

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ

ΔΟΜΗ & ΛΕΙΤΟΥΡΓΙΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ

(Θεωρία)

ΒΙΒΛΙΟ ΜΑΘΗΤΗ

2ου ΚΥΚΛΟΥ

ΚΑΤΕΥΘΥΝΣΗ-ΕΙΔΙΚΟΤΗΤΑ:
ΗΛΕΚΤΡΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΤΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΔΟΜΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ



ISBN 960-06-1039-8

ΟΡΓΑΝΙΣΜΟΣ ΕΚΔΟΣΕΩΝ ΔΙΔΑΚΤΙΚΩΝ ΒΙΒΛΙΩΝ
ΑΘΗΝΑ



**ΔΟΜΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ
ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ**

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ

Κ. Πεκμεστζή, Ι. Βογιατζής, Γ. Λιβιεράτος, Π. Μπουγάς

**ΔΟΜΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ
ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ**
(Θεωρία)

ΟΜΑΔΑ ΣΥΓΓΡΑΦΗΣ

- Πεκμεστζή Κιαμάλ, Αναπληρωτής Καθηγητή ΕΜΠ
- Βογιατζής Ιωάννης, Δρ. του Τμήματος Πληροφορικής του Πανεπιστημίου Αθηνών
- Λιβιεράτος Γεώργιος, Ηλεκτρολόγος Μηχανικός PhD in Electronics Πανεπιστημίου Southampton, UK
- Μπουγάς Πέτρος, Ηλεκτρολόγος Μηχανικός ΕΜΠ-Υποψήφιος Διδάκτορας

ΟΜΑΔΑ ΚΡΙΣΗΣ

- Μελέτης Χρήστος, Ηλεκτρολόγος Μηχανικός ΕΜΠ-Υποψήφιος Διδάκτορας
- Παναγιωτακόπουλος Γεώργιος, Ηλεκτρονικός ΤΕΙ Αθηνών, MSc State University of New York
- Σκανδαλίδης Χρήστος, Ηλεκτρολόγος Μηχανικός, Καθηγητής Δ/βάθμιας εκπαίδευσης ΠΕ-12

ΣΥΝΤΟΝΙΣΤΗΣ

Πεκμεστζή Κιαμάλ, Αναπληρωτής Καθηγητή ΕΜΠ

ΕΠΙΜΕΛΕΙΑ ΕΚΔΟΣΗΣ

Μπουγάς Πέτρος, Ηλεκτρολόγος Μηχανικός ΕΜΠ-Υποψήφιος Διδάκτορας

ΗΛΕΚΤΡΟΝΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΚΕΙΜΕΝΟΥ

Γκότσης Κωνσταντίνος

ΓΛΩΣΣΙΚΗ ΕΠΙΜΕΛΕΙΑ

Θεοδοσίου Αναστασία, Φιλολόγος, Καθηγήτρια Δ/θμιας Εκπαίδευσης, ΠΕ2

ΣΧΕΔΙΑΣΜΟΣ ΕΞΩΦΥΛΛΟΥ & ΠΡΟΕΚΤΥΠΩΣΗ

dimiourgies

ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ

Επιστημονικός Υπεύθυνος του Τομέα "ΗΛΕΚΤΡΟΝΙΚΩΝ",

Δρ ΧΑΡΑΛΑΜΠΟΣ ΔΗΜ. ΚΑΝΕΛΛΟΠΟΥΛΟΣ (PH.D)

(Σύμβουλος του Παιδαγωγικού Ινστιτούτου)

Με απόφαση της Ελληνικής Κυβερνήσεως τα διδακτικά βιβλία του Δημοτικού του Γυμνασίου και του Λυκείου τυπώνονται από τον Οργανισμό Εκδόσεων Διδακτικών Βιβλίων και διανέμονται δωρεάν.

ΒΙΒΛΙΟ ΜΑΘΗΤΗ

Α' τάξη
2ου ΚΥΚΛΟΥ

ΚΑΤΕΥΘΥΝΣΗ-ΕΙΔΙΚΟΤΗΤΑ:
ΗΛΕΚΤΡΟΝΙΚΟΣ ΟΠΤΙΚΟΑΚΟΥΣΤΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΟΡΓΑΝΙΣΜΟΣ ΕΚΔΟΣΕΩΣ ΔΙΔΑΚΤΙΚΩΝ ΒΙΒΛΙΩΝ
ΑΘΗΝΑ

Πρόλογος

Το βιβλίο αυτό αναφέρεται στη Δομή και Λειτουργία των Μικροϋπολογιστών και είναι σύμφωνο με τη φυσιογνωμία και το πρόγραμμα σπουδών του αντίστοιχου μαθήματος του 2ου Κύκλου Σπουδών, της Κατεύθυνσης "Ηλεκτρονικός Υπολογιστικών Συστημάτων και Δικτύων" του Τομέα Ηλεκτρονικών των ΤΕΕ. Προσεγγίζει τα Μικροϋπολογιστικά Συστήματα από τη σκοπιά της ηλεκτρονικής και των εφαρμογών τους που θα συναντήσουν οι απόφοιτοι αυτού του Τομέα. Είναι γνωστό ότι η σύγχρονη τεχνολογία βασίζεται στους Μικροεπεξεργαστές. Ειδικότερη μορφή τους είναι οι Μικροελεγκτές και εμφανίζονται σχεδόν σε κάθε ηλεκτρονική συσκευή και σχεδόν στο σύνολο των εφαρμογών της ηλεκτρονικής. Είναι λοιπόν απαραίτητο οι απόφοιτοι αυτού του Τομέα να είναι εξοικειωμένοι με αυτήν την τεχνολογία. Το αντικείμενο βέβαια είναι δύσκολο και πολύ εκτεταμένο. Για την προσέγγισή του από εκπαιδευτική άποψη επιλέχθηκε η εξής δομή: Για την πληρότητα της παρουσίασης τα δυο πρώτα κεφάλαια είναι εισαγωγικά. Στα επόμενα δυο κεφάλαια (3-4) γίνεται η παρουσίαση των βασικών εννοιών των Μικροϋπολογιστών και Μικροεπεξεργαστών σε μια γενική μορφή χωρίς να αναφερόμαστε σε ένα συγκεκριμένο τύπο. Ειδικότερα στο κεφάλαιο 3 γίνεται η παρουσίαση των Μικροϋπολογιστών, Μικροεπεξεργαστών και Μικροελεγκτών ενώ στο κεφάλαιο 4 παρουσιάζονται οι τρόποι διασύνδεσής τους. Τα παραπάνω είναι απαραίτητα γιατί είναι αναλλοίωτα στο χρόνο. Αν όμως η παρουσίαση έμενε στο γενικό επίπεδο δεν θα έδινε στο μαθητή ένα πρακτικό εφόδιο. Για το σκοπό αυτό κρίθηκε σκόπιμο να συγκεκριμενοποιηθεί η παρουσίαση σε ένα σύγχρονο, νεότερης τεχνολογίας Μικροελεγκτή που ταυτόχρονα να είναι και σχετικά απλός. Έτσι επιλέχθηκε ένας απλός Μικροελεγκτής PIC της εταιρείας Microchip που έχει ευρεία αποδοχή και χρησιμοποιείται στη πράξη σε μικρές εφαρμογές. Τα εκπαιδευτικά εργαλεία που παρέχονται δωρεάν για το συγκεκριμένο Μικροελεγκτή στη διεύθυνση www.microchip.com ήταν ένας ακόμα παράγοντας για την επιλογή του. Τα δυο λοιπόν τελευταία κεφάλαια (5-6) αφιερώθηκαν στην παρουσίαση του Μικροελεγκτή PIC (αρχιτεκτονικής και εντολών) και της χρήσης του σε εφαρμογές. Το μάθημα βέβαια για την σωστή κατανόησή του υποστηρίζεται από αντίστοιχο εργαστήριο.

Ελπίζοντας ότι θα δοθεί στο άμεσο μέλλον η δυνατότητα αναθεωρημένης έκδοσης του βιβλίου αναμένουμε παρατηρήσεις και σχόλια από τους αναγνώστες του βιβλίου στην ηλεκτρονική διεύθυνση paul@microlab.ntua.gr.

Περιεχόμενα

Κεφάλαιο 1. Βασικές Αρχές Δομής και Λειτουργίας των Υπολογιστικών Συστημάτων	
1.1 Αριθμητικά Συστήματα	15
1.2 Μετατροπή αριθμών από ένα σύστημα αρίθμησης σε άλλο	16
1.3 Πράξεις στο δυαδικό σύστημα	20
1.3.1 Πρόσθεση	20
1.3.2 Αφαίρεση	21
1.3.3 Πολλαπλασιασμός	22
1.3.4 Διάρθρωση	23
1.4 Πράξεις στο δεκαεξαδικό σύστημα	23
1.4.1 Πρόσθεση	23
1.4.2 Αφαίρεση	24
1.5 Παράσταση Αριθμών στον Υπολογιστή	25
1.6 Συστήματα αναπαράστασης συμβόλων στα υπολογιστικά συστήματα	26
1.7 Βασική δομή Υπολογιστικού Συστήματος - Αρχιτεκτονική	28
Κεφάλαιο 2. Εφαρμογές των ψηφιακών Ηλεκτρονικών στα Υπολογιστικά Συστήματα	
2.1 Άλγεβρα Boole	35
2.2 Λογικές Πύλες	36
2.3 Απομονωτές και transceivers	40
2.4 Κωδικοποιητές και αποκωδικοποιητές	42
2.4.1 Αποκωδικοποιητές	42
2.4.2 Κωδικοποιητές	45
2.5 Πολυπλέκτες και αποπλέκτες	46
2.5.1 Πολυπλέκτες	46
2.5.2 Αποπλέκτες	47
2.6 Στοιχειώδεις μονάδες άθροισης	48
2.6.1 Αθροιστές	48
2.6.2 Αφαιρέτες	52
2.7 Ακολουθιακές μονάδες-Στοιχεία μνήμης	53
2.8 Καταχωρητές	58
2.8.1 Παράλληλοι καταχωρητές	58
2.8.2 Καταχωρητές ολίσθησης	58
2.8.3 Καταχωρητές ολίσθησης με παράλληλη φόρτιση	59
2.9 Μετρητές ή απαριθμητές	61

Κεφάλαιο 3. Αρχιτεκτονική Ηλεκτρονικού Τμήματος (hardware) των Υπολογιστικών Συστημάτων	
3.1 Βασικά στοιχεία αρχιτεκτονικής μικρουπολογιστικών Συστημάτων	67
3.2 Αρχές λειτουργίας και αρχιτεκτονική μικροεπεξεργαστών	71
3.3 Εντολές μικροεπεξεργαστών	76
3.3.1 Εκτέλεση εντολής	77
3.3.2 Γλώσσα μηχανής και συμβολική γλώσσα	77
3.3.3 Κύκλοι εντολής και κύκλοι μηχανής	79
3.3.4 Είδη εντολών	80
3.4 Τρόποι αναφοράς στη μνήμη	81
3.5 Χαρακτηριστικά και κατηγορίες μικροεπεξεργαστών	84
3.6 Οικογένειες μικροεπεξεργαστών	89
3.7 Μικροελεγκτές	90
Κεφάλαιο 4. Σύνδεση μικροεπεξεργαστών και μικροελεγκτών	
4.1 Ακροδέκτες και συνδέσεις μικροεπεξεργαστών και μικροελεγκτών	95
4.1.1 Πολυπλεξία διαδρόμων	96
4.2 Προσπέλαση Συσκευών εισόδου-εξόδου	97
4.2.1 Θύρες εισόδου-εξόδου	98
4.2.2 Διευθυνσιοδότηση συσκευών εισόδου-εξόδου	99
4.2.3 Τρόποι προσπέλασης συσκευών εισόδου-εξόδου	100
4.3 Διακοπές	102
4.3.1 Πλεονεκτήματα της μεθόδου των διακοπών	102
4.4 Λειτουργία Απευθείας Προσπέλασης μνήμης	105
4.5 Είσοδος και έξοδος ψηφιακών δεδομένων σε μικροεπεξεργαστή	107
4.5.1 Είσοδος δεδομένων	107
4.5.2 Έξοδος δεδομένων	111
Κεφάλαιο 5. Αρχιτεκτονική και προγραμματισμός του μικροελεγκτή PIC	
5.1 Δομή του μικροελεγκτή PIC	117
5.1.1 Γενικά στοιχεία	117
5.1.2 Κεντρική Μονάδα Επεξεργασίας	117
5.1.3 Μνήμη	119
5.1.4 Εντολές	120
5.1.5 Λειτουργίες Διακοπών	121
5.1.6 Περιφερειακές μονάδες	122
5.2 Καταχωρητές	123
5.3 Τύποι εντολών του PIC	125
5.4 Κύκλος εκτέλεσης εντολής	127
5.5 Η γλώσσα Assembly του PIC	129
5.5.1 Εντολές αλλαγής περιεχομένου καταχωρητή	131
5.5.2 Εντολές αύξησης / μείωσης τιμής καταχωρητή	136
5.5.3 Εντολές αριθμητικών πράξεων	141
5.5.4 Εντολές λογικών πράξεων και διαδικασιών	149
5.5.5 Εντολές επεξεργασίας δυαδικού ψηφίου (bit)	166
5.5.6 Εντολές άλματος (αλλαγής ροής προγράμματος)	170

5.6 Μεθοδολογία προγραμματισμού σε γλώσσα Assembly	171
5.6.1 Ανάγνωση - Εγγραφή μνήμης	173
5.6.2 Ρουτίνες	175
5.6.3 Αλλαγή ροής προγράμματος υπό συνθήκη	179
5.6.4 Χρονική καθυστέρηση	181
5.6.5 Βρόχος WHILE - DO	182
5.7 Οι διακοπές του PIC	184
5.7.1 Καταχωρητής INTCON	184
5.7.2 Καταχωρητές PIE1 και PIR1	185
5.7.3 Καταχωρητές PIE2 και PIF2	186

Κεφάλαιο 6 Χρήση του μικροελεγκτή σε εφαρμογές	
6.1 Θύρες εισόδων / εξόδων	192
6.1.1 Θύρα A	192
6.1.2 Θύρα B	195
6.1.3 Θύρα C	197
6.1.4 Θύρες D και E	198
6.2 Χρονιστές	200
6.2.1 TIMER0	200
6.2.2 TIMER2	204
6.3 Μονάδα Σύλληψης / Σύγκρισης / Διαμόρφωσης εύρους παλμού (PWM)	207
6.3.1 Λειτουργία σύλληψης	209
6.3.2 Μονάδα σύγκρισης	209
6.3.3 Μονάδα διαμόρφωσης εύρους παλμού	209
6.4 Θύρα σειριακής επικοινωνίας	213
6.5 Μετατροπέας αναλογικού σήματος σε ψηφιακό	221
6.6 Επίλογος	226

Κεφάλαιο 1ο

Βασικές Αρχές Δομής και Λειτουργίας των Υπολογιστικών Συστημάτων

Περιεχόμενα

- 1.1 Αριθμητικά Συστήματα
- 1.2 Μετατροπή αριθμών από ένα σύστημα αρίθμησης σε άλλο
- 1.3 Πράξεις στο δυαδικό σύστημα
- 1.4 Πράξεις στο δεκαεξαδικό σύστημα
- 1.5 Παράσταση Αριθμών στον Υπολογιστή
- 1.6 Συστήματα αναπαράστασης συμβόλων στα υπολογιστικά συστήματα
- 1.7 Βασική δομή Υπολογιστικού Συστήματος
- Αρχιτεκτονική

1.1 Αριθμητικά Συστήματα

Ένα αριθμητικό σύστημα αποτελείται από ένα σύνολο ψηφίων και κανόνες εκτέλεσης των πράξεων ανάμεσα στους αριθμούς με βάση τα ψηφία αυτά.

Βάση (base) ενός αριθμητικού συστήματος είναι ένας αριθμός b ο οποίος χαρακτηρίζει το σύστημα. Το πλήθος των διαφορετικών ψηφίων του συστήματος είναι b . Τα πιο συχνά χρησιμοποιούμενα συστήματα είναι το δεκαδικό (με βάση το 10) το οποίο χρησιμοποιούμε στην καθημερινή ζωή, το δυαδικό (με βάση το 2) και το δεκαεξαδικό (με βάση το 16).

Η γνώση του δυαδικού συστήματος είναι ιδιαίτερα χρήσιμη στην κατανόηση των αρχών λειτουργίας των υπολογιστικών συστημάτων διότι η παράσταση της πληροφορίας και οι πράξεις στους υπολογιστές γίνονται στο δυαδικό σύστημα αρίθμησης. Το δεκαεξαδικό σύστημα από την άλλη έχει το πλεονέκτημα ότι το πλήθος των ψηφίων ενός αριθμού στο δεκαεξαδικό σύστημα είναι πολύ μικρότερο από το πλήθος των ψηφίων του ίδιου αριθμού στο δυαδικό σύστημα ενώ υπάρχει ένας εύκολος τρόπος μετατροπής των αριθμών από το ένα σύστημα στο άλλο και αντίστροφα. Έτσι, στους υπολογιστές συχνά, αντί να χρησιμοποιούμε τη δυαδική παράσταση ενός αριθμού χρησιμοποιούμε τη δεκαεξαδική παράσταση του.

Στον πίνακα 1.1 φαίνονται τα ψηφία τα οποία χρησιμοποιούνται σε κάθε ένα από τα συστήματα αυτά.

Δυαδικό σύστημα	Δεκαδικό σύστημα	Δεκαεξαδικό σύστημα
0	0	0
1	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
		A
		B
		C
		D
		E
		F

Πίνακας 1.1 Τα ψηφία του δυαδικού, δεκαδικού και δεκαεξαδικού συστήματος

Στόχοι του κεφαλαίου:

Όταν ολοκληρώσεις το κεφάλαιο αυτό θα μπορείς να...

- εξηγείς τη χρησιμότητα του δυαδικού και του δεκαεξαδικού συστήματος αρίθμησης
- εκτελείς πράξεις στο δυαδικό και δεκαεξαδικό σύστημα
- εκτελείς μετατροπές από το ένα αριθμητικό σύστημα στο άλλο
- αναφέρεις τη χρησιμότητα ενός συνόλου χαρακτήρων
- αναφέρεις τα πιο γνωστά σύνολα χαρακτήρων
- περιγράφεις τη βασική δομή των υπολογιστικών συστημάτων

Στο Κεφάλαιο αυτό θα δώσουμε τις βασικές αρχές της δομής και λειτουργίας των υπολογιστικών συστημάτων.

Αρχικά θα αναφερθούμε στην έννοια των αριθμητικών συστημάτων. Θα περιγράψουμε τα αριθμητικά συστήματα που χρησιμοποιούνται στους υπολογιστές (δυαδικό και δεκαεξαδικό) και θα εξηγήσουμε τη χρησιμότητά τους.

Στη συνέχεια θα περιγράψουμε πώς μπορούμε να μετατρέψουμε αριθμούς από οποιοδήποτε από τα συστήματα αυτά σε οποιοδήποτε άλλο, καθώς και τον τρόπο εκτέλεσης των πράξεων στο δυαδικό και το δεκαεξαδικό σύστημα.

Επιπλέον, θα αναφερθούμε στους τρόπους με τους οποίους μπορούμε να παραστήσουμε αριθμούς σε ένα υπολογιστικό σύστημα.

Ακόμη, θα αναφερθούμε στα συστήματα που χρησιμοποιούνται στα υπολογιστικά συστήματα για την αναπαράσταση συμβόλων (μη αριθμητικών χαρακτήρων), στη χρησιμότητά τους και θα περιγράψουμε τα πιο γνωστά από αυτά.

Τέλος, θα αναφερθούμε στη δομή των υπολογιστικών συστημάτων, καθώς και στον τρόπο διακίνησης και επεξεργασίας της πληροφορίας μέσα στα υπολογιστικά συστήματα.

Τα ψηφία A, B, C, D, E, F χρησιμοποιούνται στο δεκαεξαδικό σύστημα για να εκφράσουν τους αριθμούς 11, 12, 13, 14, 15, για τους οποίους δεν υπάρχουν αντίστοιχα ψηφία στο δεκαδικό σύστημα. Η γενική μορφή παράστασης ενός αριθμού N σε ένα αριθμητικό σύστημα είναι η ακόλουθη:

$$N=(a_{m-1}a_{m-2}...a_1a_0)_b=a_{m-1}b^{m-1} + a_{m-2}b^{m-2} + \dots + a_1b^1 + a_0b^0$$

Οι αριθμοί $a_0, a_1, \dots, a_{m-2}, a_{m-1}$ αποτελούν τα ψηφία του αριθμού και δε μπορούν να είναι μεγαλύτεροι από τη βάση b.

Ένας αριθμός μπορεί να εκφραστεί σε διαφορετικά αριθμητικά συστήματα. Έτσι, είναι δυνατό να επιβεβαιώσει κανείς ότι ο ίδιος αριθμός (28 στο δεκαδικό σύστημα) εκφράζεται στα συστήματα που αναφέρθηκαν όπως φαίνεται στη συνέχεια.

$$\begin{aligned} (28)_{10} &= 2 \times 10^1 + 8 \times 10^0 \\ (28)_{10} &= 1 \times 16^1 + 12 \times 16^0 = (1C)_{16} \\ (28)_{10} &= 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = (11100)_2 \end{aligned}$$

Αντίστροφα, η ίδια ακολουθία ψηφίων μπορεί να συμβολίζει διαφορετικούς αριθμούς σε διαφορετικά συστήματα, για παράδειγμα,

$$\begin{aligned} (11)_{16} &= 1 \times 16^1 + 1 \times 16^0 = 16 + 1 = (17)_{10} \\ (11)_{10} &= 1 \times 10^1 + 1 \times 10^0 = (11)_{10} \\ (11)_2 &= 1 \times 2^1 + 1 \times 2^0 = 2 + 1 = (3)_{10} \end{aligned}$$

Στη συνέχεια θα αναφερθούμε σε μετατροπές και πράξεις μόνο ακέραιων (όχι κλασματικών) αριθμών.

1.2 Μετατροπή αριθμών από ένα σύστημα αρίθμησης σε άλλο

Στην παράγραφο αυτή θα αναφερθούμε στις διαδικασίες μετατροπής ενός αριθμού από ένα σύστημα αρίθμησης σε κάποιο άλλο. Πιο συγκεκριμένα, θα παρουσιάσουμε τις διαδικασίες μετατροπής αριθμών από (α) το δυαδικό ή το δεκαεξαδικό στο δεκαδικό, (β) το δεκαδικό στο δυαδικό ή δεκαεξαδικό και (γ) το δυαδικό στο δεκαεξαδικό και αντίστροφα.

α. Μετατροπή από δυαδικό ή δεκαεξαδικό σε δεκαδικό

Για να μετατρέψουμε έναν αριθμό από το δυαδικό ή το δεκαεξαδικό στο δεκαδικό σύστημα αρίθμησης υπολογίζουμε την τιμή της παράστασης

$$a_{m-1}b^{m-1} + a_{m-2}b^{m-2} + \dots + a_1b^1 + a_0b^0$$

όπου με b συμβολίζουμε τη βάση του συστήματος, η οποία είναι το 2 ή το 16. Για

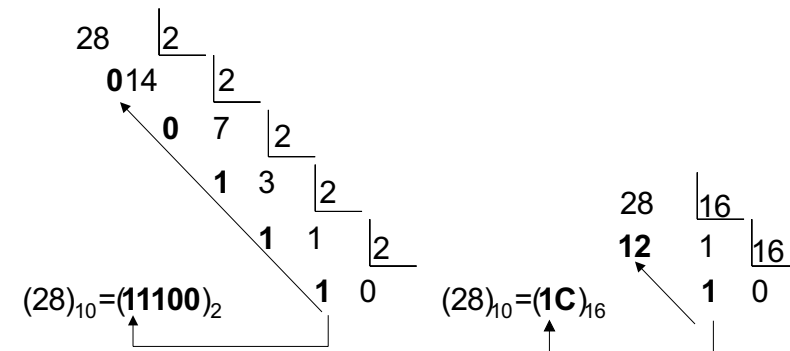
παράδειγμα, για να μετατρέψουμε τον αριθμό $(11001)_2$ στο δεκαδικό σύστημα, υπολογίζουμε την τιμή της παράστασης

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 16 + 8 + 1 = 25$$

β. Μετατροπή από το δεκαδικό στο δυαδικό ή το δεκαεξαδικό

Για να μετατρέψουμε το ακέραιο μέρος του αριθμού, το διαιρούμε με τη βάση του συστήματος (2 ή 16) και παίρνουμε ένα υπόλοιπο (Υ) και ένα πηλίκο (Π). Το πηλίκο διαιρείται και πάλι με το b (2 ή 16) και παίρνουμε ένα νέο πηλίκο Π και υπόλοιπο Υ. Η διαδικασία αυτή επαναλαμβάνεται μέχρι το πηλίκο Π να γίνει 0. Η ζητούμενη αναπαράσταση είναι τα υπόλοιπα (Υ), με την αντίστροφη σειρά από εκείνη που τα βρήκαμε.

Για παράδειγμα, στο Σχήμα 1.1 φαίνεται η διαδικασία μετατροπής του αριθμού 28 στο δυαδικό και το δεκαεξαδικό σύστημα αντίστοιχα. Οι αναπαραστάσεις του δεκαδικού αριθμού 28 στα δύο συστήματα είναι $(11100)_2$ και $(1C)_{16}$.



Σχήμα 1.1 Μετατροπή του αριθμού $(28)_{10}$ στο δυαδικό και το δεκαεξαδικό σύστημα αρίθμησης

γ. Μετατροπή από δυαδικό σε δεκαεξαδικό και αντίστροφα

Υπάρχουν δύο τρόποι για να μετατρέψουμε έναν αριθμό από το δυαδικό στο δεκαεξαδικό σύστημα αρίθμησης και αντίστροφα.

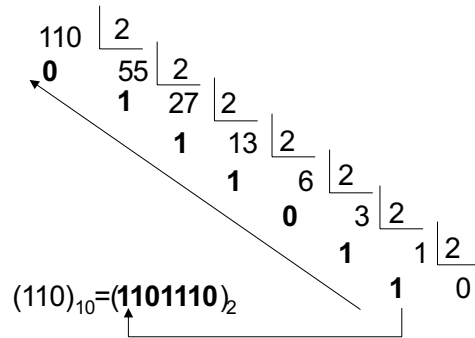
Ο πρώτος τρόπος είναι να χρησιμοποιήσουμε ως ενδιάμεσο το δεκαδικό σύστημα. Στον τρόπο αυτό μετατρέπουμε από το ένα σύστημα στο δεκαδικό και στη συνέχεια από το δεκαδικό στο άλλο όπως περιγράψαμε προηγουμένως. Στο δεύτερο τρόπο, μετατρέπουμε απευθείας από το ένα σύστημα στο άλλο. Οι δύο αυτοί τρόποι περιγράφονται στη συνέχεια.

Μετατροπή μέσω του δεκαδικού

Για να μετατρέψουμε το δεκαεξαδικό αριθμό 6E στο δυαδικό σύστημα μπορούμε να τον μετατρέψουμε πρώτα στο δεκαδικό αριθμό

$$6 \times 16^1 + 14 \times 16^0 = (110)_{10}$$

Στη συνέχεια μετατρέπουμε το δεκαδικό αριθμό στον αντίστοιχο δυαδικό αριθμό όπως φαίνεται στο Σχήμα 1.2.

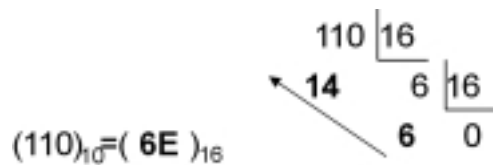


Σχήμα 1.2 Μετατροπή του αριθμού $(110)_{10}$ στο δυαδικό σύστημα αρίθμησης

Επομένως, η δυαδική παράσταση του αριθμού $(6E)_{16}$ είναι η $(1101110)_2$. Αντίστροφα, για τη δεκαεξαδική παράσταση του αριθμού $(1101110)_2$ βρίσκουμε πρώτα τη δεκαδική αναπαράσταση που είναι

$$2^6 + 2^5 + 2^3 + 2^2 + 2^1 = 110$$

Στη συνέχεια η μετατροπή στο δεκαεξαδικό σύστημα θα δώσει $(6E)_{16}$.



Σχήμα 1.3 Μετατροπή του αριθμού $(110)_{10}$ στο δεκαεξαδικό σύστημα αρίθμησης

Απευθείας μετατροπή

Η απευθείας μετατροπή αριθμών από το δυαδικό στο δεκαεξαδικό σύστημα και αντίστροφα στηρίζεται στο γεγονός ότι $16=2^4$, επομένως ένα ψηφίο στο δεκαεξαδικό σύστημα αντιστοιχεί σε τέσσερα ακριβώς ψηφία στο δυαδικό σύστημα. Με βάση την παρατήρηση αυτή μπορούμε να ακολουθήσουμε τη διαδικασία που περιγράφουμε στη συνέχεια.

Για την απευθείας μετατροπή ενός δεκαεξαδικού αριθμού στο δυαδικό σύστημα αντικαθιστούμε κάθε ψηφίο του αριθμού με ένα τετραψήφιο δυαδικό αριθμό σύμφωνα με τον πίνακα 1.2.

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

Πίνακας 1.2 Αντιστοιχία δεκαεξαδικών και δυαδικών ψηφίων

Μπορεί κανείς να παρατηρήσει ότι ο δυαδικός αριθμός (π.χ. 1100) είναι η έκφραση του δεκαεξαδικού ψηφίου στο δυαδικό σύστημα (π.χ. C). Έτσι, για παράδειγμα, ο δεκαεξαδικός αριθμός 77F αντιστοιχεί στο δυαδικό αριθμό 0111 0111 1111 όπως φαίνεται στη συνέχεια.

7 7 F
 0111 0111 1111

Αντίστροφα, για να μετατρέψουμε έναν αριθμό από το δυαδικό σύστημα στο δεκαεξαδικό, χωρίζουμε τα ψηφία του σε τετράδες προσθέτοντας, αν χρειαστεί, μηδενικά στην αρχή και αντικαθιστούμε κάθε τετράδα με το αντίστοιχο δεκαεξαδικό ψηφίο. Έτσι, ο δυαδικός αριθμός 1 1010 1101 αντιστοιχεί στο δεκαεξαδικό αριθμό 1AD, όπως φαίνεται στη συνέχεια (με πλάγια γράμματα φαίνονται τα μηδενικά που προσθέσαμε στην αρχή -αριστερά- του αριθμού προκειμένου να συμπληρωθούν τετράδες ψηφίων).

0001 1010 1101
 1 A D

Μπορεί κανείς να διαπιστώσει ότι η απευθείας μετατροπή είναι πολύ πιο εύκολη και γρήγορη από ότι η μετατροπή χρησιμοποιώντας το δεκαδικό σύστημα.

1.3 Πράξεις στο δυαδικό σύστημα

Στην παράγραφο αυτή θα περιγράψουμε τον τρόπο εκτέλεσης των τεσσάρων βασικών πράξεων (πρόσθεση, αφαίρεση, πολλαπλασιασμός και διαίρεση) στο δυαδικό σύστημα.

Το να γνωρίζουμε τον τρόπο εκτέλεσης των πράξεων στο δυαδικό σύστημα, θα μας βοηθήσει να καταλάβουμε τον τρόπο με τον οποίο πραγματοποιείται η εκτέλεση των πράξεων στα υπολογιστικά συστήματα.

1.3.1 Πρόσθεση

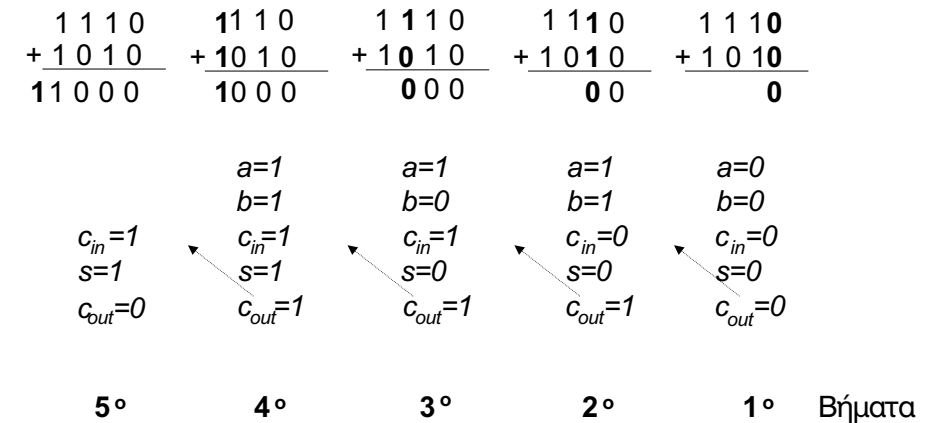
Στην πρόσθεση ξεκινάμε από δεξιά (χαμηλότερης τάξης ψηφία) και προσθέτουμε τα αντίστοιχα ψηφία των αριθμών. Κάθε φορά που προκύπτει κρατούμενο το προσθέτουμε στα αμέσως υψηλότερης τάξης ψηφία.

Για την κατανόηση της εκτέλεσης της πρόσθεσης στο δυαδικό σύστημα θα μας βοηθήσει ο πίνακας 1.3, που δίνει για τα δυνατά ζεύγη των προσθετέων ψηφίων (a, b) και του κρατουμένου (c_{in}), το αποτέλεσμα (s) και το κρατούμενο προς την επόμενη βαθμίδα (c_{out}).

a	b	c_{in}	s	c_{out}
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 1.3 Πίνακας αληθείας της πρόσθεσης δύο ψηφίων και του κρατουμένου

Στο Σχήμα 1.4 φαίνεται η διαδικασία πρόσθεσης των αριθμών $(1110)_2$ και $(1010)_2$ που δίνει αποτέλεσμα $(11000)_2$. Η αντίστοιχη πρόσθεση στο δεκαδικό σύστημα δίνει $14+10=24$ που επαληθεύει το αποτέλεσμα. Η διαδικασία της πρόσθεσης παρουσιάζεται από τα δεξιά προς τα αριστερά όπως δείχνει η αρίθμηση των βημάτων.



Σχήμα 1.4 Πρόσθεση στο δυαδικό σύστημα

1.3.2 Αφαίρεση

Στην αφαίρεση ξεκινάμε επίσης από δεξιά αφαιρώντας τα αντίστοιχα ψηφία των αριθμών. Σε κάθε βαθμίδα δημιουργείται ένα δανεικό (borrow) ψηφίο, το οποίο προστίθεται στο ψηφίο του αφαιρέτη της επόμενης βαθμίδας. Ο πίνακας 1.4 δίνει για τα ζεύγη των ψηφίων του αφαιρέτη, του αφαιρετέου και του δανεικού, το αποτέλεσμα και το δανεικό προς την επόμενη βαθμίδα.

a	b	r_{in}	d	r_{out}
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 1.4 Πίνακας αληθείας της αφαίρεσης δύο ψηφίων και του δανεικού

Για παράδειγμα, η διαδικασία αφαίρεσης των αριθμών $(1100)_2$ και $(1010)_2$ φαίνεται στο Σχήμα 1.5.

$\begin{array}{r} 1100 \\ -1010 \\ \hline 0010 \end{array}$	$\begin{array}{r} 1100 \\ -1010 \\ \hline 010 \end{array}$	$\begin{array}{r} 1100 \\ -1010 \\ \hline 10 \end{array}$	$\begin{array}{r} 1100 \\ -1010 \\ \hline 0 \end{array}$
$a=1$ $b=1$ $r_{in}=0$ $d=0$ $r_{out}=0$	$a=1$ $b=0$ $r_{in}=1$ $d=0$ $r_{out}=0$	$a=0$ $b=1$ $r_{in}=0$ $d=1$ $r_{out}=1$	$a=0$ $b=0$ $r_{in}=0$ $d=0$ $r_{out}=0$

Σχήμα 1.5 Αφαίρεση στο δυαδικό σύστημα

Η αντίστοιχη αφαίρεση στο δεκαδικό σύστημα θα έδινε $12-10=2$ που επαληθεύει το αποτέλεσμα.

1.3.3 Πολλαπλασιασμός

Ο πολλαπλασιασμός στο δυαδικό σύστημα αρίθμησης γίνεται, όπως και στο δεκαδικό, με διαδοχικές προσθέσεις. Κάθε ψηφίο του πολλαπλασιαστή πολλαπλασιάζεται με όλα τα ψηφία του πολλαπλασιαστέου και σχηματίζει ένα μερικό γινόμενο. Κάθε μερικό γινόμενο γράφεται κάτω από το προηγούμενο ολισθημένο κατά μία θέση προς τα αριστερά. Στη συνέχεια, προσθέτουμε ανά δύο τα μερικά γινόμενα.

Στο Σχήμα 1.6 φαίνεται ο δυαδικός πολλαπλασιασμός των αριθμών $(1110)_2$ και $(110)_2$. Ο πολλαπλασιασμός στο δεκαδικό σύστημα θα έδινε $14 \times 6 = 84 = (1010100)_2$.

$\begin{array}{r} 1110 \\ \times 110 \\ \hline 0000 \\ 1110 \\ \hline 1110 \end{array}$	$\begin{array}{r} 1110 \\ \times 110 \\ \hline 0000 \\ 1110 \\ \hline 1110 \end{array}$
← μερικό γινόμενο 1°	← μερικό γινόμενο 2°
$\begin{array}{r} 1110 \\ \times 10 \\ \hline 0000 \\ 1110 \\ \hline 1110 \end{array}$	$\begin{array}{r} 1110 \\ \times 10 \\ \hline 0000 \\ 1110 \\ \hline 1110 \end{array}$
← μερικό γινόμενο 3°	
	$\begin{array}{r} 1010100 \end{array}$

Σχήμα 1.6 Πολλαπλασιασμός στο δυαδικό σύστημα

1.3.4 Διαίρεση

Η διαίρεση στο δυαδικό σύστημα πραγματοποιείται με διαδοχικές αφαιρέσεις του διαιρέτη από το διαιρετέο. Στο Σχήμα 1.7 φαίνεται η διαδικασία διαίρεσης του αριθμού $(11011)_2$ με τον $(101)_2$ που δίνει πηλίκο $(101)_2$ και υπόλοιπο $(10)_2$. Η αντίστοιχη πράξη στο δεκαδικό σύστημα $(27 : 5)$ θα έδινε πηλίκο 5 και υπόλοιπο 2. Η διαδικασία είναι ανάλογη με αυτή που ακολουθούμε στους δεκαδικούς αριθμούς.

$\begin{array}{r} \overset{\cdot}{1} \overset{\cdot}{1} \overset{\cdot}{0} 1 1 \\ \underline{101} \\ 001 \end{array}$	$\begin{array}{r} \overset{\cdot}{1} \overset{\cdot}{1} \overset{\cdot}{0} 1 \\ \underline{101} \\ 0011 \end{array}$	$\begin{array}{r} \overset{\cdot}{1} \overset{\cdot}{1} \overset{\cdot}{0} 1 1 \\ \underline{101} \\ 00111 \end{array}$	$\begin{array}{r} \overset{\cdot}{1} \overset{\cdot}{1} \overset{\cdot}{0} 1 1 \\ \underline{101} \\ 00111 \\ \underline{101} \\ 010 \end{array}$
---	--	---	---

Σχήμα 1.7 Διαίρεση στο δυαδικό σύστημα

1.4 Πράξεις στο δεκαεξαδικό σύστημα

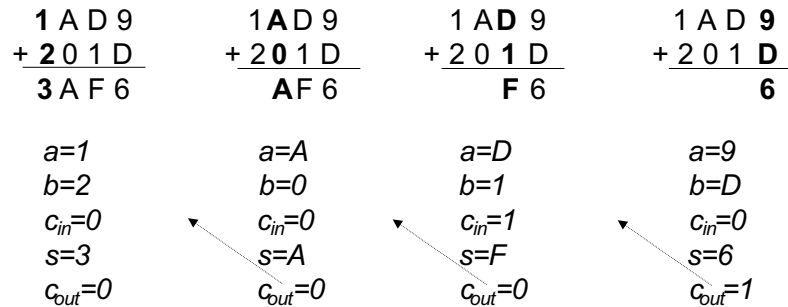
Η εκτέλεση των πράξεων στο δεκαεξαδικό σύστημα είναι πιο πολύπλοκη από ότι στο δυαδικό. Ο λόγος που μαθαίνουμε πράξεις στο σύστημα αυτό είναι ότι πολλές φορές χρησιμοποιούμε το δεκαεξαδικό σύστημα αντί του δυαδικού, επειδή το πλήθος των ψηφίων ενός αριθμού είναι πολύ μικρότερο από ότι στο δυαδικό σύστημα.

1.4.1 Πρόσθεση

Η πρόσθεση στο δεκαεξαδικό σύστημα γίνεται όπως στο δεκαδικό, ξεκινώντας από τα δεξιά και προσθέτοντας ανά δύο τα ψηφία, συν το κρατούμενο που προέκυψε από την πρόσθεση των προηγούμενων ψηφίων (αν υπάρχει).

Για να προσθέσουμε δύο ψηφία στο δεκαεξαδικό σύστημα υπολογίζουμε τη δεκαδική τους τιμή (αν κάποιο από αυτά είναι μεταξύ του 'A' και του 'F'), προσθέτουμε τις δεκαδικές τους τιμές. Αν το αποτέλεσμα είναι μικρότερο του 16 το παριστάνουμε με το αντίστοιχο δεκαεξαδικό ψηφίο. Αν το αποτέλεσμα είναι μεγαλύτερο ή ίσο του 16, τότε

αφαιρούμε το 16, και το υπόλοιπο είναι το αποτέλεσμα και έχουμε κρατούμενο 1. Έτσι, η πρόσθεση 9 + 9 δίνει αποτέλεσμα (18)₁₀ επομένως έχουμε αποτέλεσμα 2 και κρατούμενο 1. Στο Σχήμα 1.8 φαίνεται η διαδικασία πρόσθεσης των αριθμών (1AD9)₁₆ και (201D)₁₆.



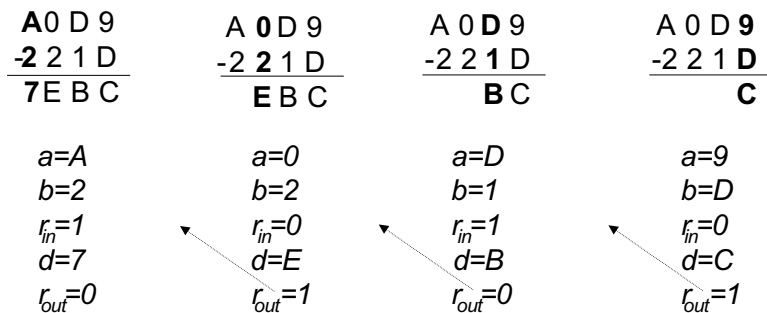
Σχήμα 1.8 Πρόσθεση στο δεκαεξαδικό σύστημα

Η αντίστοιχη πράξη στο δεκαδικό σύστημα (6873 + 8221) δίνει 15094 = (3AF6)₁₆ που επαληθεύει το αποτέλεσμα.

1.4.2 Αφαίρεση

Η αφαίρεση στο δεκαεξαδικό σύστημα γίνεται όπως στο δεκαδικό, ξεκινώντας από τα δεξιά και αφαιρώντας το ψηφίο του αφαιρέτη από το ψηφίο του αφαιρετέου. Αν υπάρχει δανεικό από προηγούμενη βαθμίδα, προστίθεται στο ψηφίο του αφαιρέτη.

Στην περίπτωση που το ψηφίο του αφαιρέτη είναι μεγαλύτερο από το ψηφίο του αφαιρετέου, δε δανειζόμαστε από την επόμενη βαθμίδα 10 όπως στο δεκαδικό σύστημα αρίθμησης, αλλά 16 (που είναι η βάση του συστήματος).



Σχήμα 1.9 Αφαίρεση στο δεκαεξαδικό σύστημα

Για παράδειγμα, η διαδικασία της αφαίρεσης των αριθμών (41177)₁₀ = (A0D9)₁₆ και (8733)₁₀ = (221D)₁₆ που δίνει αποτέλεσμα (32444)₁₀ = (7EBC)₁₆ φαίνεται στο Σχήμα 1.9.

1.5 Παράσταση Αριθμών στον Υπολογιστή

Όπως γνωρίζουμε, οι πληροφορίες αποθηκεύονται στη μνήμη του υπολογιστή. Η μνήμη του υπολογιστή είναι οργανωμένη σε λέξεις (ομάδες δυαδικών ψηφίων). Μια λέξη είναι μια ομάδα δυαδικών ψηφίων (συνήθως 8 ή 16 δυαδικά ψηφία).

Στην παράγραφο αυτή θα ασχοληθούμε με τον τρόπο με τον οποίο αποθηκεύονται οι αριθμοί στη μνήμη του υπολογιστή. Από όσα έχουμε αναφέρει μέχρι τώρα, γνωρίζουμε ότι στον υπολογιστή χρησιμοποιείται η δυαδική αναπαράσταση των αριθμών. Αυτό που μένει να συζητηθεί είναι: πώς παρίστανται οι θετικοί και πώς οι αρνητικοί αριθμοί;

Παράσταση αρνητικών αριθμών

Για να παραστήσουμε σε δυαδική μορφή θετικούς και αρνητικούς αριθμούς χρησιμοποιούμε το αριστερότερο ψηφίο της λέξης (most significant bit). Αν το ψηφίο αυτό είναι 0, τότε ο αριθμός είναι θετικός, διαφορετικά είναι αρνητικός.

Για να παραστήσουμε ένα θετικό αριθμό, χρησιμοποιούμε το αριστερότερο δυαδικό ψηφίο για το πρόσημο (0) και τα υπόλοιπα ψηφία τα χρησιμοποιούμε για να κωδικοποιήσουμε το μέτρο του αριθμού.

Έτσι για παράδειγμα, χρησιμοποιώντας 8 δυαδικά ψηφία (όλα για το ακέραιο μέρος του αριθμού) ο αριθμός (+18)₁₀ είναι (00010010)₂.

Υπάρχουν τρεις τρόποι για να κωδικοποιήσουμε έναν αρνητικό αριθμό:

- η παράσταση προσήμου μέτρου
- η παράσταση συμπληρώματος ως προς 1
- η παράσταση συμπληρώματος ως προς 2

Στη συνέχεια θα αναφερθούμε συνοπτικά στις παραστάσεις αυτές.

Παράσταση Προσήμου Μέτρου

Στην παράσταση προσήμου μέτρου, χρησιμοποιούμε το αριστερότερο ψηφίο σαν ένδειξη του προσήμου, και τα υπόλοιπα ψηφία για το μέτρο του αριθμού.

Για παράδειγμα, αν χρησιμοποιούμε οκτώ δυαδικά ψηφία για την παράσταση των αριθμών (όλα για το ακέραιο μέρος του αριθμού) και θέλουμε να παραστήσουμε τον αριθμό -18 με την παράσταση μέτρου, θα εργαστούμε ως εξής. Βρίσκουμε την παράσταση του αριθμού 18 με οκτώ δυαδικά ψηφία (00010010). Στη συνέχεια αντιστρέφουμε το πρώτο ψηφίο, προκειμένου να δείξουμε ότι ο αριθμός είναι

αρνητικός. Έτσι η ζητούμενη παράσταση είναι η 10010010.

Παράσταση Συμπληρώματος ως προς 1

Στην παράσταση συμπληρώματος ως προς 1, βρίσκουμε την αντίστοιχη δυαδική παράσταση του θετικού αριθμού και αντιστρέφουμε όλα τα ψηφία του.

Για παράδειγμα για να παραστήσουμε τον αριθμό -18 με παράσταση συμπληρώματος ως προς 1, ξεκινάμε από την παράσταση του θετικού αριθμού 00010010 και αντιστρέφουμε τα ψηφία του για να καταλήξουμε στην παράσταση 11101101.

Παράσταση συμπληρώματος ως προς 2

Στην παράσταση συμπληρώματος ως προς 2, χρησιμοποιούμε το θετικό αριθμό, αντιστρέφουμε όλα τα ψηφία του και στη συνέχεια προσθέτουμε μία μονάδα.

Έτσι, για να παραστήσουμε τον αριθμό -18 με παράσταση συμπληρώματος ως προς 2, ξεκινάμε από την παράσταση του θετικού αριθμού 00010010. Αντιστρέφοντας τα ψηφία βρίσκουμε την παράσταση 11101101. Στη συνέχεια, προσθέτοντας το 1, βρίσκουμε την παράσταση συμπληρώματος ως προς 2 που είναι η 11101110.

Αξίζει να παρατηρήσει κανείς, ότι και στους τρεις τρόπους παράστασης των αρνητικών αριθμών που αναφέρθηκαν, το αριστερότερο ψηφίο του αριθμού δείχνει αν ο αριθμός είναι θετικός ή αρνητικός. Έτσι, αν το ψηφίο είναι '1', μπορεί κανείς να συμπεράνει ότι ο αριθμός είναι αρνητικός.

1.6 Συστήματα αναπαράστασης συμβόλων στα υπολογιστικά συστήματα

Γνωρίζουμε ότι, γενικά, εκτός από αριθμούς μπορούμε να αναπαραστήσουμε σε έναν υπολογιστή και σύμβολα, όπως γράμματα, σημεία στίξης και αριθμητικά ψηφία. Οι πληροφορίες αυτές παριστάνονται στους υπολογιστές ως σύνολα δυαδικών ψηφίων.

Ένα πρόβλημα που παρουσιάστηκε στα πρώτα χρόνια της εμφάνισης των υπολογιστών ήταν η αντιστοίχιση των συμβόλων σε δυαδικά ψηφία. Τα χρόνια εκείνα ήταν δυνατό σε έναν υπολογιστή το γράμμα 'Ε' να συμβολίζεται με την ακολουθία δυαδικών ψηφίων 0010 0111, ενώ σε έναν άλλο υπολογιστή με την ίδια ακολουθία να αντιστοιχίζεται το γράμμα 'Α'. Αν κανείς μετέφερε ένα αρχείο κειμένου από τον έναν υπολογιστή στον άλλο, το γράμμα 'Ε' θα εκλαμβάνονταν στο δεύτερο υπολογιστή ως 'Α' με τις συνέπειες που θα είχε αυτό.

Εμφανίστηκε η ανάγκη ύπαρξης μιας κοινά αποδεκτής αντιστοίχισης χαρακτήρων σε δυαδικές ακολουθίες. Μια τέτοια αντιστοίχιση ονομάζεται σύνολο χαρακτήρων (character set) και συνήθως καθορίζεται από κάποιο διεθνή οργανισμό τυποποίησης. Ένα τέτοιο σύνολο χαρακτήρων, για μικροϋπολογιστές και σταθμούς εργασίας είναι ο

κώδικας ASCII (American Standard Code for Information Interchange, Αμερικανικός πρότυπος κώδικας για την ανταλλαγή πληροφοριών). Στον κώδικα ASCII, κάθε χαρακτήρας αναπαρίσταται από 8 δυαδικά ψηφία. Με τον κώδικα αυτό μπορούμε να παραστήσουμε $2^8=256$ διαφορετικά σύμβολα. Μπορούμε να παραστήσουμε τα γράμματα του λατινικού αλφάβητου, τα αριθμητικά ψηφία, τα σύμβολα στίξης, των πράξεων, και ειδικούς χαρακτήρες (#,\$,%,&,@,#), κλπ.

Ένα άλλο σύνολο χαρακτήρων είναι ο διευρυμένος δυαδικός κώδικας δεκαδικών για επικοινωνία (Extended Binary Coded Decimal Interchange Code, EBCDIC). Στον επόμενο πίνακα δίνουμε παραδείγματα συμβόλων στους κώδικες ASCII και EBCDIC.

Σύμβολο	Κώδικας ASCII	Κώδικας EBCDIC
A	01000001	11000001
B	01000010	11000010
C	01000011	11000011
D	01000100	11000100
E	01000101	11000101
0	00110000	11110000
1	00110001	11110001
2	00110010	11110010
!	00100001	01011010
#	00100011	01111011
\$	00100100	01011011
%	00100101	01101100
(00101000	01001101
+	00101011	01001110
-	00101001	01001111
*	00101010	01011100

Πίνακας 1.5 Σύμβολα στους κώδικες ASCII και EBCDIC

Για την αναπαράσταση χαρακτήρων που δεν περιλαμβάνονται στο λατινικό αλφάβητο, κάθε χώρα έχει θεσπίσει ένα πρότυπο, το οποίο αποτελεί επέκταση των παραπάνω προτύπων. Το πιο γνωστό πρότυπο για την παράσταση των ελληνικών και λατινικών χαρακτήρων, είναι το πρότυπο 928 του Ελληνικού Οργανισμού Τυποποίησης (ΕΛΟΤ 928). Ο κώδικας αυτός αποτελεί μία επέκταση του κώδικα ASCII και περιλαμβάνει εκτός των λατινικών και τους ελληνικούς χαρακτήρες κεφαλαίους, πεζούς, τονούμενους, σημεία στίξης, κλπ.

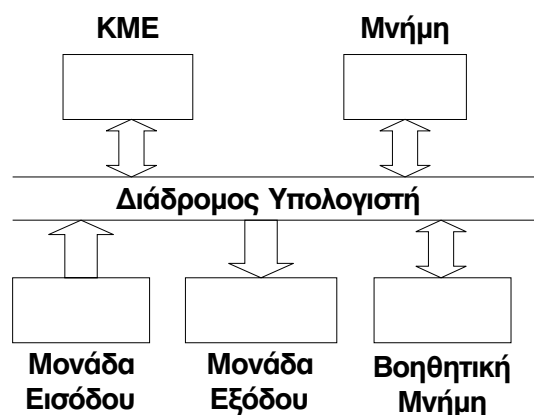
Με τους 256 χαρακτήρες που μπορούν να αναπαραστήσουν οι κώδικες ASCII και EBCDIC δεν είναι δυνατό να παρασταθούν οι χαρακτήρες όλων των αλφαβητών (π.χ. ελληνικό, ασιατικά, γερμανικό, γαλλικό). Για το λόγο αυτό δημιουργήθηκε ένα ακόμη

σύνολο χαρακτήρων με το όνομα **Unicode**. Στο σύνολο αυτό χρησιμοποιούνται 16 δυαδικά ψηφία για την αναπαράσταση κάθε ψηφίου (είναι, όπως λέμε, ένας 16-δικός κώδικας). Το πλήθος των διαφορετικών ψηφίων που μπορεί να παραστήσει ο κώδικας αυτός είναι και $2^{16}=65536$. Επομένως, το σύνολο αυτό καλύπτει τις ανάγκες για αναπαράσταση όλων των υπαρχόντων αλφάβητων.

1.7 Βασική δομή υπολογιστικού συστήματος - Αρχιτεκτονική

Στην παράγραφο αυτή θα περιγράψουμε συνοπτικά τη γενική δομή ενός υπολογιστικού συστήματος. Στο Κεφάλαιο 3 θα παρουσιάσουμε πιο αναλυτικά τη δομή αυτή.

Όπως γνωρίζουμε, ένα υπολογιστικό σύστημα είναι ένα σύστημα που αποτελείται από υλικό και λογισμικό και το οποίο επεξεργάζεται δεδομένα. Με τον όρο υλικό αναφερόμαστε στις συσκευές και με τον όρο λογισμικό στα προγράμματα, δηλαδή σε μια σειρά εντολών οι οποίες εκτελούνται από το υπολογιστικό σύστημα. Η γενική δομή του υλικού ενός υπολογιστικού συστήματος φαίνεται στο Σχήμα 1.10.



Σχήμα 1.10 Γενική δομή υπολογιστικού συστήματος

Σύμφωνα με το Σχήμα 1.10, ένα υπολογιστικό σύστημα αποτελείται από:

- **μονάδες εισόδου** με τις οποίες μπορούμε να εισάγουμε δεδομένα στον υπολογιστή (π.χ. πληκτρολόγιο).

- **μονάδες εξόδου** με τις οποίες το υπολογιστικό σύστημα εμφανίζει τα αποτελέσματα της επεξεργασίας (π.χ. οθόνη, εκτυπωτής).
- **Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)** η οποία επεξεργάζεται τα δεδομένα.
- **κύρια μνήμη** στην οποία αποθηκεύονται προσωρινά δεδομένα.
- **μονάδες βοηθητικής μνήμης** στις οποίες αποθηκεύονται δεδομένα όπου θα παραμείνουν και μετά τη λήξη της λειτουργίας του υπολογιστικού συστήματος.
- **τους διαδρόμους**, με τους οποίους επικοινωνούν μεταξύ τους οι παραπάνω μονάδες.

Στη συνέχεια περιγράψουμε τη δομή και τη λειτουργία των μονάδων αυτών.

Κεντρική Μονάδα Επεξεργασίας

Η Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) επεξεργάζεται δεδομένα. Η επεξεργασία των δεδομένων γίνεται σε μια σειρά από βήματα, κάθε ένα από τα οποία ονομάζεται εντολή. Οι εντολές που εκτελούνται από την ΚΜΕ είναι εντολές σε γλώσσα μηχανής. Μια εντολή σε γλώσσα μηχανής είναι μια σειρά από δυαδικά ψηφία όπου είναι κωδικοποιημένο το είδος της εντολής. Οι εντολές της γλώσσας μηχανής είναι αποθηκευμένες στην κύρια μνήμη, από όπου τις ανακαλεί και τις εκτελεί η ΚΜΕ.

Η ΚΜΕ αποτελείται από τρία τμήματα:

- την αριθμητική και λογική μονάδα (Arithmetic and Logic Unit, ALU), η οποία εκτελεί τις αριθμητικές και λογικές πράξεις
- τη μονάδα ελέγχου (control unit) η οποία συντονίζει την εκτέλεση των εντολών και την εκτέλεση των πράξεων στην αριθμητική και λογική μονάδα
- τους καταχωρητές (registers) οι οποίοι χρησιμεύουν ως χώροι αποθήκευσης δεδομένων (καταχωρητές δεδομένων) ή διευθύνσεων της μνήμης (καταχωρητές διευθύνσεων)

Κύρια Μνήμη

Στην κύρια μνήμη φυλάσσονται δεδομένα ή εντολές τις οποίες θα εκτελέσει η ΚΜΕ. Η κύρια μνήμη αποτελείται από λέξεις μνήμης (memory words), κάθε μια από τις οποίες αποτελείται από δυαδικά ψηφία. Κάθε θέση έχει μια συγκεκριμένη διεύθυνση (address). Για να διαβάσουμε από μια θέση μνήμης ή να γράψουμε σε αυτή πρέπει να γνωρίζουμε τη διεύθυνσή της.

Η κύρια μνήμη ενός υπολογιστικού συστήματος χωρίζεται σε δύο τμήματα: στη μνήμη από την οποία η ΚΜΕ μπορεί να διαβάσει μόνο (Read Only Memory, ROM) και στη μνήμη στην οποία η ΚΜΕ μπορεί και να γράψει και να διαβάσει. Στη μνήμη αυτή αναφερόμαστε με τον όρο μνήμη τυχαίας προσπέλασης (Random Access Memory, RAM). Τα περιεχόμενα της μνήμης RAM χάνονται όταν σταματήσει να λειτουργεί το υπολογιστικό σύστημα.

Στη μνήμη RAM η ΚΜΕ μπορεί να γράψει δεδομένα (εγγραφή) ή να διαβάσει δεδομένα (ανάγνωση). Στην εγγραφή, η ΚΜΕ μεταφέρει σε μια θέση μνήμης ένα δεδομένο. Η μνήμη δέχεται τη διεύθυνση στην οποία θα γίνει η εγγραφή και τα περιεχόμενα που θα γραφούν στη θέση αυτή.

Στην ανάγνωση, τα περιεχόμενα μιας θέσης μεταφέρονται στην ΚΜΕ. Η μνήμη δέχεται τη διεύθυνση από την οποία θα διαβαστούν δεδομένα και επιστρέφει τα περιεχόμενα της θέσης αυτής.

Μονάδες Εισόδου - Εξόδου

Με τον όρο μονάδες εισόδου αναφερόμαστε στο σύνολο των συσκευών ή διατάξεων, που επιτρέπουν τη μετατροπή πληροφοριών (κείμενο, εικόνα, ήχο, video κ.λπ.) σε ψηφιακή αναπαράσταση, ώστε να εισαχθεί στον υπολογιστή (π.χ. *πληκτρολόγιο, ποντίκι, σαρωτής*). Οι μονάδες εξόδου μετατρέπουν την πληροφορία από ψηφιακή αναπαράσταση σε κείμενο, ήχο κ.λπ (π.χ. *οθόνη, εκτυπωτής*). Οι μονάδες που χρησιμοποιούνται για την είσοδο αλλά και για την έξοδο δεδομένων ονομάζονται μονάδες εισόδου και εξόδου (π.χ. *modems, κάρτες ήχου και video*).

Βοηθητική μνήμη

Στις μονάδες βοηθητικής μνήμης αποθηκεύονται δεδομένα τα οποία θα παραμείνουν και μετά τη λήξη της λειτουργίας του. Οι πιο γνωστές μονάδες βοηθητικής μνήμης είναι τα μαγνητικά και οπτικά μέσα αποθήκευσης (σκληροί δίσκοι, δισκέτες, μαγνητικές ταινίες, οπτικοί δίσκοι).

Διάδρομοι

Ένας διάδρομος είναι μια ομάδα αγωγών που χρησιμοποιείται για την επικοινωνία μεταξύ των μονάδων του υπολογιστή.

Ο διάδρομος χωρίζεται λειτουργικά σε τρία μέρη: το *διάδρομο δεδομένων* (data bus), το *διάδρομο διευθύνσεων* (address bus) και το *διάδρομο ελέγχου* (control bus).

Μέσω του *διαδρόμου δεδομένων* μεταφέρονται τα δεδομένα που θέλουμε να γράψουμε ή να διαβάσουμε κάθε φορά (π.χ. τα δυαδικά ψηφία που συνθέτουν το περιεχόμενο μιας θέσης μνήμης, ενός καταχωρητή της ΚΜΕ, ή δεδομένα από κάποια άλλη μονάδα).

Μέσω του *διαδρόμου διευθύνσεων* μεταφέρονται δυαδικά ψηφία που σχηματίζουν τη διεύθυνση μιας θέσης μνήμης ή τη διεύθυνση μιας συσκευής εισόδου-εξόδου, δηλαδή προσδιορίζουν πού θα γραφτεί ή από πού θα διαβαστεί ένα δεδομένο.

Μέσω του *διαδρόμου ελέγχου* η ΚΜΕ πληροφορεί τη μνήμη ή τις περιφερειακές συσκευές για την ενέργεια που προτίθεται να κάνει (π.χ. να διαβάσει ή να γράψει δεδομένα).

Αξίζει να σημειωθεί ότι κάθε χρονική στιγμή μόνο δύο συσκευές μπορούν να επικοινωνούν μέσω του διαδρόμου. Αν κάποια στιγμή επικοινωνεί μέσω του διαδρόμου η ΚΜΕ με τη μνήμη, μια μονάδα εισόδου δε μπορεί να στείλει δεδομένα, αλλά πρέπει να περιμένει να ολοκληρωθεί η επικοινωνία μεταξύ της ΚΜΕ και της κύριας μνήμης.

ΕΡΩΤΗΣΕΙΣ

1. Πού χρησιμοποιείται το δυαδικό και πού το δεκαεξαδικό σύστημα αρίθμησης;
2. Πού χρησιμεύει ένα σύνολο χαρακτήρων;
3. Ποια είναι τα πιο γνωστά σύνολα χαρακτήρων;
4. Από ποια τμήματα αποτελείται ένα υπολογιστικό σύστημα;

ΑΣΚΗΣΕΙΣ

1. Εκτελέστε τις ακόλουθες μετατροπές
 - $(111)_2 = (X)_{10}$
 - $(FFA)_{16} = (X)_{10}$
 - $(154)_{10} = (X)_{16}$
 - $(22)_{10} = (X)_2$
 - $(10010010)_2 = (X)_{16}$
 - $(65F)_{16} = (X)_2$
2. Εκτελέστε τις ακόλουθες πράξεις στο δυαδικό σύστημα
 - $111 + 1001$
 - $111-11$
 - 11×10
 - $1111 : 11$
3. Εκτελέστε τις ακόλουθες πράξεις στο δεκαεξαδικό σύστημα
 - $AF9 + 11B$
 - $AA9 - 1B8$

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Κ.Ζ. Πεκμεστζή**, "Συστήματα Μικροϋπολογιστών", Εκδόσεις Συμμετρία, 1995.
2. **Γ. Παπακωνσταντίνου, Π. Τσανάκας, Ν. Κοζύρης, Α. Μανουσοπούλου, Π. Ματζάκος**, "Τεχνολογία Υπολογιστικών Συστημάτων και Λειτουργικά Συστήματα". ΥΠΕΠΘ, Παιδαγωγικό Ινστιτούτο, 1999.
3. **Mano M. Morris**, "Ψηφιακή Σχεδίαση", Παπασωτηρίου 1992, Δεύτερη Έκδοση, Μετάφραση Απ. Τραγανίτης.

Κεφάλαιο 2ο

Εφαρμογές των Ψηφιακών Ηλεκτρονικών στα Υπολογιστικά Συστήματα

Περιεχόμενα

- 2.1 Αλγεβρα Boole
- 2.2 Λογικές Πύλες
- 2.3 Απομονωτές και transceivers
- 2.4 Κωδικοποιητές και αποκωδικοποιητές
- 2.5 Πολυπλέκτες και αποπλέκτες
- 2.6 Στοιχειώδεις μονάδες άθροισης
- 2.7 Ακολουθιακές μονάδες - Στοιχεία μνήμης
- 2.8 Καταχωρητές
- 2.9 Μετρητές ή απαριθμητές

Στόχοι του κεφαλαίου:

Όταν ολοκληρώσεις το κεφάλαιο αυτό θα μπορείς να...

εξηγείς τη λειτουργία βασικών ψηφιακών κυκλωμάτων όπως:

λογικές πύλες

κωδικοποιητές και αποκωδικοποιητές

πολυπλέκτες και αποπλέκτες

καταχωρητές

μετρητές

αθροιστές και αφαιρέτες

εξηγείς τη χρήση των κυκλωμάτων αυτών στα υπολογιστικά συστήματα

εξηγείς τη δομή των τμημάτων ενός υπολογιστικού συστήματος με τη βοήθεια των ψηφιακών ηλεκτρονικών

Στο Κεφάλαιο αυτό θα αναφερθούμε στις εφαρμογές των ψηφιακών ηλεκτρονικών στα υπολογιστικά συστήματα. Με τον όρο ψηφιακά ηλεκτρονικά εννοούμε ηλεκτρονικές συσκευές που επεξεργάζονται πληροφορίες σε δυαδική μορφή (0 ή 1). Όπως αναφέραμε στο Κεφάλαιο 1, τα υπολογιστικά συστήματα επεξεργάζονται τις πληροφορίες σε δυαδική μορφή (0 ή 1), γι' αυτό και στα υπολογιστικά συστήματα χρησιμοποιούνται ψηφιακά ηλεκτρονικά κυκλώματα για την επεξεργασία των δεδομένων. Με τον όρο επεξεργασία αναφερόμαστε σε ενέργειες όπως αποθήκευση δεδομένων (προσωρινή ή μόνιμη) και η εκτέλεση πράξεων (λογικών και αριθμητικών).

Η επεξεργασία αυτή πραγματοποιείται με τη βοήθεια λογικών κυκλωμάτων (απομονωτών και λογικών πυλών), κωδικοποιητών και αποκωδικοποιητών, πολυπλεκτών και αποπλεκτών, καταχωρητών, μετρητών και μονάδων άθροισης.

Στη συνέχεια αναφέρουμε για κάθε μια από τις μονάδες αυτές τη χρησιμότητά της (το ρόλο της σε ένα υπολογιστικό σύστημα), τη λειτουργία της και το σχηματικό της διάγραμμα. Τα ψηφιακά κυκλώματα διακρίνονται, όπως γνωρίζουμε, σε δύο κατηγορίες: τα συνδυαστικά και τα ακολουθιακά. Στα συνδυαστικά κυκλώματα, η τιμή της εξόδου ή των εξόδων εξαρτάται από την τιμή των εισόδων εκείνη τη χρονική στιγμή. Στα ακολουθιακά κυκλώματα η έξοδος εξαρτάται από την τιμή των εισόδων όχι μόνο εκείνη τη χρονική στιγμή, αλλά και από την τιμή που είχαν σε προηγούμενες χρονικές στιγμές (από μια ακολουθία τιμών εισόδου). Οι λογικές πύλες, οι απομονωτές, οι αποκωδικοποιητές και οι κωδικοποιητές, οι πολυπλέκτες και οι αποπλέκτες, οι αθροιστές και οι αφαιρέτες ανήκουν στα συνδυαστικά κυκλώματα. Αντιθέτως, οι καταχωρητές και οι απαριθμητές ανήκουν στα ακολουθιακά κυκλώματα. Οι λογικές πύλες χρησιμοποιούνται για την υλοποίηση λογικών πράξεων. Με τον όρο λογικές πράξεις αναφερόμαστε σε πράξεις των οποίων οι τελεστές είναι λογικά ψηφία (δυαδικά ψηφία) και τα αποτελέσματα επίσης λογικά ψηφία. Για την υλοποίηση των κυκλωμάτων αυτών, χρησιμοποιούνται γνώσεις από την άλγεβρα Boole.

2.1 Άλγεβρα Boole

Η δίτιμη άλγεβρα Boole είναι ένα αλγεβρικό σύστημα το οποίο περιλαμβάνει δύο στοιχεία (τα οποία συμβολίζουμε συνήθως με '0' και '1') και τρεις τελεστές, τους οποίους συμβολίζουμε με + (OR, Η), · (AND, ΚΑΙ) ~ (NOT, ΟΧΙ). Συχνά, αντί για το σύμβολο ~ χρησιμοποιούμε το σύμβολο ' ($\sim x = x'$). Οι κανόνες για τους τελεστές αυτούς ορίζονται σύμφωνα με τον πίνακα 2.1.

x	y	x+y	x	y	x·y	x	
0	0	0	0	0	0	0	
0	1	1	0	1	0	1	
1	0	1	1	0	0		
1	1	1	1	1	1		

Πίνακας 2.1 Τελεστές της δίτιμης άλγεβρας Boole

Οι μεταβλητές της άλγεβρας Boole μπορούν να πάρουν δύο τιμές '0' ή '1'. Η άλγεβρα Boole χαρακτηρίζεται από μια σειρά αξιώματα, θεωρήματα και ιδιότητες (πχ. αντιμεταθετική, προσεταιριστική κ.λπ.).

Συναρτήσεις Boole

Μια συνάρτηση Boole είναι μια έκφραση που σχηματίζεται από δυαδικές μεταβλητές, τους τελεστές Η, ΚΑΙ, ΟΧΙ και παρενθέσεις. Για μια δεδομένη τιμή των μεταβλητών η τιμή της συνάρτησης μπορεί να είναι είτε '0' είτε '1'. Μια συνάρτηση Boole μπορεί να οριστεί είτε με τον τύπο της, είτε με τον πίνακα αληθείας της. Για παράδειγμα, η συνάρτηση

$$F = xyz + xy'z$$

Παίρνει την τιμή '1' είτε όταν $x=y=z=1$ είτε όταν $x=z=1$, και $y=0$ (με y' συμβολίζουμε την αντίστροφη της μεταβλητής y , η οποία είναι 0 όταν η y είναι 1 και αντίστροφα).

Ο πίνακας αληθείας της συνάρτησης F δίνεται στη συνέχεια.

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

Πίνακας 2.2 Πίνακας αληθείας της συνάρτησης $F = xyz + xy'z$

Απλοποίηση συνάρτησης Boole

Με τον όρο απλοποίηση συνάρτησης Boole εννοούμε τη διαδικασία μέσω της οποίας η έκφραση μιας συνάρτησης μετασχηματίζεται σε μια μορφή απλούστερη. Με τον όρο 'απλούστερη' αναφερόμαστε συνήθως στην υλοποίηση της συνάρτησης με λογικές πύλες. Η απλοποίηση πραγματοποιείται με τη βοήθεια του πίνακα Karnaugh όπως φαίνεται στο Σχήμα 2.1.

		yz	y'z	y'z	yz	yz'
x		00	01	11	10	
x'	0	0	0	0	0	
x	1	0	1	1	0	
			xy'z	xyz		

Σχήμα 2.1 Απλοποίηση της συνάρτησης F με το χάρτη Karnaugh

Η απλοποιημένη μορφή της συνάρτησης F είναι η $F=xz$.

2.2 Λογικές Πύλες

Μια λογική πύλη υλοποιεί μια συνάρτηση Boole και έχει μία, δύο ή περισσότερες εισόδους και μία έξοδο.

Κάθε λογική πύλη έχει ένα σύμβολο και έναν πίνακα αληθείας. Ο πίνακας αυτός δείχνει την τιμή της εξόδου για τις διαφορετικές τιμές των εισόδων. Οι λογικές πύλες χρησιμοποιούνται ακόμη στην υλοποίηση των άλλων μονάδων (αποκωδικοποιητών πολυπλεκτών, αθροιστών, κ.λπ.) που θα περιγραφούν στη συνέχεια.

Οι πιο γνωστές λογικές πράξεις είναι η σύζευξη (AND), η διάζευξη (OR) και η άρνηση (NOT). Οι πράξεις αυτές υλοποιούνται με τις λογικές πύλες AND, OR και NOT αντίστοιχα. Ακόμη, πολύ συχνά χρησιμοποιούνται οι πύλες NAND και NOR, οι οποίες εκτελούν τις αντίστροφες λογικές πράξεις από τις AND και OR αντίστοιχα, καθώς και οι πύλες της αποκλειστικής διάζευξης (XOR) και της άρνησής της (XNOR).

Πύλη NOT

Η λογική πράξη της άρνησης έχει μια μεταβλητή εισόδου και ένα αποτέλεσμα το οποίο είναι '1' αν η είσοδος είναι '0', διαφορετικά το αποτέλεσμα είναι '0'. Στο Σχήμα 2.2 παρουσιάζουμε το λογικό σύμβολο της πύλης NOT, καθώς και τον πίνακα αληθείας της.

a	x
0	1
1	0



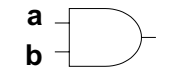
Σχήμα 2.2 Λογικό σύμβολο της πύλης NOT και πίνακας αληθείας

Συχνά, στην υλοποίηση λογικών συναρτήσεων, αντί για το σύμβολο της πύλης NOT χρησιμοποιούμε για απλότητα ένα κυκλάκι.

Πύλη AND

Η λογική πράξη της σύζευξης έχει δύο (ή περισσότερους) τελεστές και ένα αποτέλεσμα το οποίο είναι '1' αν όλα τα δεδομένα είναι '1'. Σε κάθε άλλη περίπτωση, το αποτέλεσμα είναι '0'. Η πράξη αυτή υλοποιείται με την πύλη AND. Στο Σχήμα 2.3 παρουσιάζουμε το λογικό σύμβολο της πύλης AND δύο εισόδων, καθώς και τον πίνακα αληθείας της.

a	b	x
0	0	0
0	1	0
1	0	0
1	1	1



Σχήμα 2.3 Λογικό σύμβολο της πύλης AND και πίνακας αληθείας

Πύλη OR

Η λογική πράξη της διάζευξης έχει δύο (ή περισσότερους) τελεστές και ένα αποτέλεσμα το οποίο είναι '0' αν όλα τα δεδομένα είναι '0'. Σε κάθε άλλη περίπτωση, το αποτέλεσμα είναι '1'. Η πράξη αυτή υλοποιείται με την πύλη OR. Στο Σχήμα 2.4 παρουσιάζουμε το λογικό σύμβολο της πύλης OR δύο εισόδων, καθώς και τον πίνακα αληθείας της.

a	b	x
0	0	0
0	1	1
1	0	1
1	1	1

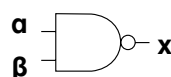


Σχήμα 2.4 Λογικό σύμβολο της πύλης OR και πίνακας αληθείας

Πύλη NAND

Η λογική πράξη της άρνησης της σύζευξης έχει δύο (ή περισσότερους) τελεστές και ένα αποτέλεσμα το οποίο είναι '0' αν όλα τα δεδομένα είναι '1'. Σε κάθε άλλη περίπτωση, το αποτέλεσμα είναι '1'. Η πράξη αυτή υλοποιείται με την πύλη NAND. Στο Σχήμα 2.5 παρουσιάζουμε το λογικό σύμβολο της πύλης NAND δύο εισόδων, καθώς και τον πίνακα αληθείας της.

a	b	x
0	0	1
0	1	1
1	0	1
1	1	0



Σχήμα 2.5 Λογικό σύμβολο της πύλης NAND και πίνακας αληθείας

Πύλη NOR

Η λογική πράξη της άρνησης της διάζευξης έχει δύο (ή περισσότερους) τελεστές και ένα αποτέλεσμα το οποίο είναι '1' αν όλα τα δεδομένα είναι '0'. Σε κάθε άλλη περίπτωση, το αποτέλεσμα είναι '0'. Στο Σχήμα 2.6 παρουσιάζουμε το λογικό σύμβολο της πύλης NOR δύο εισόδων, καθώς και τον πίνακα αληθείας της.

a	b	x
0	0	1
0	1	0
1	0	0
1	1	0



Σχήμα 2.6 Λογικό σύμβολο της πύλης NOR και πίνακας αληθείας

Πύλη XOR

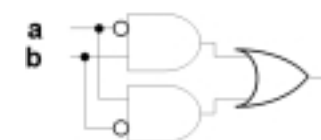
Η λογική πράξη της αποκλειστικής διάζευξης με δύο τελεστές έχει αποτέλεσμα '1' αν οι δύο τελεστές έχουν διαφορετική τιμή και '0' διαφορετικά. Η πράξη αυτή υλοποιείται με την πύλη XOR. Στο Σχήμα 2.7 παρουσιάζουμε το λογικό σύμβολο της πύλης XOR δύο εισόδων, καθώς και τον πίνακα αληθείας της.

a	b	x
0	0	0
0	1	1
1	0	1
1	1	0



Σχήμα 2.7 Λογικό σύμβολο της πύλης XOR και πίνακας αληθείας

Συνήθως, η λογική πράξη της αποκλειστικής διάζευξης συμβολίζεται με το +, δηλαδή $x=a+b$. Μπορεί κανείς να παρατηρήσει ότι η λογική συνάρτηση την οποία υλοποιεί η πύλη XOR είναι $x=a'b + ab'$. Έτσι, η πύλη XOR είναι ισοδύναμη με την ακόλουθη υλοποίηση.



Σχήμα 2.8 Πύλη XOR υλοποιημένη με πύλες AND, OR, NOT

Πύλη XNOR

Η λογική πράξη της άρνησης της αποκλειστικής διάζευξης με δύο τελεστές έχει αποτέλεσμα '0' αν οι δύο τελεστές έχουν διαφορετική τιμή και '1' διαφορετικά. Η πράξη αυτή υλοποιείται με την πύλη XNOR. Στο Σχήμα 2.9 παρουσιάζουμε το λογικό σύμβολο της πύλης XNOR δύο εισόδων, καθώς και τον πίνακα αληθείας της.

a	b	x
0	0	1
0	1	0
1	0	0
1	1	1



Σχήμα 2.9 Λογικό σύμβολο της πύλης XNOR και πίνακας αληθείας

2.3 Απομονωτές και transceivers

Όπως είδαμε στο Κεφάλαιο 1, σε ένα υπολογιστικό σύστημα, υπάρχει ένας διάδρομος, πάνω στον οποίο συνδέονται τα τμήματα του υπολογιστικού συστήματος (ΚΜΕ, μνήμη, συσκευές εισόδου-εξόδου κ.λπ.). Είδαμε επίσης ότι κάθε στιγμή μόνο δύο συσκευές μπορούν να ανταλλάσσουν δεδομένα μέσω του διαδρόμου (η μία να στέλνει δεδομένα και η άλλη να λαμβάνει).

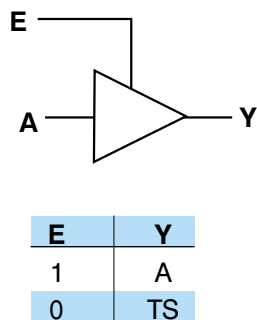
Για να μπορέσει να πραγματοποιηθεί αυτή η λειτουργία, πρέπει να υπάρχει ένας τρόπος ώστε οι έξοδοι (ή εισοδοί) μιας συσκευής ή μονάδας που είναι συνδεμένη στο διάδρομο να απομονωθούν, να λειτουργήσουν δηλαδή με τέτοιο τρόπο ώστε η συσκευή ή μονάδα να μην επηρεάζει καθόλου (και να μην επηρεάζεται από) το κύκλωμα στο οποίο συνδέεται. Για να πραγματοποιηθεί αυτό, στους ακροδέκτες κάθε μιας από τις συσκευές που είναι συνδεμένες στο διάδρομο συνδέεται ένας απομονωτής (buffer) ή transeiver.

Ένας απομονωτής επιτρέπει μεταφορά δεδομένων μόνο προς τη μία κατεύθυνση, ενώ ένας transceiver επιτρέπει μεταφορά σημάτων και προς τις δύο κατευθύνσεις, με την ενεργοποίηση κατάλληλου σήματος επιλογής.

Όταν η έξοδος ενός απομονωτή ή transceiver δεν είναι ενεργοποιημένη, δεν επηρεάζει καθόλου το κύκλωμα στο οποίο συνδέεται (λέμε ότι έχει άπειρη αντίσταση). Οι απομονωτές (buffers) χρησιμοποιούνται ακόμη ως ενισχυτές του διαδρόμου διευθύνσεων.

Απομονωτής

Ο απομονωτής έχει μία είσοδο δεδομένων (A), μια είσοδο ελέγχου (E) και μια έξοδο (Y). Όταν η είσοδος ελέγχου βρίσκεται στη λογική κατάσταση '1' (επίτρεψη), η είσοδος A μεταφέρεται στην έξοδο. Αν η είσοδος ελέγχου είναι στη λογική κατάσταση '0' (απαγόρευση), η έξοδος είναι σε κατάσταση υψηλής αντίστασης (Three-state, TS).

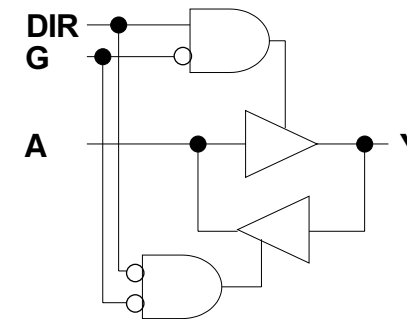


Σχήμα 2.10 Απομονωτής και πίνακας λειτουργίας

Transceiver

Για την ενίσχυση γραμμών στις οποίες είναι συνδεδεμένες μονάδες διπλής κατεύθυνσης (πχ. μονάδες εισόδου-εξόδου) χρησιμοποιούνται ειδικοί απομονωτές που ονομάζονται transceivers (transmitters/receivers).

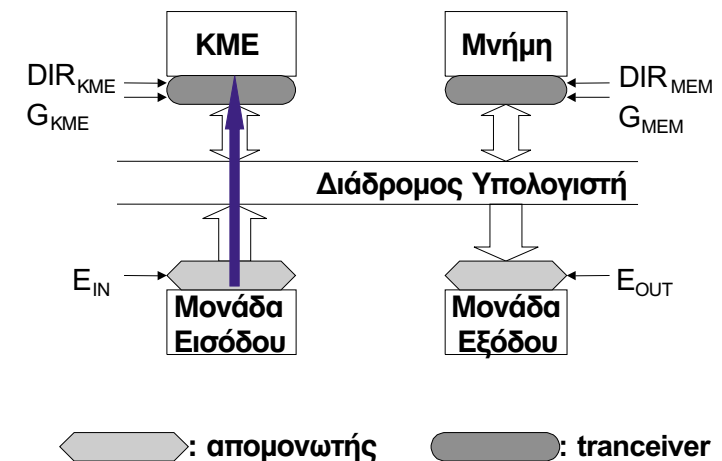
Ο transceiver διαθέτει δύο εισόδους ελέγχου G και DIR. Όταν η είσοδος G (επίτρεψης) είναι σε λογική κατάσταση '0', τότε ο transceiver λειτουργεί σε κατάσταση απομόνωσης. Όταν G=1, τότε, η μία είσοδος μεταφέρεται στην άλλη ανάλογα με την τιμή του σήματος DIR.



G	DIR	Λειτουργία
0	x	TS
1	0	A => Y
1	1	Y => A

Σχήμα 2.11 Transceiver και πίνακας λειτουργίας

Για να φανεί η χρησιμότητα των απομονωτών και των transceivers στα υπολογιστικά συστήματα, ας θεωρήσουμε το ακόλουθο απλοποιημένο διάγραμμα ενός υπολογιστικού συστήματος, στο οποίο οι μονάδες συνδέονται μέσω ενός διαδρόμου.



Σχήμα 2.12 Υπολογιστικό σύστημα με απομονωτές και transceivers

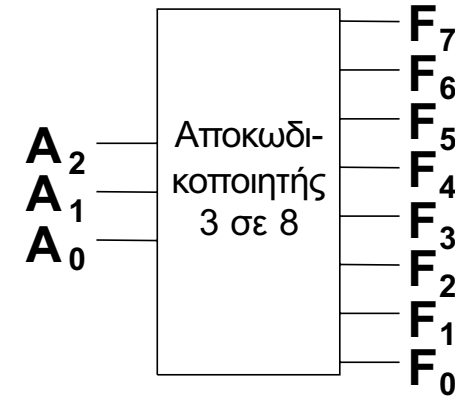
Σε αυτό το υπολογιστικό σύστημα προκειμένου να αποσταλούν δεδομένα από τη μονάδα εισόδου στην ΚΜΕ, πρέπει τα σήματα να πάρουν τις τιμές που φαίνονται στον πίνακα 2.3.

Σήμα	Τιμή	Λειτουργία
DIR _{ΚΜΕ}	0	Είσοδος δεδομένων στην ΚΜΕ
G _{ΚΜΕ}	1	Ενεργοποίηση transceiver
DIR _{ΜΕΜ}	X	Δεν ενδιαφέρει η κατεύθυνση
G _{ΜΕΜ}	0	Απενεργοποίηση μνήμης
E _{IN}	1	Ενεργοποίηση μονάδας εισόδου
E _{OUT}	0	Απενεργοποίηση μονάδας εξόδου

Πίνακας 2.3 Λειτουργία των transceivers και απομονωτών για μεταφορά δεδομένων από τη μονάδα εισόδου στην ΚΜΕ.

Με τις τιμές των σημάτων που φαίνονται στον Πίνακα 2.3, μόνο η μονάδα εισόδου μπορεί να στέλνει σήματα, και μόνο η ΚΜΕ μπορεί να λαμβάνει τα σήματα αυτά.

Σχήμα 2.13 φαίνεται το λογικό διάγραμμα και ο πίνακας αληθείας του αποκωδικοποιητή 3-σε-8.



A ₂	A ₁	A ₀	F ₇	F ₆	F ₅	F ₄	F ₃	F ₂	F ₁	F ₀
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Σχήμα 2.13 Αποκωδικοποιητής 3-σε-8 και πίνακας αληθείας

Είναι εύκολο να εξάγει κανείς τη συνάρτηση Boole για κάθε μια από τις εξόδους F₀ έως F₇ του αποκωδικοποιητή. Στο Σχήμα 2.14 παρουσιάζουμε πώς μπορούμε να υλοποιήσουμε έναν αποκωδικοποιητή χρησιμοποιώντας πύλες AND και αντιστροφείς. Στο σχήμα αυτό, με κυκλάκι συμβολίζουμε την αντιστροφή (πύλη NOT). Ακόμη, με A' συμβολίζουμε την αντίστροφη τιμή της μεταβλητής A.

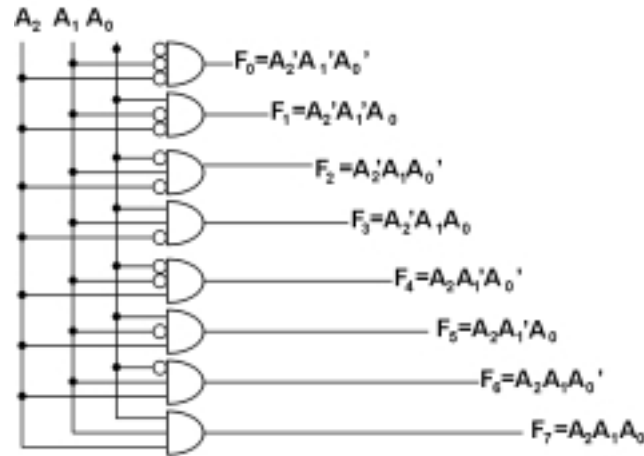
2.4 Κωδικοποιητές και αποκωδικοποιητές

Όταν περισσότερες από δύο συσκευές είναι συνδεδεμένες σε ένα διάδρομο (όπως συμβαίνει στα υπολογιστικά συστήματα), πρέπει να αποφασιστεί ποια από όλες θα επιλεγεί ώστε να πραγματοποιήσει μεταφορά των δεδομένων. Αυτό επιτυγχάνεται με την ενεργοποίηση κατάλληλου σήματος. Αν ο αριθμός των συσκευών που είναι συνδεδεμένες στο διάδρομο είναι μεγάλος, τότε μπορούμε να αποδώσουμε ένα δυαδικό αριθμό σε κάθε συσκευή που θα αποτελεί τη διεύθυνσή της. Έτσι με λίγες γραμμές (πχ. 3) μπορούμε να ελέγξουμε την επιλογή μεγάλου αριθμού συσκευών (πχ. 8). Προκύπτει όμως η ανάγκη κυκλωμάτων τα οποία θα μετατρέπουν την πληροφορία της επιλογής από τις λίγες γραμμές στις πολλές (μια για κάθε συσκευή). Τα κυκλώματα αυτά ονομάζονται κυκλώματα αποκωδικοποίησης ή αποκωδικοποιητές.

2.4.1 Αποκωδικοποιητές

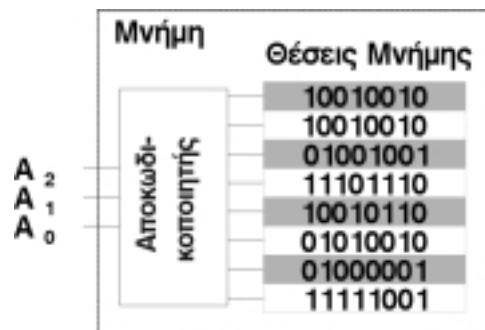
Ένας αποκωδικοποιητής είναι ένα κύκλωμα (μονάδα) που έχει n γραμμές εισόδου και 2ⁿ γραμμές εξόδου. Κάθε στιγμή μόνο μια από τις 2ⁿ γραμμές εξόδου είναι ενεργή (έχει την τιμή '1'), ανάλογα με το συνδυασμό των τιμών εισόδου. Για παράδειγμα, στο

A ₂	A ₁	A ₀	F ₀	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇
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



Σχήμα 2.14 Υλοποίηση αποκωδικοποιητή 3-σε-8 με πύλες AND και NOT

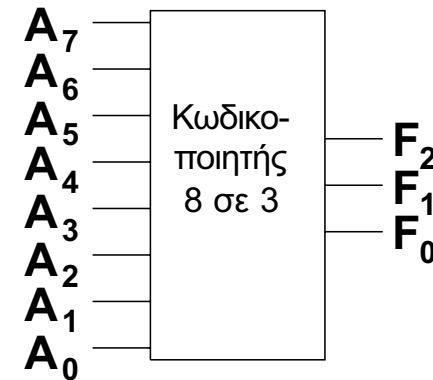
Αποκωδικοποιητές χρησιμοποιούνται στην επιλογή μιας θέσης μνήμης. Όπως γνωρίζουμε, η μνήμη αποτελείται από θέσεις, κάθε μια από τις οποίες έχει μια διεύθυνση. Γνωρίζουμε ακόμη, ότι η επιλογή της θέσης μνήμης γίνεται μέσω του διαδρόμου διευθύνσεων. Αν ο διάδρομος διευθύνσεων αποτελείται από n καλώδια, μπορεί να διεθυσιοδοτήσει μέχρι 2ⁿ θέσεις μνήμης. Η επιλογή της επιθυμητής θέσης μνήμης γίνεται μέσα στη μονάδα μνήμης, χρησιμοποιώντας έναν αποκωδικοποιητή.



Σχήμα 2.15 Μνήμη 8 θέσεων που διεθυσιοδοτείται από 3 γραμμές διευθύνσεων χρησιμοποιώντας αποκωδικοποιητή

2.4.2 Κωδικοποιητές

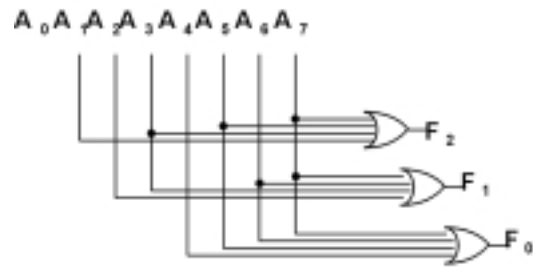
Ένας κωδικοποιητής είναι ένα λογικό κύκλωμα με 2ⁿ εισόδους και n εξόδους. Για να λειτουργήσει ο κωδικοποιητής, κάθε στιγμή μόνο μια από τις εισόδους μπορεί να είναι ενεργοποιημένη. Η λογική τιμή της εξόδου εξαρτάται από το συνδυασμό των εισόδων. Στο Σχήμα 2.16 φαίνεται το λογικό διάγραμμα και ο πίνακας αληθείας ενός κωδικοποιητή 8-σε-3.



A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	F ₂	F ₁	F ₀
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Σχήμα 2.16 Κωδικοποιητής 8-σε-3 και πίνακας αληθείας

Στο Σχήμα 2.17 παρουσιάζουμε την υλοποίηση ενός κωδικοποιητή 8-σε-3 χρησιμοποιώντας πύλες OR.



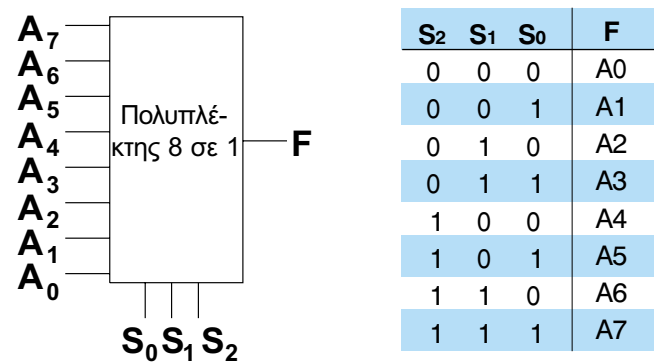
Σχήμα 2.17 Υλοποίηση κωδικοποιητή 8-σε-3 με πύλες OR

2.5 Πολυπλέκτες και αποπλέκτες

Ο όρος 'πολυπλεξία' σημαίνει τη μεταβίβαση ενός μεγάλου αριθμού πληροφοριών μέσα από ένα μικρότερο αριθμό καναλιών ή γραμμών.

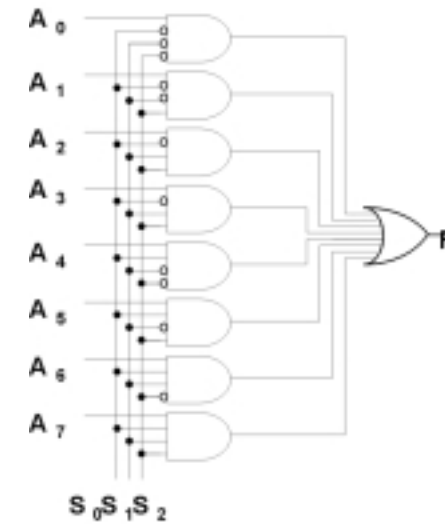
2.5.1 Πολυπλέκτες

Ένας ψηφιακός πολυπλέκτης είναι ένα συνδυαστικό κύκλωμα που επιλέγει δυαδικές πληροφορίες ανάμεσα σε πολλές γραμμές εισόδου και τις κατευθύνει σε μια μοναδική γραμμή εξόδου. Η επιλογή της γραμμής εισόδου γίνεται μέσω γραμμών επιλογής. Υπάρχουν 2^n γραμμές εισόδου και n γραμμές επιλογής. Η τιμή των γραμμών επιλογής καθορίζει ποια είσοδος επιλέγεται.



Σχήμα 2.18 Πολυπλέκτης 8 εισόδων και πίνακας αληθείας

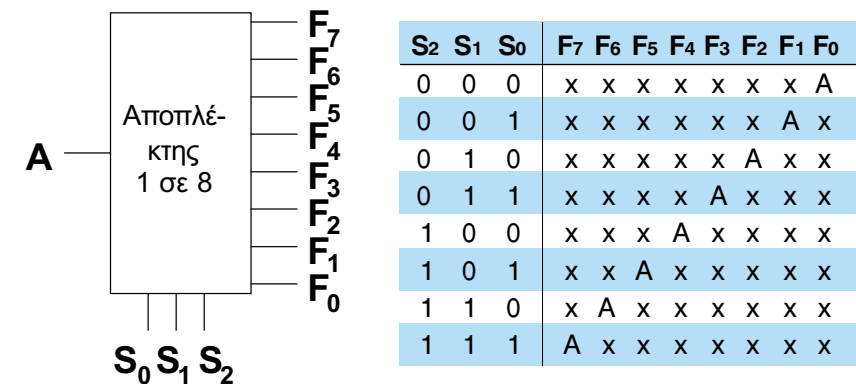
Στο Σχήμα 2.19 παρουσιάζουμε την υλοποίηση ενός πολυπλέκτη 8-σε-1 χρησιμοποιώντας πύλες AND, NOT και μια πύλη OR. Με κυκλάκι συμβολίζουμε την αντιστροφή (πύλη NOT).



Σχήμα 2.19 Υλοποίηση πολυπλέκτη 8-σε-1 με πύλες AND, NOT, OR

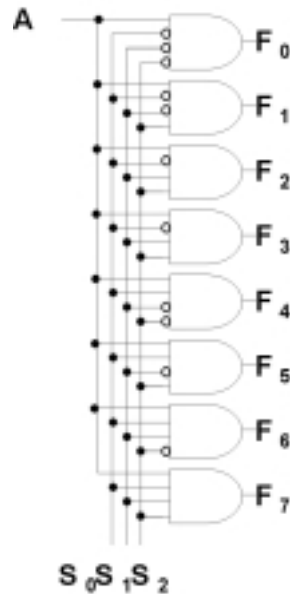
2.5.2 Αποπλέκτες

Η απόπλεξη είναι η αντίστροφη διαδικασία από την πολύπλεξη. Στην απόπλεξη υπάρχει μια είσοδος και 2^n εξόδοι. Η τιμή της εισόδου μεταφέρεται σε κάποια από τις εξόδους ανάλογα με την τιμή των n σημάτων επιλογής. Για παράδειγμα, στο Σχήμα 2.20 παρουσιάζουμε το λογικό διάγραμμα και τον πίνακα αληθείας για τον αποπλέκτη $2^3=8$ εξόδων.



Σχήμα 2.20 Αποπλέκτης 8 εξόδων και πίνακας αληθείας

Στο Σχήμα 2.21 φαίνεται η υλοποίηση ενός αποπλέκτη 8 εξόδων χρησιμοποιώντας πύλες AND και NOT.



Σχήμα 2.21 Υλοποίηση αποπλέκτη 8 εξόδων με πύλες AND και NOT

2.6 Στοιχειώδεις μονάδες άθροισης

Όπως αναφέραμε, μια από τις πιο σημαντικές λειτουργίες που εκτελούν τα υπολογιστικά συστήματα είναι οι αριθμητικές πράξεις. Οι πιο σημαντικές πράξεις, πάνω στις οποίες στηρίζονται οι υπόλοιπες, είναι η πρόσθεση και η αφαίρεση. Για την εκτέλεση των πράξεων αυτών χρησιμοποιούνται οι αθροιστές και οι αφαιρέτες.

2.6.1 Αθροιστές

Η πιο βασική αριθμητική πράξη είναι η πρόσθεση δύο δυαδικών ψηφίων. Όπως αναφέραμε, για την πρόσθεση δύο δυαδικών ψηφίων υπάρχουν τέσσερις δυνατές περιπτώσεις: $0+0=0$, $0+1=1$, $1+0=1$, $1+1=10$. Οι τρεις πρώτες πράξεις δημιουργούν ένα άθροισμα που το μήκος του είναι ένα ψηφίο. Όταν και οι δύο προσθετέοι είναι 1, το δυαδικό άθροισμα αποτελείται από δύο ψηφία. Το πιο σημαντικό από αυτά τα δύο ψηφία ονομάζεται 'κρατούμενο'. Όταν οι δύο αριθμοί

περιέχουν και άλλα δυαδικά ψηφία, το κρατούμενο που βγαίνει από την πρόσθεση δύο ψηφίων προστίθεται στο επόμενο μεγαλύτερης σημαντικότητας ζευγάρι δυαδικών ψηφίων. Ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση δυο δυαδικών ψηφίων λέγεται 'ημιαθροιστής'. Ένα συνδυαστικό κύκλωμα που εκτελεί την πρόσθεση τριών δυαδικών ψηφίων (δύο δυαδικών ψηφίων και ενός προηγούμενου κρατούμενου) λέγεται 'πλήρης αθροιστής'. Το όνομα του ημιαθροιστή προέρχεται από το γεγονός ότι δυο ημιαθροιστές μπορούν να χρησιμοποιηθούν για να υλοποιήσουν έναν πλήρη αθροιστή, όπως θα δούμε στη συνέχεια.

Ημιαθροιστής

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

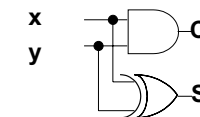
x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Πίνακας 2.4 Πίνακας αληθείας ημιαθροιστή

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

$$C = xy$$

Όπως έχουμε δει, η συνάρτηση $x'y + xy'$ είναι η πύλη XOR. Το λογικό διάγραμμα της υλοποίησής τους χρησιμοποιώντας πύλες AND και XOR φαίνεται στο Σχήμα 2.22:



Σχήμα 2.22 Ημιαθροιστής

Πλήρης Αθροιστής

Ο πλήρης αθροιστής είναι ένα συνδυαστικό κύκλωμα που σχηματίζει το άθροισμα τριών δυαδικών ψηφίων εισόδου. Έχει τρεις εισόδους και δύο εξόδους. Οι δύο από τις μεταβλητές εισόδου παριστάνουν τα δύο σημαντικά ψηφία που προστίθενται. Η τρίτη είσοδος παριστάνει το κρατούμενο από τη λιγότερο σημαντική βαθμίδα. Ο πίνακας αληθείας του πλήρη αθροιστή δίνεται στον Πίνακα 2.5:

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

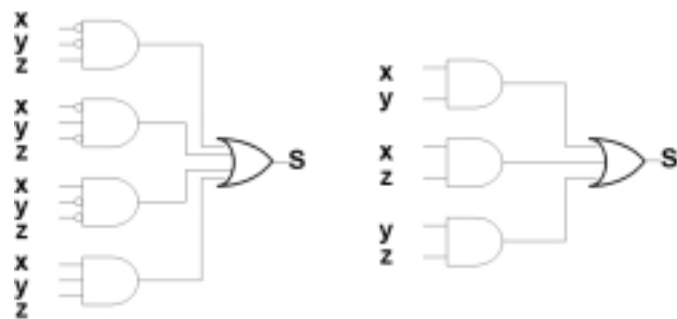
Πίνακας 2.5 Πίνακας αληθείας πλήρους αθροιστή

Μπορούμε, χρησιμοποιώντας τον πίνακα αληθείας, να εξάγουμε τις ακόλουθες απλοποιημένες μορφές για τις συναρτήσεις S και C του πλήρη αθροιστή:

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

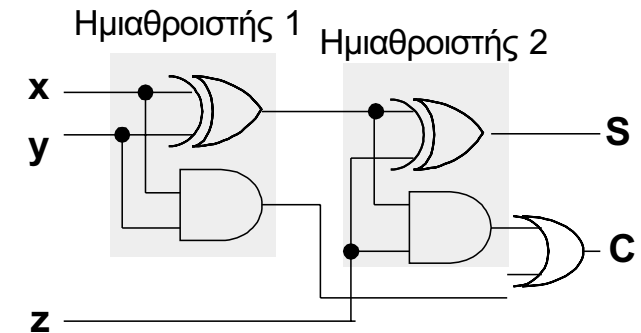
$$C = xy + xz + yz$$

Έτσι, μπορούμε να δώσουμε την υλοποίηση που φαίνεται στο Σχήμα 2.23 για το κύκλωμα του πλήρη αθροιστή.



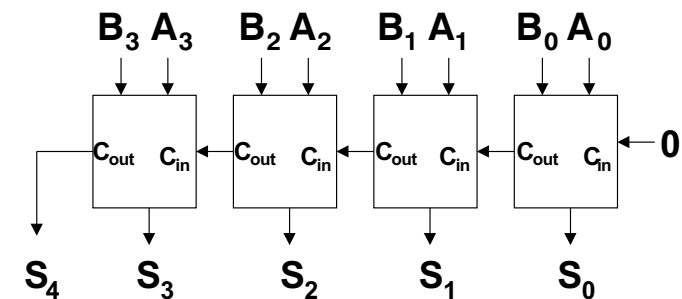
Σχήμα 2.23 Πλήρης αθροιστής

Είναι αρκετά ενδιαφέρον να παρατηρήσει κανείς ότι ένα ισοδύναμο σχήμα για τον πλήρη αθροιστή μπορεί να δοθεί με τη βοήθεια ημιαθροιστών, όπως φαίνεται στο Σχήμα 2.24.



Σχήμα 2.24 Πλήρης αθροιστής υλοποιημένος με ημιαθροιστές

Προκειμένου να προσθέσουμε δυαδικούς αριθμούς που αποτελούνται από περισσότερα του ενός ψηφία, χρησιμοποιούμε περισσότερους από έναν πλήρεις αθροιστές. Κάθε πλήρης αθροιστής αντιστοιχεί και υπολογίζει ένα ψηφίο αθροίσματος. Το κρατούμενο εξόδου κάθε αθροιστή συνδέεται στο κρατούμενο εισόδου του αθροιστή που αντιστοιχεί στην επόμενη βαθμίδα. Για παράδειγμα, στο Σχήμα 2.25 παρουσιάζεται ένα τέτοιος αθροιστής τεσσάρων βαθμίδων. Ο αθροιστής αυτός προσθέτει τους τετραψηφίους δυαδικούς αριθμούς $A_3A_2A_1A_0$ και $B_3B_2B_1B_0$ και δίνει ως αποτέλεσμα τον πενταψηφίο δυαδικό αριθμό $S_4S_3S_2S_1S_0$.



Σχήμα 2.25 Αθροιστής τεσσάρων (4) δυαδικών ψηφίων

2.6.2 Αφαιρέτες

Η αφαίρεση δύο δυαδικών αριθμών πραγματοποιείται παίρνοντας το συμπλήρωμα ως προς 2 του αφαιρέτη και προσθέτοντάς το στον αφαιρετέο. Με αυτήν την παρατήρηση, η πράξη της αφαίρεσης μετατρέπεται σε πρόσθεση και για την υλοποίησή της μπορούν να χρησιμοποιηθούν πλήρεις αθροιστές. Εναλλακτικά, μπορούμε να υλοποιήσουμε την αφαίρεση με άμεσο τρόπο (όπως την κάνουμε με χαρτί και μολύβι). Με αυτή τη μέθοδο, κάθε δυαδικό ψηφίο του αφαιρέτη αφαιρείται από το αντίστοιχης σημαντικότητας δυαδικό ψηφίο του αφαιρετέου και δίνει ένα δυαδικό ψηφίο διαφοράς. Αν το δυαδικό ψηφίο του αφαιρέτη είναι μεγαλύτερο από το ψηφίο του αφαιρετέου, δανειζόμαστε 1 από την επόμενη σημαντική θέση. Για να μεταβιβαστεί το γεγονός αυτό στην επόμενη βαθμίδα χρησιμοποιείται ένα σήμα δανεισμού (borrow). Όπως ακριβώς υπάρχουν ημιαθροιστές και πλήρεις αθροιστές, υπάρχουν ημιαφαιρέτες και πλήρεις αφαιρέτες.

Ημιαφαιρέτης

Ο ημιαφαιρέτης είναι ένα συνδυαστικό κύκλωμα το οποίο αφαιρεί δύο δυαδικά ψηφία και δίνει τη διαφορά τους. Έχει επίσης μία έξοδο που καθορίζει αν χρειάζεται να δανειστούμε (borrow) μια μονάδα. Αν συμβολίσουμε το ψηφίο του αφαιρετέου με x και του αφαιρέτη με y , για να κάνουμε την αφαίρεση $x-y$ πρέπει να ελέγξουμε τα σχετικά μεγέθη των x και y . Αν $x > y$ τότε υπάρχουν τρεις περιπτώσεις: $0-0=0$, $1-0=1$ $1-1=0$. Το αποτέλεσμα λέγεται δυαδικό ψηφίο διαφοράς. Αν $x < y$, τότε έχουμε $0-1$, και έτσι χρειάζεται να δανειστούμε μια μονάδα από την επόμενη βαθμίδα. Το 1 που δανειζόμαστε από την επόμενη θέση έχει διπλάσιο βάρος και προσθέτει 2 στο δυαδικό ψηφίο του αφαιρέτη, όπως στο δεκαδικό σύστημα το κρατούμενο της αφαίρεσης προσθέτει δέκα στον αφαιρετέο. Έτσι, με τον αφαιρετέο ίσο με 2, η διαφορά γίνεται $2-1=1$. Ο ημιαφαιρέτης χρειάζεται δύο εξόδους. Η μια έξοδος παράγει τη διάφορα και θα συμβολίζεται με D (Difference, διαφορά). Η δεύτερη έξοδος που θα συμβολίζεται με B (Borrow, δανεικό) παράγει το δυαδικό σήμα που πληροφορεί την επόμενη βαθμίδα αν δανειστήκαμε μια μονάδα ή όχι. Ο πίνακας αληθείας του ημιαφαιρέτη είναι ο ακόλουθος:

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

Πίνακας 2.6 Πίνακας αληθείας ημιαφαιρέτη

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

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

$$B = xy'$$

Πλήρης αφαιρέτης

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

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

Πίνακας 2.7 Πίνακας αληθείας πλήρους αφαιρέτη

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

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

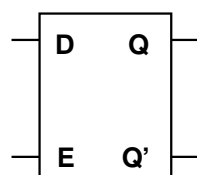
$$B = x'y + x'z + yz$$

2.7 Ακολουθιακές μονάδες - Στοιχεία Μνήμης

Οι μονάδες στις οποίες αναφερθήκαμε μέχρι το σημείο αυτό ήταν συνδυαστικές, δηλαδή η τιμή της εξόδου σε κάποια χρονική στιγμή εξαρτάται από την τιμή της εισόδου εκείνης της χρονικής στιγμής. Στη συνέχεια θα αναφερθούμε σε ακολουθιακές μονάδες. Στις ακολουθιακές μονάδες η τιμή της εξόδου εξαρτάται από την τιμή των εισόδων όχι μόνο εκείνης της χρονικής στιγμής, αλλά και από τις τιμές τους τις προηγούμενες χρονικές στιγμές. Κύριο χαρακτηριστικό των ακολουθιακών μονάδων είναι το ότι χρησιμοποιούμε στοιχεία μνήμης. Υπάρχουν δύο είδη στοιχείων μνήμης: ο *μανδαλωτής* (latch) και το *flip-flop*.

Μανδαλωτής

Ένας μανδαλωτής είναι μια ψηφιακή συσκευή στην έξοδο της οποίας μπορούμε να αποθηκεύσουμε ένα '1' ή ένα '0'. Ο μανδαλωτής έχει μια είσοδο δεδομένων (D, data) και μια είσοδο επίτρεψης (E, Enable). Έχει δύο εξόδους, τις οποίες συμβολίζουμε με Q και Q'. Η τιμές των εξόδων Q και Q' είναι συμπληρωματικές, με άλλα λόγια όταν η έξοδος Q έχει την τιμή '1', η έξοδος Q' έχει την τιμή '0' και αντίστροφα. Οι λόγοι που στο μανδαλωτή περιλαμβάνεται και η αντιστροφή της εξόδου είναι ότι η ύπαρξη της αντεστραμμένης εξόδου είναι πολύ χρήσιμη στην πράξη, καθώς και το ότι η υλοποίησή της είναι πολύ απλή. Το σχηματικό διάγραμμα και ο πίνακας αληθείας του μανδαλωτή φαίνονται στο Σχήμα 2.26.



D	E	Q
X	0	Q _n
0	1	0
1	1	1

Σχήμα 2.26 Λογικό διάγραμμα και πίνακας αληθείας μανδαλωτή

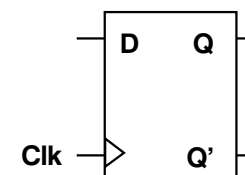
Ο μανδαλωτής λειτουργεί ως εξής: Αν η είσοδος επίτρεψης είναι ενεργοποιημένη ('1'), τότε η τιμή της εισόδου δεδομένων (D) μεταφέρεται στην έξοδο (Q). Διαφορετικά, η τιμή της εξόδου παραμένει ίδια με την προηγούμενη τιμή της (την οποία συμβολίζουμε με Q_n). Με άλλα λόγια ο μανδαλωτής φυλάσσει (μανδαλώνει) μια τιμή στην έξοδό του όσο E=0.

Flip-flop

Το flip-flop είναι μια συσκευή η οποία λειτουργεί παρόμοια με το μανδαλωτή μεταφέροντας την τιμή της εισόδου στην έξοδο. Η διαφορά του flip-flop είναι πως η μεταφορά γίνεται τη στιγμή που το σήμα επίτρεψης (που στο flip-flop λέγεται ρολόι και συμβολίζεται με clk) αλλάζει τιμή από 0 σε 1¹. Με άλλα λόγια το flip-flop λειτουργεί στην ακμή του ρολογιού.

¹ Τα flip-flops που λειτουργούν όταν η είσοδος χρονισμού μεταβάλλεται από '0' σε '1', ονομάζονται flip-flops θετικής ακμής. Υπάρχουν ακόμη flip-flops τα οποία λειτουργούν όταν η είσοδος χρονισμού μεταβάλλεται από '1' σε '0'. Τα flip-flops αυτά ονομάζονται flip-flops αρνητικής ακμής. Τα flip-flops που θα χρησιμοποιήσουμε στη συνέχεια, είναι θετικής ακμής.

Στο Σχήμα 2.27 φαίνεται το σχηματικό διάγραμμα του flip-flop (που μοιάζει αρκετά με εκείνο του μανδαλωτή) και ο πίνακας λειτουργίας του.

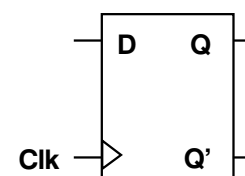


D	Clk	Q
x	0	Q _n
x	1	Q _n
0	⌋▶	0
1	⌋▶	1

Σχήμα 2.27 Flip-flop τύπου D θετικής ακμής και πίνακας λειτουργίας

Στον πίνακα λειτουργίας του flip-flop φαίνεται ότι όταν η τιμή του σήματος clk έχει οποιαδήποτε τιμή (0 ή 1), η έξοδος του flip-flop δεν επηρεάζεται (παραμένει στην προηγούμενη κατάσταση Q_n). Αντίθετα, όταν το σήμα clk μεταβαίνει από την κατάσταση '0' στην '1' (συμβολίζουμε αυτή τη μετάβαση με το βελάκι που φαίνεται στον πίνακα), η τιμή της εισόδου D μεταφέρεται στην έξοδο Q.

Ένας εναλλακτικός (και πιο απλός) τρόπος παρουσίασης της λειτουργίας του flip-flop, είναι ο χαρακτηριστικός πίνακας. Ο πίνακας αυτός δίνει την τιμή της εξόδου ως συνάρτηση της μεταβλητής εισόδου, αφού συμφωνήσουμε ότι οι αλλαγές στην έξοδο γίνονται μόνο στη μετάβαση του παλμού χρονισμού από το 0 στο 1. Έτσι, ο χαρακτηριστικός πίνακας του flip-flop τύπου D είναι ο ακόλουθος.



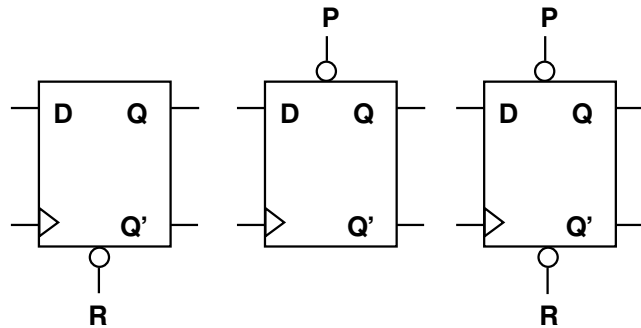
D	Q
0	0
1	1

Σχήμα 2.28 Flip-flop τύπου D θετικής ακμής και χαρακτηριστικός πίνακας

Ασύγχρονες εισόδους

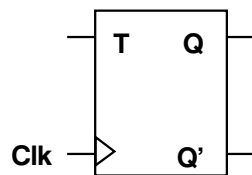
Ένα flip-flop μπορεί να περιλαμβάνει ασύγχρονες εισόδους θέσης (Preset, P) και μηδένισης (Reset, R). Πιο συγκεκριμένα, μπορεί να περιλαμβάνει καμία από τις εισόδους αυτές (όπως τα flip-flops που είδαμε μέχρι τώρα), οποιαδήποτε από τις δύο ή και τις δύο.

Η λειτουργία ασύγχρονων εισόδων είναι η εξής: Όταν η τιμή τους είναι 1, δεν επηρεάζεται η λειτουργία του flip-flop. Όταν η είσοδος P είναι 0, τότε η τιμή της εξόδου Q γίνεται 1, ανεξάρτητα από την τιμή των εισόδων D και clk. Όταν η είσοδος R είναι 0, τότε η τιμή της εξόδου γίνεται 0, ανεξάρτητα από την τιμή της εισόδου χρονισμού (clk). Οι ασύγχρονες εισόδους P και R δε μπορούν να είναι ταυτόχρονα 0, διότι, στην περίπτωση αυτή, η έξοδος του flip-flop θα είναι απροσδιόριστη. Το Σχήμα 2.29 δείχνει τα σύμβολα του flip-flop τύπου D με ασύγχρονη είσοδο P, ασύγχρονη είσοδο R και με τις δύο ασύγχρονες εισόδους.



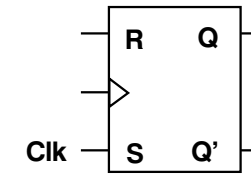
Σχήμα 2.29 Flip-flop τύπου D με (α) ασύγχρονη είσοδο μηδένισης (β) ασύγχρονη είσοδο θέσης και (γ) και τις δύο ασύγχρονες εισόδους

Το flip-flop στο οποίο αναφερθήκαμε ονομάζεται flip-flop τύπου D. Το όνομα αυτό προέρχεται από την αγγλική λέξη data (δεδομένα). Η λειτουργία του είναι να μεταφέρει τα δεδομένα της εισόδου στην έξοδο. Εκτός από τον τύπο αυτό, υπάρχουν και άλλοι τύποι flip-flop, όπως το flip-flop reset-set (RS flip-flop), Toggle (T flip-flop), J-K flip-flop. Η λειτουργία αυτών των flip-flops φαίνεται στα Σχήματα 2.30 και 2.31.



T	Q
0	Q _n
1	Q _n

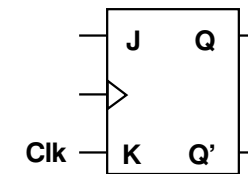
Σχήμα 2.30 Flip-flop τύπου T θετικής ακμής και χαρακτηριστικός πίνακας



R	S	Q
0	0	Q _n
0	1	0
1	0	1
1	1	απρόβλεπτη

Σχήμα 2.31 Flip-flop τύπου RS θετικής ακμής και χαρακτηριστικός πίνακας

Στο flip-flop τύπου RS, οι εισόδους R και S δε μπορούν να είναι ταυτόχρονα στο 1, γιατί στην περίπτωση αυτή η έξοδος του flip-flop είναι απρόβλεπτη (αυτό οφείλεται στην κατασκευή του flip-flop, με την οποία δε θα ασχοληθούμε εδώ). Λέμε ότι ο συνδυασμός εισόδων R=S=1 δεν είναι επιτρεπτός, και όταν σχεδιάζουμε ψηφιακά συστήματα προσπαθούμε να διασφαλίσουμε ότι ο συνδυασμός αυτός δεν εμφανίζεται στην είσοδο του flip-flop.



J	K	Q
0	0	Q _n
0	1	0
1	0	1
1	1	Q _n

Σχήμα 2.32 Flip-flop τύπου JK θετικής ακμής και χαρακτηριστικός πίνακας

Τα flip-flops των τύπων T, RS, JK μπορούν να επεκταθούν με ασύγχρονες εισόδους P και R, όπως και τα flip-flops τύπου D. Η λειτουργία των ασύγχρονων εισόδων είναι ίδια με εκείνη που έχουν στα flip-flops τύπου D.

Στη συνέχεια, θα περιγράψουμε τη λειτουργία απλών ακολουθιακών μονάδων χρησιμοποιώντας flip-flops τύπου D. Οι ακολουθιακές μονάδες στις οποίες θα αναφερθούμε είναι οι καταχωρητές και οι μετρητές.

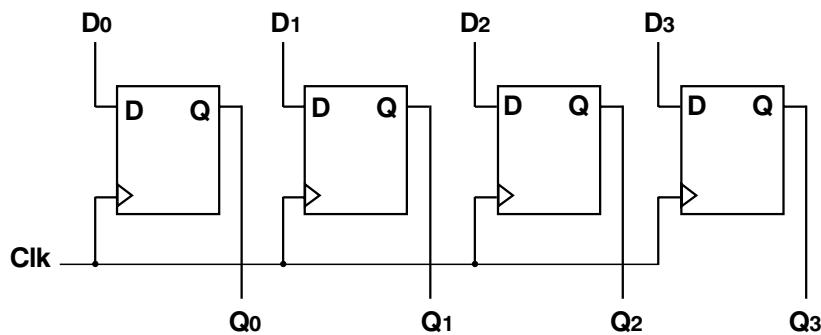
2.8 Καταχωρητές

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

2.8.1 Παράλληλοι Καταχωρητές

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

Στο Σχήμα 2.33 φαίνεται το σχηματικό διάγραμμα ενός παράλληλου καταχωρητή 4 βαθμίδων. Συνήθως αναφερόμαστε σε ένα τέτοιο καταχωρητή με τον όρο 4-μπιτο (4-bit), επειδή μπορεί να αποθηκεύσει πληροφορία τεσσάρων δυαδικών ψηφίων.



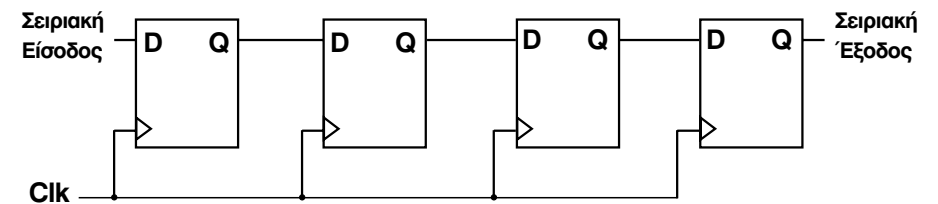
Σχήμα 2.33 Παράλληλος καταχωρητής 4 βαθμίδων

2.8.2 Καταχωρητές Ολίσθησης

Ένας καταχωρητής που μπορεί να 'ολισθαίνει' τις πληροφορίες που περιέχει προς τη μια ή προς την άλλη κατεύθυνση ονομάζεται 'καταχωρητής ολίσθησης' (shift register). Ένας τέτοιος καταχωρητής αποτελείται από μια αλυσίδα από flip-flops συνδεδεμένα στη σειρά, στα οποία η έξοδος του ενός τροφοδοτεί την είσοδο του γειτονικού του.

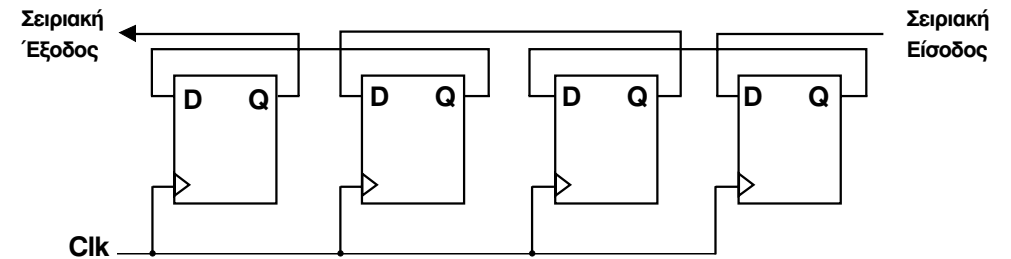
Όλα τα flip-flops παίρνουν ένα κοινό ρολόι, η ενεργοποίηση του οποίου προκαλεί την ολίσθηση από τη μια βαθμίδα στην επόμενη.

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



Σχήμα 2.34 Καταχωρητής δεξιάς ολίσθησης τεσσάρων βαθμίδων

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



Σχήμα 2.35 Καταχωρητής αριστερής ολίσθησης τεσσάρων βαθμίδων

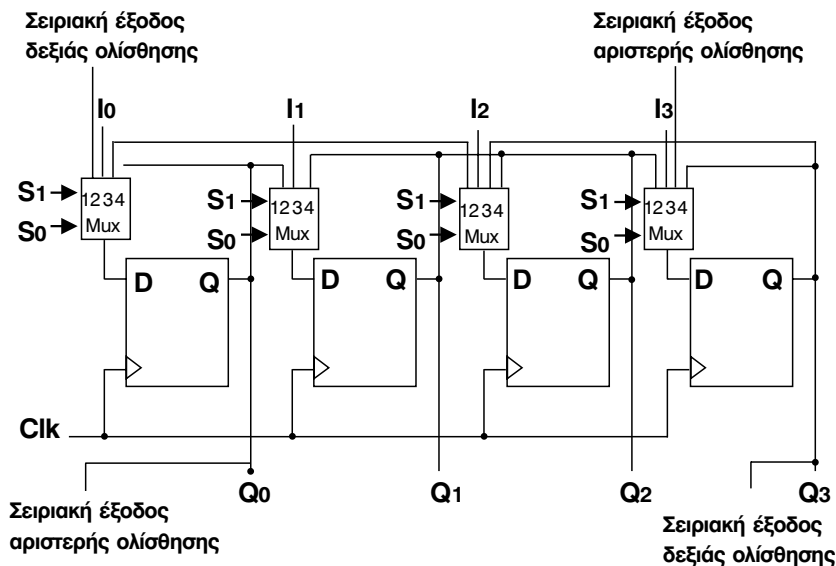
2.8.3 Καταχωρητές ολίσθησης με παράλληλη φόρτωση

Εάν έχουμε στη διάθεσή μας τις εξόδους όλων των flip-flops ενός καταχωρητή ολίσθησης ώστε να μπορούμε να κάνουμε τις κατάλληλες συνδέσεις, τότε τις πληροφορίες που εισάγουμε σειριακά στον καταχωρητή μπορούμε να τις πάρουμε παράλληλα από τις εξόδους των flip-flops. Αν προσθέσουμε στον καταχωρητή ολίσθησης και τη δυνατότητα παράλληλης φόρτισης, τότε μπορούμε επιπλέον να παίρνουμε σειριακά από την έξοδο τα δεδομένα εκείνα τα οποία βάλαμε παράλληλα στην είσοδο, ολισθαίνοντάς τα μέσα από τον καταχωρητή. Έτσι, οι καταχωρητές ολίσθησης μπορούν να χρησιμοποιηθούν για τη μετατροπή σειριακών δεδομένων σε παράλληλα και αντίστροφα. Ένας καταχωρητής ολίσθησης με παράλληλη φόρτωση έχει τις ακόλουθες δυνατότητες:

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

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

Ένας καταχωρητής που έχει τις δυνατότητες αυτές φαίνεται στο Σχήμα 2.36. Αποτελείται από 4 flip-flops, η είσοδος δεδομένων (D) καθενός τροφοδοτείται από την έξοδο ενός πολυπλέκτη (multiplexer, MUX). Οι πολυπλέκτες έχουν δύο κοινές μεταβλητές επιλογής, s_1 και s_0 , οι οποίες ρυθμίζουν τη λειτουργία του καταχωρητή.



Σχήμα 2.36 Αμφίδρομος καταχωρητής ολίσθησης με παράλληλη φόρτωση 4 ψηφίων

Ο επόμενος Πίνακας δείχνει τη λειτουργία του καταχωρητή για τις διαφορετικές τιμές των s_1 και s_0 .

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

Πίνακας 2.8 Λειτουργία καταχωρητή ολίσθησης με παράλληλη φόρτωση

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

Χρήση καταχωρητών ολίσθησης

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

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

Για παράδειγμα, ας θεωρήσουμε ότι το υπολογιστικό σύστημα θέλει να αποστείλει δεδομένα (πχ. 1 byte, 8 δυαδικά ψηφία) σειριακά. Για το σκοπό αυτό μπορεί να χρησιμοποιήσει έναν καταχωρητή ολίσθησης 8 βαθμίδων. Ο καταχωρητής αυτός φορτώνεται παράλληλα ($s_1s_0=01$). Στη συνέχεια, λειτουργεί με δεξιά ολίσθηση για 8 κύκλους και μεταφέρει τα δεδομένα, bit-προς-bit, στη μονάδα μέσω της σειριακής εξόδου δεξιάς ολίσθησης.

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

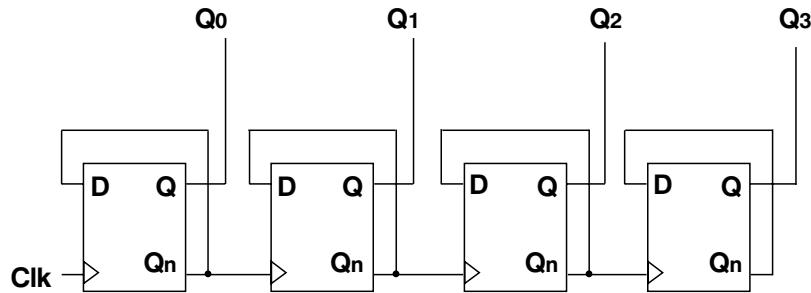
2.9 Μετρητές ή απαριθμητές

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

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

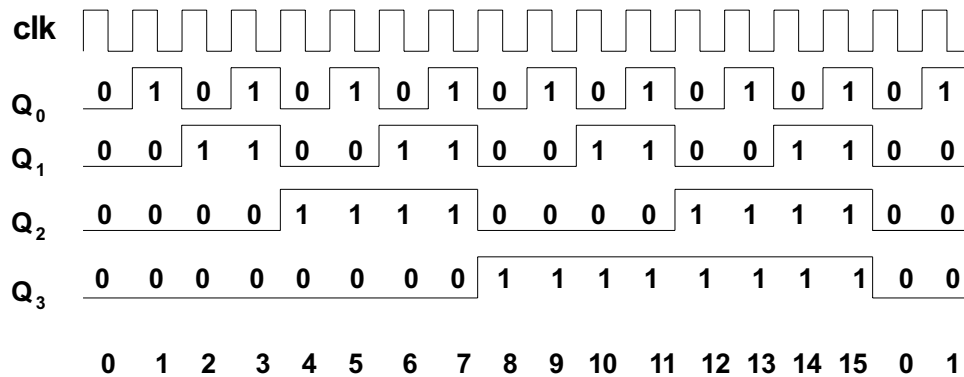
Ασύγχρονοι μετρητές

Ένας ασύγχρονος μετρητής αποτελείται από flip-flops συνδεδεμένα έτσι ώστε η έξοδος κάθε flip-flop να συνδέεται στην είσοδο χρονισμού του flip-flop της επόμενης βαθμίδας. Το flip-flop της πρώτης βαθμίδας δέχεται τους εισερχόμενους παλμούς μέτρησης. Ένας ασύγχρονος απαριθμητής 4 βαθμίδων αποτελούμενος από flip-flops τύπου D φαίνεται στο Σχήμα 2.37.



Σχήμα 2.37 Ασύγχρονος απαριθμητής 4 βαθμίδων

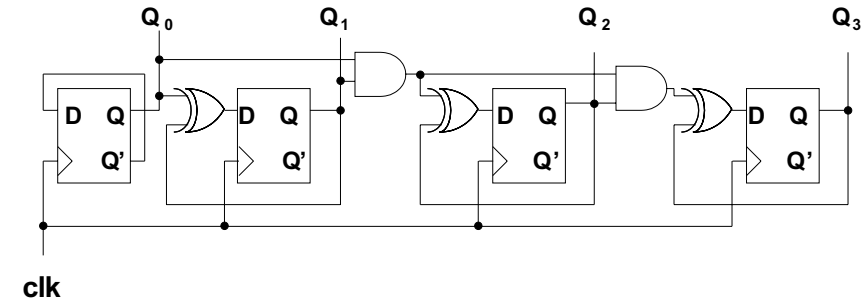
Ο απαριθμητής αυτός λειτουργεί ως εξής. Αρχικά οι έξοδοι όλων των βαθμίδων είναι 0. Όταν το σήμα χρονισμού clk γίνει 1 από 0, η τιμή του flip-flop Q₀ θα αλλάξει από 0 σε 1. Σε κάθε μεταβολή του σήματος χρονισμού από '0' σε '1', η τιμή του flip-flop Q₀ θα μεταβάλλεται. Επιπλέον, κάθε φορά που η τιμή του flip-flop Q₀ αλλάζει από 0 σε 1, η τιμή του flip-flop Q₁ θα μεταβάλλεται. Γενικά, όταν η τιμή μιας βαθμίδας μεταβάλλεται από '0' σε '1', η τιμή της επόμενης βαθμίδας θα αλλάζει. Στο Σχήμα 2.38 φαίνεται η ακολουθία που παράγεται από το δυαδικό απαριθμητή του προηγούμενου σχήματος ξεκινώντας από την τιμή 0000.



Σχήμα 2.38 Ακολουθία που παράγεται από τον απαριθμητή 4-βαθμίδων

Σύγχρονοι Μετρητές

Οι σύγχρονοι μετρητές διαφέρουν από τους μετρητές ριπής στο ότι οι παλμοί του ρολογιού εφαρμόζονται στις εισόδους χρονισμού όλων των flip-flops. Ο κοινός παλμός πυροδοτεί όλα τα flip-flops συγχρόνως, και όχι το ένα μετά το άλλο, όπως στους μετρητές ριπής. Ένας σύγχρονος δυαδικός μετρητής σχεδιασμένος με flip-flops τύπου D φαίνεται στο Σχήμα 2.39.



Σχήμα 2.39 Σύγχρονος δυαδικός απαριθμητής τεσσάρων βαθμίδων

Η είσοδος δεδομένων κάθε flip-flop οδηγείται από μια πύλη XOR της οποίας η μια είσοδος είναι η έξοδος του flip-flop. Η απόφαση σχετικά με το αν η έξοδος ενός flip-flop πρέπει να αντιστραφεί ή όχι στηρίζεται στην τιμή της εξόδου της πύλης AND που τροφοδοτεί την άλλη έξοδο της πύλης XOR. Η έξοδος του πρώτου flip-flop αντιστρέφεται σε κάθε κύκλο ρολογιού. Η έξοδος του δεύτερου flip-flop αντιστρέφεται όταν η έξοδος του πρώτου flip-flop είναι 1. Η έξοδος των επόμενων βαθμίδων αντιστρέφεται όταν οι έξοδοι όλων των προηγούμενων βαθμίδων είναι '1'. Έτσι, το Q₂ αντιστρέφεται όταν Q₀=Q₁=1, ενώ η Q₃ αντιστρέφεται όταν Q₀=Q₁=Q₂=1.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Κ.Ζ. Πεκμεστζή**, "Συστήματα Μικροϋπολογιστών", Εκδόσεις Συμμετρία, 1995.
2. **Mano M. Morris**, "Ψηφιακή Σχεδίαση", Παπασωτηρίου 1992, Δεύτερη Έκδοση, Μετάφραση Απ. Τραγανίτης.

Κεφάλαιο 3ο

Αρχιτεκτονική Ηλεκτρονικού Τμήματος (hardware) των Υπολογιστικών Συστημάτων

Περιεχόμενα

- 3.1 Βασικά στοιχεία αρχιτεκτονικής μικροπολογιστικών Συστημάτων
- 3.2 Αρχές λειτουργίας και αρχιτεκτονική μικροεπεξεργαστών
- 3.3 Εντολές μικροεπεξεργαστών
- 3.4 Τρόποι αναφοράς στη μνήμη
- 3.5 Χαρακτηριστικά και κατηγορίες μικροεπεξεργαστών
- 3.6 Οικογένειες μικροεπεξεργαστών
- 3.7 Μικροελεγκτές

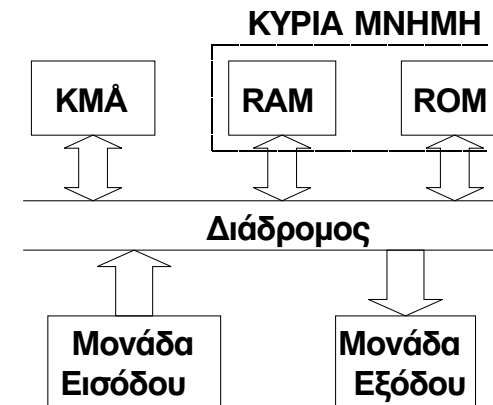
Στόχοι του κεφαλαίου:

Όταν ολοκληρώσεις το κεφάλαιο αυτό θα μπορείς να...

αναφέρεις τα βασικά τμήματα της δομής των υπολογιστικών συστημάτων
 αναφέρεις τη λειτουργία και χρήση κάθε τμήματος ενός υπολογιστικού συστήματος
 αναφέρεις τα τμήματα από τα οποία αποτελείται ένας μικροεπεξεργαστής, καθώς και τη λειτουργία τους
 αναφέρεις τους βασικούς καταχωρητές ενός μικροεπεξεργαστή
 αναφέρεις τα είδη εντολών ενός τυπικού μικροεπεξεργαστή
 περιγράφεις τον τρόπο εκτέλεσης των εντολών σε ένα μικρο-επεξεργαστή
 αναφέρεις τις διαφορές της συμβολικής γλώσσας από τη γλώσσα μηχανής
 αναφέρεις τους τρόπους διευθυνοδότησης της μνήμης
 αναφέρεις τα χαρακτηριστικά ως προς τα οποία διακρίνουμε τους μικροεπεξεργαστές
 αναφέρεις τα είδη των μικροεπεξεργαστών ως προς καθένα από τα χαρακτηριστικά αυτά
 αναφέρεις τις πιο γνωστές οικογένειες επεξεργαστών, καθώς και τα μέλη κάθε μιας από τις οικογένειες αυτές
 αναφέρεις τις διαφορές ενός μικροελεγκτή από ένα μικροεπεξεργαστή
 εξηγείς γιατί οι μικροελεγκτές είναι χρήσιμοι στο σχεδιασμό μικρουπολογιστικών συστημάτων

3.1 Βασικά στοιχεία αρχιτεκτονικής μικροϋπολογιστικών Συστημάτων

Με τον όρο υπολογιστικό σύστημα, αναφερόμαστε σε ένα σύστημα το οποίο αποτελείται από μια Κεντρική Μονάδα Επεξεργασίας, μνήμη και συσκευές εισόδου και εξόδου. Γενικά, στόχος μας όταν σχεδιάζουμε ένα τέτοιο σύστημα είναι να το χρησιμοποιήσουμε για την εκτέλεση υπολογισμών ή για να ελέγξουμε άλλες συσκευές. Το γενικό διάγραμμα ενός υπολογιστικού συστήματος φαίνεται στο Σχήμα 3.1.



Σχήμα 3.1 Γενικό διάγραμμα υπολογιστικού συστήματος

Κεντρική Μονάδα Επεξεργασίας και μικροεπεξεργαστές

Ένα υπολογιστικό σύστημα στο οποίο ως Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) χρησιμοποιείται ένας μικροεπεξεργαστής (micro-processor) ονομάζεται μικροϋπολογιστικό σύστημα. Ένας μικροεπεξεργαστής είναι ένα ολοκληρωμένο κύκλωμα γενικού σκοπού, το οποίο μπορεί να προγραμματιστεί.



Σχήμα 3.2 Μικροεπεξεργαστής 486 της εταιρείας Intel

Η επεξεργασία των δεδομένων γίνεται σε μια σειρά από βήματα, κάθε ένα από τα οποία ονομάζεται εντολή. Οι εντολές που εκτελούνται από το μικροεπεξεργαστή είναι εντολές σε γλώσσα μηχανής. Μια εντολή σε γλώσσα μηχανής είναι μια σειρά από δυαδικά ψηφία όπου είναι κωδικοποιημένο το είδος της εντολής. Οι εντολές της γλώσσας μηχανής είναι αποθηκευμένες στην κύρια μνήμη, από όπου τις ανακαλεί και τις εκτελεί ο μικροεπεξεργαστής.

Η δομή και η λειτουργία ενός μικροεπεξεργαστή θα περιγραφούν στη συνέχεια του κεφαλαίου αυτού.

Κύρια Μνήμη

Όπως αναφέραμε, η κύρια μνήμη είναι ένας χώρος στον οποίο ο υπολογιστής φυλάει δεδομένα ή εντολές προς εκτέλεση. Η κύρια μνήμη αποτελείται από λέξεις μνήμης (memory words), κάθε μια από τις οποίες αποτελείται από έναν αριθμό δυαδικών ψηφίων. Ο αριθμός αυτός ονομάζεται μήκος λέξης της μνήμης. Κάθε θέση προσδιορίζεται από έναν αριθμό που ονομάζεται διεύθυνση (address). Για να διαβάσουμε από ή να γράψουμε ένα δεδομένο σε μια θέση μνήμης πρέπει να γνωρίζουμε τη διεύθυνσή της.

Η κύρια μνήμη του υπολογιστή χωρίζεται σε δύο μέρη: στη μνήμη από την οποία ο μικροεπεξεργαστής μπορεί να διαβάσει μόνο (**Read Only Memory, ROM**) και στη μνήμη στην οποία η ΚΜΕ μπορεί και να γράψει και να διαβάσει. Στη μνήμη αυτή αναφερόμαστε με τον όρο μνήμη τυχαίας προσπέλασης (**Random Access Memory, RAM**).

Οι λειτουργίες με τις οποίες ο μικροεπεξεργαστής επικοινωνεί με τη μνήμη RAM είναι η εγγραφή και η ανάγνωση.

Στην εγγραφή, η μνήμη δέχεται τη διεύθυνση στην οποία θα γίνει η εγγραφή και τα περιεχόμενα που θα γραφούν στη θέση αυτή. Η διεύθυνση της θέσης μνήμης στην οποία θα πραγματοποιηθεί η εγγραφή τοποθετείται στον καταχωρητή διευθύνσεων του μικροεπεξεργαστή, ενώ το δεδομένο τοποθετείται στον καταχωρητή δεδομένων του μικροεπεξεργαστή. Στη συνέχεια, η διεύθυνση της θέσης μνήμης μεταφέρεται μέσω ειδικού διαδρόμου (του διαδρόμου διευθύνσεων, όπως θα δούμε σε επόμενη παράγραφο) ενώ το δεδομένο μέσω άλλου διαδρόμου (του διαδρόμου δεδομένων) και πραγματοποιείται η εγγραφή.

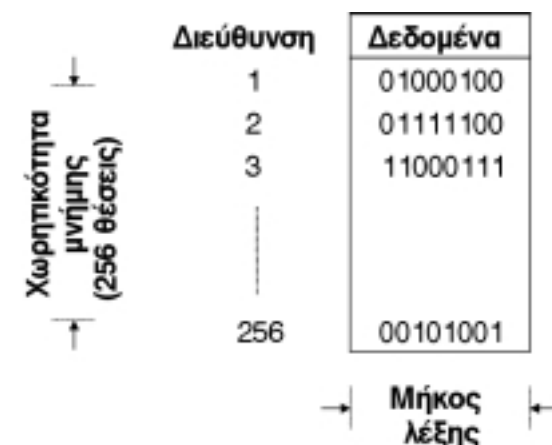
Στην ανάγνωση, τα περιεχόμενα της θέσης μνήμης που υποδεικνύει ο καταχωρητής διευθύνσεων του μικροεπεξεργαστή μεταφέρονται στον καταχωρητή δεδομένων του μικροεπεξεργαστή.

Η διεύθυνση της θέσης μνήμης μεταφέρεται μέσω του διαδρόμου διευθύνσεων, πραγματοποιείται η ανάγνωση της μνήμης και το δεδομένο (το περιεχόμενο της θέσης μνήμης) μεταφέρεται μέσω του διαδρόμου δεδομένων στον καταχωρητή δεδομένων του μικροεπεξεργαστή.

Η μνήμη ROM περιέχει πληροφορίες βασικές για τη λειτουργία του υπολογιστή, τις οποίες έχει τοποθετήσει εκεί ο κατασκευαστής.

Τα βασικά χαρακτηριστικά της κύριας μνήμης είναι τα ακόλουθα:

- Μήκος λέξης
- Χωρητικότητα
- Χρόνος προσπέλασης



Σχήμα 3.3 Μνήμη 256 θέσεων, με μήκος λέξης 8 δυαδικά ψηφία

Το μήκος λέξης είναι το πλήθος των δυαδικών ψηφίων κάθε λέξης της μνήμης (ίδιο για όλες τις λέξεις). Το μήκος λέξης είναι πολλαπλάσιο του byte και είναι συνήθως 1, 2, ή 4 byte (8, 16, ή 32 δυαδικά ψηφία). Πολύ συχνά, το μέγεθος λέξης της μνήμης είναι ίσο με το μήκος λέξης στο οποίο μπορεί να εκτελέσει πράξεις ο μικροεπεξεργαστής.

Με τον όρο *χωρητικότητα* αναφερόμαστε στο μέγεθος της μνήμης, με άλλα λόγια στο πλήθος των bytes που μπορεί να χωρέσει. Συνήθως το μέγεθος αυτό μετράται σε πολλαπλάσια του byte, τα πιο γνωστά από τα οποία είναι τα Kilobyte, Megabyte και Gigabyte, όπως φαίνεται στον ακόλουθο πίνακα.

1 Kbyte	2^{10} byte	≈ 1.000 byte
1 Mbyte	2^{20} byte	$\approx 1.000.000$ byte
1 Gbyte	2^{30} byte	$\approx 1.000.000.000$ byte

Πίνακας 3.1 Πολλαπλάσια του byte

Ο χρόνος προσπέλασης είναι ο χρόνος που περνάει από τη στιγμή που ο μικροεπεξεργαστής ζητάει από τη μνήμη το περιεχόμενο μιας θέσης, μέχρι τη στιγμή που η μνήμη δίνει το περιεχόμενο αυτό στο μικροεπεξεργαστή. Ο χρόνος προσπέλασης μετράται σε nanoseconds (1 nanosecond ισούται με

1 δισεκατομμυριοστό του δευτερολέπτου, $1\text{ns}=10^{-9}\text{ sec}$) και είναι ένα μέτρο της ταχύτητας της μνήμης. Ένας άλλος τρόπος με τον οποίο μετράμε την ταχύτητα της μνήμης είναι η συχνότητα λειτουργίας της, με άλλα λόγια πόσες φορές μπορούμε να προσπελάσουμε τη μνήμη (για ανάγνωση ή εγγραφή) στη μονάδα του χρόνου (δευτερόλεπτο).

Μονάδες Εισόδου - Εξόδου

Με τον όρο μονάδες εισόδου αναφερόμαστε στο σύνολο των συσκευών ή διατάξεων, που επιτρέπουν τη μετατροπή πληροφοριών (κείμενο, εικόνα, ήχο, video κ.λπ.) σε ψηφιακή αναπαράσταση, ώστε να εισαχθεί στον υπολογιστή (πχ. πληκτρολόγιο, ποντίκι, σαρωτής). Οι μονάδες εξόδου μετατρέπουν την πληροφορία από ψηφιακή αναπαράσταση σε κείμενο, ήχο κ.λπ (πχ. οθόνη, εκτυπωτής). Οι μονάδες που χρησιμοποιούνται και για την είσοδο αλλά και για την έξοδο δεδομένων ονομάζονται μονάδες εισόδου και εξόδου (πχ. modems, κάρτες ήχου και video).

Μια συσκευή εισόδου-εξόδου μπορεί να συνδεθεί στο σύστημα μέσω μιας θύρας εισόδου-εξόδου (I/O port). Οι θύρες μπορούν να θεωρηθούν ως εξωτερικοί καταχωρητές τους οποίους μπορεί να προσπελάσει ο μικροεπεξεργαστής.

Διάδρομοι

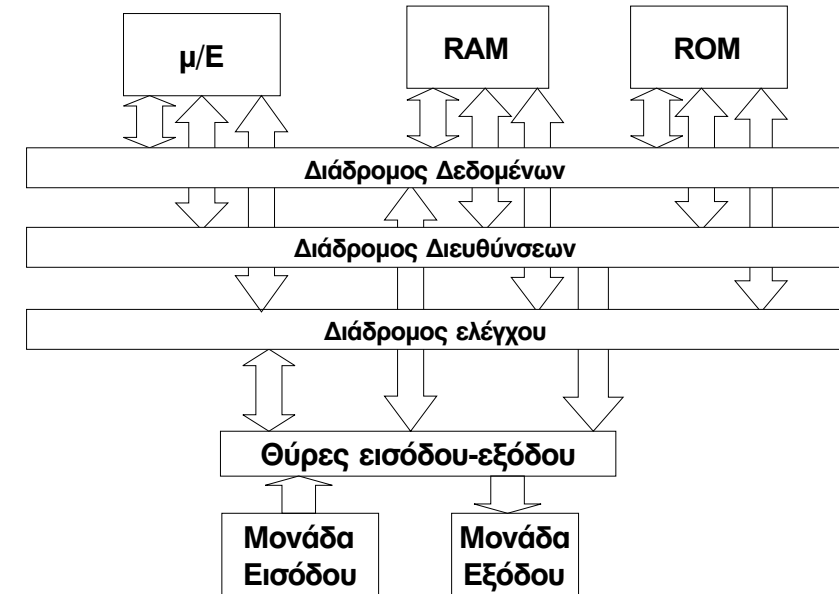
Ένας διάδρομος είναι μια ομάδα αγωγών που χρησιμοποιείται για την επικοινωνία μεταξύ των μονάδων του υπολογιστή.

Ένας διάδρομος χωρίζεται λειτουργικά σε τρία μέρη: το διάδρομο δεδομένων (data bus), το διάδρομο διευθύνσεων (address bus) και το διάδρομο ελέγχου (control bus).

Μέσω του διαδρόμου δεδομένων μεταφέρονται τα δεδομένα που θέλουμε να γράψουμε ή να διαβάσουμε κάθε φορά (πχ. τα δυαδικά ψηφία που συνθέτουν το περιεχόμενο μιας θέσης μνήμης, ενός καταχωρητή του μικροεπεξεργαστή, ή δεδομένα από κάποια άλλη μονάδα). Όταν γράφουμε, μεταφέρονται δεδομένα από το μικροεπεξεργαστή προς τη μνήμη RAM ή προς τις μονάδες εξόδου. Όταν διαβάζουμε, μεταφέρονται δεδομένα από τις μνήμες RAM ή ROM και από τις μονάδες εισόδου. Μέσω του διαδρόμου διευθύνσεων μεταφέρονται δυαδικά ψηφία που σχηματίζουν τη διεύθυνση μιας θέσης μνήμης ή τη διεύθυνση μιας συσκευής εισόδου-εξόδου, δηλαδή προσδιορίζουν πού θα γραφτεί ή από πού θα διαβαστεί ένα δεδομένο.

Μέσω του διαδρόμου ελέγχου ο μικροεπεξεργαστής πληροφορεί τη μνήμη ή τις περιφερειακές συσκευές για την ενέργεια που προτίθεται να κάνει (π.χ. να διαβάσει ή να γράψει δεδομένα).

Αξίζει να σημειωθεί ότι κάθε χρονική στιγμή μόνο δύο συσκευές μπορούν να επικοινωνούν μέσω του διαδρόμου. Έτσι, αν κάποια στιγμή επικοινωνεί μέσω του διαδρόμου ο μικροεπεξεργαστής με τη μνήμη, μια μονάδα εισόδου δε μπορεί να στείλει δεδομένα, αλλά πρέπει να περιμένει να ολοκληρωθεί η επικοινωνία μεταξύ του μικροεπεξεργαστή και της κύριας μνήμης. Ένα τυπικό μικροϋπολογιστικό σύστημα φαίνεται στο Σχήμα 3.4.



Σχήμα 3.4 Τυπικό μικροϋπολογιστικό σύστημα

Γενικά, σε ένα μικροϋπολογιστικό σύστημα ο μικροεπεξεργαστής ελέγχει και συντονίζει μεταφορές και πράξεις στα δεδομένα σύμφωνα με τις εντολές που διαβάζει από κάποιο πρόγραμμα που βρίσκεται στη μνήμη.

3.2 Αρχές λειτουργίας και αρχιτεκτονική μικροεπεξεργαστών

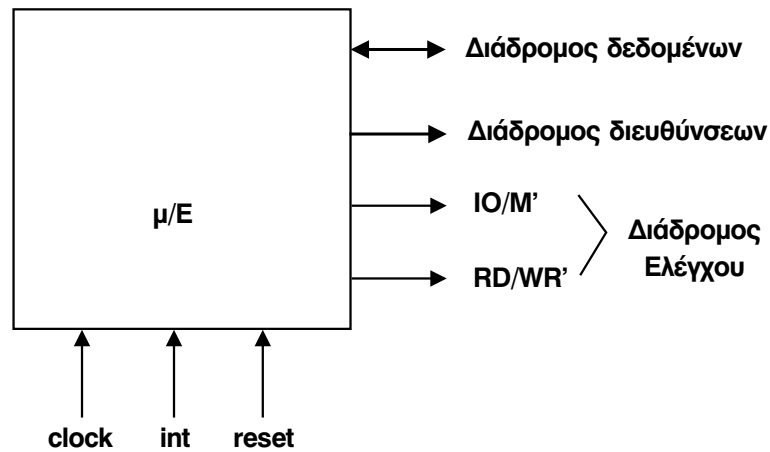
Ένας μικροεπεξεργαστής είναι ένα ολοκληρωμένο κύκλωμα που μπορεί να προγραμματιστεί.

Οι μικροεπεξεργαστές ξεκίνησαν ως μια ενδιάμεση φθηνή λύση στον έλεγχο συστημάτων μεταξύ των υπολογιστών και του ειδικού υλικού που έπρεπε να κατασκευαστεί κατά περίπτωση. Η εμφάνισή τους άλλαξε τη σχεδίαση των υπολογιστικών συστημάτων απλοποιώντας την αρκετά.

Το υλικό που βασίζεται σε μικροεπεξεργαστή μπορεί να χρησιμοποιηθεί χωρίς μετατροπές (με τροποποίηση του προγράμματος) σε ποικιλία εφαρμογών, ενώ ένα ψηφιακό σύστημα που κατασκευάζεται με πύλες χρησιμοποιείται μόνο σε μια εφαρμογή. Έτσι, με τους μικροεπεξεργαστές το πρόβλημα της σχεδίασης ψηφιακών

συστημάτων μετατοπίστηκε από το υλικό στο λογισμικό και η σχεδίαση απαλλάχθηκε από την ακαμψία του υλικού.

Το Σχήμα 3.5 περιγράφει τις εξωτερικές συνδέσεις ενός μικροεπεξεργαστή.



Σχήμα 3.5 Εξωτερικές συνδέσεις τυπικού μικροεπεξεργαστή

Ο διάδρομος ελέγχου συνήθως περιλαμβάνει τουλάχιστον δύο σήματα, τα οποία συμβολίζουμε RD/WR' (read/write)², IO/M' (Input Output/ Memory). Τα σήματα αυτά παράγονται από το μικροεπεξεργαστή και έχουν την ακόλουθη σημασία :

Αν RD/WR'=1, πρόκειται να πραγματοποιηθεί ανάγνωση από κάποια θέση της μνήμης ή από μονάδα εισόδου, διαφορετικά πρόκειται να πραγματοποιηθεί εγγραφή. Αν IO/M'=1, πρόκειται να προσπελαστεί (για ανάγνωση ή εγγραφή) μια μονάδα εισόδου ή εξόδου, διαφορετικά θα προσπελαστεί η μνήμη. Ο Πίνακας 3.2 συνοψίζει τις λειτουργίες αυτές.

RD/WR'	IO/M'	Λειτουργία
0	0	εγγραφή σε μνήμη (RAM)
0	1	εγγραφή σε συσκευή εξόδου
1	0	ανάγνωση από μνήμη (RAM ή ROM)
1	1	ανάγνωση από συσκευή εισόδου

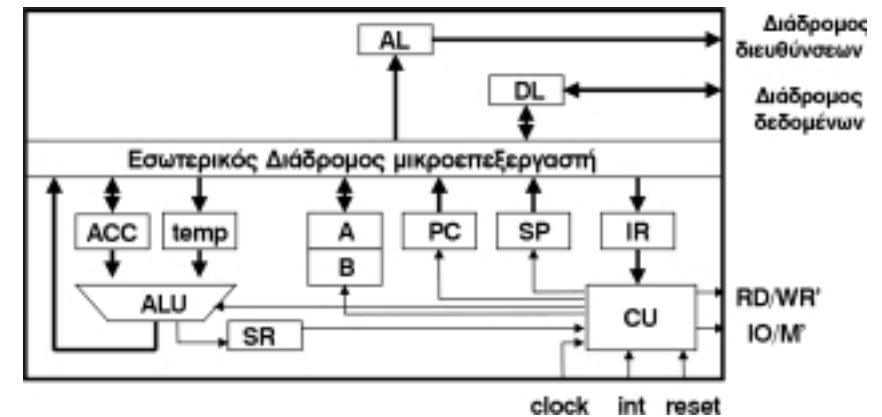
Πίνακας 3.2 Λειτουργίες του μικροπολογιστικού συστήματος του προηγούμενου σχήματος

² Με το συμβολισμό RD/WR' εννοούμε ότι το σήμα έχει την ακόλουθη σημασία: Όταν RD/WR'=1, λειτουργεί ως σήμα ανάγνωσης (Read). Αν RD/WR'=0, τότε λειτουργεί ως σήμα εγγραφής (Write).

Όπως γνωρίζουμε, ένας μικροεπεξεργαστής αποτελείται από τα ακόλουθα τμήματα:

- την αριθμητική και λογική μονάδα (Arithmetic and Logic Unit, ALU)
- τη μονάδα ελέγχου (control unit, CU)
- τους καταχωρητές (registers)

Το σχηματικό διάγραμμα της εσωτερικής δομής ενός τυπικού μικροεπεξεργαστή φαίνεται στο Σχήμα 3.6.



- ALU:** αριθμητική και λογική μονάδα (arithmetic and logic unit)
- CU:** μονάδα ελέγχου (control unit)
- ACC:** συσσωρευτής (accumulator)
- temp:** προσωρινός καταχωρητής
- A,B:** καταχωρητές γενικού σκοπού
- IR:** καταχωρητής εντολών (instruction register)
- SR:** καταχωρητής κατάστασης (status register)
- PC:** μετρητής προγράμματος (program counter)
- SP:** δείκτης στοίβας (stack pointer)
- AL:** απομονωτής διευθύνσεων (address latch)
- DL:** απομονωτής δεδομένων (data latch)

Σχήμα 3.6 Εσωτερική δομή τυπικού μικροεπεξεργαστή

Αριθμητική και λογική μονάδα

Η αριθμητική και λογική μονάδα εκτελεί (αριθμητικές και λογικές) πράξεις. Τα δεδομένα στα οποία εκτελούνται οι πράξεις αυτές βρίσκονται σε δύο καταχωρητές εκ των οποίων ο ένας ονομάζεται συνήθως συσσωρευτής. Το αποτέλεσμα της εκτέλεσης της πράξης φυλάσσεται στο συσσωρευτή. Συνηθισμένες πράξεις που εκτελούνται στην αριθμητική και λογική μονάδα είναι οι ακόλουθες:

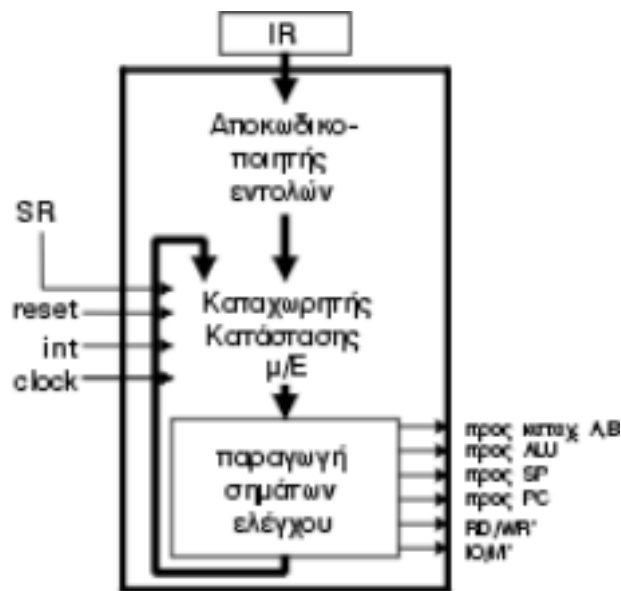
- δυαδική πρόσθεση και αφαίρεση
- λογικό ΚΑΙ, Η, αποκλειστικό Η (AND, OR, XOR).
- συμπλήρωμα (ως προς 1 και ως προς 2).
- ολίσθηση και περιστροφή (δεξιά ή αριστερά)
- πολλαπλασιασμός και διαίρεση (δεν περιλαμβάνονται σε όλους τους μικροεπεξεργαστές)

Το αποτέλεσμα της εκτέλεσης της πράξης στην ALU επηρεάζει επίσης μια σειρά από flip-flops που ονομάζονται σημαίες (flags). Για παράδειγμα, μια σημαία μπορεί να δείχνει αν προέκυψε κρατούμενο από το πιο σημαντικό ψηφίο μετά την εκτέλεση μιας πρόσθεσης, ενώ μια άλλη σημαία μπορεί να δείχνει αν το αποτέλεσμα μιας πράξης είναι 0. Το σύνολο των σημαιών αποτελεί τον καταχωρητή κατάστασης (status register) του μικροεπεξεργαστή, στον οποίο θα αναφερθούμε εκτενέστερα στη συνέχεια.

Μονάδα ελέγχου

Η μονάδα ελέγχου ενός μικροεπεξεργαστή ελέγχει και συγχρονίζει τη μεταφορά και επεξεργασία των δεδομένων και είναι το κύριο υποσύστημα του ίδιου του μικροεπεξεργαστή. Όλες οι ενέργειες σε ένα μικροεπεξεργαστή πραγματοποιούνται με την επίβλεψη της μονάδας ελέγχου.

Η μονάδα ελέγχου χρησιμοποιεί ως είσοδο ένα εξωτερικό ρολόι (clk) και παράγει σήματα χρονισμού και ελέγχου που ρυθμίζουν τις μεταφορές δεδομένων και τις πράξεις που περιλαμβάνονται σε κάθε εντολή. Το σχηματικό διάγραμμα της μονάδας ελέγχου φαίνεται στο Σχήμα 3.7.



Σχήμα 3.7 Μονάδα ελέγχου τυπικού μικροεπεξεργαστή

Καταχωρητές

Οι καταχωρητές χρησιμεύουν ως χώροι αποθήκευσης δεδομένων (καταχωρητές δεδομένων) ή διευθύνσεων της μνήμης (καταχωρητές διευθύνσεων).

Οι καταχωρητές που συναντάμε στην πλειοψηφία των μικροεπεξεργαστών είναι οι ακόλουθοι:

- ο απαριθμητής προγράμματος (Program Counter, PC),
- ο καταχωρητής εντολών (Instruction Register, IR)
- ο δείκτης στοίβας (stack pointer) και
- ο καταχωρητής κατάστασης (Status Register, SR)

Απαριθμητής προγράμματος

Ο απαριθμητής προγράμματος είναι ένας καταχωρητής διευθύνσεων, στον οποίο φυλάγεται η διεύθυνση της μνήμης από την οποία θα ανακληθεί η επόμενη προς εκτέλεση εντολή.

Ο απαριθμητής προγράμματος μπορεί να απευθυνθεί σε οποιαδήποτε θέση μνήμης από εκείνες που μπορεί να δει ο μικροεπεξεργαστής για να ανακαλέσει από εκεί μια εντολή.

Η εκκίνηση του μικροεπεξεργαστή γίνεται ως εξής. Όταν η είσοδος reset είναι στο '1', ο απαριθμητής προγράμματος παίρνει την τιμή 0000 (ή γενικότερα κάποια αρχική τιμή). Όταν το σήμα reset ξαναγυρίσει στο '0' η μονάδα ελέγχου μεταφέρει τα περιεχόμενα του απαριθμητή προγράμματος στον απομονωτή διευθύνσεων (address latch) παρέχοντας έτσι τη διεύθυνση της πρώτης εντολής που θα εκτελεστεί.

Με αυτόν τον τρόπο (συνήθως) η εκτέλεση των προγραμμάτων σε ένα μικροϋπολογιστικό σύστημα ξεκινάει με την εντολή που βρίσκεται στη θέση μνήμης 0. Για το λόγο αυτό, οι πρώτες θέσεις μνήμης αντιστοιχούν σε διευθύνσεις της μνήμης ROM. Τα προγράμματα που περιέχονται στις θέσεις αυτές (πχ. βασικές ρουτίνες εισόδου-εξόδου δεδομένων) είναι σημαντικά για την ομαλή λειτουργία του μικροϋπολογιστικού συστήματος.

Καταχωρητής εντολών

Ο καταχωρητής εντολών είναι ένας καταχωρητής στον οποίο μεταφέρεται η εντολή που διαβάστηκε από τη μνήμη. Στη συνέχεια, η εντολή αποκωδικοποιείται μέσω του αποκωδικοποιητή εντολών προκειμένου να εκτελεστεί.

Οι εντολές ενός μικροεπεξεργαστή μπορεί να έχουν μήκος 1 ή περισσότερες λέξεις μνήμης. Η πρώτη λέξη περιέχει τον κωδικό λειτουργίας (operation code ή opcode), με άλλα λόγια το είδος της εντολής που ανακαλείται (αν πρόκειται δηλαδή για εντολή εκτέλεσης αριθμητικής πράξης, μεταφοράς δεδομένων, και ποια). Με την ανάγνωση του κωδικού της εντολής ο μικροεπεξεργαστής είναι σε θέση να γνωρίζει αν η εντολή καταλαμβάνει περισσότερες θέσεις μνήμης και ποιες, καθώς και ποιες λειτουργίες θα εκτελέσει στη συνέχεια, ποια σήματα θα πρέπει να ενεργοποιήσει κ.λπ.

Κατά τη διάρκεια ανάκλησης της εντολής (instruction fetch), η πρώτη λέξη μνήμης μεταφέρεται από τη μνήμη μέσω του εξωτερικού διαδρόμου δεδομένων (data bus)

στον καταχωρητή εντολών (instruction register). Ο απαριθμητής προγράμματος αυξάνεται αυτόματα κατά 1, οπότε περιέχει τη διεύθυνση της επόμενης εντολής (αν η τρέχουσα εντολή καταλαμβάνει μια μόνο θέση μνήμης) ή στην επόμενη θέση μνήμης αυτής της εντολής αν η εντολή περιέχει 2 ή περισσότερα bytes.

Στην περίπτωση που μια εντολή καταλαμβάνει πολλές θέσεις μνήμης (multiword instruction), η μονάδα ελέγχου προχωρά στην ανάγνωση των επόμενων θέσεων μνήμης όπως αναφέραμε. Για το σκοπό αυτό, η μονάδα ελέγχου χρησιμοποιεί τα σήματα του αποκωδικοποιητή εντολών και παράγει τα κατάλληλα σήματα χρονισμού (timing) καθώς και ελέγχου εξωτερικών συσκευών.

Όταν φορτωθούν όλες οι λέξεις στο μικροεπεξεργαστή, εκτελείται η εντολή. Η εκτέλεση της εντολής, εκτός από τις λειτουργίες που γίνονται εσωτερικά στο μικροεπεξεργαστή, μπορεί να απαιτήσει μεταφορά δεδομένων μεταξύ του μικροεπεξεργαστή και των μονάδων μνήμης και εισόδου-εξόδου.

Δείκτης στοίβας

Ο δείκτης στοίβας (Stack Pointer, SP) είναι ένας δείκτης διευθύνσεων της κορυφής της στοίβας η οποία βρίσκεται στην κύρια μνήμη. Η στοίβα είναι μια περιοχή της μνήμης η οποία χρησιμοποιείται κυρίως για την εξυπηρέτηση κλήσεων διακοπών και υπορουτινών, όπως θα περιγράψουμε στο επόμενο κεφάλαιο.

Καταχωρητής κατάστασης

Ο καταχωρητής αυτός δίνει πληροφορίες σχετικά με τα αποτελέσματα της τελευταίας εντολής που εκτελέστηκε και σχετίζεται, όπως αναφέραμε, με την ALU. Ο καταχωρητής αυτός περιλαμβάνει κάποιες σημαίες συνθήκης, οι πιο γνωστές από τις οποίες είναι οι σημαίες μηδενισμού (zero), προσήμου (sign), ισοτιμίας (parity) και κρατουμένου (carry). Οι σημαίες αυτές επηρεάζονται από την εκτέλεση των εντολών ως εξής:

- Σημαία μηδενισμού (Z, zero): αν το αποτέλεσμα μιας εντολής είναι 0, τότε $Z=1$, διαφορετικά $Z=0$.
- Σημαία προσήμου (S, sign): αν το περισσότερο σημαντικό ψηφίο του αποτελέσματος μιας πράξης είναι '1' (ο αριθμός είναι αρνητικός), τότε $S=1$, αλλιώς $S=0$.
- Σημαία ισοτιμίας (P, parity): αν το αποτέλεσμα μιας πράξης έχει άρτιο αριθμό '1', δηλαδή άρτια ισοτιμία, τότε $P=1$, αλλιώς $P=0$.
- Σημαία κρατουμένου (C, carry): αν η εντολή είχε ως αποτέλεσμα να προκύψει κρατούμενο (από πρόσθεση) ή δανεικό (από αφαίρεση) τότε $C=1$, αλλιώς $C=0$.

3.3 Εντολές μικροεπεξεργαστών

Μέχρι τώρα, περιγράψαμε τη δομή του μικροεπεξεργαστή. Όπως γνωρίζουμε ο

μικροεπεξεργαστής εκτελεί εντολές τις οποίες ανακαλεί από τη μνήμη. Μια εντολή είναι μια σειρά από δυαδικά ψηφία αποθηκευμένα στη μνήμη. Στην παράγραφο αυτή θα αναφερθούμε στη διαδικασία μέσω της οποίας οι εντολές ανακαλούνται από τη μνήμη και εκτελούνται, καθώς και στα είδη των εντολών που υπάρχουν στους μικροεπεξεργαστές.

3.3.1 Εκτέλεση εντολής

Η διαδικασία που ακολουθείται, προκειμένου να εκτελεστεί η εντολή περιλαμβάνει τις ακόλουθες φάσεις: κλήση εντολής, αποκωδικοποίηση εντολής και εκτέλεση της εντολής.

Στη φάση κλήσης της εντολής μεταφέρεται ο κώδικας της επόμενης εντολής από τη μνήμη (από τη θέση που δείχνει ο μετρητής προγράμματος PC).

Στην αποκωδικοποίηση (του κώδικα της εντολής), αν η εντολή περιέχει και άλλα byte δεδομένων, τότε μεταφέρεται από τη μνήμη η υπόλοιπη εντολή.

Η φάση εκτέλεσης της εντολής εξαρτάται από την εντολή που πρόκειται να εκτελεστεί.

Όπως γνωρίζουμε, οι πληροφορίες που αποθηκεύονται στη μνήμη μπορούν να ερμηνευθούν από το μικροεπεξεργαστή είτε ως δεδομένα είτε ως εντολές. Για να εκτελεστεί ένα πρόγραμμα από το μικροεπεξεργαστή, πρέπει οι εντολές του προγράμματος να είναι αποθηκευμένες σε κάποιες από τις θέσεις μνήμης. Συνήθως μάλιστα, οι εντολές αποθηκεύονται σε διαδοχικές θέσεις της μνήμης σύμφωνα με τη σειρά με την οποία πρέπει να εκτελεστούν.

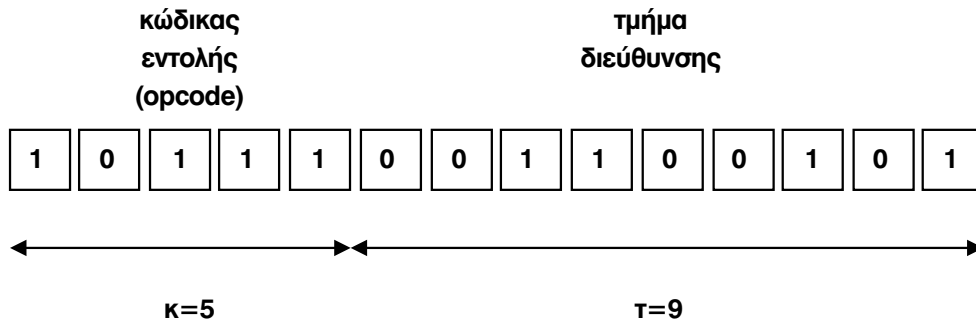
3.3.2 Γλώσσα μηχανής και συμβολική γλώσσα

Κάθε εντολή παριστάνεται με μια ή περισσότερες λέξεις του υπολογιστή και χωρίζεται σε δύο τμήματα:

- τον κώδικα εντολής (opcode) που έχει μήκος k δυαδικά ψηφία
- το τμήμα διεύθυνσεως που έχει μήκος t δυαδικά ψηφία.

Το άθροισμα $k+t$ των δυαδικών ψηφίων του κώδικα και του τμήματος διεύθυνσης δίνει το μήκος της εντολής. Ο κώδικας της εντολής ορίζει τη λειτουργία που πρέπει να εκτελεστεί από το μικροεπεξεργαστή, ενώ το τμήμα διεύθυνσεως περιέχει τα δεδομένα της.

Στο Σχήμα 3.8 φαίνεται ένα παράδειγμα μιας εντολής μήκους οκτώ δυαδικών ψηφίων ενός μικροεπεξεργαστή, από τα οποία τα 5 χρησιμοποιούνται για το τμήμα εντολής, ενώ τα 9 για το τμήμα διεύθυνσης. Στην εντολή αυτή ο κώδικας της εντολής είναι '10111', ενώ το τμήμα διεύθυνσης της εντολής είναι '001100101'.



Σχήμα 3.8 Κώδικας εντολής και τμήμα διεύθυνσης

Το σύνολο των εντολών τις οποίες μπορεί να εκτελέσει ένας μικροεπεξεργαστής ονομάζεται ρεπερτόριο εντολών (instruction set) του μικροεπεξεργαστή (στο ρεπερτόριο εντολών μικροεπεξεργαστή θα αναφερθούμε πιο διεξοδικά σε επόμενη παράγραφο του Κεφαλαίου). Κάθε μια από τις εντολές αυτές αντιστοιχίζεται σε ένα δυαδικό αριθμό των κ δυαδικών ψηφίων ο οποίος θα τη συμβολίζει.

Επομένως, ένας μικροεπεξεργαστής μπορεί να διαθέτει έως 2^κ διαφορετικές εντολές. Για παράδειγμα, ένας μικροεπεξεργαστής ο οποίος μπορεί να εκτελέσει εντολές όπως αυτή του προηγούμενου σχήματος, μπορεί να αναγνωρίσει $2^5=32$ διαφορετικές εντολές. Οι εντολές χωρίζονται σε ομάδες, ανάλογα με την εργασία που επιτελούν. Έτσι, υπάρχουν εντολές εκτέλεσης άλματος, εντολές ολίσθησης, εντολές εισόδου-εξόδου κ.λπ.

Το τμήμα διεύθυνσης της εντολής, που έχει μήκος τ είναι ένας δυαδικός αριθμός. Συνήθως, ο αριθμός αυτός παριστάνει μια διεύθυνση της μνήμης. Ανάλογα με την εντολή όμως, μπορεί να παριστάνει ένα καταχωρητή, όπως θα δούμε στη συνέχεια.

Συμβολική γλώσσα

Οι εντολές που μπορεί να εκτελέσει ένας μικροεπεξεργαστής είναι γραμμένες σε αριθμητική μορφή, και μάλιστα στο δυαδικό σύστημα. Ένα πρόγραμμα στη γλώσσα αυτή ονομάζεται πρόγραμμα σε γλώσσα μηχανής (machine language).

Όμως, η συγγραφή προγραμμάτων στη γλώσσα αυτή είναι κουραστική και απαιτεί χρόνο. Τα πράγματα θα ήταν πιο απλά αν μπορούσε κανείς να γράψει τις εντολές με τρόπο συμβολικό, έτσι ώστε να θυμίζει τη λειτουργία τους. Μια γλώσσα που αποτελείται από τέτοιες συμβολικές εντολές ονομάζεται συμβολική γλώσσα (assembly language). Για κάθε εντολή από το ρεπερτόριο εντολών του μικροεπεξεργαστή, υπάρχει και η αντίστοιχη εντολή σε συμβολική γλώσσα.

Για να μπορέσει να εκτελεστεί από το μικροεπεξεργαστή ένα πρόγραμμα σε συμβολική γλώσσα, πρέπει να μεταφραστεί σε γλώσσα μηχανής. Για να πραγματοποιηθεί η μετάφραση αυτή χρησιμοποιείται ένα κατάλληλο πρόγραμμα που ονομάζεται συμβολομεταφραστής (assembler).

Στη συνέχεια της παραγράφου θα αναφερθούμε στα είδη των εντολών που συναντάει κανείς στους μικροεπεξεργαστές, καθώς και στους διαφορετικούς τρόπους με τους οποίους μπορεί να χρησιμοποιηθεί το τμήμα διεύθυνσης μιας εντολής προκειμένου να σχηματιστεί η ενεργή διεύθυνση (active address) από την οποία θα διαβαστεί ένα δεδομένο.

3.3.3 Κύκλοι εντολής και κύκλοι μηχανής

Όπως έχει αναφερθεί, κατά τη λειτουργία του ο μικροεπεξεργαστής ανακαλεί και εκτελεί εντολές ακολουθιακά τη μια μετά την άλλη. Η ανάκληση και εκτέλεση μιας εντολής αποτελεί ένα κύκλο εντολής.

Ένας κύκλος εντολής αποτελείται από μια ή περισσότερες λειτουργίες ανάγνωσης ή εγγραφής στη μνήμη ή στη μονάδα εισόδου-εξόδου. Κάθε αναφορά σε μονάδα εισόδου-εξόδου ή στη μνήμη απαιτεί έναν κύκλο μηχανής.

Συνεπώς κάθε φορά που μια λέξη μνήμης μεταφέρεται προς το μικροεπεξεργαστή ή αντίστροφα, εκτελείται ένας κύκλος μηχανής. Οι πιο συνηθισμένοι κύκλοι που μπορούμε να συναντήσουμε σε ένα μικροεπεξεργαστή είναι οι ακόλουθοι.

- ανάκληση κώδικα (opcode fetch)
- ανάγνωση από τη μνήμη (memory read)
- εγγραφή στη μνήμη (memory write)
- ανάγνωση I/O (I/O read): είσοδος
- εγγραφή I/O (I/O write): έξοδος
- αναγνώριση διακοπής (interrupt acknowledge)
- "άεργος" κύκλος (bus idle)

Ένας κύκλος μηχανής ενεργοποιείται από τη μονάδα ελέγχου του μικροεπεξεργαστή μόλις η μονάδα ελέγχου λάβει τον κώδικα της εντολής. Ο πίνακας 3.3 δείχνει τις τιμές των σημάτων IO/M', RD/WR', για κάθε ένα από τους κύκλους αυτούς.

Η ανάκληση εντολής σε ένα κύκλο εντολής απαιτεί ένα κύκλο μηχανής για κάθε λέξη της εντολής που ανακαλείται. Έτσι, αν οι εντολές ενός μικροεπεξεργαστή έχουν μήκος από 1 ως 3 λέξεις μνήμης η ανάκληση μιας εντολής διαρκεί από 1 ως 3 κύκλους μηχανής αντίστοιχα.

Το πλήθος των κύκλων μηχανής που απαιτούνται για την εκτέλεση μιας εντολής εξαρτάται από την εντολή. Κάποιες εντολές δεν απαιτούν για την εκτέλεσή τους πρόσθετους κύκλους μηχανής από την ανάκληση της εντολής και μετά. Άλλες, χρειάζονται επιπλέον κύκλους για να εκτελέσουν μια πράξη, να γράψουν ή να διαβάσουν στη μνήμη ή σε θύρα εισόδου-εξόδου.

Κύκλος μηχανής	IO/M'	RD/WR'
ανάκληση κώδικα	0	1
ανάγνωση από τη μνήμη	0	1
εγγραφή στη μνήμη	0	0
ανάγνωση I/O: είσοδος	1	1
εγγραφή I/O: έξοδος	1	0
αναγνώριση διακοπής	1	X
'άεργος' κύκλος	0	X

Πίνακας 3.3 Κύκλοι μηχανής τυπικού μικροεπεξεργαστή

3.3.4 Είδη εντολών

Οι εντολές ενός μικροεπεξεργαστή διακρίνονται γενικά στις ακόλουθες κατηγορίες:

Εντολές μεταφοράς δεδομένων.

Οι εντολές αυτές μεταφέρουν δεδομένα μεταξύ καταχωρητών και μνήμης και δεν επηρεάζουν τις σημαίες κατάστασης.

Εντολές αριθμητικών πράξεων

Οι εντολές αυτές πραγματοποιούν αριθμητικές πράξεις σε δεδομένα που βρίσκονται στους καταχωρητές και στη μνήμη. Πρέπει να σημειωθεί ότι η ALU μπορεί να εκτελέσει πράξεις μόνο μεταξύ των καταχωρητών του μικροεπεξεργαστή. Στην περίπτωση που θέλουμε να εκτελεστούν πράξεις μεταξύ αριθμών που βρίσκονται στη μνήμη πρέπει να μεταφερθούν οι αριθμοί αυτοί στους καταχωρητές του μικροεπεξεργαστή. Γενικά, οι εντολές αυτές επηρεάζουν τις σημαίες.

Εντολές λογικών πράξεων

Οι εντολές αυτές υλοποιούν λογικές πράξεις σε δεδομένα που βρίσκονται στους καταχωρητές στη μνήμη και σε σημαίες κατάστασης. Η λογική πράξη επιτελείται συνήθως μεταξύ περισσότερων από ένα bit. Έτσι, με μια εντολή εκτελούνται ταυτόχρονα η λογικές πράξεις. Για παράδειγμα, στο Σχήμα 3.9 φαίνεται η εκτέλεση της λογικής πράξης ΚΑΙ (AND) μεταξύ δύο δυαδικών αριθμών 8 δυαδικών ψηφίων.

Εντολές άλματος

Κανονικά, οι εντολές ενός προγράμματος εκτελούνται σε σειρά ή μια μετά την άλλη. Οι εντολές άλματος αλλάζουν την ροή εκτέλεσης, γι αυτό και ονομάζονται και εντολές διακλάδωσης. Έτσι, όταν εκτελείται μια εντολή άλματος, αντί να εκτελεστεί η επόμενη εντολή, ο έλεγχος μεταφέρεται σε άλλο σημείο του προγράμματος. Το σημείο αυτό μπορεί να είναι οπουδήποτε στο πρόγραμμα, σε διεύθυνση που καθορίζεται από την εντολή άλματος.

Υπάρχουν εντολές άλματος υπό συνθήκη και χωρίς συνθήκη. Οι εντολές άλματος

A 1 0 1 1 1 0 0 1

B 0 1 0 1 0 1 1 1

A AND B 0 0 0 1 0 0 0 1

Σχήμα 3.9 Λογική πράξη AND μεταξύ δύο 8-ψηφίων δυαδικών αριθμών

χωρίς συνθήκη απλά μεταφέρουν την καθορισμένη διεύθυνση στο μετρητή προγράμματος. Οι εντολές άλματος υπό συνθήκη εξετάζουν την κατάσταση κάποιας από τις σημαίες για να καθοριστεί αν πρέπει να μεταφερθεί ή όχι ο έλεγχος. Για παράδειγμα, αν η εντολή άλματος έχει ως συνθήκη (προϋπόθεση) η σημαία κρατούμενου να είναι 1, τότε ο έλεγχος του προγράμματος θα μεταφερθεί μόνο στην περίπτωση που το αποτέλεσμα της τελευταίας πράξης έχει παρουσιάσει υπερχειλίση.

3.4 Τρόποι αναφοράς στη μνήμη

Όπως αναφέραμε, μια εντολή αποτελείται από το τμήμα κώδικα εντολής και το τμήμα διεύθυνσης. Το τμήμα διεύθυνσης χρησιμοποιείται για να βρεθεί η ενεργή διεύθυνση (effective address), δηλαδή η θέση από την οποία θα διαβαστεί το όρισμα (δεδομένο) της εντολής. Οι τρόποι αναφοράς στη μνήμη έχουν σχέση με τους μηχανισμούς που χρησιμοποιούνται για τη διαμόρφωση της ενεργής διεύθυνσης.

Στη συνέχεια περιγράφονται οι κυριότεροι τρόποι αναφοράς στη μνήμη και οι δομές που χρησιμοποιούνται στους περισσότερους μικροεπεξεργαστές για τη διαμόρφωση της ενεργής διεύθυνσης. Στην πράξη, δεν υποστηρίζονται όλοι οι τρόποι αναφοράς στη μνήμη από όλους τους μικροεπεξεργαστές. Ακόμη, είναι δυνατό διαφορετικοί μικρο-επεξεργαστές να αναφέρονται στον ίδιο τρόπο αναφοράς στη μνήμη χρησιμοποιώντας διαφορετικά ονόματα.

Άμεση αναφορά στη μνήμη (Immediate addressing)

Στην άμεση αναφορά στη μνήμη η τιμή του ορίσματος (ή του δεδομένου) είναι αποθηκευμένη σε κάποιο τμήμα της εντολής (η εντολή αποτελείται από μια ή

περισσότερες λέξεις και το όρισμα είτε είναι τμήμα της εντολής είτε ακολουθεί μετά τον κωδικό εντολής). Και ο κωδικός αλλά και το όρισμα της εντολής καλούνται από τη μνήμη χρησιμοποιώντας το μετρητή προγράμματος.

Η άμεση αναφορά στη μνήμη δεν είναι στην πραγματικότητα 'αναφορά στη μνήμη', εφόσον δεν απαιτείται επιπλέον προσπέλαση στη μνήμη (πέραν της ανάγνωσης της εντολής) για να βρεθεί το όρισμα.

Στο Σχήμα 3.10 φαίνεται ο τρόπος με τον οποίο αποθηκεύεται στη μνήμη μια εντολή η οποία χρησιμοποιεί άμεση αναφορά στη μνήμη.



Σχήμα 3.10 Άμεση αναφορά στη μνήμη

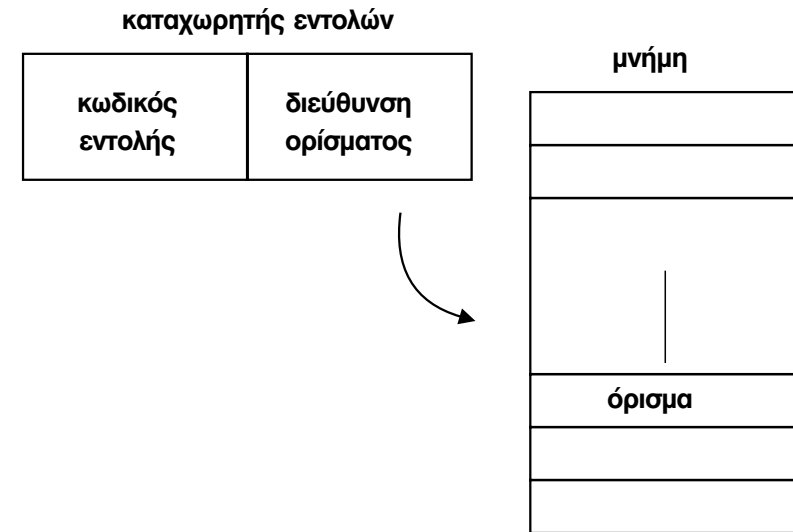
Για παράδειγμα, η εντολή 'ADD 30h' θα μπορούσε να σημαίνει 'πρόσθεσε στην τρέχουσα τιμή του συσσωρευτή το δεκαεξαδικό αριθμό 30' (30 hexadecimal).

Απευθείας αναφορά στη μνήμη (Direct addressing)

Στην απευθείας αναφορά στη μνήμη η διεύθυνση της μνήμης στην οποία θα βρεθεί το δεδομένο περιέχεται μέσα στην εντολή αμέσως μετά τον κωδικό της εντολής. Αυτός ο τρόπος αναφοράς είναι πολύ γρήγορος, αλλά έχει το μειονέκτημα ότι ο αριθμός των λέξεων που μπορούν να διευθυνσιοδοτηθούν περιορίζεται από τον αριθμό των δυαδικών ψηφίων του πεδίου διεύθυνσης της εντολής.

Για παράδειγμα, η εντολή 'ADD [30h]³' θα μπορούσε να σημαίνει 'πρόσθεσε στην τρέχουσα τιμή του συσσωρευτή την τιμή που βρίσκεται στη διεύθυνση μνήμης 30'.

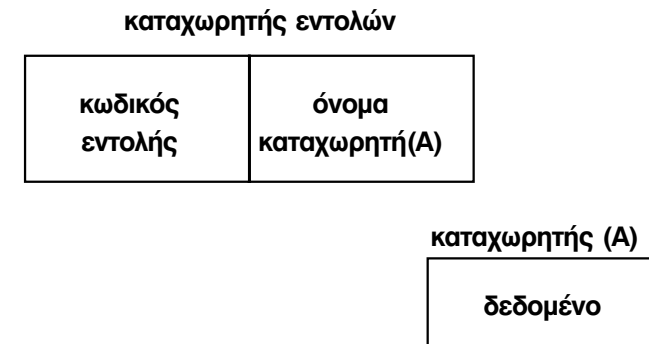
3 Σε πολλές συμβολικές (assembly) γλώσσες χρησιμοποιούνται οι αγκύλες [] για να συμβολίσουν το περιεχόμενο μιας θέσης μνήμης ή ενός καταχωρητή.



Σχήμα 3.11 Απευθείας αναφορά στη μνήμη

Αναφορά στη μνήμη καταχωρητών (register addressing)

Με αυτόν τον τρόπο το δεδομένο της εντολής περιέχεται σε έναν εσωτερικό καταχωρητή του επεξεργαστή.



Σχήμα 3.12 Αναφορά στη μνήμη μέσω καταχωρητή

Για παράδειγμα, η εντολή 'ADD A' θα μπορούσε να σημαίνει 'πρόσθεσε στην τρέχουσα τιμή του συσσωρευτή την τιμή του καταχωρητή A'.

Έμμεση αναφορά μέσω καταχωρητή (register indirect addressing)

Στην έμμεση αναφορά μέσω καταχωρητή, ο κώδικας της εντολής προσδιορίζει ένα καταχωρητή του μικροεπεξεργαστή που περιέχει τη διεύθυνση που θα χρησιμοποιηθεί για την προσπέλαση του ορίσματος της μνήμης.



Σχήμα 3.13 Έμμεση αναφορά μέσω καταχωρητή

Για παράδειγμα, η εντολή 'ADD [A]' θα μπορούσε να σημαίνει 'πρόσθεσε στην τρέχουσα τιμή του συσσωρευτή την τιμή που βρίσκεται στη θέση μνήμης, της οποίας η διεύθυνση βρίσκεται στον καταχωρητή A'.

3.5 Χαρακτηριστικά και κατηγορίες μικροεπεξεργαστών

Στην παράγραφο αυτή θα αναφερθούμε στα κυριότερα χαρακτηριστικά των μικροεπεξεργαστών. Τα χαρακτηριστικά αυτά είναι

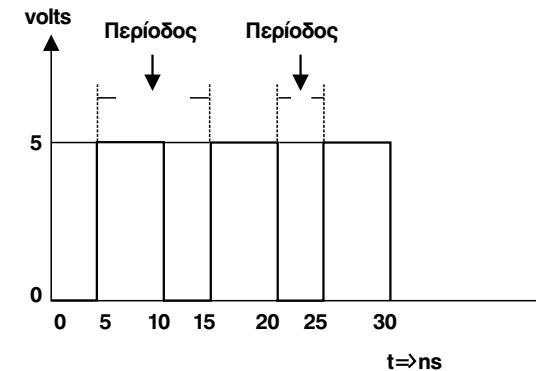
- η συχνότητα λειτουργίας (operating frequency)
- το μήκος λέξης (word length)
- το ρεπερτόριο εντολών (instruction set)

Συχνότητα λειτουργίας μικροεπεξεργαστή

Για να εκτελέσει μια εντολή ο μικροεπεξεργαστής, εκτελεί όπως είδαμε μια σειρά από διαδοχικές λειτουργίες. Κάθε μία από τις λειτουργίες αυτές διαρκεί ένα χρονικό

διάστημα. Για το συγχρονισμό των λειτουργιών αυτών, είναι απαραίτητο κάποιο ρολόι. Σε κάθε "κτύπο" του ρολογιού ο μικροεπεξεργαστής εκτελεί μία στοιχειώδη λειτουργία.

Το ρολόι είναι ένα σήμα το οποίο παράγεται από ένα εξωτερικό κύκλωμα (το οποίο ονομάζεται ταλαντωτής) και η τάση του ως συνάρτηση του χρόνου φαίνεται στο Σχήμα 3.14.



Σχήμα 3.14 Κυματομορφή ρολογιού με περίοδο 10 ns

Το σήμα του ρολογιού, εναλλάσσεται μεταξύ της στάθμης των 5 Volts και αυτής των 0 Volt. Λόγω του σχήματος του το σήμα αυτό ονομάζεται τετραγωνική παλμοσειρά. Το χρονικό διάστημα μεταξύ δύο διαδοχικών εναλλαγών του σήματος, για παράδειγμα από τα 0 Volt στα 5 Volts και πάλι στα 0 Volt είναι ίσο με μια ημιπερίοδο του ρολογιού. Δύο διαδοχικές ημιπερίοδοι αποτελούν μια περίοδο ή έναν κύκλο του ρολογιού.

Η συχνότητα του ρολογιού δείχνει το πλήθος των κύκλων του ρολογιού στη διάρκεια ενός δευτερολέπτου. Για τους σύγχρονους μικροεπεξεργαστές το μέγεθος αυτό είναι της τάξης των εκατοντάδων MHz. Το ένα MHz είναι ίσο με ένα εκατομμύριο κύκλους το δευτερόλεπτο. Έτσι, αν ένας μικροεπεξεργαστής έχει περίοδο 10 nsec, η συχνότητα λειτουργίας του είναι $1/(10 \times 10^{-9}) = 10^8 = 100\text{MHz}$.

Κάθε μικροεπεξεργαστής είναι σχεδιασμένος να λειτουργεί μέχρι κάποια μέγιστη συχνότητα. Συνήθως η συχνότητα του ρολογιού, επιλέγεται να είναι ίση με τη μέγιστη επιτρεπόμενη συχνότητα λειτουργίας του μικροεπεξεργαστή και αυτό γιατί όσο μεγαλύτερη είναι η συχνότητα του ρολογιού τόσο πιο γρήγορα εκτελεί τις εντολές ο μικροεπεξεργαστής. Η συχνότητα του ρολογιού ονομάζεται συχνότητα λειτουργίας του μικρο-επεξεργαστή.

Εάν προσπαθήσουμε να υπερβούμε τη μέγιστη συχνότητα λειτουργίας, το αποτέλεσμα θα είναι είτε η λανθασμένη λειτουργία του ολοκληρωμένου είτε ακόμα και η καταστροφή του ολοκληρωμένου κυκλώματος του μικροεπεξεργαστή.

Ο ρυθμός με τον οποίο εκτελούνται οι εντολές είναι συνάρτηση της συχνότητας λειτουργίας του μικροεπεξεργαστή. Για την ακρίβεια, η συχνότητα του ρολογιού

δείχνει το πλήθος των κύκλων μηχανής που εκτελούνται σε ένα δευτερόλεπτο. Έτσι, αν το ρολόι του μικροεπεξεργαστή είναι 100 MHz (100.000.000 κύκλους το δευτερόλεπτο) αυτό σημαίνει ότι εκτελούνται 100.000.000 κύκλοι μηχανής το δευτερόλεπτο.

Όπως έχουμε αναφέρει, ένα κύκλος εντολής αποτελείται από (περισσότερους από ένα) κύκλους μηχανής. Έτσι, αν για παράδειγμα μια εντολή πρόσθεσης χρειάζεται 5 κύκλους μηχανής για να ανακληθεί και να εκτελεστεί και η συχνότητα λειτουργίας του μικροεπεξεργαστή είναι 100 MHz (δηλαδή κάθε κύκλος μηχανής διαρκεί 10 ns) τότε η εντολή της πρόσθεσης χρειάζεται συνολικά $5 \cdot 10 \text{ ns} = 50 \text{ ns}$ για να εκτελεστεί. Συνεπώς είναι δυνατό να εκτελούνται 20.000.000 εκατομμύρια προσθέσεις το δευτερόλεπτο.

Μήκος λέξης (word length) μικροεπεξεργαστή

Ένας μικροεπεξεργαστής μπορεί να κάνει αριθμητικές ή λογικές πράξεις μεταξύ των καταχωρητών του. Συνεπώς κάθε φορά τα δεδομένα εισάγονται από τη μνήμη ή από τις περιφερειακές μονάδες σε κάποιο καταχωρητή του μικροεπεξεργαστή και μετά πραγματοποιείται η επεξεργασία τους.

Ένα σημαντικό χαρακτηριστικό των καταχωρητών του μικροεπεξεργαστή (αλλά και του ίδιου του μικροεπεξεργαστή) είναι το εύρος των καταχωρητών σε δυαδικά ψηφία bits. Όσο πιο "μεγάλος" είναι ένας καταχωρητής τόσο "περισσότερα δεδομένα χωράει".

Για παράδειγμα, έστω ότι θέλουμε να εκτελέσουμε μια πρόσθεση χρησιμοποιώντας ένα μικροεπεξεργαστή με τρεις καταχωρητές, τους A, B, C. Για να εκτελέσει ο μικροεπεξεργαστής την πρόσθεση θα πρέπει πρώτα να μεταφερθούν σε δύο από τους καταχωρητές του οι δύο αριθμοί. Έστω ότι για να αποθηκεύσουμε καθένα από τους αριθμούς χρειαζόμαστε 16 bits. Αν ο μικροεπεξεργαστής διαθέτει καταχωρητές των 16 bits τότε η αποθήκευση των δεδομένων θα γίνει στους καταχωρητές A και B. Η πρόσθεση θα γίνει με μια μόνο εντολή, και το αποτέλεσμα θα αποθηκευτεί στον καταχωρητή C. Η μεταφορά του αποτελέσματος από το μικροεπεξεργαστή στη μνήμη ή σε κάποια περιφερειακή μονάδα χρειάζεται επίσης μόνο μια εντολή.

Αν όμως ο μικροεπεξεργαστής που χρησιμοποιούμε διαθέτει καταχωρητές των 8 bits τότε η αποθήκευση του κάθε αριθμού θα χρειαστεί τουλάχιστον δύο βήματα.

1ο Βήμα

Στο πρώτο βήμα θα πρέπει να βάλουμε στον καταχωρητή A τα 8 χαμηλής τάξης δυαδικά ψηφία του πρώτου αριθμού, ενώ στον καταχωρητή B τα 8 χαμηλής τάξης δυαδικά ψηφία του δεύτερου αριθμού. Το αποτέλεσμα της πρόσθεσης αποθηκεύεται στον καταχωρητή C. Αν έχει προκύψει κρατούμενο κατά την πρόσθεση των δύο καταχωρητών, τότε πρέπει να τεθεί η σημαία κρατουμένου της ALU. Τέλος το αποτέλεσμα αποθηκεύεται στη μνήμη.

2ο Βήμα

Στο δεύτερο βήμα, στους καταχωρητές A και B του μικροεπεξεργαστή θα φορτωθούν τα υπόλοιπα υψηλότερης αξίας δυαδικά ψηφία των αριθμών.

Στην πρόσθεση τώρα πρέπει να συμπεριλάβουμε και το κρατούμενο που έχει προκύψει από την προηγούμενη πρόσθεση. Συνεπώς, αν έχει προκύψει κρατούμενο, θα πρέπει να προστεθεί άλλη μια μονάδα στο άθροισμα που προκύπτει από την πρόσθεση των καταχωρητών A και B που περιέχουν τα υψηλότερης αξίας δεδομένα. Το αποτέλεσμα αυτής της πρόσθεσης αποθηκεύεται στον καταχωρητή C.

Έτσι, για την εκτέλεση της πρόσθεσης απαιτούνται δύο εντολές.

Από το παράδειγμα που αναφέραμε, μπορεί κανείς να συμπεράνει ότι το μήκος των καταχωρητών του μικροεπεξεργαστή επηρεάζει την ταχύτητα με την οποία εκτελούνται οι πράξεις στο μικροεπεξεργαστή. Όσο πιο μεγάλοι είναι οι καταχωρητές του, τόσο πιο γρήγορα κατά κανόνα εκτελούνται τα προγράμματα.

Συνήθως οι καταχωρητές έχουν ίδιο μήκος με το μήκος σε bits των υπόλοιπων εσωτερικών μονάδων του μικροεπεξεργαστή, όπως η ALU, το οποίο λέμε και μήκος λέξης του μικροεπεξεργαστή. Η παράμετρος αυτή είναι αρκετά σημαντική για την απόδοση του συστήματος.

Οι μικροεπεξεργαστές που χρησιμοποιούνται στην πράξη, χωρίζονται σε τρεις κατηγορίες:

- τους μικροεπεξεργαστές με μέγεθος καταχωρητή 8 δυαδικά ψηφία (αναφερόμαστε στους μικροεπεξεργαστές αυτούς με τον όρο '8-μπιτοι μικροεπεξεργαστές')
- τους μικροεπεξεργαστές με μέγεθος καταχωρητή 16 δυαδικά ψηφία (αναφερόμαστε στους μικροεπεξεργαστές αυτούς με τον όρο '16-μπιτοι μικροεπεξεργαστές')
- τους μικροεπεξεργαστές με μέγεθος καταχωρητή 32 δυαδικά ψηφία (αναφερόμαστε στους μικροεπεξεργαστές αυτούς με τον όρο '32-μπιτοι μικροεπεξεργαστές')

Στον Πίνακα 3.4 παρουσιάζουμε το μεγαλύτερο ακέραιο αριθμό που μπορούμε να αποθηκεύσουμε σε έναν καταχωρητή ανάλογα με το εύρος του (θεωρώντας ότι παριστάνουμε μόνο θετικούς αριθμούς).

Εύρος	δυαδική παράσταση	Ακέραια τιμή
8	1111 1111	$2^8 - 1 = 255$
16	1111 1111 1111 1111	$2^{16} - 1 = 65535$
32	1111 1111 1111 1111 1111 1111 1111 1111	$2^{32} - 1 = 4.294.967.295$

Πίνακας 3.4 Εύρος καταχωρητή (σε δυαδικά ψηφία) και μέγιστος αριθμός που μπορεί να χωρέσει.

Ρεπερτόριο εντολών

Ένα άλλο σημαντικό χαρακτηριστικό ενός μικροεπεξεργαστή είναι το ρεπερτόριο των εντολών που διαθέτει. Με τον όρο ρεπερτόριο εντολών (instruction set) ενός μικροεπεξεργαστή, αναφερόμαστε στις εντολές που μπορεί να εκτελέσει.

Το ρεπερτόριο εντολών ενός μικροεπεξεργαστή είναι καθοριστικός παράγοντας για να αποφασίσουμε αν είναι κατάλληλος για να χρησιμοποιηθεί σε μια εφαρμογή. Για παράδειγμα, σε μια εφαρμογή ψηφιακής επεξεργασίας της ανθρώπινης φωνής, ο μικροεπεξεργαστής θα πρέπει να διαθέτει εντολές που εκτελούν γρήγορα μαθηματικές πράξεις. Αντιθέτως, για να χρησιμοποιηθεί ένας μικροεπεξεργαστής σε μια εφαρμογή όπως η μεταγωγή κλήσεων σε ένα ψηφιακό τηλεφωνικό κέντρο, πρέπει να διαθέτει εντολές για γρήγορη μεταφορά δεδομένων από τη μνήμη και τις περιφερειακές μονάδες.

Ένα άλλο σημαντικό χαρακτηριστικό των εντολών ενός μικροεπεξεργαστή είναι η **συμβατότητα με παλαιότερους μικροεπεξεργαστές**. Η πείρα έδειξε ότι ένας νέος μικροεπεξεργαστής είναι καλό να μπορεί να εκτελεί προγράμματα που εκτελούνταν σε παλαιότερους μικροεπεξεργαστές. Με άλλα λόγια το ρεπερτόριο των εντολών ενός νέου μικροεπεξεργαστή, θα έπρεπε να περιέχει όλες τις εντολές του προηγούμενου μικροεπεξεργαστή της ίδιας εταιρείας. Με αυτό τον τρόπο, τα προγράμματα που έχουν γραφτεί για ένα παλαιότερο μικροεπεξεργαστή δε χρειάζεται να ξαναγραφτούν από την αρχή.

Με βάση το κριτήριο του μεγέθους του ρεπερτορίου εντολών, οι μικροεπεξεργαστές διακρίνονται σε δύο κατηγορίες. Τους μικροεπεξεργαστές διευρυμένου ρεπερτορίου εντολών και τους **μικροεπεξεργαστές μειωμένου ρεπερτορίου εντολών**. Συχνότατα στην πράξη, αναφερόμαστε στα μικροεπεξεργαστικά συστήματα που χρησιμοποιούν τους μικροεπεξεργαστές αυτούς. Έτσι, μιλάμε για **υπολογιστικά συστήματα διευρυμένου ρεπερτορίου εντολών (complex instruction set computers, CISC)** και για **υπολογιστικά συστήματα μειωμένου ρεπερτορίου εντολών (reduced instruction set computers RISC)**.

Είναι σαφές ότι το ρεπερτόριο εντολών ενός μικροεπεξεργαστή διευρυμένου ρεπερτορίου εντολών περιλαμβάνει πιο πολύπλοκες εντολές, επομένως μπορεί με λιγότερες εντολές να εκτελέσει πιο πολύπλοκες διαδικασίες. Οι μικροεπεξεργαστές που σχεδιάζονταν αρχικά ανήκαν στην κατηγορία αυτή. Στη φάση αυτή, οι σχεδιαστές προσπαθούσαν να σχεδιάζουν μικροεπεξεργαστές που μπορούσαν να εκτελέσουν όλο και πιο πολύπλοκες εντολές, επομένως θα ήταν και πιο ισχυροί.

Με την πάροδο όμως του χρόνου και ύστερα από μελέτες, προέκυψαν τα ακόλουθα συμπεράσματα:

- Όσο μεγαλύτερο είναι το πλήθος του ρεπερτορίου εντολών ενός μικροεπεξεργαστή, τόσο πιο πολύπλοκη είναι η σχεδίασή του, επομένως τόσο περισσότερο καθυστερεί η εκτέλεση κάθε εντολής.
- Κατά την εκτέλεση ενός τυπικού προγράμματος, μόνο ένα μικρό μέρος των εντολών χρησιμοποιείται συχνότατα, ενώ το μεγαλύτερο πλήθος των (πολύπλοκων συνήθως) εντολών χρησιμοποιείται σπάνια.

Ξεκίνησε λοιπόν μια νέα αντίληψη στην κατεύθυνση της σχεδίασης μικροεπεξεργαστών. Στην κατεύθυνση αυτή άρχισαν να σχεδιάζονται

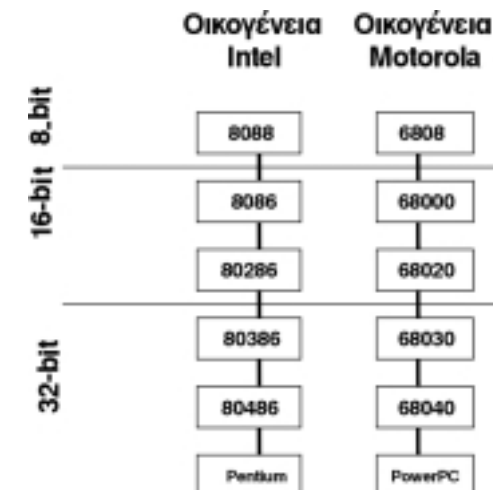
μικροεπεξεργαστές με όσο το δυνατό πιο μικρό σύνολο εντολών, κάθε μια από τις οποίες εκτελείται όσο το δυνατό πιο γρήγορα. Έτσι, η σχεδίαση του μικροεπεξεργαστή απλοποιείται σημαντικά, και κάθε εντολή μπορεί να εκτελεστεί πολύ πιο γρήγορα από ότι μια εντολή ενός μικροεπεξεργαστή διευρυμένου ρεπερτορίου εντολών. Οι μικροεπεξεργαστές που σχεδιάστηκαν σύμφωνα με τη φιλοσοφία αυτή είναι οι μικροεπεξεργαστές μειωμένου ρεπερτορίου εντολών (Reduced Instruction Set Computers, RISC).

3.6 Οικογένειες μικροεπεξεργαστών

Με τον όρο **οικογένεια μικροεπεξεργαστών** εννοούμε μια σειρά επεξεργαστών που έχουν κατασκευαστεί από την ίδια εταιρεία και είναι συμβατοί μεταξύ τους. Με τον όρο **συμβατότητα**, εννοούμε ότι ένα νεότερο μέλος της οικογένειας μπορεί να εκτελέσει εντολές ενός προηγούμενου (αρχαιότερου) μέλους.

Η μεγάλη πλειοψηφία των επεξεργαστών που κυκλοφορούν σήμερα στην αγορά ανήκει σε μια από τις δύο οικογένειες επεξεργαστών: την οικογένεια επεξεργαστών της Intel της σειράς 80X86 και την οικογένεια επεξεργαστών της σειράς 68000 της εταιρείας Motorola.

Κάθε μια από τις οικογένειες αυτές έχει μέλη που ανήκουν στην κατηγορία των 8-bit μικροεπεξεργαστών, στην κατηγορία των 16-bit μικροεπεξεργαστών και 32-bit μικροεπεξεργαστές. Στο Σχήμα 3.15 φαίνεται η εξέλιξη αυτή για τις δύο αυτές οικογένειες.



Σχήμα 3.15 Εξέλιξη μικροεπεξεργαστών INTEL και MOTOROLA

Ο πίνακας 3.5 περιγράφει τα τεχνικά χαρακτηριστικά των μικροεπεξεργαστών της εταιρείας INTEL.

μ/Ε	Μήκος λέξης	Πλάτος Διαδρόμου	
	Επεξεργαστή	Δεδομένων	Διευθύνσεων
8088	16-bit	8-bit	20-bit
8086	16-bit	16-bit	20-bit
80286	16-bit	16-bit	24-bit
80386	32-bit	16-bit	24-bit
80486	32-bit	32-bit	32-bit
Pentium	32-bit	64-bit	32-bit

Πίνακας 3.5 Μικροεπεξεργαστές της εταιρείας INTEL

3.7 Μικροελεγκτές

Όπως είδαμε, ένα μικροϋπολογιστικό σύστημα αποτελείται από ένα μικροεπεξεργαστή, μνήμη και μονάδες εισόδου - εξόδου.

Ένας μικροεπεξεργαστής μπορεί να εκτελεί υπολογισμούς και να επεξεργάζεται δεδομένα γρήγορα και αξιόπιστα και προκειμένου να επικοινωνήσει με το περιβάλλον του πρέπει να συνδεθεί με κάποιες περιφερειακές μονάδες, όπως:

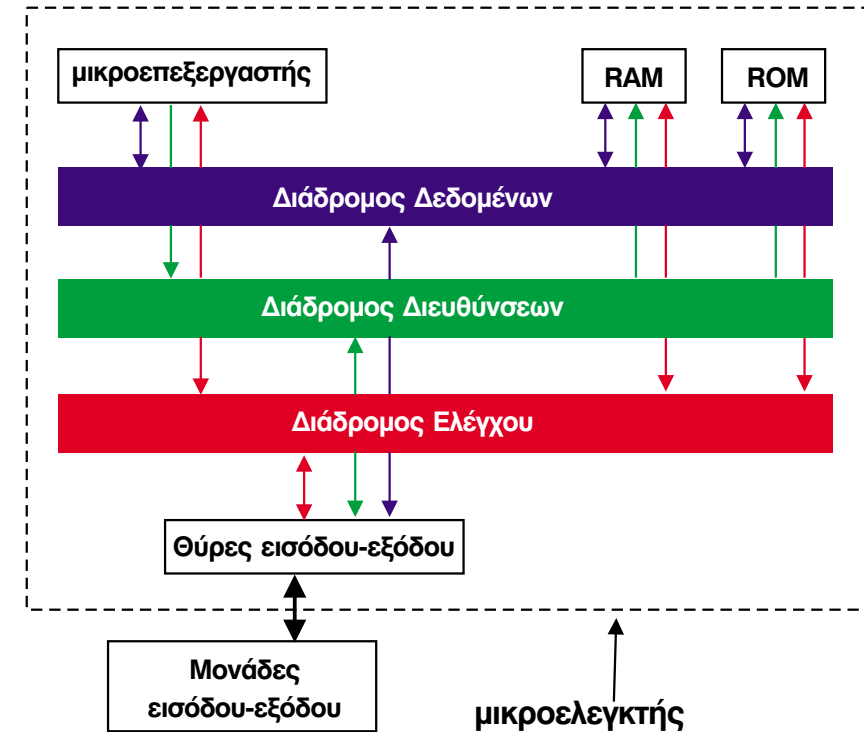
- μνήμη (RAM και ROM),
- ελεγκτής διακοπών (προκειμένου να δεχτεί και να αναγνωρίσει τα σήματα διακοπών από περιφερειακές μονάδες)
- θύρες (ports) επικοινωνίας με τις μονάδες εισόδου-εξόδου

Ακόμη, ένα μικροϋπολογιστικό σύστημα μπορεί να χρησιμοποιηθεί για να ελέγξει ένα σύστημα (πχ. εργοστασιακό συγκρότημα, βιομηχανικό αυτοματισμό κ.λπ.). Στην περίπτωση αυτή, πρέπει να δεχτεί αναλογικά σήματα (τιμές θερμοκρασίας, πίεσης κ.λπ.). Προκειμένου να αξιοποιήσει ένας μικροεπεξεργαστής τα σήματα αυτά πρέπει να παρεμβληθούν εξαρτήματα τα οποία να μπορούν να μετατρέψουν τις τιμές των σημάτων αυτών από αναλογικές σε ακολουθίες δυαδικών ψηφίων τις οποίες μπορεί να επεξεργαστεί ο μικροεπεξεργαστής. Οι συσκευές αυτές ονομάζονται αναλογικό-ψηφιακοί μετατροπείς (Analog-to-Digital Converters, A/D).

Έτσι, η σχεδίαση ενός μικροϋπολογιστικού συστήματος περιλαμβάνει, εκτός από το ολοκληρωμένο κύκλωμα του μικροεπεξεργαστή και τις περιφερειακές συσκευές, η σύνδεση και ο συγχρονισμός των οποίων καθιστά πολύπλοκη τη σχεδίαση του ολοκληρωμένου συστήματος.

Μια λύση στο πρόβλημα θα ήταν να υπήρχε ένα ολοκληρωμένο κύκλωμα το οποίο να

περιλαμβάνει κάποιες από τις απαραίτητες περιφερειακές συσκευές. Ένα τέτοιο κύκλωμα ονομάζεται μικροελεγκτής (microcontroller) και το σχηματικό του διάγραμμα φαίνεται στο Σχήμα 3.16.



Σχήμα 3.16 Σχέση μικροεπεξεργαστή και μικροελεγκτή

Σύμφωνα με τα παραπάνω, οι μικροελεγκτές είναι κατάλληλοι για εφαρμογές στις οποίες υπάρχει αυξημένη ανάγκη για χρήση περιφερειακών συσκευών. Πολλές φορές, το κριτήριο επιλογής ενός μικροελεγκτή είναι το είδος και οι δυνατότητες των περιφερειακών που διαθέτει.

Οι πιο γνωστές από τις περιφερειακές μονάδες που χρειάζονται για να λειτουργήσει ένα μικροϋπολογιστικό σύστημα (και οι οποίες περιλαμβάνονται στο ολοκληρωμένο κύκλωμα ενός μικροεπεξεργαστή) είναι οι εξής:

- μνήμη (RAM και ROM)
- θύρες εισόδου / εξόδου, καθώς και θύρες σειριακής και παράλληλης επικοινωνίας
- μονάδα αναγνώρισης διακοπών
- μετατροπείς αναλογικού σήματος σε ψηφιακό και αντίστροφα
- μετρητές χρόνου-χρονιστές (timers)

Στα επόμενα Κεφάλαια θα αναφερθούμε διεξοδικά στη χρησιμότητα κάθε μιας από τις μονάδες αυτές καθώς και στον τρόπο με τον οποίο μπορούμε να τις χρησιμοποιήσουμε σε ένα μικροελεγκτή.

ΕΡΩΤΗΣΕΙΣ

Ποια τα βασικά τμήματα ενός υπολογιστικού συστήματος και ποια η λειτουργία του καθενός από αυτά;
 Από ποια τμήματα αποτελείται ένας μικροεπεξεργαστής;
 Ποιοι είναι οι βασικοί καταχωρητές ενός μικροεπεξεργαστή;
 Ποια τα είδη εντολών ενός τυπικού μικροεπεξεργαστή;
 Περιγράψτε τα στάδια εκτέλεσης μιας εντολής σε ένα τυπικό μικροεπεξεργαστή.
 Σε τι διαφέρει η συμβολική γλώσσα από τη γλώσσα μηχανής;
 Αναφέρετε τους τρόπους διεθνισιοδότησης της μνήμης
 Αναφέρετε τα χαρακτηριστικά ως προς τα οποία διακρίνουμε τους μικροεπεξεργαστές. Σε ποια είδη διακρίνουμε τους μικρο-επεξεργαστές ως προς καθένα από αυτά;
 Ποιες είναι οι πιο γνωστές οικογένειες επεξεργαστών; ποια τα μέλη κάθε μιας από αυτές;
 Σε τι διαφέρει ένας μικροελεγκτής από ένα μικροεπεξεργαστή;
 Γιατί οι μικροελεγκτές είναι χρήσιμοι στο σχεδιασμό μικρο-υπολογιστικών συστημάτων;

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Κ.Ζ. Πεκμεστζή**, "Συστήματα Μικροϋπολογιστών", Εκδόσεις Συμμετρία, 1995.
2. **Γ. Παπακωνσταντίνου, Π. Τσανάκας, Ν. Κοζύρης, Α. Μανουσοπούλου, Π. Ματζάκος**, "Τεχνολογία Υπολογιστικών Συστημάτων και Λειτουργικά Συστήματα". ΥΠΕΠΘ, Παιδαγωγικό Ινστιτούτο, 1999.
3. **S. Mueller**, "Upgrading and Repairing PCs", Scott Mueller, 10th edition, Que Corporation, 1998.
4. **Κ. Δημόπουλος, Α. Παρασκευόπουλος**, "80X86: Αρχιτεκτονική, Σχεδίαση και Προγραμματισμός", Παπασωτηρίου, 1992.

Κεφάλαιο 4ο

ΣΥΝΔΕΣΗ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ ΚΑΙ ΜΙΚΡΟΕΛΕΓΚΤΩΝ

Περιεχόμενα

- 4.1 Ακροδέκτες και συνδέσεις μικροεπεξεργαστών και μικροελεγκτών
- 4.2 Προσπέλαση Συσκευών εισόδου-εξόδου
- 4.3 Διακοπές
- 4.4 Λειτουργία Απευθείας Προσπέλασης μνήμης
- 4.5 Είσοδος και έξοδος ψηφιακών δεδομένων σε μικροεπεξεργαστή

4.1 Ακροδέκτες και συνδέσεις μικροεπεξεργαστών και μικροελεγκτών

Όταν ολοκληρώσεις το κεφάλαιο αυτό θα μπορείς να...

αναφέρεις τις συνδέσεις των μικροεπεξεργαστών και μικροελεγκτών
 αναφέρεις τι είναι η πολυπλεξία ακροδεκτών και ποια η χρησιμότητά της
 περιγράφεις τη διαδικασία πολυπλεξίας χρησιμοποιώντας ψηφιακά ηλεκτρονικά
 αναφέρεις τους τρόπους διευθυνσιοδότησης συσκευών εισόδου-εξόδου
 συγκρίνεις τους τρόπους διευθυνσιοδότησης συσκευών εισόδου-εξόδου
 σχεδιάζεις μια θύρα εισόδου-εξόδου χρησιμοποιώντας ψηφιακά ηλεκτρονικά
 περιγράφεις τους τρόπους προσπέλασης περιφερειακών συσκευών
 αναφέρεις τους τρόπους με τους οποίους μπορεί να εξυπηρετήσει ένας μικροεπεξεργαστής μια μονάδα εισόδου / εξόδου
 περιγράφεις τη διαδικασία εξυπηρέτησης της συσκευής με κάθε ένα από τους τρόπους αυτούς
 περιγράφεις τη διαδικασία εξυπηρέτησης περιφερειακών συσκευών με τη μέθοδο των διακοπών
 αναφέρεις τα πλεονεκτήματα της χρήσης διακοπών για την εξυπηρέτηση περιφερειακών συσκευών
 περιγράφεις τη διαδικασία μεταφοράς δεδομένων μεταξύ μιας περιφερειακής συσκευής και της μνήμης χρησιμοποιώντας τη μέθοδο της απευθείας προσπέλασης μνήμης (DMA)
 αναφέρεις τα πλεονεκτήματα της μεταφοράς δεδομένων μεταξύ μιας περιφερειακής συσκευής και της μνήμης χρησιμοποιώντας τη μέθοδο της απευθείας προσπέλασης μνήμης

Όπως έχει αναφερθεί, ο μικροεπεξεργαστής είναι ένα ολοκληρωμένο κύκλωμα. Το κύκλωμα αυτό συνδέεται μέσω ακροδεκτών (pins) σε μια πλακέτα ολοκληρωμένων κυκλωμάτων (integrated circuit board). Οι ακροδέκτες αυτοί χρησιμοποιούνται για:

- την επικοινωνία με τα υπόλοιπα ολοκληρωμένα κυκλώματα του υπολογιστικού συστήματος
- την παροχή τροφοδοσίας και γείωσης

Μερικοί από τους ακροδέκτες που συναντάμε στην πλειοψηφία των μικροεπεξεργαστών καθώς και η λειτουργικότητά τους δίνονται στον Πίνακα 4.1.

Ακροδέκτης	I/O	Σημασία
GND	I	Γη
VCC	I	Τάση τροφοδοσίας (+5V)
NMI	I	Non-Maskable Interrupt, αίτηση για διακοπή η οποία δε μπορεί να παρεμποδιστεί
INT	I	Αίτηση για διακοπή με μάσκα (μπορεί να παρεμποδιστεί)
CLK	I	
RESET	I	Τερματισμός δραστηριότητας του μικροεπεξεργαστή.
RD	O	read, ένδειξη ότι πρόκειται να εκτελεστεί μια ανάγνωση μνήμης ή περιφερειακής συσκευής
WR	O	write, ένδειξη ότι πρόκειται να εκτελεστεί εγγραφή στη μνήμη ή σε περιφερειακή συσκευή
A[0:n-1]	O	Οι ακροδέκτες του διαδρόμου διευθύνσεων. Ο μικροεπεξεργαστής μπορεί να διευθυνσιοδοτήσει μέχρι 2 ⁿ διαφορετικές διευθύνσεις.
D[0:k-1]	I/O	k ακροδέκτες του διαδρόμου δεδομένων

Πίνακας 4.1 Ακροδέκτες τυπικού μικροεπεξεργαστή

Εκτός από τα παραπάνω σήματα, είναι δυνατό να υπάρχει ένα σήμα εξόδου από το μικροεπεξεργαστή με το όνομα IO/M' (Input-Output/ Memory). Το σήμα αυτό δείχνει αν θα εκτελεστεί λειτουργία (ανάγνωσης ή εγγραφής) στη μνήμη ή σε συσκευή εισόδου-εξόδου. Το σήμα IO/M' χρησιμοποιείται μόνο αν ο μικροεπεξεργαστής χρησιμοποιεί ξεχωριστό χώρο διευθύνσεων εισόδου-εξόδου από εκείνο της μνήμης, όπως θα εξηγήσουμε σε επόμενη παράγραφο.

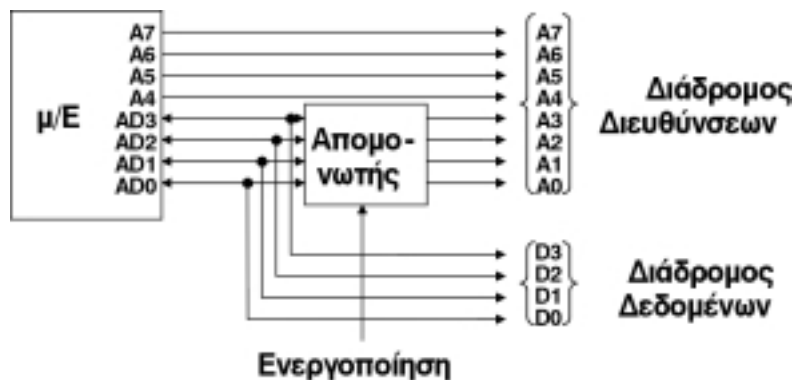
4.1.1 Πολυπλεξία διαδρόμων

Το πλήθος των ακροδεκτών του μικροεπεξεργαστή συχνά καθορίζει το κόστος κατασκευής του ολοκληρωμένου κυκλώματος, αλλά και ολόκληρου του συστήματος.

Για το λόγο αυτό, προκειμένου να μειωθεί το κόστος κατασκευής του ολοκληρωμένου κυκλώματος προσπαθούμε να μειώσουμε το πλήθος των ακίδων του. Όπως είδαμε, η πλειοψηφία των γραμμών εισόδου και εξόδου του μικροεπεξεργαστή είναι οι γραμμές διαδρόμων και διευθύνσεων.

Μια καλή ιδέα για να μειωθεί το πλήθος των ακροδεκτών του μικροεπεξεργαστή, είναι κάποιιοι (ή και όλοι) οι ακροδέκτες που χρησιμοποιούνται για το διάδρομο διευθύνσεων να χρησιμοποιούνται και για το διάδρομο δεδομένων. Για να γίνει αυτό πρέπει να ληφθεί μέριμνα κατά τη σχεδίαση του μικροεπεξεργαστή, ώστε να μην υπάρχουν ταυτόχρονα στους ακροδέκτες αυτούς και διευθύνσεις και δεδομένα. Η τεχνική με τη βοήθεια της οποίας οι ίδιες γραμμές (και ακροδέκτες του μικροεπεξεργαστή) μπορούν να χρησιμοποιηθούν σε διαφορετικές χρονικές στιγμές για διαφορετικό σκοπό (μεταφορά δεδομένων ή διευθύνσεων) ονομάζεται *πολυπλεξία (multiplexing)*.

Για παράδειγμα, στο Σχήμα 4.1 φαίνεται το σχηματικό διάγραμμα ενός μικροεπεξεργαστή ο οποίος μπορεί να συνδεθεί σε διάδρομο διευθύνσεων πλάτους 8 δυαδικών ψηφίων, και διάδρομο δεδομένων πλάτους 4 δυαδικών ψηφίων.

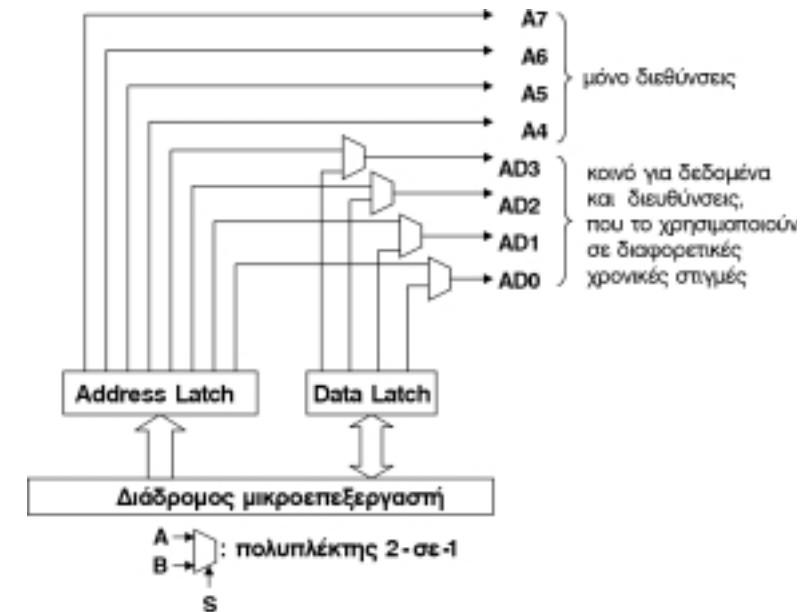


Σχήμα 4.1 Πολυπλεξία διαδρόμου διευθύνσεων και δεδομένων

Σε αυτό το μικροεπεξεργαστή οι γραμμές διευθύνσεων A0-A3 *πολυπλέκονται* με τις γραμμές δεδομένων D0-D3. Για τη σύνδεση στο διάδρομο δεδομένων χρησιμοποιείται ένας απομονωτής. Όταν στις γραμμές AD3-AD0 περιέχεται πληροφορία σχετικά με διεύθυνση, ενεργοποιείται το σήμα 'ενεργοποίηση', και η πληροφορία μεταφέρεται στο διάδρομο διευθύνσεων. Αν τα σήματα AD0-AD3 περιέχουν δεδομένα, τότε το σήμα 'επίτρεψη' απενεργοποιείται, και τα σήματα δε μεταφέρονται στο διάδρομο διευθύνσεων.

Μέσα στο ολοκληρωμένο κύκλωμα του μικροεπεξεργαστή η πολυπλεξία υλοποιείται

χρησιμοποιώντας πολυπλέκτες 2-σε-1, όπως φαίνεται στο Σχήμα 4.2.



Σχήμα 4.2 Υλοποίηση της πολυπλεξίας διαδρόμου στο εσωτερικό του μικροεπεξεργαστή

4.2 Προσπέλαση συσκευών εισόδου-εξόδου

Στις εφαρμογές μικροϋπολογιστικών συστημάτων, εκτός από τη μεταφορά δεδομένων μεταξύ του μικροεπεξεργαστή και της μνήμης, απαιτείται η μεταφορά δεδομένων μεταξύ του μικροεπεξεργαστή ή της μνήμης και μονάδων εισόδου-εξόδου. Η λειτουργία αυτή ονομάζεται λειτουργία εισόδου-εξόδου.

Η λειτουργία εισόδου-εξόδου είναι ιδιαίτερα σημαντική, ειδικά σε εφαρμογές στις οποίες το μικροϋπολογιστικό σύστημα χρησιμοποιείται για τον έλεγχο συσκευών ή διατάξεων, σε αυτοματισμούς κ.λπ. Η λειτουργία αυτή διαφέρει από την επικοινωνία του μικροεπεξεργαστή με τη μνήμη κυρίως στα ακόλουθα σημεία:

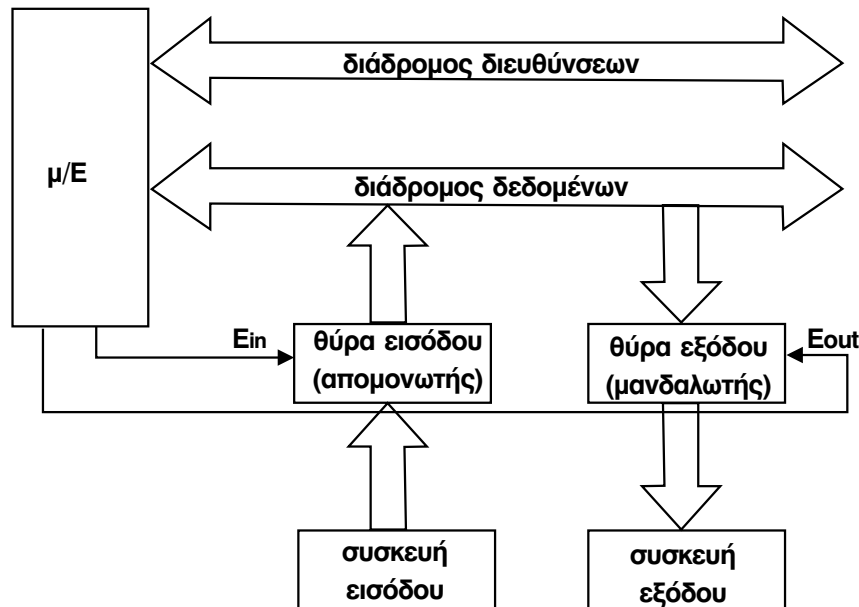
- η μνήμη δε ζητάει εξυπηρέτηση από το μικροεπεξεργαστή. Ο μικροεπεξεργαστής προσπελαίνει για τη μνήμη να εγγράψει δεδομένα σε αυτή ή να διαβάσει από αυτή δεδομένα. Αντιθέτως, μια μονάδα εισόδου-εξόδου μπορεί να ζητήσει εξυπηρέτηση. Για παράδειγμα, όταν πληκτρολογούμε ένα χαρακτήρα, το πληκτρολόγιο πρέπει να ενημερώσει το μικροεπεξεργαστή για την ενέργειά μας αυτή.
- η μνήμη αποτελεί μια ενιαία οντότητα. Για να προσπελάσει ο μικροεπεξεργαστής

μια θέση μνήμης χρησιμοποιεί τη διεύθυνση της. Αντιθέτως, όταν μια περιφερειακή συσκευή ζητάει εξυπηρέτηση, ο μικροεπεξεργαστής πρέπει να γνωρίζει ποια συσκευή είναι ώστε να την εξυπηρετήσει.

4.2.1 Θύρες εισόδου-εξόδου

Όπως έχει αναφερθεί, η επικοινωνία μεταξύ του μικροεπεξεργαστή και των συσκευών εισόδου-εξόδου πραγματοποιείται με τη βοήθεια θυρών εισόδου-εξόδου.

Οι θύρες εισόδου-εξόδου (I/O ports) μπορούν να θεωρηθούν ως εξωτερικοί καταχωρητές. Στην απλούστερη μορφή της, μια θύρα εισόδου αποτελείται από έναν απομονωτή (buffer) τριών καταστάσεων ενός δυαδικού ψηφίου, ενώ για μια θύρα εξόδου χρησιμοποιείται ένας μανδαλωτής (ή flip-flop) για κάθε δυαδικό ψηφίο της θύρας. Στο σχήμα 4.3 φαίνεται το απλοποιημένο διάγραμμα μιας θύρας εισόδου και μιας θύρας εξόδου σε ένα μικροϋπολογιστικό σύστημα.



Σχήμα 4.3 Θύρα εισόδου και θύρα εξόδου σε μικροϋπολογιστικό σύστημα

Τα σήματα Ein και Eout παράγονται από το μικροεπεξεργαστή και μεταφέρονται μέσω του διαδρόμου ελέγχου.

Αξίζει να αναφερθεί ακόμη ότι στην αγορά υπάρχουν ολοκληρωμένα κυκλώματα θυρών εισόδου-εξόδου (τεχνολογίας LSI) τα οποία μπορούν να χρησιμοποιηθούν για το χειρισμό πολύπλοκων καταστάσεων εισόδου-εξόδου όπως π.χ. μετατροπή παράλληλης επικοινωνίας σε σειριακή και αντίστροφα. Τα ολοκληρωμένα αυτά κυκλώματα είναι συνήθως προγραμματιζόμενες συσκευές, με την έννοια ότι μπορούν

να διαμορφωθούν σε θύρες εισόδου, εξόδου ή διπλής κατεύθυνσης (εισόδου και εξόδου).

4.2.2 Διευθυνσιοδότηση συσκευών εισόδου-εξόδου

Οι θύρες εισόδου-εξόδου μπορούν να προσπελαστούν με ένα από δύο τρόπους: είτε ως κοινές θέσεις μνήμης, είτε με ειδικές εντολές. Έτσι, υπάρχουν δύο τεχνικές χειρισμού των θυρών εισόδου-εξόδου. Η τεχνική μέσω ειδικών εντολών εισόδου εξόδου (isolated I/O) και η τεχνική με απεικόνιση μνήμης (memory mapped).

Είσοδος-έξοδος με ειδικές εντολές

Στην ξεχωριστή ή απομονωμένη είσοδο-έξοδο (isolated I/O) η μεταφορά των δεδομένων γίνεται χρησιμοποιώντας δύο ειδικές εντολές, οι οποίες σε συμβολική γλώσσα ονομάζονται συνήθως IN και OUT. Η διακίνηση των δεδομένων πραγματοποιείται μέσω του συσσωρευτή.

Έτσι, με την εντολή εισόδου το περιεχόμενο του απομονωτή της θύρας μεταφέρεται στο συσσωρευτή, ενώ στην εντολή εξόδου συμβαίνει το αντίστροφο. Κατά τη μεταφορά, η διεύθυνση της θύρας μπαίνει στο διάδρομο διευθύνσεων.

Οι εντολές εισόδου-εξόδου απομονώνουν τη μνήμη από τις θύρες εισόδου-εξόδου. Έτσι, ο χώρος μνήμης να μην επηρεάζεται από το χώρο των θυρών εισόδου-εξόδου. Έτσι, οι διευθύνσεις των θυρών εισόδου-εξόδου δεν αποτελούν τμήμα των διευθύνσεων μνήμης. Γι αυτό το λόγο, στο διάδρομο ελέγχου εκτός από τα σήματα που μας δίνουν τη δυνατότητα να αναγνωρίσουμε αν έχουμε ανάγνωση ή εγγραφή, έχουμε και επιπρόσθετα σήματα που μας πληροφορούν αν η προσπέλαση γίνεται σε μνήμη ή περιφερειακή συσκευή. Με την απομονωμένη είσοδο-έξοδο ο μικροϋπολογιστής μπορεί να διαθέσει όλες τις δυνατές διευθύνσεις του για μνήμη.

Είσοδος-έξοδος με απεικόνιση μνήμης

Δεν είναι όμως πάντα απαραίτητο να έχουμε ξεχωριστές εντολές εισόδου-εξόδου. Σε πολλούς μικροεπεξεργαστές οι εντολές που χρησιμοποιούνται για την ανταλλαγή πληροφοριών μεταξύ μικροεπεξεργαστή και μνήμης μπορούν να χρησιμοποιηθούν και για είσοδο-έξοδο.

Στην περίπτωση αυτή, οι θύρες εισόδου-εξόδου σχεδιάζονται έτσι ώστε να συμπεριφέρονται ως διευθύνσεις μνήμης. Συγκεκριμένα, οι καταχωρητές που σχετίζονται με τις θύρες εισόδου-εξόδου αντιστοιχούν σε θέσεις μνήμης. Η τεχνική αυτή, που χρησιμοποιεί εντολές προσπέλασης μνήμης για την είσοδο-έξοδο ονομάζεται είσοδος-έξοδος με απεικόνιση μνήμης (memory mapped I/O). Τα συστήματα που λειτουργούν έτσι, επιτρέπουν στον προγραμματιστή να χρησιμοποιεί όλες τις δυνατότητες των εντολών του μικροεπεξεργαστή, τόσο για δεδομένα της μνήμης, όσο και για δεδομένα των θυρών εισόδου-εξόδου.

Η είσοδος-έξοδος με απεικόνιση μνήμης έχει τα ακόλουθα πλεονεκτήματα:

- Εντολές αναφοράς στη μνήμη μπορούν να χρησιμοποιηθούν και για εισόδο-εξόδο (για παράδειγμα, μπορούν να γίνουν αριθμητικές πράξεις με τα περιεχόμενα μίας θύρας εισόδου ή εξόδου χωρίς να τοποθετηθούν τα δεδομένα της σε προσωρινούς καταχωρητές.
- Οι περισσότεροι καταχωρητές του μικροεπεξεργαστή μπορούν να ανταλλάξουν πληροφορίες με συσκευές εισόδου-εξόδου, σε αντίθεση με την τεχνική εισόδου-εξόδου με ειδικές εντολές (στις οποίες η μεταφορά δεδομένων γίνεται, όπως είδαμε, από και προς τον καταχωρητή).

Τα μειονεκτήματα της τεχνικής αυτής είναι ότι:

- Μειώνεται το πλήθος των θέσεων μνήμης που μπορεί να προσπελάσει ο μικροεπεξεργαστής.
- Οι εντολές προσπέλασης μνήμης είναι συνήθως μεγαλύτερες από τις εντολές τύπου εισόδου-εξόδου (για παράδειγμα, στο μικροεπεξεργαστή 8085 οι εντολές προσπέλασης μνήμης είναι συνήθως τριών bytes, ενώ οι ειδικές εντολές εισόδου-εξόδου είναι μόνο δύο bytes). Έτσι, το μήκος του προγράμματος αυξάνει.

4.2.3 Τρόποι προσπέλασης συσκευών εισόδου-εξόδου

Στην παράγραφο αυτή θα αναφερθούμε στους τρόπους με τους οποίους μπορεί να επικοινωνήσει ο μικροεπεξεργαστής και η μνήμη με τις συσκευές εισόδου-εξόδου. Οι τρόποι αυτοί είναι:

- ελεγχόμενη από το πρόγραμμα
- με χρήση διακοπών
- χρησιμοποιώντας ειδικό επεξεργαστή

Οι τρόποι αυτοί διαφέρουν στον τρόπο με τον οποίο ο μικροεπεξεργαστής αρχικοποιεί τη λειτουργία και ελέγχει τη μεταφορά των δεδομένων.

Είσοδος-εξόδος ελεγχόμενη από πρόγραμμα

Με εισόδο-εξόδο που ελέγχεται από πρόγραμμα, η μεταφορά των δεδομένων βρίσκεται κάτω από τη συνεχή παρακολούθηση και τον έλεγχο του μικροεπεξεργαστή. Μια λειτουργία εισόδου-εξόδου γίνεται μόνο όταν κατά την εκτέλεση ενός προγράμματος υπάρχει μια εντολή εισόδου-εξόδου. Πριν γίνει η μεταφορά των δεδομένων πρέπει να προσδιοριστεί αν η περιφερειακή συσκευή είναι σε θέση να επικοινωνήσει με το μικροεπεξεργαστή.

Στην περίπτωση αυτή πρέπει να ελεγχθεί μια ή περισσότερες εξωτερικές σημαίες ή δυαδικά ψηφία κατάστασης που σχετίζονται με την περιφερειακή συσκευή εισόδου-εξόδου.

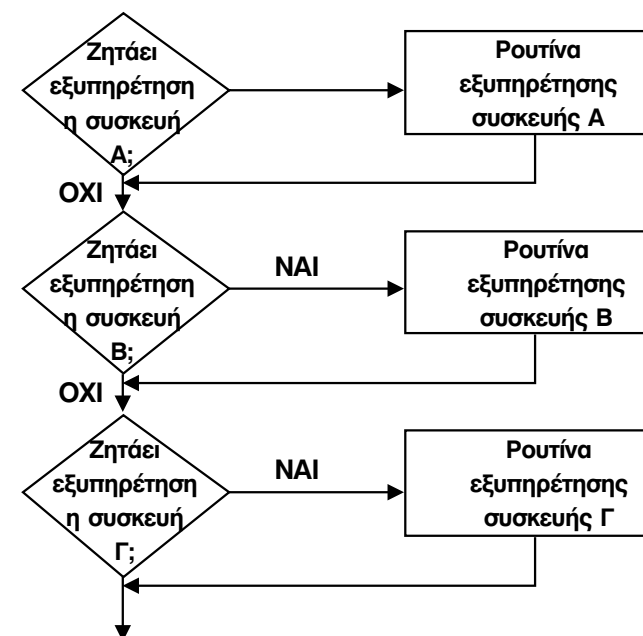
Το πρόγραμμα καθορίζει ποια συσκευή πρόκειται να εξυπηρετηθεί εξετάζοντας τα bits ενός καταχωρητή κατάστασης. Η εξέταση των bits αυτών ονομάζεται *περίοδευση (polling)* ενώ το πρόγραμμα που χρησιμοποιείται λέγεται *βρόγχος περίοδευσης*

(*polling loop*). Το πρόγραμμα εκτελεί το βρόγχο περίοδευσης ελέγχοντας διαδοχικά όλες τις συσκευές που είναι συνδεδεμένες στο διάδρομο δεδομένων και εξετάζοντας αν κάποιες από αυτές ζητούν εξυπηρέτηση. Τα πλεονεκτήματα της τεχνικής περίοδευσης είναι τα ακόλουθα:

- απαιτείται ελάχιστο υλικό και καμία εξειδικευμένη γραμμή
- είναι σύγχρονη με την εκτέλεση του προγράμματος. Αυτό σημαίνει ότι ο προγραμματιστής γνωρίζει πότε θα ερωτηθεί μια συσκευή

Τα μειονεκτήματα της μεθόδου αυτής είναι:

- Το γεγονός ότι ελέγχονται όλες οι συσκευές κάθε φορά που ξεκινάει ένας βρόγχος περίοδευσης, ενώ πρακτικά οι περισσότερες δε θα χρειάζονται εξυπηρέτηση.
- Το χρονικό διάστημα μεταξύ της χρονικής στιγμής που μια συσκευή εισόδου-εξόδου είναι έτοιμη και της χρονικής στιγμής κατά την οποία θα εξυπηρετηθεί δεν είναι συγκεκριμένο.



Σχήμα 4.4 Βρόγχος περίοδευσης

Η επικοινωνία μέσω διακοπών και με χρήση ειδικού επεξεργαστή αποτελεί το αντικείμενο των επόμενων παραγράφων.

4.3 Διακοπές

Σε πολλές περιπτώσεις ένα σημαντικό μέρος ή και ολόκληρος ο χρόνος του μικροεπεξεργαστή καταναλώνεται στην εξυπηρέτηση περιφερειακών συσκευών, όποτε αυτές το χρειαστούν.

Αυτό συμβαίνει στη μέθοδο της συνεχούς περιόδευσης (rolling), όπως είδαμε. Στη μέθοδο αυτή ο μικροεπεξεργαστής πρέπει να παρακολουθεί συνεχώς τις περιφερειακές συσκευές μέσω μιας θύρας εισόδου, προκειμένου να διαπιστώσει αν κάποια συσκευή χρειάζεται εξυπηρέτηση. Αυτή η συνεχής παρακολούθηση έχει μια χρονική επιβάρυνση, η οποία είναι τόσο μεγαλύτερη όσο μεγαλύτερη είναι η συχνότητα με την οποία ζητούν να εξυπηρετηθούν οι περιφερειακές συσκευές.

Οι διακοπές είναι ένας άλλος τρόπος εξυπηρέτησης των περιφερειακών συσκευών, ο οποίος αφήνει το μικροεπεξεργαστή αφιερωμένο στην εκτέλεση του κύριου προγράμματος. Με τη μέθοδο των διακοπών ο μικροεπεξεργαστής ασχολείται με κάποια περιφερειακή συσκευή μόνο όταν αυτή ζητήσει εξυπηρέτηση.

4.3.1 Πλεονεκτήματα της μεθόδου των διακοπών

Σύμφωνα με τη μέθοδο των διακοπών, όταν μια περιφερειακή συσκευή χρειασθεί εξυπηρέτηση, έχει τη δυνατότητα να ειδοποιήσει το μικροεπεξεργαστή ενεργοποιώντας κάποιο ακροδέκτη του (int, από τη λέξη interrupt), που προορίζεται ειδικά για το σκοπό αυτό. Ο μικροεπεξεργαστής απαντώντας στην αίτηση για εξυπηρέτηση, διακόπτει το πρόγραμμα που εκτελείται εκείνη τη στιγμή και εκτελεί μια υπορουτίνα, που έχει γραφεί για την εξυπηρέτηση της συσκευής αυτής. Όταν ολοκληρωθεί η εκτέλεση της υπορουτίνας, το πρόγραμμα που διακόπηκε συνεχίζεται από το σημείο που έγινε η διακοπή.

Στην περίπτωση αυτή λέμε ότι η συσκευή προκάλεσε διακοπή ή ότι έκανε αίτηση διακοπής και ο μικροεπεξεργαστής, αφού αναγνώρισε τη διακοπή, την εξυπηρέτησε. Η υπορουτίνα που εκτελείται στην περίπτωση αυτή ονομάζεται υπορουτίνα εξυπηρέτησης διακοπής. Ο ειδικός ακροδέκτης μέσω του οποίου προκαλείται διακοπή ονομάζεται είσοδος διακοπής.

Με τον τρόπο αυτό ο μικροεπεξεργαστής δε χρειάζεται να παρακολουθεί συνεχώς κάποια θύρα εισόδου, και εξοικονομείται χρόνος, γιατί ο μικροεπεξεργαστής μπορεί να αφιερώσει όλο το χρόνο του στο κυρίως πρόγραμμα.

Προτεραιότητα διακοπών

Συνήθως ένας μικροεπεξεργαστής έχει περισσότερες από μια περιφερειακές συσκευές γι' αυτό διαθέτει περισσότερες από μια εισόδους διακοπής. Επίσης, πολλές συσκευές μπορούν να είναι συνδεδεμένες σε μια κοινή γραμμική διακοπή.

Οι πολλές εισόδους διακοπής σε συνδυασμό με τις πολλαπλές συσκευές καθιστούν

απαραίτητο τον καθορισμό προτεραιότητας για την περίπτωση που περισσότερες από μια συσκευές ζητήσουν ταυτόχρονα εξυπηρέτηση. Η προτεραιότητα καθορίζεται από το σχεδιαστή του μικροεπεξεργαστή και συνήθως δε μπορεί να μεταβληθεί.

Ενεργοποίηση και απενεργοποίηση μηχανισμού διακοπών

Ορισμένες φορές, σε ένα πρόγραμμα που εκτελείται στο μικροεπεξεργαστή υπάρχουν κρίσιμα τμήματα τα οποία θα πρέπει να εκτελεστούν χωρίς διακοπές. Για παράδειγμα, αυτό θα μπορούσε να συμβεί στην περίπτωση που υπάρχουν χρονικοί περιορισμοί για την ολοκλήρωση της εκτέλεσης ενός προγράμματος.

Στην περίπτωση αυτή, είναι αναγκαίο να απενεργοποιηθεί για κάποιο διάστημα ο μηχανισμός των διακοπών. Όταν συμβεί αυτό, ο μικροεπεξεργαστής αγνοεί οποιαδήποτε αίτηση διακοπής μέχρι ο μηχανισμός των διακοπών να ενεργοποιηθεί και πάλι. Η ενεργοποίηση και η απενεργοποίηση γίνεται με ειδικές εντολές που πρέπει να ενσωματωθούν στα κατάλληλα σημεία ενός προγράμματος.

Άλλες φορές, δε χρειάζεται να απενεργοποιηθούν όλες οι διακοπές αλλά να παρεμποδιστούν επιλεκτικά κάποιες από αυτές. Οι περισσότεροι μικροεπεξεργαστές μας παρέχουν τη δυνατότητα να παρεμποδίζουμε, (να θέτουμε, όπως λέμε, μάσκα διακοπών) σε όλες ή σε κάποιες από τις εισόδους διακοπών. Οι εισοδοί αυτές ονομάζονται παρεμποδιζόμενες διακοπές επειδή μπορούν να παρεμποδιστούν με τη χρήση μάσκας. Το ρεπερτόριο εντολών των μικροεπεξεργαστών περιέχει ειδικές εντολές για την τοποθέτηση και την αφαίρεση της μάσκας διακοπών. Έτσι, το ποιες διακοπές θα παρεμποδιστούν και σε ποια χρονική στιγμή είναι ευθύνη του προγράμματος.

Πολυεπεξεργασία και Εξυπηρέτηση διακοπών

Ένας μικροεπεξεργαστής είναι δυνατό να εκτελεί ταυτόχρονα περισσότερα από ένα προγράμματα. Με τον όρο ταυτόχρονα δεν εννοούμε ότι τα δυο ή περισσότερα προγράμματα εκτελούνται από το μικροεπεξεργαστή την ίδια χρονική στιγμή, αλλά βρίσκονται στη μνήμη και ο μικροεπεξεργαστής μπορεί να επιλέγει να εκτελεί τότε το ένα τότε το άλλο, ανάλογα με τις ανάγκες του μικροϋπολογιστικού συστήματος. Ας θεωρήσουμε για παράδειγμα ένα μικροϋπολογιστικό σύστημα που έχει δυο προγράμματα στη μνήμη, τα οποία ας ονομάσουμε A και B. Αρχικά, ξεκινάει η εκτέλεση του προγράμματος A. Σε κάποιο σημείο κατά τη διάρκεια εκτέλεσης του προγράμματος A, είναι απαραίτητο να εκτελεστεί το πρόγραμμα B. Αυτή η μεταφορά του ελέγχου στο πρόγραμμα B απαιτεί ο μετρητής προγράμματος να φορτωθεί με τη διεύθυνση έναρξης του προγράμματος B. Αν μετά την ολοκλήρωση της εκτέλεσης του προγράμματος B το πρόγραμμα A είναι αναγκασμένο να ξαναρχίσει από την αρχή, τα αποτελέσματα που είχαν υπολογισθεί πριν τη διακοπή θα χαθούν.

Έτσι, μπορούμε να συμπεράνουμε ότι όταν η εκτέλεση ενός προγράμματος διακόπτεται για να συνεχιστεί αργότερα έχει ιδιαίτερη σημασία να αποθηκεύσουμε την κατάσταση του μικροεπεξεργαστή. Η κατάσταση του μικροεπεξεργαστή είναι το σύνολο της πληροφορίας που περιγράφει τη φάση στην οποία βρίσκεται ένας μικροεπεξεργαστής. Έτσι, αν τη στιγμή της διακοπής του προγράμματος Α η κατάσταση του μικροεπεξεργαστή σωθεί (και αργότερα αποκατασταθεί), η επεξεργασία γυρνά στο πρόγραμμα Α χωρίς να χαθούν τα αποτελέσματα της προηγούμενης επεξεργασίας. Μερικές από τις πληροφορίες που περιλαμβάνει η κατάσταση του μικροεπεξεργαστή είναι τα περιεχόμενα του μετρητή προγράμματος, του συσσωρευτή, του καταχωρητή κατάστασης και των καταχωρητών γενικού σκοπού. Η διακοπή ενός προγράμματος δεν είναι τόσο σπάνια όσο πιθανό να φαντάζεται κανείς. Για παράδειγμα, με την εκτέλεση μιας εντολής κλήσης υπορουτίνας ή από ένα σήμα που προέρχεται από εξωτερική συσκευή η σειριακή επεξεργασία διακόπτεται. Η εντολή κλήσης υπορουτίνας προκαλεί μια διακλάδωση από ένα μέρος προγράμματος σε άλλο, την υπορουτίνα. Μια εξωτερική συσκευή μπορεί να απαιτήσει εξυπηρέτηση προκαλώντας (πχ. με μια διακοπή υλικού) μια διακλάδωση σε προκαθορισμένη θέση. Η θέση μνήμης στην οποία γίνεται η διακλάδωση είναι η θέση έναρξης της υπορουτίνας εξυπηρέτησης της συσκευής που προκάλεσε τη διακοπή. Η αποθήκευση της κατάστασης του μικροεπεξεργαστή πραγματοποιείται με τη βοήθεια της στοίβας. Η στοίβα (stack) είναι μια δομή αποθήκευσης στην οποία ο μικροεπεξεργαστής αποθηκεύει τα περιεχόμενα των καταχωρητών του κατά την κλήση υπορουτινών και διακοπών. Η στοίβα αποτελείται από ένα σύνολο θέσεων στη μνήμη RAM. Για να κρατάμε τη διεύθυνση μιας θέσης μέσα στη στοίβα είναι απαραίτητος ένας δείκτης στοίβας (stack pointer). Ο δείκτης αυτός είναι σχεδιασμένος με τέτοιο τρόπο ώστε τα δεδομένα να διαβάζονται από τη στοίβα με την αντίστροφη σειρά από εκείνη με την οποία γράφτηκαν.

Διαδικασία εξυπηρέτησης διακοπής

Συνοψίζοντας, μπορούμε να πούμε ότι η διαδικασία εξυπηρέτησης μιας διακοπής αποτελείται από τα ακόλουθα βήματα:

- ολοκληρώνεται η εντολή την οποία εκτελούσε ο μικροεπεξεργαστής
- απενεργοποιούνται όλες οι διακοπές
- σώζεται στη στοίβα η κατάσταση του μικροεπεξεργαστή (συνήθως σαν κατάσταση σώζεται ο απριθμητής προγράμματος και ο καταχωρητής κατάστασης).
- εκτελείται η υπορουτίνα εξυπηρέτησης διακοπής
- ανακτάται η κατάσταση του μικροεπεξεργαστή από τη στοίβα και επιστρέφεται ο έλεγχος στο πρόγραμμα στην επόμενη εντολή από εκείνη που είχε τελευταία εκτελεστεί

4.4 Λειτουργία απευθείας προσπέλασης μνήμης

Αν ο ρυθμός μεταφοράς δεδομένων από και προς μια περιφερειακή συσκευή είναι σχετικά χαμηλός, τότε η επικοινωνία μπορεί να εκτελεστεί με είσοδο-έξοδο ελεγχόμενη με πρόγραμμα είτε με χρήση διακοπών.

Όπως είδαμε όμως, η εκτέλεση εντολών και η διαδικασία των διακοπών απαιτεί περισσότερο χρόνο, λόγω της αποθήκευσης και ανάκτησης της κατάστασης του μικροεπεξεργαστή. Ορισμένες συσκευές απαιτούν υψηλούς ρυθμούς μεταφοράς δεδομένων. Έτσι, είναι ασύμφορο να εξυπηρετούνται μεταφέροντας ένα byte ή μια λέξη κάθε φορά.

Συχνά, ο ρυθμός μεταφοράς δεδομένων από περιφερειακές συσκευές καθορίζεται από τις ίδιες τις συσκευές και όχι από το μικροεπεξεργαστή. Έτσι, το υπολογιστικό σύστημα πρέπει να εκτελεί την είσοδο-έξοδο σύμφωνα με τη μέγιστη ταχύτητα της συσκευής. Σε περιπτώσεις στις οποίες απαιτούνται υψηλοί ρυθμοί μεταφοράς δεδομένων, χρησιμοποιείται απευθείας προσπέλαση της μνήμης (direct memory access, DMA).

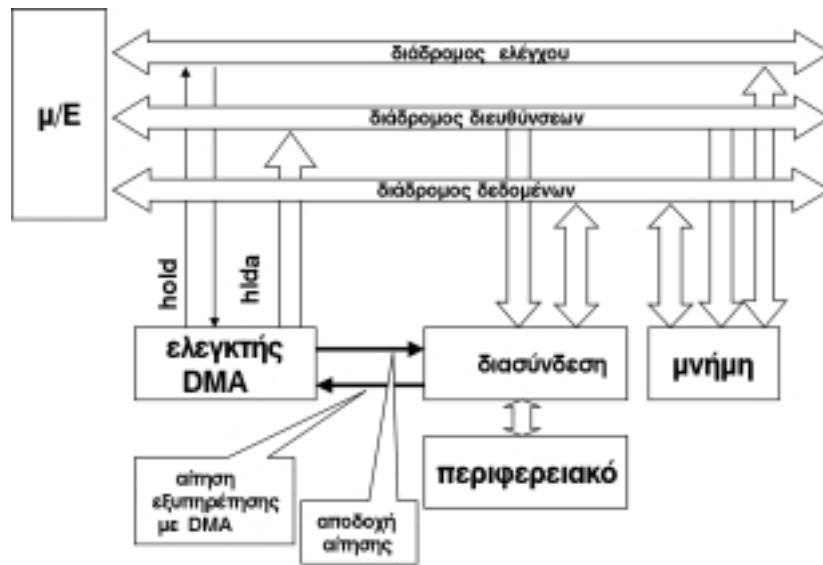
Η μεταφορά ενός byte ή μιας λέξης μέσω των διαδρόμων του συστήματος ονομάζεται κύκλος μηχανής. Σε ένα κύκλο μηχανής κάποια από τις μονάδες του συστήματος που συνδέεται στο διάδρομο του συστήματος ελέγχει το διάδρομο. Η μονάδα αυτή ονομάζεται διαχειριστής (master) του διαδρόμου κατά τη διάρκεια αυτού του κύκλου, και το στοιχείο με το οποίο επικοινωνεί ονομάζεται υποτελής (slave).

Κανονικά, ο διαχειριστής του διαδρόμου είναι ο μικροεπεξεργαστής. Υπάρχει όμως η δυνατότητα μια (ειδικά σχεδιασμένη) μονάδα του συστήματος, να αποκτήσει τον έλεγχο του διαδρόμου στέλνοντας στο μικροεπεξεργαστή μια αίτηση απόκτησης του διαδρόμου (bus request). Μετά την ολοκλήρωση του κύκλου μηχανής ο μικροεπεξεργαστής επιστρέφει ένα σήμα απόδοσης διαδρόμου και η μονάδα που έστειλε την αίτηση γίνεται διαχειριστής.

Ο διαχειριστής τοποθετεί διευθύνσεις στο διάδρομο διευθύνσεων και συντονίζει τη δραστηριότητα του διαδρόμου κατά τη διάρκεια ενός κύκλου μηχανής, όπως ακριβώς ο μικροεπεξεργαστής. Μια μονάδα που μπορεί να γίνει διαχειριστής είναι ο ελεγκτής DMA.

Στο Σχήμα 4.5 φαίνεται το σχηματικό διάγραμμα ενός μικροϋπολογιστικού συστήματος το οποίο περιλαμβάνει ένα ελεγκτή DMA.

Ο ελεγκτής περιλαμβάνει (εκτός από τους καταχωρητές κατάστασης και ελέγχου) δύο επιπλέον καταχωρητές, ένα για να κρατά τη διεύθυνση της επόμενης θέσης μνήμης και ένα στον οποίο θα αποθηκεύεται ο αριθμός των δεδομένων που πρέπει ακόμη να μεταφερθούν. Οι καταχωρητές αυτοί είναι θύρες εισόδου-εξόδου. Αμέσως μετά τη μεταφορά ενός δεδομένου ο καταχωρητής διεύθυνσης αυξάνεται ενώ ο απριθμητής δεδομένων μειώνεται.



Σχήμα 4.5 Λειτουργία DMA

Για τη μεταφορά κάποιων δεδομένων από ένα περιφερειακό προς τη μνήμη μέσω DMA, εκτελούνται τα ακόλουθα βήματα:

- η διασύνδεση της περιφερειακής συσκευής στέλνει στον ελεγκτή μια αίτηση για εξυπηρέτηση DMA.
- ο ελεγκτής ενεργοποιεί το σήμα αίτησης για απόκτηση του διαδρόμου (HOLD). Ο μικροεπεξεργαστής αποκρίνεται με ένα σήμα αποδοχής αίτησης απόκτησης (HLDA) και ελευθερώνει το διάδρομο δεδομένων, το διάδρομο διευθύνσεων και το διάδρομο ελέγχου. Ο ελεγκτής DMA αποκτά τον έλεγχο των διαδρόμων.
- Ο ελεγκτής στέλνει στη διασύνδεση ένα σήμα αποδοχής DMA με το οποίο ζητά από αυτή να τοποθετήσει τα δεδομένα στο διάδρομο δεδομένων (για λειτουργία εισόδου), ή να πάρει τα επόμενα δεδομένα που είναι τοποθετημένα πάνω στο διάδρομο (για λειτουργία εξόδου).
- Το προς μεταφορά δεδομένο μεταφέρεται από ή προς τη θέση μνήμης που δηλώνεται μέσω του διαδρόμου διευθύνσεων που ελέγχεται από τον καταχωρητή διεύθυνσης DMA.
- στη συνέχεια αυξάνει ο καταχωρητής διευθύνσεων και ο απεριθμητής δεδομένων μειώνεται κατά 1 μέχρι να μεταφερθούν όλα τα δεδομένα.

Πρέπει να σημειωθεί ότι αν ζητήσουν ταυτόχρονα εξυπηρέτηση δύο ή περισσότερα περιφερειακά, η σειρά με την οποία θα εξυπηρετηθούν εξαρτάται από τον τρόπο με τον οποίο έχει καθοριστεί η προτεραιότητα στο σύστημα. Υπάρχουν πολλές τεχνικές για την απόδοση της προτεραιότητας αυτής, εκ των οποίων άλλες βασίζονται στο υλικό, άλλες σε λογισμικό και άλλες σε συνδυασμό των δύο.

Τρόποι μεταφοράς DMA

Κατά τη διάρκεια της μεταφοράς DMA ο μικροεπεξεργαστής αδρανοποιείται μέχρι να μεταφερθούν όλα τα δεδομένα. Με αυτόν τον τρόπο επιτυγχάνεται ο μέγιστος ρυθμός μεταφοράς. Ο τρόπος αυτός χρησιμοποιείται στην περίπτωση που πρέπει να μεταφερθεί μεγάλη ποσότητα δεδομένων σε μικρό χρονικό διάστημα.

Υπάρχουν όμως περιπτώσεις, στις οποίες δεν απαιτείται ο μέγιστος ρυθμός μεταφοράς. Για τις περιπτώσεις αυτές οι ελεγκτές DMA παρέχουν και ένα άλλο τρόπο λειτουργίας στον οποίο η μεταφορά DMA γίνεται παράλληλα με τη λειτουργία του μικροεπεξεργαστή. Για να γίνει αυτό, κάθε φορά που μεταφέρεται ένα byte, ο ελεγκτής DMA παραχωρεί τον έλεγχο του διαδρόμου στο μικροεπεξεργαστή και στη συνέχεια υποβάλλει καινούρια αίτηση για μεταφορά DMA. Έτσι ο ελεγκτής κλέβει κύκλους μηχανής από το μικροεπεξεργαστή για να μεταφέρει τα δεδομένα. Στην περίπτωση αυτή λέμε ότι η μεταφορά DMA γίνεται με τη μέθοδο της κλοπής κύκλου (cycle stealing).

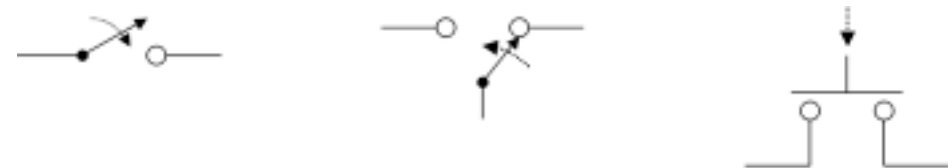
4.5 Είσοδος και έξοδος ψηφιακών δεδομένων σε μικροεπεξεργαστή

Όπως αναφέραμε, μια από τις κυριότερες λειτουργίες ενός μικροεπεξεργαστή είναι η επικοινωνία με το περιβάλλον. Στην παράγραφο αυτή θα αναφερθούμε στις απλές συσκευές με τις οποίες ο μικροεπεξεργαστής μπορεί να ανταλλάξει (εισάγει και εξάγει) ψηφιακά δεδομένα με το περιβάλλον.

4.5.1 Είσοδος δεδομένων

Οι πιο γνωστές συσκευές με τις οποίες εισάγουμε ψηφιακά δεδομένα στο μικροεπεξεργαστή, είναι οι μηχανικοί διακόπτες και τα πληκτρολόγια. Στη συνέχεια θα αναφερθούμε στις συσκευές αυτές.

Οι μηχανικοί διακόπτες χρησιμοποιούνται για τη χειρωνακτική εισαγωγή δεδομένων σε μικροϋπολογιστικά συστήματα. Στην αγορά υπάρχει ποικιλία διακοπών, οι οποίοι διαφέρουν στην κατασκευή, οι πιο γνωστοί από τους οποίους φαίνονται στο Σχήμα 4.6.



Σχήμα 4.6 Μηχανικοί Διακόπτες

Για να εισαχθούν δεδομένα σε ένα μικροϋπολογιστικό σύστημα, η κατάσταση του