



ΗΛΕΚΤΡΟΝΙΚΟΙ ΥΠΟΛΟΓΙΣΤΕΣ

Τηλέμαχου Ι. Καλβουρίδη

ΦΥΣΙΚΟΥ - ΠΡΟΓ/ΤΗ Η/Υ - ΒΟΗΘΟΥ Ε.Μ.Π.





1954

ΙΔΡΥΜΑ ΕΥΓΕΝΙΔΟΥ
ΧΡΥΣΟΥΝ ΜΕΤΑΛΛΙΟΝ ΑΚΑΔΗΜΙΑΣ ΑΘΗΝΩΝ

ΠΡΟΛΟΓΟΣ ΙΔΡΥΜΑΤΟΣ ΕΥΓΕΝΙΔΟΥ

Ο Ευγένιος Ευγενίδης, ο ιδρυτής και χορηγός του «Ιδρύματος Ευγενίδου», πολύ νωρίς προέβλεψε και σχημάτισε την πεποίθηση ότι η άρπα κατάρτιση των τεχνικών μας, σε συνδυασμό με την εθνική αγωγή, θα ήταν αναγκαίος και αποφασιστικός παράγων για την πρόοδο του Έθνους μας.

Την πεποίθησή του αυτή ο Ευγενίδης εκδήλωσε με τη γενναιόφρονα πράξη ευεργεσίας, να κληροδοτήσει σεβαστό ποσό για τη σύσταση Ιδρύματος, που θα είχε ως σκοπό να συμβάλλει στην τεχνική εκπαίδευση των νέων της Ελλάδας.

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

Από το 1956 έως σήμερα η συμβολή του Ιδρύματος στην τεχνική εκπαίδευση πραγματοποιείται με διάφορες δραστηριότητες. Όμως απ' αυτές η σημαντικότερη, που κρίθηκε από την αρχή ως πρώτης ανάγκης, είναι η έκδοση βιβλίων για τους μαθητές των Τεχνικών και Επαγγελματικών Σχολών και Λυκείων.

Μέχρι σήμερα, με τη συνεργασία με τα Υπουργεία Εθνικής Παιδείας και Θρησκευμάτων και Εμπορικής Ναυτιλίας, εκδόθηκαν εκατοντάδες τόμοι βιβλίων, που έχουν διατεθεί σε πολλά εκπαιδυμένα αντίτυπα. Τα βιβλία αυτά κάλυπταν ή καλύπτουν ανάγκες των Κατωτέρων και Μέσων Τεχνικών Σχολών του Υπ. Παιδείας, των Σχολών του Οργανισμού Απασχολήσεως Εργατικού Δυναμικού (ΟΑΕΔ), των Τεχνικών και Επαγγελματικών Λυκείων, των Τεχνικών Επαγγελματικών Σχολών και των Δημοσίων Σχολών Εμπορικού Ναυτικού.

Μοναδική φροντίδα του Ιδρύματος σ' αυτή την εκδοτική του προσπάθεια ήταν και είναι η συγγραφή και έκδοση βιβλίων ποιότητας, από όποιψη όχι μόνον επιστημονική, παιδαγωγική και γλωσσική, αλλά και ως προς την εμφάνιση, ώστε το βιβλίο να αγαπηθεί από τους μαθητές,

Για την επιστημονική και παιδαγωγική αρτιότητα των βιβλίων τα κείμενα υποβάλλονται σε πολλές επεξεργασίες και βελτιώνονται πριν από κάθε νέα έκδοση συμπληρούμενα καταλλήλως.

Ιδιαίτερη σημασία απέδωσε το Ίδρυμα από την αρχή στη γλωσσική διατύπωση των βιβλίων, γιατί πιστεύει ότι και τα τεχνικά βιβλία, όταν είναι γραμμένα σε γλώσσα σωστή και ομοιόμορφη αλλά και κατάλληλη για τη στάθμη των μαθητών, μπορούν να συμβάλλουν στη γλωσσική κατάρτιση των μαθητών.

Έτσι, με απόφαση που ίσχυσε ήδη από το 1956, όλα τα βιβλία της Βιβλιοθήκης του Τεχνίτη, δηλαδή τα βιβλία για τις τότε Κατώτερες Τεχνικές Σχολές, όπως αργότερα και για τις Σχολές του ΟΑΕΔ, ήταν γραμμένα σε γλώσσα δημοτική, με βάση τη γραμματική του Τριανταφυλλίδη, ενώ όλα τα άλλα βιβλία ήταν γραμμένα στην απλή καθαρεύουσα. Σήμερα ακαλούθεται η γραμματική που διδάσκεται στα σχολεία της δευτεροβάθμιας εκπαίδευσης. Η γλωσσική επεξεργασία των βιβλίων αναπτύθε-

ται σε φιλολόγους του Ιδρύματος και έτσι εξασφαλίζεται η εναία σύνταξη και ορολογία κάθε κατηγορίας βιβλίων.

Η ποιότητα του χαρτιού, το είδος των τυπογραφικών στοιχείων, τα σωστά σχήματα, η καλαίσθητη σελιδοποίηση, το εξώφυλλο και το μέγεθος του βιβλίου, περιλαμβάνονται και αυτά στις φροντίδες του Ιδρύματος και συμβάλλουν στη σωστή «λειτουργικότητα» των βιβλίων.

Το Ίδρυμα θεώρησε ότι είναι υποχρέωσή του, σύμφωνα με το πνεύμα του ιδρυτή του, να θέση στη διάθεση του Κράτους όλη αυτή την πείρα του των 20 ετών, αναλαμβάνοντας το 1978 και την έκδοση των βιβλίων για τις νέες Τεχνικές Επαγγελματικές Σχολές και τα Τεχνικά και Επαγγελματικά Λύκεια, σύμφωνα πάντοτε με τα εγκεκριμένα Αναλυτικά Προγράμματα του Π.Ι. και του ΥΠΕΠΘ.

ΕΠΙΤΡΟΠΗ ΕΚΔΟΣΕΩΝ ΙΔΡΥΜΑΤΟΣ ΕΥΓΕΝΙΔΟΥ

Μιχαήλ Αγγελόπουλος, καθηγητής ΕΜΠ, Πρόεδρος.

Αλέξανδρος Σταυρόπουλος, καθηγητής Πανεπιστημίου Πειραιώς, Αντιπρόεδρος.

Ιωάννης Τεγόπουλος, καθηγητής ΕΜΠ.

Σταύρος Παλαιοκρασσός, Σύμβουλος - Αντιπρόεδρος Παιδαγωγικού Ινστιτούτου.

Εμμανουήλ Τρανούδης, Δικτής Σπ. Δευτ. Εκπαιδεύσεως ΥΠΕΠΘ.

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

Γραμματέας της Επιτροπής, Γεώργιος Ανδρέακος.

Διατελέσαντα μέλη ή σύμβουλοι της Επιτροπής

Γεώργιος Κακριδής (1955-1959) Καθηγητής ΕΜΠ, Άγγελος Καλογερός (1957-1970) Καθηγητής ΕΜΠ, Δημήτριος Νιάνιας (1957-1965) Καθηγητής ΕΜΠ, Μιχαήλ Σπετσιέρης (1956-1959), Νικόλαος Βασιώτης (1960-1967), Θεόδωρος Κουζέλης (1968-1976) Μηχ. Ηλ. ΕΜΠ, Παναγιώτης Χατζηιωάννου (1977-1982) Μηχ. Ηλ. ΕΜΠ, Αλέξανδρος Ι. Παππάς (1955-1983) Καθηγητής ΕΜΠ, Χριστόστομος Καρουνίδης (1955-1984) Μηχ. Ηλ. ΕΜΠ, Γεώργιος Ραύσος (1970-1987) Χημ.-Μηχ. ΕΜΠ, Θεοδόσιος Παπαθεοδοσίου (1982-1984) Δρ. Μηχανολόγος-Μηχανικός, Ιγνάτιος Χατζηευστρατίου (1985-1988) Μηχανολόγος, Γεν. Διευθυντής Σιβίτανιδείου Σχολής, Γεώργιος Σταματίου (1988-1990) Σχολ. σύμβουλος.



ΗΛΕΚΤΡΟΝΙΚΟΙ ΥΠΟΛΟΓΙΣΤΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΑΡΙΘΜΟΜΗΧΑΝΕΣ

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

ΑΘΗΝΑ
1994



ΠΡΟΛΟΓΟΣ

Το βιβλίο αυτό περιέχει δλη την ύλη που καθορίζει το αναλυτικό πρόγραμμα του Υπουργείου Παιδείας για το μάθημα «Αριθμομηχανές - Προγραμματισμός - Ηλεκτρονικοί υπολογιστές». Το κείμενο διαιρείται σε 14 κεφάλαια. Τα κεφάλαια 3 και 4 καθώς και τα 11,12 είναι περιγραφικά και δεν χρειάζεται ο διδασκόμενος να αποστηθεί τα αριθμητικά μεγέθη που αναφέρονται σε αυτά και που είναι απλώς ενδεικτικά. Όσα κεφάλαια ή παράγραφοι έχουν γραφεί με μικρά στοιχεία καλό θα είναι να παραλειφθούν σε πρώτη φάση και να διδαχθούν στο τέλος, αν υπάρχει φυσικά χρόνος. Κρίθηκαν απαραίτητα για να έχει ο μαθητής μια αλοκληρωμένη γνώση του γνωστικού αντικειμένου που ζητάει το αναλυτικό πρόγραμμα. Κατά τη διδασκαλία πρέπει να δοθεί ιδιαίτερη προσοχή και να αφιερωθεί δρο γίνεται περισσότερος χρόνος στα κεφάλαια 6, 9 και 10, που αναφέρονται στο διάγραμμα ροής και στον προγραμματισμό σε γλώσσες BASIC και FORTRAN.

Στο τέλος των περισσότερων κεφαλαίων υπάρχουν ορισμένες ασκήσεις. Ο διάσκων μπορεί αν θέλει να προτείνει και δικές του προσαρμοσμένες δύο είναι δυνατό στο πνεύμα του βιβλίου.

Γνωρίζω πως οι διστοκλίες που θα παρουσιασθούν στη διδασκαλία ενός μαθήματος που για πρώτη φορά θα ακούσουν οι μαθητές, όπως αυτό, θα είναι αρκετές. Πιστεύω όμως ότι με τη συνεργασία διδασκόντων και διδασκομένων θα υπερπηδηθούν.

Τελειώνοντας θέλω να εκφράσω τις ευχαριστίες μου στους συνεργάτες του Ιδρύματος που μόχθησαν τόσο πολύ για την κατά το δυνατόν καλύτερη εμφάνιση του βιβλίου.

Ο συγγραφέας

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΑΡΙΘΜΟΜΗΧΑΝΕΣ ΚΑΙ ΣΤΟΥΣ ΨΗΦΙΑΚΟΥΣ ΗΛΕΚΤΡΟΝΙΚΟΥΣ ΥΠΟΛΟΓΙΣΤΕΣ – ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

1.1 Η αριθμηση στους αρχαίους λαούς.

Από τη σπιγμή που δημιουργήθηκαν οι πρώτες οργανωμένες ομάδες και άρχισαν οι άνθρωποι να συναλλάσσονται μεταξύ τους δημιουργήθηκε και η ανάγκη για την εκτέλεση διαφόρων υπολογισμών.

Στην αρχή οι άνθρωποι για τη μέτρηση χρησιμοποίησαν τα μέλη του σώματός τους, όπως τα δάκτυλα των χεριών και των ποδιών, τους αγκώνες και τους ώμους τους. Ακόμα και σήμερα σε μερικές φυλές, που εξακολουθούν να ζουν μακριά από τον πολιτισμό, χρησιμοποιείται το σύστημα αυτό μετρήσεως.

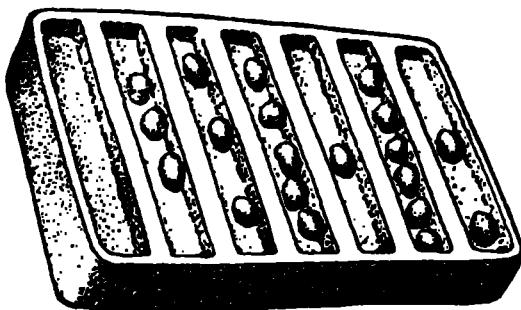
Με την πάροδο του χρόνου η οικονομία στις ανθρώπινες κοινωνίες έγινε πολυπλοκότερη και το μέτρημα με τα μέλη του ανθρωπίνου σώματος δεν έπαρκούσε. Ακριβώς τότε πρωτευματίσθηκαν και οι αριθμοί-σύνδρολα, που ο καθένας τους αντιπροσώπευε μια συγκεκριμένη ποσότητα από αντικέίμενα. Παράλληλα οι άνθρωποι χρησιμοποίησαν βότσαλα, σπόρους από σιτάρι ή καλαμπόκι, κόκκους άμμου, και δίλλα μέσα για να λογαριάσουν και να μετρήσουν.

Αργότερα έμφανισθηκαν και τα πρώτα αριθμητικά σύμβολα. Πολλοί λαοί έχουν να μάς παρουσιάσουν από την αρχαία ιστορία τους διαφορετικά συστήματα αριθμήσεως και απεικόνισεως των αριθμών.

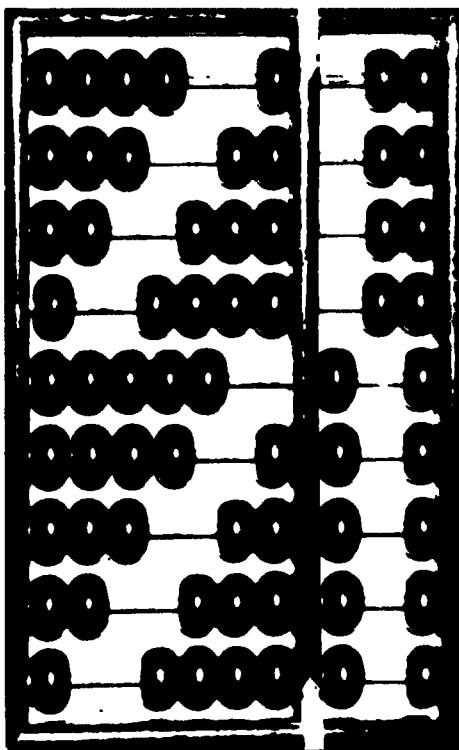
Σιγά-σιγά η ανάγκη για υπολογισμούς δυσκολότερους μεγάλωνε. Η ιστορία μας πληροφορεί ότι οι αρχαίοι Αιγύπτιοι έκαναν πολύπλοκους λογαριασμούς για να βρουν π.χ. τις ιδεώδεις διαστάσεις και για να σχεδιάσουν τους βασιλικούς τάφους, και άκριμα κρατούσαν λογιστικά βιβλία, όπου σημείωναν και παρακολουθούσαν τις εμπορικές συναλλαγές τους. Οι αρχαίοι Έλληνες έλυναν δύσκολα προβλήματα μαθηματικών και μηχανικής. Άλλα και οι Φοίνικες, οι Ασσύριοι, οι Βαβυλώνιοι, οι Πέρσες, οι Κινέζοι είχαν ασχοληθεί εντατικά με τα μαθηματικά.

1.2 Η εμφάνιση των πρώτων υπολογιστικών οργάνων.

Λέγεται ότι η πιο απλή υπολογιστική συσκευή, ο άβακας, που είναι ο μακρινός πρόγονος του ταπεινού και γνωστού μας αριθμηταρίου, ήταν γνωστός και τον χρησιμοποιούσαν στην αρχαία Ελλάδα και αργότερα στην αρχαία Ρώμη. Ο άβακας (σχ. 1.2a) ήταν μια πέτρινη πλάκα, που είχε επάνω σκαλισμένα μερικά αυλάκια. Μέσα σε κάθε αυλάκι τοποθετούσαν μικρές πέτρες που κάθε μιά αναπλήρωνε κάποιον αριθμό.



Σχ. 1.2α.
Ο Άβακας.



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

Στην Κίνα από το 3000 π.Χ. περίπου, εμφανίσθηκε μια λογιστική συσκευή, το «Σουάν-παν», που έμοιαζε πολύ με τον άβακα και το χρησιμοποιούσαν σχέδόν με τον ίδιο τρόπο. Μια αρκετά εξελιγμένη μορφή του «Σουάν-παν» βλέπομε στο σχήμα 1.2β. Η παληά πέτρινη πλάκα αντικαταστάθηκε από ένα πλαίσιο που ένας δια-

χωριστικός κανόνας το χώριζε σε δύο τμήματα. Καθένα από τα δύο σφαιρίδια, που βρίσκονταν στο δεξιό τμήμα, ισοδυναμούσε με 5 σφαιρίδια που βρίσκονταν στο αριστερό τμήμα. Ενώ καθένα από τα 5 σφαιρίδια του αριστερού τμήματος ισοδυναμούσε με μία μονάδα ή δεκάδα κλπ. Ή Γιαπωνέζικη συσκευή «Σαρομπάν» διέφερε από την Κινέζικη μόνο στο ότι άντι για δύο σφαιρίδια στο δεξιό τμήμα, είχε ένα μόνο.

Στο Μεσαίωνα, η εκτέλεση μιας πράξεως πολλαπλασιασμού εξακολουθούσε να αποτελεί έργο ειδικών. Στα 1617 ο Σκύτος μαθηματικός *Napier*¹ κατασκεύασε μια πρωτόγονη συσκευή, που με τη βοήθειά της μπορούσε να εκτελεί ένα πολλαπλασιασμό. Η συσκευή αυτή εγκαταλείφθηκε πολύ γρήγορα. Λίγο αργότερα, στα 1630 ο *Napier* ανακάλυψε τους λογαρίθμους και κατόπιν σχεδίασε τον πρώτο λογαριθμικό κανόνα, που είναι και το παληότερο αναλογικό υπολογιστικό δργανό.

1.3 Οι πρώτες αριθμομηχανές.

Η επινόηση και η κατασκευή των πρώτων αριθμομηχανών συνδέονται με τα ονόματα των *Pascal*² και *Leibniz*³. Η μηχανή του *Pascal* που πρωτοεμφανίσθηκε στα 1642 μπορούσε να εκτελεί προρθέσεις και αφαιρέσεις. Την αποτελούσαν (σχ. 1.3) μια σειρά τροχών με αριθμούς γύρω από τον καθένα. Ο πρώτος τροχός συμβόλιζε τις μονάδες, ο δεύτερος τις δεκάδες, ο τρίτος τις εκατοντάδες κ.ο.κ. Όταν ο πρώτος τροχός, που συμβόλιζε τις μονάδες, έκανε μια πλήρη περιστροφή, ο δεύτερος τροχός που συμβόλιζε τις δεκάδες στρεφόταν κατά ένα αριθμό, κλπ.

Η μηχανή του *Leibniz*, που εμφανίσθηκε λίγο αργότερα, στα 1673-74, αποτελούσε βελτιωμένη μορφή της μηχανής του *Pascal* και μπορούσε να εκτελεί όλες τις αριθμητικές πράξεις.

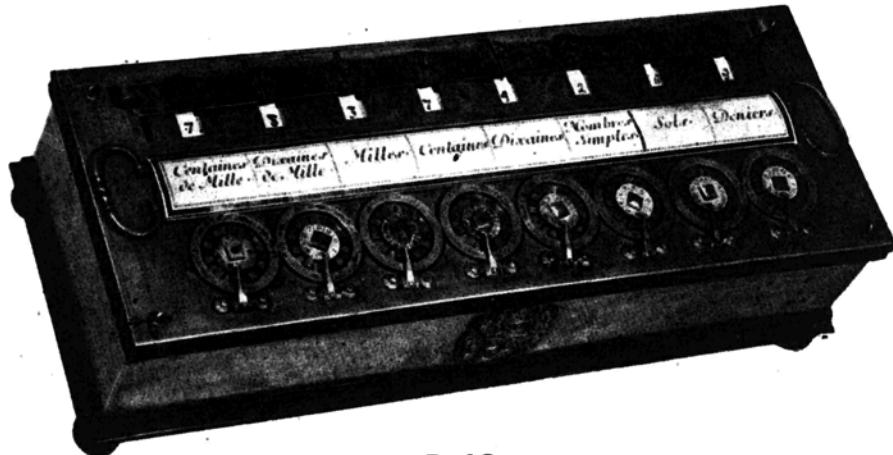
Από την εποχή αυτή και έπειτα εμφανίζονται πολλές βελτιωμένες διατάξεις, που όλες όμως στηρίζονται στις αρχικές ιδέες των *Pascal* και *Leibniz*.

Στα 1872 ο *Frank Baldwin* κατασκεύασε μια αριθμομηχανή που ένα χρόνο αργότερα άρχισε να την παράγει μαζικά. Η χρονιά εκείνη έστημανε την αρχή της βιομηχανικής παραγωγής των αριθμομηχανών. Στα 1890 ο *Burroughs* εφοδίασε τις

1 *John Napier*: Σκύτος μαθηματικός. Γεννήθηκε κοντά στο Εδιμβούργο το 1550 και πέθανε εκεί το 1617. Είναι ο παπέας των λογαρίθμων και του λογαριθμικού κανόνα. Είναι ο πρώτος που εισήγαγε το κλασματικό σημείο.

2 *Blaise Pascal*: Γάλλος μαθηματικός και φυσικός. Γεννήθηκε στη Clermont Ferrand της Γαλλίας το 1623 και πέθανε στο Παρίσι το 1662. Σε ηλικία 18 ετών δημοσίευσε ένα βιβλίο με μελέτες του πάνω στη Γεωμετρία των κωνικών τομών. Υπήρχε ένας από τους θεμελιώτες της σύγχρονης θεωρίας των Πιθανοτήτων. Μελέτησε ακόμη τη συμπεριφορά των ρευστών και διατύπωσε τη γνωστή αρχή για τη μετάδοση των πέσεων στα υγρά. Το επιστημονικό του έργο είναι τεράστιο.

3 *Gottfried Wilhelm Leibniz*: Γερμανός φιλόσοφος και μαθηματικός. Γεννήθηκε στη Λειψία το 1646 και πέθανε στο Αννβέρο το 1716. Ονομάσθηκε ο «Αριστοτέλης της εποχής του». Ήταν ο πρώτος που αντιλήφθηκε την αξία του δυαδικού συστήματος. Ανακάλυψε τις ορίζουσες και το 1693 έδωσε μια δική του διατύπωση για τη διατήρηση της μηχανικής ενέργειας. Ανακάλυψε ακόμη και κατασκεύασε το μεταλλικό βαρόμετρο. Στα 1700 εξέλεγε μαζί με το *Newton* μέλος της Παριστονής Ακαδημίας Επιστημών. Παρ' όλα το τεράστιο επιστημονικό του έργο, πέθανε αγνοημένος και ξεχασμένος από όλους.



Σχ. 1.3
Η μηχανή του *Pascal*.

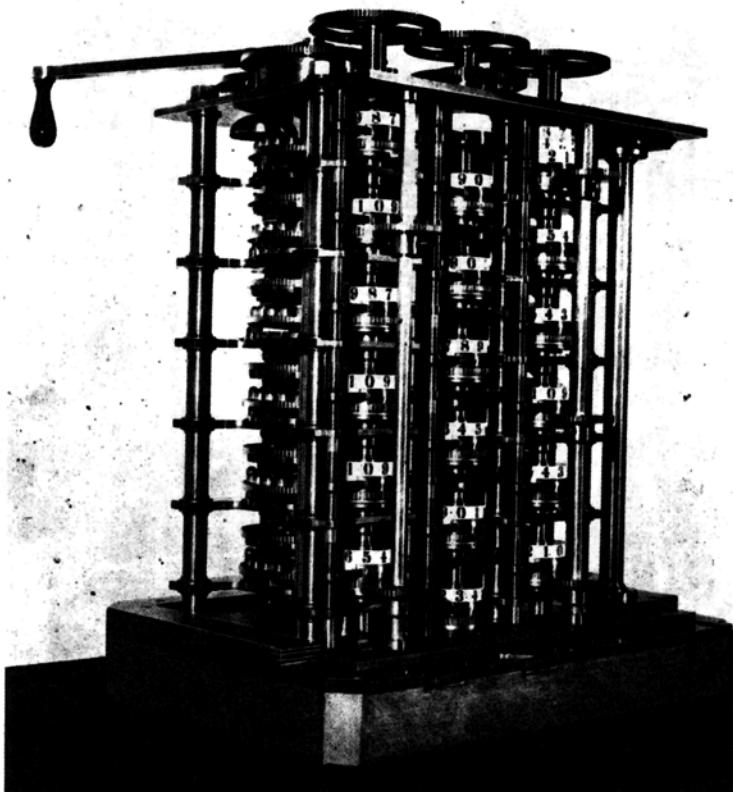
αριθμομηχανές με πληκτρολόγιο όμοιο με το σημερινό και με σύστημα έκτυπωσεώς των αριθμών και αποτελεσμάτων.

Όλες αυτές όμως οι μηχανές, ενώ θα μπορούσαν να θεωρηθούν πρόδρομοι των συγχρόνων ηλεκτρικών και πλεκτρονικών αριθμομηχανών, δεν θεωρούνται έτσι όσον αφορά στους ψηφιακούς ηλεκτρονικούς υπολογιστές. Η χρησιμοποίησή τους απαιτούσε τη συνεχή παρακολούθησή τους από το χειριστή και το συνεχή εφοδιασμό τους με στοιχεία για να λειτουργήσουν, οι δε υπολογισμοί γίνονταν κατά τρόπο διακεκομένο.

1.4 Η ιδέα της πρώτης αυτόματης υπολογιστικής μηχανής.

Η ιδέα της κατασκευής μιας αυτόματης μηχανής που θα μπορούσε όχι μόνο να εκτελεί διακεκριμένα τις τέσσερις αριθμητικές πράξεις, αλλά ακόμα και μια ολόκληρη συνεχή σειρά υπολογισμών, οφείλεται στο *Charles Babbage*¹, καθηγητή στο Πανεπιστήμιο του Cambridge στην Αγγλία. Ο *Babbage* σχεδίασε στα 1812 μια αυτόματη υπολογιστική μηχανή που την ονόμασε διαφορική μηχανή (σχ. 1.4a), γιατί σκόπευε να τη χρησιμοποιήσει στον υπολογισμό μαθηματικών πινάκων διαφορών. Ασχολήθηκε με την κατασκευή της περισσότερο από είκοσι χρόνια χωρίς να καταφέρει να την ολοκληρώσει, γιατί δεν τον βοηθούσε, σε πολλές περιπτώσεις η τεχνολογία της εποχής του. Ο *Babbage* εγκατέλειψε τη διαφορική μηχανή του, αλλά συνέχισε με αισιοδοξία τις έρευνες και τις μελέτες του. Αποτέλεσμα αυ-

¹ *Charles Babbage*: Άγγλος μαθηματικός. Γεννήθηκε στο Teignmouth τα 1792 και πέθανε στο Λονδίνο το 1871. Ήταν γιος τραπεζίτη. Διδάχθηκε μόνος του μαθηματικά και το 1810 εισήχθη στο Πανεπιστήμιο του Cambridge. Μετά την αποφότησή του ασχολήθηκε με μελέτες μαθηματικών. Ήταν όμως ανήσυχο πνεύμα και τήν ενεργητικότητα την διοχέτευσε και σ' άλλους επιστημονικούς κλάδους. Θεωρείται ένας από τους θεμελιώτες της Επιχειρησιακής Έρευνας. Στα 1847 ανακάλυψε το οφθαλμοσκόπιο, ενώ από το 1812 είχε αρχίσει να ασχολείται με τη μηχανοποίηση των υπολογισμών.



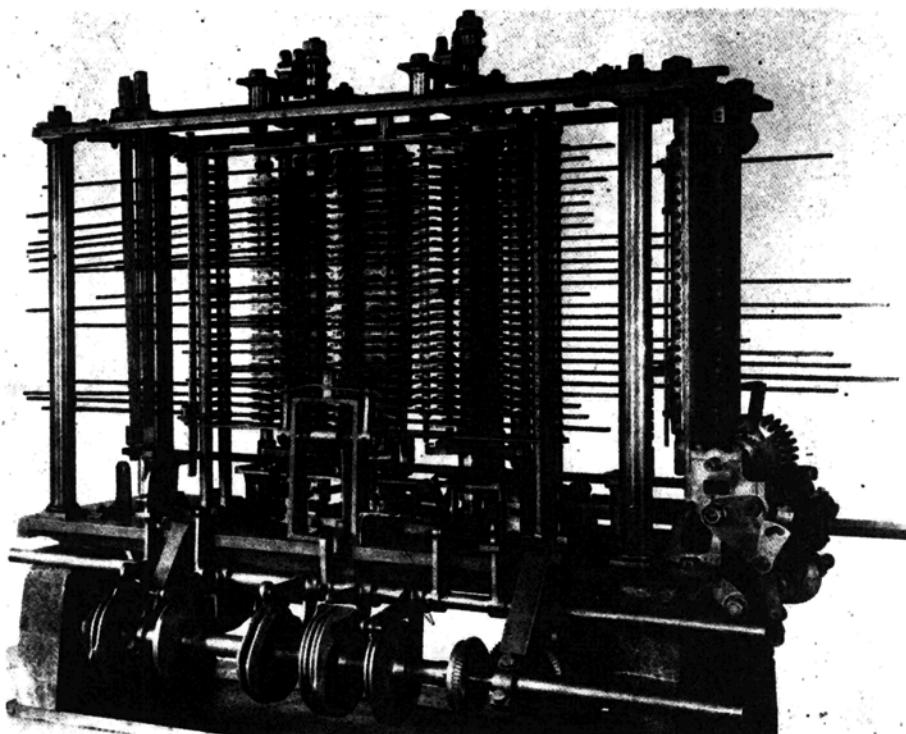
Σχ. 1.4α.
Η διαφορική μηχανή του *Babbage*.

της της προσπάθειας ήταν να σχεδιάσει μια νέα και πιο φιλόδοξη υπολογιστική μηχανή, που την ονόμασε αναλυτική μηχανή (σχ. 1.4β).

Τη μηχανή αυτή την αποτελούσαν τρία μέρη. Στο πρώτο, την «αποθήκη», αποταμιεύονταν πολλοί αριθμοί συγχρόνως. Στο δεύτερο, το «μύλο», εκτελούνταν οι αριθμητικές πράξεις και στο τρίτο βρίσκονταν οι «μηχανισμοί ακολουθιών», που επέλεγαν τους κατάλληλους αριθμούς από την «αποθήκη» και τους οδηγούσαν στο «μύλο» για να εκτελεσθούν στη συνέχεια οι προβλεπόμενες αριθμητικές πράξεις. Ούτε όμως και αυτή η μηχανή ολοκληρώθηκε παρ' όλο που μερικά από τα τμήματά της είχαν κατασκευασθεί. Ο *Babbage* πέθανε απογοητευμένος και τα σχέδιά του ξεχάσθηκαν για πολλά χρόνια.

1.5 Η ανακάλυψη της διάτρητης καρτέλας.

Μια ανακάλυψη που έγινε στα 1804 συνδέθηκε αργότερα με τη χρησιμοποίηση των ψηφιακών ηλεκτρονικών υπολογιστών. Πρόκειται για την ιδέα του Γάλλου



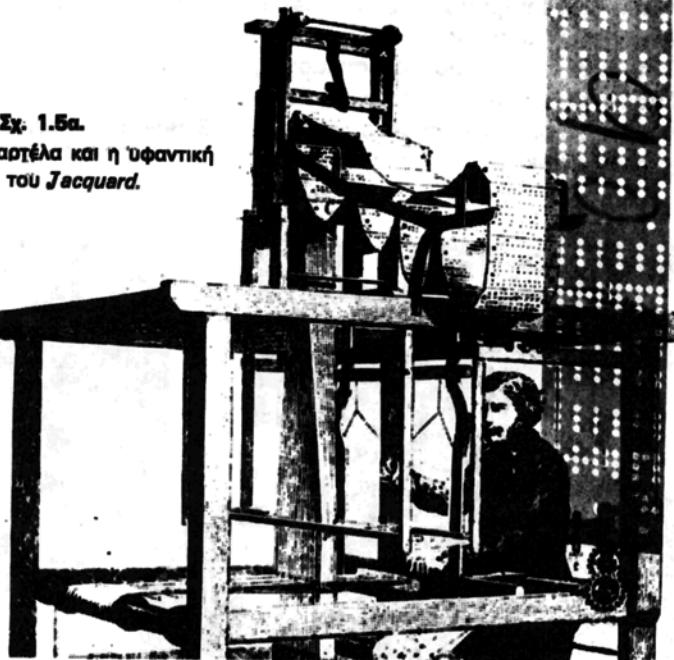
Σχ. 1.4β.
Η αναλυτική μηχανή του *Babbage*.

μηχανικού *Jacquard*, να χρησιμοποιήσει διάτρητες καρτέλες για να ελέγχει και να καθοδηγεί τα νήματα στις υφαντικές μηχανές (σχ. 1.5α).

Τις διάτρητες αυτές καρτέλες σκέφθηκε να χρησιμοποιήσει ο στατιστικολόγος *Herman Hollerith* στα 1890, για να διευκολύνει το έργο της μεγάλης απογραφής του πληθυσμού των Η.Π.Α. Σχεδίασε για το σκοπό αυτό μια μηχανή που τρυπούσε καρτέλες, τις επεξεργαζόταν και στη συνέχεια τις ταξινομούσε με τη βοήθεια ηλεκτρομηχανικών διατάξεων (σχ. 1.5β).

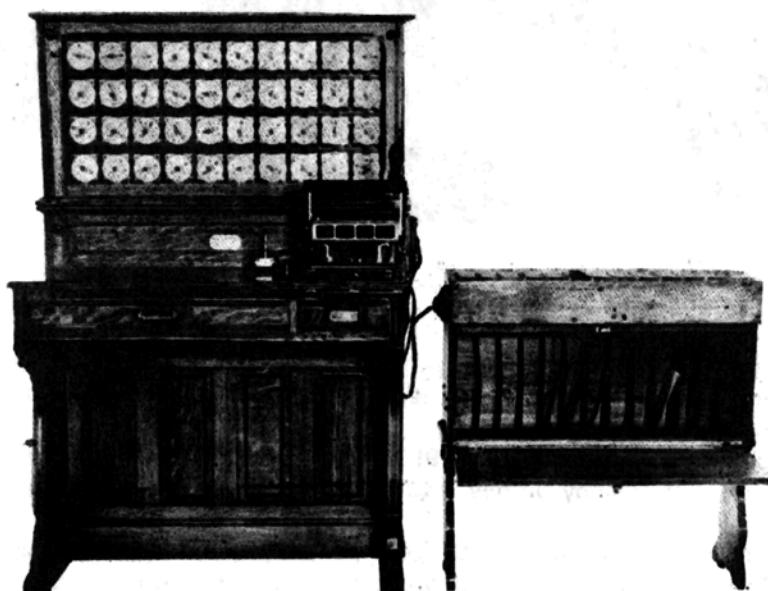
1.6 Ο πρώτος αυτόματος υπολογιστής.

Ο πρώτος αυτόματος ψηφιακός υπολογιστής κατασκευάσθηκε το 1944 στα εργαστήρια του Πανεπιστημίου του Harvard από μιά ομάδα επιστημόνων με επικεφαλής τον καθηγητή *Howard Aiken*. Ονομάσθηκε A.S.C.C (Automatic Sequence Controlled Calculator) και η κατασκευή του βασίσθηκε σε ηλεκτρομηχανικές αρχές και ειδικότερα στην τεχνική των ηλεκτρομηχανικών διακοπτών, που χρησιμοποιούνται στις τηλεφωνικές συνδέσεις (Relay Techniques). Μπορούσε να προσθέτει δύο 23ψήφιους δεκαδικούς αριθμούς σε χρόνο 1/3 s ή να τους πολλαπλασιάζει σε χρόνο 6 s. δια την κατασκευή του χρειάσθηκαν 750.000 εξαρτήματα και χρησιμοποιήθηκε για τις διάφορες συνδέσεις καλώδιο μήκους 500 μιλίων.



Σχ. 1.5α.

Η διάτρητη καρτέλα και η ύφαντική μηχανή του *Jacquard*.

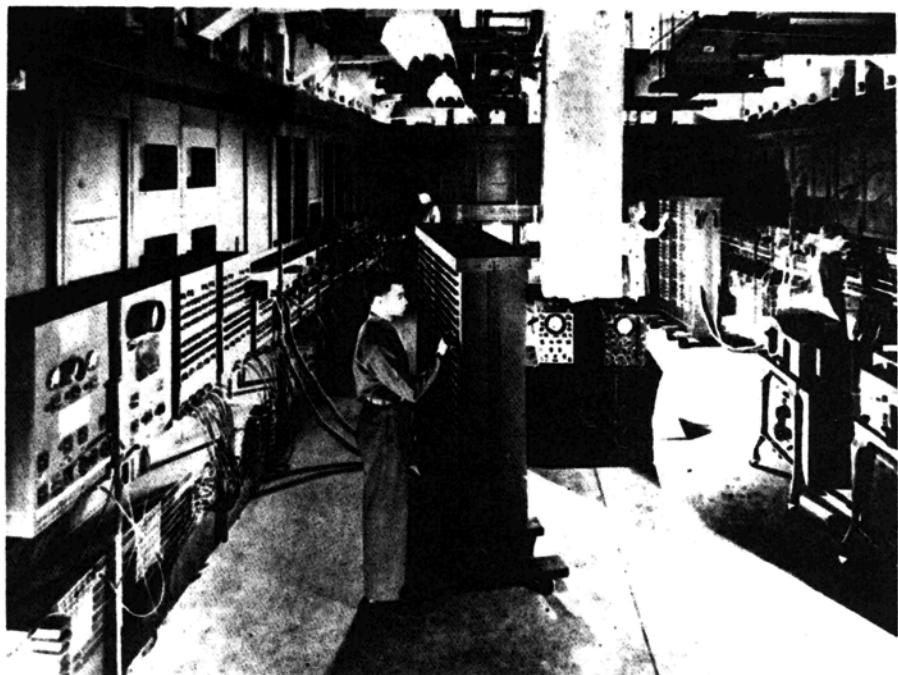


Σχ. 1.5β.

Η μηχανή του *Hollerith*.

Λίγο αργότερα, το 1946, μια άλλη επιστημονική ομάδα με επικεφάλης τον καθηγητή της Φυσικής J. Mauchly κατασκεύασε στο Πανεπιστήμιο της Πενσυλβανίας ένα αυτόματο ηλεκτρονικό ψηφιακό υπολογιστή, που τον ονόμασε E.N.I.A.C (Electronic Numerical Integrator and Calculator) (σχ. 1.6a). Συνδυασμός 18.000 ηλεκτρονικών λυχνιών αποτελούσε τον υπολογιστή αυτόν, που μπορούσε να εκτελεί 5000 προσθέσεις ή 500 πολλαπλασιασμούς στο δευτερόλεπτο.

Από το 1950 η εξέλιξη στην κατασκευή βελτιωμένων ψηφιακών ηλεκτρονικών υπολογιστών είναι χωρίς προηγούμενο και σε αυτό βέβαια συντελεί και η μεγάλη πρόοδος της ηλεκτρονικής.



Σχ. 1.6a.
Ο υπολογιστής E.N.I.A.C.

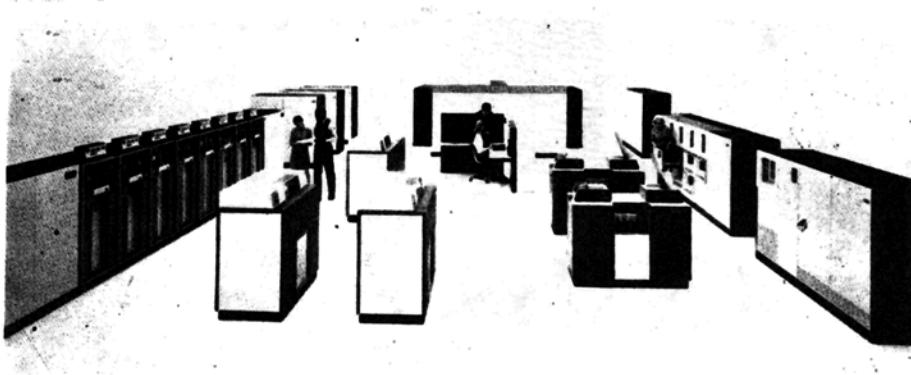
Η πρώτη γενιά των ψηφιακών ηλεκτρονικών υπολογιστών χαρακτηρίζεται από τη χρήση ηλεκτρονικών λυχνιών, το μεγάλο όγκο της κατασκευής και τη σχετικά μικρή ταχύτητα υπολογισμών.

Η δεύτερη γενιά, που αρχίζει το 1959, χρησιμοποιεί ημιαγωγούς, επιτυγχάνοντας έτσι τη μείωση του όγκου των μηχανών και την αύξηση της ταχύτητας υπολογισμών.

Η τρίτη γενιά αρχίζει από το 1965, χρησιμοποιεί μικροκυκλώματα και μονολιθικά ολοκληρωμένα κυκλώματα, με αποτέλεσμα να μειωθούν σημαντικά οι διαστάσεις τους και να αυξηθεί η ταχύτητα υπολογισμών, που φθάνει στην τάξη μεγέθους νανοδευτερολέπτου (1 δισεκατομμυριοστού του δευτερολέπτου). Εισάγεται

επίσης ή τεχνική του πολυπρογραμματισμού και αυξάνονται οι δυνατότητές των περιφερειακών μονάδων.

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



Σχ. 1.6β.

Ένα σύγχρονο συγκρότημα ψηφιακού ηλεκτρονικού υπολογιστού 4ης γενιάς.

1.7 Ρίχνοντας μια ματιά στα τελευταία επιτεύγματα της έποιστήμης.

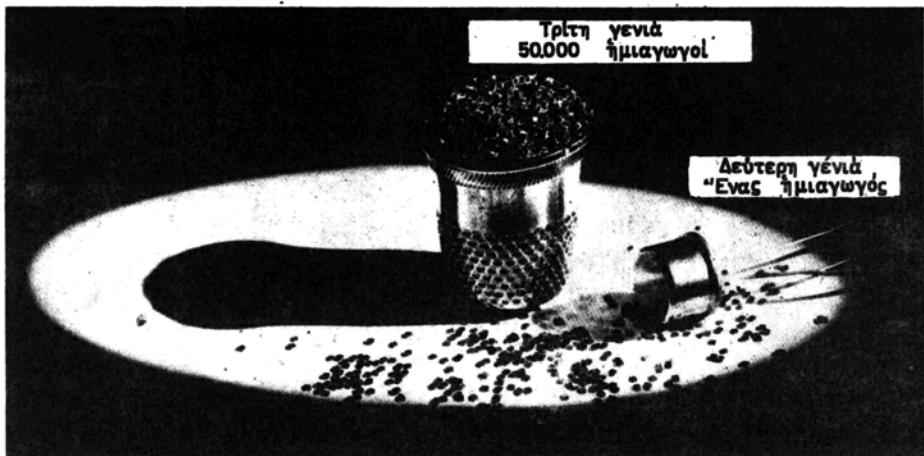
Η δεκαετία του 1970 όχι μόνο χαρακτηρίζεται από την τέταρτη γενιά υπολογιστών, αλλά διακρίνεται και από άλλα καταπληκτικά επιτεύγματα της έποιστήμης. Όπως η κατασκευή μικροσκοπικών προγραμματιζομένων υπολογιστικών μηχανών, η μετάδοση δεδομένων και η ανταλλαγή πληροφοριών με τη βοήθεια δορυφόρων ανάμεσα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

Στη δεκαετία που διανύμε άρχισαν να κατασκευάζονται πειραματικά υπολογιστές, που θα χρησιμοποιούν για τη λειτουργία τους κινούμενα υγρά και υγροδυναμικές βαλβίδες, αντί για ηλεκτρονικά κυκλώματα, γεγονός που θα σημάνει σημαντική έλαπτωση 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 οι έποιστήμονες μελετούν τη χρησιμοποίηση νέων ηλεκτροοπτικών μνημών, που θα αποτελούνται από εξαιρετικά λεπτά στρώματα σιδηροηλεκτρικών υλικών και θα μπορούν να μεταβάλλουν την κατάστασή τους με την ταχύτητα του φωτός.

1.8 Ο ηλεκτρονικός υπολογιστής θά υποκαταστήσει εντελώς τον ανθρώπο;

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

Ας μη ξεχνάμε ότι ο ηλεκτρονικός υπολογιστής, που είναι προϊόν της ανθρώπινης εξυπνάδας και ικανότητας, δεν είναι παρά μια άψυχη μηχανή χωρίς δική της εξελιγμένη λογική, η οποία δύναται διαθέτει δύο μεγάλες δυνατότητες: την εκτέλεση μεγάλου σύγκου υπολογισμών με τρομακτική ταχύτητα και την αποθήκευση μεγάλων ποσοτήτων στοιχείων, δεδομένων και πληροφοριών.



Σχ. 1.8.

Στη φωτογραφία δεξιά φαίνεται ο ημιαγωγός που χρησιμοποιήθηκε στους υπολογιστές της δέυτερης γενιάς. Αριστερά μέσα στη δαχτυλίθρα υπάρχουν 50.000 ημιαγωγοί νεώτερής τεχνολογίας που χρησιμοποιήθηκαν στους υπολογιστές της τρίτης γενιάς.

1.9 Κατηγορίες υπολογιστικών μηχανών.

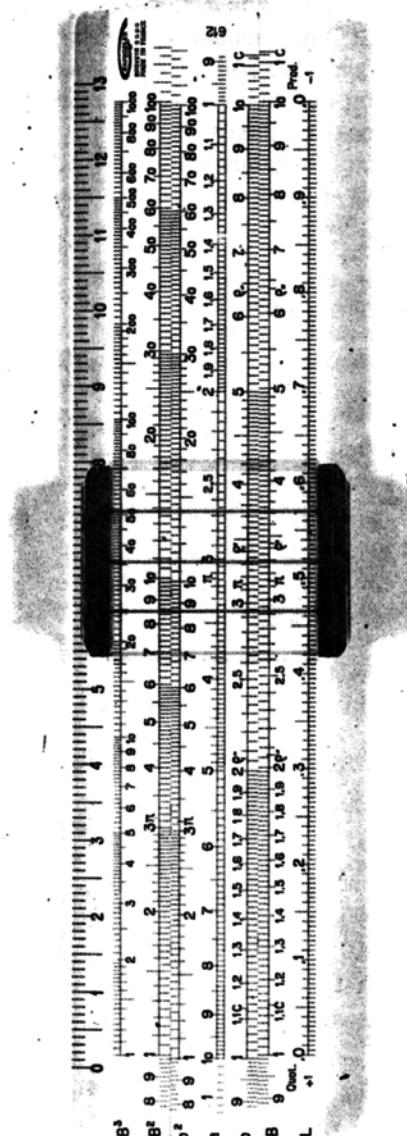
Τις υπολογιστικές μηχανές τις κατατάσσουμε σε τρεις μεγάλες κατηγορίες:

- α) Ψηφιακές (Digital).
- β) Αναλογικές (Analog).
- γ) Υβριδικές (Hybrid).

Στις αναλογικές, οι αριθμητικές ποσότητες παριστάνονται αναλογικά με κάποιο άλλο φυσικό μέγεθος όπως το μήκος, η γωνία στροφής, το ηλεκτρικό δυναμικό, η μαγνητική ροή κλπ. Η πιο γνωστή αναλογική υπολογιστική συσκευή είναι ο λογαριθμικός κανόνας (Slide Rule, σχήμα 1.9) όπου οι λογάριθμοι των πραγματικών αριθμών παριστάνονται με ανάλογες αποστάσεις από κάποια αρχή, πάνω στον κανόνα. Έτσι, αν θέλομε να πολλαπλασιάσουμε δύο πραγματικούς αριθμούς, επειδή όπως ξέρομε το άθροισμα των λογαρίθμων δύο αριθμών ισούται με το λογάριθμο του γινομένου τους¹, ο πολλαπλασιασμός γίνεται στον κανόνα με απλή πρόσθεση των μηκών των λογαρίθμων τους και στη συνέχεια αντιλογαρίθμηση.

Οι σύγχρονοι αναλογικοί υπολογιστές κατασκευάζονται με ηλεκτρονικά κυκλώματα και έχουν αρκετές δυνατότητες, χρησιμοποιούν δε σαν αναλογικά μεταβαλλόμενο μέγεθος το ηλεκτρικό δυναμικό.

¹ λογ(A·B) = λογA + λογB



Σχ. 1.8.
Ο λογαριθμικός κανόνας

Η πιο παλιά από τις ψηφιακές υπολογιστικές συσκευές είναι ο άβακας (σχήμα 1.2a) και η πιο εξελιγμένη, ο ψηφιακός ηλεκτρονικός υπολογιστής (Digital Electronic Computer).

Η πιο βασική διαφορά ανάμεσα στις δύο κατηγορίες που αναφέραμε, είναι ότι στους ψηφιακούς υπολογιστές οι αριθμητικές ποσότητες μπορούν να μεταβάλλονται μόνο κατά τρόπο ασυνεχή, ενώ στους αναλογικούς μεταβάλλονται κατά τρόπο συνεχή. Έτσι, ενώ ο αριθμός $1/3$ δεν μπορεί να παρασταθεί σε ένα ψηφιακό υπολογιστή ακριβώς με ένα πεπερασμένο πλήθος από ψηφία, σε ένα αναλογικό υπολογιστή που χρησιμοποιεί σαν αναλογικά μεταβαλλόμενο μέγεθος το μήκος, το $1/3$ της μονάδας μήκους μπορεί να βρεθεί.

Οι υβριδικοί υπολογιστές που ανήκουν στην τρίτη κατηγορία συνδυάζουν χαρακτηριστικά των αναλογικών και των ψηφιακών υπολογιστών.

Ερωτήσεις:

1. Ποιος θέωρεται ο πατέρας των συγχρόνων υπολογιστών;
 2. Ποια είναι η ιστορία της διάτρητης καρτέλας;
 3. Τι μέσα καταγραφής χρησιμοποιούσαν οι αρχαίοι λαοί;
 4. Ποιες οι διάφορες γεννήσεις των ψηφιακών ηλεκτρονικών υπολογιστών; Από ποιο χαρακτηρίζεται η κάθε μια;
 5. Σε πόσες και ποιες κατηγορίες κατατάσσομε τις υπολογιστικές μηχανές. Αναφέρατε παραδείγματα για τις δύο κυριότερες κατηγορίες.
-

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

ΑΡΙΘΜΗΤΙΚΑ ΣΥΣΤΗΜΑΤΑ – Η ΑΡΙΘΜΗΤΙΚΗ ΤΩΝ ΨΗΦΙΑΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

2.1 Γενικά.

Το δεκαδικό αριθμητικό σύστημα, που χρησιμοποιούμε στις καθημερινές μας συναλλαγές, προέρχεται από τους Ἀράβες και βασίσθηκε στη χρήση των δέκα δακτύλων των χεριών. Στην Ευρώπη έγινε γνωστό γύρω στα 1200 μ.Χ. Το σύστημα όμως αυτό δεν είναι και το μοναδικό. Υπάρχουν, όπως θα δούμε, και άλλα αριθμητικά συστήματα, που είναι περισσότερο ή λιγότερο κατάλληλα, για να τα χρησιμοποιήσουμε στην παράσταση των αριθμών στένα ηλεκτρονικό υπολογιστή, όπως το δυαδικό, το οκταδικό, το δεκαεξαδικό κλπ.

Η ονομασία κάθε αριθμητικού συστήματος προέρχεται από το πλήθος των διαφορετικών ψηφίων, που έχουμε στη διάθεσή μας, για να παραστήσουμε τις αριθμητικές ποσότητες. Έτσι το δυαδικό σύστημα περιλαμβάνει μόνο δύο ψηφία, το οκταδικό 8, το δεκαεξαδικό 16 κλπ. Κάθε αριθμό, που εκφράζεται σε κάποιο αριθμητικό σύστημα, τον γράφομε μέσα σε παρενθέσεις. Έξω από αυτή και δεξιά κάτω σημειώνομε αριθμητικά το σύστημα που ανήκει π.χ. ο αριθμός $(124)_8$ ανήκει στο οκταδικό σύστημα, ο αριθμός $(521)_{16}$ ανήκει στο δεκαεξαδικό, ο $(143)_{10}$ στο δεκαδικό κ.ο.κ. Το ίδιο ψηφίο βέβαια μπορεί να υπάρχει σε πολλά διαφορετικά συστήματα αλλά η αξία του αλλάζει από σύστημα σε σύστημα, έτσι π.χ. στον μεν αριθμό $(12)_{10}$ το 1 έχει αξία μιας δεκάδας στόν δε $(12)_8$, έχει αξία μιας οκτάδας κ.ο.κ. Η περιγραφή των αριθμητικών συστημάτων, που ακολουθεί, είναι βασική για την εξήγηση και κατανόηση της λειτουργίας των ηλεκτρονικών υπολογιστών. Αναλυτικότερα θα εξηγηθεί η χρήση τους στα ειδικά κεφάλαια.

2.2 Το δεκαδικό αριθμητικό σύστημα.

Το σύστημα αυτό περιλαμβάνει, όπως είναι γνωστό, δέκα ψηφία 0, 1, 2, ...9. Η αξία κάθε ψηφίου εξαρτάται από τη θέση που κατέχει μέσα σε ένα αριθμό. Π.χ. στον αριθμό $(124)_{10}$, το 4 έχει αξία τεσσάρων μονάδων, το 2 έχει αξία δύο δεκάδων ή 20 μονάδων, και το 1 έχει αξία μιας εκατοντάδας. Δηλαδή όπως έχω γράψει τον αριθμό είναι σα να εννοώ $1 \times 100 + 2 \times 10 + 4 \times 1 \equiv 1 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$. Όπως βλέπουμε, κάθε αριθμό του δεκαδικού συστήματος μπορούμε να τον παραστήσουμε σαν ένα άθροισμα με δυνάμεις του 10, όπου τα ψηφία του αριθμού εμφανίζονται σαν συντελεστές. Αν αντίστροφα έχω το άθροισμα $2 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 0 \times 10^0$, αυτό δεν είναι άλλο από τον αριθμό $(2350)_{10}$. Το 10 ονομάζεται **βάση** του δεκαδικού συστήματος.

Πώς όμως θα παραστήσουμε έναν κλασματικό¹ αριθμό π.χ. τον $(210.12)_{10}$? Απλούστατα θα χρησιμοποιήσουμε για την παράσταση των κλασματικών ψηφίων του αριθμού, αρνητικές δυνάμεις του 10. Έτσι μπορώ να τον 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 Το δυαδικό αριθμητικό σύστημα.

Στο σύστημα αυτό βάση είναι το 2 και υπάρχουν μόνο δύο ψηφία το 0 και το 1. Όλοι οι αριθμοί εκφράζονται ως συνδυασμοί του 0 και του 1. Έτσι, ο ακέραιος δυαδικός αριθμός $(11001)_2$ ισοδυναμεί με τον $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = 25$ του δεκαδικού συστήματος. Επίσης ο μικτός δυαδικός αριθμός 1101.11 ισοδυναμεί με τον $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 + 1 \times \frac{1}{2} + 1 \times \frac{1}{4} = (13.75)_{10}$ του δεκαδικού συστήματος.

2.4 Το οκταδικό αριθμητικό σύστημα.

Το σύστημα αυτό περιλαμβάνει 8 ψηφία δηλ. 0, 1, ..., 7, και έχει βάση το 8. Έτσι ο ακέραιος οκταδικός αριθμός $(3470)_8$ ισοδυναμεί με τον $3 \times 8^3 + 4 \times 8^2 + 7 \times 8^1 + 0 \times 8^0 = 1536 + 256 + 56 + 0 = (1848)_{10}$.

Επίσης ο μικτός οκταδικός αριθμός 123.14 ισοδυναμεί με τον $1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 + 1 \times 8^{-1} + 4 \times 8^{-2} = 16 + 16 + 3 + \frac{1}{8} + \frac{1}{64} = (35.1406)_{10}$.

Κάθε ψηφίο του οκταδικού συστήματος εκφράζεται με τρία ψηφία του δυαδικού. Δηλαδή ο αριθμός $(3470)_8$ απεικονίζεται με 12 δυαδικά ψηφία.

2.5 Το δεκαεξαδικό αριθμητικό σύστημα.

Το δεκαεξαδικό σύστημα έχει ως βάση το 16 και αποτελείται από 16 ψηφία 0, 1, ... 9, A, B, C, D, E, F. Όπως βλέπομε, επειδή τα 10 ψηφιακά σύμβολα, που γνωρίζουμε από το δεκαδικό σύστημα δεν φθάνουν, χρησιμοποιούμε και τα 6 πρώτα γράμματα του Λατινικού Αλφαριθμητικού. Δηλαδή το A συμβολίζει το 10, το B συμβολίζει το 11 κ.ο.κ. Έτσι, ο ακέραιος 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 του δεκαεξαδικού συστήματος απεικονίζεται με τέσσερα ψηφία του δυαδικού συστήματος. Δηλαδή ο αριθμός $(4CD)_{16}$ θα εκφράζεται στο δυαδικό σύστημα με 12 ψηφία.

¹ Κλασματικό αριθμό θα ονομάζουμε ότι μέχρι σήμερα από κακή συνήθεια ονομάζαμε «δεκαδικό».

Ο δεκαδικός αριθμός είναι ένας αριθμός που εκφράζεται στο δεκαδικό αριθμητικό σύστημα και όχι ένας αριθμός που περιλαμβάνει κλάσματα της ακέραιας μονάδας.

Μικτός αριθμός είναι ένας αριθμός που περιλαμβάνει και ακέραιο και κλασματικό μέρος.

Το σημείο της υποδιαστολής θα το παριστάνουμε με τελεία, γιατί έτσι παριστάνεται στους υπολογιστές.

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

α) Γενικά.

Επειδή το δυαδικό σύστημα είναι το απλούστερο που υπάρχει μια και αποτελείται μόνο από δύο ψηφία 0 και 1, είναι το καταλληλότερο για να εξυπηρετήσει την αριθμητική των ψηφιακών υπολογιστών. Σημειώνομε ότι είναι πιο εύκολο να βρει κανείς φυσικά συστήματα, που να παίρνουν μόνο δύο διακεκριμένες τιμές, όπως διακόπτης αυνοικτός ή κλειστός, μεταλλικός δακτύλιος μαγνητισμένος η αμάγνητιστος, λάμπτα αναμμένη ή σβηστή, αγωγός ηλεκτροφόρος ή ουδέτερος κλπ.

Το οκταδικό και το δεκαεξαδικό σύστημα χρησιμοποιούνται για λόγους ευκολίας έξω από τον υπολογιστή. Είναι λοιπόν χρήσιμο να γνωρίζουμε πως ένας δεκαδικός, οκταδικός ή δεκαεξαδικός αριθμός μετατρέπεται σε δυαδικό και αντίστροφα.

β) Μετατροπή ακέραιου δεκαδικού αριθμού σε δυαδικό.

Για να μετατρέψουμε ένα ακέραιο δεκαδικό αριθμό σε δυαδικό, τον διαιρούμε με τη βάση του δυαδικού συστήματος, δηλαδή το 2. Το υπόλοιπο της διαιρέσεως θα είναι το 0 ή το 1, ενώ το ακέραιο πηλίκο θα είναι κάποιος άλλος δεκαδικός. Διαιρούμε το πηλίκο πάλι με το 2. Το υπόλοιπο της νέας διαιρέσεως θα είναι 0 ή 1 και το πηλίκο κάποιος νέος δεκαδικός αριθμός. Συνεχίζουμε με αυτόν τον τρόπο τις διαιρέσεις των νέων πηλίκων, έως ότου αυτό μηδενισθεί.

Αν π.χ. θέλουμε να μετατρέψουμε τον ακέραιο δεκαδικό αριθμό $(45)_{10}$ σε δυαδικό, θα προβούμε στη σειρά στις εξής πράξεις:

$$\begin{aligned} 45 : 2 &= 22 \quad \text{και υπόλοιπο } 1 \\ 22 : 2 &= 11 \quad \text{και υπόλοιπο } 0 \\ 11 : 2 &= 5 \quad \text{και υπόλοιπο } 1 \\ 5 : 2 &= 2 \quad \text{και υπόλοιπο } 1 \\ 2 : 2 &= 1 \quad \text{και υπόλοιπο } 0 \\ 1 : 2 &= 0 \quad \text{και υπόλοιπο } 1 \end{aligned}$$

Αν τώρα τοποθετήσουμε τα υπόλοιπα που βρήκαμε, στις διαδοχικές διαιρέσεις σε αντίστροφη σειρά.. Θα έχουμε το ζητούμενο αριθμό. Δηλαδή:

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

γ) Μετατροπή κλασματικού δεκαδικού αριθμού σε δυαδικό.

Για να κάνουμε αυτή τη μετατροπή έργαζόμαστε ως εξής: Πολλαπλασιάζομε τον κλασματικό με τη βάση 2. Αν το γινόμενο είναι μεγαλύτερο από τη μονάδα την αφαιρούμε και το αποτέλεσμα το πολλαπλασιάζομε πάλι με το 2. Συνεχίζουμε μ' αυτόν τον τρόπο μέχρις ότου πετύχομε μια επιθυμητή ακρίβεια. (Δηλαδή στην περίπτωση αυτή η διαδικασία της μετατροπής δεν τελειώνει μόνη της, αλλά συνεχίζεται όσο θέλουμε). Για τον κλασματικό $(0.379)_{10}$ θα έχουμε:

$$\begin{aligned} 0.379 \times 2 &= 0.758 \quad \text{και ακέραιο μέρος } 0 \\ 0.758 \times 2 &= 0.516 \quad \text{και ακέραιο μέρος } 1 \\ 0.516 \times 2 &= 0.032 \quad \text{και ακέραιο μέρος } 1 \\ 0.032 \times 2 &= 0.064 \quad \text{και ακέραιο μέρος } 0 \\ 0.064 \times 2 &= 0.128 \quad \text{και ακέραιο μέρος } 0 \end{aligned}$$

$$\begin{array}{llll}
 C & 28 \times 2 = 0.256 & \text{kai} & \text{akéraio} \quad \text{méros} \quad 0 \\
 C & !56 \times 2 = 0.512 & \text{kai} & \text{akéraio} \quad \text{méros} \quad 0 \\
 O & .12 \times 2 = 0.024 & \text{kai} & \text{akéraio} \quad \text{méros} \quad 1 \text{ klp.}
 \end{array}$$

Δηλαδή $(0.379)_{10} = (0.011000001..)_2$

Όταν ο δεκαδικός αριθμός έχει και ακέραιο και κλασματικό μέρος τότε το μεν ακέραιο μέρος το μετατρέπομε με τὸν τρόπο που είδαμε στην παράγραφο 2.6(β) το δε κλασματικό με τὸν τρόπο που εξηγήσαμε τώρα.

δ) Μετατροπή ακέραιου δυαδικού αριθμού σε δεκαδικό.

Για να μετατρέψουμε ένα ακέραιο δυαδικό αριθμό σε ακέραιο δεκαδικό, πολλαπλασιάζομε το πρώτο από τὰ αριστερά ψηφίο του δυαδικού με τὸ 2 και στο γινόμενο προσθέτομε τὸ ἐπόμενο στὴ σειρά ψηφίο του δυαδικού. Πολλαπλασιάζομε κατόπιν τὸ άθροισμα που βρήκαμε πάλι με τὸ 2 και στὸ νέο γινόμενο προσθέτομε τὸ ἐπόμενο ψηφίο κ.ο.κ. Δηλαδή, γιὰ να μετατρέψουμε τὸν δυαδικό $(110101)_2$ σε δεκαδικό θὰ ενεργήσομε ως εξῆς:

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

ε) Μετατροπή κλασματικού δυαδικού αριθμού σε δεκαδικό.

Για να μετατρέψουμε ένα κλασματικό δυαδικό αριθμό σε δεκαδικό διαιρούμε τὸ τελευταίο ψηφίο του αριθμού (δηλαδή αυτὸ που έχει τὴ μικρότερη αξία) με τὴ βάση 2. Στὸ πηλίκο προσθέτομε τὸ αμέσως προηγούμενο ψηφίο τοῦ δυαδικού και διαιρούμε πάλι με τὸ 2. Στὸ νέο πηλίκο προσθέτομε τὸ τρίτο από τὰ δεξιά ψηφίο και συνεχίζομε με αυτὸν τὸν τρόπο ὡσπου να φθάσουμε στὸ τελευταίο πρὸς τὰ αριστερά ψηφίο του αριθμού π.χ. γιὰ τὴ μετατροπὴ του $(0.1101)_2$ σε δεκαδικό, θὰ κάνομε τὰ 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 ακέραιου δεκαδικού αριθμού σε οκταδικό.

Η μετατροπή γίνεται διαιρώντας διαδοχικά το δοθέντα αριθμό με τη βάση 8. Τα υπόλοιπα από τις διαιρέσεις τοποθετημένα σε αντίστροφη σειρά μας δίνουν το ζητούμενο οκταδικό, π.χ. για τον $(79)_{10}$ θα έχομε:

$$\begin{array}{rcl} 79 : 8 = 9 & \text{και} & \text{υπόλοιπο} & 7 \\ 9 : 8 = 1 & \text{και} & \text{υπόλοιπο} & 1 \\ 1 : 8 = 0 & \text{και} & \text{υπόλοιπο} & 1 \\ \text{άρα } (79)_{10} = (117)_8 \end{array}$$

Ω) Μετατροπή ακέραιου οκταδικού αριθμού σε δεκαδικό.

Πολλαπλασιάζομε το πρώτο ψηφίο του αριθμού με το 8 και στο γινόμενο προσθέτομε το επόμενο ψηφίο. Πολλαπλασιάζομε το άθροισμα πάλι με το 8 και προσθέτομε το ψηφίο του αριθμού κ.ο.κ. Π.χ. ο οκταδικός $(117)_8$ μετατρέπεται σε δεκαδικό ως εξής:

$$\begin{array}{r} 1 \times 8 = 8 \\ 8 + 1 = 9 \\ 9 \times 8 = 72 \\ 72 + 7 = \boxed{79} \\ \text{άρα } (117)_8 = (79)_{10} \end{array}$$

η) Μετατροπή ακέραιου δεκαδικού αριθμού σε δεκαεξαδικό.

Χρησιμοποιούμε τη μέθοδο, που περιγράψαμε στις παραγράφους 2.6(β) και 2.6(στ) μόνο που εδώ θεωρούμε ως βάση το 16. Έτσι, για να μετατρέψουμε τον ακέραιο δεκαδικό αριθμό $(146)_{10}$ σε δεκαεξαδικό, θα ακολουθήσουμε τα ακόλουθα στάδια:

$$\begin{array}{rcl} 146 : 16 = 9 & \text{και} & \text{υπόλοιπο} & 2 \\ 9 : 16 = 0 & \text{και} & \text{υπόλοιπο} & 9 \\ \text{άρα } (146)_{10} = (92)_{16} \end{array}$$

θ) Μετατροπή ακέραιου δεκαεξαδικού αριθμού σε δεκαδικό.

Χρησιμοποιώντας τη μέθοδο που είδαμε στις παραγράφους 2.6(δ) και 2.6(ζ) μπορούμε να μετατρέψουμε τον δεκαεξαδικό $(4EF)_{16}$ σε δεκαδικό ως εξής:

$$\begin{array}{r} 4 \quad \times 16 = 64 \\ 64 \quad + E = 78 \\ 78 \quad \times 16 = 1248 \\ 1248 \quad + F = \boxed{1263} \\ \text{άρα } (4EF)_{16} = (1263)_{10} \end{array}$$

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

Όταν θέλομε να μετατρέψουμε έναν αριθμό μη δεκαδικό σε κάποιο άλλο μη δεκαδικό σύστημα, τότε χρησιμοποιούμε το δεκαδικό σύστημα ως ενδίμεσο. Δηλαδή, για να μετατρέψουμε ένα πενταδικό αριθμό σε δεκαεξαδικό, μετατρέπουμε πρώτο τον πενταδικό σε δεκαδικό και κατόπιν τον δεκαδικό, που βρήκαμε, σε δεκαεξαδικό.

Στην ειδική περίπτωση, που θέλομε να μετατρέψουμε ένα δυαδικό αριθμό σε κάποιο άλλο σύστημα, που έχει ως βάση μια δύναμη του 2, όπως είναι π.χ. το οκταδικό και το δεκαεξαδικό, υπάρχει απλούστερη μέθοδος όπως θα δούμε λίγο πιο κάτω.

2.7 Αριθμητικές πράξεις στο δυαδικό σύστημα.

α) Πρόσθεση.

Η πρόσθεση δύο δυαδικών αριθμών γίνεται, όπως και στους δεκαδικούς αριθμούς. Ξεκινώντας από την πρόσθεση των τελευταίων ψηφίων των αριθμών και συνεχίζοντας προς τα αριστερά. Πρέπει όμως να έχομε υπ' όψη μας τους εξής βασικούς κανόνες:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \quad \text{και} \quad \text{κρατούμενο} \quad \text{για} \quad \text{μεταφορά} \quad 1$$

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) Αφαίρεση.

Αφαιρούμε τα αντίστοιχα ψηφία των αριθμών αρχίζοντας από τα τελευταία, έχοντας υπ' όψη μας τους εξής κανόνες:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \quad \text{με δανεισμό από το προηγούμενο ψηφίο μιας μονάδας } 1.$$

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 όπως και στο δεκαδικό σύστημα. Για τις ενδιάμεσες προσθέσεις χρησιμοποιούμε τους κανόνες της δυαδικής προσθέσεως.

Παραδείγματα:

$$\begin{array}{r}
 (110)_2 \\
 \times (11)_2 \\
 \hline
 110 \\
 110 \\
 \hline
 (10010)_2
 \end{array}
 \quad
 \begin{array}{r}
 (1101)_2 \\
 \times (101)_2 \\
 \hline
 1101 \\
 0000 \\
 1101 \\
 \hline
 (1000001)_2
 \end{array}
 \quad
 \begin{array}{r}
 (1001)_2 \\
 \times (111)_2 \\
 \hline
 1001 \\
 1001 \\
 1001 \\
 \hline
 (111111)_2
 \end{array}$$

δ) Διαιρεση.

Και αυτή η πράξη γίνεται όπως στο δεκαδικό σύστημα. Για τις ενδιάμεσες αφαιρέσεις χρησιμοποιούμε τους κανόνες της δυαδικής αφαιρέσεως.

Παραδείγματα:

$$\begin{array}{r}
 (1101)_2 \left| \begin{array}{r} (11)_2 \\ (100)_2 \end{array} \right. \\
 -11 \\
 \hline
 000 \\
 -\ 00 \\
 \hline
 01 \\
 -00 \\
 \hline
 01
 \end{array}
 \quad
 \begin{array}{r}
 (110110)_2 \left| \begin{array}{r} (110)_2 \\ (1001)_2 \end{array} \right. \\
 -110 \\
 \hline
 0001 \\
 -\ 000 \\
 \hline
 011 \\
 -000 \\
 \hline
 110 \\
 -110 \\
 \hline
 000
 \end{array}
 \quad
 \begin{array}{r}
 (111)_2 \left| \begin{array}{r} (10)_2 \\ (11)_2 \end{array} \right. \\
 -10 \\
 \hline
 11 \\
 -10 \\
 \hline
 01
 \end{array}$$

2.8 Αριθμητικές πράξεις στο οκταδικό σύστημα.**α) Πρόσθιση.**

Προσθέτουμε τα αντίστοιχα ψηφία των αριθμών αρχίζοντας από το τέλος. Κάθε φορά που το άθροισμά τους ξεπερνά το 8 σημειώνουμε τη διαφορά από το 8 και μεταφέρουμε μια οκταδική μονάδα, για να την προσθέσουμε στο άθροισμα των αιμέσως προηγουμένων ψηφίων.

Παραδείγματα:

$$\begin{array}{r}
 (746)_8 \\
 + (23)_8 \\
 \hline
 (771)_8
 \end{array}
 \quad
 \begin{array}{r}
 (1374)_8 \\
 +(6510)_8 \\
 \hline
 (10104)_8
 \end{array}$$

β) Αφαιρεση.

Αφαιρούμε τα αντίστοιχα ψηφία των αριθμών αρχίζοντας από το τέλος. Αν δυνατός είναι ψηφίο του αφαιρέτη είναι μεγαλύτερο από το αντίστοιχο του αφαιρετέου, τότε δανειζόμαστε μια οκταδική μονάδα από το αμέσως προηγούμενο ψηφίο του διαιρέτη.

Παραδείγματα:

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

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

γ) Πολλαπλασιασμός.

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

Παραδείγματα:

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

$$\begin{array}{r} (47)_8 \\ \times (12)_8 . \\ \hline 116 \\ 47 \\ \hline (606)_8 \end{array}$$

ε) Διαίρεση.

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

Παραδείγματα:

$$\begin{array}{r} (145)_8 \quad | \quad (12)_8 \\ -12 \quad | \quad (12)_8 \\ \hline 25 \\ -24 \\ \hline 01 \end{array}$$

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

Οι αριθμητικές πράξεις στα άλλα αριθμητικά συστήματα εκτελούνται με εντελώς 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 αντιστοιχίζομε 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 $B = 2^n$

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

ПИНАКАΣ 2.9.1.

Κωδικοποίηση των αργίων του ακταδικού και δεκτεροδικού συστήματος με διαδικτύο

Ψηφία του οκταδικού συστήματος	Τα οκταδικά ψηφία κωδικοποιημένα με διαδικτικά ψηφία	Ψηφία του δεκαεξαδικού συστήματος	Τα δεκαεξαδικά ψηφία κωδικοποιημένα με διαδικτικά 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

$$\begin{array}{c} \text{'Ετσι ο αριθμός } (123)_8 = 001 \mid 010 \mid 011 \\ \text{ο αριθμός } (59AF)_{16} = 0101 \mid 1001 \mid 1010 \mid 1111 \end{array}$$

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 το πλήθος των ψηφίων του δυαδικού δεν διαιρέται με το 3 ή το 4 συμπληρώνομε στην αρχή μπδενικά.

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

Για το οκταδικό 110 | 111 | 011
 6 | 7 | 3 → (673)₈

Για το δεκαεξαδικό θα πρέπει να προσθέσουμε στην αρχή τρία μηδενικά για να μπορέσουμε να κάνουμε το χωρισμό των ψηφίων. Δηλαδή:

0001 | 1011 | 1011
 1 | B | B → (1BB)₁₆

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

Ο κώδικας αυτός χρησιμοποιείται συχνά για την παράσταση των αριθμών μέσα στον υπολογιστή. Κάθε αριθμητικός χαρακτήρας κωδικοποιείται με τέσσερα δυαδικά ψηφία, που έχουν αξία 8,4,2,1 αντίστοιχα. Την αντιστοιχία την βλέπομε στον ακόλουθο πίνακα:

ΠΙΝΑΚΑΣ 2.8.2.

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. Με τη βοήθεια του κώδικα αυτού ω δεκαδικός (954)₁₀. γράφεται:

Δεκαδικός	9	5	4
BCD	1001	0101	0100

Το πλεονέκτημα του κώδικα BCD είναι η μεγάλη ευκολία, με την οποία ένας αριθμός παριστάνεται με δυαδικά ψηφία, ενώ το μειονέκτημα είναι η μεγαλύτερη δυσκολία στην εκτέλεση αριθμητικών πράξεων.

γ) Βιτρίφιος αλφαριθμητικός κώδικας.

Στον κώδικα αυτό κάθε χαρακτήρας παριστάνεται με 6 δυαδικά ψηφία. Από αυ-

τα τα δύο πρώτα, που τα συμβολίζομε γενικά με Β και Α, αποτελούν το λεγόμενο τμήμα ζώνης. Τα επόμενα 4 ψηφία έχουν αξία κατά σειρά 8, 4, 2, 1 και αποτελούν το αριθμητικό τμήμα. Τον κώδικα αυτό βλέπομε στον 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).

Ο κώδικας αυτός χρησιμοποιείται όπως και ο BCD, με τη διαφορά ότι κάθε χαρακτήρας παριστάνεται με 8 δυαδικά ψηφία. Από αυτά τα 4 πρώτα αποτελούν το λεγόμενο τμήμα ζώνης και τα 4 επόμενα το αριθμητικό τμήμα. Η αντιστοιχία ανάμεσα στους διάφορους χαρακτήρες και στους δυαδικούς αριθμούς του EBCDIC φαίνεται στον επόμενο πίνακα.

ΠΙΝΑΚΑΣ 2.9.4.

Η δυαδική παράσταση κάθε χαρακτήρα αποδίδεται με οκτώ δυαδικά ψηφία που αποτελούν ένα 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		

e) Κώδικας Hollerith.

Είναι ο πρώτος δυαδικός κώδικας που χρησιμοποιήθηκε για απεικόνιση πληροφοριών. Δημιουργήθηκε από τον Herman Hollerith. Στον κώδικα αυτόν, κάθε χαρακτήρας απεικονίζεται με 12 δυαδικά ψηφία (bits) που αντιστοιχούν στις 12 θέσεις διατρήσεως μιας στήλης ενός δελτίου 80 στηλών. Οι 10 τελευταίες θέσεις

¹ Ένα byte είναι μια ομάδα που αποτελείται συνήθως από οκτώ δυαδικά ψηφία (bits).

ψηφίων αποτελούν το ψηφιακό τμήμα, του κώδικα, ενώ οι τρεις πρώτες το τμήμα ζώνης (μία θέση είναι κοινή). Ο πλήρης πίνακας κωδικοποίησεως των χαρακτήρων παρουσιάζεται πιο κάτω (παράγραφος 3.4a). Με τον κώδικα αυτό μπορούν να απεικονισθούν μέχρι 120 διαφορετικοί χαρακτήρες.

2.10 Δυαδικά ψηφία ισοτιμίας.

Η λειτουργία ενός υπολογιστή χαρακτηρίζεται από συνεχή μετακίνηση μεγάλου δύκου δεδομένων κατά τη διάρκεια της έπεξεργασίας. Παρ' όλη την τεχνική τελειότητα της μηχανής, μπορεί να συμβούν σφάλματα ή να χαθούν πληροφορίες, επειδή κάποιο από τα ηλεκτρονικά κυκλώματα δεν λειτουργησε σωστά για ένα κλάσμα του δευτερολέπτου. Για το λόγο αυτό οι σχεδιαστές υπολογιστών εφοδιάζουν τις μηχανές με διάφορους μηχανισμούς ελέγχου σφαλμάτων. Ανάμεσά τους ο πιο γνωστός φέρει την ονομασία δυαδικό ψηφίο ισοτιμίας¹ (parity bit ή binary check bit). Το ψηφίο ισοτιμίας προστίθεται σε κάθε δυαδική απεικόνιση δεδομένων. Επομένως αν χρησιμοποιείται ο τετραψήφιος BCD κώδικας με την προσθήκη του ψηφίου ισοτιμίας θα γίνει πενταψήφιος, αν χρησιμοποιείται ο εξαψήφιος αλφαριθμητικός θα γίνει επταψήφιος αν δε χρησιμοποιείται ο οκταψήφιος EBCDIC, θα γί-

ΠΙΝΑΚΑΣ 2.10.1.

Ο πίνακας 8421 με δρτιά και περιπτή ισοτιμία

Ψηφία δεκαδικού συστήματος	Κωδικοποίηση με τον Κώδικα 8, 4, 2, 1	Κωδικοποίηση με τον Κώδικα 8, 4, 2, 1 και με δρτιά ισοτιμία	Κωδικοποίηση με τον Κώδικα 8, 4, 2, 1 και με περιπτή ισοτιμία
	8 4 2 1	8 4 2 1 A	8 4 2 1 P
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 εννεαψήφιος. Με τον τρόπο αυτό αποκαθιστούμε, όπως λέμε, μια ισοτιμία μεταξύ των κωδικοποιημένων αριθμών. Η ισοτιμία αυτή είναι δύο ειδών:

Δρτιά ισοτιμία. (Even parity bit). Σύμφωνα με αυτή, προσθέτομε στο τέλος του κωδικοποιημένου χαρακτήρα το 0 ή το 1 έτσι ώστε μετά από αυτή την προσθήκη,

¹ Από δώ ή πέρα το δυαδικό ψηφίο θα το ονομάζομε bit.

ο κωδικοποιημένος χαρακτήρας νά έχει άρτιο συναλικό πλήθος μονάδων 1.

Περιπτή ισοτυμία (Odd parity bit). Σύμφωνα με αυτή, προσθέτουμε σε κάθε κωδικοποιημένο χαρακτήρα το 0 ή το 1 έτσι, ώστε μετά την προσθήκη αυτή ο κωδικοποιημένος χαρακτήρας να έχει περιπτό πλήθος μονάδων 1. Την εφαρμογή της άρτιας ή περιπτής ισοτυμίας στον τετραψήφιο κώδικα BCD βλέπουμε στον παρακάτω πίνακα.

Ασκήσεις:

1. Να μετατραπούν οι δεκαδικοί αριθμοί 543, 156, 256 σε διαδικούς, οκταδικούς και δεκαεξαδικούς.
2. Να μετατραπούν οι δεκαδικοί αριθμοί 15.112, 164.34, 78.07 σε διαδικούς.
3. Να μετατραπούν οι οκταδικοί 175, 24, 502 σε δεκαδικούς.
4. Να μετατραπούν οι δεκαεξαδικοί 1AD, 454E, 22F σε δεκαδικούς.
5. Να μετατραπούν οι οκταδικοί 54, 73, 601 σε διαδικούς.
6. Να μετατραπούν οι δεκαεξαδικοί 4EF2, AB, 146 σε διαδικούς.
7. Το τριαδικό σύστημα αποτελείται από τα ψηφία 0,1,2 και έχει βάση το 3. Χρησιμοποιώντας κατάλληλα τη μέθοδο της παραγράφου 2.8(β) να μετατρέψετε τον δεκαδικό 4789 σε τριαδικό.
8. Να γίνουν οι ακόλουθες πράξεις ανάμεσα στους διαδικούς αριθμούς:

101101 1111	1001101 +1110011	1011110 - 10111	111011 + 111
----------------	---------------------	--------------------	-----------------

9. Να γίνουν οι εξής πράξεις ανάμεσα στους οκταδικούς αριθμούς:

4765 +5231	3762 - 145	23501 + 7647	5243 - 351
---------------	---------------	-----------------	---------------

10. Να γίνουν οι εξής πράξεις ανάμεσα στους οκταδικούς αριθμούς:

236 x 25	310 + 62	325 x 12	416 + 132
-------------	-------------	-------------	--------------

11. Να μετατραπούν οι διαδικοί 1100110101 και 101111001 σε οκταδικούς και δεκαεξαδικούς με τη βοήθεια του πίνακα 2.9.1.
12. Να μετατραπούν οι οκταδικοί 376 και 1432 και οι δεκαεξαδικοί 4ED2 και 18CAF σε διαδικούς, με τη βοήθεια του πίνακα 2.9.1.
13. Πώς θα κωδικοποιηθούν οι δεκαδικοί αριθμοί 145, 36, 2051 στον κώδικα BCD με άρπα ισοτυμία;
14. Πώς θα κωδικοποιηθούν οι δεκαδικοί αριθμοί 48,17, 163 στον κώδικα BCD με περιπτή ισοτυμία;
15. Πώς θα κωδικοποιηθούν οι δεκαδικοί αριθμοί 157, 249 στον κώδικα EBCDIC;

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

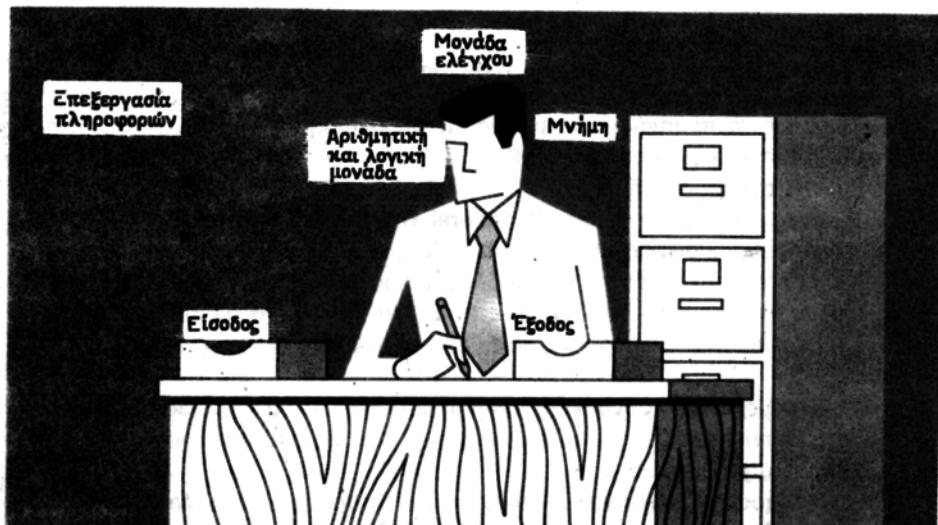
ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΨΗΦΙΑΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ I

3.1 Γενικά.

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

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

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



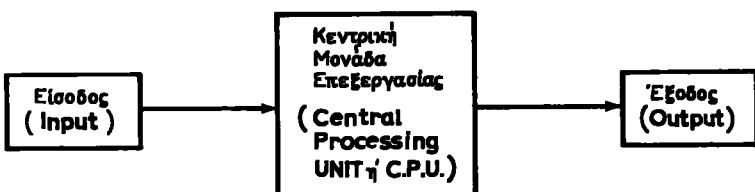
Σχ. 3.1a.

Η επεξεργασία πληροφοριών από τον άνθρωπο.

Μια παρόμοια διαδικασία ακολουθείται για να λυθεί κάποιο πρόβλημα από τον ηλεκτρονικό υπολογιστή. Έτσι και εδώ θα πρέπει να δοθούν στον υπολογιστή οι αναγκαίες πληροφορίες, δηλαδή τα δεδομένα και οι οδηγίες γραμμένες πάνω σε κάποιο φορέα, τις οποίες κατόπιν θα ακολουθήσει εκτελώντας τις και θα βγάλει αποτελέσματα. Βέβαια οι πληροφορίες δίνονται στον υπολογιστή κωδικοποιημένες σε κάποιον από τους κώδικες του δυαδικού συστήματος, που γνωρίσαμε ήδη, γιατί ο υπολογιστής δεν καταλαβαίνει παρά μόνο τη γλώσσα των δυαδικών αριθμών.

Το τμήμα του υπολογιστή, από όπου εισάγομε σ' αυτόν τις διάφορες πληροφορίες, ονομάζεται **είσοδος (Input)**, ενώ το τμήμα από όπου παίρνουμε τα αποτελέσματα, **Έξοδος (Output)**. Η είσοδος και η έξοδος στην πραγματικότητα αποτελούνται από ξεχωριστές μονάδες διαφόρων τύπων που ονομάζονται **περιφερειακές μονάδες εισόδου-έξόδου αντίστοιχα (I/O Peripheral Units)** και η καθεμιά τους μπορεί να διαβάζει τις πληροφορίες ή να γράφει τα αποτελέσματα πάνω σε κάποιο συγκεκριμένο φορέα.

Το τμήμα του υπολογιστή, όπου γίνεται η επεξεργασία των πληροφοριών, ονομάζεται **κεντρική μονάδα επεξεργασίας ή C.P.U (Central Processing Unit), (σχ. 3.1β)**.



Σχ. 3.1β.
Η επεξεργασία των πληροφοριών από τον ηλεκτρονικό υπολογιστή.

3.2 Τα βασικά μέρη ενός ψηφιακού ηλεκτρονικού υπολογιστή.

Σύμφωνα με τα προηγούμενα, μπορούμε να θεωρήσουμε ότι ένα συγκρότημα ψηφιακού ηλεκτρονικού υπολογιστή (Digital Electronic Computer System) αποτελείται βασικά από δύο τμήματα.

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

β) Την **κεντρική μονάδα επεξεργασίας (Central Processing Unit ή C.P.U)**.

Μερικές από τις περιφερειακές μονάδες, όπως είδαμε, χρησιμεύουν για να εισάγομε στον υπολογιστή πληροφορίες, που είναι δεδομένα (Data) και οδηγίες - εντολές (Instructions) και ονομάζονται **περιφερειακές μονάδες εισόδου (Input Devices)**.

Άλλες περιφερειακές μονάδες χρησιμοποιούνται για να αποδώσουν στον άνθρωπο τα αποτελέσματα της επεξεργασίας και ονομάζονται **περιφερειακές μονάδες έξόδου (Output devices)**.

Τέλος υπάρχουν περιφερειακές μονάδες που μπορούν δίλλοτε να χρησιμοποιηθούν ως μονάδες εισόδου και δίλλοτε ως μονάδες έξόδου (μικτές).

Η κεντρική μονάδα επεξεργασίας είναι το σημαντικότερο τμήμα ενός ψηφιακού υπολογιστή, η «καρδιά» ή καλύτερα ο εγκέφαλος του συγκροτήματος. Σ' αυτήν

πραγματοποιείται η επεξεργασία των πληροφοριών. Δεν είναι όμως μια ενιαία μονάδα, αλλά αποτελείται από τρία κυρίως τμήματα.

α) Την κεντρική ή κύρια ή εσωτερική μνήμη (Central or main or internal memory).

β) Την αριθμητική και λογική μονάδα (Arithmetic and Logical Unit).

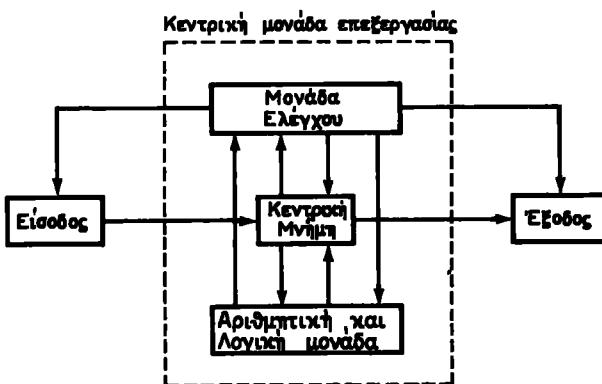
γ) Τη μονάδα ελέγχου (Control Unit).

Η κεντρική μνήμη είναι το τμήμα της κεντρικής μονάδας επεξεργασίας, όπου αποθηκεύονται – κωδικοποιημένα δυαδικά – τα δεδομένα, οι εντολές και τα ενδιάμεσα αποτελέσματα.

Η αριθμητική και λογική μονάδα είναι το τμήμα όπου εκτελούνται οι προβλεπόμενες αριθμητικές και λογικές πράξεις¹ ανάμεσα στα δεδομένα.

Η μονάδα ελέγχου, ελέγχει, καθοδηγεί και γενικά συντονίζει τη λειτουργία όλων των μονάδων του συγκροτήματος.

Στο σχήμα 3.2 βλέπουμε σχηματικά τη σχέση συνδέσεως ανάμεσα στα διάφορα τμήματα του υπολογιστή.



Σχ. 3.2.

Ο τρόπος συνδέσεως ανάμεσα στα διάφορα τμήματα του υπολογιστή.

3.3 Φορείς και μέσα καταγραφής πληροφοριών - Περιφερειακές μονάδες.

Όπως είπαμε πιο πάνω για να εισάγομε τις πληροφορίες σ' ένα ψηφιακό υπολογιστή πρέπει προηγουμένως να τις εγγράψουμε σε κωδικοποιημένη μορφή πάνω σε κάποιο φορέα. Σε κωδικοποιημένη ή και συνηθισμένη μορφή θα πάρομε και τα αποτελέσματα της επεξεργασίας. Τα μέσα καταγραφής πληροφοριών τα διακρίνουμε σε δύο κατηγορίες:

α) Μη μαγνητικά.

β) Μαγνητικά.

¹ Με τον όρο λογικές πράξεις εννοούμε τις λογικές συγκρίσεις ή τον έλεγχο λογικών συνθηκών π.χ. $a > b$ ή $\beta < x < y$ κλπ.

Στα μη μαγνητικά μέσα περιλαμβάνονται:

- Το δελτίο (Card).
- Η χαρτοταινία (Papertape).
- Το χαρτί εκτυπώσεως (Paper).
- Τα δελτία ή έντυπα με οπτικά αναγνωριζόμενους χαρακτήρες (Optical Characters Recognition ή O.C.R.).

Στα μαγνητικά μέσα περιλαμβάνονται:

- Η μαγνητική ταινία (Magnetic Tape).
- Ο μαγνητικός δίσκος (Magnetic Disk).
- Το μαγνητικό τύμπανο (Magnetic Drum).
- Οι μαγνητικές κάρτες (Magnetic Cards).
- Τα έντυπα με χαρακτήρες μαγνητικής μελάνης (Magnetic Ink Character Recognition ή M.I.C.R.).

Οι φορείς αυτοί των πληροφοριών τοποθετούνται πάνω σε ανάλογες περιφερειακές μονάδες, που τις διακρίνομε έτσι σε:

- α) Μη μαγνητικές.
- β) Μαγνητικές.

Μη μαγνητικές περιφερειακές μονάδες είναι:

- Η αναγνωστική διατρήτων δελτίων (Card Reader).
- Η αναγνωστική διάτρητης χαρτοταινίας (Papertape Reader).
- Η εκτυπωτική (Printer).
- Η διατρητική δελτίων (Card Punch).
- Η διατρητική χαρτοταινίας (Papertape Punch).
- Ο οπτικός αναγνώστης (Optical Character Reader).
- Η γραφομηχανή επικοινωνίας (Console Typewriter).
- Η οθόνη (Data Display Unit).
- Η μονάδα σχεδιάσεως (Graph Plotter).

Μαγνητικές μονάδες είναι:

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

Από τις μονάδες αυτές άλλες χρησιμοποιούνται αποκλειστικά και μόνο ως μονάδες εισόδου, όπως η αναγνωστική δελτίων, η αναγνωστική χαρτοταινίας, ο οπτικός αναγνώστης και ο αναγνώστης χαρακτήρων μαγνητικής μελάνης, άλλες δε χρησιμοποιούνται μόνο ως μονάδες εξόδου, όπως η εκτυπωτική, η διατρητική δελτίων ή χαρτοταινίας, ή σχεδιαστική. Μονάδες μικτές είναι η μονάδα μαγνητικής ταινίας, μαγνητικού δίσκου, μαγνητικού τυμπάνου, η οθόνη και η γραφομηχανή επικοινωνίας.

Παρακάτω θα εξετάσουμε χωριστά κάθε φορέα πληροφοριών και θα περιγράψουμε τη μονάδα πάνω στην οποία τοποθετείται.

3.4 Το δελτίο (Card).

Ο πιο γνωστός και διαδεδομένος φορέας πληροφοριών είναι το δελτίο. Υπάρ-

χουν διάφορα είδη δελτίων, που περιγράφομε στη συνέχεια.

α) Δελτίο 80 σπηλών.

Είναι η πιο γνωστή μορφή δελτίου. Κατασκευάζεται από ανθεκτικό μη ηλεκτρα-γιγαντό χαρτί και έχει διαστάσεις $7 \frac{3}{8}'' \times 3 \frac{1}{4}''$ ¹ και πάχος 0,007''. Τα δεδομένα καταγράφονται έπάνω του με τη μορφή μικρών αρθρογωνίων τρυπών. Κάθε δελτίο χωρίζεται κατακόρυφα σε 80 στήλες και οριζόντια σε 12 γραμμές (σχ. 3.4a). Άρα το δελτίο αυτό διαθέτει 960 (80 x 12) συνολικά θέσεις διατρήσεως. Σε κάθε στήλη καταχωρίζεται ένας μόνο χαρακτήρας, δηλαδή σε ένα δελτίο μπορούν να καταχωρίσθούν 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 μια τρύπα στην αντίστοιχη πάντοτε γραμμή του ψηφιακού τμήματος, π.χ. ο αριθμός 8 θα παριστάνεται με μία τρύπα στη γραμμή 8 κάποιας στήλης.

Κάθε αλφαριθμητικός 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. 3.4b).

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

Παράδειγμα

Στο δελτίο του σχήματος 3.4γ βλέπουμε καταχωρισμένα:

- α) Στις στήλες 1-6 τον αριθμό 123661.
- β) Στις στήλες 11-17 το όνομα ΒΑΛΑΚΟΣ.
- γ) Στις στήλες 24-33 το όνομα ΑΝΑΣΤΑΣΙΟΣ.

Όλα αυτά υποτίθεται ότι αποτελούν πληροφορίες.

β) Δελτίο 90 σπηλών.

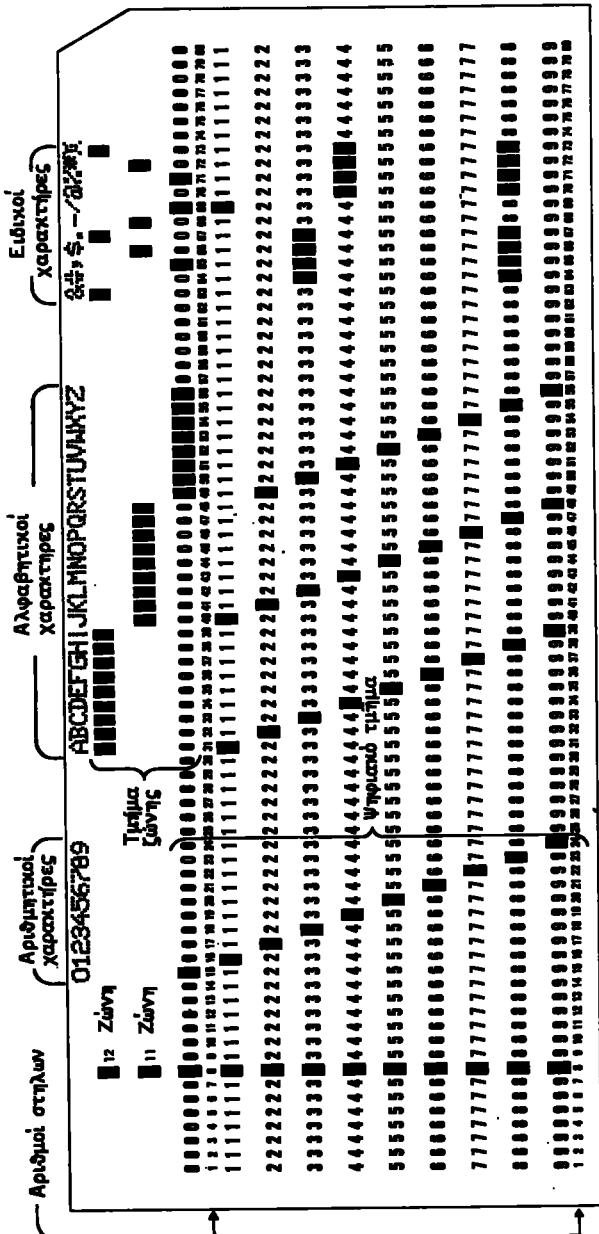
Το δελτίο αυτό χωρίζεται σε 90 στήλες, που όμως δεν βρίσκονται στην ίδια ευθεία, αλλά σε δύο οριζόντια διαζώματα με 45 στήλες στο καθένα (1 - 45 και 46 - 90) (σχ. 3.4d).

Κάθε διάζωμα υποδιαιρείται σε 6 γραμμές. Επομένως το δελτίο αυτό διαθέτει 540 (90 x 6) συνολικά θέσεις διατρήσεως. Σε κάθε στήλη καταχωρίζεται ένας χαρακτήρας, σύμφωνα με τον κώδικα διατρήσεως του σχήματος 3.4d. Δηλαδή συνολικά μπορούν να καταχωρισθούν 90 χαρακτήρες. Το δελτίο αυτό δεν χρησιμοποιείται σήμερα πολύ.

¹ Επειδή πολλά από τα φυσικά μεγέθη που θα αναφερόμαστε έκφραζονται συχνά διεθνώς στο Αγγλοσαξονικό 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.
Το δεκτίο των 80 στηλών.

	ΧΑΡΑΚΤΗΡΕΣ	ΚΩΔΙΚΑΣ ΔΙΑΤΡΗΣΩΝ
ΑΡΙΘΜΗΤΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
ΑΛΦΑΒΗΤΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ	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	12.1 12.2 12.3 12.4 12.5 12.6 12.7 12.8 12.9 11.1 11.2 11.3 11.4 11.5 11.6 11.7 11.8 11.9 0.2 0.3 0.4 0.5 0.6 0.7 0.8 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 σε ένα 80σπηλο δελτίο.

Σχ. 3.4γ. Παράδειγμα διατρυπημένου 80στροφού δελτίου.

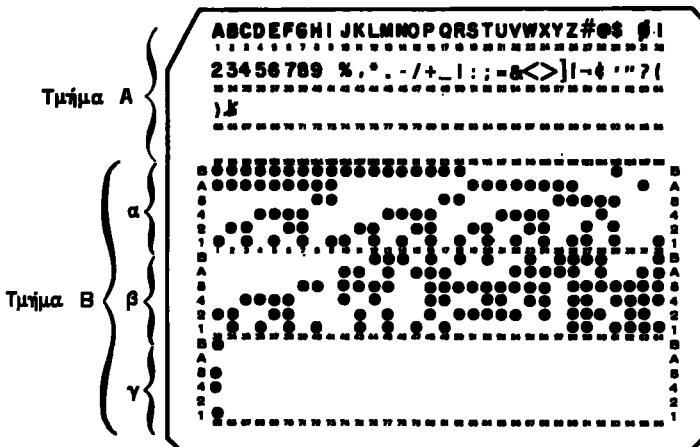


8 1 2 3 4 5 6 7 8 9

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			
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			
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		
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			
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			
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			
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
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		

Зг. 345.

То бетто тув 90 орнамен.



Σχ. 3.4ε.
Το δελτίο των 96 σπηλών.

γ) Δελτίο 96 σπηλών.

Το δελτίο των 96 σπηλών πρωτοπαρουσιάστηκε γύρω στα 1970. Είναι πιο κοντό ($2,63'' \times 3,25''$), από το ογδοντάστηλο δελτίο. Χωρίζεται οριζόντια σε δύο τμήματα (σχ. 3.4ε). Στο επάνω τμήμα (Α) γίνεται η έκτυπωση των χαρακτήρων που μπορούν να διατρηθούν. Στο κάτω τμήμα (Β) γίνονται οι διατρήσεις. Το τμήμα Α αποτελείται από 4 οριζόντιες ζώνες, όπου μπορούν να έκτυπωθούν 4 σειρές από χαρακτήρες. Το τμήμα Β των διατρήσεων χωρίζεται οριζόντια σε τρία διαζώματα: το επάνω (α) που περιλαμβάνει τις στήλες 1 - 32, το μεσαίο (β) που περιέχει τις στήλες 33 - 64 και το κάτω (γ) με τις στήλες 65 - 96 (σχ. 3.4ε).

Κάθε χαρακτήρας απεικονίζεται με ένα συνδυασμό από διατρήσεις. Ο κώδικας διατρήσεως είναι ο θψηφίος αλφαριθμητικός (BA 84 21). Η ύπαρξη τρύπας αντιστοιχεί στο δυαδικό ψηφίο 1. Ανυπαρξία τρύπας ισοδύναμει με το δυαδικό ψηφίο 0.

3.5 Μερικές χρήσιμες έννοιες.

Είναι χρήσιμο στο σημείο αυτό να επιχειρήσουμε να αναλύσουμε μερικές έννοιες, που θα τις χρησιμοποιήσουμε συχνά πιο κάτω:

Πρώτη και βασική είναι η έννοια του *ἀρχείου* (File). Σαν αρχείο εννοούμε μια οργανωμένη συλλογή από στοιχεία και πληροφορίες. Ας υποθέσουμε π.χ. ότι θέλομε να προβούμε στη μισθοδοσία των υπαλλήλων μιας έπιχειρήσεως. Στην περίπτωση αυτή πρέπει να συγκεντρώσουμε ορισμένα στοιχεία, που αφορούν σε κάθε υπάλληλο και που έχουν σχέση με τη μισθοδοσία του, όπως το ονοματεπώνυμό του, τις βασικές αποδοχές του, τις διάφορες κρατήσεις στο μισθό του, τα επιδόματα που του παρέχονται κ.ο.κ. Όλα αυτά τα στοιχεία ταξινομημένα κατάλληλα αποτελούν ένα αρχείο, το αρχείο μισθοδοσίας.

Το ίδιο θα συνέβαινε αν θέλαμε να παρακολουθήσουμε την κίνηση των υλικών

που βρίσκονται στις αποθήκες κάποιου εργοστασίου. Για κάθε είδος υλικού θα πρέπει να συγκεντρώσουμε πληροφορίες που αναφέρονται σ' αυτό, όπως τὸν κωδικό του αριθμό, την ποσότητα που υπάρχει, την τιμή μονάδας του κ.ο.κ. Και πάλι δύλα αυτά τα στοιχεία, που έχουμε συγκεντρώσει για όλα τα είδη των υλικών, που υπάρχουν στις αποθήκες, ταξινομημένα κατάλληλα, αποτελούν ένα άλλο αρχείο, το αρχείο των ύλικών.

Ανάλογη διαδικασία μπορεί να χρησιμοποιηθεί για την επεξεργασία πληροφοριών με τον ηλεκτρονικό υπαλογιστή.

Όλες οι πληροφορίες που περιέχονται σε ένα αρχείο καταχωρίζονται πάνω σε κάποιο φορέα, όπως τα δελτία, τη μαγνητική ταινία, το δίσκο κλπ.

Ας δούμε τώρα πως είναι οργανωμένο ένα αρχείο. Τα στοιχεία ή οι πληροφορίες, που αναφέρονται σε ένα συγκεκριμένο πρόσωπο ή πράγμα (δηλ. τα στοιχεία που αφορούν σε ένα υπάλληλο, στο παράδειγμα της μισθοδοσίας ή τα στοιχεία που έχουν σχέση με ένα υλικό, όπως στο παράδειγμα του αρχείου υλικών) συγκροτούν ξεχωριστές ομάδες, που τις ονομάζομε λογικές ενότητες (*Records*). Με άλλα λόγια το αρχείο μισθοδοσίας θα περιέχει τόσες λογικές ενότητες όσοι είναι και οι υπάλληλοι της έπιχειρήσεως, ή το αρχείο υλικών θα περιέχει τόσες λογικές ενότητες όσα και τα είδη των υλικών που βρίσκονται στις αποθήκες¹.

Η λογική ενότητα σχεδιάζεται έτσι, ώστε οι πληροφορίες που θα υπάρχουν σ' αυτή να κατέχουν συγκεκριμένες περιοχές της. Οι περιοχές αυτές, που μέσα στη κάθε μια υπάρχει μια μόνο μορφή πληροφορίας, ονομάζονται *πεδία* (*Fields*).

Στο παρακάτω σχήμα 3.5α δίνεται ένα παράδειγμα σχεδιασμένης λογικής ενότητας ενός φανταστικού αρχείου μισθοδοσίας, που αποτελείται από δελτία.

ΟΝΟΜΑΤΕΠΩΝΥΜΟ	ΟΝΟΜΑ ΠΑΤΡΟΣ	ΒΑΣΙΚΟ ΕΠΙΔΗΜΙΟΣ ΚΡΑΤΟΣ	ΕΠΙΔΗΜΙΟΣ ΚΡΑΤΟΣ
ΑΡΙΣΤΑΡΧΟΣ ΜΗΤΡΟΣ ΠΤΠΙΔΗΜΙΟΣ ΚΡΑΤΟΣ			

Σχ. 3.5α.

Παράδειγμα σχεδιασμένης λογικής ενότητας ενός φανταστικού αρχείου μισθοδοσίας.

¹ Εδώ έχουμε απλοποιήσει τις έννοιες των λογικών ενστήτων (*Records*) για να γίνουν κατανοητές. Στην πραγματικότητα αντιμετωπίζεται τό ενδεχόμενο να υπάρχουν δύο ή περισσότερα *Records* για κάθε πρόσωπο ή πράγμα (κυρίως στις περιπτώσεις αρχείων, που αποτελούνται από δελτία).

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

Για κάθε υπάλληλο υπάρχει μία λογική ένότητα του τύπου αυτού. Όλες οι λογικές ένότητες ενός αρχείου πρέπει να έχουν την ίδια μορφή και να περιέχουν τα ίδια πεδία στις ίδιες σχετικές θέσεις. Φυσικά το περιεχόμενό τους θα αλλάζει από υπάλληλο σε υπάλληλο δηλαδή από λογική ένότητα σε λογική ένότητα.

Κάθε πεδίο έχει ένα συγκεκριμένο μήκος, δηλαδή μπορεί να χωρέσει σε αυτό ορισμένο πλήθος από χαρακτήρες. Το πρώτο πεδίο στο παράδειγμά μας είναι το πεδίο του αριθμού μητρώου του υπαλλήλου και έχει μήκος 5 χαρακτήρες. Μέσα στο πεδίο αυτό καταχωρίζεται κάθε φορά ένας πενταψήφιος αριθμός που είναι ο αριθμός μητρώου του υπαλλήλου. Στο δεύτερο πεδίο, που έχει μήκος 25 χαρακτήρες, καταχωρίζεται το ονοματεπώνυμό του κ.ο.κ.

Σε ορισμένα πεδία καταχωρίζομε μόνο αριθμητικά δεδομένα και γι' αυτό ονομάζονται αριθμητικά πεδία. Σε άλλα πάλι πεδία καταχωρίζομε μόνο αλφαριθμητικούς χαρακτήρες και τα ονομάζομε αλφαριθμητικά πεδία. Στην περίπτωση που μέσα σ' ένα πεδίο καταχωρίζομε και αλφαριθμητικούς και αριθμητικούς χαρακτήρες, το πεδίο ονομάζεται αλφαριθμητικό (Alphanumeric Field). Έτσι στο παραπάνω παράδειγμα, τα πεδία: αριθμός μητρώου, βασικός μισθός, κρατήσεις και έπιδρματα, είναι αριθμητικά, ενώ τα πεδία: ονοματεπώνυμο και όνομα πατρός είναι αλφαριθμητικά.

Στο σχήμα 3.5β βλέπομε μερικές από τις λογικές ένότητες που ανήκουν στο αρχείο μισθοδοσίας του παραδείγματός μας.

00142 ΠΕΛΗΝΟΣ ΔΙΚΗΠΡΙΟΣ ΒΑΣΙΛΕΙΟΣ 17555 3841 1975	00066 ΓΕΩΠΟΝΙΚΟΥ ΕΦΗΜΑΣ ΝΙΚΟΛΑΟΣ 15678 2954 1367	00054 ΔΙΟΙΚΗΣΗ ΗΓΕΤΕΛΙΟΥΣ ΠΑΥΛΟΣ 16844 3568 1997	00051 ΙΩΑΝΝΙΤΗΣ ΜΙΧΑΗΛ ΙΩΑΝΝΗΣ 14568 2931 1857	00001 ΑΝΤΙΓΗΙΑΣΗΣ ΓΕΩΓΓΙΩΣ ΠΕΤΡΟΣ 11245 1852 1636	
11111 11111	22222 22222	33333 33333	44444 44444	55555 55555	66666 66666
77777 77777	88888 88888	99999 99999	00000 00000	1954	

Σχ. 3.5β.

Μερικές από τις λογικές ένότητες που ανήκουν στο αρχείο μισθοδοσίας του παραδείγματός μας.

Εξ αιτίας του σταθερού μήκους των πεδίων σε όλες τις λογικές ενότητες του αρχείου, υπάρχει πιθανότητα να μη καταλαμβάνονται, σε μερικές από αυτές όλες οι θέσεις τους από χαρακτήρες όπως κάλλιστα μπορούμε να διαπιστώσουμε στο παράδειγμα του σχήματος 3.5β.

Σε ορισμένες μορφές αρχείων, όπως αυτά που βρίσκονται πάνω σε μαγνητική ταινία, δίσκο ή τύμπανο, οι λογικές ενότητες τους μπορούν να ομαδοποιηθούν και να συγκροτήσουν μεγαλύτερες ενότητες, που ονομάζονται **φυσικές ενότητες** (Blocks). Στην περίπτωση των φυσικών ενοτήτων χρησιμοποιούμε την έννοια του συντελεστή ή παράγοντα ομαδοποίησεως (Blocking Factor ή B.F), για να δεξιούμε το πλήθος των λογικών ενοτήτων που περιέχονται σε αυτές.

3.6 Κλασσικές μηχανές.

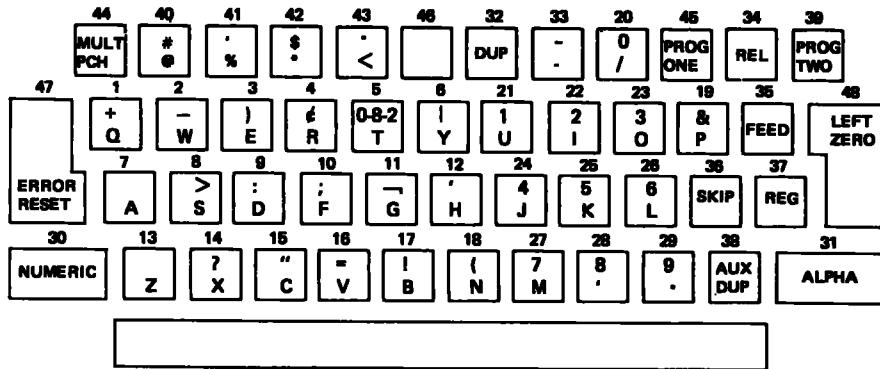
Ως κλασσικές μηχανές χαρακτηρίζομε τις μηχανές, που διατρυπούν ή επεξεργάζονται διάτρητα δελτία, αλλά που δεν συνδέονται με το κύριο συγκρότημα του υπολογιστή.¹ Ο ρόλος τους είναι καθαρά βοηθητικός και προπαρασκευαστικός.

α) Διατρητικές μηχανές (Keypunching machines).

Με τη βοήθεια των μηχανών αυτών τα δεδομένα, που έχουν συγκεντρωθεί και ένταση γραμμένα πάνω σε ειδικά σχεδιασμένα έντυπα, τα **παραστατικά**, διατρυπώνται σε δελτία. Διαθέτουν πληκτρολόγιο όμοιο με το πληκτρολόγιο μιας κοινής ηλεκτρικής γραφομηχανής (σχ. 3.6α) και η ταχύτητα διατρήσεως εξαρτάται από τις ικανότητες του χειριστή.



¹ Με τον όρο υπολογιστής θα εννοούμε από εδώ και εμπρός τον ψηφιακό ηλεκτρονικό υπολογιστή.



Σχ. 3.6α.
Το πληκτρολόγιο μιας διατρητικής μηχανής.

Η μηχανή λειτουργεί με τον έξις τρόπο. Τοποθετούνται τα λευκά αδιάτρητα δελτία στην υποδοχή 1 (Card Reader) [σχ. 3.6β(α)]. Με την πίεση ενός πλήκτρου, ένα δελτίο οδηγείται στο σταθμό διατρήσεως 2 (Punching station). Με κάθε νέα πίεση ενός πλήκτρου χαρακτήρα, ο αντίστοιχος χαρακτήρας διατρυπάται σε μία στήλη, ενώ το δελτίο προχωρεί μια θέση προς τα εμπρός, έτοιμο για να διατρυπηθεί ο επόμενος χαρακτήρας στην επόμενη στήλη κ.ο.κ. Όταν διατρυπηθούν όλοι οι χαρακτήρες, που προβλέπεται να απεικονισθούν σ' αυτό, το δελτίο οδηγείται στο σταθμό αναγνώσεως 3 (Reading station). Ο σκοπός που εξυπηρετεί ο σταθμός αναγνώσεως, είναι να διαβασθούν τα δεδομένα, που μόλις πριν διατρήθηκαν, ώστε να αποτυπωθούν, αν το επιθυμούμε, όλα ή μερικά, στο δελτίο που ακολουθεί, χωρίς να χρειάζεται να επαναληφθεί η ίδια πληκτρολόγηση.

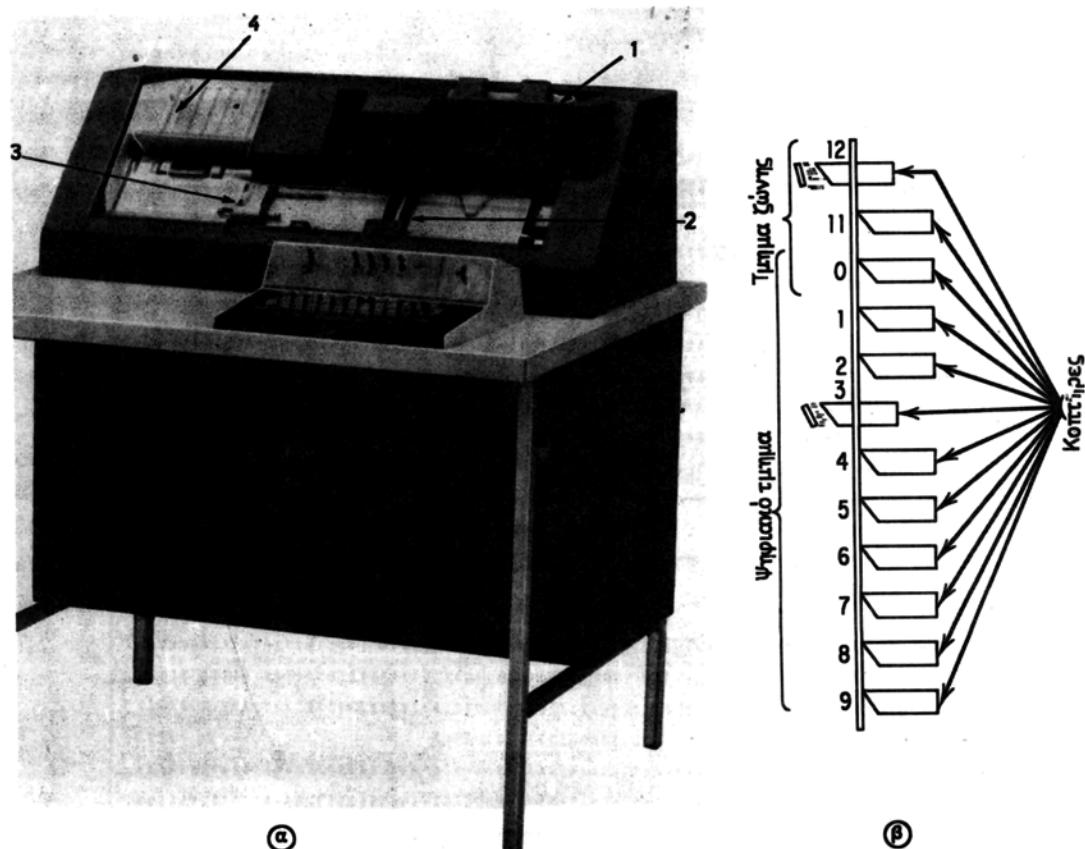
Μετά το σταθμό αναγνώσεως, το δελτίο οδηγείται στη χοάνη εξόδου (Card stacker) 4.

Μερικοί τύποι διατρητικών μηχανών διαθέτουν μνήμη, όπου αποθηκεύονται προσωρινά όλοι οι χαρακτήρες, που πληκτρολογούνται, και που πρόκειται να διατρηθούν σ' ένα δελτίο. Με την πίεση κατόπιν ενός άλλου πλήκτρου όλοι οι χαρακτήρες διατρυπώνται αμέσως και το δελτίο οδηγείται στη χοάνη εξόδου.

β) Επαληθευτική μηχανή (Verifier or Verifying machine).

Στα δελτία που διατρήθηκαν από τη διατρητική μηχανή είναι ενδεχόμενο να έχουν γίνει σφάλματα. Τα σφάλματα αυτά μπορούν να εντοπισθούν με τη βοήθεια τής επαληθευτικής μηχανής.

Η μηχανή αυτή διαθέτει όμοιο πληκτρολόγιο με τη διατρητική, δεν κάνει όμως διατρήσεις. Ο χειριστής ξανακτυπά στο πληκτρολόγιο τους χαρακτήρες, που πρέπει να έχουν διατρηθεί στο δελτίο. Σε περίπτωση που έχει διατρηθεί σε κάποια στήλη λανθασμένος χαρακτήρας, η μηχανή τον έπισημανεί δημιουργώντας μια εγκοπή στο δελτίο και πάνω από τη στήλη, όπου ανιχνεύθηκε ο λάθος χαρακτήρας [σχ. 3.6γ(α)]. Αν όλο το δελτίο έχει διατρηθεί σωστά, τότε η μηχανή δημιουργεί μια εγκοπή κοντά στην επάνω δεξιά γωνία του δελτίου [σχ. 3.6γ(β)].



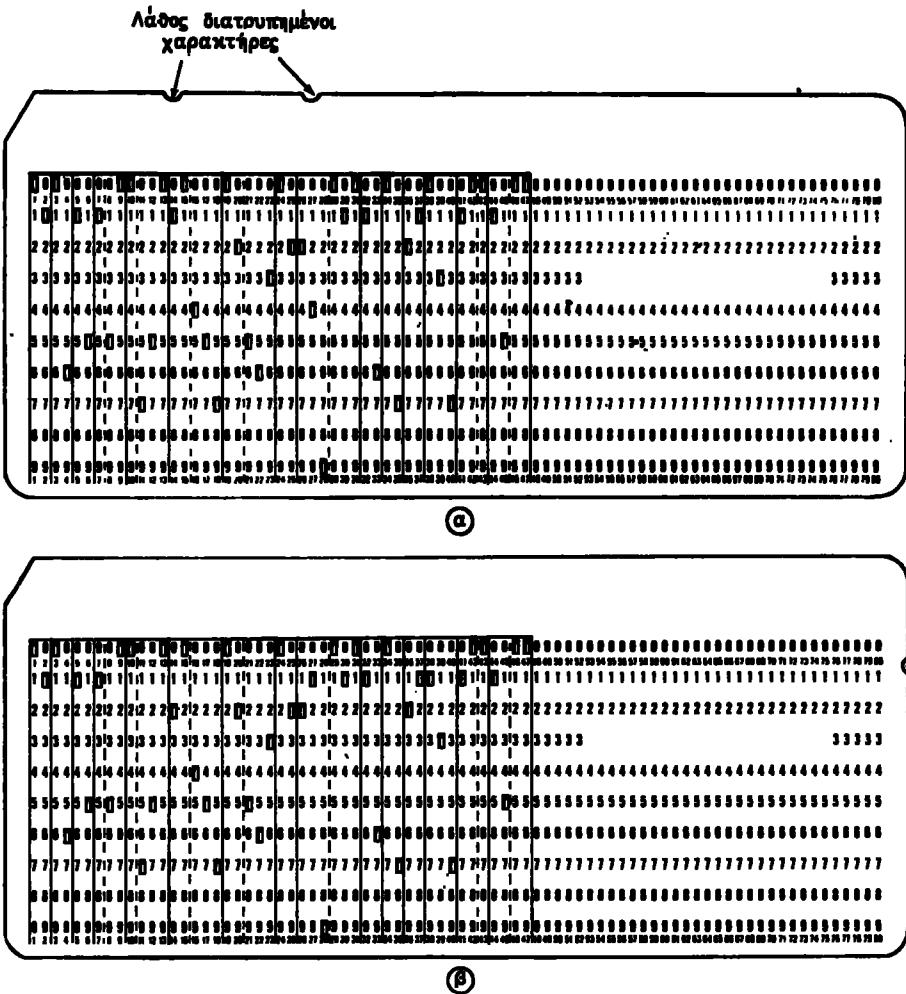
γ) Μεταφραστική μηχανή (Interpreter).

Η μηχανή αυτή εκτυπώνει στό επάνω λευκό περιθώριο του δελτίου τους χαρακτήρες που έχουν διατρηθεί σ' αυτό.

Πολλές μηχανές μπορεί να είναι ταυτόχρονα διατρητικές και μεταφραστικές, άλλες δε μπορούν να χρησιμοποιηθούν και ως επαληθευτικές.

δ) Διαλογική μηχανή (Card sorter).

Πολλές φορές τα δελτία με τα δεδομένα πρέπει να τα ταξινομήσουμε προτού εισαχθούν για επεξεργασία στον υπολογιστή, να τα βάλομε δηλαδή σε κάποια καθορισμένη σειρά. Η ταξινόμηση αυτή των δελτίων ονομάζεται **διαλογή** (Sort) και μπορεί να είναι αριθμητική ή αλφαριθμητική και ακόμα αυξουσα ή φθίνουσα, γίνεται δε με τη βοήθεια των διαλογικών μηχανών (Sorters) [σχ. 3.6δ(α)]. Η μηχανή αυτή ταξινομεί τα δελτία σύμφωνα με το περιεχόμενο ενός ή περισσοτέρων πεδίων (Sort Key Fields). Έτσι αν π.χ. υπάρχει ένα πεδίο σε κάθε δελτίο, όπου καταχωρίζεται κάποιο όνομα, με τη διαλογική μηχανή μπορούμε να τοποθετήσουμε τα δελτία

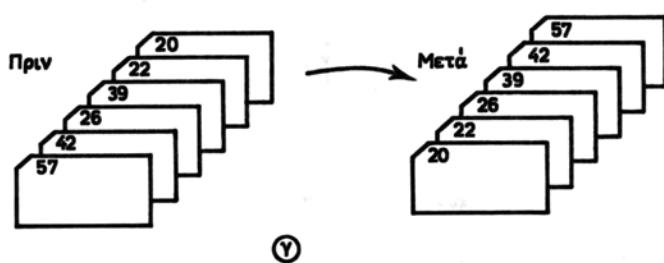
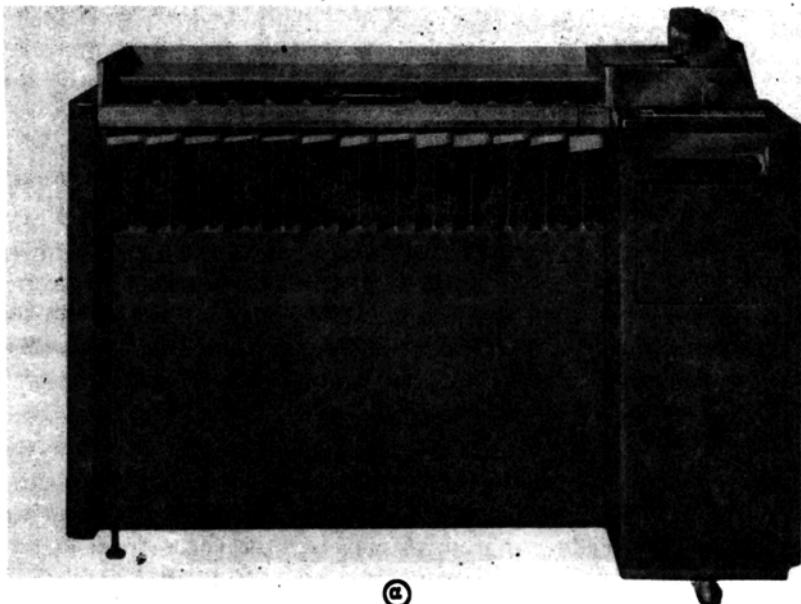


Σχ. 3.6γ.
(a) Λάθος διάτρηση του δελτίου (β) Σωστή διάτρηση του δελτίου.

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

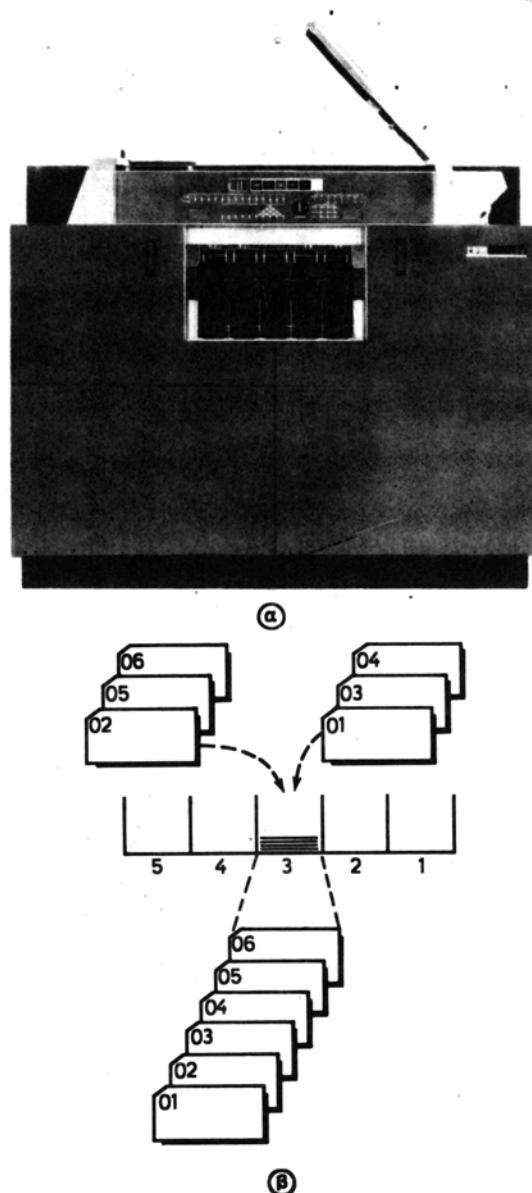
ε) Συζευκτική μηχανή (Collector).

Ας υποθέσουμε ότι μας έχουν δοθεί δύο ομάδες δελτίων, που κάθε μία έχει ταξινομηθεί ως προς το ίδιο πεδίο με τον ίδιο τρόπο (αύξουσα ή φθίνουσα σειρά). Με τη βοήθεια της μηχανής αυτής [σχ. 3.6ε(α)] μπορούμε να συνενώσουμε αυτές τις δύο ομάδες σε μία, όπου τα δελτία θα παρουσιάζονται και πάλι ταξινομημένα



Σχ. 3.65.

- (α) Η διαλογική μηχανή, (β) Αλφαβητική ταξινόμηση κατά αύξουσα σειρά (γ) Αριθμητική ταξινόμηση κατά αύξουσα σειρά.



Σχ. 3.8e.

(a) Η Συζευκτική μηχανή, (β) Ένα παράδειγμα αριθμητικής συζεύξεως.

ως προς το ίδιο πεδίο-κλειδί και με τον ίδιο τρόπο όπως και οι αρχικές [σχ. 3.6ε(β)].

στ) Αναπαραγωγική μηχανή (Reproducer).

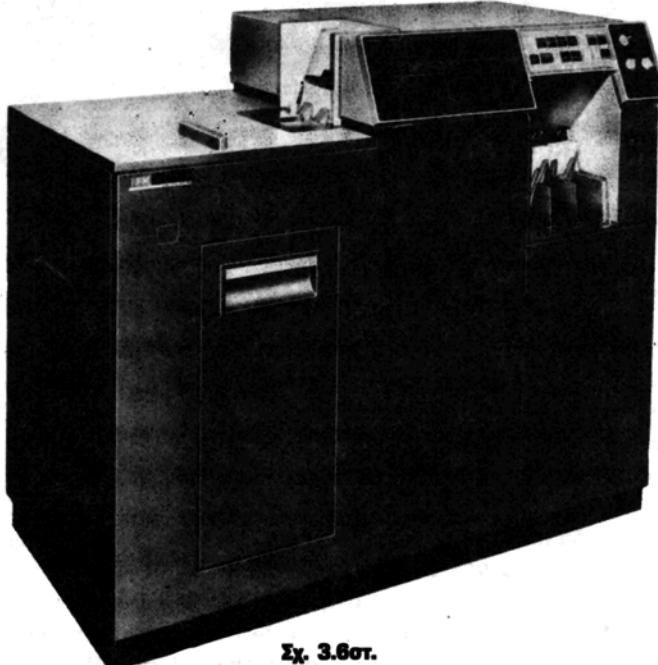
Η μηχανή αυτή εκτελεί διάφορες εργασίες όπως:

- **Απλή αναπαραγωγή.** Με αυτήν αντιγράφεται αυτόματα ολόκληρο η μέρος από το περιεχόμενο ενός δελτίου σε ένα άλλο λευκό.
- **Ομαδική διάτρηση.** Με αυτήν διατρυπάται ένα δεδομένο, που είναι κοινό σε μια 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) Απλή 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 απλές συγκρίσεις δεδομένων.
- Η διάτρηση δεδομένων και αποτελεσμάτων σε λευκά δελτία.

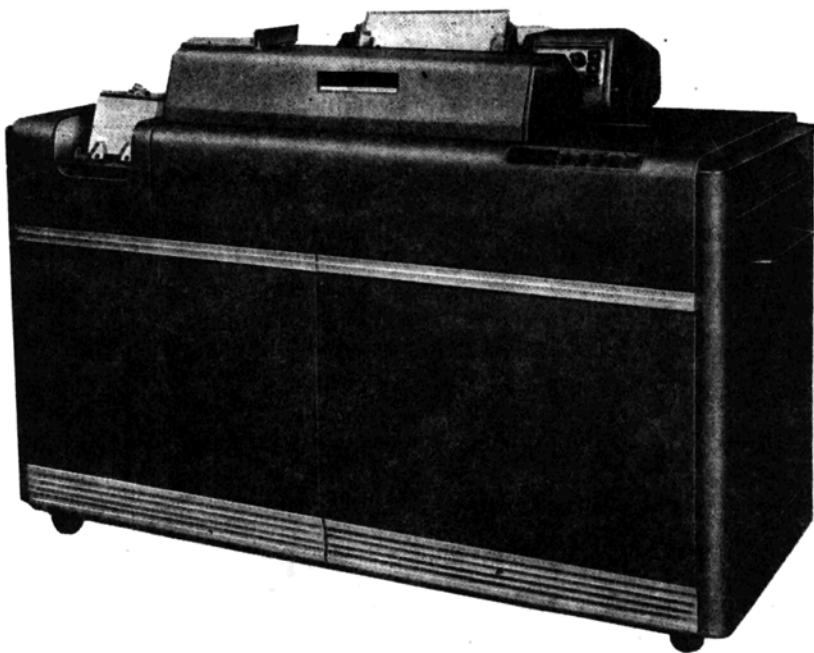


Σχ. 3.6στ.
Η απλή υπολογιστική μηχανή.

Η υπολογιστική μηχανή δέν δέχεται πρόγραμμα με εντολές, αλλά οι λειτουργίες της κατευθύνονται από ένα ειδικό πίνακα, στον οποίο έχει γίνει προγρουμένως 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 (η μόνη από τις κλασσικές μηχανές που μπορεί για τυπώσει) (σχ. 3.6ζ').



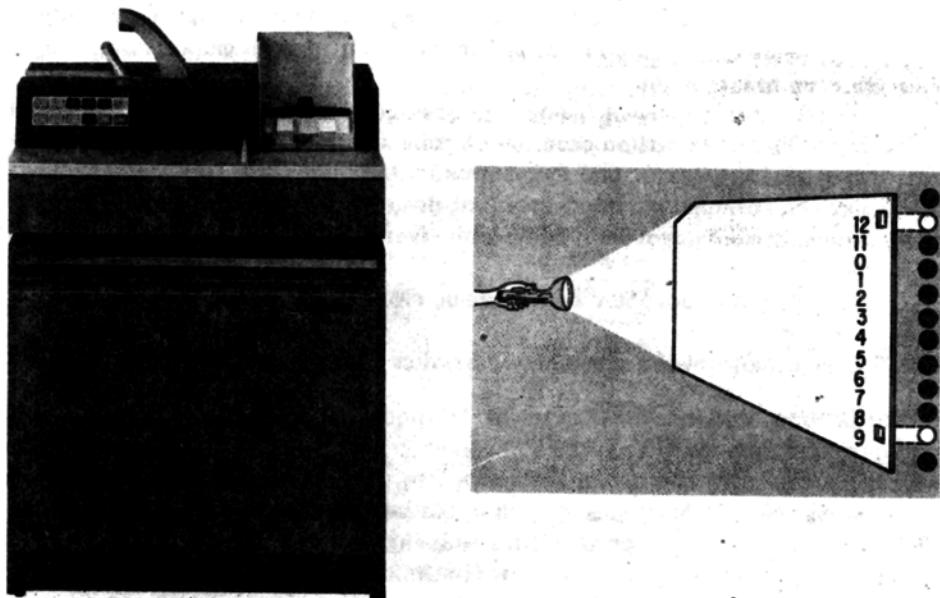
Σχ. 3.6ζ.
Η πινακοποιητική μηχανή.

3.7 Περιφερειακή μονάδα αναγνώσεως διατρίτων δελτίων (Card reader).

Τα δεδομένα που έχουν διατεθεί επάνω σε δελτία εισάγονται στον υπολογιστή με τη βοήθεια της μονάδας αναγνώσεως διατρίτων δελτίων (σχ. 3.7α).

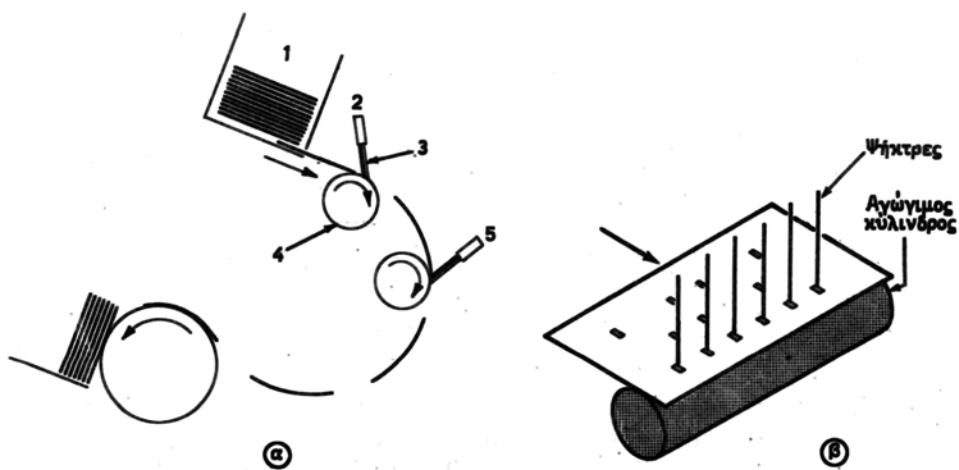
Παρακάτω θα περιγράψουμε τον τρόπο αναγνώσεως σε μια μονάδα σχεδιασμένη για να διαβάζουμε μ' αυτήν 80στήλα δελτία.

- Τα διατρημένα δελτία τοποθετούνται στην υποδοχή 1 της αναγνωστικής (σχ. 3.7β).
- Με μία εντολή από τη μονάδα ελέγχου, ένα-ένα τα διάτρητα δελτία οδηγούνται στο σταθμό αναγνώσεως 2. Εκεί βρίσκονται σε σειρά επάνω από το δελτίο 80 ψήκτρες 3, όσες και οι στήλες του δελτίου (ανάγνωση κατά στήλη), που ακουμπούν ελαφρά επάνω του. Το δελτίο ολισθαίνει πάνω σ' ένα αγώγιμο κύλινδρο 4 που βρίσκεται κάτω από τις ψήκτρες. Κάθε ψήκτρα ελέγχει την περιοχή μιας στήλης [σχ. 3.7β(β)]. Καθώς το δελτίο κινείται, όπου υπάρχει διάτρηση σε μια στήλη, η αντίστοιχη ψήκτρα έρχεται σ' επαφή με τον αγώγιμο κύλινδρο 4. Σε κάθε τέτοια επαφή κλείνει ένα κύκλωμα και παράγεται ένας μικρής διάρκειας ηλεκτρικός παλμός, που ισοδυναμεί με το δυαδικό ψηφίο 1. Όταν δεν αποκαθίσταται επαφή και επομένως δεν παράγεται ηλεκτρικός παλμός, τότε έχομε το δυαδικό ψηφίο 0.
- Οι παλμοί που παράγονται μεταφέρονται στην κεντρική μνήμη, όπου αποθηκεύονται. Οι αναγνωστικές είναι εφοδιασμένες, τις περισσότερες φορές, και μ' ένα δεύτερο σταθμό αναγνώσεως 5, όπου ελέγχεται το σωστό διάβασμα του δελτίου από τον πρώτο σταθμό.



Σχ. 3.7α.

Μονάδα αναγνώσεως διατρήτων δελτίων (αριστερά) και σχηματικό ανάλογο του τρόπου αναγνώσεως (δεξιά).



Σχ. 3.7β.

(a) Η διαδικασία αναγνώσεως. (b) Ο μηχανισμός ανιχνεύσεως των διατρήσεων με ψήκτρες.

Επειδή τα δεδομένα που βρίσκονται επάνω στο δελτίο είναι κωδικοποιημένα σύμφωνα με τον κώδικα που χρησιμοποιείται, ενώ στη μνήμη θα πρέπει να αποθηκευθούν με τη μορφή κάποιου άλλου κώδικα (θψήφιο αλφαριθμητικό, EBCDIC

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

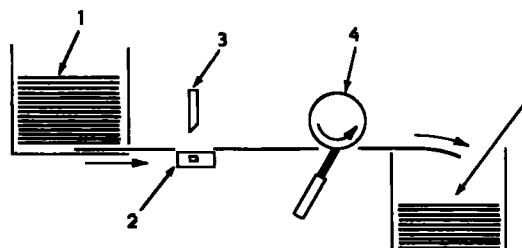
Σε άλλους τύπους αναγνωστικών ο μηχανισμός ανιχνεύσεως με ψήκτρες έχει αντικατασταθεί από μία σειρά φωτοηλεκτρικών κυττάρων. Καθώς το δελτίο κινείται, όταν μια δέσμη φωτός από ένα φωτοκύτταρο συναντήσει μια τρύπα, κλείνει ένα ηλεκτρικό κύκλωμα και παράγεται ένας βραχύχρονος παλμός. Σε μερικούς τύπους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

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

3.8 Περιφερειακή μονάδα διατρήσεως 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 ενσωματωμένη στη μονάδα αναγνώσεως.

Η διαδικασία της διατρήσεως είναι όμοια με τη διαδικασία της αναγνώσεως. Ένα - ένα τα λευκά (αδιάτρητα) δελτία, που είναι τοποθετημένα στην υποδοχή 1 (σχ. 3.8) περνούν από το σταθμό διατρήσεως 2. Εκεί υπάρχει μια σειρά από μαχαιρίδια-κοπτήρες 3. Τη σπιγμή που μερικά από αυτά ενεργοποιούνται, προκαλούν διατρήσεις. Στη συνέχεια το διατρημένο δελτίο οδηγείται στο σταθμό αναγνώσεως-ελέγχου 4, όπου με μια 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 διατρήσεως 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 διατρήσεως έχει ταχύτητα 100 - 600 δελτία το λεπτό.

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

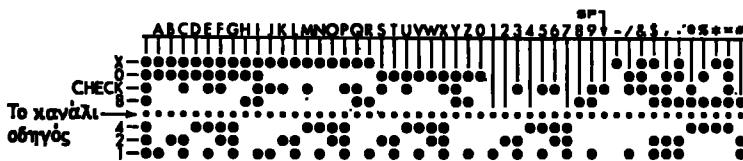
3.9 Η χαρτοταινία (Papertape).

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

Χωρίζεται σε όλο το μήκος της σε 5 ως 8 παράλληλες ζώνες τα κανάλια (tracks or channels). Στο μέσο περίπου του πλάτους της υπάρχει μια επί πλέον ζώνη, που φέρει συνεχή σειρά από μικρές τρύπες και χρησιμεύει για την κίνηση της ταινίας.

Ο αριθμός των καναλιών εξαρτάται από τον κώδικα, που χρησιμοποιείται για τη παράσταση των δεδομένων. Έτσι π.χ. στις χαρτοταινίες με 8 κανάλια, οι χαρακτήρες απεικονίζονται στον βψήφιο αλφαριθμητικό κώδικα, όπου στα 6 ψήφια του κώδικα προστίθεται το ψηφίο ιστοιμίας (x) και ένα ακόμα ψηφίο (EL) που συμβολίζει το τέλος γραμμής.

Σε κάθε στήλη της χαρτοταινίας απεικονίζεται ένας χαρακτήρας. Το σχήμα 3.9 απεικονίζει τον κώδικα διατρήσεων μιας χαρτοταινίας 8 καναλιών. Η πυκνότητα καταγραφής είναι περίπου 10 χαρακτήρες ανά ίντσα (10ch/1").



Σχ. 3.9.

Η κωδικοποίηση των χαρακτήρων στη χαρτοταινία των 8 καναλιών.

Τα πλεονεκτήματα από τη χρήση της χαρτοταινίας είναι ότι δεν κοστίζει ακριβά και μπορούν πολλά δεδομένα να αποθηκευθούν εύκολα και να καταλάβουν μικρό σχετικό δύγκο.

Το κυριότερο μειονέκτημα είναι ότι είναι εξαιρετικά δύσκολη η διόρθωση των χαρακτήρων, που έχουν διατρηθεί λάθος.

3.10 Μηχανή διατρήσεως χαρτοταινίας (Papertape punching machine).

Τα δεδομένα διατρυπώνται στη χαρτοταινία είτε από ειδική μηχανή, που είναι εφοδιασμένη με πληκτρολόγιο ανάλογο προς εκείνο της μηχανής διατρήσεως δελτίων, είτε με τη βοήθεια συσκευών, που προσαρμόζονται σε άλλες μηχανές, όπως το τηλέτυπο, οι λογιστικές και ταμιακές μηχανές κλπ.

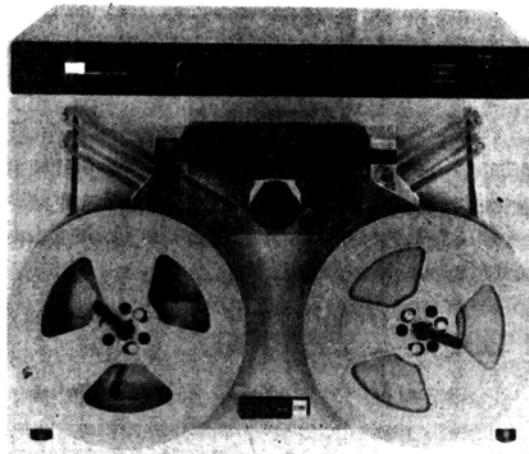
Η μηχανή διατρήσεως χαρτοταινίας δεν συνδέεται με το κεντρικό συγκρότημα του υπολογιστή.

3.11 Περιφερειακή μονάδα αναγνώσεως χαρτοταινίας (Punched papertape reader).

Με τη μονάδα αυτή διαβάζεται η χαρτοταινία και τα δεδομένα μεταβιβάζονται στον υπολογιστή. Η ταχύτητά της είναι 200 - 1200 χαρακτήρες το δευτερόλεπτο (σχ. 3.11).

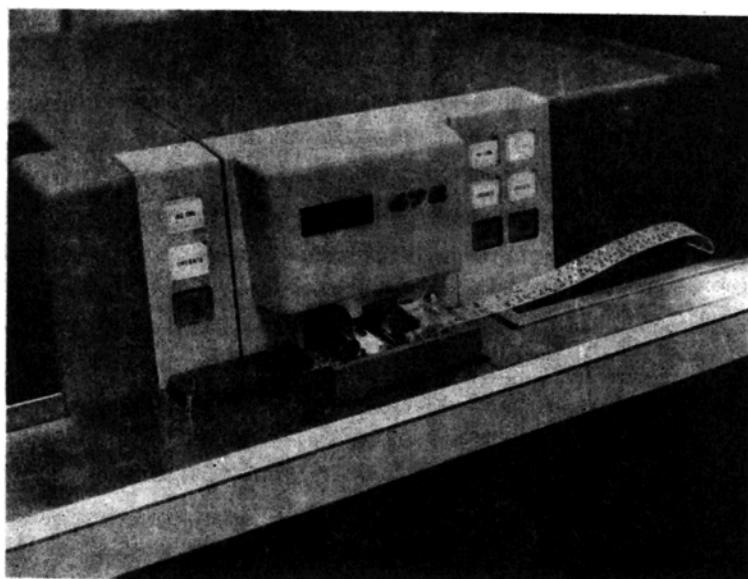
3.12 Περιφερειακή μονάδα διατρήσεως χαρτοταινίας (Papertape punch unit).

Στη μονάδα αυτή διατρυπώνται πάνω σε χαρτοταινία πληροφορίες, που προέρχονται από τον υπολογιστή (σχ. 3.12). Η ταχύτητα διατρήσεως είναι περίπου 100



Σχ. 3.11.
Μονάδα αναγνώσεως διάτρητης χαρτοταινίας.

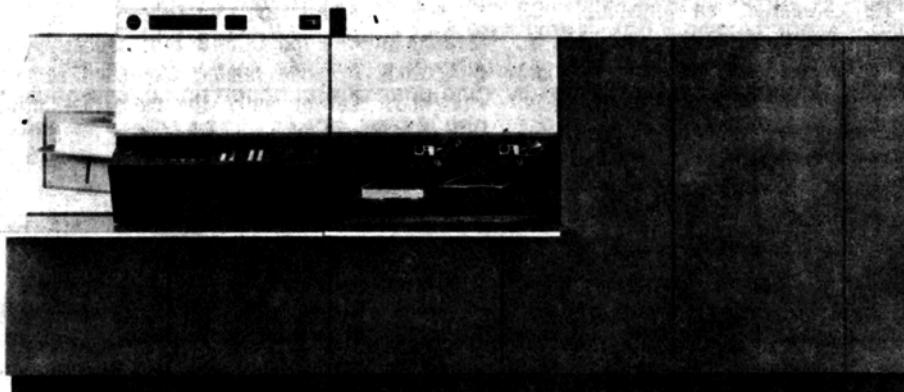
χαρακτήρες το δευτερόλεπτο. Είναι μάλλον αργή μονάδα και γι' αυτό δεν χρησιμοποιείται συχνά. Σε μερικούς τύπους η διατρητική χαρτοταινίας είναι ενσωματωμένη στην αναγνωστική.



Σχ. 3.12.
Μονάδα διατρήσεως χαρτοταινίας.

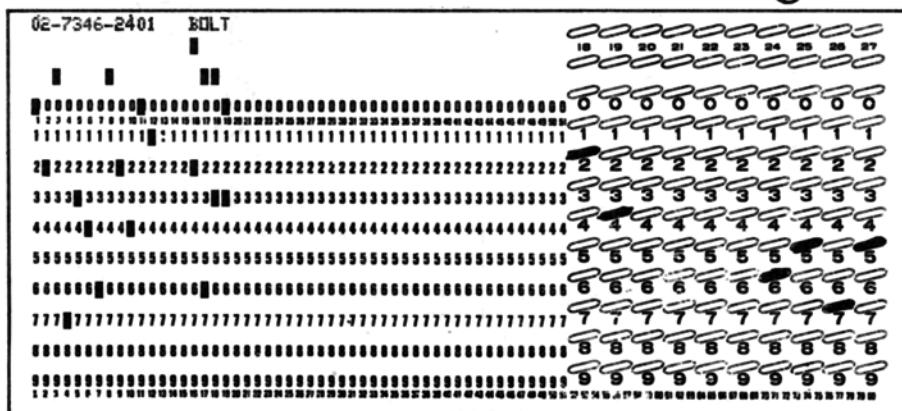
3.13 Οπτικοί αναγνώστες (Optical characters readers) (σχ. 3.13a).

Είναι μονάδες ικανές να διαβάζουν τους χαρακτήρες-δεδομένα που έχουν γρα-



Σχ. 3.13α.
Οπικός αναγνώστης.

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
0 1 2 3 4 5 6 7 8 9
@ , : ; = + / \$ * " & |
' - { } % ? І Й І Й
Ӯ Ӯ Ӱ Ӷ Ӹ ӹ ӻ Ӽ



(a) Ένας τύπος χαρακτήρων που μπορούν για αναγνωρισθούν οπικά. (β) Δεδομένα σημειωμένα ως απλά σημάδια σε ειδικές θέσεις επάνω σε δελτίο.

φει με κοινή γραφή αλλά ορισμένο τύπο [σχ. 3.13β(α)] επάνω σε ειδικά παραστατικά (βλ. παράγραφο 3.6α). Η ταχύτητά τους φθάνει τους 400 χαρακτήρες το δευτερόλεπτο.

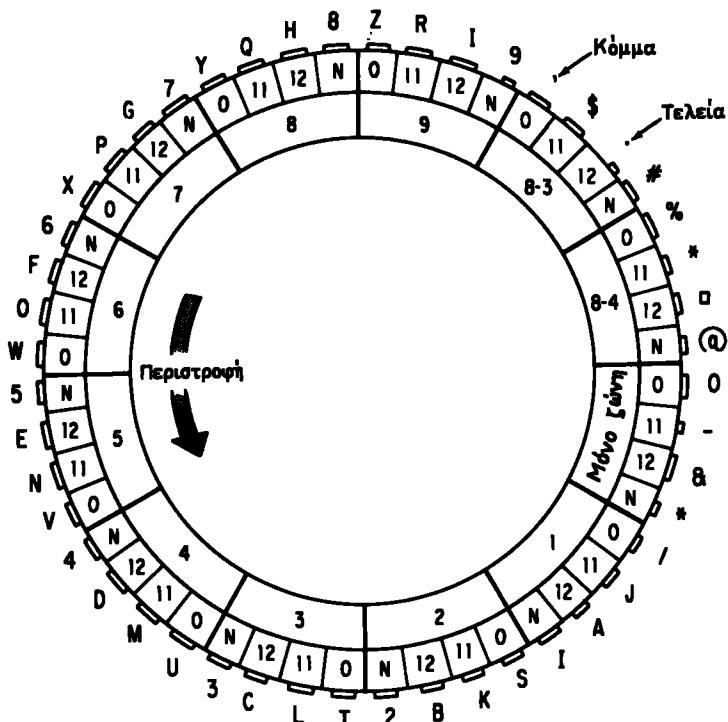
Σε μερικούς τύπους αναγνωστών χρησιμοποιούνται παραστατικά, στα οποία τα δεδομένα έχουν στημειωθεί ως απλά σημάδια σε ειδικές θέσεις επάνω σε δελτία, με κοινό μολύβι (Mark-sensing). Ο αναγνώστης αυτός τα διαβάζει και συγχρόνως τα διατρυπά σε άλλη περιοχή του δελτίου [σχ. 3.13β(β)].

3.14 Περιφερειακή μονάδα εκτυπώσεως (Printer).

Η μονάδα αυτή μπορεί να εκτυπώσει επάνω σε χαρτί, δεδομένα, αποτελέσματα ή άλλες πληροφορίες, που προέρχονται από την κεντρική μονάδα. Είναι μια από τις λίγες περιφερειακές μονάδες, που παρέχει τις πληροφορίες με τέτοιο τρόπο, ώστε να μπορούν να διαβασθούν αμέσως από τον άνθρωπο.

Υπάρχουν δύο βασικοί τύποι εκτυπωτών:

- Με επαφή (Impact), που τυπώνουν τους χαρακτήρες σε κοινό χαρτί με μηχανικό τρόπο.
- Ξωρίς επαφή (Non impact), που σχηματίζουν τους χαρακτήρες επάνω σε ειδικό χαρτί με θερμικό, φωτοχημικό ή ηλεκτροχημικό τρόπο.



Σχ. 3.14α.
Ένας από τους δακτυλίους του εκτυπωτή με δακτυλίους

a) Εκτυπωτές με επαφή (Impact printers).

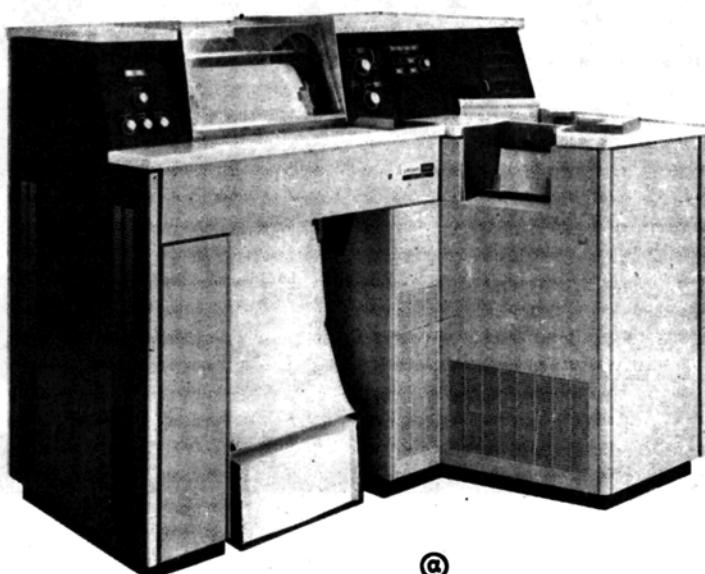
Στην κατηγορία αυτή ανήκουν:

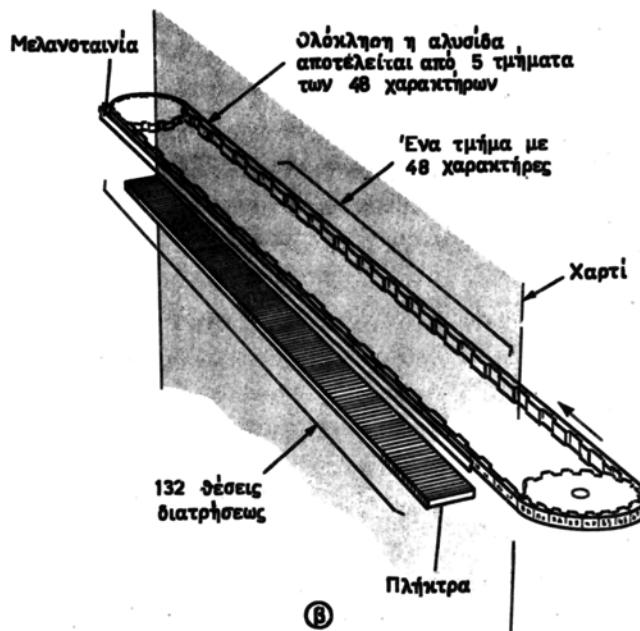
1) **Οι εκτυπωτές με δίσκους ή δακτυλίους (Wheel printers).** Το τμήμα εκτυπώσεως αυτών των μηχανών αποτελείται από πολλούς (συνήθως 120) λεπτούς κυκλικούς μεταλλικούς δακτυλίους, που είναι προσαρμοσμένοι ο ένας δίπλα στον άλλο σ' ένα κοινό οριζόντιο άξονα: ο άξονας αυτός περνά από τα κέντρα τους και μπορεί να τους περιστρέψει. Ο κάθε δίσκος φέρει στην περιφέρειά του ανάγλυφους τους αριθμητικούς, αλφαριθμητικούς και ειδικούς χαρακτήρες (σχ. 3.14a). Το χαρτί, όπου πρόκειται να εκτυπωθούν οι χαρακτήρες, ξετυλίγεται μπροστά από το σύστημα των δακτυλίων. Η ταχύτητά τους είναι σχετικά μικρή (150 πλήρεις γραμμές τών 120 χαρακτήρων το λεπτό).

2) **Εκτυπωτές με αλυσίδα (Chain printers)** [σχ. 3.14β (α)]. Το σύστημα εκτυπώσεως τους αποτελείται από μία οριζόντια αλυσίδα, που χωρίζεται σε 5 τμήματα. Το κάθε τμήμα περιέχει 48 χαρακτήρες [σχ. 3.14β (β)]. Και τα 5 τμήματα περιέχουν την ίδια ομάδα από χαρακτήρες. Η αλυσίδα στρέφεται με μεγάλη ταχύτητα. Μπροστά από αυτήν και σε κάποια απόσταση είναι τοποθετημένα στη σειρά πολλά μικρά σφυριά-πλήκτρα (μέχρι 132). Μεταξύ των πλήκτρων και της αλυσίδας ξετυλίγεται κατακόρυφα το χαρτί. Όταν ένα πλήκτρο ενεργοποιηθεί την κατάλληλη στιγμή, κτυπά το χαρτί επάνω στην αλυσίδα και αποτυπώνει το χαρακτήρα που βρέθηκε εκείνη τη στιγμή μπροστά του.

Η ταχύτητα εκτυπώσεως φθάνει τις 1000 γραμμές το λεπτό.

3) **Εκτυπωτές με τύμπανο (Drum printers).** Το σύστημα εκτυπώσεως τους αποτελείται από ένα κοίλο κυλινδρικό μεταλλικό τύμπανο. Οι διάφοροι χαρακτήρες βρίσκονται ανάγλυφοι στην εξωτερική του επιφάνεια τοποθετημένοι στη σειρά κα-

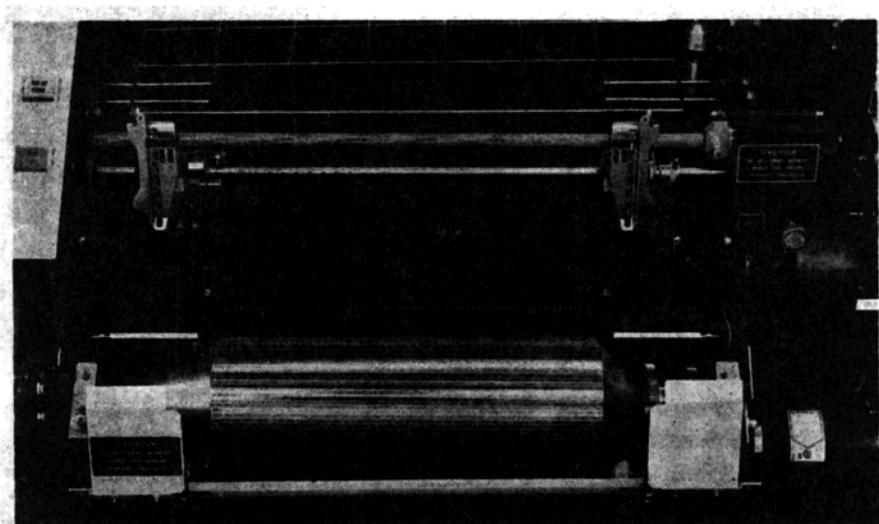




Σχ. 3.14β.

(a) Εκτυπωτής με αλυσίδα. (b) Ο μηχανισμός εκτυπώσεως ενός εκτυπωτή με αλυσίδα.

τα μήκος των γενετειρών του (σχ. 3.14γ). Μπροστά από το τύμπανο υπάρχει πάλι μια σειρά από σφυράκια-πλήκτρα. Ανάμεσα στα πλήκτρα και το τύμπανο ξετυλίγε-



Σχ. 3.14γ.

Το σύστημα εκτυπώσεως ενός εκτυπωτή με τύμπανο.

ται το χαρτί. Το τύμπανο περιστρέφεται με πολύ μεγάλη ταχύτητα (περίπου 1000 στροφές το λεπτό). Η ταχύτητα εκτυπώσεώς τους φθάνει τις 2000 γραμμές το λεπτό και μπορούν να τυπώσουν μέχρι 150 χαρακτήρες σε κάθε γραμμή.

4) *Γραφομηχανή (Typewriter).* Μπορεί να θεωρηθεί και αυτή ως εκτυπωτική διάταξη με πολύ όμως μικρότερη ταχύτητα.

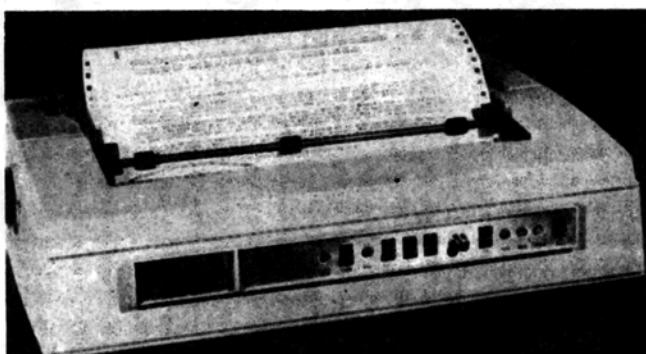
β) Εκτυπωτές χωρίς επαφή (Non impact printers).

Στους εκτυπωτές αυτούς οι διάφοροι χαρακτήρες σχηματίζονται με φωτοχημικό ή θερμικό ή ηλεκτροστατικό τρόπο έπάνω σε φωτοευαίσθητο ή θερμοευαίσθητο χαρτί (σχ. 3.14δ).

Στην κατηγορία αυτή ανήκουν:

- 1) Οι ηλεκτροστατικοί εκτυπωτές (Electrostatic printers).
- 2) Οι θερμικοί εκτυπωτές (Thermal printers).
- 3) Οι ξηρογραφικοί εκτυπωτές (Xerographic printers).
- 4) Οι ηλεκτροοπτικοί εκτυπωτές (Electro - optical printers).

Τα πλεονεκτήματά τους είναι ότι η ταχύτητά τους μπορεί να φθάσει τις 64000 γραμμές το λεπτό και ότι είναι αθρόυστοι. Τα μειονεκτήματά τους είναι ότι κοστίζει πολύ το ειδικό χαρτί που χρησιμοποιούν και ότι δεν μπορούν να τυπώσουν σε περισσότερα από ένα αντίτυπα συγχρόνως.

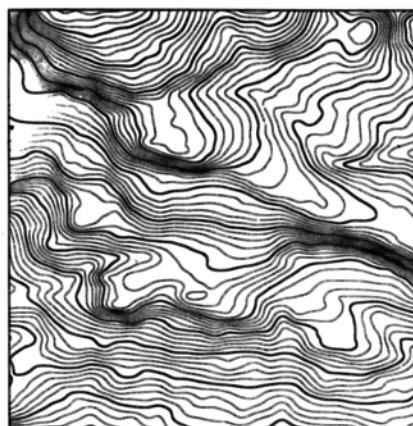


**Σχ. 3.14δ.
Ηλεκτροστατικός εκτυπωτής.**

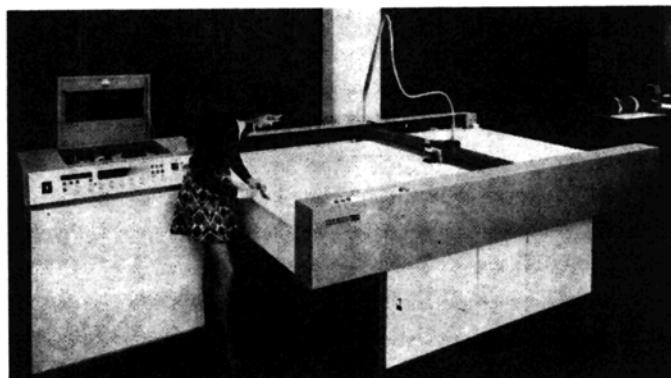
Οι περισσότεροι από τους εκτυπωτές που περιγράψαμε μπορούν να τυπώσουν μια ολόκληρη γραμμή κάθε φορά (Line - at - a - time printers). Άλλες όμως εκτυπωτικές διατάξεις, όπως η ηλεκτρική γραφομηχανή και το τηλέτυπο, τυπώνουν ένα - ένα τους χαρακτήρες κάθε φορά (Character - at - a - time printers).

3.15 Περιφερειακή μονάδα σχεδιάσεως (Graph Plotter).

Στη μονάδα αυτή γίνεται ο σχεδιασμός καμπυλών, διαγραμμάτων και άλλων γραφικών απεικονίσεων [σχ. 3.15(α)] σύμφωνα με πληροφορίες, που προέρχονται



(α)



(β)

Σχ. 3.15.

(α) Ένας τοπογραφικός χάρτης σχεδιασμένος από περιφερειακή μονάδα σχεδιάσεως. (β) Μια περιφερειακή μονάδα σχεδιάσεως (Plotter).

από την κεντρική μονάδα και κάτω από τον έλεγχό της.

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

Χρησιμοποιείται σ' εφαρμογές μηχανικής, αστρονομίας, πυρηνικής φυσικής, τοπογραφίας, ωκεανογραφίας κλπ. Είναι η πιο αργή 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 πράξ των υπολογιστή μηνυμάτων. Οι απαν-

τήσεις του υπολογιστή έκτυπώνονται σε χαρτί που ξετυλίγεται μπροστά από τη μηχανή (σχ. 3.16). Η ταχύτητα έκτυπωσεως είναι μικρή (10 - 30 χαρακτήρες το δευτέρολεπτο) (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.



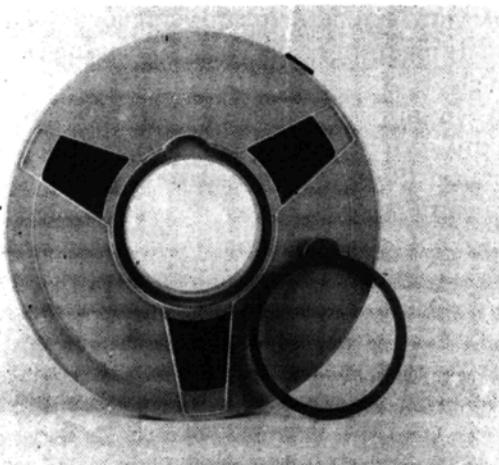
Σχ. 3.16.
Γραφομηχανή επικοινωνίας.

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 η μια της επιφάνεια έχει επιστρωθεί με κάποιο μαγνητικό υλικό. Τυλίγεται γύρω από ένα καρούλι (μπομπίνα), [σχ. 3.17α (α)]. Το πλάτος της είναι $\frac{1}{2}$ " - 1" και το μήκος της κυμαίνεται από 1800 - 3600 ft.

Η ταινία διαιρείται κατά μήκος σε παράλληλες μεταξύ τους ζώνες, τα κανάλια (Channels or tracks). Κάθε χαρακτήρας καταχωρίζεται κωδικοποιημένος κατά το πλάτος της με τη μορφή ορισμένου αριθμού μαγνητικών στίγματων. Η ύπαρξη σε ένα κανάλι ενός μαγνητικού στίγματος ισοδυναμεί με το διαδικτό ψηφίο 1, και η α-



(α)

	Αριθμητικοί χαρακτήρες	Αλφαριθμητικοί χαρακτήρες	Ειδικοί χαρακτήρες
Ψηφίο ισοτιμίας	0 1 2 3 4 5 6 7 8 9	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	8 . - \$ * / , % # @
Τμήμα ζώνης	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
Ψηφιακό τμήμα	2 3 4 5 6 7 8 9	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
	2 3 4 5 6 7 8 9	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
	2 3 4 5 6 7 8 9	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
	2 3 4 5 6 7 8 9	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1

(β)

Σχ. 3.17a.
Μαγνητική ταινία:

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

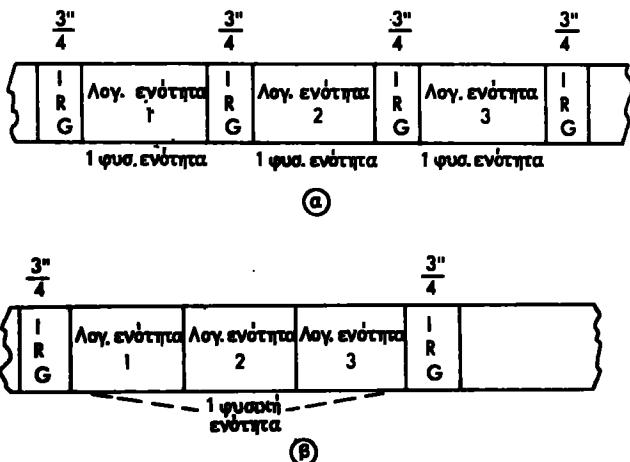
νυπαρξία σε ένα κανάλι μαγνητικού στίγματος ισοδυναμεί με το δυαδικό ψηφίο 0.

Το πλήθος των καναλιών μιας ταινίας εξαρτάται από τον κώδικα, που χρησιμοποιείται για την παράσταση των πληροφοριών. Έτσι, αν π.χ. χρησιμοποιείται ο διψήφιος αλφαριθμητικός κώδικας με δυαδικό ψηφίο ισοτιμίας, τότε η ταινία θα έχει 7 κανάλια. Ο κώδικας αυτός φαίνεται στο σχήμα [3.17a (β)].

Εξ άλλου αν χρησιμοποιείται ο κώδικας EBCDIC με δυαδικό ψηφίο ισοτιμίας, η ταινία θα έχει 9 κανάλια (8 + 1) κ.ο.κ.

Στη μαγνητική ταινία χρησιμοποιείται ο όρος πυκνότητα εγγραφής (Density) για να καταδειχθεί η ποσότητα των πληροφοριών, που έχει εγγραφεί στη μονάδα μήκους. Μετράται με το bpi (Bytes per inch) ή με τους χαρακτήρες ανά ίντσα (ch/inch). Η πυκνότητα φθάνει μέχρι τους 1800 (3000 στους νεώτερους τύπους) χαρακτήρες ανά ίντσα, πράγμα που σημαίνει ότι σε μια ταινία μήκους 2400 ποδών μπο-

ρούν να περιληφθούν περισσότεροι από 40 εκατομμύρια χαρακτήρες που θα χρειάζονταν 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 (Inter-block gaps) [σχ. 3.17β(β)].



Σχ. 3.17β.

(a) Κενά ανάμεσα στις λογικές ενότητες (IRG). (b) Κενά ανάμεσα στις φυσικές ενότητες (IRG).

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

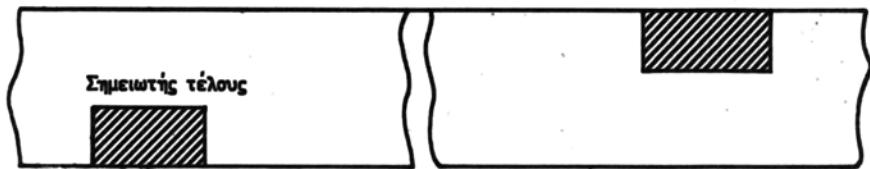
Η εγγραφή ή η ανάγνωση των πληροφοριών γίνεται καθώς η ταινία περνά μπροστά από τις κεφαλές εγγραφής-αναγνώσεως της μονάδας μαγνητικής ταινίας, ακριβώς όπως συμβαίνει σε ένα κοινό μαγνητόφωνο.

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 (inches per second), ο δε χρόνος λήψεως μιας 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 ετικέτες και ονομάζονται σημειωτές αρχής (Load point marker) και τέλους (End - of tape marker) αντίστοιχα (σχ. 3.17γ).

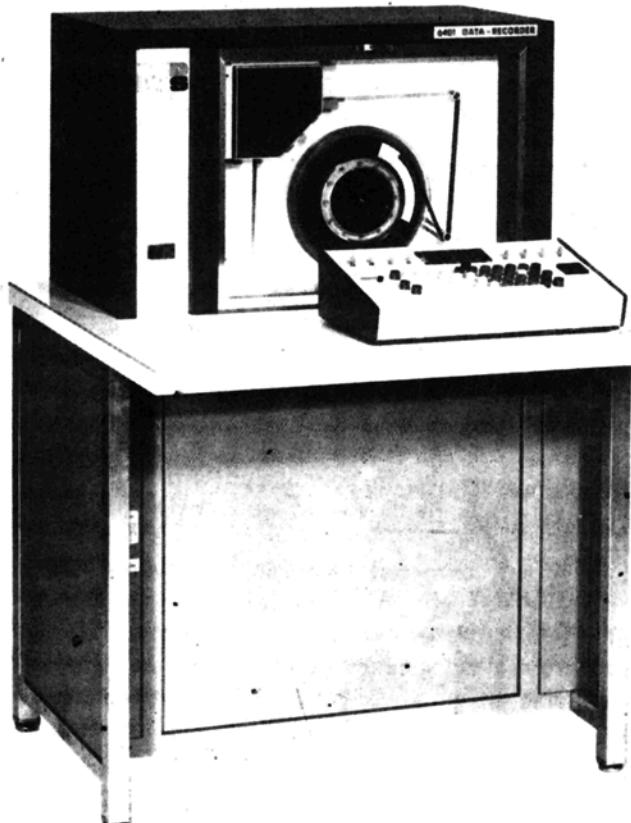
Σημειωτής αρχής



Σχ. 3.17γ.
Οι σημειωτές αρχής και τέλους της ταινίας.

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

Παληότερα για να εγγραφούν δεδομένα σε μια ταινία, έπρεπε να διατρυπηθούν πρώτα σε δελτία ή άλλο μη μαγνητικό φορέα και στη συνέχεια να μεταφερθούν στην ταινία με τη βοήθεια του ίδιου του υπολογιστή.



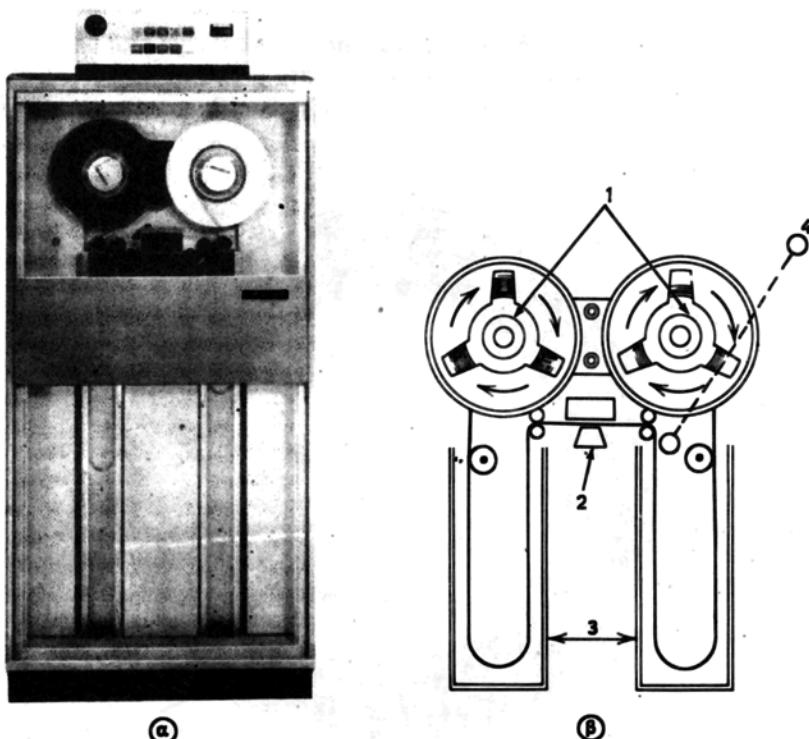
Σχ. 3.18.
Ο κωδικοποιητής μαγνητικής ταινίας.

Σήμερα, εκτός από τον κλασσικό αυτό τρόπο, χρησιμοποιούνται ειδικές μονάδες, που δεν συνδέονται με το κεντρικό συγκρότημα. Στις μονάδες αυτές γίνεται η εγγραφή στην ταινία των πληροφοριών απ' ευθείας από τα παραστατικά. Η μονάδα διαθέτει για το σκοπό αυτό πληκτρολόγιο όμοιο με της κοινής γραφομηχανής (σχ. 3.18). Στην ίδια μονάδα γίνεται και η επαλήθευση για την ορθότητα εγγραφής. Η ταινία στη συνέχεια είναι έτοιμη για να τόποθετηθεί στην περιφερειακή μονάδα μαγνητικής ταινίας και να χρησιμοποιηθεί από τον υπολογιστή.

3.19 Περιφερειακή μονάδα μαγνητικής ταινίας (Magnetic tape unit).

Στη μονάδα αυτή τοποθετείται η μαγνητική ταινία [σχ. 3.19 (α)]. Αποτελείται από τα εξής βασικά τμήματα [σχ. 3.19(β)]:

- Δύο υποδοχές 1, όπου τοποθετούνται η γεμάτη και η κενή μπομπίνες.
- Τρία συστήματα μαγνητικών κεφαλών 2 για τη διαγραφή, την ανάγνωση και την εγγραφή πληροφοριών αντίστοιχα.
- Δύο θαλάμους κενού 3 για την ομαλή κίνηση της ταινίας, που την προφυλάσσουν από του να κοπεί στις πολύ απότομες κινήσεις της.
- Δύο φωτοκύπταρα 4 που ανιχνεύουν τους σημειωτές τέλους και αρχής.
- Μια σειρά από πλήκτρα με τα οποία ελέγχομε εξωτερικά την κίνηση της ταινίας.



Σχ. 3.19.

(α) Περιφερειακή μονάδα μαγνητικής ταινίας.(β) Το εσωτερικό της μονάδας.

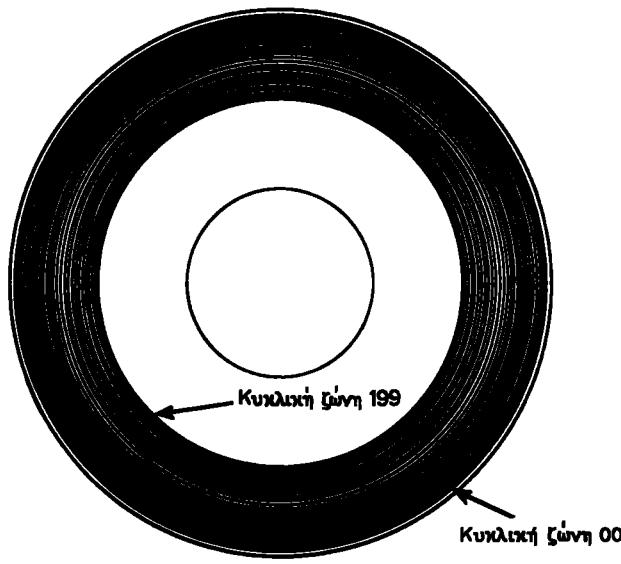
3.20 Μαγνητικός Δίσκος (Magnetic disk).

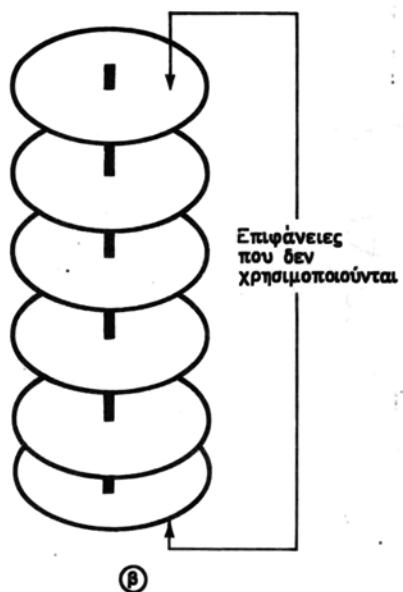
Ο μαγνητικός δίσκος όπως και η μαγνητική ταινία, είναι μέσο μαζικής αποθηκεύσεως πληροφοριών (Mass storage medium): γι' αυτό αναφέρεται πολλές φορές και ως **εξωτερική μνήμη** (External memory).

Μοιάζει με ένα κοινό δίσκο γραμμοφώνου. Έχει σχήμα κυκλικό με διάμετρο 30 - 60 cm και κατασκευάζεται από μέταλλο. Οι δύο επιφάνειές του είναι επιστρωμένες με μαγνητικό υλικό [σχ. 3.20a (a)]. Πολλοί μαζί δίσκοι στερεωμένοι επάνω σε ένα κοινό κατακόρυφο άξονα, που περνά από τα κέντρα τους, σχηματίζουν μία δέσμη δίσκων (Disk pack). Συνήθως οι δέσμες αυτές [σχ. 3.20a(β,γ)] αποτελούνται από 6 δίσκους. Στην περίπτωση της δέσμης η 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 το 0.

Η χωρητικότητα μιας δέσμης δίσκων κυμαίνεται μεταξύ 3 - 50 εκατομμύρια χαρακτήρες. Η δέσμη των δίσκων τοποθετείται στη μονάδα μαγνητικού δίσκου (Magnetic disk unit) και μπορεί να αφαιρείται 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. 3.20a (a)]. Συνήθως κάθε επιφάνεια δίσκου





Σχ. 3.20α.

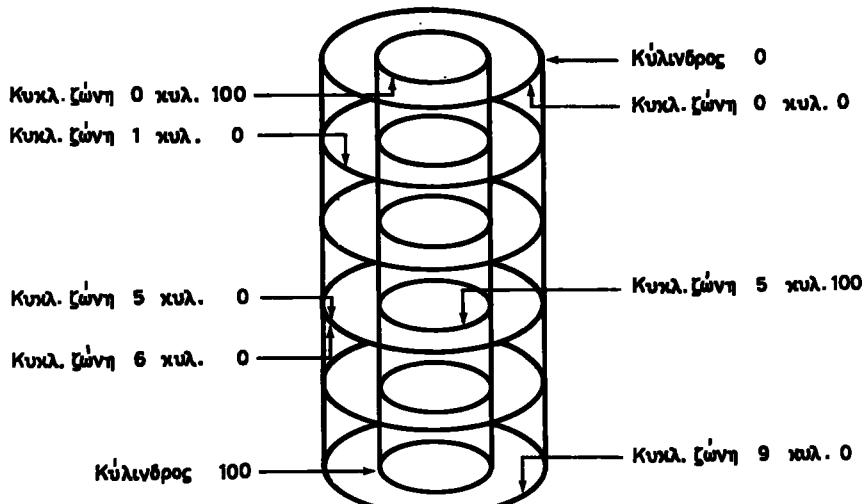
(α) Ο μαγνητικός δίσκος (β, γ) Δέσμες μαγνητικών δίσκων (Disk Packs).

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 το 0. Οι κυκλικές ζώνες όλων των επιφανειών μιας δέσμης δίσκων, που έχουν τον ίδιο αύξοντα αριθμό, σχηματίζουν μια νοητή κυλινδρική επιφάνεια που ονομάζεται κύλινδρος (cylinder). Επομένως υπάρχουν τόσοι κύλινδροι, όσες και οι κυκλικές ζώνες σε μια 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.

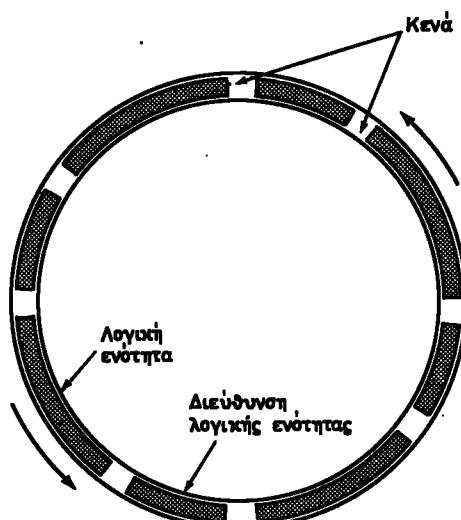
Σε κάθε κυκλική ζώνη μπορούν να εγγραφούν κωδικοποιημένοι 3600 - 7300 χαρακτήρες, η πυκνότητα δημιουργίας εγγραφής μεταβάλλεται από τύπο σε τύπο.

Η θέση όπου βρίσκεται μια λογική ενότητα (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 ενότητας (σχ. 3.20γ).

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



Σχ. 3.20θ.
Η έννοια του κυλίνδρου.



Σχ. 3.20γ.
Η έννοια της διεύθυνσεως.

διαβάσει. Για το λόγο αυτό ο μαγνητικός δίσκος χαρακτηρίζεται ως μέσο *τυχαίας* ή *δμεστης προσπελάσεως* (*random or direct access*).

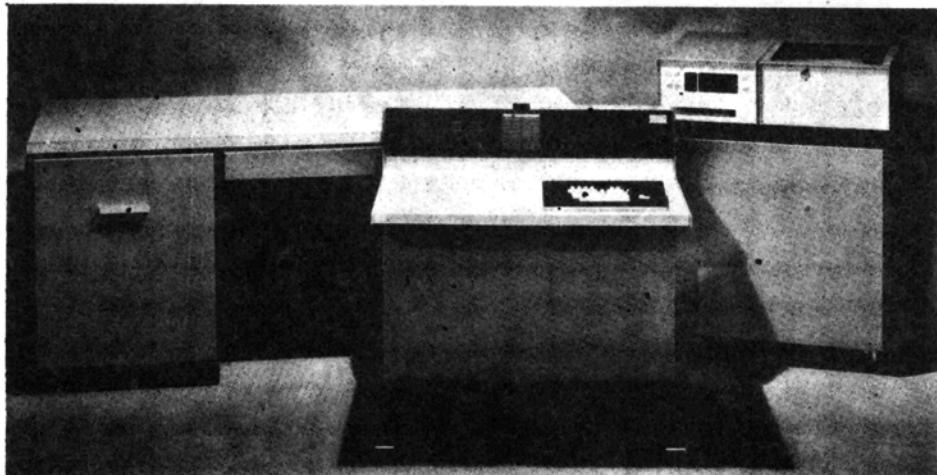
Ο μαγνητικός δίσκος κατά τη διάρκεια της επεξεργασίας μπορεί να περιστρέφεται με ταχύτητα μέχρι 3000 στροφών το λεπτό.

Πλεονεκτήματα του δίσκου είναι η πολύ μεγάλη χωρητικότητά του και η υψηλή

ταχύτητα επεξεργασίας, ενώ μειονεκτήματα είναι το υψηλό κόστος του, οι ιδιαίτερες συνθήκες, που χρειάζεται για την αποθήκευσή του και οι πολύπλοκες τεχνικές οργανώσεως πληροφοριών που απαιτεί.

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

Όπως έχουμε πει και στην περίπτωση της μαγνητικής ταινίας, έτσι και στο μαγνητικό δίσκο παλιότερα δεν είχαμε τη δυνατότητα να εγγράψουμε τις πληροφορίες από τα παραστατικά απ' ευθείας επάνω του, αλλά έπρεπε να προηγηθεί η εγγραφή τους επάνω σε κάποιο άλλο φορέα, π.χ. δελτία.



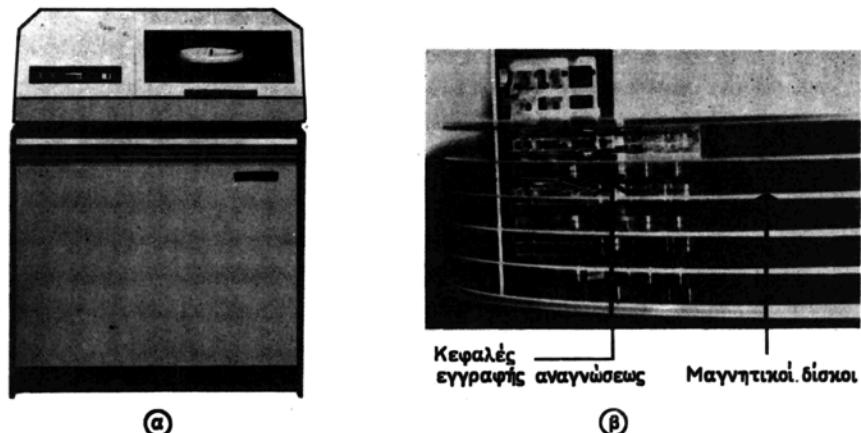
Σχ. 3.21.
Ο κωδικοποιητής μαγνητικού δίσκου.

Σήμερα υπάρχουν ειδικές μηχανές, οι **κωδικοποιητές μαγνητικού δίσκου** (σχ. 3.21) με τις οποίες γίνεται απ' ευθείας η εγγραφή στο δίσκο, με τη βοήθεια πληκτρολογίου. Οι μηχανές αυτές εκτός από το πληκτρολόγιο διαθέτουν συχνά και οθόνη, επάνω στην οποία προβάλλονται τα πληκτρολογούμενα δεδομένα. Μπορούν ακόμα να ελέγχουν την ορθότητα εγγραφής όπου χρειάζεται.

3.22 Περιφερειακή μονάδα μαγνητικού δίσκου (Magnetic disk unit).

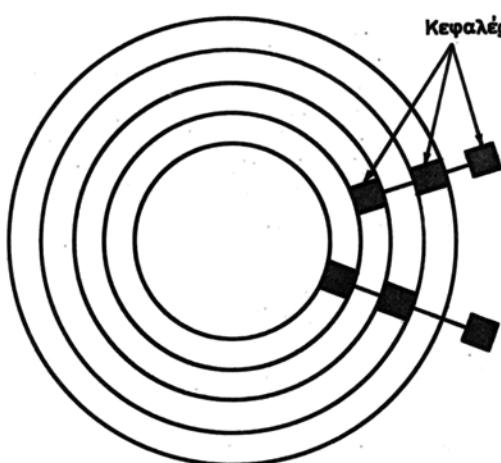
Στη μονάδα αυτή τοποθετούνται οι μαγνητικοί δίσκοι (σχ. 3.22a). Ορισμένοι τύποι δέχονται μόνο απλούς δίσκους, άλλοι διπλούς και άλλοι δέσμες μέχρι και 10 δίσκων. Διαθέτει ειδική υποδοχή, όπου τοποθετούνται οι δίσκοι καθώς και ένα σύστημα από βραχίονες (access arms). Στον καθένα απ' αυτούς προσαρμόζονται οι κεφαλές εγγραφής-αναγνώσεως (Read-write heads). Ο αριθμός των βραχιόνων και των κεφαλών ποικίλλει από τύπο σε τύπο.

Στους πιο εξελιγμένους τύπους οι βραχίονες διαθέτουν πολλές κεφαλές, ώστε η



Σχ. 3.22α.

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



Σχ. 3.22β.

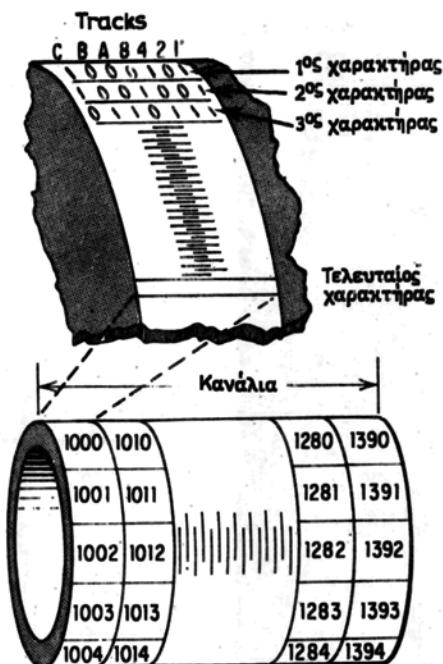
Μονάδα μαγνητικού δίσκου με μια κεφαλή σε κάθε κυκλική ζώνη.

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

Ο χρόνος, που απαιτείται για να φθάσει η κεφαλή στο σημείο του δίσκου, όπου έχει γραφεί μια πληροφορία, ονομάζεται χρόνος προσπελάσεως (access time) και στα νεώτερα συστήματα κυμαίνεται από 25 έως 40 milliseconds.

3.23 Μαγνητικό τύμπανο (Magnetic drum).

Το μαγνητικό τύμπανο έχει τη μορφή κοίλου κυλίνδρου. Είναι μεταλλικό, με διάμετρο 10 - 30 cm, η δε εξωτερική του επιφάνεια έχει επιστρωθεί με κάποιο μα-



Σχ. 3.23.
Το μαγνητικό τύμπανο.

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

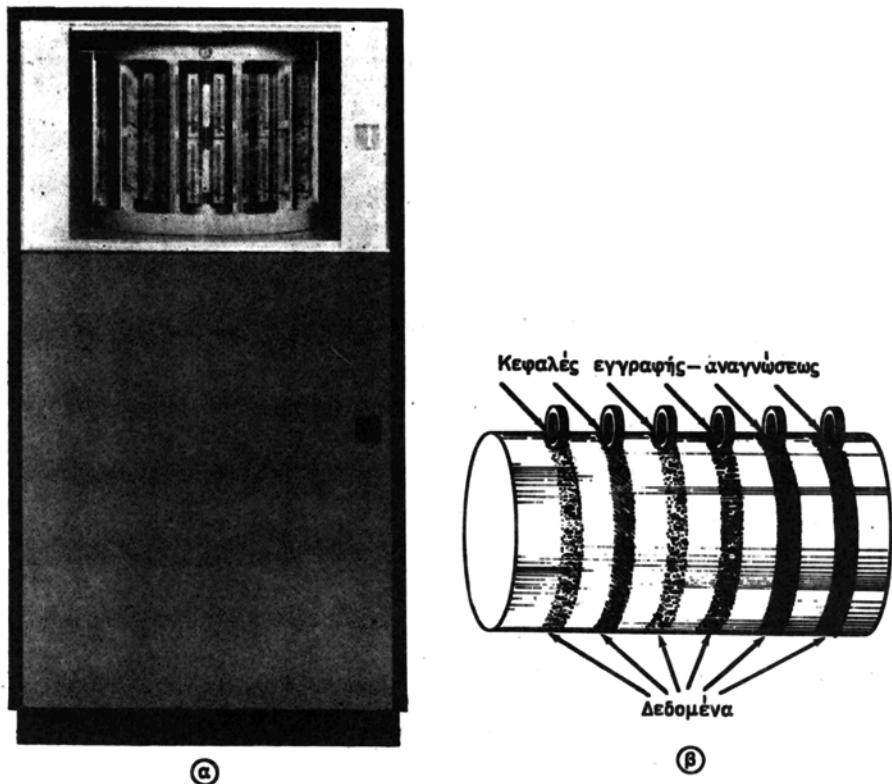
Η μαγνητική του επιφάνεια διαιρείται σε δακτυλοειδείς ζώνες (κάθετα προς το διαμήκη άξονά του) που ονομάζονται *tracks* και σε επιμήκεις λωρίδες (παράλληλες προς τον άξονά του) που ονομάζονται *κανάλια* (*channels*) (σχ. 3.23).

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

3.24 Περιφερειακή μονάδα μαγνητικού τυμπάνου (Magnetic drum unit).

Στη μονάδα αυτή τοποθετείται το μαγνητικό τύμπανο [σχ. 3.24(a)]. Αποτελείται βασικά:

- Από ένα άξονα, στον οποίο προσαρμόζεται το τύμπανο και που μπορεί να το περιστρέψει με σταθερή ταχύτητα 3000 - 12500 στροφές το λεπτό.
- Από μία σειρά κεφαλών αναγνώσεως-εγγραφής, που είναι τοποθετημένες



Σχ. 3.24.

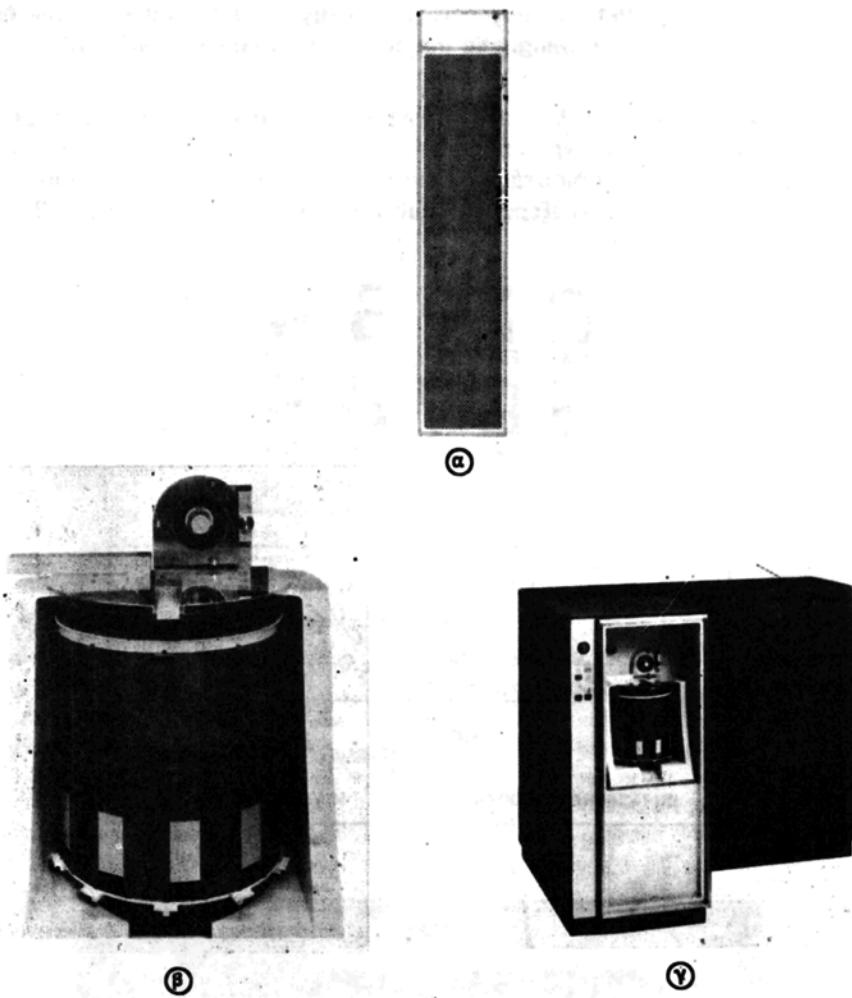
(a) Μονάδα μαγνητικού τυμπάνου (β) Το σύστημα κεφαλών αναγνώσεως - εγγραφής.

κατά μήκος μιας γεννέτειρας του τυμπάνου και σε μικρή απόσταση από την επιφάνειά του. Υπάρχουν τόσες κεφαλές, όσες και οι ζώνες του τυμπάνου [σχ. 3.24(β)].

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

Ενώ η μαγνητική ταινία, ο δίσκος και το τύμπανο είναι φορείς, που χαρακτηρίζονται από μία συνεχή και μεγάλη μαγνητική επιφάνεια, η μαγνητική κάρτα αντίθετα είναι μία ορθογώνια πλαστική κάρτα μικρών διαστάσεων ($1 - 3'' \times 3,5 - 14''$). Η κάρτα αυτή επικαλύπτεται από τη μία της επιφάνεια με ένα λεπτό στρώμα μαγνητικού υλικού. Η επιφάνεια της κάρτας χωρίζεται καθ' όλο το μήκος της σε παράλληλες ζώνες (tracks), όπως και στην ταινία [σχ. 3.25(a)]. Πολλές μαζί κάρτες είναι τοποθετημένες σε ειδική θήκη, που ονομάζεται κυψέλη (cell). Η κυψέλη [σχ. 3.25(β)] τοποθετείται σε μια υποδοχή της μονάδας μαγνητικών καρτών [σχ. 3.25(γ)].

Για να διαβασθούν τα δεδομένα, που βρίσκονται επάνω σε μία κάρτα, αναζητείται και επιλέγεται η κάρτα με κάποιο μηχανισμό και στη συνέχεια οδηγείται σε ένα



Σχ. 3.25.

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

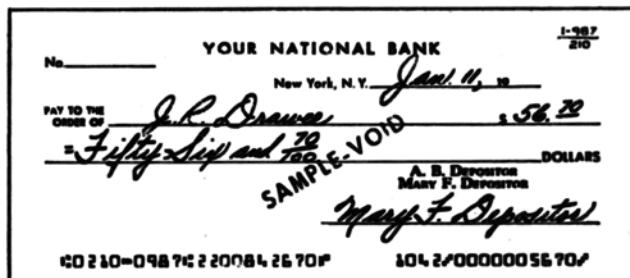
γρήγορα περιστρεφόμενο τύμπανο, όπου και περιτυλίγεται. Επάνω από το τύμπανο βρίσκεται το σύστημα των κεφαλών αναγνώσεως-εγγραφής. Όταν τελειώσει η ανάγνωση, η κάρτα απελευθερώνεται και επιστρέφει στη θέση της. Η διαδικασία αυτή μοιάζει με τη διαδικασία επιλογής ενός δίσκου σε ένα «τζουκ - μποξ». Για την εγγραφή δεδομένων σε μια κάρτα ακολουθείται ο ίδιος τρόπος. Οι μαγνητικές κάρτες έχουν βραδύ σχετικά ρυθμό προσπελάσεως (100 - 600 m/s), έχουν όμως ως πλεονεκτήματα το ότι είναι φθηνές, μπορούν εύκολα να αντικατασταθούν και να αποταμιεύσουν μεγάλο όγκο πληροφοριών. Η χωρητικότητα μιας μονάδας μαγνητικών καρτών ξεπερνά τα 40 εκατομμύρια χαρακτήρες.

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

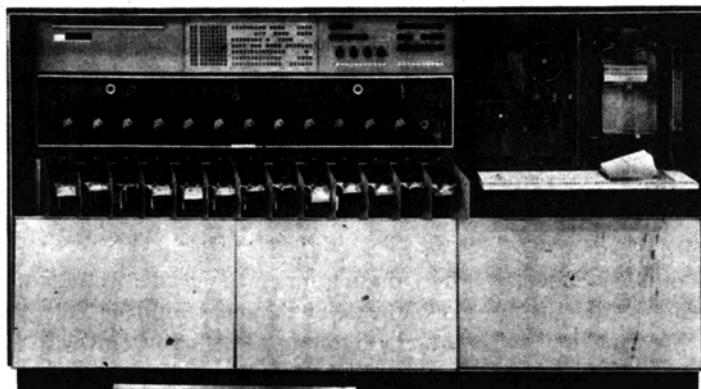
Μαγνητική μελάνη (δηλαδή κοινή τυπογραφική μελάνη έμπλουτισμένη με ψίγιματα από μαγνητικό υλικό) χρησιμοποιείται για το τύπωμα χαρακτήρων [σχ. 3.26(a)] επάνω σε διαφόρων διαστάσεων έντυπα. Η μέθοδος αυτή χρησιμοποιείται κυρίως στην έκδοση και επεξεργασία των τραπεζικών επιταγών [σχ. 3.26(b)].



(a)



(b)



(γ)

Σχ. 3.26.

- (a) Ένας τύπος χαρακτήρων μαγνητικής μελάνης.(β) Επιταγή με χαρακτήρες μαγνητικής μελάνης.(γ) Αναγνώστης χαρακτήρων μαγνητικής μελάνης.

Ο αναγνώστης χαρακτήρων μαγνητικής μελάνης [σχ. 3.26(γ)] μπορεί να διαβάζει 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 τηλεοράσεως 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.

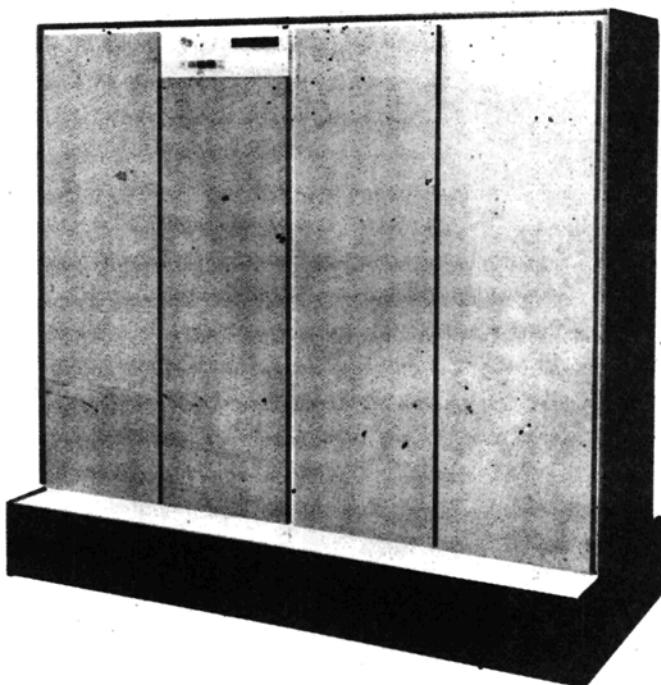
Υπάρχουν δύο τύποι:

- 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 η 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 ακουστικής αποκρίσεως (Audio response unit).

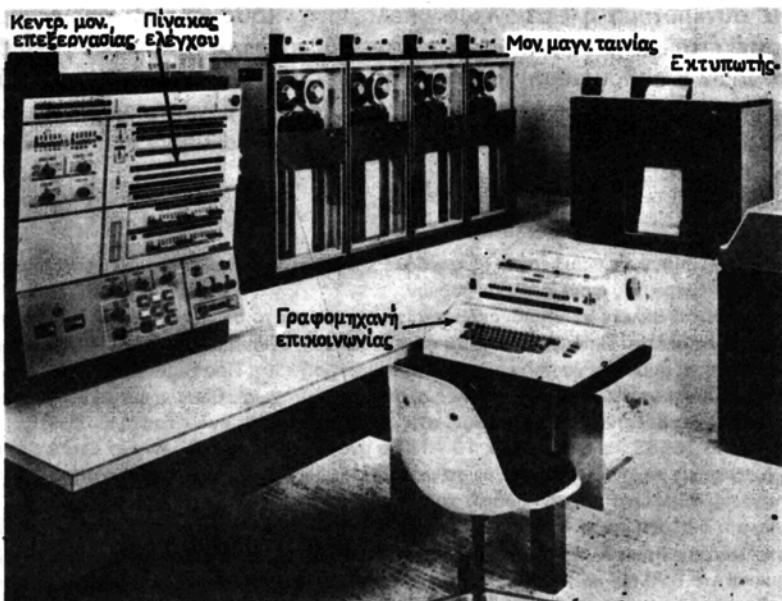
Μία μονάδα ιδιαίτερα χρήσιμη στα συστήματα πραγματικού χρόνου (Real time systems), για τα οποία θα μιλήσουμε πιο αναλυτικά στην παράγραφο 16.3.2, είναι και η μονάδα ακουστικής αποκρίσεως (σχ. 3.28). Η μονάδα αυτή δίνει πληροφορίες από τον υπολογιστή με τη μορφή φωνητικών σημάτων-λέξεων. Ένα πλήθος από φωνητικές συλλαβές βρίσκεται αποθηκευμένο σε μια βοηθητική μνήμη π.χ. ένα μαγνητικό τύμπανο. Το ηχητικό μήνυμα της απαντήσεως σε κάποια ερώτηση, συναρμολογείται με τη βοήθεια ειδικού προγράμματος από τις αποθηκευμένες συλλαβές ή λέξεις και μεταδίδεται από μεγάφωνο ή ακουστικό τηλεφώνου.



Σχ. 3.28.
Μονάδα ηχητικής αποκρίσεως.

3.29 Πίνακας ελέγχου (Console).

Είναι μια μονάδα για τον εξωτερικό έλεγχο της καταστάσεως του κεντρικού συγκροτήματος. Περιέχει διακόπτες, ενδεικτικές λυχνίες και πλήκτρα (σχ. 3.29), με τη βοήθεια των οποίων ο χειριστής μπορεί να θέσει σε λειτουργία το συγκρότημα ή να το σταματήσει και ακόμα μπορεί να ελέγχει κάθε στιγμή, αν η λειτουργία είναι ομαλή. Η μονάδα αυτή πλαισιώνεται από τη γραφομηχανή επικοινωνίας (I/O Type-writer).



Σχ. 3.29.

Ένα συγκρότημα ψηφιακού ηλεκτρονικού υπολογιστή. Αριστερά διακρίνεται η κεντρική μονάδα επεξεργασίας, ο πίνακας ελέγχου (console) και στο βάθος 4 μονάδες μαγνητικής ταινίας επάνω δεξιά διακρίνεται ένας εκτυπωτής (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 μια μικρή μονάδα μνήμης.

Έστι ένας απλός τυπικός τερματικός σταθμός μπορεί να αποτελείται μόνο από ένα τηλέτυπο ή μια μονάδα οθόνης με πληκτρολόγιο ή να περιέχει άλλες μονάδες, όπως μονάδα αναγνώσεως δελτίων και εκτυπωτή ή μονάδα αναγνώσεως χαρτοταινίας και εκτυπωτή κ.ά. μαζί και με μια μονάδα μνήμης.

3.31 Μερικές παρατηρήσεις.

Όλες οι περιφερειακές μονάδες, που περιγράψαμε πιο πάνω δεν συναντώνται

σε 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 τον κάθε τύπο, όπως π.χ. 4 μονάδες μαγνητικής ταινίας, 2 μονάδες μαγνητικού δίσκου κλπ. Βέβαια υπάρχουν πάντα δυνατότητες επεκτάσεως (Expandibility), που δεν είναι όμως απεριόριστες.

Ερωτήσεις - ασκήσεις.

1. Σε τι χρησιμεύουν οι περιφερειακές μονάδες και σε τι η κεντρική μονάδα επεξεργασίας;
 2. Αναφέρετε μερικούς φορείς πληροφοριών.
 3. Υποθέστε ότι θέλετε να δημιουργήσετε ένα αρχείο για τα υλικά μιας αποθήκης επάνω σε ογδοντάστηλα δελτία. Τι πληροφορίες θα έπρεπε καπά τη γνώμη σας να συγκεντρώσετε για κάθε υλικό, ώστε να μπορείτε να παρακολουθείτε την κίνησή τους; Προσπαθείστε κατόπιν να σχεδιάστε κατάλληλα επάνω στο δελτίο τα πεδία, όπου θα καταχωρισθούν οι διάφορες πληροφορίες.
 4. Υποθέστε ότι θέλετε να διατρυπήσετε στις στήλες 1 - 5 ενός ογδονταστήλου δελτίου το δεκαδικό αριθμό 15683 και στις στήλες 10 - 19 το όνομα NICOLAIDIS. Αν δεν έχετε στη διάθεσή σας διατρητική μηχανή, μαυρίστε με το μολύβι σας τις θέσεις του δελτίου, όπου θα γίνουν οι διατρήσεις σύμφωνα με τον κώδικα του Hollerith.
 5. Τι καλείται διαλογή και τι συνένωση δύο ομάδων δελτίων; Ποια σχέση υπάρχει μεταξύ τους;
 6. Δίνεται μια μαγνητική ταινία με αφέλιμο μήκος 2400 ft. Αν εμείς καταχωρίζομε τους χαρακτήρες με πυκνότητα 800 bpi και 100 χαρακτήρες αποτελούν μιά λογική ενότητα (record), βρήτε πόσες λογικές ενότητες μπορούν να εγγραφούν στην ταινία, αν υποτεθεί ότι οι λογικές ενότητες δεν είναι αμαδοποιημένες.
 7. Η ίδια άσκηση με την προηγούμενη με τη διαφορά ότι τώρα οι λογικές ενότητες είναι ομαδοποιημένες σε ομάδες με 10 ενότητες ή κάθε μία.
 8. Δίνεται μια δέσμη από 10 δίσκους. Κάθε επιφάνεια υποδιαιρέταται σε 100 κυκλικές ζώνες και κάθε ζώνη σε 100 τομείς. Μέσα σε κάθε τομέα καταχωρίζονται 80 χαρακτήρες. Ποια είναι η χωρητικότητα της δέσμης σε χαρακτήρες;
 9. Αν σας ζητούσαν να δημιουργήσετε ένα αρχείο επάνω σε κάποιο φορέα, τι θα προτιμαστείτε, τα δελτία, τη μαγνητική ταινία, το μαγνητικό δίσκο, το μαγνητικό τύμπανο ή τη χαρτοταινία και γιατί;
 10. Αν σας δίνουν ένα αρχείο σε μαγνητική ταινία, της μισθοδοσίας των υπαλλήλων μιας επιχειρήσεως, και ένα αρχείο σε διάτρητα δελτία με τις μεταβολές τους, και σας ζητούν να κάνετε επεξεργασία και να τυπώσετε μια κατάσταση, πόσες και ποιες περιφερειακές μονάδες θα χρειασθείτε;
-

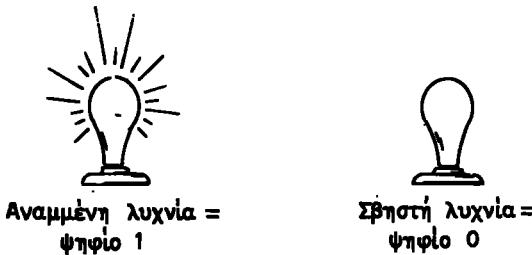
ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΨΗΦΙΑΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ II Η ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ

4.1 Η κεντρική μνήμη — Γενικά.

Όπως είπαμε στην αρχή του πρόγραμμανού κεφαλαίου, στην κεντρική μνήμη αποθηκεύονται οι διάφορες πληροφορίες είτε ως εντολές, είτε ως απλά δεδομένα σε κωδικοποιημένη μορφή με τη βοήθεια ενός από τα αριθμητικά συστήματα, που περιγράψαμε.

Η κωδικοποίηση των πληροφοριών με δυαδική μορφή σε κάποιο σύστημα παρουσιάζει μεγαλύτερη απλότητα από την κωδικοποίηση με οποιοδήποτε άλλο τρόπο. Αυτό οφείλεται στο γεγονός ότι τό δυαδικό έχει μόνο δύο ψηφία, το 0 και το 1, που μπορούν εύκολα να παρασταθούν από οποιοδήποτε φυσικό σύστημα που παρουσιάζει δύο διαφορετικές καταστάσεις. Μπορούμε π.χ να παραστήσουμε το ψηφίο 0 με μία σβηστή λυχνία και το 1 με μία αναμμένη (σχ. 4.1α).



Σχ. 4.1α.
Παράσταση των ψηφίων 0 και 1 με μία λυχνία.

Αν λοιπόν έχομε δύο λυχνίες, τη μία δίπλα στην άλλη, τότε ανάλογα με την κατάστασή τους μπορούμε να παραστήσουμε 4 διαφορετικούς διψήφιους δυαδικούς αριθμούς (σχ. 4.1β).

Αν διαθέτομε τρεις λυχνίες στη σειρά, μπορούμε να παραστήσουμε 8 διαφορετικούς τριψήφιους δυαδικούς αριθμούς (σχ. 4.1γ).

Και γενικά αν διαθέτομε n λυχνίες μπορούμε να παραστήσουμε 2ⁿ διαφορετικούς n-ψήφιους δυαδικούς αριθμούς.

Η λυχνία δύμας δεν είναι για πολλούς λόγους πρακτικό μέσο αποθηκεύσεως πληροφοριών, γι' αυτό χρησιμοποιούμε άλλες τεχνικές, που θα περιγράψουμε στη συνέχεια.

	Δυαδικός αριθμός	Αντίστοιχος δεκαδικός
	0 0	0
	0 1	1
	1 0	2
	1 1	3

Σχ. 4.1β.
Παράσταση διψήφιων δυαδικών αριθμών με δύο λυχνίες.

	Δυαδικός αριθμός	Δεκαδικός αριθμός
	0 0 0	0
	0 0 1	1
	0 1 0	2
	0 1 1	3
	1 0 0	4
	1 0 1	5
	1 1 0	6
	1 1 1	7

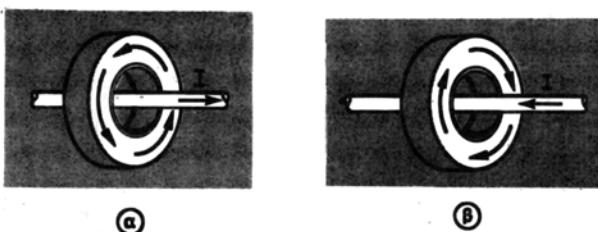
Σχ. 4.1γ.
Παράσταση τριψήφιων δυαδικών αριθμών με τρεις λυχνίες.

4.2 Μαγνητικός πυρήνες (Magnetic Cores).

Είναι η πιο παλιά και η πιο διαδεδομένη μορφή μνημών και βασίζεται στην ιδιότητα, που έχει το ηλεκτρικό ρεύμα, να μαγνητίζει ορισμένα υλικά.

Ας πάρομε ένα μικρό δακτύλιο φτιαγμένο από κάποιο μαγνητιζόμενο υλικό (φερρίτη) και ας περάσουμε κάθετα μέσα από αυτό ένα λεπτό αγώγιμο σύρμα. Ο δακτύλιος είναι στην αρχή αμαγνήτιστος. Αν μέσα από το σύρμα διοχετεύσουμε ρεύμα κατάλληλης εντάσεως, ο δακτύλιος θα μαγνητισθεί έτσι, ώστε η φορά των δυναμικών γραμμών του πεδίου του να καθορίζεται, όπως γνωρίζομε από την ηλεκτροτεχνία, από το γνωστό κανόνα του δεξιού χεριού.¹

Άν διακόψωμε το ρεύμα στο δακτύλιο θα διαπρήσει τη μαγνητική του κατάσταση [σχ. 4.2a(α)]. Αν αντιστρέψουμε τη φορά του ρεύματος, τότε αντιστρέφεται και η φορά των δυναμικών γραμμών του μαγνητικού πεδίου [σχ. 4.2a(β)]. Για ευκολία μας θα ονομάζομε τις δύο διαφορετικές καταστάσεις μαγνητίσεως θετική και αρνητική.



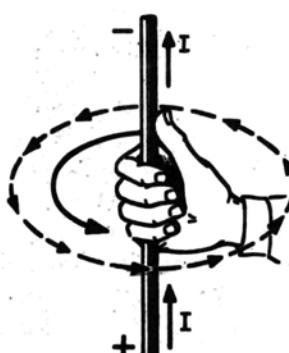
Σχ. 4.2a.

a) Η κατάσταση του δακτυλίου είναι 1. β) Η κατάσταση του δακτυλίου είναι 0.

Μπορούμε λοιπόν να αντιστοιχίσουμε στην κατάσταση θετικής μαγνητίσεως του δακτυλίου το δυαδικό ψηφίο 1 και στην κατάσταση αρνητικής μαγνητίσεως, το δυαδικό ψηφίο 0 (σχ. 4.2a).

Τους δακτυλίους αυτούς, που έχουν διάμετρο περίπου $1/16''$, τους ονομάζομε πυρήνες ή και φερρίτες, από το υλικό που είναι κατασκευασμένοι. Πολλοί πυρήνες

¹ Σύμφωνα με τον κανόνα αυτό αν με το δεξί χέρι κρατήσουμε ένα αγωγό έτσι, ώστε ο ανοικτός αντίχειρας να δείχνει τη φορά του ρεύματος I, τότε τα σφιγμένα δάκτυλα θα δείχνουν τη φορά των μαγνητικών γραμμών.

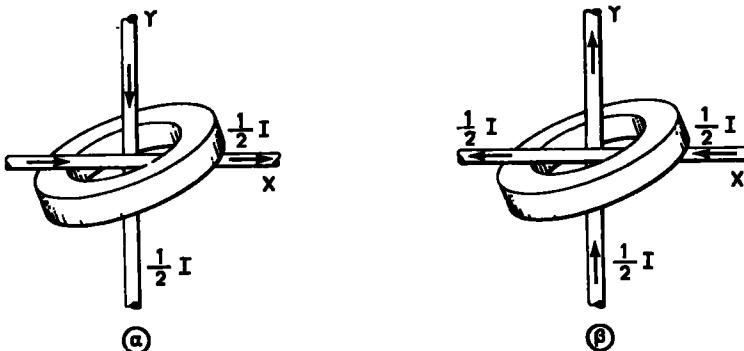


περασμένοι σε ένα αγωγό, όπως οι χάνδρες ενός κομπολογιού, θα συμβολίζουν, όπως εξυπακούεται, μια σειρά από δυαδικά ψηφία (bits)..

Ας υποθέσουμε ότι η ένταση του ρεύματος, που χρειάζεται για να μαγνητισθεί ένας πυρήνας, είναι I και ότι να περνά μέσα από τον πυρήνα ένας μόνο αγωγός, περνούν δύο, ο X και ο Y κάθετοι μεταξύ τους. Ο καθένας από αυτούς σχηματίζει γωνία 45° με το δακτύλιο και μπορεί να διαρρέεται από ρεύμα εντάσεως $1/2 I$.

Αν από τον αγωγό X περάσει ρεύμα $1/2 I$ με φορά προς τα δεξιά και από τον αγωγό Y περάσει ρεύμα $1/2 I$ με φορά προς τα κάτω [(σχ. 4.2β(α))], ο αρχικά αμαγνήτιστος πυρήνας θα μαγνητισθεί θετικά, πράγμα που σημαίνει κατάσταση 1 (ή τιμή του bit = 1). Αν όμως η φορά των ρευμάτων στους δύο αγωγούς είναι αντίθετη, η μαγνήτιση του πυρήνα θα είναι και αυτή αντίθετη, πράγμα που αντιστοιχεί σε κατάσταση 0 (ή τιμή του bit = 0) [σχ. 4.2β(β)].

Αν περάσει μόνο από τον ένα αγωγό ρεύμα $1/2 I$, ο πυρήνας δεν θα μαγνητισθεί. Το ίδιο θα συμβεί και όταν το ρεύμα που περνά από τον X έχει φορά προς τα δεξιά, ενώ το ρεύμα που περνά από τον Y έχει φορά προς τα επάνω, και αντίστροφα.



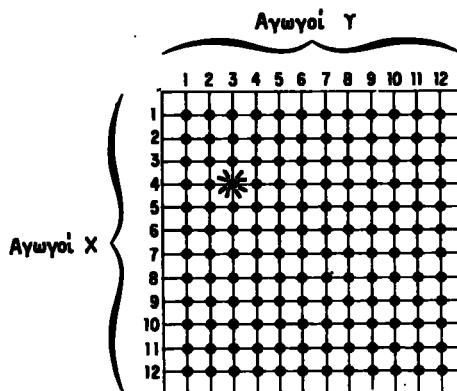
Σχ. 4.2β.

Οι φορές των ρευμάτων των αγωγών X και Y για θετική (κατάσταση 1) και αρνητική (κατάσταση 0) μαγνήτιση του δακτυλίου.

Σημειώνομε ότι όταν μιλάμε για ρεύμα, στην πραγματικότητα εννοούμε ηλεκτρικούς παλμούς πολύ μικρής χρονικής διάρκειας.

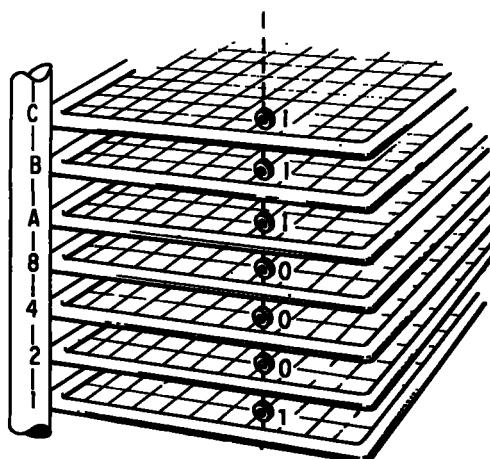
Αν τοποθετήσουμε πολλούς πυρήνες σε μια δυδιάστατη διάταξη (σχ. 4.2γ) θα σχηματίσουμε ένα επίπεδο (Magnetic - core plane), όπου κάθε πυρήνας θα βρίσκεται στη διασταύρωση ενός αγωγού X και ενός Y . Όταν από ένα αγωγό X , π.χ. τον X_4 , περάσει ρεύμα εντάσεως $1/2 I$ με φορά προς τα δεξιά και από τον αγωγό Y_3 περάσει επίσης ρεύμα $1/2 I$ με φορά προς τα κάτω, τότε ο πυρήνας, που βρίσκεται στη διασταύρωση τους, θα βρεθεί σε κατάσταση 1. Κανένας άλλος πυρήνας του επιπέδου δεν θα μεταβάλει την κατάστασή του.

Πολλά επίπεδα πυρήνων τοποθετημένα το ένα επάνω από το άλλο δημιουργούν μια τριδιάστατη διάταξη. Οι πυρήνες που βρίσκονται επάνω στην ίδια στήλη, χρησιμοποιούνται για την αποθήκευση ενός κωδικοποιημένου χαρακτήρα. Έτσι αν



Σχ. 4.2y.
Ένα επίπεδο μαγνητικών πυρήνων.

Έχουμε π.χ. 7 επίπεδα πυρήνων το ένα επάνω από το άλλο, τότε τα 7 στοιχεία (bits), που βρίσκονται σε μια τυχαία στήλη της διατάξεως, θα μπορούν να παραστήσουν ένα χαρακτήρα κωδικοποιημένο στὸν επταψήφιο αλφαριθμητικό κώδικα (δηλαδή τὸν 6-ψηφίο αλφαριθμητικό BCD + τὸ ψηφίο τῆς ισοπιμίας). Ο χαρακτήρας Α π.χ στὸν κώδικα αυτὸν έχει τὴ μορφὴ 1110001. Διοχέτευοντας λοιπὸν ρεύματα με εντάσεις 1/2 Ι απὸ τοὺς κατάλληλους αγωγούς Χ καὶ Υ τῶν·7 επιπέδων, μπορούμε να σχηματίσομε κατά μήκος μιας στήλης τὸν κωδικοποιημένο χαρακτήρα Α (σχ. 4.2δ).



Σχ. 4.2δ.
Καταχώριση του κωδικοποιημένου χαρακτήρα Α στη μνήμη.

Φυσικά η μνήμη δεν αποτελείται από 7 μόνο επίπεδα μαγνητικών πυρήνων, αλλά από πολὺ περισσότερα έτσι, που επάνω στην ίδια στήλη να μπορούν να απεικο-

νίζονται περισσότεροι από ένας χαρακτήρες. Τα bits από τα οποία αποτελείται ένας κωδικοποιημένος χαρακτήρας, συνιστούν ένα byte¹. Πολλοί χαρακτήρες μαζί αποτελούν συχνά μια 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 το παρέχει η κατασκευάστρια εταιρία.

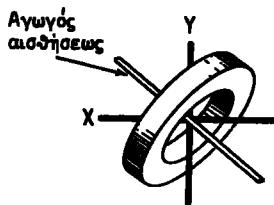
Το πιο χαρακτηριστικό μέγεθος της μνήμης είναι η χωρητικότητά της (capacity ή memory size), που εκφράζεται σε k - bytes ή k - words ($1\text{k} = 1024$). Οι χωρητικότητες κυμαίνονται συνήθως από 16 k έως 256 k bytes.

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, στέλνομε από τους αγωγούς X και Y, που τον διαρρέουν, ρεύματα εντάσεως $1/2 I$ και με φορές τέτοιες, σαν να θέλομε να γράψουμε σ' αυτόν την τιμή 0.

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



Σχ. 4.3.
Ο αγωγός αισθήσεως (Sense Wire).

Στον αγωγό αισθήσεως δημιουργούνται ρεύματα από επαγωγή κάθε φορά που μεταβάλλεται η μαγνητική κατάσταση ενός πυρήνα του επιπέδου. Άρα, αν ο πυρήνας, που διαβάζομε, βρισκόταν αρχικά στην κατάσταση 1, με τη διαδικασία της αναγνώσεως θα μεταβάλλει τη μαγνητική του κατάσταση σε 0, γεγονός που θα προκαλέσει τη δημιουργία επαγωγικού ρεύματος στον αγωγό αισθήσεως με φορά αντίθετη προς τη φορά της μεταβολής (κανόνας του Lenz) που με τη σειρά του θα επαναφέρει τον πυρήνα και πάλι στην κατάσταση 1.

Αν ο πυρήνας βρισκόταν αρχικά στην κατάσταση 0, τότε με τη διαδικασία της αναγνώσεως δεν θα υπάρξει καμιά μεταβολή, ο αγωγός αισθήσεως δεν θα διαρρέεται από ρεύμα και ο πυρήνας θα διατηρήσει την κατάστασή του, δηλαδή την τιμή 0.

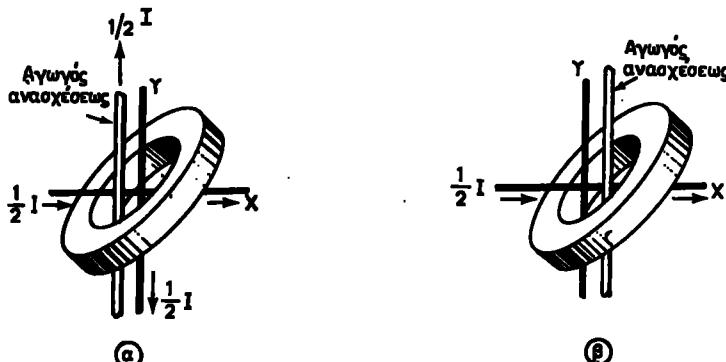
Με άλλα λόγια, όταν ο αγωγός αισθήσεως διαρρέεται από ρεύμα, σημαίνει ότι η τιμή του χαρακτήρα που διαβάζεται είναι 1, ενώ αν δεν διαρρέεται από ρεύμα, σημαίνει ότι η τιμή του είναι 0.

¹ Από εδώ και πέρα κάθε κωδικοποιημένο δυαδικό χαρακτήρα, θα τον αναφέρομε με τον αγγλικό όρο byte. Φυσικά το μέγεθος ενός byte εξαρτάται από τον κώδικα που χρησιμοποιούμε για τη δυαδική απεικόνιση των χαρακτήρων. Στο μέγεθος του byte δεν υπολογίζεται το ψηφίο της ιστιμάς (Parity bit).

4.4 Εγγραφή μιας τιμής σε ένα πυρήνα.

Η διαδικασία εγγραφής μιας τιμής σε κάποιο πυρήνα είναι πιο πολύπλοκη και περιλαμβάνει είτε την επαναποθήκευση της αρχικής τιμής, είτε την αποθήκευση μιας νέας τιμής διαφορετικής από την αρχική.

Στην περίπτωση της εγγραφής στέλνομε από τους αγωγούς X και Y ρεύματα εντάσεως $1/2 I$ με φορές τέτοιες, σαν να θέλαμε να δώσουμε στον πυρήνα αυτό τιμή 1. Έτσι αν ο πυρήνας είχε προηγούμενη τιμή 1 και θέλουμε να τη διαπρήσουμε με τις φορές που έχουν τα ρεύματα, η τιμή αυτή θα διαπηρηθεί. Αν είχε αρχικά τιμή 0 και θέλουμε να γίνει 1, τότε και πάλι οι φορές των ρευμάτων εξασφαλίζουν αυτή την αλλαγή. Υπάρχει όμως το ενδεχόμενο ο πυρήνας να είχε αρχικά τιμή 0 και να θέλουμε να τη διαπρήσουμε, ενώ με τη διαδικασία της εγγραφής η τιμή 0 θα καταστραφεί. Για να μη συμβεί αυτό εισάγουμε μέσα στον πυρήνα και ένας τέταρτος αγωγός παράλληλα προς τον Y, που ονομάζεται αγωγός ανασχέσεως (Inhibit wire) (σχ. 4.4). Ο αγωγός αυτός περνά μέσα από δύος τους πυρήνες ενός έπιπέδου.



Σχ. 4.4.

- a. Ο αγωγός ανασχέσεως διαρρέεται από ρεύμα. β. Ο αγωγός ανασχέσεως δεν διαρρέεται από ρεύμα.

Ενεργοποιείται μόνο στην περίπτωση που κάνουμε εγγραφή και θέλουμε να διαφυλάξουμε την τιμή 0 ενός πυρήνα, ώστε να μη γίνει 1 με τη διαδικασία της εγγραφής. Κάθε φορά που ο αγωγός ανασχέσεως ενεργοποιείται, διαρρέεται από ρεύμα εντάσεως $1/2 I$ με φορά αντίθετη προς τη φορά του Y, ώστε να εξουδετερώνει το ρεύμα του και έτσι να διατηρείται η τιμή 0.

4.5 Μνήμες από ημιαγωγούς.

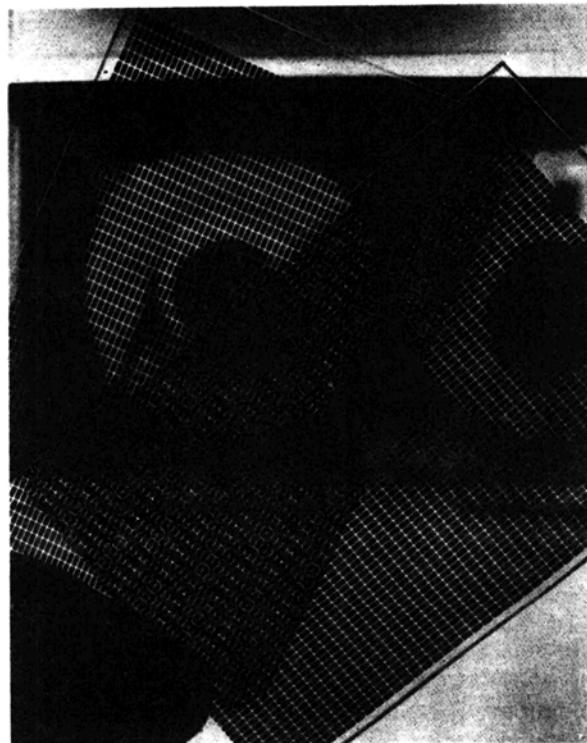
Μία μνήμη ημιαγωγών αποτελείται από πολλά ολοκληρωμένα μικροκυκλώματα (Chips), που συνδέονται μεταξύ τους με διακόπτες. Το κύριο συστατικό ενός τέτοιου μικροκυκλώματος είναι ημιαγωγό τύπου MOSFET.

Κάθε ολοκληρωμένο μικροκύκλωμα έχει χωρητικότητα 1024 bits.

Η μνήμη ημιαγωγού είναι πιο συμπαγής (Compact) και πιο γρήγορη από τη μνήμη μαγνητικών πυρήνων.

4.6 Άλλα είδη μνήμης.

Εκτός από τις μνήμες με μαγνητικούς πυρήνες χρησιμοποιούνται επίσης και μνήμες από λεπτά υμένια (Thin films memories) (σχ. 4.6), από λεπτά πεπλαστούσμένα σύρματα (Plated wire memories), φωτοψηφιακές μνήμες (Photodigital memo-



Σχ. 4.6.
Μνήμες από λεπτά υμένια.

ries), ηλεκτροοπτικές (Electrooptical memories) κλπ. που η λειτουργία τους στηρίζεται επάνω σε διάφορα φαινόμενα της φυσικής, όπως π.χ. η ηλεκτρική ή οπτική πόλωση Κ.Ο.Κ.

4.7 Οργάνωση της μνήμης.

a) Η έννοια της διευθύνσεως.

Στην κεντρική μνήμη, όπως είπαμε, αποθηκεύονται εντολές και δεδομένα σε διαδικτική μορφή. Για να είναι η μνήμη χρήσιμη, θα πρέπει να υπάρχει κάποιος τρόπος, ώστε να μπορούμε να αποθηκεύμε νέα δεδομένα ή να ανασύρουμε και να επεξεργαζόμαστε δεδομένα, που έχουν ήδη αποθηκευθεί.

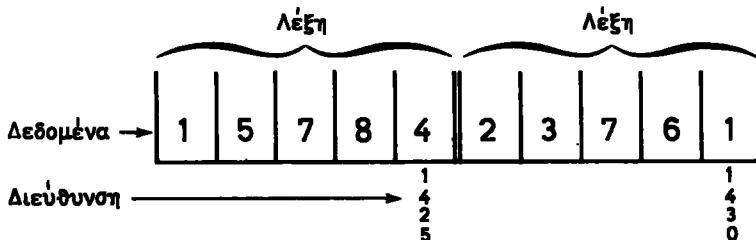
Θα πρέπει λοιπόν να γνωρίζομε σε ποιες ακριβώς θέσεις έχουν τοποθετηθεί οι διάφορες πληροφορίες. Για το σκοπό αυτό εφοδιάζομε τη μνήμη με διευθύνσεις. Όστε η διεύθυνση μνήμης (memory address) είναι ένας αριθμός, που καθορίζει μονοσήμαντα μια περιοχή της μνήμης, όπου αποθηκεύεται μια στοιχειώδης πληροφορία και από όπου μπορεί να ανασυρθεί αργότερα, για να χρησιμοποιηθεί σε κάποια επεξεργασία.

Οι πιμές 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.

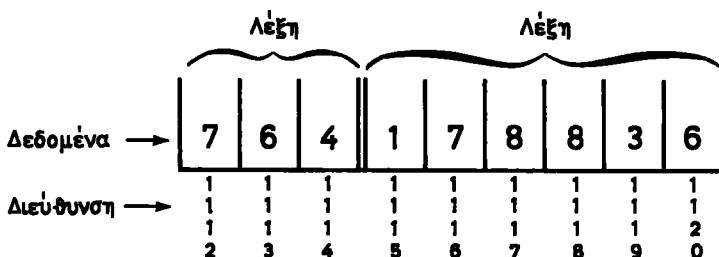
Μια λέξη υπολογιστή (Word) είναι σύνολο χαρακτήρων, που ο υπολογιστής το χειρίζεται σαν μια μονάδα. Ο αριθμός των χαρακτήρων σε μια λέξη μπορεί να είναι σταθερός ή μεταβλητός.

Στους υπολογιστές που έχουν λέξεις σταθερού μήκους (Fixed word length) δίνεται μια ξεχωριστή διεύθυνση σε κάθε περιοχή της μνήμης που αντιστοιχεί σε μια 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. Σημειώνομε ότι ένας χαρακτήρας στον αλφαριθμητικό 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 23781.



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 η καταχώριση της λέξεως στη μνήμη.

4.8 Καταχωριστές (Registers).

Καταχωριστής είναι μια μικρή μνήμη, που χρησιμοποιείται για την προσωρινή αποθήκευση πληροφοριών. Το μέγεθός του κυμαίνεται συνήθως από 1-4 λέξεις. Καταχωριστές υπάρχουν στη μονάδα ελέγχου 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) Απλός 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 έκτελούνται.
- γ) Καταχωριστής διευθύνσεως (address register), όπου καταχωρίζεται η διεύθυνση της μνήμης, όπου βρίσκεται η επόμενη να εκτελεσθεί εντολή.
- δ) Καταχωριστής συσσωρευτής ή αθροιστικός καταχωριστής (accumulator register), όπου καταχωρίζεται το αθροισμα δύο αριθμών.
- ε) Καταχωριστής γινομένου (multiplier register), όπου καταχωρίζεται το αποτέλεσμα ενός πολλαπλασιασμού.
- ζ) Καταχωριστής πηλίκου (quotient register), όπου καταχωρίζεται το πηλίκο μιας διαιρέσεως.
- η) Καταχωριστής αθροίσματος (adder register), όπου εκτελείται μια πρόσθεση.
- θ) Καταχωριστής συγκρίσεως (comparation register), όπου γίνεται η σύγκριση δύο δεδομένων.

4.9 Φλιπ-Φλοπ (Flip-Flops).

Φλιπ-φλοπ είναι ένας ηλεκτρονικός διακόπτης, το κύκλωμα του οποίου αποτελείται βασικά από δύο τρίδοες ηλεκτρονικές λυχνίες ή δύο ημιαγωγούς (κρυσταλλοτριόδους) και περιέχει ένα τμήμα εισόδου για την εισαγωγή του παλμού διεγέρσεως του κυκλώματος και ένα τμήμα εξόδου με δύο ακροδέκτες. Κατάλληλοι ηλεκτρικοί παλμοί, που εισάγονται από το τμήμα εισόδου, άλλοτε προκαλούν την εμφάνιση σήματος από τον ένα ακροδέκτη της εξόδου και άλλοτε από τον άλλο. Η ύπαρξη σήματος σε ένα από τους ακροδέκτες εξόδου αντιστοιχεί στο δυαδικό ψηφίο 1, ενώ η ανυπαρξία σήματος αντιστοιχεί στο δυαδικό ψηφίο 0.

Υπάρχουν διαφόρων τύπων φλιπ-φλοπς, όπως το φλιπ-φλοπ δύο σταθερών καταστάσεων (bistable), μιας σταθερής καταστάσεως (monostable) και το ελεύθερης ταλαντώσεως (astable).

Στα φλιπ-φλοπς δύο σταθερών καταστάσεων, η αρχική κατάσταση των ακροδεκτών εξόδου, μετά τη σπιγμαία διέγερση του κυκλώματος από κάποιο εξωτερικό παλμό, μπορεί να μεταβληθεί. Η επάνοδος στην αρχική κατάσταση γίνεται μόνο με την εφαρμογή ενός νέου κατάλληλου παλμού διεγέρσεως. Αν δεν υπάρχει νέος παλμός διεγέρσεως η κατάσταση των ακροδεκτών παραμένει συνεχώς σταθερή.

Τα φλιπ-φλοπς μιας σταθερής καταστάσεως όταν διεγερθούν παραμένουν για ορισμένο χρονικό διάστημα στη νέα κατάσταση και κατόπιν επανέρχονται στην αρχική.

Τέλος τα φλιπ-φλοπς ελεύθερης ταλαντώσεως διεγέρονται περιοδικά από κά-

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

Πολλά φλιπ-φλοπς, συνδεμένα κατάλληλα μπορούν λόγω του τρόπου λειτουργίας τους να χρησιμοποιηθούν στις μνήμες των υπολογιστών καθώς επίσης και σαν απλοί καταχωριστές (registers) ή σαν μετρητές (counters).

4.10 Μονάδα ελέγχου (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-εντολών 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.

Στον καταχωριστή εντολών τοποθετείται κάθε φορά, φερμένη από τη μνήμη, η εντολή που πρόκειται να εκτελεσθεί, αφού προηγουμένως αποκωδικοποιηθεί, ενώ στον καταχωριστή διευθύνσεων, τοποθετείται κάθε φορά η διεύθυνση της έπομενης 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 απλούς ή αθροιστικούς, και καταχωριστές συγκρίσεως ή αποτελεσμάτων (γινομένου, πηλίκου, διαφοράς, αθροίσματος) 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.

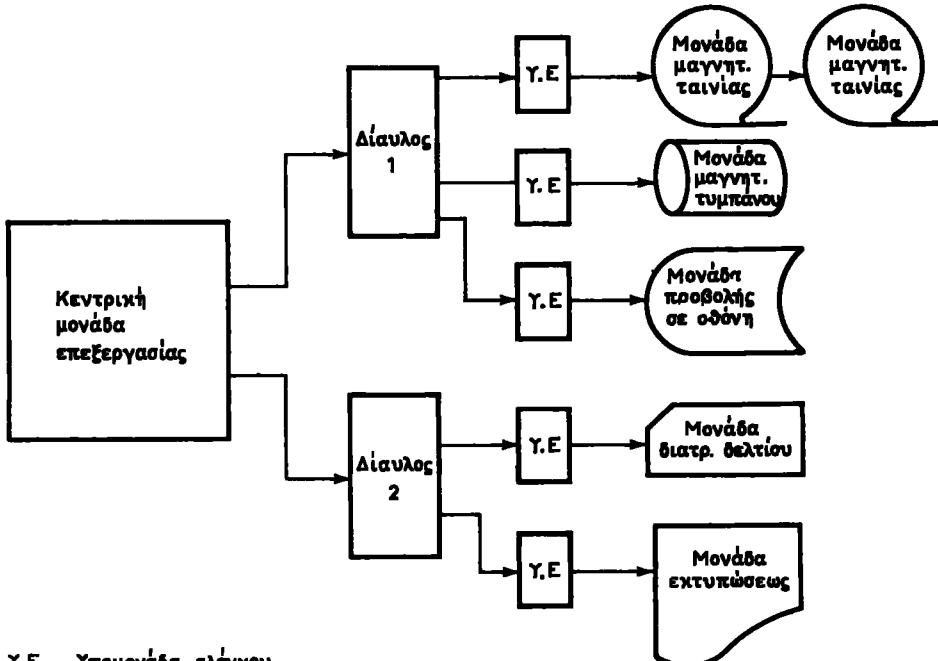
Ένας δίαυλος μπορεί να εξυπηρετεί πολλές περιφερειακές μονάδες (σχ. 4.12).

4.13 Υπομονάδες ελέγχου των περιφερειακών (Controllers).

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

(Status) της μονάδας. Επί πλέον δημιουργεί και ελέγχει τα ψηφία ιστορίας.

Σε πολλές περιφερειακές μονάδες, όπως π.χ η αναγνωστική δελτίων, η υπομονάδα ελέγχου είναι ενσωματωμένη σ' αυτή. Σε άλλες πάλι περιπτώσεις εμφανίζονται ως ανεξάρτητες συσκευές. Σε κάθε υπομονάδα ελέγχου μπορούν να συνδέθουν μέχρι 8 ομοειδείς περιφερειακές μονάδες (σχ. 4.12).



Υ.Ε Υπομονάδα ελέγχου

Σχ. 4.12.

Σύνδεση των περιφερειακών μονάδων με τη CPU μέσω των διαύλων και των υπομονάδων ελέγχου.

Ερωτήσεις.

1. Περιγράψτε με συντομία τον τρόπο αναγνώσεως της τιμής ενός πυρήνα.
2. Υποθέστε ότι σε ένα πυρήνα, που βρίσκεται σε κατάσταση 0, θέλομε να εγγράψουμε την τιμή 1. Ποια διαδικασία θα ακολουθήσουμε;
3. Αναφέρατε τους κυριότερους τύπους καταχωριστών και το σκοπό που εξυπηρετεί ο καθένας.
4. Ποιος ο ρόλος της μονάδας ελέγχου σε ένα συγκρότημα υπολογιστή;
5. Ποιος ο ρόλος της αριθμητικής μονάδας σε ένα συγκρότημα υπολογιστή;
6. Τι είναι οι δίαυλοι (Channels);
7. Τι είναι οι υπομονάδες ελέγχου (Controllers);

ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

5.1 Εντολές – Πρόγραμμα – Γλώσσα υπολογιστή.

Οι υπολογιστές είναι αυτόματες υπολογιστικές μηχανές, που σημαίνει ότι μπορούν να εκτελούν μια ολόκληρη σειρά από λειτουργίες χωρίς την ανθρώπινη παρέμβαση. Πραγματικά, ο υπολογιστής μπορεί να εκτελεί χιλιάδες πράξεις χωρίς καμιά εξωτερική βοήθεια από τον άνθρωπο. Για να το πετύχει όμως χρειάζεται να έχει πάρει από αυτόν τις κατάλληλες οδηγίες. Οι οδηγίες αυτές ονομάζονται **εντολές** (Instructions).

Κάθε εντολή αντιπροσωπεύει μια ξεχωριστή λειτουργία. Οι εντολές που θα εκτελέσει ο υπολογιστής προετοιμάζονται προσεκτικά και γράφονται έτσι, ώστε να αντιπροσωπεύουν τη λογική σειρά, που θα ακολουθήσει για να επιλύσει το συγκεκριμένο πρόβλημα. Αποθηκεύονται στη μνήμη προκαταβολικά.

Ολόκληρη η σειρά των εντολών που απαιτούνται για να επιλυθεί ένα πρόβλημα, αποτελεί το λεγόμενο **πρόγραμμα** (Program).

Οι εντολές διατυπώνονται με καθορισμένο τρόπο έτσι, ώστε να τίς «καταλάβει» ο υπολογιστής. Ένα πλήρες ρεπερτόριο από εντολές αποτελεί μια γλώσσα προγραμματισμού.

5.2 Γλώσσα μηχανής και συμβολική γλώσσα.

Βασικά ένας υπολογιστής υπακούει μόνο στη δική του γλώσσα, τη λεγόμενη **γλώσσα μηχανής** (Machine language). Μια τέτοια γλώσσα περιλαμβάνει εντολές κωδικοποιημένες δυαδικά. Όπως θα δούμε με περισσότερες λεπτομέρειες στην § 5.4, μια εντολή αποτελείται βασικά από δύο τμήματα, τον **κώδικα λειτουργίας** και τις **διευθύνσεις μνήμης** των στοιχείων-δεδομένων, που συμμετέχουν στη λειτουργία. Και ο κώδικας και οι διευθύνσεις εκφράζονται στο δυαδικό σύστημα. Με τη μορφή αυτή οι εντολές είναι αμέσως κατανοητές από τον υπολογιστή. Οι περισσότεροι υπολογιστές της πρώτης γενιάς, για την οποία μιλήσαμε ήδη, πρόγραμματίζονταν μόνο σε γλώσσα μηχανής. Εξ αιτίας όμως της δυαδικής πολυψηφιακής παραστάσεώς τους η απομνημόνευση και ο προγραμματισμός είναι πολύ δύσκολος.

Σαν πρώτο βήμα απλουστεύσεως έχομε την αντιστοίχιση των κωδικών εντολών και των διευθύνσεων με δεκαδικούς αριθμούς. Στον πίνακα 5.2.1, βλέπομε μια μορφή κωδικοποίησεως των λειτουργιών ενός υποθετικού υπολογιστή.

Παρ' όλο όμως που η κωδικοποίηση αυτή είναι βελτιωμένη, ο προγραμματισμός

εξακολουθεί να αποτελεί δυσκολότατη εργασία. Μεγαλύτερη απλούστευση είναι δυνατή με την αντιστοίχιση γραμμάτων ή και μικρών λέξεων στους αριθμούς, που συμβολίζουν τους κωδικούς εντολών (βλέπε πίνακα 5.2.1), ενώ οι διευθύνσεις παραμένουν στη δεκαδική αριθμητική τους έκφραση.

Και με τη μορφή αυτή ακόμα οι εντολές θεωρούμε ότι εκφράζουν τη γλώσσα μηχανής.

ΠΙΝΑΚΑΣ 5.2.1.

Λεπτουργία	Δυαδική έκφραση	Δεκαδική έκφραση	Συμβολική έκφραση
Πρόσθεση	001010	10	A
Αφαίρεση	001011	11	S
Πολλαπλασιασμός	001100	12	X
Διαίρεση	001101	13	D
Άλμα	001111	15	J
Εγγραφή στον αθροιστικό καταχωριστή Ac ¹	010000	16	R
Ανάγνωση από τον αθροιστικό καταχωριστή Ac	010001	17	C
Εκτύπωση	010010	18	W
Έλεγχος θετικού προσήμου	010100	20	P
Έλεγχος αρνητικού προσήμου	010101	21	M
Έλεγχος μηδενικού περιεχομένου	010110	22	Z
Διακοπή	011001	25	H

1 Τον αθροιστικό καταχωριστή ή συσσωρευτή (Accumulator Register) θα τον συμβολίζομε από δω και πέρα με Ac.

Για μεγαλύτερη ακόμα απλούστευση χρησιμοποιήθηκαν γράμματα στη θέση των δεκαδικών αριθμών για τον συμβολισμό των διευθύνσεων. Έτσι δημιουργήθηκαν οι λεγόμενες συμβολικές γλώσσες (Assembly Languages) με τη βοήθεια των οποίων προγραμματίζονταν οι περισσότερες μηχανές της δεύτερης γενιάς.

Οι συμβολικές γλώσσες διακρίνονται σε δύο κατηγορίες:

- α) Σε εκείνες, όπου σε κάθε εντολή της συμβολικής γλώσσας αντιστοιχεί μία ακριβώς εντολή της γλώσσας μηχανής.
- β) Σε εκείνες, όπου μία εντολή συμβολικής γλώσσας αντιστοιχεί σε ένα σύνολο από εντολές σε γλώσσα μηχανής και που ονομάζονται μακροεντολές (macros).

Η μετατροπή ενός προγράμματος, που έχει γραφεί σε συμβολική γλώσσα, στη γλώσσα μηχανής γίνεται αυτόματα από τη μηχανή με τη βοήθεια ειδικού προγράμματος αντιστοιχίας, του λεγόμενου μεταφραστικού προγράμματος (Assembler ή Assembler compiler), με το οποίο η κατασκευάστρια εταιρία εφοδιάζει το συγκεκριμένο υπολογιστή.

Τέτοιες συμβολικές γλώσσες υπάρχουν αρκετές με διάφορες παραλλαγές, όπως η BAL (Basic Assembly Language), η COMPASS, η SIGMA κ.ο.κ.

Παρ' όλο που οι γλώσσες μηχανής και οι συμβολικές γλώσσες κάνουν πιο αποδοτική την εκμετάλλευση της μηχανής, εν τούτοις είναι γλώσσες προσαρμοσμένες στη μηχανή (Machine Oriented Languages). Αυτό υποχρεώνει τον προγραμματι-

στη να περιορίζει τή λειτουργικότητα ενός προγράμματός του σε συγκεκριμένα συστήματα ή να τον αναγκάζει να ξαναγράψει το πρόγραμμα σε καινούργια γλώσσα, αν θα ήθελε να το χρησιμοποιήσει σε διαφορετικό σύστημα.

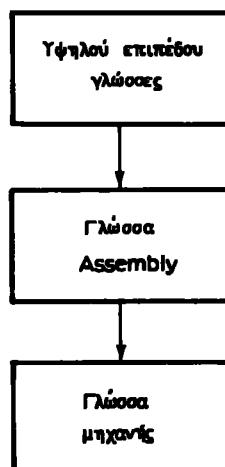
5.3 Γλώσσες υψηλού επιπέδου (High Level Languages).

Η δημιουργία πολλών τύπων υπολογιστών ανάγκασε τους ειδικούς να σχεδιάσουν νέες προγραμμάτισης διεθνείς γλώσσες συνεννοήσεως με τον υπολογιστή, οι οποίες είναι ανεξάρτητες από τον τύπο της μηχανής (Machine independent), αλλά προσαρμοσμένες στις απαιτήσεις ενός συγκεκριμένου τύπου προβλημάτων (Problem Oriented Languages). Οι γλώσσες αυτές, που μπορούν να χρησιμοποιηθούν από πολλούς διαφορετικούς τύπους (της ίδιας ή διαφορετικών κατασκευαστριών εταιριών), χαρακτηρίζονται και ως γλώσσες υψηλού επιπέδου.

Χωρίζονται σε δύο κυρίως κατηγορίες. Στην πρώτη ανήκουν εκείνες, που χρησιμοποιούνται στην επίλυση επιστημονικών και τεχνικών προβλημάτων, όπου δηλαδή κυριαρχεί η χρήση των μαθηματικών. Τέτοιες γλώσσες είναι η FORTRAN, η ALGOL, η BASIC κλπ. Στην άλλη κατηγορία ανήκουν οι γλώσσες που χρησιμοποιούνται στην επίλυση εμπορικών και οικονομικών προβλημάτων (λογιστήρια, αποθήκες, μισθοδοσίες κλπ). Τέτοιες γλώσσες είναι η COBOL, η RPG, η PL1 κλπ.

Επειδή όμως ο υπολογιστής, όπως είπαμε, καταλαβαίνει μόνο τη δική του γλώσσα, τα προγράμματα που είναι γραμμένα σε κάποια από τις γλώσσες υψηλού επιπέδου (Source programs) μεταφράζονται στη γλώσσα μηχανής (Object programs) από τόν ίδιο τον υπολογιστή, με τη βοήθεια ειδικών μεταφραστικών προγραμμάτων (Compilers), με τα οποία οι κατασκευάστριες εταιρίες εφοδιάζουν τις μηχανές τους.

Την ιεραρχία των γλωσσών προγραμματισμού βλέπομε στο σχήμα 5.3.



Σχ. 5.3.
Η ιεραρχία των γλωσσών προγραμματισμού.

5.4 Κώδικες λειτουργίας και διευθύνσεις.

Μια εντολή αποτελείται, τις περισσότερες φορές, από δύο κύρια τμήματα:

- α) Τον κώδικα λειτουργίας ή εντολής (Operation Code).
- β) Τις διευθύνσεις (Operands).

Ο κώδικας λειτουργίας καθορίζει τη συγκεκριμένη λειτουργία που πρόκειται να εκτελεσθεί, ενώ οι διευθύνσεις παριστάνουν τις διευθύνσεις των θέσεων της μνήμης, τα περιεχόμενα των οποίων. Θα συμμετάσχουν στη λειτουργία αυτή. Το περιεχόμενο μιας διευθύνσεως Ad θα συμβολίζομε από δω και πέρα με c (Ad) ή απλώς [Ad], ενώ το περιεχόμενο του συσσωρευτή¹ με c (Ac) ή απλώς [Ac].

Τις εντολές τις διακρίνομε ανάλογα με το πλήθος των διευθύνσεων που εμφανίζονται στο 2ο τμήμα τους, ως εντολές μιας, δύο ή τριών διευθύνσεων. Το πλήθος των στοιχείων, που διατίθεται για την καταχώριση του κώδικα εντολής και των διευθύνσεων, εξαρτάται από τον τύπο του υπολογιστή.

α) Εντολή μιας διευθύνσεως (One Address Instruction).

Έχει τη γενική μορφή

Κώδικας εντολής	Διεύθυνση
-----------------	-----------

Στην εκτέλεση μιας εντολής αυτού του τύπου, εκτός από το περιεχόμενο της διευθύνσεως της μνήμης, μετέχει ενεργά και ο συσσωρευτής με το περιεχόμενό του. Έτσι π.χ. στην περίπτωση μιας απλής προσθέσεως η εντολή θα έχει τη μορφή

Κώδικας εντολής	Διεύθυνση
10	1472

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

και σημαίνει ότι προστίθεται το περιεχόμενο της διευθύνσεως 1472 στο περιεχόμενό του συσσωρευτή και το άθροισμα τίθεται πάλι στο συσσωρευτή. Ο κώδικας της προσθέσεως είναι 10.

β) Εντολή δύο διεθύνσεων.

Έχει τη γενική μορφή

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

Στην εντολή αυτή δίνονται οι διευθύνσεις των δύο στοιχείων που θα πάρουν μέρος σε μια πράξη. Το αποτέλεσμα μπαίνει στο συσσωρευτή. Π.χ.:

Κώδικας εντολής	Διεύθυνση 1	Διεύθυνση 2
11	1652	7212

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

¹ Υπενθυμίζομε ότι ο συσσωρευτής είναι ένας καταχωριστής (register), που βρίσκεται στην αριθμητική μονάδα.

και σημαίνει ότι αφαιρείται (κώδικας εντολής = 11) το c(7212) από το c(1652) και η διαφορά καταχωρίζεται στο συσσωρευτή.

γ) Εντολή τριών διευθύνσεων.

Έχει τη γενική μορφή:

Κώδικας εντολής	Διεύθυνση 1	Διεύθυνση 2	Διεύθυνση 3
-----------------	-------------	-------------	-------------

Στην περίπτωση αυτή δίνονται οι διευθύνσεις των δύο στοιχείων, που συμμετέχουν στη λειτουργία που περιγράφει ο κώδικας εντολής (διεύθυνση 1 και διεύθυνση 2) και η διεύθυνση όπου θα τοποθετηθεί το αποτέλεσμα (διεύθυνση 3). Π.χ.:

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

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

και σημαίνει ότι αφαιρείται από το περιεχόμενο της διεύθυνσεως 3046 το περιεχόμενο της διεύθυνσεως 7923 και το αποτέλεσμα καταχωρίζεται στη διεύθυνση 8561.

5.5 Μηχανισμός εκτελέσεως ενός προγράμματος.

Όπως είπαμε στην παράγραφο 4.10 στη μονάδα ελέγχου υπάρχει ένας καταχωριστής εντολών, που τον συμβολίζουμε συνήθως με IR (Instruction Register), στον οποίο μεταφέρεται από τη μνήμη η εντολή που πρόκειται κάθε φορά να εκτελεσθεί, και ένας καταχωριστής διευθύνσεων, όπου καταχωρίζεται η διεύθυνση της μνήμης, που βρίσκεται αποθηκευμένη η επόμενη να εκτελεσθεί εντολή.

Ας δούμε τώρα πώς λειτουργεί ο μηχανισμός εκτελέσεως ενός προγράμματος σε ένα υπολογιστή δύο διευθύνσεων με ένα απλό παράδειγμα.

Έστω ότι θέλομε να προσθέσουμε τους αριθμούς 524 και 242, που έχουν διατυπωθεί επάνω σε ένα δελτίο. Το πρόγραμμα θα αποτελείται από τις εξής εντολές (σε περιγραφική μορφή):

- α) Διάβασε από το δελτίο τους δύο αριθμούς.
- β) Κάνε την πρόσθεση.
- γ) Μετάφερε το αποτέλεσμα στη μνήμη.
- δ) Γράψε το αποτέλεσμα.
- ε) Σταμάτησε.

Ας υποθέσουμε ότι το κωδικοποιημένο αυτό πρόγραμμα το έχομε αποθηκεύσει στις θέσεις της μνήμης με διευθύνσεις 101, 102, 103, 104 και 105. Μόλις δοθεί το σήμα για την εκκίνηση, ο καταχωριστής διευθύνσεων έχει τιμή 101 και η πρώτη εντολή μεταφέρεται μέσα στον καταχωριστή εντολών (IR) και αρχίζει η εκτέλεσή της. Έτσι θα διαβασθεί ένα δελτίο και τα δεδομένα θα τοποθετηθούν στις διευθύνσεις 201 και 202 αντίστοιχα:

Εντολή 1η

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

Η περιοχή της μνήμης όπου αποθηκεύθηκαν το πρόγραμμα και τα δεδομένα θα παρουσιάζει την εξής εικόνα (σχ. 5.5).

MNHMH
ΔΙΕΥΘΥΝΣΗ ΠΕΡΙΕΧΟΜΕΝΟ ΔΙΕΥΘΥΝΣΗ ΠΕΡΙΕΧΟΜΕΝΟ

101	1η εντολή	201	524
102	2η εντολή	202	242
103	3η εντολή	203	
104	4η εντολή		
105	5η εντολή		
106			

Σχ. 5.5.
Η κατάσταση της μνήμης.

Παράλληλα η τιμή του καταχωριστή διευθύνσεων θα αυξηθεί αυτόματα κατά 1 και έτσι θα μεταφερθεί στον καταχωριστή εντολών η επόμενη εντολή, που βρίσκεται στη διεύθυνση 102 και έχει τη μορφή:

Εντολή 2η

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

Με την εκτέλεσή της θα προστεθεί το περιεχόμενο της διευθύνσεως 201 (δηλαδή που βρίσκεται καταχωρισμένος ο αριθμός 524) στο περιεχόμενο της διευθύνσεως 202, (όπου βρίσκεται ο 242) και το αποτέλεσμα θα καταχωρισθεί στο συσσωρευτή. Συγχρόνως η τιμή του καταχωριστή διευθύνσεων θα γίνει 103 και θα μεταφερθεί η τρίτη εντολή στον καταχωριστή εντολών, που έχει τη μορφή:

Εντολή 3η

Μετέφερε	Ac	.203
----------	----	------

Με την εκτέλεσή της θα μεταφερθεί το αποτέλεσμα από το συσσωρευτή στη διεύθυνση 203 της μνήμης.

Η τιμή του καταχωριστή διευθύνσεων εν τω μεταξύ έχει γίνει 104 και στον καταχωριστή εντολών μεταφέρεται η τέταρτη εντολή, που έχει τη μορφή:

Εντολή 4η

Γράψε	203	-
-------	-----	---

με την οποία γράφει το αποτέλεσμα, που βρίσκεται στη διεύθυνση 203, στην περιφερειακή μονάδα εκτυπώσεως.

Με την τελευταία εντολή, που θα μεταφερθεί στον καταχωριστή εντολών γίνεται και η διακοπή της λειτουργίας της μηχανής.

Αν σχεδιάζαμε το διο πρόγραμμα για μια μηχανή μιας διευθύνσεως, θα έπρεπε να προσθέσουμε και άλλες εντολές (πην εντολή προσθέσεως θα αντικαθιστούσαν δύο εντολές προσθέσεως), ενώ σε μια μηχανή τριών διευθύνσεων θα είχαμε λιγότερες εντολές προσθέσεως.

τερες εντολές (η εντολή προσθέσεως και η εντολή μεταφοράς θα είχαν αντικατασταθεί από μία και μόνη εντολή).

5.8 Κατηγορίες εντολών.

Η εκλογή ενός κώδικα εντολών απαιτεί μεγάλη προσοχή και παρουσιάζει αρκετές τεχνικές δυσκολίες στο σχεδιασμό των καταλλήλων κυκλωμάτων. Από την άλλη όμως πλευρά πρέπει να λαμβάνεται υπ' όψη και η ευκολία με την οποία θα μπορεί να προγραμματισθεί κάποιο πρόβλημα.

Ποιες όμως είναι οι βασικές εντολές για τη λειτουργία ενός προγράμματος; Πολλές από τις πρώτες μηχανές εκτελούσαν μόνο πρόσθεση, αφαίρεση και πολλαπλασιασμό, ενώ η διαίρεση έπρεπε να προγραμματισθεί κατάλληλα. Όλοι οι σημερινοί υπολογιστές εκτελούν αυτόματα και τις 4 αριθμητικές πράξεις.

Οι εντολές αυτές μαζί με εντολές για μετακίνηση δεδομένων από τη μνήμη προς την αριθμητική μονάδα και αντίστροφα, όπως επίσης και από μια περιοχή της μνήμης σε άλλη αποτελούν τη βάση για οποιοδήποτε κώδικα. Κάθε μηχανή, εξ άλλου πρέπει να διαθέτει επίσης τουλάχιστον δύο εντολές ελέγχου, που θα ελέγχουν την ικανοποίηση ορισμένων συνθηκών και θα καθορίζουν τη σειρά, με την οποία θα εκτελεσθούν οι διάφορες εντολές ενός προγράμματος, ανάλογα με τα αποτελέσματα της συγκρίσεως.

Ο λόγος γι' αυτό είναι ότι πρακτικά οι περισσότεροι υπολογισμοί, που πρέπει να γίνουν, επαναλαμβάνονται πολλές φορές, ώστε οι ίδιες εντολές να εκτελούνται με την ίδια σειρά αλλά με διαφορετικά δεδομένα για κάθε επανάληψη. Αν λοιπόν δεν υπήρχαν οι εντολές ελέγχου, η χρήση του υπολογιστή θα ήταν εξαιρετικά αντιοκονιμική, εφ' όσον ο χρόνος για την προετοιμασία του προγράμματος θα υπερέβαινε το χρόνο υπολογισμού με το χέρι. Θα αναφερθούμε σε ένα πολύ απλό παράδειγμα.

Ας υποθέσουμε ότι θέλομε να προσθέσουμε 100 αριθμούς που έχουν τοποθετηθεί στη μνήμη, στις διευθύνσεις 1-100. Ένα πρόγραμμα, που θα έκανε αυτή τη δουλειά, θα έπρεπε να περιέχει 100 εντολές προσθέσεως όπως:

«Πρόσθεσε το περιεχόμενο της διευθύνσεως 1 στο συσσωρευτή»

«Πρόσθεσε το περιεχόμενο της διευθύνσεως 2 στο συσσωρευτή»

«Πρόσθεσε το περιεχόμενο της διευθύνσεως 3 στο συσσωρευτή»

K.O.K.

Υποθέτοντας βέβαια ότι ο συσσωρευτής είναι αρχικά κενός. Αυτό όμως είναι εξαιρέτικά αντιοκονιμικό, αν μάλιστα η διαδικασία, στην προκειμένη περίπτωση η πρόσθεση, περιέχει πολλές εντολές αντί για μία και επαναλαμβάνεται 1000 φορές αντί για 100. Αυτό σημαίνει επίσης και τεράστιο χώρο, για την καταχώριση όλων αυτών των ομίσων εντολών στη μνήμη, αλλά και σπατάλη χρόνου για το γράψιμο των εντολών.

Σε τέτοιες περιπτώσεις η όλη διαδικασία γίνεται εύκολα και απλά με τη βοήθεια μιας εντολής ελέγχου (ή διακλαδώσεως). Έτσι, αν θέλομε να εκτελεσθεί επαναληπτικά οι φορές η ίδια ομάδα από εντολές και έπειτα από τις να επαναλήψεις να ακολουθηθεί και να εκτελεσθεί μία άλλη ομάδα από εντολές, αποθηκεύομε στην αρχή σε κάποιο καταχωριστή την τιμή 1 και προσθέτομε σ' αυτόν μετά από κάθε επανά-

ληψη τη μονάδα, ενώ συγχρόνως ελέγχομε το περιεχόμενό του αν ισούται με την τιμή ν. Ύστερα από ν επαναλήψεις ο καταχωριστής θα έχει φθάσει την τιμή ν και η εντολή διακλαδώσεως θα παραπέμψει σε άλλο κλάδο του προγράμματος.

Τέλος θα πρέπει να υπάρχουν δύο εντολές από τις οποίες με τη μία θα μπορούμε να διαβάσουμε δεδομένα από κάποια περιφερειακή μονάδα και με την άλλη να δώσουμε αποτελέσματα σε κάποιαν άλλη.

Μπορούμε λοιπόν να κατατάξουμε τις εντολές ενός υπολογιστή στις εξής βασικές κατηγορίες:

- α) Αριθμητικές εντολές.
- β) Λογικές εντολές ή εντολές συγκρίσεως ή διακλαδώσεως.
- γ) Εντολές μεταφοράς ή μετακινήσεως.
- δ) Εντολές εισόδου-εξόδου.

Με τις αριθμητικές εκτελούμε τις 4 αριθμητικές πράξεις, με τις λογικές, συγκρίσεις και ελέγχους, με τις διακινήσεως εκτελούμε μεταφορά από μία περιοχή της μνήμης σε άλλη ή από τη μνήμη στο συσσωρευτή και αντίστροφα. Τέλος με τις εντολές εισόδου-εξόδου προβαίνουμε σε ανάγνωση ή εγγραφή.

5.7 Τι είναι αλγόριθμος.

Συχνά θα ακούσουμε να γίνεται λόγος για αλγόριθμους. Αλγόριθμος είναι μία μέθοδος αριθμητικής επιλύσεως ενός μαθηματικού προβλήματος, που βασίζεται στην επαναληπτική εφαρμογή και εκτέλεση μιας σειράς απλών διαδικασιών (αριθμητικών πράξεων και λογικών συγκρίσεων).

Έτσι π.χ. η διαδικασία για την εύρεση του Μέγιστου Κοινού Διαιρέτη ή του Ελάχιστου Κοινού Πολλαπλασίου, της τετραγωνικής ρίζας κλπ. αποτελούν αλγόριθμους.

Η λέξη αλγόριθμος προέρχεται από τον Άραβα μαθηματικό *Alkhowarazim*, που έδωσε διάφορους κανόνες για την εκτέλεση των 4 βασικών πράξεων της αριθμητικής.

Επειδή ο υπολογιστής μπορεί να εκτελέσει μόνο απλές αριθμητικές πράξεις, σύνθετοι υπολογισμοί, που περιλαμβάνουν παραγωγίσεις, ολοκληρώσεις κλπ. θα πρέπει να περιγραφούν με τη βοήθεια καταλλήλων αλγορίθμων. Τις μεθόδους αυτές μελετά ένας ξεχωριστός κλάδος των μαθηματικών, που ονομάζεται Αριθμητική Ανάλυση και που παρουσιάζει ιδιαίτερη ανάπτυξη τα τελεύταία χρόνια.

Ασκήσεις.

1. Υποθέσατε ότι διαθέτετε υπολογιστή δύο διευθύνσεων. Να περιγράψετε με τον ίδιο τρόπο, που χρησιμοποίήσαμε στην παράγραφο 5.5, ένα στοιχειώδες πρόγραμμα και τον τρόπο λειτουργίας του, αν θέλομε να υπολογίσουμε και να τυπώσουμε την τιμή της παραστάσεως

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

όπου οι τιμές των A, B, C, D είναι διατρυπμένες ανά δύο σε δύο δελτία.

2. Υποθέσατε ότι θέλετε να υπολογίσετε το μέσο όρο 4 αριθμών A, B, C, D, που βρίσκονται καταχωρισμένοι στις διευθύνσεις 510, 511, 512, 513 της μνήμης. Αν ο υπολογιστής, που διαθέτετε, είναι μιας διευθύνσεως, πώς θα κατασκευάσετε το αντίστοιχο πρόγραμμα; Να χρησιμοποιήσετε ως κωδικούς εντολών τους κωδικούς που αναφέρονται στον πίνακα 5.2.1 της παραγράφου 5.2. Πώς θα διασκευάσετε το πρόγραμμα, αν ο υπολογιστής ήταν δύο διευθύνσεων;

ΚΕΦΑΛΑΙΟ ΕΚΤΟ

ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ

6.1 Σύμβολα διαγράμματος ροής.

Όπως είπαμε, για να μας επιλύσει κάποιο πρόβλημα ο υπολογιστής πρέπει να τον εφοδιάσουμε με συγκεκριμένες οδηγίες-εντολές, για τι ακριβώς επιθυμούμε να κάνει. Οι οδηγίες αυτές καταγράφονται έτσι, ώστε να εμφανίζουν τη λογική σειρά, που θα πρέπει να τηρηθεί στην επίλυση του προβλήματος.

Η απεικόνιση με ειδικά σύμβολα των οδηγιών, που θα ακολουθηθούν για την επίλυση ενός προβλήματος, ονομάζεται διάγραμμα ροής (Flow chart) ή λογικό διάγραμμα. Κάθε σύμβολο διαγράμματος ροής παριστάνει μια συγκεκριμένη και διακεκριμένη λειτουργία.

Παρακάτω παρέχουμε τα κυριότερα σύμβολα, που χρησιμοποιούνται στο σχεδιασμό λογικών διαγραμμάτων.

Σύμβολο	Λειτουργία
	Απλή λειτουργία. Εκτέλεση μιας απλής πράξεως ή ενός απλού υπολογισμού (Process).
	Υποβολή ερωτήσεως, ή σημείο αποφάσεως, σύγκριση, ικανοποίηση μιας συνθήκης κλπ. (Decision).
	Γενικό σύμβολο για είσοδο-ανάγνωση δεδομένων ή εξόδο-εγγραφή αποτελεσμάτων. (Input-Output).
	Βέλη, που δείχνουν τη διεύθυνση ροής του λογικού διαγράμματος.
	Εγγραφή ή ανάγνωση από αρχείο σε μαγνητική ταινία (Magnetic tape).

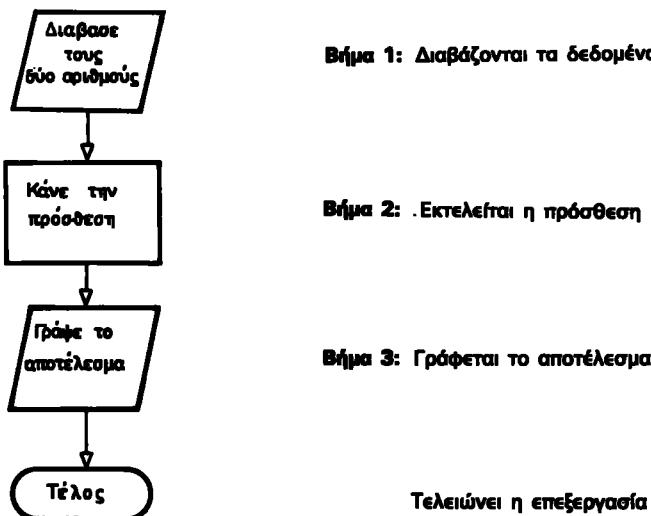
Σύμβολο	Λειτουργία
	Εγγραφή ή ανάγνωση από αρχείο σε μαγνητικό τύμπανο (Magnetic drum).
	Ανάγνωση δεδομένων από διάτρητα δελτία ή διάτρηση αποτελεσμάτων σε δελτία (Punched cards).
	Εγγραφή ή ανάγνωση από αρχείο σε μαγνητικό δίσκο (Magnetic disk).
	Εκτύπωση σε χαρτί (Printer).
	Ανάγνωση από χαρτοταινία ή διάτρηση σε χαρτοταινία (Punched Paper tape).
	Εισαγωγή δεδομένων μέσω πληκτρολόγησης (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 δέθροισμα 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.

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



Παράδειγμα 2.

Έστω ότι μας ζητούν να επιλύσουμε μια εξίσωση πρώτου βαθμού με ένα άγνωστο. Όπως ξέρουμε, 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 μια πιο ολοκληρωμένη μελέτη του προβλήματος θα πρέπει να προχωρήσουμε και σε μια μικρή διερεύνηση. Έτσι για τις διάφορες τιμές των A , B μπορεί να έχουμε τις εξής περιπτώσεις:

- a) Αν $A \neq 0$, τότε η λύση που θα δίνεται από την (1) θα είναι κάποιος πραγματικός αριθμός.
 - β) Αν $A = 0$ και $B \neq 0$ τότε λέμε ότι έχουμε λύση αδύνατη.
 - γ) Αν $A = 0$ και $B = 0$ τότε λέμε ότι έχουμε λύση αδριστη.
- Ας δούμε πώς θα διαμορφώσουμε το λογικό διάγραμμα.

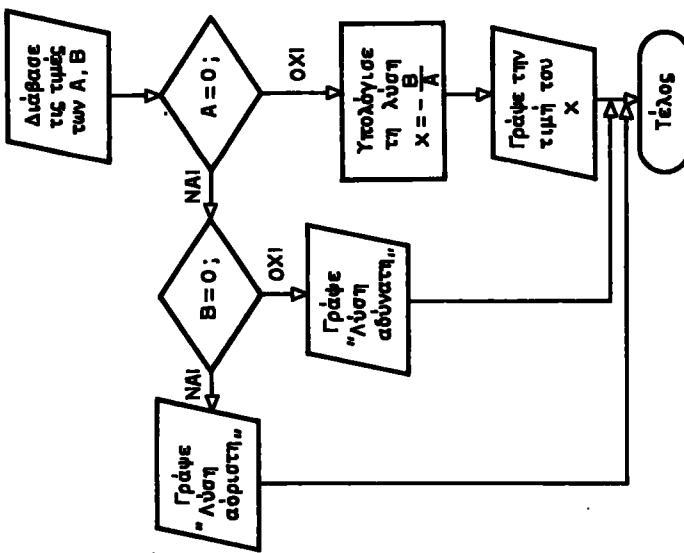
Βήμα 1: Διαβάζουμε τα δεδομένα A και B.

Βήμα 2: Ρυτάμε αν το A είναι μηδέν. Αν δχ προχωρούμε ποι κάπια. Αν όχι ακολουθούμε τον αριστερό κλάδο.

Βήμα 3: Υπολογίζουμε τη λύση.

Βήμα 4: Την τυπώνουμε.

Τέλος: Επεξεργασίας.

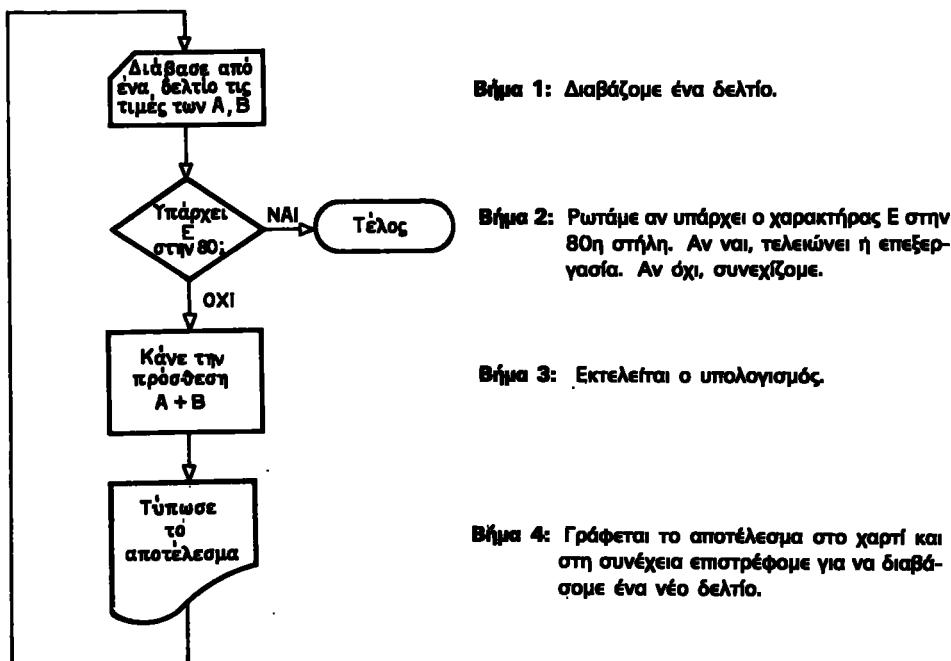


Βήμα 2α: Σε περίπτωση που το A είναι 0, ρυτάμε αν και το B = 0. Αν ναι τυπάνουμε το μήνυμα «ΛΥΣΗ ΑΟΡΙΣΤΗ». Αν δχ τυπάνουμε το μήνυμα «ΛΥΣΗ ΑΛΓΕΝΑΤΗ». Και στης δύο περιπτώσεις αδηγούμαστε στη συνέχεια το τέλος επεξεργασίας.

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

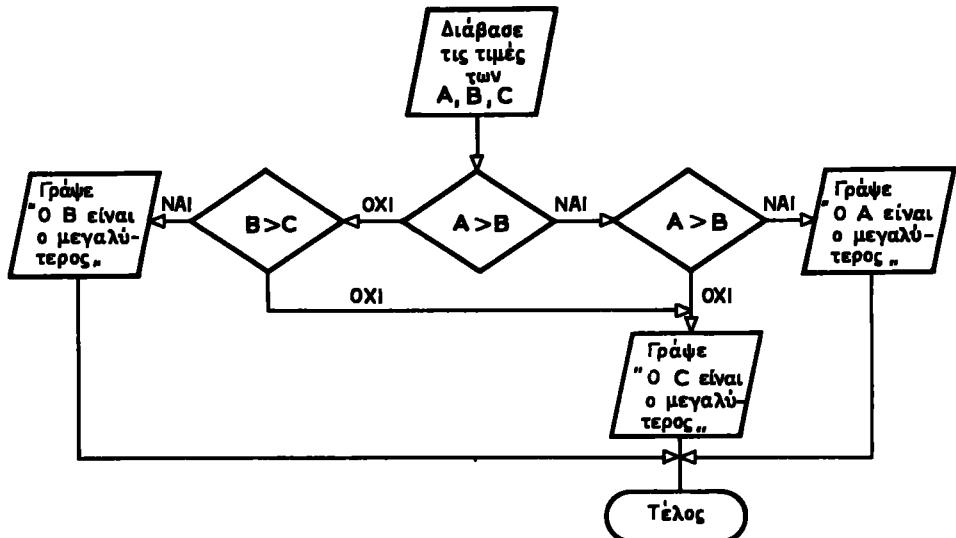
Παράδειγμα 3.

Ας υποθέσουμε ότι θέλομε να λύσομε και πάλι το πρόβλημα του παραδείγματος 1 μέ τη διαφορά ότι δεν δίνεται ένα μόνο ζευγάρι τιμών A και B, αλλά περισσότερα. Ας υποθέσουμε ακόμα ότι τα δεδομένα μας τα έχομε 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, προσθέτομε στο τέλος ένα δελτίο, που έχει επάνω του κάποιο σημάδι που δεν το έχουν όλα τα άλλα, π.χ το χαρακτήρα E στην 80η στήλη. Κάθε φορά, επομένως, που θα διαβάζεται ένα δελτίο, πριν προχωρήσουμε στην επεξεργασία, θα ρωτάμε μήπως είναι το τελευταίο, δηλαδή αν υπάρχει στην 80η στήλη ο χαρακτήρας E. Έτσι σχεδιάζουμε το παρακάτω διάγραμμα ροής.



Παράδειγμα 4.

Έστω ότι δίνονται τρεις αριθμοί A, B, C άνισοι μεταξύ τους και ζητείται να βρεθεί ο μεγαλύτερος από αυτούς. Το λογικό διάγραμμα θα έχει την εξής μορφή:

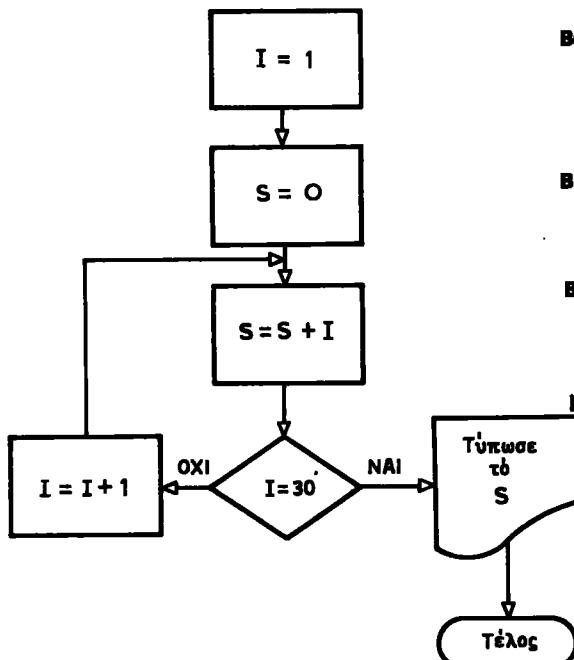


Παράδειγμα 5.

Έστω ότι ζητείται να βρεθεί το άθροισμα των 30 πρώτων φυσικών αριθμών (σαν φυσικοί χαρακτηρίζονται οι θετικοί ακέραιοι 1, 2, 3, 4, ...). Στην περίπτωση αυτή δεν χρειάζεται να διαβασθούν οι 30 φυσικοί αριθμοί, αλλά είναι δυνατό να ορισθεί ένας μετρητής (Counter), δηλαδή μια 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 αθροιστή, 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 αθροιστής 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 η συνθήκη που ακολουθεί, θα μας οδηγήσει στον αριστερό κλάδο του διαγράμματος, όπου στον I προστίθεται μία μονάδα και αποκτά περιεχόμενο 2, που είναι ο επόμενος φυσικός αριθμός. Στη συνέχεια οδηγούμαστε στην εντολή S = S + I όπου προστίθεται η νέα τιμή του I = 2 στον S, που αποκτά περιεχόμενο $1 + 2 = 3$. Γίνεται ξανά η ερώτηση για το αν ο μετρητής I έχει φθάσει την τιμή 30. Η απάντηση είναι αρνητική και οδηγούμαστε και πάλι στον αριστερό κλάδο κ.ο.κ.

Η κυκλική αυτή διαδρομή επαναλαμβάνεται 30 φορές μέχρι τη στιγμή που ο I θα έχει αποκτήσει την τιμή 30, οπότε η ερώτηση θα μας οδηγήσει στο δεξιό κλάδο. Τη στιγμή αυτή τυπώνεται το περιεχόμενο του S που είναι και η τιμή που ζητάμε και τελειώνει η επεξεργασία.



Βήμα 1: Δίνομε στο μετρητή I αρχική τιμή 1.

Βήμα 2: Δίνομε στον αθροιστή S αρχική τιμή 0.

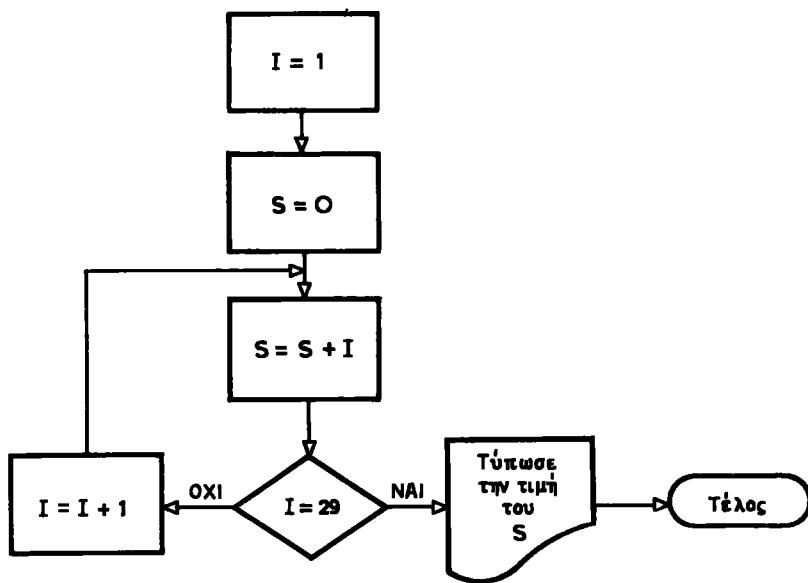
Βήμα 3: Προσθέτομε την τιμή του I (που συμπίπτει με ένα από τους φυσικούς αριθμούς) στον S .

Βήμα 4: Ρωτάμε αν ο μετρητής I έχει φθάσει την τιμή 30. Αν ναι, ακολούθούμε το δεξό κλάδο, γράφομε το αποτέλεσμα (δηλ. την τελευταία τιμή του S) και τελεώνομε. Αν όχι, ακολουθούμε τον αριστερό κλάδο όπου αυξάνεται η τιμή του I κατά 1 και επιστρέφομε στην εντολή του βήματος 3.

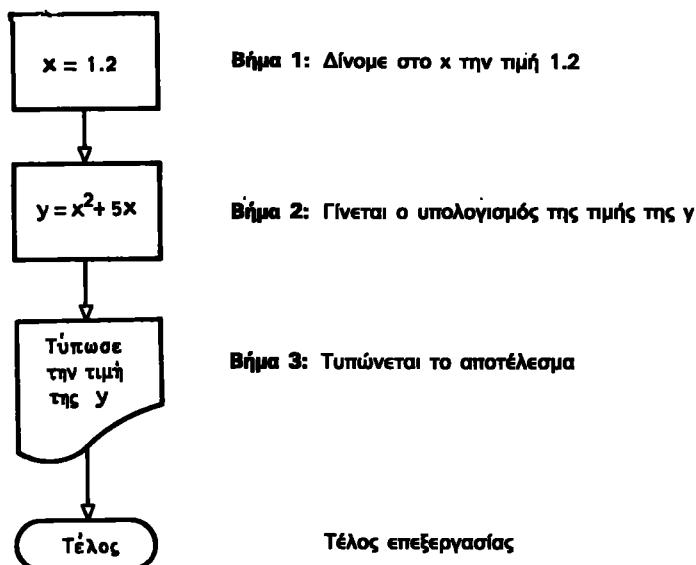
Παράδειγμα 6.

Να βρεθεί το άθροισμα των περιπτών αριθμών από το 1 ως το 30. Εδώ το πρόβλημά μας είναι λίγο διαφορετικό από το προηγούμενο. Πώς θα μπορούσαμε να σχηματίσουμε τους ζητούμενους διαδοχικούς περιπτούς αριθμούς χωρίς να χρειάζεται να τους διαβάσουμε; Θα πρέπει και πάλι να χρησιμοποιήσουμε ένα μετρητή I με αρχική τιμή 1¹, αλλά τώρα να αυξάνομε το περιεχόμενό του κατά 2 (βήμα) μονάδες κάθε φορά, έτσι που να πάρομε τους 1, 3, 5,... Επίσης αντί για 30 στην ερώτηση, πρέπει να γράψουμε 29, γιατί ο 29 είναι ο τελευταίος περιπτώς αριθμός πριν το 30. Κατά τα άλλα, το διάγραμμα ροής παραμένει το ίδιο.

¹ Η εκλογή των αρχικών τιμών εξαρτάται από τη μορφή της εφαρμογής. Κατά περίπτωση επίσης εκλέγεται και ο αριθμός-κριτήριο διακοπής της επεξεργασίας που εμφανίζεται στην ερώτηση. Πάντως όταν χρησιμοποιούμε αθροιστές πρέπει να τους μηδενίζουμε στην αρχή, ενώ όταν χρησιμοποιούμε πολλαπλασιαστές πρέπει να δίνομε αρχική τιμή 1.

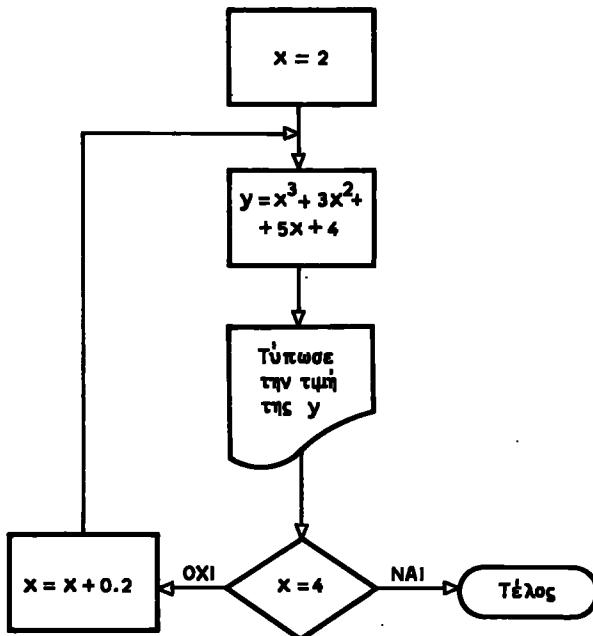
**Παράδειγμα 7.**

Να βρεθεί η τιμή της συναρτήσεως $y = x^2 + 5x$ για $x = 1.2$



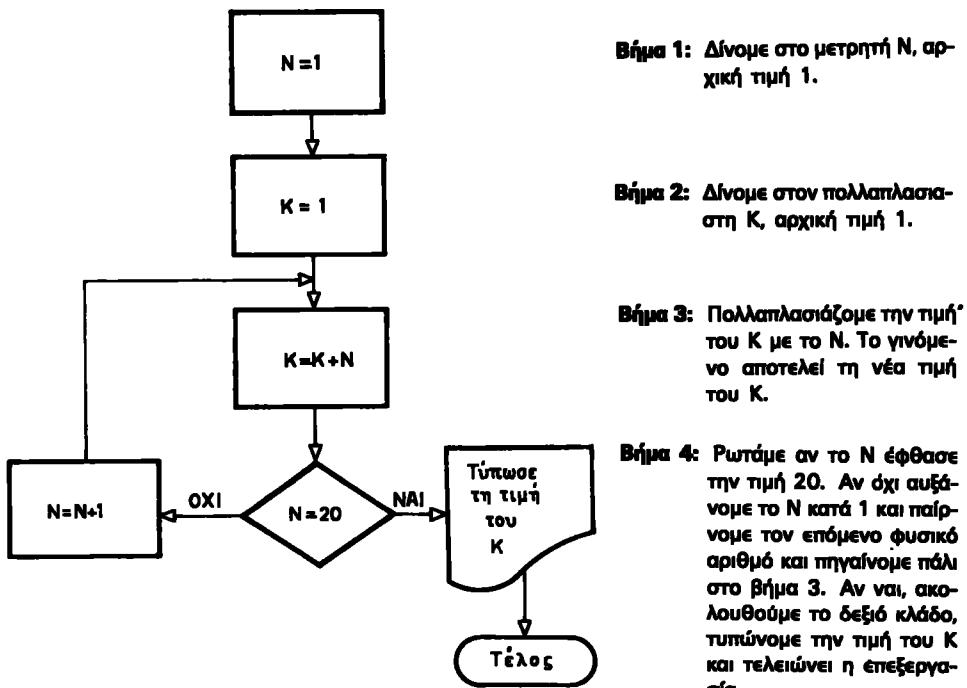
Παράδειγμα 8.

Να σχεδιασθεί διάγραμμα ροής, με το οποίο θα βρίσκονται και θα τυπώνονται οι τιμές της συναρτήσεως $y = x^3 + 3x^2 + 5x + 4$ για $x = 2$ (0.2)4. Ο συμβολισμός αυτός σημαίνει ότι η ανεξάρτητη x παίρνει σαν πρώτη τιμή την $x = 2$ και σαν τελευταία την $x = 4$ με βήμα 0.2. Δεν χρειάζεται και εδώ να διαβάζονται οι τιμές του x αλλά μπορούμε να αποταμιεύσουμε στην περιοχή x την αρχική τιμή 2 την οποία να αυξάνομε κατά 0.2 μετά από κάθε κύκλο.



Παράδειγμα 9.

Να υπολογισθεί το γινόμενο των 20 πρώτων φυσικών αριθμών. Και εδώ, όπως στο παράδειγμα 5, δεν χρειάζεται να διαβάσουμε όλους τους φυσικούς αριθμούς γιατί μπορούμε να τους σχηματίσουμε, δίνοντας σε ένα μετρητή N αρχική τιμή 1 και αυξάνοντας μετά από κάθε κύκλο το περιεχόμενό του κατά 1. Επίσης ορίζουμε ένα πολλαπλασιαστή K , στον οποίο δίνομε αρχική τιμή 1. Σχεδιάζομε έτσι το παρακάτω λογικό διάγραμμα.



6.3 Μερικοί χρήσιμοι μαθηματικοί συμβολισμοί.

a) Αθροίσματα.

Αρκετές φορές θα συναντήσουμε το σύμβολο \sum . Με το σύμβολο αυτό παριστάνομε το άθροισμα των δρων μιας 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.

Με το σύμβολο \prod παριστάνομε το γινόμενο ενός πλήθους ομοίων όρων.
Π.χ.

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

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

$$\prod_{v=1}^8 x^v \text{ ημνx} = (x \text{ ημx}) \cdot (x^2 \text{ ημ}2x) \dots (x^8 \text{ ημ}8x)$$

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

6.4 Η διαδικασία του προγραμματισμού.

Για να δημιουργήσομε ένα πρόγραμμα, που να επιλύει κάποιο πρόβλημα, κάνομε τα εξής διαδικαστικά βήματα:

- Καθορίζομε, αναλύομε και περιγράφομε το πρόβλημα που θέλομε να προγραμματίσουμε.
- Διαλέγομε μια μέθοδο λύσεως.
- Σχεδιάζομε ένα λογικό διάγραμμα, όπου απεικονίζομε τις διάφορες λειτουργίες με τη λογική σειρά, που θα ακολουθηθεί για να επιλυθεί το πρόβλημα.
- Διαλέγομε μια γλώσσα προγραμματισμού και έκφραζόμε τα διάφορα σύμβολα του λογικού διαγράμματος με τις αντίστοιχες εντολές της γλώσσας, που τις γράφομε επάνω σε ειδικά έντυπα (Coding Forms).
- Μεταφέρομε τις γραμμένες εντολές του προγράμματος επάνω σε κάποιο φορέα, π.χ τις διατρυπούμε σε δελτία.
- Διοχετεύομε το πρόγραμμα στον υπολογιστή. Αυτός αναλαμβάνει να το μεταφράσει στη γλώσσα μηχανής με το ειδικό μεταφραστικό πρόγραμμα που

διαθέτει, ενώ συγχρόνως προβαίνει και σε έλεγχο, για το αν υπάρχουν συντακτικά ή ορθογραφικά λάθη στις εντολές που έχουμε γράψει.

- η) Στις περιπτώσεις που δεν υπάρχουν λάθη γλωσσικά, δίνομε δοκιμαστικά δεδομένα (Sample Data) και συγκρίνομε τα αποτελέσματα από τη λειτουργία του προγράμματος, στον υπολογιστή με τα αποτελέσματα από υπολογισμούς που έχουμε κάνει με το χέρι για τα ίδια δεδομένα. Εφ' όσον τα δύο αποτελέσματα συμπίπουν, σημαίνει ότι η λογική, που χρησιμοποιήσαμε για το σχεδιασμό του προγράμματος, ήταν σωστή.
- θ) Το πρόγραμμα μπορεί να λειτουργήσει με οποιαδήποτε δεδομένα.
- ι) Τεκμηριώνομε το πρόγραμμα (Program documentation). Δηλαδή δημιουργούμε ένα φάκελλο,..μέσα στον οποίο γράφομε όλες τις πληροφορίες, που αναφέρονται στο συγκεκριμένο πρόγραμμα, όπως π.χ. τα λογικά διάγραμματα που σχεδιάσαμε, τη μέθοδο επιλύσεως που ακολουθήσαμε, τα δεδομένα που χρησιμοποιήσαμε για test, τα αποτελέσματα που πήραμε, τις διάφορες εκτυπώσεις (listings), κ.ο.κ.

Ασκήσεις.

1. Να σχεδιασθεί λογικό διάγραμμα για τον υπολογισμό του αθροίσματος των τετραγώνων των 30 πρώτων φυσικών αριθμών.
2. Να σχεδιασθεί λογικό διάγραμμα για τον υπολογισμό του γινομένου των 20 πρώτων αρτίων φυσικών αριθμών.
3. Να σχεδιασθεί λογικό διάγραμμα για τον υπολογισμό του αθροίσματος των 10 πρώτων όρων της αικολούθιας $1/2, 1/4, 1/8, 1/16, \dots$
4. Δίνονται 2000 αριθμοί. Ο καθένας έχει διατρυπηθεί σε ένα δελτίο. Να σχεδιασθεί λογικό διάγραμμα για την εύρεση του μικρότερου από αυτούς.
5. Να σχεδιασθεί λογικό διάγραμμα για την επίλυση μιας δευτεροβάθμιας εξισώσεως με ένα άγνωστο $ax^2 + bx + c = 0$, της οποίας οι συντελεστές a, b, c έχουν διατρυπηθεί σε ένα δελτίο. Να προβλεφθεί και μια στοιχειώδης διερεύνηση για τη μορφή των ρίζών.
6. Δίνονται 5 αριθμοί διατρυπημένοι στο ίδιο δελτίο. Να σχεδιασθεί λογικό διάγραμμα, με το οποίο να βρίσκεται και να τυπώνεται ο μέσος όρος τους.
7. Να σχεδιασθεί λογικό διάγραμμα για τον υπολογισμό της τιμής της παραστάσεως:

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

Η τιμή του x θα διαβάζεται στην αρχή από ένα δελτίο.

8. Να σχεδιασθεί λογικό διάγραμμα για τον υπολογισμό της τιμής της παραστάσεως:

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

Οι τιμές των $a_1, x_1, a_2, x_2, \dots$ διαβάζονται από 24 δελτία. Σε κάθε δελτίο υπάρχει μια τιμή a και η αντίστοιχη x .

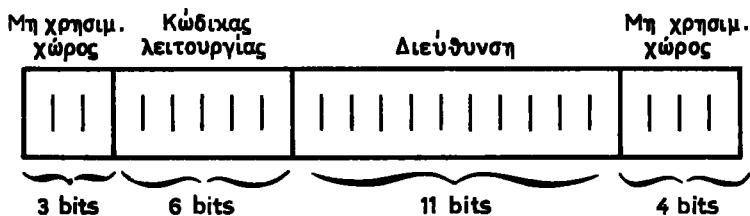
ΚΕΦΑΛΑΙΟ ΕΒΔΟΜΟ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕ ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ

7.1 Ένας υποθετικός εκπαιδευτικός υπολογιστής.

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

Ο υπολογιστής αυτός διαθέτει μνήμη 2048 θέσεων, σε ένα μέρος της οποίας αποθηκεύομε το πρόγραμμα και τα δεδομένα. Οι εντολές είναι τύπου μιας διεύθυνσεως, 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 το 0 ως το 2047, η σε δυαδική μορφή από το 0 0 0 0 0 0 0 0 0 0 0 0 ως το 1 1 1 1 1 1 1 1 1. Χρειάζονται λοιπόν 11 bits για τη δυαδική παράσταση της διεύθυνσεως κάθε θέσεως. Για τους κωδικούς λειτουργίας εντολών χρειάζονται 6 bits. Έτσι η δομή μιας εντολής είναι:



Δεν υπάρχει διάκριση στη θέση μιας μνήμης, δηλαδή μπορεί άλλοτε να περιέχει μια 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), που χωρίζεται σε δύο ίσα μέρη, το βασικό συσσωρευτή (Accumulator) και το βοηθητικό συσσωρευτή (Upper Accumulator), που 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 πράξεως.

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

πολλά ψηφία, έστω με 23 δυαδικά ο καθένας, πολλαπλασιασθούν, το γινόμενό τους θα περιέχει 46 ψηφία και δεν θα χωρά στο βασικό συσσωρευτή. Για το λόγο αυτό διασπάται και το μισό με τα περισσότερο σημαντικά ψηφία τοποθετείται στο βοηθητικό συσσωρευτή, ενώ το υπόλοιπο μισό στο βασικό συσσωρευτή.

Παράδειγμα.

Έστω $11001000 \times 110000, 0101000000$.

Το αποτέλεσμα θα καταχωρισθεί ως εξής:

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

Σημειώνομε ότι σε παρόμοιες περιπτώσεις το πρώτο στοιχείο του βοηθητικού συσσωρευτή και του βασικού συσσωρευτή περιέχει το πρόσημο του αριθμού, που φυσικά είναι το ίδιο.

Η μονάδα ελέγχου περιλαμβάνει ένα καταχωριστή εντολών με μήκος 24 bits, όπου καταχωρίζεται κάθε φορά μια εντολή, που μεταφέρεται από τη μνήμη, και ένα καταχωριστή διευθύνσεων, του οποίου το αριθμητικό περιεχόμενο δείχνει τη διεύθυνση της έπομενης εντολής που πρέπει, λογικά, να εκτελεσθεί. Περιλαμβάνει ακόμα και ένα αποκωδικοποιητή εντολών.

Ο TK1 διαθέτει δύο διαύλους εισόδου και δύο εξόδου. Με τους διαύλους εισόδου συνδέεται με μια περιφερειακή μονάδα αναγνώσεως διατρήτων δελτίων και μια μονάδα αναγνώσεως διάτρητης χαρτοταπινίας, ενώ με τους διαύλους εξόδου συνδέεται με ένα εκτυπωτή και μια μονάδα διατρήσεως δελτίων.

Είναι επίσης δυνατή η αυτόματη μετατροπή από το δεκαδικό στο δυαδικό σύστημα σε όλους τους διαύλους εισόδου και από το δυαδικό στο δεκαδικό σε όλους τους διαύλους εξόδου.

Περιγράφεται στη συνέχεια το σύνολο των εντολών, που μπορεί να δεχθεί ο TK1.

7.2 Το σύνολο περιγραφών των εντολών του TK1.

Δεκαδική παράσταση του κώδικα λειτουργίας	Εντολή	Περιγραφή της λειτουργίας
16	R Ad1	Αντικατάσταση του περιεχόμενου του συσσωρευτή από το περιεχόμενο της διευθύνσεως μνήμης Ad1. Το προηγούμενο περιεχόμενο του συσσωρευτή καταστρέφεται. Το περιεχόμενο της Ad1 διατηρείται.

Δεκαδική παράσταση του κώδικα λειτουργίας	Εντολή	Περιγραφή της λειτουργίας
17	C Ad1	Αντικατάσταση του περιεχομένου της διεύθυνσεως 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 διεύθυνσεως μνήμης Ad1 στο περιεχόμενο του συσσωρευτή. Το άθροισμα τοποθετείται στο συσσωρευτή. Η Ad1 διατηρεί το περιεχόμενό της.
11	S Ad1	Αφαίρεση του περιεχομένου της διεύθυνσεως 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 διεύθυνσεως 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 εντολή, 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 εντολή που βρίσκεται στη διεύθυνση Ad1 της μνήμης.

Δεκαδική παράσταση του κώδικα λειτουργίας	Εντολή	Περιγραφή της λειτουργίας
22	Z Ad1	Έλεγχος για το αν το περιεχόμενο του όμσσωρευτή είναι μηδέν. Αν ναι, τότε συνεχίζεται η εκτέλεση του προγράμματος από την εντολή, που βρίσκεται στη διεύθυνση Ad1. Αν όχι συνεχίζει στην επόμενη εντολή.
15	J Ad1	Άλμα στην εντολή που βρίσκεται στη διεύθυνση Ad1.
30	B —	Μεταφορά του περιεχομένου του βοηθητικού συσσωρευτή στο βασικό συσσωρευτή.
28	E Ad1	Ανταλλαγή του περιεχομένου του συσσωρευτή με το περιεχόμενο της διεύθυνσεως Ad1.
23	N —	Άλλαγή στο πρόστιμο του περιεχομένου του συσσωρευτή.
32	V Ad1	Εγγραφή του περιεχομένου του συσσωρευτή στη διεύθυνση μνήμης Ad1 και μηδενισμός του συσσωρευτή.
40	I Ad1	Ανάγνωση από τη μονάδα εισόδου (αναγνωστική διατρήτων δελτίων) ενός αριθμού και τοποθέτησή του μέσω του συσσωρευτή στη διεύθυνση Ad1.
41	W Ad1	Εγγραφή του περιεχομένου της διεύθυνσεως Ad1 μέσω του συσσωρευτή στη μονάδα εξόδου (εκτυπωτή).
42	T Ad1	Τοποθέτηση της διεύθυνσεως Ad1 στον καταχωριστή διεύθυνσεως.
25	H —	Διακοπή της εκτελέσεως του προγράμματος.

7.3 Πώς λειτουργεί ο TK1.

Όλες οι εντολές ενάς προγράμματος θα πρέπει αρχικά να αποθηκευθούν στή 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 δείχνομε αυτό στο γράψιμο του προγράμματος με ένα χαρακτήρα ή μια λέξη. Εδώ 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 κωδικογραφήσεως.

Οι εντολές αναγράφονται επάνω σε ειδικά σχεδιασμένα έντυπα, τα έντυπα κωδικογραφήσεως. 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.

Στην παράγραφο αυτή θα μάθομε να γράφομε απλά προγράμματα στη γλώσσα μηχανής, στη γλώσσα του TK1.

Ας υποθέσουμε ότι θέλουμε σαν μια 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 καταχωρίσεως εντολής	Εντολή		Επεξήγηση
	Κώδικας λειτουργίας	Διεύθυνση	
M → 16	R	162	Μεταφέρεται το περιεχόμενο της διεύθυνσεως 162 στο συσσωρευτή.
17	C	403	Μεταφέρεται το περιεχόμενο του συσσωρευτή στη διεύθυνση 403 της μνήμης.

Ας δούμε τώρα τι 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 τὸν δεκαδικά εκφρασμένο αριθμό

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

Φάσεις	Περιεχόμενο του καταχωριστή διεύθυνσεων	Περιεχόμενο του καταχωριστή εντολών	Περιεχόμενο του συσσωρευτή	Περιεχόμενο της διεύθυνσεως 162	Περιεχόμενο της διεύθυνσεως 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

Ας δούμε τώρα ποιες εντολές θα χρησιμοποιήσουμε για να προσθέσουμε το c(162) στο c(297) και το άθροισμα να το τοποθετήσουμε στη διεύθυνση 403.

Διεύθυνση καταχωρήσεως της εντολής	Εντολή		Επεξήγηση
	Κωδικός	Διεύθυνση	
M → 16	R	162	Αντικατάσταση του c(Ac) με το c(162).
17	A	297	Πρόσθεση του c(297), στο c(Ac)
18	C	403	Μεταφορά και αποθήκευση του c(Ac) στη διεύθυνση 403.

Στη συνέχεια θα μελετήσουμε την περίπτωση, που θέλομε να αφαιρέσουμε το περιεχόμενο της διεύθυνσεως 297 από το περιεχόμενο της διεύθυνσεως 162 και να αποθηκεύσουμε τη διαφορά στη διεύθυνση 403.

Διεύθυνση καταχωρήσεως της εντολής	Εντολή		Επεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	162	Αντικατάσταση του c(Ac) από το c(162)
17	S	297	Αφαίρεση του c(297) από το c(Ac)
18	C	403	Μεταφορά του c(Ac) στην 403

Αν $c(162) = 4902$ και $c(297) = 6845$, τότε μετά το τέλος της εκτελέσεως των εντολών· θα έχομε $c(403) = (-)1943$. Το αρνητικό πρόστιμο δείχνεται με τη παρουσία του δυαδικού ψηφίου 1 στην ακραία αριστερή θέση της λέξεως της διεύθυνσεως 403.

¹ Από το σημείο αυτό και έπειτα θα χρησιμοποιούμε τους συμβολισμούς για τα περιεχόμενα των θέσεων της μνήμης, που καθορίσαμε στην παράγραφο 5.4 δηλαδή $c(162)$, $c(Ac)$ (για το βασικό συσσωρευτή), $c(Up)$ (για το βοηθητικό συσσωρευτή κ.ο.κ.).

Στη συνέχεια θα πολλαπλασιάσουμε το c (162) με το c (297) και θα τοποθετήσουμε το γινόμενο στη διεύθυνση 403.

Διεύθυνση καταχωρίσεως εντολής	Εντολή		Επεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	162	Αντικατάσταση του c(Ac) από το c(162).
17	X	297	Πολλαπλασιασμός του c(Ac) με το c(297).
18	C	403	Μεταφορά του c(Ac) στην 403.

Αν δε c (162) = 4902 και c (297) = 45, τότε με την εκτέλεση των παραπάνω εντολών θα έχομε c (403) = $4902 \times 45 = 220590$. Από την άλλη μεριά αν c (297) = 6845, τότε c (403) = $4902 \times 6845 = 33554190$. Επειδή ο αριθμός αυτός είναι πολύ μεγάλος, για να χωρέσει στο βασικό συσσωρευτή, γ' αυτό ένα μέρος του διαρρέει στο βοηθητικό συσσωρευτή. Έτσι θα έχομε την κατάσταση:

Αριθμός

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

Καταχωριστές

Βοηθητικός συσσωρευτής

0 000, 000000000, 0000000011

(+) 25 165, 824

Βασικός συσσωρευτής

0 111, 111111111, 1100001110

(+) 8, 388, 366

Ας δούμε κάποιο άλλο πρόγραμμα, όπου θέλομε να πολλαπλασιάσουμε το c (162) με το c (297) και το γινόμενο να το διαιρέσουμε με το c (201).

Διεύθυνση καταχωρίσεως της εντολής	Εντολή		Επεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	162	To c(162) μεταφέρεται στο συσσωρευτή.
17	X	297	Πολλαπλασιάζεται το c(Ac) με το c(297)
18	D	201	και το αποτέλεσμα τίθεται στο συσσωρευτή.
19	C	403	Διαιρείται το c(Ac) με το c(201). Το αποτέλεσμα τίθεται στο συσσωρευτή.
			Μεταφέρεται το c(Ac) στην θέση 403.

Δηλαδή πολλαπλασιάζεται το περιεχόμενο της διευθύνσεως 297 με το της 162, το γινόμενο καταχωρίζεται στο συσσωρευτή και κατόπιν διαιρείται το περιεχόμενο του συσσωρευτή με το περιεχόμενο της 201 και το αποτέλεσμα τοποθετείται στην 403.

7.6 Εντολές αποφάσεων.

Μια από τις πιο σπουδαίες εργασίες που εκτελεί ο υπολογιστής, είναι η λήψη αποφάσεων. Ας δούμε μια απλή πλέρη πτώση.

Έστω ότι θέλομε να αφαιρέσουμε το μικρότερο από δύο αριθμούς, που βρίσκονται στις θέσεις 162 και 297 από το μεγαλύτερο και να τοποθετήσουμε το αποτέλεσμα στην 403. Δεν γνωρίζομε βέβαια από πριν σε ποια θέση είναι ο μεγαλύτερος. Θα χρειασθούν 5 εντολές, οι εξής:

Διεύθυνση καταχωρίσεως της εντολής	Εντολή		'Επεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	162	Μεταφέρεται το c(162) στο συσσωρευτή.
17	S	297	Αφαιρείται το c(297) από το c(Ac). Η διαφορά τοποθετείται στο συσσωρευτή.
18	P	21	Ελέγχεται το πρόσημο του συσσωρευτή. Αν το c(Ac) είναι θετικό, γίνεται δῆμα στην εντολή, που βρίσκεται στη διεύθυνση 21. Διαφορετικά εκτελείται η επόμενη εντολή.
19	R	297	Ανπικατάσταση του c(Ac) από το c(297).
20	S	162	Αφαιρείται το c(162) από το c(Ac)
18 → 21	C	403	Μεταφέρεται το c(Ac) στην 403.

Οι δύο εναλλακτικοί δρόμοι, που ανοίγονται στην εντολή 18, ονομάζονται *κλάδοι* και η εντολή 18, *εντολή διακλαδώσεως*.

Αν ζητάμε αντίστροφα να αφαιρέσουμε το μεγαλύτερο από το μικρότερο για τους ίδιους δύο αριθμούς, θα πρέπει να τροποποιήσουμε ελαφρά το προηγούμενο πρόγραμμα. Έτσι θα έχουμε τις εξής εντολές:

Διεύθυνση καταχωρίσεως της εντολής	Εντολή		'Επεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	162	Μεταφέρεται το c(162) στο συσσωρευτή.
17	S	297	Αφαιρείται το c(297) από το c(Ac).
18	M	21	Αν το c(Ac) είναι αρνητικό, τότε πηγαίνει στην εντολή 21. Διαφορετικά συνεχίζει στην 19.
19	R	297	Μεταφέρεται το c(297) στο συσσωρευτή.
20	S	162	Αφαιρείται το c(162) από το c(Ac).
18 → 21	C	403	Μεταφέρεται το c(Ac) στη 403.

Ας δούμε ένα ακόμα παράδειγμα. Έστω ότι θέλομε να ελέγξουμε το περιεχόμενο της διευθύνσεως 403 και αν αυτό είναι μηδέν να το αφήσουμε αμετάβλητο, διαφορετικά να μεταφέρουμε το περιεχόμενο της διευθύνσεως 162.

Διεύθυνση καταχωρίσεως της εντολής	Εντολή		'Επεξήγηση
	Κώδικας	Διεύθυνση	
M → 16 17	R Z	403 19	Μεταφέρεται το c(403) στο συσσωρευτή. Αν το c(Ac) είναι μηδέν, τότε πηγαίνει στην εντολή 19. Διαφορετικά συνεχίζει στην 18.
18 17 → 19	R C	162 403	Μεταφέρεται το c(162) στο συσσωρευτή. Μεταφέρεται το c(Ac) στη διεύθυνση 403.

7.7 Τρόπος χρησιμοποίησεως μερικών ακόμα εντολών.

Στο σημείο αυτό θα δούμε πώς χρησιμοποιούνται σε ένα πρόγραμμα 5 ακόμη εντολές, οι B, I, V, W και H.

Με την εντολή B χειρίζομαστε το περιεχόμενο του βοηθητικού συσσωρευτή. Έστω ότι είναι γνωστό ότι κάποιο σύνολο, που θέλουμε να βρούμε, είναι τόσο μεγάλο, ώστε να χρειάζεται 2 θέσεις π.χ. 403 και 404 για να αποθηκευθεί (το μικρότερο τμήμα στην 404 και το πιο σημαντικό στην 403). Με το παρακάτω πρόγραμμα παρατίθεται ο αριθμός που βρίσκεται στη θέση 162 στο «διπλό αυτό αριθμό».

Διεύθυνση καταχωρίσεως της εντολής	Εντολή		'Επεξήγηση
	Κώδικας	Διεύθυνση	
M → 16	R	404.	Αντικαθίσταται το c(Ac) από το c(404).
17	A	162	Προστίθεται το c(162) στο c(Ac).
18	C	404	Εγγράφεται το c(Ac) στη θέση 404.
19	B	—	Μεταφέρεται το c(Up) στο συσσωρευτή.
20	A	403	Προστίθεται το c(403) στο c(Ac).
21	C	403	Εγγράφεται το c(Ac) στη θέση 403.

Όπως ήδη έχουμε πει, υπάρχουν 4 ανεξάρτητοι δίσιλοι εισόδου - εξόδου συνδεδεμένοι με τον TK1. Έστω λοιπόν ότι θέλουμε να διαβάσουμε 1000 αριθμούς από τη μονάδα αναγνώσεως δελτίων και να τοποθετήσουμε το άθροισμά τους στη διεύθυνση 4. Το πρόγραμμα θα έχει την παρακάτω μορφή.

Διεύθυνση καταχωρίσεως της εντολής	Εντολή		'Επεξήγηση
	Κώδικας	Διεύθυνση	
M → 19	I	6	Διαβάζεται ένας αριθμός-δεδομένο και τοποθετείται στη θέση 6.
20	A	6	Προστίθεται το c(6) στο c(Ac) που είναι αρχικά μηδέν.
21	C	4	Μεταφέρεται το άθροισμα [c(Ac)] στη θέση 4. Ο συσσωρευτής διατηρεί το περιεχόμενό του ξαναπηγαίνει στην εντολή 19
19 ← 22	J	19	

Στο πρόγραμμα αυτό δύναται μια 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 επιτύχομε 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.

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

Διεύθυνση καταχωρίσεως της εντολής	Εντολή		Επεξήγηση
	Κώδικας	Διεύθυνση	
16	V	6	c(Ac) = 0, c(6) = 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. Προστίθεται το c(6) στο συασωρευτή.
21	A	6	Μεταφέρεται το c(Ac) στη θέση 4.
22	C	4	Αντικαθίσταται το c(Ac) με το c(5).
23	R	5	Προστίθεται το c(32) στο c(Ac).
24	A	32	Μεταφέρεται το c(32) στη θέση 5.
25	C	5	Αφαιρέται το c(33) από το c(Ac).
26	S	33	Ελέγχεται το c(Ac). Αν είναι μηδέν εκτελείται στη συνέχεια η εντολή που βρίσκεται στη διεύθυνση 30. Διαφορετικά η επόμενη (28).
27	Z	30	Αντικαθίσταται το c(Ac) με το c(4). Γίνεται άλμα στην εντολή, που βρίσκεται στη διεύθυνση 20.
28	R	4	Εγγράφεται στον εκτυπωτή το c(4).
20 ← 29	J	20	Τέλος της λειτουργίας.
30	W	4	Η διεύθυνση 32 περιέχει τον αριθμό 1.
31	H	—	Η διεύθυνση 33 περιέχει τον αριθμό 1000.
32	—	1	
33	—	1000	

Άσκησεις.

- 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. $(c(15) + c(21)) c(30) - c(21)$
- 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. Τοποθετείστε το τελικό αποτέλεσμα στη διεύθυνση 10.
- Να δημιουργήσετε μια δίκιη σας κωδικοποίηση για ορισμένες 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.

Μία 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 τον κώδικα εντολής (Operation Code) όσο και για τις διεύθυνσεις (Operands). Όσον αφορά στις τελευταίες, η χρήση των συμβολικών μνημονικών χαρακτηρών κατήργησε τη χρησιμοποίηση των απολύτων διευθύνσεων των θέσεων της μνήμης (δηλαδή των αριθμητικών διευθύνσεων, όπως έχουν δοθεί αρχικά με την πρώτη λειτουργία του υπολογιστή και όπου κάθε διεύθυνση ορίζει μονοσήμαντα μια θέση μνήμης). Σε αυτή την περίπτωση χαρακτηρίζουμε μια θέση μνήμης με ένα συμβολικό όνομα και αναφέρομαστε 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 συμβαλίσουμε με κάποιαν άλλη ονομασία. Ο υπολογιστής αναλαμβάνει να καταχωρίσει την εντολή ενός προγράμματος, με τρόπο συνεχή σε μια περιοχή της μνήμης και η αντιτοπίζει απολύτων και συμβολικών διευθύνσεων γίνεται έτσι, ώστε να τηρείται η δομένη λογική σειρά στην εκτέλεση των εντολών.

Επειδή, όπως είπαμε, η μηχανή αντιλαμβάνεται μόνο τη δική της γλώσσα, ένα *Assembly* πρόγραμμα πρέπει να μεταφρασθεί 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 (*Assembler*) στη γλώσσα μηχανής (*Object program*).
- Μετά τη μετάφραση εκτυπώνονται οι εντολές του *Assembly* προγράμματός μας (*Listing*), όπου έχουν επισημανθεί συντακτικά ή ορθογραφικά λάθη, που βρέθηκαν στο στάδιο της μεταφράσεως. Δίπλα στην κάθε εντολή *Assembly* αναγράφεται και η αντίστοιχη εντολή της μηχανής.

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

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

PROGRAM	PROGRAMMER			PAGE OF	CARD ELECTRO NUMBER
		PUNCHING INSTRUCTIONS	GRAPHIC PUNCH	IDENTIFICATION SEQUENCE	
				73	
				71	
				70	
				69	
				68	
				67	
				66	
				65	
				64	
				63	
				62	
				61	
				60	
				59	
				58	
				57	
				56	
				55	
				54	
				53	
				52	
				51	
				50	
				49	
				48	
				47	
				46	
				45	
				44	
				43	
				42	
				41	
				40	
				39	
				38	
				37	
				36	
				35	
				34	
				33	
				32	
				31	
				30	
				29	
				28	
				27	
				26	
				25	
				24	
				23	
				22	
				21	
				20	
				19	
				18	
				17	
				16	
				15	
				14	
				13	
				12	
				11	
				10	
				9	
				8	
				7	
				6	
				5	
				4	
				3	
				2	
				1	
				.	

Σχ. 8.3.
Το έγγραφο κωδικογράφησεως Assembly:

σπήλες. Σε κάθε γραμμή περιγράφεται μία μόνο εντολή.

Στις στήλες 1-8 (Name) δίνεται μία ονομασία, που είναι η συμβολική διεύθυνση μιας εντολής. Δεν χρειάζεται να δίνουμε συμβολική διεύθυνση σε κάθε εντολή. Παρέχομε μόνο στις εντολές εκείνες, όπου μας παραπέμπει το πρόγραμμα μετά τη εκτέλεση μιας εντολής διακλαδώσεως. Η συμβολική ονομασία πρέπει να αρχίζει με αλφαριθμητικό χαρακτήρα στη στήλη 1. Η στήλη 4 αφήνεται κενή.

Στις στήλες 10-14 (Operation) γράφεται ο συμβολικός κώδικας λειτουργίας της εντολής. Η στήλη 15 μένει κενή.

Από τη στήλη 16 γράφομε τις συμβολικές διευθύνσεις (Operands) των όρων της περιγραφόμενης εντολής.

Σχόλια, παρατηρήσεις ή επεξηγήσεις (Comments) για τη λειτουργία μιας εντολής ή ενός τμήματος του προγράμματος μπορούμε να γράφομε σε οποιοδήποτε γραμμή εντολής μετά την περιγραφή των συμβολικών διευθύνσεων, αφήνοντας μία τουλάχιστον κενή στήλη μετά από αυτές. Μπορούμε όμως να χρησιμοποιήσουμε και μια ολόκληρη γραμμή τού έντυπου θέτοντας απλώς στην 1η στήλη της, τον αστερίσκο (*¹). Η στήλη 72 όταν περιέχει οποιοδήποτε χαρακτήρα σημαίνει συνέχιση στην επόμενη γραμμή από τη στήλη 16.

Στις στήλες 73 - 80 (Identification Sequence) γράφεται η κωδική ονομασία του προγράμματος (προδιαγραφή που αγνοείται στο στάδιο της μεταφράσεως), μαζί με την αύξουσα αριθμητη των εντολών.

8.4 Γενική περιγραφή του υπολογιστή που θα χρησιμοποιήσουμε.

Ο υπολογιστής για τον οποίο θα μιλάμε στα επόμενα, έχει μνήμη που αποτελείται από λέξεις των 32 bits. Κάθε λέξη χωρίζεται σε δύο τμήματα (Halfwords) των 16 bits το καθένα. Κάθε μισή λέξη αποτελείται από 2 bytes των 8 bits. Μέσα σε κάθε byte μπορεί να καταχωρισθεί ένας χαρακτήρας κωδικοποιημένος στο σύστημα EBCDIC.

Στην κεντρική μονάδα υπάρχουν 16 καταχωριστές (Registers) με μήκος 32 bits ο καθένας. Η χρήση καταχωριστών στη γλώσσα Assembly αποτελεί τη βάση της δομής της.

8.5 Κατηγορίες εντολών Assembly.

Θα περιγράψουμε στα επόμενα τη γλώσσα Assembly που χρησιμοποιείται σε αρκετούς υπολογιστές.

Οι εντολές της γλώσσας αυτής χωρίζονται σε 6 κατηγορίες:

a) **Εντολές τόπου RR.**

Αναφέρονται σε πράξεις που γίνονται ανάμεσα στα περιεχόμενα δύο καταχωριστών (Register and Register).

b) **Εντολές τόπου RX.**

Αναφέρονται σε πράξεις που γίνονται ανάμεσα στα περιεχόμενα ενός καταχωριστή και μιας θέσεως στη μνήμη (Register and Storage).

γ) **Εντολές τόπου RS.**

Το ίδιο όπως και οι RX (διαφέρει η δομή τους).

δ) **Εντολές τόπου SI.**

Για πράξεις ανάμεσα στο περιεχόμενο μιας θέσεως της μνήμης και μιας σταθεράς.

ε) **Εντολές τόπου SS1.**

Αναφέρονται σε πράξεις ανάμεσα στα περιεχόμενα δύο ισομήκων περιοχών της μνήμης.

στ) **Εντολές τόπου SS2.**

Αναφέρονται σε πράξεις ανάμεσα στα περιεχόμενα δύο ανισομήκων περιοχών της μνήμης.

¹ Ο αστερίσκος στη στήλη 1 προειδοποιεί τον υπολογιστή να μη θεωρήσει το περιεχόμενο της γραμμής σαν εντολή.

8.6 Εντολές τύπου RR.

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

Κώδικας λειτουργίας της εντολής	R ₁	R ₂
bits 0 7 8 11 12 15		

Δηλαδή ο κώδικας λειτουργίας της εντολής καταχωρίζεται στο πρώτο byte της (0-7 bits) και στο δεύτερο byte καταχωρίζονται δύο αριθμοί που συμβολίζουν δύο καταχωριστές R₁ και R₂¹.

Στην κατηγορία αυτή ανήκουν οι εντολές:

a) **Εντολή προσθέσεως (AR).**

Η μορφή της είναι:

AR R₁, R₂

Με τη βαθιθεία της προστίθεται το περιεχόμενο του καταχωριστή R₂ στο περιεχόμενο του καταχωριστή R₁. Το δέροισμα τοποθετείται στον R₁. Το περιεχόμενο του R₂ δεν αλλάζει π.χ. AR 11,14 σημαίνει ότι θα προστεθεί το περιεχόμενο του καταχωριστή 14 στο περιεχόμενο του καταχωριστή 11, και το δέροισμα θα καταχωρισθεί στο 11. Δηλαδή [11] + [14] = [11].

b) **Εντολή αφαίρέσεως (SR).**

Η μορφή της είναι:

SR R₁, R₂

Με την εντολή αυτή αφαιρείται το περιεχόμενο του καταχωριστή R₂ από το περιεχόμενο του καταχωριστή R₁. Η διαφορά καταχωρείται στον R₁. Το περιεχόμενο του R₂ δεν αλλάζει. Π.χ.

SR 8, 2

Η εκτέλεση της εντολής 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.

Πριν προχωρήσουμε στη περιγραφή των επομένων εντολών, θα πούμε δυο λόγια για μια σημαντική προδιαγραφή της Assembly, τον δείκτη συνθήκης.

Οι αποφάσεις και οι διακλαδώσεις αποτελούν σημαντικές ενέργειες σε ένα πρόγραμμα, ο δε τρόπος με τον οποίο μπορούν να προσδιορισθούν με ανάλογες εντολές ένδιαφέρουν άμεσα τον πρόγραμμαστισμό.

Στη γλώσσα Assembly, που εξετάζουμε, η λειτουργία των εντολών αυτών βασίζεται στη χρησιμοποίηση του δείκτη συνθήκης. Αυτός αναφέρεται σε πολλές εντολές και η τιμή του ελέγχεται από μια εντολή διακλαδώσεως υπό συνθήκη (Branch on condition).

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

Π.χ. μετά την εκτέλεση μιας εντολής προσθέσεως, ο δείκτης συνθήκης μπορεί να δείχνει κατά πόσο το δέροισμα είναι θετικό, αρνητικό ή μηδέν. Ακόμη, μετά την εκτέλεση μιας εντολής συγκρίσεως, ο δείκτης συνθήκης δείχνει κατά πόσο ο πρώτος όρος είναι μεγαλύτερος από, ή ίσος προς, ή μικρότερος από το δεύτερο όρο. Ο δείκτης συνθήκης μπορεί να πάρει τέσσερις τιμές 0, 1, 2 και 3.

Αφού ο δείκτης συνθήκης ενεργοποιηθεί κατάλληλα παίρνοντας μια από τις 4 τιμές που είπαμε, η κατέσταση, στην οποία βρίσκεται μπορεί να ελεγχθεί με τη χρησιμοποίηση στη συνέχεια μιας εντολής διακλαδώσεως με συνθήκη π.χ. την εντολή RC που θα δούμε στα επόμενα. Στην εντολή αυτή ο

1 Οι καταχωριστές που διαφέρεται ο υπολογιστής αριθμούνται απόλυτα από το 1-16.

ρίζονται τά 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 (οπότε 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 υπ' αριθ. 8-11 bits της εντολής είναι 1010, θα γίνει η διακλάδωση. Αν όμως η τιμή αυτών των bits είναι 1000, δεν θα γίνει διακλάδωση.

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

ΠΙΝΑΚΑΣ 8.6.1.

Μορφή των bits 8-11 της εντολής διακλαδώσεως	Δεκαδική τιμή	Ελεγχόμενη τιμή τού δείκτη	Ελεγχόμενη συνθήκη μεταξύ δύο όρων A και B
0000	0	Καμμία	
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 ≠ B
0111	7	1 ή 2 ή 3	
1000	8	0	A = B
1001	9	0 ή 3	
1010	10	0 ή 2	A > B
1011	11	0 ή 2 ή 3	
1100	12	0 ή 1	A ≤ 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 δινευ όρων

Μετά από την σύντομη αυτή παρεμβολή, ξαναγυρίζομε στην περιγραφή των εντολών τύπου RR.

γ) Εντολή συγκρίσεως (CR).

Η μορφή της είναι:

CR R₁, R₂

Με την εντολή αυτή συγκρίνονται αλγεβρικά, τα περιεχόμενα των καταχωριστών R₁ και R₂. Από το αποτέλεσμα της συγκρίσεως ο δείκτης συνθήκης παίρνει ανάλογη τιμή (0 αν [R₁] = [R₂]¹, 1 αν

¹ Θυμίζομε από ο συμβολισμός [R₂] σημαίνει το περιεχόμενο του καταχωριστή R₂.

$[R_1] < [R_2]$ και 2 αν $[R_1] > [R_2]$). Η εντολή αυτή συνδύζεται 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 θα πάρει τιμή 0, αν $[6] < [7]$, θα πάρει τιμή 1 και αν $[6] > [7]$ θα πάρει τιμή 2.

ε) Εντολή διακλιδώσεως υπό συνθήκη (BCR).

Έχει τη μορφή:

BCR M, R₃

Προκαλεί διακλάδωση στη διεύθυνση που περιέχεται στον καταχωριστή R₃, όταν ικανοποιείται η συνθήκη που περιγράφει ο M, που είναι ο αριθμός - τιμή του δείκτη συνθήκης. Διαφορετικά εκτελείται η εντολή που ακολουθεί την BCR.

Π.χ.:

CR 4, 5

BCR 8,10

Με την πρώτη από τις δύο αυτές εντολές συγκρίνομε το [4] με το [5]. Με τη δεύτερη δηλώνομε ότι σε περίπτωση που [4] = [5] να εκτελεσθεί σαν επόμενη, η εντολή που βρίσκεται στη διεύθυνση που ιστούαι με το περιεχόμενο του καταχωριστή 10.

ε) Εντολή διακλιδώσεως και συνδόσεως (BALR).

Έχει τη μορφή:

BALR R₁, R₂

Η εντολή αυτή λειτουργεί ως εξής: Τοποθετείται στον καταχωριστή R₁ η διεύθυνση της επόμενης, σε φυσική σειρά εντολής και γίνεται διακλάδωση στην εντολή με διεύθυνση το [R₂]. Αν το R₂ είναι 0, δεν γίνεται διακλάδωση.

Π.χ.:

BALR 8, 9

SUM AR 1, 2

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

δ) Εντολή μεταφοράς (LR).

Η εντολή αυτή έχει τη μορφή:

LR R₁, R₂

και με τη βοήθειά της, μεταφέρεται το περιεχόμενο του R₂ στον R₁, ενώ ο R₂ εξακολουθεί να διατηρεί την τιμή του. Π.χ.:

LR 3, 9

μεταφέρεται το [9] στον καταχωριστή 3. Ο καταχωριστής 9 διατηρεί τη τιμή του.

ε) Εντολή λογιαξής συγκρίσεως (CLR).

Έχει τη 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.

ε) Εντολή μετατροπής αριθμητικών περιεχομένων σε θετικά (LPR).

Έχει τη μορφή:

LPR R_1, R_2

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

δ) Εντολή (SVC).

Έχει τη μορφή:

SVC, I

όπου I ακέραιος. Είναι συνήθως η τελευταία εντολή ενός προγράμματος. Με αυτή ο έλεγχος του προγράμματος επιστρέφεται στο εποπτεύον πρόγραμμα (Supervisor).

8.7 Εντολές τύπου RX.

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

Kώδικας λειτουργίας της εντολής	R_1	X_2	B_2	D_2	
bits 0 7 8 11 12 15 16 19 20 31					

Στα 8 πρώτα bits (0-7) καταχωρίζεται ο κώδικας λειτουργίας της εντολής. Στα επόμενα 4 bits (8 – 11) καταχωρίζεται ο αριθμός του καταχωριστή, που αποτελεί τον πρώτο όρο μιας 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 $D_2 + [X_2] + [B_2]$.

ε) Εντολή 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 εντολή αυτή προστίθεται το περιεχόμενο της διεύθυνσεως POS στο περιεχόμενο του καταχωριστή 10.

ε) Εντολή συγκρίσεως (C).

Έχει τη μορφή:

C $R_1, D_2 (X_2, B_2)$

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

C 8, TIMH

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

γ) Εντολή διακλάδωσεως με συνθήκη (BC).

Έχει τη μορφή:

BC M₁, D₂ (X₂, B₂)

και μοιάζει με την εντολή BCR. Π.χ.:

BC 4, FOROS

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

δ) Εντολή μεταφοράς (L).

Η μορφή της είναι:

L R₁, D₂ (X₂, B₂)

και είναι ανάλογη προς την LR Π.χ.:

L 6, TOKOS

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

ε) Εντολή μεταφοράς στη μνήμη (ST).

Η μορφή της είναι:

ST R₁, D₂ (X₂, B₂)

Με αυτή μεταφέρεται το περιεχόμενο του καταχωριστή R₁ στη θέση της μνήμης με συμβολική διεύθυνση D₂ (X₂, B₂): Π.χ.

ST 4, ELP

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

σ) Εντολή αφαιρέσεως (S).

Έχει τη μορφή:

S R₁, D₂ (X₂, B₂)

Με αυτήν αφαιρέται το περιεχόμενο της μνήμης με διεύθυνση D₂ (X₂, B₂) Π.χ.:

S 8, MEG

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

8.8 Εντολές τύπου RS.

Κάθε εντολή του τύπου αυτού καταλαμβάνει χώρο 32 bits και καταχωρίζεται ως εξής:

Kώδικας, λειτουργίας της εντολής	I ₁	B ₁	D ₁	
bits 0	7 8	15 16	19 20	31

Στα 8 πρώτα bits καταχωρίζεται ο κώδικας λειτουργίας της εντολής, R₁, και R₃ είναι δύο καταχωρίστες. Στη λειτουργία της εντολής μετέχουν με τα περιεχόμενά τους όλοι οι καταχωριστές, από τον R₁, ως τον R₂, καθώς και η διεύθυνση μνήμης [B₂] + D₂.

α) Εντολή πολλαπλής ταυτόχρονης καταχωρίσεως (LM).

Η μορφή της είναι:

LM R₁, R₃, D₂ (B₂)

Με την εντολή αυτή μεταφέρονται και καταχωρίζονται στους καταχωριστές από τον R₁ μέχρι τον R₃ (και όλους τους ενδιάμεσους), τα περιεχόμενα ισαριθμών θέσεων της μνήμης αρχίζοντας από τη διεύθυνση D₂(B₂) και πέρα. Π.χ.:

LM 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 διευθύνσεως 1080 θα τοποθετηθεί στον καταχωριστή 4, της 1081 στον 5 κ.ο.κ.

β) Εντολή πολλαπλής αποθήκευσης στη μνήμη (STM).

Η μορφή της είναι:

STM R₁, R₃, D₂ (B₂)

και είναι αντίστροφη της LM, δηλαδή μεταφέρονται τα περιεχόμενα των καταχωριστών R1 έως και R₃ σε διαδοχικές θέσεις της μνήμης, αρχίζοντας από τη διεύθυνση D₂ (B₂). Π.χ.:

STM 4,7 AMAR

μεταφέρεται το περιεχόμενο των καταχωριστών 4,5,6, και 7 διαδοχική στις 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 Εντολές τύπου SI.

Μια εντολή της κατηγορίας αυτής καπαλαμβάνει στη μνήμη χώρο 32 bits και καταχωρίζεται ως εξής:

Κώδικας, λειτουργίας της εντολής	R ₁	R ₃	B ₂	D ₂	
bits 0	7 8	11 12	15 16	19 20	31

Όπου I₁ είναι μια αριθμητική σταθερή. Στη λειτουργία της εντολής συμμετέχει και το περιεχόμενο της θέσεως της μνήμης με διεύθυνση D₁ + [B₁].

α) Εντολή προσθέσεως σταθερής (AI).

Έχει τη μορφή:

AI D₁ (B₁), I₁

με αυτή, η σταθερή I₁ (μήκους 1 byte) προστίθεται στο περιεχόμενο της ημιλέξεως με διεύθυνση D₁ + [B₁] Π.χ.:

AI 0(6), 40

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

β) Εντολή συγκρίσεως (CLI).

Έχει τη μορφή:

CLI D₁ (B₁), I₁

και γίνεται σύγκριση ανάμεσα στο περιεχόμενο της διευθύνσεως D₁ + [B₁] της μνήμης και της σταθερης I₁. Π.χ.:

CLI POSON, 86

συγκρίνεται το περιεχόμενο της διευθύνσεως POSON με τη σταθερή 86.

γ) Εντολή διακοπής εκτέλεσης του προγράμματος (H).

Έχει τη μορφή:

H D₁ (B₁)

και με αυτήν προκαλείται διακοπή στην εκτέλεση ενός προγράμματος. Αν ο χειριστής πέσει το πλήκτρο START στον πίνακα ελέγχου τότε θα συνεχισθεί η εκτέλεση του προγράμματος από την εντολή που βρίσκεται στη διεύθυνση D₁ + [B₁] της μνήμης.

δ) Εντολή μεταφοράς σταθερής (MVI).

Η μορφή της είναι:

MVI D₁ (B₁), I₁

και με αυτήν μεταφέρεται ο αριθμός I₁ στη θέση που ορίζεται από τη διεύθυνση D₁ + [B₁]. Π.χ.

MVI BAT, 28

μεταφέρεται ο αριθμός 28 στη θέση BAT της μνήμης.

8.10 Εντολές τύπου SS1.

Κάθε εντολή της κατηγορίας αυτής καταλαμβάνει χώρο 1.5 λέξεως (6 bytes) και καταχωρίζεται ως εξής:

Κώδικας λειτουργίας της εντολής	L	B ₁	D ₁	B ₂	D ₂
bits 0	7 8	15 16	19 20	31 32	35 36
					47

Στη λεπτουργία της εντολής συμμετέχουν τα περιεχόμενα των ισομήκων θέσεων της μνήμης με διεύθυνσεις D₁ + [B₁] και D₂ + [B₂]. Το L παριστάνει το κοινό μέγεθος σε bytes.

ε) Εντολή συγκρίσεως (CLC).

Έχει τη μορφή:

CLC D₁ (L,B₁), D₂ (L,B₂)

Με αυτήν συγκρίνονται τα περιεχόμενα των θέσεων της μνήμης με διεύθυνσεις D₁ (B₁) και D₂ (B₂). Π.χ.:

CLC ROM, CIP

συγκρίνονται τα περιεχόμενα των διεύθυνσεων ROM και CIP της μνήμης.

η) Εντολή μεταφοράς (MVC).

Έχει τη μορφή:

MVC D₁ (L, B₁), D₂ (L, B₂)

Με αυτήν, μεταφέρεται το περιεχόμενο της περιοχής με διεύθυνση D₂ (B₂) στην περιοχή με διεύθυνση D₁ (B₁). Π.χ.:

MVC ARAG (30), KATAR

μεταφέρονται 30 bytes από την περιοχή KATAR στην περιοχή ARAG. Η περιοχή ARAG πρέπει, ως εκ τούτου, να έχει μέγεθος 30 bytes.

8.11 Εντολές τύπου SS2.

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

8.12 Ψευδοεντολές (Pseudoinstructions).

Οι εντολές αυτές είναι βαθητικές και χρησιμοποιούνται για να δώσουν στον υπολογιστή τις απαραίτητες γενικές οδηγίες για τη λειτουργία ενός προγράμματος.

Τις κυριότερες από αυτές θα δούμε αμέσως.

α) Εντολή DC.

Με την εντολή 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 a_1, a_2

όπου a_1, a_2 προδιαγραφές που θα δούμε αμέσως.

Με την προδιαγραφή a_1 , περιγράφουμε τον τύπο της πληροφορίας που πρόκειται να καταχωρισθεί σε μια 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 αριστερά 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 σταν ο αριθμός καταχωρίζεται σε περιοχή μήκους 4 bytes και H σταν η καταχώριση γίνεται σε χώρο 2 bytes).

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

Παραδείγματα.

a) 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 δικτεθούν 2 bytes.

γ) ALPHA DC C 'A'

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

β) Εντολή DS.

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

α) DEC DS F

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

β) DIM DS H

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

γ) Εντολή START.

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

START n

όπου: η δεκαδικά εκφρασμένη διεύθυνση της μνήμης της πρώτης εντολής του προγράμματος. Π.χ.:
START 256

σημαίνει ότι η καταχώριση του προγράμματος στη μνήμη, θα αρχίσει από τη διεύθυνση (256)₁₀ ή (100)₁₆.

d) Εντολή 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 καταχωρίζομε σ' αυτόν σαν περιεχόμενο. Π.χ. USING 200,8 Σημαίνει ότι σαν καταχωριστή-βάση θα χρησιμοποιήσουμε τον R₈ καταχωρίζομε δε σ' αυτόν την τιμή 200.

USING *,15 Σημαίνει ότι σαν καταχωριστή-βάση θα χρησιμοποιήσουμε τον καταχωριστή 15 και ότι θα καταχωρίσουμε σ' αυτόν τα περιεχόμενο του καταχωριστή διευθύνσεων.

e) Εντολή END.

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

END D₁

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

END BEGIN

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

f) Εντολή 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 γράφομε ανάμεσα σε αποστρόφους, μπορεί δε να αποτελείται από πολλούς χαρακτήρες (θεωρητικά μέχρι 100). Π.χ.:

TITLE 'EXAMPLE ONE'

Εδώ 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 σε προγράμματα επεξεργασίας αρχείων. Με τον όρο επεξεργασία αρχείου περιλαμβάνονται λειτουργίες όπως:

- α) Άνοιγμα ενός αρχείου, δηλαδή η κατάλληλη προετοιμασία του, ώστε να μπορεί να χρησιμοποιηθεί από το πρόγραμμά μας για να διαβασθούν ή να γραφούν πληροφορίες σ' αυτό.
- β) Ανάγνωση πληροφοριών από ένα αρχείο.
- γ) Εγγραφή πληροφοριών σε ένα αρχείο.
- δ) Κλείσιμο ενός αρχείου, δηλαδή επαναφορά στην αρχική του κατάσταση (στην κατάσταση που βρισκόταν πριν αρχίσει να χρησιμοποιείται στο πρόγραμμα).

Με τον όρο μακροεντολή εννοούμε μια προηγμένη εντολή που κατά τη φάση της μεταφράσεως θα αναλυθεί σε περισσότερες από μία εντολές γλώσσας μηχανής.

Ο καταχωριστής – βάση (Base Register) είναι ένας από τους καταχωριστές που διαθέτει ο υπολογιστής. Τον χρησιμοποιούμε για να καταχωρίσουμε σ' αυτόν κάποιο σταθερό αριθμό. Ο καταχωριστής αυτός δανείζει το περιεχόμενό του στις διάφορες εντολές, που είδαμε στα προηγούμενα (όπου τον αναφέραμε σαν B) για το σχηματισμό της διευθύνσεως μιας θέσεως στη μνήμη.

Οι κυριότερες από τις μακροεντολές της Assembly είναι οι εξής:

α) Εντολή OPEN.

Με αυτή ανοίγεται ένα αρχείο. Έχει τη μορφή:

OPEN Name

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

OPEN PAYROL

με την εντολή αυτή ανοίγεται το αρχείο με την ονομασία PAYROL.

β) Εντολή 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 η διεύθυνση της μνήμης, όπου θα γίνει η καταχώριση των πληροφοριών που διαβάζονται και D₂ η διεύθυνση της εντολής, 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 εντολή με συμβολική διεύθυνση TELOS.

γ) Εντολή εγγραφής PUT.

Με αυτή γράφομε πληροφορίες σε ένα αρχείο. Μοιάζει με την εντολή GET και έχει μορφή:

PUT Name, D₁, D₂

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

PUT RAG, DIM, K1

δ) Εντολή CLOSE.

Με αυτή κλείνομε ένα αρχείο. Έχει τη μορφή:

CLOSE Name

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

CLOSE PAYROL

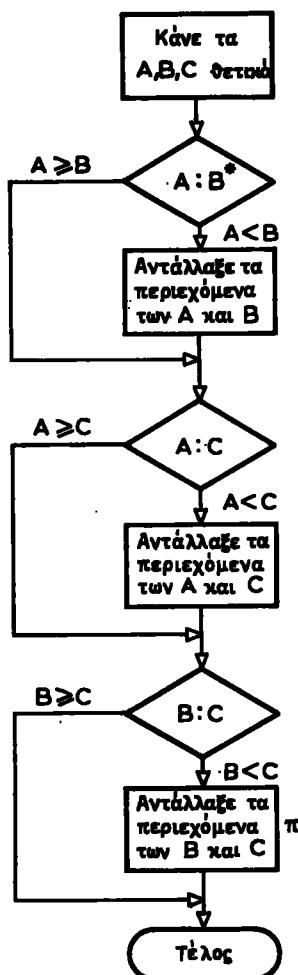
και σημαίνει ότι κλείνομε το αρχείο που έχει την ονομασία PAYROL.

8.14 Παραδείγματα προγραμμάτων ASSEMBLY.

Παράδειγμα 1.

Έστω ότι δίνονται τρεις αριθμοί A,B,C θετικοί ή αρνητικοί. Ζητείται να γραφεί πρόγραμμα Assembly, που θα μετατρέπει τους αρνητικούς (αν υπάρχουν) σε θετικούς και θα τους τακτοποιεί έτσι, ώστε ο A να περιέχει το μεγαλύτερο, ο B τον αμέσως μικρότερο και ο C τον πιο μικρό.

Το διάγραμμα ροής στην περίπτωση αυτή θα έχει την εξής μορφή (σχ. 8.14 a).



* Το $A:B$ σημαίνει σύγκριση του A με το B .

Σχ. 8.14α.

Το πρόγραμμα θα έχει την παρακάτω μορφή (σχ. 8.14β, 8.14γ).

PROGRAM

PROGRAMMER

DATE

1	Name	8	10	Operation ₁₄	16	20	Operand ₂₅	30
				START	256			
B E G I N				BALR	15, 0			
				U S I N G	* , 15			
				L M	1, 3, A			1
				L P R	1, 1			2
				L P R	2, 2			3
				L P R	3, 3			4
				C R	1, 2			5
				B C	10, COMP2			
				L R	6, 1			
				L R	1, 2			
				L R	2, 6			6
C O M P 2				C R	1, 3			
				B C	10, COMP3			
				L R	6, 1			
				L R	1, 3			
				L R	3, 6			8
C O M P 3				C R	2, 3			9
				B C	10, OUT			
				L R	6, 2			
				L R	2, 3			10
				L R	3, 6			
O U T				S T M	1, 3, A			11
				S V C	0			

Zx. 8.14B.

Σελ. 8.14γ.

Επεξήγηση των ευτολών.

Με την εντολή 1 τοποθετούνται στους καταχωριστές 1,2 και 3 οι τιμές των A,B,C.

Με τις έντολές 2, 3 και 4 μετατρέπονται οι τιμές των A,B,C σε θετικές. Με την 5 συγκρίνεται το περιεχόμενο του A με το περιεχόμενο του B. Με την 5 γίνεται ανταλλαγή των περιεχομένων αν είναι απαραίτητο. Με την 7 συγκρίνεται το περιεχόμενο του καταχωριστή 1, που περιέχει την τιμή του A, με το περιεχόμενο του καταχωριστή 3, που περιέχει τη τιμή του C.

Με την 8 γίνεται ανταλλαγή των περιεχομένων των Α και Σ αν είναι αναγκαίο. Με την 9 συγκρίνεται το περιεχόμενο του καταχωριστή 2 (δηλαδή το Β) με το περιεχόμενο του καταχωριστή 3 (δηλαδή το Σ).

Με τη 10 γίνεται ανταλλαγή των περιεχουμένων των B και C αν χρειάζεται.

Με την 11 αποθηκεύονται οι ταξινομημένες τιμές. Με τη 12 εισάγονται οι αρχικές τιμές των A, B, και C.

Паробєгуня 2.

Να γραφεί πρόγραμμα Assembly για τον υπολογισμό του αθροίσματος των 20 πρώτων φυσικών αριθμών.

Στην περίπτωση αυτή το πρόγραμμα θα έχει τη παρακάτω μορφή (σχ. 8.14δ,ε).

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	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 B 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 '			

Σx 8.146.

Ex. 8.14e.

Ασκήσεις

1. Να γραφεί πρόγραμμα Assembly για τον υπολογισμό του αθροίσματος των τετραγώνων των 20 πρώτων φυσικών αριθμών.
 2. Να γραφεί πρόγραμμα Assembly για τον υπολογισμό της τιμής της συναρτήσεως $y = 3x^2 + 4$ για $x = 1.5$
 3. Να γραφεί πρόγραμμα Assembly για τον υπολογισμό του μέσου όρου 4 αριθμών A,B,C,D.
 4. Να γραφεί πρόγραμμα Assembly για τον υπολογισμό της τιμής της συναρτήσεως $y = 2x^2 - 3x + 5$ για $x = 2.3$
 5. Να 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 η πιο απλή και η πιο εύχρηστη από τις γλώσσες προγραμματισμού υψηλού επιπέδου. Παρουσιάζει αρκετές ομοιότητες με τη γλώσσα FORTRAN που θα γνωρίσουμε στο επόμενο κεφάλαιο.

Η ονομασία της προέρχεται από τα αρχικά των *Beginner's All-Purpose Symbolic Instruction Code*, που σημαίνουν σε ελεύθερη μετάφραση, Συμβολικός Κώδικας Εντολών για αρχάριους, επειδή σχεδιάσθηκε αρχικά για να εξυπηρετήσει εκπαιδευτικά προγράμματα, όπου οι σπουδαστές επικοινωνούσαν με τον υπολογιστή μέσω τηλετύπων (*Teletypes*).

Αποτελείται από εντολές που οι κωδικοί τους είναι λέξεις της Αγγλικής γλώσσας. Τις εντολές, που συνιστούν ένα πρόγραμμα, τις γράφομε τη μία κάτω από την άλλη ακολουθώντας τη σειρά του λογικού διαγράμματος, η σχεδίαση του οποίου έχει προηγηθεί.

9.2 Βασικά στοιχεία της BASIC.

Έξι βασικά στοιχεία της BASIC θα γνωρίσουμε στην παράγραφο αυτή, τα εξής:

a) *Αριθμοί γραμμών ή αριθμοί εντολών (Line numbers ή statement numbers).*

Κάθε γραμμή που περιλαμβάνει μια εντολή στο πρόγραμμα BASIC πρέπει να αρχίζει με ένα μοναδικό αριθμό, που να αποτελείται από 5 το πολύ δεκαδικά ψηφία, δηλαδή από το 1 έως το 99999. Ο υπολογιστής εκτελεί τις εντολές του προγράμματος με τη σειρά που καθορίζεται από τους αριθμούς γραμμών.

b) *Σταθερές (Constants).*

Είναι θετικοί ή αρνητικοί δεκαδικοί αριθμοί, ακέραιοι ή κλασματικοί όπως π.χ.:

43 –172 4.546 0.0719 –65.080 κ.ο.κ.

και εμφανίζονται μέσα σε μια εντολή (κυρίως αντικαταστάσεως), ακριβώς όπως και σε μια απλή αλγεβρική παράσταση. Αριθμοί χωρίς πρόσημο θεωρούνται θετικοί.

Εκφράσεις όπως $16/3$, $-7/5$ δεν θεωρούνται απλές σταθερές, αλλά ότι αποτελούνται από δύο σταθερές.

Επειδή οι υπολογιστές δέχονται αριθμούς με περιορισμένο πλήθος ψηφίων, τους πολύ μεγάλους ή πολύ μικρούς αριθμούς είναι πιο βολικό να τους εκφράζομε σε εκθετική μορφή. Στη μορφή αυτή ο αριθμός αποτελείται από δύο τμήματα, το

ψηφιακό τμήμα και το τμήμα του εκθέτη. Το τμήμα του εκθέτη περιλαμβάνει το γράμμα E που συμβολίζει τη βάση 10 και ένα ακέραιο αριθμό που είναι η δύναμη του 10 π.χ.:

$$\begin{array}{l} \overbrace{1.74}^{\text{Ψηφιακό} \\ \text{Τμήμα}} \quad \overbrace{E + 4}^{\text{Τμήμα} \\ \text{Εκθέτη}} = 1.74 \times 10^{14} \\ -0.098 E - 2 = -0.098 \times 10^{-2} \\ 17E3 = 17 \times 10^3 \quad \text{k.o.k.} \end{array}$$

γ) Μεταβλητές (Variables).

Οι μεταβλητές τόσο στη BASIC όσο και στις άλλες γλώσσες προγραμματισμού παίζουν το ρόλο των συμβολικών διευθύνσεων της μνήμης.

Οι ονομασίες των μεταβλητών σχηματίζονται εδώ με ένα κεφαλαίο χαρακτήρα του λατινικού αλφαριθμητικού, που μπορεί να ακολουθείται και από ένα αριθμητικό χαρακτήρα, π.χ. οι μεταβλητές:

A, B, J, X5,A1 κλπ.

είναι δεκτές στη BASIC. Οι μεταβλητές:

AB, X22, 4A, κλπ.

δεν είναι δεκτές.

Κάθε μεταβλητή παριστάνει, όπως έπιπλο, μια θέση μνήμης στην οποία δίνεται κάποιο αριθμητικό περιεχόμενο είτε με τη βοήθεια μιας εντολής αναγνώσεως, είτε μιας εντολής αντικαταστάσεως, όπως θα δούμε παρακάτω.

δ) Σύμβολα αριθμητικών πράξεων.

Τα σύμβολα που χρησιμοποιούμε στη BASIC για να απεικονίσουμε τις διάφορες αριθμητικές πράξεις ανάμεσα σε δύο δεδομένα, φαίνονται στον Πίνακα 9.2.1.

ΠΙΝΑΚΑΣ 9.2.1.

Τα σύμβολα των αριθμητικών πράξεων της BASIC

Αριθμητική πράξη	Σύμβολο της Αριθμητικής	Σύμβολο της BASIC
Πρόσθεση	+	+
Αφαίρεση	-	-
Πολλαπλασιασμός	· ή x	*
Διαίρεση	: ή / ή —	/
Υψωση σε δύναμη	δεν υπάρχει	**

ε) Κωδικοποίηση αριθμητικών και αλγεβρικών εκφράσεων.

Στην άλγεβρα συχνά χρησιμοποιούμε μικρούς ελληνικούς αλφαριθμητικούς χαρακτήρες για να παριστάνουμε τους άγνωστους μιας εξισώσεως ή τις παραμέτρους και τις ανεξάρτητες μεταβλητές μιας συναρτήσεως ή τους συντελεστές ενός πολυωνύμου.

μου κλπ. Σε όλες τις γλωσσες προγραμματισμού χρησιμοποιούμε, όπως είπαμε, με ανάλογο τρόπο, μόνο κεφαλαία γράμματα του λατινικού αλφαβήτου, τις μεταβλητές.

Τα σύμβολα των αριθμητικών πράξεων χρησιμοποιούνται σε συνδυασμό με γράμματα και αριθμούς για να σχηματίσουν αλγεβρικές εκφράσεις. Το ίδιο και στη BASIC, όπου με συνδυασμό μεταβλητών, σταθερών και συμβόλων, σχηματίζονται ανάλογες εκφράσεις. Παράδειγμα τέτοιων κωδικοποίησεων βλέπομε στον Πίνακα 9.2.2 που ακολουθεί:

ΠΙΝΑΚΑΣ 9.2.2

Παραδείγματα πωδικοποίησεων αλγεβρικών παραστάσεων

Αλγεβρική παράσταση	Έκφραση BASIC
$a + b$	$A + B$
$4x - 3$	$4^{\circ} X - 3$
$\frac{y}{2} + d - \frac{5}{6}$	$C / 2 + D - 5 / 6$
$ax^2 + bx + c$	$A^{\circ} X ^{**} 2 + B^{\circ} X + C$
$2a + \frac{4}{\beta}$	$2^{\circ} A + 4 / B$

Θα πρέπει να έχομε υπ' όψη μας ότι σε τέτοιες σύνθετες εκφράσεις, η εκτέλεση από τον υπολογιστή των επί μέρους αριθμητικών πράξεων ακολουθεί κάποια ιεραρχία. Έτσι, πρώτα εκτελούνται οι υψώσεις σε δύναμη, κατόπιν οι πολλαπλασιασμοί και οι διαιρέσεις και τελευταία οι προσθέσεις και οι αφαιρέσεις.

Όταν στην ίδια παράσταση υπάρχουν δύο ή περισσότερες ιεραρχικά ισοδύναμες πράξεις, π.χ. μια πρόσθεση και μια αφαίρεση ή τρεις πολλαπλασιασμοί κλπ., τότε εκτελείται πρώτα αυτή που προηγείται στην παράσταση από τα αριστερά.

στη Παρενθέσεις.

Στη BASIC χρησιμοποιούμε τις παρενθέσεις, όπως και στην άλγεβρα, όταν θέλουμε να ξεχωρίσουμε το ότι ορισμένες πράξεις εκτελούνται μόνο σε μερικούς από τους όρους μιας παραστάσεως.

Θα πρέπει όμως να προσέξουμε περισσότερο στο γράψιμο αριθμητικών παραστάσεων και στη χρήση παρενθέσεων, έχοντας υπ' όψη και την ιεραρχία που υπάρχει στην εκτέλεση των αριθμητικών πράξεων, όπως είπαμε προηγουμένως.

Δηλαδή αν υποθέσουμε ότι μας δίνεται για κωδικοποίηση η έκφραση $\frac{a}{\beta + 5}$

Αν τη γράψομε $A/B + 5$ θα έχομε κάνει σφάλμα, γιατί, σύμφωνα με τα προγράμματα, θα γίνει πρώτα η διαιρέση του A με το B και στο πλήρο θα προστεθεί το 5, ενώ εμείς θέλομε να διαιρέσουμε το A με το B + 5. Γι' αυτό κλείνομε όλο το B + 5 μέσα σε μια παρένθεση και έτσι η σωστή κωδικοποίηση θα είναι:

$$A/(B + 5)$$

Μπορούμε να βάζουμε παρενθέσεις σε μια παράσταση χωρίς περιορισμό όσον αφορά το πλήθος τους, αρκεί να θυμόμαστε ότι θα πρέπει τελικά η παράστασή μας να κλείνει, δηλαδή να έχουμε τόσες αριστερές παρενθέσεις, όσες και δεξιές. Διαφορετικά ο υπολογιστής θα δύσει μήνυμα λάθους για την παράσταση.

Μερικά παραδείγματα για τον τρόπο χρήσεως των παρενθέσεων βλέπομε στον Πίνακα 9.2.3.

ΠΙΝΑΚΑΣ 9.2.3.

Παραδείγματα για τον τρόπο χρήσεως των παρενθέσεων στη BASIC

Άλγεβρική έκφραση	Έκφραση BASIC
$\left(\frac{a + b}{y} \right)^2$	$((A+B) / C) ** 2$
$4(a+b) + \gamma \delta (x-3)$	$4^* (A+B) + C^* D^* (X-3)$
$2 \left(3a - \frac{b}{y+\delta} \right)$	$2^* (3^* A - B / (C + D))$

9 Μαθηματικές συναρτήσεις.

Η BASIC χρησιμοποιείται αρκετά συχνά για τη λύση απλών μαθηματικών προβλημάτων. Στα προβλήματα αυτά προκύπτει πολλές φορές ή ανάγκη για τον υπολογισμό ενός τριγωνομετρικού αριθμού (ημ, συν, κλπ.) ή μιας τετραγωνικής ρίζας ή ενός λογαρίθμου κ.ο.κ. Για να μην είμαστε υποχρεωμένοι να γράφομε κάθε φορά και ένα ξεχωριστό πρόγραμμα για κάθε τέτοιο υπολογισμό, οι κατασκευαστές έχουν ενσωματώσει στο μεταφραστικό πρόγραμμα της BASIC αρκετά βοηθητικά υποπρογράμματα, για τον υπολογισμό των συναρτήσεων που εμφανίζονται συχνότερα. Τα υποπρογράμματα αυτά μπορούμε να τα χρησιμοποιούμε μέσα σε κάθε πρόγραμμά μας, καλώντας τα απλώς με βάση προσδιορισμένη ονομασία και με παράμετρο τη μεταβλητή, για την τιμή της οποίας ζητείται ο υπολογισμός. Τα ονομάζομε συναρτήσεις του συστήματος.

Οι πιο γνωστές συναρτήσεις συστήματος της BASIC με τις κωδικές τους ονομασίες φαίνονται στον Πίνακα 9.2.4.

Οι συναρτήσεις αυτές αναφέρονται σε μια κωδικοποιημένη έκφραση ακριβώς όπως και οι απλές μεταβλητές. Παραδείγματα βλέπομε στον Πίνακα 9.2.5.

9.3 Κατηγορίες εντολών BASIC.

Τις εντολές της BASIC κατατάσσουμε στις εξής κατηγορίες:

- α) Εντολές αντικαταστάσεως.
- β) Εντολές εισδοου-εξόδου.
- γ) Εντολές ελέγχου ή διακλαδώσεως.
- ε) Εντολές δηλωτικές.

ΠΙΝΑΚΑΣ 9.2.4.
Οι συναρτήσεις συστήματος της BASIC

Συνάρτηση	Λειτουργία
SIN(X)	Υπολογίζει το ημίτονο του X. Το X εκφράζεται σε ακτίνια (rad).
COS(X)	Υπολογίζει το συνημίτονο του X. Το X εκφράζεται σε ακτίνια.
ATN(X)	Υπολογίζει το τάξιο εφαπτομένης του X.
EXP(X)	Υπολογίζει την πιμή του e^x . Το e είναι η βάση των φυσικών ή νεπερίων λογαρίθμων = 2.73
LOG(X)	Υπολογίζει το φυσικό λογάριθμο (με βάση το e) του X.
LGT(X)	Υπολογίζει το δεκαδικό λογάριθμο (με βάση το 10) του X.
ABS(X)	Υπολογίζει την απόλυτη πιμή του X.
SQR(X)	Υπολογίζει την τετραγωνική ρίζα του X.
INT(X)	Υπολογίζει το ακέραιο τμήμα του X.

ΠΙΝΑΚΑΣ 9.2.5.
Παραδείγματα κωδικοποίησεως εκφράσεων που περιέχουν συναρτήσεις

Αλγεβρική έκφραση	Έκφραση BASIC
$3 \text{ ημ } x + x + \gamma d$ $e^x \text{ συν } x + \frac{3}{4}$ $2\sqrt{x} + a^2 + b^2$ $\frac{1}{x} + 5 \text{ λογ } x$	$3 * \text{SIN}(X) + \text{ABS}(X) + C * D$ $\text{EXP}(X) * \text{COS}(X) + 3 / 4$ $2 * \text{SQR}(X) + A ** 2 + B ** 2$ $1 / X + 5 * \text{LGT}(X)$

9.4 Τρόπος αναγραφής των εντολών BASIC.

Δεν υπάρχουν ειδικά έντυπα κωδικογραφήσεως για την αναγραφή των εντολών BASIC.

Κάθε εντολή αρχίζει να γράφεται από την πρώτη στήλη μιας γραμμής, που χωρίζεται συνολικά σε 80 στήλες και που αντιστοιχεί στο περιεχόμενο ενός δελτίου.

9.5 Εντολή αντικαταστάσεως. Η εντολή LET.

Η εντολή αυτή που είναι από τις πιο βασικές της BASIC έχει την εξής μορφή:

η LET μεταβλητή = έκφραση ή σταθερή

Το ο είναι ο αριθμός εντολής. Η υπόλοιπη εντολή αποτελείται από τη λέξη LET, που ακολουθείται από μια μεταβλητή (με κάποιο κενό διάστημα μεταξύ τους), το ίσον(=) και από μία κωδικοποιημένη έκφραση BASIC ή μία σταθερή.

Με τη βοήθεια της εντολής LET γίνεται ο υπολογισμός της αριθμητικής τιμής της εκφράσεως του δευτέρου μέλους της ισότητας (γι' αυτό θα πρέπει όλες οι μεταβλητές που αναφέρονται στην έκφραση να έχουν γνωστό περιεχόμενο) και στη συνέχεια καταχωρίζεται σαν περιεχόμενο στη μεταβλητή του πρώτου μέλους. Το προηγούμενο περιεχόμενο, που ενδεχομένως είχε, καταστρέφεται. Γι' αυτό το λόγο και η εντολή αυτή λέγεται **εντολή αντικαταστάσεως**. Π.χ. στην εντολή:

30 LET Z2 = SQR(X4) – 10

Θα γίνει ο υπολογισμός της τιμής του δευτέρου μέλους, δηλαδή του SQR(X4) – 10 και το αποτέλεσμα θα καταχωρισθεί σαν περιεχόμενο της Z2, της οποίας το προηγούμενο περιεχόμενο — αν είχε — χάνεται. Φυσικά για να υπολογισθεί η έκφραση SQR(X4) – 10, θα πρέπει να είναι γνωστή η τιμή (ή καλύτερα το περιεχόμενο) της μεταβλητής X4, τη στιγμή που εκτελείται η παραπάνω εντολή.

Ας δούμε ένα άλλο παράδειγμα:

20 LET A = 4*A – 17

Εδώ βλέπομε ότι η μεταβλητή A εμφανίζεται και στο πρώτο και στο δεύτερο μέλος. Πώς θα εκτελεσθεί. Εφ' όσον θα γίνει, όπως είπαμε, πρώτα ο υπολογισμός του δευτέρου μέλους, η μεταβλητή A θα μετάσχει στην έκφραση με το παλό της περιεχόμενο, δηλαδή αυτό που είχε τη στιγμή που αρχίζει να εκτελείται η εντολή. Το αποτέλεσμα του υπολογισμού θα καταχωρισθεί στη συνέχεια πάλι στην A, καταστρέφοντας την προηγούμενη τιμή της.

Παρόμοιες μορφές εντολών LET χρησιμοποιούμε για να αυξάνομε το περιεχόμενο μιας μεταβλητής, που τη χρησιμοποιούμε σαν κοινό αθροιστή, πολλαπλασιαστή ή σαν απαριθμητή (Counter) σ' ένα πρόγραμμα π.χ.:

25 LET	S = S + X**2	περίπτωση αθροιστή
30 LET	J = J*K	περίπτωση πολλαπλασιαστή
42 LET	N = N + 1	περίπτωση απαριθμητή

Σε νεώτερους τύπους της BASIC γίνονται δεκτές και μορφές πολλαπλής εντολής αντικαταστάσεως, όπως π.χ.:

40 LET A = B = C = D = 0

που στους αρχικούς τύπους θα έπρεπε να αναλυθεί στις εξής απλές εντολές LET:

40	LET	A = 0
41	LET	B = 0
42	LET	C = 0
43	LET	D = 0

9.6 Εντολές εισάγου-εξόδου.

α) Εντολές READ και DATA.

Με τις εντολές αυτές εισάγονται οι κατάλληλες αριθμητικές τιμές δεδομένων στις μεταβλητές που χρησιμοποιούνται στο BASIC πρόγραμμα.

Η δομή των δύο εντολών είναι:

- η READ μεταβλητή, μεταβλητή, μεταβλητή, κλπ.
- η DATA σταθερή, σταθερή, σταθερή, κλπ.

Η είναι ο αριθμός εντολής. Όταν οι μεταβλητές στη READ ή οι σταθερές στη DATA είναι περισσότερες από μία, τότε τις ξεχωρίζομε, γράφοντας κόμματα μεταξύ τους. Καμιά από τις εντολές αυτές δεν εμφανίζεται μόνη της σε ένα πρόγραμμα. Η εντολή DATA περιλαμβάνει τις αξίες των δεδομένων, που θα δοθούν στις μεταβλητές, οι οποίες αναφέρονται στην εντολή READ.

Κανονικά, όσες είναι οι μεταβλητές που εμφανίζονται στις διάφορες εντολές READ, όσες θα πρέπει να είναι και οι σταθερές που εμφανίζονται στις διάφορες εντολές DATA του ίδιου προγράμματος. Η αντιστοίχιση είναι μία προς μία με τη σειρά έμφανίσεως, π.χ.:

```
10  READ  A, B1, C
    20  READ  D, F8
    30  DATA  40, 1.2, 5.6, 7.8
    40  DATA  -1.8
```

Εδώ στην πρώτη μεταβλητή, την A, θα δοθεί η τιμή της πρώτης σταθερής, η 40. Ακολούθως στη B1 θα δοθεί η τιμή 1.2, στη C η 5.6 κ.ο.κ.

Όπως δηλαδή μπορούμε εύκολα να διαπιστώσουμε, θα ήταν δυνατό να είχαμε ενσωματώσει τη δεύτερη εντολή DATA στην πρώτη και να έχομε το ίδιο αποτέλεσμα:

```
10  READ  A, B1, C
    20  READ  D, F8
    30  DATA  40, 1.2, 5.6, 7.8, -1.8
```

Σύμφωνα με το σχεδιασμό της BASIC, πριν από την εκτέλεση του προγράμματος, όλες οι σταθερές των εντολών DATA αποτελούν μια ομάδα δεδομένων (Data Block), που εμφανίζονται με την ίδια τάξη, όπως και στο πρόγραμμα. Κάθε φορά που εκτελείται μια εντολή READ, γίνεται η σωστή αντιστοίχιση μιας σταθερής, από αυτές που βρίσκονται στην περιοχή DATA, με την αντίστοιχη σε τάξη μεταβλητή της εντολής 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 το λάθος είναι σοβαρό, γιατί δεν μπορεί να γίνει αντιστοίχιση σε όλες τις μεταβλητές των εντολών READ και η εκτέλεση του προγράμματος σταματά, ενώ συγχρόνως ο 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

```

Το λάθος εδώ είναι σοβαρό, γιατί δεν υπάρχει σταθερή για να αντιστοιχηθεί στην τελευταία μεταβλητή, δηλαδή την C.

β) Εντολή INPUT.

Η εντολή αυτή χρησιμοποιείται όταν εισάγονται τα δεδομένα του προβλήματος με τη βοήθεια της περιφερειακής μονάδας αναγνώσεως δελτίων (CARD READER). Η δομή της εντολής είναι:

η INPUT μεταβλητή, μεταβλητή,...

Αντίθετα με την εντολή READ που πρέπει να συνοδεύεται και από εντολή DATA, η INPUT εμφανίζεται μόνη της. Τα δεδομένα στην περίπτωση αυτή, εισάγονται στον υπολογιστή με δελτίο, από τη μονάδα αναγνώσεως δελτίων. Π.χ.:

10 INPUT A,B,C

Τα δεδομένα διατρυπώνονται επάνω στο δελτίο με συνεχή τρόπο, χωριζόμενα με κόμματα.

γ) Εντολή PRINT.

Η εντολή PRINT χρησιμοποιείται για να τυπωθούν: α) Τιμές, περιεχόμενα μεταβλητών. β) Αποτελέσματα αριθμητικών υπολογισμών. γ) Διάφορα μηνύματα και δ) συνδυασμός των α,β,γ περιπτώσεων.

Στην περίπτωση (α) η δομή της είναι:

η PRINT μεταβλητή, μεταβλητή,...

π.χ.

60 PRINT A, B, F8, C

Στην περίπτωση (β) η δομή της είναι:

η PRINT έκφραση BASIC, έκφραση BASIC,...

π.χ.

40 PRINT (- B + R) / (Z*A), (-B-R)/ (Z*A)

Υποτίθεται, βέβαια, ότι οι μεταβλητές που αναφέρονται στις εκφράσεις έχουν γνωστό περιεχόμενο.

Στην περίπτωση (γ) τα μηνύματα, που θέλουμε να τυπώσουμε, τα περικλείομε μέσα σε απόστροφους. Πρέπει να είναι γραμμένα με κεφαλαίους λατινικούς χαρακτήρες. Η δομή της εντολής είναι:

η PRINT μήνυμα, μήνυμα,...

π.χ.

20 PRINT "THE SQUARE ROOT OF X IS"

9.7 Μερικές ακόμη εντολές.

α) Εντολή REM.

Επεξηγηματικά σχόλια, που γράφονται σε διάφορα σημεία ενός προγράμματος, το κάνουν πιο κατανοητό. Τέτοια σχόλια αναφέρονται στην ταυτότητα του προγράμματος, στην περιληπτική περιγραφή της μεθόδου που χρησιμοποιείται για την επίλυση του προβλήματος, στο ρόλο που διαδραματίζουν ορισμένες εντολές κλπ.

Τα σχόλια (Comments) γράφονται με τη βοήθεια της εντολής REM, που έχει τη μορφή:

η REM σχόλια

Η εντολή αυτή δεν εκτελείται. Ο ρόλος της είναι σαφώς βοηθητικός. Επομένως η εμφάνισή της δεν επηρεάζει καθόλου την εκτέλεση ενός προγράμματος. Μπορούμε να εμφανίζομε δύσες εντολές REM θέλομε και σε οποιοδήποτε σημείο του προγράμματος, π.χ.:

```
10    REM    PROGRAM    FOR    THE
20    REM    SOLUTION    OF    A    QUADRATIC
30    REM    EQUATION
```

Τα σχόλια γράφονται με κεφαλαίους λατινικούς χαρακτήρες.

β) Εντολή END.

Η END είναι η τελευταία εντολή κάθε προγράμματος BASIC καί γι' αυτό πρέπει να έχει και το μεγαλύτερο αριθμό εντολής (συνήθως της δίνομε τον αριθμό 99999).

9.8 Μερικά απλά προγράμματα BASIC.

Σύμφωνα με τα δύο είπαμε μέχρι εδώ, θα προσπαθήσουμε να γράψουμε μερικά απλά προγράμματα στη γλώσσα BASIC.

Παράδειγμα 1.

Ας υποθέσουμε ότι θέλομε να βρούμε το μέσο όρο τεσσάρων αριθμών A, B, C, D. Το πρόγραμμα θα έχει τη μορφή:

```
10    REM    YPOLOGISMOS TOY MESOY OROY
20    READ  A,B,C,D
30    LET    M = (A + B + C + D)/4
40    PRINT "O MESOS OROS EINAI", M
50    DATA  1.2, 4.5, 7.8, 9.4
99999 END
```

Παράδειγμα 2.

Έστω τώρα ότι θέλομε να υπολογίσουμε την τιμή της συναρτήσεως $\psi = x^3 - 5x^2 + 2\sqrt{x}$ για $x = 1.8$. Το πρόγραμμα θα είναι:

```

10 REM YPOLOGISMOS THS TIMHS SYNARTHSEOS
20 READ X
30 LET Y = X**3 - 5*X**2 + 2*SQR(X)
40 PRINT "H TIMH THS SYNARTHSEOS EINAI", Y
50 DATA 1.8
9999 END

```

9.9 Εντολές ελέγχου και διακλαδώσεως.

Με τις εντολές αυτές πετυχαίνομε:

- α) Άλμα χωρίς συνθήκη σε κάποιο άλλο σημείο του προγράμματος.
- β) Διακλάδωση με συνθήκη.

Μπορεί έτσι ο υπολογιστής να πάρνει λογικές αποφάσεις βασιζόμενος σε προηγούμενους υπολογισμούς και να εκτελεί κατά περίπτωση διάφορες ομάδες από εντολές, ή να εκτελεί για ορισμένες φορές, επαναληπτικά, την ίδια ομάδα εντολών.

α) Εντολή GOTO.

Σε μερικά προγράμματα χρειάζεται να εκτελεσθούν οι εντολές με διαφορετική σειρά, από εκείνη που καθορίζουν οι αριθμοί εντολών. Ένας τρόπος για να αλλάξει η κανονική ροή εκτελέσεως του προγράμματος, είναι να χρησιμοποιηθεί η εντολή GOTO που έχει τη μορφή:

n GOTO αριθμός εντολής

όπου: σαν αριθμό εντολής αναφέρομε τον αριθμό της εντολής που πρέπει να εκτελεσθεί στη συνέχεια, π.χ.:

30 GOTO 40

σημαίνει ότι σαν επόμενη εντολή θα εκτελεσθεί η εντολή με αριθμό 40. Η εντολή αυτή μπορεί να βρίσκεται σε κάποιο χαμηλότερο σημείο του προγράμματος, οπότε όλες οι άλλες εντολές που παρεμβάλλονται, αγνοούνται.

Παράδειγμα.

Το παρακάτω πρόγραμμα επιλύει χωρίς διερεύνηση μια εξίσωση δευτέρου βαθμού με ένα άγνωστο. Η επίλυση επαναλαμβάνεται δύο φορές για δύο εξισώσεις.

```

10 PRINT 'A', 'B', '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

```

Με το πρόγραμμα αυτό διαβάζεται μία σειρά δεδομένων (οι συντελεστές του ανώστου), υπολογίζονται οι ρίζες R1 και R2 και τυπώνονται. Με την εντολή GOTO μεταφέρεται ο έλεγχος πάλι στην εντολή 20, με την οποία διαβάζεται άλλη μία τριάδα δεδομένων και επαναλαμβάνεται η ίδια διαδικασία, μέχρις ότου εξαντληθούν τα δεδομένα που αναφέρονται στην εντολή DATA.

8) Εντολή IF-THEN.

Με την εντολή GOTO, όπως είπαμε, μπορούμε να πετύχομε άλμα σε ένα σημείο του προγράμματος χωρίς συνθήκη. Σε πολλές όμως περιπτώσεις θέλομε αυτή η διακλάδωση να γίνεται μόνο όταν ισχύει μια συνθήκη, ενώ διαφορετικά να συνεχίζεται η κανονική ροή. Τότε χρησιμοποιούμε την εντολή IF-THEN που έχει μορφή:

η IF λογική έκφραση συνθήκης THEN αριθμός εντολής

Πριν όμως την περιγραφή της, ας μιλήσουμε λίγο για τις λογικές συνθήκες.

Οι λογικές συνθήκες είναι εντελώς ανάλογες με τις μαθηματικές συνθήκες ή περιορισμούς που θέτομε σε ένα πρόβλημα. Συχνά λέμε π.χ. ότι θα πρέπει $x < a$ ή $x < \beta \leq \psi$ ή $0 \leq x < 3$ ή $\delta = 0$ κ.ο.κ. Τα ίδια αυτά σύμβολα (λογικά σύμβολα) $<$, $>$, $=$ κλπ., τα χρησιμοποιούμε για να κωδικοποιήσουμε στη BASIC τέτοιες λογικές σχέσεις (Πίνακας 9.9.1). Αν η λογική συνθήκη που αναφέρεται μετά το IF ισχύει (ικανοποιείται), τότε το πρόγραμμα συνεχίζεται με την εκτέλεση της εντολής, της οποίας ο αριθμός αναφέρεται μετά τη λέξη THEN. Διαφορετικά συνεχίζεται κανονικά με την εκτέλεση της επόμενης εντολής.

ΠΙΝΑΚΑΣ 9.9.1.

Τα σύμβολα των λογικών σχέσεων στη BASIC

Σύμβολο	Επεξήγηση	Παράδειγμα	Ανάλυση
=	ίσο προς	IF X=Y THEN 30	Αν το X είναι ίσο προς το Y, τότε η εκτέλεση του προγράμματος θα συνεχισθεί από την εντολή 30, διαφορετικά από την αμέσως επόμενη.
< >	διάφορο του, όχι ίσο με	IF X < > THEN 40	Αν το X είναι διάφορο του Y (άνισο), τότε θα εκτελεσθεί η εντολή 40, διαφορετικά η επόμενη.
<	μικρότερο από	IF X < Y THEN 50	Η εκτέλεση θα συνεχισθεί από την εντολή 50, μόνο αν X < Y.
<=	μικρότερο ή ίσο	IF X <= Y THEN 45	Η εντολή 45 θα εκτελεσθεί αμέσως μετά την εντολή IF, μόνο όταν X < Y
>	μεγαλύτερο από	IF X > Y THEN 20	Η εντολή 20 θα εκτελεσθεί αμέσως μετά την εντολή IF, μόνο όταν X > Y
> =	μεγαλύτερο ή ίσο	IF X > Y THEN 60	Η εντολή 60 θα εκτελεσθεί αμέσως μετά την εντολή IF μόνο όταν X > Y

Τις λογικές συνθήκες τις διακρίνουμε σε απλές και σύνθετες. Απλές είναι οι συνθήκες που περιλαμβάνουν μία λογική σχέση ανάμεσα σε δύο μόνο όρους, π.χ. $a < \beta$, $x > \psi$, $z > 0$ κλπ. Σύνθετες είναι οι συνθήκες που περιλαμβάνουν σχέσεις μεταξύ περισσοτέρων όρων, π.χ. $a < \beta < \gamma$, $-2 < z < 4$, $0 < x < 6$ κλπ. Τις σύνθετες συνθήκες στη BASIC τις αναλύουμε σε ανάλογο αριθμό απλών συνθηκών, π.χ. την $a < \beta < \gamma$ στις απλές $a < \beta$ και $\beta < \gamma$.

Είναι δυνατή επίσης η σύγκριση όχι μόνο ανάμεσα σε δύο απλούς όρους-μεταβλητές, αλλά ακόμη και σε δύο απλές μαθηματικές εκφράσεις π.χ.:

IF $B^{**2} < 4 * A^*C$ THEN 100

Παράδειγμα 1.

Σαν πρώτο παράδειγμα, όπου χρησιμοποιούμε την εντολή IF, δίνεται η επίλυση μιας δευτεροβάθμιας εξισώσεως με ένα άγνωστο, που περιλαμβάνει και τη σχετική διερεύνηση για το είδος των ριζών και αποτελεί βελτίωση του προγράμματος της 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.

Έστω ότι δίνονται αριθμοί διάφοροι του μηδενός και ζητείται να γραφεί πρόγραμμα BASIC που να βρίσκει το μεγαλύτερο. Επειδή όλοι οι αριθμοί είναι διάφοροι του μηδενός, μπορούμε να ρυθμίσουμε το τέλος του προγράμματος προσθέτοντας στο τέλος όλων των κανονικών δεδομένων τον πλασματικό δεδομένο-αριθμό 0 και να ελέγχουμε κάθε φορά που διαβάζεται ένα δεδομένο, αν έχει τιμή 0. Αν ναι, τότε θα σημαίνει ότι είναι το πλασματικό δεδομένο, οπότε θα πρέπει να τελειώσει

το πρόγραμμα, αν όχι, θα σημαίνει ότι είναι κανονικό δεδομένο και θα ακολουθήσει η προβλεπόμενη κανονική επεξεργασία. Το πρόγραμμα θα έχει, λοιπόν, τη μορφή:

```

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

```

Βούλαμε μια περιοχή της μνήμης, όπου κρατείται ο μεγαλύτερος από τους αριθμούς που συγκρίνομε κάθε φορά.

Παράδειγμα 3.

Στο παράδειγμα αυτό θα δούμε τη χρήση της εντολής IF-THEN στον έλεγχο του περιεχομένου ενός απαριθμητού (Counter).

Έστω ότι ζητείται ο υπολογισμός της τιμής της παραστάσεως:

$$Sx^v = x^1 + x^2 + \dots + x^{30} \text{ για τη τιμή } x = 1.3$$

Στην περίπτωση αυτή πρέπει να προστεθούν 30 όροι της μορφής x^v . Το πρόγραμμα θα έχει την 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 τον απαριθμητή (Counter) N για να μετρά τις φορές που επαναλαμβάνεται η διαδικασία της προσθέσεως. Μετά από κάθε επανάληψη, το περιεχόμενό του αυξάνεται κατά 1. Παράλληλα ο N δανείζει το περιεχόμενό του για τη διαμόρφωση του όρου x^v κάθε φορά. Η μεταβλητή S παριστάνει ένα αθροιστή, στον οποίο δύνομε, ώπως και στον N, αρχική τιμή 0. Η εντολή 50 IF N < 30 THEN 40 ελέγχει το περιεχόμενό του N. Αν αυτό είναι μικρότερο του 30, τότε γίνεται άλμα στήν εντολή 40, από όπου συνεχίζεται η εκτέλεση του προγράμματος, διαφορετικά συνεχίζεται με την άμεσως επόμενη εντολή 70 PRINT S.

γ) Εντολή ON-GOTO.

Είναι μια λιγότερο εύχρηστη εντολή ελέγχου. Η μορφή της είναι:

η ON έκφραση BASIC GOTO αριθ. εντολής, αριθ. εντολής,...
(μέχρι 4)

Η εντολή λειτουργεί ως έξής: Υπολογίζεται πρώτα η τιμή της εκφράσεως BASIC και κρατείται το ακέραιο μέρος της. Αν ισούται με 1, τότε θα εκτελεσθεί στη συνέχεια η εντολή με αριθμό τον πρώτο που εμφανίζεται μετά τη λέξη GOTO. Αν ισούται με 2, τότε θα εκτελεσθεί η εντολή με αριθμό το δεύτερο μετά τη GOTO κ.ο.κ. Φυσικά θα πρέπει ή τιμή του ακεραίου μέρους της εκφράσεως να είναι >1 και ≤ 4 , π.χ.:

90 ON C GOTO 210, 330, 460, 570

που σημαίνει ότι, αν το ακέραιο μέρος του C είναι 1, θα εκτελεσθεί στη συνέχεια η εντολή με αριθμό 210. Αν το ακέραιο μέρος είναι 2, θα εκτελεσθεί η εντολή 330, αν το ακέραιο μέρος είναι 3, θα εκτελεσθεί η 460 και τέλος, αν το ακέραιο μέρος είναι 4, θα εκτελεσθεί η 570.

δ) Εντολή STOP.

Πολλές φορές, κατά τον έλεγχο μιας συνθήκης, θέλομε μια από τις διακλαδώσεις του προγράμματος να οδηγεί στον τερματισμό του π.χ.:

70 GOTO 99999

.....
99999 END

Αντί της πρώτης εντολής, θέτομε την
70 STOP

και το αποτέλεσμα είναι το ίδιο.

9.10 Εντολές FOR-NEXT.

Στα προηγούμενα είδαμε πως με τη βοήθεια των εντολών GOTO και IF-THEN μπορούμε να επαναλάβουμε την εκτέλεση ενός τμήματος του προγράμματος για περισσότερες από μία φορές. Σε παρόμοιες περιπτώσεις η κωδικογράφηση διευκολύνεται πάρα πολύ από το ζεύγος εντολών FOR και NEXT. Η FOR έχει τη γενική μορφή:

η FOR μεταβλητή = έκφραση TO έκφραση STEP έκφραση

και συνδιάζεται με την εντολή NEXT που έχει τη μορφή:

NEXT μεταβλητή

Οι δύο αυτές εντολές εμφανίζονται πάντα μαζί. Προηγείται η FOR και ακολου-

Θεί η NEXT. Μεταξύ τους παρεμβάλλεται η ομάδα των έντολών, των οποίων θέλομε τη συνεχή επαναληπτική εκτέλεση.

Η μεταβλητή, που εμφανίζεται μετά τη λέξη FOR, είναι ένας δείκτης (δείκτης της FOR) που παίρνει αρχική τιμή, την τιμή που δείχνει η έκφραση μετά το = και τελευταία την τιμή της εκφράσεως που εμφανίζεται μετά το TO. Η τιμή της εκφράσεως μετά τη λέξη STEP είναι το βήμα, δηλαδή το ποσό, με το οποίο αυξάνει κάθε φορά η τιμή του δείκτη.

Στην εντολή NEXT, η μεταβλητή είναι η ίδια που αναφέραμε και στη FOR.

Η διαδικασία επαναλήψεως ενός τμήματος του προγράμματος καλείται Loop.

Ας δούμε όμως ένα παράδειγμα.

Έστω ότι θέλομε να βρούμε το άθροισμα των 100 πρώτων φυσικών αριθμών.

Το πρόγραμμα που θα γράψουμε με τη βοήθεια των εντολών FOR-NEXT θα είναι το εξής:

```

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 με το πρόγραμμα αυτό, η εντολή 30 θα εκτελεσθεί συνολικά 100 φορές. Το πλήθος αυτό των ανακυκλώσεων καθορίζεται από τις εντολές 20 και 40. Ο δείκτης I παιζεί διπλό ρόλο. Αφ' ενός μεν χρησιμοποιείται σαν απαριθμητής του πλήθους των επαναλήψεων (Loops), αφ' ετέρου δε δανείζει τις τιμές του στην κύρια σχέση υπολογισμού:

30 LET SUM = SUM + I

Η αρχική τιμή που παίρνει ο I είναι 1 και η τελική 100, ενώ το βήμα είναι 1. Με την εντολή 40 σημειώνεται το τέλος του επαναλαμβανομένου τμήματος του προγράμματος.

Ας εξετάσουμε ποι αναλυτικά τον τρόπο εκτέλεσεως αυτού του προγράμματος.

Στην αρχή εκτελείται η εντολή 10 και κατόπιν (εντολή 20) η επανάληψη του προγράμματος (εντολές 20-40). Την πρώτη φορά έχομε I = 1 και θα εκτελεσθεί για πρώτη φορά η εντολή

30 LET SUM = SUM + I

Με την εντολή 40 επιστρέφει ο έλεγχος στην 20, το I γίνεται 2 (δηλ. προηγούμενη τιμή + βήμα) και πραγματοποιείται για δεύτερη φορά η εκτέλεση της εντολής 30. Με τον ερχομό στη NEXT, ξαναεπιστρέφει ο έλεγχος στην 20, το I γίνεται 3, εκτελείται για τρίτη φορά η εντολή 30 κ.ο.κ. Όταν η τιμή του I γίνει 100, τότε η εντολή 30 θα εκτελεσθεί για έκατοστή φορά και ο έλεγχος θα οδηγήσει έξω από την επανάληψη, όπου τυπώνεται το SUM (εντολή 50) και το πρόγραμμα τελειώνει.

Όστε με τις εντολές FOR-NEXT μπορούμε σε αρκετές περιπτώσεις να ελέγξουμε και το τέλος ενός προγράμματος.

Δίνονται μερικά ακόμη παραδείγματα:

Παράδειγμα 1.

Να γραφεί πρόγραμμα που να υπολογίζει την τιμή της παραστάσεως:

$$\prod_{v=1}^6 (x^v + a) \quad \text{για } x = 1.8 \text{ και } a = 3.4.$$

```

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.

Να 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 αλγεβρα χρησιμοποιούμε γράμματα με ένα ή δύο μικρούς αριθμητικούς δείκτες στο κάτω δεξιό μέρος του γράμματος. Π.χ. συνηθίζομε να γράφουμε τη γενική μορφή ενός πολυωνύμου ως εξής:

$$a_0 x^0 + a_1 x^1 + \dots + a_v x^v$$

όπου: τους συντελεστές του τους συμβολίζουμε με το ίδιο γράμμα, το a , αλλά με ένα δείκτη διαφορετικό ώστε να διακρίνονται μεταξύ τους. Η γραφή αυτή χρησιμοποιήθηκε αρχικά, όταν ήταν τόσες πολλές οι παράμετροι η οι συντελεστές, που δεν έφθαναν τα γράμματα του αλφαριθμητικού για να τους παραστήσουμε με αυτά. Καθιερώθηκε όμως και γενικεύθηκε, γιατί είναι πιο απλή, εύχρηστη και διευκολύνει το γενικό συμβολισμό μιας παραστάσεως.

Γράμματα με δείκτη χρησιμοποιούνται επίσης για την παράσταση ομοειδών δεδομένων στη στατιστική. Έτσι, όταν π.χ. μετρούμε τα ύψη 10 διαφορετικών ατόμων, αντί να τα συμβολίσουμε με διαφορετικό γράμμα το καθένα, a , b , c ..., χρησιμοποιούμε το συμβολισμό με δείκτη, π.χ. x_1, x_2, \dots, x_{10} όπου με x_i παριστάνομε το 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 στήν οποία εμφανίζονται ένας ή δύο αριθμητικοί δείκτες (με τιμή φυσικούς αριθμούς), χωρίζομενοι με κόμμα. Π.χ.:

A(12), B(4), X(1,2), Z(76)

Αντί για αριθμητικές σταθερές, σαν δείκτης μπορεί να εμφανίζεται και μια μεταβλητή ή ακόμη και μια απλή έκφραση BASIC γνωστής ακέραιης 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. Τα 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)

ανήκουν 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)

ανήκουν στους δισδιάστατους πίνακες A, R και X. Στην περίπτωση των δισδιαστάτων πινάκων, μπορούμε να φαντασθούμε τα στοιχεία τους διατεταγμένα επάνω στο επίπεδο σε γραμμές και στήλες. Ο πρώτος αριθμητικός δείκτης συμβολίζει τότε τον αύξοντα αριθμό γραμμής και ο δεύτερος τον αύξοντα αριθμό στήλης, στη διασταύρωση 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 διασταύρωση της δεύτερης γραμμής (πρώτος δείκτης = 2) και της τρίτης στήλης (δεύτερος δείκτης = 3).

Η επεξεργασία ομοειδών δεδομένων διευκολύνεται πάρα πολύ με την εξομίσωσή τους προς τα στοιχεία ενός πίνακα και με χρήση των εντολών FOR-NEXT. Μερικά τέτοια παραδείγματα θα δούμε στη συνέχεια:

9.12 Εντολή DIM.

Στις περιπτώσεις που σε ένα πρόγραμμα χρησιμοποιούμε κάποιο πίνακα, είναι απαραίτητο να δηλώσομε στην αρχή, με τη βοήθεια της εντολής DIM, τον αριθμό των στοιχείων από τα οποία αποτελείται. Η δομή της είναι:

η DIM μεταβλητή (αριθμητικός δείκτης), μεταβλητή (αριθμητικός δείκτης),... π.χ.

10 DIM A(10), G(20)

που σημαίνει ότι μέσα στο πρόγραμμα χρησιμοποιείται η μεταβλητή A για 10 στοιχεία (που αποτελούν πίνακα), με δείκτη που παίρνει τιμές από 1-10 και η 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 καταχωρίσει τα στοιχεία κάθε πίνακα.

9.13 Παραδείγματα.

Παράδειγμα 1.

Έστω ότι θέλομε να προσθέσουμε 17 αριθμούς a_1, a_2, \dots, a_{17} , και να τυπώσουμε το άθροισμά τους.

Το αντίστοιχο πρόγραμμα θα 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
9999 END

```

Παράδειγμα 2.

Να βρεθεί ο μέσος όρος των αριθμών $a_1, a_2, a_3, a_4, a_5, a_6, a_7$. Το πρόγραμμα θα έχει την εξής 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
9999 END

```

9.14 Επεξεργασία αρχείων στη BASIC.

Η BASIC χρησιμοποιείται όχι μόνο για την επίλυση απλών μαθηματικών προ-

βλημάτων, αλλά ακόμη και για την απλή επεξεργασία αρχείων. Με τον όρο επεξεργασία εννοούμε 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) Εγγραφή πληροφοριών σε υπάρχον αρχείο.
- ε) Ανάγνωση πληροφοριών από υπάρχον αρχείο.

Θα δούμε αναλυτικά τις διάφορες φάσεις επεξεργασίας αρχείων. Ένα αρχείο πριν από τη χρησιμοποίησή του πρέπει να «ανοιχθεί», δηλαδή να προετοιμασθεί, ενεργοποιούμενο από τη μονάδα ελέγχου, πριν αρχίσει η μεταφορά των πληροφοριών από και προς τη μνήμη.

Τη σπηλιή αυτή θα πρέπει να καθορισθεί κατά πόσον από το αρχείο αυτό θα διαβάζονται πληροφορίες, δηλαδή θα είναι ένα αρχείο εισόδου (Input File) ή θα γράφονται σ' αυτό πληροφορίες, δηλαδή θα είναι ένα αρχείο εξόδου (Output File).

Στην περίπτωση που ανοίγομε-ενεργοποιούμε ένα αρχείο, χρησιμοποιούμε τις εντολές:

n OPEN u,f INPUT
n OPEN u,f OUTPUT

Για κάθε μία από τις περιπτώσεις χρήσεως του αρχείου που αναφέρθηκαν και στα δύο είδη εντολών, το u είναι μια αριθμητική σταθερή ή μεταβλητή και παριστάνει τη συμβολική ονομασία που δίνομε στο αρχείο και με την οποία θά το καλούμε μέσα στο πρόγραμμά μας, ενώ το f είναι μια άλλη σταθερή ή μεταβλητή, που αποτελεί την πραγματική ονομασία του αρχείου. Π.χ.:

OPEN 5, A1 INPUT

Με την εντολή αυτή ανοίγομε το αρχείο, που το πραγματικό του όνομα είναι A1 και που θα χρησιμοποιηθεί στο πρόγραμμά μας με τη συμβολική ονομασία 5.

Οι εντολές αναγνώσεως και εγγραφής ενός αρχείου είναι παρόμοιες με τις εντολές INPUT και PRINT, που είδαμε στις παραγράφους 8.6.2 και 8.6.3, με τη διαφορά ότι αναφέρεται σ' αυτές και το όνομα του αρχείου. Π.χ.

90 INPUT: CLS: A,B

Διαβάζονται από το αρχείο CLS, δύο τιμές-δεδομένα για το A και B αντίστοιχα. Επίσης:

100 PRINT: CS: A,B

Εγγράφονται στο αρχείο με όνομα CLS, οι τιμές των μεταβλητών A και B. Σε μερικά συστήματα υπολογιστών, αντί της INPUT, χρησιμοποιείται η GET και αντί της PRINT, η PUT ως εξής:

90 GET S: A,B
180 PUT 5: A,B

9.15 Υποπρογράμματα (Subprograms).

Υποπρόγραμμα είναι ένα σύνολο από εντολές, που εκτελούν μια καθορισμένη εργασία. Με τη μορφή αυτή χρησιμοποιείται σαν βιοηθητικό πρόγραμμα σε διάφορα σημεία ενός κυρίου προγράμματος (Main program). Τα υποπρογράμματα στη BASIC τα διακρίνομε σε δύο κατηγορίες:

- α) Συναρτήσεις (Function Subprograms).
- β) Υπορρουτίνες (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 μιας 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 εκφράσεως SIN (X/5+3). Η μεταβλητή X στην παρένθεση ονομάζεται παράμετρος (argument) και χρησιμοποιείται σαν ανεξάρτητη μεταβλητή μιας αλγεβρικής συναρτήσεως.

Αφού ορίσθηκε η FNS, μπορεί να χρησιμοποιηθεί στο πρόγραμμα. Έτσι, αν συναντήσουμε 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.

Να γραφεί συνάρτηση, που να υπολογίζει το εμβαδόν ενός ορθογωνίου τριγώνου από τις κάθετες πλευρές του, α και β.

10 DEF FNA (A,B) = 1/2*A*B

β) Υπορρουτίνες (Subroutines).

Η υπορρουτίνα είναι και αυτή ένα υποπρόγραμμα, αλλά η δομή και ο τρόπος χρήσεώς της σε ένα πρόγραμμα είναι τελείως διαφορετική από ό,τι μιας συναρτήσεως.

Σε μια υπορρουτίνα (που αποτελείται από περισσότερες από μία εντολές) μπορούμε να χρησιμοποιήσουμε τις περισσότερες εντολές στη BASIC, ενώ, αντίθετα, σε μια συνάρτηση, μόνο μία εντολή αντικαταστάσεως.

Η υπορρουτίνα γράφεται στο κάτω μέρος του κυρίως προγράμματος και δεν μπορεί να χρησιμοποιηθεί σαν όρος μέσα σε μια έκφραση BASIC, όπως μία συ-

νάρτηση, αλλά καλείται με μια ειδική εντολή, την GOSUB, στο κατάλληλο σημείο του προγράμματος. Η εντολή κλήσεως υπορρουτίνας έχει τη μορφή:

n GOSUB m

όπου m φυσικός αριθμός που αποτελεί τον αριθμό της πρώτης εντολής της υπορρουτίνας. Π.χ.:

100 GOSUB 460

και σημαίνει ότι θα μεταφέρθει από το σημείο αυτό ο έλεγχος του προγράμματος στην εντολή 460, που είναι και η πρώτη εντολή της υπορρουτίνας, που χρησιμοποιείται στο πρόγραμμα.

Η τελευταία εντολή μιας υπορρουτίνας είναι η:

n RETURN

Παράδειγμα.

Σαν παράδειγμα, θα υπολογίσουμε το πλήθος των συνδυασμών των N πραγμάτων υπότιμα J, χρησιμοποιώντας τη σχέση:

$$\binom{N}{J} = \frac{N!}{J!(N-J)!} \quad \text{Δεχόμαστε ότι } 0! = 1$$

όπου $N! = 1 \times 2 \dots (N-1) \times N$ και το διαβάζομε «N παραγοντικό»

$J! = 1 \times 2 \dots (J-1) \times J$ και το διαβάζομε «J παραγοντικό»

$(N-J)! = 1 \times 2 \dots (N-J+1) \times (N-J)$ και το διαβάζομε «N-J παραγοντικό»

Το πρόγραμμα θα έχει την εξής 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

```

Ασκήσεις.

1. Κωδικοποιείστε στη BASIC τις εξής μαθηματικές εκφράσεις:

a) $2\pi^2$

b) $a+x[\beta+v(\gamma+\delta v)]$

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) $\frac{ax^2 + \beta x + \gamma}{\epsilon^2 - 4\delta}$

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) $\frac{1 + \sigma\nu\psi}{1 - \eta\mu\psi}$

β) $|x|\beta^x + \frac{\alpha}{\lambda\psi x}$

γ) $2 - x^2 + \sqrt{e^x \lambda\psi x}$

δ) $4\sqrt{x} - 5 \eta\mu x + 6e^x$

ε) $x\eta\mu x + \sqrt{\alpha^2 + \beta^2}$

ζ) $\frac{1 + e^x}{1 - e^x}$

η) $\pi \mu^2 x + \sin^2 \psi$

θ) $\alpha \log x + (\beta \log \psi)^2$

3. Να γράψετε πρόγραμμα BASIC που να υπολογίζει την περίμετρο, τη διαγώνιο, το εμβαδό και την ακτίνα του εγγεγραμμένου κύκλου, ενός τετραγώνου πλευράς α (εφαρμογή με $\alpha = 1.5$).

4. Να γράψετε πρόγραμμα BASIC, για τον υπολογισμό του εμβαδού ενός τριγώνου, όταν είναι γνωστά τα μήκη των πλευρών του α,β,γ. Χρησιμοποιείστε τον τύπο:

$$E = \sqrt{t(t-\alpha)(t-\beta)(t-\gamma)} \text{ σπου}$$

$$T = \frac{\alpha+\beta+\gamma}{2} \quad \text{η ημιπερίμετρος του τριγώνου.}$$

5. Να γραφεί πρόγραμμα που να υπολογίζει το μικρότερο από 3 αριθμούς α,β,γ.

6. Να γραφεί πρόγραμμα που να υπολογίζει την τιμή της παραστάσεως

$$\sum_{v=1}^{12} \frac{x^v}{v} \quad (\text{εφαρμογή για } x = 0.8)$$

7. Να γραφεί υποπρόγραμμα, που να υπολογίζει το εμβαδόν ενός τριγώνου από τις πλευρές του και ώστερα να χρησιμοποιηθεί μέσα σε ένα κυρίως πρόγραμμα, για τον υπολογισμό του εμβαδού της επιφάνειας ενός τετραέδρου, του οποίου δύνονται τα μήκη των ακμών α,β,γ,δ,ε,ζ.

8. Να γραφεί πρόγραμμα BASIC που να υπολογίζει την τιμή της παραστάσεως.

$$\prod_{v=1}^{10} \frac{a^v}{v} \quad \text{για } a = 1.2$$

9. Να γραφεί πρόγραμμα BASIC που να υπολογίζει το γινόμενο των 20 πρώτων φυσικών περιπτών αριθμών.

10. Να γραφεί συνάρτηση για τον υπολογισμό της $\psi = x^3 - 3x^2 + 5$



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 η χρήση των μαθηματικών. Παρουσιάζει αρκετές ομοιότητες με τη BASIC, που γνωρίσαμε στο προηγούμενο κεφάλαιο. Η ονομασία της προέρχεται από τα αρχικά των λέξεων FORμulae 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 εντολές 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 διακρίνομε σε ακέραιες (Integer) και κλασματικές ή πραγματικές (Real). Ακέραιη σταθερή (Integer constant) είναι κάθε ακέραιος δεκαδικός αριθμός (αρνητικός 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 υπαρξη εκθέτη συμβολίζομε με το γράμμα E, που συνοδεύεται από κάποιο διψήφιο ακέραιο προσημασμένο 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})

Το πλήθος των ψηφίων που μπορεί να έχει μια ακέραια ή πραγματική σταθερή, εξαρτάται από τον τύπο του υπολογιστή.

Τις μεταβλητές τις διακρίνομε σε δύο κατηγορίες: α) Απλές μεταβλητές και β) μεταβλητές με δείκτη. Η ονομασία μιας μεταβλητής (Variable name) μπορεί να πε-

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 το K
MOSF	Σωστή
EQUATION	Λάθος, γιατί περιέχει περισσότερους από 8 χαρακτήρες
BAT+8	Λάθος, γιατί περιέχεται ο ειδικός χαρακτήρας +

β) Σύμβολα αριθμητικών πράξεων.

Όπως και στη BASIC, χρησιμοποιούμε και εδώ τα εξής σύμβολα για τις αριθμητικές πράξεις:

ΠΙΝΑΚΑΣ 10.2.3.

Τα σύμβολα των αριθμητικών πράξεων

Σύμβολο	Πράξη
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
**	Υψωση σε δύναμη

Ποτέ δεν πρέπει να γράφομε δύο σύμβολα το ένα δίπλα στο άλλο, π.χ. η διάταξη $A+*B$ είναι λάθος.

10.3 Εκφράσεις FORTRAN (FORTRAN expressions).

Οι εκφράσεις FORTRAN είναι αλγεβρικές 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, δύον αφορά στη χρήση ακεραίων ή πραγματικών μεταβλητών και σταθερών.

Με τους κανόνες αυτούς μπορούμε να κωδικοποιήσομε μια αλγεβρική 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 εκφρασθεί μια παράσταση, χαρακτηρίζεται και αυτή σαν ακέραιη ή πραγματική.

α) Μερική ανάμιξη.

Στη μερική ανάμιξη επιτρέπεται να υπάρχουν σε μία σχέση, μεταβλητές και σταθερές μόνο του ίδιου τύπου. Στους παρακάτω πίνακες δίνονται οι επιτρεπόμενοι συνδυασμοί ανάμεσα σε δύο όρους μιας απλής εκφράσεως A και B, καθώς και ο τύπος του αποτελέσματός. Με το I και το R παριστάνομε ένα ακέραιο ή πραγματικό όρο αντίστοιχα.

ΠΙΝΑΚΑΣ 10.3.1.

Οι επιτρεπόμενοι συνδυασμοί στη μερική ανάμιξη

A	B	I	R
I		I	-
R		-	R

Για τις πράξεις $+, -, /$
ανάμεσα στο A και το B

A	B	I	R
I		I	-
R		R	R

Για την ύψωση σε δύναμη ανάμεσα
στο A (βάση) και το B (εκθέτη)

Παραδείγματα κωδικοποίησεως στη μερική ανάμιξη (Πίνακας 10.3.2)

ΠΙΝΑΚΑΣ 10.3.2.

Παραδείγματα εκφράσεων κωδικοποιημένων στη μερική ανάμιξη

Αλγεβρική παράσταση	Έκφραση FORTRAN
$\alpha + \beta + \gamma$ $2x + 4$ $x^2 + \frac{6}{4}$ $\beta^2 - 4\alpha\gamma$ $\frac{3\alpha\beta}{5}$	$A + B + C$ $2. * X + 4.$ $X**2 + 6./ 4.$ $B ** 2 - 4. * A * C$ $3. * A * B / 5.$

Όπως βλέπομε, μια ακέραιη σταθερή, που εμφανίζεται σε μια έκφραση, μπορούμε να τη μετατρέψουμε σε πραγματική, προσθέτοντας απλώς μια τελεία. Η αντίστροφη ενέργεια δεν μπορεί να γίνει.

β) Πλήρη ανάμιξη.

Στην πλήρη ανάμιξη είναι δεκτός κάθε συνδυασμός. Ο τύπος του αποτελέσματος φαίνεται στον Πίνακα 10.3.3.

ΠΙΝΑΚΑΣ 10.3.3.*Οι επιτρέπονται συνδυασμοί στην πλήρη ανάμεξη*

A	B	I	R
I		I	R
R		R	R

Για όλες τις πράξεις ανάμεσα στο A και το B (+,-,*,/,**)

Παραδείγματα πωδικοποίησεως σε πλήρη ανάμεξη (Πίνακας 10.3.4).**ΠΙΝΑΚΑΣ 10.3.4.***Παραδείγματα εκφράσεων πωδικοποιημένων στην πλήρη ανάμεξη*

Αλγεβρική παράσταση	Εκφραση FORTRAN
$2x^2 + 4x - 3$	$2 * X ** 2 + 4 * X - 3$
$a + \frac{3}{\beta} - k$.Ακέραιη Πραγματική Ακέραιη Πραγματική Ακέραιη 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 έλαφρά, όπως βλέπουμε και στον Πίνακα 10.4.1.

ΠΙΝΑΚΑΣ 10.4.1.*Οι συναρτήσεις συστήματος της FORTRAN*

Ονομασία συναρτήσεως	Λειτουργία
SIN(X)	Υπολογίζει το ημίτονο του X
COS(X)	Υπολογίζει το συνημίτονο του X
ATAN(X)	Υπολογίζει το τόξο εφαπτομένης 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 ακέραιο N σε κλασματικό με προσθήκη μιας τελείας
IFIX(X)	Μετατρέπει τον κλασματικό X σε ακέραιο

Οι συναρτήσεις αυτές, που είναι όλες τύπου Real, αναφέρονται σαν απλοί όροι μέσα σε μία έκφραση FORTRAN. Παραδείγματα δίνονται στον Πίνακα 10.4.2.

ΠΙΝΑΚΑΣ 10.4.2.

Παραδείγματα κωδικοποιήσεων εκφράσεων που περιέχουν συναρτήσεις.

Αλγεβρική παράσταση	Έκφραση FORTRAN
$\begin{aligned} & \left(\frac{4}{3} \cdot \pi r^3 \right) \cdot e^x \\ & \eta \rho x + 2 \sin x + 4 \\ & \sqrt{x^2 - 1} + x \\ & a + e^x \\ & 2 \log x + \beta e^{-x} \\ & \eta \mu^2 x + \sin^2 x \end{aligned}$	$\begin{aligned} & (4 / 3 * 3.1415 * R ** 3) * EXP(X) \\ & SIN(X) + 2. * COS(X) + 4. \\ & SQRT(X**2 - 1.) + ABS(X) \\ & A + EXP(X) \\ & 2. * ALOG10(x) + B * EXP(-X) \\ & SIN(X) ** 2 + COS(X) ** 2 \end{aligned}$

10.5 Παρενθέσεις.

Χρησιμοποιούνται και εδώ με τον ίδιο ακριβώς τρόπο, που τις χρησιμοποιήσαμε και στη BASIC.

10.6 Ιεραρχία στην εκτέλεση των πράξεων.

Ισχύουν τα όσα αναφέραμε στη BASIC. Ακολουθείται δηλαδή η εξής σειρά στην εκτέλεση των πράξεων σε μία έκφραση FORTRAN:

- α) Υψώσεις σε δύναμη.
- β) Πολλαπλασιασμοί και διαιρέσεις.
- γ) Προσθήσεις και αφαιρέσεις.

Αν υπάρχουν παρενθέσεις, υπολογίζονται πρώτα οι παραστάσεις που περιέχονται σ' αυτές, με σειρά υπολογισμού πρώτα των εσωτερικών και ύστερα των εξωτερικών.

10.7 Λογικές εκφράσεις.

α) Απλές λογικές εκφράσεις.

Στη FORTRAN, όπως και στη BASIC, μπορούμε να κωδικοποιήσουμε με ειδικά σύμβολα, λογικές σχέσεις μεταξύ δύο όρων. Τα σύμβολα αυτά βλέπομε στον Πίνακα 10.7.1.

Σημειώνομε ότι κατά την κωδικοποίηση μιας απλής λογικής σχέσεως οι δύο όροι θα πρέπει να είναι του ίδιου τύπου, πραγματική-πραγματική ή ακέραιη-ακέραιη (Real-Real, Integer-Integer) εφ' όσον χρησιμόποιούμε, όπως είπαμε, τη μερική ανάμιξη.

ΠΙΝΑΚΑΣ 10.7.1.**Τα σύμβολα λογικών σχέσεων της FORTRAN**

Μαθηματικό λογικό σύμβολο	Λογικό σύμβολο FORTRAN	Επεξήγηση
=	.EQ.	ίσο προς
≠	.NE.	άνισο
<	.LT.	μικρότερο από
≤	.LE.	μικρότερο προς ή ίσο
>	.GT.	μεγαλύτερο από
≥	.GE.	μεγαλύτερο ή ίσο

Παραδείγματα κωδικοποίησεως απλών λογικών σχέσεων (Πίνακας 10.7.2).**ΠΙΝΑΚΑΣ 10.7.2.****Παραδείγματα κωδικοποίησεως απλών λογικών σχέσεων**

Μαθηματική λογική σχέση	Λογική έκφραση FORTRAN
$\alpha < \beta$ $x_1 \neq 4$ $\psi + 1 \geq z^2$ $3 < z$ $\frac{\alpha}{\beta} \neq \frac{\gamma}{\delta}$	A. LT. B X (1) . NE . 4. (Y + 1.) . GE . Z** 2 3.. LE. Z (A / B) . NE . (C / D)

β) Σύνθετες λογικές εκφράσεις.

Πολλές φορές στα μαθηματικά, μια λογική σχέση είναι σύνθετη, αποτελούμενη από απλές, που συνδέουν περισσότερους από δύο όρους, π.χ. $\alpha < \beta < \gamma$, $4 \leq x < 3$, κλπ. Τις σχέσεις αυτές μπορούμε να τις διασπάσουμε σε απλές, π.χ. την $\alpha < \beta < \gamma$ στις $\alpha < \beta$ και $\beta < \gamma$. Ακόμη συναντούμε περιπτώσεις, όπου πρέπει να ικανοποιούνται συγχρόνως δύο ή περισσότερες, ξεχωριστές, απλές ή σύνθετες λογικές σχέσεις, π.χ. λέμε ότι κάποιο σύστημα έχει λύση, αν $x \leq 0$ και $y > 0$, ή ακόμη ότι μία εξισώση έχει πραγματικές ρίζες, όταν $x \leq a$ ή $\psi < \beta < 0$ κ.ο.κ. Το if και το and συνδέουν τις ξεχωριστές συνθήκες μεταξύ τους. Στη FORTRAN χρησιμοποιούνται για τον ίδιο σκοπό τα λεγόμενα λογικά σύμβολα συσχετίσεως . OR . και . AND .

Παραδείγματα κωδικοποίησεως συνθέτων λογικών σχέσεων βλέπομε στον Πίνακα (10.7.3).

ΠΙΝΑΚΑΣ 10.7.3.*Παραδείγματα κωδικοποίησεως συνθέτων λογικών σχέσεων*

Μαθηματική λογική σχέση	Λογική έκφραση FORTRAN
$0 < x < 3$ $a \neq 0$ και $\beta < \gamma$ $x = \psi \quad \text{ή} \quad z = 2$ $x > a \quad \text{ή} \quad \psi < \beta < \gamma$	$0..LE.X AND. X. LT. 3.$ $A. NE. O.. AND. B. LT. C$ $X. EQ. Y. OR. Z. EQ. 2.$ $X. GT. A. OR. Y. LT. B. AND. B. LT. C$

10.8 Κατηγορίες εντολών FORTRAN.**Διακρίνομε τις έξις κατηγορίες εντολών FORTRAN:**

- Εντολές αντικαταστάσεως.
- Εντολές ελέγχου ή διακλαδώσεως.
- Εντολές εισόδου-εξόδου.
- Δηλωτικές εντολές.

10.9 Περιγραφή του εντύπου κωδικογραφήσεως FORTRAN (FORTRAN Coding Form)

Τις εντολές, ενδός προγράμματος FORTRAN τις γράφομε σε ένα ειδικό έντυπο, το φύλλο κωδικογραφήσεως FORTRAN (FORTRAN Coding Form σχ. 10.9), με τη σειρά που θα εκτελεσθούν.

Το φύλλο αυτό περιλαμβάνει ορισμένο αριθμό γραμμών. Κάθε γραμμή χωρίζεται σε 80 στήλες, που αριθμούνται από το 1-80. Σε κάθε στήλη-θέση να καταχωρίζεται και ένας χαρακτήρας. Σημειώνομε εδώ ότι και το κόμμα (,), η τελεία (.), το ίσον (=) και όλοι οι ιδιοί χαρακτήρες, καταχωρίζονται σε ξεχωριστή στήλη του εντύπου. Το περιεχόμενο μιας γραμμής του εντύπου μεταφέρεται και διατρυπά σε ένα 80-στηλο δελτίο.

Στις στήλες 1-5 γράφομε τον αριθμό εντολής. Δεν είναι υποχρεωτική η αρίθμηση κάθε εντολής, όπως στη BASIC. Μπορούμε να χρησιμοποιούμε αριθμούς μέχρι πενταψήφιους, χωρίς διάκριση και χωρίς να βρίσκονται σε οποιαδήποτε σειρά (στη BASIC πρέπει να είναι σε αύξουσα σειρά).

Η στήλη 6 συμπληρώνεται με ένα χαρακτήρα, όταν η γραμμή, στην οποία ανήκει, αποτελεί συνέχεια της προηγούμενης εντολής (Continuation), που επειδή ήταν μεγάλη, δεν μπόρεσε να χωρέσει σε μία γραμμή.

Στις στήλες 7-72 καταχωρίζομε την εντολή (ξεκινώντας από τη στήλη 7).

Οι στήλες 73-80 χρησιμοποιούνται για την αναγραφή της κωδικής ονομασίας του προγράμματος (αγνοούνται κατά τη φάση της μεταφράσεως του προγράμματος).

10.10 Αριθμητική εντολή αντικαταστάσεως.

Είναι η πιο συνηθισμένη από τις εντολές της FORTRAN, αντίστοιχη προς την

ΠΡΟΓΡΑΜΜΑ	FORTRAN CODING FORM	ΗΜΕΡΟΜΗΝΙΑ.....
ΠΡΟΓΡΑΜΜΑΤΙΣΤΗΣ		
ΣΕΛΙΔΑ		ΑΓΩ
ΣΤΗΝ Η ΓΙΑ ΣΧΟΛΑ	ΕΝΤΟΛΗ FORTRAN	
ΑΡΙΘΜΟΣ ΕΓΓΥΗΤΗΣ	Σελ. 1	73
		80
		72

Σχ. 10.9.
Έντολο κωδικογράφησες FORTRAN.

εντολή 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 εκφράσεως του δευτέρου μέλους (πραγματική-πραγματική, ακέραιη-ακέραιη).

Η εντολή, λειτουργεί ως εξής: Υπολογίζεται πρώτα η αριθμητική τιμή της εκφράσεως του δευτέρου μέλους και τοποθετείται ύστερα σαν περιεχόμενο της μεταβλητής του πρώτου μέλους, αντικαθιστώντας κάθε προηγούμενο περιεχόμενό της. Φυσικά θα πρέπει όλοι οι όροι της εκφράσεως να έχουν γνωστή τιμή.

Στην εντολή π.χ.:

$$A = X ^\star 2 - 3.X + 4.$$

υπολογίζεται πρώτα η τιμή της $X^2 - 3X + 4$ και τοποθετείται κατόπιν στη μεταβλητή A, αντικαθιστώντας το προηγούμενο περιεχόμενό της. Η τιμή της X πρέπει να είναι γνωστή.

Παραδείγματα.

$$R = SQRT(Y^\star 2 - B^\star A^\star C) + 4.$$

$$P = 3.14159 + 4.^* SIN(X)$$

$$I = I + 3$$

$$X = A(1) + B(4) ^\star 2$$

$$P(I) = A(1,I) + B(J,2)$$

Όταν η μεταβλητή είναι ακέραιη και η έκφραση πραγματική, τότε στην ακέραιη μεταβλητή μεταφέρεται μόνο το ακέραιο μέρος από την υπολογιζόμενη τιμή της εκφράσεως. Π.χ.:

αν [X] = 1.4 και [Y] = 2.8 και εκτελεσθεί η εντολή

$$I = 2.^*X + Y$$

τότε το αποτέλεσμα θα είναι [I] = 5, αντί του σωστού 5.8.

Όταν η μεταβλητή είναι πραγματική και η έκφραση ακέραιη, τότε το αποτέλεσμα του υπολογισμού της εκφράσεως μετατρέπεται σε πραγματικό με την προσθήκη της κλασματικής τελείας (.) στο τέλος του αριθμού. Π.χ.:

αν [I] = 12 και [J] = 3 και εκτελεσθεί η εντολή

$$S = 3^\star I - J$$

τότε το αποτέλεσμα θα είναι [S] = 33.

10.11 Εντολές ελέγχου και διακλαδώσεως.

α) Εντολή GO TO χωρίς συνθήκη.

Η μορφή της εντολής είναι:

|| GO TO n

όπου ή φυσικός αριθμός, που παριστάνει τον αριθμό της εντολής, στην οποία μεταφέρεται ο έλεγχος του προγράμματος. Η διπλή γραμμή αριστερά συμβολίζει την δη στήλη του εντύπου κωδικογραφήσεως.

Παράδειγμα 1.

35	GO TO 35
.....
.....
.....
35	X = X + 4

Με την εκτέλεση της GO TO 35, σαν επόμενη εντολή θα εκτελεσθεί η $X = X + 4$ ένω όλες οι ενδιάμεσες αγνοούνται.

Παράδειγμα 2.

12	S = S + 4.*A
.....
.....
.....
.....
.....	GO TO 12

Αμέσως μετά την εκτέλεση της GO TO 12 θα εκτελεσθεί η $S = S + 4.*A$ και κατόπιν όλες οι επόμενες εντολές, που βρίσκονται κάτω από αυτή.

B) Εντολή GO TO με συνθήκη.

Η μορφή της εντολής αυτης είναι:

|| GO TO (n₁, n₂, n₃...), Ακέραιη μεταβλητή ή έκφραση,
όπου n₁, n₂... = αριθμοί εντολών.

Λειτουργεί ως εξής: Αν η αριθμητική τιμή της ακέραιης μεταβλητής ή έκφρασης είναι 1 ή 2, ή 3 κ.ο.κ. θα εκτελεσθεί σαν επόμενη η εντολή, με αριθμό των n₁, ή n₂ ή n₃ κ.ο.κ. Π.χ.:

34	X = X + B
.....
.....
.....
.....
42	GO TO (15, 25, 34, 42), I
.....
.....
.....
.....
25	X = X + B-A
.....
.....
.....
15	X = 2.*B
.....

Αν η τιμή του I τη σπιγμή που εκτελείται η εντολή GO TO είναι 1, τότε θα εκτελεσθεί αριστος μετά η εντολή 15 και θα παραλειφθούν όλες οι ενδιάμεσες. Αν η τιμή του I είναι 3, τότε θα εκτελεσθεί στη συνέχεια η 34 και κατόπιν όλες οι κάτω από την 34 εντολές κ.ο.κ.

γ) Αριθμητική εντολή IF.

Η δομή της εντολής είναι:

|| IF (μεταβλητή ή έκφραση FORTRAN) αριθ. εντολής 1, αριθ. εντολής 2, αριθ. εντολής 3.

Η εντολή IF λειτουργεί ως εξής: Ελέγχεται η τιμή της μεταβλητής ή της εκφράσεως FORTRAN μέσα στην παρένθεση. Αν είναι αρνητική (< 0), τότε σαν επόμενη θα εκτελεσθεί η εντολή με αριθμό, τον αριθμό εντολής 1. Αν η τιμή είναι 0, θα εκτελεσθεί η εντολή με αριθμό τον δεύτερο που εμφανίζεται στην εντολή και τέλος, αν η τιμή είναι θετική (> 0), θα εκτελεσθεί η εντολή με αριθμό τον τρίτο που εμφανίζεται στο δεξιό της εντολής IF.

Η αναγραφή και των τριών αριθμών στην εντολή IF είναι υποχρεωτική, διαχωρίζονται δε μεταξύ τους με κόμματα.

Παράδειγμα 1.

22	
	
IF (I + K) 22, 44, 33	
44	
33	
	

Αν η τιμή του $I + K$, τη σπιγμή που εκτελείται η IF, είναι < 0 , τότε θα εκτελεσθεί σαν επόμενη, η εντολή 22. Αν $I + K = 0$, τότε θα εκτελεσθεί σαν επόμενη εντολή η 44. Τέλος, αν $I + K > 0$, θα εκτελεσθεί σαν επόμενη εντολή η 33.

Παράδειγμα 2.

18	
	
IF (A) 21, 21, 18	
21	
	

Εδώ, αν η τιμή της μεταβλητής A είναι μικρότερη ή ίση με το μηδέν (< 0), τότε θα εκτελεσθεί η εντολή 21. Αν είναι μεγαλύτερη του μηδενός (> 0), θα εκτελεσθεί η εντολή 18.

Παράδειγμα 3.

52	IF (B**2—4.* A*C) 52, 51, 51
51

Αν η τιμή της εκφράσεως μέσα στην παρένθεση είναι < 0 , τότε θα εκτελεσθεί η εντολή 52, αν δε είναι ≥ 0 , τότε θα εκτελεσθεί η 51 με παράλειψη όλων των ενδιαμέσων εντολών.

δ) Λογική εντολή IF.

Έχει τη μορφή:

IF (λογική έκφραση FORTRAN) εντολή 1
εντολή 2

Η εντολή λειτουργεί ως εξής: Ελέγχεται η λογική έκφραση μέσα στη παρένθεση. Αν η συνθήκη που θέτει πληρούται, τότε εκτελείται η εντολή 1, που βρίσκεται έξω από την παρένθεση και το πρόγραμμα συνεχίζεται κανονικά. Αν η συνθήκη δεν ισχύει, τότε η εντολή 1 αγνοείται και σαν επόμενη εκτελείται η εντολή 2.

Η εντολή 1 δεν πρέπει να είναι μια άλλη εντολή IF (λογική ή αριθμητική) ή μία εντολή DO, που θα γνωρίσουμε στη συνέχεια.

Παράδειγμα 1.

44	IF(X. LE. 0.) GO TO 44
	X = X + 4.

	X = SQRT(5.)

Αν η συνθήκη μέσα στην παρένθεση ισχύει, δηλαδή αν $X < 0$, τότε ο έλεγχος θα μεταφερθεί στην εντολή 44 (αφού εκτελεσθεί η GO TO 44). Αν η συνθήκη μέσα στην παρένθεση δεν ισχύει, δηλαδή αν $X > 0$, τότε η εντολή GO TO 44 αγνοείται και εκτελείται σαν επόμενη η $X = X + 4$.

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 η $S = X + 5.*Y$ και κατόπιν η επόμενη, που είναι η $I = I + 1$. Αν όμως $X \neq 0$ και $Y = 0$ ή $X = 0$ και $Y \neq 0$, τότε η $S = X + 5.*Y$ θα αγνοηθεί και θα εκτελεσθεί σαν επόμενη εντολή η $I = I + 1$ και όλες οι επόμενες.

10.12 Εντολή DO.

Όπως είπαμε και στη BASIC, συχνά χρειάζεται να επαναλαμβάνεται η εκτέλεση ενός τμήματος του προγράμματος περισσότερες από μία φορές. Στις περιπτώσεις αυτές χρησιμοποιούμε μία αντίστοιχη προς τη FOR-NEXT εντολή της BASIC, τη DO που συνοδεύεται από την εντολή CONTINUE.

Η μορφή της είναι:

$n \parallel$ $\cdot \cdot \cdot \cdot \cdot$ $\cdot \cdot \cdot \cdot \cdot$ $\cdot \cdot \cdot \cdot \cdot$ $\cdot \cdot \cdot \cdot \cdot$ $n \parallel$ CONTINUE	$DO \ n \ \text{ακέραιη μεταβλητή} = k_1, k_2, k_3$ $\cdot \cdot \cdot \cdot \cdot$ $\cdot \cdot \cdot \cdot \cdot$ $\cdot \cdot \cdot \cdot \cdot$ $\cdot \cdot \cdot \cdot \cdot$ $\left. \begin{array}{c} \cdot \cdot \cdot \cdot \cdot \\ \cdot \cdot \cdot \cdot \cdot \end{array} \right\} \text{Εντολές του DO}$
---	--

n = Φυσικός αριθμός που παριστάνει τον αριθμό εντολής του CONTINUE. k_1, k_2, k_3 = Ακέραιες σταθερές ή μεταβλητές με τιμές μόνο θετικές.

Η πρώτη και η δεύτερη, (k_1 και k_2) αποτελούν αντίστοιχα την πρώτη και την τελευταία τιμή που μπορεί να πάρει η ακέραιη μεταβλητή (δείκτης του DO). Η τρίτη (k_3), αποτελεί το βήμα. Όταν λείπει, υποτίθεται ότι το βήμα = 1. Τα k_1, k_2, k_3 χωρίζονται μεταξύ τους με κόμματα.

Η ομάδα των εντολών που βρίσκεται μεταξύ των DO και CONTINUE θα εκτελεσθεί τόσες φορές, όσο είναι το πλήθος των τιμών που μπορεί να πάρει ο δείκτης του DO.

Παράδειγμα 1.

$IS = 0$ $DO \ 14 \ I = 1,3$ $IS = IS + I$ $14 \parallel$ CONTINUE $\cdot \cdot \cdot \cdot \cdot$
--

Εδώ ο δείκτης του DO, I μπορεί να πάρει τις τιμές 1, 2 και 3. Τό ότι λείπει η τρίτη σταθερή k_3 , σημαίνει ότι το βήμα είναι 1. Μέσα στο DO υπάρχει μόνο μία εντολή, η $IS = IS + I$, που θα εκτελεσθεί συνολικά τρεις φορές. Στο δεύτερο μέλος της $IS = IS + I$ συμμετέχει και ο δείκτης του DO με τις εκάστοτε τιμές του.

Ας δούμε τις διάφορες φάσεις εκτελέσεως των εντολών αυτών.

Στον πρώτο κύκλο εκτελέσεως, το $I = 1$ και η $IS = 0 + 1 = 1$. Μετά επανέρχεται, λόγω της CONTINUE, στη DO, όπου το I γίνεται 2. Εκτελείται για δεύτερη φο-

ρά η $IS = IS + I = 1 + 2 = 3$ και επιστρέφει πάλι στη DO όπου το I παίρνει την τελευταία τιμή 3, εκτελέσται η $IS = IS + I = 3 + 3 = 6$ και κατόπιν ο έλεγχος του προγράμματος βγαίνει έξω από το DO, για να συνεχισθεί η εκτέλεση των επομένων εντολών του προγράμματος.

Σημειώνομε ότι, όταν τελευταία εντολή του DO είναι μία εντολή αντικαταστάσεως ή εισόδου-εξόδου, τότε μπορούμε να μην αναφέρομε καθόλου την CONTINUE, το δε αριθμό εντολής της να τον θέσομε στην τελευταία αυτή εντολή του DO.

Παράδειγμα 2.

$$\begin{array}{l} DO \ 15 \quad I = 1,4 \\ \quad K(I) = 3*I - 2 \\ 15 \quad L(I) = I^{**}2 \end{array}$$

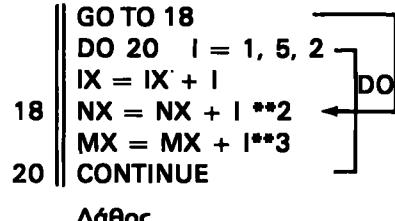
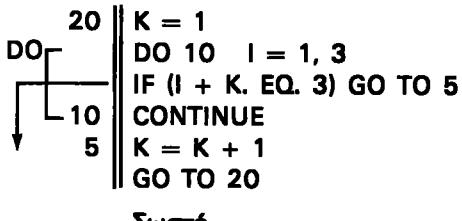
Στο παράδειγμα αυτό η εντολή CONTINUE δεν υπάρχει και ο αριθμός εντολής της έχει τεθεί στην τελευταία εντολή του DO, που είναι μία εντολή αντικαταστάσεως.

Μέσα στο DO υπάρχουν δύο εντολές: η $K(I) = 3*I - 2$ και η $L(I) = I^{**}2$, που θα εκτελεσθούν συνολικά 4 φορές η κάθε μία (όσες και οι τιμές που μπορεί να πάρει ο δείκτης του DO, I).

Βλέπομε ακόμη ότι ο δείκτης I δανείζει τις τιμές του τόσο στους δείκτες των μεταβλητών $K(I)$ και $L(I)$, οσο και στα δεύτερα μέλη των εντολών.

Στην πρώτη εκτέλεση το I έχει τιμή 1 και υπολογίζονται οι τιμές των στοιχείων $K(1) = 3*1 - 2 = 1$ και $L(1) = 1^{**}2 = 1$. Στη δεύτερη υπολογίζονται δύο νέα στοιχεία των πινάκων K και L, τα $K(2) = 3*2 - 2 = 4$ και $L(2) = 2^{**}2 = 4$. Στην τρίτη υπολογίζονται τα $K(3) = 3*3 - 2 = 7$ και $L(3) = 3^{**}2 = 9$. Στην τελευταία εκτέλεση υπολογίζονται τα $K(4) = 3*4 - 2 = 10$ και $L(4) = 4^{**}2 = 16$. Ο έλεγχος κατόπιν οδηγεί έξω από το DO, για να συνεχισθεί με τη σειρά, η εκτέλεση των υπολογίτων εντολών του προγράμματος.

Σημειώνομε ότι ενώ είναι δυνατή η έξοδος από ένα DO, για την εκτέλεση μιας άλλης εντολής, πριν συμπληρωθούν όλοι οι προβλεπόμενοι κύκλοι του (εξ αιτίας μιας εντολής διακλαδώσεως που ανήκει στην ομάδα εντολών του), είναι αδύνατη η είσοδος μέσα σε οποιαδήποτε εντολή του DO. Πχ.



10.13 Μερικές απλές εφαρμογές.

Παράδειγμα 1.

Να γραφεί πρόγραμμα που να υπολογίζει το άθροισμα των 20 πρώτων φυσι-

κών αριθμών 1, 2, 3..., 20.

Στην περίπτωση αυτή δεν χρειάζεται να εισαχθούν δεδομένα, γιατί τους φυσικούς αριθμούς μπορεί να μας τους δώσει ο δείκτης ενός DO, αν η αρχική τιμή του είναι 1, τελική η 20 και το βήμα 1.

Έτσι το πρόγραμμα θα έχει την εξής μορφή:

$$\begin{array}{l} \parallel IS = 0 \\ \parallel DO \quad 10 \quad I = 1,20 \\ 10 \parallel IS = IS + I \end{array}$$

IS είναι ο αθροιστής που στο τέλος θα περιέχει το ζητούμενο αθροισμα.

Παράδειγμα 2.

Να υπολογισθεί η τιμή της παραστάσεως:

$$\prod_{v=1}^{12} \frac{1}{v}$$

Εδώ το πρόγραμμα θα είναι:

$$\begin{array}{l} \parallel P = 1. \\ \parallel DO \quad 30 \quad N = 1,12 \\ 30 \parallel P = P*1./FLOAT(N) \end{array}$$

10.14 Εντολές εισόδου-εξόδου.

α) Εντολή READ.

Με τη βοήθειά της μπορούν να διαβάζονται δεδομένα από κάποια περιφερειακή μονάδα (συνήθως αναγνωστική δελτίων). Η μορφή της είναι:

$$\parallel READ (i, n) \text{ μεταβλητή, μεταβλητή, ...}$$

Δηλαδή αποτελείται από την κωδική λέξη READ και μια παρένθεση, μέσα στην οποία γράφομε δύο φυσικούς αριθμούς, χωρισμένους με κόμμα. Ο πρώτος, που εδώ τον συμβολίσαμε με i, είναι ο κωδικός αριθμός της περιφερειακής μονάδας, από όπου γίνεται η ανάγνωση. Ο κωδικός αυτός είναι ίδιος για μια συγκεκριμένη περιφερειακή μονάδα και αλλάζει με τον τύπο της μονάδας καθώς και από κατασκευαστή σε κατασκευαστή. Για την αναγνωστική δελτίων θα χρησιμοποιήσουμε το 5.

Ο δεύτερος αριθμός που είναι ο αριθμός εντολής FORMAT. Με την εντολή αυτή, όπως θα δούμε στη συνέχεια, περιγράφομε τη μορφή, με την οποία είναι καταχωρισμένα τα δεδομένα επάνω στα δελτία.

Οι μεταβλητές που εμφανίζονται έξω από τη παρένθεση μπορεί να είναι απλές ή με δείκτη, ακέραιες ή πραγματικές. Θα δεχθούν σαν περιεχόμενο τις τιμές των αντιστοίχων δεδομένων που 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 εντολή 24 FORMAT, τις τιμές που θα πάρουν οι μεταβλητές A, B, C. Έτσι, το πρώτο δεδομένο που θα διαβασθεί θα τοποθετηθεί σαν περιεχόμενο του A, το δεύτερο σαν περιεχόμενο του B και το τρίτο σαν περιεχόμενο του C.

Παράδειγμα 2.

32	READ (5, 32) A(1), I
	FORMAT (.)

Εδώ διαβάζονται από την αναγνωστική δελτίων ($i = 5$) και με τρόπο που περιγράφεται στην εντολή 32 δύο τιμές, από τις οποίες η πρώτη θα αποτελέσει το περιεχόμενο της A(1) και η δεύτερη το περιεχόμενο της I.

β) Εντολή WRITE.

Με τη βοήθειά της γράφονται περιεχόμενα μεταβλητών σε κάποια περιφερειακή μονάδα, συνήθως στον εκτυπωτή. Η μορφή της είναι:

|| WRITE (i, n) μεταβλητή, μεταβλητή, ...

Ι είναι ένας φυσικός αριθμός, που συμβολίζει τον κωδικό της περιφερειακής μονάδας, όπου γίνεται η εγγραφή των τιμών. Άλλαζει από τύπο σε τύπο και από τη μία κατασκευάστρια εταιρεία στην άλλη. Εδώ θα χρησιμοποιήσουμε το $i = 6$ για εγγραφή στον εκτυπωτή. Ι είναι ο αριθμός FORMAT, που προσδιορίζει τη μορφή της εγγραφής. Έξω από την παρένθεση παρατίθενται οι μεταβλητές, απλές ή με δείκτη, το περιεχόμενο των οποίων θέλουμε να εγγραφεί στην περιφερειακή μονάδα.

Παράδειγμα 1.

4	WRITE (6,4) A, B
	FORMAT (.)

Θα γραφούν στον εκτυπωτή ($i = 6$) τα περιεχόμενα των πραγματικών μεταβλητών A και B, με τρόπο που περιγράφεται στην εντολή 4 FORMAT.

Παράδειγμα 2.

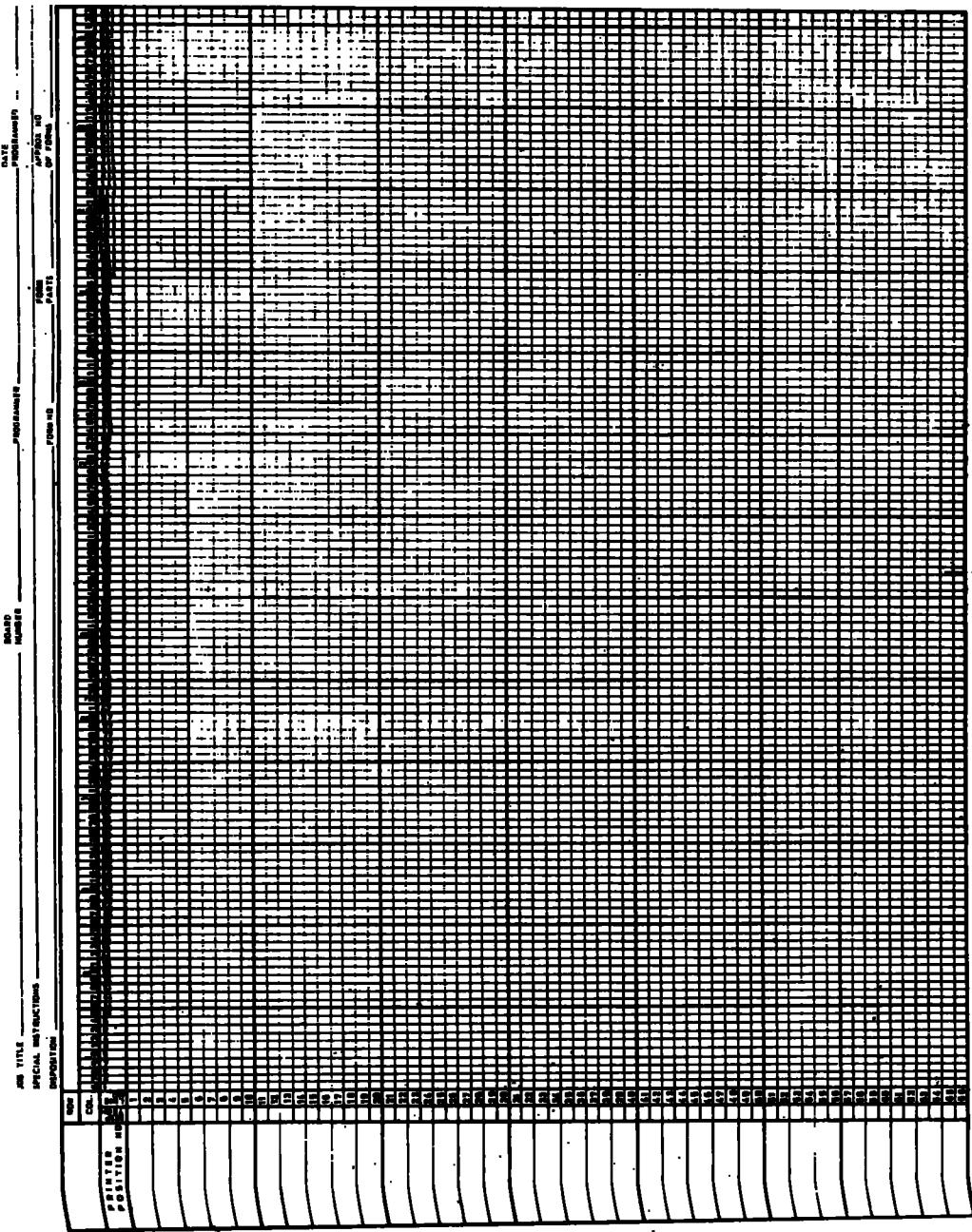
2	WRITE (6,2) C, B(12), D
	FORMAT (.)

Θα εκτυπωθούν οι τιμές-περιεχόμενα των μεταβλητών C, B(12) και D, με τρόπο που περιγράφεται στην εντολή 2 FORMAT.

10.15 Έντυπο σχεδίασμού εκτυπώσεων (Printer spacing chart).

Αποτελείται από ορισμένο αριθμό γραμμών (σχ. 10.15). Κάθε γραμμή αντι-

PRINTER SPACING CHART



Σχ. 10.15. Το έντυπο σχεδίασμού εκτυπώσεων (Printer Spacing Chart).

στοιχεί σε μία λογική ενότητα (Record) εκτυπώσεως και χωρίζεται σε στήλες. Ανάλογα με τον τύπο και τις δυνατότητες του εκτυπωτή μπορούν να εκτυπωθούν 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 περιγράφομε τον τρόπο που θα διαβασθούν ή θα εγγραφούν πληροφορίες από ή σε κάποια λογική ενότητα, όπου μπορούν να καταχωρισθούν περισσότερες από μία πληροφορίας, που μπορεί να είναι αριθμοί, αλφαριθμητικοί χαρακτήρες ή άκομη άλλες να είναι πραγματικές και άλλες ακέραιες.

Κάθε ξεχωριστή πληροφορία (πεδίο) μιας λογικής ενότητας περιγράφεται με κάπιο κωδικό FORMAT. Η μορφή της είναι:

`n || FORMAT (κωδικός, κωδικός,...)`

Οι κωδικοί που αναφέρομε 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	Που χρησιμοποιείται
Iw Fw,d Ew.d wX wH	Για ακέραια αριθμητικά πεδία (Integer) Για απλά πραγματικά αριθμητικά πεδία (Real) Για πραγματικά αριθμητικά πεδία με εκθέτη Για κενά μεσοδιαστήματα Για απλή εμφάνιση χαρακτήρων (μηνύματα).

Θα περιγράψουμε τώρα τη χρήση καθενός από τους παραπάνω κωδικούς.

Κωδικός Iw.

Χρησιμοποιείται για την ανάγνωση ή εγγραφή ενάς 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 ακραίες δεξιές θέσεις του πεδίου. Για το δελτίο του παραδείγματος που ακολουθεί θα δώσουμε FORMAT (I3, 14).

346 12											
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9

β) Για την εκτύπωση.

Για να γράψουμε το περιεχόμενο ενός ακέραιου πεδίου στον εκτυπωτή, πρέπει να έχουμε υπ' όψη μας ότι, αν δώσουμε στον κωδικό I σαν w ένα αριθμό μεγαλύτερο από το πλήθος των ψηφίων του αριθμού που πρόκειται να τυπωθεί, αυτός θα τυπωθεί στις ακραίες δεξιές θέσεις του πεδίου. Αν δώμας σαν w δώσουμε ένα αριθμό μικρότερο από το πλήθος των ψηφίων του αριθμού, τότε θα εκτυπωθούν αστερίσκοι σε όλο το μήκος του πεδίου. Έτσι δίνεται προειδοποίηση ότι πρέπει να αυξήσουμε την τιμή του w στον κωδικό I για να έχουμε σωστή εκτύπωση.

Παραδείγματα εκτυπώσεως με χρήση του κωδικού /w (Πίνακας 10.6.2).

ΠΙΝΑΚΑΣ 10.16.2.

Παραδείγματα εκτυπώσεως με χρήση του κωδικού /w

Αριθμός μέσα στη μνήμη	Κωδικός I	Θα τυπωθεί
548	13	5 4 8
28	14	2 8
4281	14	4 2 8 1
5048	13	* * *
892	15	8 9 2
4	13	4

Κωδικός wX.

Με τον κωδικό αυτό περιγράφομε τις κενές στήλες που παρεμβάλλονται μεταξύ δύο πεδίων στη λογική ενόπτητα εισόδου ή εξόδου. w είναι ένας αριθμός, που παριστάνει το πλήθος των κενών στηλών.

Παραδείγματα αναγνώσεως με χρήση του κωδικού wX.

ΔΕΛΤΙΑ ΔΕΔΟΜΕΝΩΝ

FORMAT ΑΝΑΓΝΩΣΕΩΣ

FORMAT (I4, 3X, I6)

FORMAT (I2, 1X, I3, 1X, I6, 2X, I2)

Παραδείγματα εκτυπώσεως με χρήση του κωδικού wX (Πίνακας 10.16.3).

ПИНАКАΣ 10.16.3.

Παραδείγματα εκπυκνώσεως με χρήση του πυκνού αλ

Κωδικός Fw.d.

Χρησιμοποιείται για την ανάγνωση ή εκτύπωση των περιεχομένων αριθμητικών πραγματικών πεδίων (Real). Τα περιεχόμενα βρίσκονται σε απλή κλασματική μορφή. Στον κωδικό αυτό, ως είναι ένας αριθμός, που παριστάνει το μέγεθος του πεδίου, ενώ ως είναι το πλήθος των κλασματικών ψηφίων του πεδίου.

α) Για την ανάγνωση.

Αν ο καταχωριζόμενος αριθμός έχει λιγότερα ωφέλια από τό μέγεθος του πεδίου, τότε τοποθετείται στις ακραίες δεξιές θέσεις.

Παραδείγματα αναγνώσεως δεδομένων με χρήση του κωδικού Fw.d.

8) Για την εκπύσωση.

Ειδικότερα στην εκτύπωση, αν το d είναι μικρότερο από το πλήθος των κλασματικών ψηφίων του αριθμού, τότε θα στρογγυλευθεί ο αριθμός αυτός σε πλήθος d κλασματικών ψηφίων. Αν το d είναι μεγαλύτερο, θα προστεθούν τόσα μηδενικά στο τέλος του αριθμού ώστε να έχουμε συνολικά πλήθος d κλασματικών ψηφίων.

Παραδείγματα εκπυσώσεως με χρήση του πινάκου F.w.d (Πίνακας 10.16.4).

DINAKAR 10-16-4.

Паробійната експедиція на хвіст тау туфти Р.н.д.

Kosmos Ew.d.

Χρησιμοποιείται για την ανάγνωση ή εκτύπωση περιεχομένων πραγματικών πεδίων (Real) σε εκθετική μορφή.

a) Για την ανάγνωση.

Όπως έχομε ήδη αναφέρει, ένας πραγματικός αριθμός μπορεί να γραφεί και σε εκθετική μορφή. Π.χ. ο αριθμός 0.147 μπορεί να γραφεί και 1.47×10^{-1} και, αν χρησιμοποιήσουμε τη κωδικοποίηση της FORTRAN σαν $1.47E-1$.

Αν λοιπόν ο αριθμός-δεδομένο έχει εκφρασθεί έτσι, τότε θα τον διαβάσουμε με τον κωδικό Ew.d, όπου w είναι το μέγεθος του πεδίου όπου έχει καταχωρισθεί ο αριθμός και d το πλήθος των κλασματικών υψηλών του. Π.χ.

1	.	7	5	6	E	+	1
---	---	---	---	---	---	---	---

$\overbrace{\hspace{1cm}}^d = 3$

$\overbrace{\hspace{7cm}}^W = 8$

E8.3

Παραδείγματα αναγνώσεως δεδομένων με χρήση του κωδικού Ew.d.

β) Για την εκτύπωση.

Ένας απλός πραγματικός αριθμός, που είναι καταχωρισμένος σε κάποια θέση της μνήμης, εκτυπώνεται με τη βοήθεια του κωδικού Ew.d στη μορφή που μας παρέχει η αριθμητική κινητής υποδιαστολής (Floating point arithmetic). Στην έκφραση αυτή θα πρέπει στους περισσότερους υπολογιστές να ικανοποιείται η σχέση $w > d + 7$.

Η μετατροπή του απλού πραγματικού σε πραγματικό με εκθέτη, γίνεται, στην περίπτωση αυτή, με μετακίνηση της υποδιαστολής προς τα δεξιά ή τα αριστερά τό-

σες Θέσεις, ώστε να βρεθεί μπροστά από το πρώτο μη μηδενικό σημαντικό¹ ψηφίο του αριθμού. Το πλήθος των θέσεων που μετακινήθηκε η υποδιαστολή, αποτελεί τον εκθέτη. Ο αριθμός που προκύπτει, στρογγυλεύεται σε τόσα κλασματικά ωφελία, όσα δείχνει ο d.

Παραδείγματα εκτυπώσεως με χρήση του πινάκου Ew.d (Πίνακας 10.16.5).

ПІНАКАЕ 10.16.б.

Παραδείγματα εκπυνώσεως με χρήση του πληροφορικού Ενδιάμεσου

Αριθμός μέσα στη μνήμη	FORMAT εκτυπώσεως	Θα εκτυπωθεί
1.756	E15.7	+ d. 1 7 5 6 0 0 0 E + 0 1
-0.0094	E13.4	+ -d. 0 9 4 0 0 E - 0 2
175.62	E11.3	+ 0 .-1 7 6 E + 0 3

Κωδικός αΗ

Χρησιμοποιείται κυρίως για εκτύπωση 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 θα τυπώσουμε. Π.χ, αν μέσα σε éva FORMAT γράψουμε:

11HTHE ROOT IS

Τότε στον εκτυπωτή θα γραφεί:

THE ROOT IS

Το ω στην περίπτωση αυτή ισούται με 11. Στον αριθμό αυτό συνυπολογίσαμε και τα κενά που παρεμβάλλαμε ανάμεσα στις λέξεις του μηνύματος.

10.17 Επαναλαμβανόμενοι κωδικοί FORMAT.

Συχνά, μέσα σε ένα FORMAT, ο ίδιος κωδικός εμφανίζεται πολλές φορές.
Π.χ.: 12, 12, 12, F3.1, 2X, I3, I3

Εδώ, ο κωδικός Ι2 εμφανίζεται για τρεις συνεχείς φορές και ο Ι3 δύο. Μπορούμε, λοιπόν, να συντομεύσουμε τη γραφή του παραπάνω FORMAT, ως εξής:

Αν σύμβαση το FORMAT

I2 E3 1 I2 2Y I3

1 Στο δεκαδικό σύστημα αποιοδήποτε από τα ψηφία 1,2,3,... 9 είναι σημαντικό. Το 0 είναι σημαντικό μόνο στις περιπτώσεις όπου βρίσκεται ανάμεσα σε δύο άλλα σημαντικά ψηφία, ή στο τέλος του αριθμού.

δεν θα μπορούσαμε να κάνομε οποιαδήποτε συντόμευση, γιατί οι όμοιοι κωδικοί 12 δεν βρίσκονται ο ένας δίπλα στον άλλο.

10.18 Δηλωτικές εντολές.

Οι εντολές αυτές δεν είναι εκτελέσιμες, γράφονται δε πάντα στην αρχή του προγράμματος, πριν από την πρώτη εκτελέσιμη εντολή. Οι πιο συνηθισμένες είναι:

α) Εντολή INTEGER.

Η εντολή αυτή έχει τη μορφή:

|| INTEGER μεταβλητή, μεταβλητή, ...

Όλες οι μεταβλητές, που δηλώνονται στην INTEGER, είναι πραγματικές. Με αυτήν μπορούμε να χρησιμοποιούμε πραγματικές μεταβλητές σε ένα πρόγραμμα σαν να ήταν ακέραιες. Π.χ.:

INTEGERX, A12, B

Οι μεταβλητές X, A12, B μετά τη δήλωση αυτή, χρησιμοποιούμενες στο πρόγραμμα, θεωρούνται σαν ακέραιες. Π.χ.:

β) Εντολή REAL.

Είναι αντίστροφη της προηγούμενης. Η μορφή της είναι:

|| REAL μεταβλητή, μεταβλητή, ...

Όλες οι μεταβλητές που δηλώνονται είναι ακέραιες, θα τις χρησιμοποιήσουμε δε στο πρόγραμμά μας, σαν να ήταν πραγματικές. Π.χ.:

REAL J, KM, LAB

γ) Εντολή DIMENSION.

Την αναφέρουμε μόνο όταν στο πρόγραμμα χρησιμοποιούνται πίνακες. Η μορφή της είναι:

|| DIMENSION μεταβλητή (αριθμητικό δείκτες), μεταβλητή (αριθμητικό δείκτες), ...

Οι αριθμητικοί δείκτες είναι οριακοί αριθμοί, που εκφράζουν το μέγεθος (διαστάσεις) των πινάκων. Π.χ.:

DIMENSION A(20), B(5,6), C(4,4)

Με την εντολή αυτή δηλώνεται ο χώρος, που επιθυμούμε να κρατηθεί στη μνήμη, για να καταχωρισθούν οι τιμές των στοιχείων για τον καθένα από τους πίνακες. Στο παράδειγμά μας, θα κρατηθούν 20 θέσεις για τα στοιχεία του πίνακα A, 30 (5 x 6) θέσεις για τα στοιχεία του B και 16 (4 x 4) θέσεις για τα στοιχεία του C.

10.19 Μερικές ακόμη εντολές.

α) Εντολή STOP.

Συνήθως είναι η προτελευταία εντολή, με την οποία και σταματά η εκτέλεση ενός FORTRAN προγράμματος.

β) Εντολή END.

Είναι πάντα η τελευταία εντολή ενός FORTRAN προγράμματος. Με αυτήν διακόπτεται εντελώς η εκτέλεση και επιστρέφει ο έλεγχος από το μεταφραστικό πρόγραμμα στο εποπτεύον σύστημα¹.

10.20 Ειδικές μορφές των εντολών READ και WRITE.

Αναφέρονται στις περιπτώσεις αναγνώσεως ή έγγραφής 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.:

```
    || READ (5,4) (A(I), I = 1,19)
4  || 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 η κάθε μία. Η αντιστοίχιση των αριθμών που διαβάζονται και των στοιχείων είναι τέτοια, ώστε ο πρώτος αριθμός να καταχωρισθεί σαν περιεχόμενο του A(1), ο δεύτερος σαν περιεχόμενο του A(2) κ.ο.κ.

Το ίδιο ακριβώς συμβαίνει και με την εντολή WRITE. Π.χ.:

```
    || WRITE (6,2) (B(I), I = 1,12)
2  || FORMAT (8(F6.1, 2X))
```

Στο παράδειγμα αυτό γράφονται στον εκτυπωτή τα περιεχόμενα 12 στοιχείων του πίνακα A, με συνεχή τρόπο σε δυο 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 θέλομε να γράψουμε σχόλια σε ένα πρόγραμμα, συμπληρώνομε στη στήλη μιας γραμμής του εντύπου κωδικογραφήσεως το χαρακτήρα C, για να μη θεωρηθούν από τον υπολογιστή σαν εντολές. Από την τρίτη στήλη και έπειτα, μπορούμε να γράψουμε με λατινικούς χαρακτήρες ο, πιθήποτε νομίζουμε ότι μπορεί να βοηθήσει στην κατανόηση του προγράμματος από κάποιον τρίτο, που θα θελήσει

¹ Για τις έννοιες αυτές θα μιλήσουμε στην παράγραφο 15.3.

να το διαβάσει, ή και από τον ίδιο τον προγραμματιστή, όταν θελήσει να ξαναδεί το πρόγραμμά του μετά από αρκετό καιρό.

Γραμμές με σχόλια μπορούμε να γράψουμε δισες θέλομε και σε οποιοδήποτε σημείο του προγράμματος. Π.χ.:

1	5
C	TO PROGRAMMA YPOLOGIZEI
C	TIS RIZES MIAS EXISOSEOS
C	DEFTEROU BATHMOU

10.22 Υποπρογράμματα (Subprograms).

Όπως είπαμε και στην περιγραφή της BASIC, σαν υποπρόγραμμα χαρακτηρίζομε ένα μικρό ανεξάρτητο βοηθητικό πρόγραμμα, που μπορεί να χρησιμοποιείται σε κάποιο σημείο του κυρίως προγράμματος (Main program), για να εκτελεί ορισμένους συγκεκριμένους υπολογισμούς.

Διακρίνομε δύο κατηγορίες υποπρογραμμάτων:

- a) Υπορρουτίνες (Subroutines).
- β) Συναρτήσεις (Functions).

Θα δούμε στα επόμενα τον τρόπο καταρτίσεως τέτοιων υποπρογραμμάτων.

a) Υπορρουτίνες (Subroutines).

Mία υπορρουτίνα έχει την εξής μορφή:

SUBROUTINE	κωδική ονομασία υπορρουτίνας (παράμετροι)
.....	} Εντολές της υπορρουτίνας
.....	
.....	
.....	
RETURN	
END	

Οι παράμετροι είναι μεταβλητές που αναφέρονται στις εντολές του υποπρόγραμματος και ονομάζονται τυπικές παράμετροι (Formal parameters).

Μέσω των παραμέτρων αυτών εισάγονται οι απαραίτητες τιμές για να λειτουργήσει το υποπρόγραμμα και, επίσης, αποδίδονται στο κυρίως πρόγραμμα τα αποτέλεσματα των υπολογισμών.

Η κωδική ονομασία της υπορρουτίνας υπόκειται στους περιορισμούς γραφής, που ισχύουν και στις απλές μεταβλητές.

Όλες οι υπορρουτίνες τελειώνουν με τις εντολές RETURN και END. Μέσα στην υπορρουτίνα μπορούμε να χρησιμοποιήσουμε όλες τις εντολές της FORTRAN.

Παράδειγμα 1.

Να γραφεί υπορρουτίνα που να υπολογίζει το εμβαδόν ενός τριγώνου, όταν είναι γνωστά τα μήκη των πλευρών του.

Η υπορρουτίνα θα έχει τη μορφή του σχήματος 10.22α.

Για να λειπουργήσει η υπορρουτίνα, πρέπει να της δοθούν οι τιμές των πλευρών A, B, C. Αυτό γίνεται με την εισαγωγή των τιμών από το κυρίως πρόγραμμα μέσω των τυπικών παραμέτρων, που συμβολίζουν τα μήκη των αντιστοίχων πλευρών. Το αποτέλεσμα αποδίδεται στο κυρίως πρόγραμμα μέσω τής παραμέτρου E.

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ		
ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝ	
1	5 6 7	ΕΝΤΟΛΗ FORTRAN
		S,U,B,R,O,U,T,I,I,N,E,I,A,I,R,E,A,(A,,B,,C,,E,)I
C		I
C	T,H,I	S,,S,U,B,R,O,U,T,I,I,N,E,,C,A,L,C,U,L,A,I,T,E,S
C	T,H,E	A,R,E,A,,O,F,,A,,T,R,I,A,N,G,L,E,
C		I
		T,I=(A,I+B,I+C,I),/12,I
		R,I=T,I*(I-T,-A,I)*I,(T,I-B,I)*I,(T,I-C,I)
		E,I=S,Q,R,I,T,I,(R,I)
		R,I,E,I,T,I,U,R,I,N,I
		E,I,N,D
		I

Σχ. 10.22α.

Παράδειγμα 2.

Να γραφεί υπορρουτίνα που να υπολογίζει το $N! = 1 \times 2 \times 3 \dots (N-1) \times N$.

Η μορφή της θα είναι (σχ. 10.22β):

Ο αριθμός του σπούδου ζητείται ο υπολογισμός του παραγοντικού, εισάγεται από το κυρίως πρόγραμμα στην υπορρουτίνα μέσω της τυπικής παραμέτρου N, ενώ το αποτέλεσμα που καταχωρίζεται στην K, αποδίδεται μέσω αυτής στο κυρίως πρόγραμμα.

β) Εντολή κλήσεως μιας υπορρουτίνας.

Μία υπορρουτίνα καλείται στο κυρίως πρόγραμμα με την εντολή CALL που έχει τη μορφή:

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ

ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΧΟΛΙΑ	ΕΝΤΟΛΗ FORTRAN
1	5	SUBROUTINE NEL(NPARG,(N,K))
C		
C	T,H	SUBROUTINES CALCULATES
C	T,HE	FUNCTION FACTORIAL
C		
	K=1	
	D,O1,1,0,I=12,N	
	I,0	K=K*I
	R,IETURN	
	E,N,D	

Σχ. 10.22B.

CALL κωδική ονομασία υπορρουτίνας (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 το πλήθος και ως προς τον τύπο (πραγματικές ή ακέραιες). Πχ. για να καλέσουμε την υπορρουτίνα του 1ου παραδείγματος σε κάποιο σημείο ενός προγράμματος, θα χρησιμοποιήσουμε την εντολή:

CALL AREA (A1, B1, C1, E1)

όπου οι παράμετροι A1, B1, C1 είναι μεταβλητές του κυρίως προγράμματος, που θα μεταβιβάσουν τις τιμές τους στις παραμέτρους A, B, C της υπορρουτίνας, για να μπορέσει η τελευταία να λειτουργήσει. Το αποτέλεσμα από τον υπολογισμό θα επιστρέψει, μέσω της παραμέτρου E της υπορρουτίνας, στο κυρίως πρόγραμμα, σαν τιμή της E1.

γ) Συναρτήσεις (Functions).

Διακρίνομε δύο κατηγορίες συναρτήσεων:

α) Συναρτήσεις σε μορφή εντολής (Function statements).

β) Συναρτήσεις υποπρογράμματα (Function subprograms).

Οι πρώτες είναι σχεδόν ίδιες με τις αντίστοιχες συναρτήσεις που περιγράψαμε στη BASIC¹ [παράγραφος 9.15 (α)].

Θα ασχοληθούμε με τις συναρτήσεις της δεύτερης κατηγορίας που μοιάζουν αρκετά στη δομή με τις υπορρουτίνες.

Η μορφή τους είναι:

```
FUNCTION κωδική ονομασία της συναρτήσεως (παράμετροι)
..... } Έντολές της συναρτήσεως
..... }
RETURN
END
```

Όλες οι μεταβλητές στην παρένθεση, που είναι οι τυπικές παράμετροι, χρησιμοποιούνται για την εισαγωγή των απαραίτητων τιμών, για να λειτουργήσει η συνάρτηση. Μια συνάρτηση μόνο ένα αποτέλεσμα μπορεί να επιστρέψει στο κυρίως πρόγραμμα (αντίθετα προς μία υπορρουτίνα). Η κωδική ονομασία της συναρτήσεως παίζει συγχρόνως και το ρόλο της μεταβλητής, όπου καταχωρίζεται η τιμή του αποτελέσματος και μέσω της οποίας επιστρέφεται στο κυρίως πρόγραμμα.

Μία συνάρτηση μπορεί να χρησιμοποιείται στο κυρίως πρόγραμμα χωρίς ιδιαίτερη εντολή κλήσεως, αλλά απλώς αναφερόμενη με την ονομασία της, σαν απλός όρος μέσα σε μια έκφραση FORTRAN, όπως ακριβώς γίνεται και με τις συναρτήσεις του συστήματος, που είδαμε στην παράγραφο 10.4.

Παράδειγμα 1.

Να γραφεί συνάρτηση (υποπρόγραμμα) για τον υπολογισμό του εμβαδού ενός τριγώνου, όταν είναι γνωστά τα μήκη των πλευρών του.

Όπως βλέπομε (σχ. 10.22γ), το ίδιο το όνομα της συναρτήσεως (AREA) χρησιμοποιείται σαν απλή μεταβλητή για την καταχώρηση του τελικού αποτελέσματος του υπολογισμού.

Παράδειγμα 2.

Να γραφεί συνάρτηση για τον υπολογισμό της τιμής του NI (N παραγοντικό) (σχ. 10.22δ).

Το αποτέλεσμα του υπολογισμού καταχωρίζεται στην ίδια την ονομασία της συναρτήσεως (NPARAG), που χρησιμοποιείται σαν απλή μεταβλητή μέσα στη συνάρτηση και μέσω της οποίας επιστρέφεται η υπολογισθείσα τιμή στο κυρίως πρόγραμμα.

¹ Αντί των λέξεων DEF και FN αναφέρεται απλώς η ονομασία της συναρτήσεως. Π.χ.:

$$F(X,Y) \doteq X^2Y - X^{**2} + Y^{**2}$$

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ

ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝ	ΕΝΤΟΛΗ FORTRAN
1	5 6 7	
		FUNCTIION AREAA(A1,B1,C1)
C1	T,H	S,FUNCTIION CALCULATE,S
C1	T,H,E	AREAOF ATRIANGLE
C1		
		T1=(A1+B1+C1)/12.0
		R1=T1*(T1-A1)*(T1-B1)*(T1-C1)
		AREAA=SQRT(R1)
		RETURN
		END

Σχ. 10.22γ.

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ

ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝ	ΕΝΤΟΛΗ FORTRAN
1	5 6 7	
		FUNCTIION NPARIAG(N)
C1	T,H	S,FUNCTIION CALCULATE,S
C1	T,H,E	N FACTORIAL
C1		
		NPARIAG=1
		DO 10 I=2,N
10		NPARIAG=NPARIAG*I
		RETURN
		END

Σχ. 10.22δ.

10.23 Έντυπο αναγραφής των δεδομένων (Data form).

Στο έντυπο αυτό (σχ. 10.23) αναγράφομε τα δεδομένα (data) σύμφωνα με τις εντολές FORMAT που έχουν δοθεί στο πρόγραμμα, πριν διατρηθούν σε δελτία.

Το έντυπο περιλαμβάνει ορισμένο αριθμό γραμμών που κάθε μία χωρίζεται σε 80 στήλες.

Το περιεχόμενο μιας γραμμής αντιπροσωπεύει το περιεχόμενο ενός δελτίου δεδομένων.

Οι γραμμές του εντύπου δεδομένων συμπληρώνονται με την ίδια σειρά, με την οποία και θα τοποθετηθούν τα δελτία στην ομάδα δελτίων εντολής του προγράμματος.

10.24 Εφαρμογές.

Θα δούμε παρακάτω μερικές πλήρεις εφαρμογές.

Παράδειγμα 1.

Να υπολογισθεί η τιμή της συναρτήσεως $\psi = x^2 + 4x - 5$ για τιμές του $x = 0.2$ (0.2) 4.

Το πρόγραμμα θα έχει την εξής μορφή (σχ. 10.24α).

Παράδειγμα 2.

Να βρεθεί το μικρότερο από τα στοιχεία ενός μονοδιάστατου πίνακα A. Ο πίνακας αποτελείται από 20 στοιχεία που έχουν διατρηθεί σε δελτία με κωδικό FORMAT (16F5.1) (σχ. 10.24β).

Το πρόγραμμα θα έχει ως εξής (σχ. 10.24γ).

Παράδειγμα 3.

Να υπολογισθεί η τιμή της παραστάσεως $\sum_{v=1}^{10} a_v x^v$ Θεωρείστε ότι οι

συντελεστές a_v αποτελούν στοιχεία ενός μονοδιάστατου πίνακα και ότι οι τιμές τους έχουν διατρηθεί σε ένα δελτίο με FORMAT (10F6.1). Η τιμή του x διαβάζεται από ένα δεύτερο δελτίο (σχ. 10.24δ).

Το πρόγραμμα έχει ως εξής (σχ. 10.24ε).

Παράδειγμα 4.

Να υπολογισθούν οι τιμές της συναρτήσεως

$$\psi = \begin{cases} x^3 - 2x & \text{dν } x > 1 \\ x^2 - 1 & \text{dν } x < 1 \\ x & \text{dν } x = 1 \end{cases}$$

για 50 τιμές του x διάτρημένες η κάθε μια σε ξεχωριστό δελτίο, με FORMAT (F4.2) (σχ. 10.24στ).

Το πρόγραμμα θα έχει την εξής μορφή (σχ. 10.24ζ).

DATA FORM

ΗΜΕΡΟΜΗΝΙΑ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΗΣ

ΣΕΛΙΔΑ _____ ΑΠΟ _____

1	6	11	16	21	26	31	36	41	46	51	56	61	66	71	76	81	86	91	96
---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Σχ. 10.23.
Έντυπο αναγραφής δεδομένων (Data Form).

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ		APIΜΟΣ ΕΝΤΟΛΗΣ	5 6 7	10	ΕΝΤΟΛΗ FORTRAN	20	FORTRAN	30																		
C	L	I	V	A	L	U	E	S	I	O	F	I	A	I	F	U	N	C	T	I	L	O	N			
X	=	0	.	0	.	2	
Y	=	X	.	*	1	2	+	4	*	1	*	X	-	5	*	1		
W	R	R	I	T	E	(6	,	1	1)	Y	
F	O	R	M	A	T	(F	,	8	,	1	4	,	1	,
I	F	(X	,	E	Q	,	4	,	1	0	1),	.	S	,	Q	P	,	
X	=	X	.	+	0	.	1	2	
G	O	,	I	O	L	,	2	
E	N	D	

Σε. 10.24α.

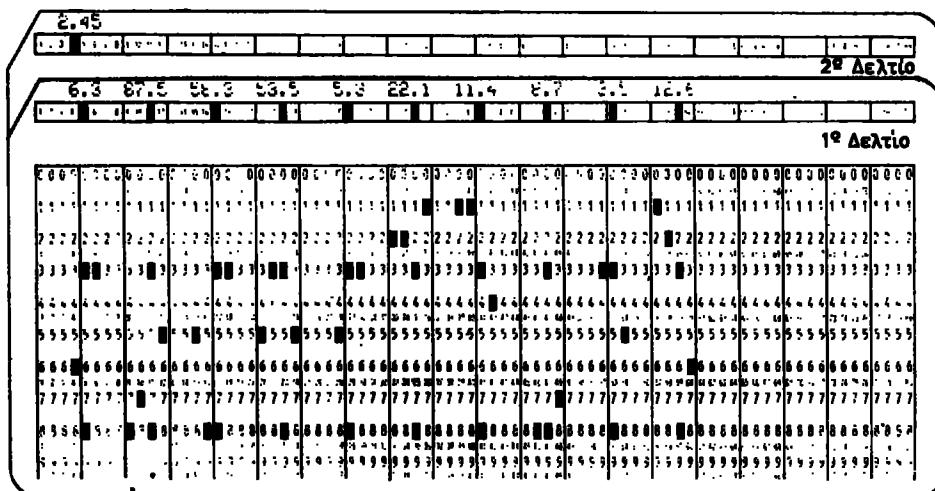
54.6 22.3 5.8 7.4	28 Δεκτίο
.8 .6 5.6 45.3 33.5 78.6 8.6 7.6 54.5 36.2 5.3 75.5 21.0 5.2 33.5 77.0	19 Δεκτίο

Σχ. 10.24β.

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ

ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝ	ΕΝΤΟΛΗ FORTRAN
1	5 6 7	ΕΝΤΟΛΗ FORTRAN
C		I M I N I M U M I E L E M E N T I O F I A N I A R R A Y I I I I
C		D I M E N S I O N I A (2 , 0)
		R E A D (5 , 1 1) , (A (I) , , I = 1 , , 2 0 ,)
	1	F O R M A T (1 . 6 , F 5 . 0 ,)
		S M I N = A (1 ,)
		D O 1 , 0 , , I = 2 , , 2 0 ,
		I F (A (I) > L E I 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 . 0 , 1 ,)
		S T O P
		E N D

Σχ. 10.24γ.



Σχ. 10.246.

ΣΤΗΛΗ ΠΑ ΣΧΟΛΙΑ

ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝΤΟΛΗ	ΕΝΤΟΛΗ FORTRAN
1	5 6 7	ΕΝΤΟΛΗ FORTRAN
		C I C A L C U L A T I O N _ O F _ A _ S U M _
		C
		D I M E N S I O N _ A (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 , 0 , 1 ,)
		R E A D (, 5 , , 2 ,) , X
2		F O R M A T (, F , 6 , 0 , 2 ,)
		S I = 0 , 0
		D O , 1 , 0 , , I = 1 , , 1 , 0 ,
1 0		S I = S I + A I (, I ,) * X * , * I
		W R I T E (, 6 , , 3 ,) , S I
3		F O R M A T (, F , 9 , 0 , 3 ,)
		S T O P
		E N D

Σχ. 10.24ε.

ΣΤΗΛΗ ΓΙΑ ΣΧΟΛΙΑ:

ΑΡΙΘΜΟΣ ΕΝΤΟΛΗΣ	ΣΥΝΟΛΟ ΕΝΤΟΛΩΝ	ΕΝΤΟΛΗ FORTRAN
1	5 6 7	ΕΝΤΟΛΗ FORTRAN
		C I N V A L U E S I O F I A F U N C T I O N
		DO 1,5,1 I=1,1,5,0
		READ(5,11) X
		1 FORMATT(F4.0,2)
		1 F1(X,-1.0),2,0,1,0,3,0
		2,0 Y=X,*,*2,-1,0
		GO TO 14,0
		11,0 Y=X
		GO TO 14,0
		13,0 Y=X,*,*3,-2,0,*1X
		14,0 WRIT(E,(6,,2)),Y
		12 FORMAT(F6.0,2)
		1,5 CONTINUE
		STOP
		END

Σχ. 10.24στ.

2.33	50F Δελτίο
6.51	10 Δελτίο

Σχ. 10.24ξ.

10.25 Διαδικασία προετοιμασίας ενός προγράμματος FORTRAN.

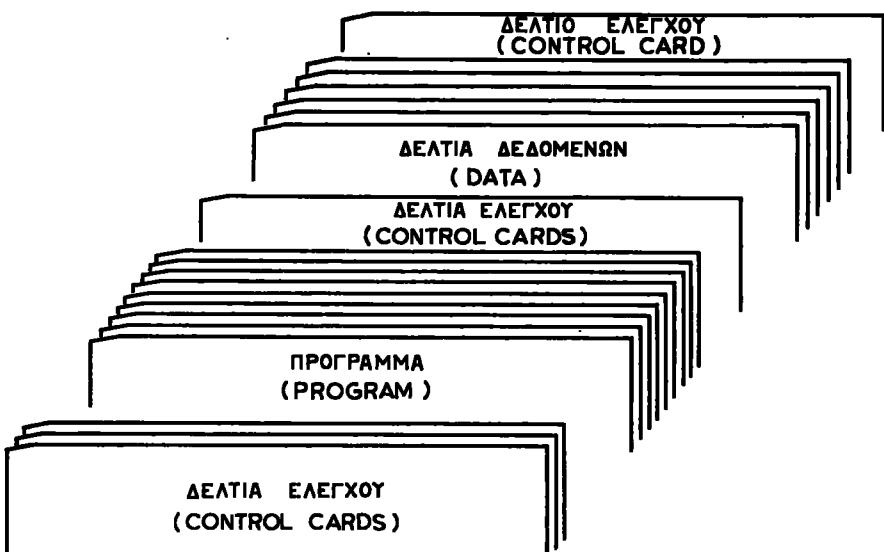
Αφού σχεδιασθεί το διάγραμμα ροής, στη συνέχεια γίνεται η κωδικοποίηση του προγράμματος γράφοντας τις εντολές στο ειδικό έντυπο κωδικογραφήσεως της FORTRAN (σχ. 10.9). Οι εντολές διατρυπώνται κατόπιν σε δελτία, έτσι ώστε σε κάθε δελτίο να αντιστοιχεί το περιεχόμενο μιας γραμμής του εντύπου. Στη συνέχεια διατρυπώνται και τα δεδομένα του προγράμματος, τα οποία έχουν γραφεί σύμφωνα με τους κωδικούς των εντολών FORMAT στα ειδικά έντυπα δεδομένων (DATA FORMS, σχήμα 10.23). Προστίθενται ορισμένα δελτία ελέγχου¹ (Control Cards) και σχηματίζεται έτσι η ομάδα των δελτίων των εντολών του προγράμματος και των δεδομένων, έτοιμο για να εισαχθεί και να εκτελεσθεί στον υπολογιστή (σχ. 10.25).

10.26 Ασκήσεις.

- Χαρακτηρίστε με τους όρους ακέραιη σταθερή, ακέραιη μεταβλητή, πραγματική σταθερή, πραγματική μεταβλητή τα εξής:

A12	KL	FRAM	784	-274
-17.4	1.8E + 4	GXB4	0.06	AREA
JB(1)	BARG	C (I)	4.E-3	177.05

1 Η μορφή των δελτίων ελέγχου διαφέρει από υπολογιστή σε υπολογιστή. Διαφέρει ακόμη και το πλήθος τους. Τα συμπληρώνομε κατάλληλα με τη βοήθεια του υπεύθυνου του συγκροτήματος. Σε κάθε δελτίο ελέγχου αναγράφονται πληροφορίες προς το σύστημα του υπολογιστή.

**Σχ. 10.25.**

Η ομάδα δελτίων των εντολών και των δεδομένων του προγράμματος.

2. Ποιες από τις παρακάτω ονομασίες μεταβλητών είναι σωστές, ποιες λάθος και γιατί;

AK1	F+5	EQUAT	MK(A)	EDR(0,8)
1ETO	KALY	R B12	L2 (4,2)	NAMIN (K, P)
B(-2)	TRIANGLE	FF 84	SUM (I+J)	ELPOS(20)

3. Να κωδικοποιηθούν με τη βοήθεια της μερικής αναμίξεως οι παρακάτω μαθηματικές σχέσεις:

$$x^2 + \frac{4}{3} \alpha x + \beta \quad 2\eta x + 5 \text{ συνx} \quad \frac{1}{5} \sigma \text{υνx} + \eta \mu^2 \psi$$

$$\frac{\sqrt{x}}{4} + \frac{1}{3} e^x \quad \alpha \beta + 2 (\gamma \delta + \epsilon \alpha) \frac{\beta}{\gamma} \quad |x| + \frac{2e^{-x}}{\eta \mu x}$$

4. Επίσης οι σχέσεις:

$$\frac{3x^2 + 6x + 7}{2x^3 - 3x^2 + 5} \quad (\eta \mu x + \sigma \text{υνx})^2 \quad \frac{1}{3} \sqrt{\alpha^2 - 2\beta \gamma + \delta}$$

$$\frac{|\alpha|}{4 + |\beta|} + \frac{6}{7} \quad [(\alpha + \beta)^2 - \beta^2 \gamma^2] \alpha \beta \gamma \quad \frac{e^{2x} + e^{3x}}{5}$$

5. Να κωδικοποιηθούν οι επόμενες απλές και σύνθετες μαθηματικές λογικές σχέσεις:

$$\alpha < \beta \quad x^2 + 1 > 4 \text{ κατ } \psi > 0 \quad x \psi + z = 0 \text{ κατ } \alpha \beta \gamma = 1$$

$$0 < \gamma < \delta \quad a = 0 \text{ if } \beta > 0 \quad a = 0 \text{ and } \frac{\delta}{\epsilon} \neq \frac{\gamma}{\beta}$$

6. Επίσης οι σχέσεις:

$$\begin{array}{lll} a_1 < a_2 < a_3 & \gamma = 0 \text{ and } x < 3 & a < \frac{1}{3} \text{ and } \beta = \frac{1}{5} \\ 0 < ab < 4 & z = 2 \text{ if } \psi = 4 \text{ if } x = 6 & x\beta - 1 > 0 \text{ if } x = \psi = 2. \end{array}$$

7. Πώς θα διατρυπήσετε 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. Ομοίως 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	Θα τυπωθεί
-17	I4	
224	I2	
3.54	F4.2	
12.448	F6.2	
7.13	F3.1	
1.548	F6.2	

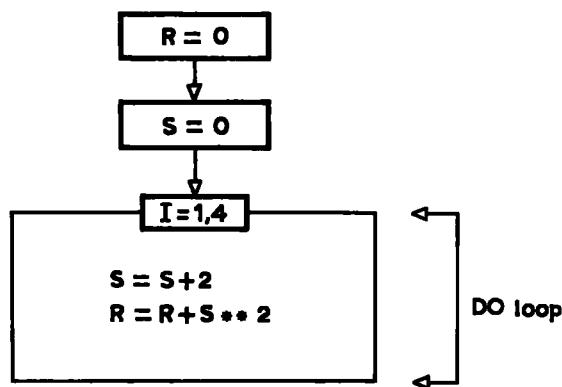
10. Επίσης τον πίνακα:

Αριθμός μέσα στη μνήμη	Κωδικός FORMAT	Θα 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	

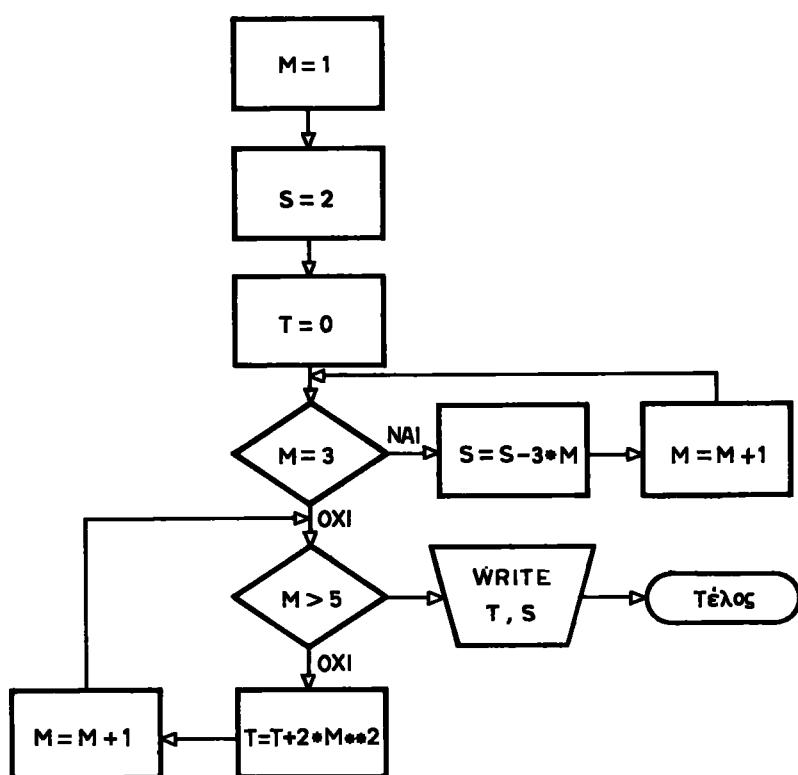
11. Να γραφεί πρόγραμμα που να υπολογίζει και να τυπώνει το άθροισμα των αντιστρόφων των 20 πρώτων φυσικών αριθμών ($1 + 1/2 + \dots + 1/20$).
12. Να γραφεί πρόγραμμα που να υπολογίζει και να τυπώνει τις τιμές της συναρτήσεως $y = x \text{ if } x + \sin \psi$ για τιμές του $x = 0.1$ (0.1) 0.7.
13. Δίνεται μονοδιάστατος πίνακας A(I) με 30 στοιχεία. Να γραφεί πρόγραμμα που να διαβάζει τα στοιχεία με κωδικούς FORMAT (16F5.1) και στη συνέχεια να υπολογίζει το μέσο όρο τους.
14. Δίνονται δύο μονοδιάστατοι πίνακες A (I) και B (I) με 15 στοιχεία ο καθένας. Να γραφεί πρόγραμμα που να τους διαβάζει και να δημιουργεί νέο πίνακα C (I) κάθε στοιχείο του οποίου θα ισούται με το άθροισμα των αντιστοίχων στοιχείων των δοθέντων πινάκων.
15. Να γραφεί υπορρομπτίνα, που να υπολογίζει τον όγκο μιας σφαίρας, όταν είναι γνωστή η ακτίνα της.
16. Να γραφεί συνάρτηση (υποπρόγραμμα), που να υπολογίζει την τιμή της συναρτήσεως

$$\psi = \begin{cases} x^2 & \text{if } x = 0 \\ x^2 + 1 & \text{if } x = 0 \end{cases}$$

17. Ποιες θα είναι οι τιμές των μεταβλητών S και R μετά την εκτέλεση του παριπθεμένου τμήματος του προγράμματος:



18. Να γίνει η κωδικοποίηση του παρακάτω διαγράμματος ροής:



ΚΕΦΑΛΑΙΟ ΕΝΔΕΚΑΤΟ

ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΜΗΧΑΝΕΣ ΓΡΑΦΕΙΟΥ - ΑΡΙΘΜΟΜΗΧΑΝΕΣ

11.1 Γενικά.

Σαν υπολογιστικές μηχανές γραφείου εννοούμε μικρές σε μέγεθος μηχανές με δυνατότητες κατά πολύ μικρότερες από εκείνες ενός ηλεκτρονικού υπολογιστή.

Σήμερα, οι εξελιγμένες μορφές των μηχανών αυτών χρησιμοποιούνται πάρα πολύ σε μικρού μεγέθους εφαρμογές, όπου η χρήση του μεγάλου ηλεκτρονικού υπολογιστή είναι ασύμφορη. Έτσι, οι ανάγκες ενός γραφείου ή μιας μικρής επιχειρήσεως καλύπτονται από τις μηχανές αυτές.

Η βιομηχανική και τεχνική ανάπτυξη μετά το 2ο Παγκόσμιο πόλεμο δεν οδήγησε μόνο στην τελείωτη των μεγάλων ηλεκτρονικών υπολογιστών, αλλά ακόμη επέρεασε θετικά και την εξέλιξη των παλαιού τύπου υπολογιστικών μέσων. Ο τρόπος λειτουργίας τους μεταβλήθηκε με την πάροδο του χρόνου, ακολουθώντας την εξέλιξη της τεχνικής.

Όπως και στους υπολογιστές, διακρίνομε 4 γενιές υπολογιστικών μηχανών.

Η πρώτη γενιά, που καλύπτει τη χρονική περίοδο 1870 - 1920, περιλαμβάνει υπολογιστικές συσκευές, ο τρόπος λειτουργίας των οποίων ήταν καθαρά μηχανικός. Τα διάφορα τμήματα έμπαιναν σε λειτουργία με τη χρήση μοχλών από το χειριστή και η κίνηση από το ένα τμήμα στο άλλο μεταδίδονταν με οδοντωτούς τροχούς. Οι μηχανές αυτές ήταν βασικά αθροιστικές και γι' αυτό χρησιμοποιήθηκαν στην αρχή κυρίως στα ταμιακές ή στην επεξεργασία στατιστικών στοιχείων. Αργότερα κατασκευάσθηκαν τύποι, που μπορούσαν να εκτελούν όχι μόνο προσθέσεις, αλλά και αφαιρέσεις και πολλαπλασιασμούς.

Η δεύτερη γενιά, που αρχίζει από το 1920 και τελειώνει το 1955, περιλαμβάνει μηχανές με ηλεκτρομηχανικό τρόπο λειτουργίας. Εδώ τα διάφορα τμήματα κινούνται με τη βοήθεια του ηλεκτρισμού. Ο χειριστής πίεζει πλήκτρα και αποκαθιστά το ρεύμα σε κατάλληλα κυκλώματα, που θέτουν σε κίνηση τα μηχανικά μέρη της συσκευής. Με την ίδια βασική αρχή λειτουργίας κατασκευάστηκαν διάφοροι τύποι, για να εξυπηρετήσουν συγκεκριμένες ανάγκες, όπως ταμιακές μηχανές, μηχανές 4 αριθμητικών πράξεων, λογιστικές κ.ο.κ. Ακόμη και σήμερα εξακολουθούν να χρησιμοποιούνται σε ορισμένες εταιρίες ή τράπεζες εξελιγμένες τέτοιες μηχανές. Οι μηχανές της δεύτερης γενιάς αναφέρονται συχνά με το γενικό όνομα κλασικές υπολογιστικές μηχανές. Τη λειτουργία κάθε τύπου και τις εργασίες που εκτελεί θα δούμε πιο κάτω.

Η τρίτη γενιά αρχίζει από το 1955 και φθάνει μέχρι το 1970. Η 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 πιο αποδοτική. Οι φθορές έχουν μειωθεί και ο χρόνος επεξεργάσιας έχει ελαττωθεί σημαντικά. Οι ηλεκτρονικές μηχανές παρουσιάζονται και αυτές σε διάφορους τύπους, όπως απλές εμπορικές, επιστημονικές κλπ.

Η εξέλιξη όμως συνεχίζεται και τα επιτεύγματα της ηλεκτρονικής στην κατεύθυνση μικροκυκλωμάτων οδήγησαν στο σχεδιασμό υπολογιστικών μηχανών με εξαιρετικά μικρές διαστάσεις και μεγάλες δυνατότητες. Ήδη διανύομε την τέταρτη γενιά υπολογιστικών μηχανών.

Όλες οι μηχανές που θα περιγραφούν στο κεφάλαιο αυτό φέρουν το γενικό όνομα Calculators. Η απόδοσή του στα Ελληνικά γίνεται με χρήση του διπλού όρου υπολογιστικές μηχανές ή αριθμομηχανές.

11.2 Κλασσικές υπολογιστικές μηχανές.

Στην κατηγορία αυτή ανήκουν μηχανές διαφόρων τύπων για εμπορική κυρίως χρήση. Οι μηχανές αυτές χρησιμοποιήθηκαν επί ολόκληρες δεκαετίες για την εκτέλεση διαφόρων λογιστικών εργασιών, τη δημιουργία στατιστικών πινάκων κ.ο.κ.

Παρά την πρόσφατη μεγάλη ανάπτυξη των ηλεκτρονικών υπολογιστών, οι μηχανές αυτές χρησιμοποιούνται και σήμερα (σε περιορισμένη όμως κλίμακα) σε πολλές δουλειές ρουτίνας.

Ταξινομούνται στις εξής κατηγορίες:

- Απλές αθροιστικές μηχανές και μηχανές 4 πράξεων με εκτύπωση (Adding machines and printing calculators).
- Αυτόματες υπολογιστικές μηχανές με πληκτρολόγιο (Key - driven calculators).
- Υπολογιστικές μηχανές, με κινητό πίνακα αποτελεσμάτων (Rotary calculators).
- Λογιστικές μηχανές (Accounting machines).

α) Απλές αθροιστικές και μηχανές 4 πράξεων με εκτύπωση.

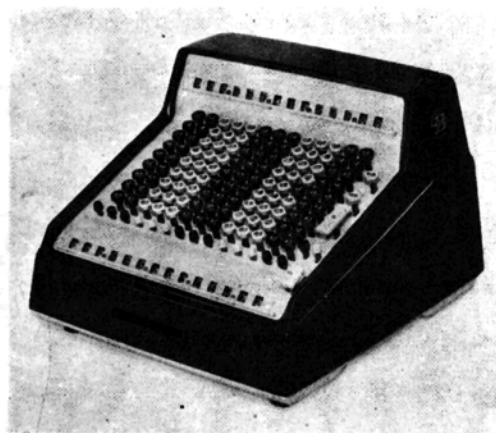
Οι μηχανές αυτές θα περιγραφούν μαζί, λόγω της ομοιότητας που παρουσιάζουν στη κατασκευή και στη λειτουργία τους. Και οι δύο διαθέτουν πληκτρολόγιο με 10 κύρια πλήκτρα αριθμητικών ψηφίων (Ten key models) και μερικά άλλα βοηθητικά (σχ. 11.2α).

Οι μηχανές αυτές έχουν σχεδιασθεί να λειτουργούν με απλή πίεση ενός πλήκτρου (One - hand touch operation). Για να εισαχθεί ένας αριθμός στη μηχανή, πιέζονται τα πλήκτρα των ψηφίων του με τη σειρά, ξεκινώντας από το πιο σημαντικό.

Οι αθροιστικές μηχανές κατασκευάζονταν παλιότερα και σε τύπους πλήρους πληκτρολογίου (Full - key board), με πλήκτρα τοποθετημένα σε κατακόρυφες στήλες τόσες, δύο και το μέγιστο πλήθος των ψηφίων, που μπορεί να δεχθεί. Σε κάθε στήλη υπάρχουν πλήκτρα αριθμημένα από το 1 έως το 9. Για να εισάγομε ένα αριθμό πιέζομε ένα πλήκτρο από κάθε στήλη που αντιστοιχεί στο ψηφίο του αριθμού. Η χωρητικότητα της κάθε μηχανής εξαρτάται από το πλήθος των στηλών του



Σχ. 11.2α.
Μία μηχανή 4 πράξεων.



Σχ. 11.2β.
Αυτόματη υπολογιστική μηχανή με πλήρες πληκτρολόγιο.

πληκτρολογίου που διαθέτει. Και οι δύο τύποι μηχανών τυπώνουν τα αποτελέσματα σε χαρτοταπινία. Επίσης μπορούν να εκτελούν και τις 4 αριθμητικές πράξεις. Η κυριότερη διαφορά μεταξύ τους αφορά στον τρόπο εκτελέσεως του πολλαπλασιασμού. Η αθροιστική μηχανή πολλαπλασιάζει εκτελώντας διαδοχικές προσθέσεις, ενώ η μηχανή 4 πράξεων πολλαπλασιάζει απ' ευθείας. Ορισμένοι τύποι μηχανών 4 πράξεων έχουν πληκτρο σταθερού συντελεστή, εκτυπώνουν κλασματικά ψηφία, κάνουν αυτόματη ύψωση στο τετράγωνο και εκτελούν πολλαπλασιάσμούς με αρνητικούς αριθμούς.

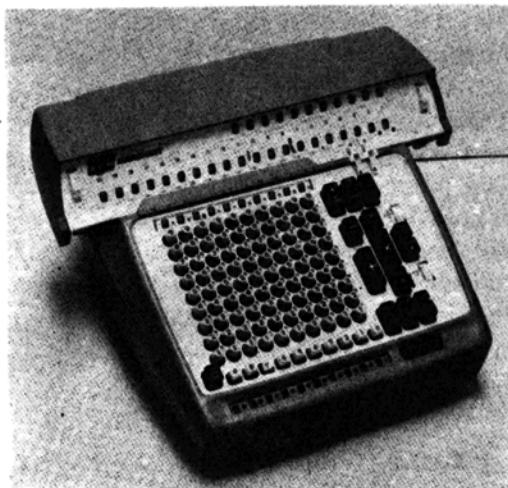
β) Αυτόματες υπολογιστικές μηχανές.

Είναι μηχανές πλήρους πληκτρολογίου (σχ. 11.2β) με στήλες από πλήκτρα αριθμημένα από το 1 έως το 9, όμοιες με τις αθροιστικές μηχανές πλήρους πληκτρολογίου. Οι υπολογισμοί γίνονται αυτόματα με την πίεση των καταλλήλων πλήκτρων. Τα αποτελέσματα εμφανίζονται συνήθως σε ένα ειδικό άνοιγμα - παράθυρο, στο πάνω ή κάτω μέρος της μηχανής. Μερικοί τύποι διαθέτουν δύο τέτοια παράθυρα. Στο ένα εμφανίζονται τα ενδιάμεσα αποτελέσματα και στο άλλο τα συγκεντρωτικά. Παρ' όλο που οι μηχανές αυτές μπορούν να εκτελούν και τις 4 αριθμητικές πράξεις, είναι περισσότερο κατάλληλες για γρήγορες προσθέσεις και πολλαπλασιασμούς (οι πολλαπλασιασμοί εκτελούνται και εδώ με διαδοχικές προσθέσεις). Για το λόγο αυτό χρησιμοποιούνται κυρίως για λογιστικούς ελέγχους, και ταμιακές εργασίες.

γ) Υπολογιστικές μηχανές με κινητό πίνακα αποτελεσμάτων.

Παρ' όλο που οι μηχανές αυτές μπορούν να εκτελούν και τις 4 αριθμητικές πράξεις, χρησιμοποιούνται κυρίως για την επίλυση προβλημάτων με πολύπλοκους πολλαπλασιασμούς και διαιρέσεις. Οι περισσότερες μηχανές του τύπου αυτού δεν τυπώνουν σε ταινία, σε μερικούς όμως τύπους υπάρχει η δυνατότητα αυτή. Άλλοι τύποι τέλος έχουν πλήρες πληκτρολόγιο (σχ. 11.2γ) και άλλοι δεκαψήφιο.

Οι περισσότερες μηχανές διαθέτουν ορισμένο αριθμό καταχωριστών (Registers), συνήθως 1 - 5.



Σχ. 11.2γ.
Υπολογιστική μηχανή με κινητό πίνακα αποτελεσμάτων.

Σε μερικούς τύπους γίνεται ύψωση στο τετράγωνο και τοποθέτηση κλασματικών ψηφίων.

Χρησιμοποιούνται για στατιστική επεξεργασία δεδομένων, όπου χρειάζεται ο υπολογισμός μεγάλου αριθμού ποσοστών και λόγων. Ακόμη χρησιμοποιούνται στις εκδόσεις λογαριασμών, τις μισθοδοσίες κλπ.

δ) Λογιστικές μηχανές.

Κάτω από τον τίτλο αυτό συνωστίζεται μια μεγάλη ποικιλία από διάφορους τύπους μηχανών που χρησιμοποιούνται για εκδόσεις λογαριασμών, εγγραφές στο καθολικό, χρεώσεις, πιστώσεις, ημερολόγια, εντολές, ισοζύγια κλπ.

Τις κατατάσσουμε σε δύο κατηγορίες:

- a) Μη περιγραφικές (Nondescriptive)
- β) Περιγραφικές (Descriptive)

α) Μη περιγραφικές. Οι μηχανές αυτές διαθέτουν μόνο αριθμητικό πληκτρολόγιο (σχ. 11.2δ) και γι' αυτό δεν μπορούν να εκτυπώσουν αλφαριθμητικούς χαρακτήρες. Διαθέτουν περιορισμένο αριθμό προσθέτων πλήκτρων με ημερομηνίες και συντομογραφικά σύμβολα, που χρησιμοποιούνται για το χαρακτηρισμό διαφόρων μεγεθών ή λειτουργιών.

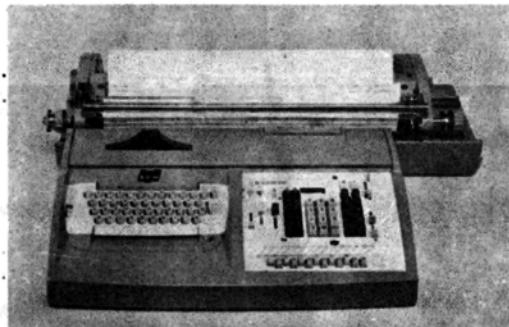
β) Περιγραφικές μηχανές. Οι μηχανές αυτές είναι εφοδιασμένες, εκτός από το αριθμητικό και με αλφαριθμητικό πληκτρολόγιο γραφομηχανής (σχ. 11.2ε), πράγμα που τις καθιστά κατάλληλες για πολλές λογιστικές δουλειές, όπως την έκδοση λογαριασμών, πιμολογίων, επιταγών κλπ., όπου απαιτείται η εκτύπωση ονομάτων, περιγραφών και άλλων αλφαριθμητικών ή αλφαριθμητικών στοιχείων.

Υπάρχουν σε τύπους δεκαψηφίου και πλήρους αριθμητικού πληκτρολογίου.



Σχ. 11.2δ.

Λογιστική μη περιγραφική μηχανή.



Σχ. 11.2ε.

Περιγραφική λογιστική μηχανή.

11.3 Ηλεκτρονικές υπολογιστικές μηχανές.

Υπάρχουν σε πολύ μεγάλη ποικιλία μεγεθών, δυνατοτήτων και χρήσεων. Θα μπορούσαμε να τις κατατάξουμε:

Ως προς τις εφαρμογές που εξυπηρετούν σε:

- α) Εμπορικές
- β) Επιστημονικές

Ως προς το μέγεθός τους σε:

- α) Επιπραπέζιες
- β) Τσέπης

Εμπορικές ηλεκτρονικές υπολογιστικές μηχανές.

Στην κατηγορία αυτή ανήκουν:

α) Ηλεκτρονικές ταμιακές μηχανές.

Διαθέτουν βασικό δεκαψήφιο αριθμητικό πληκτρολόγιο, καθώς και πλήκτρα για άλλες βιοθητικές εργασίες. Διαθέτουν επίσης αριθμό καταχωριστών-αθροιστών (1 - 20). Υπολογίζουν ποσοστά, φόρους, εκπτώσεις, καθώς και σύνολα πωλήσεων σε διάφορα επίπεδα (αναλυτικές, ημερήσιες, εβδομαδιαίες κλπ.). Εκδίδουν αποδείξεις, και τυπώνουν σε ταινία. Συγχρόνως εμφανίζουν τα διάφορα ποσά σε οθό-

νη με αμφίπλευρη ορατότητα. Η χωρητικότητα της οθόνης κυμαίνεται από 8 - 12 ψηφία με δυνατότητα παρουσίασεως και 2 κλασματικών ψηφίων (σχ. 11.3a).



Σχ. 11.3a.
Ηλεκτρονική ταμιακή μηχανή.

β) Ηλεκτρονικές αριθμομηχανές 4 πράξεων.

Τις συναντάμε σε τύπους επιτραπέζιους (Desk calculators) και τσέπης (Pocket calculators).

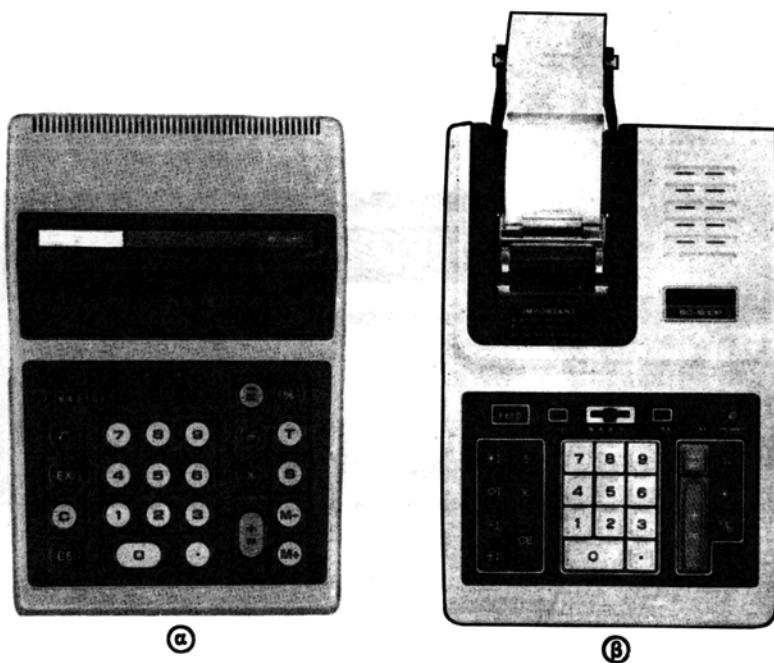
Διαθέτουν βασικό δεκαψήφιο αριθμητικό πληκτρολόγιο. Εκτός από τις 4 αριθμητικές πράξεις, έχουν δυνατότητα για σταθερό συντελεστή, ποσοστά, καθορισμό πλήθους κλασματικών ψηφίων κλπ.

Διαθέτουν επίσης 1 - 3 καταχωριστές δεδομένων καθώς και καταχωριστές εισόδου¹ και αποτελέσματος².

Σε ορισμένους τύπους, οι πληκτρολογούμενοι αριθμοί και τα αποτελέσματα εμφανίζονται σε οθόνη (με 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 έχομε 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 μιας πράξεως. Ο καταχωριστής αυτός είναι αντίστοιχος προς τον καταχωριστή - συσσωρευτή (accumulator) των μεγάλων υπολογιστών.



Ⓐ

Ⓑ

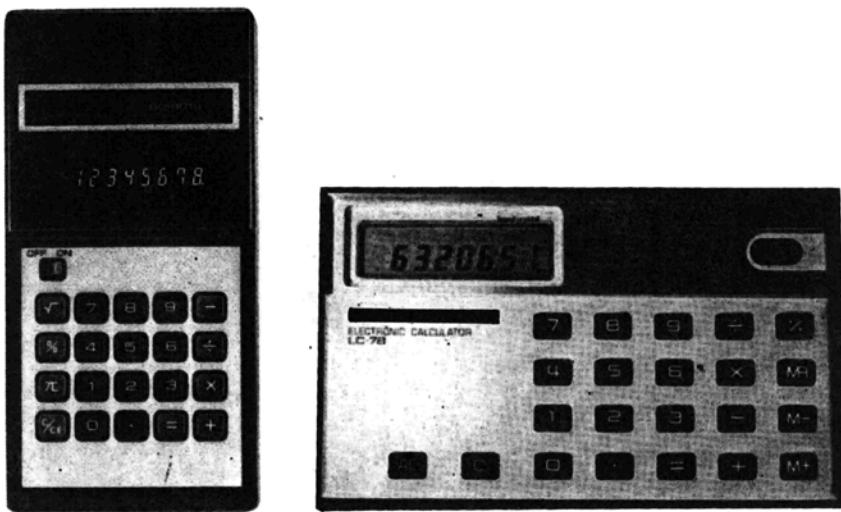


Ⓨ

Σχ. 11.38.

- α) Επιτραπέζια ηλεκτρονική αριθμομηχανή με οθόνη. β) Επιτραπέζια ηλεκτρονική αριθμομηχανή με εκτύπωση. γ) Επιτραπέζια ηλεκτρονική αριθμομηχανή με εκτύπωση και οθόνη.

Οι τύποι αριθμομηχανής ταξέπις έχουν σχεδόν τα ίδια χαρακτηριστικά με τους επιτραπέζιους, αλλά διαθέτουν μόνο οθόνη (σχ. 11.3γ).



Σχ. 11.3γ.
Ηλεκτρονικές αριθμομηχανές τσέπης.

γ) Εποπτημονικές ηλεκτρονικές υπολογιστικές μηχανές.

Διακρίνομε τις εξής τρεις κατηγορίες:

- 1) Προπρογραμματισμένες (Preprogrammed)
- 2) Προγραμματιζόμενες με πληκτρολόγηση (Programmable)
- 3) Προγραμματιζόμενες εξωτερικά με μαγνητική κάρτα, καρτελίδιο ή κασέτα (Programmable with a magnetic medium).

Οι υπολογιστές της κατηγορίας 1 εμφανίζονται κυρίως σε μοντέλα τσέπης (Pocket calculators). Στις άλλες δύο κατηγορίες υπάρχουν και επιπραπέζια και μοντέλα τσέπης.

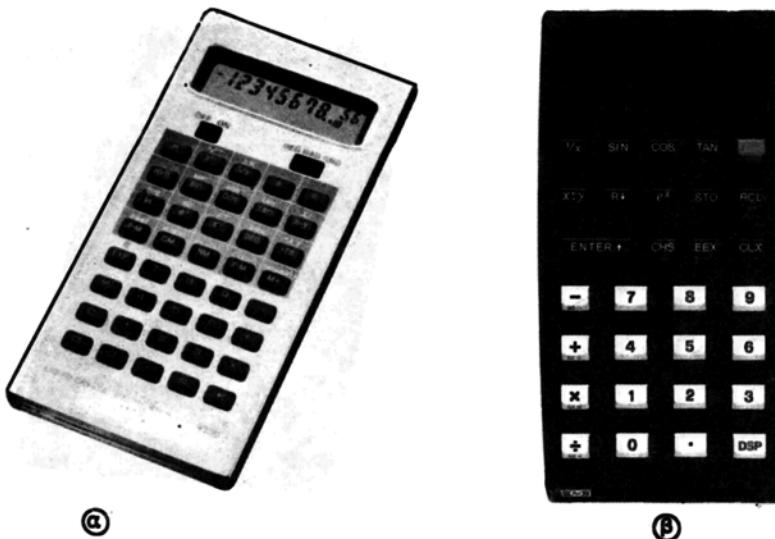
1) Προπρογραμματισμένες ηλεκτρονικές υπολογιστικές μηχανές.

Οι μηχανές αυτές εκτός από το δεκαψήφιο αριθμητικό πληκτρολόγιο, διαθέτουν και πλήκτρα, με τη βοήθεια των οποίων υπολογίζονται οι τιμές διαφόρων συναρτήσεων, όπως: τριγωνομετρικών (ημίτονο, συνημίτονο, εφαπτομένη), τετραγωνική, λογάριθμοι (φυσικοί και δεκαδικοί), αντίστροφα αριθμών, κλπ.

Διαθέτουν 1 - 10 καταχωριστές δεδομένων καθώς και καταχωριστές εισόδου και αποτελέσματος. Διαθέτουν επίσης οθόνη χωρητικότητας 8 - 12 ψηφίων. Τέλος οι περισσότερες έχουν δυνατότητα απεικόνισεως αριθμών στη μορφή της κινητής υποδιαστολής (Floating - point). Μερικές τέτοιες μηχανές βλέπομε στο σχήμα 11.3δ.

2) Προγραμματιζόμενες με πληκτρολόγηση ηλεκτρονικές υπολογιστικές μηχανές.

Οι υπολογιστές της κατηγορίας αυτής, εκτός από τις λειτουργίες μιας προπρ-



Σχ. 11.36.

(a) και (b) Προπρογραμματισμένες ηλεκτρονικές υπολογιστικές μηχανές ταέπης.

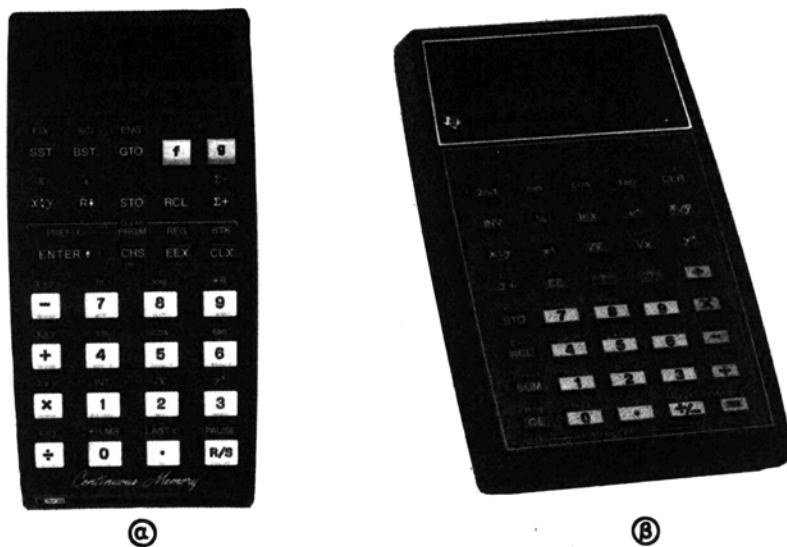
γραμματισμένης μηχανής, που μπορούν να εκτελέσουν, μπορούν ακόμη να προγραμματισθούν για να εκτελέσουν και άλλες απλές εργασίες. Ο προγραμματισμός γίνεται με εντολές της γλώσσας μηχανής τους (άρα διαφέρουν από τύπο σε τύπο). Διαθέτουν μνήμη, σε ένα μέρος της οποίας καταχωρίζεται το πρόγραμμα. Δέχονται 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. 11.3e).

3) Προγραμματιζμένες εξωτερικές 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 χρειασθούμε. Μπορούμε έτσι να σχηματίσουμε μία ολόκληρη βιβλιοθήκη από έτοιμα προγράμματα, που να επιλύουν πολλά διαφορετικά προβλήματα.

Υπάρχουν σε τύπους επιπραπέζιους 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.



Σχ. 11.3ε.

(α) και (β) Προγραμματιζόμενες με πληκτρολόγηση ηλεκτρονικές υπολογιστικές μηχανές τσέπης.



@



(α)



(β)

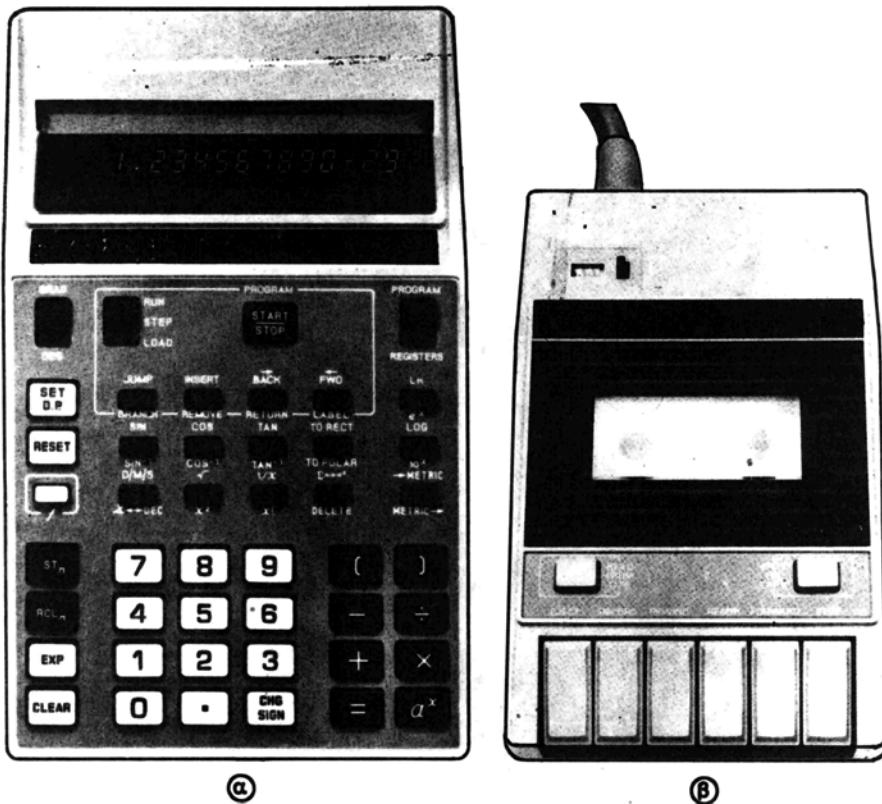
Σχ. 11.3στ.

(α) Προγράμματιζόμενη με καρτελίδιο ηλεκτρονική υπολογιστική μηχανή τούπης.(β) Προγράμματιζόμενη με καρτέλα επιτραπέζια μηχανή μόνον με εκτύπωση.(γ) Προγράμματιζόμενη με καρτελίδιο επιτραπέζια μηχανή με εκτύπωση και οθόνη.

11.4 Χειρισμός μιας ηλεκτρονικής υπολογιστικής μηχανής.

Θα χρησιμοποιήσουμε για τις παρακάτω περιγραφές μας, μια μηχανή προγραμματιζόμενη εξωτερικά που, όπως είπαμε, συγκεντρώνει όλα τα χαρακτηριστικά των μηχανών και των τριών κατηγοριών (παράγραφος 13.3).

Η μηχανή αυτή [σχ. 11.4α(α)] διαθέτει φωτεινή οθόνη χωρητικότητας 16 ψηφίων, 12 απλούς καταχωριστές δεδομένων (Data registers), ένα καταχωριστή εισ-



Σχ. 11.4α.

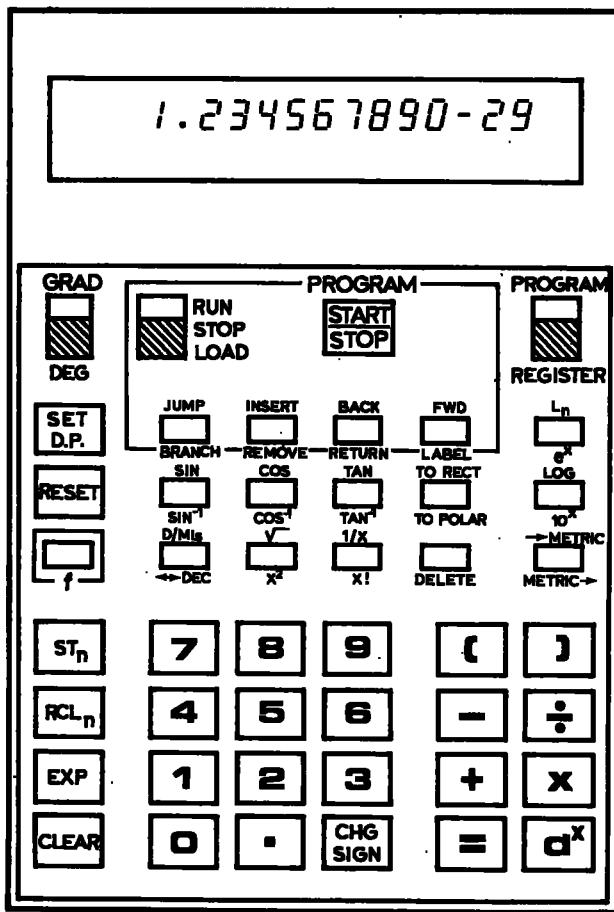
(a) Η ηλεκτρονική υπολογιστική μηχανή που θα χρησιμοποιήσουμε στη μελέτη του χειριστού (β) Η περιφερειακή μονάδα μαγνητικής ταινίας της μηχανής που εξετάζομε.

όδου (Entry register) και ένα καταχωριστή αποτελέσματος. Διαθέτει ακόμη εσωτερική μνήμη, όπου μπορούμε να αποθηκεύσουμε ένα πρόγραμμα μέχρι 160 βημάτων.

Μπορεί να συνδεθεί εξωτερικά με μια περιφερειακή μονάδα μαγνητικής κασέτας, όπου έχουμε τη δυνατότητα να εγγράψουμε όχι μόνο πρόγραμμα, αλλά και δεδομένα [σχ. 11.4α(β)].

Όπως βλέπουμε στο σχήμα 11.4α, εκτός από το βασικό δεκαψήφιο αριθμητικό πληκτρολόγιο, διαθέτει πλήκτρα συναρτήσεων καθώς και ειδικά πλήκτρα 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).

Στη συνέχεια θα περιγράψουμε συνοπτικά τις σημαντικότερες λειτουργίες και τους απαραίτητους χειρισμούς για τη κάθε μία τους, έχοντας ως οδηγό το σχ. 11.4β.



Σχ. 11.48.
Το πληκτρολόγιο της μηχανής που περιγράφουμε.

α) Εκτέλεση απλών αριθμητικών πράξεων.

Για την εκτέλεση απλών αριθμητικών πράξεων ανάμεσα σε δύο όρους της μορφής A ορ Β όπου ορ = +, -, ×, ÷.

α) Πληκτρολογούμε τον πρώτο όρο - αριθμό της πράξεως (προσθετέο, αφαιρέτεο, κλπ.).

β) Πιέζομε το πλήκτρο της επιθυμητής πράξεως.

γ) Πληκτρολογούμε το δεύτερο αριθμό.

δ) Πιέζομε το πλήκτρο [=]

Στο τέλος της φάσεως (δ), το αποτέλεσμα εμφανίζεται στην οθόνη (στην οθόνη εμφανίζονται ενδιάμεσα και όλοι οι αριθμοί τη στιγμή που πληκτρολογούνται).

Παραδείγματα.

Υπολογισμός	Εκτέλεση – Χειρισμός	Αποτέλεσμα στην οθόνη
$8 + 4$	8 + 4 =	12.0000
$16 - 5$	1 6 - 5 =	11.0000
7×6.3	7 x 6 . 3 =	44.1000
$1.8 \div 3$	1 . 8 ÷ 3 =	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 πα-

ρενθέσεις στη δοθείσα σχέση, είτε όχι. Π.χ. στη σύνθετη σχέση $3 + \frac{5}{4}$, παρ' όλο που δεν εμφανίζονται παρενθέσεις η διαδικασία πρέπει να γίνει έτσι, σαν να είχαμε γράψει τη σχέση ως εξής: $3 + (\frac{5}{4})$. Διαφορετικά, αν πληκτρολογήσομε τους όρους και τα σύμβολα με τη σειρά, όπως τα βλέπομε στην παράσταση, θα κάνομε λάθος, γιατί θα υπολογίσομε, 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 το 3). Το να βρούμε όμως τη σωστή σειρά κάθε φορά, είναι κάπι δύσκολο.

Παραδείγματα.

Γιοτολογισμός	Εκτέλεση – Χειρισμός	Αποτέλεσμα στην οθόνη
$6 + \frac{8}{4}$		8.0000
$5 \times \left(\left(\frac{2}{3} - 1 \right) + \frac{3}{4} \right)$		2.0833
$6.42 + 5 - 7 \times 2.1$		-3.2800

γ) **Η εκτέλεση συνθέτων αριθμητικών πράξεων στις αριθμομηχανές με μνήμη μιας Θέσεως.**

Πολλές από τις ηλεκτρονικές αριθμομηχανές τεσσάρων πράξεων ή και τις προπρογραμματισμένες διαθέτουν μικρή μνήμη μιας Θέσεως και μεγέθους 6 - 12 ψηφίων (ίστη και η χωρητικότητα της οθόνης τους), εκτός από τους καταχωριστές εισόδου και αποτελέσματος. Οι περισσότερες από τις μηχανές αυτές δεν διαθέτουν πλήκτρα με παρενθέσεις. Παρ' όλα αυτά μπορούν να εκτελέσουν σύνθετες αριθμητικές πράξεις με τη βοήθεια της μνήμης τους και με τη τήρηση κάποιας ιεραρχίσεως δύον αφορά στην εκτέλεση των πράξεων.

Οι μηχανές αυτές διαθέτουν τα εξής πλήκτρα:

M+ Με τη πίεση του πλήκτρου αυτού προσθέτεται το περιεχόμενο του καταχωριστή εισόδου στο περιεχόμενο της μνήμης και το άθροισμα ξανατοποθετείται στη μνήμη (Memory add).

M- Με τη πίεση του πλήκτρου αυτού, αφαιρείται το περιεχόμενο του καταχωριστή εισόδου από το περιεχόμενο της μνήμης και η διαφορά ξανατοποθετείται στη μνήμη (Memory subtract).

RM ή **MR** Με τη πίεση του πλήκτρου αυτού το περιεχόμενο της μνήμης μεταφέρεται και καταχωρίζεται στον καταχωριστή εισόδου, ενώ παράλληλα εμφανίζεται και στην οθόνη (Recall memory).

CM Με το πλήκτρο αυτό μηδενίζεται το περιεχόμενο της μνήμης (Clear memory).

Στη κατηγορία αυτή οι μηχανές αντί για τα πλήκτρα **RESET** και **CLEAR** διαθέτουν τα πλήκτρα **C** και **CE**, που μηδενίζουν το μεν πρώτο όλους τους καταχωριστές εκτός από τη μνήμη, το δέ δεύτερο, τον καταχωριστή εισόδου (Σε μερικές μηχανές υπάρχει μόνο ένα πλήκτρο το **C/CE** που μηδενίζει όλους τους καταχωριστές.

Παραδείγματα.

Υπολογισμός	Χειρισμός	Αποτέλεσμα
$\frac{2}{3} \times \frac{4}{5} \times 7$	2 ÷ 3 × 4 ÷ 5 × 7 =	3.7333324
$(9 \times 8) + (14 \times 3) - \left(\frac{51}{3}\right)$	9 × 8 = M+ 1 4 × 3 M+ 5 7 ÷ 3 M- RM	

Σημειώνομε στο σημείο αυτό ότι, αν πληκτρολογήσουμε λάθος κάποιο αριθμό, τότε απλώς πιέζομε το πλήκτρο **CLEAR** και τον ξαναχτυπάμε χωρίς να χρειάζεται να ξαναπληκτρολογήσουμε όλη τη σχέση από την αρχή.

δ) Υψωση αριθμού σε δύναμη.

Χρησιμοποιούμε το πλήκτρο **a^x** ¹. Η διαδικασία περιλαμβάνει τα εξής βήματα:

- α) Πληκτρολογούμε τον αριθμό, που είναι η βάση (a)
- β) Πιέζομε το πλήκτρο **a^x**
- γ) Πληκτρολογούμε τον αριθμό, που είναι ο εκθέτης (x).
- δ) Πιέζομε το πλήκτρο **=**

Παραδείγματα.

Υπολογισμός	Εκτέλεση – Χειρισμός	Αποτέλεσμα στην οθόνη
2^4	2 a^x 4 =	16.0000
$(3+2)^2$	3 + 2 a^x 2 =	25.0000
$2^{1.3}$	2 a^x 1 . 3 =	2.4623

1 Σε μερικές μηχανές το πλήκτρο αυτό εμφανίζεται σαν **y^x**

ε) Τρόπος χρήσεως των καταχωριστών δεδομένων (Data storage register).

Σε ένα καταχωριστή δεδομένων μπορούμε να αποθηκεύσουμε ένα αριθμό ή ένα ενδιάμεσο αποτέλεσμα μιας πράξεως για να το χρησιμοποιήσουμε αργότερα.

Κάθε καταχωριστής δεδομένων ταυτίζεται με ένα από τα πλήκτρα **0** έως **9**.

και

CHG
SIGN

Μπορούμε να καταχωρίσουμε ένα αριθμό σε ένα καταχωριστή ως εξής:

α) Πληκτρολογούμε τον αριθμό.

β) Πιέζουμε το πλήκτρο **ST_n** (Store to register).

γ) Πιέζουμε ένα από τα πλήκτρα των καταχωριστών.

Αν ο αριθμός για καταχώριση έχει προκύψει σαν αποτέλεσμα προηγούμενου υπολογισμού, τότε το βήμα (α) παραλείπεται.

Για να καλέσουμε ένα αριθμό καταχωρισμένο σε ένα καταχωριστή:

α) Πιέζουμε το πλήκτρο **RCL_n**

β) Πιέζουμε το πλήκτρο που αντιπροσωπεύει το συγκεκριμένο καταχωριστή.

Παραδείγματα:

Λειτουργία	Εκτέλεση – Χειρισμός	Αποτέλεσμα στην οθόνη
Να καταχωρθεί ο 3.14 στον καταχωριστή 1.	3 . 1 4 ST₁ 1	3.1400
Να ανακληθεί ο 3.14 από τον καταχωριστή 1.	RCL₁ 1	3.1400
Να καταχωρθεί ο 2.4 στον καταχωριστή 8.	2 . 4 CHG SIGN ST₈ 8	- 2.4000
Να ανακληθεί ο 2.4 στον καταχωριστή 8.	RCL₈ 8	- 2.4000

στ) Υπολογισμός πυών συναρτήσεων (Functions).

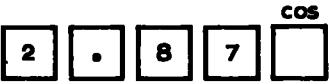
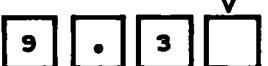
Ο υπολογιστής, που μελετούμε, έχει ένα αριθμό πλήκτρων για τον υπολογισμό της τιμής διαφόρων συναρτήσεων. Κάθε τέτοιο πλήκτρο έχει δύο λειτουργίες. Με την απλή πίεση υπολογίζεται η τιμή της συναρτήσεως που η ονομασία της αναγράφεται επάνω από το πλήκτρο.

Πιέζοντας πρώτα το πλήκτρο  και κατόπιν ένα πλήκτρο συναρτήσεως,

υπολογίζεται η τιμή της συναρτήσεως, η ονομασία της οποίας αναγράφεται κάτω από το πλήκτρο. Κατά τα άλλα η διαδικασία είναι απλή:

- Πληκτρολογούμε τὸν αριθμὸν γιὰ τὸν οποῖο θέλομε τὸν υπολογισμὸν τῆς συναρτήσεως.
- Πιέζομε τὸ πλήκτρο τῆς αντίστοιχης συναρτήσεως.

Παραδείγματα.

Υπολογισμός	Εκτέλεση - Χειρισμός	Αποτέλεσμα στὴν οθόνη
$\frac{1}{6}$		0.1667
sin(2.87)		0.9987
6.4^2		40.9600
4!		24.0000
$\sqrt{9.3}$		3.0490

σ) Υπολογισμός μαθηματικών σχέσεων που περιλαμβάνουν και συναρτήσεις.

Υπολογίζονται με τὸν ἴδιο τρόπο που υπολογίζονται καὶ οἱ σύνθετες αριθμητικές σχέσεις¹.

1. Σε κάθε περίπτωση αδύνατου υπολογισμού π.χ. διαιρέσεως ἐνδὲ αριθμοῦ με τὸ 0 ή υπολογισμού του λογαρίθμου του 0 κλπ. εμφανίζεται στὴν οθόνη ὁ χαρακτήρας E (Error = λάθος). Σε ἄλλες μηχανές στὶς περιπτώσεις αὐτές αναβοσβήνει ὁ αριθμός 0.

Παραδείγματα.

Υπολογισμός	Εκτέλεση - Χειρισμός	Αποτέλ. στην οθόνη
$4 \times \sqrt{6} + \frac{7}{4}$		11.5479
$2 \times \text{πμ}(3.1) + 6$		6.0973
$3.2 \times e^{2.1} + \sqrt{7}$		28.7774

η) Μετατροπή γωνιών.

Μετατροπή από μοίρες σε ακτίνια.

Η διαδικασία είναι:

- 1) Θέτομε το διακόπτη στην κάτω θέση (DEG)
- 2) Πληκτρολογούμε την τιμή της γωνίας (που εκφράζεται σε μοίρες και κλάσμα της μοίρας).
- 3) Πιέζομε το πλήκτρο →Metric
- 4) Πιέζομε το πλήκτρο

Παραδείγματα.

Υπολογισμός	Εκτέλεση - Χειρισμός	Αποτέλεσμα στην οθόνη
Na μετατραπούν οι 48° σε ακτίνια		0.8377
Na βρεθεί το τημίτονο των $56^{\circ},74$		0.0172

Μετατροπή από ακτίνια σε μοίρες.

Η διαδικασία είναι:

- 1) Θέτουμε το διακόπτη στην κάτω θέση.
- 2) Πληκτρολογούμε τη γωνία.
- 3) Πιέζομε το πλήκτρο
- 4) Πιέζομε το πλήκτρο metric →
- 5) Πιέζομε το πλήκτρο

Παραδείγματα.

Υπολογισμός	Εκτέλεση - Χειρισμός	Αποτελ. στην οθόνη
Να μετατραπούν τα 2.851 ακτίνια σε μοίρες.	GRAD 2 . 8 5 1 f CHG SIGN DEG	163.3502 Metric -
Να μετατραπούν τα 14.35 ακτίνια σε μοίρες.	GRAD 1 4 . 3 5 f CHG SIGN DEG	822.1944 Metric -

8) Καθορισμός πλήθους κλασματικών ψηφίων.

Γίνεται πιέζοντας πρώτα το πλήκτρο και κατόπιν τον αριθμό, που παριστάνει το επιθυμητό πλήθος κλασματικών ψηφίων.

11.5 Εφαρμογές - Παραδείγματα.

Παράδειγμα 1.

Να υπολογισθεί το βεληνεκές χ ενός βλήματος πυροβόλου, που εκτοξεύεται υπό γωνία $\theta = 10^\circ$ από τη θέση A με αρχική ταχύτητα: $V_0 = 400 \text{ m/s}$.

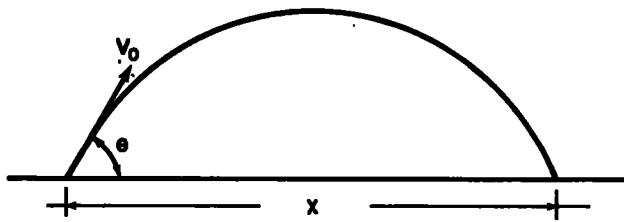
Όπως είναι γνωστό, ο τύπος του βεληνεκούς είναι:

$$\text{όπου: } x = \frac{V_0^2 \eta \mu 2\theta}{g}$$

$$\theta = 10^\circ$$

$$V_0 = 400 \text{ m/s}$$

$$g = 9,81 \text{ m/s}^2$$



Για να βρούμε το αποτέλεσμα θα κάνομε τους εξής χειρισμούς:

$$\begin{array}{ccccccccc}
 4 & 0 & 0 & \boxed{0} & \boxed{\times} & [& 2 & \times & 1 \\
 & & & f & & 0 &] & & 0 \\
 & & & x^2 & & & & & \div \\
 \hline
 9 & \cdot & 8 & 1 & = & & & &
 \end{array}
 \quad \text{SIN}$$

Το αποτέλεσμα που θα εμφανισθεί στην οθόνη θα είναι:

5040.0325

Παράδειγμα 2.

Στη θεωρία της σχετικότητας, η μάζα ενός σώματος που κινείται ισοταχώς με ταχύτητα V , υπολογίζεται από τη σχέση του *Einstein*:

$$m = \frac{m_0}{\sqrt{1 - \frac{V^2}{C^2}}}$$

όπου: m_0 η μάζα ηρεμίας του σώματος (όταν είναι ακίνητο)

C η ταχύτητα του φωτός (2.9979×10^8 m/s).

Ζητείται να υπολογισθεί η μάζα ενός ηλεκτρονίου που κινείται με ταχύτητα $0.96C$. Η μάζα ηρεμίας του ηλεκτρονίου είναι $m_0 = 9.1086 \times 10^{-31}$ kg.

Για τον υπολογισμό αυτό θα κάνομε τους εξής χειρισμούς:

$$\begin{array}{ccccccccc}
 9 & \cdot & 1 & 0 & 8 & 6 & EXP & CHG & \boxed{3} \\
 & & & & & & SIGN & & 1 \\
 & & & & & & & \div & [\\
 & & & & & & & & 1 \\
 \hline
 \cdot & 9 & . & 6 & \boxed{0} & \boxed{]} & & \boxed{=} & - \\
 & & & f & & & & &
 \end{array}
 \quad \sqrt{x^2}$$

Το αποτέλεσμα που θα εμφανισθεί στην οθόνη θα είναι:

3.253071428 - 30

11.6 Προγραμματισμός.

Η έννοια του προγράμματος και των εντολών είναι η ίδια με αυτή που γνωρίσαμε και περιγράψαμε στα προηγούμενα κεφάλαια.

Ο τύπος της μηχανής, που χρησιμοποιούμε, όπως είπαμε, διαθέτει μνήμη, όπου μπορεί να καταχωρισθεί ένα πρόγραμμα μέχρι 160 βημάτων.

Κάθε βήμα περιλαμβάνει μία εντολή γλώσσας μηχανής της οποίας η λειτουργία παριστάνεται με ένα πλήκτρο. Ο κώδικας λειτουργίας της εντολής είναι ένας τριψήφιος δεκαδικός αριθμός, που χρειάζεται να τον ξέρομε, μόνο όταν κάνομε έλεγχο για τη σωστή γραφή ενός προγράμματος.

Ο προγραμματισμός εδώ ακολουθεί τα εξής βήματα:

- Σχεδιάζεται ένα απλό διάγραμμα ροής, που αντιπροσωπεύει τη λογική που θα χρησιμοποιηθεί στην επίλυση του προβλήματος.
- Κωδικοποιείται με τη βοήθεια των εντολών της μηχανής.
- Το πρόγραμμα πληκτρολογείται και αποθηκεύεται στη μνήμη.
- Το πρόγραμμα είναι έτοιμο για εκτέλεση.

α) Προκαταρκτικοί χειρισμοί.

- Με τη βοήθεια του διακόπτη RUN/STEP/LOAD αποθηκεύεται το πρόγραμμα στη μνήμη ή προετοιμάζεται για εκτέλεση.

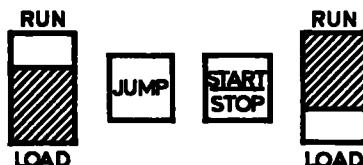
Όταν ο διακόπτης αυτός βρίσκεται στη θέση LOAD, αποθηκεύομε τις εντολές του προγράμματος σε διαδοχικές θέσεις της μνήμης, πιέζοντας τα κατάλληλα πλήκτρα.

Όταν μετά την αποθήκευση του προγράμματος ο διακόπτης αυτός τεθεί στη θέση RUN, το πρόγραμμα είναι έτοιμο για εκτέλεση.

- Με τη βοήθεια του πλήκτρου **START STOP** 1) αρχίζει η εκτέλεση του προγράμματος και 2) γίνεται η προσωρινή παύση στην εκτέλεση για να εισαχθεί κάποιο δεδομένο ή για να απεικονισθεί στην οθόνη κάποιο ενδιάμεσο αποτέλεσμα.

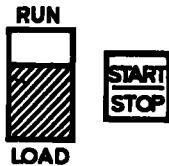
- Το πλήκτρο **JUMP** χρησιμοποιείται για áλματα από ένα σημείο του προγράμματος σε άλλο. Όταν σήμας προηγείται η **START STOP** τότε ο έλεγχος του προγράμματος μεταφέρεται στην πρώτη εντολή.

- Η διαδικασία για την αποθήκευση ενός προγράμματος στη μνήμη περιλαμβάνει τους χειρισμούς



και ακολουθεί η πληκτρολόγηση των εντολών.

- Η διαδικασία για την εκτέλεση ενός προγράμματος περιλαμβάνει τους χειρισμούς



Παράδειγμα.

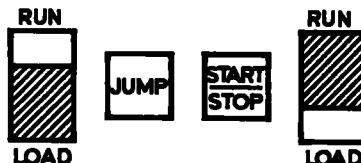
Να γραφεί πρόγραμμα για τον υπολογισμό του όγκου ενός κυλίνδρου, όταν είναι γνωστή η διάμετρος της βάσεως του d και το ύψος του h .

Ο τύπος του όγκου κυλίνδρου είναι:

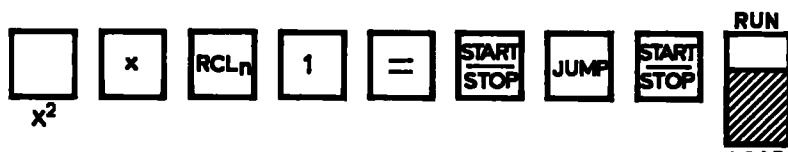
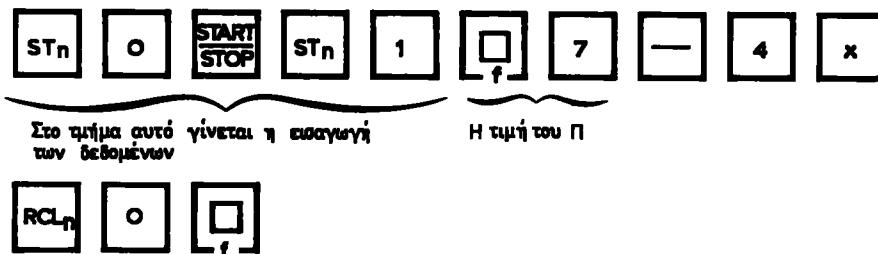
$$V = \pi \frac{d^2}{4} h$$

Το πρόγραμμα θα έχει την παρακάτω μορφή:

α) Για την αποθήκευση του προγράμματος θα γίνουν οι χειρισμοί:



β) Οι εντολές του προγράμματος θα είναι:



Στη συνέχεια δίνομε τα δεδομένα:



Το 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.

Σαν labels χρησιμοποιούνται τα αριθμητικά πλήκτρα **0** έως **9**. Το **.** και το **CHG SIGN** μαζί με τα 

Οι Labels χρησιμοποιούνται κυρίως σε προγράμματα, όπου υπάρχουν εντολές διακλαδώσεως, ή άλματος υπό συνθήκη.

γ) Εντολές άλματος υπό συνθήκη.

Με τις εντολές αυτές μπορούμε να ελέγξουμε, αν ένας αριθμός που βρίσκεται στον καταχωριστή εισόδου ¹ (Entry register) ικανοποιεί μια ιδιαίτερη συνθήκη και αν ναι, είναι δυνατό να γίνει κάποιο άλμα σε άλλο σημείο του προγράμματος. Αν η συνθήκη δεν ισχύει, το άλμα δεν πραγματοποιείται.

Έχομε τη δυνατότητα να ελέγξουμε το περιεχόμενο του καταχωριστή εισόδου για τις εξής περιπτώσεις:

Συνθήκη	Εντολή – Χειρισμός		
Μεγαλύτερο του μηδενός (> 0)	JUMP	+	
Μικρότερο του μηδενός (< 0)	JUMP	-	
Ίσο με το μηδέν ($= 0$)	JUMP	=	
Μεγαλύτερο τή ώστε του μηδενός (≥ 0)	JUMP	+	=
Μικρότερο τή ίσο του μηδενός (≤ 0)	JUMP	-	=
Διάφορο του μηδενός ($\neq 0$)	JUMP	+	-

Παράδειγμα.

Να βρεθεί το άθροισμα των 20 πρώτων φυσικών αριθμών. Το πρόγραμμα θα έχει την εξής μορφή²:

RUN LOAD	Επεξήγηση
JUMP START STOP	Προετοιμασία για την εισαγωγή των εντολών του προγράμματος στη μνήμη.
CLEAR	
ST _n O ST _n 1	Μηδενίζονται οι καταχωριστές 0 και 1.
f Label 3	Στο σημείο αυτό του προγράμματος τίθεται μια αριθμητική ένδειξη (Label), η 3.
RCL _n O ST _n + 1	Προστίθεται το περιεχόμενο του καταχωριστή 0 στο περιεχόμενο του καταχωριστή 1.
- 2 O =	Αφαιρείται ο αριθμός 20 από τον καταχωριστή εισόδου (Entry register).
JUMP = 4	Αν το περιεχόμενο του καταχωριστή εισόδου είναι ίσο με το μηδέν, γίνεται άλμα στο σημείο του προγράμματος με αριθμητική ένδειξη 4, διαφορετικά συνεχίζεται η εκτέλεση του προγράμματος με τις επόμενες εντολές.
1 ST _n + O	Αυξάνεται το περιεχόμενο του καταχωριστή 0 και 1.
JUMP 3	Γίνεται άλμα στο σημείο του προγράμματος με την αριθμητική ένδειξη 3.

¹ Στον καταχωριστή εισόδου τοποθετείται ο αριθμός που κάθε φορά πληκτρολογείται. Το περιεχόμενό του εμφανίζεται πάντα στην οθόνη.

² Η κατακόρυφη ανάπτυξη των εντολών είναι ότι εποπτική όπως δικτυωνύμεται από τον τρόπο παρουσιάσεως των παραδειγμάτων των παραγράφων 11.6(a) και 11.6(γ).

<p>1 Label RCL-n 1 START JUMP START STOP RUN</p>	<p>Στο σημείο αυτό τίθεται η αριθμητική ένδειξη 4.</p> <p>Εμφανίζεται στην οθόνη το περιεχόμενο του καταχωριστή 1.</p> <p>Επιστροφή στην αρχή του προγράμματος και προετοιμασία για την εκτέλεσή του.</p>
---	---

Το πρόγραμμα αυτό αρχίζει να εκτελείται, μόλις πιέσουμε το πλήκτρο.



Στο τέλος της εκτελέσεως θα εμφανισθεί στην οθόνη το αποτέλεσμα, που θα είναι:

210.0000

Ασκήσεις.

1. Να εκτελεσθούν οι πράξεις:

$$\begin{array}{lll} -2.012 + 4.756 & 6.24 \times 0.15 & 7.4 \div 2.1 \\ 3.1415 \times 6.28 & -100.05 + 724 & 243.1 \times 506 \end{array}$$

2. Επίσης οι πράξεις:

$$1.84 + 3.51 - 7.4 \quad 2 \times (3.12 - 1.8) + 6.54 \quad 99.9 \div 3 + 1$$

$$5.02 \times [3.1 + (2.8 + \frac{1}{6})] \quad \frac{-752 + 286}{3} \quad 951 \times 1000 - \frac{7}{6} \times 9.1$$

3. Επίσης οι πράξεις:

$$4.1 \times \eta\mu(2.87) + 6.1 \times \sigma\nu(0.13) \quad 3 \times \sqrt{1.4} - 200 \times e^{1.3} + 5.1$$

$$\frac{7.71}{6.08} \times \lambda\gamma 2.3 + 5 \times (-6.6) \quad (2.1 + 4.5)^2 + 11.2 - 6.8$$

4. Επίσης οι πράξεις:

$$\frac{7 \times [\eta\mu(2.1) + \sigma\nu(3.0)] - (4.1 + 5.2)^2}{6 \times [3 \times (4 \times (5-7) + 8) - 9]} \quad \frac{1 + 6 e^{2.1} - \sqrt{3.7}}{2 + 6 \eta\mu(0.7)}$$

5. Αυτοκίνητο κινείται με ευθύγραμμη ομαλά έπιπταχυνόμενη κίνηση. Αν η επιπτάχυνσή του είναι 4 m/s^2 και βρίσκεται σε ηρεμία 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 (εφαρμογή 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 των ηλεκτρονικών υπολογιστών.

Τους ηλεκτρονικούς υπολογιστές τους κατατάσσουμε ανάλογα με το μέγεθός τους στις εξής κατηγορίες:

α) **Μίνι-υπολογιστές (Minicomputers).** Οι υπολογιστές αυτοί πρωτοεμφανίσθηκαν στα 1966 και η εξέλιξή τους από τότε υπήρξε ραγδαία. Είναι μικροί σε μέγεθος και φθηνοί, έχουν όμως μεγάλες δυνατότητες, που εξαρτώνται κατά κύριο λόγο από το μηχανικό εξοπλισμό τους (Hardware), δηλαδή τη μνήμη, τις περιφερειακές μονάδες κλπ. Τους υπολογιστές αυτούς θα τους μέλετήσουμε λεπτομερέστερα πιο κάτω, επειδή παρουσιάζουν ξεχωριστό ενδιαφέρον. Μερικούς τύπους μίνι-υπολογιστών βλέπουμε στο σχήμα 12.1α.

β) **Μικρού μεγέθους υπολογιστές (Small-size computers).** Η διάκριση ανάμεσα στους υπολογιστές της κατηγορίας αυτής και στους μίνι-υπολογιστές είναι ασαφής. Μερικός τους συμπεριλαμβάνουν στην ίδια κατηγορία.

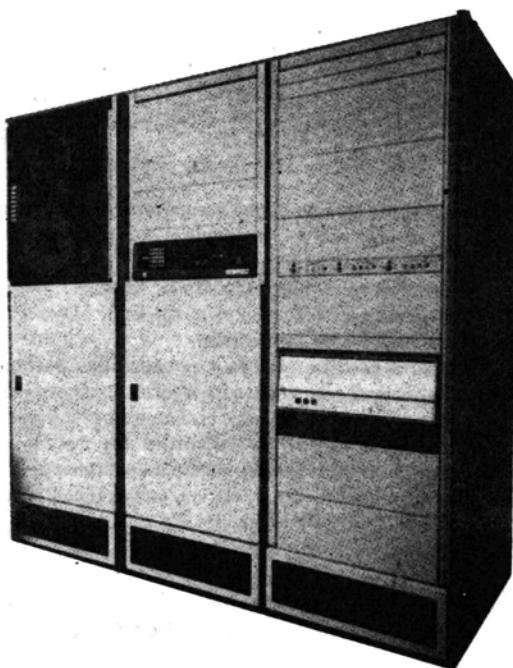
Είναι γρήγοροι υπολογιστές, με ταχύτερες περιφερειακές μονάδες από εκείνες των μίνι-υπολογιστών. Ένα σύστημα μικρού μεγέθους υπολογιστή αποτελείται βασικά από μία κεντρική μονάδα επεξεργασίας με μνήμη χωρητικότητας 4K-64K bytes· μία μονάδα αναγνώσεως-διατρίσεως δελτίων



ⓐ



(P)



Σχ. 12.1α.

(α), (β), (γ). Διάφοροι τύποι Μίνι υπολογιστών.

(80 ή 96 στηλών), 2-4 μονάδες μαγνητικής ταινίας και ένα εκτυπωτή. 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 τάχυτερους εκτυπωτές.

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, η 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.

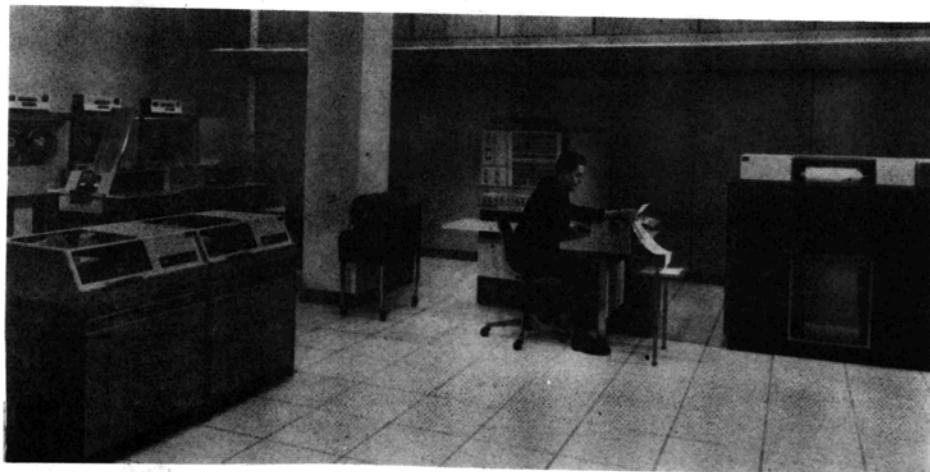
12.2 Τι είναι ένας μίνι-υπολογιστής.

Η δομή ενός μίνι-υπολογιστή είναι ίδια με τη δομή των μεγαλυτέρων ηλεκτρονικών υπολογιστών. Ο σχεδιασμός και η κατασκευή κυκλωμάτων με ελάχιστες διαστάσεις, μείωσε σημαντικά το μέγεθος των υπολογιστών.

Ένας μίνι-υπολογιστής διαθέτει μια 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.



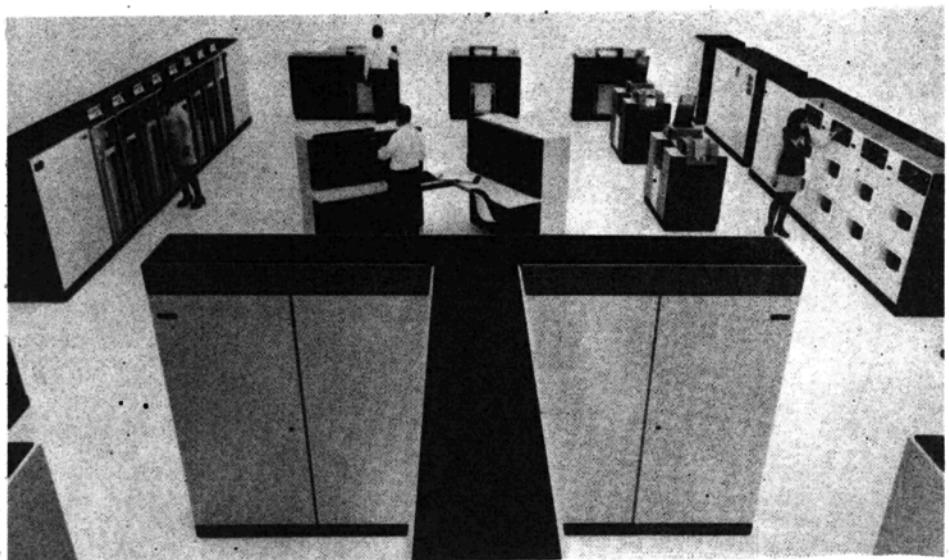
Ⓐ



Ⓑ

Σχ. 12.1γ.

(α.) (β). Διάφοροι τύποι υπολογιστών μεσαίου μεγέθους.



(α)



(β)

Σχ. 12.16.
(α), (β). Υπολογιστές μεγάλου μεγέθους.

Για τους μίνι-υπολογιστές έχουν σχεδιασθεί ειδικές μικρές περιφερειακές μονάδες, που χρησιμοποιούνται σήμερα ευρύτατα.

Οι περισσότεροι μίνι-υπολογιστές είναι υπολογιστές για γενικές εφαρμογές και εκτελούν πργράμματα, που αποθηκεύονται σε ένα μέρος της μνήμης τους.

Οι διαφορές που παρουσιάζουν με τους μεγαλύτερους υπολογιστές, συνίστανται κυρίως στο μέγεθος μιας 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 Το μέγεθος της λέξεως (Word size).

Το 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 στή συνέχεια.

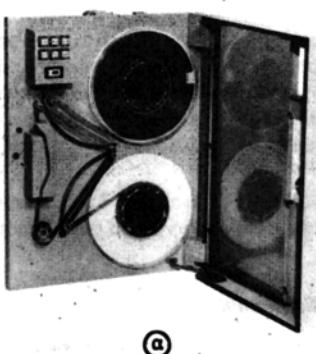
a) Μονάδες μαγνητικής ταινίας σε μικρό καρόβι (Small magnetic tape unit).

Η ταινία, που τοποθετείται σε μια τέτοια μονάδα, έχει μικρότερο μήκος από την κανονική, άρα και η χωρητικότητά της είναι μικρότερη.

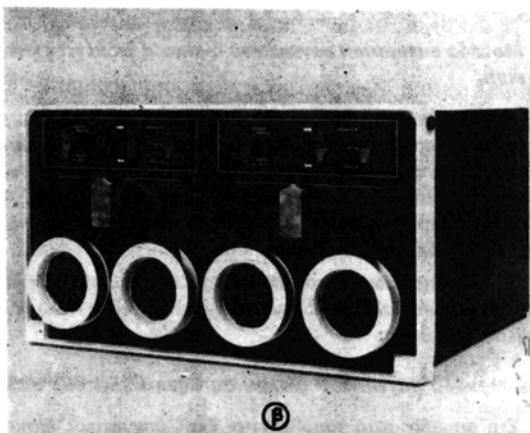
Ο όγκος της μονάδας, όπως φαίνεται και στο σχήμα 12.4a(α) είναι σαφώς πιο μικρός. Η πυκνότητα εγγραφής είναι 1600 ή 1800 bpi στα 9 κανάλια. Στο σχήμα 12.4 a(β) βλέπομε μια δίδυμη μονάδα μαγνητικής ταινίας για μίνι-υπολογιστή.

b) Μονάδα μαγνητικής ταινίας σε κασέτα (Magnetic - tape cassette).

Στη μονάδα αυτή τοποθετείται κασέτα μαγνητικής ταινίας, όμοια με τις κοινές κασέτες των φορητών μαγνητόφωνων. Η χωρητικότητα της κασέτας φθάνει τους 500.000 χαρακτήρες (σχ. 12.4b).



(a)



(b)

Σχ. 12.4a.

(a) Μια μικρού μεγέθους μονάδα μαγνητικής ταινίας. (β) Μια δίδυμη μονάδα μαγνητικής ταινίας για μίνι υπολογιστή.



Σχ. 12.4β.
Μονάδα μαγνητικής ταινίας σε κασέτα.

γ) Κωδικοποιητής μαγνητικής ταινίας σε κασέτα (Key - to - cassette unit).

Είναι μικρού μεγέθους συσκευή, με την οποία τα δεδομένα καταγράφονται κωδικοποιημένα απ' ευθείας σε κασέτα, με απλή πληκτρολόγηση. Είναι συνεπώς μια μικρογραφία του κωδικοποιητή μαγνητικής ταινίας, που είδαμε στην παράγραφο 3.18. Στη συνέχεια η κασέτα με τα δεδομένα τοποθετείται στη μονάδα μαγνητικής κασέτας έτοιμη για επεξεργασία.

δ) Μονάδα μαγνητικής ταινίας σε θήκη (Magnetic tape cartridge unit).

Η μονάδα αυτή είναι πολύ εύχρηστη. Μία θήκη μπορεί να περιέχει 300 πόδια μαγνητικής ταινίας, όπου καταχωρούνται μέχρι 2 εκατομμύρια χαρακτήρες με πυκνότητα εγγραφής μέχρι 1800 bits/inch.

Επομένως η χωρητικότητά τους είναι τετραπλάσια από όση μιας κασέτας (σχ. 12.4γ).

ε) Μονάδα εύκαμπτου μαγνητικού δίσκου ή δισκέττας (Flexible disk unit ή diskette unit ή floppy disk unit).

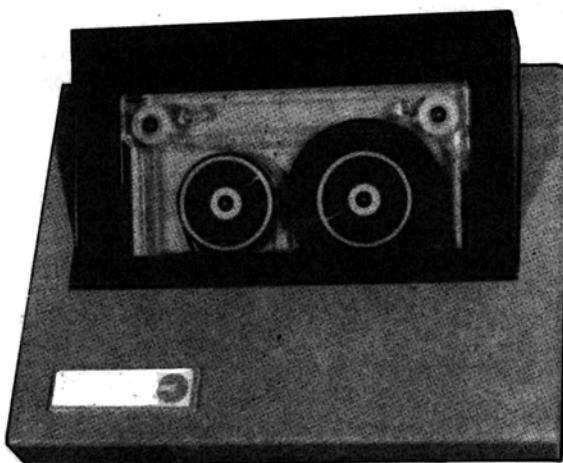
Η δισκέττα είναι ένας νέος τύπος μαγνητικού δίσκου, που άρχισε να χρησιμοποιείται πρόσφατα. Έχει μικρότερες διαστάσεις (περίπου όσο ένας μικρός δίσκος γραμμοφώνου) από τον κανονικό, είναι εύκαμπτος και η επιφάνειά του είναι καλυμμένη με ένα στρώμα μαγνητικού υλικού. Η δισκέττα είναι τοποθετημένη μέσα σε πλαστική θήκη (φάκελλο) (σχ. 12.4δ). Ολόκληρη η θήκη εισάγεται στη μονάδα δίσκου, προφυλάσσοντας έτσι το περιεχόμενό της από εξωτερικές φθορές (σκόνη, υγρασία, επαφή με τα δάχτυλα του χεριού κλπ.).

Σε μια τέτοια δισκέττα μπορούν να καταχωρισθούν μέχρι 500.000 χαρακτήρες. Το κόστος των μονάδων είναι πολύ μικρό, ενώ οι δισκέττες είναι πολύ φθηνές. Γι' αυτό και η μονάδα αυτή χρησιμοποιείται ευρύτατα.

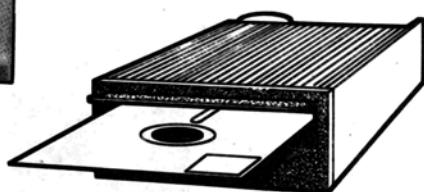
στ) Μονάδα μαγνητικού δίσκου σε θήκη (Magnetic disk cartridge unit).

Στη μονάδα αυτή τοποθετείται ένας μαγνητικός δίσκος σε θήκη (cartridge) [σχ. 12.4ε(α)].

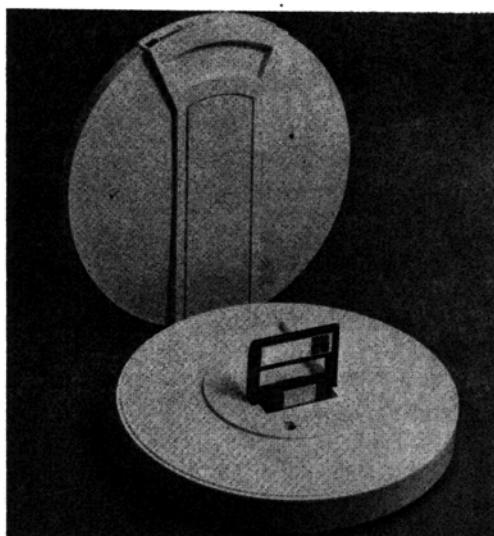
Η χωρητικότητα του δίσκου φθάνει τα 5 εκατομμύρια bytes. Στο σχήμα 12.4ε(β) βλέπομε μια δίδυμη μονάδα μαγνητικού δίσκου σε θήκη.



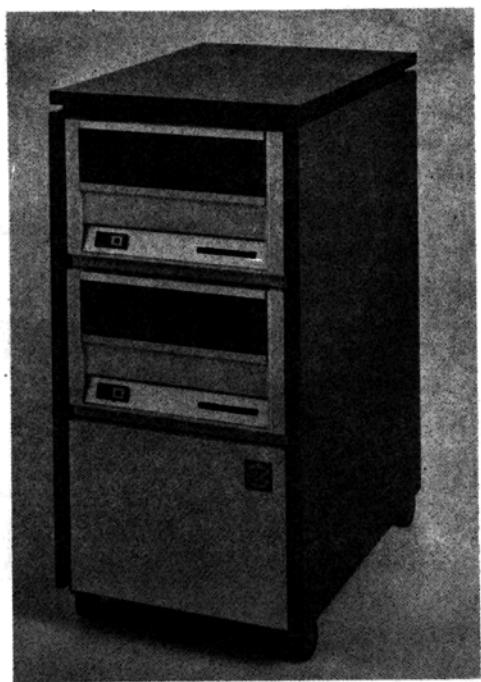
Σχ. 12.4γ.
Μονάδα μαγνητικής ταινίας σε θήκη.



Σχ. 12.4δ.
Μονάδα μαγνητικής δίσκετας και η δίσκετα.



Ⓐ



Ⓑ

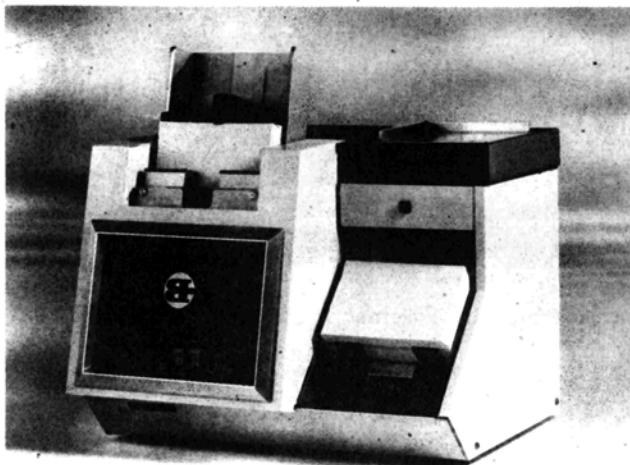
Σχ. 12.4ε.
(α) Μαγνητικός δίσκος σε θήκη. (β) Δίδυμη μονάδα μαγνητικού δίσκου σε θήκη.

Ω) Μονάδα αναγνώσεως διάπριτων δελτίων (Card reader).

Η ταχύτητα αναγνώσεως της μονάδας αυτής φθάνει τα 600 δελτία το λεπτό. Μερικοί τύποι της μονάδας αυτής μπορούν να διαβάζουν δελτία 96 στηλών, ενώ άλλοι τύποι δελτία 80 στηλών (σχ. 12.4στ.).

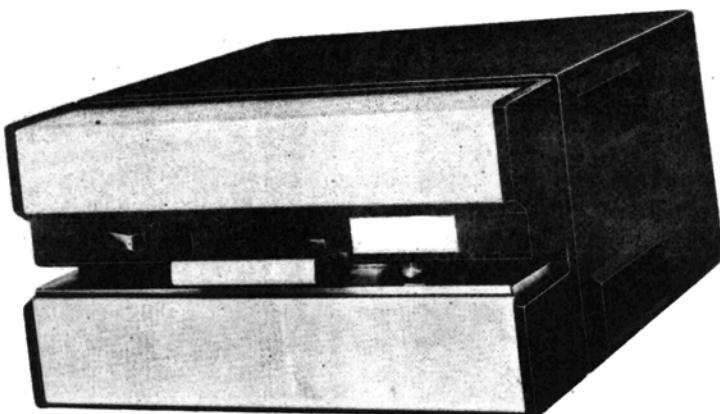
η) Μονάδα αναγνώσεως διάπριτης χαρτοταινίας (Paper tape reader).

Η ταχύτητα αναγνώσεως της φθάνει τους 700 χαρακτήρες το δευτερόλεπτο (σχ. 12.4ζ.).



Σχ. 12.4στ.

Μονάδα αναγνώσεως 80 στηλών διάπριτων δελτίων.

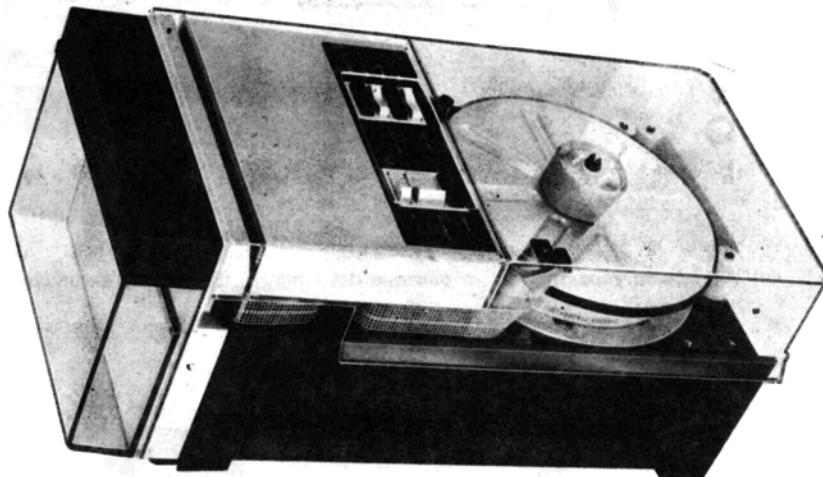


Σχ. 12.4ζ.

Μονάδα αναγνώσεως διάπριτης χαρτοταινίας.

θ) Μονάδα διατρίσεως χαρτοταινίας (Tape punch).

Μπορεί να διατρυπά απλή ή πλαστικοποιημένη χαρτοταινία με τον πενταψήφιο ή τριψήφιο κώδικα, με ταχύτητα μέχρι 80 χαρακτήρων το δευτερόλεπτό (σχ. 12.4η).



Σχ. 12.4η.
Μονάδα διατρίσεως χαρτοταινίας.

ι) Μονάδα εκτυπώσεως (Printer).

Σχεδόν όλες οι μονάδες εκτυπώσεως είναι τύπου dot matrix, δηλαδή αποτυπώνουν κάθε χαρακτήρα στο χαρτί, σχηματίζοντάς τον με τελείες (dots). Σε κάθε θέση εκτυπώσεως υπάρχει μια ορθογωνική διάταξη από 7×5 ή 7×7 σύρματα [σχ. 12.4θ(α)], τα οποία, όταν ενεργοποιηθούν, κτυπούν στο χαρτί σχηματίζοντας κάποιο χαρακτήρα.

Μια τέτοια μονάδα είναι σχεδόν αθρόυβη και μπορεί να τυπώσει μέχρι 132 χαρακτήρες σε μια γραμμή του χαρτού, με ταχύτητα μέχρι 200 γραμμές το λεπτό [σχ. 12.4θ(β)].

ια) Μονάδα σχεδίασης (Plotter).

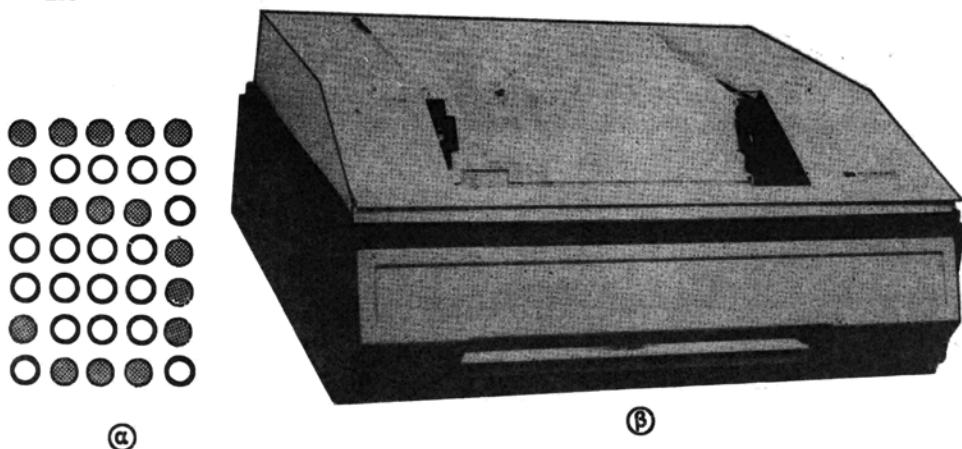
Μία τέτοια μονάδα μπορεί να σχεδιάζει διαγράμματα, και μπύλες κλπ. σε χαρτί, με διαστάσεις περίπου $14 \times 20''$ (σχ. 12.4ι).

ιβ) Οθόνη (Display).

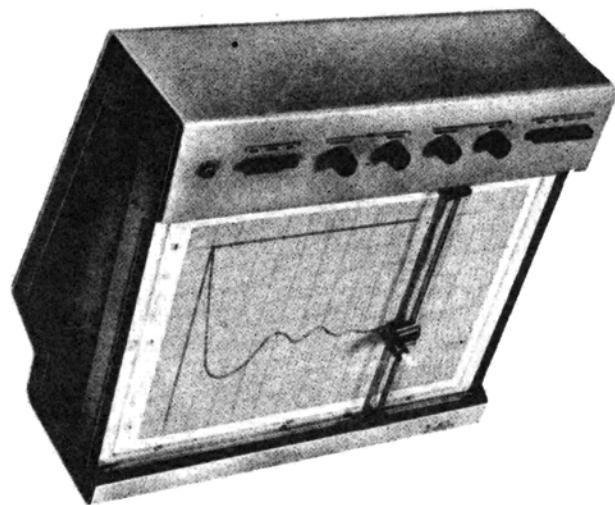
Χρησιμοποιείται όπως και στα μεγάλα συγκροτήματα υπολογιστών, σαν βασική περιφερειακή μονάδα, για την είσοδο δεδομένων και τη λήψη αποτελεσμάτων. Το σύνολο των εντολών των προγραμμάτων ή τρίματά τους μπορούν να απεικονισθούν στην οθόνη. Πολλές οθόνες είναι εφοδιασμένες με μνήμη 3K-5K, bytes (σχ. 12.4ια).

12.5 Ένα τυπικό συγκρότημα μίνι-υπολογιστή.

Ένα βασικό συγκρότημα μίνι-υπολογιστή, περιλαμβάνει μία μονάδα εισόδου-εξόδου, όπως το τη-

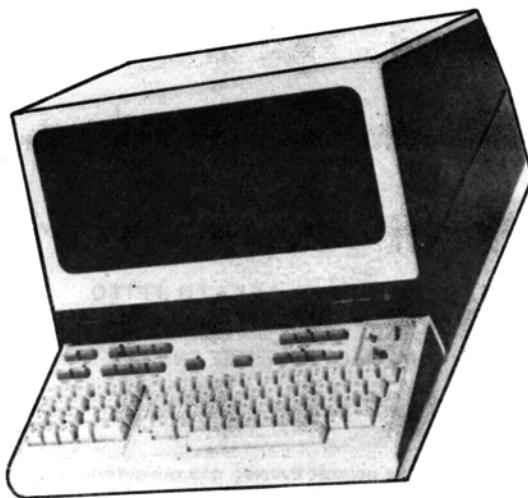
**Σχ. 12.40.**

(a) Απεικόνιση χαρακτήρα με το σύστημα dot - matrix. (β) Μονάδα εκτυπώσεως.

**Σχ. 12.41.**
Μονάδα σχεδιάσεως.

λέτυπο ή η οθόνη, μία ή περισσότερες μικρές μονάδες μαγνητικής ταινίας σε καρούλι ή σε κασέτα ή σε θήκη και μία ή δύο μονάδες διακέττας.

Όλα όμως τα συστήματα μίνι-υπολογιστών χαρακτηρίζονται από μεγάλη επεκτατικότητα (exrapability), δηλαδή μπορούν να συνδεθούν στο βασικό συγκρότημα και άλλες περιφερειακές μονάδες διαφόρων τύπων και λειτουργιών. Ο αριθμός των περιφερειακών μονάδων εξαρτάται από τις ανάγκες αυτών που χρησιμοποιούν το συγκρότημα (Users).



Σχ. 12.4α.
Όθόνη με πληκτρολόγιο.

12.6 Προγραμματισμός.

Συνήθως οι μίνι-υπολογιστές προγραμματίζονται σε απλοποιημένες εκδόσεις υψηλών γλωσσών προγραμματισμού όπως η FORTRAN, η BASIC, η RPG., επειδή η μνήμη που διαθέτουν είναι περιορισμένη. Άλλοι μίνι-υπολογιστές προγραμματίζονται σε ειδικά σχεδιασμένες συμβολικές γλώσσες, όπως η HPL κλπ.

12.7 Πλεονεκτήματα των μίνι-υπολογιστών.

Είναι πολλά τα πλεονεκτήματά τους, γι' αυτό και τα τελευταία χρόνια η προσοχή όλων των κατασκευαστριών εταιριών στράφηκε στό σχεδιασμό και την κατασκευή τέτοιων συστημάτων.

Οι μίνι-υπολογιστές, είναι μικρού μεγέθους, χαμηλού κόστους αγοράς και συντηρήσεως, με φθηνές περιφερειακές μονάδες και πολύ φθηνούς φορείς πληροφοριών. Το προσωπικό που χρειάζεται για το χειρίσμό τους είναι ελάχιστο (1 άτομο, που συνήθως είναι ο ίδιος ο προγραμματιστής). Η ταχύτητα επεξεργασίας τους είναι αρκετά ικανοποιητική, με μεγάλες δυνατότητες προγραμματισμού και ευρύ φάσμα εφαρμογών. Παρουσιάζουν ελάχιστες φθορές, που είναι πολύ εύκολο να διορθωθούν. Μερικοί τύποι μίνι-υπολογιστών εμφανίζονται τώρα και με δυνατότητα πολυπρογραμματισμού (βλ. Κεφάλαιο 13).

Οι ειδικοί προβλέπουν ότι οι μίνι-υπολογιστές όχι μόνο θα καλύψουν όλες τις ανάγκες των επιχειρήσεων, αλλά και ότι στο μέλλον θα χρησιμοποιηθούν και για την εξυπηρέτηση των διαφόρων οικιακών αναγκών.

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ ΤΡΙΤΟ

ΜΕΡΙΚΕΣ ΧΡΗΣΙΜΕΣ ΕΝΝΟΙΕΣ

13.1 Γενικά.

Στο κεφάλαιο αυτό θα γνωρίσουμε μερικές έννοιες, που συναντήσαμε στα προηγούμενα κεφάλαια και που είναι χρήσιμες και βασικές.

13.2 Τι ονομάζουμε Hardware και τι Software.

Τους όρους Hardware και Software τους ακούμε πολύ συχνά. Τι σημαίνουν όμως;

· **Hardware**, είναι το μηχανικό μέρος του υπολογιστή. Δηλαδή τα κυκλώματα, τα εξαρτήματα και οι μονάδες του.

Software, είναι όλα τα ειδικά προγράμματα, με τα οποία είναι εφοδιασμένος ο υπολογιστής και τα οποία καθοδηγούν, ελέγχουν και συντονίζουν τη λειτουργία των διαφόρων τμημάτων του.

13.3 Εποπτεύον πρόγραμμα - Μεταφραστικά προγράμματα - Λειτουργικό σύστημα.

Εποπτεύον πρόγραμμα (Supervisory program ή απλώς Supervisor), είναι ένα πρόγραμμα, οι εντολές του οποίου κατευθύνουν τη λειτουργία του υπολογιστή αξιοποιώντας παράλληλα δλες τις δυνατότητές του. Κατά τη διάρκεια της λειτουργίας, το εποπτεύον πρόγραμμα βρίσκεται αποθηκευμένο στη μνήμη.

Μεταφραστής ή μεταφραστικό πρόγραμμα (Compiler), είναι ένα πρόγραμμα που αναλαμβάνει, κάτω από τον έλεγχο του εποπτεύοντος προγράμματος.

- α) Να εξετάσει την ορθότητα συντάξεως ενός προγράμματος εφαρμογής¹ που έχει γραφεί σε κάποια από τις υψηλού επιπέδου γλώσσες προγραμματισμού,
- β) να το μεταφράσει στη γλώσσα μηχανής,
- γ) να επισημάνει τα συντακτικά λάθη του προγράμματος και να δώσει σχετικά μηνύματα (διαγνωστικά),
- δ) να συνδέσει στο πρόγραμμα δλες τις ρουτίνες² που θα χρειασθούν στην εκτέλεσή του. Υπολογιστής που πρόγραμματίζεται σε διάφορες γλώσσες, διαθέτουν ένα μεταφραστικό πρόγραμμα για κάθε γλώσσα.

Προγράμματα ειδικής χρήσεως (Utilities). Είναι βοηθητικά προγράμματα, που εκτελούν διάφορες κοινές εργασίες που εμφανίζονται συχνά σε κάποια επεξεργασία, όπως π.χ. ταξινόμηση ενός αρχείου (Sort), σύζευξη δύο αρχείων (Merge), μεταφορά δεδομένων από κάποια περιφερειακή μονάδα σε άλλη (π.χ. από ταινία σε ταινία, από ταινία σε δίσκο κ.ο.κ) κλπ.

¹ Στον όρο πρόγραμμα εφαρμογής δίνομε την έννοια που δίναμε στα προηγούμενα κεφάλαια. Η προσθήκη της λέξεως έφαρμογή γίνεται για να διακριθεί ένα τέτοιο πρόγραμμα από ένα πρόγραμμα του Software.

² Με τον όρο «ρουτίνα» εννοούμε ένα πρόγραμμα που χρησιμοποιείται για την εκτέλεση μιας τυποποιημένης διαδικασίας.

Λεπτομερές σύστημα (Operating system), είναι το σύνολο των:

- α) Εποπτεύοντος προγράμματος.
- β) Μεταφραστικού προγράμματος.
- γ) Προγραμμάτων ειδικής χρήσεως.

13.4 Μέθοδοι επεξεργασίας.

α) Σειριακή επεξεργασία προγραμμάτων (Batch processing).

Έτσι ονομάζομε την εκτέλεσή των διαφόρων προγραμμάτων εφαρμογών με τη σειρά, δηλαδή το ένα ύστερα από τό τόλλο, με τη σειρά όπως διαβάζονται από τη μονάδα αναγνώσεως δελτίων. Με τού τρόπο αυτό επεξεργάζονται τα προγράμματα εφαρμογής οι μικρού μεγέθους υπολογιστές.

Κάθε πρόγραμμα που διαβάζεται, απασχολεί όλη την κεντρική μονάδα επεξεργασίας και δεσμεύει όλες τις περιφερειακές μονάδες, έστω και αν δεν τις χρησιμοποιεί.

Πολλές φορές χρησιμοποιείται και ο συνώνυμος όρος μονοπρογραμματισμός.

β) Πολυπρογραμματισμός (Multiprogramming).

Είναι μία τεχνική επεξεργασίας προγραμμάτων εφαρμογών που εμφανίζεται κυρίως στους μεσαίους και μεγάλους μεγέθους υπολογιστές, προκειμένου να αξιοποιηθούν οι δυνατότητες της κεντρικής μονάδας επεξεργασίας και να απασχοληθούν πλήρως όλες οι περιφερειακές μονάδες.

Σύμφωνα με τη μέθοδο αυτή, πολλά προγράμματα αποθηκεύονται συγχρόνως στον υπολογιστή που αναλαμβάνει την ταυτόχρονη επεξεργασία τους, εφ' όσον τα τμήματα και οι μονάδες που διαθέτει, ανταποκρίνονται στις αντίστοιχες συνολικές απαιτήσεις όλων των προς επεξεργασία προγραμμάτων.

γ) Καταμερισμός του χρόνου επεξεργασίας (Time sharing).

Η τεχνική αυτή διαφέρει από τον πολυπρογραμματισμό κατά το ότι διάφορα προγράμματα εφαρμογών που αποθηκεύονται στον υπολογιστή δεν επεξεργάζονται ταυτόχρονα, αλλά εναλλακτικά (όηλ, πότε το ένα και πότε τόλλο, με πολύ μικρά ενδιάμεσα χρονικά διαστήματα) και κατά διαδοχικά βήματα.

Η τεχνική αυτή χρησιμοποιείται κυρίως σε συστήματα πραγματικού χρόνου - που θα δούμε στη συνέχεια - όπου πολλοί Users, (όπως λέγονται όσοι χρησιμοποιούν τον ηλεκτρονικό υπολογιστή), συνδέονται μέσω τερματικών σταθμών (Terminals) με το κεντρικό συγκρότημά του, εξυπήρετούμενοι ταυτόχρονα από αυτόν.

13.5 Συστήματα πραγματικού χρόνου (Real time systems).

Ο όρος αυτός δίνεται σε συστήματα όπου η επεξεργασία των πληροφοριών γίνεται αμέσως με την είσοδό τους στον υπολογιστή. Τα αποτελέσματα που προκύπτουν εμφανίζονται ύστερα από τόσο χρονικό διάστημα, όσο χρειάζεται ο υπολογιστής για να τα βρει. Όλα τα απαιτούμενα αρχεία και οι περιφερειακές μονάδες είναι κάθε στιγμή διαθέσιμα για να δώσουν τις διάφορες απαντήσεις.

13.6 Τηλεπεξεργασία (Teleprocessing).

Είναι ένα σύστημα επεξεργασίας πληροφοριών σε συνδυασμό με τηλεπικοινωνίες, με τη βοήθεια των οποίων γίνεται η μετάδοση και λήψη πληροφοριών ανάμεσα σε δύο σημεία που απέχουν πολύ μεταξύ τους.

Στα συστήματα αυτά υπάρχει σε κάποιο σημείο ένα κεντρικό συγκρότημα υπολογιστή, ενώ στα απομακρυσμένα σημεία υπάρχουν τερματικοί σταθμοί (Remote Terminals), που συνδέονται με το κεντρικό συγκρότημα.

Τα συστήματα αυτά είναι γνωστά επίσης και με τα ονόματα Telecommunication systems, Data communication systems, Teleprocessing systems κ.ο.κ.

13.7 Off - Line καὶ On - Line λειτουργία του υπολογιστή.

Off - Line ονομάζομε τον τρόπο λειτουργίας του υπολογιστή, όπου η διαδικασία για την επεξεργασία των πληροφοριών ακολουθεί τη σειρά: α) Συλλογή των πληροφοριών, β) καταχώρισή τους σε κάποιο φορέα, γ) επεξεργασία τους από τον υπολογιστή (όπως το σύστημα της μισθοδοσίας κλπ).

On - Line ονομάζομε τον τρόπο λειτουργίας του υπολογιστή, όπου ο πληροφορίες εισάγονται στον υπολογιστή τη στιγμή που δημιουργούνται (όπως το σύστημα των καταθέσεων στις Τράπεζες, κρατήσεις θέσεων στις Αεροπορικές Εταιρείες κλπ).

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ ΤΕΤΑΡΤΟ

ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

14.1 Γενικά.

Η μεγάλη εξέλιξη των ηλεκτρονικών υπολογιστών, ώστε να έχουν μεγαλύτερες δυνατότητες από τις παληότερες υπολογιστικές συσκευές, είναι αποτέλεσμα της ανάγκης για γρηγορότερη, καλύτερη και φθηνότερη επεξεργασία δεδομένων.

Οι εφαρμογές του ηλεκτρονικού υπολογιστή καλύπτουν σήμερα όλες τις δραστηριότητες της ανθρώπινης ζωής. Μερικές από αυτές θα αναφερθούν στα επόμενα.

14.2 Οικονομικά προβλήματα.

α) Μισθοδοσία.

Η μισθοδοσία, περιλαμβάνει μία σειρά από υπολογισμούς για κάθε υπάλληλο, ώστε να βρεθούν οι ακαθάριστες αποδοχές του (βασικός μισθός, υπερωρίες, επιδόματα κλπ.), οι κρατήσεις του (φόροι, κοινωνική ασφάλιση, δάνεια ή άλλες οφειλές προς το κράτος) και οι καθαρές αποδοχές του. Τα οικονομικά αυτά μεγέθη, που αναφέρονται σε κάθε υπάλληλο, μεταβάλλονται με το χρόνο και γι' αυτό πρέπει να γίνεται ενημέρωσή τους σε τακτά χρονικά διαστήματα.

β) Παρακολούθηση λογαριασμών των πελατών.

Τα μεγάλα καταστήματα, οι βιομηχανίες, οι τράπεζες και γενικά όλες οι οικονομικές μονάδες, που προσφέρουν αγαθά ή υπηρεσίες, πρέπει να παρακολουθούν τους λογαριασμούς των πελατών τους. Οι τράπεζες π.χ. πρέπει να έχουν μία εγγραφή για κάθε λογαριασμό πελάτη, που να τον ελέγχουν και να τον ενημερώνουν, κάθε φορά που ο πελάτης κάνει μία κατάθεση ή ανάληψη.

γ) Τίτληση αποθηκών.

Η εφαρμογή αυτή, που ανήκει στην απογραφική λογιστική, περιλαμβάνει την τίτληση εγγραφών για όλα τα υλικά, που βρίσκονται στις αποθήκες ενός καταστήματος ή ενός εργοστασίου. Οι εγγραφές αυτές περιέχουν στοιχεία, όπως η υπάρχουσα ποσότητα, η τιμή μονάδας, η ποσότητα ασφαλείας (ελάχιστη ποσότητα που πρέπει να υπάρχει στην αποθήκη, πριν γίνει η επόμενη παραγγελία) κ.ά.

δ) Σχεδιασμός - προγραμματισμός της παραγωγής.

Όταν πρόκειται να ληφθεί κάποια απόφαση για μεταβολή της παραγωγικότητας

μιας οικονομικής μονάδας. Θα πρέπει να γίνουν πολύπλοκες εκτιμήσεις, σχετικά με τό ανθρώπινο δυναμικό (ικανότητα, εκπαίδευση των υπαλλήλων, πρόσληψη νέων υπαλλήλων, καθιέρωση υπερωριών, κινήτρων κλπ.) και τον απαιτούμενο μηχανικό εξοπλισμό (αύξηση της λειτουργικότητας του ήδη υπάρχοντος, αγορά νέου, ενοικίαση χώρων κλπ.) για την επίτευξη του τελικού σκοπού.

Οι εκτιμήσεις αυτές καταγράφονται μαζί με τις έκτιμησεις στο χώρο της αγοράς (ανταγωνισμός, γενική πολιτική τιμών, εξαγωγές, προβλέψεις βραχυπρόθεσμες και μακροπρόθεσμες κλπ.) με τη μεγαλύτερη δυνατή ακρίβεια και επεξεργάζονται από τον υπολογιστή με βάση ειδικά σχεδιασμένα πρότυπα.

14.3 Προχωρημένα πληροφοριακά συστήματα.

Πολλά μηχανογραφικά συστήματα είναι απλές μετατροπές χειροκινήτων συστημάτων (Manual Systems). Δεν είναι όμως λίγοι εκείνοι, που αναγνωρίζουν το γεγονός, ότι ο υπολογιστής προσθέτει νέες διαστάσεις και ανοίγει νέους δρόμους στο σχεδιασμό πληροφοριακών συστημάτων. Μερικά από τα πιο συνηθισμένα εξελιγμένα πληροφοριακά συστήματα θα δούμε αιμέσως.

α) Συστήματα άλογληρωμένης επεξεργασίας δεδομένων (IDP - Integrated Data Processing).

Στα συστήματα αυτά η συλλογή και η κωδικοποίηση των δεδομένων σχεδιάζονται και συντονίζονται έτσι ώστε να πετυχαίνεται το καλύτερο δυνατό αποτέλεσμα με ταυτόχρονη ελαχιστοποίηση του πλήθους των απαιτουμένων λειτουργιών.

Σε ένα τέτοιο σύστημα τα βασικά δεδομένα μεταγράφονται τη στιγμή που γίνεται μια εμπορική πράξη (ή μια διλή συναλλαγή), σε κάποιο φορέα (χαρτοταπίνια, δελτία κλπ.), ο οποίος στη συνέχεια θα εισαχθεί για παραπέρα επεξεργασία στον υπολογιστή.

β) Συστήματα πραγματικού χρόνου (Real time - Systems).

Ένα τέτοιο σύστημα αποτελείται από ένα κεντρικό υπολογιστή, με τον οποίο συνδέεται ένας αριθμός από διάταξεις, μέσω των οποίων στέλνονται απ' ευθείας δεδομένα στον υπολογιστή και λαμβάνονται απαντήσεις από αυτόν. Η ονομασία «πραγματικός χρόνος» προέρχεται από το γεγονός ότι το χρονικό διάστημα, που μεσολαβεί ανάμεσα στην υποβολή της ερωτήσεως και στην απάντηση, είναι ελάχιστο και ίσο με το χρόνο που χρειάζεται ο υπολογιστής για να τη δώσει.

Όλα τα συστήματα αυτά είναι, όπως λέμε, ον-λάιν (On line), που σημαίνει ότι όλα τα απαιτούμενα αρχεία και οι περιφερειακές μονάδες είναι κάθε στιγμή διαθέσιμα για να δώσουν τις διάφορες απαντήσεις ή αποτελέσματα.

Στην κατηγορία αυτή ανήκει και το σύστημα για τις κρατήσεις θέσεων και εκδόσεως εισιτηρίων στις αεροπορικές εταιρίες.

14.4 Προβλήματα επιστημονικού προγραμματισμού.

Τέτοια προβλήματα απαιτούσαν στο παρελθόν πολύπλοκους υπολογισμούς με το χέρι από εξειδικευμένο προσωπικό και ήταν εξαιρετικά δύσκολη η επίλυσή 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 καθορίζομε 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 καταστρώνται μία σειρά από εξισώσεις και ανισώσεις, που εκφράζουν τις επιθυμητές συνθήκες και που συγκροτούν μεγάλα γραμμικά αλγεβρικά συστήματα, των οποίων η επίλυση άλλοτε ήταν πολύ δύσκολη, γιατί γινόταν με το 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 όυσχετίζουν χρονικά και καθορίζουν τις προτεραιότητές τους και τους τρόπους χρονικής διασυνδέσεώς τους. Τέτοιες τεχνικές είναι γνωστές με διάφορα ονόματα όπως PERT, CPM κλπ.

γ) Εξομοίωση συστημάτων.

Ο όρος εξομοίωση συστήματος (Systems Simulation) σημαίνει ότι αντιγράφομε 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 πρόδοση στην τεχνολογία συνοδεύθηκε και από τεράστια συσσώρευση πληροφοριών. Μέρα με τη μέρα γίνεται και πιο δύσκολη η ενημέρωση ακόμη και ειδικών επιστημόνων έπάνω 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, ευρετηριάζεται 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 μια συνοπτική περίληψη του περιεχομένου.

β) Ιατρική διάγνωση.

Για τη σωστή διάγνωση μιας ασθένειας, ο γιατρός πρέπει να σημειώσει με ακρίβεια τα συμπτώματα που παρουσιάζει ο ασθενής και να τα συγκρίνει με τα συμπτώματα που παρουσιάζονται στις διάφορες ασθένειες. Από τη σύγκριση ο γιατρός αποφαίνεται για την ασθένεια που ανταποκρίνεται περισσότερο προς την κατάσταση του αρρώστου.

Η επιβεβαίωση γίνεται με εργαστηριακές εξετάσεις, που συμπληρώνουν την εικόνα της καταστάσεως. Η δουλειά αυτή είναι εξαιρετικά δύσκολη και λεπτή. Σήμερα ο υπολογιστής προσφέρει σημαντική βοήθεια στην ιατρική επιστήμη, κυρίως στον τομέα της διαγνώσεως. Τα συμπτώματα όλων των ασθενειών έχουν καταγραφεί σε ειδικά αρχεία. Ο υπολογιστής τροφοδοτείται με τα συμπτώματα, που παρουσιάζει ο ασθενής, κάνει τις συγκρίσεις και βγάζει συμπεράσματα, εντοπίζοντας την εργαστηριακή έρευνα σε στενότερα όρια.

γ) Μετάφραση γλωσσών.

Ο υπολογιστής χρησιμοποιείται ακόμη, για τη μετάφραση κειμένων από μία γλώσσα σε άλλη. Το σύστημα λειτουργεί κυρίως για επιστημονικά και τεχνικά κείμενα, όπου χρησιμοποιείται καθορισμένη ορολογία και όπου δεν υπάρχουν ιδιωματισμοί της γλώσσας.

Καθώς μια γλώσσα αποτελείται από λέξεις, που συγκροτούνται σε φράσεις με τη βοήθεια ορισμένων κανόνων, τό ειδικό πρόγραμμα του υπολογιστή προβαίνει στη μετάφραση, βρίσκοντας τις αντίστοιχες λέξεις μιας άλλης γλώσσας και συνδυάζοντάς τις σύμφωνα με μέρικούς απλούς κανόνες.

14.6: Εκπαίδευση.

Ειδικά προγράμματα υπολογιστή εξυπηρετούν διάφορους τομείς της εκπαίδευσης.

α) Προγραμματισμένη διδασκαλία.

Αυτή η μορφή της διδασκαλίας περιλαμβάνει μικρά διδακτικά κείμενα, που η κατάνοησή τους ελέγχεται με ειδική δοκιμασία από τον υπολογιστή. Αν ο σπουδαστής ανταποκρίθει ίκανοποιητικά στην εξέταση, προχωρεί στα επόμενα, αν όχι, του επισημαίνονται οι αδυναμίες, για να μπορέσει να προετοιμασθεί κατάλληλα για μια επανεξέτασή του.

β) Διδασκαλία για τυφλούς.

Καταργώντας το πολυδάπανο σύστημα γραφής Braille, η διδασκαλία με υπολογιστή είναι πολύ πιο άνετη. Με τη βοήθεια ενός οπτικού αναγνώστη καλύπτεται το κείμενο ενός βιβλίου και μεταφέρεται σε ένα υπολογιστή, που διαθέτει μονάδα ηχητικής αποκρίσεως (Audio Response Unit). Οι λέξεις του κειμένου εκφωνούνται στη συνέχεια από μεγάφωνο.

14.7 Καθοδήγηση - έλεγχος.

Ο υπολογιστής χρησιμοποιείται επίσης και σε συστήματα κατευθύνσεως και ελέγχου.

α) Αυτόματη καθοδήγηση βλήμάτων.

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 οδηγής κυκλοφορίας.

Το έργο του ελέγχου 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 εναέριας 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 δρια, η λειτουργία των μηχανημάτων διορθώνεται ή μεταβάλλεται.

14.8 Ο υπολογιστής και οι τέχνες.

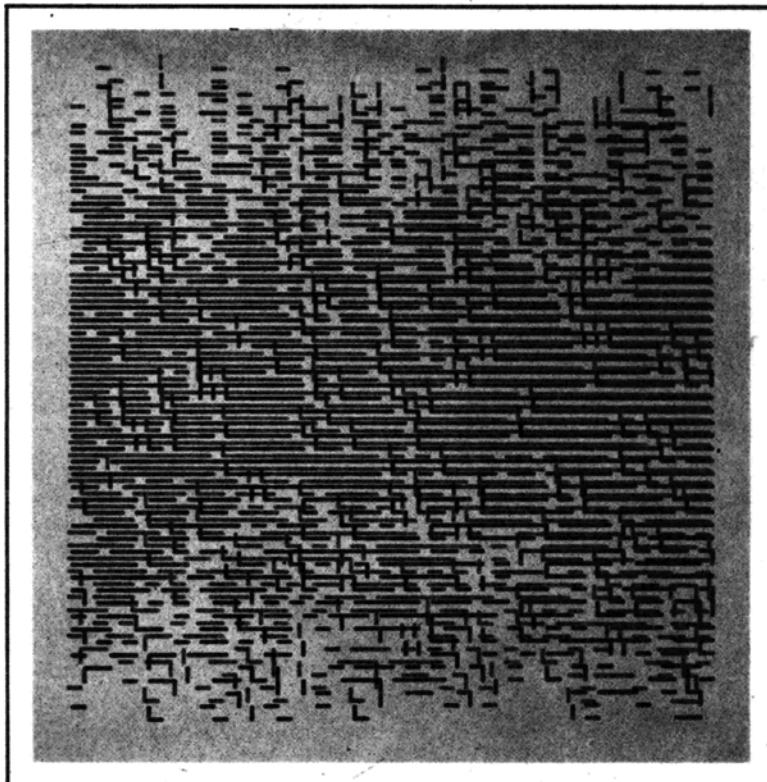
Η χρήση του υπολογιστή στις διάφορες μορφές τέχνης ξεκίνησε στην αρχή σαν παιχνίδι, αργότερα σαν πειραματισμός και τελικά εξελίχθηκε σε ολόκληρη έπιστημα.

α) Μουσική με υπολογιστή.

Όταν μια 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 υπολογιστή.

Η ζωγραφική με υπολογιστή ξεκίνησε στα 1952, αλλά η συστηματική έρευνα άρχισε στα 1957. Σήμερα έχει φθάσει σε υψηλό σημείο εξελίξεως. Ένα τέτοιο δείγμα βλέπομε στο σχήμα που ακολουθεί (σχ. 14.8).



Σχ. 14.8.
Ζωγραφική με υπολογιστή.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

Εισαγωγή στις Αριθμητικές και στους ψηφιακούς ηλεκτρονικούς υπολογιστές – Ιστορική αναδρομή

1.1	Η αρίθμηση στους αρχαίους λαούς	1
1.2	Η εμφάνιση των πρώτων υπολογιστικών οργάνων	1
1.3	Οι πρότεις αριθμητικές	3
1.4	Η ιδέα της πρότης αυτόματης υπολογιστικής μηχανής	4
1.5	Η ανακάλυψη της διάτρητης καρτέλας	5
1.6	Ο πρώτος αυτόματος υπολογιστής	6
1.7	Ρίχνοντας μια ματιά στα τελευταία επιτεύγματα της επιστήμης	9
1.8	Ο ηλεκτρονικός υπολογιστής θα υποκαταστήσει εντελώς τον δίνηρο;	9
1.9	Κατηγορίες υπολογιστικών μηχανών	10

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

Αριθμητικά συστήματα – Η αριθμητική των ψηφιακών ηλεκτρονικών υπολογιστών

2.1	Γενικά	13
2.2	Το δεκαδικό αριθμητικό σύστημα	13
2.3	Το δυαδικό αριθμητικό σύστημα	14
2.4	Το οκταδικό αριθμητικό σύστημα	14
2.5	Το δεκαεξαδικό αριθμητικό σύστημα	14
2.6	Μετατροπή αριθμών από ένα σύστημα σε κάποιο άλλο με διαφορετική βάση	15
2.7	Αριθμητικές πράξεις στο δυαδικό σύστημα	18
2.8	Αριθμητικές πράξεις στο οκταδικό σύστημα	19
2.9	Δυαδικοί κώδικες	20
2.10	Δυαδικά γηφία ιστομίας	25

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

Περιγραφή των ψηφιακών ηλεκτρονικών υπολογιστών I

3.1	Γενικά	27
3.2	Τα βασικά μέρη ενός ψηφιακού ηλεκτρονικού υπολογιστή	28
3.3	Φορείς και μέσα καταγραφής πληροφοριών – Περιφερειακές μονάδες	29
3.4	Το δελτίο (Card)	30
3.5	Μερικές χρήσιμες έννοιες	36
3.6	Κλασικές μηχανές	39
3.7	Περιφερειακή μονάδα αναγνώσεως διατρήσων δελτίων (Card reader)	46
3.8	Περιφερειακή μονάδα διατρήσεως δελτίων (Card punch)	48
3.9	Η 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 Οπτικοί αναγνώστες (Optical characters readers)	50
3.14 Περιφερειακή μονάδα εκτυπώσεως (Printer)	52
3.15 Πέριφερειακή μονάδα σχεδιάσεως (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 χαρακτήρες μαγνητικής μελάνης και αναγνώστες χαρακτήρων 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 ελέγχου (Console)	72
3.30 Τερματικοί σταθμοί (Terminals)	73
3.31 Μερικές παρατηρήσεις	73

ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

Περιγραφή των ψηφιακών τηλεκτρονικών υπολογιστών II Η κεντρική μονάδα επεξεργασίας

4.1 Η κεντρική μνήμη — Γενικά	75
4.2 Μαγνητικοί πυρήνες (Magnetic Cores)	77
4.3 Ανάγνωση της τιμής ενός πυρήνα	80
4.4 Εγγραφή μιας τιμής σε ένα πυρήνα	81
4.5 Μνήμες από τημαγωγούς	81
4.6 Άλλα είδη μνήμης	81
4.7 Οργάνωση της μνήμης	82
4.8 Καταχωριστές (Registers)	84
4.9 Φλιπ-Φλόπς (Flip-Flops)	84
4.10 Μονάδα ελέγχου (Control Unit)	85
4.11 Αριθμητική και λογική μονάδα (Arithmetic and logical unit)	85
4.12 Δίαυλοι (Channels)	85
4.13 Υπομονάδες ελέγχου των περιφερειακών (Controllers)	85

ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ

Προγραμματισμός

5.1 Εντολές — 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 εκτέλεσεως 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 διαγράμματων ροής	97
6.3 Μερικοί χρήσιμοι μαθηματικοί συμβολαισμοί	104
6.4 Η διαδικασία του προγραμματισμού	105

ΚΕΦΑΛΑΙΟ ΙΒΔΟΜΟ

Προγραμματισμός των γλώσσων μηχανής

7.1 Ένας υποθετικός εκπαιδευτικός υπολογιστής	107
7.2 Το σύνολο περγαριών των εντολών του ΤΚ1	108
7.3 Πάς λειτουργία στο ΤΚ1	111
7.4 Έντοκα κωδικογράφησης	111
7.5 Προγραμματισμός στη γλώσσα του ΤΚ1	111
7.6 Εντολές αποφάσεων	114
7.7 Τρόπος χρησιμοποίησης μαρκών απόμα εντολέν	115

ΚΕΦΑΛΑΙΟ ΟΓΔΟΟ

Γλώσσα προγραμματισμού Assembly

8.1 Γενικά	117
8.2 Διαδικασία προγραμματισμού στη γλώσσα Assembly	117
8.3 Έντοκα κωδικογράφησης Assembly (Assembly Coding Form)	117
8.4 Γενική παραγραφή του απολογιστή που θα χρησιμοποιήσουμε	119
8.5 Κατηγορίες εντολών Assembly	119
8.6 Εντολές τόκου RR	120
8.7 Εντολές τόκου RX	123
8.8 Εντολές τόκου RS	124
8.9 Εντολές τόκου SI	125
8.10 Εντολές τόκου SS1	126
8.11 Εντολές τόκου 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 Βασική στοιχεία της BASIC	135
9.3 Κατηγορίες εντολών BASIC	138
9.4 Τρόπος αναγραφής των εντολών BASIC	139
9.5 Εντολή αντικαταστάσεως. Η εντολή LET	139
9.6 Εντολές αισθητού—εξέδον	141
9.7 Μαρκές ακόμη εντολές	143
9.8 Μαρκές απλά προγράμματα BASIC	143
9.9 Εντολές ελέγχου και διεκλαδώσεως	144
9.10 Εντολές FOR-NEXT	148
9.11 Πίνακες (Arrays). Μεταβλητές με δείκτες	150
9.12 Εντολή DIM	151

9.13 Παραδείγματα	152
9.14 Επεξεργασία αρχείων στη BASIC	152
9.15 Υποπρογράμματα (Subprograms)	154

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ

Η γλώσσα προγραμματισμού FORTRAN

10.1 Γενικά	158
10.2 Γενικά χαρακτηριστικά της FORTRAN	158
10.3 Εκφράσεις FORTRAN (FORTRAN expressions)	160
10.4 Συναρτήσεις του συστήματος	162
10.5 Παρενθέσεις	163
10.6 Ιεραρχία στην εκτέλεση των πράξεων	163
10.7 Λογικές εκφράσεις	163
10.8 Κατηγορίες εντολών FORTRAN	165
10.9 Περιγραφή του εντόπου κωδικογραφήσας FORTRAN (FORTRAN Coding Form) ..	165
10.10 Αριθμητική εντολή αντικειμενότητος	165
10.11 Εντολές ελέγχου και διακλαδόσεων	167
10.12 Εντολή DO	171
10.13 Μερικές απλές εφαρμοσής	172
10.14 Εντολές αισθόσου - εξόδου	173
10.15 Εντυπο σχεδίασμασύ εκτυπώσεων (Printer spacing chart)	175
10.16 FORMAT	176
10.17 Επινιώμημανυμενοι κωδικοι FORM.VI	181
10.18 Δηλωτικές εντολές	182
10.19 Μερικές απόμη εντολές	183
10.20 Ειδικές μορφές των εντολών READ και WRITE	183
10.21 Σχόλια (Comments)	183
10.22 Υποπρογράμματα (Subprograms)	184
10.23 Εντυπο αναγραφής των δεδομένων (Data form)	189
10.24 Εφαρμογές	189
10.25 Διαδικασία προετοιμασίας ενός προγράμματος FORTRAN	195

ΚΕΦΑΛΑΙΟ ΕΝΔΕΚΑΤΟ

Υπολογιστικές μηχανές γραφίσιου – Αριθμοριγονές

11.1 Γενικά	199
11.2 Κλασικές υπολογιστικές μηχανές	200
11.3 Ηλεκτρονικές υπολογιστικές μηχανές	203
11.4 Χειρισμός μιας ηλεκτρονικής υπολογιστικής μηχανής	209
11.5 Εφαρμογές – Παραδείγματα	218
11.6 Προγραμματισμός	220

ΚΕΦΑΛΑΙΟ ΔΩΔΕΚΑΤΟ

Μίνι-υπολογιστές

12.1 Γενική ταξινόμηση των ηλεκτρονικών υπολογιστών	226
12.2 Τι είναι ένας μίνι-υπολογιστής	228
12.3 Το μέγεθος της λέξεως (Word size)	231
12.4 Περιφερειακές μονάδες	231
12.5 Ένα τυπικό συγκρότημα μίνι-υπολογιστή	235
12.6 Προγραμματισμός	237
12.7 Πλεονεκτήματα των μίνι-υπολογιστών	237

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ ΤΡΙΤΟ

Μακριές χρόνιμες έννοιες

13.1	Γενικά	238
13.2	Τι ονομάζουμε Hardware και τι Software	238
13.3	Εποκτύσιον πρόγραμμα – Μεταφραστικά προγράμματα – Λαϊτουργικό σύστημα	238
13.4	Μέθοδοι επεξεργασίας	239
13.5	Συστήματα πραγματικού χρόνου (Real time systems)	239
13.6	Τηλεπεξεργασία (Teleprocessing)	239

ΚΕΦΑΛΑΙΟ ΔΕΚΑΤΟ ΤΕΤΑΡΤΟ

Εφαρμογές των τηλεκτρονικών υπολογιστών

14.1	Γενικά	241
14.2	Οικονομικά προβλήματα	241
14.3	Προχωρημένα πληροφοριακά συστήματα	242
14.4	Προβλήματα επιστημονικού προγραμματισμού	242
14.5	Αναζήτηση πληροφοριών	243
14.6	Εκπαίδευση	244
14.7	Κεθοδήγηση – Θλεγχος	245
14.8	Ο υπολογιστής και οι τέχνες	245