

ΕΚΔΟΣΕΙΣ "ΚΛΕΙΔΑΡΙΘΜΟΣ"

Τρίτη Αμερικανική έκδοση
ενημερωμένη

ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ

ΑΡΧΕΣ & ΠΡΑΚΤΙΚΕΣ



JOHN F. WAKERLY

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΙΚΟ ΣΗΜΕΙΩΜΑ	xvii
ΠΡΟΛΟΓΟΣ	xix
1 Εισαγωγή	1
1.1 Λίγα λόγια για την ψηφιακή σχεδίαση	1
1.2 Αναλογικά και ψηφιακά συστήματα	4
1.3 Ψηφιακές διατάξεις	8
1.4 Ζητήματα ηλεκτρονικής στην ψηφιακή σχεδίαση	9
1.5 Ζητήματα λογισμικού στην ψηφιακή σχεδίαση	10
1.6 Ολοκληρωμένα κυκλώματα	15
1.7 Προγραμματιζόμενες λογικές διατάξεις	19
1.8 Ολοκληρωμένα κυκλώματα εξειδικευμένα για εφαρμογές	20
1.9 Πλακέτες τυπωμένων κυκλωμάτων	22
1.10 Επίπεδα της ψηφιακής σχεδίασης	23
1.11 Η ουσία της υπόθεσης	28
1.12 Το επόμενο βήμα	29
Προβλήματα για εξάσκηση	30
2 Αριθμητικά συστήματα και κώδικες	31
2.1 Αριθμητικά συστήματα θέσης	31
2.2 Οκταδικοί και δεκαεξαδικοί αριθμοί	33
2.3 Γενικές μετατροπές στα αριθμητικά συστήματα θέσης	36
2.4 Πρόσθεση και αφαίρεση μη δεκαδικών αριθμών	37
2.5 Αναπαράσταση αρνητικών αριθμών	40
2.5.1 Αναπαράσταση προσημασμένου μεγέθους	40
2.5.2 Αριθμητικά συστήματα συμπληρώματος	42
2.5.3 Αναπαράσταση συμπληρώματος ως προς τη βάση	42
2.5.4 Αναπαράσταση συμπληρώματος ως προς δύο	44
2.5.5 Αναπαράσταση συμπληρώματος ως προς τη βάση πλην ένα	45
2.5.6 Αναπαράσταση συμπληρώματος ως προς ένα	45
2.5.7 Αναπαραστάσεις υπέρβασης	46

2.6	Πρόσθεση και αφαίρεση αριθμών συμπληρώματος ως προς δύο	47
2.6.1	<i>Κανόνες πρόσθεσης</i>	47
2.6.2	<i>Γραφική αναπαράσταση</i>	48
2.6.3	<i>Υπερχείλιση</i>	49
2.6.4	<i>Κανόνες αφαίρεσης</i>	49
2.6.5	<i>Αριθμοί συμπληρώματος ως προς δύο και απρόσημοι δυαδικοί αριθμοί</i>	50
2.7	Πρόσθεση και αφαίρεση αριθμών συμπληρώματος ως προς ένα	53
2.8	Πολλαπλασιασμός δυαδικών	54
2.9	Διαίρεση δυαδικών	56
2.10	Δυαδικοί κώδικες δεκαδικών αριθμών	57
2.11	Κώδικας Gray	60
2.12	Κώδικες χαρακτήρων	62
2.13	Κώδικες για ενέργειες, συνθήκες και καταστάσεις	63
2.14	<i>n</i> -διάστατοι κύβοι και απόσταση	67
2.15	Κώδικες για εντοπισμό και διόρθωση σφαλμάτων	68
2.15.1	<i>Κώδικες ανίχνευσης σφαλμάτων</i>	68
2.15.2	<i>Κώδικες διόρθωσης σφαλμάτων και κώδικες ανίχνευσης πολλαπλών σφαλμάτων</i>	70
2.15.3	<i>Κώδικες Hamming</i>	72
2.15.4	<i>Κώδικες CRC</i>	76
2.15.5	<i>Διδιάστατοι κώδικες</i>	76
2.15.6	<i>Κώδικες αθροίσματος ελέγχου</i>	79
2.15.7	<i>Κώδικες m από n</i>	79
2.16	Κώδικες για σειριακή μετάδοση και αποθήκευση	80
2.16.1	<i>Παράλληλα και σειριακά δεδομένα</i>	80
2.16.2	<i>Κώδικες σειριακής γραμμής</i>	81
	Παραπομπές	85
	Προβλήματα για εξάσκηση	86
	Ασκήσεις	88
3	Ψηφιακά κυκλώματα	93
3.1	Λογικά σήματα και πύλες	94
3.2	Οικογένειες λογικών κυκλωμάτων	99
3.3	Λογικά κυκλώματα CMOS	101
3.3.1	<i>Λογικά επίπεδα CMOS</i>	101
3.3.2	<i>Τρανζίστορ MOS</i>	102
3.3.3	<i>Βασικό κύκλωμα αντιστροφέα CMOS</i>	104
3.3.4	<i>Πύλες CMOS NAND και NOR</i>	106
3.3.5	<i>Μέγιστο πλήθος εισόδων (Fan-In)</i>	109
3.3.6	<i>Μη αναστρέφουσες πύλες</i>	110
3.3.7	<i>Πύλες CMOS AND-OR-INVERT και OR-AND-INVERT</i>	111
3.4	Ηλεκτρική συμπεριφορά κυκλωμάτων CMOS	113
3.4.1	<i>Επισκόπηση</i>	113
3.4.2	<i>Φύλλα δεδομένων και προδιαγραφές</i>	115
3.5	Ηλεκτρική συμπεριφορά σταθερής κατάστασης στα κυκλώματα CMOS	117
3.5.1	<i>Λογικά επίπεδα και περιθώρια θορύβου</i>	118
3.5.2	<i>Συμπεριφορά κυκλώματος με ωμικά φορτία</i>	121
3.5.3	<i>Συμπεριφορά κυκλώματος με μη ιδανικές εισόδους</i>	127

3.5.4	Μέγιστο πλήθος εξόδων	128
3.5.5	Τα αποτελέσματα της υπερφόρτωσης	130
3.5.6	Αχρησιμοποίητες είσοδοι	131
3.5.7	Αιχμές ρεύματος και πυκνωτές απόξευξης	132
3.5.8	Πώς να καταστρέψετε μια διάταξη CMOS	132
3.6	Δυναμική ηλεκτρική συμπεριφορά των κυκλωμάτων CMOS	134
3.6.1	Χρόνος μετάβασης	135
3.6.2	Καθυστέρηση διάδοσης	141
3.6.3	Κατανάλωση ισχύος	143
3.7	Άλλες δομές εισόδου και εξόδου CMOS	145
3.7.1	Πύλες μετάδοσης	145
3.7.2	Είσοδοι σκανδάλης Schmitt	146
3.7.3	Έξοδοι τριών καταστάσεων	148
3.7.4	Έξοδοι ανοικτής υποδοχής	151
3.7.5	Οδήγηση των LED	153
3.7.6	Διάυλοι πολλών πηγών σήματος	156
3.7.7	Καλωδιωμένη λογική	157
3.7.8	Ελκτικές προς τα πάνω αντιστάσεις	157
3.8	Λογικές οικογένειες CMOS	160
3.8.1	HC και HCT	160
3.8.2	VHC και VHCT	161
3.8.3	Ηλεκτρικά χαρακτηριστικά των HC, HCT, VHC, και VHCT	162
3.8.4	Οικογένειες FCT και FCT-T	169
3.8.5	Τα ηλεκτρικά χαρακτηριστικά της οικογένειας FCT-T	169
3.9	Κυκλώματα διπολικής λογικής	171
3.9.1	Δίοδοι	172
3.9.2	Λογική διόδων	175
3.9.3	Διπολικά τρανζίστορ επαφής	177
3.9.4	Λογικός αντιστροφέας με τρανζίστορ	182
3.9.5	Τρανζίστορ Schottky	183
3.10	Λογική τρανζίστορ-τρανζίστορ	184
3.10.1	Βασική πύλη TTL NAND	185
3.10.2	Λογικά επίπεδα και περιθώρια θορύβου	188
3.10.3	Μέγιστο πλήθος εξόδων (Fanout)	190
3.10.4	Μη χρησιμοποιούμενες είσοδοι	193
3.10.5	Άλλοι τύποι πυλών TTL	195
3.11	Οικογένειες κυκλωμάτων TTL	197
3.11.1	Οι πρώτες οικογένειες TTL	197
3.11.2	Οικογένειες Schottky TTL	198
3.11.3	Χαρακτηριστικά οικογενειών TTL	199
3.11.4	Φύλλο δεδομένων μιας πύλης TTL	199
3.12	Διασύνδεση κυκλωμάτων CMOS με TTL	202
3.13	Λογικά κυκλώματα CMOS χαμηλής τάσης και διασύνδεση	203
3.13.1	Λογικές οικογένειες 3.3V LVTTTL και LVCMOS	204
3.13.2	Είσοδοι ανεκτικές στα 5V	205
3.13.3	Έξοδοι ανεκτικές στα 5V	207
3.13.4	Σύνοψη της διασύνδεσης TTL με LVTTTL	208
3.13.5	Λογικά κυκλώματα στα 2,5V και στα 1,8V	208
3.14	Λογική ζεύξης εκπομπού	209
3.14.1	Το βασικό κύκλωμα CML	210

3.14.2	Οικογένειες ECL 10K/10H	213
3.14.3	Οικογένεια ECL 100K	216
3.14.4	Θετικά ECL (PECL)	217
	Παραπομπές	218
	Προβλήματα για εξάσκηση	220
	Ασκήσεις	224
4	Αρχές συνδυαστικής λογικής σχεδίασης	231
4.1	Άλγεβρα μεταγωγής	233
4.1.1	Αξιώματα	233
4.1.2	Θεωρήματα μίας μεταβλητής	236
4.1.3	Θεωρήματα δύο και τριών μεταβλητών	237
4.1.4	Θεωρήματα n μεταβλητών	239
4.1.5	Δυικότητα	242
4.1.6	Καθιερωμένες αναπαραστάσεις λογικών συναρτήσεων	245
4.2	Ανάλυση συνδυαστικών κυκλωμάτων	249
4.3	Σύνθεση συνδυαστικών κυκλωμάτων	255
4.3.1	Περιγραφές και σχεδιάσεις κυκλωμάτων	255
4.3.2	Χειρισμοί κυκλωμάτων	258
4.3.3	Ελαχιστοποίηση συνδυαστικών κυκλωμάτων	261
4.3.4	Χάρτες Karnaugh	263
4.3.5	Ελαχιστοποίηση αθροισμάτων γινομένων	264
4.3.6	Απλοποίηση γινομένων αθροισμάτων	273
4.3.7	Αδιάφοροι συνδυασμοί εισόδων	275
4.3.8	Ελαχιστοποίηση πολλών εξόδων	276
4.4	Προγραμματιζόμενες μέθοδοι ελαχιστοποίησης	279
4.4.1	Αναπαράσταση όρων γινομένου	280
4.4.2	Εύρεση πρωταρχικών όρων με συνδυασμό όρων γινομένου	284
4.4.3	Εύρεση ελάχιστης κάλυψης με χρήση του πίνακα πρωταρχικών όρων	286
4.4.4	Άλλες μέθοδοι ελαχιστοποίησης	288
4.5	Κίνδυνοι χρονισμού	289
4.5.1	Στατικοί κίνδυνοι	290
4.5.2	Εύρεση στατικών κινδύνων με χρήση χαρτών	291
4.5.3	Δυναμικοί κίνδυνοι	293
4.5.4	Σχεδίαση κυκλωμάτων χωρίς κινδύνους	294
4.6	Γλώσσα περιγραφής υλικού ABEL	295
4.6.1	Δομή προγράμματος ABEL	296
4.6.2	Λειτουργία του μεταγλωττιστή ABEL	299
4.6.3	Εντολές WHEN και ομάδες εξισώσεων	300
4.6.4	Πίνακες αληθείας	304
4.6.5	Πεδία τιμών, σύνολα και σχέσεις	305
4.6.6	Αδιάφορες εισοδοί	308
4.6.7	Διανύσματα έλεγχου	310
4.7	Γλώσσα περιγραφής υλικού VHDL	313
4.7.1	Σχεδιαστική ροή	314
4.7.2	Δομή του προγράμματος	318
4.7.3	Τύποι και σταθερές	323
4.7.4	Συναρτήσεις και διαδικασίες	329
4.7.5	Βιβλιοθήκες και πακέτα	334

4.7.6	Στοιχεία δομικής σχεδίασης	337
4.7.7	Στοιχεία σχεδίασης ροής δεδομένων	342
4.7.8	Στοιχεία σχεδίασης σύμφωνα με τη συμπεριφορά	345
4.7.9	Η χρονική διάσταση και η προσομοίωση	353
4.7.10	Σύνθεση	356
	Παραπομπές	357
	Προβλήματα για εξάσκηση	361
	Ασκήσεις	364
5	Πρακτικές της συνδυαστικής λογικής σχεδίασης	371
5.1	Πρότυπα τεκμηρίωσης	372
5.1.1	Δομικά διαγράμματα	375
5.1.2	Σύμβολα πυλών	377
5.1.3	Ονόματα σημάτων και ενεργά επίπεδα	378
5.1.4	Ενεργά επίπεδα των ακροδεκτών	380
5.1.5	Λογική σχεδίαση “από φυσαλίδα σε φυσαλίδα”	382
5.1.6	Διάταξη σχεδίου	386
5.1.7	Διάυλοι	390
5.1.8	Συμπληρωματικές πληροφορίες σχηματικών διαγραμμάτων	392
5.2	Χρονισμός κυκλωμάτων	394
5.2.1	Διαγράμματα χρονισμού	394
5.2.2	Καθυστέρηση διάδοσης	397
5.2.3	Προδιαγραφές χρονισμού	397
5.2.4	Ανάλυση χρονισμού	401
5.2.5	Εργαλεία ανάλυσης χρονισμού	402
5.3	Συνδυαστικές προγραμματιζόμενες λογικές διατάξεις (PLD)	403
5.3.1	Προγραμματιζόμενοι λογικοί πίνακες	403
5.3.2	Διατάξεις PAL (προγραμματιζόμενη λογική πίνακα)	406
5.3.3	Διατάξεις γενικής λογικής πίνακα (GAL)	410
5.3.4	Διπολικά κυκλώματα προγραμματιζόμενης λογικής διάταξης	413
5.3.5	Κυκλώματα προγραμματιζόμενων λογικών διατάξεων CMOS	414
5.3.6	Προγραμματισμός και δοκιμή διατάξεων	417
5.4	Αποκωδικοποιητές	419
5.4.1	Δυαδικοί αποκωδικοποιητές	420
5.4.2	Λογικά σύμβολα για στοιχεία μεγαλύτερης κλίμακας	422
5.4.3	Ο διπλός αποκωδικοποιητής 2-προς-4 74x139	424
5.4.4	Ο αποκωδικοποιητής 3-προς-8 74x138	426
5.4.5	Αλυσιδωτοί δυαδικοί αποκωδικοποιητές	430
5.4.6	Αποκωδικοποιητές σε ABEL και προγραμματιζόμενες λογικές διατάξεις (PLD)	430
5.4.7	Αποκωδικοποιητές σε VHDL	437
5.4.8	Αποκωδικοποιητές επτά τμημάτων	443
5.5	Κωδικοποιητές	447
5.5.1	Κωδικοποιητές προτεραιότητας	447
5.5.2	Ο κωδικοποιητής προτεραιότητας 74x148	449
5.5.3	Κωδικοποιητές σε ABEL και προγραμματιζόμενες λογικές διατάξεις (PLD)	452
5.5.4	Κωδικοποιητές σε VHDL	455
5.6	Διατάξεις τριών καταστάσεων	456
5.6.1	Απομονωτές τριών καταστάσεων	457

5.6.2	Τυπικοί απομονωτές τριών καταστάσεων ολοκλήρωσης μικρής και μέσης κλίμακας (SSI/MSI)	459
5.6.3	Έξοδοι τριών καταστάσεων σε ABEL και προγραμματιζόμενες λογικές διατάξεις (PLD)	464
5.6.4	Έξοδοι τριών καταστάσεων σε VHDL	469
5.7	Πολυπλέκτες	471
5.7.1	Τυποποιημένοι πολυπλέκτες μέσης κλίμακας ολοκλήρωσης (MSI)	473
5.7.2	Επέκταση των πολυπλεκτών	476
5.7.3	Πολυπλέκτες, αποπολυπλέκτες και δίαυλοι	479
5.7.4	Πολυπλέκτες σε ABEL και προγραμματιζόμενες λογικές διατάξεις (PLD)	480
5.7.5	Πολυπλέκτες σε VHDL	485
5.8	Πύλες αποκλειστικού OR και κυκλώματα ισοτιμίας	486
5.8.1	Πύλες αποκλειστικού OR και αποκλειστικού NOR	486
5.8.2	Κυκλώματα ισοτιμίας	489
5.8.3	Γεννήτρια ισοτιμίας 9 bit 74x280	489
5.8.4	Εφαρμογές ελέγχου ισοτιμίας	489
5.8.5	Πύλες αποκλειστικού OR και κυκλώματα ισοτιμίας σε ABEL και (PLD)	492
5.8.6	Πύλες αποκλειστικού OR και κυκλώματα ισοτιμίας σε VHDL	493
5.9	Συγκριτές	495
5.9.1	Δομή συγκριτή	497
5.9.2	Επαναληπτικά κυκλώματα	497
5.9.3	Ένα επαναληπτικό κύκλωμα συγκριτή	498
5.9.4	Πρότυποι συγκριτές MSI	499
5.9.5	Συγκριτές σε ABEL και προγραμματιζόμενες λογικές διατάξεις (PLD)	503
5.9.6	Συγκριτές σε VHDL	504
5.10	Αθροιστές, αφαιρέτες και αριθμητικές και λογικές μονάδες (ΑΛΜ)	508
5.10.1	Ημιαθροιστές και πλήρεις αθροιστές	508
5.10.2	Αθροιστές κυμάτωσης	509
5.10.3	Αφαιρέτες	510
5.10.4	Αθροιστές πρόβλεψης κρατουμένου	512
5.10.5	Αθροιστές μέσης κλίμακας ολοκλήρωσης (MSI)	514
5.10.6	Αριθμητικές και λογικές μονάδες μέσης κλίμακας ολοκλήρωσης (MSI)	517
5.10.7	Πρόβλεψη κρατουμένου ομάδας	519
5.10.8	Αθροιστές σε ABEL και προγραμματιζόμενες λογικές διατάξεις (PLD)	521
5.10.9	Αθροιστές σε VHDL	523
5.11	Συνδυαστικοί πολλαπλασιαστές	525
5.11.1	Δομές συνδυαστικών πολλαπλασιαστών	525
5.11.2	Πολλαπλασιασμός σε ABEL και προγραμματιζόμενες λογικές διατάξεις (PLD)	528
5.11.3	Πολλαπλασιασμός σε VHDL	529
	Παραπομπές	535
	Προβλήματα για εξάσκηση	536
	Ασκήσεις	539
6	Παραδείγματα σχεδίασης συνδυαστικών κυκλωμάτων	551
6.1	Παραδείγματα σχεδίασης με δομικά στοιχεία	552
6.1.1	Ο κυκλικός ολισθητής	552
6.1.2	Απλός κωδικοποιητής κινητής υποδιαστολής	556
6.1.3	Κωδικοποιητής διπλής προτεραιότητας	559
6.1.4	Επάλληλοι συγκριτές	562

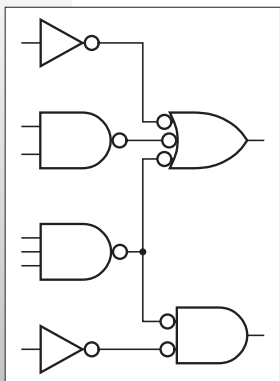
6.1.5	Συγκριτής εξαρτώμενος από την κατάσταση λειτουργίας	562
6.2	Παραδείγματα σχεδίασης με χρήση ABEL και PLD	565
6.2.1	Κυκλικός ολισθητής	565
6.2.2	Απλός κωδικοποιητής κινητής υποδιαστολής	568
6.2.3	Κωδικοποιητής διπλής προτεραιότητας	571
6.2.4	Επάλληλοι συγκριτές	573
6.2.5	Συγκριτής εξαρτώμενος από την τρόπο λειτουργίας	576
6.2.6	Απαριθμητής μονάδων	579
6.2.7	Τρίλιζα	581
6.3	Παραδείγματα σχεδίασης με χρήση VHDL	590
6.3.1	Κυκλικός ολισθητής	590
6.3.2	Απλός κωδικοποιητής κινητής υποδιαστολής	600
6.3.3	Κωδικοποιητής διπλής προτεραιότητας	604
6.3.4	Επάλληλοι συγκριτές	606
6.3.5	Συγκριτής εξαρτώμενος από τον τρόπο λειτουργίας	609
6.3.6	Απαριθμητής μονάδων	610
6.3.7	Τρίλιζα	614
	Ασκήσεις	622
7	Αρχές ακολουθιακής λογικής σχεδίασης	625
7.1	Δισταθή στοιχεία	628
7.1.1	Ψηφιακή ανάλυση	628
7.1.2	Αναλογική ανάλυση	629
7.1.3	Μετασταθής συμπεριφορά	630
7.2	Κυκλώματα μανδάλωσης και φλιπ-φλοπ	632
7.2.1	Κύκλωμα μανδάλωσης $S-R$	633
7.2.2	Κύκλωμα μανδάλωσης $\overline{S}-\overline{R}$	635
7.2.3	Κύκλωμα μανδάλωσης $S-R$ με έγκριση	636
7.2.4	Κύκλωμα μανδάλωσης D	637
7.2.5	Φλιπ-φλοπ D ενεργοποιούμενο με ακμή	639
7.2.6	Ενεργοποιούμενο με ακμή φλιπ-φλοπ D με είσοδο έγκρισης	642
7.2.7	Φλιπ-φλοπ σάρωσης	643
7.2.8	Φλιπ-φλοπ $S-R$ κυρίου-υπηρέτη	645
7.2.9	Φλιπ-φλοπ $J-K$ κυρίου-υπηρέτη	647
7.2.10	Φλιπ-φλοπ $J-K$ ενεργοποιούμενο με ακμή	648
7.2.11	Φλιπ-φλοπ T	650
7.3	Ανάλυση χρονισμένης σύγχρονης μηχανής καταστάσεων	652
7.3.1	Δομή της μηχανής καταστάσεων	652
7.3.2	Λογική εξόδου	653
7.3.3	Χαρακτηριστικές εξισώσεις	655
7.3.4	Ανάλυση μηχανών καταστάσεων με κυκλώματα φλιπ-φλοπ D	656
7.3.5	Ανάλυση μηχανών καταστάσεων με κυκλώματα φλιπ-φλοπ $J-K$	665
7.4	Σχεδίαση χρονισμένης σύγχρονης μηχανής καταστάσεων	667
7.4.1	Παράδειγμα σχεδίασης πίνακα καταστάσεων	669
7.4.2	Ελαχιστοποίηση καταστάσεων	674
7.4.3	Αντιστοίχιση καταστάσεων	675
7.4.4	Σύνθεση με χρήση φλιπ-φλοπ D	680
7.4.5	Σύνθεση με τη χρήση κυκλωμάτων φλιπ-φλοπ $J-K$	684
7.4.6	Περισσότερα παραδείγματα σχεδίασης με τη χρήση κυκλωμάτων φλιπ-φλοπ D	689

7.5	Σχεδίαση μηχανών καταστάσεων με χρήση διαγραμμάτων καταστάσεων	694
7.6	Σύνθεση μηχανών καταστάσεων με λίστες μετάβασης	702
7.6.1	Εξισώσεις μετάβασης	703
7.6.2	Εξισώσεις διέγερσης	704
7.6.3	Παραλλαγές πάνω στο θέμα	705
7.6.4	Υλοποίηση της μηχανής καταστάσεων	706
7.7	Ένα ακόμα παράδειγμα σχεδίασης μηχανής καταστάσεων	706
7.7.1	Το παιχνίδι της “πρόβλεψης”	706
7.7.2	Αχρησιμοποίητες καταστάσεις	710
7.7.3	Αντιστοίχιση καταστάσεων κωδικοποιημένη ως προς την έξοδο	711
7.7.4	Κωδικοποίηση “αδιάφορων” καταστάσεων	712
7.8	Αποσύνθεση μηχανών καταστάσεων	714
7.9	Ακολουθιακά κυκλώματα ανάδρασης	717
7.9.1	Ανάλυση	718
7.9.2	Ανάλυση κυκλωμάτων με περισσότερους από έναν βρόχους ανάδρασης	723
7.9.3	Συναγωνισμοί	726
7.9.4	Πίνακες καταστάσεων και πίνακες ροής	727
7.9.5	Ανάλυση κυκλωμάτων CMOS φλιπ-φλοπ D	730
7.10	Σχεδίαση ακολουθιακού κυκλώματος ανάδρασης	731
7.10.1	Κυκλώματα μανδάλωσης	732
7.10.2	Σχεδίαση πίνακα ροής θεμελιώδους τρόπου λειτουργίας	734
7.10.3	Ελαχιστοποίηση του πίνακα ροής	736
7.10.4	Αντιστοίχιση καταστάσεων χωρίς συναγωνισμούς	737
7.10.5	Εξισώσεις διέγερσης	741
7.10.6	Ουσιαστικοί κίνδυνοι	742
7.10.7	Ανακεφαλαίωση	746
7.11	Λειτουργίες σχεδίασης ακολουθιακών κυκλωμάτων της ABEL	747
7.11.1	Καταχωρισμένες έξοδοι	747
7.11.2	Διαγράμματα καταστάσεων	749
7.11.3	Εξωτερική μνήμη κατάστασης	755
7.11.4	Προσδιορισμός εξόδων Moore	757
7.11.5	Προσδιορισμός εξόδων Mealy και διοχετευμένων εξόδων με την εντολή WITH.	757
7.11.6	Διανύσματα ελέγχου	760
7.12	Λειτουργίες σχεδίασης ακολουθιακών κυκλωμάτων της VHDL	764
7.12.1	Ακολουθιακά κυκλώματα ανάδρασης	764
7.12.2	Χρονισμένα κυκλώματα	766
	Παραπομπές	768
	Προβλήματα για εξάσκηση	770
	Ασκήσεις	774
8	Πρακτικές ακολουθιακής λογικής σχεδίασης	785
8.1	Πρότυπα τεκμηρίωσης ακολουθιακών κυκλωμάτων	786
8.1.1	Γενικές απαιτήσεις	786
8.1.2	Λογικά σύμβολα	787
8.1.3	Τρόποι περιγραφής μηχανών καταστάσεων	788
8.1.4	Διαγράμματα χρονισμού και προδιαγραφές χρονισμού	789
8.2	Κυκλώματα μανδάλωσης και φλιπ-φλοπ	794
8.2.1	Κυκλώματα μανδάλωσης και φλιπ-φλοπ SSI	794

8.2.2	Αποκλυνδωνισμός διακοπών	795
8.2.3	Ο απλούστερος αποκλυνδωνιστής διακοπών	796
8.2.4	Κυκλώμα δέσμευσης διαύλου	798
8.2.5	Καταχωρητές και κυκλώματα μανδάλωσης πολλών bit	800
8.2.6	Καταχωρητές και κυκλώματα μανδάλωσης στην ABEL και τις προγραμματιζόμενες λογικές διατάξεις	804
8.2.7	Καταχωρητές και κυκλώματα μανδάλωσης στη VHDL	809
8.3	Ακολουθιακές προγραμματιζόμενες λογικές διατάξεις	814
8.3.1	Διπολικές ακολουθιακές προγραμματιζόμενες λογικές διατάξεις	814
8.3.2	Ακολουθιακές διατάξεις GAL	818
8.3.3	Προδιαγραφές χρονισμού προγραμματιζόμενων λογικών διατάξεων	824
8.4	Μετρητές	827
8.4.1	Μετρητές κύματωσης	828
8.4.2	Σύγχρονοι μετρητές	828
8.4.3	Μετρητές και εφαρμογές MSI	830
8.4.4	Αποκωδικοποίηση καταστάσεων δυαδικού μετρητή	838
8.4.5	Μετρητές στην ABEL και σε προγραμματιζόμενες λογικές διατάξεις	841
8.4.6	Μετρητές στη VHDL	844
8.5	Καταχωρητές ολίσθησης	849
8.5.1	Δομή καταχωρητών ολίσθησης	849
8.5.2	Καταχωρητές ολίσθησης MSI	851
8.5.3	Η μεγαλύτερη εφαρμογή καταχωρητών ολίσθησης του κόσμου	856
8.5.4	Μετατροπή σειριακού σε παράλληλο	857
8.5.5	Μετρητές με καταχωρητές ολίσθησης	863
8.5.6	Δακτυλιοειδείς μετρητές	864
8.5.7	Μετρητές Johnson	867
8.5.8	Μετρητές με καταχωρητές ολίσθησης γραμμικής ανάδρασης	869
8.5.9	Καταχωρητές ολίσθησης στην ABEL και στις προγραμματιζόμενες λογικές διατάξεις	873
8.5.10	Καταχωρητές ολίσθησης στη VHDL	884
8.6	Επαναληπτικά και ακολουθιακά κυκλώματα	889
8.7	Μεθοδολογία σύγχρονης σχεδίασης	892
8.7.1	Δομή σύγχρονου συστήματος	893
8.7.2	Παράδειγμα σχεδίασης σύγχρονου συστήματος	896
8.8	Εμπόδια στη σύγχρονη σχεδίαση	901
8.8.1	Απόκλιση ρολογιού	901
8.8.2	Οδήγηση του ρολογιού μέσω πύλης	905
8.8.3	Ασύγχρονες είσοδοι	907
8.9	Αστοχία του συγχρονιστή και μεταστάθεια	910
8.9.1	Αστοχία του συγχρονιστή	911
8.9.2	Χρόνος ανάλυσης μεταστάθειας	912
8.9.3	Σχεδίαση αξιόπιστου συγχρονιστή	912
8.9.4	Ανάλυση μετασταθούς χρονισμού	913
8.9.5	Καλύτεροι συγχρονιστές	916
8.9.6	Άλλες σχεδιάσεις συγχρονιστών	919
8.9.7	Κυκλώματα φλιπ-φλοπ με αντοχή μεταστάθειας	921
8.9.8	Συγχρονισμός μεταφορών δεδομένων υψηλής ταχύτητας	922
	Παραπομπές	935
	Προβλήματα για εξάσκηση	939

Ασκήσεις	941
9 Παραδείγματα σχεδίασης ακολουθιακών κυκλωμάτων	951
9.1 Παραδείγματα σχεδίασης με ABEL και PLD	952
9.1.1 Χρονισμός και συσκευασία μηχανών καταστάσεων με βάση προγραμματιζόμενες λογικές διατάξεις	953
9.1.2 Μερικές απλές μηχανές	956
9.1.3 Τα πίσω φώτα του Φορντ Thunderbird	960
9.1.4 Το παιχνίδι πρόβλεψης	962
9.1.5 Ας ξαναεφεύρουμε τους ελεγκτές των φωτεινών σηματοδοτών	967
9.2 Παραδείγματα σχεδίασης με χρήση της VHDL	972
9.2.1 Μερικές απλές μηχανές	972
9.2.2 Τα πίσω φώτα ενός Φορντ Thunderbird	982
9.2.3 Το παιχνίδι πρόβλεψης	983
9.2.4 Ας ξαναεφεύρουμε τους ελεγκτές των φωτεινών σηματοδοτών	986
Ασκήσεις	990
10 Μνήμες, διατάξεις CPLD, και FPGA	993
10.1 Μνήμη μόνο για ανάγνωση	994
10.1.1 Χρήση μνήμης ROM για “τυχαίες” συνδυαστικές λογικές συναρτήσεις	995
10.1.2 Εσωτερική δομή της μνήμης ROM	999
10.1.3 Διδιάστατη αποκωδικοποίηση	1002
10.1.4 Τύποι μνήμης ROM που κυκλοφορούν στο εμπόριο	1005
10.1.5 Είσοδοι ελέγχου και χρονισμός της μνήμης ROM	1009
10.1.6 Εφαρμογές για τις μνήμες ROM	1013
10.2 Μνήμη ανάγνωσης-εγγραφής	1020
10.3 Στατική RAM	1021
10.3.1 Είσοδοι και έξοδοι στατικής μνήμης RAM	1021
10.3.2 Εσωτερική δομή μιας μνήμης SRAM	1023
10.3.3 Χρονισμός μιας στατικής μνήμης RAM	1025
10.3.4 Τυπικές στατικές μνήμες RAM	1028
10.3.5 Σύγχρονη μνήμη SRAM	1030
10.4 Δυναμική RAM	1036
10.4.1 Δομή δυναμικής RAM	1036
10.4.2 Χρονισμός δυναμικής μνήμης RAM	1039
10.4.3 Σύγχρονες μνήμες DRAM	1042
10.5 Σύνθετες προγραμματιζόμενες λογικές διατάξεις	1043
10.5.1 Η οικογένεια διατάξεων CPLD XC9500 της Xilinx	1045
10.5.2 Αρχιτεκτονική των λειτουργικών δομικών μονάδων	1048
10.5.3 Αρχιτεκτονική των δομικών μονάδων εισόδου/εξόδου	1051
10.5.4 Μήτρα μεταγωγής	1053
10.6 Προγραμματιζόμενοι από το χρήστη πίνακες πυλών	1057
10.6.1 Η οικογένεια FPGA XC4000 της Xilinx	1057
10.6.2 Διευθετήσιμη λογική δομική μονάδα	1059
10.6.3 Δομική μονάδα εισόδου/εξόδου	1062
10.6.4 Προγραμματιζόμενη διασύνδεση	1064
Παραπομπές	1067
Προβλήματα για εξάσκηση	1069
Ασκήσεις	1069

11	Πρόσθετα θέματα από τον πραγματικό κόσμο	1073
11.1	Εργαλεία σχεδίασης με τη βοήθεια υπολογιστή	1073
11.1.1	<i>Γλώσσες περιγραφής υλικού</i>	1074
11.1.2	<i>Σχηματική αποτύπωση</i>	1075
11.1.3	<i>Σχέδια και προδιαγραφές χρονισμού</i>	1078
11.1.4	<i>Ανάλυση και προσομοίωση κυκλώματος</i>	1078
11.1.5	<i>Διάταξη πλακέτας τυπωμένου κυκλώματος</i>	1081
11.2	Σχεδίαση με στόχο την ελεγχιμότητα	1083
11.2.1	<i>Έλεγχοι</i>	1084
11.2.2	<i>Έλεγχος σε κλίνη ακίδων και εντός του κυκλώματος</i>	1085
11.2.3	<i>Μέθοδοι σάρωσης</i>	1089
11.3	Εκτίμηση της αξιοπιστίας ενός ψηφιακού συστήματος	1091
11.3.1	<i>Συχνότητα αστοχιών</i>	1092
11.3.2	<i>Αξιοπιστία και μέσος χρόνος μεταξύ αστοχιών</i>	1095
11.3.3	<i>Αξιοπιστία συστήματος</i>	1095
11.4	Γραμμές μετάδοσης, ανακλάσεις, και τερματισμός	1097
11.4.1	<i>Βασική θεωρία των γραμμών μετάδοσης</i>	1097
11.4.2	<i>Διασυνδέσεις λογικών σημάτων ως γραμμές μετάδοσης</i>	1100
11.4.3	<i>Τερματισμοί λογικών σημάτων</i>	1103
	Παραπομπές	1106
	ΕΥΡΕΤΗΡΙΟ	1109



4 κεφάλαιο

Αρχές συνδυαστικής λογικής σχεδίασης

Τα λογικά κυκλώματα κατατάσσονται σε δύο τύπους, τα “συνδυαστικά” και τα “ακολουθιακά”. Συνδυαστικό λογικό κύκλωμα είναι αυτό του οποίου οι έξοδοι εξαρτώνται μόνο από τις τρέχουσες εισόδους. Το περιστροφικό κουμπί επιλογής καναλιών μιας παλιάς τηλεόρασης λειτουργεί όπως ένα συνδυαστικό κύκλωμα: η “έξοδος” του επιλέγει ένα κανάλι με βάση την τρέχουσα θέση του κουμπιού (“είσοδος”).

Οι έξοδοι ενός ακολουθιακού λογικού κυκλώματος εξαρτώνται όχι μόνο από τις τρέχουσες εισόδους, αλλά και από την προηγούμενη ακολουθία εισόδων, η οποία μπορεί να φτάνει σε οποιοδήποτε βάθος χρόνου. Ο επιλογέας καναλιών που ελέγχεται από κουμπιά επιλογής πάνω και κάτω σε μια τηλεόραση ή ένα βίντεο είναι ένα ακολουθιακό κύκλωμα — η επιλογή των καναλιών εξαρτάται από την προηγούμενη ακολουθία πατημάτων των κουμπιών πάνω και κάτω, τουλάχιστον από τη στιγμή που αρχίσατε να βλέπετε 10 ώρες πριν, και πιθανώς από τότε που θέσατε σε λειτουργία τη συσκευή. Τα ακολουθιακά κυκλώματα περιγράφονται στα Κεφάλαια 7 και 8.

Ένα συνδυαστικό κύκλωμα είναι δυνατόν να περιέχει οποιονδήποτε αριθμό λογικών πυλών και αντιστροφών, αλλά όχι βρόχους ανάδρασης. Ο βρόχος ανάδρασης είναι μια διαδρομή σήματος σε ένα κύκλωμα η οποία επιτρέπει στην έξοδο μιας πύλης να μεταδίδεται πίσω στην είσοδο της ίδιας πύλης. Ένας τέτοιος βρόχος δημιουργεί ακολουθιακή συμπεριφορά στο κύκλωμα.

Στην *ανάλυση* των συνδυαστικών κυκλωμάτων ξεκινάμε με ένα λογικό διάγραμμα και προχωράμε σε μια τυπική περιγραφή της λειτουργίας που εκτελείται από αυτό το κύκλωμα, όπως π.χ. με έναν πίνακα αληθείας ή με μια λογική παράσταση. Στη *σύνθεση* κάνουμε το αντίστροφο, ξεκινώντας με την τυπική περιγραφή και προχωρώντας σε ένα λογικό διάγραμμα.

**ΣΥΝΘΕΣΗ ΚΑΙ
ΣΧΕΔΙΑΣΗ**

Η σχεδίαση λογικών κυκλωμάτων είναι ένα υπερσύνολο της σύνθεσης, εφόσον σε ένα πραγματικό πρόβλημα σχεδίασης ξεκινάμε συνήθως με μια άτυπη (λεκτική ή νοητή) περιγραφή του κυκλώματος. Συχνά το πιο δημιουργικό και δύσκολο μέρος της σχεδίασης είναι η διατύπωση της περιγραφής του κυκλώματος, με τον καθορισμό των σημάτων εισόδου και εξόδου του κυκλώματος και τον καθορισμό της λειτουργικής του συμπεριφοράς με τη βοήθεια πινάκων αληθείας και εξισώσεων. Αφού δημιουργήσουμε την τυπική περιγραφή του κυκλώματος, συνήθως συνεχίζουμε με μια διαδικασία σύνθεσης τύπου “περιστροφή της μανιβέλας” για να πάρουμε το λογικό διάγραμμα ενός κυκλώματος με την απαιτούμενη λειτουργική συμπεριφορά. Το υλικό που περιλαμβάνεται στις πρώτες τέσσερις ενότητες του κεφαλαίου αυτού αποτελεί τη βάση των διαδικασιών τύπου “περιστροφή της μανιβέλας”, όπου η “μανιβέλα” γυρίζει με το χέρι ή με τον υπολογιστή. Οι δύο τελευταίες ενότητες περιγράφουν πραγματικές γλώσσες σχεδίασης, την ABEL και τη VHDL. Όταν δημιουργούμε μια σχεδίαση με μια από τις γλώσσες αυτές, το πρόγραμμα στον υπολογιστή μπορεί να εκτελεί τα βήματα της σύνθεσης για λογαριασμό μας. Στα επόμενα κεφάλαια θα συναντήσουμε πολλά παραδείγματα της πραγματικής διαδικασίας σχεδίασης.

Τα συνδυαστικά κυκλώματα είναι δυνατόν να έχουν μία ή περισσότερες εισόδους. Οι περισσότερες τεχνικές ανάλυσης και σύνθεσης είναι δυνατόν να επεκταθούν με προφανή τρόπο από κυκλώματα μίας εξόδου σε κυκλώματα πολλών εξόδων (π.χ. επαναλαμβάνοντας τα ίδια βήματα για κάθε έξοδο). Επίσης υποδεικνύουμε τον τρόπο επέκτασης μερικών τεχνικών με όχι και τόσο προφανή τρόπο για βελτιωμένη αποτελεσματικότητα στην περίπτωση των πολλών εξόδων.

Ο σκοπός αυτού του κεφαλαίου είναι να σας δώσει γερές θεωρητικές βάσεις όσον αφορά την ανάλυση και τη σύνθεση συνδυαστικών λογικών κυκλωμάτων, οι οποίες είναι σημαντικές για δύο λόγους, όπως θα δούμε όταν φτάσουμε στα ακολουθιακά κυκλώματα. Αν και οι περισσότερες από τις διαδικασίες ανάλυσης και σύνθεσης που αναφέρονται σε αυτό το κεφάλαιο είναι σήμερα αυτοματοποιημένες χάρη στη χρήση εργαλείων σχεδίασης με τη βοήθεια υπολογιστή, χρειάζεστε μια βασική κατανόηση των θεμελιωδών αρχών χρήσης των εργαλείων αυτών ώστε να εντοπίζετε το πρόβλημα όταν παίρνετε απρόσμενα ή ανεπιθύμητα αποτελέσματα.

Όταν θα έχετε πλέον κατανοήσει τις θεμελιώδεις αρχές, το επόμενο βήμα είναι να κατανοήσετε με ποιον τρόπο μπορούν να εκφραστούν και να αναλυθούν οι συνδυαστικές λειτουργίες με τη χρήση γλωσσών περιγραφής υλικού (HDL). Έτσι, οι δύο τελευταίες ενότητες του κεφαλαίου αυτού περιγράφουν τα βασικά χαρακτηριστικά των γλωσσών ABEL και VHDL, τις οποίες θα χρησιμοποιήσουμε για τη σχεδίαση διαφόρων λογικών κυκλωμάτων σε όλο το βιβλίο.

Προτού ξεκινήσουμε τη συζήτηση για τα συνδυαστικά λογικά κυκλώματα, πρέπει να κάνουμε μια εισαγωγή στην άλγεβρα μεταγωγής, το βα-

σικό μαθηματικό εργαλείο ανάλυσης και σύνθεσης κάθε τύπου λογικών κυκλωμάτων.

4.1 Άλγεβρα μεταγωγής

Οι τυπικές τεχνικές ανάλυσης ψηφιακών κυκλωμάτων έχουν τις ρίζες τους στην εργασία ενός Άγγλου μαθηματικού, του George Boole. Το 1854 ο Boole επινόησε ένα αλγεβρικό σύστημα δύο τιμών, που σήμερα λέγεται *άλγεβρα Boole*, με σκοπό να “εκφράσει τους θεμελιώδεις νόμους του συλλογισμού στη συμβολική γλώσσα ενός Λογισμού”. Με το σύστημα αυτό ένας φιλόσοφος, κάποιος που ασχολείται με την επιστήμη της λογικής, ή ένας μόνιμος κάτοικος άλλου πλανήτη μπορεί να διατυπώνει προτάσεις που να είναι αληθείς ή ψευδείς, να τις συνδυάζει μεταξύ τους για να φτιάχνει νέες προτάσεις, καθώς και να προσδιορίζει την αλήθεια ή το ψεύδος των νέων προτάσεων. Για παράδειγμα, αν συμφωνήσουμε ότι “Οι άνθρωποι που δεν έχουν μελετήσει αυτή την ύλη είτε είναι αποτυχημένοι είτε δεν είναι σπασίκλες” και ότι “Κανένας σχεδιαστής υπολογιστών δεν είναι αποτυχημένος”, μπορούμε να απαντήσουμε σε ερωτήσεις όπως π.χ. “Αν είσαι σπασίκλας σχεδιαστής υπολογιστών, τότε έχεις ήδη μελετήσει αυτό το αντικείμενο;”.

άλγεβρα Boole

Αρκετά μετά τον Boole, το 1938, ο ερευνητής Claude E. Shannon των Εργαστηρίων Bell παρουσίασε τον τρόπο προσαρμογής της *άλγεβρας Boole* για την ανάλυση και την περιγραφή της συμπεριφοράς κυκλωμάτων που κατασκευάζονται από ηλεκτρονόμεους (ρελέ), που ήταν τα πιο διαδεδομένα ψηφιακά λογικά στοιχεία εκείνης της εποχής. Στην *άλγεβρα μεταγωγής* του Shannon η κατάσταση της επαφής ενός ρελέ, δηλαδή ανοικτή ή κλειστή, αναπαρίσταται από μια μεταβλητή X η οποία είναι δυνατόν να έχει μια από δύο δυνατές τιμές, 0 ή 1. Στις σημερινές τεχνολογίες λογικής, αυτές οι τιμές αντιστοιχούν σε μεγάλη ποικιλία φυσικών συνθηκών, π.χ. υψηλή τάση (HIGH) ή χαμηλή τάση (LOW), αναμμένο ή σβηστό φως, πυκνωτής φορτισμένος ή εκφορτισμένος, ασφάλεια καμένη ή όχι κ.λπ., όπως είδαμε αναλυτικά στον Πίνακα 3-1, στην Ενότητα 3.1.

άλγεβρα μεταγωγής

Στο υπόλοιπο μέρος της ενότητας αυτής αναπτύσσουμε την *άλγεβρα μεταγωγής*, χρησιμοποιώντας τις “πρώτες βασικές αρχές” και τις γνώσεις που ήδη έχουμε σχετικά με τη συμπεριφορά των λογικών στοιχείων (πυλών και αντιστροφών). Για περισσότερες ιστορικές ή/και μαθηματικές αναφορές σε αυτό το θέμα, συμβουλευθείτε τις Παραπομπές.

4.1.1 Αξιώματα

Στην *άλγεβρα μεταγωγής* χρησιμοποιούμε μια συμβολική μεταβλητή, όπως π.χ. X , για να αναπαραστήσουμε την κατάσταση ενός λογικού σήματος. Ένα λογικό σήμα είναι σε μία από δύο δυνατές καταστάσεις, π.χ. χαμηλή ή υψηλή τάση, αναμμένο ή σβηστό φως κ.λπ., ανάλογα με την τεχνολογία. Λέμε ότι το X έχει τιμή “0” για τη μια από αυτές τις καταστάσεις και “1” για την άλλη.

σύμβαση θετικής
λογικής
σύμβαση αρνητικής
λογικής

Για παράδειγμα, αναφορικά με τα λογικά κυκλώματα CMOS και TTL του Κεφαλαίου 3, η *σύμβαση θετικής λογικής* μάς επιβάλλει να αντιστοιχίζουμε την τιμή “0” στην τάση LOW (χαμηλή) και την τιμή “1” στην τάση HIGH (υψηλή). Η *σύμβαση αρνητικής λογικής* κάνει την αντίθετη συσχέτιση: 0 = HIGH και 1 = LOW. Ωστόσο, η επιλογή της θετικής ή της αρνητικής λογικής δεν επηρεάζει τη δυνατότητά μας να αναπτύξουμε μια συνεπή αλγεβρική περιγραφή της συμπεριφοράς του κυκλώματος, παρά μόνο την αφαίρεση “από τη φυσική στην αλγεβρική μορφή”, όπως θα εξηγήσουμε αργότερα στην ανάλυση της έννοιας της *δουκότητας*. Για την ώρα, μπορούμε να αγνοήσουμε τη φυσική πραγματικότητα των λογικών κυκλωμάτων και να προσποιηθούμε ότι λειτουργούν άμεσα με τα λογικά σύμβολα 0 και 1.

αξίωμα

Τα *αξιώματα* ενός μαθηματικού συστήματος είναι ένα ελάχιστο σύνολο βασικών ορισμών που θεωρούμε ότι είναι αληθείς, από τους οποίους μπορούν να παραχθούν όλες οι υπόλοιπες πληροφορίες του συστήματος. Τα δύο πρώτα αξιώματα της *άλγεβρας μεταγωγής* ενσωματώνουν την “ψηφιακή αφαίρεση”, με την τυπική διατύπωση ότι η μεταβλητή X μπορεί να πάρει μόνο μία από δύο δυνατές τιμές:

$$(A1) \quad X=0 \quad \text{αν} \quad X \neq 1 \quad (A1') \quad X=1 \quad \text{αν} \quad X \neq 0$$

Σημειώστε ότι διατυπώσαμε τα αξιώματα ως ζεύγος, όπου η μόνη διαφορά ανάμεσα στο $A1$ και το $A1'$ είναι η εναλλαγή των συμβόλων 0 και 1. Αυτό είναι ένα χαρακτηριστικό όλων των αξιωμάτων της *άλγεβρας μεταγωγής* και αποτελεί τη βάση της αρχής της “*δουκότητας*” που θα μελετήσουμε αργότερα.

συμπληρωματικό
τόνος (')

Στην Ενότητα 3.3.3 δείξαμε τη σχεδίαση ενός αντιστροφέα, ο οποίος είναι ένα λογικό κύκλωμα του οποίου η στάθμη του σήματος εξόδου είναι το αντίθετο (ή το *συμπληρωματικό*) της στάθμης του σήματος εισόδου. Ο *τόνος* (') υποδηλώνει τη λειτουργία αντιστροφέα. Αυτό σημαίνει ότι αν η μεταβλητή X δείχνει ένα σήμα στην είσοδο ενός αντιστροφέα, τότε το X' δείχνει την τιμή ενός σήματος στην έξοδο του αντιστροφέα. Αυτός ο συμβολισμός περιγράφεται τυπικά στο δεύτερο ζευγάρι αξιωμάτων:

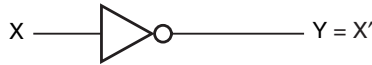
$$(A2) \quad \text{αν} \quad X = 0, \text{ τότε} \quad X' = 1 \quad (A2') \quad \text{αν} \quad X = 1, \text{ τότε} \quad X' = 0$$

αλγεβρικός τελεστής
παράσταση
πράξη NOT

Όπως φαίνεται στην Εικόνα 4-1, η έξοδος ενός αντιστροφέα με σήμα εισόδου X μπορεί να έχει ένα οποιοδήποτε όνομα σήματος, π.χ. Y . Αλγεβρικά, ωστόσο, γράφουμε $Y=X'$ για να πούμε ότι “το σήμα Y έχει πάντα τιμή αντίθετη από εκείνη του σήματος X ”. Ο *τόνος* (') είναι ένας *αλγεβρικός τελεστής*, ενώ το X' είναι μια *παράσταση* την οποία μπορείτε να διαβάσετε ως “ X τόνος” ή ως “NOT X ”. Η χρήση αυτή είναι ανάλογη με εκείνη που έχετε μάθει στις γλώσσες προγραμματισμού, όπου αν το J είναι μια ακέραια μεταβλητή, τότε το $\neg J$ είναι μια παράσταση της οποίας η τιμή είναι 0 - J . Παρότι αυτά μπορεί να φαίνονται ασήμαντα, θα μάθουμε ότι η διάκριση ανάμεσα στα ονόματα των σημάτων (X , Y), τις παραστάσεις (X') και τις εξισώσεις ($Y=X'$) έχει μεγάλη σημασία κατά τη με-

Εικόνα 4-1

Ονοματολογία και αλγεβρική σημειογραφία των σημάτων ενός αντιστροφέα.



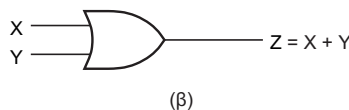
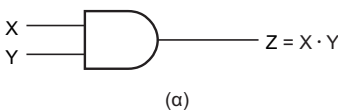
λέτη προτύπων τεκμηρίωσης και εργαλείων προγραμματισμού για τη σχεδίαση λογικών κυκλωμάτων.

Στην Ενότητα 3.3.6 είδαμε πώς κατασκευάζεται μια πύλη AND δύο εισόδων τύπου CMOS, ένα κύκλωμα του οποίου οι εξοδοι είναι 1 αν και οι δύο εισόδους είναι 1. Η λειτουργία μιας πύλης AND δύο εισόδων λέγεται μερικές φορές *λογικός πολλαπλασιασμός* και συμβολίζεται αλγεβρικά με μια *τελεία πολλαπλασιασμού* ($\cdot$). Αυτό σημαίνει ότι μια πύλη AND με εισόδους X και Y έχει ένα σήμα εξόδου του οποίου η τιμή είναι $X \cdot Y$, όπως φαίνεται στην Εικόνα 4-2(α). Μερικοί συγγραφείς, ειδικά μαθηματικοί και επιστήμονες της λογικής, εκφράζουν το λογικό πολλαπλασιασμό με ένα σύμβολο γωνίας ($X \wedge Y$). Εμείς τηρούμε την τυπική τεχνική πρακτική χρησιμοποιώντας την τελεία ($X \cdot Y$). Κατά τη μελέτη γλώσσών περιγραφής υλικού (HDL) συναντάμε κάποια άλλα σύμβολα τα οποία χρησιμοποιούνται για να δείξουν το ίδιο πράγμα.

λογικός
πολλαπλασιασμός
τελεία
πολλαπλασιασμού ($\cdot$)

Στην Ενότητα 3.3.6 περιγράψαμε επίσης πώς κατασκευάζεται μια πύλη OR δύο εισόδων τύπου CMOS, ένα κύκλωμα του οποίου η έξοδος είναι 1 αν οποιαδήποτε από τις εισόδους είναι 1. Η λειτουργία μιας πύλης OR δύο εισόδων μερικές φορές λέγεται *λογική πρόσθεση* και συμβολίζεται αλγεβρικά με το σύμβολο συν ($+$). Μια πύλη OR με εισόδους X και Y δίνει σήμα εξόδου με τιμή $X + Y$, όπως φαίνεται στην Εικόνα 4-2(β).

λογική πρόσθεση

**Εικόνα 4-2**

Ονοματολογία και αλγεβρική σημειογραφία σημάτων: (α) πύλη AND, (β) πύλη OR.

ΣΗΜΕΙΩΣΗ ΓΙΑ ΤΗ ΣΗΜΕΙΟΓΡΑΦΙΑ

Μερικοί συγγραφείς χρησιμοποιούν επίσης τις σημειογραφίες X , $\sim X$ και $\neg X$ για το συμπλήρωμα του X . Η σημειογραφία με τη γραμμή από πάνω ($\bar{X}$) είναι μάλλον η ευρύτερα χρησιμοποιούμενη και η πιο εμφανίσιμη από τυπογραφική άποψη. Ωστόσο, εμείς χρησιμοποιούμε τη βασική σημειογραφία για να συνηθίσετε στη γραφή λογικών παραστάσεων σε μια γραμμή κειμένου, χωρίς την πιο γραφική γραμμή από πάνω, και για να αναγκαστείτε να χρησιμοποιείτε παρενθέσεις για τις σύνθετες συμπληρωματικές επιμέρους παραστάσεις, καθώς αυτό είναι που πρέπει να κάνετε όταν χρησιμοποιείτε γλώσσες HDL και άλλα εργαλεία.

Μερικοί συγγραφείς εκφράζουν τη λογική πρόσθεση με το σύμβολο “V” ($X \vee Y$), αλλά εμείς ακολουθούμε την τυπική τεχνική πρακτική της χρήσης του συμβόλου συν ($X+Y$). Τονίζεται για άλλη μια φορά ότι μπορούν να χρησιμοποιηθούν και άλλα σύμβολα στις γλώσσες HDL. Συμβατικά, θεωρούμε ότι στις λογικές παραστάσεις που περιλαμβάνουν πολλαπλασιασμό και πρόσθεση ο πολλαπλασιασμός έχει *προτεραιότητα*, όπως οι ακέριαι παραστάσεις στις συμβατικές γλώσσες προγραμματισμού. Αυτό σημαίνει ότι η έκφραση $W \cdot X + Y \cdot Z$ είναι ισοδύναμη με την $(W \cdot X) + (Y \cdot Z)$.

Τα τρία τελευταία ζεύγη αξιωμάτων διατυπώνουν τους τυπικούς ορισμούς των λειτουργιών AND και OR με την αναγραφή της εξόδου που παράγεται από κάθε πύλη για κάθε δυνατό συνδυασμό εισόδων:

(A3)	$0 \cdot 0 = 0$	(A3')	$1 + 1 = 1$
(A4)	$1 \cdot 1 = 1$	(A4')	$0 + 0 = 0$
(A5)	$0 \cdot 1 = 1 \cdot 0 = 0$	(A5')	$1 + 0 = 0 + 1 = 1$

Τα πέντε ζεύγη αξιωμάτων A1-A5 και A1'-A5' ορίζουν πλήρως την άλγεβρα μεταγωγής. Όλα τα υπόλοιπα συμβάντα του συστήματος αποδεικνύονται με τη χρήση των αξιωμάτων αυτών ως σημείων αρχής.

ΓΙΑ ΣΤΑΘΕΙΤΕ ΕΝΑ ΛΕΠΤΟ...

Στα παλαιότερα κείμενα χρησιμοποιείται η απλή *παράθεση* (XY) για να παρασταθεί ο λογικός πολλαπλασιασμός, εμείς όμως δεν τη χρησιμοποιούμε. Γενικά, η παράθεση είναι σαφής σημειογραφία μόνο όταν τα ονόματα των σημάτων περιορίζονται στον ένα χαρακτήρα. Διαφορετικά, τίθεται το ερώτημα: το XY είναι ένα λογικό γινόμενο ή είναι ένα όνομα σήματος με δύο χαρακτήρες; Τα ονόματα μεταβλητών με ένα χαρακτήρα είναι συνηθισμένα στην άλγεβρα, αλλά στα πραγματικά προβλήματα σχεδίασης προτιμάται η χρήση ονομάτων σημάτων με πολλούς χαρακτήρες οι οποίοι έχουν κάποιο νόημα. Έτσι, χρειαζόμαστε ένα διαχωριστικό ανάμεσα στα ονόματα, το οποίο μπορεί να είναι απλώς μια τελεία πολλαπλασιασμού αντί για ένα διάστημα. Το ισοδύναμο της HDL για την τελεία πολλαπλασιασμού (συνήθως * ή &) είναι απολύτως απαραίτητο όταν γράφονται λογικοί τύποι σε μια γλώσσα περιγραφής υλικού.

4.1.2 Θεωρήματα μίας μεταβλητής

Κατά τη διάρκεια της ανάλυσης ή της σύνθεσης λογικών κυκλωμάτων, συχνά γράφουμε αλγεβρικές παραστάσεις οι οποίες χαρακτηρίζουν την πραγματική ή την επιθυμητή συμπεριφορά. Τα *θεωρήματα* της άλγεβρας μεταγωγής είναι προτάσεις που γνωρίζουμε ότι είναι πάντα αληθείς και μας επιτρέπουν να χειριζόμαστε αλγεβρικές παραστάσεις οι οποίες επιτρέπουν την απλούστερη ανάλυση ή την αποτελεσματικότερη σύνθεση των αντίστοιχων κυκλωμάτων. Για παράδειγμα, το θεώρημα $X+0=X$ μας επιτρέπει να αντικαθιστούμε κάθε εμφάνιση του $X+0$ σε μια παράσταση με το X .

Πίνακας 4-1

Θεωρήματα
άλγεβρας
μεταγωγής με μία
μεταβλητή

(T1)	$X + 0 = X$	(T1')	$X \cdot 1 = X$	(ταυτότητες)
(T2)	$X + 1 = 1$	(T2')	$X \cdot 0 = 0$	(ουδέτερα στοιχεία)
(T3)	$X + X = X$	(T3')	$X \cdot X = X$	(αυτοδυναμία)
(T4)	$(X')' = X$			(ενέλιξη)
(T5)	$X + X' = 1$	(T5')	$X \cdot X' = 0$	(συμπληρώματα)

Ο Πίνακας 4-1 παρουσιάζει τα θεωρήματα της άλγεβρας μεταγωγής που περιλαμβάνουν μία μεταβλητή X . Πώς ξέρουμε ότι αυτά τα θεωρήματα είναι αληθή; Μπορούμε είτε να τα αποδείξουμε μόνοι μας είτε να πάρουμε την απόδειξη από κάποιον που την έχει κάνει ήδη. Εντάξει, τώρα ακόμα μαθαίνουμε, ας δούμε πώς μπορούμε να τα αποδείξουμε.

Τα περισσότερα θεωρήματα της άλγεβρας μεταγωγής αποδεικνύονται πάρα πολύ εύκολα με τη χρήση μιας τεχνικής που ονομάζεται *τέλεια επαγωγή*. Το αξίωμα A1 είναι το βασικό στοιχείο αυτής της τεχνικής: εφόσον μια μεταβλητή μεταγωγής μπορεί να πάρει μόνο δύο διαφορετικές τιμές, 0 και 1, μπορούμε να αποδείξουμε ένα θεώρημα που περιλαμβάνει μια μεταβλητή X αποδεικνύοντας ότι αυτό είναι αληθές τόσο για $X=0$ όσο και για $X=1$. Για παράδειγμα, για να αποδείξουμε το θεώρημα T1, κάνουμε δύο αντικαταστάσεις:

$$[X=0] \quad 0+0=0 \quad \text{αληθές, σύμφωνα με το αξίωμα A4'}$$

$$[X=1] \quad 1+0=1 \quad \text{αληθές, σύμφωνα με το αξίωμα A5'}$$

Όλα τα θεωρήματα του Πίνακα 4-1 είναι δυνατόν να αποδειχθούν με χρήση της τέλει επαγωγής, όπως θα σας ζητηθεί να κάνετε στα Προβλήματα 4.2 και 4.3.

4.1.3 Θεωρήματα δύο και τριών μεταβλητών

Τα θεωρήματα της άλγεβρας μεταγωγής με δύο ή τρεις μεταβλητές παρουσιάζονται στον Πίνακα 4-2. Κάθε ένα από τα θεωρήματα αυτά αποδεικνύεται εύκολα με τη χρήση της τέλει επαγωγής, με αξιολόγηση της πρότασης του θεωρήματος για τους τέσσερις δυνατούς συνδυασμούς των X και Y ή τους οκτώ δυνατούς συνδυασμούς των X , Y , και Z .

Τα δύο πρώτα ζεύγη θεωρημάτων αφορούν την αντιμεταθετικότητα και την προσεταιριστικότητα της λογικής πρόσθεσης και πολλαπλασιασμού, και είναι ταυτόσημα με τους νόμους της αντιμετάθεσης και του προσεταιρισμού της πρόσθεσης και του πολλαπλασιασμού ακεραίων και πραγματικών αριθμών. Η συνδυαστική χρήση τους δείχνει ότι η τοποθέτηση παρενθέσεων ή η σειρά των όρων σε ένα λογικό άθροισμα ή λογικό γινόμενο δεν έχει σημασία. Για παράδειγμα, από καθαρά αλγεβρική άποψη, μια παράσταση όπως η $W \cdot X \cdot Y \cdot Z$ είναι ασαφής. Μπορεί να γραφεί ως $(W \cdot (X \cdot (Y \cdot Z)))$ ή $((W \cdot X) \cdot Y) \cdot Z$ ή $(W \cdot X) \cdot (Y \cdot Z)$ (δείτε την Άσκηση 4.34). Ωστόσο, σύμφωνα με τα θεωρήματα, η ασαφής μορφή της παράστασης δε δημιουργεί πρόβλημα, αφού παίρνουμε τα ίδια αποτελέσματα σε κά-

θεώρημα

*τέλεια
επαγωγή
πεπερασμένη
επαγωγή*

Πίνακας 4-2 Θεωρήματα άλγεβρας μεταγωγής με δύο ή τρεις μεταβλητές.

(T6)	$X+Y=Y+X$	(T6')	$X \cdot Y=Y \cdot X$	(Αντιμεταθετικότητα)
(T7)	$(X+Y)+Z=X+(Y+Z)$	(T7')	$(X \cdot Y) \cdot Z=X \cdot (Y \cdot Z)$	(Προσεταιριστικότητα)
(T8)	$X \cdot Y+X \cdot Z=X \cdot (Y+Z)$	(T8')	$(X+Y) \cdot (X+Z)=X+Y \cdot Z$	(Επιμεριστικότητα)
(T9)	$X+X \cdot Y=X$	(T9')	$X \cdot (X+Y)=X$	(Κάλυψη)
(T10)	$X \cdot Y+X \cdot Y'=X$	(T10')	$(X+Y) \cdot (X+Y')=X$	(Συνδυασμός)
(T11)	$X \cdot Y+X' \cdot Z+Y \cdot Z=X \cdot Y+X' \cdot Z$			(Κοινή συναίνεση)
(T11')	$(X+Y) \cdot (X'+Z) \cdot (Y+Z)=(X+Y) \cdot (X'+Z)$			

θε περίπτωση. Ακόμη κι αν αλλάζαμε τη σειρά των μεταβλητών (π.χ. $X \cdot Z \cdot Y \cdot W$), θα παίρναμε και πάλι τα ίδια αποτελέσματα.

Όσο και αν αυτή η συζήτηση φαίνεται περιττή, το θέμα αυτό είναι πολύ σημαντικό καθώς διαμορφώνει τη θεωρητική βάση για τη χρήση λογικών πυλών με περισσότερες από δύο εισόδους. Έχουμε ορίσει τα σύμβολα $\cdot$ και $+$ ως *δυναδικούς τελεστές*, δηλαδή τελεστές που συνδυάζουν δύο εισόδους. Ωστόσο, στην πράξη χρησιμοποιούμε λογικές πύλες AND και OR δύο, τριών, ή και περισσότερων εισόδων. Σύμφωνα με τα θεωρήματα, μπορούμε να συνδέσουμε τις εισόδους των πυλών με οποιαδήποτε σειρά. Πολλά προγράμματα πλακετών τυπωμένων κυκλωμάτων και διάταξης ASIC, μάλιστα, επωφελούνται από αυτό το πλεονέκτημα. Μπορούμε να χρησιμοποιήσουμε ισοδύναμα είτε μια πύλη n εισόδων είτε $(n - 1)$ πύλες των 2 εισόδων η κάθε μία, αν και η καθυστέρηση και το κόστος είναι δυνατόν να είναι υψηλότερο στην περίπτωση των πολλών πυλών 2 εισόδων.

Το θεώρημα T8 είναι ταυτόσημο με τον επιμεριστικό νόμο των ακεραίων και των πραγματικών αριθμών, που σημαίνει ότι ο λογικός πολλαπλασιασμός επιμερίζεται πάνω στη λογική πρόσθεση. Συνεπώς, μπορούμε να “εκτελέσουμε τους επιμέρους πολλαπλασιασμούς” μιας παράστασης για να την πάρουμε σε μορφή αθροίσματος γινομένων, όπως στο παρακάτω παράδειγμα:

$$V \cdot (W+X) \cdot (Y+Z)=V \cdot W \cdot Y+V \cdot W \cdot Z+V \cdot X \cdot Y+V \cdot X \cdot Z$$

Ωστόσο, η άλγεβρα μεταγωγής έχει και την ασυνήθιστη ιδιότητα να αληθεύει και το αντίστροφο, δηλαδή η λογική πρόσθεση επιμερίζεται στο λογικό πολλαπλασιασμό, όπως αποδεικνύεται από το θεώρημα T8'. Έτσι μπορούμε να “εκτελέσουμε τις επιμέρους προσθέσεις” μιας παράστασης για να την πάρουμε σε μορφή γινομένου αθροισμάτων:

$$(V \cdot W \cdot X)+(Y \cdot Z)=(V+Y) \cdot (V+Z) \cdot (W+Y) \cdot (W+Z) \cdot (X+Y) \cdot (X+Z)$$

θεώρημα κάλυψης
κάλυψη
θεώρημα
συνδυασμού

Τα θεωρήματα T9 και T10 χρησιμοποιούνται ευρέως στην ελαχιστοποίηση των λογικών συναρτήσεων. Αν, για παράδειγμα, εμφανίζεται η επιμέρους παράσταση $X+X \cdot Y$ σε μια λογική παράσταση, το *θεώρημα κάλυψης* T9 λέει ότι μόνο το X χρειάζεται να συμπεριληφθεί στην παράσταση. Το X λέγεται ότι *καλύπτει* το $X \cdot Y$. Το *συνδυαστικό θεώρημα* T10

λέει ότι αν εμφανίζεται η επιμέρους παράσταση $X \cdot Y + X \cdot Y'$ σε μια παράσταση, μπορούμε να την αντικαταστήσουμε με το X . Εφόσον το Y μπορεί να είναι 0 ή 1, οποιαδήποτε από τις δύο περιπτώσεις της αρχικής επιμέρους παράστασης δίνουν 1 αν και μόνο αν το X είναι 1.

Παρά το γεγονός ότι μπορούμε εύκολα να αποδείξουμε το T9 με την τέλεια επαγωγή, η αλήθεια του T9 είναι προφανής αν την αποδείξουμε χρησιμοποιώντας τα θεωρήματα που αποδείξαμε ως τώρα:

$$\begin{aligned} X + X \cdot Y &= X \cdot 1 + X \cdot Y && (\text{σύμφωνα με το T1'}) \\ &= X \cdot (1 + Y) && (\text{σύμφωνα με το T8}) \\ &= X \cdot 1 && (\text{σύμφωνα με το T2}) \\ &= X && (\text{σύμφωνα με το T1'}) \end{aligned}$$

Με παρόμοιο τρόπο, τα υπόλοιπα θεωρήματα είναι δυνατόν να χρησιμοποιηθούν για να αποδείξουμε το T10, όπου το κύριο βήμα είναι να χρησιμοποιήσουμε το T8 για να ξαναγράψουμε το αριστερό μέρος ως $X \cdot (Y + Y')$.

Το θεώρημα T11 είναι γνωστό ως *θεώρημα κοινής συναίνεσης* (*consensus*). Ο όρος $Y \cdot Z$ λέγεται *όρος κοινής συναίνεσης* των $X \cdot Y$ και $X' \cdot Z$. Με άλλα λόγια, αν το $Y \cdot Z$ είναι 1, τότε είτε το $X \cdot Y$ είτε το $X' \cdot Z$ πρέπει να είναι επίσης 1, εφόσον το Y και το Z είναι και τα δύο 1 και είτε το X είτε το X' πρέπει να είναι 1. Έτσι, ο όρος $Y \cdot Z$ είναι πλεονάζων και μπορεί να απαλειφθεί από το δεξιό μέρος του T11. Το θεώρημα κοινής συναίνεσης έχει δύο σημαντικές εφαρμογές. Μπορεί να χρησιμοποιηθεί για την εξάλειψη ορισμένων κινδύνων χρονισμού σε συνδυαστικά λογικά κυκλώματα, όπως θα δούμε στην Ενότητα 4.5. Διαμορφώνει επίσης τη βάση της επαναληπτικής μεθόδου κοινής συναίνεσης για την εύρεση των πρωταρχικών όρων (prime implicants — δείτε τις Παραπομπές).

*θεώρημα κοινής
συναίνεσης
κοινή συναίνεση*

Σε όλα τα θεωρήματα, κάθε μεταβλητή είναι δυνατόν να αντικατασταθεί με οποιαδήποτε λογική παράσταση. Μια απλή αντικατάσταση είναι να συμπληρώσετε μία ή περισσότερες μεταβλητές:

$$(X + Y') + Z' = X + (Y' + Z') \quad (\text{βάσει του T7})$$

Είναι επίσης δυνατόν να αντικατασταθούν και πιο πολύπλοκες παραστάσεις:

$$(V' + X) \cdot (W \cdot (Y' + Z)) + (V' + X) \cdot (W \cdot (Y' + Z))' = V' + X \quad (\text{βάσει του T10})$$

4.1.4 Θεωρήματα n μεταβλητών

Αρκετά σημαντικά θεωρήματα, που παρατίθενται στον Πίνακα 4-3, είναι αληθή για οποιονδήποτε αριθμό μεταβλητών, n . Τα περισσότερα από τα θεωρήματα αυτά είναι δυνατόν να αποδειχθούν με τη χρήση μιας μεθόδου δύο βημάτων που λέγεται *πεπερασμένη επαγωγή*: αρχικά αποδεικνύουμε ότι το θεώρημα είναι αληθές για $n=2$ (*βασικό βήμα*) και στη συνέχεια αποδεικνύουμε ότι αν το θεώρημα είναι αληθές για $n=i$ τότε είναι επίσης αληθές για $n=i+1$ (*βήμα επαγωγής*). Για παράδειγμα, θεωρούμε το γενικευμένο θεώρημα αυτοδυναμίας T12. Για $n=2$, το T12 είναι ισοδύ-

βασικό βήμα

βήμα επαγωγής

ναμο με το T3 και επομένως είναι αληθές. Αν είναι αληθές για το λογικό άθροισμα των i X , τότε είναι επίσης αληθές για το άθροισμα των $i+1$ X , σύμφωνα με τον παρακάτω συλλογισμό:

$$\begin{aligned} X+X+X+\dots+X &= X+(X+X+\dots+X) && (i+1 \text{ } X \text{ σε οποιαδήποτε πλευρά}) \\ &= X+(X) && (\text{αν το T12 είναι αληθές για } n=i) \\ &= X && (\text{σύμφωνα με το T3}) \end{aligned}$$

Συνεπώς, το θεώρημα είναι αληθές για όλες τις πεπερασμένες τιμές του n .

θεωρήματα
DeMorgan

Τα θεωρήματα του DeMorgan (T13 και T13') είναι ίσως τα πιο συχνά χρησιμοποιούμενα θεωρήματα της άλγεβρας μεταγωγής. Σύμφωνα με το θεώρημα T13, μια πύλη AND n εισόδων της οποίας η έξοδος υφίσταται συμπλήρωμα είναι ισοδύναμη με μια πύλη OR n εισόδων της οποίας οι εισόδοι υφίστανται συμπλήρωμα. Αυτό σημαίνει ότι οι Εικόνες 4-3(α) και (β) είναι ισοδύναμες.

Στην Ενότητα 3.3.4 δείξαμε πώς κατασκευάζεται μια πύλη NAND τύπου CMOS. Η έξοδος μιας πύλης NAND για οποιοδήποτε σύνολο εισόδων είναι το συμπλήρωμα της εξόδου μιας πύλης AND για τις ίδιες εισόδους. Συνεπώς, μια πύλη NAND μπορεί να έχει το λογικό σύμβολο της Εικόνας 4-3(γ). Ωστόσο, το κύκλωμα NAND τύπου CMOS δε σχεδιάζεται ως μια πύλη AND ακολουθούμενη από έναν αντιστροφέα με τρανζίστορ (πύλη NOT), αλλά είναι απλώς μια συλλογή από τρανζίστορ που συμβαίνει να εκτελούν τη λειτουργία AND-NOT. Το θεώρημα T13, μάλιστα, λέει ότι το λογικό σύμβολο στο (δ) υποδεικνύει την ίδια λογική λειτουργία (τα κυκλάκια στις εισόδους της πύλης OR υποδεικνύουν τη λογική αντιστροφή). Αυτό σημαίνει ότι μια πύλη NAND μπορεί θεωρηθεί ότι εκτελεί μια λειτουργία NOT-OR.

Παρατηρώντας τις εισόδους και την έξοδο μιας πύλης NAND, είναι αδύνατο να προσδιορίσουμε κατά πόσον έχει κατασκευαστεί εσωτερικά ως μια πύλη AND ακολουθούμενη από έναν αντιστροφέα, ως αντιστροφείς ακολουθούμενοι από μια πύλη OR, ή ως απευθείας υλοποίηση CMOS, καθώς όλα τα κυκλώματα NAND εκτελούν ακριβώς την ίδια λει-

Πίνακας 4-3 Θεωρήματα άλγεβρας μεταγωγής με n μεταβλητές.

(T12)	$X+X+\dots+X = X$	(Γενικευμένη αυτοδυναμία)
(T12')	$X \cdot X \cdot \dots \cdot X = X$	
(T13)	$(X_1 \cdot X_2 \cdot \dots \cdot X_n)' = X_1' + X_2' + \dots + X_n'$	(Θεωρήματα DeMorgan)
(T13')	$(X_1 + X_2 + \dots + X_n)' = X_1' \cdot X_2' \cdot \dots \cdot X_n'$	
(T14)	$[F(X_1, X_2, \dots, X_n, +, \cdot)]' = F(X_1', X_2', \dots, X_n', \cdot, +)$	(Γενικευμένο θεώρημα DeMorgan)
(T15)	$F(X_1, X_2, \dots, X_n) = X_1 \cdot F(1, X_2, \dots, X_n) + X_1' \cdot F(0, X_2, \dots, X_n)$	(Θεωρήματα επέκτασης του Shannon)
(T15')	$F(X_1, X_2, \dots, X_n) = [X_1 + F(0, X_2, \dots, X_n)] \cdot [X_1' + F(1, X_2, \dots, X_n)]$	

τουργία. Αν και η επιλογή του συμβόλου δεν έχει επιπτώσεις στη λειτουργικότητα ενός κυκλώματος, στην Ενότητα 5.1 θα δείξουμε ότι η κατάλληλη επιλογή μπορεί να κάνει πιο κατανοητή τη λειτουργία του κυκλώματος.

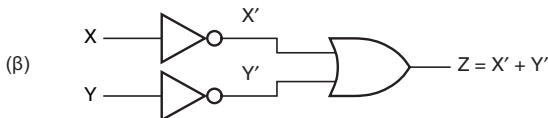
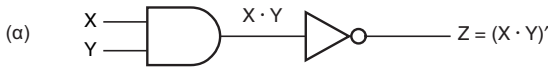
Ένα παρόμοιο συμβολικό ισοδύναμο μπορεί να συναχθεί από το θεώρημα T13'. Όπως φαίνεται στην Εικόνα 4-4, μια πύλη NOR μπορεί να υλοποιηθεί ως πύλη OR ακολουθούμενη από έναν αντιστροφέα ή ως αντιστροφείς ακολουθούμενοι από μια πύλη AND.

Τα θεώρηματα T13 και T13' είναι απλώς ειδικές περιπτώσεις του γενικευμένου θεωρήματος του DeMorgan T14, το οποίο εφαρμόζεται σε οποιαδήποτε λογική παράσταση F . Εξ ορισμού, το συμπλήρωμα μιας λογικής παράστασης, που εκφράζεται ως (F') , είναι μια παράσταση της οποίας η τιμή είναι αντίθετη εκείνης της F για κάθε δυνατό συνδυασμό εισόδων. Το θεώρημα T14 είναι πολύ σημαντικό καθώς μας υποδεικνύει έναν τρόπο χειρισμού και απλούστευσης του συμπληρώματος μιας παράστασης.

Το θεώρημα T14 λέει ότι μπορούμε να πάρουμε το συμπλήρωμα οποιασδήποτε λογικής παράστασης n μεταβλητών, εναλλάσσοντας τα $+$ και $\cdot$ μεταξύ τους και υπολογίζοντας τα συμπληρώματα όλων των μεταβλητών. Για παράδειγμα, έστω ότι έχουμε:

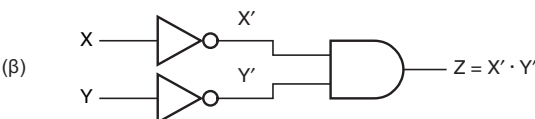
$$\begin{aligned} F(W,X,Y,Z) &= (W' \cdot X) + (X \cdot Y) + (W \cdot (X' + Z')) \\ &= ((W)' \cdot X) + (X \cdot Y) + (W \cdot ((X)' + (Z)')) \end{aligned}$$

γενικευμένο
θεώρημα DeMorgan
συμπλήρωμα
λογικής παράστασης



Εικόνα 4-3 Ισοδύναμα κυκλώματα σύμφωνα με το θεώρημα DeMorgan T13:

(α) AND-NOT, (β) NOT-OR, (γ) λογικό σύμβολο για πύλη NAND, (δ) ισοδύναμο σύμβολο για πύλη NAND.



Εικόνα 4-4 Ισοδύναμα κυκλώματα σύμφωνα με το θεώρημα του DeMorgan T13':

(α) OR-NOT, (β) NOT-AND, (γ) λογικό σύμβολο μιας πύλης NOR, (δ) ισοδύναμο σύμβολο μιας πύλης NOR.

Στη δεύτερη γραμμή έχουμε τοποθετήσει τα συμπληρώματα των μεταβλητών σε παρενθέσεις για να μας υπενθυμίζουν ότι το ' είναι ένας τελεστής και όχι μέρος του ονόματος της μεταβλητής. Εφαρμόζοντας το θεώρημα T14, παίρνουμε

$$[F(W,X,Y,Z)]' = ((W') + X') \cdot (X' + Y') \cdot (W' + ((X') \cdot (Z')))$$

Χρησιμοποιώντας το θεώρημα T4, η παράσταση μπορεί να απλοποιηθεί ως εξής:

$$[F(W,X,Y,Z)]' = (W + X') \cdot (X' + Y') \cdot (W' + (X \cdot Z))$$

Γενικά, μπορούμε να χρησιμοποιήσουμε το θεώρημα T14 για να πάρουμε το συμπλήρωμα μιας παράστασης με παρενθέσεις, εναλλάσσοντας τα + και - μεταξύ τους και υπολογίζοντας το συμπλήρωμα όλων των μη συμπληρωματικών μεταβλητών, καθώς και αποκαθιστώντας τις συμπληρωματικές μεταβλητές στη μη συμπληρωματική τους μορφή.

Μπορούμε να αποδείξουμε το γενικευμένο θεώρημα του DeMorgan T14 αν δείξουμε ότι όλες οι λογικές συναρτήσεις είναι δυνατόν να γραφούν είτε ως άθροισμα είτε ως γινόμενο επιμέρους συναρτήσεων και κατόπιν αν εφαρμόσουμε αναδρομικά τα θεωρήματα T13 και T13'. Ωστόσο, η απόδειξη που βασίζεται στην αρχή της δυικότητας, η οποία περιγράφεται παρακάτω, είναι πολύ πιο κατατοπιστική και ικανοποιητική.

4.1.5 Δυικότητα

Διατυπώσαμε όλα τα αξιώματα της άλγεβρας μεταγωγής σε ζευγάρια. Η τονισμένη εκδοχή κάθε αξιώματος (π.χ. A5') βγαίνει από τη μη τονισμένη εκδοχή του (π.χ. A5) με απλή εναλλαγή των 0 και 1 και, αν υπάρχουν, των $\cdot$ και + μεταξύ τους. Επομένως, μπορούμε να διατυπώσουμε το παρακάτω *μεταθεώρημα*, δηλαδή ένα θεώρημα για τα θεωρήματα:

Αρχή της δυικότητας Κάθε θεώρημα ή ταυτότητα της άλγεβρας μεταγωγής παραμένει αληθές αν εναλλάξουμε παντού τα 0 και 1 και τα $\cdot$ και + μεταξύ τους.

μεταθεώρημα

Το μεταθεώρημα είναι αληθές επειδή τα δυικά όλων των αξιωμάτων είναι αληθή, έτσι τα δυικά όλων των θεωρημάτων της άλγεβρας μεταγωγής είναι δυνατόν να αποδειχθούν με χρήση των δυικών των αξιωμάτων.

Τελικά, τι σημασία έχει το ζήτημα αυτό για τα ονόματα ή τα σύμβολα; Αν το λογισμικό που χρησιμοποιήθηκε για τη στοιχειοθεσία αυτού του βιβλίου είχε ένα σφάλμα που θα αντάλλαζε τα 0 - 1 και τα $\cdot$ - + παντού σε αυτό το κεφάλαιο, θα μαθαίνατε τη ίδια ακριβώς άλγεβρα μεταγωγής. Μόνο η ονοματολογία θα ήταν λίγο περίεργη, αν χρησιμοποιούνταν όροι όπως "γινόμενο" για την περιγραφή μιας πράξης που θα χρησιμοποιούσε το σύμβολο "+".

Η δυικότητα είναι σημαντική καθώς διπλασιάζει τη χρησιμότητα όλων όσων μάθατε για την άλγεβρα μεταγωγής και το χειρισμό των συναρτήσεων μεταγωγής. Πρόκειται για μια πιο πρακτική για τους σπουδαστές διατύπωση, που μειώνει στο μισό την προς εκμάθηση ύλη! Για παράδειγμα, από τη στιγμή που θα μάθει κανείς πώς να συνθέτει λογικά

κυκλώματα AND-OR δύο σταδίων από παραστάσεις με αθροίσματα γινομένων, αυτομάτως γνωρίζει μια δυική τεχνική για τη σύνθεση κυκλωμάτων OR-AND από παραστάσεις με γινόμενα αθροισμάτων.

Υπάρχει μία μόνο σύμβαση στην άλγεβρα μεταγωγής όπου δε θεωρούμε ταυτόσημα τα $\cdot$ και $+$ και συνεπώς η δυικότητα δε διατηρείται απαραίτητα αληθής. Μπορείτε να αντιληφθείτε ποια είναι προτού διαβάσετε την απάντηση που ακολουθεί; Θεωρήστε την ακόλουθη διατύπωση του θεωρήματος T9 και τη σαφώς άτοπη “δυική” της:

$$X+X \cdot Y = X \quad (\text{θεώρημα T9})$$

$$X \cdot X + Y = X \quad (\text{μετά την εφαρμογή της αρχής της δυικότητας})$$

$$X+Y = X \quad (\text{μετά την εφαρμογή του θεωρήματος T3'})$$

Προφανώς η τελευταία γραμμή είναι ψευδής, αλλά πού μπορεί να κάνουμε λάθος; Το πρόβλημα εντοπίζεται στην προτεραιότητα των τελεστών. Έχουμε τη δυνατότητα να γράψουμε το αριστερό μέρος της πρώτης γραμμής χωρίς παρενθέσεις λόγω της σύμβασης κατά την οποία το $\cdot$ έχει προτεραιότητα. Ωστόσο, εφαρμόζοντας την αρχή της δυικότητας, θα έπρεπε να έχουμε δώσει προτεραιότητα στο $+$ αντί για το $\cdot$ ή να γράψουμε τη δεύτερη γραμμή ως $X \cdot (X+Y) = X$. Ο καλύτερος τρόπος για να αποφύγουμε προβλήματα όπως αυτό είναι να τοποθετούμε παρενθέσεις στην παράσταση προτού πάρουμε τη δυική της.

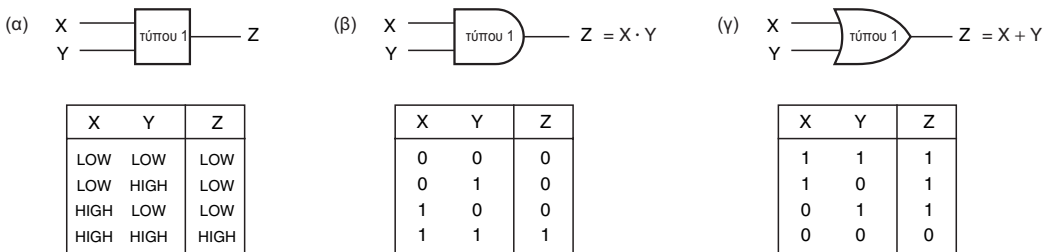
Ας ορίσουμε τυπικά τη *δυική μιας λογικής παράστασης*. Αν $F(X_1, X_2, \dots, X_n, +, \cdot, ')$ είναι μια λογική παράσταση που διατυπώνεται εξ ολοκλήρου με παρενθέσεις και περιλαμβάνει τις μεταβλητές $X_1, X_2, \dots, X_n$ και τους τελεστές $+$, $\cdot$ και $'$, τότε η δυική της παράστασης F , που γράφεται ως F^D , είναι η ίδια παράσταση με εναλλαγή των $+$ και $\cdot$ μεταξύ τους:

$$F^D(X_1, X_2, \dots, X_n, +, \cdot, ') = F(X_1, X_2, \dots, X_n, \cdot, +, ')$$

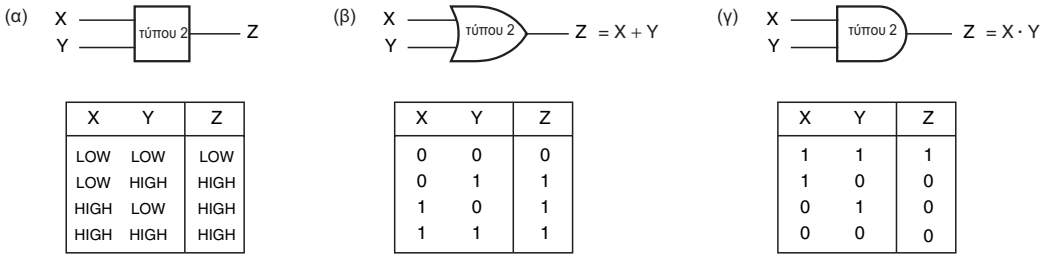
Φυσικά το γνωρίζετε ήδη αυτό, αλλά γράψαμε τον ορισμό με αυτόν τον τρόπο μόνο και μόνο για να δώσουμε έμφαση στην ομοιότητα ανάμεσα στη δυικότητα και στο γενικευμένο θεώρημα του DeMorgan T14, το οποίο μπορεί να επαναδιατυπωθεί ως εξής:

$$[F(X_1, X_2, \dots, X_n)]' = F^D(X_1', X_2', \dots, X_n')$$

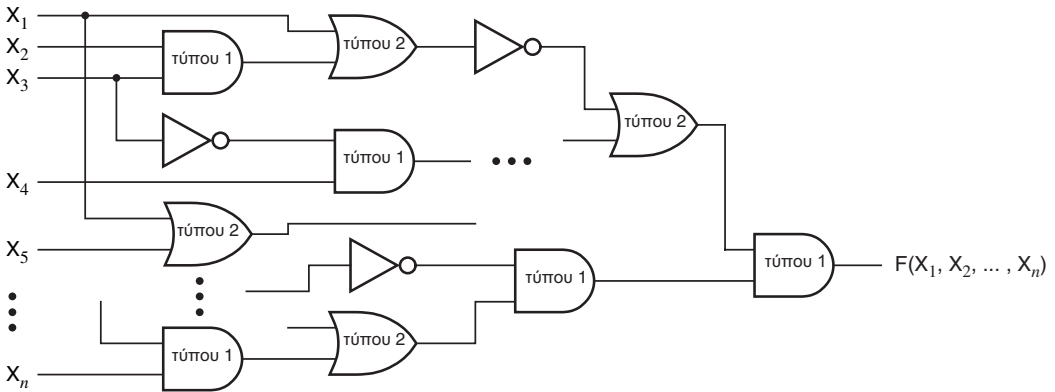
Ας εξετάσουμε την πρόταση αυτή στα πλαίσια ενός φυσικού δικτύου.



Εικόνα 4-5 Μια λογική πύλη “τύπου 1”: (α) πίνακας ηλεκτρικών λειτουργιών, (β) πίνακας λογικών λειτουργιών και σύμβολο με θετική λογική, (γ) πίνακας λογικών λειτουργιών και σύμβολο με αρνητική λογική.



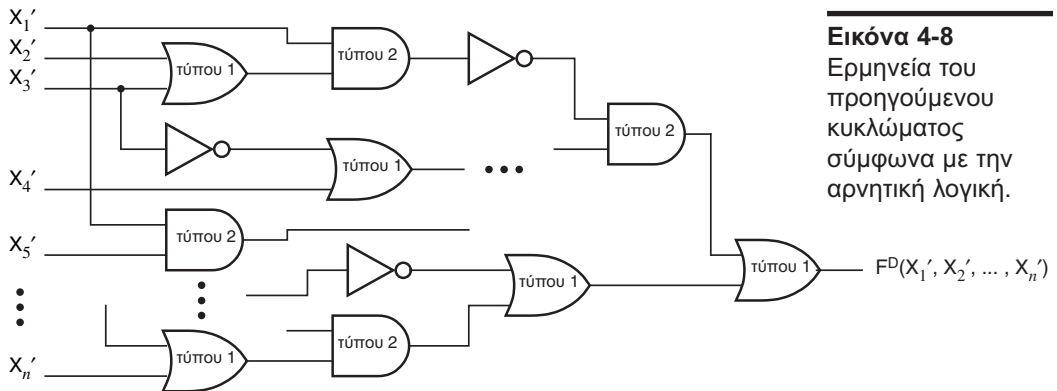
Εικόνα 4-6 Λογική πύλη “τύπου 2”: (α) πίνακας ηλεκτρικών λειτουργιών, (β) πίνακας λογικών λειτουργιών και σύμβολο με θετική λογική, (γ) πίνακας λογικών λειτουργιών και σύμβολο με αρνητική λογική.



Εικόνα 4-7 Κύκλωμα λογικής λειτουργίας που χρησιμοποιεί αντιστροφείς και πύλες τύπου 1 και τύπου 2 με σύμβαση θετικής λογικής.

Η Εικόνα 4-5(α) δείχνει τον πίνακα ηλεκτρικής λειτουργίας ενός λογικού στοιχείου το οποίο θα το αποκαλούμε απλώς πύλη “τύπου 1”. Σύμφωνα με τη σύμβαση της θετικής λογικής (LOW=0 και HIGH=1), πρόκειται για μια πύλη AND ενώ, σύμφωνα με τη σύμβαση της αρνητικής λογικής (LOW=1 και HIGH=0), πρόκειται για πύλη OR, όπως φαίνεται στα (β) και (γ). Μπορούμε επίσης να φανταστούμε μια πύλη “τύπου 2”, όπως φαίνεται στην Εικόνα 4-6, η οποία είναι πύλη OR θετικής λογικής ή πύλη AND αρνητικής λογικής. Παρόμοιοι πίνακες μπορούν να αναπτυχθούν για πύλες με περισσότερες από δύο εισόδους.

Έστω ότι μας δίνεται μια οποιαδήποτε λογική παράσταση $F(X_1, X_2, \dots, X_n)$. Ακολουθώντας τη σύμβαση θετικής λογικής, μπορούμε να κατασκευάσουμε ένα κύκλωμα που θα αντιστοιχεί στην παράσταση αυτή χρησιμοποιώντας αντιστροφείς για τις πράξεις NOT, πύλες τύπου 1 για τις πράξεις AND, και πύλες τύπου 2 για τις πράξεις OR, όπως φαίνεται στην Εικόνα 4-7. Υποθέτουμε τώρα ότι, χωρίς να αλλάξουμε αυτό το



κύκλωμα, αλλάζουμε απλώς τη λογική σύμβαση από θετική σε αρνητική. Μετά μπορούμε να επανασχεδιάσουμε το κύκλωμα όπως φαίνεται στην Εικόνα 4-8. Είναι ξεκάθαρο ότι, για κάθε δυνατό συνδυασμό τάσεων εισόδου (HIGH και LOW), το κύκλωμα εξακολουθεί να παράγει την ίδια τάση εξόδου. Ωστόσο, αν το δούμε από τη σκοπιά της άλγεβρας μεταγωγής, η τιμή εξόδου, 0 ή 1, θα είναι αντίθετη από εκείνη που θα παραγόταν με τη σύμβαση θετικής λογικής. Ομοίως, κάθε τιμή εισόδου είναι αντίθετη από εκείνη που ήταν προηγουμένως. Επομένως, για κάθε δυνατό συνδυασμό εισόδων στο κύκλωμα της Εικόνας 4-7, η έξοδος είναι αντίθετη από εκείνη που παράγεται από τον αντίθετο συνδυασμό που εφαρμόζεται στο κύκλωμα της Εικόνας 4-8:

$$F(X_1, X_2, \dots, X_n) = [F^D(X_1', X_2', \dots, X_n')]'$$

Παίρνοντας το συμπλήρωμα και των δύο πλευρών, προκύπτει το γενικευμένο θεώρημα του DeMorgan:

$$[F(X_1, X_2, \dots, X_n)]' = F^D(X_1', X_2', \dots, X_n')$$

Καταπληκτικό!

**Η ΔΥΙΚΟΤΗΤΑ
ΕΙΝΑΙ ΒΟΛΙΚΗ
ΚΑΙ ΓΙΑ ΤΟΥΣ
ΦΟΙΤΗΤΕΣ ΚΑΙ
ΓΙΑ ΤΟΥΣ
ΣΥΓΓΡΑΦΕΙΣ**

Διαπιστώσατε ότι η δυικότητα αποτελεί τη βάση του γενικευμένου θεωρήματος του DeMorgan. Στην πορεία, η δυικότητα θα περιορίσει στο μισό τον αριθμό των μεθόδων που πρέπει να μάθετε για να χειρίζεστε και να απλοποιείτε λογικές συναρτήσεις. Επίσης, περιορίσει στο μισό την ύλη που είχα να γράψω σε αυτές τις ενότητες!

4.1.6 Καθιερωμένες αναπαραστάσεις λογικών συναρτήσεων

Προτού προχωρήσουμε στην ανάλυση και τη σύνθεση συναρτήσεων συνδυαστικής λογικής, θα παρουσιάσουμε την αναγκαία ονοματολογία και σημειογραφία.



JOHN F. WAKERLY

ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ

ΑΡΧΕΣ & ΠΡΑΚΤΙΚΕΣ

ΤΡΙΤΗ ΑΜΕΡΙΚΑΝΙΚΗ ΕΚΔΟΣΗ ΕΝΗΜΕΡΩΜΕΝΗ

Αυτό το πρόσφατα αναθεωρημένο βιβλίο συνδυάζει την ακαδημαϊκή ακρίβεια με την πρακτική πείρα σε μια έγκυρη εισαγωγή ως προς τις βασικές αρχές της ψηφιακής σχεδίασης και των πρακτικών απαιτήσεων, τόσο για συστήματα σε επίπεδο πλακέτας όσο και σε συστήματα VLSI. Ο John Wakerly, ο οποίος έχει περισσότερα από 20 χρόνια πείρας τόσο στην πανεπιστημιακή όσο και στη βιομηχανική πρακτική, έχει εκπαιδεύσει άμεσα χιλιάδες σπουδαστές μηχανικούς, έχει διδάξει έμμεσα μέσω των βιβλίων του δεκάδες χιλιάδες σπουδαστές, και έχει συμμετάσχει ο ίδιος στο σχεδιασμό πραγματικών ψηφιακών συστημάτων τα οποία απέφεραν έσοδα δεκάδων εκατομμυρίων δολαρίων.

Το βιβλίο αυτό καλύπτει τα βασικά δομικά υλικά της ψηφιακής σχεδίασης σε διάφορα αφαιρετικά επίπεδα, από τις πύλες CMOS μέχρι τις γλώσσες περιγραφής υλικού. Σε καθένα από τα επίπεδα αυτά αναλύονται οι σημαντικές λειτουργίες όπως οι πύλες, οι αποκωδικοποιητές, οι πολυπλέκτες, τα κυκλώματα flip-flop, οι καταχωρητές, και οι μετρητές.

Στα νέα στοιχεία αυτής της έκδοσης περιλαμβάνεται η μετάβαση από τις παλιές "χειρωνακτικές" διαδικασίες στη σχεδίαση MSI, καθώς και η εξαρχής κάλυψη των διατάξεων PLD, FPGA, και των γλωσσών περιγραφής υλικού έτσι ώστε οι αναγνώστες να μπορούν να έχουν τη μέγιστη δυνατή υποστήριξη από τα σύγχρονα στοιχεία και εργαλεία λογισμικού. Η κάλυψη των γλωσσών περιγραφής υλικού περιλαμβάνει πλέον και τη VHDL, εκτός από την ABEL.

SILOS and SIMUCAD are registered trademarks and SILOS III is a trademark of Simucad, Inc. Xilinx is a registered trademark of Xilinx Inc.



ΚΛΕΙΔΑΡΙΘΜΟΣ

Σολωμού 57, 10432, ΑΘΗΝΑ, Τηλ. 210-5237635

Επισκεφθείτε μας στο Internet
<http://www.klidarithmos.gr>

ISBN 960-209-728-0



9 789602 097281

