλώματα, που χρησιμοποιούνται για τη μελέτη και κατασκευή των ηλεκτρονικών υπολογιστικών συστημάτων, είναι διατάξεις που αποτελούνται κυρίως από ηλεκτρονικά εξαρτήματα ημιαγωγών, όπως κοινές διόδους, διόδους Zener, διπολικά τρανζίστορ (BJT), τρανζίστορ επίδρασης πεδίου (FET-Field Effect Transistor), ολοκληρωμένα κυκλώματα (IC's-Integrated Circuits) και τέλος την καινούργια τεχνολογία των memristors και των καρβονικών νανοσωληνών.

Σήμερα τα υπολογιστικά συστήματα κατασκευάζονται με ολοκληρωμένα κυκλώματα που αποτελούν την τελευταία λέξη της τεχνολογίας. Χαρακτηριστικό των κυκλωμάτων αυτών είναι η μεγάλη ταχύτητα λειτουργίας και ο μικρός όγκος των συστημάτων που αναπτύσσονται με αυτά. Η συνεχιζόμενη έρευνα στον τομέα των ηλεκτρονικών βελτιώνει συνεχώς τις επιδόσεις τόσο των εξαρτημάτων όσο και των υπολογιστικών συστημάτων γενικότερα.

5.2. Πρακτικά κυκλώματα πυλών OR και AND με τη χρήση διακριτών στοιχείων

Θα μελετήσουμε την κατασκευή πρακτικών κυκλωμάτων OR και AND με ηλεκτρονικά εξαρτήματα. Τόσο για τα κυκλώματα OR και AND, όσο και για όλα τα άλλα που θα εξετάσουμε στη συνέχεια, θα ισχύουν οι ίδιοι κανόνες ως προς τη χρησιμοποιούμενη λογική. Οι χρησιμοποιούμενες λογικές είναι δύο:

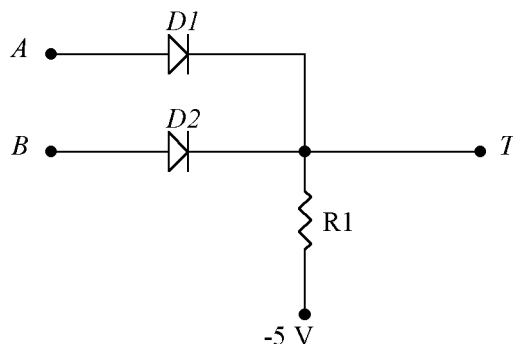
α) η **θετική λογική** και β) η **αρνητική λογική**. (Nashelsky, 1994; Δεληγιάννης, 2004; Katz, 2005; Wakerly, 2006; Κοσσίδης, & Γιαννακόπουλος 2006; Tocci, et al. 2010; Jaeger, & Blalock, 2011; Sedra, & Smith, 2015).

Κατά τη θετική λογική, αντιστοιχούμε στην τάση των 0 Volt τη λογική κατάσταση «0» και στην τάση των +V Volt, τη λογική κατάσταση «1». Δηλαδή κατά τη θετική λογική, η λογική κατάσταση «1» έχει μεγαλύτερο δυναμικό ως προς τη λογική κατάσταση «0». Αντίστροφα, κατά την αρνητική λογική, αντιστοιχούμε στη λογική κατάσταση «1» το δυναμικό 0 Volt και στη λογική κατάσταση «0» το δυναμικό +V Volt. Δηλαδή η τάση του λογικού «1» είναι μικρότερη της τάσης του λογικού «0».

Η ίδια αρχή ισχύει και για την περίπτωση όπου η τάση V είναι αρνητική. Η τάση 0 Volt αντιστοιχεί σε λογικό «1» και η τάση -V Volt σε λογικό «0» κατά τη θετική λογική, ενώ ισχύει το αντίστροφο για την περίπτωση της αρνητικής λογικής. Παρακάτω θα αναφέρουμε ορισμένα παραδείγματα κυκλωμάτων OR και AND.

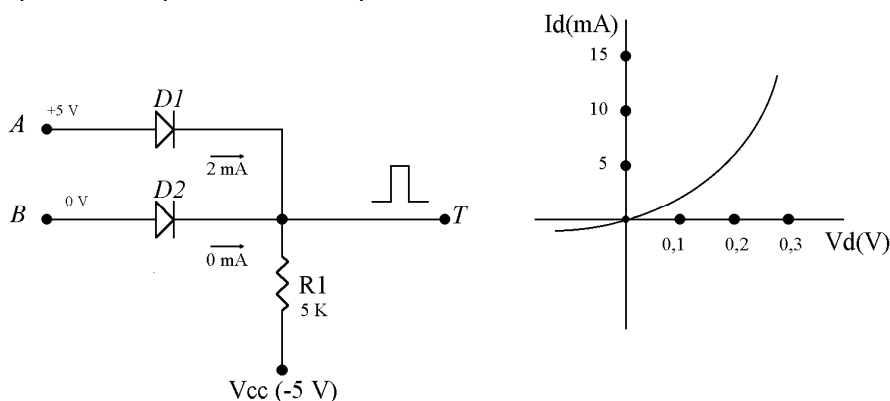
Στο Σχήμα 5-1 δίνεται πύλη OR κατασκευασμένη από ημιαγωγικές διόδους. Οι είσοδοι A, B μπορούν να παίρνουν την τιμή «1» ή «0» ανάλογα με τις εφαρμοζόμενες τάσεις. Χρησιμοποιώντας θετική λογική, το κύκλωμα εργάζεται ως εξής: Έστω ότι A=B=0 (V= 0 Volt), τότε οι διόδοι θα πολωθούν κατά την ορθή φορά με αποτέλεσμα, λόγω της μικρής εσωτερικής αντίστασης που παρουσιάζουν, να μεταφέρουν την τάση εισό-

δου από τα σημεία A, B στην έξοδο, δηλαδή T=0. Για A=1 (V=+5 Volt) B=0 (V=0 Volt), ή και το αντίστροφο, η τάση στην έξοδο θα είναι +5 Volt, γιατί και πάλι οι δύο διόδους θα άγουν κατά τέτοιον τρόπο, ώστε να μεταφέρουν την τιμή «1» στην έξοδο. Δε χρειάζεται φυσικά να αναφέρουμε πως ίδια τάση θα έχουμε στην έξοδο, ακόμα και όταν οι δύο εισόδοι έχουν ταυτόχρονα V=+5 Volt (A=B=1).



Σχήμα 5.1. Πύλη OR με διόδους

Στο σχήμα 5.2 μαζί με το παραπάνω ψηφιακό κύκλωμα βλέπουμε και τη γραφική παράσταση που περιγράφει τη σχέση μεταξύ της τάσης που επιβάλλεται στα άκρα της διόδου και του ρεύματος που τη διαρρέει. Με τις παρούσες τιμές των εξαρτημάτων, το ρεύμα που θα διέρχεται μέσα από την ορθά πολωμένη διόδο αλλά και μέσα από την αντίσταση R₁ θα είναι ίσο με 2 mA.



Σχήμα 5.2. Υπολογισμός ρευμάτων της πύλης OR

Ο πίνακας 5.1, που περιγράφει τη λειτουργία του κυκλώματος, παρατίθεται κατωτέρω και προκύπτει η λογική συνάρτηση: $T = \bar{A} \cdot B + A \cdot \bar{B} + A \cdot B = B + A \cdot \bar{B} = A + B$

Από τα παραπάνω προκύπτει ότι, χρησιμοποιώντας θετική λογική, το κύκλωμα συμπεριφέρεται σαν πύλη OR.

A	B	T
0 (+0 Volt)	0 (+0 Volt)	0 (+0 Volt)
0 (+0 Volt)	1 (+5 Volt)	1 (+5 Volt)
1 (+5 Volt)	0 (+0 Volt)	1 (+5 Volt)
1 (+5 Volt)	1 (+5 Volt)	1 (+5 Volt)

Πίνακας 5.1. Πίνακας αλήθειας του σχήματος 5.2 για θετική λογική

Αντίστροφα, εάν χρησιμοποιούσαμε αρνητική λογική, η λογική συνάρτηση στην περίπτωση αυτή θα ήταν η: $T = A \cdot B$. Από τα παραπάνω προκύπτει ότι, χρησιμοποιώντας αρνητική λογική, το κύκλωμα συμπεριφέρεται σαν πύλη AND. Ο πίνακας αλήθειας είναι ο πίνακας 5.2:

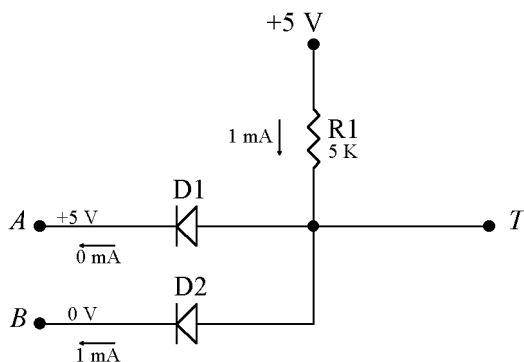
Στα κυκλώματα που αναφέραμε παραπάνω περιοριστήκαμε στη χρήση πυλών με το μικρότερο δυνατό αριθμό εισόδων. Θα μπορούσαμε, βεβαίως, να έχουμε χρησιμοποιήσει πύλες με περισσότερες εισόδους. Ένας εύκολος τρόπος, να επαυξήσουμε τις εισόδους των κυκλωμάτων αυτών, προϋποθέτει την προσθήκη πε-

ρισσότερων διόδων. Εφόσον οι διόδοι αυτές διεγείρονται με τις στάθμες, που ήδη αναφέραμε στο παράδειγμά μας, η λειτουργία τους θα είναι η ίδια.

A	B	T
0 (+5 Volt)	0 (+5 Volt)	0 (+5 Volt)
0 (+5 Volt)	1 (+0 Volt)	0 (+5 Volt)
1 (+0 Volt)	0 (+5 Volt)	0 (+5 Volt)
1 (+0 Volt)	1 (+0 Volt)	1 (+0 Volt)

Πίνακας 5.2. Πίνακας αλήθειας του σχήματος 5.1 για αρνητική λογική

Στο σχήμα 5.3 δίνεται πύλη AND δύο εισόδων, θετικής λογικής, κατασκευασμένη με τη βοήθεια δύο διόδων. Το κύκλωμα εργάζεται ως εξής: Εάν εφαρμόσουμε στις εισόδους A και B τις τιμές, που φαίνονται στο σχήμα 5.3, η διάδος D1 δεν άγει, σε αντίθεση με την D2 η οποία, επειδή είναι ορθά πολωμένη, μεταφέρει το δυναμικό του σημείου B στην έξοδο T. Αυτό σημαίνει ότι T=0. Σημειώστε ότι το ρεύμα που διαρρέει την αντίσταση των 5 KΩ, λόγω της αγωγιμότητας της διόδου D2, θα είναι ίσο με 1 mA περίπου.



Σχήμα 5.3. Πύλη AND με διόδους

Για τους υπόλοιπους συνδυασμούς τιμών στις εισόδους A και B θα έχουμε τον παρακάτω πίνακα 5.3.

A	B	T
0 (+0 Volt)	0 (+0 Volt)	0 (+0 Volt)
0 (+0 Volt)	1 (+5 Volt)	0 (+0 Volt)
1 (+5 Volt)	0 (+0 Volt)	0 (+0 Volt)
1 (+5 Volt)	1 (+5 Volt)	1 (+5 Volt)

Πίνακας 5.3. Πίνακας αλήθειας του σχήματος 5.3 για θετική λογική

Από αυτόν προκύπτει η συνάρτηση του λογικού AND δηλαδή : $T = AB$

Εάν χρησιμοποιήσουμε αρνητική λογική, το κύκλωμα του σχήματος 5.3 εργάζεται σαν πύλη OR. Η συμπεριφορά του τότε περιγράφεται αναλυτικά από τον παρακάτω πίνακα 5.4, στον οποίο συσχετίζονται πάλι οι τάσεις εισόδου και εξόδου του κυκλώματος.

Από τον πίνακα 5.4 προκύπτει η συνάρτηση: $T = A \cdot \bar{B} + \bar{A} \cdot B + A \cdot B = B + A \cdot \bar{B} = A + B$

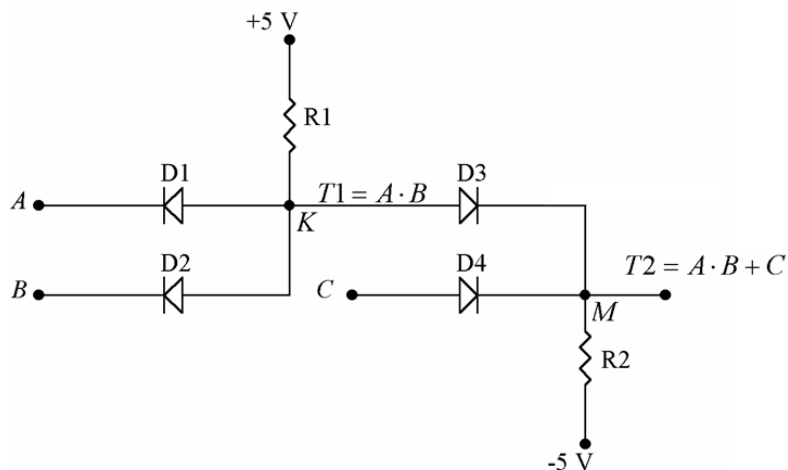
A	B	T
0 (+5 Volt)	0 (+5 Volt)	0 (+5 Volt)
0 (+5 Volt)	1 (+0 Volt)	1 (+0 Volt)
1 (+0 Volt)	0 (+5 Volt)	1 (+0 Volt)
1 (+0 Volt)	1 (+0 Volt)	1 (+0 Volt)

Πίνακας 5.4. Πίνακας αλήθειας του σχήματος 5.3 (για αρνητική λογική)

Κατά τη δημιουργία των λογικών κυκλωμάτων πολλές φορές απαιτούνται περισσότερα του ενός λογικά κυκλώματα, τα οποία είναι κατά κανόνα συνδεδεμένα το ένα μετά το άλλο σε διαδοχή. Στη συνέχεια θα δείξουμε ότι δεν είναι δυνατόν να χρησιμοποιήσουμε περισσότερα των δύο κυκλωμάτων συνδεδεμένα σε σειρά.

Στο σχήμα 5.4 δίνονται δύο πύλες θετικής λογικής: μία AND και μία OR. Η έξοδος της πύλης AND (σημείο K) οδηγείται σε μία από τις δύο εισόδους της πύλης OR. Η είσοδος C της πύλης OR είναι διαθέσιμη προς χρήση από οποιοδήποτε άλλο κύκλωμα.

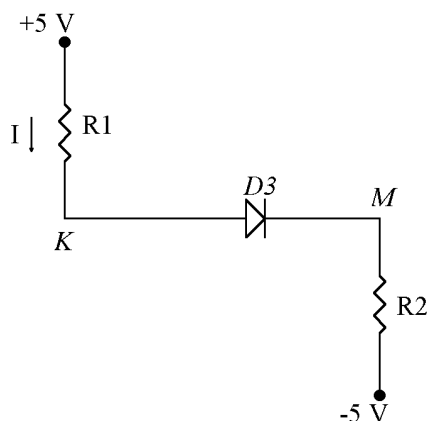
Όταν οι εισόδους A, B έχουν τιμή «0», η έξοδος T₁ έχει επίσης «0», οπότε η κατάσταση της εξόδου T₂ θα εξαρτάται τόσο από την τιμή της C, όσο και από αυτή της T₁. Σ' αυτήν την περίπτωση, οι διόδους D₁, D₂ άγουν κανονικά. Εάν τώρα στις εισόδους A και B επιβάλουμε την τιμή «1», η έξοδος T₁ θα ισούται με «1» δηλαδή θα ισχύει ότι V_K = +5V. Αυτό όμως δε συμβαίνει, διότι η τάση στο σημείο K εξαρτάται και από την τιμή της R₂.



Σχήμα 5.4. Πύλες AND, OR σε σειρά

Κάτω από αυτές τις προϋποθέσεις, το υπό εξέταση κύκλωμα θα είναι αυτό που παρουσιάζεται στο ισοδύναμο του σχήματος 5.5. Η τάση V_K θα ισούται, επομένως, με:

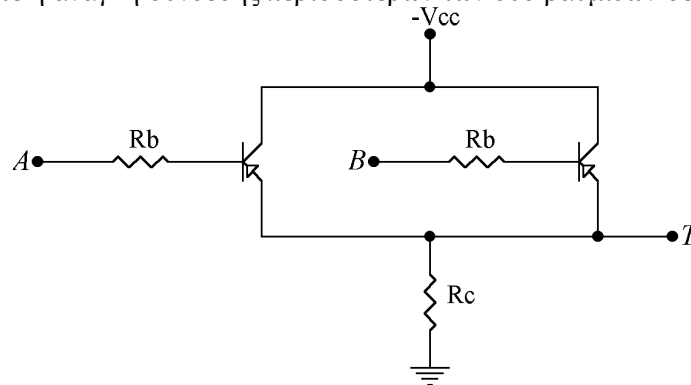
$$V_K = I \cdot R_2 - 5 = \frac{10}{R_1 + R_2} \cdot R_2 - 5 \quad (5-1)$$



Σχήμα 5.5. Κύκλωμα για τον υπολογισμό των επί μέρους τάσεων

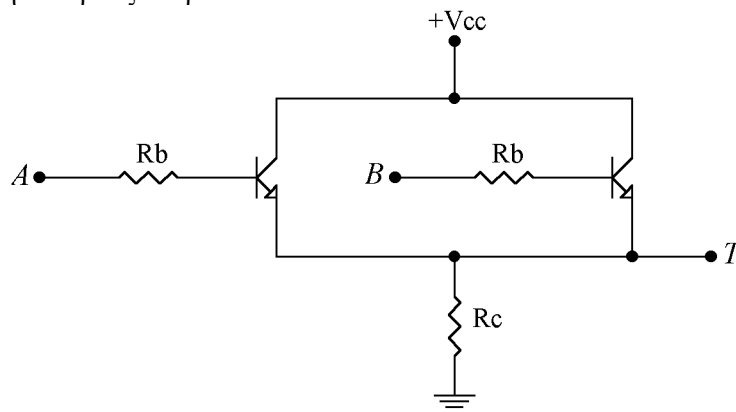
Από τον παραπάνω τύπο φαίνεται ότι, εάν θέλουμε η V_K να έχει τιμή της τάξης των +5V, θα πρέπει να επιλέξουμε την R₂ ώστε να είναι κατά πολύ μεγαλύτερη της R₁. Η αντίσταση της διόδου D₃ θεωρείται πολύ μικρή ως προς τις R₁, R₂, οπότε η επίδρασή της είναι αμελητέα. Η τιμή της R₂ δε μπορεί να γίνει απεριόριστα μεγάλη, γιατί επιδρά στη συχνότητα λειτουργίας του κυκλώματος. Εάν τοποθετούσαμε μία τρίτη βαθμίδα σε διαδοχή, θα έπρεπε η R₃ >> R₂ κ.ο.κ.

Δεδομένου ότι κάθε κύκλωμα AND ή OR μπορεί να έχει πολύ περισσότερες από δύο εισόδους, δεν παρουσιάζεται σχεδόν ποτέ η ανάγκη σύνδεσης περισσότερων των δύο βαθμίδων σε διαδοχή.



Σχήμα 5.6. Πύλη AND με διπολικά τρανζίστορ

Στα σχήματα 5.6 και 5.7 δίνονται αντίστοιχα πύλες AND και OR θετικής λογικής κατασκευασμένες με τρανζίστορ. Στο σχήμα 5.6 τα χρησιμοποιούμενα τρανζίστορ είναι τύπου PNP, σε αντίθεση με το σχήμα 5.7 όπου χρησιμοποιήθηκαν τρανζίστορ τύπου NPN.



Σχήμα 5.7. Πύλη OR με διπολικά τρανζίστορ NPN

Η λειτουργία των κυκλωμάτων είναι απλή και στηρίζεται στην αγωγιμότητα των τρανζίστορ.

Προτείνεται στον αναγνώστη να κατασκευάσει τους πίνακες αλήθειας των παραπάνω πυλών. Αξίζει να παρατηρήσετε ότι, χρησιμοποιώντας τρανζίστορ PNP ή NPN, οι αντίστοιχες πύλες λογικής είναι συμπληρωματικές.

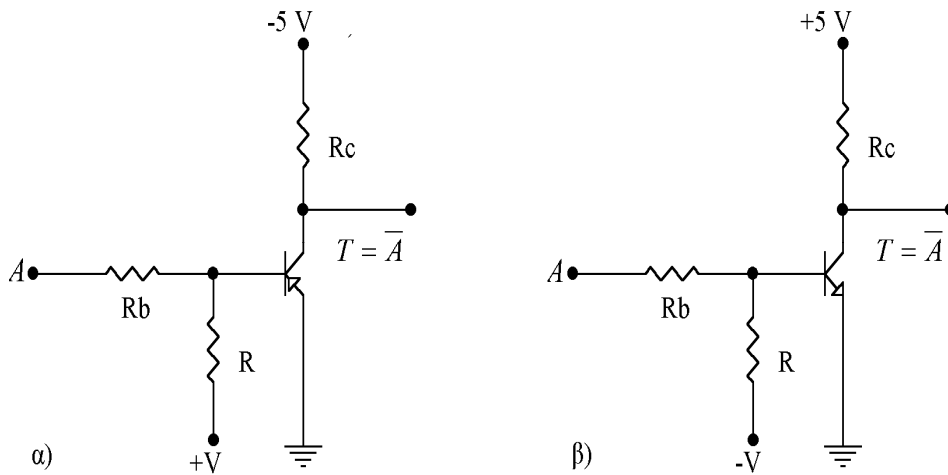
5.3. Πρακτικά κυκλώματα πυλών NOT

Τα σχήματα 5.8 α, β παριστάνουν κυκλώματα πυλών NOT κατασκευασμένων με τρανζίστορ PNP και NPN. Τα δύο κυκλώματα είναι όμοια, αλλά διαφέρουν ως προς την πολικότητα των τάσεων που τροφοδοτούνται.

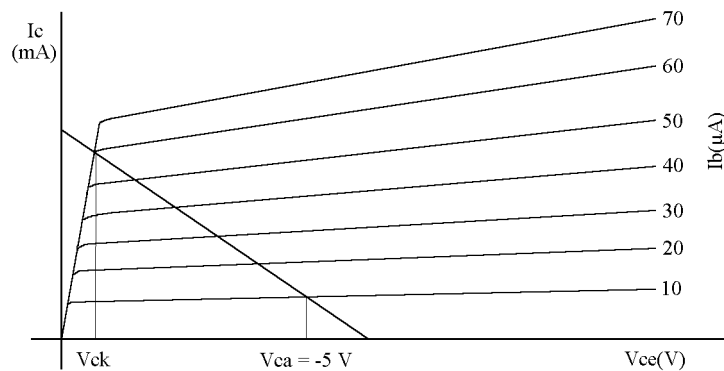
Η λειτουργία του τρανζίστορ σαν πύλη NOT στηρίζεται στη μεταγωγή του από την κατάσταση αποκοπής στην κατάσταση αγωγιμότητας, καθώς το σήμα εισόδου που το διεγείρει αλλάζει λογική στάθμη. Οι αντιστάσεις στη βάση του τρανζίστορ έχουν σαν σκοπό να μετατρέπουν την τάση εισόδου σε ρεύμα βάσης, ενώ η αντίσταση του συλλέκτη του να μετατρέπει το ρεύμα συλλέκτη σε τάση εξόδου.

Χρησιμοποιώντας τις χαρακτηριστικές του τρανζίστορ PNP (Σχήμα 5.9), που είναι σε διάταξη κοινού εκπομπού, μπορούμε εύκολα να καθορίσουμε τις δύο παραπάνω καταστάσεις. Αυτό γίνεται εύκολα, χαράσσοντας την ευθεία φόρτου του κυκλώματος του σχήματος 5.8α (ανάλογα ισχύουν και για το σχήμα 5.8β).

Οι τάσεις V_{ck} και V_{ca} αναπτύσσονται μεταξύ συλλέκτη και γης και αντιπροσωπεύουν τις καταστάσεις λειτουργίας του τρανζίστορ σε κόρο και σε αποκοπή. Όπως φαίνεται από τις χαρακτηριστικές, η τάση κόρου είναι της τάξης των δεκάτων του Volt, σε αντίθεση με την τάση αποκοπής η οποία προσεγγίζει την τάση τροφοδοσίας. (Nashelsky, 1994; Κοσσιδάς 1996; Δεληγιάννης, 2004; Katz, 2005; Wakerly, 2006; Tocci, et al. 2010; Mano, & Ciletti, 2014).



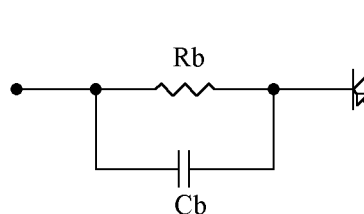
Σχήμα 5.8. Πύλη NOT με PNP (α) και NPN (β)



Σχήμα 5.9. Σημεία λειτουργίας πύλης NOT (σχήματος 5.8α)

Κατά συνέπεια, οι δύο τιμές V_{ck} και V_{ca} απέχουν αισθητά μεταξύ τους, οπότε δεν υπάρχει ποτέ περίπτωση σύγχυσης. Κατά τη λειτουργία ενός τρανζίστορ κάτω από αυτές τις συνθήκες, ο χρόνος εναλλαγής από τη μία κατάσταση στην άλλη είναι πολύ μικρός, ώστε να μπορούμε να πούμε ότι το τρανζίστορ βρίσκεται σε μια από τις δύο καταστάσεις, ανάλογα με το σήμα που εφαρμόζεται στην είσοδο.

Για να ελαττώσουμε ακόμα περισσότερο το χρόνο εναλλαγής, τοποθετούμε παράλληλα με την R_b μια χωρητικότητα τιμής C_b (Σχήμα 5.10) έτσι ώστε αυτή να παρέχει το επιπλέον ρεύμα βάσης τις στιγμές των εναλλαγών. Το μειονέκτημα αυτής της μεθόδου είναι ότι έχουμε σαν συνέπεια μία μικρή αύξηση της τάσης θορύβου στο κύκλωμα, η οποία όμως μπορεί να θεωρηθεί αμελητέα σε σχέση με την αύξηση της ταχύτητας που πετυχαίνουμε.



Σχήμα 5.10. Κύκλωμα ελάττωσης του χρόνου εναλλαγής

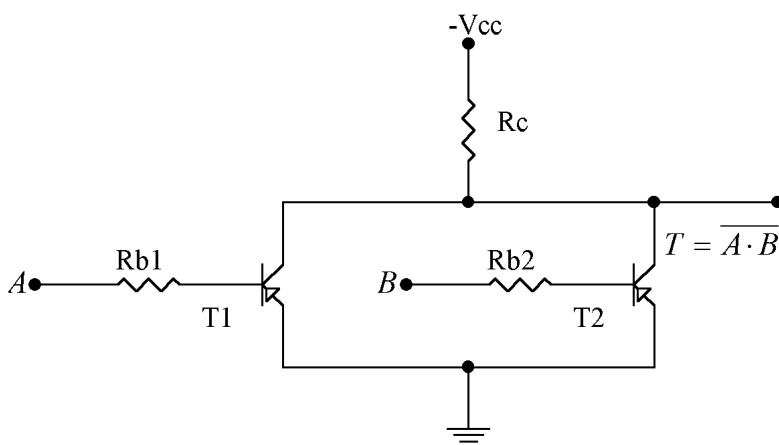
A (Είσοδος)	T (Εξοδος)
0 (-5 Volt)	1 (+0 Volt)
1 (+0 Volt)	0 (-5 Volt)

Πίνακας 5.5. Πύλη NOT θετικής λογικής

Στον πίνακα 5.5 δίνεται ο πίνακας αλήθειας της πύλης NOT θετικής λογικής του σχήματος 5.8α. Η τιμή της αντίστασης R_b είναι τέτοια, ώστε το αντίστοιχο ρεύμα εισόδου να ελέγχει εύκολα το τρανζίστορ, ακόμα και με μικρές τιμές της τάσης εισόδου.

5.4. Πρακτικά κυκλώματα πυλών NAND, NOR

Στο Σχήμα 5.11α δίνεται πύλη NAND με τρανζίστορ PNP θετικής λογικής, τεχνολογίας RTL (Resistor Transistor Logic). Το κύκλωμα αυτό αποτελείται από δύο κυκλώματα NOT, συνδεδεμένα μεταξύ τους με τέτοιο τρόπο, ώστε να έχουν κοινή αντίσταση φορτίου. Πράγματι, όταν το δυναμικό που εφαρμόζεται στις βάσεις των τρανζίστορ είναι θετικό, τότε και τα δύο τρανζίστορ είναι σε αποκοπή και η τιμή της τάσης εξόδου είναι ίση με την τιμή της τάσης τροφοδοσίας. (Nelson, 1995; Κοσσιδάς 1996; Δεληγιάννης, 2004; Wakerly, 2006; Tocci, et al., 2010; Jaeger, & Blalock, 2011; Sedra, & Smith, 2015).



Σχήμα 5.11α. Πύλη NAND με τρανζίστορ τύπου PNP

Ακολουθεί ο πίνακας αλήθειας του κυκλώματος (πίνακας 5.6):

A	B	T
0 (-V Volt)	0 (-V Volt)	1 (+0 Volt)
0 (-V Volt)	1 (+0 Volt)	1 (+0 Volt)
1 (+0 Volt)	0 (-V Volt)	1 (+0 Volt)
1 (+0 Volt)	1 (+0 Volt)	0 (-V Volt)

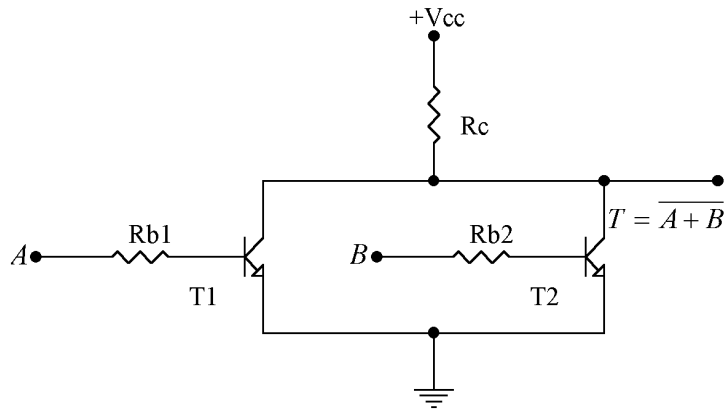
Πίνακας 5.6. Πίνακας αλήθειας πύλης NAND θετικής λογικής

Η λογική συνάρτηση που προκύπτει είναι η εξής:

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

Εάν συνδέσουμε με τον ίδιο τρόπο περισσότερα από δύο κυκλώματα NOT, η αντίστοιχη συνάρτηση εξόδου θα έχει τη μορφή: $T = \overline{A + B + C + \dots + N}$

Στο Σχήμα 5.11β δίνεται η πύλη NOR θετικής λογικής με τρανζίστορ τύπου NPN. Η λειτουργία του κυκλώματος είναι παρόμοια με αυτήν του Σχήματος 5.11α. Άξιο παρατήρησης είναι ότι, αλλάζοντας τον τύπο του τρανζίστορ από PNP σε NPN και διατηρώντας τον ίδιο τύπο λογικής (π.χ. θετικής), επιτυγχάνουμε πάντοτε συζυγείς πύλες. Ένα τέτοιο παράδειγμα φαίνεται στα σχήματα 5.11α και 5.11β.



Σχήμα 5.11β: Πύλη NOR με NPN τρανζίστορ

Στο σχήμα 5.12 δίνεται το κύκλωμα μιας πύλης NAND, κατασκευασμένης με τη βοήθεια δύο τρανζίστορ σε σειρά. Η χρησιμοποιούμενη λογική είναι η θετική και ο τύπος των τρανζίστορ ο NPN.

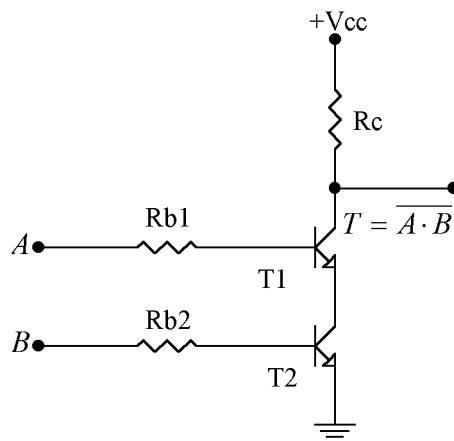
Το κύκλωμα εργάζεται σαν δύο διακόπτες συνδεσμολογημένοι σε σειρά. Εάν το ένα από τα δύο τρανζίστορ είναι σε διακοπή, το κύκλωμα δε λειτουργεί και, κατά συνέπεια, η τάση εξόδου θα ισούται με την τάση V_{cc} της πηγής τροφοδοσίας, δηλαδή $T=1$.

Μόνο εάν και οι δύο είσοδοι του κυκλώματος έχουν ταυτόχρονα θετικό δυναμικό, δηλαδή και τα δύο τρανζίστορ βρίσκονται σε αγωγιμότητα, θα διέρχεται ρεύμα από τον θετικό πόλο της πηγής V_{cc} προς τη γη. Τότε η τάση της εξόδου θα είναι πολύ μικρή, οπότε μπορούμε να υποθέσουμε ότι $T=0$. Στον πίνακα 5.7, περιγράφεται με ακρίβεια η ηλεκτρική συμπεριφορά του κυκλώματος.

A	B	T
0 (+0 Volt)	0 (+0 Volt)	1 (+V Volt)
0 (+0 Volt)	1 (+V Volt)	1 (+V Volt)
1 (+V Volt)	0 (+0 Volt)	1 (+V Volt)
1 (+V Volt)	1 (+V Volt)	0 (+0 Volt)

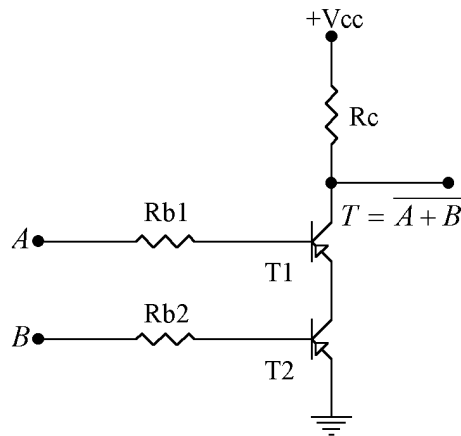
Πίνακας 5.7 Πίνακας αλήθειας πύλης NAND θετικής λογικής

Η λογική συνάρτηση που προκύπτει είναι η: $T = \bar{A} \cdot \bar{B} + \bar{A} \cdot B + A \cdot \bar{B} = \bar{A} + A \cdot \bar{B} = \bar{A} + \bar{B} = \overline{A \cdot B}$



Σχήμα 5.12. Πύλη NAND με τρανζίστορ NPN

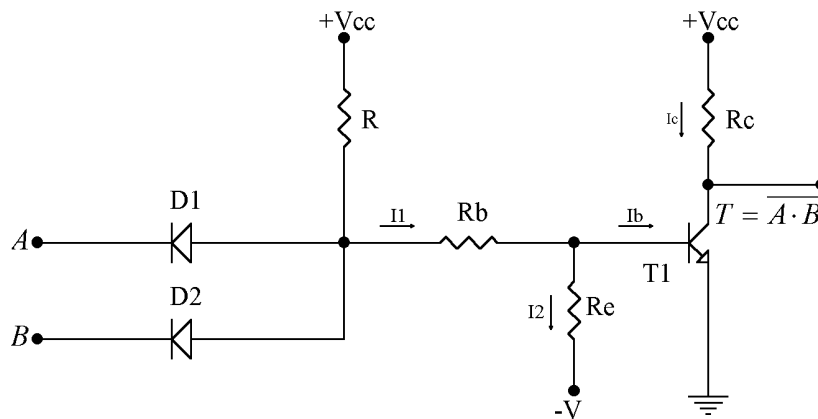
Στο σχήμα 5.13 δίνεται πύλη NOR θετικής λογικής, συζυγής της προηγούμενης του σχήματος 5.12, υλοποιημένη με τρανζίστορ PNP.



Σχήμα 5.13. Πύλη NOR

5.4.1. Πύλες NOR και NAND με διόδους και τρανζίστορ DTL

Οι πύλες στις οποίες θα αναφερθούμε παρακάτω εργάζονται σε σχετικά χαμηλές συχνότητες. Το κόστος κατασκευής τους είναι μικρότερο, γιατί χρησιμοποιούν ένα μόνο τρανζίστορ. Η αρχή λειτουργίας των πυλών αυτών στηρίζεται στην προσθήκη μιας πύλης NOT αμέσως μετά από πύλες AND ή OR, κατασκευασμένες με διόδους. Στο Σχήμα 5.14, δίνεται το κύκλωμα πύλης NAND θετικής λογικής. (Κοσσιδάς, 1996; Δεληγιάννης, 2004)



Σχήμα 5.14. Πύλη NAND με διόδους και τρανζίστορ

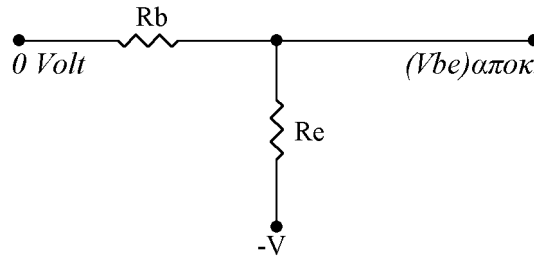
Το πρώτο μέρος του κυκλώματος είναι μία πύλη AND σαν αυτή του σχήματος 5.3, ενώ το δεύτερο μέρος είναι πύλη άρνησης NOT θετικής λογικής σαν αυτή του Σχήμα 5.8β. Ο αριθμός των εισόδων μπορεί να αυξηθεί ανάλογα με τις ανάγκες. Το κύκλωμα εργάζεται σε δύο καταστάσεις: αποκοπή και λειτουργία σε κόρο. Στην κατάσταση κόρου η έξοδος της πύλης AND είναι το λογικό «1». Το ρεύμα βάσης I_b του τρανζίστορ ισούται με:

$$I_b = I_1 - I_2$$

$$I_1 = \frac{V - (V_{be})_{αγωγ.}}{R_b + R} \quad I_2 = \frac{V + (V_{be})_{αγωγ.}}{R_e} \quad (5-2)$$

$$I_c = \frac{V - V_{ce}}{R_c}$$

Κατά συνέπεια, όταν το τρανζίστορ είναι σε αγωγιμότητα, η τάση εξόδου είναι μικρή και η T έχει λογική τιμή «0». Στην περίπτωση που μία από τις εισόδους της πύλης έχει οδηγηθεί σε λογική κατάσταση «0», η είσοδος του κυκλώματος άρνησης θα έχει ομοίως λογικό «0» και θα ισχύουν τα παρακάτω: (βλέπε σχήμα 5.15).

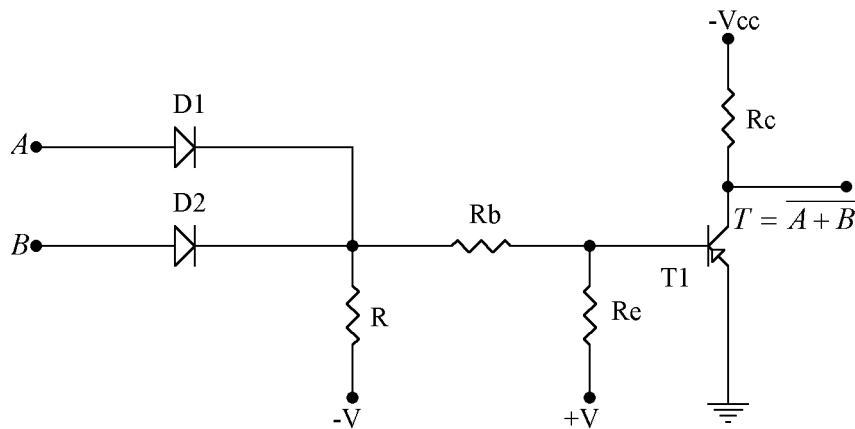


Σχήμα 5.15. Κύκλωμα για τον υπολογισμό επί μέρους τάσεων

$$\frac{(V_{be})_{\alpha\text{ποκ.}}}{R_b} = \frac{-V}{R_b + R_e} \Rightarrow (V_{be})_{\alpha\text{ποκ.}} = -\frac{R_b}{R_e + R_b} \cdot V$$

δηλαδή η T έχει λογική τιμή «1».

Κατ' αναλογία, το κύκλωμα του σχήματος 5.16 παριστάνει πύλη NOR.



Σχήμα 5.16. Πύλη NOR

5.4.2. Η πύλη XOR με διακριτά στοιχεία

Η λογική συνάρτηση $T = \overline{A \cdot B} \cdot \overline{\overline{A \cdot B}}$, που περιγράφει την έξοδο του σχήματος 5.6β, χρησιμοποιείται και για την υλοποίηση του κυκλώματος του σχήματος 5.17, όπου αποδίδεται η πύλη XOR με διακριτά στοιχεία. Όπως εύκολα γίνεται αντιληπτό, στη σχέση αυτή υπεισέρχονται μόνο οι λογικές πράξεις NAND και NOT οι οποίες είναι εύκολο να πραγματοποιηθούν με τη χρήση κοινών τρανζίστορ τύπου NPN.

Το τρανζίστορ T_1 συμπεριφέρεται σαν αναστροφέας παράγοντας τη λογική ποσότητα \overline{B} που είναι συμπληρωματική της μεταβλητής B . Ανάλογη εργασία πραγματοποιεί το τρανζίστορ T_2 το οποίο φροντίζει για την ανάδειξη της ποσότητας \overline{A} από τη μεταβλητή A .

Οι δύο ανεστραμμένες μεταβλητές οδηγούνται στη βαθμίδα που σχηματίζουν τα τρανζίστορ T_5 και T_6 , συνδεδεσολογημένα σε διάταξη πύλης NAND, για να δώσουν τελικά την ποσότητα $\overline{A \cdot B}$. Οι μεταβλητές εισόδου, χωρίς καμία τροποποίηση, εισάγονται στις βάσεις των τρανζίστορ T_3 και T_4 , για να δώσουν με όμοιο τρόπο την ποσότητα $\overline{A \cdot B}$.

Οι λογικές ποσότητες που παράγονται από τις δύο αυτές πύλες NAND οδηγούνται κατόπιν σε μια ακόμα πύλη NAND που αποτελείται από τα τρανζίστορ T_7 και T_8 , τα οποία τις επεξεργάζονται, για να δώσουν τελικά τη ποσότητα $\overline{\overline{A \cdot B} \cdot \overline{\overline{A \cdot B}}}$. Η λογική αυτή συνάρτηση ισοδυναμεί με το αντίστροφο της συνάρτησης T που επιθυμούμε να πραγματοποιήσουμε. Έτσι αυτό που μένει προκειμένου να αναιρέσουμε το σύμβολο της συμπλήρωσης είναι να οδηγήσουμε τον συλλέκτη του T_7 στη βάση του T_9 , που εργάζεται σαν αναστροφέας στο συλλέκτη του τελευταίου θα έχουμε την επιθυμητή συνάρτηση T δηλαδή την $T = \overline{A \cdot B} \cdot \overline{\overline{A \cdot B}}$.

Αξίζει να σημειώσουμε ότι, αν το ζητούμενο από αυτό το κύκλωμα δεν ήταν η πραγματοποίηση της λογικής πράξης XOR αλλά της πράξης XNOR, δε θα κάναμε καμία αλλαγή στις συνδεσμολογίες, αλλά θα α-

φαιρούσαμε απλώς τη βαθμίδα αναστροφής που είναι σχεδιασμένη γύρω από το T_9 . Αυτό είναι αναμενόμενο, αφού έχει αποδειχθεί ότι οι δύο αυτές συναρτήσεις είναι συμπληρωματικές.

5.5. Ολοκληρωμένα κυκλώματα

Όπως αναφέραμε και στην αρχή του κεφαλαίου, το ηλεκτρονικό μέρος των συγχρόνων υπολογιστικών συστημάτων κατασκευάζεται σήμερα με τη βοήθεια ολοκληρωμένων κυκλωμάτων (integrated circuits). Ένα ολοκληρωμένο κύκλωμα αποτελείται από ένα κρύσταλλο ημιαγωγίου υλικού, κυρίως πυριτίου, στο οποίο με κατάλληλη επεξεργασία δημιουργούνται όλα τα στοιχεία του κυκλώματος και οι μεταξύ αυτών συνδέσεις. Οι χρησιμοποιούμενες μέθοδοι επεξεργασίας είναι ανάλογες μ' αυτές που ακολουθούνται για την κατασκευή των τρανζίστορ και των διόδων ημιαγωγού. Τα ολοκληρωμένα κυκλώματα διατίθενται σήμερα στο εμπόριο σε μεγάλη ποικιλία και σε διάφορες μορφές, ανάλογα με το πεδίο εφαρμογής τους και την κατασκευάστρια εταιρεία.

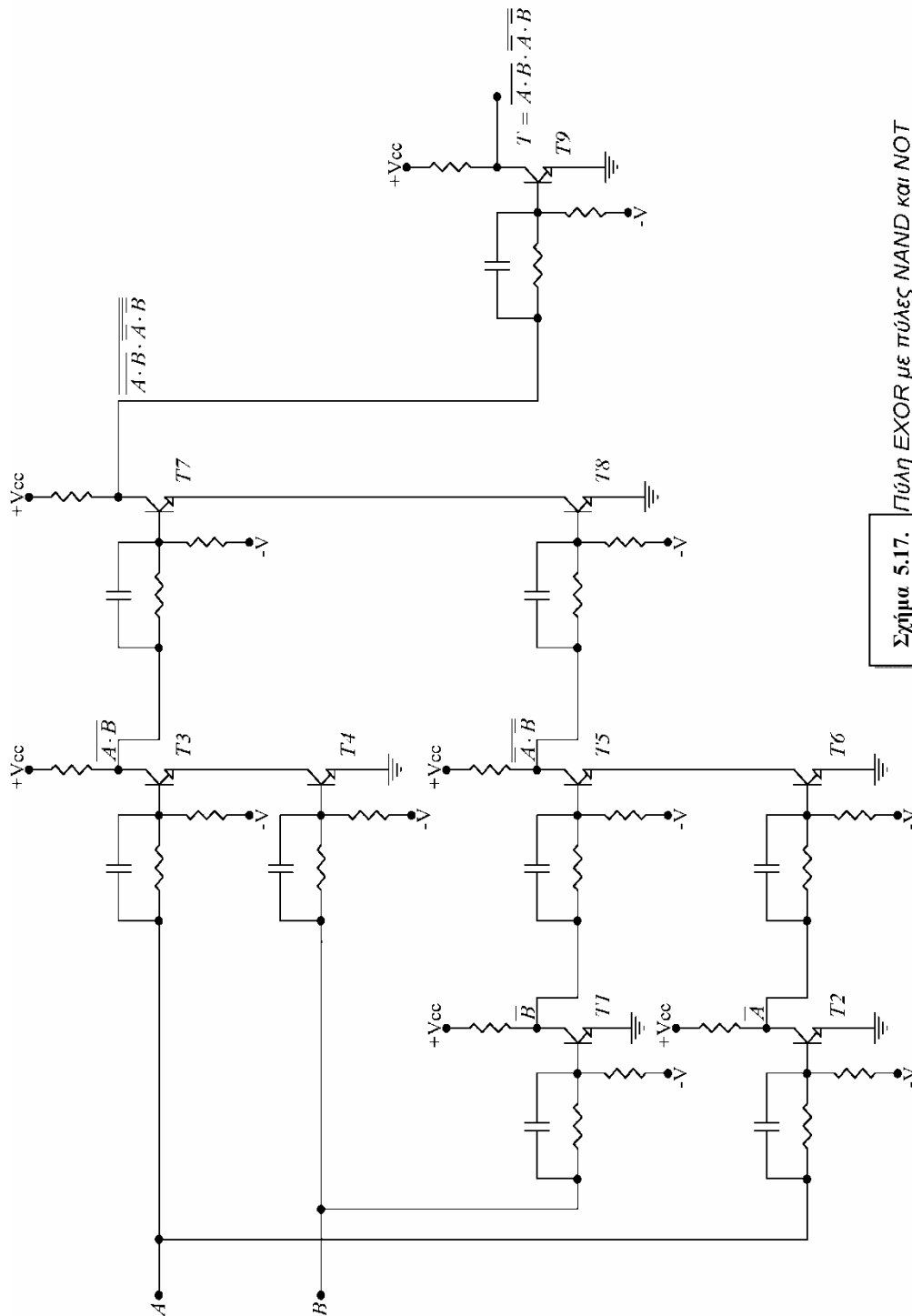
Τελευταία τα ολοκληρωμένα κυκλώματα έχουν κατακλύσει το χώρο της ηλεκτρονικής, λόγω του πολύ μικρού όγκου τους, κάτι που έχει επιτευχθεί χάρη στη μεγάλη ανάπτυξη της μικροηλεκτρονικής. Τα προηγούμενα σαράντα χρόνια, δηλαδή από την ανάπτυξη του τρανζίστορ και μέχρι πρότινος, η πρόοδος στην ελαχιστοποίηση των επεξεργασιών είχε εκθετική καμπύλη, το δε μέγεθος μειωνόταν στο μισό κάθε 18 μήνες. Το κόστος της κατασκευής ενός τρανζίστορ είναι πλέον το ένα δισεκατομμυριοστό του κόστους του το 1950, γεγονός που δεν έχει συμβεί σε καμία άλλη βιομηχανία παραγωγής υλικού χρήσιμου για τον άνθρωπο.

Σήμερα πλέον η τεχνολογία έχει αναπτυχθεί τόσο πολύ, που μιλάμε πλέον για νανοδιατάξεις και νανοδομές. Το Δεκέμβριο του 2005 ανακοινώθηκε η τότε νέα προσπάθεια για την κατασκευή διατάξεων σε κλίμακα κάτω των 45nm. Βέβαια είναι προφανές ότι πλέον οι ηλεκτρονικές διατάξεις έχουν γίνει εξαιρετικά πολύπλοκες, αποτελούμενες από χιλιάδες ανεξάρτητα μέρη και στοιχεία που οδηγούν σε αυξανόμενες κατασκευαστικές δυσκολίες και βλάβες. Ελαττώνοντας όμως τον αριθμό των εξαρτημάτων και των συνδέσεων μιας συσκευής, μέσω της χρήσης ολοκληρωμένων κυκλωμάτων, επιτυγχάνεται η ευκολότερη κατασκευή της και ως εκ τούτου η μεγαλύτερη αξιοπιστία της.

Οι όροι μικρής κλίμακας ολοκλήρωση (SSI), μεσαίας κλίμακας ολοκλήρωση (MSI), μεγάλης κλίμακας ολοκλήρωση (LSI), πολύ-μεγάλης κλίμακας ολοκλήρωση (VLSI), και υπερ-υψηλής κλίμακας ολοκλήρωση (ULSI) αναφέρονται στην εξέλιξη ολοκλήρωσης τρανζίστορ σε ICs. Η SSI αποδείχτηκε θεμελιώδης για τη δημιουργία ψηφιακών υπολογιστών μικρού βάρους για τα προγράμματα αεροδιαστημικής, όπως τα προγράμματα των πυραύλων Minuteman και Απόλλων.

Περισσότερα για τον τρόπο κατασκευής των ολοκληρωμένων κυκλωμάτων, όπως και για την τεχνολογία τους, μπορεί να βρει ο αναγνώστης στα αντίστοιχα βιβλία εφαρμοσμένης ηλεκτρονικής. Στο παρόν βιβλίο θα περιορίσουμε το ενδιαφέρον μας στη χρήση των ολοκληρωμένων κυκλωμάτων στις εφαρμογές με λογικά κυκλώματα.

Οι πιο συνηθισμένες πύλες που συναντώνται σε μορφή ολοκληρωμένων κυκλωμάτων είναι οι NOR και NAND, των οποίων η λειτουργία τους εξηγήθηκε σε προηγούμενες ενότητες. Οι πύλες που περιγράφηκαν σ' αυτές ήταν κατασκευασμένες με αντιστάσεις και τρανζίστορ, υλοποιώντας τις αρχές σχεδίασης της οικογένειας RTL (Resistor Transistor Logic), ή από διόδους και τρανζίστορ, υλοποιώντας τις αρχές σχεδίασης της οικογένειας DTL (Diode Transistor Logic). Σήμερα οι περισσότεροι κατασκευαστές ηλεκτρονικών ψηφιακών συσκευών προτιμούν τη χρήση ολοκληρωμένων κυκλωμάτων που ανήκουν στις οικογένειες TTL (Transistor Transistor Logic) ή CMOS (Complementary Metal Oxide Semiconductors), των οποίων η λειτουργική συμπεριφορά έχει να επιδείξει καλύτερα χαρακτηριστικά. Στις εφαρμογές που είναι απαραίτητη η χρήση υψηλών συχνοτήτων, τα ολοκληρωμένα της οικογένειας ECL (Emitter Coupled Logic) είναι τα μοναδικά που είναι σε θέση να ανταποκριθούν, αφού η συχνότητα λειτουργίας τους φθάνει το 1 GHz. (*Δεληγιάννης, 2004; Jaeger, & Blalock, 2011; Sedra, & Smith, 2015*).



Σχήμα 5.17. Πύλη EXOR με πύλες NAND και NOT

Τα χαρακτηριστικά που πρέπει να λαμβάνονται υπόψη για την εκλογή ενός ολοκληρωμένου λογικού κυκλώματος είναι βασικά τα παρακάτω:

- α) η τάση τροφοδοσίας του κυκλώματος,
- β) η κατανάλωση,
- γ) το κόστος,
- δ) η ταχύτητα (συχνότητα) λειτουργίας.
- ε) ο μέγιστος αριθμός εισόδων (Fan-In) που μπορεί να έχει το ολοκληρωμένο κύκλωμα.
- ζ) ο μέγιστος αριθμός εισόδων που μπορούν να οδηγηθούν από την έξοδο του ολοκληρωμένου κυκλώματος (Fan-Out).

A/A	Περιγραφή	Τύπος
1	Τέσσερις πύλες NAND 2 εισόδων	7400
2	Τέσσερις πύλες NOR 2 εισόδων	7402
3	Έξι πύλες NOT	7404
4	Τέσσερις πύλες AND 4 εισόδων	7408
5	Τρεις πύλες NAND 3 εισόδων	7410
6	Τέσσερις πύλες OR 2 εισόδων	7432
7	Πύλη NAND 8 εισόδων	7430
8	Μετατροπέας BCD σε μορφή 7 τμημάτων	7442
9	Διπλό F-F τύπου D	7474
10	Δυαδικός αθροιστής 4 bit με γρήγορο κρατούμενο	7483
11	Συγκριτής 4 bit	7485
12	Τέσσερις πύλες XOR 2 εισόδων	7486
13	Δεκαδικός μετρητής	7490
14	Δυαδικός μετρητής 4 bit	7493
15	Διπλός επαναδιεγερόμενος μονοδονητής	74123
16	Αποκωδικοποιητής 3 εισόδων σε 8 εξόδους	76138
17	Διπλός αποκωδικοποιητής 2 εισόδων σε 4 εξόδους	74139
18	Διπλός πολυπλέκτης 4 εισόδων	74155
19	Αριθμητική Λογική Μονάδα (ALU) 4 bit	74181
20	Προγραμματιζόμενος δυαδικός μετρητής 4 bit	74197

Πίνακας 5.7. Αντιπροσωπευτικοί τύποι ολοκληρωμένων κυκλωμάτων της οικογένειας T.T.L.

Εκτός των ολοκληρωμένων κυκλωμάτων SSI (Small Scale Integration) που περιλαμβάνουν απλές πύλες λογικής, έχουν κατασκευασθεί και πολλά ακόμα που περιλαμβάνουν σύνθετα λογικά κυκλώματα όπως, καταχωρητές, ολισθητές, απαριθμητές, ημιαθροιστές, αθροιστές, κωδικοποιητές κ.λπ., που, στο σύνολο τους, ονομάζονται M.S.I. (Medium Scale Integration). Ακόμα πιο πολύπλοκα κυκλώματα όπως Αριθμητικές Μονάδες (ALU), Μετατροπείς D/A (Digital to Analog) και A/D (Analog to Digital) κ.λπ., περιλαμβάνουν τα ολοκληρωμένα κυκλώματα LSI (Large Scale Integration). Όλα τα παραπάνω κυκλώματα αποτελούν ύλη του μαθήματος.

Τέλος με την ονομασία VLSI (Very Large Scale Integration) έχουν κατασκευασθεί μικροεπεξεργαστές (microprocessors) και μικροελεγκτές (microcontrollers) που είναι ό,τι καλύτερο έχει να επιδείξει η ηλεκτρονική τεχνολογία. Σήμερα πλέον χρησιμοποιείται η τεχνολογία ULSI.

Στον πίνακα 5.7 σημειώνονται οι τύποι μερικών αντιπροσωπευτικών ολοκληρωμένων ψηφιακών κυκλωμάτων της οικογένειας T.T.L., που είναι κατασκευασμένα σύμφωνα με τις τεχνολογίες S.S.I. και M.S.I.

Ολοκληρώνοντας, αναφέρουμε ότι η MSI έφερε την ενσωμάτωση των εκατοντάδων των τρανζίστορ ανά chip. Ένας Itanium επεξεργαστής σχεδιασμένος από την Intel Corporation (αρχικά με την κωδική ονομασία Tanglewood, που πήρε τελικά την κωδική ονομασία Tukwila (πόλη του Seattle), κυκλοφόρησε το 2010 με το όνομα Itanium 9300 Series) ήταν ο πρώτος επεξεργαστής που περιέχει περισσότερα από 2 δισεκατομμύρια τρανζίστορ και έχει επιφάνεια 698,75 mm². Ενημερωτικά αναφέρουμε τον Itanium 9500- 64-bit (κωδικός Roulson) που είναι με τεχνολογία των 32 nm και παρουσιάστηκε τον Νοέμβριο του 2012 με δισεκατομμύρια τρανζίστορ, 8 πυρήνες και 54 MB μνήμη στο τσίπ.

Αυτό αναδεικνύει μια σημαντική πτυχή της ολοκλήρωσης, δηλαδή την ικανότητα να κατασκευάζουμε πολύ μεγάλα, πολύ περίπλοκα, πολύ υψηλής απόδοσης ηλεκτρονικά κυκλώματα αξιόπιστα και με σχετικά χαμηλό κόστος. Η προσπάθεια αυτή αναδεικνύει, επίσης, μία από τις βασικές προκλήσεις της ολοκλήρωσης αυτής της ισχύος και των συναφών συνεπειών της.

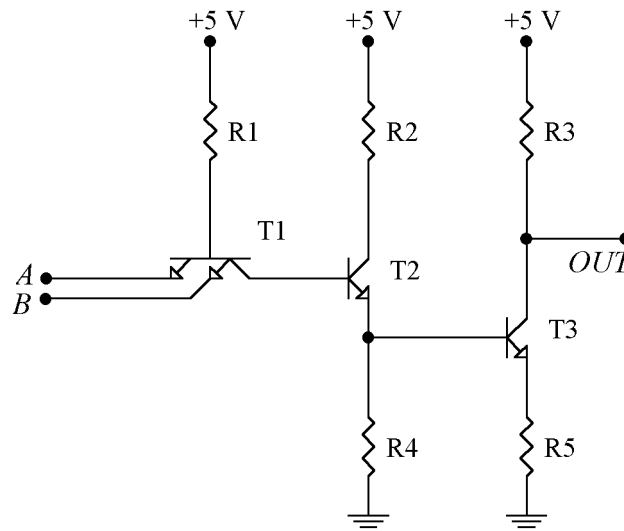
5.5.1. Οικογένεια ολοκληρωμένων κυκλωμάτων TTL

Στην οικογένεια των λογικών κυκλωμάτων αυτής της κατηγορίας, το βασικό στοιχείο, που περιλαμβάνεται στην κατασκευή οποιασδήποτε πύλης, είναι το τρανζίστορ των πολλαπλών εκπομπών, το οποίο κατασκευάζεται μόνο σε μορφή ολοκληρωμένου κυκλώματος. Πρόκειται για ένα τρανζίστορ, του οποίου η συμπεριφορά προσομοιάζει στη συμπεριφορά του κυκλώματος αντίστασης - διόδων του σχήματος 5.10, εξασφαλίζοντας

όμως μεγαλύτερη ταχύτητα λειτουργίας σε σχέση με το κύκλωμα των διόδων. (Κοσσιδάς, 1996; Δεληγιάννης, 2004; Tocci, et al., 2010; Jaeger, & Blalock, 2011; Sedra, & Smith, 2015).

Στο Σχήμα 5.18 βλέπουμε ένα τυπικό κύκλωμα πύλης NAND, που αποτελεί άλλωστε και τη βασική πύλη της οικογένειας. Απαραίτητη για την κατανόηση του κυκλώματος είναι η θεώρηση ότι το τρανζίστορ ισοδυναμεί με τρεις διόδους ενωμένες με τέτοιο τρόπο, ώστε να σχηματίζουν τους ακροδέκτες του τρανζίστορ. Μιλώντας πιο συγκεκριμένα, μπορούμε να θεωρήσουμε ότι ένα ζευγάρι διόδων σχηματίζει τις επαφές βάσης –εκπομπού, έχοντας τις δύο ανόδους ενωμένες στον ακροδέκτη της βάσης και αφήνοντας τις καθόδους να αποτελούν τις εισόδους του κυκλώματος, ενώ μια ακόμα διόδος, που σχηματίζει την επαφή βάσης - συλλέκτη, έχει την άνοδό της συνδεδεμένη στη βάση, αφήνοντας την κάθοδό της να αποτελεί το συλλέκτη του εξαρτήματος.

Όταν μια τουλάχιστον από τις δύο εισόδους του πολυεπαφικού τρανζίστορ οδηγηθεί σε λογικό «0», το ρεύμα που θα διαρρέει τη συγκεκριμένη επαφή βάσης - εκπομπού θα δεσμεύει την τάση στη βάση του τρανζίστορ στα 0,7 V περίπου (τάση ορθής φοράς επαφής πυριτίου). Αν όμως το δυναμικό στο σημείο αυτό έχει αυτή την τιμή, τότε η διόδος, που παίζει το ρόλο της επαφής βάσης - συλλέκτη του ίδιου τρανζίστορ, δε θα μπορεί να γίνει αγωγή, αφού, για να συμβεί κάτι τέτοιο, απαιτείται τάση τουλάχιστον ίση με 2,1 V. Η τάση αυτή προκύπτει από το άθροισμα των τάσεων ορθής φοράς των επαφών βάσης - εκπομπού των τρανζίστορ T₂ και T₃ και, επιπλέον, της διόδου βάσης - συλλέκτη του T₁. Κάτω από αυτές τις συνθήκες, είναι αδύνατο να γίνουν αγωγή τα T₂ και T₃, με αποτέλεσμα η τάση εξόδου του συνολικού κυκλώματος να μένει καθλωμένη στα +5 V περίπου.



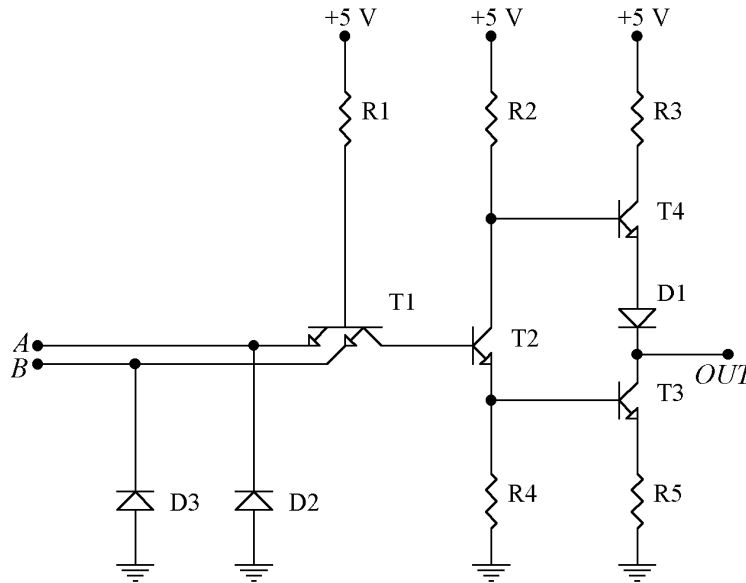
Σχήμα 5.18. Τυπικό κύκλωμα πύλης NAND οικογένειας TTL

Αν και οι δύο εισοδοί της πύλης οδηγηθούν ταυτόχρονα σε λογικό «1», αποκτήσουν δηλαδή τάση που πλησιάζει την τάση τροφοδοσίας, τότε η τάση στη βάση του πολυεπαφικού τρανζίστορ θα είναι σημαντικά υψηλότερη από την τάση των 2,1 V, που απαιτείται για την αγωγιμότητα της ισοδύναμης διόδου βάσης - συλλέκτη του T₁, με αποτέλεσμα να ρέει ρεύμα βάσης στα T₂ και T₃. Συνέπεια της ροής ρεύματος βάσης θα είναι αγωγιμότητα των T₂ και T₃ και η εμφάνιση, στην έξοδο του κυκλώματος, τάσης ίσης με 0,2 V (τάση κορεσμού του τρανζίστορ) που αντιστοιχεί σε λογικό «0».

Οι λογικές καταστάσεις που επιβάλλονται στις εισόδους του κυκλώματος επαληθεύουν τον πίνακα αλήθειας της πύλης NAND (πίνακας 5.2), που περιγράφηκε στην παράγραφο 5.2.

Στο Σχήμα 5.19, που ακολουθεί απεικονίζεται η τελική μορφή μιας πύλης NAND, όπως αυτή υφίσταται μέσα στο ολοκληρωμένο κύκλωμα 7400, που αποτελεί ένα από τα αντιπροσωπευτικότερα της οικογένειας T.T.L. Οι διόδοι D₂ και D₃, οι οποίες έχουν τοποθετηθεί μεταξύ των εισόδων και της γης, προστατεύουν το κύκλωμα από τις καταστροφικές συνέπειες της επιβολής μιας αρνητικής τάσης στις εισόδους, ενώ το «υπερτιθέμενο» τρανζίστορ T4 μαζί με τη διόδο D1 εξασφαλίζουν την υψηλή ταχύτητα μεταγωγής της εξόδου, όταν τα συνδεδεμένα σ' αυτή φορτία παρουσιάζουν έντονη χωρητική συμπεριφορά.

Παραλλαγές της βασικής οικογένειας T.T.L. 74xxx αποτελούν και οι οικογένειες 74Lxxx (Low power), 74Sxxx (Schottky), 74LSxxx (Low power Schottky), 74Fxxx (Fast) και άλλες. Όλες αυτές έχουν να επιδείξουν κάποια χαρακτηριστικά τα οποία, ανάλογα με το πεδίο εφαρμογής, αποκτούν ιδιαίτερη σημασία.

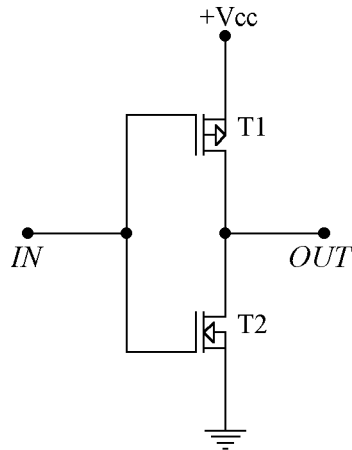


Σχήμα 5.19. Πλήρες κύκλωμα πύλης NAND οικογένειας TTL

5.5.2. Οικογένεια ολοκληρωμένων κυκλωμάτων CMOS

Το μεγαλύτερο μειονέκτημα των ολοκληρωμένων της οικογένειας TTL είναι η σχετικά υψηλή κατανάλωση ρεύματος που δρα περιοριστικά στη σχεδίαση. Αυτό συμβαίνει, γιατί στη σχεδίαση απαιτείται μεγάλος αριθμός ολοκληρωμένων κυκλωμάτων. Η οικογένεια C.M.O.S παρακάμπτει αυτό το εμπόδιο, προσφέροντας ολοκληρωμένα κυκλώματα με καταναλώσεις της τάξης των μA , ενώ, παράλληλα, εξασφαλίζει την υψηλή ταχύτητα λειτουργίας των TTL. Σήμερα η οικογένεια των CMOS θεωρείται αναντικατάστατη στην κατασκευή διατάξεων, πιστεύεται δε ότι η περαιτέρω ανάπτυξη μετά την εισαγωγή των νανοδιατάξεων δε θα είναι ποτέ της ίδιας ποιότητας όπως με τα CMOS τρανζίστορ.

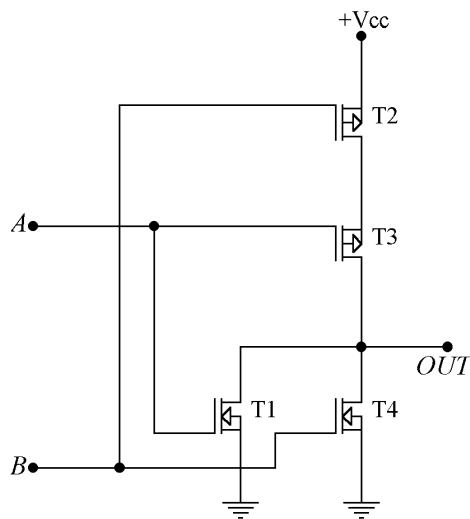
Ο αναστροφέας CMOS (πύλη NOT) κατασκευάζεται πιο εύκολα απ' όλες τις άλλες πύλες, αφού απαιτεί μόνο δύο συμπληρωματικά τρανζίστορ πεδίου μονωμένης πύλης (τρανζίστορ MOSFET). Η συνδεσμολογία τους είναι αυτή που φαίνεται στο Σχήμα 5.20, από την οποία προκύπτει και η λειτουργία της πύλης. Αν στην είσοδο του κυκλώματος επιβληθεί λογικό «1», το τρανζίστορ T_2 (καναλιού N) θα γίνει αγωγίμο, ενώ το T_1 (καναλιού P) θα οδηγηθεί σε αποκοπή. Σαν αποτέλεσμα θα έχουμε την εμφάνιση, στην έξοδο του κυκλώματος, μιας τάσης ίσης σχεδόν μ' αυτήν της γης, δηλαδή τάσης που θα αντιστοιχεί στο λογικό «0». Αν στην είσοδο επιβληθεί λογικό «0», τότε οι παραπάνω καταστάσεις αντιστρέφονται, και στην έξοδο εμφανίζεται λογικό «1». (Κοσσίδης, 1996; Tocci, et al., 2010; Jaeger, & Blalock, 2011; Sedra, & Smith, 2015).



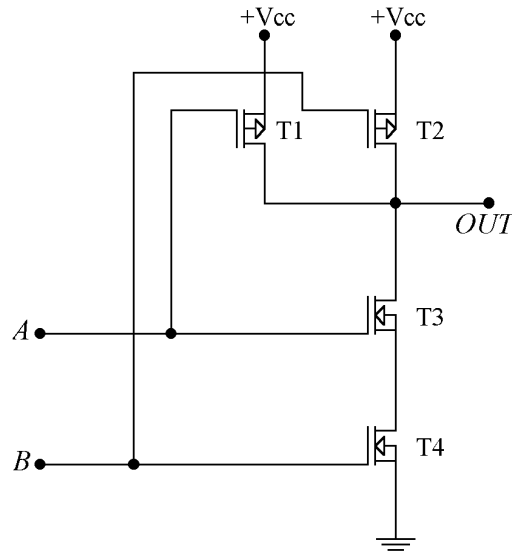
Σχήμα 5.20. Αναστροφέας οικογένειας CMOS

Επειδή μόνο ένα από τα δύο τρανζίστορ βρίσκεται κάθε φορά σε αγωγιμότητα, είναι φανερό ότι η κατανάλωση της πύλης είναι θεωρητικά μηδενική και πρακτικά πάρα πολύ μικρή. Τα δύο τρανζίστορ διαρρέονται από ρεύματα μόνο τις στιγμές εκείνες που μετáγονται από τη μια κατάσταση στην άλλη. Όσο πιο συχνά γίνονται μεταγωγές, τόσο πιο αυξημένη είναι η κατανάλωσή τους. Γι' αυτό άλλωστε στους καταλόγους των κατασκευαστών η κατανάλωσή τους δίδεται πάντα σε σχέση με τη συχνότητα λειτουργίας τους.

Στο Σχήμα 5.21 απεικονίζεται μια πύλη NOR, ενώ στο Σχήμα 5.22 μια πύλη NAND. Εύκολα γίνεται αντιληπτό ότι τα δύο κυκλώματα είναι συμμετρικά και, ως εκ τούτου, η περιγραφή της λειτουργίας τους μοιάζει. Επιβάλλοντας λογικό «0» και στις δύο εισόδους της NOR, τα τρανζίστορ T1 και T4 μένουν σε κατάσταση αποκοπής, ενώ τα T2 και T3 οδηγούνται σε κορεσμό. Στην έξοδο θα εμφανισθεί τότε τάση ίση με την τάση της θετικής πηγής τροφοδοσίας, δηλαδή λογικό «1». Αν σε μια από τις εισόδους επιβληθεί λογικό «0», τότε μόνο το ένα από τα δύο σε σειρά τρανζίστορ T2 και T3 θα είναι αγωγίμο, εμποδίζοντας την τάση τροφοδοσίας να αναδειχθεί στην έξοδο, ενώ η αγωγιμότητα ενός εκ των δύο παραλληλισμένων τρανζίστορ θα εξασφαλίσει ότι στην έξοδο θα έχουμε λογικό «0». Με ανάλογο τρόπο μπορεί να εξηγηθεί και η λειτουργία της πύλης NAND.



Σχήμα 5.21. Πύλη NOR οικογένειας CMOS



Σχήμα 5.22. Πύλη NAND οικογένειας CMOS

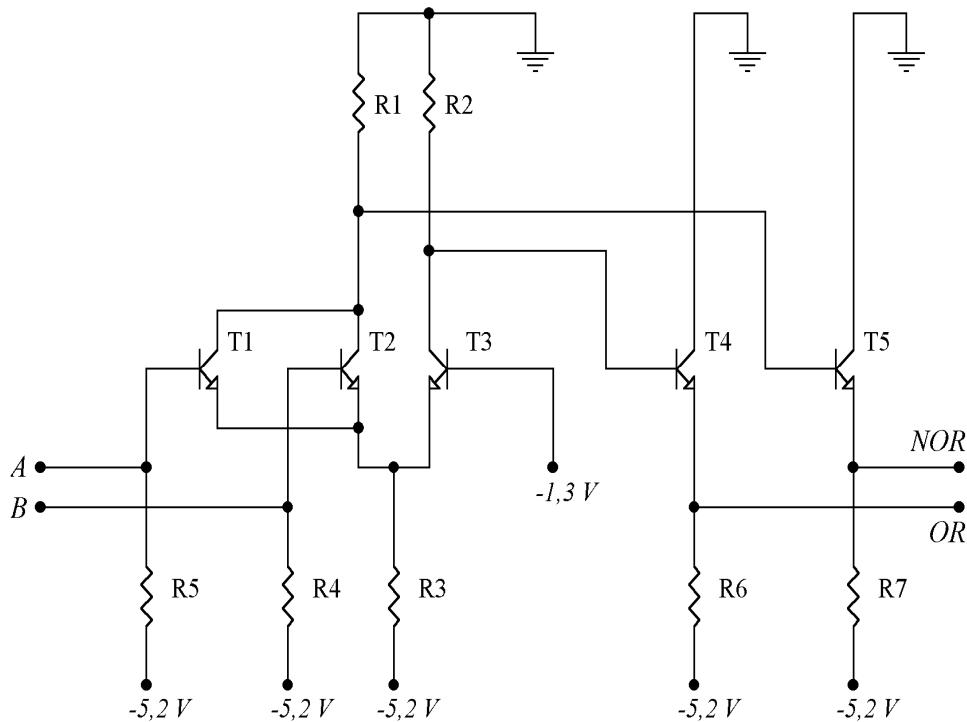
Αντιπροσωπευτικά κυκλώματα τεχνολογίας CMOS περιλαμβάνονται στις επιμέρους οικογένειες CD40xx, CD45xx, 74Cxxx, 74HCxxx και 74HCTxxx. Οι δύο πρώτες είναι γνωστές για την εξαιρετικά μικρή τους κατανάλωση (μA), αλλά και για τη χαμηλή συχνότητα λειτουργίας τους (μερικά MHz), ενώ οι υπόλοιπες, που έχουν ίδιο πρόθεμα με εκείνο των TTL, αποτελούνται από ολοκληρωμένα κυκλώματα που περιλαμβάνουν στο εσωτερικό τους τις ίδιες πύλες μ' αυτές της οικογένειας TTL, χρησιμοποιώντας τεχνολογία κατασκευής CMOS. Οι διαφορές που συναντάμε στις τρεις τελευταίες, έχουν σχέση με τις ηλεκτρικές στάθμες που γίνονται αντιληπτές σαν λογικό «1» και σαν λογικό «0», ή ακόμα και με την ταχύτητά τους.

5.5.3. Οικογένεια ολοκληρωμένων κυκλωμάτων ECL

Αυτή η οικογένεια ψηφιακών κυκλωμάτων, προκειμένου να εξασφαλίσει την υψηλότερη δυνατή ταχύτητα, παρουσιάζει αισθητές διαφορές από όλες τις προηγούμενες. Η ουσιαστικότερη από αυτές είναι ότι τα τρανζίστορ, που σχηματίζουν τις διάφορες πύλες δε μεταφέρονται μεταξύ κόρου και αποκοπής, αλλά εργάζονται πάντα σ' ένα μικρό σημείο της γραμμικής περιοχής. Η λειτουργία στην περιοχή αυτή δικαιολογεί την υψηλή ταχύτητά τους αλλά και την υπερβολική κατανάλωση ρεύματος που τα χαρακτηρίζει. Μια ακόμα διαφορά εντοπίζεται στο γεγονός ότι τροφοδοτούνται με αρνητική τάση (-5,2 V), απαραίτητη για την επίτευξη της μέγιστης δυνατής αδιαφορίας στο θόρυβο. (Κοσσιδάς 1996; Δεληγιάννης, 2004; Jaeger, & Blalock, 2011; Sedra, & Smith, 2015).

Η βασική πύλη της οικογένειας ECL είναι η OR/NOR, που φαίνεται στο Σχήμα 5.23. Όπως εύκολα γίνεται αντιληπτό, η συνδεσμολογία, που έχει τον πρωτεύοντα ρόλο είναι αυτή του διαφορικού ενισχυτή, ο οποίος, από τη μια πλευρά του σχηματίζεται από ένα τρανζίστορ του οποίου η βάση έχει σταθερό δυναμικό ίσο με -1,3 V, ενώ η άλλη σχηματίζεται από παραλληλισμένα τρανζίστορ, των οποίων ο αριθμός εξαρτάται από το πλήθος των εισόδων της πύλης. Τα δύο τρανζίστορ, που είναι συνδεσμολογημένα σε διάταξη κοινού συλλέκτη, μεταφέρουν στις δύο εξόδους τις λογικές καταστάσεις των εξόδων του διαφορικού ενισχυτή, εξασφαλίζοντας ταυτόχρονα υψηλά ρεύματα στα δύο φορτία.

Η λειτουργία του κυκλώματος είναι η εξής: Έστω ότι μια τουλάχιστον από τις δύο εισόδους οδηγείται σε λογικό «1» (στάθμη -0,8 V). Αυτό θα έχει σαν συνέπεια το τρανζίστορ εισόδου στο οποίο επιβλήθηκε η τάση αυτή να οδηγηθεί σε αγωγιμότητα, επιβάλλοντας στον εκπομπού του T_3 τάση ίση με -1,6 V (η τάση ορθής φοράς των επαφών βάσης - εκπομπού των τρανζίστορ είναι ίση με 0,8 V). Κάτω από αυτές τις συνθήκες, το T_3 οδηγείται σε αποκοπή, αφήνοντας όλο το ρεύμα που το διέρρει να οδηγηθεί στη βάση του T_4 .



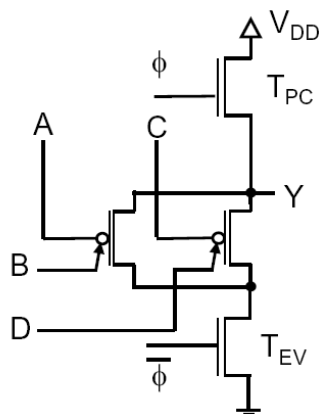
Σχήμα 5.23. Πύλη OR / NOR της οικογένειας ECL

Θεωρώντας αμελητέα την πτώση τάσης στα άκρα της R2, η τάση που θα εμφανισθεί στον εκπομπό του T4 (έξοδος πύλης OR) θα είναι ίση με την τάση στη βάση του, μειωμένη, όμως, λόγω της τάσης μεταξύ βάσης-εκπομπού του τρανζίστορ. Η τάση αυτή θα είναι $-0,8\text{ V}$, θα αντιστοιχεί, δηλαδή, στο λογικό «1». Ο εκπομπός του T5 (έξοδος πύλης NOR), λόγω της αγωγιμότητας του τρανζίστορ εισόδου, θα βρίσκεται σε δυναμικό ίσο με $-1,8\text{ V}$, που αντιστοιχεί σε λογικό «0». (Η τιμή της αντίστασης R1 έχει επιλεγεί με τέτοιο τρόπο, ώστε η πτώση τάση στα άκρα της, όταν διαρρέεται από ρεύμα, να είναι ίση με 1 V).

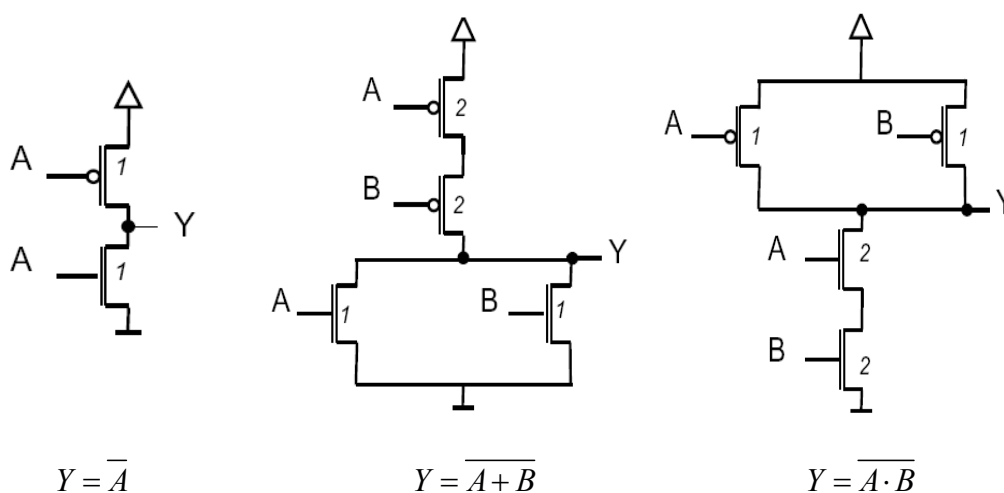
Όταν όλες οι εισόδου οδηγηθούν σε λογικό «0» ταυτόχρονα, τότε είναι προφανές ότι θα οδηγείται σε αγωγιμότητα το τρανζίστορ T3, οδηγώντας τις εξόδους των T4 και T5 σε καταστάσεις αντίθετες από αυτές που αναφέραμε προηγουμένως. (Tocci, et al., 2010; Jaeger, & Blalock, 2011; Sedra, & Smith, 2015).

5.6. Γενικευμένες πύλες NOR και NAND

Τα νέα τεχνολογικά επιτεύγματα έχουν εισάγει πλέον την νέα τεχνολογία με CNTFET, έναντι της τεχνολογίας CMOS, καθώς και την έννοια των γενικευμένων πυλών NOR (generalized NOR) και των γενικευμένων πυλών NAND (Haykel Ben Jamma, et al. 2011). Με βάση τις πρωτογενείς λειτουργίες των γενικευμένων πυλών NOR-NAND-AOI-OAI, οι προτεινόμενες πύλες στατικών αμφιπολικών CNTFET εφαρμόζουν αποτελεσματικά τις λειτουργίες της πύλης XOR, παρέχοντας πλήρη μεταγωγή της εξόδου. Η τεχνολογική απεικόνιση της δοκιμασίας επιδόσεων διαφόρων λογικών διατάξεων πολλαπλών επιπέδων-συμπεριλαμβανομένων των πολλαπλασιαστών, αθροιστών και γραμμικών κυκλωμάτων - δείχνει ότι, κατά μέσο όρο, είναι δυνατόν να μειωθεί τόσο ο αριθμός των πυλών όσο και η επιφάνεια κατά $\sim 38\%$, βελτιώνοντας παράλληλα την απόδοση κατά 6,9 φορές. (Haykel Ben Jamma, 2009). Στο παράδειγμα που ακολουθεί αποδίδεται το διάγραμμα της νέας πύλης του τύπου GNOR που αποτελεί το δομικό στοιχείο για τα νέα PLAs. Η συνάρτηση που υλοποιεί η κατωτέρω δυναμική γενικευμένη πύλη είναι η $Y=$

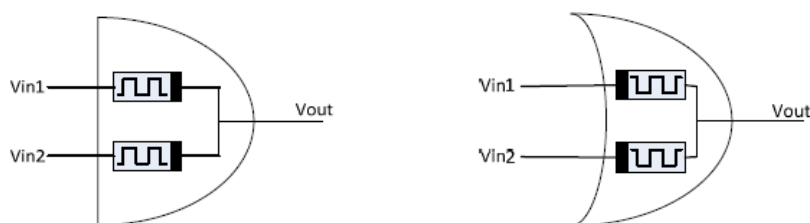


Σχήμα 5.24α. Απεικόνιση της συνάρτησης $Y = (A \oplus B) + (C \oplus D)$, με γενικευμένη πύλη CMOS



Σχήμα 5.24β. Απεικονίσεις πυλών με την χρήση της τεχνολογίας των CNTFET

5.6.1. Υβριδικά λογικά CMOS memristor



Σχήμα 5.24γ. Απεικονίσεις πυλών AND και OR με Memristor

Στην οικογένεια των υβριδικών CMOS memristor λογικής, οι λογικές τιμές αποθηκεύονται σαν τάσεις και τα memristor χρησιμοποιούνται μόνο σαν μονάδες υπολογισμού. Περιλαμβάνονται μόνο πύλες AND και OR και, συνεπώς, απαιτείται η χρήση ή δημιουργία και άλλων βασικών πυλών. Η διαφορά των δύο σχημάτων του Σχήματος 5.24γ (πύλη AND και πύλη OR) οφείλεται στην πολικότητα των memristor. (Kvatinsky, Satat, & Nimrod, 2012).

5.7. Κωδικοποιητές – Αποκωδικοποιητές με διακριτά στοιχεία

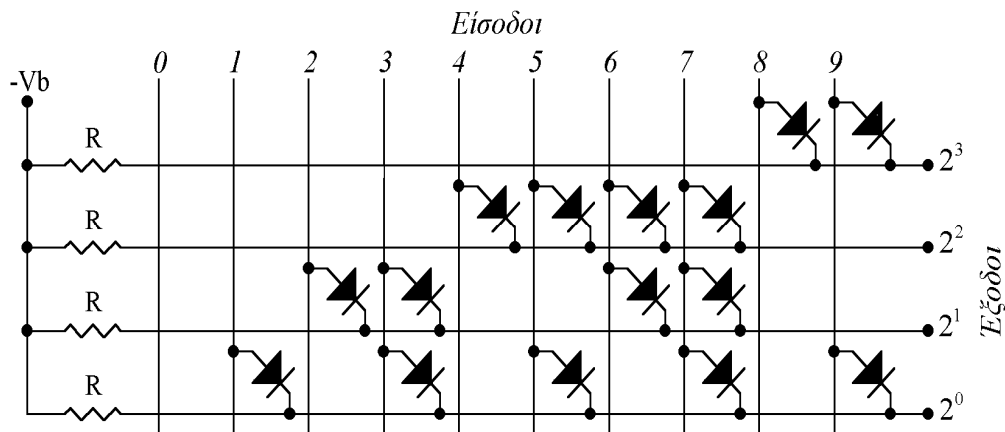
Οι κωδικοποιητές - αποκωδικοποιητές είναι λογικά κυκλώματα τα οποία μετατρέπουν χαρακτήρες από ένα σύστημα κωδικοποίησης σ' ένα άλλο. Κλασικό παράδειγμα κωδικοποιητών-αποκωδικοποιητών είναι η μετα-

τροπή δεκαδικών αριθμών σε δυαδικούς και αντιστρόφως. (Δεληγιάννης, 2004; Κοσσίδας, 1996; Sedra, & Smith, 2015)

Ως γνωστόν η αντιστοιχία μεταξύ δεκαδικών και δυαδικών αριθμών στο σύστημα BCD-8421 είναι η ακόλουθη:

Δεκαδικό	0	1	2	3	4	5	6	7	8	9
Δυαδικό	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Στο σχήμα 5.25α δίνεται το κύκλωμα κωδικοποίησης των δεκαδικών αριθμών στους αντίστοιχους δυαδικούς με διακριτά στοιχεία (δίοδοι και αντιστάσεις). Όπως φαίνεται στο κύκλωμα, υπάρχουν 4 πύλες OR με διαφορετικό αριθμό εισόδων η κάθε μία, που παράγουν τα ψηφία του τετραψηφίου δυαδικού αριθμού. Π.χ. το bit με βάρος $2^0=1$ εμφανίζεται, όταν ο υπό κωδικοποίηση δεκαδικός αριθμός, που επιβάλλεται στην είσοδο, είναι 1, 3, 5, 7 ή 9.



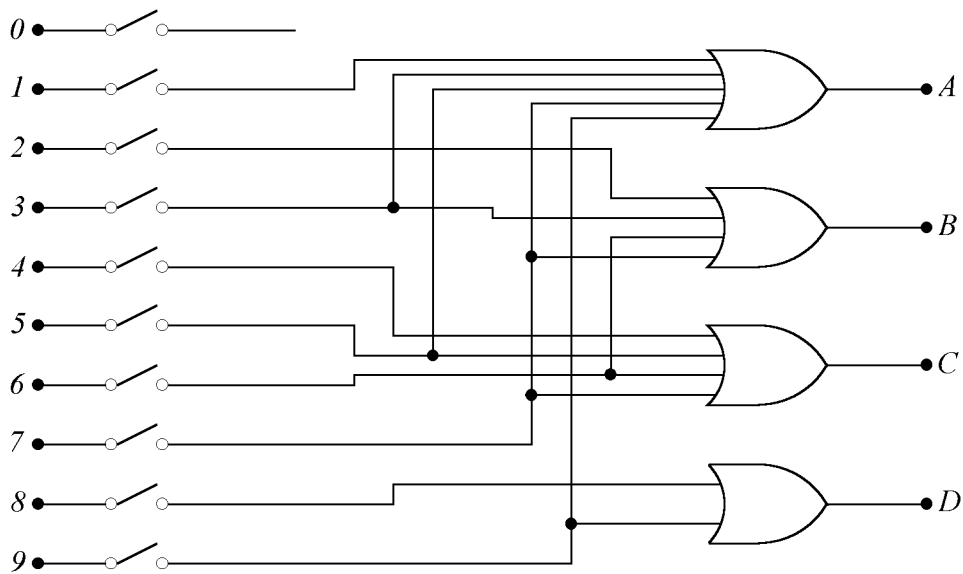
Σχήμα 5.25α. Κύκλωμα κωδικοποίησης δεκαδικού σε δυαδικό

Στο σχήμα 5.25β δίνεται το λογικό κύκλωμα του κωδικοποιητή με διακόπτες. Εδώ οι πύλες OR αναλαμβάνουν να κωδικοποιήσουν τις καταστάσεις 10 διαφορετικών μηχανικών διακοπών. Παρατηρήστε ότι ο διακόπτης του 0 δεν καταλήγει πουθενά.

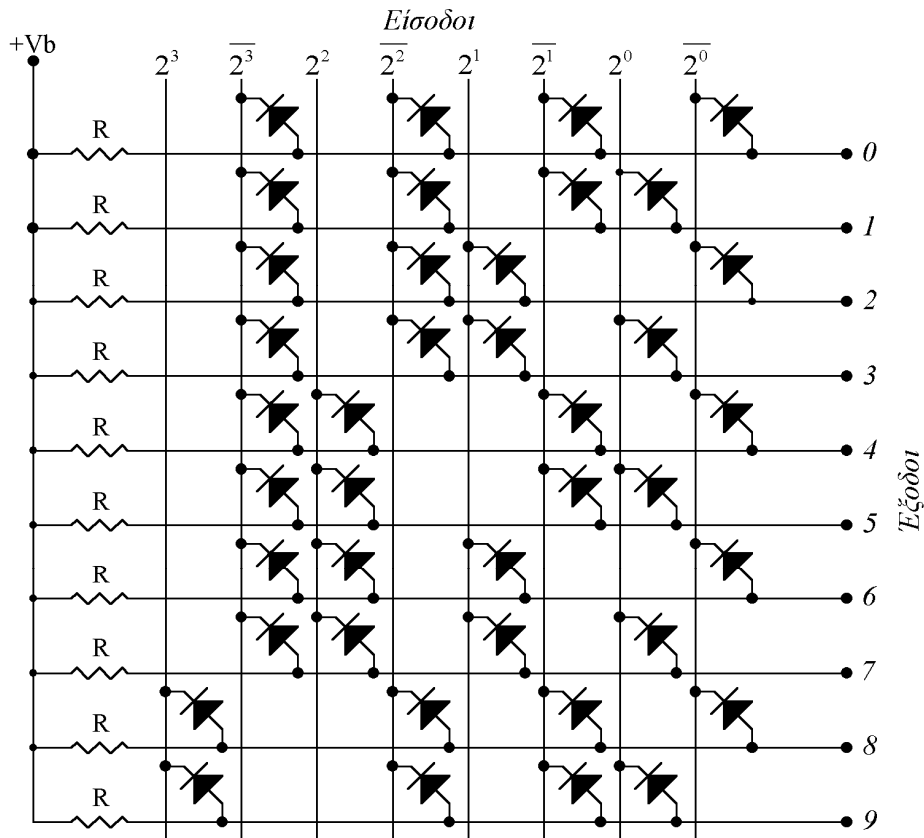
Στη συνέχεια παρουσιάζουμε ένα κύκλωμα αποκωδικοποίησης δυαδικών αριθμών στους αντίστοιχους του δεκαδικού συστήματος, που βασίζεται σε πύλες AND.

Όπως φαίνεται στο σχήμα 5.26, υπάρχουν 10 κυκλώματα AND με 4 εισόδους το καθένα, από τις εξόδους των οποίων λαμβάνουμε τους υπό κωδικοποίηση αριθμούς, εκφρασμένους με μορφή παλμού σε μια από αυτές.

Επίσης πρέπει να ληφθεί υπόψη ότι στην είσοδο του κυκλώματος υπάρχουν τόσο οι παλμοί με βάρη **1, 2, 4, 8** όσο και οι αρνήσεις αυτών $\bar{1}, \bar{2}, \bar{4}, \bar{8}$. Κατά συνέπεια, στον υπολογιστή είναι αναγκαίο μαζί με την ύπαρξη ενός παλμού να υπάρχει και η άρνησή του που παράγεται με τη βοήθεια ενός αναστροφέα (πύλη NOT). Επομένως στο κύκλωμα του σχήματος 5.26 χρειάζονται τέσσερις αναστροφείς, για να συμπληρωθούν πλήρως οι εισόδοι. Πολλές φορές στα κυκλώματα κωδικοποίησης - αποκωδικοποίησης, συνηθίζεται η τοποθέτηση ενός ψηφίου ισοτιμίας (parity bit), ώστε να υπάρχει, κατά τη μεταβίβαση, η δυνατότητα ελέγχου για το αν η μετάδοση της πληροφορίας είναι επιτυχής.



Σχήμα 5.25β. Κωδικοποιητής δεκαδικού με διακόπτες



Σχήμα 5.26. Κύκλωμα αποκωδικοποίησης δυαδικού αριθμού σε δεκαδικό

Στον πίνακα 5.8 που ακολουθεί, δίνονται όλες οι περιπτώσεις αποκωδικοποίησης ενός δυαδικού αριθμού στον ισοδύναμο δεκαδικό. Για τιμές πάνω από 1010 δεν έχουμε έξοδο. Κατόπιν τούτου, οι τιμές που ξεπερνούν το όριο αυτό δεν πρέπει να λαμβάνονται υπόψη.

Από τον πίνακα 5.8 προκύπτουν οι λογικές εξισώσεις:

$$0 = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$$

$$1 = \overline{A} \cdot \overline{B} \cdot C \cdot D$$

$$2 = \bar{A} \cdot \bar{B} \cdot C \cdot \bar{D}$$

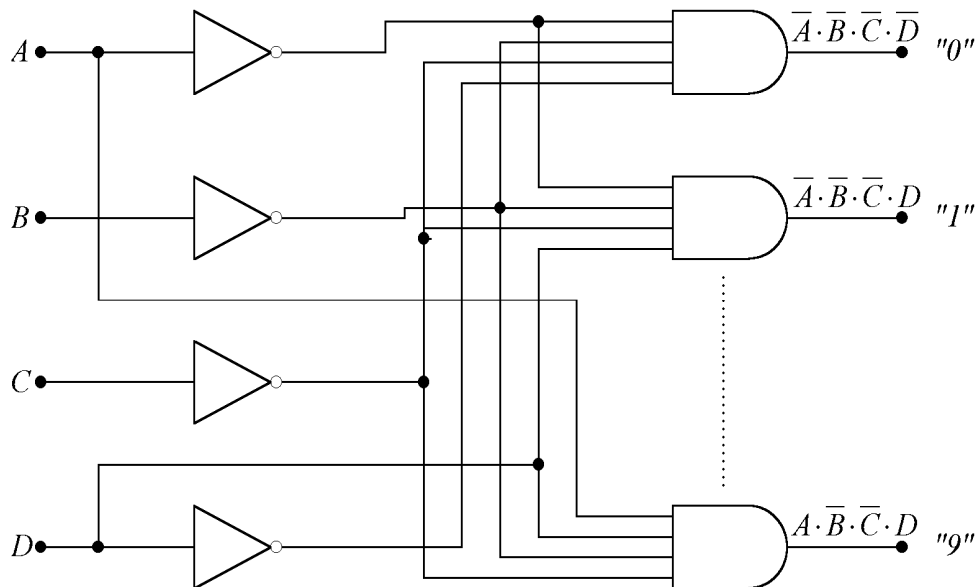
⋮
⋮
⋮

$$9 = A \cdot \bar{B} \cdot \bar{C} \cdot D$$

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

Πίνακας 5.8. Αποκωδικοποίηση δυαδικού αριθμού σε δεκαδικό

που με τη σειρά τους οδηγούν στο λογικό κύκλωμα του αποκωδικοποιητή (Σχήμα 5.27):



Σχήμα 5.27. Αποκωδικοποιητής δυαδικού αριθμού σε δεκαδικό

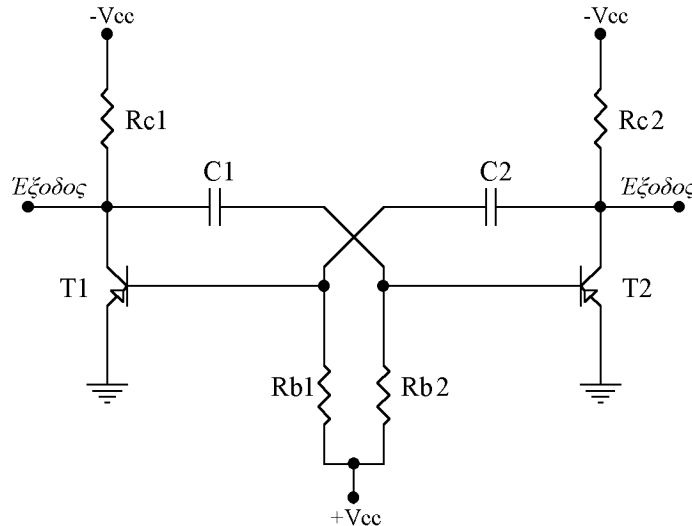
5.8. Κυκλώματα Πολυδονητών

Θα μελετήσουμε τα κυκλώματα πολυδονητών, που είναι παρά πολύ διαδεδομένα στους ηλεκτρονικούς υπολογιστές, κάνοντας μία πλήρη περιγραφή. (Nashelsky, 1994; Κοσσίδης, 1996; Sedra & Smith, 2015)

5.8.1. Ασταθής πολυδονητής

Ο ασταθής πολυδονητής (Astable multivibrator) χρησιμοποιείται πολύ στα κυκλώματα υπολογιστών, αφού παράγει τα σήματα χρονισμού βάσει των οποίων πραγματοποιούνται όλες οι εντολές των προγραμμάτων που εκτελεί. (Nashelsky, 1994; Κοσσιδάς, 1996; Δεληγιάννης, 2004; Sedra & Smith, 2015)

Στο Σχήμα 5.28 δίδεται το κύκλωμα ενός ασταθής συμμετρικού πολυδονητή, κατασκευασμένου από διακριτά εξαρτήματα. Αποτελείται από δύο τρανζίστορ το T_1 και το T_2 , καθένα από τα οποία έχει το συλλέκτη του συνδεδεμένο στη βάση του άλλου, διαμέσου ενός δικτυώματος αντίστασης-πυκνωτή.



Σχήμα 5.28. Ασταθής πολυδονητής

Μελετώντας το κύκλωμα, εύκολα διαπιστώνουμε ότι, έχοντας εφαρμόσει τις κατάλληλες τάσεις στο κύκλωμα, αυτό δεν μπορεί να παραμείνει σε μια σταθερή κατάσταση λειτουργίας, κάτι που συνέβαινε για τα προηγούμενα δύο είδη πολυδονητών.

Εάν υποθέσουμε ότι ένα από τα δύο τρανζίστορ άγει, έστω, π.χ., το τρανζίστορ T_1 , τότε θα ισχύει ότι η τάση $V_{C1} = (V_{CE})$ **αγκ1** και ότι η τάση $V_{C2} = (V_{CE})$ **αποκ1** = $-V_{CC}$ (Σχήμα 5.29).

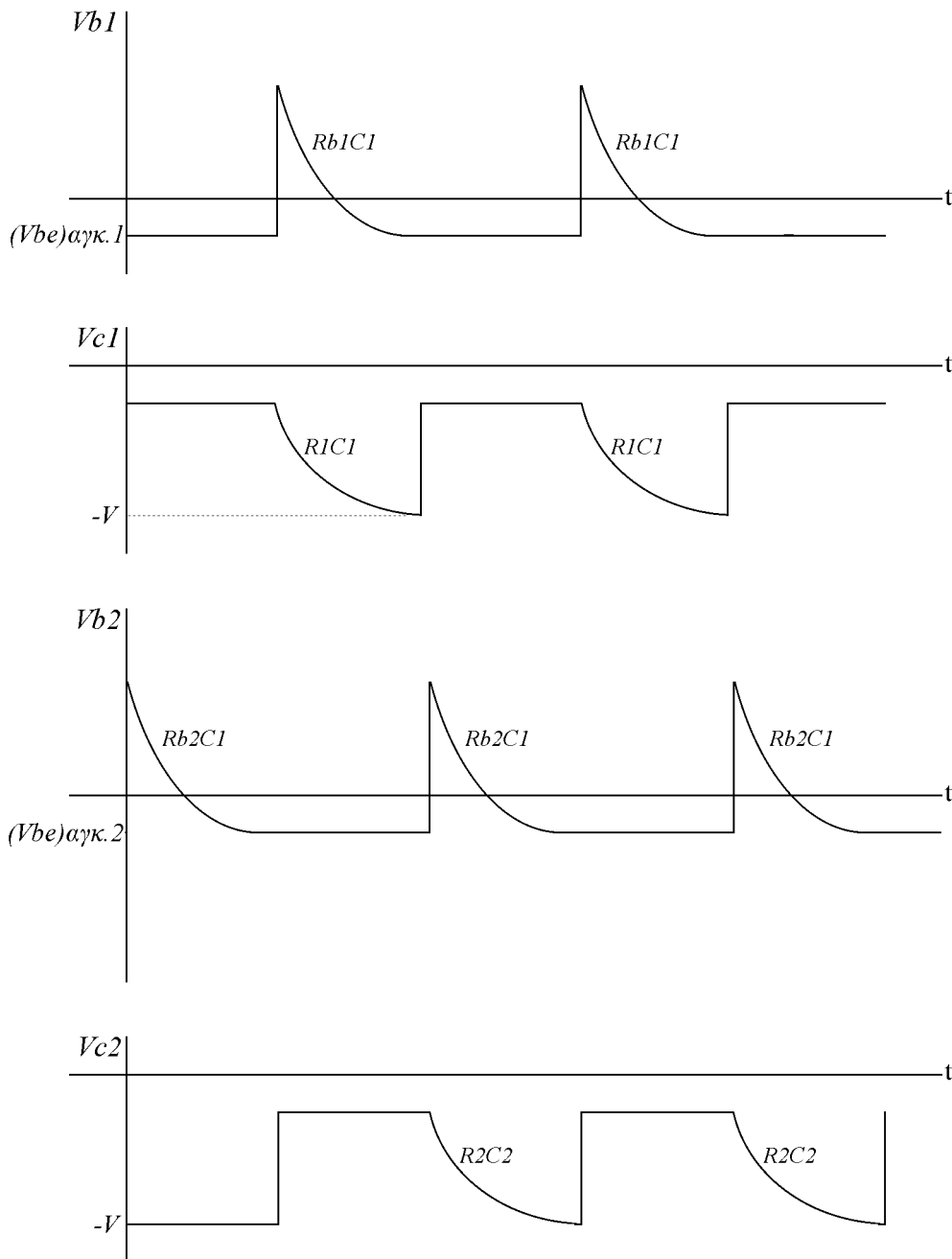
Από αυτή τη χρονική στιγμή, ο πυκνωτής C_1 εκφορτίζεται μέσω του T_1 και της R_{b2} με σταθερά χρόνου $R_{b2}C_1$. Καθώς εκφορτίζεται ο C_1 , κάποια στιγμή η τάση V_{b2} στη βάση του T_2 θα φθάσει την τιμή αγωγιμότητας (αρνητική) του T_2 , (V_{be}) **αγκ2**. Τότε το τρανζίστορ αυτό θα αρχίσει να άγει (εναλλαγή), με αποτέλεσμα την απότομη πτώση της τάσης στο συλλέκτη του (κατ' απόλυτη τιμή). Αυτό με τη σειρά του θα προκαλέσει την αποκοπή του T_1 λόγω της αύξησης της V_{b1} .

Στη συνέχεια αρχίζει η εκφόρτιση του C_2 με αποτέλεσμα την εκ νέου εναλλαγή, όταν η V_{b1} πάρει την κατάλληλη τιμή. Ο χρόνος εκφόρτισης του πυκνωτή είναι, κατά πολύ, μεγαλύτερος συγκριτικά με το χρόνο εναλλαγής.

Στο Σχήμα 5.29 δίνονται οι κυματομορφές των τάσεων στα διάφορα σημεία του κυκλώματος, υποθέτοντας ότι $C_1 = C_2 = C$, και $R_{b1} = R_{b2} = R_b$ και $R_{C1} = R_{C2} = R_C$. Ο χρόνος κατά τον οποίο μένει αγωγίμο το κάθε τρανζίστορ είναι, κατά προσέγγιση, της τάξης του $0.7 R_b C$. Επομένως η περίοδος επαναλειτουργίας θα είναι: $T = 2(0.7 R_b C)$ ενώ η συχνότητα λειτουργίας είναι:

$$f = \frac{1}{T} = \frac{1}{1,4R_b C}$$

Στο κύκλωμα που αναφέραμε παρατηρούνται δύο σταθερές χρόνου: η μία είναι η $R_b C$ με την οποία εκφορτίζονται οι πυκνωτές C_1 και C_2 , και έτσι μεταβάλλεται η τάση στις βάσεις των τρανζίστορ, και η άλλη είναι η, $R_C C$ με την οποία μεταβάλλεται η τάση στους συλλέκτες των ίδιων τρανζίστορ. Πρέπει να ληφθεί υπόψη ότι άλματα τάσης, χωρίς παρεμβολή των σταθερών χρόνου, έχουμε για μεν τις τάσεις στους συλλέκτες, όταν τα αντίστοιχα τρανζίστορ δεν άγουν (ανοιχτός διακόπτης), για δε τις βάσεις όταν τα αντίστοιχα τρανζίστορ άγουν.

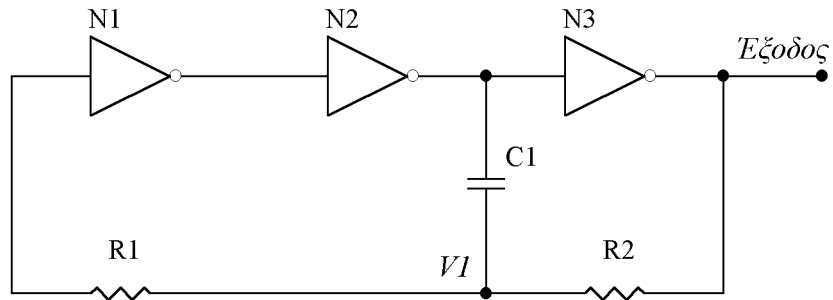


Σχήμα 5.29. Διάγραμμα μεταβολής τάσεων σε ασταθή πολυδονητή

5.8.2. Ασταθής πολυδονητής με πύλες

Ένα σχετικά απλό κύκλωμα, που βασίζεται και αυτό σε πύλες CMOS, απεικονίζεται στο Σχήμα 5.30. Η πύλη NOT N3, παρέχει από την έξοδό της ένα ορθογώνιο σήμα το οποίο μπορεί να χρησιμοποιηθεί για το χρονοισμό των οποιονδήποτε κυκλωμάτων ακολουθούν. Αν υποθέσουμε ότι η έξοδος της πύλης αυτής βρίσκεται σε λογικό «1», τότε η είσοδος της θα βρίσκεται σε λογικό «0», με αποτέλεσμα ο πυκνωτής C1 να φορτίζεται μέσω της R2. Μόλις η τάση στο σημείο V1 (κόμβος R1, R2, C1) φθάσει στο μισό περίπου της τάσης εξόδου της N3 τότε η N1 θα οδηγήσει την έξοδό της σε λογικό «0», αναγκάζοντας την N2 να επιβάλλει στην είσοδο της N3 λογικό «1». Μόλις όμως συμβεί αυτό, η έξοδος της N3 θα οδηγηθεί σε λογικό «0», προκαλώντας την εκφόρτιση του πυκνωτή C1.

Αμέσως μετά την εκφόρτιση αρχίζει ο επόμενος κύκλος φόρτισης έτσι ώστε το κύκλωμα αυτό να συμπεριφέρεται πάντα σαν ταλαντωτής. (Κοσσιδάς, 1996; Δεληγιάννης, 2004)



Σχήμα 5.30. Ασταθής πολυδονητής με πύλες

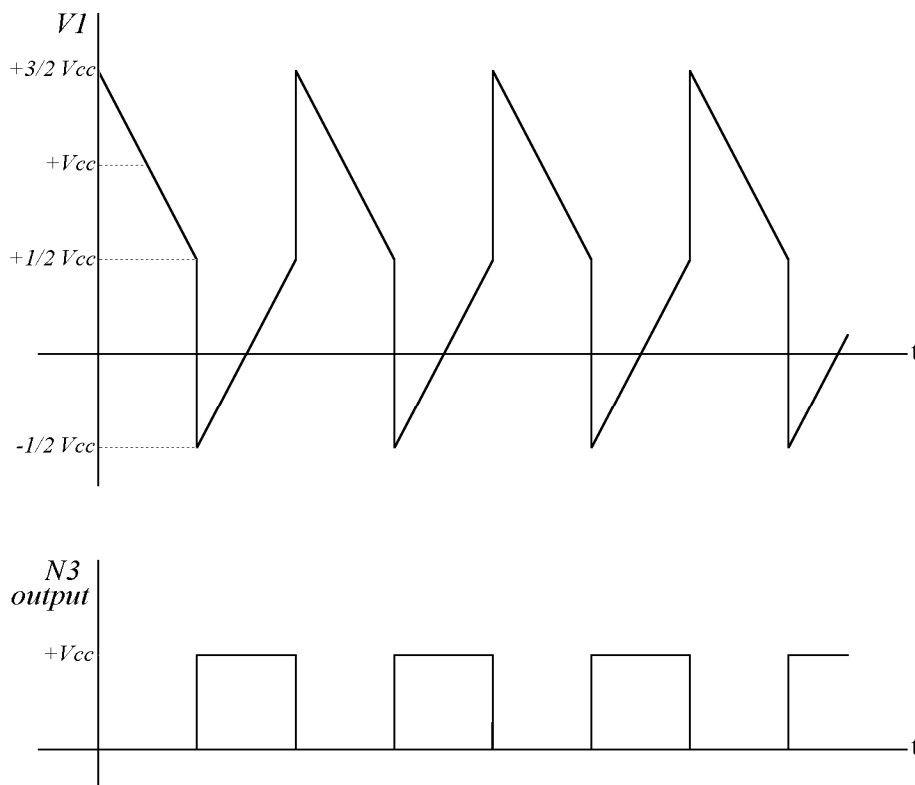
Για το κύκλωμα αυτό ισχύουν οι παρακάτω σχέσεις, που καθορίζουν κατά προσέγγιση τη συχνότητα λειτουργίας του. (Η τιμή R αντιπροσωπεύει την τιμή του παράλληλου συνδυασμού R1, R2).

$$\text{Av } R1 = R2 \quad f \cong \frac{0,559}{R \cdot C_1}$$

$$\text{Av } R1 \gg R2 \quad f \cong \frac{0,455}{R \cdot C_1}$$

$$\text{Av } R1 \ll R2 \quad f \cong \frac{0,722}{R \cdot C_1}$$

Στο Σχήμα 5.31 αποδίδονται οι κυματομορφές που υφίστανται στον κόμβο V1 και στην έξοδο του κυκλώματος. Παρατηρήστε ότι όλες οι αλλαγές καταστάσεων πραγματοποιούνται γύρω από το 50% της τάσης τροφοδοσίας, όπως, επίσης, και τις υπερτάσεις που προκαλούνται από τη φόρτιση-εκφόρτιση του πυκνωτή C1. Οι υπερτάσεις αυτές αποσβένονται στο εσωτερικό των πυλών, με τη βοήθεια των διόδων προστασίας που υπάρχουν στις εισόδους των πυλών NOT.



Σχήμα 5.31. Διαγράμματα μεταβολής τάσεων σ' ένα ασταθή πολυδονητή με πύλες

Το κύκλωμα του παραπάνω ταλαντωτή είναι εξαιρετικά σταθερό έναντι των μεταβολών της τάσης τροφοδοσίας. Αυτό είναι αναμενόμενο, αφού τα κατώφλια αλλαγής κατάστασης ορίζονται παραμετρικά ως προς αυτήν. Η σταθερότητα αυξάνεται στις χαμηλότερες συχνότητες, αφού σ' αυτές η καθυστέρηση μετάδοσης (propagation delay) της κάθε πύλης αποτελεί ένα μικρό κλάσμα της συνολικής περιόδου ταλάντωσης.

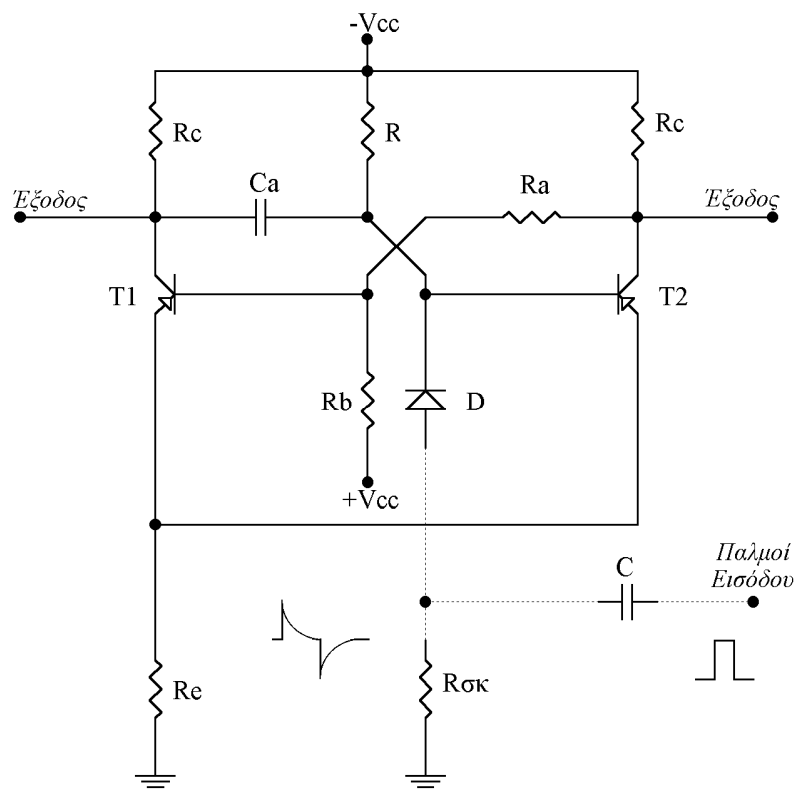
Περαιτέρω βελτίωση της σταθερότητας θα έχουμε αν η αντίσταση R2 γίνει αρκετά μεγάλη, ώστε να αντισταθμίζει οποιοσδήποτε αλλαγές της αντίστασης εξόδου των αναστροφών.

5.8.3. Μονοσταθής πολυδονητής

Ο μονοσταθής πολυδονητής (Monostable multivibrator) χρησιμοποιείται και αυτός αρκετά στα κυκλώματα χρονισμού. Στο Σχήμα 5.32 δίνεται το κύκλωμα ενός μονοσταθούς πολυδονητή (πολυδονητής μιας κατάστασης) καθώς και το κύκλωμα εφαρμογής των παλμών σκανδαλισμού (διακεκομμένες γραμμές). (One shot)

Ο μονοσταθής πολυδονητής έχει μία μόνο σταθερή κατάσταση λειτουργίας. Στο Σχήμα 5.32, που ακολουθεί, θεωρούμε ότι το T₂ είναι σε κατάσταση αγωγιμότητας, ενώ το T₁ σε κατάσταση αποκοπής. Στις καταστάσεις αυτές παραμένουν, μέχρις ότου επιβληθεί στο κύκλωμα κάποια εξωτερική διέγερση. Εφαρμόζοντας έναν παλμό στη βάση του τρανζίστορ T₂, αυτό παύει να άγει, προκαλώντας την αγωγιμότητα του T₁ για ένα μικρό χρονικό διάστημα, που καθορίζεται από το ίδιο το κύκλωμα (συχνά είναι κλάσμα του msec). Στη συνέχεια επανέρχεται στην προηγούμενη σταθερή κατάσταση (αγωγιμότητα του T₂), στην οποία και παραμένει, έως ότου εφαρμοσθεί ο επόμενος παλμός. (Nashelsky, 1994; Κοσσίδης, 1996; Δεληγιάννης, 2004 Sedra, & Smith, 2015)

Το κύκλωμα σκανδαλισμού εργάζεται με τον ίδιο τρόπο, όπως δηλαδή αναφέρθηκε προηγουμένως για το δισταθή πολυδονητή. Η παρουσία ενός παλμού διέγερσης στη βάση του τρανζίστορ είναι αναγκαία για ένα χρονικό διάστημα τόσο, ώστε να προφθάσει να οδηγηθεί σε αγωγιμότητα το T₁. Ο εν λόγω θετικός παλμός εφαρμόζεται δια μέσου της D τόσο στη βάση του T₂, όσο και στον C_a, μεταβάλλοντας έτσι το φορτίο του και, συνεπώς, την τάση που αναπτύσσεται στα άκρα του. Το αποτέλεσμα αυτής της ενέργειας θα είναι η αγωγιμότητα του T₁, και η οδήγηση σε κατάσταση αποκοπής του τρανζίστορ T₂.

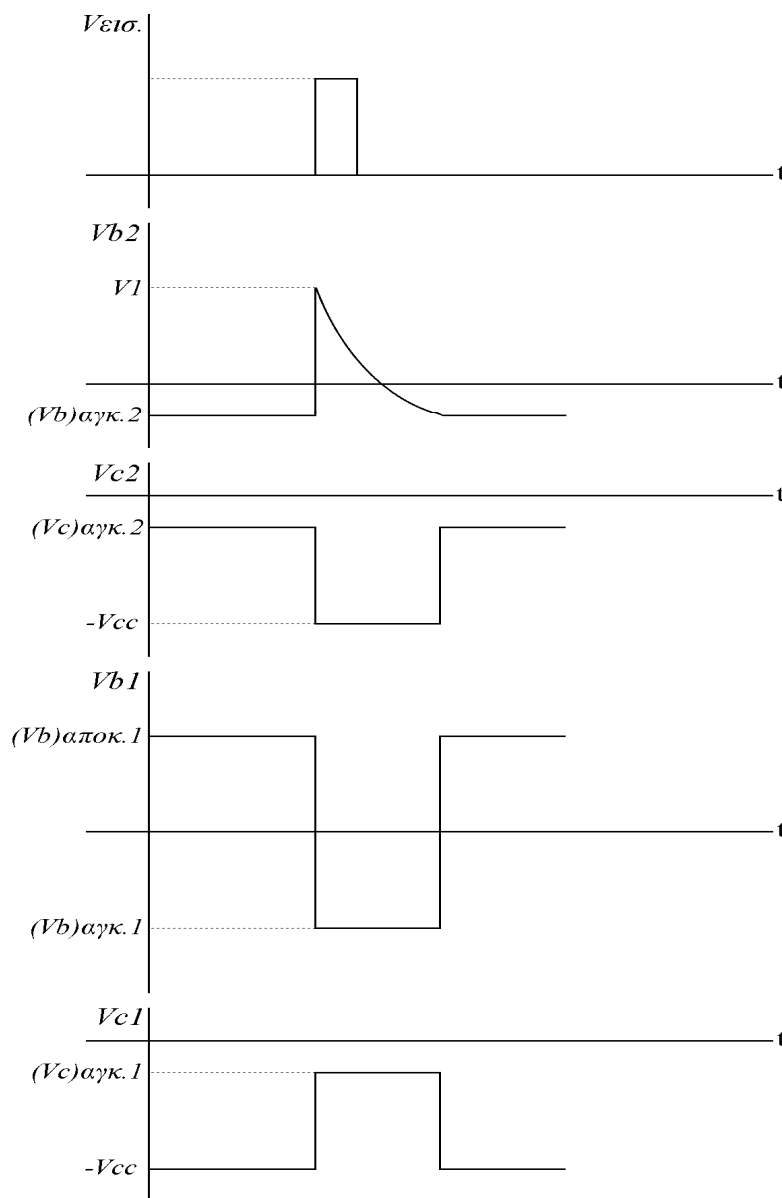


Σχήμα 5.32. Μονοσταθής πολυδονητής

Κατά τη διάρκεια της αγωγιμότητας του T_1 , ο πυκνωτής C_a εκφορτίζεται μέσω του T_1 με σταθερά χρόνου ίση με $T = (R + R_e) \cdot C_a$. Όταν η τιμή της τάσης του γίνει ίση με $(V_{be})_{αγκ.2}$, αυτόματα παύει να άγει το T_1 και επανέρχεται σε αγωγιμότητα το T_2 . Ο κύκλος αυτός θα επαναλαμβάνεται αδιάκοπα, εφόσον εφαρμόσουμε παλμούς στο κύκλωμα σκανδαλισμού.

Στο Σχήμα 5.33 δίνονται αναλυτικά οι κυματομορφές στα διάφορα σημεία του κυκλώματος, έχοντας σαν σήμα αναφοράς τον παλμό που επιβάλλεται στην είσοδο του κυκλώματος διέγερσης. Η τάση εισόδου προκαλεί την αύξηση της τάσης στη βάση του τρανζίστορ T_2 . Αυτό έχει σαν αποτέλεσμα την οδήγηση του συλλέκτη του ίδιου τρανζίστορ σε στάθμη ίση μ' αυτήν της πηγής $-V_{cc}$. Την ίδια επίδραση θα δεχθεί και η βάση του T_1 το οποίο, αντιδρώντας στη διέγερσή του, θα οδηγήσει τον συλλέκτη σε δυναμικό ίσο μ' αυτό της γης. Ο χρόνος εναλλαγής των τρανζίστορ εξαρτάται από τη σταθερά χρόνου εκφόρτισης του πυκνωτή C_a και γενικά είναι της τάξης $\sqrt{2} \cdot T$, όπου T η σταθερά χρόνου $(R + R_e) \cdot C_a$.

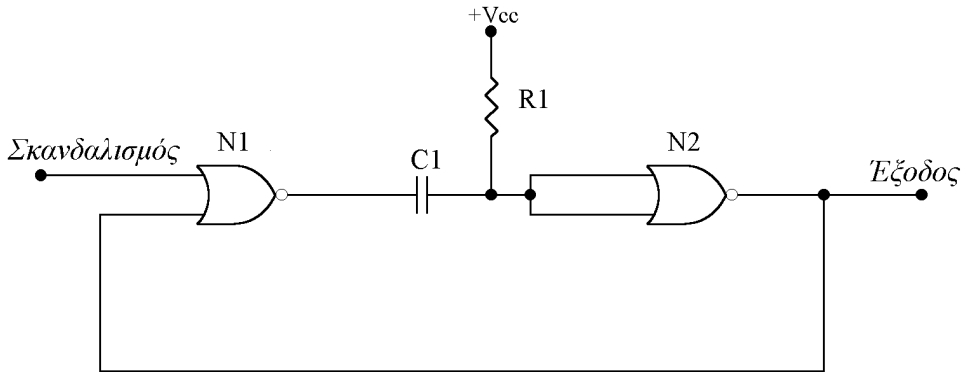
Τέλος θα πρέπει να ληφθεί υπόψη ότι ο πολυδονητής μιας κατάστασης χρησιμεύει για τη μεταβολή της διάρκειας ή και του εύρους κυματομορφών. Πράγματι, εφαρμόζοντας στην είσοδο του πολυδονητή σήμα μεταβλητού εύρους ή διάρκειας, λαμβάνουμε στην έξοδό του παλμό σταθερής διάρκειας εξαρτώμενου από τη σταθερά χρόνου T και σταθερού εύρους εξαρτώμενου από την τιμή της $-V_{cc}$.



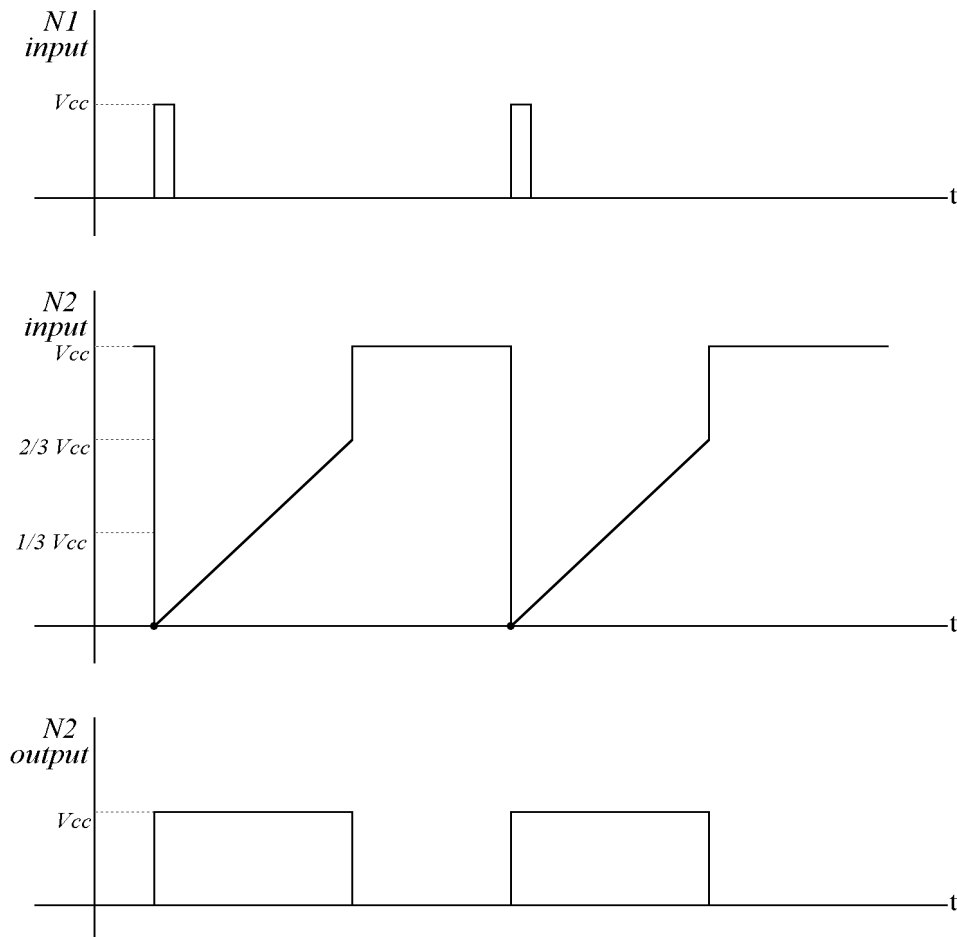
Σχήμα 5.33. Διαγράμματα μεταβολής τάσεων μονοσταθούς πολυδονητή

5.8.4. Μονοσταθής πολυδονητής με πύλες

Αν και τα τρανζίστρος αποτελούν μια εύκολη και φθηνή λύση για την κατασκευή πολυδονητών, τα ολοκληρωμένα κυκλώματα υπερτερούν προσφέροντας πιο εύχρηστες λύσεις. Στο Σχήμα 5.34 βλέπουμε ένα κύκλωμα που παράγει στην έξοδό του παλμούς συγκεκριμένης διάρκειας, χρησιμοποιώντας μόνο δύο πύλες NOR τύπου CMOS και ένα δικτύωμα πυκνωτή-αντίστασης. (Naselsky, 1994; Κοσσίδης, 1996; Δεληγιάννης, 2004)



Σχήμα 5.34. Μονοσταθής πολυδονητής με πύλες



Σχήμα 5.35. Διαγράμματα μεταβολής τάσεων σ' ένα μονοσταθή πολυδονητή με πύλες

Στην κατάσταση ηρεμίας, η έξοδος της πύλης N_2 βρίσκεται σε λογικό «0», ενώ η ίδια κατάσταση επιβάλλεται και στην είσοδο σκανδαλισμού της πύλης N_1 . Η έξοδος της N_1 θα βρίσκεται κάτω από αυτές τις

συνθήκες σε λογικό «1», διατηρώντας τον πυκνωτή C_1 αφόρτιστο και επιβεβαιώνοντας ότι η έξοδος της N_2 θα βρίσκεται σε λογικό «0».

Μόλις ένας θετικός παλμός μικρής διάρκειας φθάσει στην είσοδο σκανδαλισμού της N_1 , η έξοδος της οδηγείται αμέσως σε λογικό «0», αναγκάζοντας τον πυκνωτή να κινηθεί, μέσω της R_1 , το μέγιστο δυνατό ρεύμα. Στις βραχυκυκλωμένες τότε εισόδους της πύλης N_2 θα εμφανισθεί λογικό «0», το οποίο με τη σειρά του, θα προκαλέσει την ανατροπή της κατάσταση της εξόδου της και φυσικά την ανατροφοδότηση της N_1 με λογικό «1».

Η έξοδος της πύλης N_1 , λόγω της ανατροφοδότησης, θα μένει σε λογικό «0» ανεξάρτητα από το αν ο παλμός σκανδαλισμού εξακολουθεί να υφίσταται ή όχι, αφού τροφοδοτείται πλέον με λογικό «1» από τη δεύτερη είσοδό της.

Η κατάσταση αυτή θα υφίσταται για όσο χρονικό διάστημα η τάση στις εισόδους της N_2 είναι μικρότερη από την τάση κατωφλίου λογικού «1», την τάση, δηλαδή, εκείνη που απαιτείται, ώστε να αλλάξει κατάσταση η έξοδος και να γίνει ίση με λογικό «0». Και εδώ ισχύει πως ο χρόνος που θα παραμένει η έξοδος ενεργή θα είναι ίσος με $\sqrt{2} \cdot T$, όπου T η σταθερά χρόνου $R_1 \cdot C_1$.

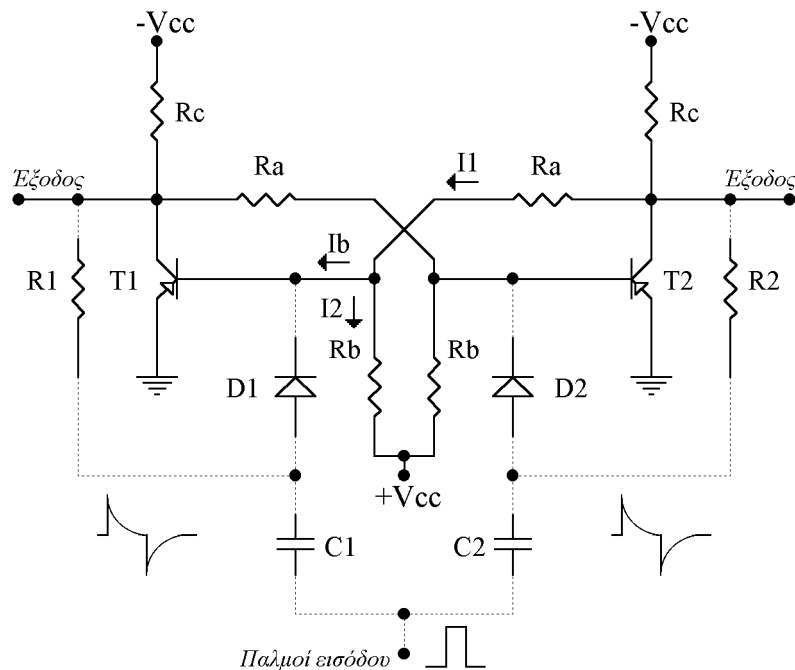
Μόλις παρέλθει το συγκεκριμένο χρονικό διάστημα, η έξοδος της N_2 θα οδηγηθεί σε λογικό «0», αναγκάζοντας και την έξοδο της N_1 να οδηγηθεί, εφόσον δεν υπάρχει παλμός σκανδαλισμού, σε λογικό «1». Ο πυκνωτής τότε, έχοντας ίδιο δυναμικό στα άκρα του, εκφορτίζεται ώστε να είναι έτοιμος για την επόμενη φόρτιση.

Όλα τα παραπάνω παριστάνονται αναλυτικά στο διάγραμμα του Σχήματος 5.35, στο οποίο αποδίδονται οι κυματομορφές στις εισόδους και στις εξόδους των πυλών. Παρατηρήστε ότι σαν τάσεις κατωφλίου για τις πύλες CMOS έχουν οριστεί τάσεις ίσες με το $1/3$ και τα $2/3$ της τάσης τροφοδοσίας τους.

5.8.5. Δισταθής πολυδονητής

Τα $f-f$ χρησιμοποιούνται ευρέως σε ηλεκτρονικά υπολογιστικά συστήματα, όπως σε κυκλώματα καταγραφής (Registers), κυκλώματα ολίσθησης (Shift Registers) κ.λπ. Στο Σχήμα 5.36 δίνεται δισταθής πολυδονητής (πολυδονητής δύο καταστάσεων), (Bistable multivibrator ή $f-f$) κατασκευασμένος με τρανζίστορ τύπου PNP.

Με τις συνεχείς γραμμές παριστάνεται το κύκλωμα του δισταθούς πολυδονητή, ενώ με τις διακεκομμένες γραμμές το κύκλωμα σκανδαλισμού, το οποίο θα εξηγήσουμε παρακάτω. (Κοσσίδης, & Γιαννακόπουλος, 2006)



Σχήμα 5.36. Πολυδονητής δύο καταστάσεων

Υποθέτουμε ότι το κύκλωμα βρίσκεται σε μια τέτοια κατάσταση, ώστε το τρανζίστορ T_1 να βρίσκεται στον κόρο (να είναι δηλαδή πλήρως αγωγίμο), ενώ το T_2 σε κατάσταση αποκοπής. Το κύκλωμα θα παραμένει στην κατάσταση αυτή, εφόσον δεν επιβληθούν κάποιες εξωτερικές διεγέρσεις, που θα προκαλέσουν τη μεταβολή του. Φυσικά το ίδιο κύκλωμα θα μπορούσε να ηρεμεί, έχοντας το τρανζίστορ T_2 σε κατάσταση αγωγιμότητας και το T_1 σε αποκοπή. Τη μονιμότητα των δύο καταστάσεων την εξασφαλίζει ο τρόπος σύνδεσης των δύο τρανζίστορ μεταξύ τους. Εδώ η σύνδεση γίνεται με τη βοήθεια αντιστάσεων, ενώ, όπως θα δούμε στη συνέχεια, εάν η μία ή και οι δύο αντιστάσεις αντικατασταθούν με πυκνωτές, θα έχουμε μόνο μία ή καμία σταθερή κατάσταση αντίστοιχα.

Όπως φαίνεται από το σχήμα, το κύκλωμα είναι συμμετρικό. Επομένως η μελέτη του κυκλώματος, όταν είναι αγωγίμο το T_1 θα είναι ακριβώς η ίδια με την περίπτωση αγωγιμότητας του T_2 .

Πράγματι όταν άγει το T_1 , η τάση στο συλλέκτη του θα είναι της τάξης των δεκάτων του Volt και αρνητική (V_{ce}) $\alpha\gamma\kappa_1 = -0,5 \text{ V}$. Αντίθετα η τάση στη βάση του T_2 , που θα προκαλεί την αποκοπή του, θα είναι θετική. Την τάση αυτή, όπως θα δούμε παρακάτω, μπορούμε να την υπολογίσουμε.

Η τάση στη βάση του T_1 θα είναι αρνητική της τάξης των δεκάτων του Volt (V_{be}) $\alpha\gamma\kappa_1 = -0,7 \text{ V}$, με ρεύμα βάσης I_b , που μπορεί να υπολογισθεί. Η τάση στον συλλέκτη του T_2 υπολογίζεται και αυτή εύκολα. Στο Σχήμα 5.37 φαίνονται σε ισοδύναμη μορφή τα δύο τμήματα του κυκλώματος του σχήματος 5.36, βάσει των οποίων μπορούν να γίνουν οι παραπάνω υπολογισμοί.

Για το T_1 έχουμε:

$$(V_{ce})\alpha\gamma\kappa_1 = -0,5 \approx 0$$

$$(V_{be})\alpha\pi\omicron\kappa_2 \approx \frac{R_a}{R_a + R_b} \cdot V_{cc} (V_{ce} = 0)$$

$$(V_{ce})\alpha\gamma\kappa_1 = -0,7 \text{ V}$$

Για το T_2 έχουμε:

$$I_b = I_1 - I_2$$

$$I_1 = \frac{V_{cc} - 0,7}{R_a + R_c}$$

$$I_2 = \frac{V_{cc} + 0,7}{R_b}$$

Από αυτό προκύπτει ότι:

$$(V_{ce})\alpha\pi\omicron\kappa_2 = -V_{cc} + R_c \cdot I_1 \approx -V_{cc} + R_c \cdot \frac{V_{cc}}{R_a + R_c} = \frac{-R_a}{R_a + R_c} \cdot V_{cc}$$

Από τους παραπάνω υπολογισμούς φαίνεται ότι η τιμή της R_b πρέπει να είναι $R_b \ll R_a$ για να είναι επαρκώς θετική η τιμή της τάσης αποκοπής στη βάση του T_2 , ώστε να εξασφαλίζεται η σταθερότητα της λειτουργίας του κυκλώματος. Επίσης η V_{ce} του T_1 πολώνει κατάλληλα τη βάση του T_1 , ώστε αυτό να διατηρείται σε αγωγιμότητα. Τέλος για την R_a καλό είναι να είναι να ισχύει $R_a > R_c$, ώστε η (V_{be}) $\alpha\gamma\kappa_1$ να δίνει ακόμη καλύτερη σταθερότητα στο κύκλωμα.

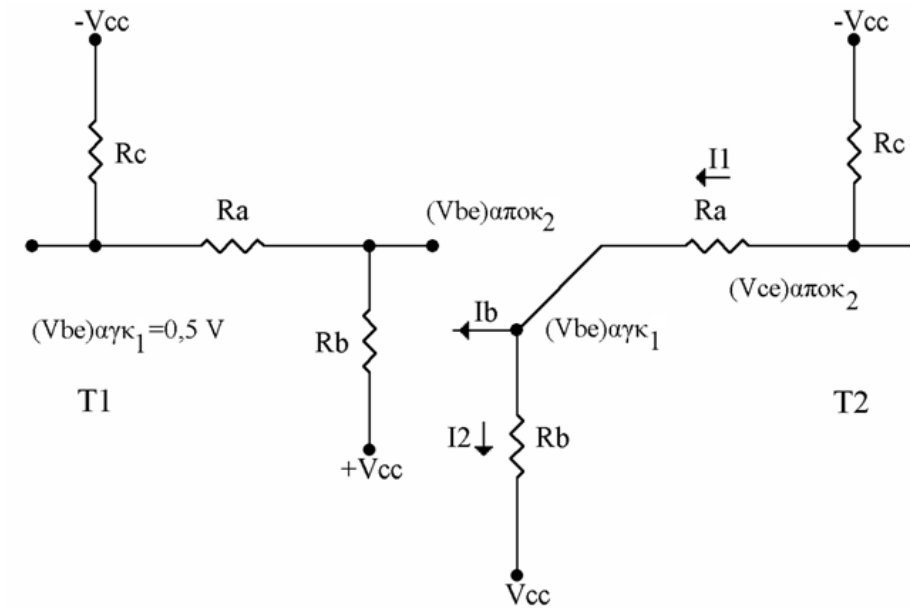
Στη συνέχεια θα εξετάσουμε το κύκλωμα σκανδαλισμού, που αποδίδεται με διακεκομμένες γραμμές στο Σχήμα 5.36.

Το κύκλωμα σκανδαλισμού χρησιμεύει για τη μετάβαση του κυκλώματος από τη μία κατάσταση λειτουργίας (αγωγιμότητα T_1) στην άλλη (αγωγιμότητα T_2). Η παρούσα σχεδίαση αποτελείται από δύο κυκλώματα διαφόρισης που σχηματίζονται από τα ζεύγη R_1C_1 και R_2C_2 . Εάν στην είσοδο του κυκλώματος σκανδαλισμού εφαρμόσουμε μια τετραγωνική τάση, στην έξοδο των κυκλωμάτων διαφόρισης, όπως ήδη γνωρίζουμε, λαμβάνουμε θετικές και αρνητικές τιμές τάσεων υπό μορφή αιχμών (βλέπε κυματομορφές στα άκρα των C_1 και C_2 στο Σχήμα 5.36).

Οι δίοδοι D_1 και D_2 έχουν τοποθετηθεί, έτσι ώστε να επιτρέπουν τη διέλευση μόνο των θετικών αιχμών που δημιουργούνται από τα ζεύγη RC (R_1C_1 ή R_2C_2) και την εφαρμογή τους στη βάση του T_1 ή του T_2 ανάλογα με την τάση με την οποία είναι φορτισμένοι οι πυκνωτές C_1 , C_2 .

Εφόσον συνδεθεί στο κύκλωμα η διάταξη σκανδαλισμού, υποθέτοντας ότι άγει το T_1 , ο C_1 θα φορτιστεί σε μία τιμή τάσης ίση με $-(V_{ce}) \alpha\gamma\kappa_1$ και ο C_2 σε μία τιμή ίση με (V_{ce}) $\alpha\pi\omicron\kappa_2$.

Εφαρμόζοντας τώρα το τετραγωνικό σήμα στην είσοδο, μόνο η θετική ημιπερίοδος αυτού θα μπορούσε να επιδράσει στις βάσεις των τρανζίστορ, λόγω της ύπαρξης των διόδων, και συγκεκριμένα μόνο σε μία από τις δύο βάσεις, αφού ο ένας από τους δύο πυκνωτές έχει τιμή τάσης τέτοια, ώστε να εξουδετερώνει το θετικό παλμό (λαμβάνεται τιμή σήματος εισόδου της τάξης της τάσης αποκοπής του τρανζίστορ).



Σχήμα 5.37. Τμήματα κυκλώματος για τον υπολογισμό των I_b και $(V_{ce})_{\alpha\pi\kappa_2}$.

Κατά συνέπεια, επιδρώντας ο παλμός στη βάση του T_1 , διακόπτει την αγωγιμότητά του, με αποτέλεσμα να αρχίζει να άγει το T_2 . Οι πυκνωτές C_1, C_2 θα πολωθούν τότε αντίστροφα λόγω της αλλαγής της αγωγιμότητας των τρανζίστορ. Συνεχίζοντας την εφαρμογή τετραγωνικών παλμών στην είσοδο, θα έχουμε σαν αποτέλεσμα τη μεταπήδηση από τη μία κατάσταση λειτουργίας στην άλλη. Από τις εξόδους λαμβάνονται πάντοτε αντίθετες τιμές τάσεων, που εξισώνονται με τις λογικές τιμές «1» και «0». (Κοσσίδας, 1996; Wakerly, 2006; Mano, & Ciletti, 2014; Jaeger, & Blalock, 2011; Sedra, & Smith, 2015).

5.9. Ασκήσεις–Ερωτήσεις

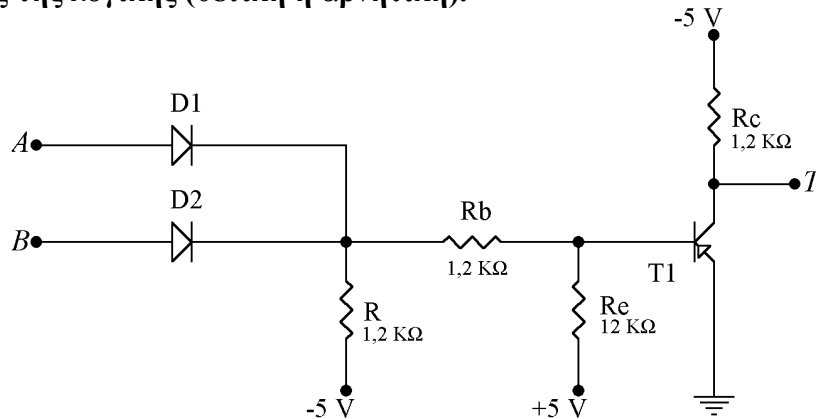
1. Να σχεδιασθεί και μελετηθεί κύκλωμα πύλης NAND θετικής λογικής με δύο εισόδους, με τρανζίστορ NPN έχοντας γνωστό ότι:
 - α) Τα τρανζίστορ συνδέονται σε σειρά.
 - β) Η τάση τροφοδοσίας είναι +12 Volt.
 - γ) Το ρεύμα κόρου $I_b=60\mu A$.
 - δ) Το ρεύμα I_c αποκοπής είναι 1 mA.Θεωρούνται γνωστές, επίσης, οι χαρακτηριστικές λειτουργίας των τρανζίστορ.
Σημείωση: Συνιστάται στον αναγνώστη να αρχίσει τη μελέτη του κυκλώματος από τον πίνακα αλήθειας της πύλης, χρησιμοποιώντας τιμές τάσεων στη θέση των μεταβλητών.
2. Κάνοντας χρήση διόδων και θετικής λογικής, να σχεδιάσετε και να μελετήσετε πύλη AND τριών εισόδων, έχοντας γνωστό ότι η μεγαλύτερη τάση εξόδου για το λογικό «0» είναι 0,1 Volt. Δίνονται επίσης:
 - α) η πηγή τροφοδοσίας 10 Volt και
 - β) η χαρακτηριστική της προς χρήση διόδου, όπως την προσδιορίζει ο παρακάτω πίνακας.

V(Volt)	I(mA)
0,1	3
0,2	8
0,3	20
-5	-0,1
-10	-0,18
-15	-0,25

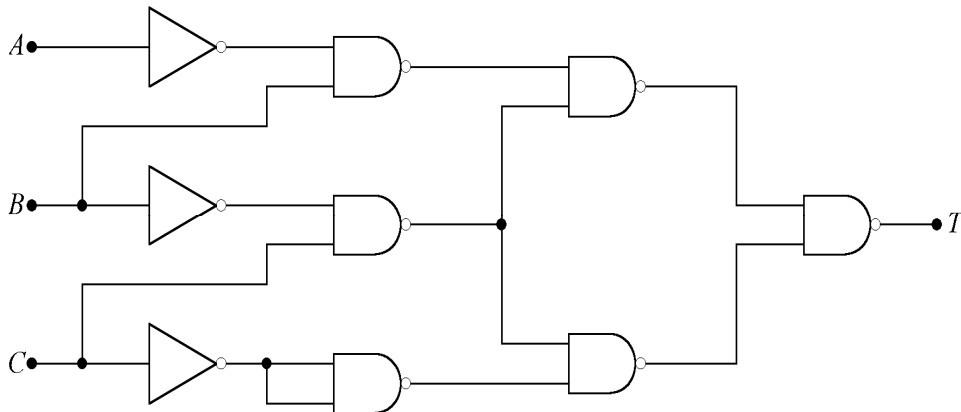
Πίνακας της I-V χαρακτηριστικής της διόδου

3. Να μελετήσετε και να σχεδιάσετε πύλη OR θετικής λογικής με διόδους. Η πύλη λειτουργεί σε χώρο σταθερής θερμοκρασίας 27°C, η δυναμική αντίσταση ορθής φοράς στο σημείο λειτουργίας είναι 25 Ω και το πηλίκο του ρεύματος ορθής φοράς προς το ρεύμα διαρροής της διόδου είναι $I/I_s=10^4$. Δίνονται επίσης η τάση τροφοδοσίας -10 Volt, καθώς και ο γνωστός από την ηλεκτρονική τύπος:
 $I/I_s=e^{(qV_0/KT)} -1$ όπου:
I : ρεύμα ορθής φοράς
 I_s : ρεύμα διαρροής (με ανάστροφη πόλωση)
q : φορτίο ηλεκτρονίου ($1,6 \times 10^{-19}$ C)
 V_0 : πτώση τάσης στη δίοδο κατά την ορθή πόλωση
K : σταθερά Boltzman ($1,38 \times 10^{-23}$ Joule/°K)
T : απόλυτη θερμοκρασία
4. Να μελετήσετε κύκλωμα αποτελούμενο από τέσσερις διακόπτες, που ενεργοποιούνται αντίστοιχα από τέσσερις θύρες αυτοκινήτου και από μια ενδεικτική λυχνία. Το κύκλωμα εργάζεται, έτσι ώστε, εάν μία ή περισσότερες από τις τέσσερις θύρες είναι ανοικτές (διακόπτες κλειστοί), να ανάβει η ενδεικτική λυχνία. Κατά τη μελέτη του κυκλώματος να υπολογισθεί η λογική συνάρτηση και να γίνει κάθε δυνατή απλοποίηση. Για τη σχεδίαση του λογικού κυκλώματος να χρησιμοποιηθούν πύλες AND, OR και να γίνει επαλήθευση της λειτουργίας του.
5. Για την παρακάτω λογική πύλη, που είναι βασισμένη στη λειτουργία με τρανζίστορ και διόδους, βρείτε:
 - α) Τον αντίστοιχο πίνακα αλήθειας
 - β) Το είδος της πύλης που παριστάνει

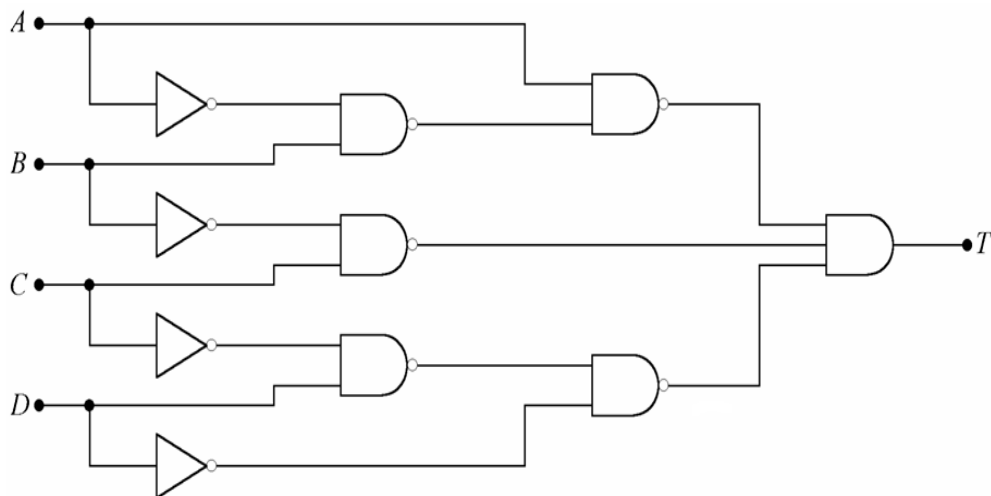
γ) Το είδος της λογικής (θετική ή αρνητική).



6. Να μελετήσετε και να σχεδιάσετε το κύκλωμα σύγκρισης δύο διψήφιων δυαδικών αριθμών X και Y. Το αποτέλεσμα της σύγκρισης να καταγραφεί στις εξόδους του κυκλώματος κατά τρόπο, ώστε να είναι δυνατή η αναγνώριση του =, < ή >.
7. Να δοθεί ο πίνακας αλήθειας, η λογική συνάρτηση και το λογικό κύκλωμα διάταξης η οποία δέχεται σαν είσοδο δύο διψήφιους δυαδικούς αριθμούς A_1A_0 , B_1B_0 και δίνει έξοδο 1 μόνο όταν οι δύο δυαδικοί αριθμοί είναι ίσοι.
8. Να βρεθούν οι λογικές συναρτήσεις και να γίνουν τυχόν απλοποιήσεις για τα παρακάτω λογικά κυκλώματα:

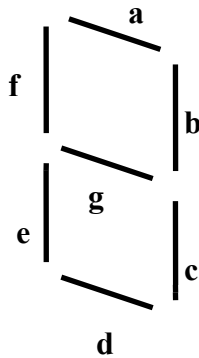


Κύκλωμα 8α



Κύκλωμα 8β

9. Να δώσετε το λογικό κύκλωμα διάταξης τριών εισόδων κατασκευασμένων με πύλες NAND, NOT, ώστε να επαληθεύεται η λογική εξίσωση: $T = A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot \overline{C}$
10. Να σχεδιασθεί λογικό κύκλωμα για την οδήγηση ψηφίου απεικόνισης (LED display), το οποίο αποτελείται από επτά φωτεινά τμήματα, έτσι ώστε να είναι δυνατός ο σχηματισμός ενός από τα ψηφία 0 έως 9. Η μορφή του ψηφίου και τα τμήματα που το αποτελούν φαίνεται στο παρακάτω σχήμα. Σαν παράδειγμα αναφέρουμε ότι για τη δημιουργία του αριθμού 3 πρέπει να ενεργοποιηθούν τα τμήματα a, b, c, d, g, για τη δημιουργία του αριθμού 4 τα τμήματα b, c, f, g κ.ο.κ.



11. Να σχεδιαστεί λογικό κύκλωμα στην είσοδο του οποίου να έχουμε τους δυαδικούς ισοδύναμους των δεκαδικών ψηφίων από 0-9 και στην έξοδο να έχουμε 1, μόνο όταν ο δυαδικός εισόδου ισοδυναμεί με άρτιο δεκαδικό.
12. Να σχεδιαστεί λογικό κύκλωμα με τέσσερις εισόδους και μία έξοδο. Η έξοδος να ισούται με ένα(1) στις παρακάτω περιπτώσεις: α) όταν όλες οι εισόδους είναι 1 β) όταν καμία από τις εισόδους δεν είναι 1 γ) όταν περιττός αριθμός εισόδων ισούται με 1.
13. Να σχεδιαστεί λογικό κύκλωμα στην είσοδο του οποίου θα έχουμε ένα δυαδικό αριθμό τριών δυαδικών ψηφίων και στην έξοδο θα έχουμε σαν αποτέλεσμα το τετράγωνο του αριθμού αυτού.
14. Να σχεδιαστεί λογικό κύκλωμα με τέσσερις γραμμές εισόδου (παριστάνουν ένα δεκαδικό ψηφίο) και τέσσερις γραμμές εξόδου. Στις γραμμές της εξόδου να δίδεται σαν αποτέλεσμα το συμπλήρωμα του αριθμού της εισόδου ως προς 2.
15. Να σχεδιάσετε ένα πλήρη αφαιρέτη με δύο ημιαφαιρέτες και μία πύλη Η.
16. Να αποδείξετε πώς μπορούμε να μετατρέψουμε ένα πλήρη αθροιστή σε πλήρη αφαιρέτη με την προσθήκη κυκλώματος αντιστροφής.
17. Να σχεδιάσετε με πύλες NAND έναν πλήρη αθροιστή χρησιμοποιώντας τις γνωστές συναρτήσεις: $C = xy + xz + yz$ και $S = \overline{C}(x + y + z) + xyz$
18. Να σχεδιαστεί ψηφιακό σύστημα το οποίο να αποτελείται από δύο συνδυαστικά κυκλώματα. Το πρώτο συνδυαστικό κύκλωμα εκτελεί τον πολλαπλασιασμό δύο δυαδικών αριθμών των δύο bits ο καθένας. Το αποτέλεσμα του πολλαπλασιασμού (P4 P3 P2 P1) είναι είσοδος του δευτέρου συνδυαστικού κυκλώματος του οποίου η έξοδος είναι 1 μόνο όταν το αποτέλεσμα του πολλαπλασιασμού είναι 6 ή 9.
19. Να σχεδιαστεί λογικό κύκλωμα με οκτώ εισόδους και μία έξοδο. Η έξοδος να γίνεται 1 μόνο όταν στην είσοδο των οκτώ δυαδικών ψηφίων εμφανίζεται πληροφορία η οποία έχει ακριβώς τρεις διαδοχικές μονάδες.
20. Σε έναν μετατροπέα BCD-σε επτά στοιχεία(seven-segment), γιατί πρέπει να χρησιμοποιηθεί ένας μετατροπέας κωδίκων;
 Α. για να μετατραπεί το 4-bit BCD σε 7-bit κώδικα
 Β. για να μετατραπεί το 4-bit BCD σε 10-bit κώδικα
 Γ. για να μετατραπεί το 4-bit BCD σε κώδικα Gray
 Δ. Δε χρειάζεται καμία μετατροπή

Αναφορές-Βιβλιογραφία

- Δεληγιάννης†, Θ. (2004). *Ηλεκτρονικά Αναλογικά και Ψηφιακά*, Ίδρυμα Ευγενίδου
- Κοσσίδης, Α.Θ. (1996). *Σχεδίαση Ψηφιακών Κυκλωμάτων*, Εκδόσεις Μπένοϋ
- Κοσσίδης, Α.Θ., Γιαννακόπουλος, Π., (2006), *Αριθμητικά Συστήματα και Ψηφιακά Κυκλώματα*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα
- Haykel Ben Jamaa, M., Mohanram, Kartik, De Micheli, Giovanni (2009). *Novel library of logic gates with ambipolar CNTFETs: Opportunities for multi-level logic synthesis*, Proceedings of the Conference on Design, Automation and Test in Europe, DATE '09
- Haykel Ben Jamaa, M. (2011). *Regular Nanofabrics in Emerging Technologies: Design and Fabrication Methods for NanoScale Digital Circuits*, Springer
- Jaeger, R., Blalock, T. (2011). *Microelectronics Circuit Design*, 4/e, Mc Graw Hill
- Katz, R. (2005). *Contemporary Logic Design*, 2/e, Prentice Hall
- Kvatinsky, Shahar, Satat, Guy, Wald, Nimrod (2012). *Logic Design with Memristors*, Israel Institute of Technology, Ανακτήθηκε 10/7/2015 από http://webee.technion.ac.il/people/skva/Memristor%20Projects/Memristor%20final%20report_Guy_Nimrod_Winter2012.pdf
- Mano, M., Ciletti, M. (2014). *Ψηφιακή Σχεδίαση*, 5^η έκδοση, Παπασωτηρίου
- Nashelsky, Louis (1994). *Introduction to Digital Technology*, 4th Ed., Prentice Hall
- Nelson, V., Nagle, H., Carroll, B., Irwin, J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall
- Sedra, A., Smith, Kenneth. (2015). *Microelectronic Circuits*, 7th edition, Oxford University Press
- Tocci, R. J., Widmer, N. S, and Moss, Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley
- Wakerly, J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Κεφάλαιο 6

Σύνοψη

Στο κεφάλαιο αυτό θα εξετάσουμε λεπτομερώς τον πολυδονητή δύο καταστάσεων ($f-f$), ο οποίος είναι βασικό στοιχείο στη μελέτη των ακολουθιακών κυκλωμάτων. Γίνεται περιγραφή όλων των τύπων των διασταθών πολυδονητών. Ακολουθεί η βασική περιγραφή των καταχωρητών, των απαριθμητών, της γεννήτριας ψευδοτυχαίων αριθμών, και δίνονται τα διαγράμματα καταστάσεων των ακολουθιακών κυκλωμάτων. Επίσης μελετώνται τα ακολουθιακά κυκλώματα και ο τρόπος λειτουργίας τους.

Ακολουθιακά κυκλώματα είναι τα λογικά κυκλώματα που εκτελούν μια σειρά από διαδικασίες και, στη συνέχεια, επανέρχονται στην αρχική τους κατάσταση. Στα ακολουθιακά κυκλώματα, η επόμενη κατάσταση εξόδου δεν είναι συνάρτηση μόνο των μεταβλητών εισόδου, που εμείς έχουμε επιβάλει, αλλά και της προηγούμενης κατάστασης, που είχε η έξοδος τους. Τα ακολουθιακά κυκλώματα αποτελούνται από ένα λογικό συνδυαστικό κύκλωμα, που συνοδεύεται από μία διάταξη ανασύζευξης μεταξύ εξόδου και εισόδου.

Οι δύο κατηγορίες ακολουθιακών κυκλωμάτων είναι τα: α) σύγχρονα και β) ασύγχρονα.

Σύγχρονα ακολουθιακά κυκλώματα είναι εκείνα στα οποία η εφαρμογή των παλμών εισόδου, η εκτέλεση των διαφόρων λειτουργιών και η έξοδος του αποτελέσματος γίνονται σε τακτά χρονικά διαστήματα, ελεγχόμενα από ένα ρολόι χρονισμού (*clock*) του όλου συστήματος.

Ασύγχρονα ακολουθιακά κυκλώματα είναι εκείνα στα οποία οι διάφορες λειτουργίες του κυκλώματος δεν εκτελούνται σε τακτά χρονικά διαστήματα (συγκεκριμένους αριθμούς παλμών) από τη στιγμή εφαρμογής των παλμών εισόδου, αλλά η ταχύτητα εκτέλεσης εξαρτάται από αυτό καθαυτό το κύκλωμα.

Το κεφάλαιο αυτό περιλαμβάνει ένα σύνολο αναλυτικά λυμένων ασκήσεων είτε ακολουθιακών κυκλωμάτων είτε κυκλωμάτων με τη χρήση μηχανών καταστάσεων και ολοκληρώνεται με την περιγραφή και τη σχεδίαση μετρητών με μηχανές καταστάσεων.

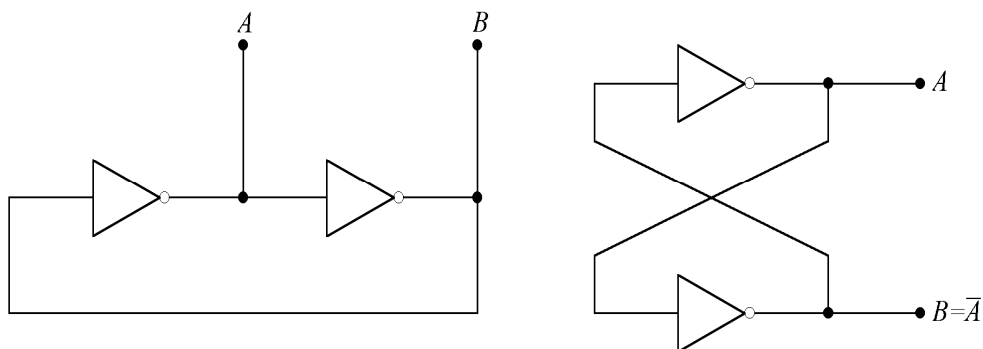
Προαπαιτούμενη γνώση

Η γνώση των βασικών πυλών και οι πίνακες αλήθειας τους.

6. Λογικά Ακολουθιακά Κυκλώματα

6.1. Διασταθείς πολυδονητές ($f-f$) με πύλες λογικής

Θα αναφερθούμε στη δυνατότητα κατασκευής $f-f$ ($F-F$) με λογικές πύλες. Εκτός του ήδη γνωστού $F-F$ (πολυδονητής δύο καταστάσεων), θα αναφερθούμε, στη συνέχεια, και στα άλλα χρήσιμα $F-F$ που θα πραγματοποιήσουμε με λογικά κυκλώματα.



Σχήμα 6.1. Πολυδονητής δύο καταστάσεων με λογικές πύλες

Στο σχήμα 6.1 βλέπουμε, κατασκευασμένο με πύλες λογικής NOT, το γνωστό από το προηγούμενο κεφάλαιο πολυδονητή δύο καταστάσεων ή διασταθή πολυδονητή, που πλέον θα ονομάζουμε *flip-flop* ($F-F$). Στο σχήμα αυτό έχει απεικονισθεί με δύο διαφορετικούς τρόπους. Ο πρώτος μοιάζει περισσότερο με το κύκλωμα του σχήματος 5.36., ενώ ο δεύτερος θα είναι εκείνος που θα χρησιμοποιούμε για την απεικόνισή του, από εδώ και στο εξής. (Δεληγιάννης, 2004; Nashelsky, 1994; Tocci, et al., 2010)

Θα εξετάσουμε τη λειτουργία αυτού του κυκλώματος. Αν υποθέσουμε ότι $A=1$, τότε $B=0$ δηλαδή $A = \overline{B}$. Αντίστοιχα, υποθέτοντας ότι $B=1$, έχουμε $A=0$, η έξοδος A είναι το συμπλήρωμα της B , και αντίστροφα. Βλέπουμε, επομένως, ότι με δυο πύλες άρνησης επιτυγχάνουμε τον πολυδονητή δυο καταστάσεων, που ήδη περιγράψαμε. Παρακάτω θα εξετάσουμε και άλλα είδη $F-F$ κατασκευασμένα με τη βοήθεια πυλών λογικής, τα οποία μπορούν να έχουν μια ή περισσότερες εισόδους διέγερσης.

6.2. Στοιχεία μνήμης (flip-flop)

Ακολουθεί η περιγραφή των στοιχείων μνήμης (Φραγκάκης 1975; Nashelsky, 1994; Nelson, et al., 1995; Κοσσιδάς, 1996; Givone 2002; Holdsworth, & Woods, 2002; Balch, 2003; Δεληγιάννης, 2004; Katz, 2005; Wakerly, 2006; Balabanian, & Carlson, 2007; Predko 2007; Tocci, et al. 2010; Mano, & Ciletti, 2014; Roth, & Kinney, 2014; Pritchard, 2015).

6.2.1. R-S flip-flop

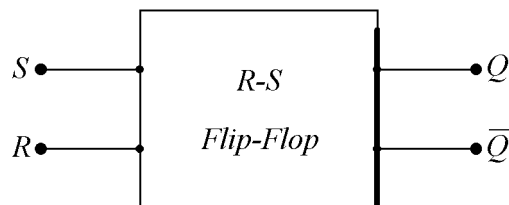
Το R-S (Reset- Set) $f-f$ διαθέτει δύο εισόδους, την R και την S , και δύο εξόδους την Q και την \overline{Q} . Στο Σχήμα 6.2 δίνεται η σχηματική παράσταση του R-S $f-f$ και στον πίνακα 6-1 η συνάρτηση εξόδου (επόμενη κατάσταση) του $f-f$ (Q_{next}), λαμβανομένων υπόψη των εισόδων R , S και της προηγούμενης τιμής της εξόδου Q .

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

ή

R_n	S_n	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	;

Πίνακας 6.1. Πίνακας αλήθειας του R-S $f-f$



Σχήμα 6.2. Δομικό διάγραμμα του R-S $f-f$

Από τον πίνακα 6.1 προκύπτει ότι, όταν το S (set) πάρει την τιμή «1», η έξοδος γίνεται $Q_{next}=1$, ενώ, όταν το R (reset) γίνει «1», η έξοδος γίνεται $Q_{next}=0$. Η περίπτωση όπου και οι δυο εισόδους είναι «1», δηλαδή $R=S=1$, δεν είναι επιτρεπτή, και το αποτέλεσμα είναι ακαθόριστο. Τέλος όταν $R=S=0$, τότε $Q_{next}=Q$.

Η μαθηματική έκφραση της συνάρτησης εξόδου μπορεί να υπολογισθεί τόσο από τον πίνακα 6.1 όσο και από τον πίνακα 6.2, στον οποίο παριστάνεται με τη βοήθεια του χάρτη Karnaugh.

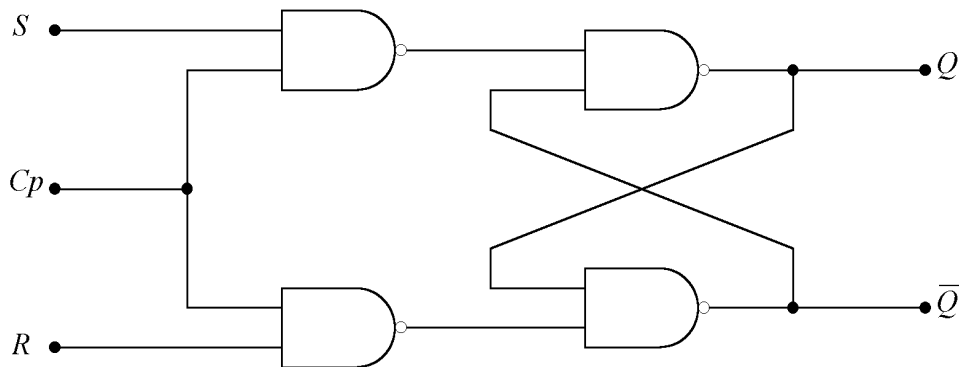
		RS			
		00	01	11	10
Q	0		1		
	1	1	1		

Πίνακας 6.2. Χάρτης Karnaugh του R-S $f-f$

Από το χάρτη Karnaugh προκύπτει ότι: $Q_{next} = \bar{R} \cdot S + \bar{R} \cdot Q$. Κατά συνέπεια, η συμπληρωματική έξοδος \bar{Q}_{next} θα είναι: $\bar{Q}_{next} = R + \bar{S} \cdot \bar{Q}$ (προκύπτει εύκολα από το χάρτη Karnaugh, παίρνοντας τους μηδενικούς όρους ή παίρνοντας το συμπλήρωμα της Q_{next} και κάνοντας χρήση του θεωρήματος απορρόφησης)

Στο σχήμα 6.3 παριστάνεται το R-S f-f, όπως αυτό κατασκευάζεται με λογικές πύλες. Όπως εύκολα γίνεται αντιληπτό, αποτελείται από 4 πύλες NAND.

Πράγματι, όταν R=1 και S=0, θα έχουμε $Q_{next}=0$. Επίσης, όταν S=1 και R=0, θα έχουμε $Q_{next}=1$, ενώ, όταν R=S=0, η έξοδος παραμένει στην ίδια κατάσταση, δηλαδή $Q_{next}=Q$. Τέλος, αν R=S=1, θα έχουμε Q_{next} απροσδιόριστο. Ο χρόνος εφαρμογής των παλμών R και S καθορίζεται από τη συχνότητα των ωρολογιακών παλμών C_p .



Σχήμα 6.3. Αναλυτικό διάγραμμα του R-S f-f με πύλες NAND

6.2.2. J-K flip-flop

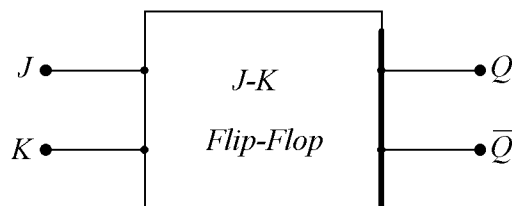
Το J-K f-f έχει δύο εισόδους, την J και την K, και δύο εξόδους την Q και την \bar{Q} , οι οποίες είναι συμπληρωματικές. Στον πίνακα 6.3 δίνεται η συνάρτηση εξόδου του J-K f-f, που προκύπτει εύκολα με τη βοήθεια του πίνακα αλήθειας. Στο σχήμα 6.4 δίνεται το δομικό του διάγραμμα.

J	K	Q	Q_{next}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

ή

J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n

Πίνακας 6.3. Πίνακας αλήθειας του J-K f-f



Σχήμα 6.4. Δομικό διάγραμμα του J-K f-f

Ο χάρτης Karnaugh του J-K $f-f$ δίνεται στον πίνακα 6.4.

	JK			
Q	00	01	11	10
0			1	1
1	1			1

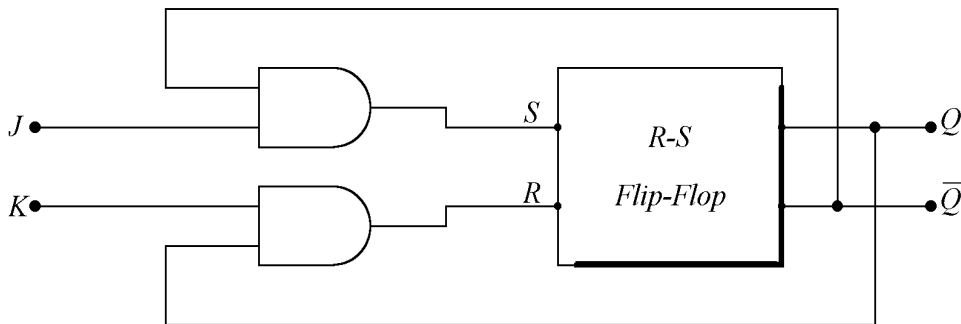
Πίνακας 6.4. Χάρτης Karnaugh του J-K $f-f$

Όπως φαίνεται από τον πίνακα αλήθειας, το J-K $f-f$ συμπεριφέρεται όπως το R-S $f-f$, εκτός των περιπτώσεων στις οποίες και οι δύο εισόδους είναι «1», και τότε έχουμε αντιστροφή της εξόδου. Το ίδιο συμβαίνει και με το T $f-f$ που θα μελετήσουμε στη συνέχεια.

Η συνάρτηση εξόδου, που προκύπτει από τον πίνακα Karnaugh, είναι η:

$$Q_{next} = \bar{K}Q + J\bar{Q}, \text{ ενώ η συμπληρωματική της είναι: } \bar{Q}_{next} = KQ + J\bar{Q} \text{ (από το χάρτη Karnaugh).}$$

Το J-K $f-f$ μπορεί να κατασκευασθεί με λογικές πύλες, όπως και το R-S $f-f$. Εδώ, βέβαια, έχουν χρησιμοποιηθεί δύο επιπλέον πύλες των δύο εισόδων αφού, εκτός από τις εισόδους που δέχονται τη διέγερση, απαιτούνται και δύο ακόμα που δέχονται την ανάδραση από τις εξόδους. Το κύκλωμα του συγκεκριμένου $f-f$ απεικονίζεται στο σχήμα 6.5.

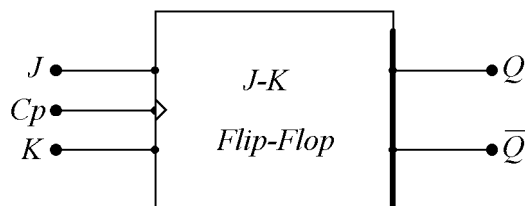


Σχήμα 6.5. Υλοποίηση του J-K $f-f$ με R-S $f-f$

Πράγματι όταν $J=K=0$ το Q_{next} παραμένει όπως έχει. Όταν $J=1$ και $K=0$, τότε $Q_{next} = 1$, διότι $R=0$ και $S=1$. Όταν είναι $J=0$ και $K=1$, τότε $Q_{next} = 0$. Τέλος για $J=K=1$ έχουμε $Q_{next} = \bar{Q}$.

6.2.3. Ωρολογιακό J-K flip-flop

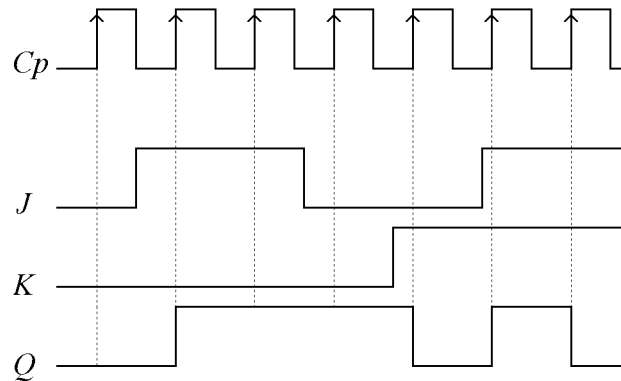
Ο πίνακας αλήθειας του ωρολογιακού J-K $f-f$ (clocked J-K $f-f$) διαφέρει αποτον αντίστοιχο του J-K $f-f$ ως προς το γεγονός ότι η αλλαγή μιας κατάστασης (διέγερση) γίνεται, μόνο όταν, ταυτόχρονα με τις εισόδους J-K, εφαρμόζουμε και παλμό στο ρολόι. Η διέγερση μπορεί να γίνει είτε με το θετικό μέτωπο των παλμών είτε με το αρνητικό, ανάλογα με τον τύπο του $f-f$.



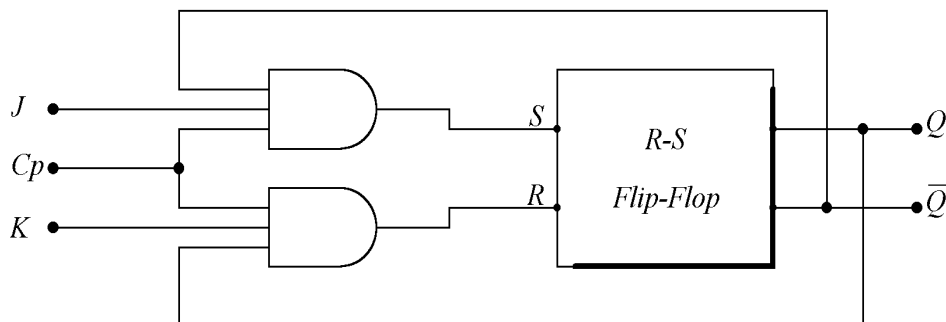
Σχήμα 6.6. Δομικό διάγραμμα ωρολογιακού J-K $f-f$

Στο σχήμα 6.6 δίνεται το δομικό διάγραμμά του και στο σχήμα 6.7 ο τρόπος λειτουργίας του. Παρατηρήστε, επίσης, στο σχήμα 6.8, στο οποίο φαίνεται ο τρόπος κατασκευής του ωρολογιακού J-K f-f, τη χρήση των πυλών NAND τριών εισόδων, που οδηγούν το βασικό R-S f-f. Αποτέλεσμα της παρουσίας τους είναι και η παρουσία του τρίτου σήματος ελέγχου, που έχει ονομασθεί CP.

Αναλυτικότερα, στο σχήμα 6.7, διαπιστώνουμε ότι, επειδή το f-f βασίζεται σε πύλες NAND, διεγείρεται από τα θετικά μέτωπα του ρολογιού. Όταν η είσοδος C_p βρίσκεται σε λογικό «0», οι καταστάσεις που υφίστανται στις εισόδους J και K δεν μπορούν να προκαλέσουν καμία αλλαγή στην έξοδο Q, διότι οι δύο πύλες NAND έχουν τις εξόδους καθηλωμένες σε λογικό «1». (Με βάση τον πίνακα 6.3)

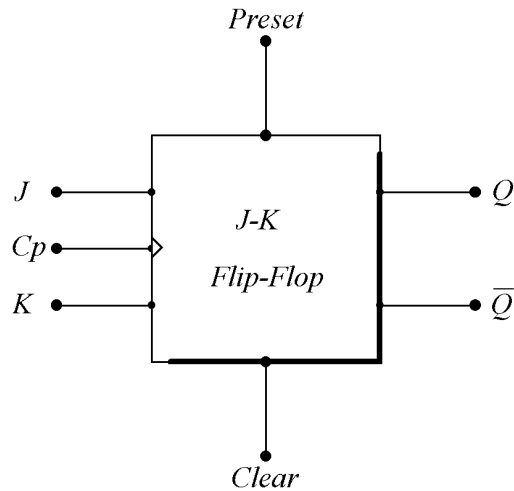


Σχήμα 6.7. Κατάσταση εξόδου ωρολογιακού J-K f-f



Σχήμα 6.8. Ωρολογιακό J-K f-f

Ένας άλλος τύπος J-K f-f είναι και αυτός που απεικονίζεται στο σχήμα 6.9. Ο τύπος αυτός του f-f περιλαμβάνει δύο ακόμη εισόδους: την preset και την clear. Με την είσοδο preset το f-f θέτει την έξοδο Q στην κατάσταση «1», ανεξάρτητα από τις τιμές των εισόδων J και K. Με την είσοδο clear θέτει την έξοδο Q στην κατάσταση «0», και στην περίπτωση αυτή, ανεξάρτητα από τις τιμές των εισόδων J και K.



Σχήμα 6.9. J-K f-f με εισόδους Preset και Clear

6.2.4. T flip-flop

Το T (Toggle) f-f συμπεριφέρεται ακριβώς όπως ο πολυδονητής δύο καταστάσεων. Έχει μία είσοδο και δύο εξόδους (Σχήμα 6.10). Η λειτουργία του δίνεται στον πίνακα 6.5, όπου T η είσοδος και Q η μία από τις δύο εξόδους του.

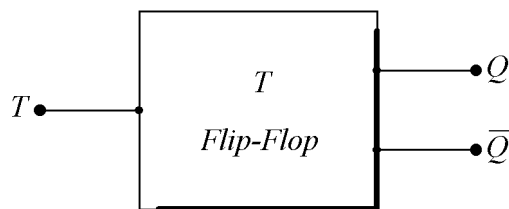
Όπως φαίνεται από τον πίνακα, το T f-f αλλάζει την τιμή εξόδου κάθε φορά που εφαρμόζουμε λογικό «1» (T=1) στην είσοδο.

T	Q	Q _{next}
0	0	0
0	1	1
1	0	1
1	1	0

ή

T	Q _{n+1}
0	Q _n
1	\bar{Q}_n

Πίνακας 6.5. Πίνακας αλήθειας του T f-f

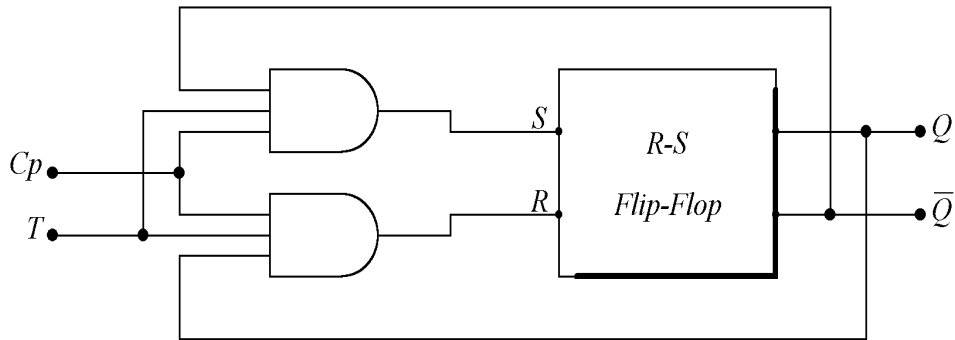


Σχήμα 6.10. Λογικό διάγραμμα του T f-f

Οι δύο εξοδοί έχουν πάντοτε τιμές συμπληρωματικές. Οι τιμές Q_{next} του πίνακα συμβολίζουν το αποτέλεσμα στην έξοδο μετά την εφαρμογή ή όχι παλμού. Η συνάρτηση μεταφοράς με βάση τον πίνακα είναι η:

$$Q_{next} = T \cdot \bar{Q} + \bar{T} \cdot Q \text{ και η συμπληρωματική της: } \bar{Q}_{next} = \bar{T} \cdot \bar{Q} + T \cdot Q = (\bar{T} + Q) \cdot (T + \bar{Q}) = T \cdot Q + \bar{T} \cdot \bar{Q}$$

Στο σχήμα 6.11 παριστάνεται το T f-f, κατασκευασμένο με πύλες και R-S f-f.



Σχήμα 6.11. T f-f με R-S f-f.

6.2.5. D flip-flop

Το **D** (Delay ή data) f-f έχει μία είσοδο και δύο εξόδους. Η λειτουργία αυτού δίνεται στον πίνακα 6.6, όπου **D** είναι η είσοδος και **Q** η μία από τις δύο εξόδους. Η άλλη έξοδος έχει τιμή συμπληρωματική ως προς την **Q** δηλαδή \bar{Q} .

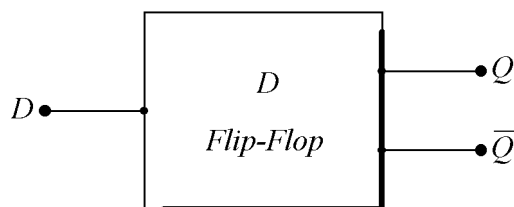
D	Q	Q _{next}
0	0	0
0	1	0
1	0	1
1	1	1

ή

D	Q _{n+1}
0	0
1	1

Πίνακας 6.6. Πίνακας αλήθειας του D f-f

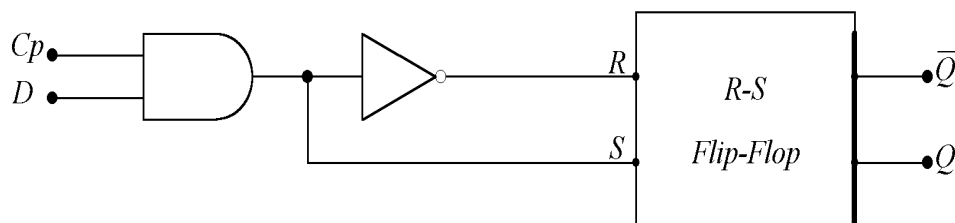
Όπως φαίνεται από τον πίνακα, στο D f-f η έξοδος Q_{next} ισούται με την είσοδο **D** μετά παρέλευση ορισμένου χρόνου. Ο χρόνος αυτός καθορίζεται με τη χρήση κατάλληλων σημάτων χρονισμού. Στο σχήμα 6.12 δίνεται το δομικό διάγραμμα του D f-f.



Σχήμα 6.12. Δομικό διάγραμμα του D f-f

Η συνάρτηση μεταφοράς είναι η: $Q_{next} = D$ και η συμπληρωματική της:

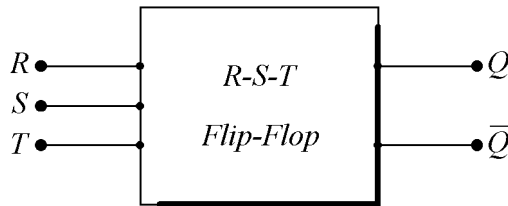
Στο σχήμα 6.13 δίνεται το D f-f, κατασκευασμένο με πύλες και R-S f-f. Παρατηρήστε την παρουσία της πύλης AND, στην οποία καταλήγει η πληροφορία που θέλουμε να μεταφερθεί στην έξοδο **Q**, μαζί με τον απαραίτητο για την καθυστέρηση παλμό ρολογιού.



Σχήμα 6.13. D f-f με R-S f-f

6.2.6. R-S-T flip-flop

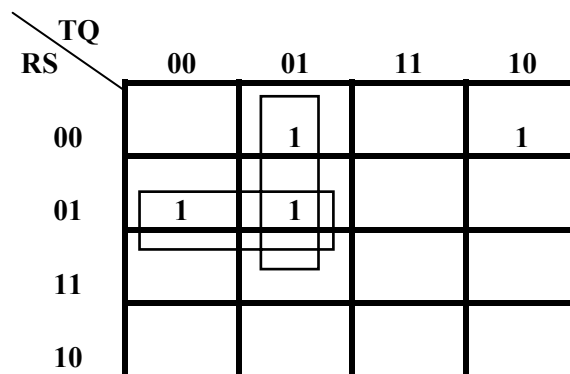
Το R-S-T *f-f* είναι ένα *f-f* τριών εισόδων, το οποίο έχει αντικατασταθεί από το J-K *f-f* σε όλες τις εφαρμογές του. Στον πίνακα 6.7 δίνεται η λειτουργία του, και στο σχήμα 6.14, το δομικό του διάγραμμα.



Σχήμα 6.14. Δομικό διάγραμμα του R-S-T *f-f*

R	S	T	Q	Q _{next}
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
1	0	0	0	0
1	0	0	1	0

Πίνακας 6.7. Πίνακας αλήθειας του R-S-T *f-f*



Πίνακας 6.8. Χάρτης Karnaugh του R-S-T *f-f*

Όπως φαίνεται στον πίνακα 6.7, η ταυτόχρονη εφαρμογή δύο παλμών στις εισόδους δεν είναι επιτρεπτή, δηλαδή RS=ST=RT=0. Οι έξοδοι του *f-f* είναι συμπληρωματικές, και, Βασικά, εργάζεται σα να ήταν συνδυασμός R-S και T *f-f*.

Η συνάρτηση εξόδου υπολογίζεται εύκολα από το χάρτη Karnaugh του πίνακα 6.8, από όπου προκύπτει: $Q_{next} = T \cdot \bar{Q} \cdot \bar{R} \cdot \bar{S} + \bar{T} \cdot Q \cdot \bar{R} + \bar{T} \cdot \bar{R} \cdot S$

Ο παραπάνω τύπος απλοποιείται προσθέτοντας τους μηδενικούς όρους $\mathbf{R \cdot S + S \cdot T + R \cdot T}$. Εκτελώντας τις πράξεις, ή χρησιμοποιώντας το χάρτη του πίνακα 6.8α, τελικά καταλήγουμε στις σχέσεις: $Q_{next} = \bar{T} \cdot Q \cdot \bar{R} + T \cdot \bar{Q} + S$ και $\bar{Q}_{next} = \bar{T} \cdot \bar{Q} \cdot \bar{S} + T \cdot Q + R$ (να το αποδείξετε με βάση τον πίνακα 6.8α)

		TQ			
RS		00	01	11	10
00			1		1
01		1	1	X	X
11		X	X	X	X
10				X	X

Πίνακας 6.8α. Χάρτης του R-S-T f-f (με χρήση των μη χρησιμοποιούμενων όρων)

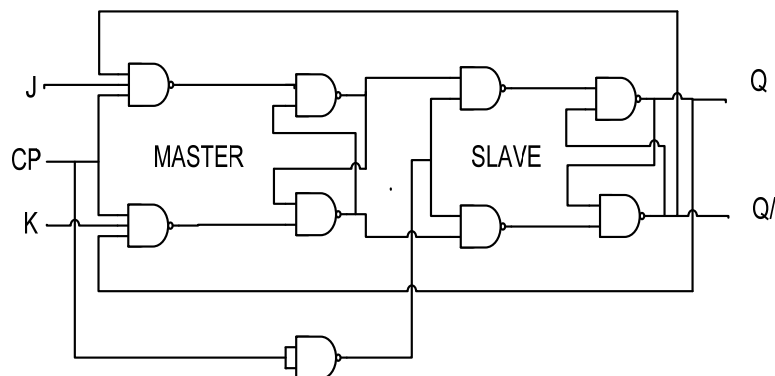
Εκτός από τα αναφερθέντα f-f σε πολλές εφαρμογές χρησιμοποιείται και το **master-slave** f-f (κύριο-εξαρτώμενο f-f).

6.2.7. Master slave flip-flop

Είναι ένα διπλό f-f, όπου με τον πρώτο παλμό του clock αποθηκεύεται η πληροφορία στο master, ενώ με το δεύτερο παλμό μεταφέρεται στο slave η πληροφορία που είχε αποθηκευτεί στο master. Σε όλα τα f-f που εξετάσαμε ως τώρα θα πρέπει ο χρόνος t_{on} , που ο παλμός ρολογιού είναι σε λογικό 1, να είναι μικρότερος από το συνολικό χρόνο καθυστέρησης διάδοσης του σήματος από την είσοδο ως την έξοδο του κυκλώματος t_{pd} ώστε οι νέες καταστάσεις των εξόδων, όταν εμφανισθούν, να μην επανατροφοδοτούνται στις πύλες εισόδου όσο διαρκεί ο παλμός ρολογιού. Σε αντίθετη περίπτωση το f-f θα περιπέσει σε αστάθεια αλλάζοντας διαρκώς τις καταστάσεις των εξόδων του έως ότου η είσοδος ρολογιού γίνει 0. Επίσης θα πρέπει η περίοδος του παλμού ρολογιού T, να είναι τέτοια ώστε ο επόμενος παλμός ρολογιού να εμφανισθεί αφού σταθεροποιηθεί η νέα κατάσταση των εξόδων.

Έτσι, για σωστή λειτουργία, πρέπει να ισχύει: $t_{on} < t_{pd} < T$. Αν δεν ισχύει η σχέση τότε, για παράδειγμα στο J-K f-f, όταν J=1 και K=1 και οι παλμοί του ρολογιού (CP=1) έχουν μεγάλη χρονική διάρκεια, τότε η κατάσταση του f-f αφού αντιστραφεί μία φορά, αντιστρέφεται συνεχώς σε όλη τη διάρκεια του παλμού του ρολογιού (δηλαδή μέχρι να γίνει CP=0). Γενικά σε μια τέτοια περίπτωση δεν ακολουθείται ο πίνακας αλήθειας και η έξοδος του f-f είναι απροσδιόριστη.

Η συνθήκη $t_{on} < t_{pd}$ δεν είναι πάντα εύκολο να ικανοποιηθεί και αυτό διότι ο χρόνος t_{pd} είναι πολύ μικρός όπως ήδη έχουμε δει. Για να αντιμετωπισθεί το πρόβλημα επινοήθηκαν τα Master-Slave f-f. Στο σχήμα 6.15 φαίνεται το κύκλωμα ενός Master-Slave J-K f-f.



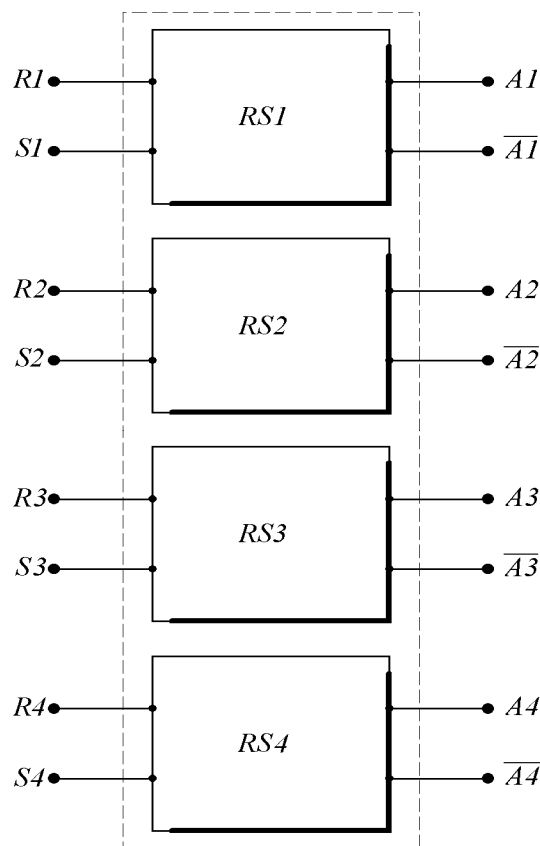
Σχήμα 6.15. Δομικό διάγραμμα του Master-Slave J-K f-f

Όπως φαίνεται στο σχήμα 6.15 το πρώτο f-f που ονομάζεται master ενεργοποιείται με το παλμό ρολογιού σε λογικό 1 ενώ το δεύτερο f-f που ονομάζεται slave ενεργοποιείται με το παλμό ρολογιού σε λογικό 0. Έτσι η αλλαγή κατάστασης εμφανίζεται στις εξόδους του slave αφού ο παλμός ρολογιού γίνει λογικό 0, και οι νέες καταστάσεις επανατροφοδοτούνται στο master f-f όταν σε αυτό ο παλμός ρολογιού είναι ήδη σε λογικό 0

με αποτέλεσμα να μην έχουμε νέες μεταβολές στις εξόδους του master, και επομένως και του slave, που είναι και οι εξόδοι όλου του Master-Slave $f-f$.

6.3. Καταχωρητές

Οι καταχωρητές (Registers) είναι ψηφιακά κυκλώματα στα οποία αποθηκεύονται πληροφορίες, και το μήκος τους είναι, συνήθως, ίσο μ' αυτό της μιας λέξης. Αποτελούνται από έναν αριθμό ανεξαρτήτων $f-f$, ίσο με τον αριθμό των δυαδικών ψηφίων που αποτελούν την κάθε λέξη. Κάθε $f-f$ είναι σε θέση να καταγράψει ένα, και μόνο ένα, δυαδικό ψηφίο. Για την εγγραφή ενός δυαδικού αριθμού με n ψηφία, χρειάζονται n $f-f$ s. Η πληροφορία, που καταγράφεται σ' έναν καταχωρητή, είναι πολύ εύκολο να ανακληθεί από αυτόν οποιαδήποτε στιγμή, αφού, αμέσως μετά την εγγραφή της, εμφανίζεται στις εξόδους Q των $f-f$ που τον αποτελούν. Οι καταχωρητές διατίθενται σε μορφή ολοκληρωμένου κυκλώματος και η κυριότερη χρήση τους γίνεται από τους ηλεκτρονικούς υπολογιστές, οι οποίοι τους χρησιμοποιούν για την αποθήκευση των αποτελεσμάτων των πράξεων που πραγματοποιούν. Το μήκος των καταχωρητών, που αποτελεί και ένα μέτρο αξιολόγησης των κεντρικών μονάδων επεξεργασίας (CPU) των υπολογιστών, μπορεί να είναι ίσο με 8, 16, 32, 64 ή ακόμη και 128 bit.



Σχήμα 6.16. Δομικό διάγραμμα καταχωρητή με R-S $f-f$

Τα χρησιμοποιούμενα $f-f$ που συνθέτουν έναν καταχωρητή είναι διαφόρων τύπων έτσι ώστε να καλύπτουν τις ιδιαίζουσες ανάγκες κάθε εφαρμογής. Στο σχήμα 6.16 δίνεται καταχωρητής με R-S $f-f$, ενώ είναι αρκετά συνηθισμένες οι περιπτώσεις καταχωρητών, κατασκευασμένων με $f-f$ τύπου D. Η εγγραφή της πληροφορίας στο εσωτερικό των καταχωρητών επιτυγχάνεται με την ταυτόχρονη επιβολή ενός παλμού ρολογιού στις ομώνυμες εισόδους όλων των $f-f$. (Κοσσιδάς, 1996; Floyd, 2006; Givone, 2002; Holdsworth, & Woods, 2002; Katz, 2005; Nashelsky, 1994; Predko 2007; Roth, & Kinney, 2014; Tocci, et al. 2010; Wakerly, 2006)

6.3.1. Καταχωρητές ολίσθησης

Οι καταχωρητές ολίσθησης (shift registers) είναι καταχωρητές των οποίων τα $f-f$ συνδέονται μεταξύ τους (εξόδοι με εισόδους) με τέτοιο τρόπο, ώστε το συνολικό περιεχόμενο του καταχωρητή να μπορεί να ολισθαίνει

προς τα δεξιά ή προς τα αριστερά, ανάλογα με τη συνδεσμολογία που έχει φτιαχτεί. Οι μετακινήσεις των δυαδικών ψηφίων, που έχουν καταχωρηθεί στα $f\text{-}f$, μπορούν να γίνουν τόσο αριστερά όσο και δεξιά, ανάλογα με την εκάστοτε εφαρμογή. Στην περίπτωση που η μετακίνηση είναι δυνατή και προς τις δύο διευθύνσεις, έχουμε τον λεγόμενο αμφίδρομο ολισθητή, του οποίου η εκάστοτε φορά ολίσθησης καθορίζεται με τη χρήση πρόσθετων κυκλωμάτων.

Στην περίπτωση ενός καταχωρητή δεξιάς ολίσθησης, κάθε φορά που πραγματοποιείται μια ολίσθηση, το περισσότερο σημαντικό ψηφίο μπορεί να «γεμίζει» με λογικό «0» ή με λογικό «1» ανάλογα με τη λογική κατάσταση που επιβάλλεται στην είσοδο του αριστερότερου $f\text{-}f$. Αν η λογική κατάσταση, που επιβάλλεται, είναι πάντοτε το λογικό «0», τότε η ολίσθηση αυτή ονομάζεται «λογική ολίσθηση». Αν στη θέση του περισσότερο σημαντικού ψηφίου επανεισάγεται (δηλαδή παραμένει) η ίδια κατάσταση, τότε έχουμε «αριθμητική ολίσθηση».

Στο πρώτο κεφάλαιο έχει γίνει πλήρης ανάλυση των αριθμητικών πράξεων. Έτσι η έκφραση «αριθμητική ολίσθηση» δηλώνει πως η ολίσθηση προς τα δεξιά ισοδυναμεί με την αριθμητική πράξη της διαίρεσης δια 2, ενός οποιουδήποτε προσημασμένου ως προς 2 αριθμού. Αν, για παράδειγμα, έχουμε τον αριθμό $00001000_{(2)} = 8_{(10)}$, τότε μετά από μια ολίσθηση προς τα δεξιά θα έχουμε τον αριθμό $00000100_{(2)} = 4_{(10)}$. Αν έχουμε τον $11000000_2 = -64_{(10)}$, και πραγματοποιήσουμε πάλι μια ολίσθηση προς τα δεξιά, τότε θα λάβουμε τον $11100000_{(2)} = -32_{(10)}$.

Με όμοιο τρόπο, γίνεται εύκολα αντιληπτό πως ολίσθηση προς τα αριστερά, με ταυτόχρονη εισαγωγή λογικού «0» στη θέση του λιγότερο σημαντικού ψηφίου, έχει σαν αποτέλεσμα τον πολλαπλασιασμό επί 2 του αριθμού που διατηρούσε αρχικά στις εξόδους του ο καταχωρητής ολίσθησης. Έτσι, αν για παράδειγμα, αρχικά ήταν καταχωρημένος ο αριθμός $00001000_{(2)} = 8_{(10)}$, τότε μετά από μια ολίσθηση θα λάβουμε τον $00010000_{(2)} = 16_{(10)}$.

Ένας άλλος τύπος ολισθητή είναι ο **κυκλικός ολισθητής**, στον οποίο το λιγότερο σημαντικό ψηφίο του καταχωρητή (δεξιά ολίσθηση) δε χάνεται, αλλά επιβάλλεται πάλι στην είσοδό του (αποκτά δηλαδή τη μέγιστη αξία) ή αντίστοιχα το περισσότερο σημαντικό ψηφίο (αριστερά ολίσθηση) επιβάλλεται στο δεξιότερο $f\text{-}f$ του καταχωρητή (γίνεται δηλαδή το λιγότερο σημαντικό). Ο πίνακας 6.9 δίνει συνοπτικά διάφορους τρόπους ολίσθησης στους καταχωρητές. Παρατηρείστε λοιπόν στα δύο πρώτα παραδείγματα με την κατακράτηση των ψηφίων ότι το τελικό περιεχόμενο περιέχει ακριβώς τα ψηφία που ολίσθησαν.

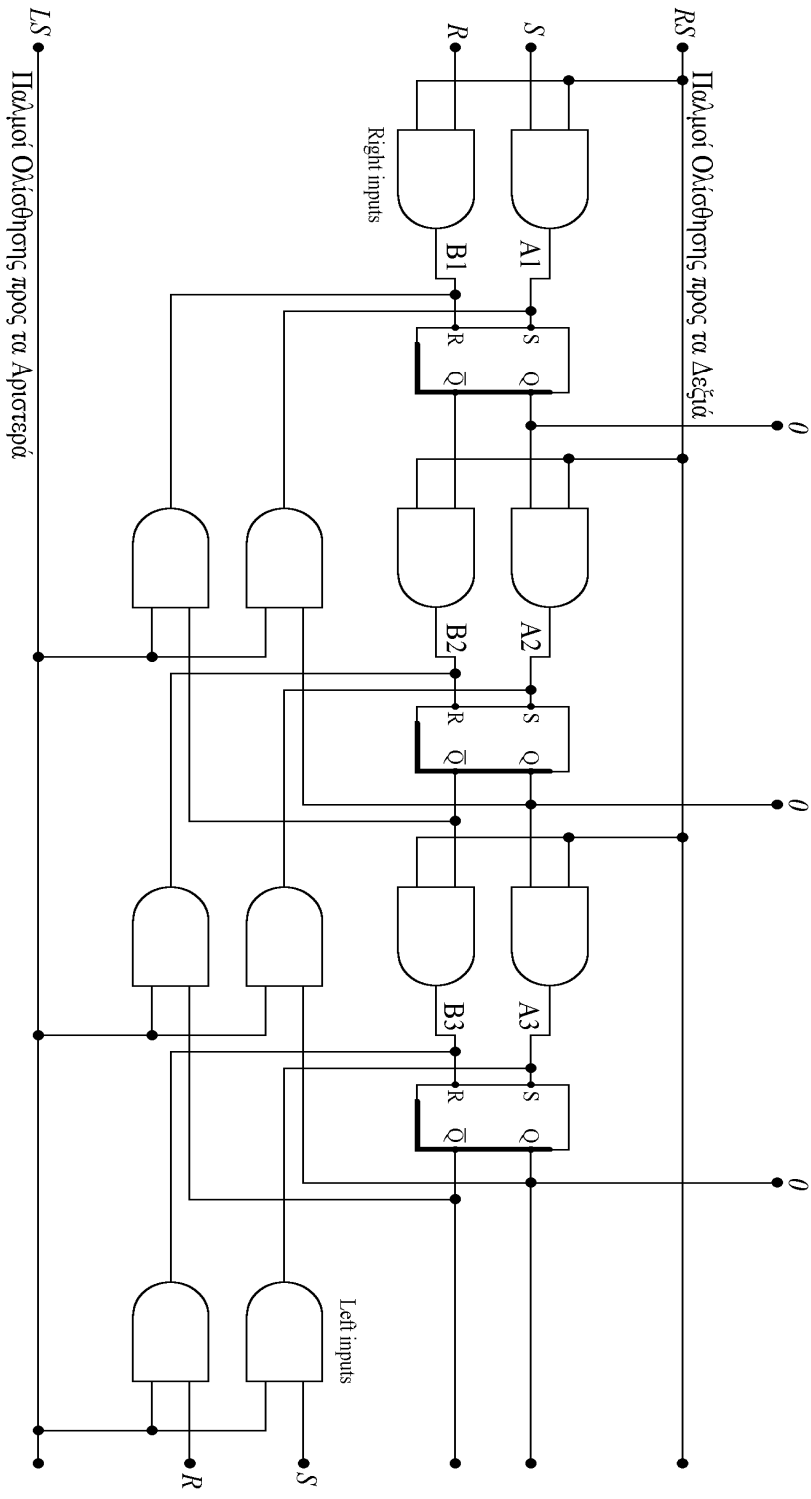
Αριθμός θέσεων ολίσθησης	3	2	1	4
Φορά	αριστερά	δεξιά	δεξιά	αριστερά
Κατακράτηση ψηφίων	ΝΑΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ
Περιεχόμενο προς ολίσθηση	01001010	11010111	01011110	10111110
Αποτέλεσμα ολίσθησης	01010010	11110101	00101111	11100000

Πίνακας 6.9. Τρόποι ολίσθησης

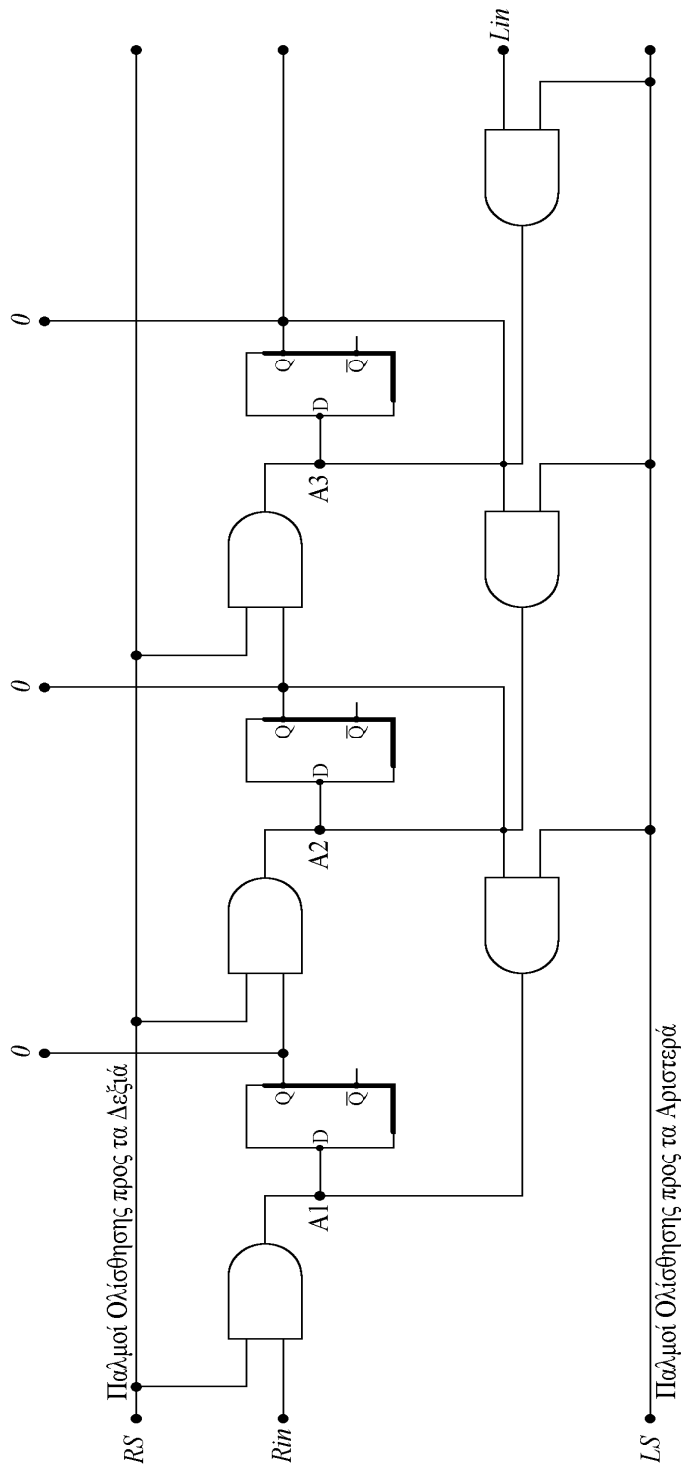
6.3.2. Ολισθητής με R-S flip-flop

Στο σχήμα 6.17 δίνεται ολισθητής με R-S flip-flop. Η ολίσθηση μπορεί να γίνει τόσο προς τα αριστερά, όσο και προς τα δεξιά, ανάλογα με το σημείο του κυκλώματος στο οποίο θα εφαρμοσθούν οι παλμοί ολίσθησης. Η σύνδεση των $f\text{-}f$ μεταξύ τους γίνεται κατά κανόνα με τη χρήση πυλών AND.

Το κατά πόσο θα εφαρμοσθεί παλμός στην είσοδο ενός συγκεκριμένου $f\text{-}f$ του σχήματος 6.17 εξαρτάται τόσο από τον παλμό ολίσθησης, όσο και από την τιμή της εξόδου του προηγούμενου $f\text{-}f$. Με λογικό «1» στην προηγούμενη έξοδο θα έχουμε επίδραση στο επόμενο $f\text{-}f$, ενώ με λογικό «0» ο επιβαλλόμενος παλμός στις γραμμές LS ή RS δεν προκαλεί κανένα αποτέλεσμα. Η διάρκεια εφαρμογής του παλμού ολίσθησης πρέπει να είναι μικρότερη του χρόνου εναλλαγής του $f\text{-}f$.



Σχήμα 6.17. Ολοθροής με R-S flip-flop. Τα σημεία A1, A2, A3, B1, B2, B3 αντιπροσωπεύουν κόμβους OR



Σχήμα 6.18. Ολισθητής με D flip-flop. Τα σημεία A1, A2, A3 αντιπροσωπεύουν κόμβους OR

Όπως φαίνεται από το σχήμα 6.17, συνδέοντας την έξοδο του τελευταίου $f-f$ με την είσοδο του πρώτου, επιτυγχάνουμε έναν κυκλικό ολισθητή δεξιάς ή αριστερής ολισθησης ανάλογα με τους εφαρμοζόμενους παλμούς. Το ίδιο κύκλωμα εργάζεται εξίσου καλά, αν αντικαταστήσουμε τα R-S $f-f$ με J-K $f-f$.

Ένας άλλος, εξίσου ενδιαφέρων, τύπος ολισθητή μπορεί να πραγματοποιηθεί με D $f-f$ και με ενδιάμεσες πύλες AND για την εφαρμογή των παλμών (Σχήμα 6.18).

6.4. Απαριθμητές

Οι απαριθμητές ή μετρητές, όπως αλλιώς ονομάζονται, είναι κυκλώματα καταχωρητών των οποίων το περιεχόμενο αυξομειώνεται κατά ένα, ανάλογα με τον εκάστοτε τύπο απαριθμητή. Οι απαριθμητές είναι δύο τύπων: (Nashelsky, 1994; Nelson, et al., 1995; Κοσσιδάς, 1996; Holdsworth, & Woods, 2002; Δεληγιάννης, 2004; Katz, 2005; Floyd, 2006; Wakerly, 2006; Predko, 2007; Balabanian, & Carlson, 2007; Roth, & Kinney, 2014; Tocci, et al., 2010; Mano, & Ciletti, 2014; Pritchard, 2015)

α) Δυαδικοί απαριθμητές:

Το περιεχόμενό τους αυξάνεται, ή ελαττώνεται, κατά μία μονάδα στο δυαδικό σύστημα αρίθμησης, ανάλογα με τους υπό εφαρμογή παλμούς και τις αντίστοιχες συνδέσεις. Η χρήση των απαριθμητών αυτών δίνει δυνατότητα απαρίθμησης μόνο με δυνάμεις του δύο.

β) Απαριθμητές ανάδρασης:

Το περιεχόμενό τους αυξάνεται, ή ελαττώνεται, κατά μια μονάδα, ανάλογα με το χρησιμοποιούμενο σύστημα αρίθμησης, με τη βοήθεια πυλών λογικής, που επιδρούν κατάλληλα στις βαθμίδες του απαριθμητή, έτσι ώστε να αλλοιώνονται περισσότεροι από μία έξοδοι του απαριθμητή. Σ' αυτήν την κατηγορία των απαριθμητών ανήκουν οι απαριθμητές άλματος, οι απαριθμητές ως προς το μέτρο (modulo) και οι απαριθμητές δακτυλίου.

6.4.1. Δυαδικοί Απαριθμητές

Οι δυαδικοί απαριθμητές (Binary Counters) είναι κυκλώματα καταχωρητών των οποίων το περιεχόμενο αυξάνεται ή ελαττώνεται, κατά μια μονάδα, στο δυαδικό σύστημα, κάθε φορά που στην είσοδό τους εφαρμόζεται ένας παλμός.

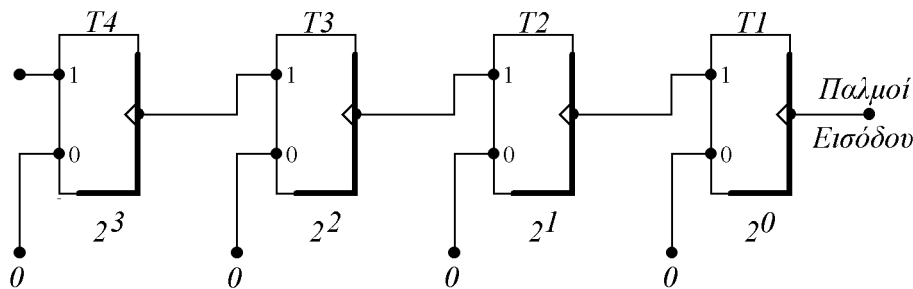
Οι απαριθμητές κατασκευάζονται από πολυδονητές δύο καταστάσεων (T - f - f) στην είσοδο των οποίων εφαρμόζονται οι προς απαρίθμηση παλμοί. Συνδέοντας διαδοχικά n f - f , μπορούμε να απαριθμήσουμε συνολικά 2^n παλμούς. Το πλήθος των f - f , που χρησιμοποιούνται, εξαρτάται από τις συγκεκριμένες ανάγκες της εφαρμογής.

Η απαρίθμηση γίνεται ως εξής:

α) Όλα τα f - f τοποθετούνται στη θέση Reset (μηδενισμός του απαριθμητή).

β) Εφαρμόζονται οι προς μέτρηση παλμοί, των οποίων ο αριθμός πρέπει να είναι $\leq 2^n$.

Στο σχήμα 6.19α, δίνεται ένας σειριακός δυαδικός απαριθμητής, αποτελούμενος από τέσσερα f - f τύπου T. Λόγω του συγκεκριμένου πλήθους των f - f , μπορεί να μετρήσει μέχρι 2^4 ψηφία, δηλαδή από το 0 έως και το 15.



Σχήμα 6.19α. Σειριακός δυαδικός απαριθμητής(αύξουσα τάξη)

Κάθε ένα από τα f - f διαθέτει δύο εξόδους, που ονομάζονται με 0 και 1. Παρατηρήστε ότι οι εξοδοί 0 αποτελούν τις εξόδους του απαριθμητή, ενώ οι εξοδοί 1 χρησιμοποιούνται για την οδήγηση των επόμενων f - f . Επειδή οι εξοδοί αυτές είναι συμπληρωματικές, είναι ευνόητο ότι, όταν, για παράδειγμα η έξοδος 0 του T1 είναι σε λογικό «0», η έξοδος 1 του ίδιου f - f θα βρίσκεται σε λογικό «1». Θεωρείται εξίσου προφανές πως μια θετική μετάβαση (από λογικό «0» σε λογικό «1») στην έξοδο 1 θα συνοδεύεται από μια αρνητική (από λογικό «1» σε λογικό «0») στην συμπληρωματική έξοδο 0.

Με την εφαρμογή του πρώτου παλμού η έξοδος του f - f T1 (σημειώνεται με 0) αλλάζει κατάσταση οδηγούμενη σε λογικό «1». Η αλλαγή αυτή δεν έχει καμιά επίδραση στο δεύτερο f - f , γιατί τη στιγμή εκείνη η έξοδος 1 του T1 οδηγείται σε λογικό «0», μια κατάσταση η οποία δεν είναι σε θέση να διεγείρει την είσοδο του T2. Θυμίζουμε ότι τα f - f αυτού του είδους διεγείρονται, όταν στην είσοδο τους επιβάλλεται μεταβολή από

«0» σε «1» και όχι το αντίθετο (βλέπε σχήμα 5.36). Κατά συνέπεια, όλες οι εξοδοί εκτός από αυτή του πρώτου $f-f$ παραμένουν αμετάβλητες.

Επιβάλλοντας ένα δεύτερο παλμό θα οδηγηθεί η έξοδος του T1 σε λογικό «0» με αποτέλεσμα την εμφάνιση στην είσοδο του δεύτερου $f-f$ μιας μεταβολής από «0» σε «1», που θα αναγκάσει την έξοδο του T2 να λάβει τιμή «1». Παρουσία τρίτου παλμού έχει σαν αποτέλεσμα την οδήγηση της εξόδου του T1 σε λογικό «1», ενώ η έξοδος του T2 θα εξακολουθεί να διατηρεί λογικό «1». Κατά συνέπεια, επιβάλλοντας αδιάκοπα παλμούς στο κύκλωμα, θα φθάσουμε στο σημείο όπου όλες οι εξοδοί θα γίνουν «1» (μέγιστος δυνατός αριθμός απαρίθμησης). Τότε η παρουσία ενός ακόμα παλμού θα μηδενίσει όλες τις εξόδους του απαριθμητή και η όλη διαδικασία θα αρχίσει και πάλι από την αρχή. Ο πίνακας 6.10 που ακολουθεί δίνει την κατάσταση όλων των εξόδων ανάλογα με τον εφαρμοζόμενο παλμό.

Παλμοί	Αρχική Κατάσταση	1 ^{ος}	2 ^{ος}	3 ^{ος}	4 ^{ος}	5 ^{ος}	6 ^{ος}	7 ^{ος}	8 ^{ος}	9 ^{ος}
Έξοδος F-F 4	0	0	0	0	0	0	0	0	1	1
Έξοδος F-F 3	0	0	0	1	1	1	1	1	0	0
Έξοδος F-F 2	0	0	1	1	0	0	1	1	0	0
Έξοδος F-F 1	0	0	0	1	0	1	0	1	0	1

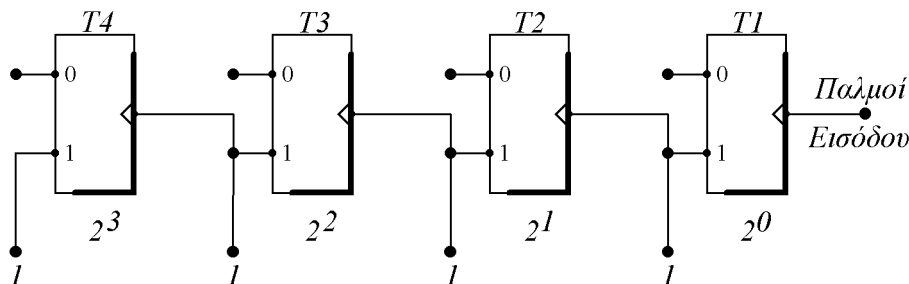
Παλμοί	10 ^{ος}	11 ^{ος}	12 ^{ος}	13 ^{ος}	14 ^{ος}	15 ^{ος}	16 ^{ος}	17 ^{ος}
Έξοδος F-F 4	1	1	1	1	1	1	0	0
Έξοδος F-F 3	0	0	1	1	1	1	0	0
Έξοδος F-F 2	1	1	0	0	1	1	0	0
Έξοδος F-F 1	0	1	0	1	0	1	0	1

Πίνακας 6.10. Καταστάσεις εξόδων δυαδικού απαριθμητή

Το κύκλωμα του σειριακού δυαδικού απαριθμητή μπορεί να χρησιμεύσει και σαν κύκλωμα άθροισης. Στέλνοντας στην είσοδο τόσους παλμούς όση είναι η τιμή του πρώτου όρου της πρόσθεσης και, στη συνέχεια τόσους παλμούς όση είναι η τιμή του δεύτερου όρου, θα έχουμε σαν αποτέλεσμα την εμφάνιση στις εξόδους του το άθροισμα των δύο όρων.

Στο Σχήμα 6.19β απεικονίζεται ένας απαριθμητής φθίνουσας τάξης, δηλαδή κάθε φορά που εφαρμόζουμε στην είσοδό του παλμούς σκανδαλισμού, το περιεχόμενό του ελαττώνεται κατά μια μονάδα.

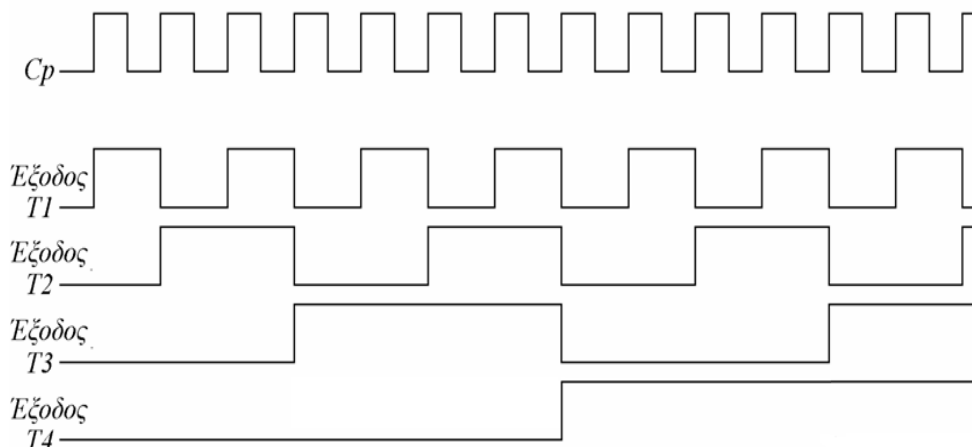
Ο απαριθμητής αυτός αποτελείται από T $f-f$ σε διαφορετική συνδεσμολογία από αυτήν του σχήματος 6.19α.



Σχήμα 6.19β. Σειριακός δυαδικός απαριθμητής (φθίνουσα τάξη)

Εάν υποθέσουμε ότι οι εξοδοί του απαριθμητή είναι όλες σε λογικό «1», εφαρμόζοντας παλμούς σκανδαλισμού στην είσοδο του πρώτου $f-f$, το περιεχόμενό του απαριθμητή θα ελαττώνεται κάθε φορά κατά μία μονάδα. Πρέπει να ληφθεί υπόψη ότι οι πολυδονητές (T $f-f$) είναι κατασκευασμένοι με τρανζίστορ PNP,

γεγονός που έχει σαν αποτέλεσμα να επενεργεί στις εισόδους τους μόνο το θετικό μέτωπο των παλμών διέγερσης.



Σχήμα 6.20. Υποβιβασμός συχνότητας με απαριθμητές

Τέλος σημειώνουμε ότι το ίδιο κύκλωμα μπορεί να χρησιμεύσει και για τον υποδιπλασιασμό συχνότητας. Πράγματι από τον προηγούμενο πίνακα, φαίνεται ότι η έξοδος του κάθε f - f αλλάζει, ακολουθώντας μια συγκεκριμένη τακτική. Το πρώτο f - f αλλάζει κατάσταση σε κάθε παλμό εισόδου, το δεύτερο κάθε δύο παλμούς εισόδου, το τρίτο σε κάθε τέσσερις παλμούς, ενώ το τέταρτο σε κάθε οκτώ. Προφανώς οι συχνότητες που θα αναδεικνύονται σ' αυτές τις εξόδους θα είναι ίσες με το $1/2$ το $1/4$ το $1/8$ και το $1/16$ της συχνότητας του σήματος εισόδου αντίστοιχα. Στο σχήμα 6.20 απεικονίζονται οι συχνότητες στις εξόδους των τεσσάρων f - f .

Συνδυάζοντας κατάλληλα τους απαριθμητές των σχημάτων 6.19α και 6.19β μέσω πυλών AND και OR, μπορούμε να δημιουργήσουμε τον ασύγχρονο απαριθμητή αύξουσα/φθίνουσα μέτρησης, αμφίδρομο απαριθμητή, (up-down counter), που απεικονίζεται στο σχήμα 6.21. Η κατασκευή του κυκλώματος αναλύεται στην παράγραφο 6.7.4. (Υποδειγματικά λυμένες ασκήσεις). (Να συγκρίνετε το κύκλωμα του σχήματος 6.21 με το κύκλωμα του σχήματος 6.38)

Η λειτουργία του κυκλώματος είναι απλή. Όταν είναι ενεργοποιημένη η γραμμή Up, οι εισοδοί των f - f (εκτός του T1) δέχονται παλμούς από τις εξόδους Q_i (έξοδοι Q των T1 έως T4), με αποτέλεσμα η μέτρηση να γίνεται κατά αύξουσα ακολουθία. Στην αντίθετη περίπτωση που η γραμμή Dn (Down) είναι ενεργός, οι εισοδοί δέχονται παλμούς από τις συμπληρωματικές εξόδους \overline{Q}_i , με αποτέλεσμα η μέτρηση να γίνεται πλέον κατά φθίνουσα ακολουθία. Οι έξοδοι απαρίθμησης λαμβάνονται και στους δύο τρόπους λειτουργίας από τις εξόδους Q_i των f - f .

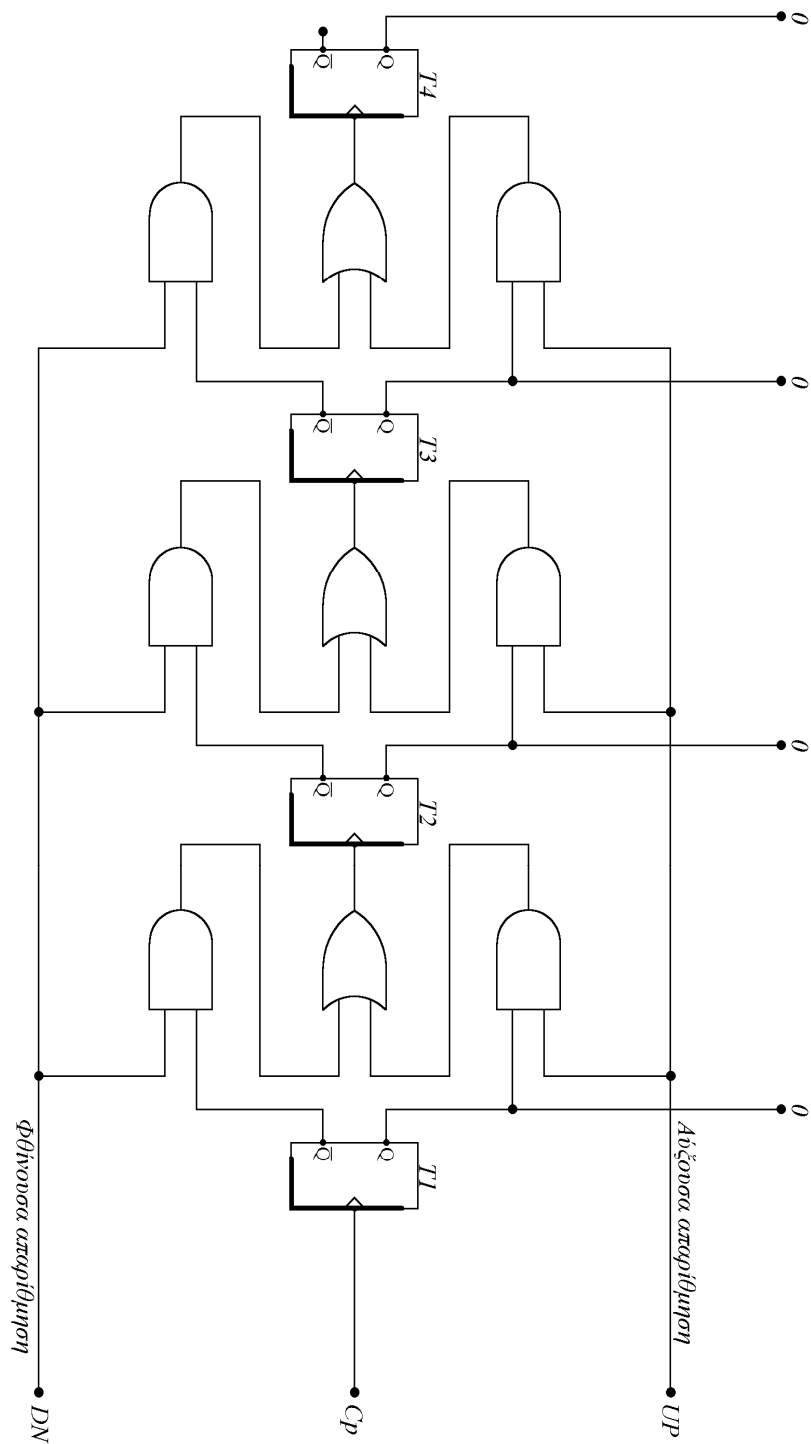
6.4.2. Σύγχρονος δυαδικός απαριθμητής

Στον σύγχρονο δυαδικό απαριθμητή, η εφαρμογή των ωρολογιακών παλμών γίνεται ταυτόχρονα στις εισόδους ρολογιού όλων των f - f . Έτσι ο ρυθμός απαρίθμησης, σε σχέση μ' έναν ασύγχρονο απαριθμητή, είναι πολύ μεγαλύτερος, αφού όλα τα f - f αλλάζουν κατάσταση ταυτόχρονα, και όχι το ένα μετά το άλλο. Ως εκ τούτου, ο χρόνος αλλαγής της κατάστασης ολοκλήρου του απαριθμητή είναι ίσος με το χρόνο λειτουργίας (αλλαγής κατάστασης) ενός μόνο f - f . Φυσικά η βελτίωση του χρόνου απόκρισης δεν πραγματοποιείται χωρίς κόστος. Το κύκλωμα του απαριθμητή γίνεται πολύπλοκότερο, αφού απαιτεί περισσότερα εξαρτήματα, προκειμένου να λειτουργήσει σωστά.

Στο Σχήμα 6.22 απεικονίζεται το κύκλωμα ενός σύγχρονου δυαδικού απαριθμητή με δυνατότητα μέτρησης από το 0 (0000₂) έως και το 15 (1111₂), αφού αποτελείται από 4 f - f . Οι πύλες AND, που το συνοδεύουν, είναι εκείνες που φροντίζουν να εμφανίζονται στις εξόδους του οι επιθυμητοί συνδυασμοί, με την προκαθορισμένη αύξουσα ακολουθία.

Όπως φαίνεται στο σχήμα 6.22, ο κάθε ωρολογιακός παλμός εφαρμόζεται ταυτόχρονα σε όλα τα f - f . Το πρώτο f - f εργάζεται σαν T (J=K=1), αναστρέφοντας κάθε φορά το περιεχόμενο της εξόδου του. Το δεύτερο f - f , ενεργοποιείται, μόνο όταν $Q_1=1$, δηλαδή κάθε δύο παλμούς εισόδου, το τρίτο, όταν $Q_1=1$ και $Q_2=1$, δηλαδή κάθε τέσσερις παλμούς και τέλος το τέταρτο όταν $Q_1=Q_2=Q_3=1$, δηλαδή κάθε οκτώ.

Σχήμα 6.21. Ασύγχρονος δυαδικός απαριθμητής UP-Down



Επιβάλλοντας έναν-έναν τους ωρολογιακούς παλμούς και υπολογίζοντας τις καταστάσεις των εξόδων, διαπιστώνουμε ότι πράγματι ο απαριθμητής παρουσιάζει αύξουσα συμπεριφορά. Επαληθεύει δηλαδή τον πίνακα 6-11 που δίδεται παρακάτω και στον οποίο φαίνονται αναλυτικά οι τρέχουσες και οι επόμενες καταστάσεις των εξόδων Q_4 , Q_3 , Q_2 και Q_1 του απαριθμητή. Παρατηρήστε, επίσης, στον πίνακα αυτό ότι ο $17^{ος}$ παλμός επαναφέρει το κύκλωμα στην αρχική του κατάσταση.

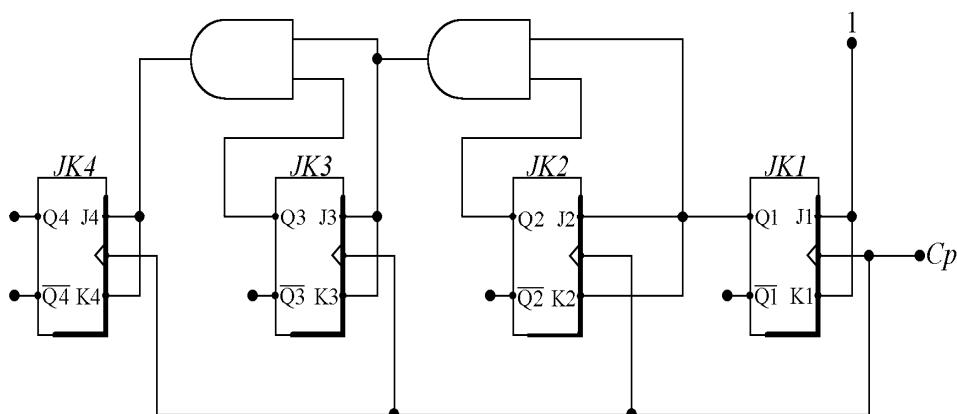
Παλμοί Απαρίθμησης	Δεκαδικός Αριθμός	Τρέχουσα Κατάσταση				Επόμενη Κατάσταση			
		Έξοδοι				Έξοδοι			
		Q4	Q3	Q2	Q1	Q4	Q3	Q2	Q1
1	0	0	0	0	0	0	0	0	1
2	1	0	0	0	1	0	0	1	0
3	2	0	0	1	0	0	0	1	1
4	3	0	0	1	1	0	1	0	0
5	4	0	1	0	0	0	1	0	1
6	5	0	1	0	1	0	1	1	0
7	6	0	1	1	0	0	1	1	1
8	7	0	1	1	1	1	0	0	0
9	8	1	0	0	0	1	0	0	1
10	9	1	0	0	1	1	0	1	0
11	10	1	0	1	0	1	0	1	1
12	11	1	0	1	1	1	1	0	0
13	12	1	1	0	0	1	1	0	1
14	13	1	1	0	1	1	1	1	0
15	14	1	1	1	0	1	1	1	1
16	15	1	1	1	1	0	0	0	0
17	16	0	0	0	0	0	0	0	1

Πίνακας 6.11. Κατάσταση εξόδων σύγχρονου δυαδικού απαριθμητή

Q	Q _{next}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Πίνακας 6.12. Πίνακας αλήθειας του J-K f-f

Για να γίνει δυνατή η κατανόηση του τρόπου σχεδίασης ενός σύγχρονου απαριθμητή, θα προχωρήσουμε στην επαλήθευση του κυκλώματος του σχήματος 6.22, χρησιμοποιώντας χάρτες Karnaugh. Θυμίζουμε ότι ο πίνακας αλήθειας του J-K f-f έχει τη μορφή που φαίνεται στον πίνακα 6.12. Το σύμβολο X υποδηλώνει την αδιάφορη κατάσταση του f-f. Αυτό σημαίνει ότι, είτε επιβάλλουμε στο συγκεκριμένο ακροδέκτη λογικό «0» είτε επιβάλλουμε λογικό «1», η κατάσταση αυτή δε θα έχει καμία απολύτως επίδραση στην κατάσταση της εξόδου (Q_{next}) μετά τον ωρολογιακό παλμό.



Σχήμα 6.22. Σύγχρονος δυαδικός απαριθμητής

Για τη σχεδίαση θα πρέπει να καταστρώσουμε οκτώ χάρτες Karnaugh, έναν για κάθε ακροδέκτη εισόδου των τεσσάρων f - f . Το περιεχόμενο των χαρτών Karnaugh θα το αντλήσουμε από τον πίνακα 6-13. Στον οποίο φαίνονται: στην πρώτη στήλη ο αύξοντας αριθμός των παλμών που επιβάλλουμε, στη δεύτερη και στην τρίτη οι τρέχουσες και οι επόμενες καταστάσεις των εξόδων Q, ενώ στις υπόλοιπες οι καταστάσεις των εισόδων J-K, που, με τη βοήθεια των ωρολογιακών παλμών, προκαλούν τις μεταβολές. Σημειώστε ότι η τελευταία γραμμή του πίνακα έχει τοποθετηθεί με σκοπό να δείξει ότι ο απαριθμητής επαναλαμβάνει τον ίδιο κύκλο μέτρησης, μόλις φθάσει την τιμή $Q_4Q_3Q_2Q_1=1111$. Ως εκ τούτου, δε συμμετέχει στον προσδιορισμό οποιασδήποτε τιμής των χαρτών Karnaugh.

Ο τρόπος με τον οποίο θα κατασκευάσουμε τους χάρτες Karnaugh είναι απλός, αλλά θέλει μεγάλη προσοχή ώστε να αποφευχθούν τα λάθη. Θα αναφέρουμε αναλυτικά το πώς συμπληρώνουμε τους δύο πρώτους χάρτες που αφορούν στο τέταρτο JK f - f , και θα αφήσουμε τον αναγνώστη να κατανοήσει τους άλλους, που αφορούν στα υπόλοιπα f - f .

Παλμός	Τρέχουσα Κατάσταση				Επόμενη Κατάσταση				J ₄	K ₄	J ₃	K ₃	J ₂	K ₂	J ₁	K ₁
	Q ₄	Q ₃	Q ₂	Q ₁	Q ₄	Q ₃	Q ₂	Q ₁								
1 ^{ος}	0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
2 ^{ος}	0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
3 ^{ος}	0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
4 ^{ος}	0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
5 ^{ος}	0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
6 ^{ος}	0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
7 ^{ος}	0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
8 ^{ος}	0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
9 ^{ος}	1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
10 ^{ος}	1	0	0	1	1	0	1	0	X	0	0	X	1	X	X	1
11 ^{ος}	1	0	1	0	1	0	1	1	X	0	0	X	X	0	1	X
12 ^{ος}	1	0	1	1	1	1	0	0	X	0	1	X	X	1	X	1
13 ^{ος}	1	1	0	0	1	1	0	1	X	0	X	0	0	X	1	X
14 ^{ος}	1	1	0	1	1	1	1	0	X	0	X	0	1	X	X	1
15 ^{ος}	1	1	1	0	1	1	1	1	X	0	X	0	X	0	1	X
16 ^{ος}	1	1	1	1	0	0	0	0	X	1	X	1	X	1	X	1
17 ^{ος}	0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X

Πίνακας 6.13. Σχέση της τρέχουσας και της επόμενης κατάστασης σύγχρονου δυαδικού απαριθμητή

Στον πίνακα 6.13 ξεχωρίζουμε μόνο τις στήλες J₄ και K₄, και εντοπίζουμε εκείνους τους συνδυασμούς εξόδων Q που δίνουν λογικό «1». Σημειώνουμε στις αντίστοιχες θέσεις των δύο χαρτών Karnaugh την τρέχουσα κατάσταση. Στο συγκεκριμένο παράδειγμα σημειώνουμε «1» στο πίνακα 6.14 στη θέση $Q_4Q_3Q_2Q_1=0111$ (πίνακας J₄), και στη θέση $Q_4Q_3Q_2Q_1=1111$ (πίνακας K₄). Κατόπιν σημειώνουμε τα σύμβολα των αδιάφορων όρων (X) και ελέγχουμε κατά πόσο αυτά μπορούν να θεωρηθούν σαν «1», ώστε να βοηθήσουν στην απλοποίηση. Στο παράδειγμά μας από όλα τα σύμβολα X είναι σκόπιμο να θεωρηθούν σαν «1» εκείνα που βρίσκονται στη θέση $Q_4Q_3Q_2Q_1=1111$ (πίνακας J₄) που μας δίνει τη συνάρτηση για την είσοδο J, και στη θέση $Q_4Q_3Q_2Q_1=0111$ (πίνακας K₄) που μας δίνει τη συνάρτηση για την είσοδο K. Το αποτέλεσμα μετά από αυτές τις απλοποιήσεις θα είναι η εξαγωγή της συνάρτησης, η οποία θα μας οδηγήσει στο ζητούμενο κύκλωμα.

Με τον ίδιο τρόπο μπορούν να εξαχθούν και οι συναρτήσεις των υπολοίπων f - f (βλ. πίνακες 6.15, 6.16, 6.17), οι οποίες, όπως θα διαπιστώσετε παρακάτω, είναι πολύ πιο απλές από αυτές που ήδη εξηγήσαμε.

		J4			
		Q_2Q_1			
Q_4Q_3		00	01	11	10
00					
01			1		
11		x	X	x	
10		x	x	x	

		K4			
		Q_2Q_1			
Q_4Q_3		00	01	11	10
00		x	x	x	x
01		x	x	X	x
11				1	
10					

Πίνακας 6.14. Προσδιορισμός των συναρτήσεων εξόδου του τέταρτου $f-f$

Οι συναρτήσεις οι οποίες προκύπτουν είναι οι εξής:

$$J4 = Q1 \cdot Q2 \cdot Q3$$

$$K4 = Q1 \cdot Q2 \cdot Q3$$

		J3			
		Q_2Q_1			
Q_4Q_3		00	01	11	10
00				1	
01		x	x	X	x
11		x	x	X	x
10				1	

		K3			
		Q_2Q_1			
Q_4Q_3		00	01	11	10
00		x	x	X	x
01				1	
11				1	
10		x	x	X	x

Πίνακας 6.15. Προσδιορισμός των συναρτήσεων εξόδου του τρίτου $f-f$

Οι συναρτήσεις οι οποίες προκύπτουν είναι οι εξής:

$$J3 = Q1 \cdot Q2$$

$$K3 = Q1 \cdot Q2$$

		J2			
		Q_2Q_1			
Q_4Q_3		00	01	11	10
00			1	X	x
01		x	1	X	x
11		x	1	X	x
10			1	X	x

		K2			
		Q_2Q_1			
Q_4Q_3		00	01	11	10
00		x	X	1	x
01		x	X	1	x
11		x	X	1	x
10		x	X	1	x

Πίνακας 6.16. Προσδιορισμός της συνάρτησης εξόδου του δεύτερου $f-f$

Οι συναρτήσεις οι οποίες προκύπτουν για το δεύτερο $f-f$ είναι οι εξής:

$J2 = Q1$
$J1$
Q_2Q_1
Q_4Q_3
00
01
11
10
00
01
11
10
1
X
X
1
1
X
X
1
1
X
X
1
1
X
X
1
1
X
X
1

$K2 = Q1$
$K1$
Q_2Q_1
Q_4Q_3
00
01
11
10
00
01
11
10
X
1
1
X
X
1
1
X
X
1
1
X
X
1
1
X
X
1
1
X
X
1

Πίνακας 6.17. Προσδιορισμός της συνάρτησης εξόδου του πρώτου f-f

Οι συναρτήσεις οι οποίες προκύπτουν είναι οι εξής:

$$J1 = 1$$

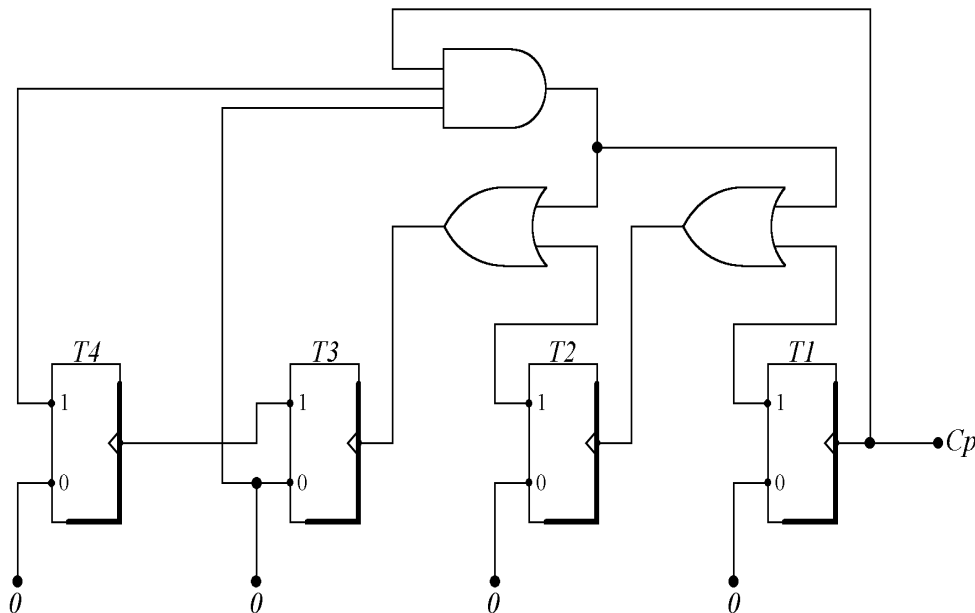
$$K1 = 1$$

Με βάση την παραπάνω ανάλυση καταλήγουμε στο κύκλωμα του σύγχρονου δυαδικού απαριθμητή, που απεικονίζεται στο Σχήμα 6.22.

6.4.3. Απαριθμητές ανάδρασης

Στην κατηγορία αυτών των απαριθμητών, το περιεχόμενο των εξόδων των flip-flop μεταβάλλεται κατά προκαθορισμένα άλματα, ανάλογα με το σύστημα αρίθμησης, το οποίο χρησιμοποιείται για τη μέτρηση.

Για παράδειγμα, για έναν απαριθμητή άλματος που χρησιμοποιεί το σύστημα BCD 2421, πρέπει από την τιμή 5 (1011) να περάσουμε στην τιμή 6 (1100), η οποία διαφέρει κατά ένα από την προηγούμενή της, αλλά απαιτεί την αλλαγή περισσότερων του ενός δυαδικών ψηφίων.



Σχήμα 6.23. Απαριθμητής BCD 2421

Στο Σχήμα 6.23 απεικονίζεται ένας απαριθμητής BCD 2421. Παρατηρώντας το κύκλωμα, φαίνεται πράγματι ότι, αρχίζοντας από την κατάσταση όπου όλες οι εξοδοί είναι 0 και μέχρι το 4, η μέτρηση γίνεται κανονικά. Στη συνέχεια, ο πέμπτος παλμός, με τη βοήθεια της πύλης AND και των πυλών OR, επενεργεί έτσι ώστε οι εξοδοί να αποκτήσουν την τιμή $T_4T_3T_2T_1=1011$. Τα χρησιμοποιούμενα τρανζίστορ είναι PNP, η λο-

γική είναι θετική, όποτε η ενεργοποίηση των εισόδων των $f-f$ γίνεται τη στιγμή στην οποία ο παλμός εισόδου μεταβαίνει από λογικό «0» σε λογικό «1».

Στον πίνακα 6-18 δίνονται όλες οι ενδιάμεσες καταστάσεις του απαριθμητή άλματος, καθώς εφαρμόζονται παλμοί στην είσοδό του.

Παλμοί εισόδου	T ₄	T ₃	T ₂	T ₁
Αρχική κατάσταση	0	0	0	0
1 ^{ος} παλμός	0	0	0	1
2 ^{ος} παλμός	0	0	1	0
3 ^{ος} παλμός	0	0	1	1
4 ^{ος} παλμός	0	1	0	0
5 ^{ος} παλμός	1	0	1	1
6 ^{ος} παλμός	1	1	0	0
7 ^{ος} παλμός	1	1	0	1
8 ^{ος} παλμός	1	1	1	0
9 ^{ος} παλμός	1	1	1	1
10 ^{ος} παλμός	0	0	0	0

Πίνακας 6.18. Κατάσταση εξόδων απαριθμητή άλματος

6.4.4. Απαριθμητές ως προς μέτρο m

Οι απαριθμητές, ως προς μέτρο, (modulo- m) δίνουν τη δυνατότητα μέτρησης, όχι με βάση το 2, αλλά με μία οποιαδήποτε άλλη βάση m , π.χ. το 10, ώστε η μέτρηση να είναι πολλαπλάσιο του δέκα. Κατ' αυτόν τον τρόπο είναι δυνατό να γίνει μέτρηση με βάση το 8, το 16 κ.ο.κ. Οι απαριθμητές ως προς μέτρο m , μετά την απαρίθμηση m παλμών, επανέρχονται στην αρχική τους κατάσταση, που κατά κανόνα είναι ίση με το μηδέν.

Οι απαριθμητές αυτού του τύπου κατασκευάζονται συνήθως από J-K ή T $f-f$. Με τη βοήθεια μιας πύλης AND εφαρμόζονται ταυτόχρονα παλμοί μηδενισμού σ' όλα τα $f-f$ του απαριθμητή (είσοδοι CLR), όταν αυτός φθάσει στην κατάσταση όπου παριστάνει τον αριθμό m (είσοδοι της πύλης AND ταυτόχρονα σε λογικό «1»).

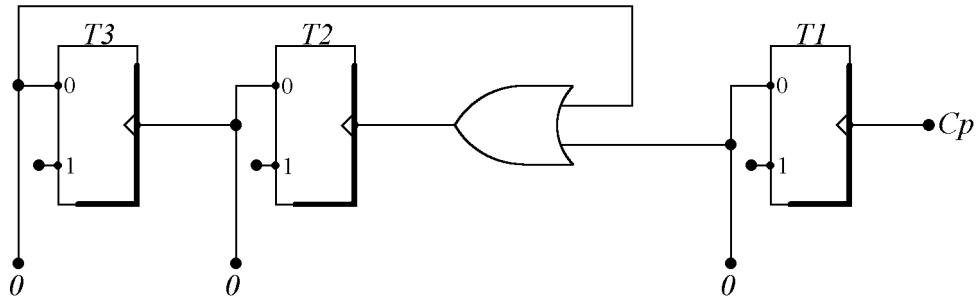
Φυσικά υπάρχουν και άλλοι τρόποι σχεδίασης απαριθμητών ως προς μέτρο m , που εφαρμόζονται κατά περίπτωση έτσι, ώστε να επιτυγχάνεται απλούστευση και οικονομία στο αντίστοιχο λογικό κύκλωμα.

Στο σχήμα 6.24. δίνεται ένας απαριθμητής ως προς μέτρο 6 φθίνουσας τάξης, με T $f-f$ και τρανζίστορ τύπου PNP. Θεωρούμε ότι αρχικά όλες οι εξοδοί βρίσκονται σε λογικό «0». Ο απαριθμητής μπορεί να χρησιμοποιηθεί και σαν διαιρέτης συχνότητας, ο οποίος θα διαιρεί με τον αριθμό έξι.

Στον πίνακα 6-19 σημειώνονται οι καταστάσεις των εξόδων του απαριθμητή αυτού.

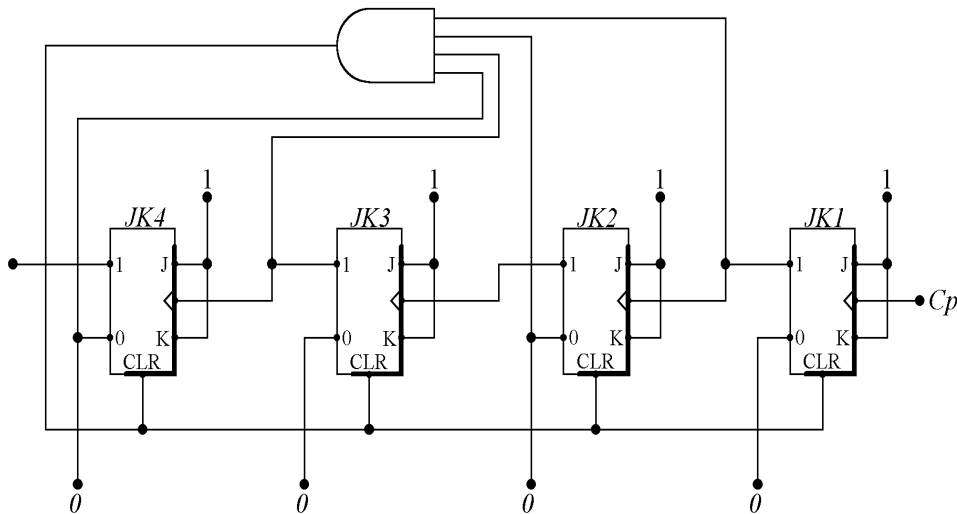
Παλμοί εισόδου	T ₃	T ₂	T ₁
Αρχική κατάσταση	0	0	0
1 ^{ος} Παλμός	0	0	1
2 ^{ος} Παλμός	0	1	0
3 ^{ος} Παλμός	0	1	1
4 ^{ος} Παλμός	1	0	0
5 ^{ος} Παλμός	1	0	1
6 ^{ος} Παλμός	0	0	0

Πίνακας 6.19. Κατάσταση εξόδων απαριθμητή ως προς μέτρο 6



Σχήμα 6.24. Απαριθμητής ως προς μέτρο 6

Στο Σχήμα 6.25 απεικονίζεται απαριθμητής ως προς μέτρο 10.



Σχήμα 6.25. Απαριθμητής ως προς μέτρο 10

Ο απαριθμητής είναι κατασκευασμένος με J-K $f-f$ τα οποία όμως είναι συνδεδεμένα με τέτοιο τρόπο, ώστε να συμπεριφέρονται σαν T $f-f$. Μέσω της πύλης AND, εφαρμόζεται παλμός μηδενισμού σε όλα τα $f-f$ του απαριθμητή, όταν αυτός φτάσει την τιμή 10.

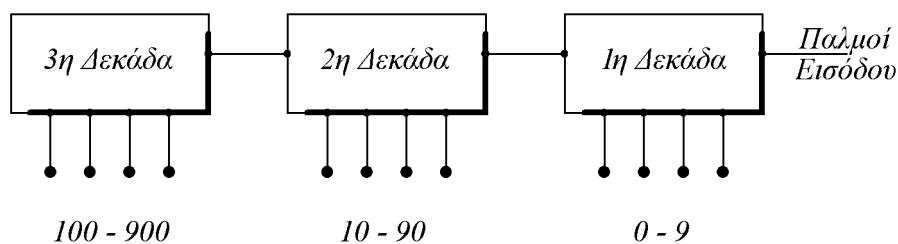
Στον πίνακα 6.20 βλέπουμε τις διαδοχικές καταστάσεις των εξόδων του απαριθμητή, καθώς στην είσοδο του φθάνουν οι παλμοί C_p . Μόλις στις εξόδους του σχηματισθεί ο αριθμός $1010_2 = 10_{10}$, η έξοδος της πύλης AND οδηγείται σε λογικό «1» αναγκάζοντας όλα τα $f-f$ να μηδενισθούν.

Παλμοί εισόδου	JK ₄	JK ₃	JK ₂	JK ₁
αρχική κατάσταση	0	0	0	0
1 ^{ος} παλμός	0	0	0	1
2 ^{ος} παλμός	0	0	1	0
3 ^{ος} παλμός	0	0	1	1
4 ^{ος} παλμός	0	1	0	0
5 ^{ος} παλμός	0	1	0	1
6 ^{ος} παλμός	0	1	1	0
7 ^{ος} παλμός	0	1	1	1
8 ^{ος} παλμός	1	0	0	0
9 ^{ος} παλμός	1	0	0	1
10 ^{ος} παλμός	1	0	1	0
	0	0	0	0

Πίνακας 6.20. Κατάσταση εξόδων απαριθμητή ως προς μέτρο 10

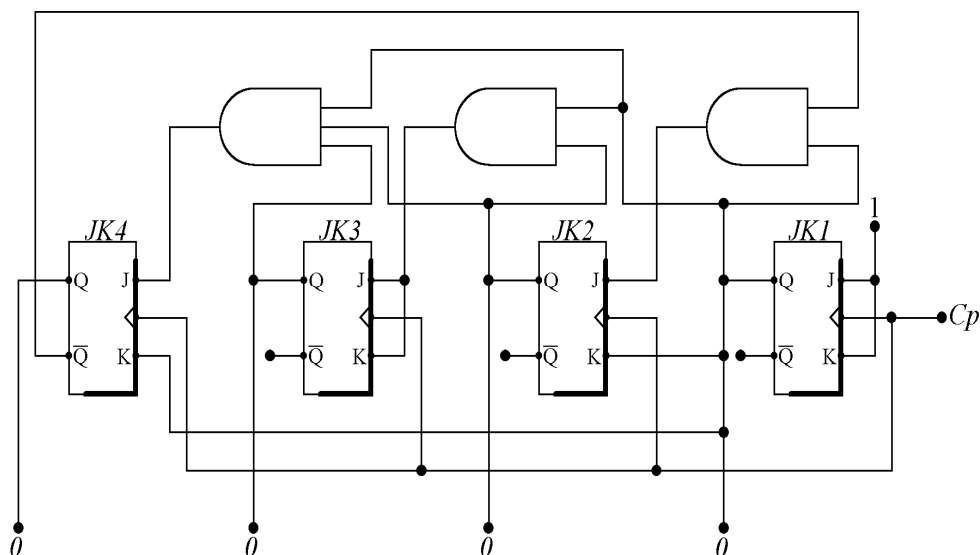
Ο απαριθμητής του σχήματος 6-25 με μία μικρή τροποποίηση θα μπορούσε, αντί να μηδενίζεται μόλις φθάσει μια μέγιστη τιμή (στην περίπτωση μας το 10), να σταματάει τη μέτρηση και να διατηρεί στις εξόδους του τη μέγιστη τιμή του. Αυτό επιτυγχάνεται εύκολα, αποσυνδέοντας την έξοδο της πύλης AND από τις εισόδους CLR των f - f και συνδέοντάς την, μέσω μιας πύλης NOT, στις εισόδους J, K του πρώτου f - f . Επειδή οι δύο αυτές εισόδους, μόλις σχηματιζόταν ο αριθμός 10 στις εξόδους, θα οδηγούνταν σε λογικό «0», θα εμπόδιζαν το f - f JK₁ να αλλάξει κατάσταση. Αν συνέβαινε όμως αυτό και τα υπόλοιπα f - f θα παρέμεναν καθηλωμένα, αφού καμιά μεταβολή δε θα έφθανε στις εισόδους clock.

Στην περίπτωση στην οποία το πλήθος των παλμών, που θέλουμε να μετρήσουμε υπερβαίνει τους 9, τότε θα πρέπει να συνδέσουμε πολλούς δεκαδικούς απαριθμητές σε σειρά έτσι, ώστε το τελικό αποτέλεσμα να ισοδυναμεί μ' έναν απαριθμητή, του οποίου η δυνατότητα απαρίθμησής να είναι ίση με μια δύναμη του 10 (10, 100, 1000, 10000 κ.λπ.). Στο Σχήμα 6.26 που ακολουθεί παριστάνεται ένα διάγραμμα με τη βοήθεια του οποίου μπορούν να μετρηθούν μέχρι 999 παλμοί.



Σχήμα 6.26. Συνδυασμός απαριθμητών modulo 10 για τη μέτρηση τριψήφιου αριθμού

Στο Σχήμα 6.27 απεικονίζεται ένας σύγχρονος BCD απαριθμητής. Ο απαριθμητής αυτός, όπως προδίδει και το όνομά του, μετράει σύμφωνα με τους κανόνες του κώδικα BCD, έχοντας σαν μέγιστη τιμή τον αριθμό 9. Είναι και αυτός ένας δεκαδικός απαριθμητής (μέτρου 10), αλλά διαφέρει από τον απαριθμητή του σχήματος 6.25 στο ότι αυτός είναι σύγχρονος.



Σχήμα 6.27. Σύγχρονος BCD απαριθμητής ως προς μέτρο 10

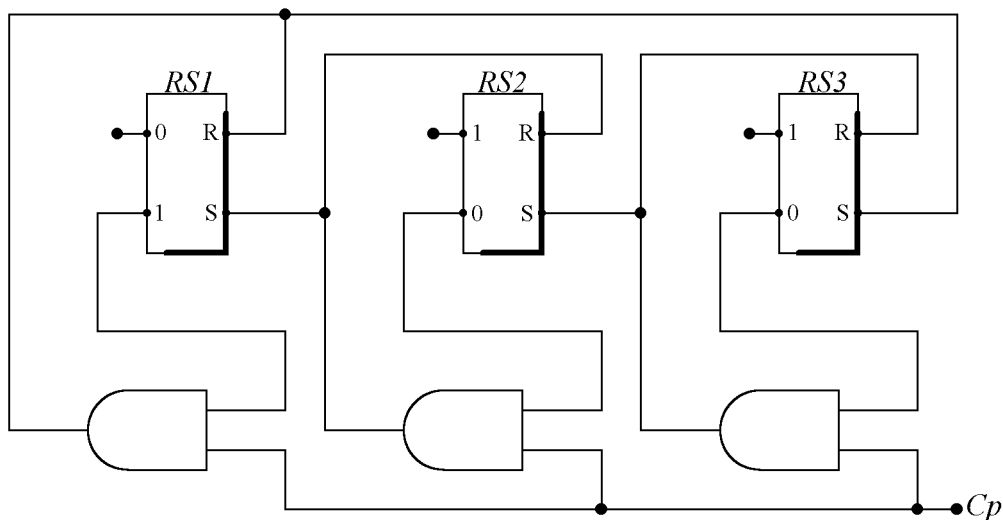
Το πρώτο J-K f - f εργάζεται σαν ένα T f - f , αφού δέχεται και στις δύο εισόδους ελέγχου λογικό «1». Στα υπόλοιπα, οι καταστάσεις που επιβάλλονται στις αντίστοιχες εισόδους τους εξαρτώνται από τις καταστάσεις των εξόδων όλων των f - f . Σαν αρχική συνθήκη του κυκλώματος δεχόμαστε ότι όλες οι εξόδους βρίσκονται σε λογικό «0». Στον πίνακα 6-21 δίνονται οι διάφορες καταστάσεις των f - f με την εφαρμογή των παλμών ρολογιού.

Παλμοί	0 ^{ος}	1 ^{ος}	2 ^{ος}	3 ^{ος}	4 ^{ος}	5 ^{ος}	6 ^{ος}	7 ^{ος}	8 ^{ος}	9 ^{ος}	10 ^{ος}
Q1	0	1	0	1	0	1	0	1	0	1	0
Q2	0	0	1	1	0	0	1	1	0	0	0
Q3	0	0	0	0	1	1	1	1	0	0	0
Q4	0	0	0	0	0	0	0	0	1	1	0

Πίνακας 6.21. Κατάσταση εξόδων σύγχρονου απαριθμητή ως προς μέτρο 10

6.4.5. Απαριθμητής δακτυλίου

Πρόκειται για μια διάταξη απαριθμητή του οποίου οι εξοδοί των $f-f$ βρίσκονται όλες σε λογικό «0» εκτός μίας. Εφαρμόζοντας παλμούς στην είσοδο του κυκλώματος, το λογικό «1» της μιας εξόδου μετακινείται στην έξοδο του επόμενου $f-f$. Η διαδικασία αυτή συνεχίζεται, έως ότου το λογικό «1» εμφανισθεί στην έξοδο του τελευταίου $f-f$, οπότε τελειώνει ο πρώτος κύκλος περιστροφής, και επανεκκινείται ο δεύτερος, καθώς η συγκεκριμένη λογική κατάσταση επανέρχεται πάλι στην έξοδο του πρώτου.



Σχήμα 6.28. Απαριθμητής δακτυλίου

Στο Σχήμα 6.28 απεικονίζεται ένας τυπικός απαριθμητής δακτυλίου, κατασκευασμένος με R-S $f-f$. Θα υποθέσουμε ότι η έξοδος του τρίτου RS βρίσκεται σε λογικό «1», οπότε όλες οι άλλες θα βρίσκονται εξ' ορισμού σε λογικό «0». Εφαρμόζοντας τον πρώτο παλμό στην είσοδο του κυκλώματος, θα γίνει «1» η έξοδος του RS1, ενώ οι άλλες θα παραμείνουν σε λογικό «0». Με το δεύτερο παλμό γίνεται «1» η έξοδος του RS2 κ.ο.κ. Στον πίνακα 6.22 δίνονται οι καταστάσεις των εξόδων μετά την εφαρμογή των παλμών εισόδου.

Παλμοί εισόδου	RS3	RS2	RS1
Αρχική κατάσταση	0	0	0
1 ^{ος} παλμός	0	0	1
2 ^{ος} παλμός	0	1	0
3 ^{ος} παλμός	1	0	0

Πίνακας 6-22 Κατάσταση εξόδων απαριθμητή δακτυλίου

6.4.6. Απαριθμητής Johnson

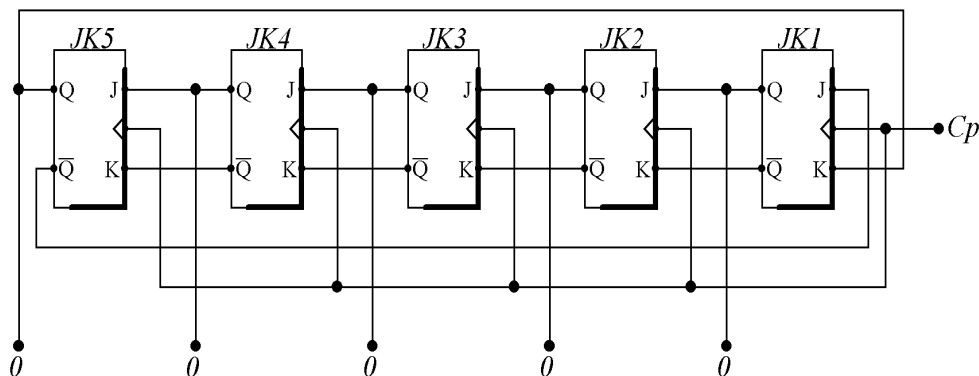
Ο απαριθμητής Johnson (Twisted Ring Counter) είναι ένας σύγχρονος απαριθμητής βαθμίδων, του οποίου τα περιεχόμενα των εξόδων του τελευταίου $f-f$ αναστρέφονται και οδηγούνται στις εισόδους του πρώτου $f-f$. Πιο

συγκεκριμένα, και όπως απεικονίζεται στο Σχήμα 6.29, η έξοδος **Q** του πέμπτου JK οδηγείται στην είσοδο **K** του πρώτου JK, ενώ η έξοδος \overline{Q} του πέμπτου JK στην είσοδο **J** του πρώτου JK.

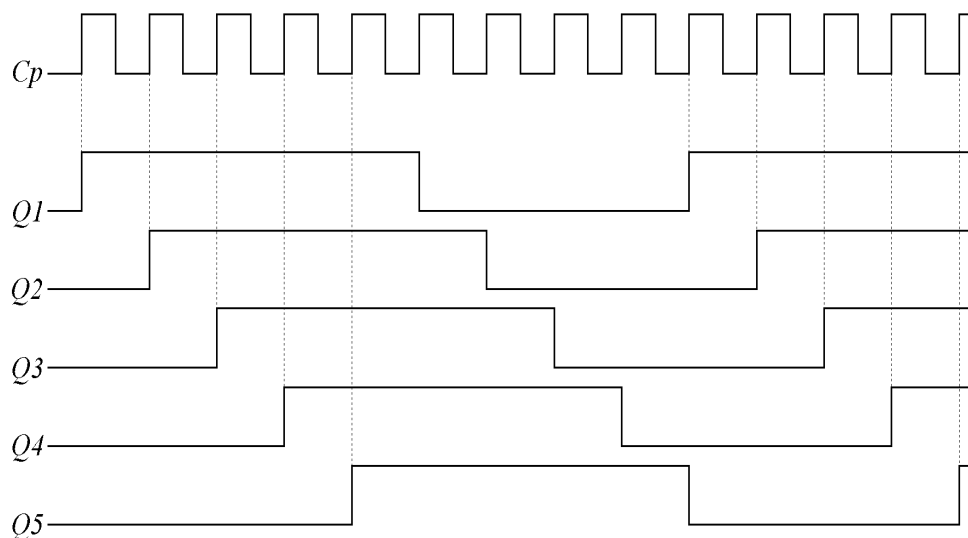
Υποθέτουμε ότι όλες οι έξοδοι **Q** βρίσκονται σε λογικό «0». Με την εφαρμογή του πρώτου ωρολογιακού παλμού, το πρώτο $f\text{-}f$ αλλάζει κατάσταση και οδηγείται σε λογικό «1». Με την εφαρμογή του δεύτερου παλμού, το πρώτο μένει σε «1», ενώ το δεύτερο γίνεται και αυτό «1» κ.ο.κ.

Στον πίνακα 6.23 σημειώνονται όλες οι λογικές καταστάσεις που εμφανίζονται στις εξόδους για κάθε παλμό που επιβάλλεται στην είσοδο του απαριθμητή Johnson. Στο Σχήμα 6.30 δίδονται οι αντίστοιχες κυματομορφές στις εξόδους των $f\text{-}f$.

Ο απαριθμητής Johnson κατά κανόνα χρησιμοποιείται σε κυκλώματα χρονισμού και για διαδοχικές λειτουργίες διαφόρων λογικών κυκλωμάτων.



Σχήμα 6.29. Απαριθμητής Johnson



Σχήμα 6.30. Κυματομορφές στις εξόδους του απαριθμητή Johnson

Παλμοί	0 ^{ος}	1 ^{ος}	2 ^{ος}	3 ^{ος}	4 ^{ος}	5 ^{ος}	6 ^{ος}	7 ^{ος}	8 ^{ος}	9 ^{ος}	10 ^{ος}
Q1	0	1	1	1	1	1	0	0	0	0	0
Q2	0	0	1	1	1	1	1	0	0	0	0
Q3	0	0	0	1	1	1	1	1	0	0	0
Q4	0	0	0	0	1	1	1	1	1	0	0
Q5	0	0	0	0	0	1	1	1	1	1	0

Πίνακας 6.23. Κατάσταση εξόδων απαριθμητή Johnson

Άσκηση με απαριθμητές modulo

Να προσδιορίσετε τη συχνότητα εξόδου ενός κασκωδικού συνδυασμού ο οποίος αποτελείται από ένα κυκλικό απαριθμητή των 10-bit, ένα 10 bit παράλληλο απαριθμητή, έναν MOD 25 απαριθμητή κυμάτωσης (ripple) και έναν 4-bit απαριθμητή Johnson. Η συχνότητα εισόδου του 10-bit κυκλικού απαριθμητή είναι 160 KHz.

Λύση

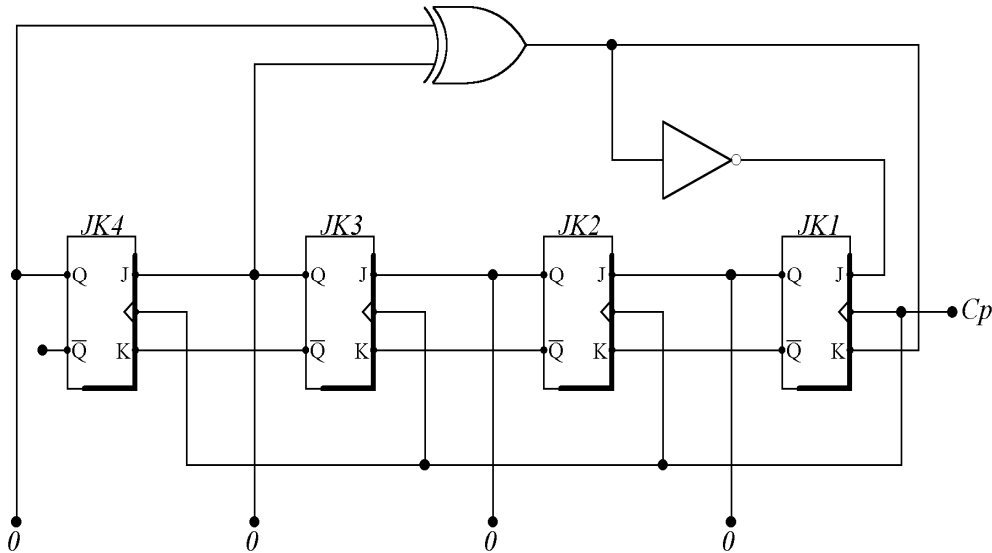
Ο 10-bit κυκλικός απαριθμητής είναι ένας MOD 10 απαριθμητής. Ο 10-bit παράλληλος απαριθμητής είναι ένας MOD 2^{10} απαριθμητής. MOD 25 απαριθμητή κυμάτωσης και τέλος ο 4-bit απαριθμητή Johnson είναι ένας MOD 8 απαριθμητής. Συνεπώς έχουμε:

$$f_{\text{εξόδου}} = \frac{160\text{KHz}}{10 \times 2^{10} \times 25 \times 8} = 0.078125\text{Hz}$$

6.5. Γεννήτρια ψευδοτυχαίων δυαδικών αριθμών

Η γεννήτρια αυτή παράγει τυχαίες δυαδικές ακολουθίες, αρχίζοντας από μία αρχική κατάσταση. Όπως απεικονίζεται στο Σχήμα 6.31, η αρχική κατάσταση είναι η 0000. Μετά από έναν αριθμό παλμών, οι δυαδικές ακολουθίες επαναλαμβάνονται (Φραγκάκης, 1975; Nelson, et al., 1995; Κοσσίδης, & Γιαννακόπουλος, 2006)

Στον πίνακα 6.24 δίνονται όλες οι καταστάσεις στις εξόδους των $f-f$ για κάθε έναν παλμό που επιβάλλεται στην είσοδο του κυκλώματος. Όπως φαίνεται στον πίνακα, μετά από 15 παλμούς το κύκλωμα επανέρχεται στην αρχική του κατάσταση.



Σχήμα 6.31. Γεννήτρια ψευδοτυχαίων δυαδικών αριθμών

Παλμός	0	1 ^{ος}	2 ^{ος}	3 ^{ος}	4 ^{ος}	5 ^{ος}	6 ^{ος}	7 ^{ος}	8 ^{ος}	9 ^{ος}	10 ^{ος}	11 ^{ος}	12 ^{ος}	13 ^{ος}	14 ^{ος}	15 ^{ος}
Q1	0	1	1	1	0	1	1	0	0	1	0	1	0	0	0	0
Q2	0	0	1	1	1	0	1	1	0	0	1	0	1	0	0	0
Q3	0	0	0	1	1	1	0	1	1	0	0	1	0	1	0	0
Q4	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1	0

Πίνακας 6.24. Κατάσταση εξόδων γεννήτριας ψευδοτυχαίων δυαδικών αριθμών

6.6. Διαγράμματα καταστάσεων στα ακολουθιακά κυκλώματα

Θα επικεντρωθούμε στα ακολουθιακά κυκλώματα, στα οποία προσθέτουμε μνήμη στο υλικό που έχουμε ήδη μελετήσει. Σε αντίθεση με τα συνδυαστικά κυκλώματα, όπως αναφέραμε στο κεφάλαιο 3, οι εξοδοί ενός ακολουθιακού κυκλώματος δεν εξαρτώνται μόνο από τις εισόδους, αλλά και από την κατάσταση, ή το τρέχον περιεχόμενο, κάποιου τμήματος της μνήμης. (Σχήμα 6.32). Παράδειγμα ακολουθιακού κυκλώματος είναι η ηλεκτρονική κλειδαριά ενός χρηματοκιβωτίου.

Αυτό κάνει τα πράγματα πιο δυσνόητα, καθώς οι ίδιες είσοδοι μπορούν να οδηγήσουν σε διαφορετικές εξόδους, οι οποίες εξαρτώνται από το τρέχον περιεχόμενο, που είναι αποθηκευμένο στη μνήμη.

Τα περιεχόμενα της μνήμης, μπορούν, επίσης, να αλλάξουν, καθώς το κύκλωμα «τρέχει». Έτσι χρειαζόμαστε καινούργιες τεχνικές ανάλυσης και σχεδίασης των ακολουθιακών κυκλωμάτων.

Πολλές συσκευές της καθημερινής ζωής είναι από τη φύση τους ακολουθιακές: Οι κλειδαριές με συνδυασμό ανοίγουν, αν εισάγουμε τους αριθμούς με τη σωστή σειρά. Οι ανελκυστήρες κινούνται πάνω, ή κάτω, και ανοίγουν, ή κλείνουν, τις πόρτες, ανάλογα με τα κουμπιά που πατάμε στους διαφορετικούς ορόφους, ή μέσα στον ανελκυστήρα. Τα φανάρια μπορούν να αλλάξουν από κόκκινο σε πράσινο ανάλογα με το εάν ένα αυτοκίνητο περιμένει στη διασταύρωση.



Σχήμα 6.32. Ακολουθιακό κύκλωμα

Το πιο σημαντικό για εμάς, που σχεδιάζουμε το υλικό των υπολογιστών, είναι ότι οι υπολογιστές είναι ακολουθιακοί! Για παράδειγμα, τα πατήματα των πλήκτρων και το πάτημα του ποντικιού σημαίνουν διαφορετικές λειτουργίες, ανάλογα με το ποιο πρόγραμμα εκτελείται και με την κατάσταση στην οποία βρίσκεται εκείνη τη στιγμή το πρόγραμμα αυτό. (Κοσσιδάς, & Γιαννακόπουλος, 2006, Mano, & Ciletti, 2014)

6.6.1. Περιγραφή ακολουθιακών κυκλωμάτων με διαγράμματα καταστάσεων

Για να περιγράψουμε συνδυαστικά κυκλώματα, χρησιμοποιήσαμε την άλγεβρα Boole και τους πίνακες αληθείας.

Στα ακολουθιακά κυκλώματα, μπορούμε επίσης να χρησιμοποιήσουμε την άλγεβρα Boole και τους πίνακες αληθείας, αλλά μπορούμε και να χρησιμοποιήσουμε ένα νέο τρόπο αναπαράστασης, που ονομάζεται *διάγραμμα καταστάσεων*.

Σχεδιάζουμε ένα κόμβο για κάθε κατάσταση, στην οποία μπορεί να βρεθεί το κύκλωμα. Τα f -fs έχουν μόνο δύο καταστάσεις: την $Q=0$ και την $Q=1$. Τα βέλη μεταξύ των κόμβων χαρακτηρίζονται με την ένδειξη «είσοδος/έξοδος» και δείχνουν πώς το κύκλωμα αλλάζει καταστάσεις και ποιες είναι οι εξοδοί του. Σε αυτήν την περίπτωση η έννοια «κατάσταση» και η έννοια «έξοδος» είναι ταυτόσημες.

Έχουμε ήδη γνωρίσει ότι τα f -fs είναι μνήμες του 1-bit και αλλάζουν την κατάστασή τους, σε συνάρτηση με το τι διαβάζουν στην είσοδό τους σε «μια συγκεκριμένη στιγμή» π.χ. στο θετικό μέτωπο ρολογιού, και διατηρούν αυτή τη κατάσταση μέχρι το επόμενο θετικό μέτωπο ρολογιού. Μπορούμε επίσης, να κάνουμε σκανδαλισμό των f -fs και στο αρνητικό μέτωπο ρολογιού.

Θα αναφερθούμε στον τρόπο με τον οποίο κατασκευάζουμε κυκλώματα με f -fs (f -f) και πύλες καθώς και στον τρόπο με τον οποίο αναλύουμε τα ακολουθιακά κυκλώματα.

6.6.2. Συνδυαστικά και ακολουθιακά κυκλώματα

Στα συνδυαστικά κυκλώματα η έξοδος εξαρτάται μόνο από την είσοδο, και η έξοδος εμφανίζεται έπειτα από λίγες καθυστερήσεις πυλών (ελέγχει συνεχώς την είσοδο).

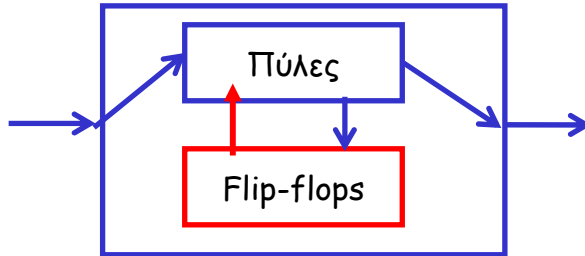


Σε ένα ακολουθιακό κύκλωμα η έξοδος εξαρτάται από την είσοδο όσο και από την τρέχουσα κατάσταση (που είναι αποθηκευμένη στα f -f). Η έξοδος δειγματοληπτεί αντιπροσωπευτικά την είσοδο στα μέτωπα του ρολογιού.

Σε ένα συνδυαστικό κύκλωμα μπορούμε να βρούμε έναν πίνακα αληθείας που θα δείχνει πώς οι εξοδοί συσχετίζονται με τις εισόδους.

Ο πίνακας καταστάσεων είναι το ακολουθιακό ανάλογο του πίνακα αληθείας. Περιλαμβάνει τις εισόδους και τις τρέχουσες καταστάσεις, στα αριστερά, και τις εξόδους και τις επόμενες καταστάσεις, στα δεξιά.

Για ένα ακολουθιακό κύκλωμα, οι εξοδοί δεν εξαρτώνται μόνο από τις εισόδους, αλλά και από την τρέχουσα κατάσταση των $f-f$. Επιπρόσθετα με την εύρεση των τιμών των εξόδων, πρέπει να βρούμε, επίσης, και την κατάσταση των $f-f$ στον επόμενο κύκλο ρολογιού.



Σχήμα 6.33. Συνδυασμός πυλών και $f-f$

Είναι γνωστό ότι σε ένα κύκλωμα υπάρχει μια είσοδος X , μία έξοδος Z , και έστω δύο $f-f$ Q_1Q_0 . Η παρούσα κατάσταση Q_1Q_0 και η είσοδος καθορίζουν την επόμενη κατάσταση και την έξοδο.

Η έξοδος εξαρτάται από την τρέχουσα κατάσταση Q_1Q_0 καθώς, επίσης, και από τις εισόδους. Η εύρεση της επόμενης κατάστασης είναι δυσκολότερη, αφού πρέπει να καταλάβουμε πώς αλλάζουν κατάσταση τα $f-f$. Ακολουθούμε λοιπόν τα παρακάτω βήματα:

Βήμα 1: Βρίσκουμε τις Boolean εκφράσεις για τις εισόδους των $f-f$, δηλαδή πώς οι εισόδοι (έστω J και K) των $f-f$ επηρεάζονται από την τρέχουσα κατάσταση και τις εισόδους.

Βήμα 2: Χρησιμοποιούμε τις εκφράσεις αυτές για να βρούμε τις πραγματικές τιμές εισόδων των $f-f$ για κάθε πιθανό συνδυασμό παρούσας κατάστασης και εισόδων, δηλαδή συμπληρώνουμε τον πίνακα καταστάσεων με νέες ενδιάμεσες στήλες.

Βήμα 3: Κατόπιν χρησιμοποιούμε τους χαρακτηριστικούς πίνακες των $f-f$ ή τις εξισώσεις, για να βρούμε τις επόμενες καταστάσεις, βασισμένοι πάντα στις τιμές των εισόδων και στις παρούσες καταστάσεις των $f-f$.

Υλοποίηση ακολουθιακού κυκλώματος

Βήμα 1: Εξισώσεις εισόδων των $f-f$

Έστω ότι οι εξισώσεις εισόδων για τα $f-f$ είναι: (βλέπε άσκηση 6, Αναγνώριση μορφών)

$$\begin{aligned} J_1 &= \bar{X}Q_0 & J_0 &= X + Q_1 \\ K_1 &= X + Q_0 & K_0 &= \bar{X} \end{aligned}$$

Είναι γνωστό ότι κάθε JK $f-f$ έχει δύο εισόδους, την J και την K . (αντίστοιχα τα D και T $f-f$ s έχουν μία είσοδο το καθένα.)

Βήμα 2: Τιμές των εισόδων των $f-f$

Παρούσα Κατάσταση		Είσοδοι	Είσοδοι των $f-f$				Επόμενη Κατάσταση	
Q_1	Q_0		J_1	K_1	J_0	K_0	Q_1	Q_0
0	0	0	0	0	0	1	0	0
0	0	1	0	1	1	0	0	1
0	1	0	1	1	0	1	1	0
0	1	1	0	1	1	0	0	1
1	0	0	0	0	1	1	1	1
1	0	1	0	1	1	0	0	1
1	1	0	1	1	1	1	0	0
1	1	1	0	1	1	0	0	1

Πίνακας 6.25. Πίνακας επομένων καταστάσεων των εισόδων Q_1 και Q_0

Με δεδομένες τις εξισώσεις του βήματος 1, μπορούμε να φτιάξουμε ένα πίνακα, όπου θα απεικονίζονται οι εισόδοι J_1, K_1, J_0 και K_0 για διαφορετικούς συνδυασμούς παρούσας κατάστασης Q_1Q_0 και της εισόδου X . (πίνακας 6-25).

Βήμα 3: Εύρεση των επομένων καταστάσεων

Θα χρησιμοποιήσουμε τους χαρακτηριστικούς πίνακες, ή τις εξισώσεις των JK $f-f$ για να βρούμε την επόμενη κατάσταση για κάθε $f-f$, βασιζόμενοι πάντα στην παρούσα κατάσταση και στις εισόδους.

Η γενική χαρακτηριστική εξίσωση του JK $f-f$ είναι: $Q(n+1) = \bar{K}Q(n) + J\bar{Q}(n)$

Στο κύκλωμα του παραδείγματός μας, έχουμε δύο JK $f-f$. Έτσι πρέπει να εφαρμόσουμε αυτή την εξίσωση σε κάθε ένα από αυτά. Μπορούμε, επίσης, να προσδιορίσουμε την επόμενη κατάσταση για τον συνδυασμό εισόδου/παρούσας κατάστασης απευθείας από το χαρακτηριστικό πίνακα αλήθειας του $f-f$ (πίνακας 6.12).

Οι επόμενες καταστάσεις για τα Q_1 και Q_0 , υπολογίζονται, χρησιμοποιώντας τις χαρακτηριστικές εξισώσεις: Έχουμε λοιπόν: $Q_1(n+1) = \bar{K}_1Q_1(n) + J_1\bar{Q}_1(n)$ και $Q_0(n+1) = \bar{K}_0Q_0(n) + J_0\bar{Q}_0(n)$ (πίνακας 6.25)

Απευθείας λήψη των στηλών του πίνακα καταστάσεων

Ο πίνακας ξεκινάει με την παρούσα κατάσταση και τις εισόδους. Η παρούσα κατάσταση και οι εισόδοι δίνουν τις εισόδους των $f-f$. Κατόπιν η παρούσα κατάσταση και οι εισόδοι των $f-f$ αποδίδουν την επόμενη κατάσταση, βασιζόμενες στους χαρακτηριστικούς πίνακες των $f-f$.

Η παρούσα κατάσταση και οι εισόδοι παράγουν τις εξόδους. Πραγματικά μας ενδιαφέρουν μόνο οι εισόδοι των $f-f$, για να μπορέσουμε να βρούμε την επόμενη κατάσταση. (πίνακας 6.26 ο οποίος είναι ο πίνακας 6.25 με άλλη μια στήλη, την έξοδο).

Παρούσα Κατάσταση		Είσοδοι X	Είσοδοι των flip-flop				Επόμενη Κατάσταση		Έξοδοι Z
Q_1	Q_0		J_1	K_1	J_0	K_0	Q_1	Q_0	
0	0	0	0	0	0	1	0	0	0
0	0	1	0	1	1	0	0	1	0
0	1	0	1	1	0	1	1	0	0
0	1	1	0	1	1	0	0	1	0
1	0	0	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0	1	0
1	1	0	1	1	1	1	0	0	0
1	1	1	0	1	1	0	0	1	1

Πίνακας 6.26. Πίνακας επομένων καταστάσεων και εξόδου των εισόδων Q_1 και Q_0

Διάγραμμα καταστάσεων

Παρούσα Κατάσταση		Είσοδοι X	Επόμενη Κατάσταση		Έξοδοι Z
Q_1	Q_0		Q_1	Q_0	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	0	1	1

Πίνακας 6.27. Συνοπτικός πίνακας καταστάσεων και εξόδου

Μπορούμε επίσης να παρουσιάσουμε γραφικά τον πίνακα καταστάσεων με ένα διάγραμμα καταστάσεων. Το διάγραμμα, που αντιστοιχεί στον πίνακα καταστάσεων του παραδείγματός μας, απεικονίζεται στο Σχήμα 6.34.

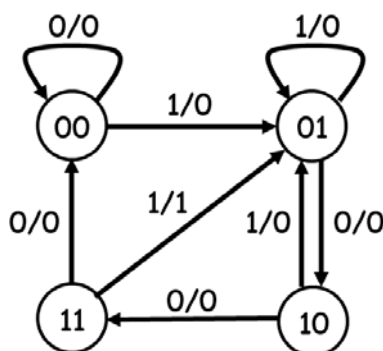
Μέγεθος του διαγράμματος καταστάσεων

Πάντοτε πρέπει να ελέγχουμε το μέγεθος του διαγράμματος καταστάσεων. Αν έχουμε n f -fs, θα πρέπει να υπάρχουν 2^n κόμβοι στο διάγραμμα. Αν υπάρχουν m εισόδοι, τότε κάθε κόμβος θα έχει 2^m εξερχόμενα βέλη, από κάθε κατάσταση!. Στο παράδειγμά μας, έχουμε δύο f -f και, συνεπώς, τέσσερις καταστάσεις ή κόμβους. Υπάρχει μια είσοδος, οπότε κάθε κόμβος έχει δύο εξερχόμενα βέλη. (Σχήμα 6.34). Συνεπώς, για να αναλύσουμε τα ακολουθιακά κυκλώματα, θα πρέπει:

Να βρούμε τις Boolean εκφράσεις για τις εξόδους του κυκλώματος και για τις εισόδους των f -fs.

Να χρησιμοποιήσουμε αυτές τις εκφράσεις για να συμπληρώσουμε τις στήλες του πίνακα καταστάσεων, που αφορούν στις εξόδους του κυκλώματος και τις εισόδους των f -f.

Τέλος να χρησιμοποιήσουμε τις χαρακτηριστικές εξισώσεις ή τους χαρακτηριστικούς πίνακες των f -f, για να συμπληρώσουμε τις στήλες της επόμενης κατάστασης, στον πίνακα καταστάσεων. Το αποτέλεσμα της ανάλυσης του ακολουθιακού κυκλώματος είναι ο πίνακας καταστάσεων ή το διάγραμμα καταστάσεων που περιγράφει το κύκλωμα.



Σχήμα 6.34. Διάγραμμα καταστάσεων όπου απεικονίζονται οι εισόδοι και οι εξόδοι

6.7. Υποδειγματικά λυμένες ασκήσεις

(Φραγκάκης, 1975; Nashelsky, 1994; Givone, 2002; Holdsworth, & Woods, 2002; Katz, 2005; Κοσσίδης, & Γιαννακόπουλος, 2006; Tocci, et al., 2010; Godse, & Godse, 2011; Mano, & Ciletti, 2014; Roth, & Kinney, 2014)

6.7.1. Άσκηση 1^η Απαριθμητής ως προς μέτρο M

Πολλές φορές χρειάζεται να σχεδιασθεί απαριθμητής που να μετρά M παλμούς, και να επαναλαμβάνει την ίδια ακολουθία από τον $M+1$ παλμό εισόδου. Ο απαριθμητής αυτός ονομάζεται *απαριθμητής ως προς μέτρο M (modulo $-M$)*

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	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	x	x	x
1	1	0	x	x	x
1	1	1	x	x	x

Αδιάφοροι όροι του πίνακα

x: αδιάφοροι όροι

Πίνακας 6.28. Πίνακας καταστάσεων απαριθμητή ως προς μέτρο -5

Έστω ότι θέλουμε να υλοποιήσουμε τον απαριθμητή ως προς μέτρο-5 (modulo-5). (Ο απαριθμητής αυτός θα μετρά από το 0 μέχρι το 4 δηλαδή οι καταστάσεις είναι πέντε συνολικά (modulo-5)). Η υλοποίηση θα γίνει με τον κώδικα BCD 8421. Για την υλοποίηση του απαριθμητή ως προς μέτρο-5 απαιτούνται 3 *f-f*. Αρχικά συμπληρώνουμε τον πίνακα μεταφοράς (πίνακα καταστάσεων), (πίνακας 6.28), και θα υλοποιήσουμε τον απαριθμητή με J-K *f-f*.

Ο πίνακας του J-K *f-f* για τις εισόδους J και K αντίστοιχα είναι γνωστός και:

- α) Το J είναι 1 για μετάβαση από το 0 στο 1, και αδιάφορο για μετάβαση από το 1 σε οτιδήποτε (0 ή 1).
 β) Το K είναι 1 για μετάβαση από το 1 στο 0, και αδιάφορο για μετάβαση από το 0 σε οτιδήποτε (0 ή 1).

Συναρτήσεις των μεταβλητών A, B, C

A) Μεταβλητή A – Για το J_A και K_A

Για να εξάγουμε τις συναρτήσεις για τα J_A και K_A με βάση τις μεταβάσεις από το A στο A', ακολουθούμε τα παρακάτω βήματα:

A) Τοποθετούμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 101, 110, 111 που δε χρησιμοποιούνται). Οι όροι αυτοί είναι κοινοί για όλους τους χάρτες που ακολουθούν, για όλες τις εισόδους και για όλες τις μεταβλητές.

B) Κατόπιν τοποθετούμε στο χάρτη Karnaugh τους άσσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_A και για μεταβάσεις από το 1 σε 0 για το K_A). Για το J_A μετάβαση από το 0 σε 1 έχουμε μόνο για τον όρο 011, και για το K_A από 1 σε 0 για τον όρο 100.

Γ) Τέλος θέτουμε στους χάρτες τους αδιάφορους όρους (AO),-οι οποίοι ισχύουν μόνο για το συγκεκριμένο *f-f*, για το J_A και το K_A αντίστοιχα. Η χρήση των AO στην απλοποίηση είναι η ίδια με τους αδιάφορους όρους του χάρτη. (Προκύπτουν από τον πίνακα 6.12 του J-K *f-f*).

Ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα (με τους λιγότερους δυνατών όρους και τις λιγότερες κατά το δυνατόν μεταβλητές ανά όρο). Οι αντίστοιχοι χάρτες Karnaugh και οι εξισώσεις που προκύπτουν είναι:

		AB			
		00	01	11	10
C	0			x	AO
	1		1	x	x

		AB			
		00	01	11	10
C	0	AO	AO	x	1
	1	AO	AO	x	x

Εύκολα προκύπτει ότι $J_A = BC$ και $K_A = 1$

B) Μεταβλητή B – Για το J_B και K_B

Θέτουμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 101, 110, 111).

Κατόπιν θέτουμε στο χάρτη Karnaugh τους άσσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_B και για μεταβάσεις από το 1 σε 0 για το K_B). Για το J_B μετάβαση από το 0 σε 1 έχουμε μόνο για τον όρο 001, και για το K_B από 1 σε 0 για τον όρο 011. Τέλος θέτουμε στους χάρτες τους αδιάφορους όρους του *f-f* για το J_B και το K_B αντίστοιχα.

Ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα. Οι αντίστοιχοι χάρτες Karnaugh και οι εξισώσεις, που προκύπτουν, είναι:

		AB			
		00	01	11	10
C	0		AO	x	
	1	1	AO	x	x

		AB			
		00	01	11	10
C	0	AO		x	AO
	1	AO	1	x	x

Έχουμε τελικά $J_B = C$ και $K_B = C$

C) Μεταβλητή C – Για το J_C και K_C

Θέτουμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 101, 110, 111).

Κατόπιν θέτουμε στο χάρτη Karnaugh τους άσους. Για το J_C μετάβαση από το 0 σε 1 έχουμε για τους όρους 000 και 010, και για το K_C από 1 σε 0 για τους όρους 001 και 011.

Τέλος θέτουμε στους χάρτες τους αδιάφορους όρους του $f-f$ για το J_C και το K_C , αντίστοιχα.

Ομαδοποιούμε τους άσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα. Οι αντίστοιχοι χάρτες Karnaugh και οι εξισώσεις, που προκύπτουν είναι:

	AB				
C		00	01	11	10
0		1	1	x	
1		AO	AO	x	x

	AB				
C		00	01	11	10
0		AO	AO	x	AO
1		1	1	x	x

Από όπου έχουμε

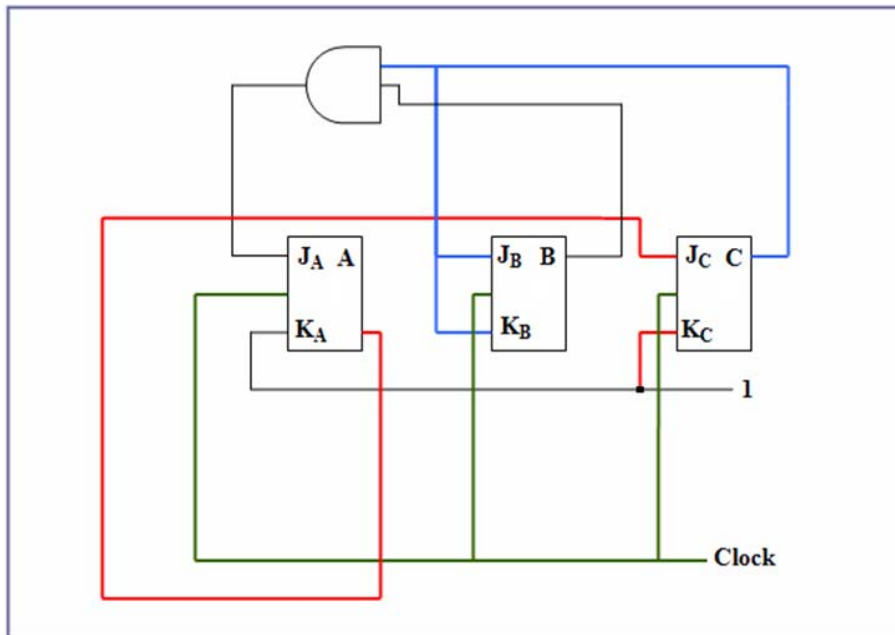
$$J_C = \bar{A} \quad \text{και} \quad K_C = 1$$

Υλοποίηση του απαριθμητή με J-K f-f και πύλες

Οι προς υλοποίηση συναρτήσεις είναι:

$$J_A = BC \quad \text{και} \quad K_A = 1, \quad J_B = C \quad \text{και} \quad K_B = C, \quad J_C = \bar{A} \quad \text{και} \quad K_C = 1$$

Το διάγραμμα, που προκύπτει, είναι το ακόλουθο (Σχήμα 6.35):



Σχήμα 6.35 Απαριθμητής μέτρου-5 με J-K f-f και πύλες

(Να συγκρίνετε το Σχήμα 6.35 με τα αντίστοιχα σχήματα των ασύγχρονων απαριθμητών μέτρου (Σχήμα 6.24 και Σχήμα 6.25)).

6.7.2. Άσκηση 2^η Δεκαδικός απαριθμητής με χρήση του κώδικα XS3

Ο δεκαδικός απαριθμητής μετρά από το 0 έως και το 9 (τα ψηφία του δεκαδικού συστήματος). Έστω ότι θέλουμε να τον υλοποιήσουμε με τον κώδικα XS3 (Excess 3). Τα βήματα που ακολουθούμε είναι:

α) Αρχικά συμπληρώνουμε τον πίνακα καταστάσεων

β) Κατόπιν για καθεμία μεταβλητή (ανάλογα με το $f-f$ που χρησιμοποιώ βρίσκουμε τους αδιάφορους όρους του πίνακα καταστάσεων (αν υπάρχουν), κατόπιν προσδιορίζουμε τους άσους που προκύπτουν για τις

εισόδους των $f-f$ και, τέλος, καταγράφουμε τους αδιάφορους όρους της κάθε εισόδου, χρησιμοποιώντας τον πίνακα των εισόδων του $f-f$ που θα χρησιμοποιήσουμε.

γ) Τέλος, αφού βρούμε τις συναρτήσεις, ακολουθεί η υλοποίηση με τις λογικές πύλες και τα $f-f$.

Για την πραγματοποίηση του δεκαδικού απαριθμητή απαιτούνται 4 $f-f$. Αρχικά συμπληρώνουμε τον πίνακα μεταφοράς (καταστάσεων), και με γνωστό τον πίνακα του J-K $f-f$ έχουμε:

Συναρτήσεις των μεταβλητών A, B, C, D

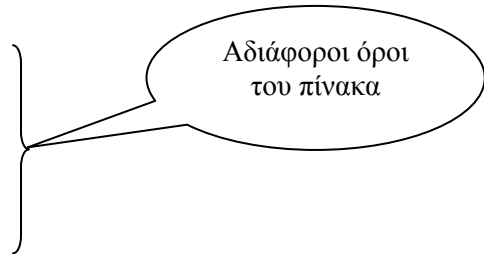
A) Μεταβλητή A – Για το J_A και K_A

Για να εξάγουμε τις συναρτήσεις για τα J_A και K_A με βάση τις μεταβάσεις από το A στο A' , ακολουθούμε τα παρακάτω βήματα:

Αρχικά θέτουμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 0000, 0001, 0010, 1101, 1110, 1111). Οι όροι αυτοί είναι κοινοί για όλους τους χάρτες που ακολουθούν, για όλες τις εισόδους και για όλες τις μεταβλητές.

Κατόπιν θέτουμε στο χάρτη Karnaugh τους άσσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_A και για μεταβάσεις από το 1 σε 0 για το K_A). Για το J_A μετάβαση από το 0 σε 1 έχουμε μόνο για τον όρο 0111, και για το K_A από 1 σε 0 για τον όρο 1100.

Παρούσα Κατάσταση				Επομένη Κατάσταση			
A	B	C	D	A'	B'	C'	D'
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	0	0	1	1
0	0	0	0	x	x	x	x
0	0	0	1	x	x	x	x
0	0	1	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x
x: αδιάφοροι όροι							



Πίνακας 6.29. Πίνακας καταστάσεων του δεκαδικού απαριθμητή με χρήση του κώδικα XS3

Θέτουμε στους χάρτες τους αδιάφορους όρους (AO) του $f-f$ για το J_A και το K_A , αντίστοιχα.

Παρατηρούμε ότι έχουμε πλήρη χάρτη Karnaugh και στις δύο περιπτώσεις που μας δίνει σαν αποτέλεσμα τη μονάδα.

		AB			
		00	01	11	10
CD	00	x		AO	AO
	01	x		x	AO
	11		1	x	AO
	10	x		x	AO

		AB			
		00	01	11	10
CD	00	x	AO	1	
	01	x	AO	x	
	11	AO	AO	x	
	10	x	AO	x	

Τέλος ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα (με τους λιγότερους δυνατόν όρους και τις λιγότερες κατά το δυνατόν μεταβλητές ανά όρο). Οι αντίστοιχοι χάρτες Karnaugh και οι εξισώσεις, που προκύπτουν, είναι: $J_A = BCD$ και $K_A = B$

B) Μεταβλητή B – Για το J_B και K_B

Για να εξάγουμε τις συναρτήσεις για τα J_B και K_B , με βάση τις μεταβάσεις από το B στο B': Αρχικά θέτουμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 0000, 0001, 0010, 1101, 1110, 1111).

Κατόπιν θέτουμε στο χάρτη Karnaugh τους άσσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_B και για μεταβάσεις από το 1 σε 0 για το K_B). Για το J_B μετάβαση από το 0 σε 1 έχουμε τους όρους 0011 και 1011, και για το K_B τους όρους 0111 και 1101.

Τέλος θέτουμε στους χάρτες τους αδιάφορους όρους (AO) του f-f για το J_B και το K_B αντίστοιχα.

Τέλος ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα.

		AB			
		00	01	11	10
CD	00	x	AO	AO	
	01	x	AO	x	
	11	1	AO	x	1
	10	x	AO	x	

		AB			
		00	01	11	10
CD	00	x		1	AO
	01	x		x	AO
	11	AO	1	x	AO
	10	x		x	AO

Εύκολα προκύπτει η $J_B = CD$ και $K_B = A + CD$

C) Μεταβλητή C – Για το J_C και K_C

Για να εξάγουμε τις συναρτήσεις για τα J_C και K_C με βάση τις μεταβάσεις από το C στο C': Θέτουμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 0000, 0001, 0010, 1101, 1110, 1111).

Κατόπιν θέτουμε στο χάρτη Karnaugh τους άσσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_C και για μεταβάσεις από το 1 σε 0 για το K_C). Για το J_C μετάβαση από το 0 σε 1 έχουμε τους όρους 1100, 0101 και 1001, και για το K_C τους όρους 0011, 0111 και 1011.

Θέτουμε στους χάρτες τους αδιάφορους όρους (AO) του f-f για το J_C και το K_C , αντίστοιχα.

Τέλος ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα. Έτσι προκύπτουν οι σχέσεις $J_C = AB + D$ και $K_C = D$

		AB			
		00	01	11	10
CD	00	x		1	
	01	x	1	x	1
	11	AO	AO	x	AO
	10	x	AO	x	AO

		AB			
		00	01	11	10
CD	00	x	AO	AO	AO
	01	x	AO	x	AO
	11	1	1	x	1
	10	x		x	

D) Μεταβλητή D – Για το J_D και K_D

Για να εξάγουμε τις συναρτήσεις για τα J_D και K_D με βάση τις μεταβάσεις από το D στο D':

Θέτουμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 0000, 0001, 0010, 1101, 1110, 1111). Κατόπιν θέτουμε στο χάρτη Karnaugh τους άσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_D και για μεταβάσεις από το 1 σε 0 για το K_D). Για το J_D μετάβαση από το 0 σε 1 έχουμε τους όρους 1100, 0101 και 1001, και για το K_D τους όρους 0011, 0111 και 1011. Τέλος θέτουμε στους χάρτες τους αδιάφορους όρους (AO) του $f-f$ για το J_D και το K_D , αντίστοιχα.

Παρατηρούμε ότι έχουμε πλήρη χάρτη Karnaugh και στις δύο περιπτώσεις, που μας δίνει σαν αποτέλεσμα τη μονάδα.

		AB			
		00	01	11	10
CD	00	x	1	1	1
	01	x	AO	x	AO
	11	AO	AO	x	AO
	10	x	1	x	1

		AB			
		00	01	11	10
CD	00	x	AO	AO	AO
	01	x	1	x	1
	11	1	1	x	1
	10	x	AO	x	AO

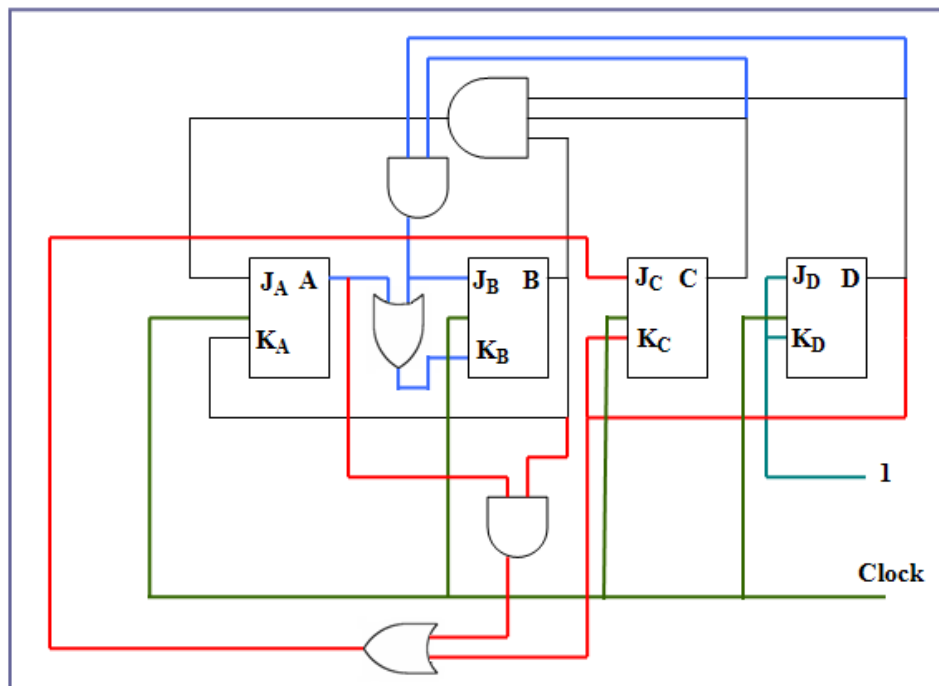
Άρα έχουμε: $J_D = 1$ και τη $K_D = 1$

Υλοποίηση με JK f-f και πύλες

Οι προς υλοποίηση συναρτήσεις είναι λοιπόν:

$$J_A = BCD \text{ και } K_A = B, J_B = CD \text{ και } K_B = A + CD, J_C = AB + D \text{ και } K_C = D \text{ } J_D = 1 \text{ και } K_D = 1$$

Το διάγραμμα, που προκύπτει, είναι το ακόλουθο (Σχήμα 6.36):



Σχήμα 6.36. Δεκαδικός απαριθμητής κώδικα XS3 με J-K f-f και πύλες

6.7.3. Άσκηση 3^η Μη σειριακός απαριθμητής

Έστω ότι ο απαριθμητής θέλουμε να μετρά περνώντας από τις καταστάσεις π.χ. 2,7,5,6,2,..., και έστω ότι θέλουμε να τον υλοποιήσουμε με τον κώδικα BCD 8421. Τα βήματα που θα ακολουθήσουμε είναι:

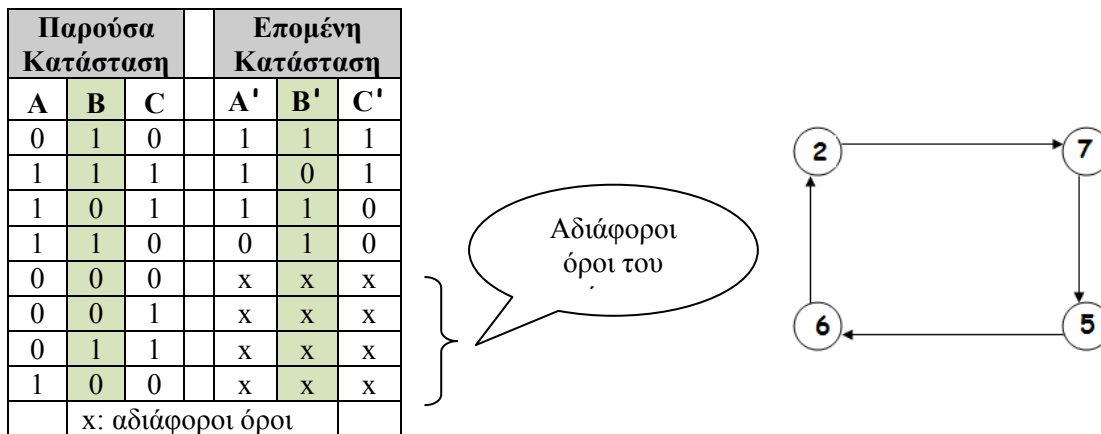
Αρχικά σχεδιάζουμε το διάγραμμα καταστάσεων και, κατόπιν, γράφουμε τον πίνακα καταστάσεων.

Στη συνέχεια, για καθεμία μεταβλητή, βρίσκω τους αδιάφορους όρους του πίνακα καταστάσεων (ανάλογα με το flip- flop που χρησιμοποιώ) κατόπιν τους άσσους και τέλος τους αδιάφορους όρους της εισόδου (αν υπάρχουν).

Αφού βρούμε τις συναρτήσεις ακολουθεί η υλοποίηση με λογικές πύλες και *f-f*.

Για την πραγματοποίηση του απαριθμητή απαιτούνται 3 *f-f*. Αρχικά συμπληρώνουμε τον πίνακα μεταφοράς (καταστάσεων) με γνωστό τον πίνακα του J-K *f-f*.

Διάγραμμα Καταστάσεων



Πίνακας 6.30. Πίνακας καταστάσεων του μη σειριακού απαριθμητή με κώδικα BCD 8421

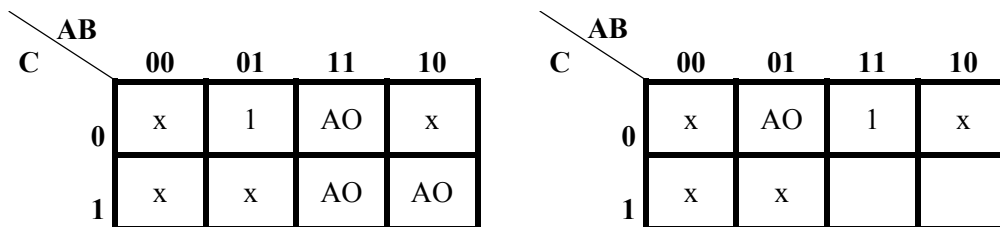
Συναρτήσεις των μεταβλητών A, B, C

A) Μεταβλητή A – Για το J_A και K_A

Για να εξάγουμε τις συναρτήσεις για τα J_A και K_A με βάση τις μεταβάσεις από το A στο A', ακολουθούμε τα παρακάτω βήματα:

Αρχικά θέτουμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 000, 001, 011, 100). Οι όροι αυτοί είναι κοινοί για όλους τους χάρτες που ακολουθούν, για όλες τις εισόδους και για όλες τις μεταβλητές. Κατόπιν θέτουμε στο χάρτη Karnaugh τους άσσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_A και για μεταβάσεις από το 1 σε 0 για το K_A). Για το J_A μετάβαση από το 0 σε 1 έχουμε τον όρο 010, και για το K_A από 1 σε 0 τον όρο 110. Τέλος θέτουμε στους χάρτες τους αδιάφορους όρους (AO) του *f-f* για το J_A και το K_A , αντίστοιχα.

Ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα (με τους λιγότερους δυνατών όρους και τις λιγότερες κατά τα δυνατόν μεταβλητές ανά όρο). Οι αντίστοιχοι χάρτες Karnaugh και οι εξισώσεις, που προκύπτουν, είναι: $J_A = 1$ και $K_A = \bar{C}$



B) Μεταβλητή B – Για το J_B και K_B

Για να εξάγουμε τις συναρτήσεις για τα J_B και K_B με βάση τις μεταβάσεις από το B στο B', ακολουθούμε τα παρακάτω βήματα:

Αρχικά θέτουμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 000, 001, 011, 100). Κατόπιν θέτουμε στο χάρτη Karnaugh τους άσσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_B και για μεταβάσεις από το 1 σε 0 για το K_B). Για το J_B μετάβαση από το 0 σε 1 έχουμε για τον όρο

111, και για το K_B , από 1 σε 0, για τον όρο 011. Τέλος θέτουμε στους χάρτες τους αδιάφορους όρους του f για το J_B και το K_B , αντίστοιχα.

Ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα. Οι αντίστοιχοι χάρτες Karnaugh και οι εξισώσεις, που προκύπτουν, είναι: $J_B = 1$ και $K_B = C$

		AB			
		00	01	11	10

		AB			
		00	01	11	10

C) Μεταβλητή C – Για το J_C και K_C

Θέτουμε στο χάρτη τους αδιάφορους όρους του πίνακα αλήθειας (x) (καταστάσεις 101, 110, 111).

Κατόπιν θέτουμε στο χάρτη Karnaugh τους άσσους. Για το J_C μετάβαση από το 0 σε 1 έχουμε για τους όρους 000 και 010, και για το K_C από 1 σε 0 για τους όρους 001 και 011. Τέλος θέτουμε στους χάρτες τους αδιάφορους όρους του f για το J_C και το K_C , αντίστοιχα.

Ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα. Οι αντίστοιχοι χάρτες Karnaugh και οι εξισώσεις, που προκύπτουν, είναι:

		AB			
		00	01	11	10

		AB			
		00	01	11	10

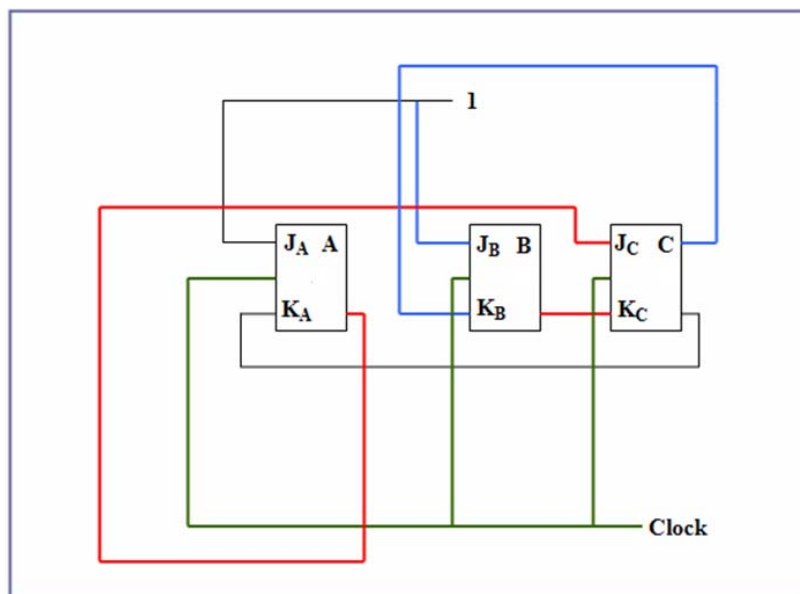
Για την $J_C = \bar{A}$ και $K_C = \bar{B}$

Υλοποίηση του μη σειριακού απαριθμητή με J-K f-f και πύλες

Οι προς υλοποίηση συναρτήσεις είναι:

$J_A = 1$ και $K_A = \bar{C}$ $J_B = 1$ και $K_B = C$, $J_C = \bar{A}$ και $K_C = \bar{B}$

Το διάγραμμα, που προκύπτει, είναι το ακόλουθο: (Σχήμα 6.37)



Σχήμα 6.37. Μη σειριακός απαριθμητής με J-K f-f και πύλες

6.7.4. Άσκηση 4^η Αμφίδρομος απαριθμητής οκτώ καταστάσεων

Ο αμφίδρομος απαριθμητής μπορεί να αυξάνει, ή να ελαττώνει, το περιεχόμενό του κατά ένα σε κάθε παλμό εισόδου, και γι' αυτό ονομάζεται αμφίδρομος απαριθμητής (up-down counter). Έστω ότι θέλουμε να τον υλοποιήσουμε με τον κώδικα BCD 8421. (Πίνακας 6.31)

Παρούσα Κατάσταση					Επομένη Κατάσταση		
A	B	C	K		A'	B'	C'
0	0	0	0		0	0	1
0	0	1	0		0	1	0
0	1	0	0		0	1	1
0	1	1	0		1	0	0
1	0	0	0		1	0	1
1	0	1	0		1	1	0
1	1	0	0		1	1	1
1	1	1	0		0	0	0
0	0	0	1		1	1	1
1	1	1	1		1	1	0
1	1	0	1		1	0	1
1	0	1	1		1	0	0
1	0	0	1		0	1	1
0	1	1	1		0	1	0
0	1	0	1		0	0	1
0	0	1	1		0	0	0

Πίνακας 6.31. Πίνακας καταστάσεων του αμφίδρομου απαριθμητή υλοποιημένου με κώδικα BCD 8421

Θα χρειαστούμε μία επιπλέον είσοδο ελέγχου, η οποία θα προσδιορίζει τη λειτουργία του απαριθμητή. Έτσι, όταν, για παράδειγμα η είσοδος είναι μηδέν, θα λειτουργεί ως αυξητικός απαριθμητής, ενώ όταν είναι ένα, σαν μειούμενος απαριθμητής. Έστω λοιπόν K αυτή η είσοδος ελέγχου της λειτουργίας του. Για την πραγματοποίηση του απαριθμητή απαιτούνται 3 *f-f* και η είσοδος K. Σχεδιάζουμε τον πίνακα καταστάσεων και με γνωστό τον πίνακα του J-K *f-f* έχουμε:

Συναρτήσεις των μεταβλητών A, B, C

A) Μεταβλητή A – Για το J_A και K_A

		AB						AB			
		00	01	11	10			00	01	11	10
CK	00			AO	AO	AO	AO				
	01	1		AO	AO	AO	AO			1	
	11			AO	AO	AO	AO				
	10		1	AO	AO	AO	AO	1			

Στην περίπτωση αυτή δεν υπάρχουν αδιάφοροι όροι του πίνακα καταστάσεων. Έτσι θέτουμε στο χάρτη Karnaugh τους άσους. Για το J_A μετάβαση από το 0 σε 1 έχουμε μόνο για τους όρους 0110, 0001 και για το K_A , από 1 σε 0, για τον όρο 1001 και τον όρο 1110.

Θέτουμε στους χάρτες τους αδιάφορους όρους (AO) του *f-f* για το J_A και το K_A , αντίστοιχα.

Τέλος ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα (με τους λιγότερους δυνατών όρους και τις λιγότερες, κατά το δυνατόν, μεταβλητές ανά όρο). Οι αντίστοιχοι χάρτες Karnaugh και οι εξισώσεις, που προκύπτουν, είναι: $J_A = \overline{BCK} + B\overline{C}\overline{K}$ και $K_A = \overline{BCK} + B\overline{C}\overline{K}$.

B) Μεταβλητή B – Για το J_B και K_B

Θέτουμε, με το γνωστό τρόπο, στο χάρτη Karnaugh τους άσσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_B και για μεταβάσεις από το 1 σε 0 για το K_B). Κατόπιν θέτουμε στους χάρτες τους αδιάφορους όρους (AO) του $f-f$ για το J_B και το K_B , αντίστοιχα. Τέλος ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα.

Εύκολα προκύπτει ότι $J_B = \overline{CK} + C\overline{K}$ και $K_B = \overline{CK} + C\overline{K}$

	AB			
CK	00	01	11	10
00		AO	AO	
01	1	AO	AO	1
11		AO	AO	
10	1	AO	AO	1

	AB			
CK	00	01	11	10
00	AO			AO
01	AO	1	1	AO
11	AO			AO
10	AO	1	1	AO

C) Μεταβλητή C – Για το J_C και K_C

Θέτουμε στο χάρτη Karnaugh τους άσσους (προκύπτουν από τις μεταβάσεις από το 0 σε 1 για το J_C και για μεταβάσεις από το 1 σε 0 για το K_C). Κατόπιν θέτουμε στους χάρτες τους αδιάφορους όρους (AO) του $f-f$ για το J_C και το K_C , αντίστοιχα. Τέλος ομαδοποιούμε τους άσσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα. Εύκολα προκύπτει ότι $J_C = 1$ και $K_C = 1$.

	AB			
CK	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	AO	AO	AO	AO
10	AO	AO	AO	AO

	AB			
CK	00	01	11	10
00	AO	AO	AO	AO
01	AO	AO	AO	AO
11	1	1	1	1
10	1	1	1	1

Υλοποίηση του αμφίδρομου απαριθμητή με JK f-f και πύλες

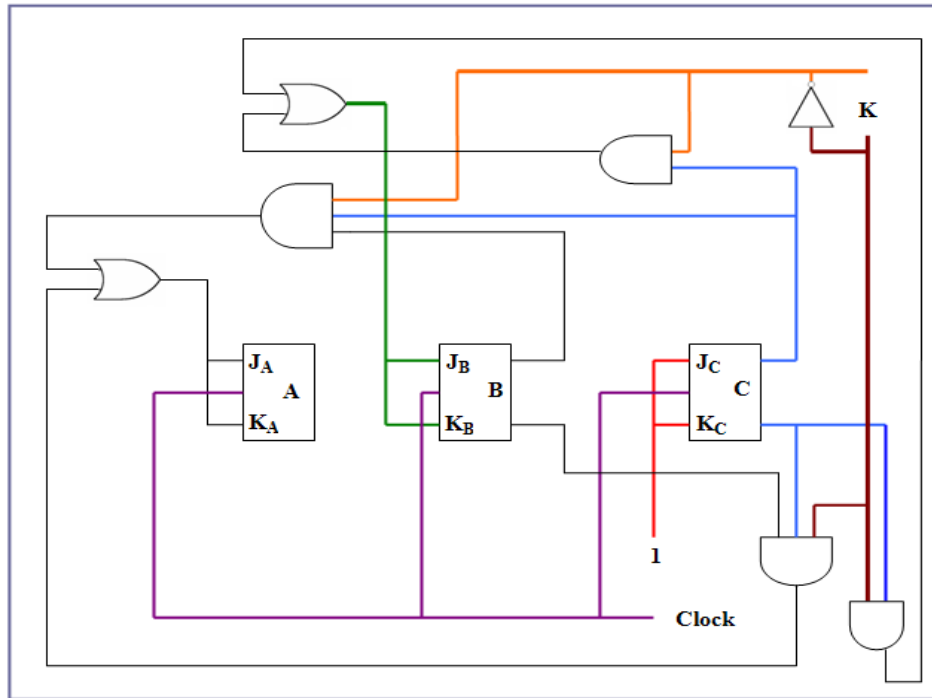
Οι προς υλοποίηση συναρτήσεις είναι:

$J_A = \overline{BCK} + B\overline{C}\overline{K}$ και $K_A = \overline{BCK} + B\overline{C}\overline{K}$

$J_B = \overline{CK} + C\overline{K}$ και $K_B = \overline{CK} + C\overline{K}$

$J_C = 1$ και $K_C = 1$

Το διάγραμμα, που προκύπτει, είναι το ακόλουθο (Σχήμα 6.38)



Σχήμα 6.38. Αμφίδρομος απαριθμητής με J-K f-f και πύλες

6.8. Μηχανές καταστάσεων

Θα εξετάσουμε τα σύγχρονα ακολουθιακά κυκλώματα, στα οποία η έξοδος δεν εξαρτάται μόνο από την παρούσα κατάσταση της εισόδου αλλά και από την κατάσταση στην οποία βρίσκεται το σύστημα πριν από την εφαρμογή της εισόδου. Όλες οι λειτουργίες είναι συγχρονισμένες, καθώς εφαρμόζεται ο παλμός του ρολογιού σε ιδιαίτερη είσοδο κάθε σύγχρονου ακολουθιακού κυκλώματος.

Στις περισσότερες περιπτώσεις, εφαρμόζουμε στην είσοδο μία σειριακή ακολουθία από δυαδικά ψηφία. Ένα σύγχρονο ακολουθιακό κύκλωμα αποτελείται από ένα συνδυαστικό κύκλωμα εισόδου, από ένα κύκλωμα μνήμης, που αποτελείται από f-f, και τέλος από ένα συνδυαστικό κύκλωμα εξόδου.

(Nelson, et al., 1995; Givone, 2002; Holdsworth, & Woods, 2002; Katz, 2005; Wakerly, 2006; Maxfield, 2009; Godse, & Godse, 2011; Mano, & Ciletti, 2014)

Για τη σχεδίαση των σύγχρονων ακολουθιακών κυκλωμάτων ακολουθούμε τα παρακάτω βήματα:

Βήμα 1

Όπως στο σχεδιασμό των συνδυαστικών κυκλωμάτων απαιτείται αρχικά ο πίνακας αλήθειας, έτσι και εδώ αρχικά απαιτείται ο αντίστοιχος πίνακας, ο πίνακας καταστάσεων. Στον πίνακα αυτό περιλαμβάνονται οι παρούσες καταστάσεις καθώς και οι επόμενες καταστάσεις του κυκλώματος, στις οποίες μεταβαίνει το κύκλωμα, σαν αποτέλεσμα της επίδρασης των εισόδων. Τέλος ο πίνακας περιλαμβάνει και τις εξόδους του κυκλώματος οι οποίες είναι συνάρτηση των εισόδων του.

Για την κατασκευή του πίνακα καταστάσεων πολύπλοκων συναρτήσεων αρχικά κατασκευάζεται το διάγραμμα καταστάσεων, όπου οι καταστάσεις παριστάνονται με κύκλους, οι οποίοι ενώνονται με γραμμές που αντιστοιχούν σε όλες τις πιθανές διαδρομές με τις οποίες πηγαίνουμε από μία κατάσταση σε κάποια άλλη με βάση την είσοδο. Υπάρχουν δύο τρόποι παράστασης των διαγραμμάτων καταστάσεων:

A) το διάγραμμα Mealy

Σε αυτό μέσα στο κύκλο σημειώνεται η κατάσταση, σε κάθε διαδρομή δε σημειώνονται και η είσοδος η οποία προκαλεί την αλλαγή της κατάστασης αλλά και η έξοδος του κυκλώματος.

B) το διάγραμμα Moore

Σε αυτό μέσα στο κύκλο σημειώνεται η κατάσταση αλλά και η έξοδος του κυκλώματος, σε κάθε διαδρομή δε, σημειώνεται μόνο η είσοδος η οποία προκαλεί την αλλαγή της κατάστασης.

Βήμα 2

Στο βήμα αυτό γίνεται η συγχώνευση ομοίων καταστάσεων, όπου δύο ή περισσότερες καταστάσεις ορίζονται σαν όμοιες, όταν έχουν για όλες τις ακολουθίες εισόδου τις ίδιες επόμενες καταστάσεις αλλά και την ίδια έξοδο. Έτσι προκύπτει ο τελικός πίνακας καταστάσεων, από τον οποίο μπορεί να κατασκευασθεί το τελικό διάγραμμα καταστάσεων.

Βήμα 3

Χρησιμοποιώντας τον τελικό πίνακα καταστάσεων, ακολουθεί η κωδικοποίηση με βάση κάποιο δυαδικό κώδικα, γεγονός που επιτρέπει την εξαγωγή των λογικών συναρτήσεων του συνδυαστικού κυκλώματος της εισόδου και της εξόδου, με βάση τις οποίες κατασκευάζεται το τελικό ακολουθιακό κύκλωμα. Η επιλογή του καταλληλότερου κώδικα, που γίνεται με τη χρήση διαφόρων μεθόδων, έχει σαν συνέπεια ένα οικονομικότερο κύκλωμα.

Με βάση τον πρώτο **κανόνα του Humphrey** (ή Armstrong) εάν δύο ή περισσότερες καταστάσεις έχουν τις ίδιες επόμενες καταστάσεις για όλες τις ακολουθίες εισόδου, τότε θα πρέπει να κωδικοποιηθούν έτσι ώστε να είναι γειτονικές, δηλαδή να διαφέρουν μόνο σε μία μεταβλητή. Επίσης καλό είναι να κωδικοποιούνται ως γειτονικές καταστάσεις, οι οποίες είναι επόμενες καταστάσεις μίας προηγούμενης κατάστασης. Κατόπιν κατασκευάζεται ο πίνακας μεταφοράς (transition table) ή πίνακας καταστάσεων.

Βήμα 4

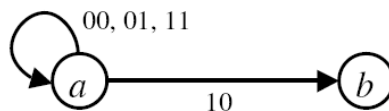
Οι εξισώσεις του τελικού κυκλώματος προκύπτουν μετά την επιλογή του $f-f$ από τον πίνακα μεταφοράς.

6.8.1. Μηχανές Mealy και Moore

Υπάρχουν δύο ειδών μηχανές καταστάσεων: α) Οι μηχανές Moore στις οποίες οι έξοδοι εξαρτώνται από τις τρέχουσες καταστάσεις καθώς και από τις εισόδους, οι οποίες συγχρονίζονται με το κύκλωμα. Οι απαριθμητές είναι ένα καλό παράδειγμα μηχανών Moore, και β) Οι μηχανές Mealy στις οποίες οι έξοδοι προσδιορίζονται τόσο από τις τρέχουσες καταστάσεις όσο και από τις εισόδους, οι οποίες δε συγχρονίζονται με το κύκλωμα.

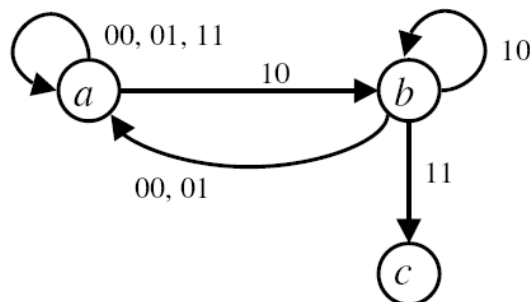
6.8.1.1 Μηχανή Καταστάσεων Moore

Έστω ότι θέλουμε να ανιχνεύσουμε την ακολουθία 10, 11, 00, 10. Το κύκλωμά μας πρέπει να έχει δύο εισόδους, ώστε να παράγονται τέσσερις καταστάσεις. Ξεκινάμε από μία κατάσταση a , την οποία δεν έχουμε προσδιορίσει ακόμα, και μεταβαίνουμε στη b , εάν λάβουμε την ακολουθία 10. Σε κάθε άλλη περίπτωση, το κύκλωμα παραμένει στην κατάσταση a . (Σχήμα 6.39)



Σχήμα 6.39. Αρχική μετάβαση της μηχανής Moore

Το κύκλωμα μεταβαίνει από την κατάσταση b στην κατάσταση c όταν ληφθεί το 11. Πιθανή λήψη του 10 επιτρέπει την παραμονή του κυκλώματος στην κατάσταση b . (Σχήμα 6.40)

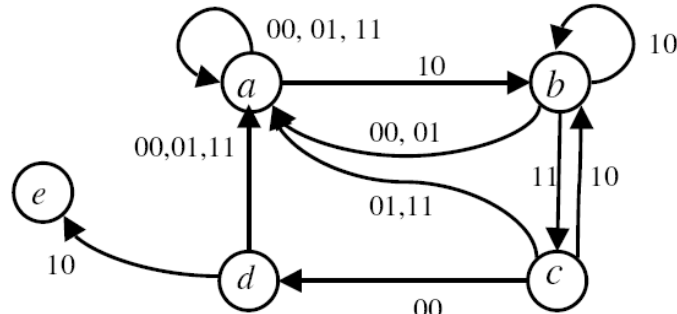


Σχήμα 6.40. Πιθανές μεταβάσεις των καταστάσεων a και b

Σχεδίαση του μοντέλου ανίχνευσης ακολουθίας κατά Moore

α) Διάγραμμα Καταστάσεων

Έστω ότι θέλουμε να ανιχνεύσουμε την ακολουθία 10, 11, 00, 10,..... Οι εισοδοί μας είναι έστω X και Y. Ακολουθώντας την προηγούμενη μεθοδολογία, σχεδιάζουμε το διάγραμμα Moore ως ακολούθως (Σχήμα 6.41):



Σχήμα 6.41. Διάγραμμα καταστάσεων της μηχανής Moore

β) Πίνακας παρούσας και επόμενης κατάστασης

Με βάση την τιμή των εισόδων X και Y έχουμε τον πίνακα που ακολουθεί για την παρούσα και την επόμενη κατάσταση.

Παρούσα κατάσταση	Είσοδοι XY	Επόμενη κατάσταση			
		00	01	11	10
a		a	a	a	b
b		a	a	c	b
c		d	a	a	b
d		a	a	a	e
e		X	X	X	X

Πίνακας 6.32. Πίνακας καταστάσεων της μηχανής Moore

Έχουμε πέντε καταστάσεις και, συνεπώς, απαιτούνται 3 f-f για την αναπαράστασή τους. Η υλοποίηση της κωδικοποίησης θα γίνει με τον κώδικα Gray. Ο πίνακας κωδικοποίησης, ώστε οι όροι να είναι γειτονικοί, είναι ο ακόλουθος: (η κωδικοποίηση είναι τυχαία)

N	WR			
	00	01	11	10
0	a	d	e	
1	b	c		

Πίνακας 6.33α. Πίνακας κωδικοποίησης της μηχανής Moore

Παρούσα κατάσταση	Είσοδοι XY	Επόμενη κατάσταση			
		00	01	11	10
a(000)		a(000)	a(000)	a(000)	b(001)
b(001)		a(000)	a(000)	c(011)	b(001)
c(011)		d(010)	a(000)	a(000)	b(001)
d(010)		a(000)	a(000)	a(000)	e(110)
e(110)		X	X	X	X

Πίνακας 6.33β. Πίνακας καταστάσεων της μηχανής Moore με χρήση της κωδικοποίησης του πίνακα 6.33

Παρούσα κατάσταση	Είσοδοι	Επόμενη κατάσταση			
	XY	00	01	11	10
000		000	000	000	001
001		000	000	011	001
011		010	000	000	001
010		000	000	000	110
110		X	X	X	X

Πίνακας 6.33γ. Πίνακας καταστάσεων μετά την κωδικοποίηση

γ) Πίνακας καταστάσεων του $f-f$

Θα χρησιμοποιήσουμε για την υλοποίηση το D -flip-flop. Ο πίνακας καταστάσεων είναι ο ακόλουθος: (Σημειώνονται όλες οι δυνατές μεταβάσεις)(Πίνακας 6.34).

Q		Q'	D
0	→	0	0
0	→	1	1
1	→	0	0
1	→	1	1

Πίνακας 6.34. Πίνακας μεταβάσεων του D -f-f

δ, ε) Απλοποίηση με χάρτη Karnaugh και Boolean εκφράσεις

Είναι προφανές ότι για κάθε είσοδο πρέπει να έχουμε τον αντίστοιχο χάρτη Karnaugh.

Από τον πίνακα καταστάσεων, και παρατηρώντας τις μεταβάσεις από 0 σε 1 και από 1 σε 1, όπου το D είναι 1, έχουμε έναν άσσο και 4 αδιάφορους όρους. (κοιτάω την κατάληξη (επόμενη κατάσταση) να είναι σε 1 καθώς είναι άσσος, όταν έχω μετάβαση από οτιδήποτε σε ένα). (Έχουμε ένα χάρτη Karnaugh πέντε μεταβλητών- οι τρεις μεταβλητές της κωδικοποίησης των καταστάσεων και οι δύο είσοδοι-).

		\bar{Y}				Y			
		WR		NX		WR		NX	
		00	01	11	10	00	01	11	10
00				x	x			x	x
01			1	x	x			x	x
11				x	x			x	x
10				x	x			x	x

Από τον παραπάνω χάρτη προκύπτει ότι: $D_0 = R\bar{N}X\bar{Y}$

Είναι προφανές ότι η χρήση του D f-f, το οποίο δεν έχει αδιάφορους όρους, μας οδηγεί σε μεγαλύτερες εκφράσεις Boolean και, συνεπώς, σε πολυπλοκότερα κυκλώματα.

Για τη δεύτερη μεταβλητή έχουμε: (άσσοι είναι οι καταστάσεις 01100, 00111, 01010). Από το χάρτη που ακολουθεί προκύπτει ότι: $D_1 = R\bar{N}X\bar{Y} + RN\bar{X}\bar{Y} + \bar{R}NXY$.

		\bar{Y}				Y				
		WR				WR				
NX		00	01	11	10		00	01	11	10
	00			x	x				x	x
	01		1	x	x				x	x
	11			x	x		1		x	x
	10		1	x	x				x	x

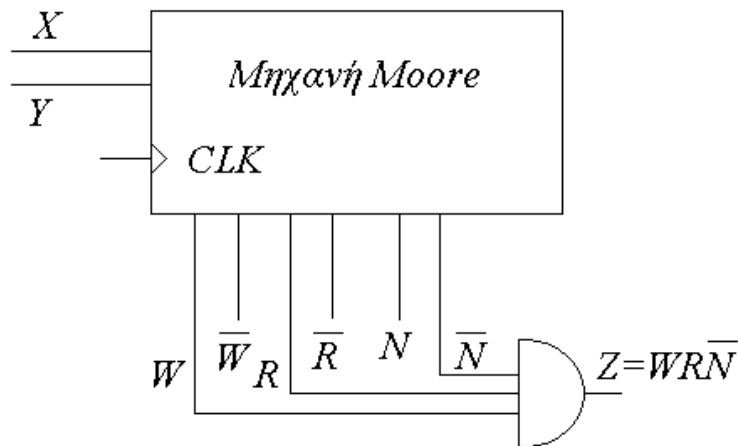
Για την τρίτη μεταβλητή έχουμε: (άσσοι είναι οι καταστάσεις 00111, 00010, 00110,01110)

		\bar{Y}				Y				
		WR				WR				
NX		00	01	11	10		00	01	11	10
	00			x	x				x	x
	01	1		x	x				x	x
	11	1	1	x	x		1		x	x
	10			x	x				x	x

Από τον παραπάνω χάρτη πέντε μεταβλητών προκύπτει ότι: $D_2 = \bar{R}X\bar{Y} + NX\bar{Y} + \bar{R}NX$.

στ) Υλοποίηση

Η υλοποίηση γίνεται κατευθείαν από τις Boolean εκφράσεις. Το μόνο που έχουμε να κάνουμε είναι να συνδέσουμε στην έξοδο τα αντίστοιχα W, R, N. Η συνάρτηση εξόδου είναι η κατάσταση e (110), δηλαδή η $Z = WR\bar{N}$.



Σχήμα 6.42. Απλοποιημένο δομικό διάγραμμα μηχανής Moore

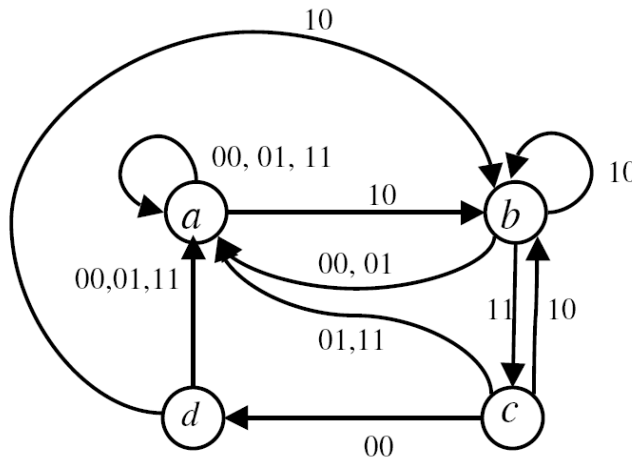
6.8.1.2 Μηχανή Καταστάσεων Mealy

Έστω ότι θέλουμε να ανιχνεύσουμε την ίδια ακολουθία 10, 11, 00, 10 όπως στην προηγούμενη περίπτωση, χρησιμοποιώντας τώρα τη μηχανή Mealy. Το κύκλωμά μας πρέπει να έχει δύο εισόδους ώστε να παράγονται τέσσερις καταστάσεις. Ξεκινάμε από μία κατάσταση a την οποία δεν έχουμε προσδιορίσει ακόμα και μεταβαίνουμε στη b , εάν λάβουμε την ακολουθία 10. Σε κάθε άλλη περίπτωση το κύκλωμα παραμένει στην κατάσταση a . Με τον ίδιο τρόπο σκέψης συμπληρώνουμε το διάγραμμα καταστάσεων (Σχήμα 6.33).

Σχεδίαση του μοντέλου ανίχνευσης ακολουθίας κατά Mealy

α) Διάγραμμα Καταστάσεων

Οι εισοδοί μας είναι και πάλι έστω X και Y . Ακολουθώντας την προηγούμενη μεθοδολογία, σχεδιάζουμε το διάγραμμα Mealy ως ακολούθως:



Σχήμα 6.43. Διάγραμμα καταστάσεων της μηχανής Mealy

β) Πίνακας παρούσας και επόμενης κατάστασης

Με βάση την τιμή των εισόδων X και Y έχουμε τον πίνακα που ακολουθεί για την παρούσα και την επόμενη κατάσταση.

Παρούσα κατάσταση	Είσοδοι XY	Επόμενη κατάσταση			
		00	01	11	10
w		w	w	w	r
r		w	w	n	r
n		m	w	w	r
m		w	w	w	r

Πίνακας 6.35α. Πίνακας καταστάσεων της μηχανής Mealy

Έχουμε τέσσερις καταστάσεις και, συνεπώς, απαιτούνται 2 f - f για την αναπαράστασή τους. Η υλοποίηση της κωδικοποίησης θα γίνει, και πάλι, με τον κώδικα Gray.

Παρούσα κατάσταση	Είσοδοι XY	Επόμενη κατάσταση			
		00	01	11	10
w(00)		w(00)	w(00)	w(00)	r(01)
r(01)		w(00)	w(00)	n(11)	r(01)
n(11)		m(10)	w(00)	w(00)	r(01)
m(10)		w(00)	w(00)	w(00)	r(01)

Πίνακας 6.35β. Πίνακας καταστάσεων της μηχανής Mealy με γειτονική κωδικοποίηση

	Είσοδοι	Επόμενη κατάσταση			
Παρούσα κατάσταση	XY	00	01	11	10
00		00	00	00	01
01		00	00	11	01
11		10	00	00	01
10		00	00	00	01

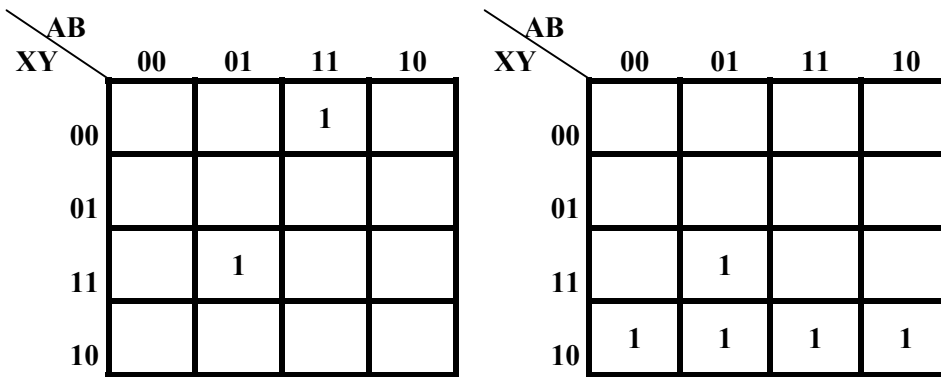
Πίνακας 6.35γ. Πίνακας καταστάσεων μετά την κωδικοποίηση

γ) Πίνακας καταστάσεων του *Flip-flop*

Θα χρησιμοποιήσουμε και σε αυτή την περίπτωση το D-f-f.

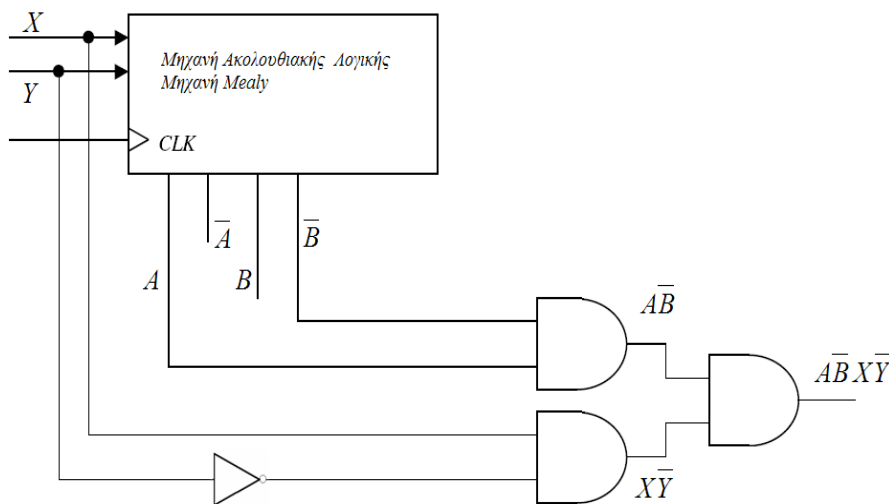
δ, ε) Απλοποίηση με χάρτη Karnaugh και Boolean έκφρασεις

Είναι προφανές ότι για κάθε είσοδο πρέπει να έχουμε τον αντίστοιχο χάρτη Karnaugh. Παρατηρούμε ότι έχουμε ένα χάρτη 4 μεταβλητών, καθώς η κωδικοποιημένη κατάσταση και η είσοδος είναι συναρτήσεις δύο μεταβλητών.



Από τους παραπάνω χάρτες (δεν υπάρχουν αδιάφοροι όροι) προκύπτει ότι: $D_0 = AB\bar{X}\bar{Y} + \bar{A}BXY$ (αριστερός χάρτης). Για τη δεύτερη μεταβλητή (δεξιός χάρτης) προκύπτει ότι: $D_1 = \bar{A}BX + X\bar{Y}$

στ) Υλοποίηση



Σχήμα 6.44. Δομικό Διάγραμμα της μηχανής Mealy

Η υλοποίηση γίνεται κατευθείαν από τις Boolean εκφράσεις. Το μόνο που έχουμε να κάνουμε είναι να ανιχνεύσουμε την παρουσία της εισόδου 1,0, όταν το κύκλωμα βρίσκεται στην κατάσταση d. Η συνάρτηση εξόδου είναι $Z = \overline{A}Bx\overline{y}$ Το δομικό διάγραμμα, που απεικονίζει το συνδυαστικό λογικό κύκλωμα, είναι το ακόλουθο (Σχήμα 6.44)

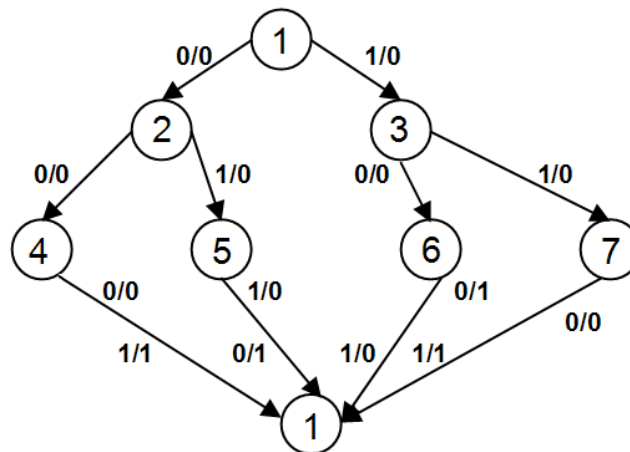
6.8.2. Υποδειγματικά λυμένες ασκήσεις

6.8.2.1 Άσκηση 5^η Σχεδίαση ακολουθιακού κυκλώματος

Να σχεδιάσετε σύγχρονο ακολουθιακό κύκλωμα του οποίου η έξοδος είναι 1, εάν το πλήθος των άσων που περιέχονται στη δυαδική πληροφορία είναι περιττό. Να θεωρήσετε ότι η αρχική πληροφορία αποτελείται από 3 δυαδικά ψηφία. Για την υλοποίηση να χρησιμοποιηθούν J-K ff.

Με βάση τα βήματα, που έχουμε προαναφέρει, αρχίζουμε με το διάγραμμα καταστάσεων Mealy και αρχίζουμε από την κατάσταση 1, και ανάλογα με την επόμενη είσοδο σχεδιάζουμε το διάγραμμα. (Σχήμα 6-55).

Μετά το τρίτο δυαδικό ψηφίο το κύκλωμα επανέρχεται στην κατάσταση 1 (αρχική κατάσταση) και είναι έτοιμο για την επόμενη ακολουθία εισόδου. Ο τρόπος με τον οποίο έχουμε κατασκευάσει το διάγραμμα είναι προφανής, και οι εξοδοί είναι 1 μόνο στην περίπτωση στην οποία, από το σύνολο των τριών άσων που εισέρχονται το πλήθος τους είναι άρτιο (εδώ δύο από αυτούς).



Σχήμα 6.45. Δομικό Διάγραμμα της μηχανής Mealy

Από το παραπάνω διάγραμμα φτιάχνουμε τον πίνακα καταστάσεων που ακολουθεί (Πίνακας 6.36)

Παρούσα Κατάσταση	Είσοδος x	Επόμενες καταστάσεις		Έξοδος Z	
		0	1	0	1
1		2	3	0	0
2		4	5	0	0
3		6	7	0	0
4		1	1	0	1
5		1	1	1	0
6		1	1	1	0
7		1	1	0	1

Πίνακας 6.36. Πίνακας καταστάσεων

Αρχικά εξετάζουμε αν υπάρχουν όμοιες καταστάσεις: Εύκολα διαπιστώνουμε ότι η κατάσταση 5 είναι όμοια με την κατάσταση 6 και, συνεπώς, συγχωνεύουμε την κατάσταση 6 ($5 \equiv 6$) και η κατάσταση 4 είναι όμοια με την κατάσταση 7 και, συνεπώς, συγχωνεύουμε την κατάσταση 7 ($4 \equiv 7$).

Παρούσα Κατάσταση	Είσοδος x:	Επόμενες καταστάσεις		Έξοδος Z	
		0	1	0	1
1		2	3	0	0
2		4	5	0	0
3		6	7	0	0
4		1	1	0	1
5		1	1	1	0

Το επόμενο βήμα είναι να κωδικοποιήσουμε τις καταστάσεις, εφαρμόζοντας τον κανόνα του Hamphrey. Έτσι οι καταστάσεις 1, 2, 3 θα κωδικοποιηθούν γειτονικά, όπως και οι καταστάσεις 4 και 5.

		AB			
		00	01	11	10
C	0	1	2	3	
	1		4	5	

Με βάση την παραπάνω κωδικοποίηση (η οποία είναι τυχαία) έχουμε:

Παρούσα Κατάσταση	Είσοδος x:	Επόμενες καταστάσεις		Έξοδος Z	
		0	1	0	1
000		010	110	0	0
010		011	111	0	0
110		111	011	0	0
011		000	000	0	1
111		000	000	1	0

Χρησιμοποιώντας και τους αδιάφορους όρους του πίνακα αλήθειας (εκτός των αδιάφορων όρων του f), στους χάρτες Karnaugh που ακολουθούν εύκολα προκύπτει ότι $J_A = \overline{C}x$ και $K_A = C+x$

		AB			
		00	01	11	10
Cx	00			AO	x
	01	1	1	AO	x
	11	x		AO	x
	10	x		AO	x

		AB			
		00	01	11	10
Cx	00	AO	AO		x
	01	AO	AO	1	x
	11	x	AO	1	x
	10	x	AO	1	x

Για τη μεταβλητή B εύκολα προκύπτει $J_B = 1$ και η $K_B = C$

	AB	00	01	11	10
Cx		00	01	11	10
	00	1	AO	AO	x
	01	1	AO	AO	x
	11	x	AO	AO	x
	10	x	AO	AO	x

	AB	00	01	11	10
Cx		00	01	11	10
	00	AO			x
	01	AO			x
	11	x	1	1	x
	10	x	1	1	x

	AB	00	01	11	10
Cx		00	01	11	10
	00		1	1	x
	01		1	1	x
	11	x	AO	AO	x
	10	x	AO	AO	x

	AB	00	01	11	10
Cx		00	01	11	10
	00	AO	AO	AO	x
	01	AO	AO	AO	x
	11	x	1	1	x
	10	x	1	1	x

Για τη μεταβλητή C εύκολα προκύπτει η $J_C = B$ και η $K_C = 1$

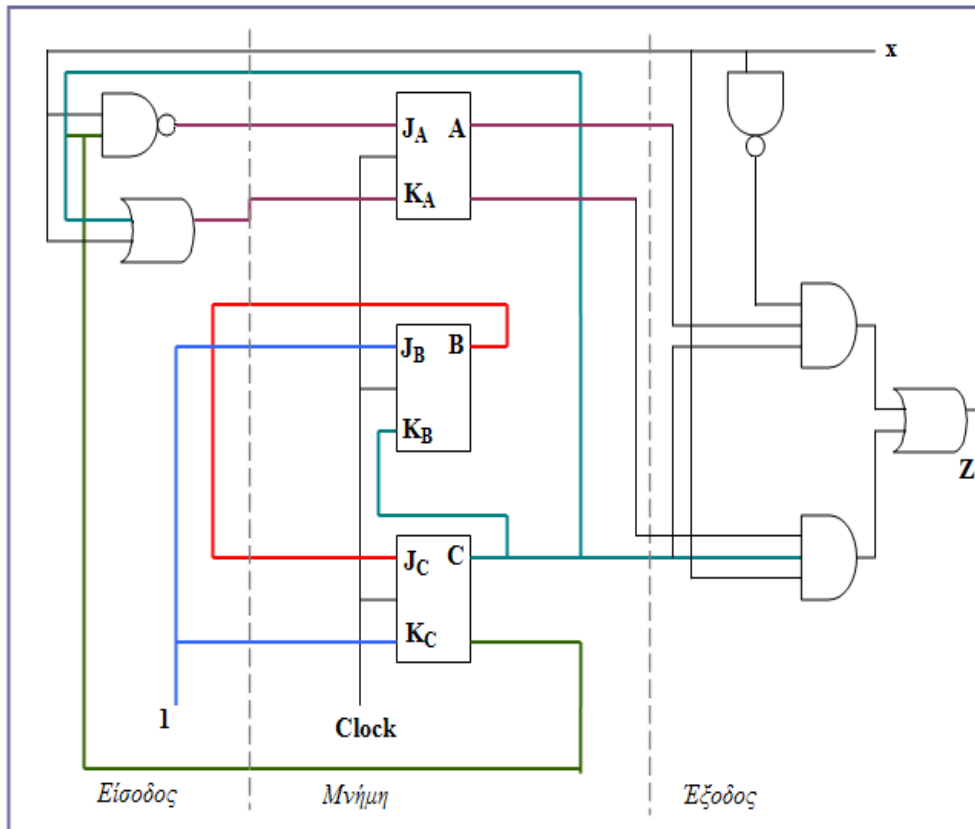
Τέλος για την έξοδο Z του κυκλώματός μας εύκολα προκύπτει ότι:

$$Z = \bar{A}Cx + AC\bar{x}$$

	AB	00	01	11	10
Cx		00	01	11	10
	00				x
	01				x
	11	x	1		x
	10	x		1	x

Υλοποίηση με JK flip-flop και πύλες

Με βάση τις παραπάνω εξισώσεις, και με δεδομένο ότι έχουμε τρία επίπεδα στην υλοποίηση, το κύκλωμα εισόδου, τη μνήμη και το κύκλωμα εξόδου, έχουμε το κύκλωμα του Σχήματος 6.46.



Σχήμα 6.46. Υλοποίηση της μηχανής Mealy με J-K f-f και πύλες

6.8.2.2 Άσκηση 6^η Αναγνώριση μορφών

Η αναγνώριση μορφών γίνεται με ένα ειδικό ακολουθιακό κύκλωμα, το οποίο ψάχνει για μία συγκεκριμένη μορφή(ακολουθία) από bit σε κάποια είσοδο.

Το κύκλωμα αναγνώρισης μορφών έχει μόνο μία είσοδο, την X. Ένα bit εισόδου παρέχεται σε κάθε κύκλο ρολογιού. Για παράδειγμα, θα χρειαζόταν 20 κύκλους, για να σαρώσει μία είσοδο των 20-bit.

Αυτός είναι ένας εύκολος τρόπος να επιτρέψουμε αυθαίρετα μεγάλες ακολουθίες στην είσοδο. Υπάρχει μια έξοδος, η Z, που γίνεται 1, όταν ανιχνευθεί η συγκεκριμένη μορφή από bit.

Στο παράδειγμα μας πρέπει να ανιχνευθεί η μορφή «1001»:

Είσοδοι: 11100110100100110 ...

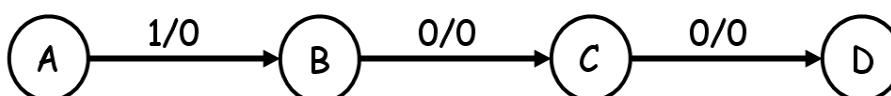
Έξοδοι: 00000100000100100 ...

Σε κάθε κύκλο ρολογιού, εμφανίζονται ένα bit εισόδου και ένα bit εξόδου. Αυτό απαιτεί ένα ακολουθιακό κύκλωμα, διότι το κύκλωμα πρέπει να «θυμάται» τις εισόδους από προηγούμενους κύκλους ρολογιού, προκειμένου να καθορίσει εάν ταυτίστηκε μια ομάδα bit με την προς ανίχνευση μορφή, ή όχι.

Βασικό διάγραμμα καταστάσεων

Ποια κατάσταση χρειαζόμαστε για το σύστημα αναγνώρισης μορφών; Πρέπει να «θυμόμαστε» εισόδους από προηγούμενους κύκλους ρολογιού.

Για παράδειγμα, αν οι προηγούμενες τρεις εισόδους ήταν 100 και η τρέχουσα είσοδος είναι 1, τότε η έξοδος θα είναι 1. Γενικά, θα πρέπει να θυμόμαστε την εμφάνιση τμημάτων της επιθυμητής μορφής – σε αυτή την περίπτωση, 1, 10, και 100. Ας αρχίσουμε με το βασικό διάγραμμα καταστάσεων: (Σχήμα 6.47)



Σχήμα 6.47 Βασικό διάγραμμα καταστάσεων

Στον πίνακα 6.37, που ακολουθεί εκφράζουμε κάθε μία κατάσταση, από αυτές που μας οδηγούν στην επιθυμητή με ένα σύμβολο.

Κατάσταση	Σημασία
A	Καμία από τις επιθυμητές μορφές (1001) δεν έχει εισαχθεί ακόμα
B	Έχουμε ήδη πάρει το πρώτο ψηφίο (1) της επιθυμητής μορφής
C	Έχουμε ήδη πάρει τα δύο πρώτα ψηφία (10) της επιθυμητής μορφής
D	Έχουμε ήδη πάρει τα τρία ψηφία (100) της επιθυμητής μορφής

Πίνακας 6.37. Ερμηνευτικός πίνακας των καταστάσεων

Διαδικασία σχεδιασμού

Βήμα 1: Φτιάχνουμε έναν πίνακα καταστάσεων, βασισμένο στην εκφώνηση του προβλήματος. Ο πίνακας θα πρέπει να δείχνει: παρούσες καταστάσεις, εισόδους, επόμενες καταστάσεις και εξόδους. (ίσως να είναι ευκολότερο να βρούμε πρώτα το διάγραμμα καταστάσεων, κι έπειτα να το μετατρέψουμε σε πίνακα.)

Βήμα 2: Εκχωρούμε τους δυαδικούς κώδικες στις καταστάσεις του πίνακα καταστάσεων. Αν έχουμε n καταστάσεις, οι δυαδικοί κώδικες θα έχουν το λιγότερο $\lceil \log_2 n \rceil$ ψηφία, και το κύκλωμα μας θα έχει το λιγότερο $\lceil \log_2 n \rceil f$ -fs.

Βήμα 3: Για κάθε f -f, και για κάθε γραμμή του πίνακα καταστάσεων, βρίσκουμε τις τιμές εισόδων των f -f, που χρειάζονται, για να παράγουν από την παρούσα κατάσταση την επόμενη κατάσταση. Εδώ μπορούμε να χρησιμοποιήσουμε τους πίνακες διέγερσης των f -fs.

Βήμα 4: Πρέπει να βρούμε απλούστερες εξισώσεις για τις εισόδους και τις εξόδους των f -fs.

Βήμα 5: Κατασκευάζουμε το κύκλωμα.

Δημιουργία του πίνακα καταστάσεων (Βήμα 1)

Το πρώτο που πρέπει να καταλάβουμε είναι πώς ακριβώς η γνώση των καταστάσεων θα μας βοηθήσει στην επίλυση του δεδομένου προβλήματος.

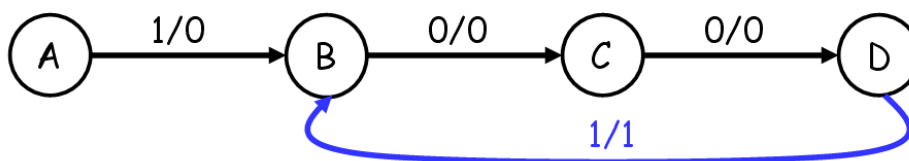
Φτιάχνουμε ένα πίνακα καταστάσεων, βασισμένο στην εκφώνηση του προβλήματος. Ο πίνακας θα πρέπει να παρουσιάζει: παρούσες καταστάσεις, εισόδους, επόμενες καταστάσεις και εξόδους. Μερικές φορές είναι πιο εύκολο να βρούμε πρώτα το διάγραμμα καταστάσεων και έπειτα να το μετατρέψουμε σε πίνακα καταστάσεων.

Αυτό είναι, συνήθως, και το πιο δύσκολο βήμα. Αν έχουμε τον πίνακα καταστάσεων η υπόλοιπη διαδικασία του σχεδιασμού είναι ίδια για όλα τα ακολουθιακά κυκλώματα. Τα κυκλώματα αναγνώρισης μορφών είναι το δυσκολότερο παράδειγμα που θα δούμε, και είναι σημαντικό να το καταλάβουμε.

Επικαλυπτόμενες καταστάσεις της επιθυμητής μορφής

Τι θα συμβεί, αν βρεθούμε στην κατάσταση D (δηλ. οι τρεις τελευταίες εισοδοί ήταν 100), και η τρέχουσα είσοδος είναι 1; (Σχήμα 6.48)

Τότε η έξοδος θα είναι 1, επειδή έχουμε βρει την επιθυμητή μορφή. Αλλά αυτός ο τελευταίος 1 θα μπορούσε, επίσης, να είναι η αρχή μιας νέας εμφάνισης της επιθυμητής μορφής! Για παράδειγμα, η 1001001 περιέχει δύο εμφανίσεις της 1001. Για να εντοπίσουμε τις επικαλυπτόμενες καταστάσεις της μορφής, η επόμενη κατάσταση θα πρέπει να είναι η B.

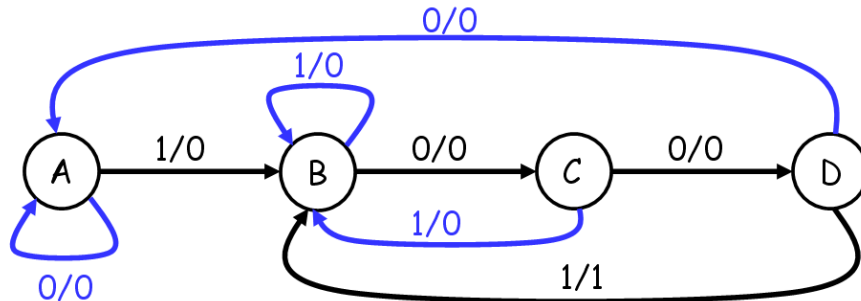


Σχήμα 6.48. Διάγραμμα καταστάσεων

Συμπληρώνοντας τις υπόλοιπες μεταβάσεις

Πρέπει να θυμηθούμε πως χρειαζόμαστε δύο εξερχόμενα βέλη για κάθε κόμβο, για να υπολογίσουμε τις πιθανότητες, του $X=0$ και του $X=1$.

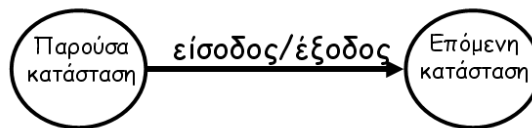
Τα εναπομείναντα βέλη που χρειαζόμαστε σχεδιάστηκαν πάνω στο αρχικό διάγραμμα με γκρι χρώμα και επιτρέπουν, επίσης, τη σωστή ανίχνευση των επικαλυπτόμενων εμφανίσεων του 1001.



Σχήμα 6.49. Ολοκλήρωση του διαγράμματος καταστάσεων

Συμπλήρωση του πίνακα καταστάσεων

Πρέπει να θυμηθούμε πώς σχετίζονται τα βέλη του διαγράμματος καταστάσεων (Σχήμα 6-60) με τις γραμμές του πίνακα καταστάσεων (πίνακας 6-35):



Σχήμα 6.50. Συσχετισμός καταστάσεων

Παρούσα Κατάσταση	Είσοδος	Επόμενη κατάσταση	Έξοδος Z
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	B	1

Πίνακας 6.38. Πίνακας Καταστάσεων

Είναι προφανές ότι ο ανωτέρω πίνακας αποδίδει ανάλογα με την παρούσα κατάσταση και την είσοδο την επόμενη κατάσταση, όπως αυτές προκύπτουν και από το πλήρες διάγραμμα του σχήματος 6.49.

Εκχώρηση δυαδικών κωδικών στις καταστάσεις (Βήμα 2)

Έχουμε τέσσερις καταστάσεις τις A, B, C, D, και έτσι χρειαζόμαστε το λιγότερο δύο $f_1 f_0$ $Q_1 Q_0$.

Είναι εύκολο να παρουσιάσουμε την κατάσταση A με $Q_1 Q_0 = 00$, την B με 01, την C με 10, και τέλος την D με 11. (Πίνακας 6.39). Η εκχώρηση των καταστάσεων μπορεί να έχει μεγάλο αντίκτυπο στην πολυπλοκότητα του κυκλώματος, αλλά αυτό δε θα μας απασχολήσει σε αυτή την φάση.

Παρούσα Κατάσταση	Είσοδος	Επόμενη κατάσταση	Έξοδος Z		Παρούσα Κατάσταση $Q_1 Q_0$	Είσοδος	Επόμενη Κατάσταση $Q_1 Q_0$	Έξοδος Z
A	0	A	0	→	0 0	0	0 0	0
A	1	B	0		0 0	1	0 1	0
B	0	C	0		0 1	0	1 0	0
B	1	B	0		0 1	1	0 1	0
C	0	D	0		1 0	0	1 1	0
C	1	B	0		1 0	1	0 1	0
D	0	A	0		1 1	0	0 0	0
D	1	B	1		1 1	1	0 1	1

Πίνακας 6.39. Καταστάσεις και κωδικοποίηση αυτών

Εύρεση των τιμών εισόδων των flip - flop (Βήμα 3)

Κατόπιν πρέπει να καταλάβουμε πώς θα κάνουμε τα $f-f$ s από την παρούσα κατάστασή τους να μεταβούν στην επιθυμητή επόμενη κατάσταση.

Αυτό εξαρτάται από το είδος των $f-f$ s που χρησιμοποιούμε. Θα χρησιμοποιήσουμε δύο J-K. Για κάθε $f-f$ Q_i , πρέπει να κοιτάξουμε την παρούσα και την επόμενη κατάσταση, και να προσδιορίσουμε ποιες θα πρέπει είναι οι εισοδοί J_i και K_i , για να δημιουργηθεί η αλλαγή της κατάστασης.

Ένας πίνακας διέγερσης δείχνει ποιες εισοδοί του $f-f$ απαιτούνται, για να επιτύχουμε την επιθυμητή μεταβολή της κατάστασης.

Παρούσα Κατάσταση	Είσοδος	Επόμενη Κατάσταση	Είσοδοι των $f-f$				Έξοδος Z
$Q_1 Q_0$		$Q_1 Q_0$	J_1	K_1	J_0	K_0	
0 0	0	0 0	0	X	0	X	0
0 0	1	0 1	0	X	1	X	0
0 1	0	1 0	1	X	X	1	0
0 1	1	0 1	0	X	X	0	0
1 0	0	1 1	X	0	1	X	0
1 0	1	0 1	X	1	1	X	0
1 1	0	0 0	X	1	X	1	0
1 1	1	0 1	X	1	X	0	1

Πίνακας 6.40. Είσοδοι των flip-flop και έξοδος

Ο πίνακας παρέχει την ίδια πληροφορία με το χαρακτηριστικό πίνακα. Αν η παρούσα κατάσταση ενός JK $f-f$ είναι 0 και θέλουμε στην επόμενη κατάσταση να έχουμε 1, τότε έχουμε δύο επιλογές για τις εισόδους του JK:

Μπορούμε να χρησιμοποιήσουμε την JK=10, για να θέσουμε την επόμενη κατάσταση του $f-f$ σε 1. Μπορούμε, επίσης, να χρησιμοποιήσουμε την JK=11, για να έχουμε ως αποτέλεσμα το συμπλήρωμα της παρούσας κατάστασης 0 (οπότε 1).

Έτσι, για να αλλάξουμε από 0 σε 1, πρέπει να θέσουμε J=1 και το K μπορεί να είναι είτε 0 είτε 1. Ομοίως, και οι υπόλοιπες πιθανές μεταβάσεις καταστάσεων μπορούν να γίνουν εξίσου καλά με δύο διαφορετικούς τρόπους.

Μπορούμε τώρα να χρησιμοποιήσουμε τον πίνακα διέγερσης του JK, για να βρούμε τις σωστές τιμές εισόδων για κάθε $f-f$, βασισμένοι στην παρούσα και την επόμενη κατάσταση. (πίνακας 6.40).

Εύρεση των εξισώσεων εισόδων και εξόδου των $f-f$ (Βήμα 4)

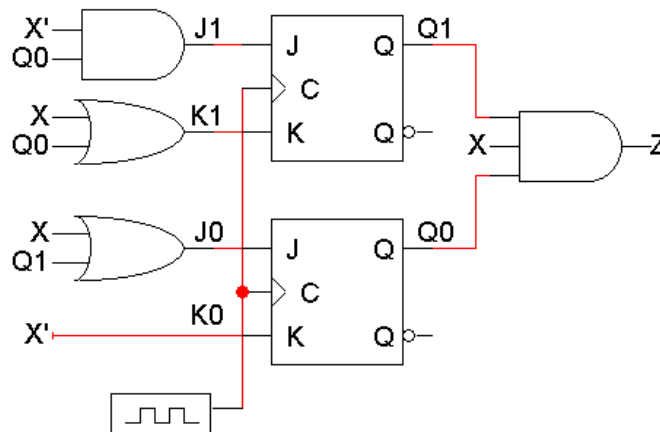
Μπορούμε πλέον να κάνουμε χάρτες Karnaugh και να βρούμε τις εξισώσεις για καθεμία από τις τέσσερις εισόδους των $f-f$, καθώς και για την έξοδο Z.

Αυτές οι εξισώσεις γράφονται συναρτήσει της παρούσας κατάστασης και των εισόδων. Το πλεονέκτημα της χρήσης των JK *f-fs* είναι πως υπάρχουν πολλοί αδιάφοροι όροι, κάτι που μπορεί να οδηγήσει σε απλούστερες εξισώσεις. (Πίνακας 6.40)

Οι εξισώσεις είναι: $J_1 = X' Q_0$, $K_1 = X + Q_0$ και $J_0 = X + Q_1$, $K_0 = X'$
 και για την έξοδο $Z = Q_1 Q_0 X$

Κατασκευή του κυκλώματος (Βήμα 5)

Τέλος θα χρησιμοποιήσουμε τις απλουστευμένες εξισώσεις για να σχεδιάσουμε το κύκλωμά μας.(Σχήμα 6.51)



Σχήμα 6.51. Υλοποίηση του κυκλώματος με J-K *f-f* και πύλες

Λιαδικασία σχεδιασμού με D flip – flop

Τι θα γίνει, αν θέλουμε να κατασκευάσουμε το ίδιο κύκλωμα, χρησιμοποιώντας D *f-fs*;

Γνωρίζουμε ήδη τον πίνακα καταστάσεων με τις εκχωρήσεις των καταστάσεων, και έτσι μπορούμε να αρχίσουμε από το βήμα 3, βρίσκοντας τις τιμές των εισόδων για τα *f-fs*. Τα D *f-fs* έχουν μόνο μια είσοδο, και έτσι ο πίνακας μας, χρειάζεται μόνο δύο στήλες, για το D1 και το D0. Αναφέρουμε εκ νέου τον πίνακα καταστάσεων του D *f-f* στον οποίο έχουμε συμπληρώσει και την λειτουργία. (Πίνακας 6.41).

D	Q _(n)	Q _(n+1)	Λειτουργία
0	0	0	Reset
0	1	0	Reset
1	0	1	Set
1	1	1	Set

Πίνακας 6.41. Πίνακας καταστάσεων του D *f-f*(επανξιημένος)

Βήμα 3: Τιμές εισόδων για τα D *f-fs*

Ο πίνακας διέγερσης του D *f-f* είναι αρκετά βαρετός: θέτουμε την είσοδο D στην τιμή που θα πρέπει να έχει η επόμενη κατάσταση. Δε χρειάζεται να φαίνονται χωριστές στήλες για τα D1 και D0. Μπορούμε απλά να χρησιμοποιήσουμε τις στήλες των επομένων καταστάσεων. (πίνακας 6.42)

Παρούσα Κατάσταση		Είσοδος	Επόμενη Κατάσταση		Είσοδοι των f - f		Εξόδος Z
Q_1	Q_0		Q_1	Q_0	D_1	D_0	
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

Πίνακας 6.42. Πίνακας καταστάσεων για D f - f

Βήμα 4: Εύρεση των εξισώσεων

Μπορούμε και πάλι να σχεδιάσουμε τους χάρτες Karnaugh, οπότε έχουμε:

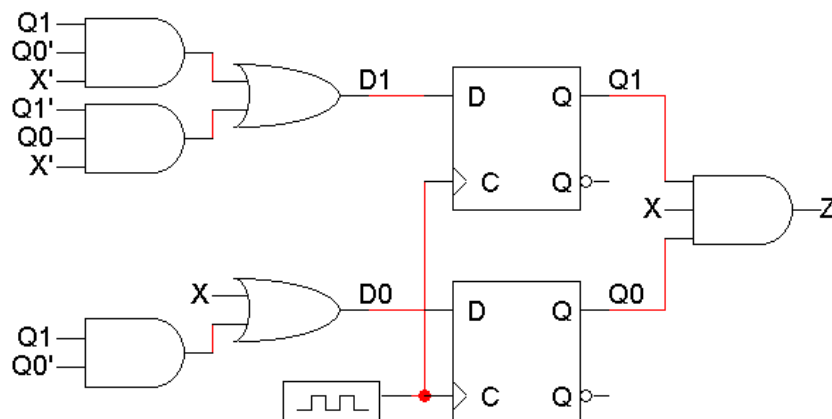
$$D_1 = Q_1 Q_0' X' + Q_1' Q_0 X'$$

$$D_0 = X + Q_1 Q_0'$$

$$\text{και τέλος } Z = Q_1 Q_0 X$$

Βήμα 5: Κατασκευή του κυκλώματος

Στο Σχήμα 6.52 αποδίδεται το κύκλωμα που προκύπτει με τη χρήση των D f - f .



Σχήμα 6.52. Υλοποίηση του κυκλώματος με D f - f και πύλες

Σύγκριση των Flip-flops (JK και D)

Τα JK f - f καθώς υπάρχουν πολλές αδιάφορες καταστάσεις στις εισόδους των f - f , μπορεί να οδηγήσουν σε ένα απλούστερο κύκλωμα.

Τα D f - f s έχουν το πλεονέκτημα ότι δε χρειάζεται να οργανώσουμε καθόλου τις τιμές των εισόδων καθώς $Q(n+1) = D$. Εν τούτοις, οι εξισώσεις των εισόδων του D είναι, συνήθως, πιο πολύπλοκες από τις αντίστοιχες εξισώσεις των εισόδων του JK.

Πρακτικά, τα D f - f s χρησιμοποιούνται συχνότερα, καθώς υπάρχει μόνο μια είσοδος για κάθε f - f και όχι δύο. Επίσης δεν υπάρχουν πίνακες διέγερσης, για να ανησυχούμε.

Τα D f - f s μπορούν να υλοποιηθούν με ελαφρώς λιγότερο υλικό από ότι τα JK f - f s.

6.9. Μετρητές με μηχανές καταστάσεων

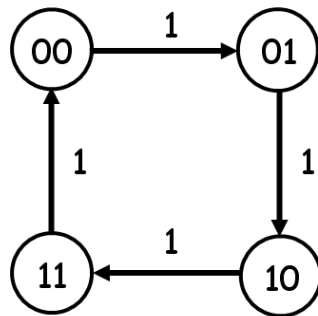
Θα μελετήσουμε διάφορα είδη μετρητών και θα αναλύσουμε τον τρόπο κατασκευής τους. Οι μετρητές δεν είναι μόνο παραδείγματα ακολουθιακής ανάλυσης και σχεδίασης, αλλά είναι συσκευές που χρησιμοποιούνται σε μεγαλύτερα κυκλώματα, όπως θα δούμε στη συνέχεια. (Katz, 2005; Κοσσίδης, & Γιαννακόπουλος 2006; Wakerly, 2006; Godse, & Godse, 2010)

Οι μετρητές είναι ακολουθιακά κυκλώματα ειδικού τύπου. Όπως και στους καταχωρητές, η κατάσταση, ή οι τιμές των f - f από μόνες τους, χρησιμεύει σαν «έξοδος».

Η τιμή της εξόδου αυξάνεται κατά 1 σε κάθε κύκλο ρολογιού. Μετά τη μέγιστη τιμή, η έξοδος «γυρίζει» πίσω στο 0. (πίνακας 6.43). Χρησιμοποιώντας 2 bit, θα παίρναμε μία απεικόνιση σαν αυτήν του Σχήματος 6.53:

Παρούσα Κατάσταση		Επόμενη Κατάσταση	
A	B	A	B
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Πίνακας 6.43. Πίνακας Μεταβάσεων



Σχήμα 6.53. Μεταβολή των καταστάσεων

Οι μετρητές μπορούν να χρησιμοποιηθούν σαν απλά ρολόγια για τη μέτρηση του χρόνου.

Μπορεί να χρειαστεί να καταγράψουμε πόσες φορές συνέβη κάτι. Π.χ. πόσα bit στάλθηκαν ή ελήφθησαν, ή πόσα βήματα πραγματοποιήθηκαν σε κάποιον υπολογισμό;

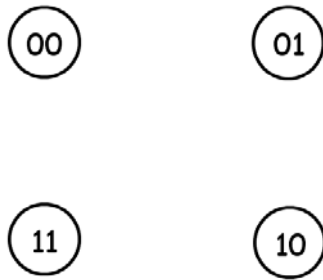
Όλοι οι επεξεργαστές περιέχουν το μετρητή προγράμματος (Program Counter, ή PC). Τα προγράμματα αποτελούνται από μία λίστα εντολών, που πρόκειται να εκτελεστούν η μία μετά την άλλη (τουλάχιστον για το μεγαλύτερο μέρος του προγράμματος).

Ο PC «δείχνει» ποια είναι η τρέχουσα εντολή που εκτελείται, αυξάνει δε την τιμή του σε κάθε κύκλο εντολής, και μετά εκτελείται η επόμενη εντολή του προγράμματος.

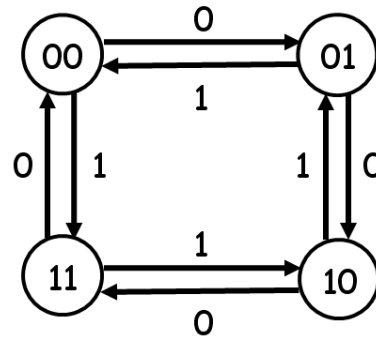
6.9.1. Μετρητής δύο δυαδικών ψηφίων

Ας προσπαθήσουμε να σχεδιάσουμε έναν, λίγο διαφορετικό, μετρητή των 2-bit: Πάλι οι έξοδοι του θα είναι 00, 01, 10 και 11, τις οποίες και θέτουμε σαν καταστάσεις μεταξύ των οποίων θα κινηθούμε ανάλογα με την είσοδο. Τώρα υπάρχει μόνο μια είσοδος, το X. Όταν το X=0, η έξοδος του μετρητή πρέπει να αυξάνεται σε κάθε κύκλο ρολογιού. Αλλά, όταν X=1, η έξοδος θα πρέπει να μειώνεται σε διαδοχικούς κύκλους ρολογιού.

Θα χρειαστούμε και πάλι 2 f - f . Στο Σχήμα 6.54α απεικονίζονται οι τέσσερις πιθανές καταστάσεις:



Σχήμα 6.54α Πιθανές καταστάσεις



Σχήμα 6.54β Διάγραμμα καταστάσεων

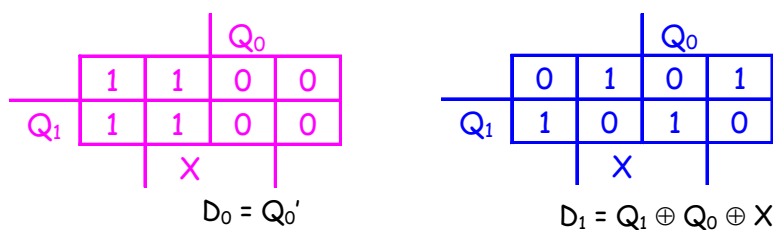
Στο Σχήμα 6.54β απεικονίζεται το ολοκληρωμένο διάγραμμα καταστάσεων, και ο αντίστοιχος πίνακας για το κύκλωμα αυτό δίνεται από τον πίνακα 6.44) (Σιγουρευτείτε ότι γνωρίζετε πώς προκύπτουν τα παρακάτω, καθώς πρόκειται για ένα απλό πρόβλημα ακολουθιακής σχεδίασης!)

Παρούσα Κατάσταση		Είσοδος X	Επόμενη Κατάσταση	
Q ₁	Q ₀		Q ₁	Q ₀
0	0	0	0	1
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

Πίνακας 6.44. Πίνακας του διαγράμματος καταστάσεων

Είσοδοι με D flip-flop

Αν χρησιμοποιήσουμε το D f-f, τότε οι D εισοδοί θα είναι ίδιες με τις επιθυμητές επόμενες καταστάσεις. Οι εξισώσεις για τις εισόδους του D f-f απεικονίζονται στον πίνακα 6.45.



Πίνακας 6.45. Εξισώσεις των εισόδων του D f-f

Είσοδοι με JK f-f

Αν χρησιμοποιήσουμε αντί για το D f-f το JK f-f, τότε πρέπει να υπολογίσουμε τις εισόδους JK για κάθε f-f, χρησιμοποιώντας τον πίνακα καταστάσεων του JK f-f.

Ας κοιτάξουμε την παρούσα και την επιθυμητή επόμενη κατάσταση και ας χρησιμοποιήσουμε το γνωστό πίνακα διεγέρσεων του JK f-f (πίνακας 6.46).

Μπορούμε να βρούμε εξισώσεις για τις εισόδους και των 4 f-f, συναρτήσας της παρούσας κατάστασης και της εισόδου. Έχουμε λοιπόν: $J_1 = K_1 = Q_0' X + Q_0 X'$ και $J_0 = K_0 = 1$

Παρούσα Κατάσταση		Είσοδος	Επόμενη Κατάσταση		Είσοδοι των f - f			
Q ₁	Q ₀	X	Q ₁	Q ₀	J ₁	K ₁	J ₀	K ₀
0	0	0	0	1	0	x	1	x
0	0	1	1	1	1	x	1	x
0	1	0	1	0	1	x	x	1
0	1	1	0	0	0	x	x	1
1	0	0	1	1	x	0	1	x
1	0	1	0	1	x	1	1	x
1	1	0	0	0	x	1	x	1
1	1	1	1	0	x	0	x	1

Πίνακας 6.46. Πίνακας καταστάσεων

Μη χρησιμοποιούμενες καταστάσεις

Τα παραδείγματα που είδαμε μέχρι τώρα είχαν 2^n καταστάσεις και χρησιμοποιούσαν n f - f . Αλλά μερικές φορές μπορεί να έχουμε μερικές μη χρησιμοποιούμενες καταστάσεις.

6.9.2. Μετρητής αυτόματης εκκίνησης

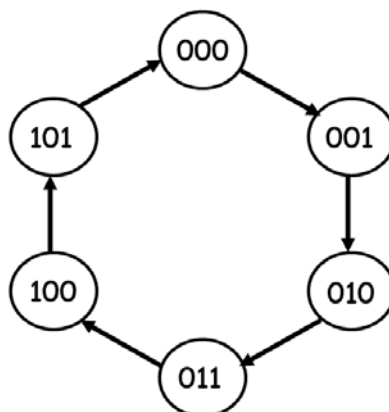
Έστω ότι έχουμε ένα πίνακα κι ένα διάγραμμα καταστάσεων ενός μετρητή, που μετρά συνεχώς από το 0 (000) μέχρι το 5 (101). (Σχήμα 6.55)

Τι πρέπει να βάλουμε στον πίνακα για τις δύο καταστάσεις που δε χρησιμοποιούμε; (πίνακας 6.47)

Για να πάρουμε το απλούστερο δυνατό κύκλωμα, μπορούμε να γεμίσουμε τις επόμενες καταστάσεις με αδιάφορους όρους. Αυτό θα έχει ως αποτέλεσμα να έχουμε αδιάφορους όρους και στις εισόδους των f - f . Με αυτό τον τρόπο μπορούμε να απλοποιήσουμε το κύκλωμά μας.

Παρούσα Κατάσταση			Επόμενη Κατάσταση		
Q2	Q1	Q0	Q2	Q1	Q0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	x	x	x
1	1	1	x	x	x

Πίνακας 6.47. Παρούσα και επομένη κατάσταση



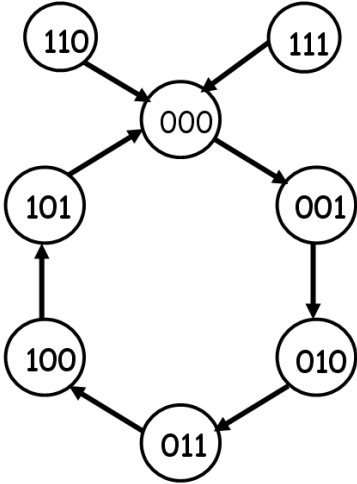
Σχήμα 6.55. Διάγραμμα καταστάσεων μετρητή

Αν, με κάποιο τρόπο το κύκλωμα καταλήξει σε κάποια από τις μη χρησιμοποιούμενες καταστάσεις (110 ή 111), η συμπεριφορά του εξαρτάται από το τι ακριβώς έχει συμπληρωθεί στους αδιάφορους όρους.

Για να πάρουμε το ασφαλέστερο δυνατό κύκλωμα, σαφώς μπορούμε να γεμίσουμε τις επόμενες καταστάσεις των μη χρησιμοποιούμενων όρων (110 και 111), (πίνακας 6.48) με βάση τον οποίο καταλήγουμε και στο Σχήμα 6.56. Αυτή η μέθοδος εγγυάται ότι ακόμα και αν το κύκλωμα, με κάποιο τρόπο, βρεθεί σε μία μη χρησιμοποιούμενη κατάσταση, τελικά θα καταλήξει σε μία έγκυρη κατάσταση. Αυτός ο μετρητής ονομάζεται μετρητής αυτόματης εκκίνησης.

Παρούσα Κατάσταση			Επόμενη Κατάσταση		
Q2	Q1	Q0	Q2	Q1	Q0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0

Πίνακας 6.48. Κωδικοποίηση παρούσας και επομένης κατάστασης



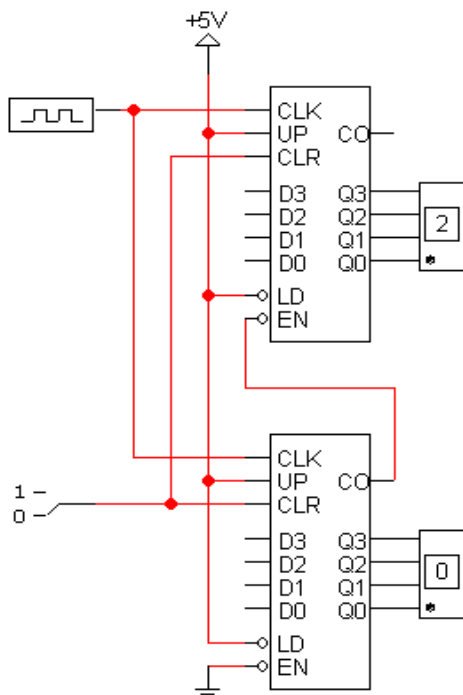
Σχήμα 6.56. Διάγραμμα καταστάσεων μετρητή

6.9.3. Μετρητής οκτώ δυαδικών ψηφίων

Όπως θα περιμένατε, μπορούμε τώρα να χρησιμοποιήσουμε απλούς μετρητές, για να κατασκευάσουμε άλλους πιο πολύπλοκους.

Στο Σχήμα 6.57 παρατίθεται ένας μετρητής 8-bit που υλοποιείται από δύο μετρητές των 4-bit. Η κάτω μονάδα αναπαριστά τα 4 λιγότερο σημαντικά ψηφία, ενώ η επάνω τα 4 περισσότερα σημαντικά ψηφία. Όταν η κάτω μονάδα φτάσει στο 1111 (δηλαδή όταν CO = 0), ενεργοποιείται η επάνω μονάδα για ένα κύκλο.

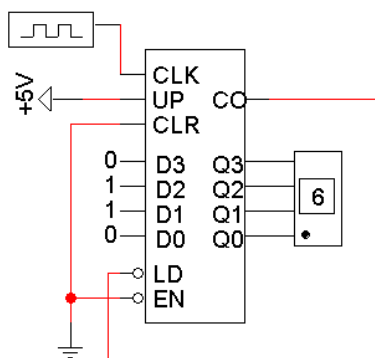
Σημειώσεις για την υλοποίηση:α) Οι μετρητές μοιράζονται το CLK και το CRL και β) Χρησιμοποιούμε δεκαεξαδική παράσταση.



Σχήμα 6.67. Δομικό διάγραμμα μετρητή 8-δωαδικών ψηφίων

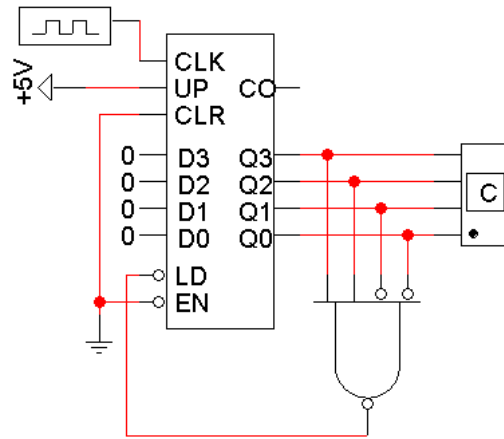
6.9.4. Άνω / κάτω περιορισμένος μετρητής των 4-bit

Μπορούμε να υλοποιήσουμε ένα μετρητή που να ξεκινάει από μία άλλη τιμή εκτός από την 0000. Στο Σχήμα 6.58 απεικονίζεται ο κάτω περιορισμένος μετρητής των 4-bit ο οποίος, όταν το CO=0 το σήμα LD εξαναγκάζει την επόμενη κατάσταση να φορτωθεί από τα D₃-D₀. Το αποτέλεσμα είναι ότι αυτός ο μετρητής μετά την κατάσταση 1111 μεταβαίνει στην 0110 (αντί για 0000). (Με βάση το συγκεκριμένο διάγραμμα).



Σχήμα 6.58. Δομικό διάγραμμα κάτω περιορισμένου μετρητή 4-bit

Μπορούμε, επίσης, να φτιάξουμε ένα κύκλωμα, που απεικονίζεται στο Σχήμα 6.59, άνω περιορισμένος μετρητής των 4-bit, το οποίο κύκλωμα θα μετράει μέχρι το 1100, αντί για το 1111. Όταν η τιμή του μετρητή φθάσει στο 1100, η πύλη NAND εξαναγκάζει τον μετρητή να «φορτώσει», οπότε η επόμενη κατάσταση είναι η 0000.

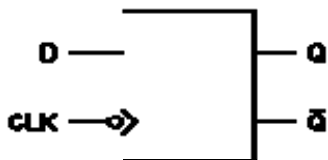


Σχήμα 6.59. Δομικό διάγραμμα άνω περιορισμένου μετρητή 4-bit

6.10. Ασκήσεις – Ερωτήσεις

ΕΡΩΤΗΣΕΙΣ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

- Ένα $J-K$ flip-flop αλλάζει κατάσταση, όταν:
 Α. $J=0, K=0$ Β. $J=1, K=0$ Γ. $J=0, K=1$ Δ. $J=1, K=1$
- Πόσα flip-flops απαιτούνται για μια συσκευή διαίρεσης δια 128;
 Α.1 Β.4 Γ.6 Δ.7
- Το φαινόμενο της διερμηνείας ανεπιθύμητων σημάτων στις εισόδους J και K ενώ το C_p (παλμός ρολογιού) είναι HIGH ονομάζεται:
 Α. Έλεγχος σφάλματος ψηφίου ισοτιμίας Β. Σύλληψη των 1
 Γ. Ψηφιακή διάκριση Δ. Ψηφιακό φίλτρο
- Σε ένα master-slave F-F, πότε το master είναι ενεργό;
 Α. Όταν η πύλη είναι LOW Β. Όταν η πύλη είναι HIGH
 Γ. Ισχύουν και τα δύο(A+B) Δ. Κανένα από τα δύο
- Ένα παράδειγμα της χρήσης του $S-R$ flip-flop είναι:
 Α. Γεννήτρια μετάβασης παλμών Β. Ασταθής ταλαντωτής
 Γ. Δυαδική αποθήκευση καταχωρητή Δ. Ταλαντωτής
- Ποιο από τα παρακάτω ισχύουν για ένα D flip-flop πύλης (gated D- $f-f$)
 Α. Η έξοδος εναλλάσσεται, αν μία από τις εισόδους γίνει HIGH.
 Β. Μόνο μία είσοδος μπορεί να είναι HIGH κάθε φορά.
 Γ. Η έξοδος ακολουθεί την είσοδο, όταν είναι ενεργό.
 Δ. Η Q έξοδος ακολουθεί την είσοδο D , όταν το enable είναι HIGH.
- Πώς μπορεί ένα F-F με διασταυρούμενη σύζευξη (cross coupled) των πυλών NAND να επιτευχθεί να έχει ενεργές –HIGH $S-R$ εισόδους;
 Α. Δε γίνεται Β. Με αντιστροφή των εξόδων Q
 Γ. Με αντιστροφή των $S-R$ εισόδων
- Πότε ένα flip-flop λέγεται ότι είναι «διαφανές»;
 Α. Όταν η έξοδος Q είναι αντίθετη της εισόδου
 Β. Όταν η έξοδος Q ακολουθεί την είσοδο
 Γ. Όταν μπορείτε να δείτε μέσα από το ολοκληρωμένο κύκλωμα
- Ένα $J-K$ flip-flop είναι σε «σταθερή» κατάσταση, όταν _____.
 Α. $J=1, K=1$ Β. $J=1, K=0$
 Γ. $J=0, K=1$ Δ. $J=0, K=0$
- Τα σύμβολα στο flip-flop του σχήματος δείχνουν διέγερση



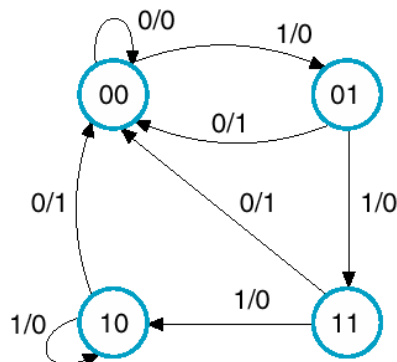
- Στην αρνητική ακμή του παλμού CLK
- Στη θετική ακμή του παλμού CLK
- Στο HIGH της CLK κυματομορφής
- Στο LOW της CLK κυματομορφής

- Ένα $J-K$ flip-flop με $J=1$ και $K=1$ έχει είσοδο με ρολόι 20 KHz. Η έξοδος Q είναι:
 Α. Συνεχώς LOW Β. Συνεχώς HIGH
 Γ. Τετραγωνικό κύμα 20 KHz Δ. Τετραγωνικό κύμα 10 KHz
- Στη VHDL, πόσες εισόδους έχει ένα JK flip-flop;
 Α. 2 Β. 3 Γ. 4 Δ. 5
- Ποια είναι η σημασία των ακροδεκτών J και K για το JK flip-flop;
 Α. Δεν υπάρχει γνωστή σημασία στην ονομασία τους.

- B. Η J αντιπροσωπεύει «άλμα», το οποίο είναι το πώς η έξοδος Q αντιδρά κάθε φορά που το ρολόι γίνεται HIGH και η είσοδος J είναι επίσης HIGH.
- Γ. Τα γράμματα έχουν επιλεγεί προς τιμήν του Jack Kilby, εφευρέτη του ολοκληρωμένου κυκλώματος.
- Δ. Όλα τα άλλα γράμματα του αλφαβήτου είναι ήδη σε χρήση.
14. Ποιο είναι το μειονέκτημα ενός S-R flip-flop
 Α. Δεν έχει είσοδο
 Β. Έχει μία απροσδιόριστη κατάσταση
 Γ. Δεν έχει είσοδο χρονισμού
 Δ. Έχει μόνο μία έξοδο.
15. Οι ασύγχρονες είσοδοι προκαλούν το flip-flop να αποκρίνεται άμεσα σε σχέση με το ρολόι εισόδου.
 Α. Σωστό
 Β. Λάθος
16. Δύο J-K flip-flops με εισόδους J-K HIGH είναι συνδεδεμένα, για να χρησιμοποιηθούν ως μετρητές. Μετά από τέσσερις παλμούς ρολογιού εισόδου, ο δυαδικός αριθμός είναι _____.
 Α. 00
 Β. 11
 Γ. 01
 Δ. 10
17. Στον πέμπτο παλμό ρολογιού, η 4-bit ακολουθία Johnson είναι $Q_0=0, Q_1=1, Q_2=1, Q_3=1$. Στον έκτο παλμό ρολογιού η ακολουθία θα είναι:
 Α. $Q_0=1, Q_1=0, Q_2=0, Q_3=0$
 Β. $Q_0=1, Q_1=1, Q_2=1, Q_3=0$
 Γ. $Q_0=0, Q_1=0, Q_2=1, Q_3=1$
 Δ. $Q_0=0, Q_1=0, Q_2=0, Q_3=1$
18. Η ακολουθία δυαδικών ψηφίων 0010 έχει εγγραφεί σειριακά (δεξιά-bit πρώτο) σε έναν 4-bit παράλληλης εξόδου καταχωρητή μετατόπισης που είναι αρχικά άδειος. Ποιές είναι οι εξοδοί Q μετά από δύο παλμούς ρολογιού;
 Α. 0000
 Β. 0010
 Γ. 1000
 Δ. 1111
19. Για να λειτουργήσει σωστά ένας απαριθμητής κυκλικής ολίσθησης αρχικά απαιτείται
 Α. Εκκαθάριση όλων των F-F
 Β. Προκαθορισμό ενός F-F και εκκαθάριση όλων των άλλων
 Γ. Εκκαθάριση ενός F-F και προκαθορισμός όλων των άλλων
 Δ. Προκαθορισμό όλων των F-F
20. Τι σημαίνει ο όρος παράλληλη φόρτωση ενός καταχωρητή ολίσθησης;
 Α. Όλα τα F-Fs είναι προ-τοποθετημένα με δεδομένα.
 Β. Κάθε F-F φορτώνεται με τα δεδομένα, ένα κάθε φορά
21. Σε μία 6-bit ακολουθία μετρητή Johnson, πόσες συνολικά καταστάσεις ή bit έχουμε;
 Α. 2
 Β. 6
 Γ. 12
 Δ. 24
22. Ένας modulo-12 κυκλικός απαριθμητής απαιτεί το λιγότερο
 Α. 10 F-F
 Β. 12 F-F
 Γ. 6 F-F
 Δ. 2 F-F
23. Ποιά η διαφορά μεταξύ καταχωρητή δεξιάς και καταχωρητή αριστερής ολίσθησης;
 Α. Δεν υπάρχει διαφορά
 Β. Η κατεύθυνση της ολίσθησης
24. Ποιά είναι η προ-τοποθετημένη κατάσταση σε έναν κυκλικό απαριθμητή ολίσθησης
 Α. Όλα τα F-Fs έχουν τιμή 1
 Β. Όλα τα F-Fs έχουν τιμή 0
 Γ. Ένα έχει τιμή 0, τα υπόλοιπα 1
 Δ. Ένα έχει τιμή 1, τα υπόλοιπα 0
25. Ποιο ΔΕΝ είναι χαρακτηριστικό των καταχωρητών ολίσθησης;
 Α. Serial in/parallel in
 Β. Serial in/parallel out
 Γ. Parallel in/serial out
 Δ. Parallel in/parallel out
26. Σε μια ακολουθία 4-bit απαριθμητή Johnson πόσες καταστάσεις ή bit έχουμε συνολικά;
 Α. 1
 Β. 2
 Γ. 4
 Δ. 8
27. Σε έναν 10-bit κυκλικό απαριθμητή με αρχική κατάσταση 1101000000, ποιά θα είναι η κατάσταση του μετά από το δεύτερο παλμό ρολογιού;
 Α. 1101000000
 Β. 0011010000
 Γ. 1100000000
 Δ. 0000000000

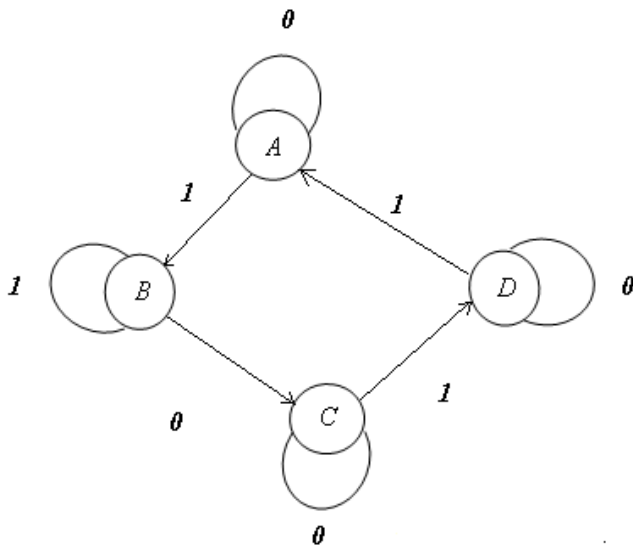
ΑΣΚΗΣΕΙΣ

1. Να μελετηθεί και να σχεδιαστεί κύκλωμα, αποτελούμενο από έναν απαριθμητή 4 ψηφίων και έναν καταχωρητή έτσι ώστε, όταν το περιεχόμενο του απαριθμητή φθάνει το 12, να σταματά την απαρίθμηση και να μεταφέρεται η τιμή του στον καταχωρητή.
2. Να σχεδιάσετε απαριθμητές ως προς μέτρο 5, 7, 9, 12, 15, 25. Για τη σχεδίαση να χρησιμοποιήσετε J-K *flip-flop*.
3. Να σχεδιάσετε ένα λογικό κύκλωμα αποτελούμενο από πύλες NAND τριών εισόδων, όπως επίσης και πύλες NOT, έτσι ώστε να επαληθεύεται η λογική εξίσωση: $T = \overline{A}BC + ABC + \overline{A}\overline{B}\overline{C}$
4. Να δοθεί ο πίνακας αλήθειας, η λογική συνάρτηση και το λογικό κύκλωμα διάταξης, η οποία δέχεται ως είσοδο δύο διψήφιους δυαδικούς αριθμούς, τους A_1A_0 , B_1B_0 και δίνει έξοδο 1, μόνο όταν οι δύο δυαδικοί αριθμοί είναι ίσοι.
5. Να σχεδιασθεί σύγχρονος απαριθμητής τεσσάρων δυαδικών ψηφίων, ο οποίος θα ξεκινά τη μέτρηση από το 0, θα καταλήγει στο 14 και αμέσως μετά θα μεταπηδάει πάλι στο 0. Για τη σχεδίαση να χρησιμοποιηθούν J-K *flip-flop*.
6. Να σχεδιασθεί σύγχρονος απαριθμητής τεσσάρων δυαδικών ψηφίων, ο οποίος θα εργάζεται στο Κώδικα Gray. Για τη σχεδίαση να χρησιμοποιηθούν J-K *flip-flop*.
7. Να σχεδιασθεί σύγχρονος απαριθμητής τριών δυαδικών ψηφίων, ο οποίος θα μετρά σύμφωνα με την παρακάτω ακολουθία: 0, 2, 4, 6, 1, 3, 5, 7, 0,... Για τη σχεδίαση να χρησιμοποιηθούν J-K *flip-flop*.
8. Κατασκευαστής πρόκειται να τοποθετήσει κλειδαριές με ηλεκτρονικό συνδυασμό σε μια σειρά χρηματοκιβωτίων. Απαιτούνται όμως οι παρακάτω προδιαγραφές:
 - α) Το χρηματοκιβώτιο θα έχει τέσσερις διακόπτες πίεσης χαρακτηριζόμενους με τα γράμματα A, B, C, D.
 - β) Χειρισμός αυτών των διακοπών κατά τη σωστή διαδοχή ανοίγει τη πόρτα του χρηματοκιβωτίου. Μια λανθασμένη διαδοχή ή η αποτυχία να εκτελεστεί η διαδικασία εντός 10 δευτερολέπτων από την πίεση του πρώτου διακόπτη, έχει ως αποτέλεσμα να τεθεί σε λειτουργία το κουδούνι κινδύνου.
 - γ) Η διαδοχή πίεσης των διακοπών πρέπει να είναι αποκλειστική για κάθε χρηματοκιβώτιο και να αποτελείται από την πίεση πέντε φορές διαφόρων διακοπών (κάθε διακόπτης ελευθερώνεται μετά από κάθε πίεση). Δεν είναι αναγκαίο να χρησιμοποιηθούν και οι τέσσερις διακόπτες, π.χ. ο συνδυασμός B, C, C, B, A, είναι δεκτός.
 - δ) Σε περίπτωση διακοπής της τάσης τροφοδοσίας, το χρηματοκιβώτιο πρέπει να ανοίγει με κατάλληλο κλειδί. Κάθε προσπάθεια να χρησιμοποιηθεί το κλειδί, εφόσον δεν έχουμε διακοπή τάσης τροφοδοσίας, θέτει σε λειτουργία το σήμα κινδύνου.
 Να σχεδιάσετε λογικό σύστημα που να ικανοποιεί τις παραπάνω προδιαγραφές.
9. Να βρείτε το ελάχιστο άθροισμα γινομένων (EAG-SOP) για τη συνάρτηση: $f(w,x,y,z) = \Sigma m(0,2,4,5,8,14,15)$ με αδιάφορους όρους τους $d(w,x,y,z) = \Sigma m(7,10,13)$
Σημείωση: Οι συνδυασμοί των εισόδων $wxyz = 0111, 1010$ and 1101 (που αναφέρονται στους EΛ-ΒΟ m_7, m_{10} and m_{13}) δεν χρησιμοποιούνται.
1. Δίνεται το παρακάτω διάγραμμα Mealy. Να εξαγάγετε τον πίνακα καταστάσεων και την έξοδό του, και να το υλοποιήσετε με πύλες και D-*flip-flop*.



Να σχεδιάσετε σύγχρονο ακολουθιακό κύκλωμα το οποίο θα δίνει έξοδο όταν θα αναγνωρίζει μέσα σε μια δυαδική ακολουθία τη μορφή 101. Για την πραγματοποίηση του κυκλώματος να χρησιμοποιήσετε JK-flip-flop. Να σχεδιάσετε και το αντίστοιχο διάγραμμα καταστάσεων.

2. Να σχεδιάσετε κύκλωμα που να υλοποιεί το παρακάτω διάγραμμα.



3. Να σχεδιάσετε ένα ασύγχρονο κύκλωμα το οποίο δέχεται στην είσοδό του δύο διαφορετικούς παλμούς έστω M και N διαφορετικών μεταξύ τους. Στο κύκλωμα αυτό, αν ο N έχει τιμή 1 πριν από τον M, τότε η έξοδος θα είναι η πράξη AND των δύο παλμών. Εντούτοις, αν το M έχει τιμή 1 πριν από το N, τότε η έξοδος πρέπει να είναι 0. Για τη διευκολυνσή σας, δίνονται τα διαγράμματα των παλμών εισόδου και το διάγραμμα καταστάσεων.

Αναφορές-Βιβλιογραφία

- Δεληγιάννης†, Θ. (2004). *Ηλεκτρονικά Αναλογικά και Ψηφιακά*, Ίδρυμα Ευγενίδου
- Κοσσίδης, Α.Θ. (1996). *Σχεδίαση Ψηφιακών Κυκλωμάτων*, Εκδόσεις Μπένο
- Κοσσίδης, Α.Θ., Γιαννακόπουλος, Π., (2006), *Αριθμητικά Συστήματα και Ψηφιακά Κυκλώματα*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα
- Φραγκάκης, Γ. (1975). *Λογικά Κυκλώματα*, Αθήνα
- Balabanian, N., Carlson, B. (2007). *Digital Logic Design Principles*, John Wiley
- Balch, M. (2003). *Complete Digital Design*, Mc Graw Hill
- Floyd, Thomas L., (2006). *Digital Fundamentals*, 9th, Pearson International Edition
- Givone, D. (2002). *Digital Principles and Design*,. Mc Graw Hill
- Godse, A.P., Godse, D.A. (2010). *Digital Logic Design and Application*, Technical Publications Pune
- Holdsworth, Brian, Woods, Clive (2002). *Digital Logic Design*, 4th Edition, Newnes
- Katz, R. (2005). *Contemporary Logic Design*, 2/e, Prentice Hall
- Mano, M., Ciletti, M. (2014). *Ψηφιακή Σχεδίαση*, 5^η έκδοση, Παπασωτηρίου
- Maxfield Clive (2009). *Bebop to the Boolean Boogie, An Unconventional Guide to Electronics*, 3rd, Newnes, Elsevier <http://english.360elib.com/datu/T/EM140371.pdf>
- Nashelsky, Louis (1994). *Introduction to Digital Technology*, 4th Ed., Prentice Hall
- Nelson, V., Nagle, H., Carroll, B., Irwin, J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall
- Predko, Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill
- Pritchard, N. (2015). *Fundamentals of Digital Electronics*, CreateSpace Independent Publishing Platform
- Roth, Charles Jr., Kinney, L.L., (2014). *Fundamentals of Logic Design*, 7th, Cengage Learning
- Tocci, R. J., Widmer, N. S, and Moss, Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley
- Wakerly, J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Κεφάλαιο 7

Σύνοψη

Στο κεφάλαιο αυτό θα εξετάσουμε ορισμένα λογικά κυκλώματα προσαρμογής, τα οποία χρησιμοποιούνται συχνά στους ηλεκτρονικούς υπολογιστές. Αρχικά θα αναφερθούμε στον τρόπο μετατροπής ψηφιακών σημάτων σε αναλογικά (Digital to Analog, D/A) και αναλογικών σε ψηφιακά (Analog to Digital, A/D), και θα αναλυθούν χρήσιμα κυκλώματα. Στη συνέχεια, θα αναφερθούμε στα υπόλοιπα κυκλώματα προσαρμογής, δίνοντας εκτενή περιγραφή των αποκωδικοποιητών, των κυκλωμάτων πολυπλεξίας, αναφέροντας και τις εφαρμογές τους, και το κεφάλαιο ολοκληρώνεται με τον απομονωτή και το modem.

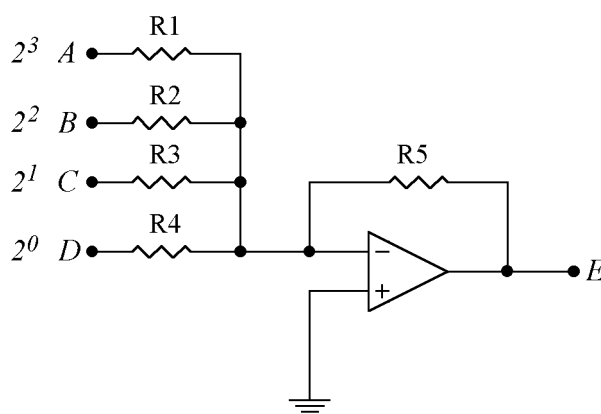
Προαπαιτούμενη γνώση

Η γνώση των βασικών ηλεκτρονικών που διδάσκεται στο μάθημα της Ηλεκτρονικής και βασικές αρχές διαμόρφωσης.

7. Λογικά Κυκλώματα Προσαρμογής

7.1. Μετατροπείς ψηφιακών σημάτων σε αναλογικά

Οι διατάξεις που επιτρέπουν τη μετατροπή ενός δυαδικού αριθμού, που αντιπροσωπεύει μια ψηφιακή ποσότητα (διακριτό μέγεθος), σε μια αναλογική (συνεχές μέγεθος) είναι γνωστές σαν «Μετατροπείς ψηφιακών σημάτων σε αναλογικά» (Digital to Analog Converters, D/A). (Taub, & Schilling, 1983; Κοσσιδάς, 1996; Holdsworth, & Woods, 2002; Δεληγιάννης, 2004; Jain, 2010; Sedra, Smith, & Kenneth, 2015)



Σχήμα 7.1. Απλό κύκλωμα μετατροπής ψηφιακού σήματος σε αναλογικό

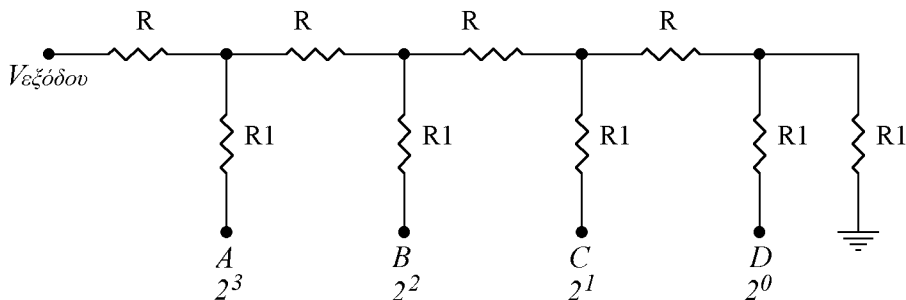
Η μετατροπή των ψηφιακών σημάτων μπορεί να πραγματοποιηθεί με πολλά διαφορετικά κυκλώματα, τα οποία μπορεί να είναι αμιγώς ηλεκτρονικά ή και ηλεκτρομηχανικά. Στο σχήμα 7.1 απεικονίζεται ένα κύκλωμα μετατροπής D/A.

Με δεδομένο ότι $R_1=2R_5$, $R_2=2R_1$, $R_3=4R_1$ και $R_4=8R_1$ και ότι η απόλυτη τιμή του κέρδους της αθροιστικής βαθμίδας που ακολουθεί είναι $A_V = R_5 / R_i$ (όπου $R_i = R_1, R_2, R_3, R_4$), είναι εύκολο να αντιληφθούμε ότι, επιβάλλοντας στάθμη ίση με +5 V (λογικό «1») σε κάθε μια από τις εισόδους A, B, C και D, η τάση εξόδου (σημείο E) θα έχει τιμή που θα εξαρτάται αποκλειστικά από τις σχέσεις των αντιστάσεων μεταξύ τους. Αν, για παράδειγμα, έχει επιβληθεί στις εισόδους ABCD η ακολουθία 1000, τότε η έξοδος V θα έχει τιμή ίση με το 1/2 της τάσης που επιβάλλεται στην είσοδο A. Αν στα ίδια σημεία εισόδου επιβληθεί ο συνδυασμός 1001, τότε η έξοδος E θα αναδείξει τάση ίση με 3,125 V (δηλαδή $1/2 * 5V + 1/8 * 5V$) κ.ο.κ.

Ένα ακόμα κύκλωμα που παρουσιάζει καλύτερη ακρίβεια απεικονίζεται στο Σχήμα 7.2. Εδώ όλες οι αντιστάσεις που έχουν τοποθετηθεί οριζόντια έχουν τιμή ίση με R, ενώ, όσες έχουν τοποθετηθεί καθέτως, έχουν τιμή $R_1=2R$.

Επιβάλλοντας στις εισόδους ABCD οποιαδήποτε δυαδική ακολουθία, στην έξοδο θα έχουμε τάσεις που θα είναι ανάλογες της τιμής που αντιπροσωπεύει η εκάστοτε ακολουθία. Η απόδειξη της συμπεριφοράς

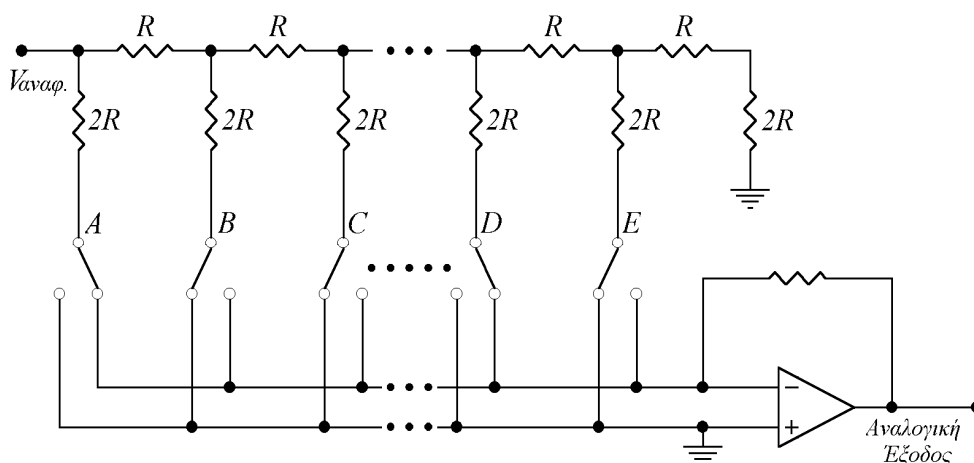
του κυκλώματος αφήνεται στον αναγνώστη, ο οποίος, κάνοντας χρήση των θεωρημάτων της ανάλυσης κυκλωμάτων (Thevenin κ.λπ.), μπορεί να αποδείξει την ορθή λειτουργία του.



Σχήμα 7.2. Μετατροπέας ψηφιακού σήματος σε αναλογικό με δικτύωμα R-2R

Αν και αυτά τα κυκλώματα εργάζονται πολύ καλά, οι σημερινές ανάγκες έχουν επιβάλει τη χρήση ακόμα πιο πολύπλοκων, που κατασκευάζονται πλέον στο εσωτερικό ολοκληρωμένων κυκλωμάτων. Στο Σχήμα 7.3, που ακολουθεί, βλέπουμε το εσωτερικό ενός μονολιθικού μετατροπέα D/A. Κύριο ρόλο και σ' αυτό το κύκλωμα μετατροπής παίζει πάλι το δικτύωμα του R-2R που αναφέραμε στο προηγούμενο σχήμα. Αυτή τη φορά όμως συνοδεύεται από μια πηγή τάσης αναφοράς και πέντε διακόπτες δύο θέσεων, που η θέση τους ελέγχεται από τη δυαδική πληροφορία που επιβάλλεται σ' αυτούς. Θεωρώντας πως οι εισόδοι του τελεστικού ενισχυτή που ακολουθεί το δικτύωμα βρίσκονται στο δυναμικό της γης και εφαρμόζοντας για μια ακόμα φορά τα θεωρήματα της ανάλυσης κυκλωμάτων, αποδεικνύουμε εύκολα πως στον κλάδο A ρέει πάντοτε ρεύμα ίσο με το $1/2 I$, στον κλάδο B ίσο με $1/4 I$ κ.ο.κ. Αν η τιμή του συνδυασμού ABCDE είναι ίση με 00000, τότε όλοι οι διακόπτες οδηγούν τα ρεύματα των κλάδων στη μη αναστρέφουσα είσοδο του τελεστικού ενισχυτή, η οποία είναι συνδεδεμένη ακλόνητα με τη γη. Αν όμως ο συνδυασμός ABCDE είναι ίσος με 11111, τότε οι διακόπτες οδηγούν το ρεύμα προς την αναστρέφουσα είσοδο του ενισχυτή, η οποία λόγω της (θεωρητικά) άπειρης αντίστασης εισόδου το αφήνει να κινηθεί μέσα από την αντίσταση ανασύζευξης. Λόγω της συγκεκριμένης συνδεσμολογίας του ενισχυτή, που είναι γνωστή σαν «μετατροπέας ρεύματος σε τάση», το τελικό αποτέλεσμα θα είναι η εμφάνιση στην έξοδο μιας τάσης που θα είναι ανάλογη του συνδυασμού που επιβλήθηκε στις εισόδους του κυκλώματος.

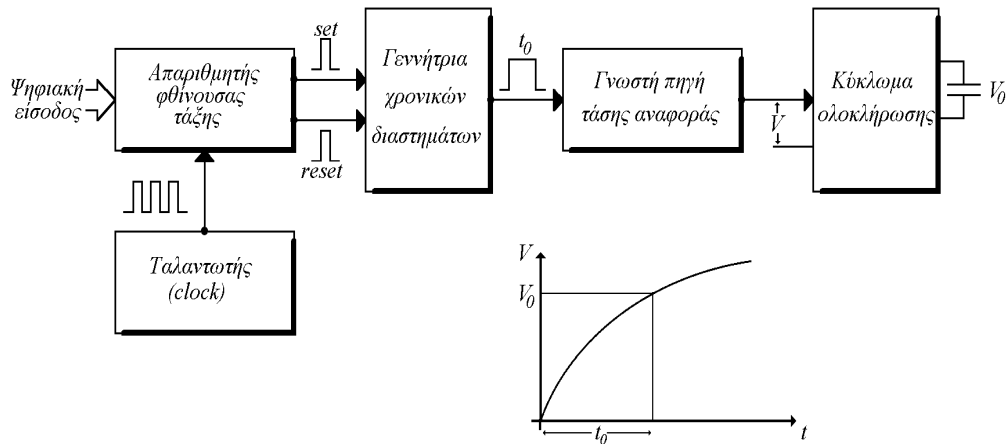
Από τις τρεις αυτές διαφορετικές προσεγγίσεις η τελευταία συνδυάζει και το χαμηλότερο κόστος (όλα τα εξαρτήματα μέσα σ' ένα IC) και την υψηλότερη αξιοπιστία (αδιαφορία έναντι των διακυμάνσεων των τάσεων που αντιστοιχούν στο λογικό «1» και στο λογικό «0»).



Σχήμα 7.3. Εσωτερική δομή ενός μετατροπέα D/A

Στη συνέχεια θα εξετάσουμε ένα άλλο είδος βραδύτερου μετατροπέα, ο οποίος χρησιμοποιείται περισσότερο για την ενεργοποίηση ηλεκτρονόμων, διακοπών ή άλλων ηλεκτρομηχανικών διατάξεων. Το Σχή-

μα 7.4 μας βοηθάει να κατανοήσουμε τη λειτουργία του. Η μετατροπή γίνεται σε δύο φάσεις: στην πρώτη, δημιουργείται ένα χρονικό διάστημα του οποίου η τιμή είναι ευθέως ανάλογη της ψηφιακής τιμής που επιβλήθηκε στην είσοδό του. Στη δεύτερη ένας πυκνωτής φορτίζεται κατά τη διάρκεια αυτού του χρονικού διαστήματος έτσι ώστε η τιμή της τάσης φόρτισής του να είναι ανάλογη της αριθμητικής τιμής εισόδου.



Σχήμα 7.4. Μετατροπέας D/A με φόρτιση πυκνωτή

Αναλυτικότερα, η εργασία έχει ως εξής: οι ψηφιακές τιμές, που αντιπροσωπεύουν την αριθμητική έξοδο του ψηφιακού υπολογιστή, εφαρμόζονται στις εισόδους προτοποθέτησης ενός προγραμματιζόμενου απαριθμητή φθίνουσας τάξης. Ο απαριθμητής με τη σειρά του επενεργεί στην έξοδο «Set» της γεννήτριας παραγωγής χρονικού διαστήματος που, στην πραγματικότητα, είναι ένας πολυδονητής δύο καταστάσεων, αναγκάζοντας την έξοδό της να οδηγηθεί σε λογικό «1». Στη συνέχεια εφαρμόζεται στον απαριθμητή, μέσω ενός ταλαντωτή, μια ακολουθία ωρολογιακών παλμών γνωστής και σταθερής συχνότητας. Οι ωρολογιακοί παλμοί, εφαρμοζόμενοι στον απαριθμητή, ελαττώνουν το περιεχόμενό του μέχρι τον τελικό μηδενισμό του. Όταν συμβεί αυτό, ένας παλμός «Reset» στέλνεται από τον απαριθμητή στον πολυδονητή δύο καταστάσεων, που με τη σειρά του οδηγεί την έξοδό του σε λογικό «0».

Με αυτόν τον τρόπο, έχει δημιουργηθεί στην έξοδο του πολυδονητή ένας τετραγωνικός παλμός του οποίου η διάρκεια του t_0 είναι ανάλογη της ψηφιακής πληροφορίας που επιβλήθηκε στην είσοδο του απαριθμητή. Ο τετραγωνικός παλμός εφαρμόζεται στη συνέχεια σε μια πύλη η οποία θέτει σε λειτουργία ένα κύκλωμα στο οποίο υπάρχει πηγή συνεχούς τάσης V . Συνέπεια αυτής της ενέργειας είναι η φόρτιση του πυκνωτή C , μέσω μιας αντίστασης R , συναρτήσει του χρόνου.

Στο τέλος του τετραγωνικού παλμού διάρκειας t_0 , ο πυκνωτής θα είναι φορτισμένος σε μια τιμή V_0 που είναι ανάλογη της διάρκειας του τετραγωνικού παλμού, δηλαδή ανάλογη της αριθμητικής τιμής της ψηφιακής εισόδου.

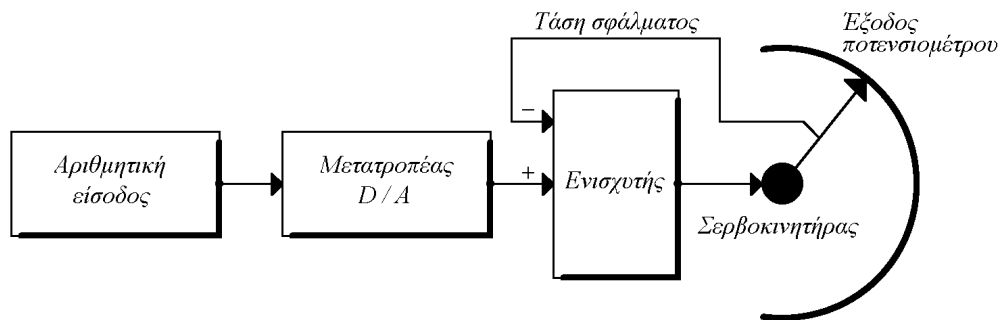
7.1.1 Χρήση των μετατροπέων D/A για την περιστροφή αξόνων

Μια εφαρμογή, στην οποία φαίνεται καθαρά η χρήση ενός μετατροπέα ψηφιακού σήματος σε αναλογικό, απεικονίζεται στο Σχήμα 7.5. Σ' αυτή την εφαρμογή επιδιώκουμε η αριθμητική πληροφορία, που προέρχεται από την έξοδο του ψηφιακού υπολογιστή, να παριστάνει τη γωνιακή θέση ενός άξονα.

Ως εκ τούτου, θα πρέπει ειδικοί σερβομηχανισμοί να στρέφουν τον ελεγχόμενο άξονα κατά μια γωνία, της οποίας η τιμή θα περιγράφεται από τη δυαδική τιμή που παράγει ο υπολογιστής.

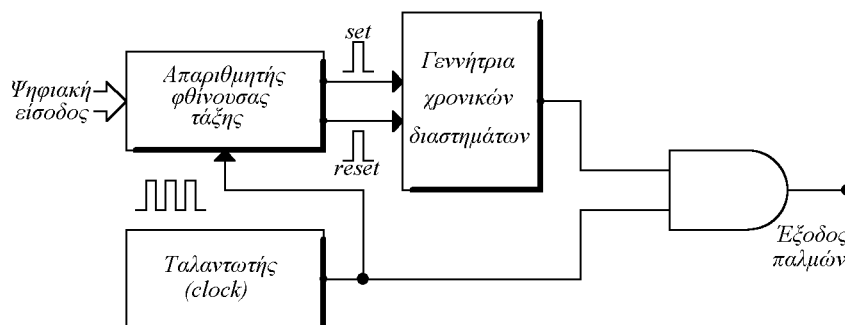
Αρχικά η ψηφιακή τιμή της εισόδου μετατρέπεται, αρχικά, σε αναλογική τάση με τη βοήθεια του αντίστοιχου μετατροπέα. Η προκύπτουσα τάση, που είναι ανάλογη με την αριθμητική τιμή εισόδου, εφαρμόζεται στην είσοδο του διαφορικού ενισχυτή που αποτελεί μέρος του σερβομηχανισμού. Η έξοδος σφάλματος του ενισχυτή οδηγεί ένα σερβοκινητήρα, του οποίου ο περιστρεφόμενος άξονας στρέφει, εκτός από τον άξονα που επιζητούμε να στρέψουμε, και το δρομέα ενός ποτενσιόμετρου. Η τάση που αναπτύσσεται στον τελευταίο (τάση ανάδρασης) επιβάλλεται στην αναστρέφουσα είσοδο του ενισχυτή. Όταν η τάση αυτή γίνει ίση με την τάση εξόδου, τότε η έξοδος του ενισχυτή μηδενίζεται, με αποτέλεσμα την ακινητοποίηση του άξονα, που μένει στραμμένος σε μια θέση ανάλογη της τάσης εισόδου. Άξιο παρατήρησης είναι ότι ο βρόχος ανάδρασης

περιλαμβάνει μόνο τα κυκλώματα τοποθέτησης του σερβοκινητήρα στην επιθυμητή θέση και δεν επιδρά στον μετατροπέα D/A, η έξοδος του οποίου χρησιμεύει για την αρχική χονδρική τοποθέτηση του άξονα.



Σχήμα 7.5. Σχηματικό διάγραμμα μετατροπής ψηφιακής εισόδου σε αναλογική γωνιακή θέση

Ένας άλλος τρόπος, αρκετά διαδεδομένος στην περίπτωση γωνιακών περιστροφών, είναι και ο εξής: Η δυαδική τιμή που εμφανίζεται στις γραμμές εξόδου του υπολογιστή παράγει, με τη βοήθεια ενός απαριθμητή και μιας γεννήτριας ωρολογιακών παλμών, μια ακολουθία παλμών των οποίων το πλήθος είναι ίσο με αυτό που αντιπροσωπεύει η δυαδική τιμή. Στη συνέχεια οι παλμοί εφαρμοζόμενοι σ' ένα βηματικό κινητήρα, τον περιστρέφουν κατά τόσα βήματα όσος είναι και ο αριθμός των παλμών. Ο αριθμός των εφαρμοζόμενων παλμών είναι, κατά βάση, ακέραιος έτσι, ώστε το όλο σύστημα να μην παρουσιάζει πολυπλοκότητα κατασκευής παρόλο που αυτό επιδρά στην ακρίβεια περιστροφής του άξονα. Ένα διάγραμμα βαθμίδων, που περιγράφει την παραπάνω διαδικασία, απεικονίζεται στο Σχήμα 7.6.



Σχήμα 7.6. Οδήγηση κινητήρων με παλμούς

7.2. Μετατροπείς αναλογικών σημάτων σε ψηφιακά σήματα

Στο φυσικό κόσμο όλα σχεδόν τα μεγέθη είναι αναλογικά. Από την άλλη πλευρά οι ηλεκτρονικοί υπολογιστές, που αντιπροσωπεύουν ό,τι πιο εξελιγμένο στον τομέα της επεξεργασίας, είναι ψηφιακοί. Από αυτήν την απλή διαπίστωση προκύπτει εύκολα η ανάγκη της χρήσης ειδικών κυκλωμάτων, που είναι γνωστά με το όνομα «Μετατροπείς αναλογικών σημάτων σε ψηφιακά» (Analog to Digital Converters, A/D). Εννοείται, φυσικά, ότι προτού το επιθυμητό μέγεθος (χρονικό διάστημα, συχνότητα, τάση, ρεύμα, ταχύτητα, επιτάχυνση, πίεση, θερμότητα κ.λπ.) ψηφιοποιηθεί, θα πρέπει να έχει μετατραπεί πρώτα σε ένα ισοδύναμο ηλεκτρικό, ικανό να διεγείρει τα κυκλώματα του μετατροπέα. (Taub, & Schilling, 1983; Nashelsky, 1994; Κοσσιδάς, 1996; Holdsworth, & Woods, 2002; Δεληγιάννης, 2004; Jain, 2010; Tocci, et al., 2010)

Η μετατροπή αυτών των συνεχών συναρτήσεων του χρόνου, σε διακριτές αριθμητικές τιμές, γίνεται λαμβάνοντας δειγματοληπτικά τιμές σε τακτά διαστήματα και αποδίδοντας την αριθμητική τους τιμή σε κατάλληλες χρονικές στιγμές. Μετά από αυτή τη διαδικασία, η συνάρτηση θα παριστάνεται σαν ένα σύνολο αριθμών που ελήφθησαν κατά τη διάρκεια της δειγματοληψίας. Τίποτα όμως δε θα γνωρίζουμε γι' αυτήν κατά τα ενδιάμεσα χρονικά διαστήματα στα οποία δε γίνεται δειγματοληψία. Αυτό γενικά δε δημιουργεί προβλήματα όταν η συνάρτηση μεταβάλλεται ομαλά με το χρόνο, όπως π.χ. η ημιτονοειδής συνάρτηση, αλλά μπορεί να προξενήσει σοβαρά σφάλματα σε συναρτήσεις που παρουσιάζουν απότομες μεταβολές με το χρόνο, όπως π.χ. οι τριγωνικές ή τετραγωνικές.

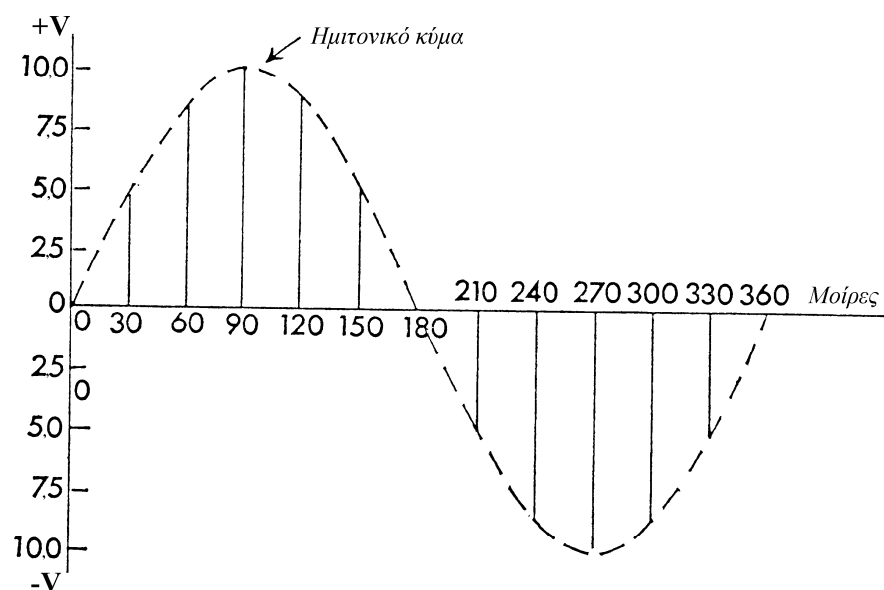
Γι' αυτόν το λόγο ο αριθμός των δειγματοληψιών θα πρέπει να είναι επαρκής, ώστε να έχουμε ικανοποιητικό προσδιορισμό της συνάρτησης. Στο Σχήμα 7.7, που ακολουθεί, δίνεται μια ημιτονοειδής κυματομορφή που παριστάνει μια μεταβολή τάσης 10 V με συχνότητα 50 Hz.

Η δειγματοληψία γίνεται κάθε 30° (θεωρείται γνωστό το θεώρημα του Shannon), δηλαδή σε χρόνο:

$$T = \frac{30^\circ}{360^\circ} \frac{1}{50} = \frac{1}{600}$$

Από τη δειγματοληψία προκύπτουν οι παρακάτω τιμές, οι οποίες αντιπροσωπεύουν πλήρως τη συνάρτηση.

Χρόνος (sec) (/600)	1	2	3	4	5	6	7	8	9	10	11	12
Τάση (V)	5	8,6	0	8,6	5	0	-5	-8,6	-10	-8,6	-5	0



Σχήμα 7.7. Δειγματοληψία σε ημιτονοειδή κυματομορφή τάσης πλάτους 10 Volt

Η τεχνική της δειγματοληψίας χρησιμοποιείται, ιδιαίτερα σε συστήματα RADAR και στη μετάδοση δεδομένων (Data Transmission). Η παραπάνω αρχή παρουσιάζει δυσκολίες, όταν ένα σύνολο φυσικών μεταβλητών πρέπει να δειγματοληπτηθεί την ίδια χρονική στιγμή, και οι αντίστοιχες ψηφιακές τιμές πρέπει όλες να τροφοδοτήσουν την είσοδο ενός ψηφιακού υπολογιστή. Η λύση στο πρόβλημα αυτό δίδεται δειγματοληπτώντας καθεμία από τις μεταβλητές εισόδου κυκλικά και στέλνοντας στον υπολογιστή τα δεδομένα που έχουν προκύψει διαδοχικά. Η τεχνική αυτή είναι γνωστή με το όνομα *πολυπλεξία* (multiplexing).

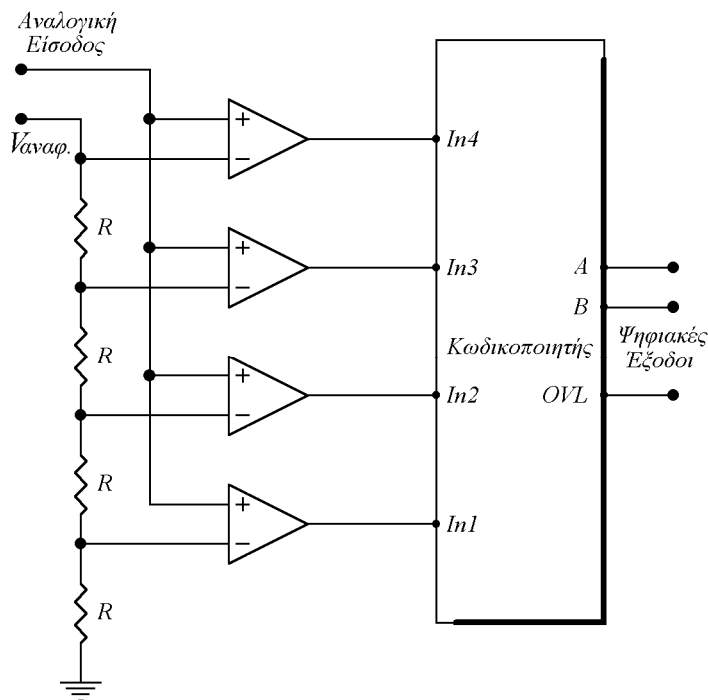
Στη συνέχεια θα μιλήσουμε για μερικά αντιπροσωπευτικά κυκλώματα μετατροπών που χρησιμοποιούνται σήμερα σε πολλές ηλεκτρονικές εφαρμογές που απαιτούν τη χρήση υπολογιστών

Ένας απλός μετατροπέας μεγάλης ταχύτητας αλλά υψηλού κόστους μπορεί να κατασκευασθεί πολύ εύκολα, χρησιμοποιώντας αναλογικούς συγκριτές, έναν ψηφιακό κωδικοποιητή και μια πηγή τάσης αναφοράς. Στο Σχήμα 7.8 απεικονίζεται ένας μετατροπέας ικανός να αναδεικνύει 2 bit πληροφορίας (και 1 υπερχειλίσ, OVL) για κάθε δείγμα του μετρούμενου μεγέθους.

Η πηγή τάσης αναφοράς καθορίζει τη μέγιστη τιμή της τάσης εισόδου που είναι σε θέση να μετρήσει ο μετατροπέας. Ο διαιρέτης των αντιστάσεων φροντίζει να οδηγούνται στις αναστρέφουσες εισόδους των συγκριτών ίσα κλάσματα της τάσης αναφοράς. Ο κωδικοποιητής δέχεται τις λογικές καταστάσεις που αναδεικνύονται στις εξόδους των συγκριτών και, σύμφωνα με τους συνδυασμούς τους, αναδεικνύει στις εξόδους του

τη δυαδική τιμή που αντιστοιχεί στο μετρούμενο μέγεθος. Η άγνωστη τάση οδηγείται σ' όλες τις μη αναστρέφουσες εισόδους των συγκριτών ταυτόχρονα.

Το κύκλωμα παρουσιάζει εξαιρετικά χαρακτηριστικά σε ό,τι αφορά στην ταχύτητα (χρόνος μετατροπής της τάση των 50...100 nsec), αλλά θεωρείται πολύ ακριβό από οικονομική άποψη, αφού το πλήθος των συγκριτών που απαιτούνται αυξάνεται υπερβολικά, καθώς το σύνολο των bit εξόδου αυξάνεται επίσης. Σκεφθείτε ότι για έξοδο 8 bit χρειάζονται $2^8 = 256$ συγκριτές. Λόγω της υψηλής ταχύτητας ονομάζεται και μετατροπέας Flash.



Σχήμα 7.8. Μετατροπέας τύπου Flash

Ο πίνακας 7.1 που ακολουθεί βοηθά στην κατανόηση της λειτουργίας του κυκλώματος.

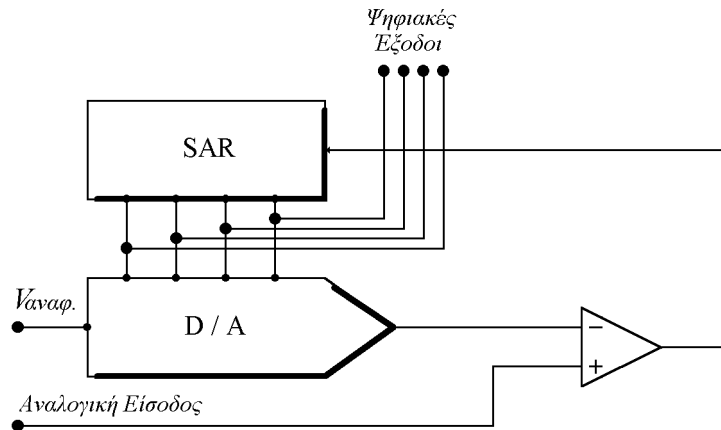
Τάση εισόδου (V)	Είσοδοι In1, In2, In3, In4	Έξοδοι A, B, OVL
0 - 1	0000	00—0
1 - 2	0001	01—0
2 - 3	0011	10—0
3 - 4	0111	11—0
>4	1111	11—1

Πίνακας 7.1. Σχέση των τιμών εισόδου και των ψηφίων εξόδου μετατροπέα flash

Ένα άλλο κύκλωμα περισσότερο οικονομικό αλλά και περισσότερο αργό, είναι αυτό που είναι γνωστό με το όνομα «Μετατροπέας διαδοχικών προσεγγίσεων». Όπως υποδηλώνει και το όνομα του, η αναζήτηση της ακριβούς δυαδικής ακολουθίας που προσδιορίζει την τιμή του γίνεται με αλληπάλληλες διαδοχικές προσπάθειες, από τις οποίες παράγεται κάθε φορά και ένα καινούριο ψηφίο πληροφορίας.

Στο σχήμα 7.9 δίνεται το διάγραμμα βαθμίδων ενός τέτοιου μετατροπέα. Όπως εύκολα γίνεται αντιληπτό, ο μετατροπέας αυτού του τύπου αποτελείται από ένα μόνο συγκριτή, ένα μετατροπέα ψηφιακού σήματος σε αναλογικό (D/A), την πηγή τάσης αναφοράς από την οποία εξαρτάται η περιοχή μέτρησης, έναν ταλαντωτή, και τέλος, ένα καινούριο εξάρτημα, που είναι γνωστό με το όνομα «Καταχωρητής διαδοχικών προσεγγίσεων» (Successive Approximation Register, SAR). Η λειτουργία του SAR μοιάζει με εκείνη ενός κοινού καταχωρητή ολίσθησης δεξιάς μετατόπισης, με τη διαφορά ότι κάθε καινούριο bit πληροφορίας που εισέρ-

χεται από το αριστερό του μέρος μένει ακλόνητο, αφήνοντας να ολισθήσουν τα υπόλοιπα. Ας δούμε όμως ένα παράδειγμα για την κατανόηση του οποίου θα χρειασθεί να συμβουλευθούμε τον πίνακα 7.2.



Σχήμα 7.9. Μετατροπέας A/D διαδοχικών προσεγγίσεων

Έστω ότι έχουμε έναν SAR των 6 bit στον οποίο η αρχική πληροφορία είναι η $Q_5Q_4Q_3Q_2Q_1Q_0$. Στον πίνακα, που ακολουθεί, φαίνεται πώς θα διαμορφώνεται η τιμή του, αν εισάγουμε σ' αυτόν 6 νέα ψηφία ($D_0...D_5$) από την αριστερή του πλευρά. Παρατηρήστε ακόμα πως τη στιγμή που μπαίνει το D_1 , αυτό τοποθετείται δεξιότερα από το D_0 . Το ίδιο συμβαίνει και με το D_2 που εισάγεται δεξιότερα των D_0, D_1 κ.λπ.

Εισαγόμενο ψηφίο	Έξοδοι SAR
-	$Q_5Q_4Q_3Q_2Q_1Q_0$
D_0	$D_0Q_5Q_4Q_3Q_2Q_1$
D_1	$D_0D_1Q_5Q_4Q_3Q_2$
D_2	$D_0D_1D_2Q_5Q_4Q_3$
D_3	$D_0D_1D_2D_3Q_5Q_4$
D_4	$D_0D_1D_2D_3D_4Q_5$
D_5	$D_0D_1D_2D_3D_4D_5$

Πίνακας 7.2. Τρόπος εισαγωγής των εισαγομένων ψηφίων σ' ένα SAR

Ας αναλύσουμε τη λειτουργία αυτού του μετατροπέα, θεωρώντας ότι έχουμε ένα κύκλωμα ίδιο μ' αυτό του προηγούμενου σχήματος, ικανό να αναδείξει 4 bit δυαδικής πληροφορίας. Θα υποθέσουμε πως η τάση αναφοράς είναι 16 V και πως η προς μέτρηση τάση έχει τιμή ίση με 11,8 V.

A/A	Έξοδοι SAR	Τάση εξόδου D/A	Έξοδος συγκριτή	Έλεγχος περιοχής	Σχόλια
	$0111_2 = 7_{10}$				δεν γίνεται μέτρηση
1	$0111_2 = 7_{10}$	7 V	1	0 - 15 V	$V_x > 7$ V
2	$1011_2 = 11_{10}$	11 V	1	7 - 15 V	$V_x > 11$ V
3	$1101_2 = 13_{10}$	13 V	0	11 - 15 V	$V_x < 13$ V
4	$1100_2 = 12_{10}$	12 V	0	11 - 13 V	$V_x < V_{D/A}$
	$1100_2 = 12_{10}$				λήξη μέτρησης

Πίνακας 7.3. Οι διαδοχικές φάσεις μιας μετατροπής βασισμένη σε SAR

Αρχικά οι έξοδοι του μετατροπέα που συμπίπτουν με τις εξόδους του SAR έχουν την τιμή $0111_2 = 7_{10}$. Αυτή η τιμή δημιουργεί στην έξοδο του μετατροπέα D/A, και συνεπώς στην αναστρέφουσα είσοδο του συγκριτή, τάση ίση με: $16 / 2^4 * 7 = 7$ V, τάση που μπορεί να θεωρηθεί ότι είναι ίση με το 1/2 της τάσης αναφοράς. Η άγνωστη τάση συγκρίνεται με τα 7 V, με αποτέλεσμα η έξοδος του συγκριτή να οδηγηθεί σε λογικό

«1» ($V_x > V_{D/A}$). Η λογική αυτή κατάσταση εισάγεται στον SAR, καταλαμβάνοντας την αριστερή θέση και ολισθαίνοντας τα υπόλοιπα bit. Η νέα τιμή, που προκύπτει, είναι η $1011_2 = 11_{10}$

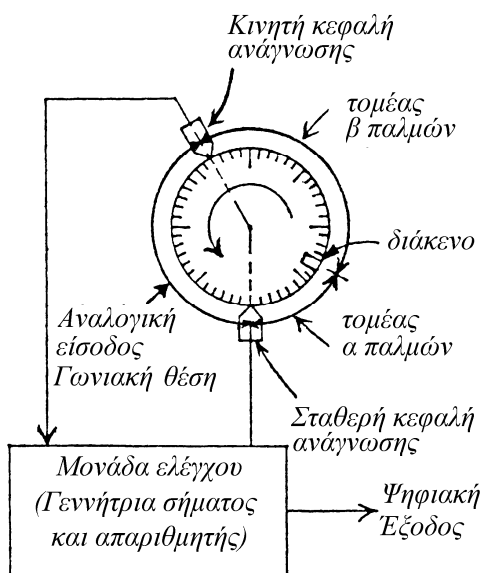
Η τιμή αυτή προκαλεί στη έξοδο του μετατροπέα A/D τάση ίση με 11 V που αντιστοιχεί στο μέσον της περιοχής τάσεων 15 V και 7 V που αντιπροσωπεύουν το μέγιστο και το μισό της τάσης που μπορεί να δημιουργήσει ο μετατροπέας D/A στην έξοδό του. Το αποτέλεσμα από τη νέα σύγκριση θα αναπαριστάνεται και εδώ με λογικό «1», αφού και τώρα ισχύει ότι $V_x > V_{D/A}$. Θα ακολουθήσουν και άλλες δύο προσπάθειες, τα αποτελέσματα των οποίων φαίνονται στον πίνακα 7.3, για να καταλήξουμε, τέλος, στο συμπέρασμα ότι η «άγνωστη τάση» αντιπροσωπεύεται από τη δυαδική τιμή $1100_2 = 12_{10}$, μια τιμή πολύ κοντά στην πραγματική των 11,8 V.

Εμβαθύνοντας καλύτερα την αρχή λειτουργίας του μετατροπέα διαδοχικών προσεγγίσεων, διαπιστώνουμε ότι με την πραγματοποίηση τόσων συγκρίσεων όσα είναι και τα bit που αναδεικνύει, «τεμαχίζει» την περιοχή τάσεων που καθορίζεται από την τάση αναφοράς, διχοτομώντας κάθε φορά ένα τμήμα της μέσα στο οποίο είναι εγκλωβισμένη η άγνωστη τάση. Στο παράδειγμα που χρησιμοποιήσαμε, η πρώτη σύγκριση διχοτομεί όλη την περιοχή τάσεων σε δύο μέρη. (0 - 7V και 7V - 15V). Επειδή η ένδειξη που λαμβάνει από τον συγκριτή είναι «1», η επόμενη τάση σύγκρισης διχοτομεί την υψηλότερη περιοχή με σκοπό την εξακρίβωση του τμήματος μέσα στο οποίο ανήκει η άγνωστη τάση. Η διαδικασία αυτή συνεχίζεται, μέχρις ότου πραγματοποιηθούν οι αναγκαίες συγκρίσεις.

Η ταχύτητα αυτού του μετατροπέα εξαρτάται, κατά κύριο λόγο, από τη συχνότητα της γεννήτριας σημάτων χρονισμού και, κατά δεύτερο λόγο, από το χρόνο που απαιτεί ο μετατροπέας D/A να αναδείξει την αναλογική τάση και από την ταχύτητα του συγκριτή. Αν η γεννήτρια εργάζεται, για παράδειγμα στους 100 KHz ($T = 10 \mu\text{sec}$) και θεωρήσουμε μηδενικούς όλους τους δευτερεύοντες χρόνους, τότε για το κύκλωμα που αναπτύξαμε προηγουμένως θα απαιτείται συνολικός χρόνος $4 * 10 \mu\text{sec} = 40 \mu\text{sec}$. Αν έχουμε συχνότητα 200 KHz ($T = 5 \mu\text{sec}$) και μετατροπέα των 12 bit, τότε απαιτείται χρόνος $5 * 12 \mu\text{sec} = 60 \mu\text{sec}$.

7.2.1. Μετατροπή της πληροφορίας γωνίας άξονα σε ψηφιακή πληροφορία

Πολύ συχνά η αναλογική πληροφορία αποτελείται από τη γωνιακή θέση ενός άξονα, π.χ. ενός σερβοκινητήρα ή ενός ποτενσιόμετρου. Η μετατροπή μιας γωνιακής αναλογικής τιμής σε ψηφιακή μορφή είναι, συνήθως, μια απλή εργασία. (Κοσσίδης, 1996; Holdsworth, & Woods, 2002; Δεληγιάννης, 2004)



Σχήμα 7.10. Σχηματικό διάγραμμα μετατροπής αναλογικής γωνιακής θέσης σε ψηφιακή μορφή

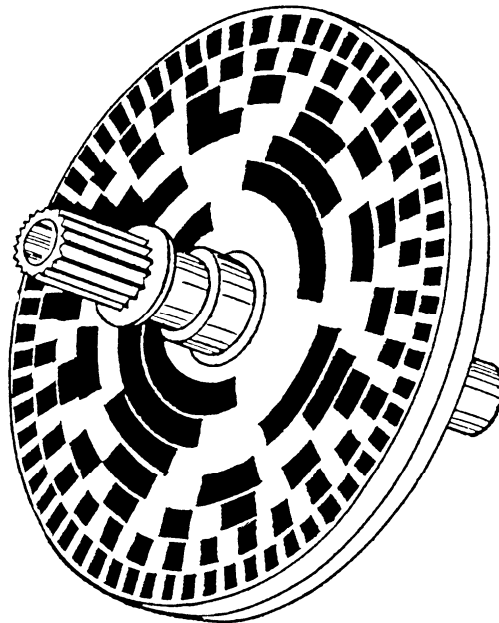
Το Σχήμα 7.10 δίνει σχηματικά έναν τρόπο μετατροπής, με τη βοήθεια δύο χαρακωμένων ομόκεντρων δίσκων οδηγούμενων από έναν σύγχρονο κινητήρα 3600 στροφών/λεπτό και από δύο κεφαλές ανάγνωσης.

Στην περιφέρεια του ενός δίσκου υπάρχουν χαραγές που διαιρούν όλο τον κύκλο σε κατάλληλο αριθμό υποδιαίρεσεων, π.χ. 360. Τα σημεία των χαραγών μπορούν να γίνουν έτσι, ώστε να παράγουν παλμούς, όταν πάνω απ' αυτά διέρχεται βραχίονας τύπου Pick-up που έχει μια σταθερή βούρτσα σε επαφή με την περι-

φέρεια του δίσκου ή μια φωτοηλεκτρική κυψέλη με πηγή φωτός ή τέλος μια μαγνητική κεφαλή ανάγνωσης. Για τον καθορισμό μιας θέσης ή ενός αριθμού περιστροφών, προσθέτουμε απλά όλους τους παλμούς που προέρχονται από τις χαραγές, με τη βοήθεια ενός απαριθμητή. Η γωνιακή θέση δίνεται από τη σχέση:

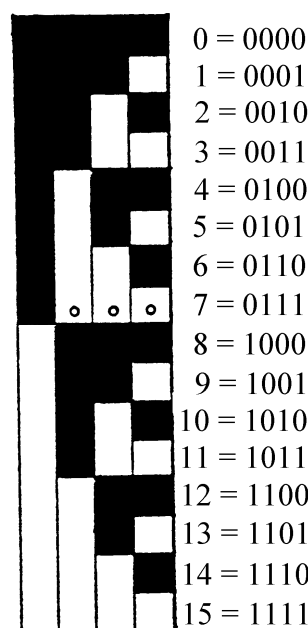
$$\text{Γωνιακή θέση} = \frac{\text{ολικός αριθμός παλμών} - (\alpha + \beta)}{\text{ολικός αριθμός παλμών}} \cdot 360$$

Για ολικό αριθμό παλμών π.χ. 360 και για $\alpha=60$ και $\beta=130$ έχουμε γωνιακή θέση 170° . Για ολικό αριθμό παλμών 400 έχουμε $\alpha=66,6$ και $\beta=144,4$ και γωνιακή θέση $170,1^\circ$.



Σχήμα 7.11. Δίσκος μετατροπής αναλογικών σημάτων σε ψηφιακά

Μια άλλη λύση για το ίδιο πρόβλημα φαίνεται στο Σχήμα 7.11, όπου φαίνεται ένας κωδικοποιημένος σε δυαδική μορφή δίσκος, που μπορεί να είναι μέρος ενός μετατροπέα αναλογικού σήματος σε ψηφιακό.



Σχήμα 7.12. Ευθυγραμμισμένο τμήμα κωδικοποιημένου δίσκου

Ο δίσκος αποτελείται από ασυνεχείς ομόκεντρους δακτυλίους, των οποίων ο αριθμός καθορίζει τον αριθμό των ψηφίων του ισοδύναμου δυαδικού αριθμού που παράγεται. Οι δακτύλιοι διαιρούνται σε ίσα γωνιακά τμήματα, των οποίων ο αριθμός εξαρτάται από τη θέση του δακτυλίου πάνω στο δίσκο.

Τα εμβαδά, που δημιουργούνται από την τομή μεταξύ των δακτυλίων και των ακτινωτών τμημάτων, παριστάνουν τα ψηφία ενός δυαδικού κώδικα, που διαβάζεται ακτινοειδώς κατά μήκος των ακτινωτών ευθύγραμμων τμημάτων. Σε κάθε θέση του άξονα περιστροφής αντιστοιχούν διάφορα ψηφία. Τα εν λόγω εμβαδά καθορίζονται ακριβώς, ώστε η ανάγνωση σ' αυτά να είναι δυνατόν να γίνει τόσο οπτικά, όσο και ηλεκτρικά ή μαγνητικά. Εφαρμογές αυτού του είδους χρησιμοποιούν, ως επί το πλείστον, οπτικό τρόπο ανάγνωσης.

Στο Σχήμα 7.12 παριστάνεται ένα ευθυγραμμισμένο τμήμα δίσκου που αποτελείται από τέσσερις δακτυλίους. Παράπλευρα δίνεται η αντίστοιχη τιμή για κάθε θέση, σε δυαδική μορφή.

7.3. Αποκωδικοποιητές

Θα μελετήσουμε αναλυτικά το ευρέως χρησιμοποιούμενο κύκλωμα του αποκωδικοποιητή, που χρησιμεύει στα παραδείγματα της ανάλυσης κυκλωμάτων και των τεχνικών σχεδίασης. Θα χρησιμοποιήσουμε αρκετές φορές αποκωδικοποιητές και πολυπλέκτες σαν δομικά στοιχεία κατασκευής, πολυπλοκότερων κυκλωμάτων.

Παλαιότερα οι εκτυπωτές έμοιαζαν σαν τις γραφομηχανές. Έτσι, για να εκτυπωθεί το γράμμα «Α», ένας τροχός γύριζε φέρνοντας την ακίδα «Α» ψηλά, και εκείνη, κατόπιν, χτυπούσε πάνω στο χαρτί.

Τα γράμματα, όπως γνωρίζουμε, κωδικοποιούνται σε κώδικα των 8 bit μέσα στον υπολογιστή. (π.χ. ASCII). Όταν ανιχνευθεί ο συγκεκριμένος συνδυασμός των bit, που κωδικοποιεί, π.χ. το «Α», θέλουμε να ενεργοποιήσουμε τη γραμμή εξόδου που αντιστοιχεί στο Α. Για να μπορέσουμε να ανιχνεύσουμε συγκεκριμένη ακολουθία από δυαδικά ψηφία, χρησιμοποιούμε τον αποκωδικοποιητή.

Με δεδομένη μία είσοδο από k bit, ανιχνεύουμε ποιος από τους 2k συνδυασμούς αναπαρίσταται και παράγουμε 2k εξόδους, από τις οποίες μόνο μία είναι «1». (Givone, 2002; Δεληγιάννης, 2004; Κοσσίδης, & Γιαννακόπουλος, 2006; Wakerly, 2006; Balabanian, & Carlson, 2007; Godse, & Godse, 2010; Tocci, et al., 2010; Mano, & Ciletti, 2014; Roth, & Kinney, 2014; Pritchard, 2015)

7.3.1. Λειτουργία του αποκωδικοποιητή

Ένας αποκωδικοποιητής n-σε-2ⁿ δέχεται μία είσοδο από n-bit και παράγει 2ⁿ εξόδους. Κατόπιν οι n είσοδοι αναπαριστούν έναν δυαδικό αριθμό που καθορίζει ποια, -μοναδική- από τις 2ⁿ εξόδους, είναι αληθής.

Ένας αποκωδικοποιητής 2-σε-4 λειτουργεί σύμφωνα με τον πίνακα 7.4.

Η δύο-δυαδικών ψηφίων είσοδος ονομάζεται S1S0, και οι τέσσερις εξοδοί έστω Q0-Q3. Αν η είσοδος είναι ο δυαδικός αριθμός i, τότε μόνο η έξοδος Qi είναι αληθής.

Για παράδειγμα, αν η είσοδος S1S0 = 10 (δεκαδικό 2), τότε η έξοδος Q2 είναι αληθής (1), και όλες οι υπόλοιπες Q0, Q1 και Q3 είναι ψευδής (0).

S1	S0	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Πίνακας 7.4. Πίνακας αλήθειας του αποκωδικοποιητή

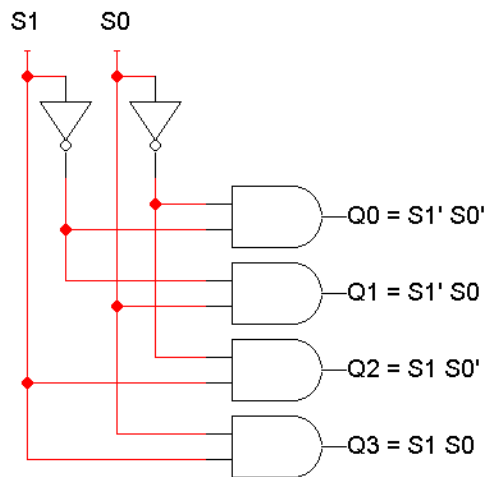
Αυτό το κύκλωμα «αποκωδικοποιεί» έναν δυαδικό αριθμό σε κώδικα «έναν από τέσσερις». Για να κατασκευάσουμε ένα αποκωδικοποιητή 2-σε-4, ακολουθούμε τις διαδικασίες σχεδίασης που ήδη γνωρίζουμε: έχουμε τον πίνακα αληθείας, και έτσι μπορούμε να γράψουμε τις εξισώσεις για κάθε μία από τις εξόδους (Q0-Q3), συναρτήσει των δύο εισόδων (S0-S1).

Σε αυτήν την περίπτωση, δεν υπάρχουν απλοποιήσεις, οπότε οι εξισώσεις είναι: (με τονούμενα εννοούμε τους αντιστροφείς).

$$\begin{aligned} Q0 &= S1' S0' & Q1 &= S1' S0 \\ Q2 &= S1 S0' & Q3 &= S1 S0 \end{aligned}$$

7.3.2. Αποκωδικοποιητής 2-σε-4

Χρησιμοποιώντας τον πίνακα αλήθειας (πίνακας 7.4), έχουμε το διάγραμμα του σχήματος 7.13 για τον αποκωδικοποιητή 2 σε 4:



Σχήμα 7.13. Αναλυτικό διάγραμμα αποκωδικοποιητή 2-σε-4

7.3.3. Είσοδοι ενεργοποίησης (Enable)

Πολλές συσκευές έχουν μία πρόσθετη είσοδο την EN (enable), η οποία χρησιμοποιείται για να «ενεργοποιηθεί» ή να «απενεργοποιηθεί» τη συσκευή.

EN	S1	S0	Q0	Q1	Q2	Q3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Πίνακας 7.5. Πίνακας αληθείας του αποκωδικοποιητή με είσοδο EN

Για έναν αποκωδικοποιητή, όταν το EN=1 ενεργοποιεί τον αποκωδικοποιητή, για να δουλέψει, όπως έχει ήδη περιγραφεί. Ακριβώς μία από τις εξόδους θα είναι 1. Όταν το EN=0 απενεργοποιείται ο αποκωδικοποιητής. Συμβατικά, αυτό σημαίνει ότι όλες οι εξόδους του αποκωδικοποιητή, θα είναι 0.

Μπορούμε να συμπεριλάβουμε αυτήν την πρόσθετη είσοδο στον πίνακα αληθείας του αποκωδικοποιητή, και προκύπτει ο πίνακας 7.5 που ακολουθεί. Από τον πίνακα 7.5 παρατηρούμε, ότι, όταν η είσοδος EN=0, οι εξόδους είναι πάντα 0, ανεξάρτητα από την τιμή των εισόδων S1 και S0.

EN	S1	S0	Q0	Q1	Q2	Q3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

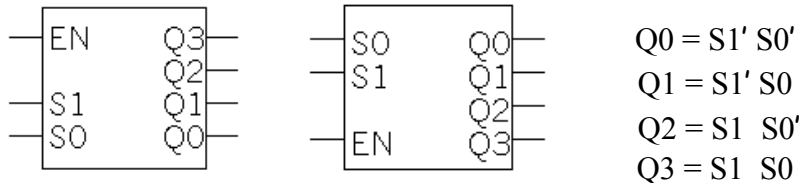
Πίνακας 7.6. Συντομευμένος πίνακας αληθείας του αποκωδικοποιητή με είσοδο EN

Μπορούμε να συντομεύσουμε τον πίνακα 7.5, θεωρώντας τα S1 και S0 σαν αδιάφορους όρους (x) όταν το EN=0. Έτσι προκύπτει ο συντομευμένος πίνακας 7.6.

7.3.4. Δομικά διαγράμματα

Οι αποκωδικοποιητές χρησιμοποιούνται τόσο συχνά, που θέλουμε να τους ενθυλακώσουμε και να τους θεωρήσουμε ξεχωριστές οντότητες.

Τα δομικά διαγράμματα για τους αποκωδικοποιητές 2-σε-4 αποδίδονται στο Σχήμα 7.14. Τα ονόματα των εισόδων και των εξόδων έχουν σημασία, και όχι η σειρά με την οποία τοποθετούνται.



Σχήμα 7.14. Δομικό διάγραμμα αποκωδικοποιητή 2-σε-4

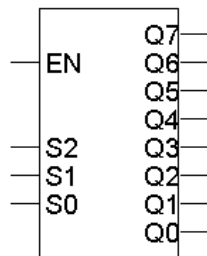
Το δομικό διάγραμμα του αποκωδικοποιητή παρέχει «αφαίρεση» (abstraction), καθώς:

Μπορούμε να χρησιμοποιήσουμε τον αποκωδικοποιητή, αρκεί να γνωρίζουμε τον πίνακα αληθείας του ή τις εξισώσεις, χωρίς να γνωρίζουμε ακριβώς από τι αποτελείται το εσωτερικό του. Έτσι, κρύβοντας την εσωτερική δομή του κυκλώματος, τα διαγράμματα γίνονται ευκολότερα.

Επίσης απλοποιεί την επαναχρησιμοποίηση του υλικού. Δε χρειάζεται να τον σχεδιάζουμε από την αρχή κάθε φορά που το χρειαζόμαστε. Αυτά τα δομικά στοιχεία (blocks) είναι σαν τις συναρτήσεις του προγραμματισμού!

7.3.5. Αποκωδικοποιητής 3-σε-8

Και οι μεγαλύτεροι αποκωδικοποιητές είναι παρόμοιοι με τον αποκωδικοποιητή 2-4, που ήδη μελετήσαμε. Παραθέτουμε το διάγραμμα ενός αποκωδικοποιητή 3-σε-8. Το δομικό του διάγραμμα αποδίδεται στο Σχήμα 7.15 και ο πίνακας αληθείας (χωρίς το EN) δίνεται στον πίνακα 7.7. Και πάλι μόνο μία έξοδος είναι αληθής για οποιοδήποτε συνδυασμό εισόδων.



Σχήμα 7.15. Δομικό διάγραμμα αποκωδικοποιητή 3-σε-8

S2	S1	S0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Πίνακας 7.7. Πίνακας αληθείας αποκωδικοποιητή 3-σε-8

Μπορούμε να φτιάξουμε έναν αποκωδικοποιητή 3-σε-8 απευθείας από τον πίνακα αληθείας και τις εξισώσεις που φαίνονται παρακάτω, ακριβώς όπως κάναμε και με τον αποκωδικοποιητή (A/K) 2-σε-4.

Στον πίνακα 7.7, παρατηρούμε μερικά πρότυπα μορφών (patterns):

Όταν S2=0, ενεργοποιούνται οι έξοδοι Q0-Q3, όπως σε ένα A/K 2-σε-4.

Όταν S2=1, ενεργοποιούνται οι έξοδοι Q4-Q7, όπως σε ένα A/K 2-σε-4.

Οι συναρτήσεις εξόδου δίνονται από τις σχέσεις: (Πίνακας 7.7)

$$\begin{aligned} Q0 &= S2' S1' S0' = m_0 & Q4 &= S2 S1' S0' = m_4 \\ Q1 &= S2' S1' S0 = m_1 & Q5 &= S2 S1' S0 = m_5 \\ Q2 &= S2' S1 S0' = m_2 & Q6 &= S2 S1 S0' = m_6 \\ Q3 &= S2' S1 S0 = m_3 & Q7 &= S2 S1 S0 = m_7 \end{aligned}$$

7.3.6. Πλεονεκτήματα αποκωδικοποιητή

Με βάση τον πίνακα 7.4 και τις εξισώσεις του αποκωδικοποιητή 2-σε-4, εύκολα αντιλαμβανόμαστε γιατί οι αποκωδικοποιητές μερικές φορές καλούνται γεννήτριες ΕΛΒΟ. Για κάθε συνδυασμό εισόδου, ακριβώς μία έξοδος είναι αληθής και κάθε εξίσωση εξόδου περιέχει όλες τις μεταβλητές εισόδου.

Αυτές οι ιδιότητες ισχύουν για όλα τα «μεγέθη» των αποκωδικοποιητών. Αυτό σημαίνει ότι μπορούμε να υλοποιήσουμε αυθαίρετες συναρτήσεις με αποκωδικοποιητές. Εάν έχουμε για μία συνάρτηση μία εξίσωση με αθροίσματα ΕΛΒΟ, τότε μπορούμε εύκολα να χρησιμοποιήσουμε έναν αποκωδικοποιητή (μία γεννήτρια ΕΛΒΟ), για να υλοποιήσουμε τη συνάρτηση αυτή.

7.4. Χρήση αποκωδικοποιητών

7.4.1. Πρόσθεση με τη χρήση αποκωδικοποιητών

Έστω ότι θέλουμε να κατασκευάσουμε ένα κύκλωμα το οποίο να προσθέτει τρεις εισόδους X, Y και Z, του 1-bit. Θα χρειαστούμε 2 bit για την αναπαράσταση του αποτελέσματος, έστω C (carry) και S (sum), για κρατούμενο και άθροισμα αντίστοιχα.

Παρατηρούμε ότι τα C και S είναι δύο ξεχωριστές συναρτήσεις των ιδίων εισόδων X, Y και Z. Ο πίνακας αληθείας και οι συναρτήσεις των C και S συναρτήσει των ΕΛΒΟ είναι οι ακόλουθες (πίνακας 7.8):

$C(X,Y,Z) = \Sigma(3,5,6,7)$ και για το $S(X,Y,Z) = \Sigma(1,2,4,7)$ αντίστοιχα.

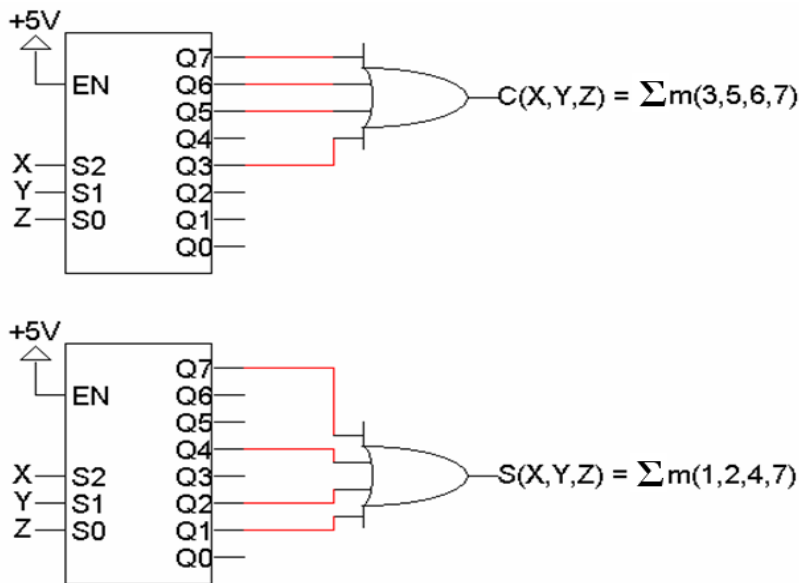
X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$0 + 1 + 1 = 10$
 $1 + 1 + 1 = 11$

Πίνακας 7.8. Πίνακας αληθείας του αθροίσματος και του κρατουμένου

Δύο αποκωδικοποιητές 3-σε-8 υλοποιούν τα C και S σαν αθροίσματα των ΕΛΒΟ. (Σχήμα 7.16)

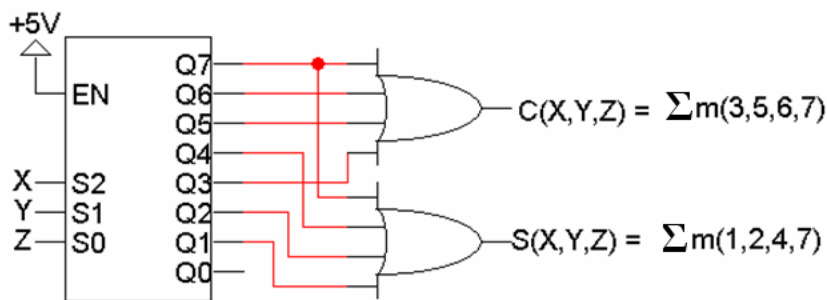
Την τάση «+5V» («5 volts») (λογικό «1») τη χρησιμοποιούμε, για να εξασφαλίσουμε ότι ο αποκωδικοποιητής θα είναι πάντα ενεργός (active).



Σχήμα 7.16. Υλοποίηση του αθροίσματος και του κρατουμένου με 2 αποκωδικοποιητές 3-σε-8

7.4.2. Υλοποίηση με τη χρήση ενός αποκωδικοποιητή

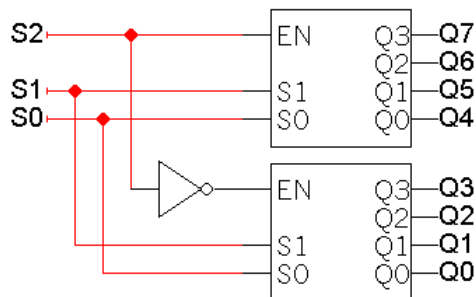
Καθώς οι συναρτήσεις των C και S έχουν και οι δύο τις ίδιες εισόδους, μπορούμε να χρησιμοποιήσουμε μόνο έναν αποκωδικοποιητή, αντί για δύο (Σχήμα 7.16). Έτσι προκύπτει το διάγραμμα του Σχήματος 7.17.



Σχήμα 7.17. Υλοποίηση του αθροίσματος και του κρατουμένου με ένα αποκωδικοποιητή 3-σε-8

7.4.3. Επέκταση των αποκωδικοποιητών

Για να συνδέσουμε αποκωδικοποιητές μεταξύ τους, μπορούμε να χρησιμοποιήσουμε τις εισόδους EN. Στο Σχήμα 7.18 παραθέτουμε ένα δομικό διάγραμμα αποκωδικοποιητή 3-σε-8, κατασκευασμένο από δύο αποκωδικοποιητές 2-σε-4:



Σχήμα 7.18. Υλοποίηση αποκωδικοποιητή 3-σε-8 με δύο αποκωδικοποιητές 2-σε-4

Στο παραπάνω διάγραμμα χρειάζεται προσοχή να μην μπερδέψουμε τις «εσωτερικές» εισόδους και εξόδους των αποκωδικοποιητών 2-σε-4 με τις «εξωτερικές» εισόδους και εξόδους του αποκωδικοποιητή 3-σε-8 (με έντονη γραφή). Αυτό είναι παρόμοιο με το να έχουμε σε ένα πρόγραμμα διαφορετικές συναρτήσεις οι οποίες να χρησιμοποιούν την ίδια παράμετρο «x».

Μπορούμε εύκολα να επαληθεύσουμε ότι αυτό το κύκλωμα είναι ένας αποκωδικοποιητής 3-σε-8, χρησιμοποιώντας τις εξισώσεις του αποκωδικοποιητή 2-σε-4, για να παράγουμε τις εξισώσεις του 3-σε-8.

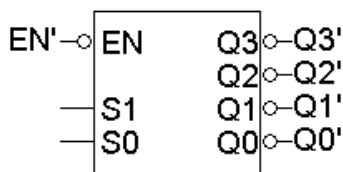
7.4.4. Τροποποίηση του συνήθους αποκωδικοποιητή

Οι αποκωδικοποιητές που είδαμε μέχρι τώρα είναι όλοι active-high. (Σχήμα 7.14 και πίνακας 7.6)

Οι active-high αποκωδικοποιητές παράγουν *ELBO*, όπως έχουμε ήδη δει. Έτσι οι εξισώσεις είναι:

$$Q_3 = S_1 S_0, Q_2 = S_1 S_0', Q_1 = S_1' S_0 \text{ και } Q_0 = S_1' S_0'.$$

Ένας active-low αποκωδικοποιητής είναι ακριβώς ίδιος με τον προηγούμενο, μόνο που έχει μια ανάστροφη είσοδο EN και ανάστροφες εξόδους. (Σχήμα 7.19)



Σχήμα 7.19. Δομικό διάγραμμα του active-low αποκωδικοποιητή 2-σε-4

EN'	S1	S0	Q0'	Q1'	Q2'	Q3'
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	x	x	1	1	1	1

Πίνακας 7.9. Πίνακας αλήθειας του active-low αποκωδικοποιητή 2-σε-4

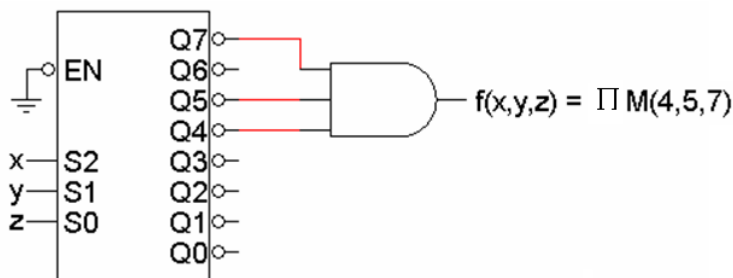
Οι εξισώσεις εξόδου για έναν active-low αποκωδικοποιητή είναι παρόμοιες, αλλά ταυτόχρονα είναι κάπως διαφορετικές:

$$Q_3' = (S_1 S_0)' = S_1' + S_0', Q_2' = (S_1 S_0')' = S_1' + S_0$$

$$Q_1' = (S_1' S_0)' = S_1 + S_0', Q_0' = (S_1' S_0')' = S_1 + S_0$$

Συμπερασματικά οι active-low αποκωδικοποιητές παράγουν *MEBO*.

7.4.5. Παράδειγμα Active-low αποκωδικοποιητή



Σχήμα 7.20. Υλοποίηση της $f(x,y,z) = \Pi M(4,5,7)$ με αποκωδικοποιητή active-low

Μπορούμε λοιπόν να χρησιμοποιήσουμε active-low αποκωδικοποιητές, για να υλοποιήσουμε και αυθαίρετες συναρτήσεις, αλλά σαν γινόμενο MEBO. Για παράδειγμα, παρουσιάζουμε την υλοποίηση της συνάρτησης, $f(x,y,z) = \Pi M(4,5,7)$, με τη χρήση ενός αποκωδικοποιητή active-low. (Σχήμα 7.20)

Η γείωση που συνδέεται στην είσοδο EN (Σχήμα 7.20) αναπαριστά το λογικό 0, και, συνεπώς, αυτός ο αποκωδικοποιητής είναι πάντα ενεργός. Υπενθυμίζουμε ότι για ένα γινόμενο των αθροισμάτων (POS), χρειαζόμαστε μία πύλη AND.

7.4.6. Παράγοντες σχεδίασης και η προέλευση τους

Τα κυκλώματα που φτιάχνονται με πύλες, χωρίς ανατροφοδότηση ονομάζονται *συνδυαστικά κυκλώματα*. Στα κυκλώματα αυτά οι έξοδοι δε συνδέονται με τις εισόδους. Αν αλλάξουμε τις τιμές των εισόδων και περιμένουμε λίγο, θα εμφανιστούν οι σωστές έξοδοι. Αυτό είναι απαραίτητο, καθώς υπάρχει Χωρητική Φόρτιση (capacitive loading): είναι ανάλογο με το: «γέμισε το νερό μέχρι τη στάθμη». Έτσι, όταν τέτοια κυκλώματα χρησιμοποιούνται σε έναν υπολογιστή, ο χρόνος που χρειάζεται, για να σταθεροποιηθούν οι έξοδοι, είναι σημαντικός. Για τον ίδιο λόγο, μια έξοδος δεν μπορεί να οδηγήσει πολλές εισόδους.

Θα είναι πολύ αργή, για να «τις γεμίσει», καθώς μπορεί να μην έχει αρκετή ενέργεια.

Έτσι τα κριτήρια της σχεδίασης είναι:

A) Η Καθυστέρηση διάδοσης (πόσοι φθάνουν σε μια σειρά από την είσοδο στην έξοδο)

Καθυστέρηση διάδοσης "propagation delay" ονομάζεται ο μέσος χρόνος που απαιτείται, για να αλλάξει κατάσταση το σήμα στην έξοδο μιας πύλης μετά την αλλαγή του σήματος στην είσοδο. Η καθυστέρηση διάδοσης καθορίζει τη μέγιστη ταχύτητα λειτουργίας της πύλης και γενικότερα των ψηφιακών κυκλωμάτων.

Ορίζονται δυο χρόνοι καθυστέρησης: Ο χρόνος καθυστέρησης, όταν η έξοδος μεταβαίνει από χαμηλή σε υψηλή στάθμη t_{PLH} , και ο χρόνος καθυστέρησης, όταν η έξοδος μεταβαίνει από υψηλή σε χαμηλή στάθμη t_{PHL} . Ο μέσος χρόνος καθυστέρησης t_{PAV} προκύπτει από το μέσο όρο των t_{PLH} και t_{PHL} . Οι χρόνοι αυτοί μετρώνται μεταξύ των σημείων όπου το σήμα βρίσκεται στη μέση στάθμη μεταξύ της χαμηλής και υψηλής στάθμης. Οι χρόνοι καθυστέρησης μετρώνται σε ns και δίνονται στα data books των κατασκευαστών.

B) Το Fan-out (Πλήθος εξόδων μιας πύλης)

Το fan-out ορίζεται ως μέγιστος αριθμός εισόδων ομοίων πυλών που μπορεί να συνδεθεί στην έξοδο μιας πύλης χωρίς να επηρεάσει την κανονική της λειτουργία. Προφανώς για κανονική λειτουργία οι τάσεις και τα ρεύματα εξόδου της πύλης πρέπει να παραμένουν εντός των ορίων που καθορίζονται από τις μέγιστες και ελάχιστες κατά περίπτωση τιμές.

Το τυπικό φορτίο -standard TTL load, για TTL κυκλώματα, καθορίζει το ποσό του ρεύματος που απαιτεί η είσοδος μιας πύλης, για να οδηγηθεί οπότε η ικανότητα οδήγησης μπορεί να δίνεται ,π.χ. ως 10 TTL standard loads. Η ικανότητα οδήγησης της εξόδου μιας πύλης βελτιώνεται με τη χρήση των απομονωτών (buffers) και

Γ) Το Fan-in το οποίο ορίζεται ως ο μέγιστος αριθμός εισόδων που μπορεί να διαθέτει μία πύλη σε μία συγκεκριμένη λογική οικογένεια.

7.5. Πολυπλέκτης

Ο πολυπλέκτης είναι ένα λογικό κύκλωμα με πολλές εισόδους και μια μόνο έξοδο. Η είσοδος που θα συνδεθεί ενεργά με την έξοδο εξαρτάται από την τιμή που έχουν οι μεταβλητές στις γραμμές επιλογής (S). (Givone, 2002; Δεληγιάννης, 2004; Κοσσίδης, & Γιαννακόπουλος, 2006; Wakerly, 2006; Balabanian, & Carlson, 2007; Godse, & Godse, 2010; Tocci, et al., 2010; Mano, & Ciletti, 2014; Roth, & Kinney, 2014; Pritchard, 2015). Οι πολυπλέκτες χρησιμοποιούνται, επίσης, ευρέως, όπως και οι αποκωδικοποιητές. Υπενθυμίζουμε ότι:

A) Χρησιμοποιούνται σαν παραδείγματα για την ανάλυση κυκλωμάτων και την αρθρωτή σχεδίαση.

B) Μπορούν να υλοποιήσουν αυθαίρετες συναρτήσεις.

Γ) Θα χρησιμοποιήσουμε αυτά τα κυκλώματα σαν δομικά στοιχεία πολυπλοκότερων κυκλωμάτων.

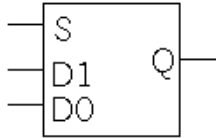
Οι πολυπλέκτες χρησιμοποιούνταν, για να επιλέξουμε μεταξύ των διαθέσιμων πόρων (ο γνωστός διακόπτης). Πριν από την ύπαρξη των δικτύων, αρκετοί υπολογιστές μπορούσαν να διαμοιραστούν έναν εκτυπωτή μέσω της χρήσης ενός διακόπτη.

Ένας πολυπλέκτης 2n-σε-1 στέλνει μία από τις 2n γραμμές εισόδου σε μία μόνο γραμμή εξόδου. Ένας πολυπλέκτης έχει, συνεπώς, δύο σύνολα εισόδων:

1) 2n γραμμές δεδομένων εισόδου

2) n γραμμές επιλογής, για να επιλέξουμε μία από τις 2n εισόδους δεδομένων

Η έξοδος του πολυπλέκτη είναι ένα μόνο bit, το οποίο είναι μία από τις 2n εισόδους δεδομένων. Το απλούστερο παράδειγμα είναι ένας πολυπλέκτης 2-σε-1: (Σχήμα 7.21)



$$\text{Συνάρτηση εξόδου: } Q = S'D0 + S D1$$

Σχήμα 7.21. Δομικό διάγραμμα και συνάρτηση του πολυπλέκτη 2-σε-1

Το bit επιλογής S (select) ελέγχει πιο από τα bit δεδομένων D0-D1 επιλέγεται. Έτσι, αν S= 0, τότε η έξοδος είναι το D0 (Q=D0) και εάν S= 1, τότε η έξοδος είναι το D1 (Q=D1).

7.5.1. Σύντμηση του πίνακα αλήθειας

Ο πίνακας αληθείας αποδίδεται στον πίνακα 7.10 και αντιστοιχεί στον πολυπλέκτη 2-σε-1, βασίζεται δε στην εξίσωση: $Q = S' D0 + S D1$

Στον πίνακα 7.11 που ακολουθεί παρουσιάζεται ο συντετμημένος πίνακας αληθείας. Οι μεταβλητές εισόδου εμφανίζονται στη στήλη της εξόδου.

Από τον πίνακα συνεπάγεται ότι, όταν S=0, η έξοδος Q=D0, και όταν S=1 η έξοδος Q=D1. Αυτό είναι σε συμφωνία με την εξίσωση.

S	D1	D0	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

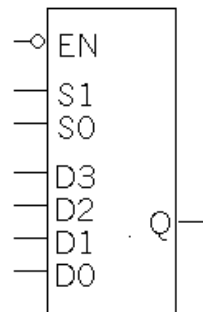
Πίνακας 7.10. Πίνακας αληθείας του πολυπλέκτη 2-σε-1

S	Q
0	D0
1	D1

Πίνακας 7.11. Συντετμημένος πίνακας αληθείας του πολυπλέκτη 2-σε-1

7.5.2. Πολυπλέκτης 4-σε-1

EN'	S1	S0	Q
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	x	x	1

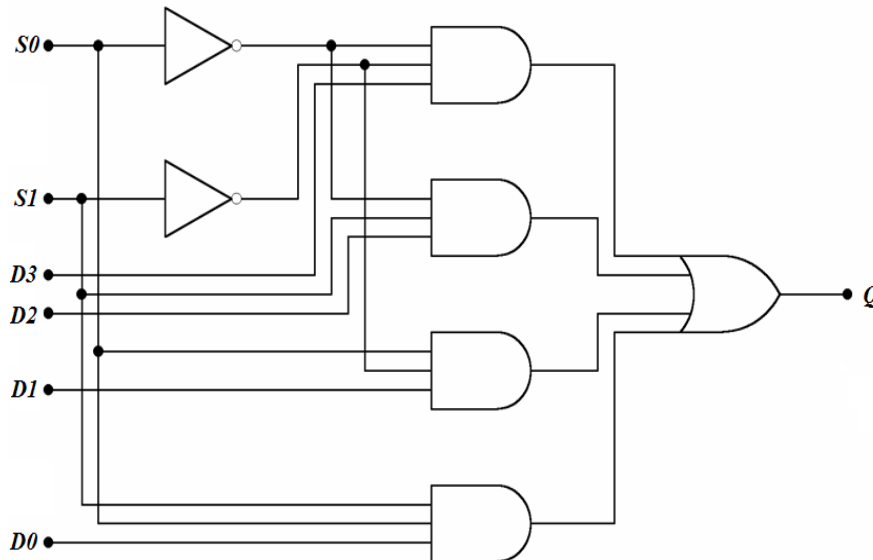


Πίνακας 7.12. Συντετμημένος πίνακας αληθείας και δομικό διάγραμμα του πολυπλέκτη 4-σε-1

Στον πίνακα 7.12 δίνεται το δομικό διάγραμμα και ο συντεταγμένος πίνακας αληθείας του πολυπλέκτη 4-σε-1. Όταν το $EN' = 1$ τότε ο πολυπλέκτης έχει πάντα έξοδο 1. Η συνάρτηση του πολυπλέκτη 4-σε-1 είναι:

$$Q = S1' S0' D0 + S1' S0 D1 + S1 S0' D2 + S1 S0 D3$$

Στο Σχήμα 7.22 δίδεται το αναλυτικό διάγραμμα ενός πολυπλέκτη 4-σε-1. Σημειώνουμε ότι και σ' αυτά τα εξαρτήματα υπάρχει μια είσοδος ενεργοποίησης. Όταν αυτή είναι ενεργή, τότε ο πολυπλέκτης εργάζεται όπως περιγράφηκε. Όταν είναι ανενεργή, τότε εμφανίζει στην έξοδό του, λογικό «0».



Σχήμα. 7.22. Αναλυτικό διάγραμμα πολυπλέκτη 4-σε-1

7.5.3. Υλοποίηση συναρτήσεων με πολυπλέκτες

Οι πολυπλέκτες χρησιμοποιούνται για την υλοποίηση αυθαίρετων συναρτήσεων. Ένας τρόπος υλοποίησης μιας συνάρτησης n μεταβλητών είναι να χρησιμοποιήσουμε έναν πολυπλέκτη n -σε-1:

Για κάθε ΕΛΒΟ m_i της συνάρτησης, θέτουμε «1» στην είσοδο δεδομένων D_i του πολυπλέκτη. Κάθε είσοδος δεδομένων αντιστοιχεί σε μία γραμμή του πίνακα αληθείας.

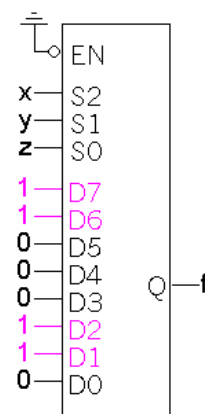
Συνδέουμε τις μεταβλητές εισόδου της συνάρτησης στις εισόδους επιλογής (bit επιλογής) του πολυπλέκτη. Αυτές χρησιμοποιούνται για να προσδιορίσουν ένα συγκεκριμένο συνδυασμό εισόδων.

7.5.3.1 Υλοποίηση συνάρτησης με πολυπλέκτη 8 σε 1

Έστω η συνάρτηση $f(x,y,z) = \Sigma m(1,2,6,7)$. Ακολουθεί η υλοποίηση με πολυπλέκτη 8-σε-1 καθώς και ο συντεταγμένος πίνακας αλήθειας της συνάρτησης. (Πίνακας 7.13)

Μπορούμε, επίσης, να υλοποιήσουμε την προηγούμενη συνάρτηση $f(x,y,z) = \Sigma m(1,2,6,7)$ με έναν πολυπλέκτη 4-σε-1, αντί για τον 8-σε-1.

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



Πίνακας 7.13. Πίνακας αληθείας και δομικό διάγραμμα του πολυπλέκτη 8-σε-1

Βήμα 1: Από τον πίνακα αληθείας της συνάρτησης ομαδοποιούμε τις σειρές σε ζεύγη. Μέσα σε κάθε ζεύγος γραμμών τα x και y είναι σταθερά, και, συνεπώς, η f είναι συνάρτηση μόνο του z .

Όταν $xy = 00$, τότε $f = z$

Όταν $xy = 01$, τότε $f = z'$

Όταν $xy = 10$, τότε $f = 0$

Όταν $xy = 11$, τότε $f = 1$

Βήμα 2: Συνδέουμε τις δύο πρώτες μεταβλητές εισόδου του πίνακα αληθείας (εδώ τα x και y) στα bit επιλογής $S1$ $S0$ του πολυπλέκτη 4-σε-1.

Βήμα 3: Συνδέουμε τις παραπάνω εξισώσεις της $f(z)$ στις εισόδους δεδομένων $D0$ - $D3$.

Ακολουθεί ο πίνακας αλήθειας και το δομικό διάγραμμα της υλοποίησης της συνάρτησης με πολυπλέκτη 4-σε-1. (Πίνακας 7.14)

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

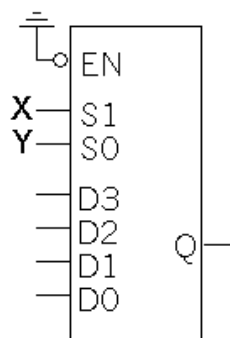
Πίνακας 7.14. Πίνακας αληθείας και δομικό διάγραμμα του πολυπλέκτη 4-σε-1

7.5.3.2 Αθροιστής με χρήση πολυπλέκτη

Ας υλοποιήσουμε τη συνάρτηση του κρατουμένου $C(X,Y,Z)$ του αθροιστή με πολυπλέκτες.

Καθώς έχουμε τρεις εισόδους, θα χρειαστούμε έναν πολυπλέκτη 4-σε-1. Αρχικά θα συνδέσουμε δύο από τις μεταβλητές εισόδου (συνήθως τις δύο πρώτες του πίνακα αληθείας) στις εισόδους επιλογής ($S1, S0$) του πολυπλέκτη.

X	Y	Z	C
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



Η εξίσωση (συνάρτηση) για τον πολυπλέκτη, με $S1=X$ και $S0=Y$, είναι

$$Q = X'Y'D_0 + X'YD_1 + XY'D_2 + XYD_3$$

Υπολογισμός της συνάρτησης του κρατουμένου και του αθροίσματος

Μπορούμε να ορίσουμε τις εισόδους δεδομένων του πολυπλέκτη $D0$ - $D3$, κρατώντας σταθερά τα X και Y και προσδιορίζοντας τις εξισώσεις του C συναρτήσει του Z .

Η συνάρτηση του κρατουμένου C δίνεται από τη σχέση:

$C = X' Y' D_0 + X' Y D_1 + X Y' D_2 + X Y D_3$ στην οποία χρησιμοποιούμε τον πίνακα που ακολουθεί και έχουμε $C = X' Y' 0 + X' Y Z + X Y' Z + X Y 1 = X' Y Z + X Y' Z + XY = \Sigma m(3, 5, 6, 7)$

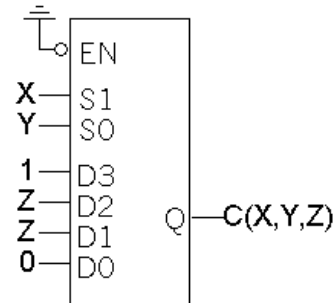
X	Y	Z	C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Όταν $XY=00$, $C=0$

Όταν $XY=01$, $C=Z$

Όταν $XY=10$, $C=Z$

Όταν $XY=11$, $C=1$



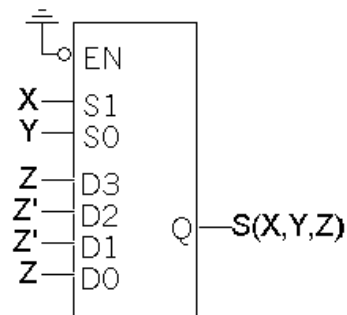
X	Y	Z	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Όταν $XY=00$, $S=Z$

Όταν $XY=01$, $S=Z'$

Όταν $XY=10$, $S=Z'$

Όταν $XY=11$, $S=Z$



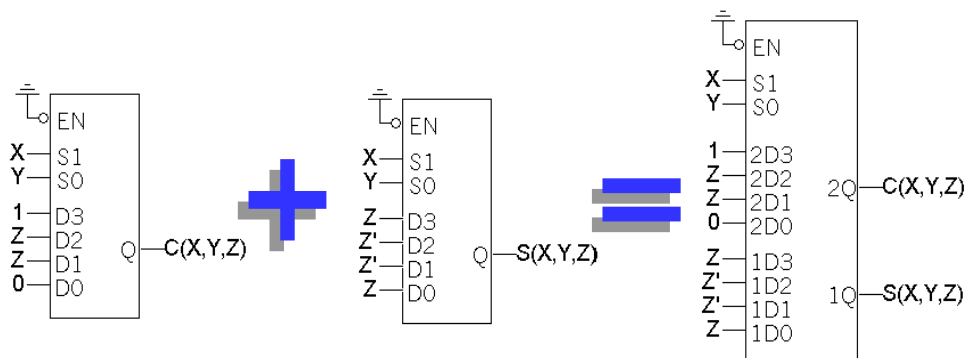
Για τη συνάρτηση του αθροίσματος $S(X,Y,Z)$, ακολουθώντας την ίδια διαδικασία, έχουμε:

$S = X' Y' D_0 + X' Y D_1 + X Y' D_2 + X Y D_3$ και συνεπώς

$S = X' Y' Z + X' Y Z' + X Y' Z' + X Y Z = \Sigma m(1, 2, 4, 7)$

7.5.3.3 Υλοποίηση πλήρους αθροιστή με διπλό πολυπλέκτη

Μερικές φορές είναι βολικό να σκεφτόμαστε την έξοδο του αθροιστή σαν να είναι ένας απλός αριθμός των 2-bit αντί για δύο ξεχωριστές συναρτήσεις. Χρειαζόμαστε δύο χωριστούς πολυπλέκτες 4-σε-1: έναν για το C και έναν για το S (Σχήμα 7.23). Ένας διπλός πολυπλέκτης 4-σε-1 δίνει την ψευδαισθηση των 2-bit δεδομένων εισόδου και εξόδου. Είναι σαν να δουλεύουμε με δύο συνδεδεμένους πολυπλέκτες 4-σε-1.



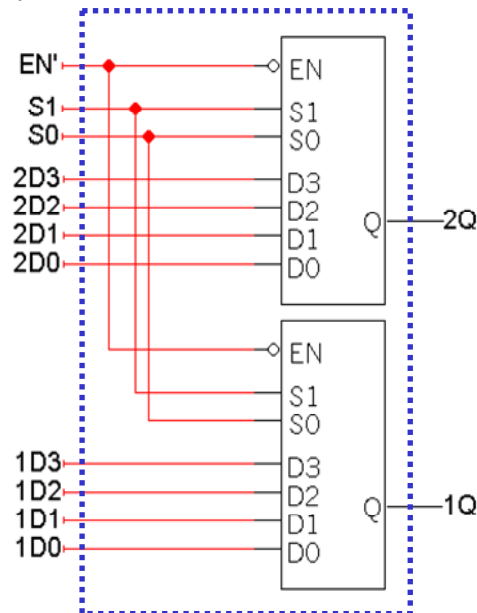
Σχήμα 7.23. Υλοποίηση πλήρους αθροιστή με διπλό πολυπλέκτη

Ανάλυση του διπλού πολυπλέκτη

Μπορούμε να φτιάξουμε ένα διπλό πολυπλέκτη 4-σε-1 συνδέοντας 2 πολυπλέκτες 4-σε-1. («Διπλός» σημαίνει «τιμές των 2-bit»)

Στο Σχήμα 7.24, που ακολουθεί, χρησιμοποιούμε τα S1-S0, για να επιλέξουμε ένα από τα παρακάτω ζεύγη εισόδων.

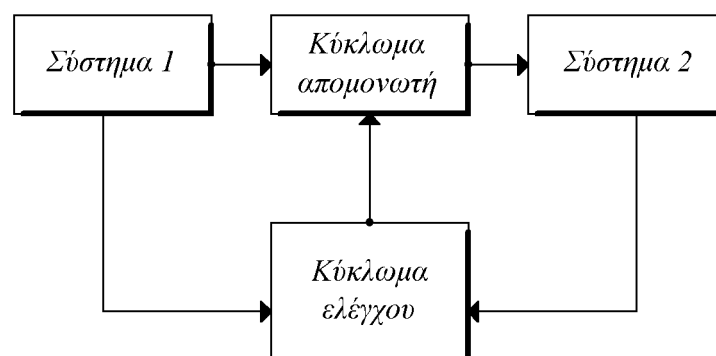
Έτσι μπορούμε να δούμε πώς ένας πολυπλέκτης 8-δρόμων (k-σε-1) μπορεί να χρησιμοποιηθεί για την επιλογή από ένα σύνολο από (k) αριθμούς των 8-bit.



Σχήμα 7.24 Αναλυτικό διάγραμμα του διπλού πολυπλέκτη

7.6. Απομονωτές (Buffers)

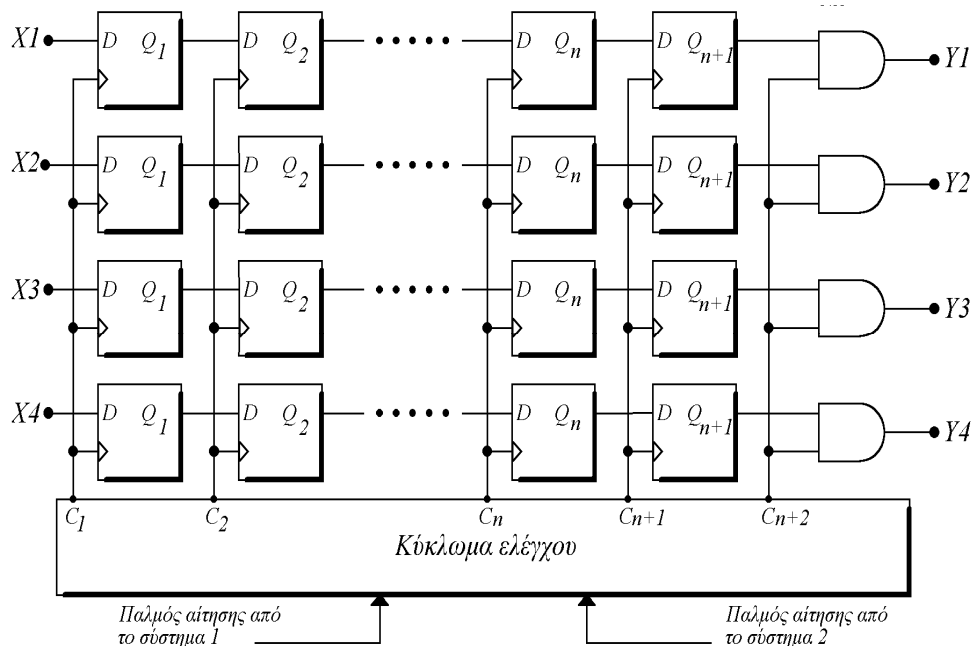
Οι απομονωτές είναι κυκλώματα προσαρμογής μεταξύ δύο συστημάτων που εργάζονται με διαφορετικές ταχύτητες. Όταν η ταχύτητα με την οποία εμφανίζονται πληροφορίες στην έξοδο ενός συστήματος είναι μεγαλύτερη από την ταχύτητα επεξεργασίας ενός δεύτερου συστήματος, τότε, για να υπάρχει σωστή λειτουργία του συνολικού συστήματος, τοποθετείται ένα κύκλωμα απομονωτή, του οποίου το διάγραμμα βαθμίδων φαίνεται στο Σχήμα 7.25. (Κοσσίδης, 1996; Δεληγιάννης, 2004; Mano, & Ciletti, 2014)



Σχήμα 7.25. Γενική συνδεσμολογία απομονωτή

Η μέση ταχύτητα λειτουργίας του συστήματος 2 πρέπει να είναι ίση, ή μεγαλύτερη, αυτής του συστήματος 1. Στην πραγματικότητα ένας απομονωτής είναι μια μνήμη, στην οποία καταγράφονται οι εξερχόμενες πληροφορίες από το σύστημα 1, για να εισέλθουν αργότερα για επεξεργασία στο σύστημα 2, όταν το τελευταίο θα τις ζητήσει στέλνοντας κατάλληλο σήμα μέσω του κυκλώματος ελέγχου

Στο Σχήμα 7.26 δίδεται το κύκλωμα ενός απομονωτή με D f-f. Ο απομονωτής αυτός αποτελείται από ένα δυσδιάστατο ολισθητή/ καταχωρητή, με δυνατότητα αποθήκευσης n λέξεων των 4 bits κάθε μία. Η μονάδα ελέγχου κανονίζει τη μετακίνηση των πληροφοριών μέσα στον απομονωτή. Η αρχή που χρησιμοποιείται, κατά κανόνα, σ' αυτούς τους απομονωτές είναι η FIFO (First In First Out), δηλαδή η πρώτη πληροφορία που μπαίνει είναι και αυτή που θα εξέλθει πρώτη.



Σχήμα 7.26. Κύκλωμα απομονωτή

7.7. MODEM

Το MODEM (**M**odulator-**D**emodulator) αποτελείται από κυκλώματα τα οποία χρησιμοποιούνται για τη διαμόρφωση-αποδιαμόρφωση πληροφοριών οι οποίες ανταλλάσσονται μεταξύ ηλεκτρονικών υπολογιστών ή τερματικών και Η/Υ που βρίσκονται σε μεγάλη απόσταση μεταξύ τους, π.χ. σε δύο διαφορετικές πόλεις. Η επικοινωνία μεταξύ των ηλεκτρονικών υπολογιστών γίνεται μέσω τηλεφωνικών γραμμών, στις οποίες έχουν συνδεθεί κατάλληλες διατάξεις MODEM. Τα MODEM μετατρέπουν ψηφιακά σήματα σε αναλογικά σε συχνότητες εντός του εύρους των τηλεφωνικών γραμμών 300 – 3400 Hz.

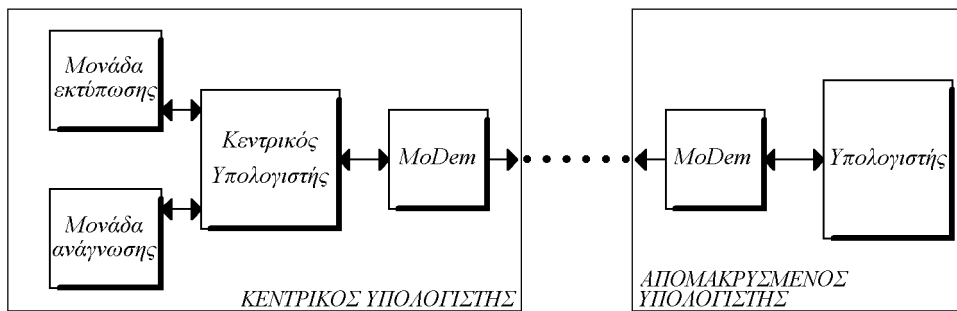
Η παρουσία των MODEM έχει αποβεί καθοριστική για τις επικοινωνίες μεταξύ των υπολογιστών, ιδίως στη σημερινή εποχή, στην οποία τα παγκόσμια δίκτυα πληροφοριών είναι προσιτά για οποιονδήποτε χρήστη ηλεκτρονικών υπολογιστών.

Ο λόγος για τον οποίο είναι αδύνατη η μεταφορά ψηφιακών δεδομένων χωρίς διαμόρφωση σε μεγάλες αποστάσεις είναι ότι η γραμμή που συνδέει τους δύο ηλεκτρονικούς υπολογιστές δρα ως φίλτρο διέλευσης στην περιοχή 300 - 3400 Hz, με αποτέλεσμα την παραμόρφωση των ψηφιακών σημάτων που είναι τετραγωνικοί παλμοί.

Στο Σχήμα 7.27 δίνεται ο τρόπος σύνδεσης μιας απομακρυσμένης τερματικής μονάδας με ηλεκτρονικό υπολογιστή μέσω MODEM.

Θεωρητικά όλα τα γνωστά κανάλια επικοινωνίας μπορούν να χρησιμοποιηθούν για τη μετάδοση ψηφιακών δεδομένων. Κάθε τύπος όμως καναλιού έχει ιδιαίτερα χαρακτηριστικά μετάδοσης, όπως, π.χ., χωρητικότητα πληροφορίας (information capacity), ρυθμό σφάλματος (error rate), κ.λπ. Δε θα αναφερθούμε εδώ λεπτομερώς στα προβλήματα που παρουσιάζονται στα κανάλια μετάδοσης δεδομένων, διότι αυτά είναι πολλά και πολυσύνθετα, και η εξήγησή τους ξεφεύγει από την ύλη του μαθήματος. Αναφέρουμε όμως ότι αυτά είναι σε θέση να επηρεάζουν, κατά πολύ, την αξιοπιστία μετάδοσης του μέσου μεταφοράς καθώς και την ταχύτητα μεταφοράς των πληροφοριών.

Μεγαλύτερες ταχύτητες επιτυγχάνονται με ιδιωτικές γραμμές (private ή leased lines) και μικρότερες με γραμμές επιλογής (dialed lines). (Κοσσιδάς, 1996; Mano, & Ciletti, 2014)



Σχήμα 7.27. Γενική παράσταση επικοινωνίας σταθμού τηλεπροσπέλασης με Η/Υ

7.7.1. Διαμόρφωση

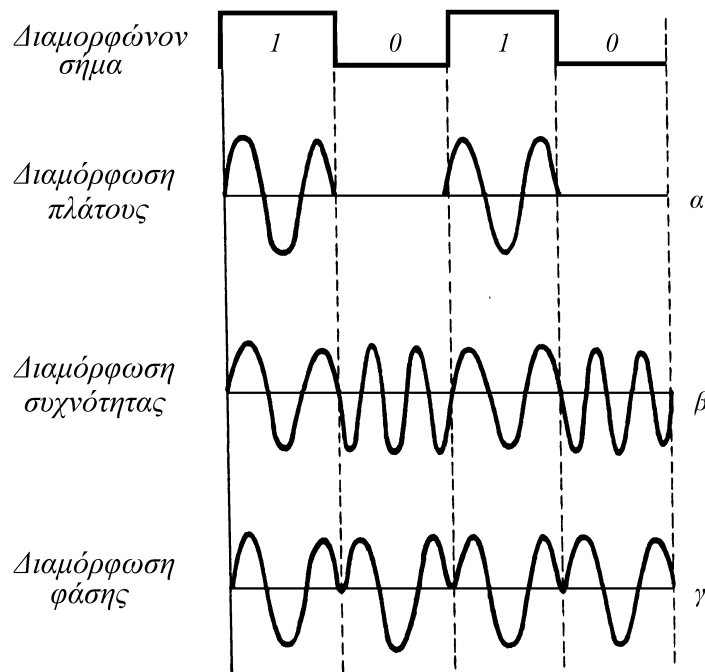
Η διαμόρφωση είναι απαραίτητη για τη μετατροπή των δεδομένων σε μορφή κατάλληλη για μετάδοση. Η διαμόρφωση ουσιαστικά μεταφέρει την ενέργεια των δεδομένων σε μια κατάλληλη ζώνη συχνοτήτων με την καλύτερη δυνατή μέθοδο.

Η μέθοδος διαμορφώσεως εκλέγεται, συνήθως, με βάση τα χαρακτηριστικά του μέσου μετάδοσης και του λαμβανόμενου σφάλματος, και όχι με βάση τα πληροφοριακά χαρακτηριστικά του συστήματος. Οι υπάρχοντες τρόποι διαμόρφωσης είναι τρεις: α) Διαμόρφωση πλάτους, β) Διαμόρφωση συχνότητας και γ) Διαμόρφωση φάσης

Κατά τη μετάδοση ψηφιακών δεδομένων μέσω τηλεφωνικού δικτύου είναι απαραίτητη η μετατροπή τους σε τόνους φωνής. Οι τόνοι αυτοί είναι, συνήθως, αρμονικές φέρουσες, όπως φαίνεται στο Σχήμα 7.28.

Στη διαμόρφωση πλάτους, τα στοιχεία πληροφορίας «1» και «0» παριστάνονται με διαμόρφωση 100%, δηλαδή πλήρη συμπίεση του φέροντος, και με διαμόρφωση 0%, δηλαδή χωρίς καμία αλλοίωση του φέροντος, αντίστοιχα (Σχήμα 7.28α).

Στη διαμόρφωση συχνότητας, τα στοιχεία πληροφορίας «0» και «1» παριστάνονται με δύο διαφορετικές συχνότητες f_1 και f_2 αντίστοιχα (Σχήμα 7.28β). (Κοσσίδης, 1996, Γιαννακόπουλος, 2012)



Σχήμα 7.28. Οι τρεις μέθοδοι διαμόρφωσης

Στη διαμόρφωση φάσης, η φάση της φέρουσας αλλάζει κατά 180° , όταν η διαμορφούσα ψηφιακή πληροφορία αλλάζει από λογικό «1» σε «0» (Σχήμα 7.28γ). Η εκλογή της φέρουσας (φορέα της πληροφορίας) πραγματοποιείται με τέτοιο τρόπο, ώστε να εξασφαλίζει μικρή εξασθένηση και αμελητέα παραμόρφωση του σήματος. Γι' αυτόν το λόγο πρέπει να εμπεριέχεται μέσα στα όρια του εύρους ζώνης συχνοτήτων του καναλιού, μέσω του οποίου γίνεται η μετάδοση. Η διαμόρφωση του φορέα επιτυγχάνεται από την ψηφιακή πληροφορία, που αποστέλλεται.

Η διαμόρφωση πλάτους δε χρησιμοποιείται συχνά, γιατί είναι ευαίσθητη στο θόρυβο και είναι δύσκολη η αποδιαμόρφωση, ειδικά όταν το σήμα είναι χαμηλό και υπάρχει θόρυβος.

Στη διαμόρφωση συχνότητας, η συχνότητα της φέρουσας μεταβάλλεται μεταξύ δύο τιμών ανάλογα με το διαμορφώνον ψηφιακό σήμα. Η αποδιαμόρφωση γίνεται απλά, ανιχνεύοντας την παρουσία της μιας ή της άλλης συχνότητας. Χρησιμοποιείται ευρέως στα περισσότερα MODEM χαμηλής ταχύτητας.

Τέλος η διαμόρφωση φάσης, που χωρίζεται σε επί μέρους κατηγορίες η ανάλυση των οποίων δεν περιλαμβάνεται στην ύλη του βιβλίου, χρησιμοποιείται πολύ στη σύγχρονη σειριακή επικοινωνία.

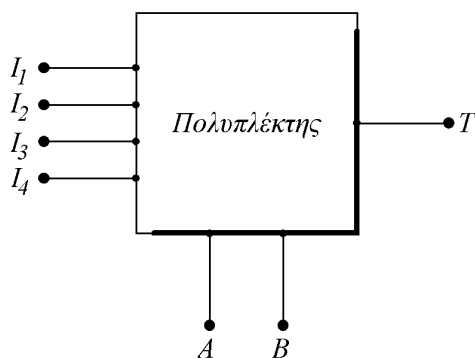
7.8. Ασκήσεις – Ερωτήσεις

ΕΡΩΤΗΣΕΙΣ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

1. Πόσες γραμμές επιλογής θα απαιτούνταν για έναν πολυπλέκτη 8-σε-1.
Α. 2 Β. 3 Γ. 4 Δ. 8
2. Ποιο τύπο μετατροπής εξυπηρετούν οι περισσότεροι αποπολυπλέκτες;
Α. Δεκαδικό σε δεκαεξαδικό Β. Μια είσοδο, πολλές εξόδους
Γ. AC σε DC Δ. Περιττή ισοτιμία σε άρτια ισοτιμία
3. Οι εισοδοί/έξοδοι ενός αναλογικού πολυπλέκτη/αποπολυπλέκτη είναι:
Α. Αμφίδρομοι Β. Μονής κατεύθυνσης
Γ. Άρτιας ισοτιμίας Δ. BCD
4. Μία εφαρμογή ενός ψηφιακού πολυπλέκτη είναι, να διευκολύνει:
Α. Την παραγωγή ψηφιακών δεδομένων
Β. Τη μετατροπή από σειριακή σε παράλληλη
Γ. Τον έλεγχο ισοτιμίας
Δ. Τον επιλογή δεδομένων

ΑΣΚΗΣΕΙΣ

1. Δίδεται ο πολυπλέκτης του παρακάτω σχήματος:



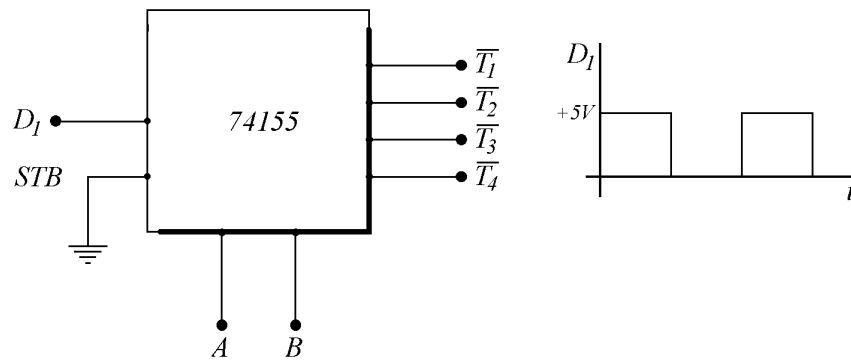
Να υπολογισθεί η έξοδος T. Επίσης να βρεθεί ποιές τιμές πρέπει να έχουν οι εισοδοί I_1, I_2, I_3, I_4 ώστε να έχουμε αντίστοιχα έξοδο:

- α) $T=A$
- β) $T=$ και
- γ) $T=A \odot B$

2. Ποια είναι η τιμή της τάσης εξόδου βαθμωτού κυκλώματος αντιστάσεων 6 εισόδων (βλέπε Σχήμα 7.2) με τάση αναφοράς 16 Volt (λογικό «1»), για τις παρακάτω τιμές σημάτων στις εισόδους;
α) 101101 β) 111111 γ) 000111
3. Από τι εξαρτάται η ακρίβεια μετατροπής των πληροφοριών από A/D και από D/A;
4. Πόσος χρόνος απαιτείται για τη μετατροπή αναλογικής πληροφορίας σε ψηφιακή, όταν η συχνότητα χρονισμού είναι 600 KHz και διαθέτουμε SAR των 12 δυαδικών ψηφίων;
5. Να σχεδιαστεί πολυπλέκτης με τέσσερις εισόδους και μία έξοδο έτσι, ώστε τα σήματα επιλογής AB (00, 01, 10, 11) να ενεργοποιούν τις αντίστοιχες πύλες, για να είναι η έξοδος του ίση με I_1, I_2, I_3, I_4 .
6. Να αναφέρετε σε ποιες περιπτώσεις πλεονεκτεί η διαμόρφωση συχνότητας και γιατί.
7. Να αποδείξετε ότι η έξοδος T του παρακάτω πίνακα αληθείας είναι ίδια, ταυτόσημη, με την έξοδο του πολυπλέκτη 4/1.

A	B	T
0	0	I_1
0	1	I_2
1	0	I_3
1	1	I_4

8. Δίνεται το παρακάτω κύκλωμα αποπλέκτη. Αν η κυματομορφή εισόδου D_1 έχει την παραπλευρώως μορφή, να σχεδιαστούν οι κυματομορφές εξόδου, $\overline{T_1}, \overline{T_2}, \overline{T_3}, \overline{T_4}$, όταν $A \cdot B = 11$



Σημείωση: Το ολοκληρωμένο κύκλωμα 74155 είναι ένας αποπλέκτης 1 προς 4 γραμμών και ελέγχει ανάλογα με την κατάσταση των A, B τη μεταφορά του σήματος εισόδου D_1 σε μια από τις εξόδους, $\overline{T_1}, \overline{T_2}, \overline{T_3}, \overline{T_4}$. Όταν το STB (strobe) είναι γειωμένο, ο αποπλέκτης είναι ενεργοποιημένος.

Αναφορές-Βιβλιογραφία

- Γιαννακόπουλος Π., (2012) *Ψηφιακές Επικοινωνίες*, Σύγχρονη Εκδοτική
- Δεληγιάννης†, Θ. (2004). *Ηλεκτρονικά Αναλογικά και Ψηφιακά*, Ίδρυμα Ευγενίδου
- Κοσσίδης, Α.Θ. (1996). *Σχεδίαση Ψηφιακών Κυκλωμάτων*, Εκδόσεις Μπένος
- Κοσσίδης, Α.Θ., Γιαννακόπουλος, Π., (2006), *Αριθμητικά Συστήματα και Ψηφιακά Κυκλώματα*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα
- Balabanian, N., Carlson, B. (2007). *Digital Logic Design Principles*, John Wiley
- Givone, D. (2002). *Digital Principles and Design*,. Mc Graw Hill
- Godse, A.P., Godse, D.A. (2010). *Digital Logic Design and Application*, Technical Publications Pune
- Holdsworth, Brian, Woods, Clive (2002). *Digital Logic Design*, 4th Edition, Newnes
- Jain R.P. (2010) *Modern Digital Electronics*, Tata McGraw-Hill Education
- Katz, R. (2005). *Contemporary Logic Design*, 2/e, Prentice Hall
- Malvino A.P., Brown P., (1993) *Digital Computer electronics*, Tata McGraw Hill
- Mano, M., Ciletti, M. (2014). *Ψηφιακή Σχεδίαση*, 5^η έκδοση, Παπασωτηρίου
- Maxfield Clive (2009). *Bebop to the Boolean Boogie, An Unconventional Guide to Electronics*, 3rd, Newnes, Elsevier <http://english.360elib.com/datu/T/EM140371.pdf>
- Nashelsky, Louis (1994). *Introduction to Digital Technology*, 4th Ed., Prentice Hall
- Predko, Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill
- Pritchard, N. (2015). *Fundamentals of Digital Electronics*, CreateSpace Independent Publishing Platform
- Roth, Charles Jr., Kinney, L.L., (2014). *Fundamentals of Logic Design*, 7th, Cengage Learning
- Sedra, A., Smith, Kenneth. (2015). *Microelectronic Circuits*, 7th edition, Oxford University Press
- Taub H., and Schilling D. L. (1983) *Digital Integrated Electronics*, McGraw-Hill
- Tocci, R. J., Widmer, N. S, and Moss, Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley
- Wakerly, J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Κεφάλαιο 8

Σύνοψη

Λογικές συναρτήσεις μπορούν να υλοποιηθούν με τη μορφή αθροισμάτων γινομένων, δηλαδή από ένα σύνολο προγραμματιζόμενων πυλών AND, που τροφοδοτούν ένα σύνολο από προγραμματιζόμενες πύλες OR. (PLA). Επίσης μπορούν να υλοποιηθούν από ένα σύνολο πυλών AND, που τροφοδοτούν ένα σύνολο από προγραμματιζόμενες πύλες OR (PROM) και, τέλος, από ένα σύνολο προγραμματιζόμενων πυλών AND, που τροφοδοτούν ένα σύνολο από πύλες OR (PAL). Στο κεφάλαιο αυτό θα μελετήσουμε την υλοποίηση λογικών συναρτήσεων με τις παραπάνω δυνατότητες, που δίνουν οι προγραμματιζόμενες λογικές διατάξεις.

Προαπαιτούμενη γνώση

Η γνώση των βασικών πυλών και η απλοποίηση συναρτήσεων με τη χρήση του χάρτη Karnaugh.

8. Προγραμματιζόμενες λογικές διατάξεις

8.1. Εισαγωγή

Ένα λογικό κύκλωμα αποτελείται, κατά κανόνα, από πάρα πολλές πύλες, f - f , καταχωρητές και άλλα ψηφιακά εξαρτήματα. Η πραγματοποίησή του απαιτεί, επομένως, μεγάλες επιφάνειες τυπωμένων πλακετών, που, με τη σειρά τους, επαυξάνουν τον όγκο της συνολικής κατασκευής. Είναι, επίσης, πολύ πιθανό, τα ολοκληρωμένα που θα χρησιμοποιηθούν για τη συνολική σχεδίαση να μην αξιοποιούνται πλήρως (να απαιτούνται μόνο οι τρεις από τις τέσσερις πύλες π.χ. ενός 74LS00), οπότε εκτός των άλλων θα υφίσταται μια αυξημένη κατανάλωση ρεύματος.

Όλα τα παραπάνω προβλήματα έρχονται να λύσουν τα νέα ολοκληρωμένα κυκλώματα προγραμματιζόμενης λογικής, το εσωτερικό των οποίων μπορεί να τροποποιηθεί με τέτοιο τρόπο, ώστε να ταιριάζει στις ανάγκες μιας συγκεκριμένης εφαρμογής.

Τα εξαρτήματα προγραμματιζόμενης λογικής αποτελούνται ουσιαστικά από λογικές πύλες το πλήθος των οποίων, όπως και ο αριθμός των εισόδων τους, εξαρτάται από τον τύπο του ολοκληρωμένου που τα χαρακτηρίζει. Δεν είναι όμως το πλήθος των πυλών αυτό που δημιουργεί την καινοτομία στη νέα αυτή μορφή ολοκληρωμένων κυκλωμάτων αλλά ο τρόπος με τον οποίο αυτές είναι συνδεδεμένες μεταξύ τους. Η εσωτερική δομή αυτών των διατάξεων βασίζεται σε συνδέσεις που πραγματοποιούνται μέσω τηκόμενων ασφαλειών (fuse programmable), και οι οποίες υφίστανται μεταξύ πυλών AND, πυλών OR και καταχωρητών. Με τη βοήθεια ενός προγραμματιζόμενου πίνακα συνδέσεων, ο οποίος μπορεί να μορφοποιηθεί είτε μέσω μάσκας (mask programming) για μεγάλες παραγωγές είτε μέσω μιας ειδικής συσκευής για μικρές, το ολοκληρωμένο κύκλωμα μπορεί να αποκτήσει μια μοναδική ταυτότητα. Γίνεται, επομένως, σαφές ότι ο χρήστης των εξαρτημάτων αυτών έχει τον τελευταίο λόγο για την τελική μορφή τους. Ο βασικός προγραμματιζόμενος πίνακας είναι λογικής AND - OR στη γνωστή αναπαράσταση αθροίσματος γινομένων SOP (Sum of Products). (Kossidas, 1996; Givone, 2002; Holdsworth, & Woods, 2002; Katz, 2005; Balabanian, & Carlson 2007; Godse, & Godse, 2010; Mano, & Ciletti, 2014; Roth, & Kinney, 2014)

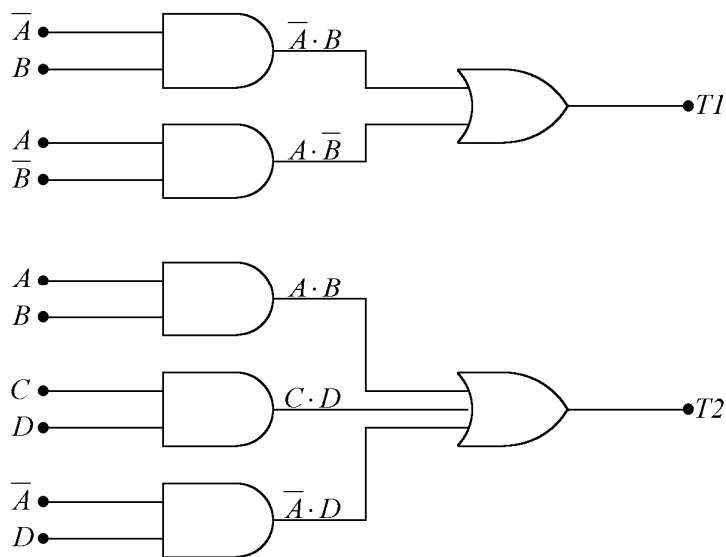
8.2. Αναπαράσταση κυκλωμάτων προγραμματιζόμενης λογικής

Οι χάρτες Karnaugh, όπως και οι αλγεβρικές μέθοδοι εξαγωγής και απλοποίησης συναρτήσεων, μας δίνουν, κατά κανόνα, σαν τελικό προϊόν, μια παράσταση η οποία αποτελείται από ένα «άθροισμα γινομένων», το οποίο υλοποιείται εύκολα με τις βασικές λογικές πύλες. Στο Σχήμα 8.1 βλέπουμε δύο κυκλώματα, τα οποία εκφράζουν τις συναρτήσεις: $T_1 = A \cdot \bar{B} + \bar{A} \cdot B = A \oplus B$ και $T_2 = A \cdot B + C \cdot D + \bar{A} \cdot D$

Διαπιστώνουμε πως και στα δύο αυτά κυκλώματα υπάρχουν κοινές μεταβλητές, γεγονός που μας ωθεί να πιστέψουμε πως τα κυκλώματα αυτά διεγείρονται από τα ίδια σήματα. Άρα μπορούν κατ' επέκταση να υλοποιηθούν στο εσωτερικό του ίδιου ολοκληρωμένου κυκλώματος προγραμματιζόμενης λογικής. Στο Σχήμα 8.2 φαίνεται μ' ένα διαφορετικό τρόπο το πώς υλοποιείται αυτή η σχεδίαση σ' ένα μοναδικό προγραμματιζόμενο ολοκληρωμένο κύκλωμα.

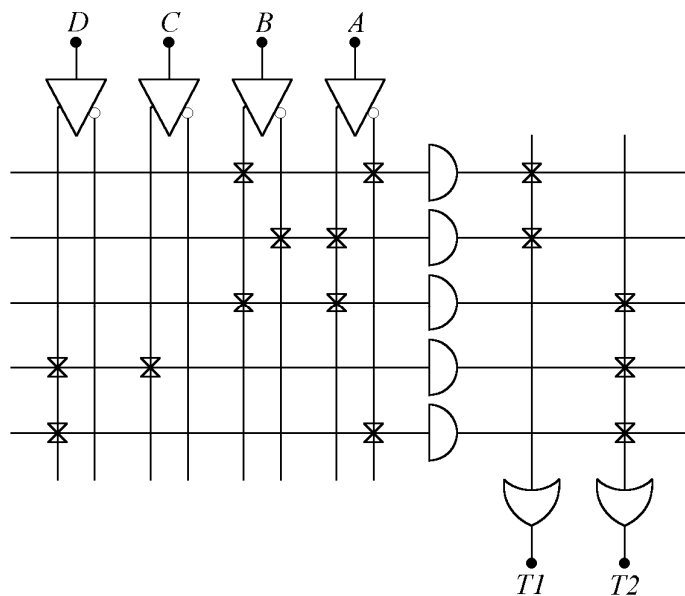
Οι μεταβλητές εισόδου **A**, **B**, **C**, **D** οδηγούνται σε τέσσερις βαθμίδες απομονωτών/αναστροφέων οι οποίες διαθέτουν δύο εξόδους η κάθε μία. Η πρώτη από αυτές αναδεικνύει μια λογική κατάσταση ίδια μ' αυτή της εισόδου, ενώ η δεύτερη δημιουργεί την ακριβώς αντίθετη. Μ' αυτό τον τρόπο παράγονται 8 διαφορε-

τικά σήματα, εκ των οποίων τα τέσσερα αντιπροσωπεύουν τις αρχικές μεταβλητές **A**, **B**, **C**, **D** ενώ τα υπόλοιπα τέσσερα τις συμπληρωματικές τους, δηλαδή τις \bar{A} , \bar{B} , \bar{C} , και \bar{D} .



Σχήμα 8.1. Συμβατική παρουσίαση λογικού κυκλώματος

Οι οκτώ διαφορετικές εξόδους των βαθμίδων εισόδου διασταυρώνονται, χωρίς να συνδέονται υποχρεωτικά, με οριζόντιες γραμμές, σχηματίζοντας έτσι έναν προγραμματιζόμενο πίνακα. Οι οριζόντιες γραμμές καταλήγουν σε πέντε πύλες AND. Παρατηρήστε ότι οι πύλες αυτές φαίνεται να έχουν μόνο μια είσοδο η καθεμία, αλλά στην πραγματικότητα δεν συμβαίνει αυτό. Κάθε μια πύλη AND έχει τόσες εισόδους, όσα και τα σήματα που καταλήγουν σ' αυτή. Στο συγκεκριμένο σχήμα, η πρώτη πύλη AND δέχεται τα σήματα \bar{A} και **B**, ενώ η δεύτερη τα σήματα **A** και **B**. Και οι δύο έχουν δηλαδή δύο εισόδους.



Σχήμα 8.2. Παρουσίαση σύμφωνα με τους κανόνες της προγραμματιζόμενης λογικής

Ένας δεύτερος προγραμματιζόμενος πίνακας δημιουργείται μεταξύ των εξόδων των πυλών AND και των εισόδων των πυλών OR, που ακολουθούν. Και εδώ οι εισόδους των πυλών OR βρίσκονται μετρώντας το πλήθος των σημάτων που καταλήγουν σ' αυτές. Έτσι η πύλη που παράγει το σήμα **T1** έχει δύο εισόδους, ενώ

αυτή που παράγει το T_2 έχει τρεις. Αυτός ο τρόπος αναπαράστασης διευκολύνει σημαντικά τον σχεδιαστή, ιδίως σ' εκείνες τις περιπτώσεις στις οποίες το σύνολο των γραμμών εισόδου και εξόδου είναι μεγάλο.

Μια μεγάλη ποικιλία προϊόντων προγραμματιζόμενης λογικής κατασκευάζονται με βάση αυτή τη δομή, εμπεριέχοντας και άλλα στοιχεία λογικής όπως, καταχωρητές, πολυπλέκτες ή και προγραμματιζόμενες εξόδους, ενεργοποιούμενες σε λογικό «0» ή σε λογικό «1».

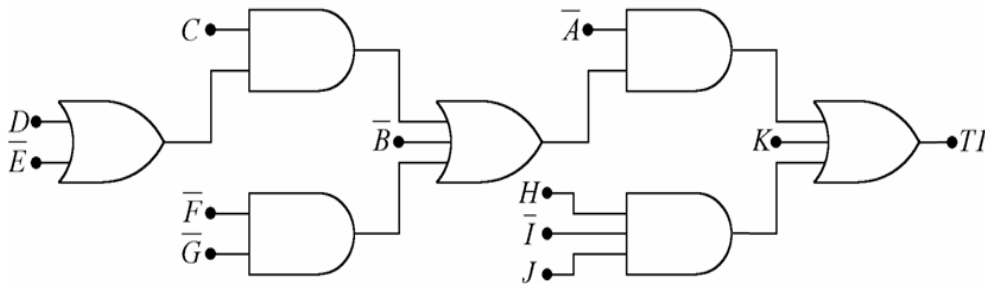
8.2.1. Πλεονεκτήματα της προγραμματιζόμενης λογικής

Τα πλεονεκτήματα, που προκύπτουν από τη χρήση προγραμματιζόμενων εξαρτημάτων είναι πολλά και σημαντικά. Ένας εύκολος τρόπος να το διαπιστώσουμε είναι μελετώντας τη λογική συνάρτηση:

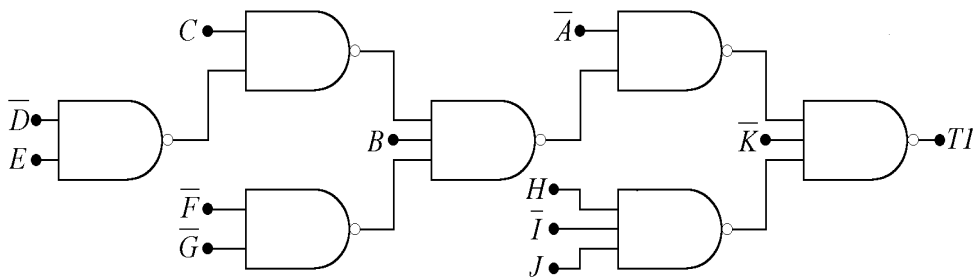
$$T_1 = \overline{A}(B + C \cdot (D + \overline{E}) + \overline{F} \cdot \overline{G}) + H\overline{I} + K$$

Αν θελήσουμε να κατασκευάσουμε το κύκλωμα που υλοποιεί τη συνάρτηση T_1 , θα πρέπει να σχεδιάσουμε το κύκλωμα του Σχήματος 8.3α, στο οποίο έχουν χρησιμοποιηθεί τέσσερα διαφορετικά είδη πυλών: OR, AND, NAND και NOR. Αν θελήσουμε να κάνουμε μια πιο οικονομική σχεδίαση, χρησιμοποιώντας μόνο πύλες NAND, καταλήγουμε στο Σχήμα 8.3β. Και τα δύο αυτά κυκλώματα επαληθεύουν τη συνάρτηση T_1 , αλλά, όπως ήδη θα έχετε παρατηρήσει, η «αλυσίδα» των πυλών είναι πολύ μεγάλη. Μεταξύ της πρώτης και της τελευταίας πύλης του καθενός σχήματος μεσολαβούν άλλες πέντε.

Στο Σχήμα 8.3γ βλέπουμε το ίδιο κύκλωμα, κατασκευασμένο με κυκλώματα πυλών AND - OR, ίδια, δηλαδή, με εκείνα που χρησιμοποιούνται στην προγραμματιζόμενη λογική. Είναι σίγουρα πολύ πιο απλό, ενώ μεταξύ των μεταβλητών εισόδου και της μεταβλητής εξόδου παρεμβάλλονται το πολύ δύο πύλες. Έχουμε, δηλαδή, ένα σημαντικό περιορισμό των επιπέδων σχεδίασης. Η παρουσία των δύο μόνο πυλών, εξασφαλίζει τους χαμηλούς χρόνους μετάδοσης ή αλλιώς την υψηλή ταχύτητα του κυκλώματος.



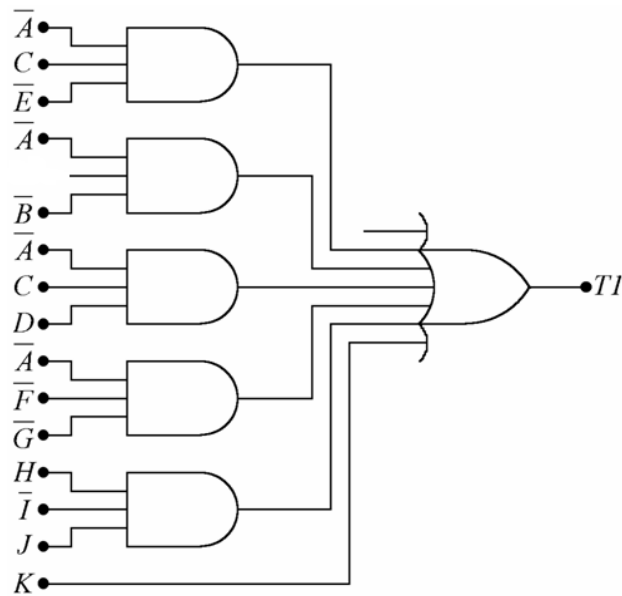
Σχήμα 8.3α. Υλοποίηση του κυκλώματος με πύλες AND και OR



Σχήμα 8.3β. Υλοποίηση του κυκλώματος α με πύλες NAND

Τα πλεονεκτήματα μπορούν σύντομα να συνοψισθούν παρακάτω:

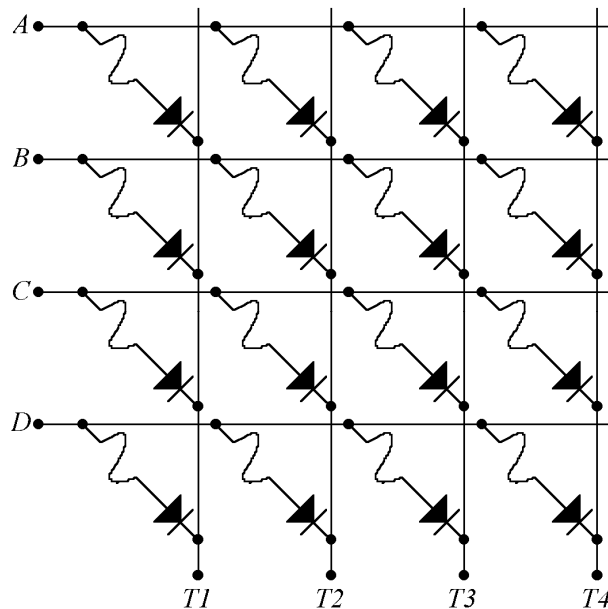
- α) Περιορισμός του πλήθους των εξαρτημάτων πάνω στο τυπωμένο κύκλωμα,
- β) Οικονομία χρόνου στη σχεδίαση συστημάτων,
- γ) Σχεδιαστική ευελιξία και προσαρμοστικότητα,
- δ) Περιορισμός της πολυεπίπεδης λογικής,
- ε) Περιορισμός κόστους, λόγω μικρότερου κόστους των εξαρτημάτων, της ανάπτυξης και της βιομηχανικής παραγωγής και της φθηνότερης συντήρησης και επισκευής.



Σχήμα 8.3γ. Υλοποίηση του κυκλώματος με τον ελάχιστο αριθμό πυλών AND και OR

8.3. Στοιχεία προγραμματιζόμενης λογικής

Το πρώτο εξάρτημα προγραμματιζόμενης λογικής σε μορφή ολοκληρωμένου κυκλώματος κατασκευάστηκε στις αρχές της δεκαετίας του '60 και περιλάμβανε μια μήτρα διόδων. Το κύριο κατασκευαστικό χαρακτηριστικό του ήταν οι μεταλλικές σειρές και στήλες που συνδέονταν στις διασταυρώσεις με τη βοήθεια διόδων και ασφαλειών από εύτηκτο αλουμίνιο. Αυτές οι ασφάλειες μπορούσαν να διακοπούν επιλεκτικά, αφήνοντας ορισμένες διασταυρώσεις ανοιχτές και ορισμένες άλλες βραχυκυκλωμένες.



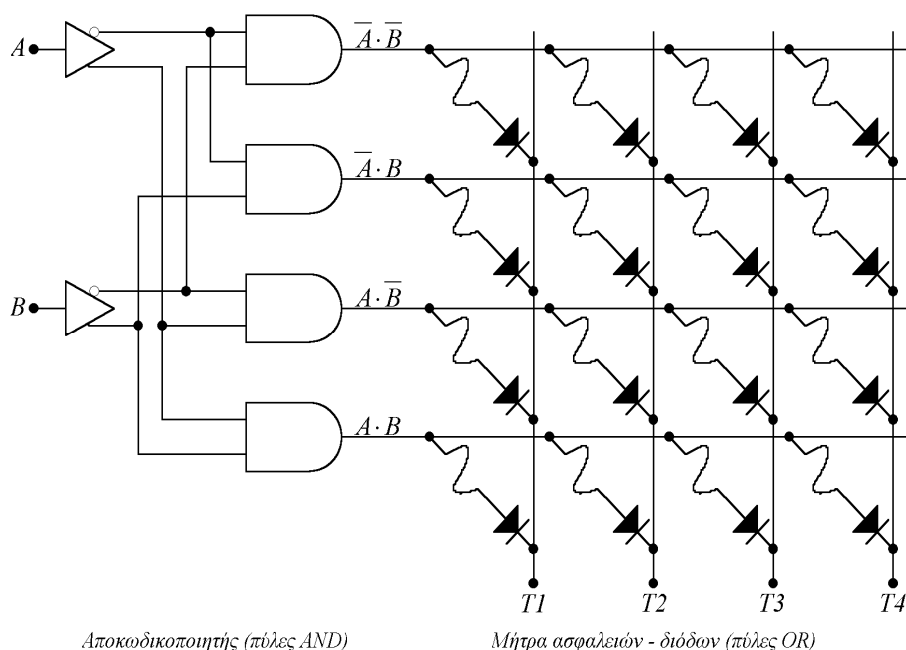
Σχήμα 8.4. Πίνακας διόδων OR

Το αποτέλεσμα ήταν η δημιουργία μιας μήτρας διόδων, από τις οποίες η κάθε μια στήλη της μπορούσε να δώσει σαν προϊόν το αποτέλεσμα της πράξης OR μεταξύ των επιλεγμένων μεταβλητών που επιβάλλονταν στις σειρές. Στο Σχήμα 8.4 βλέπουμε αυτή τη μήτρα διόδων και ασφαλειών.

8.4. Μνήμη PROM

Το επόμενο βήμα στην εξέλιξη της βασικής μήτρας διόδων ήταν η προσθήκη ενός αποκωδικοποιητή, του οποίου οι εξόδοι οδηγούσαν τις σειρές της μήτρας. Η παρουσία αυτής της βαθμίδας είχε σαν αποτέλεσμα την ελαχιστοποίηση των εισόδων του καινούριου εξαρτήματος, ενώ, ταυτόχρονα, το έκανε συμβατό με στάθμες TTL, κάτι που δεν ίσχυε για την ανεξάρτητη μήτρα διόδων. Το εξάρτημα αυτό έγινε γνωστό σαν μνήμη PROM (Programmable Read Only Memory). Το εσωτερικό του φαίνεται στο Σχήμα 8.5.

Ο αποκωδικοποιητής κατασκευάζεται από πύλες AND και αναστροφείς, ενώ η μήτρα των διόδων ισοδυναμεί με τόσες πύλες OR όσες είναι και οι εξόδοι (T_1 έως T_4) της μνήμης. Βλέπουμε, δηλαδή, ότι η μνήμη PROM αποτελεί στην πραγματικότητα ένα εξάρτημα προγραμματιζόμενης λογικής AND - OR, ίδιο μ' αυτό που περιγράψαμε στις προηγούμενες παραγράφους. (Nashelsky, 1994; Κοσσιδάς, 1996; Givone, 2002; Δεληγιάννης, 2004; Balabanian, & Carlson, 2007; Godse, & Godse, 2010; Tocci, et al., 2010; Mano, & Ciletti, 2014; Roth, & Kinney, 2014).



Σχήμα 8.5. Αποκωδικοποιητής συνδεδεμένος με πίνακα διόδων OR

Η χρήση της μνήμης PROM εντοπίζεται στην αποθήκευση ενός συγκεκριμένου αριθμού δυαδικών πληροφοριών (bit), των οποίων το πλήθος ποικίλλει, ανάλογα με το μέγεθος της μνήμης. Οι μνήμες «προγραμματίζονται» είτε από το εργοστάσιο σύμφωνα με τις πληροφορίες που εμείς θέλουμε να εισάγουμε σ' αυτές είτε με τη βοήθεια ειδικών συσκευών από τον ίδιο τον σχεδιαστή της εφαρμογής. Η παραμονή των πληροφοριών μέσα σ' αυτές σε μόνιμη βάση εξασφαλίζεται με το επιλεκτικό «κάψιμο» ορισμένων ασφαλειών από αυτές που σχηματίζουν τη μήτρα των διόδων.

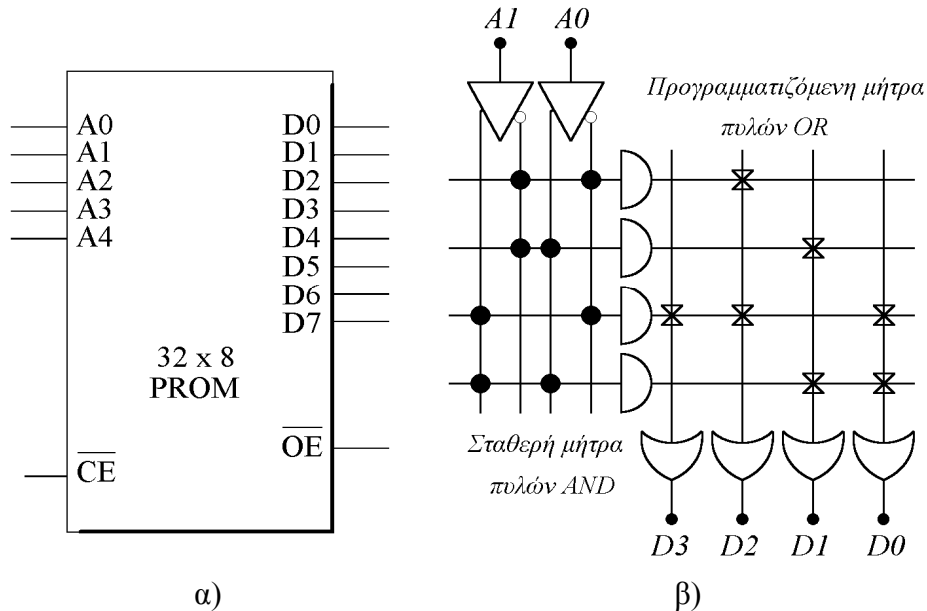
Μια PROM χαρακτηρίζεται τόσο από τον αριθμό των λέξεων, που μπορεί να αποθηκεύσει, όσο και από τον αριθμό των εξόδων της: π.χ. 32 x 8 είναι μια PROM με 32 λέξεις και 8 εξόδους. Για να έχουμε 32 λέξεις, χρειάζονται 5 εισόδοι ($32=2^5$). Στο Σχήμα 8.6α σημειώνεται μια τέτοια μνήμη. Με A_0 έως A_4 συμβολίζουμε τις εισόδους και με D_0 έως D_7 τις εξόδους. Οι υπόλοιποι δύο ακροδέκτες δέχονται σήματα ελέγχου.

Στο Σχήμα 8.6β βλέπουμε το εσωτερικό μια μνήμης PROM 4 λέξεων ($2^2=4$) και 4 εξόδων, όπως απεικονίζεται με σύμβολα προγραμματιζόμενης λογικής. Μπορούμε να θεωρήσουμε ότι η μνήμη αυτή σχηματίζεται από μια μήτρα πυλών AND, που είναι εκ των προτέρων κατασκευασμένη (σταθερή), και μια μήτρα πυλών OR, που μπορεί να μορφοποιηθεί σύμφωνα με τις ανάγκες της εφαρμογής.

Το μεγαλύτερο πλεονέκτημα, που εξασφαλίζει η χρήση των μνημών αυτού του τύπου προέρχεται από τη μαζική εργοστασιακή παραγωγή τους. Επειδή παράγονται σε πολύ μεγάλες ποσότητες, έχουν πολύ μικρό κόστος, που τις κάνει ιδιαίτερα προσιτές και προτιμότερες στις εφαρμογές. Το γεγονός εξάλλου ότι ο ενσωματωμένος αποκωδικοποιητής έχει τόσες εξόδους όσες είναι και το σύνολο των συνδυασμών των μεταβλη-

τών εισόδου, κάνει δυνατή την υλοποίηση οποιασδήποτε λογικής συνάρτησης ζητηθεί, χωρίς κανέναν απολύτως περιορισμό.

Το μειονέκτημά τους εντοπίζεται στον ενεργό διπλασιασμό των κυκλωμάτων τους στην περίπτωση αύξησης των γραμμών εισόδου τους. Έτσι, αν, π.χ., αντί για τέσσερις μεταβλητές εισόδου θέλουμε να έχουμε πέντε, η επιφάνεια του ημιαγωγού υλικού του ολοκληρωμένου κυκλώματος πρέπει να διπλασιασθεί. Αυτό οφείλεται στο ότι ο εσωτερικός αποκωδικοποιητής θα πρέπει να αποκτήσει 32 εξόδους αντί για 16, που θα είχε, αν δεχόταν τέσσερις μεταβλητές, οδηγώντας φυσικά τον ίδιο αριθμό γραμμών και στη μήτρα διόδων.



Σχήμα 8.6. Μνήμη PROM

8.4.1. Παραδείγματα χρήσης μνημών PROM

Οι PROM έχουν σπουδαίες εφαρμογές στο σχεδιασμό των διαφόρων λογικών κυκλωμάτων. Στη συνέχεια θα δώσουμε μερικά παράδειγμα πραγματοποίησης λογικών συναρτήσεων, στα οποία γίνεται φανερή η χρήση αυτών των εξαρτημάτων.

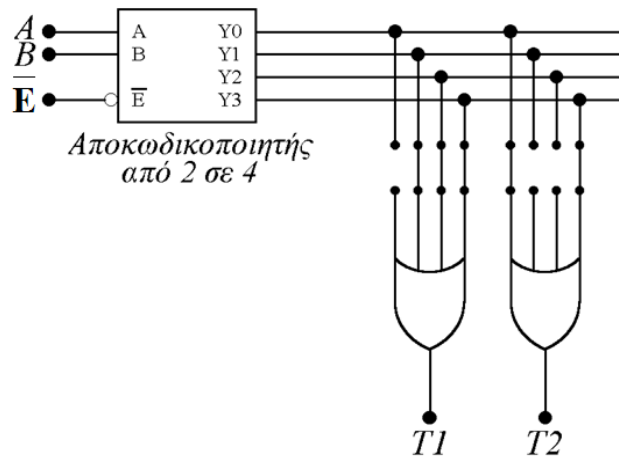
8.4.1.1 Σχεδίαση πυλών OR, XNOR

Έστω ότι έχουν δοθεί οι συναρτήσεις: $T_1(A, B) = \Sigma(1, 2, 3)$ και η $T_2(A, B) = \Sigma(0, 3)$. Ακολουθεί ο πίνακας αλήθειας που τις περιγράφει και δείχνει πότε οι εξοδοι των δύο κυκλωμάτων θα είναι ίσες με λογικό «1»:

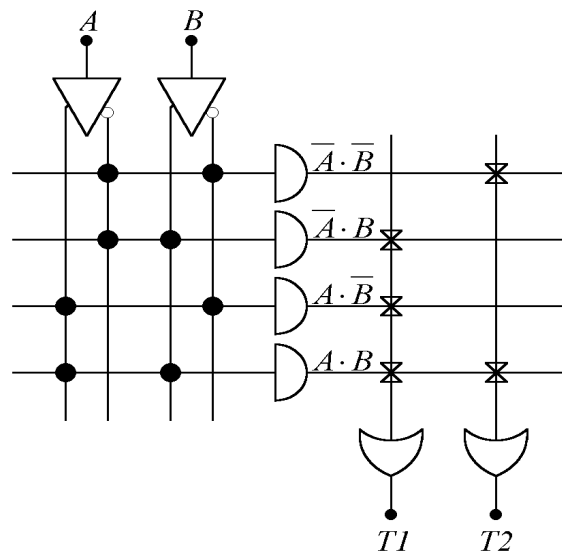
A	B	T ₁	T ₂
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	1

Πίνακας 8.1. Πίνακας Συναρτήσεων

Χρησιμοποιώντας τις βασικές δομές AND – OR, είναι εύκολο να ξεχωρίσουμε τις εξόδους των πυλών AND (έξοδοι αποκωδικοποιητή) που οδηγούνται σε λογικό «1» για τους συγκεκριμένους συνδυασμούς των εισόδων. Στη συνέχεια οδηγούμε αυτές τις εξόδους στις εισόδους δύο πυλών OR, που με τη σειρά τους παράγουν τα επιθυμητά αποτελέσματα, όπως φαίνεται στο Σχήμα 8.7. Οι εισοδοι A και B του αποκωδικοποιητή δέχονται την πληροφορία εισόδου, ενώ οι εξοδοι των πυλών αναδεικνύουν το αποτέλεσμα. Η είσοδος του αποκωδικοποιητή πρέπει να βρίσκεται σε λογικό «0», για να μπορέσει το κύκλωμα να λειτουργήσει.



Σχήμα 8.7. Λειτουργική μορφή συνδυαστικών κυκλωμάτων δομής AND – OR



Σχήμα 8.8. Υλοποίηση συναρτήσεων με PROM

Η PROM, που είναι σε θέση να αντικαταστήσει το παραπάνω κύκλωμα, θα πρέπει να είναι των 4 λέξεων και των 2 εξόδων. Στο Σχήμα 8.8 φαίνεται το εσωτερικό μιας τέτοιας μνήμης, σχεδιασμένο σύμφωνα με τους συμβολισμούς της προγραμματιζόμενης λογικής.

8.4.1.2 Σχεδίαση πλήρους αθροιστή με PROM

Είναι γνωστό ότι ένας πλήρης αθροιστής αποτελείται από ένα κύκλωμα παραγωγής ψηφίου αθροίσματος (πύλη XOR) και ένα κύκλωμα παραγωγής ψηφίου κρατουμένου (πύλη NAND). Οι συναρτήσεις που περιγράφουν τον αθροιστή είναι οι:

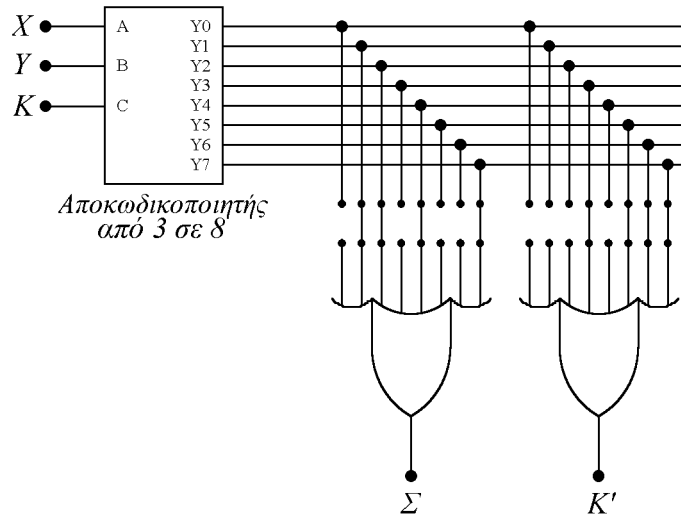
$$\Sigma(X, Y, K) = \Sigma(1, 2, 4, 7) \text{ και } K'(X, Y, K) = \Sigma(3, 5, 6, 7)$$

Είναι προφανές πως θα χρειασθούμε κύκλωμα με τρεις εισόδους, δηλαδή κύκλωμα που θα βασίζεται σε αποκωδικοποιητή από 3 σε 8. Διαλέγοντας, όπως προηγουμένως, τις εξόδους εκείνες του αποκωδικοποιητή που γίνονται «1» στους επιθυμητούς συνδυασμούς εισόδου και οδηγώντας τις σε πύλες OR, παίρνουμε το λειτουργικό διάγραμμα του Σχήματος 8.9. Στο Σχήμα 8.10 φαίνεται η ίδια σχεδίαση, όπως θα μπορούσε να υλοποιηθεί στο εσωτερικό ενός ολοκληρωμένου κυκλώματος PROM.

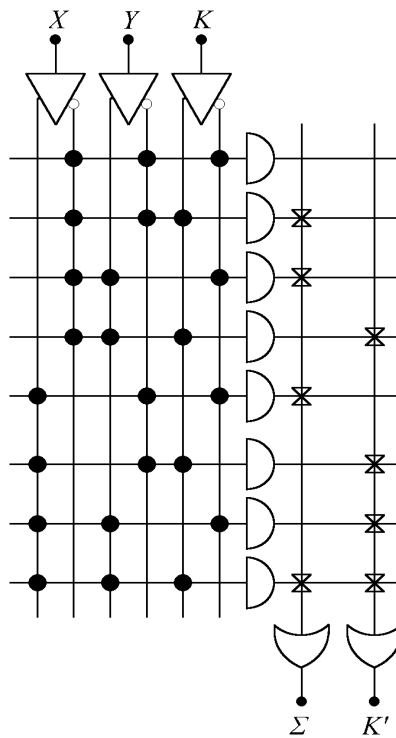
Αντίστοιχα ο πίνακας αλήθειας του πλήρους αθροιστή θα έχει τη μορφή (Πίνακας 8.2)

X	Y	K	Σ	K'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Πίνακας 8.2. Πίνακας αληθείας του πλήρους αθροιστή



Σχήμα 8.9. Λειτουργική μορφή πλήρους αθροιστή δομής AND – OR



Σχήμα 8.10. Υλοποίηση πλήρους αθροιστή με PROM

8.4.1.3 Αποκωδικοποιητής BCD σε δεκαδικό

Τα κυκλώματα των αποκωδικοποιητών είναι κατασκευασμένα με τέτοιο τρόπο, ώστε να ενεργοποιούν μια μοναδική έξοδο για κάθε ένα συνδυασμό εισόδων. Ως εκ τούτου, έχουν περισσότερες εξόδους από εισόδους. Ένα αρκετά γνωστό κύκλωμα αποκωδικοποιητή είναι αυτό που μετατρέπει μια δεκάδα τετραψήφιων δυαδικών αριθμών σε μια ακολουθία δέκα άλλων δυαδικών, από τους οποίους ο καθένας εκφράζει και ένα μονοψήφιο αριθμό του δεκαδικού συστήματος. (Κοσσίδης, 1996)

Ο αποκωδικοποιητής αυτός πρέπει να έχει 4 εισόδους, όπου θα δέχεται τους 10 διαφορετικούς συνδυασμούς αριθμών BCD, και 10 εξόδους, που η κάθε μια θα ενεργοποιείται και για ένα διαφορετικό αριθμό. Οι συναρτήσεις που περιγράφουν την κάθε μια έξοδο είναι οι παρακάτω:

$$D_0(A,B,C,D) = \Sigma(0)$$

$$D_1(A,B,C,D) = \Sigma(1)$$

$$D_2(A,B,C,D) = \Sigma(2)$$

$$D_3(A,B,C,D) = \Sigma(3)$$

$$D_4(A,B,C,D) = \Sigma(4)$$

$$D_5(A,B,C,D) = \Sigma(5)$$

$$D_6(A,B,C,D) = \Sigma(6)$$

$$D_7(A,B,C,D) = \Sigma(7)$$

$$D_8(A,B,C,D) = \Sigma(8)$$

$$D_9(A,B,C,D) = \Sigma(9)$$

Ο πίνακας αλήθειας (Πίνακας 8.3), στον οποίο φαίνονται και οι αδιάφοροι όροι, επαληθεύει τις παραπάνω εξισώσεις. (Είναι προφανές ότι οι παραπάνω όροι είναι αποτέλεσμα της απλοποίησης με τη χρήση των αδιάφορων όρων).

Από τα σύμβολα των αδιάφορων όρων που έχουν σημειωθεί στον πίνακα, γίνεται φανερός ο λόγος για τον οποίο οι λογικές συναρτήσεις δεν περιέχουν όλες τέσσερις όρους, αλλά τρεις ή και δύο ακόμα. Αυτό, βέβαια έχει σα συνέπεια να εμφανίζεται λογικό «1» στις εξόδους όχι μόνο για τους αναμενόμενους συνδυασμούς αλλά και για μερικούς άλλους. Π.χ. η έξοδος D_9 θα ενεργοποιείται όχι μόνο με το συνδυασμό ABCD=1001 αλλά και με τον ABCD=1101 ή τον ABCD=1011. Το πρόβλημα αυτό, όμως, δε γίνεται αντιληπτό, αφού από την στιγμή που αποφασίζουμε να χρησιμοποιήσουμε έναν τέτοιο αποκωδικοποιητή, έχουμε μεριμνήσει, ώστε οι συνδυασμοί των εισόδων να μην υπερβαίνουν τον κώδικα BCD (0 έως 9).

A	B	C	D	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0	X	X	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X	X	X

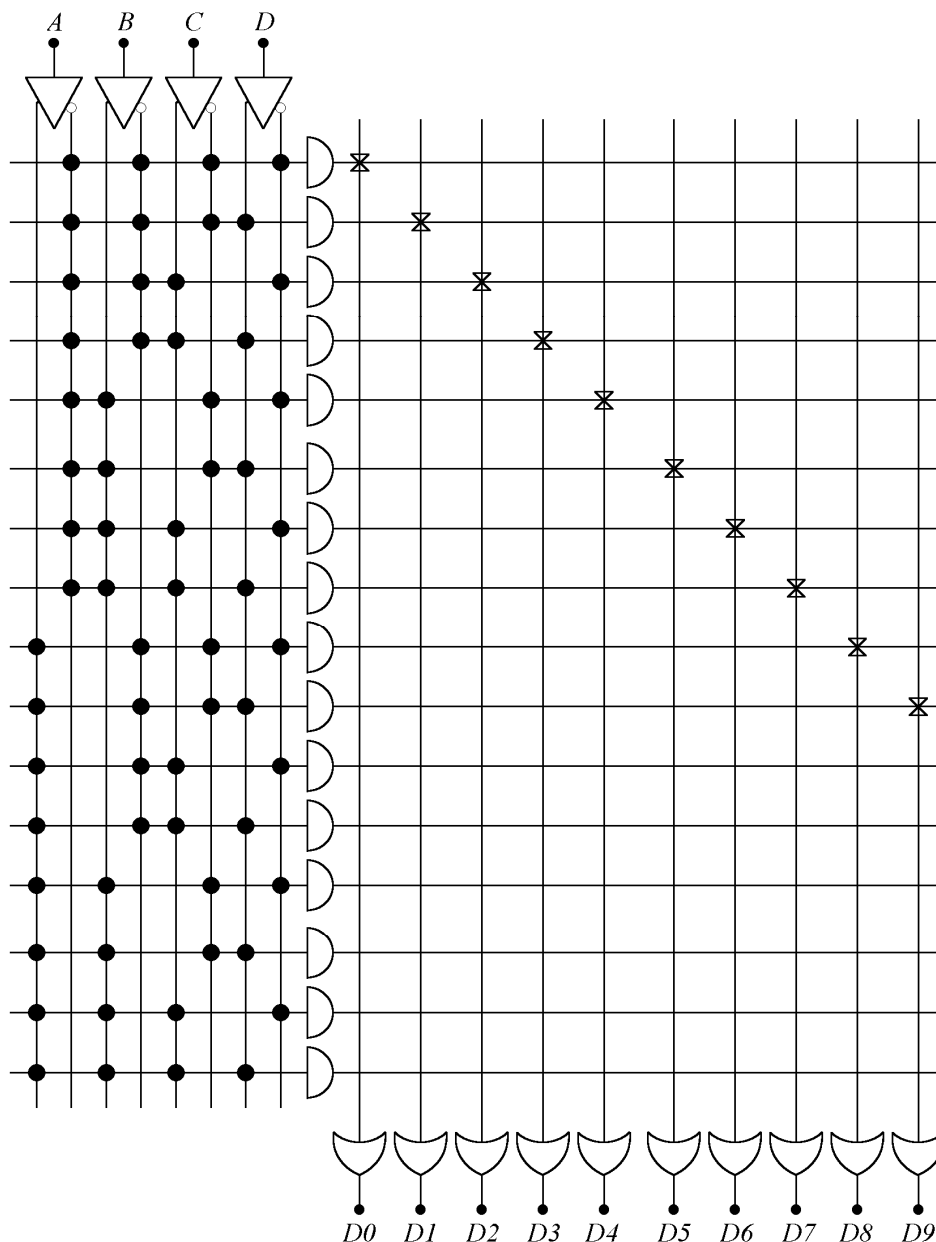
Πίνακας 8.3. Πίνακας αληθείας αποκωδικοποιητή BCD σε δεκαδικό

Αν, παρόλα αυτά, θέλουμε να είμαστε βέβαιοι πως οι έξοδοι θα ενεργοποιούνται για τους συγκεκριμένους συνδυασμούς και μόνο, τότε δεν έχουμε παρά να βασιστούμε στη σχεδίαση που ακολουθεί, και κάνει χρήση ολοκληρωμένου τύπου PROM.

Επειδή το κύκλωμα είναι αρκετά μεγάλο, παρακάμπτουμε τη λειτουργική σχεδίαση και προχωρούμε κατευθείαν σ' αυτή που βασίζεται σε PROM, και η οποία φαίνεται στο Σχήμα 8.11.

Παρατηρήστε ότι χρησιμοποιούνται μόνο οι 10 από τις 16 εξόδους πυλών AND που σχηματίζουν έναν αποκωδικοποιητή από 4 σε 16 γραμμές, κάτι αναμενόμενο, αφού οι συνδυασμοί που υπερβαίνουν τον κώδικα BCD θεωρούνται άκυροι, και, φυσικά, δεν πρέπει να ενεργοποιούν καμία έξοδο.

Οι εξοδοί των πυλών AND οδηγούνται σ' έναν προγραμματιζόμενο πίνακα του οποίου οι στήλες αποτελούν τις εισόδους των πυλών OR. Στη συγκεκριμένη εφαρμογή, θεωρούμε ότι η μνήμη έχει 10 εξόδους. Σε κάθε μια πύλη OR χρησιμοποιείται μόνο μια είσοδος η οποία καταλήγει σε μια μόνο έξοδο πύλης AND. Έτσι η AND που δίνει λογικό «1» όταν ABCD=0000, συνδέεται στην είσοδο της πύλης που παράγει το D₀, η AND που δίνει λογικό «1» όταν ABCD=0001, στην είσοδο της πύλης που παράγει το D₁ κ.λπ.



Σχήμα 8.11. Σχεδίαση αποκωδικοποιητή 4 σε 10 βασισμένη σε PROM 16x10

Στα προηγούμενα παραδείγματα φάνηκε ότι η PROM αποτελείται από ένα σταθερό, μη προγραμματιζόμενο, πίνακα AND που αποτελεί στην πραγματικότητα έναν αποκωδικοποιητή όλων των δυνατών τιμών των μεταβλητών εισόδου.

Αντίθετα ο πίνακας OR, που σχηματίζεται στις εξόδους των πυλών AND και υλοποιείται με διασυνδέσεις ασφαλειοδιόδων, είναι προγραμματιζόμενος, κάνοντας αυτό το εξάρτημα να συμπεριφέρεται κάθε φορά με διαφορετικό τρόπο. Γίνεται, επομένως, φανερό πως η PROM είναι ένα εξάρτημα λογικής AND-OR με σταθερή μήτρα (πίνακα) AND και προγραμματιζόμενη μήτρα (πίνακα) OR.

8.5. Εισαγωγή στα PLDs

Οι προγραμματιζόμενες λογικές μονάδες (PLD) χωρίζονται σε τρεις μεγάλες κατηγορίες:

α) Τα SPLD (simple programmable Logic Devices) που είναι ο παλαιότερος τύπος λογικών μητρώων (με βασικούς τύπους τα PAL τα οποία θα μελετήσουμε αναλυτικά, τα GAL (generic array logic) και τέλος τα PLA. Και οι τρεις χρησιμοποιούνται για δεδομένες συναρτήσεις και για μικρό αριθμό πυλών. Τα PAL είναι μια φορά προγραμματιζόμενα ενώ τα GAL είναι επαναπρογραμματιζόμενα. Και στα δύο η δομή περιλαμβάνει μια προγραμματιζόμενη μήτρα AND και μια σταθερή μήτρα OR. Τα GAL με τη δυνατότητα επαναπρογραμματισμού χρησιμοποιούνται, κυρίως, στην ανάπτυξη προϊόντων και για εκπαιδευτικούς λόγους. Η τεχνολογία που χρησιμοποιούν είναι EEPROM (E²CMOS) αντί των διόδων και ασφαλειών.

β) Τα CPLDs (complex PLDs) είναι πολλαπλές μήτρες που διασυνδέονται σε ένα τσιπ. Σε αυτήν την ομάδα ανήκουν τα SPLD EPLD, CPLD, EEPLD, XPLD. Τα δομικά PLD ονομάζονται μικροκυψέλες (microcells) και συνδέονται μέσω του γενικού πίνακα διασύνδεσης.

γ) τα FPGA (Field Programmable Gate Arrays) έχουν μεγαλύτερη χωρητικότητα από τα CPLD και μεγαλύτερη ευελιξία. Σε αυτήν την ομάδα ανήκουν τα FPGA, SPGA (system), XPGA (eXpanded), pASIC (programmable ASIC) και τα LCA (Logic Cell array).

Η δομή του FPGA έχει 3 μέρη: το λογικό μπλοκ, το μπλοκ διασύνδεσης και εισόδου/εξόδου. Τα μπλοκ εισόδου/εξόδου που συνδέονται με τους ακροδέκτες (pins) I/O στη συσκευασία. Το Λογικό μπλοκ αποτελείται από LUTs (πίνακες look-up) και στοιχεία μνήμης (FF). Αυτά χρησιμοποιούνται για την υλοποίηση διαφόρων λογικών συναρτήσεων. Τα μπλοκ διασύνδεσης συνδέουν τα λογικά μπλοκ. Η διασύνδεση μπορεί να προγραμματιστεί. (Nelson et al., 1995; Κοσσιδάς, 1996; Holdsworth, & Woods, 2002; Givone, 2002; Balch, 2003; Katz, 2005; Balabanian, & Carlson 2007; Godse, & Godse, 2010; Mano, & Ciletti, 2014; Roth, & Kinney, 2014).

8.5.1. Προγραμματιζόμενες λογικές διατάξεις

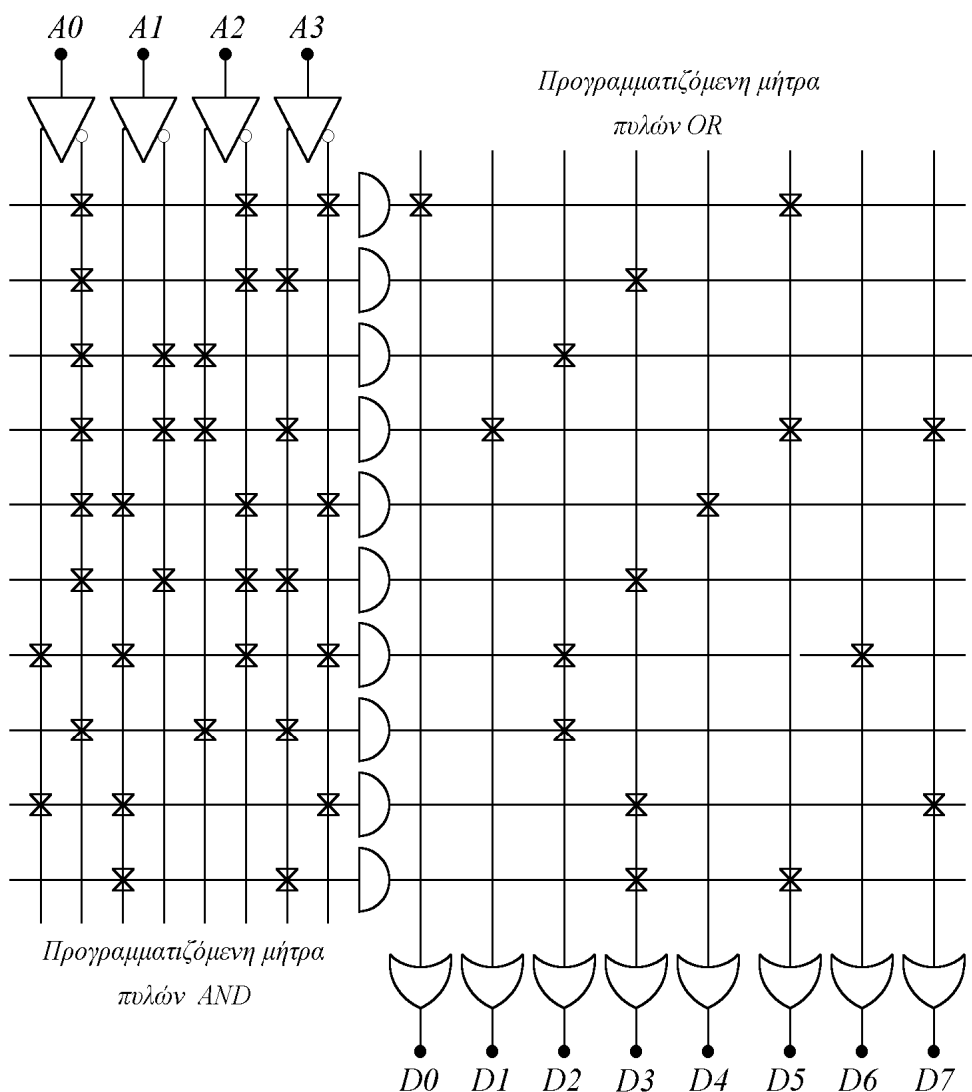
Είναι γνωστό ότι, αν έχουμε, π.χ., μια μνήμη PROM πέντε εισόδων, οι συνδυασμοί των μεταβλητών που μπορεί να δεχθεί είναι ίσοι με $2^5=32$. Για να είναι σε θέση η μνήμη να ανταποκριθεί σε οποιονδήποτε από αυτούς, θα πρέπει να περιλαμβάνει αποκωδικοποιητή ισαρίθμων εξόδων και φυσικά τα κυκλώματα που ακολουθούν αυτόν. Στην πράξη, όμως, έχει αποδειχθεί πως όταν έχουμε n μεταβλητές, σπάνια θα έχουμε 2^n συνδυασμούς, ειδικά αν οι μεταβλητές αυτές συμμετέχουν στη δημιουργία διαφορετικών συναρτήσεων. Σε μια τέτοια περίπτωση, εύκολα γίνεται αντιληπτό ότι τα κυκλώματα αποκωδικοποίησης, που συναντώνται στην είσοδο των PROM, δεν αξιοποιούνται πλήρως, με συνέπεια το ίδιο το ολοκληρωμένο να χαρακτηρίζεται μη αποδοτικό και ακριβό από οικονομική άποψη.

Το πρόβλημα αυτό έρχεται να το λύσει ένα άλλου είδους ολοκληρωμένο κύκλωμα, που είναι γνωστό με το όνομα PLA (Programmable Logic Array) (προγραμματιζόμενες λογικές διατάξεις). Το PLA μοιάζει με την PROM με τη διαφορά ότι δεν κάνει πλήρη αποκωδικοποίηση των μεταβλητών εισόδου, μη αναδεικνύοντας, επομένως, όλους τους δυνατούς συνδυασμούς.

Στο PLA έχουμε πάλι μια ομάδα από πύλες AND που το πλήθος τους όμως καθορίζεται από τον κατασκευαστή του εξαρτήματος, και είναι μικρότερο από το 2^n . Ακόμα, οι εισοδοί των πυλών αυτών δεν είναι σταθερά συνδεδεμένες στις εξόδους των βαθμίδων εισόδου, αλλά μπορούν να συνδεθούν σε όποιες και όσες επιθυμεί ο σχεδιαστής της εφαρμογής. Στο Σχήμα 8.13α δίνεται η μορφή ενός PLA 4 εισόδων και 8 εξόδων το οποίο όμως περιέχει 10 μόνο (αντί 16 όπως μια PROM) πύλες AND. Το γεγονός ότι το πλήθος των συνδυασμών που μπορούν να αναδειχθούν είναι μικρότερο από 2^n , είναι αυτό που επιτρέπει στο PLA να διαχειρίζεται μεγαλύτερο αριθμό εισόδων σε σχέση με αυτόν που διαχειρίζεται μια PROM, ολοκληρώνοντας στο ίδιο ολοκληρωμένο κύκλωμα τα ίδια εξαρτήματα μ' αυτά της PROM.

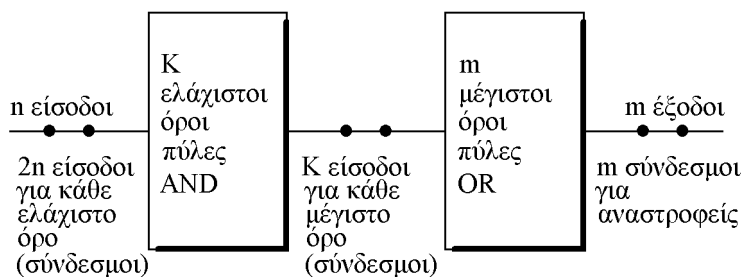
Οι έξοδοι των πυλών AND συνδυάζονται στην, επίσης, προγραμματιζόμενη μήτρα εξόδου που καταλήγει στις εισόδους των OR, για να δημιουργηθεί μια λογική εξίσωση βασισμένη σε δομές AND-OR.

Γνωρίζοντας τα παραπάνω, είναι φανερό πως η πραγματοποίηση του αποκωδικοποιητή BCD σε δεκαδικό, που αναφέραμε σε προηγούμενη παράγραφο, γίνεται πολύ πιο απλή, αφού δεν είναι αναγκαίο να χρησιμοποιήσουμε PROM με 16 πύλες AND τη στιγμή που μας χρειάζονται μόνο 10.



Σχήμα 8.13α. PLA με 4 εισόδους, 8 εξόδους και 10 πύλες AND

Τα PLA είναι σε θέση να διαχειρισθούν πολύ περισσότερες μεταβλητές σε σχέση με τις PROM. Εξαιτίας όμως, των δύο προγραμματιζόμενων πινάκων και του κόστους των κυκλωμάτων που απαιτούνται για τον προγραμματισμό, τα PLA δεν μπορούν να χρησιμοποιηθούν αποδοτικά και, συνεπώς, οικονομικά σε εφαρμογές μικρής πολυπλοκότητας. Αυτός είναι και ο λόγος για τον οποίο τα PLA δεν έχουν εκτοπίσει ολοκληρωτικά τις PROM. Μια γενική μορφή παράστασης του PLA φαίνεται στο Σχήμα 8.13β.



Σχήμα 8.13β Γενική μορφή ενός PLA

8.5.1.1 Παράδειγμα με PLA

Για να γίνει δυνατός ο καθορισμός των συνδέσεων στις μήτρες των πυλών AND και OR, ακολουθούμε μια συγκεκριμένη σειρά ενεργειών, που οδηγεί στην κατάστρωση ενός πίνακα. Ο πίνακας αυτός περιλαμβάνει μία στήλη στην οποία σημειώνονται οι όροι γινομένου που συμμετέχουν στις συναρτήσεις που θέλουμε να πραγματοποιηθούν μέσω του PLA, μία δεύτερη στήλη στην οποία σημειώνονται οι εισόδοι του κυκλώματος και, τέλος, μία τρίτη στην οποία σημειώνονται οι λογικές καταστάσεις των εξόδων του κυκλώματος.

Είναι προφανές ότι το PLA, που θα χρησιμοποιηθεί για την πραγματοποίηση της σχεδίασης, θα πρέπει να περιέχει τουλάχιστον τόσες πύλες AND όσοι είναι και οι όροι γινομένου, και τόσες πύλες OR όσες και οι συναρτήσεις, που επιθυμούμε να πραγματοποιήσουμε.

Περαιτέρω για τον τρόπο με τον οποίο συμπληρώνεται ο πίνακας, όπως και για τον τρόπο με τον οποίο μεταφέρονται τα δεδομένα του πίνακα στις μήτρες των πυλών AND και OR, θα αναφέρουμε στο παρακάτω παράδειγμα.

Έστω ότι θέλουμε να υλοποιήσουμε με PLA τις παρακάτω λογικές συναρτήσεις:

$$T_1(A, B) = \Sigma(4, 5, 7) \text{ και } T_2(A, B) = \Sigma(3, 5, 7). \text{ ή}$$

$$T_1 = A \cdot \bar{B} + A \cdot C \text{ και } T_2 = A \cdot C + B \cdot C$$

Ο πίνακας αλήθειας, που τις επαληθεύει, δίνεται στον πίνακα 8.4:(να γράψετε την απλοποιημένη συνάρτηση).

Παρατηρούμε, καταρχήν, ότι στις δύο αυτές συναρτήσεις συμμετέχουν τρεις διαφορετικοί όροι γινομένου. Από αυτό προκύπτει ότι ο πίνακας θα αποτελείται από τρεις γραμμές. Σημειώνουμε, επομένως, σε κάθε μια γραμμή της πρώτης στήλης, τους όρους αυτούς. Στη δεύτερη στήλη του πίνακα που αποτελείται από τις τρεις επιμέρους στήλες **A**, **B**, **C**, σημειώνουμε «1», «0», «-», ανάλογα με το αν ο συγκεκριμένος όρος γινομένου περιλαμβάνει τη συγκεκριμένη μεταβλητή, το συμπλήρωμά της ή δεν περιλαμβάνει καθόλου αυτή τη μεταβλητή, αντίστοιχα. Στην τρίτη στήλη, που αποτελείται από τις δύο επιμέρους στήλες T_1 και T_2 , θα σημειώσουμε, γράφοντας «1» ή «-» το αν στη συγκεκριμένη συνάρτηση συμμετέχει, ή όχι, κάποιος από τους τρεις όρους γινομένου.

A	B	C	T_1	T_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

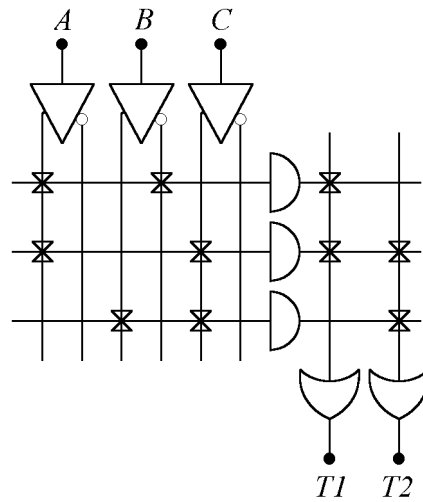
Πίνακας 8.4. Πίνακας αληθείας αποκωδικοποιητή BCD σε δεκαδικό

Μετά από τα παραπάνω, γίνονται εύκολα κατανοητά τα σύμβολα που έχουν καταχωρηθεί στον πίνακα 8.1.

Όροι γινομένου	Είσοδοι			Έξοδοι	
	A	B	C	T_1	T_2
$A \cdot \bar{B}$	1	0	-	1	-
$A \cdot C$	1	-	1	1	1
$B \cdot C$	-	1	1	-	1

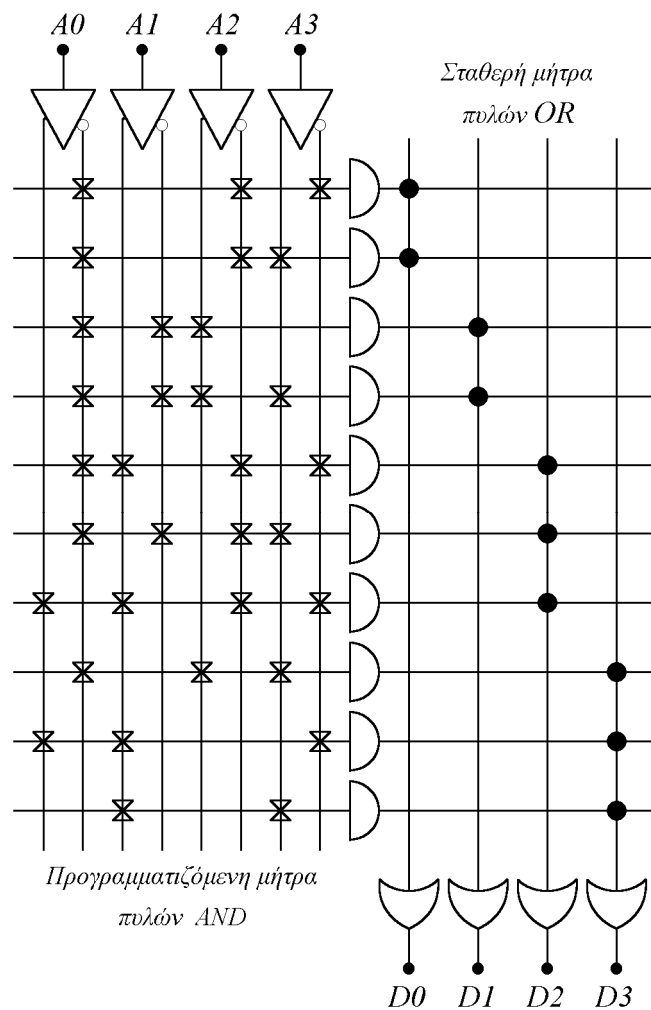
Στη συνέχεια, πρέπει να μεταφέρουμε τα συμπεράσματα του πίνακα στις δύο μήτρες του PLA. Στη μήτρα εισόδου θα συνδέσουμε στις δύο εισόδους της πρώτης πύλης AND τις μεταβλητές **A** και **C**, στις εισόδους της δεύτερης τις μεταβλητές **A** και **C**, ενώ στις εισόδους της τρίτης τις μεταβλητές **B** και **C**. Στη μήτρα εξόδου θα συνδέσουμε στις εισόδους της πρώτης πύλης OR τις εξόδους της πρώτης και της δεύτερης πύλης AND, ενώ στις εισόδους της δεύτερης πύλης OR τις εξόδους της δεύτερης και της τρίτης πύλης AND. Στο

Σχήμα 8.14 φαίνεται το εσωτερικό ενός PLA τριών εισόδων και δύο εξόδων, έτσι όπως έχει διαμορφωθεί σύμφωνα με την παραπάνω μεθοδολογία.



Σχήμα 8.14. Πραγματοποίηση συναρτήσεων με PLA

8.5.2. Προγραμματιζόμενες λογικές διατάξεις(PAL)

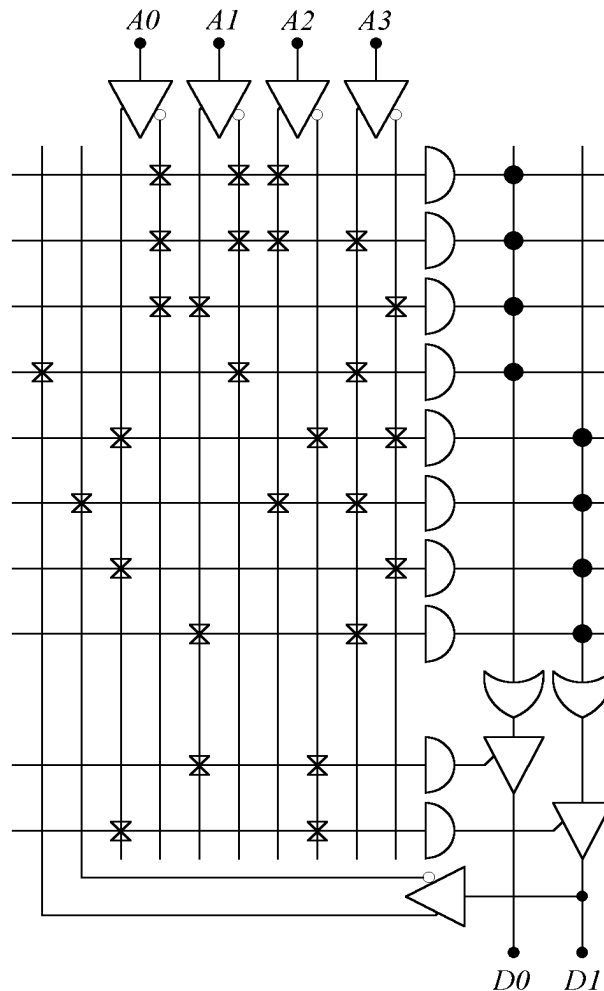


Σχήμα 8.15. Τυπικό κύκλωμα PAL

Το αυξημένο κόστος των PLA έρχονται να αντιμετωπίσουν τα PAL(Programmable-Array Logic), τα οποία διαφέρουν σε σχέση με τα PLA στο ότι μπορούν να προγραμματισθούν μόνο στη μήτρα της εισόδου τους. Αυτό έχει σα συνέπεια να προγραμματίζονται οι επιθυμητές συναρτήσεις μόνο στις εισόδους των πυλών AND, οι έξοδοι των οποίων είναι ακλόνητα συνδεδεμένες στις εισόδους των πυλών OR που ακολουθούν. Στο Σχήμα 8.15 αποδίδεται ένα τυπικό PAL 4 εισόδων και 4 εξόδων.

Ο περιορισμός των συναρτήσεων, που μπορούν να πραγματοποιήσουν, λόγω της κατασκευής τους, αντιμετωπίζεται με τη διάθεση στο εμπόριο πολλών διαφορετικών ολοκληρωμένων κυκλωμάτων PAL, τα οποία διαφέρουν στο πλήθος των εισόδων τους, τον αριθμό των εξόδων τους και στο πλήθος των όρων γινόμενου που μπορούν να δεχθούν σε κάθε πύλη OR που περιλαμβάνουν. Έτσι θα συναντήσουμε ολοκληρωμένα κυκλώματα με 2, 8, ή 10 εξόδους, που η κάθε μια πύλη OR που περιλαμβάνουν μπορεί να τροφοδοτηθεί από 2, 4, 7 ή 8 εξόδους πυλών AND. Φυσικά καθεμία από τις πύλες AND διαθέτει αριθμό εισόδων ο οποίος ισούται με το διπλάσιο των μεταβλητών εισόδου που μπορεί να διαχειριστεί το PAL.

Η τεχνολογία στην κατασκευή των PAL έχει προχωρήσει αρκετά έτσι ώστε, εκτός από τις καθαρά συνδυαστικές λειτουργίες, να μπορούν τα PAL να πραγματοποιήσουν και ακολουθιακές. Σε αρκετούς τύπους PAL, οι έξοδοι μερικών ή όλων των πυλών OR καταλήγουν στις εισόδους D των ομώνυμων f - f , των οποίων οι έξοδοι ενεργοποιούνται με τη βοήθεια παλμών ρολογιού που επιβάλλεται σε ειδικό ακροδέκτη του εξαρτήματος. Επιπλέον βελτιώσεις επιτρέπουν την περαιτέρω τροποποίηση των χαρακτηριστικών λειτουργίας, έτσι ώστε να γίνεται δυνατή η μετατροπή των f - f σε τύπου T, R-S, ή J-K, εξυπηρετώντας, με αυτόν τον τρόπο, οποιαδήποτε σχεδιαστική ιδιαιτερότητα.



Σχήμα 8.16. PAL με ανατροφοδότηση από τις εξόδους

Οι λογικές καταστάσεις που αναδεικνύονται στις εξόδους των f - f επανεισάγονται στο εσωτερικό των PAL, με σκοπό την περαιτέρω επεξεργασία. Η παρουσία μιας ακόμα βαθμίδας που είναι γνωστή σαν «Απο-

μονωτής τριών καταστάσεων» (Hi-Z buffer), στις εξόδους των βελτιωμένων PAL, επιτρέπει τη σύνδεσή τους στα διάφορα bus των υπολογιστών, κάνοντας τα εξαρτήματα αυτά κατάλληλα για χρήση σε πλακέτες ηλεκτρονικών υπολογιστών. Οι βαθμίδες αυτές επιτρέπουν, όταν ο ακροδέκτης ελέγχου τους είναι ενεργός, τη μεταφορά της ψηφιακής πληροφορίας μέσα από αυτές, ενώ, όταν είναι ανενεργός, την εμποδίζουν, εμφανίζοντας στις εξόδους τους κατάσταση υψηλής αντίστασης. Με αυτόν τον τρόπο οι ακίδες εξόδου μπορούν να μετατραπούν σε ακίδες εισόδου. Η τελευταία λειτουργία αποδεικνύεται πολύ χρήσιμη στις περιπτώσεις εκείνες στις οποίες ένα κύκλωμα απαιτεί πολλές εισόδους, και μόνο μια ή δύο εξόδους, και φυσικά το PAL, που πρέπει να χρησιμοποιήσουμε, διαθέτει περισσότερες εξόδους από αυτές που χρειαζόμαστε. Στο Σχήμα 8.16 βλέπουμε ένα τέτοιο κύκλωμα στο οποίο, κάτω από ορισμένες συνθήκες η έξοδος D₁ μετατρέπεται σε είσοδο.

8.5.2.1 Παράδειγμα με PAL

Η σχεδίαση κυκλωμάτων στο εσωτερικό των PAL είναι πολύ πιο εύκολη από εκείνη των PLA. Αρκεί να τροφοδοτήσουμε τις μεταβλητές που συνθέτουν τους όρους γινομένου της συνάρτησης στις πύλες AND, για να έχουμε στην έξοδο της πύλης OR, που οδηγούν οι AND, το αποτέλεσμα της συνάρτησης που επιθυμούμε. Θα πρέπει, φυσικά, προτού αρχίσουμε τη σχεδίαση, να βεβαιωθούμε ότι τα μερικά γινόμενα, που αποτελούν τη συνάρτηση, δεν ξεπερνούν στο σύνολο τους το πλήθος των εισόδων της πύλης OR, η οποία θα μας δώσει το αποτέλεσμα.

Δεν θα πρέπει, φυσικά, να παραβλέψουμε τους περιορισμούς που εισάγουν οι ίδιες οι έξοδοι των PAL. Αν, π.χ., θέλετε σ' ένα μόνο PAL να πραγματοποιήσετε πέντε διαφορετικές συναρτήσεις τη στιγμή που οι έξοδοι του είναι μόνο τέσσερις, τότε κάτι τέτοιο σίγουρα δεν είναι εφικτό. Αν, αντίθετα, χρειάζεστε εννέα εισόδους τη στιγμή που το PAL έχει μόνο οκτώ, τότε ίσως μπορέσετε να δώσετε λύση στο πρόβλημα, «κλέβοντας» μια ακίδα εξόδου και μετατρέποντάς τη σε είσοδο.

Έστω ότι θέλουμε να υλοποιήσουμε με PAL τις παρακάτω συναρτήσεις:

$$\begin{aligned} T_1 &= A\bar{B} + AC & T_2 &= B\bar{C} + AC \\ T_3 &= AC & T_4 &= \bar{A} \end{aligned}$$

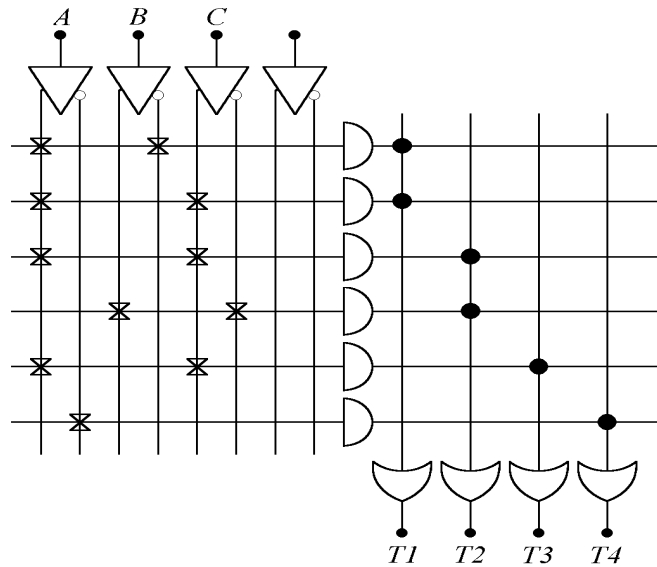
Ακολουθεί ο πίνακας αλήθειας των συναρτήσεων αποδίδεται στον πίνακα 8.5:

A	B	C	T ₁	T ₂	T ₃	T ₄
0	0	0	0	0	0	1
0	0	1	0	0	0	1
0	1	0	0	1	0	1
0	1	1	0	0	0	1
1	0	0	1	0	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	0
1	1	1	1	1	1	0

Πίνακας 8.5. Πίνακας αληθείας αποκοδικοποιητή BCD σε δεκαδικό

Για την πραγματοποίηση των παραπάνω συναρτήσεων, θα χρειασθούμε ένα PAL το οποίο θα πρέπει να διαθέτει τουλάχιστον 3 εισόδους (η πλεονάζουσα είσοδος του Σχήματος 8.17 δείχνει αυτή τη δυνατότητα) και 4 εξόδους.

Θα πρέπει, ακόμα, να δώσουμε σημασία στο σύνολο των πυλών AND που οδηγούν τις πύλες OR. Για τις δύο πρώτες συναρτήσεις θα πρέπει οι έξοδοι T₁ και T₂ να ενεργοποιούνται από δύο όρους γινομένου, ενώ για τις δύο τελευταίες οι T₃ και T₄ θα πρέπει να ενεργοποιούνται από έναν όρο. Στο Σχήμα 8.17 βλέπουμε το εσωτερικό του PAL, όπως διαμορφώνεται μετά τον προγραμματισμό του.

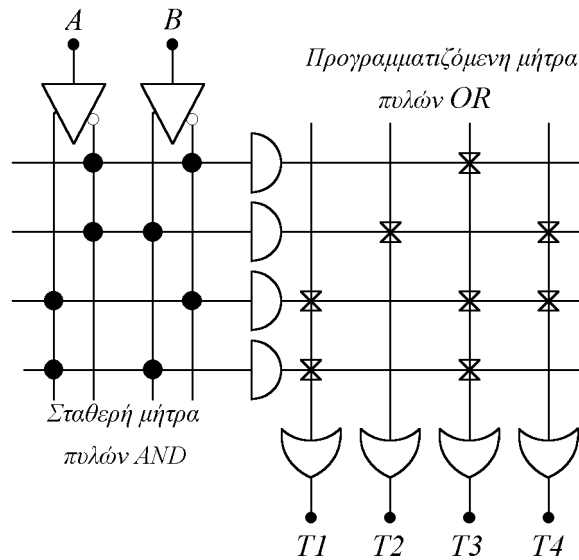


Σχήμα 8.17. Πραγματοποίηση συναρτήσεων με PAL

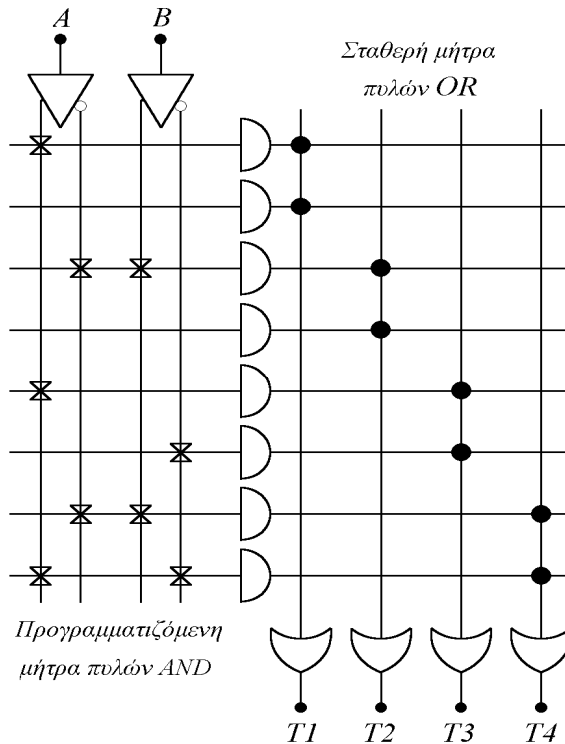
8.5.3. Σύγκριση των PROM - PLA - PAL

Στο Σχήμα 8.18, που ακολουθεί, φαίνονται οι τρόποι που πραγματοποιούνται τέσσερις λογικές συναρτήσεις στο εσωτερικό μιας μνήμης PROM, ενός PLA και ενός PAL. Για να γίνει ευκολότερη η σύγκριση, καθεμία υλοποίηση παριστάνεται σαν ένας πίνακας πυλών AND, ακολουθούμενη από ένα πίνακα πυλών OR. Οι συναρτήσεις που δίδονται είναι οι:

$$\begin{aligned}
 T_1 &= A & T_2 &= \bar{A}B \\
 T_3 &= \bar{B} + A & T_4 &= A\bar{B} + \bar{A}B
 \end{aligned}$$



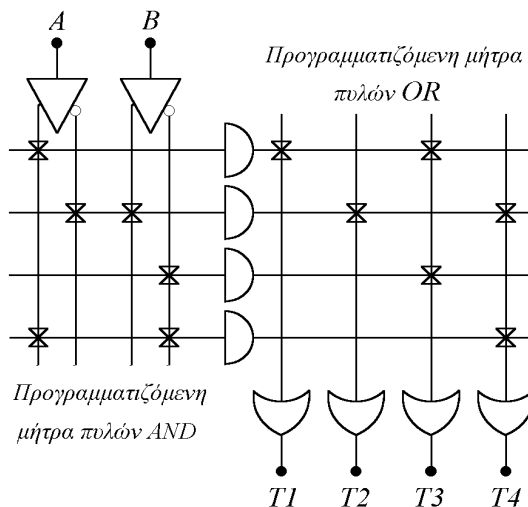
Σχήμα 8.18α. Επίλυση με PROM



Σχήμα 8.18β. Επίλυση με PAL

Η λύση της PROM, που φαίνεται στο παραπάνω σχήμα, απαιτεί ένα πίνακα 16 σημείων (ασφαλειών), ενώ τα PLA και PAL απαιτούν 32 ασφάλειες το καθένα. Αν προσθέταμε ακόμη μια μεταβλητή εισόδου, τότε το πλήθος των ασφαλειών στην PROM θα αυξανόταν σε 32, στο PLA κατά 8 και στο PAL κατά 16. Έτσι φαίνεται ότι, αυξάνοντας τις μεταβλητές εισόδου, οι PROM απαιτούν περισσότερες ασφάλειες από ότι τα PLA και τα PAL.

Οι παραπάνω αναφερθείσες διατάξεις προγραμματιζόμενης λογικής (PLD) κατασκευάζονται σε συγκεκριμένα κυκλώματα με εσωτερικές λογικές πύλες, κατάλληλα συνδεδεμένες με ηλεκτρονικούς συνδέσμους. Ο «προγραμματισμός» των PLD γίνεται με το «κάψιμο» των συνδέσμων έτσι, ώστε να προκύψει η αντίστοιχη λογική συνάρτηση. Θα πρέπει να θεωρείται αναμενόμενο ότι σε αυτά τα εξαρτήματα η έννοια «προγραμματισμός» αναφέρεται στο hardware, αφού πρέπει να επέμβουμε με ηλεκτρικό τρόπο στους αντίστοιχους συνδέσμους. Τέλος πρέπει να αναφερθεί ότι το πλεονέκτημα της χρήσης των PLD στο σχεδιασμό λογικών κυκλωμάτων είναι ότι μπορούν να ενσωματώσουν πολύπλοκες λογικές συναρτήσεις μέσα σ' ένα μοναδικό κύκλωμα LSI.



Σχήμα 8.18γ. Επίλυση με PLA

Σχήμα 8.18. Σύγκριση των PROM, PLA, PAL

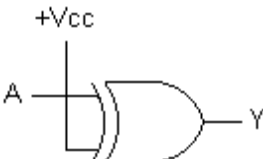
8.6. Σύγκριση GAL και PAL

Γενικά η σύγκριση των GAL και των PAL επιφέρει μια σύγχυση στους αναγνώστες. Σε γενικές γραμμές, λοιπόν, τα GAL είναι συσκευές 2ης γενιάς που έχουν τη δυνατότητα εγγραφής-διαγραφής και είναι σε θέση να εκπληρώσουν οποιαδήποτε λειτουργία των συσκευών της πρώτης γενιάς, όπως είναι οι PAL.

Τα τσιπ των PAL έχουν μια σειρά από ασφάλειες στο εσωτερικό τους. Οι ασφάλειες χρησιμοποιούνται στην διασύνδεση των λογικών πυλών του. Αυτά προγραμματίζονται με το «κάψιμο» των ασφαλειών. Τα τσιπ των GAL χρησιμοποιούν πίνακες EEPROM για την αποθήκευση αυτών των συνδέσεων και με αυτό τον τρόπο δεν χρησιμοποιούν ασφάλειες. Είναι επαναπρογραμματιζόμενες, αλλά όχι με τον τρόπο που γνωρίζουμε σήμερα για τις επαναπρογραμματιζόμενες συσκευές. Το ελάχιστο όριο για τον αριθμό των φορών που μπορούσε να προγραμματιστεί ήταν περίπου 100. Σε αντίθεση με τα PAL, των οποίων οι έξοδοι συνδέονταν με καλώδια με την έξοδο μιας λογικής πύλης, τα τσιπ των GAL είχαν προγραμματιζόμενη λογική εξόδου. Οι δύο αυτές τεχνολογίες, περιλαμβάνονται στην παρούσα τεχνολογία των PLDs. (Tocci et al., 2010)

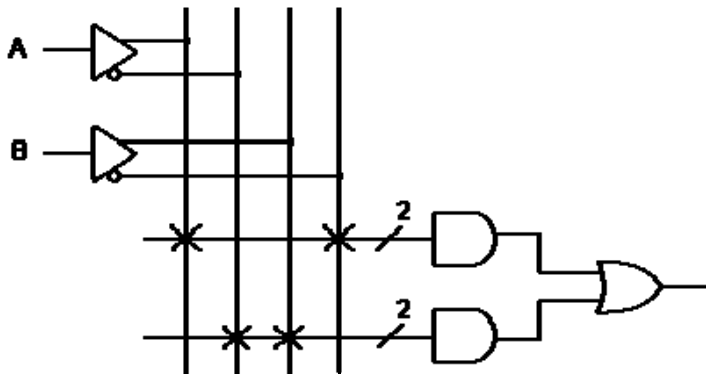
8.7. Ασκήσεις – Ερωτήσεις

- Το GAL είναι ουσιαστικά:
 - Ένα μη-επαναπρογραμματιζόμενο PAL
 - Ένα PAL προγραμματισμένο μόνο από τον κατασκευαστή του
 - Ένα πολύ μεγάλο PAL
 - Ένα επαναπρογραμματιζόμενο PAL
- Η διαφορά μεταξύ ενός PLA με το PAL είναι:
 - Το PLA έχει μια προγραμματιζόμενη μήτρα OR σε συνδυασμό με μία προγραμματιζόμενη μήτρα AND, ενώ το PAL έχει μόνο προγραμματιζόμενες μήτρες AND.
 - Το PAL έχει μία προγραμματιζόμενη μήτρα OR σε συνδυασμό με μία προγραμματιζόμενη μήτρα AND, ενώ το PLA έχει μόνο προγραμματιζόμενες μήτρες AND.
 - Το PAL είναι πιο διαδεδομένο σε σχέση με το PLA
 - Το PAL είναι το ίδιο με το PLA
- Το GAL16V8 έχει:
 - 16 αποκλειστικές εισόδους
 - 8 ακροδέκτες ειδικών λειτουργιών
 - 8 ακροδέκτες που χρησιμοποιούνται για είσοδο ή Έξοδο
 - Όλα τα παραπάνω
- ___1___ χρησιμοποιούνται στις εισόδους των PAL/GAL συσκευών, προκειμένου να αποτραπεί η φόρτωση της εισόδου από ένα μεγάλο αριθμό πυλών AND.
 - Απλοποιημένες πύλες AND
 - Ασφάλειες
 - Buffers (Απομονωτής)
 - Latch (Μανταλωτής)
- Γιατί τα PLDs έχουν καταλάβει μεγάλο μερίδιο αγοράς ;
 - Ένα PLD εκτελεί διεργασίες πολλών ICs
 - Είναι φτηνά
 - Καταναλώνουν λιγότερη ενέργεια
 - Όλα τα παραπάνω
- Τι παραπάνω μπορεί να κάνει ένα GAL22V10, σε σχέση με ένα GAL16V8;
 - Περιλαμβάνει ένα εξαιρετικά μεγάλο πίνακα
 - Προγραμματίζεται πάνω στο εκάστοτε σύστημα
 - Έχει τους διπλάσιους ακροδέκτες ειδικών λειτουργιών
 - Όλα τα παραπάνω
- Η έξοδος του κυκλώματος θα είναι πάντα:



- 1
 - 0
 - \overline{A}
 - A
- Το PAL16L8 έχει:
 - 10 εισόδους, 8 εξόδους
 - 8 εισόδους, 8 εξόδους
 - 16 εισόδους, 16 εξόδους
 - 16 εισόδους, 8 εξόδους
- Πόσες φορές μπορούν ένα διαγραφτούν δεδομένα από ένα GAL, και να επαναπρογραμματιστεί ;
 - 0
 - τουλάχιστον 100
 - τουλάχιστον 1.000
 - πάνω από 10.000
- Ποια είναι η μορφοποίηση των ακροδεκτών ενός GAL22V10
 - Έχει 10 ακροδέκτες για έξοδο και 12 για είσοδο
 - Έχει 2 ακροδέκτες ειδικού σκοπού
 - Έχει 8 ακροδέκτες που συμπεριφέρονται είτε σαν εισοδοί είτε σαν εξοδοί
 - Όλα τα παραπάνω

11. Ο κύριος τρόπος κατασκευής προγραμματιζόμενων μητρών γίνεται με τη χρήση:
 Α. πυλών OR
 Β. πυλών AND
 Γ. πυλών NOR και NAND
 Δ. πυλών AND και OR
12. Ποια είναι η σωστή λογική συνάρτηση για το παρακάτω διάγραμμα PAL;



- Α. $X = \overline{AB} + AB$
 Β. $X = \overline{AB} + \overline{BA}$
 Γ. $X = \overline{AB} + \overline{BA}$
 Δ. $X = \overline{AB} + \overline{BA}$

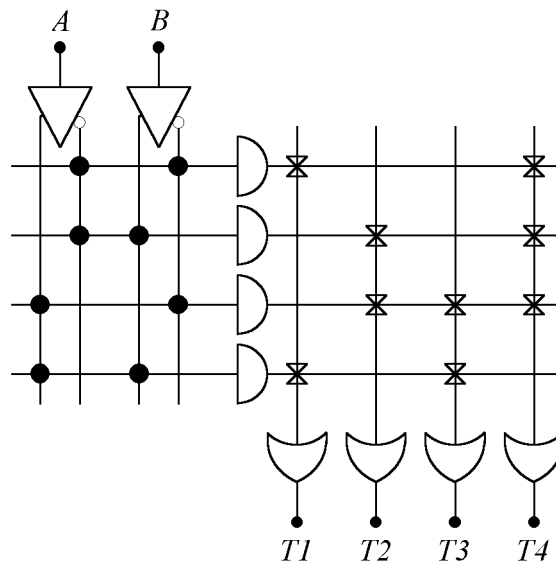
13. Ένα SPLD (simple programmable logic device) καταχωρούμενο ως 22V10 έχει:
 Α. 10 εισόδους, 10 εξόδους και χρειάζεται τροφοδοσία 22V
 Β. 11 εισόδους, 11 εξόδους και χρειάζεται τροφοδοσία 10V
 Γ. 22 εισόδους, 10 εξόδους
 Δ. 10 εισόδους, 22 εξόδους
14. Ένα CPLD (complex programmable logic device) περιέχει PAL τύπου προγραμματιζόμενα SPLDs που ονομάζονται:
 Α. Macrocells
 Β. Microcells
 Γ. Μήτρες από AND/OR
 Δ. Μήτρες με σύνδεση με ασφαλείας (fuse-link arrays)
15. Σε μια μήτρα PLD ποιού τύπου πύλες δίνουν σαν έξοδο τους όρους γινομένου;
 Α. OR
 Β. XOR
 Γ. AND
 Δ. Flip-Flop

ΑΣΚΗΣΕΙΣ

1. Έστω συνδυαστικό κύκλωμα που ορίζεται από τις συναρτήσεις:
 $T_1(A, B, C) = \Sigma(3, 5, 6, 7)$
 $T_2(A, B, C) = \Sigma(0, 2, 4, 7)$
 Να σχεδιασθεί κύκλωμα PLA τριών εισόδων και δύο εξόδων που να υλοποιεί τις παραπάνω συναρτήσεις.
2. Να σχεδιάσετε ένα συνδυαστικό κύκλωμα, χρησιμοποιώντας μια PROM. Το κύκλωμα να έχει είσοδο έναν αριθμό των 3 bit και να δίνει σαν έξοδο ένα δυαδικό αριθμό ίσο με το τετράγωνο του αριθμού εισόδου.
3. Να σχεδιάσετε συνδυαστικό κύκλωμα που να παράγει το συμπλήρωμα ως προς 9 ενός BCD ψηφίου.
4. Να σχεδιασθεί κύκλωμα το οποίο, σύμφωνα με την κατάσταση μιας γραμμής ελέγχου S, θα πραγματοποιεί πρόσθεση δύο μονοψήφιων δυαδικών αριθμών (S=0) ή την αφαίρεση των ίδιων αριθμών (S=1). Οι εισοδοί του, εκτός από την S, ονομάζονται I₁ και I₂ και οι εξοδοί του ονομάζονται O₁ και O₂ (η O₁ θα έχει το άθροισμα ή τη διαφορά, ανάλογα με την κατάσταση του σήματος ελέγχου S, ενώ η O₂ το κρατούμενο που πιθανώς έχει προκύψει).
 Να προσαρμόσετε τη σχεδίαση σας, έτσι ώστε αυτή να πραγματοποιείται με ολοκληρωμένα τύπου PROM, PLA, ή PAL Ποια από τις τρεις προσεγγίσεις είναι η καλύτερη;
5. Να σχεδιαστεί κύκλωμα το οποίο θα αθροίζει δύο αριθμούς των δύο bit και θα παράγει ένα διψήφιο αποτέλεσμα συνοδευόμενο από ψηφίο κρατουμένου. Για την πραγματοποίησή του να χρησιμοποιήσετε PROM ή PAL.
6. Χρησιμοποιώντας το Σχήμα 8.16, να απαντήσετε στις παρακάτω ερωτήσεις:
 • Για ποιους συνδυασμούς των μεταβλητών εισόδου η έξοδος D₁ μετατρέπεται σε είσοδο;

- Ποια είναι η συνάρτηση που προσδιορίζει την έξοδο D_0 και ποια αυτή που προσδιορίζει την D_1 ;

7. Αφού αναγνωρίσετε το είδος του κυκλώματος, που φαίνεται παρακάτω, προσδιορίστε τις συναρτήσεις που παράγονται από τις εξόδους T_1, T_2, T_3 και T_4 .



8. Δίνονται οι συναρτήσεις $T_1 = B$, $T_2 = \bar{A} \cdot B$, $T_3 = \bar{A} + \bar{B}$ και $T_4 = \bar{A} \cdot B + A \cdot \bar{B}$

Να πραγματοποιήσετε τις τέσσερις λογικές συναρτήσεις στο εσωτερικό μιας μνήμης PROM, ενός PLA και ενός PAL. Να γίνει η σύγκριση μεταξύ τους.

Αναφορές-Βιβλιογραφία

- Δεληγιάννης†, Θ. (2004). *Ηλεκτρονικά Αναλογικά και Ψηφιακά*, Ίδρυμα Ευγενίδου
- Κοσσίδης, Α.Θ. (1996). *Σχεδίαση Ψηφιακών Κυκλωμάτων*, Εκδόσεις Μπένο
- Κοσσίδης, Α.Θ., Γιαννακόπουλος, Π., (2006), *Αριθμητικά Συστήματα και Ψηφιακά Κυκλώματα*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα
- Balabanian, N., Carlson, B. (2007). *Digital Logic Design Principles*, John Wiley
- Givone, D. (2002). *Digital Principles and Design*,. Mc Graw Hill
- Godse, A.P., Godse, D.A. (2010). *Digital Logic Design and Application*, Technical Publications Pune
- Holdsworth, Brian, Woods, Clive (2002). *Digital Logic Design*, 4th Edition, Newnes
- Katz, R. (2005). *Contemporary Logic Design*, 2/e, Prentice Hall
- Mano, M., Ciletti, M. (2014). *Ψηφιακή Σχεδίαση*, 5^η έκδοση, Παπασωτηρίου
- Maxfield Clive (2009). *Bebop to the Boolean Boogie, An Unconventional Guide to Electronics*, 3rd, Newnes, Elsevier <http://english.360elib.com/datu/T/EM140371.pdf>
- Nashelsky, Louis (1994). *Introduction to Digital Technology*, 4th Ed., Prentice Hall
- Nelson, V., Nagle, H., Carroll, B., Irwin, J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall
- Predko, Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill
- Roth, Charles Jr., Kinney, L.L., (2014). *Fundamentals of Logic Design*, 7th, Cengage Learning
- Tocci, R. J., Widmer, N. S, and Moss, Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley
- Wakerly, J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Κεφάλαιο 9

Σύνοψη

Η κεντρική μονάδα επεξεργασίας του υπολογιστή αποτελείται από τους καταχωρητές, τη μονάδα ελέγχου, την αριθμητική και λογική μονάδα, το ρολόι και, φυσικά, τους διαύλους δεδομένων. Θα γίνει μια επισκόπηση των βασικών αυτών μονάδων, ολοκληρώνοντας την παρουσίαση των θεμελιωδών γνώσεων που απαιτούνται για έναν υπολογιστή. Αρχικά γίνεται μια εισαγωγή στους καταχωρητές με τη χρήση των ήδη μελετημένων $f-f$. Ακολουθεί εκτενής περιγραφή των μνημών RAM, με πλήρη ανάλυση του τρόπου κατασκευής τους, η μνήμη ROM και άλλες χρησιμοποιούμενες δυναμικές μνήμες. Επίσης παρουσιάζεται η περιγραφή των βασικών πράξεων της αριθμητικής και λογικής μονάδας και δίνεται η δομή της. Ολοκληρώνοντας, αναφέρονται οι δικτυακές λειτουργίες και, τέλος, δίνεται η περιγραφή της αρχιτεκτονικής των υπολογιστών.

Προαπαιτούμενη γνώση

Η γνώση της λειτουργίας των $f-f$

9. Καταχωρητές, Μνήμες, Αριθμητική και Λογική Μονάδα

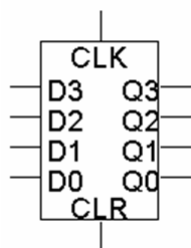
9.1 Εισαγωγή στους καταχωρητές

Οι καταχωρητές είναι ένα καλό παράδειγμα ακολουθιακής ανάλυσης και σχεδίασης, και για αυτό τον λόγο περιλαμβάνονται στην ύλη αυτού του βιβλίου. Χρησιμοποιούνται συχνά για τη κατασκευή μεγαλύτερων ακολουθιακών κυκλωμάτων, αποθηκεύουν δε μεγαλύτερη ποσότητα δεδομένων από ό,τι τα κοινά $f-f$ και είναι η καρδιά της σχεδίασης των μοντέρνων επεξεργαστών. (Holdsworth, & Woods, 2002; Katz, 2005; Κοσσίδας & Γιαννακόπουλος, 2006; Floyd, 2006; Balabanian, & Carlson 2007; Mano, & Ciletti, 2014; Roth, & Kinney, 2014)

Τα όρια των $f-f$ είναι πεπερασμένα, γιατί μπορούν να αποθηκεύσουν μόνο ένα bit. Στους μετρητές των δύο-bit, έπρεπε να χρησιμοποιήσουμε δύο $f-f$, όπως έχουμε δει. Οι περισσότεροι υπολογιστές δουλεύουν με ακέραιους αριθμούς και αριθμούς κινητής υποδιαστολής απλής ακριβείας, με μήκος 32-bits.

Ένας καταχωρητής είναι μια επέκταση ενός $f-f$, και μπορεί να αποθηκεύσει πολλά bits. Οι καταχωρητές, συνήθως, χρησιμοποιούνται σε έναν επεξεργαστή για την προσωρινή αποθήκευση δεδομένων. Είναι πιο γρήγοροι και πιο εύχρηστοι από ό,τι η κύρια μνήμη. Οι περισσότεροι καταχωρητές μπορούν να βοηθήσουν στην επιτάχυνση πολύπλοκων υπολογισμών.

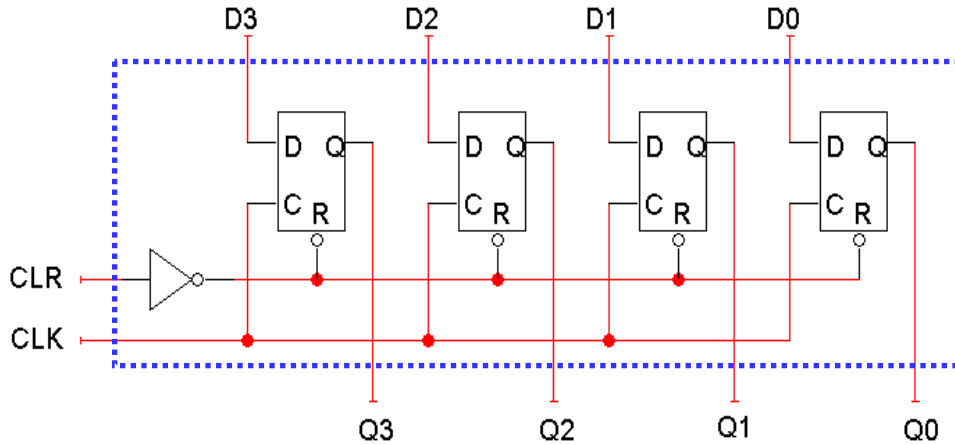
Οι βασικοί καταχωρητές είναι εύκολοι στην κατασκευή. Μπορούμε να αποθηκεύσουμε πολλά ψηφία, απλά τοποθετώντας μαζί, ένα σύνολο από $f-f$. Το δομικό διάγραμμα ενός καταχωρητή 4-bit απεικονίζεται στο Σχήμα 9.1, η δε εσωτερική του δομή αποδίδεται στο Σχήμα 9.2.



Σχήμα 9.1. Δομικό διάγραμμα καταχωρητή 4 δυαδικών ψηφίων

Όλα τα $f-f$ μοιράζονται κοινό σήμα CLK και CLR. Θα χρησιμοποιήσουμε D $f-f$, και έτσι είναι πολύ εύκολο να αποθηκεύσουμε δεδομένα, χωρίς να ανησυχούμε για τις εξισώσεις εισόδου των $f-f$.

Υπάρχουν πολλά είδη διαφορετικών καταχωρητών. Θα μελετήσουμε κάποιες εφαρμογές αυτών των ειδικών καταχωρητών. Στη συνέχεια, θα μελετήσουμε και την RAM, και θα δούμε, επίσης, πώς χρησιμοποιούνται οι καταχωρητές στο σχεδιασμό και το προγραμματισμό της Κεντρικής Μονάδας Επεξεργασίας (KME).



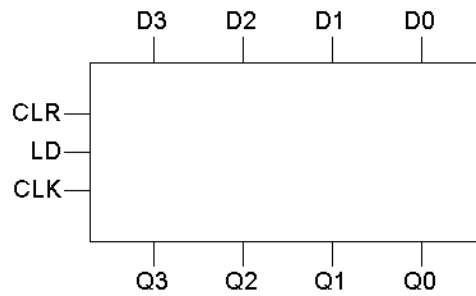
Σχήμα 9.2. Εσωτερική δομή ενός καταχωρητή 4 δυαδικών ψηφίων

Κατά τη λειτουργία της παράλληλης φόρτωσης, οι εισόδους D_3-D_0 αντιγράφονται στις εξόδους Q_3-Q_0 σε κάθε κύκλο ρολογιού. Για να αποθηκεύσουμε την τρέχουσα τιμή για περισσότερο από έναν κύκλο ρολογιού, θα πρέπει να προσθέσουμε ένα σήμα εισόδου LD (Load) στον καταχωρητή. (Σχήμα 9.3)

Εάν το $LD = 0$, ο καταχωρητής κρατάει τις τρέχουσες τιμές.

Εάν το $LD = 1$, ο καταχωρητής αποθηκεύει μια νέα τιμή, την οποία παίρνει από τις εισόδους D_3-D_0 .

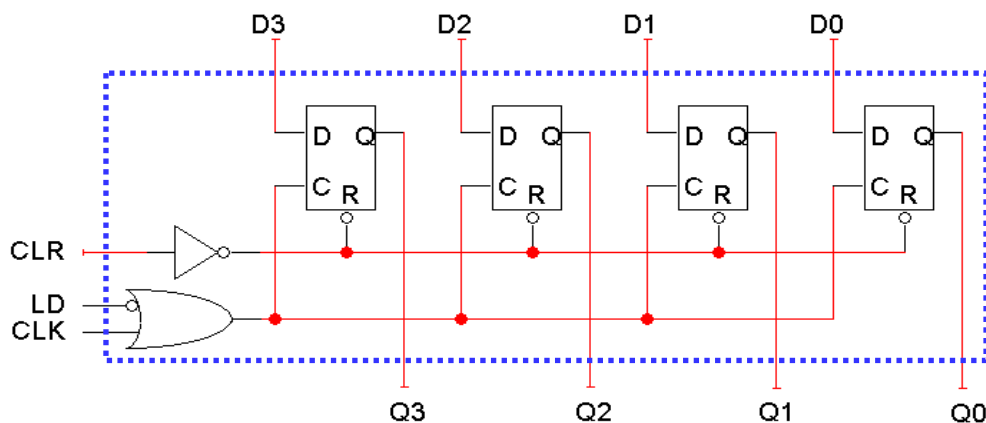
LD	$Q(n+1)$
0	$Q(n)$
1	D_3-D_0



Σχήμα 9.3. Δομικό διάγραμμα ενός καταχωρητή 4 δυαδικών ψηφίων με είσοδο LD

Θα μπορούσαμε να υλοποιήσουμε τη λειτουργία της φόρτωσης, μεταβάλλοντας την είσοδο CLK, όπως φαίνεται παρακάτω στο Σχήμα 9.4.

Όταν η $LD = 0$, λόγω της αναστροφής που υπάρχει πριν την πύλη, οι εισόδους C των $f-f$ διατηρούνται στο 1 ανεξάρτητα από το CLK. Δεν υπάρχει θετικό μέτωπο ρολογιού, και έτσι τα $f-f$ διατηρούν τις τρέχουσες τιμές τους. Όταν η $LD = 1$, λόγω της πύλης OR, η είσοδος CLK έχει κυρίαρχο ρόλο. Έτσι τα $f-f$ μπορούν να φορτώσουν μια καινούργια τιμή από τις εισόδους D_3-D_0 .



Σχήμα 9.4. Εσωτερική δομή ενός καταχωρητή 4 δυαδικών ψηφίων με είσοδο LD

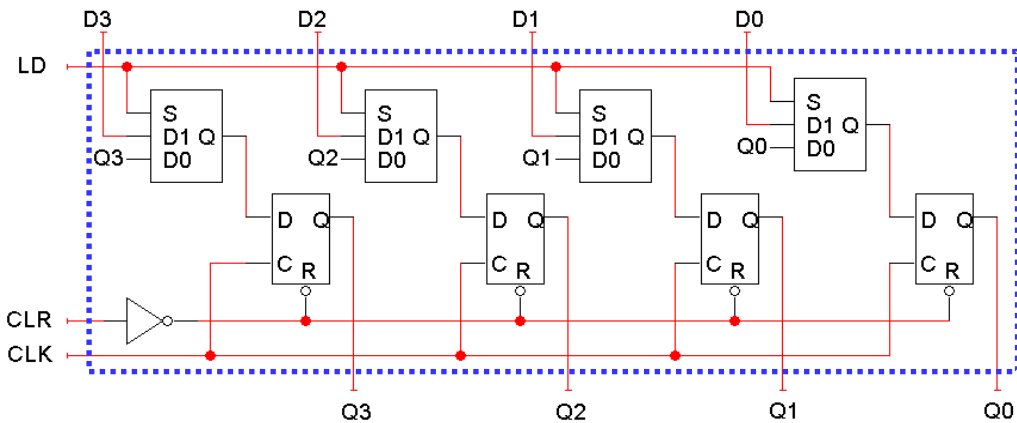
Η χρήση πυλών στο ρολόι, κατά κανόνα, αποφεύγεται. Όταν προστίθενται πύλες στο σήμα ρολογιού, έχουμε το λεγόμενο clock gating.

Υπάρχουν, όμως, προβλήματα χρονισμού. Εδώ, η είσοδος LD πρέπει να διατηρηθεί στο 1 για το ικανό χρονικό διάστημα (ένα κύκλο ρολογιού), και όχι περισσότερο.

Το ρολόι καθυστερεί λίγο, λόγω της πύλης OR. Σε πιο περίπλοκες περιπτώσεις, διαφορετικά $f-f$ στο σύστημα μπορούν να λάβουν το σήμα ρολογιού σε ελάχιστα διαφορετικούς χρόνους. Αυτή η «απόκλιση ρολογιού» μπορεί να οδηγήσει σε προβλήματα συγχρονισμού.

Μια καλύτερη παράλληλη είσοδος φόρτωσης είναι τροποποιώντας τις εισόδους D των $f-f$ και όχι το σήμα ρολογιού. (Σχήμα 9.5)

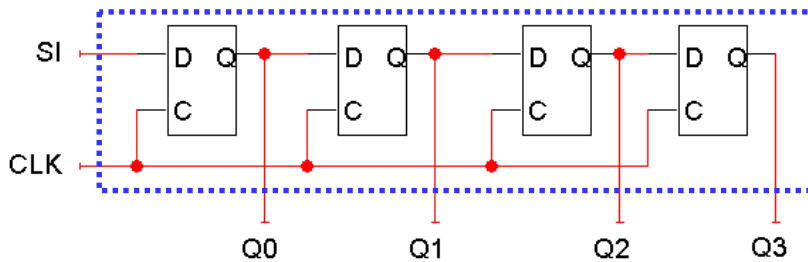
Όταν η LD = 0, οι εισόδους των $f-f$ είναι Q_3-Q_0 , και έτσι κάθε $f-f$ απλά διατηρεί την τρέχουσα τιμή του. Όταν η LD = 1, οι εισόδους των $f-f$ είναι D_3-D_0 , και αυτή η νέα τιμή «φορτώνεται» στον καταχωρητή.



Σχήμα 9.5. Τροποποιημένο διάγραμμα των εισόδων των $f-f$

9.1.1. Καταχωρητές Ολίσθησης

Ένας καταχωρητής ολίσθησης «ολισθαίνει» τις εξόδους του, μία φορά σε κάθε κύκλο ρολογιού. (Σχήμα 9.6)



Σχήμα 9.6 Καταχωρητής ολίσθησης

$$\begin{aligned} \text{όπου} \quad Q_0(n+1) &= SI, & Q_1(n+1) &= Q_0(n) \\ Q_2(n+1) &= Q_1(n), & Q_3(n+1) &= Q_2(n) \end{aligned}$$

Ο ακροδέκτης SI (Serial Input) είναι μια είσοδος που παρέχει ένα νέο bit, το οποίο θα ολισθήσει «μέσα» στον καταχωρητή. Για παράδειγμα, αν σε κάποιο θετικό μέτωπο ρολογιού έχουμε:

$$SI = 1 \text{ και } Q_0-Q_3 = 0110 \text{ τότε η επόμενη κατάσταση θα είναι: } Q_0-Q_3 = 1011$$

Η τρέχουσα τιμή Q_3 (0 στο παράδειγμα μας) θα χαθεί στον επόμενο κύκλο. Από το κύκλωμα του σχήματος 9.6 φαίνεται πως ο καταχωρητής ολισθαίνει προς τα «δεξιά», δηλαδή έχει «δεξιά διεύθυνση ολίσθησης», αλλά πραγματικά εξαρτάται από τη δική μας ερμηνεία για την προτεραιότητα των δυαδικών ψηφίων.

Παρόντα Q_0-Q_3	SI	Επόμενα Q_0-Q_3
ABCD	X	XABC

Αντίθετα, αν, θεωρήσουμε ότι το Q3 είναι το περισσότερο σημαντικό ψηφίο, τότε ο καταχωρητής ολισθαίνει προς την αντίθετη κατεύθυνση!

Παρόντα Q ₀ -Q ₃	SI	Επόμενα Q ₀ -Q ₃
DCBA	X	CBAX

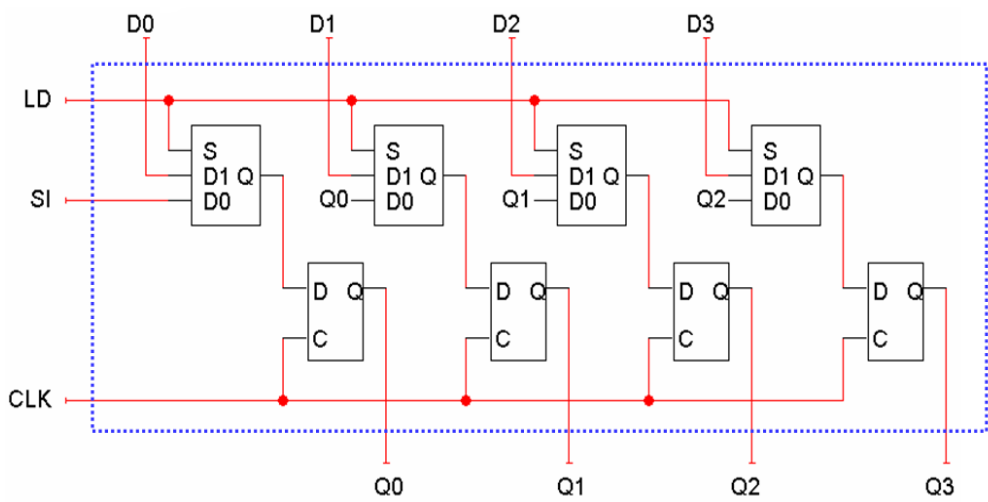
9.1.2. Καταχωρητές ολίσθησης με παράλληλη φόρτωση

Μπορούμε να προσθέσουμε μια παράλληλη είσοδο load, όπως κάναμε στους απλούς καταχωρητές.

Όταν η LD = 0, οι εισοδοί των f-f θα είναι SIQ₀Q₁Q₂, και έτσι ο καταχωρητής ολισθαίνει στο επόμενο θετικό μέτωπο ρολογιού.

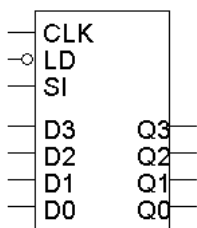
Όταν η LD = 1, οι εισοδοί των f-f είναι D₀-D₃, και η νέα τιμή φορτώνεται στον καταχωρητή ολίσθησης, στο επόμενο θετικό μέτωπο ρολογιού.

Η εσωτερική του δομή του καταχωρητή ολίσθησης τεσσάρων bits απεικονίζεται στο Σχήμα 9.7. (Να το συγκρίνετε με το κύκλωμα του Σχήματος 9.5).



Σχήμα 9.7. Καταχωρητής ολίσθησης τεσσάρων bits

Στο Σχήμα 9.8 αποδίδεται το δομικό διάγραμμα του σχήματος 9.7 με τη διαφορά ότι η είσοδος LD παρουσιάζεται σαν active-low. (παρατηρήστε την άρνηση πριν από την είσοδο LD).



Σχήμα 9.8. Δομικό διάγραμμα καταχωρητή ολίσθησης τεσσάρων bits

9.1.2.1 Εφαρμογή του καταχωρητή ολίσθησης

Μια εφαρμογή του καταχωρητή ολίσθησης είναι η μετατροπή μεταξύ «σειριακών» και «παράλληλων» δεδομένων. Οι υπολογιστές τυπικά λειτουργούν με ποσότητες πολλαπλών bit. Οι χαρακτήρες ASCII έχουν μήκος 8 bits. Οι ακέραιοι, οι αριθμοί κινητής υποδιαστολής απλής ακριβείας και τα εικονοστοιχεία της οθόνης έχουν μήκος μέχρι 32 bits.

Μερικές φορές είναι απαραίτητο να στείλουμε ή να λάβουμε δεδομένα σειριακά, ή ένα bit κάθε φορά. Κάποια τέτοια παραδείγματα περιλαμβάνουν: Συσκευές εισόδου, όπως τα πληκτρολόγια και τα ποντίκια, συσκευές εξόδου, όπως οι εκτυπωτές.

Επίσης περιλαμβάνουν οποιαδήποτε σειριακή πόρτα, USB ή σε διατάξεις Firewire οι οποίες μεταφέρουν τα δεδομένα σειριακά και, τέλος, τους πρόσφατους διακόπτες, από παράλληλη ATA σε σειριακή ATA, στους οδηγούς σκληρών δίσκων.

9.1.2.1.1 Σειριακή λήψη δεδομένων

Για να λάβουμε δεδομένα σειριακά χρησιμοποιούμε έναν καταχωρητή ολίσθησης: Η σειριακή συσκευή συνδέεται στην είσοδο SI του καταχωρητή. Οι έξοδοι Q3-Q0 του καταχωρητή ολίσθησης συνδέονται στον υπολογιστή.

Η σειριακή συσκευή (π.χ. ένα ποντίκι) στέλνει ένα bit δεδομένων σε κάθε κύκλο ρολογιού. Αυτά τα bits πηγαίνουν στην είσοδο SI του καταχωρητή ολίσθησης του Σχήματος 9.8. Μετά από τέσσερις κύκλους ρολογιού, ο καταχωρητής ολίσθησης θα περιέχει μία λέξη τεσσάρων bit. Ο υπολογιστής τότε διαβάζει, ταυτόχρονα και τα τέσσερα bits, από τις εξόδους Q3-Q0.

9.1.2.1.2 Σειριακή αποστολή δεδομένων

Για να στείλουμε δεδομένα σειριακά με έναν καταχωρητή ολίσθησης, εκτελούμε την αντίστροφη διαδικασία: Η ΚΜΕ είναι συνδεδεμένη στις εισόδους D του καταχωρητή. Η έξοδος του ολισθητή (μόνο η Q3 σε αυτή την περίπτωση) συνδέεται με τη σειριακή συσκευή (π.χ. ένας εκτυπωτής).

Ο υπολογιστής πρώτα αποθηκεύει μια λέξη τεσσάρων bit στον καταχωρητή, σε έναν κύκλο ρολογιού. (είσοδοι D3- D0 του καταχωρητή).

Η σειριακή συσκευή τότε μπορεί να διαβάσει την έξοδο του ολισθητή (Q3). Στην έξοδο Q3 εμφανίζεται ένα bit σε κάθε κύκλο ρολογιού. Μετά από τέσσερις κύκλους, θα έχει αποσταλεί ολόκληρη η λέξη των 4-bit.

9.1.3. Οι καταχωρητές στο σύγχρονο υλικό

Οι καταχωρητές αποθηκεύουν δεδομένα στη ΚΜΕ και επίσης χρησιμοποιούνται, για να παρέχουν τιμές στην Αριθμητική και Λογική Μονάδα. Τέλος χρησιμοποιούνται, για να αποθηκεύουν τα αποτελέσματα.

Αφού μπορούμε να χρησιμοποιήσουμε καταχωρητές, γιατί ασχολούμαστε ακόμα με τη RAM; Ασχολούμαστε γιατί απλά οι καταχωρητές είναι ακριβοί!

Οι καταχωρητές καταλαμβάνουν τον πιο ακριβό χώρο σε μία ψηφίδα(chip) - τον πυρήνα. Οι μνήμες L1 και L2 είναι πολύ γρήγορες μνήμες τυχαίας προσπέλασης (RAM), – αλλά όχι τόσο γρήγορες όσο οι καταχωρητές.

Στον πίνακα 9.1 παραθέτουμε μερικούς τύπους κεντρικών μονάδων επεξεργασίας με τα αντίστοιχα μεγέθη καθώς και τις cache μνήμες. (Μικροαρχιτεκτονικές MIPS, 2015)

CPU	GPR's	Χρονολογία	Συχνότητα MHz	Μέγεθος bits	L1 Cache (KB)	L2 Cache (KB)
Pentium 4	8	2000	1300	32	8	512
Athlon XP	8	2000	500	32	64	512
Athlon 64	16	2003	1000	64	64	1024
Power PC 970	32	2002	450	64	64	512
Itanium II	128	2006	900	64	16	256
MIPS IV R14000	32	2001	500	64	32	512-16000 εξωτερική
MIPS 32 roAptiv	32	2012	1500	64	32	Μέχρι 8000 εσωτερική
MIPS 32 Release 6		2014	1000	64	64	

Πίνακας 9.1. Τύποι κεντρικών μονάδων επεξεργασίας

9.2. Μνήμες

Όλα τα ακολουθιακά κυκλώματα εξαρτώνται από την παρουσία μνήμης. Ένα f - f μπορεί να αποθηκεύσει ένα bit πληροφορίας. Ένας καταχωρητής μπορεί να αποθηκεύσει μια λέξη, των 32-64 bits.

Η μνήμη τυχαίας προσπέλασης, (Random Access Memory ή RAM), μας επιτρέπει να αποθηκεύσουμε ακόμη μεγαλύτερα μεγέθη δεδομένων. Θα μελετήσουμε λοιπόν τη βασική διεπαφή προς τη μνήμη και τον

τρόπο υλοποίησης της ψηφίδας της στατικής RAM, ιεραρχικά. Αυτό είναι το τελευταίο τμήμα που μας χρειάζεται, για να συναρμολογήσουμε έναν υπολογιστή! (Katz, 2005; Κοσσίδης & Γιαννακόπουλος, 2006; Wakerly, 2006; Balabanian, & Carlson 2007; Godse, & Godse, 2010; Tocci, et al., 2010; Mano, & Ciletti, 2014)

9.2.1 Η μνήμη RAM

Η RAM, παρέχει μεγάλο χώρο για προσωρινή αποθήκευση δεδομένων σε ένα υπολογιστικό σύστημα.

Ας θυμηθούμε τις βασικές δυνατότητες μιας μνήμης: Μία μνήμη θα πρέπει να μπορεί να αποθηκεύσει μια τιμή, η οποία θα πρέπει να είναι δυνατόν να διαβαστεί. Τέλος πρέπει να μπορούμε να αλλάξουμε την ήδη αποθηκευμένη τιμή.

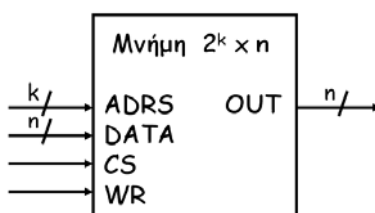
Μια RAM είναι παρόμοια, εκτός του ότι μπορεί να αποθηκεύσει πολλές τιμές. Μια διεύθυνση θα καθορίζει την «περιοχή» της μνήμης για την οποία ενδιαφερόμαστε. Κάθε «περιοχή» είναι μία λέξη πολλαπλών bits (π.χ. 32 bits).

Προσδιορίζουμε λοιπόν τις ιδιότητες της μνήμης ως εξής: Η RAM θα πρέπει να μπορεί να:

- αποθηκεύσει πολλές λέξεις, μία σε κάθε διεύθυνση,
- διαβάσει τη λέξη που αποθηκεύτηκε σε συγκεκριμένη διεύθυνση,
- αλλάξει τη λέξη που αποθηκεύτηκε σε συγκεκριμένη διεύθυνση.

9.2.1.1 Δομικό διάγραμμα της RAM

Στο Σχήμα 9.9, που ακολουθεί, απεικονίζεται το δομικό διάγραμμα της RAM, το οποίο παρουσιάζει τη βασική διεπαφή (interface) προς την RAM.



Σχήμα 9.9. Δομικό διάγραμμα της μνήμης RAM

Το CS (Chip Select) (Μονάδα επιλογής της ψηφίδας), ενεργοποιεί ή απενεργοποιεί την RAM. Το ADRS προσδιορίζει τη διεύθυνση, ή τη θέση, από την οποία θέλουμε να διαβάσουμε, ή τη θέση στην οποία θέλουμε να γράψουμε.

Το WR επιλέγει το διάβασμα από τη μνήμη ή το γράψιμο στη μνήμη. Για να διαβάσουμε από τη μνήμη, το WR πρέπει να έχει τιμή 0. Το OUT θα είναι η τιμή των n-bit που είχε αποθηκευτεί στον ADRS.

Για να γράψουμε στη μνήμη, θέτουμε το WR = 1. Το DATA θα είναι η τιμή (n-bit) που θα αποθηκευτεί στη μνήμη. Η διεπαφή αυτή, όπως θα δούμε, μας διευκολύνει στην ένωση πολλών RAM.

Με βάση τα χαρακτηριστικά της μνήμης, έχουμε τον πίνακα λειτουργίας της RAM (πίνακας 9.2).

CS	WR	Λειτουργία Μνήμης
0	x	Καμία
1	0	Ανάγνωση Λέξης
1	1	Εγγραφή Λέξης

Πίνακας 9.2. Πίνακας λειτουργίας της μνήμης

9.2.1.2 Απεικόνιση της μνήμης

Μπορούμε να σκεφτούμε τη μνήμη ενός υπολογιστή σαν ένα μεγάλο πίνακα δεδομένων. Η διεύθυνση θα προσδιορίζεται από το δείκτη του πίνακα. (Σχήμα 9.9).

Μπορούμε να διαβάσουμε ή να τροποποιήσουμε τα δεδομένα οποιασδήποτε διεύθυνσης μνήμης, ακριβώς όπως μπορούμε να διαβάσουμε ή να τροποποιήσουμε τα περιεχόμενα σε οποιονδήποτε δοθέντα δείκτη ενός πίνακα.

Όταν δουλεύουμε με pointers στη C ή τη C++, τότε έχουμε ήδη εργαστεί με διευθύνσεις μνήμης.

Δεδομένα									
Διεύθυνση	00000000	00000001	00000002				FFFFFFFD	FFFFFFFE	FFFFFFF

Σχήμα 9.10. Απεικόνιση της μνήμης

9.2.1.3 Τα μεγέθη της μνήμης

Θα αναφερόμαστε στα μεγέθη της μνήμης με τον όρο $2^k \times n$. Υπάρχουν k γραμμές διευθύνσεων, που μπορούν να προσδιορίσουν μια από τις 2^k διευθύνσεις. Κάθε διεύθυνση περιέχει μια λέξη των n -bit.

Για παράδειγμα, μία $2^{24} \times 16$ μνήμη RAM περιέχει $2^{24} = 16\text{M}$ λέξεις, των 16 bits. Η RAM θα χρειαστεί 24 γραμμές διευθύνσεων. Η συνολική χωρητικότητα της μνήμης είναι $2^{24} \times 16 = 2^{28}$ bits.

Το μέγεθος της μνήμης, συνήθως, προσδιορίζεται από τον αριθμό των bytes (8 bits) που χωράει η μνήμη. Μια μνήμη των 2^{28} -bit σύμφωνα με τα προηγούμενα μεταφράζεται ως εξής: 2^{28} bits / 8 bits κάθε byte = 2^{25} bytes.

Με βάση τις κύριες συντομογραφίες που αποδίδονται στον πίνακα 9.3, αυτό το ποσό ισοδυναμεί με 32 MB.

Προσοχή! Αναφέρουμε ότι το μέγεθος της RAM μετριέται σε μονάδες με βάση το 2, ενώ το μέγεθος ενός σκληρού δίσκου μετριέται σε μονάδες με βάση το 10. Εμείς θα ασχοληθούμε μόνο με μονάδες που έχουν σαν βάση το 2.

	Πρόθεμα	Βάση 2	Βάση 10
K	Kilo	$2^{10} = 1,024$	$10^3 = 1,000$
M	Mega	$2^{20} = 1,048,576$	$10^6 = 1,000,000$
G	Giga	$2^{30} = 1,073,741,824$	$10^9 = 1,000,000,000$

Πίνακας 9.3. Κύριες συντομογραφίες

Μερικές τυπικές χωρητικότητες μνήμης είναι για τα PC συνήθως από 512MB–1GB RAM. Τα PDAs, αντίστοιχα, έχουν 8–64MB μνήμης. Οι ψηφιακές κάμερες και τα MP3 μπορούν να έχουν μνήμη από 32MB και πάνω.

Πολλά λειτουργικά συστήματα χρησιμοποιούν την εικονική μνήμη, η οποία κάνει τις μνήμες να φαίνονται μεγαλύτερες από ό,τι πραγματικά είναι. Τα περισσότερα συστήματα επιτρέπουν μέχρι 32-bit διευθύνσεις. Αυτό έχει ως αποτέλεσμα το 2^{32} διαφορετικές διευθύνσεις. Με μέγεθος δεδομένων ενός byte, το αποτέλεσμα είναι μια μνήμη 4GB!. Τα λειτουργικά συστήματα χρησιμοποιούν ένα μέρος του σκληρού δίσκου για αντικατάσταση της «πραγματικής» μνήμη.

9.2.1.4 Ανάγνωση- Εγγραφή στη μνήμη RAM

Για να διαβάσουμε, από την RAM, το κύκλωμα ελέγχου (Σχήμα 9.10) θα πρέπει:

- Να ενεργοποιεί την ψηφίδα εξασφαλίζοντας ότι CS = 1.
- Να επιλέγει τη λειτουργία του διαβάσματος, θέτοντας το WR = 0.
- Να στέλνει την επιθυμητή διεύθυνση στην είσοδο του ADRS.

Έπειτα από λίγο, τα περιεχόμενα αυτής της διεύθυνσης εμφανίζονται στην έξοδο OUT. Πρέπει να σημειώσουμε πως η είσοδος DATA δε χρησιμοποιείται για τις λειτουργίες της ανάγνωσης.

Για να γράψουμε στην RAM, πρέπει το κύκλωμα ελέγχου να:

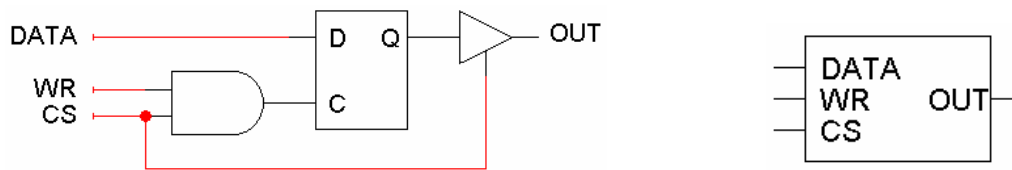
- Ενεργοποιεί την ψηφίδα θέτοντας CS = 1.
- Επιλέγει τη λειτουργία της εγγραφής, θέτοντας WR = 1.
- Στέλνει την επιθυμητή διεύθυνση στην είσοδο ADRS.
- Στέλνει την προς αποθήκευση λέξη στην είσοδο DATA.

Η έξοδος OUT δε χρησιμοποιείται στη λειτουργία της εγγραφής στη μνήμη.

9.2.1.5 Τύποι στατικής μνήμη

Πώς μπορούμε να κατασκευάσουμε μία ψηφίδα μνήμης; Υπάρχουν πολλά, διαφορετικά είδη μνήμης RAM. Θα αρχίσουμε, μελετώντας τη στατική μνήμη, που χρησιμοποιείται συχνότερα στις cache και στις κάρτες video.

Η στατική μνήμη κατασκευάζεται, χρησιμοποιώντας δύο μόνο πύλες NAND ή δύο πύλες NOR (latch) για κάθε αποθηκευόμενο bit. Συνήθως χρησιμοποιούμε τα latch, και όχι τα $f-f$, καθώς ένα $f-f$ απαιτεί το λιγότερο, το διπλάσιο υλικό για την κατασκευή του. Γενικά το μικρότερο είναι γρηγορότερο, φθηνότερο και απαιτεί λιγότερη ισχύ. Το τίμημα είναι πως είναι δύσκολο να πετύχει κάποιος τέλει συγχρονισμό.



Σχήμα 9.11. Διαγράμματα κυττάρου μνήμης RAM(αναλυτικό και μπλόκ)

Αργότερα θα αναφερθούμε και στη δυναμική μνήμη, η οποία διαμορφώνει το εσωτερικό της κύριας μνήμης ενός υπολογιστή

Αρχικά θα χρησιμοποιήσουμε ένα μανταλωτή (latch) για να αποθηκεύσουμε ένα δυαδικό ψηφίο. Ένα κύτταρο (cell) RAM ενός bit απεικονίζεται στο Σχήμα 9.11.

Καθώς είναι μια μνήμη του ενός bit, η είσοδος ADRS του σχήματος 9.9. δεν χρειάζεται.

Για την **εγγραφή** στο κύτταρο της RAM:

Όταν το CS = 1 και το WR = 1, η είσοδος ελέγχου του latch θα γίνει 1.

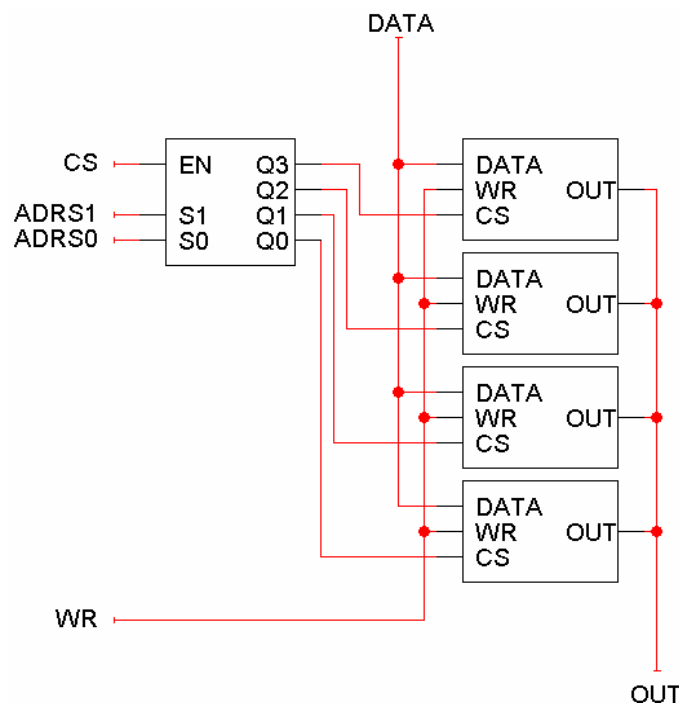
Τότε η είσοδος DATA θα αποθηκευτεί στο D latch.

Για την **ανάγνωση** από ένα κύτταρο RAM και διατήρηση του τρέχοντος περιεχομένου:

Όταν το CS = 0 ή όταν το WR = 0, η είσοδος ελέγχου του latch είναι επίσης 0, έτσι ώστε το latch απλά διατηρεί την παρούσα κατάσταση του.

Το τρέχον περιεχόμενο του latch θα εμφανιστεί στην έξοδο OUT.

Υλοποίηση της 4x1 RAM



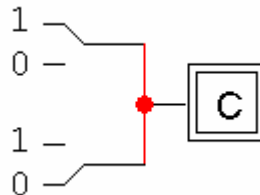
Σχήμα 9.12. Διάγραμμα μνήμης RAM 4x1

Μπορούμε να χρησιμοποιήσουμε το μπλόκ διάγραμμα της μνήμης, για να φτιάξουμε μια 4x1 RAM.(Σχήμα 9.12). Δεδομένου ότι υπάρχουν τέσσερις λέξεις, η ADRS θα είναι των δύο bit. Κάθε λέξη είναι του ενός bit, και έτσι η είσοδος DATA και η έξοδος OUT θα είναι, επίσης, του ενός bit.

Η επιλογή της λέξης γίνεται με έναν αποκωδικοποιητή συνδεδεμένο στις εισόδους CS των κυττάρων RAM. Μόνο ένα κύτταρο μπορεί να κάνει ανάγνωση ή εγγραφή κάθε φορά. Ας σημειώσουμε πως οι εξόδοι είναι συνδεδεμένοι με μια μοναδική γραμμή εξόδου!

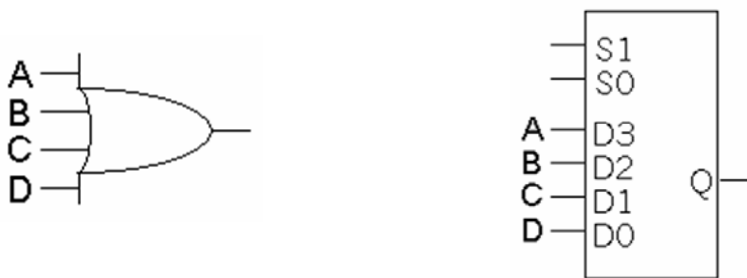
Διασύνδεση εξόδων

Πρακτικά, δεν είναι καλό να συνδέουμε μεταξύ τους τις εξόδους. Αν οι εξόδοι έχουν διαφορετικές τιμές (Σχήμα 9.13), τότε προκύπτει αντίφαση C(conflict).



Σχήμα 9.13. Περίπτωση αντίφασης

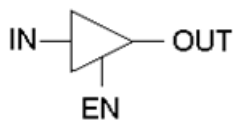
Ο σωστός τρόπος είναι να «ενώνουμε» τις εξόδους είτε χρησιμοποιώντας πύλες OR είτε πολυπλέκτες. (Σχήμα 9.14). Αυτό, βέβαια, μπορεί να κοστίζει αρκετά, καθώς θα έχει πολλά καλώδια σύνδεσης και πύλες με μεγάλο πλήθος εισόδων.



Σχήμα 9.14. Διασύνδεση των εξόδων με πύλες OR ή πολυπλέκτες

Τριγωνικό διάγραμμα

Το τριγωνικό διάγραμμα (Σχήμα 9.15) προσδιορίζει έναν απομονωτή τριών καταστάσεων. Αντίθετα με τις κανονικές λογικές πύλες, η έξοδος μπορεί να είναι μια από τρεις διαφορετικές πιθανές καταστάσεις, όπως φαίνεται στον πίνακα του σχήματος 9.15 που ακολουθεί.



EN	IN	OUT
0	x	Αποσυνδεδεμένο
1	0	0
1	1	1

Σχήμα 9.15. Τριγωνικό Διάγραμμα

Η έννοια «Αποσυνδεδεμένο» σημαίνει πως δεν εμφανίζεται τίποτα στην έξοδο. Στην περίπτωση αυτή είναι ασφαλές να συνδέσουμε την έξοδο OUT με κάποιο άλλο σήμα εξόδου.

Η τιμή «αποσύνδεση» μερικές φορές ονομάζεται υψηλή σύνθετη αντίσταση ή Hi-Z.

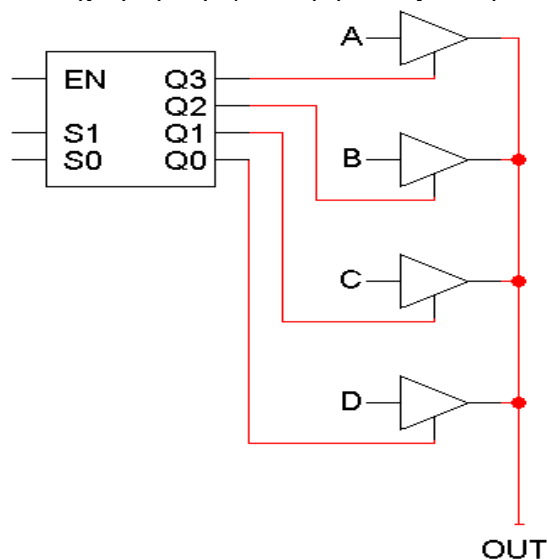
Διασύνδεση απομονωτών τριών καταστάσεων

Ο πιο εύκολος τρόπος, να πετύχουμε τη διασύνδεση απομονωτών τριών καταστάσεων είναι χρησιμοποιώντας έναν αποκωδικοποιητή, (Σχήμα 9.16).

Αν ο αποκωδικοποιητής είναι απενεργοποιημένος, τότε όλοι οι απομονωτές τριών καταστάσεων θα εμφανίζονται ως «Αποσυνδεδεμένοι», και η έξοδος OUT θα εμφανίζεται, επίσης, «Αποσυνδεδεμένη».

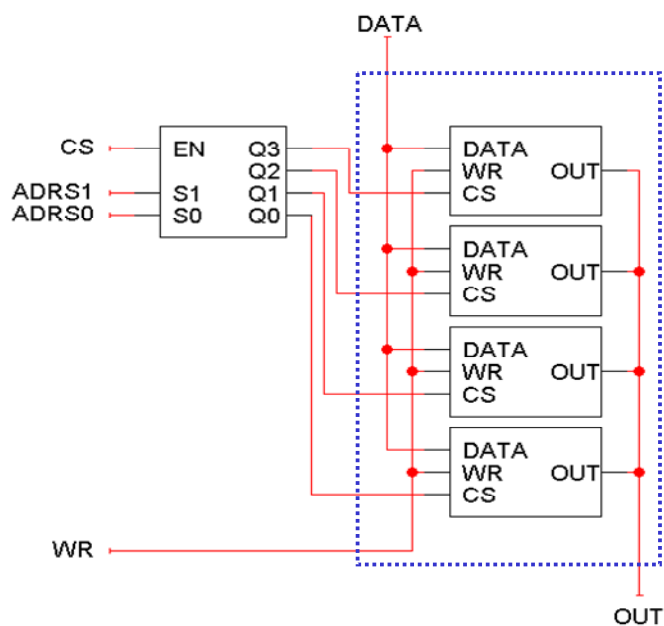
Αν ο αποκωδικοποιητής είναι ενεργοποιημένος, τότε μόνο μια από τις εξόδους θα είναι αληθής, και, συνεπώς, μόνο ένας από τους απομονωτές θα είναι συνδεδεμένος και θα παρουσιάζει μια έξοδο.

Το τελικό αποτέλεσμα είναι πως μπορούμε να μειώσουμε το κόστος από τις δαπάνες των καλωδίων και των πυλών. Επίσης μπορούμε να έχουμε μια μεγαλύτερη ευελιξία στη διασύνδεση των κυκλωμάτων.



Σχήμα 9.16. Διασύνδεση απομονωτών τριών καταστάσεων με αποκωδικοποιητή

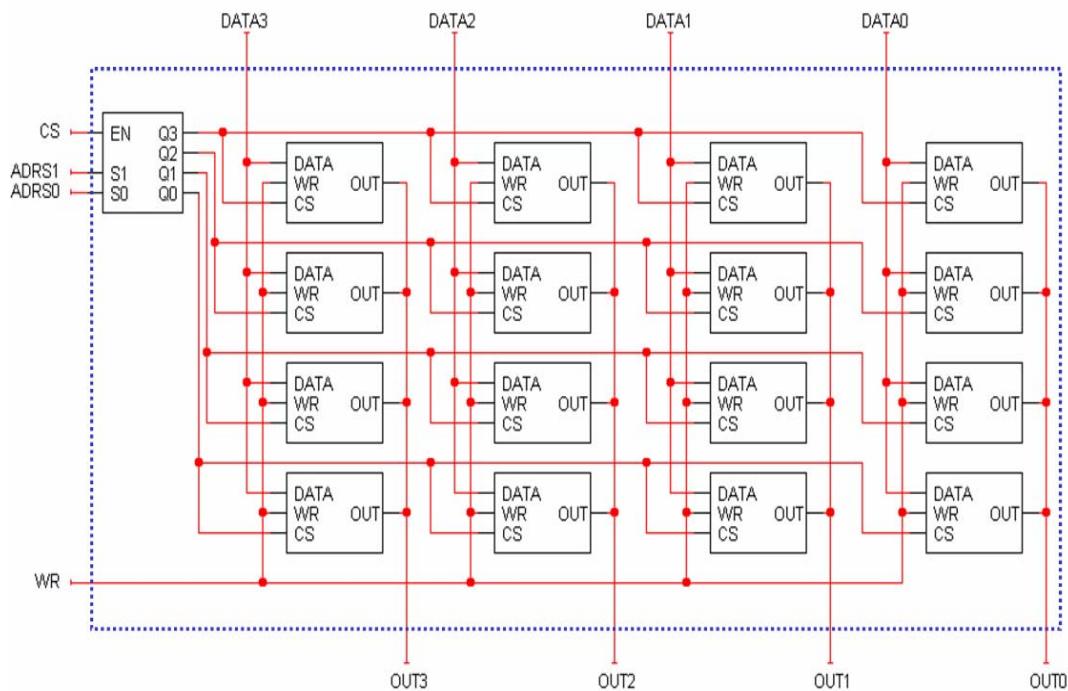
Στο Σχήμα 9.17 απεικονίζεται μια 4x1 RAM. Μπορούμε εύκολα να επεκτείνουμε τη μνήμη ώστε να έχει περισσότερα bit για κάθε λέξη, π.χ. να είναι μια 4x4 RAM; Βεβαίως, πολύ απλά!, και συγκεκριμένα, αντιγράφοντας το περιεχόμενο του δεξιού πλαισίου του σχήματος!



Σχήμα 9.17. Διάγραμμα 4x1 RAM

9.2.1.6 RAM 4x4

Στο Σχήμα 9.18 απεικονίζεται μία μνήμη RAM 4x4. Οι εισόδοι DATA και οι έξοδοι OUT έχουν μήκος 4-bits η καθεμία, και έτσι μπορούμε να διαβάσουμε και να γράψουμε λέξεις των 4-bits.

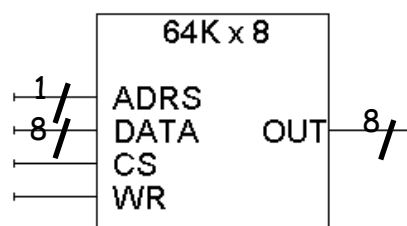


Σχήμα 9.18. Διάγραμμα μνήμης RAM 4x4

9.2.1.7 Κατασκευή μεγαλύτερων μνημών RAM

Μπορούμε να χρησιμοποιήσουμε μικρές RAM σαν δομικά στοιχεία, για να φτιάξουμε μεγαλύτερες μνήμες, ακολουθώντας τις ίδιες αρχές, με αυτές που χρησιμοποιήσαμε στα προηγούμενα παραδείγματα.

Για παράδειγμα, ας υποθέσουμε πως έχουμε μερικές 64Kx8 RAM (Σχήμα 9.19), για να ξεκινήσουμε: $64K = 2^6 \times 2^{10} = 2^{16}$, έτσι θα έχουμε 16 γραμμές διευθύνσεων. Επίσης θα έχουμε 8 γραμμές δεδομένων.

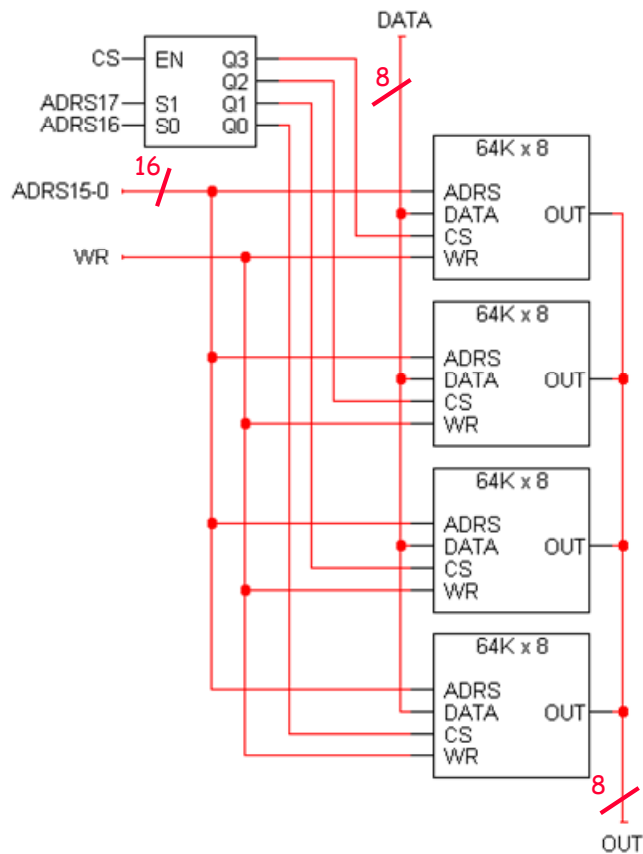


Σχήμα 9.19. Δομικό Διάγραμμα Μνήμης RAM 64Kx8

Μπορούμε να τοποθετήσουμε μαζί τέσσερις ψηφίδες(chip) 64K x 8, για να φτιάξουμε μια μεγαλύτερη μνήμη, έστω 256K x 8. Για τις 256K λέξεις, θα χρειαστούμε 18 γραμμές διευθύνσεων. (Σχήμα 9.20)

Οι δυο περισσότερο σημαντικές γραμμές διευθύνσεων οδηγούνται στον αποκωδικοποιητή, ο οποίος και επιλέγει μία από τα τέσσερις ψηφίδες των 64K x 8 RAM.

Οι άλλες 16 γραμμές διευθύνσεων διαμοιράζονται μεταξύ των 64K x 8 ψηφίδες, οι οποίες, επίσης διαμοιράζονται τις εισόδους WR και DATA. Αυτό προϋποθέτει, βεβαίως, πώς οι 64K x 8 ψηφίδες έχουν έξοδο τριών καταστάσεων.



Σχήμα 9.20. Διάγραμμα μνήμης 256K x 8 RAM

9.2.1.8 Ανάλυση της 256K x 8 RAM

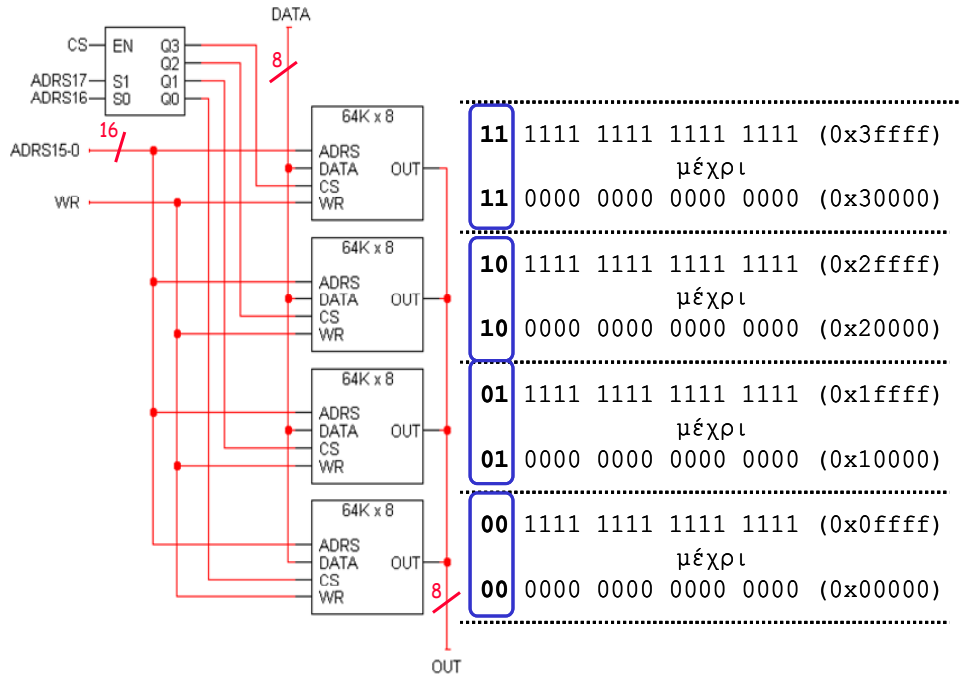
Υπάρχουν 256K λέξεις μνήμης, που κατανέμονται στις τέσσερις μικρότερες ψηφίδες της 64K x 8 RAM. (Βλέπε Σχήμα 9.20)

Όταν τα δυο περισσότερο σημαντικά ψηφία των γραμμών διευθύνσεων είναι (00), τότε επιλέγεται η RAM που βρίσκεται στο κάτω τμήμα. Αποθηκεύει τα δεδομένα των πρώτων 64K διευθύνσεων.

Η επόμενη ψηφίδα προς τα επάνω, ενεργοποιείται, όταν τα δυο περισσότερο σημαντικά ψηφία των γραμμών διευθύνσεων είναι (01), και διατηρεί τα δεδομένα των επόμενων 64K διευθύνσεων. Αντίστοιχα, η τρίτη ψηφίδα διατηρεί τα δεδομένα των επομένων 64K (10), η δε τελευταία ψηφίδα περιέχει τα δεδομένα των τελευταίων 64K διευθύνσεων (11).

9.2.1.9 Εύρος διευθύνσεων

Στο Σχήμα 9.21, που ακολουθεί απεικονίζονται οι διευθύνσεις της μνήμης του προηγούμενου Σχήμα 9.20. Τα δύο πρώτα δυαδικά ψηφία προσδιορίζουν και τον πρώτο δεκαδικό αριθμό των διευθύνσεων.

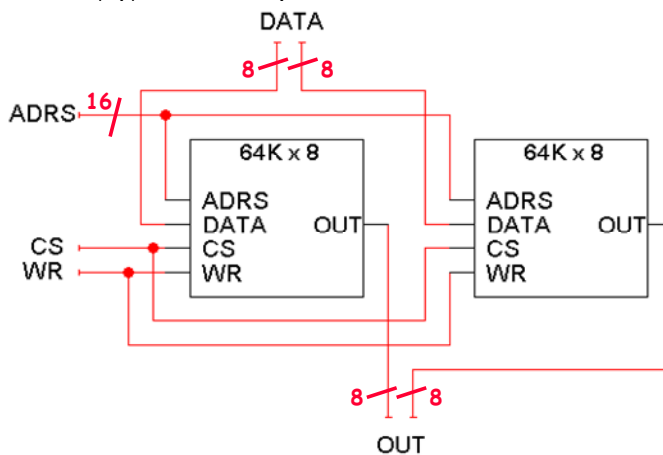


Σχήμα 9.21. Απεικόνιση των διευθύνσεων μιας 256K x 8 μνήμης RAM

9.2.1.10 Κατασκευή «ευρύτερης» μνήμης

Μπορούμε, επίσης, να συνδυάσουμε μικρότερες ψηφίδες, για να φτιάξουμε ευρύτερες μνήμες, δηλαδή μνήμες με τον ίδιο αριθμό γραμμών διευθύνσεων αλλά περισσότερα ψηφία για κάθε λέξη.

Στο Σχήμα 9.22 απεικονίζεται μια 64Kx16 RAM, που έχει δημιουργηθεί από δύο ψηφίδες 64Kx 8. Η αριστερή ψηφίδα περιέχει τα 8 περισσότερο σημαντικά ψηφία των δεδομένων. Η δεξιά ψηφίδα περιέχει τα 8 λιγότερο σημαντικά ψηφία των δεδομένων.



Σχήμα 9.22. Διάγραμμα μιας 64K x 16 μνήμης RAM

9.2.2. Μνήμες RAM ειδικού τύπου

Ήδη δείξαμε πώς μπορούμε να κατασκευάσουμε αυθαίρετα μεγάλες στατικές μνήμες, από κύτταρα RAM του ενός bit. Θα μελετήσουμε και μνήμες άλλων τύπων. Η δυναμική RAM χρησιμοποιείται για το εσωτερικό μιας μνήμης υπολογιστή. (Katz, 2005; Predko, 2005; Κοσσίδας & Γιαννακόπουλος, 2006; Maxfield, 2009; Godse, & Godse, 2010; Tocci, et al., 2010; Mano, & Ciletti, 2014)

Οι μόνο αναγνώσιμες μνήμες (ROM) και τα PLA είναι δύο «συσκευές προγραμματιζόμενης λογικής», οι οποίες μπορούν και να θεωρηθούν ως μνήμες ειδικού τύπου.

9.2.2.1 Δυναμική Μνήμη RAM (DRAM)

Η δυναμική μνήμη κατασκευάζεται με πυκνωτές. Ένα αποθηκευμένο φορτίο στον πυκνωτή αντιπροσωπεύει το λογικό 1, ενώ το λογικό 0 αντιπροσωπεύεται με την απουσία φορτίου. Όπως είναι γνωστό, οι πυκνωτές χάνουν το φορτίο τους μετά από μερικά msec. Η μνήμη λοιπόν χρειάζεται διαρκή ανανέωση για να επαναφορτιστούν οι πυκνωτές. (Γι' αυτό ονομάζεται και «δυναμική»).

Η δυναμική RAM τείνει να είναι μικρότερη σε όγκο από τη στατική RAM. Ένα bit δεδομένων μπορεί να αποθηκευτεί σε ένα πυκνωτή και σε ένα τρανζίστορ, ενώ το κύτταρο της στατικής μνήμης απαιτεί 4 – 6 τρανζίστορ.

Αυτό σημαίνει ότι η δυναμική RAM είναι φθηνότερη και πυκνότερη - που σημαίνει ότι περισσότερα bit μπορούν να αποθηκευτούν στην ίδια επιφάνεια.

SDRAM (Synchronous Dynamic Random Access Memory)

Η Σύγχρονη DRAM, ή SDRAM, είναι πλέον ένας από τους πιο κοινούς τύπους μνήμης για τα PC. Οι ψηφίδες της μνήμης οργανώνονται σε «ενότητες» που συνδέονται με την CPU μέσω ενός διαύλου των 64-bit (8-byte).

Η ταχύτητα μετριέται σε MHZ: Οι PC66, PC100 και PC133 μνήμες τρέχουν στα 66MHZ, 100MHZ και 133MHZ αντίστοιχα.

Το εύρος ζώνης της μνήμης μπορεί να υπολογιστεί, πολλαπλασιάζοντας τον αριθμό των μεταφορών ανά δευτερόλεπτο με το μέγεθος της κάθε μεταφοράς.

Η PC100 μπορεί να μεταφέρει μέχρι 800MB το δευτερόλεπτο (100MHZ x 8 bytes/κύκλο), ενώ η PC133 μπορεί να μεταφέρει πάνω από 1 GB το δευτερόλεπτο (133MHZ x 8 bytes/κύκλο = 1064 MB/s).

DDR-RAM (Double Data Rate Random Access Memory)

Ένα νέο είδος μνήμης είναι η διπλού ρυθμού δεδομένων RAM, ή DDR-RAM. Είναι παρόμοια με την απλή SDRAM, με τη διαφορά, όμως, ότι τα δεδομένα μπορούν να μεταφερθούν και κατά το θετικό και κατά το αρνητικό μέτωπο του ρολογιού. Για τους διαύλους των 100-133MHZ, η πραγματική ταχύτητα της μνήμης φαίνεται να είναι 200-266MHZ.

Η μνήμη αυτή καλείται PC1600 και PC2100 RAM, διότι:

200MHZ x 8 byte/κύκλο = 1600MB/s και αντίστοιχα 266MHZ x 8 bytes/κύκλο = 2100MB/s. Τέλος η νέα PC2600 RAM έχει 2x 166MHZ x 8 byte/κύκλο = 2656MB/s.

Η DDR-RAM παρουσιάζει μικρότερη κατανάλωση ισχύος, χρησιμοποιώντας 2.5V αντί για 3.3V που χρειάζεται η SDRAM. Αυτό την κάνει κατάλληλη για τα notebooks και για άλλες κινητές συσκευές.

RDRAM (Rambus Dynamic Random Access Memory)

Ένας άλλος νέος τύπος μνήμης ονομάζεται RDRAM, και χρησιμοποιείται στο Playstation 2, όπως, επίσης, και σε μερικούς υπολογιστές Pentium 4. Ο δίαυλος δεδομένων είναι μόνο 16 bit, αλλά η μνήμη τρέχει στα 400MHZ, και τα δεδομένα μπορούν να μεταφερθούν τόσο στο θετικό όσο και στο αρνητικό μέτωπο του ρολογιού. Αυτό σημαίνει ότι έχουν μέγιστο ρυθμό μετάδοσης 1.6GB το δευτερόλεπτο.

Μπορούμε, επίσης, να ενώσουμε δύο «κανάλια» μνήμης, και να έχουμε σαν αποτέλεσμα εύρος ζώνης μνήμης μέχρι 3.2GB/s. Σήμερα αρκετοί κατασκευαστές δουλεύουν μια εναλλακτική αρχιτεκτονική μνήμης την SLDRAM (SyncLink DRAM).

9.2.3. Σύγκριση της δυναμικής και της στατικής μνήμης

Η δυναμική μνήμη χρησιμοποιείται ως η κύρια μνήμη του υπολογιστή, καθώς είναι πιο φθηνή και μπορούμε σε μικρό χώρο να αποθηκεύσουμε περισσότερα δεδομένα. Σήμερα μπορούμε να αγοράσουμε 512MB μνήμης έναντι 50€. Μπορούμε, επίσης, να βάλουμε σε ένα σύστημα, μνήμη 1.5GB ή και ακόμα περισσότερη.

Το μειονέκτημα της δυναμικής μνήμης είναι η ταχύτητά της. Ο ρυθμός μεταφοράς μπορεί να φτάσει μέχρι και τα 800MHZ, που είναι πολύ πιο αργή από τον επεξεργαστή. Πρέπει, επίσης, να λάβουμε υπόψη μας την καθυστέρηση, ή το χρόνο που χρειάζονται τα δεδομένα, για να φτάσουν από την RAM στον επεξεργαστή.

Τα πραγματικά συστήματα αυξάνουν τη δυναμική μνήμη με μικρά αλλά γρήγορα τεμάχια στατικής μνήμης που ονομάζονται cache. Η τυπική cache του επεξεργαστή έχει μέγεθος από 128KB ως 320KB. Η μνήμη αυτή είναι μικρή, συγκρινόμενη με τα 128MB της κύριας μνήμης, αλλά είναι αρκετή, για να αυξήσει σημαντικά την τελική ταχύτητα ενός υπολογιστή.

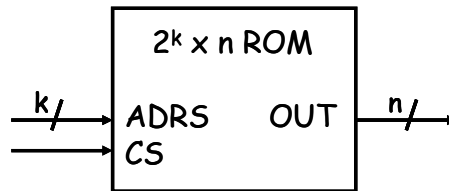
9.2.4. Η μνήμη μόνο ανάγνωσης (ROM)

Η μνήμη ROM, είναι ένα ειδικό είδος μνήμης, της οποίας τα περιεχόμενα δεν μπορούν να αλλάζουν.

Οι ROM είναι χρήσιμες για τη διατήρηση δεδομένων που δεν αλλάζουν ποτέ. Τα αριθμητικά κυκλώματα μπορούν να χρησιμοποιήσουν πίνακες, για να επιταχύνουν τους υπολογισμούς των λογαρίθμων ή των διαιρέσεων. (Katz, 2005; Predko, 2005; Κοσσιδάς & Γιαννακόπουλος, 2006; Balabanian, & Carlson 2007; Maxfield, 2009; Godse, & Godse, 2010; Tocci, et al., 2010; Mano, & Ciletti, 2014)

Πολλοί υπολογιστές χρησιμοποιούν την ROM, για να αποθηκεύσουν σημαντικά προγράμματα, τα οποία δεν πρόκειται να μεταβληθούν, όπως το σύστημα BIOS.

Οι είσοδοι WR και DATA, που είδαμε στις RAM, δε χρειάζονται. (Σχήμα 9.23). Τα δεδομένα αποθηκεύονται σε μία ψηφίδα ROM με τη χρήση ειδικών εργαλείων του υλικού.



Σχήμα 9.23. Δομικό Διάγραμμα μιας $2^k \times n$ μνήμης RAM

Οι παιγνιδομηχανές, τα PDA's, τα κυψελιδωτά τηλέφωνα, οι μηχανές αυτόματης πώλησης και άλλες ηλεκτρονικές συσκευές μπορούν, επίσης, να περιέχουν μη τροποποιούμενα προγράμματα.

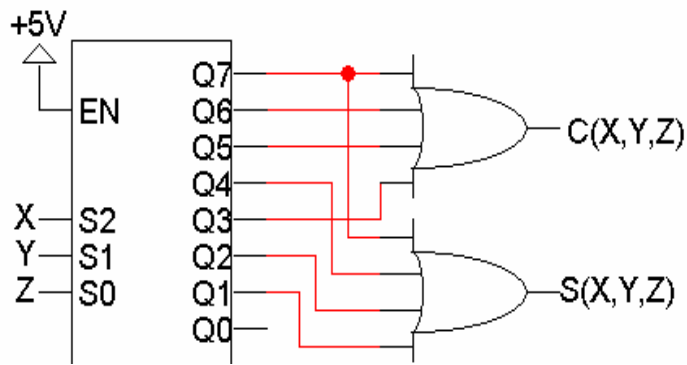
Οι μνήμες ROM είναι συνδυαστικές συσκευές, και όχι ακολουθιακές! Δεν μπορούμε να αποθηκεύσουμε αυθαίρετα δεδομένα σε μία μνήμη ROM, και έτσι η ίδια διεύθυνση θα περιέχει πάντα τα ίδια δεδομένα. Μπορούμε να θεωρήσουμε την ROM σαν ένα συνδυαστικό κύκλωμα, το οποίο παίρνει μια διεύθυνση σαν είσοδο, και παράγει κάποια δεδομένα σαν έξοδο.

Ένας πίνακας ROM είναι βασικά ένας πίνακας αληθείας. Ο πίνακας δείχνει τι δεδομένα είναι αποθηκευμένα σε κάθε διεύθυνση της ROM. (Πίνακας 9.4). Μπορούμε να παράγουμε τα δεδομένα αυτά συνδυαστικά, χρησιμοποιώντας τη διεύθυνση σαν είσοδο.

Διεύθυνση $A_2A_1A_0$	Δεδομένα $V_2V_1V_0$
000	000
001	100
010	110
011	100
100	101
101	000
110	011
111	011

Πίνακας 9.4. Περιεχόμενα διευθύνσεων μνήμης ROM

Μπορούμε εύκολα να μετατρέψουμε τους πίνακες αληθείας σε κυκλώματα, με τους αποκωδικοποιητές. (Σχήμα 9.24 και πίνακας 9.5)



Σχήμα 9.24. Δομικό Διάγραμμα μιας $2^k \times n$ μνήμης ROM

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

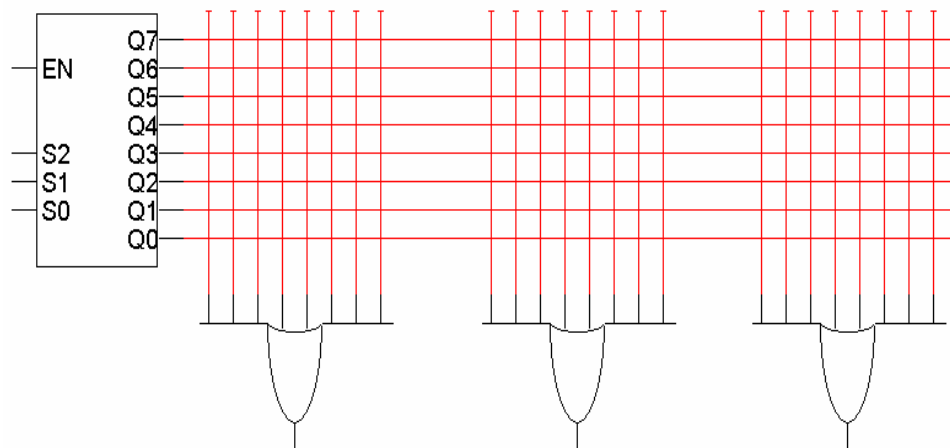
Πίνακας 9.5. Πίνακας αθροιστή τριών ψηφίων

Για παράδειγμα, μπορούμε να θεωρήσουμε αυτό το γνωστό μας κύκλωμα, σαν μία μνήμη που «αποθηκεύει» τις εξόδους S(άθροισμα) και C(κρατούμενο) του πίνακα αληθείας (πίνακας 9.5).

9.2.4.1 Οργάνωση της ROM

Οι μνήμες ROM βασίζονται στην προηγούμενη υλοποίηση των συναρτήσεων με αποκωδικοποιητές. Μία άγραφη ROM απλά περιέχει έναν αποκωδικοποιητή και μερικές πύλες OR. Οι συνδέσεις μεταξύ του αποκωδικοποιητή και των πυλών OR είναι προγραμματιζόμενες, και έτσι μπορούν να υλοποιηθούν διαφορετικές συναρτήσεις.

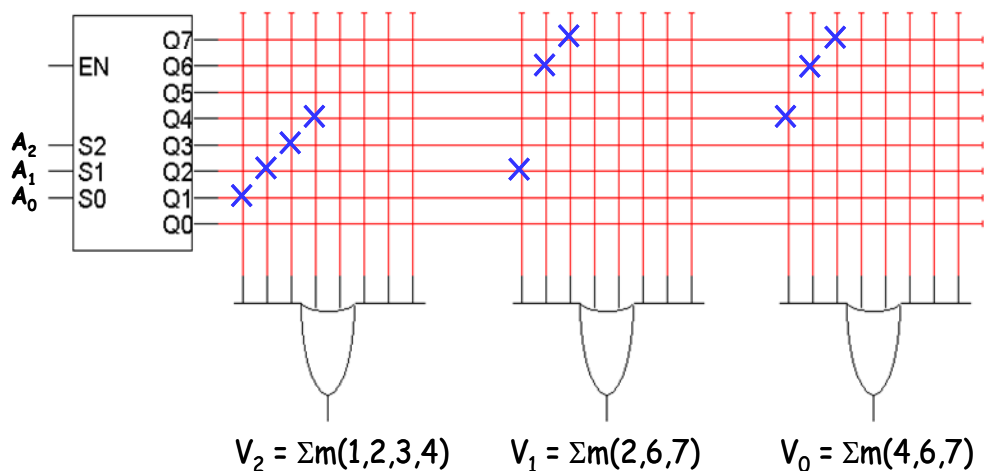
Για να προγραμματίσουμε μια ROM, απλώς πραγματοποιούμε τις επιθυμητές συνδέσεις μεταξύ των εξόδων του αποκωδικοποιητή και των εισόδων των πυλών OR. (Σχήμα 9.25)



Σχήμα 9.25 Άγραφη μνήμη ROM με αποκωδικοποιητή και πύλες OR

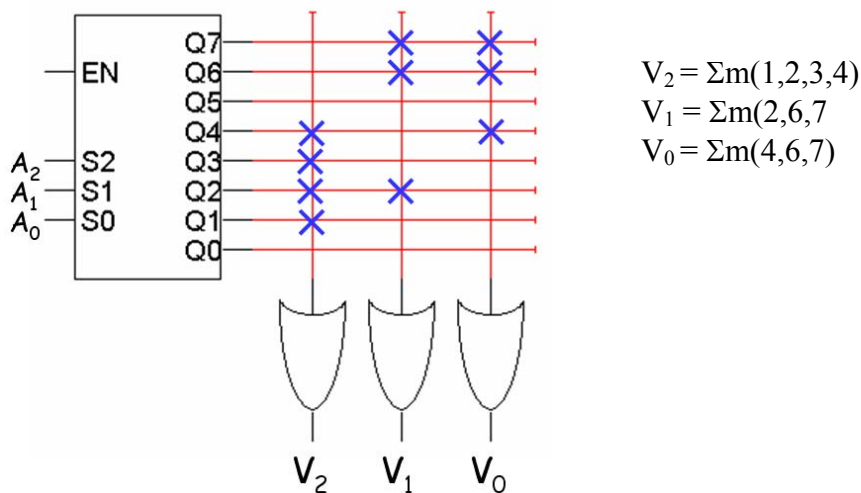
Παράδειγμα 1ο:

Παραθέτουμε τις τρεις συναρτήσεις $V_2V_1V_0$, που υλοποιούνται με μία 8x3 ROM (Σχήμα 9.26). Με (X) σημειώνεται η σύνδεση μεταξύ των εξόδων του αποκωδικοποιητή και των πυλών OR. Σε άλλη περίπτωση δεν υπάρχει καμία σύνδεση.



Σχήμα 9.26. Μνήμη ROM 8 x 3

Μια εναλλακτική υλοποίηση της 8x3 μνήμης ROM του σχήματος 9.26, είναι η υλοποίηση με τη χρήση «συντεταγμένων» πυλών OR, ώστε το σχήμα να γίνει πιο απλό. (Σχήμα 9.27)



Σχήμα 9.27. Μνήμη ROM 8 x 3 με υλοποίηση με τη χρήση «συντεταγμένων» πυλών OR

Το προηγούμενο συνδυαστικό κύκλωμα μπορεί να θεωρηθεί σαν μία μνήμη ROM, καθώς αποθηκεύει οκτώ λέξεις δεδομένων, η καθεμία των οποίων αποτελείται από 3-bit.

Οι εισοδοί του αποκωδικοποιητή σχηματίζουν μία διεύθυνση, η οποία αναφέρεται σε μία από τις οκτώ διαθέσιμες λέξεις. Έτσι κάθε συνδυασμός εισόδων αντιπροσωπεύει μία διεύθυνση, η οποία «διαβάζεται» για να παράγει μία έξοδο δεδομένων των 3-bit.

(Να συγκρίνετε την ανωτέρω υλοποίηση με αυτή της παραγράφου 5.2 του κεφαλαίου 6)

9.2.5. Σύγκριση της ROM και της RAM

Υπάρχουν μερικές σημαντικές διαφορές μεταξύ της ROM και της RAM. Οι ROM είναι «μη-πτητικές» – τα δεδομένα διατηρούνται ακόμα και χωρίς τάση. Αντίθετα, τα περιεχόμενα των RAM χάνονται μόλις μηδενισθεί η τάση.

Οι ROM χρειάζονται ειδικές (και πιο αργές) τεχνικές για την εγγραφή δεδομένων, και γι' αυτό και θεωρούνται σαν συσκευές μόνο αναγνώσιμες. Μερικοί νεότεροι τύποι ROM επιτρέπουν την ευκολότερη εγγραφή, αν και οι ταχύτητές τους ακόμα δε συγκρίνονται με αυτές των μνημών RAM.

Τα MP3 player, οι ψηφιακές κάμερες και άλλα παιχνίδια χρησιμοποιούν κάρτες CompactFlash, Secure Digital ή MemoryStick για τη μη-πτητική αποθήκευση.

Πολλές συσκευές επιτρέπουν στο χρήστη την αναβάθμιση των αποθηκευμένων, σε μία «Flash ROM», προγραμμάτων. Αναφέρουμε, ενδεικτικά, μερικές τιμές χωρητικότητας των μνημών αυτών:

Compact Flash Pro: 128MB- 4GB	Memory Stick : 64- 128MB
Duo: 64MB- 4GB	SD memory card : 2- 8GB

9.3. Βασική περιγραφή της Αριθμητικής μονάδας

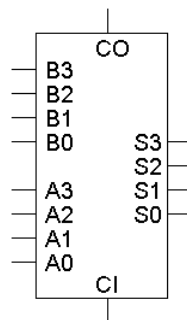
Μία αριθμητική και λογική μονάδα (ΑΛΜ), εκτελεί πολλές διαφορετικές αριθμητικές και λογικές λειτουργίες. Η ΑΛΜ είναι η «καρδιά» του επεξεργαστή, και μπορούμε να πούμε, πως οτιδήποτε άλλο υπάρχει στην ΚΜΕ βρίσκεται εκεί για να υποστηρίξει την ΑΛΜ. (Katz, 2005; Κοσσίδης, & Γιαννακόπουλος, 2006; Holdsworth, & Woods, 2002; Godse, & Godse, 2010)

Θα αναπτύξουμε πρώτα μία αριθμητική μονάδα, παίρνοντας βασικές ιδέες από το κύκλωμα του αθροιστή – αφαιρέτη.

Κατόπιν θα μιλήσουμε για τις λογικές λειτουργίες, και θα κατασκευάσουμε μία λογική μονάδα. Στο τέλος θα ενώσουμε τα δύο αυτά τμήματα, δηλαδή την αριθμητική και τη λογική μονάδα, χρησιμοποιώντας ένα πολυπλέκτη.

9.3.1. Αθροιστής 4-bit

Ο βασικός αθροιστής των 4-bit (Σχήμα 9.28) υπολογίζει πάντα τη συνάρτηση: $S = A+B+CI$.



Σχήμα 9.28. Αθροιστής δύο τετραψήφιων αριθμών

Αλλάζοντας τα σήματα των εισόδων A, B και CI του αθροιστή, μπορούμε να αλλάξουμε την έξοδο του S. Αυτό ακριβώς εφαρμόζουμε, για να κατασκευάσουμε το συνδυασμένο κύκλωμα του αθροιστή – αφαιρέτη που ακολουθεί.

9.3.2. Ο Αθροιστής – Αφαιρέτης

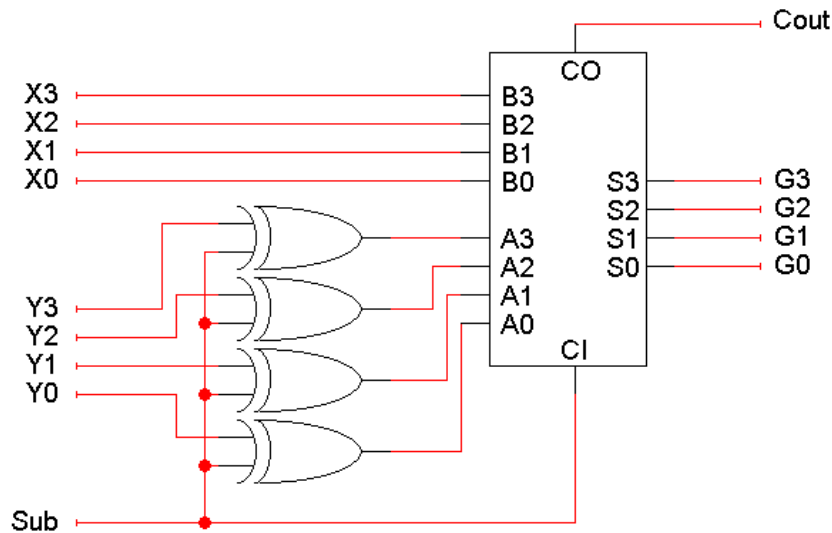
Στο διάγραμμα του σχήματος 9.29 που ακολουθεί το σήμα Sub και οι πύλες XOR αλλάζουν τις εισόδους του αθροιστή.

Όταν το Sub = 0, οι εισόδοι A, B, CI του αθροιστή είναι Y, X, 0 και έτσι ο αθροιστής δίνει στην έξοδο $G = X + Y + 0$, ή αλλιώς $G = X + Y$.

Όταν το Sub = 1, οι εισόδοι του αθροιστή είναι, αντίστοιχα, Y', X και 1, και η έξοδος του αθροιστή είναι $G = X + Y' + 1$, ή απλά η με συμπλήρωμα ως προς 2, πράξη των δύο μεταβλητών $G = X - Y$.

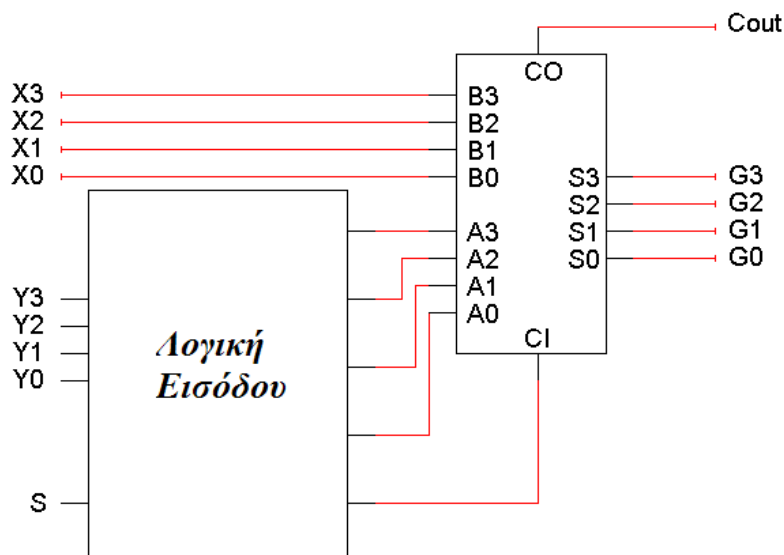
Έχουμε λοιπόν ένα αθροιστή, ο οποίος εκτελεί δύο διαφορετικές λειτουργίες. Η είσοδος «Sub» δρα σαν επιλογή της συνάρτησης, και καθορίζει αν το κύκλωμα θα κάνει πρόσθεση ή αφαίρεση. Βλέποντας το από την πλευρά του κυκλώματος, το μόνο που κάνει η «Sub» είναι να τροποποιεί τις εισόδους A και CI του αθροιστή.

Μεταβάλλοντας τις εισόδους του αθροιστή, μπορούμε να χρησιμοποιήσουμε έναν αθροιστή για την εκτέλεση και άλλων λειτουργιών.



Σχήμα 9.29 Διάγραμμα Αθροιστή-Αφαιρέτη

Απλά πρέπει να ξεκαθαρίσουμε ποιες συναρτήσεις θέλουμε να υλοποιήσουμε, και μετά να τοποθετήσουμε την κατάλληλη συνδεσμολογία στη λογική είσοδο του κυκλώματος, που είναι αρχικά «κενή». (Σχήμα 9.30)



Σχήμα 9.30 Αθροιστής – Αφαιρέτης με «κενή» λογική εισόδου

Είδαμε τον τρόπο με τον οποίο μπορούμε να ορίζουμε τις εισόδους A, B, CI του αθροιστή, για να υπολογίζει είτε την $X + Y$ είτε την $X - Y$. Μερικά κύρια ερωτήματα είναι:

- Πώς μπορούμε να παράγουμε τη συνάρτηση αύξησης κατά 1, δηλαδή $G = X + 1$;
Ένας τρόπος υπάρχει: Θέτουμε $A = 0000$, $B = X$ και $CI = 1$.
- Για την ελάττωση κατά 1 αντίστοιχα, δηλαδή $G = X - 1$;
Θέτουμε $A = 1111$ (-1), $B = X$, $CI = 0$
- Η συνάρτηση μεταφοράς $G = X$; (Μπορεί να φανεί χρήσιμη)
Σε αυτή την περίπτωση $A = 0000$, $B = X$, $CI = 0$

Οι λειτουργίες της μεταφοράς και της αύξησης έχουν τις ίδιες εισόδους A, B, και διαφέρουν μόνο στην είσοδο CI.

Γενικότερα μπορούμε να πάρουμε και άλλες συναρτήσεις (δεν είναι όλες χρήσιμες), χρησιμοποιώντας το $CI = 0$ και το $CI = 1$.

Παράδειγμα 2^ο:

Η αφαίρεση με συμπλήρωμα ως προς δύο λαμβάνεται, εάν θέσουμε: $A = Y'$, $B = X$, και $CI = 1$, συνεπώς $G = X + Y' + 1$.

Αν κρατήσουμε το $A = Y'$ και το $B = X$, αλλά θέσουμε το $CI = 0$, τότε παίρνουμε στην έξοδο: $G = X + Y'$. Αυτή η λειτουργία είναι πλέον η αφαίρεση με συμπλήρωμα ως προς ένα.

9.3.3. Αριθμητικές συναρτήσεις

Στον πίνακα 9.6, που ακολουθεί, παραθέτουμε μερικές από τις αριθμητικές πράξεις –λειτουργίες, που είναι πιθανόν να μας χρειαστούν.

Θα πρέπει να βρούμε ένα τρόπο, για να προσδιορίσουμε τη συνάρτηση για την οποία ενδιαφερόμαστε. Συνεπώς, τυχαία έχουμε εκχωρήσει ένα κώδικα επιλογής για κάθε λειτουργία.

S2	S1	S0	Αριθμητική λειτουργία	
0	0	0	X	(μεταφορά)
0	0	1	X + 1	(αύξηση κατά 1)
0	1	0	X + Y	(πρόσθεση)
0	1	1	X + Y + 1	
1	0	0	X + Y'	(αφαίρεση με 1's C)
1	0	1	X + Y' + 1	(αφαίρεση με 2's C)
1	1	0	X - 1	(ελάττωση κατά 1)
1	1	1	X	(μεταφορά)

Πίνακας 9.6 Αριθμητικές Πράξεις

Απεικόνιση του Πίνακα σε ένα Αθροιστή

Ο πίνακας 9.7, που ακολουθεί, δείχνει ποιες θα πρέπει να ήταν οι είσοδοι του αθροιστή για κάθε μία από τις οκτώ επιθυμητές αριθμητικές λειτουργίες.

Κώδικας επιλογής			Επιθυμητή Αριθμητική λειτουργία G (A + B + CI)	Απαιτούμενη Είσοδος αθροιστή		
S2	S1	S0		A	B	CI
0	0	0	X (μεταφορά)	0000	X	0
0	0	1	X + 1 (αύξηση κατά 1)	0000	X	1
0	1	0	X + Y (πρόσθεση)	Y	X	0
0	1	1	X + Y + 1	Y	X	1
1	0	0	X + Y' (αφαίρεση με 1's C)	Y'	X	0
1	0	1	X + Y' + 1 (αφαίρεση με 2's C)	Y'	X	1
1	1	0	X - 1 (ελάττωση κατά 1)	1111	X	0
1	1	1	X (μεταφορά)	1111	X	1

Πίνακας 9.7. Αριθμητικές λειτουργίες

Η είσοδος CI είναι πάντα ίδια με τον κώδικα επιλογής της S_0 .

Η B τίθεται (συνδέεται) πάντα στο X.

Η A εξαρτάται μόνο από τις S_2 και S_1 .

Αυτές οι εξισώσεις εξαρτώνται και από την επιθυμητή πράξη και από τον τρόπο που έχει γίνει η εκχώρηση στον κώδικα επιλογής.

Το μόνο που χρειάζεται είναι να υπολογίσουμε την είσοδο A του αθροιστή, με δεδομένη την αριθμητική μονάδα εισόδου Y και τον κώδικα της συνάρτησης επιλογής S (βασικά μόνο τα S_2 και S_1).

Στον πίνακα 9.7α παραθέτουμε τον συντετηγμένο πίνακα αλήθειας του πίνακα 9.7:


S2	S1	A
0	0	0000
0	1	Y
1	0	Y'
1	1	1111

Πίνακας 9.7a. Συντεταγμένος πίνακας αληθείας

Θέλουμε να επιλέξουμε μία από τις τέσσερις πιθανές τιμές της A, ανάλογα με τις τιμές των S₂ και S₁. Θα μπορούσαμε να κατασκευάσουμε το κύκλωμα αυτό με βασικές πύλες. Αν θέλουμε να χρησιμοποιήσουμε χάρτη Karnaugh για την απλοποίηση, τότε θα πρέπει πρώτα να επεκτείνουμε τον προηγούμενο πίνακα αληθείας.

Τα Y και Y' που εμφανίζονται στη στήλη εξόδου (A) του πρώτου πίνακα είναι μία είσοδος. Αυτό γίνεται σαφές στον πίνακα 9.8 που ακολουθεί:

S ₂	S ₁	A
0	0	0000
0	1	Y
1	0	Y'
1	1	1111



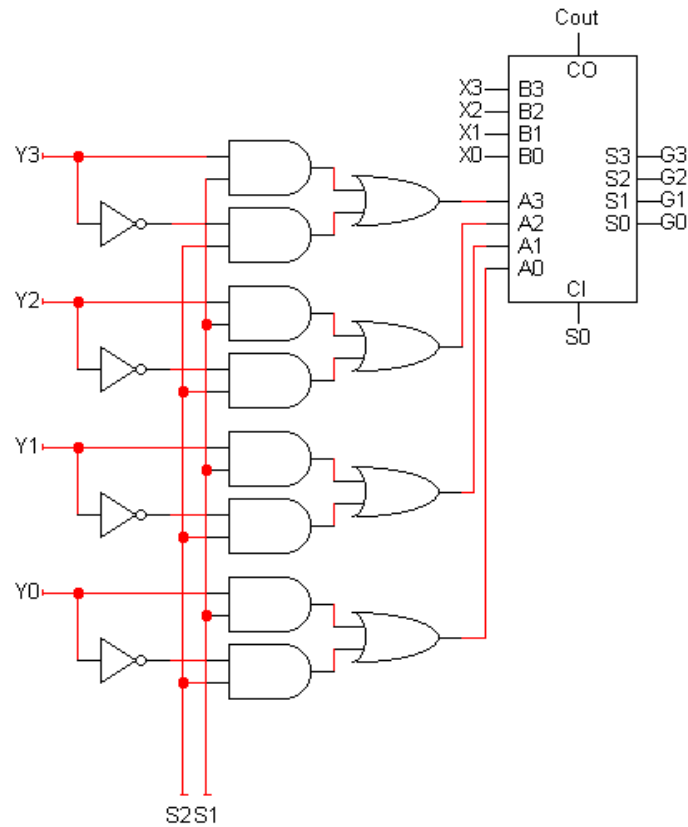
S ₂	S ₁	Y _i	A _i
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Πίνακας 9.8 Επέκταση του πίνακα αλήθειας με τη χρήση του Y_i

Υπενθυμίζεται ότι τα A και Y έχουν μήκος 4-bit. Με βάση τον πίνακα αληθείας μπορούμε να βρούμε το ΕΑΓ: $A_i = S_2 Y_i' + S_1 Y_i$

	S ₂ S ₁			
Y _i	0	0	1	1
	0	1	1	0

Πρέπει πάλι να επαναλάβουμε αυτήν τη διαδικασία για κάθε bit Y₃-Y₀, που συνδέεται στις εισόδους A₃-A₀ του αθροιστή. Τελικά από τις προκύπτουσες συναρτήσεις καταλήγουμε στο διάγραμμα του σχήματος 9.31 που ακολουθεί. Αυτό ολοκληρώνει την αριθμητική μονάδα μας.



Σχήμα 9.31. Κύκλωμα Λογικής Εισόδου με βασικές Πύλες

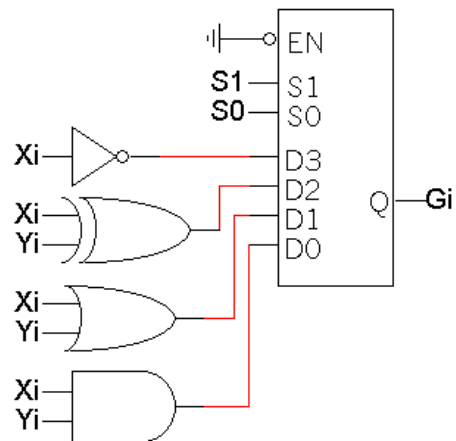
9.4. Βασική περιγραφή της Λογικής Μονάδας

Μία λογική μονάδα υποστηρίζει διαφορετικές λογικές συναρτήσεις σε δύο εισόδους X και Y πολλαπλών bit, παράγοντας μία έξοδο G.

Ο συντεταγμένος πίνακας 9.9 που ακολουθεί δείχνει τέσσερις πιθανές συναρτήσεις και εκχωρεί ένα κώδικα επιλογής S σε κάθε μια.

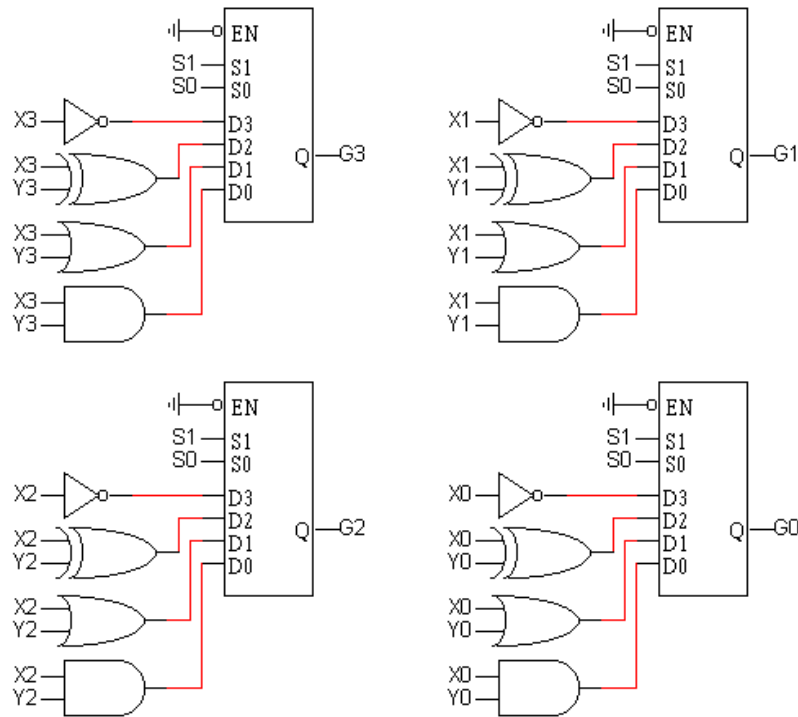
S1	S0	Έξοδος
0	0	$G_i = X_i Y_i$
0	1	$G_i = X_i + Y_i$
1	0	$G_i = X_i \oplus Y_i$
1	1	$G_i = X_i'$

Πίνακας 9.9. Συντεταγμένος πίνακας λογικών συναρτήσεων



Σχήμα 9.32. Υλοποίηση του πίνακα 9.9

Για την υλοποίηση της λογικής μονάδας θα χρησιμοποιήσουμε το δομικό διάγραμμα του Σχήματος 9.32 (Σχήμα 9.33).



Σχήμα 9.33. Λογική Μονάδα

Στην απλή λογική μονάδα του σχήματος 9.32 χρησιμοποιήσαμε πολυπλέκτες και βασικές πύλες για να την υλοποιήσουμε. Και πάλι χρειαζόμαστε έναν πολυπλέκτη για κάθε ψηφίο (bit) των X και Y. Τέλος αναφέρουμε ότι τα ψηφία εισόδου και εξόδου είναι τα ακόλουθα

Οι εισοδοι είναι: και η έξοδος είναι:
 X (4 bits) G (4 bits)
 Y (4 bits)
 S (2 bits)

9.5. Συνδυάζοντας την Αριθμητική και τη Λογική Μονάδα

Τώρα έχουμε πλέον αναλύσει πλήρως τα δύο τμήματα της αριθμητικής και λογικής μονάδας:

- Μία αριθμητική μονάδα που μπορεί να υπολογίσει οκτώ συναρτήσεις για εισόδους των 4-bit.
- Μία λογική μονάδα που μπορεί να υλοποιήσει τέσσερις συναρτήσεις σε εισόδους των 4-bit.

Μπορούμε να συνδυάσουμε αυτές τις δύο μονάδες σε ένα μόνο κύκλωμα, που ονομάζεται αριθμητική – λογική μονάδα (ΑΛΜ).

Πίνακας λειτουργίας της ALU

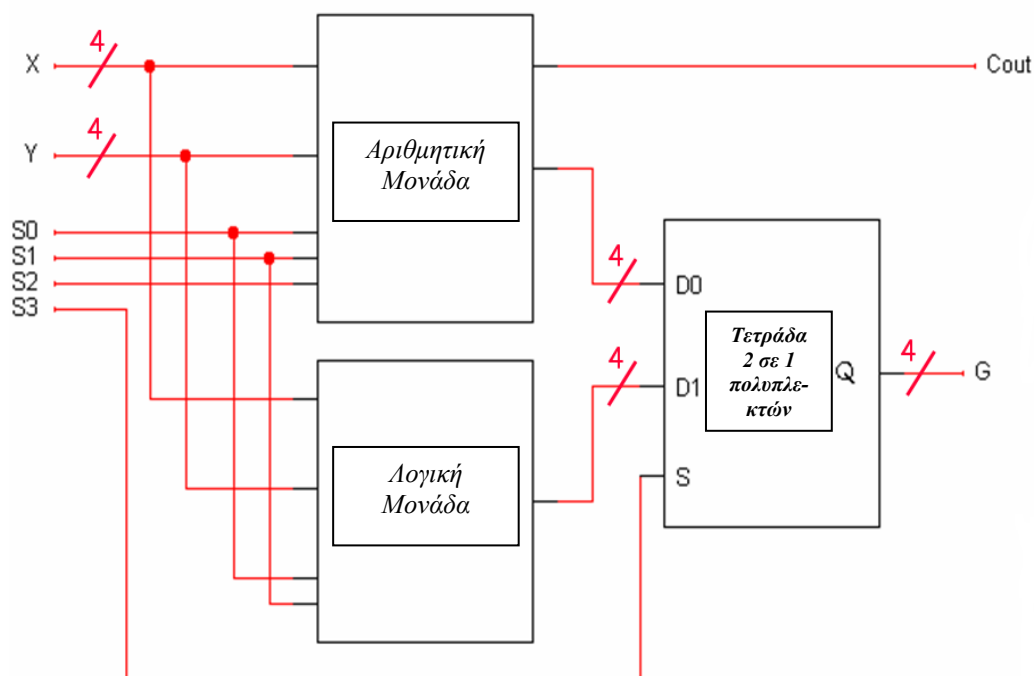
Στον πίνακα 9.10, που ακολουθεί, παρατίθεται ένα δείγμα συναρτήσεων μίας ΑΛΜ. Όλες οι αριθμητικές λειτουργίες έχουν $S3=0$, και όλες οι λογικές $S3=1$. Αυτές είναι οι ίδιες συναρτήσεις που είδαμε, όταν κατασκευάζαμε, προηγουμένως, την αριθμητική και τη λογική μονάδα.

Καθώς η ΑΛΜ έχει μόνο 4 λογικές πράξεις, δε χρειαζόμαστε την είσοδο $S2$. Η λειτουργία που εκτελείται από τη λογική μονάδα εξαρτάται μόνο από τις εισόδους $S1$ και $S0$.

S ₃	S ₂	S ₁	S ₀	Λειτουργία
0	0	0	0	G = X
0	0	0	1	G = X + 1
0	0	1	0	G = X + Y
0	0	1	1	G = X + Y + 1
0	1	0	0	G = X + Y'
0	1	0	1	G = X + Y' + 1
0	1	1	0	G = X - 1
0	1	1	1	G = X
1	x	0	0	G = X and Y
1	x	0	1	G = X or Y
1	x	1	0	G = X ⊕ Y
1	x	1	1	G = X'

Πίνακας 9.10. Πιθανές συναρτήσεις μίας ΑΛΜ

Το πλήρες κύκλωμα της Αριθμητικής και Λογικής Μονάδας αποδίδεται στο Σχήμα 9.34.



Σχήμα 9.34. Κύκλωμα αριθμητικής και λογικής μονάδας

Παρατηρήσεις:

Το σύμβολο 4/ υποδηλώνει ότι στην πραγματικότητα υπάρχουν τέσσερις γραμμές.

Η Cout πρέπει να αγνοηθεί, όταν εκτελούνται λογικές λειτουργίες (όταν S₃=1).

Η G είναι η τελική έξοδος της ΑΛΜ.

Όταν S₃ = 0, η τελική έξοδος έρχεται από την αριθμητική μονάδα.

Όταν S₃ = 1, η έξοδος έρχεται από τη λογική μονάδα.

Η αριθμητική και η λογική μονάδα μοιράζονται τις εισόδους S₁ και S₀, αλλά μόνο η αριθμητική μονάδα χρησιμοποιεί την S₂.

Σχόλια για τον Πολυπλέκτη

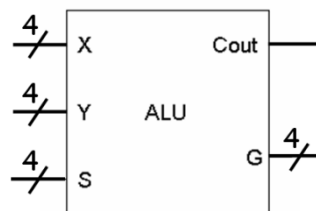
Τόσο η αριθμητική όσο και η λογική μονάδα είναι «ενεργές» παράγον, δηλαδή, έξοδο. Ο πολυπλέκτης καθορίζει, αν το τελικό αποτέλεσμα προέρχεται από την αριθμητική ή τη λογική μονάδα και η έξοδος της άλλης μονάδας αγνοείται.

Η διάταξη του υλικού μας, μοιάζει σαν μια αποτυχημένη προσπάθεια, αλλά στην πραγματικότητα δεν είναι. «Απενεργοποιώντας» τη μια ή την άλλη μονάδα δε θα κερδίζαμε αρκετό χρόνο. Πρέπει, έτσι κι αλλιώς, να κατασκευάσουμε υλικό και για τις δύο μονάδες. Άρα, κάλλιστα, θα μπορούσαμε να τις εκτελούμε ταυτόχρονα. Αυτή είναι μία συνήθης χρήση των πολυπλεκτών στη λογική σχεδίαση.

Η ολοκληρωμένη ALM

Η μονάδα ALM είναι ένα καλό παράδειγμα ιεραρχικού σχεδιασμού. Με 12 εισόδους, ο πίνακας αληθείας θα είχε $2^{12} = 4096$ γραμμές, πράγμα πολύ δύσκολο να σχεδιασθεί στο χαρτί. Αντίθετα, έχουμε τη δυνατότητα να χρησιμοποιήσουμε εξαρτήματα, που έχουμε ήδη γνωρίσει. Μπορούμε λοιπόν να κατασκευάσουμε ολόκληρο το κύκλωμα από μερικά, απλά στην κατανόηση εξαρτήματα.

Όπως, πάντα, ενθυλακώνουμε το κύκλωμα μέσα σε ένα «κουτί» έτσι, ώστε να μπορούμε να το ξαναχρησιμοποιήσουμε σε πολυπλοκότερα κυκλώματα (Σχήμα 9.35).



Σχήμα 9.35. Δομικό διάγραμμα της ALU

9.6. Πράξεις και δικτυακές λειτουργίες με ένα δυαδικό ψηφίο

Οι περισσότεροι υπολογιστές υποστηρίζουν λογικές πράξεις, όπως οι AND, OR και NOT, αλλά για λέξεις πολλαπλών bit και όχι για μόνο ένα bit. (Κοσσίδης, & Γιαννακόπουλος, 2006)

Για να εφαρμόσουμε μία λογική λειτουργία σε δύο λέξεις X και Y, εφαρμόζουμε τη λειτουργία για κάθε ζευγάρι bits X_i και Y_i :

	1	0	1	1
AND	1	1	1	0
	1	0	1	0

	1	0	1	1
OR	1	1	1	0
	1	1	1	1

	1	0	1	1
XOR	1	1	1	0
	0	1	0	1

Τις παραπάνω λειτουργίες τις έχουμε ήδη δει αναλυτικά, στην αριθμητική συμπληρώματος ως προς 2, όταν αναλύσαμε την έννοια του συμπληρώματος σε όλα τα bits ενός αριθμού (με βάση το δύο).

Πράξεις στις γλώσσες προγραμματισμού

Γλώσσες σαν τις C, C++ και Java προσφέρουν πράξεις λογικής του 1-bit:

& (AND) | (OR) ^ (XOR) ~ (NOT)

Αυτές οι λειτουργίες μεταχειρίζονται κάθε ακέραιο σαν ένα σύνολο από ανεξάρτητα bit:

Έτσι $13 \& 25 = 9$ διότι $01101 \& 11001 = 01001$

Δε συμβαίνει το ίδιο με τους τελεστές **&&**, **||** και **!**, οι οποίοι μεταχειρίζονται κάθε ακέραιο σαν μία λογική τιμή (0 είναι false και όλα τα άλλα true):

$13 \&\& 25 = 1$ διότι $true \&\& true = true$

Οι τελεστές του 1-bit χρησιμοποιούνται συχνά στα προγράμματα, για να θέσουν σε ένα σύνολο από Boolean επιλογές, ή flags, ένα όρισμα.

Είναι εύκολο να αναπαραστήσουμε, σύνολα από συγκεκριμένα μεγέθη, με bits: 1: ανήκει 0: δεν ανήκει.

Δικτυακές λειτουργίες

Οι διευθύνσεις IP στην πραγματικότητα είναι δυαδικοί αριθμοί 32-bit, και οι λειτουργίες με 1-bit μπορούν να χρησιμοποιηθούν για να βρεθούν πληροφορίες για το δίκτυο.

Για παράδειγμα, μπορούμε να εκτελέσουμε την πράξη AND 1-bit στη διεύθυνση 192.168.10.43 με μια «μάσκα υποδικτύου» για να βρούμε την «διεύθυνση δικτύου», ή το δίκτυο στο οποίο είναι συνδεδεμένη η μηχανή.

$$\begin{array}{r}
 192.168. 10. 43 = 11000000.10101000.00001010.00101011 \\
 \& 255.255.255.224 = 11111111.11111111.11111111.11100000 \\
 \hline
 192.168. 10. 32 = 11000000.10101000.00001010.00100000
 \end{array}$$

Μπορούμε να χρησιμοποιήσουμε πάλι την πράξη OR 1-bit, για να παράγουμε μία «διεύθυνση μετάδοσης», για να στείλουμε δεδομένα σε όλα τα μηχανήματα που είναι στο τοπικό δίκτυο.

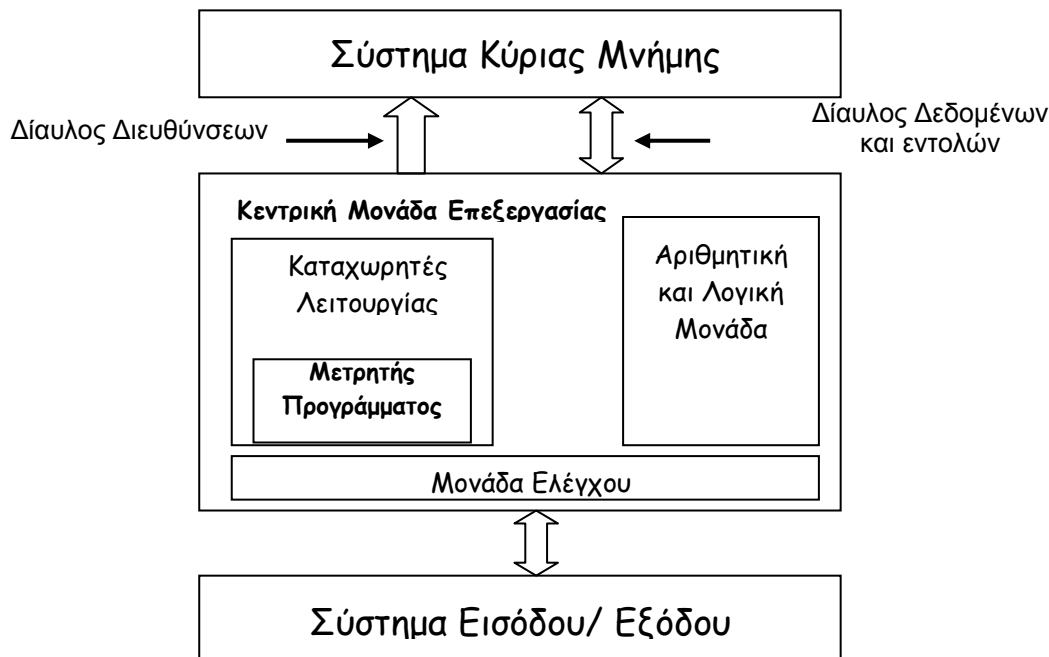
$$\begin{array}{r}
 192.168. 10. 43 = 11000000.10101000.00001010.00101011 \\
 | \quad 0. \quad 0. \quad 0. 31 = 00000000.00000000.00000000.00011111 \\
 \hline
 192.168. 10. 63 = 11000000.10101000.00001010.00111111
 \end{array}$$

9.7. Αρχιτεκτονική των υπολογιστών

Θα αναφέρουμε περιληπτικά τις βασικές αρχιτεκτονικές, που χρησιμοποιούνται από το 1945 μέχρι και σήμερα στα υπολογιστικά συστήματα. (Κοσσιδάς, & Γιαννακόπουλος, 2006; Κοσσιδάς, Ραγκούση, & Ματιάτος, 1996)

Α) Δομή Von Neumann

Ο John von Neumann (1903-1957) – ήταν πρωτοπόρος των Η/Υ και εφευρέτης. Εργάστηκε στο Institute of Advanced Studies του Princeton στον σχεδιασμό των Η/Υ από τα μέσα της δεκαετίας του 1940. Οι απλούστερες αρχιτεκτονικές των Η/Υ ονομάστηκαν προς τιμή του, και είναι γνωστές σαν μηχανές του von Neumann. Όλες οι μηχανές μέχρι και σήμερα, είναι μηχανές von Neumann.



Σχήμα 9.36. Δομή Von Neumann

Στην πραγματικότητα και ο μοντέρνος σχεδιασμός των Η/Υ βασίζεται στις τρεις έννοιες- κλειδιά του Von Neumann:

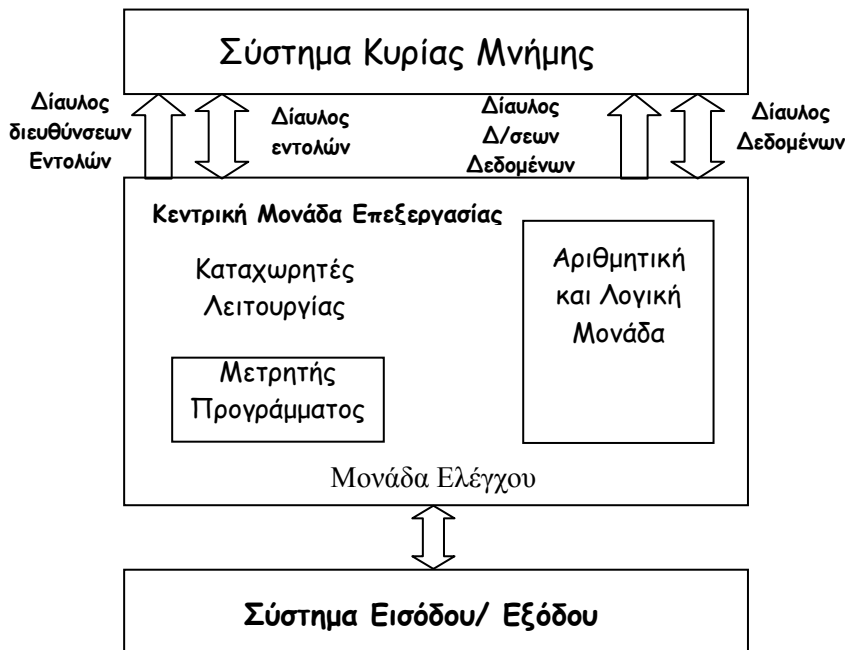
1. **Δεδομένα** και **εντολές** αποθηκεύονται σε μία **μνήμη** εγγραφής-ανάγνωσης.
2. Παίρνουμε τα περιεχόμενα μίας θέσης μνήμης, ανάλογα με τη θέση, **διεύθυνση**, στην οποία αναφερόμαστε, χωρίς να μας ενδιαφέρει ο τύπος των δεδομένων που εμπεριέχονται.
3. Η εκτέλεση των εντολών γίνεται με ένα σειριακό τρόπο από τη μία εντολή στην επόμενη της. (Εκτός εάν αυτή η ακολουθία είναι σαφώς τροποποιημένη με βρόχους και άλματα) και ικανοποιεί τα κάτωθι (4) κριτήρια:

Έχει 3 υποσυστήματα υλικού

- Μία ΚΜΕ
- Σύστημα κύριας μνήμης
- Σύστημα εισόδου/εξόδου

Είναι ένας υπολογιστής αποθήκευσης προγράμματος και εκτελεί τις εντολές σειριακά. Έχει, ή φαίνεται να έχει, ένα απλό δίαυλο επικοινωνίας μεταξύ του συστήματος της κύριας μνήμης και της ΚΜΕ.

B) Δομή Harvard



Σχήμα 9.37. Δομή Harvard

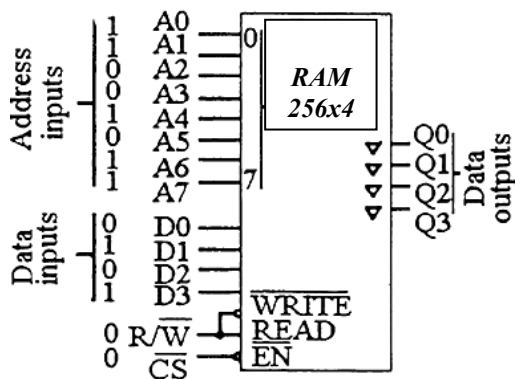
Το όνομα προέρχεται από την εργασία του Howard Aiken στο Harvard. (Τα συστήματα υπολογιστών Mk-I μέχρι και τον Mk-IV). Είναι αυστηρά μία κατηγορία μηχανών von Neumann.

Περιλαμβάνουν χωριστούς διαύλους για τις διευθύνσεις των εντολών και τις διευθύνσεις των δεδομένων. Επίσης έχουν χωριστούς διαύλους για τα δεδομένα και τις εντολές. Μπορεί να περιέχουν ξεχωριστή διάρθρωση μνήμης για το πρόγραμμα και τα δεδομένα.

9.8. Ασκήσεις – Ερωτήσεις

ΕΡΩΤΗΣΕΙΣ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

1. Διαλέξτε την πρόταση η οποία περιγράφει καλύτερα την μνήμη ROM (Read-Only Memory)
 - A. Μη πτητική, χρησιμοποιείται για να αποθηκεύει πληροφορίες που αλλάζουν κατά τη διάρκεια λειτουργίας του συστήματος.
 - B. Μη πτητική, χρησιμοποιείται για να αποθηκεύει πληροφορίες που δεν αλλάζουν κατά τη διάρκεια λειτουργίας του συστήματος.
 - Γ. Πτητική, χρησιμοποιείται για να αποθηκεύει πληροφορίες που αλλάζουν κατά τη διάρκεια λειτουργίας του συστήματος.
 - Δ. Πτητική, χρησιμοποιείται για να αποθηκεύει πληροφορίες που δεν αλλάζουν κατά τη διάρκεια λειτουργίας του συστήματος.
2. Ποιο από τα παρακάτω περιγράφει καλύτερα τις μνήμες EPROMs
 - A. Οι EPROMs μπορούν να προγραμματιστούν μόνο μια φορά
 - B. Οι EPROMs μπορούν να σβηστούν με ακτινοβολία UV
 - Γ. Οι EPROMs μπορούν να σβηστούν γειώνοντας όλες τις εισόδους τους
 - Δ. Όλα τα παραπάνω
3. Για το κύκλωμα που ακολουθεί ποιο από τα παρακάτω είναι σωστό;



- A. Ο αριθμός 5 έχει γραφεί στη μνήμη, στη θέση μνήμης 203.
 - B. Το chip δεν έχει ενεργοποιηθεί, καθώς το EN είναι 0. Γι αυτό, τίποτα δεν θα εγγραφεί στο chip και η έξοδος θα είναι τριών καταστάσεων.
 - Γ. Ο δεκαδικός 10 έχει γραφεί στη θέση μνήμης 211.
 - Δ. Η γραμμή read/write είναι LOW, και έτσι ο δεκαδικός 5 αποθηκεύεται στη θέση μνήμης 211.
4. Μια λέξη των 64-bit αποτελείται από

A. 4 bytes	B. 8 bytes	Γ. 10 bytes	Δ. 12 bytes
------------	------------	-------------	-------------
 5. Πόσα $1K \times 4$ RAM chips απαιτούνται, για να δημιουργήσουμε ένα $1K \times 8$ σύστημα μνήμης;

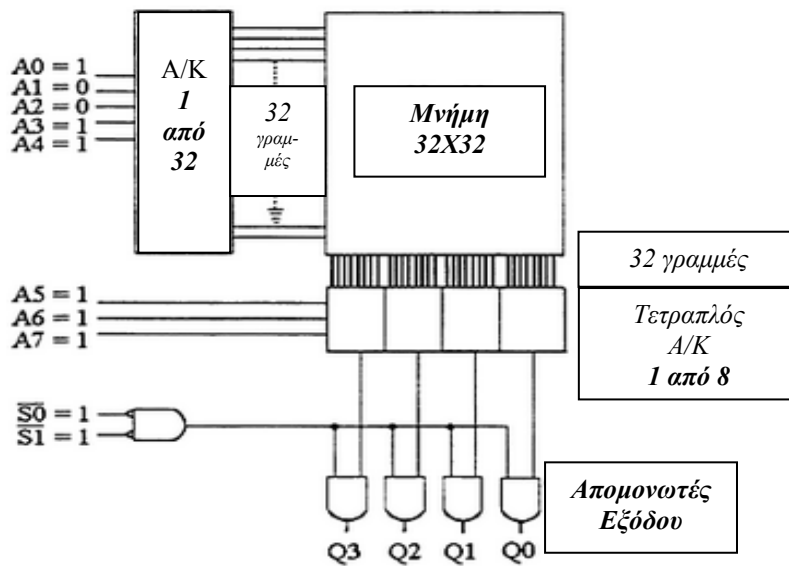
A. 2	B. 4	Γ. 8	Δ. 16
------	------	------	-------
 6. Πόσα $8k \times 1$ RAMs απαιτούνται, για να πετύχουμε μια μνήμη με χωρητικότητα λέξης των 8 k και μήκος λέξης των 8 bits

A. 8	B. 4	Γ. 2	Δ. 1
------	------	------	------
 7. Ποιό από τα παρακάτω δεν είναι κατάσταση ή λειτουργία μιας μνήμης flash

A. Εγγραφή	B. Διαγραφή
Γ. Προγραμματισμός	Δ. Έξαρση(burst)
 8. Η L1 είναι γνωστή σαν

A. Πρωτεύουσα cache	B. Δευτερεύουσα cache
Γ. DRAM	Δ. SRAM

9. Για το κύκλωμα που ακολουθεί, ποιο είναι το μήκος της λέξης εξόδου σε bit;



ΕΡΩΤΗΣΕΙΣ

1. Να ορίσετε τη λογική Μονάδα.
2. Να συγκρίνετε τη δομή Harvard με τη δομή Von Neumann.
3. Να αναπτύξετε τις μνήμες ειδικού τύπου.
4. Να αναφέρετε τον τρόπο εγγραφής και ανάγνωσης στη μνήμη RAM.
5. Τι γνωρίζετε για τη μνήμη ROM;
6. Ποιες είναι οι κυριότερες πράξεις και δικτυακές λειτουργίες με ένα δυαδικό ψηφίο;

Αναφορές-Βιβλιογραφία

- Κοσσίδης, Α.Θ., Γιαννακόπουλος, Π., (2006), *Αριθμητικά Συστήματα και Ψηφιακά Κυκλώματα*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα
- Κοσσίδης Α., Ραγκούση Μ. Ν., Ματιάτος (1996) Σπ. *Συμβολικές γλώσσες και αρχιτεκτονική μικροεπεξεργαστών*, Εκδόσεις Μπένος
- Μικροαρχιτεκτονικές MIPS. In Wikipedia, The Free Encyclopedia. Retrieved, July 17, 2015, from https://en.wikipedia.org/wiki/List_of_MIPS_microarchitectures
- Balabanian, N., Carlson, B. (2007). *Digital Logic Design Principles*, John Wiley
- Floyd, Thomas L., (2006). *Digital Fundamentals*, 9th, Pearson International Edition
- Godse, A.P., Godse, D.A. (2010). *Digital Logic Design and Application*, Technical Publications Pune
- Holdsworth, Brian, Woods, Clive (2002). *Digital Logic Design*, 4th Edition, Newnes
- Katz, R. (2005). *Contemporary Logic Design*, 2/e, Prentice Hall
- Mano, M., Ciletti, M. (2014). *Ψηφιακή Σχεδίαση*, 5^η έκδοση, Παπασωτηρίου
- Maxfield Clive (2009). *Bebop to the Boolean Boogie, An Unconventional Guide to Electronics*, 3rd, Newnes, Elsevier (Online: <http://english.360elib.com/datu/T/EM140371.pdf>)
- Nelson, V., Nagle, H., Carroll, B., Irwin, J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall
- Predko, Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill
- Roth, Charles Jr., Kinney, L.L., (2014). *Fundamentals of Logic Design*, 7th, Cengage Learning
- Tocci, R. J., Widmer, N. S, and Moss, Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley
- Wakerly, J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Κεφάλαιο 10

Σύνοψη

Στο κεφάλαιο αυτό θα δοθούν τέσσερις συχνά χρησιμοποιούμενες εφαρμογές που δίνουν τη χρήση των λογικών κυκλωμάτων στην καθημερινή μας ζωή.

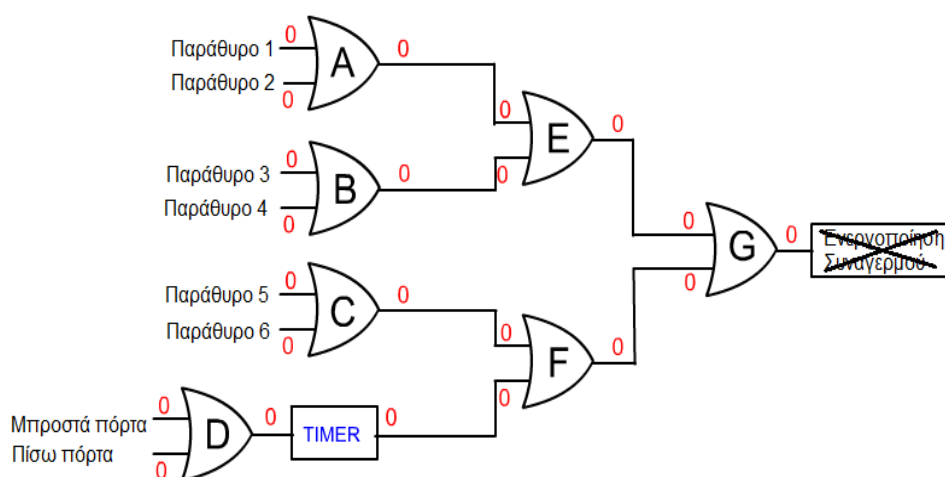
Προαπαιτούμενη γνώση

Οι λειτουργίες των λογικών πυλών, οι χάρτες απλοποίησης λογικών εκφράσεων.

10. Παραδείγματα εφαρμογής λογικών κυκλωμάτων

10.1. Παράδειγμα 1^ο (Οικιακή προστασία)

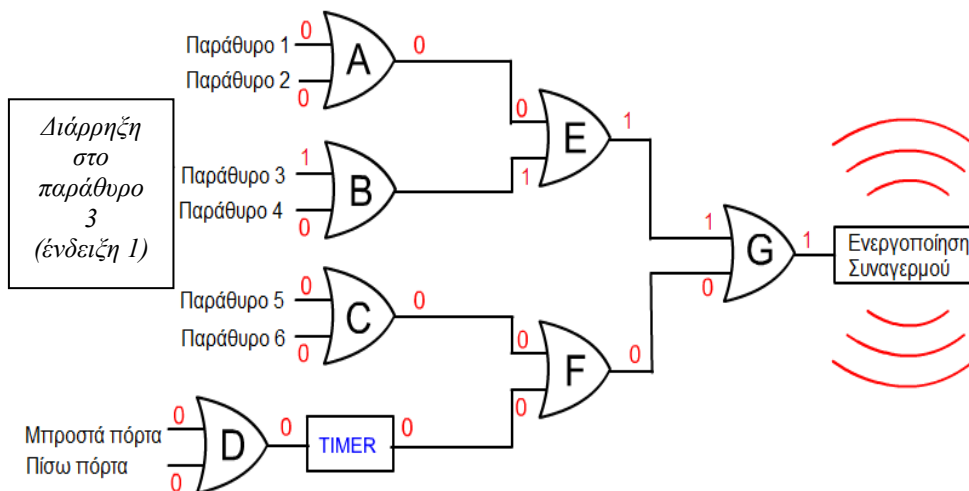
Στο σχήμα 10.1 απεικονίζεται το λογικό κύκλωμα που χρησιμοποιείται για την ενεργοποίηση ενός απλού οικιακού συναγερμού. Ο συναγερμός αυτός προστατεύει τις υπάρχουσες δύο πόρτες του σπιτιού (μία μπροστά και μια πίσω) καθώς και έξι παράθυρά του. Όταν ο συναγερμός είναι ενεργοποιημένος, αν παραβιαστεί οποιαδήποτε από τις πόρτες ή τα παράθυρα, ο συναγερμός θα ηχήσει. Στο διάγραμμα που ακολουθεί χρησιμοποιούνται πύλες OR δύο εισόδων. Ο χρονοδιακόπτης (timer) επιτρέπει στον ιδιοκτήτη του σπιτιού να μπει στο σπίτι είτε από την μπροστινή πόρτα είτε από την πίσω πόρτα, και να απενεργοποιήσει το συναγερμό, πριν αυτός χτυπήσει. Οι εισοδοί κάθε πύλης, που αναπαριστούν τις πόρτες και τα παράθυρα, μπορούν να συνδεθούν σε ένα ευρύ φάσμα αισθητήρων (πχ. αισθητήρες κίνησης και μαγνητικοί αισθητήρες).



Σχήμα 10.1. Αρχική κατάσταση του συναγερμού

Στο κύκλωμα οι καταστάσεις της εισόδου για καθένα από τους αισθητήρες είναι το '0' (false, low, off). Αυτό σημαίνει ότι δεν έχουν εντοπίσει κάποιον εισβολέα. Ως αποτέλεσμα ο συναγερμός είναι ανενεργός.

Η κατάσταση αλλάζει, όταν ένας κακοποιός παραβιάζει, π.χ., το παράθυρο 3 (Σχήμα 10.2). Παρατηρούμε ότι η λογική κατάσταση της εισόδου της πύλης B αλλάζει από «Ψευδής» σε «Αληθής». Η κατάσταση εξόδου της πύλης B αλλάζει σε true, και ακολουθεί η είσοδος της πύλης E και η έξοδος αυτής. Η είσοδος και η έξοδος της πύλης G, επίσης, αλλάζει σε «Αληθής». Αυτή η αλληλουχία γεγονότων οδηγεί στην ενεργοποίηση του συναγερμού. Επειδή έχουν χρησιμοποιηθεί πύλες OR, απαιτείται μόνο μια είσοδος να αλλάξει σε «Αληθής» στα παράθυρα ή τις πόρτες, για να ενεργοποιηθεί ο συναγερμός. (Ryan, 2003; Predko, 2005)



Σχήμα 10.2. Κατάσταση ενεργοποίησης του συναγερμού

10.2. Παράδειγμα 2^ο (Βιομηχανική ασφάλεια)

Στη βιομηχανία η ασφαλής χρήσης των μηχανημάτων είναι πολύ σημαντική. Όλα τα μηχανήματα θα πρέπει να είναι στημένα κατά τέτοιο τρόπο, ώστε ο χειριστής της μηχανής να είναι αδύνατον να πάθει ατύχημα. Ένα μηχάνημα «Α», όταν ο προστατευτικός οδηγός δεν είναι στη θέση του, είναι μη ασφαλές γιατί είναι δυνατόν να μπει σε λειτουργία και να χρησιμοποιηθεί ειδικά αν χρησιμοποιείται από κάποιον απλό μηχανουργό. Αυτό σημαίνει ότι τα χέρια του χειριστή μπορεί να τραυματιστούν σοβαρά από την επικίνδυνη λεπίδα, κατά τη διάρκεια κοπής του υλικού. Εναλλακτικά, εάν σε ένα μηχάνημα «Β» έχει τοποθετηθεί ένα λογικό κύκλωμα, ώστε να διασφαλίζει ότι ο προστατευτικός οδηγός είναι στη σωστή θέση, και, συγχρόνως, να ενεργοποιείται με την χρήση ενός διακόπτη λειτουργίας «ON/OFF», πριν το μηχάνημα ξεκινήσει τη λειτουργία του. Αυτό σημαίνει ότι ο χειριστής πρέπει να κρατά με το ελεύθερο χέρι του το κουμπί λειτουργίας πατημένο, αλλιώς η ηλεκτρική παροχή του μηχανήματος θα διακοπεί, σταματώντας τη λειτουργία του. Όταν ο προστατευτικός οδηγός είναι στη σωστή θέση και ο μικρο-διακόπτης ενεργοποιηθεί και είναι σε κατάσταση «ON», οι λογικές καταστάσεις και των 2 εισόδων είναι «1», και, συνεπώς, η λογική κατάσταση της εξόδου είναι «1» και το μηχάνημα λειτουργεί κανονικά. (Χρήση πύλης AND)

10.3. Παράδειγμα 3^ο (Τάισμα κατοικίδιου)

Κατά τον σχεδιασμό ενός συστήματος ταΐσματος κατοικίδιου χρησιμοποιούνται εκτός των μηχανικών μερών και λογικές πύλες για να παίρνονται απλές αποφάσεις. Ένα παράδειγμα αυτού του είδους κυκλωμάτων είναι βασισμένο σε ολοκληρωμένο κύκλωμα που περιλαμβάνει 4 πύλες AND δύο εισόδων. Έστω λοιπόν ένας ιδιοκτήτης κατοικίδιου που θέλει να κατασκευάσει ένα αυτόματο μηχάνημα ταΐσματος, για να λειτουργεί τη νύχτα, όταν το κατοικίδιο πατήσει ένα διακόπτη (ένα αισθητήρα πίεσης). Αυτού του είδους η συσκευή θα ταΐζει αυτόματα το κατοικίδιο

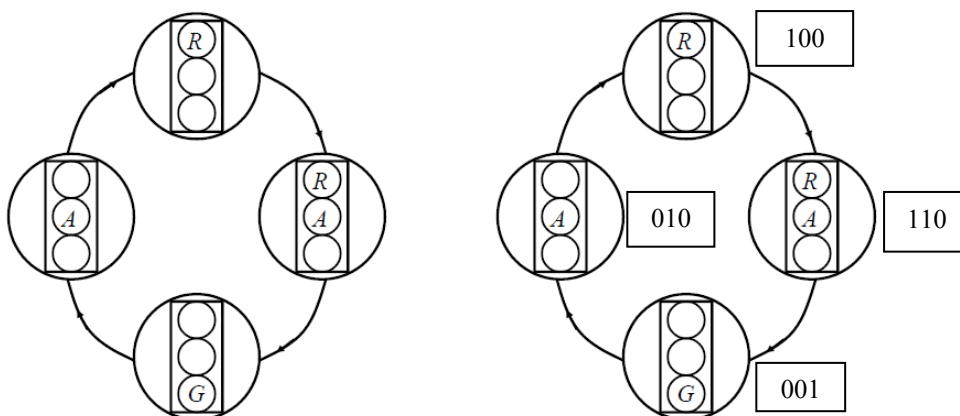
Το ολοκληρωμένο κύκλωμα που χρησιμοποιείται ανιχνεύει τότε οι δυο διακόπτες που χρησιμοποιούνται είναι ενεργοποιημένοι, και οι οποίοι αφορούν ο ένας το κατοικίδιο και ο άλλος τον έλεγχο εξωτερικού φωτισμού (εάν έχει βραδιάσει) – και ένας μικρός κινητήρας που επιτρέπει στο φαγητό να ελευθερωθεί από τον σωλήνα τροφοδοσίας του. Ένας μικρο-διακόπτης (αισθητήρας πίεσης) χρησιμοποιείται σαν η μια συσκευή εισόδου και ένα κύκλωμα ανίχνευσης φωτισμού σαν την δεύτερη συσκευή. Η πύλη AND που χρησιμοποιείται έχει δυο εισόδους. Αν και οι δυο εισοδοι είναι ενεργοποιημένες, δηλαδή και ο αισθητήρας φωτός και ο διακόπτης που ενεργοποιείται από το κατοικίδιο, η λογική κατάσταση της εξόδου αλλάζει σε high («1») και ο κινητήρας αφήνει το φαγητό να πέσει στο πιάτο του κατοικίδιου. Είναι προφανές από τον πίνακα αλήθειας της πύλης AND, πως, όταν ο διακόπτης έχει πατηθεί και υπάρχει εξωτερικός φωτισμός, ο κινητήρας είναι ανενεργός.

Το ολοκληρωμένο κύκλωμα περιλαμβάνει έναν αριθμό πυλών AND, και εδώ έχει χρησιμοποιηθεί μόνο μια από τις πύλες AND του ολοκληρωμένου. Ο διακόπτης 1 ΚΑΙ ο διακόπτης 2 πρέπει να πατηθούν, πριν ενεργοποιηθεί το ρελέ. (Παρατήρηση: Όταν τα ολοκληρωμένα χρησιμοποιούνται σε ένα κύκλωμα οι

είσοδοι από τις μη χρησιμοποιούμενες πύλες πρέπει να συνδέονται στο θετικό ή στο αρνητικό, χωρίς να έχει σημασία σε ποιο από τα δυο). (Ryan, 2003)

10.4. Παράδειγμα 4ο (Έλεγχος Φαναριού (Κοινωνική εφαρμογή))

Θα σχεδιάσουμε μια μηχανή Moore, για να υλοποιήσουμε έναν ελεγκτή του φαναριού. Για να οπτικοποιήσουμε το πρόβλημα, συχνά μας βοηθάει να σχεδιάσουμε ένα διάγραμμα με τις αλλαγές καταστάσεων. Αυτό χρησιμοποιείται, για να δημιουργήσουμε τον πίνακα μεταβάσεων, ο οποίος χρησιμοποιείται, για να παράξουμε τον πίνακα των παρούσων και επομένων καταστάσεων καθώς και την συνάρτηση εξόδου. (Nelson, et al., 1995; Katz, 2005; Wakerly, 2006; Tocci, et al., 2010)



Σχήμα 10.8. Καταστάσεις σηματοδότη και κωδικοποίηση των καταστάσεων λειτουργίας

Όπως διαπιστώνουμε, έχουμε 4 καταστάσεις. Άρα μπορούμε να χρησιμοποιήσουμε 2 f-fs. Παρόλα αυτά, χρησιμοποιώντας 3 f-fs, θα δούμε ότι δε χρειάζεται να υπολογίσουμε κάποια συνάρτηση εξόδου, και θα χρησιμοποιήσουμε τις 4 από τις 8 δυνατές καταστάσεις.

Γενικά η κωδικοποίηση των καταστάσεων είναι ένα δύσκολο πρόβλημα, και η βέλτιστη λύση δεν είναι πάντα προφανής. Χρησιμοποιώντας 3 F-Fs (θα χρησιμοποιήσουμε D f-f), μπορούμε να αναθέσουμε ένα σε κάθε μια από τις απαιτούμενες εξόδους (R, A, G), εξαλείφοντας για τη λογική εξόδου.

Αρχικά κατασκευάζουμε τον πίνακα μεταβάσεων. Έτσι στον πίνακα βλέπουμε ότι η επόμενη κατάσταση της κατάστασης 100 είναι η 110. Ονομάζουμε στις εξόδους των f-fs αντίστοιχα με τα χρώματα ως R, A και G. Θυμηθείτε ότι δε χρειάζεται να συμπεριλάβουμε στήλες για τη διέγερση των F-F, επειδή χρησιμοποιούμε D f-f, και σε αυτό είναι ταυτόσημες με την επόμενη κατάσταση. Άρα έχουμε τον ακόλουθο πίνακα μεταβάσεων.

Παρούσα Κατάσταση			Επομένη Κατάσταση		
R	A	G	R'	A'	G'
1	0	0	1	1	0
1	1	0	0	0	1
0	0	1	0	1	0
0	1	0	1	0	0
0	0	0	x	x	x
0	1	1	x	x	x
1	0	1	x	x	x
1	1	1	x	x	x

Αφού αυτές οι καταστάσεις, που αποδίδονται με ΑΟ, δε θα υπάρξουν ποτέ, δε μας απασχολεί τι έξοδο δίνει στην επόμενη κατάσταση. Αυτές οι αδιάφορες καταστάσεις μπορούν να χρησιμοποιηθούν μόνο για να απλοποιήσουμε τη συνάρτηση.

Τώρα πρέπει να προσδιορίσουμε τις συναρτήσεις. Για το R F-F πρέπει να προσδιορίσουμε το D_R . Για να γίνει αυτό θα χρησιμοποιήσουμε ένα χάρτη Karnaugh. (Το D είναι 1 όταν η επόμενη κατάσταση είναι 1). Ομαδοποιούμε τους άσους, ώστε να πάρουμε το βέλτιστο αποτέλεσμα. Οι αντίστοιχοι χάρτες Karnaugh και οι εξισώσεις, που προκύπτουν, είναι:

(Είναι προφανές ότι ανάλογα με την αρχική κωδικοποίηση λαμβάνουμε διαφορετικό αποτέλεσμα.)

	RA	00	01	11	10
G	0	AO	1		1
G	1		AO	AO	AO

$$D_R = R\bar{A} + \bar{R}A = R \oplus A$$

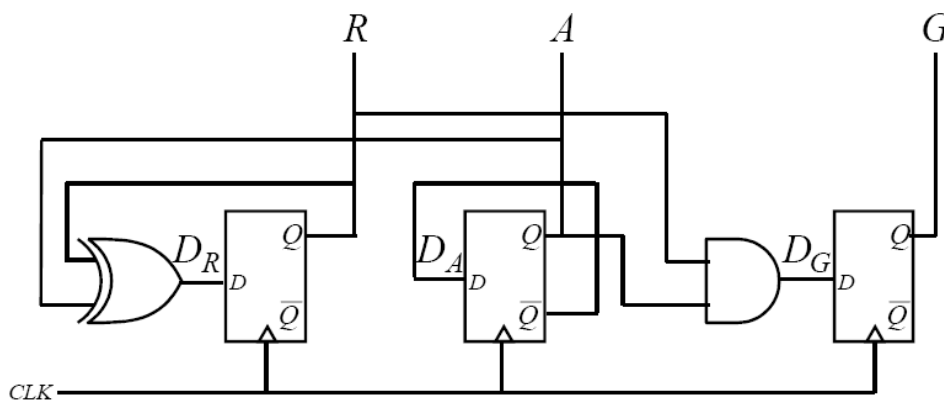
	RA	00	01	11	10
G	0	AO	1		1
G	1		AO	AO	AO

$$D_G = RA$$

	RA	00	01	11	10
G	0	AO			1
G	1	1	AO	AO	AO

$$D_A = \bar{A}$$

Χρησιμοποιώντας τις παραπάνω συναρτήσεις το τελικό διάγραμμα δίνεται από το ακόλουθο σχήμα 10.9:



Σχήμα 10.9. Διάγραμμα υλοποίησης του ελέγχου των φαναριών

Βιβλιογραφία-Αναφορές

Katz, R. (2005). *Contemporary Logic Design*, 2/e, Prentice Hall

Nelson, V., Nagle, H., Carroll, B., Irwin, J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall

Predko, Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill

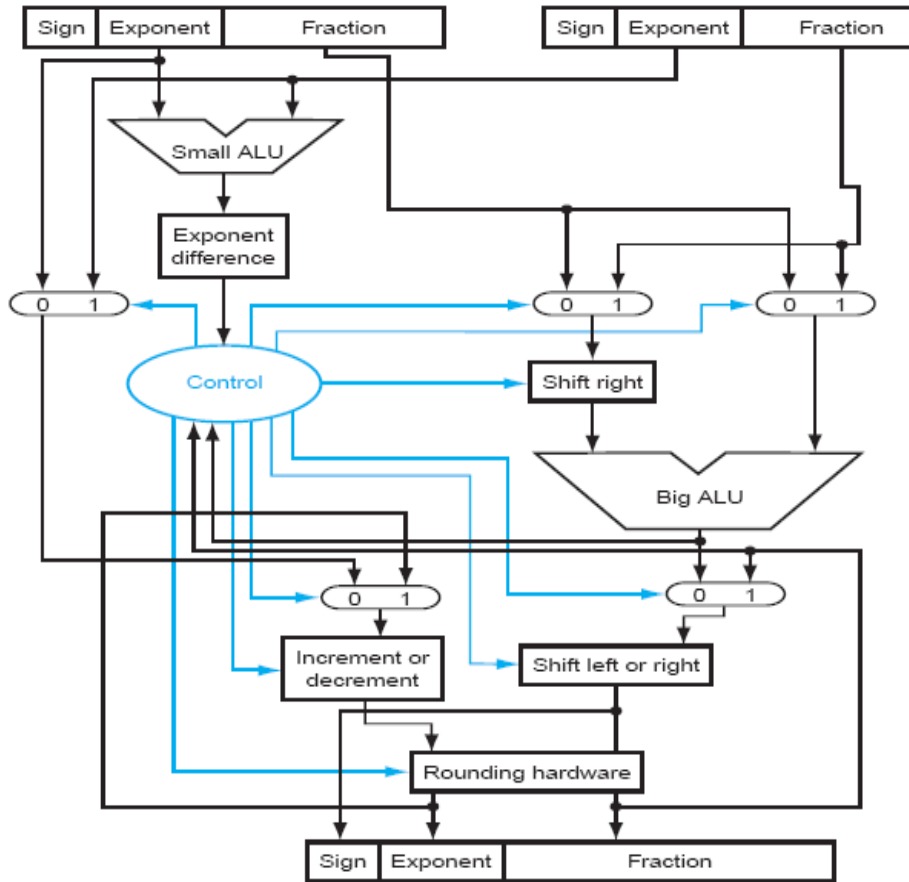
Ryan V., (2003), <http://www.technologystudent.com/elec1/elecex.htm>

Tocci, R. J., Widmer, N. S, and Moss, Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley

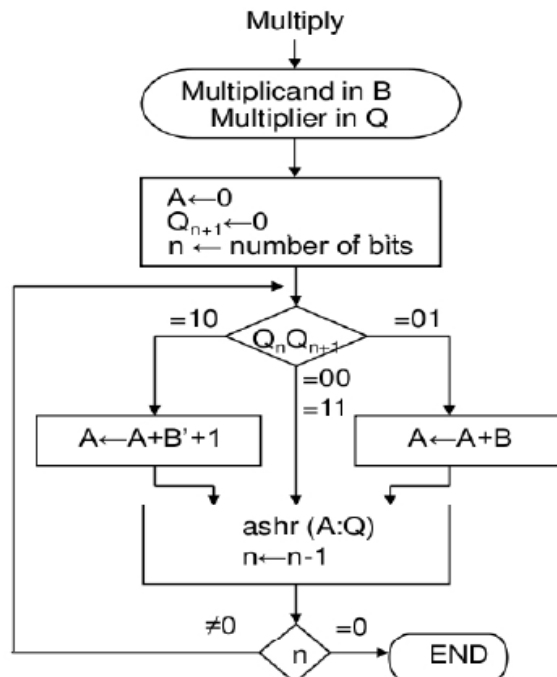
Wakerly, J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

ΠΑΡΑΡΤΗΜΑ

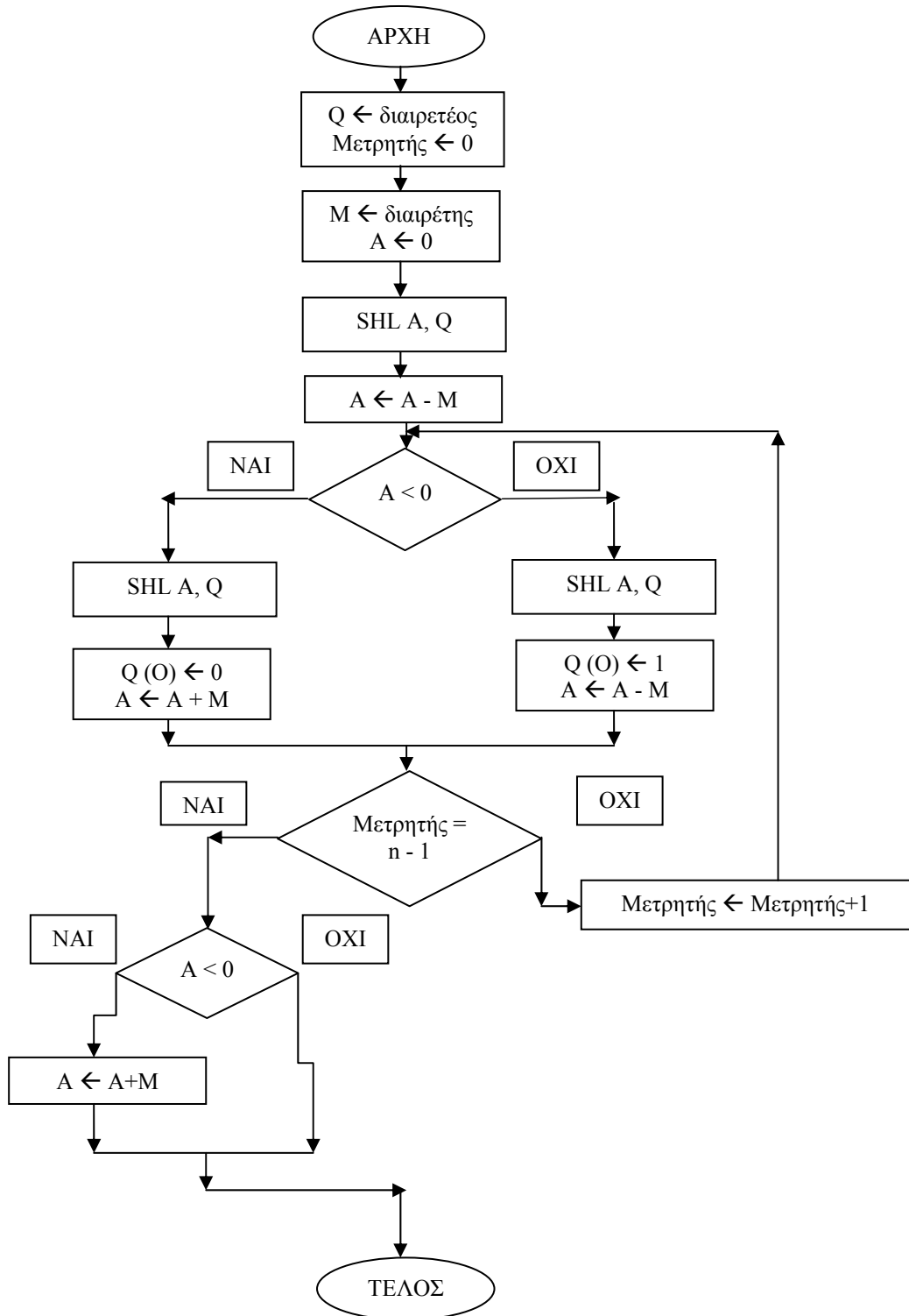
ΠΙΝΑΚΑΣ 1. Διάγραμμα πρόσθεσης αριθμών κινητής υποδιαστολής(IEEE)



ΠΙΝΑΚΑΣ 2. Διάγραμμα ροής του αλγόριθμου πολλαπλασιασμού κατά Booth



ΠΙΝΑΚΑΣ 3. Λογικό διάγραμμα διαίρεσης θετικών ακεραίων χωρίς ανάκτηση. Με SHL απεικονίζεται η αριστερή ολίσθηση, με Q το τελικό πηλίκο και με A το υπόλοιπο της διαίρεσης δοθέντων αριθμών (διααιρετέου και διαιρέτη)



ΠΙΝΑΚΑΣ 4. Αντιστοιχία εξαψήφιου αλφαριθμητικού κώδικα και κώδικα EBCDIC

Στοιχείο	Δυαδική παράσταση	EBCDIC	Στοιχείο	Δυαδική παράσταση	EBCDIC
0	001010	11110000	A	110001	11000001
1	000001	11110001	B	110010	11000010
2	000010	11110010	C	110011	11000011
3	000011	11110011	D	110100	11000100
4	000100	11110100	E	110101	11000101
5	000101	11110101	F	110110	11000110
6	000110	11110110	G	110111	11000111
7	000111	11110111	H	111000	11001000
8	001000	11111000	I	111001	11001001
9	001001	11111001	J	100001	11010001
.	111011	01001011	K	100010	11010010
(111101	01001101	L	100011	11010011
<	111110	01001100	M	100100	11010100
\$	101011	01011011	N	100101	11010101
*	101100	01011100	O	100110	11010110
)	101101	01011101	P	100111	11010111
;	101110	01011110	Q	101000	11011000
-	100000	01100000	R	101001	11011001
/	010001	01100001	S	010010	11100010
,	011011	01101011	T	010011	11100011
%	011100	01101100	U	010100	11100100
\	011110		V	010101	11100101
:	001101	01111010	W	010110	11100110
>	001110	01101110	X	010111	11100111
?	111010	01101111	Y	011000	11101000
!	101110	01011010	Z	011001	11101001
'		01111101	κενό	000000	
..		01111111			

ΠΙΝΑΚΑΣ 5. Διεθνής κώδικας ASCII

Στοιχείο	Δυαδική παράσταση	Στοιχείο	Δυαδική παράσταση	Στοιχείο	Δυαδική παράσταση	Στοιχείο	Δυαδική παράσταση
NUL	00000000	SP	00100000	@	01000000	`	01100000
SOH	00000001	!	00100001	A	01000001	a	01100001
STX	00000010	"	00100010	B	01000010	b	01100010
ETX	00000011	#	00100011	C	01000011	c	01100011
EOT	00000100	\$	00100100	D	01000100	d	01100100
ENQ	00000101	%	00100101	E	01000101	e	01100101
ACK	00000110	&	00100110	F	01000110	f	01100110
BEL	00000111	'	00100111	G	01000111	g	01100111
BS	00001000	(00101000	H	01001000	h	01101000
HT	00001001)	00101001	I	01001001	i	01101001
LF	00001010	*	00101010	J	01001010	j	01101010
VT	00001011	+	00101011	K	01001011	k	01101011
FF	00001100	,	00101100	L	01001100	l	01101100
CR	00001101	-	00101101	M	01001101	m	01101101
SO	00001110	.	00101110	N	01001110	n	01101110
SI	00001111	/	00101111	O	01001111	o	01101111
DLE	00010000	0	00110000	P	01010000	p	01110000
DC1	00010001	1	00110001	Q	01010001	q	01110001
DC2	00010010	2	00110010	R	01010010	r	01110010
DC3	00010011	3	00110011	S	01010011	s	01110011
DC4	00010100	4	00110100	T	01010100	t	01110100
NAK	00010101	5	00110101	U	01010101	u	01110101
SYN	00010110	6	00110110	V	01010110	v	01110110
ETB	00010111	7	00110111	W	01010111	w	01110111
CAN	00011000	8	00111000	X	01011000	x	01111000
EM	00011001	9	00111001	Y	01011001	y	01111001
SUB	00011010	:	00111010	Z	01011010	z	01111010
ESC	00011011	;	00111011	[01011011	{	01111011
FS	00011100	<	00111100	\	01011100	 	01111100
GS	00011101	=	00111101]	01011101	}	01111101
RS	00011110	>	00111110	^	01011110	~	01111110
US	00011111	?	00111111	_	01011111	□	01111111

ΠΙΝΑΚΑΣ 6. Τοπικός κώδικας ASCII- (ISO 10646)

Στοιχείο	Διαδική παράσταση	Στοιχείο	Διαδική παράσταση	Στοιχείο	Διαδική παράσταση	Στοιχείο	Διαδική παράσταση
	10000000		10100000	İ	11000000	ÿ	11100000
	10000001		10100001	A	11000001	α	11100001
,	10000010	Α	10100010	B	11000010	β	11100010
f	10000011	£	10100011	Γ	11000011	γ	11100011
„	10000100		10100100	Δ	11000100	δ	11100100
...	10000101	¥	10100101	E	11000101	ε	11100101
†	10000110		10100110	Z	11000110	ζ	11100110
‡	10000111	§	10100111	H	11000111	η	11100111
	10001000	..	10101000	Θ	11001000	θ	11101000
‰	10001001	©	10101001	I	11001001	ι	11101001
	10001010		10101010	K	11001010	κ	11101010
<	10001011	«	10101011	Λ	11001011	λ	11101011
	10001100	¬	10101100	M	11001100	μ	11101100
	10001101	-	10101101	N	11001101	ν	11101101
	10001110	®	10101110	Ξ	11001110	ξ	11101110
	10001111		10101111	O	11001111	ο	11101111
	10010000	◦	10110000	Π	11010000	π	11110000
‘	10010001	±	10110001	P	11010001	ρ	11110001
’	10010010	²	10110010		11010010	ς	11110010
«	10010011	³	10110011	Σ	11010011	σ	11110011
«	10010100	´	10110100	T	11010100	τ	11110100
•	10010101	μ	10110101	Υ	11010101	υ	11110101
—	10010110	¶	10110110	Φ	11010110	φ	11110110
—	10010111	·	10110111	X	11010111	χ	11110111
	10011000	Έ	10111000	Ψ	11011000	ψ	11111000
™	10011001	Η	10111001	Ω	11011001	ω	11111001
	10011010	Ι	10111010	İ	11011010	ï	11111010
>	10011011	»	10111011	ÿ	11011011	ÿ	11111011
	10011100	Ό	10111100	ά	11011100	ό	11111100
	10011101	½	10111101	έ	11011101	έ	11111101
	10011110	Υ	10111110	ή	11011110	ώ	11111110
	10011111	Ω	10111111	ί	11011111		11111111

Αντιστοίχιση Βασικών Ελληνικών Όρων στην Αγγλική

Αδιάφορος (όρος)	Don't care state
Άθροισμα Γινομένων	Sum of products (SOP)
Ακολουθιακά Κυκλώματα	Sequential Circuits
Αμφίδρομος απαριθμητής	Up Down Counter
Αναπαράσταση προσήμου	Sign representation
Απαριθμητές ως προς μέτρο	Modulo counters
Απαριθμητής δακτυλίου	Ring Counters
Απαριθμητής Κυμάτων	Ripple counter
Αποδιαμόρφωση	Demodulation
Αποκλειστικής Διάζευξης	Exclusive OR (XOR)
Αποκλειστικής Σύζευξης	Coincidence (XNOR)
Αποκωδικοποιητής	Decoder
Απομονωτής	Buffer
Απόσταση Hamming	Hamming Distance
Αριθμητική και Λογική Μονάδα (ΑΛΜ)	Arithmetic Logic Unit(ALU)
Βάση αριθμητικού συστήματος	Base, Radix
Γεννήτρια Ψευδοτυχαίων αριθμών	Pseudo-random generator
Γινόμενο Άθροισμάτων	Product of Sums(POS)
Δυαδικοί απαριθμητές	Binary Counters
Διαίρεση με ανάκτηση	Restoring division
Διαίρεση χωρίς ανάκτηση	Non restoring division
Διακόπτης	Switch
Διαμόρφωση	Modulation
Διπλή Ακρίβεια	Double precision
Δισταθής πολυδονητής	Bistable multivibrator
Δυαδικό Σύστημα e	Binary System
Δυδεκαδικός κώδικας	Bindecimal code
Δυπενταδικός κώδικας	Biquinary code
Είσοδος κρατουμένου	Carry Input
Εκθέτης	Exponent
Ελαχιστοβάθμιος όρος (ΕΛΒΟ)	Minterm
Ενεργοποίηση	Enable
Ημισθροιστής	Half Adder
Θεώρημα της επέκτασης	Expansion theorem
Κανόνας του Humphrey	First condition of Humphrey
Καταχωρητής	Register
Καταχωρητής Ολίσθησης	Shift register
Κεντρική Μονάδα Επεξεργασίας(KME)	Central Processing Unit (CPU)
Κινητή Υποδιαστολή	Floating Point
Κλάσμα	Fraction
Κυκλική μεταφορά κρατουμένου	End around carry
Κυκλικός Πλεονάζων Κώδικας Ελέγχου	Cyclic Redundant Check Code
Κύριο εξαρτώμενο flip-flop	Master slave flip-flop
Κώδικας Υπερβολής	Excess code
Κωδικοποιητής	Coder
Μέγεθος	Mantissa
Μεγιστοβάθμιος όρος(MEBO)	Maxterm
Μηχανές Καταστάσεων	State Machines
Μονοσταθής πολυδονητής	Monostable multivibrator, One shot
Ουσιώδης όρος	Essential Term
Παράλληλος αθροιστής	Parallel Full adder
Πίνακας Αλήθειας	Truth Table
Πίνακας Lookup	Look up Table (LUT)
Πλήρης αθροιστής	Full adder
Πολυπλέκτης	Multiplexer
Πρόγνωση κρατουμένου	Carry Look Ahead
Προγραμματιζόμενες Λογικές Μήτρες	Programming Logic Arrays

Προγραμματιζόμενες Λογικές Μονάδες
Προσημασμένη Διαίρεση
Πρόσημο
Σειριακός Πλήρης Αθροιστής
Συγκεραστικοί κώδικες
Συγκριτής
Συμπλήρωμα ως προς 1
Συμπλήρωμα ως προς 2
Συνδυαστικά Κυκλώματα
Χάρτης Karnaugh
Χρονοδιακόπτης
Ψηφίο ισοτιμίας
Ωρολογιακό flip-flop

Programming Logic Devices
Signed division
Sign
Serial Full adder
Convolutional codes
Comparator
1's complement
2's complement
Combinational Circuits
Karnaugh Map
Timmer
Parity bit
Clocked flip-flop

Αλφαβητάρι Βασικών Όρων

D flip-flop, 187
Fan-out, 265
J-K flip-flop, 183
Master Slave flip-flop, 189
PLA, 288
RAM 4x4, 312
R-S (Reset- Set) *f-f*, 182
R-S-T flip-flop, 188
T flip-flop, 186

A

Αδιάφοροι όροι, 94
Αθροιστής δύο τετρανήφιων αριθμών, 121
Αθροιστής με πολυπλέκτη, 268
Αθροιστής των 4-bit, 319
Άλγεβρα Boole, 73
Αμφίδρομος απαριθμητής, 219
Αναγνώριση μορφών, 231
Ανάλυση κυκλωμάτων, 112
Αναστροφέας οικογένειας CMOS, 159
Απαριθμητές ως προς μέτρο, 202
Απαριθμητής BCD 2421, 201
Απαριθμητής Johnson, 205
Απαριθμητής δακτυλίου, 205
Αποκωδικοποιητές, 258
Αποκωδικοποιητής 2-σε-4, 260
Αποκωδικοποιητής 3-σε-8, 261
Απομονωτές, 270
Αριθμοί απλής ακριβείας, 63
Αριθμοί διπλής ακριβείας, 63
Αριθμοί κινητής υποδιαστολής, 62
Ασταθής πολυδονητής, 166
Αφαίρεση δυαδικών αριθμών, 31

Γ

Γεννήτρια ψευδοτυχαίων δυαδικών αριθμών, 207

Δ

Δεκαδικό σύστημα, 18
Δεκαεξαδικό σύστημα αρίθμησης, 46
Διάγραμμα Venn, 86
Διάγραμμα καταστάσεων, 208
Διαίρεση δυαδικών αριθμών, 34
Διαίρεση με ανάκτηση, 35
Διαίρεση χωρίς ανάκτηση, 35
Δικτυακές λειτουργίες, 326
Δισταθής πολυδονητής *f-f*, 172
Δομή Harvard, 328
Δομή Von Neumann, 327
Δυαδικοί απαριθμητές, 194
Δυαδικός ημιαθροιστής, 114
Δυαδικός κώδικας BCD, 47
Δυαδικός πολλαπλασιαστής 2x2, 125
Δυδεκαδικός κώδικας, 48
Δυπενταδικός κώδικας, 48

E

Ελαχιστοβάθμιος όρος (EΛΒΟ), 82

Θ

Θεωρήματα De Morgan, 75

K

Καθυστέρηση διάδοσης, 265
Κανονικοποίηση, 85
Καταχωρητές ολίσθησης, 190
Καταχωρητές Ολίσθησης, 304
Καταχωρητής 4 δυαδικών ψηφίων, 302
Κυκλικός Πλεονάζων Κώδικας Ελέγχου, 52
Κώδικας 2 -από -5 (2-out of-5), 49
Κώδικας Gray, 49
Κώδικας υπερβολής κατά τρία (Excess three, XS3), 48
Κωδικοποιητής δεκαδικού με διακόπτες, 164
Κυκλική μεταφορά κρατουμένου, 38
Κύκλωμα αφαίρεσης, 124
Κύκλωμα κωδικοποίησης με διακριτά στοιχεία, 163
Κύκλωμα πύλης NAND οικογένειας TTL, 158
Κυκλώματα Προσαρμογής, 249
Κώδικας 5311, 47
Κώδικας 5421, 47
Κώδικας Aiken, 47
Κώδικας Hamming, 53

Λ

Λογική Μονάδα, 323

M

Μη σειριακός απαριθμητής, 216
Μνήμη ROM, 316
Μεγιστοβάθμιος όρος (MEBO), 83
Μέθοδος Quine-Mc Cluskey, 92
Μετατροπείς D/A για την περιστροφή αξόνων, 251
Μετατροπείς αναλογικών σημάτων σε ψηφιακά, 252
Μετατροπείς ψηφιακών σημάτων σε αναλογικά, 249
Μετατροπή από IEEE-754 σε δεκαδικό αριθμό, 64
Μετατροπή αριθμού από το δυαδικό στο δεκαδικό, 22
Μετατροπή δεκαδικού αριθμού σε IEEE-754, 64
Μετατροπή δεκαδικού κλασματικού αριθμού, 21
Μετατροπή της πληροφορίας γωνίας άξονα σε ψηφιακή, 256
Μετρητής αυτόματης εκκίνησης, 239
Μετρητής δύο δυαδικών ψηφίων, 237
Μηχανή Καταστάσεων Mealy, 226
Μηχανή Καταστάσεων Moore, 222
Μνήμη PROM, 282
Μνήμη RAM, 307
MODEM, 271
Μονοσταθής πολυδονητής, 169
Μονοσταθής πολυδονητής με πύλες, 171
Μοντέλο μετατροπής και απεικόνισης ryian, 19

Ο

Οκταδικό σύστημα αρίθμησης, 45
Ολισθητής με R-S *flip-flop*, 191

Π

Παράλληλος πλήρης αθροιστής, 119
Παράσταση διακοπών, 77
Παράσταση θετικών και αρνητικών δυαδικών αριθμών με συμπλήρωμα ως προς 1, 26
Παράσταση θετικών και αρνητικών δυαδικών αριθμών με συμπλήρωμα ως προς 2, 27
Πλήρης αθροιστής, 116
Πλήρης αθροιστής με PROM, 284
Πλήρης αθροιστής με διπλό πολυπλέκτη, 269
Πολλαπλασιασμός αριθμών κινητής υποδιαστολής, 66
Πολλαπλασιασμός δυαδικών αριθμών, 31
Πολλαπλασιασμός θετικών δυαδικών αριθμών, 32
Πολλαπλασιασμός προσημασμένων δυαδικών (αρνητικός πολλαπλασιαστέος), 33
Πολλαπλασιασμός προσημασμένων δυαδικών (αρνητικός πολλαπλασιαστής), 34
Πολλαπλασιασμός προσημασμένων δυαδικών αριθμών, 33
Πολυδονητής δύο καταστάσεων με πύλες, 181
Πολυπλέκτης, 265
Πολυπλέκτης 8 σε 1, 267
Προγραμματιζόμενες λογικές διατάξεις, 278
Προγραμματιζόμενες λογικές διατάξεις(PAL), 291
Προγραμματιζόμενες λογικές μονάδες, 288
Προσημασμένη παράσταση θετικών και αρνητικών δυαδικών αριθμών, 24
Προσημασμένη Υπερχείλιση, 40
Πρόσθεση αριθμών κινητής υποδιαστολής, 65
Πρόσθεση αριθμών με συμπλήρωμα ως προς δύο, 38
Πρόσθεση αριθμών με συμπλήρωμα ως προς ένα, 36
Πρόσθεση δυαδικών αριθμών, 30
Πρόσθεση με τη χρήση αποκωδικοποιητών, 262
Πρότυπο IEEE-754, 61

Πύλες AOI, 110

Πύλες OAI, 110

Πύλη AND με διόδους, 147

Πύλη AND με διπολικά τρανζίστορ, 149

Πύλη **NAND**, 106

Πύλη NAND με διόδους και τρανζίστορ, 153

Πύλη NAND με τρανζίστορ, 151, 152

Πύλη NAND οικογένειας CMOS, 160

Πύλη **NOR**, 106

Πύλη NOR με τρανζίστορ, 151

Πύλη OR / NOR της οικογένειας ECL, 161

Πύλη OR με διόδους, 146

Πύλη OR με διπολικά τρανζίστορ, 149

Πύλη **XNOR**, 110

Πύλη XOR με διακριτά στοιχεία, 154

Πύλη αποκλειστικής διάζευξης XOR, 108

Πύλη λογικής άρνησης **NOT**, 104

Πύλη λογικής πρόσθεσης **OR**, 104

Πύλη λογικού γινομένου **AND**, 104

Σ

Σειριακός δυαδικός συγκριτής, 127

Σειριακός πλήρης αθροιστής, 119

Σειριακός πλήρης αφαιρέτης, 123

Σύγχρονος δυαδικός απαριθμητής, 196

Συμπληρώματα αριθμών, 23

Συνδυαστικά κυκλώματα, 208

Σχεδίαση αποκωδικοποιητή 4 σε 10 βασισμένη σε PROM 16x10, 287

Υ

Υβριδικά CMOS memristor λογικής, 162

X

Χάρτης Karnaugh, 87

Ω

Ωρολογιακό J-K flip-flop, 184