



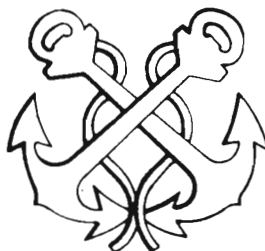
ΛΥΚΕΙΑ ΔΟΚΙΜΩΝ ΑΞΙΩΜΑΤΙΚΩΝ
ΕΜΠΟΡΙΚΟΥ ΝΑΥΤΙΚΟΥ

ΣΤΟΙΧΕΙΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Γεωργίου Κυρ. Φλαντίνη
ΠΛΩΤΑΡΧΟΥ ΜΗΧΑΝΙΚΟΥ Π.Ν.



ΙΔΡΥΜΑ ΕΥΓΕΝΙΔΟΥ
ΧΡΥΣΟΥΝ ΜΕΤΑΛΛΙΟΝ ΑΚΑΔΗΜΙΑΣ ΑΘΗΝΩΝ



ΕΚΠΑΙΔΕΥΤΙΚΟ ΚΕΙΜΕΝΟ
Α.Δ.Σ.Ε.Ν.
ΥΠΟΥΡΓΕΙΟΥ ΕΜΠΟΡΙΚΗΣ ΝΑΥΤΙΛΙΑΣ

ΠΡΟΛΟΓΟΣ ΙΔΡΥΜΑΤΟΣ ΕΥΓΕΝΙΔΟΥ

Ο Ευγένιος Ευγενίδης, ιδρυτής και χορηγός του «Ιδρύματος Ευγενίδου» προείδε ενωρίτατα και σχημάτισε τη βαθιά πεποίθηση ότι αναγκαίο παράγοντα για την πρόοδο του έθνους θα αποτελούσε η άρτια κατάρτιση των τεχνικών μας σε συνδυασμό προς την ηθική τους αγωγή.

Την πεποίθησή του αυτή τη μετέτρεψε σε γενναία πράξη ευεργεσίας, όταν κληροδότησε σεβαστό ποσό για τη σύσταση Ιδρύματος, που θα είχε ως σκοπό να συμβάλλει στην τεχνική εκπαίδευση των νέων της Ελλάδας.

Έτσι, το Φεβρουάριο του 1956 συστήθηκε το «Ίδρυμα Ευγενίδου», του οποίου τη διοίκηση ανέλαβε η αδελφή του Μαρ. Σίμου, σύμφωνα με την επιθυμία του διαθέτη. Το έργο του Ιδρύματος συνεχίζει από το 1981 ο κ. Νικόλαος Βερνίκος - Ευγενίδης.

Κατά την κλιμάκωση των σκοπών του, το Ίδρυμα πρόταξε την έκδοση τεχνικών βιβλίων τόσο για λόγους θεωρητικούς όσο και πρακτικούς. Διαπιστώθηκε πράγματι ότι αποτελεί πρωταρχική ανάγκη ο εφοδιασμός των μαθητών με σειρές από βιβλία, τα οποία θα έθεταν ορθά θεμέλια στην παιδεία τους και θα αποτελούσαν συγχρόνως πολύτιμη βιβλιοθήκη για κάθε τεχνικό.

Ειδικότερα, όσον αφορά στα εκπαιδευτικά βιβλία των σπουδαστών των Δημοσίων Σχολών Εμπορικού Ναυτικού, το Ίδρυμα ανέλαβε την έκδοσή τους σε πλήρη και στενή συνεργασία με τη Διεύθυνση Ναυτικής Εκπαιδεύσεως του Υπουργείου Εμπορικής Ναυτιλίας, υπό την εποπτεία του οποίου υπάγονται οι Σχολές αυτές.

Η ανάθεση στο Ίδρυμα έγινε με την υπ' αριθ. 61288/5031, της 9ης Αυγούστου 1966, απόφαση του Υπουργείου Εμπορικής Ναυτιλίας, οπότε και συγκροτήθηκε και η Επιτροπή Εκδόσεων.

Κύριος σκοπός των εκδόσεων αυτών, των οποίων το περιεχόμενο είναι σύμφωνο με τα εκάστοτε ισχύοντα αναλυτικά προγράμματα του Υ.Ε.Ν., είναι η παροχή προς τους σπουδαστές των ναυτικών σχολών ΑΔΣΕΝ και Ναυτικών Λυκείων των αναγκαίων εκπαιδευτικών κειμένων, τα οποία αντιστοιχούν προς τα μαθήματα που διδάσκονται στις Σχολές αυτές.

Επίσης ελήφθη πρόνοια, ώστε τα βιβλία αυτά να είναι γενικότερα χρήσιμα για όλους τους αξιωματικούς του Εμπορικού Ναυτικού, που ασκούν ήδη το επάγγελμα και εξελίσσονται στην ιεραρχία του κλάδου τους, χωρίς αυτό να σημαίνει ότι επέρχεται μεταβολή στη στάθμη του περιεχομένου τους.

Οι συγγραφείς και η Επιτροπή Εκδόσεων του Ιδρύματος καταβάλλουν κάθε προσπάθεια, ώστε τα βιβλία να είναι επιστημονικώς άρτια αλλά και προσαρμοσμένα στις ανάγκες και τις δυνατότητες των σπουδαστών. Γι' αυτό και τα βιβλία αυτά έχουν προσεγμένη γλωσσική διατύπωση και η διαπραγμάτευση των θεμάτων είναι ανάλογη προς τη στάθμη της εκπαιδεύσεως, για την οποία προορίζεται κάθε σειρά των βιβλίων.

Έτσι προσφέρονται στους καθηγητές, τους σπουδαστές της ναυτικής μας εκπαιδεύσεως και όλους τους αξιωματικούς του Ε.Ν. οι εκδόσεις του Ιδρύματος, των οποίων η συμβολή στην πραγματοποίηση του σκοπού του Ευγενίου Ευγενίδου ελπίζεται να είναι μεγάλη.

ΕΠΙΤΡΟΠΗ ΕΚΔΟΣΕΩΝ ΙΔΡΥΜΑΤΟΣ ΕΥΓΕΝΙΔΟΥ

Μιχαήλ Αγγελόπουλος, καθηγητής ΕΜΠ, Πρόεδρος.

Αλέξανδρος Σταυρόπουλος, καθηγητής Α.Β.Σ. Πειραιώς, Αντιπρόεδρος.

Ιωάννης Τεγόπουλος, καθηγητής ΕΜΠ.

Ελευθέριος Μακράκης, πλοίαρχος Α.Σ., Διευθ. Ναυτ. Εκπ. Υ.Ε.Ν.

Σύμβουλος επί των εκδόσεων του Ιδρύματος **Κων. Μανάφης**, καθηγ. Φιλ. Σχολής Παν/μίου Αθηνών.

Γραμματέας της Επιτροπής, **Γεώργιος Ανδρεάκος**.

Ι Δ Ρ Υ Μ Α Ε Υ Γ Ε Ν Ι Δ Ο Υ
Β Ι Β Λ Ι Ο Θ Η Κ Η Τ Ο Υ Ν Α Υ Τ Ι Κ Ο Υ

ΣΤΟΙΧΕΙΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΓΕΩΡΓΙΟΥ ΚΥΡ. ΦΛΑΝΤΙΝΗ

ΠΛΩΤΑΡΧΟΥ ΜΗΧΑΝΙΚΟΥ Π.Ν.

Διπλ. Ήλεκτρολόγου - Ήλεκτρονικού Μηχ.

(B.Sc. M.Sc. Degree Electrical Engineer)

Επιστημονικού Συνεργάτου Έδρας Ψηφιακών Συστημάτων και Υπολογιστών Ε.Μ.Π.

ΑΘΗΝΑ
1993

Α' ΕΚΔΟΣΗ 1982

ΠΡΟΛΟΓΟΣ

Τό ανθρώπινο μυαλό, παρόλο πού αποτελεί αποτελεσματικό υπολογιστικό σύστημα, παρουσιάζεται αρκετά περιορισμένο ως προς τήν Ικανότητά του νά απορροφά καί νά απομνημονεύει πληροφορίες. Αποτέλεσμα τοῦ περιορισμοῦ αὐτοῦ εἶναι ὅτι παρουσιάζει δυσχέρειες στήν ταχεία ἐκτέλεση πολυπλόκων υπολογιστικῶν πράξεων. Γιά νά ἀντιμετωπίσει τό πρόβλημα αὐτό ὁ ἄνθρωπος ἐπινόησε καί κατασκεύασε κατά καιρούς διάφορα ὄργανα καί μηχανές υπολογιστικές.

Ἡ τεχνολογία κατά τή διάρκεια τοῦ 20οῦ αἰῶνα σημείωσε σημαντική πρόοδο καί βοήθησε στήν ἐξέλιξη τῶν υπολογιστικῶν μηχανῶν.

Σήμερα ἡ ἀλματώδης ἐξέλιξη τῶν υπολογιστικῶν συστημάτων καί κυρίως τῶν μικροηλεκτρονικῶν κυκλωμάτων, μείωσε τό κόστος καί τόν ὄγκο τῶν ηλεκτρονικῶν υπολογιστῶν καί ἔδωσε τεράστια ὠθηση στίς διάφορες ἐφαρμογές, ἀνοίγοντας νέους ὁρίζοντες γιά τήν ἐφαρμογή τῶν ηλεκτρονικῶν υπολογιστῶν.

Στό βιβλίο αὐτό γίνεται μία προσπάθεια ἀναπτύξεως διαφόρων θεμάτων τά ὁποῖα ἔχουν σχέση μέ τοὺς ηλεκτρονικούς υπολογιστές, ὅπως π.χ. ἡ ἐσωτερική δομή τους, ὁ τρόπος ροῆς τῶν πληροφοριῶν, τά διάφορα λειτουργικά συστήματα καί οἱ γλῶσσες προγραμματισμοῦ ἀνώτερου ἐπιπέδου.

Ἐπιθυμῶ καί ἀπό τή θέση αὐτή νά εὐχαριστήσω τόν καθηγητή τοῦ Πανεπιστημίου Ἀθηνῶν κ. Γεώργιο Φιλοκύπρου γιά τή συμβολή του στήν πληρέστερη συγγραφή τοῦ βιβλίου καί τοὺς συνεργάτες τοῦ Ἰδρύματος Εὐγενίδου γιά τή βοήθειά τους στήν ἔκδοση τοῦ βιβλίου.

Ὁ συγγραφέας

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΗΛΕΚΤΡΟΝΙΚΟΥΣ ΥΠΟΛΟΓΙΣΤΕΣ

1.1 Γενικά.

Οι τρεις τεχνολογικές επαναστάσεις που χαρακτηρίζουν τον 20ο αιώνα θα μπορούσε να πει κανείς σέ γενικές γραμμές ότι είναι: η ατομική ενέργεια, η κατάκτηση του διαστήματος και οι ηλεκτρονικοί υπολογιστές.

Η ιδιαίτερη εξέλιξη των ηλεκτρονικών, και κυρίως των μικροηλεκτρονικών κυκλωμάτων, μείωσε τό κόστος και τόν όγκο των ηλεκτρονικών υπολογιστών και έδωσε μιά τεράστια ώθηση στις εφαρμογές τους.

Έτσι σήμερα οι υπολογιστές χρησιμοποιούνται στο έμπόριο, στη βιομηχανία, στην εκπαίδευση, ακόμη και στις οικιακές εργασίες.

Κανένας αυτή τή στιγμή δέν μπορεί νά προβλέψει τήν εξέλιξη των εφαρμογών των ηλεκτρονικών υπολογιστών είτε αυτοί είναι μικροϋπολογιστές είτε μίνι-υπολογιστές είτε μεγάλοι υπολογιστές.

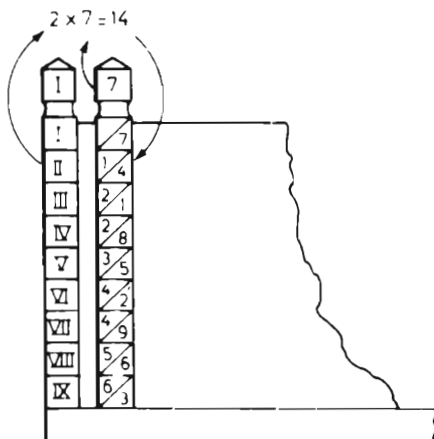
1.2 Οι πρώτες υπολογιστικές μηχανές.

Από τήν αρχαιότητα ο άνθρωπος προσπάθησε νά κατασκευάσει μηχανές που θά τόν βοηθοῦσαν στήν σωματική αλλά και στή διανοητική του εργασία. Τό πρώτο στοιχειώδες όργανο υπολογισμού που κατασκευάσθηκε τό 5.000 π.Χ. περίπου στήν Κίνα είναι ο άβακας. Οι πρώτες όμως σοβαρές προσπάθειες γιά τήν κατασκευή υπολογιστικών μηχανών άρχισαν τό 17ο αιώνα ύστερα από τήν επινόηση των λογαρίθμων από τόν Σκωτσέζο μαθηματικό John Napier τό έτος 1617. Η πρώτη υπολογιστική μηχανή του J. Napier, (σχ. 1.2α) αποτελούνταν από ράβδους πάνω στις οποίες ήταν χαραγμένοι αριθμοί. Μέ τούς αριθμούς αυτούς ή πράξη του πολλαπλασιασμού γινόταν εύκολότερη.

Τό επόμενο βήμα σημειώθηκε τό 1642, όταν ο Γάλλος μαθηματικός Blaise Pascal επενόησε τήν πρώτη υπολογιστική μηχανή μέ όδοντωτούς τροχούς (σχ. 1.2β). Κάθε όδοντωτός τροχός είχε 10 δόντια και υπήρχε μία συστοιχία όδοντωτών τροχών συνδεμένων μεταξύ τους έτσι, ώστε όταν ο ένας τροχός προχωρούσε από τό 9 στο 0, ο άμέσως επόμενος τροχός μετακινούνταν κατά μία θέση. Όπως δηλαδή λειτουργούν σήμερα οι χιλιομετρητές των αυτοκινήτων.

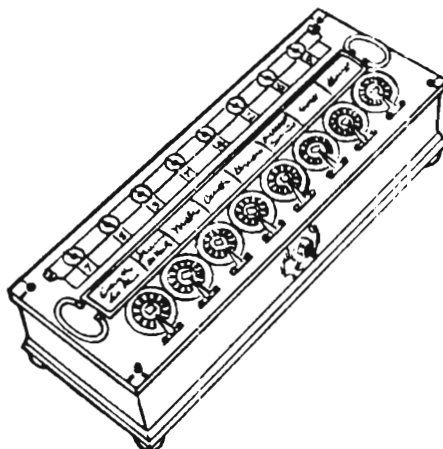
Η μηχανή του Pascal μπορούσε νά κάνει πρόσθεση και πολλαπλασιασμό. Τά αποτελέσματα εμφανιζόταν σέ πλευρικές θέσεις, όπως φαίνεται και στο σχήμα 1.2β.

Τό 1694 ο Leibnitz βελτίωσε τή μηχανή του Pascal προσδίδοντας σ' αυτή τήν δυνατότητα νά κάνει αφαίρεση και διαίρεση. Στο σχήμα 1.2γ φαίνεται ή μηχανή του Leibnitz.



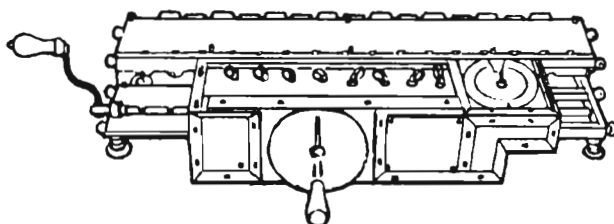
Σχ. 1.2α.

Ἡ ὑπολογιστικὴ μηχανὴ τοῦ Napier



Σχ. 1.2β.

Ἡ ὑπολογιστικὴ μηχανὴ τοῦ Pascal.



Σχ. 1.2γ.

Ἡ ὑπολογιστικὴ μηχανὴ τοῦ Leibnitz.

Τό 1745 ὁ Γάλλος μηχανικός Joseph Jacquard χρησιμοποίησε γιά πρώτη φορά διάτρητες κάρτες γιά τήν ἐπιλογή νημάτων σέ ἓνα μηχανικό ἀργαλειό.

Ἡ ἰδέα τῆς χρησιμοποίησεως τῶν διατρήτων καρτελῶν σέ μηχανή δέν ἐφαρμόστηκε μέχρι τά μέσα περίπου τοῦ 19ου αἰώνα πού ὁ Δρ. Herman Hollerith χρησιμοποίησε μιὰ ἄλλη περισσότερο ἐξελιγμένη μορφή καρτέλας ἡ ὁποία μπορούσε νά διατηρηθεῖ ἀνάλογα μέ τήν πληροφορία πού θέλαμε νά συμβολίσουμε. Ἐδῶ μπορούμε νά ποῦμε μέ βεβαιότητα ὅτι αὐτή ἡ μέθοδος παραστάσεως πληροφοριῶν σέ διάτρητες καρτέλες ἦταν ἡ πιό σπουδαία ἐξέλιξη στήν ἐπεξεργασία πληροφοριῶν, γιατί γιά πρώτη φορά εἰσήγαγε τήν ἰδέα τῆς ἀποθηκεύσεως τῆς πληροφορίας μηχανικά.

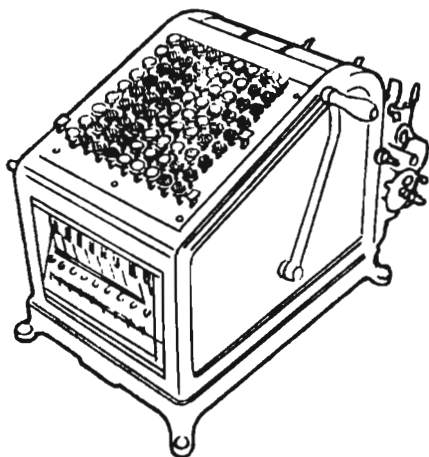
Στό τέλος περίπου τοῦ 1880, ἡ κυβέρνηση τῶν Η.Π.Α., ἀνέθεσε στόν Δρ. Herman Hollerith τήν ἀνάπτυξη μιᾶς πιό τελειοποιημένης μεθόδου γιά τήν ἐπεξεργασία τῶν ἀποτελεσμάτων τῆς μεγάλης ἀπογραφῆς τοῦ πληθυσμοῦ πού ἔγινε τό 1890.

Ὁ Hollerith ἐπινόησε ἓνα μηχανικό σύστημα ἐγγραφῆς, ἐπεξεργασίας καί πινακοποίησεως τῶν ἀποτελεσμάτων. Ἡ βάση τῆς λειτουργίας τοῦ συστήματος ἦταν ἡ χρησιμοποίηση ταινιῶν χαρτιοῦ πάνω στίς ὁποῖες εἶχαν διατηρηθεῖ τά διάφορα στοιχεῖα σύμφωνα μέ κάποια ὁρισμένη κωδικοποίηση. Τό νέο αὐτό σύστημα βελ-

τίωσε αίσθητά την ακρίβεια των αποτελεσμάτων και μείωσε το κόστος και το χρόνο προετοιμασίας της απογραφής. Τα αποτελέσματα ήταν έτοιμα μέσα σε 2 1/2 χρόνια.

Τό 1900 περίπου ο Hollerith αντικατέστησε τις χαρτοταινίες με καρτέλες ορισμένου μεγέθους και κατασκεύασε μία διατρητική μηχανή η οποία θεωρείται ο πρόγονος των σημερινών διατρητικών μηχανών.

Στό τέλος του 1800 περίπου ο Burroughs κατασκεύασε την πρώτη άθροιστική μηχανή (σχ. 1.26).



Σχ. 1.26.

Ἡ άθροιστική μηχανή του Burroughs.

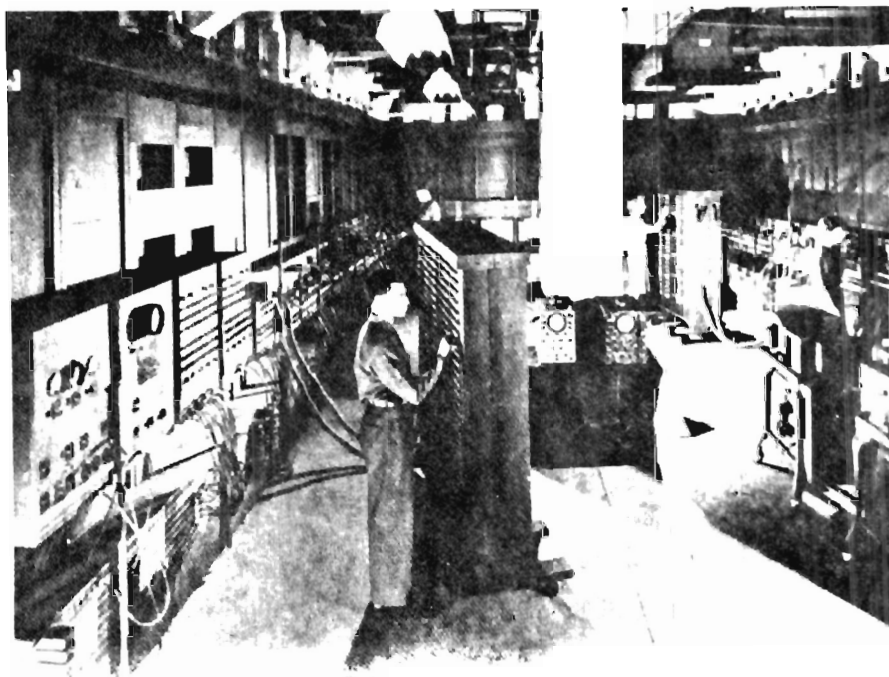
1.3 Ἡλεκτρονική επεξεργασία δεδομένων — Ἐξέλιξη ηλεκτρονικῶν ὑπολογιστῶν.

Ἡ ηλεκτρονική επεξεργασία δεδομένων επιτυγχάνεται περισσότερο με βάση τις αρχές της Ἡλεκτρονικής καί ὄχι με τή βοήθεια τῶν μηχανικῶν ἢ ηλεκτρομηχανικῶν κατασκευῶν.

Ἡ πρώτη προσπάθεια κατασκευῆς ἑνός ὑπολογιστῆ, ὁ ὁποῖος στό μεγαλύτερο μέρος τῆς ἐργασίας του θά ἦταν αυτόνομος, ἔγινε τό ἔτος 1830 στήν Ἀγγλία ἀπό τόν Charles Babbage. Ἡ προσπάθεια γενικά ἀπέτυχε ἐξαιτίας τῆς χαμηλῆς τεχνολογίας τῆς ἐποχῆς ἐκείνης, ἀλλά οἱ βασικές ἀρχές κατασκευῆς τῆς μηχανῆς ἐξακολουθοῦν ἀκόμη νά ἰσχύουν.

Τό 1939 ὁ Ἀμερικανός Samuel Williams κατασκεύασε τόν πρῶτο ηλεκτρομηχανικό ψηφιακό ὑπολογιστή στά ἐρευνητικά ἐργαστήρια Bell Telephone Laboratories. Ὁ ὑπολογιστής αὐτός χρησιμοποιοῦσε δυαδικό ἀριθμητικό σύστημα γιά τήν παράσταση τῆς πληροφορίας καί ηλεκτρονόμους γιά τήν παράσταση τῶν δυαδικῶν ψηφίων.

Ὁ πρῶτος ηλεκτρονικός ὑπολογιστής, ὁ ὀνομαζόμενος ENIAC (Electrical Numerical Integrator and Calculator) κατασκευάσθηκε στό πανεπιστήμιο τῆς Pennsylvania κάτω ἀπό τήν ἐποπτεία τῶν καθηγητῶν J.W. Eckert καί J.W. Mauchly στό διάστημα τῶν ἐτῶν 1942 καί 1946. Ἦταν ὁ πρῶτος ὑπολογιστής πού χρησιμοποιοῦσε γιά τή λειτουργία του ηλεκτρονικές λυχνίες ἀντί γιά ηλεκτρομαγνητικούς διακόπτες. Ὁ ὑπολογιστής αὐτός (σχ. 1.3α) εἶχε διαστάσεις μιᾶς μεγάλης αἰθουσας καί γιά τήν κατασκευή του χρησιμοποιήθηκαν 18.000 λυχνίες.



Σχ. 1.3α.

Ο πρώτος υπολογιστής ENIAC.

Τό 1951 κατασκευάστηκε από την εταιρία Remington ο πρώτος προοριζόμενος για έμπορικές εφαρμογές, υπολογιστής. Ονομάστηκε UNIVAC.

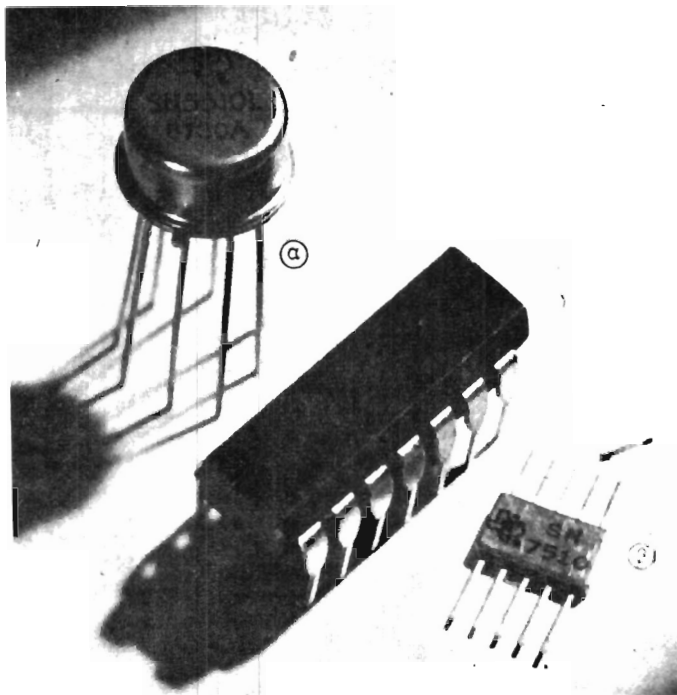
Τό 1953 η IBM (International Business Machines) κατασκεύασε τό πρώτο δικό της μοντέλο ηλεκτρονικού υπολογιστή, τόν IBM - 701.

Έκει στην αρχή περίπου τής δεκαετίας του '50 συντελούνται δύο συνταρακτικά τεχνολογικά επιτεύγματα: η ανακάλυψη των τρανζίστορς και η χρήση των μαγνητικών δακτυλίων ως στοιχείων μνήμης.

Οι πρώτοι υπολογιστές χρησιμοποιούσαν ως μνήμη μαγνητικό τύμπανο και μπορούσαν νά έκτελουν γύρω στίς 60 έντολές/sec. Η ταχύτητα αυτή ήταν γιά την εποχή εκείνη καταπληκτική. Δέν μπορεί όμως νά συγκριθεί μέ αυτή των συγχρόνων υπολογιστών πού έκτελουν πάνω από 10 έκατομμύρια (10×10^6) πράξεις/sec. Η αύξηση αυτή τής ταχύτητας μπορεί νά μεταφρασθεί και ως ελάττωση του κόστους ανά έκτελούμενη πράξη. Πράγματι και τό κόστος αυτό, μειώθηκε κατά 5.000 φορές περίπου.

Η μετέπειτα εξέλιξη των ηλεκτρονικών και ιδιαίτερα η ανάπτυξη των ψηφιακών ηλεκτρονικών και των ολοκληρωμένων κυκλωμάτων, έδωσε τεράστια ώθηση στην τεχνολογία των Ηλεκτρονικών Υπολογιστών. Τό έτος 1966 κυκλοφόρησε ο πρώτος mini-υπολογιστής τής Digital Equipment Corporation μέ τό όνομα PDP-8. Τέλος, τό 1971 η INTEL κατασκεύασε σέ ένα μόνο ολοκληρωμένο κύκλωμα την κεντρική μονάδα επεξεργασίας ενός micro - υπολογιστή (4004).

Η εξέλιξη στους Η/Υ συνεχίζεται άλματωδώς. Καθημερινά σχεδόν αναγγέλλονται νέες κατασκευές και νέες εξελίξεις. Στο σχήμα 1.3β φαίνεται η εξέλιξη κατα-



Σχ. 1.3β.

α) Σύγκριση μεγέθους ενός τρανζίστορ και ενός ολοκληρωμένου κυκλώματος. β) Έπίπεδο ολοκληρωμένο κύκλωμα.



Σχ. 1.3γ.

Ένας μικρο-ύπολογιστής τύπου τσέπης ο οποίος προγραμματίζεται σε γλώσσα Basic και του οποίου τό πρόγραμμα μπορεί να εγγραφεί σε κασέτα ή σε μνήμη CMOS.

σκευής ηλεκτρονικών εξαρτημάτων, ενώ στο σχήμα 1.3γ φαίνεται ένας μικροϋπολογιστής τύπου τσέπης.

Τό παράδειγμα της κατασκευής από την INTEL μικρο - υπολογιστών σε ένα μόνο κύκλωμα ακολούθησαν και άλλοι κατασκευαστές. Έτσι έχουμε τον Z - 80 της ZI-

LOG καί τόν 6800 τής MOTOROLA.

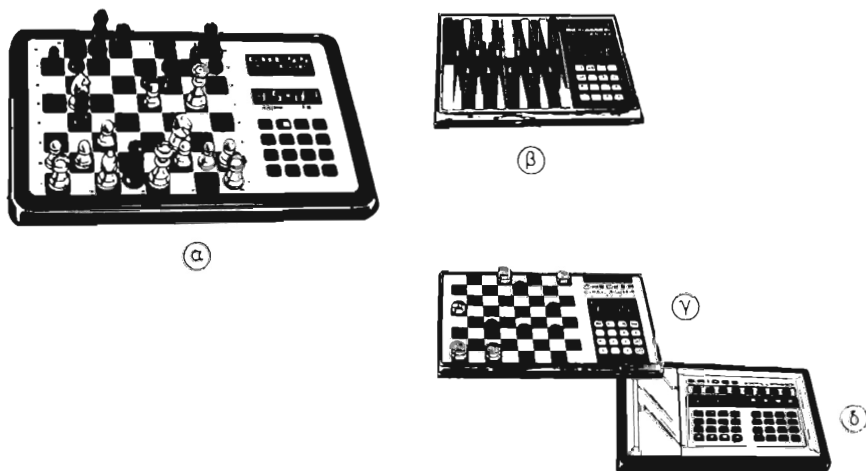
Ἀρχικῶς οἱ μικροϋπολογιστές αὐτοί εἶχαν μήκος λέξεως 4 δυαδικῶν ψηφίων (bits). Χρησιμοποιοῦνταν δηλαδή 4 δυαδικά ψηφία γιά τήν ἀπεικόνιση ἑνός ψηφίου ἢ ἀριθμοῦ. Στή συνέχεια τό μήκος τῆς λέξεως αὐξήθηκε σέ 8 δυαδικά ψηφία.

Ἡ Digital Equipment Corporation κατασκεύασε τόν μίνι ὑπολογιστή LSI - 1* μέ μήκος λέξεως 16 δυαδικά ψηφία σέ ἕνα ὁλοκληρωμένο κύκλωμα.

Ἔτσι καθιερώθηκαν οἱ ὅροι μικροϋπολογιστής γιά ὑπολογιστές μέ μήκος λέξεως 4 - 8 δυαδικά ψηφία καί μίνι-ὑπολογιστής γιά ὑπολογιστές μέ μήκος λέξεως 16 δυαδικά ψηφία.

Ἡ ἐξέλιξη τῆς τεχνολογίας ὅμως ἐπέτρεψε τήν κατασκευή μικρο - ὑπολογιστῶν μέ μήκος λέξεως 16 δυαδικῶν ψηφίων, ὅπως εἶναι ὁ Z8000 τῆς Zilog καί ὁ 68000 τῆς Motorola. Τελευταία μάλιστα ἔχει ἀνακοινωθεῖ μικρο - ὑπολογιστής μέ μήκος λέξεως 32 δυαδικῶν ψηφίων.

Περисσότερα ὅμως γιά τήν ἐξέλιξη ἀλλά καί τή χρήση τῶν μικρο - ὑπολογιστῶν θά ποῦμε ἀργότερα στά ἐπόμενα κεφάλαια. Ἄς σημειωθεῖ ἐδῶ ὅτι ἡ χρήση τῶν Η/Υ ἔχει ἐπεκταθεῖ καί σέ διάφορα παιχνίδια ὅπως εἶναι ἠλεκτρονικές σκακιέρες, παγινίδια UFO κλπ. (σχ. 1.3δ).



Σχ. 1.35.

Διάφορα ἠλεκτρονικά παγινίδια τά ὁποῖα περιέχουν μικρο-ὑπολογιστές. α) Ἐλεκτρονική σκακιέρα. β) Ἐλεκτρονικό τάβλι. γ) Ἐλεκτρονική ντάμα. δ) Ἐλεκτρονικό παγινίδι Bridge.

1.4 Κατηγορίες Ἐλεκτρονικῶν Ὑπολογιστῶν.

Οἱ ἠλεκτρονικοὶ ὑπολογιστές ἀνάλογα μέ τή μορφή τῆς πληροφορίας πού δέχονται στήν εἴσοδο καί τήν ἐπεξεργασία στήν ὁποία τήν ὑποβάλλουν, διαίροῦνται σέ τρεῖς κατηγορίες:

- Στούς ἀναλογικοὺς ὑπολογιστές (σχ. 1.4α)
- Στούς ψηφιακοὺς ὑπολογιστές (σχ. 1.4β) καί
- Στούς μικτοὺς ἢ ὑβριδικούς ὑπολογιστές.



Σχ. 1.4α.

Σύγχρονος αναλογικός υπολογιστής.

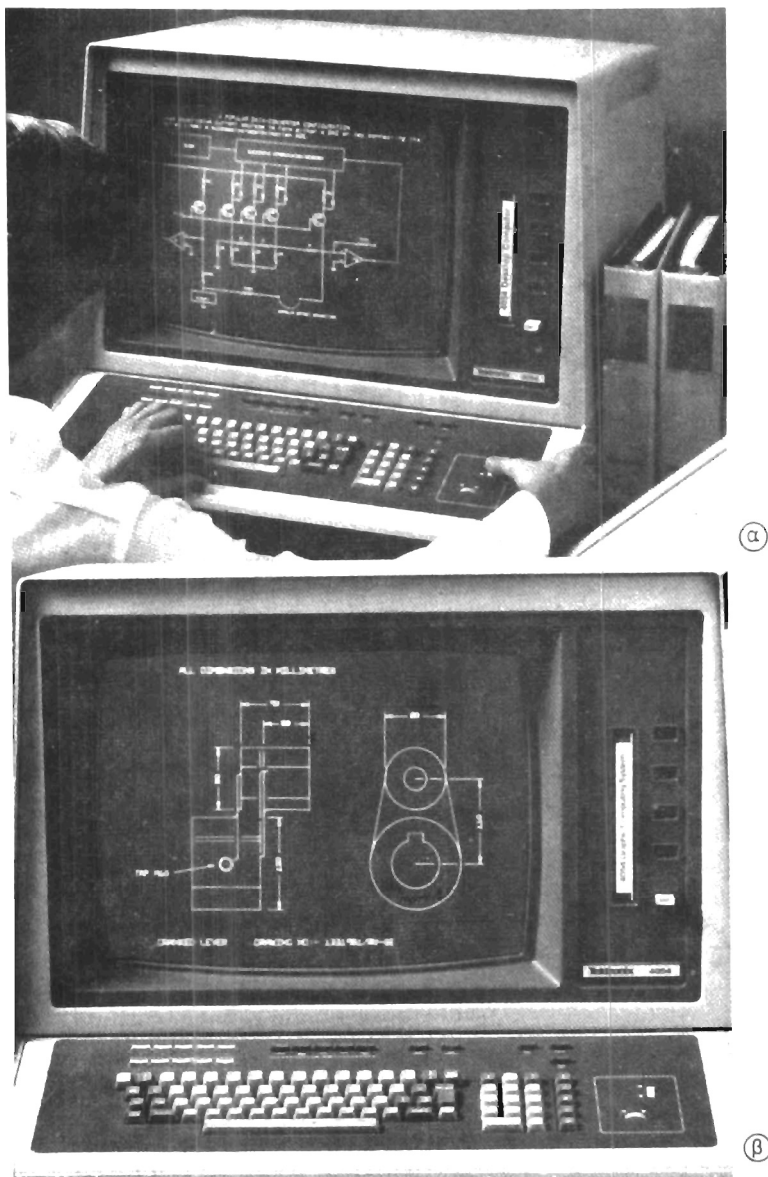
1.4.1 Αναλογικοί Υπολογιστές (σχ. 1.4α).

Με τούς αναλογικούς υπολογιστές μελετούμε φυσικά συστήματα τά όποια περιγράφονται μέ διαφορικές εξισώσεις (π.χ. ή λειτουργία τής γυροπυξίδας). Οί διάφορες ποσότητες παριστάνονται αναλογικά μέ κάποιο άλλο φυσικό μέγεθος τό όποίο συνήθως είναι τό ήλεκτρικό δυναμικό. Τό φυσικό σύστημα πού μελετούμε εξομοιώνεται μέ ένα άλλο Ισοδύναμο ήλεκτρικό κύκλωμα. Έτσι, επιλύοντας τό Ισοδύναμο αυτό ήλεκτρικό κύκλωμα μπορούμε νά βγάλουμε συμπεράσματα γιά τή συμπεριφορά τού πραγματικού φυσικού συστήματος.

Οί αναλογικοί υπολογιστές, τών όποιών τό βασικό στοιχείο δομής είναι ό τελεστικός ένισχυτής (operational amplifier) είναι γενικά γρήγοροι καί άπλοί στή λειτουργία τους. Ή ακρίβεια όμως τών αποτελεσμάτων πού μās δίνουν είναι περιορισμένα.

1.4.2 Ψηφιακοί Υπολογιστές (σχ. 1.4β).

Οί αριθμοί στους ψηφιακούς υπολογιστές παριστάνονται μέ τά αντίστοιχα ψηφία τους (συνήθως τά δυαδικά). Ή λειτουργία τους συνίσταται στή διαδοχική εκτέλεση στοιχειωδών έντολών επί τών αριθμών αυτών, πού δίνονται από τόν άνθρωπο μέ τή μορφή προγράμματος.



Σχ. 1.48.

Ή χρήση ενός Η/Υ στη σχεδίαση γραφικών παραστάσεων.
α) Σχεδίαση λογικών κυκλωμάτων. β) Σχεδίαση μιάς ΜΕΚ.

Οι ψηφιακοί υπολογιστές έχουν άσυγκρίτως μεγαλύτερες δυνατότητες από τους αναλογικούς και εφαρμόζονται σε πολύ μεγαλύτερη κατηγορία προβλημάτων, ακόμα και προβλημάτων που επιλύονται από τους αναλογικούς υπολογιστές.

Ένας διαχωρισμός, που γίνεται συνήθως στους ψηφιακούς ηλεκτρονικούς υπολογιστές, είναι σε υπολογιστές γενικής και ειδικής χρήσεως.

Οι υπολογιστές γενικής χρήσεως σχεδιάζονται για να καλύψουν ένα ευρύ φάσμα εφαρμογών, αντίθετα προς τους ειδικής χρήσεως οι οποίοι σχεδιάζονται για μία συγκεκριμένη εφαρμογή. Η διαφορά τους, ανάλογα με τη χρήση, αναφέρεται κυρίως στο μέγεθος και την ταχύτητα της μνήμης, καθώς και στον αριθμό των περιφερειακών μονάδων που μπορούν να εξυπηρετήσουν.

1.4.3 Υβριδικοί ή Μικτοί Υπολογιστές.

Στους υβριδικούς υπολογιστές συνδυάζονται τα πλεονεκτήματα και των αναλογικών και των ψηφιακών υπολογιστών. Έτσι τό ένα τμήμα του υβριδικού υπολογιστή είναι αναλογικός υπολογιστής και τό άλλο ψηφιακός υπολογιστής που δέχεται στοιχεία από τόν αναλογικό, τά μετατρέπει σέ ψηφιακή μορφή και τά επεξεργάζεται. Στην έξοδο γίνεται συνήθως ή αντίστροφη μετατροπή. Για τή μετατροπή από τήν αναλογική παράσταση των στοιχείων στην ψηφιακή και αντίστροφα, χρησιμοποιούνται ειδικοί μετατροπείς οι οποίοι ονομάζονται 'Αναλογοψηφιακοί Μετατροπείς, γνωστοί διεθνώς ως Analog to Digital ή Digital to Analog Converters.

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

ΑΛΓΟΡΙΘΜΟΙ - ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ

2.1 Τί είναι αλγόριθμος.

Μέ απλά λόγια θά μπορούσαμε νά πούμε ότι αλγόριθμος είναι ένα σύνολο απλών διαδικασιών, οι οποίες, εκτελούμενες μέ κάποια καθορισμένη σειρά, πού μπορεί νά περιέχει καί επαναλήψεις, μās οδηγούν στή λύση ενός προβλήματος.

Ό τρόπος π.χ. πού πολλαπλασιάζομε δύο αριθμούς ή ο τρόπος εύρεσεως του λογαρίθμου ενός αριθμού είναι ένας αλγόριθμος. Οι μαθηματικές πράξεις μπορούν νά εκτελεσθούν καί σχεδόν κάθε πρόβλημα μπορεί νά λυθεί μέ τή βοήθεια καταληλών αλγορίθμων.

2.2 Διαγράμματα ροής.

Όπως είπαμε προηγουμένως, ο αλγόριθμος αποτελείται από τά βήματα τά όποια θά πρέπει νά ακολουθηθούν γιά τήν επίλυση ενός προβλήματος. Γιά νά δοθεί ένας αλγόριθμος υπάρχουν διάφοροι τρόποι από τούς όποιους οι συνηθέστεροι είναι οι εξής:

- Περιγραφή κάθε βήματος μέ λόγια.
- Περιγραφή κάθε βήματος μέ συμβολικό διάγραμμα ροής.

Η χρήση τής πρώτης μεθόδου είναι περισσότερο πλήρης. Είναι όμως αρκετά δύσκολο νά μεταφράσει κανείς αυτή τήν περιγραφή σέ ένα πρόγραμμα ηλεκτρονικού υπολογιστή. Πρός βοήθεια λοιπόν των προγραμματιστών έχει καθιερωθεί ή χρήση διαγραμμάτων ροής, τά όποια δέν είναι τίποτε άλλο παρά μία πινακοποίηση πού δείχνει μέ άμεσο τρόπο πώς πρέπει νά προγραμματισθεί ή λύση του προβλήματος.

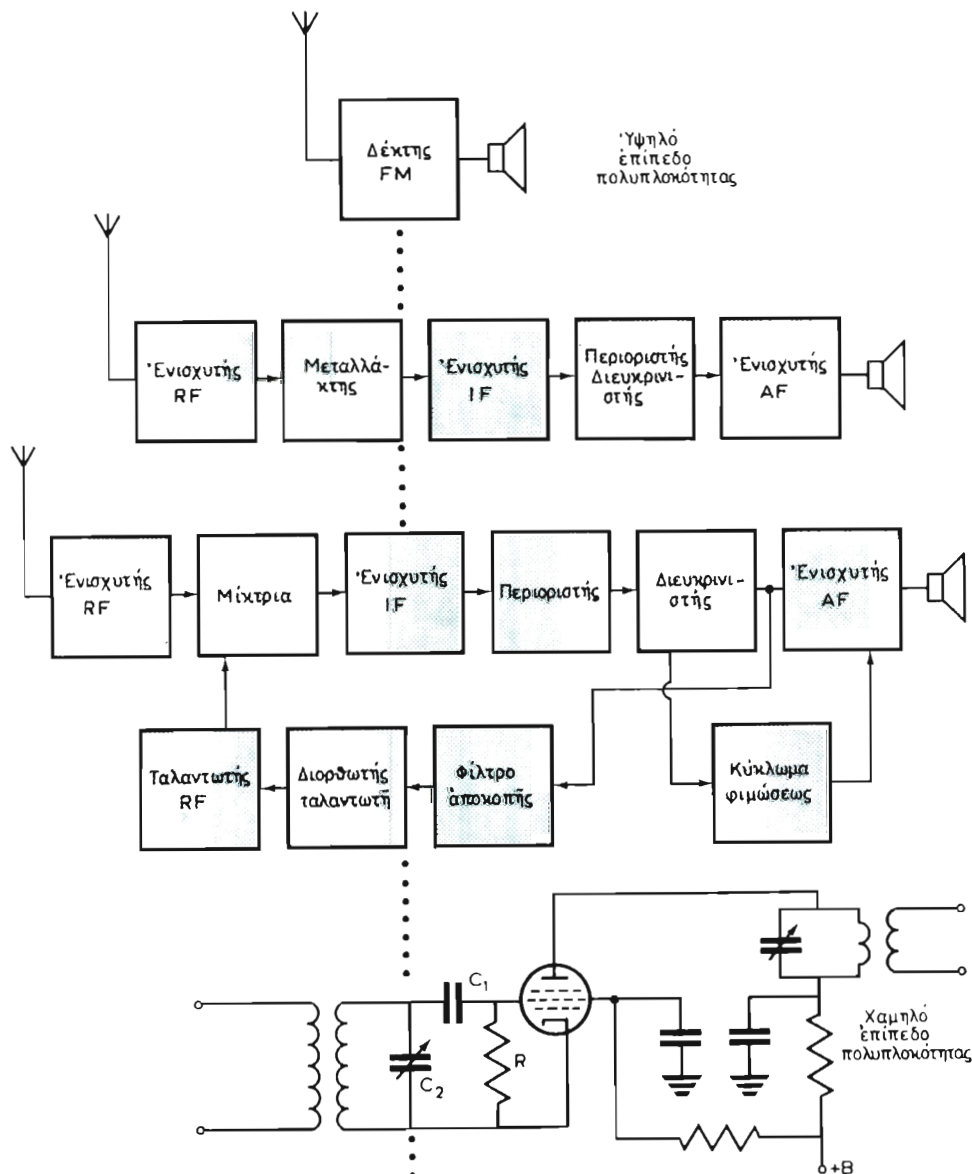
Όταν έχομε νά γράφομε ένα σύνθετο πρόγραμμα, τό διάγραμμα ροής μπορεί νά χρησιμοποιηθεί γιά τή διαίρεση του όλου προβλήματος σέ πιό απλά ανεξάρτητα προβλήματα τά όποια τελικώς πρέπει νά συνδέονται μεταξύ τους. Έτσι κάθε ανεξάρτητο πιά πρόβλημα μπορεί νά ανατεθεί σέ διαφορετικό προγραμματιστή. Ό κάθε προγραμματιστής θά πρέπει νά γνωρίζει ότι διάφορα μέρη του προγράμματος του πιθανόν νά έχουν περιληφθεί στό πρόγραμμα κάποιου άλλου προγραμματιστή ως μέρος τής εργασίας του ή ως ανεξάρτητη εργασία.

Στήν περίπτωση αυτή τό γενικό διάγραμμα ροής μπορεί νά χρησιμοποιηθεί γιά τήν ανεύρεση λαθών στή σύνδεση των προγραμμάτων καί ακόμη νά υποβοηθήσει στή διόρθωσή τους είτε στό αρχικό στάδιο είτε στό μελλοντικό, όταν θά αλλάξουν οι απαιτήσεις του προβλήματος.

2.2.1 Επίπεδα διαγραμμάτων ροής.

Τά διάφορα διαγράμματα ροής μπορούν νά κατασκευασθούν σέ διάφορα επίπεδα πολυπλοκότητας. Ένα διάγραμμα ροής ύψηλου επιπέδου συνίσταται από πολύ μικρό αριθμό συμβόλων καί παριστάνει μία πολύ γενική όψη του προβλήματος, έ-

νώ ένα διάγραμμα ροής χαμηλού επιπέδου μπορεί να φθάσει σε τέτοιο σημείο, ώστε να υπάρχει άμεση αντίστοιχία μεταξύ των συμβόλων του διαγράμματος ροής και των πραγματικών εντολών ενός υπολογιστή. Συνήθως υπάρχουν πολλά διαγράμματα ροής για μία περιοχή ενός προβλήματος διαφόρων επιπέδων πολυπλοκότητας. Τα επίπεδα του διαγράμματος ροής και τα επίπεδα περιγραφής ενός συστήματος, π.χ. ενός δέκτη FM, είναι περίπου τα ίδια, όπως φαίνεται στα σχήματα 2.2α και 2.2β.

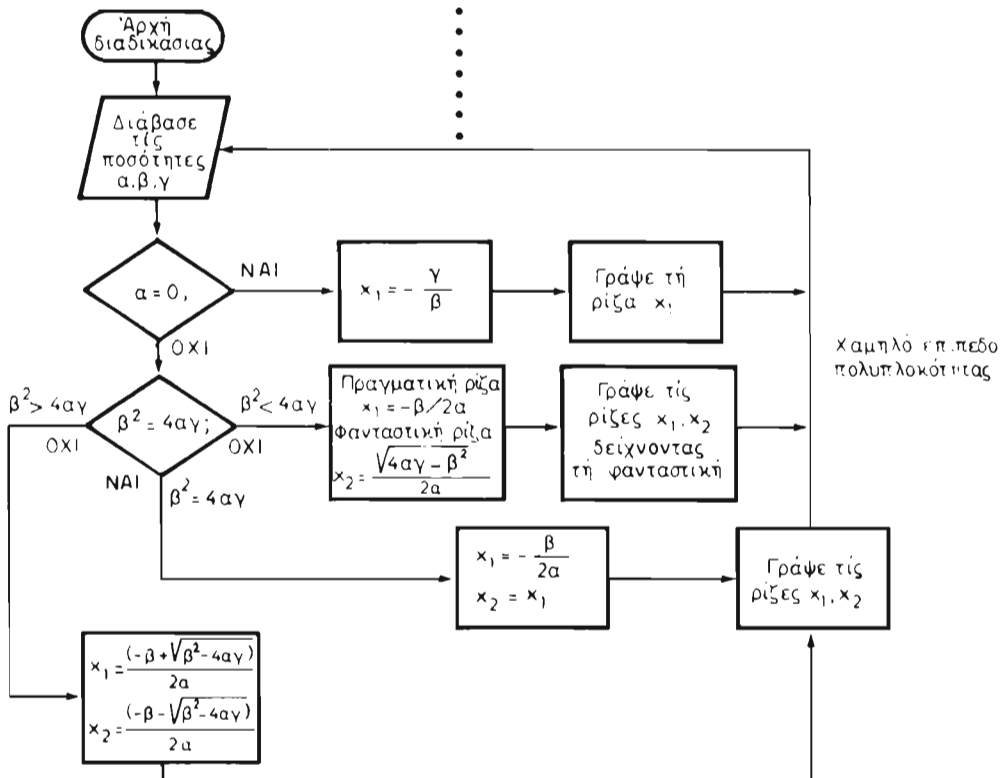
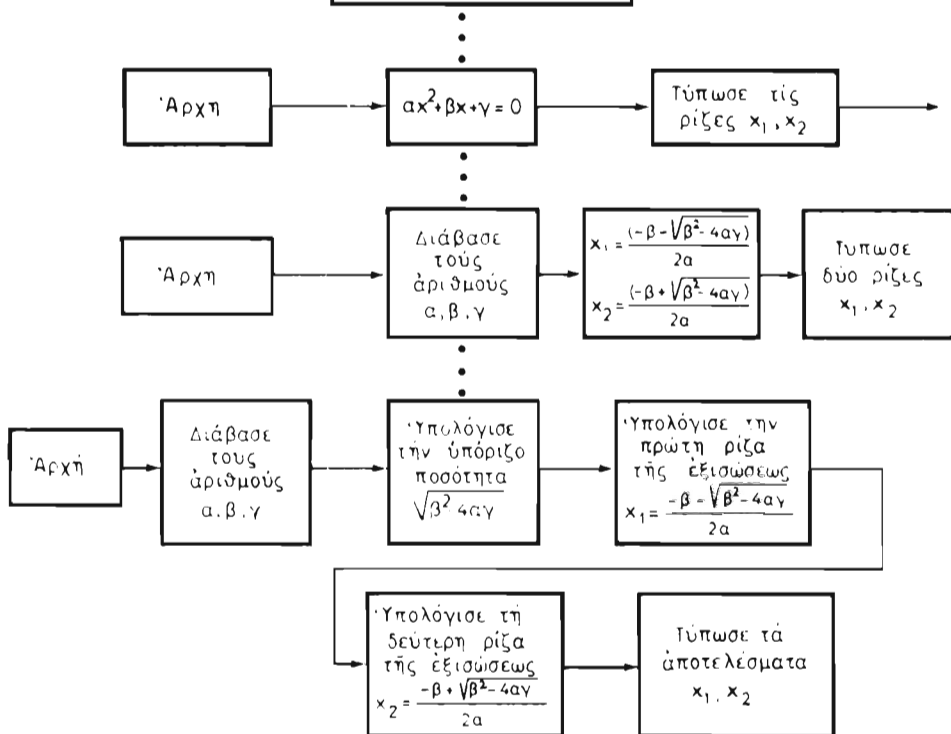


Σχ. 2.2α.

Παράσταση ανάπτυξης ενός λειτουργικού διαγράμματος δέκτη FM από την πιο πολύπλοκη στην πιο απλή μορφή.

Να επιλυθεί ή δευτεροβάθμια
έξισωση $ax^2 + bx + \gamma = 0$

Υψηλό επίπεδο
πολυπλοκότητας



Σχ. 2.2β.

Τά επίπεδα πολυπλοκότητας διαγράμματος ροής.

Τά λειτουργικά διαγράμματα ροής ενός προβλήματος πού πρόκειται νά δοθεῖ στόν προγραμματιστή θά πρέπει νά αναλύονται πρῶτα γιά νά γνωρίζει ὁ προγραμματιστής μέ ποιά σειρά καί ποιό τρόπο θά ἐπιλυθεῖ τό πρόβλημα καί ὕστερα γιά νά μπορέσομε νά κάνομε ἀργότερα διορθώσεις τοῦ προγράμματος ἐφ' ὅσον ἀπαιτηθοῦν.

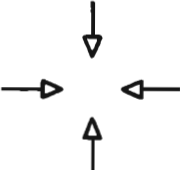
Ἡ γενική ἀρχή πάντως εἶναι ὅτι ὅσο πιά πολύπλοκο εἶναι τό πρόβλημα, τόσο πιά σπουδαία σημασία ἀπαιτεῖ τό λειτουργικό διάγραμμα ροής.

2.2.2 Σύμβολα διαγράμματος ροής.

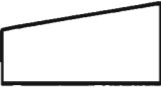
Γιά νά ὑπάρχει κάποια λογική συνέχεια τῆς περιγραφῆς τοῦ τρόπου ἐπιλύσεως τοῦ προβλήματος μέσω τοῦ διαγράμματος ροής χρησιμοποιοῦμε διάφορα σύμβολα. Τά σύμβολα αὐτά φαίνονται στόν πίνακα 2.2.1.

ΠΙΝΑΚΑΣ 2.2.1.

Τά κυριότερα σύμβολα γιά τό σχεδιασμό τῶν λογικῶν διαγραμμάτων ροής.

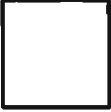
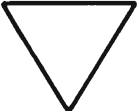
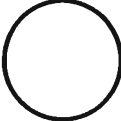
Σύμβολο	Περιγραφή λειτουργίας του
	Ἀρχή ἢ τέλος ἐπεξεργασίας.
	Ἀπλή λειτουργία ἐπεξεργασίας.
	Γενικό σύμβολο γιά εἴσοδο - ἀνάγνωση ἢ ἔξοδο - ἐγγραφή ἀποτελεσμάτων.
	Βέλη πού δείχνουν τή ροή τῆς πληροφορίας.
	Ἀπόφαση, Ἑρώτηση, Σύγκριση.

(συνεχίζεται)

Σύμβολο	Περιγραφή λειτουργίας του
	Άνάγνωση δεδομένων από διάτρητα δελτία.
	Άνάγνωση δεδομένων από διάτρητη ταινία.
	Απεικόνιση σέ όθόνη τηλεόρασεως.
	Έγγραφή ή ανάγνωση από άρχείο μαγνητικού δίσκου.
	Αποθήκευση πληροφοριών ON - LINE (άπευθείας σύνδεση Η/Υ καί χειριστή).
	Χειροκίνητη είσοδος στοιχείων (πληκτρολόγιο).
	Δίαυλος έπικοινωνίας (Communication link).
	Έκτύπωση στοιχείων σέ χαρτί (Printer).

(συνεχίζεται)

Τά κυριότερα σύμβολα για τό σχεδίασμό τών λογικών διαγραμμάτων ροής.

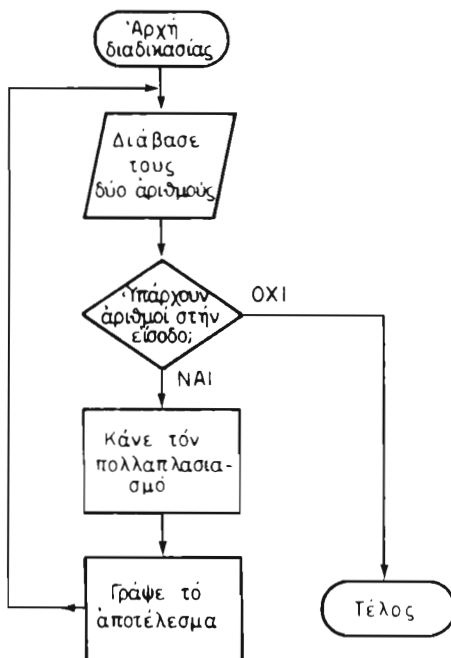
Σύμβολο	Περιγραφή λειτουργίας του
	Βοθητική λειτουργία.
	Αποθήκευση πληροφοριών OFF - LINE (δέν υπάρχει σύνδεση μεταξύ Η/Υ καί χειριστή).
	Μαγνητική ταινία αποθηκεύσεως - αναγνώσεως πληροφοριών.
	Μαγνητικός δίσκος αποθηκεύσεως - αναγνώσεως πληροφοριών.
	Τροποποίηση ή μεταβολή προγράμματος.
	Διαλογή - ταξινόμηση (Sort).
	Σημείο συνδέσεως.
	Μαγνητικό τύμπανο αποθηκεύσεως - αναγνώσεως πληροφοριών.

Γιά τήν καλύτερη κατανόηση τῆς χρήσεως τῶν παραπάνω συμβόλων, θά παραθέσουμε παρακάτω διάφορα παραδείγματα σχεδιασμοῦ διαγραμμάτων ροῆς.

Παράδειγμα 1.

Ζητεῖται τό γινόμενο δύο ἀριθμῶν. Ἀφοῦ βρεθεῖ τό γινόμενο, ζητεῖται ἡ ἐγγραφή τοῦ ἀποτελέσματος καί στή συνέχεια ἀναζητεῖται ἕνα δεύτερο ζεῦγος ἀριθμῶν γιά ἕνα καινούργιο πολλαπλασιασμό. Ἄν δέν ὑπάρχει ἄλλο ζεῦγος ἡ ἐπεξεργασία τελειώνει.

Στήν περίπτωση αὕτη τό διάγραμμα ροῆς θά εἶναι ὅπως παρακάτω (σχ. 2.2γ).



Σχ. 2.2γ.

Βῆμα πρῶτο: Διαβάζομε τά δεδομένα (ἂν δέν ὑπάρχουν, τοῦτο ἀναγνωρίζεται καί τό πρόγραμμα τελειώνει).

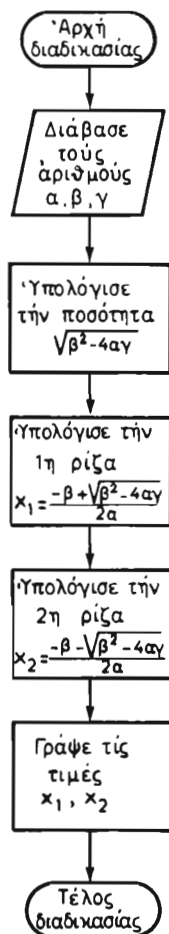
Βῆμα δεύτερο: Οἱ δύο ἀριθμοί πολλαπλασιάζονται.

Βῆμα τρίτο: Γράφεται τό ἀποτέλεσμα.

Βῆμα τέταρτο: Ξαναγυρίζομε στήν ἀρχή γιά νά διαβάσομε νέους ἀριθμούς καί νά ἐκτελέσομε καί πάλι τό πρόγραμμα.

Παράδειγμα 2.

Ας υποθέσουμε ότι επιθυμούμε να υπολογίσουμε τις ρίζες της εξισώσεως $ax^2 + \beta x + \gamma = 0$ μέσω της γνωστής σχέσεως $x = (-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma})/2\alpha$. Για ευκολία θεωρούμε ότι η ποσότητα $(\beta^2 - 4\alpha\gamma)$ είναι πάντοτε θετική. Το διάγραμμα ροής θα είναι όπως παρακάτω (σχ. 2.26).

**Σχ. 2.26.**

Βήμα πρώτο: Διάβασε τις ποσότητες α , β , γ .

Βήμα δεύτερο: Υπολόγισε την ποσότητα $\sqrt{\beta^2 - 4\alpha\gamma}$.

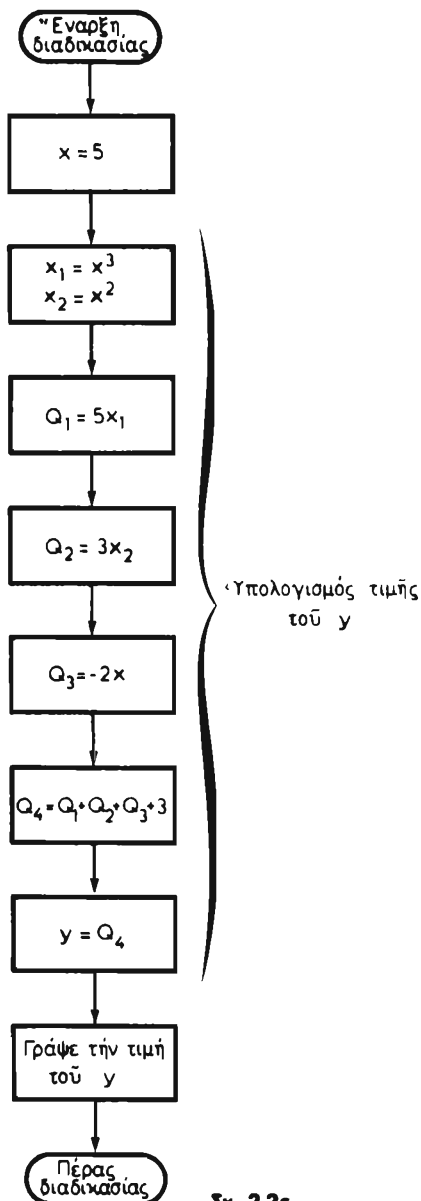
Βήμα τρίτο: Υπολόγισε τις ρίζες x_1 , x_2 .

Βήμα τέταρτο: Γράψε στην έξοδο τις τιμές των ριζών της εξισώσεως.

Βήμα πέμπτο: Τέλος διαδικασίας.

Παράδειγμα 3.

Σχεδιάστε ένα διάγραμμα ροής με το οποίο θα βρεθεί η τιμή της συνάρτησεως $y = 5x^3 + 3x^2 - 2x + 3$ για την τιμή $x = 5$. Το διάγραμμα ροής θα είναι όπως παρακάτω (σχ. 2.2ε).

**Σχ. 2.2ε.**

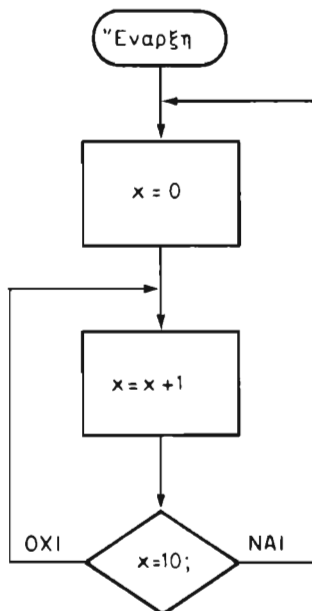
Βήμα πρώτο: Δίνουμε την τιμή 5 στην ανεξάρτητη μεταβλητή x .

Βήμα δεύτερο: Υπολογίζουμε την τιμή του y .

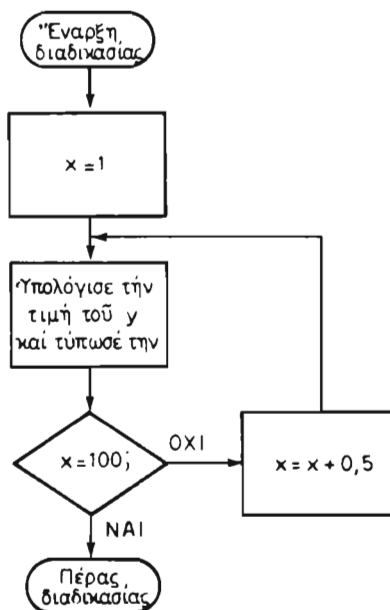
Βήμα τρίτο: Τυπώνουμε το αποτέλεσμα.

Παράδειγμα 4.

Σχεδιάστε το διάγραμμα ροής μέσω του οποίου μπορούμε να μετρήσουμε από το 0 μέχρι το 10 και στη συνέχεια να επαναλάβουμε τη μέτρηση. Το διάγραμμα ροής που θα σχεδιασθεί φαίνεται στο σχήμα 2.2στ.



Σχ. 2.2στ.



Σχ. 2.2ζ.

Παράδειγμα 5.

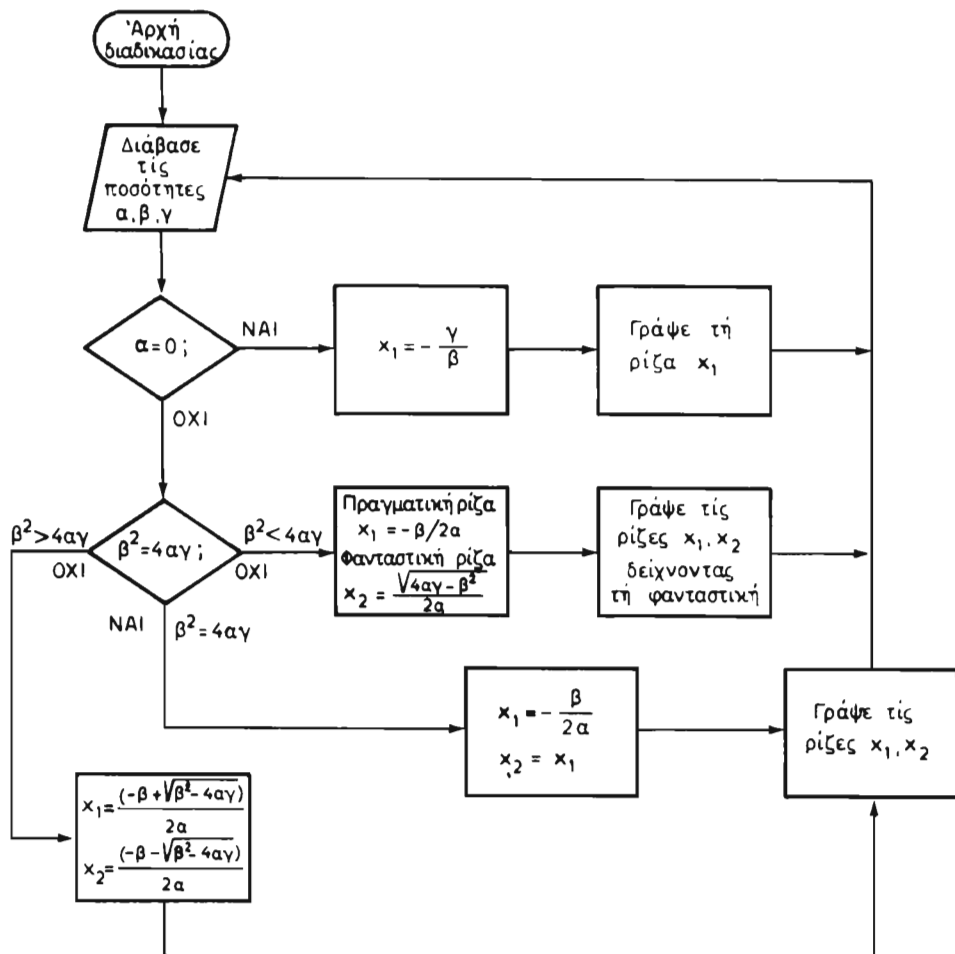
Ας δοκιμάσουμε τώρα να βρούμε τιμές της προηγούμενης συνάρτησεως $y = 5x^3 + 3x^2 - 2x + 3$ για τιμές του x από 1 έως το 100, αύξάνοντας το x κάθε φορά κατά 0.5. Δηλαδή η ανεξάρτητη μεταβλητή x θα παίρνει τις τιμές 1, 1.5, 2, 2.5, ..., 99.5, 100. Το διάγραμμα ροής που θα σχεδιασθεί φαίνεται στο σχήμα 2.2ζ. Παρατηρήστε ότι όλη η διαδικασία υπολογισμού της τιμής της συνάρτησεως του παραδείγματος 3, εδώ περιλαμβάνεται σαν υπομύημα του όλου διαγράμματος ροής.

Παράδειγμα 6.

Ής προσπαθήσουμε τώρα νά σχεδιάσουμε ένα πλήρες διάγραμμα ροής γιά τήν επίλυση τής δευτεροβάθμιας εξισώσεως, χωρίς όμως νά έχομε τόν περιορισμό ότι ή υπόριξη ποσότητα $\beta^2 - 4\alpha\gamma$ θά πρέπει νά είναι θετική.

Στήν περίπτωση αύτή εκτός από τά βήματα πού ακολουθήσαμε στό προηγούμενο παράδειγμα, θά πρέπει νά ελέγξομε κατά πόσον ή υπόριξη ποσότητα είναι μεγαλύτερη, μικρότερη ή ίση μέ τό μηδέν.

Τό διάγραμμα ροής πού θά κατασκευασθεί φαίνεται στό σχήμα 2.2η.



Σχ. 2.2η.

2.3 Έρωτήσεις.

1. Τί είναι αλγόριθμος και ποῦ χρησιμοποιεῖται;
2. Σέ τί μᾶς βοηθᾷ ἓνα διάγραμμα ροῆς;

2.4 Ασκήσεις.

1. Νά σχεδιασθεῖ ἓνα λογικό διάγραμμα ροῆς γιά τόν ὑπολογισμό τοῦ γινομένου τῶν ἀριθμῶν ἀπό $1 \div 15$ (δηλαδή $1 \times 2 \times 3 \times \dots \times 15$).
2. Νά σχεδιασθεῖ ἓνα λογικό διάγραμμα ροῆς τό ὁποῖο θά διαβάζει τίς τιμές τριῶν ἀριθμῶν (Α,Β,Γ) θά ὑπολογίζει τό ἄθροισμα τῶν τετραγώνων τους καί στή συνέχεια θά διαιρεῖ τό ἀποτέλεσμα αὐτό διά τοῦ ἀριθμοῦ $\pi = 3.14159$ καί θά τό τυπώνει. Ἡ διαδικασία αὐτή θά πρέπει νά ἐπαναλαμβάνεται ἐφόσον $A \neq B \neq \Gamma \neq 0$.
3. Νά σχεδιασθεῖ ἓνα λογικό διάγραμμα ροῆς τό ὁποῖο θά ὑπολογίζει τόν μέσο ὄρο 25 ἀριθμῶν τοῦ ὁποῖου θά διαβάζει τόν καθένα ξεχωριστά.
4. Νά σχεδιασθεῖ τό λογικό διάγραμμα ροῆς μέ τό ὁποῖο ὑπολογίζομε τήν τιμή τῆς παραστάσεως:

$$x = 57\gamma^2 + 17\gamma + 28.2$$

γιά τιμές τοῦ γ ἀπό 0 ὥς τό 10, ἀνεβαίνοντας κατά 0.2 (δηλαδή $\gamma = 0, 0.2, 0.4, 0.6, \dots, 10$).

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

ΑΡΙΘΜΗΤΙΚΑ ΣΥΣΤΗΜΑΤΑ – ΚΩΔΙΚΕΣ

3.1 Αριθμητικά συστήματα.

Τό δεκαδικό αριθμητικό σύστημα πού χρησιμοποιούμε σήμερα τό πήραμε από τούς Άραβες.

Είναι τό πλέον διαδεδομένο σύστημα στίς καθημερινές συναλλαγές καί οι περισσότεροι άνθρωποι νομίζουν ότι είναι καί τό μοναδικό σύστημα αριθμήσεως πού υπάρχει.

Όπως θά δοῦμε ὅμως στό κεφάλαιο αὐτό, υπάρχουν καί ἄλλα ἀριθμητικά συστήματα ὅπως τό δυαδικό, τό ὀκταδικό καί τό δεκαεξαδικό. Τά συστήματα αὐτά ἔχουν πλεονεκτήματα καί μειονεκτήματα σέ σχέση μέ τό δεκαδικό, ἔχουν ὅμως τό κοινό χαρακτηριστικό ὅτι εἶναι καί αὐτά συστήματα μέ «βάση».

Γιά νά μπορούμε νά ξεχωρίζουμε σέ ποιο ἀριθμητικό σύστημα ἐκφραζόμαστε, θά χρησιμοποιούμε ἀπό ἐδῶ καί πέρα τό συμβολισμό $A_{(r)}$. Μέ τό γράμμα A συμβολίζουμε τόν ἀριθμό καί μέ τό γράμμα (r) τό σύστημα ἀριθμήσεως τό ὁποῖο χρησιμοποιούμε. Αὐτό ὀνομάζεται καί «βάση» ἀριθμήσεως. Παραδείγματος χάρι οἱ ἀριθμοί $15_{(10)}$ καί $15_{(8)}$ μέ τήν πρώτη ματιά φαίνονται ἴδιοι, ὅμως εἶναι τελείως διαφορετικοί ὅπως θά δοῦμε στό κεφάλαιο αὐτό. Αὐτοί διαβάζονται ὡς ἑξῆς:

$15_{(10)}$ Ὁ ἀριθμός 15 μέ βάση τό 10.

$15_{(8)}$ Ὁ ἀριθμός 15 μέ βάση τό 8.

Ἡ περιγραφή τῶν διαφόρων αὐτῶν ἀριθμητικῶν συστημάτων πού ἀκολουθεῖ θά μᾶς δώσει τά ἀπαραίτητα στοιχεῖα γιά τήν κατανόηση τῆς λειτουργίας τῶν ηλεκτρονικῶν ὑπολογιστῶν γιά τήν ὁποία θά κάνουμε λόγο στά ἐπόμενα κεφάλαια.

3.1.1 Τό δεκαδικό σύστημα ἀριθμήσεως.

Τό δεκαδικό σύστημα ἀριθμήσεως περιλαμβάνει δέκα ψηφία, τά 0, 1, 2,... 9. Ἡ ἀξία κάθε ψηφίου ἐξαρτᾶται ἀπό τή θέση πού κατέχει μέσα σέ ἕναν ἀριθμό. Ἔτσι ὁ ἀριθμός 873 καί ὁ ἀριθμός 783 δέν εἶναι ἴδιοι, ἀν καί περιέχουν τά ἴδια ψηφία ἀλλά σέ διαφορετική θέση. Στό δεκαδικό σύστημα ὁ ἀριθμός 10 ὀνομάζεται βάση τοῦ συστήματος.

Όταν ἐξετάζουμε ἕναν ἀριθμό στό δεκαδικό σύστημα ἀριθμήσεως, σπανίως ἐξετάζουμε τή σύνθεση τοῦ ἀριθμοῦ. Ὁ ἀριθμός 642 π.χ. περιέχει 6 ἑκατοντάδες, 4 δεκάδες καί 2 μονάδες.

Μπορεῖ ὅμως νά διατυπώνεται καί ἄλλοιῶς. Δηλαδή:

$$642 = 6 \times 10^2 + 4 \times 10^1 + 2 \times 10^0$$

$$= 6 \text{ ἑκατοντάδες} + 4 \text{ δεκάδες} + 2 \text{ μονάδες}$$

$$\text{ὅπου } 10^2 = 100, 10^1 = 10 \text{ καί } 10^0 = 1$$

Έτσι γενικότερα ένας αριθμός N μπορεί να γραφεί ως εξής:

$$N = A_n \times r^n + A_{n-1} \times r^{n-1} + \dots + A_2 \times r^2 + A_1 \times r^1 + A_0 \times r^0$$

όπου n είναι η τάξη και A_n το ψηφίο στην κάθε θέση.

Η βάση του συστήματος (r) και ο δείκτης (εκθέτης) δίνουν την τιμή της θέσεως του ψηφίου σε σχέση με τα άλλα ψηφία του αριθμού. Ο παράγοντας A_n μπορεί να πάρει στο δεκαδικό σύστημα τις τιμές 0, 1, 2, ..., 8, 9.

Παραδείγματος χάρι ο αριθμός 75183 στο δεκαδικό σύστημα αριθμώσεως θα γραφεί ως εξής:

$$75183_{(10)} = 7 \times 10^4 + 5 \times 10^3 + 1 \times 10^2 + 8 \times 10^1 + 3 \times 10^0$$

3.1.2 Το δυαδικό σύστημα αριθμώσεως.

Τό σύστημα αριθμώσεως με τη μικρότερη βάση είναι το σύστημα με βάση τό 2. Τό σύστημα αυτό λέγεται δυαδικό και έχει δύο μόνο σύμβολα: τό 0 και τό 1 με τό οποία σχηματίζονται όλοι οι αριθμοί.

Έτσι ένας άκεραίος αριθμός στην περίπτωση του δυαδικού συστήματος μπορεί να περιγραφεί και πάλι από τη σχέση:

$$N = A_n \times r^n + \dots + A_2 \times r^2 + A_1 \times r^1 + A_0 \times r^0$$

όπου τώρα οι παράγοντες $A_n, \dots, A_2, A_1, A_0$ είναι τά ψηφία 0 ή 1 και η βάση $r = 2$.

Έτσι ένας άκεραίος δυαδικός αριθμός $N = 1001101$ θα μπορούσε να γραφεί ως εξής:

$$\begin{aligned} 1001101_{(2)} &= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 64 + 0 + 0 + 8 + 4 + 0 + 1 = 77_{(10)} \end{aligned}$$

Στόν πίνακα 3.1.1 φαίνεται η αντίστοιχία τών αριθμών 0 – 9₍₁₀₎ με τούς αντίστοιχούς δυαδικούς αριθμούς.

ΠΙΝΑΚΑΣ 3.1.1.
Η αντίστοιχία δεκαδικού και δυαδικού συστήματος.

Δεκαδικό	Δυαδικό
0	0 → 0
1	1 → (1 × 2 ⁰)
2	10 → (1 × 2 ¹ + 0 × 2 ⁰)
3	11 → (1 × 2 ¹ + 1 × 2 ⁰)
4	100 → (1 × 2 ² + 0 × 2 ¹ + 0 × 2 ⁰)
5	101 → (1 × 2 ² + 0 × 2 ¹ + 1 × 2 ⁰)
6	110 → (1 × 2 ² + 1 × 2 ¹ + 0 × 2 ⁰)
7	111 → (1 × 2 ² + 1 × 2 ¹ + 1 × 2 ⁰)
8	1000 → (1 × 2 ³ + 0 × 2 ² + 0 × 2 ¹ + 0 × 2 ⁰)
9	1001 → (1 × 2 ³ + 0 × 2 ² + 0 × 2 ¹ + 1 × 2 ⁰)

Όπως βλέπομε τό δυαδικό σύστημα απαιτεί περισσότερα ψηφία, γιατί απαιτούνται μέχρι και τέσσερα ψηφία γιά να αποδοθεί ένας αριθμός ό οποίος στό δεκαδικό σύστημα συμβολίζεται με ένα μόνο ψηφίο.

Παρόλα αυτά, η χρήση του δυαδικού συστήματος είναι εύρεία στός ηλεκτρονι-

κούς υπολογιστές λόγω της απλότητας των δύο συμβόλων 0 και 1 τα οποία μπορούν να συμβολίζουν ένα ανοικτό (0) ή κλειστό (1) κύκλωμα, μία παρουσία τάσεως (1) ή απουσία τάσεως (0) κ.ά.

“Ας δοῦμε ὅμως τί γίνεται ὅταν ὁ ἀριθμὸς εἶναι κλασματικός (ἢ σύνθετος). Στό δεκαδικό π.χ. σύστημα ὁ ἀριθμὸς 152.25 θά μπορούσε νά παρασταθεῖ ὡς ἐξῆς:

$$152.25_{(10)} = 1 \times 10^2 + 5 \times 10^1 + 2 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}$$

ὅπου οἱ μετὰ τὴν ὑποδιαστολή ἀριθμοὶ (μικρότεροι ἀπὸ τὴ μονάδα) παριστάνονται μὲ ἀντίστοιχες δυνάμεις τῆς βάσεως ἀλλὰ μὲ ἀρνητικό ἐκθέτη.

“Ἐτσι λοιπὸν καὶ στὴν περίπτωση ἑνὸς σύνθετου δυαδικοῦ ἀριθμοῦ, π.χ. τοῦ ἀριθμοῦ 1011101 . 1101, θά ἔχομε:

$$\begin{aligned} 1011101 . 1101 &= 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &\quad + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = \\ &= 64 + 0 + 16 + 8 + 4 + 0 + 1 + 0.5 + 0.25 + 0 + 0.062 \\ &= 93.812_{(10)} \end{aligned}$$

3.1.3 Τὸ ὀκταδικὸ σύστημα ἀριθμήσεως.

“Ἐνα ἄλλο σύστημα ἀριθμήσεως εἶναι τὸ ὀκταδικὸ σύστημα τὸ ὁποῖο περιλαμβάνει 8 σύμβολα. Συνήθως αὐτὰ εἶναι τὰ ψηφία ἀπὸ τὸ 0 ὡς τὸ 7. Θά μπορούσαν ὅμως νά χρησιμοποιηθοῦν καὶ ἄλλα σύμβολα.

“Ἐτσι, ὅπως καὶ στὰ προηγούμενα ἀριθμητικὰ συστήματα, ὁ ἀριθμὸς 367₍₈₎ σημαίνει:

$$\begin{aligned} 367_{(8)} &= 3 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 \\ &= 3 \times 64 + 6 \times 8 + 7 \\ &= 247_{(10)} \end{aligned}$$

Ἐνῶ ὁ κλασματικός ἀριθμὸς 245.528₍₈₎ σημαίνει:

$$\begin{aligned} 245.528_{(8)} &= 2 \times 8^2 + 4 \times 8^1 + 5 \times 8^0 + 5 \times 8^{-1} + 2 \times 8^{-2} + 8 \times 8^{-3} \\ &= 128 + 32 + 5 + 0.625 + 0.0312 + 0.015 \\ &= 165.6712_{(10)} \end{aligned}$$

Πρέπει νά σημειωθεῖ ὅτι τώρα ὁ παράγοντας A_n παίρνει τιμές ἀπὸ 0 - 7 σὲ ἀντίθεση μὲ τὸ δεκαδικὸ σύστημα πού παίρνει τίς τιμές 0 - 9 καὶ τὸ δυαδικὸ πού παίρνει τίς τιμές 0 καὶ 1.

3.1.4 Τὸ δεκαεξαδικὸ σύστημα ἀριθμήσεως.

Τὸ τελευταῖο σύστημα πού θά ἐξετάσουμε εἶναι τὸ δεκαεξαδικὸ σύστημα ἀριθμήσεως. Δέν θά πρέπει νά νομισθεῖ ὅτι δέν ὑπάρχουν καὶ ἄλλα συστήματα, ἀλλὰ τὸ δεκαδικὸ, τὸ δυαδικὸ, τὸ ὀκταδικὸ καὶ τὸ δεκαεξαδικὸ χρησιμοποιοῦνται περισσότερο στοὺς ἠλεκτρονικοὺς υπολογιστές.

Τὸ δεκαεξαδικὸ σύστημα ἀριθμήσεως περιλαμβάνει 16 σύμβολα. Γιὰ τὴν ἀκρίβεια περιλαμβάνει τοὺς ἀριθμοὺς 0, 1, 2... 9 καὶ τὰ ψηφία A, B, C, D, E, F. Σὲ ἀντιστοιχία λοιπὸν μὲ τὸ δεκαδικὸ σύστημα, τὸ A συμβολίζει τὸ 10, τὸ B τὸ 11, τὸ C τὸ 12, τὸ D τὸ 13, τὸ E τὸ 14 καὶ τὸ F τὸ 15. Καὶ πάλι τονίζεται ὅτι ἡ ἐκλογή τῶν συμβόλων εἶναι αὐθαίρετη καὶ θά μπορούσαμε νά χρησιμοποιήσουμε ὁποιαδήποτε ἄλλα. “Ἐτσι λοιπὸν, σύμφωνα καὶ μὲ τὰ προηγούμενα συστήματα, ὁ ἀριθμὸς 82FA₍₁₆₎ σημαίνει:

$$\begin{aligned}
 82FA_{(16)} &= 8 \times 16^3 + 2 \times 16^2 + 15 \times 16^1 + 10 \times 16^0 = \\
 &= 32768 + 512 + 240 + 10 \\
 &= 33530_{(10)}
 \end{aligned}$$

ένω ό αριθμός $5C.F3_{(16)}$ σημαίνει:

$$\begin{aligned}
 5C.F3_{(16)} &= 5 \times 16^1 + 12 \times 16^0 + 15 \times 16^{-1} + 3 \times 16^{-2} \\
 &= 80 + 12 + 0.9375 + 0.0117 \\
 &= 92.9492_{(10)}
 \end{aligned}$$

3.1.5 Μετατροπή ενός αριθμού από ένα σύστημα αριθμήςσεως σέ άλλο.

α) Γιατί είναι αναγκαία ή μετατροπή.

Ό ηλεκτρονικός ύπολογιστής από τήν κατασκευή του μπορεί νά χρησιμοποιήσει ένα καί μόνο σύστημα αριθμήςσεως: τό δυαδικό. Όμως γιατί δέ χρησιμοποιούμε μόνο τό δυαδικό σύστημα αριθμήςσεως;

Ή άπάντηση είναι εύκολη. Για νά παραστήσομε ένα μεγάλο δεκαδικό αριθμό σέ δυαδική μορφή, θά χρειαζόμασταν πάρα πολλά δυαδικά ψηφία. Ή χρήση του όκταδικού συστήματος μειώνει τόν αριθμό αυτό των ψηφίων στο ένα τρίτο καί ή χρήση του δεκαεξαδικού συστήματος στο ένα τέταρτο. Έχοντας ύπόψη μας ότι ή μετατροπή από τό όκταδικό ή τό δεκαεξαδικό στο δυαδικό σύστημα αριθμήςσεως είναι άπλή, χρησιμοποιούμε τό όκταδικό ή δεκαεξαδικό σύστημα αντί για τό δυαδικό για δική μας καί μόνο διευκόλυνση.

β) Τρόπος μετατροπής ενός αριθμού από τό δεκαδικό σύστημα αριθμήςσεως σέ ένα άλλο.

Στό προηγούμενο κεφάλαιο είπαμε ότι άλγόριθμος είναι ό συστηματικός τρόπος επίλύσεως ενός προβλήματος μέ καθορισμένη σειρά, δηλαδή μέ συγκεκριμένους κανόνες.

Ό άλγόριθμος πού ακολουθείται γενικά για τή μετατροπή ενός δεκαδικού αριθμού $N_{(10)}$ σέ έναν αντίστοιχο αριθμό $N_{(r)}$ μέ βάση r (όπου $r = 2, 3, 4, \dots, 8, \dots, 16, \dots$) είναι ό παρακάτω:

$$\begin{aligned}
 (N)_{10}/r &= b + a_0/r \\
 b/r &= c + a_1/r \\
 c/r &= d + a_2/r \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 n/r &= \dots z + a_n/r
 \end{aligned}$$

Τελικά ό ζητούμενος αριθμός $N_{(r)}$ Ισοϋται μέ:

$$N_{(r)} = a_n a_{n-1} \dots a_2 a_1 a_0_{(r)}$$

Δηλαδή ό ζητούμενος αριθμός σχηματίζεται από τά *υπόλοιπα* των διαιρέσεων του άρχικού αριθμού διά της βάσεως του ζητουμένου αριθμού.

γ) Μετατροπή από το δεκαδικό στο δυαδικό αριθμητικό σύστημα.

Σύμφωνα με τον παραπάνω αλγόριθμο, διαιρούμε διαδοχικά το δεκαδικό αριθμό που θέλουμε να μετατρέψουμε με το 2. Το υπόλοιπο κάθε διαιρέσεως είναι οι συντελεστές του αριθμού ο οποίος ζητείται στο νέο αριθμητικό σύστημα. Η διαίρεση συνεχίζεται μέχρις ότου βρούμε πηλίκο μηδέν. Τελικά ο δυαδικός αριθμός σχηματίζεται από τα υπόλοιπα της διαιρέσεως και έχει μικρότερο σημαντικό ψηφίο το πρώτο υπόλοιπο των διαιρέσεων.

Παράδειγμα 1.

Ής υποθέσουμε ότι θέλουμε να μετατρέψουμε το δεκαδικό αριθμό 125 σε δυαδικό. Ακολουθώντας την προηγούμενη διαδικασία, θά έχουμε:

$$\begin{aligned} 125 : 2 &= 62 \text{ και υπόλοιπο } 1 \text{ (μικρότερο σημαντικό ψηφίο)} \\ 62 : 2 &= 31 \text{ και υπόλοιπο } 0 \\ 31 : 2 &= 15 \text{ και υπόλοιπο } 1 \\ 15 : 2 &= 7 \text{ και υπόλοιπο } 1 \\ 7 : 2 &= 3 \text{ και υπόλοιπο } 1 \\ 3 : 2 &= 1 \text{ και υπόλοιπο } 1 \\ 1 : 2 &= 0 \text{ και υπόλοιπο } 1 \text{ (μεγαλύτερο σημαντικό ψηφίο)} \end{aligned}$$

Επομένως ο ζητούμενος αριθμός είναι:

$$125_{(10)} = 1111101_{(2)}$$

Αν επιχειρήσουμε την επαλήθευση, θά έχουμε:

$$\begin{aligned} 1111101_{(2)} &= 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 64 + 32 + 16 + 8 + 4 + 0 + 1 \\ &= 125_{(10)} \end{aligned}$$

Εδώ θά πρέπει να προσέξουμε αν ο αριθμός που θέλουμε να μετατρέψουμε είναι σύνθετος ή μικτός.

Αν συμβαίνει αυτό, τότε χωρίζουμε τον αριθμό σε δύο μέρη, δηλαδή στο ακέραιο και στο κλασματικό.

Για το ακέραιο μέρος ακολουθούμε τη διαδικασία που αναφέραμε παραπάνω. Για το κλασματικό όμως την αντιστρέφουμε.

Συγκεκριμένα **πολλαπλασιάζουμε** αρχικά το κλασματικό μέρος επί 2. Το ακέραιο μέρος του αποτελέσματος (0 ή 1) αποτελεί το μεγαλύτερο σημαντικό ψηφίο του σχηματιζομένου αριθμού. Στη συνέχεια πολλαπλασιάζουμε το κλασματικό μέρος του αποτελέσματος πάλι επί 2. Το ακέραιο μέρος του αποτελέσματος που προκύπτει αποτελεί το επόμενο σημαντικό ψηφίο του δυαδικού αριθμού κ.ο.κ. Ο διπλασιασμός αυτός συνεχίζεται μέχρις ότου το κλασματικό μέρος του αριθμού γίνει μηδέν ή μέχρις ότου αποκτήσουμε την απαιτούμενη ακρίβεια.

Παράδειγμα 2.

Ής υποθέσουμε ότι θέλουμε να μετατρέψουμε το δεκαδικό αριθμό $243.4375_{(10)}$ σε δυαδικό. Καταρχήν χωρίζουμε το δεκαδικό αριθμό σε δύο μέρη: στο ακέραιο και στο δεκαδικό και ακολουθούμε τις διαδικασίες που περιγράψαμε παραπάνω.

(i) Μετατροπή άκεραίου μέρους.

$$\begin{aligned}
243 : 2 &= 121 + \text{υπόλοιπο } 1 \\
121 : 2 &= 60 + \text{υπόλοιπο } 1 \\
60 : 2 &= 30 + \text{υπόλοιπο } 0 \\
30 : 2 &= 15 + \text{υπόλοιπο } 0 \\
15 : 2 &= 7 + \text{υπόλοιπο } 1 \\
7 : 2 &= 3 + \text{υπόλοιπο } 1 \\
3 : 2 &= 1 + \text{υπόλοιπο } 1 \\
1 : 2 &= 0 + \text{υπόλοιπο } 1
\end{aligned}$$

όποτε τό άκέραιο μέρος τοῦ αριθμοῦ μετετράπηκε σέ

$$243_{(10)} = 11110011_{(2)}$$

(ii) Μετατροπή δεκαδικού μέρους.

$0.4375 \times 2 = 0.8750$ κλασματικό καί 0 άκέραιο (μεγαλύτερο σημαντικό ψηφίο)

$0.8750 \times 2 = 0.7500$ κλασματικό καί 1 άκέραιο

$0.7500 \times 2 = 0.5000$ κλασματικό καί 1 άκέραιο

$0.5000 \times 2 = 0.0000$ κλασματικό καί 1 άκέραιο

$0.0000 \times 2 = 0.0000$ κλασματικό καί 0 άκέραιο (Μικρότερο σημαντικό ψηφίο)

Έτσι έχομε:

$$0.4375_{(10)} = 0.01110_{(2)}$$

Συνδυάζοντας τά δύο αποτελέσματα, έχομε:

$$243.4375_{(10)} = 11110011.01110_{(2)}$$

Άν προχωρήσομε στήν επαλήθευση, θά έχομε:

$$\begin{aligned}
11110011.01110_{(2)} &= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + \\
&\quad 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + \\
&\quad 1 \times 2^{-4} + 0 \times 2^{-5} = 243.4375_{(10)}
\end{aligned}$$

δ) Μετατροπή από τό δεκαδικό σύστημα στό όκταδικό.

Καί έδῶ εφαρμόζεται ό ίδιος αλγόριθμος ό όποίος εφαρμόσθηκε καί προηγου-
μένως. Δηλαδή έκτελοῦμε διαδοχικές διαιρέσεις διά τής βάσεως τοῦ νέου συστή-
ματος στό όποίο θά μετατρέψομε τό δεκαδικό αριθμό. Έδῶ ή βάση είναι τό 8.
Έτσι ό δεκαδικός αριθμός $1752_{(10)}$ θά μετατραπεί σέ όκταδικό ὡς έξής:

$$\begin{aligned}
1752 : 8 &= 219 + \text{υπόλοιπο } 0 \\
219 : 8 &= 27 + \text{υπόλοιπο } 3 \\
27 : 8 &= 3 + \text{υπόλοιπο } 3 \\
3 : 8 &= 0 + \text{υπόλοιπο } 3
\end{aligned}$$

$$\text{Άρα: } 1752_{(10)} = 3330_{(8)}$$

Καί προχωρώντας στήν επαλήθευση:

$$\begin{aligned}
3330_{(8)} &= 3 \times 8^3 + 3 \times 8^2 + 3 \times 8^1 + 0 \times 8^0 = 1536 + 192 + 24 + 0 \\
&= 1752_{(10)}
\end{aligned}$$

Θά πρέπει νά παρατηρήσομε ότι τά ψηφία τά όποία δεχόμαστε ὡς ὑπόλοιπα τής
διαιρέσεως είναι ανάλογα μέ τό αριθμητικό σύστημα τό όποίο χρησιμοποιόυμε.
Δηλαδή στήν προκειμένη περίπτωση οι αριθμοί 0, 1, 2...7, είναι παραδεκτά ψηφία.

‘Ας υποθέσουμε τώρα ότι έχουμε έναν κλασματικό αριθμό στο δεκαδικό σύστημα και θέλουμε να τον μετατρέψουμε στο οκταδικό, π.χ. τον αριθμό $673.218_{(10)}$. Ακολουθούμε και πάλι την ίδια διαδικασία όπως και στην προηγούμενη παράγραφο. Χωρίζουμε δηλαδή τον αριθμό στο άκεραιο και στο κλασματικό μέρος και έκτε-
λούμε τις επιμέρους μετατροπές, ενώνοντας στη συνέχεια τα αποτελέσματα. Δη-
λαδή:

(i) Μετατροπή άκεραίου μέρους.

$$\begin{aligned} 673 : 8 &= 84 + \text{υπόλοιπο } 1 \\ 84 : 8 &= 10 + \text{υπόλοιπο } 4 \\ 10 : 8 &= 1 + \text{υπόλοιπο } 2 \\ 1 : 8 &= 0 + \text{υπόλοιπο } 1 \end{aligned}$$

Επομένως

$$673_{(10)} = 1241_{(8)}$$

(ii) Μετατροπή κλασματικού μέρους.

$0.218 \times 8 = 0.744$ κλασματικό + 1 άκεραία μονάδα (μεγαλύτερο σημαντικό ψηφίο)

$0.744 \times 8 = 0.952$ κλασματικό + 5 άκεραιες μονάδες

$0.952 \times 8 = 0.616$ κλασματικό + 7 άκεραιες μονάδες

$0.616 \times 8 = 0.928$ κλασματικό + 4 άκεραιες μονάδες

$0.928 \times 8 = 0.424$ κλασματικό + 7 άκεραιες μονάδες

$0.424 \times 8 = 0.392$ κλασματικό + 3 άκεραιες μονάδες

$0.392 \times 8 = 0.136$ κλασματικό + 3 άκεραιες μονάδες

$0.136 \times 8 = 0.088$ κλασματικό + 1 άκεραία μονάδα

$0.088 \times 8 = 0.704$ κλασματικό + 0 άκεραιες μονάδες (μικρότερο σημαντικό ψηφίο)

Σ’ αυτό τό σημείο θα σταματήσουμε την επανάληψη του αλγορίθμου, γιατί θα πρέπει να έχουμε φτάσει σε αρκετά μεγάλο ποσοστό ακρίβειας, χωρίς αυτό να ση-
μαίνει ότι δεν μπορούμε να συνεχίσουμε τη διαδικασία.

Βρήκαμε δηλαδή ότι:

$$0.218_{(10)} = 0.157473310_{(8)}$$

(iii) Έπαληθευση.

$$\begin{aligned} 0.157473310_{(8)} &= 1 \times 8^{-1} + 5 \times 8^{-2} + 7 \times 8^{-3} + 4 \times 8^{-4} + 7 \times 8^{-5} + 3 \times 8^{-6} + \\ &\quad 3 \times 8^{-7} + 1 \times 8^{-8} + 0 \times 8^{-9} \\ &= 0.125 + 0.078 + 0.013 + 0.0009 + 0.0002 + 0.00001 + \\ &\quad 0.000001 + 0.0000001 \\ &= 0.217201_{(10)} \end{aligned}$$

Εάν συνεχίζαμε την παραπάνω διαδικασία, θα φτάναμε σε κάποιο σημείο όπου η επιθυμητή ακρίβεια θα ήταν πλέον παραδεκτή.

Τελικά βρέθηκε ότι ο αριθμός:

$$673.218_{(10)} = 1241.157473310_{(8)}$$

ε) Μετατροπή από το δεκαδικό σύστημα στο δεκαεξαδικό.

‘Ας πούμε ότι θέλουμε να μετατρέψουμε τον αριθμό $30718.512_{(10)}$ σε δεκαεξαδικό αριθμό.

Ακολουθοῦμε τὴν ἴδια διαδικασία ὅπως καὶ προηγουμένως. Μόνο πού τώρα ἡ βάση τοῦ συστήματος εἶναι τὸ 16.

(i) Μετατροπὴ ἀκεραίου μέρους.

$$30718 : 16 = 1919 + \text{ὑπόλοιπο } 14_{(10)} \rightarrow E_{(16)}$$

$$1919 : 16 = 119 + \text{ὑπόλοιπο } 15_{(10)} \rightarrow F_{(16)}$$

$$119 : 16 = 7 + \text{ὑπόλοιπο } 7$$

$$7 : 16 = 0 + \text{ὑπόλοιπο } 7$$

Ἐπομένως:

$$30718_{(10)} = 77FE_{(16)}$$

(ii) Μετατροπὴ κλασματικοῦ μέρους.

$$0.512 \times 16 = 0.192 \text{ κλασματικό} + 8 \text{ ἀκέραιο}$$

$$0.192 \times 16 = 0.072 \text{ κλασματικό} + 3 \text{ ἀκέραιο}$$

$$0.072 \times 16 = 0.152 \text{ κλασματικό} + 1 \text{ ἀκέραιο}$$

$$0.152 \times 16 = 0.432 \text{ κλασματικό} + 2 \text{ ἀκέραιο}$$

$$0.432 \times 16 = 0.912 \text{ κλασματικό} + 6 \text{ ἀκέραιο}$$

$$0.912 \times 16 = 0.592 \text{ κλασματικό} + 14 \text{ ἀκέραιο (E)}$$

$$0.592 \times 16 = 0.472 \text{ κλασματικό} + 9 \text{ ἀκέραιο}$$

Ὅπως καὶ στὸ προηγούμενο παράδειγμα, θεωροῦμε ὅτι ἡ ἀκρίβεια πού ἐπιτύχαμε εἶναι ἀρκετή.

Ἐπομένως:

$$0.512_{(10)} = .83126E9_{(16)}$$

Καί τελικά:

$$30718.512_{(10)} = 77FE.83126E9_{(16)}$$

(iii) Ἐπαλήθευση.

$$\begin{aligned} 77FE.83126E9_{(16)} &= 7 \times 16^3 + 7 \times 16^2 + 15 \times 16^1 + 14 \times 16^0 + 8 \times 16^{-1} \\ &\quad + 3 \times 16^{-2} + 1 \times 16^{-3} + 2 \times 16^{-4} + 6 \times 16^{-5} + 14 \times 16^{-6} \\ &\quad + 9 \times 16^{-7} \end{aligned}$$

$$= 30718 + 0.5 + 0.011 + 0.0002 + 0.00003 + \dots$$

$$= 30718.51159_{(10)}$$

Ἀποτέλεσμα τὸ ὁποῖο θεωρεῖται ἀρκετὰ ἀκριβές.

στ) Μετατροπὴ ἀπὸ τὸ δυαδικὸ σύστημα ἀριθμήσεως στὸ ὀκταδικὸ καὶ δεκαεξادικὸ καὶ ἀντιστρόφως.

Ἡ μετατροπὴ ἀριθμοῦ ἀπὸ τὸ δυαδικὸ σύστημα στὸ ὀκταδικὸ εἶναι σχετικὰ ἀπλή ἂν σκεφθεῖ κανεὶς ὅτι μεταξὺ τῶν βάσεων τῶν δύο αὐτῶν συστημάτων ὑπάρχει μία ἀπευθείας ἀναλογία. Δηλαδή ἡ βάση τοῦ ὀκταδικοῦ συστήματος (τὸ 8) εἶναι

ο κύβος της βάσεως του δυαδικού συστήματος ($2^3 = 8$).

Έτσι υπάρχει μία άμεση σχέση μεταξύ των 3 διαδοχικών καθ' ομάδα ψηφίων ενός δυαδικού αριθμού με τα αντίστοιχα ψηφία ενός οκταδικού αριθμού.

Μία παρόμοια σχέση υπάρχει και για τη μετατροπή από το δυαδικό σύστημα στο δεκαεξαδικό έφόσον $2^4 = 16$.

Έτσι για να μετατρέψουμε ένα δυαδικό αριθμό σε οκταδικό, χωρίζουμε τον αριθμό *από δεξιά προς τα αριστερά* σε ομάδες των τριών ψηφίων και στη συνέχεια λαμβάνουμε τον αντίστοιχο δεκαδικό αριθμό κάθε ομάδας, σχηματίζοντας έτσι τον τελικό οκταδικό αριθμό με μικρότερο ψηφίο το πρώτο δεκαδικό ψηφίο από δεξιά.

Για τη μετατροπή ενός δυαδικού αριθμού σε δεκαεξαδικό, ακολουθούμε την ίδια διαδικασία, με τη μόνη διαφορά ότι τώρα χωρίζουμε τον αριθμό από δεξιά προς τα αριστερά *ανά τετράδες ψηφίων*.

Τα επόμενα παραδείγματα θα μας βοηθήσουν να καταλάβουμε καλύτερα τις παραπάνω διαδικασίες.

Παράδειγμα 3.

Να μετατραπεί ο δυαδικός αριθμός 01101101101111 σε οκταδικό και δεκαεξαδικό.

(i) Μετατροπή σε οκταδικό αριθμό.

Χωρίζουμε το δυαδικό αριθμό σε ομάδες τριών ψηφίων αρχίζοντας από δεξιά προς τα αριστερά και γράφουμε κάτω από κάθε τριάδα τον αντίστοιχο οκταδικό αριθμό. Δηλαδή:

$$\begin{array}{cccccc} \underline{001} & 101 & 101 & 101 & 111 \\ 1 & 5 & 5 & 5 & 7 \end{array}$$

Επομένως ο ζητούμενος οκταδικός αριθμός είναι:

$$N = 15557_{(8)}$$

(ii) Μετατροπή σε δεκαεξαδικό αριθμό.

Επαναλαμβάνουμε την ίδια διαδικασία, αλλά τώρα χωρίζουμε το δυαδικό αριθμό σε ομάδες των 4 ψηφίων, γράφοντας από κάτω από κάθε ομάδα τον αντίστοιχο δεκαεξαδικό αριθμό. Δηλαδή:

$$\begin{array}{cccc} \underline{0001} & 1011 & 0110 & 1111 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & B & 6 & F \end{array}$$

Επομένως ο ζητούμενος αριθμός είναι:

$$N = 1B6F_{(16)}$$

Θα πρέπει εδώ να παρατηρήσουμε ότι και στις δύο παραπάνω μετατροπές, οι τελευταίες ομάδες ψηφίων, μετά το χωρισμό του δυαδικού αριθμού, δεν είναι πλήρεις (δηλαδή δεν αποτελούνται από 3 ή 4 ψηφία). Τα ψηφία που λείπουν συμπληρώνονται με αντίστοιχα μηδενικά.

Παράδειγμα 4.

Νά μετατραπεί ο δυαδικός αριθμός 1110101111011 σε όκταδικό και δεκαεξαδικό.

Έργαζόμενοι όπως και προηγουμένως, θά έχουμε:

(i) **Μετατροπή σε όκταδικό αριθμό.**

<u>001</u>	110	101	111	011
↓	↓	↓	↓	↓
1	6	5	7	3

Έπομένως:

$$N = 16573_{(8)}$$

(ii) **Μετατροπή σε δεκαεξαδικό αριθμό.**

<u>0001</u>	1101	0111	1011
↓	↓	↓	↓
1	D	7	B

Έπομένως ο αριθμός είναι:

$$N = 1D7B_{(16)}$$

Γιά την αντίστροφη μετατροπή, δηλαδή για την μετατροπή από δεκαεξαδικό ή όκταδικό σύστημα στο δυαδικό, ακολουθούμε την αντίστροφη πορεία. Δηλαδή εκφράζουμε κάθε ψηφίο του όκταδικού ή του δεκαεξαδικού αριθμού σε δυαδικό αριθμό 3 ή 4 ψηφίων αντίστοιχως.

Παράδειγμα 5.

Νά μετατραπεί ο όκταδικός αριθμός 675₍₈₎ στον αντίστοιχο δυαδικό.

Γράφουμε τα ψηφία του όκταδικού αριθμού και κάτω από αυτά τόν αντίστοιχο δυαδικό αριθμό. Δηλαδή:

6	7	5
↓	↓	↓
110	111	101

Έπομένως ο ζητούμενος αριθμός είναι:

$$N = 110111101_{(2)}$$

Παράδειγμα 6.

Νά μετατραπεί ο αριθμός 4FD7E₍₁₆₎ σε δυαδικό αριθμό.

Ακολουθώντας την ίδια διαδικασία, θά έχουμε:

4	F	D	7	E
0100	1111	1101	0111	1110

Έπομένως ο ζητούμενος αριθμός είναι:

$$N = 01001111\ 1101\ 0111\ 1110_{(2)}$$

3.1.6 Αριθμητικές πράξεις στο δυαδικό σύστημα.

α) Πρόσθεση.

Η πρόσθεση στο δυαδικό σύστημα αριθμώσεως γίνεται όπως και η δεκαδική πρόσθεση (δηλαδή από δεξιά προς τα αριστερά) με την εξαίρεση ότι ισχύουν οι εξής βασικοί κανόνες:

$$A + B = F$$

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0$$

καί κρατούμενο 1 για μεταφορά.

Παραδείγματα:

$$\begin{array}{r} 100_{(2)} \\ + 10_{(2)} \\ \hline 110_{(2)} \end{array} \Rightarrow \begin{array}{r} 4_{(10)} \\ + 2_{(10)} \\ \hline 6_{(10)} \end{array}$$

$$\begin{array}{r} 001101_{(2)} \\ + 100101_{(2)} \\ \hline 110010_{(2)} \end{array} \Rightarrow \begin{array}{r} 13_{(10)} \\ + 37_{(10)} \\ \hline 50_{(10)} \end{array}$$

$$\begin{array}{r} 1011011_{(2)} \\ + 1011010_{(2)} \\ \hline 10110101_{(2)} \end{array} \Rightarrow \begin{array}{r} 91_{(10)} \\ + 90_{(10)} \\ \hline 181_{(10)} \end{array}$$

β) Αφαίρεση.

Εκτελείται όπως η δεκαδική αφαίρεση. Ισχύουν οι εξής βασικοί κανόνες:

$$A - B = F$$

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1$$

μέ δανεισμό μίας μονάδας από τό προηγούμενο ψηφίο.

Παραδείγματα.

$$\begin{array}{r} 11010_{(2)} \\ - 01100_{(2)} \\ \hline 01110_{(2)} \end{array} \Rightarrow \begin{array}{r} 26_{(10)} \\ - 12_{(10)} \\ \hline 14_{(10)} \end{array}$$

$$\begin{array}{r} 110.01_{(2)} \\ - 100.10_{(2)} \\ \hline 001.11_{(2)} \end{array} \Rightarrow \begin{array}{r} 6.25_{(10)} \\ - 4.50_{(10)} \\ \hline 1.75_{(10)} \end{array}$$

Έκτός από τον παραπάνω τρόπο υπάρχουν και άλλοι οι οποίοι χρησιμοποιούνται κυρίως στη σχεδίαση των κυκλωμάτων των ηλεκτρονικών υπολογιστών για την ταχύτερη εκτέλεση των πράξεων.

Η πράξη της αφαιρέσεως μπορεί να γίνει με την πρόθεση στο μειωτέο ενός αρνητικού αριθμού. Μέ άλλα λόγια ισχύει η σχέση:

$$20 - 15 = 20 + (-15)$$

Έτσι η αφαίρεση ανάγεται σε πρόθεση δύο αριθμών. Έδω όμως προκύπτει το πρόβλημα: Πώς μπορούμε να παραστήσουμε έναν αρνητικό αριθμό στο δυαδικό σύστημα;

Στους ηλεκτρονικούς υπολογιστές οι αριθμοί παριστάνονται με ένα καθορισμένο για κάθε τύπο υπολογιστή πλήθος ψηφίων.

Ένας αριθμός που εκφράζεται με το πλήθος αυτό των ψηφίων, λέμε ότι αποτελεί μία λέξη του υπολογιστή. Ο αριθμός των ψηφίων μιας λέξεως λέγεται **μήκος της λέξεως**.

Για να παραστήσουμε ένα προσημασμένο αριθμό, χρησιμοποιούμε δύο τρόπους:

- Τήν παράσταση προσημασμένου μεγέθους και
- τήν παράσταση προσημασμένου συμπληρώματος.

Θα εξετάσουμε σε συντομία τους δύο αυτούς τρόπους παραστάσεως των προσημασμένων δυαδικών αριθμών.

(i) Πρώτος τρόπος.

Κατά τη παράσταση προσημασμένου μεγέθους χρησιμοποιούμε ως πρόσημο το πιο σημαντικό ψηφίο της λέξεως του υπολογιστή. Τα υπόλοιπα ψηφία της λέξεως παριστάνουν την απόλυτη τιμή του αριθμού.

Παράδειγμα 1.

Οι αριθμοί $-9_{(10)}$ και $-26_{(10)}$ παριστάνονται σε υπολογιστή μήκους λέξεως 8 δυαδικών ψηφίων ως εξής:

$$\begin{array}{ll} + 9_{(10)} \rightarrow 00001001_{(2)} & + 26_{(10)} \rightarrow 00011010_{(2)} \\ - 9_{(10)} \rightarrow 10001001_{(2)} & - 26_{(10)} \rightarrow 10011010_{(2)} \end{array}$$

Έδω θα πρέπει να σημειώσουμε ότι ο δεκαδικός αριθμός 0 στην παράσταση προσημασμένου μεγέθους παριστάνεται ως εξής:

$$0_{(10)} \rightarrow 00000000_{(2)}$$

και

$$0_{(10)} \rightarrow 10000000_{(2)}$$

Κατά την παράσταση προσημασμένου συμπληρώματος, μία λέξη υπολογιστή ή όποια έχει μήκος n δυαδικών ψηφίων, χρησιμοποιείται για να παραστήσουμε 2^{n-1} θετικούς αριθμούς συμπεριλαμβανομένου και του μηδενός, και 2^{n-1} αρνητικούς αριθμούς. Οι θετικοί αριθμοί εκφράζονται με την απόλυτη τιμή τους και οι αρνητικοί αριθμοί εκφράζονται με τό συμπλήρωμα ως προς 1 ή ως προς 2 της απόλυτης τιμής τους. Θά εξετάσουμε τώρα αυτούς τους δύο τρόπους συμπληρώματος της απόλυτης τιμής ενός δυαδικού αριθμού για να μπορέσουμε να αντιληφθούμε τη χρησιμότητά τους.

1. Τό συμπλήρωμα ως προς 1.

Με τη χρησιμοποίηση της μεθόδου αυτής τό συμπλήρωμα ως προς 1 ενός δυαδικού αριθμού ορίζεται ως εξής:

$$(2^n - N) - 1$$

όπου N είναι ο υπόψη δυαδικός αριθμός σέ δεκαδική μορφή και n τό πλήθος τών ψηφίων του δυαδικού αριθμού.

Παραδείγματος χάρη τό συμπλήρωμα ως προς 1 του αριθμού $N \Rightarrow 8_{(10)} \Rightarrow 1000_{(2)}$, σύμφωνα μέ τήν παραπάνω σχέση, θά ισοϋται προς:

$$(2^n - N) - 1 \Rightarrow (2^4 - 8) - 1 \Rightarrow 7_{(10)} \Rightarrow 0111_{(2)}$$

Ένας πολύ πρακτικός και εύκολος τρόπος να βρούμε τό συμπλήρωμα ως προς 1 ενός δυαδικού αριθμού, είναι να αλλάξουμε τά ψηφία του αντίστοιχου δυαδικού αριθμού από 0 σέ 1 και 1 σέ 0. Έτσι ο αριθμός $+13_{(10)}$ παριστάνεται από τό δυαδικό αριθμό $1101_{(2)}$, ενώ ο αριθμός $-13_{(10)}$ παριστάνεται από τό δυαδικό αριθμό $0010_{(2)}$ ο οποίος βρέθηκε μέ αντιστροφή τών δυαδικών ψηφίων (από 0 $\rightarrow$ 1 και από 1 $\rightarrow$ 0).

2) Τό συμπλήρωμα ως προς 2.

Τό συμπλήρωμα ενός δυαδικού αριθμού ως προς 2 ορίζεται από τή σχέση:

$$2^n - N$$

όπου n είναι ο αριθμός τών ψηφίων του δυαδικού αριθμού και N ή δεκαδική παράστασή του.

Έτσι τό συμπλήρωμα ως προς 2 του αριθμού $12_{(10)} = 1100_{(2)}$

είναι ο αριθμός $2^4 - 12 = 4_{(10)} = 0100_{(2)}$

Καί έδω προς αποφυγή χρήσεως μαθηματικών τύπων υπάρχει ένας απλός κανόνας. Όταν θέλουμε να βρούμε τό συμπλήρωμα ενός δυαδικού αριθμού ως προς 2 τότε βρίσκουμε τό συμπλήρωμα ως προς 1 και προσθέτουμε 1 σ' αυτό.

Παραδείγματος χάρη τό συμπλήρωμα του δυαδικού αριθμού $11000_{(2)}$ ως προς 2, θά είναι:

$$00111 + 00001 = 01000$$

Έτσι χρησιμοποιώντας τήν παράσταση ενός δυαδικού αριθμού μέ τό συμπλή-

ρωμα αυτού ως προς 1 ή ως προς 2, ή πράξη της αφαιρέσεως ανάγεται σε πρόσθεση.

Παράδειγμα 2.

Δίνονται οι αριθμοί $35_{(10)}$ και $-15_{(10)}$ και ζητείται το άθροισμά τους σε όκταψήφιο υπολογιστή.

Στό δεκαδικό σύστημα θά είχαμε: $X = 35 + (-15) = 20$

Στό δυαδικό σύστημα θά είχαμε:

$$\begin{array}{r} 35_{(10)} \rightarrow \quad 00100011 \\ 15_{(10)} \rightarrow \quad - 00001111 \\ \hline \quad \quad 00010100 \Rightarrow 20_{(10)} \end{array}$$

Στό δυαδικό μέ παράσταση προσημασμένου μεγέθους, θά είχαμε:

$$\begin{array}{r} 35_{(10)} \rightarrow \quad 00100011 \\ -15_{(10)} \rightarrow \quad - 10001111 \\ \hline \quad \quad 00010100 \end{array}$$

Τό τελικό κρατούμενο δέ λαμβάνεται υπ' όψη.

Στό δυαδικό μέ παράσταση στό συμπλήρωμα του 1, θά είχαμε:

$$\begin{array}{r} 35_{(10)} \rightarrow \quad 00100011 \\ -15_{(10)} \rightarrow \quad + 11110000 \\ \hline \quad \quad 00010011 \\ \quad \quad + \quad \quad 1 \\ \hline \quad \quad 00010100 \end{array}$$

Τό τελικό κρατούμενο της προσθέσεως προστίθεται στό λιγότερο σημαντικό ψηφίο του άθροίσματος.

Στό δυαδικό μέ παράσταση στό συμπλήρωμα του 2, θά είχαμε:

$$\begin{array}{r} 00100011 \\ + 11110001 \\ \hline + 00010100 \end{array}$$

Τό τελικό κρατούμενο της προσθέσεως δέ λαμβάνεται υπ' όψη.

Άπό τό παραπάνω παράδειγμα φαίνεται ότι για έναν ηλεκτρονικό υπολογιστή ή απλούστερη μέθοδος αφαιρέσεως δύο δυαδικών αριθμών είναι ή μέθοδος χρησιμοποίησης του συμπληρώματος ως προς 2 του αφαιρετέου και προσθέσεώς του στό μειωτέο.

γ) Πολλαπλασιασμός.

Οι κανόνες πολλαπλασιασμού δύο δυαδικών αριθμών είναι οι παρακάτω:

$$0 \times 0 = 0$$

$$1 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 1 = 1$$

Δηλαδή ουσιαστικά ο πολλαπλασιασμός των δυαδικών αριθμών είναι ίδιος με τον πολλαπλασιασμό των δεκαδικών αριθμών.

Παράδειγμα 3.

$$\begin{array}{r} 110101_{(2)} \\ \times 111_{(2)} \\ \hline 110101 \\ 110101 \\ 110101 \\ \hline 101110011_{(2)} \end{array} \Rightarrow \begin{array}{r} 53_{(10)} \\ \times 7_{(10)} \\ \hline 371_{(10)} \end{array}$$

Όταν στον πολλαπλασιασμό ο πολλαπλασιαστής είναι 1, τότε ο πολλαπλασιαστέος προστίθεται στο προηγούμενο άθροισμα αφού τον μεταφέρουμε μία θέση αριστερότερα. Αν ο πολλαπλασιαστής είναι 0, τότε προστίθεται μηδέν.

Στούς ηλεκτρονικούς υπολογιστές συμβαίνει τό εξής:

Όταν εμφανίζεται μηδέν στον πολλαπλασιαστή δεν εκτελείται πρόσθεση αλλά απλώς μεταφορά προς τα αριστερά (όλίσθηση):

$$\begin{array}{r} 01101 \\ \times 01001 \\ \hline 01101 \\ 01101ss \\ \hline 01110101_{(2)} \end{array}$$

όπου $s \rightarrow$ όλίσθηση.

Κάτι παρόμοιο γίνεται και για την πράξη της διαιρέσεως με τη διαφορά ότι η όλίσθηση γίνεται προς τα δεξιά.

δ) Διάρθρωση.

Οι κανόνες της διαιρέσεως δύο δυαδικών αριθμών είναι οι παρακάτω:

$$0 : 1 = 0$$

$$1 : 1 = 1$$

Όπως και στο δεκαδικό σύστημα διαίρεση διά του μηδενός είναι αδύνατη.

Παράδειγμα 4.

$$\begin{array}{r|l}
 110111 & 111 \\
 -0111 & 111 \\
 \hline
 01101 & \\
 -111 & \\
 \hline
 001101 & \\
 -111 & \\
 \hline
 000110 &
 \end{array}
 \Rightarrow
 \begin{array}{r|l}
 55 & 7 \\
 -49 & 7 \\
 \hline
 6 &
 \end{array}$$

3.2 Κώδικες.

Σέ ένα υπολογιστικό σύστημα χρησιμοποιούνται συνήθως διάφοροι μέθοδοι για την παράσταση των πληροφοριών. Κάθε μέθοδος αντιπροσωπεύει ξεχωριστό σύστημα κωδικοποίησης. Για να κωδικοποιήσουμε τα ψηφία του δεκαδικού συστήματος μέσω του δυαδικού απαιτούνται τουλάχιστον 4 δυαδικά ψηφία, γιατί η σχέση $10 \leq 2^n$ ισχύει για $n \geq 4$. Κάθε αντιστοιχία μεταξύ των 10 ψηφίων του δεκαδικού συστήματος και των 10 διαφορετικών n -ψηφίων δυαδικών αριθμών αποτελεί ένα κώδικα δεκαδικό κωδικοποιημένο δυαδικώς. Διεθνώς οι κώδικες αυτοί χαρακτηρίζονται ως κώδικες BCD από τα αρχικά των λέξεων Binary Coded Decimal. Έτσι υπάρχουν τετραψήφιοι, πενταψήφιοι κλπ. κώδικες BCD.

3.2.1 Τετραψήφιοι κώδικες BCD.

Μέ $n = 4$ υπάρχουν $2^4 = 16$ διαφορετικοί δυαδικοί αριθμοί. Από αυτούς μόνο οι 10 μπορούν να αντιστοιχήσουν στα ψηφία του δεκαδικού συστήματος. Οι υπόλοιποι απορρίπτονται. Υπάρχουν 29 δισεκατομμύρια περίπου τρόποι αντιστοιχίσεως των 10 ψηφίων του δεκαδικού συστήματος προς τους 10 από τους 16 τετραψήφιους δυαδικούς αριθμούς. Από όλους αυτούς τους τρόπους - κώδικες, χρησιμοποιούνται μόνο ορισμένοι λόγω ιδιαιζόντων χρησίμων χαρακτηριστικών όπως είναι η ευκολία μετατροπής, ή αναγνώριση σφαλμάτων κλπ. (πίνακας 3.2.1).

Οι κώδικες αυτοί ονομάζονται αντιστοίχως 8 - 4 - 2 - 1, 7 - 4 - 2 - 1, 5 - 4 - 2 - 1 κλπ. Σ' αυτούς, το δεκαδικό ψηφίο προκύπτει από τα αντίστοιχα δυαδικά μέ πολλαπλασιασμό του κάθε δυαδικού ψηφίου με τον αριθμό δ οποίος βρίσκεται πάνω από τη στήλη στην οποία αυτό ανήκει. Π.χ. ο αριθμός 0100 στον κώδικα 8 - 4 - 2 - 1 αντιστοιχεί στον αριθμό $4_{(10)}$ γιατί

$$0 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1 = 4_{(10)}$$

ένω ο αριθμός 1110 στον κώδικα 8 - 4 - 2 - 1 αντιστοιχεί στον αριθμό $8_{(10)}$, γιατί

$$1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 8_{(10)}$$

Ο αριθμός δ οποίος βρίσκεται πάνω από την κάθε στήλη ενός κώδικα, καλείται **βάρος της στήλης** και το σύνολο των βαρών κάθε κώδικα καλείται **βάρος του κώδικα**.

ΠΙΝΑΚΑΣ 3.2.1.
Διάφοροι τετραψήφιοι κώδικες BCD.

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

Είναι σαφής η διαφορά που υπάρχει μεταξύ των BCD κωδίκων και του δυαδικού συστήματος. Η διαφορά θα γίνει περισσότερο σαφής από τα παρακάτω παραδείγματα:

Συγκρίνате π.χ. τούς δυαδικούς αριθμούς με τούς αντίστοιχους σε κώδικα BCD 8 - 4 - 2 - 1 για τούς παρακάτω δεκαδικούς αριθμούς:

Δεκαδικός	Δυαδικός	BCD 8 - 4 - 2 - 1
141	10001101	0001 0100 0001
2179	100010000011	0010 0001 0111 1001

Μέχρι εδώ το πρόβλημα της κωδικοποίησης είναι αρκετά σαφές: κωδικοποιούμε για την εκπλήρωση κάποιας ανάγκης. Όταν όμως έλθουμε στην περίπτωση αριθμητικών πράξεων, τότε τα πράγματα δεν είναι και τόσο απλά.

Ας παρατηρήσουμε τί συμβαίνει όταν προσθέσουμε τούς αριθμούς 9 και 5 στις δύο παραπάνω περιπτώσεις.

Δεκαδικός	Δυαδικός	BCD
9	1001	1001
+ 5	+ 0101	+ 0101
14	1110	1110

Η μορφή όμως του τελικού αθροίσματος στο σύστημα BCD δεν είναι παραδεκτή, γιατί ο κώδικας αυτός παριστάνει δεκαδικούς αριθμούς από το 0 - 9 σε τετραψήφιους αριθμούς, οπότε το αποτέλεσμα για να ήταν σωστό, θα έπρεπε να δοθεί, ως 00010100 και όχι ως 1110.

Σ' αυτή την περίπτωση απαιτούνται ειδικά κυκλώματα για να λειτουργήσουν με τον κώδικα BCD 8-4-2-1.

3.2.2 Κώδικας «υπερβολής - κατά - 3» (*excess - 3 code*).

Ο κώδικας αυτός είναι τροποποιημένη μορφή του κώδικα BCD. Ο «υπερβολής

- κατά - 3» κώδικας είναι κατά τρία ψηφία μεγαλύτερος από τον κώδικα BCD. Π.χ. τό $6_{(10)} \rightarrow 0110_{(2)}$ γράφεται ως 1001, πού είναι τό 9. Ό κώδικας αυτός βοηθά στίς αριθμητικές πράξεις.

ΠΙΝΑΚΑΣ 3.2.2

Δεκαδικός	BCD 8-4-2-1	Ύπερβολής - κατά - 3	Κώδικας του Gray
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0011
3	0011	0110	0010
4	0100	0111	0110
5	0101	1000	0111
6	0110	1001	0101
7	0111	1010	0100
8	1000	1011	1100
9	1001	1100	1101

Στό παρακάτω παράδειγμα φαίνεται πώς ό κώδικας «ύπερβολής - κατά - 3» βοηθά στίς αριθμητικές πράξεις:

$$\begin{array}{r}
 3 \\
 + 9 \\
 \hline
 12
 \end{array}
 \quad
 \begin{array}{r}
 0110 \\
 + 1100 \\
 \hline
 10011
 \end{array}
 \quad
 \begin{array}{r}
 0010 \\
 0011 \\
 \hline
 0101
 \end{array}
 \quad \leftarrow \text{προσθέτομε τόν αριθμό } 3_{(10)}$$

3.2.3 Κώδικας Gray.

Σέ πολλές πρακτικές εφαρμογές, όπως π.χ. μετατροπή από αναλογικά σήματα σέ ψηφιακά (A - D Conversion), απαιτείται ή χρησιμοποίηση δυαδικού κώδικα του οποίου οι παραστάσεις διαδοχικών δεκαδικών ψηφίων πρέπει νά διαφέρουν μεταξύ τους μόνο σέ ένα ψηφίο. Δηλαδή οι δυαδικοί συνδυασμοί νά είναι «γειτονικοί». Οι κώδικες πού έχουν αυτή τήν ιδιότητα αναφέρονται ως «κυκλικοί κώδικες». Σέ όλους τούς κυκλικούς κώδικες, ή δυαδική παράσταση τών ψηφίων 0 καί 9 διαφέρει σέ ένα μόνο ψηφίο. Ό σπουδαιότερος κυκλικός κώδικας είναι ό «κώδικας Gray». Τό χαρακτηριστικό ότι ό κώδικας Gray είναι κυκλικός, άπλοποιεί πολύ τή διαδικασία μετατροπής από τό δυαδικό σύστημα στόν κώδικα Gray.

Κυρίως ό κώδικας Gray χρησιμοποιείται σέ περιπτώσεις πού θέλομε νά ψηφιοποιήσουμε τήν κυκλική κίνηση ενός άξονα.

Στόν πίνακα 3.2.2 φαίνονται σέ αντίστοιχία μεταξύ τους οι κώδικες BCD, 8 - 4 - 2 - 1, ύπερβολής - κατά - 3 καί Gray.

3.2.4 Κώδικας ASCII.

Ό κώδικας ASCII (American Standard Code for Information Interchange) είναι ένας έπταψηφιος κώδικας πού δημιουργήθηκε ως τυποποιημένος κώδικας γιά τήν άπλοποίηση τής επικοινωνίας ύπολογιστή - μέ - ύπολογιστή καί συστήματος - μέ - σύστημα.

Επειδή όρισμένοι ύπολογιστές δέχονται χαρακτηήρες τών 8 ψηφίων, δημιουρ-
γήθηκε ό όκταψηφιος κώδικας ASCII - 8 (πίνακας 3.2.3). Παρακάτω δίνονται πα-
ραδείγματα χρήσεως κώδικα ASCII.

Παράδειγμα 1.

Χρησιμοποιώντας τόν πίνακα 3.2.3 γράψτε τό \$31.27 σέ δυαδική μορφή.
Η λύση είναι:

0100 0100 0101 0011 0101 0001 0100 1110 0101 0010 0101 0111

ή πιό σύντομα μέ δεκαεξαδικό συμβολισμό πού άναφέραμε προηγουμένως:
4453514E5257₍₁₆₎

Παράδειγμα 2.

Ποιά σειρά συμβόλων κωδικοποιήσαμε σύμφωνα μέ τόν πίνακα 3.2.3 στό μή-
νυμα: 59594E54544540B0F5F2E5;

Δεκαδικά ζευγάρια	άποκωδικοποιημένα σύμβολα
59	9
59	9
4E	.
54	4
54	4
45	%
40	κενό
B0	P
F5	u
F2	r
E5	e

Επομένως τό μήνυμα είναι 99.44% Pure.

3.2.5 Ό κώδικας EBCDIC (Extended Binary Coded Decimal Interchange Code).

Στόν κώδικα αυτό χρησιμοποιούμε 8 δυαδικά ψηφία γιά τήν παράσταση τών ά-
ριθμών καί τών διαφόρων γραμμάτων καί σημείων στίξεως. Μέ αυτό τόν κώδικα
μπορούμε νά παραστήσουμε $2^8 - 1 = 255$ σύμβολα. Όμως δέν χρησιμοποιούνται
όλα τά σύμβολα. Ειδικά τά 4 πρώτα χρησιμοποιούνται ως ψηφία τμήματος ζώνης,
όπως καλούνται, μέ άποτέλεσμα ό κώδικας αυτός νά περιλαμβάνει μόνο 52 σύμ-
βολα (πίνακας 3.2.4). Η δυαδική παράσταση κάθε στοιχείου άποδίδεται μέ όκτώ
δυαδικά ψηφία πού άποτελούν μία δυαδική λέξη.

3.2.6 Κώδικες άναγνωρίσεως σφαλμάτων.

Σέ κάθε ύπολογιστικό σύστημα καθώς καί σέ όλα τά συστήματα μεταδόσεως
πληροφοριών συχνά συμβαίνουν λάθη. Τά λάθη αυτά όφείλονται είτε στό θόρυβο
πού μπαίνει στίς γραμμές μεταδόσεως τών πληροφοριών, είτε σέ κάποια βλάβη έ-

ΠΙΝΑΚΑΣ 3.2.4.
Σύμβολα κώδικα EBCDIC.

Στοιχείο	Δυαδική παράσταση	Στοιχείο	Δυαδική παράσταση
0	11110000	T	11100011
1	11110001	U	11100100
2	11110010	V	11100101
3	11110011	W	11100110
4	11110100	X	11100111
5	11110101	Y	11101000
6	11110110	Z	11101001
7	11110111	.	01001011
8	11111000	(	01001101
9	11111001	>	01001100
A	11000001	\$	01011011
B	11000010	*	01011100
C	11000011	)	01011101
D	11000100	;	01011110
E	11000101	-	01100000
F	11000110	/	01100001
G	11000111	,	01101011
H	11001000	%	01101100
I	11001001	ⓐ	01111100
J	11010001	:	01111010
K	11010010	<	01101110
L	11010011	?	01101111
M	11010100	!	01011010
N	11010101		
O	11010110		
P	11010111		
Q	11011000		
R	11011001		
S	11100010		

νός ή και περισσότερων τμημάτων του υπολογιστικού συστήματος. Έπειδή δεν είναι δυνατόν να εξαλειφθεί τελείως ο θόρυβος και επίσης είναι δύσκολο ή και αδύνατο να κατασκευασθεί υπολογιστικό σύστημα που να μη παθαίνει βλάβες, έχουν επινοηθεί διάφοροι κώδικες, ο καθένας από τους οποίους μπορεί να διαπιστώσει ή και να διορθώσει έναν αριθμό σφαλμάτων.

Στους κώδικες BCD με τέσσερα δυαδικά ψηφία, όταν συμβεί ένα λάθος, δηλαδή κάποιο 0 γίνει 1 ή κάποιο 1 γίνει 0, είναι ενδεχόμενο ο συνδυασμός που θα προκύψει να μην περιλαμβάνεται μέσα στις 10 επιτρεπτές καταστάσεις του κώδικα. Π.χ. ο αριθμός 1001 στον κώδικα BCD 8 - 4 - 2 - 1 με ένα λάθος στο δεύτερο ψηφίο του γίνεται 1101 ο οποίος είναι έξω από από τον κώδικα αυτό.

ΠΙΝΑΚΑΣ 3.2.5.
Κώδικας με περισσότερα από 4 ψηφία.

Δεκαδικό ψηφίο	Κώδικας 2 - από - 5	Κώδικας Δυπενταδικός 2 - από - 7
0	01100	01 00001
1	10001	01 00010
2	10010	01 00100
3	00011	01 01000
4	10100	01 10000
5	00101	10 00001
6	00110	10 00010
7	11000	10 00100
8	01001	10 01000
9	01010	10 10000

Ένας τρόπος επομένως ελέγχου είναι να εξετάζεται κατά πόσο ο κάθε BCD αριθμός βρίσκεται μέσα στον κώδικά του. Αυτός ο τρόπος όμως δεν καλύπτει όλες τις περιπτώσεις. Αν π.χ. τό λάθος πού παρατηρήθηκε στον προηγούμενο αριθμό, οφείλεται σε μόνιμη βλάβη του δεύτερου πιά σημαντικού δυαδικού ψηφίου, ο αριθμός 0001 θά γίνει 0101 ο οποίος βρίσκεται μέσα στον κώδικα.

Έτσι, γιά τόν έλεγχο σφαλμάτων έχουν δημιουργηθεί ειδικοί κώδικες οι οποίοι βασίζονται στην προσθήκη περισσοτέρων ψηφίων στον κώδικα. Ένας τέτοιος κώδικας είναι ο κώδικας «2 από 5» (πίνακας 3.2.5). Στόν κώδικα αυτό κάθε ψηφίο του δεκαδικού συστήματος παριστάνεται μέ 5 δυαδικά ψηφία μέ τέτοιο τρόπο, ώστε σε κάθε συνδυασμό του κώδικα νά υπάρχουν μόνο 2 μονάδες. Έτσι όταν συμβεί ένα σφάλμα, ο αριθμός τών μονάδων θά γίνει 1 ή 3.

Άπλός Έλεγχος Ισοτιμίας (Parity Check).

Άλλος τρόπος ελέγχου τής εμφάνισης σφάλματος είναι μέ πρόσθεση ενός ακόμα ψηφίου ελέγχου έτσι, ώστε τό πλήθος τών 1 του συνδυασμού νά είναι είτε άρτιος αριθμός είτε περιττός. Τό επιπλέον αυτό ψηφίο καλείται ψηφίο Ισοτιμίας (parity bit). Στόν πίνακα 3.2.6 δίνεται τό ψηφίο Ισοτιμίας δύο διαφορετικών κωδίκων μέ άρτια καί περιττή Ισοτιμία (ψηφίο p).

Έτσι αν ένα μήνυμα περιέχει n ψηφία, τό κωδικοποιημένο μήνυμα θά περιέχει $n + 1$ ψηφία.

Κατά τή λήψη τώρα, μετρούμε τά ψηφία 1 στό μήνυμα όπου ένας μονός αριθμός σε κώδικα μέ άρτια Ισοτιμία θά μās παρέχει ένδειξη ότι ύφίσταται ένα **τουλάχιστο** σφάλμα στό μήνυμα. Αντίστοιχα, ένας ζυγός αριθμός θά μās παρέχει ένδειξη ότι ύφίσταται ένα τουλάχιστο σφάλμα σε κώδικα μέ περιττή Ισοτιμία.

Προφανώς μέ αυτόν τόν τρόπο κωδικοποίησης ένα διπλό σφάλμα δεν μπορεί νά άνιχνευθεί όπως επίσης καί ένας ζυγός αριθμός σφαλμάτων.

Έκτός από τούς κώδικες πού αναφέραμε υπάρχουν καί πολλοί άλλοι, όπως οι

ΠΙΝΑΚΑΣ 3.2.6.
Κώδικας με Άρτια και περιττή Ισοτιμία.

Δεκαδικό ψηφίο	Άρτια Ισοτιμία	Περιττή Ισοτιμία
	Κώδικας (BCD) p 8 4 2 1	Κώδικας (7-4-2-1) p 7 4 2 1
0	0 0 0 0 0	1 0 0 0 0
1	1 0 0 0 1	0 0 0 0 1
2	1 0 0 1 0	0 0 0 1 0
3	0 0 0 1 1	1 0 0 1 1
4	1 0 1 0 0	0 0 1 0 0
5	0 0 1 0 1	1 0 1 0 1
6	0 0 1 1 0	1 0 1 1 0
7	1 0 1 1 1	0 1 0 0 0
8	1 1 0 0 0	1 1 0 0 1
9	0 1 0 0 1	1 1 0 1 0

A - N (Άλφαριθμητικοί κώδικες), ο κώδικας Hamming και οι κώδικες μεταβλητού μήκους. Οι κώδικες αυτοί χρησιμοποιούνται ευρέως κυρίως έπειδή παρέχουν τη δυνατότητα όχι μόνο αναγνώρισης σφαλμάτων αλλά και διορθώσεώς τους.

3.3 Έρωτήσεις.

1. Τι καλείται βάση ενός συστήματος αρίθμησης;
2. Γιατί το δυαδικό σύστημα αρίθμησης είναι το πιο ευχρηστο σύστημα για τους Η/Υ;
3. Τι καλείται συμπλήρωμα ενός αριθμού ως προς 1; Τι ως προς 2;
4. Σέ ποιά κριτήρια βασίζεται η έκλογή του καταλλήλου ψηφιακού κώδικα έλέγχου σφαλμάτων;
5. Τι είναι οι κώδικες μέ βαρή; Ποϋ χρησιμοποιούνται;
6. Ποϋ χρησιμοποιείται κυρίως ο κώδικας του Gray;
7. Ποϋ χρησιμοποιείται ο κώδικας ASCII;
8. Ποιά πλεονεκτήματα προσφέρουν οι κώδικες αναγνώρισης και διορθώσεως σφαλμάτων;

3.4 Ασκήσεις.

1. Ποιά δεκαδικό αριθμό παριστάνει ο δυαδικός αριθμός 101001001;
2. Ποιά δεκαδικό αριθμό παριστάνουν οι αριθμοί: (α) 100010100111₍₂₎ (β) 45742₍₈₎ (γ) 124A52C₍₁₆₎.
3. Νά μετατραποϋν οι αριθμοί 752.612₍₁₀₎ και 1453.46₍₁₀₎ σέ δυαδικούς.
4. Νά μετατραπεί ο αριθμός 646.46₍₁₀₎ σέ όκταδικό και σέ δεκαεξαδικό αριθμό. Νά έπαληθευθοϋν τά άποτελέσματα.
5. Νά μετατραπεί ο αριθμός 657.64₍₈₎ σέ δυαδικό αριθμό. Νά έπαληθεϋσετε τό άποτέλεσμα.
6. Νά μετατραπεί ο αριθμός A7254C₍₁₆₎ σέ όκταδικό αριθμό. Νά έπαληθεϋσετε τό άποτέλεσμα.
7. Νά μετατραπεί ο αριθμός 011101111010₍₂₎ σέ όκταδικό και δεκαεξαδικό αριθμό. Νά έπαληθεϋσετε τό άποτέλεσμα.
8. Προσθέσετε τούς δυαδικούς αριθμούς 010100 και 1010101. Νά έπαληθευθεί τό άποτέλεσμα μέ τήν μετατροπή τών αριθμών αϋτών σέ δεκαδικούς αριθμούς.
9. Νά γίνει άφαίρεση τών δυαδικών αριθμών 10101 και 10111. Νά έπαληθευθεί τό άποτέλεσμα μέ τήν μετατροπή τών αριθμών σέ δεκαδική μορφή. (Τό πρώτο ψηφίο παριστάνει τό πρόσημο τού αριθμού 1).
10. Νά γίνει ή άφαίρεση τών δυαδικών αριθμών 1100101 και 1011011 (α) Μέ τή μέθοδο τού

συμπληρώματος ως προς 1 και (β) Μέ τη μέθοδο του συμπληρώματος ως προς 2.

11. Πολλαπλασιάσετε τούς δυαδικούς αριθμούς 1101 και 1011 και επαληθεύσατε τό αποτέλεσμα.
 12. Νά γίνει ή διαίρεση τῶν δυαδικῶν ἀριθμῶν $100110:1101$
 13. Νά παρασταθεῖ ὁ ἀριθμός $649.812_{(10)}$ μέ κώδικα BCD 8 - 4 - 2 - 1.
 14. Νά παρασταθεῖ ὁ ἀριθμός $72.14_{(10)}$ μέ τούς κώδικες BCD (α) 7 - 4 - 2 - 1 (β) 5 - 2 - 1 - 1 καί 5 - 4 - 2 - 1.
 15. Τί σημαίνει τό κωδικοποιημένο μήνυμα A8E5ECECEF4041 σέ κώδικα ASCII;
-

ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

ΛΟΓΙΚΑ ΚΥΚΛΩΜΑΤΑ

4.1 'Η άλγεβρα του Boole.

4.1.1 Γενικά.

Τό έτος 1938 ό Claude Shannon προσπαθώντας νά βρεΐ Ένα συστηματικό τρόπο γιά τήν μελέτη τής συμπεριφοράς τών κυκλωμάτων μέ διακόπτες, πού άποτελοΐν τή βάση τών ηλεκτρονικών ύπολογιστών, διαπίστωσε ότι ή Άλγεβρα τοΐ Boole, πού άναπτύχθηκε στό βιβλίο του «An Investigation of the Laws of Thought» τό έτος 1854, μπορούσε χωρίς καμμιά άλλαγή νά χρησιμοποιηθεΐ. Βέβαια ό G. Boole άνέπτυξε τήν άλγεβρά του γιά νά μελετήσει τίς λογικές σχέσεις άνάμεσα σέ προτάσεις. Άνέπτυξε δηλαδή τήν άλγεβρα τής λογικής.

Ό Shannon μεταφέροντας τίς ίδιες έννοιες στοΐς διακόπτες, τίς διασυνδέσεις καΐ τίς πράξεις πού μπορούν νά εκτελοΐν, διεμόρφωσε τήν άλγεβρα τοΐ Boole στή μορφή πού εΐναι γνωστή σάν Άλγεβρα τών λογικών κυκλωμάτων.

4.1.2 Άξιώματα τής άλγεβρας τοΐ Boole.

Γιά τήν άνάπτυξη τής άλγεβρας τοΐ Boole, χρησιμοποιούμε γιά σύμβολα τίς πράξεις «+», «-», ή «'», «.» καΐ τό σύμβολο «=», καθώς επίσης καΐ τήν κλάση M πού τή θεωρούμε ότι περιέχει τά στοιχεΐα A,B,Γ,Δ,...

Τά άξιώματα τής άλγεβρας τοΐ Boole εΐναι τά εξής:

Άξίωμα Πρώτο.

- 1(α) Άν A καΐ B εΐναι δύο στοιχεΐα πού άνήκουν στήν κλάση M, τότε καΐ τό στοιχεΐο (A+B) άνήκει στή κλάση M.
- 1(β) Άν A καΐ B άνήκουν στήν κλάση M, τότε καΐ τό στοιχεΐο (A.B) άνήκει στήν κλάση M.

Άξίωμα Δεύτερο.

- 2(α) Υπάρχει Ένα στοιχεΐο 0 πού άνήκει στήν κλάση M. Τό στοιχεΐο αυτό όνομάζεται μηδενικό στοιχεΐο καΐ Ισχύει ή σχέση:

$$A + 0 = A$$

(γιά κάθε στοιχεΐο τής κλάσεως M)

- 2(β) Υπάρχει Ένα στοιχεΐο 1 πού άνήκει στήν κλάση M. Τό στοιχεΐο αυτό όνομάζεται μοναδιαΐο στοιχεΐο καΐ Ισχύει ή σχέση:

$$A.1 = A$$

(γιά κάθε στοιχεΐο τής κλάσεως M)

Άξιώμα Τρίτο.

3(α) Για κάθε ζευγάρι στοιχείων A και B που ανήκουν στην κλάση M έχουμε:

$$A+B=B+A$$

3(β) Για κάθε ζευγάρι στοιχείων A και B που ανήκουν στην κλάση M έχουμε:

$$A \cdot B = B \cdot A$$

Άξιώμα Τέταρτο.

4(α) Για κάθε τριάδα στοιχείων A, B, Γ που ανήκουν στην κλάση M έχουμε:

$$A + (B \cdot \Gamma) = (A + B) \cdot (A + \Gamma)$$

4(β) Για κάθε τριάδα στοιχείων A, B, Γ που ανήκουν στην κλάση M έχουμε:

$$A \cdot (B + \Gamma) = A \cdot B + A \cdot \Gamma$$

Άξιώμα Πέμπτο.

Για κάθε στοιχείο A της κλάσεως M υπάρχει πάντα ένα στοιχείο $\bar{A}$ (ή A') της κλάσεως M τέτοιο, ώστε:

$$5(\alpha) \quad A + \bar{A} = A + A' = 1 \text{ και}$$

$$5(\beta) \quad A \cdot \bar{A} = A \cdot A' = 0$$

Άξιώμα Έκτο.

Στην κλάση M υπάρχουν τουλάχιστον δύο στοιχεία A και B , ώστε:

$$A \neq B$$

Παρατηρούμε ότι μεταξύ των αξιωμάτων που τα χαρακτηρίζουμε με δείκτη (α) και (β) υπάρχει ένας «δυϊσμός», αν εντικαταστήσουμε την πράξη (+) με την δυϊκή της (·) και το στοιχείο 1 (ή 0) με το δυϊκό του 0 (ή 1).

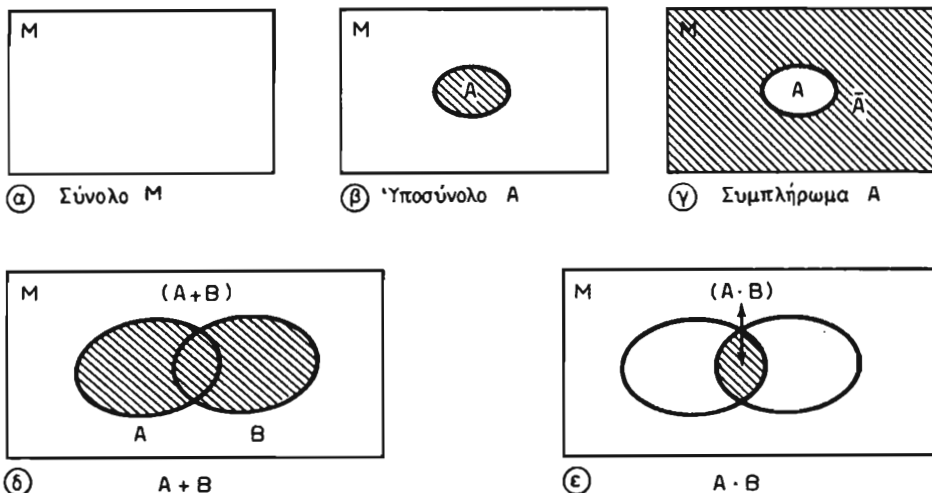
4.1.3 Διαγράμματα Venn.

Στά διαγράμματα Venn η κλάση M παριστάνει ένα βασικό σύνολο και τα στοιχεία $A, B, \Gamma, \dots$ παριστάνουν υποσύνολα του βασικού συνόλου M . Το σύμβολο «+» παριστάνει την «ένωση» συνόλων, το σύμβολο «·» την τομή, το σύμβολο «-» το συμπλήρωμα και το σύμβολο «=» την Ισότητα συνόλων. Επίσης το μοναδιαίο στοιχείο παριστάνει το βασικό σύνολο M , ενώ το μηδενικό στοιχείο παριστάνει το κενό σύνολο.

Τό σύνολο M μπορεί να παρασταθεί ως τό σύνολο των σημείων του επιπέδου τα όποια βρίσκονται μέσα σ' ένα τετράγωνο [σχ. 4.1(α)]. Ένα υποσύνολο A του M μπορεί να παρασταθεί ως σύνολο των σημείων του M τα όποια βρίσκονται μέσα σέ μία κλειστή γραμμή [σχ. 4.1(β)]. Στο σχήμα 4.1(γ) τό γραμμοσκιασμένο τμήμα του τετραγώνου παριστάνει τό σύνολο $\bar{A}$. Στο σχήμα 4.1(δ) τό γραμμοσκιασμένο τμήμα παριστάνει τό σύνολο $(A + B)$, δηλαδή τή συνένωση των υποσυνόλων A και B . Τέλος στο σχήμα 4.1(ε) τό γραμμοσκιασμένο τμήμα παριστάνει τό σύνολο $(A \cdot B)$, δηλαδή τήν τομή των υποσυνόλων A και B .

Αν τά υποσύνολα A και B δέν έχουν κοινό στοιχείο μεταξύ τους, τότε λέγονται ξένα μεταξύ τους και ή τομή τους $A \cdot B$ είναι τό μηδενικό σύνολο, $A \cdot B = 0$.

Μέ τά διαγράμματα Venn μπορούμε νά παραστήσουμε οποιαδήποτε έκφραση τής άλγεβρας των συνόλων και συνεπώς και οποιαδήποτε έκφραση τής άλγεβρας Boole.



Σχ. 4.1.

Διαγράμματα Venn για την παράσταση του συμπληρώματος, της τομής και της ένωσης υποσυνόλων.

4.1.4 Άλγεβρα Λογικής – Πίνακες Άληθείας.

Στήν άλγεβρα της λογικής ή κλάση M παριστάνει το σύνολο όλων των δυνατών προτάσεων και τα στοιχεία $A, B, \Gamma, \dots$ παριστάνουν προτάσεις. Κάθε πρόταση θεωρείται ότι μπορεί να είναι «άληθής» ή «ψευδής». Αν μία πρόταση A είναι άληθής, τότε λέμε ότι έχει την τιμή 1 και γράφουμε $A = 1$. Αν είναι ψευδής, λέμε ότι έχει την τιμή 0 και γράφουμε $A = 0$. Όταν έχουμε δύο προτάσεις A και B , τότε μπορεί να είναι και οι δύο ψευδείς, ή η A να είναι ψευδής και η B άληθής ή η A άληθής και η B ψευδής ή να είναι και οι δύο άληθεις. Οι τέσσερις αυτές περιπτώσεις φαίνονται στις δύο πρώτες στήλες του πίνακα 4.1.1. Φυσικά για την περίπτωση 3 προτάσεων A, B και Γ οι απαιτούμενοι συνδυασμοί αυξάνονται στους 8 και γενικά για n προτάσεις έχουμε 2^n δυνατές περιπτώσεις συνδυασμού των τιμών τους.

Στήν άλγεβρα της λογικής το σύμβολο «+» παριστάνει το λογικό «Ή», το σύμβολο «.» παριστάνει το λογικό «Και», το σύμβολο « $\neg$ » το λογικό «Όχι» και το σύμβολο « $\Rightarrow$ » παριστάνει την ισότητα προτάσεων. Το λογικό «Ή» ονομάζεται και λογική πρόσθεση, το λογικό «Και» ονομάζεται και λογικός πολλαπλασιασμός και το λογικό «Όχι» ονομάζεται και λογική άρνηση ή αντίστροφη.

Η πρόταση $(A + B)$ είναι ψευδής μόνο όταν και οι δύο προτάσεις A και B είναι ψευδείς.

Η πρόταση $(A \cdot B)$ είναι άληθής μόνο όταν και οι δύο προτάσεις A και B είναι άληθεις.

Η πρόταση A είναι άληθής, μόνο όταν η πρόταση $\bar{A}$ είναι ψευδής και αντίστροφα.

Η πρόταση $\Gamma = A$ είναι άληθής, όταν η A είναι άληθής και ψευδής όταν η A είναι ψευδής.

Στόν πίνακα 4.1.1 δίνεται, για κάθε συνδυασμό τιμών των προτάσεων A και B , η αντίστοιχη τιμή των προτάσεων $A + B$, $A \cdot B$, $\bar{A}$ και $\Gamma = A$. Κάθε πίνακας στόν δ-

ποιο περιλαμβάνονται όλοι οι δυνατοί συνδυασμοί τιμών για έναν αριθμό προτάσεων και δίνεται, για κάθε συνδυασμό, η τιμή μιās προτάσεως ή όποια περιλαμβάνει μία ή περισσότερες απ' αυτές, λέγεται «**πίνακας αληθείας**».

ΠΙΝΑΚΑΣ 4.1.1.

Πίνακας αληθείας για τό λογικό «Η», ΚΑΙ, ΟΧΙ, =.

A	B	A+B	A.B	$\bar{A}$ ή A'	$\Gamma=A$
0	0	0	0	1	0
0	1	1	0	1	0
1	0	1	0	0	1
1	1	1	1	0	1

4.1.5 Θεωρήματα της Άλγεβρας Boole.

Στά παρακάτω αναφέρονται όρισμένα βασικά θεωρήματα της Άλγεβρας Boole. Μέ τή βοήθεια αυτών μπορούμε νά άπλοποιούμε λογικές συναρτήσεις.

Πρέπει όμως νά έχομε υπ' όψη μας ότι όσο μεγαλώνει ό αριθμός των μεταβλητών τόσο δυσχερέστερη γίνεται ή άπλοποίηση.

Στήν παράγραφο αυτή θά προσπαθήσομε νά άπλοποιήσομε μερικές λογικές συναρτήσεις, ώστε ή χρήση των θεωρημάτων νά γίνει πιό κατανοητή.

Θά πρέπει νά τονισθεϊ ότι από έδω καί πέρα τό σύμβολο της λογικής πράξεως ΚΑΙ (.) θά παραλείπεται για χάρη εύκολίας. Άντί $X = A \cdot B$ θά έχομε $X = AB$.

Σημειώνεται τέλος ότι υπάρχουν διάφοροι τρόποι άπλοποιήσεως των λογικών συναρτήσεων, πού βασίζονται βέβαια στά θεωρήματα, καί οι όποιοι χρησιμοποιούν πίνακες (όπως οι πίνακες Karnaugh ή ή μέθοδος Quine Mc Clusky) μέ τή βοήθεια των όποίων γίνεται στήν πράξη ή άπλοποίηση.

α) Τά Θεωρήματα.

1. Θεώρημα μοναδικότητας.

- α) Τό στοιχείο 1 είναι μοναδικό.
- β) Τό στοιχείο 0 είναι μοναδικό.
- γ) Τό $\bar{A}$ του A είναι μοναδικό.

2. Θεώρημα διπλής άρνήσεως.

$$(\bar{A}) = \bar{\bar{A}} = A.$$

3. Θεώρημα του De Morgan.

- α) $\overline{(A + B)} = \bar{A} \cdot \bar{B}$
- β) $\overline{(A \cdot B)} = \bar{A} + \bar{B}$

4. Θεώρημα συμπληρώματος.

- α) $A + \bar{A} = 1$
- β) $A \cdot \bar{A} = 0$

5. Θεώρημα αυτοενώσεως καί αυτοτομής.

- α) $A + A = A$
- β) $A \cdot A = A$

6. Θεώρημα κυριαρχικότητας.

α) $A + 1 = 1$

β) $A \cdot 0 = 0$

7. Θεώρημα απορροφητικότητας.

α) $A \cdot (A + B) = A$

β) $A + (A \cdot B) = A$

8. Θεώρημα αντιμεταθέσεως.

α) $A + B = B + A$

β) $A \cdot B = B \cdot A$

9. Θεώρημα προσεταιρισμού.

α) $A + (B + \Gamma) = (A + B) + \Gamma$

β) $A \cdot (B \cdot \Gamma) = (A \cdot B) \cdot \Gamma$

10. Θεώρημα έπιμερισμού.

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

β) $A + \bar{A} \cdot B = A + B$

11. Θεωρήματα μηδενός και μονάδας.

α) $A + 0 = A$

β) $A \cdot 1 = A$

καί

α) $A + 1 = 1$

β) $A \cdot 0 = 0$

Στά παραπάνω θεωρήματα παρατηρείται ή έννοια τοῦ δυΐσμοῦ καί ή άντιστοιχία τῶν νόμων τῶν συνόλων π.χ.

$$A + A = A, \quad A \cdot A = A$$

Αυτό οδηγεί στό συμπέρασμα ότι ή άλγεβρα Boole είναι ούσιαστικά άλγεβρα συνόλων.

Ή απόδειξη τῶν παραπάνω θεωρημάτων μπορεί νά γίνει μέ αξιώματα τῆς άλγεβρας τοῦ Boole ή μέ τά διαγράμματα Venn ή μέ τούς πίνακες ἀληθείας.

β) Ἀπλοποιήσεις λογικῶν συναρτήσεων.

Νά ἀπλοποιηθοῦν οἱ παρακάτω λογικές συναρτήσεις.

$$F_1 = A + ABC' + ABC + BC + B$$

$$F_2 = AB + C' (A + B)$$

$$F_3 = ABC' + AB'C + A'BC + B (A' + B + C)$$

$$F_4 = ABC' + BC' + A'C$$

$$F_5 = (B' + C') (A + B + C)$$

$$F_6 = ABCD + A'B' + AC'D' + BC'D'$$

Λύση.

Γιά τή συνάρτηση F_1 , θά έχομε:

$$\begin{aligned}
 F_1 &= A + ABC' + ABC + BC + B \\
 &= A (1 + BC' + BC) + B (C + 1) \\
 &= A [1 + B (C + C')] + B (C + 1) \\
 &= A [1 + B (1)] + B (1) \\
 &= A [1 + B] + B (1) \\
 &= A (1) + B (1) \\
 &= A + B
 \end{aligned}$$

Γιά τη συνάρτηση F_2 θά έχουμε:

$$\begin{aligned}
 F_2 &= AB + C' (A + B) \\
 &= AB + C'A + C'B
 \end{aligned}$$

Η συνάρτηση αυτή δεν απλοποιείται περισσότερο. Έτσι μπορούμε να χρησιμοποιήσουμε οποιαδήποτε μορφή θέλομε.

Γιά τη συνάρτηση F_3 θά έχουμε:

$$\begin{aligned}
 F_3 &= ABC' + AB'C + A'BC + B (A' + B + C) \\
 &= ABC' + AB'C + A'BC + BA' + BB + BC \\
 &= ABC' + AB'C + A'BC + BA' + B + BC \\
 &= A (BC' + B'C) + A'B (C + 1) + B (C + 1) \\
 &= A (BC' + B'C) + A'B (1) + B (1) \\
 &= A (BC' + B'C) + A'B + B \\
 &= A (BC' + B'C) + B (A' + 1) \\
 &= A (BC' + B'C) + B (1) \\
 &= A (BC' + B'C) + B \\
 &= ABC' + AB'C + B \\
 &= B (AC' + 1) + AB'C \\
 &= B (1) + AB'C \\
 &= B + AB'C
 \end{aligned}$$

Γιά τη συνάρτηση F_4 θά έχουμε:

$$\begin{aligned}
 F_4 &= ABC' + BC' + A'C \\
 &= BC' (A + 1) + A'C \\
 &= BC' (1) + A'C \\
 &= BC' + A'C
 \end{aligned}$$

Γιά τη συνάρτηση F_5 θά έχουμε:

$$\begin{aligned}
 F_5 &= (B' + C') (A + B + C) \\
 &= B'A + B'B + B'C + C'A + C'B + C'C \\
 &= B'A + 0 + B'C + C'A + C'B + 0 \\
 &= B'A + B'C + C'A + C'B \\
 &= A (B' + C') + B'C + BC'
 \end{aligned}$$

Γιά τη συνάρτηση F_6 θά έχουμε:

$$\begin{aligned}
 F_6 &= ABCD + A'B' + AC'D' + BC'D' \\
 &= ABCD + A'B' + C'D' (A + B)
 \end{aligned}$$

4.2 Πραγματοποίηση Λογικών Κυκλωμάτων.

Ός εδώ μάς είναι γνωστές οι έννοιες των λογικών μεταβλητών και των λογικών συναρτήσεων. Στην πράξη οι λογικές συναρτήσεις πραγματοποιούνται με ψηφιακά (λογικά) κυκλώματα, δηλαδή ηλεκτρονικά κυκλώματα με τα οποία μπορούμε να κά-νομε λογικές πράξεις δίνοντας στις εισόδους τους ηλεκτρικά σήματα που παριστά-νουν τις λογικές μεταβλητές και παίρνοντας στην έξοδο τους ηλεκτρικά σήματα που παριστάνουν την τιμή της λογικής συναρτήσεως.

Στην παράγραφο αυτή θα προσπαθήσουμε να δώσουμε μία πολύ σύντομη περι-γραφή μερικών χαρακτηριστικών ηλεκτρονικών λογικών κυκλωμάτων δίνοντας ταυτόχρονα το αντίστοιχο ηλεκτρονικό κύκλωμα που πραγματοποιεί την απαιτού-μενη λογική πράξη. Δέ θα αναπτύξουμε τόν τρόπο λειτουργίας του (από ήλε-κτρονικής πλευράς) γιατί αυτό ξεφεύγει από τόν σκοπό τού βιβλίου. Άλλωστε ή περιγραφή τής κατασκευής ενός λογικού κυκλώματος με διακεκριμένα ηλεκτρονι-κά στοιχεία (αντιστάσεις, πυκνωτές, τρανζίστορ, διόδους κλπ.) δέν έχει πιά καμιά πρακτική έννοια άφου σήμερα χρησιμοποιούνται αποκλειστικά τά ολοκληρωμένα κυκλώματα με άριστες επιδόσεις.

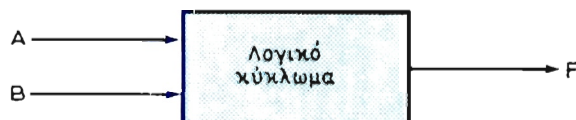
Τά λογικά κυκλώματα μπορούμε νά τά κατατάξουμε σέ δύο γενικές κατηγορίες:

- Στά συνδυαστικά κυκλώματα καί
- στά άκολουθιακά κυκλώματα.

Στά συνδυαστικά κυκλώματα ή έξοδος σέ κάθε χρονική στιγμή είναι συνάρτηση των τιμών που έχουν οι εισοδοι τού κυκλώματος, τήν ίδια χρονική στιγμή καί μόνο. Στά άκολουθιακά κυκλώματα ή έξοδος είναι συνάρτηση καί των προηγούμενων εισόδων τού κυκλώματος. Δηλαδή είναι κυκλώματα τά όποια θυμούνται τήν προη-γούμενη ή τις προηγούμενες καταστάσεις τους.

4.2.1 Θετική καί άρνητική λογική — Λογικές πύλες.

Στό σχήμα 4.2α φαίνεται διαγραμματικώς ένα λογικό κύκλωμα με εισόδους Α καί Β καί έξοδο F. Υποθέτουμε ότι οι εισοδοι καί ή έξοδος μπορούν νά πάρουν δύο μόνο τιμές δυναμικού, μία *ύψηλή* π.χ. +5V, τήν όποία συμβολίζουμε με τό γράμμα H (High), καί μία *χαμηλή* π.χ. 0V, τήν όποία συμβολίζουμε με τό γράμμα L (Low). Τό λογικό αυτό κύκλωμα είναι έτσι κατασκευασμένο, ώστε νά συμπεριφέρεται σύμ-φωνα με τόν πίνακα 4.2.1.



Σχ. 4.2α.

Σχηματική παράσταση ενός λογικού κυκλώματος.

Ό πίνακας 4.2.2 είναι ίδιος με τόν πίνακα 4.2.1, με τή διαφορά ότι αντικαταστή-σαμε τό ύψηλό δυναμικό των +5V με τό γράμμα H καί τό χαμηλό δυναμικό των 0V με τό γράμμα L.

Άν τώρα αντικαταστήσουμε τά σύμβολα H καί L με τά δυαδικά σύμβολα 1 καί 0 άντιστοίχως, τότε έχουμε ένα πίνακα «άληθείας» τής λογικής διατάξεις τού σχήμα-τος 4.2α. Γενικά ή χρήση των πινάκων άληθείας μάς βοηθά στην κατανόηση τής

ΠΙΝΑΚΑΣ 4.2.1.

Είσοδοι		Έξοδος
A	B	F
0V	0V	0V
0V	+5V	0V
+5V	0V	0V
+5V	+5V	+5V

ΠΙΝΑΚΑΣ 4.2.2.

Είσοδοι		Έξοδος
A	B	F
L	L	L
L	H	L
H	L	L
H	H	H

λειτουργίας των διαφόρων λογικών κυκλωμάτων. Θα δοϋμε στις επόμενες παραγράφους την χρήση αυτών των πινάκων.

Έτσι όπως αντικαταστήσαμε τα σύμβολα H και L με τό 1 και τό 0, εφαρμόσαμε τη λεγόμενη «θετική λογική». Θα μπορούσαμε όμως τό ίδιο εύκολα νά αντικαταστήσουμε τό λογικό 1 με τό χαμηλό δυναμικό των 0V και τό λογικό 0 με τό ύψηλό δυναμικό των +5V. Τότε θα λέγαμε ότι εφαρμόσαμε τη λεγόμενη «άρνητική λογική».

Η έκλογή της θετικής ή της άρνητικής λογικής, είναι τελείως αυθαίρετη γιά τόν κατασκευαστή των λογικών κυκλωμάτων. Στην πραγματικότητα, καμία δέν πλεονεκτεί από την άλλη. Γιά χάρη απλότητας θα χρησιμοποιούμε από έδω καί στό έξής τη θετική λογική ($H \Rightarrow 1$, $L \Rightarrow 0$).

Όπως ήδη αναφέραμε στην Άλγεβρα του Boole, οι τρείς βασικές λογικές πράξεις είναι: ΚΑΙ (AND), Ή (OR) καί ή άρνηση (NOT). Τά λογικά κυκλώματα τά όποια πραγματοποιούν τίς λογικές αυτές πράξεις λέγονται **πύλες** (gates). Στόν πίνακα 4.2.3 παριστάνονται τά σύμβολα πού χρησιμοποιούνται συνήθως γιά τη γραφική παράσταση των πυλών κατά τη σχεδίαση των λογικών κυκλωμάτων με τούς πίνακες άληθείας τους. Στόν ίδιο πίνακα φαίνονται καί μερικά άλλα λογικά κυκλώματα με τούς πίνακες άληθείας καί τά σύμβολά τους.

4.2.2 Πραγματοποίηση λογικών κυκλωμάτων με διακεκριμένα στοιχεία.

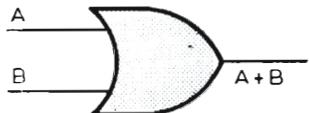
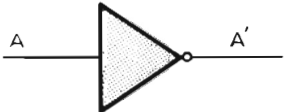
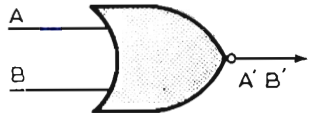
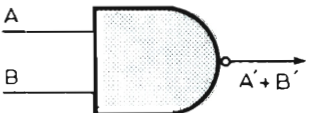
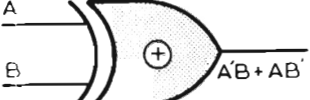
α) Πύλη Ή (OR).

Στό σχήμα 4.2β παριστάνεται μία πύλη Ή στή θετική λογική. Άν σέ μία από τίς εισόδους A, B ή καί στίς δύο, εμφανισθεΐ μία θετική τάση, τότε ή αντίστοιχη είσοδος γίνεται άγώγιμη καί ή θετική τάση εμφανίζεται στην έξοδο.

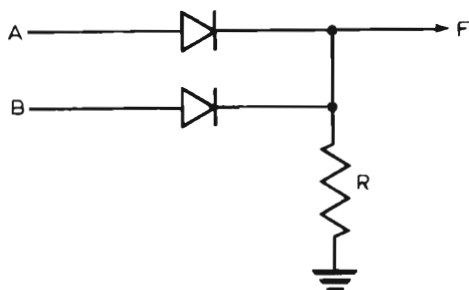
β) Πύλη ΚΑΙ (AND).

Στό σχήμα 4.2γ παριστάνεται τό ηλεκτρονικό κύκλωμα μιās πύλης ΚΑΙ στή θε-

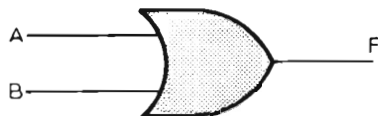
ΠΙΝΑΚΑΣ 4.2.3.**Σύμβολα γραφικής απεικόνισης λογικών κυκλωμάτων και πίνακες αληθείας.**

Λογική Συνάρτηση	Σύμβολο	Πίνακας Αληθείας															
Βασικά λογικά κυκλώματα																	
ΚΑΙ (AND)		<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
Ή (OR)		<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
Άρνηση (NOT)		<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0									
A	F																
0	1																
1	0																
Σύνθετα λογικά κυκλώματα																	
ΟΧΙ Ή (NOR)		<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
A	B	F															
0	0	1															
0	1	0															
1	0	0															
1	1	0															
ΟΧΙ ΚΑΙ (NAND)		<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F															
0	0	1															
0	1	1															
1	0	1															
1	1	0															
ΑΠΟΚΛΕΙΣΤΙΚΗ Ή (EXCLUSIVE OR)		<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F															
0	0	0															
0	1	1															
1	0	1															
1	1	0															

τική λογική. Αν μία από τις εισόδους τεθεί σε δυναμικό μηδέν, θα γίνει αγωγή και στην έξοδο θα εμφανισθεί το ίδιο μηδενικό δυναμικό, το οποίο σύμφωνα με τη θετική λογική αντιστοιχεί στη λογική κατάσταση 0. Αν αντίθετα και οι δύο εισόδους τεθούν συγχρόνως σε ένα θετικό δυναμικό, ίσο προς το δυναμικό τροφοδοσίας V_{cc} , οι δύο δίοδοι παύουν να άγουν και η έξοδος αποκτά το θετικό δυναμικό τροφοδοσίας που αντιστοιχεί στην κατάσταση 1.



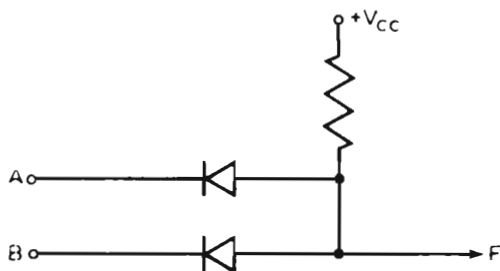
(α)



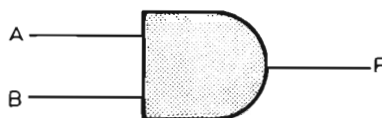
(β)

Σχ. 4.2β.

α) Πραγματοποίηση μίας πύλης Ή (OR) στή θετική λογική. β) Ό γραφικός συμβολισμός τής πύλης OR.



(α)



(β)

Σχ. 4.2γ.

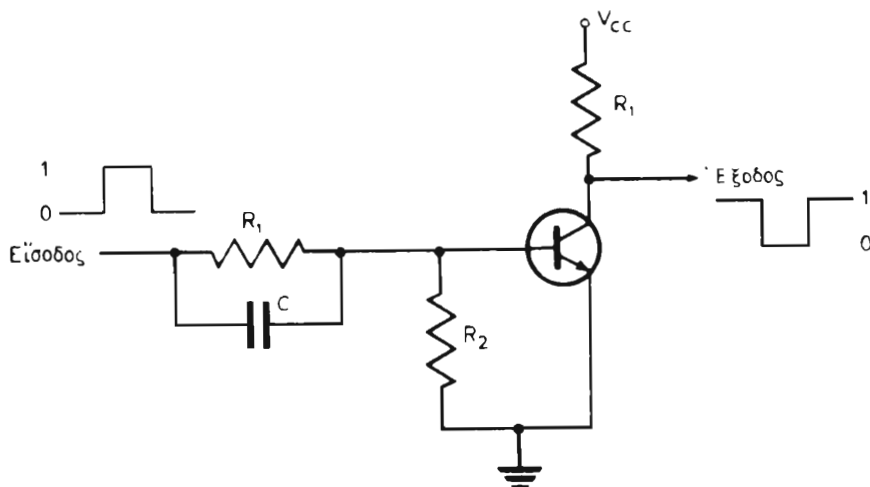
α) Πραγματοποίηση μιᾶς πύλης ΚΑΙ (AND) στή θετική λογική. β) Ό γραφικός συμβολισμός τής πύλης ΚΑΙ.

γ) Κύκλωμα ΟΧΙ ή Άντιστροφέας (NOT).

Στό σχήμα 4.2δ παριστάνεται τό ηλεκτρονικό κύκλωμα ἑνός ἀντιστροφέα (NOT) στή θετική λογική. Γιά τήν πραγματοποίησή του χρησιμοποιοῦμε ἕνα τρανζίστορ τύπου npn καί ἀντιστάσεις. Γιά νά ἐπιτύχομε τήν ἄρνηση τής εισόδου, ἐκμεταλλευόμαστε τή δυνατότητα λειτουργίας τοῦ τρανζίστορ σέ δύο διακεκριμένες καταστάσεις ὅπως φαίνεται στίς χαρακτηριστικές καμπύλες τοῦ σχήματος 4.2ε.

Ἡ μία κατάσταση εἶναι τής πλήρους ἀγωγιμότητας (σημεῖο 1 στό σχήμα 4.2ε), στό ὁποῖο ἡ τάση $V_{ce, κορ}$ εἶναι πολύ χαμηλή, (0,3 - 0,7 Volt), καί ἡ δεύτερη εἶναι ἡ κατάσταση ἀποκοπῆς (σημεῖο 2 στό σχήμα 4.2ε) πού ἀντιστοιχεῖ στήν τάση V_{cc} (4 - 5 Volt).

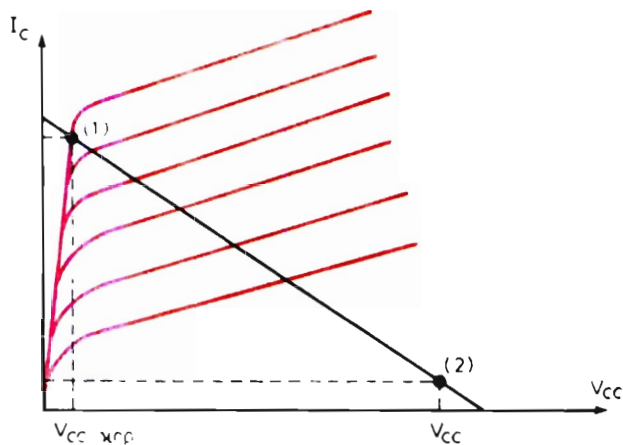
Ἐτσι, ἂν τό τρανζίστορ βρίσκεται στήν κατάσταση ἀποκοπῆς, στήν ἐξοδο θά ἔχομε τάση περίπου ἴση μέ V_{cc} (θετική), τήν ὁποία καί ἀντιστοιχοῦμε στήν κατάσταση 1. Ἄν τώρα στήν εἴσοδο ἐφαρμόσουμε ἕνα θετικό παλμό, τό τρανζίστορ ἀρχίζει νά ἄγει. Ἐτσι προκαλεῖται μία πώση τάσεως στήν ἐξοδο στήν τιμή $V_{ce, κορ} = 0,3 - 0,7 \text{ V}$ στήν ὁποία μπορούμε νά ἀντιστοιχήσουμε τήν κατάσταση 0. Ἐτσι ὁ παλμός πού τροφοδοτοῦμε στήν εἴσοδο βγαίνει στήν ἐξοδο κατεστραμμένος.



Σχ. 4.26.

Τό ηλεκτρονικό κύκλωμα άντιστροφέα (NOT) στη θετική λογική.

Με τό συνδυασμό του άντιστροφέα και τίς πύλες ΚΑΙ και Ή των προηγούμενων κυκλωμάτων, μπορούμε νά κατασκευάσουμε πιο πολύπλοκα ηλεκτρονικά λογικά κυκλώματα.



Σχ. 4.2ε.

Χαρακτηριστικές καμπύλες ενός τρανζίστορ πηρ.

4.3 Συνδυαστικά Λογικά Κυκλώματα.

Ή έξοδος στα συνδυαστικά κυκλώματα είναι συνάρτηση μόνο των εισόδων του κυκλώματος. Αυτό όμως δέ συμβαίνει στα ακολουθιακά κυκλώματα.

Στήν παράγραφο αυτή θά δούμε πώς μπορούμε νά σχεδιάσουμε ένα συνδυαστικό λογικό κύκλωμα με πύλες.

4.3.1 Κατασκευή του λογικού κυκλώματος από τη συνάρτηση.

Κατ' αρχήν προσπαθούμε να φέρουμε τη συνάρτηση στην πιο απλή της μορφή. Στη συνέχεια παρατηρούμε πόσες και ποιές πράξεις σημειώνονται στην τελική απλοποιημένη έκφραση της συναρτήσεως, γιατί από αυτό θα εξαρτηθεί ποιές και πόσες πύλες θα χρησιμοποιήσουμε. Μετά από αυτό προχωρούμε σταδιακά στη σχεδίαση του κυκλώματος.

Παρακάτω δίνονται δύο παραδείγματα κατασκευής λογικού κυκλώματος από τη συνάρτηση.

Παράδειγμα 1.

Νά σχεδιασθεί το λογικό κύκλωμα της συναρτήσεως:

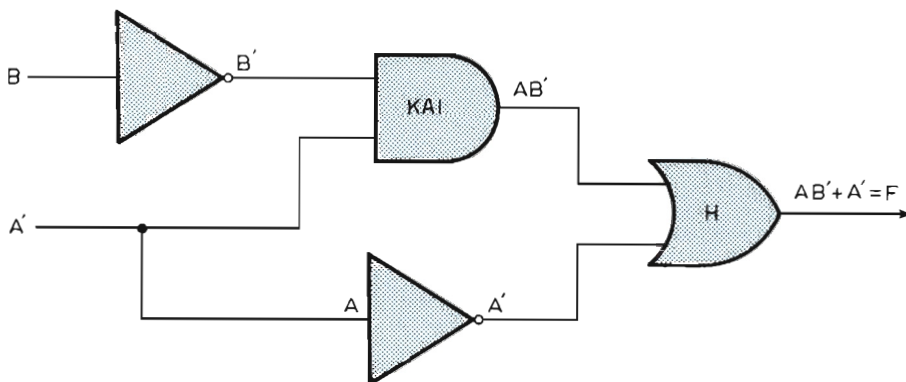
$$F = AB' + A'B + A'B'$$

μέ πύλες ΚΑΙ, Ή και αντιστροφεῖς.

Καταρχήν προσπαθούμε να φέρουμε τη συνάρτηση αυτή σε πιο απλοποιημένη μορφή:

$$\begin{aligned} F &= AB' + A'B + A'B' \\ &= AB' + A' \cdot (B + B') \\ &= AB' + A' \end{aligned}$$

Από την απλοποιημένη σχέση βλέπουμε ότι αν έχουμε τις δύο λογικές εισόδους Α και Β, τότε το κύκλωμά μας θα πρέπει να πραγματοποιεί δύο αντιστροφές (A' , B'), μία λογική πρόσθεση ($AB' + A'$) και ένα λογικό πολλαπλασιασμό (AB'). Έτσι προχωρούμε στη σχεδίαση του κυκλώματος όπως φαίνεται στο σχήμα 4.3α.

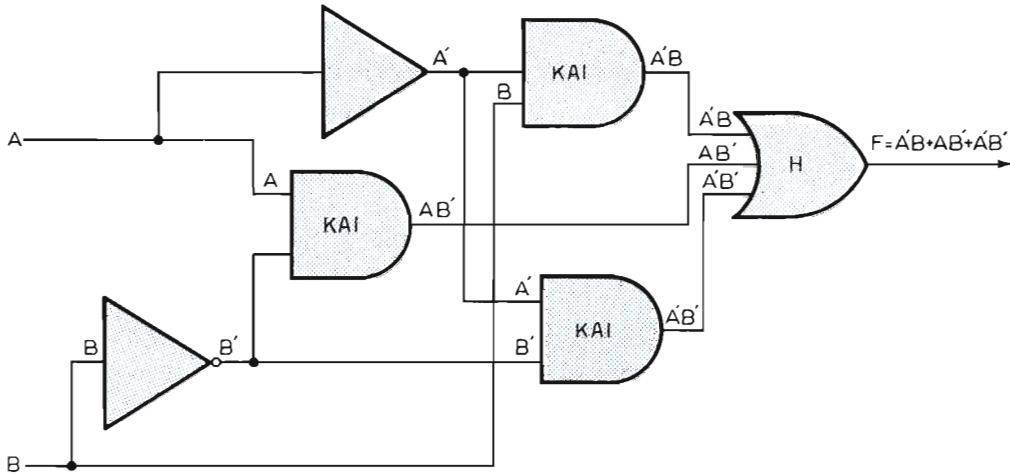


Σχ. 4.3α.

Το λογικό κύκλωμα της συναρτήσεως $F = AB' + A'$.

Αν δεν είχαμε απλοποιήσει τη συνάρτηση τότε το λογικό κύκλωμα, το οποίο θα έπρεπε να σχεδιάσουμε είναι αυτό που φαίνεται στο σχήμα 4.3β.

Αν συγκρίνουμε τα δύο αυτά λογικά διαγράμματα θα καταλάβουμε ότι είναι αναγκαία η απλοποίηση των λογικών συναρτήσεων πριν από οποιαδήποτε προσπάθεια σχεδίασεως του λογικού κυκλώματος.



Σχ. 4.3β.

Τό λογικό διάγραμμα της συνάρτησεως $F = AB' + A'B + A'B'$ χωρίς τήν άπλοποίηση.

Παράδειγμα 2.

Νά σχεδιασθεῖ τό λογικό κύκλωμα τής συνάρτησεως:

$$F = A'B' + C'D' + AC$$

Έδω παρατηρούμε ότι δέν είναι δυνατή καμιά άπλοποίηση. Έτσι προχωρούμε στή σχεδίαση τοῦ λογικοῦ διαγράμματος. Παρατηρώντας τή συνάρτηση, βλέπομε ότι χρειαζόμαστε τέσσερις αντιστροφείς (A' , B' , C' , D'), τρεῖς λογικούς πολλαπλασιαστές καί ένα τριπλό λογικό άθροιστή. Τό τελικό κύκλωμα φαίνεται στό σχήμα 4.3γ.

4.3.2 Εύρεση τής λογικῆς συνάρτησεως από τό διάγραμμα τοῦ λογικοῦ κυκλώματος.

Έδω ἡ ὅλη εργασία γίνεται ὡς ἐξῆς:

Ξεκινούμε από άριστερά πρὸς τά δεξιά τοῦ διαγράμματος καί σημειώνομε τήν έξοδο κάθε λογικοῦ κυκλώματος. Τέλος στό τελικό στάδιο σημειώνομε τά αποτελέσματα τοῦ προηγούμενου βήματος πού είναι ἡ τιμή τής τελικῆς λογικῆς συνάρτησεως.

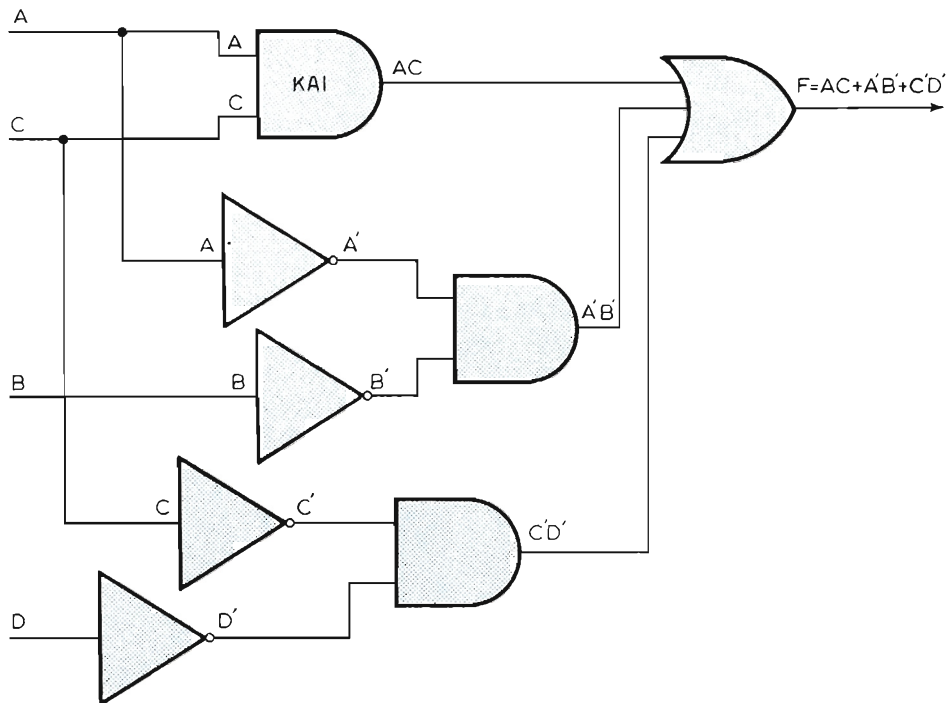
Παρακάτω δίνονται δύο παραδείγματα εύρέσεως λογικῆς συνάρτησεως από διάγραμμα λογικοῦ κυκλώματος.

Παράδειγμα 1.

Νά βρεθεῖ ἡ λογική συνάρτηση τοῦ κυκλώματος τοῦ σχήματος 4.3δ.

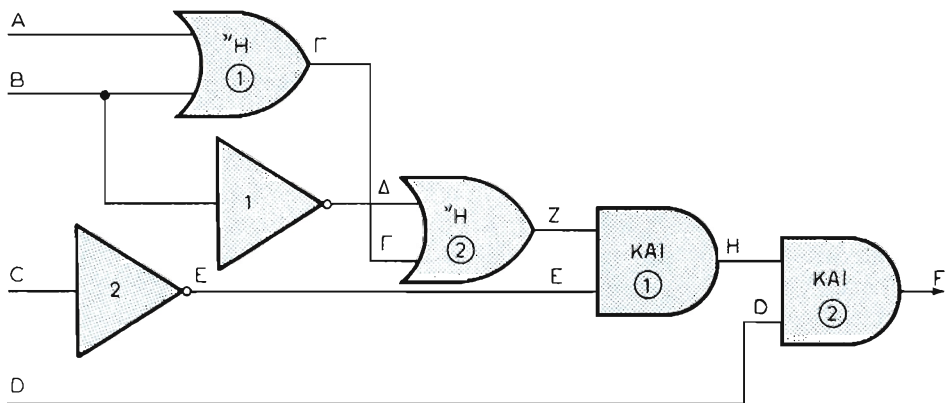
Λύση:

Σταδιακά εργαζόμενοι προσδιορίζομε τίς αλλαγές οἱ ὁποῖες προκύπτουν από τά διάφορα στοιχεῖα τοῦ λογικοῦ κυκλώματος ὡς ἐξῆς:



Σχ. 4.3γ.

Το λογικό κύκλωμα της συναρτήσεως $F = AC + A'B' + C'D'$.



Σχ. 4.3δ.

$$\begin{aligned}
 \text{'Η έξοδος της πύλης 'H No 1 είναι:} & \quad \Gamma = A + B \\
 \text{'Η έξοδος της πύλης 'H No 2 είναι:} & \quad Z = \Delta + \Gamma = B' + A + B \\
 \text{'Η έξοδος της πύλης ΚΑΙ No 1 είναι:} & \quad H = ZE \\
 & \quad = (B' + A + B) (C') \\
 & \quad = C'B' + C'A + C'B \\
 \text{'Η έξοδος της πύλης ΚΑΙ No 2 είναι:} & \quad F = HD \\
 & \quad = (C'B' + C'A + C'B)D \\
 & \quad = DC'B' + DC'A + DC'B
 \end{aligned}$$

Παράδειγμα 2.

Νά βρεθεί η λογική συνάρτηση F του λογικού κυκλώματος του σχήματος 4.3ε.

Επειδή εδώ τό σχήμα είναι πολύπλοκο, σημειώνομε τίς διάφορες εξόδους ως εξής:

- 1 $\rightarrow A'$
- 2 $\rightarrow A'B'$
- 3 $\rightarrow A'B'C'$
- 4 $\rightarrow B'$
- 5 $\rightarrow A'B$
- 6 $\rightarrow A'BC'$
- 7 $\rightarrow C'$
- 8 $\rightarrow AB'$
- 9 $\rightarrow AB$
- 10 $\rightarrow ABC$
- 11 $\rightarrow AB'C'$
- 12 $\rightarrow AB'C' + ABC$
- 13 $\rightarrow A'B'C + A'BC'$
- 14 $\rightarrow AB'C' + ABC + A'B'C + A'BC'$

Επομένως:

$$F = AB'C' + ABC + A'B'C + A'BC'$$

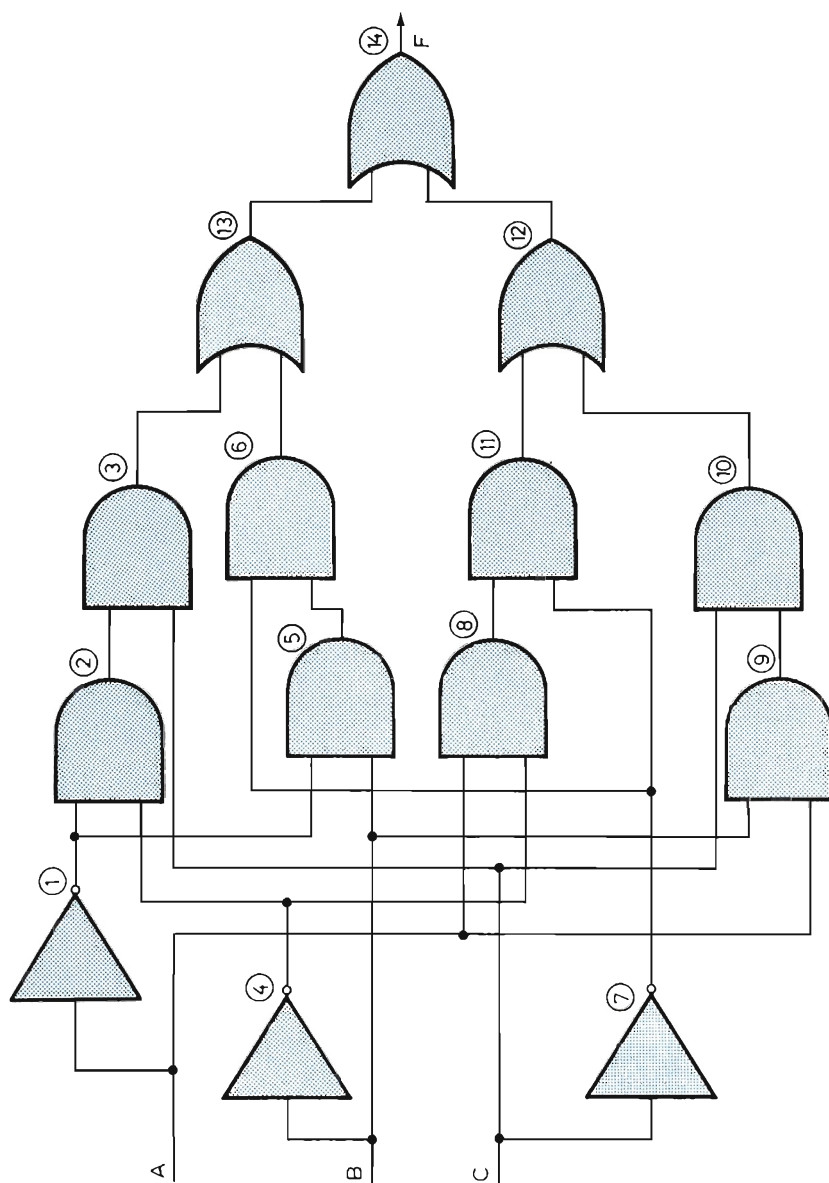
4.3.3 Τά συνδυαστικά κυκλώματα στον Ήλεκτρονικό Υπολογιστή.

Στήν παράγραφο αυτή θά περιγράψομε τίς αρχές λειτουργίας διαφόρων συνδυαστικών κυκλωμάτων τά όποϊα αποτελούν πολύ χρήσιμα στοιχεία στή λειτουργία ενός ηλεκτρονικού υπολογιστή. Γιά τήν ακρίβεια, πολλά από αυτά αποτελούν τά απαραίτητα δομικά στοιχεία γιά τήν έκτέλεση τών διαφόρων λογικών πράξεων στίς λειτουργικές μονάδες του ήλεκτρονικού υπολογιστή.

Παραδείγματος χάρη, ή κάθε πληροφορία πού θέλομε νά διακινήσομε ή νά επεξεργασθοΰμε μέσα στον υπολογιστή θά πρέπει νά μεταφερθεί στή γλώσσα πού ό υπολογιστής «καταλαβαίνει». Καί φυσικά τό αντίθετο συμβαίνει στήν έξοδο του υπολογιστή, όταν τά αποτελέσματα παρουσιάζονται στον τελικό τους δέκτη, τόν άνθρωπο.

Έτσι δημιουργήθηκαν οι διάφοροι κωδικοποιητές καί αποκωδικοποιητές πού είναι λογικά κυκλώματα τά όποϊα μετατρέπουν τήν πληροφορία σέ τέτοια μορφή, ώστε νά είναι κατανοητή από τόν αποδέκτη της, όποιος καί αν είναι αυτός.

Επίσης κατά τήν επεξεργασία τών διαφόρων πληροφοριών απαιτείται έκτέλεση



Σχ. 4.3ε.

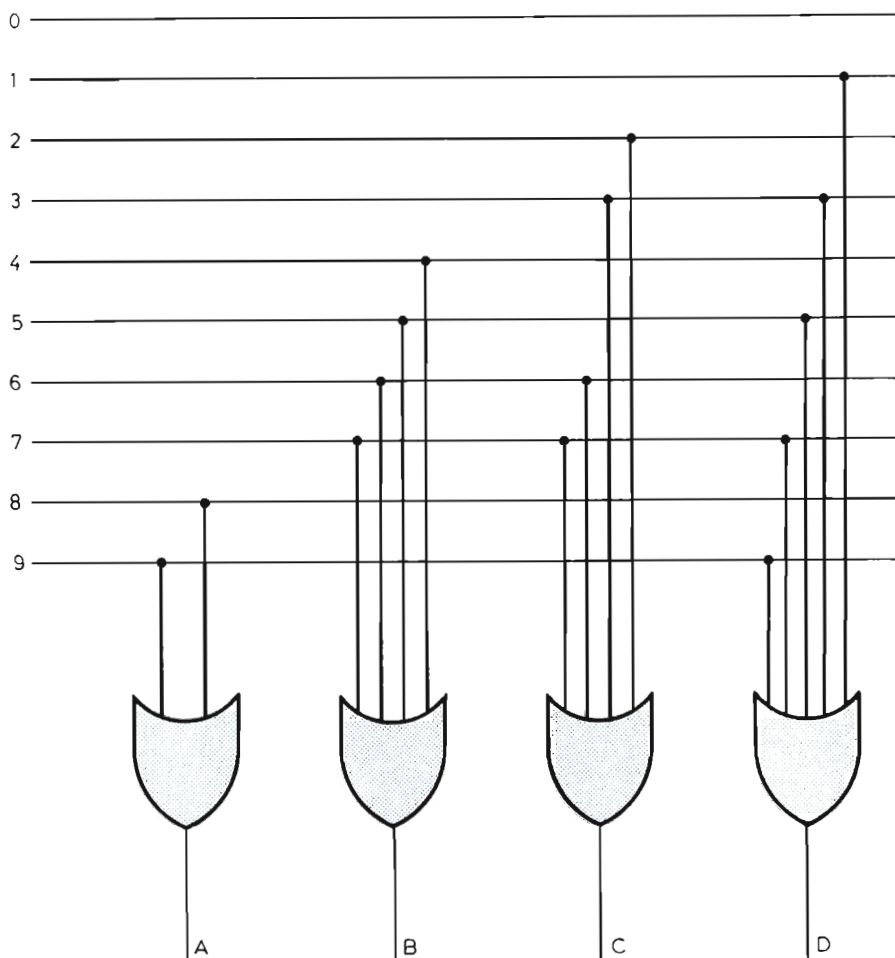
διαφόρων αριθμητικῶν πράξεων ὅπως ἡ πρόσθεση ἡ ἀφαίρεση, ὁ πολλαπλασιασμός, καὶ ἡ διαίρεση ἢ λογικῶν πράξεων, ὅπως ἡ σύγκριση δύο μεγεθῶν κ.ά.

α) Κωδικοποιητές καὶ Ἀποκωδικοποιητές.

Εἶναι διατάξεις πού μετατρέπουν μία πληροφορία, ἐκφρασμένη σέ ἓνα κώδικα, σέ κάποιο ἄλλο κώδικα. Ἔτσι, ἡ ἔννοια τῆς κωδικοποιήσεως καὶ τῆς ἀποκωδικοποιήσεως εἶναι ἐντελῶς σχετική καὶ ἐξαρτᾶται ἀπὸ τὸ ποῖόν κώδικα ἢ ἀριθμητικὸ σύστημα θεωροῦμε γνωστὸ ἢ πρὶς συνηθισμένο.

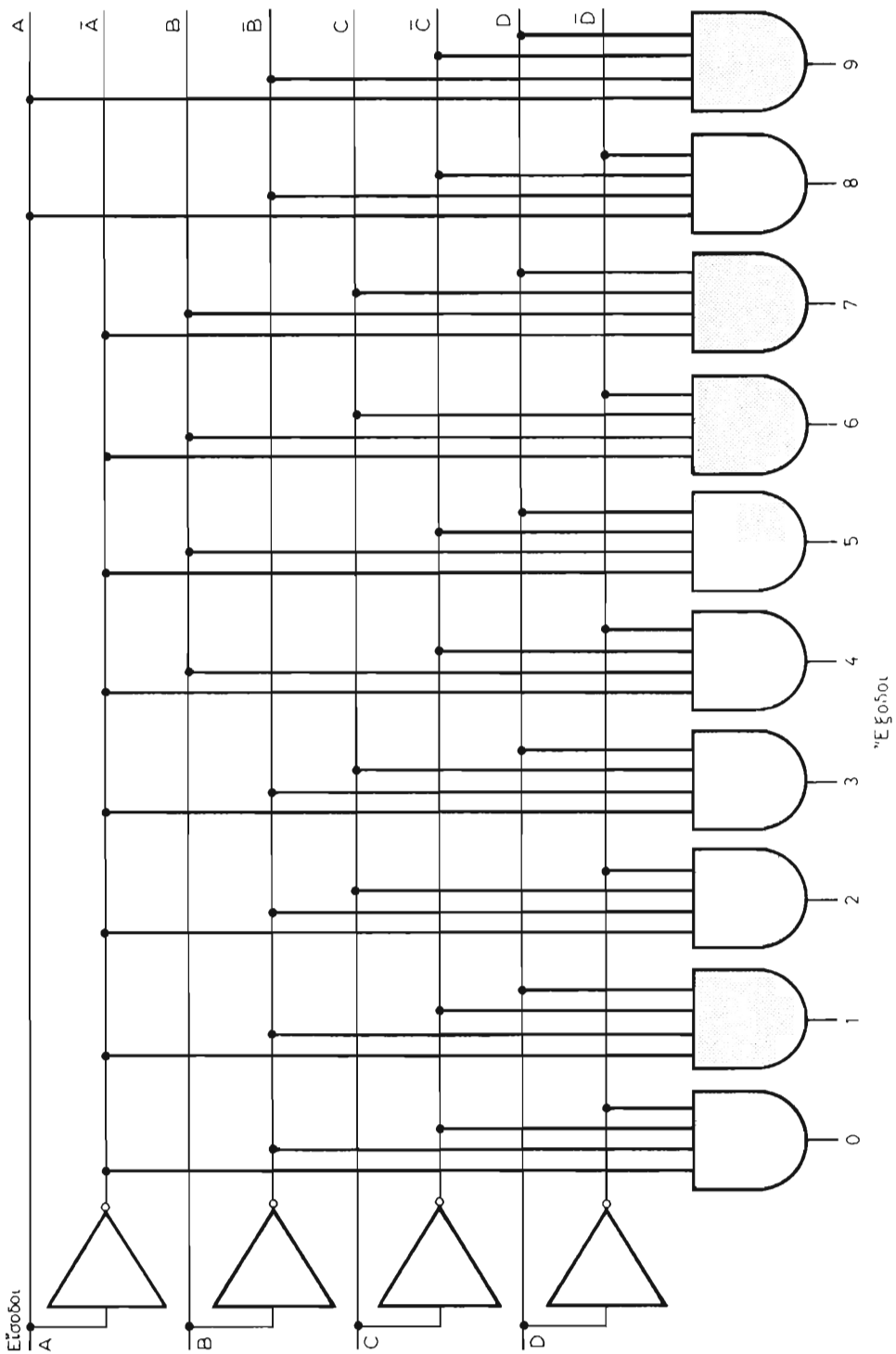
Στά σχήματα 4.3στ καὶ 4.3ζ δίνονται ἀντιστοίχως τὰ διαγράμματα κωδικοποιητῆ καὶ ἀποκωδικοποιητῆ ἑνὸς δεκαδικοῦ ἀριθμοῦ σέ δυαδικό καὶ ἀντιστρόφως. Στὸν πίνακα 4.3.1 δίνεται ἡ ἀντιστοιχία τῆς ἀπαιτούμενης μετατροπῆς.

Εἴσοδοι



Σχ. 4.3στ.

Διάγραμμα κωδικοποιητῆ ἀριθμοῦ ἀπὸ τὸ δεκαδικὸ σύστημα στὸ δυαδικό.



Σχ. 4.3ζ.

Διάγραμμα κωδικοποιητή αριθμού από τó δυαδικό σύστημα στó δεκαδικό.

ΠΙΝΑΚΑΣ 4.3.1.

Δεκαδικό ψηφίο	Δεκαδική μορφή A B C D
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Όπως βλέπομε από τό σχήμα 4.3στ, τό μόνο πού απαιτείται εἶναι 4 πύλες ἡ πολλαπλῶν εισόδων.

Τήν ἀντίστροφη δουλειά κάνει τό λογικό κύκλωμα τοῦ σχήματος 4.3ζ. Ὁ ἀριθμός τῶν εισόδων εἶναι ἐδῶ 4 καί ὁ ἀριθμός τῶν ἐξόδων 10.

β) Κυκλώματα ἀριθμητικῶν πράξεων.

Τά κυκλώματα αὐτά προσδιορίζουν, κυρίως, τήν ἱκανότητα, τοῦ ηλεκτρονικοῦ ὑπολογιστή, ἐκτελέσεως ἀριθμητικῶν πράξεων. Στά σχήματα 4.3η καί 4.3θ φαίνονται ἕνας ἡμισθροιστής καί ἕνας ἡμισφαιρέτης. Ὁ ἡμισθροιστής εἶναι τό ἀπλούστερο κύκλωμα τοῦ Η/Υ καί ἀποτελεῖ τή βάση γιά τήν ἀριθμητική πράξη τῆς προσθέσεως. Προσθέτει μονοψηφίους δυαδικούς ἀριθμούς καί μᾶς δίνει τό τελευταῖο ψηφίο τοῦ ἀποτελέσματος R καί τό ὑπόλοιπο C. Στόν πίνακα 4.3.2 φαίνεται ἡ σχέση μεταξύ τῶν εισόδων A, B καί τῶν ἐξόδων R καί C.

Στόν πίνακα 4.3.3 φαίνεται ἡ σχέση μεταξύ τῶν εισόδων AB καί τῶν ἐξόδων R καί D ἑνός ἡμισφαιρέτη. Μέ τό γράμμα D παρίσταται τό ψηφίο δανεισμοῦ.

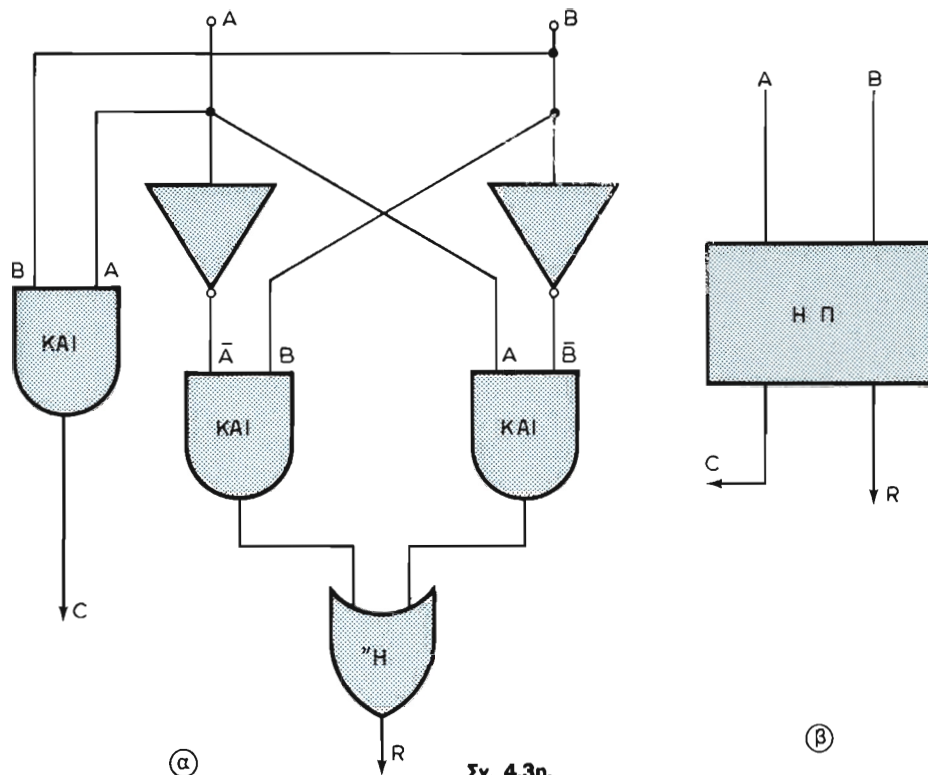
ΠΙΝΑΚΑΣ 4.3.2.**Ἡμισθροιστής.**

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

ΠΙΝΑΚΑΣ 4.3.3.**Ἡμισφαιρέτης.**

A	B	R	D
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Μετά ἀπό αὐτά πού ἀναφέραμε παραπάνω, μπορούμε νά κατασκευάσουμε ἕναν

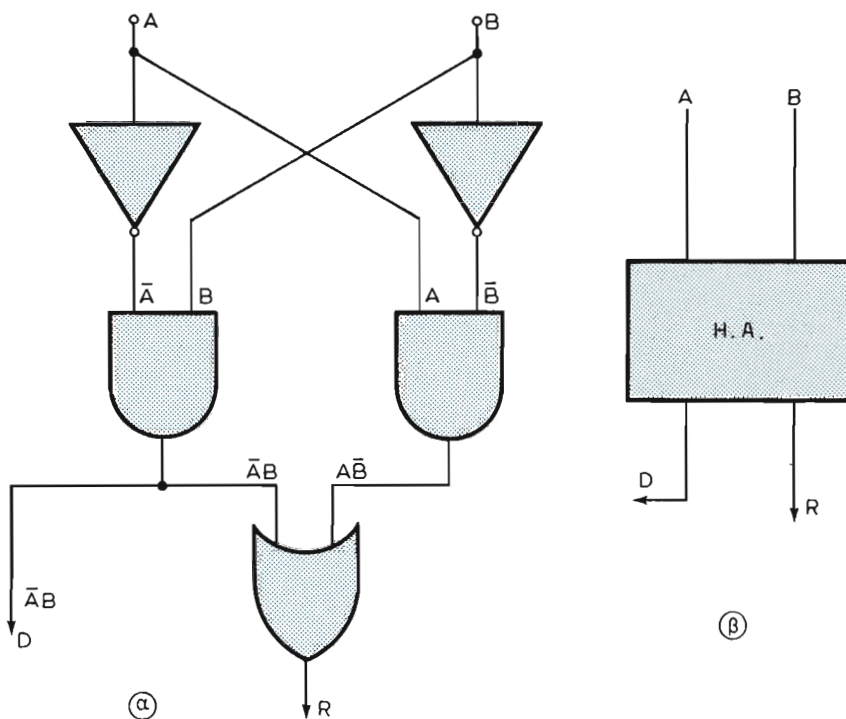


α

β

Σχ. 4.3η.

α) Τό λογικό διάγραμμα ήμισυθροιστή. β) Γενική παράστασή του.



α

β

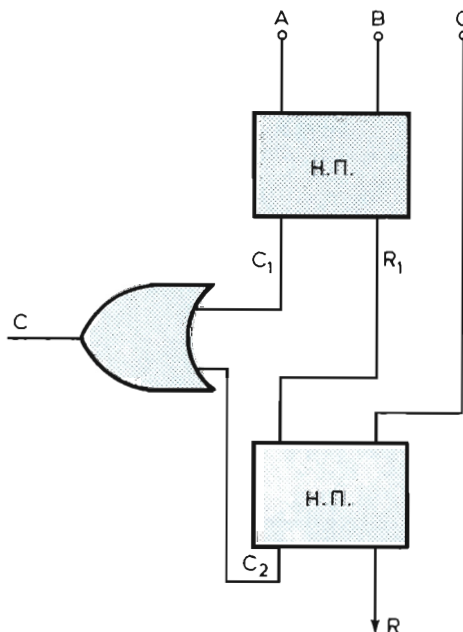
Σχ. 4.3θ.

α) Τό λογικό διάγραμμα ήμισυφαιρέτη. β) Γενική παράστασή του.

πλήρη άθροιστή και έναν πλήρη άφαιρέτη για περισσότερα από δύο ψηφία. Στα σχήματα 4.3ι και 4.3ια φαίνονται τά λογικά κυκλώματα ενός πλήρους άθροιστή και ενός πλήρους άφαιρέτη, ενώ στους πίνακες 4.3.4 και 4.3.5 φαίνεται ή λογική έκά-
στου κυκλώματος.

ΠΙΝΑΚΑΣ 4.3.4.
Πλήρης άθροιστής.

A	B	C	R	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



Σχ. 4.3ι.

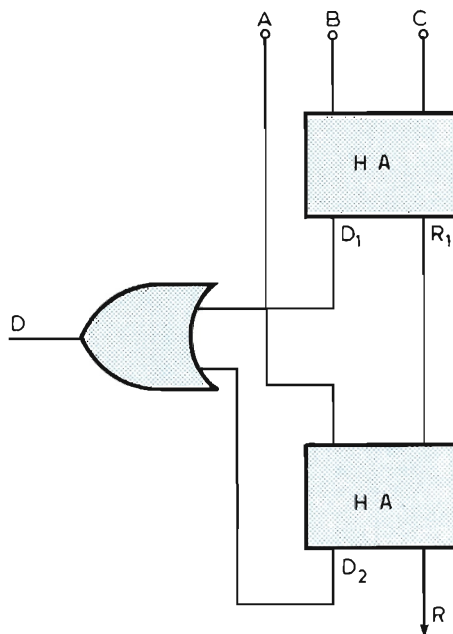
Τό λογικό διάγραμμα πλήρους άθροιστή. Προσέξτε ότι τά τετράγωνα με τό συμβολισμό ΗΠ παριστά-
νουν έναν ήμιαθροιστή. Οι συμβολισμοί C_1 , R_1 και C_2 παριστάνουν ενδιάμεσα αποτελέσματα.

Τέλος στό σχήμα 4.3ιβ παριστάνεται τό λογικό κύκλωμα για τόν πολλαπλασια-
σμό.

Γιά νά άντιληφθοϋμε τή λειτουργία αύτοϋ τοϋ κυκλώματος, θά πρέπει καταρχήν
νά θυμηθοϋμε τόν τρόπο με τόν όποιο γίνεται ό πολλαπλασιασμός και ή διαίρεση

ΠΙΝΑΚΑΣ 4.3.5.**Πλήρης άφαιρέτης.**

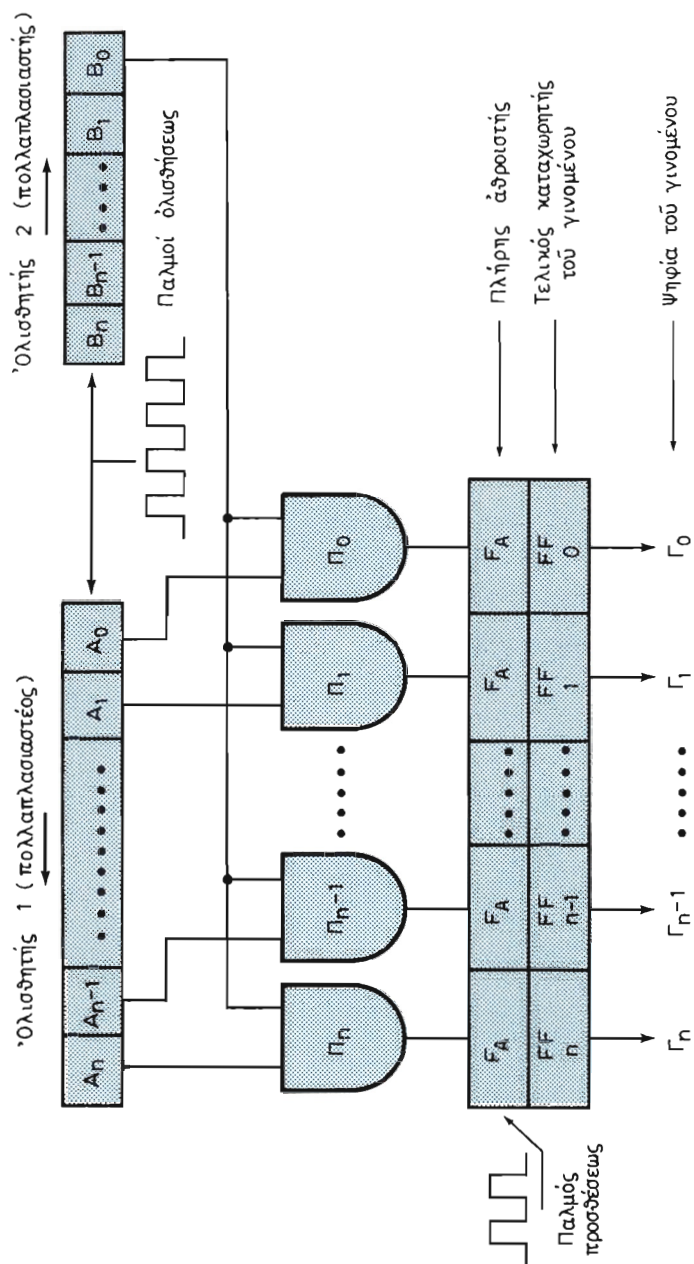
A	B	C	R	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

**Σχ. 4.3ια.**

Τό λογικό διάγραμμα ενός πλήρους άφαιρέτη. Προσέξτε ότι τά τετράγωνα μέ τό συμβολισμό HA παριστάνουν έναν ήμιαφαιρέτη. Οί συμβολισμοί R_1 , D_1 καί D_2 παριστάνουν ένδιάμεσα άποτελέσματα.

δύο δυαδικών αριθμών. Έπίσης έδώ θά πρέπει νά αναφερθοῦμε, έστω καί συνοπτικά, στά στοιχεΐα μνήμης άν καί αυτά θά άναπτυχθούν καλύτερα στην έπόμενη παράγραφο.

Όρίζομε ώς ηλεκτρονικό στοιχείο μνήμης ή πολυδονητή (Flip - Flop) ένα κύκλωμα μέ μία ή περισσότερες εισόδους καί μία ή περισσότερες έξόδους, τό όποιο μπορεί νά πάρει μία άπό δύο διακεκριμένες καταστάσεις κάθε φορά καί νά τή διατηρήσει κάτω άπό όρισμένες συνθήκες εισόδου. Μία σειρά άπό Flip - Flop πού άπομνημονεύουν μία δυαδική άκολουθία τής μορφής 10010001111... όνομάζεται **καταχωρητής**.



Σχ. 4.3ιβ.

Τό λογικό κύκλωμα ενός πολλαπλασιαστή.

“Αν τώρα ο καταχωρητής αυτός μπορεί να μεταφέρει κατά κάποιο τρόπο το περιεχόμενό του προς τα αριστερά ή προς τα δεξιά, τότε λέγεται **όλισθητής**.

Τώρα, αν ένας καταχωρητής μπορεί να προσθέσει στο περιεχόμενό του το περιεχόμενο ενός άλλου καταχωρητή ή όλισθητή και να κρατήσει το άθροισμα, τότε ονομάζεται **συσσωρευτής**.

Έτσι η λειτουργία του κυκλώματος του πολλαπλασιασμού σε γενικές γραμμές έχει ως εξής:

Εισάγουμε στους όλισθητές 1 και 2 τούς δύο αριθμούς που θέλουμε να πολλαπλασιάσουμε. “Αν τό λιγότερο σημαντικό ψηφίο (LSB) του πολλαπλασιαστή είναι 1, οι πύλες $\Pi_n \dots \Pi_0$ ενεργοποιούνται και ο πολλαπλασιαστής καταχωρείται στο συσσωρευτή. “Αν $B_0 = 0$, τότε οι πύλες παραμένουν αδρανείς.

Μέ τον πρώτο παλμό όλισθήσεως τό περιεχόμενο του όλισθητή 1 μετατοπίζεται μία θέση προς τα αριστερά και του 2 μία θέση προς τα δεξιά. Έτσι στον πολλαπλασιαστή ως μικρότερο σημαντικό ψηφίο υπάρχει τώρα τό B_1 .

“Αν τό $B_1 = 0$, τίποτε δέν αλλάζει, ενώ εάν $B_1 = 1$, τότε ο πολλαπλασιαστής, μετατοπισμένος μία θέση αριστερά, καταχωρείται και προστίθεται στο παλιό περιεχόμενο του καταχωρητή, ύστερα από κάποιον παλμό όπως φαίνεται στο σχήμα 4.3ιβ.

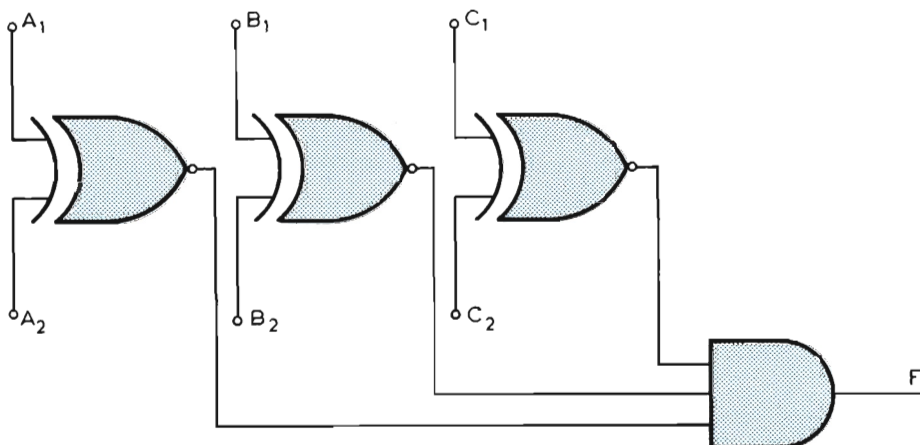
Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου τό πιδ σημαντικό ψηφίο του πολλαπλασιαστή (B_n) εμφανισθεί στην τελευταία δεξιά θέση του πολλαπλασιαστή.

Κάπως ανάλογα δημιουργείται και τό αντίστοιχο κύκλωμα τής διαιρέσεως δύο δυαδικών αριθμών, μέ τή διαφορά ότι εδώ πλέον κάνομε αφαίρεση αντί γιά πρόσθεση.

γ) Κυκλώματα συγκρίσεως.

Τά κυκλώματα αυτά χρησιμοποιούνται κυρίως γιά τήν σύγκριση δύο δυαδικών αριθμών.

Ένα τέτοιο κύκλωμα φαίνεται στο σχήμα 4.3ιγ.



Σχ. 4.3ιγ.

Τό λογικό κύκλωμα τής συγκρίσεως δύο δυαδικών αριθμών.

Ἡ ἔξοδος F τοῦ κυκλώματος γίνεται 1, μόνο ὅταν $A_1 = A_2$, $B_1 = B_2$ καὶ $C_1 = C_2$. Διαφορετικά: $F = 0$.

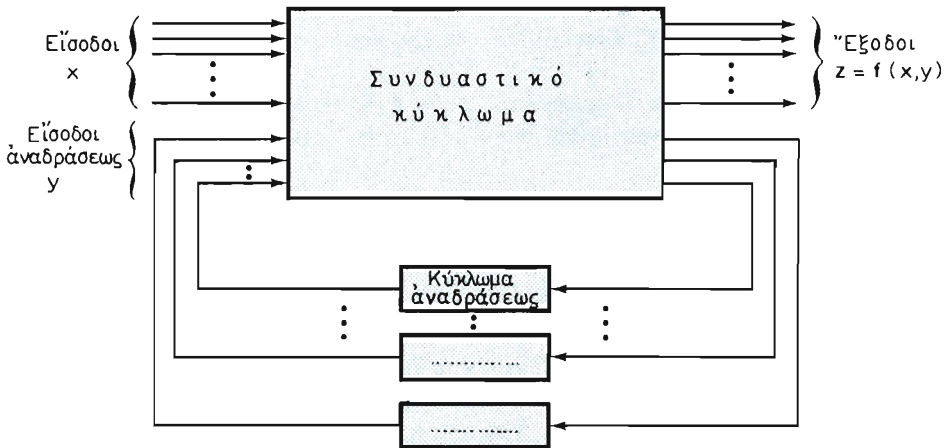
Τό παραπάνω κύκλωμα μιᾶς πύλης «ΑΠΟΚΛΕΙΣΤΙΚΗ Η» μέ ἄρνηση λέγεται κύκλωμα συμπτώσεως καὶ ἡ ἔξοδός του δίνεται ἀπὸ τὴ σχέση:

$$F_1 = AB + A'B' = A \odot B$$

4.4 Ἀκολουθιακά κυκλώματα.

Στά ἀκολουθιακά κυκλώματα ἡ ἔξοδος ἐξαρτᾶται ὄχι μόνο ἀπὸ τὶς εἰσόδους σέ μία χρονικὴ στιγμή, ἀλλὰ καὶ ἀπὸ τὶς τιμές τῶν εἰσόδων σέ προηγούμενη χρονικὴ στιγμή. Μὲ λίγα λόγια τὰ ἀκολουθιακά κυκλώματα «θυμοῦνται» τὶς προηγούμενες εἰσόδους καὶ ἔτσι τὰ χαρακτηρίζομε ὡς κυκλώματα μνήμης.

Στὸ σχῆμα 4.4α φαίνεται μία χαρακτηριστικὴ μορφή ἀκολουθιακοῦ κυκλώματος.



Σχ. 4.4α.

Σχηματικὴ παράσταση ἀκολουθιακοῦ κυκλώματος.

Ὅπως φαίνεται ἀπὸ τὸ σχῆμα ἓνα ἀκολουθιακὸ κύκλωμα μπορεῖ νὰ πραγματοποιηθεῖ μέ ἓνα συνδυαστικὸ κύκλωμα καὶ ἓνα κύκλωμα ἀναδράσεως.

Ἀνάλογα μέ τὸν χρονισμό τους, τὰ ἀκολουθιακά κυκλώματα μποροῦν νὰ διακριθοῦν σέ δύο κατηγορίες:

α) **Στά σύγχρονα ἀκολουθιακά κυκλώματα**, στά ὁποῖα οἱ εἰσοδοὶ ἐφαρμόζονται σέ τακτές χρονικὲς στιγμές οἱ ὁποῖες ρυθμίζονται ἀπὸ μίᾶ πηγή χρονισμού πού ὀνομάζεται ρολοῖ (clock). Οἱ ἔξοδοι λαμβάνονται ἐπίσης σέ τακτές χρονικὲς στιγμές, πού εἶναι συνάρτηση τῶν εἰσόδων (κανονικῶν καὶ ἀναδράσεως) καὶ τῶν χρόνων καθυστερήσεως τοῦ κυκλώματος λόγω τῆς ἀναδράσεως.

β) **Στά ἀσύγχρονα ἀκολουθιακά κυκλώματα**, στά ὁποῖα δέν ὑπάρχει συγχρονισμός καὶ στά ὁποῖα ἡ ταχύτητα ἐκτελέσεως τῆς λογικῆς διαδικασίας ἐξαρτᾶται ἀπὸ τὸ ἴδιο τὸ κύκλωμα.

Τό στοιχείο μνήμης, πού χρησιμοποιούμε συνήθως στά άκολουθιακά κυκλώματα, είναι ένα ηλεκτρονικό κύκλωμα του οποίου ή έξοδος έχει δύο καταστάσεις και ονομάζεται **ηλεκτρονικό στοιχείο μνήμης ή πολυδονητής** (φλίπ - φλόπ).

4.4.1 Τύποι τών ηλεκτρονικών στοιχείων μνήμης.

Χρησιμοποιούνται πολλοί τύποι πολυδονητών ανάλογα μέ τή λειτουργία τους. Στόν πίνακα 4.4.1 φαίνονται οι κυριότεροι από αυτούς. Γενικά, μπορούμε νά πούμε ότι ένας **πολυδονητής** έχει εισόδους «πληροφορίας» και εισόδους «έλέγχου». Στίς εισόδους πληροφορίας εφαρμόζονται οι άκολουθίες παλμών πού θά τοποθετηθούν στή μνήμη ή θά διαβασθούν μέσω τών πολυδονητών. Ένώ στίς εισόδους έλέγχου φθάνουν παλμοί πού κάνουν διάφορες εργασίες, όπως είναι τό ρολόι συγχρονισμού, παλμοί μηδενισμού τής έξόδου, και παλμοί προτοποθετήσεως, οι όποιοι θέτουν τήν έξοδο στή λογική κατάσταση «1».

Σέ κάθε **πολυδονητή** υπάρχουν δύο συμπληρωματικές έξοδοι Q και $\bar{Q}$. Στούς περισσότερους ή κατάσταση τής έξόδου δέν αλλάζει όταν φθάνουν οι παλμοί τών δεδομένων αλλά όταν φθάνει ό παλμός του ρολογιού συγχρονισμού.

Μέ τή βοήθεια τών διαφόρων είδών ηλεκτρονικών στοιχείων μνήμης του πίνακα 4.4.1 μπορούμε νά κατασκευάσουμε τά διάφορα κυκλώματα μνήμης πού χρησιμοποιούνται εύρέως στούς ηλεκτρονικούς ύπολογιστές γιά τήν άπομνημόνευση δυαδικών ψηφίων όπως είναι οι καταχωρητές, οι όλισθητές κ.ά.

Στόν πίνακα 4.4.1, ή περιγραφή του κάθε στοιχείου γίνεται μέ τό νά δίνομε ποιά είναι ή νέα κατάσταση τής έξόδου Q_{n+1} , στήν όποία θά πάει ηλεκτρονικό στοιχείο μνήμης, από μία κατάσταση Q_n πού βρίσκεται, όταν οι είσοδοί του πάρουν τήν τιμή πού δείχνει ό πίνακας.

Γιά τήν άποθήκευση μιås πληροφορίας, επειδή αυτή **κωδικοποιείται** σέ περισσότερα από ένα ψηφία 4,8,16,32 κλπ. χρειάζονται Ισάριθμοι πολυδονητές. Τό σύνολο τών πολυδονητών, πού προορίζονται γιά τήν άποθήκευση μιås πληροφορίας, καλείται «καταχωρητής» (register). Ο πιό άπλός από αυτούς φαίνεται στό σχήμα 4.4β. Ή πληροφορία είναι τά ψηφία $x_1, x_2, \dots, x_n$ τά όποια φθάνουν άπευθείας στίς εισόδους D τών πολυδονητών του καταχωρητή. Μετά τήν άφιξη του παλμού χρονισμού C , ή πληροφορία $x_1 \dots x_n$ φθάνει στίς αντίστοιχες έξόδους $Q_1, Q_2 \dots Q_n$. Έκεί παραμένει μέχρις ότου ή γραμμή R (Read) ένεργοποιηθεί, όποτε ή πληροφορία $x_1, x_2 \dots x_n$ μεταφέρεται στήν έξοδο $z_1, z_2 \dots z_n$. Έτσι έχομε μία «παράλληλη» μεταφορά τής πληροφορίας από τήν είσοδο στήν έξοδο.

Τά δυαδικά ψηφία τής πληροφορίας μπορούν νά μεταφερθούν και σέ σειρά τό ένα μετά τό άλλο από μία μόνο γραμμή. Στήν περίπτωση αυτή ή μετάδοση (έγγραφή και άνάγνωση) γίνεται μέ «όλίσθηση» τών δυαδικών ψηφίων τής πληροφορίας πρós τά δεξιά ή άριστερά. Ένας τέτοιος καταχωρητής καλείται **καταχωρητής όλισθήσεως ή όλισθητής** (shift register) και φαίνεται στό σχήμα 4.4γ.

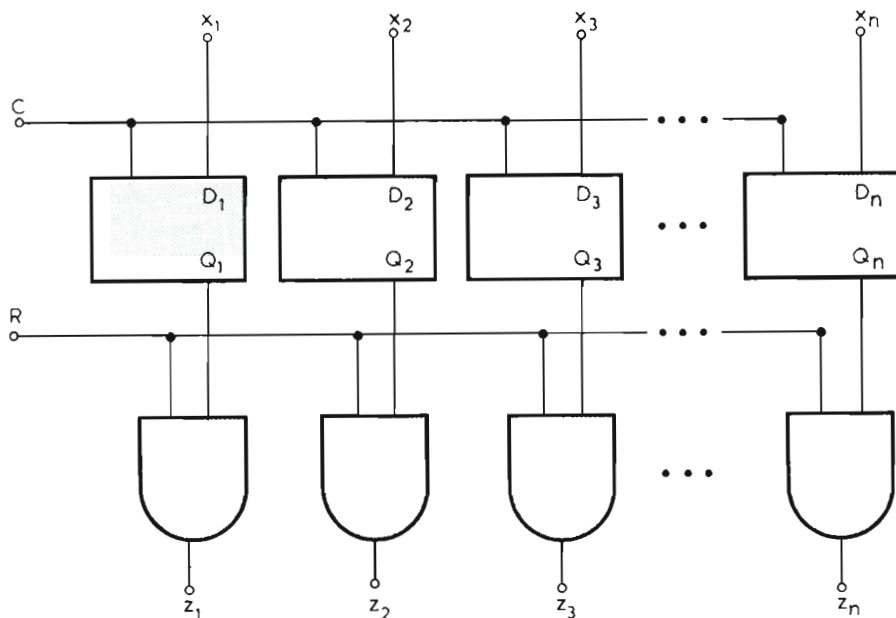
Ό **συσσωρευτής** (accumulator) είναι ένας άθροιστικός καταχωρητής. Δηλαδή προσθέτει δυαδικά τό δυαδικό άριθμό πού προϋπήρχε μέσα σ' αυτόν μέ τό δυαδικό άριθμό πού εισάγεται στήν είσοδό του και κρατά τό άθροισμα πού προκύπτει κάθε φορά από αυτή τή δυαδική πρόσθεση.

Οι συσσωρευτές, οι καταχωρητές και οι όλισθητές αποτελούν τά βασικά κυκλώματα από τά όποια αποτελείται ή «Άριθμητική και Λογική Μονάδα» (ALU:

ΠΙΝΑΚΑΣ 4.4.1.
Τύποι ηλεκτρονικών στοιχείων μνήμης και πίνακες αληθείας τους.

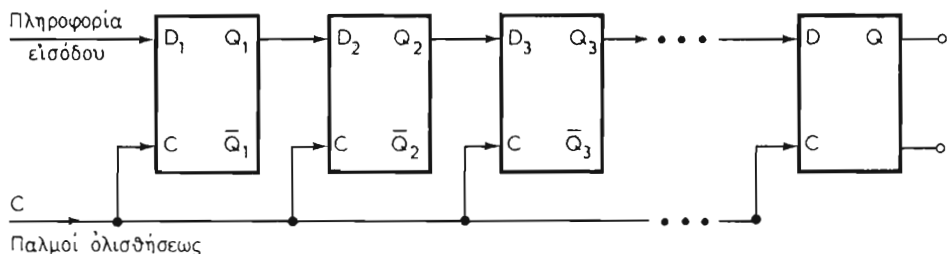
Όνομασία	Γραφικός Συμβολισμός	Πίνακας Άληθείας																																				
Φλίπ - Φλόπ τύπου T (Trigger flip-flop)		<table><tr><th>T</th><th colspan="2">$Q_n \rightarrow Q_{n+1}$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	T	$Q_n \rightarrow Q_{n+1}$		0	0	0	0	1	1	1	0	1	1	1	0																					
T	$Q_n \rightarrow Q_{n+1}$																																					
0	0	0																																				
0	1	1																																				
1	0	1																																				
1	1	0																																				
Φλίπ - Φλόπ τύπου D (Delay flip-flop)		<table><tr><th>D</th><th colspan="2">$Q_n \rightarrow Q_{n+1}$</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	D	$Q_n \rightarrow Q_{n+1}$		0	0	0	0	1	0	1	0	1	1	1	1																					
D	$Q_n \rightarrow Q_{n+1}$																																					
0	0	0																																				
0	1	0																																				
1	0	1																																				
1	1	1																																				
Φλίπ - Φλόπ τύπου R - S (Reset - Set flip - flop)		<table><tr><th>R</th><th>S</th><th colspan="2">$Q_n \rightarrow Q_{n+1}$</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>*</td></tr><tr><td>1</td><td>1</td><td>1</td><td>*</td></tr></table>	R	S	$Q_n \rightarrow Q_{n+1}$		0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	*	1	1	1	*
R	S	$Q_n \rightarrow Q_{n+1}$																																				
0	0	0	0																																			
0	0	1	1																																			
0	1	0	1																																			
0	1	1	1																																			
1	0	0	0																																			
1	0	1	0																																			
1	1	0	*																																			
1	1	1	*																																			
Φλίπ - Φλόπ τύπου J - K (J - K flip - flop)		<table><tr><th>J</th><th>K</th><th colspan="2">$Q_n \rightarrow Q_{n+1}$</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	J	K	$Q_n \rightarrow Q_{n+1}$		0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	0
J	K	$Q_n \rightarrow Q_{n+1}$																																				
0	0	0	0																																			
0	0	1	1																																			
0	1	0	0																																			
0	1	1	0																																			
1	0	0	1																																			
1	0	1	1																																			
1	1	0	1																																			
1	1	1	0																																			

* Άνεπίτρεπτες καταστάσεις.



Σχ. 4.4β.

Τό λογικό διάγραμμα ενός απλού καταχωρητή με πολυδονητές τύπου D.



Σχ. 4.4γ.

Τό λογικό διάγραμμα ενός όλισθητή.

Arithmetic and Logic Unit) του ηλεκτρονικού υπολογιστή, στην οποία πραγματοποιούνται όλες οι λογικές και αριθμητικές πράξεις.

Τό βασικό κύκλωμα ενός παραλλήλου συσσωρευτή φαίνεται στο σχήμα 4.4δ.

Έκτός από τα ακολουθιακά κυκλώματα που αναφέραμε, υπάρχουν και άλλα κυκλώματα, όπως είναι οι διάφοροι τύποι άριθμητών, τα οποία δέ θα αναπτυχθούν στο κεφάλαιο αυτό.

4.5 Έρωτήσεις.

1. Τι είναι τα διαγράμματα Venn;
2. Πώς αλλιώς ονομάζεται η λογική πράξη «H»; 'Η λογική πράξη «ΚΑΙ»; 'Η λογική πράξη «ΟΧΙ»;
3. Τι καλούμε θετική και τί άρνητική λογική;

ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ

ΔΟΜΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ ΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

5.1 Γενικά.

Όπως έχουμε αναφέρει, ο ηλεκτρονικός υπολογιστής αποτελεί ένα σύστημα επεξεργασίας πληροφοριών και λειτουργεί σύμφωνα με ένα καθορισμένο πρόγραμμα, τό οποίο καθορίζει τον τρόπο επεξεργασίας των πληροφοριών. Τό πρόγραμμα αποτελεί τό σύνολο των εντολών τίς οποίες δίνουμε στον ηλεκτρονικό υπολογιστή γιά νά εκτελέσει κάποια λειτουργία.

Ένας υπολογιστής αποτελείται γενικά από τά εξής τμήματα:

- Τή μονάδα έλέγχου (control unit).
- Τήν αριθμητική καί λογική μονάδα (arithmetic and logic unit - ALU).
- Τή μονάδα μνήμης (memory unit).
- Τή μονάδα εισόδου - εξόδου (Input - Output unit - I/O).

Η μονάδα έλέγχου καί ή αριθμητική μονάδα ονομάζονται μαζί συνήθως «κεντρική μονάδα επεξεργασίας» (Central Processing Unit - CPU).

Στό σχήμα 5.1 φαίνονται διαγραμματικά τά μέρη από τά όποια αποτελείται ό ηλεκτρονικός υπολογιστής.

Όπως φαίνεται από τό σχήμα υπάρχουν δύο είδη συνδέσεων μεταξύ των διαφόρων μονάδων του ηλεκτρονικού υπολογιστή.

- Οι συνδέσεις γιά τή διακίνηση πληροφοριών (διπλές γραμμές), καί
- οι συνδέσεις γιά τή διακίνηση σημάτων έλέγχου (μονές γραμμές).

Η λειτουργία κάθε μονάδας του ηλεκτρονικού υπολογιστού σέ γενικές γραμμές είναι ή εξής:

α) Μονάδα μνήμης.

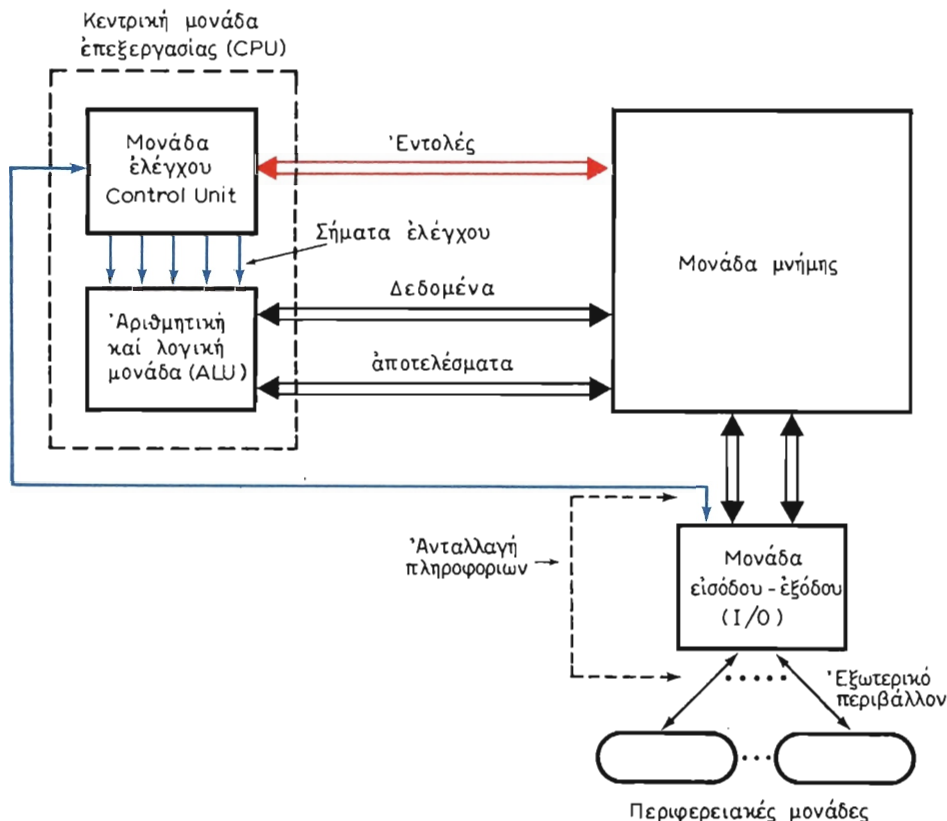
Χρησιμεύει γιά τήν αποθήκευση των αρχικών δεδομένων καί των μερικών αποτελεσμάτων πού προκύπτουν από τούς διάφορους υπολογισμούς καθώς καί των τελικών αποτελεσμάτων τά όποια πρόκειται νά δοθούν ως απάντηση τής έπιλύσεως των προβλημάτων. Στή μονάδα μνήμης έπίσης αποθηκεύονται καί οι έντολές του προγράμματος.

β) Αριθμητική - λογική μονάδα.

Έκτελεί τίς διάφορες βασικές αριθμητικές πράξεις (+, -, x, ÷), τίς λογικές πράξεις (OR, AND κλπ.) καί τίς διάφορες βοηθητικές εργασίες (όλίσθηση κλπ.).

γ) Μονάδα εισόδου - εξόδου (I/O).

Έπιτρέπει τήν επικοινωνία του υπολογιστή μέ τον άνθρωπο καί γενικά μέ τό πε-



Σχ. 5.1.

ριβάλλον. Μ' αυτή συνδέονται οι έκτυπωτές, οι όθόνες, τα τηλέτυπα και οι άλλες περιφερειακές μονάδες.

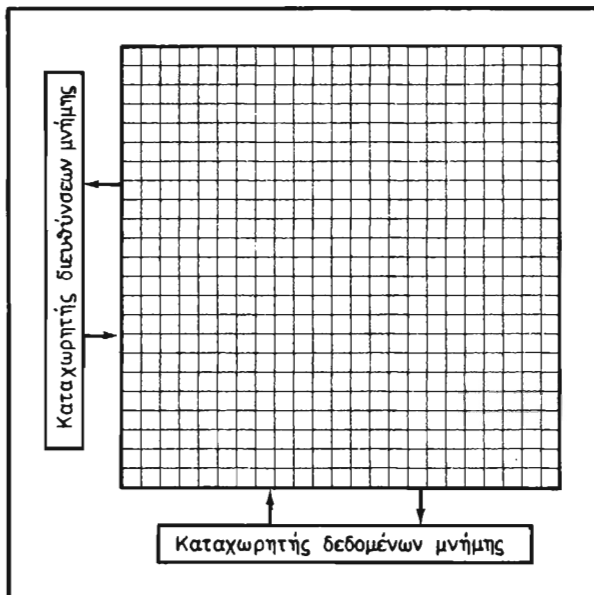
δ) Μονάδα έλέγχου.

Είναι ο γενικός ελεγκτής της όλης εργασίας του υπολογιστή. Αυτή κανονίζει τις χρονικές στιγμές που κάθε μονάδα θα συνδεθεί και θα λειτουργήσει σε συνδυασμό με τις άλλες, ώστε να έχομε τό επιθυμητό αποτέλεσμα.

5.2 Η μονάδα μνήμης.

Η μονάδα μνήμης, όπως είπαμε και προηγουμένως, είναι η μονάδα όπου αποθηκεύονται προσωρινά όλες οι πληροφορίες του υπολογιστή, οι οποίες περιλαμβάνουν τις εντολές και τα δεδομένα. Η μονάδα αυτή βοηθά στη διακίνηση των πληροφοριών, τόσο μέσα στον υπολογιστή, όσο και μεταξύ του εξωτερικού περιβάλλοντος και του υπολογιστή.

Επίσης, η μονάδα μνήμης συνδέεται (σχ. 5.1) με την Αριθμητική και Λογική Μονάδα, η οποία εκτελεί τις διάφορες πράξεις και με τη μονάδα έλέγχου η οποία κατευθύνει τη ροή των πληροφοριών μέσα στον υπολογιστή.



Σχ. 5.2.

Η μνήμη αποτελείται από ένα σύνολο «θέσεων μνήμης». Κάθε μία θέση μνήμης έχει τη δυνατότητα αποθηκεύσεως μιας λέξεως. Για κάθε τύπο ηλεκτρονικού υπολογιστή η λέξη ορίζεται ως ένα σύνολο δυαδικών ψηφίων (bits).

Τό πλήθος των δυαδικών ψηφίων που αποτελεί τη «λέξη» (word) του υπολογιστή λέγεται «μήκος λέξεως» (word length) του υπολογιστή. Η αποθηκευμένη λέξη μπορεί να παριστάνει έναν αριθμό, ένα σύνολο γραμμάτων ή μία έντολή.

Ο αριθμός που αντιστοιχεί, όταν αριθμήσουμε τις θέσεις μνήμης, σε κάθε θέση της μνήμης ονομάζεται «διεύθυνση» (address).

Τά σπουδαιότερα χαρακτηριστικά της μνήμης ενός ηλεκτρονικού υπολογιστή είναι τα εξής:

– **Χωρητικότητα.**

Δηλαδή τό πλήθος των λέξεων που μπορεί να αποθηκευθούν στη μνήμη.

– **Ταχύτητα.**

Δηλαδή ή ταχύτητα μέ την οποία μία πληροφορία μπορεί να αναγνωσθεί ή να εγγραφεί στη μνήμη καί ή συχνότητα μέ την οποία μπορούμε να ζητάμε τέτοιου είδους λειτουργίες από τη μνήμη.

– **Μονιμότητα αποθηκεύσεως.**

Δηλαδή πόσο χρονικό διάστημα μπορούν να διατηρηθούν οι πληροφορίες που έχουν αποθηκευθεί στη μνήμη, χωρίς να χρειασθεί να επαναποθηκευθούν.

Στό σχήμα 5.2 φαίνεται ή οργάνωση της μονάδας μνήμης που μοιάζει μέ μία μήτρα. Έκτός από «στοιχεία μνήμης» περιέχει καί δύο καταχωρητές: τόν «καταχωρητή διευθύνσεων μνήμης» καί τόν «καταχωρητή δεδομένων μνήμης». Ένα στοιχείο

μνήμης συνήθως, αποτελείται από ένα σύνολο ηλεκτρονικών στοιχείων μνήμης, τα οποία είναι τόσα όσο και τό μήκος της δυαδικής λέξεως του υπολογιστή.

Όταν μία λέξη πρόκειται να αποθηκευθεί στη μνήμη, γράφεται πρώτα στον καταχωρητή δεδομένων μνήμης. Η διεύθυνση στην οποία θα αποθηκευθεί γράφεται στον καταχωρητή διευθύνσεων μνήμης. Στη συνέχεια τό περιεχόμενο του καταχωρητή δεδομένων μεταφέρεται στη μνήμη, στη διεύθυνση, πού καθορίζει τό περιεχόμενο του καταχωρητή διευθύνσεων, ενώ τό προηγούμενο περιεχόμενο της διευθύνσεως μνήμης χάνεται. Η εργασία αυτή λέγεται «έγγραφή» μιᾶς λέξεως στη μνήμη. Η αντίστροφη εργασία λέγεται «ανάγνωση» μιᾶς λέξεως από τή μνήμη.

5.3 Ἀριθμητική καί λογική μονάδα (ALU).

Ἡ ἀριθμητική καί λογική μονάδα ἐκτελεῖ τίς ἀριθμητικές καί τίς λογικές πράξεις καθώς καί διάφορες ἄλλες βοηθητικές πράξεις, ὅπως πράξεις μετασχηματισμοῦ καί ἀνακατατάξεως λέξεων.

Οἱ παραπάνω πράξεις ἐκτελοῦνται διαδοχικῶς καί μέ βάση θεμελιώδεις λειτουργίες. Οἱ θεμελιώδεις αὐτές λειτουργίες εἶναι ἡ πρόσθεση, ἡ ὀλίσθηση καί οἱ βασικές λογικές πράξεις ΚΑΙ, Ή καί ΟΧΙ. Ἐτσι, π.χ. ὁ πολλαπλασιασμός δύο ἀριθμῶν ἐκτελεῖται μέ διαδοχικές ὀλισθήσεις, συγκρίσεις καί προσθέσεις, ἐνῶ ἡ ἀφαίρεση ἑνός ἀριθμοῦ ἀπό ἕναν ἄλλο ἐκτελεῖται μέ τό συμπλήρωμα τοῦ δευτέρου καί μέ πρόσθεση. Γι' αὐτό τό λόγο στήν Ἀριθμητική καί Λογική μονάδα ὑπάρχουν διάφορες ὑπομονάδες μέ τήν βοήθεια τῶν ὁποίων εἶναι δυνατή ἡ ἐκτέλεση τῶν θεμελιωδῶν λειτουργιῶν.

Στό σχῆμα 5.3 φαίνεται ἡ ἀριθμητική καί λογική μονάδα ἑνός υπολογιστή. Ἡ μονάδα ἀποτελεῖται ἀπό δύο συσσωρευτές Α καί Μ καί ἀπό τά διάφορα ἀπαιτούμενα λογικά κυκλώματα.

Ὁ συσσωρευτής εἶναι ἕνας καταχωρητής ὁ ὁποῖος προσθέτει ἀριθμούς πού τοῦ δίνονται διαδοχικά. Τό ἐκάστοτε ἄθροισμα τῶν ἀριθμῶν συγκρατεῖται ἐπίσης ἀπό τό συσσωρευτή. Ἡ ὀλίσθηση γίνεται μέσω ὀλισθητῶν οἱ ὁποῖοι εἶναι καταχωρητές πού μποροῦν νά ὀλισθαίνουν τό περιεχόμενό τους κατά μία θέση πρὸς τά δεξιά ἢ πρὸς τά ἀριστερά.

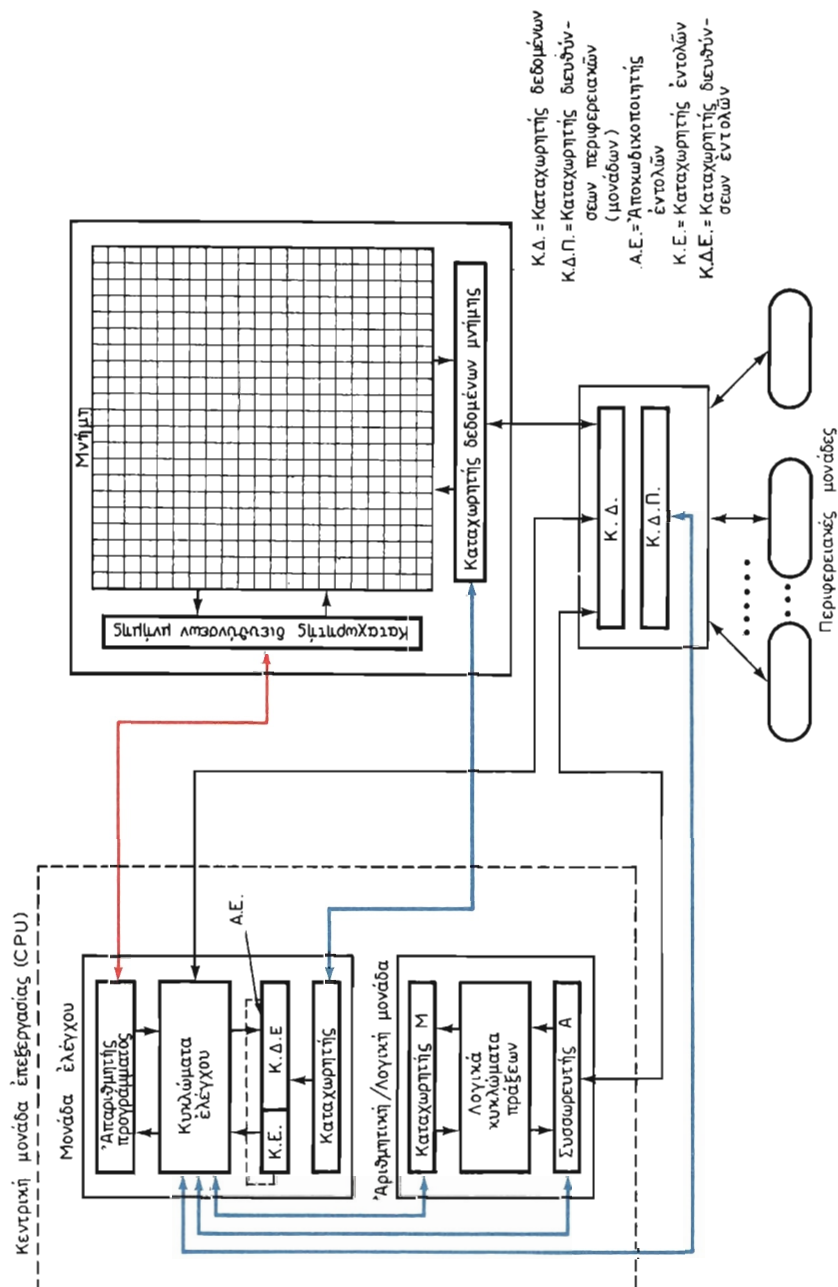
Τέλος οἱ διάφορες λογικές πράξεις ἐκτελοῦνται μέ καταχώρηση τῶν ἀριθμῶν σέ ἀντίστοιχους καταχωρητές οἱ ὁποῖοι ἐπικοινωνοῦν μεταξύ τους μέσω λογικῶν κυκλωμάτων. Ἡ ἔξοδος αὐτῶν τῶν κυκλωμάτων δίνει τό ἀποτέλεσμα.

Γενικά μποροῦμε νά ποῦμε ὅτι ἡ ἀριθμητική καί λογική μονάδα ἀποτελεῖται ἀπό ἕνα σύνολο καταχωρητῶν πού ἐπικοινωνοῦν μεταξύ τους μέσω ἐλεγχόμενων ὁδῶν. Ἐνας ἢ περισσότεροι ἀπό τοὺς καταχωρητές αὐτοὺς μποροῦν νά δεχθοῦν δεδομένα ἀπό τόν καταχωρητή δεδομένων μνήμης καί νά δώσουν σ' αὐτόν ἀριθμούς οἱ ὁποῖοι εἶναι τό ἀποτέλεσμα πράξεων.

Ὁ τρόπος ἐπικοινωνίας τῶν καταχωρητῶν μεταξύ τους, καθώς καί ἡ ἐπικοινωνία τους μέ τόν καταχωρητή δεδομένων μνήμης, ἐλέγχεται καί καθορίζεται ἀπό τήν μονάδα ἐλέγχου τοῦ υπολογιστή.

Σύμφωνα μέ τό σχῆμα 5.3, ἡ μονάδα ἐλέγχου γιά τήν ἐκτέλεση μιᾶς προσθέσεως θά δώσει τίς παρακάτω ἐντολές.

- Μεταφορά τοῦ πρώτου ἀριθμοῦ ἀπό τήν ἀντίστοιχη θέση μνήμης στό συσσωρευτή Μ.



Σχ. 5.3.

Σχηματική παράσταση των μονάδων ενός ηλεκτρονικού υπολογιστή και τα διάφορα βασικά λειτουργικά στοιχεία κάθε μονάδας.

- Μεταφορά του ίδιου αριθμού από τό συσσωρευτή Μ στον Α.
- Μεταφορά του δεύτερου αριθμού από τή μνήμη στο συσσωρευτή Μ.
- Μεταφορά του δεύτερου αριθμού από τό συσσωρευτή Μ καί πρόσθεση μέ τόν αριθμό πού περιέχεται στο συσσωρευτή Α. Τό άθροισμα παραμένει στο συσσωρευτή Α.
- Μεταφορά νέου αριθμού από τή μνήμη στο συσσωρευτή Μ.

5.4 Μονάδα έλέγχου.

Ή μονάδα έλέγχου συντονίζει τή διαδοχή τών πράξεων (έντολών) καί γενικά τή λειτουργία τών ύπολοίπων μονάδων του ύπολογιστή.

Κατά τήν εκτέλεση ενός προγράμματος, ή μονάδα έλέγχου παίρνει από τή μνήμη τίς έντολές του προγράμματος οι οποίες είχαν αποθηκευθεί προηγουμένως σ' αυτήν. Χρησιμοποιώντας πλέον τίς έντολές αυτές ή μονάδα έλέγχου δίνει σήματα πρós:

- Τή μνήμη, γιά νά έπιτρέψει τήν έξαγωγή ή τή λήψη πληροφοριών πρós ή από τίς άλλες μονάδες του ύπολογιστή.
- Τήν αριθμητική καί λογική μονάδα, γιά τήν εκτέλεση τών επιθυμητών πράξεων καί
- πρós τή μονάδα εισόδου - έξόδου, γιά τήν επικοινωνία του ύπολογιστή μέ τό έξωτερικό περιβάλλον.

Στή μονάδα έλέγχου περιλαμβάνονται:

α) 'Ο καταχωρητής έντολών (ΚΕ).

Όπως είπαμε καί προηγουμένως, στή μνήμη του ύπολογιστή αποθηκεύονται δεδομένα καί έντολές. Μέ τίς έντολές καθορίζεται ό τρόπος επεξεργασίας τών δεδομένων, καθώς επίσης καί οι θέσεις μνήμης πού βρίσκονται τά δεδομένα αυτά. Στόν καταχωρητή έντολών καταχωρείται τό τμήμα εκείνο τής έντολής πού καθορίζει τόν τρόπο επεξεργασίας π.χ. αριθμητική πράξη.

β) 'Ο καταχωρητής διευθύνσεων έντολών (ΚΔΕ).

Σ' αυτόν καταχωρείται ή διεύθυνση τής έντολής πού πρόκειται νά εκτελεσθεΐ. Συνήθως ό καταχωρητής αυτός είναι ένας άπαριθμητής, ό όποιος όταν αύξηθεΐ τό περιεχόμενό του κατά 1, δείχνει τή διεύθυνση τής επομένης έντολής, ή όποία καί μεταφέρεται στόν καταχωρητή διευθύνσεων μνήμης γιά νά ανακληθεΐ ή επόμενη έντολή. Ύπάρχουν όμως περιπτώσεις πού ή επόμενη έντολή έχει διαφορετική διεύθυνση από εκείνη πού ακολουθεΐ τή διεύθυνση τής εκτελούμενης έντολής. Αυτό συμβαίνει όταν εκτελούνται έντολές έλέγχου, οι οποίες καθορίζουν κάποιο άλμα, τό όποίο συνήθως έξαρτάται από τό αποτέλεσμα κάποιας συγκρίσεως. Σ' αυτές τίς περιπτώσεις τό περιεχόμενο του καταχωρητή διευθύνσεων έντολών αύξάνεται ή έλαττώνεται όσο καθορίζει ή έντολή έλέγχου, δηλαδή όσο είναι τό απαιτούμενο άλμα.

γ) 'Ο αποκωδικοποιητής έντολών (ΑΕ).

Αυτός είναι ένα λογικό κύκλωμα τό όποίο αποκωδικοποιεί τόν κωδικό έντολής.

Κάθε έντολή ενεργοποιεί μία ή περισσότερες εξόδους του αποκωδικοποιητή. Μέ αυτόν τον τρόπο καθορίζονται τα κυκλώματα έλέγχου τα όποια θα πρέπει να ενεργοποιηθούν για να εκτελεσθεί η έντολή.

δ) Οι καταχωρητές δεικτών.

Τό περιεχόμενο των καταχωρητών δεικτών καθορίζει μία μεταβολή διευθύνσεως. 'Η διεύθυνση αυτή είναι ή διεύθυνση κάποιου δεδομένου στο όποιο αναφέρεται κάποια έντολή. Έτσι ή μεταβολή διευθύνσεως είναι μία πρόσθεση ή αφαίρεση του περιεχομένου ενός καταχωρητή δεικτών μέ την αντίστοιχη διεύθυνση.

ε) Τα κυκλώματα χρονισμού.

Γιά να εκτελεσθεί μία έντολή, απαιτείται ή μετακίνηση λέξεων από τον έναν καταχωρητή στον άλλο. Κάθε μετακίνηση πληροφορίας πρέπει να πραγματοποιείται μόνο όταν είναι βέβαιο ότι τα κυκλώματα που την περιέχουν βρίσκονται σε κατάσταση Ισορροπίας. 'Η διαδοχή των λειτουργιών που απαιτούνται για την εκτέλεση κάθε μιάς έντολης καί ή καθορισμός της χρονικής στιγμής που θα εκτελεσθεί κάποια έντολή καθορίζεται από είδικά κυκλώματα χρονισμού τα όποια δίνουν παλμούς ενεργοποίησης σε προκαθορισμένες στιγμές.

Κάθε τέτοιος παλμός προκαλεί την εκτέλεση μιάς στοιχειώδους λειτουργίας. Τα κυκλώματα χρονισμού ενεργοποιούνται από μία κεντρική πηγή ώρολογιακών παλμών, τό «ρολόι» (Clock). Περιλαμβάνουν, επίσης άπαριθμητές για τον ύποβιβασμό της συχνότητας του ρολογιού.

στ) Τα κυκλώματα έλέγχου.

Είναι συνήθως άπαριθμητές καί συγκριτές μέσω των οποίων παρακολουθείται ό άριθμός των στοιχειωδών βημάτων για να καθορισθεί τό τέλος διαφόρων λειτουργιών.

5.5 'Η μονάδα Εισόδου - 'Εξόδου (I/O).

'Η μονάδα εισόδου - έξόδου δέχεται τίς πληροφορίες που δίνονται από τον άνθρωπο μέσω των περιφερειακών μονάδων. Μετά την επεξεργασία τους από τον ύπολογιστή τίς μεταφέρει πρós τό έξωτερικό περιβάλλον.

'Η είσοδος των πληροφοριών πρós τον ύπολογιστή μπορεί να γίνει μέσω ενός πληκτρολογίου, διατηρήτων καρτών ή μέσω συσκευών μετρήσεως (π.χ. θερμοκρασίας, πιέσεως κλπ.). Ένώ ή έξοδος των πληροφοριών πρós τό έξωτερικό περιβάλλον μπορεί να γίνει μέσω ενός εκτυπωτή, μιάς όθόνης καθοδικού σωλήνα (CRT) ή μιάς άλλης περιφερειακής μονάδας καταγραφής πληροφοριών.

'Η μονάδα εισόδου - έξόδου περιλαμβάνει διατάξεις που ρυθμίζουν την εισαγωγή καί εξαγωγή πληροφοριών καθώς καί τή λειτουργία των περιφερειακών μονάδων. Οι διατάξεις αυτές καλούνται **διώρυγες** (channels). Οι περιφερειακές μονάδες χρησιμοποιούνται για την εισαγωγή καί λήψη πληροφοριών από τον άνθρωπο, καθώς καί για την αποθήκευση ένδιαμέσων αποτελεσμάτων.

'Η μονάδα εισόδου - έξόδου περιέχει έναν «καταχωρητή δεδομένων» (ΚΔ) (data register) που δέχεται τίς λέξεις που θα μεταφερθούν από τίς περιφερειακές μο-

νάδες προς τόν υπολογιστή ή αντίστροφα και έναν «καταχωρητή διευθύνσεως περιφερειακών» (peripheral address register) (ΚΔΠ). Αυτός περιέχει τη διεύθυνση του περιφερειακού από τό οποίο ή προς τό οποίο πρόκειται νά γίνει ή μεταφορά μιās λέξεως. Έτσι κάθε στιγμή ο υπολογιστής γνωρίζει μέ ποιό περιφερειακό τερματικό (όθόνη, έκτυπωτής κλπ.) «συνομιλεί».

Γιά νά μπορέσουμε νά συμβιβάσουμε τήν μεγάλη διαφορά ταχύτητας μεταξύ τής κεντρικής μνήμης και τής μνήμης πού βρίσκεται στίς περιφερειακές μονάδες (βοηθητική μνήμη), χρησιμοποιούμε ένα ενδιάμεσο στάδιο «απομονώσεως» (buffer) στήν είσοδο (έξοδο) του υπολογιστή. Έτσι στήν περίπτωση εισόδου πληροφοριών από τή βοηθητική μνήμη προς τόν υπολογιστή, οι πληροφορίες εισέρχονται και αποθηκεύονται στό στάδιο απομονώσεως μέ σχετικά μικρή ταχύτητα, ενώ ταυτόχρονα ο υπολογιστής εκτελεί άλλες λειτουργίες. Μόλις θελήσει, τά αποθηκευμένα αυτά στοιχεία, μεταφέρονται στήν κεντρική μνήμη μέ ταχύτητα υπολογιστού, ή οποία είναι τουλάχιστον κατά 1000 φορές μεγαλύτερη από τήν ταχύτητα τής βοηθητικής μνήμης. Τό αντίθετο συμβαίνει όταν ο υπολογιστής στέλνει πληροφορίες γιά αποθήκευση στή βοηθητική μνήμη. Οι πληροφορίες μεταφέρονται από τήν κεντρική μνήμη στή μονάδα απομονώσεως μέ ταχύτητα υπολογιστού και έν συνεχεία, τήν ώρα πού ο υπολογιστής συνεχίζει τήν λειτουργία του μέ άλλες εργασίες, οι πληροφορίες αυτές μεταφέρονται γιά αποθήκευση μέ χαμηλή ταχύτητα στή βοηθητική μνήμη.

Η συνεργασία κάθε περιφερειακού μέ τή μονάδα εισόδου - έξόδου του υπολογιστή γίνεται μέσω μιās ειδικής μονάδας ή οποία καλείται «μονάδα έλέγχου» (control unit) του περιφερειακού. Η μονάδα αυτή πληροφορεί τόν υπολογιστή γιά τήν κατάσταση (status) του περιφερειακού. Τόν πληροφορεί π.χ. αν τό περιφερειακό είναι ελεύθερο ή αν έχει βλάβη και έτσι μέ τά σήματα πού δέχεται από τόν υπολογιστή, οδηγεί τό περιφερειακό νά δεχθεί ή νά δώσει πληροφορίες στόν υπολογιστή.

5.6 Διαδικασία εκτελέσεως μιās έντολης. Αρχή τών δύο φάσεων.

Όπως έχομε πεί μέχρι τώρα, στή μνήμη εκτός από τά δεδομένα και τά μερικά ή τελικά αποτελέσματα πού προκύπτουν από τίς αριθμητικές ή λογικές πράξεις τής αριθμητικής και λογικής μονάδας, αποθηκεύονται και οι έντολές του προγράμματος, κατάλληλα κωδικοποιημένες. Οι έντολές αποθηκεύονται σέ διαδοχικές θέσεις μνήμης κατά τή σειρά εκτελέσεώς τους.

Κάθε έντολή εκτελείται σέ δύο φάσεις:

- Τή φάση τής ανακλήσεως (fetch phase).
- Τή φάση τής εκτελέσεως (execution phase).

Όπως είπαμε προηγουμένως, στό μετρητή προγράμματος βρίσκεται ή διεύθυνση τής νέας έντολης. Έτσι κατά τή φάση τής ανακλήσεως, ή διεύθυνση αυτή μεταφέρεται στόν καταχωρητή διευθύνσεων μνήμης και μέ τή βοήθεια τής διευθύνσεως αναζητείται στή μνήμη ή έντολή, βρίσκεται και μεταφέρεται στόν καταχωρητή δεδομένων μνήμης. Τό περιεχόμενο αυτού μεταφέρεται στή μονάδα έλέγχου και αποθηκεύεται στόν καταχωρητή έντολών. Ο καταχωρητής πού βρίσκεται πριν από τόν καταχωρητή έντολών, χρησιμεύει ως αποθήκη τής έντολης μέχρις ότου ή έντολή αυτή μεταφερθεί στόν καταχωρητή έντολών, όταν ο καταχωρητής περιέχει

κάποια άλλη έντολή πού δέν έχει εκτελεσθεῖ ἀκόμη. Ὄταν ἡ έντολή φτάσει στόν καταχωρητή έντολῶν, τότε τό περιεχόμενο τοῦ ἀπαριθμητῆ προγράμματος αὐξάνεται κατά μία μονάδα καί περιέχει τή διευθύνση τῆς ἐπομένης έντολῆς.

Κατά τή φάση τῆς ἐκτελέσεως τώρα, ἀποκωδικοποιεῖται τό περιεχόμενο τοῦ καταχωρητῆ έντολῶν. Πρῶτα ἀποκωδικοποιεῖται ὁ κωδικός έντολῆς καί ἡ έντολή ἀναλύεται σέ στοιχειώδη βήματα (μικροέντολές) γιά τά ὁποῖα ἡ μονάδα ἐλέγχου στέλνει τά ἀπαραίτητα σήματα πρὸς τίς ὑπόλοιπες μονάδες τά ὁποῖα προειδοποιοῦν γιά τήν ἐκτέλεση τῶν μικροέντολῶν.

Μόλις αὐτές ἐκτελεσθοῦν, ἀρχίζει ἡ νέα φάση ἀνακλήσεως κ.ο.κ. μέχρις ὅτου περατωθεῖ ὅλο τό ὑπὸ ἐκτέλεση πρόγραμμα. Οἱ κύκλοι λοιπόν τῶν δύο φάσεων θά ἐπαναλαμβάνονται συνεχῶς μέχρι νά ἐμφανισθεῖ καί νά ἐκτελεσθεῖ μία έντολή STOP ἢ ἄλλη εἰδική έντολή μέ τήν ὁποία διακόπτεται ἡ λειτουργία τοῦ προγράμματος. Ἄν δέν ὑπάρχει ἄλλο πρόγραμμα γιά νά ἐκτελεσθεῖ, διακόπτεται καί ἡ λειτουργία τοῦ ὑπολογιστῆ.

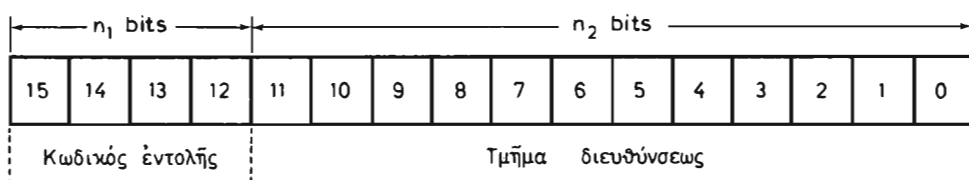
5.7 Λειτουργία τοῦ ἡλεκτρονικοῦ ὑπολογιστῆ.

Ἡ αὐτόματη ἐκτέλεση τοῦ προγράμματος ἐπιτυγχάνεται μέ τήν ἐπαναληπτική χρησιμοποίηση ὀρισμένων βασικῶν λειτουργιῶν τοῦ ὑπολογιστῆ, ἀπό τίς ὁποῖες οἱ σπουδαιότερες εἶναι οἱ ἑξῆς:

- Ἀνάκληση ἀπὸ τή μνήμη τῆς ἐπομένης έντολῆς.
- Ἀποκωδικοποίηση τῆς έντολῆς.
- Ἀνάκληση τῶν δεδομένων ἀπὸ τή μνήμη, ἂν ἀπαιτεῖται.
- Ἐκτέλεση τῆς ἀποκωδικοποιημένης έντολῆς.

Κάθε έντολή παριστάνεται συνήθως μέ μία λέξη τοῦ ὑπολογιστῆ. Ὑπάρχουν ὅμως καί ὑπολογιστές στούς ὁποίους μία έντολή παριστάνεται μέ δύο ἢ περισσότερες λέξεις ἢ καί ἀντίστροφα, μία λέξη δηλαδή μπορεῖ νά παριστάνει πολλές έντολές.

Στὴν περίπτωση πού χρησιμοποιεῖται μία λέξη γιά κάθε έντολή, τότε ἡ λέξη αὐτῇ χωρίζεται σέ δύο τμήματα (σχ. 5.7α). Τό πρῶτο τμήμα ὀνομάζεται «κωδικός έντολῆς» καί καταλαμβάνει τά πρῶτα n_1 σημαντικά ψηφία (bits).



Σχ. 5.7α.

Σχηματική παράσταση μιᾶς έντολῆς Η/Υ ἡ ὁποία ἀποτελεῖται ἀπὸ μία λέξη μήκους 16 δυαδικῶν ψηφίων (1 λέξη = 16 Bits).

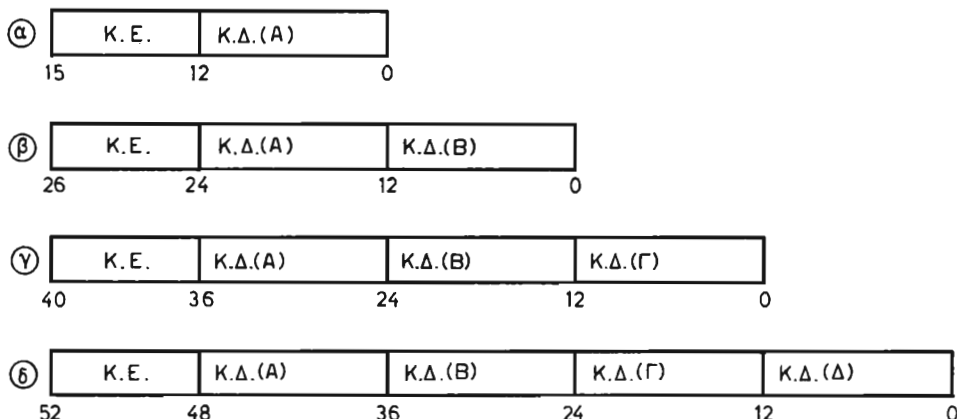
Τό δεύτερο, πού ὀνομάζεται «τμήμα διευθύνσεως», μᾶς ὑποδεικνύει σέ ποιά διευθύνση μνήμης βρίσκονται τά δεδομένα γιά νά ἐκτελεσθεῖ ἡ έντολή πού ὑποδεικνύεται μέ τόν κωδικό έντολῆς.

Ὁ κωδικός έντολῆς εἶναι ἓνας δυαδικὸς ἀριθμὸς μήκους n_1 ψηφίων. Ἐπομένως

ο μέγιστος αριθμός εντολών που μπορεί να έχει ένας υπολογιστής με κωδικό έντολης n , ψηφίων είναι 2^n . Το σύνολο των εντολών ενός υπολογιστή λέγεται «**ρεπερτόριο εντολών**».

Για τον υπολογιστή του οποίου η λέξη είναι όπως στο σχήμα 5.7α, το ρεπερτόριο των εντολών δεν μπορεί να περιλαμβάνει πάνω από $2^4 = 16$ εντολές.

Στήν γενική τους μορφή οι εντολές χωρίζονται σε τέσσερις κατηγορίες: στις εντολές μιάς, δύο, τριών ή τεσσάρων διευθύνσεων, όπως φαίνεται στο σχήμα 5.7β.



Σχ. 5.7β.

Σχηματική παράσταση εντολών α) μιάς β) δύο γ) τριών δ) τεσσάρων διευθύνσεων
(ΚΕ = κωδικός έντολης· ΚΔ = Κωδικός διευθύνσεων).

Στήν εντολή του σχήματος 5.7β(β), αν ο κωδικός έντολης είναι η λογική πράξη «ΚΑΙ», σημαίνει ότι πρέπει να εκτελεσθεί η πράξη ΚΑΙ με τα δεδομένα που βρίσκονται στις θέσεις μνήμης Α και Β και ότι το αποτέλεσμα θα αποθηκευθεί στη θέση μνήμης Α.

Η εντολή του σχήματος 5.7β(γ) προσδιορίζει ακόμα τη θέση μνήμης (Γ) που θα αποθηκευθεί το αποτέλεσμα αντί για τη θέση Α, ενώ στην εντολή του σχήματος 5.7β(δ) το τμήμα Δ μας δίνει τη διεύθυνση της τελικής θέσεως της μνήμης.

Οι εντολές του υπολογιστή, χωρίζονται σε ομάδες, ανάλογα με την εργασία τους και τον τρόπο εκτελέσεώς τους:

Έτσι έχουμε:

- Έντολές αριθμητικών πράξεων.
- Έντολές λογικών πράξεων.
- Έντολές αναφοράς στη μνήμη.
- Έντολές άλματος.
- Έντολές όλισθήσεως.
- Έντολές εισόδου - εξόδου.
- Έντολές καταχωρητή δείκτη.
- Ειδικές εντολές.

1. Έντολές αριθμητικῶν πράξεων.

Όπως π.χ. πρόσθεση, αφαίρεση, πολλαπλασιασμός καί διαίρεση. Μέ τις έντολές αυτές καθορίζεται καί τό χρησιμοποιούμενο αριθμητικό σύστημα (π.χ. δυαδικό, δεκαδικό κλπ.).

2. Έντολές λογικῶν πράξεων.

Όπως π.χ. γιά τις λογικές πράξεις «ΚΑΙ», «Ή», «ΟΧΙ ΚΑΙ» κλπ. μεταξύ δύο μεγεθῶν.

3. Έντολές ἀναφορᾶς στή μνήμη.

Αὐτές πού χρησιμοποιοῦνται γιά τήν ἀνάκληση ἀπό τή μνήμη ἢ τήν τοποθέτηση στήν μνήμη τῶν διαφόρων δεδομένων ἢ έντολῶν τοῦ προγράμματος.

4. Έντολές ἄλματος.

Αὐτές πού χρησιμοποιοῦνται γιά τή διακοπή τῆς διαδοχικῆς ἐκτελέσεως τῶν έντολῶν τοῦ προγράμματος. Ὑπάρχουν δύο εἶδη έντολῶν ἄλματος: ἡ έντολή ἄλματος «χωρίς συνθήκη» καί ἡ έντολή ἄλματος «μέ συνθήκη». Κατά τήν ἐκτέλεση τῆς έντολῆς ἄλματος, τό περιεχόμενο τοῦ κωδικοῦ διευθύνσεως τοῦ καταχωρητῆ έντολῶν μεταφέρεται καί γράφεται στό μετρητή προγράμματος. Ἐτσι στήν ἐπόμενη φάση ἀνακλήσεως θά γίνει ἀνάκληση ἀπό τή μνήμη τῆς έντολῆς, τῆς ὁποίας τή διεύθυνση δείχνει ὁ μετρητής προγράμματος, δηλαδή τή διεύθυνση τῆς έντολῆς ἄλματος. Ἐντολή ἄλματος χωρίς συνθήκη σημαίνει ὅτι ὅπωςδήποτε διακόπτομε τήν ἐκτέλεση τῆς ἐπόμενης σέ σειρά έντολῆς τοῦ προγράμματος καί ἐκτελοῦμε κάποια ἄλλη. Ἐνῶ έντολή ἄλματος μέ συνθήκη, σημαίνει ὅτι πρῶτα ἐλέγχομε κάποια κατάσταση καί ἂν αὐτή δέν εἶναι ἡ ἐπιθυμητή, τότε ἐκτελοῦμε έντολή ἄλματος. Διαφορετικά ὁ ὑπολογιστής συνεχίζει τή διαδοχική ἐκτέλεση τοῦ προγράμματος.

5. Έντολές ὀλισθήσεως.

Αὐτές πού ἀναγκάζουν τό περιεχόμενο τοῦ συσσωρευτῆ Α νά ὀλισθήσει πρὸς τά ἀριστερά ἢ πρὸς τά δεξιά κατά Ν θέσεις. Ἐτσι λοιπόν ἔχομε δύο εἰδῶν έντολές ὀλισθήσεως: τήν «έντολή ὀλισθήσεως πρὸς τά ἀριστερά» καί τήν «έντολή ὀλισθήσεως πρὸς τά δεξιά». Τό τμήμα διευθύνσεως τῆς έντολῆς ὀλισθήσεως δέν περιέχει μία διεύθυνση ἀλλά τόν ἀριθμό Ν τῶν ὀλισθήσεων πού θά πραγματοποιηθοῦν.

6. Έντολές εἰσόδου - ἐξόδου.

Αὐτές πού χρησιμοποιοῦνται γιά τήν ἐπικοινωνία τοῦ ἀνθρώπου καί τοῦ ηλεκτρονικοῦ ὑπολογιστῆ μέσω τῶν περιφερειακῶν μονάδων. Ὑπάρχουν δύο εἰδῶν έντολές: ἡ «έντολή ἐξόδου» καί ἡ «έντολή εἰσόδου». Μέ τήν έντολή ἐξόδου τό περιεχόμενο τοῦ συσσωρευτῆ μεταφέρεται στόν καταχωρητῆ δεδομένων τῆς μονάδας εἰσόδου - ἐξόδου ὅπου μέ τή βοήθεια τοῦ κωδικοποιητῆ τῆς μονάδας εἰσόδου - ἐξόδου, μετατρέπεται σέ ἀριθμό ἀναγνωρίσιμο ἀπό τήν περιφερειακή μονάδα. Ἡ έντολή εἰσόδου ἔχει ὡς ἀποτέλεσμα τήν εἰσαγωγή μιᾶς πληροφορίας ἀπό τήν περιφερειακή μονάδα πρὸς τόν ὑπολογιστή. Στήν πραγματικότητα ὁ ὑπολογιστής περιμένει τό χειριστή τῆς περιφερειακῆς μονάδας νά πατήσει π.χ. ἓνα πληκτρο πού ἀντιστοιχεῖ σέ κάποιο χαρακτήρα. Ὁ χαρακτήρας ἀναγνωρίζεται ἀπό τήν μονάδα

είσοδου - έξόδου, κωδικοποιείται και εγγράφεται στον καταχωρητή δεδομένων της μονάδας και στη συνέχεια μεταφέρεται στο συσσωρευτή Α.

Όπως και στις προηγούμενες εντολές, ο κωδικός διευθύνσεως της εντολής δέ δείχνει καμιά διεύθυνση. Αντίθετα τό περιεχόμενό του καθορίζει άλλα στοιχεία, όπως π.χ. αν ο χαρακτήρας πού περιέχει είναι οκταδικός ή ASCII. Επίσης πληροφορεί γιά τό ποιά περιφερειακή μονάδα θά λάβει ή έστειλε τήν πληροφορία αυτή.

7. Έντολές καταχωρητή δείκτη.

Είναι οι εντολές πού χρησιμοποιούνται γιά νά γνωρίζομε π.χ. πόσες φορές έχουμε επαναλάβει μία ίδια διαδικασία. Γενικώς οι ηλεκτρονικοί υπολογιστές διαθέτουν έναν αριθμό καταχωρητών πού ονομάζονται «καταχωρητές δείκτη» και βρίσκονται όπως είπαμε στη μονάδα έλέγχου. Οι καταχωρητές αυτοί είναι ιδιαίτερα χρήσιμοι κατά τή διάρκεια τής εκτέλεσεως τού προγράμματος. Υπάρχουν εντολές «αναγνώσεως» και «άποθηκεύσεως» τών καταχωρητών δεικτών καθώς και εντολές «δείκτη άλματος».

8. Ειδικές εντολές.

Χρησιμοποιούνται γιά νά βοηθήσουν στήν άποτελεσματικότερη εκτέλεση τού προγράμματος. Μιά τέτοια εντολή είναι ή εντολή μεταφοράς μιās ομάδας δεδομένων από ένα τμήμα τής μνήμης σέ άλλο.

5.8 Έρωτήσεις.

1. Ποιά τά βασικά δομικά στοιχεία ενός Η/Υ και σέ τί χρησιμοποιείται τό καθένα από αυτά;
2. Τι καλείται μήκος λέξεως;
3. Ποία τά κυριότερα χαρακτηριστικά τής μνήμης Η/Υ.
4. Ποιά ή διεργασία άποθηκεύσεως μιās λέξεως στη μνήμη;
5. Ποιές εργασίες εκτελεί ή αριθμητική και λογική μονάδα τού υπολογιστή;
6. Ποιές εργασίες εκτελεί ή μονάδα έλέγχου τού υπολογιστή;
7. Τι είναι ο καταχωρητής και τί ο άποκωδικοποιητής εντολών;
8. Ποιές εργασίες εκτελεί ή μονάδα εισόδου - έξόδου;
9. Ποιά ή διαδικασία εκτέλεσεως μιās εντολής;
10. Πώς μπορούμε νά παραστήσομε μία εντολή τριών διευθύνσεων μέ μήκος 18 δυαδικών ψηφίων και μήκος λέξεως 8 δυαδικών ψηφίων;
11. Αναφέρετε τούς διάφορους τύπους εντολών ενός Η/Υ ανάλογα μέ τήν εργασία πού εκτελείται και ανάλογα μέ τόν τρόπο εκτελέσεώς τους.
12. Ποιοί οι παράγοντες πού επηρεάζουν τήν εκλογή τού μήκους λέξεως ενός Η/Υ;

ΚΕΦΑΛΑΙΟ ΕΚΤΟ

ΣΥΣΤΗΜΑΤΑ ΑΠΟΘΗΚΕΥΣΕΩΣ ΠΛΗΡΟΦΟΡΙΩΝ

6.1 Γενικά.

Μία από τις πιά βασικές λειτουργίες ενός ηλεκτρονικού υπολογιστή είναι ή δυνατότητα αποθηκεύσεως καί ανακλήσεως πληροφοριών. Τά συστήματα του υπολογιστή πού έχουν τή δυνατότητα αύτή είναι ή κεντρική μνήμη καί ή βοηθητική (Auxilliary) ή μαζική (Mass) μνήμη, ή όποία βρίσκεται σέ «περιφερειακές μονάδες μνήμης» διαφόρων τύπων.

Ή κεντρική μνήμη του υπολογιστή αποτελείται από ένα σύνολο στοιχείων μνήμης τά όποια μās παρέχουν τή δυνατότητα εύκολης προσπελάσεως καί τά όποια, όπως είπαμε στήν παράγραφο 5.2, είναι συνήθως διατεταγμένα σέ σχήμα μήτρας.

Ή κεντρική μνήμη του υπολογιστή είναι ένα από τά άκριβότερα τμήματά του, γί' αύτό χρησιμοποιείται γιά τήν αποθήκευση πληροφοριών όταν θέλομε νά έχουμε μεγάλη ταχύτητα έπεξεργασίας.

Ένα από τά κυριότερα χαρακτηριστικά στοιχεία ενός συστήματος αποθηκεύσεως πληροφοριών είναι ό χρόνος προσπελάσεως (access time) δηλαδή ό χρόνος πού απαιτείται γιά νά τοποθετηθεί ή νά διαβαστεί μία πληροφορία στήν ή από τή μνήμη, από τή στιγμή πού θά δοθεί ή διεύθυνση τής πληροφορίας στή μνήμη.

Οι περιφερειακές μονάδες μνήμης χρησιμοποιούνται όταν δέν απαιτείται ύψηλή ταχύτητα προσπελάσεως καί ως μέσα εισαγωγής ή εξαγωγής πληροφοριών από τόν «έξωτερικό κόσμο» στόν υπολογιστή. Έχουν τή δυνατότητα αποθηκεύσεως μεγάλου όγκου πληροφοριών, συνήθως πολύ μεγαλύτερου από ό,τι ή κεντρική μνήμη του υπολογιστή.

Τό κόστος τών μονάδων αυτών ανά μονάδα αποθηκευομένης πληροφορίας είναι πολύ χαμηλότερο από τό αντίστοιχο τής κεντρικής μνήμης. Τέτοια συστήματα μνήμης είναι κυρίως οι μονάδες μαγνητικών δίσκων, οι μονάδες μαγνητικών ταινιών, οι μονάδες μαγνητικών τυμπάνων, οι μονάδες εύκάμπτων μαγνητικών δισκετών, κλπ.

Μπορούμε ακόμη νά θεωρήσουμε ως περιφερειακές μονάδες μνήμης τίσ μονάδες αναγνώσεως- διατρήσεως χαρτοταινιών καί δελτίων.

6.2 Κεντρική Μνήμη — Βασικά χαρακτηριστικά.

Ένα από τά κυριότερα προβλήματα σέ μία κεντρική μνήμη είναι ό χρόνος πού απαιτείται γιά τήν προσπέλαση σ' αύτή γιατί ό χρόνος αυτός καθορίζει στήν ούσία καί τήν όλη ταχύτητα ενός ηλεκτρονικού υπολογιστή.

Τά βασικά χαρακτηριστικά της μνήμης ενός ηλεκτρονικού υπολογιστή είναι:

α) Τό μέγεθος μνήμης.

Είναι τό πλήθος τῶν λέξεων ἢ ἀντίστοιχα τό πλήθος τῶν διευθύνσεων τῆς μνήμης. Δεδομένου ὅτι οἱ διευθύνσεις μνήμης καθορίζονται ἀπό τόν καταχωρητή διευθύνσεων μνήμης, ἂν αὐτός περιλαμβάνει N δυαδικά ψηφία, τότε τό δυνατό μέγεθος μνήμης αὐτοῦ τοῦ υπολογιστή εἶναι 2^N λέξεις. Ὡς μονάδα μετρήσεως τοῦ μεγέθους τῆς μνήμης θεωροῦμε μία K - λέξη (Kilo - word) ἴση πρὸς $2^{10} = 1024$ λέξεις ἢ μία M - λέξη (Mega - word) ἴση πρὸς $2^{20} = 1.048.576$ λέξεις.

Ἐννοεῖται ὅτι ὅταν ἕνας τύπος υπολογιστή ἔχει μέγιστο δυνατό μέγεθος μνήμης 2^N λέξεις, δέν ἔπεται ὅτι ὑποχρεωτικά καί ἡ μνήμη του πρέπει νά εἶναι τόσο μεγάλη. Συνήθως οἱ κατασκευαστές δίνουν τή δυνατότητα νά προσθέτομε τμήματα μνήμης ὡς ἀκέραια πολλαπλάσια τῶν K - λέξεων μέχρις ὅτου φθάσαμε τό 2^N .

β) Ἡ ταχύτητα μνήμης.

Ἡ ταχύτητα τῆς μνήμης ὀρίζεται ἀπό τόν κύκλο μνήμης ὁ ὁποῖος εἶναι ἴσος πρὸς τό χρόνο ἀναγνώσεως/ἐγγραφῆς.

γ) Ὁ τρόπος προσπελάσεως.

Υπάρχουν δύο τρόποι προσπελάσεως τῆς μνήμης: ἡ **τυχαία προσπέλαση** (Random Access) καί ἡ **διαδοχική προσπέλαση** (Sequential Access). Τυχαία προσπέλαση χρησιμοποιοῦν οἱ μνήμες ὑψηλῆς ταχύτητας καί αὐτό γιατί ὁ χρόνος πού ἀπαιτεῖται γιά τήν ἀνεύρεση ὁποιασδήποτε πληροφορίας, πού εἶναι ἀποθηκευμένη στή μνήμη (ἢ πού πρόκειται νά ἀποθηκευθεῖ) εἶναι ὁ ἴδιος.

Γιά νά φθάσουμε σέ μία θέση μνήμης στή διαδοχική προσπέλαση θά πρέπει πρῶτα νά περάσουμε ἀπ' ὅλες τίς προηγούμενες θέσεις. Γιά νά φθάσουμε σέ μία λέξη, ὅταν π.χ. ἔχομε νά διαβάσουμε μία μαγνητική ταινία, ἂν ἡ λέξη εἶναι ἀποθηκευμένη στό τέλος τῆς ταινίας, θά πρέπει νά διαβάσουμε ὅλη τήν ταινία.

δ) Ἡ μονιμότητα ἐγγραφῆς.

Σέ ὀρισμένα εἶδη μνήμης οἱ πληροφορίες πού ἔχουν ἀποθηκευθεῖ καταστρέφονται ἂν διακοπεῖ ἡ τάση τροφοδοσίας τοῦ υπολογιστή. Αὐτές οἱ μνήμες λέγονται «μνήμες μέ πτητικό περιεχόμενο» (Volatile Memory). Τέτοιου τύπου μνήμες εἶναι οἱ ἡμιαγωγιμικές. Ὑπάρχουν ὅμως μνήμες, ὅπως οἱ μαγνητικές, πού διατηροῦν τό περιεχόμενό τους ἀκόμη καί ὅταν διακοπεῖ ἡ τάση τροφοδοσίας τοῦ υπολογιστή. Αὐτές λέγονται **μνήμες μέ μή πτητικό περιεχόμενο** (Non - Volatile Memory).

ε) Ὁ τρόπος ἀποθηκεύσεως.

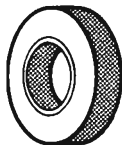
Ἡ κεντρική μνήμη τοῦ υπολογιστή πρέπει νά ἔχει τή δυνατότητα νά ἀποθηκεύει ὁποιαδήποτε πληροφορία θέλομε σέ ὁποιαδήποτε θέση καί νά κάνει τήν πληροφορία αὕτη ἀναγνώσιμη. Αὐτοῦ τοῦ εἶδους οἱ μνήμες ὀνομάζονται **μνήμες ἀναγνώσεως/ἐγγραφῆς** (Read - Write Memory ἢ RAM). Πολλές φορές θέλομε ὅμως ἕνα τμήμα τῆς μνήμης νά περιέχει πληροφορίες οἱ ὁποῖες νά μὴν καταστρέφονται μέ κανένα τρόπο. Οἱ μνήμες αὐτοῦ τοῦ εἶδους λεγονται **μνήμες μόνο ἀναγνώσεως** (Read - Only Memory - ROM).

στ) Ἡ κατασκευή τῆς μνήμης.

Μέ τόν ὄρο κατασκευή ἐννοοῦμε τό ὑλικό πού χρησιμοποιεῖται γιά τήν κατασκευή τῆς μνήμης. Μέχρι πρὶν λίγα χρόνια ἡ κεντρικὴ μνήμη κατασκευαζόταν ἀποκλειστικά ἀπὸ μαγνητικούς δακτύλιους (ferite core), σήμερα ὅμως σέ ὅλους τοὺς τύπους τῶν ὑπολογιστῶν χρησιμοποιοῦνται ἡμιαγωγικὲς μνῆμες ὑπὸ μορφὴ ὁλοκληρωμένων κυκλωμάτων.

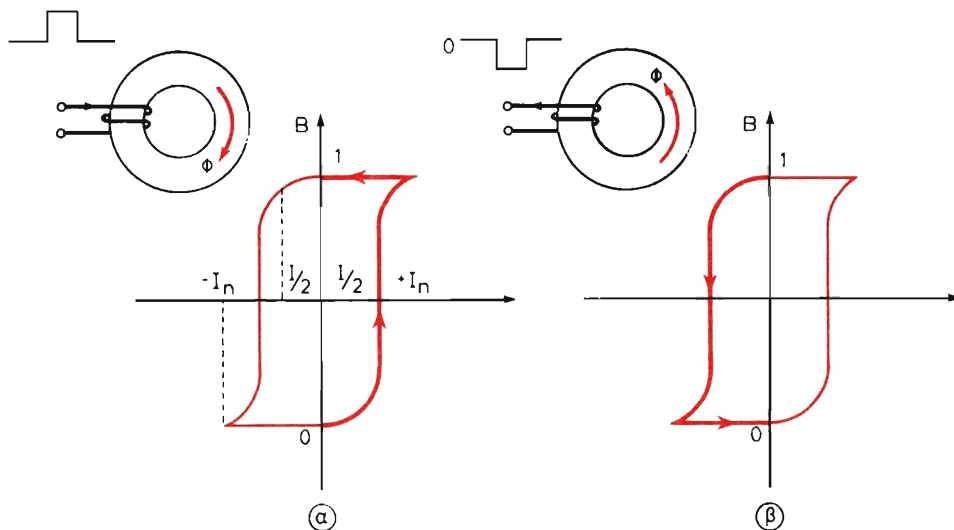
6.2.1 Ὁ μαγνητικὸς δακτύλιος ὡς στοιχεῖο κεντρικῆς μνήμης.

Ἀπὸ τὰ πρῶτα χρόνια τῆς κατασκευῆς ὑπολογιστῶν, ὡς στοιχεῖο κεντρικῆς μνήμης χρησιμοποιήθηκε ὁ μαγνητικὸς δακτύλιος (σχ. 6.2α). Ὁ δακτύλιος αὐτὸς ἀποτελεῖται ἀπὸ σιδηρομαγνητικὸ ὑλικό (μὲ διάμετρο 0.5 - 1 mm) ἐιδικῆς συστάσεως κυρίως ὡς πρὸς τὴν καμπύλη ὑστερήσεώς του, ἡ ὁποία εἶναι σχεδὸν ὀρθογώνια ὅπως φαίνεται στὸ σχῆμα 6.2β. Ἡ καμπύλη ὑστερήσεως συντελεῖ ἀφενὸς στὴν εὐκολία μαγνητίσεως καὶ ἀπομαγνητίσεως καὶ ἀφετέρου στὸ νὰ παραμένει μαγνητισμένος ὁ δακτύλιος, ἀν αὐτὸ βέβαια εἶναι ἐπιθυμητὸ (Non Volatile Memory Cell).



Σχ. 6.2α.

Μαγνητικὸς δακτύλιος ἀπὸ μαγνητικὸ ὑλικό.



Σχ. 6.2β.

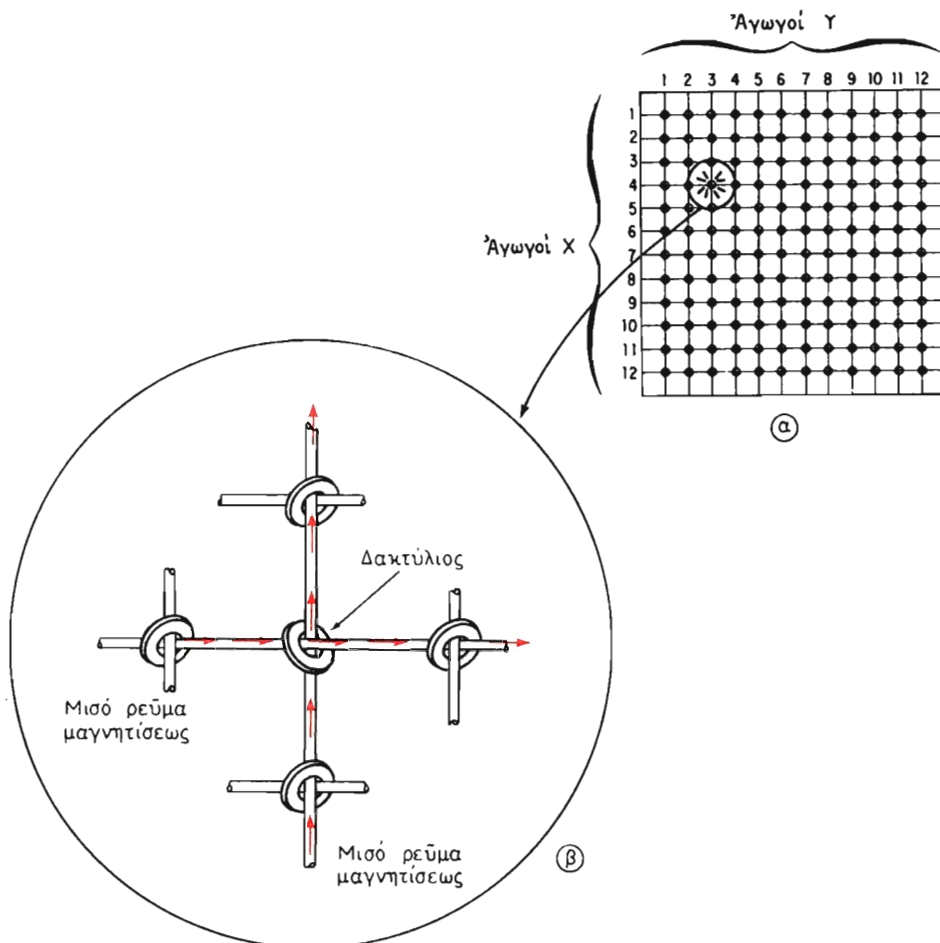
Καμπύλη ὑστερήσεως ἑνὸς δακτυλίου ἀπὸ μαγνητικὸ ὑλικό.

α) Ἐγγραφή ἀπὸ 0 σέ 1. β) Ἐγγραφή ἀπὸ 1 σέ 0.

Ἡ μαγνήτιση (κατάσταση 1) ἑνὸς δακτυλίου ἀπὸ σιδηρομαγνητικὸ ὑλικὸ πραγματοποιεῖται μὲ τὴ ροὴ ἑνὸς παλμοῦ ρεύματος μέσω ἑνὸς πηνίου [σχ. 6.2β(α)], ἐνῶ ἡ ἀπομαγνήτιση (κατάσταση 0) πραγματοποιεῖται μὲ τὴ ροὴ ἑνὸς ἴσου καὶ ἀντίθετης φορᾶς ρεύματος [σχ. 6.2β(β)]. Ὁ χρόνος ἀλλαγῆς καταστάσεως τοῦ πυρήνα εἶναι τῆς τάξεως τοῦ 0.1 msec.

Ἐπειδὴ ἐπιδιώκεται ἡ κατασκευὴ ὅσο τὸ δυνατόν μικροτέρων δακτυλίων, τὸ πηνίο (σχ. 6.2β) ἀντικαθίσταται ἀπὸ ἕνα σύρμα, λεπτὸ ὅπως ἡ ἀνθρώπινη τρίχα (σχ. 6.2γ).

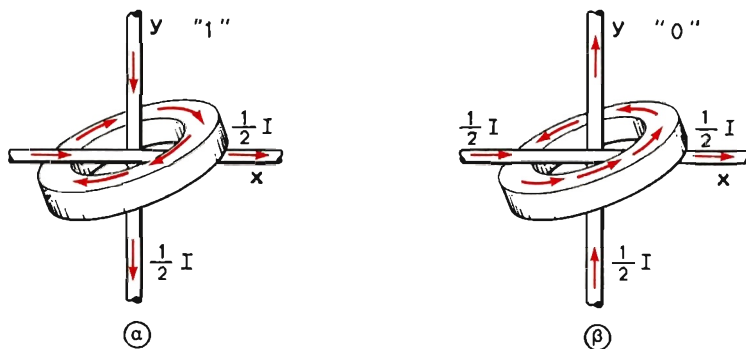
Οἱ μνῆμες ἀπὸ μαγνητικὸς δακτυλίου (core memory) εἶναι διατεταγμένες ὑπὸ μορφὴ μῆτρας ὅπως φαίνεται στὸ σχῆμα 6.2γ(α), ἔτσι ὥστε τὸ μαγνητικὸ πεδίο καθὲ δακτυλίου νὰ μὴν ἐπηρεάζεται ἀπὸ τὰ πεδία τῶν γειτονικῶν του, ὅπως φαίνεται στὸ σχῆμα 6.2γ(β).



Σχ. 6.2γ.

α) Σχηματικὴ διάταξη τῶν μαγνητικῶν δακτυλίων σὲ μῆτρα. β) Τοποθέτηση τῶν δακτυλίων στὸ χωρὸ γιὰ τὴν ἐξουδετέρωση τῶν μαγνητικῶν πεδίων τῶν γειτονικῶν δακτυλίων.

Από τόν κάθε δακτύλιο περνούν 2 σύρματα μαγνητίσεως, κάθετα μεταξύ τους καί από κάθε σύρμα περνά τό μισό ρεύμα μαγνητίσεως. Τό αποτέλεσμα τών δύο ρευμάτων είναι άθροιστικό, μπορεί δηλαδή νά μαγνητίσει καί νά άπομαγνητίσει τό δακτύλιο ένw τό ρεύμα ενός μόνο σύρματος δέν φθάνει γιά νά αλλάξει τή μαγνητική κατάσταση τοῦ δακτυλίου. Τό αποτέλεσμα τής ταυτόχρονης διαρροής ενός δακτυλίου από δύο ρεύματα φαίνεται στό σχήμα 6.2δ.



Σχ. 6.2δ.

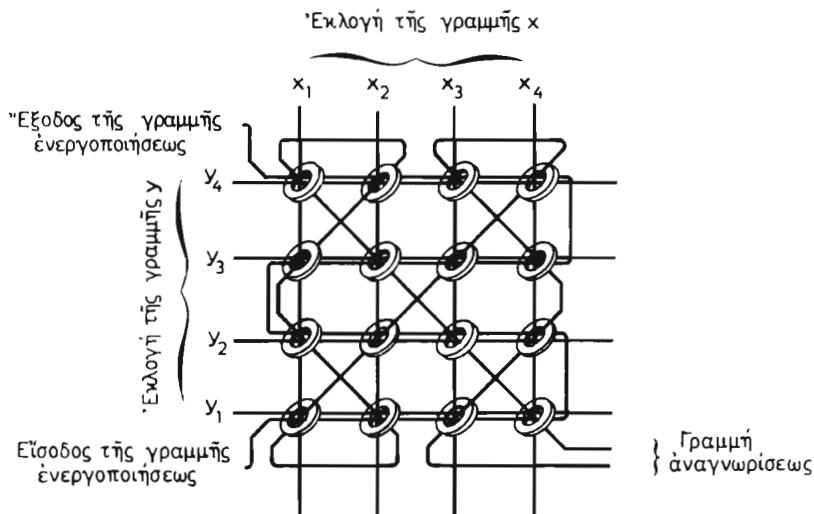
Τό αποτέλεσμα τής διαρροής τών δύο ρευμάτων σέ ένα δακτύλιο. α) Μαγνήτιση τοῦ δακτυλίου. β) Άπομαγνήτισή τοῦ δακτυλίου.

Τό πρακτικό αποτέλεσμα τής χρησιμοποίησης δύο ρευμάτων γιά τήν αλλαγή καταστάσεως ενός δακτυλίου πού βρίσκεται στήν θέση (x_4, y_3) φαίνεται καθαρά στό σχήμα 6.2γ. Άπό τό σχήμα φαίνεται ότι μπορούμε νά αλλάζουμε τή μαγνητική κατάσταση τοῦ ἐπιλεγέντος δακτυλίου χωρίς νά νά αλλάζουμε τή μαγνητική κατάσταση τών γειτονικῶν δακτυλίων.

Αν ἐπιχειρήσουμε νά διαβάσουμε τήν αποθηκευμένη, μέ τή μορφή μιᾶς ὀρισμένης μαγνητικής ροῆς, πληροφορία σέ ένα μαγνητικό δακτύλιο, θά πρέπει νά μεταβάλουμε τή ροή. Ἡ μεταβολή ὅμως τής ροῆς ἔχει ὡς αποτέλεσμα τή διαγραφή τής αποθηκευμένης πληροφορίας, ὁπότε θά πρέπει νά ἔχουμε προβλέψει κυκλώματα τέτοια ὥστε μετά τήν ἀνάγνωση τής πληροφορίας ἀπό τή θέση π.χ. (x_4, y_3) ἡ πληροφορία νά ἀποθηκεύεται ξανά στό ἴδιο στοιχείο μνήμης ὅπως καί πρίν.

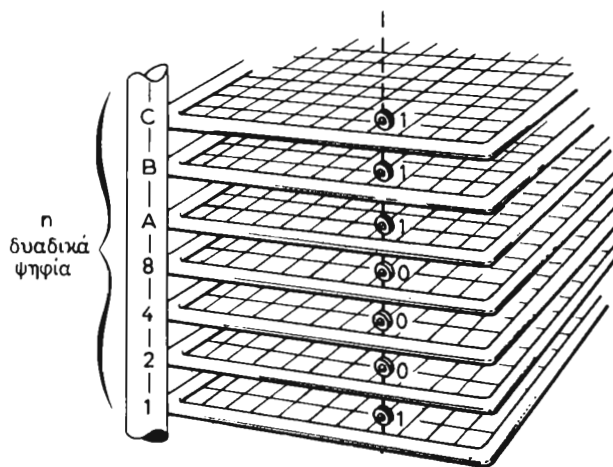
Σέ ένα ἐπίπεδο μήτρας, ὅπως αὐτό τοῦ σχήματος 6.2ε μπορούμε νά ἀποθηκεύουμε ἢ νά διαβάζουμε τό περιεχόμενο ενός μόνο δακτυλίου κάθε φορά. Δηλαδή μπορούμε σέ κάθε προσπέλαση νά γράφουμε ἢ νά διαβάζουμε μόνο ένα δυαδικό ψηφίο. Γιά νά μπορεί νά ἀποθηκεύεται μέ κάθε προσπέλαση μία ὁλόκληρη λέξη τοῦ ὑπολογιστή, χρησιμοποιοῦνται τόσα ἐπίπεδα ὅσο εἶναι τό μήκος λέξεως τοῦ ὑπολογιστή, στά ὁποῖα ἐπίπεδα γίνεται ταυτόχρονη προσπέλαση ὅπως φαίνεται στό σχήμα 6.2στ.

Γιά νά ἀποθηκευθεῖ μία πληροφορία στά διάφορα ἐπίπεδα τών δακτυλίων, μεταφέρεται πρῶτα στόν καταχωρητή δεδομένων τής μονάδας μνήμης. Στή συνέχεια ἡ διεύθυνση τών θέσεων μνήμης πού βρίσκεται στόν καταχωρητή διευθύνσεως μνήμης, ἀποκωδικοποιεῖται (γιά κάθε ἐπίπεδο ξεχωριστά) καί, ἀφοῦ ἐπιλεγεί τό ἀντίστοιχο στοιχείο μνήμης κάθε ἐπιπέδου, ἡ πληροφορία μεταφέρεται ἀπό τόν κα-



Σχ. 6.2ε.

Σχηματική διάταξη μιάς μήτρας μαγνητικών δακτυλίων από φερομαγνητικό υλικό για τή δημιουργία μνήμης ενός ηλεκτρονικού υπολογιστή (ή μήτρα περιλαμβάνει $4 \times 4 = 16$ στοιχεία μνήμης ή μνήμη 16 δυαδικών ψηφίων).



Σχ. 6.2στ.

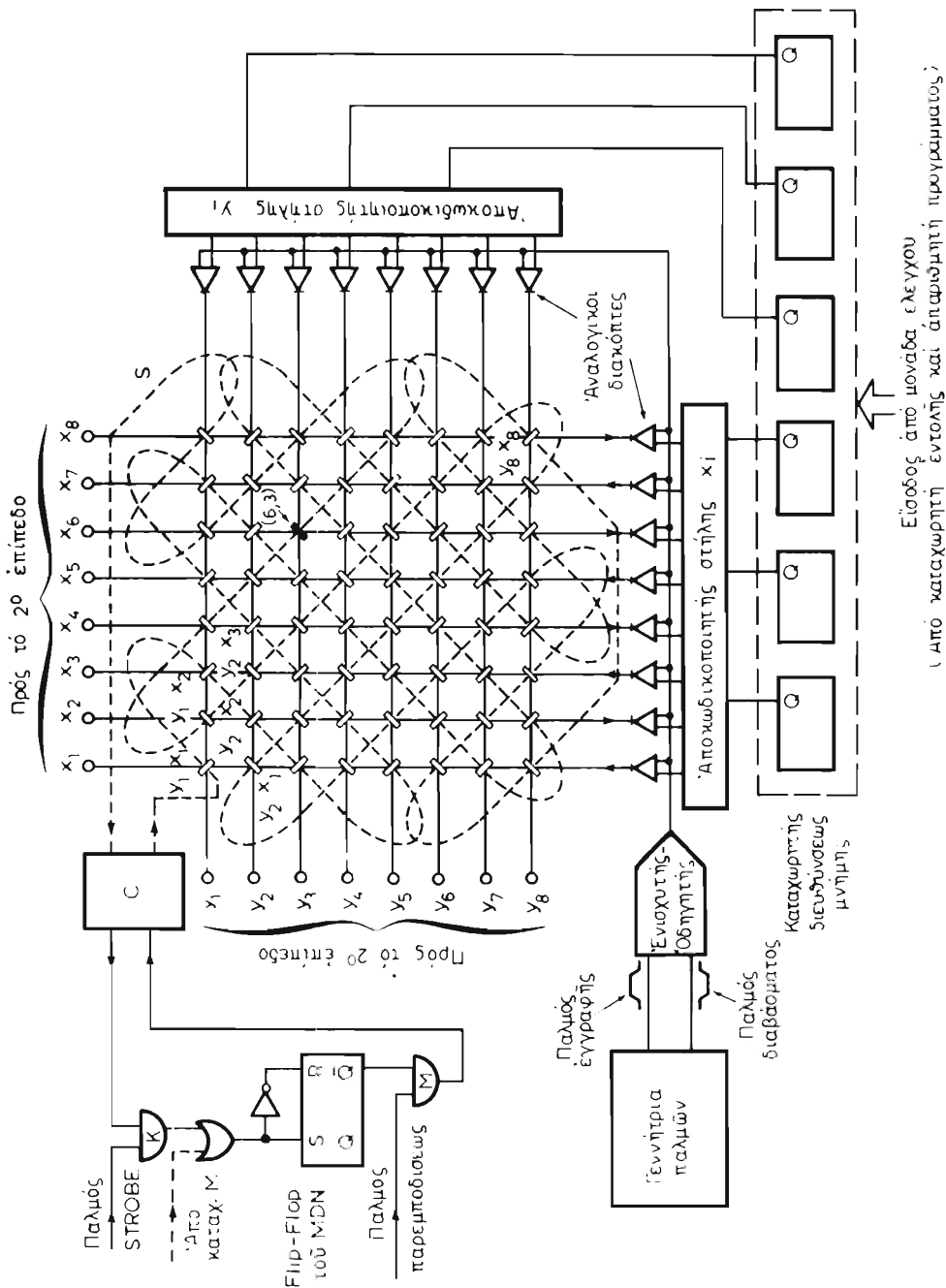
Όργάνωση μιάς πολυεπίπεδης μνήμης σε τρεις διαστάσεις καί σε λέξεις των n δυαδικών ψηφίων.

ταχρηρή δεδομένων καί αποθηκεύεται στίς αντίστοιχες θέσεις μνήμης.

Στό σχήμα 6.2ζ φαίνεται ή όργάνωση μνήμης ενός έπιπέδου σε τρεις διαστάσεις.

6.2.2 Οι μνήμες ήμιαγωγών ως στοιχεία κεντρικής μνήμης.

Παρόλο πού στήν κατασκευή μνημών μαγνητικών δακτυλίων έχει σημειωθεί



Σχ. 6.2f.

Η οργάνωση μνήμης ενός επιπέδου σε τρεις διαστάσεις.

σημαντική πρόοδος, τό μέγεθός τους καί ἡ κατανάλωση ἰσχύος πού ἀπαιτεῖται γιά τή λειτουργία τους, τίς κάνει δυσκολόχρηστες σέ σημεῖο πού νά μή χρησιμοποιοῦνται πλέον.

Ἡ ἐξέλιξη τῆς ἡλεκτρονικῆς τεχνολογίας καί ἰδιαίτερα τῆς τεχνολογίας τῶν ολοκληρωμένων κυκλωμάτων ἔδωσε τεράστια ὠθηση στήν κατασκευή μνημῶν ἀπό ἡμιαγωγούς. Ἡ μνήμη ἀπό ἡμιαγωγούς χρησιμοποιεῖται σήμερα εὐρύτατα σέ ὄλους σχεδόν τοῦς τύπους τῶν ἡλεκτρονικῶν ὑπολογιστῶν. Οἱ πολύ μικρές διαστάσεις τους (64 Kbyte βρίσκονται σέ ἓνα ολοκληρωμένο κύκλωμα διαστάσεων 4 mm²), τό χαμηλό τους κόστος καί ἡ μεγάλη ταχύτητα προσπελάσεως τίς ἔχουν καταστήσει ἀσυναγώνιστες.

Ἀνάλογα μέ τή χρήση τους καί τήν κατασκευή τους, οἱ μνήμες ἡμιαγωγῶν διακρίνονται στίς παρακάτω κατηγορίες:

- Μνήμη ἀναγνώσεως μόνο (Read Only Memory ἢ ROM).
- Μνήμη τυχαίας προσπελάσεως (Random Access Memory ἢ RAM).
- Προγραμματιζόμενη μνήμη ἀναγνώσεως μόνο (Programmable Read Only Memory ἢ PROM).
- Ἐπαναπρογραμματιζόμενη μνήμη ἀναγνώσεως μόνο (Erasable Programmable Read Only Memory ἢ EPROM).
- Ἡλεκτρικῶς Ἐπαναπρογραμματιζόμενη μνήμη ἀναγνώσεως μόνο (Electrically Alterable Read Only Memory ἢ EARAM).

α) Μνήμη μόνο ἀναγνώσεως (ROM).

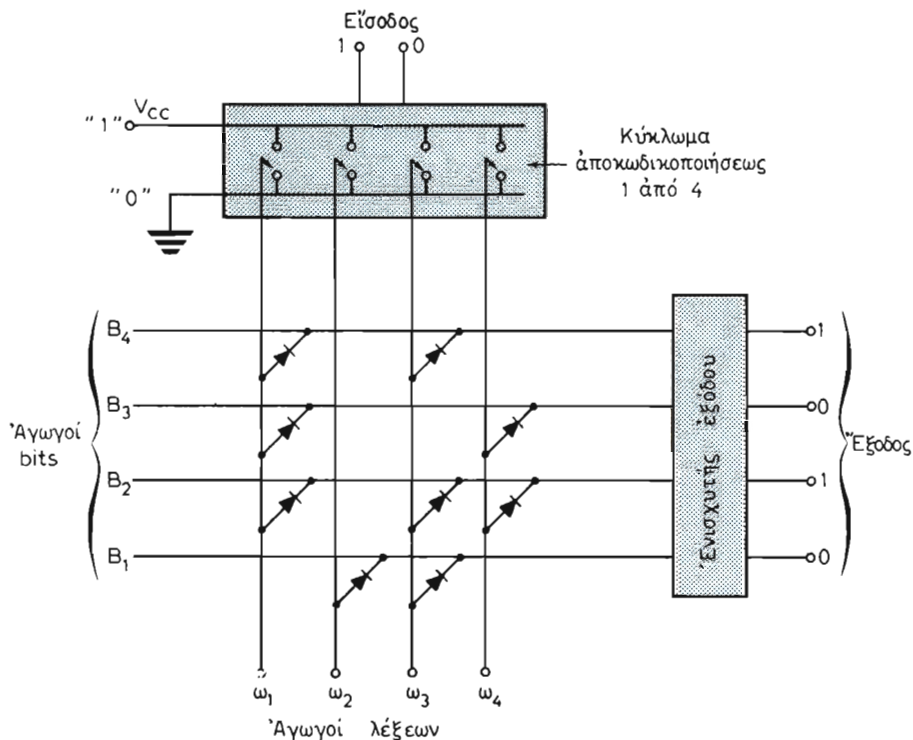
Στόν τύπο αὐτό τῆς ἡμιαγωγικῆς μνήμης, ὅπως φανερώνει καί τό ὄνομά της, μποροῦμε νά διαβάσουμε μόνο τό περιεχόμενό της. Δηλαδή, οἱ διάφορες πληροφορίες τοποθετοῦνται στίς διάφορες διευθύνσεις τῆς μνήμης κατά τό στάδιο τῆς κατασκευῆς της καί δέν ἀλλοιώνονται, κάτω ἀπό ὁμαλές συνθηκῆς λειτουργίας τοῦ ὑπολογιστή οὔτε ἀκόμη καί ὅταν διακοπεῖ ἡ τροφοδοσία τῶν κυκλωμάτων. Οἱ μνήμες αὐτοῦ τοῦ τύπου εἶναι γνωστές καί ὡς **προγραμματισμένες μνήμες μέ μάσκα** (Mask programmable ROM).

Στίς μνήμες αὐτές ἀποποθηκεύονται συνήθως μικροπρογράμματα καί σταθερές οἱ ὁποῖες χρησιμοποιοῦνται συχνά ἀπό τόν ἡλεκτρονικό ὑπολογιστή κατά τή διάρκεια τῆς ἐπεξεργασίας.

Γιά νά μπορέσει ὁ χειριστής νά διαβάσει δεδομένα ἀπό μία ROM δίνει στόν ὑπολογιστή τή διεύθυνση τῆς μνήμης τοῦ κατάλληλου δυαδικοῦ ψηφίου πού πρέπει νά διαβαστεῖ, καί μετὰ παίρνει στήν κατάλληλη γραμμή ἐξόδου τή ζητούμενη πληροφορία.

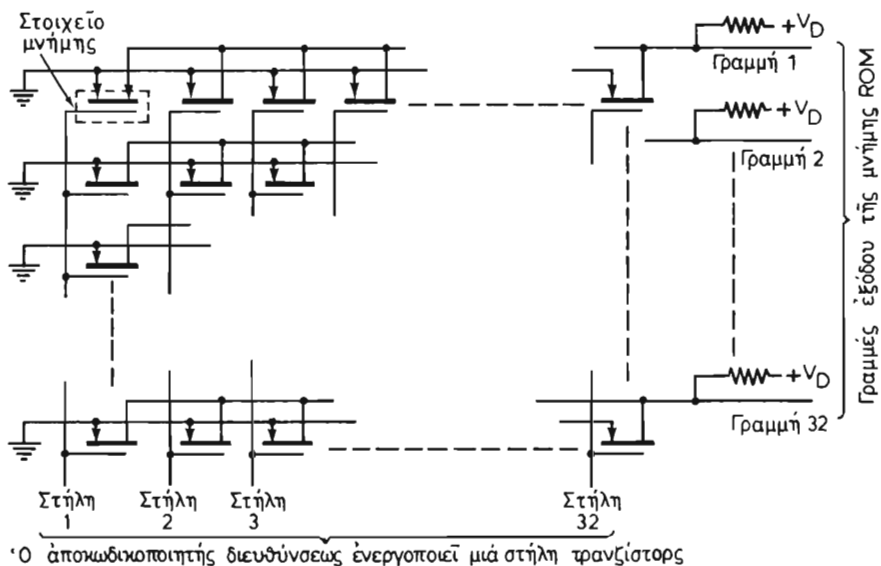
Ἡ μνήμη ROM συνήθως εἶναι ὁργανωμένη κατά τέτοιο τρόπο, ὥστε νά εἶναι δυνατή ἡ ἀνάγνωση πολλῶν δυαδικῶν ψηφίων (bits) ταυτόχρονα. Ἔτσι μποροῦμε νά διαβάσουμε ταυτόχρονα λέξεις τῶν 4,8 ἢ 16 δυαδικῶν ψηφίων. Ὅταν λέμε λοιπόν ὅτι ἔχομε μία μνήμη ROM π.χ. 2048 bits ἐννοοῦμε ὅτι εἶναι ὁργανωμένη σέ 2048 × 1 ἢ σέ 256 × 8 bits, ἀνάλογα πάντα μέ τό μήκος τῆς χρησιμοποιούμενης λέξεως.

Στό σχῆμα 6.2η παριστάνεται ἡ δομή μιᾶς ROM μέ διόδους καί στό σχῆμα 6.2θ



Σχ. 6.2η.

Μνήμη ROM με διόδους χωρητικότητας 16 bit (4 x 4 bit).



Σχ. 6.2θ.

Μνήμη ROM με τρανζίστορ μετάλλου - ήμιαγωγού.

μία μνήμη ROM με τρανζίστορ μετάλλου - ήμιαγωγού (MOS - Metal oxide semiconductor).

Η μνήμη του σχήματος 6.2η είναι μνήμη χωρητικότητας 4 λέξεων των 4 ψηφίων. Έτσι, όπως φαίνεται και στο σχήμα, το περιεχόμενο της μνήμης είναι:

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

Για να διαβάσουμε το περιεχόμενο μιας λέξης της ROM, ενεργοποιούμε με τον κωδικοποιητή τον άνωγό που αντιστοιχεί στη λέξη αυτή (στην προκειμένη περίπτωση τον άνωγό ω_3) και παίρνουμε στην έξοδο το περιεχόμενο της λέξεως μνήμης ω_3 που είναι η σειρά 1010.

Οι μνήμες ROM είναι προγραμματισμένες από τον κατασκευαστή για μία ειδική χρήση και πωλούνται ως τυποποιημένα προϊόντα, όπως οι γεννήτριες χαρακτήρων, οι κωδικοποιητές, οι μετατροπείς κωδίκων κ.ά.

Σε πολλές περιπτώσεις, όπως η περίπτωση χρησιμοποίησης μικρο - υπολογιστών σε οικιακές συσκευές, απαιτείται η έγγραφη σε ROM ενός ειδικευμένου προγράμματος. Σ' αυτή την περίπτωση ο πελάτης, δηλαδή η εταιρία κατασκευής των οικιακών συσκευών, διατυπώνει τις απαιτήσεις του για το περιεχόμενο της ROM και ο κατασκευαστής κατασκευάζει τη μνήμη.

Σήμερα κατασκευάζονται ήμιαγωγικές μνήμες ROM με χωρητικότητα μέχρι 64 Kbytes σε ένα κρύσταλλο ήμιαγωγού διαστάσεων 4 mm².

Ο χρόνος προπελάσεως των ROM είναι συνήθως από 100 - 300 nsec (όπου 1 nsec = 10⁻⁹ sec).

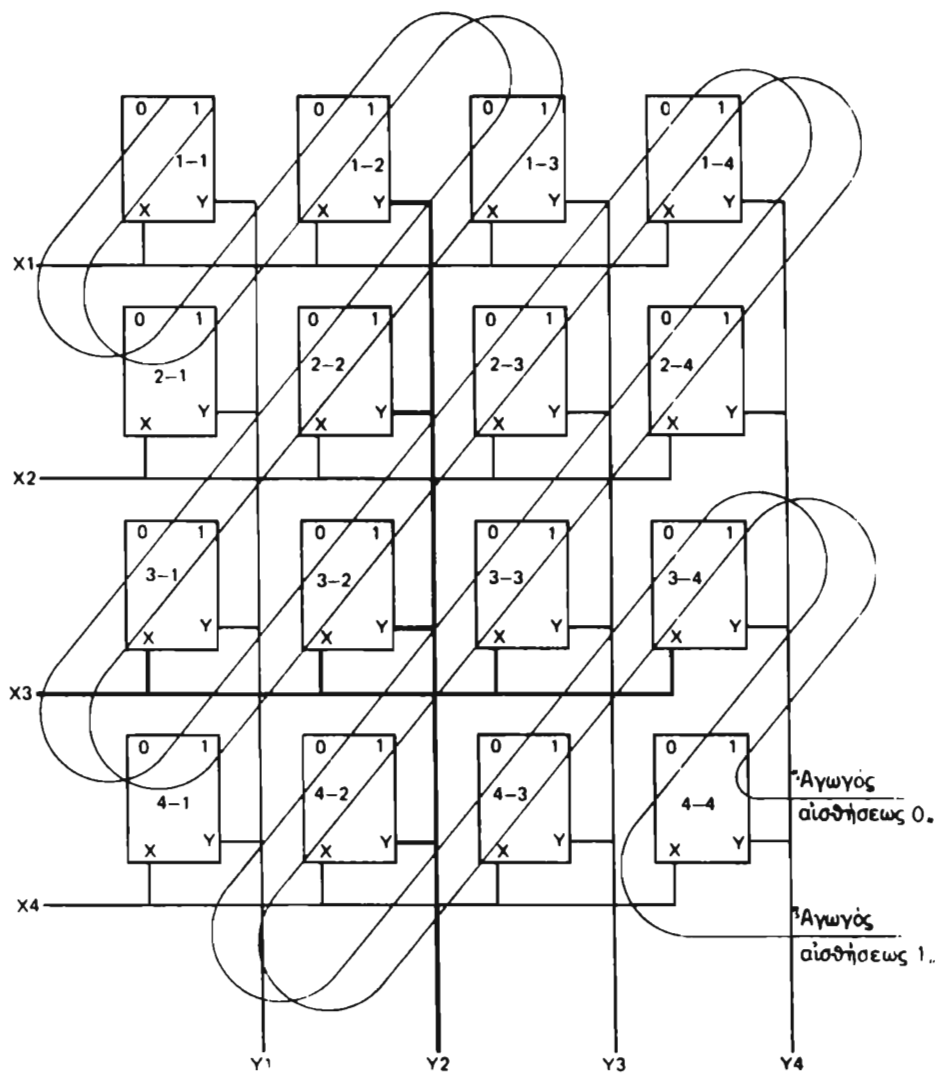
β) Μνήμη τυχαίας προσπελάσεως (RAM).

Μπορούμε να γράψουμε σε οποιαδήποτε διεύθυνση της μνήμης RAM και να διαβάσουμε το περιεχόμενο οποιασδήποτε διευθύνσεώς της.

Οι μνήμες τυχαίας προσπελάσεως χωρίζονται σε δύο κατηγορίες: τη **στατική** (static) και τη **δυναμική** (dynamic). Όταν έγγραφει σε μία θέση στατικής μνήμης μία πληροφορία, αυτή δεν αλλοιώνεται παρά μόνο όταν σταματήσει ή τροφοδοτήση των κυκλωμάτων της. Στη δυναμική όμως μνήμη οι πληροφορίες αλλοιώνονται με την πάροδο του χρόνου και έτσι απαιτείται επανατοποθέτηση των πληροφοριών. Και τα δύο αυτά είδη των μνημών RAM χάνουν το περιεχόμενό τους όταν σταματήσει ή τροφοδοτήση των κυκλωμάτων τους. Είναι δηλαδή μνήμες πτητικού περιεχομένου (volatile memory) σε αντίθεση με τις μνήμες ROM που είναι μνήμες μη πτητικού περιεχομένου (non - volatile).

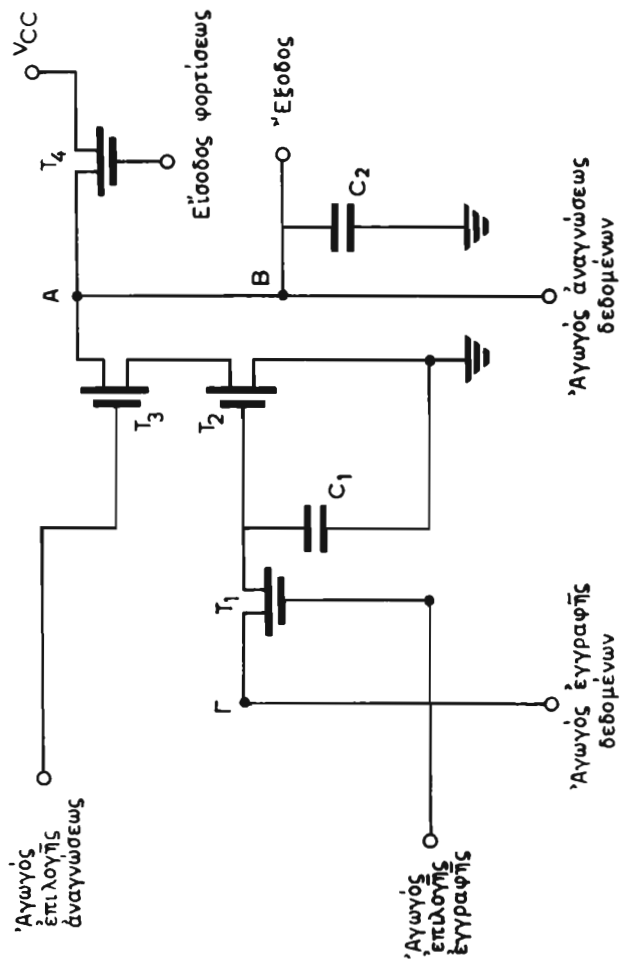
Στις στατικές μνήμες RAM ως στοιχεία μνήμης χρησιμοποιούμε συνήθως τα ηλεκτρονικά στοιχεία μνήμης (φλίπ - φλόπ), ενώ στις δυναμικές μνήμες χρησιμοποιούμε ειδικούς πυκνωτές ή διάφορα άλλα κατάλληλα ήμιαγωγικά στοιχεία.

Στο σχήμα 6.2ι φαίνεται η τυπική όργευνση μιας στατικής μνήμης RAM με πολυδονητές, ενώ στο σχήμα 6.2ια φαίνεται η τυπική όργάνωση μιας δυναμικής μνήμης RAM με πυκνωτές. Στα σχήματα 6.2ιβ, 6.2ιγ και 6.2ιδ φαίνονται αντίστοιχα μικροφωτογραφία μνήμης RAM χωρητικότητας 64 Kbytes, καρτέλα στατικής μνήμης RAM συνολικής χωρητικότητας 64 Kbytes, και καρτέλα δυναμικής μνήμης RAM χωρητικότητας 500 Kbytes.



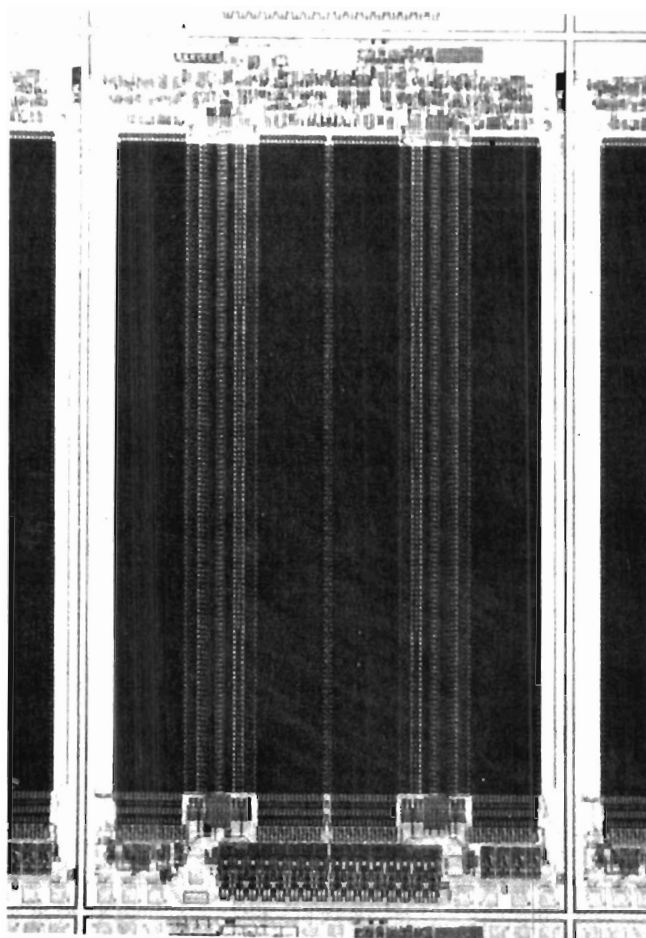
Σχ. 6.2i.

Τυπική οργάνωση στατικής μνήμης RAM με πολυδονητές.



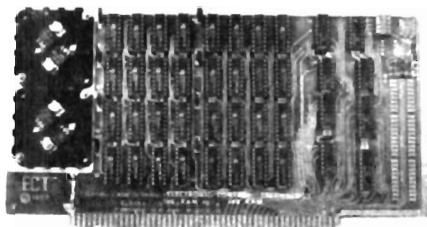
Σχ. 6.2α.

Τυπική οργάνωση δυναμικής μνήμης RAM με πυκνωτές.



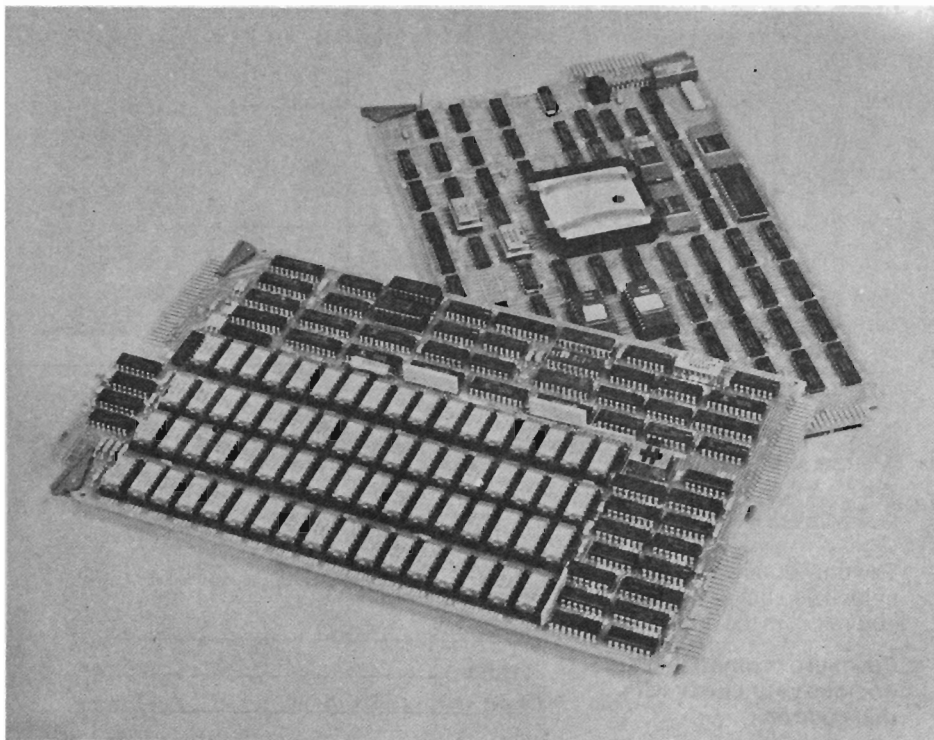
Σχ. 6.2ιβ.

Μικροφωτογραφία μιάς μνήμης RAM χωρητικότητας 64 Kbytes.



Σχ. 6.2ιγ.

Καρτέλα στατικής μνήμης RAM συνολικής χωρητικότητας 64 Kbytes. Διακρίνονται τα ολοκληρωμένα κυκλώματα της μνήμης (32 x 2 K).



Σχ. 6.2ιδ.

Καρτέλα δυναμικής μνήμης RAM μνήμης χωρητικότητας 500 Kbytes.

γ) Προγραμματιζόμενη μνήμη μόνο ανάγνωσης (PROM).

Ο τύπος αυτός της μνήμης είναι μία παραλλαγή της μνήμης ROM ή όποια πωλείται από τον κατασκευαστή κενή, δηλαδή χωρίς να έχει αποθηκεύσει σ' αυτή οποιαδήποτε πληροφορία. Στο σχήμα 6.2ιε παριστάνεται η δομή μία μνήμης PROM με διόδους. Η μνήμη περιλαμβάνει τρεις λέξεις των 3 ψηφίων.

Η μνήμη αυτή μπορεί να προγραμματισθεί με μία ειδική συσκευή που ονομάζεται «προγραμματιστής μνημών PROM» (PROM Programmer).

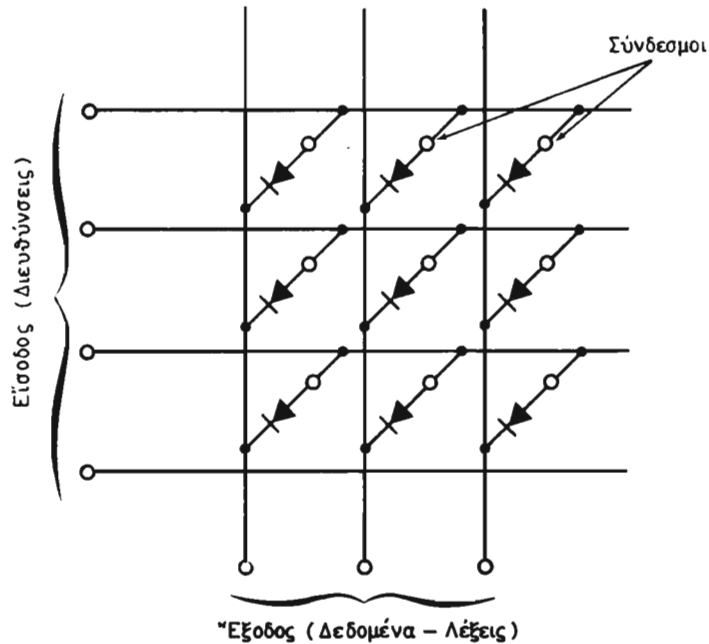
Στο σχήμα 6.2ιστ φαίνεται ένας τέτοιος προγραμματιστής.

Κατά τη διάρκεια του προγραμματισμού της, οι σύνδεσμοι των γραμμών της μνήμης καταστρέφονται όταν αποθηκεύεται σ' αυτή ένα «0» και παραμένουν όταν αποθηκεύεται ένα «1».

Οι μνήμες του τύπου αυτού παρέχουν τη δυνατότητα στον κατασκευαστή συστημάτων να τις προγραμματίζει μόνος του αντί να τις προγραμματίζει ο κατασκευαστής των μνημών, πράγμα τό όποιο είναι πολύ οικονομικότερο.

δ) Έπαναπρογραμματιζόμενη μνήμη μόνο ανάγνωσης (EPROM - EAROM).

Γιά να μπορέσει να καλυφθεί τό κενό που υπάρχει μεταξύ των δυνατοτήτων των μνημών PROM, ROM και RAM έχουν κατασκευασθεί οι έπαναπρογραμματιζόμενες μνήμες μόνο ανάγνωσης, οι όποιες άφενός μās παρέχουν τά πλεονεκτήματα των μνημών ROM, άφετέρου τίς δυνατότητες, μερικώς, των μνημών RAM ά-



Σχ. 6.2α.
Μνήμη PROM με διόδους (3×3 bit).

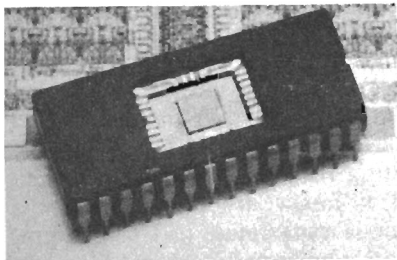


Σχ. 6.2ιστ.
Προγραμματιστής μνημών PROM.

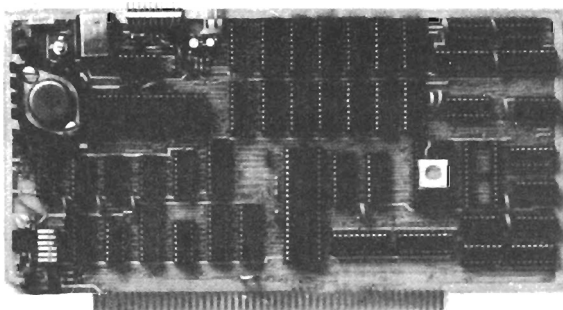
φού μπορούν νά επαναπρογραμματισθούν με μία ειδική διαδικασία.

Υπάρχουν δύο είδη τέτοιων μνημών: εκείνες στις οποίες οι πληροφορίες, πού είναι γραμμένες σβήνονται με υπεριώδη ακτινοβολία, και λέγονται EPROM (Erasable Programmable Read Only Memory) και εκείνες πού οι πληροφορίες σβήνονται με την εφαρμογή υψηλής τάσεως και λέγονται EAROM (Electrically Alternate Read Only Memory).

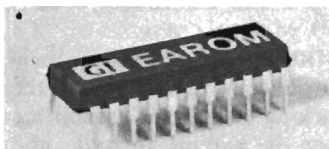
Στό σχήμα 6.2ιζ φαίνεται μία μνήμη EPROM τής εταιρίας INTEL επί μιάς καρτέλας υπολογιστού χωρητικότητας 16 Kbytes. Φαίνεται χαρακτηριστικά τό ολο-



α



β



γ

Σχ. 6.2ιζ.

- α) Μία μνήμη EPROM της INTEL με χωρητικότητα 64 Kbytes και χρόνο προσπελάσεως 250 ns (INTEL 2764).
 β) Καρτέλα υπολογιστή στην οποία φαίνεται μία μνήμη EPROM που περιέχει τό λειτουργικό σύστημα του υπολογιστή.
 γ) Μνήμη EAROM σε ολοκληρωμένο κύκλωμα χωρητικότητας 16 Kbytes (= 16.000).

κληρωμένο κύκλωμα της μνήμης μέσα από τό γυάλινο κάλυμμά του τό όποιο χρησιμοποιείται για τό σβήσιμο τών πληροφοριών με υπεριώδη ακτινοβολία.

Στό κάλυμμα αυτό τοποθετούμε αυτόκόλλητο άδιαφανές ύλικό, για νά άποφύγομε τήν τυχαία διαγραφή πληροφοριών από τήν πτώση υπεριώδους άκτινοβολίας π.χ. από μία λάμπα φθορισμού.

6.3 Περιφερειακές Μονάδες Μνήμης (μαζική μνήμη).

Ή κεντρική μνήμη χρησιμοποιείται από τόν υπολογιστή για άποθήκευση προγράμματος καί δεδομένων για μικρό χρονικό διάστημα, δηλαδή για όσο διάστημα άπαιτείται για τήν έπεξεργασία.

Όταν χρειάζεται άποθήκευση δεδομένων ή προγραμμάτων για μεγάλο χρονικό διάστημα ή όταν ό όγκος τών δεδομένων ή τών προγραμμάτων είναι μεγάλος, τότε ό υπολογιστής χρησιμοποιεί για τήν άποθήκευσή τους τίς περιφερειακές μονάδες μνήμης.

Οι κυριότεροι τύποι περιφερειακής μνήμης είναι οι έξης:

- Μονάδα μαγνητικής ταινίας (magnetic tape) καί μαγνητικής κασέτας (cassette tape).
- Μονάδα μαγνητικών δίσκων (Magnetic disks).

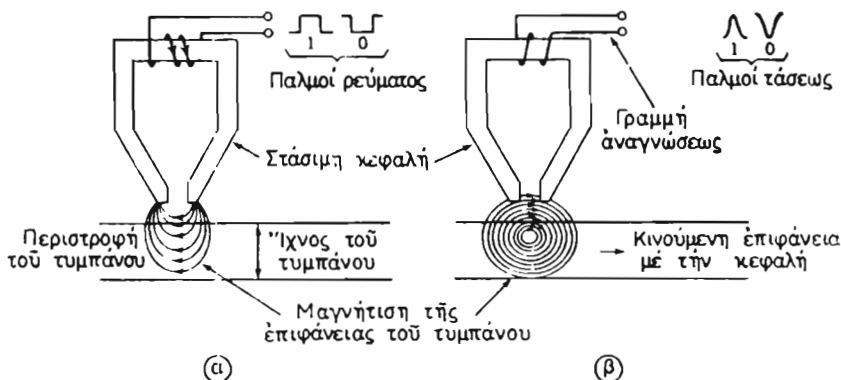
- Μονάδα μαγνητικού τυμπάνου (Magnetic drum).
- Μονάδα εύκαμπτων μαγνητικών δισκετών (Floppy disk).
- Μονάδα διατρήσεως/άναγνώσεως δελτίων (Punched card).
- Μονάδα διατρήσεως/άναγνώσεως χαρτοταινίας (Paper tape).

Όπως βλέπομε, οι περισσότεροι τύποι βοηθητικής μνήμης είναι μαγνητικού τύπου. Βασική αρχή λειτουργίας τους είναι η κίνηση μιάς επιφάνειας καλυμμένης με ένα λεπτό στρώμα από μαγνητικό υλικό μέσα σέ ένα διακοπόμενο μαγνητικό πεδίο. Έτσι, κάθε σημείο της επιφάνειας αυτής μαγνητίζεται μόνιμα, εφόσον υπάρχει μαγνητικό πεδίο, την ώρα που η επιφάνεια, κινείται μπροστά από τό σύστημα τό όποιο δημιουργεί τό μαγνητικό πεδίο. Τό σύστημα δημιουργίας ενός μαγνητικού πεδίου λέγεται *μαγνητική κεφαλή* καί είναι ουσιαστικά ένας ηλεκτρομαγνήτης μέ ένα πολύ μικρό διάκενο μεταξύ τών πόλων του. Όταν, την ώρα της άναγνώσεως, ή μαγνητισμένη επιφάνεια διέρχεται από τή μαγνητική κεφαλή, δημιουργείται μία τάση έξ' επαγωγής ανάλογη μέ τό μαγνητισμό του σημείου της επιφάνειας που βρίσκεται μπροστά από τήν κεφαλή.

Στό σχήμα 6.3α(α) φαίνεται ό τρόπος μέ τόν όποιο γίνεται μία μαγνητική έγγραφη σέ μία επιφάνεια, ένω στό σχήμα 6.3α(β) ό τρόπος μέ τόν όποιο γίνεται ή άνάγνωση της μαγνητικής έγγραφης.

Έδώ θά πρέπει νά αναφέρομε τά διάφορα βασικά χαρακτηριστικά τών περιφερειακών μνημών τά όποια μς δίνουν τή βάση συγκρίσεως τών διαφόρων τύπων μεταξύ τους. Αυτά είναι:

- Η πυκνότητα έγγραφης σέ δυαδικά ψηφία ανά μονάδα μήκους της επιφάνειας του μέσου έγγραφης.
- Η όλική χωρητικότητα της μνήμης σέ Kbytes ή Mbytes.
- Η ταχύτητα προσπελάσεως.
- Η ταχύτητα μεταφοράς πληροφοριών.
- Τό κόστος της μνήμης σέ σχέση μέ τήν όλική χωρητικότητά της ή τό κόστος ανά δυαδικό ψηφίο που άποθηκεύεται.



Σχ. 6.3α.

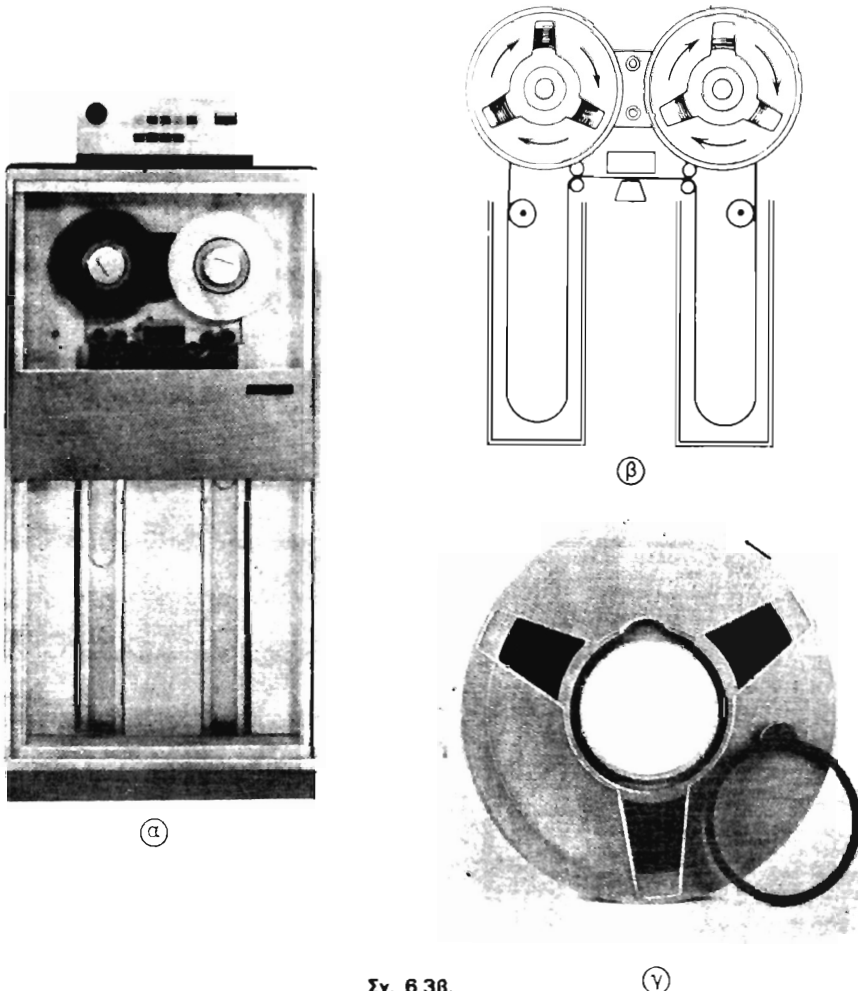
Σχηματική παράσταση τρόπου α) έγγραφης καί β) άναγνώσεως μιάς μαγνητικής έγγραφης πάνω σέ μία επιφάνεια μαγνητικού υλικού.

6.3.1 Μονάδα μαγνητικής ταινίας.

Οι μονάδες αυτές καταγράφουν και διαβάζουν τις πληροφορίες πάνω σε μία μαγνητική ταινία (σχ. 6.3β).

Οι διάφορες πληροφορίες ή δεδομένα καταγράφονται στη μαγνητική ταινία όπως ακριβώς γίνεται και στα κοινά μαγνητόφωνα.

Γενικά η έγγραφη και η ανάγνωση των πληροφοριών στις περιφερειακές μονάδες μπορεί να γίνει διαδοχικά (serial) ή τυχαία (random). Στις μαγνητικές ταινίες η αποθήκευση και η ανάγνωση γίνεται διαδοχικά μόνο. Έτσι για να βρούμε μία συγκεκριμένη πληροφορία, θα πρέπει να βρούμε πρώτα τό σημείο της ταινίας που είναι γραμμένη η συγκεκριμένη πληροφορία που ζητούμε. Ή προσπέλαση δηλαδή σε αυτό το είδους τις μνήμες είναι αρκετά αργή. Παρόλα αυτά, οι μαγνητικές ταινίες χρησιμοποιούνται πάρα πολύ και για την έγγραφη ενδιαμέσων αποτελεσμάτων, αλλά κυρίως για τη μονιμότερη αποθήκευση μεγάλων αρχείων πληροφοριών.



Σχ. 6.3β.

α) Μονάδα μαγνητικών ταινιών. β) Σχηματική παράσταση της λειτουργίας μονάδας μαγνητικών ταινιών. γ) Μαγνητική ταινία.



Σχ. 6.3γ.

Έγγραφή πληροφοριών σε μαγνητική ταινία των 7 καναλιών (χρησιμοποιείται κωδικοποίηση BCD).

Τό πλάτος της μαγνητικής ταινίας είναι συνήθως 1/2 ίντσα και τό μήκος της από 2400 ως 3600 πόδια. Κάθε χαρακτήρας (byte) γράφεται κατά τό πλάτος της ταινίας ως συνδυασμός 7 ή 9 δυαδικών ψηφίων, ανάλογα μέ τόν τύπο τοῦ ὑπολογιστή. Έτσι κάθε δυαδικό ψηφίο γράφεται πάνω σέ ἓνα «*ἄχνος*» (κανάλι) τῆς ταινίας (σχ. 6.3γ).

Ἡ πυκνότητα ἐγγραφῆς, δηλαδή τό πλῆθος τῶν δυαδικῶν λέξεων πού μποροῦν νά καταγραφοῦν σέ μία ίντσα τῆς ταινίας, κυμαίνεται συνήθως ἀπό 200 ως 6250 bpi (bytes per inch) καί ἡ ταχύτητα κινήσεως τῶν μαγνητικῶν ταινιῶν κυμαίνεται ἀπό 12.5 ως 200 ips (inches per second). Έτσι μία μαγνητική ταινία τῶν 1600 bpi, ἡ ὁποία περιστρέφεται μέ ταχύτητα 200 ips καί ἔχει μήκος 2450 πόδια, ἔχει ταχύτητα μεταφορᾶς δεδομένων ἀπό ἡ πρός τόν ἠλεκτρονικό ὑπολογιστή $1600 \times 200 = 320000$ bytes/sec (ἡ χαρακτήρες/sec) καί χωρητικότητα $1600 \times 2400 \times 12 \approx 46$ Mbytes (1 foot = 12 inches).

Οι κώδικες πού χρησιμοποιοῦνται γιά τήν ἐγγραφή τῶν πληροφοριῶν στίς μαγνητικές ταινίες εἶναι ὁ κώδικας BCD, ὁ κώδικας EBCDIC καί ὁ δυαδικός κώδικας.

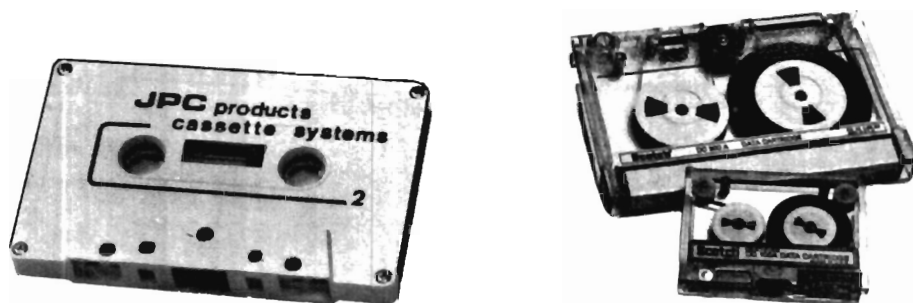
Τελευταία ἀρχισε νά χρησιμοποιεῖται σέ εὐρεία κλίμακα ἡ μαγνητική κασέτα πού χρησιμοποιεῖται κυρίως γιά τήν ἀποθήκευση μικρῆς ποσότητας πληροφοριῶν. Οι πιό πολλοί τύποι κασετῶν εἶναι ἴδιοι μέ αὐτούς πού χρησιμοποιοῦνται στά κοινά κασετόφωνα (σχ. 6.3δ).

Ἡ χρησιμοποίηση τῆς μαγνητικῆς ταινίας ως μονάδας περιφερειακῆς ἢ βοηθητικῆς μνήμης παρουσιάζει τά ἐξῆς **πλεονεκτήματα**:

- Μικρό κόστος
- Μεγάλη χωρητικότητα ἀποθηκεύσεως πληροφοριῶν
- Μικρό ὄγκο καί
- Δυνατότητα ἐπαναχρησιμοποίησεως.

Ὡς **μειονεκτήματα** μποροῦμε νά θεωρήσουμε τά ἐξῆς:

- Τήν διαδοχική προσπέλαση τῶν πληροφοριῶν
- Τή σχετικά μικρή ταχύτητα μεταφορᾶς δεδομένων
- Τή δυσκολία ἀλλαγῶν τῶν ἀποθηκευμένων πληροφοριῶν καί



Σχ. 6.36.

Τύποι μαγνητικών κασετών.

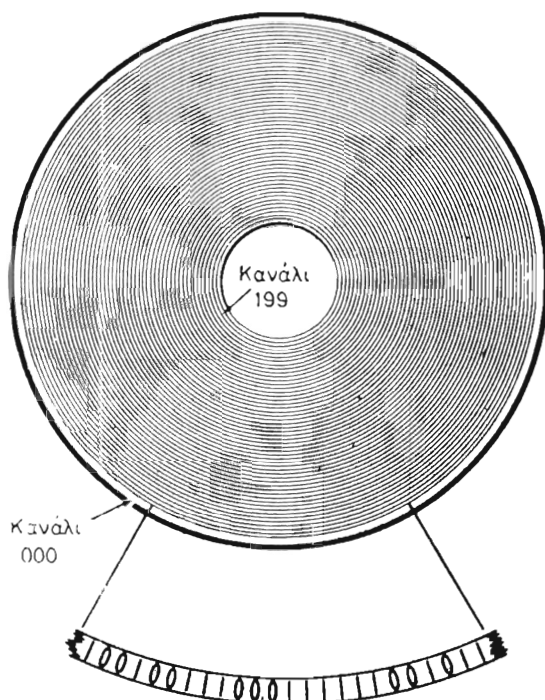
- Τίς απαιτήσεις του χώρου φυλάξεως των πληροφοριών.
(Προφύλαξη δηλαδή από υγρασία, θερμοκρασία, σκόνη κλπ.).

6.3.2 Μονάδα μαγνητικών δίσκων.

Τό πιό σπουδαίο καί τό πιό άποδοτικό μέσο άποθηκεύσεως πληροφοριών είναι ο μαγνητικός δίσκος. Αύτός συνδυάζει όλα τά πλεονεκτήματα τής μαγνητικής ταινίας σύν τό πλεονέκτημα τής τυχαίας προσπελάσεως των πληροφοριών.

Ο μαγνητικός δίσκος αποτελείται από μία κυκλική μεταλλική πλάκα, έπιστρωμένη μέ μαγνητικό ύλικό. Η διάμετρος του δίσκου κυμαίνεται από 6 ίντσες μέχρι καί 3 πόδια καί έχει πάχος 1/6 τής ίντσας.

Η κάθε έπιφάνεια του δίσκου καλύπτεται από ίχνη (ή κανάλια) (100 ως 1000) σέ μορφή όμόκεντρων κύκλων όπως φαίνεται στό σχήμα 6.3ε. Κατά μήκος των Ι-

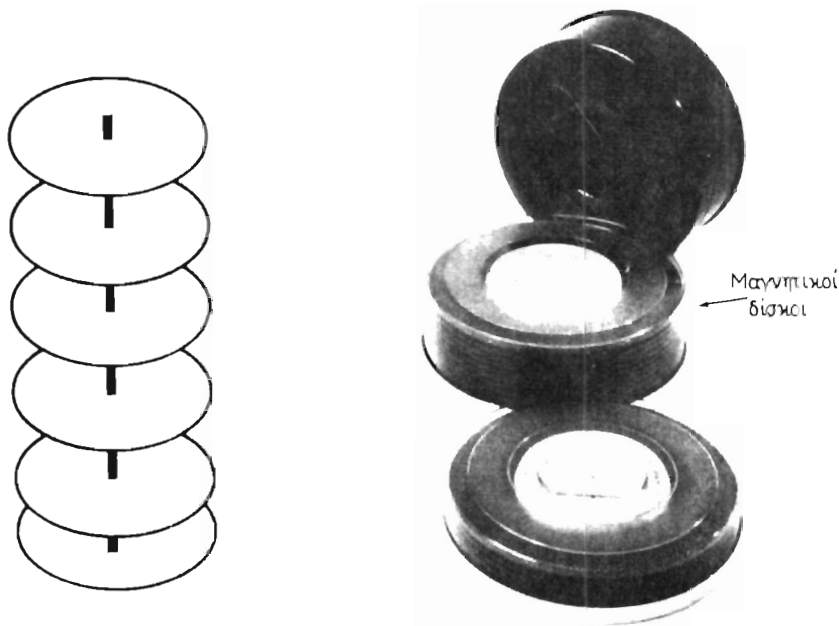


Σχ. 6.3ε.

Μία κάτοψη ενός δίσκου διακοσίων καναλιών καί τμήμα ενός καναλιού.

χνών αποθηκεύονται μαγνητικά οι πληροφορίες. Τό πλήθος των δυαδικών ψηφίων που μπορούν νά αποθηκευθούν σέ ένα ίχνος είναι 25000 - 50000.

Πολλοί μαζί δίσκοι, πάνω σέ έναν κοινό άξονα, σχηματίζουν μία δέσμη δίσκων (disk rack), πού περιστρέφεται μέ ύψηλή ταχύτητα (15 ως 2400 στροφές τό λεπτό) (σχ. 6.3στ). Στό διάκενο μεταξύ των δίσκων υπάρχουν οι μαγνητικές κεφαλές οι όποιες μέ τή βοήθεια κινητών βραχιόνων, όπως φαίνεται στό σχήμα 6.3ζ τοποθετούνται πάνω σέ οποιοδήποτε κανάλι τοῦ δίσκου στόν όποιο άντιστοιχοῦν. Μία δέσμη δίσκων έχει συνήθως χωρητικότητα από 5 ως 500 Mbytes.



Σχ. 6.3στ.

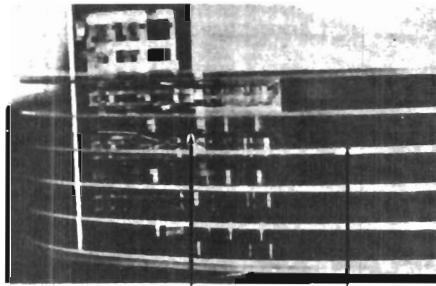
Δέσμη μαγνητικῶν δίσκων: α) Σέ διαγραμματική διάταξη, β) σέ φωτογραφία.

Υπάρχουν δύο είδη δίσκων: Οι κινητοί (Removable) καί οι σταθεροί (Fixed). Οι κινητοί δίσκοι έχουν τό πλεονέκτημα τής άπεριόριστης χωρητικότητας, γιατί όταν συμπληρώνεται μία δέσμη δίσκων, αφαιρείται καί μία νέα κενή δέσμη τοποθετείται στή θέση της. Όπως δηλαδή γίνεται καί στίς μαγνητικές ταινίες. Έπειδή ή σταθερή δέσμη δίσκων δέ μετακινείται, έχει τό πλεονέκτημα ότι περιέχει συνήθως περισσότερους δίσκους, καί έτσι έχει μεγαλύτερη χωρητικότητα καί λιγότερες βλάβες.

Υπάρχουν μονάδες μαγνητικῶν δίσκων οι όποιες περιέχουν ένα κινητό καί ένα σταθερό δίσκο.

Στό σχήμα 6.3η φαίνεται μονάδα μαγνητικῶν δίσκων. Στό σχήμα 6.3θ παριστάνεται μονάδα δίσκου (σταθεροῦ) μικρής χωρητικότητας.

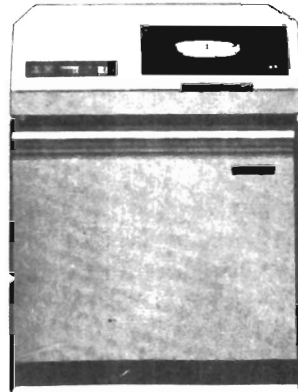
Η ταχύτητα μεταφοράς των πληροφοριών στους δίσκους είναι πολύ μεγαλύτερη από ό,τι στίς μαγνητικές ταινίες. Κυμαίνεται από 100 Kbytes/sec μέχρι 10 Mbytes/sec ενώ ό μέσος χρόνος τής άναγνώσεως ή έγγραφής μίας πληροφορίας κυμαίνεται από 30 ως 400 msec.



Κεφαλές
έγγραφής ανάγνωσης Μαγνητικοί δίσκοι

Σχ. 6.3ζ.

Τό σύστημα των μαγνητικών δίσκων.



Σχ. 6.3η.

Μονάδα μαγνητικών δίσκων.



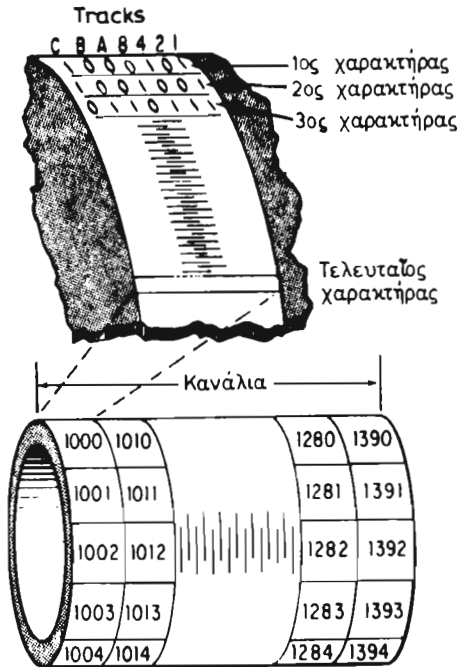
Σχ. 6.3θ.

Φωτογραφία μιάς συσκευής σταθερού μαγνητικού δίσκου μικρής χωρητικότητας.

Ός **πλεονεκτήματα** τής χρησιμοποιήσεως των μαγνητικών δίσκων θεωρούμε: τή μεγάλη χωρητικότητα, τή δυνατότητα έπαναχρησιμοποίησεως, τήν τυχαία ή διαδοχική προσπέλαση καί τή μεγάλη ταχύτητα μεταφοράς δεδομένων. Ός **μειονέκτημα**: τό μεγάλο κόστος καί τή δυσκολία τής σχεδιάσεως τής λειτουργίας έ- νός τέτοιου συστήματος.

6.3.3 Μονάδα μαγνητικού τυμπάνου.

Η άρχή λειτουργίας των μαγνητικών τυμπάνων είναι ίδια όπως καί στους προη-



Σχ. 6.3ι.
Σχηματική παράσταση ενός
μαγνητικού τυμπάνου.

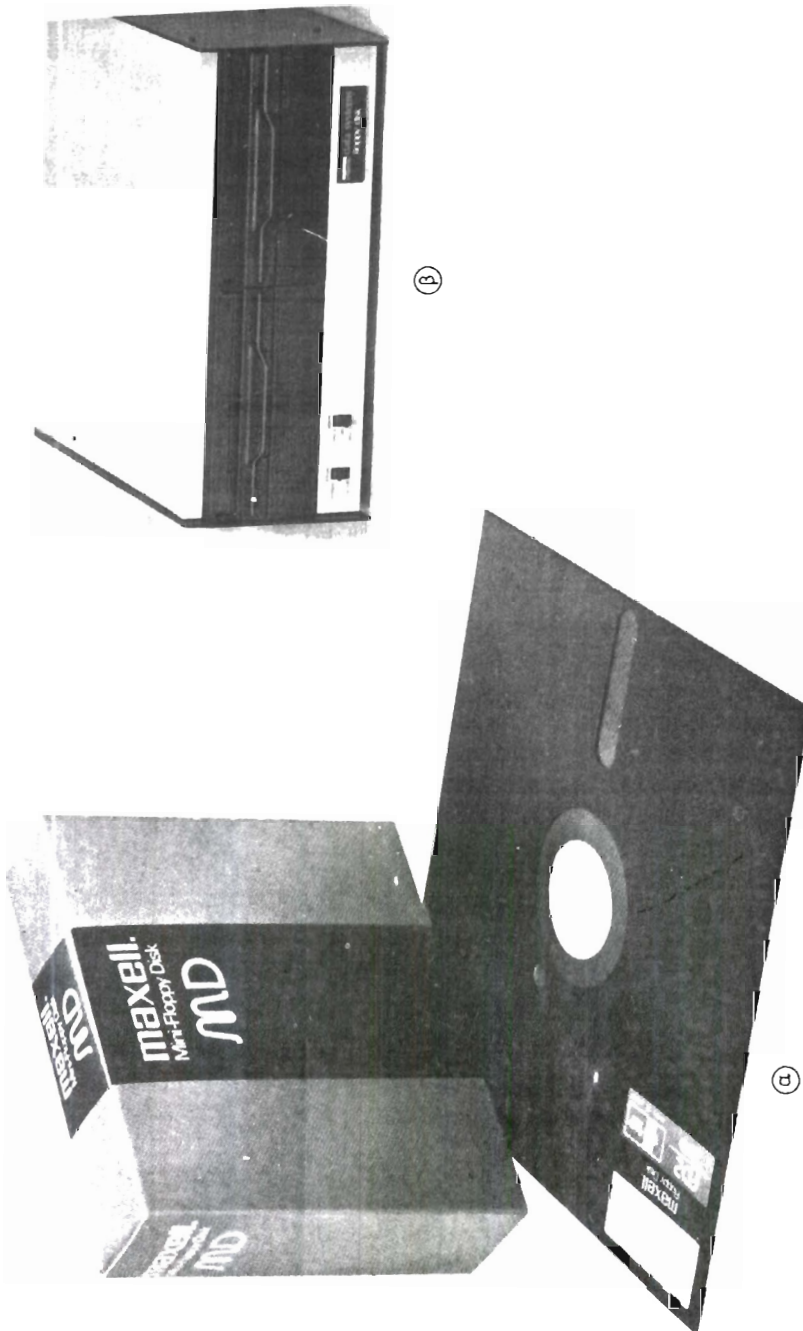
γούμενους τύπους μαγνητικών μνημών. Ἡ διαφορά τους από αυτές είναι ότι τό μαγνητικό ὑλικό τοποθετεῖται στήν ἐπιφάνεια ἑνός περιστρεφόμενου τυμπάνου. Κάθε τύμπανο (σχ. 6.3ι) φέρει στήν ἐπιφάνειά του ἕνα ἀριθμό καναλιῶν (γύρω ἀπό τό τύμπανο) πάνω στά ὁποῖα καταγράφονται οἱ πληροφορίες. Γιά κάθε κανάλι τοῦ τυμπάνου χρησιμοποιεῖται ξεχωριστή κεφαλὴ ἀναγνώσεως καί ἐγγραφῆς.

Τό τύμπανο περιστρέφεται μέ σταθερὴ ταχύτητα, συνήθως 3000 στροφές/min. Ἡ ταχύτητα περιστροφῆς κυμαίνεται μεταξύ 8 καί 35 m/sec. Ὁ μέσος χρόνος προσπελάσεως ἑνός χαρακτήρα εἶναι ἴσος πρὸς τό μισό χρόνο μιᾶς πλήρους περιστροφῆς. Ἡ ταχύτητα μεταφορᾶς δεδομένων εἶναι ἀρκετά ὑψηλὴ (1 Mbyte/sec). Ἡ χωρητικότητά ὅμως τῶν τυμπάνων εἶναι ἀρκετά χαμηλότερη ἀπὸ ὅ,τι στοὺς μαγνητικούς δίσκους. Συνήθως κυμαίνεται ἀπὸ 1 ὠς 8 Mbytes καί τό κόστος τῶν τυμπάνων εἶναι ἀρκετά ὑψηλό.

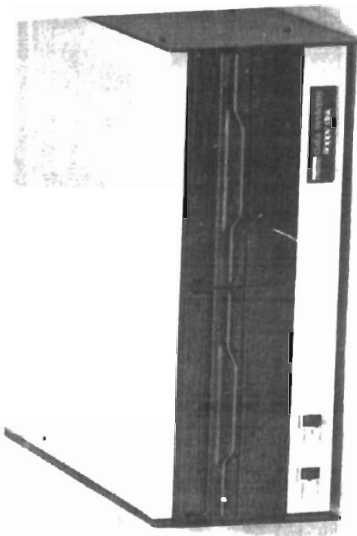
6.3.4 Μονάδα εὐκάμπτων μαγνητικῶν δισκετῶν (floppy disk).

Ὁ εὐκάμπτος μαγνητικὸς δίσκος (σχ. 6.3ια) δέν κοστίζει ἀκριβὰ. Ἡ χωρητικότητά του κυμαίνεται ἀπὸ 80 Kbytes μέχρι 1 Mbyte. Ἀποτελεῖται ἀπὸ ἕνα δίσκο ἀπὸ πλαστικό μέ ἐπικάλυψη ἀπὸ μαγνητικό ὑλικό. Ἡ ἐπιφάνεια τοῦ δίσκου ἔχει περίπου 70 κανάλια καί ἡ πυκνότητά ἐγγραφῆς ἀνὰ ἴντσα κυμαίνεται ἀπὸ 3200 ὠς 6400 bits. Περιστρέφεται μέ ταχύτητα 360 rpm.

Τό εἶδος αὐτὸ τῶν περιφερειακῶν μνημῶν χρησιμοποιεῖται εὐρέως σέ συστήματα μικρο καί μίνι ὑπολογιστῶν. Τό μικρὸ κόστος καί τό μικρὸ μέγεθος καθιστοῦν τὴν εὐκάμπτη μαγνητικὴ δισκέτα πολὺ εὐχρηστὴ καί χρήσιμη ὅταν ἀπαιτεῖται μικρὸ μέγεθος βοηθητικῆς μνήμης.



β



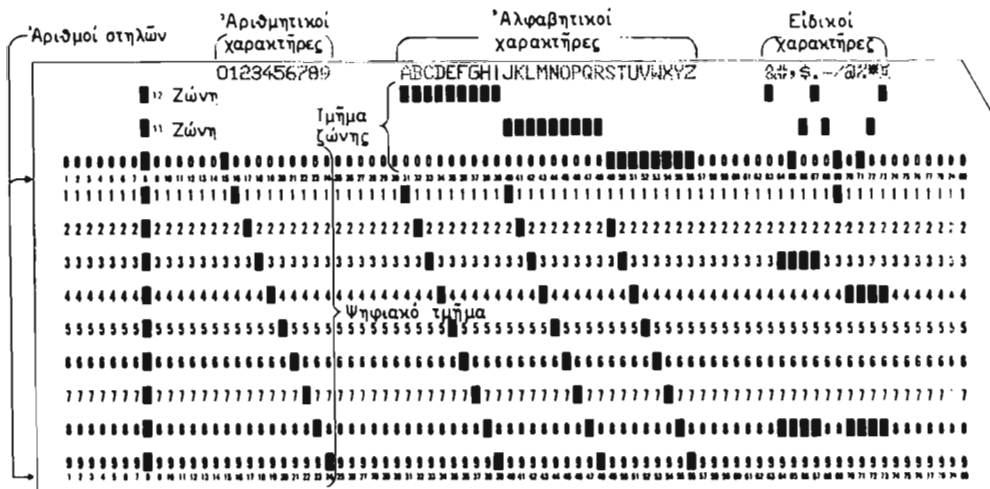
Σχ. 6.3α.
α) Εύκαρτος μαγνητικός δίσκος. β) Μονάδα ανάγνωσης μαγνητικού δίσκου.

6.3.5 Μονάδα διατήρησης/αναγνώσεως δελτίων.

Τό διάτρητο δελτίο αποτέλεσε γιά ένα πολύ μεγάλο διάστημα, από την αρχή της εφαρμογής των υπολογιστών μέχρι καί σήμερα, όχι μόνο τό συνηθέστερο μέσο εισόδου πληροφοριών στον ηλεκτρονικό υπολογιστή, αλλά καί τό συνηθέστερο μέσο αποθηκεύσεως προγραμμάτων καί δεδομένων.

Οι πληροφορίες καταγράφονται στό δελτίο μέ τή διάτρηση καταλλήλων συνδυασμών όπών, διαφορετικών γιά κάθε χαρακτήρα.

Υπάρχουν διάφορα είδη διατρήτων δελτίων. Τό πλέον επικρατέστερο είναι αυτό πού φαίνεται στό σχήμα 6.3ιβ. Τό δελτίο αυτό αποτελείται από 80 στήλες.



Σχ. 6.3ιβ.

Τό διάτρητο δελτίο των 80 στηλών.

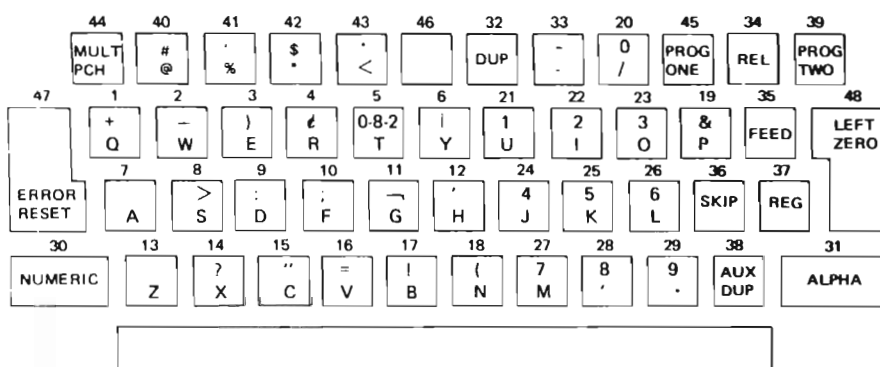
Τό δελτίο κατασκευάζεται από ανθεκτικό μή ηλεκτραγωγό χαρτί, διαστάσεων $7\frac{3}{8}'' \times 3\frac{1}{4}''$ καί πάχους 0.007''. Κάθε δελτίο χωρίζεται κατακόρυφα σέ 80 στήλες καί όριζόντια σέ 12 γραμμές. Έπομένως διαθέτει 960 θέσεις διατρήσεως. Σέ κάθε στήλη καταχωρείται ένας μόνο χαρακτήρας καί έτσι σέ ένα δελτίο μπορούν νά καταχωρηθούν 80 χαρακτήρες. Οι γραμμές 0 - 9 αποτελούν τό αριθμητικό τμήμα καί οι γραμμές 0 - 12 τό τμήμα ζώνης. Κάθε αριθμητικός χαρακτήρας παριστάνεται πάνω στό δελτίο μέ μία τρύπα όρθογωνίου σχήματος στην αντίστοιχη γραμμή του ψηφιακού τμήματος, καί κάθε άλφαριθμητικός χαρακτήρας παριστάνεται μέ δύο τρύπες από τίς όποιες ή μία βρίσκεται στό ψηφιακό τμήμα καί ή άλλη στό τμήμα ζώνης.

Μέ αυτό τόν τρόπο μπορούμε νά αποθηκεύσουμε δεδομένα ή καί όλόκληρα προγράμματα σέ διάτρητες κάρτες.

Γιά τήν διάτρηση των καρτελών αυτών χρησιμοποιούνται ειδικές μηχανές, οι «διατρητικές μηχανές» (Keyrunching machines). Τά δεδομένα ή τά προγράμματα τά όποια θά περαστούν στις καρτέλες μέ τή βοήθεια των διατρητικών μηχανών, γράφονται πρώτα σέ ειδικά έντυπα. Στό σχήμα 6.3ιγ φαίνεται μία διατρητική μηχανή μέ τό πληκτρολόγιό της.



α

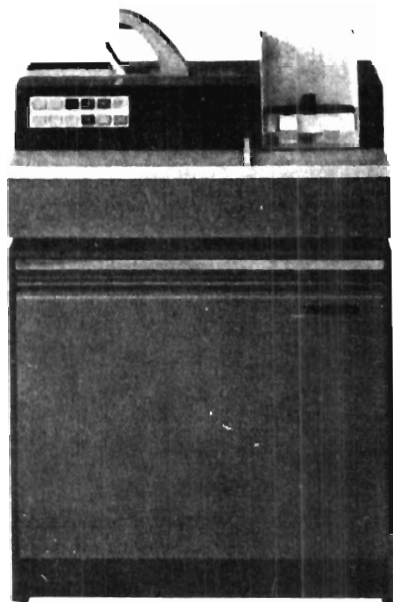


β

Σχ. 6.3ιγ.

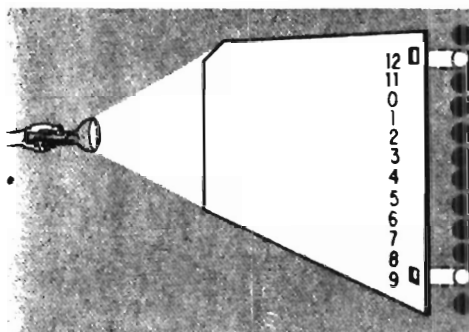
α) Διατρητική μηχανή: 1) Θέση τοποθέτησεως των άδιατρήτων καρτελών. 2) Τοποθέτηση της άδιάτρητης καρτέλας για διάτρηση. 3) Έκτύπωση. 4) Πακέτο καρτελών που έχουν ήδη διατρηθεί.

β) Το πληκτρολόγιο μιάς διατρητικής μηχανής.



Σχ. 6.3ιδ.

Τυπική μονάδα ανάγνωσης
διατρήτων δελτίων.



Σχ. 6.3ιε.

Σχηματική παράσταση του τρόπου ανάγνωσης
μέ φωτοκύτταρα.

Τά δεδομένα τά όποια έχουν διατρηθεί έπάνω στό δελτίο, εισάγονται στόν ηλεκτρονικό ύπολογιστή μέ τή βοήθεια τής μονάδας ανάγνωσης δελτίων (σχ. 6.3ιδ). 'Η ανάγνωση τών δεδομένων γίνεται μέσω φωτοκυττάρων ή ψηκτρών (σχ. 6.3ιε).

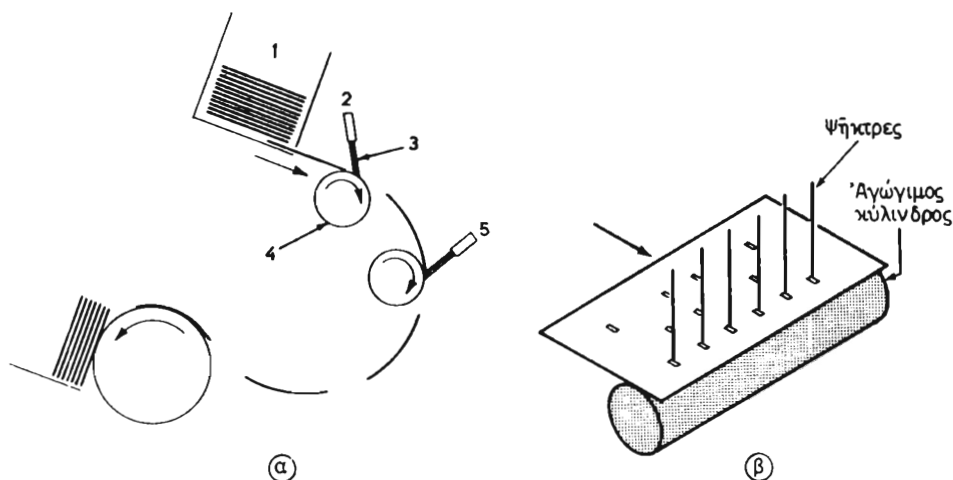
Στό σχήμα 6.3ιστ φαίνονται ή λειτουργία του μηχανισμού ανάγνωσης μέ ψηκτρες καί ανάγνωσης μέ φωτοκύτταρα.

'Εκτός από τίς παραπάνω μηχανές πού απαιτούνται γιά τήν διάτρηση καί τήν ανάγνωση τών διατρήτων δελτίων, ένα ύπολογιστικό κέντρο χρειάζεται άκόμα καί άλλες μηχανές, έφόσον χρησιμοποιεί τά διάτρητα δελτία ως βοηθητική μνήμη. Οι μηχανές αυτές είναι οι έξής:

- **'Η έπαληθευτική μηχανή (Verifier).** Χρησιμοποιείται γιά τόν έντοπισμό σφαλμάτων διατρήσεως.
- **'Η διαλογική μηχανή (Sorting Machine).** Διαχωρίζει καί ταξινομεί τά διάτρητα δελτία.
- **'Η αναπαραγωγική μηχανή (Reproducer).** Μπορεί νά αναπαραγάγει αυτόματα όλόκληρο ή μέρος από τό περιεχόμενο ενός δελτίου σέ ένα άλλο λευκό.
- **'Η μηχανή διατρήσεως δελτίων χωρίς πληκτρολόγιο (Card Punch).** Χρησιμοποιείται γιά τήν έξοδο τών στοιχείων από τόν ύπολογιστή (δεδομένων ή προγραμμάτων) καί τή διαφύλαξή τους σέ δελτία.

"Όπως είπαμε καί προηγουμένως, τό πρώτο μέσο βοηθητικής ή περιφερειακής μνήμης πού χρησιμοποιήθηκε ήταν τά διάτρητα δελτία λόγω τού χαμηλού τους κόστους.

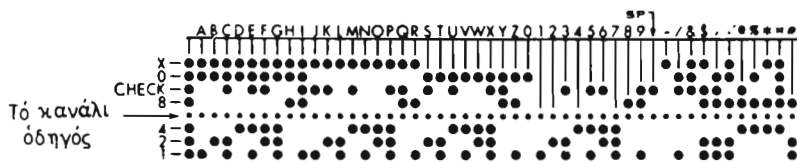
Μετά δμως τήν ανακάλυψη καί ανάπτυξη τών διαφόρων μαγνητικών μέσων βοηθητικής μνήμης ή χρήση τους τείνει νά έκλείψει.



Σχ. 6.31στ.

α) Διαγραμματική λειτουργία του μηχανισμού ανάγνωσης με ψήκτρες: 1) Διάτρητες καρτέλες προς ανάγνωση. 2) Ψήκτρα ανάγνωσης. 3) Στέλεχος ψήκτρας. 4) Αγώγιμος κύλινδρος. 5) Επόμενη σειρά ψηκτρών.

β) Διαγραμματική διάταξη ανάγνωσης με ψήκτρες.



Σχ. 6.31ζ.

Σχηματική παράσταση της διάταξης των καναλιών και των στηλών σε μία χαρτοταινία. Η κωδικοποίηση των χαρακτήρων είναι για 8 bits ανά byte (ψηφία ανά χαρακτήρα).

6.3.6 Μονάδα διατήρησης/ανάγνωσης χαρτοταινίας (Paper Tape – Read, Punch).

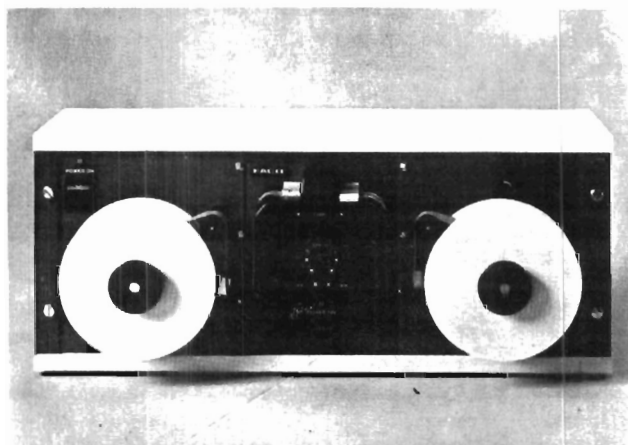
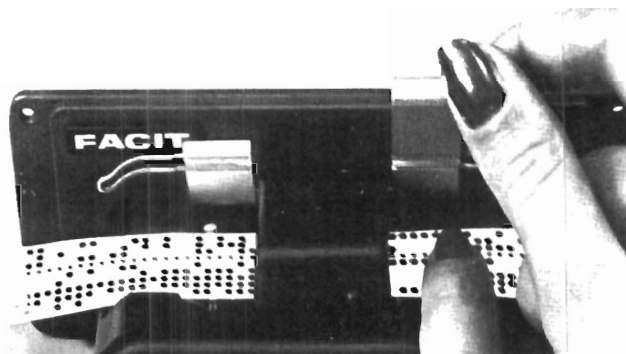
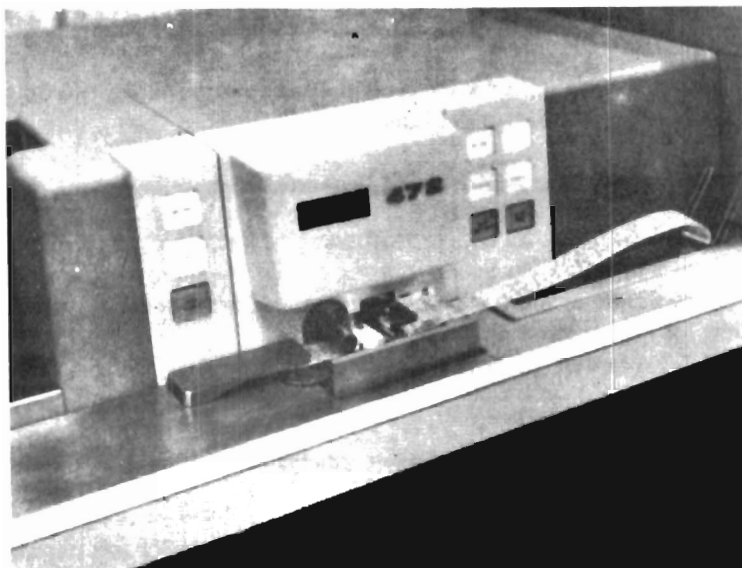
Η χαρτοταινία είναι ταινία από μη αγώγιμο χαρτί με μήκος από 200 ως 1000 πόδια και πλάτους περίπου 1 ίντσα (σχ. 6.31ζ).

Όπως και η μαγνητική ταινία, έτσι και αυτή χωρίζεται, κατά μήκος σε 5 ως 8 παράλληλα ίχνη ή κανάλια (tracks or channels). Στή μέση φέρει μία σειρά από μικρές τρύπες οι οποίες χρησιμοποιούνται ως οδηγοί για την κίνηση της ταινίας.

Σε κάθε στήλη της ταινίας διατρέχεται ένας χαρακτήρας. Το αριθμητικό σύστημα κωδικοποιήσεως που χρησιμοποιείται στην ταινία είναι ανάλογο με τον αριθμό των καναλιών.

Η πυκνότητα έγγραφης είναι περίπου 10 bytes ανά ίντσα και η ταχύτητα μετακινήσεως είναι σχετικά μικρή (50 - 100 πόδια/sec) λόγω του κινδύνου καταστροφής (σχισίματος).

Η διάτρηση και ανάγνωση των ταινιών αυτών γίνεται με ανάλογο με τα διάτρη-



Σχ. 6.3ιη.
Μηχανή διατήρησης και ανάγνωσης διατρήτων ταινιών χάρτου.

τα δελτία τρόπο. Στο σχήμα 6.3ιη φαίνονται οι ειδικές συσκευές ανάγνωσης και διατήρησης χαρτοταινιών.

Η χρήση της διάτρητης ταινίας χαρτιού παρέχει τό πλεονέκτημα της φθηνής αποθηκεύσεως δεδομένων, αλλά μειονεκτεί σε σύγκριση με τις άλλες βοηθητικές μνήμες σε θέματα χωρητικότητας, ταχύτητας, μεταφοράς δεδομένων και αξιοπιστία, γιατί σκίζεται πολύ εύκολα.

6.4 Έρωτήσεις.

1. Τι καλείται χρόνος προσπελάσεως σε έναν H/Y;
2. Ποιά τά βασικά χαρακτηριστικά της κεντρικής μνήμης ενός H/Y;
3. Πώς είναι οργανωμένη ή κεντρική μνήμη ενός H/Y;
4. Τι καλείται μνήμη Μόνο Άναγνώσεως;
5. Τι καλείται μνήμη Τυχαίας Προσπελάσεως;
6. Τι καλείται Προγραμματιζόμενη Μνήμη Μόνο Άναγνώσεως;
7. Ποιά πλεονεκτήματα και ποιά μειονεκτήματα προσφέρει ή ηλεκτρικώς έναναπρογραμματιζόμενη μνήμη μόνο άναγνώσεως σε σύγκριση με τή μνήμη τυχαίας προσπελάσεως;
8. Άναφέρετε διαφόρους τύπους βοηθητικής ή περιφερειακής μνήμης.
9. Ποιά τά πλεονεκτήματα χρησιμοποίησεως βοηθητικής μνήμης σε ένα H/Y;

ΚΕΦΑΛΑΙΟ ΕΒΔΟΜΟ

ΠΕΡΙΦΕΡΕΙΑΚΕΣ ΜΟΝΑΔΕΣ

7.1 Γενικά.

Όπως αναφέρθηκε καί στα προηγούμενα κεφάλαια, η επικοινωνία ενός ηλεκτρονικού υπολογιστή με το έξωτερικό περιβάλλον γίνεται με τις περιφερειακές μονάδες εισόδου - εξόδου των πληροφοριών (I/O).

Τέτοιες μονάδες, όπως είδαμε, είναι ή μονάδα διατήρησης καί αναγνώσεως καρτών, οι μαγνητικές ταινίες, οι μαγνητικοί δίσκοι καί άλλες. Στο κεφάλαιο αυτό θα ασχοληθούμε με τις υπόλοιπες μονάδες εισόδου - εξόδου στοιχείων, δηλαδή τα τηλέτυπα, τις οθόνες, τους εκτυπωτές, τα καταγραφικά (μηχανήματα σχεδίασεως) κλπ.

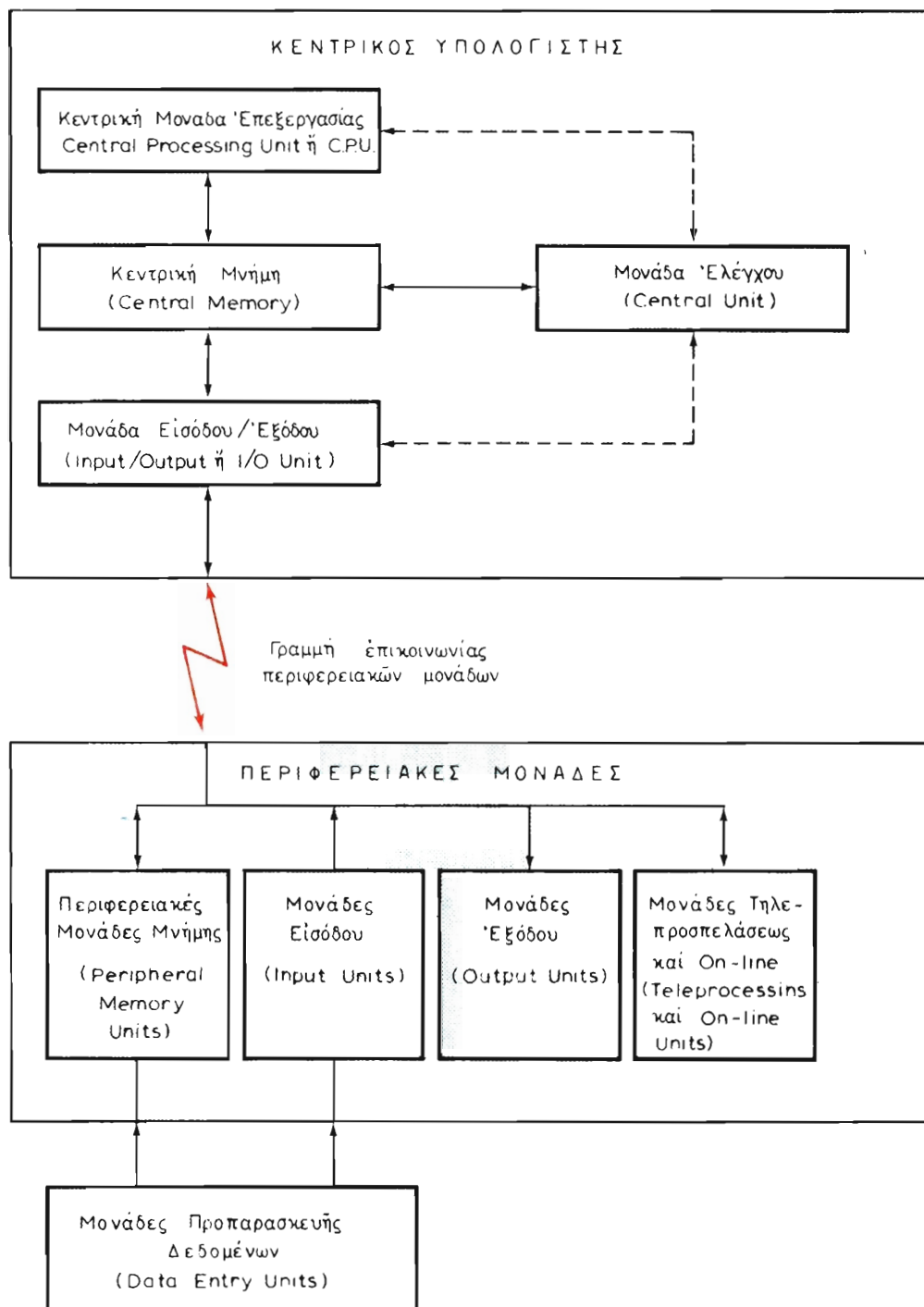
Η καθεμία από αυτές τις μονάδες διαθέτει, ειδική μονάδα ελέγχου (Controller) ή οποία συνεργάζεται με την κεντρική μονάδα επεξεργασίας του υπολογιστή μέσω της μονάδας εισόδου/εξόδου.

Στό σχήμα 7.1α φαίνεται διάγραμμα γενικής οργάνωσης ηλεκτρονικού υπολογιστή. Όπως παρατηρούμε στο σχήμα, οι διάφορες περιφερειακές μονάδες συνδέονται με τον υπολογιστή μέσω γραμμών επικοινωνίας. Μπορεί να βρίσκονται όλες δίπλα στον υπολογιστή ή να είναι μακριά από αυτόν. Στην περίπτωση που οι περιφερειακές μονάδες βρίσκονται μακριά από τον υπολογιστή, (π.χ. σε άλλο κτίριο ή σε άλλη πόλη), τότε λέμε ότι έχουμε «Τηλεπροσπέλαση» (Remote Access).

Η γραμμή επικοινωνίας σ' αυτή την περίπτωση μπορεί να είναι μία τηλεφωνική γραμμή, μία τηλεγραφική γραμμή, μία ασύρματη ζεύξη ή μία ειδική γραμμή, π.χ. γραμμή οπτικών ινών (Fibers).

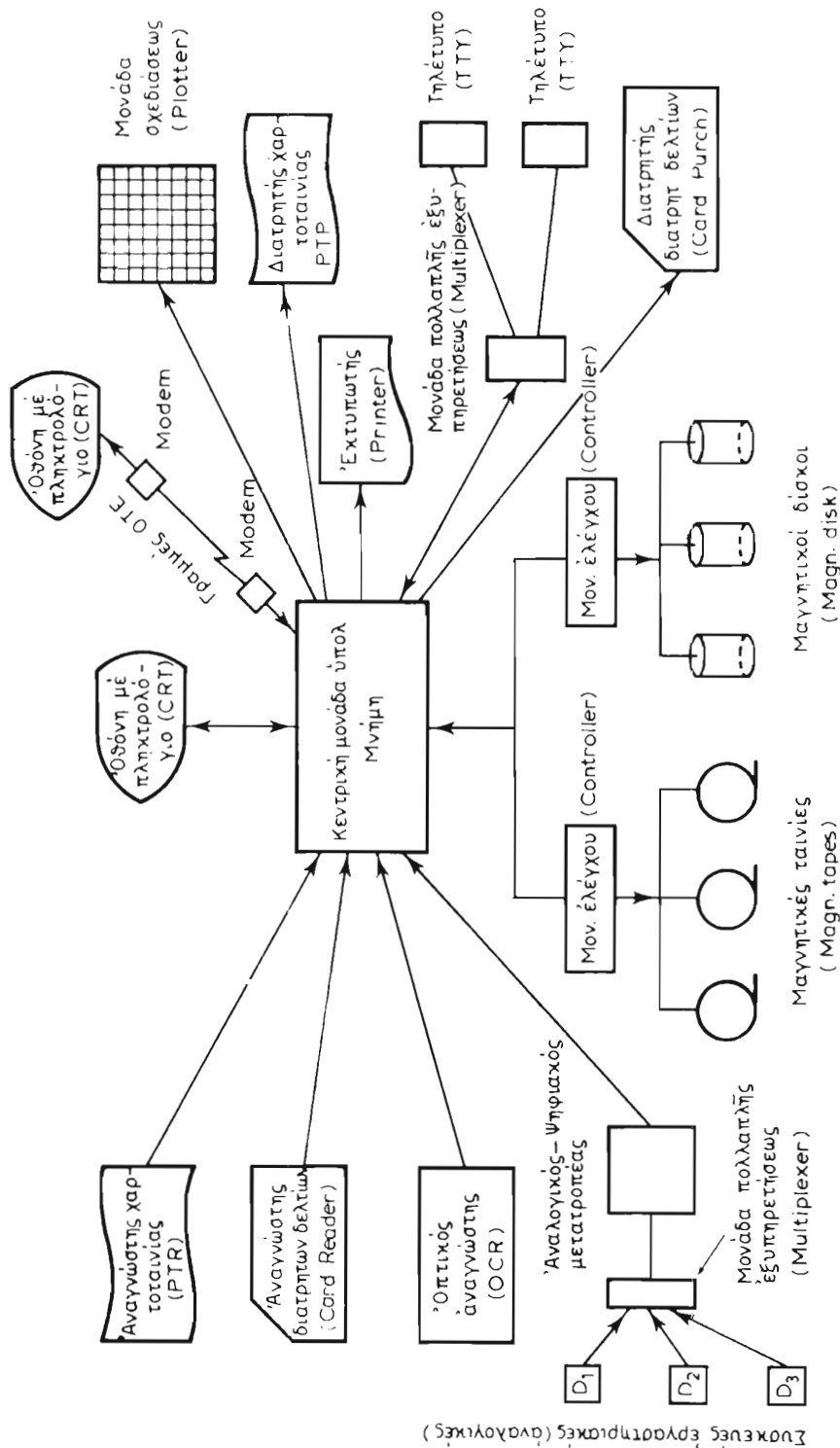
Όταν έχουμε τηλεπροσπέλαση, τότε μεταξύ της περιφερειακής μονάδας καί του υπολογιστή παρεμβάλλονται ειδικές συσκευές οι οποίες ονομάζονται «**διαμορφωτές - αποδιαμορφωτές**» (Modems). Αυτές μετατρέπουν τα δεδομένα σε κατάλληλη μορφή για μετάδοση μέσω των γραμμών επικοινωνίας. Υπάρχει επίσης περίπτωση ομαδοποίησης ορισμένων ομοίων περιφερειακών μονάδων όπως π.χ. μονάδες μαγνητικών ταινιών ή μαγνητικών δίσκων, όποτε η επικοινωνία τους με την κεντρική μονάδα του υπολογιστή γίνεται μέσω κοινής γραμμής. Στο σχήμα 7.1β φαίνεται η διάρθρωση ενός πραγματικού συστήματος Η/Υ καί ο τρόπος συνδέσεως των διαφόρων περιφερειακών μονάδων με τον υπολογιστή.

Μπορούμε να έχουμε ένα κεντρικό ηλεκτρονικό υπολογιστή στον οποίο συνδέονται οι διάφορες περιφερειακές μονάδες όπως φαίνεται στο σχήμα 7.1β· μπορούμε όμως να έχουμε καί έναν υπολογιστή ο οποίος συνδέεται με κάποιο άλλο υπολογιστή για την ανταλλαγή πληροφοριών καί δεδομένων. Τότε λέμε ότι έχουμε



Σχ. 7.1α.

Διάγραμμα της γενικής οργάνωσης ηλεκτρονικού υπολογιστή.



Σχ. 7.1β.

Διάγραμμα ενός ηλεκτρονικού υπολογιστή με τις βασικές του περιφερειακές μονάδες.

ένα «**δίκτυο υπολογιστών**» (computer network). Σ' αυτή την περίπτωση ένας υπολογιστής μπορεί νά «χρησιμοποιεί» τις περιφερειακές μονάδες ενός άλλου υπολογιστή.

7.2 Τηλέτυπα (TTY).

Τό τηλετύπο (teletype ή TTY) είναι από τις πιο κοινές συσκευές εισόδου - εξόδου. Αποτελείται από έναν έκτυπωτή χαρτιού σχετικά μικρής ταχύτητας και από ένα άλφαριθμητικό πληκτρολόγιο. Τό τηλετύπο χρησιμοποιείται για την αποστολή ή λήψη πληροφοριών προς ή από τόν υπολογιστή.

Τά τηλετύπα (σχ. 7.2) χρησιμοποιούνται συνηθέστερα σέ συστήματα μίνι ή μικρο-υπολογιστών. Επίσης χρησιμοποιούνται ως απομακρυσμένες περιφερειακές μονάδες εισόδου - εξόδου στοιχείων σέ συστήματα Η/Υ άπευθείας επικοινωνίας (ON-LINE). Συνδέονται μέ τόν υπολογιστή μέ τίς γραμμές τηλεπικοινωνίας. Ή μεταβίβαση τών δεδομένων από τό τηλετύπο πρós τόν υπολογιστή καί αντίστροφα, γίνεται χαρακτήρα πρós χαρακτήρα.



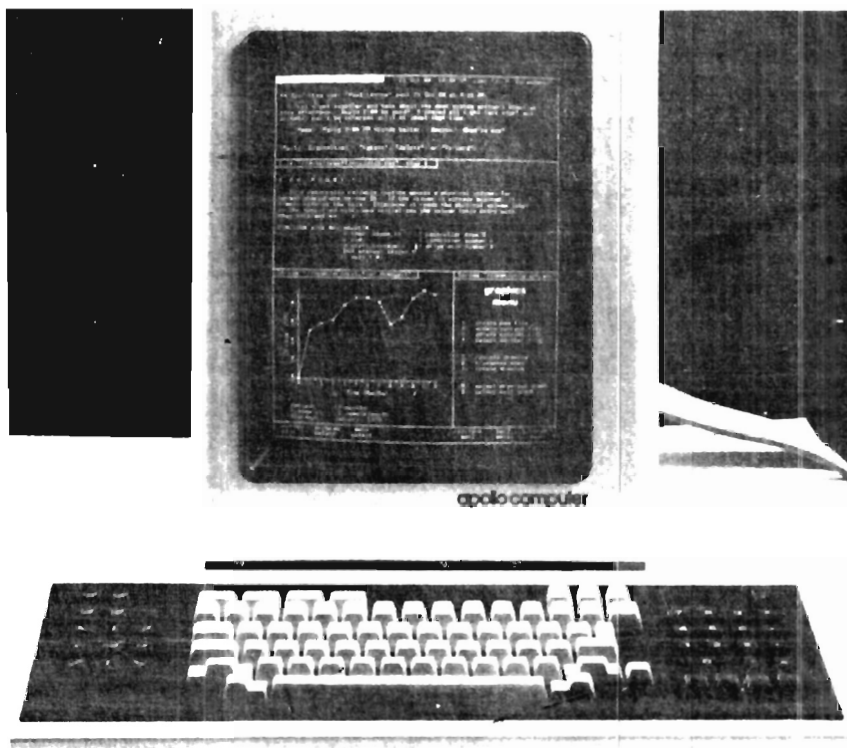
Σχ. 7.2.
Τηλέτυπο.

Λόγω τής χαμηλής του ταχύτητας (10 - 30 χαρακτήρες/sec) τό τηλετύπο σπάνια χρησιμοποιείται ως αύτοτελής μονάδα εισόδου - εξόδου. Συνήθως χρησιμοποιείται γιά τή «συννομιλία» χειριστή καί τοῦ ηλεκτρονικοῦ υπολογιστή. Στά τηλετύπα χρησιμοποιείται, συνήθως, ό κώδικας ASCII.



Σχ. 7.3α.

Όθονη, προβολής άλφαριθμητικῶν χαρακτήρων μόνο.



Σχ. 7.3β.

Όθονη προβολής άλφαριθμητικῶν χαρακτήρων καὶ καμπυλῶν.

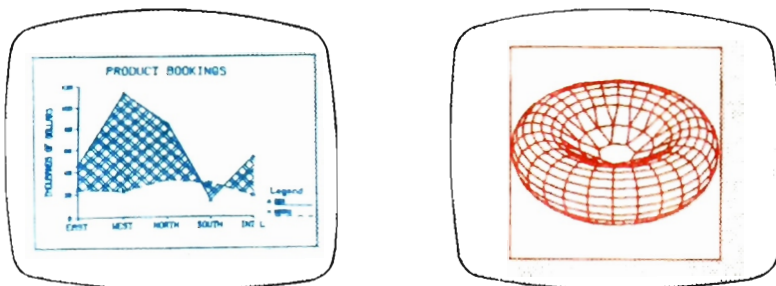
7.3 Όθόνες (Cathode Ray Tubes - CRT).

Ἡ μονάδα αὐτὴ ἀποτελεῖται ἀπὸ μία ὀθόνη καθοδικοῦ σωλήνα, ὅπως ἡ ὀθόνη τῆς τηλεοράσεως καὶ ἀπὸ ἓνα πληκτρολόγιο μὲ άλφαριθμητικούς χαρακτήρες. Ἡ

όθονη έχει συνήθως, διάμετρο από 6" ως 12" και ο αριθμός των χαρακτήρων που μπορούν να προβληθούν επάνω της είναι ανάλογος με το μέγεθός της.

Οι μονάδες αυτές χρησιμοποιούνται και στο κεντρικό συγκρότημα ενός υπολογιστή, αλλά και ως περιφερειακές τερματικές μονάδες (terminal).

Υπάρχουν, κυρίως, δύο τύποι: οι όθόνες στις οποίες προβάλλονται μόνο αλφαριθμητικοί χαρακτήρες (σχ. 7.3α) και οι όθόνες στις οποίες έχουμε τη δυνατότητα προβολής σχημάτων ή διαγραμμάτων (graphics) (σχ. 7.3β). Επίσης υπάρχουν όθόνες μαυράσπρες και όθόνες έγχρωμες (σχ. 7.3γ).



Σχ. 7.3γ.

Φωτογραφίες διαφόρων γραφικών απεικονίσεων σε έγχρωμη όθονη CRT.

Η είσοδος των πληροφοριών στον υπολογιστή γίνεται από το πληκτρολόγιο της όθονης. Οι πληροφορίες αυτές τοποθετούνται με την πληκτρολόγηση σε μία προσωρινή μνήμη (buffer) της συσκευής και ταυτόχρονα εικονίζονται στην όθονη της για όπτικό έλεγχο από το χειριστή. Εάν έχει γίνει κάποιο λάθος, αυτό διορθώνεται από το χειριστή και η λέξη, ή ολόκληρη η γραμμή, ή ακόμη ολόκληρη η σελίδα (ή όθονη γεμής χαρακτήρες) στέλνεται στον υπολογιστή για επεξεργασία. Με ανάλογο τρόπο, κατά την έξοδο των στοιχείων, ο υπολογιστής σχηματίζει κατάλληλα τους χαρακτήρες τους οποίους διαβιβάζει στην όθονη.

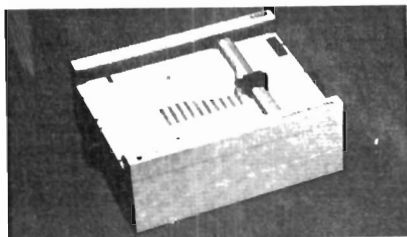
Η ταχύτητα ανταλλαγής πληροφοριών μεταξύ όθονης και υπολογιστή κυμαίνεται από 110 ως 19200 ψηφία/δευτερόλεπτο. Για την επικοινωνία αυτή χρησιμοποιείται συνήθως ο κώδικας ASCII.

7.4 Μονάδες σχεδιάσεως (Plotters) και Μονάδες ψηφιοποίησης (Digitizers).

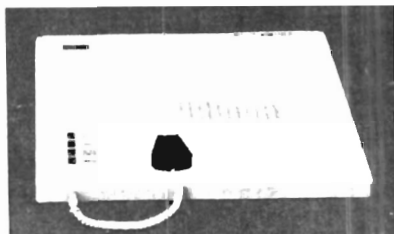
Οι μονάδες σχεδιάσεως, γνωστές και ως καταγραφικά, είναι συσκευές έξοδου που χρησιμοποιούνται κυρίως για το σχεδιασμό γραφικών παραστάσεων και σχεδίων [σχ. 7.4α(α)]. Αντίθετα, οι συσκευές ψηφιοποίησης είναι συσκευές εισόδου και χρησιμοποιούνται αντίστοιχα για την εισαγωγή στον υπολογιστή πληροφοριών που τίς δίνουμε με μορφή σχεδίων ή γραφικών παραστάσεων [σχ. 7.4α(β)]. Πολλές φορές αυτές οι συσκευές συνδυάζονται σε μία και μόνο συσκευή [σχ. 7.4α(γ)].

Υπάρχουν δύο κυρίως είδη καταγραφικών:

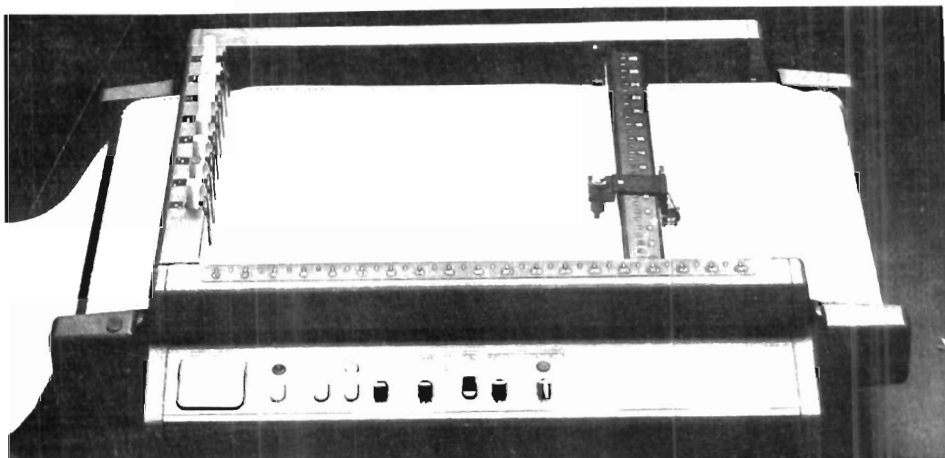
- Τά καταγραφικά με γραφίδα τά οποία σχεδιάζουν πάνω σε χαρτί γραμμές, αποτυπώνουν σημεία και γράφουν γράμματα.



α)



β)



γ)

Σχ. 7.4α.

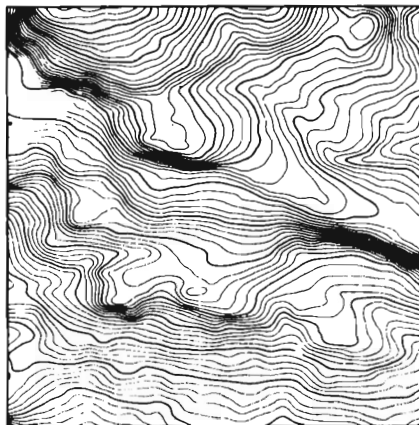
α) Περιφερειακή μονάδα καταγραφής. β) Περιφερειακή μονάδα ψηφιοποίησης. γ) Μικτή μονάδα καταγραφής και ψηφιοποίησης.

- Τά θερμικά (Thermal) καταγραφικά, τά όποια άποτυπώνουν τό σχέδιο μ.ās δ-λόκληρης σελίδας έπάνω σέ είδικό χαρτί.

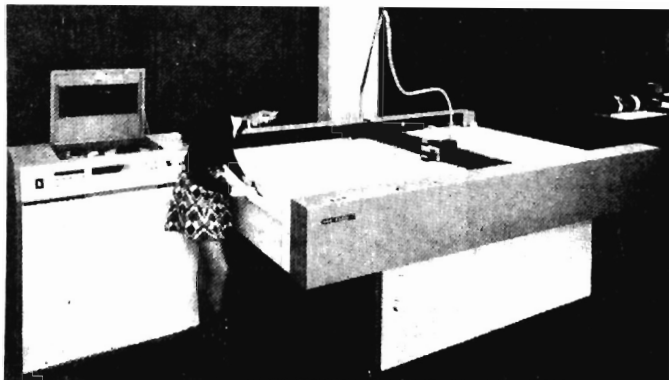
Τά καταγραφικά μικρού και μεσαίου μεγέθους συνδέονται άπευθείας μέ τόν ύπολογιστή. Τά πολύ μεγάλα καταγραφικά, στά όποια μπορούν νά σχεδιάζονται άρχιτεκτονικά, μηχανολογικά κλπ. σχέδια μέ μεγάλες λεπτομέρειες, συνδέονται πρώτα μέ μίνι ύπολογιστή. Σ' ατή τήν περίπτωση, ή όλη έργασία τής σχεδιάσεως έτοιμάζεται σέ ένα μεγάλο ύπολογιστή και τά άποτελέσματα λαμβάνονται σέ μαγνητική ταινία, ή όποία διαβάζεται άπό τό μίνι ύπολογιστή του καταγραφικού (σχεδίαση OFF-LINE).

Στό σχήμα 7.4β φαίνεται τοπογραφικός χάρτης σχεδιασμένος άπό περιφερειακή μονάδα σχεδίασεως και περιφερειακή μονάδα σχεδίασεως.

Οί ψηφιοποιητές αντίστοιχα χρησιμοποιούνται μόνο γιά τήν είσοδο στοιχείων, τά όποια πρόκειται είτε νά έπεξεργασθουν άπό τόν ύπολογιστή, είτε νά άποθηκευθουν στή μνήμη του.



α



β

Σχ. 7.4β.

α) Τοπογραφικός χάρτης σχεδιασμένος από περιφερειακή μονάδα σχεδιάσεως. β) Περιφερειακή μονάδα σχεδιάσεως (Plotter).

Η εισαγωγή των στοιχείων γίνεται με τη βοήθεια μίας ειδικής ακίδας ή οποία περνά πάνω από κάθε γραμμή του σχεδίου, δίνοντας έτσι ως πληροφορίες τις συντεταγμένες της θέσεώς της, καθώς αυτή διαγράφει τις γραμμές του σχεδίου.

7.5 Έκτυπωτές χαρτιού.

Οι έκτυπωτές χαρτιού χρησιμοποιούνται για να έκτυπώσουν πάνω σε χαρτί δεδομένα και άλλες πληροφορίες που προέρχονται από την κεντρική μονάδα του υπολογιστή.

Διακρίνονται:

- Σέ έκτυπωτές κρούσεως (impact printers) και
- σέ έκτυπωτές χωρίς κρούση (non-impact printers).

Οι έκτυπωτές κρούσεως έκτυπώνουν τούς διάφορους χαρακτήρες πάνω στο χαρτί με μηχανικό τρόπο (όπως π.χ. μία κοινή γραφομηχανή), ενώ οι έκτυπωτές χωρίς κρούση έκτυπώνουν τούς χαρακτήρες πάνω σε ειδικό χαρτί, συνήθως θερμικό.

Οι έκτυπωτές διακρίνονται επίσης:

- Σέ έκτυπωτές χαρακτήρων,
- σέ έκτυπωτές γραμμών καί
- σέ έκτυπωτές σελίδων.

Η βασική τους διαφορά είναι στην ταχύτητα λειτουργίας τους. Οι έκτυπωτές χαρακτήρων έκτυπώνουν τόν ένα χαρακτήρα μετά τόν άλλο, οι έκτυπωτές γραμμών έκτυπώνουν ταυτόχρονα μία ολόκληρη γραμμή ενώ οι έκτυπωτές σελίδων έκτυπώνουν ταυτόχρονα μία ολόκληρη σελίδα.

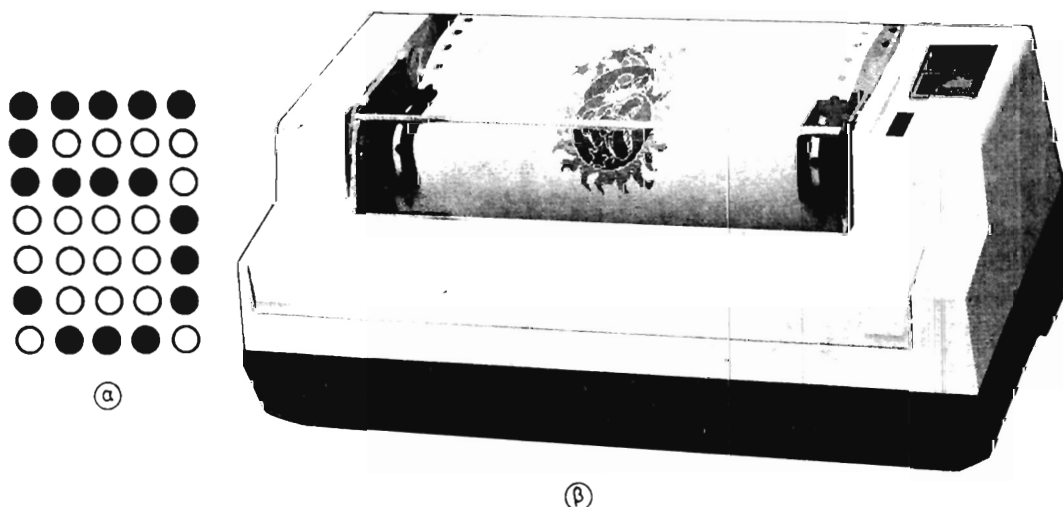
Οι ταχύτητες έκτυπώσεως είναι:

- Για τούς έκτυπωτές χαρακτήρων από 30 ως 300 χαρακτήρες/1".
- Για τούς έκτυπωτές γραμμών από 300 ως 5000 γραμμές τών 132 χαρακτήρων /1".

Γιά τούς έκτυπωτές σελίδων ή ταχύτητα έκτυπώσεως είναι πολύ μεγαλύτερη.

7.5.1 Έκτυπωτές χαρακτήρων (Character Printers).

Οι έκτυπωτές αυτοί μπορεί να είναι έκτυπωτές κρούσεως ή έκτυπωτές χωρίς κρούση. Στούς έκτυπωτές με κρούση έχουμε έκτύπωση ολοκλήρου του γράμματος ως συνόλου όπως στη γραφομηχανή, ή έκτύπωση με τη βοήθεια μιας μήτρας από λεπτές βελόνες διατεταγμένες σε ορθογώνιο σχήμα συνήθως 5×7 ή 7×9 . Ο υπολογιστής ενεργοποιεί κατάλληλα τις βελόνες ανάλογα με το γράμμα που θέλουμε να γραφεί. Οι έκτυπωτές σίτιοι λέγονται βελονωτοί έκτυπωτές (Needle ή Matrix Printers). Στο σχήμα 7.5α φαίνεται απεικόνιση χαρακτήρα με το σύστημα βελονωτού έκτυπωτή (Dot - Matrix) καί μονάδα έκτυπώσεως του ίδιου έκτυπωτή.



Σχ. 7.5α.

α) Απεικόνιση χαρακτήρα με το σύστημα βελονωτού - έκτυπωτή (Dot Matrix). β) Μονάδα έκτυπώσεως βελονωτού έκτυπωτή.

Οι έκτυπωτές, με κρούση σέ σύγκριση μέ τούς άλλους έκτυπωτές εἶναι ταχύτεροι στή λειτουργία. Ἡ ποιότητα ὁμως τῶν γραμμάτων τους εἶναι σχετικά χαμηλή (σχ. 7.5β). Μεταξύ τῆς έκτυπωτικῆς κεφαλῆς καί τοῦ χαρτιοῦ πού έκτυπώνεται ἡ πληροφορία, μεσολαβεῖ εἰδική μελανοταινία ἡ ὁποία μέ τό κτύπημα ἀποτυπώνει τήν πληροφορία στό χαρτί.

LIST

```

1000 SFMODE
1010 INTEGER I,J,K,L,M
1020 OPEN\1,100\ "AF"
1030 INPUT I,J,K
1040 FOR L=I TO J
1050   FOR M=J TO K
1060     A=0.5*L
1070     B=1.5*M
1080     C=A+B**2
1090     PUT\1\I,J,K,A,B,C
1100   PRINT I,J,K,A,B,C
1110   NEXT M
1120 NEXT L
1130 CLOSE\1\
1140 END

```

Σχ. 7.5β.

Δεῖγμα έκτυπώσεως ἀπό βελονωτό έκτυπωτή.

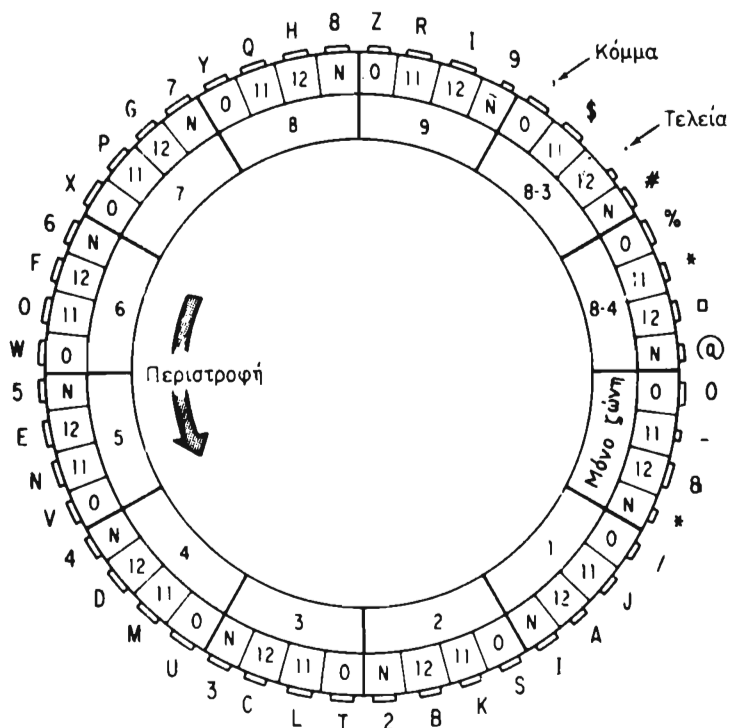
Οι έκτυπωτές χωρίς κρούση έκτυπώνουν τούς χαρακτήρες πάνω σέ εἰδικό χαρτί μέ θερμικό, χημικό ἢ ἄλλο τρόπο. Στούς έκτυπωτές αὐτούς οἱ ταχύτητες έκτυπώσεως εἶναι συνήθως μικρές. Ἄν ὁμως γιά τήν έκτύπωση χρησιμοποιεῖται τεχνολογία δέσμης Laser, τότε οἱ ταχύτητες φθάνουν σέ πολύ μεγάλες τιμές.

7.5.2 Ἐκτυπωτές γραμμῶν.

Οι έκτυπωτές γραμμῶν πού εἶναι έκτυπωτές κρούσεως, διακρίνονται σέ **έκτυπωτές μέ δίσκους**, σέ **έκτυπωτές μέ ἀλυσίδα** καί σέ **έκτυπωτές μέ τύμπανο**.

α) Ἐκτυπωτές μέ δίσκους (Wheel Printers).

Ἡ έκτυπωτική κεφαλὴ σ' αὐτούς ἀποτελεῖται ἀπό πολλούς (συνήθως 120) ὁμό-



Σχ. 7.5γ.

“Ένας από τούς δίσκους του έκτυπωτή με δίσκους.

κεντρους δίσκους πάνω σ’ ένα περιστρεφόμενο άξονα. Κάθε δίσκος έχει στην περιφέρειά του ανάγλυφους τούς χαρακτήρες έκτυπώσεως (σχ. 7.5γ). Μεταξύ της κεφαλής και του χαρτιού μεσολαβεί και πάλι μελανοταινία. Πίσω από τό χαρτί υπάρχουν ειδικά ηλεκτρομηχανικά σφυριά τά όποια κτυπώντας τό χαρτί αποτυπώνουν τούς χαρακτήρες έπάνω του.

‘Η ταχύτητα αύτών των έκτυπωτών είναι γύρω στίς 150 μέ 200 γραμμές ανά λεπτό.

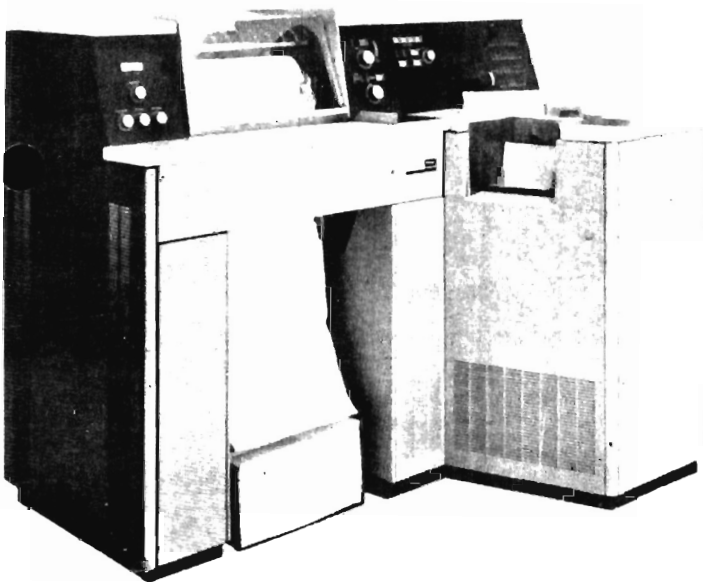
β) Έκτυπωτές με άλυσίδα (σχ. 7.5δ).

Τά γράμματα βρίσκονται ένάγλυφα πάνω σέ μία κλειστή μεταλλική ή πλαστική άλυσίδα. Συνήθως, ή άλυσίδα χωρίζεται σέ 2 ως 5 τμήματα από τά όποια τό καθένα περιλαμβάνει όλους τούς χαρακτήρες πού έκτυπώνει ό έκτυπωτής.

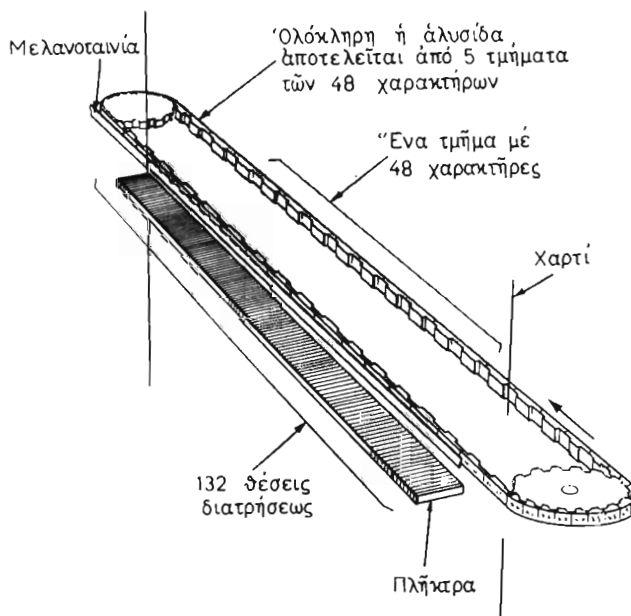
‘Η άλυσίδα περιστρέφεται μέ μεγάλη ταχύτητα και μπροστά από αύτή υπάρχει ή μελανοταινία, ακολουθεί τό χαρτί και πίσω από τό χαρτί μία σειρά ηλεκτρομηχανικών σφυριών.

Όταν ένεργοποιηθεί ένα πλήκτρο, κτυπά τό χαρτί πάνω στή άλυσίδα και αποτυπώνεται έπάνω του ό χαρακτήρας πού βρίσκεται εκείνη τή στιγμή κάτω από τό σφυρί.

Στό σχήμα 7.5ε φαίνεται ό μηχανισμός έκτυπώσεως έκτυπωτή μέ άλυσίδα. ‘Η ταχύτητα των έκτυπωτών αύτών κυμαίνεται από 600 ως 2000 γραμμές ανά λεπτό.

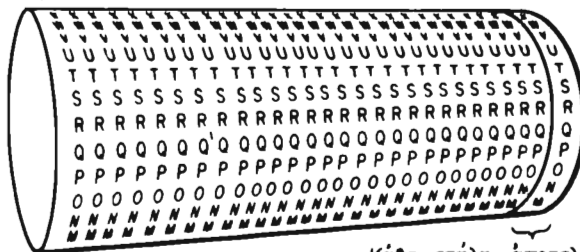


Σχ. 7.5δ.
Έκτυπωτής με άλυσίδα.



Σχ. 7.5ε.
Ο μηχανισμός έκτυπώσεως έκτυπωτή με άλυσίδα.

Ο αριθμός των στηλών αντιστοιχεί
στο μήκος της γραμμής έκτυπωσης
(π.χ. 120 στήλες)



Κάθε στήλη αποτελείται
από όλους τους διαδέσμους
χαρακτήρες έκτυπωσης
(π.χ. 64 χαρακτήρες)

Σχ. 7.5στ.

Ο κύλινδρος έκτυπωσης ενός έκτυπωτή τυμπάνου.

γ) Έκτυπωτές με τύμπανο (Drum Printers).

Οι χαρακτήρες σ' αυτούς είναι ανάγλυφοι πάνω σέ ένα κυλινδρικό τύμπανο όπως φαίνεται στο σχήμα 7.5στ.

Ο μηχανισμός έκτυπωσης είναι παρόμοιος με τούς προηγούμενους δύο έκτυπωτές. Η ταχύτητα έκτυπωσης είναι γύρω στις 1500 γραμμές ανά λεπτό.

7.5.3 Έκτυπωτές σελίδων.

Οι έκτυπωτές σελίδων αποτελούν σήμερα την τελευταία λέξη της τεχνολογίας και λειτουργούν με βάση τις ακτίνες Laser. Οι ταχύτητες έκτυπωσης αυτών των έκτυπωτών είναι πολύ μεγάλες.

7.6 Άλλες περιφερειακές μονάδες.

Εκτός από τις βασικές μονάδες που περιγράφηκαν στίς προηγούμενες παραγράφους, υπάρχουν και άλλες περιφερειακές μονάδες που βοηθούν στην είσοδο και στην έξοδο των στοιχείων στον υπολογιστή.

Τέτοιες μονάδες είναι:

α) **Ο οπτικός αναγνώστης (optical reader).** Είναι μία συσκευή που μπορεί να διαβάσει τυπωμένους χαρακτήρες ή ειδικά σύμβολα γραμμένα με τό χέρι.

β) **Ο αναγνώστης μαγνητικής μελάνης (MICR).** Είναι μία συσκευή που διαβάει χαρακτήρες ειδικού σχήματος, γραμμένους με μαγνητική μελάνη.

γ) **Η ακίδα αναγνώσεως πληροφοριών υπό μορφή στηλών (bar code reader).** Χρησιμοποιείται για την είσοδο στοιχείων τιμών μεγάλων καταστημάτων στον υπολογιστή. Η μονάδα αυτή μοιάζει με μολύβι τό όποιο στήν άκρη του έχει μία φωτοδίοδο που συνδέεται με τόν υπολογιστή. Όταν με αυτό σαρώσουμε ένα σύνολο από παράλληλες γραμμές τοποθετημένες με ειδικό κώδικα, θά εισαχθούν οι αντίστοιχες πληροφορίες στον υπολογιστή.

Σήμερα έχουν κυκλοφορήσει οι πρώτες συσκευές που έπιτρέπουν την εισαγω-

γή πληροφοριών στον υπολογιστή με την ανθρώπινη φωνή, καθώς και τη λήψη των πληροφοριών από τον υπολογιστή με φωνή που μοιάζει με την ανθρώπινη. Βέβαια θα απαιτηθεί πολύς καιρός για να ελαττωθεί το μεγάλο κόστος των συσκευών αυτών και να βελτιωθεί η ποιότητά τους, έτσι ώστε να γίνουν κατάλληλες για εύρεία χρήση.

7.7 Έρωτήσεις.

1. Ποιό χρησιμοποιούνται συνήθως τα τηλέτυπα και οι οθόνες CRT;
 2. Ποιοί κώδικες επικοινωνίας των τερματικών συσκευών με τον κεντρικό υπολογιστή χρησιμοποιούνται συνήθως;
 3. Τι είναι οι συσκευές ψηφιοποίησης; Ποιό χρησιμοποιούνται;
 4. Σε ποιές κατηγορίες χωρίζονται οι έκτυπωτές χαρτιού;
 5. Ποιά η διαφορά έκτυπωτών χαρακτήρα και βελονωτών έκτυπωτών.
 6. Πώς λειτουργούν οι έκτυπωτές με τύμπανο;
 7. Πώς λειτουργούν οι έκτυπωτές με άλυσίδα;
-

ΚΕΦΑΛΑΙΟ ΟΓΔΟΟ

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΑΝΩΤΕΡΟΥ ΕΠΙΠΕΔΟΥ

8.1 Προγραμματικός εξοπλισμός (Software).

Σέ κάθε υπολογιστή διακρίνουμε τή **Σκευή** (Hardware) καί τόν **Προγραμματικό Έξοπλισμό** (Software). Ὁ ὅρος «Σκευή» (Hardware) ἀναφέρεται στά μηχανικά, ηλεκτρικά καί ηλεκτρονικά μέρη τοῦ υπολογιστή τά ὁποῖα γενικά θά μπορούσαμε νά ποῦμε ὅτι δέν μεταβάλλονται κατά τό χρόνο λειτουργίας του.

Ἀπό αὐτό προήλθε ἡ λέξη Hardware, ἡ ὁποία σέ «ἀκριβή» μετάφραση σημαίνει σκληρός, δύσκαμπος, ἀμετάβλητος.

Ὁ ὅρος «Προγραμματικός εξοπλισμός» (Software) ἀναφέρεται σέ ἕνα σύνολο προγραμμάτων τά ὁποῖα ἀποτελοῦν τό συνδετικό κρίκο μεταξύ τῶν χειριστῶν τοῦ υπολογιστή καί τοῦ Hardware τοῦ υπολογιστή.

Στά ἀρχικά συστήματα Η/Υ κάθε ἐπεξεργασία γινόταν σχεδόν ἀποκλειστικά μέσω εἰδικοῦ Hardware. Ἡ ἀνάπτυξη τῶν ηλεκτρονικῶν κυκλωμάτων καί κυρίως τῶν κυκλωμάτων μνήμης, ἐπέτρεψε τήν χρησιμοποίηση τοῦ Software τό ὁποῖο μετασχηματίζει τίς λειτουργίες τοῦ Hardware ἔτσι, ὥστε αὐτές νά προσαρμόζονται στίς διάφορες ἀπαιτήσεις.

Ἡ ἀνάπτυξη τῶν μεθόδων προγραμματισμοῦ κατά τήν τελευταία 20ετία ἔχει αὐξήσει τήν ταχύτητα συγγραφῆς προγραμμάτων κατά 10 - 100 φορές. Ἡ προσθήκη ἑνός καλοῦ Software ἐπιτρέπει ὀχι μόνο τήν ταχύτερη καί οἰκονομικότερη ἐκτέλεση τῆς ἐργασίας ἀπό ἕναν υπολογιστή ἀλλά καί τή λύση πολλῶν πολυπλόκων προβλημάτων.

Τό Software ἑνός υπολογιστή περιλαμβάνει, σέ βασικές γραμμές, τά παρακάτω:

- Τό Λειτουργικό Σύστημα (Operating System).
- Τούς ἐπεξεργαστές γλωσσῶν, δηλαδή τούς συμβολομεταφραστές (Assemblers), τούς μεταφραστές (Compilers) καί τούς διερμηνεῖς (Interpreters).
- Τά εἰδικά προγράμματα ἐφαρμογῶν.

Ἐκτός ἀπό τούς ὅρους Hardware-Software χρησιμοποιεῖται καί ὁ ὅρος Firmware πού ἀναφέρεται στά μόνιμα ἀποθηκευμένα προγράμματα στή μνήμη τοῦ υπολογιστή σέ ἀντίθεση μέ τό Software πού ἀναφέρεται σέ προγράμματα τά ὁποῖα μποροῦμε νά ἀλλάξουμε.

8.2 Ἱστορική ἐπισκόπηση τοῦ Software.

Πρὶν προχωρήσουμε στήν παρουσίαση τῶν διαφόρων γλωσσῶν προγραμματισμοῦ θά ἦταν χρήσιμη μιά ἱστορική ἐπισκόπηση τῆς ἐξελίξεως τοῦ υπολογιστή.

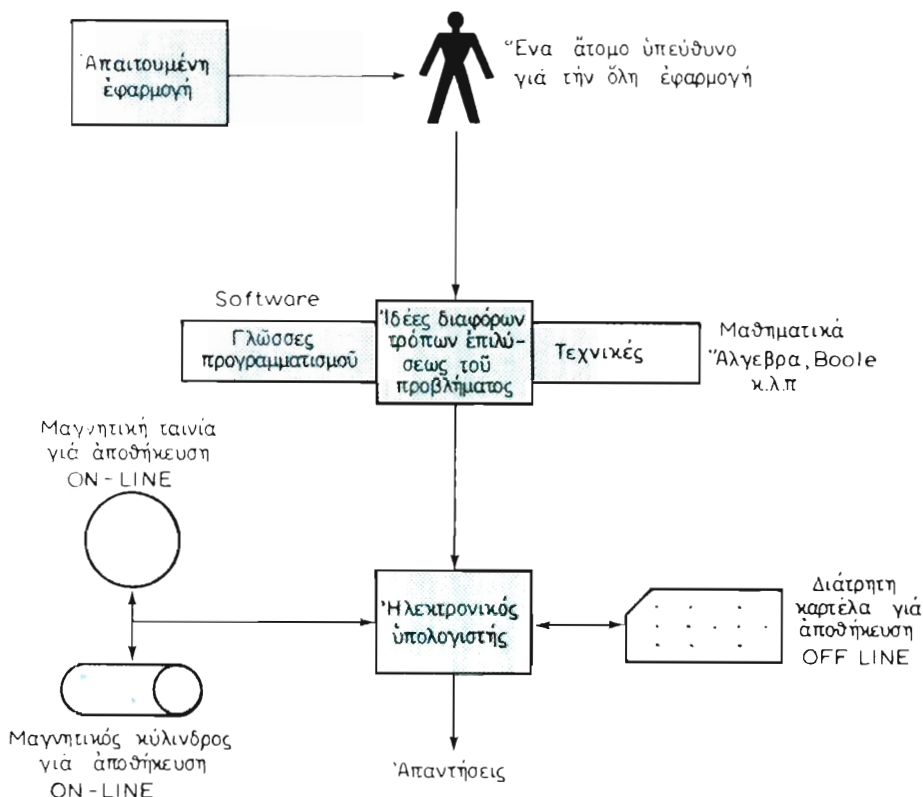
Γενικά, διακρίνουμε τὰ ἑξῆς στάδια:

- 1ο στάδιο: Ἀνάπτυξη τοῦ Hardware.
- 2ο στάδιο: Ἀνάπτυξη τῶν ἀπαραιτήτων γνώσεων Software καί γλωσσῶν προγραμματισμοῦ.
- 3ο στάδιο: Ἀνάπτυξη τῶν λειτουργικῶν συστημάτων.
- 4ο στάδιο: Αὐτοματισμός στὴν ἀνάπτυξη τῶν συστημάτων.

8.2.1 Πρῶτο στάδιο.

Ἡ ἀνάπτυξη καί ἡ κατασκευὴ τῶν πρώτων ὑπολογιστῶν προήλθε ἀπὸ τὴν ἀνάγκη ἐπιλύσεως πολυπλόκων μαθηματικῶν προβλημάτων. Οἱ πρώτοι πού χρησιμοποίησαν ἠλεκτρονικούς ὑπολογιστές ἦταν μαθηματικοὶ ἢ μηχανικοί, γιὰ τοὺς ὁποίους ὁ προγραμματισμός ἦταν μιά ἀναγκαία ἀλλὰ καθόλου εὐχάριστη ἐργασία. Βέβαια σὲ τοῦτο συνέτεινε καί τὸ ὅτι οἱ πρώτες γλώσσες προγραμματισμοῦ ἦταν δύσκολες.

Στὸ πρῶτο αὐτὸ τὸ στάδιο ἓνας προγραμματιστὴς ἀναλάμβανε πλήρως μιά ἐφαρμογή. Κάθε πρόγραμμα ἦταν ἀποτέλεσμα προσπαθειῶν ἐνὸς καί μόνο προγραμματιστῆ. Αὐτὸς ὁ τρόπος ἐργασίας φαίνεται καλύτερα στὸ σχῆμα 8.2α.



Σχ. 8.2α.

Πρῶτο στάδιο ἐξελίξεως Hardware - Software.

“Ένα άλλο χαρακτηριστικό στοιχείο, κατά το πρώτο στάδιο της εξέλιξης, είναι ότι τα διάφορα προβλήματα, τα οποία επιλύονταν με τη βοήθεια του ηλεκτρονικού υπολογιστή δεν ήταν δύσκολα και ακολουθούσαν για τη λύση τους μία κάποια τυποποιημένη, θά έλεγε κανείς, μέθοδο.

Το πρώτο μεγάλο βήμα για την αναγνώριση της σπουδαιότητας του προγραμματισμού ως ιδιαίτερου επαγγελματικού κλάδου έγινε όταν άρχισαν οι έμπορικές εφαρμογές.

Οι άνθρωποι οι οποίοι δούλευαν με τις μηχανές αυτές δεν ήξεραν τί θά πή αλγεβρα Βοολε ή αριθμητική ανάλυση. Γνώριζαν σέ γενικές γραμμές το πρόβλημά τους αλλά δεν γνώριζαν πώς θά τό επιλύσουν. Έτσι, γι’ αυτές τις εφαρμογές θά έπρεπε νά ακολουθήσουν τά εξής στάδια:

- Λεπτομερή καθορισμό του προβλήματος.
- Περιγραφή συστηματικού τρόπου για την επίλυση του προβλήματος.
- Μετατροπή της περιγραφής αυτής σέ πρόγραμμα.

Γιά νά ανταπεξέλθει στίς δυσκολίες τών έμπορικών εφαρμογών, ο προγραμματιστής προσπάθησε νά βρεί τρόπους πιά εύκολου προγραμματισμού τών εφαρμογών. Κατ’ αρχήν χρησιμοποίησε τό δεκαδικό σύστημα αριθμώσεως καί έτσι δημιουργήθηκαν προγράμματα τά όποια μετέτρεπαν τούς δεκαδικούς αριθμούς σέ δυαδικούς καί αντίστροφως. Έπειδή ο προγραμματιστής ήθελε πιά εύκολο καί πιά αποτελεσματικό τρόπο προσπελάσεως στά αρχεία τά όποια είχε αποθηκεύσει στόν υπολογιστή (εύκολο τρόπο εισόδου καί έξόδου τών στοιχείων από τόν ηλεκτρονικό υπολογιστή), οι κατασκευαστές Η/Υ δημιούργησαν τις συμβολικές γλώσσες καί τούς συμβολομεταφραστές.

“Ας σημειωθεί ότι τά προγράμματα σ’ αυτό τό στάδιο εξέλιξης τών Η/Υ δεν αποθηκεύονταν στή μνήμη του υπολογιστή.

8.2.2 Δεύτερο στάδιο.

Μετά την εμφάνιση μεγαλύτερων σέ υπολογιστική δύναμη υπολογιστών προέκυψε ή ανάγκη επιλύσεως πολυπλοκοτέρων προβλημάτων.

Τά προγράμματα κατά τό στάδιο αυτό δεν αποθηκεύονται έξω από τόν υπολογιστή. Δημιουργείται ειδική βιβλιοθήκη προγραμμάτων μέσα στόν υπολογιστή στήν όποια αποθηκεύονται καί από τήν όποια καλούνται καί έκτελούνται από τούς χρήστες.

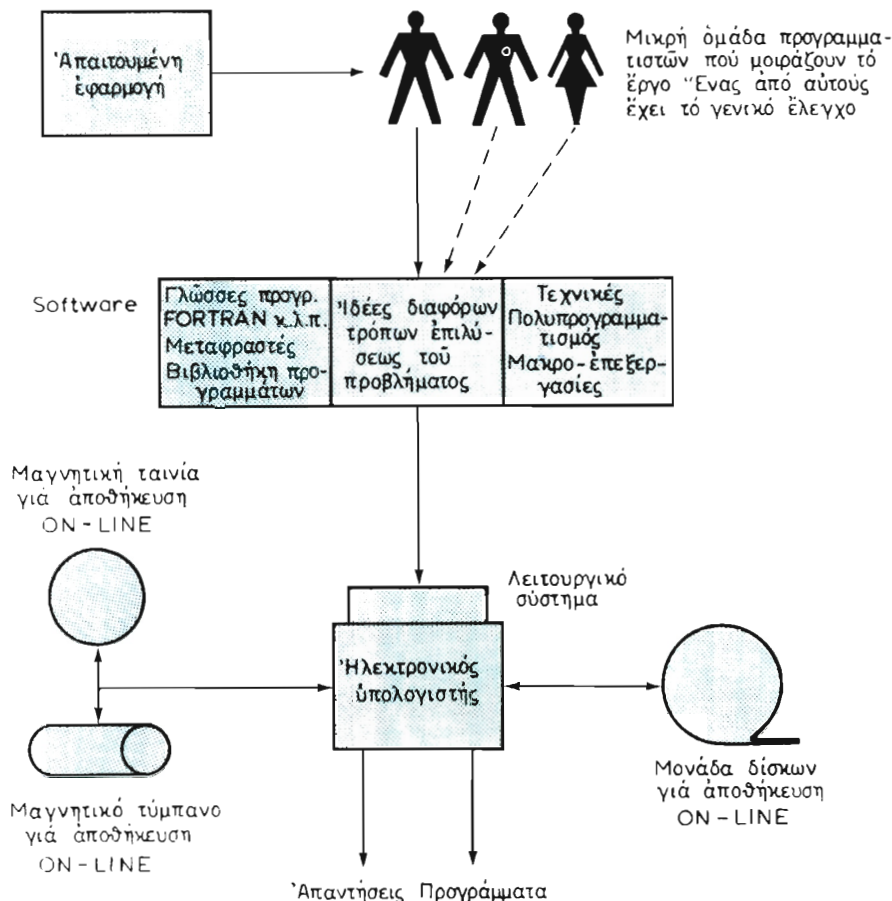
Έπίσης οι γλώσσες προγραμματισμού αναπτύσσονται. Ένώ μέχρι τώρα τά προγράμματα έπρεπε νά γράφονται σέ γλώσσα μηχανής ή σέ συμβολική γλώσσα, τώρα οι νέες γλώσσες άνωτέρου επιπέδου επιτρέπουν στόν προγραμματιστή νά χρησιμοποιεί μία γλώσσα πιά εύχρηστη καί προσαρμοσμένη στίς ανάγκες του προβλήματος. Η γλώσσα Fortran π.χ. είναι πιά εύκολη στήν εκμάθησή της άπο τή συμβολική γλώσσα καί άρκετά εύχρηστη σέ μαθηματικές εφαρμογές.

Τώρα όμως άπαιτείται ή ύπαρξη ενός προγράμματος μεταφραστή ή όποιος θά μετατρέψει τήν παραπάνω γλώσσα σέ κώδικα αναγνωρίσιμο από τόν υπολογιστή.

Στό σχήμα 8.2β φαίνεται σχηματικά τό δεύτερο στάδιο εξέλιξης τών Η/Υ.

8.2.3 Τρίτο στάδιο.

Βρισκόμαστε στή δεκαετία του ‘60. Η εξέλιξη τών ηλεκτρονικών υπολογιστών



Σχ. 8.2β.

Δεύτερο στάδιο εξέλιξης Hardware - Software.

συνεχίζεται και τα εύεργετήματα από τη χρήση των ηλεκτρονικών υπολογιστών είναι πλέον φανερά. Όπως φανερά είναι επίσης και η ανάγκη υπάρξεως περισσότερο αποδοτικών μεθόδων προγραμματισμού και καλύτερων προγραμματιστών. Τουτό επιτεύχθηκε μέ:

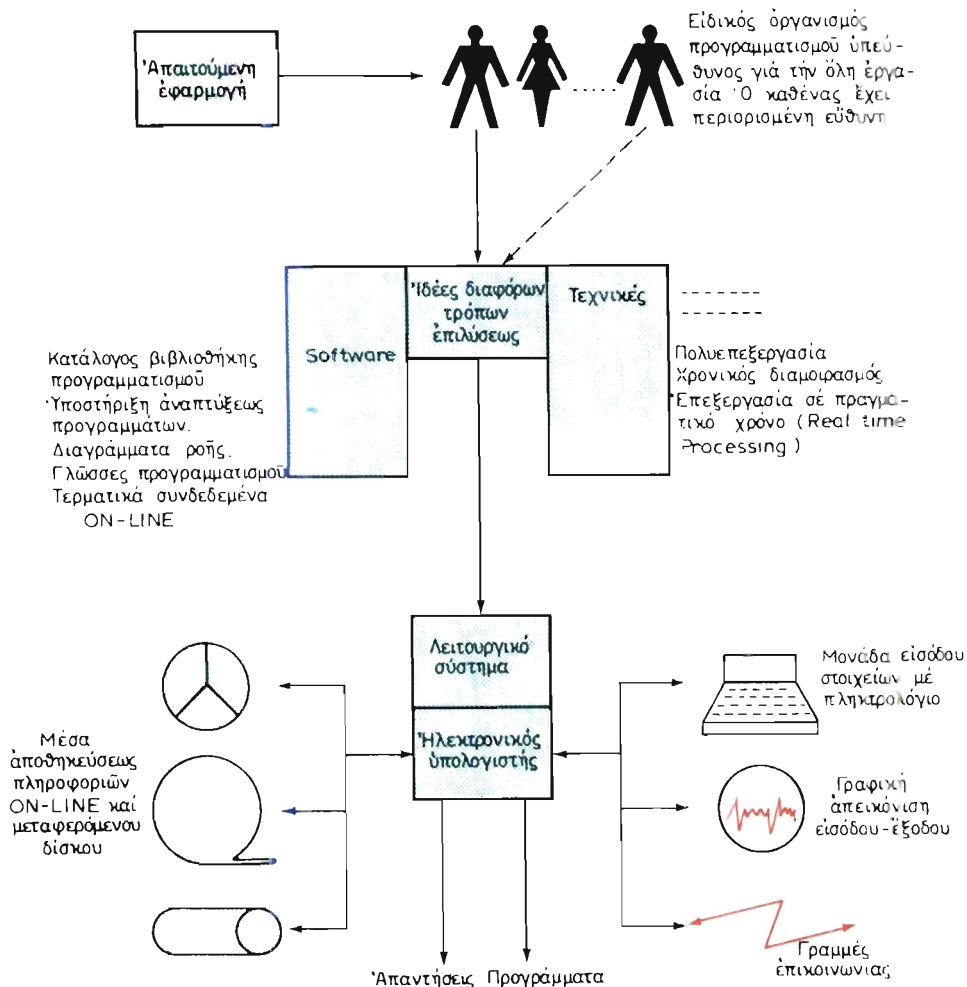
- Τήν καλύτερη και αποδοτικότερη εκπαίδευση στον προγραμματισμό.
- Τή δημιουργία πιδό αποδοτικών προγραμμάτων.
- Τή συνεργασία των διαφόρων οργανισμών.
- Τή μείωση του κόστους των υπολογιστών.

Σ' αυτό τό στάδιο έχομε πλέον τεχνολογικές εξελίξεις λόγω των νέων ηλεκτρονικών εξαρτημάτων και περιφερειακών μονάδων και λόγω τής καλύτερης γνώσεως των απαιτήσεων στήν επεξεργασία των πληροφοριών.

Άλλά και ό τρόπος και οι γλώσσες προγραμματισμού παρουσιάζουν εξέλιξη. Παρατηρείται ή εμφάνιση των γλωσσών άνωτέρου επιπέδου, Basic, Algol, Cobol κλπ. αλλά και ή δυνατότητα καλύτερου προγραμματισμού λόγω αναπτύξεως των

μεθόδων μετρήσεως αποδόσεως των διαφόρων προγραμμάτων. Επίσης λόγω της αναπτύξεως των λειτουργικών συστημάτων των υπολογιστών, έχουμε δυνατότητες ταυτόχρονης χρησιμοποίησεως ενός υπολογιστή από πολλούς χειριστές (time sharing) έτσι, ώστε ο καθένας να νομίζει ότι ο υπολογιστής εργάζεται αποκλειστικά για αυτόν.

Τά κύρια χαρακτηριστικά εξέλιξεως των Η/Υ κατά τό τρίτο στάδιο φαίνονται στό σχήμα 8.2γ.



Σχ. 8.2γ.

Τρίτο στάδιο Έξελιξεως Hardware - Software.

8.2.4 Τέταρτο στάδιο.

Τό στάδιο αυτό πού άρχίζει τό 1970 χαρακτηρίζεται άπό:

α) Τή μεγάλη μείωση του κόστους και του όγκου των ηλεκτρονικών υπολογιστών.

β) Τήν ανάπτυξη πιδά αποδοτικών γλωσσών και ιδιαίτερα τήν ανάπτυξη των συστημάτων διαχείρισεως βάσεως πληροφοριών (Data Base Management System ή DBMS).

γ) Τή δημιουργία μεγάλων βάσεων πληροφοριών που προσπελάζονται από πολούς χρήστες.

δ) Τή δημιουργία των πρώτων δικτύων ηλεκτρονικών υπολογιστών.

ε) Τή φοβερή εξάπλωση των μικρο-υπολογιστικών συστημάτων τά όποια λόγω του πολύ μικρού κόστους που έχουν μπορεί νά τά έχει ό καθένας. Γί αυτό και όνομάζονται «προσωπικοί υπολογιστές» (Personal Computers).

Στό σχήμα 8.2δ φαίνεται σχηματικά τό τέταρτο στάδιο εξέλιξεως Η/Υ.

8.3 Προγραμματισμός σέ γλώσσα Μηχανής.

Στούς πρώτους ηλεκτρονικούς υπολογιστές, ό προγραμματιστής έπρεπε άπαιτητάς νά χρησιμοποιήσει τίς δυαδικές έντολές του κάθε υπολογιστή.

Κάθε έντολή του υπολογιστή παριστάνεται από μία σειρά δυαδικών ψηφίων. Π.χ. ή έντολή πολλαπλασίασε τό Α και τό Β μπορεί νά παριστάνεται ως 0111010010001.

Όταν ένα πρόγραμμα γράφεται μέ αυτό τόν τρόπο, δηλαδή όταν γράφομε άπευθείας τίς έντολές του υπολογιστή μέ δυαδικούς (ή όκταδικούς ή δεκαεξαδικούς) άριθμούς, τότε λέμε ότι προγραμματίζομε τόν υπολογιστή σέ γλώσσα μηχανής (machine language). Κατά τόν προγραμματισμό, σέ γλώσσα μηχανής ό προγραμματιστής πρέπει νά όρίζει τίς θέσεις μνήμης στίς όποιες θά άποθηκευθεί τό πρόγραμμα, τά δεδομένα του προβλήματος επάνω στά όποια θά έργασθεί τό πρόγραμμα, καθώς και τά ένδιάμεσα άποτελέσματα, μέ τέτοιο τρόπο, ώστε κάθε νέα έγγραφή στή μνήμη νά μήν καταστρέφει χρήσιμες πληροφορίες.

Μέ αυτό τό τρόπο όμως και ή δουλειά του προγραμματισμού είναι πολύ επίπονη και υπάρχει ό κίνδυνος δημιουργίας πολλών σφαλμάτων τόσο στή διατύπωση των έντολών όσο και στή διανομή τής μνήμης.

8.3.1 Οι έντολές του υπολογιστή και ή εκτέλεσή τους γενικά.

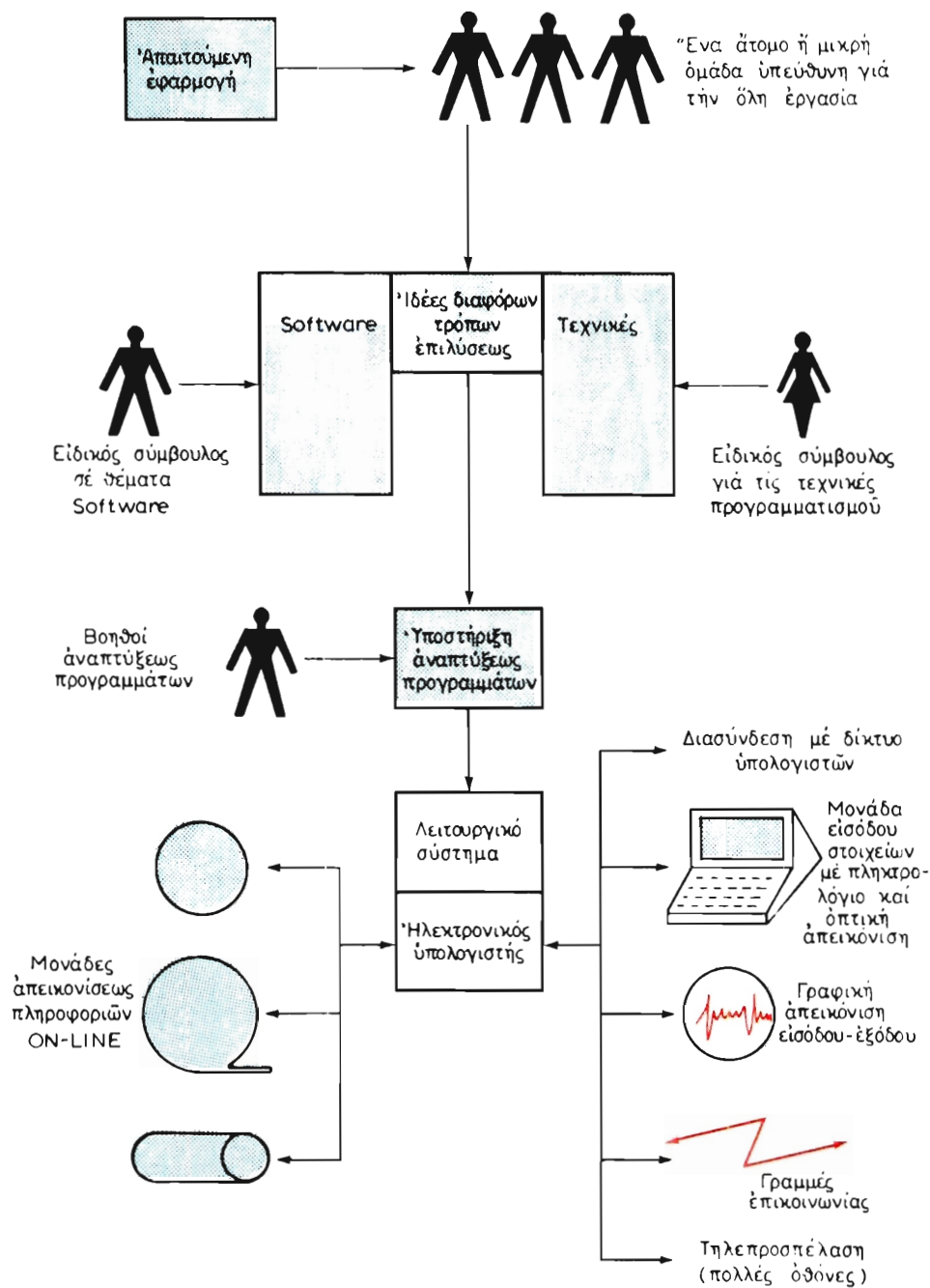
Οι έντολές κάθε προγράμματος άποθηκεύονται σέ διαδοχικές θέσεις μνήμης κατά τή σειρά που θά εκτελεσθούν.

Κάθε έντολή εκτελείται σέ δύο φάσεις:

- Τή φάση άνακλήσεως (Fetch phase).
- Τή φάση εκτελέσεως (Execution phase).

Κατά τή φάση άνακλήσεως ό υπολογιστής τοποθετεί στόν μετρητή προγράμματος τήν διεύθυνση τής πρώτης έντολής και μεταφέρει τόν κωδικό τής έντολής στόν καταχωρητή έντολής τής μονάδας έλέγχου. Κατά τή φάση τής εκτελέσεως ή κωδικοποιημένη έντολή του προγράμματος άποκωδικοποιείται και εκτελείται. Στή συνέχεια τό περιεχόμενο του μετρητή προγράμματος αυξάνεται κατά ένα και έτσι βρίσκεται ή επόμενη έντολή.

Γιά νά δοϋμε πώς γίνεται ή διαδικασία αυτή, άς θεωρήσομε ότι έχομε έναν υπο-



Σχ. 8.26.

Τέταρτο στάδιο εξέλιξης Hardware - Software.

λογιστή που έχει μήκος λέξεως 9 bits, και που έχει τīs έντολές που φαίνονται στόν πίνακα 8.3.1.

Στόν υπολογιστή αυτό θεωρούμε ότι τά τρία πρώτα δυαδικά ψηφία δίνουν τόν κωδικό έντολής και τά υπόλοιπα τό πεδίο διευθύνσεως. Τό σύμβολο Χ στό πεδίο διευθύνσεως, σημαίνει ότι αυτό καθορίζεται από τόν προγραμματιστή, σύμφωνα μέ τīs ανάγκες του προγράμματος.

ΠΙΝΑΚΑΣ 8.3.1.

Έντολές ενός υποθετικού υπολογιστή.

Κωδικός Έντολής	Πεδίο Διευθύνσεως	Περιγραφή έντολής
000	XXXXXX	Πέρασ διαδικασίας. Σ' αυτή τήν έντολή τά τελευταία 6 ψηφία δέν έχουν σημασία.
001	XXXXXX	Νά μεταφερθεί τό περιεχόμενο τής θέσεως μνήμης XXXXXX στό συσσωρευτή. Τό περιεχόμενο αυτού χάνεται και τό περιεχόμενο τής θέσεως μνήμης XXXXXX περιέχεται στή μνήμη και στό συσσωρευτή.
010	XXXXXX	Πρόσθεσε τόν αριθμό που περιέχεται στή θέση μνήμης XXXXXX μέ τό περιεχόμενο του συσσωρευτή. Τό άθροισμα θά παραμείνει στόν συσσωρευτή, ενώ τό περιεχόμενο τής θέσεως μνήμης XXXXXX δέ θά χαθεί (πρόκειται γιά αριθμητικό και όχι λογικό άθροισμα).
011	XXXXXX	Νά μεταφερθεί τό περιεχόμενο του συσσωρευτή στή θέση μνήμης XXXXXX σβήνοντας ό,τι ύπήρχε προηγουμένως σ' αυτή τή θέση. Ό συσσωρευτής συνεχίζει νά περιέχει τό ίδιο περιεχόμενο.
100	XXXXXX	Έκτύπωσε τό περιεχόμενο τής θέσης XXXXXX τής μνήμης στόν έκτυπωτή. Τό περιεχόμενο τής θέσεως μνήμης XXXXXX δέν αλλάζει.
101	XXXXXX	Διάβασε τήν επόμενη έντολή από τήν θέση μνήμης XXXXXX. Αυτό λέγεται «πήδημα» (jump) και υπερχαλύπτει τήν κανονική διαδικασία εκτέλεσεως του προγράμματος.
110	XXXXXX	Διάβασε τήν επόμενη έντολή από τήν θέση μνήμης XXXXXX, ΑΝ ό αριθμός ό αποθηκευμένος στό συσσωρευτή είναι άρνητικός. Άλλιώς συνέχισε τήν κανονική διαδικασία.
111	XXXXXX	Νά αφαιρεθεί τό περιεχόμενο τής θέσεως μνήμης XXXXXX από τό περιεχόμενο του συσσωρευτή. Τό αποτέλεσμα θά παραμείνει στό συσσωρευτή και τό περιεχόμενο τής θέσεως μνήμης XXXXXX θά χαθεί.

8.3.2 Παραδείγματα προγραμματισμού σε γλώσσα μηχανής.

Παράδειγμα 1.

Ποιό είναι το αποτέλεσμα της εκτελέσεως των εντολών με τη σειρά που δίνονται παρακάτω;

```
001010000
010010011
011010000
100010000
000111111
```

Για να μπορέσουμε να αντιληφθούμε τί πρόκειται να γίνει, θα πρέπει να ακολουθήσουμε βήμα προς βήμα τις εντολές όπως ακριβώς θα τις εκτελέσει και ο υπολογιστής.

Με τη βοήθεια του πίνακα 8.3.1 έχουμε:

- 001 010000: Να μεταφερθεί το περιεχόμενο της θέσεως μνήμης 010000 (16₁₀) στον συσσωρευτή. Το περιεχόμενο της θέσεως μνήμης 010000 παραμένει όπως είναι, ενώ ό,τι υπήρχε στο συσσωρευτή χάνεται.
- 010 010011: Πρόσθεσε τον αριθμό που υπάρχει στη θέση μνήμης 010011 (19₁₀) με το περιεχόμενο του συσσωρευτή. Το αποτέλεσμα παραμένει στο συσσωρευτή.
- 011 010000: Να μεταφερθεί το άθροισμα στη θέση μνήμης 010000.
- 100 010000: Εκτύπωσε το αποτέλεσμα του αθροίσματος που βρίσκεται στην θέση μνήμης 010000.
- 000 111111: Πέρας διαδικασίας.

Αν κοιτάξουμε προσεκτικά, εκείνο που κάναμε με τις τρεις αυτές εντολές ήταν να προσθέσουμε τους δύο αριθμούς οι οποίοι περιέχοντο στις θέσεις μνήμης 010000 και 010011 και να τοποθετήσουμε το άθροισμα στη θέση μνήμης 010000.

Τώρα για να γράψουμε αυτό το πρόγραμμα στον υπολογιστή, θα πρέπει να καθορίσουμε σε ποιά διεύθυνση μνήμης τοποθετείται κάθε εντολή. Έτσι το πρόγραμμα στην πλήρη του μορφή θα ήταν ως εξής:

Διεύθυνση μνήμης	Περιεχόμενο μνήμης
000001	001 010000
000010	010 010011
000011	011 010000
000100	100 010000
000101	000 111111

Αν τώρα θελήσουμε να χρησιμοποιήσουμε το παραπάνω πρόγραμμα, δεν έχουμε να κάνουμε τίποτε άλλο παρά να δώσουμε στο μετρητή προγράμματος τη διεύθυνση μνήμης έναρξεως του προγράμματος, δηλαδή τη διεύθυνση 000001 και να ζητήσουμε την έναρξη εκτελέσεώς του.

Παράδειγμα 2.

Έστω ότι θέλουμε να γράψουμε ένα πρόγραμμα σε γλώσσα μηχανής το οποίο υπολογίζει την τιμή της παραστάσεως:

$$x = (a + b + c) - (a_1 + b_1 + c_1)$$

Οι τιμές των a , b , c , a_1 , b_1 , c_1 είναι αποθηκευμένες στις έξι θέσεις μνήμης.

$a \rightarrow 010000$

$b \rightarrow 011000$

$c \rightarrow 011001$

$a_1 \rightarrow 011100$

$b_1 \rightarrow 011110$

$c_1 \rightarrow 010110$

Τό πρόγραμμα μαζί με την ανάλυσή του θα έχει ως εξής:

Διεύθυνση Μνήμης	Έντολή	Επεξήγηση Έντολης
000001	001 010000	Μεταφορά αριθμού a στο συσσωρευτή.
000010	010 011000	Πρόσθεσε τον αριθμό b στο περιεχόμενο του συσσωρευτή.
000011	010 011001	Πρόσθεσε τον αριθμό c στο περιεχόμενο του συσσωρευτή.
000100	011 010101	Νά μεταφερθεί τό άθροισμα $a + b + c$ στή θέση μνήμης 010101.
000101	001 011100	Μεταφορά του αριθμού a_1 στο συσσωρευτή.
000110	010 011110	Πρόσθεσε τον αριθμό b_1 στον a_1 στον συσσωρευτή.
000111	010 010110	Πρόσθεσε τον αριθμό c_1 στο άθροισμα $a_1 + b_1$.
001000	011 100000	Μεταφορά του άθροίσματος $a_1 + b_1 + c_1$ στή θέση μνήμης 100000.
001001	001 010101	Μεταφορά του άθροίσματος $a + b + c$ από τή θέση μνήμης 010101 στον συσσωρευτή.

001010	111 100000	Νά αφαιρεθεί τό άθροισμα $a_1 + b_1 + c_1$ από τόν αριθμό πού περιέχεται στό συσσωρευτή. Τό αποτέλεσμα παραμένει στό συσσωρευτή.
001011	011 101000	Νά μεταφερθεί τό αποτέλεσμα $(a + b + c) - (a_1 + b_1 + c_1)$ από τό συσσωρευτή στη θέση μνήμης 101000.
001100	100 101000	Νά έκτυπωθεί τό αποτέλεσμα τής άφαιρέσεως.
001101	000 000000	Πέρασ διαδικασίας.

8.4 Προγραμματισμός σέ Συμβολική Γλώσσα (Assembly Language Programming).

8.4.1 Γενικά.

Είμαι δυνατό στους κώδικες των έντολών του κύκλου (ρεπερτορίου) έντολών ένός ύπολογιστή νά δοθούν συμβολικά όνόματα τά όποία αποτελούνται από άλφα-βητικά στοιχεία. Μέ τόν τρόπο αυτό είναι εύκολη ή άπομνημόνευση των έντολών από τόν άνθρωπο. Επίσης, έκτός από τίς αριθμητικές διευθύνσεις μπορούμε νά χρησιμοποιήσουμε συμβολικές διευθύνσεις καί συμβολικά όνόματα μεταβλητών. Όταν οι έντολές ένός προγράμματος γράφονται μέ αυτό τόν τρόπο, τότε λέμε ότι τό πρόγραμμα είναι γραμμένο σέ Συμβολική Γλώσσα.

Γιά νά είναι δυνατή ή έκτέλεση ένός προγράμματος γραμμένου στη συμβολική γλώσσα, πρέπει νά υπάρχει δυνατότης μεταφράσεως των συμβολικών όνομάτων των έντολών του προγράμματος, στό δυαδικό κώδικα έντολών καί των συμβολικών διευθύνσεων σέ πραγματικές.

Η μετάφραση αυτή πραγματοποιείται μέ τή βοήθεια ένός ειδικού μεταφραστήκού προγράμματος τό όποίο καλεΐται «Συμβολομεταφραστής» (Assembler). Τό πρόγραμμα αυτό είναι γραμμένο σέ γλώσσα μηχανής καί δίνεται από τόν κατασκευαστή του ύπολογιστή. Έκτός από τή μετάφραση, τό πρόγραμμα αυτό μάς βοηθά στην άνίχνευση απλών σφαλμάτων γραφής του προγράμματος.

Ένα πρόγραμμα γραμμένο σέ συμβολική γλώσσα εισάγεται στον ύπολογιστή άφού προηγουμένως έχει εισαχθεί ό συμβολομεταφραστής.

Ός αποτέλεσμα λαμβάνεται τό πρόγραμμα σέ γλώσσα μηχανής τό όποίο εισάγεται αυτόμάτως στη συνέχεια στον ύπολογιστή γιά έκτέλεση.

Γενικά θά μπορούσαμε νά πούμε ότι ό προγραμματισμός μέ συμβολική γλώσσα είναι πιό εύκολος από τόν προγραμματισμό μέ γλώσσα μηχανής. Φυσικά, έπειδή ό συμβολομεταφραστής μεταφράζει σέ γλώσσα μηχανής, διαφέρει από τόν ένα τυπο ύπολογιστή στον άλλο.

8.4.2 Πίνακας εντολών σέ συμβολική γλώσσα ενός υποθετικού υπολογιστή.

Οι συνήθειες περιορισμοί στή διατύπωση τών συμβολικών ονομάτων τών εντολών είναι οι εξής:

- Τά συμβολικά όνόματα πρέπει νά αποτελούνται από 6 ή λιγότερους χαρακτήρες, από τούς όποιους ό πρώτος πρέπει νά είναι άλφαβητικός.
- Τό πεδίο θέσεως τής εντολής στή μνήμη, μπορεί νά αναφέρεται μέ συμβολικό όνομα. Τό ίδιο συμβολικό όνομα δέν πρέπει νά χρησιμοποιείται δύο ή περισσότερες φορές στό πεδίο θέσεως. Τό σύμβολο * στήν πρώτη στήλη τής γραμμής καθορίζει ότι οι πληροφορίες πού υπάρχουν στή γραμμή δέν είναι εντολές αλλά έπεξηγήσεις καί πρέπει νά άγνοηθοούν από τό συμβολομεταφραστή.
- Τά συμβολικά όνόματα πού αναφέρονται στό πεδίο διευθύνσεως, θά πρέπει άπαραιτήτως νά έχουν όρισθεϊ ως συμβολικές διευθύνσεις στό πεδίο θέσεως κάποιας εντολής.

Στή συνέχεια δίνονται μερικές από τίς εντολές σέ συμβολική γλώσσα ενός υποθετικού υπολογιστή.

Κωδικός εντολής	Πεδίο Διευθύνσεως	Περιγραφή εντολής
HLT		STOP. Σταματά τήν έκτέλεση τοῦ προγράμματος.
LDA	N	(A): = (N). Φόρτωσε τό περιεχόμενο τής μνήμης μέ συμβολική διεύθυνση N, στό συσσωρευτή.
STA	N	(N): = (A). 'Αποθήκευσε τό περιεχόμενο τοῦ συσσωρευτή στή μνήμη πού έχει συμβολική διεύθυνση N.
ADA	N	(A): = (A) + (N). Πρόσθεσε στό συσσωρευτή τό περιεχόμενο τής μνήμης μέ συμβολική διεύθυνση N.
SBA	N	(A): = (A) - (N). 'Αφαίρεσε από τό συσσωρευτή τό περιεχόμενο τής μνήμης μέ συμβολική διεύθυνση N.
MLA	N	(A): = (A) . (N). Πολλαπλασίασε τό περιεχόμενο τοῦ συσσωρευτή μέ αυτό τής μνήμης N.
DVA	N	(A): = (A)/(N). Διαίρεσε τό περιεχόμενο τοῦ συσσωρευτή μέ αυτό τής μνήμης N.

JMP	N	(PC): = N. Ό υπολογιστής εκτελεί άλμα στην έντολή που περιέχεται στη μνήμη N.
JAN	N	(PC): = N αν (A) < 0. Εκτελείται άλμα στην έντολή που περιέχεται στη μνήμη N, μόνο εφόσον τό περιεχόμενο του συσσωρευτή είναι άρνητικό.
OUT		Έντολή εξόδου (έγγραφής).
INP	N	Έντολή εισόδου (άναγνώσεως).
SAL	N	Όλίσθηση του συσσωρευτή κατά N θέσεις προς τά άριστερά.
SAR	N	Όλίσθηση του συσσωρευτή κατά N θέσεις προς τά δεξιά.
LDI	N	(B): = (N). Φόρτωσε τόν καταχωρητή δείκτη μέ τό περιεχόμενο τής μνήμης N.
STI	N	(N): = (B). Άποθήκευσε στη μνήμη N τό περιεχόμενο του καταχωρητή δείκτη.

Παράδειγμα 1.

Ποιό είναι τό αποτέλεσμα τής εκτελέσεως του έπόμενου προγράμματος;

Διεύθυνση μνήμης	Κωδικός έντολής	Πεδίο διευθύνσεως
X ADR TAG RES ZERO	LDA	TAG
	JAN	ADR
	STA	RES
	HLT	
	LDA	ZERO
	SBA	TAG
	JMP	X
	0	0

Έλέγχουμε ένα προς ένα τα βήματα του προγράμματος με βάση τον παραπάνω πίνακα. Έτσι έχουμε:

	LDA	TAG	: Τό περιεχόμενο της μνήμης TAG μεταφέρεται στο συσσωρευτή.
	JAN	ADR	: Άν $(A) < 0$, άλλα στην εντολή της θέσεως ADR.
X	STA	RES	: Φόρτωσε τό περιεχόμενο του συσσωρευτή στη μνήμη RES (έφ' όσον $(A) \geq 0$).
	HLT		: STOP
ADR	LDA	ZERO	: Τό περιεχόμενο της μνήμης ZERO μεταφέρεται στο συσσωρευτή (μηδενισμός του συσσωρευτή).
	SBA	TAG	: Αφαίρεσε από τό συσσωρευτή (δηλαδή από τό 0) τό περιεχόμενο της μνήμης TAG. Παίρνομε έτσι τό αντίθετο του άρνητικού άριθμού που περιέχει ή μνήμη TAG.
	JMP	X	: Άλλα πίσω στη θέση X. Θα έκτελεσθεί ή εντολή STA RES καί μετά ή HLT.

Όπως γίνεται εύκολα άντιληπτό, τό πρόγραμμα αυτό άποθηκεύει, στη διεύθυνση RES της μνήμης, την άπόλυτη τιμή του περιεχομένου της διεύθυνσεως TAG.

Παράδειγμα 2.

Νά γραφεί σέ συμβολική γλώσσα τό πρόγραμμα του παραδείγματος 2 της προηγούμενης παραγράφου 8.3.

Έστω ότι οι τιμές a, b, c, a_1, b_1, c_1 βρίσκονται άποθηκευμένες στις έξης θέσεις μνήμης:

$a \rightarrow \text{ALPHA}$
 $b \rightarrow \text{BETA}$
 $c \rightarrow \text{GAMA}$
 $a_1 \rightarrow \text{ALPHA1}$
 $b_1 \rightarrow \text{BETA1}$
 $c_1 \rightarrow \text{GAMA1}$

Άν τό πρόγραμμα άναλυθεί, θα έχει ως έξης:

Διεύθ. μνήμης	Κωδ. Έντολης	Πεδίο Διευθ.	Έπεξήγηση έντολης
....	LDA	ALPHA	Μεταφορά του a στο συσσωρευτή.
....	ADA	BETA	Πρόσθεσε τόν αριθμό b στο περιεχόμενο του συσσωρευτή.
....	ADA	GAMA	Πρόσθεσε τόν αριθμό c στο περιεχόμενο του συσσωρευτή.
....	STA	SUM	Μεταφορά του άθροισματος $a+b+c$ στή θέση μνήμης SUM.
....	LDA	ALPHA1	Μεταφορά του a_1 στο συσσωρευτή.
....	ADA	BETA1	Πρόσθεσε τόν αριθμό b_1 στο περιεχόμενο του συσσωρευτή.
....	ADA	GAMA1	Πρόσθεσε τόν αριθμό c_1 στο περιεχόμενο του συσσωρευτή.
....	STA	SUM1.	Μεταφορά του άθροισματος $a_1 + b_1 + c_1$ στή θέση SUM1.
....	LDA	SUM	Μεταφορά του άθροισματος $a + b + c$ στο συσσωρευτή.
....	SBA	SUM1	Αφαίρεσε τό άθροισμα $a_1 + b_1 + c_1$ από τό περιεχόμενο του συσσωρευτή.
....	OUT		Έκτύπωση του αποτελέσματος.
....	HLT		Πέρας διαδικασίας.

8.5 Προγραμματισμός σε γλώσσες ανωτέρου επιπέδου.

Σήμερα χρησιμοποιούνται γλώσσες προγραμματισμού ανωτέρου επιπέδου με τις οποίες μπορούμε να γράψουμε προγράμματα, χωρίς να γράφουμε μία - μία τις εντολές του υπολογιστή και χωρίς ακόμα να γνωρίζει ο προγραμματιστής τις εντολές του υπολογιστή και τις ακριβείς θέσεις μνήμης που καταχωρείται το πρόγραμμα, τα δεδομένα και τα ενδιάμεσα αποτελέσματα.

Κάθε εντολή σε μία από τις γλώσσες ανωτέρου επιπέδου αντιστοιχεί σε πολλές εντολές μηχανής. Οι ανώτερες γλώσσες περιέχουν εντολές και σύμβολα και προσαρμόζονται στις εφαρμογές και όχι στον υπολογιστή. Έτσι η γραφή ενός προγράμματος δεν εξαρτάται από το συγκεκριμένο υπολογιστή.

Τέτοιες γλώσσες ανωτέρου επιπέδου είναι οι παρακάτω:

- **FORTRAN (FORmula TRANslation).**

Χρησιμοποιείται κυρίως για την επίλυση μαθηματικών προβλημάτων.

- **COBOL (COmmon Business Oriented Language).**

Χρησιμοποιείται για προγραμματισμό διοικητικών και οικονομικών εφαρμογών.

- **ALGOL (ALGOrithmic Language).**

Συνδυάζει τις δύο παραπάνω χρήσεις.

- **BASIC (Beginners All - purpose Symbolic Instruction Code).**

Χρησιμοποιείται για απλές εφαρμογές και κυρίως για αρχάριους προγραμματιστές.

Για να είναι δυνατή η εκτέλεση ενός προγράμματος γραμμένου σε μία από τις ανώτερες γλώσσες, πρέπει το πρόγραμμα να μεταφρασθεί σε γλώσσα μηχανής. Η μετάφραση αυτή γίνεται από ένα ειδικό μεταφραστικό πρόγραμμα το οποίο καλείται μεταφραστής (compiler). Το πρόγραμμα αυτό είναι πολύπλοκο, και γραμμένο σε γλώσσα μηχανής. Από κάθε κατασκευαστή δίνονται για κάθε υπολογιστή ένας ή περισσότεροι μεταφραστές οι οποίοι καθορίζουν την ποικιλία γλωσσών που μπορούν να χρησιμοποιηθούν με το συγκεκριμένο υπολογιστή.

Κάθε μεταφραστής εκτελεί τις παρακάτω εργασίες:

- Αναγνωρίζει τα σύμβολα του προγράμματος.
- Καθορίζει την συντακτική δομή του προγράμματος.
- Βρίσκει τις τυχόν παραβιάσεις των κανόνων του λεξιλογίου, γραμματικής και συντακτικού.
- Μεταφράζει το πρόγραμμα σε γλώσσα μηχανής.
- Διανέμει τις θέσεις της μνήμης.

Ο προγραμματισμός σε μία από τις γλώσσες ανωτέρου επιπέδου είναι πλέον εύκολος. Παρακάτω θα εξετάσουμε τον προγραμματισμό σε γλώσσα Basic, γιατί η γλώσσα αυτή είναι η πιο απλή και η πιο εύχρηστη.

8.5.1 Προγραμματισμός σε γλώσσα BASIC.

Όπως είδαμε, η ονομασία της γλώσσας BASIC προέρχεται από τα αρχικά των λέξεων «Beginners All Purpose Symbolic Instruction Code» το οποίο σημαίνει συμβολικός κώδικας εντολών γενικού σκοπού για αρχάριους.

Υπάρχουν διάφορες παραλλαγές της γλώσσας Basic, ανάλογα για ποιά σύστημα υπολογιστών έχει συνταχθεί.

Στήν παράγραφο αυτή θα προσπαθήσουμε να δώσουμε όσο πιο γενικά μπορούμε τις βασικές αρχές που πρέπει να γνωρίζουμε για να γράψουμε ένα πρόγραμμα στην γλώσσα Basic.

α) Παράσταση αριθμών στη γλώσσα Basic.

Οι διάφοροι αριθμοί του δεκαδικού συστήματος παριστάνονται κανονικά. Παραδείγματος χάριν οι αριθμοί 48.752, 1762, — 14.4, — 0.005 παριστάνονται, όπως ακριβώς τους έχουμε γράψει. Έτσι αν θέλουμε μία σταθερά να λάβει μία από τις παραπάνω τιμές, θα γράψουμε: $A = 48.752$ ή $B = - 0.005$.

Όταν όμως έχουμε να παραστήσουμε αριθμούς γραμμένους με έκθετική μορφή, ή παράστασή τους στη γλώσσα Basic αλλάζει:

Παραδείγματος χάρις ο αριθμός

$$A = 55.2 \times 10^7$$

στην Basic γράφεται ως:

$$A = 55.2E+7$$

όπου το τμήμα του εκθέτη περιλαμβάνει το γράμμα E που συμβολίζει τη βάση 10 και ένα ακέραιο αριθμό που είναι ή δύναμη του 10.

β) Παράσταση συμβόλων αριθμητικών πράξεων.

Τά διάφορα σύμβολα αριθμητικών πράξεων που χρησιμοποιούμε στη γλώσσα Basic φαίνονται στον Πίνακα 8.5.1.

ΠΙΝΑΚΑΣ 8.5.1.

Τά σύμβολα των αριθμητικών πράξεων της BASIC.

Αριθμητική πράξη	Σύμβολο της Αριθμητικής	Σύμβολο της Basic
Πρόσθεση	+	+
Αφαίρεση	—	—
Πολλαπλασιασμός	. ή ×	*
Διαίρεση	: ή / ή —	/
Ύψωση σε δύναμη	δέν υπάρχει	**

γ) Παράσταση αριθμητικών και άλγεβρικών πράξεων.

Θά πρέπει να έχουμε υπόψη μας ότι στη γλώσσα Basic οι παράμετροι και οι ανεξάρτητες μεταβλητές μιας εξισώσεως ή αριθμητικής πράξεως, παριστάνονται πάντα με κεφαλαία γράμματα.

Στόν πίνακα 8.5.2 φαίνονται μερικά παραδείγματα παραστάσεως αριθμητικών και άλγεβρικών πράξεων.

ΠΙΝΑΚΑΣ 8.5.2.**Παράσταση αριθμητικών και άλγεβρικών πράξεων.**

Αριθμητική ή Άλγεβρική παράσταση	Παράσταση στη γλώσσα Basic
$a + b + \gamma$	$A + B + C$
$5x + 15$	$5 * X + 15$
$ax^2 + bx + \gamma$	$A * (X ** 2) + B * X + C$
$5a + \frac{6}{b} + \gamma^2$	$5 * A + 6 / B + C ** 2$
$\frac{a}{b + 18}$	$A / (B + 18)$
$(\frac{a + b}{3\gamma + \delta})^2$	$((A + B) / (3 * C + D)) ** 2$
$3(2a - \frac{b}{\gamma + \delta})$	$3 * (2 * A - B / (C + D))$

δ) Παράσταση μαθηματικών συναρτήσεων.

Υπάρχουν πολλές στοιχειώδεις μαθηματικές συναρτήσεις στη γλώσσα Basic όπως φαίνεται στον πίνακα 8.5.3.

Στόν πίνακα 8.5.4 φαίνονται παραδείγματα κωδικοποιήσεως άλγεβρικών εκφράσεων που περιέχουν συναρτήσεις, σε γλώσσα Basic.

ΠΙΝΑΚΑΣ 8.5.3.**Οι στοιχειώδεις συναρτήσεις της γλώσσας Basic.**

Συνάρτηση	Λειτουργία
SIN(X)	Υπολογίζει τό ημίτονο του X. Τό X εκφράζεται σε άκτίνια (rad).
COS(X)	Υπολογίζει τό συνημίτονο του X. Τό X εκφράζεται σε άκτίνια.
ATN(X)	Υπολογίζει τό τόξο έφαπτομένης του X.
EXP(X)	Υπολογίζει την τιμή του e^x . Τό e είναι ή βάση των φυσικών ή νεπερίων λογαρίθμων $= 2.73$
LOG(X)	Υπολογίζει τό φυσικό λογάριθμο (μέ βάση τό e) του X.
LGT(X)	Υπολογίζει τό δεκαδικό λογάριθμο (μέ βάση τό 10) του X.
ABS(X)	Υπολογίζει την απόλυτη τιμή του X.
SQR(X)	Υπολογίζει την τετραγωνική ρίζα του X.
INT(X)	Υπολογίζει τό άκέραιο τμήμα του X.

ΠΙΝΑΚΑΣ 8.5.4.**Παραδείγματα κωδικοποιήσεως εκφράσεων που περιέχουν συναρτήσεις σε γλώσσα Basic.**

Άλγεβρική Έκφραση	Έκφραση Basic
$3 \sin x + x + y\delta$	$3 * \text{SIN}(X) + \text{ABS}(X) + C * D$
$e^x \sin x + \frac{3}{4}$	$\text{EXP}(X) * \text{COS}(X) + 3 / 4$
$2\sqrt{x} + a^2 + b^2$	$2 * \text{SQR}(X) + A ** 2 + B ** 2$
$\frac{1}{x} + 5 \log x$	$1 / X + 5 * \text{LGT}(X)$

ΠΙΝΑΚΑΣ 8.5.5.**Τα σύμβολα των λογικών σχέσεων και οι εντολές άλματος με προϋποθέσεις στη Basic.**

Σύμβολο	Έπεξήγηση	Παράδειγμα	Ανάλυση
=	Ίσο προς	IF X=Y THEN 30	Αν τό Χ είναι Ίσο προς τό Υ, τότε ή εκτέλεση του προγράμματος θά συνεχισθεί από την έντολή 30, διαφορετικά από την άμέσως επόμενη.
< >	διάφορο του, όχι Ίσο μέ	IF X < > Y THEN 40	Αν τό Χ είναι διάφορο του Υ (δνισο), τότε θά εκτελεσθεί ή έντολή 40, διαφορετικά ή επόμενη.
<	μικρότερο από	IF X<Y THEN 50	Η εκτέλεση θά συνεχισθεί από την έντολή 50, μόνο αν Χ<Υ.
<= ή =<	μικρότερο ή Ίσο	IF X<= Y THEN 45	Η έντολή 45 θά εκτελεσθεί άμέσως μετά την έντολή IF, μόνο όταν Χ≤Υ.
>	μεγαλύτερο από	IF X>Y THEN 20	Η έντολή 20 θά εκτελεσθεί άμέσως μετά την έντολή IF, μόνο όταν Χ>Υ.
>= ή ..>	μεγαλύτερο ή Ίσο	IF X>= Y THEN 60	Η έντολή 60 θά εκτελεσθεί άμέσως μετά την έντολή IF μόνο όταν Χ≥Υ.

8.5.2 Διάφορα χαρακτηριστικά της γλώσσας Basic.

Στήν παράγραφο αυτή θά προσπαθήσομε νά δείξομε διάφορα χαρακτηριστικά της γλώσσας Basic μέ τή βοήθεια ενός προγράμματος Basic τό όποιο συγκρίνει δύο αριθμούς, γιά νά καθορίσει αν ό ένας αριθμός είναι μεγαλύτερος, ίσος ή μικρότερος από τόν άλλο.

```

LIST
COMPAR
100 REM ΤΟ ΠΡΟΓΡΑΜΜΑ ΑΥΤΟ ΣΥΓΚΡΙΝΕΙ
110 REM ΔΥΟ ΑΡΙΘΜΟΥΣ ΜΕΤΑΞΥ ΤΟΥΣ
120 REM
130 READ A, B
140 IF A = . 01 THEN 220
145 IF A = B THEN 160
148 IF A<B THEN 180
150 PRINT A; «ΕΙΝΑΙ ΜΕΓΑΛΥΤΕΡΟ ΑΠΟ»; B
155 GOTO 130
160 PRINT A; «ΕΙΝΑΙ ΙΣΟ ΠΡΟΣ ΤΟ»; B
170 GOTO 130
180 PRINT A; «ΕΙΝΑΙ ΜΙΚΡΟΤΕΡΟ ΑΠΟ»; B
190 GOTO 130
200 DATA 3, 4, 1.7, 1.1, 31, 31, -3, 2, 0, 0
210 DATA .01, 0
220 END

RUN

```

```

3 ΕΙΝΑΙ ΜΙΚΡΟΤΕΡΟ ΑΠΟ 4
1.7 ΕΙΝΑΙ ΜΕΓΑΛΥΤΕΡΟ ΑΠΟ 1.1
31 ΕΙΝΑΙ ΙΣΟ ΠΡΟΣ ΤΟ 31
-3 ΕΙΝΑΙ ΜΙΚΡΟΤΕΡΟ ΑΠΟ 2
0 ΕΙΝΑΙ ΙΣΟ ΠΡΟΣ ΤΟ 0

```

α) 'Η έντολή LIST.

Αυτή είναι η πρώτη έντολή και είναι μία ειδική έντολή της γλώσσας Basic την οποία δίνουμε στον υπολογιστή για να μάς εκτυπώσει το πρόγραμμα «COMPAR» τό όποιο βρίσκεται στην μνήμη του. (Υποτίθεται ότι τό πρόγραμμα ήταν αποθηκευμένο σέ κάποιο μέσο δευτερεύουσας μνήμης και τό καλέσαμε πρίν δώσουμε την έντολή LIST ή μόλις τώρα τό έχομε βάλει στή μνήμη και μέ την έντολή LIST θέλομε να τό δοϋμε δλο).

Μόλις λοιπόν δώσουμε την έντολή LIST, ο υπολογιστής αυτόματως γράφει τό δνομα του προγράμματος, πού σ' αυτή την περίπτωση τό ονομάζομε COMPAR, και στή συνέχεια τό τυπώνει. Έάν τώρα προσέξομε στό πρόγραμμα, θά δοϋμε ότι μπροστά από κάθε έντολή υπάρχει ένας αριθμός. 'Ο αριθμός αυτός λέγεται «διεύθυνση της έντολης» και είναι απαραίτητος στή γλώσσα Basic.

β) Έντολή REM.

Στίς διευθύνσεις 100, 110 και 120 παρατηρούμε ότι υπάρχει μία έντολή REM ακολουθούμενη από άλλους χαρακτήρες ή όχι. 'Η έντολή REM στή γλώσσα Basic χρησιμοποιείται για την άναγραφή διαφόρων παρατηρήσεων πρós υποβοήθηση του χρήστη. 'Ο υπολογιστής δέν τίς λαμβάνει υπόψη του, αλλά τίς τυπώνει μόνο όταν δώσουμε την έντολή LIST.

γ) Έντολή READ - DATA.

Στίς γραμμές 130, 200, 210 βλέπομε ότι υπάρχουν δύο έντολές. 'Η έντολή READ καί ή έντολή DATA. 'Η έντολή READ A,B σημαίνει ότι δίνομε έντολή στον υπολογιστή νά διαβάσει τίς σταθερές A καί B από τόν πίνακα δεδομένων τά όποια χαρακτηρίζει ή έντολή DATA.

Μόλις ό υπολογιστής αναγνωρίσει τήν έντολή READ A,B μεταβαίνει στην έντολή DATA καί διαβάσει τόσους αριθμούς, όσες καί οι σταθερές τίς όποιες του έχομε δώσει έντολή νά διαβάσει (στήν προκειμένη περίπτωση δύο A καί B). Έτσι από όλους τούς αριθμούς πού βρίσκονται στην έντολή DATA διαβάσει τούς δύο πρώτους (δηλαδή 3 καί 4) καί καθορίζει τίς τιμές των σταθερών ως:

$$A = 3$$

$$B = 4$$

δ) Έντολή άλματος μέ προϋπόθεση (IF).

Στίς γραμμές 140, 145 καί 148 υπάρχουν τρείς έντολές άλματος μέ προϋπόθεση (IF). 'Η έντολή της γραμμής 140 χρησιμοποιείται γιά νά σταματήσει ή εκτέλεση του προγράμματος καί νά αρχίσει έτσι ό υπολογιστής τήν εκτέλεση της έντολής στη γραμμή 220 (στήν προκειμένη περίπτωση είναι END) έφόσον βέβαια ή τιμή της σταθεράς A είναι 0.01. Οι γραμμές 145 καί 148 καθοδηγούν τόν υπολογιστή στις αντίστοιχες έντολές εκτυπώσεως (PRINT) ανάλογα μέ τή σχέση μεταξύ των τιμών A καί B.

Οι έντολές IF είναι έντολές άλματος μέ προϋπόθεση πού σημαίνει ότι κάτω από όρισμένες προϋποθέσεις πού καθορίζονται μέσα στην έντολή, γίνονται άλματα κατά τήν εκτέλεση των έντολών του προγράμματος. Στόν πίνακα 8.5.5 φαίνονται όλες οι έντολές άλματος μέ προϋπόθεση στη γλώσσα Basic.

ε) Έντολή εκτυπώσεως PRINT.

Στίς γραμμές 150, 160 καί 180 ή έντολή PRINT έχει ως αποτέλεσμα τήν εκτύπωση του στοιχείου πού ακολουθεί τή λέξη PRINT στη διεύθυνση αυτή, στην τερματική περιφερειακή μονάδα (όθόνη CRT ή τηλέτυπο). Παρατηρώντας τίς έντολές αυτές, βλέπομε ότι στη γλώσσα Basic επιτρέπεται ή ανάμιξη άλφαβητικών καί αριθμητικών χαρακτήρων.

Παραδείγματος χάρη ή έντολή 150 PRINT A; «ΕΙΝΑΙ ΜΕΓΑΛΥΤΕΡΟ ΑΠΟ»; B υπαγορεύει στον υπολογιστή νά εκτυπώσει πρώτα τόν αριθμό A. Μετά από ένα πολύ μικρό διάστημα θά εκτυπωθεί στη συνέχεια ό,τι περιέχεται μεταξύ των εισαγωγικών « », πού μπορεί νά είναι κεφαλαία, ή μικρά γράμματα, αριθμοί κλπ. Στη συνέχεια, ύστερα πάλι από ένα μικρό διάστημα θά εκτυπωθεί ή τιμή της σταθεράς B.

Έάν αντί γιά τό σύμβολο (:) χρησιμοποιήσομε κόμμα (,), αυτό θά είχε ως αποτέλεσμα μεγαλύτερα διαστήματα κατά τήν εκτύπωση.

στ) Έντολή άλματος GOTO.

Στίς γραμμές 155, 170, 190 υπάρχει μία έντολή GOTO. 'Η έντολή GOTO στη γλώσσα Basic χρησιμοποιείται γιά νά δείξει στον υπολογιστή τή διεύθυνση έντολης, ή όποια θά πρέπει νά εκτελεσθεί άμέσως μετά τήν πρώτη έντολή. Έτσι, όταν ό υπολογιστής φθάσει στη γραμμή 155, ή επόμενη έντολή πού θά εκτελεσθεί εί-

ναι ή έντολή 130 καί όχι ή έντολή 160. Τό ίδιο φυσικά Ισχύει καί γιά τίς έντολές των γραμμών 170 καί 190. Δηλαδή ή έντολή GOTO είναι μία έντολή άλματος χωρίς προϋπόθεση.

ζ) *Η έντολή END.*

Η έντολή αυτή αναγκάζει τόν υπολογιστή νά σταματήσει τήν εκτέλεση του προγράμματος. Δηλαδή ή έντολή END είναι έντολή πέραςτος.

η) *Η έντολή RUN.*

Μέ τήν έντολή αυτή ο υπολογιστής αρχίζει τήν εκτέλεση του προγράμματος τό όποιο έχει φορτωθεϊ ήδη στή μνήμη του. Στήν προκειμένη περίπτωση δίνουμε έντολή στόν υπολογιστή νά εκτελέσει τό πρόγραμμα COMPAR.

θ) *Διάφορες άλλες έντολές τής γλώσσας Basic.*

Εκτός από τίς έντολές πού αναφέραμε, υπάρχουν καί πολλές άλλες έντολές τής γλώσσας Basic. Μερικές από αυτές είναι οι έντολές INPUT, DIM, FOR-NEXT καί άλλες. Οι έντολές αυτές μπορούν νά επεξηγηθούν καλύτερα μέ τήν βοήθεια του παρακάτω προγράμματος.

```
10 REM ΠΡΟΓΡΑΜΜΑ ΥΠΟΛΟΓΙΣΜΟΥ
20 REM ΤΟΥ ΑΘΡΟΙΣΜΑΤΟΣ Ν ΑΡΙΘΜΩΝ
30 REM ΤΗΣ ΜΟΡΦΗΣ  $X = A1 + A2 + A3 + \dots + AN$ 
40 REM
100 PRINT «ΠΟΣΟΥΣ ΑΡΙΘΜΟΥΣ ΘΕΛΕΤΕ»
200 PRINT «ΝΑ ΠΡΟΣΘΕΣΕΤΕ;»
300 INPUT N
400 DIM X(N)
500 S = 0
600 FOR I = 1 TO N STEP 1
700 INPUT A(I)
800 S = S + A(I)
900 NEXT I
1000 PRINT «ΤΟ ΑΘΡΟΙΣΜΑ ΕΙΝΑΙ:»; S
1100 END
```

SAVE «MYPROG. BAS»

LIST

Μετά τήν έντολή αυτή τό πρόγραμμα όλο εκτυπώνεται στήν όθόνη μας

RUN

ΠΟΣΟΥΣ ΑΡΙΘΜΟΥΣ ΘΕΛΕΤΕ ΝΑ ΠΡΟΣΘΕΣΕΤΕ;

→ ? 5

→ ? 2

→ ? 7

→ ? 3

→ ? -2

→ ? 45

(Τό → σημαίνει ότι εδώ ο υπολογιστής περιμένει απάντηση από τον χειριστή.)

ΤΟ ΑΘΡΟΙΣΜΑ ΕΙΝΑΙ: 55

** END **

Θά προσπαθήσουμε τώρα να παρακολουθήσουμε την εκτέλεση του παραπάνω προγράμματος, εξηγώντας πρώτα τις διάφορες νέες εντολές.

1. Έντολή INPUT.

Η έντολή αυτή αναγκάζει τον υπολογιστή να σταματήσει την εκτέλεση του προγράμματος και να περιμένει την είσοδο των δεδομένων από το χειριστή.

2. Έντολή DIM.

Η έντολή αυτή χρησιμοποιείται στην περίπτωση που σέ ένα πρόγραμμα χρησιμοποιούμε πολλές τιμές μιās μεταβλητής π.χ. A(1), A(2).... κλπ. Η έντολή DIM X(N) δηλώνει ότι η μεταβλητή X προβλέπεται να λάβει N διαφορετικές τιμές.

3. Έντολή FOR-NEXT.

Η έντολή αυτή χρησιμοποιείται γιά να δώσουμε μία τιμή σέ μία μεταβλητή πολλές φορές ή κοινώς γιά να επαναλάβουμε μία σειρά εντολών πολλές φορές.

Η μεταβλητή που εμφανίζεται μετά τή λέξη FOR είναι δείκτης που παίρνει ως αρχική τιμή την τιμή της εκφράσεως, μετά τό ίσον και ως τελική τιμή, την τιμή της εκφράσεως μετά τό ενδεικτικό TO. Η έκφραση που ακολουθεί τό ενδεικτικό STEP καθορίζει την αύξηση ή τό βήμα μέ τό όποιο θά αύξάνει κάθε φορά ή τιμή της μεταβλητής. Π.χ.

FOR I = 1 TO 50 STEP .01

NEXT I

σημαίνει ότι ή μεταβλητή I παίρνει αρχικά την τιμή 1 και τό πρόγραμμα συνεχίζει την εκτέλεση των διαδοχικών εντολών που ακολουθούν μέχρις ότου φθάσει στην έντολή

NEXT I

όποτε ξαναγυρίζει πάλι πίσω στην έντολή

FOR I = 1 TO 50 STEP .01

Αλλά τώρα ή τιμή της μεταβλητής I αύξάνεται κατά την τιμή του βήματος, δηλαδή γίνεται 1.01. Η διαδικασία αυτή συνεχίζεται μέχρις ότου ή μεταβλητή I πάρει όλες τις τιμές από τό 1 ως τό 50 σέ βήματα αυξανόμενα κατά 0.01. Μετά μπορούμε να συνεχίσουμε μέ την εκτέλεση των εντολών που υπάρχουν μετά την έντολή NEXT I. Εάν δέν δώσουμε την έντολή STEP, ο υπολογιστής υποθέτει ότι τό βήμα είναι 1.

4. 'Η έντολή SAVE «...».

Μέ την έντολή αυτή ο υπολογιστής αποθηκεύει τό πρόγραμμα αυτό στο όποιο έχομε δώσει τό όνομα «MYPROG. BAS». 'Η προέκταση BAS βοηθά τό χειριστή νά καταλάβει σέ ποία γλώσσα είναι γραμμένο τό πρόγραμμα MYPROG.

Ής υποθέσομε τώρα ότι τό πρόγραμμα έχει αποθηκευθεϊ στή βοηθητική μνήμη του υπολογιστή (π.χ. σέ μία εύκαμπτη δισκέτα) καί ότι θέλομε νά τό εκτελέσομε.

Καταρχήν δίνεται ή έντολή:

LOAD «MYPROG. BAS»

μέ την όποία ο υπολογιστής φορτώνει τό πρόγραμμα MYPROG. BAS στήν κεντρική μνήμη. Στή συνέχεια δίνεται ή έντολή LIST γιά νά δοϋμε (αν θέλομε) τίς έντολές του προγράμματος. 'Αφού βεβαιωθοϋμε ότι αυτό είναι τό πρόγραμμα, τό όποιο θέλομε νά εκτελέσομε, δίνομε την έντολή RUN όποτε αρχίζει καί ή εκτέλεση του προγράμματος.

Στήν όθόνη μας (αν χρησιμοποιοϋμε όθόνη) εμφανίζεται ή έρώτηση «πόσους αριθμούς θέλετε νά προσθέσετε;», καί ο υπολογιστής σταματά την εκτέλεση του προγράμματος περιμένοντας τό χειριστή νά άπαντήσει. 'Όταν ο χειριστής άπαντήσει π.χ. μέ τον αριθμό 5, ο υπολογιστής εκτελεί τίς έντολές που ακολουθούν οι όποιες στήν περίπτωση αυτή είναι οι έντολές:

FOR I = 1 TO 5

.
.
.

NEXT I

Έτσι ρωτάει τό χειριστή 5 φορές (για τό παράδειγμά μας) για τίς τιμές των αριθμών που πρόκειται νά προστεθοϋν μεταξύ τους. 'Όταν τελειώσει καί αυτό, ο υπολογιστής δίνει την άπαιτούμενη άπάντηση καί τελειώνει την εκτέλεση του προγράμματος.

Παρατηροϋμε τά έξής:

Τό άθροισμα στο πρόγραμμα τό συμβολίζομε μέ τό γράμμα S. 'Αρχικά στο όλο άθροισμα δίνομε την τιμή $S = 0$. Μόλις ο χειριστής δώσει την πρώτη τιμή αριθμού που θά προστεθεϊ, αυτομάτως τό αρχικό άθροισμα γίνεται:

$$S = S + A(1) = 0 + A(1)$$

Όταν ο χειριστής έρωτηθεϊ για δεύτερη φορά καί δώσει την τιμή $A(2)$, τό άθροισμα S αύξάνεται σύμφωνα μέ την τιμή αυτή καί γίνεται:

$$S = S' + A(2) \text{ όπου } S' = 0 + A(1)$$

Έπομένως:

$$S = 0 + A(1) + A(2)$$

κ.ο.κ. μέχρις ότου δωθοϋν όλοι οι αριθμοί που πρόκειται νά προστεθοϋν.

8.6 'Υποπρογράμματα (Subprograms).

Ός υποπρόγραμμα καθορίζομε ένα σύνολο έντολών που εκτελοϋν μία συγκεκριμένη έπεξεργασία ή όποία κατά τή διάρκεια ενός προγράμματος μπορεί νά χρησιμοποιηθεϊ πολλές φορές.

RUN

ΠΟΣΟΥΣ ΑΡΙΘΜΟΥΣ ΘΕΛΕΤΕ ΝΑ ΠΡΟΣΘΕΣΕΤΕ;

→ ? 5

→ ? 2

→ ? 7

→ ? 3

→ ? -2

→ ? 45

(Τό → σημαίνει ότι έδω ό ύπολογιστής περιμένει απάντηση από τόν χειριστή.)

ΤΟ ΑΘΡΟΙΣΜΑ ΕΙΝΑΙ: 55

** END **

Θά προσπαθήσουμε τώρα νά παρακολουθήσουμε τήν εκτέλεση του παραπάνω προγράμματος, εξηγώντας πρώτα τίς διάφορες νέες έντολές.

1. Έντολή INPUT.

Η έντολή αυτή αναγκάζει τόν ύπολογιστή νά σταματήσει τήν εκτέλεση του προγράμματος καί νά περιμένει τήν είσοδο των δεδομένων από τό χειριστή.

2. Έντολή DIM.

Η έντολή αυτή χρησιμοποιείται στην περίπτωση που σέ ένα πρόγραμμα χρησιμοποιούμε πολλές τιμές μιός μεταβλητής π.χ. A(1), A(2).... κλπ. Η έντολή DIM X(N) δηλώνει ότι ή μεταβλητή X προβλέπεται νά λάβει N διαφορετικές τιμές.

3. Έντολή FOR-NEXT.

Η έντολή αυτή χρησιμοποιείται γιά νά δώσουμε μία τιμή σέ μία μεταβλητή πολλές φορές ή κοινώς γιά νά επαναλάβουμε μία σειρά έντολών πολλές φορές.

Η μεταβλητή που εμφανίζεται μετά τή λέξη FOR είναι δείκτης που παίρνει ως αρχική τιμή τήν τιμή τής εκφράσεως, μετά τό ίσον καί ως τελική τιμή, τήν τιμή τής εκφράσεως μετά τό ένδεικτικό TO. Η έκφραση που ακολουθεί τό ένδεικτικό STEP καθορίζει τήν αύξηση ή τό βήμα μέ τό όποιο θά αυξάνει κάθε φορά ή τιμή τής μεταβλητής. Π.χ.

FOR I = 1 TO 50 STEP .01

NEXT I

σημαίνει ότι ή μεταβλητή I παίρνει αρχικά τήν τιμή 1 καί τό πρόγραμμα συνεχίζει τήν εκτέλεση των διαδοχικών έντολών που ακολουθούν μέχρις ότου φθάσει στην έντολή

NEXT I

όποτε ξαναγυρίζει πάλι πίσω στην έντολή

FOR I = 1 TO 50 STEP .01

Άλλά τώρα ή τιμή τής μεταβλητής I αυξάνεται κατά τήν τιμή του βήματος, δηλαδή γίνεται 1.01. Η διαδικασία αυτή συνεχίζεται μέχρις ότου ή μεταβλητή I πάρει όλες τίς τιμές από τό 1 ως τό 50 σέ βήματα αυξανόμενα κατά 0.01. Μετά μπορούμε νά συνεχίσουμε μέ τήν εκτέλεση των έντολών που υπάρχουν μετά τήν έντολή NEXT I. Εάν δέν δώσουμε τήν έντολή STEP, ό ύπολογιστής υποθέτει ότι τό βήμα είναι 1.

```

10 INPUT N, J
20 «ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΝΙ»
30 LET K = N
40 GOSUB 1000
50 LET N1 = F
60 REM «ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ JΙ»
70 LET K = J
80 GOSUB 1000
90 LET J1 = F
100 REM «ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ (N - J)Ι»
110 LET K = N - J
120 GOSUB 1000
130 LET C = N1/(J1 * F)
140 PRINT «ΤΟ ΠΛΗΘΟΣ ΤΩΝ ΣΥΝΔΥΑΣΜΩΝ»; N; «ΠΡΑΓΜΑΤΩΝ»
150 PRINT «ΑΝΑ»; J; «ΕΙΝΑΙ:»; C
160 STOP
170 REM
1000 REM ΥΠΟΛΟΓΙΣΜΟΣ ΤΗΣ ΣΥΝΑΡΤΗΣΕΩΣ F = KΙ
2000 LET F = 1
2100 IF K = 0 THEN 2600
2200 FOR L = 1 TO K
2300 LET F = F * L
2400 NEXT L
2600 RETURN
5000 DATA 8,3
9999 END

```

Η μόνη καινούργια έντολή είναι η έντολή LET ή οποία απλώς χρησιμοποιείται για να καθορίσουμε την τιμή μιάς μεταβλητής. Είναι σχεδόν ίδια με την έκφραση «=».

8.7 Έρωτήσεις.

1. Τι καλείται Software και τί Hardware;
2. Τι καλείται Firmware;
3. Ποιά είναι τα στάδια αναπτύξεως του Software/Hardware.
4. Πώς χωρίζονται οι έντολές του υπολογιστή ανάλογα με την εργασία τους και τον τρόπο εκτέλεσής τους;
5. Τι σημαίνει η έντολή: FOR K= 0.1 STEP 0.02;
6. Τι καλείται υποπρόγραμμα; Ποιές οι διαφορές ή οι ομοιότητες του με τις υπορουτίνες;

8.8 Άσκήσεις.

1. Γράψτε ένα πρόγραμμα σε γλώσσα μηχανής, όπως αυτή που περιγράφεται στο κείμενο, το οποίο να εκτελεί την πράξη:

$$x = a + b - c$$

2. Να γραφεί το πρόγραμμα της προηγούμενης άσκησης σε γλώσσα μηχανής Assembly.
3. Να γραφεί ένα πρόγραμμα σε γλώσσα Basic το οποίο να υπολογίζει το έμβαδόν και την περίμετρο ενός κύκλου.
4. Να γραφεί ένα πρόγραμμα σε γλώσσα Basic το οποίο να υπολογίζει την τιμή της παραστάσεως:

$$x = at^2 + bt + \gamma$$

για σταθερές τιμές των παραμέτρων α , β και γ και για τιμές της μεταβλητής $t = 0, 0.1, 0.2, 0.3, \dots, 99.99, 100$ και να εκτυπώνει τα αποτελέσματα.

5. Να γραφεί ένα πρόγραμμα σε γλώσσα Basic το οποίο να υπολογίζει την τιμή της παραστάσεως:

$$x = \begin{cases} \alpha \sin(\omega t) + \beta^2 \eta \mu(\omega t) & \text{για τιμές του } t \text{ μεταξύ } 9 \leq t \leq 10 \\ \alpha^t + \beta^{1/3} \eta \mu(\omega t) & \text{για όλες τις άλλες τιμές του } t. \end{cases}$$

ΚΕΦΑΛΑΙΟ ΕΝΑΤΟ

ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΥΠΟΛΟΓΙΣΤΗ

9.1 Γενικά.

Γιά να εκτελέσει ένα πρόγραμμα ο υπολογιστής, πρέπει, εκτός από την εκτέλεση των εντολών του προγράμματος, να εκτελέσει και όρισμένες εργασίες απαραίτητες για την εισαγωγή του προγράμματος, την παρακολούθηση της εκτελέσεώς του, τη διαβίβαση μηνυμάτων στο χειριστή (τά όποια αφορούν τό εκτελούμενο πρόγραμμα), τόν καθορισμό της κατάστασης του υπολογιστή, την εισαγωγή του μεταφραστή πού χρειάζεται τό πρόγραμμα στή μνήμη του κλπ.

Οι διαδικασίες αυτές είναι φανερό ότι είναι ανεξάρτητες από τό πρόγραμμα τό όποιο εκτελείται. Έξασφαλίζονται μέ ένα ειδικό πρόγραμμα πού είναι γραμμένο σέ γλώσσα μηχανής καί τό όποιο λέγεται **«Λειτουργικό σύστημα του υπολογιστή» (Operating System)**.

Τό πρόγραμμα αυτό περιέχει διάφορα τμήματα, από τά όποια τό καθένα χωριστά εκτελεί μία συγκεκριμένη εργασία. Μερικά από τά τμήματα αυτά πρέπει νά βρίσκονται συνεχώς στή κεντρική μνήμη του υπολογιστή, ένω τά υπόλοιπα μπορεί νά βρίσκονται στίς μονάδες περιφερειακής μνήμης καί νά καλούνται όταν χρειάζονται.

Στό κεφάλαιο αυτό θά εξετάσομε τά βασικότερα από τά τμήματα του λειτουργικού συστήματος.

9.2 Τό Πρόγραμμα Έκκινήσεως (Bootstrap).

Γιά νά εισαχθεί ένα πρόγραμμα στόν υπολογιστή, θά πρέπει νά προηγηθεί επικοινωνία του υπολογιστή μέ μία περιφερειακή μονάδα εισαγωγής πληροφοριών. Κατά τή διάρκεια της επικοινωνίας αυτής ο υπολογιστής εκτελεί όρισμένες «έντολές εισόδου» οι όποιες προφανώς πρέπει νά υπάρχουν στή μνήμη του υπολογιστή. Τό σύνολο των εντολών αυτών αποτελεί τό πρόγραμμα εκκινήσεως. Αυτό εισάγεται στή μνήμη του υπολογιστή είτε μέσω διακοπών, είτε αυτόματως μέ τήν εκκίνηση του υπολογιστή, όταν βρίσκεται μόνιμα αποθηκευμένο σ' αυτήν (π.χ. στή ROM). Μέ τήν εκτέλεση του προγράμματος προκαλείται ή εισαγωγή στή μνήμη ενός άλλου προγράμματος, τό όποιο ονομάζεται **«πρόγραμμα φορτώσεως» (Loader)**.

9.3 Τό πρόγραμμα φορτώσεως.

Αυτό έχει ως αποστολή νά φορτώνει τό πρόγραμμα πού θά εκτελεσθεί στή

μνήμη του υπολογιστή. Βασική εργασία του είναι ο υπολογισμός των διευθύνσεων της μνήμης που θα τοποθετηθεί τό πρόγραμμα που φορτώνεται έτσι, ώστε να αποφύγουμε την τυχαία επικάλυψη ορισμένων τμημάτων της μνήμης με άλλα προγράμματα, όπως συμβαίνει, όταν διάφορα μικρά προγράμματα ενώνονται, γιά να σχηματίσουν ένα μεγαλύτερο.

Μιά άλλη βασική αποστολή του προγράμματος φορτώσεως είναι ή συνένωση (Link) ενός προγράμματος με τά υποπρογράμμά του.

9.4 Τά προγράμματα οδηγήσεως (Drivers).

Είναι προγράμματα που επιβλέπουν τή μεταφορά πληροφοριών από τή κεντρική μνήμη του υπολογιστή στις περιφερειακές μονάδες καί αντίστροφα, απαλλάσσοντας κάθε φορά τό πρόγραμμά μας από τήν έπιτηρήση της επικοινωνίας του υπολογιστή με τά περιφερειακά του. Κάθε πρόγραμμα οδηγήσεως αντιστοιχεί σέ μία ομάδα περιφερειακών μονάδων π.χ. πρόγραμμα οδηγήσεως μαγνητικών δίσκων ή πρόγραμμα οδηγήσεως μαγνητικών ταινιών κλπ.

Ο συντονισμός της μεταφοράς των πληροφοριών γίνεται ύστερα από εξέταση της κατάστασεως που βρίσκεται ή περιφερειακή μονάδα (π.χ. ελεύθερη ή κατειλημμένη), από εξέταση της εκτελέσεως των εντολών εισόδου/εξόδου, από μετατροπή της μορφής των πληροφοριών με κατάλληλους κώδικες κλπ.

9.5 Άλλα μέρη του Λειτουργικού Συστήματος.

Έκτός από τίς παραπάνω βασικές λειτουργίες τό λειτουργικό σύστημα περιλαμβάνει καί άλλα προγράμματα τά όποια διευκολύνουν τό χειριστή καί αύξάνουν τήν απόδοση του υπολογιστή. Π.χ. τό πρόγραμμα **«παρακολουθήσεως»** (Scheduler) με τό όποιο άσκεϊται έλεγχος της λειτουργίας των περιφερειακών μονάδων (έναρξη-πέρας) που άπαιτεί τό έκτελούμενο πρόγραμμα. Ο έλεγχος αυτός άσκεϊται μέσω ενός προγράμματος **«επικοινωνίας χειριστή»** (Operator's communication), τό όποιο μεταβιβάζει στό χειριστή πληροφορίες γιά τήν πορεία εκτελέσεως του προγράμματος καί διάφορες αιτήσεις προς τό χειριστή, όπως π.χ. αλλαγή ενός μαγνητικού δίσκου κλπ.

Τά προγράμματα αυτά επιτρέπουν καί τήν αντίστροφη επικοινωνία. Δηλαδή, ό χειριστής μπορεί νά κατευθύνει ό ίδιος τήν εκτέλεση του προγράμματος με ειδικά μηνύματα π.χ. διακοπή εργασίας κλπ.

Ένα άλλο πρόγραμμα είναι τό πρόγραμμα **«Λογιστικής Διαχειρήσεως»** (accounting) τό όποιο πληροφορεί τό χειριστή γιά τό χρόνο άπασχολήσεως της κεντρικής μονάδας του υπολογιστή από ένα συγκεκριμένο πρόγραμμα, γιά τό μέγεθος της μνήμης που χρησιμοποιήθηκε, τό χρόνο άπασχολήσεως των διωρύγων κλπ.

Τό λειτουργικό σύστημα διαφέρει από υπολογιστή σέ υπολογιστή ανάλογα με τό μέγεθος, τίς απαιτήσεις, τό κόστος καί τόν κατασκευαστή. Οι διάφοροι κατασκευαστές κατασκευάζουν διάφορα λειτουργικά συστήματα ανάλογα με τίς απαιτήσεις της αγοράς.

9.6 Ο τρόπος λειτουργίας του υπολογιστή.

Τό πρόγραμμα του ήλεκτρονικού υπολογιστή αποτελείται από μία ακολουθία

έντολών. Για να αρχίσει ή εκτέλεση μιᾶς έντολής, πρέπει απαραίτητα να ἔχει συμπληρωθεῖ ή εκτέλεση τῆς προηγούμενης. Κάθε έντολή για τήν εκτέλεσή της ἀπασχολεῖ πολύ μικρό τμήμα τῆς ὑπολογιστικῆς δυνάμεως τοῦ ὑπολογιστή. Οἱ ἀριθμητικές π.χ. έντολές δέν ἀπασχολοῦν καθόλου τίς διώρυγες, ἐνῶ οἱ έντολές εισόδου-ἐξόδου δέν ἀπασχολοῦν καθόλου τήν ἀριθμητική καί λογική μονάδα.

Ἔτσι οὐσιαστικά μπορούμε να ποῦμε ὅτι ὁ ὑπολογιστής ὑποἀπασχολεῖται.

Ἡ ἀπόδοση τοῦ ὑπολογιστή εἶναι δυνατόν να αὐξάνεται μόνο ὅταν ὁ ὑπολογιστής ἐπεξεργάζεται πολλά προγράμματα ταυτόχρονα.

Δύο βασικές ἔννοιες στή λειτουργία τοῦ ὑπολογιστή εἶναι οἱ ἔννοιες τοῦ OFF-LINE καί τοῦ ON-LINE.

Off Line σημαίνει ὅτι ὁ χειριστής δέν ἐπικοινωνεῖ ἀπευθείας μέ τόν ὑπολογιστή, ἀλλά ἐτοιμάζει τό πρόγραμμά του π.χ. σέ δελτία μέ τά ὁποῖα τροφοδοτεῖ τή μονάδα ἀναγνώσεως δελτίων.

On Line σημαίνει ὅτι τό πρόγραμμα ἢ τά δεδομένα εἰσάγονται ἀπευθείας στόν κυρίως ὑπολογιστή. Αὐτό γίνεται συνήθως μέ μία ὁθόνη CRT ἢ τηλέτυπο. Ἐπίσης, λέμε ὅτι μία διαδικασία εἶναι On-Line μέ τόν ὑπολογιστή, ἂν ἡ διαδικασία δίνει ἢ παίρνει πληροφορίες ἀπό τόν ὑπολογιστή μέ ἀπευθείας σύνδεση μ' αὐτόν (π.χ. μέσω ενός A/D ἢ D/A μετατροπέα).

Ἡ λειτουργία On-Line εἶναι ἀπαραίτητη γιά τόν ὑπολογισμό ὅταν ὁ ὄγκος τῶν δεδομένων εἶναι μεγάλος ἢ ὅταν τά δεδομένα παράγονται μέ γρήγορο ρυθμό ἢ ἀκόμη ὅταν εἶναι ἀπαραίτητος ὁ ἔλεγχος τῶν δεδομένων καί ἡ ἀπόρριψη τῶν ἀκατάλληλων πρὶν ἀπό τήν ἀποθήκευση ἢ πρὶν ἀπό τήν ἐπεξεργασία τους. Παράδειγμα ἐφαρμογῆς On-Line ἀποτελεῖ ἡ αὐτόματη λειτουργία ἐργοστασίων κατευθυνόμενων ἀπό ἓνα ἢ περισσότερους ὑπολογιστές κλπ.

9.7 Κατηγορίες λειτουργικῶν συστημάτων.

9.7.1 *Μαζική ἐπεξεργασία (Batch processing).*

Ἄν ὁ ἠλεκτρονικός ὑπολογιστής εἶναι ἐφοδιασμένος μέ λειτουργικό σύστημα μαζικῆς ἐπεξεργασίας, μπορεῖ να δεχθεῖ στήν εἴσοδό του μία ὁμάδα προγραμμάτων τό ἓνα μετά τό ἄλλο, τά ὁποῖα προγράμματα ἐπεξεργάζεται μέ τή σειρά πού παίζουν ἢ ἀνάλογα μέ τήν προτεραιότητά τους.

Στά συστήματα μαζικῆς ἐπεξεργασίας ἡ εἰσαγωγή τῶν στοιχείων γίνεται συνήθως μέσω τῆς μονάδας ἀναγνώσεως διατρήτων δελτίων.

Τό κύριο μειονέκτημα ενός τέτοιου συστήματος εἶναι ἡ χαμηλή ταχύτητα τῆς μονάδας εισόδου στοιχείων πού στήν περίπτωση αὕτή εἶναι ἡ μονάδα ἀναγνώσεως δελτίων.

Γιά να ὑπερνηκεθεῖ ἡ δυσκολία αὕτη, ἔχουν ἀναπτυχθεῖ συστήματα ἐτοιμασίας δεδομένων ἢ προγραμμάτων μέ τά ὁποῖα μπορούμε να ἀποθηκεύσουμε τά προγράμματά μας ἀπευθείας σέ δίσκους ἢ ταινίες, πού στή συνέχεια «διαβάζονται» ἀπό τίς ἀντίστοιχες μονάδες τοῦ ὑπολογιστή. Αὐτό ἔχει ὡς συνέπεια τήν αὐξηση τῆς ταχύτητας εισόδου δεδομένων στόν κεντρικό ὑπολογιστή, γιατί ἡ ταχύτητα μεταφοράς δεδομένων (transfer rate), ἀπό μία ταινία ἢ ἓνα δίσκο στόν ὑπολογιστή, εἶναι πολύ μεγαλύτερη ἀπό ὅ,τι ἀπό μία μονάδα ἀναγνώσεως δελτίων.

Πολλές φορές οἱ μονάδες αὐτές, πού εἶναι μονάδες βοηθητικῆς μνήμης, συνεργάζονται καί μέ κάποιο μικρότερο ὑπολογιστή ὁ ὁποῖος ἐκτός ἀπό τήν ἀποδοχή καί

αποθήκευση τῶν προγραμμάτων ἢ τῶν δεδομένων πρὶν ἀπὸ τὴν ἀποστολὴν τοὺς στὸν κύριο ὑπολογιστὴ, ἐκτελεῖ καὶ ἄλλες ἐργασίες βοηθητικῆς φύσεως, ὅπως ἡ ἐπαλήθευση δεδομένων καὶ ἡ διόρθωση (validation) λαθῶν, ἡ συγκέντρωση στατιστικῶν στοιχείων, ἡ ἀλλαγὴ κώδικα κλπ. Ἔτσι αὐξάνεται ἡ ἀπόδοση τοῦ ὅλου συστήματος, γιατί ἀφαιρεῖται τὸ βάρος ὅλων αὐτῶν τῶν λειτουργιῶν ἀπὸ τὸν κύριο ὑπολογιστὴ καὶ ἔτσι τοῦ ἐπιτρέπεται νὰ ἀσχολεῖται μὲ τίς κύριες λειτουργίες του.

9.7.2 Πολυπρογραμματισμός (Multiprogramming).

Μόλις τελειώσει ἡ ἐπεξεργασία ἑνὸς προγράμματος ἀπὸ τὸν ὑπολογιστὴ ἀρχίζει ἡ ἐπεξεργασία τοῦ ἐπομένου. Αὐτὸ ὅμως ἔχει ὡς συνέπεια ὅτι πολλὲς μονάδες τοῦ κεντρικοῦ ὑπολογιστῆ παραμένουν σὲ ὑποαπασχόληση, πρῶτον γιατί πολλὰ προγράμματα μπορεῖ νὰ εἶναι μικρά σὲ μέγεθος καὶ δεῦτερον λόγω τῶν ὑψηλῶν ταχυτήτων ἐπεξεργασίας καὶ τῶν πολλαπλῶν δυνατοτήτων τοῦ ὑπολογιστῆ.

Στὴν προσπάθεια νὰ μειωθεῖ ὁ χρόνος ὑποαπασχολήσεως τῶν διαφόρων μονάδων, ἀναπτύχθηκε τὸ λειτουργικὸ σύστημα τοῦ πολυπρογραμματισμοῦ τὸ ὁποῖο ἐπιτρέπει στὸν ὑπολογιστὴ νὰ ἐπεξεργάζεται συγχρόνως πολλὰ προγράμματα.

Ἔτσι, περισσότερα ἀπὸ ἓνα προγράμματα φορτώνονται στὴ μνήμη τοῦ ὑπολογιστῆ καὶ ἐκτελοῦνται συγχρόνως. Ἡ ἔννοια «συγχρόνως» εἶναι λίγο παραπαιστική. Συγχρόνως ἐννοοῦμε ὅτι κάθε πρόγραμμα ἀπασχολεῖ μία μόνο μονάδα τοῦ ὑπολογιστῆ κάθε στιγμή.

Δηλαδή ἄλλο πρόγραμμα ἀπασχολεῖ τὸ κανάλι εἰσόδου δεδομένων, ἄλλο τὸ κανάλι ἐξόδου δεδομένων καὶ ἄλλο τὴν κεντρικὴ μονάδα ἐπεξεργασίας.

Τὸ λειτουργικὸ σύστημα πολυπρογραμματισμοῦ εἶναι πολύπλοκο, δηλαδή καταλαμβάνει μεγαλύτερο μέρος στὴ μνήμη τοῦ ὑπολογιστῆ, γιατί πρέπει νὰ παρακολουθεῖ τὴν ἐκτέλεση τῶν προγραμμάτων ποὺ βρίσκονται στὴ μνήμη του, νὰ τὰ διώχνει ὅταν τελειώνουν, νὰ φέρνει νέα ποὺ χωροῦν στὴ θέση ποὺ ἄδειασε, νὰ ἐλέγχει τὴν προτεραιότητα τῶν προγραμμάτων ποὺ ἐκτελοῦνται, νὰ κρατεῖ λογιστικά καὶ στατιστικὰ στοιχεῖα γιὰ τὸ καθένα κλπ.

Στὸν πολυπρογραμματισμό, τὸ πρόγραμμα τοῦ κάθε χειριστῆ καταλαμβάνει ἓνα τμήμα τῆς μνήμης τοῦ ὑπολογιστῆ καὶ ὅλα τὰ προγράμματα ἐπεξεργάζονται ταυτόχρονα ἀπὸ τὸν ἠλεκτρονικὸ ὑπολογιστὴ. Ὁ ἀριθμὸς τῶν χειριστῶν εἶναι ἀρκετὰ περιορισμένος καὶ ἐξαρτᾶται ἀπὸ τὸ μέγεθος τῆς κύριας μνήμης τοῦ ὑπολογιστῆ καὶ ἀπὸ τὸ μέγεθος τῶν προγραμμάτων.

9.7.3 Τηλεεπεξεργασία (Teleprocessing).

Τὰ λειτουργικὰ συστήματα τὰ ὁποῖα ἐξετάσαμε παρουσιάζουν τὸ μειονέκτημα ὅτι ὁ χειριστὴς ἢ ὁ προγραμματιστὴς πρέπει νὰ ὑποβάλλουν τὰ προγράμματά τους καὶ νὰ ἔλθουν ἀργότερα νὰ τὰ πάρουν, χωρὶς νὰ ἔχουν καμία ἀπευθείας ἐπαφὴ μὲ τὸν ἴδιο τὸν ἠλεκτρονικὸ ὑπολογιστὴ.

Τὴν ἀνάπτυξη τῶν ἠλεκτρονικῶν ὑπολογιστῶν ἀκολούθησε ἀπὸ κοντὰ καὶ ἡ ἀνάπτυξη τῶν διαφόρων περιφερειακῶν μονάδων εἰσόδου-ἐξόδου δεδομένων καὶ ἡ ἀνάπτυξη τῶν συστημάτων ἐπικοινωνίας τῶν μονάδων μὲ ἓναν ἀπομακρυσμένο ὑπολογιστὴ. Ἔτσι ἀναπτύχθηκε ἡ τηλεεπεξεργασία κατὰ τὴν ὁποία οἱ διάφορες τερματικές μονάδες οἱ ὁποῖες βρίσκονται μακριὰ ἀπὸ τὸν ὑπολογιστὴ ἐπικοινωνοῦν μ' αὐτόν μέσω γραμμῶν τηλεπικοινωνίας, ὅπως εἶναι οἱ τηλεφωνικὲς γραμμές, ὁ ἀσύρματος, ἡ μικροκυματικὴ σύνδεση καὶ ἄλλα.

“Ετσι ο προγραμματιστής ή ο χειριστής μπορεί και «συνομιλεί» με τον ηλεκτρονικό υπολογιστή. Δηλαδή υπάρχει η δυνατότητα αλληλεπιδράσεως (interaction) μεταξύ χειριστή και υπολογιστή. “Ένα λάθος του χειριστή παραδείγματος χάρη αναγνωρίζεται άμέσως από τον υπολογιστή και ειδοποιείται ο χειριστής να κάνει τη διόρθωση. Μέ αυτό τον τρόπο παρέχεται η δυνατότητα να εργάζονται πολλοί χειριστές ταυτόχρονα με τον ίδιο κεντρικό υπολογιστή.

Τά λειτουργικά συστήματα που επιτρέπουν την άμοιβαία επικοινωνία μεταξύ χειριστών και υπολογιστή είναι, τό σύστημα καταμερισμού χρόνου (Time sharing) και τό σύστημα πραγματικού χρόνου (Real time).

Στόν προγραμματισμό είπαμε ότι τό πρόγραμμα του κάθε χειριστή καταλαμβάνει ένα τμήμα της μνήμης του υπολογιστή και όλα τά προγράμματα επεξεργάζονται ταυτόχρονα από τον ηλεκτρονικό υπολογιστή. “Ο αριθμός των χειριστών είναι αρκετά περιορισμένος και εξαρτάται από τό μέγεθος της κύριας μνήμης του υπολογιστή και από τό μέγεθος των προγραμμάτων.

“Εάν T είναι ο χρόνος που διατίθεται για την είσοδο ή την έξοδο των πληροφοριών προς ή από τον υπολογιστή με όλες τις N τερματικές μονάδες, τότε φυσικά ο χρόνος που κάθε τερματική μονάδα είναι συνδεδεμένη με τον υπολογιστή είναι της τάξεως των T/N δευτερολέπτων. “Ο χρόνος αυτός είναι αρκετός γιατί αν υπολογίσουμε ότι ένας γρήγορος χειριστής μιås τερματικής μονάδας με ηλεκτρολόγιο στέλνει 80 περίπου χαρακτήρες τό λεπτό, τότε, αν κάθε χαρακτήρας κωδικοποιείται σε 8 ψηφία/χαρακτήρα, έχουμε 640 ψηφία/λεπτό ή περίπου 10.6 ψηφία/δευτερόλεπτο. “Ο υπολογιστής όμως μπορεί να δέχεται π.χ. 5 Mbytes/δευτ. (40 Mbits/δευτ.). “Επομένως μέσα σε ένα δευτερόλεπτο μπορεί να δεχθεί είσοδο από πολλές περιφερειακές μονάδες. Στήν πραγματικότητα ο αριθμός αυτός δέν είναι τόσο μεγάλος, γιατί ή ταχύτητα απόστολής μέσω των διαύλων επικοινωνίας κυμαίνεται από 110 – 19200 ψηφία/δευτ. με συνήθη ταχύτητα τά 2400 ψηφία/δευτ. “Ετσι, ο αριθμός των μονάδων που μπορούν να συνδεθούν σε έναν υπολογιστή φθάνει τις μερικές εκατοντάδες.

Μέ τό σύστημα καταμερισμού χρόνου καμία περιφερειακή μονάδα δέν έχει προτεραιότητα. “Όλες ελέγχονται από τό κεντρικό υπολογιστή με τη σειρά. Σύμφωνα με τό σύστημα καταμερισμού χρόνου, (time sharing) ο χρόνος διαιρείται σε μικρά χρονικά διαστήματα και για κάθε χρονικό διάστημα εξυπηρετείται ένας μόνο χειριστής. Ουσιαστικά δηλαδή διαθέτουμε ένα χρονικό διάστημα t δευτερολέπτων για την επικοινωνία του υπολογιστή με μία περιφερειακή μονάδα, τά επόμενα t δευτερόλεπτα διατίθενται για την επικοινωνία με τη δεύτερη περιφερειακή μονάδα κλπ.

“Εάν κάποιο τερματικό δέν είναι συνδεδεμένο (δηλαδή δέν στέλνει ή δέν δέχεται πληροφορίες), τότε ο χρόνος ο οποίος ήταν αφιερωμένος γι’ αυτό, μοιράζεται στα άλλα που είναι συνδεδεμένα.

“Ετσι έχουμε τη δυνατότητα της εξυπηρετήσεως πολύ περισσότερων χρηστών.

“Εννοείται ότι τό σύστημα πολυπρογραμματισμού μπορεί να συνυπάρχει με τό σύστημα καταμερισμού χρόνου.

“Ο χρόνος που χρειάζεται ένας υπολογιστής για να ανταποκριθεί στις απαιτήσεις ενός χρήστη, δηλαδή ο χρόνος για τη συμπλήρωση της εργασίας του, εξαρτάται από τό φορτίο του υπολογιστή που είναι συνάρτηση του αριθμού των χρηστών και του είδους των εργασιών που ο καθένας ζητά. “Υπάρχουν περιπτώσεις που ή αντα-

πόκριση του υπολογιστή πρέπει να είναι άμεση. Π.χ. όταν ένας υπολογιστής υπολογίζει τα στοιχεία της τροχιάς ενός πυραύλου που ελέγχει, πρέπει να είναι συνδεδεμένος με διάφορα σημεία του πυραύλου, να συλλέγει στοιχεία και να τα επεξεργάζεται μέσα σε χρόνο ο οποίος να ανταποκρίνεται στην ταχύτητα κινήσεως του πυραύλου.

Σ' αυτές τις περιπτώσεις λέμε ότι έχουμε λειτουργία του υπολογιστή σε «Πραγματικό Χρόνο» (Real Time).

9.8 Έρωτήσεις.

1. Ποιός ο λόγος ύπαρξης του λειτουργικού συστήματος ενός Η/Υ.
2. Ποιές εργασίες εκτελεί τό πρόγραμμα εκκινήσεως;
3. Ποιές εργασίες εκτελεί τό πρόγραμμα φορτώσεως;
4. Ποιές εργασίες εκτελούν τά προγράμματα οδηγίσεως;
5. Τί σημαίνει λειτουργία Off-Line; On-Line. Δώστε παραδείγματα.
6. Ποιός είναι ο τρόπος λειτουργίας ενός Η/Υ εφοδιασμένου με λειτουργικό σύστημα μαζικής επεξεργασίας. Ποιά τά μειονεκτήματά του;
7. Ποιός ο τρόπος λειτουργίας ενός Η/Υ εφοδιασμένου με λειτουργικό σύστημα πολυπρογραμματισμού; Ποιά τά πλεονεκτήματα ή μειονεκτήματά του;
8. Έξηγηστε τόν δρο τηλεπεξεργασία.
9. Πώς εξηγούνται οι δροι: «καταμερισμός χρόνου» και «πραγματικός χρόνος» όταν αυτοί αναφέρονται σε λειτουργικά συστήματα υπολογιστών.
10. Μπορεί ένα λειτουργικό σύστημα τηλεπεξεργασίας να είναι ταυτόχρονα και σύστημα πολυπρογραμματισμού; Έξηγηστε τήν απάντησή σας.

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ

ΜΙΚΡΟ-ΥΠΟΛΟΓΙΣΤΕΣ

Μία από τις μεγαλύτερες κατακτήσεις της τεχνολογίας είναι η κατασκευή των ολοκληρωμένων κυκλωμάτων μεγάλης πυκνότητας (Large Scale Integration-LSI).

Τά κυκλώματα αυτά κατασκευάζονται με διάφορες φωτογραφικές και χημικές μεθόδους επάνω σε ένα πολύ μικρό κομμάτι (φέτα) κρυστάλλου από πυρίτιο (συνήθεις διαστάσεις 4 mm²). Η πυκνότητα των κυκλωμάτων αυτών επιτρέπει την κατασκευή πάνω στο ίδιο κομμάτι πυριτίου αρκετών χιλιάδων λογικών και άλλων ηλεκτρονικών κυκλωμάτων.

Στήν αρχή σε ένα ολοκληρωμένο κύκλωμα κατασκευαζόταν ένα τμήμα της μονάδας επεξεργασίας ενός υπολογιστή π.χ. ένας άπαριθμητής ή ένας συσσωρευτής. Για την κατασκευή της αριθμητικής και λογικής μονάδας π.χ. έπρεπε να γίνει η σύνδεση ενός αριθμού τέτοιων κυκλωμάτων, τα οποία συνήθως ήταν τοποθετημένα πάνω σε μία πλακέτα τυπωμένου κυκλώματος. Με την πρόοδο της τεχνολογίας έγινε δυνατή η κατασκευή, πάνω στο ίδιο κομμάτι κρυστάλλου, ολόκληρης της αριθμητικής και λογικής μονάδας (ALU) ενός υπολογιστή.

Στή συνέχεια έγινε δυνατή η κατασκευή πάνω στο ίδιο κομμάτι κρυστάλλου όχι μόνο της ALU αλλά και της μονάδας ελέγχου του υπολογιστή, που ονομάσαμε στα προηγούμενα «Κεντρική Μονάδα Έπεξεργασίας». Η μονάδα αυτή πήρε το όνομα «Μικροεπεξεργαστής» (Microprocessor ή μP).

Οι πρώτοι μικροεπεξεργαστές είχαν μήκος λέξεως 4 δυαδικών ψηφίων. Σήμερα κατασκευάζονται μικροεπεξεργαστές με μήκος λέξεως 8 και 16 δυαδικών ψηφίων, ενώ βρίσκονται στο τελικό στάδιο παραγωγής μικροεπεξεργαστές των 32 δυαδικών ψηφίων.

Οι «φέτοιδοι» (bit-sliced) μικροεπεξεργαστές, είναι μικροεπεξεργαστές του ενός ή των δύο δυαδικών ψηφίων, έχουν όμως τη δυνατότητα να συνδεθούν μεταξύ τους έτσι, ώστε να αποτελέσουν μία μονάδα με μήκος λέξεως όσο θέλουμε. Όπως έχουμε πει αναπτύχθηκαν επίσης και οι ηλεκτρονικές μνήμες μεγάλης πυκνότητας, καθώς επίσης και οι μονάδες για την είσοδο και την έξοδο πληροφοριών από έναν υπολογιστή. Έτσι με τη σύνδεση ενός μικρού αριθμού ολοκληρωμένων κυκλωμάτων μεγάλης πυκνότητας είναι σήμερα δυνατή η κατασκευή υπολογιστών που έχουν υπολογιστική ισχύ πολύ μεγαλύτερη από τους πρώτους ηλεκτρονικούς υπολογιστές. Η ισχύ τους πλησιάζει την υπολογιστική ισχύ των mini-υπολογιστών.

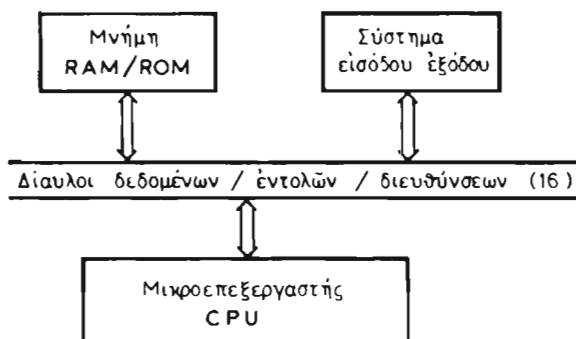
Οι διατάξεις αυτές είναι γνωστές ως «μικρο-υπολογιστές» (Microcomputers-μC). Άς σημειωθεί ότι σήμερα σε ένα ολοκληρωμένο κύκλωμα κατασκευάζονται αυτότελες μικρο-υπολογιστές οι οποίοι διαθέτουν μνήμη RAM μέχρι 1Kbyte και μνήμη ROM μέχρι 2 Kbytes.

Ἡ τεράστια ανάπτυξη καὶ διάδοση περιφερειακῶν μονάδων, ὅπως εἶναι ὀθόνες CRT, ἐκτυπωτές, μίνι-δισκέτες κλπ. συνετέλεσε, ὥστε ὅλο τὸ κόστος ἑνὸς μικρο-ὕπολογιστῆ με μνήμη 64 Kbytes, με δύο δισκέτες, με ἓνα ἐκτυπωτὴ καὶ μία ὀθὸν CRT, νὰ μὴν ξεπερνᾷ τίς 300000 δρχ.

Ταυτόχρονα ἔχουν ἀναπτυχθεῖ λειτουργικὰ συστήματα, μεταφραστές διαφορῶν γλωσσῶν καὶ προγράμματα ἐφαρμογῶν γιὰ τοὺς μικρο-ὕπολογιστές, οἱ ὁποῖο σήμερα χρησιμοποιοῦνται τόσο ὡς ἀνεξάρτητα συστήματα ἐπεξεργασίας πληροφοριῶν ὅσο καὶ ὡς εἰδικοί ὑπολογιστές ἐλέγχου διαδικασιῶν. Ἦδη σὲ συσκευές τηλεοράσεως, στὰ ἠλεκτρικὰ πλυντήρια, στὰ αὐτοκίνητα, σὲ διάφορα παιχνίδια κλπ. χρησιμοποιοῦνται μικρο-ὕπολογιστές.

Στὸ σχῆμα 10.1 δίνεται ἡ διάρθρωση ἑνὸς μικρο-ὕπολογιστικοῦ συστήματος στὸ ὁποῖο φαίνεται ὁ μικροεπεξεργαστής, οἱ μονάδες εἰσόδου/ἐξόδου καὶ ἡ μνήμη.

Ἐκεῖνο πού θά πρέπει νὰ σημειωθεῖ εἶναι ὁ τρόπος με τὸν ὁποῖο συνδέονται τὰ διάφορα τμήματα τοῦ μικρο-ὕπολογιστῆ. Δηλαδή, ἡ χρησιμοποίησις ἑνὸς κοινοῦ «δρόμου» (bus) γιὰ τὴ μεταφορὰ τῶν δεδομένων (data bus), καὶ ἑνὸς κοινοῦ δρόμου γιὰ τὴ μεταφορὰ τῶν διευθύνσεων (address bus). Ὑπάρχουν ἐπίσης καὶ ὀρισμένες ἄλλες γραμμές πού εἶναι γνωστές ὡς γραμμές ἢ δρόμοι «ἐλέγχου» (Control bus).



Σχ. 10.1.

Μικροὕπολογιστικό σύστημα.

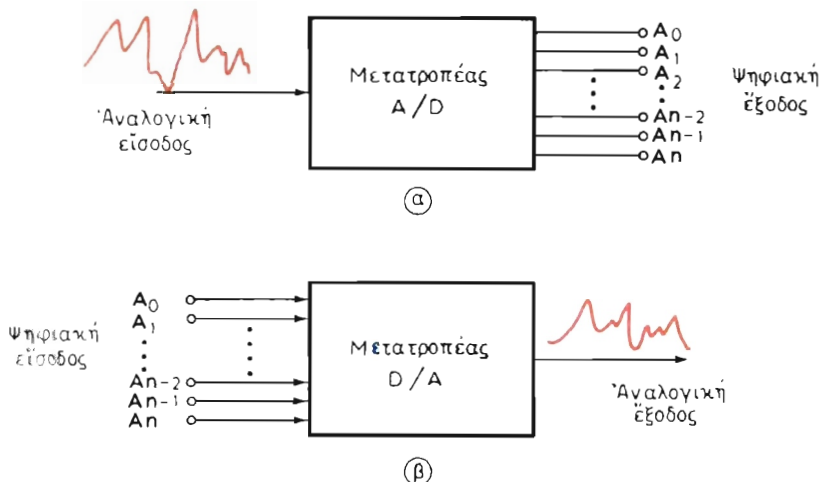
ΚΕΦΑΛΑΙΟ ΕΝΔΕΚΑΤΟ

ΜΕΤΑΤΡΟΠΗ ΑΝΑΛΟΓΙΚΩΝ ΣΗΜΑΤΩΝ ΣΕ ΨΗΦΙΑΚΑ

11.1 Γενικά.

Η μετατροπή ενός αναλογικού σήματος, π.χ. της φωνής του ανθρώπου, της θερμοκρασίας, της πίεσης κλπ., σε ψηφιακό (Analog to digital conversion) και η αντίστροφη μετατροπή ενός ψηφιακού σήματος σε αναλογικό (Digital to analog conversion) είναι από τις σπουδαιότερες επεξεργασίες του Η/Υ.

Τό αποτέλεσμα της επεξεργασίας ενός σήματος από τον ηλεκτρονικό υπολογιστή, δέν μπορεί νά εφαρμοσθεί άπευθείας σέ μία συσκευή πού έλέγχει ό υπολογιστής χωρίς νά μετατραπεί από τόν μετατροπέα σε αναλογικό και μάλιστα μέ τήν κατάλληλη μορφή π.χ. τάση, ρεύμα κλπ. (σχ. 11.1).



Σχ. 11.1.

Σχηματική παράσταση α) ενός μετατροπέα A/D και β) ενός μετατροπέα D/A.

11.2 Μετατροπείς A/D.

Οι μετατροπείς A/D αποτελούν μία από τις βασικές μονάδες του **συστήματος συλλογής δεδομένων από φυσικές διαδικασίες**, πού συνδέονται μέ υπολογιστή. Π.χ. ό έλεγχος ενός έργουστασίου αυτόματου κατασκευής εξαρτημάτων, ό έλεγχος

της πορείας ενός πυραύλου κλπ. Αυτό τό κύκλωμα βασικά γεφυρώνει τό κενό μεταξύ τών συνεχών παραμέτρων του πραγματικού κόσμου καί του ψηφιακού κόσμου τών υπολογιστών.

Υπάρχουν πολλές μέθοδοι μετατροπής ενός αναλογικού σήματος σε ψηφιακό.

Οι σπουδαιότερες είναι:

- 'Η μέθοδος της ολοκληρώσεως (integration method).
- 'Η μέθοδος της άμεσης συγκρίσεως (comparison method) και
- ή μέθοδος τών διαδοχικών προσεγγίσεων (successive approximation method).

'Η μετατροπή στο μετατροπέα A/D γίνεται ως εξής:

'Εφαρμόζομε στην είσοδο τό αναλογικό σήμα καί παίρνομε στην έξοδο μία διαδοχή από δυαδικούς αριθμούς πού αντιπροσωπεύουν τήν τιμή του σήματος σε ίσαπεχουσες χρονικές στιγμές. Τά κύρια χαρακτηριστικά έπομένως ενός μετατροπέα A/D είναι ό αριθμός τών δειγμάτων πού παίρνει από τό αναλογικό σήμα στή μονάδα του χρόνου, καθώς καί τό πλήθος τών ψηφίων μέ τό όποιο παριστάνει τό κάθε δείγμα. Π.χ. ένας μετατροπέας A/D του όποιου ή έξοδος αποτελείται από 3 μόνο δυαδικά ψηφία, θά μπορεί νά μάς δώσει, γιά τό αναλογικό σήμα πού εφαρμόζομε στην είσοδο, μόνο $2^3 = 8$ διαφορετικές τιμές. Αυτό ίσοδυναμεί μέ τό νά μετρούμε τίς συντεταγμένες τών σημείων μιάς καμπύλης μέ ένα χάρακα ό όποιος έχει μόνο 8 άραιές υποδιαιρέσεις. Λέμε τότε ότι ό μετατροπέας αυτός έχει χαμηλή διακριτικότητα (resolution).

11.2.1 'Η μέθοδος της ολοκληρώσεως.

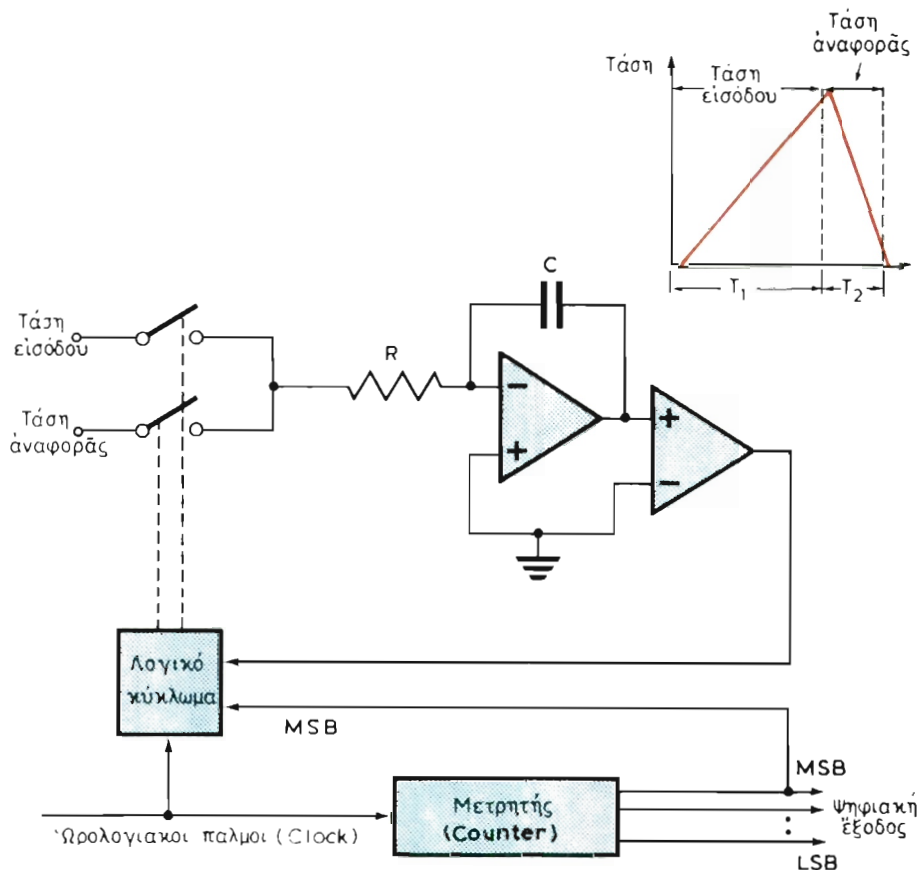
'Η βάση της μεθόδου αυτής είναι ό χρόνος πού χρειάζεται ένας πυκνωτής γιά νά φορτισθεί μέχρι μία άγνωστη τάση καί μετά νά έκφορτισθεί μέχρι μία τάση αναφοράς. Δηλαδή, φορτίζομε τόν πυκνωτή μέ τήν άγνωστη τάση, όποτε τό φορτίο του πυκνωτή αύξάνεται. Μετά από ένα χρονικό διάστημα, εφαρμόζομε μία γνωστή άρνητική τάση αναφοράς στον πυκνωτή καί μετρούμε τό χρόνο πού χρειάζεται γιά νά φθάσει τό φορτίο του στή τάση αυτή (σχ. 11.2α). 'Η τεχνική της ολοκληρώσεως μάς δίνει μεγάλη ακρίβεια καί έπιπλέον έλαχιστοποιεί τίς επιδράσεις του θορύβου. Τό μειονέκτημά της είναι ότι είναι πολύ άργή, δηλαδή μπορεί νά πάρει μόνο λίγα δείγματα ανά δευτερόλεπτο. Στο σχήμα 11.2α φαίνεται τό διάγραμμα ενός μετατροπέα A/D μέ τή μέθοδο της ολοκληρώσεως.

11.2.2 'Η μέθοδος της άμεσης συγκρίσεως.

'Η μέθοδος αυτή χρησιμοποιείται όταν απαιτούνται μεγάλες ταχύτητες μετατροπής ($< 1 \text{ sec}$) ανά δείγμα καί όταν ή έξοδος του μετατροπέα δέ χρειάζεται νά έχει πολλά ψηφία. Στο σχήμα 11.2β φαίνεται τό διάγραμμα ενός μετατροπέα A/D μέ τή μέθοδο της άμεσης συγκρίσεως.

Τό κύκλωμα όπως φαίνεται από τό σχήμα, αποτελείται από $2^n - 1$ συγκριτές, όπου η τό πλήθος τών δυαδικών ψηφίων πού δίνει ό μετατροπέας στην έξοδο. Οι συγκριτές αυτοί συγκρίνουν συγχρόνως τήν αναλογική τάση εισόδου μέ όρισμένες τάσεις αναφοράς.

Γιά παράδειγμα, άς πάρομε ένα μετατροπέα μέ 3 συγκριτές. 'Η είσοδος μπορεί νά μετρηθεί μέ τή βοήθεια $2^3 = 8$ «επιπέδων» αναφοράς. Έτσι μέ 7 συγκριτές κα-



Σχ. 11.2α.

Το διάγραμμα ενός μετατροπέα A/D με τη μέθοδο της ολοκληρώσεως.

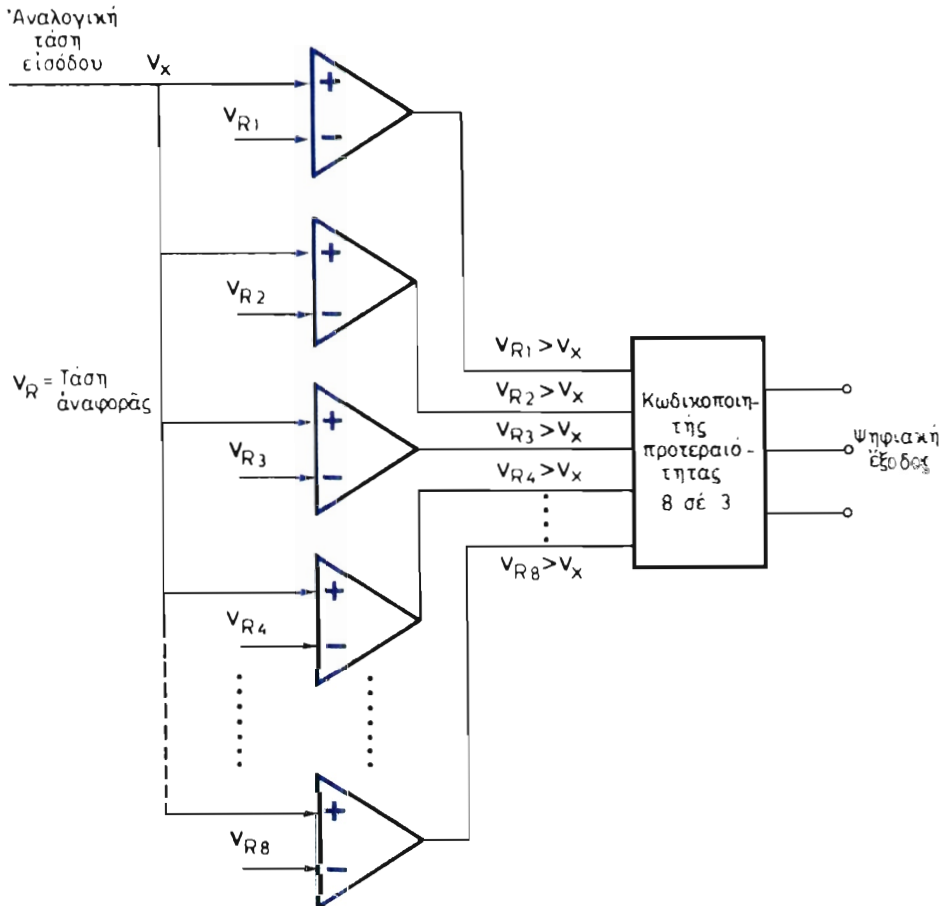
Θορίζεται αν η τάση εισόδου είναι μεγαλύτερη ή μικρότερη από καθεμιά από τις 8 τιμές αναφοράς.

11.2.3 Η μέθοδος των διαδοχικών προσεγγίσεων.

Η μέθοδος αυτή χρησιμοποιείται εύρεως επειδή η αναλογία κόστους - ταχύτητας είναι πολύ καλή. Η τεχνική των διαδοχικών προσεγγίσεων λειτουργεί ως εξής:

Τό αναλογικό σήμα εισόδου συγκρίνεται με μία τάση αναφοράς, ίση ακριβώς με τη μισή τιμή του εύρους κλίμακας εισόδου του μετατροπέα. Εάν π.χ. ο μετατροπέας λειτουργεί για σήματα εισόδου κυμαινόμενα από 0V ως 10V, τότε η τάση αναφοράς είναι 5V. Εάν λοιπόν τό σήμα αναφοράς είναι μικρότερο από τό σήμα εισόδου, τό πρώτο ψηφίο στη ψηφιακή λέξη γίνεται «1». Εάν είναι μεγαλύτερο γίνεται «0». Αυτό τό πρώτο ψηφίο είναι καί τό περισσότερο σημαντικό ψηφίο.

Στή συνέχεια η τάση αναφοράς υποδιπλασιάζεται. Αν τό πρώτο ψηφίο είναι 1, ή νέα τιμή τής τάσεως αναφοράς προστίθεται στήν παλιά. Με άλλα λόγια τό δεύτε-

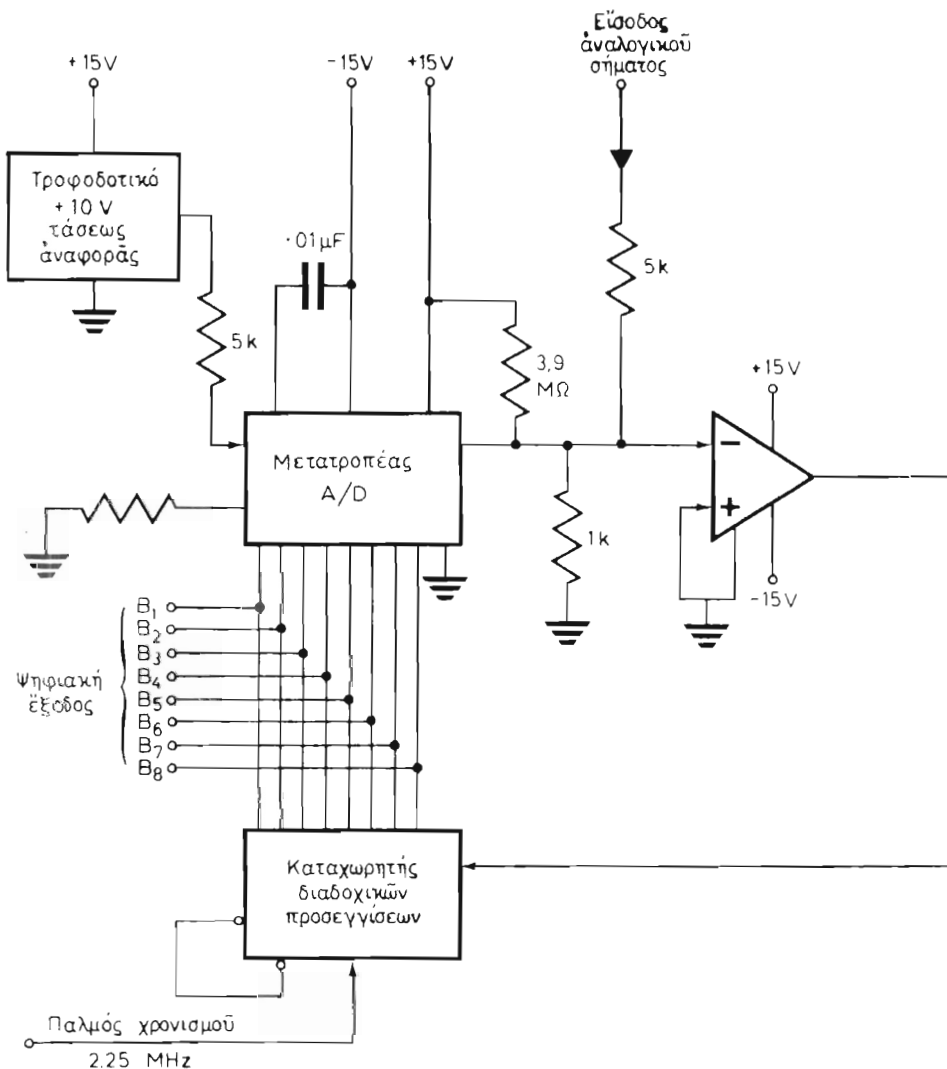


Σχ. 11.28.

Διάγραμμα ενός μετατροπέα A/D με τη μέθοδο της άμεσης συγκρίσεως.

ρο βήμα συγκρίνει το σήμα εισόδου με μία ποσότητα, τόση δση με τό μισό του εύρους κλίμακας του μετατροπέα σύν $1/4$ του εύρους. Όταν η πραγματική τιμή ξεπερνά την τιμή αναφοράς, το αντίστοιχο ψηφίο γίνεται «1». Κάθε διαδοχική σύγκριση προσθέτει $1/2$ της προηγούμενης τιμής που προστέθηκε ($1/8, 1/16, 1/32, 1/64$ κλπ.) στο προηγούμενο σύνολο, με την προϋπόθεση ότι το προηγούμενο ψηφίο είναι «1». Εάν σε κάποια σύγκριση η πραγματική τιμή είναι μικρότερη από την τιμή συγκρίσεως, το αντίστοιχο ψηφίο γίνεται «0» και η τελευταία προστεθείσα τιμή στην τάση αναφοράς παραλείπεται από την επόμενη σύγκριση. Αν π.χ. τα δύο πρώτα ψηφία ήταν «11», τότε η τιμή συγκρίσεως στο τρίτο βήμα θα ήταν $1/2 + 1/4 + 1/8$. Αν όμως τα δύο πρώτα ψηφία ήταν «10», τότε η τιμή συγκρίσεως στο τρίτο βήμα θα ήταν $1/2 + 1/8$.

Αυτή η διαδικασία επαναλαμβάνεται η φορές για ένα μετατροπέα των n ψηφίων, προσεγγίζοντας την τιμή του σήματος εισόδου με συνεχώς μικρότερα βήματα. Έτσι, για ένα μετατροπέα των 12 ψηφίων, το σήμα συγκρίσεως (δηλαδή το εύ-



Σχ. 11.2γ.

Τό λογικό διάγραμμα ενός μετατροπέα A/D με τη μέθοδο των διαδοχικών προσεγγίσεων.

ρος της κλίμακας εισόδου) διαιρείται σε $2^n = 2^{12} = 4096$ μέρη. Δηλαδή η μικρότερη τιμή συγκρίσεως είναι $1/4096$ του εύρους του σήματος εισόδου.

Στό σχήμα 11.2γ φαίνεται τό διάγραμμα ενός μετατροπέα A/D με τη μέθοδο των διαδοχικών προσεγγίσεων.

11.3 Μετατροπείς D/A.

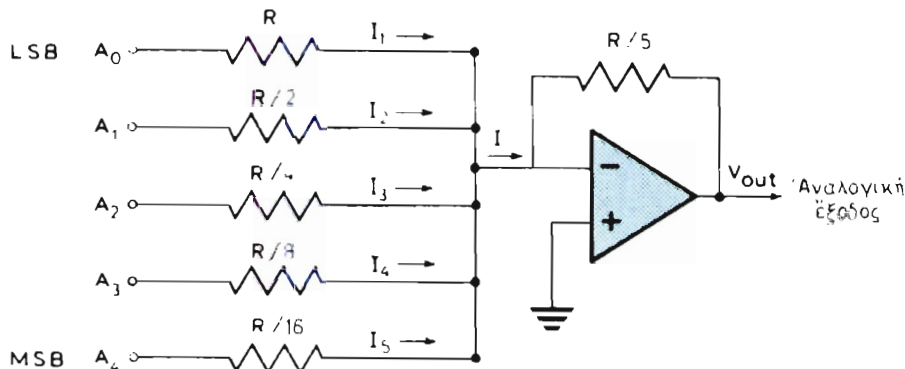
Γιά τήν αντίθετη τώρα μετατροπή, δηλαδή γιά τήν μετατροπή από τό ψηφιακό

σύστημα στο αναλογικό διακρίνουμε τούς παρακάτω βασικούς τρόπους:

- Μετατροπή D/A με χρήση αντιστάσεων διαφόρων βαρών (weight - resistor network).
- Μετατροπή D/A με χρήση «κλιμακωτών» κυκλωμάτων (ladder networks).

11.3.1 Μετατροπή D/A με χρήση αντιστάσεων διαφόρων βαρών.

Ένας τέτοιος μετατροπέας φαίνεται στο σχήμα 11.3α. Τό κύκλωμα αυτό περιέχει ένα σύνολο αντιστάσεων στις οποίες εφαρμόζεται τό ψηφιακό σήμα.



Σχ. 11.3α.

Κύκλωμα ενός μετατροπέα D/A με χρήση αντιστάσεων διαφόρων βαρών.

Όπως βλέπουμε οι αντιστάσεις αυτές έχουν διάφορες τιμές (βάρη) σύμφωνα με τό δυαδικό κώδικα ($R = 2^1$, $R/2 = 2^2$, $R/4 = 2^4 \dots$ κ.ο.κ.).

Τό κύκλωμα αυτό λειτουργεί όπως ένας αναλογικός άθροιστής. Ή κατασκευή του είναι σχετικά απλή. Άς υποθέσουμε π.χ. ότι $R = 500 \text{ K}\Omega$ καί ότι ή λογική κατάσταση «1» αντιστοιχεί σέ τάση $+5\text{V}$, ενώ ή λογική κατάσταση «0» αντιστοιχεί σέ τάση 0V . Άς υποθέσουμε ακόμη ότι ή ψηφιακή είσοδος είναι ή 10001. Τότε θα έχουμε:

$$\begin{aligned}
 I &= I_1 + I_2 + I_3 + I_4 + I_5 \\
 &= \frac{5\text{V}}{500\text{K}} + \frac{0\text{V}}{\frac{500\text{K}}{2}} + \frac{0\text{V}}{\frac{500\text{K}}{4}} + \frac{0\text{V}}{\frac{500\text{K}}{8}} + \frac{5\text{V}}{\frac{500\text{K}}{16}} = \\
 &= 1 \times 10^{-5} + 0 + 0 + 0 + 1.6 \cdot 10^{-4} \text{ A} \\
 &= 1.7 \cdot 10^{-4} \text{ A} = 0.17 \text{ mA}
 \end{aligned}$$

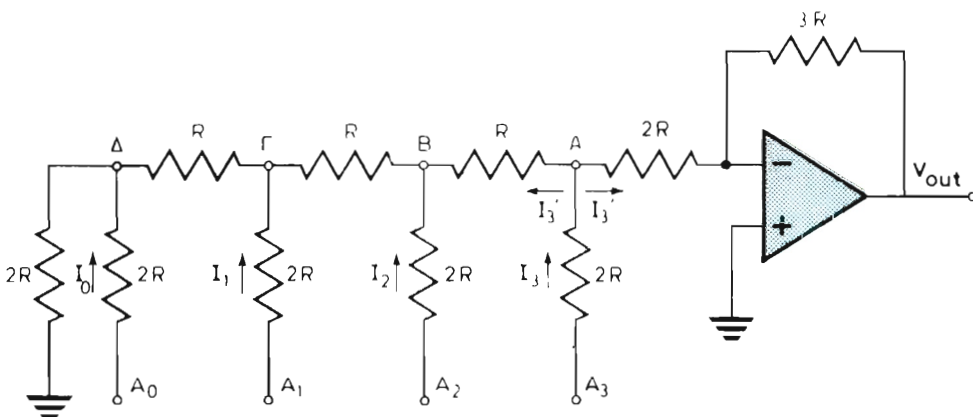
Επομένως: $V_{\text{out}} = -0.17 \times 500\text{K}/5 = -17 \text{ Volt} (= 10001_{(2)})$

Τό κυριότερο μειονέκτημα τού κυκλώματος αυτού είναι ή χρησιμοποίηση αντίστασεων ακρίβειας, οι οποίες δέν θά πρέπει νά αλλάζουν μέ τήν αλλαγή θερμοκρασίας, πράγμα αρκετά δύσκολο νά επιτευχθεί. Άλλο μειονέκτημα είναι ότι οι αντιστάσεις έχουν βάρη καί έτσι δύσκολα μπορούν νά κατασκευασθούν. Γιατί όταν

χρησιμοποιούμε ψηφιακές λέξεις μεγάλου μήκους ο λόγος της μεγαλύτερης προς τη μικρότερη αντίσταση είναι σημαντικός. Αν π.χ. χρησιμοποιούμε ψηφιακές λέξεις των 12 ψηφίων, τότε η μικρότερη αντίσταση είναι $R/2^{12-1} = R/2^{11}$. Αν αυτή η αντίσταση πρέπει να είναι 1 KΩ, τότε η μεγαλύτερη αντίσταση θα είναι 2048 KΩ = 2.048 MΩ. Φυσικά τέτοια αντίσταση είναι και δύσκολο αλλά και δαπανηρό να κατασκευασθεί.

11.3.2 Μετατροπή D/A με χρήση κλιμακωτών κυκλωμάτων.

Γιά να αποφευχθούν τα μειονεκτήματα του προηγούμενου μετατροπέα D/A, ε-
πινοήθηκε τό κύκλωμα του σχήματος 11.3β, τό όποιο περιέχει αντιστάσεις μέ δύο
μόνο τιμές R καί 2R.



Σχ. 11.3β.

Διάγραμμα ενός μετατροπέα D/A με χρήση κλιμακωτών κυκλωμάτων.

Η λειτουργία του κυκλώματος αυτού εξηγείται ως εξής:

Αν στην είσοδο του πιο σημαντικού ψηφίου A_3 εφαρμοσθεί η τάση $V_R = 5V$ (που αντιστοιχεί στο λογικό «1») και αν V_A είναι η τάση στον κόμβο A, τότε θα έχουμε:

$$I_3' = \frac{V_A}{2R}$$

Τό ρεύμα I_3 που διαρρέει τόν κατακόρυφο κλάδο του κόμβου A θα είναι επομένως, σύμφωνα μέ τό νόμο του Kirchhoff:

$$I_3 = I_3' + I_3' = 2I_3' = \frac{2V_A}{2R} = \frac{V_A}{R}$$

Τό ρεύμα I_3 όμως δίνεται καί από τή σχέση:

$$I_3 = \frac{V_R - V_A}{2R}$$

Επομένως:

$$\frac{V_R - V_A}{2R} = \frac{V_A}{R}$$

καί επιλύοντας ως προς V_A βρίσκουμε ότι:

$$V_A = \frac{V_R}{3}$$

Επομένως στην έξοδο του τελεστικού ενισχυτή ή τάση V_{out} θα είναι:

$$V_{out} = V_A \left(\frac{-3R}{2R} \right) = \frac{V_R}{3} \left(\frac{-3R}{2R} \right) = -\frac{V_R}{2}$$

Με τον ίδιο τρόπο, αν στην είσοδο του δεύτερου πύλο σημαντικού ψηφίου A_2 εφαρμόσουμε την τάση $V_R = 5V$ (ενώ τις άλλες εισόδους τις αφήσουμε ανοικτές) θα βρούμε ότι:

$$I_2' = \frac{V_B}{2R}$$

καί:

$$I_2 = \frac{V_B}{R}$$

καί η τάση εξόδου V_{out} θα είναι:

$$V_{out} = -\frac{V_R}{4}$$

Αν τώρα στην είσοδο του τρίτου πύλο σημαντικού ψηφίου εφαρμόσουμε την τάση $V_R = 5V$, θα βρούμε:

$$I_1' = \frac{V_\Gamma}{2R}$$

$$\text{καί } I_1 = \frac{V_\Gamma}{R}$$

$$\text{καί } V_{out} = -\frac{V_R}{8}$$

Με όμοιο τρόπο για την είσοδο του λιγότερο σημαντικού ψηφίου θα βρούμε:

$$V_{out} = -\frac{V_R}{16}$$

Εάν σέ όλες τίς εισόδους εφαρμοσθεῖ ταυτόχρονα ἡ τάση $V_R = 5V$, θά ἔχομε ὅτι ἡ τάση ἐξόδου V_{out} , σύμφωνα μέ τό νόμο τῆς ὑπερθέσεως, θά εἶναι:

$$V_{out} = - \left(\frac{V_R}{2} + \frac{V_R}{4} + \frac{V_R}{8} + \frac{V_R}{16} \right) = - \frac{15}{16} V_R$$

Εάν σέ όλες τίς εισόδους εφαρμοσθεῖ ταυτόχρονα ἡ τάση $V_R = 0V$ (πού ἀντιστοιχεῖ στό λογικό «0»), τότε:

$$V_{out} = 0V$$

Εάν τώρα εφαρμοσθεῖ στήν εἴσοδο τοῦ μετατροπέα ὁ δυαδικός συνδυασμός 1101 τοῦ προηγούμενου παραδείγματος, θά ἔχομε:

$$V_{out} = - \left(\frac{V_R}{2} + \frac{V_R}{4} + 0 + \frac{V_R}{16} \right) = - \frac{11}{16} V_R$$

Από τά παραπάνω μπορούμε νά συμπεράνομε ὅτι μέ τό μετατροπέα αὐτό μπορούμε νά μετατρέψομε ψηφιακά σήματα σέ ἀναλογικά καί σέ βήματα τῶν $1/16 V_R$ Volts. Ἐπομένως ἡ διακριτικότητα τοῦ μετατροπέα αὐτοῦ εἶναι: $1/16 V_R$ Volt.

11.4 Ἑρωτήσεις.

1. Γιά ποιό λόγο ἀπαιτεῖται ἡ ὕπαρξη τῶν ἀναλογοψηφιακῶν μετατροπέων;
2. Ποιές εἶναι οἱ συνήθως χρησιμοποιούμενες μέθοδοι μετατροπῆς ἀπό τό ἀναλογικό στό ψηφιακό σύστημα;
3. Περιγράψατε τή μέθοδο μετατροπῆς A/D μέ διαδοχικές προσεγγίσεις.
4. Ποιοί εἶναι οἱ βασικοί τρόποι μετατροπῆς ἀπό τό ψηφιακό στό ἀναλογικό σύστημα; Ποῦ ἀπαιτεῖται αὕτη ἡ μετατροπή;
5. Ποιά εἶναι τά μειονεκτήματα τῶν μετατροπέων D/A μέ χρήση ἀντιστάσεων διαφόρων βαρῶν;

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

Είσαγωγή στους ηλεκτρονικούς υπολογιστές

1.1 Γενικά	1
1.2 Οι πρώτες υπολογιστικές μηχανές	1
1.3 Ηλεκτρονική επεξεργασία δεδομένων – Έξελιξη ηλεκτρονικών υπολογιστών	3
1.4 Κατηγορίες ηλεκτρονικών υπολογιστών	5
1.4.1 Αναλογικοί Υπολογιστές	7
1.4.2 Ψηφιακοί Υπολογιστές	7
1.4.3 Υβριδικοί ή Μικτοί Υπολογιστές	9

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

Αλγόριθμοι – Διαγράμματα ροής

2.1 Τι είναι αλγόριθμος	10
2.2 Διαγράμματα ροής	10
2.2.1 Επίπεδα διαγραμμάτων ροής	10
2.2.2 Σύμβολα διαγράμματος ροής	13
2.3 Ερωτήσεις	21
2.4 Ασκήσεις	21

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

Αριθμητικά συστήματα – Κώδικες

3.1 Αριθμητικά συστήματα	22
3.1.1 Το δεκαδικό σύστημα αρίθμησης	22
3.1.2 Το δυαδικό σύστημα αρίθμησης	23
3.1.3 Το οκταδικό σύστημα αρίθμησης	24
3.1.4 Το δεκαεξαδικό σύστημα αρίθμησης	24
3.1.5 Μετατροπή ενός αριθμού από ένα σύστημα αρίθμησης σε άλλο	25
3.1.6 Αριθμητικές πράξεις στο δυαδικό σύστημα	32
3.2 Κώδικες	37
3.2.1 Τετραψήφιοι κώδικες BCD	37
3.2.2 Κώδικας «υπερβολής» κατά 3*	38
3.2.3 Κώδικας Gray	39
3.2.4 Κώδικας ASCH	39
3.2.5 Ο κώδικας EBCDIC (Extended Binary Coded Decimal Interchange Code)	41
3.2.6 Κώδικες αναγνώρισης σφαλμάτων	41
3.3 Ερωτήσεις	44
3.4 Ασκήσεις	44

ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

Λογικά κυκλώματα

4.1	Ἡ ἀλγεβρα τοῦ Boole	46
4.1.1	Γενικά	46
4.1.2	Ἀξιώματα τῆς ἀλγεβρας τοῦ Boole	46
4.1.3	Διαγράμματα Venn	47
4.1.4	Ἀλγεβρα Λογικῆς - Πίνακες Ἀληθείας	48
4.1.5	Θεωρήματα τῆς Ἀλγεβρας Boole	49
4.2	Πραγματοποίηση Λογικῶν Κυκλωμάτων	52
4.2.1	Θετική καὶ ἀρνητική λογική - Λογικές πύλες	52
4.2.2	Πραγματοποίηση λογικῶν κυκλωμάτων μέ διακεκριμένα στοιχεία	53
4.3	Συνδυαστικά Λογικά Κυκλώματα	56
4.3.1	Κατασκευή τοῦ λογικοῦ κυκλώματος ἀπὸ τὴ συνάρτηση	57
4.3.2	Εὑρεση τῆς λογικῆς συναρτήσεως ἀπὸ τὸ διάγραμμα τοῦ λογικοῦ κυκλώματος	58
4.3.3	Τὰ συνδυαστικά κυκλώματα στὸν Ἡλεκτρονικό Ὑπολογιστὴ	60
4.4	Ἀκολουθιακά κυκλώματα	70
4.4.1	Τύποι τῶν ἠλεκτρονικῶν στοιχείων μνήμης	71
4.5	Ἑρωτήσεις	73
4.6	Ἀσκήσεις	75

ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ

Δομὴ καὶ λειτουργία τῶν ἠλεκτρονικῶν ὑπολογιστῶν

5.1	Γενικά	76
5.2	Ἡ μονάδα μνήμης	77
5.3	Ἀριθμητική καὶ λογική μονάδα (ALU)	79
5.4	Μονάδα ἐλέγχου	81
5.5	Ἡ μονάδα Εἰσόδου - Ἐξόδου (I/O)	82
5.6	Διαδικασία ἐκτελέσεως μιᾶς ἐντολῆς. Ἀρχὴ τῶν δύο φάσεων	83
5.7	Λειτουργία τοῦ ἠλεκτρονικοῦ ὑπολογιστῆ	84
5.8	Ἑρωτήσεις	87

ΚΕΦΑΛΑΙΟ ΕΚΤΟ

Συστήματα ἀποθηκεύσεως πληροφοριῶν

6.1	Γενικά	88
6.2	Κεντρικὴ Μνήμη - Βασικά χαρακτηριστικά	88
6.2.1	Ὁ μαγνητικός δακτύλιος ὡς στοιχεῖο κεντρικῆς μνήμης	90
6.2.2	Οἱ μνῆμες ἡμιαγωγῶν ὡς στοιχεῖα κεντρικῆς μνήμης	92
6.3	Περιφερειακὲς Μονάδες Μνήμης (μαζικὴ μνήμη)	102
6.3.1	Μονάδα μαγνητικῆς ταινίας	102
6.3.2	Μονάδα μαγνητικῶν δίσκων	107
6.3.3	Μονάδα μαγνητικοῦ τυμπάνου	109
6.3.4	Μονάδα εὐκάμπτων μαγνητικῶν δισκετῶν	110
6.3.5	Μονάδα διατρήσεως ἀναγνώσεως δελτίων	112
6.3.6	Μονάδα διατρήσεως ἀναγνώσεως χαρτοταινίας (Paper Tape Reed, Punch)	114
6.4	Ἑρωτήσεις	117

ΚΕΦΑΛΑΙΟ ΕΒΔΟΜΟ

Περιφερειακές μονάδες

7.1 Γενικά	118
7.2 Τηλέτυπα (TTY)	121
7.3 Όθονες (Cathode Ray Tubes - CRT)	122
7.4 Μονάδες σχεδίασεως (Plottere) και Μονάδες ψηφοποίησης (Digitizers)	123
7.5 Έκτυπώτες χαρτιού	125
7.5.1 Έκτυπώτες χαρακτήρων (Character Printers)	126
7.5.2 Έκτυπώτες γραμμών	127
7.5.3 Έκτυπώτες σελίδων	130
7.6 Άλλες περιφερειακές μονάδες	130
7.7 Ερωτήσεις	131

ΚΕΦΑΛΑΙΟ ΟΓΔΩΟ

Γλώσσες προγραμματισμού ανώτερου επιπέδου

8.1 Προγραμματικός εξοπλισμός (Software)	132
8.2 Ιστορική επισκόπηση του Software	132
8.2.1 Πρώτο στάδιο	133
8.2.2 Δεύτερο στάδιο	134
8.2.3 Τρίτο στάδιο	134
8.2.4 Τέταρτο στάδιο	136
8.3 Προγραμματισμός σε γλώσσα Μηχανής	137
8.3.1 Οι εντολές του υπολογιστή και η εκτέλεσή τους γενικά	137
8.3.2 Παραδείγματα προγραμματισμού σε γλώσσα μηχανής	140
8.4 Προγραμματισμός σε Συμβολική Γλώσσα (Assembly Language Programming)	142
8.4.1 Γενικά	142
8.4.2 Πίνακες εντολών σε συμβολική γλώσσα ενός υποθετικού υπολογιστή	143
8.5 Προγραμματισμός σε γλώσσες ανώτερου επιπέδου	147
8.5.1 Προγραμματισμός σε γλώσσα Basic	147
8.5.2 Διάφορα χαρακτηριστικά της γλώσσας Basic	150
8.6 Υποπρογράμματα (Subprograms)	155
8.7 Ερωτήσεις	157
8.8 Άσκήσεις	157

ΚΕΦΑΛΑΙΟ ΕΝΑΤΟ

Τό λειτουργικό σύστημα του ηλεκτρονικού υπολογιστή

9.1 Γενικά	159
9.2 Τό πρόγραμμα Έκκινήσεως (Bootstrap)	159
9.3 Τό πρόγραμμα φορτώσεως	159
9.4 Τα προγράμματα οδηγήσεως (Drivers)	160
9.5 Άλλα μέρη του Λειτουργικού Συστήματος	160
9.6 Ο τρόπος λειτουργίας του υπολογιστή	160
9.7 Κατηγορίες λειτουργικών συστημάτων	161
9.7.1 Μαζική επεξεργασία (Batch processing)	161
9.7.2 Πολυπρογραμματισμός (Multiprogramming)	162
9.7.3 Τηλεεπεξεργασία (Teleprocessing)	162
9.8 Ερωτήσεις	164

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ

Μικρο-υπολογιστές

ΚΕΦΑΛΑΙΟ ΕΝΔΕΚΑΤΟ

Μετατροπή αναλογικών σημάτων σε ψηφιακά

11.1 Γενικά	167
11.2 Μετατροπείς A/D	167
11.2.1 Ή μέθοδος της ολοκληρώσεως	168
11.2.2 Ή μέθοδος της άμεσης συγκρίσεως	168
11.2.3 Ή μέθοδος των διαδοχικών προσεγγίσεων	169
11.3 Μετατροπείς D/A	171
11.3.1 Μετατροπή D/A με χρήση αντιτάσεων διαφόρων βαρών	172
11.3.2 Μετατροπή D/A με χρήση κλιμακωτών κυκλωμάτων	173
11.4 Έρωτήσεις	175

COPYRIGHT ΙΔΡΥΜΑΤΟΣ ΕΥΓΕΝΙΔΟΥ
