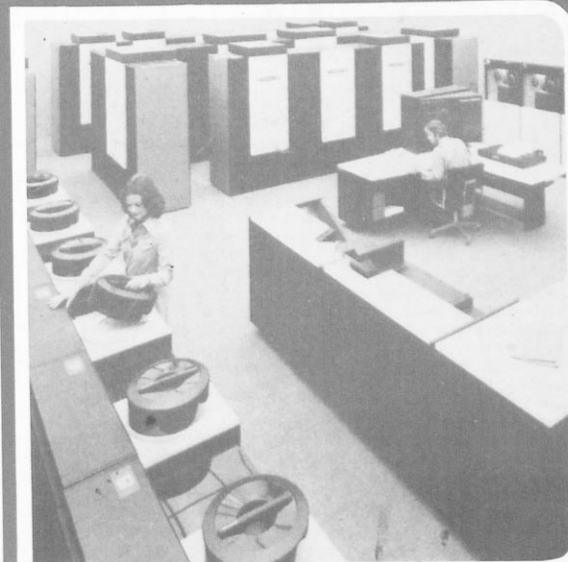


ΗΛΕΚΤΡΟΝΙΚΟΙ ΥΠΟΛΟΓΙΣΤΕΣ

Τηλέμαχου Ι. Καλβουρίδη

ΦΥΣΙΚΟΥ - ΠΡΟΓ/ΤΗ Η/Υ - ΒΟΗΘΟΥ Ε.Μ.Π.





1954

ΙΔΡΥΜΑ ΕΥΓΕΝΙΔΟΥ
ΧΡΥΣΟΥΝ ΜΕΤΑΛΛΙΟΝ ΑΚΑΔΗΜΙΑΣ ΑΘΗΝΩΝ



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ ΔΙΔΑΚΤΙΚΩΝ ΒΙΒΛΙΩΝ (ΙΤΥΕ Δ.Ε.)



Ε 39
Καλβουρίδης, Τηλέμαχος]

Β' ΤΑΞΗ ΕΠΑΓΓΕΛΜΑΤΙΚΟΥ ΛΥΚΕΙΟΥ

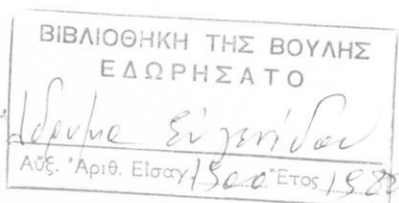
ΗΛΕΚΤΡΟΝΙΚΟΙ ΥΠΟΛΟΓΙΣΤΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΑΡΙΘΜΟΜΗΧΑΝΕΣ

ΤΗΛΕΜΑΧΟΥ ΙΩΑΝ. ΚΑΛΒΟΥΡΙΔΗ
ΦΥΣΙΚΟΥ
ΠΡΟΓΡΑΜΜΑΤΙΣΤΟΥ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΒΟΗΘΟΥ Ε.Μ.Π.

ΑΘΗΝΑ
1979



002
415
ET2B
2130



ΠΡΟΛΟΓΟΣ ΙΔΡΥΜΑΤΟΣ ΕΥΓΕΝΙΔΟΥ

Ὁ Εὐγένιος Εὐγενίδης, ὁ ἱδρυτὴς καὶ χορηγὸς τοῦ «Ἰδρύματος Εὐγενίδου», πολὺ νῦν πρόβλεψε καὶ σχημάτισε τὴν πεποίθησιν ὅτι ἡ ἄρτια κατάρτισις τῶν τεχνικῶν μας, σέ συνδυασμὸν μὲ τὴν ἐθνικὴν ἀγωγὴν, θὰ ἦταν ἀναγκαῖος καὶ ἀποφασιστικὸς παράγοντας τῆς προόδου τοῦ Ἑθνους μας.

Τὴν πεποίθησίν του αὐτὴ ὁ Εὐγενίδης ἐκδήλωσε μὲ τὴ γενναιοφρονα πράξιν εὐεργεσίας, νὰ κληροδοτήσῃ σεβαστὸ ποσὸ γιὰ τὴ σύστασιν Ἰδρύματος πού θὰ εἶχε σκοπὸ νὰ συμβάλλῃ στὴν τεχνικὴ ἐκπαίδευσιν τῶν νέων τῆς Ἑλλάδος.

Ἔτσι τὸ Φεβρουάριον τοῦ 1956 συστήθηκε τὸ «Ἰδρυμα Εὐγενίδου», τοῦ ὁποίου τὴν διοίκησιν ἀνέλαβε ἡ ἀδελφὴ του κυρία Μαριάνθη Σίμου, σύμφωνα μὲ τὴν ἐπιθυμίαν τοῦ διαθέτη.

Ἀπὸ τὸ 1956 μέχρι σήμερον ἡ συμβολὴ τοῦ Ἰδρύματος στὴν τεχνικὴ ἐκπαίδευσιν πραγματοποιεῖται μὲ διάφορες δραστηριότητες. Ὅμως ἀπ' αὐτὲς ἡ σημαντικότερη, πού κρίθηκε ἀπὸ τὴν ἀρχὴν ὡς πρώτης ἀνάγκης, εἶναι ἡ ἐκδόσις βιβλίων γιὰ τοὺς μαθητὰς τῶν τεχνικῶν σχολῶν.

Μέχρι σήμερον ἐκδόθηκαν 150 τόμοι βιβλίων, πού ἔχουν διατεθεῖ σὲ πολλὰ ἑκατομύρια τεύχη, καὶ καλύπτουν ἀνάγκας τῶν Κατώτερων καὶ Μέσων Τεχνικῶν Σχολῶν τοῦ Ὑπ. Παιδείας, τῶν Σχολῶν τοῦ Ὁργανισμοῦ Ἀπασχολήσεως Ἑργατικοῦ Δυναμικοῦ (ΟΑΕΔ) καὶ τῶν Δημοσίων Σχολῶν Ἑμπορικοῦ Ναυτικοῦ.

Μοναδικὴ φροντίδα τοῦ Ἰδρύματος σ' αὐτὴ τὴν ἐκδοτικὴν του προσπάθειαν ἦταν καὶ εἶναι ἡ ποιότης τῶν βιβλίων, ἀπὸ ἀποψη δὲ μόνον ἐπιστημονικὴ, παιδαγωγικὴ καὶ γλωσσικὴ, ἀλλὰ καὶ ἀπὸ ἀποψη ἐμφανίσεως, ὥστε τὸ βιβλίον νὰ ἀγαπηθεῖ ἀπὸ τοὺς νέους.

Γιὰ τὴν ἐπιστημονικὴν καὶ παιδαγωγικὴν ποιότητα τῶν βιβλίων, τὰ κείμενα ὑποβάλλονται σὲ πολλὰς ἐπεξεργασίας καὶ βελτιώνονται πρὶν ἀπὸ κάθε νέα ἐκδόσις.

Ἰδιαίτερη σημασία ἀπέδωσε τὸ Ἰδρυμα ἀπὸ τὴν ἀρχὴν στὴν ποιότητα τῶν βιβλίων ἀπὸ γλωσσικὴ ἀποψη, γιατί πιστεύει ὅτι καὶ τὰ τεχνικὰ βιβλία, ὅταν εἶναι γραμμένα σὲ γλῶσσα ἄρτια καὶ ὁμοιόμορφη ἀλλὰ καὶ κατάλληλη γιὰ τὴν στάθμην τῶν μαθητῶν, μποροῦν νὰ συμβάλλουν στὴν γλωσσικὴ διαπαιδαγώγησιν τῶν μαθητῶν.

Ἔτσι μὲ ἀπόφασιν πού πάρθηκε ἤδη ἀπὸ τὸ 1956 ὅλα τὰ βιβλία τῆς Βιβλιοθήκης τοῦ Τεχνίτη, δηλαδὴ τὰ βιβλία γιὰ τίς Κατώτερες Τεχνικὰς Σχολάς, ὅπως ἀργότερα καὶ γιὰ τίς Σχολάς τοῦ ΟΑΕΔ, εἶναι γραμμένα σὲ γλῶσσα δημοτικὴ μὲ βάση τὴν γραμματικὴν τοῦ Τριανταφυλλίδου, ἐνῶ ὅλα τὰ ἄλλα βιβλία εἶναι γραμμένα στὴν ἀπλὴ καθαρεύουσα. Ἡ γλωσσικὴ ἐπεξεργασία τῶν βιβλίων γίνεται ἀπὸ φιλόλογους τοῦ Ἰδρύματος καὶ ἔτσι ἐξασφαλίζεται ἡ ἐνιαία σύνταξις καὶ ὁρολογία καθὲς κατηγορίας βιβλίων.

Ἡ ποιότητα τοῦ χαρτιοῦ, τὸ εἶδος τῶν τυπογραφικῶν στοιχείων, τὰ σωστά σχήματα καὶ ἡ καλαίσθητη σελιδοποίηση, τὸ ἐξώφυλλο καὶ τὸ μέγεθος τοῦ βιβλίου περιλαμβάνονται καὶ αὐτὰ στίς φροντίδες τοῦ Ἰδρύματος.

Τὸ Ἰδρυμα θεώρησε ὅτι εἶναι ὑποχρέωσή του, σύμφωνα μέ τὸ πνεῦμα τοῦ ἰδρυτῆ του, νά θέσει στήν διάθεση τοῦ Κράτους ὅλη αὐτὴ τὴν πείρα του τῶν 20 ἐτῶν, ἀναλαμβάνοντας τὴν ἐκδόση τῶν βιβλίων καὶ γιὰ τίς νέες Τεχνικές καὶ Ἐπαγγελματικές Σχολές καὶ τὰ νέα Τεχνικά καὶ Ἐπαγγελματικά Λύκεια, σύμφωνα μέ τὰ Ἀναλυτικά Προγράμματα τοῦ Κ.Ε.Μ.Ε.

Τὰ χρονικά περιθώρια γι' αὐτὴ τὴν νέα ἐκδοτικὴ προσπάθεια ἦταν πολὺ περιορισμένα καὶ ἴσως γι' αὐτό, ιδίως τὰ πρῶτα βιβλία αὐτῆς τῆς σειρᾶς, νά παρουσιάσουν ἀτέλειες στὴ συγγραφὴ ἢ στὴν ἐκτύπωση, ποὺ θὰ διορθωθοῦν στὴ νέα τους ἐκδόση. Γι' αὐτό τὸ σκοπὸ ἐπικαλούμαστε τὴν βοήθεια ὧν ὅσων θὰ χρησιμοποιοῦν τὰ βιβλία, ὥστε νά μᾶς γνωστοποιήσουν κάθε παρατήρησή τους γιὰ νά συμβάλλουν καὶ αὐτοὶ στὴ βελτίωση τῶν βιβλίων.

ΕΠΙΤΡΟΠΗ ΕΚΔΟΣΕΩΝ ΙΔΡΥΜΑΤΟΣ ΕΥΓΕΝΙΔΟΥ

Ἀλέξανδρος Ι. Παππᾶς, Ὁμ. Καθηγητῆς ΕΜΠ, Πρόεδρος.

Χρυσόστομος Φ. Καβουνίδης, Διπλ. Μηχ.-Ἡλ. ΕΜΠ, Ἀντιπρόεδρος.

Μιχαὴλ Γ. Ἀγγελόπουλος, Τακτικὸς Καθηγητῆς ΕΜΠ, Διοικητῆς ΔΕΗ.

Παναγιώτης Χατζηιωάννου, Μηχ.-Ἡλ. ΕΜΠ, Γεν. Δ/τῆς Ἐπαγ/κῆς Ἐκπ. Ὑπ. Παιδείας.

Ἐπιστημ. Σύμβουλος, **Γ. Ροῦσσος**, Χημ.-Μηχ. ΕΜΠ.

Σύμβουλος ἐπὶ τῶν ἐκδόσεων τοῦ Ἰδρύματος **Κ.Α. Μανάφης**, Καθηγητῆς Φιλοσοφικῆς Σχολῆς Παν/μίου Ἀθηνῶν.

Γραμματεὺς, **Δ.Π. Μεγαρίτης**.

Διατελέσαντα μέλη ἢ σύμβουλοι τῆς Ἐπιτροπῆς

Γεώργιος Κακριδῆς † (1955 — 1959) Καθηγητῆς ΕΜΠ. **Ἄγγελος Καλογεράς** † (1957 — 1970) Καθηγητῆς ΕΜΠ, **Δημήτριος Νιάνις** (1957 — 1965) Καθηγητῆς ΕΜΠ, **Μιχαὴλ Σπετσιέρης** (1956 — 1959), **Νικόλαος Βασιώτης** (1960 — 1967) Θεόδωρος Κουζέλης (1968 — 1976) Μηχ.-Ἡλ. ΕΜΠ.

ΠΡΟΛΟΓΟΣ

Τό βιβλίο αυτό περιέχει ὅλη τήν ὕλη πού καθορίζει τό ἀναλυτικό πρόγραμμα τοῦ Ὑπουργείου Παιδείας γιά τό μάθημα « Ἀριθμομηχανές - Προγραμματισμός - Ἡλεκτρονικοί ὑπολογιστές ». Τό κείμενο διαιρεῖται σέ 14 κεφάλαια. Τά κεφάλαια 3 καί 4 καθώς καί τά 11, 12 εἶναι περιγραφικά καί δέν χρειάζεται ὁ διδασκόμενος νά ἀποστηθίζει τά ἀριθμητικά μεγέθη πού ἀναφέρονται σέ αὐτά καί πού εἶναι ἀπλῶς ἐνδεικτικά. Ὅσα κεφάλαια ἢ παράγραφοι ἔχουν γραφεῖ μέ μικρά στοιχεῖα καλό θά εἶναι νά παραλειφθοῦν σέ πρώτη φάση καί νά διδαχθοῦν στό τέλος, ἂν ὑπάρχει φυσικά χρόνος. Κρίθηκαν ἀπαραίτητα γιά νά ἔχει ὁ μαθητής μιᾶ ὀλοκληρωμένη γνώση τοῦ γνωστικοῦ ἀντικειμένου πού ζητάει τό ἀναλυτικό πρόγραμμα. Κατά τή διδασκαλία πρέπει νά δοθεῖ ἰδιαίτερη προσοχή καί νά ἀφιερωθεῖ ὅσο γίνεται περισσότερος χρόνος στά κεφάλαια 6, 9 καί 10, πού ἀναφέρονται στό διάγραμμα ροῆς καί στόν προγραμματισμό σέ γλῶσσες BASIC καί FORTRAN.

Στό τέλος τῶν περισσοτέρων κεφαλαίων ὑπάρχουν ὀρισμένες ἀσκήσεις. Ὁ διδάσκων μπορεῖ ἂν θέλει νά προτείνει καί δικές του προσαρμοσμένες ὅσο εἶναι δυνατό στό πνεῦμα τοῦ βιβλίου.

Γνωρίζω πῶς οἱ δυσκολίες πού θά παρουσιασθοῦν στή διδασκαλία ἐνός μαθη-
ματος πού γιά πρώτη φορά θά ἀκούσουν οἱ μαθητές, ὅπως αὐτό, θά εἶναι ἀρκετές. Πιστεύω ὅμως ὅτι μέ τή συνεργασία διδασκόντων καί διδασκόμενων θά ὑπερπη-
δηθοῦν.

Τελειώνοντας θέλω νά ἐκφράσω τίς εὐχαριστίες μου στούς συνεργάτες τοῦ Ἰδρύματος πού μόχθησαν τόσο πολύ γιά τήν κατά τό δυνατόν καλύτερη ἐμφάνιση τοῦ βιβλίου.

Ὁ συγγραφέας

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΑΡΙΘΜΟΜΗΧΑΝΕΣ ΚΑΙ ΣΤΟΥΣ ΨΗΦΙΑΚΟΥΣ ΗΛΕΚΤΡΟΝΙΚΟΥΣ ΥΠΟΛΟΓΙΣΤΕΣ — ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ.

1.1 Ἡ ἀρίθμηση στους ἀρχαίους λαούς.

Ἀπὸ τὴ στιγμή πού δημιουργήθηκαν οἱ πρῶτες ὁργανωμένες ομάδες καὶ ἀρχισαν οἱ ἄνθρωποι νά συναλλάσσονται μεταξύ τους, δημιουργήθηκε καὶ ἡ ἀνάγκη γιὰ τὴν ἐκτέλεση διαφόρων ὑπολογισμῶν.

Στὴν ἀρχὴ οἱ ἄνθρωποι γιὰ τὴ μέτρηση χρησιμοποίησαν τὰ μέλη τοῦ σώματός τους, ὅπως τὰ δάκτυλα τῶν χεριῶν καὶ τῶν ποδιῶν, τοὺς ἀγκῶνες καὶ τοὺς ὠμους τους. Ἀκόμα καὶ σήμερα σέ μερικές φυλές, πού ἐξακολουθοῦν νά ζοῦν μακριὰ ἀπὸ τὸν πολιτισμό, χρησιμοποιεῖται τὸ σύστημα αὐτὸ μετρήσεως.

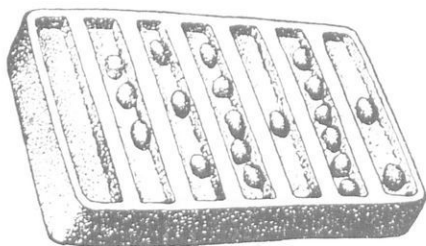
Μὲ τὴν πάροδο τοῦ χρόνου ἡ οἰκονομία στίς ἀνθρώπινες κοινωνίες ἐγίνε πολυπλοκότερη καὶ τὸ μέτρημα μὲ τὰ μέλη τοῦ ἀνθρωπίνου σώματος δέν ἐπαρκοῦσε. Ἀκριβῶς τότε πρωτοεμφανίσθηκαν καὶ οἱ ἀριθμοί-σύνολα, πού ὁ καθένας τους ἀντιπροσώπευε μιά συγκεκριμένη ποσότητα ἀπὸ ἀντικείμενα. Παράλληλα οἱ ἄνθρωποι χρησιμοποίησαν βότσαλα, σπόρους ἀπὸ σιτάρη ἢ καλαμπόκι, κόκκους ἄμμου, καὶ ἄλλα μέσα γιὰ νά λογαριάσουν καὶ νά μετρήσουν.

Ἀργότερα ἐμφανίσθηκαν καὶ τὰ πρῶτα ἀριθμητικὰ σύμβολα. Πολλοὶ λαοὶ ἔχουν νά μᾶς παρουσιάσουν ἀπὸ τὴν ἀρχαία ἱστορία τους διαφορετικὰ συστήματα ἀριθμῆσεως καὶ ἀπεικονίσεως τῶν ἀριθμῶν.

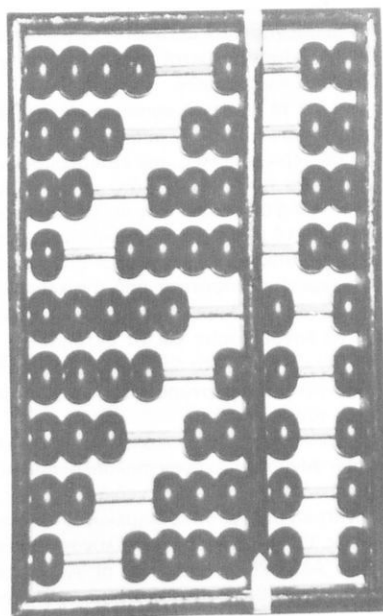
Σιγά-σιγά ἡ ἀνάγκη γιὰ ὑπολογισμούς δυσκολότερους μεγάλωνε. Ἡ ἱστορία μᾶς πληροφορεῖ ὅτι οἱ ἀρχαῖοι Αἰγύπτιοι ἔκαναν πολὺπλοκους λογαριασμούς γιὰ νά βροῦν π.χ. τίς ἰδεώδεις διαστάσεις καὶ γιὰ νά σχεδιάσουν τοὺς βασιλικούς τάφους, καὶ ἀκόμα κρατοῦσαν λογιστικὰ βιβλία, ὅπου σημείωναν καὶ παρακολουθοῦσαν τίς ἐμπορικές συναλλαγές τους. Οἱ ἀρχαῖοι Ἕλληνες ἔλυναν δύσκολα προβλήματα μαθηματικῶν καὶ μηχανικῆς. Ἀλλὰ καὶ οἱ Φοίνικες, οἱ Ἀσσύριοι, οἱ Βαβυλώνιοι, οἱ Πέρσες, οἱ Κινέζοι εἶχαν ἀσχοληθεῖ ἐντατικά μὲ τὰ μαθηματικά.

1.2 Ἡ ἐμφάνιση τῶν πρώτων ὑπολογιστικῶν ὀργάνων.

Λέγεται ὅτι ἡ πιὸ ἀπλὴ ὑπολογιστικὴ συσκευή, ὁ ἄβακας, πού εἶναι ὁ μακρινὸς πρόγονος τοῦ ταπεινοῦ καὶ γνωστοῦ μᾶς ἀριθμηταρίου, ἦταν γνωστός καὶ τὸν χρησιμοποιοῦσαν στὴν ἀρχαία Ἑλλάδα καὶ ἀργότερα στὴν ἀρχαία Ρώμη. Ὁ ἄβακας (σχ. 1.2α) ἦταν μιά πέτρινη πλάκα, πού εἶχε ἐπάνω σκαλισμένα μερικά αὐλάκια. Μέσα σέ κάθε αὐλάκι τοποθετοῦσαν μικρές πέτρες πού κάθε μιά ἀναπλήρωνε κάποιον ἀριθμό.



Σχ. 1.2α.
Ο Αβακας.



Σχ. 1.2β.
Εξελιγμένη μορφή του «Σουάν - Πάν».

Στήν Κίνα από τό 3000 π.Χ. περίπου, εμφανίσθηκε μία λογιστική συσκευή, τό «Σουάν-πάν», πού έμοιαζε πολύ μέ τόν άβακα καί τό χρησιμοποιούσαν σχεδόν μέ τόν ίδιο τρόπο. Μιά άρκετά εξελιγμένη μορφή του «Σουάν-πάν» βλέπομε στό σχήμα 1.2β. Ή παλιά πέτρινη πλάκα άντικαταστάθηκε άπό ένα πλαίσιο πού ένας δια-

χωριστικός κανόνας τό χώριζε σέ δύο τμήματα. Καθένα από τά δύο σφαιρίδια, πού βρίσκονταν στό δεξιό τμήμα, ίσοδυναμούσε μέ 5 σφαιρίδια πού βρίσκονταν στό άριστερό τμήμα. Ένw καθένα από τά 5 σφαιρίδια του άριστερού τμήματος ίσοδυναμούσε μέ μία μονάδα ή δεκάδα κλπ. Ή Γιαπωνέζικη συσκευή «Σορομπάν» διέφερε από τήν Κινέζικη μόνο στό ότι άντί για δύο σφαιρίδια στό δεξιό τμήμα, είχε ένα μόνο.

Στό Μεσαίωνα, ή εκτέλεση μις πράξεως πολλαπλασιασμού εξακολουθούσε να άποτελεϊ έργο ειδίκwν. Στά 1617 ο Σκωτος μαθηματικός *Napier*¹ κατασκεύασε μι ά πρωτόγονη συσκευή, πού μέ τή βοήθειά της μπορούσε να εκτελεϊ ένα πολλαπλασιασμό. Ή συσκευή αύτή εγκαταλείφθηκε πολύ γρήγορα. Λίγο άργότερα, στα 1630 ο *Napier* ανακάλυψε τούς λογαρίθμους και κατόπιν σχεδίασε τόν πρώτο λογαριθμικό κανόνα, πού είναι και τό παληότερο αναλογικό ύπολογιστικό όργανο.

1.3 Οι πρώτες αριθμομηχανές.

Ή επινόηση και ή κατασκευή τών πρώτων αριθμομηχανών συνδέονται μέ τά όνόματα τών *Pascal*² και *Leibniz*³. Ή μηχανή του *Pascal* πού πρωτοεμφανίσθηκε στα 1642 μπορούσε να εκτελεϊ προσθέσεις και αφαιρέσεις. Τήν άποτελούσαν (σχ. 1.3) μι ά σειρά τροχών μέ αριθμούς γύρω από τόν καθένα. Ό πρώτος τροχός συμβόλιζε τίς μονάδες, ο δεύτερος τίς δεκάδες, ο τρίτος τίς εκατοντάδες κ.ο.κ. Όταν ο πρώτος τροχός, πού συμβόλιζε τίς μονάδες, έκανε μι ά πλήρη περιστροφή, ο δεύτερος τροχός πού συμβόλιζε τίς δεκάδες στρεφόταν κατά ένα αριθμό, κλπ.

Ή μηχανή του *Leibniz*, πού εμφανίσθηκε λίγο άργότερα, στα 1673-74, άποτελοΰσε βελτιωμένη μορφή της μηχανής του *Pascal* και μπορούσε να εκτελεϊ όλες τίς άριθμητικές πράξεις.

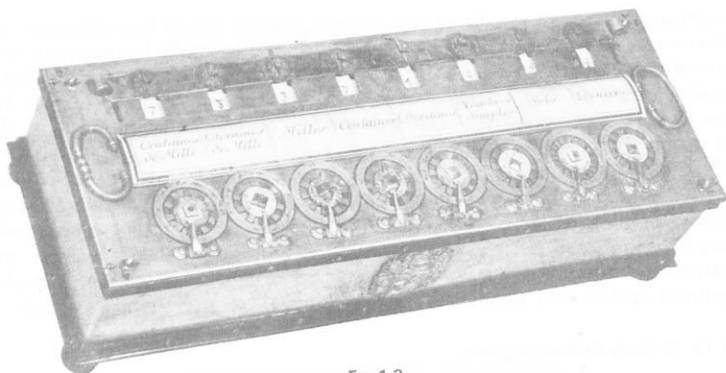
Άπό τήν εποχή αύτή και έπειτα εμφανίζονται πολλές βελτιωμένες διατάξεις, πού όλες όμως στηρίζονται στις άρχικές ιδέες τών *Pascal* και *Leibniz*.

Στά 1872 ο *Frank Baldwin* κατασκεύασε μι ά αριθμομηχανή πού ένα χρόνο άργότερα άρχισε να τήν παράγει μαζίκά. Ή χρονιά εκείνη εσήμανε τήν άρχή της βιομηχανικής παραγωγής τών αριθμομηχανών. Στά 1890 ο *Burroughs* έφοδίασε τίς

1 *John Napier*: Σκωτος μαθηματικός. Γεννήθηκε κοντά στο Έδιμβουργο τό 1550 και πέθανε εκεί τό 1617. Είναι ο πατέρας τών λογαρίθμων και του λογαριθμικού κανόνα. Είναι ο πρώτος πού εισήγαγε τό κλασματικό σημείο.

2 *Blaise Pascal*: Γάλλος μαθηματικός και φυσικός. Γεννήθηκε στο Clermont Ferrand της Γαλλίας τό 1623 και πέθανε στο Παρίσι τό 1662. Σέ ηλικία 16 έτών δημοσίευσε ένα βιβλίο μέ μελέτες του πάνω στη Γεωμετρία τών κωνικών τομών. Ύπηρξε ένας από τούς θεμελιωτές της σύγχρονης θεωρίας τών Πιθανοτήτων. Μελέτησε άκόμη τή συμπεριφορά τών ρευστών και διατύπωσε τή γνωστή άρχή για τή μετάδοση τών πιέσεων στα υγρά. Τό έπιστημονικό του έργο είναι τεράστιο.

3 *Gottfried Wilhelm Leibniz*: Γερμανός φιλόσοφος και μαθηματικός. Γεννήθηκε στη Λειψία τό 1646 και πέθανε στο Άννόβερο τό 1716. Όνομάσθηκε ο «Άριστοτέλης» της εποχής του. Ήταν ο πρώτος πού αντίληφθηκε τήν άξία του δυαδικού συστήματος. Ανακάλυψε τίς όρίζουσες και τό 1693 έδωσε μι ά δική του διατύπωση για τή διατήρηση της μηχανικής ενέργειας. Ανακάλυψε άκόμη και κατασκεύασε τό μεταλλικό βαρόμετρο. Στα 1700 εξέλεγή μαζί μέ τό *Newton* μέλος της Παρισινής Ακαδημίας Έπιστημών. Παρ' όλο τό τεράστιο έπιστημονικό του έργο, πέθανε άγνωσμένος και ξεχασμένος από όλους.



Σχ. 1.3
Ἡ μηχανή τοῦ Pascal.

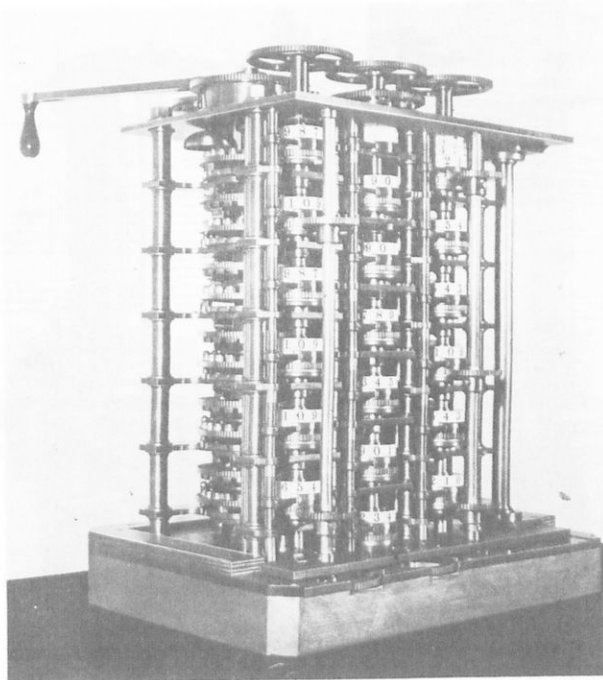
ἀριθμομηχανές μέ πληκτρολόγιο ὁμοιο μέ τό σημερινό καί μέ σύστημα ἐκτυπώσεως τῶν ἀριθμῶν καί ἀποτελεσμάτων.

Ὅλες αὐτές ὅμως οἱ μηχανές, ἐνῶ θά μπορούσαν νά θεωρηθοῦν πρόδρομοι τῶν συγχρόνων ἡλεκτρικῶν καί ἡλεκτρονικῶν ἀριθμομηχανῶν, δέν θεωροῦνται ἔτσι ὅσον ἀφορᾷ στοὺς ψηφιακοὺς ἡλεκτρονικοὺς ὑπολογιστές. Ἡ χρησιμοποίησή τους ἀπαιτοῦσε τή συνεχῆ παρακολούθησή τους ἀπό τό χειριστή καί τό συνεχές ἐφοδιασμό τους μέ στοιχεῖα γιά νά λειτουργήσουν, οἱ δέ ὑπολογισμοί γίνονταν κατὰ τρόπο διακεκομμένο.

1.4 Ἡ ἰδέα τῆς πρώτης αὐτόματης ὑπολογιστικῆς μηχανῆς.

Ἡ ἰδέα τῆς κατασκευῆς μιᾶς αὐτόματης μηχανῆς πού θά μπορούσε ὄχι μόνο νά ἐκτελεῖ διακεκριμένα τίς τέσσερις ἀριθμητικές πράξεις, ἀλλά ἀκόμα καί μιᾶ ὁλόκληρη συνεχῆ σειρά ὑπολογισμῶν, ὀφείλεται στό *Charles Babbage*¹, καθηγητὴ στό Πανεπιστήμιο τοῦ Cambridge. Μετὰ τὴν ἀποφοίτησή του ἀσχολήθηκε μέ μελέτες μαθηματικῶν. Ἦταν ὁμοιο πνεῦμα καί τὴν ἐνεργητικότητα τὴν διοχετεύσε καί σ' ἄλλους ἐπιστημονικοὺς κλάδους. Θεωρεῖται ἓνας ἀπὸ τοὺς θεμελιωτὲς τῆς Ἐπιχειρησιακῆς Ἑρευνας. Στὰ 1847 ἀνακάλυψε τό ὀφθαλμοσκοπιο, ἐνῶ ἀπὸ τό 1812 εἶχε ἀρχίσει νά ἀσχολεῖται μέ τὴ μηχανοποίηση τῶν ὑπολογισμῶν.

¹ *Charles Babbage*: Ἀγγλος μαθηματικός. Γεννήθηκε στό Teignmouth τό 1792 καί πέθανε στό Λονδὶν τό 1871. Ἦταν γιὸς τραπεζίτη. Διδάχθηκε μόνος του μαθηματικά καί τό 1810 εἰσῆλθε στό Πανεπιστήμιο τοῦ Cambridge. Μετὰ τὴν ἀποφοίτησή του ἀσχολήθηκε μέ μελέτες μαθηματικῶν. Ἦταν ὁμοιο ἀνήσυχο πνεῦμα καί τὴν ἐνεργητικότητα τὴν διοχετεύσε καί σ' ἄλλους ἐπιστημονικοὺς κλάδους. Θεωρεῖται ἓνας ἀπὸ τοὺς θεμελιωτὲς τῆς Ἐπιχειρησιακῆς Ἑρευνας. Στὰ 1847 ἀνακάλυψε τό ὀφθαλμοσκοπιο, ἐνῶ ἀπὸ τό 1812 εἶχε ἀρχίσει νά ἀσχολεῖται μέ τὴ μηχανοποίηση τῶν ὑπολογισμῶν.



Σχ. 1.4α.

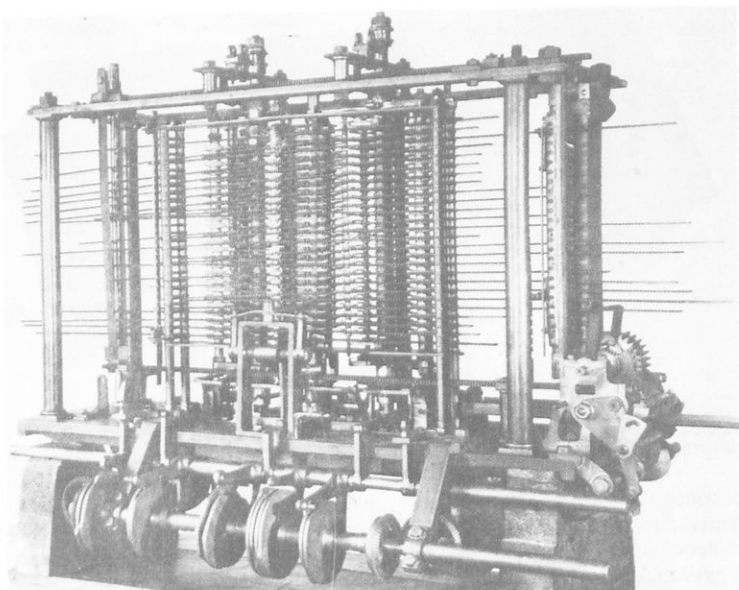
Ἡ διαφορική μηχανή τοῦ Babbage.

τῆς τῆς προσπάθειας ἦταν νά σχεδιάσει μιά νέα καί πιό φιλόδοξη ὑπολογιστική μηχανή, πού τήν ὀνόμασε *ἀναλυτική μηχανή* (σχ. 1.4β).

Τῇ μηχανή αὐτῇ τήν ἀποτελοῦσαν τρία μέρη. Στό πρῶτο, τήν «ἀποθήκη», ἀποταμιεύονταν πολλοί ἀριθμοί συγχρόνως. Στό δεύτερο, τό «μύλο», ἐκτελοῦνταν οἱ ἀριθμητικές πράξεις καί στό τρίτο βρίσκονταν οἱ «μηχανισμοί ἀκολουθιῶν», πού ἐπέλεγαν τοὺς κατάλληλους ἀριθμούς ἀπό τήν «ἀποθήκη» καί τοὺς ὀδηγοῦσαν στό «μύλο» γιά νά ἐκτελεστοῦν στή συνέχεια οἱ προβλεπόμενες ἀριθμητικές πράξεις. Οὔτε ὅμως καί αὐτή ἡ μηχανή ὀλοκληρώθηκε παρ' ὅλο πού μερικά ἀπό τά τμήματά της εἶχαν κατασκευασθεῖ. Ὁ Babbage πέθανε ἀπογοητευμένος καί τά σχέδιά του ξεχάσθηκαν γιά πολλά χρόνια.

1.5 Ἡ ἀνακάλυψη τῆς διάτρητης καρτέλας.

Μιά ἀνακάλυψη πού ἐγινε στά 1804 συνδέθηκε ἀργότερα μέ τή χρησιμοποίηση τῶν ψηφιακῶν ἠλεκτρονικῶν ὑπολογιστῶν. Πρόκειται γιά τήν ἰδέα τοῦ Γάλλου



Σχ. 1.4β.

Ἡ ἀναλυτικὴ μηχανὴ τοῦ Babbage.

μηχανικοῦ *Jacquard*, νὰ χρησιμοποιήσῃ διὰ τρητὲς καρτέλες γιὰ νὰ ἐλέγχει καὶ νὰ καθοδηγεῖ τὰ νήματα στὶς ὑφαντικὲς μηχανές (σχ. 1.5α).

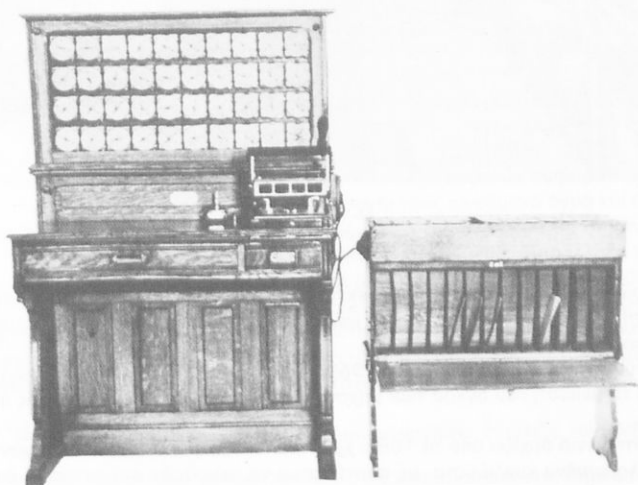
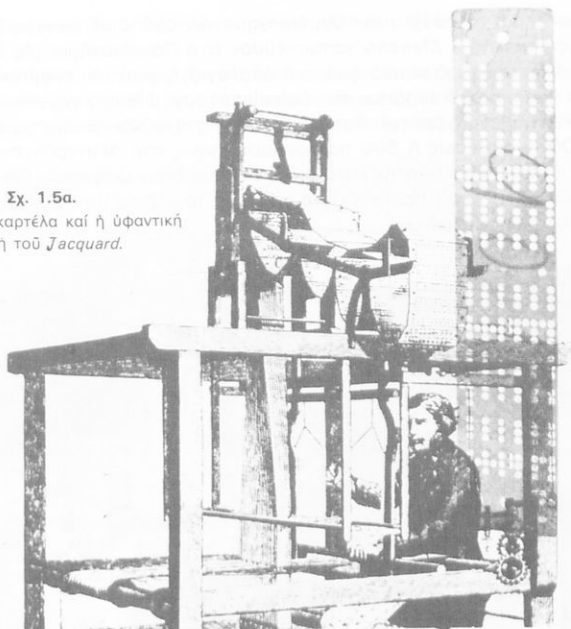
Τὶς διὰ τρητὲς αὐτὲς καρτέλες σκέφθηκε νὰ χρησιμοποιήσῃ ὁ στατιστικολόγος *Herman Hollerith* στὰ 1890, γιὰ νὰ διευκολύνῃ τὸ ἔργο τῆς μεγάλης ἀπογραφῆς τοῦ πληθυσμοῦ τῶν Η.Π.Α. Σχεδίασε γιὰ τὸ σκοπὸ αὐτὸ μιά μηχανή πού τρυποῦσε καρτέλες τὶς ἐπεξεργάζονταν καὶ στὴ συνέχεια τὶς ταξινομοῦσε μὲ τὴ βοήθεια ἡλεκτρομηχανικῶν διατάξεων (σχ. 1.5β).

1.6 Ὁ πρῶτος αὐτόματος ὑπολογιστής.

Ὁ πρῶτος αὐτόματος ψηφιακὸς ὑπολογιστής κατασκευάσθηκε τὸ 1944 στὰ ἐργαστήρια τοῦ Πανεπιστημίου τοῦ Harvard ἀπὸ μιά ομάδα ἐπιστημόνων μὲ ἐπικεφαλὴ τὸν καθηγητὴ *Howard Aiken*. Ὀνομάσθηκε A.S.C.C (Automatic Sequence Controlled Calculator) καὶ ἡ κατασκευὴ του βασίσθηκε σὲ ἡλεκτρομηχανικὲς ἀρχές καὶ εἰδικότερα στὴν τεχνικὴ τῶν ἡλεκτρομηχανικῶν διακοπῶν, πού χρησιμοποιοῦνται στὶς τηλεφωνικὲς συνδέσεις (Relay Techniques). Μποροῦσε νὰ προσθέτῃ δύο 23ψήφιους δεκαδικούς ἀριθμούς σὲ χρόνῳ $1/3$ s ἢ νὰ τοὺς πολλαπλασιάζῃ σὲ χρόνῳ 6 s. Διὰ τὴν κατασκευὴν τοῦ χρειάσθηκαν 750.000 ἐξαρτήματα καὶ χρησιμοποιήθηκε γιὰ τὶς διάφορες συνδέσεις καλὼδιο μήκους 500 μιλίων.

Σχ. 1.5α.

Ἡ διάτρητη καρτέλα καὶ ἡ ὑφαντική
μηχανή τοῦ *Jacquard*.

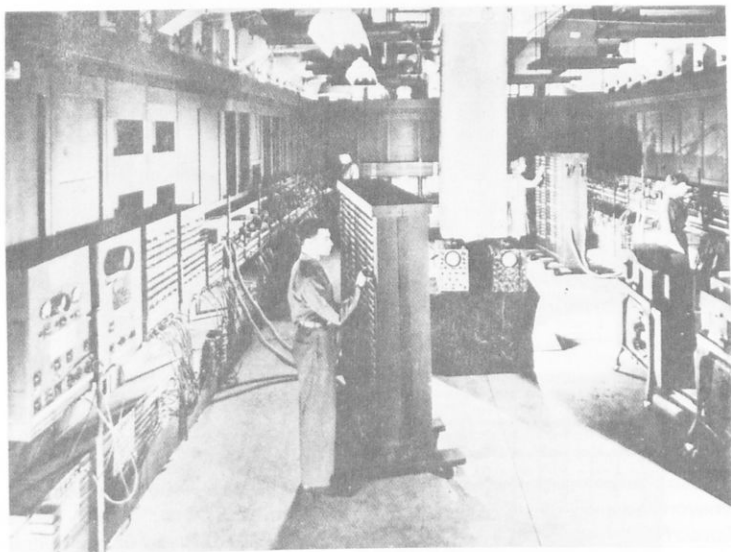


Σχ. 1.5β.

Ἡ μηχανή τοῦ *Hollerith*.

Λίγο αργότερα, τό 1946, μιά άλλη έπιστημονική ομάδα μέ έπικεφαλής τόν καθηγητή τής Φυσικής *J. Manchy* κατασκεύασε στό Πανεπιστήμιο τής Πενσυλβανίας ένα αυτόματο ηλεκτρονικό ψηφιακό υπολογιστή, πού τόν όνόμασε E.N.I.A.C (Electronic Numerical Integrator and Calculator) (σχ. 1.6α). Συνδυασμός 18.000 ηλεκτρονικών λυχνιών αποτελούσε τόν υπολογιστή αυτόν, πού μπορούσε νά έκτελεί 5000 προσθέσεις ή 500 πολλαπλασιασμούς στό δευτερόλεπτο.

Από τό 1950 ή εξέλιξη στην κατασκευή βελτιωμένων ψηφιακών ηλεκτρονικών υπολογιστών είναι χωρίς προηγούμενο καί σέ αυτό βέβαια συντελεί καί ή μεγάλη πρόοδος τής ηλεκτρονικής.



Σχ. 1.6α.
Ο υπολογιστής E.N.I.A.C.

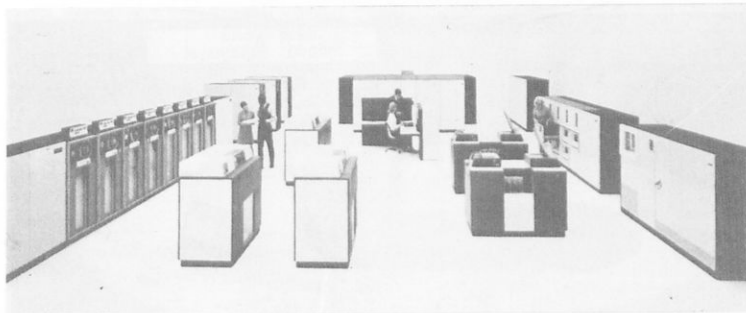
Η πρώτη γενιά τών ψηφιακών ηλεκτρονικών υπολογιστών χαρακτηρίζεται από τή χρήση ηλεκτρονικών λυχνιών, τό μεγάλο όγκο τής κατασκευής καί τή σχετικά μικρή ταχύτητα υπολογισμών.

Η δεύτερη γενιά, πού άρχίζει τό 1959, χρησιμοποιεί ήμιαγωγούς, έπιτυγχάνοντας έτσι τή μείωση του όγκου τών μηχανών καί τήν αύξηση τής ταχύτητας υπολογισμών.

Η τρίτη γενιά άρχίζει από τό 1965, χρησιμοποιεί μικροκυκλώματα καί μονολιθικά ολοκληρωμένα κυκλώματα, μέ αποτέλεσμα νά μειωθούν σημαντικά οι διαστάσεις τους καί νά αύξηθεί ή ταχύτητα υπολογισμών, πού φθάνει στην τάξη μεγέθους νανοδευτερολέπτου (1 δισεκατομμυριοστού του δευτερολέπτου). Εισάγεται

έπίσης η τεχνική του πολυπρογραμματισμού και αυξάνονται οι δυνατότητες των περιφερειακών μονάδων.

Από τό 1970 διανύομε την τέταρτη περίοδο όπου όχι μόνο η τεχνολογία των υπολογιστών έχει βελτιωθεί ακόμα περισσότερο, αλλά έχουν εισαχθεί και νέες ί-σχυρότατες και πιο ευέλικτες γλώσσες προγραμματισμού (σχ. 1.6β).



Σχ. 1.6β.

Ένα σύγχρονο συγκρότημα ψηφιακού ηλεκτρονικού υπολογιστού 4ης γενιάς.

1.7 Ρίχνοντας μία ματιά στα τελευταία επίτεύγματα της επιστήμης.

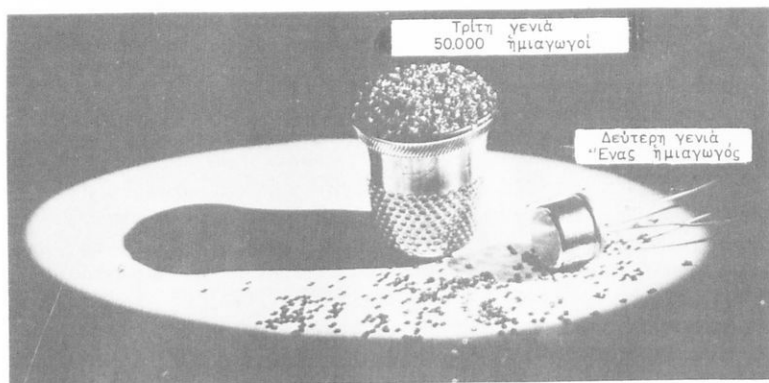
Η δεκαετία του 1970 όχι μόνο χαρακτηρίζεται από την τέταρτη γενιά υπολογιστών, αλλά διακρίνεται και από άλλα καταπληκτικά επίτεύγματα της επιστήμης, όπως η κατασκευή μικροσκοπικών προγραμματιζόμενων υπολογιστικών μηχανών, η μετάδοση δεδομένων και η ανταλλαγή πληροφοριών με τη βοήθεια δορυφόρων ανάμεσα σε δύο συγκροτήματα ηλεκτρονικών υπολογιστών, που βρίσκονται σε διαμετρικά αντίθετα σημεία του πλανήτη μας και η εφαρμογή των άρχων της κρυογενικής (cryogenics), που ελαττώνει την ευαισθησία των υπολογιστών στη θερμοότητα, διατηρώντας τη θερμοκρασία τους κοντά στο απόλυτο μηδέν.

Στή δεκαετία που διανύομε άρχισαν να κατασκευάζονται πειραματικά υπολογιστές, που θα χρησιμοποιούν για τη λειτουργία τους κινούμενα υγρά και υγροδυναμικές βαλβίδες, αντί για ηλεκτρονικά κυκλώματα, γεγονός που θα σήμανει σημαντική ελάττωση του κόστους λειτουργίας και συντηρήσεως. Παράλληλα πειραματίζονται και με υπολογιστές που θα λειτουργούν με laser και θα μπορούν να εκτελούν 1 τρισεκατομμύριο πράξεις σε ένα δευτερόλεπτο. Τέλος οί' επιστήμονες μελετούν τη χρησιμοποίηση νέων ηλεκτροοπτικών μνημών, που θα αποτελούνται από εξαιρετικά λεπτά στρώματα σιδηροηλεκτρικών υλικών και θα μπορούν να μεταβάλλουν την κατάσταση τους με την ταχύτητα του φωτός.

1.8 Ό ηλεκτρονικός υπολογιστής θά υποκαταστήσει έντελως τον άνθρωπο;

Παρ' όλο που οι ηλεκτρονικοί υπολογιστές χρησιμοποιούνται σήμερα και εξυπηρετούν όλους τους τομείς της ανθρώπινης δραστηριότητας, η απάντηση στο έρωτημα είναι ξεκάθαρα αρνητική.

“Ας μὴ ξεχνᾶμε ὅτι ὁ ἠλεκτρονικός ὑπολογιστής, ποὺ εἶναι προϊόν τῆς ἀνθρώπινης ἐξυπνάδας καὶ ἰκανότητας, δὲν εἶναι παρὰ μιά ἀψυχη μηχανὴ χωρὶς δική της ἐξελιγμένη λογική, ἡ ὁποία ὁμως διαθέτει δύο μεγάλες δυνατότητες: τὴν ἐκτέλεση μεγάλου ὄγκου ὑπολογισμῶν μὲ τρομακτικὴ ταχύτητα καὶ τὴν ἀποθήκευση μεγάλων ποσοτήτων στοιχείων, δεδομένων καὶ πληροφοριῶν.



Σχ. 1.8.

Στὴ φωτογραφία δεξιὰ φαίνεται ὁ ἡμιαγωγὸς ποὺ χρησιμοποιήθηκε στοὺς ὑπολογιστές τῆς δευτέρης γενιᾶς. Ἀριστερὰ μέσα στὴ δαχτυλῆθρα ὑπάρχουν 50.000 ἡμιαγωγοὶ νεώτερης τεχνολογίας ποὺ χρησιμοποιήθηκαν στοὺς ὑπολογιστές τῆς τρίτης γενιᾶς.

1.9 Κατηγορίες ὑπολογιστικῶν μηχανῶν.

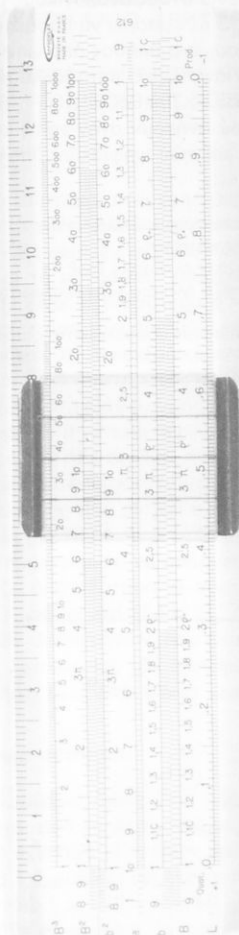
Τίς ὑπολογιστικές μηχανές τίς κατατάσσουμε σέ τρεῖς μεγάλες κατηγορίες:

- α) Ψηφιακές (Digital).
- β) Ἀναλογικές (Analog).
- γ) Ὑβριδικές (Hybrid).

Στὶς ἀναλογικές, οἱ ἀριθμητικὲς ποσότητες παριστάνονται ἀναλογικά μὲ κάποιο ἄλλο φυσικὸ μέγεθος ὅπως τὸ μήκος, ἡ γωνία στροφῆς, τὸ ἠλεκτρικὸ δυναμικὸ, ἡ μαγνητικὴ ροή κλπ. Ἡ πιὸ γνωστὴ ἀναλογικὴ ὑπολογιστικὴ συσκευὴ εἶναι ὁ λογαριθμικὸς κανόνας (Slide Rule, σχῆμα 1.9) ὅπου οἱ λογάριθμοι τῶν πραγματικῶν ἀριθμῶν παριστάνονται μὲ ἀνάλογες ἀποστάσεις ἀπὸ κάποια ἀρχή, πάνω στὸν κανόνα. Ἐτσι, ἂν θέλομε νά πολλαπλασιάσουμε δύο πραγματικοὺς ἀριθμοὺς, ἐπειδὴ ὅπως ξέρομε τὸ ἄθροισμα τῶν λογαρίθμων δύο ἀριθμῶν ἴσοῦται μὲ τὸ λογάριθμο τοῦ γινομένου τους¹, ὁ πολλαπλασιασμὸς γίνεται στὸν κανόνα μὲ ἀπλὴ πρόσθεση τῶν μηκῶν τῶν λογαρίθμων τους καὶ στὴ συνέχεια ἀντιλογαρίθμηση.

Οἱ σύγχρονοι ἀναλογικοὶ ὑπολογιστές κατασκευάζονται μὲ ἠλεκτρονικά κυκλώματα καὶ ἔχουν ἀρκετὲς δυνατότητες, χρησιμοποιοῦν δέ σὰν ἀναλογικὰ μεταβαλλόμενο μέγεθος τὸ ἠλεκτρικὸ δυναμικὸ.

¹ $\log(A \cdot B) = \log A + \log B$



Σχ. 1.9.
Ο λογαριθμικός κανόνας.

Ἡ πῶ παλιά ἀπό τίς ψηφιακές ὑπολογιστικές συσκευές εἶναι ὁ ἄβακας (σχῆμα 1.2α) καί ἡ πῶ ἐξελιγμένη, ὁ ψηφιακός ἠλεκτρονικός ὑπολογιστής (Digital Electronic Computer).

Ἡ πῶ βασική διαφορά ἀνάμεσα στίς δύο κατηγορίες πού ἀναφέραμε, εἶναι ὅτι στούς ψηφιακοὺς ὑπολογιστές οἱ ἀριθμητικές ποσότητες μποροῦν νά μεταβάλλονται μόνο κατὰ τρόπο ἀσυνεχή, ἐνῶ στούς ἀναλογικοὺς μεταβάλλονται κατὰ τρόπο συνεχή. Ἔτσι, ἐνῶ ὁ ἀριθμός $1/3$ δέν μπορεῖ νά παρασταθεῖ σέ ἓνα ψηφιακό ὑπολογιστή ἀκριβῶς μέ ἓνα πεπερασμένο πλῆθος ἀπό ψηφία, σέ ἓνα ἀναλογικό ὑπολογιστή πού χρησιμοποιεῖ σάν ἀναλογικά μεταβαλλόμενο μέγεθος τό μήκος, τό $1/3$ τῆς μονάδας μήκους μπορεῖ νά βρεθεῖ.

Οἱ ὑβριδικοί ὑπολογιστές πού ἀνήκουν στήν τρίτη κατηγορία συνδυάζουν χαρακτηριστικά τῶν ἀναλογικῶν καί τῶν ψηφιακῶν ὑπολογιστῶν.

Ἑρωτήσεις.

1. Ποιός θεωρεῖται ὁ πατέρας τῶν συγχρόνων ὑπολογιστῶν;
2. Ποιά εἶναι ἡ ἱστορία τῆς διάτρητης καρτέλας;
3. Τί μέσα καταγραφῆς χρησιμοποιοῦσαν οἱ ἀρχαῖοι λαοί;
4. Ποιές οἱ διάφορες γενεές τῶν ψηφιακῶν ἠλεκτρονικῶν ὑπολογιστῶν. Ἀπό τί χαρακτηρίζεται ἡ κάθε μία.
5. Σέ πόσες καί ποιές κατηγορίες κατατάσσομε τίς ὑπολογιστικές μηχανές. Ἀναφέρατε παραδείγματα γιά τίς δύο κυριότερες κατηγορίες.

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

ΑΡΙΘΜΗΤΙΚΑ ΣΥΣΤΗΜΑΤΑ — Η ΑΡΙΘΜΗΤΙΚΗ ΤΩΝ ΨΗΦΙΑΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

2.1 Γενικά.

Τό δεκαδικό αριθμητικό σύστημα, πού χρησιμοποιούμε στίς καθημερινές μας συναλλαγές, προέρχεται από τούς Άραβες καί βασίσθηκε στή χρήση τῶν δέκα δακτύλων τῶν χειρῶν. Στήν Εὐρώπη ἔγινε γνωστό γύρω στά 1200 μ.Χ. Τό σύστημα ὅμως αὐτό δέν εἶναι καί τό μοναδικό. Ὑπάρχουν, ὅπως θά δοῦμε, καί ἄλλα αριθμητικά συστήματα, πού εἶναι περισσότερο ἢ λιγότερο κατάλληλα, γιά νά τά χρησιμοποιήσουμε στήν παράσταση τῶν ἀριθμῶν σέ ἓνα ἠλεκτρονικό ὑπολογιστή, ὅπως τό δυαδικό, τό ὀκταδικό, τό δεκαεξαδικό κλπ.

Ἡ ὀνομασία κάθε αριθμητικοῦ συστήματος προέρχεται ἀπό τό πλῆθος τῶν διαφορετικῶν ψηφίων, πού ἔχομε στή διάθεσή μας, γιά νά παραστήσουμε τίς ἀριθμητικές ποσότητες. Ἐτσι τό δυαδικό σύστημα περιλαμβάνει μόνο δύο ψηφία, τό ὀκταδικό 8, τό δεκαεξαδικό 16 κλπ. Κάθε ἀριθμό, πού ἐκφράζεται σέ κάποιο ἀριθμητικό σύστημα, τόν γράφουμε μέσα σέ παρενθέσεις. Ἐξω ἀπό αὐτή καί δεξιά κάτω σημειώνουμε ἀριθμητικά τό σύστημα πού ἀνήκει π.χ. ὁ ἀριθμός $(124)_8$ ἀνήκει στό ὀκταδικό σύστημα, ὁ ἀριθμός $(521)_{16}$ ἀνήκει στό δεκαεξαδικό, ὁ $(143)_{10}$ στό δεκαδικό κ.ο.κ. Τό ἴδιο ψηφίο βέβαια μπορεῖ νά ὑπάρχει σέ πολλά διαφορετικά συστήματα ἀλλά ἡ ἀξία του ἀλλάζει ἀπό σύστημα σέ σύστημα, ἔτσι π.χ. στόν μὲν ἀριθμό $(12)_{10}$ τό 1 ἔχει ἀξία μιᾶς δεκάδας στόν δέ $(12)_8$, ἔχει ἀξία μιᾶς ὀκτάδας κ.ο.κ. Ἡ περιγραφή τῶν ἀριθμητικῶν συστημάτων, πού ἀκολουθεῖ, εἶναι βασική γιά τήν ἐξήγηση καί κατανόηση τῆς λειτουργίας τῶν ἠλεκτρονικῶν ὑπολογιστῶν. Ἀναλυτικότερα θά ἐξηγηθεῖ ἡ χρήση τους στά εἰδικά κεφάλαια.

2.2 Τό δεκαδικό ἀριθμητικό σύστημα.

Τό σύστημα αὐτό περιλαμβάνει, ὅπως εἶναι γνωστό, δέκα ψηφία 0, 1, 2, ..., 9. Ἡ ἀξία κάθε ψηφίου ἐξαρτᾶται ἀπό τή θέση πού κατέχει μέσα σέ ἓνα ἀριθμό. Π.χ. στόν ἀριθμό $(124)_{10}$, τό 4 ἔχει ἀξία τεσσάρων μονάδων, τό 2 ἔχει ἀξία δύο δεκάδων ἢ 20 μονάδων, καί τό 1 ἔχει ἀξία μιᾶς ἑκατοντάδας. Δηλαδή ὅπως ἔχω γράψει τόν ἀριθμό εἶναι σά νά ἔννοω $1 \times 100 + 2 \times 10 + 4 \times 1$ ἢ $1 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$. Ὅπως βλέπομε, κάθε ἀριθμό τοῦ δεκαδικοῦ συστήματος μποροῦμε νά τόν παραστήσουμε σάν ἓνα ἄθροισμα μέ δυνάμεις τοῦ 10, ὅπου τά ψηφία τοῦ ἀριθμοῦ ἐμφανίζονται σάν συντελεστές. Ἄν ἀντίστροφα ἔχω τό ἄθροισμα $2 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 0 \times 10^0$, αὐτό δέν εἶναι ἄλλο ἀπό τόν ἀριθμό $(2350)_{10}$. Τό 10 ὀνομάζεται **βάση** τοῦ δεκαδικοῦ συστήματος.

Πώς όμως θα παραστήσουμε έναν κλασματικό¹ αριθμό π.χ. τόν $(210.12)_{10}$; 'Απλούστατα θα χρησιμοποιήσουμε γιά τήν παράσταση τών κλασματικῶν ψηφίων τοῦ αριθμοῦ, ἀρνητικές δυνάμεις τοῦ 10. Ἔτσι μπορῶ νά τόν γράψω: $(210.12)_{10} = 2 \times 10^2 + 1 \times 10^1 + 0 \times 10^0 + 1 \times 10^{-1} + 2 \times 10^{-2}$.

2.3 Τό δυαδικό ἀριθμητικό σύστημα.

Στό σύστημα αὐτό βάση εἶναι τό 2 καί ὑπάρχουν μόνο δύο ψηφία τό 0 καί τό 1. Ὅλοι οἱ ἀριθμοί ἐκφράζονται ὡς συνδυασμοί τοῦ 0 καί τοῦ 1. Ἔτσι, ὁ ἀκέραιος δυαδικός ἀριθμός $(11001)_2$ ἰσοδυναμεῖ μέ τόν $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = 25$ τοῦ δεκαδικοῦ συστήματος. Ἐπίσης ὁ μικτός δυαδικός ἀριθμός 1101.11 ἰσοδυναμεῖ μέ τόν $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times \frac{1}{2} + 1 \times \frac{1}{4} = (13.75)_{10}$ τοῦ δεκαδικοῦ συστήματος.

2.4 Τό ὀκταδικό ἀριθμητικό σύστημα.

Τό σύστημα αὐτό περιλαμβάνει 8 ψηφία δηλ. 0, 1, ..., 7, καί ἔχει βάση τό 8. Ἔτσι ὁ ἀκέραιος ὀκταδικός ἀριθμός $(3470)_8$ ἰσοδυναμεῖ μέ τόν $3 \times 8^3 + 4 \times 8^2 + 7 \times 8^1 + 0 \times 8^0 = 1536 + 256 + 56 + 0 = (1848)_{10}$.

Ἐπίσης ὁ μικτός ὀκταδικός ἀριθμός 123.14 ἰσοδυναμεῖ μέ τόν $1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 + 1 \times 8^{-1} + 4 \times 8^{-2} = 16 + 16 + 3 + \frac{1}{8} + \frac{1}{64} = (35.1406)_{10}$.

Κάθε ψηφίο τοῦ ὀκταδικοῦ συστήματος ἐκφράζεται μέ τρία ψηφία τοῦ δυαδικοῦ. Δηλαδή ὁ ἀριθμός $(3470)_8$ ἀπεικονίζεται μέ 12 δυαδικά ψηφία.

2.5 Τό δεκαεξαδικό ἀριθμητικό σύστημα.

Τό δεκαεξαδικό σύστημα ἔχει ὡς βάση τό 16 καί ἀποτελεῖται ἀπό 16 ψηφία 0, 1, ..., 9, A, B, C, D, E, F. Ὅπως βλέπομε, ἐπειδή τά 10 ψηφιακά σύμβολα, πού γνωρίζομε ἀπό τό δεκαδικό σύστημα δέν φθάνουν, χρησιμοποιοῦμε καί τά 6 πρῶτα γράμματα τοῦ Λατινικοῦ Ἀλφαβήτου. Δηλαδή τό A συμβολίζει τό 10, τό B συμβολίζει τό 11 κ.ο.κ. Ἔτσι, ὁ ἀκέραιος δεκαεξαδικός 4CD σημαίνει: $4 \times 16^2 + C \times 16^1 + D \times 16^0 = 4 \times 16^2 + 12 \times 16^1 + 13 \times 16^0 = 1024 + 192 + 13 = (1229)_{10}$.

Κάθε ψηφίο τοῦ δεκαεξαδικοῦ συστήματος ἀπεικονίζεται μέ τέσσερα ψηφία τοῦ δυαδικοῦ συστήματος. Δηλαδή ὁ ἀριθμός $(4CD)_{16}$ θά ἐκφράζεται στό δυαδικό σύστημα μέ 12 ψηφία.

¹ Κλασματικό ἀριθμό θά ὀνομάζομε ὁ,τι μέχρι σήμερα ἀπό κακή συνήθεια ὀνομάζομε «δεκαδικό». Ὁ δεκαδικός ἀριθμός εἶναι ἕνας ἀριθμός πού ἐκφράζεται στό δεκαδικό ἀριθμητικό σύστημα καί ὄχι ἕνας ἀριθμός πού περιλαμβάνει κλάσματα τῆς ἀκέραιας μονάδας. Μικτός ἀριθμός εἶναι ἕνας ἀριθμός πού περιλαμβάνει καί ἀκέραιο καί κλασματικό μέρος. Τό σημείο τῆς ὑποδιαστολῆς θά τό παριστάνομε μέ τελεία, γιατί ἔτσι παριστάνεται στοὺς υπολογιστές.

2.6 Μετατροπή αριθμών από ένα σύστημα σε κάποιο άλλο με διαφορετική βάση.

α) Γενικά.

Επειδή τό δυαδικό σύστημα είναι τό απλούστερο πού υπάρχει μιά καί αποτελείται μόνο από δύο ψηφία 0 καί 1, είναι τό καταλληλότερο γιά νά εξυπηρετήσει τήν αριθμητική τών ψηφιακών υπολογιστών. Σημειώνουμε ότι είναι πιά εύκολο νά βρεῖ κανείς φυσικά συστήματα, πού νά παίρνουν μόνο δύο διακεκριμένες τιμές, ὅπως διακόπτης ἀνοικτός ἢ κλειστός, μεταλλικός δακτύλιος μαγνητισμένος ἢ ἀμαγνήτιστος, λάμπα ἀναμμένη ἢ σβηστή, ἀγωγός ἤλεκτροφόρος ἢ οὐδέτερος κλπ.

Τό ὀκταδικό καί τό δεκαεξαδικό σύστημα χρησιμοποιοῦνται γιά λόγους εὐκολίας ἔξω ἀπό τόν υπολογιστή. Εἶναι λοιπόν χρήσιμο νά γνωρίζουμε πῶς ἕνας δεκαδικός, ὀκταδικός ἢ δεκαεξαδικός ἀριθμός μετατρέπεται σέ δυαδικό καί ἀντίστροφα.

β) Μετατροπή ἀκέραιου δεκαδικοῦ ἀριθμοῦ σέ δυαδικό.

Γιά νά μετατρέψουμε ἕνα ἀκέραιο δεκαδικό ἀριθμό σέ δυαδικό, τόν διαιροῦμε μέ τή βάση τοῦ δυαδικοῦ συστήματος, δηλαδή τό 2. Τό ὑπόλοιπο τῆς διαιρέσεως θά εἶναι τό 0 ἢ τό 1, ἐνῶ τό ἀκέραιο πηλίκο θά εἶναι κάποιος ἄλλος δεκαδικός. Διαιροῦμε τό πηλίκο πάλι μέ τό 2. Τό ὑπόλοιπο τῆς νέας διαιρέσεως θά εἶναι 0 ἢ 1 καί τό πηλίκο κάποιος νέος δεκαδικός ἀριθμός. Συνεχίζουμε μέ αὐτόν, τόν τρόπο τίς διαιρέσεις τών νέων πηλίκων, ἕως ὅτου αὐτό μηδενισθεῖ.

Ἄν π.χ. θέλομε νά μετατρέψουμε τόν ἀκέραιο δεκαδικό ἀριθμό $(45)_{10}$ σέ δυαδικό, θά προβοῦμε στή σειρά στίς ἐξῆς πράξεις:

$$45 : 2 = 22 \quad \text{καί} \quad \text{ὑπόλοιπο } 1$$

$$22 : 2 = 11 \quad \text{καί} \quad \text{ὑπόλοιπο } 0$$

$$11 : 2 = 5 \quad \text{καί} \quad \text{ὑπόλοιπο } 1$$

$$5 : 2 = 2 \quad \text{καί} \quad \text{ὑπόλοιπο } 1$$

$$2 : 2 = 1 \quad \text{καί} \quad \text{ὑπόλοιπο } 0$$

$$1 : 2 = 0 \quad \text{καί} \quad \text{ὑπόλοιπο } 1$$

Ἄν τώρα τοποθετήσουμε τά ὑπόλοιπα πού βρήκαμε, στίς διαδοχικές διαιρέσεις σέ ἀντίστροφη σειρά, θά ἔχομε τό ζητούμενο ἀριθμό. Δηλαδή:

$$(45)_{10} = (101101)_2$$

γ) Μετατροπή κλασματικοῦ δεκαδικοῦ ἀριθμοῦ σέ δυαδικό.

Γιά νά κάνουμε αὐτή τή μετατροπή ἐργαζόμαστε ὡς ἐξῆς: Πολλαπλασιάζουμε τόν κλασματικό μέ τή βάση 2. Ἄν τό γινόμενο εἶναι μεγαλύτερο ἀπό τή μονάδα τήν ἀφαιροῦμε καί τό ἀποτέλεσμα τό πολλαπλασιάζουμε πάλι μέ τό 2. Συνεχίζουμε μ' αὐτόν τόν τρόπο μέχρις ὅτου πετύχομε μιά ἐπιθυμητή ἀκρίβεια. (Δηλαδή στήν περίπτωση αὐτή ἡ διαδικασία τῆς μετατροπῆς δέν τελειώνει μόνη της, ἀλλά συνεχίζεται ὅσο θέλομε). Γιά τόν κλασματικό $(0.379)_{10}$ θά ἔχομε:

$$0.379 \times 2 = 0.758 \quad \text{καί} \quad \text{ἀκέραιο μέρος } 0$$

$$0.758 \times 2 = 0.516 \quad \text{καί} \quad \text{ἀκέραιο μέρος } 1$$

$$0.516 \times 2 = 0.032 \quad \text{καί} \quad \text{ἀκέραιο μέρος } 1$$

$$0.032 \times 2 = 0.064 \quad \text{καί} \quad \text{ἀκέραιο μέρος } 0$$

$$0.064 \times 2 = 0.128 \quad \text{καί} \quad \text{ἀκέραιο μέρος } 0$$

$$\begin{aligned} 0.128 \times 2 &= 0.256 \quad \text{και} \quad \text{άκεραιο} \quad \text{μέρος} \quad 0 \\ 0.256 \times 2 &= 0.512 \quad \text{και} \quad \text{άκεραιο} \quad \text{μέρος} \quad 0 \\ 0.512 \times 2 &= 0.024 \quad \text{και} \quad \text{άκεραιο} \quad \text{μέρος} \quad 1 \text{ κλπ.} \end{aligned}$$

Δηλαδή $(0.379)_{10} = (0.011000001...)_{2}$

"Όταν ο δεκαδικός αριθμός έχει και άκεραιο και κλασματικό μέρος τότε το μέν άκεραιο μέρος τό μετατρέπομε μέ τόν τρόπο πού είδαμε στήν παράγραφο 2.6(β) τό δέ κλασματικό μέ τόν τρόπο πού εξηγήσαμε τώρα.

δ) Μετατροπή άκεραιού δυαδικού αριθμού σέ δεκαδικό.

Γιά νά μετατρέψομε ένα άκεραιο δυαδικό αριθμό σέ άκεραιο δεκαδικό, πολλαπλασιάζομε τό πρώτο άπό τά άριστερά ψηφίο του δυαδικού μέ τό 2 και στό γινόμενο προσθέτομε τό επόμενο στή σειρά ψηφίο του δυαδικού. Πολλαπλασιάζομε κατόπιν τό άθροισμα πού βρήκαμε πάλι μέ τό 2 και στό νέο γινόμενο προσθέτομε τό επόμενο ψηφίο κ.ο.κ. Δηλαδή, γιά νά μετατρέψομε τόν δυαδικό $(110101)_2$ σέ δεκαδικό θά ένεργήσομε ως εξής:

$$\begin{aligned} 1 \times 2 &= 2 \\ 2 + 1 &= 3 \\ 3 \times 2 &= 6 \\ 6 + 0 &= 6 \\ 6 \times 2 &= 12 \\ 12 + 1 &= 13 \\ 13 \times 2 &= 26 \\ 26 + 0 &= 26 \\ 26 \times 2 &= 52 \\ 52 + 1 &= \boxed{53} \\ \text{άρα } (110101)_2 &= (53)_{10} \end{aligned}$$

ε) Μετατροπή κλασματικού δυαδικού αριθμού σέ δεκαδικό.

Γιά νά μετατρέψομε ένα κλασματικό δυαδικό αριθμό σέ δεκαδικό διαιρούμε τό τελευταίο ψηφίο του αριθμού (δηλαδή αυτό πού έχει τή μικρότερη αξία) μέ τή βάση 2. Στο πηλίκο προσθέτομε τό άμέσως προηγούμενο ψηφίο του δυαδικού και διαιρούμε πάλι μέ τό 2. Στο νέο πηλίκο προσθέτομε τό τρίτο άπό τά δεξιά ψηφίο και συνεχίζομε μέ αυτόν τόν τρόπο ώσπου νά φθάσομε στό τελευταίο πρós τά άριστερά ψηφίο του αριθμού π.χ. γιά τή μετατροπή του $(0.1101)_2$ σέ δεκαδικό, θά κάνομε τά ακόλουθα βήματα:

$$\begin{aligned} 1 &: 2 = 0.5 \\ 0.5 &+ 0 = 0.5 \\ 0.5 &: 2 = 0.25 \\ 0.25 &+ 1 = 1.25 \\ 1.25 &: 2 = 0.63 \\ 0.63 &+ 1 = 1.63 \\ 1.63 &: 2 = \boxed{0.81} \\ \text{άρα } (0.1101)_2 &= (0.81)_{10} \end{aligned}$$

στ) Μετατροπή άκεραίου δεκαδικού αριθμού σε όκταδικό.

Η μετατροπή γίνεται διαιρώντας διαδοχικά τό δοθέντα αριθμό μέ τή βάση 8. Τά υπόλοιπα από τίς διαιρέσεις τοποθετημένα σε αντίστροφη σειρά μάς δύνουν τό ζητούμενο όκταδικό. π.χ. γιά τόν $(79)_{10}$ θά έχομε:

$$79 : 8 = 9 \quad \text{καί} \quad \text{υπόλοιπο} \quad 7$$

$$9 : 8 = 1 \quad \text{καί} \quad \text{υπόλοιπο} \quad 1$$

$$1 : 8 = 0 \quad \text{καί} \quad \text{υπόλοιπο} \quad 1$$

$$\text{άρα } (79)_{10} = (117)_8$$

ζ) Μετατροπή άκεραίου όκταδικού αριθμού σε δεκαδικό.

Πολλαπλασιάζομε τό πρώτο ψηφίο τοῦ αριθμοῦ μέ τό 8 καί στό γινόμενο προσθέτομε τό επόμενο ψηφίο. Πολλαπλασιάζομε τό άθροισμα πάλι μέ τό 8 καί προσθέτομε τό ψηφίο τοῦ αριθμοῦ κ.ο.κ. Π.χ. ό όκταδικός $(117)_8$ μετατρέπεται σε δεκαδικό ως εξής:

$$1 \times 8 = 8$$

$$8 + 1 = 9$$

$$9 \times 8 = 72$$

$$72 + 7 = \boxed{79}$$

$$\text{άρα } (117)_8 = (79)_{10}$$

η) Μετατροπή άκεραίου δεκαδικού αριθμού σε δεκαεξαδικό.

Χρησιμοποιούμε τή μέθοδο, πού περιγράψαμε στίς παραγράφους 2.6(β) καί 2.6(στ) μόνο πού έδω θεωρούμε ως βάση τό 16. Έτσι, γιά νά μετατρέψομε τόν άκέραιο δεκαδικό αριθμό $(146)_{10}$ σε δεκαεξαδικό, θά ακολουθήσομε τά ακόλουθα στάδια:

$$146 : 16 = 9 \quad \text{καί} \quad \text{υπόλοιπο} \quad 2$$

$$9 : 16 = 0 \quad \text{καί} \quad \text{υπόλοιπο} \quad 9$$

$$\text{άρα } (146)_{10} = (92)_{16}$$

θ) Μετατροπή άκεραίου δεκαεξαδικού αριθμού σε δεκαδικό.

Χρησιμοποιώντας τή μέθοδο πού είδαμε στίς παραγράφους 2.6(δ) καί 2.6(ζ) μπορούμε νά μετατρέψομε τόν δεκαεξαδικό $(4EF)_{16}$ σε δεκαδικό ως εξής:

$$4 \times 16 = 64$$

$$64 + E = 78$$

$$78 \times 16 = 1248$$

$$1248 + F = \boxed{1263}$$

$$\text{άρα } (4EF)_{16} = (1263)_{10}$$

ι) Μετατροπή μή δεκαδικού αριθμού σε άλλο μή δεκαδικό σύστημα.

Όταν θέλομε νά μετατρέψομε έναν αριθμό μή δεκαδικό σε κάποιο άλλο μή δεκαδικό σύστημα, τότε χρησιμοποιούμε τό δεκαδικό σύστημα ως ενδιάμεσο. Δηλαδή, γιά νά μετατρέψομε ένα πενταδικό αριθμό σε δεκαεξαδικό, μετατρέπομε πρώτο τόν πενταδικό σε δεκαδικό καί κατόπιν τόν δεκαδικό, πού βρήκαμε, σε δεκαεξαδικό.

Στήν ειδική περίπτωση, πού θέλουμε νά μετατρέψουμε ἓνα δυαδικό ἀριθμό σέ κάποιο ἄλλο σύστημα, πού ἔχει ὡς βάση μιά δύναμη τοῦ 2, ὅπως εἶναι π.χ. τὸ ὀκταδικό καί τὸ δεκαεξαδικό, ὑπάρχει ἀπλούστερη μέθοδος ὅπως θά δοῦμε λίγο πῶς κά-τω.

2.7 Ἀριθμητικές πράξεις στό δυαδικό σύστημα.

α) Πρόσθεση.

Ἡ πρόσθεση δύο δυαδικῶν ἀριθμῶν γίνεται, ὅπως καί στοὺς δεκαδικούς ἀριθμούς. Ξεκινώντας ἀπὸ τὴν πρόσθεση τῶν τελευταίων ψηφίων τῶν ἀριθμῶν καί συνεχίζοντας πρὸς τὰ ἀριστερά. Πρέπει ὅμως νά ἔχομε ὑπ' ὄψη μας τοὺς ἐξῆς βα-σικούς κανόνες

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \quad \text{καί κρατούμενο γιὰ μεταφορά } 1$$

Παραδείγματα:

$$\begin{array}{r} (100)_2 \\ + (1)_2 \\ \hline (101)_2 \end{array}$$

$$\begin{array}{r} (1011)_2 \\ + (10)_2 \\ \hline (1101)_2 \end{array}$$

$$\begin{array}{r} (110111)_2 \\ + (1011)_2 \\ \hline (1000010)_2 \end{array}$$

β) Ἀφαίρεση.

Ἀφαιροῦμε τὰ ἀντίστοιχα ψηφία τῶν ἀριθμῶν ἀρχίζοντας ἀπὸ τὰ τελευταῖα, ἔχοντας ὑπ' ὄψη μας τοὺς ἐξῆς κανόνες:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \quad \text{μὲ δανεισμό ἀπὸ τὸ προηγούμενο ψηφίο μιᾶς μονάδας 1.}$$

Παραδείγματα:

$$\begin{array}{r} (1011)_2 \\ - (10)_2 \\ \hline (1001)_2 \end{array}$$

$$\begin{array}{r} (1010)_2 \\ - (101)_2 \\ \hline (101)_2 \end{array}$$

$$\begin{array}{r} (11011)_2 \\ - (1111)_2 \\ \hline (1100)_2 \end{array}$$

γ) Πολλαπλασιασμός.

Γίνεται ὅπως καί στό δεκαδικό σύστημα. Γιὰ τίς ἐνδιάμεσες προσθέσεις χρησιμοποιοῦμε τοὺς κανόνες τῆς δυαδικῆς προσθέσεως.

Παραδείγματα:

$$\begin{array}{r}
 (110)_2 \\
 \times (11)_2 \\
 \hline
 110 \\
 110 \\
 \hline
 (10010)_2
 \end{array}$$

$$\begin{array}{r}
 (1101)_2 \\
 \times (101)_2 \\
 \hline
 1101 \\
 0000 \\
 1101 \\
 \hline
 (1000001)_2
 \end{array}$$

$$\begin{array}{r}
 (1001)_2 \\
 \times (111)_2 \\
 \hline
 1001 \\
 1001 \\
 1001 \\
 \hline
 (111111)_2
 \end{array}$$

δ) Διαίρεση.

Καί αυτή ή πράξη γίνεται όπως στο δεκαδικό σύστημα. Για τις ένδιάμεσες αφαιρέσεις χρησιμοποιούμε τούς κανόνες της δυαδικής αφαιρέσεως.

Παραδείγματα:

$$\begin{array}{r|l}
 (1101)_2 & (11)_2 \\
 \hline
 11 & (100)_2 \\
 \hline
 000 & \\
 - 00 & \\
 \hline
 01 & \\
 - 00 & \\
 \hline
 01 &
 \end{array}$$

$$\begin{array}{r|l}
 (110110)_2 & (110)_2 \\
 \hline
 110 & (1001)_2 \\
 \hline
 0001 & \\
 - 000 & \\
 \hline
 011 & \\
 - 000 & \\
 \hline
 110 & \\
 - 110 & \\
 \hline
 000 &
 \end{array}$$

$$\begin{array}{r|l}
 (111)_2 & (10)_2 \\
 \hline
 10 & (11)_2 \\
 \hline
 11 & \\
 - 10 & \\
 \hline
 01 &
 \end{array}$$

2.8 Αριθμητικές πράξεις στο όκταδικό σύστημα.**α) Πρόσθεση.**

Προσθέτουμε τά αντίστοιχα ψηφία τών αριθμών αρχίζοντας από τό τέλος. Κάθε φορά πού τό άθροισμά τους ξεπερνά τό 8 σημειώνουμε τή διαφορά από τό 8 καί μεταφέρουμε μιά όκταδική μονάδα, γιά νά τήν προσθέσουμε στό άθροισμα τών άμέσως προηγούμενων ψηφίων.

Παραδείγματα:

$$\begin{array}{r}
 (746)_8 \\
 + (23)_8 \\
 \hline
 (771)_8
 \end{array}$$

$$\begin{array}{r}
 (1374)_8 \\
 + (6510)_8 \\
 \hline
 (10104)_8
 \end{array}$$

β) Αφαίρεση.

Αφαιρούμε τα αντίστοιχα ψηφία των αριθμών αρχίζοντας από το τέλος. Αν όμως ένα ψηφίο του αφαιρέτη είναι μεγαλύτερο από το αντίστοιχο του αφαιρέτου, τότε δανειζόμαστε μία όκταδική μονάδα από το άμεσως προηγούμενο ψηφίο του διαιρέτη.

Παραδείγματα:

$$\begin{array}{r} (456)_8 \\ - (72)_8 \\ \hline (364)_8 \end{array} \qquad \begin{array}{r} (224)_8 \\ - (15)_8 \\ \hline (207)_8 \end{array}$$

γ) Πολλαπλασιασμός.

Γίνεται όπως και στο δεκαδικό, και χρησιμοποιούμε για τις ενδιάμεσες προσθέσεις τον κανόνα που αναφέραμε πιο πάνω.

Παραδείγματα:

$$\begin{array}{r} (24)_8 \\ \times (12)_8 \\ \hline 50 \\ 24 \\ \hline (310)_8 \end{array} \qquad \begin{array}{r} (47)_8 \\ \times (12)_8 \\ \hline 116 \\ 47 \\ \hline (606)_8 \end{array}$$

ε) Δίαιρεση.

Γίνεται όπως και στο δεκαδικό, χρησιμοποιώντας όμως για τις ενδιάμεσες αφαιρέσεις τον κανόνα της αφαίρεσης, που είδαμε προηγουμένως.

Παραδείγματα:

$$\begin{array}{r} (145)_8 \quad | \quad (12)_8 \\ - 12 \quad | \quad (12)_8 \\ \hline 25 \\ - 24 \\ \hline 01 \end{array} \qquad \begin{array}{r} (477)_8 \quad | \quad (6)_8 \\ - 44 \quad | \quad (65)_8 \\ \hline 037 \\ - 36 \\ \hline 01 \end{array}$$

Οι αριθμητικές πράξεις στα άλλα αριθμητικά συστήματα εκτελούνται με έντε-λώς παρόμοιο τρόπο.

2.9 Δυαδικοί κώδικες.

Κώδικας είναι ένα σύστημα συμβόλων, το καθένα από τα οποία παριστάνει σύμφωνα με κάποιο νόμο κάποιο χαρακτήρα, που ανήκει σε άλλο σύστημα. Η διεργασία της αντιστοιχίσεως που είναι αμφιμονοσήμαντη καλείται κωδικοποίηση.

Όλες οι πληροφορίες, που με αυτές τροφοδοτείται ένας υπολογιστής κωδικοποιούνται με βάση το δυαδικό σύστημα. Δηλαδή αντιστοιχίζουμε σε κάθε χαρακτήρα αλφαριθμητικό ή αριθμητικό ένα δυαδικό αριθμό. Τέτοιοι τρόποι αντιστοιχίσεως υπάρχουν πολλοί και αποτελούν τους λεγόμενους δυαδικούς κώδικες. Από τους δυαδικούς κώδικες θα περιγράψουμε τους κυριότερους.

α) Κωδικοποίηση αριθμητικών συστημάτων με βάση $B = 2$.

Στήν περίπτωση που ένας αριθμός ανήκει σε ένα αριθμητικό σύστημα που έχει βάση μία δύναμη του 2 (τέτοια συστήματα είναι το οκταδικό και το δεκαεξαδικό, από αυτά που γνωρίζουμε), μπορούμε να τον μετατρέψουμε σε δυαδικό απ' ευθείας αν κάθε του ψηφίο το αντικαταστήσουμε με τόσα δυαδικά ψηφία, όση είναι η δύναμη του 2. Π.χ. κάθε ψηφίο του οκταδικού συστήματος, που έχει βάση τό $8 = 2^3$ αντιστοιχεί σε 3 δυαδικά ψηφία, ενώ κάθε ψηφίο δεκαεξαδικού, που έχει βάση τό $16 = 2^4$ αντιστοιχεί σε 4 δυαδικά ψηφία κλπ. Βασιζόμενοι στα προηγούμενα δημιουργούμε τον ακόλουθο πίνακα αντιστοιχιών.

ΠΙΝΑΚΑΣ 2.9.1.

Κωδικοποίηση των ψηφίων του οκταδικού και δεκαεξαδικού συστήματος με δυαδικά ψηφία

Ψηφία του οκταδικού συστήματος	Τά οκταδικά ψηφία κωδικοποιημένα με δυαδικά ψηφία	Ψηφία του δεκαεξαδικού συστήματος	Τά δεκαεξαδικά ψηφία κωδικοποιημένα με δυαδικά ψηφία
0	000	0	$\%0000$
1	001	1	0001
2	010	2	0010
3	011	3	0011
4	100	4	0100
5	101	5	0101
6	110	6	0110
7	111	7	0111
		8	1000
		9	1001
		A	1010
		B	1011
		C	1100
		D	1101
		E	1110
		F	1111

$$\begin{aligned} \text{Έτσι ο αριθμός } (123)_8 &= \overset{1}{001} \overset{2}{010} \overset{3}{011} \\ \text{ο αριθμός } (59AF)_{16} &= \overset{5}{0101} \overset{9}{1001} \overset{A}{1010} \overset{F}{1111} \end{aligned}$$

Αν αντίστροφα μας δοθεί ο δυαδικός αριθμός και ζητηθεί ο αντίστοιχος οκταδικός ή δεκαεξαδικός, τότε χωρίζουμε τα ψηφία του δυαδικού σε τριάδες ή τετράδες αντίστοιχα, αρχίζοντας από το πρώτο και χρησιμοποιώντας τον ίδιο πίνακα γράφοντας στη σειρά τα αντίστοιχα ψηφία του οκταδικού ή δεκαεξαδικού. Αν τό πλήθος των ψηφίων του δυαδικού δεν διαιρείται με τό 3 ή τό 4 συμπληρώνουμε στην αρχή μηδενικά.

π.χ. ο αριθμός 110 111 011 θά μας δώσει:

Γιά τό όκταδικό $110 \mid 111 \mid 011$
 $6 \mid 7 \mid 3 \rightarrow (673)_8$

Γιά τό δεκαεξαδικό θά πρέπει νά προσθέσομε στήν άρχή τρία μηδενικά γιά νά μπορέσομε νά κάνομε τό χωρισμό τών ψηφίων. Δηλαδή:

$0001 \mid 1011 \mid 1011$
 $1 \mid B \mid B \rightarrow (1BB)_{16}$

β) Ο κώδικας B.C.D (Binary Coded Decimal).

Ο κώδικας αυτός χρησιμοποιείται συχνά γιά τήν παράσταση τών αριθμών μέσα στόν ύπολογιστή. Κάθε αριθμητικός χαρακτήρας κωδικοποιείται μέ τέσσερα δυαδικά ψηφία, πού έχουν άξία 8,4,2,1 αντίστοιχα. Τήν αντίστοιχία τήν βλέπομε στόν ακόλουθο πίνακα:

ΠΙΝΑΚΑΣ 2.9.2.

Ο κώδικας BCD

Ψηφίο δεκαδικού συστήματος	BCD			
	8	4	2	1
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

π.χ. Μέ τή βοήθεια του κώδικα αυτού ο δεκαδικός $(954)_{10}$ γράφεται:

Δεκαδικός	9	5	4
BCD	1001	0101	0100

Τό πλεονέκτημα του κώδικα BCD είναι ή μεγάλη εύκολία, μέ τήν όποία ένας αριθμός παριστάνεται μέ δυαδικά ψηφία, ενώ τό μειονέκτημα είναι ή μεγαλύτερη δυσκολία στήν εκτέλεση αριθμητικών πράξεων.

γ) 6ψηφιος άλφαριθμητικός κώδικας

Στόν κώδικα αυτό κάθε χαρακτήρας παριστάνεται μέ 6 δυαδικά ψηφία. Από αύ-

τά τα δύο πρώτα, πού τα συμβολίζουμε γενικά με Β και Α, αποτελούν τό λεγόμενο τμήμα ζώνης. Τά επόμενα 4 ψηφία έχουν άξια κατά σειρά 8, 4, 2, 1 καί αποτελούν τό αριθμητικό τμήμα. Τόν κώδικα αυτό βλέπομε στόν παρακάτω πίνακα.

ΕΞΑΨΗΦΙΟΣ ΑΛΦΑΡΙΘΜΗΤΙΚΟΣ ΚΩΔΙΚΑΣ

Χαρακτήρας	Δυαδική παράσταση ΒΑ, 8, 4, 2, 1	Χαρακτήρας	Δυαδική παράσταση ΒΑ, 8, 4, 2, 1
0	001010	R	101001
1	000001	S	010010
2	000010	T	010011
3	000011	U	010100
4	000100	Y	010101
5	000101	W	010110
6	000110	X	010111
7	000111	Y	011000
8	001000	Z	011001
9	001001	KENO	000000
A	110001	.	111011
B	110010	[	111101
C	110011	<	111110
D	110100	\$	101011
E	110101	*	101100
F	110110	]	101101
G	110111	;	101110
H	111000	-	100000
I	111001	/	010001
J	100001	,	011011
K	100010	%	011100
L	100011	\	011110
M	100100	@	001100
N	100101	:	001101
O	100110	>	001110
P	100111	?	111010
Q	101000	!	101010

δ) 'Ο Κώδικας EBCDIC (Extended Binary Coded Decimal Interchange Code).

Ό κώδικας αυτός χρησιμοποιείται όπως καί ό BCD, με τή διαφορά ότι κάθε χαρακτήρας παριστάνεται με 8 δυαδικά ψηφία. Από αυτά τά 4 πρώτα αποτελούν τό λεγόμενο *τμήμα ζώνης* καί τά 4 επόμενα τό *αριθμητικό τμήμα*. Η αντίστοιχία ανάμεσα στους διάφορους χαρακτήρες καί στους δυαδικούς αριθμούς του EBCDIC φαίνεται στόν επόμενο πίνακα.

ΠΙΝΑΚΑΣ 2.9.4.

Ἡ δυαδική παράσταση κάθε χαρακτήρα ἀποδίδεται μέ ὀκτώ δυαδικά ψηφία πού ἀποτελοῦν ἓνα byte¹

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		

ε) Κώδικας Hollerith.

Εἶναι ὁ πρῶτος δυαδικός κώδικας πού χρησιμοποιήθηκε γιά ἀπεικόνιση πληροφοριῶν. Δημιουργήθηκε ἀπό τόν *Herman Hollerith*. Στόν κώδικα αὐτόν, κάθε χαρακτήρας ἀπεικονίζεται μέ 12 δυαδικά ψηφία (bits) πού ἀντιστοιχοῦν στίς 12 θέσεις διατρήσεως μιᾶς στήλης ἑνός δελτίου 80 στήλων. Οἱ 10 τελευταῖες θέσεις

¹ Ἐνα byte εἶναι μιά ὁμάδα πού ἀποτελεῖται συνήθως ἀπό ὀκτώ δυαδικά ψηφία (bits).

ψηφίων αποτελούν τό ψηφιακό τμήμα, τοῦ κώδικα, ἐνῶ οἱ τρεῖς πρῶτες τό τμήμα ζώνης (μία θέση εἶναι κοινή). Ὁ πλήρης πίνακας κωδικοποιήσεως τῶν χαρακτήρων παρουσιάζεται πῶ κάτω (παράγραφος 3.4α). Μὲ τόν κώδικα αὐτό μποροῦν νά ἀπεικονισθοῦν μέχρι 120 διαφορετικοί χαρακτήρες.

2.10 Δυαδικά ψηφία ἰσοτιμίας.

Ἡ λειτουργία ἐνός ὑπολογιστῆ χαρακτηρίζεται ἀπό συνεχή μετακίνηση μεγάλου ὄγκου δεδομένων κατὰ τή διάρκεια τῆς ἐπεξεργασίας. Παρ' ὅλη τήν τεχνική τελειότητα τῆς μηχανῆς, μπορεῖ νά συμβοῦν σφάλματα ἢ νά χαθοῦν πληροφορίες, ἐπειδή κάποιο ἀπό τά ἠλεκτρονικά κυκλώματα δέν λειτούργησε σωστά γιά ἓνα κλάσμα τοῦ δευτερολέπτου. Γιά τό λόγο αὐτό οἱ σχεδιαστές ὑπολογιστῶν ἐφοδιάζουν τίς μηχανές μέ διάφορους μηχανισμούς ἐλέγχου σφαλμάτων. Ἀνάμεσά τους ὁ πῶ γνωστός φέρει τήν ὀνομασία *δυαδικό ψηφίο ἰσοτιμίας*¹ (parity bit ἢ binary check bit). Τό ψηφίο ἰσοτιμίας προστίθεται σέ κάθε δυαδική ἀπεικόνιση δεδομένων. Ἐπομένως ἂν χρησιμοποιεῖται ὁ τετραψήφιος BCD κώδικας μέ τήν προσθήκη τοῦ ψηφίου ἰσοτιμίας θά γίνῃ πενταψήφιος, ἂν χρησιμοποιεῖται ὁ ἐξαψήφιος ἀλφαριθμητικός θά γίνῃ ἑπταψήφιος ἂν δέ χρησιμοποιεῖται ὁ ὀκταψήφιος EBCDIC, θά γί-

ΠΙΝΑΚΑΣ 2.10.1.

Ὁ κώδικας 8421 μέ ἄρτια καί περιττή ἰσοτιμία

Ψηφία δεκαδικοῦ συστήματος	Κωδικοποίηση μέ τόν Κώδικα 8, 4, 2, 1	Κωδικοποίηση μέ τόν Κώδικα 8, 4, 2, 1 καί μέ ἄρτια ἰσοτιμία	Κωδικοποίηση μέ τόν Κώδικα 8, 4, 2, 1 καί μέ περιττή ἰσοτιμία
	8 4 2 1	8 4 2 1 Α	8 4 2 1 Π
0	0 0 0 0	0 0 0 0 0	0 0 0 0 1
1	0 0 0 1	0 0 0 1 1	0 0 0 1 0
2	0 0 1 0	0 0 1 0 1	0 0 1 0 0
3	0 0 1 1	0 0 1 1 0	0 0 1 1 1
4	0 1 0 0	0 1 0 0 1	0 1 0 0 0
5	0 1 0 1	0 1 0 1 0	0 1 0 1 1
6	0 1 1 0	0 1 1 0 0	0 1 1 0 1
7	0 1 1 1	1 0 0 0 1	0 1 1 1 0
8	1 0 0 0	1 0 0 0 1	1 0 0 0 0
9	1 0 0 1	1 0 0 1 0	1 0 0 1 1

νει ἐννεαψήφιος. Μὲ τόν τρόπο αὐτό ἀποκαθιστοῦμε, ὅπως λέμε, μία ἰσοτιμία με-
ταξύ τῶν κωδικοποιημένων ἀριθμῶν. Ἡ ἰσοτιμία αὐτή εἶναι δύο εἰδῶν:

Ἄρτια ἰσοτιμία. (Even parity bit). Σύμφωνα μέ αὐτή, προσθέτομε στό τέλος τοῦ κωδικοποιημένου χαρακτήρα τό 0 ἢ τό 1 ἔτσι ὥστε μετά ἀπό αὐτή τήν προσθήκη,

¹ Ἀπό δῶ καί πέρα τό δυαδικό ψηφίο θά τό ὀνομάζομε bit.

ὁ κωδικοποιημένος χαρακτήρας νά ἔχει ἄρτιο συνολικό πλῆθος μονάδων 1.

Περιττή ἰσοτιμία (Odd parity bit) Σύμφωνα μέ αὐτή, προσθέτομε σέ κάθε κωδικοποιημένο χαρακτήρα τό 0 ἢ τό 1 ἔτσι, ὥστε μετά τήν προσθήκη αὐτή ὁ κωδικοποιημένος χαρακτήρας νά ἔχει περιττό πλῆθος μονάδων 1. Τήν ἐφαρμογή τῆς ἄρτιας ἢ περιττῆς ἰσοτιμίας στόν τετραψήφιο κώδικα BCD βλέπομε στόν παρακάτω πίνακα.

Ἀσκήσεις.

1. Νά μετατραποῦν οἱ δεκαδικοί ἀριθμοί 543, 156, 256 σέ δυαδικούς, ὀκταδικούς καί δεξαεξαδικούς.
2. Νά μετατραποῦν οἱ δεκαδικοί ἀριθμοί 15, 112, 164, 34, 76, 07 σέ δυαδικούς.
3. Νά μετατραποῦν οἱ ὀκταδικοί 175, 24, 502 σέ δεκαδικούς.
4. Νά μετατραποῦν οἱ δεκαεξαδικοί 1AD, 454E, 22F σέ δεκαδικούς.
5. Νά μετατραποῦν οἱ ὀκταδικοί 54, 73, 601 σέ δυαδικούς.
6. Νά μετατραποῦν οἱ δεκαεξαδικοί 4EF2, AB, 146 σέ δυαδικούς.
7. Τό τριαδικό σύστημα ἀποτελεῖται ἀπό τὰ ψηφία 0, 1, 2 καί ἔχει βάση τό 3. Χρησιμοποιώντας κατάλληλα τή μέθοδο τῆς παραγράφου 2.6(β) νά μετατρέψετε τόν δεκαδικό 4789 σέ τριαδικό.
8. Νά γίνουν οἱ ἀκόλουθες πράξεις ἀνάμεσα στούς δυαδικούς ἀριθμούς:

$\begin{array}{r} 101101 \\ 1111 \\ \hline \end{array}$	$\begin{array}{r} 1001101 \\ +1110011 \\ \hline \end{array}$	$\begin{array}{r} 1011110 \\ - 10111 \\ \hline \end{array}$	$\begin{array}{r} 111011 \\ \div 111 \\ \hline \end{array}$
---	--	---	---

9. Νά γίνουν οἱ ἐξῆς πράξεις ἀνάμεσα στούς ὀκταδικούς ἀριθμούς:

$\begin{array}{r} 4765 \\ +5231 \\ \hline \end{array}$	$\begin{array}{r} 3762 \\ - 145 \\ \hline \end{array}$	$\begin{array}{r} 23501 \\ + 7647 \\ \hline \end{array}$	$\begin{array}{r} 5243 \\ - 351 \\ \hline \end{array}$
--	--	--	--

10. Νά γίνουν οἱ ἐξῆς πράξεις ἀνάμεσα στούς ὀκταδικούς ἀριθμούς:

$\begin{array}{r} 236 \\ \times 25 \\ \hline \end{array}$	$\begin{array}{r} 310 \\ \div 62 \\ \hline \end{array}$	$\begin{array}{r} 325 \\ \times 12 \\ \hline \end{array}$	$\begin{array}{r} 416 \\ \div 132 \\ \hline \end{array}$
---	---	---	--

11. Νά μετατραποῦν οἱ δυαδικοί 1100110101 καί 101111001 σέ ὀκταδικούς καί δεκαεξαδικούς μέ τή βοήθεια τοῦ πίνακα 2.9.1.
12. Νά μετατραποῦν οἱ ὀκταδικοί 376 καί 1432 καί οἱ δεκαεξαδικοί 4ED2 καί 18CAF σέ δυαδικούς, μέ τή βοήθεια τοῦ πίνακα 2.9.1.
13. Πῶς θά κωδικοποιηθοῦν οἱ δεκαδικοί ἀριθμοί 145, 36, 2051 στόν κώδικα BCD μέ ἄρτια ἰσοτιμία;
14. Πῶς θά κωδικοποιηθοῦν οἱ δεκαδικοί ἀριθμοί 48, 17, 163 στόν κώδικα BCD μέ περιττή ἰσοτιμία;
15. Πῶς θά κωδικοποιηθοῦν οἱ δεκαδικοί ἀριθμοί 157, 249 στόν κώδικα EBCDIC.

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

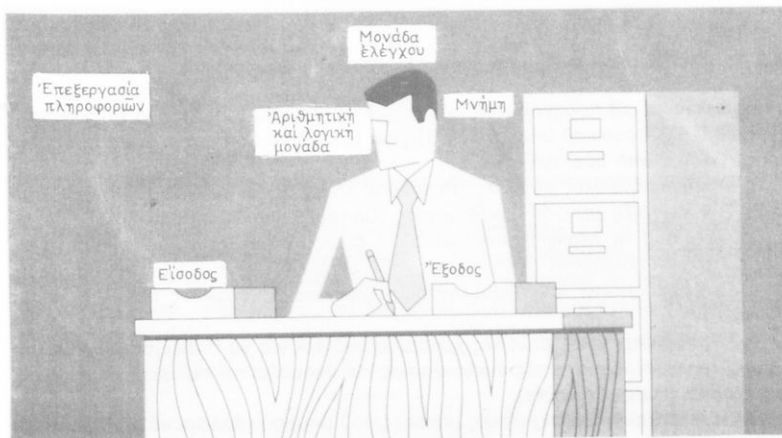
ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΨΗΦΙΑΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ I

3.1 Γενικά.

Γιά να μπορέσει ο άνθρωπος να λύσει κάποιο πρόβλημα, χρειάζεται να του δοθούν ορισμένες πληροφορίες, που είναι από τη μιά μεριά τα δεδομένα (Data) του προβλήματος και από την άλλη οι οδηγίες (Instructions) για τον τρόπο με τον οποίο θα τό επιλύσει. Οι πληροφορίες αυτές του παρέχονται γραπτές ή προφορικές.

Έχοντας τώρα πάρει τις πληροφορίες ο άνθρωπος μπορεί να επεξεργασθεί τό πρόβλημα και να καταλήξει σε αποτελέσματα. Έτσι π.χ. για να λύσει μιά δευτέρου βαθμού εξίσωση με ένα άγνωστο, $ax^2 + bx + \gamma = 0$, θα πρέπει να του δοθεί ή μορφή της εξίσωσης — δηλαδή οι συντελεστές του άγνωστου a, b και ο σταθερός όρος γ — καθώς και ο τρόπος της λύσεως, γραμμένα πάνω σε κάποιο φορέα π.χ. σε ένα κομμάτι χαρτί. Οι τιμές των a, b και γ , αποτελούν στην περίπτωση αυτή τά δεδομένα του προβλήματος, ή δέ περιγραφή του τρόπου λύσεως είναι οι οδηγίες.

Μέ αυτές τις πληροφορίες ο άνθρωπος χρησιμοποιώντας στη συνέχεια τό μυαλό του, θα προβεί στους αναγκαίους υπολογισμούς και τά αποτελέσματα, που θα βρει — δηλαδή τις τιμές του άγνωστου — θα τά γράψει πάλι σε ένα φορέα, ένα άλλο κομμάτι χαρτί, ώστε να μπορεί να τά διαβάσει και κάποιος άλλος (σχ. 3.1α).



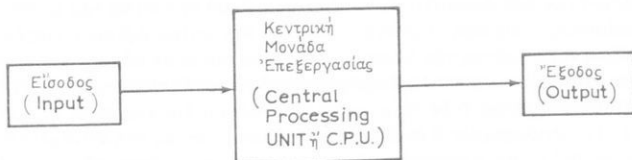
Σχ. 3.1α.

Η επεξεργασία πληροφοριών από τον άνθρωπο.

Μιά παρόμοια διαδικασία ακολουθείται για νά λυθεί κάποιο πρόβλημα από τόν ηλεκτρονικό υπολογιστή. Έτσι καί εδώ θά πρέπει νά δοθούν στόν υπολογιστή οι αναγκαίες πληροφορίες, δηλαδή τά δεδομένα καί οι οδηγίες γραμμένες πάνω σέ κάποιο φορέα, τίς όποιες κατόπιν θά ακολουθήσει έκτελώντάς τις καί θά βγάλει άποτελέσματα. Βέβαια οι πληροφορίες δίνονται στόν υπολογιστή κωδικοποιημένες σέ κάποιον από τούς κώδικες του δυαδικού συστήματος, πού γνωρίσαμε ήδη, γιατί ό υπολογιστής δέν καταλαβαίνει παρά μόνο τή γλώσσα τών δυαδικών άριθμών.

Τό τμήμα του υπολογιστή, από όπου εισάγομε σ' αυτόν τίς διάφορες πληροφορίες, ονομάζεται *είσοδος* (Input), ενώ τό τμήμα από όπου παίρνομε τά άποτελέσματα, *έξοδος* (Output). 'Η είσοδος καί ή έξοδος στήν πραγματικότητα άποτελούνται από ξεχωριστές μονάδες διαφόρων τύπων πού ονομάζονται *περιφερειακές μονάδες* είσόδου-έξόδου αντίστοιχα (I/O Peripheral Units) καί ή καθεμιά τους μπορεί νά διαβάζει τίς πληροφορίες ή νά γράφει τά άποτελέσματα πάνω σέ κάποιο συγκεκριμένο φορέα.

Τό τμήμα του υπολογιστή, όπου γίνεται ή έπεξεργασία τών πληροφοριών, ονομάζεται *κεντρική μονάδα έπεξεργασίας* ή C.P.U (Central Processing Unit), (σχ. 3.1β).



Σχ. 3.1β.

‘Η έπεξεργασία τών πληροφοριών από τόν ηλεκτρονικό υπολογιστή.

3.2 Τά βασικά μέρη ενός ψηφιακού ηλεκτρονικού υπολογιστή.

Σύμφωνα μέ τά προηγούμενα, μπορούμε νά θεωρήσομε ότι ένα συγκρότημα ψηφιακού ηλεκτρονικού υπολογιστή (Digital Electronic Computer System) άποτελείται βασικά από δύο τμήματα.

α) Τίς *περιφερειακές μονάδες* (Peripheral Units ή I/O Devices).

β) Τήν *κεντρική μονάδα έπεξεργασίας* (Central Processing Unit ή C.P.U.).

Μερικές από τίς περιφερειακές μονάδες, όπως είδαμε, χρησιμεύουν για νά εισάγομε στόν υπολογιστή πληροφορίες, πού είναι δεδομένα (Data) καί οδηγίες - έντολές (Instructions) καί ονομάζονται *περιφερειακές μονάδες είσόδου* (Input Devices).

Άλλες περιφερειακές μονάδες χρησιμοποιούνται για νά αποδώσουν στόν άνθρωπο τά άποτελέσματα τής έπεξεργασίας καί ονομάζονται *περιφερειακές μονάδες έξόδου* (Output devices).

Τέλος υπάρχουν περιφερειακές μονάδες πού μπορούν άλλοτε νά χρησιμοποιηθούν ως μονάδες είσόδου καί άλλοτε ως μονάδες έξόδου (μικτές).

‘Η κεντρική μονάδα έπεξεργασίας είναι τό σημαντικότερο τμήμα ενός ψηφιακού υπολογιστή, ή «καρδιά» ή καλύτερα ό έγκέφαλος του συγκροτήματος. Σ’ αυτήν

πραγματοποιείται ή έπεξεργασία τών πληροφοριών. Δέν είναι όμως μιά ένιαία μονάδα, αλλά αποτελείται από τρία κυρίως τμήματα.

α) Τήν κεντρική ή κύρια ή έσωτερική μνήμη (Central or main or internal memory).

β) Τήν αριθμητική καί λογική μονάδα (Arithmetic and Logical Unit).

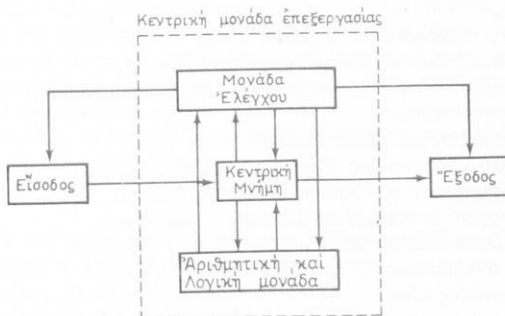
γ) Τή μονάδα έλέγχου (Control Unit).

Η κεντρική μνήμη είναι τό τμήμα τής κεντρικής μονάδας έπεξεργασίας, όπου αποθηκεύονται — κωδικοποιημένα δυαδικά — τά δεδομένα, οι έντολές καί τά ένδιάμεσα αποτελέσματα.

Η αριθμητική καί λογική μονάδα είναι τό τμήμα όπου εκτελούνται οι προβλεπόμενες αριθμητικές καί λογικές πράξεις¹ ανάμεσα στά δεδομένα.

Η μονάδα έλέγχου, έλέγχει, καθοδηγεί καί γενικά συντονίζει τή λειτουργία όλων τών μονάδων του συγκροτήματος.

Στό σχήμα 3.2 βλέπομε σχηματικά τή σχέση συνδέσεως ανάμεσα στά διάφορα τμήματα του υπολογιστή.



Σχ. 3.2.

Ο τρόπος συνδέσεως ανάμεσα στά διάφορα τμήματα του υπολογιστή.

3.3 Φορείς καί μέσα καταγραφής πληροφοριών - Περιφερειακές μονάδες.

Όπως είπαμε πιό πάνω γιά νά εισάγομε τίς πληροφορίες σ' ένα ψηφιακό υπολογιστή πρέπει προηγουμένως νά τίς έγγράφομε σέ κωδικοποιημένη μορφή πάνω σέ κάποιο φορέα. Σέ κωδικοποιημένη ή καί συνηθισμένη μορφή θά πάρομε καί τά αποτελέσματα τής έπεξεργασίας. Τά μέσα καταγραφής πληροφοριών τά διακρίνομε σέ δύο κατηγορίες:

- Μή μαγνητικά.
- Μαγνητικά.

1. Μέ τόν δοο λογικές πράξεις έννοοϋμε τίς λογικές συγκρίσεις ή τόν έλεγχο λογικών συνθηκών.
π.χ. $a > b$ ή $b < x < y$ κλπ.



Στά *μή μαγνητικά μέσα* περιλαμβάνονται:

- Τό δελτίο (Card).
- Ή χαρτοταινία (Papertape).
- Τό χαρτί έκτυπώσεως (Paper).
- Τά δελτία ή έντυπα μέ όπτικά άναγνωριζόμενους χαρακτήρες (Optical Characters Recognition ή O.C.R).

Στά *μαγνητικά μέσα* περιλαμβάνονται:

- Ή μαγνητική ταινία (Magnetic Tape).
- Ό μαγνητικός δίσκος (Magnetic Disk).
- Τό μαγνητικό τύμπανο (Magnetic Drum).
- Οι μαγνητικές κάρτες (Magnetic Cards).
- Τά έντυπα μέ χαρακτήρες μαγνητικής μελάνης (Magnetic Ink Character Recognition ή M.I.C.R).

Οί φορείς αύτοί τών πληροφοριών τοποθετούνται πάνω σέ άνάλογες περιφερειακές μονάδες, πού τίς διακρίνομε έτσι σέ:

α) Μή μαγνητικές.

β) Μαγνητικές.

Μή μαγνητικές περιφερειακές μονάδες είναι:

- Ή άναγνωστική διατρήτων δελτίων (Card Reader).
- Ή άναγνωστική διάτρητης χαρτοταινίας (Papertape Reader).
- Ή έκτυπωτική (Printer).
- Ή διατρητική δελτίων (Card Punch).
- Ή διατρητική χαρτοταινίας (Papertape Punch).
- Ό όπτικός άναγνώστης (Optical Character Reader).
- Ή γραφομηχανή έπικοινωνίας (Console Typewriter).
- Ή όθόνη (Data Display Unit).
- Ή μονάδα σχεδιάσεως (Graph Plotter).

Μαγνητικές μονάδες είναι:

- Ή μονάδα μαγνητικής ταινίας (Magnetic Tape Unit).
- Ή μονάδα μαγνητικού δίσκου (Magnetic Disk Unit).
- Ή μονάδα μαγνητικού τυμπάνου (Magnetic Drum Unit).
- Ή μονάδα μαγνητικών καρτών (Data Cell).
- Ό άναγνώστης χαρακτήρων μαγνητικής μελάνης (Magnetic Ink Card Reader).

Από τίς μονάδες αυτές άλλες χρησιμοποιούνται άποκλειστικά καί μόνο ως μονάδες είσόδου, όπως ή άναγνωστική δελτίων, ή άναγνωστική χαρτοταινίας, ό όπτικός άναγνώστης καί ό άναγνώστης χαρακτήρων μαγνητικής μελάνης, άλλες δέ χρησιμοποιούνται μόνο ως μονάδες έξόδου, όπως ή έκτυπωτική, ή διατρητική δελτίων ή χαρτοταινίας, ή σχεδιαστική. Μονάδες μικτές είναι ή μονάδα μαγνητικής ταινίας, μαγνητικού δίσκου, μαγνητικού τυμπάνου, ή όθόνη καί ή γραφομηχανή έπικοινωνίας.

Παρακάτω θά έξετάσμε χωριστά κάθε φορέα πληροφοριών καί θά περιγράφομε τή μονάδα πάνω στήν όποία τοποθετείται.

3.4 Τό δελτίο (Card).

Ό πió γνωστός καί διαδεδομένος φορέας πληροφοριών είναι τό δελτίο. Ύπάρ-

χουν διάφορα είδη δελτίων, που περιγράφουμε στη συνέχεια.

α) Δελτίο 80 στηλών.

Είναι η πιο γνωστή μορφή δελτίου. Κατασκευάζεται από ανθεκτικό μή ηλεκτρωγυφό χαρτί και έχει διαστάσεις $7 \frac{3}{8}'' \times 3 \frac{1}{4}''$ και πάχος 0,007". Τά δεδομένα καταγράφονται επάνω του με τη μορφή μικρών ορθογωνίων τρυπών. Κάθε δελτίο χωρίζεται κατακόρυφα σε 80 στήλες και οριζόντια σε 12 γραμμές (σχ. 3.4α). Άρα το δελτίο αυτό διαθέτει 960 (80×12) συνολικά θέσεις διατρήσεως. Σε κάθε στήλη καταχωρίζεται ένας μόνο χαρακτήρας, δηλαδή σε ένα δελτίο μπορούν να καταχωρισθούν μέχρι 80 χαρακτήρες. Οι γραμμές 0 - 9 αποτελούν το λεγόμενο *ψηφιακό τμήμα* (Digit Portion), ενώ οι γραμμές 0, 11, 12, το *τμήμα ζώνης* (Zone Portion). Η γραμμή 0 είναι κοινή και στα δύο τμήματα.

Κάθε αριθμητικός χαρακτήρας παριστάνεται πάνω στο δελτίο με μία τρύπα στην αντίστοιχη πάντοτε γραμμή του ψηφιακού τμήματος π.χ. ο αριθμός 8 θα παριστάνεται με μία τρύπα στη γραμμή 8 κάποιας στήλης.

Κάθε αλφαβητικός χαρακτήρας παριστάνεται με δύο τρύπες, από τις οποίες η μία βρίσκεται στο ψηφιακό τμήμα και η άλλη στο τμήμα ζώνης π.χ. ο χαρακτήρας Α παριστάνεται με μία τρύπα στη γραμμή 1 του ψηφιακού τμήματος και με μία διάτρηση στη ζώνη 12 της ίδιας στήλης. Ο κώδικας αυτός, που ονομάζεται και κώδικας *Hollerith*, φαίνεται στο σχήμα (σχ. 3.4β).

Σε ποιές ακριβώς στήλες θα διατρηθούν οι διάφοροι χαρακτήρες-δεδομένα είναι θέμα σχεδιασμού του δελτίου.

Παράδειγμα.

Στο δελτίο του σχήματος 3.4γ βλέπομε καταχωρισμένα:

α) Στίς στήλες 1-6 τόν αριθμό 123661.

β) Στίς στήλες 11-17 τό όνομα ΒΑΛΑΚΟΣ.

γ) Στίς στήλες 24-33 τό όνομα ΑΝΑΣΤΑΣΙΟΣ.

Όλα αυτά υποτίθεται ότι αποτελούν πληροφορίες.

β) Δελτίο 90 στηλών.

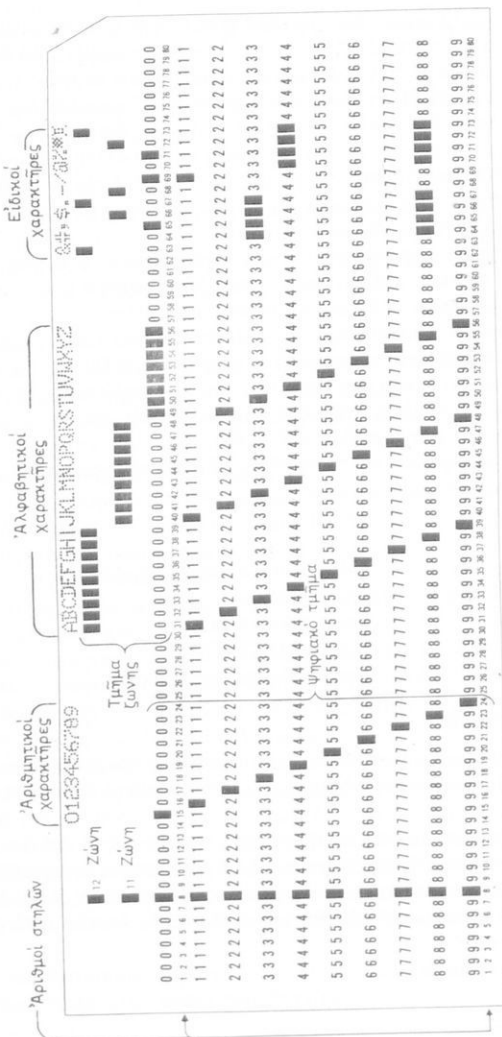
Τό δελτίο αυτό χωρίζεται σε 90 στήλες, που όμως δεν βρίσκονται στην ίδια ευθεία, αλλά σε δύο οριζόντια διαζώματα με 45 στήλες στο καθένα (1 - 45 και 46 - 90) (σχ. 3.4δ).

Κάθε διάζωμα υποδιαιρείται σε 6 γραμμές. Έπομένως τό δελτίο αυτό διαθέτει 540 (90×6) συνολικά θέσεις διατρήσεως. Σε κάθε στήλη καταχωρίζεται ένας χαρακτήρας, σύμφωνα με τόν κώδικα διατρήσεως του σχήματος 3.4δ. Δηλαδή συνολικά μπορούν να καταχωρισθούν 90 χαρακτήρες. Τό δελτίο αυτό δεν χρησιμοποιείται σήμερα πολύ.

¹ Έπειδή πολλά από τά φυσικά μεγέθη που θά αναφερόμαστε έκφράζονται συχνά διεθνώς στο Άγγλοσαξωνικό σύστημα μονάδων υπενθυμίζομε ότι:

1 πόδι (1 ft) = 12 ίντσες (12") = 30,48 cm

1 ίντσα (1" ή 1 inch) = 2,54 cm.



Σχ. 3.4α.

	ΧΑΡΑΚΤΗΡΕΣ	ΚΩΔΙΚΑΣ ΔΙΑΤΡΗΣΕΩΝ
ΑΡΙΘΜΗΤΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
ΑΛΦΑΒΗΤΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ	A	12.1
	B	12.2
	C	12.3
	D	12.4
	E	12.5
	F	12.6
	G	12.7
	H	12.8
	I	12.9
	J	11.1
	K	11.2
	L	11.3
	M	11.4
	N	11.5
	O	11.6
	P	11.7
	Q	11.8
	R	11.9
	S	0.2
	T	0.3
	U	0.4
	V	0.5
	W	0.6
	X	0.7
	Y	0.8
	Z	0.9
ΕΙΔΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ	+	12.6.8
	-	11
	*	11.4.8
	(	12.5.8
	)	11.5.8
	.	12.3.8
	/	0.3.8
	'	0.1
	~	5.8
	-	6.8

Σχ. 3.48

Ο κώδικας Hollerith για τη διάτρηση των χαρακτήρων σέ ένα 80στηλο δελτίο.

123661	BARCODE	ANASTASIEIDE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
--------	---------	--------------	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

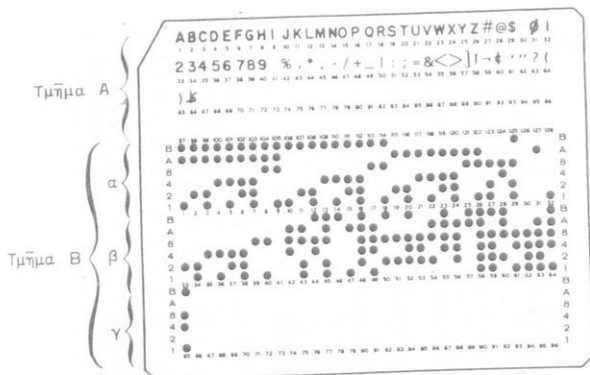
 $\Sigma x = 3.4y$

Παράδειγμα διατρυπημένου 80σπηλου δελτίου.

Q	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45		
12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	
34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34
56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56
78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z																																															
12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	
34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34
56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56
78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78
90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90

Σχ. 3.46.

Τό δελτίο των 90 στηλών.



Σχ. 3.4ε.

Τό δελτίο των 96 στηλών.

γ) Δελτίο 96 στηλών.

Τό δελτίο των 96 στηλών πρωτοπαρουσιάστηκε γύρω στά 1970. Είναι πιά κοντό, (2,63" x 3,25") από τό όγδοντάστηλο δελτίο. Χωρίζεται όριζόντια σέ δύο τμήματα (σχ. 3.4ε). Στό έπάνω τμήμα (Α) γίνεται ή έκτύπωση τών χαρακτήρων πού μπορούν νά διατρηθοϋν. Στό κάτω τμήμα (Β) γίνονται οί διατρήσεις. Τό τμήμα Α άποτελείται από 4 όριζόντιες ζώνες, όπου μπορούν νά έκτυπωθοϋν 4 σειρές από χαρακτήρες. Τό τμήμα Β τών διατρήσεων χωρίζεται όριζόντια σέ τρία διαζώματα: τό έπάνω (α) πού περιλαμβάνει τίς στήλες 1 - 32, τό μεσαίο (β) πού περιέχει τίς στήλες 33 - 64 καί τό κάτω (γ) μέ τίς στήλες 65 - 96 (σχ. 3.4ε).

Κάθε χαρακτήρας άπεικονίζεται μέ ένα συνδυασμό από διατρήσεις. 'Ο κώδικας διατρήσεως είναι ό 6ψήφιος άλφαριθμητικός (BA 84 21). 'Η ύπαρξη τρύπας άντιστοιχεί στό δυαδικό ψηφίο 1. 'Ανυπαρξία τρύπας ίσοδυναμεί μέ τό δυαδικό ψηφίο 0.

3.5 Μερικές χρήσιμες έννοιες.

Είναι χρήσιμο στό σημείο αυτό νά έπιχειρήσουμε νά αναλύσουμε μερικές έννοιες, πού θά τίς χρησιμοποιήσουμε συχνά πιά κάτω.

Πρώτη καί βασική είναι ή έννοια του *άρχειου* (File). Σάν *άρχειο* έννοοϋμε μιά όργανωμένη συλλογή από στοιχεία καί πληροφορίες. "Ας υποθέσουμε π.χ. ότι θέλομε νά προβοϋμε στή μισθοδοσία τών ύπαλλήλων μιάς έπιχειρήσεως. Στήν περίπτωση αύτή πρέπει νά συγκεντρώσουμε όρισμένα στοιχεία, πού άφορούν σέ κάθε ύπάλληλο καί πού έχουν σχέση μέ τή μισθοδοσία του, όπως τό όνοματεπώνυμό του, τίς βασικές άποδοχές του, τίς διάφορες κρατήσεις στό μισθό του, τά έπιδόματα πού του παρέχονται κ.ο.κ. Όλα αυτά τά στοιχεία ταξινομημένα κατάλληλα αποτελοϋν ένα άρχειο, τό άρχειο μισθοδοσίας.

Τό ίδιο θά συνέβαινε άν θέλαμε νά παρακολουθήσουμε τήν κίνηση τών ύλικών

πού βρίσκονται στις αποθήκες κάποιου έργου. Για κάθε είδος υλικού θά πρέπει νά συγκεντρώσουμε πληροφορίες πού ἀναφέρονται σ' αὐτό, ὅπως τὸν κωδικό του ἀριθμό, τὴν ποσότητα πού ὑπάρχει, τὴν τιμὴ μονάδας του κ.ο.κ. Καί πάλι ὅλα αὐτὰ τὰ στοιχεῖα, πού ἔχομε συγκεντρώσει γιὰ ὅλα τὰ εἶδη τῶν υλικῶν, πού ὑπάρχουν στὶς ἀποθήκες, ταξινομημένα κατ'ἀλλήλα, ἀποτελοῦν ἕνα ἄλλο ἀρχεῖο, τὸ ἀρχεῖο τῶν υλικῶν.

Ἀνάλογη διαδικασία μπορεῖ νά χρησιμοποιηθεῖ γιὰ τὴν ἐπεξεργασία πληροφοριῶν μέ τὸν ἠλεκτρονικὸ ὑπολογιστή.

Ὅλες οἱ πληροφορίες πού περιέχονται σέ ἕνα ἀρχεῖο καταχωρίζονται πᾶνω σέ κάποιον φορέα, ὅπως τὰ δελτία, τὴ μαγνητικὴ ταινία, τὸ δίσκο κλπ.

Ἄς δοῦμε τώρα πῶς εἶναι ὀργανωμένο ἕνα ἀρχεῖο. Τὰ στοιχεῖα ἢ οἱ πληροφορίες, πού ἀναφέρονται σέ ἕνα συγκεκριμένο πρόσωπο ἢ πράγμα (δηλ. τὰ στοιχεῖα πού ἀφοροῦν σέ ἕνα ὑπάλληλο, στὸ παράδειγμα τῆς μισθοδοσίας ἢ τὰ στοιχεῖα πού ἔχουν σχέση μέ ἕνα υλικό, ὅπως στὸ παράδειγμα τοῦ ἀρχείου υλικῶν) συγκροτοῦν ξεχωριστές ὁμάδες, πού τίς ὀνομάζομε *λογικὲς ἐνότητες* (*Records*). Μὲ ἄλλα λόγια τὸ ἀρχεῖο μισθοδοσίας θά περιέχει τόσες λογικὲς ἐνότητες ὅσοι εἶναι καί οἱ ὑπάλληλοι τῆς ἐπιχειρήσεως, ἢ τὸ ἀρχεῖο υλικῶν θά περιέχει τόσες λογικὲς ἐνότητες ὅσα καί τὰ εἶδη τῶν υλικῶν πού βρίσκονται στὶς ἀποθήκες¹.

Ἡ λογικὴ ἐνότητα σχεδιάζεται ἔτσι, ὥστε οἱ πληροφορίες πού θά ὑπάρχουν σ' αὐτὴ νά κατέχουν συγκεκριμένες περιοχές τῆς. Οἱ περιοχές αὐτές, πού μέσα στή κάθε μιά ὑπάρχει μιά μόνο μορφή πληροφορίας, ὀνομάζονται *πεδία* (*Fields*).

Στὸ παρακάτω σχῆμα 3.5α δίνεται ἕνα παράδειγμα σχεδιασμένης λογικῆς ἐνότητος ἐνός φανταστικοῦ ἀρχείου μισθοδοσίας, πού ἀποτελεῖται ἀπὸ δελτία.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	
[ΟΝΟΜΑΤΕΠΩΝΥΜΟ]										[ΟΝΟΜΑ ΠΑΤΡΟΣ]										[ΒΑΣ. ΜΙΣΘ]										[ΚΡΑΤ.]										[ΕΠΙΔ.]										
ΑΡΙΘ.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
ΜΗΤ.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
ΥΠΑΛ.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	

Σχ. 3.5α.

Παράδειγμα σχεδιασμένης λογικῆς ἐνότητος ἐνός φανταστικοῦ ἀρχείου μισθοδοσίας.

¹ Ἐδὼ ἔχομε ἀπλοποιήσει τίς ἔννοιες τῶν λογικῶν ἐνοτήτων (*Records*) γιὰ νά γίνουν κατανοητές. Στὴν πραγματικότητα ἀντιμετωπίζεται τὸ ἐνδεχόμενο νά ὑπάρχουν δύο ἢ περισσότερα *Records* γιὰ κάθε πρόσωπο ἢ πράγμα (κυρίως στὶς περιπτώσεις ἀρχείων, πού ἀποτελοῦνται ἀπὸ δελτία).

Όπως βλέπομε στο παράδειγμα αυτό, η λογική ένότητα αποτελείται από 6 πεδία, πού τό κάθε ένα κατέχει συγκεκριμένες θέσεις (π.χ. τό πεδίο αριθμός μητρώου κατέχει τίς θέσεις 1 - 5 του δελτίου κ.ο.κ.).

Γιά κάθε υπάλληλο υπάρχει μία λογική ένότητα του τύπου αυτού. Όλες οι λογικές ένότητες ενός αρχείου πρέπει νά έχουν τήν ίδια μορφή καί νά περιέχουν τά ίδια πεδία στίς ίδιες σχετικές θέσεις. Φυσικά τό περιεχόμενό τους θα αλλάζει από υπάλληλο σέ υπάλληλο δηλαδή από λογική ένότητα σέ λογική ένότητα.

Κάθε πεδίο έχει ένα συγκεκριμένο μήκος, δηλαδή μπορεί νά χωρέσει σέ αυτό ορισμένο πλήθος από χαρακτήρες. Τό πρώτο πεδίο στο παράδειγμά μας είναι τό πεδίο του αριθμού μητρώου του υπαλλήλου καί έχει μήκος 5 χαρακτήρες. Μέσα στο πεδίο αυτό καταχωρίζεται κάθε φορά ένας πενταψήφιος αριθμός πού είναι ο αριθμός μητρώου του υπαλλήλου. Στο δεύτερο πεδίο, πού έχει μήκος 25 χαρακτήρες, καταχωρίζεται τό ονοματεπώνυμό του κ.ο.κ.

Σέ ορισμένα πεδία καταχωρίζομε μόνο αριθμητικά δεδομένα καί γι' αυτό ονομάζονται *αριθμητικά πεδία*. Σέ άλλα πάλι πεδία καταχωρίζομε μόνο αλφαβητικούς χαρακτήρες καί τά ονομάζομε *άλφαβητικά πεδία*. Στην περίπτωση πού μέσα σ' ένα πεδίο καταχωρίζομε καί αλφαβητικούς καί αριθμητικούς χαρακτήρες, τό πεδίο ονομάζεται *άλφαριθμητικό* (Alphanumeric Field). Έτσι στο παραπάνω παράδειγμα, τά πεδία: αριθμός μητρώου, βασικός μισθός, κρατήσεις καί επιδόματα, είναι αριθμητικά, ενώ τά πεδία: ονοματεπώνυμο καί όνομα πατρός είναι αλφαβητικά.

Στό σχήμα 3.5β βλέπομε μερικές από τίς λογικές ένότητες πού ανήκουν στο άρχείο μισθοδοσίας του παραδείγματός μας.

00142	ΠΑΠΑΔΕ	ΧΡΗΣΤΙΔΕ	ΕΛΤΙΝΙΔΕ	17555	3041	1975
00066	ΘΕΟΔΩΡΟΥ	ΚΑΚΑΔΕ	ΝΙΚΟΛΑΟΥ	15678	2954	1967
00034	ΣΑΡΓΙΔΕ	ΑΓΓΕΛΙΔΕ	ΓΕΩΡΓΙΟΥ	16844	3068	1977
00021	ΒΟΥΚΙΔΕ	ΜΙΝΑΝΗ	ΙΩΑΝΝΙΔΕ	14568	2931	1857
00001	ΑΝΤΩΝΙΔΕ	ΓΕΩΡΓΙΟΥ	ΓΕΩΡΓΙΟΥ	11245	1850	1806

Σχ. 3.5β.

Μερικές από τίς λογικές ένότητες πού ανήκουν στο άρχείο μισθοδοσίας του παραδείγματός μας.

Έξ αιτίας του σταθερού μήκους των πεδίων σέ όλες τις λογικές ενότητες του άρχείου, υπάρχει πιθανότητα νά μή καταλαμβάνονται, σέ μερικές από αυτές όλες οι θέσεις τους από χαρακτήρες όπως κάλλιστα μπορούμε νά διαπιστώσουμε στο παράδειγμα του σχήματος 3.5β.

Σέ όρισμένες μορφές άρχείων, όπως αυτά πού βρίσκονται πάνω σέ μαγνητική ταινία, δίσκο ή τύμπανο, οι λογικές ενότητές τους μπορούν νά ομαδοποιηθούν καί νά συγκροτήσουν μεγαλύτερες ενότητες, πού ονομάζονται *φυσικές ενότητες* (Blocks). Στην περίπτωση των φυσικών ενότητων χρησιμοποιούμε την έννοια του συντελεστή ή παράγοντα ομαδοποίησης (Blocking Factor ή B.F), γιά νά δείξουμε τό πλήθος των λογικών ενότητων πού περιέχονται σέ αυτές.

3.6 Κλασσικές μηχανές.

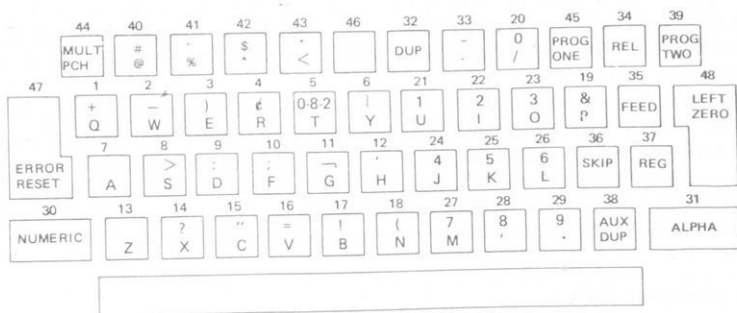
Ός κλασσικές μηχανές χαρακτηρίζουμε τις μηχανές, πού διατρυπών ή έπεξεργάζονται διάτρητα δελτία, αλλά πού δέν συνδέονται μέ τό κύριο συγκρότημα του υπολογιστή.¹ Ο ρόλος τους είναι καθαρά βοηθητικός καί προπαρασκευαστικός.

α) Διατρητικές μηχανές (Key punching machines).

Μέ τή βοήθεια των μηχανών αυτών τά δεδομένα, πού έχουν συγκεντρωθεί καί είναι γραμμένα πάνω σέ ειδικά σχεδιασμένα έντυπα, τά *παραστατικά*, διατρυπώνται σέ δελτία. Διαθέτουν πληκτρολόγιο όμοιο μέ τό πληκτρολόγιο μιās κοινής ηλεκτρικής γραφομηχανής (σχ. 3.6α) καί ή ταχύτητα διατρήσεως εξαρτάται από τις ικανότητες του χειριστή.



¹ Μέ τόν όρο υπολογιστής θά έννοούμε από έδω καί έμπρός τόν ψηφιακό ήλεκτρονικό υπολογιστή.



Σχ. 3.6α.

Τό πληκτρολόγιο μιάς διατρητικής μηχανής.

Ἡ μηχανή λειτουργεῖ μέ τόν ἐξῆς τρόπο. Τοποθετοῦνται τά λευκά ἀδιάρτητα δελτία στήν ὑποδοχή 1 (Card Hoper) [σχ. 3.6β(α)]. Μέ τήν πίεση ἑνός πλήκτρου, ἕνα δελτίο ὁδηγεῖται στό σταθμό διατρήσεως 2 (Punching station). Μέ κάθε νέα πίεση ἑνός πλήκτρου χαρακτήρα, ὁ ἀντίστοιχος χαρακτήρας διατυπῶνται σέ μία στήλη, ἐνῶ τό δελτίο προχωρεῖ μιά θέση πρός τά ἔμπρός, ἔτοιμο γιά νά διατυπῶνται ὁ ἐπόμενος χαρακτήρας στήν ἐπόμενη στήλη κ.ο.κ. Ὅταν διατυπῶνται ὅλοι οἱ χαρακτήρες, πού προβλέπεται νά ἀπεικονισθοῦν σ' αὐτό, τό δελτίο ὁδηγεῖται στό σταθμό ἀναγνώσεως 3 (Reading station). Ὁ σκοπός πού ἐξυπηρετεῖ ὁ σταθμός ἀναγνώσεως, εἶναι νά διαβασθοῦν τά δεδομένα, πού μόλις πρὶν διατρήθηκαν, ὥστε νά ἀποτυπωθοῦν, ἂν τό ἐπιθυμοῦμε, ὅλα ἡ μερικά, στό δελτίο πού ἀκολουθεῖ, χωρίς νά χρειάζεται νά ἐπαναληφθεῖ ἡ ἴδια πληκτρολόγηση.

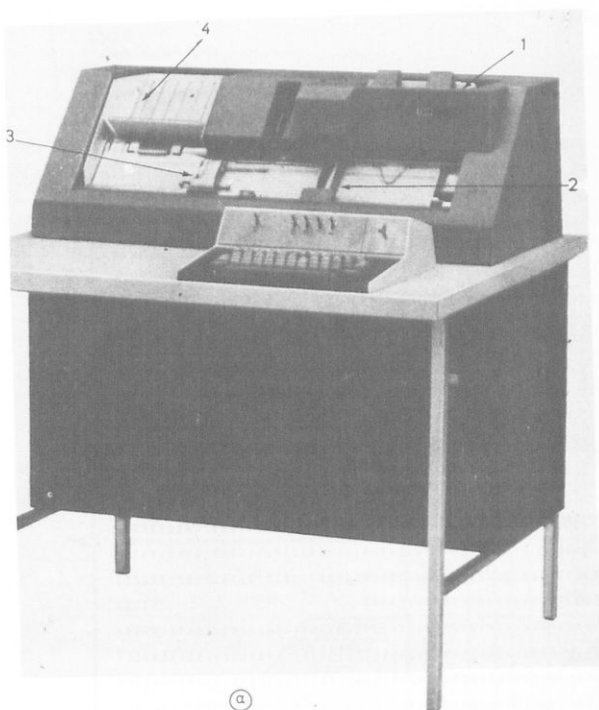
Μετά τό σταθμό ἀναγνώσεως, τό δελτίο ὁδηγεῖται στή χοάνη ἐξόδου (Card stacker) 4.

Μερικοί τύποι διατρητικῶν μηχανῶν διαθέτουν *μνήμη*, ὅπου ἀποθηκεύονται προσωρινά ὅλοι οἱ χαρακτήρες, πού πληκτρολογοῦνται, καί πού πρόκειται νά διατυπῶνται σ' ἕνα δελτίο. Μέ τήν πίεση κατόπιν ἑνός ἄλλου πλήκτρου ὅλοι οἱ χαρακτήρες διατυπῶνται ἀμέσως καί τό δελτίο ὁδηγεῖται στή χοάνη ἐξόδου.

β) Ἐπαληθευτική μηχανή (Verifier ἢ Verifying machine).

Στά δελτία πού διατρήθηκαν ἀπό τή διατρητική μηχανή εἶναι ἐνδεχόμενο νά ἔχουν γίνεи σφάλματα. Τά σφάλματα αὐτά μποροῦν νά ἐντοπισθοῦν μέ τή βοήθεια τῆς ἐπαληθευτικῆς μηχανῆς.

Ἡ μηχανή αὕτη διαθέτει ὁμοιο πληκτρολόγιο μέ τή διατρητική, δέν κάνει ὅμως διατρήσεις. Ὁ χειριστής ξανακτυπᾷ στό πληκτρολόγιο τούς χαρακτήρες, πού πρέπει νά ἔχουν διατρηθεῖ στό δελτίο. Σέ περίπτωση πού ἔχει διατρηθεῖ σέ κάποια στήλη λανθασμένος χαρακτήρας, ἡ μηχανή τόν ἐπισημαίνει δημιουργώντας μιά ἐγκοπὴ στό δελτίο καί πάνω ἀπό τή στήλη, ὅπου ἀνιχνεύθηκε ὁ λάθος χαρακτήρας [σχ. 3.6γ(α)]. Ἄν ὅλο τό δελτίο ἔχει διατρηθεῖ σωστά, τότε ἡ μηχανή δημιουργεῖ μιά ἐγκοπὴ κοντά στήν ἐπάνω δεξιά γωνία τοῦ δελτίου [σχ. 3.6γ(β)].



Σχ. 3.6β.

α) 'Η διατρητική μηχανή. β) 'Ο μηχανισμός διατρυπήσεως.

γ) Μεταφραστική μηχανή (Interpreter).

Ἡ μηχανή αὐτή ἐκτυπώνει στό ἐπάνω λευκό περιθώριο τοῦ δελτίου τοὺς χαρακτηριστές πού ἔχουν διατρηθεῖ σ' αὐτό.

Πολλές μηχανές μπορεῖ νά εἶναι ταυτόχρονα διατρητικές καί μεταφραστικές, ἄλλες δέ μποροῦν νά χρησιμοποιηθοῦν καί ὡς ἐπαληθευτικές.

δ) Διαλογική μηχανή (Card sorter).

Πολλές φορές τά δελτία μέ τά δεδομένα πρέπει νά τά ταξινομήσουμε προτοῦ εἰσαχθοῦν γιά ἐπεξεργασία στόν ὑπολογιστή, νά τά βάλομε δηλαδή σέ κάποια καθορισμένη σειρά. Ἡ ταξινόμηση αὐτή τῶν δελτίων ὀνομάζεται *διαλογή* (Sort) καί μπορεῖ νά εἶναι ἀριθμητική ἢ ἀλφαβητική καί ἀκόμα αὐξοῦσα ἢ φθίνουσα, γίνεται δέ μέ τή βοήθεια τῶν διαλογικῶν μηχανῶν (Sorters) [σχ. 3.6δ(α)]. Ἡ μηχανή αὐτή ταξινομεῖ τά δελτία σύμφωνα μέ τό περιεχόμενο ἑνός ἢ περισσοτέρων πεδίων (Sort Key Fields). Ἔτσι ἂν π.χ. ὑπάρχει ἕνα πεδίο σέ κάθε δελτίο, ὅπου καταχωρίζεται κάποιο ὄνομα, μέ τή διαλογική μηχανή μπορούμε νά τοποθετήσουμε τά δελτία

Λάθος διατυπωμένοι
χαρακτήρες

(α)

(β)

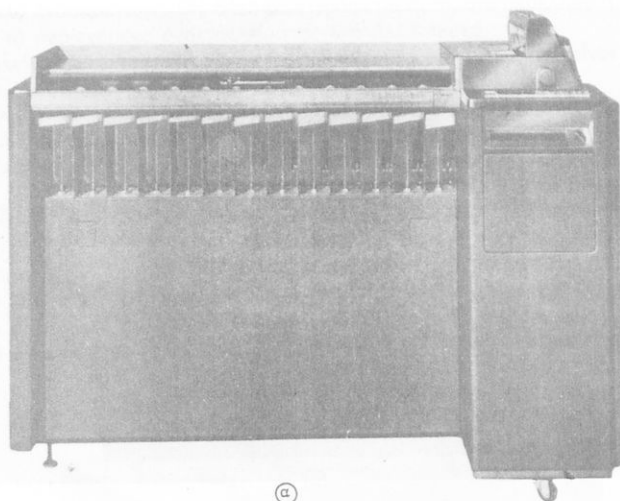
Σχ. 3.6γ.

(α) Λάθος διάτρηση του δελτίου (β) Σωστή διάτρηση του δελτίου.

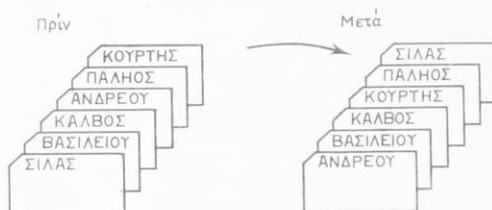
έτσι, ώστε να εμφανίζονται κατ' αλφαβητική σειρά ονόματος [σχ. 3.6δ(β)]. Τό ίδιο ισχύει καί για τους αριθμούς [σχ. 3.6δ(γ)].

ε) Συζευκτική μηχανή (Collator).

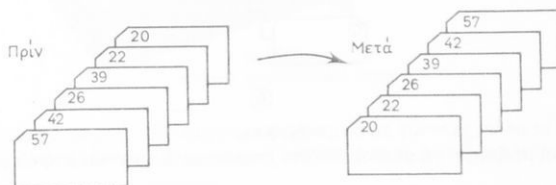
“Ας υποθέσουμε ότι μας έχουν δοθεί δύο ομάδες δελτίων, που κάθε μία έχει ταξινομηθεί ως προς τό ίδιο πεδίο μέ τόν ίδιο τρόπο (αύξουσα ή φθίνουσα σειρά). Μέ τή βοήθεια τής μηχανής αυτής [σχ. 3.6ε(α)] μπορούμε νά συνενώσουμε αυτές τīs δύο ομάδες σέ μία, όπου τά δελτία θά παρουσιάζονται καί πάλι ταξινομημένα



α



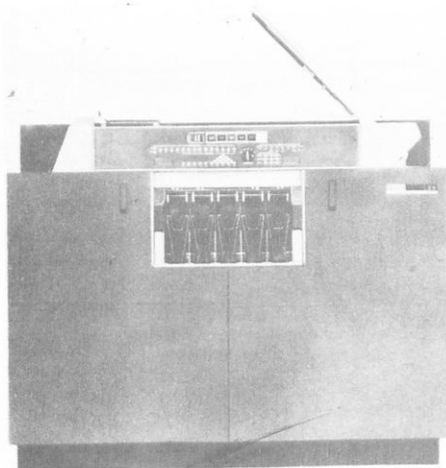
β



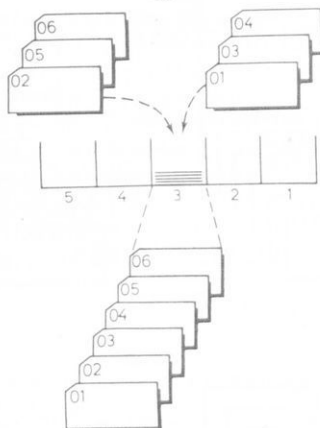
γ

Σχ. 3.66.

(α) Ἡ διαλογικὴ μηχανή, (β) Ἀλφαβητικὴ ταξινόμηση κατὰ αὐξοῦσα σειρά (γ) Ἀριθμητικὴ ταξινόμηση κατὰ αὐξοῦσα σειρά.



(α)



(β)

Σχ. 3.6ε.

(α) Ἡ Συζευκτική μηχανή, (β) Ἕνα παράδειγμα ἀριθμητικῆς συζεύξεως.

ὡς πρὸς τὸ ἴδιο πεδίο-κλειδί καὶ μέ τὸν ἴδιο τρόπο ὅπως καὶ οἱ ἀρχικὲς [σχ. 3.6ε(β)].

στ) Ἀναπαραγωγικὴ μηχανή (Reproducer).

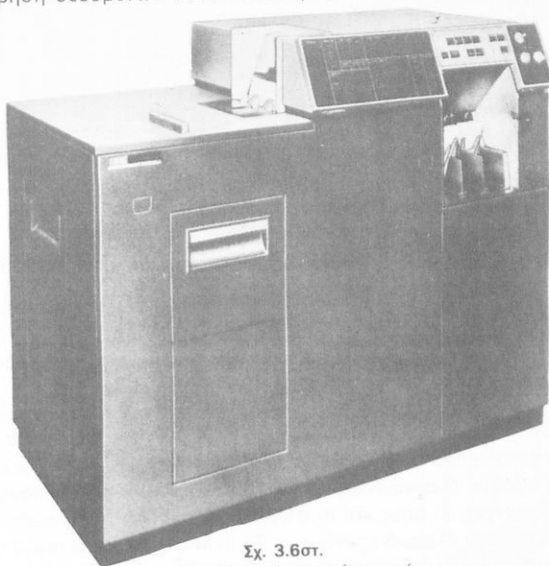
Ἡ μηχανή αὐτή ἐκτελεῖ διάφορες ἐργασίες ὅπως:

- Άπλή αναπαραγωγή. Με αυτήν αντιγράφεται αυτόματα ολόκληρο ή μέρος από το περιεχόμενο ενός δελτίου σε ένα άλλο λευκό.
- Ομαδική διάτρηση. Με αυτήν διατρύπεται ένα δεδομένο, που είναι κοινό σε μία ομάδα δελτίων, σε όλα τα δελτία της ομάδας.
- Αναγνώριση - επισημάνσεως με ειδικό μολύβι. Γίνεται διάτρηση δελτίων με βάση σύμβολα που είναι καταχωρισμένα στα ίδια δελτία με ειδικό μολύβι (Mark sensing).

ζ) Άπλή υπολογιστική μηχανή (Card calculator).

Η μηχανή αυτή επεξεργάζεται δεδομένα, που έχουν διατρηθεί επάνω σε δελτία (σχ. 3.6στ). Οι κυριότερες λειτουργίες της είναι:

- Η εκτέλεση των τεσσάρων αριθμητικών πράξεων.
- Οι άπλές συγκρίσεις δεδομένων.
- Η διάτρηση δεδομένων και αποτελεσμάτων σε λευκά δελτία.



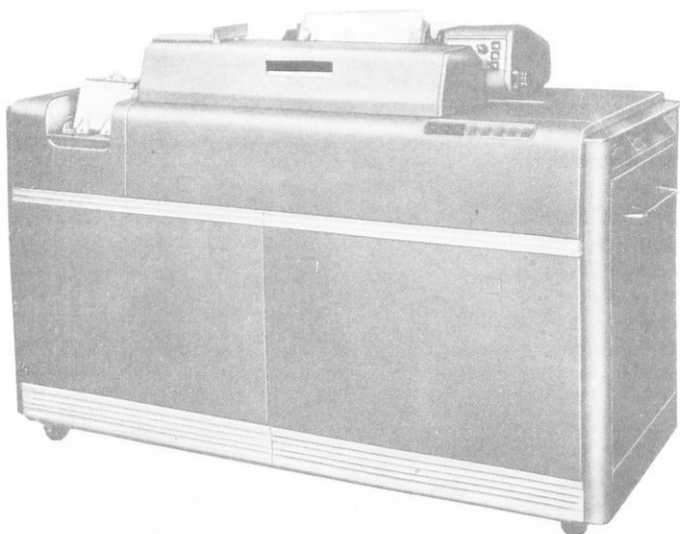
Σχ. 3.6στ.

Η άπλή υπολογιστική μηχανή.

Η υπολογιστική μηχανή δεν δέχεται πρόγραμμα με εντολές, αλλά οι λειτουργίες της κατευθύνονται από ένα ειδικό πίνακα, στον οποίο έχει γίνει προηγουμένως κατάλληλη συνδεσμολογία καλωδίων.

η) Πινακοποιητική ή Λογιστική μηχανή (Tabulator or Accounting machine).

Η μηχανή αυτή επεξεργάζεται δεδομένα που έχουν διατρηθεί επάνω σε δελτία, κάνει αριθμητικές πράξεις, συγκρίνει δεδομένα, βρίσκει σύνολα και εκτυπώνει καταστάσεις (ή μόνη από τις κλασσικές μηχανές που μπορεί να τυπώσει) (σχ. 3.6ζ).



Σχ. 3.6ζ.

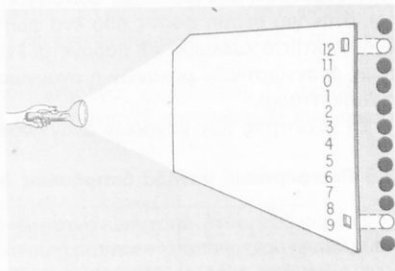
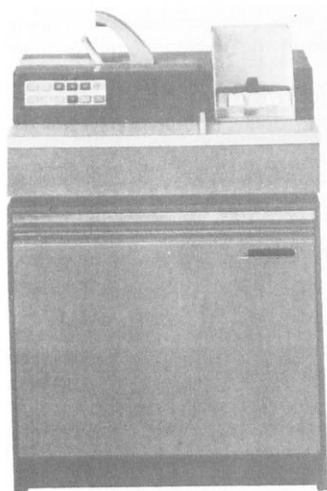
Ἡ πινακοποιητική μηχανή.

3.7 Περιφερειακή μονάδα ἀναγνώσεως διατρήτων δελτίων (Card reader).

Τά δεδομένα πού ἔχουν διατεθεῖ ἐπάνω σέ δελτία εἰσάγονται στὸν ὑπολογιστὴ μὲ τὴ βοήθεια τῆς μονάδας ἀναγνώσεως διατρήτων δελτίων (σχ. 3.7α).

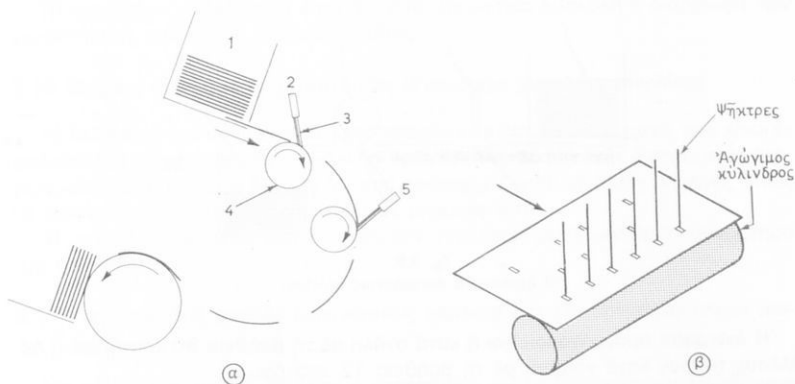
Παρακάτω θά περιγράψουμε τὸν τρόπο ἀναγνώσεως σέ μιά μονάδα σχεδιασμένη γιὰ νά διαβάζουμε μ' αὐτὴν 80σσηλα δελτία.

- Τά διατρημένα δελτία τοποθετοῦνται στὴν ὑποδοχὴ 1 τῆς ἀναγνωστικῆς (σχ. 3.7β).
- Μὲ μία ἐντολὴ ἀπὸ τὴ μονάδα ἐλέγχου, ἕνα-ἕνα τὰ διάτρητα δελτία ὁδηγοῦνται στοὺς σταθμοὺς ἀναγνώσεως 2. Ἐκεῖ βρίσκονται σέ σειρά ἐπάνω ἀπὸ τὸ δελτίο 80 ψῆκτες 3, ὅσες καὶ οἱ στήλες τοῦ δελτίου (ἀνάγνωση κατὰ στήλη), πού ἀκουμποῦν ἐλαφρὰ ἐπάνω του. Τὸ δελτίο ὀλισθαίνει πάνω σ' ἕνα ἀγώγιμο κύλινδρο 4 πού βρίσκεται κάτω ἀπὸ τὶς ψῆκτες. Κάθε ψῆκτρα ἐλέγχει τὴν περιοχὴ μιᾶς στήλης [σχ. 3.7β(β)]. Καθὼς τὸ δελτίο κινεῖται, ὅπου ὑπάρχει διάτρηση σέ μιά στήλη, ἡ ἀντίστοιχη ψῆκτρα ἔρχεται σ' ἐπαφὴ μὲ τὸν ἀγώγιμο κύλινδρο 4. Σέ κάθε τέτοια ἐπαφὴ κλείνει ἕνα κύκλωμα καὶ παράγεται ἕνας μικρῆς διάρκειας ἠλεκτρικὸς παλμός, πού ἰσοδυναμεῖ μὲ τὸ δυαδικὸ ψηφίο 1. Ὅταν δὲν ἀποκαθίσταται ἐπαφὴ καὶ ἐπομένως δὲν παράγεται ἠλεκτρικὸς παλμός, τότε ἔχομε τὸ δυαδικὸ ψηφίο 0.
- Οἱ παλμοὶ πού παράγονται μεταφέρονται στὴν κεντρικὴ μνήμη, ὅπου ἀποθηκεύονται. Οἱ ἀναγνωστικὲς εἶναι ἐφοδιασμένες, τὶς περισσότερες φορές, καὶ μ' ἕνα δεῦτερο σταθμὸ ἀναγνώσεως 5, ὅπου ἐλέγχεται τὸ σωστὸ διάβασμα τοῦ δελτίου ἀπὸ τὸν πρῶτο σταθμὸ.



Σχ. 3.7α.

Μονάδα ανάγνωσης διατρήτων δελτίων (αριστερά) καί σχηματικό ανάλογο του τρόπου ανάγνωσης (δεξιά).



Σχ. 3.7β.

(α) Ἡ διαδικασία ἀνάγνωσης. (β) Ὁ μηχανισμός ἀνιχνεύσεως τῶν διατρήσεων μέ ψηκτρες.

Ἐπειδὴ τὰ δεδομένα πού βρίσκονται ἐπάνω στό δελτίο εἶναι κωδικοποιημένα σύμφωνα μέ τόν κώδικα πού χρησιμοποιεῖται, ἐνῶ στή μνήμη θά πρέπει νά ἀποθηκευθοῦν μέ τή μορφή κάποιου ἄλλου κώδικα (δψήφιο ἀλφαριθμητικό, EBCDIC

κλπ.) υπάρχουν ειδικές διατάξεις (μερικές φορές είναι ένσωματωμένες στη μονάδα), που εκτελούν τη μετατροπή αυτή κατά τη φάση της μεταφοράς των δεδομένων στην κεντρική μνήμη.

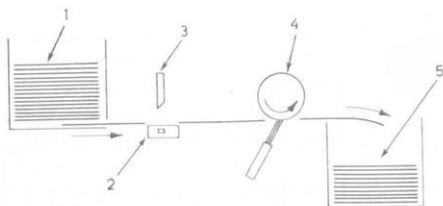
Σε άλλους τύπους αναγνώστικων ο μηχανισμός άνιχνεύσεως με ψήκτρες έχει αντικατασταθεί από μία σειρά φωτοηλεκτρικών κυττάρων. Καθώς το δελτίο κινείται, όταν μία δέσμη φωτός από ένα φωτοκύτταρο συναντήσει μία τρύπα, κλείνει ένα ηλεκτρικό κύκλωμα και παράγεται ένας βραχύχρονος παλμός. Σε μερικούς τύπους αναγνώστικων μηχανών η ανάγνωση γίνεται κατά ζώνη (μέ 12 ψήκτρες ή φωτοκύτταρα).

Οι ταχύτητες των μονάδων αναγνώσεως είναι 300 - 2000 δελτία τό λεπτό.

3.8 Περιφερειακή μονάδα διατήσεως δελτίων (Card punch).

Στη μονάδα αυτή αποτυπώνονται με διάτρηση πάνω σε δελτία πληροφορίες, που μεταφέρονται από την κεντρική μονάδα του υπολογιστή. Σε αρκετές περιπτώσεις η μονάδα αυτή είναι ένσωματωμένη στη μονάδα αναγνώσεως.

Η διαδικασία της διατήσεως είναι ίδια με τη διαδικασία της αναγνώσεως. Ένα - ένα τα λευκά (άδιάτρητα) δελτία, που είναι τοποθετημένα στην υποδοχή 1 (σχ. 3.8) περνούν από το σταθμό διατήσεως 2. Εκεί υπάρχει μία σειρά από μαχαιρίδια-κοπτήρες 3. Τη στιγμή που μερικά από αυτά ενεργοποιούνται, προκαλούν διατρήσεις. Στη συνέχεια το διατρημένο δελτίο οδηγείται στο σταθμό αναγνώσεως-ελέγχου 4, όπου με μία σειρά από αγώγιμες ψήκτρες διαβάζονται τα δελτία, για να διαπιστωθεί αν έχει γίνει η σωστή διάτρηση. Μετά τον έλεγχο τα δελτία οδηγούνται στη χοάνη εξόδου 5.



Σχ. 3.8.

Η διαδικασία διατήσεως δελτίου.

Η διάτρηση πραγματοποιείται ή κατά στήλη με τη βοήθεια 80 κοπτήρων ή σε άλλους τύπους κατά γραμμή με τη βοήθεια 12 κοπτήρων.

Η μονάδα διατήσεως έχει ταχύτητα 100 - 600 δελτία τό λεπτό.

Σε μερικούς τύπους η μονάδα αυτή έχει ένσωματωθεί στη μονάδα αναγνώσεως.

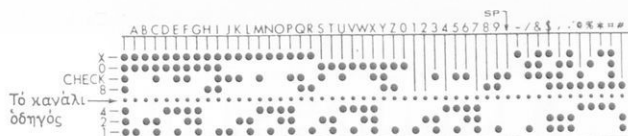
3.9 Η χαρτοταινία (Papertape).

Η χαρτοταινία είναι μία μεγάλου μήκους (200 - 1000 ft) λωρίδα από μη ηλεκτραγωγό χαρτί πλάτους περίπου 1 ίντσας.

Χωρίζεται σέ 8 τό μήκος της σέ 5 ως 8 παράλληλες ζώνες τά κανάλια (tracks or channels). Στό μέσο περίπου τοῦ πλάτους της ὑπάρχει μιὰ ἐπί πλέον ζώνη, πού φέρει συνεχή σειρά ἀπό μικρές τρύπες καί χρησιμεύει γιά τήν κίνηση τῆς ταινίας.

Ὁ ἀριθμός τῶν καναλιῶν ἐξαρτᾶται ἀπό τόν κώδικα, πού χρησιμοποιεῖται γιά τή παράσταση τῶν δεδομένων. Ἐτσι π.χ. στίς χαρτοταινίες μέ 8 κανάλια, οἱ χαρακτῆρες ἀπεικονίζονται στόν 6ψήφιο ἀλφαριθμητικό κώδικα, ὅπου στά 6 ψήφια τοῦ κώδικα προστίθεται τό ψηφίο ἰσοτιμίας (x) καί ἕνα ἀκόμα ψηφίο (EL) πού συμβολίζει τό τέλος γραμμῆς.

Σέ κάθε στήλη τῆς χαρτοταινίας ἀπεικονίζεται ἕνας χαρακτήρας. Τό σχῆμα 3.9 ἀπεικονίζει τόν κώδικα διατρήσεων μιᾶς χαρτοταινίας 8 καναλιῶν. Ἡ πυκνότητα καταγραφῆς εἶναι περίπου 10 χαρακτῆρες ἀνά ἴντσα (10ch/1").



Σχ. 3.9.

Ἡ κωδικοποίηση τῶν χαρακτήρων στήν χαρτοταινία τῶν 8 καναλιῶν.

Τά πλεονεκτήματα ἀπό τή χρήση τῆς χαρτοταινίας εἶναι ὅτι δέν κοστίζει ἀκριβὰ καί μποροῦν πολλά δεδομένα νά ἀποθηκευθοῦν εὐκόλα καί νά καταλάβουν μικρό σχετικό ὄγκο.

Τό κυριώτερο μειονέκτημα εἶναι ὅτι εἶναι ἐξαιρετικά δύσκολη ἡ διόρθωση τῶν χαρακτήρων, πού ἔχουν διατρηθεῖ λάθος.

3.10 Μηχανή διατρήσεως χαρτοταινίας (Papertape punching machine).

Τά δεδομένα διατρυπῶνται στή χαρτοταινία εἴτε ἀπό εἰδική μηχανή, πού εἶναι ἐφοδιασμένη μέ πληκτρολόγιο ἀνάλογο πρός ἐκεῖνο τῆς μηχανῆς διατρήσεως δελτίων, εἴτε μέ τή βοήθεια συσκευῶν, πού προσαρμῶνται σέ ἄλλες μηχανές, ὅπως τό τηλέτυπο, οἱ λογιστικές καί ταμιακές μηχανές κλπ.

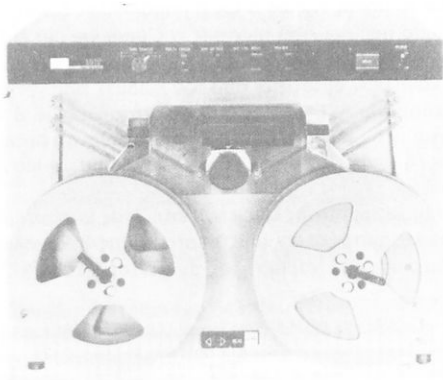
Ἡ μηχανή διατρήσεως χαρτοταινίας δέν συνδέεται μέ τό κεντρικό συγκρότημα τοῦ ὑπολογιστή.

3.11 Περιφερειακή μονάδα ἀναγνώσεως χαρτοταινίας (Punched papertape reader).

Μέ τή μονάδα αὐτή διαβάζεται ἡ χαρτοταινία καί τά δεδομένα μεταβιβάζονται στόν ὑπολογιστή. Ἡ ταχύτητά της εἶναι 200 - 1200 χαρακτῆρες τό δευτερόλεπτο (σχ. 3.11).

3.12 Περιφερειακή μονάδα διατρήσεως χαρτοταινίας (Papertape punch unit).

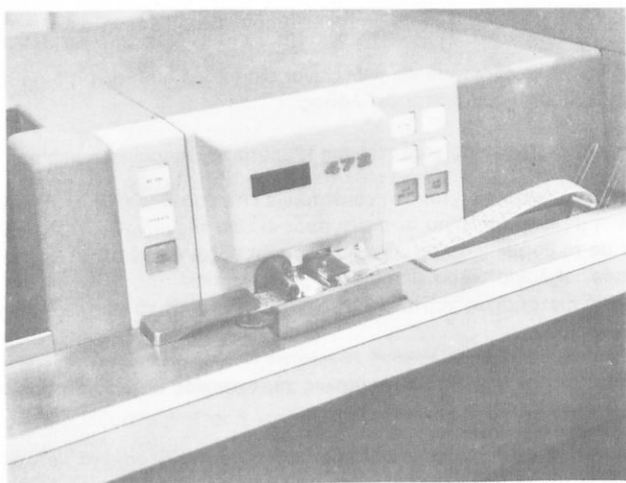
Στή μονάδα αὐτή διατρυπῶνται πάνω σέ χαρτοταινία πληροφορίες, πού προέρχονται ἀπό τόν ὑπολογιστή (σχ. 3.12). Ἡ ταχύτητα διατρήσεως εἶναι περίπου 100



Σχ. 3.11.

Μονάδα αναγνώσεως διάτρητης χαρτοταινίας.

χαρακτήρες τό δευτερόλεπτο. Είναι μάλλον άργή μονάδα καί γι' αυτό δέν χρησιμοποιείται συχνά. Σέ μερικούς τύπους ή διατρητική χαρτοταινίας είναι ένσωματωμένη στην αναγνώστκή.

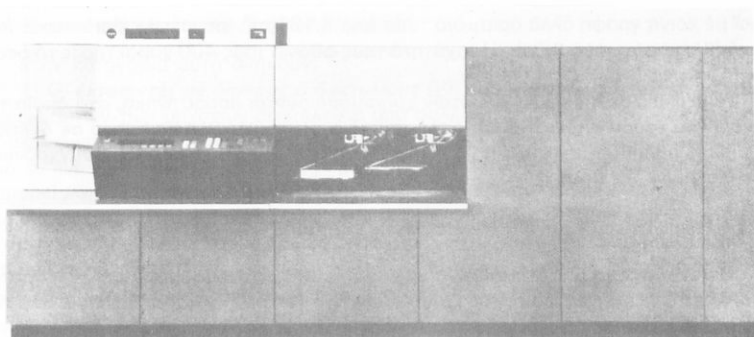


Σχ. 3.12.

Μονάδα διατρήσεως χαρτοταινίας.

3.13 Όπτικοί αναγνώστες (Optical characters readers) (σχ. 3.13α).

Είναι μονάδες ικανές νά διαβάζουν τούς χαρακτήρες-δεδομένα πού έχουν γρα-



Σχ. 3.13α.
Όπτικός ανάγνωσης.

ABCDEFGHIJKLM
 NOPQRSTUVWXYZ
 0123456789
 α γ : ; = + / \$ * ^ & |
 ' - { } % ? [\] ^ _
 Ü Ñ Ä Ø Ö Å Æ £ ¥

06-7346-2401 BOLT

	18	19	20	21	22	23	24	25	26	27
0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9

Σχ. 3.13β.

(α) Ένας τύπος χαρακτήρων που μπορούν να αναγνωρισθούν όπτικά. (β) Δεδομένα σημειωμένα ως απλά σημάδια σε ειδικές θέσεις επάνω σε δελτίο.

α) Έκτυπωτές με έπαφή (Impact printers).

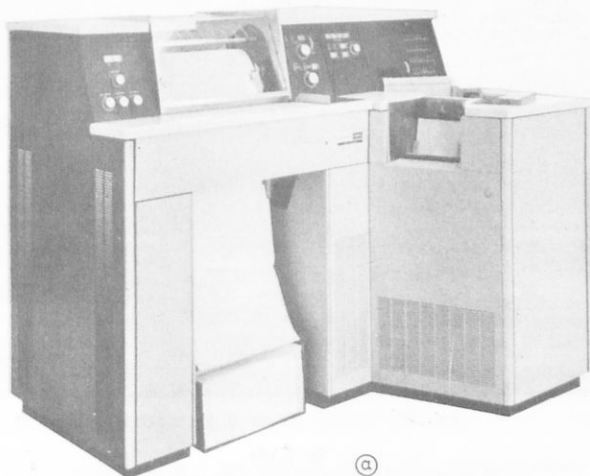
Στήν κατηγορία αυτή ανήκουν:

1) **Οι έκτυπωτές με δίσκους ή δακτυλίους (Wheel printers).** Τό τμήμα έκτυπώσεως αυτών των μηχανών αποτελείται από πολλούς (συνήθως 120) λεπτούς κυκλικούς μεταλλικούς δακτυλίους, πού είναι προσαρμοσμένοι ό ένας δίπλα στόν άλλο σ' ένα κοινό όριζόντιο άξονα· ό άξονας αυτός περνά από τά κέντρα τους καί μπορεί νά τούς περιστρέφει. Ό κάθε δίσκος φέρει στήν περιφέρειά του ανάγλυφους τούς αριθμητικούς, άλφαβητικούς καί είδικούς χαρακτήρες (σχ. 3.14α). Τό χαρτί, όπου πρόκειται νά έκτυπωθούν οί χαρακτήρες, ξετυλίγεται μπροστά από τό σύστημα των δακτυλίων. Ή ταχύτητά τους είναι σχετικά μικρή (150 πλήρεις γραμμές των 120 χαρακτήρων τό λεπτό).

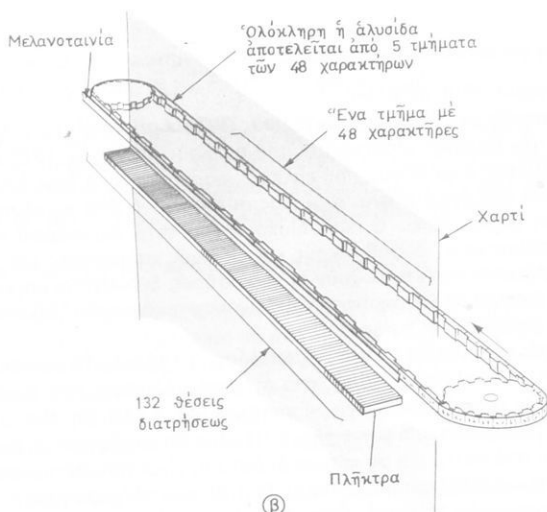
2) **Έκτυπωτές με άλυσίδα (Chain printers)** [σχ. 3.14β (α)]. Τό σύστημα έκτυπώσεώς τους αποτελείται από μία όριζόντια άτέρμονα άλυσίδα, πού χωρίζεται σέ 5 τμήματα. Τό κάθε τμήμα περιέχει 48 χαρακτήρες [σχ. 3.14β (β)]. Καί τά 5 τμήματα περιέχουν τήν ίδια ομάδα από χαρακτήρες. Ή άλυσίδα στρέφεται μέ μεγάλη ταχύτητα. Μπροστά από αυτήν καί σέ κάποια απόσταση είναι τοποθετημένα στή σειρά πολλά μικρά σφυριά-πλήκτρα (μέχρι 132). Μεταξύ των πλήκτρων καί τής άλυσίδας ξετυλίγεται κατακόρυφα τό χαρτί. Όταν ένα πλήκτρο ένεργοποιηθεί τήν κατάλληλη στιγμή, κτυπά τό χαρτί επάνω στήν άλυσίδα καί άποτυπώνει τό χαρακτήρα πού βρέθηκε εκείνη τή στιγμή μπροστά του.

Ή ταχύτητα έκτυπώσεως φθάνει τίς 1000 γραμμές τό λεπτό.

3) **Έκτυπωτές μέ τύμπανο (Drum printers).** Τό σύστημα έκτυπώσεώς τους αποτελείται από ένα κοίλο κυλινδρικό μεταλλικό τύμπανο. Οί διάφοροι χαρακτήρες βρίσκονται ανάγλυφοι στήν έξωτερική του επιφάνεια τοποθετημένοι στή σειρά κα-



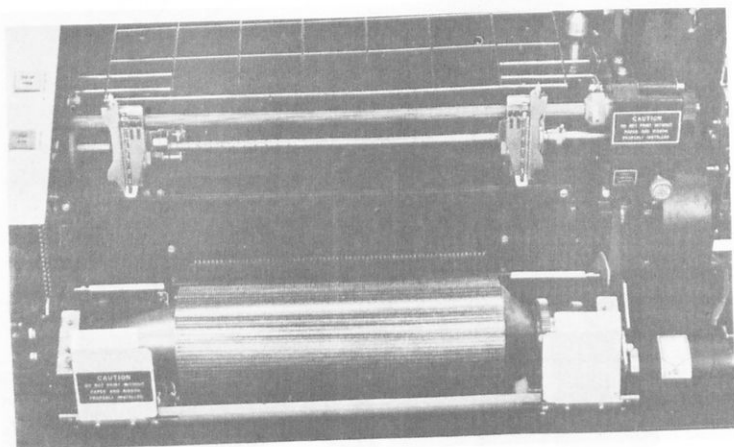
(α)



Σχ. 3.14β.

(α) Έκτυπωτής με αλυσίδα. (β) Ο μηχανισμός έκτυπώσεως ενός έκτυπωτή με αλυσίδα.

τά μήκος των γενετειρών του (σχ. 3.14γ). Μπροστά από τό τύμπανο υπάρχει πάλι μία σειρά από σφυράκια-πλήκτρα. Άνάμεσα στα πλήκτρα καί τό τύμπανο ξετυλίγε-



Σχ. 3.14γ.

Τό σύστημα έκτυπώσεως ενός έκτυπωτή με τύμπανο.

ται τό χαρτί. Τό τύμπανο περιστρέφεται μέ πολύ μεγάλη ταχύτητα (περίπου 1000 στροφές τό λεπτό). Ή ταχύτητα έκτυπώσεώς τους φθάνει τίς 2000 γραμμές τό λεπτό καί μπορούν νά τυπώσουν μέχρι 150 χαρακτήρες σέ κάθε γραμμή.

4) **Γραφομηχανή (Typewriter).** Μπορεί νά θεωρηθεί καί αὐτή ως έκτυπωτική διάταξη μέ πολύ ὅμως μικρότερη ταχύτητα.

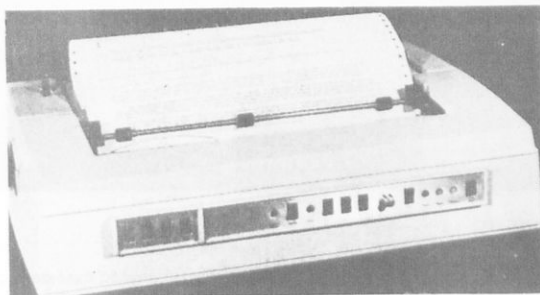
β) Έκτυπωτές χωρίς ἐπαφή (Non impact printers).

Στούς έκτυπωτές αὐτούς οἱ διάφοροι χαρακτήρες σχηματίζονται μέ φωτοχημικό ἢ θερμικό ἢ ἠλεκτροστατικό τρόπο ἐπάνω σέ φωτοευαίσθητο ἢ θερμοευαίσθητο χαρτί (σχ. 3.14δ).

Στήν κατηγορία αὐτή ἀνήκουν:

- 1) Οἱ ἠλεκτροστατικοί έκτυπωτές (Electrostatic printers).
- 2) Οἱ θερμικοί έκτυπωτές (Thermal printers).
- 3) Οἱ ξηρογραφικοί έκτυπωτές (Xerographic printers).
- 4) Οἱ ἠλεκτροοπτικοί έκτυπωτές (Electro - optical printers).

Τά πλεονεκτήματά τους εἶναι ὅτι ἡ ταχύτητά τους μπορεῖ νά φθάσει τίς 64000 γραμμές τό λεπτό καί ὅτι εἶναι ἀθόρυβοι. Τά μειονεκτήματά τους εἶναι ὅτι κοστίζει πολύ τό εἰδικό χαρτί πού χρησιμοποιοῦν καί ὅτι δέν μπορούν νά τυπώσουν σέ περισσότερα ἀπό ἓνα ἀντίτυπα συγχρόνως.

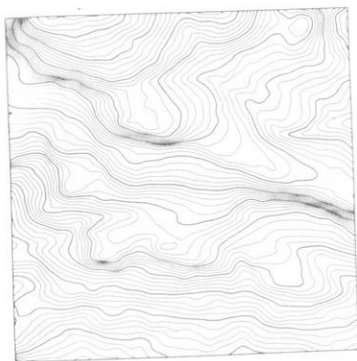


Σχ. 3.14δ.
Ἡλεκτροστατικός έκτυπωτής.

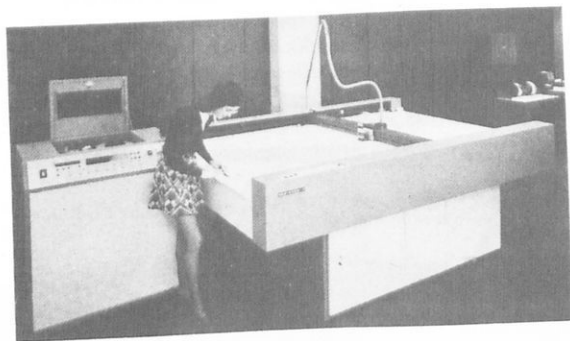
Οἱ περισσότεροι ἀπό τούς έκτυπωτές πού περιγράψαμε μπορούν νά τυπώσουν μία ὁλόκληρη γραμμή κάθε φορά (Line - at - a - time printers). Ἄλλες ὅμως έκτυπωτικές διατάξεις, ὅπως ἡ ἠλεκτρική γραφομηχανή καί τό τηλέτυπο, τυπώνουν ἓνα - ἓνα τούς χαρακτήρες κάθε φορά (Character - at - a - time printers).

3.15 Περιφερειακή μονάδα σχεδιάσεως (Graph Plotter).

Στή μονάδα αὐτή γίνεται ὁ σχεδιασμός καμπυλῶν, διαγραμμάτων καί ἄλλων γραφικῶν ἀπεικονίσεων [σχ. 3.15(α)] σύμφωνα μέ πληροφορίες, πού προέρχονται



(α)



(β)

Σχ. 3.15.

(α) Ένας τοπογραφικός χάρτης σχεδιασμένος από περιφερειακή μονάδα σχεδίασης. (β) Μία περιφερειακή μονάδα σχεδίασης (Plotter).

από την κεντρική μονάδα και κάτω από τον έλεγχο της.

Η περιφερειακή μονάδα σχεδίασης αποτελείται από μία ορθογωνική επίπεδη οριζόντια βάση, επάνω στην οποία τοποθετείται ένα φύλλο κοινό χαρτί. Έπάνω από το χαρτί κινείται μία γραφίδα, οριζόντια ή κάθετα [σχ. 3.15 (β)].

Χρησιμοποιείται σ' εφαρμογές μηχανικής, αστρονομίας, πυρηνικής φυσικής, τοπογραφίας, ωκεανογραφίας κλπ. Είναι η πιο άργη περιφερειακή μονάδα.

3.16 Γραφομηχανή επικοινωνίας (Input / output typewriter).

Η μονάδα αυτή χρησιμοποιείται για την επικοινωνία ανάμεσα στον υπολογιστή και το χειριστή. Διαθέτει πληκτρολόγιο όμοιο με της κοινής γραφομηχανής αλλά και επί πλέον πλήκτρα για τη μετάδοση προς τον υπολογιστή μηνυμάτων. Οι άπαν-

τήσεις του υπολογιστή έκτυπώνονται σε χαρτί που ξετυλίγεται μπροστά από τη μηχανή (σχ. 3.16). Η ταχύτητα έκτυπώσεως είναι μικρή (10 - 30 χαρακτήρες το δευτερόλεπτο) (ch/s).

Η γραφομηχανή επικοινωνίας μπορεί να βρίσκεται σε άμεση σχέση με τον υπολογιστή, δίπλα στο κεντρικό συγκρότημα, ή να είναι τοποθετημένη σε κάποιο απομακρυσμένο σημείο (Remote terminal) και να μεταδίδει από εκεί τα μηνύματα προς τον υπολογιστή.



Σχ. 3.16.

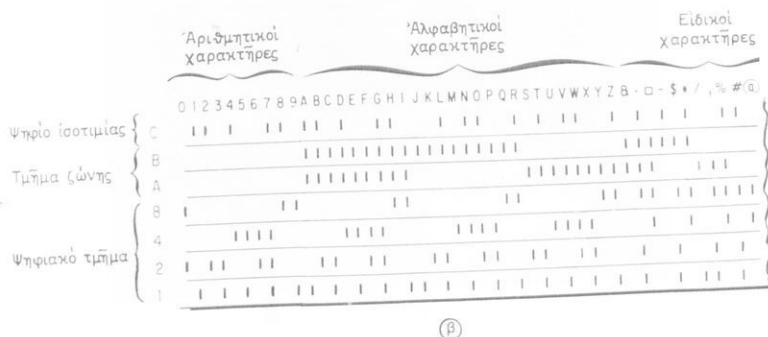
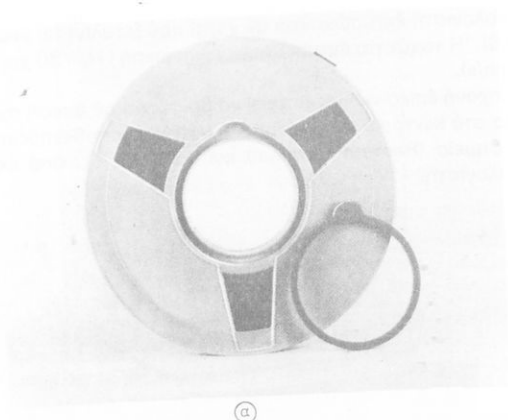
Γραφομηχανή επικοινωνίας.

3.17 Μαγνητική ταινία (Magnetic tape).

Η μαγνητική ταινία είναι πολύ διαδεδομένο μέσο καταγραφής πληροφοριών, γιατί και οικονομική είναι και μπορούμε να αποθηκεύσουμε σ' αυτήν μεγάλο όγκο δεδομένων.

Η μαγνητική ταινία είναι επιμήκης, πλαστική με μικρό πάχος, που ή μία της επιφάνεια έχει επιστρωθεί με κάποιο μαγνητικό υλικό. Τυλίγεται γύρω από ένα καρούλι (μπομπίνα), [σχ. 3.17α (α)]. Τό πλάτος της είναι $\frac{1}{2}$ " - 1" και τό μήκος της κυμαίνεται από 1800 - 3600 ft.

Η ταινία διαιρείται κατά μήκος σε παράλληλες μεταξύ τους ζώνες, τά κανάλια (Channels or tracks). Κάθε χαρακτήρας καταχωρίζεται κωδικοποιημένος κατά τό πλάτος της με τή μορφή ορισμένου αριθμού μαγνητικών στιγμάτων. Η ύπαρξη σε ένα κανάλι ενός μαγνητικού στίγματος ίσοδυναμεί με τό δυαδικό ψηφίο 1 και ή ά-



Σχ. 3.17α.

Μαγνητική ταινία:

(α) Τό καρούλι με τη μαγνητική ταινία («μπομπίνα»). (β) Η κωδικοποίηση σε μία ταινία 7 καναλιών.

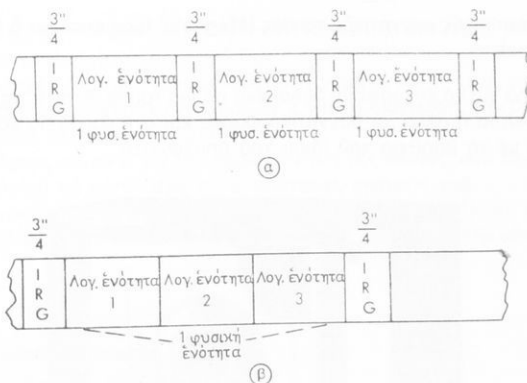
νυπαρξία σε ένα κανάλι μαγνητικού στίγματος ισοδυναμεί με τό δυαδικό ψηφίο 0.

Τό πλήθος των καναλιών μιās ταινίας εξαρτάται από τόν κώδικα, πού χρησιμοποιείται γιά τήν παράσταση των πληροφοριών. Έτσι, αν π.χ. χρησιμοποιείται ό βνήφιος άλφαριθμητικός κώδικας με δυαδικό ψηφίο ίσοτιμίας, τότε ή ταινία θά έχει 7 κανάλια. Ό κώδικας αυτός φαίνεται στό σχήμα [3.17α (β)].

Έξ άλλου αν χρησιμοποιείται ό κώδικας EBCDIC με δυαδικό ψηφίο ίσοτιμίας, ή ταινία θά έχει 9 κανάλια (8 + 1) κ.ο.κ.

Στή μαγνητική ταινία χρησιμοποιείται ό όρος πυκνότητα έγγραφής (Density) γιά νά καταδειχθεί ή ποσότητα των πληροφοριών, πού έχει έγγραφεί στή μονάδα μή- κους. Μετράται με τό bpi (Bytes per inch) ή με τούς χαρακτήρες ανά ίντσα (ch/1"). Η πυκνότητα φθάνει μέχρι τούς 1800 (3000 στους νεώτερους τύπους) χαρα- κήρες ανά ίντσα, πράγμα πού σημαίνει ότι σε μία ταινία μήκους 2400 ποδών μπο-

ρουν να περιληφθούν περισσότεροι από 40 εκατομμύρια χαρακτηρισές που θα χρειαζόνταν διαφορετικά 500.000 δελτία των 80 στηλών. Οι πληροφορίες επάνω στην ταινία συγκροτούν λογικές ενότητες (Records), που και αυτές μπορούν να ομαδοποιηθούν σε μεγαλύτερες φυσικές ενότητες (Blocks). Μεταξύ των λογικών ενότητων όταν αυτές δεν έχουν ομαδοποιηθεί, παρεμβάλλονται κενά διαστήματα μήκους $\frac{3}{4}$ " (Interrecord gaps) [σχ. 3.17β(α)]. Όταν οι λογικές ενότητες δεν έχουν ομαδοποιηθεί, τα κενά παρεμβάλλονται ανάμεσα στις φυσικές ενότητες (Interblock gaps) [σχ. 3.17β(β)].



Σχ. 3.17β.

(α) Κενά ανάμεσα στις λογικές ενότητες (IRG). (β) Κενά ανάμεσα στις φυσικές ενότητες (IRG).

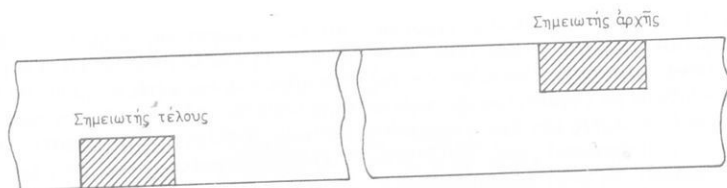
Τα κενά αυτά είναι απαραίτητο να υπάρχουν γιατί, όταν είναι επιθυμητή ή ανάγνωση μιάς συγκεκριμένης λογικής ενότητας, τη στιγμή που αυτή περνά μπροστά από το μηχανισμό ανάγνωσης της μονάδας μαγνητικής ταινίας, ή ταινία χρειάζεται κάποιο χρόνο (χρόνος επιβραδύνσεως) για να άκνητοποιηθεί. Στο χρόνο αυτό διανύει το διάστημα των $\frac{3}{4}$ ".

Η έγγραφη ή η ανάγνωση των πληροφοριών γίνεται καθώς η ταινία περνά μπροστά από τις κεφαλές έγγραφης-ανάγνωσης της μονάδας μαγνητικής ταινίας, άκριβώς όπως συμβαίνει σε ένα κοινό μαγνητόφωνο.

Η επεξεργασία των πληροφοριών πραγματοποιείται με τη σειρά που εμφανίζονται γραμμένες στη μαγνητική ταινία και αυτό γιατί η ταινία κινείται συνεχώς προς τα εμπρός.

Η ταινία κινείται μπροστά από τις κεφαλές με ταχύτητα 15 - 28 ips (ίντσες ανά δευτερόλεπτο) (inches per second), ο δε χρόνος λήψης μιάς πληροφορίας είναι 5 - 20 milliseconds (χιλιοστά του δευτερολέπτου).

Στην αρχή και στο τέλος της ταινίας υπάρχουν δύο τμήματα μήκους 1 ίντσας που καλύπτονται από μικρές μεταλλικές étikétes και ονομάζονται σημειωτές αρχής (Load point marker) και τέλους (End - of tape marker) αντίστοιχα (σχ. 3.17γ).

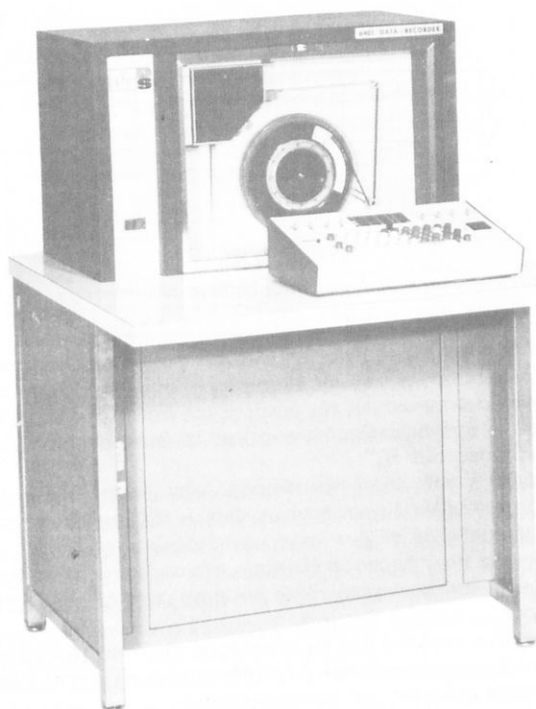


Σχ. 3.17γ.

Οι σημειωτές αρχής και τέλους της ταινίας.

3.18 Κωδικοποιητής μαγνητικής ταινίας (Magnetic tape encoder ή Key - to - tape machine).

Παλιότερα για να έγγραφον δεδομένα σε μία ταινία, έπρεπε να διατρυπηθούν πρώτα σε δελτία ή άλλο μή μαγνητικό φορέα και στη συνέχεια να μεταφερθούν στην ταινία με τη βοήθεια του ίδιου του υπολογιστή.



Σχ. 3.18.

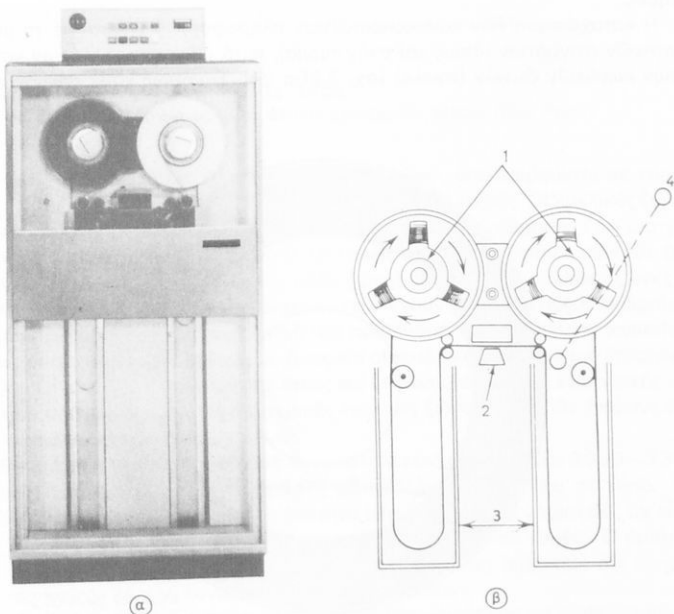
Ο κωδικοποιητής μαγνητικής ταινίας.

Σήμερα, εκτός από τον κλασσικό αυτό τρόπο, χρησιμοποιούνται ειδικές μονάδες, που δέν συνδέονται με το κεντρικό συγκρότημα. Στίς μονάδες αυτές γίνεται ή έγγραφη στήν ταινία των πληροφοριών απ' εύθείας από τά παραστατικά. Ή μονάδα διαθέτει γιά τό σκοπό αυτό πληκτρολόγιο ὁμοιο μέ τῆς κοινῆς γραφομηχανῆς (σχ. 3.18). Στήν ἴδια μονάδα γίνεται καί ή ἐπαλήθευση γιά τήν ὀρθότητα ἐγγραφῆς. Ή ταινία στή συνέχεια εἶναι ἔτοιμη γιά νά τοποθετηθεῖ στήν περιφερειακή μονάδα μαγνητικῆς ταινίας καί νά χρησιμοποιηθεῖ ἀπό τόν ὑπολογιστή.

3.19 Περιφερειακή μονάδα μαγνητικῆς ταινίας (Magnetic tape unit).

Στή μονάδα αὕτη τοποθετεῖται ή μαγνητική ταινία [σχ. 3.19 (α)]. Ἀποτελεῖται ἀπό τά ἐξῆς βασικά τμήματα [σχ. 3.19(β)]:

- Δύο ὑποδοχές 1, ὅπου τοποθετοῦνται ή γεμάτη καί ή κενή μπομπίνες.
- Τρία συστήματα μαγνητικῶν κεφαλῶν 2 γιά τή διαγραφή, τήν ἀνάγνωση καί τήν ἐγγράφη πληροφοριῶν ἀντίστοιχα.
- Δύο θαλάμους κενοῦ 3 γιά τήν ὁμαλή κίνηση τῆς ταινίας, πού τήν προφυλάσσουν ἀπό τοῦ νά κοπεῖ στίς πολύ ἀπότομες κινήσεις τῆς.
- Δύο φωτοκύτταρα 4 πού ἀνιχνεύουν τοὺς σημειωτέες τέλους καί ἀρχῆς.
- Μιά σειρά ἀπό πλῆκτρα μέ τά ὁποῖα ἐλέγχουμε ἐξωτερικά τήν κίνηση τῆς ταινίας.



Σχ. 3.19.

(α) Περιφερειακή μονάδα μαγνητικῆς ταινίας. (β) Τό ἐσωτερικό τῆς μονάδας.

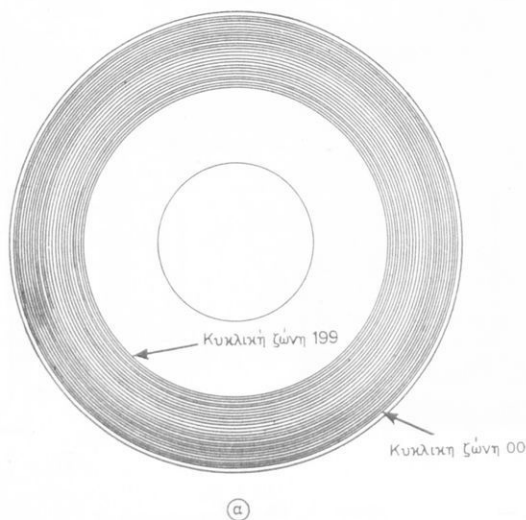
3.20 Μαγνητικός Δίσκος (Magnetic disk).

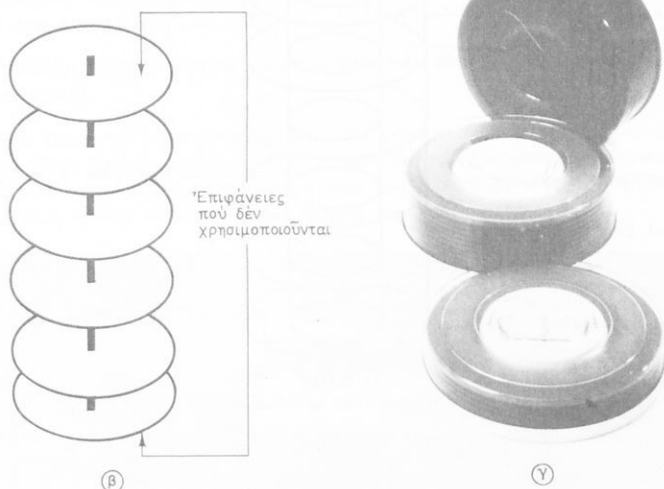
Ο μαγνητικός δίσκος όπως και η μαγνητική ταινία, είναι μέσο μαζικής αποθήκευσης πληροφοριών (Mass storage medium)· γι' αυτό αναφέρεται πολλές φορές και ως *έξωτερική μνήμη* (External memory).

Μοιάζει με ένα κοινό δίσκο γραμμοφώνου. Έχει σχήμα κυκλικό με διάμετρο 30 - 60 cm και κατασκευάζεται από μέταλλο. Οι δύο επιφάνειές του είναι επστρωμένες με μαγνητικό υλικό [σχ. 3.20α (α)]. Πολλοί μαζί δίσκοι στερεωμένοι επάνω σε ένα κοινό κατακόρυφο άξονα, που περνά από τα κέντρα τους, σχηματίζουν μία *δέσμη δίσκων* (Disk rack). Συνήθως οι δέσμες αυτές [σχ. 3.20α(β,γ)] αποτελούνται από 6 δίσκους. Στην περίπτωση της δέσμης ή επάνω επιφάνεια του πρώτου δίσκου και η κάτω του τελευταίου δεν χρησιμοποιούνται. Δηλαδή οι χρήσιμες επιφάνειες, στην περίπτωση αυτή είναι 10 και αριθμούνται με συνεχή αρίθμηση αρχίζοντας από τό 0.

Η χωρητικότητα μιάς δέσμης δίσκων κυμαίνεται μεταξύ 3 - 50 εκατομμύρια χαρακτήρες. Η δέσμη των δίσκων τοποθετείται στη *μονάδα μαγνητικού δίσκου* (Magnetic disk unit) και μπορεί να αφαιρεθεί για να τοποθετηθεί άλλη. Σε μερικές όμως μονάδες οι δίσκοι τοποθετούνται μόνιμα χωρίς να μπορούν να μετακινηθούν (Disk files ή Modules). Στην περίπτωση αυτή η δέσμη αποτελείται από περισσότερους δίσκους και η χωρητικότητα μπορεί να ξεπεράσει τα 200 εκατομμύρια χαρακτήρες.

Η καταχώρηση των κωδικοποιημένων πληροφοριών γίνεται με τη μορφή μαγνητικών στιγμάτων (όπως και στην ταινία), κατά μήκος παραλλήλων και ομοκέντρων *κυκλικών ζωνών* (tracks) [σχ. 3.20α (α)]. Συνήθως κάθε επιφάνεια δίσκου





Σχ. 3.20α.

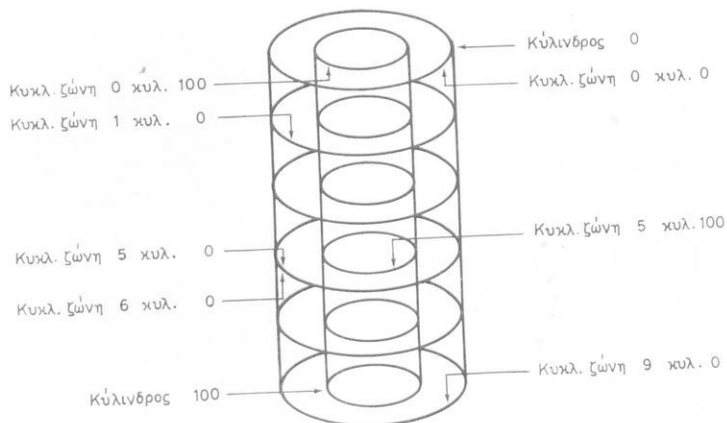
(α) Ο μαγνητικός δίσκος (β, γ) Δέσμες μαγνητικών δίσκων (Disk Packs).

περιέχει 100 - 200 κυκλικές ζώνες. Οι κυκλικές ζώνες υποδιαιρούνται σε τομείς (Sectors). Κάθε κυκλική ζώνη υποδιαιρείται σε 50 - 200 τομείς. Οι κυκλικές ζώνες αριθμούνται επάνω σε κάθε επιφάνεια δίσκου με συνεχή αρίθμηση, έτσι ώστε η εξωτερική να χαρακτηρίζεται ως ζώνη 0. Οι τομείς αριθμούνται πάνω σε κάθε ζώνη με συνεχή αρίθμηση αρχίζοντας (σε κάθε ζώνη) από τό 0. Οι κυκλικές ζώνες όλων των επιφανειών μιας δέσμης δίσκων, που έχουν τον ίδιο αυξαντα αριθμό, σχηματίζουν μία νοητή κυλινδρική επιφάνεια που ονομάζεται *κύλινδρος* (cylinder). Έπομένως υπάρχουν τόσοι κύλινδροι, όσες και οι κυκλικές ζώνες σε μία επιφάνεια (σχ. 3.20β). Οι κύλινδροι αριθμούνται όπως και οι ζώνες. Δηλαδή ο εξωτερικός κύλινδρος, που αποτελείται από τις εξωτερικές κυκλικές ζώνες όλων των επιφανειών, θα είναι ο κύλινδρος 0 κ.ο.κ.

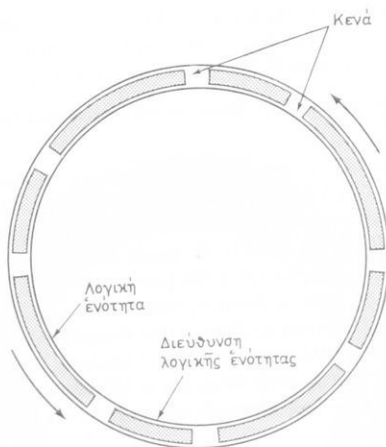
Σε κάθε κυκλική ζώνη μπορούν να εγγραφούν κωδικοποιημένοι 3600 - 7300 χαρακτήρες, ή πυκνότητα όμως εγγραφής μεταβάλλεται από τύπο σε τύπο.

Η θέση όπου βρίσκεται μία λογική ένότητα (Record) μπορεί να βρεθεί, αν είναι γνωστός ο αριθμός της επιφάνειας της κυκλικής ζώνης και του τομέα. Ο αριθμός αυτός είναι μοναδικός και αποτελεί τη λεγόμενη *διεύθυνση* (address) της πληροφορίας, προηγείται δε της λογικής ένότητας (σχ. 3.20γ).

Γιά να βρεθεί επομένως κάποια πληροφορία δεν χρειάζεται να διαβασθούν όλες οι προηγούμενες, όπως συμβαίνει στην ταινία, αλλά ο μηχανισμός αναγνώσεως μπορεί να κατευθυνθεί κατ' ευθείαν στη συγκεκριμένη διεύθυνση και να τη



Σχ. 3.20β.
Ἡ ἔννοια τοῦ κυλίνδρου.



Σχ. 3.20γ.
Ἡ ἔννοια τῆς διευθύνσεως.

διαβάσει. Γιά τὸ λόγο αὐτό ὁ μαγνητικὸς δίσκος χαρακτηρίζεται ὡς μέσο *τυχαίας* ἢ *ἀμεσης προσπελάσεως* (random or direct access).

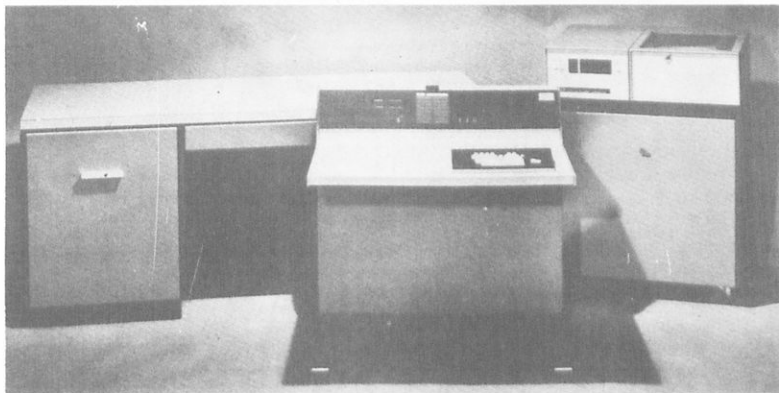
Ὁ μαγνητικὸς δίσκος κατὰ τὴ διάρκεια τῆς ἐπεξεργασίας μπορεῖ νὰ περιστρέφεται μέ ταχύτητα μέχρι 3000 στροφῶν τὸ λεπτό.

Πλεονεκτήματα τοῦ δίσκου εἶναι ἡ πολὺ μεγάλη χωρητικότητά του καί ἡ ὑψηλή

ταχύτητα έπεξεργασίας, ενώ μειονεκτήματα είναι τό υψηλό κόστος του, οι ιδιαίτερες συνθήκες, πού χρειάζεται για τήν αποθήκευσή του καί οι πολύπλοκες τεχνικές οργάνωσης πληροφοριών πού άπαιτεί.

3.21 Κωδικοποιητής μαγνητικού δίσκου (Magnetic disk encoder or Key - to - disk machine).

“Όπως έχομε πει καί στην περίπτωση τής μαγνητικής ταινίας, έτσι καί στό μαγνητικό δίσκο παλιότερα δέν είχαμε τή δυνατότητα νά εγγράψομε τίς πληροφορίες άπό τά παραστατικά άπ’ ευθείας επάνω του, άλλα έπρεπε νά προηγηθεί ή έγγραφή τους επάνω σέ κάποιο άλλο φορέα π.χ. δελτία.



Σχ. 3.21.

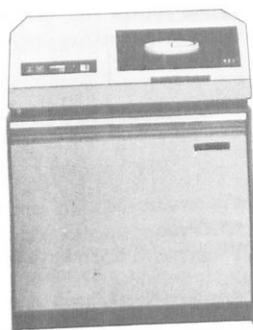
“Ο κωδικοποιητής μαγνητικού δίσκου.

Σήμερα υπάρχουν ειδικές μηχανές, οι *κωδικοποιητές μαγνητικού δίσκου* (σχ. 3.21) μέ τίς όποιες γίνεται άπ’ ευθείας ή έγγραφή στό δίσκο, μέ τή βοήθεια ηλεκτρολογίου. Οι μηχανές αυτές εκτός άπό τό ηλεκτρολόγιο διαθέτουν συχνά καί όθόνη, επάνω στην όποία προβάλλονται τά ηλεκτρολογούμενα δεδομένα. Μπορούν ακόμα νά έλέγχουν τήν όρθότητα έγγραφής όπου χρειάζεται.

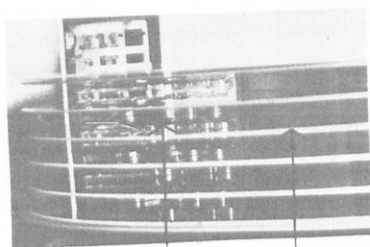
3.22 Περιφερειακή μονάδα μαγνητικού δίσκου (Magnetic disk unit).

Στή μονάδα αυτή τοποθετούνται οι μαγνητικοί δίσκοι (σχ. 3.22α). “Ορισμένοι τύποι δέχονται μόνο άπλούς δίσκους, άλλοι διπλούς καί άλλοι δέσμες μέχρι καί 10 δίσκων. Διαθέτει ειδική ύποδοχή, όπου τοποθετούνται οι δίσκοι καθώς καί ένα σύστημα άπό *βραχίονες* (access arms). Στόν καθένα άπ’ αυτούς προσαρμόζονται οι κεφαλές έγγραφής-άναγνώσεως (Read-write heads). “Ο αριθμός των βραχιόνων καί των κεφαλών ποικίλλει άπό τύπο σέ τύπο.

Στούς πιά εξελιγμένους τύπους οι βραχίονες διαθέτουν πολλές κεφαλές, ώστε ή



(α)

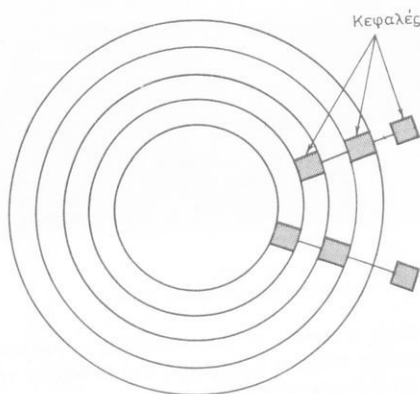


Κεφαλές
έγγραφης ανάγνωσης Μαγνητικοί δίσκοι

(β)

Σχ. 3.22α.

α) Η μονάδα μαγνητικού δίσκου. β) Δέσμη μαγνητικῶν δίσκων.



Σχ. 3.22β.

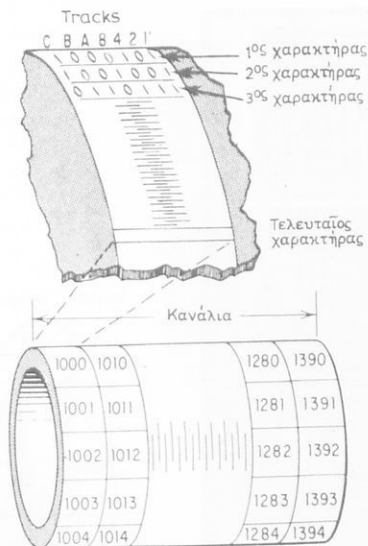
Μονάδα μαγνητικού δίσκου με μία κεφαλή σε κάθε κυκλική ζώνη.

κάθε μία από αυτές να ἐλέγχει μία συγκεκριμένη κυκλική ζώνη (σχ. 3.22β).

Ο χρόνος, πού απαιτείται για να φθάσει ή κεφαλή στο σημείο του δίσκου, όπου έχει γραφεί μία πληροφορία, ονομάζεται *χρόνος προσπελάσεως* (access time) καί στά νεώτερα συστήματα κυμαίνεται από 25 έως 40 milliseconds.

3.23 Μαγνητικό τύμπανο (Magnetic drum).

Τό μαγνητικό τύμπανο έχει τή μορφή κοίλου κυλίνδρου. Είναι μεταλλικό, με διάμετρο 10 - 30 cm, ή δέ εξωτερική του ἐπιφάνεια έχει ἐπιστρωθεί με κάποιο μα-



Σχ. 3.23.
Τό μαγνητικό τύμπανο.

γνητικό υλικό. Χαρακτηρίζεται, όπως και η ταινία και ο δίσκος, ως εξωτερική μνήμη.

Η μαγνητική του επιφάνεια διαιρείται σε δακτυλοειδείς ζώνες (κάθετα προς τό διαμήκη άξονά του) που ονομάζονται tracks και σε επιμήκεις λωρίδες (παράλληλες προς τόν άξονά του) που ονομάζονται κανάλια (channels) (σχ. 3.23).

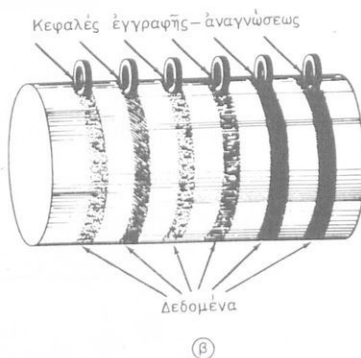
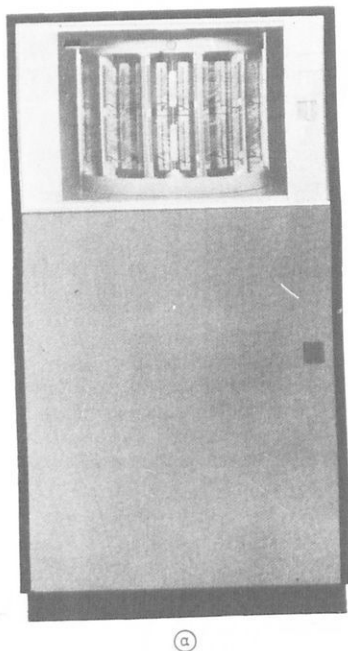
Οι πληροφορίες καταχωρίζονται κωδικοποιημένες, με τή μορφή μαγνητικών στιγμάτων, κατά μήκος των ζωνών. Κάθε διακεκριμένη πληροφορία έχει μία διεύθυνση, που αντιστοιχεί στή ζώνη και στό κανάλι, στή διασταύρωση των οποίων καταχωρίζεται (σχ. 3.23). Η χωρητικότητα των μαγνητικών τυμπάνων είναι $10^5 - 10^9$ bits δηλαδή περίπου $10^4 - 10^8$ χαρακτήρες. Η πυκνότητα έγγραφης κυμαίνεται από 10 - 60 bits/mm. Τό μαγνητικό τύμπανο χαρακτηρίζεται, όπως και ο δίσκος, ως μέσο τυχαίας προσπελάσεως.

3.24 Περιφερειακή μονάδα μαγνητικού τυμπάνου (Magnetic drum unit).

Στή μονάδα αυτή τοποθετείται τό μαγνητικό τύμπανο [σχ. 3.24(α)]. Αποτελείται βασικά:

α) Από ένα άξονα, στόν όποιο προσαρμόζεται τό τύμπανο και που μπορεί νά τό περιστρέφει με σταθερή ταχύτητα 3000 - 12500 στροφές τό λεπτό.

β) Από μία σειρά κεφαλών άναγνώσεως-έγγραφης, που είναι τοποθετημένες



Σχ. 3.24.

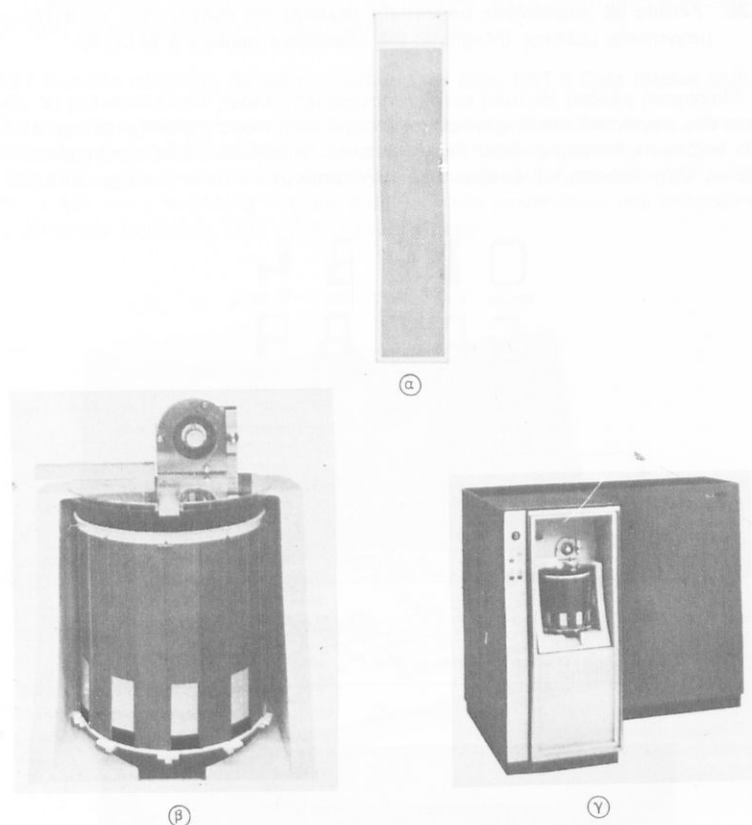
(α) Μονάδα μαγνητικού τυμπάνου. (β) Τό σύστημα κεφαλών άναγνώσεως - έγγραφής.

κατά μήκος μιās γεννέτειρας του τυμπάνου καί σέ μικρή άπόσταση άπό τήν έπιφάνεια του. Υπάρχουν τόσες κεφαλές, όσες καί οί ζώνες του τυμπάνου [σχ. 3.24(β)].

3.25 Μαγνητικές κάρτες (Magnetic cards or strips) καί μονάδα μαγνητικών καρτών (Data cell).

Ένω ή μαγνητική ταινία, ό δίσκος καί τό τύμπανο είναι φορείς, πού χαρακτηρίζονται άπό μία συνεχή καί μεγάλη μαγνητική έπιφάνεια, ή μαγνητική κάρτα αντίθετα είναι μία όρθογώνια πλαστική κάρτα μικρών διαστάσεων (1 - 3" x 3,5 - 14"). Η κάρτα αύτή έπικαλύπτεται άπό τή μία της έπιφάνεια μέ ένα λεπτό στρώμα μαγνητικού ύλικού. Η έπιφάνεια τής κάρτας χωρίζεται καθ' όλο τό μήκος της σέ παράλληλες ζώνες (tracks), όπως καί στήν ταινία [σχ. 3.25(α)]. Πολλές μαζί κάρτες είναι τοποθετημένες σέ είδική θήκη, πού όνομάζεται *κυψέλη* (cell). Η κυψέλη [σχ. 3.25(β)] τοποθετείται σέ μία ύποδοχή τής μονάδας μαγνητικών καρτών [σχ. 3.25(γ)].

Γιά νά διαβασθοϋν τά δεδομένα, πού βρίσκονται έπάνω σέ μία κάρτα, άναζητείται καί έπιλέγεται ή κάρτα μέ κάποιο μηχανισμό καί στή συνέχεια οδηγείται σέ ένα



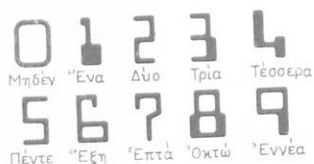
Σχ. 3.25.

(α) Μαγνητική κάρτα. (β) Κυψέλη μαγνητικών καρτών (γ) Μονάδα μαγνητικών καρτών.

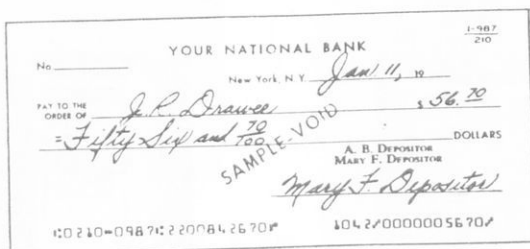
γρήγορα περιστρεφόμενο τύμπανο, όπου και περιτυλίγεται. Έπάνω από τό τύμπανο βρίσκεται τό σύστημα των κεφαλών ανάγνωσης-έγγραφης. Όταν τελειώσει ή ανάγνωση, ή κάρτα απελευθερώνεται καί έπιστρέφει στη θέση της. Η διαδικασία αυτή μοιάζει μέ τή διαδικασία έπιλογής ενός δίσκου σέ ένα «τζούκ - μπόξ». Για τήν έγγραφή δεδομένων σέ μιά κάρτα ακολουθείται ό ίδιος τρόπος. Οι μαγνητικές κάρτες έχουν βραδυ σχετικά ρυθμό προσπελάσεως (100 - 600 m/s), έχουν όμως ως πλεονεκτήματα τό ότι είναι φθηνές, μπορούν εύκολα νά αντικατασταθούν καί νά άποταμιεύσουν μεγάλο όγκο πληροφοριών. Η χωρητικότητα μιάς μονάδας μαγνητικών καρτών ξεπερνά τά 40 έκατομμύρια χαρακτήρες.

3.26 Έντυπα με χαρακτήρες μαγνητικής μελάνης και αναγνώστες χαρακτήρων μαγνητικής μελάνης (Magnetic ink characters readers ή M.I.C.R).

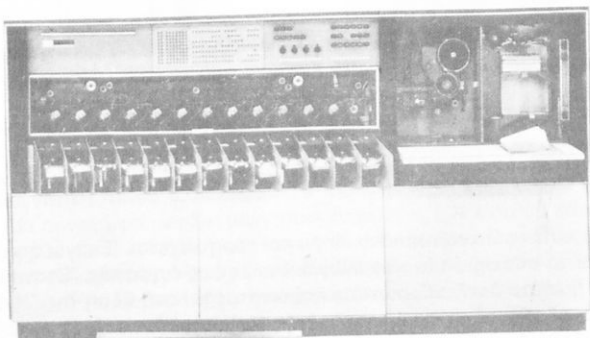
Μαγνητική μελάνη (δηλαδή κοινή τυπογραφική μελάνη εμπλουτισμένη με σίδηρα από μαγνητικό ύλικό) χρησιμοποιείται για το τύπωμα χαρακτήρων [σχ. 3.26 (α)] επάνω σε διαφόρων διαστάσεων έντυπα. Η μέθοδος αυτή χρησιμοποιείται κυρίως στην έκδοση και επεξεργασία των τραπεζικών έπιταγών [σχ. 3.26(β)].



(α)



(β)



(γ)

Σχ. 3.26.

(α) Ένας τύπος χαρακτήρων μαγνητικής μελάνης. (β) Έπιταγή με χαρακτήρες μαγνητικής μελάνης (γ) Αναγνώστης χαρακτήρων μαγνητικής μελάνης.

Ο αναγνώστης χαρακτήρων μαγνητικής μελάνης [σχ. 3.26(γ)] μπορεί να διαβάσει τους χαρακτήρες με ηλεκτρομαγνητικό τρόπο.

3.27 Μονάδα προβολής σε οθόνη (Cathode ray tube, CRT ή Data display unit).

Η μονάδα αυτή αποτελείται από μία οθόνη (καθοδικό σωλήνα) όμοια με την οθόνη της τηλεόρασης και από ένα πληκτρολόγιο, όμοιο με το πληκτρολόγιο κοινής γραφομηχανής (σχ. 3.27). Η οθόνη έχει διάφορες διαστάσεις (6" x 8", 12" x 12" κλπ.). Ανάλογος είναι και ο αριθμός των χαρακτήρων που μπορούν να προβληθούν (συνήθως 200 - 4000 χαρακτήρες).



Σχ. 3.27.
Μονάδα προβολής σε οθόνη.

Οι μονάδες αυτές χρησιμοποιούνται και σε κεντρικό συγκρότημα ενός υπολογιστή, αλλά κυρίως σε τερματικούς σταθμούς (terminals), που βρίσκονται μακριά από το κεντρικό συγκρότημα.

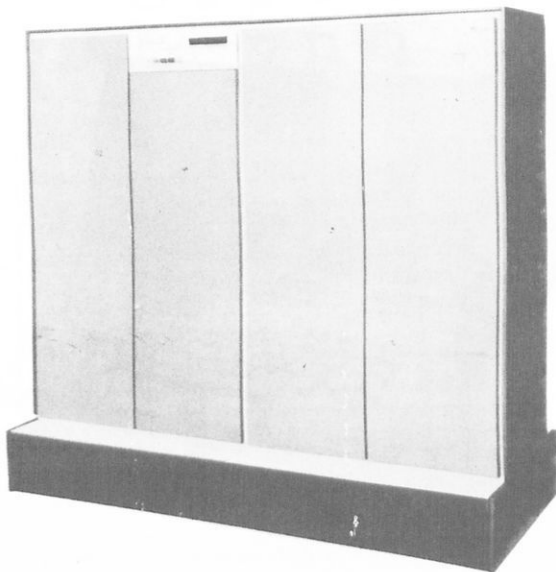
Υπάρχουν δύο τύποι:

- Οθόνες, όπου προβάλλονται μόνο αλφαριθμητικοί και άλλοι χαρακτήρες (Visual Display Units ή VDU).
- Οθόνες, όπου εκτός από χαρακτήρες προβάλλονται και καμπύλες, διαγράμματα κλπ. (graphics). Αυτές είναι εφοδιασμένες με ειδική γραφίδα-φωτοκύτταρο (light pen), που με τη βοήθειά της γίνονται διορθώσεις ή συμπληρώσεις των προβαλλομένων καμπυλών.

Σε μερικούς τύπους είναι δυνατή η αποτύπωση του περιεχομένου της οθόνης επάνω σε μικροφίλμ ή κανονικό φωτογραφικό φίλμ.

3.28 Περιφερειακή μονάδα ήχουστικής άποκρίσεως (Audio response unit).

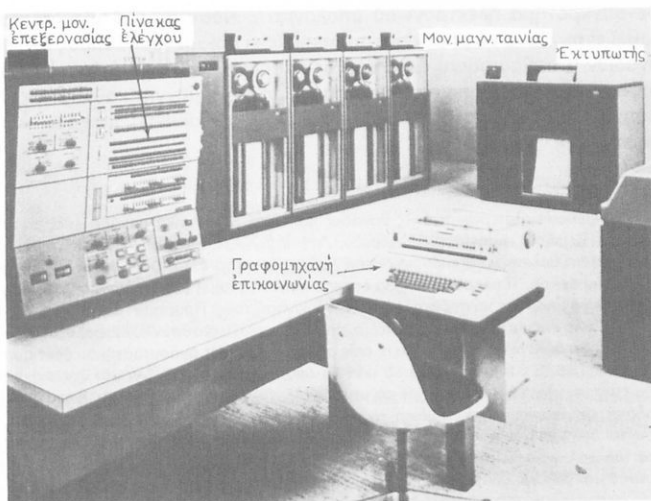
Μία μονάδα ιδιαίτερα χρήσιμη στά συστήματα πραγματικού χρόνου (Real time systems), γιά τά όποία θά μιλήσουμε πιό άναλυτικά στήν παράγραφο 16.3.2, είναι καί ή μονάδα ήχουστικής άποκρίσεως (σχ. 3.28). Ή μονάδα αύτή δίνει πληροφορίες άπό τόν ύπολογιστή μέ τή μορφή φωνητικών σημάτων-λέξεων. Ένα πλήθος άπό φωνητικές συλλαβές βρίσκειται άποθηκευμένο σέ μία βοηθητική μνήμη π.χ. ένα μαγνητικό τύμπανο. Τό ήχητικό μήνυμα τής άπαντήσεως σέ κάποια έρώτηση, συναρμολογείται μέ τή βοήθεια ειδικού προγράμματος άπό τίς άποθηκευμένες συλλαβές ή λέξεις καί μεταδίδεται άπό μεγάφωνο ή ήχουστικό τηλεφώνου.



Σχ. 3.28.
Μονάδα ήχητικής άποκρίσεως.

3.29 Πίνακας έλέγχου (Console).

Είναι μία μονάδα γιά τόν έξωτερικό έλεγχο τής καταστάσεως του κεντρικού συγκροτήματος. Περιέχει διακόπτες, ένδεικτικές λυχνίες καί πλήκτρα, (σχ. 3.29) μέ τή βοήθεια των όποιων ό χειριστής μπορεί νά θέσει σέ λειτουργία τό συγκρότημα ή νά τό σταματήσει καί άκόμα μπορεί νά έλέγχει κάθε στιγμή, άν ή λειτουργία είναι όμαλή. Ή μονάδα αύτή πλαισιώνεται άπό τή γραφομηχανή έπικοινωνίας (I/O Type-writer).



Σχ. 3.29.

Ένα συγκρότημα ψηφιακού ηλεκτρονικού υπολογιστή. Άριστερά διακρίνεται η κεντρική μονάδα επεξεργασίας, ο πίνακας ελέγχου (console) και στο βάθος 4 μονάδες μαγνητικής ταινίας επάνω δεξιά διακρίνεται ένας έκτυπωτής (Printer). Μπροστά στη φωτογραφία είναι η γραφομηχανή επικοινωνίας (I/O Typewriter).

3.30 Τερματικοί σταθμοί (Terminals).

Η κεντρική μονάδα ενός ηλεκτρονικού υπολογιστή έχει τη δυνατότητα να συνδεθεί σε αρκετές περιπτώσεις με περιφερειακές μονάδες, που βρίσκονται σε μεγάλες αποστάσεις από αυτήν. Οι περιφερειακές αυτές μονάδες, που χρησιμοποιούνται για την είσοδο ή έξοδο πληροφοριών προς και από το κεντρικό συγκρότημα, συγκροτούν ένα τερματικό σταθμό.

Η σύνδεση του τερματικού σταθμού με το κεντρικό συγκρότημα γίνεται με τη βοήθεια τηλεφωνικών ή τηλεγραφικών γραμμών. Ο τερματικός σταθμός, εκτός από τις περιφερειακές μονάδες, διαθέτει τις περισσότερες φορές και μία μικρή μονάδα μνήμης.

Έτσι ένας απλός τυπικός τερματικός σταθμός μπορεί να αποτελείται μόνο από ένα τηλέτυπο ή μία μονάδα οθόνης με πληκτρολόγιο ή να περιέχει άλλες μονάδες, όπως μονάδα ανάγνωσης δελτίων και έκτυπωτή ή μονάδα ανάγνωσης χαρτοταινίας και έκτυπωτή κ.ά. μαζί και με μία μονάδα μνήμης.

3.31 Μερικές παρατηρήσεις.

Όλες οι περιφερειακές μονάδες, που περιγράψαμε πιο πάνω δεν συναντώνται

σέ κάθε συγκρότημα ηλεκτρονικού υπολογιστή. Κάθε βασικό συγκρότημα έχει σχεδιασθεί έτσι, ώστε να μπορεί να συμπεριλάβει ορισμένους τύπους περιφερειακών μονάδων (Configuration) αλλά επίσης και ορισμένο αριθμό από τόν κάθε τύπο, όπως π.χ. 4 μονάδες μαγνητικής ταινίας, 2 μονάδες μαγνητικού δίσκου κλπ. Βέβαια υπάρχουν πάντα δυνατότητες επέκτασής (Expandibility), που δεν είναι όμως απαραίτητες.

Έρωτήσεις - ασκήσεις.

1. Σέ τί χρησιμεύουν οι περιφερειακές μονάδες και σέ τί η κεντρική μονάδα επεξεργασίας;
2. Αναφέρετε μερικούς φορείς πληροφοριών.
3. Υποθέσατε ότι θέλετε να δημιουργήσετε ένα αρχείο για τα υλικά μιάς αποθήκης επάνω σε όγδοντάστημα δελτία. Τί πληροφορίες θα έπρεπε κατά τη γνώμη σας να συγκεντρώσετε για κάθε υλικό, ώστε να μπορείτε να παρακολουθείτε την κίνησή τους; Προσπαθείστε κατόπιν να σχεδιάσετε κατάλληλα επάνω στο δελτίο τα πεδία, όπου θα καταχωρισθούν οι διάφορες πληροφορίες.
4. Υποθέσατε ότι θέλετε να διατρυπήσετε στίς στήλες 1 - 5 ενός όγδονταστίλου δελτίου τό δεκαδικό αριθμό 15683 και στίς στήλες 10 - 19 τό όνομα NICOLAIDIS. Αν δεν έχετε στη διάθεσή σας διατρητική μηχανή, μαυρίστε μέ τό μολύβι σας τίς θέσεις του δελτίου, όπου θα γίνουν οι διατρήσεις, σύμφωνα μέ τόν κώδικα του *Hollerith*.
5. Τί καλείται διαλογή και τί συνένωση δύο ομάδων δελτίων; Ποιά σχέση υπάρχει μεταξύ τους;
6. Δίνεται μία μαγνητική ταινία μέ ωφέλιμο μήκος 2400 ft. Αν έμεις καταχωρίζομε τούς χαρακτήρες μέ πυκνότητα 800 bpi και 100 χαρακτήρες αποτελούν μία λογική ένότητα (record), βρήτε πόσες λογικές ένότητες μπορούν να έγγραφούν στην ταινία, αν υποθεθεί ότι οι λογικές ένότητες δεν είναι ομαδοποιημένες.
7. Η ίδια άσκηση μέ την προηγούμενη μέ τη διαφορά ότι τώρα οι λογικές ένότητες είναι ομαδοποιημένες σε ομάδες μέ 10 ένότητες ή κάθε μία.
8. Δίνεται μία δέσμη από 10 δίσκους. Κάθε επιφάνεια υποδιαιρείται σε 100 κυκλικές ζώνες και κάθε ζώνη σε 100 τομείς. Μέσα σε κάθε τομέα καταχωρίζονται 80 χαρακτήρες. Ποιά είναι η χωρητικότητα της δέσμης σε χαρακτήρες;
9. Αν σās ζητούσαν να δημιουργήσετε ένα αρχείο επάνω σε κάποιο φορέα, τί θα προτιμούσατε, τά δελτία, τη μαγνητική ταινία, τό μαγνητικό δίσκο, τό μαγνητικό τύμπανο ή τη χαρτοταινία και γιατί;
10. Αν σās δίνουν ένα αρχείο σε μαγνητική ταινία, της μισθοδοσίας των υπαλλήλων μιάς επιχείρησης, και ένα αρχείο σε διάτρητα δελτία μέ τίς μεταβολές τους, και σās ζητούν να κάνετε επεξεργασία και να τυπώσετε μία κατάσταση, πόσες και ποιές περιφερειακές μονάδες θα χρειασθείτε;

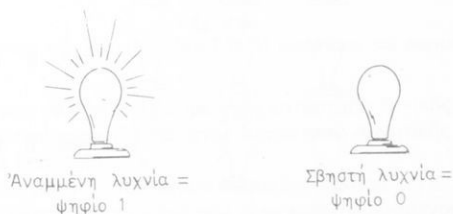
ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΨΗΦΙΑΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ II Η ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ

4.1 Ἡ κεντρική μνήμη — Γενικά.

Ὅπως εἶπαμε στὴν ἀρχὴ τοῦ προηγούμενου κεφαλαίου, στὴν κεντρικὴ μνήμη ἀποθηκεύονται οἱ διάφορες πληροφορίες εἴτε ὡς ἐντολές, εἴτε ὡς ἀπλά δεδομένα σὲ κωδικοποιημένη μορφή μὲ τὴ βοήθεια ἐνὸς ἀπὸ τὰ ἀριθμητικὰ συστήματα, πού περιγράψαμε.

Ἡ κωδικοποίηση τῶν πληροφοριῶν μὲ δυαδική μορφή σὲ κάποιο σύστημα παρουσιάζει μεγαλύτερη ἀπλότητα ἀπὸ τὴν κωδικοποίηση μὲ ὁποιοδήποτε ἄλλο τρόπο. Αὐτό ὀφείλεται στοῦ γεγονότος ὅτι τὸ δυαδικό ἔχει μόνο δύο ψηφία, τὸ 0 καὶ τὸ 1, πού μποροῦν εὐκόλα νά παρασταθοῦν ἀπὸ ὁποιοδήποτε φυσικό σύστημα πού παρουσιάζει δύο διαφορετικὲς καταστάσεις. Μποροῦμε π.χ νά παραστήσουμε τὸ ψηφίο 0 μὲ μία σβηστή λυχνία καὶ τὸ 1 μὲ μία ἀναμμένη (σχ. 4.1α).



Σχ. 4.1α.

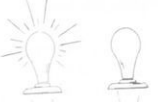
Παράσταση τῶν ψηφίων 0 καὶ 1 μὲ μία λυχνία.

Ἄν λοιπόν ἔχομε δύο λυχνίες, τὴ μία δίπλα στὴν ἄλλη, τότε ἀνάλογα μὲ τὴν κατάστασή τους μποροῦμε νά παραστήσουμε 4 διαφορετικούς διψήφιους δυαδικούς ἀριθμούς (σχ. 4.1β).

Ἄν διαθέτομε τρεῖς λυχνίες στὴ σειρά, μποροῦμε νά παραστήσουμε 8 διαφορετικούς τριψήφιους δυαδικούς ἀριθμούς (σχ. 4.1γ).

Καί γενικά ἂν διαθέτομε n λυχνίες μποροῦμε νά παραστήσουμε 2^n διαφορετικούς n -ψηφίους δυαδικούς ἀριθμούς.

Ἡ λυχνία ὅμως δὲν εἶναι γιὰ πολλοὺς λόγους πρακτικό μέσο ἀποθηκεύσεως πληροφοριῶν, γι' αὐτὸ χρησιμοποιοῦμε ἄλλες τεχνικές, πού θὰ περιγράψουμε στὴ συνέχεια.

	Δυαδικός αριθμός	Αντίστοιχος δεκαδικός
	$\Rightarrow$ 0 0	0
	$\Rightarrow$ 0 1	1
	$\Rightarrow$ 1 0	2
	$\Rightarrow$ 1 1	3

Σχ. 4.1β.

Παράσταση διψήφιων δυαδικών
αριθμών με δύο λυχνίες.

	Δυαδικός αριθμός	Δεκαδικός αριθμός
	$\Rightarrow$ 0 0 0	0
	$\Rightarrow$ 0 0 1	1
	$\Rightarrow$ 0 1 0	2
	$\Rightarrow$ 0 1 1	3
	$\Rightarrow$ 1 0 0	4
	$\Rightarrow$ 1 0 1	5
	$\Rightarrow$ 1 1 0	6
	$\Rightarrow$ 1 1 1	7

Σχ. 4.1γ.

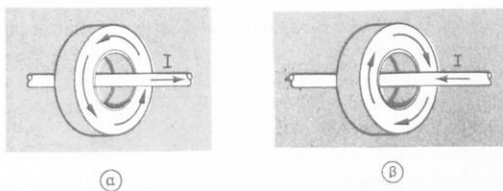
Παράσταση τριψήφιων δυαδικών
αριθμών με τρεις λυχνίες.

4.2 Μαγνητικοί πυρήνες (Magnetic Cores).

Είναι ή πιό παλιά καί ή πιό διαδεδομένη μορφή μνημών καί βασίζεται στην ιδιότητα, πού έχει τό ηλεκτρικό ρεύμα, νά μαγνητίζει όρισμένα ύλικά.

“Ας πάρουμε ένα μικρό δακτύλιο φτιαγμένο από κάποιο μαγνητιζόμενο ύλικό (φερρίτη) καί ας περάσουμε κάθετα μέσα από αυτό ένα λεπτό άγώγιμο σύρμα. “Ο δακτύλιος είναι στην άρχή άμαγνήτιστος. “Αν μέσα από τό σύρμα διοχετεύσουμε ρεύμα κατάλληλης έντάσεως, ό δακτύλιος θά μαγνητισθεί έτσι, ώστε ή φορά των δυναμικών γραμμών του πεδίου του νά καθορίζεται, όπως γνωρίζουμε από την ηλεκτροτεχνία, από τό γνωστό κανόνα του δεξιού χεριού.¹

“Αν διακόψουμε τό ρεύμα ό δακτύλιος θά διατηρήσει τή μαγνητική του κατάσταση [σχ. 4.2α(α)]. “Αν αντίστρέψουμε τή φορά του ρεύματος, τότε αντίστρέφεται καί ή φορά των δυναμικών γραμμών του μαγνητικού πεδίου [σχ. 4.2α(β)]. Για εύκολία μας θά ονομάζουμε τίς δύο διαφορετικές καταστάσεις μαγνήτισεως *θετική* καί *άρνητική*.



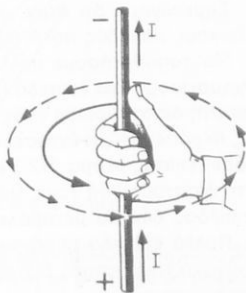
Σχ. 4.2α.

α) “Η κατάσταση του δακτυλίου είναι 1, β) “Η κατάσταση του δακτυλίου είναι 0.

Μπορούμε λοιπόν νά αντιστοιχίσουμε στην κατάσταση θετικής μαγνήτισεως του δακτυλίου τό δυαδικό ψηφίο 1 καί στην κατάσταση άρνητικής μαγνήτισεως, τό δυαδικό ψηφίο 0 (σχ. 4.2α).

Τούς δακτυλίους αυτούς, πού έχουν διάμετρο περίπου 1/16”, τούς ονομάζουμε πυρήνες ή καί φερρίτες, από τό ύλικό πού είναι κατασκευασμένοι. Πολλοί πυρήνες

¹ Σύμφωνα μέ τόν κανόνα αυτό άν μέ τό δεξιό χέρι κρατήσουμε ένα άγώγιμο έτσι, ώστε ό άνοικτός άντίχειρας νά δείχνει τή φορά του ρεύματος I , τότε τά σφιγμένα δάκτυλα θά δείχνουν τή φορά των μαγνητικών γραμμών.

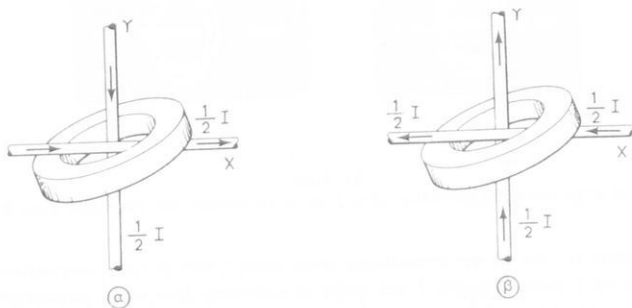


περασμένοι σέ ένα άγωγό, όπως οί χάνδρες ενός κομπολογιού, θά συμβολίζουν, όπως έξυπακούεται, μία σειρά από δυαδικά ψηφία (bits).

“Ας ύποθέσομε ότι ή ένταση του ρεύματος, πού χρειάζεται γιά νά μαγνητισθεί ένας πυρήνας, είναι I καί ότι άντί νά περνά μέσα από τόν πυρήνα ένας μόνο άγωγός, περνούν δύο, ό X καί ό Y κάθετοι μεταξύ τους. Ό καθένας από αυτούς σχηματίζει γωνία 45° μέ τό δακτύλιο καί μπορεί νά διαρρέεται από ρεύμα έντάσεως $1/2 I$.

“Αν από τόν άγωγό X περάσει ρεύμα $1/2 I$ μέ φορά πρós τά δεξιά καί από τόν άγωγό Y περάσει ρεύμα $1/2 I$ μέ φορά πρós τά κάτω [(σχ. 4.2β(α)), ό άρχικά άμαγνήτιστος πυρήνας θά μαγνητισθεί θετικά, πράγμα πού σημαίνει κατάσταση 1 (ή τιμή του bit = 1). “Αν όμως ή φορά των ρευμάτων στους δύο άγωγούς είναι αντίθετη, ή μαγνήτιση του πυρήνα θά είναι καί αυτή αντίθετη, πράγμα πού αντιστοιχεί σέ κατάσταση 0 (ή τιμή του bit = 0) [σχ. 4.2β(β)].

“Αν περάσει μόνο από τόν ένα άγωγό ρεύμα $1/2 I$, ό πυρήνας δέν θά μαγνητισθεί. Τό ίδιο θά συμβεί καί όταν τό ρεύμα πού περνά από τόν X έχει φορά πρós τά δεξιά, ενώ τό ρεύμα πού περνά από τόν Y έχει φορά πρós τά επάνω, καί αντίστροφα.



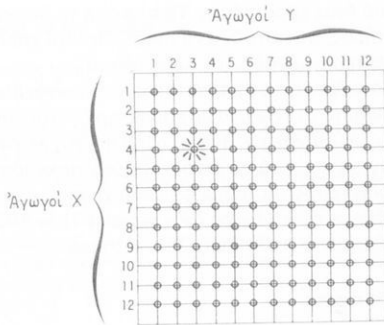
Σχ. 4.2β.

Οί φορές των ρευμάτων των άγωγών X καί Y γιά θετική (κατάσταση 1) καί άρνητική (κατάσταση 0) μαγνήτιση του δακτυλίου.

Σημειώνουμε ότι όταν μιλάμε γιά ρεύμα, στην πραγματικότητα έννοούμε ηλεκτρικούς παλμούς πολύ μικρής χρονικής διάρκειας.

“Αν τοποθετήσομε πολλούς πυρήνες σέ μία δυδιάστατη διάταξη (σχ. 4.2γ) θά σχηματίσομε ένα επίπεδο (Magnetic - core plane), όπου κάθε πυρήνας θά βρίσκεται στή διασταύρωση ενός άγωγού X καί ενός Y . “Όταν από ένα άγωγό X , π.χ. τόν X_4 , περάσει ρεύμα έντάσεως $1/2 I$ μέ φορά πρós τά δεξιά καί από τόν άγωγό Y_3 περάσει επίσης ρεύμα $1/2 I$ μέ φορά πρós τά κάτω, τότε ό πυρήνας, πού βρίσκεται στή διασταύρωσή τους, θά βρεθεί σέ κατάσταση 1. Κανένας άλλος πυρήνας του επιπέδου δέν θά μεταβάλει την κατάστασή του.

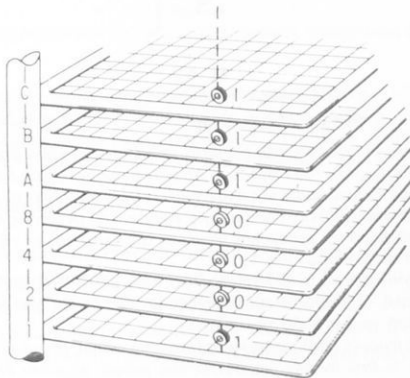
Πολλά επίπεδα πυρήνων τοποθετημένα τό ένα επάνω από τό άλλο δημιουργούν μία τριδιάστατη διάταξη. Οί πυρήνες πού βρίσκονται επάνω στην ίδια στήλη, χρησιμοποιούνται γιά την αποθήκευση ενός κωδικοποιημένου χαρακτήρα. “Ετσι άν



Σχ. 4.2γ.

Ένα επίπεδο μαγνητικών πυρήνων.

εχομε π.χ. 7 επίπεδα πυρήνων τό ένα επάνω από τό άλλο, τότε τά 7 στοιχειά (bits), πού βρίσκονται σέ μία τυχαία στήλη τής διατάξεως, θά μπορούν νά παραστήσουν ένα χαρακτήρα κωδικοποιημένο στόν έπταψηφίο άλφαριθμητικό κώδικα (δηλαδή τόν 6-ψηφίο άλφαριθμητικό BCD + τό ψηφίο τής ίσοτιμίας). Ό χαρακτήρας Α π.χ στόν κώδικα αυτόν έχει τή μορφή 1110001. Διοχετεύοντας λοιπόν ρεύματα μέ έντάσεις $1/2 I$ από τούς κατάλληλους άγωγούς Χ καί Υ τών 7 επιπέδων, μπορούμε νά σχηματίσουμε κατά μήκος μιάς στήλης τόν κωδικοποιημένο χαρακτήρα Α (σχ. 4.2δ).



Σχ. 4.2δ.

Καταχώριση του κωδικοποιημένου χαρακτήρα Α στή μνήμη.

Φυσικά ή μνήμη δέν αποτελείται από 7 μόνο επίπεδα μαγνητικών πυρήνων, αλλά από πολύ περισσότερα έτσι, πού επάνω στήν ίδια στήλη νά μπορούν νά άπείκο-

νίζονται περισσότεροι από ένας χαρακτήρες. Τα bits από τα οποία αποτελείται ένας κωδικοποιημένος χαρακτήρας, συνιστούν ένα byte¹. Πολλοί χαρακτήρες μαζί αποτελούν συχνά μία μεγαλύτερη μονάδα, τη λέξη (Word).

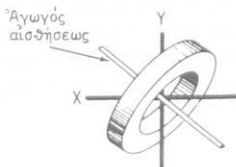
Για να είναι εύκολη η ανεύρεση των χαρακτήρων, που είναι αποθηκευμένοι στις διάφορες περιοχές της μνήμης, οι τελευταίες χαρακτηρίζονται από διευθύνσεις. Η φυσική αντιστοίχιση των διευθύνσεων γίνεται στην πρώτη λειτουργία του υπολογιστή με ένα ειδικό πρόγραμμα, που τό παρέχει η κατασκευάστρια εταιρία.

Τό πιο χαρακτηριστικό μέγεθος της μνήμης είναι η χωρητικότητά της (capacity ή memory size), που εκφράζεται σε k - bytes ή k - words ($1k = 1024$). Οι χωρητικότητες κυμαίνονται συνήθως από 16 k έως 256 k bytes.

4.3 Ανάγνωση της τιμής ενός πυρήνα.

Ας δούμε τώρα με ποιο τρόπο μπορούμε να ανιχνεύσουμε την κατάσταση ενός πυρήνα. Για να διαβάσουμε την τιμή ενός πυρήνα, στέλνουμε από τους άγωγούς X και Y, που τόν διαρρέουν, ρεύματα έντασης $1/2 I$ και με φορές τέτοιες, σαν να θέλουμε να γράψουμε σ' αυτόν την τιμή 0.

Έτσι αν η αρχική κατάσταση του πυρήνα ήταν 1, θά μεταβληθεί και θά γίνει 0, αν δέ ήταν αρχικά 0 θά παραμείνει αμετάβλητη. Για να μη καταστραφεί όμως η αρχική του κατάσταση, προστίθεται στον πυρήνα και ένας τρίτος άγωγός, που ονομάζεται *άγωγός αίσθησεως* (Sense wire) (σχ. 4.3). Ο άγωγός αυτός περνά κάθετα μέσα από όλους τους πυρήνες ενός επιπέδου (δηλαδή υπάρχει ένας άγωγός αίσθησεως για κάθε επίπεδο).



Σχ. 4.3.

Ο άγωγός αίσθησεως (Sense Wire).

Στόν άγωγό αίσθησεως δημιουργούνται ρεύματα από έπαγωγή κάθε φορά που μεταβάλλεται η μαγνητική κατάσταση ενός πυρήνα του επιπέδου. Άρα, αν ο πυρήνας, που διαβάζουμε, βρίσκεται αρχικά στην κατάσταση 1, με τη διαδικασία της αναγνώσεως θά μεταβάλλει τη μαγνητική του κατάσταση σε 0, γεγονός που θά προκαλέσει τη δημιουργία έπαγωγικού ρεύματος στόν άγωγό αίσθησεως με φορά αντίθετη προς τη φορά της μεταβολής (κανόνας του Lenz) που με τη σειρά του θά έπαναφέρει τον πυρήνα και πάλι στην κατάσταση 1.

Αν ο πυρήνας βρίσκεται αρχικά στην κατάσταση 0, τότε με τη διαδικασία της αναγνώσεως δέν θά υπάρξει καμιά μεταβολή, ο άγωγός αίσθησεως δέν θά διαρρέεται από ρεύμα και ο πυρήνας θά διατηρήσει την κατάστασή του, δηλαδή την τιμή 0.

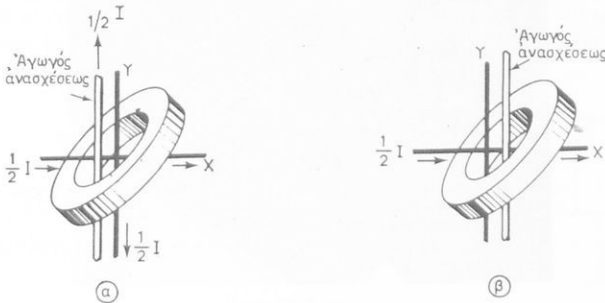
Μέ άλλα λόγια, όταν ο άγωγός αίσθησεως διαρρέεται από ρεύμα, σημαίνει ότι η τιμή του χαρακτήρα που διαβάζεται είναι 1, ενώ αν δέν διαρρέεται από ρεύμα, σημαίνει ότι η τιμή του είναι 0.

1 Άπο εδώ και πέρα κάθε κωδικοποιημένο δυαδικό χαρακτήρα θά τόν αναφέρουμε με τόν αγγλικό δυο byte. Φυσικά τό μέγεθος ενός byte εξαρτάται από τόν κώδικα που χρησιμοποιούμε για τη δυαδική απεικόνιση των χαρακτήρων. Στο μέγεθος του byte δέν υπολογίζεται τό ψηφίο της ίσοτιμίας (Parity bit).

4.4 Έγγραφή μιάς τιμής σέ ένα πυρήνα.

Η διαδικασία έγγραφης μιάς τιμής σέ κάποιο πυρήνα είναι πió πολýπλοκη καί περιλαμβάνει είτε τήν έπαναποθήκευση τής αρχικής τιμής, είτε τήν αποθήκευση μιάς νέας τιμής διαφορετικής από τήν αρχική.

Στήν περίπτωση τής έγγραφης στέλνουμε από τούς άγωγούς Χ καί Υ ρεύματα έντάσεως $1/2 I$ μέ φορές τέτοιες, σάν νά θέλαμε νά δώσουμε στόν πυρήνα αυτό τιμή 1. Έτσι αν ο πυρήνας είχε προηγούμενα τιμή 1 καί θέλουμε νά τή διατηρήσουμε μέ τίς φορές πού έχουν τά ρεύματα, ή τιμή αυτή θά διατηρηθεί. Αν είχε αρχικά τιμή 0 καί θέλουμε νά γίνει 1, τότε καί πάλι οι φορές τών ρευμάτων εξασφαλίζουν αυτή τήν αλλαγή. Υπάρχει όμως τό ένδεχόμενο ο πυρήνας νά είχε αρχικά τιμή 0 καί νά θέλουμε νά τή διατηρήσουμε, ενώ μέ τή διαδικασία τής έγγραφης ή τιμή 0 θά καταστραφεί. Για νά μή συμβεί αυτό εισάγεται μέσα στόν πυρήνα καί ένας τέταρτος άγωγός παράλληλα πρός τόν Υ, πού όνομάζεται *άγωγός άνασχέσεως* (Inhibit wire) (σχ. 4.4). Ο άγωγός αυτός περνά μέσα από όλους τούς πυρήνες ενός επιπέδου.



Σχ. 4.4.

α. Ο άγωγός άνασχέσεως διαρρέεται από ρεύμα. β. Ο άγωγός άνασχέσεως δέν διαρρέεται από ρεύμα.

Ένεργοποιείται μόνο στήν περίπτωση πού κάνουμε έγγραφή καί θέλουμε νά διαφυλάξουμε τήν τιμή 0 ενός πυρήνα, ώστε νά μή γίνει 1 μέ τή διαδικασία τής έγγραφης. Κάθε φορά πού ο άγωγός άνασχέσεως ένεργοποιείται, διαρρέεται από ρεύμα έντάσεως $1/2 I$ μέ φορά αντίθετη πρός τή φορά του Υ, ώστε νά έξουδετερώνει τό ρεύμα του καί έτσι νά διατηρείται ή τιμή 0.

4.5 Μνήμες από ήμιαγωγούς.

Μία μνήμη ήμιαγωγών αποτελείται από πολλά ολοκληρωμένα μικροκυκλώματα (Chips), πού συνδέονται μεταξύ τους μέ διακόπτες. Τό κύριο συστατικό ενός τέτοιου μικροκυκλώματος είναι ήμιαγωγοί τύπου MOSFET.

Κάθε ολοκληρωμένο μικροκύκλωμα έχει χωρητικότητα 1024 bits.

Η μνήμη ήμιαγωγού είναι πió συμπαγής (Compact) καί πió γρήγορη από τή μνήμη μαγνητικών πυρήνων.

4.6 Άλλα είδη μνήμης.

Έκτός από τίς μνήμες μέ μαγνητικούς πυρήνες χρησιμοποιούνται επίσης καί μνήμες από λεπτά *ύμένα* (Thin films memories) (σχ. 4.6), από λεπτά *πεπλατυσμένα σύρματα* (Plated wire memories), *φωτοψηφιακές* μνήμες (Photodigital memo-



Σχ. 4.6.

Μνήμες από λεπτά υμένια.

ries), *ηλεκτροοπτικές* (Electrooptical memories) κλπ. πού η λειτουργία τους στηρίζεται επάνω σε διάφορα φαινόμενα της φυσικής, όπως π.χ η ηλεκτρική ή οπτική πόλωση κ.ο.κ.

4.7 Όργάνωση της μνήμης.

α) *Η έννοια της διευθύνσεως.*

Στήν κεντρική μνήμη, όπως είπαμε, αποθηκεύονται έντολές και δεδομένα σε δυαδική μορφή. Για να είναι η μνήμη χρήσιμη, θα πρέπει να υπάρχει κάποιος τρόπος, ώστε να μπορούμε να αποθηκεύουμε νέα δεδομένα ή να ανασύρουμε και να επεξεργαζόμαστε δεδομένα, πού έχουν ήδη αποθηκευθεί.

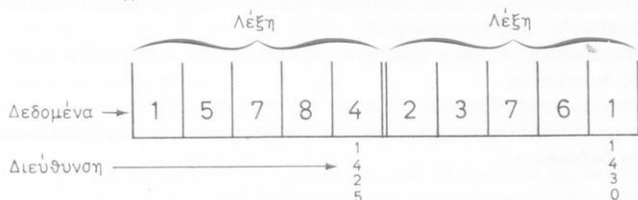
Θά πρέπει λοιπόν να γνωρίζουμε σε ποιές ακριβώς θέσεις έχουν τοποθετηθεί οι διάφορες πληροφορίες. Για τό σκοπό αυτό εφοδιάζουμε τή μνήμη με διευθύνσεις. Ωστε η *διεύθυνση μνήμης* (memory address) είναι ένας αριθμός, πού καθορίζει μονοσήμαντα μία περιοχή της μνήμης, όπου αποθηκεύεται μία στοιχειώδης πληροφορία και από όπου μπορεί να ανασυρθεί άργότερα, για να χρησιμοποιηθεί σε κάποια επεξεργασία.

Οι τιμές των διευθύνσεων μεταβάλλονται με συνεχή τρόπο, αρχίζοντας από το 0. Η διεύθυνση μνήμης μπορεί να παρομοιασθεί με τη διεύθυνση ενός σπιτιού, που η θέση του καθορίζεται μονοσήμαντα, όταν δοθούν ο δρόμος και ο αριθμός του.

β) Υπολογιστές με λέξεις σταθερού ή μεταβλητού μήκους.

Μία λέξη υπολογιστή (Word) είναι σύνολο χαρακτήρων, που ο υπολογιστής το χειρίζεται σαν μία μονάδα. Ο αριθμός των χαρακτήρων σε μία λέξη μπορεί να είναι σταθερός ή μεταβλητός.

Στους υπολογιστές που έχουν λέξεις σταθερού μήκους (Fixed word length) δίνεται μία ξεχωριστή διεύθυνση σε κάθε περιοχή της μνήμης που αντιστοιχεί σε μία λέξη (σχ. 4.7α). Το σύνολο αυτό των χαρακτήρων θεωρείται ενιαίο κατά την επεξεργασία του (μετακίνηση από περιοχή σε περιοχή, τοποθέτηση σε καταχωριστές κλπ.). Πιο συνηθισμένοι είναι οι υπολογιστές των 24, 30, 32, 36, 48 και 54 bits. Σημειώνουμε ότι ένας χαρακτήρας στον άλφαριθμητικό κώδικα BCD έχει 6 bits, ενώ στον EBCDIC έχει 8 bits.



Σχ. 4.7α.

Λέξεις σταθερού μήκους. Το σταθερό μήκος είναι εδώ 5 χαρακτήρες. Η διεύθυνση 1425 δόθηκε στη λέξη που περιέχει τους χαρακτήρες 15784, η διεύθυνση 1430 δόθηκε στη λέξη που περιέχει τους χαρακτήρες 23761.



Σχ. 4.7β.

Λέξεις μεταβλητού μήκους.

Στους υπολογιστές με λέξεις μεταβλητού μήκους (Variable word length) το πλήθος των χαρακτήρων μιας λέξεως είναι μεταβλητό. Κάθε χαρακτήρας έχει τη δική του διεύθυνση (σχ. 4.7β) και μπορεί να ανασυρθεί από τη μνήμη και να πάρει μέρος σε επεξεργασία μόνος του. Η διεύθυνση μιας λέξεως είναι η διεύθυνση του χαρακτήρα με τον οποίο αρχίζει ή καταχώριση της λέξεως στη μνήμη.

4.8 Καταχωριστές (Registers).

Καταχωριστής είναι μία μικρή μνήμη, που χρησιμοποιείται για την προσωρινή αποθήκευση πληροφοριών. Τό μέγεθός του κυμαίνεται συνήθως από 1-4 λέξεις. Καταχωριστές υπάρχουν στη μονάδα έλεγχου και στην αριθμητική μονάδα, όπου και αποτελούν κύρια λειτουργικά τμήματά της.

Υπάρχουν διάφορα είδη καταχωριστών, που έχουν ονομασίες ανάλογες με το σκοπό που εξυπηρετούν. Οι κυριότεροι είναι:

- α) Άπλός καταχωριστής (storage register), όπου αποθηκεύονται απλές πληροφορίες, που μεταφέρονται από τη μνήμη.
- β) Καταχωριστής εντολών (instruction register), όπου μεταφέρονται οι εντολές ενός προγράμματος με τη σειρά που εκτελούνται.
- γ) Καταχωριστής διευθύνσεως (address register), όπου καταχωρίζεται η διεύθυνση της μνήμης, όπου βρίσκεται η επόμενη να εκτελεσθεί εντολή.
- δ) Καταχωριστής συσσωρευτής ή άθροιστικός καταχωριστής (accumulator register), όπου καταχωρίζεται τό άθροισμα δύο αριθμών.
- ε) Καταχωριστής γινομένου (multiplier register), όπου καταχωρίζεται τό αποτέλεσμα ενός πολλαπλασιασμού.
- ζ) Καταχωριστής πηλίκου (quotient register), όπου καταχωρίζεται τό πηλίκο μίας διαιρέσεως.
- η) Καταχωριστής άθροίσματος (adder register), όπου εκτελείται μία πρόσθεση.
- θ) Καταχωριστής συγκρίσεως (comparison register), όπου γίνεται ή σύγκριση δύο δεδομένων.

4.9 Φλίπ-Φλόπς (Flip-Flops).

Φλίπ-φλόπ είναι ένας ηλεκτρονικός διακόπτης, τό κύκλωμα του οποίου αποτελείται βασικά από δύο τρίοδες ηλεκτρονικές λυχνίες ή δύο ήμιαγωγούς (κρυσταλλοτρίοδους) και περιέχει ένα τμήμα εισόδου για την είσαγωγή του παλμού διεγέρσεως του κυκλώματος και ένα τμήμα εξόδου με δύο άκροδέκτες. Κατάλληλοι ηλεκτρικοί παλμοί, που εισάγονται από τό τμήμα εισόδου, άλλοτε προκαλούν την εμφάνιση σήματος από τόν ένα άκροδέκτη της εξόδου και άλλοτε από τόν άλλο. Η ύπαρξη σήματος από τόν ένα άκροδέκτη της εξόδου αντιστοιχεί στο δυαδικό ψηφίο 1, ενώ ή άνυπαρξία σήματος αντιστοιχεί στο δυαδικό ψηφίο 0.

Υπάρχουν διαφόρων τύπων φλίπ-φλόπς, όπως τό φλίπ-φλόπ δύο σταθερών καταστάσεων (bistable), μιάς σταθερής καταστάσεως (monostable) και τό ελεύθερης ταλαντώσεως (astable).

Στά φλίπ-φλόπς δύο σταθερών καταστάσεων, ή αρχική κατάσταση των άκροδεκτών εξόδου, μετά ή στιγμιαία διεγερση του κυκλώματος από κάποιο έξωτερικό παλμό, μπορεί νά μεταβληθεί. Η επάνοδος στην αρχική κατάσταση γίνεται μόνο με την εφαρμογή ενός νέου κατάλληλου παλμού διεγέρσεως. Αν δέν υπάρξει νέος παλμός διεγέρσεως ή κατάσταση των άκροδεκτών παραμένει συνεχώς σταθερή.

Τά φλίπ-φλόπς μιάς σταθερής καταστάσεως όταν διεγερθούν παραμένουν για όρισμένο χρονικό διάστημα στή νέα κατάσταση και κατόπιν επανέρχονται στην αρχική.

Τέλος τά φλίπ-φλόπς ελεύθερης ταλαντώσεως διεγείρονται περιοδικά από κά-

ποιο εξωτερικό σήμα που παρέχει ένας βασικός ταλαντωτής και η κατάσταση των άκροδεκτών εξόδου τους αναγκάζεται να εναλλάσσεται συνεχώς με συχνότητα ίση με τη συχνότητα του σήματος διεγέρσεως.

Πολλά φλίπ-φλόπς, συνδεδεμένα κατάλληλα μπορούν λόγω του τρόπου λειτουργίας τους να χρησιμοποιηθούν στις μνήμες των υπολογιστών καθώς επίσης και σαν άπλοι καταχωριστές (registers) ή σαν μετρητές (counters).

4.10 Μονάδα έλεγχου (Control Unit).

Είναι το πιο σύνθετο από όλα τα τμήματα του υπολογιστή. Η μονάδα αυτή είναι επιφορτισμένη με το συντονισμό και τον έλεγχο της λειτουργίας όλων των άλλων μονάδων, έτσι ώστε να ακολουθείται η λογική σειρά των οδηγιών-έντολών που έχουν δοθεί.

Διαθέτει μία σειρά από καταχωριστές εντολών και διευθύνσεων και βρίσκεται σε συνεχή επικοινωνία με την αριθμητική μονάδα, την κεντρική μνήμη και τις περιφερειακές μονάδες.

Στόν καταχωριστή εντολών τοποθετείται κάθε φορά, φερμένη από τη μνήμη, η εντολή που πρόκειται να εκτελεσθεί, αφού προηγουμένως αποκωδικοποιηθεί, ενώ στόν καταχωριστή διευθύνσεων, τοποθετείται κάθε φορά η διευθύνση της επόμενης να εκτελεσθεί εντολής.

4.11 Αριθμητική και λογική μονάδα (Arithmetic and logical unit).

Η μονάδα αυτή εκτελεί πολλές λειτουργίες. Με τη βοήθειά της εκτελούνται οι αριθμητικές πράξεις επάνω στα δεδομένα, που μεταφέρονται σε αυτήν από τη μνήμη. Εκτελεί ακόμα λογικές συγκρίσεις και είναι επιφορτισμένη με το χειρισμό των δεδομένων (Data handling).

Στους περισσότερους υπολογιστές η αριθμητική μονάδα περιλαμβάνει μία σειρά από ειδικούς καταχωριστές, όπως άπλους ή άθροιστικούς, και καταχωριστές συγκρίσεως ή αποτελεσμάτων (γινομένου, πηλίκου, διαφοράς, άθροίσματος) κλπ.

4.12 Δίαυλοι (Channels).

Οι δίαυλοι είναι συσκευές που συνδέουν τις περιφερειακές μονάδες με την κεντρική μονάδα επεξεργασίας. Μέσω των διαύλων, μεταβιβάζονται από την κεντρική μονάδα επεξεργασίας, σήματα που ρυθμίζουν και ελέγχουν τη λειτουργία των περιφερειακών μονάδων, όπως επίσης και τη ροή των πληροφοριών από και προς αυτές.

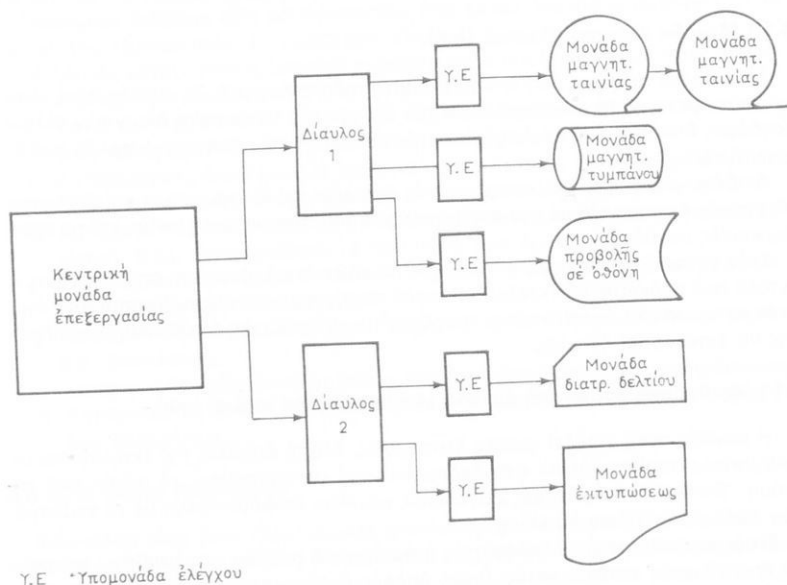
Ένας δίαυλος μπορεί να εξυπηρετεί πολλές περιφερειακές μονάδες (σχ. 4.12).

4.13 Υπομονάδες έλεγχου των περιφερειακών (Controlllers).

Η υπομονάδα έλεγχου συγχρονίζει την άργή λειτουργία μιας περιφερειακής μονάδας με την πολύ γρηγορότερη της κεντρικής μονάδας επεξεργασίας. Για το σκοπό αυτό δέχεται σήματα έλεγχου από το δίαυλο, ελέγχει το χρονισμό της μεταβίβασης των πληροφοριών και δίνει πληροφορίες για τη λειτουργική κατάσταση

(Status) της μονάδας. 'Επί πλέον δημιουργεί καί ἐλέγχει τὰ ψηφία ἰσοτιμίας.

Σέ πολλές περιφερειακές μονάδες, ὅπως π.χ ἡ ἀναγνωστική δελτίων, ἡ ὑπομονάδα ἐλέγχου εἶναι ἐνσωματωμένη σ' αὐτή. Σέ ἄλλες πάλι περιπτώσεις ἐμφανίζονται ὡς ἀνεξάρτητες συσκευές. Σέ κάθε ὑπομονάδα ἐλέγχου μποροῦν νά συνδεθοῦν μέχρι 8 ὁμοειδεῖς περιφερειακές μονάδες (σχ. 4.12).



Σχ. 4.12.

Σύνδεση τῶν περιφερειακῶν μονάδων μέ τή CPU μέσω τῶν διαύλων καί τῶν ὑπομονάδων ἐλέγχου.

Ἑρωτήσεις

1. Περιγράψατε μέ συντομία τόν τρόπο ἀναγνώσεως τῆς τιμῆς ἐνός πυρήνα.
2. Ὑποθέσατε ὅτι σέ ἓνα πυρήνα, πού βρίσκεται σέ κατάσταση 0, θέλομε νά ἐγγράψομε τήν τιμή 1. Ποιά διαδικασία θά ἀκολουθήσομε;
3. Ἀναφέρατε τοὺς κυριότερους τύπους καταχωριστῶν καί τό σκοπό πού ἐξυπηρετεῖ ὁ καθένας.
4. Ποίος ὁ ρόλος τῆς μονάδας ἐλέγχου σέ ἓνα συγκρότημα ὑπολογιστῆ;
5. Ποίος ὁ ρόλος τῆς ἀριθμητικῆς μονάδας σέ ἓνα συγκρότημα ὑπολογιστῆ;
6. Τί εἶναι οἱ δίαυλοι (Channels);
7. Τί εἶναι οἱ ὑπομονάδες ἐλέγχου (Controllers);

ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

5.1 Έντολές — Πρόγραμμα — Γλώσσα ύπολογιστή.

Οι ύπολογιστές είναι αυτόματες ύπολογιστικές μηχανές, πού σημαίνει ότι μπορούν νά έκτελουν μιά όλόκληρη σειρά άπό λειτουργίες χωρίς τήν ανθρώπινη παρέμβαση. Πραγματικά, ό ύπολογιστής μπορεί νά έκτελεϊ χιλιάδες πράξεις χωρίς καμιά έξωτερική βοήθεια άπό τόν άνθρωπο. Για νά τό πετύχει όμως χρειάζεται νά έχει πάρει άπό αυτόν τίς κατάλληλες οδηγίες. Οι οδηγίες αυτές ονομάζονται έντολές (Instructions).

Κάθε έντολή αντιπροσωπεύει μιά ξεχωριστή λειτουργία. Οι έντολές πού θά έκτελέσει ό ύπολογιστής προετοιμάζονται προσεκτικά καί γράφονται έτσι, ώστε νά αντιπροσωπεύουν τή λογική σειρά, πού θά ακολουθήσει για νά έπιλύσει τό συγκεκριμένο πρόβλημα. Άποθηκεύονται στή μνήμη προκαταβολικά.

Όλόκληρη ή σειρά τών έντολών πού άπαιτούνται για νά έπιλυθεί ένα πρόβλημα, άποτελεϊ τό λεγόμενο *πρόγραμμα* (Program).

Οι έντολές διατυπώνονται μέ καθορισμένο τρόπο έτσι, ώστε νά τίς «καταλάβει» ό ύπολογιστής. Ένα πλήρες ρεπερτόριο άπό έντολές άποτελεϊ μιά *γλώσσα προγραμματισμού*.

5.2 Γλώσσα μηχανής καί συμβολική γλώσσα.

Βασικά ένας ύπολογιστής ύπακούει μόνο στή δική του γλώσσα, τή λεγόμενη *γλώσσα μηχανής* (Machine language). Μιά τέτοια γλώσσα περιλαμβάνει έντολές κωδικοποιημένες δυαδικά. Όπως θά δούμε μέ περισσότερες λεπτομέρειες στήν § 5.4, μιά έντολή άποτελεϊται βασικά άπό δύο τμήματα, τόν *κώδικα λειτουργίας* καί τίς *διευθύνσεις μνήμης* τών στοιχείων-δεδομένων, πού συμμετέχουν στή λειτουργία. Καί ό κώδικας καί οι διευθύνσεις έκφράζονται στό δυαδικό σύστημα. Μέ τή μορφή αυτή οι έντολές είναι άμέσως κατανοητές άπό τόν ύπολογιστή. Οι περισσότεροι ύπολογιστές τής πρώτης γενιάς, για τήν όποία μιλήσαμε ήδη, προγραμματίζονταν μόνο σέ γλώσσα μηχανής. Έξ αίτίας όμως τής δυαδικής πολυψηφιακής παραστάσεώς τους ή άπομνημόνευση καί ό προγραμματισμός είναι πολύ δύσκολος.

Σάν πρώτο βήμα άπλουστεύσεως έχομε τήν άντιστοίχιση τών κωδικών έντολών καί τών διευθύνσεων μέ δεκαδικούς άριθμούς. Στόν πίνακα 5.2.1, βλέπομε μιά μορφή κωδικοποίησης τών λειτουργιών ενός ύποθετικού ύπολογιστή.

Παρά όλο όμως πού ή κωδικοποίηση αυτή είναι βελτιωμένη, ό προγραμματισμός

έξακολουθεῖ νά ἀποτελεῖ δυσκολότατη ἐργασία. Μεγαλύτερη ἀπλούστευση εἶναι δυνατή μέ τήν ἀντιστοίχιση γραμμάτων ἤ καί μικρῶν λέξεων στούς ἀριθμούς, πού συμβολίζουν τούς κωδικούς ἐντολῶν (βλέπε πίνακα 5.2.1), ἐνῶ οἱ διευθύνσεις παραμένουν στή δεκαδική ἀριθμητική τους ἔκφραση.

Καί μέ τή μορφή αὐτή ἀκόμα οἱ ἐντολές θεωροῦμε ὅτι ἐκφράζουν τή γλώσσα μηχανῆς.

ΠΙΝΑΚΑΣ 5.2.1.

Λειτουργία	Δυαδική ἔκφραση	Δεκαδική ἔκφραση	Συμβολική ἔκφραση
Πρόσθεση	001010	10	A
Ἀφαίρεση	001011	11	S
Πολλαπλασιασμός	001100	12	X
Διαίρεση	001101	13	D
Ἄλμα	001111	15	J
Ἐγγραφή στόν ἀθροιστικό καταχωριστή Ac ¹	010000	16	R
Ἀνάγνωση ἀπό τόν ἀθροιστικό καταχωριστή Ac	010001	17	C
Ἐκτύπωση	010010	18	W
Ἐλεγχος θετικοῦ προσήμου	010100	20	P
Ἐλεγχος ἀρνητικοῦ προσήμου	010101	21	M
Ἐλεγχος μηδενικοῦ περιεχομένου	010110	22	Z
Διακοπή	011001	25	H

1 Τόν ἀθροιστικό καταχωριστή ἢ συσσωρευτή (Accumulator Register) θά τόν συμβολίζομε ἀπό δῶ καί πέρα μέ Ac.

Γιά μεγαλύτερη ἀκόμα ἀπλούστευση χρησιμοποιήθηκαν γράμματα στή θέση τῶν δεκαδικῶν ἀριθμῶν γιά τόν συμβολισμό τῶν διευθύνσεων. Ἐτσι δημιουργήθηκαν οἱ λεγόμενες *συμβολικές γλώσσες* (Assembly Languages) μέ τή βοήθεια τῶν ὁποίων προγραμματίζονταν οἱ περισσότερες μηχανές τῆς δεύτερης γενιᾶς.

Οἱ συμβολικές γλώσσες διακρίνονται σέ δύο κατηγορίες:

- Σέ ἐκεῖνες, ὅπου σέ κάθε ἐντολή τῆς συμβολικῆς γλώσσας ἀντιστοιχεῖ μία ἀκριβῶς ἐντολή τῆς γλώσσας μηχανῆς.
- Σέ ἐκεῖνες, ὅπου μία ἐντολή συμβολικῆς γλώσσας ἀντιστοιχεῖ σέ ἕνα σύνολο ἀπό ἐντολές σέ γλώσσα μηχανῆς καί πού ὀνομάζονται *μακροεντολές* (macros).

Ἡ μετατροπή ἐνός προγράμματος, πού ἔχει γραφεῖ σέ συμβολική γλώσσα, στή γλώσσα μηχανῆς γίνεται αὐτόματα ἀπό τή μηχανή μέ τή βοήθεια εἰδικοῦ προγράμματος ἀντιστοίχιας, τοῦ λεγόμενου *μεταφραστικοῦ προγράμματος* (Assembler ἢ Assembly compiler), μέ τό ὁποῖο ἡ κατασκευάστρια ἐταιρία ἐφοδιάζει τό συγκεκριμένο ὑπολογιστή.

Τέτοιες συμβολικές γλώσσες ὑπάρχουν ἀρκετές μέ διάφορες παραλλαγές, ὅπως ἡ BAL (Basic Assembly Language), ἡ COMPASS, ἡ SIGMA κ.ο.κ.

Παρ' ὅλο πού οἱ γλώσσες μηχανῆς καί οἱ συμβολικές γλώσσες κάνουν πῶς ἀποδοτική τήν ἐκμετάλλευση τῆς μηχανῆς, ἐν τούτοις εἶναι γλώσσες προσαρμοσμένες στή μηχανή (Machine Oriented Languages). Αὐτό ὑποχρεώνει τόν προγραμματιστή

στή νά περιορίζει τή λειτουργικότητα ενός προγράμματός του σέ συγκεκριμένα συστήματα ή νά τόν αναγκάζει νά ξαναγράψει τό πρόγραμμα σέ καινούργια γλώσσα, ἂν θά ἤθελε νά τό χρησιμοποιήσει σέ διαφορετικό σύστημα.

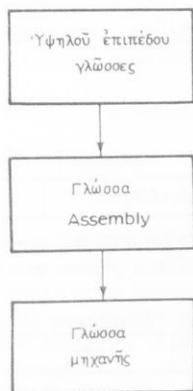
5.3 Γλώσσες ὑψηλοῦ ἐπιπέδου (High Level Languages).

Ἡ δημιουργία πολλῶν τύπων ὑπολογιστῶν ἀνάγκασε τοὺς εἰδικούς νά σχεδιάσουν νέες προηγμένες διεθνεῖς γλώσσες συνεννοήσεως μέ τόν ὑπολογιστή, οἱ ὁποῖες εἶναι ἀνεξάρτητες ἀπό τόν τύπο τῆς μηχανῆς (Machine independent), ἀλλά προσαρμοσμένες στίς ἀπαιτήσεις ενός συγκεκριμένου τύπου προβλημάτων (Problem Oriented Languages). Οἱ γλώσσες αὐτές, πού μποροῦν νά χρησιμοποιηθοῦν ἀπό πολλοὺς διαφορετικούς τύπους (τῆς ἴδιας ἢ διαφορετικῶν κατασκευαστριῶν ἐταιριῶν), χαρακτηρίζονται καί ὡς *γλώσσες ὑψηλοῦ ἐπιπέδου*.

Χωρίζονται σέ δύο κυρίως κατηγορίες. Στὴν πρώτη ἀνήκουν ἐκεῖνες, πού χρησιμοποιοῦνται στὴν ἐπίλυση ἐπιστημονικῶν καί τεχνικῶν προβλημάτων, ὅπου δηλαδή κυριαρχεῖ ἡ χρῆση τῶν μαθηματικῶν. Τέτοιες γλώσσες εἶναι ἡ FORTRAN, ἡ ALGOL, ἡ BASIC κλπ. Στὴν ἄλλη κατηγορία ἀνήκουν οἱ γλώσσες πού χρησιμοποιοῦνται στὴν ἐπίλυση ἐμπορικῶν καί οἰκονομικῶν προβλημάτων (λογιστήρια, ἀποθήκες, μισθοδοσίες κλπ). Τέτοιες γλώσσες εἶναι ἡ COBOL, ἡ RPG, ἡ PL1 κλπ.

Ἐπειδὴ ὅμως ὁ ὑπολογιστής, ὅπως εἴπαμε, καταλαβαίνει μόνο τὴ δική του γλώσσα, τὰ προγράμματα πού εἶναι γραμμένα σέ κάποια ἀπὸ τίς γλώσσες ὑψηλοῦ ἐπιπέδου (Source programs) μεταφράζονται στὴ γλώσσα μηχανῆς (Object programs) ἀπὸ τὸν ἴδιο τόν ὑπολογιστή, μέ τὴ βοήθεια εἰδικῶν μεταφραστικῶν προγραμμάτων (Compilers), μέ τὰ ὁποῖα οἱ κατασκευάστριες ἐταιρίες ἐφοδιάζουν τίς μηχανές τους.

Τὴν ἱεραρχία τῶν γλωσσῶν προγραμματισμοῦ βλέπομε στὸ σχῆμα 5.3.



Σχ. 5.3.

Ἡ ἱεραρχία τῶν γλωσσῶν προγραμματισμοῦ.

5.4 Κώδικες λειτουργίας και διευθύνσεις.

Μιά έντολή αποτελείται, τς περισσότερες φορές, από δύο κύρια τμήματα:

α) Τόν κώδικα λειτουργίας ή έντολης (Operation Code).

β) Τίς διευθύνσεις (Operands).

Ο κώδικας λειτουργίας καθορίζει τή συγκεκριμένη λειτουργία πού πρόκειται νά εκτελεσθεῖ, ἐνῶ οἱ διευθύνσεις παριστάνουν τίς διευθύνσεις τῶν θέσεων τῆς μνήμης, τὰ περιεχόμενα τῶν ὁποίων θά συμμετάσχουν στή λειτουργία αὐτή. Τό περιεχόμενο μιᾶς διευθύνσεως Ad θά συμβολίζομε ἀπό δῶ καί πέρα μέ c (Ad) ἢ ἀπλῶς [Ad], ἐνῶ τό περιεχόμενο τοῦ συσσωρευτῆ¹ μέ c (Ac) ἢ ἀπλῶς [Ac].

Τίς ἐντολές τίς διακρίνομε ἀνάλογα μέ τό πλῆθος τῶν διευθύνσεων πού ἐμφανίζονται στό 2ο τμήμα τους, ὡς ἐντολές μιᾶς, δύο ἢ τριῶν διευθύνσεων. Τό πλῆθος τῶν στοιχείων, πού διατίθεται γιά τήν καταχώριση τοῦ κώδικα ἐντολῆς καί τῶν διευθύνσεων, ἐξαρτᾶται ἀπό τόν τύπο τοῦ ὑπολογιστῆ.

α) Ἐντολή μιᾶς διευθύνσεως (One Address Instruction).

Ἐχει τή γενική μορφή

Κώδικας ἐντολῆς	Διεύθυνση
-----------------	-----------

Στήν ἐκτέλεση μιᾶς ἐντολῆς αὐτοῦ τοῦ τύπου, ἐκτός ἀπό τό περιεχόμενο τῆς διευθύνσεως τῆς μνήμης, μετέχει ἐνεργά καί ὁ συσσωρευτής μέ τό περιεχόμενό του. Ἐτσι π.χ. στήν περίπτωση μιᾶς ἀπλῆς προσθέσεως ἡ ἐντολή θά ἔχει τή μορφή

Κώδικας ἐντολῆς	Διεύθυνση
10	1472

$$c(Ac) = c(Ac) + c(1472)$$

καί σημαίνει ὅτι προστίθεται τό περιεχόμενο τῆς διευθύνσεως 1472 στό περιεχόμενο τοῦ συσσωρευτῆ καί τό ἄθροισμα τίθεται πάλι στό συσσωρευτή. Ὁ κώδικας τῆς προσθέσεως εἶναι 10.

β) Ἐντολή δύο διευθύνσεων.

Ἐχει τή γενική μορφή

Κώδικας	Διεύθυνση 1	Διεύθυνση 2
---------	-------------	-------------

Στήν ἐντολή αὐτή δίνονται οἱ διευθύνσεις τῶν δύο στοιχείων πού θά πάρουν μέρος σέ μιᾶ πράξη. Τό ἀποτέλεσμα μπαίνει στό συσσωρευτή. Π.χ.:

Κώδικας ἐντολῆς	Διεύθυνση 1	Διεύθυνση 2
11	1652	7212

$$c(Ac) = c(1652) - c(7212)$$

¹ Ὑπενθυμίζομε ὅτι ὁ συσσωρευτής εἶναι ἓνας καταχωριστής (register), πού βρίσκεται στήν ἀριθμητική μονάδα.

καί σημαίνει ότι αφαιρείται (κώδικας έντολης = 11) τό $c(7212)$ από τό $c(1652)$ καί ή διαφορά καταχωρίζεται στό συσσωρευτή.

γ) Έντολή τριῶν διευθύνσεων.

Έχει τή γενική μορφή:

Κώδικας έντολης	Διεύθυνση 1	Διεύθυνση 2	Διεύθυνση 3
-----------------	-------------	-------------	-------------

Στήν περίπτωση αὐτή δίνονται οἱ διευθύνσεις τῶν δύο στοιχείων, πού συμμετέχουν στή λειτουργία πού περιγράφει ὁ κώδικας έντολης (διεύθυνση 1 καί διεύθυνση 2) καί ή διεύθυνση ὅπου θά τοποθετηθεῖ τό ἀποτέλεσμα (δευθύνση 3). Π.χ.:

Κώδικας έντολης	Διεύθυνση 1	Διεύθυνση 2	Διεύθυνση 3
10	3046	7923	8561

$$c(8561) = c(3046) - c(7923)$$

καί σημαίνει ότι αφαιρείται ἀπό τό περιεχόμενο τῆς διευθύνσεως 3046 τό περιεχόμενο τῆς διευθύνσεως 7923 καί τό ἀποτέλεσμα καταχωρίζεται στή διεύθυνση 8561.

5.5 Μηχανισμός ἐκτελέσεως ἑνός προγράμματος.

“Ὅπως εἶπαμε στήν παράγραφο 4.10 στή μονάδα ἐλέγχου ὑπάρχει ἕνας καταχωριστής έντολῶν, πού τόν συμβολίζομε συνήθως μέ IR (Instruction Register), στόν ὁποῖο μεταφέρεται ἀπό τή μνήμη ή έντολή πού πρόκειται κάθε φορά νά ἐκτελεσθεῖ, καί ἕνας καταχωριστής διευθύνσεων, ὅπου καταχωρίζεται ή διεύθυνση τῆς μνήμης, πού βρίσκεται ἀποθηκευμένη ή ἐπόμενη νά ἐκτελεσθεῖ έντολή.

“Ἄς δοῦμε τώρα πῶς λειτουργεῖ ὁ μηχανισμός ἐκτελέσεως ἑνός προγράμματος σέ ἕνα ὑπολογιστή δύο διευθύνσεων μέ ἕνα ἀπλό παράδειγμα.

“Ἐστω ὅτι θέλομε νά προσθέσομε τοὺς ἀριθμούς 524 καί 242, πού ἔχουν διατυπηθεῖ ἐπάνω σέ ἕνα δελτίο. Τό πρόγραμμα θά ἀποτελεῖται ἀπό τίς ἐξῆς έντολές (σέ περιγραφική μορφή):

- Διάβασε ἀπό τό δελτίο τοὺς δύο ἀριθμούς.
- Κάνε τήν πρόσθεση.
- Μετάφερε τό ἀποτέλεσμα στή μνήμη.
- Γράψε τό ἀποτέλεσμα.
- Σταμάτησε.

“Ἄς ὑποθέσομε ὅτι τό κωδικοποιημένο αὐτό πρόγραμμα τό ἔχομε ἀποθηκεύσει στίς θέσεις τῆς μνήμης μέ διευθύνσεις 101, 102, 103, 104 καί 105. Μόλις δοθεῖ τό σῆμα γιά τήν ἐκκίνηση, ὁ καταχωριστής διευθύνσεων ἔχει τιμή 101 καί ή πρώτη έντολή μεταφέρεται μέσα στόν καταχωριστή έντολῶν (IR) καί ἀρχίζει ή ἐκτέλεσή της. Ἐτσι θά διαβασθεῖ ἕνα δελτίο καί τά δεδομένα θά τοποθετηθοῦν στίς διευθύνσεις 201 καί 202 ἀντίστοιχα:

Έντολή 1η

Διάβασε	Διεύθυνση 1 = 201	Διεύθυνση 2 = 202
---------	-------------------	-------------------

Ἡ περιοχή τῆς μνήμης ὅπου ἀποθηκεύθηκαν τό πρόγραμμα καί τά δεδομένα θά παρουσιάζει τήν ἐξῆς εἰκόνα (σχ. 5.5).

ΜΝΗΜΗ			
ΔΙΕΥΘΥΝΣΗ	ΠΕΡΙΕΧΟΜΕΝΟ	ΔΙΕΥΘΥΝΣΗ	ΠΕΡΙΕΧΟΜΕΝΟ
101	1η ἐντολή	201	524
102	2η ἐντολή	202	242
103	3η ἐντολή	203	
104	4η ἐντολή		
105	5η ἐντολή		
106			

Σχ. 5.5.

Ἡ κατάσταση τῆς μνήμης.

Παράλληλα ἡ τιμὴ τοῦ καταχωριστῆ διευθύνσεων θά αὐξηθεῖ αὐτόματα κατὰ 1 καί ἔτσι θά μεταφερθεῖ στὸν καταχωριστὴ ἐντολῶν ἡ ἐπόμενη ἐντολή, πού βρίσκεται στῇ διεύθυνση 102 καί ἔχει τή μορφή:

Ἐντολή 2η

Πρόσθεσε	201	202
----------	-----	-----

Μέ τήν ἐκτέλεσή της θά προστεθεῖ τό περιεχόμενο τῆς διευθύνσεως 201 (ὅπου βρίσκεται καταχωρισμένος ὁ ἀριθμὸς 524) στοῦ περιεχόμενου τῆς διευθύνσεως 202 (ὅπου βρίσκεται ὁ 242) καί τό ἀποτέλεσμα θά καταχωρισθεῖ στοῦ συσσωρευτή. Συγχρόνως ἡ τιμὴ τοῦ καταχωριστῆ διευθύνσεων θά γίνεῖ 103 καί θά μεταφερθεῖ ἡ τρίτη ἐντολή στὸν καταχωριστὴ ἐντολῶν, πού ἔχει τή μορφή:

Ἐντολή 3η

Μετέφερε	Ac	203
----------	----	-----

Μέ τήν ἐκτέλεσή της θά μεταφερθεῖ τό ἀποτέλεσμα ἀπὸ τό συσσωρευτὴ στῇ διεύθυνση 203 τῆς μνήμης.

Ἡ τιμὴ τοῦ καταχωριστῆ διευθύνσεων ἐν τῷ μεταξύ ἔχει γίνεῖ 104 καί στὸν καταχωριστὴ ἐντολῶν μεταφέρεται ἡ τέταρτη ἐντολή, πού ἔχει τή μορφή:

Ἐντολή 4η

Γράψε	203	—
-------	-----	---

μέ τήν ὁποία γράφει τό ἀποτέλεσμα, πού βρίσκεται στῇ διεύθυνση 203, στήν περιφερειακὴ μονάδα ἐκτυπώσεως.

Μέ τήν τελευταία ἐντολή, πού θά μεταφερθεῖ στὸν καταχωριστὴ ἐντολῶν γίνε-ται καί ἡ διακοπὴ τῆς λειτουργίας τῆς μηχανῆς.

Ἄν σχεδιάζαμε τό ἴδιο πρόγραμμα γιὰ μιά μηχανή μιᾶς διευθύνσεως, θά ἔπρεπε νά προσθέσουμε καί ἄλλες ἐντολές (τὴν ἐντολή προσθέσεως θά ἀντικαθιστοῦσαν δύο ἐντολές προσθέσεως), ἐνῶ σέ μιά μηχανή τριῶν διευθύνσεων θά εἶχαμε λιγό-

τερες έντολές (ή έντολή προσθέσεως και ή έντολή μεταφοράς θά είχαν αντικατασταθεί από μία και μόνη έντολή).

5.6 Κατηγορίες έντολῶν.

‘Η έκλογή ενός κώδικα έντολῶν απαιτεῖ μεγάλη προσοχή και παρουσιάζει αρκετές τεχνικές δυσκολίες στό σχεδιασμό τῶν καταλλήλων κυκλωμάτων. ‘Από τήν ἄλλη ὁμως πλευρά πρέπει νά λαμβάνεται ὑπ’ ὄψη και ή εὐκολία μέ τήν ὁποία θά μπορεῖ νά προγραμματισθεῖ κάποιο πρόβλημα.

Ποιές ὁμως εἶναι οἱ βασικές έντολές γιά τή λειτουργία ενός προγράμματος; Πολλές ἀπό τίς πρῶτες μηχανές ἐκτελοῦσαν μόνο πρόσθεση, ἀφαίρεση και πολλαπλασιασμό, ἐνῶ ή διαίρεση ἔπρεπε νά προγραμματισθεῖ κατάλληλα. ‘Ολοι οἱ σημερινοί ὑπολογιστές ἐκτελοῦν αὐτόματα και τίς 4 ἀριθμητικές πράξεις.

Οἱ έντολές αὐτές μαζί μέ έντολές γιά μετακίνηση δεδομένων ἀπό τή μνήμη πρός τήν ἀριθμητική μονάδα και ἀντίστροφα, ὅπως ἐπίσης και ἀπό μία περιοχή τῆς μνήμης σέ ἄλλη ἀποτελοῦν τή βάση γιά ὁποιοδήποτε κώδικα. Κάθε μηχανή, ἐξ ἄλλου πρέπει νά διαθέτει ἐπίσης τουλάχιστον δύο έντολές ἐλέγχου, πού θά ἐλέγχουν τήν ἱκανοποίηση ὀρισμένων συνθηκῶν και θά καθορίζουν τή σειρά, μέ τήν ὁποία θά ἐκτελεσθοῦν οἱ διάφορες έντολές ενός προγράμματος, ἀνάλογα μέ τά ἀποτελέσματα τῆς συγκρίσεως.

‘Ο λόγος γι’ αὐτό εἶναι ὅτι πρακτικά οἱ περισσότεροι ὑπολογισμοί, πού πρέπει νά γίνουν, ἐπαναλαμβάνονται πολλές φορές, ὥστε οἱ ἴδιες έντολές νά ἐκτελοῦνται μέ τήν ἴδια σειρά ἀλλά μέ διαφορετικά δεδομένα γιά κάθε ἐπανάληψη. ‘Αν λοιπόν δέν ὑπῆρχαν οἱ έντολές ἐλέγχου, ή χρήση τοῦ ὑπολογιστή θά ἦταν ἐξαιρετικά ἀντοικονομική, ἐφ’ ὅσον ὁ χρόνος γιά τήν προετοιμασία τοῦ προγράμματος θά ὑπερέβαινε τό χρόνο ὑπολογισμοῦ μέ τό χέρι. Θά ἀναφερθοῦμε σέ ἕνα πολύ ἀπλό παράδειγμα.

‘Ας υποθέσουμε ὅτι θέλομε νά προσθέσουμε 100 ἀριθμούς πού ἔχουν τοποθετηθεῖ στή μνήμη, στίς διευθύνσεις 1-100. ‘Ενα πρόγραμμα, πού θά ἔκανε αὐτή τή δουλειά, θά ἔπρεπε νά περιέχει 100 έντολές προσθέσεως ὅπως:

«Πρόσθεσε τό περιεχόμενο τῆς διευθύνσεως 1 στό συσσωρευτή»

«Πρόσθεσε τό περιεχόμενο τῆς διευθύνσεως 2 στό συσσωρευτή»

«Πρόσθεσε τό περιεχόμενο τῆς διευθύνσεως 3 στό συσσωρευτή»

K.O.K.

‘Υποθέτοντας βέβαια ὅτι ὁ συσσωρευτής εἶναι ἀρχικά κενός. Αὐτό ὁμως εἶναι ἐξαιρετικά ἀντοικονομικό, ἀν μάλιστα ή διαδικασία, στήν προκειμένη περίπτωση ή πρόσθεση, περιέχει πολλές έντολές ἀντί γιά μία και ἐπαναλαμβάνεται 1000 φορές ἀντί γιά 100. Αὐτό σημαίνει ἐπίσης και τεράστιο χῶρο, γιά τήν καταχώριση ὅλων αὐτῶν τῶν ὁμοίων έντολῶν στή μνήμη, ἀλλά και σπατάλη χρόνου γιά τό γράψιμο τῶν έντολῶν.

Σέ τέτοιες περιπτώσεις ή ὅλη διαδικασία γίνεται εὐκολα και ἀπλά μέ τή βοήθεια μιᾶς *έντολης ἐλέγχου* (ή διακλαδώσεως). ‘Ετσι, ἀν θέλομε νά ἐκτελεσθεῖ ἐπαναληπτικά ν φορές ή ἴδια ὁμάδα ἀπό έντολές και ἔπειτα ἀπό τίς ν ἐπαναλήψεις νά ἀκολουθηθεῖ και νά ἐκτελεσθεῖ μία ἄλλη ὁμάδα ἀπό έντολές, ἀποθηκεύομε στήν ἀρχή σέ κάποιο καταχωριστή τήν τιμή 1 και προσθέτομε σ’ αὐτόν μετά ἀπό κάθε ἐπανά-

ληψη τῆ μονάδα, ἐνῶ συγχρόνως ἐλέγχουμε τό περιεχόμενό του ἂν ἰσοῦται μέ τὴν τιμὴ ν. Ὑστερα ἀπὸ ν ἐπαναλήψεις ὁ καταχωριστὴς θά ἔχει φθάσει τὴν τιμὴ ν καὶ ἡ ἐντολὴ διακλαδώσεως θά παραπέμψει σέ ἄλλο κλάδο τοῦ προγράμματος.

Τέλος θά πρέπει νά ὑπάρχουν δύο ἐντολές, ἀπὸ τίς ὁποῖες μέ τὴ μία θά μπορούμε νά διαβάσουμε δεδομένα ἀπὸ κάποια περιφερειακὴ μονάδα καὶ μέ τὴν ἄλλη νά δώσουμε ἀποτελέσματα σέ κάποια ἄλλη.

Μποροῦμε λοιπὸν νά κατατάξουμε τίς ἐντολές ἐνὸς ὑπολογιστῆ στὶς ἐξῆς βασικὲς κατηγορίες:

- α) Ἀριθμητικὲς ἐντολές.
- β) Λογικὲς ἐντολές ἢ ἐντολές συγκρίσεως ἢ διακλαδώσεως.
- γ) Ἐντολές μεταφορᾶς ἢ μετακινήσεως.
- δ) Ἐντολές εἰσόδου-ἐξόδου.

Μέ τίς ἀριθμητικὲς ἐκτελοῦμε τίς 4 ἀριθμητικὲς πράξεις, μέ τίς λογικὲς, συγκρίσεις καὶ ἐλέγχους, μέ τίς διακινήσεις ἐκτελοῦμε μεταφορὰ ἀπὸ μία περιοχὴ τῆς μνήμης σέ ἄλλη ἢ ἀπὸ τὴ μνήμη στοῦ συσσωρευτῆ καὶ ἀντίστροφα. Τέλος μέ τίς ἐντολές εἰσόδου-ἐξόδου προβαίνομε σέ ἀνάγνωση ἢ ἐγγραφή.

5.7 Τί εἶναι ἀλγόριθμος.

Συχνὰ θά ἀκούσαμε νά γίνεται λόγος γιὰ *ἀλγόριθμους*. Ἀλγόριθμος εἶναι μία μέθοδος ἀριθμητικῆς ἐπιλύσεως ἐνὸς μαθηματικοῦ προβλήματος, πού βασίζεται στὴν ἐπαναληπτικὴ ἐφαρμογὴ καὶ ἐκτέλεση μιᾶς σειρᾶς ἀπλῶν διαδικασιῶν (ἀριθμητικῶν πράξεων καὶ λογικῶν συγκρίσεων).

Ἔτσι π.χ. ἡ διαδικασία γιὰ τὴν εὕρεση τοῦ Μέγιστου Κοινοῦ Διαιρέτη ἢ τοῦ Ἐλάχιστου Κοινοῦ Πολλαπλασίου, τῆς τετραγωνικῆς ρίζας κλπ. ἀποτελοῦν ἀλγόριθμους.

Ἡ λέξη ἀλγόριθμος προέρχεται ἀπὸ τὸν Ἀραβὰ μαθηματικὸ *Alkhowarazim*, πού ἔδωσε διάφορους κανόνες γιὰ τὴν ἐκτέλεση τῶν 4 βασικῶν πράξεων τῆς ἀριθμητικῆς.

Ἐπειδὴ ὁ ὑπολογιστὴς μπορεῖ νά ἐκτελέσει μόνο ἀπλές ἀριθμητικὲς πράξεις, σύνθετοι ὑπολογισμοί, πού περιλαμβάνουν παραγωγίσεις, ὁλοκληρώσεις κλπ. θά πρέπει νά περιγραφοῦν μέ τὴ βοήθεια καταλλήλων ἀλγορίθμων. Τίς μεθόδους αὐτὲς μελετᾷ ἕνας ξεχωριστὸς κλάδος τῶν μαθηματικῶν, πού ὀνομάζεται Ἀριθμητικὴ Ἀνάλυση καὶ πού παρουσιάζει ἰδιαίτερη ἀνάπτυξη τὰ τελευταῖα χρόνια.

Ἀσκήσεις.

1. Ὑποθέσατε ὅτι διαθέτετε ὑπολογιστὴ δύο διευθύνσεων. Νά περιγράψετε μέ τὸν ἴδιο τρόπο, πού χρησιμοποιήσαμε στὴν παράγραφο 5.5, ἕνα στοιχειῶδες πρόγραμμα καὶ τὸν τρόπο λειτουργίας του, ἂν θέλομε νά υπολογίσουμε καὶ νά τυπώσουμε τὴν τιμὴ τῆς παραστάσεως

$$A + B - C \times D$$

ὅπου οἱ τιμές τῶν A, B, C, D εἶναι διατυπημένες ἀνά δύο σέ δύο δελτία.

2. Ὑποθέσατε ὅτι θέλετε νά υπολογίσετε τὸ μέσο ὄρο 4 ἀριθμῶν A, B, C, D, πού βρίσκονται χωρισμένοι στὶς διευθύνσεις 510, 511, 512, 513 τῆς μνήμης. Ἄν ὁ ὑπολογιστὴς πού διαθέτετε εἶναι μιᾶς διευθύνσεως, πῶς θά κατασκευάσετε τὸ ἀντίστοιχο πρόγραμμα; Νά χρησιμοποιήσετε ὡς κωδικούς ἐντολῶν τοὺς κωδικούς πού ἀναφέρονται στὸν πίνακα 5.2.1 τῆς παραγράφου 5.2. Πῶς θά διασκευάζατε τὸ πρόγραμμα, ἂν ὁ ὑπολογιστὴς ἦταν δύο διευθύνσεων;

ΚΕΦΑΛΑΙΟ ΕΚΤΟ

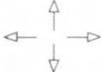
ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ

6.1 Σύμβολα διαγράμματος ροής.

Όπως είπαμε, για να μās επιλύσει κάποιο πρόβλημα ο υπολογιστής πρέπει να τόν έφοδιάσoμε μέ συγκεκριμένες οδηγίες-έντολές για τó τί ακριβώς έπιθυμούμε να κάνει. Οι οδηγίες αυτές καταγράφονται έτσι, ώστε να έμφανίζoυν τή λογική σειρά, πού θά πρέπει να τηρηθεί στην έπίλυση τοῦ προβλήματος.

Η άπεικόνιση μέ ειδικά σύμβολα τών οδηγιών, πού θά ακολουθηθoύν για τήν έπίλυση ενός προβλήματος, ονομάζεται *διάγραμμα ροής* (Flow chart) ή *λογικό διάγραμμα*. Κάθε σύμβολο διαγράμματος ροής παριστάνει μία συγκεκριμένη καί ανακεκριμένη λειτουργία.

Παρακάτω παρέχoμε τά κυριότερα σύμβολα, πού χρησιμοποιoύνται στό σχεδιασμό λογικών διαγραμμάτων.

Σύμβολο	Λειτουργία
	Άπλή λειτουργία. Έκτέλεση μιάς άπλης πράξεως ή ενός άπλου υπολογισμού (Process).
	Υποβολή έρωτήσεων, ή σημείο άποφάσεως, σύγκριση, ίκανοποίηση μιάς συνθήκης κλπ. (Decision).
	Γενικό σύμβολο για είσοδο-ανάγνωση δεδομένων ή έξοδο-έγγραφή αποτελεσμάτων. (Input-Output).
	Βέλη, πού δείχνoυν τή διεύθυνση ροής τοῦ λογικού διαγράμματος.
	Έγγραφή ή ανάγνωση από άρχείο σέ μαγνητική ταινία (Magnetic tape).

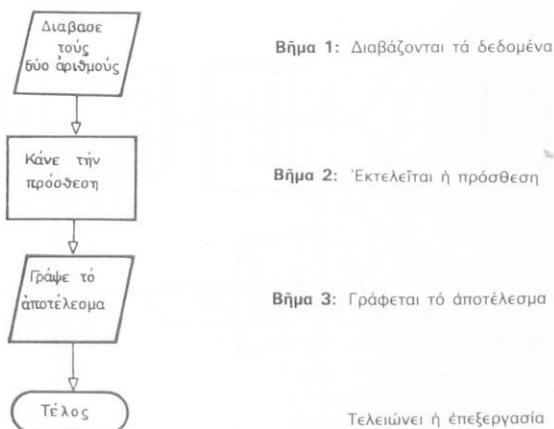
Σύμβολο	Λειτουργία
	Έγγραφή ή ανάγνωση από αρχείο σε μαγνητικό τύμπανο (Magnetic drum).
	Ανάγνωση δεδομένων από διάτρητα δελτία ή διάτρηση αποτελεσμάτων σε δελτία (Punched cards).
	Έγγραφή ή ανάγνωση από αρχείο σε μαγνητικό δίσκο (Magnetic disk).
	Έκτύπωση σε χαρτί (Printer).
	Ανάγνωση από χαρτοταινία ή διάτρηση σε χαρτοταινία (Punched Papertape).
	Είσαγωγή δεδομένων με πληκτρολόγηση (Manual Keyboard Input).
	Τροποποίηση, μεταβολή προγράμματος (Program modification).
	Διαλογή, ταξινόμηση (Sort).
	Απεικόνιση σε οθόνη (Display).
	Βοηθητική λειτουργία (Auxiliary operation).
	Σημείο συνδέσεως (Connector).
	Αρχή ή τέλος επεξεργασίας.

6.2 Παραδείγματα σχεδιασμού διαγραμμάτων ροής.

Παράδειγμα 1.

“Ας υποθέσουμε ότι μας ζητούν να βρούμε το άθροισμα δύο αριθμών. Για να εκτελέσουμε την πρόσθεση, θα πρέπει να γνωρίζουμε αυτούς τους αριθμούς, δηλαδή θα πρέπει να τους διαβάσουμε, κατόπιν να τους προσθέσουμε και να γράψουμε το άθροισμα.

Στήν περίπτωση αυτή, το λογικό διάγραμμα θα έχει την παρακάτω μορφή:



Παράδειγμα 2.

“Εστω ότι μας ζητούν να επιλύσουμε μία εξίσωση πρώτου βαθμού με ένα άγνωστο. Όπως ξέρομε, η γενική μορφή της είναι $Ax + B = 0$ και η γενική της λύση $x = -B/A$ (1). Δηλαδή για να τη λύσουμε θα πρέπει να γνωρίζουμε το A και το B . Για μία πιο ολοκληρωμένη μελέτη του προβλήματος θα πρέπει να προχωρήσουμε και σε μία μικρή διερεύνηση. Έτσι για τις διάφορες τιμές των A , B μπορεί να έχουμε τις εξής περιπτώσεις:

- “Αν $A \neq 0$, τότε η λύση που θα δίνεται από την (1) θα είναι κάποιος πραγματικός αριθμός.
- “Αν $A = 0$ και $B \neq 0$ τότε λέμε ότι έχουμε λύση αδύνατη.
- “Αν $A = 0$ και $B = 0$ τότε λέμε ότι έχουμε λύση άοριστη.

“Ας δούμε πώς θα διαμορφώσουμε το λογικό διάγραμμα.

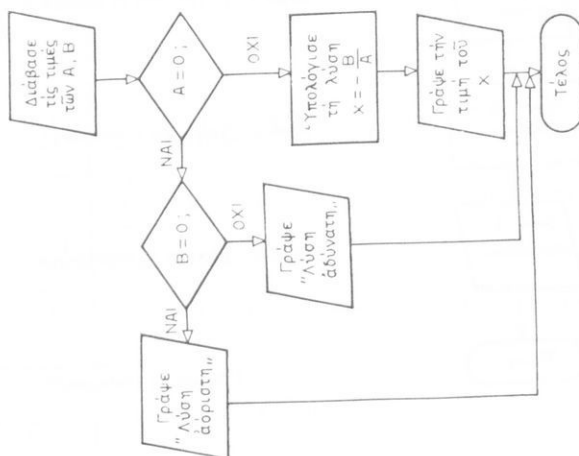
Βήμα 1: Διαβάζουμε τα δεδομένα A και B.

Βήμα 2: Ρωτάμε αν το A είναι μηδέν. "Αν όχι προχωρούμε πιο κάτω. "Αν ναι ακολουθούμε τον άριστερό κλάδο.

Βήμα 3: Υπολογίζουμε τη λύση.

Βήμα 4: Τήν τυπώνουμε.

Τέλος επεξεργασίας.

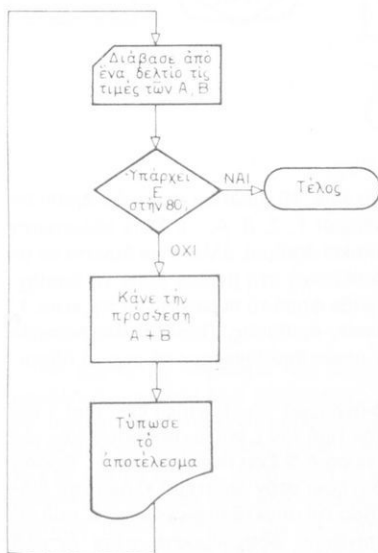


Βήμα 2α: Σέ περίπτωση που το A είναι 0, ρωτάμε αν και το B = 0. "Αν ναι τυπώνουμε το μήνυμα «ΛΥΣΗ ΑΟΡΙΣΤΗ». "Αν όχι τυπώνουμε το μήνυμα «ΛΥΣΗ ΑΔΥΝΑΤΗ». Και στις δύο περιπτώσεις οδηγούμαστε στη συνέχεια στο τέλος επεξεργασίας.

Φυσικά για ένα συγκεκριμένο ζευγάρι αριθμών A και B θα ακολουθηθεί ο ένας μόνο από τους τρεις δυνατούς δρόμους.

Παράδειγμα 3.

“Ας υποθέσουμε ότι θέλουμε να λύσουμε και πάλι το πρόβλημα του παραδείγματος 1 με τη διαφορά ότι δεν δίνεται ένα μόνο ζευγάρι τιμών A και B , αλλά περισσότερα. Ας υποθέσουμε ακόμα ότι τα δεδομένα μας τα έχουμε καταγράψει σε δελτία έτσι, ώστε σε κάθε δελτίο να υπάρχει μία τιμή A και μία τιμή B . Δηλαδή θα έχουμε τόσα δελτία, όσα και ζεύγη αριθμών A και B . Για να ελέγξουμε πότε θα τελειώσει η επεξεργασία, προσθέτουμε στο τέλος ένα δελτίο, που έχει επάνω του κάποιο σημάδι που δεν τὸ έχουν όλα τὰ άλλα, π.χ τὸ χαρακτήρα E στὴν 80η στήλη. Κάθε φορά, ἐπομένως, πού θὰ διαβάζεται ἓνα δελτίο, πρὶν προχωρήσουμε στὴν επεξεργασία, θὰ ρωτᾶμε μήπως εἶναι τὸ τελευταῖο, δηλαδή ἂν ὑπάρχει στὴν 80η στήλη ὁ χαρακτήρας E . Ἐτσι σχεδιάζουμε τὸ παρακάτω διάγραμμα ροῆς.



Βῆμα 1: Διαβάζουμε ἓνα δελτίο.

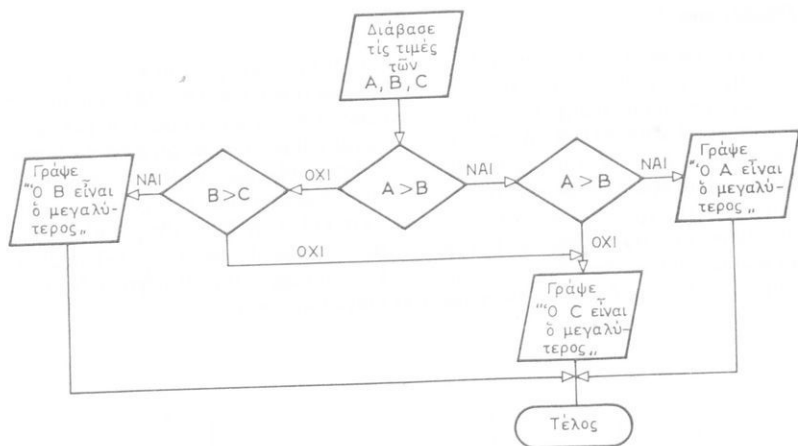
Βῆμα 2: Ρωτᾶμε ἂν ὑπάρχει ὁ χαρακτήρας E στὴν 80η στήλη. “Αν ναί, τελειώνει ἡ επεξεργασία. “Αν ὄχι, συνεχίζουμε.

Βῆμα 3: Ἐκτελεῖται ὁ ὑπολογισμός.

Βῆμα 4: Γράφεται τὸ ἀποτέλεσμα σὸ χαρτί καὶ στὴ συνέχεια ἐπιστρέφουμε γιὰ νὰ διαβάσουμε ἓνα νέο δελτίο.

Παράδειγμα 4.

“Εστω ὅτι δίνονται τρεῖς ἀριθμοὶ A , B , C ἄνισοι μεταξύ τους καὶ ζητεῖται νὰ βρεθεῖ ὁ μεγαλύτερος ἀπὸ αὐτοὺς. Τὸ λογικὸ διάγραμμα θὰ ἔχει τὴν ἐξῆς μορφή:

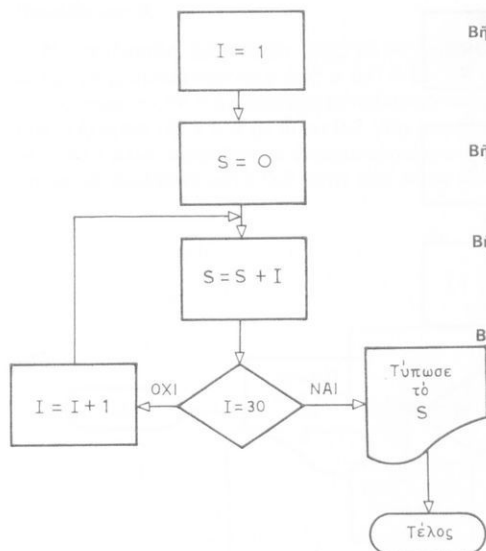


Παράδειγμα 5.

Έστω ότι ζητείται να βρεθεί το άθροισμα των 30 πρώτων φυσικών αριθμών (σάν φυσικοί χαρακτηρίζονται οι θετικοί άκεραίοι 1, 2, 3, 4, ...). Στην περίπτωση αυτή δεν χρειάζεται να διαβασθούν οι 30 φυσικοί αριθμοί, αλλά είναι δυνατό να όριστεί ένας μετρητής (Counter), δηλαδή μία περιοχή στη μνήμη, όπου να αποθηκεύσουμε αρχικά την τιμή 1 και να αυξάνομε κάθε φορά το περιεχόμενό της κατά 1, ώστε να παίρνομε τους διαδοχικούς φυσικούς αριθμούς. Παράλληλα χρησιμοποιούμε και ένα άθροιστή, τον S, μέσα στον οποίο αποθηκεύομε το μερικό άθροισμα των φυσικών αριθμών.

Την πρώτη φορά που θα εκτελεσθεί το διάγραμμα, ο μετρητής I έχει τιμή 1 και ο άθροιστής S τιμή 0. Προσθέτομε στον S την τιμή του I, που αντιπροσωπεύει τον πρώτο φυσικό αριθμό, δηλαδή τον 1. Έτσι τώρα ο S έχει περιεχόμενο 1. Επειδή $I = 1$ ή συνθήκη που ακολουθεί, θα μās οδηγήσει στον άριστερο κλάδο του διαγράμματος, όπου στον I προστίθεται μία μονάδα και αποκτā περιεχόμενο 2, που είναι ο επόμενος φυσικός αριθμός. Στη συνέχεια οδηγούμαστε στην έντολή $S = S + I$ όπου προστίθεται ή νέα τιμή του I = 2 στον S, που αποκτā περιεχόμενο $1 + 2 = 3$. Γίνεται ξανά ή έρώτηση για τό αν ο μετρητής I έχει φθάσει την τιμή 30. Η απάντηση είναι αρνητική και οδηγούμαστε και πάλι στον άριστερο κλάδο κ.ο.κ.

Η κυκλική αυτή διαδρομή επαναλαμβάνεται 30 φορές μέχρι τή στιγμή που ο I θα έχει αποκτήσει την τιμή 30, όποτε ή έρώτηση θα μās οδηγήσει στο δεξιό κλάδο. Τή στιγμή αυτή τυπώνεται τό περιεχόμενο του S που είναι και ή τιμή που ζητάμε και τελειώνει ή έπεξεργασία.



Βήμα 1: Δίνουμε στο μετρητή I αρχική τιμή 1.

Βήμα 2: Δίνουμε στον άθροιστή S αρχική τιμή 0.

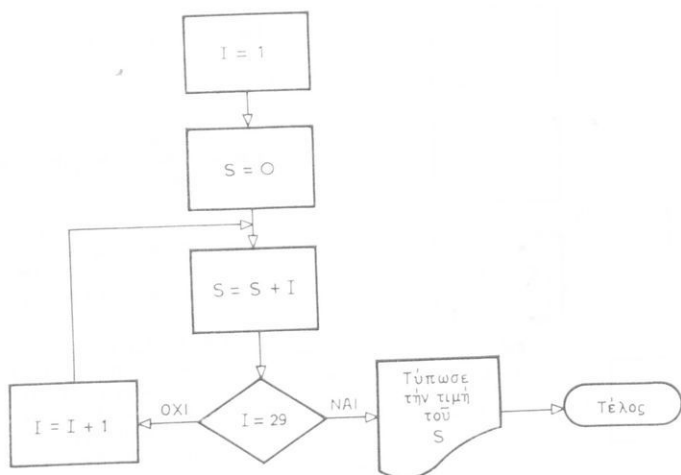
Βήμα 3: Προσθέτουμε την τιμή του I (πού συμπίπτει με ένα από τους φυσικούς αριθμούς) στον S .

Βήμα 4: Ρωτάμε αν ο μετρητής I έχει φθάσει την τιμή 30. Αν ναι, ακολουθούμε το δεξιό κλάδο, γράφουμε το αποτέλεσμα (δηλ. την τελευταία τιμή του S) και τελειώνουμε. Αν όχι, ακολουθούμε τον αριστερό κλάδο όπου αυξάνεται η τιμή του I κατά 1 και επιστρέφουμε στην έντολή του βήματος 3.

Παράδειγμα 6.

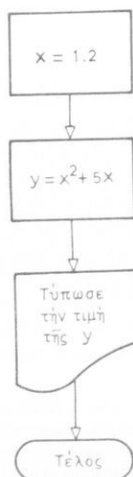
Νά βρεθεί το άθροισμα των περιπτών αριθμών από τό 1 ως τό 30. Έδω τό πρόβλημά μας είναι λίγο διαφορετικό από τό προηγούμενο. Πώς θά μπορούσαμε νά σχηματίσουμε τούς ζητούμενους διαδοχικούς περιττούς αριθμούς χωρίς νά χρειάζεται νά τούς διαβάσουμε; Θά πρέπει καί πάλι νά χρησιμοποιήσουμε ένα μετρητή I μέ αρχική τιμή 1¹, αλλά τώρα νά αυξάνουμε τό περιεχόμενό του κατά 2 (βήμα) μονάδες κάθε φορά, έτσι πού νά πάρουμε τούς 1, 3, 5,... Έπίσης αντί γιά 30 στήν έρώτηση, πρέπει νά γράψουμε 29, γιατί ό 29 είναι ό τελευταίος περιττός αριθμός πρίν τό 30. Κατά τά άλλα, τό διάγραμμα ροής παραμένει τό ίδιο.

¹ Η έκλογή των αρχικών τιμών εξαρτάται από τή μορφή τής εφαρμογής. Κατά περίπτωση επίσης εκλέγεται καί ό αριθμός-κριτήριο διακοπής τής επεξεργασίας πού εμφανίζεται στήν έρώτηση. Πάντως όταν χρησιμοποιούμε άθροιστές πρέπει νά τούς μηδενίζουμε στήν αρχή, ενώ όταν χρησιμοποιούμε πολλαπλασιαστές πρέπει νά δίνουμε αρχική τιμή 1.



Παράδειγμα 7.

Νά βρεθεί η τιμή της συνάρτησεως $y = x^2 + 5x$ για $x = 1.2$



Βήμα 1: Δίνουμε στο x την τιμή 1.2

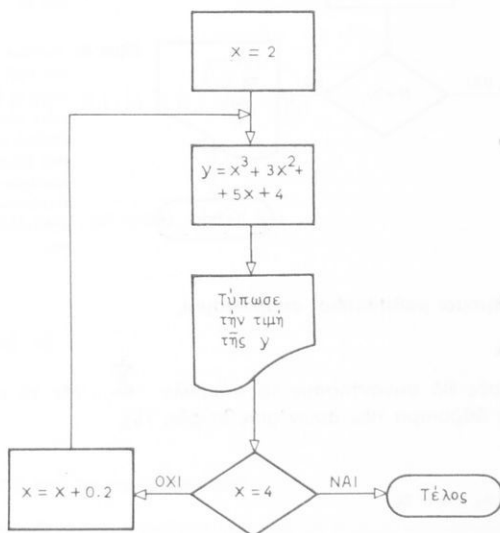
Βήμα 2: Γίνεται ο υπολογισμός της τιμής της y

Βήμα 3: Τυπώνεται το αποτέλεσμα

Τέλος επεξεργασίας

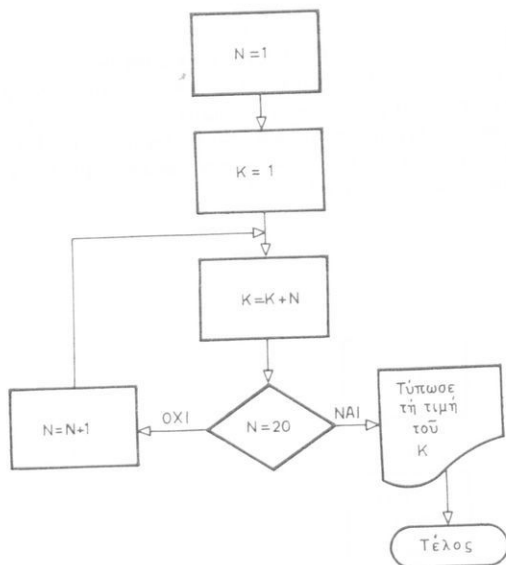
Παράδειγμα 8.

Νά σχεδιασθεί διάγραμμα ροής, μέ τό όποίο θά βρίσκονται καί θά τυπώνονται οί τιμές τής συναρτήσεως $y = x^3 + 3x^2 + 5x + 4$ γιά $x = 2$ (0.2)4. Ὁ συμβολισμός αὐτός σημαίνει ὅτι ἡ ἀνεξάρτητη μεταβλητή x παίρνει σάν πρώτη τιμή τήν $x = 2$ καί σάν τελευταία τήν $x = 4$ μέ βήμα 0.2. Δέν χρειάζεται καί ἐδῶ νά διαβάζονται οί τιμές τοῦ x ἀλλά μποροῦμε νά ἀποταμιεύσουμε στήν περιοχή x τήν ἀρχική τιμή 2 τήν ὁποία νά αὐξάνομε κατά 0.2 μετά ἀπό κάθε κύκλο.



Παράδειγμα 9.

Νά ὑπολογισθεῖ τό γινόμενο τῶν 20 πρώτων φυσικῶν ἀριθμῶν. Καί ἐδῶ, ὅπως στό παράδειγμα 5, δέν χρειάζεται νά διαβάσουμε ὅλους τούς φυσικούς ἀριθμούς, γιατί μποροῦμε νά τούς σχηματίσουμε, δίνοντας σέ ἕνα μετρητή N ἀρχική τιμή 1 καί αὐξάνοντας μετά ἀπό κάθε κύκλο τό περιεχόμενό του κατά 1. Ἐπίσης ὀρίζομε ἕνα πολλαπλασιαστή K , στόν ὁποῖο δίδομε ἀρχική τιμή 1. Σχεδιάζομε ἔτσι τό παρακάτω λογικό διάγραμμα.



Βήμα 1: Δίνουμε στο μετρητή N, αρχική τιμή 1.

Βήμα 2: Δίνουμε στον πολλαπλασιαστή K, αρχική τιμή 1.

Βήμα 3: Πολλαπλασιάζουμε την τιμή του K με το N. Τό γινόμενο αποτελεί τη νέα τιμή του K.

Βήμα 4: Ρωτάμε αν το N έφθασε την τιμή 20. "Αν όχι αυξάνουμε το N κατά 1 και παίρνουμε τον επόμενο φυσικό αριθμό και πηγαίνουμε πάλι στο βήμα 3. "Αν ναι, ακολουθούμε το δεξιό κλάδο, τυπώνουμε την τιμή του K και τελειώνει η επεξεργασία.

6.3 Μερικοί χρήσιμοι μαθηματικοί συμβολισμοί.

α) Άθροισματα.

Άρκετές φορές θα συναντήσουμε το σύμβολο $\sum$. Με το σύμβολο αυτό παριστάνουμε το άθροισμα των όρων μιάς σειράς. Π.χ.

$$\sum_{v=1}^8 a_v = a_1 + a_2 + \dots + a_8$$

$$\sum_{v=1}^5 a_v x^{v-1} = a_1 x^0 + a_2 x^1 + \dots + a_5 x^4$$

$$\sum_{v=1}^6 \frac{v(v+1)}{x^v} = \frac{1(1+1)}{x^1} + \frac{2(2+1)}{x^2} + \dots + \frac{6(6+1)}{x^6}$$

$$\sum_{v=0}^4 a^v = a^0 + a^1 + \dots + a^4$$

$$\sum_{v=2}^7 (a_v + x^v) = (a_2 + x^2) + (a_3 + x^3) + \dots + (a_7 + x^7)$$

β) Γινόμενα.

Μέ το σύμβολο $\prod$ παριστάνομε τό γινόμενο ενός πλήθους όμοίων όρων.
 $\prod \cdot x$.

$$\prod_{v=1}^4 a_v = a_1 \cdot a_2 \cdot a_3 \cdot a_4$$

$$\prod_{v=2}^5 (a_v + \beta_v) = (a_2 + \beta_2) \cdot (a_3 + \beta_3) \dots (a_5 + \beta_5)$$

$$\prod_{v=1}^8 x^v \eta \mu v x = (x \eta \mu x) \cdot (x^2 \eta \mu 2x) \dots (x^8 \eta \mu 8x)$$

$$\prod_{v=1}^3 a_v^v = a_1^1 \cdot a_2^2 \cdot a_3^3$$

6.4 Ή διαδικασία του προγραμματισμού.

Γιά νά δημιουργήσουμε ένα πρόγραμμα, πού νά έπιλύει κάποιο πρόβλημα, κάνομε τά έξής διαδικαστικά βήματα

- Καθορίζομε, αναλύομε καί περιγράφομε τό πρόβλημα πού θέλομε νά προγραμματίσουμε.
- Διαλέγομε μιά μέθοδο λύσεως.
- Σχεδιάζομε ένα λογικό διάγραμμα, όπου άπεικονίζομε τίς διάφορες λειτουργίες μέ τή λογική σειρά, πού θά άκολουθηθεῖ γιά νά έπιλυθεῖ τό πρόβλημα.
- Διαλέγομε μιά γλώσσα προγραμματισμού καί έκφράζομε τά διάφορα σύμβολα του λογικού διαγράμματος μέ τίς αντίστοιχες έντολές τής γλώσσας, πού τίς γράφομε έπάνω σέ είδικά έντυπα (Coding Forms).
- Μεταφέρομε τίς γραμμένες έντολές του προγράμματος έπάνω σέ κάποιο φορέα, π.χ τίς διατυπουΐμε σέ δελτία.
- Διοχετεύομε τό πρόγραμμα στον ύπολογιστή. Αύτός αναλαμβάνει νά τό μεταφράσει στη γλώσσα μηχανής μέ τό είδικό μεταφραστικό πρόγραμμα πού

διαθέτει, ενώ συγχρόνως προβαίνει καί σέ έλεγχο, γιά τό άν υπάρχουν συν-τακτικά ή όρθογραφικά λάθη στίς έντολές πού έχομε γράψει.

- η) Στίς περιπτώσεις πού δέν υπάρχουν λάθη γλωσσικά, δίνομε δοκιμαστικά δεδομένα (Sample Data) καί συγκρίνομε τά άποτελέσματα από τή λειτουργία του προγράμματος, στόν ύπολογιστή μέ τά άποτελέσματα από ύπολογισμούς πού έχομε κάνει μέ τό χέρι γιά τά ίδια δεδομένα. Έφ' όσον τά δύο άποτελέσματα συμπίπτουν, σημαίνει ότι ή λογική, πού χρησιμοποιήσαμε γιά τό σχεδιασμό του προγράμματος, ήταν σωστή.
- θ) Τό πρόγραμμα μπορεί νά λειτουργήσει μέ όποιαδήποτε δεδομένα.
- ι) Τεκμηριώνομε τό πρόγραμμα (Program documentation). Δηλαδή δημιουργούμε ένα φάκελλο, μέσα στόν όποιο γράφομε όλες τίς πληροφορίες, πού άναφέρονται στό συγκεκριμένο πρόγραμμα, όπως π.χ. τά λογικά διαγράμματα πού σχεδιάσαμε, τή μέθοδο έπιλύσεως πού άκολουθήσαμε, τά δεδομένα πού χρησιμοποιήσαμε γιά test, τά άποτελέσματα πού πήραμε, τίς διάφορες έκτυπώσεις (listings), κ.ο.κ.

Άσκησης.

1. Νά σχεδιασθεϊ λογικό διάγραμμα γιά τόν ύπολογισμό του άθροίσματος των τετραγώνων των 30 πρώτων φυσικών αριθμών.
2. Νά σχεδιασθεϊ λογικό διάγραμμα γιά τόν ύπολογισμό του γινομένου των 20 πρώτων άρτίων φυσικών αριθμών.
3. Νά σχεδιασθεϊ λογικό διάγραμμα γιά τόν ύπολογισμό του άθροίσματος των 10 πρώτων όρων τής άκολουθίας $1/2, 1/4, 1/8, 1/16, \dots$
4. Δίνονται 2000 αριθμοί. Ό καθένας έχει διατυπηθεϊ σέ ένα δελτίο. Νά σχεδιασθεϊ λογικό διάγραμμα γιά τήν εύρεση του μικρότερου από αυτούς.
5. Νά σχεδιασθεϊ λογικό διάγραμμα γιά τήν έπίλυση μιås δευτεροβάθμιας εξισώσεως μέ ένα άγνωστο $ax^2 + bx + c = 0$, τής όποιås οί συντελεστές a, b, c έχουν διατυπηθεϊ σέ ένα δελτίο. Νά προβλεφθεϊ καί μιá στοιχειώδης διερεύνηση γιά τή μορφή των ριζών.
6. Δίνονται 5 αριθμοί διατυπημένοι στό ίδιο δελτίο. Νά σχεδιασθεϊ λογικό διάγραμμα, μέ τό όποίο νά βρίσκεται καί νά τυπώνεται ό μέσος όρος τους.
7. Νά σχεδιασθεϊ λογικό διάγραμμα γιά τόν ύπολογισμό τής τιμής τής παραστάσεως:

$$\sum_{v=1}^8 \frac{x^v}{v}$$

Ό τιμή του x θα διαβάζεται στην άρχή από ένα δελτίο.

8. Νά σχεδιασθεϊ λογικό διάγραμμα γιά τόν ύπολογισμό τής τιμής τής παραστάσεως:

$$\prod_{v=1}^{24} (a_v \cdot x_v)$$

Οί τιμές των $a_1, x_1, a_2, x_2, \dots$ διαβάζονται από 24 δελτία. Σέ κάθε δελτίο υπάρχει μιá τιμή a καί ή άντίστοιχη x .

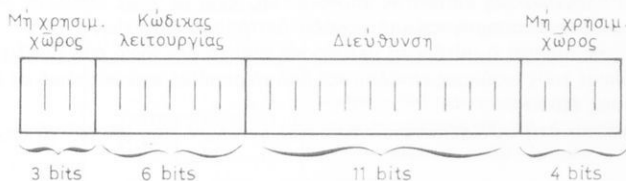
ΚΕΦΑΛΑΙΟ ΕΒΔΟΜΟ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕ ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ

7.1 Ένας υποθετικός εκπαιδευτικός υπολογιστής.

Στό κεφάλαιο αυτό θα γνωρίσουμε και θα χρησιμοποιήσουμε ένα υποθετικό εκπαιδευτικό ψηφιακό υπολογιστή, πού θα τον ονομάσουμε TK1.

Ο υπολογιστής αυτός διαθέτει μνήμη 2048 θέσεων, σε ένα μέρος της οποίας αποθηκεύουμε το πρόγραμμα και τα δεδομένα. Οι εντολές είναι τύπου μιας διευθύνσεως, δηλαδή κάθε εντολή αποτελείται από δύο κύρια τμήματα, τον κώδικα λειτουργίας και μία διεύθυνση. Κάθε θέση μνήμης αποτελείται από 24 bits. Στις θέσεις μνήμης έχουν δοθεί διευθύνσεις με δεκαδική σειριακή αρίθμηση από 0 ως 2047, ή σε δυαδική μορφή από 0 0 0 0 0 0 0 0 0 0 ως 1 1 1 1 1 1 1 1 1 1. Χρειάζονται λοιπόν 11 bits για τη δυαδική παράσταση της διευθύνσεως κάθε θέσεως. Για τους κωδικούς λειτουργίας εντολών χρειάζονται 6 bits. Έτσι η δομή μιας εντολής είναι:



Δέν υπάρχει διάκριση στη θέση μιας μνήμης, δηλαδή μπορεί άλλοτε να περιέχει μία εντολή και άλλοτε κάποιον αριθμό-δεδομένο.

Η αριθμητική μονάδα του TK1 περιέχει διάφορους καταχωριστές εκτελέσεως πράξεων προσθετή (Adder), αφαιρετή (Subtractor), πολλαπλασιαστή (Multiplier) κλπ. και ένα καταχωριστή διπλής χωρητικότητας (48 bits), πού χωρίζεται σε δύο ίσα μέρη, το βασικό συσσωρευτή (Accumulator) και το βοηθητικό συσσωρευτή (Upper Accumulator), πού μπορούν να λειτουργούν και ανεξάρτητα.

Ο συσσωρευτής χρησιμοποιείται με πολλούς τρόπους όπως:

- Για να δεχθεί πληροφορίες από τη μνήμη ή από τους διαύλους εισόδου.
- Για να στείλει πληροφορίες στη μνήμη ή τους διαύλους εξόδου.
- Για να κρατήσει το αποτέλεσμα από την εκτέλεση μιας πράξης.

Ο βοηθητικός συσσωρευτής είναι μία επέκταση του βασικού συσσωρευτή, και χρησιμοποιείται κυρίως στην εντολή του πολλαπλασιασμού. Όταν δύο αριθμοί με

πολλά ψηφία, έστω μέ 23 δυαδικά ό καθένas, πολλαπλασιασθούν, τό γινόμενό τους θά περιέχει 46 ψηφία καί δέν θά χωρά στο βασικό συσσωρευτή. Για τό λόγο αυτό διασπάται καί τό μισό μέ τά περισσότερο σημαντικά ψηφία τοποθετείται στο βοηθητικό συσσωρευτή, ένώ τό υπόλοιπο μισό στο βασικό συσσωρευτή.

Παράδειγμα.

Έστω $11001000 \times 110000, 0101000000$.
Τό αποτέλεσμα θά καταχωρισθεί ως έξής:

Βοηθητικός συσσωρευτής		Βασικός συσσωρευτής	
0	000,0000000000,0000001001	0	100,0100101101,0000000000
↑	πρόσημο	↑	πρόσημο

Σημειώνομε ότι σέ παρόμοιες περιπτώσεις τό πρώτο στοιχείο του βοηθητικού συσσωρευτή καί του βασικού συσσωρευτή περιέχει τό πρόσημο του αριθμού, πού φυσικά είναι τό ίδιο.

Η μονάδα έλέγχου περιλαμβάνει ένα καταχωριστή έντολών μέ μήκος 24 bits, όπου καταχωρίζεται κάθε φορά μία έντολή, πού μεταφέρεται από τή μνήμη, καί ένα καταχωριστή διευθύνσεων, του όποιου τό αριθμητικό περιεχόμενο δείχνει τή διεύθυνση τής επόμενης έντολής πού πρέπει, λογικά, νά εκτελεσθεί. Περιλαμβάνει ακόμα καί ένα αποκωδικοποιητή έντολών.

Ο TK1 διαθέτει δύο διαύλους εισόδου καί δύο έξόδου. Μέ τούς διαύλους εισόδου συνδέεται μέ μία περιφερειακή μονάδα άναγνώσεως διατρήτων δελτίων καί μία μονάδα άναγνώσεως διατρήτης χαρτοταινίας, ένώ μέ τούς διαύλους έξόδου συνδέεται μέ ένα έκτυπωτή καί μία μονάδα διατρήσεως δελτίων.

Είναι επίσης δυνατή ή αυτόματη μετατροπή από τό δεκαδικό στο δυαδικό σύστημα σέ όλους τούς διαύλους εισόδου καί από τό δυαδικό στο δεκαδικό σέ όλους τούς διαύλους έξόδου.

Περιγράφεται στή συνέχεια τό σύνολο των έντολών, πού μπορεί νά δεχθεί ό TK1.

7.2 Τό σύνολο περιγραφών των έντολών του TK1.

Δεκαδική παράσταση του κώδικα λειτουργίας	Έντολή	Περιγραφή της λειτουργίας		
16	<table><tr><td>R</td><td>Ad1</td></tr></table>	R	Ad1	Αντικατάσταση του περιεχομένου του συσσωρευτή από το περιεχόμενο της διευθύνσεως μνήμης Ad1. Το προηγούμενο περιεχόμενο του συσσωρευτή καταστρέφεται. Το περιεχόμενο της Ad1 διατηρείται.
R	Ad1			

Δεκαδική παράσταση του κώδικα λειτουργίας	Έντολή	Περιγραφή της λειτουργίας
17	C Ad1	Αντικατάσταση του περιεχομένου της διευθύνσεως μνήμης Ad1 από το περιεχόμενο του συσσωρευτή. Το περιεχόμενο του συσσωρευτή διατηρείται, ενώ το προηγούμενο περιεχόμενο της Ad1 καταστρέφεται.
10	A Ad1	Πρόσθεση του περιεχομένου της διευθύνσεως μνήμης Ad1 στο περιεχόμενο του συσσωρευτή. Το άθροισμα τοποθετείται στο συσσωρευτή. Η Ad1 διατηρεί το περιεχόμενό της.
11	S Ad1	Αφαίρεση του περιεχομένου της διευθύνσεως μνήμης Ad1 από το περιεχόμενο του συσσωρευτή. Η διαφορά τοποθετείται στο συσσωρευτή. Η Ad1 διατηρεί το περιεχόμενό της.
12	X Ad1	Πολλαπλασιασμός του περιεχομένου της διευθύνσεως Ad1 με το περιεχόμενο του συσσωρευτή. Το γινόμενο τοποθετείται στο συσσωρευτή. Η Ad1 διατηρεί το περιεχόμενό της.
13	D Ad1	Διαίρεση του περιεχομένου του συσσωρευτή με το περιεχόμενο της Ad1. Το πηλίκο τίθεται στο συσσωρευτή. Η Ad1 διατηρεί το περιεχόμενό της.
21	M Ad1	Έλεγχος για το πρόσημο του περιεχομένου του συσσωρευτή. Αν η τιμή του είναι αρνητική, συνεχίζεται η εκτέλεση του προγράμματος από την έντολή, που βρίσκεται στη διεύθυνση Ad1 της μνήμης.
20	P Ad1	Έλεγχος για το πρόσημο του περιεχομένου του συσσωρευτή. Αν η τιμή του είναι θετική, συνεχίζεται η εκτέλεση του προγράμματος από την έντολή που βρίσκεται στη διεύθυνση Ad1 της μνήμης.

Δεκαδική παράσταση του κώδικα λειτουργίας	Έντολή	Περιγραφή της λειτουργίας		
22	<table><tr><td>Z</td><td>Ad1</td></tr></table>	Z	Ad1	Έλεγχος για το αν το περιεχόμενο του συσσωρευτή είναι μηδέν. "Αν ναι, τότε συνεχίζεται η εκτέλεση του προγράμματος από την έντολή, που βρίσκεται στη διεύθυνση Ad1. "Αν όχι συνεχίζει στην επόμενη έντολή.
Z	Ad1			
15	<table><tr><td>J</td><td>Ad1</td></tr></table>	J	Ad1	Άλμα στην έντολή που βρίσκεται στη διεύθυνση Ad1.
J	Ad1			
30	<table><tr><td>B</td><td>—</td></tr></table>	B	—	Μεταφορά του περιεχομένου του βοηθητικού συσσωρευτή στο βασικό συσσωρευτή.
B	—			
28	<table><tr><td>E</td><td>Ad1</td></tr></table>	E	Ad1	Ανταλλαγή του περιεχομένου του συσσωρευτή με το περιεχόμενο της διεύθυνσης Ad1.
E	Ad1			
23	<table><tr><td>N</td><td>—</td></tr></table>	N	—	Άλλαξη στο πρόσημο του περιεχομένου του συσσωρευτή.
N	—			
32	<table><tr><td>V</td><td>Ad1</td></tr></table>	V	Ad1	Έγγραφη του περιεχομένου του συσσωρευτή στη διεύθυνση μνήμης Ad1 και μηδενισμός του συσσωρευτή.
V	Ad1			
40	<table><tr><td>I</td><td>Ad1</td></tr></table>	I	Ad1	Ανάγνωση από τη μονάδα εισόδου (ανάγνωστική διατρήτων δελτίων) ενός αριθμού και τοποθέτησή του μέσω του συσσωρευτή στη διεύθυνση Ad1.
I	Ad1			
41	<table><tr><td>W</td><td>Ad1</td></tr></table>	W	Ad1	Έγγραφη του περιεχομένου της διεύθυνσης Ad1 μέσω του συσσωρευτή στη μονάδα εξόδου (έκτυπητή).
W	Ad1			
42	<table><tr><td>T</td><td>Ad1</td></tr></table>	T	Ad1	Τοποθέτηση της διεύθυνσης Ad1 στον καταχωριστή διεύθυνσης.
T	Ad1			
25	<table><tr><td>H</td><td>—</td></tr></table>	H	—	Διακοπή της εκτέλεσης του προγράμματος.
H	—			

7.3 Πώς λειτουργεί ο TK1.

“Όλες οι εντολές ενός προγράμματος θα πρέπει αρχικά να αποθηκευθούν στη μνήμη. “Ας υποθέσουμε ότι η πρώτη εντολή καταχωρίζεται στη θέση της μνήμης με διεύθυνση 16. Τότε η μηχανή, όταν αρχίσει να λειτουργεί θα ξεκινήσει εκτελώντας αυτή την εντολή. Πολλές φορές το δείχνουμε αυτό στο γράψιμο του προγράμματος με ένα χαρακτήρα ή μία λέξη. Έδω θα χρησιμοποιούμε το χαρακτήρα M. “Όπως ήδη είπαμε, η μηχανή διαθέτει ένα καταχωριστή διευθύνσεων, που το αρχικό αριθμητικό του περιεχόμενο είναι η διεύθυνση της πρώτης να εκτελεσθεί εντολής. Μετά την εκτέλεση μιας εντολής το περιεχόμενό του αυξάνει αυτόματα κατά 1, για να εκτελεσθεί η επόμενη κ.ο.κ., εκτός αν, όπως θα δούμε, υπάρχει διακλάδωση σε κάποιο σημείο του προγράμματος. Για να μπορέσει λοιπόν να αρχίσει την εκτέλεση του προγράμματος ξεκινώντας από τη διεύθυνση 16, όπου υποθέσαμε ότι βρίσκεται η πρώτη εντολή, είναι απαραίτητο να τοποθετηθεί ο αριθμός 16 στον καταχωριστή διευθύνσεων.

7.4 Έντυπα κωδικογράφησης.

Οι εντολές αναγράφονται επάνω σε ειδικά σχεδιασμένα έντυπα, τα έντυπα κωδικογράφησης. Ειδικότερα αυτά, που χρησιμοποιούνται για τον προγραμματισμό σε γλώσσα μηχανής, παρουσιάζουν μεταξύ τους μικρές διαφορές, ανάλογα με τον τύπο της μηχανής. Μοιάζουν πάντως με τα έντυπα που χρησιμοποιούνται για τη γλώσσα (Assembly) που θα τα περιγράψουμε στο επόμενο κεφάλαιο.

7.5 Προγραμματισμός στη γλώσσα του TK1.

Στήν παράγραφο αυτή θα μάθουμε να γράφουμε απλά προγράμματα στη γλώσσα μηχανής, στη γλώσσα του TK1.

“Ας υποθέσουμε ότι θέλουμε σαν μία πρώτη εφαρμογή, να μεταφέρουμε απλώς στη θέση με διεύθυνση 403 της μνήμης, το περιεχόμενο της θέσεως με διεύθυνση 162. Το πρόγραμμα που θα πρέπει να δημιουργήσουμε θα έχει την εξής μορφή:

Διεύθυνση καταχωρήσεως έντολης	Έντολή		Επεξήγηση
	Κώδικας λειτουργίας	Διεύθυνση	
M → 16	R	162	Μεταφέρεται το περιεχόμενο της διευθύνσεως 162 στο συσσωρευτή. Μεταφέρεται το περιεχόμενο του συσσωρευτή στη διεύθυνση 403 της μνήμης.
17	C	403	

“Ας δούμε τώρα τί αλλαγές θα γίνουν στα περιεχόμενα του συσσωρευτή και των θέσεων 162 και 403 της μνήμης κατά τη διάρκεια των διαφόρων φάσεων της εκτελέσεως των δύο παραπάνω εντολών. Θα υποθέσουμε ότι ο συσσωρευτής έχει αρχικά κάποιο τυχαίο περιεχόμενο, έστω τον δεκαδικά εκφρασμένο αριθμό

732363, από κάποιο προηγούμενο ύπολογισμό. Έστω ότι η θέση 162 περιέχει τον αριθμό 4902 και η θέση 403 τον αριθμό 2091¹.

Φάσεις	Περιεχόμενο του καταχωριστή διευθύνσεων	Περιεχόμενο του καταχωριστή εντολών	Περιεχόμενο του συσσωρευτή	Περιεχόμενο της διευθύνσεως 162	Περιεχόμενο της διευθύνσεως 403
—	16	—	732363	4902	2091
1	17	R 162	732363	4902	2091
2	17	R 162	4902	4902	2091
3	18	C 403	4902	4902	2091
4	18	C 403	4902	4902	4902

Άς δοῦμε τώρα ποιές εντολές θά χρησιμοποιήσουμε γιά νά προσθέσουμε τό c (162) στό c (297) καί τό άθροισμα νά τό τοποθετήσουμε στή διεύθυνση 403.

Διεύθυνση καταχωρήσεως τής εντολής	Έντολή		Έπεξήγηση
	Κωδικός	Διεύθυνση	
M → 16 17 18	R A C	162 297 403	Άντικατάσταση του c(Ac) μέ τό c(162). Πρόσθεση του c(297), στό c(Ac) Μεταφορά καί άποθήκευση του c(Ac) στή διεύθυνση 403.

Στή συνέχεια θά μελετήσουμε τήν περίπτωση, πού θέλουμε νά αφαιρέσουμε τό περιεχόμενο τής διευθύνσεως 297 από τό περιεχόμενο τής διευθύνσεως 162 καί νά άποθηκεύσουμε τή διαφορά στή διεύθυνση 403.

Διεύθυνση καταχωρήσεως τής εντολής	Έντολή		Έπεξήγηση
	Κωδικός	Διεύθυνση	
M → 16 17 18	R S C	162 297 403	Άντικατάσταση του c(Ac) από τό c(162) Άφαίρεση του c(297) από τό c(Ac) Μεταφορά του c(Ac) στήν 403

Άν $c(162) = 4902$ καί $c(297) = 6845$, τότε μετά τό τέλος τής εκτέλεσεως των εντολών θά έχομε $c(403) = (-)1943$. Τό άρνητικό πρόσημο δείχνεται μέ τή παρουσία του δυαδικού ψηφίου 1 στήν άκράία άριστερή θέση τής λέξεως τής διευθύνσεως 403.

¹ Από τό σημείο αυτό καί έπειτα θά χρησιμοποιούμε τούς συμβολισμούς γιά τά περιεχόμενα των θέσεων τής μνήμης, πού καθορίσαμε στήν παράγραφο 5.4 δηλαδή c (162), c (Ac) (γιά τό βασικό συσσωρευτή), c (Up) (γιά τό βοηθητικό συσσωρευτή κ.ο.κ.).

Στή συνέχεια θά πολλαπλασιάσουμε τό c (162) μέ τό c (297) καί θά τοποθετή-
σομε τό γινόμενο στή διεύθυνση 403.

Διεύθυνση καταχωρήσεως έντολης	Έντολή		Έπεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	162	Αντικατάσταση του $c(Ac)$ από τό $c(162)$. Πολλαπλασιασμός του $c(Ac)$ μέ τό $c(297)$. Μεταφορά του $c(Ac)$ στήν 403.
17	X	297	
18	C	403	

Αν δέ $c(162) = 4902$ καί $c(297) = 45$, τότε μέ τήν εκτέλεση τῶν παραπάνω
έντολῶν θά ἔχομε $c(403) = 4902 \times 45 = 220590$. Από τήν ἄλλη μεριά αν
 $c(297) = 6845$, τότε $c(403) = 4902 \times 6845 = 33554190$. Ἐπειδὴ ὁ ἀριθμός
αὐτός εἶναι πολὺ μεγάλος, γιά νά χωρέσει στό βασικό συσσωρευτή, γι' αὐτό ἕνα
μέρος του διαρρέει στό βοηθητικό συσσωρευτή. Ἔτσι θά ἔχομε τήν κατάσταση:

Ἀριθμός
11111, 1111111111, 1100001110
(33,554,190)

Καταχωριστές

Βοηθητικός συσσωρευτής

Βασικός συσσωρευτής

0	000,0000000000,0000000011
---	---------------------------

(+) 25, 165, 824

0	111, 1111111111, 1100001110
---	-----------------------------

(+) 8, 388, 366

Ἄς δοῦμε κάποιο ἄλλο πρόγραμμα, ὅπου θέλομε νά πολλαπλασιάσουμε τό c
(162) μέ τό c (297) καί τό γινόμενο νά τό διαιρέσουμε μέ τό c (201).

Διεύθυνση καταχωρήσεως τῆς έντολης	Έντολή		Έπεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	162	Τό $c(162)$ μεταφέρεται στό συσσωρευτή. Πολλαπλασιάζεται τό $c(Ac)$ μέ τό $c(297)$ καί τό ἀποτέλεσμα τίθεται στό συσσωρευτή. Διαιρεῖται τό $c(Ac)$ μέ τό $c(201)$. Τό ἀπο- τέλεσμα τίθεται στό συσσωρευτή. Μεταφέρεται τό $c(Ac)$ στήν θέση 403.
17	X	297	
18	D	201	
19	C	403	

Δηλαδή πολλαπλασιάζεται τό περιεχόμενο τῆς διευθύνσεως 297 μέ τό τῆς
162, τό γινόμενο καταχωρίζεται στό συσσωρευτή καί κατόπιν διαιρεῖται τό περι-
εχόμενο τοῦ συσσωρευτῆ μέ τό περιεχόμενο τῆς 201 καί τό ἀποτέλεσμα τοποθε-
τεῖται στήν 403.



7.6 Έντολές αποφάσεων.

Μιά από τις πύλοππουδαιές έργασίες πού έκτελεί ό ύπολογιστής, είναι ή λήψη άποφάσεων. Άς δούμε μιά άπλή περίπτωση.

Έστω ότι θέλομε νά άφαιρέσομε τό μικρότερο από δύο άριθμούς, πού βρίσκονται στis θέσεις 162 και 297 από τό μεγαλύτερο και νά τοποθετήσομε τό άποτέλεσμα στην 403. Δέν γνωρίζομε βέβαια από πρίν σε ποιά θέση είναι ό μεγαλύτερος. Θά χρειασθούν 5 έντολές, οι έξης:

Διεύθυνση καταχωρίσεως της έντολής	Έντολή		Έπεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	162	Μεταφέρεται τό c(162) στον συσσωρευτή.
17	S	297	Άφαιρείται τό c(297) από τό c(Ac). Η διαφορά τοποθετείται στο συσσωρευτή.
18	P	21	Έλέγχεται τό πρόσημο του συσσωρευτή. Άν τό c(Ac) είναι θετικό, γίνεται άλλα στην έντολή, πού βρίσκεται στη διεύθυνση 21. Διαφορετικά έκτελείται ή επόμενη έντολή.
19	R	297	Άντικατάσταση του c(Ac) από τό c(297).
20	S	162	Άφαιρείται τό c(162) από τό c(Ac)
18 → 21	C	403	Μεταφέρεται τό c(Ac) στην 403.

Οι δύο έναλλακτικοί δρόμοι, πού άνοίγονται στην έντολή 18, όνομάζονται κλάδοι και ή έντολή 18, έντολή διακλαδώσεως.

Άν ζητάμε αντίστροφα νά άφαιρέσομε τό μεγαλύτερο από τό μικρότερο γιά τούς ίδιους δύο άριθμούς, θά πρέπει νά τροποποιήσομε έλαφρά τό προηγούμενο πρόγραμμα. Έτσι θά έχομε τις έξης έντολές:

Διεύθυνση καταχωρίσεως της έντολής	Έντολή		Έπεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	162	Μεταφέρεται τό c(162) στο συσσωρευτή.
17	S	297	Άφαιρείται τό c(297) από τό c(Ac).
18	M	21	Άν τό c(Ac) είναι άρνητικό, τότε πηγαίνει στην έντολή 21. Διαφορετικά συνεχίζει στην 19.
19	R	297	Μεταφέρεται τό c(297) στο συσσωρευτή.
20	S	162	Άφαιρείται τό c(162) από τό c(Ac).
18 → 21	C	403	Μεταφέρεται τό c(Ac) στη 403.

Άς δούμε ένα άκόμα παράδειγμα. Έστω ότι θέλομε νά έλέγξομε τό περιεχόμενο της διευθύνσεως 403 και άν αυτό είναι μηδέν νά τό άφήσομε άμετάβλητο, διαφορετικά νά μεταφέρομε τό περιεχόμενο της διευθύνσεως 162.

Διεύθυνση καταχωρίσεως τῆς ἐντολῆς	Ἐντολή		Ἐπεξήγηση
	Κώδικας	Διεύθυνση	
M → 16 17	R Z	403 19	Μεταφέρεται τὸ c(403) στὸ συσσωρευτή. Ἄν τὸ c(Ac) εἶναι μηδέν, τότε πηγαίνει στὴν ἐντολή 19. Διαφορετικὰ συνεχίζει στὴν 18.
18 17 → 19	R C	162 403	Μεταφέρεται τὸ c(162) στὸ συσσωρευτή. Μεταφέρεται τὸ c(Ac) στὴ διεύθυνση 403.

7.7 Τρόπος χρησιμοποίησεως μερικῶν ἀκόμα ἐντολῶν.

Στὸ σημεῖο αὐτὸ θὰ δοῦμε πῶς χρησιμοποιοῦνται σὲ ἓνα πρόγραμμα 5 ἀκόμη ἐντολές, οἱ B, I, V, W καὶ H.

Μέ τὴν ἐντολή B χειριζόμαστε τὸ περιεχόμενο τοῦ βοηθητικοῦ συσσωρευτή. Ἐστω ὅτι εἶναι γνωστὸ ὅτι κάποιο σύνολο, ποῦ θέλομε νὰ βροῦμε, εἶναι τόσο μεγάλο, ὥστε νὰ χρειάζεται 2 θέσεις π.χ. 403 καὶ 404 γιὰ νὰ ἀποθηκευθεῖ (τὸ μικρότερο τμήμα στὴν 404 καὶ τὸ πῶς σημαντικό στὴν 403). Μέ τὸ παρακάτω πρόγραμμα προστίθεται ὁ ἀριθμὸς ποῦ βρίσκεται στὴ θέση 162 στὸ «διπλὸ αὐτὸ ἀριθμὸ».

Διεύθυνση καταχωρίσεως τῆς ἐντολῆς	Ἐντολή		Ἐπεξήγηση
	Κώδικας	Διεύθυνση	
M → 16 17 18 19 20 21	R A C B A C	404 162 404 — 403 403	Ἀντικαθίσταται τὸ c(Ac) ἀπὸ τὸ c(404). Προστίθεται τὸ c(162) στὸ c(Ac). Ἐγγράφεται τὸ c(Ac) στὴ θέση 404. Μεταφέρεται τὸ c(Ur) στὸ συσσωρευτή. Προστίθεται τὸ c(403) στὸ c(Ac). Ἐγγράφεται τὸ c(Ac) στὴ θέση 403.

Ὅπως ἤδη ἔχομε πεῖ, ὑπάρχουν 4 ἀνεξάρτητοι δίαυλοι εἰσόδου - ἐξόδου συνδεδεμένοι μέ τὸν TK1. Ἐστω λοιπὸν ὅτι θέλομε νὰ διαβάσουμε 1000 ἀριθμούς ἀπὸ τὴ μονάδα ἀναγνώσεως δελτίων καὶ νὰ τοποθετήσουμε τὸ ἄθροισμά τους στὴ διεύθυνση 4. Τὸ πρόγραμμα θὰ ἔχει τὴν παρακάτω μορφή.

Διεύθυνση καταχωρίσεως τῆς ἐντολῆς	Ἐντολή		Ἐπεξήγηση
	Κώδικας	Διεύθυνση	
M → 19 20 21 19 ← 22	I A C J	6 6 4 19	Διαβάζεται ἓνας ἀριθμὸς-δεδομένο καὶ τοποθετεῖται στὴ θέση 6. Προστίθεται τὸ c(6) στὸ c(Ac) ποῦ εἶναι ἀρχικὰ μηδέν. Μεταφέρεται τὸ ἄθροισμα [c(Ac)] στὴ θέση 4. Ὁ συσσωρευτὴς διατηρεῖ τὸ περιεχόμενό του. Ξαναπηγαίνει στὴν ἐντολή 19

Στό πρόγραμμα αυτό όμως γίνεται μία συνεχώς ανακυκλούμενη εκτέλεση των εντολών 19 έως 22 χωρίς να υπάρχει πρόβλεψη για αυτόματο σταμάτημα της λειτουργίας. Ο υπολογιστής θα σταματήσει υποχρεωτικά, όταν εξαντληθούν τα δεδομένα.

Για να επιτύχουμε το αυτόματο σταμάτημα του προγράμματος μετά την πρόσθεση και του 1000ου διαβαζόμενου αριθμού, τοποθετούμε τον αριθμό 1000 στη διεύθυνση 33 και τον αριθμό 1 στη διεύθυνση 32.

Μία πιο εξελιγμένη μορφή προγράμματος για την προηγούμενη εφαρμογή είναι η ακόλουθη:

Διεύθυνση καταχωρίσεως της εντολής	Έντολή		Έπεξήγηση
	Κώδικας	Διεύθυνση	
16	V	6	$c(Ac) = 0$, $c(6) = \text{κάτι}$
17	C	4	$c(4) = 0$
18	C	5	$c(5) = 0$
19	C	6	$c(6) = 0$
29 → 20	I	6	Διαβάζεται ένας αριθμός και καταχωρίζεται στη διεύθυνση 6.
21	A	6	Προστίθεται το $c(6)$ στο συσσωρευτή.
22	C	4	Μεταφέρεται το $c(Ac)$ στη θέση 4.
23	R	5	Αντικαθίσταται το $c(Ac)$ με το $c(5)$.
24	A	32	Προστίθεται το $c(32)$ στο $c(Ac)$.
25	C	5	Μεταφέρεται το $c(Ac)$ στη θέση 5.
26	S	33	Αφαιρείται το $c(33)$ από το $c(Ac)$.
27	Z	30	Ελέγχεται το $c(Ac)$. Αν είναι μηδέν εκτελείται στη συνέχεια η εντολή που βρίσκεται στη διεύθυνση 30. Διαφορετικά η επόμενη (28).
28	R	4	Αντικαθίσταται το $c(Ac)$ με το $c(4)$.
20 ← 29	J	20	Γίνεται άλμα στην εντολή, που βρίσκεται στη διεύθυνση 20.
30	W	4	Εγγράφεται στον έκτυπωτή το $c(4)$.
31	H	—	Τέλος της λειτουργίας.
32	—	1	Η διεύθυνση 32 περιέχει τον αριθμό 1.
33	—	1000	Η διεύθυνση 33 περιέχει τον αριθμό 1000.

Άσκησης.

- Έστω ότι $c(15) = 207$, $c(21) = 45$, $c(30) = 100$. Να γράψετε πρόγραμμα σε γλώσσα μηχανής, χρησιμοποιώντας τις προδιαγραφές του TK1, για να υπολογίσετε την τιμή της παραστάσεως, $(c(15) + c(21)) \cdot c(30) - c(21)$.
- Έστω ότι $c(18) = 10$, $c(19) = 15$, $c(20) = 25$, $c(21) = 20$. Να γράψετε πρόγραμμα σε γλώσσα μηχανής (στη γλώσσα του TK1), για να υπολογίσετε την τιμή του μέσου όρου τους, την οποία να καταχωρίσετε στη θέση 30.
- Να συνθέσετε πρόγραμμα στη γλώσσα του TK1, που να υπολογίζει το άθροισμα των 20 πρώτων φυσικών αριθμών. Να τοποθετήσετε το τελικό άθροισμα στη διεύθυνση 105.
- Να γράψετε πρόγραμμα στη γλώσσα του TK1, που να υπολογίζει το γινόμενο των 10 πρώτων φυσικών αριθμών. Τοποθετήστε το τελικό αποτέλεσμα στη διεύθυνση 10.
- Να δημιουργήσετε μία δική σας κωδικοποίηση για όρισμένες βασικές εντολές (όπως κάναμε και για τον TK1) και μετά να την χρησιμοποιήσετε για να επιλύσετε την άσκηση 1.

ΚΕΦΑΛΑΙΟ ΟΓΔΟΟ

ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ASSEMBLY

8.1 Γενικά.

Μιά συμβολική γλώσσα προγραμματισμού τύπου *Assembly* βρίσκεται, όπως έχουμε ξαναπεί, κατά μία βαθμίδα ψηλότερα από τη γλώσσα μηχανής στην ιεραρχία των γλωσσών προγραμματισμού.

Επειδή ο προγραμματισμός σε γλώσσα μηχανής είναι δύσκολη και επίπονη εργασία, ενώ ο κίνδυνος να γίνουν λάθη είναι μεγάλος, δημιουργήθηκε ένας τρόπος εκφράσεως των εντολών της γλώσσας μηχανής με χρήση συμβόλων τόσο για τόν *κώδικα έντολης* (Operation Code) όσο και για τις *διευθύνσεις* (Operands). Όσον αφορά στις τελευταίες, η χρήση των συμβολικών μνημονικών χαρακτηριστικών στη χρησιμοποίηση των απόλυτων διευθύνσεων των θέσεων της μνήμης (δηλαδή των αριθμητικών διευθύνσεων, όπως έχουν δοθεί αρχικά με την πρώτη λειτουργία του υπολογιστή και όπου κάθε διεύθυνση δρίζει μονοσήμαντα μία θέση μνήμης). Σε αυτή την περίπτωση χαρακτηρίζουμε μία θέση μνήμης με ένα συμβολικό όνομα και αναφερόμαστε μετά σε αυτήν με το όνομα αυτό, χωρίς να χρειάζεται να γνωρίζουμε την απόλυτη διεύθυνσή της. Φυσικά είναι δυνατό σε κάποιο άλλο πρόγραμμα την ίδια θέση μνήμης να τη συμβολίσουμε με κάποιοι άλλη ονομασία. Ο υπολογιστής αναλαμβάνει να καταχωρίσει την έντολη ενός προγράμματος με τρόπο συνεχή σε μία περιοχή της μνήμης και η αντίστοιχη απόλυτων και συμβολικών διευθύνσεων γίνεται έτσι, ώστε να τηρείται η δοσμένη λογική σειρά στην εκτέλεση των εντολών.

Επειδή, όπως είπαμε, η μηχανή αντιλαμβάνεται μόνο τη δική της γλώσσα, ένα *Assembly* πρόγραμμα πρέπει να μεταφραστεί στη γλώσσα μηχανής πριν εκτελεσθεί. Η μετάφραση αυτή επιτυγχάνεται με ένα ειδικό μεταφραστικό πρόγραμμα που ονομάζεται *Assembler* ή *Assembly Compiler*.

8.2 Διαδικασία προγραμματισμού στη γλώσσα *Assembly*.

Η διαδικασία για τον προγραμματισμό σε γλώσσα *Assembly* περιλαμβάνει τα εξής στάδια:

- Σχεδιάζουμε ένα διάγραμμα ροής, όπου απεικονίζουμε τη λογική, που θα ακολουθηθεί για την επίλυση του προβλήματος.

- Κωδικοποιούμε το διάγραμμα, με τη βοήθεια των εντολών της *Assembly*, τις οποίες γράφουμε επάνω στα ειδικά έντυπα (*Assembly Coding Forms* σχ. 8.3).

- Οι εντολές του προγράμματος, διατρυπώνται σε δελτία και σχηματίζεται το σύνολο των δελτίων του πηγαίου προγράμματος (*Source program deck*).

- Τα δελτία του προγράμματος διαβάζονται από την περιφερειακή μονάδα αναγνώσεως δελτίων και το περιεχόμενό τους μεταφέρεται στην κεντρική μονάδα, όπου μεταφράζεται με τη βοήθεια του ειδικού μεταφραστικού προγράμματος (*Assembly*) στη γλώσσα μηχανής (*Object program*).

- Μετά τη μετάφραση εκτυπώνονται οι εντολές του *Assembly* προγράμματός μας (*Listing*), όπου έχουν επισημανθεί συντακτικά ή όρθογραφικά λάθη, που βρέθηκαν στο στάδιο της μεταφράσεως. Δίπλα στην κάθε έντολη *Assembly* αναγράφεται και η αντίστοιχη έντολη της μηχανής.

8.3 Έντυπο κωδικογραφήσεως *Assembly* (*Assembly Coding Form*).

Τό έντυπο αυτό σχ. 8.3 περιλαμβάνει ορισμένο αριθμό γραμμών, που κάθε μία χωρίζεται σε 80

PROGRAM		PUNCHING INSTRUCTIONS		GRAPHIC PUNCH		PAGE OF	
PROGRAMMER		DATE		STATEMENT		CARD ELECTRO NUMBER	
1	Name	10	Operator	16	20	25	Operator
		30		35		40	
		45		50		55	
		60		65		70	
		75		80		85	
		90		95		100	
		105		110		115	
		120		125		130	
		135		140		145	
		150		155		160	
		165		170		175	
		180		185		190	
		195		200		205	
		210		215		220	
		225		230		235	
		240		245		250	
		255		260		265	
		270		275		280	
		285		290		295	
		300		305		310	
		315		320		325	
		330		335		340	
		345		350		355	
		360		365		370	
		375		380		385	
		390		395		400	
		405		410		415	
		420		425		430	
		435		440		445	
		450		455		460	
		465		470		475	
		480		485		490	
		495		500		505	
		510		515		520	
		525		530		535	
		540		545		550	
		555		560		565	
		570		575		580	
		585		590		595	
		600		605		610	
		615		620		625	
		630		635		640	
		645		650		655	
		660		665		670	
		675		680		685	
		690		695		700	
		705		710		715	
		720		725		730	
		735		740		745	
		750		755		760	
		765		770		775	
		780		785		790	
		795		800		805	
		810		815		820	
		825		830		835	
		840		845		850	
		855		860		865	
		870		875		880	
		885		890		895	
		900		905		910	
		915		920		925	
		930		935		940	
		945		950		955	
		960		965		970	
		975		980		985	
		990		995		1000	

Σχ. 8.3.

Τό έντυπο κωδικογραφήσεως Assembly.

στήλες. Σε κάθε γραμμή περιγράφεται μία μόνο έντολη.

Στις στήλες 1-8 (Name) δίνεται μία όνομασία, που είναι ή συμβολική διεύθυνση μιας έντολης. Δεν χρειάζεται να δίνουμε συμβολική διεύθυνση σε κάθε έντολη. Παρέχουμε μόνο στίς έντολές εκείνες, όπου μās παραπέμπει τό πρόγραμμα μετά τη έκτέλεση μιας έντολης διακλαδώσεως. Ή συμβολική όνομασία πρέπει να άρχίζει με άλφαβητικό χαρακτήρα στη στήλη 1. Ή στήλη 4 άφήνεται κενή.

Στις στήλες 10-14 (Operation) γράφεται ό συμβολικός κώδικας λειτουργίας της έντολης. Ή στήλη 15 μένει κενή.

Ή από τη στήλη 16 γράφουμε τίς συμβολικές διευθύνσεις (Operands) των όρων της περιγραφόμενης έντολης.

Σχόλια, παρατηρήσεις ή επεξηγήσεις (Comments) για τη λειτουργία μιας έντολης ή ενός τμήματος του προγράμματος μπορούμε να γράφουμε σε οποιαδήποτε γραμμή έντολης μετά την περιγραφή των συμβολικών διευθύνσεων, αφήνοντας μία τουλάχιστον κενή στήλη μετά από αυτές. Μπορούμε όμως να χρησιμοποιήσουμε και μία όλοκληρη γραμμή του έντύπου θέτοντας άπλως στην 1η στήλη της, τον άστερίσκο (*)¹. Ή στήλη 72 όταν περιέχει όποιοδήποτε χαρακτήρα σημαίνει συνέχιση στην επόμενη γραμμή από τη στήλη 16.

Στις στήλες 73 - 80 (Identification Sequence) γράφεται ή κωδική όνομασία του προγράμματος (προδιαγραφή που άγνοείται στο στάδιο της μεταφράσεως), μαζί με την αύξουσα άρίθμηση των εντολών.

8.4 Γενική περιγραφή του υπολογιστή που θά χρησιμοποιήσουμε.

Ή υπολογιστής, για τόν όποιο θά μιλάμε στα επόμενα, έχει μνήμη που άποτελείται από λέξεις των 32 bits. Κάθε λέξη χωρίζεται σε δύο τμήματα (Halfwords) των 16 bits τό καθένα. Κάθε μισή λέξη άποτελείται από 2 bytes των 8 bits. Μέσα σε κάθε byte μπορεί να καταχωρισθεϊ ένας χαρακτήρας κωδικοποιημένος στο σύστημα EBCDIC.

Στήν κεντρική μονάδα υπάρχουν 16 καταχωριστές (Registers) με μήκος 32 bits ό καθένας. Ή χρήση καταχωριστών στη γλώσσα Assembly άποτελεί τη βάση της δομής της.

8.5 Κατηγορίες εντολών Assembly.

Ή περιγράφομε στα επόμενα τη γλώσσα Assembly που χρησιμοποιείται σε άρκετούς υπολογιστές.

Όι έντολές της γλώσσας αυτής χωρίζονται σε 6 κατηγορίες:

α) **Έντολές τύπου RR.**

Ή αναφέρονται σε πράξεις που γίνονται άνάμεσα στα περιεχόμενα δύο καταχωριστών (Register and Register).

β) **Έντολές τύπου RX.**

Ή αναφέρονται σε πράξεις που γίνονται άνάμεσα στα περιεχόμενα ενός καταχωριστή και μίας θέσεως στη μνήμη (Register and Storage).

γ) **Έντολές τύπου RS.**

Τό ίδιο όπως και οι RX (διαφέρει ή δομή τους).

δ) **Έντολές τύπου SI.**

Για πράξεις άνάμεσα στο περιεχόμενο μιας θέσεως της μνήμης και μίας σταθεράς.

ε) **Έντολές τύπου SS1.**

Ή αναφέρονται σε πράξεις άνάμεσα στα περιεχόμενα δύο ίσομήκων περιοχών της μνήμης.

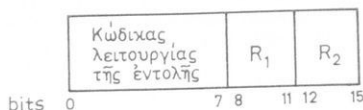
στ) **Έντολές τύπου SS2.**

Ή αναφέρονται σε πράξεις άνάμεσα στα περιεχόμενα δύο άνισομήκων περιοχών της μνήμης.

¹ Ή άστερίσκος στη στήλη 1 προειδοποιεί τόν υπολογιστή να μη θεωρήσει τό περιεχόμενο της γραμμής σαν έντολη.

8.6 Έντολές τύπου RR.

Κάθε έντολή του τύπου αυτού καταλαμβάνει χώρο 16 bits στη μνήμη και καταχωρίζεται ως εξής:



Δηλαδή ο κώδικας λειτουργίας της έντολης καταχωρίζεται στο πρώτο byte της (0-7 bits και στο δεύτερο byte καταχωρούνται δύο αριθμοί που συμβολίζουν δύο καταχωριστές R_1 και R_2 ¹.

Στην κατηγορία αυτή ανήκουν οι έντολές:

α) Έντολή προσθέσεως (AR).

Η μορφή της είναι:

AR R_1 , R_2

Με τη βοήθεια της προστίθεται το περιεχόμενο του καταχωριστή R_2 στο περιεχόμενο του καταχωριστή R_1 . Το άθροισμα τοποθετείται στον R_1 . Το περιεχόμενο του R_2 δεν αλλάζει π.χ. AR 11, 14 σημαίνει ότι θα προστεθεί το περιεχόμενο του καταχωριστή 14 στο περιεχόμενο του καταχωριστή 11, και τό άθροισμα θα καταχωρηθεί στο 11. Δηλαδή $[11] + [14] = [11]$.

β) Έντολή αφαιρέσεως (SR).

Η μορφή της είναι:

SR R_1 , R_2

Με την έντολή αυτή αφαιρείται το περιεχόμενο του καταχωριστή R_2 από το περιεχόμενο του καταχωριστή R_1 . Η διαφορά καταχωρείται στον R_1 . Το περιεχόμενο του R_2 δεν αλλάζει. Π.χ.

SR 8, 2

Η εκτέλεση της έντολης αυτής θα έχει σαν αποτέλεσμα να αφαιρεθεί το περιεχόμενο του καταχωριστή 2 από το περιεχόμενο του καταχωριστή 8 και η διαφορά να τοποθετηθεί στον καταχωριστή 8.

Πρίν προχωρήσουμε στη περιγραφή των επόμενων έντολων, θα πούμε δυό λόγια γιά μία σημαντική προδιαγραφή της Assembly, *τόν δείκτη συνθήκης*.

Οι αποφάσεις και οι διακλαδώσεις αποτελούν σημαντικές ενέργειες σε ένα πρόγραμμα, ο δε τρόπος με τον τον όπου μπορούν να προσδιορισθούν με ανάλογες έντολές ενδιαφέρουν άμεσα τον προγραμματιστή.

Στή γλώσσα Assembly, που εξετάζουμε, η λειτουργία των έντολων αυτών βασίζεται στη χρησιμοποίηση του δείκτη συνθήκης. Αυτός αναφέρεται σε πολλές έντολές και η τιμή του ελέγχεται από μία έντολή διακλαδώσεως υπό συνθήκη (Branch on condition).

Μέρος της λειτουργίας πολλών αριθμητικών ή λογικών έντολων αποτελεί και η ενεργοποίηση του δείκτη συνθήκης, ανάλογα με τό είδος του αποτελέσματος, μετά την εκτέλεση της έντολης αυτής.

Π.χ. μετά την εκτέλεση μιάς έντολης προσθέσεως, ο δείκτης συνθήκης μπορεί να δείχνει κατά πόσο τό άθροισμα είναι θετικό, άρνητικό ή μηδέν. Άκόμη, μετά την εκτέλεση μιάς έντολης συγκρίσεως, ο δείκτης συνθήκης δείχνει κατά πόσο ο πρώτος όρος είναι μεγαλύτερος από, ή ίσος πρός, ή μικρότερος από τον δεύτερο όρο. Ο δείκτης συνθήκης μπορεί να πάρει τέσσερις τιμές 0, 1, 2 και 3.

Άφού ο δείκτης συνθήκης ενεργοποιηθεί κατάλληλα παίρνοντας μιά από τίς 4 τιμές που είπαμε, η κατάσταση, στην οποία βρίσκεται μπορεί να ελεγχθεί με τη χρησιμοποίηση στη συνέχεια μιάς έντολης διακλαδώσεως με συνθήκη π.χ. την έντολή RC που θα δούμε στα επόμενα. Στην έντολή αυτή ο-

¹ Οι καταχωριστές που διαθέτει ο υπολογιστής αριθμούνται απόλυτα από τό 1-16.

ρίζονται τα 4 από τα bits της (8-11) για να δηλωθεί η τιμή του δείκτη που μας ενδιαφέρει. Από τα 4 αυτά bits το πρώτο ελέγχει αν η τιμή του δείκτη είναι 0 (όποτε η τιμή του 1ου bit θα είναι 1), το δεύτερο αν η τιμή του δείκτη είναι 1, το τρίτο αν είναι 2 και το τέταρτο αν είναι 3.

Διακλάδωση θα γίνει όταν η τιμή του bit, στην οποία αντιστοιχεί η κατάσταση όπου βρίσκεται ο δείκτης συνθήκης, είναι 1, π.χ. "Αν η κατάσταση του δείκτη συνθήκης είναι 2 και η τιμή των υπό αριθ. 8-11 bits της έντολης είναι 1010, θα γίνει η διακλάδωση. "Αν όμως η τιμή αυτών των bits είναι 1000, δεν θα γίνει διακλάδωση.

Η αντιστοιχία μεταξύ καταστάσεων δείκτη και περιεχομένου των 4 bits (8-11) φαίνεται στον Πίνακα 8.6.1.

ΠΙΝΑΚΑΣ 8.6.1.

Μορφή των bits 8-11 της έντολης διακλάδωσης	Δεκαδική τιμή	Έλεγχομένη τιμή του δείκτη	Έλεγχομένη συνθήκη μεταξύ δύο δρων A και B
0000	0	Καμία	
0001	1	3	
0010	2	2	$A > B$
0011	3	2 ή 3	
0100	4	1	$A < B$
0101	5	1 ή 3	
0110	6	1 ή 2	$A \neq B$
0111	7	1 ή 2 ή 3	
1000	8	0	$A = B$
1001	9	0 ή 3	
1010	10	0 ή 2	$A \geq B$
1011	11	0 ή 2 ή 3	
1100	12	0 ή 1	$A \leq B$
1101	13	0 ή 1 ή 3	
1110	14	0 ή 1 ή 2	
1111	15	0 ή 1 ή 2 ή 3	Διακλάδωση άνευ δρων

Μετά από την σύντομη αυτή παρεμβολή, ξαναγυρίζουμε στην περιγραφή των έντολων τύπου RR.

γ) Έντολη συγκρίσεως (CR).

Η μορφή της είναι:

CR R_1, R_2

Με την έντολη αυτή συγκρίνονται αλγεβρικά, τα περιεχόμενα των καταχωριστών R_1 και R_2 . Από το αποτέλεσμα της συγκρίσεως ο δείκτης συνθήκης παίρνει ανάλογη τιμή (0 αν $[R_1] = [R_2]^1$, 1 αν

¹ Θυμίζουμε ότι ο συμβολισμός $[R_2]$ σημαίνει το περιεχόμενο του καταχωριστή R_2 .

$[R_1] < [R_2]$ και 2 αν $[R_1] > [R_2]$). 'Η έντολή αυτή συνδυάζεται με την επόμενη που θα περιγράψουμε.
Π.χ.:

CR 6, 7

συγκρίνονται τα περιεχόμενα των καταχωριστών 6 και 7. 'Αν $[6] = [7]$, ο δείκτης συνθήκης θά πάρει τιμή 0, αν $[6] < [7]$, θά πάρει τιμή 1 και αν $[6] > [7]$ θά πάρει τιμή 2.

δ) Έντολή διακλαδώσεως υπό συνθήκη (BCR).

'Εχει τή μορφή:

BCR M, R₃

Προκαλεί διακλάδωση στη διεύθυνση που περιέχεται στον καταχωριστή R₂, όταν ικανοποιείται ή συνθήκη που περιγράφει ο M, που είναι ο αριθμός - τιμή του δείκτη συνθήκης. Διαφορετικά εκτελείται η έντολή που ακολουθεί την BCR.

Π.χ.:

CR 4, 5

BCR 8, 10

Με την πρώτη από τις δύο αυτές έντολές συγκρίνομε τό [4] με τό [5]. Με τη δεύτερη δηλώνομε ότι σέ περίπτωση που $[4] = [5]$ νά εκτελεσθεί σάν επόμενη, η έντολή που βρίσκεται στη διεύθυνση που ίσοῦται με τό περιεχόμενο του καταχωριστή 10.

ε) Έντολή διακλαδώσεως και συνδέσεως (BALR).

'Εχει τή μορφή:

BALR R₁, R₂

'Η έντολή αυτή λειτουργεί ως εξής: Τοποθετείται στον καταχωριστή R₁ η διεύθυνση της επόμενης, σέ φυσική σειρά έντολης και γίνεται διακλάδωση στην έντολή με διεύθυνση τό [R₂]. 'Αν τό R₂ είναι 0, δέν γίνεται διακλάδωση.

Π.χ.:

BALR 8, 9

SUM AR 1, 2

Με την πρώτη από τις δύο έντολές καταχωρίζεται στον register 8 η διεύθυνση SUM της επόμενης έντολης και γίνεται διακλάδωση στη διεύθυνση που περιέχεται στον R₂.

ζ) Έντολή μεταφοράς (LR).

'Η έντολή αυτή έχει τή μορφή:

LR R₁, R₂

καί με τή βοήθειά της, μεταφέρεται τό περιεχόμενο του R₂ στον R₁, ενώ ο R₂ εξακολουθεί νά διατηρεί τήν τιμή του. Π.χ.:

LR 3, 9

μεταφέρεται τό [9] στον καταχωριστή 3. 'Ο καταχωριστής 9 διατηρεί τή τιμή του.

η) Έντολή λογικής συγκρίσεως (CLR).

'Εχει τή μορφή:

CLR R₁, R₂

Με αυτήν συγκρίνονται τα περιεχόμενα των καταχωριστών R_1 και R_2 . Ο δείκτης συνθήκης παίρνει τιμή ανάλογη με το αποτέλεσμα της συγκρίσεως.

θ) Έντολή μετατροπής αρνητικών περιεχομένων σε θετικά (LPR).

Έχει τη μορφή:

LPR R_1, R_2

Λειτουργεί ως εξής: Αν το περιεχόμενο των R_1 και R_2 είναι αρνητικό, βρίσκει το συμπλήρωμα του καθενός και το ξαναποθετεί στον καθένα καταχωριστή.

ι) Έντολή (SVC).

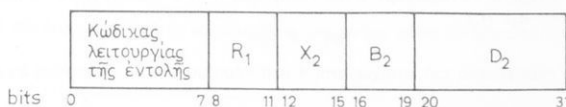
Έχει τη μορφή:

SVC, I

όπου I άκεραίος. Είναι συνήθως η τελευταία έντολή ενός προγράμματος. Με αυτή ο έλεγχος του προγράμματος επιστρέφεται στο έποπτευον πρόγραμμα (Supervisor).

8.7 Έντολές τύπου RX.

Κάθε έντολή του τύπου αυτού καταλαμβάνει 32 bits μνήμης και καταχωρίζεται ως εξής:



Στα 8 πρώτα bits (0-7) καταχωρίζεται ο κώδικας λειτουργίας της έντολης. Στα επόμενα 4 bits (8 — 11) καταχωρίζεται ο αριθμός του καταχωριστή, που αποτελεί τον πρώτο όρο μιάς πράξεως (π.χ. προσθέσεως, αφαιρέσεως κλπ), της οποίας ο δεύτερος όρος είναι το περιεχόμενο εκείνης της θέσεως της μνήμης, που η διεύθυνσή της βρίσκεται, αν προστεθεί το άθροισμα των περιεχομένων των καταχωριστών X_2 και B_2 στην τιμή D_2 , δηλαδή $D_2 + [X_2] + [B_2]$.

α) Έντολή προσθέσεως (A).

Έχει τη μορφή:

A $R_1, D_2 (X_2, B_2)$

Με αυτή προστίθεται το περιεχόμενο της θέσεως στη μνήμη με διεύθυνση $D_2 + [X_2] + [B_2]$, στο περιεχόμενο του καταχωριστή R_1 . Π.χ.:

A 10, POS

Με την έντολή αυτή προστίθεται το περιεχόμενο της διεύθυνσεως POS στο περιεχόμενο του καταχωριστή 10.

β) Έντολή συγκρίσεως (C).

Έχει τη μορφή:

C $R_1, D_2 (X_2, B_2)$

και η λειτουργία της είναι ανάλογη προς τη λειτουργία της έντολης CR. Π.χ.:

C 8, TIMH

Συγκρίνεται το περιεχόμενο του καταχωριστή 8 με το περιεχόμενο της θέσεως της μνήμης, που έχει τη συμβολική διεύθυνση TIMH. Ο δείκτης συνθήκης παίρνει τιμή ανάλογη με το αποτέλεσμα της σύγκρισης.

γ) Έντολή διακλαδώσεως με συνθήκη (BC).

Έχει τη μορφή:

$$BC \quad M_1, D_2 (X_2, B_2)$$

καί μοιάζει με την έντολή BCR. Π.χ.:

$$BC \quad 4, \text{FOROS}$$

Με αυτήν γίνεται διακλάδωση στη διεύθυνση FOROS, αν η τιμή του δείκτη συνθήκης είναι 4 (σάν αποτέλεσμα από μία σύγκριση που προηγήθηκε).

δ) Έντολή μεταφοράς (L).

Η μορφή της είναι:

$$L \quad R_1, D_2 (X_2, B_2)$$

καί είναι ανάλογη προς την LR. Π.χ.:

$$L \quad 6, \text{TOKOS}$$

με αυτή μεταφέρεται το περιεχόμενο της διεύθυνσεως μνήμης TOKOS στον καταχωριστή 6.

ε) Έντολή μεταφοράς στη μνήμη (ST).

Η μορφή της είναι:

$$ST \quad R_1, D_2 (X_2, B_2)$$

Με αυτή μεταφέρεται το περιεχόμενο του καταχωριστή R_1 στη θέση της μνήμης με συμβολική διεύθυνση $D_2 (X_2, B_2)$. Π.χ.

$$ST \quad 4, \text{ELP}$$

μεταφέρεται το περιεχόμενο του καταχωριστή 4 στη θέση μνήμης με συμβολική διεύθυνση ELP.

στ) Έντολή αφαιρέσεως (S).

Έχει τη μορφή:

$$S \quad R_1, D_2 (X_2, B_2)$$

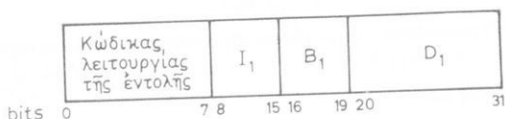
Με αυτήν αφαιρείται το περιεχόμενο της μνήμης με διεύθυνση $D_2 (X_2, B_2)$. Π.χ.:

$$S \quad 8, \text{MEG}$$

αφαιρείται το περιεχόμενο της θέσεως μνήμης, με συμβολική διεύθυνση MEG, από το περιεχόμενο του καταχωριστή 8.

8.8 Έντολές τύπου RS.

Κάθε έντολή του τύπου αυτού καταλαμβάνει χώρο 32 bits καί καταχωρίζεται ως εξής:



Στά 8 πρώτα bits καταχωρίζεται ο κώδικας λειτουργίας της έντολης. R_1 καί R_2 είναι δύο καταχωριστές. Στη λειτουργία της έντολης μετέχουν με τα περιεχόμενά τους όλοι οι καταχωριστές, από τον R_1 ως τον R_2 , καθώς καί η διεύθυνση μνήμης $[B_2] + D_2$.

α) Έντολή πολλαπλής ταυτόχρονης καταχωρήσεως (LM).

Η μορφή της είναι:

$$LM \quad R_1, R_3, D_2 (B_2)$$

Μέ την έντολή αυτή μεταφέρονται και καταχωρίζονται στους καταχωριστές από τον R_1 μέχρι τον R_3 (και όλους τους ενδιάμεσους), τα περιεχόμενα ίσαριθμων θέσεων της μνήμης αρχίζοντας από τη διεύθυνση $D_2(B_2)$ και πέρα. Π.χ.:

$$LM \quad 4,7, 1080$$

που σημαίνει ότι μέσα στους καταχωριστές 4 έως 7 δηλ. (4, 5, 6, 7) τοποθετούνται τα περιεχόμενα 4 θέσεων της μνήμης, από τη διεύθυνση 1080 και πέρα. Δηλαδή το περιεχόμενο της διεύθυνσης 1080 θα τοποθετηθεί στον καταχωριστή 4, της 1081 στον 5 κ.ο.κ.

β) Έντολή πολλαπλής αποθηκεύσεως στη μνήμη (STM).

Η μορφή της είναι:

$$STM \quad R_1, R_2, D_2 (B_2)$$

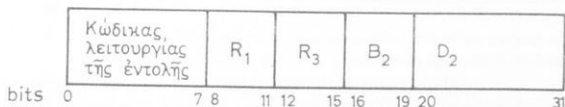
καί είναι αντίστροφη της LM, δηλαδή μεταφέρονται τα περιεχόμενα των καταχωριστών R_1 έως και R_3 σε διαδοχικές θέσεις της μνήμης, αρχίζοντας από τη διεύθυνση $D_2 (B_2)$. Π.χ.:

$$STM \quad 4,7 \text{ AMAR}$$

μεταφέρεται το περιεχόμενο των καταχωριστών 4,5,6, και 7 διαδοχική στίς θέσεις της μνήμης, αρχίζοντας από τη διεύθυνση AMAR. Δηλαδή στη θέση AMAR θα μεταφερθεί το περιεχόμενο του καταχωριστή 4, στην επόμενη της AMAR θέση θα μεταφερθεί το περιεχόμενο του 5 κ.ο.κ.

8.9 Έντολές τύπου SI.

Μιά έντολή της κατηγορίας αυτής καταλαμβάνει στη μνήμη χώρο 32 bits και καταχωρίζεται ως εξής:



Όπου I_1 είναι μία αριθμητική σταθερή. Στη λειτουργία της έντολης συμμετέχει και το περιεχόμενο της θέσεως της μνήμης με διεύθυνση $D_1 + [B_1]$.

α) Έντολή προσθέσεως σταθερής (AI).

Έχει τη μορφή:

$$AI \quad D_1 (B_1), I_1$$

μέ αυτή, η σταθερή I_1 (μήκους 1 byte) προστίθεται στο περιεχόμενο της ημιλέξεως με διεύθυνση $D_1 + [B_1]$ Π.χ.:

$$AI \quad 0(6), 40$$

προστίθεται ο αριθμός 40 στο περιεχόμενο της μνήμης με διεύθυνση 0 + [6].

β) Έντολή συγκρίσεως (CLI).

Έχει τη μορφή:

$$CLI \quad D_1 (B_1), I_1$$

καί γίνεται σύγκριση ανάμεσα στο περιεχόμενο της διεύθυνσης $D_1 + [B_1]$ της μνήμης και της σταθερής I_1 . Π.χ.:

$$CLI \quad \text{POSON}, 86$$

συγκρίνεται τό περιεχόμενο τής διευθύνσεως POSON με τή σταθερή 86.

γ) Έντολή διακοπής εκτελέσεως του προγράμματος (H).

Έχει τή μορφή:

$$H \quad D_1 \quad (B_1)$$

καί με αὐτήν προκαλεῖται διακοπή στήν ἐκτέλεση ἑνός προγράμματος. Ἄν ὁ χειριστής πιέσει τό πλήκτρο START στόν πίνακα ἐλέγχου τότε θά συνεχισθεῖ ἡ ἐκτέλεση τοῦ προγράμματος ἀπό τήν ἐντολή πού βρίσκεται στή διεύθυνση. $D_1 + [B_1]$ τῆς μνήμης.

δ) Έντολή μεταφορᾶς σταθερῆς (MVI).

Ἡ μορφή τῆς εἶναι:

$$MVI \quad D_1 \quad (B_1), I_1$$

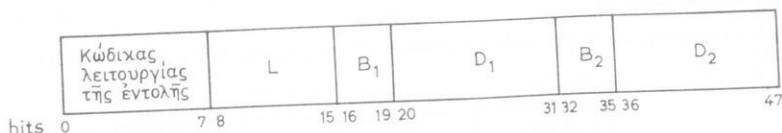
καί με αὐτήν μεταφέρεται ὁ ἀριθμός I_1 στή θέση πού ὀρίζεται ἀπό τή διεύθυνση $D_1 + [B_1]$. Π.χ.

$$MVI \quad BAT, 28$$

μεταφέρεται ὁ ἀριθμός 28 στή θέση BAT τῆς μνήμης.

8.10 Έντολές τύπου SS1.

Κάθε ἐντολή τῆς κατηγορίας αὐτῆς καταλαμβάνει χῶρο 1.5 λέξεως (6 bytes) καί καταχωρίζεται ὡς ἑξῆς:



Στή λειτουργία τῆς ἐντολῆς συμμετέχουν τά περιεχόμενα τῶν ἰσομήκων θέσεων τῆς μνήμης με διευθύνσεις $D_1 + [B_1]$ καί $D_2 + [B_2]$. Τό L παριστάνει τό κοινό μέγεθος σέ bytes.

α) Έντολή συγκρίσεως (CLC).

Έχει τή μορφή:

$$CLC \quad D_1 \quad (L, B_1), D_2 \quad (L, B_2)$$

Με αὐτήν συγκρίνονται τά περιεχόμενα τῶν θέσεων τῆς μνήμης με διευθύνσεις $D_1 \quad (B_1)$ καί $D_2 \quad (B_2)$. Π.χ.:

$$CLC \quad ROM, CIP$$

συγκρίνονται τά περιεχόμενα τῶν διευθύνσεων ROM καί CIP τῆς μνήμης.

β) Έντολή μεταφορᾶς (MVC).

Έχει τή μορφή:

$$MVC \quad D_1 \quad (L, B_1), D_2 \quad (L, B_2)$$

Με αὐτήν, μεταφέρεται τό περιεχόμενο τῆς περιοχῆς με διεύθυνση $D_2 \quad (B_2)$ στήν περιοχή με διεύθυνση $D_1 \quad (B_1)$. Π.χ.:

$$MVC \quad ARAG \quad (30), KATAP$$

μεταφέρονται 30 bytes ἀπό τήν περιοχή KATAP στήν περιοχή ARAG. Ἡ περιοχή ARAG πρέπει, ὡς ἐκ τούτου, νά ἔχει μέγεθος 30 bytes.

8.11 Έντολές τύπου SS2.

Οι έντολές αυτές αναφέρονται σε πράξεις ανάμεσα στα περιεχόμενα δύο άνισομήκων περιοχών της μνήμης. Θα τις παραλείψουμε όμως λόγω της πολυπλοκότητάς τους.

8.12 Ψευδοέντολές (Pseudoinstructions).

Οι έντολές αυτές είναι βοηθητικές και χρησιμοποιούνται για να δώσουν στον υπολογιστή τις απαραίτητες γενικές οδηγίες, για τη λειτουργία ενός προγράμματος.

Τις κυριότερες από αυτές θα δούμε άμέσως.

α) Έντολή DC.

Με την έντολή αυτή μπορούμε να τοποθετήσουμε δεδομένα σε θέσεις της μνήμης, από όπου θα τα χρησιμοποιήσουμε στο πρόγραμμά μας. Η τοποθέτηση αυτή γίνεται στη φάση της μεταφράσεως του προγράμματος.

Η μορφή της είναι:

$$DC \quad a_1 a_2$$

όπου a_1, a_2 προδιαγραφές που θα δούμε άμέσως.

Με την προδιαγραφή a_1 περιγράφουμε τον τύπο της πληροφορίας που πρόκειται να καταχωρισθεί σε μία θέση της μνήμης. Έτσι, αν τα δεδομένα αποτελούνται μόνο από αλφαριθμητικούς χαρακτήρες, γράφουμε στη θέση της a_1 το χαρακτήρα C. Στην περίπτωση αυτή η καταχώριση στο πεδίο γίνεται από την αρχή προς το τέλος του (από άριστερά προς τα δεξιά). Αν τα δεδομένα είναι δεκαδικοί αριθμοί, που θέλουμε να καταχωρισθούν σε δυαδική μορφή, τότε θα γράψουμε στη θέση του a_1 το χαρακτήρα F ή H (F όταν ο αριθμός καταχωρίζεται σε περιοχή μήκους 4 bytes και H όταν η καταχώριση γίνεται σε χώρο 2 bytes).

Με την προδιαγραφή a_2 περιγράφουμε το ίδιο το δεδομένο, κλείνοντάς το ανάμεσα σε άποστρόφους.

Παραδείγματα.

α) DATA1 DC F '2'

Σημαίνει ότι θα καταχωρισθεί στη θέση μνήμης με συμβολική διεύθυνση DATA1, ο δεκαδικός αριθμός 2 σε δυαδική μορφή. Η καταχώριση θα γίνει σε χώρο 4 bytes.

β) SUM DC H '72'

Σημαίνει ότι θα καταχωρισθεί στη θέση της μνήμης με συμβολική διεύθυνση SUM ο δεκαδικός 72 σε δυαδική μορφή. Για τη καταχώρισή του θα διατεθούν 2 bytes.

γ) ALPHA DC C 'A'

Θα καταχωρισθεί στη θέση της μνήμης με συμβολική διεύθυνση ALPHA ο χαρακτήρας A.

β) Έντολή DS.

Με αυτή καθορίζουμε απλώς μία περιοχή της μνήμης ορισμένου μήκους, χωρίς να καταχωρίσουμε σε αυτή οποιαδήποτε δεδομένα. Η μορφή της μοιάζει κατά τα άλλα με τη μορφή της DC. Π.χ.

α) DEC DS F

Σημαίνει ότι ορίζουμε μία περιοχή της μνήμης με συμβολική διεύθυνση DEC, μήκους 4 bytes.

β) DIM DS H

Σημαίνει ότι ορίζουμε μία περιοχή της μνήμης με συμβολική διεύθυνση DIM, μεγέθους 2 bytes.

γ) Έντολή START.

Με αυτήν καθορίζουμε τη διεύθυνση της μνήμης, από όπου αρχίζει η καταχώριση του προγράμματος. Έχει τη μορφή:

$$START \quad n$$

όπου: n ή δεκαδικά έκφρασμένη διεύθυνση της μνήμης της πρώτης έντολής του προγράμματος. Π.χ.:

$$START \quad 256$$

σημαίνει ότι η καταχώριση του προγράμματος στη μνήμη, θα αρχίσει από τη διεύθυνση (256)₁₆ ή (100)₁₆.

δ) Έντολή USING.

Με αυτή καθορίζεται ο καταχωριστής-βάση¹ για το συγκεκριμένο πρόγραμμα. Έχει τη μορφή:

USING X₁, R₁

R₁ είναι ο καταχωριστής-βάση και X₁ η τιμή που καταχωρίζουμε σ' αυτόν σαν περιεχόμενο. Π.χ. USING 200,8

Σημαίνει ότι σαν καταχωριστή-βάση θα χρησιμοποιήσουμε τον R₈, καταχωρίζουμε δέ σ' αυτόν την τιμή 200.

USING *,15

Σημαίνει ότι σαν καταχωριστή-βάση θα χρησιμοποιήσουμε τον καταχωριστή 15 και ότι θα καταχωρίσουμε σ' αυτόν το περιεχόμενο του καταχωριστή διευθύνσεων.

ε) Έντολή END.

Είναι η τελευταία έντολή ενός Assembly προγράμματος. Έχει τη μορφή:

END D₁

όπου D₁ είναι η διεύθυνση της έντολης, από όπου θα αρχίσει την εκτέλεσή του το πρόγραμμα Assembly. Π.χ.:

END BEGIN

όπου BEGIN είναι η συμβολική διεύθυνση της έντολης του προγράμματος, που θα εκτελεσθεί πρώτη.

στ) Έντολή TITLE.

Με αυτή προσδιορίζεται το όνομα του προγράμματος, το οποίο θα αναγραφεί στην αρχή κάθε σελίδα των εκτυπωμένων εντολών του προγράμματος (Listing). Το όνομα το γράφουμε ανάμεσα σέ άποστρόφους, μπορεί δέ νά αποτελείται από πολλούς χαρακτήρες (θεωρητικά μέχρι 100). Π.χ.:

TITLE 'EXAMPLE ONE'

Έδω ο τίτλος του προγράμματος είναι EXAMPLE ONE και θα εκτυπώνεται στην αρχή κάθε σελίδας των εκτυπωμένων εντολών του προγράμματος.

8.13 Μακροεντολές (Macros).

Χρησιμοποιούνται κυρίως σέ προγράμματα έπεξεργασίας αρχείων. Με τόν όρο έπεξεργασία αρχείου περιλαμβάνονται λειτουργίες όπως:

- Ανοιγμα ενός αρχείου, δηλαδή ή κατάλληλη προετοιμασία του, ώστε νά μπορεί νά χρησιμοποιηθεί από τό πρόγραμμά μας γιά νά διαβασθοῦν ή νά γραφοῦν πληροφορίες σ' αυτό.
- Ανάγνωση πληροφοριών από ένα αρχείο.
- Έγγραφή πληροφοριών σέ ένα αρχείο.
- Κλείσιμο ενός αρχείου, δηλαδή έπαναφορά στην αρχική του κατάσταση (στην κατάσταση που βρισκόταν πρίν αρχίσει νά χρησιμοποιείται στό πρόγραμμα).

Με τόν όρο μακροεντολή έννοούμε μία προηγμένη έντολή που κατά τή φάση τής μεταφράσεως θα αναλυθεί σέ περισσότερες από μία έντολές γλώσσας μηχανής.

¹ Ο καταχωριστής — βάση (Base Register) είναι ένας από τούς καταχωριστές που διαθέτει ο υπολογιστής. Τόν χρησιμοποιούμε γιά νά καταχωρίσουμε σ' αυτόν κάποιο σταθερό αριθμό. Ο καταχωριστής αυτός δανείζει τό περιεχόμενό του στίς διάφορες έντολές, που είδαμε στά προηγούμενα (όπου τόν αναφέραμε σαν B) γιά τόν σχηματισμό τής διευθύνσεως μιάς θέσεως στή μνήμη.

Οι κυριότερες από τις μακροεντολές της Assembly είναι οι εξής:

α) Έντολή OPEN.

Με αυτή ανοίγεται ένα αρχείο. Έχει τη μορφή:

OPEN Name

όπου Name είναι το συμβολικό όνομα του αρχείου. Π.χ.:

OPEN PAYROL

μέ την έντολή αυτή ανοίγεται το αρχείο με την ονομασία PAYROL.

β) Έντολή GET.

Με αυτή διαβάζονται πληροφορίες από κάποιο αρχείο. Η μορφή της είναι:

GET Name, D₁, D₂

Name είναι το όνομα του αρχείου, από όπου διαβάζονται οι πληροφορίες, D₁ ή διεύθυνση της μνήμης, όπου θα γίνει η καταχώριση των πληροφοριών που διαβάζονται και D₂ ή διεύθυνση της έντολης, όπου θα μεταφερθεί ο έλεγχος του προγράμματος, όταν εξαντληθούν οι πληροφορίες που περιέχει το αρχείο Π.χ.:

GET RAG, SIM, TELOS

σημαίνει ότι διαβάζονται πληροφορίες από το αρχείο RAG, των οποίων η καταχώριση αρχίζει από τη διεύθυνση της μνήμης SIM. Όταν οι πληροφορίες του RAG εξαντληθούν θα μεταφερθεί ο έλεγχος του προγράμματος στην έντολή με συμβολική διεύθυνση TELOS.

γ) Έντολή έγγραφης PUT.

Με αυτή γράφουμε πληροφορίες σε ένα αρχείο. Μοιάζει με την έντολή GET και έχει μορφή:

PUT Name, D₁, D₂

Name είναι το όνομα του αρχείου, D₁ ή διεύθυνση μνήμης, όπου βρίσκονται οι πληροφορίες που θα εγγραφούν στο αρχείο και D₂ ή διεύθυνση της έντολης, που θα εκτελεσθεί, όταν ανιχνευθεί το φυσικό τέλος του φορέα του αρχείου (π.χ. τέλος της σελίδας του έκτυπτωτή, τέλος της ταινίας κ.λπ.) Π.χ.

PUT RAG, DIM, K1

δ) Έντολή CLOSE.

Με αυτή κλείνουμε ένα αρχείο. Έχει τη μορφή:

CLOSE Name

όπου Name είναι το συμβολικό όνομα του αρχείου που κλείνουμε. Π.χ.:

CLOSE PAYROL

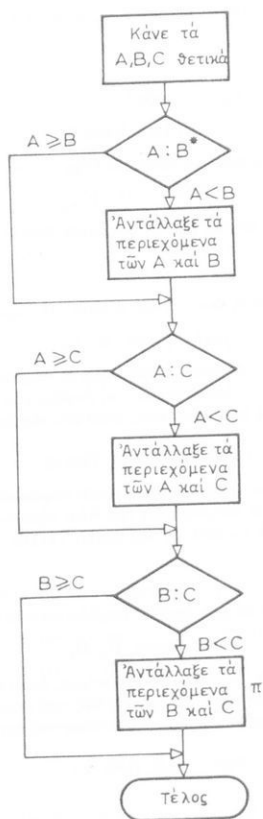
και σημαίνει ότι κλείνουμε το αρχείο που έχει την ονομασία PAYROL.

8.14 Παραδείγματα προγραμμάτων ASSEMBLY.

Παράδειγμα 1.

Έστω ότι δίνονται τρεις αριθμοί A, B, C θετικοί ή άρνητικοί. Ζητείται να γραφεί πρόγραμμα Assembly, που θα μετατρέπει τους άρνητικούς (αν υπάρχουν) σε θετικούς και θα τους τακτοποιεί έτσι, ώστε ο A να περιέχει το μεγαλύτερο, ο B τόν άμεσα μικρότερο και ο C τόν πιο μικρό.

Το διάγραμμα ροής στην περίπτωση αυτή θα έχει την εξής μορφή (σχ. 8.14α).



* Τὸ $A:B$ σημαίνει σύγκριση τοῦ A μέ τὸ B .

Σχ. 8.14α.

Τὸ πρόγραμμα θά ἔχει τήν παρακάτω μορφή (σχ. 8.14β, 8.14γ).

PROGRAM

PROGRAMMER

DATE

1	Name	8	10	Operation	14	16	20	Operand	25	30
				S T A R T		2	5	6		
B	E	G	I	N		B	A	L	R	
				U	S	I	N	G		
						*		1	5	
				L	M			1		3
										A
				L	P	R		1		1
				L	P	R		2		2
				L	P	R		3		3
				C	R			1		2
				B	C			1	0	
										C O M P 2
				L	R			6		1
				L	R			1		2
				L	R			2		6
C	O	M	P	2		C	R			1
										3
				B	C			1	0	
										C O M P 3
				L	R			6		1
				L	R			1		3
				L	R			3		6
C	O	M	P	3		C	R			2
										3
				B	C			1	0	
										O U T
				L	R			6		2
				L	R			2		3
				L	R			3		6
O	U	T			S	T	M			1
										3
										A
				S	V	C				0

Σχ. 8.14β.

PROGRAM

PROGRAMMER

DATE

Name								Operation							Operand																																									
1								8							10							14							16							20							25							30						
								S T A R T							2 5 6																																									
B E G I N								B A L R							1 5 , 0																																									
								U S I N G							* , 1 5																																									
								S R							8 , 8																																									
								S R							1 1 , 1 1																																									
L O O P								A							8 , T A B L E (1 1)																																									
								A							1 1 , C 4																																									
								C							1 1 , C 8 0																																									
								B C							4 , L O O P																																									
								S T							8 , S U M																																									
								S V C							0																																									
T A B L E								D C							F ' 1 \																																									
								D C							F ' 2 \																																									
								D C							F ' 3 \																																									
								D C							F ' 4 \																																									
								D C							F ' 5 \																																									
								D C							F ' 6 \																																									
								D C							F ' 7 \																																									
								D C							F ' 8 \																																									
								D C							F ' 9 \																																									
								D C							F ' 1 0 \																																									
								D C							F ' 1 1 \																																									
								D C							F ' 1 2 \																																									
								D C							F ' 1 3 \																																									
								D C							F ' 1 4 \																																									

Σχ. 8.145.

PROGRAM	DATE
PROGRAMMER	

Name								Operation								Operand													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
									D	C					F	'	1	5	'										
									D	C					F	'	1	6	'										
									D	C					F	'	1	7	'										
									D	C					F	'	1	8	'										
									D	C					F	'	1	9	'										
									D	C					F	'	2	0	'										
									D	S					F														
									D	C					F	'	4	'											
									D	C					F	'	8	0	'										
									E	N	D				B	E	G	I	N										

Σχ. 8.14ε.

Άσκησης.

- Νά γραφεί πρόγραμμα Assembly για τον υπολογισμό του αθροίσματος των τετραγώνων των 20 πρώτων φυσικών αριθμών.
- Νά γραφεί πρόγραμμα Assembly για τον υπολογισμό της τιμής της συναρτήσεως $y = 3x^2 + 4$ για $x = 1.5$
- Νά γραφεί πρόγραμμα Assembly για τον υπολογισμό του μέσου όρου 4 αριθμών A,B,C,D.
- Νά γραφεί πρόγραμμα Assembly για τον υπολογισμό της τιμής της συναρτήσεως $y = 2x^2 - 3x + 5$ για $x = 2.3$
- Νά γραφεί πρόγραμμα Assembly για τον υπολογισμό της τιμής της παραστάσεως $4A + 5B + 6/C + 7D$ όπου $A = 1.1$, $B = 2.0$, $C = 3.4$ και $D = 0.7$.

ΚΕΦΑΛΑΙΟ ΕΝΑΤΟ

Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ BASIC

9.1 Γενικά.

Η BASIC είναι ή πιά άπλή καί ή πιά εύχρηστη από τίς γλώσσες προγραμματισμού ύψηλου επίπεδου. Παρουσιάζει αρκετές ομοιότητες μέ τή γλώσσα FORTRAN πού θά γνωρίσoμε στό επόμενο κεφάλαιο.

Η όνομασία της προέρχεται από τά αρχικά των Άγγλικών λέξεων Beginner's All-Purpose Symbolic Instruction Code, πού σημαίνουν σέ ελεύθερη μετάφραση, Συμβολικός Κώδικας Έντολών γιά αρχάριους, έπειδή σχεδιάσθηκε αρχικά γιά νά έξυπηρετήσει εκπαιδευτικά προγράμματα, όπου οι σπουδαστές επικοινωνούσαν μέ τόν ύπολογιστή μέσω τηλετύπων (Teletypes).

Αποτελείται από έντολές πού οι κωδικοί τους είναι λέξεις τής Άγγλικής γλώσσας. Τίς έντολές, πού συνιστούν ένα πρόγραμμα, τίς γράφομε τή μία κάτω από τήν άλλη ακολουθώντας τή σειρά του λογικού διαγράμματος, ή σχεδίαση του οποίου έχει προηγηθεί.

9.2 Βασικά στοιχεία τής BASIC.

Έξι βασικά στοιχεία τής BASIC θά γνωρίσoμε στην παράγραφο αυτή, τά εξής:

α) Άριθμοί γραμμών ή αριθμοί έντολών (Line numbers ή statement numbers).

Κάθε γραμμή πού περιλαμβάνει μία έντολή στό πρόγραμμα BASIC πρέπει νά αρχίζει μέ ένα μοναδικό αριθμό, πού νά αποτελείται από 5 τό πολύ δεκαδικά ψηφία, δηλαδή από τό 1 έως τό 99999. Ό ύπολογιστής εκτελεί τίς έντολές του προγράμματος μέ τή σειρά πού καθορίζεται από τους αριθμούς γραμμών.

β) Σταθερές (Constants).

Είναι θετικοί ή άρνητικοί δεκαδικοί αριθμοί, άκέραιοι ή κλασματικοί όπως π.χ.:

43 -172 4.546 0.0719 -65.080 κ.ο.κ.

καί εμφανίζονται μέσα σέ μία έντολή (κυρίως αντικαταστάσεις), άκριβώς όπως καί σέ μία άπλή άλγεβρική παράσταση. Άριθμοί χωρίς πρόσημο θεωρούνται θετικοί. Έκφράσεις όπως $16/3$, $-7/5$ δέν θεωρούνται άπλές σταθερές, αλλά ότι αποτελούνται από δύο σταθερές.

Έπειδή οι ύπολογιστές δέχονται αριθμούς μέ περιορισμένο πλθος ψηφίων, τους πολύ μεγάλους ή πολύ μικρούς αριθμούς είναι πιά βολικό νά τους εκφράζoμε σέ έκθετική μορφή. Στή μορφή αυτή ο αριθμός αποτελείται από δύο τμήματα, τό

ψηφιακό τμήμα και τό τμήμα του ἐκθέτη. Τό τμήμα του ἐκθέτη περιλαμβάνει τό γράμμα E πού συμβολίζει τή βάση 10 καί ἕνα ἀκέραιο ἀριθμό πού εἶναι ἡ δύναμη τοῦ 10 π.χ.:

$$\begin{array}{rcl} \underbrace{1.74}_{\text{Ψηφιακό Τμήμα}} \underbrace{E + 4}_{\text{Τμήμα Ἐκθέτη}} & = & 1,74 \times 10^{14} \\ -0,098 E - 2 & = & -0,098 \times 10^{-2} \\ 17E3 & = & 17 \times 10^3 \quad \text{κ.ο.κ.} \end{array}$$

γ) Μεταβλητές (Variables).

Οἱ μεταβλητές τόσο στή BASIC ὥς καί στίς ἄλλες γλώσσες προγραμματισμοῦ παίξουν τό ρόλο τῶν συμβολικῶν διευθύνσεων τῆς μνήμης.

Οἱ ὀνομασίες τῶν μεταβλητῶν σχηματίζονται ἐδῶ μέ ἕνα κεφαλαῖο χαρακτήρα τοῦ λατινικοῦ ἀλφαβήτου, πού μπορεῖ νά ἀκολουθεῖται καί ἀπό ἕνα ἀριθμητικό χαρακτήρα, π.χ. οἱ μεταβλητές:

A, B, J, X5, A1 κλπ.

εἶναι δεκτές στή BASIC. Οἱ μεταβλητές:

AB, X22, 4A, κλπ.

δέν εἶναι δεκτές.

Κάθε μεταβλητή παριστάνει, ὅπως εἴπαμε, μιά θέση μνήμης, στήν ὁποία δίνεται κάποιο ἀριθμητικό περιεχόμενο εἴτε μέ τή βοήθεια μιᾶς ἐντολῆς ἀναγνώσεως, εἴτε μιᾶς ἐντολῆς ἀντικαταστάσεως, ὅπως θά δοῦμε παρακάτω.

δ) Σύμβολα ἀριθμητικῶν πράξεων.

Τά σύμβολα πού χρησιμοποιοῦμε στή BASIC γιά νά ἀπεικονίσουμε τίς διάφορες ἀριθμητικές πράξεις ἀνάμεσα σέ δύο δεδομένα, φαίνονται στόν Πίνακα 9.2.1.

ΠΙΝΑΚΑΣ 9.2.1.

Τά σύμβολα τῶν ἀριθμητικῶν πράξεων τῆς BASIC

Ἀριθμητική πράξη	Σύμβολο τῆς Ἀριθμητικῆς	Σύμβολο τῆς BASIC
Πρόσθεση	+	+
Ἀφαίρεση	-	-
Πολλαπλασιασμός	· ἢ ×	*
Διαίρεση	: ἢ /	/
Ὑψωση σέ δύναμη	δέν ὑπάρχει	**

ε) Κωδικοποίηση ἀριθμητικῶν καί ἀλγεβρικῶν ἐκφράσεων.

Στήν ἀλγεβρα συχνά χρησιμοποιοῦμε μικροῦς ἐλληνικοῦς ἀλφαβητικούς χαρακτήρες γιά νά παριστάνουμε τοὺς ἀγνωστους μιᾶς ἐξισώσεως ἢ τίς παραμέτρους καί τίς ἀνεξάρτητες μεταβλητές μιᾶς συναρτήσεως ἢ τοὺς συντελεστές ἑνός πολυωνύ-

μου κλπ. Σέ όλες τίς γλώσσες προγραμματισμοῦ χρησιμοποιοῦμε, ὅπως εἶπαμε, μέ ἀνάλογο τρόπο, μόνο κεφαλαία γράμματα τοῦ λατινικοῦ ἀλφαβήτου, τίς μεταβλητές.

Τά σύμβολα τῶν ἀριθμητικῶν πράξεων χρησιμοποιοῦνται σέ συνδυασμό μέ γράμματα καί ἀριθμούς γιά νά σχηματίσουν ἀλγεβρικές ἐκφράσεις. Τό ἴδιο καί στή BASIC, ὅπου μέ συνδυασμό μεταβλητῶν, σταθερῶν καί συμβόλων, σχηματίζονται ἀνάλογες ἐκφράσεις. Παράδειγμα τέτοιων κωδικοποιήσεων βλέπομε στόν Πίνακα 9.2.2 πού ἀκολουθεῖ:

ΠΙΝΑΚΑΣ 9.2.2.

Παραδείγματα κωδικοποιήσεως ἀλγεβρικῶν παραστάσεων

Ἀλγεβρική παράσταση	Ἐκφραση BASIC
$a + b$	$A + B$
$4x - 3$	$4 * X - 3$
$\frac{y}{2} + d - \frac{5}{6}$	$C / 2 + D - 5 / 6$
$ax^2 + bx + y$	$A * X ** 2 + B * X + C$
$2a + \frac{4}{b}$	$2 * A + 4 / B$

Θά πρέπει νά ἔχομε ὑπ' ὄψη μας ὅτι σέ τέτοιες σύνθετες ἐκφράσεις, ἡ ἐκτέλεση ἀπό τόν ὑπολογιστή τῶν ἐπί μέρους ἀριθμητικῶν πράξεων ἀκολουθεῖ κάποια ἱεραρχία. Ἔτσι, πρῶτα ἐκτελοῦνται οἱ ὑψώσεις σέ δύναμη, κατόπιν οἱ πολλαπλασιασμοί καί οἱ διαιρέσεις καί τελευταῖα οἱ προσθέσεις καί οἱ ἀφαιρέσεις.

Ὅταν στήν ἴδια παράσταση ὑπάρχουν δύο ἢ περισσότερες ἱεραρχικά ἰσοδύναμες πράξεις, π.χ. μιά πρόσθεση καί μιά ἀφαίρεση ἢ τρεῖς πολλαπλασιασμοί κλπ., τότε ἐκτελεῖται πρῶτα αὐτή πού προηγεῖται στήν παράσταση ἀπό τά ἀριστερά.

στ) Παρενθέσεις.

Στή BASIC χρησιμοποιοῦμε τίς παρενθέσεις, ὅπως καί στήν ἀλγεβρα, ὅταν θέλομε νά ξεχωρίσουμε τό ὅτι ὁρισμένες πράξεις ἐκτελοῦνται μόνο σέ μερικούς ἀπό τοὺς ὁρους μιᾶς παραστάσεως.

Θά πρέπει ὅμως νά προσέξομε περισσότερο στό γράψιμο ἀριθμητικῶν παραστάσεων καί στή χρήση παρενθέσεων, ἔχοντας ὑπ' ὄψη καί τήν ἱεραρχία πού ὑπάρχει στήν ἐκτέλεση τῶν ἀριθμητικῶν πράξεων, ὅπως εἶπαμε προηγουμένως.

Διηλαδῇ ἂν ὑποθέσουμε ὅτι μᾶς δίνεται γιά κωδικοποίηση ἡ ἐκφραση $\frac{a}{b + 5}$

Ἄν τή γράψομε $A/B + 5$ θά ἔχομε κάνει σφάλμα, γιατί, σύμφωνα μέ τά προηγούμενα, θά γίνει πρῶτα ἡ διαίρεση τοῦ A μέ τό B καί στό πηλίκο θά προστεθεῖ τό 5 , ἐνῶ ἐμεῖς θέλομε νά διαιρέσουμε τό A μέ τό $B + 5$. Γί' αὐτό κλείνομε ὅλο τό $B + 5$ μέσα σέ μιά παρένθεση καί ἔτσι ἡ σωστή κωδικοποίηση θά εἶναι:

$$A/(B + 5)$$

Μπορούμε νά βάζομε παρενθέσεις σέ μιά παράσταση χωρίς περιορισμό όσον άφορά τό πλήθος τους, άρκεί νά θυμόμαστε ότι θά πρέπει τελικά ή παράστασή μας νά κλείνει, δηλαδή νά έχομε τόσες άριστερές παρενθέσεις, όσες και δεξιές. Διαφορετικά ό ύπολογιστής θά δώσει μήνυμα λάθους γιά τήν παράσταση.

Μερικά παραδείγματα γιά τόν τρόπο χρήσεως τών παρενθέσεων βλέπομε στόν Πίνακα 9.2.3.

ΠΙΝΑΚΑΣ 9.2.3.

Παραδείγματα γιά τόν τρόπο χρήσεως τών παρενθέσεων στή BASIC

Άλγεβρική έκφραση	Έκφραση BASIC
$\left(\frac{a+b}{\gamma}\right)^2$	((A+B) / C) ** 2
$4(a+b) + \gamma\delta (x-3)$	4 * (A+B) + C * D * (X-3)
$2\left(3a - \frac{\beta}{\gamma+\delta}\right)$	2 * (3 * A - B / (C + D))

ζ) Μαθηματικές συναρτήσεις.

Η BASIC χρησιμοποιείται άρκετά συχνά γιά τή λύση άπλών μαθηματικών προβλημάτων. Στά προβλήματα αυτά προκύπτει πολλές φορές ή ανάγκη γιά τόν ύπολογισμό ενός τριγωνομετρικού άριθμού (ήμ, συν, κλπ.) ή μιός τετραγωνικής ρίζας ή ενός λογαρίθμου κ.ο.κ. Γιά νά μήν είμαστε ύποχρεωμένοι νά γράφομε κάθε φορά καί ένα ξεχωριστό πρόγραμμα γιά κάθε τέτοιο ύπολογισμό, οί κατασκευαστές έχουν ένσωματώσει στό μεταφραστικό πρόγραμμα τής BASIC άρκετά βοηθητικά ύποπρογράμματα, γιά τόν ύπολογισμό τών συναρτήσεων πού έμφανίζονται συχνότερα. Τά ύποπρογράμματα αυτά μπορούμε νά τά χρησιμοποιούμε μέσα σέ κάθε πρόγραμμά μας, καλώντας τα άπλως μέ βάση προσδιορισμένη όνομασία καί μέ παράμετρο τή μεταβλητή, γιά τήν τιμή τής όποίας ζητείται ό ύπολογισμός. Τά όνομάζομε *συναρτήσεις τοϋ συστήματος*.

Οί πιό γνωστές συναρτήσεις συστήματος τής BASIC μέ τίς κωδικές τους όνομασίες φαίνονται στόν Πίνακα 9.2.4.

Οί συναρτήσεις αυτές αναφέρονται σέ μιά κωδικοποιημένη έκφραση άκριβώς όπως καί οί άπλές μεταβλητές. Παραδείγματα βλέπομε στόν Πίνακα 9.2.5.

9.3 Κατηγορίες έντολών BASIC.

Τίς έντολές τής BASIC κατατάσσομε στίς έξής κατηγορίες:

- Έντολές άντικαταστάσεως.
- Έντολές είσόδου-έξόδου.
- Έντολές έλέγχου ή διακλαδώσεως.
- Έντολές δηλωτικές.

ΠΙΝΑΚΑΣ 9.2.4.**Οι συναρτήσεις συστήματος της BASIC**

Συνάρτηση	Λειτουργία
SIN(X)	Υπολογίζει τό ήμίτονο 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 τήν 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 τήν απόλυτη τιμή του X.
SQR(X)	Υπολογίζει τήν τετραγωνική ρίζα του X.
INT(X)	Υπολογίζει τό άκέραιο τμήμα του X.

ΠΙΝΑΚΑΣ 9.2.5.**Παραδείγματα κωδικοποιήσεως εκφράσεων πού περιέχουν συναρτήσεις**

Άλγεβρική έκφραση	Έκφραση BASIC
$3 \sin x + x + \frac{3}{4}$	$3 * \text{SIN}(X) + \text{ABS}(X) + C * D$
$e^x \sin x + \frac{3}{4}$	$\text{EXP}(X) * \text{COS}(X) + 3/4$
$2\sqrt{x} + a^2 + b^2$	$2 * \text{SQR}(X) + A ** 2 + B ** 2$
$\frac{1}{x} + 5 \log x$	$1/X + 5 * \text{LGT}(X)$

9.4 Τρόπος άναγραφής των έντολών BASIC.

Δέν υπάρχουν είδικά έντυπα κωδικογραφήσεως γιά τήν άναγραφή των έντολών BASIC.

Κάθε έντολή άρχίζει νά γράφεται από τήν πρώτη στήλη μιās γραμμής, πού χωρίζεται συνολικά σε 80 στήλες και πού άντιστοιχεί στό περιεχόμενο ενός δελτίου.

9.5 Έντολή άντικαταστάσεως. Η έντολή LET.

Η έντολή αύτή πού είναι από τίς πιό βασικές της BASIC έχει τήν εξής μορφή:

n LET μεταβλητή = έκφραση ή σταθερή

Τό n είναι ο αριθμός έντολης. 'Η υπόλοιπη έντολή αποτελείται από τή λέξη LET, πού ακολουθείται από μία μεταβλητή (μέ κάποιο κενό διάστημα μεταξύ τους), τό (=) σον(=) καί από μία κωδικοποιημένη έκφραση BASIC ή μία σταθερή.

Μέ τή βοήθεια τής έντολης LET γίνεται ό υπολογισμός τής αριθμητικής τιμής τής έκφράσεως του δευτέρου μέλους τής ισότητος (γι' αυτό θά πρέπει όλες οι μεταβλητές πού αναφέρονται στήν έκφραση νά έχουν γνωστό περιεχόμενο) καί στή συνέχεια καταχωρίζεται σάν περιεχόμενο στή μεταβλητή του πρώτου μέλους. Τό προηγούμενο περιεχόμενο, πού ένδεχομένως είχε, καταστρέφεται. Γι' αυτό τό λόγο καί ή έντολή αυτή λέγεται *έντολή αντικαταστάσεως*. Π.χ. στήν έντολή:

30 LET Z2 = SQR(X4) - 10

Θά γίνει ό υπολογισμός τής τιμής του δευτέρου μέλους, δηλαδή του SQR(X4) - 10 καί τό αποτέλεσμα θά καταχωρισθεί σάν περιεχόμενο τής Z2, τής οποίας τό προηγούμενο περιεχόμενο — αν είχε — χάνεται. Φυσικά γιά νά υπολογισθεί ή έκφραση SQR(X4) - 10, θά πρέπει νά είναι γνωστή ή τιμή (ή καλύτερα τό περιεχόμενο) τής μεταβλητής X4, τή στιγμή πού εκτελείται ή παραπάνω έντολή.

'Ας δούμε ένα άλλο παράδειγμα:

20 LET A = 4*A - 17

'Εδώ βλέπομε ότι ή μεταβλητή A εμφανίζεται καί στό πρώτο καί στό δεύτερο μέλος. Πώς θά εκτελεσθεί. 'Εφ' όσον θά γίνει, όπως είπαμε, πρώτα ό υπολογισμός του δευτέρου μέλους, ή μεταβλητή A θά μετάσχει στήν έκφραση μέ τό παλιό της περιεχόμενο, δηλαδή αυτό πού είχε τή στιγμή πού αρχίζει νά εκτελείται ή έντολή. Τό αποτέλεσμα του υπολογισμού θά καταχωρισθεί στή συνέχεια πάλι στήν A, καταστρέφοντας τήν προηγούμενη τιμή της.

Παρόμοιες μορφές έντολών LET χρησιμοποιούμε γιά νά αυξάνομε τό περιεχόμενο μιās μεταβλητής, πού τή χρησιμοποιούμε σάν κοινό άθροιστή, πολλαπλασιαστή ή σάν άπαριθμητή (Counter) σ' ένα πρόγραμμα π.χ.:

25	LET	S = S + X**2	περίπτωση άθροιστή
30	LET	J = J*K	περίπτωση πολλαπλασιαστή
42	LET	N = N + 1	περίπτωση άπαριθμητή

Σέ νεώτερους τύπους τής BASIC γίνονται δεκτές καί μορφές πολλαπλής έντολης αντικαταστάσεως, όπως π.χ.:

40 LET A = B = C = D = 0

πού στους άρχικούς τύπους θά έπρεπε νά αναλυθεί στίς έξις άπλές έντολές LET:

40	LET	A = 0
41	LET	B = 0
42	LET	C = 0
43	LET	D = 0

9.6 Έντολές εισόδου-εξόδου.

α) Έντολές READ και DATA.

Μέ τις έντολές αυτές εισάγονται οι κατάλληλες αριθμητικές τιμές δεδομένων στις μεταβλητές που χρησιμοποιούνται στο BASIC πρόγραμμα.

Η δομή των δύο εντολών είναι:

n READ μεταβλητή, μεταβλητή, μεταβλητή, κλπ.

n DATA σταθερή, σταθερή, σταθερή, κλπ.

n είναι ο αριθμός έντολης. Όταν οι μεταβλητές στη READ ή οι σταθερές στη DATA είναι περισσότερες από μία, τότε τις ξεχωρίζουμε, γράφοντας κόμματα μεταξύ τους. Καμιά από τις έντολές αυτές δεν εμφανίζεται μόνη της σέ ένα πρόγραμμα. Η έντολή DATA περιλαμβάνει τις αξίες των δεδομένων, που θα δοθούν στις μεταβλητές, οι οποίες αναφέρονται στην έντολή READ.

Κανονικά, όσες είναι οι μεταβλητές που εμφανίζονται στις διάφορες έντολές READ, τόσες θα πρέπει να είναι και οι σταθερές που εμφανίζονται στις διάφορες έντολές DATA του ίδιου προγράμματος. Η αντιστοίχιση είναι μία προς μία με τη σειρά εμφάνισης, π.χ.:

```
10 READ A, B1, C
20 READ D, F8
30 DATA 40, 1.2, 5.6, 7.8
40 DATA -1.8
```

Έδω στην πρώτη μεταβλητή, την A, θα δοθεί η τιμή της πρώτης σταθερής, ή 40. Ακολουθώς στη B1 θα δοθεί η τιμή 1.2, στη C ή 5.6 κ.ο.κ.

Όπως δηλαδή μπορούμε εύκολα να διαπιστώσουμε, θα ήταν δυνατό να είχαμε ενσωματώσει τη δεύτερη έντολή DATA στην πρώτη και να είχαμε τό ίδιο αποτέλεσμα:

```
10 READ A, B1, C
20 READ D, F8
30 DATA 40, 1.2, 5.6, 7.8, -1.8
```

Σύμφωνα με τό σχεδιασμό της BASIC, πριν από την εκτέλεση του προγράμματος, όλες οι σταθερές των εντολών DATA αποτελούν μία ομάδα δεδομένων (Data Block), που εμφανίζονται με την ίδια τάξη, όπως και στο πρόγραμμα. Κάθε φορά που εκτελείται μία έντολή READ, γίνεται ή σωστή αντιστοίχιση μιάς σταθερής, από αυτές που βρίσκονται στην περιοχή DATA, με την αντίστοιχη σέ τάξη μεταβλητή της έντολης READ. Αν στην ομάδα δεδομένων εμφανίζονται περισσότερες σταθερές από όσες είναι οι μεταβλητές, τότε όσες περισσεύουν απλώς δεν χρησιμοποιούνται. Αν όμως στην ομάδα δεδομένων εμφανίζονται λιγότερες, τότε τό λάθος είναι σοβαρό, γιατί δεν μπορεί να γίνει αντιστοίχιση σέ όλες τις μεταβλητές των εντολών READ και ή εκτέλεση του προγράμματος σταματά, ενώ συγχρόνως ό υπολογιστής δίνει σχετικό μήνυμα λάθους («OUT OF DATA») π.χ.:

```

10 READ D1, Q4, R
20 READ A, B, C
30 DATA 8.2, 4.5, 6.7, -7.1, 12

```

Τό λάθος έδω είναι σοβαρό, γιατί δέν υπάρχει σταθερή γιά νά αντιστοιχηθεῖ στήν τελευταία μεταβλητή, δηλαδή τήν C.

β) Έντολή INPUT.

Ἡ έντολή αὐτή χρησιμοποιεῖται ὅταν εἰσάγονται τά δεδομένα τοῦ προβλήματος μέ τή βοήθεια τῆς περιφερειακῆς μονάδας ἀναγνώσεως δελτίων (CARD READER). Ἡ δομή τῆς έντολῆς εἶναι:

n INPUT μεταβλητή, μεταβλητή,...

Ἀντίθετα μέ τήν έντολή READ πού πρέπει νά συνοδεύεται καί ἀπό έντολή DATA, ἡ INPUT ἐμφανίζεται μόνη της. Τά δεδομένα στήν περίπτωση αὐτή, εἰσάγονται στόν ὑπολογιστή μέ δελτίο, ἀπό τή μονάδα ἀναγνώσεως δελτίων. Π.χ.:

10 INPUT A,B,C

Τά δεδομένα διατυπώνονται ἐπάνω στό δελτίο μέ συνεχῆ τρόπο, χωριζόμενα μέ κόμματα.

γ) Έντολή PRINT.

Ἡ έντολή PRINT χρησιμοποιεῖται γιά νά τυπωθοῦν: α) Τιμές, περιεχόμενα μεταβλητῶν. β) Ἀποτελέσματα ἀριθμητικῶν ὑπολογισμῶν. γ) Διάφορα μηνύματα καί δ) συνδυασμός τῶν α,β,γ περιπτώσεων.

Στήν περίπτωση (α) ἡ δομή της εἶναι:

n PRINT μεταβλητή, μεταβλητή,...

Π.χ.

60 PRINT A, B, F8, C

Στήν περίπτωση (β) ἡ δομή της εἶναι:

n PRINT ἔκφραση BASIC, ἔκφραση BASIC,...

Π.χ.

40 PRINT (— B + R) / (Z*A), (—B—R) / (Z*A)

Ὑποτίθεται, βέβαια, ὅτι οἱ μεταβλητές πού ἀναφέρονται στίς ἔκφράσεις ἔχουν γνωστό περιεχόμενο.

Στήν περίπτωση (γ) τά μηνύματα, πού θέλομε νά τυπώσουμε, τά περικλείομε μέ-σα σέ ἀπόστροφους. Πρέπει νά εἶναι γραμμένα μέ κεφαλαίους λατινικούς χαρακτῆρες. Ἡ δομή τῆς έντολῆς εἶναι:

n PRINT μήνυμα, μήνυμα,...

Π.χ.

20 PRINT "THE SQUARE ROOT OF X IS"

9.7 Μερικές ακόμη εντολές.

α) Έντολή REM.

Επεξηγηματικά σχόλια, που γράφονται σέ διάφορα σημεία ενός προγράμματος, τό κάνουν πιό κατανοητό. Τέτοια σχόλια αναφέρονται στην ταυτότητα του προγράμματος, στην περιληπτική περιγραφή της μεθόδου που χρησιμοποιείται γιά τήν επίλυση του προβλήματος, στό ρόλο που διαδραματίζουν όρισμένες εντολές κλπ.

Τά σχόλια (Comments) γράφονται μέ τή βοήθεια της εντολής REM, που έχει τή μορφή:

n REM σχόλια

Η εντολή αυτή δέν εκτελείται. Ο ρόλος της είναι σαφώς βοηθητικός. Έπομένως ή εμφάνισή της δέν επηρεάζει καθόλου τήν εκτέλεση ενός προγράμματος. Μπορούμε νά εμφανίζουμε όσες εντολές REM θέλομε καί σέ όποιοδήποτε σημείο του προγράμματος, π.χ.:

```
10 REM PROGRAM FOR THE
20 REM SOLUTION OF A QUADRATIC
30 REM EQUATION
```

Τά σχόλια γράφονται μέ κεφαλαίους λατινικούς χαρακτήρες.

β) Έντολή END.

Η END είναι ή τελευταία εντολή κάθε προγράμματος BASIC καί γι' αυτό πρέπει νά έχει καί τό μεγαλύτερο αριθμό εντολής (συνήθως της δίνουμε τόν αριθμό 99999).

9.8 Μερικά απλά προγράμματα BASIC.

Σύμφωνα μέ τά όσα είπαμε μέχρι εδώ, θά προσπαθήσουμε νά γράψομε μερικά απλά προγράμματα στη γλώσσα BASIC.

Παράδειγμα 1.

Ας υποθέσουμε ότι θέλομε νά βρούμε τό μέσο όρο τεσσάρων αριθμών A, B, C, D. Τό πρόγραμμα θά έχει τή μορφή:

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΜΕΣΟΥ ΟΡΟΥ
20 READ A,B,C,D
30 LET M = (A + B + C + D)/4
40 PRINT "Ο ΜΕΣΟΣ ΟΡΟΣ ΕΙΝΑΙ", M
50 DATA 1.2, 4.5, 7.8, 9.4
99999 END
```

Παράδειγμα 2.

Έστω τώρα ότι θέλομε νά υπολογίσομε τήν τιμή της συνάρτησεως $\psi = x^3 - 5x^2 + 2\sqrt{x}$ γιά $x = 1.8$. Τό πρόγραμμα θά είναι:

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΤΗΣ ΤΙΜΗΣ ΣΥΝΑΡΤΗΣΕΩΣ
20 READ X
30 LET Y = X**3 - 5*X**2 + 2*SQR(X)
40 PRINT "Η ΤΙΜΗ ΤΗΣ ΣΥΝΑΡΤΗΣΕΩΣ ΕΙΝΑΙ", Y
50 DATA 1.8
9999 END

```

9.9 Έγτολές έλέγχου καί διακλαδώσεως.

Μέ τις έγτολές αυτές πετυχαίνομε:

- α) Άλλα χωρίς συνθήκη σέ κάποιο άλλο σημείο του προγράμματος.
- β) Διακλάδωση μέ συνθήκη.

Μπορεί έτσι ο υπολογιστής νά παίρνει λογικές αποφάσεις βασιζόμενος σέ προηγούμενους υπολογισμούς καί νά έκτελεϊ κατά περίπτωση διάφορες ομάδες άπό έγτολές, ή νά έκτελεϊ γιά όρισμένες φορές, επαναληπτικά, τήν ίδια ομάδα έγτολών.

α) Έγτολή GOTO.

Σέ μερικά προγράμματα χρειάζεται νά έκτελεσθοῦν οί έγτολές μέ διαφορετική σειρά, άπό εκείνη πού καθορίζουν οί αριθμοί έγτολών. Ένας τρόπος γιά νά αλλάξει ή κανονική ροή έκτελέσεως του προγράμματος, είναι νά χρησιμοποιηθεϊ ή έγτολή GOTO πού έχει τή μορφή:

n GOTO αριθμός έγτολής

όπου: σάν αριθμό έγτολής αναφέρομε τόν αριθμό τής έγτολής πού πρέπει νά έκτελεσθεϊ στή συνέχεια, π.χ.:

30 GOTO 40

σημαίνει ότι σάν έπόμενη έγτολή θά έκτελεσθεϊ ή έγτολή μέ αριθμό 40. Έ ή έγτολή αυτή μπορεί νά βρίζεται σέ κάποιο χαμηλότερο σημείο του προγράμματος, όποτε όλες οί άλλες έγτολές πού παρέμβάλλονται, άγνοοῦνται.

Παράδειγμα.

Τό παρακάτω πρόγραμμα έπιλύει χωρίς διερεύνηση μιά έξίσωση δευτέρου βαθμού μέ ένα άγνωστο. Έ ή έπίλυση επαναλαμβάνεται δύο φορές γιά δύο έξισώσεις.

```

10 PRINT "Α", "Β", "C", "R1", "R2"
20 READ A,B,C
30 LET R = SQR(B + 2 - 4*A*C)
40 LET R1 = (-B+R) / (2*A)
50 LET R2 = (-B-R) / (2*A)
60 PRINT A, B, C, R1, R2
70 GOTO 20
80 DATA 1, 3, 2, 1, 5, 4
99999 END

```


Μέ τό πρόγραμμα αυτό διαβάζεται μία σειρά δεδομένων (οί συντελεστές του ά- γνώστου), ύπολογίζονται οί ρίζες R1 καί R2 καί τυπώνονται. Μέ τήν έντολή GOTO μεταφέρεται ό έλεγχος πάλι στήν έντολή 20, μέ τήν όποία διαβάζεται άλλη μία τριάδα δεδομένων καί έπαναλαμβάνεται ή ίδια διαδικασία, μέχρις ότου έξαντλη- θούν τά δεδομένα πού αναφέρονται στήν έντολή DATA.

β) Έντολή IF-THEN.

Μέ τήν έντολή GOTO, όπως είπαμε, μπορούμε νά πετύχομε άλλα σέ ένα σημείο του προγράμματος χωρίς συνθήκη. Σέ πολλές όμως περιπτώσεις θέλομε αυτή ή διακλάδωση νά γίνεται μόνο όταν ίσχύει μιά συνθήκη, ενώ διαφορετικά νά συνεχί- ζεται ή κανονική ροή. Τότε χρησιμοποιούμε τήν έντολή IF-THEN πού έχει μορφή:

n IF λογική έκφραση συνθήκης THEN άριθμός έντολής

Πρίν όμως τήν περιγραφή της, άς μιλήσουμε λίγο γιά τίς λογικές συνθήκες.

Οί λογικές συνθήκες είναι έντελώς ανάλογες μέ τίς μαθηματικές συνθήκες ή πε- ριορισμούς πού θέτομε σέ ένα πρόβλημα. Συχνά λέμε π.χ. ότι θά πρέπει $x < a$ ή $x < b \leq \psi$ ή $0 \leq x < 3$ ή $\delta = 0$ κ.ο.κ. Τά ίδια αυτά σύμβολα (λογικά σύμβολα) $<$, $>$, $=$ κλπ., τά χρησιμοποιούμε γιά νά κωδικοποιήσουμε στή BASIC τέτοιες λογικές σχέ- σεις (Πίνακας 9.9.1). "Αν ή λογική συνθήκη πού αναφέρεται μετά τό IF ίσχύει (ικα- νοποιείται), τότε τό πρόγραμμα συνεχίζεται μέ τήν έκτέλεση της έντολής, της ό- ποίας ό άριθμός αναφέρεται μετά τή λέξη THEN. Διαφορετικά συνεχίζεται κανονι- κά μέ τήν έκτέλεση της επόμενης έντολής.

ΠΙΝΑΚΑΣ 9.9.1.

Τά σύμβολα των λογικών σχέσεων στή BASIC

Σύμβολο	Έπεξήγηση	Παράδειγμα	Άνάλυση
=	Ίσο πρός	IF X=Y THEN 30	"Αν τό X είναι Ίσο πρός τό Y, τότε ή έκτέ- λεση του προγράμματος θά συνεχισθεί από τήν έντολή 30, διαφορετικά από τήν άμέσως επόμενη.
< >	διάφορο του, όχι Ίσο μέ	IF X < > THEN 40	"Αν τό X είναι διάφορο του Y (άνισο), τότε θά έκτελεσθεί ή έντολή 40, διαφορετικά ή επόμενη.
<	μικρότερο από	IF X < Y THEN 50	"Η έκτέλεση θά συνεχισθεί από τήν έντολή 50, μόνο αν $X < Y$.
< = ή = <	μικρότερο ή Ίσο	IF X ≤ Y THEN 45	"Η έντολή 45 θά έκτελεσθεί άμέσως μετά τήν έντολή IF, μόνο όταν $X < Y$
>	μεγαλύτερο από	IF X > Y THEN 20	"Η έντολή 20 θά έκτελεσθεί άμέσως μετά τήν έντολή IF, μόνο όταν $X > Y$
> = ή = >	μεγαλύτερο ή Ίσο	IF X ≥ Y THEN 60	"Η έντολή 60 θά έκτελεσθεί άμέσως μετά τήν έντολή IF μόνο όταν $X > Y$

Τις λογικές συνθήκες τις διακρίνομε σέ απλές καί σύνθετες. 'Απλές εἶναι οἱ συνθήκες πού περιλαμβάνουν μία λογική σχέση ανάμεσα σέ δύο μόνο ὅρους, π.χ. $a < b$, $x > y$, $z \geq 0$ κλπ. Σύνθετες εἶναι οἱ συνθήκες πού περιλαμβάνουν σχέσεις μεταξύ περισσοτέρων ὄρων, π.χ. $a < b < c$, $-2 \leq z < 4$, $0 \leq x \leq 6$ κλπ. Τις σύνθετες συνθήκες στή BASIC τις ἀναλύομε σέ ἀνάλογο ἀριθμό ἀπλῶν συνθηκῶν, π.χ. τήν $a < b < c$ στίς ἀπλές $a < b$ καί $b < c$.

Εἶναι δυνατή ἐπίσης ἡ σύγκριση ὄχι μόνο ἀνάμεσα σέ δύο ἀπλοῦς ὅρους-μεταβλητές, ἀλλά ἀκόμη καί σέ δύο ἀπλές μαθηματικές ἐκφράσεις π.χ.:

IF B**2 < 4 *A*C THEN 100

Παράδειγμα 1.

Σάν πρῶτο παράδειγμα, ὅπου χρησιμοποιοῦμε τήν ἐντολή IF, δίνεται ἡ ἐπίλυση μιᾶς δευτεροβάθμιας ἐξίσωσης μέ ἓνα ἄγνωστο, πού περιλαμβάνει καί τή σχετική διερεύνηση γιά τό εἶδος τῶν ριζῶν καί ἀποτελεῖ βελτίωση τοῦ προγράμματος τῆς παραγράφου 8.9.1.

```

10 READ A,B,C
20 LET R = B↑2 - 4*A*C
30 IF R < 0 THEN 100
40 REM ROOTS ARE REAL
50 LET R = SQR(R)
60 PRINT "ROOTS ARE REAL"
70 PRINT "R1 =", (-B + R) / (2*A), "R2 =", (-B - R) / (2*A)
80 PRINT
90 GOTO 10
100 REM ROOTS ARE COMPLEX
110 LET R = SQR(-R)
120 PRINT "ROOTS ARE COMPLEX"
130 PRINT "REAL PART =", -B/(2*A)
140 PRINT "IMAG PART =", R/(2*A)
150 PRINT
160 GOTO 10
170 DATA 1, 3, 2, 1, 3, 4
99999 END

```

Παράδειγμα 2.

Ἐστω ὅτι δίνονται ἀριθμοί διάφοροι τοῦ μηδενός καί ζητεῖται νά γραφεῖ πρόγραμμα BASIC πού νά βρίσκει τό μεγαλύτερο. Ἐπειδὴ ὅλοι οἱ ἀριθμοί εἶναι διάφοροι τοῦ μηδενός, μποροῦμε νά ρυθμίσουμε τό τέλος τοῦ προγράμματος προσθέτοντας στό τέλος ὧν τῶν κανονικῶν δεδομένων τόν πλᾶσματικό δεδομένο-ἀριθμό 0 καί νά ἐλέγχομε κάθε φορά πού διαβάζεται ἓνα δεδομένο, ἂν ἔχει τιμὴ 0. Ἄν ναί, τότε θά σημαίνει ὅτι εἶναι τό πλᾶσματικό δεδομένο, ὁπότε θά πρέπει νά τελειώσει

τό πρόγραμμα, αν όχι, θά σημαίνει ότι είναι κανονικό δεδομένο και θά ακολουθη-
σει ή προβλεπόμενη κανονική επεξεργασία. Τό πρόγραμμα θά έχει, λοιπόν, τή μορ-
φή:

```

10  READ  B
20  READ  X
30  IF   X = 0 THEN 70
40  IF   X < B THEN 20
50  LET   B = X
60  GOTO  20
70  PRINT "LARGEST ELEMENT IS;" B
80  DATA 1.7, 2.5, -1, 2.3, 1.4, 7.1, 9.0
99999 END

```

Β ονομάζομε μία περιοχή τής μνήμης, όπου κρατείται ό μεγαλύτερος από τούς
αριθμούς πού συγκρίνομε κάθε φορά.

Παράδειγμα 3.

Στό παράδειγμα αυτό θά δούμε τή χρήση τής έντολης IF-THEN στόν έλεγχο του
περιεχομένου ενός άπαιριθμητου (Counter).

"Εστω ότi ζητείται ό ύπολογισμός τής τιμής τής παραστάσεως:

$$Σx^v = x^1 + x^2 + ... + x^{30} \text{ γιά τή τιμή } x = 1.3$$

Στήν περίπτωση αυτή πρέπει νά προστεθούν 30 όροι τής μορφής x^v . Τό
πρόγραμμα θά έχει τήν παρακάτω μορφή:

```

10  LET S = 0
20  LET N = 0
30  READ X
40  LET N = N + 1
50  LET S = S + X **N
60  IF N < 30 THEN 40
70  PRINT S
80  DATA 1.3
99999 END

```

Χρησιμοποιήσαμε τόν άπαιριθμητή (Counter) N γιά νά μετρά τις φορές πού έπα-
ναλαμβάνεται ή διαδικασία τής προσθέσεως. Μετά από κάθε επανάληψη, τό πε-
ριεχόμενό του αύξάνεται κατά 1. Παράλληλα ό N δανείζει τό περιεχόμενό του γιά
τή διαμόρφωση του όρου x^v κάθε φορά. Ή μεταβλητή S παριστάνει ένα
άθροιστή, στόν όποίο δίνομε, όπως και στόν N, άρχική τιμή 0. Ή έντολή 50 IF N <
30 THEN 40 έλέγχει τό περιεχόμενο του N. "Αν αυτό είναι μικρότερο του 30, τότε
γίνεται άλμα στήν έντολή 40, από όπου συνεχίζεται ή έκτέλεση του προγράμμα-
τος, διαφορετικά συνεχίζεται μέ τήν άμέσως επόμενη έντολή 70 PRINT S.

γ) Έντολή ON-GOTO.

Είναι μία λιγότερο εύχρηστη έντολή έλέγχου. Η μορφή της είναι:

n ON έκφραση BASIC GOTO άριθ. έντολης, άριθ. έντολης,...
(μέχρι 4)

Η έντολή λειτουργεί ως έξης: Υπολογίζεται πρώτα ή τιμή της έκφράσεως BASIC και κρατείται τό άκέραιο μέρος της. "Αν ίσοῦται μέ 1, τότε θά έκτελεσθεϊ στή συνέχεια ή έντολή μέ άριθμό τόν πρώτο πού έμφανίζεται μετά τή λέξη GOTO. "Αν ίσοῦται μέ 2, τότε θά έκτελεσθεϊ ή έντολή μέ άριθμό τό δεύτερο μετά τή GOTO κ.ο.κ. Φυσικά θά πρέπει ή τιμή του άκεραίου μέρους της έκφράσεως νά είναι ≥ 1 καί ≤ 4 , π.χ.:

90 ON C GOTO 210, 330, 460, 570

πού σημαίνει ότι, αν τό άκέραιο μέρος του C είναι 1, θά έκτελεσθεϊ στή συνέχεια ή έντολή μέ άριθμό 210. "Αν τό άκέραιο μέρος είναι 2, θά έκτελεσθεϊ ή έντολή 330, αν τό άκέραιο μέρος είναι 3, θά έκτελεσθεϊ ή 460 καί τέλος, αν τό άκέραιο μέρος είναι 4, θά έκτελεσθεϊ ή 570.

δ) Έντολή STOP.

Πολλές φορές, κατά τόν έλεγχο μιās συνθήκης, θέλομε μία από τίς διακλαδώσεις του προγράμματος νά οδηγεί στόν τερματισμό του π.χ.:

70 GOTO 99999

.....
99999 END

Αντί της πρώτης έντολης, θέτομε τήν
70 STOP

καί τό άποτέλεσμα είναι τό ίδιο.

9.10 Έντολές FOR-NEXT.

Στά προηγούμενα είδαμε πώς μέ τή βοήθεια των έντολών GOTO καί IF-THEN μπορούμε νά επαναλάβομε τήν έκτέλεση ενός τμήματος του προγράμματος για περισσότερες από μία φορές. Σέ παρόμοιες περιπτώσεις ή κωδικογράφηση διευκολύνεται πάρα πολύ από τό ζευγος έντολών FOR καί NEXT. Η FOR έχει τή γενική μορφή:

n FOR μεταβλητή = έκφραση TO έκφραση STEP έκφραση

καί συνδιάζεται μέ τήν έντολή NEXT πού έχει τή μορφή:

NEXT μεταβλητή

Οι δύο αυτές έντολές έμφανίζονται πάντα μαζί. Προηγείται ή FOR καί άκολου-

θεϊ ή NEXT. Μεταξύ τους παρεμβάλλεται ή ομάδα τών έντολών, τών όποίων θέλο-
με τή συνεχή έπαναληπτική έκτέλεση.

Ή μεταβλητή, πού έμφανίζεται μετά τή λέξη FOR, είναι ένας δείκτης (δείκτης
της FOR) πού παίρνει άρχική τιμή, τήν τιμή πού δείχνει ή έκφραση μετά τό = καί
τελευταία τήν τιμή της έκφράσεως πού έμφανίζεται μετά τό TO. Ή τιμή της έκφρά-
σεως μετά τή λέξη STEP είναι τό βήμα, δηλαδή τό ποσό, μέ τό όποιο αύξάνει κάθε
φορά ή τιμή του δείκτη.

Στήν έντολή NEXT, ή μεταβλητή είναι ή ίδια πού αναφέραμε καί στή FOR.

Ή διαδικασία έπαναλήψεως ενός τμήματος του προγράμματος καλεϊται Loop.

Ής δοϋμε όμως ένα παράδειγμα.

Έστω ότι θέλομε νά βροϋμε τό άθροισμα τών 100 πρώτων φυσικών αριθμών.

Τό πρόγραμμα πού θά γράψομε μέ τή βοήθεια τών έντολών FOR-NEXT θά εί-
ναι τό έξης:

```
10 LET SUM = 0
20 FOR I = 1 TO 100 STEP 1
30 LET SUM = SUM + I
40 NEXT I
50 PRINT SUM
99999 END
```

Σύμφωνα μέ τό πρόγραμμα αυτό, ή έντολή 30 θά εκτελεσθεϊ συνολικά 100 φο-
ρές. Τό πλήθος αυτό τών άνακυκλώσεων καθορίζεται από τίς έντολές 20 καί 40.
Ό δείκτης I παίζει διπλό ρόλο. Άφ' ενός μέν χρησιμοποιεϊται σάν άπαριθμητής του
πλήθους τών έπαναλήψεων (Loops), άφ' έτέρου δέ δανείζει τίς τιμές του στήν κύ-
ρια σχέση ύπολογισμού:

```
30 LET SUM = SUM + I
```

Ή άρχική τιμή πού παίρνει ό I είναι 1 καί ή τελική 100, ένw τό βήμα είναι 1. Μέ
τήν έντολή 40 σημειώνεται τό τέλος του έπαναλαμβανομένου τμήματος του προ-
γράμματος.

Ής εξετάσομε πió αναλυτικά τόν τρόπο εκτέλεσεως αυτού του προγράμματος.

Στήν άρχή εκτελεϊται ή έντολή 10 καί κατόπιν (έντολή 20) ή έπανάληψη του
προγράμματος (έντολές 20-40) Τήν πρώτη φορά έχομε I = 1 καί θά εκτελεσθεϊ
γιά πρώτη φορά ή έντολή

```
30 LET SUM = SUM + I
```

Μέ τήν έντολή 40 έπιστρέφει ό έλεγχος στήν 20, τό I γίνεται 2 (δηλ. προηγού-
μενη τιμή + βήμα) καί πραγματοποιεϊται γιά δεϋτερη φορά ή εκτέλεση της έντολής
30. Μέ τόν έρχομό στή NEXT, ξαναεπιστρέφει ό έλεγχος στήν 20, τό I γίνεται 3,
εκτελεϊται γιά τρίτη φορά ή έντολή 30 κ.ο.κ. Όταν ή τιμή του I γίνει 100, τότε ή
έντολή 30 θά εκτελεσθεϊ γιά έκατοστή φορά καί ό έλεγχος θά όδηγήσει έξw από
τήν έπανάληψη, όπου τυπώνεται τό SUM (έντολή 50) καί τό πρόγραμμα τελειώνει.

“Ωστε με τις εντολές FOR-NEXT μπορούμε σε αρκετές περιπτώσεις να ελέγξουμε και το τέλος ενός προγράμματος.

Δίνονται μερικά ακόμη παραδείγματα:

Παράδειγμα 1.

Νά γραφεί πρόγραμμα που να υπολογίζει την τιμή της παραστάσεως:

$$\prod_{v=1}^6 (x^v + a) \quad \text{γιά } x = 1.8 \text{ καί } a = 3.4.$$

```

10 LET P = 1
20 READ X, A
30 FOR N = 1 TO 6 STEP 1
40 LET P = P*(X**N + A)
50 NEXT N
60 DATA 1.8, 3.4
99999 END

```

Παράδειγμα 2.

Νά υπολογισθεί η τιμή της συναρτήσεως $\psi = x^3 - 5x^2 + 2$ για τιμές του $x = 1(1)5$.

```

10 FOR X = 1 TO 5 STEP 1
20 LET Y = X**3 - 5*X**2 + 2
30 PRINT Y
40 NEXT X
99999 END

```

9.11 Πίνακες (Arrays). Μεταβλητές με δείκτες.

Πολύ συχνά στην άλγεβρα χρησιμοποιούμε γράμματα με ένα ή δύο μικρούς αριθμητικούς δείκτες στο κάτω δεξιό μέρος του γράμματος. Π.χ. συνηθίζουμε να γράφουμε τη γενική μορφή ενός πολυωνύμου ως εξής:

$$a_0 x^0 + a_1 x^1 + \dots + a_v x^v$$

όπου: τούς συντελεστές του τούς συμβολίζουμε με το ίδιο γράμμα, τό a , αλλά με ένα δείκτη διαφορετικό ώστε να διακρίνονται μεταξύ τους. Η γραφή αυτή χρησιμοποιήθηκε αρχικά, όταν ήταν τόσες πολλές οι παράμετροι ή οι συντελεστές, που δέν έφθαναν τά γράμματα του άλφαβήτου για να τούς παραστήσομε με αυτά. Καθιερώθηκε όμως καί γενικεύθηκε, γιατί είναι πιό άπλή, εύχρηστη καί διευκολύνει τό γενικό συμβολισμό μιås παραστάσεως.

Γράμματα με δείκτη χρησιμοποιούνται επίσης για την παράσταση όμοειδών δεδομένων στη στατιστική. Έτσι, όταν π.χ. μετρούμε τά ύψη 10 διαφορετικών ατόμων, άντί να τά συμβολίσομε με διαφορετικό γράμμα τό καθένα, $\alpha, \beta, \gamma, \dots$, χρησιμοποιούμε τό συμβολισμό με δείκτη, π.χ. $x_1, x_2, \dots, x_{10}$ όπου με x_1 παριστάνομε τό ύψος του πρώτου ατόμου, με x_2 του δευτέρου κ.ο.κ.

Παρόμοιος συμβολισμός χρησιμοποιείται και στη BASIC, όπου οι μεταβλητές με δείκτη περιέχουν, εκτός από την ονομασία τους, και μία παρένθεση, μέσα στην οποία εμφανίζονται ένας ή δύο αριθμητικοί δείκτες (μέ τιμή φυσικούς αριθμούς), χωριζόμενοι με κόμμα. Π.χ.:

A(12), B(4), X(1,2), Z(76)

Αντί για αριθμητικές σταθερές, σάν δείκτης μπορεί να εμφανίζεται και μία μεταβλητή ή ακόμη και μία απλή έκφραση BASIC γνωστής ακεραίας τιμής π.χ.:

A(M), B(N + 4), X(J + 4*K)

Όλες οι μεταβλητές με την ίδια ονομασία, αλλά διαφορετική αριθμητική τιμή του δείκτη, αποτελούν ένα πίνακα (Array) π.χ. οι μεταβλητές A(1), A(2), A(3), A(4) αποτελούν τον πίνακα A. Τά A(1), A(2), A(3), A(4) ονομάζονται στοιχεία (Elements) του πίνακα A. Δηλαδή, στην περίπτωση αυτή, ο πίνακας A αποτελείται από 4 στοιχεία.

Όταν τα στοιχεία ενός πίνακα έχουν μόνο ένα δείκτη, τότε λέμε ότι ο πίνακας είναι μονοδιάστατος. Π.χ. τα στοιχεία:

A(4), B(M), E(2*I), Z(N + K)

άνηκουν σε μονοδιάστατους πίνακες, τους A, B, E, Z αντίστοιχα.

Όταν τα στοιχεία ενός πίνακα αποτελούνται από δύο δείκτες, τότε λέμε ότι ανήκουν σε δισδιάστατο πίνακα (two-dimensional array). Π.χ. τα στοιχεία:

A(4,3), R(J,M), X (2*I,N)

άνηκουν στους δισδιάστατους πίνακες A, R και X. Στην περίπτωση των δισδιάστατων πινάκων, μπορούμε να φαντασθούμε τα στοιχεία τους διατεταγμένα επάνω στο επίπεδο σε γραμμές και στήλες. Ο πρώτος αριθμητικός δείκτης συμβολίζει τότε τον αυξαντα αριθμό γραμμής και ο δεύτερος τον αυξαντα αριθμό στήλης, στη δισδιαύρωση των οποίων βρίσκεται το συγκεκριμένο στοιχείο, π.χ.:

	1η στήλη	2η στήλη	3η στήλη	4η στήλη
1η γραμμή	A(1,1)	A(1,2)	A(1,3)	A(1,4)
2η γραμμή	A(2,1)	A(2,2)	A(2,3)	A(2,4)
3η γραμμή	A(3,1)	A(3,2)	A(3,3)	A(3,4)

Δηλαδή το στοιχείο A(2,3) βρίσκεται στη δισδιαύρωση της δεύτερης γραμμής (πρώτος δείκτης = 2) και της τρίτης στήλης (δεύτερος δείκτης = 3).

Η επεξεργασία όμοιων δεδομένων διευκολύνεται πάρα πολύ με την εξομώσή τους προς τα στοιχεία ενός πίνακα και με χρήση των εντολών FOR-NEXT. Μερικά τέτοια παραδείγματα θα δούμε στη συνέχεια:

9.12 Έντολη DIM.

Στις περιπτώσεις που σε ένα πρόγραμμα χρησιμοποιούμε κάποιο πίνακα, είναι απαραίτητο να δηλώσουμε στην αρχή, με τη βοήθεια της εντολής DIM, τον αριθμό των στοιχείων από τα οποία αποτελείται. Η δομή της είναι:

n DIM μεταβλητή (αριθμητικός δείκτης), μεταβλητή (αριθμητικός δείκτης),... π.χ.

10 DIM A(10), G(20)

πού σημαίνει ότι μέσα στο πρόγραμμα χρησιμοποιείται ή μεταβλητή A για 10 στοιχεία (πού αποτελούν πίνακα), με δείκτη που παίρνει τιμές από 1-10 και ή μεταβλητή G για ένα άλλο πίνακα με 20 στοιχεία και με δείκτη που παίρνει τιμές από 1-20. Συγχρόνως δίνονται οδηγίες στον υπολογιστή να κρατήσει ανάλογο αριθμό θέσεων στη μνήμη, για να καταχωρήσει τὰ στοιχεία κάθε πίνακα.

9.13 Παραδείγματα.

Παράδειγμα 1.

Έστω ότι θέλουμε να προσθέσουμε 17 αριθμούς $a_1, a_2, \dots, a_{17}$ και να τυπώσουμε τό άθροισμά τους.
Τό αντίστοιχο πρόγραμμα θά είναι:

```

10 REM SUM OF NUMBERS
15 DIM A(17)
20 LET S = 0
30 FOR I = 1 TO 17
40 READ A(I)
50 LET S = S + A(I)
60 NEXT I
70 PRINT "THE SUM IS", S
80 DATA 5, -7, 12, 6, 8, 9, -6, 2, -1, 4, 6
90 DATA 10, 13, -14, 1, 0, 8
99999 END

```

Παράδειγμα 2.

Νά βρεθεί ο μέσος όρος τῶν αριθμῶν $a_1, a_2, a_3, a_4, a_5, a_6, a_7$. Τό πρόγραμμα θά έχει τήν ἑξῆς μορφή:

```

10 REM ARITHMETIC MEAN
15 DIM A(7)
20 LET S = 0
30 FOR J = 1 TO 7
40 READ A(J)
50 LET S = S + A(J)
60 NEXT J
70 LET M = S/7
80 PRINT "THE AVERAGE IS", M
90 DATA 22, 21, 24, 23, 25, 28, 20
99999 END

```

9.14 Έπεξεργασία ἀρχείων στή BASIC.

Ἡ BASIC χρησιμοποιεῖται ὄχι μόνο γιά τήν ἐπίλυση ἀπλῶν μαθηματικῶν προ-

βλημάτων, αλλά ακόμη και για την απλή επεξεργασία αρχείων. Μέ τον όρο επεξεργασία έννοούμε μία από τις εξής λειτουργίες.

- α) Δημιουργία ενός αρχείου με τη βοήθεια του υπολογιστή επάνω σε κάποιο φορέα, συνήθως μαγνητικό (ταινία ή δίσκο).
- β) Άνοιγμα ενός αρχείου.
- γ) Κλείσιμο ενός αρχείου.
- δ) Έγγραφή πληροφοριών σε υπάρχον αρχείο.
- ε) Ανάγνωση πληροφοριών από υπάρχον αρχείο.

Θά δούμε αναλυτικά τις διάφορες φάσεις επεξεργασίας αρχείων. Ένα αρχείο πρίν από τη χρησιμοποίησή του πρέπει νά «άνοιχθεϊ», δηλαδή νά προετοιμασθεϊ, ένεργοποιούμενο από τή μονάδα έλέγχου, πρίν άρχίσει ή μεταφορά τών πληροφοριών από καί πρός τή μνήμη.

Τή στιγμή αὐτή θά πρέπει νά καθορισθεϊ κατά πόσον από τό αρχείο αὐτό θά διαβάζονται πληροφορίες, δηλαδή θά εἶναι ἕνα αρχείο εισόδου (Input File) ή θά γράφονται σ' αὐτό πληροφορίες, δηλαδή θά εἶναι ἕνα αρχείο εξόδου (Output File).

Στήν περίπτωση πού ανοίγομε-ένεργοποιούμε ἕνα αρχείο, χρησιμοποιούμε τις έντολές:

n OPEN u,f INPUT
n OPEN u,f OUTPUT

Γιά κάθε μία από τις περιπτώσεις χρήσεως τοῦ αρχείου πού αναφέρθηκαν καί στά δύο είδη έντολῶν, τό u εἶναι μιά αριθμητική σταθερή ή μεταβλητή καί παριστάνει τή συμβολική όνομασία πού δίνομε στό αρχείο καί μέ τήν όποία θά τό καλοῦμε μέσα στό πρόγραμμά μας, ένῶ τό f εἶναι μιά ἄλλη σταθερή ή μεταβλητή, πού άποτελεῖ τήν πραγματική όνομασία τοῦ αρχείου. Π.χ.:

OPEN 5, A1 INPUT

Μέ τήν έντολή αὐτή ανοίγομε τό αρχείο, πού τό πραγματικό του όνομα εἶναι A1 καί πού θά χρησιμοποιηθεϊ στό πρόγραμμά μας μέ τή συμβολική όνομασία 5.

Οἱ έντολές άναγνώσεως καί έγγραφής ενός αρχείου εἶναι παρόμοιες μέ τις έντολές INPUT καί PRINT, πού είδαμε στis παραγράφους 8.6.2 καί 8.6.3, μέ τή διαφορά ὅτι αναφέρεται σ' αὐτές καί τό όνομα τοῦ αρχείου. Π.χ.

90 INPUT: CLS: A,B

Διαβάζονται από τό αρχείο CLS, δύο τιμές-δεδομένα γιά τό A καί B αντίστοιχα. Έπίσης:

100 PRINT: CS: A,B

Έγγράφονται στό αρχείο μέ όνομα CLS, οἱ τιμές τών μεταβλητῶν A καί B.

Σέ μερικά συστήματα υπολογιστῶν, αντί τῆς INPUT, χρησιμοποιεῖται ή GET καί αντί τῆς PRINT, ή PUT ὡς εξής:

90 GET S: A,B
180 PUT 5: A,B

9.15 Υποπρογράμματα (Subprograms).

Υποπρόγραμμα είναι ένα σύνολο από εντολές, που εκτελούν μία καθορισμένη εργασία. Με τη μορφή αυτή χρησιμοποιείται σαν βοηθητικό πρόγραμμα σε διάφορα σημεία ενός κυρίου προγράμματος (Main program). Τα υποπρογράμματα στη BASIC τα διακρίνουμε σε δύο κατηγορίες:

- α) Συναρτήσεις (Function Subprograms).
- β) Υπορουτίνες (Subroutines).

α) Συναρτήσεις (Functions).

Όπως είδαμε ήδη, στο μεταφραστικό πρόγραμμα της BASIC (Compiler) έχουν ενσωματωθεί διάφορες συναρτήσεις, όπως COS, SIN, ABS, SQR κλπ., για την εκτέλεση διαφόρων υπολογισμών. Η BASIC δίνει τη δυνατότητα στον προγραμματιστή να δημιουργήσει και αυτός μέσα στο πρόγραμμά του συναρτήσεις δικές του, που να υποβοηθούν και να συντομεύουν τη λειτουργία του προγράμματός του.

Ο καθορισμός μιάς συναρτήσεως γίνεται στην αρχή ενός προγράμματος με την εντολή DEF, που η δομή της είναι:

```
n DEF FN μεταβλητή (παράμετροι) = έκφραση BASIC
10 DEF FNS(X) = SIN (X/5+3)
```

όπου δημιουργήσαμε μία συνάρτηση με όνομα FNS, με την οποία υπολογίζεται η τιμή της έκφρασης $\sin(X/5+3)$. Η μεταβλητή X στην παρένθεση ονομάζεται παράμετρος (argument) και χρησιμοποιείται σαν ανεξάρτητη μεταβλητή μιάς άλλης βρικής συναρτήσεως.

Αφού όριστηκε η FNS, μπορεί να χρησιμοποιηθεί στο πρόγραμμα. Έτσι, αν συναντήσαμε π.χ. την εντολή

```
100 LET A = B*FNS (T/2),
```

θα υπολογισθεί η τιμή της καλούμενης συναρτήσεως FNS με αξία της παραμέτρου X την T/2.

Παράδειγμα.

Νά γραφεί συνάρτηση, που να υπολογίζει το έμβαδόν ενός ορθογωνίου τριγώνου από τις κάθετες πλευρές του, α και β.

```
10 DEF FNA (A,B) = 1/2*A*B
```

β) Υπορουτίνες (Subroutines).

Η υπορουτίνα είναι και αυτή ένα υποπρόγραμμα, αλλά η δομή και ο τρόπος χρήσεώς της σε ένα πρόγραμμα είναι τελείως διαφορετική από ό,τι μιάς συναρτήσεως.

Σε μία υπορουτίνα (που αποτελείται από περισσότερες από μία εντολές) μπορούμε να χρησιμοποιήσουμε τις περισσότερες εντολές στη BASIC, ενώ, αντίθετα, σε μία συνάρτηση, μόνο μία εντολή αντικαταστάσεως.

Η υπορουτίνα γράφεται στο κάτω μέρος του κυρίως προγράμματος και δεν μπορεί να χρησιμοποιηθεί σαν όρος μέσα σε μία έκφραση BASIC, όπως μία συ-

νάρτηση, αλλά καλείται με μία ειδική έντολή, την GOSUB, στο κατάλληλο σημείο του προγράμματος. Η έντολή κλήσεως υπορουτίνας έχει τη μορφή:

n GOSUB m

όπου m φυσικός αριθμός που αποτελεί τον αριθμό της πρώτης έντολής της υπορουτίνας. Π.χ.:

100 GOSUB 460

καί σημαίνει ότι θα μεταφερθεί από τό σημείο αυτό ο έλεγχος του προγράμματος στην έντολή 460, που είναι καί ή πρώτη έντολή της υπορουτίνας, που χρησιμοποιείται στο πρόγραμμα.

Η τελευταία έντολή μιάς υπορουτίνας είναι ή:

n RETURN

Παράδειγμα.

Σάν παράδειγμα, θά υπολογίσουμε τό πλήθος τών συνδυασμών τών N πραγμάτων ανά J, χρησιμοποιώντας τή σχέση:

$$\binom{N}{J} = \frac{N!}{J!(N-J)!} \quad \text{Δεχόμεστε ότι } 0! = 1$$

όπου $N! = 1 \times 2 \times \dots (N-1) \times N$ καί τό διαβάζουμε «N παραγοντικό»

$J! = 1 \times 2 \times \dots (J-1) \times J$ καί τό διαβάζουμε «J παραγοντικό»

$(N-J)! = 1 \times 2 \times \dots (N-J+1) \times (N-J)$ καί τό διαβάζουμε «N-J παραγοντικό»

Τό πρόγραμμα θά έχει τήν έξης μορφή:

```

10  READ  N,J
20  REM  CALCULATE  N  FACTORIAL
30  LET  K = N
40  GOSUB 500
50  LET  N1 = F
60  REM  CALCULATE  J  FACTORIAL
70  LET  R = J
80  GOSUB 500
90  LET  J1 = F
100 REM  CALCULATE  N-J  FACTORIAL
110 LET  K = N-J
120 GOSUB 500
130 LET  C = N1/(J1*F)
140 PRINT  N,J,C
150 STOP
500 REM  COMPUTATION OF  F = K  FACTORIAL
510 LET  F = 1
520 IF  K = 0 THEN 560
530 FOR  L = 1  TO  K
540 LET  F = F*L
550 NEXT  L
560 RETURN
1000 DATA 8.3
99999 END

```

Άσκησης.

1. Κωδικοποιήστε στη BASIC τις εξής μαθηματικές εκφράσεις:

α) $2\pi r^2$

β) $a+x[\beta+v(\gamma+\delta v)]$

γ) $\frac{a+\beta}{\gamma+\frac{\delta}{\epsilon}}$

δ) $x^3 + \frac{\beta}{\gamma+\delta}$

ε) $a\beta\gamma + \frac{2}{4}(x-\psi)$

ζ) $1 + \frac{1}{x}(a+\beta) - \gamma\delta$

η) $\frac{ax^2 + \beta x + \gamma}{\epsilon^3 - 4\delta}$

θ) $0.3x^3 + \frac{7}{8}\psi^2 - x\psi$

ι) $\frac{1}{6}(a+\beta+\gamma)(\gamma\delta+a\beta)$

κ) $(3+a)[(x^2+1)+2(\psi-3)]$

2. Επίσης τις εκφράσεις:

α) $\frac{1+\sin\psi}{1-\eta\mu\psi}$

β) $|x|e^x + \frac{a}{\log x}$

γ) $2-x^2 + \sqrt{e^x \log x}$

δ) $4\sqrt{x} - 5\eta\mu x + 6e^x$

ε) $x\eta\mu x + \sqrt{a^2 + \beta^2}$

ζ) $\frac{1+e^x}{1-e^x}$

$$\eta) \pi \eta \mu^2 x + \sigma \nu \nu^3 \psi$$

$$\theta) \alpha \log x + (\beta \log \psi)^2$$

3. Νά γράψετε πρόγραμμα BASIC πού νά υπολογίζει τήν περίμετρο, τή διαγώνιο, τό έμβαδό καί τήν άκτίνα τοῦ έγγεγραμμένου κύκλου, ένός τετραγώνου πλευράς α (έφαρμογή μέ $\alpha = 1.5$).
4. Νά γράψετε πρόγραμμα BASIC, γιά τόν υπολογισμό τοῦ έμβαδοῦ ένός τριγώνου, όταν εἶναι γνωστά τά μήκη τῶν πλευρῶν του α, β, γ . Χρησιμοποιεῖστε τόν τύπο:

$$E = \sqrt{\tau(\tau-\alpha)(\tau-\beta)(\tau-\gamma)} \quad \text{όπου}$$

$$T = \frac{\alpha + \beta + \gamma}{2} \quad \text{ή ήμiperίμετρος τοῦ τριγώνου.}$$

5. Νά γραφεῖ πρόγραμμα πού νά υπολογίζει τό μικρότερο άπό 3 άριθμούς α, β, γ .
6. Νά γραφεῖ πρόγραμμα πού νά υπολογίζει τήν τιμή τῆς παραστάσεως

$$\sum_{v=1}^{12} \frac{x^v}{v} \quad (\text{έφαρμογή γιά } x = 0.8)$$

7. Νά γραφεῖ υποπρόγραμμα, πού νά υπολογίζει τό έμβαδόν ένός τριγώνου άπό τίς πλευρές του καί ύστερα νά χρησιμοποιηθεῖ μέσα σέ ένα κυρίως πρόγραμμα, γιά τόν υπολογισμό τοῦ έμβαδοῦ τῆς έπιφάνειας ένός τετραέδρου, τοῦ όποίου δίνονται τά μήκη τῶν άκμῶν $\alpha, \beta, \gamma, \delta, \epsilon, \zeta$.
8. Νά γραφεῖ πρόγραμμα BASIC πού νά υπολογίζει τήν τιμή τῆς παραστάσεως.

$$\prod_{v=1}^{10} \frac{a^v}{v} \quad \text{γιά } a = 1.2$$

9. Νά γραφεῖ πρόγραμμα BASIC πού νά υπολογίζει τό γινόμενο τῶν 20 πρώτων φυσικῶν περιττῶν άριθμῶν.
10. Νά γραφεῖ συνάρτηση γιά τόν υπολογισμό τῆς $\psi = x^3 - 3x^2 + 5$

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ

Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ FORTRAN

10.1 Γενικά.

Η γλώσσα προγραμματισμού FORTRAN είναι μία από τις πιο γνωστές γλώσσες υψηλού επιπέδου, που χρησιμοποιείται σχεδόν αποκλειστικά για την επίλυση τεχνικών και επιστημονικών προβλημάτων, όπου κυριαρχεί ή χρήση των μαθηματικών. Παρουσιάζει αρκετές ομοιότητες με τη BASIC, που γνωρίσαμε στο προηγούμενο κεφάλαιο. Η όνομασία της προέρχεται από τα αρχικά των λέξεων: FORMulae TRANslator που σημαίνουν «μεταφραστής σχέσεων». Υπάρχουν διάφοροι τύποι της FORTRAN. Εμείς θα βασισθούμε στον τύπο FORTRAN IV, που είναι και ο πιο συνηθισμένος.

10.2 Γενικά χαρακτηριστικά της FORTRAN.

α) Σταθερές και μεταβλητές.

Στις διάφορες έντολές της FORTRAN χρησιμοποιούμε, όπως και στη BASIC, μεταβλητές (Variables) και σταθερές (Constants).

Σταθερή (Constant) είναι κάθε αριθμός που εκφράζεται στο δεκαδικό σύστημα. Όπως στην αριθμητική, έτσι και στη FORTRAN τις σταθερές τις διακρίνουμε σε άκεραιες (Integer) και κλασματικές ή πραγματικές (Real). Άκεραιη σταθερή (Integer constant) είναι κάθε άκεριος δεκαδικός αριθμός (άρνητικός ή θετικός), π.χ. οι αριθμοί:

-17 23 0 456 +542 -289

Πραγματική σταθερή (Real constant) είναι κάθε δεκαδικός αριθμός σε απλή κλασματική μορφή ή σε κλασματική μορφή με εκθέτη. Στη δεύτερη περίπτωση, την ύπαρξη εκθέτη συμβολίζουμε με το γράμμα E, που συνοδεύεται από κάποιο διψήφιο άκεραιο προσσημασμένο αριθμό, που αποτελεί την αριθμητική τιμή του, π.χ.:

1.754	54.	783.015	1.53E+04	(-1.53 × 10 ⁴)
-0.08	2.0	221.3	-2.0E-2	(-2 × 10 ⁻²)

Τό πλήθος των ψηφίων που μπορεί να έχει μία άκεραια ή πραγματική σταθερή, εξαρτάται από τον τύπο του υπολογιστή.

Τίς μεταβλητές τις διακρίνουμε σε δύο κατηγορίες: α) Άπλές μεταβλητές και β) μεταβλητές με δείκτη. Η όνομασία μιας μεταβλητής (Variable name) μπορεί να πέ-

ριλαμβάνει μέχρι 6 χαρακτήρες αλφαβητικούς και αριθμητικούς, με τον πρώτο όμως υποχρεωτικά αλφαβητικό.

Οι μεταβλητές με δείκτη συμβολίζουν τα στοιχεία ενός πίνακα, όπως ήδη αναφέραμε στην παράγραφο 9.11. Στη FORTRAN IV γίνονται δεκτοί μέχρι τρεις δείκτες (τριδιάστατοι πίνακες).

Οι μεταβλητές, είτε είναι άπλες, είτε με δείκτη, χαρακτηρίζονται σαν άκεραιες (Integer) ή πραγματικές (Real).

Σαν *άκεραιη μεταβλητή* (Integer variable) χαρακτηρίζεται εκείνη της οποίας ο πρώτος χαρακτήρας είναι ένας από τους:

I, J, K, L, M, N

Μία άκεραιη μεταβλητή μπορεί να έχει σαν περιεχόμενο μόνο ακέραιους αριθμούς.

Σαν *πραγματική μεταβλητή* (Real variable) χαρακτηρίζεται εκείνη, της οποίας ο πρώτος χαρακτήρας είναι ένα από τα υπόλοιπα γράμματα του λατινικού αλφαβήτου. Μία πραγματική μεταβλητή δέχεται σαν περιεχόμενο μόνο πραγματικούς αριθμούς.

Χρειάζεται λοιπόν, προσοχή, αντίθετα προς τη BASIC, όπου χρησιμοποιούμε τις μεταβλητές χωρίς διάκριση.

Στόν πίνακα 10.2.1 βλέπουμε περιπτώσεις άκεραίων και πραγματικών μεταβλητών.

Πίνακας 10.2.1.

Παραδείγματα πραγματικών και άκεραίων μεταβλητών

Μεταβλητή	Χαρακτηρισμός
A12	Άπλη πραγματική μεταβλητή
B(23)	Πραγματική μεταβλητή με δείκτη
M(12,4)	Άκεραιη μεταβλητή με δείκτη
TJK	Άπλη πραγματική μεταβλητή
NAG	Άπλη άκεραιη μεταβλητή
GMIN	Άπλη πραγματική μεταβλητή

Γιά τό σχηματισμό της ονομασίας μιās μεταβλητής, πρέπει να έχουμε υπ' όψη ιας τούς εξής κανόνες:

α) Δεν πρέπει να έχει συνολικά περισσότερους από 6 χαρακτήρες.

β) Ο πρώτος χαρακτήρας να είναι όπωςσδήποτε αλφαβητικός.

γ) Να μην υπάρχουν ειδικοί χαρακτήρες (+, ·, —, * κλπ.)

ε) Να μη παρεμβάλλονται κενά (blanks) ανάμεσα στους χαρακτήρες.

Στόν Πίνακα 10.2.2 βλέπουμε περιπτώσεις σωστής και λάθους γραφής μεταβλητών.

Στίς μεταβλητές με δείκτη, θά πρέπει ο δείκτης να είναι σε άκεραιη μορφή, είτε σταθερή, είτε μεταβλητή, είτε και άπλη άκεραιη έκφραση FORTRAN. Δεν μπορεί όμως να παίρνει αρνητικές ή μηδενικές τιμές, π.χ.

A(J) B(4) TK(3,M) S12 (J,I) AMAX(JOB)

ΠΙΝΑΚΑΣ 10.2.2.

Παραδείγματα σωστής και λάθους γραφής μεταβλητών

Μεταβλητή	Χαρακτηρισμός
AGMIN	Σωστή
TJ23	Σωστή
41AB	Λάθος, γιατί ο πρώτος χαρακτήρας είναι αριθμός
R K12	Λάθος, γιατί υπάρχει κενό (blank) ανάμεσα στο R και τό K
MOSF	Σωστή
EQUATION	Λάθος, γιατί περιέχει περισσότερους από 6 χαρακτήρες
BAT+8	Λάθος, γιατί περιέχεται ο ειδικός χαρακτήρας +

β) Σύμβολα αριθμητικών πράξεων.

Όπως και στη BASIC, χρησιμοποιούμε και εδώ τα έξι σύμβολα για τις αριθμητικές πράξεις:

ΠΙΝΑΚΑΣ 10.2.3.

Τα σύμβολα των αριθμητικών πράξεων

Σύμβολο	Πράξη
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
**	Ύψωση σε δύναμη

Ποτέ δεν πρέπει να γράφουμε δύο σύμβολα τό ένα δίπλα στο άλλο, π.χ. η διάταξη $A+B$ είναι λάθος.

10.3 Έκφράσεις FORTRAN (FORTRAN expressions).

Οι εκφράσεις FORTRAN είναι άλγεβρικές παραστάσεις κωδικοποιημένες στη γλώσσα FORTRAN. Μία έκφραση FORTRAN αποτελείται από μεταβλητές και σταθερές, που συνδέονται μεταξύ τους με τα σύμβολα των πράξεων.

Η κωδικοποίηση γίνεται με ορισμένους κανόνες, όσον αφορά στη χρήση ακεραίων ή πραγματικών μεταβλητών και σταθερών.

Με τούς κανόνες αυτούς μπορούμε να κωδικοποιήσουμε μία άλγεβρική παράσταση με δύο τρόπους:

α) Με μερική ανάμιξη (Partially mixed mode).

β) Με πλήρη ανάμιξη (Fully mixed mode).

Οι υπολογιστές, που δέχονται μερική ανάμιξη, δεν μπορούν να δεχθούν πλήρη, ενώ όσοι δέχονται πλήρη ανάμιξη, δέχονται και μερική. Όπως και αν έχει έκφρασθει μία παράσταση, χαρακτηρίζεται και αυτή σαν άκερη ή πραγματική.

α) Μερική ανάμιξη.

Στή μερική ανάμιξη επιτρέπεται να υπάρχουν σέ μία σχέση, μεταβλητές καί σταθερές μόνο του ίδιου τύπου. Στούς παρακάτω πίνακες δίνονται οί επιτρεπόμενοι συνδυασμοί ανάμεσα σέ δύο όρους μιᾶς ἀπλῆς ἐκφράσεως, Α καί Β, καθώς καί ό τύπος τοῦ ἀποτελέσματος. Μέ τό Ι καί τό R παριστάνομε ἕνα ἀκέραιο ἢ πραγματικό ὄρο ἀντίστοιχα.

ΠΙΝΑΚΑΣ 10.3.1.

Οί ἐπιτρεπόμενοι συνδυασμοί στή μερική ανάμιξη

A \ B	I	R
I	I	—
R	—	R

Γιά τίς πράξεις +, *, —, /
ἀνάμεσα στό Α καί τό Β

A \ B	I	R
I	I	—
R	R	R

Γιά τήν ὕψωση σέ δύναμη ἀνάμεσα
στό Α (βάση) καί τό Β (ἐκθέτη)

Παραδείγματα κωδικοποιήσεως στή μερική ανάμιξη (Πίνακας 10.3.2)

ΠΙΝΑΚΑΣ 10.3.2.

Παραδείγματα ἐκφράσεων κωδικοποιημένων στή μερική ανάμιξη

Ἀλγεβρική παράσταση	Ἐκφραση FORTRAN
$a + b + c$	A + B + C
$2x + 4$	2. *X + 4.
$x^2 + \frac{6}{4}$	X** 2 + 6./ 4.
$b^2 - 4ac$	B ** 2 - 4. * A * C
$\frac{3ab}{5}$	3. * A * B / 5.

Ὅπως βλέπομε, μιᾶ ἀκέραιη σταθερή, πού ἐμφανίζεται σέ μιᾶ ἐκφραση, μπορούμε νά τή μετατρέψομε σέ πραγματική, προσθέτοντας ἀπλῶς μιᾶ τελεία. Ἡ ἀντίστροφη ἐνέργεια δέν μπορεῖ νά γίνει.

β) Πλήρη ανάμιξη.

Στήν πλήρη ανάμιξη εἶναι δεκτός κάθε συνδυασμός. Ὁ τύπος τοῦ ἀποτελέσματος φαίνεται στόν Πίνακα 10.3.3.

ΠΙΝΑΚΑΣ 10.3.3.

Οι έπιτρεπόμενοι συνδυασμοί στην πλήρη ανάμιξη

A \ B	I	R
I	I	R
R	R	R

Για όλες τις πράξεις ανάμεσα στο A και το B (+, -, *, /, **)

Παραδείγματα κωδικοποιήσεως σε πλήρη ανάμιξη (Πίνακας 10.3.4).**ΠΙΝΑΚΑΣ 10.3.4.**

Παραδείγματα έκφρασεων κωδικοποιημένων στην πλήρη ανάμιξη

Άλγεβρική παράσταση	Έκφραση FORTRAN
$2x^2 + 4x - 3$	$2 * X ** 2 + 4 * X - 3$
$a + \frac{3}{b} - k$	← Άκέραιη Πραγματική ← Άκέραιη Πραγματική ← Άκέραιη $A + 3 / B - K$
$\frac{6+x}{x} - \frac{1}{1+x}$	$(6 + X) / X - 1 / (1 + X)$

10.4 Συναρτήσεις του συστήματος.

Όπως αναφέραμε και στη BASIC, υπάρχουν ορισμένες συναρτήσεις του συστήματος, που εκτελούν διάφορους υπολογισμούς. Οι ονομασίες για τις αντίστοιχες συναρτήσεις FORTRAN διαφέρουν ελαφρά, όπως βλέπουμε και στον Πίνακα 10.4.1.

ΠΙΝΑΚΑΣ 10.4.1.

Οι συναρτήσεις συστήματος της FORTRAN

Όνομασία συναρτήσεως	Λειτουργία
SIN (X)	Υπολογίζει το ημίτονο του X
COS (X)	Υπολογίζει το συνημίτονο του X
ATAN(X)	Υπολογίζει το τόξο έφαπτομένης του X
EXP (X)	Υπολογίζει το e^x
SQRT(X)	Υπολογίζει την τετραγωνική ρίζα του X
ABS(X)	Υπολογίζει την απόλυτη τιμή του X
ALOG(X)	Υπολογίζει το φυσικό λογάριθμο του X
ALOG10(X)	Υπολογίζει το δεκαδικό λογάριθμο του X
FLOAT(N)	Μετατρέπει τον άκέραιο N σε κλασματικό με προσθήκη μιάς τελείας
IFIX(X)	Μετατρέπει τον κλασματικό X σε άκέραιο

Οι συναρτήσεις αυτές, πού είναι όλες τύπου Real, αναφέρονται σάν απλοί όροι μέσα σέ μία έκφραση FORTRAN. Παραδείγματα δίνονται στόν Πίνακα 10.4.2.

ΠΙΝΑΚΑΣ 10.4.2.

Παραδείγματα κωδικοποιήσεων εκφράσεων πού περιέχουν συναρτήσεις.

Άλγεβρική παράσταση	Έκφραση FORTRAN
$\frac{4}{3} \text{pr}^3 e^x$	$(4./ 3.* 3. 1415 * R ** 3) * \text{EXP}(X)$
$\eta \mu x + 2 \sigma \upsilon \nu x + 4$	$\text{SIN}(X) + 2. * \text{COS}(X) + 4.$
$\sqrt{x^2 - 1} + x $	$\text{SQRT}(X** 2 - 1.) + \text{ABS}(X)$
$a + e^x$	$A + \text{EXP}(X)$
$2 \lambda \omicron \gamma x + \beta e^{-x}$	$2. * \text{ALOG10}(x) + B * \text{EXP}(-X)$
$\eta \mu^2 x + \sigma \upsilon \nu^2 x$	$\text{SIN}(X) ** 2 + \text{COS}(X) ** 2$

10.5 Παρενθέσεις.

Χρησιμοποιούνται καί εδώ μέ τόν ίδιο ακριβώς τρόπο, πού τίς χρησιμοποιήσαμε καί στή BASIC.

10.6 Ίεραρχία στήν εκτέλεση τών πράξεων.

Ίσχύουν τά όσα αναφέραμε στή BASIC. Άκολουθείται δηλαδή ή εξής σειρά στήν εκτέλεση τών πράξεων σέ μία έκφραση FORTRAN:

- Υψώσεις σέ δύναμη.
- Πολλαπλασιασμοί καί διαιρέσεις.
- Προσθέσεις καί αφαιρέσεις.

Άν υπάρχουν παρενθέσεις, υπολογίζονται πρώτα οί παραστάσεις πού περιέχονται σ' αυτές, μέ σειρά υπολογισμού πρώτα τών έσωτερικών καί ύστερα τών έξωτερικών.

10.7 Λογικές εκφράσεις.

α) Άπλές λογικές εκφράσεις.

Στή FORTRAN, όπως καί στή BASIC, μπορούμε νά κωδικοποιήσουμε μέ ειδικά σύμβολα, λογικές σχέσεις μεταξύ δύο όρων. Τά σύμβολα αυτά βλέπομε στόν Πίνακα 10.7.1.

Σημειώνομε ότι κατά τήν κωδικοποίηση μιās απλής λογικής σχέσεως οί δύο όροι θά πρέπει νά είναι τοῦ ἴδιου τύπου, πραγματική-πραγματική ή άκέραιη-άκέραιη (Real-Real, Integer-Integer) έφ' όσον χρησιμοποιούμε, όπως είπαμε, τή μερική ανάμιξη.

ΠΙΝΑΚΑΣ 10.7.1.

Τά σύμβολα λογικών σχέσεων της FORTRAN

Μαθηματικό λογικό σύμβολο	Λογικό σύμβολο FORTRAN	Έπεξήγηση
=	.EQ.	ίσο προς
≠	.NE.	άνισο
<	.LT.	μικρότερο από
≤	.LE.	μικρότερο προς ή ίσο
>	.GT.	μεγαλύτερο από
≥	.GE.	μεγαλύτερο ή ίσο

Παραδείγματα κωδικοποιήσεως απλών λογικών σχέσεων (Πίνακας 10.7.2).

ΠΙΝΑΚΑΣ 10.7.2.

Παραδείγματα κωδικοποιήσεως απλών λογικών σχέσεων

Μαθηματική λογική σχέση	Λογική έκφραση FORTRAN
$a < b$	A .LT. B
$x_1 \neq 4$	X (1) . NE . 4 .
$\psi + 1 \geq z^2$	(Y + 1.) . GE . Z** 2
$3 \leq z$	3.. .LE. Z
$\frac{a}{b} \neq \frac{\gamma}{\delta}$	(A / B) . NE . (C / D)

β) Σύνθετες λογικές εκφράσεις.

Πολλές φορές στά μαθηματικά, μία λογική σχέση είναι σύνθετη, αποτελούμενη από απλές, που συνδέουν περισσότερους από δύο όρους, π.χ. $a < b < \gamma$, $4 \leq x < 3$, κλπ. Τίς σχέσεις αυτές μπορούμε νά τίς διασπάσουμε σέ απλές, π.χ. τήν $a < b < \gamma$ στίς $a < b$ **καί** $b < \gamma$. 'Ακόμη συναντούμε περιπτώσεις, όπου πρέπει νά ικανοποιούνται συγχρόνως δύο ή περισσότερες, ξεχωριστές, απλές ή σύνθετες λογικές σχέσεις, π.χ. λέμε ότι κάποιο σύστημα έχει λύση, αν $x \leq 0$ **καί** $y > 0$, ή ακόμη ότι μία εξίσωση έχει πραγματικές ρίζες, όταν $x \leq a$ **ή** $\psi < b < 0$ κ.ο.κ. Τό **ή** καί τό **καί** συνδέουν τίς ξεχωριστές συνθήκες μεταξύ τους. Στή FORTRAN χρησιμοποιούνται γιά τόν ίδιο σκοπό τά λεγόμενα λογικά σύμβολα συσχετίσεως . OR . καί . AND .

Παραδείγματα κωδικοποιήσεως συνθέτων λογικών σχέσεων βλέπομε στόν Πίνακα (10.7.3).

ΠΙΝΑΚΑΣ 10.7.3.

Παραδείγματα κωδικοποίησης συνθέτων λογικών σχέσεων

Μαθηματική λογική σχέση	Λογική έκφραση FORTRAN
$0 \leq x < 3$ $\alpha \neq 0$ και $\beta < \gamma$ $x = \psi$ ή $z = 2$ $x > \alpha$ ή $\psi < \beta < \gamma$	O..LE.X. AND. X. LT. 3. A. NE. O.. AND. B. LT. C X. EQ. Y. OR. Z. EQ. 2. X. GT. A. OR. Y. LT. B. AND. B. LT. C

10.8 Κατηγορίες εντολών FORTRAN.

Διακρίνουμε τīs έξής κατηγορίες εντολών FORTRAN:

- Έντολές άντικαταστάσεως.
- Έντολές έλέγχου ή διακλαδώσεως.
- Έντολές εισόδου-έξόδου.
- Δηλωτικές εντολές.

10.9 Περιγραφή του έντύπου κωδικογραφήσεως FORTRAN (FORTRAN Coding Form).

Τīs εντολές ενός προγράμματος FORTRAN τīs γράφομε σέ ένα είδικό έντυπο, τό φύλλο κωδικογραφήσεως FORTRAN (FORTRAN Coding Form σχ. 10.9), μέ τή σειρά πού θά έκτελεσθούν.

Τό φύλλο αυτό περιλαμβάνει όρισμένο άριθμό γραμμών. Κάθε γραμμή χωρίζεται σέ 80 στήλες, πού άριθμούνται άπό τό 1-80. Σέ κάθε γραμμή γράφομε μόνο μία έντολή, έτσι ώστε σέ κάθε στήλη-θέση νά καταχωρίζεται καί ένας χαρακτήρας. Σημειώνομε έδω ότι καί τό κόμμα (,), ή τελεία (.), τό ίσον (=) καί όλοι οι άλλοι είδικοί χαρακτήρες, καταχωρίζονται σέ ξεχωριστή στήλη του έντύπου. Τό περιεχόμενο μιάς γραμμής του έντύπου μεταφέρεται καί διατυπώ σέ ένα 80-στήλο δελτίο.

Στīs στήλες 1-5 γράφομε τόν άριθμό έντολής. Δέν είναι ύποχρεωτική ή άρίθμηση κάθε έντολής, όπως στή BASIC. Μπορούμε νά χρησιμοποιούμε άριθμούς μέχρι πενταψηφίους, χωρίς διάκριση καί χωρίς νά βρίσκονται σέ οποιαδήποτε σειρά (στή BASIC πρέπει νά είναι σέ αύξουσα σειρά).

Η στήλη 6 συμπληρώνεται μέ ένα χαρακτήρα, όταν ή γραμμή, στήν όποία άνήκει, άποτελεί συνέχεια τής προηγούμενης έντολής (Continuation), πού έπειδή ήταν μεγάλη, δέν μπόρεσε νά χωρέσει σέ μία γραμμή.

Στīs στήλες 7-72 καταχωρίζομε τήν έντολή (ξεκινώντας άπό τή στήλη 7).

Οί στήλες 73-80 χρησιμοποιούνται για τήν άναγραφή τής κωδικής όνομασίας του προγράμματος (άγνοούνται κατά τή φάση τής μεταφράσεως του προγράμματος).

10.10 Άριθμητική έντολή άντικαταστάσεως.

Είναι ή πιο συνηθισμένη άπό τīs εντολές τής FORTRAN, άντίστοιχη πρός τήν

FORTRAN CODING FORM

HMEPOMHNIA

ΣΕΛΙΔΑ ΑΠΟ

ПРОГРАММА

ΠΡΟΓΡΑΜΜΑΤΙΣΤΗΣ

— ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ

ΑΡΙΘΜΟΣ	3
ΕΝΤΟΛΗΣ	567

ENTOAH FORTAN

72

Σχ. 10.9.

Εντυπο κωδικογράφησης FORTRAN.

έντολή LET της BASIC. Η μορφή της είναι:

Μεταβλητή = έκφραση FORTRAN

Η μεταβλητή μπορεί να είναι απλή ή με δείκτη και ο τύπος της κανονικά πρέπει να είναι ίδιος με τον τύπο της έκφρασης του δευτέρου μέλους (πραγματική-πραγματική, άκέραιη-άκέραιη).

Η έντολή, λειτουργεί ως εξής: Υπολογίζεται πρώτα η αριθμητική τιμή της έκφρασης του δευτέρου μέλους και τοποθετείται ύστερα σάν περιεχόμενο της μεταβλητής του πρώτου μέλους, αντικαθιστώντας κάθε προηγούμενο περιεχόμενό της. Φυσικά θά πρέπει όλοι οι όροι της έκφρασης να έχουν γνωστή τιμή.

Στήν έντολή π.χ.:

$$A = X ** 2 - 3.*X + 4.$$

υπολογίζεται πρώτα η τιμή της $X^2 - 3X + 4$ και τοποθετείται κατόπιν στή μεταβλητή A, αντικαθιστώντας τό προηγούμενο περιεχόμενό της. Η τιμή της X πρέπει να είναι γνωστή.

Παραδείγματα.

$$R = \text{SQRT}(Y**2 - B*A*C) + 4.$$

$$P = 3.14159 + 4.* \text{SIN}(X)$$

$$I = I + 3$$

$$X = A(1) + B(4) ** 2$$

$$P(I) = A(1,I) + B(J,2)$$

Όταν η μεταβλητή είναι άκέραιη και η έκφραση πραγματική, τότε στήν άκέραιη μεταβλητή μεταφέρεται μόνο τό άκέραιο μέρος από τήν υπολογιζόμενη τιμή της έκφρασης. Π.χ.:

άν $[X] = 1.4$ και $[Y] = 2.8$ και εκτελεσθεῖ ἡ έντολή

$$I = 2.*X + Y$$

τότε τό αποτέλεσμα θά είναι $[I] = 5$, αντί του σωστού 5.6.

Όταν η μεταβλητή είναι πραγματική και η έκφραση άκέραιη, τότε τό αποτέλεσμα του υπολογισμού της έκφρασης μετατρέπεται σέ πραγματικό μέ τήν προσθήκη της κλασματικής τελείας (.) στό τέλος του αριθμού. Π.χ.:

άν $[I] = 12$ και $[J] = 3$ και εκτελεσθεῖ ἡ έντολή

$$S = 3*I - J$$

τότε τό αποτέλεσμα θά είναι $[S] = 33$.

10.11 Έντολές ελέγχου και διακλαδώσεως.

α) Έντολή GO TO χωρίς συνθήκη.

Η μορφή της έντολης είναι:

|| GO TO n

όπου n φυσικός αριθμός, που παριστάνει τόν αριθμό της έντολης, στήν οποία μεταφέρεται ο έλεγχος του προγράμματος. 'Η διπλή γραμμή άριστερά συμβολίζει τήν 6η στήλη του έντύπου κωδικογραφήσεως.

Παράδειγμα 1.

```

|| GO TO 35
|| .....
35 || X = X + 4

```

Μέ τήν εκτέλεση της GO TO 35, σάν επόμενη έντολή θά εκτελεσθεῖ ἡ $X = X + 4$ ἐνῶ ὅλες οἱ ἐνδιάμεσες ἀγνοοῦνται.

Παράδειγμα 2.

```

12 || S = S + 4.*A
|| .....
|| GO TO 12

```

Άμέσως μετά τήν εκτέλεση της GO TO 12 θά εκτελεσθεῖ ἡ $S = S + 4.*A$ καί κατόπιν ὅλες οἱ επόμενες έντολές, που βρίσκονται κάτω ἀπό αὐτή.

β) Έντολή GO TO μέ συνθήκη.

Ἡ μορφή της έντολης αὐτῆς εἶναι:

|| GO TO ($n_1, n_2, n_3, \dots$), 'Ακέραιη μεταβλητή ἢ ἔκφραση,

όπου $n_1, n_2, \dots$ = ἀριθμοί έντολῶν.

Λειτουργεῖ ὡς ἑξῆς: "Αν ἡ ἀριθμητική τιμή της ἀκέραιης μεταβλητῆς ἢ ἔκφρασεως εἶναι 1 ἢ 2, ἢ 3 κ.ο.κ, θά εκτελεσθεῖ σάν επόμενη ἡ έντολή, μέ ἀριθμό τόν n_1 , ἢ n_2 ἢ n_3 κ.ο.κ. Π.χ.:

```

34 || X = X + B
|| .....
|| GO TO (15, 25, 34, 42), I
42 || X = X + B ** 2
|| .....
25 || X = X + B-A
|| .....
15 || X = 2.*B
|| .....

```


“Αν ή τιμή του I τή στιγμή πού έκτελεΐται ή έντολή GO TO εΐναι 1, τότε θά έκτελεσθεΐ άμέσως μετά ή έντολή 15 καΐ θά παραλειφθοϋν όλες οι ένδιάμεσες. “Αν ή τιμή του I εΐναι 3, τότε θά έκτελεσθεΐ στή συνέχεια ή 34 καΐ κατόπιν όλες οι κάτω από τήν 34 έντολές κ.ο.κ.

γ) Άριθμητική έντολή IF.

“Η δομή τής έντολής εΐναι:

|| IF (μεταβλητή ή έκφραση FORTRAN) άριθ. έντολής 1, άριθ. έντολής 2, άριθ. έντολής 3.

“Η έντολή IF λειτουργεί ως εξής: “Ελέγχεται ή τιμή τής μεταβλητής ή τής έκφράσεως FORTRAN μέσα στήν παρένθεση. “Αν εΐναι άρνητική (< 0), τότε σάν επόμενη θά έκτελεσθεΐ ή έντολή μέ άριθμό, τόν άριθμό έντολής 1. “Αν ή τιμή εΐναι 0, θά έκτελεσθεΐ ή έντολή μέ άριθμό τόν δεύτερο πού εμφανίζεται στήν έντολή καΐ τέλος, αν ή τιμή εΐναι θετική (> 0), θά έκτελεσθεΐ ή έντολή μέ άριθμό τόν τρίτο πού εμφανίζεται στό δεξιό τής έντολής IF.

“Η άναγραφή καΐ τών τριών άριθμών στήν έντολή IF εΐναι ύποχρεωτική, διαχωρίζονται δέ μεταξύ τους μέ κόμματα.

Παράδειγμα 1.

```

22 || .....
    || .....
    || IF (I + K) 22, 44, 33
44 || .....
    || .....
33 || .....
    || .....

```

“Αν ή τιμή του $I + K$, τή στιγμή πού έκτελεΐται ή IF, εΐναι < 0 , τότε θά έκτελεσθεΐ σάν επόμενη, ή έντολή 22. “Αν $I + K = 0$, τότε θά έκτελεσθεΐ σάν επόμενη έντολή ή 44. Τέλος, αν $I + K > 0$, θά έκτελεσθεΐ σάν επόμενη έντολή ή 33.

Παράδειγμα 2.

```

18 || IF (A) 21, 21, 18
    || .....
    || .....
    || .....
21 || .....

```

“Εδώ, αν ή τιμή τής μεταβλητής A εΐναι μικρότερη ή ίση μέ τό μηδέν (≤ 0), τότε θά έκτελεσθεΐ ή έντολή 21. “Αν εΐναι μεγαλύτερη του μηδενός (> 0), θά έκτελεσθεΐ ή έντολή 18.

Παράδειγμα 3.

	IF (B**2-4.* A*C) 52, 51, 51
52	
	
51	
	

“Αν ή τιμή τής έκφράσεως μέσα στην παρένθεση είναι < 0 , τότε θά εκτελεσθεϊ ή έντολή 52, άν δέ είναι ≥ 0 , τότε θά εκτελεσθεϊ ή 51 μέ παράλειψη όλων τών ένδιαμέσων έντολών.

δ) Λογική έντολή IF.

Έχει τή μορφή:

IF (λογική έκφραση FORTRAN) έντολή 1
έντολή 2

“Η έντολή λειτουργεί ως έξης: Έλέγχεται ή λογική έκφραση μέσα στή παρένθεση. “Αν ή συνθήκη πού θέτει πληροϋται, τότε εκτελείται ή έντολή 1, πού βρίσκεται έξω από τήν παρένθεση καί τό πρόγραμμα συνεχίζεται κανονικά. “Αν ή συνθήκη δέν ισχύει, τότε ή έντολή 1 άγνοείται καί σάν επόμενη εκτελείται ή έντολή 2.

“Η έντολή 1 δέν πρέπει νά είναι μία άλλη έντολή IF (λογική ή αριθμητική) ή μία έντολή DO, πού θά γνωρίσαμε στή συνέχεια.

Παράδειγμα 1.

	IF(X. LE. 0.) GO TO 44
	X = X + 4.
	
44	X = SQRT(5.)
	

“Αν ή συνθήκη μέσα στην παρένθεση ισχύει, δηλαδή άν $X \leq 0$, τότε ό έλεγχος θά μεταφερθεϊ στην έντολή 44 (άφοϋ εκτελεσθεϊ ή GO TO 44). “Αν ή συνθήκη μέσα στην παρένθεση δέν ισχύει, δηλαδή άν $X > 0$, τότε ή έντολή GO TO 44 άγνοείται καί εκτελείται σάν επόμενη ή $X = X + 4$.

Παράδειγμα 2.

IF (X. EQ. 0.. AND. Y. NE. 0.) S = X + 5.*Y
I = I + 1
.....
.....

"Αν $X = 0$ **και** $Y \neq 0$, τότε θα εκτελεσθεί ή $S = X + 5 * Y$ και κατόπιν ή επόμενη, πού είναι ή $I = I + 1$. "Αν όμως $X \neq 0$ **και** $Y = 0$ ή $X = 0$ **και** $Y = 0$ ή $X \neq 0$ **και** $Y \neq 0$, τότε ή $S = X + 5 * Y$ θα άγνοηθεϊ και θα εκτελεσθεί σάν επόμενη έντολή ή $I = I + 1$ και όλες οι επόμενες.

10.12 Έντολή DO.

"Όπως είπαμε και στη BASIC, συχνά χρειάζεται να επαναλαμβάνεται ή εκτέλεση ενός τμήματος του προγράμματος περισσότερες από μία φορές. Στις περιπτώσεις αυτές χρησιμοποιούμε μία αντίστοιχη προς ή FOR-NEXT έντολή της BASIC, ή DO πού συνοδεύεται από ήν έντολή CONTINUE.

Η μορφή της είναι:

n		DO n άκέραιη μεταβλητή = k_1, k_2, k_3	}	Έντολές του DO
				
				
				
				
		CONTINUE		

n = Φυσικός αριθμός πού παριστάνει τόν αριθμό έντολης του CONTINUE. k_1, k_2, k_3 = Άκέραιες σταθερές ή μεταβλητές μέ τιμές μόνο θετικές.

Η πρώτη και ή δεύτερη, (k_1 και k_2) αποτελούν αντίστοιχα ήν πρώτη και ήν τελευταία τιμή πού μπορεί να πάρει ή άκέραιη μεταβλητή (δείκτης του DO). Η τρίτη (k_3), αποτελεί τό βήμα. Όταν λείπει, υποτίθεται ότι τό βήμα = 1. Τά k_1, k_2, k_3 χωρίζονται μεταξύ τους μέ κόμματα.

Η ομάδα των έντολών πού βρίσκεται μεταξύ των DO και CONTINUE θα εκτελεσθεί τόσες φορές, όσο είναι τό πλήθος των τιμών πού μπορεί να πάρει ό δείκτης του DO.

Παράδειγμα 1.

14		IS = 0
		DO 14 I = 1,3
		IS = IS + I
		CONTINUE
		

Έδώ ό δείκτης του DO, I μπορεί να πάρει τις τιμές 1, 2 και 3. Τό ότι λείπει ή τρίτη σταθερή k_3 , σημαίνει ότι τό βήμα είναι 1. Μέσα στό DO υπάρχει μόνο μία έντολή, ή $IS = IS + I$, πού θα εκτελεσθεί συνολικά τρεις φορές. Στο δεύτερο μέλος της $IS = IS + I$ συμμετέχει και ό δείκτης του DO μέ τις εκάστοτε τιμές του.

Ας δοϋμε τις διάφορες φάσεις εκτέλεσεως των έντολών αυτών.

Στόν πρώτο κύκλο εκτελέσεως, τό $I = 1$ και ή $IS = 0 + 1 = 1$. Μετά επανέρχεται, λόγω της CONTINUE, στη DO, όπου τό I γίνεται 2. Έκτελείται για δεύτερη φο-

ρά ή $IS = IS + I = 1 + 2 = 3$ και έπιστρέφει πάλι στή DO όπου τό I παίρνει τήν τελευταία τιμή 3, εκτελείται ή $IS = IS + I = 3 + 3 = 6$ και κατόπιν ό έλεγχος του προγράμματος βγαίνει έξω από τό DO , γιά νά συνεχισθεί ή εκτέλεση των επομένων εντολών του προγράμματος.

Σημειώνομε ότι, όταν τελευταία εντολή του DO είναι μία εντολή αντικαταστάσεως ή εισόδου-εξόδου, τότε μπορούμε νά μήν αναφέρομε καθόλου τήν $CONTINUE$, τό δέ αριθμό εντολής της νά τόν θέσομε στήν τελευταία αυτή εντολή του DO .

Παράδειγμα 2.

```

DO 15 I = 1,4
  K(I) = 3*I - 2
  L(I) = I**2
15

```

Στό παράδειγμα αυτό ή εντολή $CONTINUE$ δέν υπάρχει και ό αριθμός εντολής της έχει τεθεί στήν τελευταία εντολή του DO , πού είναι μία εντολή αντικαταστάσεως.

Μέσα στό DO υπάρχουν δύο εντολές: ή $K(I) = 3*I - 2$ και ή $L(I) = I**2$, πού θά εκτελεσθούν συνολικά 4 φορές ή κάθε μία (δσες και οι τιμές πού μπορεί νά πάρει ό δείκτης του DO , I).

Βλέπομε ακόμη ότι ό δείκτης I δανείζει τίς τιμές του τόσο στους δείκτες των μεταβλητών $K(I)$ και $L(I)$, όσο και στα δεύτερα μέλη των εντολών.

Στήν πρώτη εκτέλεση τό I έχει τιμή 1 και υπολογίζονται οι τιμές των στοιχείων $K(1) = 3*1 - 2 = 1$ και $L(1) = 1**2 = 1$. Στή δεύτερη υπολογίζονται δύο νέα στοιχεία των πινάκων K και L , τά $K(2) = 3*2 - 2 = 4$ και $L(2) = 2**2 = 4$. Στήν τρίτη υπολογίζονται τά $K(3) = 3*3 - 2 = 7$ και $L(3) = 3**2 = 9$. Στήν τελευταία εκτέλεση υπολογίζονται τά $K(4) = 3*4 - 2 = 10$ και $L(4) = 4**2 = 16$. Ό έλεγχος κατόπιν οδηγεί έξω από τό DO , γιά νά συνεχισθεί μέ τή σειρά, ή εκτέλεση των υπολοίπων εντολών του προγράμματος.

Σημειώνομε ότι ενώ είναι δυνατή ή έξοδος από ένα DO , γιά τήν εκτέλεση μιās άλλης εντολής, πρίν συμπληρωθούν όλοι οι προβλεπόμενοι κύκλοι του (έξ αίτίας μιās εντολής διακλαδώσεως πού ανήκει στήν ομάδα εντολών του), είναι αδύνατη ή είσοδος μέσα σέ οποιαδήποτε εντολή του DO . Πχ.

```

DO 20
  K = 1
  DO 10 I = 1, 3
    IF (I + K. EQ. 3) GO TO 5
  CONTINUE
  K = K + 1
  GO TO 20

```

Σωστό

```

GO TO 18
DO 20 I = 1, 5, 2
  IX = IX + I
  NX = NX + I**2
  MX = MX + I**3
20 CONTINUE

```

Λάθος

10.13 Μερικές απλές εφαρμογές.

Παράδειγμα 1.

Νά γραφεί πρόγραμμα πού νά υπολογίζει τό άθροισμα των 20 πρώτων φυσι-

κῶν ἀριθμῶν 1, 2, 3,..., 20.

Στὴν περίπτωση αὐτὴ δὲν χρειάζεται νά εἰσαχθοῦν δεδομένα, γιατί τοὺς φυσικοὺς ἀριθμοὺς μπορεῖ νά μᾶς τοὺς δώσει ὁ δείκτης ἐνός DO, ἂν ἡ ἀρχικὴ τιμὴ τοῦ εἶναι 1, τελικὴ ἢ 20 καὶ τὸ βῆμα 1.

Ἔτσι τὸ πρόγραμμα θά ἔχει τὴν ἐξῆς μορφή:

```
10 || IS = 0
    || DO 10 I = 1,20
    || IS = IS + 1
```

IS εἶναι ὁ ἀθροιστής πού στό τέλος θά περιέχει τὸ ζητούμενο ἀθροισμα.

Παράδειγμα 2.

Νά ὑπολογισθεῖ ἡ τιμὴ τῆς παραστάσεως:

$$\prod_{v=1}^{12} \frac{1}{v}$$

Ἐδῶ τὸ πρόγραμμα θά εἶναι:

```
30 || P = 1.
    || DO 30 N = 1,12
    || P = P*1./FLOAT(N)
```

10.14 Ἐντολὲς εἰσόδου-ἐξόδου.

α) Ἐντολὴ READ.

Μέ τὴ βοήθειά της μποροῦν νά διαβάζονται δεδομένα ἀπὸ κάποια περιφερειακὴ μονάδα (συνήθως ἀναγνώστικὴ δελτίων). Ἡ μορφή της εἶναι:

|| READ (i, n) μεταβλητή, μεταβλητή,...

Δηλαδή ἀποτελεῖται ἀπὸ τὴν κωδικὴ λέξη READ καὶ μιά παρένθεση, μέσα στὴν ὁποία γράφομε δύο φυσικοὺς ἀριθμοὺς, χωρισμένους μέ κόμμα. Ὁ πρῶτος, πού ἐδῶ τὸν συμβολίσαμε μέ i, εἶναι ὁ κωδικὸς ἀριθμὸς τῆς περιφερειακῆς μονάδας, ἀπὸ ὅπου γίνεται ἡ ἀνάγνωση. Ὁ κωδικὸς αὐτὸς εἶναι ἴδιος γιὰ μιά συγκεκριμένη περιφερειακὴ μονάδα καὶ ἀλλάζει μέ τὸν τύπο τῆς μονάδας καθὼς καὶ ἀπὸ κατασκευαστὴ σέ κατασκευαστὴ. Γιὰ τὴν ἀναγνώστικὴ δελτίων θά χρησιμοποιήσουμε τὸ 5.

Ὁ δεῦτερος ἀριθμὸς n εἶναι ὁ ἀριθμὸς ἐντολῆς FORMAT. Μέ τὴν ἐντολὴ αὐτὴ, ὅπως θά δοῦμε στὴ συνέχεια, περιγράφουμε τὴ μορφή, μέ τὴν ὁποία εἶναι καταχωρισμένα τὰ δεδομένα ἐπάνω στὰ δελτία.

Οἱ μεταβλητὲς πού ἐμφανίζονται ἔξω ἀπὸ τὴ παρένθεση μπορεῖ νά εἶναι ἀπλές ἢ μέ δείκτη, ἀκέραιες ἢ πραγματικές. Θά δεχθοῦν σάν περιεχόμενο τὶς τιμές τῶν ἀντιστοιχῶν δεδομένων πού διαβάζονται.

Παράδειγμα 1.

```

24 || READ (5, 24) A, B, C
    || FORMAT (. . . . .
    || . . . . .

```

Σημαίνει ότι διαβάζονται από την αναγνωστική δελτίων ($i = 5$) και με τρόπο που περιγράφεται στην έντολη 24 FORMAT, τīs τιμές που θά πάρουν οι μεταβλητές A, B, C. Έτσι, τό πρώτο δεδομένο που θά διαβασθεί θά τοποθετηθεί σάν περιεχόμενο του A, τό δεύτερο σάν περιεχόμενο του B και τό τρίτο σάν περιεχόμενο του C.

Παράδειγμα 2.

```

32 || READ (5, 32) A(1), I
    || FORMAT (. . . . .
    || . . . . .

```

Έδω διαβάζονται από την αναγνωστική δελτίων ($i = 5$) και με τρόπο που περιγράφεται στην έντολη 32 δύο τιμές, από τīs όποιες ή πρώτη θά αποτελέσει τό περιεχόμενο της A(1) και ή δεύτερη τό περιεχόμενο της I.

β) Έντολη WRITE.

Μέ τή βοήθειά της γράφονται περιεχόμενα μεταβλητών σέ κάποια περιφερειακή μονάδα, συνήθως στόν έκτυπωτή. Η μορφή της είναι:

```

|| WRITE (i, n) μεταβλητή, μεταβλητή,...

```

i είναι ένας φυσικός αριθμός, που συμβολίζει τόν κωδικό της περιφερειακής μονάδας, όπου γίνεται ή έγγραφη των τιμών. Αλλάζει από τύπο σέ τύπο και από τή μία κατασκευάστρια εταιρεία στήν άλλη. Έδω θά χρησιμοποιήσομε τό $i = 6$ για έγγραφη στόν έκτυπωτή. n είναι ό αριθμός FORMAT, που προσδιορίζει τή μορφή της έγγραφης. Έξω από τήν παρένθεση παρατίθενται οι μεταβλητές, απλές ή μέ δείκτη, τό περιεχόμενο των οποίων θέλομε νά έγγραφει στήν περιφερειακή μονάδα.

Παράδειγμα 1.

```

4 || WRITE (6,4) A, B
    || FORMAT (. . . . .

```

Θά γραφούν στόν έκτυπωτή ($i = 6$) τά περιεχόμενα των πραγματικών μεταβλητών A και B, μέ τρόπο που περιγράφεται στην έντολη 4 FORMAT.

Παράδειγμα 2.

```

2 || WRITE (6,2) C, B(12), D
    || FORMAT (. . . . .

```

Θά εκτυπωθούν οι τιμές-περιεχόμενα των μεταβλητών C, B(12) καί D, με τρόπο που περιγράφεται στην έντολή 2 FORMAT.

10.15 Έντυπο σχεδιασμού έκτυπώσεων (Printer spacing chart).

Αποτελείται από όρισμένο αριθμό γραμμών (σχ. 10.15). Κάθε γραμμή αντι-

ROW	COL	PRINTED POSITION
1	1	
2	1	
3	1	
4	1	
5	1	
6	1	
7	1	
8	1	
9	1	
10	1	
11	1	
12	1	
13	1	
14	1	
15	1	
16	1	
17	1	
18	1	
19	1	
20	1	
21	1	
22	1	
23	1	
24	1	
25	1	
26	1	
27	1	
28	1	
29	1	
30	1	
31	1	
32	1	
33	1	
34	1	
35	1	
36	1	
37	1	
38	1	
39	1	
40	1	
41	1	
42	1	
43	1	
44	1	
45	1	
46	1	
47	1	
48	1	
49	1	
50	1	
51	1	
52	1	
53	1	
54	1	
55	1	
56	1	
57	1	
58	1	
59	1	
60	1	
61	1	
62	1	
63	1	
64	1	
65	1	
66	1	
67	1	
68	1	
69	1	
70	1	
71	1	
72	1	
73	1	
74	1	
75	1	
76	1	
77	1	
78	1	
79	1	
80	1	
81	1	
82	1	
83	1	
84	1	
85	1	
86	1	
87	1	
88	1	
89	1	
90	1	
91	1	
92	1	
93	1	
94	1	
95	1	
96	1	
97	1	
98	1	
99	1	
100	1	

Σχ. 10.15.
Τό έντυπο σχεδιασμού έκτυπώσεων (Printer Spacing Chart).

στοιχεί σε μία λογική ένότητα (Record) έκτυπωσης και χωρίζεται σε στήλες. 'Ανάλογα με τον τύπο και τις δυνατότητες του έκτυπωτή μπορούν να έκτυπωθούν σε μία γραμμή μέχρι και 128, 132, 136 ή 144 χαρακτήρες.

10.16 FORMAT.

FORMAT είναι μία μη εκτελέσιμη εντολή της FORTRAN, με τη βοήθεια της οποίας περιγράφουμε τον τρόπο που θα διαβασθούν ή θα εγγραφούν πληροφορίες από ή σε κάποια λογική ένότητα, όπου μπορούν να καταχωρισθούν περισσότερες από μία πληροφορίες, που μπορεί να είναι αριθμοί, αλφαριθμητικοί χαρακτήρες ή ακόμη άλλες να είναι πραγματικές και άλλες άκεραιες.

Κάθε ξεχωριστή πληροφορία (πεδίο) μιας λογικής ένότητας περιγράφεται με κάποιο κωδικό FORMAT. 'Η μορφή της είναι:

n || FORMAT (κωδικός, κωδικός,...)

Οι κωδικοί που αναφέρουμε μέσα στην παρένθεση, χρησιμοποιούνται για την περιγραφή των διαφόρων τύπων πεδίων και φαίνονται στον πίνακα 10.16.1.

ΠΙΝΑΚΑΣ 10.16.1.

Κωδικοί FORMAT

Κωδικός	Πού χρησιμοποιείται
Iw	Για άκεραία αριθμητικά πεδία (Integer)
Fw.d	Για απλά πραγματικά αριθμητικά πεδία (Real)
Ew.d	Για πραγματικά αριθμητικά πεδία με εκθέτη
wX	Για κενά μεσοδιαστήματα
wH	Για άπλη εμφάνιση χαρακτήρων (μηνύματα).

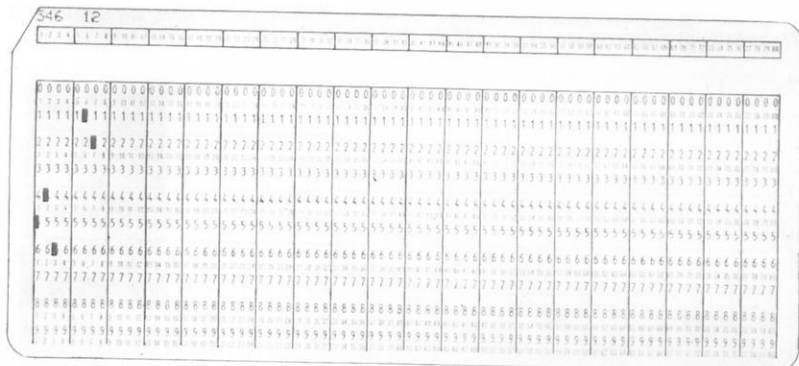
Θα περιγράψουμε τώρα τη χρήση καθενός από τους παραπάνω κωδικούς.

Κωδικός Iw.

Χρησιμοποιείται για την ανάγνωση ή έγγραφη ενός ακεραίου αριθμητικού πεδίου, του οποίου το μέγεθος σε πλήθος χαρακτήρων είναι w (w φυσικός αριθμός).

α) Για την ανάγνωση.

Αν ο αριθμός έχει λιγότερους χαρακτήρες από το μέγεθος του πεδίου, θα πρέπει να τον καταχωρίσουμε στις άκραίες δεξιές θέσεις του πεδίου. Για το δελτίο του παραδείγματος που ακολουθεί θα δώσουμε FORMAT (I3, I4).



β) Για την έκτύπωση.

Για να γράψουμε τό περιεχόμενο ενός άκεραίου πεδίου στον έκτυπωτή, πρέπει να έχουμε υπ' όψη μας ότι, αν δώσουμε στον κωδικό l σαν w ένα αριθμό μεγαλύτερο από τό πλήθος των ψηφίων του αριθμού πού πρόκειται να τυπωθεί, αυτός θά τυπωθεί στις άκραιές δεξιές θέσεις του πεδίου. Άν όμως σαν w δώσουμε ένα αριθμό μικρότερο από τό πλήθος των ψηφίων του αριθμού, τότε θά έκτυπωθούν άστερίσκοι σέ όλο τό μήκος του πεδίου. Έτσι δίνεται προειδοποίηση ότι πρέπει νά αύξη-σμε τήν τιμή του w στον κωδικό l γιά νά έχομε σωστή έκτύπωση.

Παραδείγματα έκτυπώσεως μέ χρήση του κωδικού lw (Πίνακας 10.6.2).

ΠΙΝΑΚΑΣ 10.6.2.

Παραδείγματα έκτυπώσεως μέ χρήση του κωδικού lw

Αριθμός μέσα στή μνήμη	Κωδικός l	Θά τυπωθεί				
546	13	<table><tr><td>5</td><td>4</td><td>6</td></tr></table>	5	4	6	
5	4	6				
28	14	<table><tr><td></td><td></td><td>2</td><td>8</td></tr></table>			2	8
		2	8			
4281	14	<table><tr><td>4</td><td>2</td><td>8</td><td>1</td></tr></table>	4	2	8	1
4	2	8	1			
5046	13	<table><tr><td>*</td><td>*</td><td>*</td></tr></table>	*	*	*	
*	*	*				
892	15	<table><tr><td></td><td>8</td><td>9</td><td>2</td></tr></table>		8	9	2
	8	9	2			
4	13	<table><tr><td></td><td></td><td>4</td></tr></table>			4	
		4				

Κωδικός wX .

Με τόν κωδικό αυτό περιγράφομε τίς κενές στήλες πού παρεμβάλλονται μεταξύ δύο πεδίων στή λογική ένότητα είσόδου ή έξόδου. w είναι ένας αριθμός, πού παρι-στάνει τό πλήθος των κενών στηλών.

Παραδείγματα αναγνώσεως με χρήση του κωδικού wX.

ΔΕΛΤΙΑ ΔΕΔΟΜΕΝΩΝ	FORMAT ΑΝΑΓΝΩΣΕΩΣ
12 146 5 14 1572 25	FORMAT (I4, 3X, I6) FORMAT (I2, 1X, I3, 1X, I6, 2X, I2)

Παραδείγματα έκτυπώσεως με χρήση του κωδικού wX (Πίνακας 10.16.3).

ΠΙΝΑΚΑΣ 10.16.3.

Παραδείγματα έκτυπώσεως με χρήση του κωδικού wX

Αριθμοί στη μνήμη	FORMAT έκτυπώσεως	Θά τυπωθούν
24 285	I5, 3X, I6	24 285
7213 49	I4, 4X, I2	7213 49
9 26 105	I2, 3X, I4, 1X, I2	9 26 105
9846 0 17	I5, 2X, I1, I4	9846 0 17

Κωδικός Fw.d.

Χρησιμοποιείται για την ανάγνωση ή έκτυπωση των περιεχομένων αριθμητικών πραγματικών πεδίων (Real). Τα περιεχόμενα βρίσκονται σε απλή κλασματική μορφή. Στόν κωδικό αυτό, w είναι ένας αριθμός, που παριστάνει το μέγεθος του πεδίου, ενώ d είναι το πλήθος των κλασματικών ψηφίων του πεδίου.

α) Για την ανάγνωση.

“Αν ο καταχωριζόμενος αριθμός έχει λιγότερα ψηφία από το μέγεθος του πεδίου, τότε τοποθετείται στις άκρατες δεξιές θέσεις.

σες θέσεις, ώστε να βρεθεί μπροστά από το πρώτο μη μηδενικό σημαντικό¹ ψηφίο του αριθμού. Το πλήθος των θέσεων που μετακινήθηκε ή υποδιαστολή, αποτελεί τον εκθέτη. Ο αριθμός που προκύπτει, στρογγυλεύεται σε τόσα κλασματικά ψηφία, όσα δείχνει ο d.

Παραδείγματα έκτυπωσης με χρήση του κωδικού Ew.d (Πίνακας 10.16.5).

ΠΙΝΑΚΑΣ 10.16.5.

Παραδείγματα έκτυπωσης με χρήση του κωδικού Ew.d

Αριθμός μέσα στη μνήμη	FORMAT έκτυπωσης	Θά έκτυπωθεί
1.756	E15.7	0 . 1 7 5 6 0 0 0 E + 0 1
-0.0094	E13.4	- 0 . 9 4 0 0 E - 0 2
175.62	E11.3	0 . 1 7 6 E + 0 3

Κωδικός wH.

Χρησιμοποιείται κυρίως για έκτύπωση μηνυμάτων και χαρακτήρων, που δεν περιέχονται σε καμιά θέση της μνήμης. Ο w είναι ένας αριθμός που παριστάνει το πλήθος των χαρακτήρων, που αποτελούν το μήνυμα που θά τυπώσουμε. Π.χ, αν μέσα σε ένα FORMAT γράψουμε:

11H THE ROOT IS

τότε στον έκτυπωτή θά γραφεί:

| T | H | E | | R | O | O | T | | I | S |

Τό w στην περίπτωση αυτή ίσοιται με 11. Στον αριθμό αυτό συνυπολογίσαμε και τα κενά που παρεμβάλαμε ανάμεσα στις λέξεις του μηνύματος.

10.17 Έπαναλαμβανόμενοι κωδικοί FORMAT.

Συχνά, μέσα σε ένα FORMAT, ο ίδιος κωδικός εμφανίζεται πολλές φορές. Π.χ.:

I2, I2, I2, F3.1, 2X, I3, I3

Εδώ, ο κωδικός I2 εμφανίζεται για τρεις συνεχείς φορές και ο I3 δύο. Μπορούμε, λοιπόν, να συντομεύσουμε τη γραφή του παραπάνω FORMAT, ως εξής:

3I2, F3.1, 2X, 2I3

Αν όμως είχαμε τό FORMAT

I2, F3.1, I2, 2X, I3

¹ Στο δεκαδικό σύστημα οποιοδήποτε από τα ψηφία 1,2,3,... 9 είναι σημαντικό. Τό 0 είναι σημαντικό μόνο στις περιπτώσεις όπου βρίσκεται ανάμεσα σε δύο άλλα σημαντικά ψηφία, ή στο τέλος του αριθμού.

δέν θά μπορούσαμε νά κάνομε οποιαδήποτε συντόμευση, γιατί οί ὅμοιοι κωδικοί 12 δέν βρίσκονται ὁ ἕνας δίπλα στόν ἄλλο.

10.18 Δηλωτικές ἐντολές.

Οί ἐντολές αὐτές δέν εἶναι ἐκτελέσιμες, γράφονται δέ πάντα στήν ἀρχή τοῦ προγράμματος, πρίν ἀπό τήν πρώτη ἐκτελέσιμη ἐντολή. Οἱ πιά συνηθισμένες εἶναι:

α) Ἐντολή **INTEGER**.

Ἡ ἐντολή αὕτη ἔχει τή μορφή:

|| **INTEGER** μεταβλητή, μεταβλητή,...

Ὅλες οἱ μεταβλητές, πού δηλώνονται στήν **INTEGER**, εἶναι πραγματικές. Μέ αὐτήν μπορούμε νά χρησιμοποιοῦμε πραγματικές μεταβλητές σέ ἕνα πρόγραμμα σάν νά ἦταν ἀκέραιες. Π.χ.:

INTEGER X, A12, B

Οἱ μεταβλητές X, A12, B μετά τή δήλωση αὕτη, χρησιμοποιούμενες στό πρόγραμμα, θεωροῦνται σάν ἀκέραιες.

β) Ἐντολή **REAL**.

Εἶναι ἀντίστροφη τῆς προηγούμενης. Ἡ μορφή της εἶναι:

|| **REAL** μεταβλητή, μεταβλητή,...

Ὅλες οἱ μεταβλητές πού δηλώνονται εἶναι ἀκέραιες, θά τίς χρησιμοποιήσουμε δέ στό πρόγραμμά μας, σάν νά ἦταν πραγματικές. Π.χ.:

REAL J, KM, LAB

γ) Ἐντολή **DIMENSION**.

Τήν ἀναφέρομε μόνο ὅταν στό πρόγραμμα χρησιμοποιοῦνται πίνακες. Ἡ μορφή της εἶναι:

|| **DIMENSION** μεταβλητή (ἀριθμητικοί δείκτες), μεταβλητή (ἀριθμητικοί δείκτες),...

Οἱ ἀριθμητικοί δείκτες εἶναι ὀριακοί ἀριθμοί, πού ἐκφράζουν τό μέγεθος (διαστάσεις) τῶν πινάκων. Π.χ.:

DIMENSION A(20), B(5,6), C(4,4)

Μέ τήν ἐντολή αὕτη δηλώνεται ὁ χώρος, πού ἐπιθυμοῦμε νά κρατηθεῖ στή μνήμη, γιά νά καταχωρισθοῦν οἱ τιμές τῶν στοιχείων γιά τόν καθένα ἀπό τοὺς πίνακες. Στό παράδειγμά μας, θά κρατηθοῦν 20 θέσεις γιά τὰ στοιχεῖα τοῦ πίνακα A, 30 (5 × 6) θέσεις γιά τὰ στοιχεῖα τοῦ B καί 16 (4 × 4) θέσεις γιά τὰ στοιχεῖα τοῦ C.

10.19 Μερικές ακόμη εντολές.

α) Έντολή STOP.

Συνήθως είναι η προτελευταία εντολή, με την οποία και σταματά η εκτέλεση ενός FORTRAN προγράμματος.

β) Έντολή END.

Είναι πάντα η τελευταία εντολή ενός FORTRAN προγράμματος. Με αυτήν διακόπτεται έντελως η εκτέλεση και επιστρέφει ο έλεγχος από το μεταφραστικό πρόγραμμα στο έποπτευον σύστημα¹.

10.20 Ειδικές μορφές των εντολών READ και WRITE

Αναφέρονται στίς περιπτώσεις αναγνώσεως ή έγγραφης όλων των τιμών των στοιχείων ενός πίνακα, που έχουν καταχωρισθεί ή πρόκειται να καταχωρισθούν, με συνεχή τρόπο σε μία ή περισσότερες λογικές ενότητες. Π.χ.:

```
4 || READ (5,4) (A(I), I = 1,19)
  || FORMAT (10F5.2)
```

Με την εντολή αυτή διαβάζονται όλες οι τιμές των στοιχείων του πίνακα A την ίδια στιγμή. Αυτές έχουν καταχωρισθεί σε ένα μόνο δελτίο, με κωδικό F5.2 ή κάθε μία. Η αντίστοιχηση των αριθμών που διαβάζονται και των στοιχείων είναι τέτοια, ώστε ο πρώτος αριθμός να καταχωρισθεί σάν περιεχόμενο του A(1), ο δεύτερος σάν περιεχόμενο του A(2) κ.ο.κ.

Τό ίδιο άκριβως συμβαίνει και με την εντολή WRITE. Π.χ.:

```
|| WRITE (6,2) (B(I), I = 1,12)
2 || FORMAT (8(F6.1, 2X))
```

Στό παράδειγμα αυτό γράφονται στον έκτυπωτή τα περιεχόμενα 12 στοιχείων του πίνακα A, με συνεχή τρόπο σε δυό γραμμές, όπου η πρώτη περιλαμβάνει τα 8 πρώτα και η δεύτερη τα υπόλοιπα 4. Μεταξύ δύο στοιχείων αφήνονται δύο κενές στήλες.

10.21 Σχόλια (Comments).

Όταν θέλουμε να γράψουμε σχόλια σε ένα πρόγραμμα, συμπληρώνουμε στή στήλη μιās γραμμής του έντυπου κωδικογραφήσεως τό χαρακτήρα C, γιά να μή θεωρηθούν από τόν υπολογιστή σάν εντολές. Από τήν τρίτη στήλη και έπειτα, μπορούμε να γράψουμε μέ λατινικούς χαρακτήρες ό,τιδήποτε νομίζουμε ότι μπορεί να βοηθήσει στήν κατανόηση του προγράμματος από κάποιον τρίτο, που θά θελήσει

¹ Γιά τίς έννοιες αυτές θά μιλήσουμε στήν παράγραφο 15.3.

νά τό διαβάσει, ή καί από τόν ίδιο τόν προγραμματιστή, όταν θελήσει νά ξαναδεί τό πρόγραμμά του μετά από άρκετό καιρό.

Γραμμές μέ σχόλια μπορούμε νά γράψομε όσες θέλομε καί σέ όποιοδήποτε σημείο του προγράμματος. Π.χ.:

1	5
C	TO PROGRAMMA YPOLOGIZEI
C	TIS RIZES MIAS EXISOSEOS
C	DEFTEROU BATHMOU

10.22 Ύποπρογράμματα (Subprograms).

“Όπως είπαμε καί στην περιγραφή της BASIC, σάν ύποπρόγραμμα χαρακτηρίζομε ένα μικρό άνεξάρτητο βοηθητικό πρόγραμμα, πού μπορεί νά χρησιμοποιείται σέ κάποιο σημείο του κυρίως προγράμματος (Main program), γιά νά έκτελεϊ όρισμένους συγκεκριμένους ύπολογισμούς.

Διακρίνομε δύο κατηγορίες ύποπρογραμμάτων:

α) Ύπορρουτίνες (Subroutines).

β) Συναρτήσεις (Functions).

Θά δούμε στά έπόμενα τόν τρόπο κατάρτισεως τέτοιων ύποπρογραμμάτων.

α) Ύπορρουτίνες (Subroutines).

Μία ύπορρουτίνα έχει τήν έξής μορφή:

SUBROUTINE κωδική όνομασία ύπορρουτίνας (παράμετροι)	
.....	}
.....	
.....	
.....	
RETURN	
END	

Έντολές τής ύπορρουτίνας

Οι παράμετροι είναι μεταβλητές πού αναφέρονται στις έντολές του ύποπρογράμματος καί όνομάζονται τυπικές παράμετροι (Formal parameters).

Μέσω τών παραμέτρων αυτών εισάγονται οι άπαραίτητες τιμές γιά νά λειτουργήσει τό ύποπρόγραμμα καί, έπίσης, αποδίδονται στό κυρίως πρόγραμμα τά άποτελέσματα τών ύπολογισμών.

Η κωδική όνομασία τής ύπορρουτίνας ύπόκειται στους περιορισμούς γραφής, πού ίσχύουν καί στις άπλές μεταβλητές.

“Όλες οι ύπορρουτίνες τελειώνουν μέ τίσ έντολές RETURN καί END. Μέσα στην ύπορρουτίνα μπορούμε νά χρησιμοποιήσομε όλες τίσ έντολές τής FORTRAN.

Παράδειγμα 1.

Νά γραφεί ύπορρουτίνα πού νά ύπολογίζει τό έμβαδόν ενός τριγώνου, όταν είναι γνωστά τά μήκη τών πλευρών του.

Ἡ ὑπορουτίνα θά ἔχει τή μορφή τοῦ σχήματος 10.22α.

Γιά νά λειτουργήσει ἡ ὑπορουτίνα, πρέπει νά τῆς δοθοῦν οἱ τιμές τῶν πλευρῶν A, B, C. Αὐτό γίνεται μέ τήν εἰσαγωγή τῶν τιμῶν ἀπό τό κυρίως πρόγραμμα μέσω τῶν τυπικῶν παραμέτρων, πού συμβολίζουν τά μήκη τῶν ἀντιστοιχῶν πλευρῶν. Τό ἀποτέλεσμα ἀποδίδεται στό κυρίως πρόγραμμα μέσω τῆς παραμέτρου E.

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ						
ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	1	5	6	7	ΕΝΤΟΛΗ FORTRAN	
					S ₁ U ₁ B ₁ R ₁ O ₁ U ₁ T ₁ I ₁ N ₁ E ₁ A ₁ R ₁ E ₁ A ₁ (₁ A ₁ , ₁ B ₁ , ₁ C ₁ , ₁ E ₁)	
					C ₁	
					C ₁ T ₁ H S ₁ S ₁ U ₁ B ₁ R ₁ O ₁ U ₁ T ₁ I ₁ N ₁ E ₁ C ₁ A ₁ L ₁ C ₁ U ₁ L ₁ A ₁ T ₁ E ₁ S ₁	
					C ₁ T ₁ H E A ₁ R ₁ E ₁ A ₁ O ₁ F A ₁ T ₁ R ₁ I ₁ A ₁ N ₁ G ₁ L ₁ E ₁	
					C ₁	
					T ₁ =(₁ A ₁ + ₁ B ₁ + ₁ C ₁) ₁ / ₁ 2* ₁	
					R ₁ = ₁ T ₁ * ₁ (₁ T ₁ - ₁ A ₁) ₁ * ₁ (₁ T ₁ - ₁ B ₁) ₁ * ₁ (₁ T ₁ - ₁ C ₁) ₁	
					E ₁ = ₁ S ₁ Q ₁ R ₁ T ₁ (₁ R ₁) ₁	
					R ₁ E ₁ T ₁ U ₁ R ₁ N ₁	
					E ₁ N ₁ D ₁	

Σχ. 10.22α.

Παράδειγμα 2.

Νά γραφεί ὑπορουτίνα πού νά ὑπολογίζει τό $N! = 1 \times 2 \times 3 \dots (N-1) \times N$.

Ἡ μορφή τῆς θά εἶναι (σχ. 10.22β):

Ὁ ἀριθμός τοῦ ὁποῖου ζητεῖται ὁ ὑπολογισμός τοῦ παραγοντικοῦ, εἰσάγεται ἀπό τό κυρίως πρόγραμμα στήν ὑπορουτίνα μέσω τῆς τυπικῆς παραμέτρου N, ἐνῶ τό ἀποτέλεσμα πού καταχωρίζεται στήν K, ἀποδίδεται μέσω αὐτῆς στό κυρίως πρόγραμμα.

β) Ἐντολή κλήσεως μιᾶς ὑπορουτίνας.

Μία ὑπορουτίνα καλεῖται στό κυρίως πρόγραμμα μέ τήν ἐντολή CALL πού ἔχει τή μορφή:

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ			
ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΕΝΤΟΛΗ FORTRAN
1	5	6	7
			SUBROUTINE
			NIPARAG(N, K)
C1			
C1	THIS	S1	SUBROUTINE
			CALCULATES
C1	THE		N1
			FACTORIAL
C1			
			K1=1
			D0
			10 I=2, N
	10		K1=K1*I
			RETURN
			END

Σχ. 10.22β.

CALL κωδική ονομασία υπορουτίνας (παράμετροι)

Η κωδική ονομασία είναι ίδια με την ονομασία της SUBROUTINE. Οι παράμετροι που εμφανίζονται μέσα στην παρένθεση ονομάζονται ενεργές παράμετροι (Actual Parameters). Οι παράμετροι αυτές αντιστοιχούν μία προς μία με τις τυπικές παραμέτρους και ως προς τό πληθος και ως προς τόν τύπο (πραγματικές ή ακέραιες). Πχ. γιά νά καλέσουμε τήν υπορουτίνα του 1ου παραδείγματος σέ κάποιο σημείο ενός προγράμματος, θά χρησιμοποιήσουμε τήν έντολή:

CALL AREA (A1, B1, C1, E1)

όπου οι παράμετροι A1, B1, C1 είναι μεταβλητές του κυρίως προγράμματος, που θά μεταβιβάσουν τίς τιμές τους στίς παραμέτρους A, B, C τής υπορουτίνας, γιά νά μπορέσει τέ τελευταία νά λειτουργήσει. Τό αποτέλεσμα από τόν ύπολογισμό θά έπιστρέψει, μέσω τής παραμέτρου E τής υπορουτίνας, στό κυρίως πρόγραμμα, σάν τιμή τής E1.

γ) Συναρτήσεις (Functions).

Διακρίνουμε δύο κατηγορίες συναρτήσεων:

α) Συναρτήσεις σε μορφή έντολης (Function statements).

β) Συναρτήσεις υποπρογράμματα (Function subprograms).

Οι πρώτες είναι σχεδόν όμοιες με τις αντίστοιχες συναρτήσεις που περιγράφουμε στη BASIC¹ [παράγραφος 9.15 (α)].

Θά ασχοληθούμε με τις συναρτήσεις της δεύτερης κατηγορίας που μοιάζουν αρκετά στη δομή με τις υπορουτίνες.

Η μορφή τους είναι:

```

FUNCTION κωδική_όνομασία_της_συναρτήσεως (παράμετροι)
.....
.....
.....
.....
RETURN
END
  
```

} Έντολές της συναρτήσεως

Όλες οι μεταβλητές στην παρένθεση, που είναι οι τυπικές παράμετροι, χρησιμοποιούνται για την είσαγωγή των απαραίτητων τιμών, για να λειτουργήσει ή συνάρτηση. Μία συνάρτηση μόνο ένα αποτέλεσμα μπορεί να επιστρέψει στο κυρίως πρόγραμμα (αντίθετα προς μία υπορουτίνα). Η κωδική όνομασία της συναρτήσεως παίζει συγχρόνως και το ρόλο της μεταβλητής, όπου καταχωρίζεται ή τιμή του αποτελέσματος και μέσω της οποίας επιστρέφεται στο κυρίως πρόγραμμα.

Μία συνάρτηση μπορεί να χρησιμοποιείται στο κυρίως πρόγραμμα χωρίς ιδιαίτερη έντολη κλήσεως, αλλά απλώς αναφερόμενη με την όνομασία της, σάν απλός όρος μέσα σε μία έκφραση FORTRAN, όπως ακριβώς γίνεται και με τις συναρτήσεις του συστήματος, που είδαμε στην παράγραφο 10.4.

Παράδειγμα 1.

Νά γραφεί συνάρτηση (υποπρόγραμμα) για τον υπολογισμό του έμβαδου ενός τριγώνου, όταν είναι γνωστά τα μήκη των πλευρών του.

Όπως βλέπομε (σχ. 10.22γ), το ίδιο το όνομα της συναρτήσεως (AREA) χρησιμοποιείται σάν απλή μεταβλητή για την καταχώρηση του τελικού αποτελέσματος του υπολογισμού.

Παράδειγμα 2.

Νά γραφεί συνάρτηση για τον υπολογισμό της τιμής του $N!$ (N παραγοντικό) (σχ. 10.22δ).

Τό αποτέλεσμα του υπολογισμού καταχωρίζεται στην ίδια την όνομασία της συναρτήσεως (NPARAG), που χρησιμοποιείται σάν απλή μεταβλητή μέσα στη συνάρτηση και μέσω της οποίας επιστρέφεται ή υπολογισθείσα τιμή στο κυρίως πρόγραμμα.

¹ Αντί των λέξεων DEF και FN αναφέρεται απλώς ή όνομασία της συναρτήσεως. Π.χ.:

$$F(X,Y) = X*Y - X**2 + Y**2$$

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ		
ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝ.	ΕΝΤΟΛΗ FORTRAN
1	5 6 7	
		F U N C T I O N I A R E A (A , B , C)
		C T H I S F U N C T I O N I C A L C U L A T E S
		C T H E A R E A O F A T R I A N G L E
		$T = (A + B + C) / 2$
		$R = T * (T - A) * (T - B) * (T - C)$
		$A R E A = S Q R T (R)$
		R E T U R N
		E N D

Σχ. 10.22γ.

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ		
ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝ.	ΕΝΤΟΛΗ FORTRAN
1	5 6 7	
		F U N C T I O N I N P A R A G (N)
		C T H I S F U N C T I O N I C A L C U L A T E S
		C T H E N F A C T O R I A L
		$N P A R A G = 1$
		D O 1 0 I = 2 , N
		1 0 N P A R A G = N P A R A G * I
		R E T U R N
		E N D

Σχ. 10.22δ.

10.23 Έντυπο ανάγραφής των δεδομένων (Data form).

Στό έντυπο αυτό (σχ. 10.23) αναγράφουμε τὰ δεδομένα (data) σύμφωνα με τὶς ἐντολές FORMAT πού ἔχουν δοθεῖ στό πρόγραμμα, πρὶν διατρηθοῦν σέ δελτία.

Τό έντυπο περιλαμβάνει ὀρισμένο ἀριθμό γραμμῶν πού κάθε μία χωρίζεται σέ 80 στήλες.

Τό περιεχόμενο μιᾶς γραμμῆς ἀντιπροσωπεύει τό περιεχόμενο ἐνός δελτίου δεδομένων.

Οἱ γραμμές τοῦ ἐντύπου δεδομένων συμπληρώνονται μέ τήν ἴδια σειρά, μέ τήν ὁποία καί θά τοποθετηθοῦν τὰ δελτία στήν ομάδα δελτίων ἐντολῆς τοῦ προγράμματος.

10.24 Ἐφαρμογές.

Θά δοῦμε παρακάτω μερικές πλήρεις ἐφαρμογές.

Παράδειγμα 1.

Νά ὑπολογισθεῖ ἡ τιμή τῆς συναρτήσεως $\psi = x^2 + 4x - 5$ γιά τιμές τοῦ $x = 0.2$ (0.2) 4.

Τό πρόγραμμα θά ἔχει τήν ἐξῆς μορφή (σχ. 10.24α).

Παράδειγμα 2.

Νά βρεθεῖ τό μικρότερο ἀπό τὰ στοιχεῖα ἐνός μονοδιάστατου πίνακα A. Ὁ πίνακας ἀποτελεῖται ἀπό 20 στοιχεῖα πού ἔχουν διατρηθεῖ σέ δελτία μέ κωδικό FORMAT (16F5.1) (σχ. 10.24β).

Τό πρόγραμμα θά ἔχει ὡς ἐξῆς (σχ. 10.24γ).

Παράδειγμα 3.

Νά ὑπολογισθεῖ ἡ τιμή τῆς παραστάσεως $\sum_{v=1}^{10} a_v x^v$ Θεωρεῖστε ὅτι οἱ

συντελεστές a_v ἀποτελοῦν στοιχεῖα ἐνός μονοδιάστατου πίνακα καί ὅτι οἱ τιμές τους ἔχουν διατρηθεῖ σέ ἕνα δελτίο μέ FORMAT (10F6.1). Ἡ τιμή τοῦ x διαβάζεται ἀπό ἕνα δεῦτερο δελτίο (σχ. 10.24δ).

Τό πρόγραμμα ἔχει ὡς ἐξῆς (σχ. 10.24ε).

Παράδειγμα 4.

Νά ὑπολογισθοῦν οἱ τιμές τῆς συναρτήσεως

$$\psi = \begin{cases} x^3 - 2x & \text{ἂν } x > 1 \\ x^2 - 1 & \text{ἂν } x < 1 \\ x & \text{ἂν } x = 1 \end{cases}$$

γιά 50 τιμές τοῦ x διατρημένες ἡ κάθε μία σέ ξεχωριστό δελτίο, μέ FORMAT (F4.2) (σχ. 10.24στ).

Τό πρόγραμμα θά ἔχει τήν ἐξῆς μορφή (σχ. 10.24ζ).

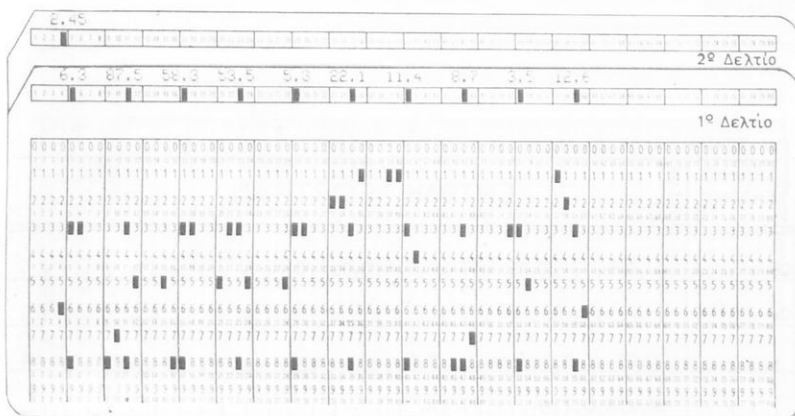
[illegible] $\Sigma\chi$. 10.24a.

[illegible]

Σχ. 10.24β.

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ	
ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝ. ΕΝΤΟΛΗ FORTRAN
5	
6	
7	
	C I M I N I M U M I E L E M E N T O F A N A R R A Y
	C
	D I M E N S I O N I A (2 , 0)
	R E A D (5 , 1) (A (I) , I = 1 , 2 0)
1	F O R M A T (1 6 F 5 . 1)
	S M I N = A (1)
	D O 1 0 I = 2 , 2 0
	I F (A (I) * L E * S M I N) S M I N = A (I)
1 0	C O N T I N U E
	W R I T E (6 , 2) S M I N
2	F O R M A T (F 6 . 1)
	S T O P
	E N D

 $\Sigma\chi. 10.24\gamma.$



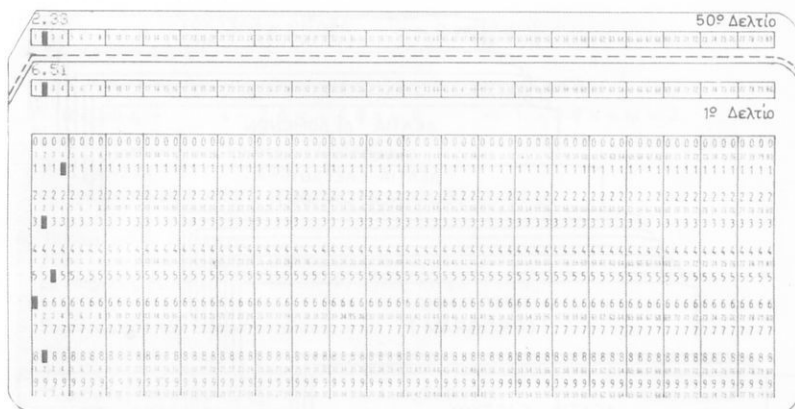
Σχ. 10.24δ.

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ			
ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝ.	ΕΝΤΟΛΗ FORTRAN	
1	5 6 7	C I C I A L C U I L A T I O N I O F A I S U M	
		C I	
		D I M E N S I O N A I (1 0)	
		R E A D (5 , 1) (A (I) , I = 1 , 1 0)	
	1	F O R M A T (1 0 F 6 . 1)	
		R E A D (5 , 2) X	
	2	F O R M A T (F 6 . 2)	
		S I = 0 .	
		D O I 1 0 I = 1 , 1 0	
	1 0	S I = S I + A (I) * X * I	
		W R I T E (6 , 3) S	
	3	F O R M A T (F 9 . 3)	
		S T O P	
		E N D	

Σχ. 10.24ε.

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ			
ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝ.	ΕΝΤΟΛΗ FORTRAN	
1	5	6	7
		C	V A L U E S , O F A F U N C T I O N ,
		C	
			D O , 1 5 , I = 1 , 5 , 0
			R E A D , (5 , 1) , X
	1		F O R M A T (F 4 . 2)
			I F (X - 1 . 0) , 2 0 , 1 0 , 3 0
	2 0		Y = X * * 2 - 1 . 0
			G O , T O , 4 0
	1 0		Y = X
			G O , T O , 4 0
	3 0		Y = X * * 3 - 2 . 0 * X
	4 0		W R I T E , (6 , 2) , Y
	2		F O R M A T (F 6 . 2)
	1 5		C O N T I N U E
			S T O P
			E N D

Σχ. 10.24στ.



Σχ. 10.24ζ.

10.25 Διαδικασία προετοιμασίας ενός προγράμματος FORTRAN.

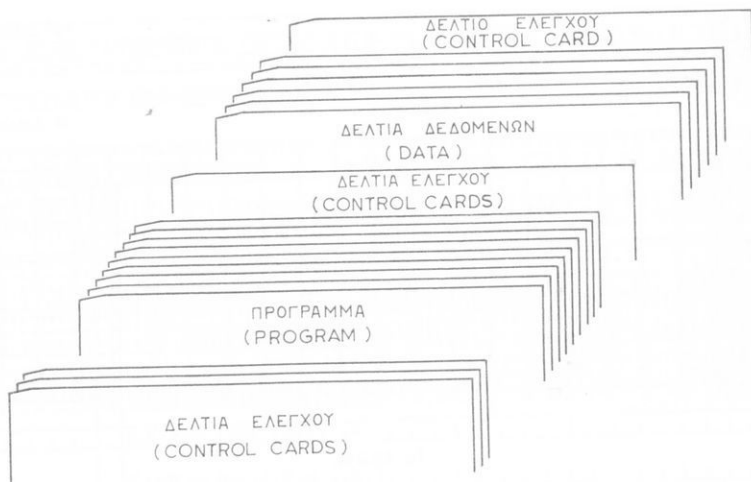
Ἄφου σχεδιασθεῖ τό διάγραμμα ροῆς, στή συνέχεια γίνεται ἡ κωδικοποίηση τοῦ προγράμματος γράφοντας τίς ἐντολές στό εἰδικό ἔντυπο κωδικογραφήσεως τῆς FORTRAN (σχ. 10.9). Οἱ ἐντολές διατρυπῶνται κατόπιν σέ δελτία, ἔτσι ὥστε σέ κάθε δελτίο νά ἀντιστοιχεῖ τό περιεχόμενο μιᾶς γραμμῆς τοῦ ἐντύπου. Στή συνέχεια διατρυπῶνται καί τά δεδομένα τοῦ προγράμματος, τά ὁποῖα ἔχουν γραφεῖ σύμφωνα μέ τούς κωδικούς τῶν ἐντολῶν FORMAT στά εἰδικά ἔντυπα δεδομένων (DATA FORMS, σχῆμα 10.23). Προστίθενται ὀρισμένα δελτία ἐλέγχου¹ (Control Cards) καί σχηματίζεται ἔτσι ἡ ὁμάδα τῶν δελτίων τῶν ἐντολῶν τοῦ προγράμματος καί τῶν δεδομένων, ἔτοιμο γιά νά εἰσαχθεῖ καί νά ἐκτελεσθεῖ στόν ὑπολογιστή (σχ. 10.25).

10.26 Άσκησης.

1. Χαρακτηρίστε με τούς όρους άκέραιη σταθερή, άκέραιη μεταβλητή, πραγματική σταθερή, πραγματική μεταβλητή τά έξής:

A12	KL	FRAM	784	-274
-17.4	1.8E + 4	GX84	0.05	AREA
JB(1)	BARG	C (I)	4.E-3	177.05

¹ Ἡ μορφή τῶν δελτίων ἐλέγχου διαφέρει ἀπὸ ὑπολογιστὴ σὲ ὑπολογιστὴ. Διαφέρει ἀκόμη καὶ τὸ πλῆθος τους. Τὰ συμπληρώσαμε κατὰλληλα μὲ τὴ βοήθεια τοῦ υπεύθυνου τοῦ συγκροτήματος. Σὲ κάθε δελτίο ἐλέγχου ἀναγράφονται πληροφορίες πρὸς τὸ σύστημα τοῦ ὑπολογιστῆ.



Σχ. 10.25.

Ἡ ὁμάδα δελτίων τῶν ἐντολῶν καὶ τῶν δεδομένων τοῦ προγράμματος.

2. Ποιές ἀπὸ τὶς παρακάτω ὀνομασίες μεταβλητῶν εἶναι σωστές, ποιές λάθος καὶ γιατί.

AK1
1ETO
B(-2)

F+5
KALY
TRIANGLE

EQUAT
R B 12
FF 84

MK(A)
L2 (4.2)
SUM (I+J)

EDR(0,8)
NAMIN (K, P)
ELPOS(20)

3. Νά κωδικοποιηθοῦν μέ τή βοήθεια τῆς μερικῆς ἀναμίξεως οἱ παρακάτω μαθηματικές σχέσεις:

$$x^2 + \frac{4}{3}ax + \beta$$

$$2\eta\mu x + 5 \sigma\upsilon\nu x$$

$$\frac{1}{5} \sigma\upsilon\nu x + \eta\mu^2\psi$$

$$\frac{\sqrt{x}}{4} + \frac{1}{3}e^x$$

$$a\beta + 2(\gamma\delta + \epsilon\alpha) \frac{\beta}{\gamma}$$

$$|x| + \frac{2e^{-x}}{\eta\mu x}$$

4. Ἐπίσης οἱ σχέσεις:

$$\frac{3x^2 + 6x + 7}{2x^3 - 3x^2 + 5}$$

$$(\eta\mu x + \sigma\upsilon\nu\psi)^2$$

$$\frac{1}{3} \sqrt{a^2 - 2\beta\gamma + \delta}$$

$$\frac{|a|}{4 + |\beta|} + \frac{6}{7}$$

$$[(a+\beta)^2 - \beta^2\gamma^2]a\beta\gamma$$

$$\frac{e^{2x} + e^{3x}}{5}$$

5. Νά κωδικοποιηθοῦν οἱ ἐπόμενες ἀπλές καὶ σύνθετες μαθηματικές λογικές σχέσεις:

$$\alpha < \beta$$

$$x^2 + 1 > 4 \text{ καὶ } \psi > 0$$

$$x\psi + z = 0 \text{ καὶ } a\beta\gamma = 1$$

$$0 \leq \gamma < \delta \quad \alpha = 0 \text{ ή } \beta > 0 \quad \alpha = 0 \text{ και } \frac{\delta}{\epsilon} \neq \frac{\gamma}{\beta}$$

6. Επίσης οι σχέσεις:

$$\alpha_1 < \alpha_2 < \alpha_3 \quad \gamma = 0 \text{ και } x < 3 \quad \alpha < \frac{1}{3} \text{ και } \beta = \frac{1}{5}$$

$$0 \leq \alpha \beta \leq 4 \quad z = 2 \text{ ή } \psi = 4 \text{ ή } x = 6 \quad x\psi - 1 > 0 \text{ ή } x = \psi = 2.$$

7. Πώς θα διατυρωήσετε τα δεδομένα-αριθμούς;

7.45 -2.0 147 2.E+3

σε ένα δελτίο με FORMAT (F6.2, 2X, F4.1, 1X, I4, 5X, E7.0).

8. Όμοιως για τα δεδομένα:

2284 0.0005 -54 0.074E-02

με κωδικούς FORMAT (2X, I6, 3X, F7.4, 3X, I3, 6X, E10.3).

9. Συμπληρώστε τον παρακάτω πίνακα:

Αριθμός μέσα στη μνήμη	Κωδικός FORMAT	Θά τυπωθεί
-17	I 4	
224	I 2	
3.54	F4.2	
12.448	F6.2	
7.13	F3.1	
1.546	F6.2	

10. Επίσης τον πίνακα:

Αριθμός μέσα στη μνήμη	Κωδικός FORMAT	Θά τυπωθεί
2.007	E10.3	
-15.746	E11.2	
65.37	E 9.2	
18.94	E12.4	
-1050.1	E 9.2	
0.007	E 8.1	
-0.0003	E 9.0	

11. Νά γραφεί πρόγραμμα που νά υπολογίζει και νά τυπώνει τό άθροισμα των αντίστροφων των 20 πρώτων φυσικών αριθμών ($1 + 1/2 + \dots + 1/20$).

12. Νά γραφεί πρόγραμμα που νά υπολογίζει και νά τυπώνει τίς τιμές της συναρτήσεως $y = x \eta \mu x + \sin \psi$ για τιμές του $x = 0.1$ (0.1) 0.7.

13. Δίνεται μονοδιάστατος πίνακας A(I) με 30 στοιχεία. Νά γραφεί πρόγραμμα που νά διαβάσει τά στοιχεία με κωδικούς FORMAT (16F5.1) και στη συνέχεια νά υπολογίζει τό μέσο όρο τους.

14. Δίνονται δύο μονοδιάστατοι πίνακες A (I) και B (I) με 15 στοιχεία ο καθένας. Νά γραφεί πρόγραμμα που νά τους διαβάσει και νά δημιουργεί νέο πίνακα C (I) κάθε στοιχείο του οποίου θά ίσούται με τό άθροισμα των αντίστοιχων στοιχείων των δοθέντων πινάκων.

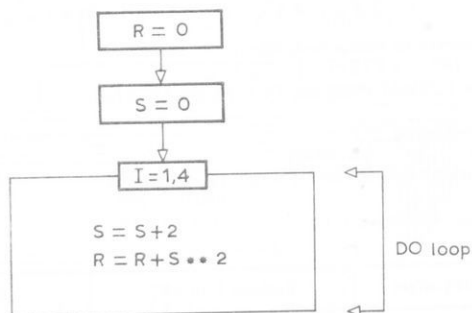
15. Νά γραφεί υπορουτίνα, που νά υπολογίζει τόν όγκο μιάς σφαίρας, όταν είναι γνωστή ή ακτίνα της.

16. Νά γραφεί συνάρτηση (υποπρόγραμμα), που νά υπολογίζει τήν τιμή της συναρτήσεως.

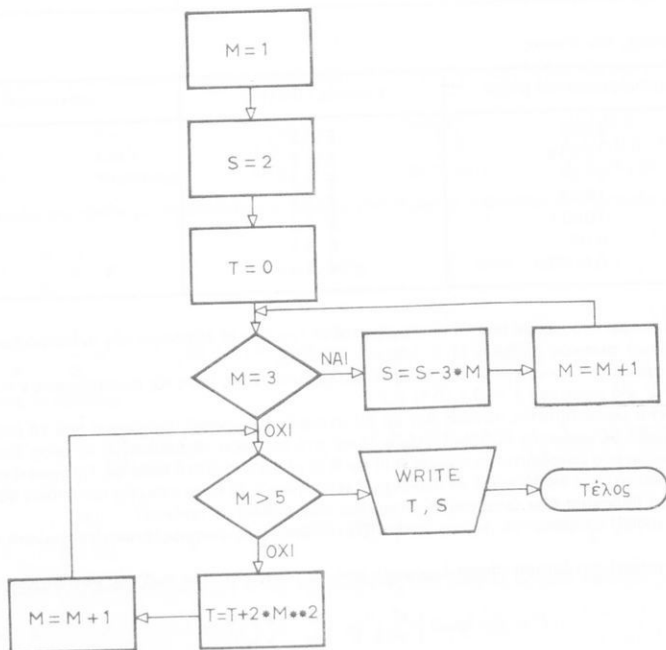
$$\psi = \begin{cases} x^2 & \text{άν } x = 0 \\ x^2 + 1 & \text{άν } x \neq 0 \end{cases}$$



17. Ποιές θα είναι οι τιμές των μεταβλητών S και R μετά την εκτέλεση του παριτιθεμένου τμήματος του προγράμματος;



18. Να γίνει η κωδικοποίηση του παρακάτω διαγράμματος ροής:



ΚΕΦΑΛΑΙΟ ΕΝΔΕΚΑΤΟ

ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΜΗΧΑΝΕΣ ΓΡΑΦΕΙΟΥ - ΑΡΙΘΜΟΜΗΧΑΝΕΣ

11.1 Γενικά.

Σάν υπολογιστικές μηχανές γραφείου έννοοῦμε μικρές σέ μέγεθος μηχανές μέ δυνατότητες κατά πολύ μικρότερες από ἐκείνες ἑνός ἠλεκτρονικοῦ ὑπολογιστή.

Σήμερα, οἱ ἐξελιγμένες μορφές τῶν μηχανῶν αὐτῶν χρησιμοποιοῦνται πάρα πολύ σέ μικροῦ μεγέθους ἐφαρμογές, ὅπου ἡ χρήση τοῦ μεγάλου ἠλεκτρονικοῦ ὑπολογιστή εἶναι ἀσύμφορη. Ἔτσι, οἱ ἀνάγκες ἑνός γραφείου ἢ μιᾶς μικρῆς ἐπιχειρήσεως καλύπτονται ἀπό τίς μηχανές αὐτές.

Ἡ βιομηχανική καί τεχνική ἀνάπτυξη μετὰ τό 2ο Παγκόσμιο πόλεμο δέν ὀδήγησε μόνο στήν τελειότητα τῶν μεγάλων ἠλεκτρονικῶν ὑπολογιστῶν, ἀλλά ἀκόμη ἐπηρέασε θετικά καί τήν ἐξέλιξη τῶν παλαιοῦ τύπου ὑπολογιστικῶν μέσων. Ὁ τρόπος λειτουργίας τους μεταβλήθηκε μέ τήν πάροδο τοῦ χρόνου, ἀκολουθώντας τήν ἐξέλιξη τῆς τεχνικῆς.

Ὅπως καί στοὺς ὑπολογιστές, διακρίνομε 4 γενιές ὑπολογιστικῶν μηχανῶν.

Ἡ πρώτη γενιά, πού καλύπτει τή χρονική περίοδο 1870 - 1920, περιλαμβάνει ὑπολογιστικές συσκευές, ὁ τρόπος λειτουργίας τῶν ὁποίων ἦταν καθαρά μηχανικός. Τά διάφορα τμήματα ἔμπαιναν σέ λειτουργία μέ τή χρήση μοχλῶν ἀπό τό χειριστή καί ἡ κίνηση ἀπό τό ἓνα τμήμα στοῦ ἄλλο μεταδίδονταν μέ ὀδοντωτοὺς τροχοὺς. Οἱ μηχανές αὐτές ἦταν βασικά ἀθροιστικές καί γι' αὐτό χρησιμοποιήθηκαν στήν ἀρχή κυρίως σάν ταμιακές ἢ στήν ἐπεξεργασία στατιστικῶν στοιχείων. Ἀργότερα κατασκευάσθηκαν τύποι, πού μπορούσαν νά ἐκτελοῦν ὄχι μόνο προσθέσεις, ἀλλά καί ἀφαιρέσεις καί πολλαπλασιασμούς.

Ἡ δεύτερη γενιά, πού ἀρχίζει ἀπό τό 1920 καί τελειώνει τό 1955, περιλαμβάνει μηχανές μέ ἠλεκτρομηχανικό τρόπο λειτουργίας. Ἐδῶ τά διάφορα τμήματα κινοῦνται μέ τή βοήθεια τοῦ ἠλεκτρισμοῦ. Ὁ χειριστής πιέζει πλήκτρα καί ἀποκαθιστᾷ τό ρεύμα σέ κατάλληλα κυκλώματα, πού θέτουν σέ κίνηση τά μηχανικά μέρη τῆς συσκευῆς. Μέ τήν ἴδια βασική ἀρχή λειτουργίας κατασκευάσθηκαν διάφοροι τύποι, γιά νά ἐξυπηρετήσουν συγκεκριμένες ἀνάγκες, ὅπως ταμιακές μηχανές, μηχανές 4 ἀριθμητικῶν πράξεων, λογιστικές κ.ο.κ. Ἀκόμη καί σήμερα ἐξελιγμένες τέτοιες μηχανές. Οἱ μηχανές τῆς δεύτερης γενιάς ἀναφέρονται συχνά μέ τό γενικό ὄνομα *κλασσικές ὑπολογιστικές μηχανές*. Τή λειτουργία κάθε τύπου καί τίς ἐργασίες πού ἐκτελεῖ θά δοῦμε πιο κάτω.

Ἡ τρίτη γενιά ἀρχίζει ἀπό τό 1955 καί φθάνει μέχρι τό 1970. Ἡ λειτουργία

τους βασίζεται στις αρχές της ηλεκτρονικής. "Όλα τα μηχανικά μέρη έχουν αντικατασταθεί από ηλεκτρονικά κυκλώματα και η λειτουργία της μηχανής έχει γίνει αθόρυβη και πιδ'άποδοτική. Οι φθορές έχουν μειωθεί και ο χρόνος έπεξεργασίας έχει ελαττωθεί σημαντικά. Οι ηλεκτρονικές μηχανές παρουσιάζονται και αυτές σε διάφορους τύπους, όπως απλές έμπορικές, έπιστημονικές κλπ.

Η εξέλιξη όμως συνεχίζεται και τα έπιτεύγματα της ηλεκτρονικής στην κατασκευή μικροκυκλωμάτων οδήγησαν στο σχεδιασμό υπολογιστικών μηχανών με έξαιρετικά μικρές διαστάσεις και μεγάλες δυνατότητες. "Ήδη διανύουμε την τέταρτη γενιά υπολογιστικών μηχανών.

"Όλες οι μηχανές πού θα περιγραφούν στο κεφάλαιο αυτό φέρουν τό γενικό όνομα Calculators. Η απόδόσή του στά Έλληνικά γίνεται μέ χρήση του διπλού όρου υπολογιστικές μηχανές ή αριθμομηχανές.

11.2 Κλασσικές υπολογιστικές μηχανές.

Στήν κατηγορία αυτή ανήκουν μηχανές διαφόρων τύπων γιά έμπορική κυρίως χρήση. Οι μηχανές αυτές χρησιμοποιήθηκαν επί ολόκληρες δεκαετίες γιά την έκτέλεση διαφόρων λογιστικών έργασιών, τή δημιουργία στατιστικών πινάκων κ.ο.κ.

Παρά τήν πρόσφατη μεγάλη ανάπτυξη των ηλεκτρονικών υπολογιστών, οι μηχανές αυτές χρησιμοποιούνται και σήμερα (σε περιορισμένη όμως κλίμακα) σε πολλές δουλειές ρουτίνας.

Ταξινομούνται στις έξης κατηγορίες:

- α) 'Απλές άθροιστικές μηχανές και μηχανές 4 πράξεων μέ έκτύπωση (Adding machines and printing calculators).
- β) Αυτόματες υπολογιστικές μηχανές μέ πληκτρολόγιο (Key - driven calculators).
- γ) 'Υπολογιστικές μηχανές, μέ κινητό πίνακα άποτελεσμάτων (Rotary calculators).
- δ) Λογιστικές μηχανές (Accounting machines).

α) 'Απλές άθροιστικές και μηχανές 4 πράξεων μέ έκτύπωση.

Οι μηχανές αυτές θα περιγραφούν μαζί, λόγω της ομοιότητας πού παρουσιάζουν στή κατασκευή και στή λειτουργία τους. Και οι δύο διαθέτουν πληκτρολόγιο μέ 10 κύρια πλήκτρα αριθμητικών ψηφίων (Ten key models) και μερικά άλλα βοηθητικά (σχ. 11.2α).

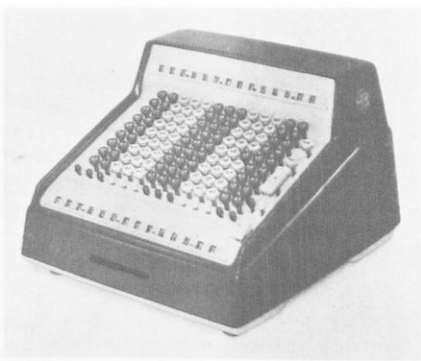
Οι μηχανές αυτές έχουν σχεδιασθεί νά λειτουργούν μέ απλή πίεση ενός πληκτρο (One - hand touch operation). Γιά νά είσασχεϊ ένας αριθμός στή μηχανή, πιέζονται τά πλήκτρα των ψηφίων του μέ τή σειρά, ξεκινώντας από τό πιδό σημαντικό.

Οι άθροιστικές μηχανές κατασκευάζονταν παληότερα και σε τύπους πλήρους πληκτρολογίου (Full - key board), μέ πλήκτρα τοποθετημένα σε κατακόρυφες στήλες τόσες, όσο και τό μέγιστο πλήθος των ψηφίων, πού μπορεί νά δεσχεϊ. Σε κάθε στήλη υπάρχουν πλήκτρα αριθμημένα από τό 1 έως τό 9. Γιά νά είσάγομε ένα αριθμό πιέζομε ένα πληκτρο από κάθε στήλη πού αντίστοιχει στό ψηφίο του αριθμού. Η χωρητικότητα της κάθε μηχανής έξαρτάται από τό πλήθος των στηλών του



Σχ. 11.2α.

Μία μηχανή 4 πράξεων.



Σχ. 11.2β.

Αυτόματη υπολογιστική μηχανή με πλήρες πληκτρολόγιο.

πληκτρολογίου πού διαθέτει. Καί οι δύο τύποι μηχανών τυπώνουν τά αποτελέσματα σέ χαρτοταινία. Επίσης μπορούν νά έκτελουν καί τίς 4 ἀριθμητικές πράξεις. Ἡ κυριότερη διαφορά μεταξύ τους ἀφορᾷ στόν τρόπο έκτελέσεως τοῦ πολλαπλασιασμοῦ. Ἡ ἀθροιστική μηχανή πολλαπλασιάζει έκτελώντας διαδοχικές προσθέσεις, ἐνῶ ἡ μηχανή 4 πράξεων πολλαπλασιάζει ἀπ' εὐθείας. Ὅρισμένοι τύποι μηχανῶν 4 πράξεων ἔχουν πλήκτρο σταθεροῦ συντελεστή, ἐκτυπώνουν κλασματικά ψηφία, κάνουν αὐτόματη ὕψωση στό τετράγωνο καί έκτελουν πολλαπλασιασμούς μέ ἀρνητικούς ἀριθμούς.

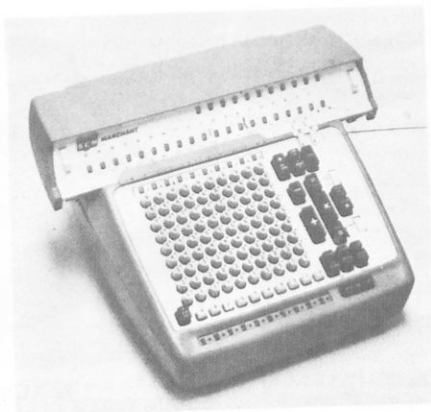
β) Αυτόματες υπολογιστικές μηχανές.

Εἶναι μηχανές πλήρους πληκτρολογίου (σχ. 11.2β) μέ στήλες ἀπό πλήκτρα ἀριθμμένα ἀπό τό 1 ἕως τό 9, ὅμοιες μέ τίς ἀθροιστικές μηχανές πλήρους πληκτρολογίου. Οἱ υπολογισμοί γίνονται αὐτόματα μέ τήν πίεση τῶν καταλλήλων πλήκτρων. Τά αποτελέσματα ἐμφανίζονται συνήθως σέ ἓνα εἰδικό ἀνοίγμα - παράθυρο, στό πάνω ἢ κάτω μέρος τῆς μηχανῆς. Μερικοί τύποι διαθέτουν δύο τέτοια παράθυρα. Στό ἓνα ἐμφανίζονται τά ἐνδιάμεσα αποτελέσματα καί στό ἄλλο τά συγκεντρωτικά. Παρ' ὅλο πού οἱ μηχανές αὐτές μπορούν νά έκτελουν καί τίς 4 ἀριθμητικές πράξεις, εἶναι περισσότερο κατάλληλες γιά γρήγορες προσθέσεις καί πολλαπλασιασμούς (οἱ πολλαπλασιασμοί έκτελοῦνται καί ἐδῶ μέ διαδοχικές προσθέσεις). Γιά τό λόγο αὐτό χρησιμοποιοῦνται κυρίως γιά λογιστικούς ἐλέγχους, καί ταμιακές ἐργασίες.

γ) Ὑπολογιστικές μηχανές μέ κινητό πίνακα ἀποτελεσμάτων.

Παρ' ὅλο πού οἱ μηχανές αὐτές μπορούν νά έκτελουν καί τίς 4 ἀριθμητικές πράξεις, χρησιμοποιοῦνται κυρίως γιά τήν ἐπίλυση προβλημάτων μέ πολύπλοκους πολλαπλασιασμούς καί διαιρέσεις. Οἱ περισσότερες μηχανές τοῦ τύπου αὐτοῦ δέν τυπώνουν σέ ταινία, σέ μερικούς ὅμως τύπους ὑπάρχει ἡ δυνατότητα αὐτή. Ἄλλοι τύποι τέλος ἔχουν πλήρες πληκτρολόγιο (σχ. 11.2γ) καί ἄλλοι δεκαψηφία.

Οἱ περισσότερες μηχανές διαθέτουν ὀρισμένο ἀριθμό καταχωριστῶν (Registers), συνήθως 1 - 5.



Σχ. 11.2γ.

Υπολογιστική μηχανή με κινητό πίνακα αποτελεσμάτων.

Σέ μερικούς τύπους γίνεται ύψωση στο τετράγωνο και τοποθέτηση κλασματικών ψηφίων.

Χρησιμοποιούνται για στατιστική επεξεργασία δεδομένων, όπου χρειάζεται ο υπολογισμός μεγάλου αριθμού ποσοστών και λόγων. Ή ακόμη χρησιμοποιούνται στις εκδόσεις λογαριασμών, τής μισθοδοσίας κλπ.

δ) Λογιστικές μηχανές.

Κάτω από τόν τίτλο αυτό συνωστίζεται μία μεγάλη ποικιλία από διάφορους τύπους μηχανών που χρησιμοποιούνται για εκδόσεις λογαριασμών, έγγραφες στο καθολικό, χρεώσεις, πιστώσεις, ήμερολόγια, έντολές, ίσοζύγια κλπ.

Τίς κατατάσσουμε σέ δύο κατηγορίες:

- α) Μή περιγραφικές (Nondescriptive)
- β) Περιγραφικές (Descriptive)

α) Μή περιγραφικές. Οι μηχανές αυτές διαθέτουν μόνο αριθμητικό πληκτρολόγιο (σχ. 11.2δ) και γι' αυτό δέν μπορούν νά έκτυπώσουν άλφαβητικούς χαρακτήρες. Διαθέτουν περιορισμένο αριθμό προσθέτων πλήκτρων μέ ήμερομηνίες και συντομογραφικά σύμβολα, που χρησιμοποιούνται γιά τόν χαρακτηρισμό διαφόρων μεγεθών ή λειτουργιών.

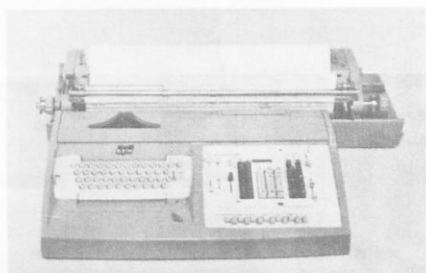
β) Περιγραφικές μηχανές. Οι μηχανές αυτές είναι έφοδιασμένες, έκτός από τό αριθμητικό και μέ άλφαβητικό πληκτρολόγιο γραφομηχανής (σχ. 11.2ε), πράγμα που τίς καθιστά κατάλληλες γιά πολλές λογιστικές δουλειές, όπως τήν έκδοση λογαριασμών, τιμολογίων, έπιταγών κλπ., όπου απαιτείται ή έκτύπωση όνομάτων, περιγραφών και άλλων άλφαβητικών ή άλφαριθμητικών στοιχείων.

Υπάρχουν σέ τύπους δεκαψηφίου και πλήρους αριθμητικού πληκτρολογίου.



Σχ. 11.2δ.

Λογιστική μη περιγραφική μηχανή.



Σχ. 11.2ε.

Περιγραφική λογιστική μηχανή.

11.3 Ήλεκτρονικές υπολογιστικές μηχανές.

Υπάρχουν σέ πολύ μεγάλη ποικιλία μεγεθών, δυνατοτήτων καί χρήσεων. Θά μπορούσαμε νά τίς κατατάξομε:

Ός πρός τίς εφαρμογές πού εξυπηρετοῦν σέ:

- α) Έμπορικές
- β) Έπιστημονικές

Ός πρός τό μέγεθος τους σέ:

- α) Έπιτραπέζιες
- β) Τσέπης

Έμπορικές ηλεκτρονικές υπολογιστικές μηχανές.

Στήν κατηγορία αὕτη ἀνήκουν:

α) Ήλεκτρονικές ταμιακές μηχανές.

Διαθέτουν βασικό δεκαψήφιο ἀριθμητικό πληκτρολόγιο, καθώς καί πληκτρα γιά ἄλλες βοηθητικές ἐργασίες. Διαθέτουν ἐπίσης ἀριθμό καταχωριστῶν-ἀθροιστῶν (1 - 20). Ὑπολογίζουν ποσοστά, φόρους, ἐκπτώσεις, καθώς καί σύνολα πωλήσεων σέ διάφορα ἐπίπεδα (ἀναλυτικές, ἡμερήσιες, ἐβδομαδιαίες κλπ.). Ἐκδίδουν ἀποδείξεις, καί τυπώνουν σέ ταινία. Συγχρόνως ἐμφανίζουν τά διάφορα ποσά σέ ὀθό-

νη με αμφίπλευρη ορατότητα. Η χωρητικότητα της οθόνης κυμαίνεται από 8 - 12 ψηφία με δυνατότητα παρουσιάσεως και 2 κλασματικών ψηφίων (σχ. 11.3α).



Σχ. 11.3α.

Ήλεκτρονική ταμιακή μηχανή.

β) Ήλεκτρονικές αριθμομηχανές 4 πράξεων.

Τίς συναντάμε σε τύπους έπιτραπέζιους (Desk calculators) και τσέπης (Pocket calculators).

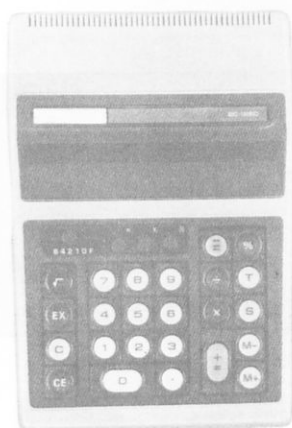
Διαθέτουν βασικό δεκαψήφιο αριθμητικό πληκτρολόγιο. Εκτός από τις 4 αριθμητικές πράξεις, έχουν δυνατότητα για σταθερό συντελεστή, ποσοστά, καθορισμό πλήθους κλασματικών ψηφίων κλπ.

Διαθέτουν επίσης 1 - 3 καταχωριστές δεδομένων καθώς και καταχωριστές εισόδου¹ και αποτελέσματος².

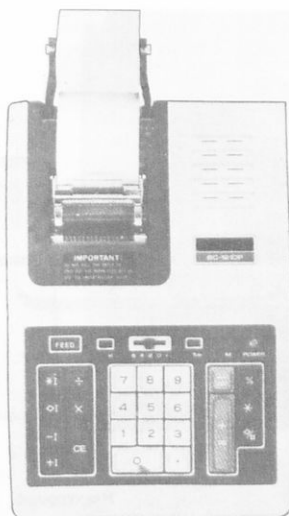
Σε όρισμένους τύπους, οι πληκτρολογούμενοι αριθμοί και τα αποτελέσματα εμφανίζονται σε οθόνη (μέ λυχνίες ή υγρούς κρυστάλλους) με χωρητικότητα 8 έως 14 ψηφίων [σχ. 11.3β(α)], ενώ σε άλλους εκτυπώνονται μόνο σε χαρτί [σχ. 11.3β(β)] και σε άλλους έχουμε ταυτόχρονη εκτύπωση και απεικόνιση σε οθόνη [σχ. 11.3β(γ)].

1 Καταχωριστής εισόδου (entry register) είναι ένας απλός καταχωριστής όπου αποθηκεύεται ο αριθμός που πληκτρολογείται κάθε φορά.

2 Καταχωριστής αποτελέσματος (number operator register) είναι ένας απλός καταχωριστής όπου αποθηκεύεται το αποτέλεσμα από την εκτέλεση μιάς πράξεως. Ο καταχωριστής αυτός είναι αντίστοιχος προς τον καταχωριστή - συσσωρευτή (accumulator) των μεγάλων υπολογιστών.



α)



β)

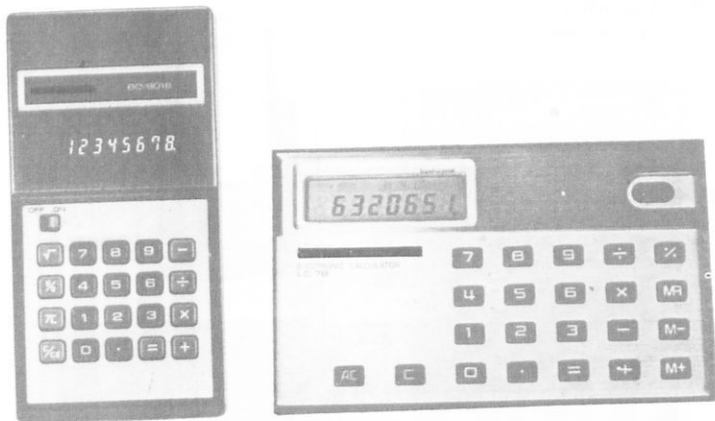


γ)

Σχ. 11.3β.

α) Ήπιτραπέζια ήλεκτρονική αριθμομηχανή με όθονη. β) Ήπιτραπέζια ήλεκτρονική αριθμομηχανή με έκτύπωση. γ) Ήπιτραπέζια ήλεκτρονική αριθμομηχανή με έκτύπωση καί όθονη.

Οί τύποι αριθμομηχανής τσέπης έχουν σχεδόν τά ίδια χαρακτηριστικά με τούς ήπιτραπέζίους, αλλά διαθέτουν μόνο όθονη (σχ. 11.3γ).



Σχ. 11.3γ.

Ήλεκτρονικές αριθμομηχανές τσέπης.

γ) Έπιστημονικές ηλεκτρονικές υπολογιστικές μηχανές.

Διακρίνουμε τīs έξις τρεις κατηγορίες:

- 1) Προπρογραμματισμένες (Preprogrammed)
- 2) Προγραμματιζόμενες μέ πληκτρολόγηση (Programmable)
- 3) Προγραμματιζόμενες έξωτερικά μέ μαγνητική κάρτα, καρτελίδιο ή κασέτα (Programmable with a magnetic medium).

Οι ύπολογιστές τής κατηγορίας 1 έμφανίζονται κυρίως σέ μοντέλα τσέπης (Pocket calculators). Στίς άλλες δύο κατηγορίες ύπάρχουν καί έπιτραπέζια καί μοντέλα τσέπης.

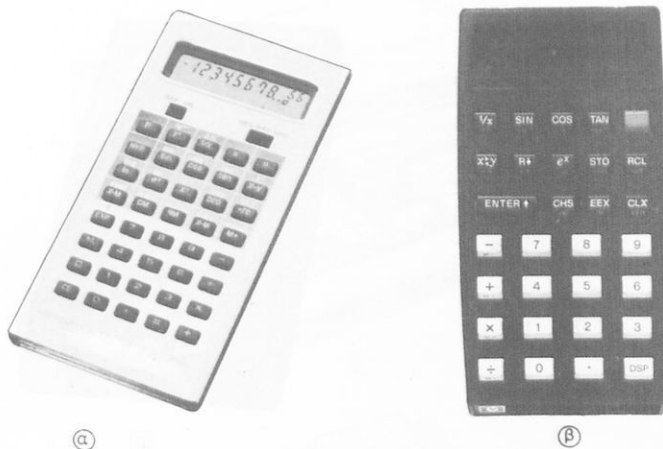
1) Προπρογραμματισμένες ηλεκτρονικές υπολογιστικές μηχανές.

Οι μηχανές αυτές έκτός άπό τό δεκαψήφιο άριθμητικό πληκτρολόγιο, διαθέτουν καί πλήκτρα, μέ τή βοήθεια τών όποιων ύπολογίζονται οι τιμές διαφόρων συναρτήσεων, όπως: τριγωνομετρικών (ήμίτονο, συνημίτονο, έφαπτομένη), τετραγωνική ρίζα, λογάριθμοι (φυσικοί καί δεκαδικόι), αντίστροφα άριθμών, κλπ.

Διαθέτουν 1 - 10 καταχωριστές δεδομένων καθώς καί καταχωριστές εισόδου καί άποτελέσματος. Διαθέτουν έπίσης όθόνη χωρητικότητας 8 - 12 ψηφίων. Τέλος οι περισσότερες έχουν δυνατότητα άπεικονίσσεως άριθμών στή μορφή τής κινητής ύποδιαστολής (Floating - point). Μερικές τέτοιες μηχανές βλέπομε στό σχήμα 11.3δ.

2) Προγραμματιζόμενες μέ πληκτρολόγηση ηλεκτρονικές υπολογιστικές μηχανές.

Οι ύπολογιστές τής κατηγορίας αυτής, έκτός άπό τīs λειτουργίες μιάς προπρο-



Σχ. 11.36.

(α) και (β) Προπρογραμματισμένες ηλεκτρονικές υπολογιστικές μηχανές τσέπης.

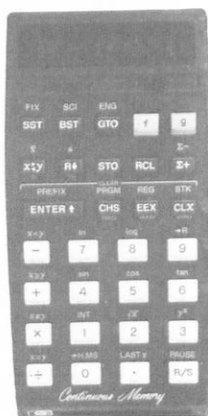
γραμματοσμένης μηχανής, που μπορούν να εκτελέσουν, μπορούν ακόμη να προγραμματισθούν για να εκτελέσουν και άλλες απλές εργασίες. Ο προγραμματισμός γίνεται με εντολές της γλώσσας μηχανής τους (άρα διαφέρουν από τύπο σε τύπο). Διαθέτουν μνήμη, σε ένα μέρος της οποίας καταχωρίζεται το πρόγραμμα. Δέχονται προγράμματα μέχρι 220 βήματα. Διαθέτουν αριθμό καταχωριστών δεδομένων (1 - 12), επιλογή πλήθους κλασματικών ψηφίων, παράσταση του αριθμού σε μορφή κινητής υποδιαστολής (Floating - point) κλπ. (σχ. 11.3ε).

3) Προγραμματιζόμενες εξωτερικά ηλεκτρονικές υπολογιστικές μηχανές.

Οι υπολογιστές της κατηγορίας αυτής, εκτός του ότι διαθέτουν όλα τα πλεονεκτήματα των δύο παραπάνω κατηγοριών, προσφέρουν και τη δυνατότητα αποθηκεύσεως σε μαγνητική κάρτα, καρτελίδιο ή κασέτα, κάθε προγράμματος που πληκτρολογείται στη μηχανή ώστε να χρησιμοποιηθεί όποτε το χρειαστούμε. Μπορούμε έτσι να σχηματίσουμε μία ολόκληρη βιβλιοθήκη από έτοιμα προγράμματα, που να επιλύουν πολλά διαφορετικά προβλήματα.

Υπάρχουν σε τύπους έπιτραπέζιους και τσέπης (σχ. 11.3στ). Μερικές εκτός από την απεικόνιση στην οθόνη, προβαίνουν και σε εκτύπωση [σχ. 11.3στ(γ)]. Άλλες απλώς εκτυπώνουν [σχ. 11.3στ(β)].

Ο χειρισμός των μηχανών αυτών, όσον αφορά στις βασικές πράξεις και στον υπολογισμό των τιμών των προπρογραμματισμένων συναρτήσεων, είναι, στις περισσότερες περιπτώσεις, ο ίδιος.



(α)



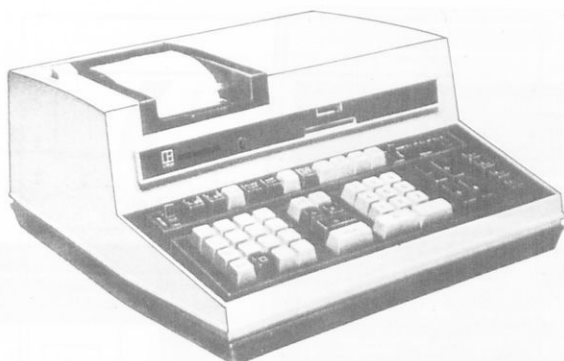
(β)

Σχ. 11.3ε.

(α) και (β) Προγραμματιζόμενες με πληκτρολόγηση ηλεκτρονικές υπολογιστικές μηχανές τσέπης.



(α)



(β)



(γ)

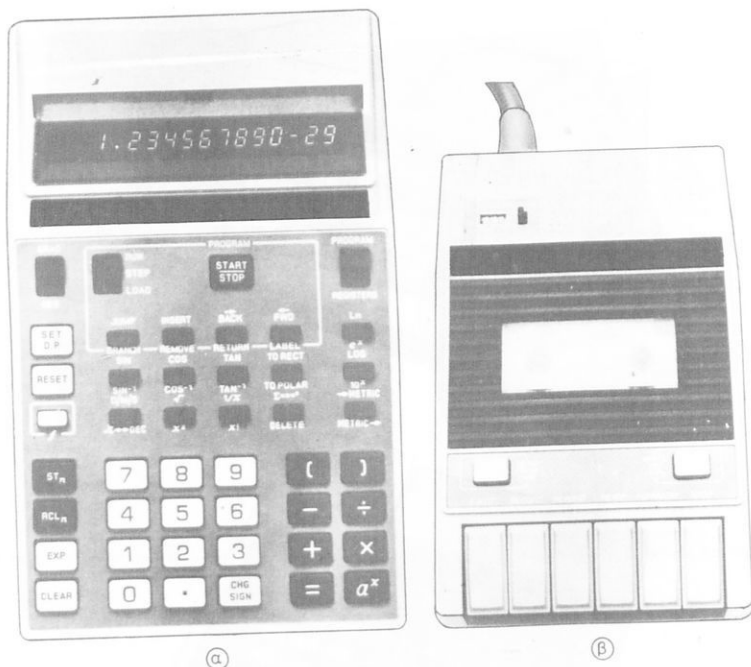
Σχ. 11.3στ.

(α) Προγραμματιζόμενη με καρτελίδιο ηλεκτρονική υπολογιστική μηχανή τσέπης (β) Προγραμματιζόμενη με καρτέλα επιτραπέζια μηχανή μόνον με έκτυπωση (γ) Προγραμματιζόμενη με καρτελίδιο επιτραπέζια μηχανή με έκτυπωση και οθόνη.

11.4 Χειρισμός μιάς ηλεκτρονικής υπολογιστικής μηχανής.

Θά χρησιμοποιήσουμε για τις παρακάτω περιγραφές μας, μία μηχανή προγραμματιζόμενη εξωτερικά πού, όπως είπαμε, συγκεντρώνει όλα τα χαρακτηριστικά των μηχανών και των τριών κατηγοριών (παράγραφος 13.3).

Ἡ μηχανή αὐτή [σχ. 11.4α(α)] διαθέτει φωτεινή οθόνη χωρητικότητας 16 ψηφίων, 12 ἀπλούς καταχωριστές δεδομένων (Data registers), ἓνα καταχωριστή είς-



Σχ. 11.4α.

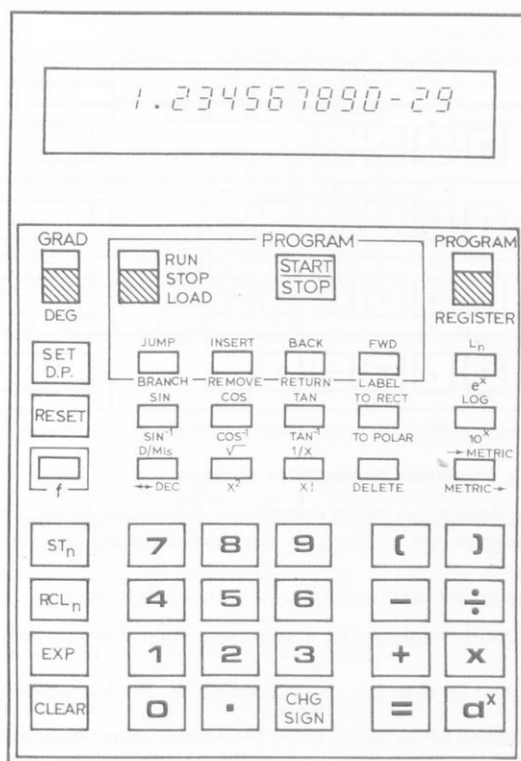
(α) 'Η ηλεκτρονική υπολογιστική μηχανή που θα χρησιμοποιήσουμε στη μελέτη του χειριστού (β) 'Η περιφερειακή μονάδα μαγνητικής ταινίας της μηχανής που εξετάζουμε.

όδου (Entry register) και ένα καταχωριστή αποτελέσματος. Διαθέτει ακόμη έσωτερική μνήμη, όπου μπορούμε να αποθηκεύσουμε ένα πρόγραμμα μέχρι 160 βημάτων.

Μπορεί να συνδεθεί εξωτερικά με μία περιφερειακή μονάδα μαγνητικής κασέτας, όπου έχουμε τη δυνατότητα να εγγράψουμε όχι μόνο προγράμματα, αλλά και δεδομένα [σχ. 11.4α(β)].

Όπως βλέπουμε στο σχήμα 11.4α, εκτός από το βασικό δεκαψήφιο αριθμητικό πληκτρολόγιο, διαθέτει πληκτρα συναρτήσεων καθώς και ειδικά πληκτρα για τον καθαρισμό των καταχωριστών (CLEAR, RESET) εισόδου και αποτελέσματος για την αποθήκευση αριθμών σε καταχωριστές δεδομένων (STn) και για τον έλεγχο του προγράμματος (START - STOP).

Στή συνέχεια θα περιγράψουμε συνοπτικά τις σημαντικότερες λειτουργίες και τούς απαραίτητους χειρισμούς για τη κάθε μία τους, έχοντας ως οδηγό το σχ. 11.4β.



Σχ. 11.4β.

Τό πληκτρολόγιο της μηχανής που περιγράφουμε.

α) Έκτέλεση απλών αριθμητικών πράξεων.

Γιά τήν έκτέλεση απλών αριθμητικών πράξεων ανάμεσα σέ δύο όρους τής μορφής A ορ B όπου ορ = +, -, $\times$, $\div$.

α) Πληκτρολογούμε τόν πρῶτο όρο - αριθμό τής πράξεως (προσθετέο, αφαιρετέο, κλπ.).

β) Πιέζομε τό πληκτρο τής επιθυμητής πράξεως.

γ) Πληκτρολογούμε τό δεύτερο αριθμό.

δ) Πιέζομε τό πληκτρο $=$

Στό τέλος τής φάσεως (δ), τό αποτέλεσμα εμφανίζεται στήν οθόνη (στήν οθόνη εμφανίζονται ενδιάμεσα καί όλοι οί αριθμοί τή στιγμή που πληκτρολογούνται).

Παραδείγματα.

Υπολογισμός	Εκτέλεση – Χειρισμός	Αποτέλεσμα στην οθόνη
$8 + 4$	8 + 4 <input <="" input="" type="text" value="="/>	12.0000
$16 - 5$	1 6 - 5 <input <="" input="" type="text" value="="/>	11.0000
7×6.3	7 x 6 . 3 <input <="" input="" type="text" value="="/>	44.1000
$1.8 \div 3$	1 . 8 ÷ 3 <input <="" input="" type="text" value="="/>	0.6000

β) Υπολογισμός συνθέτων αριθμητικών πράξεων.

Μία σύνθετη αριθμητική σχέση αποτελείται από πολλές απλές. Στην περίπτωση αυτή χρησιμοποιούμε και τα σύμβολα των παρενθέσεων, είτε εμφανίζονται παρενθέσεις στη δοθείσα σχέση, είτε όχι. Π.χ. στη σύνθετη σχέση $3 + \frac{5}{4}$, παρ' όλο που δεν εμφανίζονται παρενθέσεις ή διαδικασία πρέπει να γίνει έτσι, σαν να είχαμε γράψει τη σχέση ως εξής: $3 + (\frac{5}{4})$. Διαφορετικά, αν πληκτρολογήσουμε τούς όρους και τα σύμβολα με τη σειρά, όπως τα βλέπουμε στην παράσταση, θα κάνουμε λάθος, γιατί θα υπολογίσουμε, αντί της σωστής τιμής, την τιμή $\frac{3+5}{4}$. Βέβαια, αν αλλάζαμε τη σειρά των πράξεων θα μπορούσαμε να μη χρησιμοποιήσουμε τις παρενθέσεις (π.χ. αν πρώτα κάναμε τη διαίρεση $\frac{5}{4}$ και κατόπιν προσθέταμε τό 3). Τό να βρούμε όμως τη σωστή σειρά κάθε φορά, είναι κάτι δύσκολο.

Παραδείγματα.

Υπολογισμός	Εκτέλεση — Χειρισμός	Αποτέλεσμα στην οθόνη
$6 + \frac{8}{4}$	6 + 8 ÷ 4 =	8.0000
$5 \times \left(\left(\frac{2}{3} - 1 \right) + \frac{3}{4} \right)$	5 × (2 ÷ 3 - 1) + 3 ÷ 4) =	2.0833
$6.42 + 5 - 7 \times 2.1$	6 . 4 2 + 5 - (7 × 2 . 1) =	-3.2800

γ) Ἡ ἐκτέλεση συνθέτων ἀριθμητικῶν πράξεων στὶς ἀριθμομηχανές μέ μνήμη μιᾶς θέσεως.

Πολλές ἀπὸ τίς ἠλεκτρονικὲς ἀριθμομηχανές τεσσάρων πράξεων ἢ καὶ τίς προγραμματισμένες διαθέτουν μικρὴ μνήμη μιᾶς θέσεως καὶ μεγέθους 6 - 12 ψηφίων (ὄση καὶ ἡ χωρητικότητά τῆς οἰκίας τους), ἐκτὸς ἀπὸ τοὺς καταχωριστὲς εἰσόδου καὶ ἀποτελέσματος. Οἱ περισσότερες ἀπὸ τίς μηχανές αὐτές δὲν διαθέτουν πληκτρά μέ παρενθέσεις. Παρ' ὅλα αὐτὰ μποροῦν νὰ ἐκτελέσουν σύνθετες ἀριθμητικὲς πράξεις μέ τὴ βοήθεια τῆς μνήμης τους καὶ μέ τὴ τήρηση κάποιας ἱεραρχήσεως ὅσον ἀφορᾷ στὴν ἐκτέλεση τῶν πράξεων.

Οἱ μηχανές αὐτές διαθέτουν τὰ ἑξῆς πληκτρά:

[M+] Μέ τὴ πίεση τοῦ πληκτροῦ αὐτοῦ προσθίεται τὸ περιεχόμενο τοῦ καταχωριστῆ εἰσόδου στοῦ περιεχόμενου τῆς μνήμης καὶ τὸ ἄθροισμα ξανατοποθετεῖται στὴ μνήμη (Memory add).

[M-] Μέ τὴ πίεση τοῦ πληκτροῦ αὐτοῦ, ἀφαιρεῖται τὸ περιεχόμενο τοῦ καταχωριστῆ εἰσόδου ἀπὸ τὸ περιεχόμενο τῆς μνήμης καὶ ἡ διαφορά ξανατοποθετεῖται στὴ μνήμη (Memory subtract).

[RM] ἢ **[MR]** Μέ τὴ πίεση τοῦ πληκτροῦ αὐτοῦ τὸ περιεχόμενο τῆς μνήμης μεταφέρεται καὶ καταχωρίζεται στὸν καταχωριστὴ εἰσόδου, ἐνῶ παράλληλα ἐμφανίζεται καὶ στὴν οἰκία (Recall memory).

[CM] Μέ τὸ πληκτροῦ αὐτὸ μηδενίζεται τὸ περιεχόμενο τῆς μνήμης (Clear memory).

Στὴ κατηγορία αὕτὴ οἱ μηχανές ἀντὶ γιὰ τὰ πληκτρά **[RESET]** καὶ **[CLEAR]** διαθέτουν τὰ πληκτρά **[C]** καὶ **[CE]**, ποὺ μηδενίζουν τὸ μὲν πρῶτο ὅλους τοὺς καταχωριστὲς ἐκτὸς ἀπὸ τὴ μνήμη, τὸ δὲ δεύτερο, τὸν καταχωριστὴ εἰσόδου (Σὲ μερικὲς μηχανές ὑπάρχει μόνο ἓνα πληκτροῦ τὸ **[C/CE]** ποὺ μηδενίζει ὅλους τοὺς καταχωριστὲς).

Παραδείγματα.

Υπολογισμός	Χειρισμός	Αποτέλεσμα
$\frac{2}{3} \times \frac{4}{5} \times 7$	2 $\div$ 3 $\times$ 4 $\div$ 5 $\times$ 7 $=$	3.7333324
$(9 \times 8) + (14 \times 3) - \left(\frac{51}{3}\right)$	9 $\times$ 8 $=$ M+ 1 4 $\times$ 3 M+ 5 7 $\div$ 3 M- RM	

Σημειώνουμε στο σημείο αυτό ότι, αν πληκτρολογήσουμε λάθος κάποιο αριθμό, τότε απλώς πιέζουμε τό πληκτρο **CLEAR** καί τόν ξαναχτυπάμε χωρίς νά χρειάζεται νά ξαναπληκτρολογήσουμε ὅλη τή σχέση ἀπό τήν ἀρχή.

δ) Ύψωση αριθμοῦ σέ δύναμη.

Χρησιμοποιοῦμε τό πληκτρο $\boxed{a^x}$ ¹ Ἡ διαδικασία περιλαμβάνει τά ἑξῆς βήματα:

- Πληκτρολογοῦμε τόν ἀριθμό, πού εἶναι ἡ βάση (a)
- Πιέζομε τό πληκτρο $\boxed{a^x}$
- Πληκτρολογοῦμε τόν ἀριθμό, πού εἶναι ὁ ἐκθέτης (x).
- Πιέζομε τό πληκτρο $\boxed{=}$

Παραδείγματα.

Υπολογισμός	Ἐκτέλεση - Χειρισμός	Αποτέλεσμα στήν ὁθόνη
2^4	2 $\boxed{a^x}$ 4 $=$	16.0000
$(3+2)^2$	3 $+$ 2 $\boxed{a^x}$ 2 $=$	25.0000
$2^{1.3}$	2 $\boxed{a^x}$ 1 $\cdot$ 3 $=$	2.4623

¹ Σέ μερικές μηχανές τό πληκτρο αὐτό ἐμφανίζεται σάν $\boxed{y^x}$

ε) Τρόπος χρήσεως των καταχωριστών δεδομένων (Data storage register).

Σέ ένα καταχωριστή δεδομένων μπορούμε νά ἀποθηκεύσουμε ἕνα ἀριθμό ἢ ἕνα ἐνδιάμεσο ἀποτέλεσμα μιᾶς πράξης, γιά νά τό χρησιμοποιήσουμε ἀργότερα.

Κάθε καταχωριστής δεδομένων ταυτίζεται μέ ἕνα ἀπό τά πλήκτρα ἕως .

καί

Μποροῦμε νά καταχωρίσουμε ἕνα ἀριθμό σέ ἕνα καταχωριστή ὡς ἐξῆς:

α) Πληκτρολογοῦμε τόν ἀριθμό.

β) Πιέζομε τό πλήκτρο (Store to register).

γ) Πιέζομε ἕνα ἀπό τά πλήκτρα τῶν καταχωριστῶν.

Ἄν ὁ ἀριθμός γιά καταχώριση ἔχει προκύψει σάν ἀποτέλεσμα προηγούμενου ὑπολογισμού, τότε τό βῆμα (α) παραλείπεται.

Γιά νά καλέσουμε ἕνα ἀριθμό καταχωρισμένο σέ ἕνα καταχωριστή:

α) Πιέζομε τό πλήκτρο

β) Πιέζομε τό πλήκτρο πού ἀντιπροσωπεύει τό συγκεκριμένο καταχωριστή.

Παραδείγματα:

Λειτουργία	Ἐκτέλεση – Χειρισμός	Ἀποτέλεσμα στήν ὀθόνη
Νά καταχωρηθεῖ ὁ 3.14 στόν καταχωρητή 1.	3 . 1 4 ST<sub>1</sub> 1	3.1400
Νά ἀνακληθεῖ ὁ 3.14 ἀπό τόν καταχωρητή 1.	RCL<sub>1</sub> 1	3.1400
Νά καταχωρηθεῖ ὁ 2.4 στόν καταχωρητή 8.	2 . 4 CHG SIGN ST<sub>1</sub> 8	- 2.4000
Νά ἀνακληθεῖ ὁ 2.4 στόν καταχωρητή 8.	RCL<sub>1</sub> 8	- 2.4000

στ) Υπολογισμός τιμών συναρτήσεων (Functions).

Ο υπολογιστής, που μελετούμε, έχει ένα αριθμό πλήκτρων για τον υπολογισμό της τιμής διαφόρων συναρτήσεων. Κάθε τέτοιο πλήκτρο έχει δύο λειτουργίες. Με την απλή πίεση υπολογίζεται η τιμή της συναρτήσεως που η όνομασία της αναγράφεται επάνω από το πλήκτρο.

Πιέζοντας πρώτα το πλήκτρο  καί κατόπιν ένα πλήκτρο συναρτήσεως,

υπολογίζεται η τιμή της συναρτήσεως, η όνομασία της οποίας αναγράφεται κάτω από το πλήκτρο. Κατά τα άλλα η διαδικασία είναι απλή:

- Πληκτρολογούμε τον αριθμό για τον οποίο θέλουμε τον υπολογισμό της συναρτήσεως.
- Πιέζουμε το πλήκτρο της αντίστοιχης συναρτήσεως.

Παραδείγματα.

Υπολογισμός	Εκτέλεση – Χειρισμός	Αποτέλεσμα στην οθόνη
$\frac{1}{6}$	$\frac{1}{x}$ 6 	0.1667
$\sin(2.87)$	$\cos$ 2 . 8 7 	0.9987
6.4^2	6 . 4 	40.9600
$4!$	4 	24.0000
$\sqrt{9.3}$	9 . 3 	3.0490

ζ) Υπολογισμός μαθηματικών σχέσεων που περιλαμβάνουν καί συναρτήσεις.

Υπολογίζονται με τον ίδιο τρόπο που υπολογίζονται καί οι σύνθετες αριθμητικές σχέσεις¹.

¹ Σε κάθε περίπτωση αδύνατου υπολογισμού π.χ. διαιρέσεως ενός αριθμού με το 0 ή υπολογισμού του λογαρίθμου του 0 κλπ. εμφανίζεται στην οθόνη ο χαρακτήρας E (Error = λάθος). Σε άλλες μηχανές στις περιπτώσεις αυτές αναβοσβήνει ο αριθμός 0.

Παραδείγματα.

Υπολογισμός	Εκτέλεση - Χειρισμός	Αποτελ. στην οθόνη
$4 \times \sqrt{6} + \frac{7}{4}$	$\sqrt{x}$ 4 x 6 $\sqrt{x}$ + [7 $\div$ 4] [] =	11.5479
$2 \times \eta\mu(3.1) + 6$	$\sin$ 2 x 3 . 1 $\sin$ + 6 =	6.0973
$3.2 \times e^{2.1} + \sqrt{7}$	3 [] 2 x 2 . 1 [$\square$] e^x + 7 $\sqrt{x}$ =	28.7774

η) Μετατροπή γωνιών.

Μετατροπή από μοίρες σε ακτίνια.

Η διαδικασία είναι:

- 1) Θέτουμε τό διακόπτη  GRAD στην κάτω θέση (DEG)
- 2) Πληκτρολογούμε την τιμή της γωνίας (που εκφράζεται σε μοίρες καί κλάσμα της μοίρας).
→Metric
- 3) Πιέζουμε τό πληκτρο 
- 4) Πιέζουμε τό πληκτρο 

Παραδείγματα.

Υπολογισμός	Εκτέλεση - Χειρισμός	Αποτέλεσμα στην οθόνη
Νά μετατραπούν οί 48° σε ακτίνια	GRAD  →Metric 4 8  DEG	0.8377
Νά βρεθεί τό ημίτονο των 56,74	GRAD  →Metric 5 6 . 7 4  DEG 	0.0172

Μετατροπή από άκτίνια σε μοίρες.

Η διαδικασία είναι:

1) Θέτουμε το διακόπτη  στην κάτω θέση.

2) Πληκτρολογούμε τη γωνία.

3) Πιέζουμε το πλήκτρο 

4) Πιέζουμε το πλήκτρο 

5) Πιέζουμε το πλήκτρο 

metric →
CHG
SIGN

Παραδείγματα.

Υπολογισμός	Εκτέλεση - Χειρισμός	Αποτελ. στην οθόνη
Νά μετατρα- πουν τα 2.851 άκτίνια σε μοίρες.	 2 8 5 1   CHG DEG SIGN	163,3502
Νά μετατρα- πουν τα 14.35 άκτίνια σε μοίρες.	 1 4 3 5   CHG DEG SIGN	822,1944

θ) Καθορισμός πλήθους κλασματικών ψηφίων.

Γίνεται πιέζοντας πρώτα το πλήκτρο  και κατόπιν τόν αριθμό, που παριστάνει το επιθυμητό πλήθος κλασματικών ψηφίων.

11.5 Έφαρμογές - Παραδείγματα.

Παράδειγμα 1.

Νά υπολογισθεί το βεληνεκές x ενός βλήματος πυροβόλου, που έκτοξεύεται υπό γωνία $\theta = 10^\circ$ από τη θέση Α με αρχική ταχύτητα: $V_0 = 400$ m/s.

Όπως είναι γνωστό, ο τύπος του βεληνεκούς είναι:

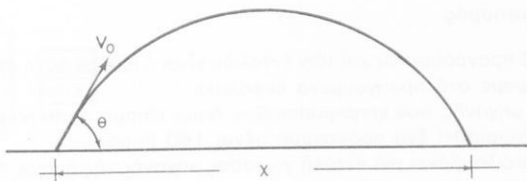
$$x = \frac{V_0^2 \sin 2\theta}{g}$$

όπου:

$$\theta = 10^\circ$$

$$V_0 = 400 \text{ m/s}$$

$$g = 9,81 \text{ m/s}^2$$



Για να βρούμε το αποτέλεσμα θα κάνουμε τούς εξής χειρισμούς:

4	0	0	$\square_f$	x^2	x	[	2	x	1	0	]	SIN	$\div$
9	.	8	1	=									

Τό αποτέλεσμα πού θα εμφανισθεῖ στήν οθόνη θα εἶναι:

5040.0325

Παράδειγμα 2.

Στή θεωρία τῆς σχετικότητας, ἡ μάζα ἑνός σώματος πού κινεῖται ἰσοταχῶς μέ ταχύτητα V , ὑπολογίζεται ἀπό τή σχέση τοῦ *Einstein*:

$$m = \frac{m_0}{\sqrt{1 - \frac{V^2}{C^2}}}$$

ὅπου: m_0 ἡ μάζα ἡρεμίας τοῦ σώματος (ὅταν εἶναι ἀκίνητο)

C ἡ ταχύτητα τοῦ φωτός (2.9979×10^8 m/s).

Ζητεῖται νά ὑπολογισθεῖ ἡ μάζα ἑνός ἠλεκτρονίου πού κινεῖται μέ ταχύτητα $0.96C$. Ἡ μάζα ἡρεμίας τοῦ ἠλεκτρονίου εἶναι $m_0 = 9.1086 \times 10^{-31}$ kg.

Γιά τόν ὑπολογισμό αὐτό θα κάνουμε τούς εξής χειρισμούς:

9	.	1	0	8	6	EXP	CHG SIGN	3	1	$\div$	[	1	-	
.	9	6	$\square_f$	x^2	]	$\sqrt{}$	=							

Τό αποτέλεσμα πού θα εμφανισθεῖ στήν οθόνη θα εἶναι:

3.253071428 - 30

11.6 Προγραμματισμός.

Ἡ ἔννοια τοῦ προγράμματος καί τῶν ἐντολῶν εἶναι ἡ ἴδια μέ αὐτή πού γνωρίσαμε καί περικύρναμε στά προηγούμενα κεφάλαια.

Ὁ τύπος τῆς μηχανῆς, πού χρησιμοποιοῦμε, ὅπως εἶπαμε, διαθέτει μνήμη, ὅπου μπορεῖ νά καταχωρισθεῖ ἕνα πρόγραμμα μέχρι 160 βημάτων.

Κάθε βῆμα περιλαμβάνει μία ἐντολή γλώσσας μηχανῆς τῆς ὁποίας ἡ λειτουργία παριστάνεται μέ ἕνα πλήκτρο. Ὁ κώδικας λειτουργίας τῆς ἐντολῆς εἶναι ἕνας τριψήφιος δεκαδικός ἀριθμός, πού χρειάζεται νά τόν ξέρομε, μόνο ὅταν κάνομε ἔλεγχο γιά τή σωστή γραφή ἑνός προγράμματος.

Ὁ προγραμματισμός ἐδῶ ἀκολουθεῖ τά ἑξῆς βήματα:

- Σχεδιάζεται ἕνα ἀπλό διάγραμμα ροῆς, πού ἀντιπροσωπεύει τή λογική πού θά χρησιμοποιηθεῖ στήν ἐπίλυση τοῦ προβλήματος.
- Κωδικοποιεῖται μέ τή βοήθεια τῶν ἐντολῶν τῆς μηχανῆς.
- Τό πρόγραμμα πληκτρολογεῖται καί ἀποθηκεύεται στή μνήμη.
- Τό πρόγραμμα εἶναι ἕτοιμο γιά ἐκτέλεση.

α) Προκαταρκτικοί χειρισμοί.

- Μέ τή βοήθεια τοῦ διακόπτη RUN/STEP/LOAD ἀποθηκεύεται τό πρόγραμμα στή μνήμη ἢ προετοιμάζεται γιά ἐκτέλεση.

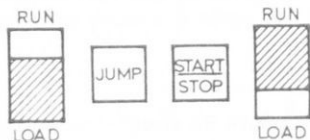
Ὅταν ὁ διακόπτης αὐτός βρίσκεται στή θέση LOAD, ἀποθηκεύομε τίς ἐντολές τοῦ προγράμματος σέ διαδοχικές θέσεις τῆς μνήμης, πιέζοντας τά κατάλληλα πλήκτρα.

Ὅταν μετά τήν ἀποθήκευση τοῦ προγράμματος ὁ διακόπτης αὐτός τεθεῖ στή θέση RUN, τό πρόγραμμα εἶναι ἕτοιμο γιά ἐκτέλεση.

- Μέ τή βοήθεια τοῦ πλήκτρου  1) ἀρχίζει ἡ ἐκτέλεση τοῦ προγράμματος καί 2) γίνεταί ἡ προσωρινή παύση στήν ἐκτέλεση γιά νά εἰσαχθεῖ κάποιο δεδομένο ἢ γιά νά ἀπεικονισθεῖ στήν ὁθόνη κάποιο ἐνδιάμεσο ἀποτέλεσμα.

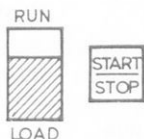
- Τό πλήκτρο  χρησιμοποιεῖται γιά ἄλλατα ἀπό ἕνα σημεῖο τοῦ προγράμματος σέ ἄλλο. Ὅταν ὁμοῦς προηγεῖται ἡ  τότε ὁ ἔλεγχος τοῦ προγράμματος μεταφέρεται στήν πρώτη ἐντολή.

- Ἡ διαδικασία γιά τήν ἀποθήκευση ἑνός προγράμματος στή μνήμη περιλαμβάνει τούς χειρισμούς



καί ἀκολουθεῖ ἡ πληκτρολόγηση τῶν ἐντολῶν.

- Ἡ διαδικασία γιά τήν ἐκτέλεση ἑνός προγράμματος περιλαμβάνει τούς χειρισμούς



Παράδειγμα.

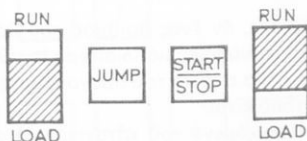
Νά γραφεί πρόγραμμα για τον υπολογισμό του όγκου ενός κυλίνδρου, όταν είναι γνωστή η διάμετρος της βάσεώς του d και το ύψος του h .

Ο τύπος του όγκου κυλίνδρου είναι:

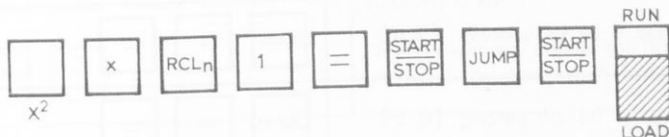
$$V = \pi \frac{d^2}{4} h$$

Τό πρόγραμμα θα έχει την παρακάτω μορφή:

α) Για την αποθήκευση του προγράμματος θα γίνουν οι χειρισμοί



β) Οι εντολές του προγράμματος θα είναι:



Στή συνέχεια δίνουμε τα δεδομένα:



Τό αποτέλεσμα όπως θα εμφανισθεί στην οθόνη είναι:

3.1415

β) 'Η Έννοια της Label.

Label είναι ένα συμβολικό όνομα που δίνεται σε ένα πρόγραμμα ή σε ένα σημείο του προγράμματος. 'Η Label αποτελεί ένα βήμα που προηγείται της πρώτης εντολής του προγράμματος ή του σημείου του προγράμματος αντίστοιχα.

Σάν labels χρησιμοποιούνται τα αριθμητικά πλήκτρα έως . Τό καί τό μαζί μέ τό LABEL

Οι Labels χρησιμοποιούνται κυρίως σε προγράμματα, όπου υπάρχουν εντολές διακλαδώσεως, ή άλματος υπό συνθήκη.

γ) 'Εντολές άλματος υπό συνθήκη.

Μέ τις εντολές αυτές μπορούμε να ελέγξουμε, αν ένας αριθμός που βρίσκεται στον καταχωριστή εισόδου ¹ (Entry register) ικανοποιεί μία ιδιαίτερη συνθήκη καί αν ναι, είναι δυνατό να γίνει κάποιο άλμα σε άλλο σημείο του προγράμματος. 'Αν ή συνθήκη δεν ισχύει, τό άλμα δεν πραγματοποιείται.

'Εχομε τή δυνατότητα να ελέγξουμε τό περιεχόμενο του καταχωριστή εισόδου γιά τις εξής περιπτώσεις:

Συνθήκη	'Εντολή - Χειρισμός
Μεγαλύτερο του μηδενός (> 0)	JUMP + □
Μικρότερο του μηδενός (< 0)	JUMP - □
Ίσο μέ τό μηδέν ($= 0$)	JUMP = □
Μεγαλύτερο ή ίσο του μηδενός (≥ 0)	JUMP + = □
Μικρότερο ή ίσο του μηδενός (≤ 0)	JUMP - = □
Διάφορο του μηδενός ($\neq 0$)	JUMP + - □

Παράδειγμα.

Νά βρεθεί το άθροισμα των 20 πρώτων φυσικῶν αριθμῶν. Τό πρόγραμμα θά ἔχει τήν ἑξῆς μορφή²:

	Ἐπεξήγηση
	Προετοιμασία γιά τήν εἰσαγωγή τῶν ἐντολῶν τοῦ προγράμματος στή μνήμη.
	Μηδενίζονται οἱ καταχωριστές 0 καί 1.
	Στό σημεῖο αὐτό τοῦ προγράμματος τίθεται μιά ἀριθμητική ἔνδειξη (Label), ἡ 3.
	Προστίθεται τό περιεχόμενο τοῦ καταχωριστή 0 στό περιεχόμενο τοῦ καταχωριστή 1.
	Ἀφαιρεῖται ὁ ἀριθμός 20 ἀπό τόν καταχωριστή εἰσόδου (Entry register).
	Ἄν τό περιεχόμενο τοῦ καταχωριστή εἰσόδου εἶναι ἴσο μέ τό μηδέν, γίνεται ἄλλα στό σημεῖο τοῦ προγράμματος μέ ἀριθμητική ἔνδειξη 4, διαφορετικά συνεχίζεται ἡ ἐκτέλεση τοῦ προγράμματος μέ τίς ἐπόμενες ἐντολές.
	Αὐξάνεται τό περιεχόμενο τοῦ καταχωριστή 0 καί 1.
	Γίνεται ἄλλα στό σημεῖο τοῦ προγράμματος μέ τήν ἀριθμητική ἔνδειξη 3.

¹ Στόν καταχωριστή εἰσόδου τοποθετεῖται ὁ ἀριθμός πού κάθε φορά πληκτρολογεῖται. Τό περιεχόμενο τοῦ ἐμφανίζεται πάντα στήν ὀθόνη.

² Ἡ κατακόρυφη ἀνάπτυξη τῶν ἐντολῶν εἶναι πιό ἐποπτική ὅπως διαπιστώνεται ἀπό τόν τρόπο παρουσιάσεως τῶν παραδειγμάτων τῶν παραγράφων 11.6(a) καί 11.6(γ).

			Στό σημείο αυτό τίθεται η αριθμητική ένδειξη 4. Έμφανίζεται στην οθόνη το περιεχόμενο του καταχωριστή 1. Έπιστροφή στην αρχή του προγράμματος και προετοιμασία για την εκτέλεσή του.
			
			

Τό πρόγραμμα αυτό αρχίζει νά εκτελείται, μόλις πιάσουμε τό πλήκτρο.



Στό τέλος τής εκτελέσεως θά έμφανισθεί στην οθόνη τό αποτέλεσμα, πού θά είναι:

210.0000

Άσκησης.

1. Νά εκτελεσθούν οι πράξεις:

$$\begin{array}{lll} -2.012 + 4.756 & 6.24 \times 0.15 & 7.4 \div 2.1 \\ 3.1415 \times 6.28 & -100.05 \div 724 & 243.1 \times 506 \end{array}$$

2. Επίσης οι πράξεις:

$$1.64 + 3.51 - 7.4 \quad 2 \times (3.12 - 1.8) + 6.54 \quad 99.9 \div 3 + 1$$

$$5.02 \times [3.1 + (2.8 + \frac{1}{6})] \quad \frac{-752 + 286}{3} \quad 951 \times 1000 - \frac{7}{6} \times 9.1$$

3. Επίσης οι πράξεις:

$$4.1 \times \eta\mu(2.87) + 6.1 \times \sigma\upsilon\nu(0.13) \quad 3 \times \sqrt{1.4} - 200 \times e^{1.3} + 5.1$$

$$\frac{7.71}{6.08} \times \log 2.3 + 5 \times (-6.6) \quad (2.1 + 4.5)^2 + 11.2 - 6.8$$

4. Επίσης οι πράξεις:

$$7 \times [\eta\mu(2.1) + \sigma\upsilon\nu(3.0)] - (4.1 + 5.2)^2 \quad \frac{1 + 6e^{2.1} - \sqrt{3.7}}{2 + 6\eta\mu(0.7)}$$

$$6 \times [3 \times (4 \times (5-7) + 8) - 9]$$

5. Αυτόκινητο κινείται με ευθύγραμμη ομαλά επιταχυνόμενη κίνηση. Αν η επιτάχυνσή του είναι 4 m/s^2 και βρίσκεται σε ήρεμία στην αρχή μετρήσεως των χρόνων, να βρεθεί το διάστημα που θα διανύσει κινούμενο επί χρόνο 40s.
6. Να γράψετε ένα πρόγραμμα, που να υπολογίζει την ολική ωμική αντίσταση δύο αντιστάσεων συνδεδεμένων παράλληλα (έφαρμογή για $R_1 = 4$ και $R_2 = 8$).
7. Να γράψετε ένα πρόγραμμα για τον υπολογισμό του μέσου όρου 4 αριθμών.

ΚΕΦΑΛΑΙΟ ΔΩΔΕΚΑΤΟ

MINI-ΥΠΟΛΟΓΙΣΤΕΣ

12.1 Γενική ταξινόμηση τῶν ηλεκτρονικῶν ὑπολογιστῶν.

Τοὺς ηλεκτρονικοὺς ὑπολογιστὲς τοὺς κατατάσσομε ἀνάλογα μὲ τὸ μέγεθός τους στὶς ἐξῆς κατηγορίες:

α) **Μίνι-ὑπολογιστές (Minicomputers).** Οἱ ὑπολογιστὲς αὐτοὶ πρωτοεμφανίσθηκαν στὰ 1966 καὶ ἡ ἐξέλιξή τους ἀπὸ τότε ὑπῆρξε ραγδαία. Εἶναι μικροὶ σὲ μέγεθος καὶ φθinhoί, ἔχουν ὁμως μεγάλες δυνατότητες, ποὺ ἐξαρτῶνται κατὰ κύριο λόγο ἀπὸ τὸ μηχανικὸ ἐξοπλισμὸ τους (Hardware), δηλαδὴ τὴ μνήμη, τὶς περιφερειακὲς μονάδες κλπ. Τοὺς ὑπολογιστὲς αὐτοὺς θὰ τοὺς μελετήσομε λεπτομερέστερα πιο κάτω, ἐπειδὴ παρουσιάζουν ξεχωριστὸ ἐνδιαφέρον. Μερικοὺς τύπους μίνι-ὑπολογιστῶν βλέπομε στὸ σχῆμα 12.1α.

β) **Μικροῦ μεγέθους ὑπολογιστές (Small-size computers).** Ἡ διάκριση ἀνάμεσα στοὺς ὑπολογιστὲς τῆς κατηγορίας αὐτῆς καὶ στοὺς μίνι-ὑπολογιστὲς εἶναι ἀσαφής. Μερικοὶ τοὺς συμπεριλαμβάνουν στὴν ἴδια κατηγορία.

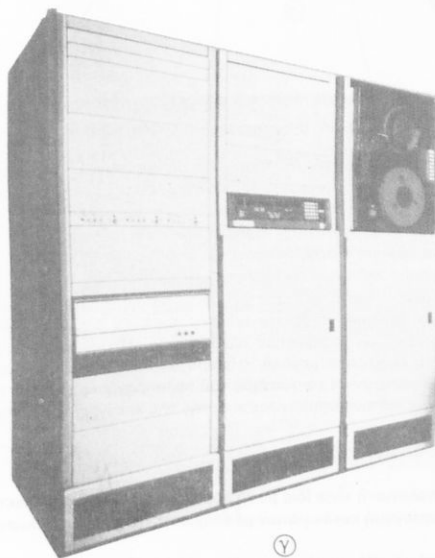
Εἶναι γρήγοροι ὑπολογιστὲς, μὲ ταχύτερες περιφερειακὲς μονάδες ἀπὸ ἐκεῖνες τῶν μίνι-ὑπολογιστῶν. Ἕνα σύστημα μικροῦ μεγέθους ὑπολογιστῆ ἀποτελεῖται βασικά ἀπὸ μία κεντρικὴ μονάδα ἐπεξεργασίας μὲ μνήμη χωρητικότητας 4K-64K bytes· μία μονάδα ἀναννώσεως-διατήρησης δελτίων



α



(β)



(γ)

Σχ. 12.1α.

(α), (β), (γ). Διάφοροι τύποι Μίνι υπολογιστών.

(80 ή 96 στηλών), 2-4 μονάδες μαγνητικής ταινίας και ένα έκτυπωτή. Σε μερικά συστήματα, αντί των ταινιών, χρησιμοποιούνται 2-3 μονάδες μαγνητικού δίσκου, όπου οι δίσκοι-φορείς σχηματίζουν δέσμη (Disk pack) ή βρίσκονται μεμονωμένοι σε θήκες (Disk cartridges.) Ένα τέτοιο υπολογιστή βλέπουμε στο σχήμα 12.1β.



Σχ. 12.1β.

“Ένας υπολογιστής μικρού μεγέθους.

γ) **Μεσαίου μεγέθους υπολογιστές (Medium-size computers).** Σε σύγκριση με τους υπολογιστές της προηγούμενης κατηγορίας, οι μηχανές της κατηγορίας αυτής πλεονεκτούν στα εξής σημεία:

- 1) Έχουν μεγαλύτερη ταχύτητα επεξεργασίας.
- 2) Έχουν μεγαλύτερη χωρητικότητα μνήμης (16K-250K θέσεις).
- 3) Έχουν ταχύτερες περιφερειακές μονάδες (κυρίως ταινίες ή δίσκους).
- 4) Έχουν μεγαλύτερους και ταχύτερους έκτυπωτές.

Μερικά τέτοια συγκροτήματα μεσαίου μεγέθους υπολογιστών βλέπουμε στο σχήμα 12.1γ.

δ) **Μεγάλου μεγέθους υπολογιστές (Large-scale computers).** Έχουν την ίδια βασική δομή με τους υπολογιστές των προηγούμενων κατηγοριών, αλλά με ασύγκριτα μεγαλύτερες δυνατότητες. Η ταχύτητα επεξεργασίας είναι εξαιρετικά μεγάλη, ή χωρητικότητα της μνήμης κυμαίνεται από 131K - 1000K bytes. Όλοι λειτουργούν με το σύστημα του πολυπρογραμματισμού, που θα δούμε στο επόμενο κεφάλαιο. Μερικούς τύπους υπολογιστών αυτής της κατηγορίας βλέπουμε στο σχήμα 12.1δ.

12.2 Τί είναι ένας μίνι-υπολογιστής.

Η δομή ενός μίνι-υπολογιστή είναι ίδια με τη δομή των μεγαλύτερων ηλεκτρονικών υπολογιστών. Ο σχεδιασμός και η κατασκευή κυκλωμάτων με ελάχιστες διαστάσεις, μείωσε σημαντικά το μέγεθος των υπολογιστών.

Ένας μίνι-υπολογιστής διαθέτει μία κεντρική μονάδα επεξεργασίας, παρόμοια με τη μονάδα των μεγάλων υπολογιστών, καθώς και περιφερειακές μονάδες. Η μεγάλη τεχνολογική ανάπτυξη στα τελευταία χρόνια επέτρεψε όχι μόνο τη σμίκρυνση των διαφόρων τμημάτων ενός υπολογιστή, αλλά και τη μεγάλη μείωση του κόστους.



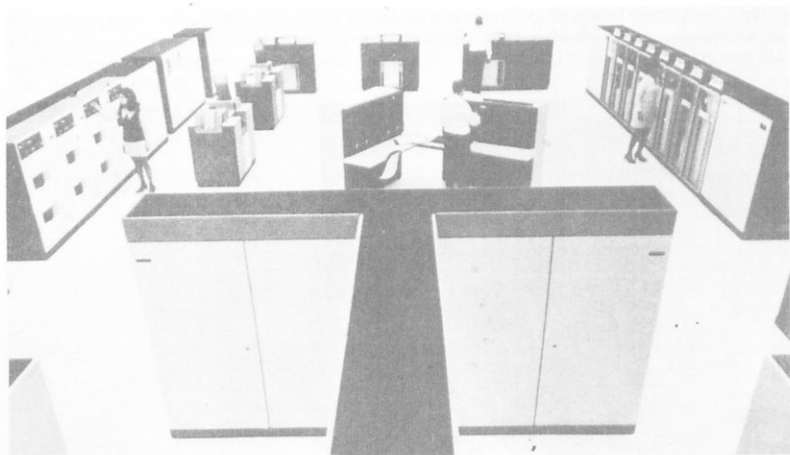
(α)



(β)

Σχ. 12.1γ.

(α.) (β). Διάφοροι τύποι υπολογιστών μεσαίου μεγέθους.



(α)



(β)

Σχ. 12.16.

(α), (β). Υπολογιστές μεγάλου μεγέθους

Για τούς μίνι-υπολογιστές έχουν σχεδιασθεί ειδικές μικρές περιφερειακές μονάδες, που χρησιμοποιούνται σήμερα εύρυθτα.

Οι περισσότεροι μίνι-υπολογιστές είναι υπολογιστές για γενικές εφαρμογές και έκτελούν προγράμματα, που αποθηκεύονται σε ένα μέρος της μνήμης τους.

Οι διαφορές που παρουσιάζουν με τούς μεγαλύτερους υπολογιστές, συνίστανται κυρίως στο μέγεθος μιάς λέξεως (θέσεως) της μνήμης, στη χωρητικότητα της μνήμης, στην ταχύτητα επεξεργασίας, στις ταχύτητες των περιφερειακών μονάδων κλπ.

Συνεπώς ο όρος μίνι-υπολογιστής προέρχεται βασικά από το μικρό μέγεθός του και το χαμηλό κόστος του.

12.3 Τό μέγεθος της λέξεως (Word size).

Τό μέγεθος της λέξεως (word) στους περισσότερους μίνι-υπολογιστές είναι 12 ή 18 bits. Βέβαια υπάρχουν και μίνι-υπολογιστές με μέγεθος λέξεως 8, 16, 24 και 32 bits.

Η χωρητικότητα της μνήμης τους φθάνει μέχρι 64K λέξεις σε ορισμένους τύπους.

12.4 Περιφερειακές μονάδες.

Θεωρητικά όλες οι περιφερειακές μονάδες που χρησιμοποιούνται στα συγκροτήματα των μεγάλων ηλεκτρονικών υπολογιστών μπορούν να χρησιμοποιηθούν και στους μίνι-υπολογιστές. Η διαφορά όμως κόστους ανάμεσα στην κεντρική μονάδα επεξεργασίας και αυτού στις περιφερειακές μονάδες είναι τέτοια, ώστε στην πράξη να χρησιμοποιούνται μικρές, ειδικά σχεδιασμένες, περιφερειακές μονάδες, αντίστοιχες προς τις μεγάλες, που είναι όμως πολύ πιο φθηνές. Μερικές από αυτές θα περιγράψουμε στη συνέχεια.

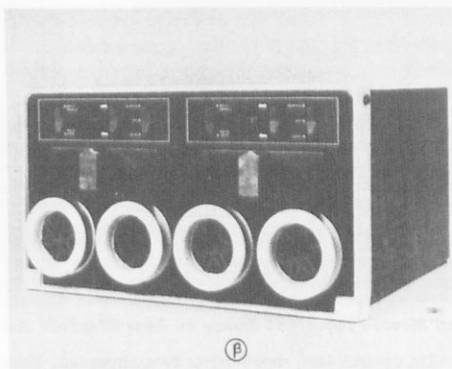
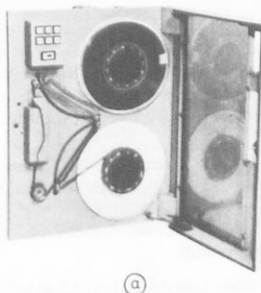
α) Μονάδες μαγνητικής ταινίας σε μικρό καρούλι (Small magnetic tape unit).

Η ταινία, που τοποθετείται σε μία τέτοια μονάδα, έχει μικρότερο μήκος από την κανονική, άρα και η χωρητικότητά της είναι μικρότερη.

Ο δγκος της μονάδας, όπως φαίνεται και στο σχήμα 12.4α(α) είναι σαφώς πιο μικρός. Η πυκνότητα έγγραφής είναι 1600 ή 1800 bpi στα 9 κανάλια. Στο σχήμα 12.4 α(β) βλέπουμε μία δίδυμη μονάδα μαγνητικής ταινίας για μίνι-υπολογιστή.

β) Μονάδα μαγνητικής ταινίας σε κασέτα (Magnetic - tape cassette).

Στη μονάδα αυτή τοποθετείται κασέτα μαγνητικής ταινίας, όμοια με τις κοινές κασέτες των φορητών μαγνητοφώνων. Η χωρητικότητά της κασέτας φθάνει τούς 500.000 χαρακτήρες (σχ. 12.4β).



Σχ. 12.4α.

(α) Μία μικρού μεγέθους μονάδα μαγνητικής ταινίας. (β) Μία δίδυμη μονάδα μαγνητικής ταινίας για μίνι υπολογιστή.



Σχ. 12.4β.

Μονάδα μαγνητικής ταινίας σέ κασέτα.

γ) Κωδικοποιητής μαγνητικής ταινίας σέ κασέτα (Key - to - cassette unit).

Είναι μικρού μεγέθους συσκευή, με την οποία τὰ δεδομένα καταγράφονται κωδικοποιημένα απ' εὐθείας σέ κασέτα, με ἀπλή πληκτρολόγηση. Είναι συνεπῶς μιά μικρογραφία τοῦ κωδικοποιητῆ μαγνητικῆς ταινίας, πού εἶδαμε στήν παράγραφο 3.18. Στή συνέχεια ἡ κασέτα μέ τὰ δεδομένα τοποθετεῖται στή μονάδα μαγνητικῆς κασέτας ἑτοιμῇ γιά ἐπεξεργασία.

δ) Μονάδα μαγνητικῆς ταινίας σέ θήκη (Magnetic tape cartridge unit).

Ἡ μονάδα αὐτή εἶναι πολύ εὐχρηστῇ. Μία θήκη μπορεῖ νά περιέχει 300 πόδια μαγνητικῆς ταινίας, ὅπου καταχωροῦνται μέχρι 2 ἑκατομμύρια χαρακτήρες μέ πυκνότητα ἐγγραφῆς μέχρι 1800 bits/inch.

Ἐπομένως ἡ χωρητικότητά τους εἶναι τετραπλάσια ἀπό ὅση μιᾶς κασέτας (σχ. 12.4γ).

ε) Μονάδα εὐκαμπτου μαγνητικοῦ δίσκου ἢ δισκέττας (Flexible disk unit ἢ diskette unit ἢ floppy disk unit).

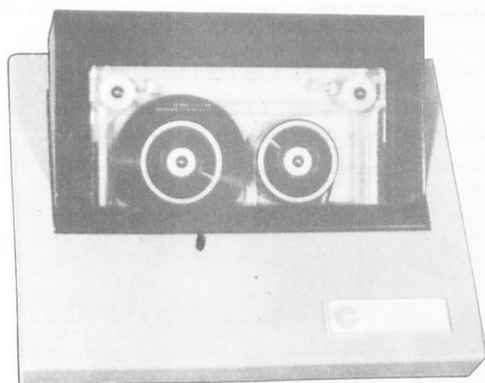
Ἡ δισκέττα εἶναι ἕνας νέος τύπος μαγνητικοῦ δίσκου, πού ἀρχισε νά χρησιμοποιεῖται πρόσφατα. Ἐχει μικρότερες διαστάσεις (περίπου ὅσο ἕνας μικρός δίσκος γραμμωφώνου) ἀπό τόν κανονικό, εἶναι εὐκαμπτος καί ἡ ἐπιφάνειά του εἶναι καλυμμένη μέ ἕνα στρώμα μαγνητικοῦ ὕλικου. Ἡ δισκέττα εἶναι τοποθετημένη μέσα σέ πλαστική θήκη (φάκελλο) (σχ. 12.4δ). Ὁλόκληρη ἡ θήκη εἰσάγεται στή μονάδα δίσκου, προφυλάσσοντας ἔτσι τό περιεχόμενό της ἀπό ἐξωτερικές φθορές (σκόνη, ὑγρασία, ἐπαφή μέ τὰ δάχτυλα τοῦ χεριοῦ κλπ.).

Σέ μιά τέτοια δισκέττα μποροῦν νά καταχωρισθοῦν μέχρι 500.000 χαρακτήρες. Τό κόστος τῶν μονάδων εἶναι πολύ μικρό, ἐνῶ οἱ δισκέττες εἶναι πολύ φθηνές. Γι' αὐτό καί ἡ μονάδα αὐτή χρησιμοποιεῖται εὐρύτατα.

στ) Μονάδα μαγνητικοῦ δίσκου σέ θήκη (Magnetic disk cartridge unit).

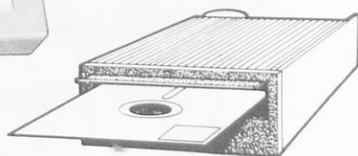
Στή μονάδα αὐτή τοποθετεῖται ἕνας μαγνητικός δίσκος σέ θήκη (cartridge) [σχ. 12.4ε(α)].

Ἡ χωρητικότητά τοῦ δίσκου φθάνει τὰ 5 ἑκατομμύρια bytes. Στό σχῆμα 12.4ε(β) βλέπομε μιά διδυμη μονάδα μαγνητικοῦ δίσκου σέ θήκη.



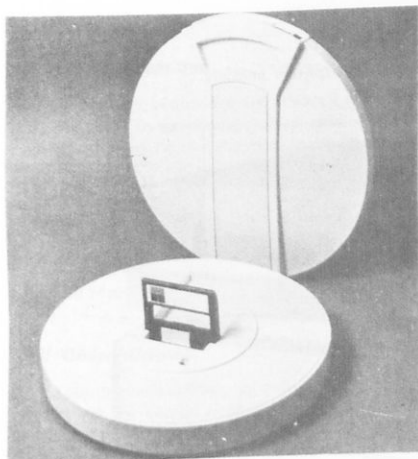
Σχ. 12.4γ.

Μονάδα μαγνητικής ταινίας σε θήκη.



Σχ. 12.4δ.

Μονάδα μαγνητικής δισκέτας καί η δισκέτα.



(α)



(β)

Σχ. 12.4ε.

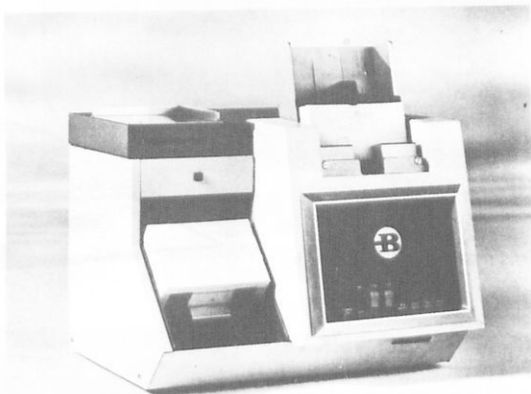
(α) Μαγνητικός δίσκος σε θήκη. (β) Δίδυμη μονάδα μαγνητικού δίσκου σε θήκη.

ζ) Μονάδα ανάγνωσης διατρήτων δελτίων (Card reader).

Ἡ ταχύτητα ἀνάγνωσης τῆς μονάδας αὐτῆς φθάνει τὰ 600 δελτία τὸ λεπτό. Μερικοὶ τύποι τῆς μονάδας αὐτῆς μποροῦν νὰ διαβάζουν δελτία 96 στηλῶν, ἐνῶ ἄλλοι τύποι δελτία 80 στηλῶν (σχ. 12.4στ).

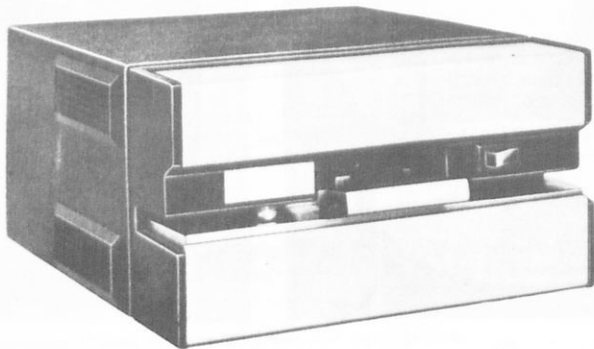
η) Μονάδα ἀνάγνωσης διάτρητης χαρτοταινίας (Paper tape reader).

Ἡ ταχύτητα ἀνάγνωσης τῆς φθάνει τοὺς 700 χαρακτῆρες τὸ δευτερόλεπτο (σχ. 12.4ζ).



Σχ. 12.4στ.

Μονάδα ἀνάγνωσης 80 στηλῶν διάτρητων δελτίων.

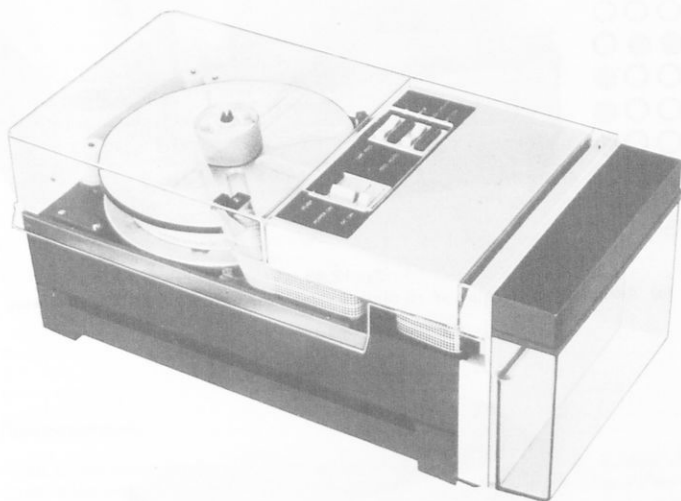


Σχ. 12.4ζ.

Μονάδα ἀνάγνωσης διάτρητης χαρτοταινίας.

θ) Μονάδα διατρήσεως χαρτοταινίας (Tape punch).

Μπορεί να διατρυπά απλή ή πλαστικοποιημένη χαρτοταινία με τόν πενταψήφιο ή τριψήφιο κώδικα, με ταχύτητα μέχρι 80 χαρακτήρων το δευτερόλεπτο (σχ. 12.4η).



Σχ. 12.4η.
Μονάδα διατρήσεως χαρτοταινίας.

ι) Μονάδα έκτυπωσης (Printer).

Σχεδόν όλες οι μονάδες έκτυπωσης είναι τύπου dot matrix, δηλαδή άποτυπώνουν κάθε χαρακτήρα στο χαρτί, σχηματίζοντάς τον με τελείες (dots). Σε κάθε θέση έκτυπωσης υπάρχει μία όρθογωνική διάταξη από 7×5 ή 7×7 σύρματα [σχ. 12.4θ(α)], τα όποια, όταν ενεργοποιηθούν, κτυπούν στο χαρτί σχηματίζοντας κάποιο χαρακτήρα.

Μία τέτοια μονάδα είναι σχεδόν άθόρυβη και μπορεί να τυπώσει μέχρι 132 χαρακτήρες σε μία γραμμή του χαρτιού, με ταχύτητα μέχρι 200 γραμμές το λεπτό [σχ. 12.4θ(β)].

ια) Μονάδα σχεδιάσεως (Plotter).

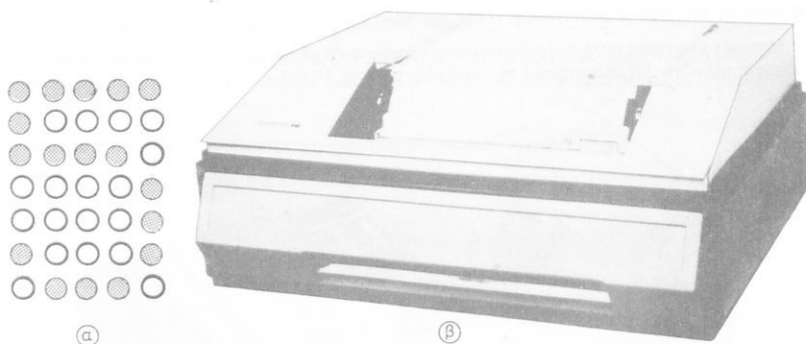
Μία τέτοια μονάδα μπορεί να σχεδιάζει διαγράμματα, καμπύλες κλπ. σε χαρτί, με διαστάσεις περίπου $14 \times 20''$ (σχ. 12.4ι).

ιβ) Όθονη (Display).

Χρησιμοποιείται όπως και στα μεγάλα συγκροτήματα υπολογιστών, σαν βασική περιφερειακή μονάδα, για την είσοδο δεδομένων και τη λήψη αποτελεσμάτων. Το σύνολο των εντολών των προγραμμάτων ή τμήματά τους μπορούν να απεικονισθούν στην όθονη. Πολλές όθονες είναι έφοδιασμένες με μνήμη 3K-5K, bytes (σχ. 12.4ια).

12.5 Ένα τυπικό συγκρότημα μίνι-υπολογιστή.

Ένα βασικό συγκρότημα μίνι-υπολογιστή, περιλαμβάνει μία μονάδα είσοδου-έξόδου, όπως το τη-



Σχ. 12.40.

(α) 'Απεικόνιση χαρακτήρα με το σύστημα dot - matrix. (β) Μονάδα έκτυπώσεως.

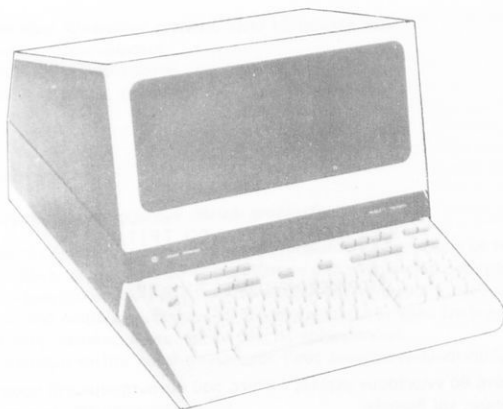


Σχ. 12.41.

Μονάδα σχεδιάσεως.

λέγυπο ή ή όθόνη, μία ή περισσότερες μικρές μονάδες μαγνητικής ταινίας σέ καρούλι ή σέ κασέτα ή σέ θήκη καί μία ή δύο μονάδες δισκέτας.

Όλα όμως τά συστήματα μίνι-ύπολογιστών χαρακτηρίζονται από μεγάλη έπεκτατικότητα (expandability), δηλαδή μπορούν νά συνδεθούν στο βασικό συγκρότημα καί άλλες περιφερειακές μονάδες διαφόρων τύπων καί λειτουργιών. Ό αριθμός των περιφερειακών μονάδων έξαρτάται από τίς ανάγκες αὐτῶν πού χρησιμοποιοῦν τό συγκρότημα (Users).



Σχ. 12.4ια.
Όθονη με πληκτρολόγιο.

12.6 Προγραμματισμός.

Συνήθως οι μίνι-υπολογιστές προγραμματίζονται σέ άπλοποιημένες έκδόσεις ύψηλών γλωσσών προγραμματισμού όπως ή FORTRAN, ή BASIC, ή RPG., έπειδή ή μνήμη πού διαθέτουν είναι περιορισμένη. Άλλοι μίνι-υπολογιστές προγραμματίζονται σέ είδικά σχεδιασμένες συμβολικές γλώσσες, όπως ή HPL κλπ.

12.7 Πλεονεκτήματα των μίνι-υπολογιστών.

Είναι πολλά τά πλεονεκτήματά τους, γι' αυτό καί τά τελευταία χρόνια ή προσοχή όλων των κατασκευαστριών εταιριών στράφηκε στό σχεδιασμό καί τήν κατασκευή τέτοιων συστημάτων.

Οι μίνι-υπολογιστές, είναι μικρού μεγέθους, χαμηλού κόστους άγοράς καί συντηρήσεως, μέ φθηνές περιφερειακές μονάδες καί πολύ φθηνούς φορείς πληροφοριών. Τό προσωπικό πού χρειάζεται γιά τό χειρισμό τους είναι έλάχιστο (1 άτομο, πού συνήθως είναι ό ίδιος ό προγραμματιστής). Ή ταχύτητα έπεξεργασίας τους είναι άρκετά ίκανοποιητική, μέ μεγάλες δυνατότητες προγραμματισμού καί εύρύ φάσμα έφαρμογών. Παρουσιάζουν έλάχιστες φθορές, πού είναι πολύ εύκολο νά διορθωθούν. Μερικοί τύποι μίνι-υπολογιστών έμφανίζονται τώρα καί μέ δυνατότητα πολυπρογραμματισμού (βλ. Κεφάλαιο 13).

Οι είδικοί προβλέπουν ότι οι μίνι-υπολογιστές όχι μόνο θά καλύψουν όλες τίς άνάγκες των έπιχειρήσεων, αλλά καί ότι στό μέλλον θά χρησιμοποιηθούν καί γιά τήν έξυπηρέτηση των διαφόρων οικιακών άναγκών.

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ ΤΡΙΤΟ

ΜΕΡΙΚΕΣ ΧΡΗΣΙΜΕΣ ΕΝΝΟΙΕΣ

13.1 Γενικά.

Στό κεφάλαιο αυτό θα γνωρίσουμε μερικές έννοιες, που συναντήσαμε στα προηγούμενα κεφάλαια και που είναι χρήσιμες και βασικές.

13.2 Τι ονομάζουμε Hardware και τί Software.

Τους όρους Hardware και Software τους ακούμε πολύ συχνά. Τι σημαίνουν όμως;

Hardware, είναι τό μηχανικό μέρος του υπολογιστή. Δηλαδή τά κυκλώματα, τά εξαρτήματα και οι μονάδες του.

Software, είναι όλα τά είδικά προγράμματα, μέ τά όποια είναι έφοδιασμένος ό υπολογιστής και τί όποια καθοδηγούν, έλέγχουν και συντονίζουν τή λειτουργία των διάφόρων τμημάτων του.

13.3 Έποπτεούν πρόγραμμα - Μεταφραστικά προγράμματα - Λειτουργικό σύστημα.

Έποπτεούν πρόγραμμα (Supervisory program ή απλώς Supervisor), είναι ένα πρόγραμμα, οι έντολές του όποίου κατευθύνουν τή λειτουργία του υπολογιστή αξιοποιώντας παράλληλα όλες τίς δυνατότητές του. Κατά τή διάρκεια τής λειτουργίας, τό έποπτεούν πρόγραμμα βρίσκεται άποθηκευμένο στη μνήμη.

Μεταφραστής ή μεταφραστικό πρόγραμμα (Compiler), είναι ένα πρόγραμμα που άναλαμβάνει, κάτω από τόν έλεγχο του έποπτεούντος προγράμματος.

- Νά εξετάσει τήν όρθότητα συντάξεως ενός προγράμματος έφαρμογής¹ που έχει γραφεί σε κάποια από τίς ύψηλου έπιπέδου γλώσσες προγραμματισμού,
- νά τό μεταφράσει στη γλώσσα μηχανής,
- νά έπίσημάνει τά συντακτικά λάθη του προγράμματος και νά δώσει σχετικά μηνύματα (διαγνωστικά),
- νά συνδέσει στο πρόγραμμα όλες τίς ρουτίνες² που θα χρειασθούν στην εκτέλεσή του. Έπολογιστής που προγραμματίζεται σε διάφορες γλώσσες, διαθέτουν ένα μεταφραστικό πρόγραμμα για κάθε γλώσσα.

Προγράμματα είδικής χρήσεως (Utilities). Είναι βοηθητικά προγράμματα, που εκτελούν διάφορες κοινές έργασίες που έμφανίζονται συχνά σε κάποια έπεξεργασία, όπως π.χ. ταξινόμηση ενός άρχείου (Sort), σύζευξη δύο άρχείων (Merge), μεταφορά δεδομένων από κάποια περιφερειακή μονάδα σε άλλο (π.χ. από ταινία σε ταινία, από ταινία σε δίσκο κ.ο.κ) κλπ.

¹ Στόν όρο πρόγραμμα έφαρμογής δίνουμε τήν έννοια που δίναμε στα προηγούμενα κεφάλαια. Έ προσθήκη τής λέξεως έφαρμογή γίνεται για νά διακριθεί ένα τέτοιο πρόγραμμα από ένα πρόγραμμα του Software.

² Μέ τόν όρο «ρουτίνα» έννοούμε ένα πρόγραμμα που χρησιμοποιείται για τήν εκτέλεση μιάς τυποποιημένης διαδικασίας.

Λειτουργικό σύστημα (Operating system), είναι το σύνολο των:

- α) Έποπτεύοντος προγράμματος.
- β) Μεταφραστικού προγράμματος.
- γ) Προγραμμάτων ειδικής χρήσεως.

13.4 Μέθοδοι έπεξεργασίας.

α) Σειριακή έπεξεργασία προγραμμάτων (Batch processing).

Έτσι ονομάζουμε την εκτέλεση των διαφόρων προγραμμάτων έφαρμογών με τη σειρά, δηλαδή τό ένα ύστερα από τό άλλο, με τη σειρά όπως διαβάζονται από την μονάδα αναγνώσεως δελτίων. Με τόν τρόπο αυτό έπεξεργάζονται τά προγράμματα έφαρμογής οι μικροί μεγέθους ύπολογιστές.

Κάθε πρόγραμμα πού διαβάζεται, άπασχολεί όλη την κεντρική μονάδα έπεξεργασίας καί δεσμεύει όλες τις περιφερειακές μονάδες, έστω καί άν δέν τις χρησιμοποιεί.

Πολλές φορές χρησιμοποιείται καί ό συνώνυμος όρος μονοπρογραμματισμός.

β) Πολυπρογραμματισμός (Multiprogramming).

Είναι μία τεχνική έπεξεργασίας προγραμμάτων έφαρμογών πού έμφανίζεται κυρίως στους με-
σαιούς καί μεγάλους μεγέθους ύπολογιστές, προκειμένου νά αξιοποιηθούν οι δυνατότητες της κεντρι-
κής μονάδας έπεξεργασίας καί νά άπασχοληθούν πλήρως όλες οι περιφερειακές μονάδες.

Σύμφωνα με τη μέθοδο αυτή, πολλά προγράμματα άποθηκεύονται συγχρόνως στόν ύπολογιστή πού αναλαμβάνει την ταυτόχρονη έπεξεργασία τους, έφ' όσον τά τμήματα καί οι μονάδες πού διαθέ-
τει, άνταποκρίνονται στις αντίστοιχες συνολικές άπαιτήσεις όλων των πρός έπεξεργασία προγραμμά-
των.

γ) Καταμερισμός του χρόνου έπεξεργασίας (Time sharing).

Η τεχνική αυτή διαφέρει από τόν πολυπρογραμματισμό κατά τό ότι διάφορα προγράμματα έφαρ-
μογών πού άποθηκεύονται στόν ύπολογιστή δέν έπεξεργάζονται ταυτόχρονα, αλλά έναλλακτικά
(δηλ. πότε τό ένα καί πότε τό άλλο, με πολύ μικρά ένδιάμεσα χρονικά διαστήματα) καί κατά διαδοχικά
βήματα.

Η τεχνική αυτή χρησιμοποιείται κυρίως σέ συστήματα πραγματικού χρόνου - πού θά δούμε στή
συνέχεια - όπου πολλοί Users, (όπως λέγονται όσοι χρησιμοποιούν τόν ηλεκτρονικό ύπολογιστή),
συνδέονται μέσω τερματικών σταθμών (Terminals) με τό κεντρικό συγκρότημά του, έξυπηρετούμε-
νοι ταυτόχρονα από αυτόν.

13.5 Συστήματα πραγματικού χρόνου (Real time systems).

Ό όρος αυτός δίνεται σέ συστήματα όπου ή έπεξεργασία των πληροφοριών γίνεται άμέσως με
την είσοδό τους στόν ύπολογιστή. Τά αποτελέσματα πού προκύπτουν έμφανίζονται ύστερα από τόσο
χρονικό διάστημα, όσο χρειάζεται ό ύπολογιστής γιά νά τά βρεί. Όλα τά απαιτούμενα άρχεία καί οι
περιφερειακές μονάδες είναι κάθε στιγμή διαθέσιμα γιά νά δώσουν τις διάφορες άπαντήσεις.

13.6 Τηλεέπεξεργασία (Teleprocessing).

Είναι ένα σύστημα έπεξεργασίας πληροφοριών σέ συνδυασμό μέ τηλεπικοινωνίες, με τη βοήθεια
των όποιων γίνεται ή μετάδοση καί λήψη πληροφοριών άνάμεσα σέ δύο σημεία πού απέχουν πολύ
μεταξύ τους.

Στά συστήματα αυτά ύπάρχει σέ κάποιο σημείο ένα κεντρικό συγκρότημα ύπολογιστή, ενώ στα ά-
πομακρυσμένα σημεία ύπάρχουν τερματικοί σταθμοί (Remote Terminals), πού συνδέονται με τό κεν-
τρικό συγκρότημα.

Τά συστήματα αυτά είναι γνωστά έπίσης καί με τά όνόματα Telecommunication systems, Data
communication systems, Teleprocessing systems κ.ο.κ.

13.7 Off - Line και On - Line λειτουργία του υπολογιστή.

Off - Line ονομάζουμε τόν τρόπο λειτουργίας του υπολογιστή, όπου ή διαδικασία για τήν έπεξεργασία τών πληροφοριών ακολουθεϊ τή σειρά: α) Συλλογή τών πληροφοριών, β) καταχώρησή τους σέ κάποιο φορέα, γ) έπεξεργασία τους από τόν υπολογιστή (όπως τό σύστημα τής μισθοδοσίας κλπ).

On - Line ονομάζουμε τόν τρόπο λειτουργίας του υπολογιστή, όπου ό πληροφορίες εισάγονται στόν υπολογιστή τή στιγμή πού δημιουργούνται (όπως τό σύστημα τών καταθέσεων στίς Τράπεζες, κρατήσεις θέσεων στίς Άεροπορικές Έταιρείες κλπ).

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ ΤΕΤΑΡΤΟ

ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

14.1 Γενικά.

Ἡ μεγάλη ἐξέλιξη τῶν ἠλεκτρονικῶν ὑπολογιστῶν, ὥστε νά ἔχουν μεγαλύτερες δυνατότητες ἀπό τίς παλιότερες ὑπολογιστικές συσκευές, εἶναι ἀποτέλεσμα τῆς ἀνάγκης γιά γρηγορότερη, καλύτερη καί φθηνότερη ἐπεξεργασία δεδομένων.

Οἱ ἐφαρμογές τοῦ ἠλεκτρονικοῦ ὑπολογιστῆ καλύπτουν σήμερα ὅλες τίς δραστηριότητες τῆς ἀνθρώπινης ζωῆς. Μερικές ἀπό αὐτές θά ἀναφερθοῦν στά ἐπόμενα.

14.2 Οἰκονομικά προβλήματα.

α) Μισθοδοσία.

Ἡ μισθοδοσία, περιλαμβάνει μία σειρά ἀπό ὑπολογισμούς γιά κάθε ὑπάλληλο, ὥστε νά βρεθοῦν οἱ ἀκαθάριστες ἀποδοχές του (βασικός μισθός, ὑπερωρίες, ἐπιδόματα κλπ.), οἱ κρατήσεις του (φόροι, κοινωνική ἀσφάλιση, δάνεια ἢ ἄλλες ὀφειλές πρὸς τό κράτος) καί οἱ καθαρές ἀποδοχές του. Τά οἰκονομικά αὐτά μεγέθη, πού ἀναφέρονται σέ κάθε ὑπάλληλο, μεταβάλλονται μέ τό χρόνο καί γι' αὐτό πρέπει νά γίνεται ἐνημέρωσή τους σέ τακτά χρονικά διαστήματα.

β) Παρακολούθηση λογαριασμῶν τῶν πελατῶν.

Τά μεγάλα καταστήματα, οἱ βιομηχανίες, οἱ τράπεζες καί γενικά ὅλες οἱ οἰκονομικές μονάδες, πού προσφέρουν ἀγαθὰ ἢ ὑπηρεσίες, πρέπει νά παρακολουθοῦν τοὺς λογαριασμούς τῶν πελατῶν τους. Οἱ τράπεζες π.χ. πρέπει νά ἔχουν μία ἐγγραφή γιά κάθε λογαριασμό πελάτη, πού νά τόν ἐλέγχουν καί νά τόν ἐνημερώνουν, κάθε φορά πού ὁ πελάτης κάνει μία κατάθεση ἢ ἀνάληψη.

γ) Τήρηση ἀποθηκῶν.

Ἡ ἐφαρμογή αὐτή, πού ἀνήκει στήν ἀπογραφική λογιστική, περιλαμβάνει τήν τήρηση ἐγγραφῶν γιά ὅλα τὰ ὑλικά, πού βρίσκονται στίς ἀποθήκες ἐνός καταστήματος ἢ ἐνός ἐργοστασίου. Οἱ ἐγγραφές αὐτές περιέχουν στοιχεῖα, ὅπως ἡ ὑπάρχουσα ποσότητα, ἡ τιμὴ μονάδας, ἡ ποσότητα ἀσφαλείας (ἐλάχιστη ποσότητα πού πρέπει νά ὑπάρχει στήν ἀποθήκη, πρὶν γίνει ἡ ἐπόμενη παραγγελία) κ.ἄ.

δ) Σχεδιασμός - προγραμματισμός τῆς παραγωγῆς.

Ὅταν πρόκειται νά ληφθεῖ κάποια ἀπόφαση γιά μεταβολὴ τῆς παραγωγικότητας

μιάς οικονομικής μονάδας, θά πρέπει νά γίνουν πολύπλοκες ἐκτιμήσεις, σχετικά μέ τό ἀνθρώπινο δυναμικό (ϊκανότητα, ἐκπαίδευση τῶν ὑπαλλήλων, πρόσληψη νέων ὑπαλλήλων, καθιέρωση ὑπερωριῶν, κινήτρων κλπ.) καί τόν ἀπαιτούμενο μηχανικό ἐξοπλισμό (αὔξηση τῆς λειτουργικότητας τοῦ ἤδη ὑπάρχοντος, ἀγορά νέου, ἐνοικίαση χώρων κλπ.) γιά τήν ἐπίτευξη τοῦ τελικοῦ σκοποῦ.

Οἱ ἐκτιμήσεις αὐτές καταγράφονται μαζί μέ τίς ἐκτιμήσεις στό χώρο τῆς ἀγορᾶς (ἀνταγωνισμός, γενική πολιτική τιμῶν, ἐξαγωγές, προβλέψεις βραχυπρόθεσμες καί μακροπρόθεσμες κλπ.) μέ τή μεγαλύτερη δυνατή ἀκρίβεια καί ἐπεξεργάζονται ἀπό τόν ὑπολογιστή μέ βάση εἰδικά σχεδιασμένα πρότυπα.

14.3 Προχωρημένα πληροφοριακά συστήματα.

Πολλά μηχανογραφικά συστήματα εἶναι ἀπλές μετατροπές χειροκινήτων συστημάτων (Manual Systems). Δέν εἶναι ὅμως λίγοι ἐκεῖνοι, πού ἀναγνωρίζουν τό γεγονός, ὅτι ὁ ὑπολογιστής προσθέτει νέες διαστάσεις καί ἀνοίγει νέους δρόμους στό σχεδιασμό πληροφοριακῶν συστημάτων. Μερικά ἀπό τά πιο συνηθισμένα ἐξελιγμένα πληροφοριακά συστήματα θά δοῦμε ἀμέσως.

α) Συστήματα ὁλοκληρωμένης ἐπεξεργασίας δεδομένων (IDP - Integrated Data Processing).

Στά συστήματα αὐτά ἡ συλλογή καί ἡ κωδικοποίηση τῶν δεδομένων σχεδιάζονται καί συντονίζονται ἔτσι ὥστε νά πετυχαίνεται τό καλύτερο δυνατό ἀποτέλεσμα μέ ταυτόχρονη ἐλαχιστοποίηση τοῦ πλήθους τῶν ἀπαιτουμένων λειτουργιῶν.

Σέ ἕνα τέτοιο σύστημα τά βασικά δεδομένα μεταγράφονται τή στιγμή πού γίνεται μιά ἐμπορική πράξη (ἢ μιά ἄλλη συναλλαγή), σέ κάποιο φορέα (χαρτοταινία, δελτία κλπ.), ὁ ὁποῖος στή συνέχεια θά εἰσαχθεῖ γιά παραπέρα ἐπεξεργασία στόν ὑπολογιστή.

β) Συστήματα πραγματικοῦ χρόνου (Real time - Systems).

Ἐνα τέτοιο σύστημα ἀποτελεῖται ἀπό ἕνα κεντρικό ὑπολογιστή, μέ τόν ὁποῖο συνδέεται ἕνας ἀριθμός ἀπό διατάξεις, μέσω τῶν ὁποίων στέλνονται ἀπ' εὐθείας δεδομένα στόν ὑπολογιστή καί λαμβάνονται ἀπαντήσεις ἀπό αὐτόν. Ἡ ὀνομασία «πραγματικός χρόνος» προέρχεται ἀπό τό γεγονός ὅτι τό χρονικό διάστημα, πού μεσολαβεῖ ἀνάμεσα στήν ὑποβολή τῆς ἐρωτήσεως καί στήν ἀπάντηση, εἶναι ἐλάχιστο καί ἴσο μέ τό χρόνο πού χρειάζεται ὁ ὑπολογιστής γιά νά τή δώσει.

Ὅλα τά συστήματα αὐτά εἶναι, ὅπως λέμε, ὄν-λαιν (On line), πού σημαίνει ὅτι ὅλα τά ἀπαιτούμενα ἀρχεῖα καί οἱ περιφερειακές μονάδες εἶναι κάθε στιγμή διαθέσιμα γιά νά δώσουν τίς διάφορες ἀπαντήσεις ἢ ἀποτελέσματα.

Στήν κατηγορία αὕτη ἀνήκει καί τό σύστημα γιά τίς κρατήσεις θέσεων καί ἐκδόσεως εἰσιτηριῶν στίς ἀεροπορικές ἐταιρίες.

14.4 Προβλήματα ἐπιστημονικοῦ προγραμματισμοῦ.

Τέτοια προβλήματα ἀπαιτοῦσαν στό παρελθόν πολύπλοκους ὑπολογισμούς μέ τό χέρι ἀπό ἐξειδικευμένο προσωπικό καί ἦταν ἐξαιρετικά δύσκολη ἡ ἐπίλυσή τους.

Σήμερα ο υπολογιστής μπορεί να επιλύει τα προβλήματα αυτά σε πολύ μικρούς χρόνους και χωρίς λάθη.

α) Γραμμικός προγραμματισμός.

Είναι μία μαθηματική μέθοδος, με τη βοήθεια της οποίας καθορίζουμε τις κατάλληλες συνθήκες, που πρέπει να ισχύουν σε ένα πρόβλημα, ώστε να επιτευχθεί το καλύτερο δυνατό αποτέλεσμα, όπως π.χ. η μεγιστοποίηση του κέρδους στην πώληση ενός προϊόντος ή η ελαχιστοποίηση του κόστους κ.ο.κ.

Σε όλες αυτές τις περιπτώσεις καταστρώνεται μία σειρά από εξισώσεις και ανισώσεις, που εκφράζουν τις επιθυμητές συνθήκες και που συγκροτούν μεγάλα γραμμικά άλγεβρικά συστήματα, των οποίων η επίλυση άλλοτε ήταν πολύ δύσκολη, γιατί γινόταν με τό χέρι. Σήμερα τέτοια συστήματα λύνονται εύκολα από τον υπολογιστή.

β) Χρονοπρογραμματισμός.

Ο σχεδιασμός των διαφόρων φάσεων μιας διαδικασίας είναι αρκετά πολύπλοκος, γιατί πρέπει να προβλεφθούν όλοι οι παράγοντες, που θα μπορούσαν να επηρεάσουν τη χρονική εξέλιξή τους.

Ειδικά προγράμματα υπολογιστών αναλύουν τα λειτουργικά βήματα μιας διαδικασίας, τα συσχετίζουν χρονικά και καθορίζουν τις προτεραιότητές τους και τους τρόπους χρονικής διασυνδέσής τους. Τέτοιες τεχνικές είναι γνωστές με διάφορα ονόματα όπως PERT, CPM κλπ.

γ) Έξομοίωση συστημάτων.

Ο όρος εξομοίωση συστήματος (Systems Simulation) σημαίνει ότι αντιγράφουμε ένα σύστημα, δημιουργώντας μια μαθηματική ή συμβολική αναπαράστασή του. Έτσι η μελέτη ενός πραγματικού συστήματος ανάγεται στη μελέτη της συμπεριφοράς της αναπαραστάσεώς του, που είναι και πιο εύκολη. Όσο μεγαλύτερη είναι η ομοιότητα του μοντέλου με το πραγματικό σύστημα, τόσο περισσότερο τα συμπεράσματα θα είναι ακριβή.

14.5 Αναζήτηση πληροφοριών.

Η μεγάλη πρόοδος στην τεχνολογία συνοδεύθηκε και από τεράστια συσσώρευση πληροφοριών. Μέρα με τη μέρα γίνεται και πιο δύσκολη η ενημέρωση ακόμη και ειδικών επιστημόνων επάνω στις εξελίξεις της επιστήμης τους. Σημαντική βοήθεια στο πρόβλημα αυτό προσφέρει ο ηλεκτρονικός υπολογιστής.

α) Αναζήτηση τεχνικών και επιστημονικών πληροφοριών.

Σε ένα τέτοιο σύστημα ο τίτλος ενός επιστημονικού κειμένου ή μιας εργασίας, εuryετηριάζεται κωδικοποιούμενος με τη βοήθεια καταλλήλων λέξεων-κλειδιών. Π.χ. ένα άρθρο που αναφέρεται στη διάβρωση των μετάλλων θα βρίσκεται και στη λέξη διάβρωση και στη λέξη μέταλλο. Συγχρόνως καταχωρίζεται και μία συνοπτική περίληψη του περιεχομένου.

β) Ιατρική διάγνωση.

Για τή σωστή διάγνωση μιᾶς ἀσθένειας, ὁ γιατρός πρέπει νά σημειώσει μέ ἀκρίβεια τὰ συμπτώματα πού παρουσιάζει ὁ ἀσθενής καί νά τά συγκρίνει μέ τὰ συμπτώματα πού παρουσιάζονται στίς διάφορες ἀσθενείες. Ἀπό τή σύγκριση ὁ γιατρός ἀποφαίνεται γιά τήν ἀσθένεια πού ἀνταποκρίνεται περισσότερο πρὸς τήν κατάσταση τοῦ ἀρρώστου.

Ἡ ἐπιβεβαίωση γίνεται μέ ἐργαστηριακές ἐξετάσεις, πού συμπληρώνουν τήν εἰκόνα τῆς καταστάσεως. Ἡ δουλειά αὕτη εἶναι ἐξαιρετικά δύσκολη καί λεπτή. Σήμερα ὁ ὑπολογιστής προσφέρει σημαντική βοήθεια στήν ἱατρική ἐπιστήμη, κυρίως στόν τομέα τῆς διαγνώσεως. Τά συμπτώματα ὅλων τῶν ἀσθενειῶν ἔχουν καταγραφῆ σέ εἰδικά ἀρχεῖα. Ὁ ὑπολογιστής τροφοδοτεῖται μέ τὰ συμπτώματα, πού παρουσιάζει ὁ ἀσθενής, κάνει τίς συγκρίσεις καί βγάζει συμπεράσματα, ἐντοπίζοντας τήν ἐργαστηριακή ἔρευνα σέ στενότερα ὅρια.

γ) Μετάφραση γλώσσων.

Ὁ ὑπολογιστής χρησιμοποιεῖται ἀκόμη, γιά τή μετάφραση κειμένων ἀπό μία γλώσσα σέ ἄλλη. Τό σύστημα λειτουργεῖ κυρίως γιά ἐπιστημονικά καί τεχνικά κείμενα, ὅπου χρησιμοποιεῖται καθορισμένη ὀρολογία καί ὅπου δέν ὑπάρχουν ἰδιωματισμοί τῆς γλώσσας.

Καθὼς μία γλώσσα ἀποτελεῖται ἀπό λέξεις, πού συγκροτοῦνται σέ φράσεις μέ τή βοήθεια ὀρισμένων κανόνων, τό εἰδικό πρόγραμμα τοῦ ὑπολογιστῆ προβαίνει στή μετάφραση, βρίσκοντας τίς ἀντίστοιχες λέξεις μιᾶς ἄλλης γλώσσας καί συνδυάζοντάς τις σύμφωνα μέ μερικούς ἀπλοῦς κανόνες.

14.6 Ἐκπαίδευση.

Εἰδικά προγράμματα ὑπολογιστῆ ἐξυπηρετοῦν διάφορους τομεῖς τῆς ἐκπαιδεύσεως.

α) Προγραμματισμένη διδασκαλία.

Αὕτη ἡ μορφή τῆς διδασκαλίας περιλαμβάνει μικρά διδακτικά κείμενα, πού ἡ κατανόησή τους ἐλέγχεται μέ εἰδική δοκιμασία ἀπό τόν ὑπολογιστή. Ἄν ὁ σπουδαστής ἀνταποκριθεῖ ἱκανοποιητικά στήν ἐξέταση, προχωρεῖ στά ἐπόμενα, ἂν ὄχι, τοῦ ἐπισημαίνονται οἱ ἀδυναμίες, γιά νά μπορέσει νά προετοιμασθεῖ κατάλληλα γιά μιά ἐπανεξέτασή του.

β) Διδασκαλία γιά τυφλοὺς.

Καταργώντας τό πολυδάπανο σύστημα γραφῆς Braille, ἡ διδασκαλία μέ ὑπολογιστῆ εἶναι πολύ πιό ἄνετη. Μέ τή βοήθεια ἐνός ὀπτικοῦ ἀναγνώστη καλύπτεται τό κείμενο ἐνός βιβλίου καί μεταφέρεται σέ ἓνα ὑπολογιστή, πού διαθέτει μονάδα ἡχητικῆς ἀποκρίσεως (Audio Response Unit). Οἱ λέξεις τοῦ κειμένου ἐκφωνοῦνται στή συνέχεια ἀπό megάφωνο.

14.7 Καθοδήγηση - Έλεγχος.

Ο υπολογιστής χρησιμοποιείται επίσης και σε συστήματα κατευθύνσεως και έλεγχου.

α) Αυτόματη καθοδήγηση βλημάτων.

Τα μεγάλα και πολυσύνθετα διηπειρωτικά βλήματα διαθέτουν μικροσκοπικούς, αλλά ισχυρούς υπολογιστές, στο τμήμα κατευθύνσεώς τους. Οι υπολογιστές αυτοί έχουν προγραμματισθεί κατάλληλα, ώστε να δέχονται πληροφορίες, που αφορούν τη ταχύτητα, τη θέση κλπ. του βλήματος και να δημιουργούν πληροφορίες υπό μορφή σημάτων, που μεταβιβάζονται στο σύστημα κατευθύνσεως, για τη διόρθωση της πορείας, όταν απαιτείται.

β) Ρύθμιση όδικης κυκλοφορίας.

Το έργο του έλεγχου της κυκλοφορίας σε ένα μεγάλο αστικό κέντρο μπορεί να ανατεθεί σε ηλεκτρονικό υπολογιστή ο οποίος θα ρυθμίζει τα φώτα της τροχαίας, το κλείσιμο ή άνοιγμα των ανισοπέδων διαβάσεων και κόμβων και θα κατευθύνει τους οδηγούς σε δρόμους με λιγότερη κυκλοφορία.

γ) Έλεγχος έναέριας κυκλοφορίας.

Ο έλεγχος της κινήσεως ενός αεροδρομίου και ο καθορισμός των πτήσεων και των δρομολογίων γίνεται με τη βοήθεια του υπολογιστή.

δ) Έλεγχος παραγωγής.

Σε περιπτώσεις παραγωγής βιομηχανικών προϊόντων εν σειρά, ο έλεγχος της ποιότητας ανατίθεται συνήθως σε υπολογιστή. Έτσι π.χ. στην παραγωγή χημικών προϊόντων, τοποθετούνται σε κατάλληλα σημεία της παραγωγικής διαδικασίας, συσκευές για μετρήσεις, που αφορούν στη σύνθεση των προϊόντων, στο βάρος, στη σωστή συσκευασία και στην καλή λειτουργία των μηχανημάτων. Οι πληροφορίες αυτές εισάγονται στον υπολογιστή, που τις συγκρίνει με τα πρότυπα (Standards) που έχουν τεθεί. Αν οι μετρήσεις δείξουν ότι η παραγωγική διαδικασία έχει αποκλίνει από τα επιθυμητά όρια, η λειτουργία των μηχανημάτων διορθώνεται ή μεταβάλλεται.

14.8 Ο υπολογιστής και οι τέχνες.

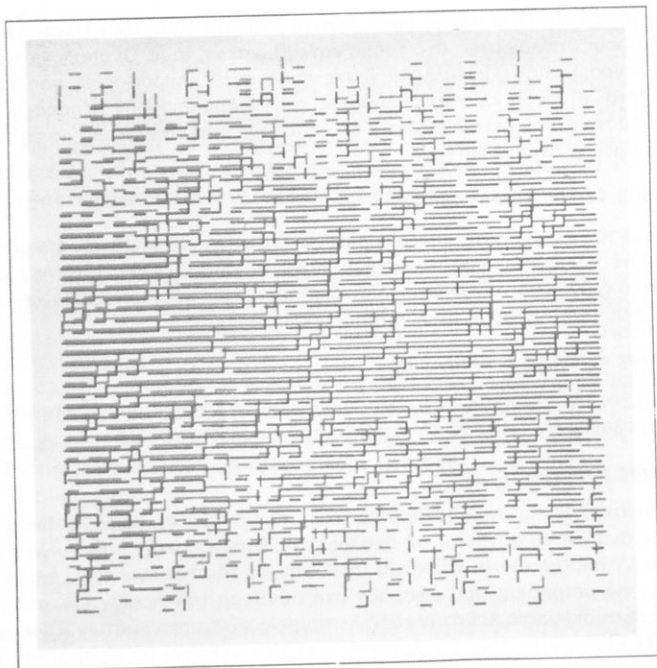
Η χρήση του υπολογιστή στις διάφορες μορφές τέχνης ξεκίνησε στην αρχή σαν παιχνίδι, αργότερα σαν πειραματισμός και τελικά εξελίχθηκε σε ολοκληρωτή επιστήμη.

α) Μουσική με υπολογιστή.

Όταν μία μαγνητική περιφερειακή μονάδα συνδεθεί κατάλληλα μέσω ενισχυτικών διατάξεων με μεγάφωνο, τότε ακούμε ήχους διαφόρων συχνοτήτων. Είναι λοιπόν δυνατό να γραφεί ένα ειδικό πρόγραμμα που η εκτέλεσή του να αποδίδει ένα μουσικό κομμάτι.

β) Ζωγραφική με ύπολογιστή.

Η ζωγραφική με ύπολογιστή ξεκίνησε στα 1952, αλλά η συστηματική έρευνα άρχισε στα 1957. Σήμερα έχει φθάσει σε ύψηλο σημείο εξέλιξης. Ένα τέτοιο δείγμα βλέπομε στο σχήμα που ακολουθεί (σχ. 14.8).



Σχ. 14.8.

Ζωγραφική με ύπολογιστή.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

Είσαγωγή στις 'Αριθμομηχανές και στους ψηφιακούς ηλεκτρονικούς υπολογιστές — 'Ιστορική αναδρομή

1.1	Ἡ ἀρίθμηση στους ἀρχαίους λαούς	1
1.2	Ἡ ἐμφάνιση τῶν πρώτων υπολογιστικῶν ὀργάνων	1
1.3	Οἱ πρώτες ἀριθμομηχανές	3
1.4	Ἡ ἰδέα τῆς πρώτης αὐτόματης υπολογιστικῆς μηχανῆς	4
1.5	Ἡ ἀνακάλυψη τῆς διατήρητης καρτέλας	5
1.6	Ὁ πρῶτος αὐτόματος υπολογιστής	6
1.7	Ρίχνοντας μιά ματιά στά τελευταῖα ἐπιτεύγματα τῆς ἐπιστήμης	9
1.8	Ὁ ηλεκτρονικός υπολογιστής θά ὑποκαταστήσει ἐντελῶς τόν ἄνθρωπο;	9
1.9	Κατηγορίες υπολογιστικῶν μηχανῶν	10

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

'Αριθμητικά συστήματα — Ἡ ἀριθμητική τῶν ψηφιακῶν ηλεκτρονικῶν υπολογιστῶν

2.1	Γενικά	13
2.2	Τό δεκαδικό ἀριθμητικό σύστημα	13
2.3	Τό δυαδικό ἀριθμητικό σύστημα	14
2.4	Τό ὀκταδικό ἀριθμητικό σύστημα	14
2.5	Τό δεκαεξαδικό ἀριθμητικό σύστημα	14
2.6	Μετατροπὴ ἀριθμῶν ἀπὸ ἓνα σύστημα σὲ κάποιο ἄλλο μὲ διαφορετικὴ βάση	15
2.7	'Αριθμητικὲς πράξεις στό δυαδικό σύστημα	18
2.8	'Αριθμητικὲς πράξεις στό ὀκταδικό σύστημα	19
2.9	Δυαδικοὶ κώδικες	20
2.10	Δυαδικὰ ψηφία ἰσοτιμίας	25

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

Περιγραφή τῶν ψηφιακῶν ηλεκτρονικῶν υπολογιστῶν I

3.1	Γενικά	27
3.2	Τὰ βασικά μέρη ἐνὸς ψηφιακοῦ ηλεκτρονικοῦ υπολογιστῆ	28
3.3	Φορεῖς καὶ μέσα καταγραφῆς πληροφοριῶν — Περιφερειακὲς μονάδες	29
3.4	Τό δελτίο (Card)	30
3.5	Μερικὲς χρήσιμες ἔννοιες	36
3.6	Κλασσικὲς μηχανές	39
3.7	Περιφερειακὴ μονάδα ἀναγνώσεως διατρήτων δελτίων (Card reader)	46
3.8	Περιφερειακὴ μονάδα διατρήσεως δελτίων (Card punch)	48
3.9	Ἡ χαρτοταινία (Papertape)	48

3.10 Μηχανή διατρήσεως χαρτοταινίας (Papertape punching machine)	49
3.11 Περιφερειακή μονάδα αναγνώσεως χαρτοταινίας (Punched papertape reader)	49
3.12 Περιφερειακή μονάδα διατρήσεως χαρτοταινίας (Papertape punch unit)	49
3.13 Όπτικοί αναγνώστες (Optical characters readers)	50
3.14 Περιφερειακή μονάδα εκτύπωσης (Printer)	52
3.15 Περιφερειακή μονάδα σχεδίασης (Graph Plotter)	55
3.16 Γραφομηχανή επικοινωνίας (Input / output typewriter)	56
3.17 Μαγνητική ταινία (Magnetic tape)	57
3.18 Κωδικοποιητής μαγνητικής ταινίας (Magnetic tape encoder ή Key-to-tape machine)	60
3.19 Περιφερειακή μονάδα μαγνητικής ταινίας (Magnetic tape unit)	61
3.20 Μαγνητικός Δίσκος (Magnetic disk)	62
3.21 Κωδικοποιητής μαγνητικού δίσκου (Magnetic disk encoder or Key-to-disk machine)	65
3.22 Περιφερειακή μονάδα μαγνητικού δίσκου (Magnetic disk unit)	65
3.23 Μαγνητικό τύμπανο (Magnetic drum)	66
3.24 Περιφερειακή μονάδα μαγνητικού τυμπάνου (Magnetic drum unit)	67
3.25 Μαγνητικές κάρτες (Magnetic cards or strips) και μονάδα μαγνητικών καρτών (Date cell)	68
3.26 Έντυπα με χαρακτηριστικές μαγνητικής μελάνης και αναγνώστες χαρακτηριστικών μαγνητικής μελάνης (Magnetic ink characters readers ή M.I.C.R.)	70
3.27 Μονάδα προβολής σε οθόνη (Cathode ray tube, CRT ή Data display unit)	71
3.28 Περιφερειακή μονάδα ακουστικής αποκρίσεως (Audio response unit)	72
3.29 Πίνακας έλεγχου (Console)	72
3.30 Τερματικοί σταθμοί (Terminals)	73
3.31 Μερικές παρατηρήσεις	73

ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

Περιγραφή των ψηφιακών ηλεκτρονικών υπολογιστών II Η κεντρική μονάδα επεξεργασίας

4.1 Η κεντρική μνήμη — Γενικά	75
4.2 Μαγνητικοί πυρήνες (Magnetic Cores)	77
4.3 Ανάγωση της τιμής ενός πυρήνα	80
4.4 Έγγραφή μίας τιμής σε ένα πυρήνα	81
4.5 Μνήμες από ήμιαγωγούς	81
4.6 Άλλα είδη μνήμης	81
4.7 Οργάνωση της μνήμης	82
4.8 Καταχωριστές (Registers)	84
4.9 Φλίπ-Φλόπς (Flip-Flops)	85
4.10 Μονάδα έλεγχου (Control Unit)	85
4.11 Αριθμητική και λογική μονάδα (Arithmetic and logical unit)	85
4.12 Δίαυλοι (Channels)	85
4.13 Υπομονάδες έλεγχου των περιφερειακών (Controllers)	85

ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ

Προγραμματισμός

5.1 Έντολές — Πρόγραμμα — Γλώσσα υπολογιστή	87
5.2 Γλώσσα μηχανής και συμβολική γλώσσα	87
5.3 Γλώσσες υψηλού επιπέδου (High Level Languages)	89
5.4 Κώδικες λειτουργίας και διευθύνσεις	90
5.5 Μηχανισμός εκτέλεσης ενός προγράμματος	91
5.6 Κατηγορίες εντολών	93
5.7 Τι είναι αλγόριθμος	94

ΚΕΦΑΛΑΙΟ ΕΚΤΟ

Διάγραμμα ροής

6.1	Σύμβολα διαγράμματος ροής	95
6.2	Παραδείγματα σχεδιασμού διαγραμμάτων ροής	97
6.3	Μερικοί χρήσιμοι μαθηματικοί συμβολισμοί	104
6.4	Η διαδικασία του προγραμματισμού	105

ΚΕΦΑΛΑΙΟ ΕΒΔΟΜΟ

Προγραμματισμός σε γλώσσα μηχανής

7.1	Ένας υποθετικός εκπαιδευτικός υπολογιστής	107
7.2	Τό σύνολο περιγραφών των εντολών του TK1	108
7.3	Πώς λειτουργεί ο TK1	111
7.4	Έντυπα κωδικογραφήσεως	111
7.5	Προγραμματισμός στη γλώσσα του TK1	111
7.6	Έντολές αποφάσεων	114
7.7	Τρόπος χρησιμοποίησης μερικών ακόμα εντολών	115

ΚΕΦΑΛΑΙΟ ΟΓΔΩΟ

Γλώσσα προγραμματισμού Assembly

8.1	Γενικά	117
8.2	Διαδικασία προγραμματισμού στη γλώσσα Assembly	117
8.3	Έντυπο κωδικογραφήσεως Assembly (Assembly Coding Form)	117
8.4	Γενική περιγραφή του υπολογιστή που θα χρησιμοποιήσουμε	119
8.5	Κατηγορίες εντολών Assembly	119
8.6	Έντολές τύπου RR	120
8.7	Έντολές τύπου RX	123
8.8	Έντολές τύπου RS	124
8.9	Έντολές τύπου SI	125
8.10	Έντολές τύπου SS1	126
8.11	Έντολές τύπου SS2	127
8.12	Ψευδοεντολές (Pseudoinstructions)	127
8.13	Μακροεντολές (Macros)	128
8.14	Παραδείγματα προγραμμάτων Assembly	129

ΚΕΦΑΛΑΙΟ ΕΝΑΤΟ

Η γλώσσα προγραμματισμού BASIC

9.1	Γενικά	136
9.2	Βασικά στοιχεία της BASIC	135
9.3	Κατηγορίες εντολών BASIC	138
9.4	Τρόπος αναγραφής των εντολών BASIC	139
9.5	Έντολη αντικατάστασεως. Η εντολή LET	139
9.6	Έντολές εισόδου-εξόδου	141
9.7	Μερικές ακόμη εντολές	143
9.8	Μερικά απλά προγράμματα BASIC	143
9.9	Έντολές έλεγχου και διακλαδώσεως	144
9.10	Έντολές FOR-NEXT	148
9.11	Πίνακες (Arrays). Μεταβλητές με δείκτες	150
9.12	Έντολη DIM	151

9.13 Παραδείγματα	152
9.14 Έπεξεργασία άρχείων στη BASIC	152
9.15 Υποπρογράμματα (Subprograms)	154

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ

Ή γλώσσα προγραμματισμού FORTRAN

10.1 Γενικά	158
10.2 Γενικά χαρακτηριστικά της FORTRAN	158
10.3 Έκφράσεις FORTRAN (FORTRAN expressions)	160
10.4 Συναρτήσεις του συστήματος	162
10.5 Παρενθώσεις	163
10.6 Ίεραρχία στην εκτέλεση των πράξεων	163
10.7 Λογικές εκφράσεις	163
10.8 Κατηγορίες εντολών FORTRAN	165
10.9 Περιγραφή του έντυπου κωδικογραφήσεως FORTRAN (FORTRAN Coding Form)	165
10.10 Αριθμητική εντολή αντικαταστάσεως	165
10.11 Έντολές έλέγχου και διακλαδώσεως	167
10.12 Έντολη DO	171
10.13 Μερικές άπλές εφαρμογές	172
10.14 Έντολές εισόδου - έξόδου	173
10.15 Έντυπο σχεδιασμού έκτυπώσεων (Printer spacing chart)	175
10.16 FORMAT	176
10.17 Έπαναλαμβανόμενοι κωδικοί FORMAT	181
10.18 Δηλωτικές εντολές	182
10.19 Μερικές άκόμη εντολές	183
10.20 Ειδικές μορφές των εντολών READ και WRITE	183
10.21 Σχόλια (Comments)	183
10.22 Υποπρογράμματα (Subprograms)	184
10.23 Έντυπο άναγραφής των δεδομένων (Data form)	189
10.24 Έφαρμογές	189
10.25 Διαδικασία προετοιμασίας ενός προγράμματος FORTRAN	195

ΚΕΦΑΛΑΙΟ ΕΝΔΕΚΑΤΟ

Ύπολογιστικές μηχανές γραφείου - Άριθμομηχανές

11.1 Γενικά	199
11.2 Κλασσικές ύπολογιστικές μηχανές	200
11.3 Ήλεκτρονικές ύπολογιστικές μηχανές	203
11.4 Χειρισμός μιάς ήλεκτρονικής ύπολογιστικής μηχανής	209
11.5 Έφαρμογές - Παραδείγματα	218
11.6 Προγραμματισμός	220

ΚΕΦΑΛΑΙΟ ΔΩΔΕΚΑΤΟ

Μίνι-ύπολογιστές

12.1 Γενική ταξινόμηση των ήλεκτρονικών ύπολογιστών	226
12.2 Τί είναι ένας μίνι-ύπολογιστής	228
12.3 Τό μέγεθος της λέξεως (Word size)	231
12.4 Περιφερειακές μονάδες	231
12.5 Ένα τυπικό συγκρότημα μίνι-ύπολογιστή	235
12.6 Προγραμματισμός	237
12.7 Πλεονεκτήματα των μίνι-ύπολογιστών	237

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ ΤΡΙΤΟ

Μερικές χρήσιμες έννοιες

13.1	Γενικά	238
13.2	Τί ονομάζουμε Hardware καί τί Software	238
13.3	Έποπτεδον πρόγραμμα — Μεταφραστικά προγράμματα — Λειτουργικό σύστημα	238
13.4	Μέθοδοι έπεξεργασίας	239
13.5	Συστήματα πραγματικού χρόνου (Real time systems)	239
13.6	Τηλεπεξεργασία (Teleprocessing)	239

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ ΤΕΤΑΡΤΟ

Έφαρμογές τών ηλεκτρονικών υπολογιστών

14.1	Γενικά	241
14.2	Οικονομικά προβλήματα	241
14.3	Προχωρημένα πληροφοριακά συστήματα	242
14.4	Προβλήματα έπιστημονικού προγραμματισμού	242
14.5	Έναζήτηση πληροφοριών	243
14.6	Έκπαίδευση	244
14.7	Καθοδήγηση — Έλεγχος	245
14.8	Ο υπολογιστής καί οί τέχνες	245



0020558238

Υπολογιστής του G. W. Leibnitz

