ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ

ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΤΕ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΡΙΘΜΟΣ 1407

ΘΕΜΑ:ΕΦΑΡΜΟΓΕΣ ΜΙΚΡΟΕΛΓΚΤΩΝ AVR.

ΕΙΣΗΓΗΤΗΣ: Κος ΚΑΡΕΛΗΣ ΔΗΜΗΤΡΙΟΣ

ΣΠΟΥΔΑΣΤΗΣ:ΡΟΜΠΟΤΗΣ ΣΠΥΡΙΔΩΝ

ΠΑΤΡΑ 2014

ΠΡΟΛΟΓΟΣ

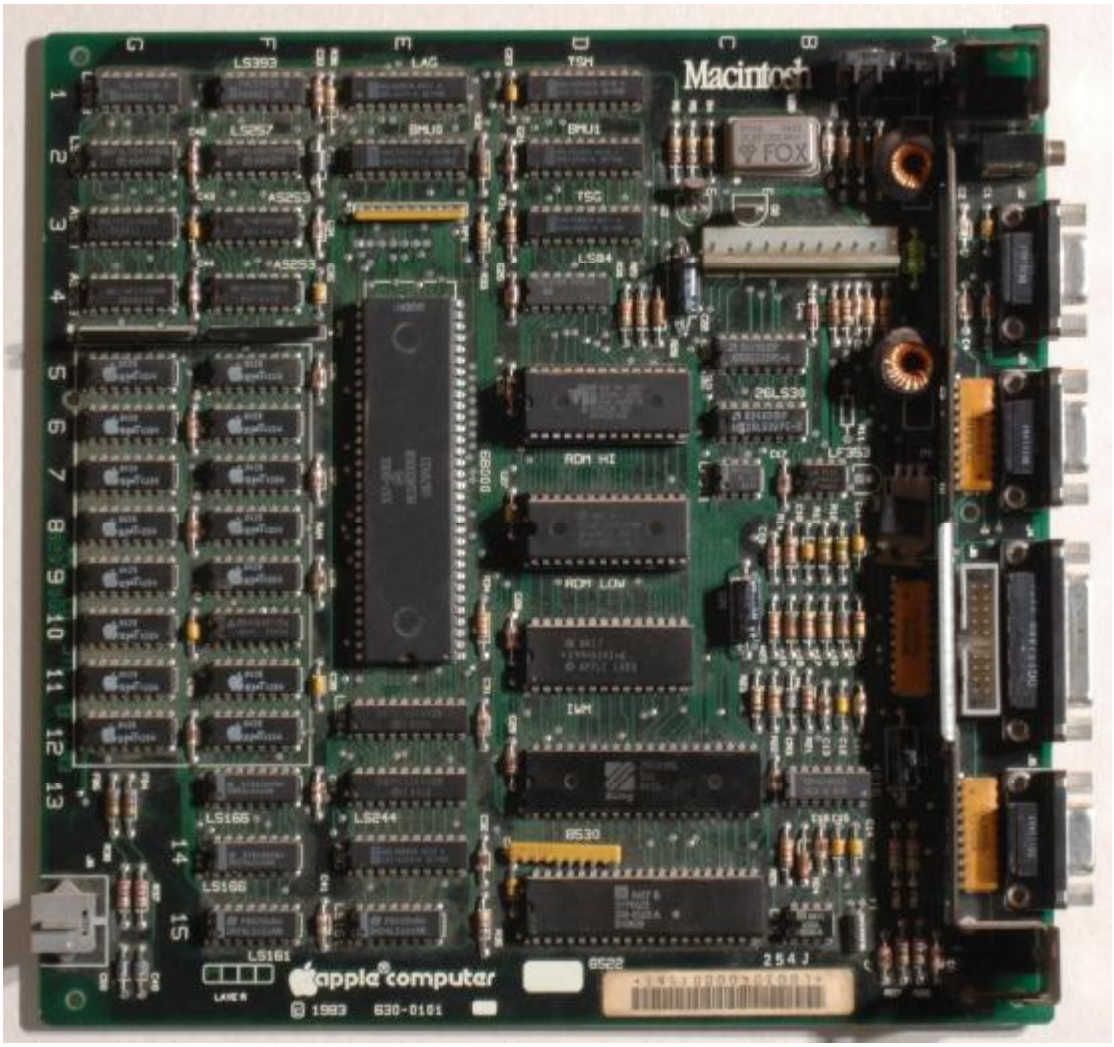
Αν και δεν υπάρχει αυστηρός ορισμός, ονομάζουμε **μικροϋπολογιστή** έναν υπολογιστή που διαθέτει μικροεπεξεργαστή ως κεντρική μονάδα επεξεργασίας. Άλλο ένα χαρακτηριστικό αυτών των υπολογιστών είναι ότι καταλαμβάνουν λίγο χώρο. Ο μικροϋπολογιστής εμφανίστηκε μετά τους μίνι υπολογιστές, με πιο αξιοσημείωτη αλλαγή την αντικατάσταση όλων των εξαρτημάτων που συναποτελούσαν την κεντρική μονάδα επεξεργασίας του μίνι υπολογιστή με έναν ολοκληρωμένο μικροεπεξεργαστή. Πολλά μοντέλα στην αρχή ήταν απλοϊκά στον σχεδιασμό, με τους πρώτους μικροεπεξεργαστές να έχουν την λειτουργικότητα των τσιπ των υπολογιστών χειρός (αριθμομηχανές). Ωστόσο, καθώς ο σχεδιασμός των μικροεπεξεργαστών αναπτύχθηκε ραγδαία από το 1970 και μετά, οι μικροϋπολογιστές έγιναν γρηγορότεροι και φτηνότεροι με παράλληλη αύξηση της δημοτικότητάς τους.

Οι επιτραπέζιοι υπολογιστές, οι παιχνιδομηχανές, οι φορητοί υπολογιστές, οι υπολογιστές χειρός και άλλες παρόμοιες συσκευές μπορεί να θεωρηθούν ότι ανήκουν στην κατηγορία των μικροϋπολογιστών.

Ο περισσότερος εξοπλισμός που χρησιμοποιείται σε έναν μικροϋπολογιστή είναι ενσωματωμένος σε ένα μόνο κουτί, αν και μερικά μέρη μπορούν να βρίσκονται σε μικρές αποστάσεις και να συνδέονται με το κουτί, όπως π.χ. το πληκτρολόγιο, το ποντίκι , η οθόνη κτλ. Γενικά ένας μικροϋπολογιστής δεν μπορεί να γίνει μεγαλύτερος, σε σημείο που να μην μπορεί να τοποθετηθεί πάνω σε ένα γραφείο. Αντίθετα ένας μίνι υπολογιστής, mainframe ή υπερυπολογιστής μπορεί να επεκτείνεται στον χώρο με την προσθήκη υπομονάδων φτάνοντας να καταλαμβάνει ολόκληρα δωμάτια.

Άλλα υποσυστήματα που αποτελούν μέρος ενός ολοκληρωμένου συστήματος μικροϋπολογιστή είναι το τροφοδοτικό και συσκευές εισόδου/εξόδου, όπως οι εκτυπωτές, η οθόνη και άλλες συσκευές διασύνδεσης με τον χρήστη.

Ο πρώτος στον κόσμο εμπορικός μικροεπεξεργαστής ήταν ο 4004 της Intel, που κυκλοφόρησε στις 15 Νοεμβρίου του 1971. Ο 4004 επεξεργαζόταν 4 δυαδικά ψηφία (BITS) δεδομένων παράλληλα, δηλαδή ήταν τετράμπιτος (4 bit). Τριάντα χρόνια μετά, οι μικροϋπολογιστές σε ενσωματωμένα συστήματα (ευρισκόμενα σε οικιακές συσκευές, οχήματα και κάθε είδους εξοπλισμό) είναι συνήθως 8-bit, 16-bit, 32-bit, ή 64-bit. Επιτραπέζιοι μικροϋπολογιστές όπως ο Apple Macintosh και τα PC είναι κυρίως 16-bit ή 32-bit αλλά σταδιακά γίνονται 64-bit, ενώ οι σταθμοί εργασίας που χρησιμοποιούν οι επιστήμονες και οι μηχανικοί και οι υπερυπολογιστές είναι 64-bit (με μία ή περισσότερες ΚΜΕ). Η πρώτη γενιά των μικροϋπολογιστών, που προορίζονταν για ειδικούς ηλεκτρολόγους μηχανικούς και για προσωπική χρήση από χομπίστες, ξεκίνησε στα μέσα της δεκαετίας 1970, με τον MITS Altair να αποτελεί το πιο γνωστό παράδειγμα. Το 1977 ξεκίνησε να κυκλοφορεί η δεύτερη γενιά, γνωστή και ως οικιακοί υπολογιστές. Αυτοί ήταν πιο εύκολοι στη χρήση από τους προκατόχους τους, οι οποίοι απαιτούσαν εξοικείωση με πρακτική ηλεκτροτεχνία. Το κρίσιμο γεγονός που μετέτρεψε το μικροϋπολογιστή από ένα χόμπι μιας μικρής ομάδας ανθρώπων σε εργαλείο για την επιχείρηση ήταν η διάθεση στην αγορά των προγραμμάτων VisiCalc(λογιστικό φύλλο) (αρχικά για τον Apple II). Μετά την έλευση από την IBM του IBM PC το 1981 ο όρος προσωπικός υπολογιστής σήμαινε κυρίως μικροϋπολογιστές συμβατούς με την αρχιτεκτονική του IBM PC (PC συμβατός).


Macintosh
LS393
LAG
TSM
LS257
BMU0
BMU1
AS253
TSC
LS04
26LS30
ROM HI
LF353
ROM LOW
IWM
8530
LS166
LS244
LS161
6522
254J
apple computer
1983 630-0101

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 –ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΕΣ

1.1 Εισαγωγή

Ο μικροϋπολογιστής είναι ενας <<μικρός>> υπολογιστης και αποτελείται από τα εξής μέρη όπως

1. **Τον επεξεργαστη η cpu**
2. **Την μνημη (RAM,ROM,Flash)**
3. **Τα κυκλωματα εισοδου- εξοδου η όπως λεγονται I/O (input/output).Τα I/O χρησιμοπιουνται για την διασυνδεση περιφεριακων <<εξωτερικων>> συσκευων όπως η οθονη,το πληκτρολογιο,ο εκτυπωτης,ο δισκος αποθηκευσης κλπ.**

Όλα αυτά τα στοιχειά είναι τοποθετημένα πάνω σε μια πλακέτα (board).Ο μικροϋπολογιστής είναι το υπολογιστικό σύστημα πάνω στο οποίο αναπτυχθηκαν,αφενός οι προσωπικοί υπολογιστές (PC's) και αφετέρου τα σύγχρονα υπολογιστικά συστήματα βασισμένα στην αρχιτεκτονική της κατανεμημένες η παράλληλης επεξεργασίας (distributed /parallel processing).Πέρα από αυτά ο μικροϋπολογιστής σήμερα υπάρχει μέσα σε οποιαδήποτε ηλεκτρονική η ηλεκτρική συσκευή χρειάζεται υπολογιστική νοημοσύνη.Ετσι ο μικροϋπολογιστής υπάρχει μέσα στο αυτοκίνητο (ABS,Injection,cruise control,αερόσακοι,διαγνωστικα,κλπ.),στις προσωπικές μας συσκευές (κινητό τηλεφωνο,ρολοι,ηλεκτρονικη ατζεντα,έξυπνες κάρτες,κλπ.)και αλλού.

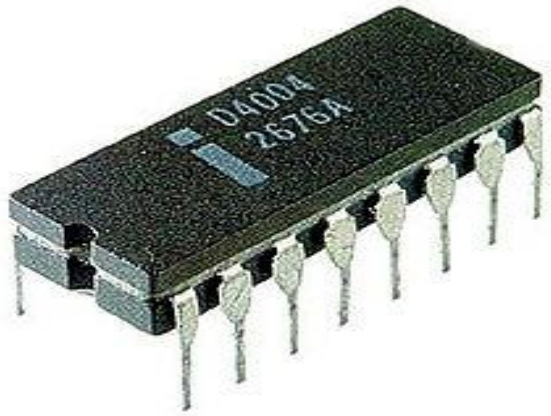
Δεν χρειάζεται να αναφερθεί η βιομηχανία βιοτεχνία και η εν γένει παράγωγη.Όλοι οι βιομηχανικοί αυτοματισμοί (PLC,industrial controllers,AC/DC Drives,Inverters,Converters,κλπ.) βασίζονται στην <<νοημοσύνη>> που παρέχει ο μικροϋπολογιστής.

1.2 Ο Μικροεπεξεργαστης

Η κεντρική επεξεργαστική μονάδα η Central Processing Unit η CPU η απλά ο μικροεπεξεργαστής αποτελεί το βασικότερο μέρος ενός μικροϋπολογιστή. Οι πρώτοι μικροεπεξεργαστές εμφανίστηκαν στις αρχές της δεκαετίας του 1970 και χρησιμοποιήθηκαν σε ηλεκτρονικές αριθμομηχανές. Η ενσωμάτωση των μικροεπεξεργαστών σε άλλες συσκευές, όπως τερματικά, εκτυπωτές κλπ, ακολούθησε σχετικά γρήγορα. Με χρήση ενός οκτάμπιτου μικροεπεξεργαστή, κατασκευάστηκε ο πρώτος μικροϋπολογιστής γενικού σκοπού στα μέσα της δεκαετίας του 1970. Η ραγδαία ανάπτυξη της τεχνολογίας των μικροεπεξεργαστών που ακολούθησε συνδέεται με τις αυξημένες απαιτήσεις από γλώσσες προγραμματισμού υψηλού επιπέδου. Ένας σύγχρονος μικροεπεξεργαστής αποτελείται από τις ακόλουθες μονάδες.

- **Καταχωρητές** (registers) Μικρά κελιά μνήμης στο εσωτερικό του επεξεργαστή, που χρησιμοποιούνται για την προσωρινή αποθήκευση των δεδομένων, καθώς αυτά υφίστανται επεξεργασία. Οι καταχωρητές διαφέρουν ανάλογα με τον τύπο του επεξεργαστή και τον κατασκευαστή, τόσο ως προς την οργάνωση όσο και ως προς τη χωρητικότητά τους.
- **Αποκωδικοποιητή Εντολών** (Command Decoder) για την μετατροπή μιας εντολής προγράμματος σε μια σειρά λειτουργειων,μεσω των οποίων υλοποιείται η εν λογω εντολή.
- **Αριθμητική και Λογική Μονάδα** (Arithmetic and Logical Unit, ALU): Η μονάδα στην οποία εκτελούνται μία προς μία οι αριθμητικές ή λογικές πράξεις, όπως υπαγορεύονται από τις εντολές που έχουν δοθεί στον υπολογιστή.
- **Κυκλωματα Χρονισμου και Ελεγχου** (Timing/Control):τα όποια με ψηφιακό ρολόι και ψηφιακούς απαριθμητές παράγον δευτερεύοντα περιοδικά και μη σηματα,για να ελέγχουν και διατηρούν την σωστή ακολουθία των λειτουργιών στα πλαίσια της εκτέλεσης εντολών προγράμματος.
- **Εσωτερικούς διαύλους επικοινωνίας** (Internal Bus) για την επικοινωνία όλων των μονάδων μεταξύ αυτων,μεσα στην CPU.
- **Διασύνδεση των εσωτερικών διαύλων με τους εξώτερικους διαύλους** (Data Address /Control Buses Interfaces) για την επικοινωνία της CPU με τα άλλα τμήματα του μικροϋπολογιστή όπως μνήμη και I/O μέσω των τριών δίαυλων δεδομένων διευθύνσεων/ Έλεγχου.

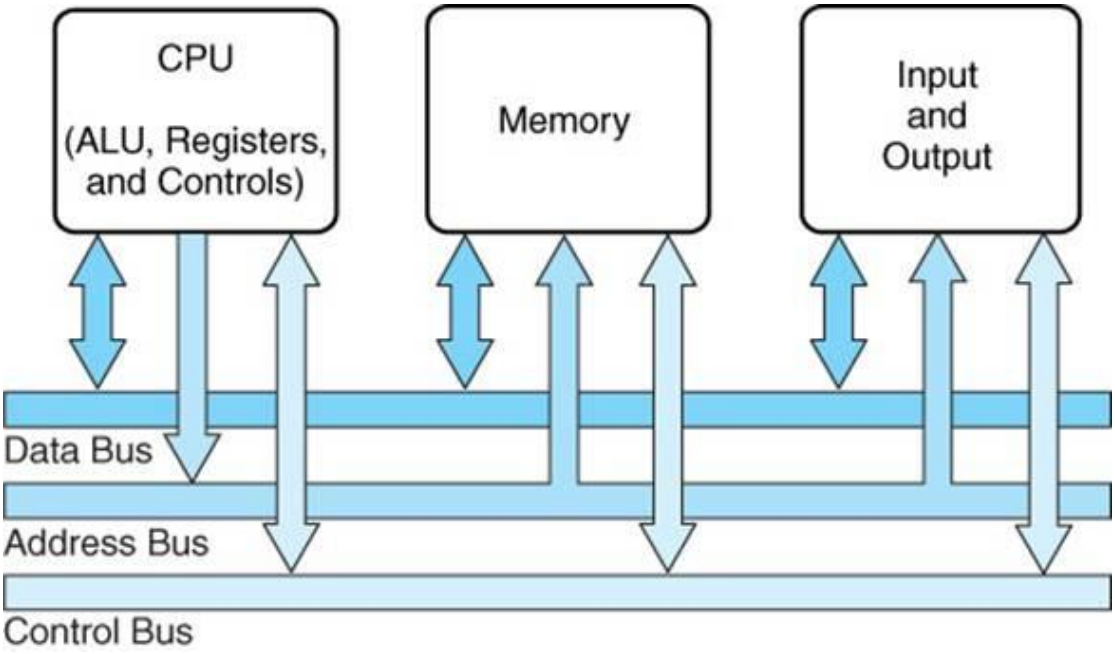
Η CPU υλοποιείται από την συνεργασια και συνλειτουργία όλων αυτών των προαναφερομένων μερών η μονάδων και είναι <<συγκεντρωμένη>> μέσα σε ένα chip.



Σχήμα 1.2: Ο **Intel 4004** ήταν ο πρώτος ολοκληρωμένος μικροεπεξεργαστής 4-bit, «όλα σε ένα», μαζικής παραγωγής και κυκλοφόρησε το 1971 από την εταιρεία Intel. Η φράση «όλα σε ένα» σημαίνει ότι σε ένα ολοκληρωμένο κύκλωμα περιλαμβάνονταν το ρολόι χρονισμού, η εσωτερική μνήμη με τη μορφή καταχωρητών (registers) και η μονάδα αριθμητικών υπολογισμών (ALU). Ήταν το πρώτο προϊόν της μετέπειτα θρυλικής εταιρείας *Intel* που σχεδιάστηκε και κατασκευάσθηκε για χρήση σε επιτραπέζιες αριθμομηχανές όπως η αριθμομηχανή *141 PF* της Busicom, ενώ δικαιολογούσε πλήρως την έννοια του ολοκληρωμένου κυκλώματος περιέχοντας 2.300 τρανζίστορ.

1.3 Βασική Δομή ενός Μικροϋπολογιστή

Η δομή αυτή φαίνεται στο παρακάτω σχήμα (εικόνα 1.3).Τα τρία μέρη του μικροϋπολογιστή ,δηλαδή η CPU,η μνήμη και τα I/O επικοινωνούν με τρεις διαύλους επικοινωνίας ,τον δίαυλο δεδομένων (Data Bus),τον δίαυλο διευθύνσεων (Address Bus) και τον δίαυλο έλεγχου (control bus).



Σχήμα 1.3

Η υπολογιστική ισχύς γενικά καθορίζεται από τα εξής στοιχειά

- Την ταχύτητα (το ρολόι) της CPU (πχ 700 MHz η 2GHz).Όσο υψηλότερη τόσο ισχυρότερος.
- Το εύρος του διαύλου δεδομένων (πχ 8-bit, 16-bit,32-bit,64-bit).Οσο περισσότερα bits τόσο ισχυρότερος.
- Το εύρος του διαύλου διευθύνσεων (πχ 16-bit 64 kbytes,32-bit 4 Gbytes).Όσο μεγαλύτερος τόσο περισσότερη μνήμη μπορεί να διευθυνσιοδοτηθεί και τόσο ισχυρότερο είναι το σύστημα
- Το πλήθος των εσωτερικών καταχωρητών της CPU.Όσο περισσότεροι τόσο ισχυρότερος.
- Την παρουσία ειδικής μνήμης (cache) δίπλα στην CPU.

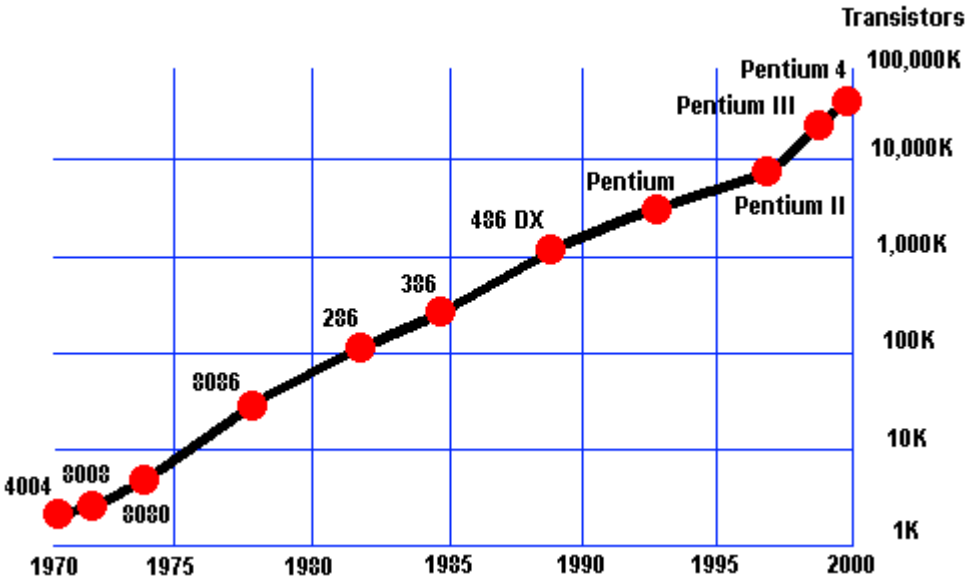
1.4 Σχεδιαστές μικροεπεξεργαστών

Ο πρώτος μικροεπεξεργαστής έκανε την εμφάνιση του στις αρχές του 1972, σχεδόν τρεις δεκαετίες μετά από τους πρώτους ηλεκτρονικούς υπολογιστές. Η εξέλιξη των μικροεπεξεργαστών θυμίζει πολύ την αντίστοιχη εξέλιξη των μεσαίων υπολογιστών. Όπως δηλαδή οι σχεδιαστές των μεσαίων υπολογιστών μετέφεραν σε αυτούς τις ιδέες τους από τη σχεδίαση μεγάλων συστημάτων, έτσι και οι σχεδιαστές των μικροεπεξεργαστών υιοθέτησαν πολλά στοιχεία της οργάνωσης και της αρχιτεκτονικής των μεσαίων και μεγάλων συστημάτων. Στους μικροεπεξεργαστές της τελευταίας γενιάς άρχισαν ήδη να εφαρμόζονται προχωρημένα στοιχεία αρχιτεκτονικής, με αποτέλεσμα σήμερα να είναι ασαφής ο διαχωρισμός ανάμεσα στους μεσαίους υπολογιστές και σε συστήματα βασισμένα σε μικροεπεξεργαστές.

Οι πιο σημαντικές και ευρέως διαδεδομένες εταιρίες σχεδιασμού μικροεπεξεργαστών της αγοράς είναι οι ακόλουθες:

- Advanced Micro Devices - Advanced Micro Devices, κατασκευαστής επεξεργαστών κυρίως για προσωπικούς υπολογιστές.
- ARM Ltd - μια από τις λίγες εταιρίες που δεν κατασκευάζει επεξεργαστές, αλλά παραχωρεί δικαιώματα κατασκευής επεξεργαστών τις σε τρίτους. Η αρχιτεκτονική ARM είναι από τις πλέον δημοφιλής στην κατηγορία των μικροελεγκτών για ενσωματωμένα (embedded) συστήματα.
- Freescale Semiconductor (πρώην τμήμα της Motorola) - σχεδιαστής αρκετών ενσωματωμένων και SoC επεξεργαστών για PowerPC.
- IBM Microelectronics - Μικροηλεκτρονικό τμήμα της IBM, αναπτύσσει επεξεργαστές που προορίζονται για POWER και PowerPC υπολογιστές, αλλά και πολλών επεξεργαστών που χρησιμοποιούνται σε κονσόλες βιντεοπαιχνιδιών.
- Intel Corp - Intel, κατασκευαστής αξιοσημείωτων μικροελεγκτών και επεξεργαστών, όπως οι 8051, IA-32, IA-64 και XScale. Επιπλέον παράγει περιφερειακά τσιπ για χρήση με τους επεξεργαστές της.
- MIPS Technologies - ανέπτυξαν την αρχιτεκτονική MIPS, μια πρωτοπόρος σχεδίαση των RISC επεξεργαστών.
- NEC Electronics - ανέπτυξαν την αρχιτεκτονική 78K0 8-bit, την αρχιτεκτονική 78K0R 16-bit, και την αρχιτεκτονική V850 32-bit.

- Sun Microsystems - Sun Microsystems, ανέπτυξαν την αρχιτεκτονική SPARC, μια σχεδίαση RISC.
- Texas Instruments - Σχεδιαστές και κατασκευαστές αρκετών διαφορετικών χαμηλής κατανάλωσης μικροελεγκτών μεταξύ των άλλων ημιαγωγικών προϊόντων τους.
- Transmeta - Δημιουργοί επεξεργαστών χαμηλής κατανάλωσης, όπως οι Crusoe και Efficeon, που είναι συμβατοί με το σετ εντολών x86.
- VIA Technologies - Κατασκευαστής, από την Ταϊβάν, χαμηλής κατανάλωσης επεξεργαστών, συμβατοί με το σετ εντολών x86.
- Atmel - Κατασκευαστής δημοφιλών μικροελεγκτών CISC και RISC χαμηλού κόστους, όπως η σειρά AVR.



Σχημα1.4 στο σχήμα απεικονίζεται η ραγδαία εξέλιξη των μικροϋπολογιστών από το 1970 έως το 2000.

1.5 Ο Μικροελεγκτης

Ο **Μικροελεγκτής** (αγγλικά, *microcontroller*) είναι ένας τύπος επεξεργαστή, ουσιαστικά μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα (embedded systems) ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα.

Ο Μικροελεγκτής είναι ένας υπολογιστής χωρίς περιφερειακά, σε ένα ολοκληρωμένο κύκλωμα. Πρόκειται για μια κατηγορία μικροεπεξεργαστή, ο οποίος βρίσκει εφαρμογή στις ηλεκτρονικές συσκευές, στις τηλεπικοινωνίες και στη βιομηχανία. Κάθε συσκευή, μηχάνημα ή ηλεκτρονικό σύστημα εκτός από τον έλεγχο της σωστής και αποδοτικής λειτουργίας του, χρειάζεται επίσης χειρισμό, ρύθμιση και παρακολούθηση. Το ζητούμενο στις εφαρμογές αυτές είναι να βρίσκονται στην ίδια ψηφίδα με το μικροεπεξεργαστή και οι περιφερειακές μονάδες. Τα ολοκληρωμένα κυκλώματα που περιλαμβάνουν στην ίδια ψηφίδα εκτός από το Μικροεπεξεργαστή και θύρες για είσοδο – έξοδο δεδομένων (σε παράλληλη και σειριακή μορφή), μετατροπείς αναλογικών σημάτων σε ψηφιακή μορφή και αντίστροφα (A/D και D/A), χρονιστές και μνήμες (ROM και RAM) ονομάζονται Μικροελεγκτές (Microcontrollers). Ο Μικροελεγκτής είναι υπεύθυνος για την είσοδο και έξοδο, την επεξεργασία, αποθήκευση και μετάδοση των αναλογικών και ψηφιακών σημάτων μιας εφαρμογής.

Για να κατανοήσουμε πόσο σημαντικός είναι ο Μικροελεγκτής μέσα σε ένα μιροϋπολογιστικό σύστημα, αρκεί να εστιάσουμε την προσοχή μας σε μια περίπτωση όπου δεν έχουμε στη διάθεσή μας μια διάταξη μικροελεγκτή. Έστω λοιπόν ότι διαθέτουμε μόνο μία απλή μονάδα κεντρικής επεξεργασίας (CPU). Για να κατασκευάσουμε ένα σύστημα ικανό να συνδεθεί με διάφορες εξωτερικές διατάξεις (κινητήρες, μονάδες απεικόνισης και άλλες), θα χρειαστούμε εξωτερική μνήμη προγράμματος και μνήμη δεδομένων ή RAM, εκτός των άλλων περιφερειακών διατάξεων που απαιτούνται για τη διασύνδεση της CPU με τις εξωτερικές μονάδες, δηλαδή τους κινητήρες, τις μονάδες απεικόνισης, τους αισθητήρες κ.λπ. Κάτι τέτοιο θα ήταν εντελώς εξωπραγματικό και ο αριθμός των απαιτούμενων εξαρτημάτων για μια τέτοια υλοποίηση θα ήταν τεράστιος.

Όταν όλες οι διακριτές μονάδες ενός υπολογιστή που αποτελούν έναν μικροϋπολογιστή τοποθετηθούν μέσα στο σώμα του ίδιου ολοκληρωμένου κυκλώματος, η διάταξη που προκύπτει καλείται Μικροελεγκτής. Αποτελείται από τη CPU, μνήμες RAM και ROM και μονάδα διασύνδεσης περιφερειακών, ενώ χρησιμοποιείται κυρίως για να ελέγχει συσκευές. Μοναδικό μειονέκτημα που φαίνεται να εμφανίζει είναι το γεγονός ότι για να γράψει κάποιος ένα πρόγραμμα χρειάζεται προσομοιωτή.

Αναλυτικά, τα πλεονεκτήματα των μικροελεγκτών είναι:

· Αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.

· Η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, οδηγεί σε χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιεί το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.

· Χαμηλό κόστος.

· Μεγαλύτερη αξιοπιστία, και πάλι λόγω των λιγότερων διασυνδέσεων.

· Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.

· Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους-εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος), λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.

· Μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

Ένας μικροελεγκτής ωστόσο εμφανίζει και ορισμένα μειονεκτήματα:

· Δεν αλλάζει το πρόγραμμά του, αφού είναι γραμμένο στη ROM.

· Είναι δύσκολος ο προγραμματισμός του.

Έχει μεγάλο χρόνο ανάπτυξης. Για να ολοκληρωθεί ένα προϊόν μπορεί να απαιτηθεί

1.6 Πρόσθετες Λειτουργίες Μικροελεγκτών

Ανάλογα με την εφαρμογή για την οποία προορίζεται ένας μικροελεγκτής, μπορεί να περιέχει και:

· Μία ή περισσότερες ασύγχρονες σειριακές θύρες επικοινωνίας (Universal Asynchronous Receiver Transmitter, UART).

· Σύγχρονες σειριακές θύρες επικοινωνίας (πχ I^2C, SPI, Ethernet).

· Ολόκληρα υποσυστήματα για την άμεση υποστήριξη από υλικολογισμικό (hardware) των πιο σύνθετων πρωτοκόλλων επικοινωνίας όπως CAN, HDLC, ISDN, ADSL.

· Μονάδα άμεσης εκτέλεσης πράξεων κινητής υποδιαστολής (Floating Point Processing Unit, FPU), η οποία είναι πάντοτε πιο γρήγορο από την ALU του επεξεργαστή. Τέτοιες μονάδες χαρακτηρίζουν τους μικροελεγκτές με δυνατότητες ψηφιακής επεξεργασίας σήματος (Digital Signal Processing, DSP). Τα τελευταία χρόνια, με την ευρύτατη διάδοση των φορητών συσκευών ήχου και εικόνας, παρατηρείται μια τάση σύγκλισης των μικροελεγκτών με τους DSP.

· Περισσότερες από μία εισόδους για μετατροπή αναλογικού σήματος σε ψηφιακό (Analog to Digital converter, ADC).

· Μετατροπέα ψηφιακού σε αναλογικό σήμα (Digital to Analog converter, DAC).

· Ελεγκτή οθόνης υγρών κρυστάλλων (Liquid Crystal Display, LCD).

· Υποσύστημα προγραμματισμού πάνω στο κύκλωμα (τύπου ISP, βλ. παραπάνω). Χάρη σε αυτό το κύκλωμα, είναι δυνατός ο επαναπρογραμματισμός (αναβάθμιση λογισμικού) της εφαρμογής, συνδέοντας στη συσκευή μια εξωτερική συσκευή προγραμματισμού (συνήθως σε θύρα UART RS232) ή ακόμη και από το διαδίκτυο. Αυτή η δυνατότητα απαιτεί την προϋπαρξη λογισμικού υποδοχής (bootstrap) μέσα στη μνήμη προγράμματος και επομένως δεν μπορεί να γίνει σε τελείως άδεια μνήμη προγράμματος.

· Υποσύστημα προγραμματισμού (τύπου ISP) και διάγνωσης (συνήθως είναι το καθιερωμένο πρότυπο JTAG). Χάρη σε αυτό, είναι δυνατός ο προγραμματισμός της μνήμης προγράμματος χωρίς να προαπαιτείται κάποιο πρόγραμμα υποδοχής. Γι αυτό το λόγο, είναι ιδιαίτερα χρήσιμο στον αρχικό προγραμματισμό, πχ κατά τη συναρμολόγηση, ή σε περίπτωση προβλήματος (bug) στο λογισμικό υποδοχής το οποίο να καθιστά αδύνατη την κανονική αναβάθμιση.

1.7 Διαδεδομένες κατηγορίες μικροελεγκτών

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Έτσι διακρίνονται οι εξής κυρίως κατηγορίες:

· Μικροελεγκτές (καμιά φορά 4-bit αλλά συνήθως 8-bit) πολύ χαμηλού κόστους, γενικής χρήσης, με πολύ μικρό αριθμό ακροδεκτών (ακόμη και λιγότερους από 8). Σχεδιάζονται με

έμφαση στη χαμηλή κατανάλωση ισχύος και την αυτάρκεια, ώστε να χρειάζονται ελάχιστα ή και καθόλου εξωτερικά εξαρτήματα και να μη μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους. Απουσιάζει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά στους ερασιτέχνες ηλεκτρονικούς, όπως πχ οι περισσότεροι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και 8051 (Intel, Atmel, Dallas κα).

· Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης, με μέτριο έως σχετικά μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I^2C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. Στους κατασκευαστές της Άπω Ανατολής (Ιαπωνία, Κορέα), συνηθίζεται η ενσωμάτωση ελεγκτών οθόνης υγρών κρυστάλλων και πληκτρολογίου. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης της μνήμης τους.

· Μικροελεγκτές (κυρίως 32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό ακροδεκτών. Χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερσιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Πχ μεταξύ των μικροελεγκτών τύπου ARM ή MIPS, το σύνολο των βασικών εντολών που αναγνωρίζει η ALU είναι ακριβώς το ίδιο, μειώνοντας έτσι τις μεγάλες αλλαγές στο λογισμικό, όταν στο μέλλον ο πελάτης υιοθετήσει ένα μικροελεγκτή άλλου κατασκευαστή (αρκεί, φυσικά, να υποστηρίζει κι αυτός το σύνολο εντολών ARM ή MIPS, αντίστοιχα).

· Μικροελεγκτές εξειδικευμένων εφαρμογών, οι οποίοι ενσωματώνουν συνήθως κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας το οποίο υλοποιείται πάντοτε σε hardware. Τέτοιοι μικροελεγκτές χρησιμοποιούνται σε τηλεπικοινωνιακές συσκευές όπως Modem.

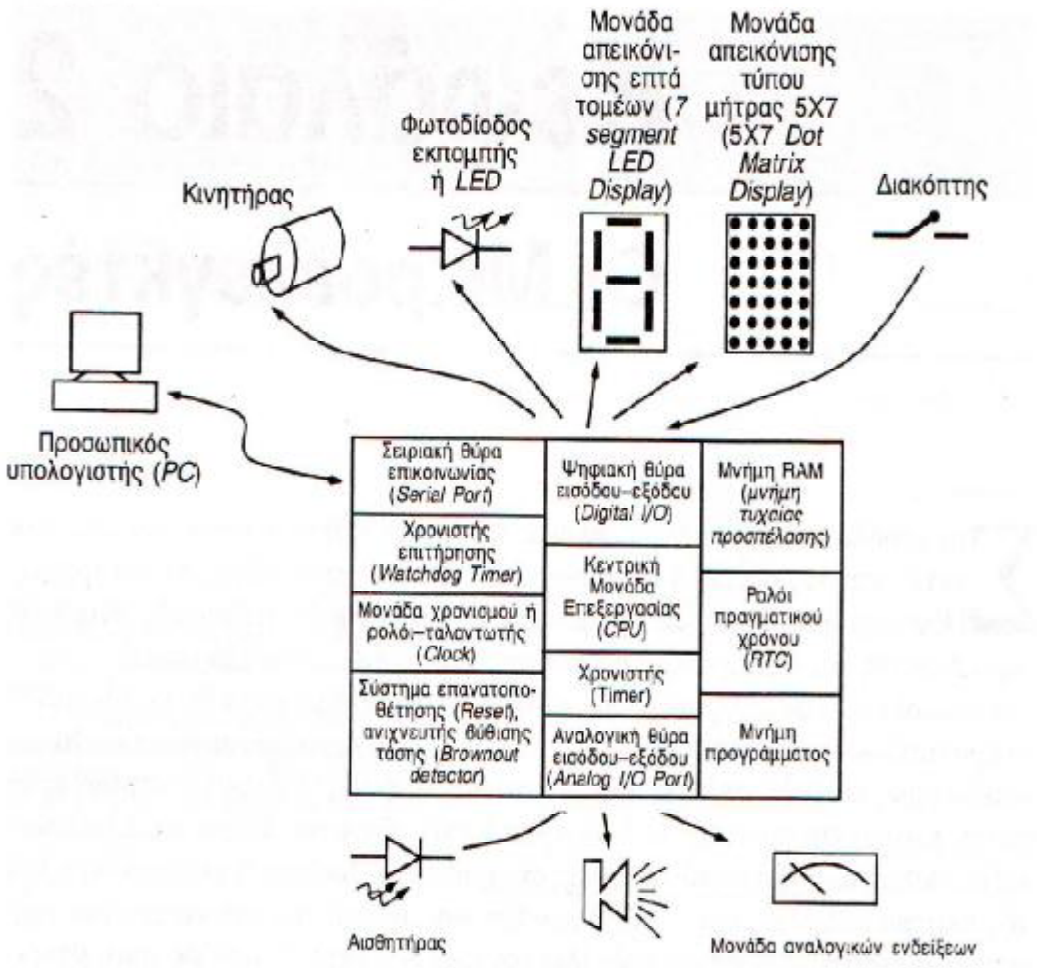
Η μεγάλη μερίδα πωλήσεων των μικροελεγκτών εξακολουθεί να αφορά αυτούς των 8 bit, καθώς είναι η κατηγορία με το χαμηλότερο κόστος και το μικρότερο μέγεθος λογισμικού για το ίδιο αποτέλεσμα, ιδίως επειδή οι σύγχρονες οικογένειες οκτάμπιτων μικροελεγκτών έχουν πολύ βελτιωμένες επιδόσεις σε σχέση με το παρελθόν.

Η επιτυχία μιας οικογένειας μικροελεγκτών καθορίζεται σε μεγάλο βαθμό από τη διαθεσιμότητα και την ευχρηστία των σχετικών εργαλείων ανάπτυξης, όπως μεταφραστές από γλώσσες υψηλού επιπέδου σε γλώσσα κατανοητή από τον μικροελεγκτή (assembly), προγραμματιστές της εσωτερικής μνήμης και εργαλεία εκσφαλμάτωσης (debuggers). Στους μικροελεγκτές, τα εργαλεία αυτά δεν αποτελούνται ποτέ μόνο λογισμικό, καθώς δεν υπάρχει

τυποποιημένος τρόπος επικοινωνίας με αυτούς. Στον τομέα των εργαλείων ανάπτυξης, δραστηριοποιούνται όχι μόνο οι ίδιοι οι κατασκευαστές μικροελεγκτών αλλά και εξειδικευμένες εταιρείες.

Η πιο διαδεδομένη γλώσσα προγραμματισμού των μικροελεγκτών είναι η C, η C++ και οι παραλλαγές τους. Σε τμήματα του λογισμικού όπου απαιτείται ταχύτητα η μικρό μέγεθος χρησιμοποιούμενης μνήμης, μπορεί να χρησιμοποιείται η Assembly. Όμως οι μεγαλύτερες απαιτήσεις σε λειτουργικότητα και η ευκολία προγραμματισμού της C έναντι της assembly, σε συνδυασμό με την επάρκεια μνήμης των σύγχρονων μικροελεγκτών, έχουν γενικά εκτοπίσει την Assembly από τις περισσότερες εφαρμογές.

Η εργασία με τους μικροελεγκτές είναι ουσιαστικά διασκέδαση, καθώς από σχεδιαστική άποψη, η χρήση τους είναι αρκετά εύκολη και προσιτή. Στο σχήμα που ακολουθεί εμφανίζονται οι δυνατότητες ενός μικροελεγκτή και συγκεκριμένα ενός τυπικού μικροελεγκτή AVR. Η μονάδα που εμφανίζεται στο κέντρο του σχήματος αντιστοιχεί ουσιαστικά στον ίδιο το μικροελεγκτή. Η μονάδα αυτή μπορεί να συνδεθεί κατάλληλα με διάφορα είδη κινητήρων, με μια ποικιλία διατάξεων απεικόνισης (Display) ως συσκευών εξόδου, να επικοινωνήσει με έναν προσωπικό υπολογιστή (PC), να διαβάσει τιμές από εξωτερικούς αισθητήρες, ενώ ακόμη μπορεί να συνδεθεί και σε ένα είδος τοπικού δικτύου άλλων παρόμοιων μικροελεγκτών και όλες αυτές οι δυνατότητες δεν απαιτούν ιδιαίτερα μεγάλο αριθμό εξωτερικών εξαρτημάτων. Το γεγονός αυτό οδηγεί σε πιο συμπαγή και για το λόγο αυτό πιο αξιόπιστα συστήματα, σε σχετικά χαμηλό κόστος (εξαιτίας του μικρού αριθμού των εξωτερικών εξαρτημάτων και των λιγότερων γενικά συνδέσεων που απαιτούνται για το λόγο αυτό).



Σχήμα 1.5: Απεικόνιση ποικίλων δυνατοτήτων του μικροελεγκτή να συνδεθεί με διάφορες εξωτερικές ως προς αυτόν διατάξεις, με τον ελάχιστο απαιτούμενο αριθμό πλέον εξαρτημάτων.

Στο σημείο αυτό θα παρουσιάσουμε συνοπτικά τα διάφορα συστατικά ενός μικροελεγκτή:

Ä Μονάδα Κεντρικής Επεξεργασίας – CPU (Central Processing Unit): αποτελεί την «καρδιά» ενός μικροελεγκτή αφού εκτελεί την ανάκληση δεδομένων (fetch) από τη μνήμη προγράμματος υπό μορφή εντολών, ενώ παράλληλα αποκωδικοποιεί τις εντολές και στη συνέχεια τις εκτελεί. Η μονάδα CPU, αποτελείται από καταχωρητές (registers), την αριθμητική λογική μονάδα (ALU – Arithmetic Logic Unit), τον αποκωδικοποιητή εντολών (Instruction Decoder) κι διάφορα κυκλώματα ελέγχου.

Ä Μνήμη Προγράμματος: σε αυτήν αποθηκεύονται οι εντολές που σχηματίζουν τον κορμό του προγράμματος. Διακρίνεται στην εσωτερική και την εξωτερική μνήμη προγράμματος.

Ä Μνήμη RAM: η μνήμη τυχαίας προσπέλασης αποτελεί τη μνήμη δεδομένων του ελεγκτή, γεγονός το οποίο σημαίνει ότι χρησιμοποιείται από τον ελεγκτή για την αποθήκευση

δεδομένων. Η CPU, χρησιμοποιεί τη μνήμη RAM για την αποθήκευση μεταβλητών, καθώς επίσης και τη λεγόμενη Στοίβα (Stack). Η Στοίβα με τη σειρά της χρησιμοποιείται από τη CPU, για την προσωρινή αποθήκευση των λεγόμενων διευθύνσεων επιστροφής, με σκοπό να συνεχίσει την εκτέλεση ενός προγράμματος το οποίο έχει διακοπεί.

Ä Ταλαντωτής Χρονισμού: ένα πρόγραμμα εκτελείται από το μικροελεγκτή με έναν καθορισμένο ρυθμό. Ο ρυθμός αυτός καθορίζεται από τη συχνότητα λειτουργίας του ταλαντωτή χρονισμού, ο οποίος μπορεί να είναι ένας εσωτερικός ταλαντωτής τύπου RC ή ένας ταλαντωτής που υλοποιείται με κάποιο εξωτερικό στοιχείο χρονισμού, όπως για παράδειγμα ένας κρύσταλλος χαλαζία, ένα κύκλωμα συντονισμού LC ή ακόμη και ένα απλό κύκλωμα RC. Η λειτουργία του ταλαντωτή ξεκινά αμέσως μετά την εφαρμογή της τάσης τροφοδοσίας.

Ä Σύστημα επανατοποθέτησης και Κύκλωμα ανίχνευσης βυθίσεων τάσης: το σύστημα επανατοποθέτησης ή μηδενισμού ή απλά Reset, εξασφαλίζει το γεγονός ότι όλες οι εσωτερικές μονάδες και τα κυκλώματα ελέγχου του μικροελεγκτή θα ξεκινήσουν να λειτουργούν κατά την εφαρμογή της τροφοδοσίας, από κάποια προκαθορισμένη αρχική κατάσταση, ενώ όλοι οι καταχωρητές του συστήματος βρίσκονται σε κατάλληλες αρχικές τιμές. Το Κύκλωμα ανίχνευσης βύθισης της τάσης τροφοδοσίας είναι ένα επίσης εσωτερικό κύκλωμα ελέγχου το οποίο παρακολουθεί συνεχώς το επίπεδο της τάσης τροφοδοσίας και εφόσον ανιχνευτεί κάποια στιγμιαία βύθιση στην τάση αυτή τότε αυτόματα θέτει τον μικροελεγκτή σε λειτουργία επανατοποθέτησης, έτσι ώστε να προστατευθούν τα πειρεχομένα των καταχωρητών και της μνήμης από πιθανή καταστροφή ή αλλοίωση, πράγμα που θα οδηγούσε τον μικροελεγκτή σε εσφαλμένη λειτουργία.

Ä Σειριακή Θύρα Επικοινωνίας: εξυπηρετεί διαμεσολαβητικό σκοπό καθώς η λειτουργία της βασίζεται στο ότι λαμβάνει δεδομένα από το μικροελεγκτή τα οποία ολισθαίνει προς την έξοδο υπό μορφή ενός δυαδικού στοιχείου bit τη φορά. Παρομοίως, λαμβάνει δεδομένα από την αντίστοιχη είσοδό της και πάλι με τη μορφή ενός bit τη φορά, σχηματίζοντας έτσι με 8 τέτοια bits μια λέξη του 1 byte, την οποία και αντιγράφει στο εσωτερικό του ελεγκτή. Μπορεί να λειτουργήσει σε οποιαδήποτε ταχύτητα μετάδοσης δεδομένων τυχόν απαιτηθεί.

Ä Ψηφιακή Θύρα Εισόδου – Εξόδου: χρησιμοποιούνται για την ανταλλαγή δεδομένων από και προς το εξωτερικό περιβάλλον. Τα δεδομένα ανταλλάσσονται υπό τη μορφή ομάδων των 8 bits ή του 1 byte.

Ä Αναλογική θύρα εισόδου – εξόδου: η ύπαρξη αναλογικών εισόδων προϋποθέτει Μετατροπείς Αναλογικού Σήματος σε Ψηφιακό. Ένας μικροελεγκτής είναι δυνατόν να διαθέτει μία ενσωματωμένη μονάδα μετατροπής, η οποία χρησιμοποιείται για την ανάγνωση δεδομένων από αισθητήρες, όπως για παράδειγμα αισθητήρες πίεσης και θερμοκρασίας. Παράλληλα,

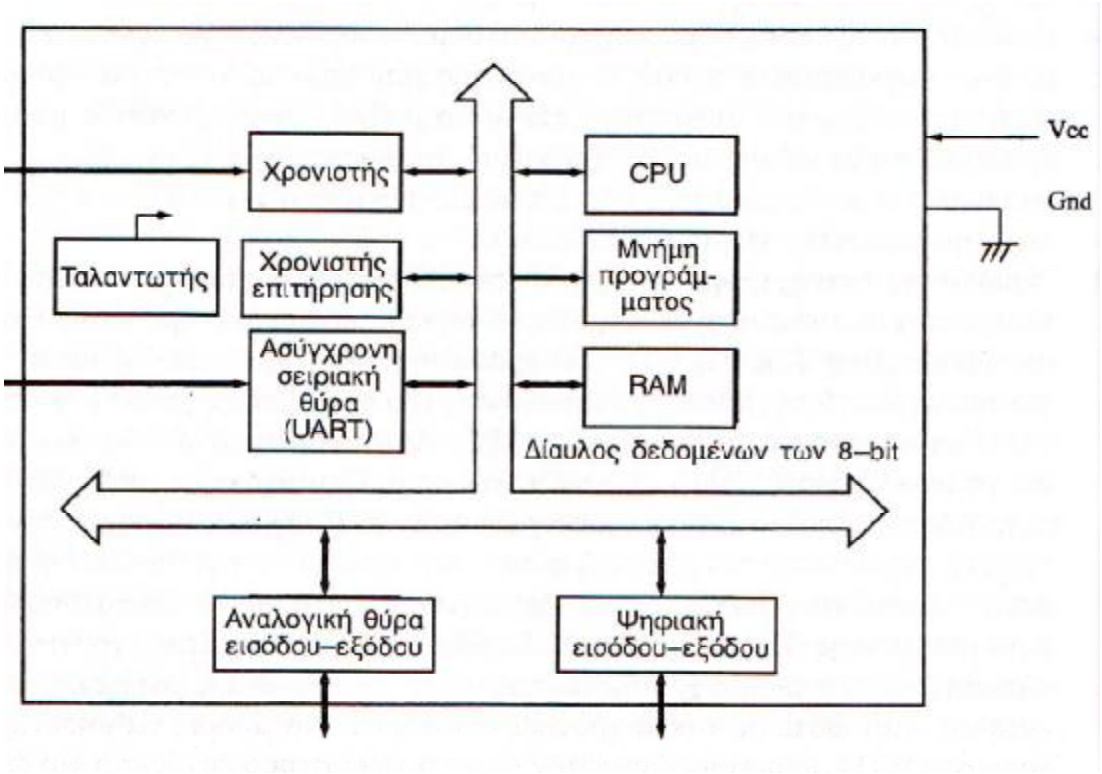
μπορούμε να έχουμε και αναλογικές εξόδους, χρησιμοποιώντας κάποιες μονάδες οι οποίες καλούνται Μετατροπείς Ψηφιακού Σήματος σε Αναλογικό. Οι μετατροπείς αυτοί χρησιμοποιούνται για την οδήγηση κινητήρων, ειδικών μονάδων απεικόνισης, για την αναπαραγωγή σημάτων ήχου ή μουσικής και αλλού.

Ä Χρονιστής (Timer) χρησιμοποιείται για το χρονισμό ή τη σηματοδότηση διαφόρων γεγονότων, όπως για παράδειγμα για να παράγει το ζητούμενο ρυθμό που απαιτείται προκειμένου να καταφέρουμε να στείλουμε επιτυχώς δεδομένα σε μια εξωτερική οθόνη. Μπορεί να χρησιμοποιηθεί επίσης και για την καταμέτρηση γεγονότων, τα οποία μπορούν να είναι είτε εσωτερικά είτε εξωτερικά.

Ä Χρονιστής Επιτήρησης (Watchdog Timer): χρησιμοποιείται προκειμένου να αποφευχθεί μια πιθανή κατάρρευση του συστήματος (crash), οπότε και ενεργοποιείται η λειτουργία ενός αυξανόμενα εσωτερικού μετρητή σε συγκεκριμένο ρυθμό. Αν το πρόγραμμα χρήσης δε μηδενίσει (ή επαναθέσει το μετρητή αυτό) τότε κάποια στιγμή θα επέλθει η λεγόμενη υπερχείλιση του παραπάνω μετρητή και θα επανατοποθετηθεί ο ελεγκτής σε λειτουργία reset. Η λογικής αυτής της τεχνικής ελέγχου στηρίζεται στην υπόθεση ότι αν το πρόγραμμα χρήσης δε μηδενίσει το χρονιστή επιτήρησης αυτό πιθανότατα σημαίνει ότι το πρόγραμμα έχει αποτύχει σε κάποια προσπάθειά του είτε εξαιτίας πιθανής κατάρρευσης ή γενικότερα κάποιας απρόβλεπτης συμπεριφοράς οπότε είναι προτιμότερο να εκκινήσει διαδικασία επανατοποθέτησης.

Ä Ρολόι Πραγματικού Χρόνου: ο σκοπός του είναι η μέτρηση και η διατήρηση της τρέχουσας ώρας της ημέρας, της ημερομηνίας κ.λπ. Μπορεί να χρησιμοποιηθεί για τη σηματοδότηση συγκεκριμένων γεγονότων με γνώμονα την τρέχουσα ώρα.

Μια τυπική μορφή μικροελεγκτή είναι αυτή που παρουσιάζεται στο σχήμα που ακολουθεί. Από τις διάφορες κατηγορίες μικροελεγκτών, εκείνη που γενικά έχει τη μεγαλύτερη απήχηση στη διεθνή αγορά είναι η κατηγορία των 8 bit, τόσο εξαιτίας του χαμηλού τους κόστους, όσο και λόγω της διαθεσιμότητας των διατάξεων αυτών σε μια μεγάλη ποικιλία, από την άποψη των επιδόσεων και των διαθέσιμων ολοκληρωμένων περιφερειακών μονάδων



Σχήμα 1.6: Ένας μικροελεγκτής των 8 bit.

1.8 Κατασκευαστές Μικροελεγκτών

Μερικοί από τους γνωστότερους κατασκευαστές μικροελεγκτών είναι οι εξής:

- ARM (δεν κατασκευάζει αλλά παραχωρεί δικαιώματα χρήσης του πυρήνα της)
- Atmel
- Epson
- Freescale Semiconductor (πρώην Motorola)
- Hitachi
- Maxim (μετά την εξαγορά της Dallas)
- Microchip
- NEC
- Toshiba
- Texas Instruments

1.9 Σύγκριση Μικροεπεξεργαστή – Μικροελεγκτή

Αναμφίβολα, τόσο οι μικροεπεξεργαστές όσο και οι Μικροελεγκτές, αποτελούν απαραίτητο και αναπόσπαστο κομμάτι των σύγχρονων τεχνολογικών εφαρμογών. Εντούτοις, οι δύο αυτές συσκευές διαφέρουν μεταξύ τους σε αρκετά σημεία. Μία ουσιώδης διαφορά έγκειται στη λειτουργικότητά τους. Για να λειτουργήσει ένας μικροεπεξεργαστής, θα πρέπει να συνδεθεί με άλλες συσκευές, όπως η μνήμη (memory) ή μια συσκευή αποστολής και λήψης δεδομένων. Συνεπώς, αποτελεί την καρδιά του συστήματος.

Από την άλλη πλευρά, ο μικροελεγκτής είναι σχεδιασμένος έτσι που να εμπεριέχει όλες τις παραπάνω συσκευές και να μην απαιτείται η παρουσία άλλων για τη λειτουργία του, αφού όλα τα απαραίτητα περιφερειακά (peripherals) είναι εξαρχής ενσωματωμένα στη δομή του. Επομένως, με την παρουσία του μικροελεγκτή, εξοικονομείται χώρος και χρόνος κατά την κατασκευή ενός συστήματος που βασίζεται σε αυτό.

Επιπρόσθετα, μία ακόμη διάφορα έγκειται στην υπολογιστική ισχύ την οποία διαθέτουν. Στους σύγχρονους μικροεπεξεργαστές για μη ενσωματωμένα συστήματα (πχ τους μικροεπεξεργαστές των προσωπικών υπολογιστών), δίνεται έμφαση στην υπολογιστική ισχύ. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία διασυνδέονται με την κεντρική μονάδα (μικροεπεξεργαστή), η οποία δεν είναι εξειδικευμένη. Αντίθετα, στους μικροεπεξεργαστές για ενσωματωμένα συστήματα (μικροελεγκτές), οι οποίοι έχουν μικρότερες ή και μηδαμινές δυνατότητες συνεργασίας με εξωτερικά περιφερειακά, αυτού του είδους η ευελιξία είναι περιορισμένη, καθώς και η υπολογιστική ισχύς. Οι μικροελεγκτές δίνουν έμφαση στο μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση.

Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών, αν και στους πρώτους είναι απαντάται συχνά η αρχιτεκτονική μνήμης τύπου Harvard, η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης προγράμματος και της μνήμης δεδομένων (πχ οι σειρές AVR από την Atmel και PIC από την Microchip). Στους κοινούς μικροεπεξεργαστές συνηθίζεται η ενιαία διάταξη μνήμης τύπου von-Neumann.

Μία ακόμη διαφορά σχετίζεται με τα συνήθη υποσυστήματα, τα οποία πλαισιώνουν έναν μικροεπεξεργαστή και έναν μικροελεγκτή. Στον μικροεπεξεργαστή, το ολοκληρωμένο κύκλωμα που τον αποτελεί περιέχει μόνο την *Λογική και Αριθμητική Μονάδα* (ALU), στοιχειώδεις καταχωρητές (registers), προσωρινή μνήμη RAM πολύ υψηλής ταχύτητας (cache memory) και, κάποιες φορές, τον ελεγκτή μνήμης (memory controller). Όμως, για τη λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος, απαιτούνται πολλά εξωτερικά υποσυστήματα και περιφερειακά. Τέτοια είναι:

- Κύκλωμα συνδετικής λογικής (glue logic) για τη σύνδεση των εξωτερικών μνημών και άλλων περιφερειακών παράλληλης σύνδεσης στην αρτηρία δεδομένων (bus) του επεξεργαστή.
- Μνήμη προγράμματος (τύπου ROM, FLASH, EPROM κλπ) η οποία περιέχει το λογισμικό του συστήματος. Σε κάποια μοντέλα, είναι δυνατό το κλείδωμα αυτής της μνήμης, μετά την εγγραφή της, ώστε να προστατευτεί το περιεχόμενό της από αντιγραφή.
- Μεγάλη ποσότητα μνήμης RAM.
- Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM) η οποία να μπορεί να γράφεται τον πυρήνα του μικροελεγκτή. Αυτή η μνήμη έχει, έναντι της FLASH, το πλεονέκτημα της δυνατότητας διαγραφής και εγγραφής οποιουδήποτε μεμονωμένου byte.
- Κύκλωμα αρχικοποίησης (reset).
- Διαχειριστή αιτήσεων διακοπής (interrupt request controller) από τα περιφερειακά.
- Κύκλωμα επιτήρησης τροφοδοσίας (brown-out detection) το οποία παρακολουθεί την τροφοδοσία και αρχικοποιεί ολόκληρο το σύστημα όταν αυτή πέσει κάτω από τα ανεκτά όρια, προλαμβάνοντας έτσι την αλλοίωση των δεδομένων.
- Κύκλωμα επιτήρησης λειτουργίας (watchdog timer) το οποίο αρχικοποιεί το σύστημα, αν αυτό εμφανίσει σημάδια δυσλειτουργίας λόγω κολλήματος (hang).
- Τοπικό ταλαντωτή για την παροχή παλμών χρονισμού (clock).
- Έναν ή περισσότερους χρονιστές-απαριθμητές υψηλής ταχύτητας (hardware timer-counter) για τη δημιουργία καθυστερήσεων, μέτρηση διάρκειας γεγονότων, απαρίθμηση γεγονότων και άλλων λειτουργιών ακριβούς χρονισμού.
- Ρολόι πραγματικού χρόνου (Real Time Clock, RTC) το οποίο τροφοδοτείται από ανεξάρτητη μπαταρία και γι αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
- Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input-Output , PIO).

Γενικά, όλες οι οικογένειες μικροελεγκτών ενσωματώνουν τα περισσότερα από τα παραπάνω περιφερειακά και έτσι δεν απαιτούνται εξωτερικά υποσυστήματα. Οι διαφοροποιήσεις εντοπίζονται κυρίως στην ύπαρξη ή μη εσωτερικής μνήμης προγράμματος και στο είδος της. Έτσι, υπάρχουν:

- Μικροελεγκτές χωρίς μνήμη προγράμματος, οι οποίοι χαρακτηρίζονται ως ***ROM-less***. Αυτοί παρέχουν πάντοτε μια παράλληλη αρτηρία (bus) δεδομένων, πάνω στην οποία συνδέονται εξωτερικές μνήμες προγράμματος και RAM. Τέτοιοι τύποι μικροελεγκτών προορίζονται για πιο ισχυρά υπολογιστικά συστήματα ελέγχου, με μεγαλύτερες απαιτήσεις μνήμης.
- Μικροελεγκτές με μνήμη ROM, η οποία κατασκευάζεται με το λογισμικό της (Mask ROM) ή γράφεται μόνο μια φορά (One Time Programmable, OTP). Παρέχουν τη δυνατότητα πολύ χαμηλού κόστους, όταν αγοράζονται σε πολύ μεγάλες ποσότητες.
- Μικροελεγκτές με μνήμη FLASH, οι οποία μπορούν συνήθως να προγραμματιστεί πολλές φορές. Αυτή είναι η πιο διαδεδομένη κατηγορία. Συχνά ο προγραμματισμός της μνήμης μπορεί να γίνει ακόμη και πάνω στο κύκλωμα της ίδιας της ενσωματωμένης (embedded) εφαρμογής (δυνατότητα In Circuit Programming, ISP). Αυτοί οι μικροελεγκτές έχουν ουσιαστικά αντικαταστήσει τους παλαιότερους τύπους EPROM που έσβηναν με υπεριώδη ακτινοβολία (από το ειδικό τζαμάκι).

ΚΕΦΑΛΑΙΟ 2 – ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΤΟΥ AVR

2.1 Εισαγωγή

Ιστορικά ο όρος AVR αναφέρεται στην αρχιτεκτονική που αναπτύχτηκε από τους Alf-Egil Bogen και Vegard Wollan ως φοιτητές του Norwegian Institute of Technology.Η αρχιτεκτονική αυτή (η όποια βασίστηκε στην προγενέστερη αρχιτεκτονική Harvard Architecture και στην στρατηγική σχεδίασης CPU Reduced Instruction Set Computing) αφού αγοραστικέ από την εταιρία ATMEL.

Κύριοι εκπρόσωποι τον οικογενειών είναι οι tinyAVR,megaAVR,90SAVR κ.α. ενώ στα πλαίσια κάθε μιας αναπτύχτηκαν μια σειρά από συγκριμένα μοντέλα.Οι κυριότερες διαφορές εστιάζονται σε θέματα όπως διαθέσιμη μνήμη προγράμματος είδος συσκευασίας το οποίο επηρεάζει το πλήθος τον διαθέσιμων I/O ακροδεκτών και τις δυνατότητες επικοινωνίας με άλλες περιφερικές συσκευές (η οποία επικοινωνία είναι από τις κύριες αποστολές ενός μικροελεγκτή).Έτσι λοιπόν η οικογένεια tinyAVR προσανατολίζεται σε εφαρμογές μικρής κλίμακας με μοντέλα που παρέχουν περιορισμένη μνήμη προγράμματος ,συσκευασίες με λίγους ακροδέκτες και κατά συνέπεια περιορισμένες δυνατότητες διασύνδεσης με περιφερικά συστήματα .από την άλλη η οικογένεια megaAVR παρέχει μοντέλα με δυνατότητα διαχείρισης μεγάλων προγραμμάτων (μέχρι 256 kb),συσκευασίας με μέχρι και 100 ακροδέκτες και κατά συνέπεια εκτεταμένες δυνατότητες διασύνδεσης.

Σημαντική παρατήρηση είναι ότι όλοι οι AVR μικροελεγκτές μοιράζομαι το ίδιο ρεπερτόριο εντολών ,οργάνωση μνήμης και γενικά αρχιτεκτονικά χαρακτηρίστηκα και κατά συνέπεια η μετάβαση από την μια οικογένεια στην άλλη είναι ιδιαίτερα εύκολη και χωρίς μεγάλες απαιτήσεις από τον προγραμματιστή.

2.2 ΚΑΤΗΓΟΡΙΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ

Όπως είδαμε στο προηγούμενο κεφάλαιο, κάθε μιροϋπολογιστικό σύστημα διακρίνεται για τα ιδιαίτερα χαρακτηριστικά και τις ιδιότητες τις οποίες παρουσιάζει και για το λόγο αυτό οι αρχιτεκτονικές των μικροελεγκτών ποικίλουν. Διακρίνουμε τις εξής κατηγορίες αρχιτεκτονικών σχημάτων:

1. Αυτή που λαμβάνει υπόψη το συνολικό αριθμό των εντολών. Έτσι έχουμε την αρχιτεκτονική CISC (Complex Instruction Set Computer ή Αρχιτεκτονική Σύνθετου

Ρεπερτορίου Εντολών), την αρχιτεκτονική RISC (Reduced Instruction Set Computer ή Αρχιτεκτονική Μειωμένου Ρεπερτορίου Εντολών) και την αρχιτεκτονική MISC (Minimum Instruction Set Computer ή Αρχιτεκτονική Ελαχίστου Ρεπερτορίου Εντολών). Ωστόσο, οι παραπάνω όροι έχουν υποστεί πολλές παραφράσεις από τους διάφορους ειδικούς προγραμματιστές και πωλητές του είδους, ενώ συχνά προκαλείται σύγχυση καθώς συμβαίνει δύο μικροελεγκτές να παρουσιάζουν πολλές κοινές ιδιότητες (για παράδειγμα ο CISC και ο RISC).

2. Αυτή που αφορά στον τρόπο με τον οποίο πραγματοποιείται η πρόσβαση στη μνήμη προγράμματος και τη μνήμη δεδομένων. Ένα τέτοιο μοναδικό μοντέλο μνήμης, το οποίο είναι γνωστό με την ονομασία Αρχιτεκτονική Princeton ή Αρχιτεκτονική Von Neumann και το οποίο σε αντίθεση με την Αρχιτεκτονική Harvard, προβλέπει ξεχωριστό χώρο μνήμης για την αποθήκευση προγράμματος και για την αποθήκευση δεδομένων αντίστοιχα.

3. Με βάση τον τρόπο αποθήκευσης και διαχείρισης που υφίστανται τα δεδομένα εντός της CPU. Αντικειμενικός σκοπός ενός συστήματος μικροελεγκτή είναι η διαχείριση δεδομένων. Την εργασία αυτή την επιτυγχάνει με τη βοήθεια του προγράμματος χρήσης. Ο τρόπος λοιπόν διαχείρισης και αποθήκευσης δεδομένων εντός της CPU, καθώς επίσης και οι διάφοροι τρόποι προσπέλασης, διαμορφώνουν μία βάση με την οποία ταξινομούνται οι διάφορες αρχιτεκτονικές των επεξεργαστών και κατά συνέπεια ένα ακόμη διαφορετικό σχήμα ταξινόμησης.

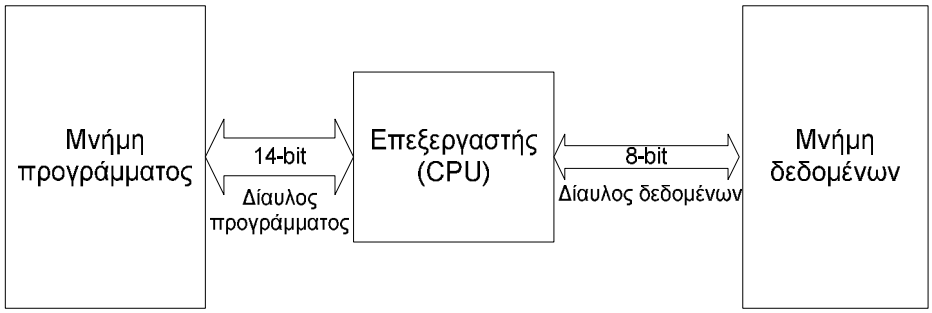
Σύμφωνα με τα παραπάνω διακρίνουμε τέσσερα βασικά πρότυπα ή μοντέλα:

1) Το μοντέλο Στοίβας (STACK),
2) Το μοντέλο Συσσωρευτή (ACCUMULATOR),
3) Το μοντέλο Καταχωρητής – Μνήμη (Register-Memory) και
4) Το μοντέλο πολλών Καταχωρητών (Register-Register) ή μοντέλο Φόρτωσης – Αποθήκευσης (Load-Store).

2.3 ΟΙ ΜΙΚΡΟΕΛΕΓΚΤΕΣ AVR

Από τα γενικά χαρακτηριστικά της αρχιτεκτονικής των μικροελεγκτών θα προχωρήσουμε στην ενότητα αυτή στην παρουσίαση της οικογένειας των μικροεπεξεργαστών AVR στο πεδίο της αρχιτεκτονικής τους. Στο επόμενο κεφάλαιο θα αναφερθούμε στις περιφερειακές μονάδες

που πλαισιώνουν το σύστημα. Με βάση την αρχιτεκτονική Harvard, ο μικροελεγκτής AVR περιλαμβάνει έναν επεξεργαστή RISC (Reduced Instruction Set Computer) και η μονάδα CPU συνεργάζεται ταυτόχρονα με μία μνήμη προγράμματος (program memory) και μία ξεχωριστή μνήμη δεδομένων (data memory), γεγονός το οποίο καθιστά την αρχιτεκτονική Harvard αποδοτική αφού καθίσταται δυνατόν να εκτελείται κάποια εντολή και παράλληλα να εγγράφεται ή να διαβάζεται η μνήμη. Με αυτόν τον τρόπο επιτυγχάνεται η εκτέλεση της εντολής σε ένα μόνο χρόνο μηχανής. Όπως φαίνεται και στο σχήμα που ακολουθεί, υπάρχει διαφορετικός δίαυλος για τη μεταφορά δεδομένων (data bus) και διαφορετικός για τη μεταφορά εντολών (instruction bus).



Σχήμα 2.1: Σύνδεση του επεξεργαστή με τη μνήμη προγράμματος και δεδομένων.

Οι μικροελεγκτές AVR της εταιρίας ATMEL παρουσιάζουν τα ακόλουθα χαρακτηριστικά:

Ä Συνδυάζουν την αρχιτεκτονική RISC με ως επί τω πλείστον σταθερού μήκους εντολές, διαδικασίες αποθήκευσης – φόρτωσης στη μνήμη και 32 καταχωρητές γενικής χρήσης.

Ä Διαθέτουν μηχανισμό συνεχούς διοχέτευσης εντολών (Instruction pipeline) σε δύο στάδια, που επιταχύνει σημαντικά τη διαδικασία εκτέλεσης.

Ä Οι περισσότερες από τις εντολές που περιλαμβάνει το ρεπερτόριό τους εκτελούνται στη διάρκεια μιας περιόδου του κεντρικού ρολογιού.

Ä Λειτουργούν σε συχνότητες χρονισμού έως 10 MHz.

Ä Διαθέτουν μεγάλη ποικιλία σε ότι αφορά ενσωματωμένες περιφερειακές μονάδες όπως, ψηφιακές εισόδους – εξόδους (I/O), μετατροπείς αναλογικού σήματος σε ψηφιακό ή ADC, μνήμη τύπου EEPROM, χρονιστές, μονάδες ασύγχρονης σειριακής επικοινωνίας ή UART (Universal Asynchronous Receiver Transmitter), ρολόγια πραγματικού χρόνου (RTC – Real Time Clock), μονάδες διαμόρφωσης εύρους παλμών (PWM – Pulse Width Modulation) και άλλα.

Ä Ενσωματωμένες μνήμες προγράμματος και δεδομένων.

Ä Δυνατότητα προγραμματισμού εντός του συστήματος (ISP – In-System Programmable).

Ä Διατίθενται σε συσκευασίες των 8 έως 64 ακροδεκτών οπότε κρίνονται κατάλληλοι για έναν μεγάλο αριθμό διαφορετικών εφαρμογών.

Ä Είναι περίπου 12 φορές ταχύτεροι και πιο αποδοτικοί σε σχέση με τους ελεγκτές κλασικής αρχιτεκτονικής CISC (Complex Instruction Set Computer).

Ä Ευρεία περιοχή τάσεων λειτουργίας από 2.7 V έως 6.0V.

Ä Η σχετικά απλή αρχιτεκτονική τους δίνει το πλεονέκτημα του μικρού σε απαίτηση κύκλου εκμάθησης στους αρχάριους.

Η οικογένεια των μικροελεγκτών AVR διαθέτει τα ακόλουθα:

1. Μία ενσωματωμένη μνήμη flash ή αλλιώς ταχείας αποθήκευσης, η οποία έχει τη δυνατότητα προγραμματισμού εντός του συστήματος (ISP, In System Programmable), ως μνήμη προγράμματος. Η δυνατότητα αυτή καθιστά περιττή την ύπαρξη εξωτερικών μνημών τύπου ROM ή EPROM οι οποίες να περιέχουν τον κώδικα του προγράμματος χρήσης.
2. 32 καταχωρητές εργασίας των 8 bit. Αυτό έχει ως αποτέλεσμα οι διάφορες μεταβλητές να μπορούν να αποθηκεύονται εντός της CPU και όχι σε κάποια μνήμη, όπου η διαδικασία πρόσβασης θεωρείται χρονοβόρα.
3. Μία ενσωματωμένη μνήμη δεδομένων (data memory), τύπου EEPROM και RAM στις περισσότερες διατάξεις της σειράς τους. Σε αυτές τις μνήμες αποθηκεύονται σταθερές τιμές και μεταβλητές.
4. Η λειτουργία τους γίνεται σε συχνότητες χρονισμού από 0 έως 10 MHz. οι περισσότερες εντολές της γλώσσας των μικροελεγκτών AVR, σε μία μόνο περίοδο του κεντρικού σήματος χρονισμού, γεγονός που οδηγεί σε μία βελτιωμένη κατά 10 φορές περίπου επίδοση σε σχέση με τους κλασικούς μικροελεγκτές που λειτουργούν στην ίδια συχνότητα χρονισμού.
5. Εσωτερικό κύκλωμα που επανατοποθετεί το σύστημα κατά την εφαρμογή της τάσης τροφοδοσίας (Power On Reset – POR).
6. Ενσωματωμένο προγραμματιζόμενο χρονιστή με μονάδα διαίρεσης συχνότητας (prescaler). Η μονάδα αυτή χρησιμοποιείται στις περιπτώσεις αυτές όπου απαιτούνται εφαρμογές κρίσιμου χρονισμού.
7. Περιλαμβάνει πηγές εσωτερικών και εξωτερικών διακοπών.
8. Προγραμματιζόμενο χρονιστή επιτήρησης (Watch Dog Timer - WDT). Ο ρόλος του είναι η αποφυγή εσφαλμένων λειτουργιών σε περίπτωση κάποιας πιθανής κατάρρευσης του προγράμματος χρήσης.

9. Λειτουργίες αποκοπής (Power Down) και ηρεμίας (Sleep). Όταν δεν υπάρχει δραστηριότητα ο μικροεπεξεργαστής τίθεται σε αυτές τις καταστάσεις προκειμένου να καταναλώνεται λιγότερη ισχύς.

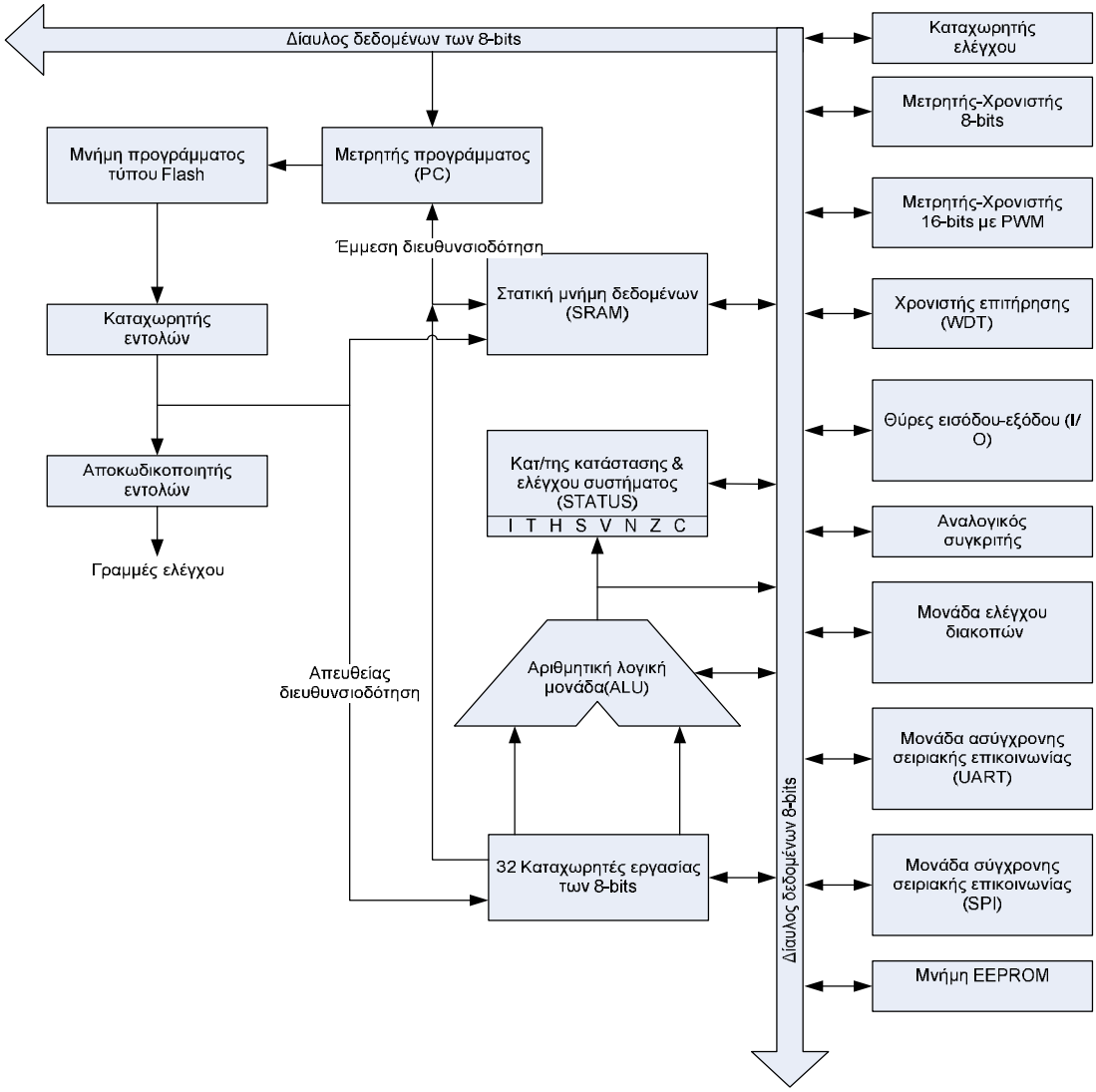
10. Διάφορα μοντέλα της σειράς AVR διαθέτουν ενσωματωμένο ταλαντωτή τύπου RC, ο οποίος όταν ενεργοποιείται συμβάλει σημαντικά στη μείωση του συνολικού αριθμού των απαιτούμενων εξωτερικών εξαρτημάτων.

11. Τέλος, οι μικροελεγκτές αυτής της σειράς ποικίλουν (από τους 8 στους 68 ακροδέκτες), οπότε καθένας μπορεί να επιλέξει εκείνον που θεωρεί ως τον καταλληλότερο για την εργασία του, ενώ παρέχεται και μία πληθώρα περιφερειακών (UART, SPI, αναλογικός συγκριτής, μετατροπέας adc).

Στο σχήμα που ακολουθεί εμφανίζονται διάφορα μοντέλα μικροελεγκτών και τα βασικά χαρακτηριστικά τους.

Μοντέλο	Ακροδέκτες	Flash	EEPROM	RAM	UART	ADC
90S1200	20	1K	64 bytes	0	Όχι	Όχι
90S2313	20	2K	128	128	Ναι	Όχι
90S2323	8	2K	128	128	Όχι	Όχι
90S2333	28	2K	128	128	Ναι	Ναι
90S4433	28	4K	256	128	Ναι	Ναι
90S4414	40	4K	256	256	Ναι	Όχι
90S8515	40	8K	512	512	Ναι	Όχι
90S4434	40	4K	256	256	Ναι	Ναι
90S2343	8	2K	128	128	Όχι	Όχι
Mega103	64	128K	4096	4096	Ναι	Ναι
Mega603	64	64K	2048	4096	Ναι	Ναι
Tiny10	8	1K	64	0	Όχι	Όχι
Tiny12	8	1K	64	0	Όχι	Όχι
Tiny13	8	2K	128	128	Όχι	Όχι

Επιπλέον, στο ακόλουθο σχήμα μπορεί να δει κανείς την αρχιτεκτονική διάταξη των μικροεπεξεργαστών AVR και των λειτουργικών μονάδων τους. Το σημαντικό σε όλους αυτούς του τύπους μικροελεγκτών είναι ότι ο πυρήνας της αρχιτεκτονικής τους είναι ίδιος και έχουν το ίδιο σύνολο εντολών, διαφέρουν όμως ως προς τα περιφερειακά τους και το μέγεθος της μνήμης που παρέχουν. Όπως είναι σύνηθες σε αυτού του τύπου τους ελεγκτές, υπάρχουν διαφορετικοί εσωτερικοί δίαυλοι για τη μνήμη προγράμματος και τη μνήμη δεδομένων.

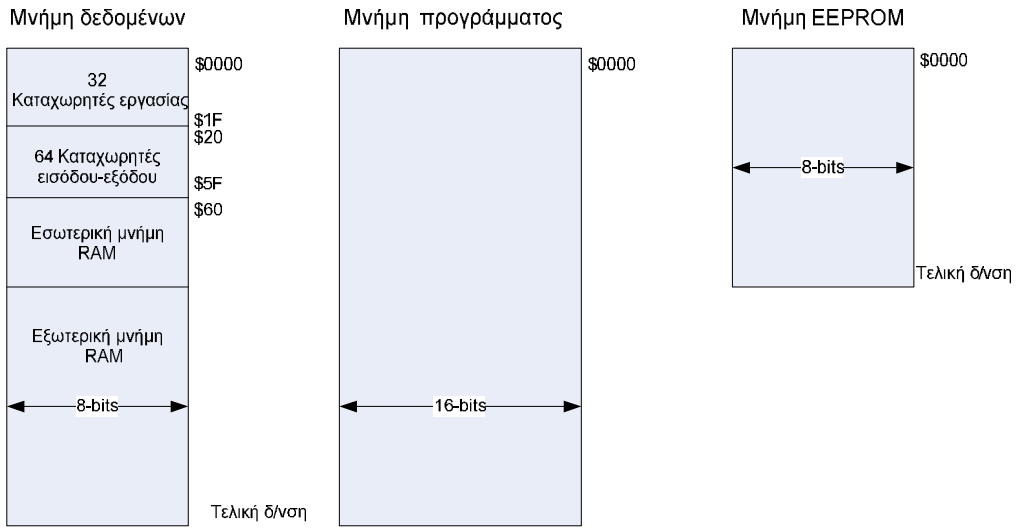


Σχήμα 2.1: Αρχιτεκτονική διάταξη μικροεπεξεργαστών AVR και περιφερειακών μονάδων τους.

2.4 Μνήμη δεδομένων και μνήμη προγράμματος

Ένας μικροελεγκτής διαθέτει τα εξής είδη μνήμης:

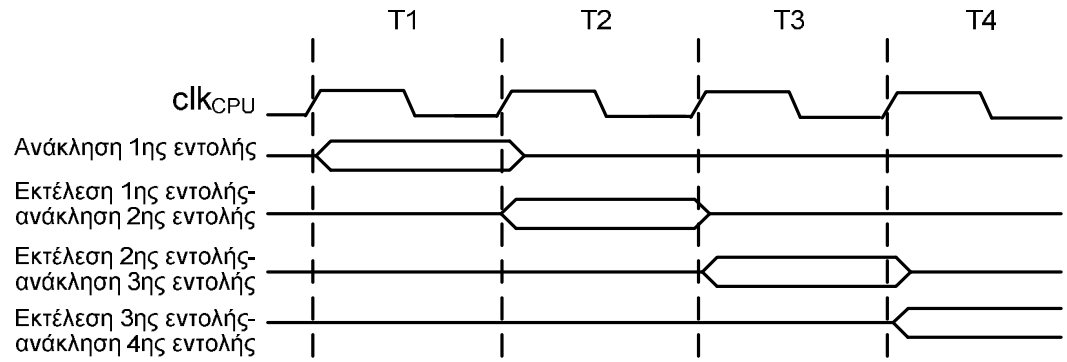
- μνήμη δεδομένων
- μνήμη προγράμματος τύπου flash
- μνήμη EEPROM με αρχική διεύθυνση $0000 και χωρητικότητα από 64bytes ως 4Kbytes.



Σχήμα 2.2 : Τύποι μνημών μικροελεγκτών AVR.

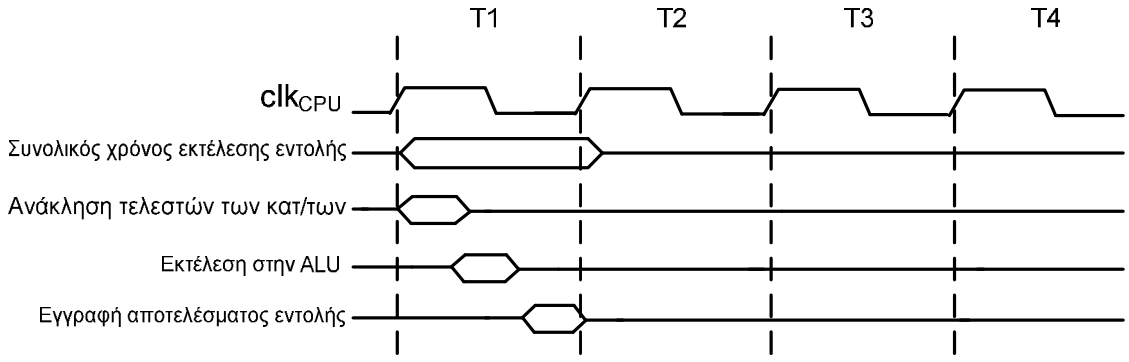
Η μνήμη προγράμματος αποτελεί έναν συνεχή χώρο μιας μνήμης τύπου flash, επιτυγχάνει ταχεία αποθήκευση-φόρτωση δεδομένων και συνδέεται με δίαυλο των 16-bit για την τροφοδοσία του καταχωρητή εντολών. Ωστόσο, η ακριβής χωρητικότητα της μνήμης αυτής διαφέρει από μικροελεγκτή σε μικροελεγκτή. Στην μνήμη προγράμματος αποθηκεύονται οι εντολές και τα διανύσματα διακοπών (interrupt vectors). Όλες οι εντολές έχουν μήκος 16 bits ή 32 bits (δηλαδή 2 bytes=1 λέξη), άρα καταλαμβάνουν μία ή δύο θέσεις της μνήμης προγράμματος, αφού η μνήμη flash, είναι οργανωμένη σε διάταξη 8K x 16. Η πρόσβαση στη μνήμη προγράμματος γίνεται σε μια περίοδο του σήματος χρονισμού. Επειδή συντελείται μία συνεχής διοχέτευση εντολών (instruction pipeline), η εκτέλεση των περισσότερων εντολών γίνεται σε μια περίοδο του κεντρικού ρολογιού. Αυτό σημαίνει ότι κατά την εκτέλεση μιας εντολής γίνεται ταυτόχρονα ανάκληση από τη μνήμη της επόμενης εντολής μέσω μιας παράλληλης διαδικασίας, όπως είναι προφανές στο ακόλουθο σχήμα. Σε αυτό απεικονίζεται η παράλληλη διαδικασία ανάκλησης (από τη μνήμη προγράμματος) και εκτέλεσης εντολών (με τη

χρήση της μνήμης δεδομένων), η οποία καθίσταται δυνατή χάρη στην αρχιτεκτονική Harvard και το ταχείας πρόσβασης Register File.



Σχήμα 2.3: Παράλληλη ανάκληση και εκτέλεση εντολών

Κάθε εντολή η οποία ανακαλείται, φορτώνεται στον καταχωρητή εντολών, ο οποίος με τη σειρά του συνδέεται με την ομάδα των καταχωρητών εργασίας και επιλέγει τους κατάλληλους καταχωρητές προκειμένου να χρησιμοποιηθούν από την αριθμητική μονάδα ALU για την εκτέλεση της συγκεκριμένης εντολής. Επιπρόσθετα, η έξοδος του καταχωρητή εντολών αποκωδικοποιείται μέσα από τη μονάδα του αποκωδικοποιητή εντολών, όπου προκύπτουν τα απαραίτητα σήματα ελέγχου για την ολοκλήρωση της εκτέλεσης της τρέχουσας εντολής. Η διάρκεια της μνήμης είναι τουλάχιστον 1000 κύκλοι εγγραφής / διαγραφής. Προκειμένου να προστατεύεται το λογισμικό, η μνήμη χωρίζεται στο τμήμα εκκίνησης και το τμήμα εφαρμογών, τα οποία διαθέτουν bits κλειδώματος για ασφάλεια εγγραφής και ανάγνωσης / εγγραφής.



Σχήμα 2.4: Διαδικασία εκτέλεσης εντολής στην ALU

Η μνήμη δεδομένων του συστήματος διαιρείται σε πολλούς διαφορετικούς τύπους και συνδέεται με έναν διάδρομο των 8 bit για την επικοινωνία των περιφερειακών μονάδων με τους καταχωρητές ελέγχου. Χωρίζεται στα ακόλουθα πέντε τμήματα:

1. Ένα τμήμα που αντιστοιχεί στην ομάδα καταχωρητών εργασίας (register file) των 8 bits. Η ομάδα αυτή των καταχωρητών υπάρχει σε όλους τους μικροελεγκτές της οικογένειας των AVR.
2. Ένα τμήμα 64 καταχωρητών εισόδου – εξόδου των 8 bits. Δε διαθέτουν όλοι οι μικροελεγκτές της σειράς AVR και τους 64 παραπάνω καταχωρητές. Ανάλογα με το πλήθος των εσωτερικών περιφερειακών μονάδων κάποιοι διαθέτουν περισσότερους.
3. Ένα τμήμα εσωτερικής στατικής μνήμης SRAM.
4. Ένα τμήμα εξωτερικής στατικής μνήμης SRAM.
5. Ένα τμήμα εσωτερικής μνήμης τύπου EEPROM. Ο τύπος αυτός εσωτερικής μνήμης είναι διαθέσιμος σχεδόν σε όλους τους μικροελεγκτές AVR και η πρόσβασή του επιτυγχάνεται σε ιδιαίτερο χώρο διευθύνσεων μνήμης. Η μνήμη αυτή μπορεί να διαβαστεί και να γραφεί από οποιοδήποτε πρόγραμμα.

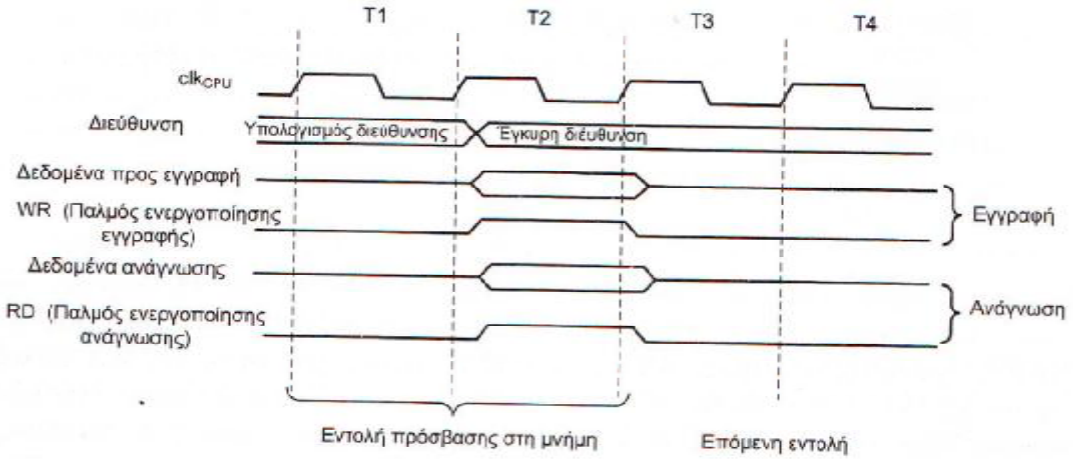
2.5 Μνήμη SRAM

Η στατική RAM (Static RAM - SRAM) είναι ένας τύπος μνήμης RAM που έχει την ικανότητα να διατηρεί αναλλοίωτα τα περιεχόμενά της για όσο χρονικό διάστημα τροφοδοτείται με ρεύμα, χωρίς να απαιτείται κάποια επιπλέον εξωτερική επέμβαση. Στις SRAMs χρησιμοποιούνται ειδικοί διακόπτες (switches), που μπορεί να είναι ανοικτοί ή κλειστοί (on/off). Ο τρόπος κατασκευής της SRAM μοιάζει περισσότερο με την τεχνολογία που εφαρμόζεται στους επεξεργαστές: πολλά ολοκληρωμένα κυκλώματα, που έχουν τοποθετηθεί πάνω σε μια μικρή πλακέτα πυριτίου. Για την αποθήκευση κάθε bit στη μνήμη SRAM, απαιτούνται τέσσερα με έξι transistors, γι’ αυτό και το μέγεθος της SRAM είναι μεγαλύτερο από το μέγεθος της DRAM (κάθε bit χρειάζεται μόνο έναν πυκνωτή). Η πολυπλοκότητα και ο μεγάλος αριθμός transistors που χρησιμοποιούνται για τη μνήμη SRAM, είναι ο βασικότερος λόγος, εξαιτίας του οποίου η μνήμη SRAM έχει μεγαλύτερο κόστος από την DRAM (Dynamic RAM[1]).

[1] Η δυναμική RAM (Dynamic RAM - DRAM) είναι ένας τύπος RAM, που μπορεί να διατηρήσει αναλλοίωτα τα περιεχόμενά της, μόνο αν γίνεται συνεχής αναζωογόνηση σε αυτά από ένα ειδικό κύκλωμα που ονομάζεται refresh circuit (κύκλωμα αναζωογόνησης). Το κύκλωμα αναζωογόνησης διαβάζει με μεγάλη συχνότητα τα περιεχόμενα κάθε πυκνωτή (σε κάθε πυκνωτή μπορούμε να

2.5.1. Εσωτερική μνήμη SRAM:

Η μνήμη δεδομένων SRAM (STATIC RAM ή αλλιώς στατική RAM), διατίθενται στα περισσότερα μοντέλα μικροελεγκτών. Στις περιπτώσεις κατά τις οποίες τα δεδομένα μπορούν να αποθηκευτούν σε καταχωρητές, η χρήση της είναι δυνατόν να αποφευχθεί. Πρόκειται για μία μνήμη η οποία δεν είναι απευθείας προσβάσιμη από την Κεντρική Μονάδα Επεξεργασίας (ALU), σε αντίθεση με τους καταχωρητές. Το μέγεθός της κυμαίνεται από 128 bytes σε 4096 bytes, ενώ για την πρόσβαση σε θέσεις μνήμης συνήθως χρησιμοποιούμε έναν καταχωρητή για προσωρινή ενδιάμεση αποθήκευση. Η μνήμη αυτή χρησιμοποιείται εκτός από την αποθήκευση μεταβλητών και ως στοίβα (Stack) του συστήματος, καθώς εάν εμφανιστεί μία διακοπή, η τρέχουσα κάθε φορά τιμή του μετρητή προγράμματος αποθηκεύεται στη στοίβα. Προφανώς οι λειτουργίες μέσω της SRAM είναι πιο αργές σε σχέση με τους καταχωρητές, γεγονός το οποίο οφείλεται στο ότι ο χρόνος πρόσβασης σε αυτή απαιτεί δύο περιόδους του σήματος χρονισμού. Κατά την πρώτη περίοδο εκτελούνται οι απαιτούμενες λειτουργίες των καταχωρητών για τον καθορισμό της διεύθυνσης και στη δεύτερη περίοδο λαμβάνει χώρα η πρόσβαση στη μνήμη (εγγραφή ή ανάγνωση). Οι φάσεις αυτές της εγγραφής και ανάγνωσης απεικονίζονται στο σχήμα που ακολουθεί.



Σχήμα 2.6: Φάσεις εγγραφής – ανάγνωσης SRAM

αποθηκεύσουμε ένα bit), ανεξάρτητα αν τα περιεχόμενα του πυκνωτή χρησιμοποιούνται ή όχι εκείνη τη στιγμή. Εξαιτίας του τρόπου κατασκευής του κυκλώματος αναζωογόνησης, το διάβασμα των περιεχομένων των πυκνωτών έχει ως αποτέλεσμα και τη διατήρηση τους. Αν το κύκλωμα αναζωογόνησης δεν υπήρχε, τότε τα περιεχόμενα της μνήμης θα χάνονταν ακόμα και όταν ο υπολογιστής τροφοδοτούνταν με ρεύμα. Η DRAM ονομάζεται δυναμική RAM, εξαιτίας της διαδικασίας αναζωογόνησης των περιεχομένων της.

Η πιο σημαντική χρήση της μνήμης SRAM είναι η λειτουργία στοίβας, αφού εκεί μπορούμε να αποθηκεύουμε (εντολή push) και μετά να ανακτούμε (εντολή pop) πληροφορίες, είτε πρόκειται για τιμές είτε για το περιεχόμενο ενός καταχωρητή ή τη διεύθυνση επιστροφής υπορουτίνας ή διακοπής του υλικού.

2.5.2. Εξωτερική μνήμη SRAM

Η μνήμη αυτή χρησιμοποιείται μόνο στους μεγαλύτερους επεξεργαστές της σειράς AVR και σε εφαρμογές υψηλών απαιτήσεων. Στους επεξεργαστές εκείνους που διαθέτουν θύρες πρόσβασης σε εξωτερικό δίαυλο δεδομένων και διευθύνσεων, μπορεί να χρησιμοποιηθεί οποιαδήποτε ποσότητα εξωτερικής μνήμης SRAM επιθυμεί ο χρήστης. Ο χρόνος πρόσβασης στην εξωτερική μνήμη διαρκεί τρεις περιόδους χρονισμού, όμως υπάρχουν περιπτώσεις που μπορεί να διαρκέσει τέσσερις.

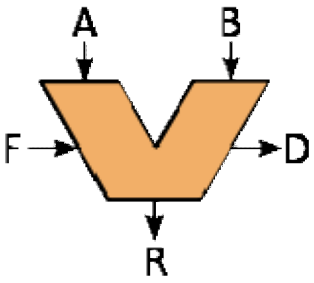
2.6 Η Αριθμητική Λογική Μονάδα (ALU)

Στην επιστήμη των υπολογιστών η Αριθμητική Λογική Μονάδα (ALU) αποτελεί την Κεντρική Μονάδα έως επεξεργαστή και είναι ένα ψηφιακό κύκλωμα το οποίο εκτελεί αριθμητικούς και λογικούς υπολογισμούς, καθώς επίσης και εντολές σε επίπεδο bit, σε σχέση με τα περιεχόμενα των καταχωρητών, ενώ αποθηκεύει τα αποτελέσματα των πράξεων σε συγκεκριμένο καταχωρητή που υποδεικνύεται από την ίδια την εντολή. Οι εντολές εκτελούνται στη διάρκεια μίας και μόνο περιόδου του σήματος χρονισμού. Κάθε πράξη που εκτελείται από την ALU, επηρεάζει τις σημαίες (flags) που αποτελούν τα bits του καταχωρητή κατάστασης (Status), ανάλογα με την εκάστοτε εντολή. Όλα τα άλλα στοιχεία του συστήματος του υπολογιστή – η μονάδα ελέγχου, οι καταχωρητές, η μνήμη, η λειτουργία I/O – υπάρχουν κυρίως για να προσκομίζουν δεδομένα στην ALU για να τα επεξεργαστεί, και κατόπιν εξάγουν τα αποτελέσματα.

Η ALU είναι θεμελιώδες δομικό στοιχείο της Κ.Μ.Ε. του υπολογιστή. Ακόμα και οι πιο απλοί μικροεπεξεργαστές έχουν μία για σκοπούς όπως η διαχείριση μετρητών. Οι σύγχρονοι επεξεργαστές και οι μονάδες επεξεργασίας γραφικών, φιλοξενούν ισχυρές και πολύ πολύπλοκες ALU. Ο μαθηματικός Τζον Φον Νόιμαν πρότεινε τον όρο ALU το 1945, όταν έγραψε μια αναφορά για την θεμέλια κατασκευή ενός καινούριου τότε υπολογιστή, του EDVAC. Το 1946, ο Φον Νόιμαν συνεργάστηκε με τους συναδέλφους του στο σχεδιασμό ενός υπολογιστή για λογαριασμό του Institute for Advanced Study του Princeton του New Jersey. Ο συγκεκριμένος

υπολογιστής αποτέλεσε το πρότυπο για όλους τους μεταγενέστερους υπολογιστές. Σε αυτή του την αναφορά, ο Φον Νόιμαν υπογράμμιζε τι πίστευε ότι θα χρειάζονταν για την υλοποίηση της μηχανής του, συμπεριλαμβανομένης και της ALU. Ο Φον Νόιμαν αναφέρει ότι η ALU είναι απολύτως σημαντική για έναν υπολογιστή, διότι εγγυάται ότι ένας υπολογιστής μπορεί να υπολογίζει βασικούς μαθηματικούς υπολογισμούς, όπως πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση. Ως εκ τούτου ο υπολογιστής πρέπει να περιέχει εξειδικευμένα όργανα για να εκτελεί αυτές τις πράξεις. Η έρευνα γύρω από τις ALU εξακολουθεί να αποτελεί σημαντικό μέρος της επιστήμης των υπολογιστών.

Μια ALU, πρέπει να επεξεργάζεται αριθμούς χρησιμοποιώντας την ίδια μορφή όπως και το υπόλοιπο ψηφιακό κύκλωμα. Ως γνωστόν, η μορφή με την οποία αναπαριστώνται όλοι οι αριθμοί (και χαρακτήρες) στους σύγχρονους υπολογιστές είναι με την χρησιμοποίηση του δυαδικού συστήματος (με 0 και 1). Οι πρώτοι υπολογιστές χρησιμοποιούσαν ένα πλήθος από διαφορετικά συστήματα. Για κάθε ένα σύστημα, οι ALU είχαν διαφορετικούς σχεδιασμούς συχνά αρκετά πολύπλοκους. Έτσι καθιερώθηκε το αριθμητικό σύστημα του συμπληρώματος ως προς 2, μια αριθμητική παράσταση που καθιστά ευκολότερους τους υπολογισμούς για την ALU. Με αυτό το αριθμητικό σύστημα, η αφαίρεση επιτυγχάνεται με την πρόσθεση του αρνητικού αριθμού (π.χ. αντί για 5-3 , 5+(-3)), και έτσι εξουδετερώνεται η ανάγκη για εξειδικευμένα κυκλώματα για την αφαίρεση. Οι περισσότερες από τις πράξεις ενός επεξεργαστή εκτελούνται από μία ή περισσότερες ALU. Μια ALU, φορτώνει δεδομένα από καταχωρητές εισόδου, μια εξωτερική μονάδα λέει στην ALU, τι πράξεις να κάνει στα δεδομένα, και στο τέλος η ALU αποθηκεύει το αποτέλεσμα σε έναν εξωτερικό καταχωρητή. Τέλος, ειδικοί μηχανισμοί μετακινούν τα δεδομένα από τους καταχωρητές στη μνήμη.



Σχήμα 2.8: Σχηματικός συμβολισμός της Αριθμητικής Λογικής Μονάδας (ALU).

Οι περισσότερες ALU, μπορούν να εκτελέσουν τις παρακάτω πράξεις :

• Ακέραιες αριθμητικές πράξεις (πρόσθεση, αφαίρεση και μερικές φορές πολ/σμό και διαίρεση, αν και η υλοποίηση τους είναι ακριβή).

• Λογικές πράξεις (και, ή , όχι)

• Μετατόπιση η περιστροφή μιας λέξης τόσο όσο ενός καθορισμένου αριθμού bits, προς τα αριστερά ή δεξιά προσημασμένους ή μη. Οι μετατοπίσεις μπορούν να υλοποιηθούν σαν πολλαπλασιασμοί με το 2 και ως διαιρέσεις με 2.

Ένας μηχανικός μπορεί να σχεδιάσει μια ALU για τον υπολογισμό οποιασδήποτε πράξης, άσχετα με το πόσο πολύπλοκο μπορεί να είναι. Το πρόβλημα έγκειται στο ότι όσο πιο πολύπλοκος είναι ένας υπολογισμός, τόσο πιο ακριβή γίνεται η ALU, καταλαμβάνει περισσότερο χώρο μέσα στον επεξεργαστή και καταναλώνει περισσότερη ενέργεια. Ως εκ τούτου, οι μηχανικοί πάντοτε θέτουν ένα συμβιβασμό, έτσι ώστε να παρέχουν στον επεξεργαστή μια ALU αρκετά δυνατή για να κάνει τον επεξεργαστεί γρήγορο, αλλά και όχι τόσο πολύπλοκο για να μην γίνεται το κόστος και το μέγεθος του απαγορευτικό. Φανταστείτε ότι πρέπει να υπολογίσετε την τετραγωνική ρίζα ενός αριθμού. Ο μηχανικός θα εξετάσει τις ακόλουθες επιλογές για την υλοποίηση αυτού του υπολογισμού :

1. Σχεδίαση μιας υπερβολικά σύνθετης ALU για τον υπολογισμό της τετραγωνικής ρίζας οποιουδήποτε αριθμού σε ένα μόνο βήμα. Αυτό ονομάζεται υπολογισμός σε έναν κύκλο ρολογιού.
2. Σχεδίαση μιας πολύ σύνθετης ALU, που υπολογίζει την τετραγωνική ρίζα οποιουδήποτε αριθμού σε διάφορα στάδια. Τα ενδιάμεσα στάδια όμως περνούν μέσα από μια σειρά κυκλωμάτων διατεταγμένα σε μια γραμμή, σαν μια γραμμή παραγωγής εργοστασίου. Αυτό επιτρέπει στην ALU να δέχεται νέους αριθμούς για υπολογισμό ακόμη και πριν ολοκληρώσει τον υπολογισμό των προηγούμενων αριθμών.
3. Σχεδίαση μιας σύνθετης ALU, που υπολογίζει την τετραγωνική ρίζα με πολλά βήματα. Αυτό ονομάζεται διαδραστικός υπολογισμός, και συνήθως ελέγχεται από μια σύνθετη μονάδα ελέγχου με ενσωματωμένο μικροκώδικα.
4. Σχεδίαση μιας απλής ALU στον επεξεργαστή, και πώληση ενός ξεχωριστού, εξειδικευμένου και πιο δαπανηρού επεξεργαστή που μπορεί να εγκαταστήσει ο πελάτης ακριβώς δίπλα, και ο οποίος υλοποιεί μία από τις παραπάνω εντολές. Αυτός ο επεξεργαστής ονομάζεται συν-επεξεργαστής.

5. Λέμε στους προγραμματιστές ότι δεν υπάρχει συν-επεξεργαστής, έτσι ώστε να σχεδιάσουν τους δικούς τους αλγορίθμους για τον υπολογισμό των τετραγωνικών ριζών σε λογισμικό. Αυτό γίνεται με τη χρήση βιβλιοθηκών λογισμικού.
6. Μίμηση της ύπαρξης ενός συν-επεξεργαστή, δηλαδή κάθε φορά που ένα πρόγραμμα επιχειρεί να εκτελέσει τον υπολογισμό μιας τετραγωνικής ρίζας, κάνε τον επεξεργαστή να ελέγξει την ύπαρξη ενός συν-επεξεργαστή, και κάνε τον να τον χρησιμοποιήσει εάν υπάρχει. Εάν δεν υπάρχει, διέκοψε την επεξεργασία του προγράμματος και επικαλέσου το λειτουργικό σύστημα να εκτελέσει τον υπολογισμό της ρίζας μέσω ενός λογισμικού αλγορίθμου.

Οι παραπάνω επιλογές πάνε από την γρηγορότερη και πιο ακριβή προς την πιο αργή και φθηνότερη. Ως εκ τούτου, ενώ ακόμα και οι πιο απλοί υπολογιστές μπορούν να υπολογίσουν ακόμα και τις πιο πολύπλοκες μαθηματικές συναρτήσεις, οι απλούστεροι υπολογιστές θα κάνουν περισσότερο χρόνο να το κάνουν εξαιτίας των πολλών βημάτων που πρέπει να διεκπεραιώσει για τον υπολογισμό της συνάρτησης. Οι είσοδοι στην ALU είναι τα δεδομένα πάνω στα οποία θα εφαρμοστούν οι διεργασίες (ονομάζονται τελεστές), και έναν κώδικα από την μονάδα ελέγχου που αναφέρει ποια λειτουργία θα εκτελεστεί. Σε πολλά σχέδια η ALU λαμβάνει επίσης ως εισόδους και τα σύνολα των κωδικών κατάστασης από τον καταχωρητή κατάστασης. Οι κωδικοί αυτοί χρησιμοποιούνται για να δηλώσουν καταστάσεις όπως η υπερχείλιση, διαίρεση με μηδέν κλπ.

2.7 Καταχωρητής εργασίας

Οι μικροελεγκτές διαθέτουν 32 καταχωρητές εργασίας τον 8-bits και συμβολίζονται R0-R31.Οι περισσότερες εντολές που χρησιμοποιούνται τους καταχωρητές εργασίας (πλην τον καταχορητον δείκτη) ολοκληρώνονται σε μια περίοδο εκτός από τις εντολές όπως ADIW,SBIW,MUL,FMUL και LD,ST,LDD,STD,LPM,SMP.Αυτές οι τελευταίες εντολές χρησιμοποιούν μονό τους R26-R31.

2.7.1 Καταχωρητές Δείκτη

Τα ζεύγη των καταχωρητών R26:R27,R28:R29 και R30:R31 η αντίστοιχα X,Y,Z χρησιμοποιούνται ως 16-bit καταχωρητές δείκτη προς θέσεις μνήμης SRAM η προς θέσεις μνήμης προγράμματος (μοναχά ο Z).

2.7.2 Καταχωρητές Κατάστασης (Status register)

Ο Καταχωρητής κατάστασης είναι 8-bits,τα όποια αντιστοιχούν στις σημαίες του συστήματος . Οι σημαίες δίνουν πληροφορίες για την κατάσταση του συστήματος και μηδενίζονται κατά την εκκίνηση η επανατοποθέτηση του συστήματος (reset).Η διεύθυνση στην μνήμη εισόδου-εξοδου είναι $3F

bit	7	6	5	4	3	2	1	0
Σημαία	I	T	H	S	V	N	Z	C

.

Η σημασία των σημαιών είναι η έξης

1) bit7 (I) καθολική ενεργοποίηση διακοπών (Global Interrupt Enable-GIE).Θέτοντας το bit αυτό ενεργοποιούμε τις διακοπές.

2) Bit6 (T) σημαία αντιγραφής-αποθηκευσης (bit Copy Storage).Με χρηση των εντολών BLD και BST επιτυγχάνουμε την ανάγνωση και αποθήκευση συγκεκριμένων bits.

3) bit5 (H). σημαία δεκαδικού κρατουμένου (Half carry flag).Ενημερώνει για την ύπαρξη δεκαδικού κρατουμένου μετά από αριθμητικές πράξεις .

4) bit4 (S) σημαία πρόσημου (Sign Flag).Ενημερώνει για το πρόσημο ενός καταχωρητή και ισούται με λογικό OR των σημαιών αρνητικού πρόσημου και υπερχείλισης (S=N or V)

5) bit3 (V) σημαία υπερχείλισης (Overflow Flag) για αριθμητική συμπληρώματος του δυο.

6) bit2 (N) σημαία αρνητικού πρόσημου (Negative flag).Τίθεται όταν το αποτέλεσμα μιας πράξης είναι μηδέν.

7) bit1 (Z) σημαία μηδενισμού (zero flag).Τίθεται όταν το αποτέλεσμα μιας πράξης είναι μηδέν.

8) bit0 (C) σημαία κρατουμένου (Carry flag).Ενημερώνει για την ύπαρξη η όχι κρατουμένου μετά από αριθμητικές η λογικές πράξεις.

Οι σημαίες θέτονται η μηδενίζονται μέσω αριθμητικών,λογικών ,σύγκρισης ,επίπεδου bit,ολίσθησης και άλλων εντολών.

2.8 ΕΝΤΟΛΕΣ AVR

Οι εντολές της γλώσσας των μικροελεγκτών μπορούν να ομαδοποιηθούν στις παρακάτω τέσσερις κατηγορίες:

1. **Εντολές μεταφοράς δεδομένων μεταξύ καταχωρητών ή μεταξύ καταχωρητών και μνήμης ή μεταξύ καταχωρητή και σταθεράς**. Ο Καταχωρητής είναι ένας από τους 32 καταχωρητές εργασίας (που αναγνωρίζονται με τα σύμβολα από R0 έως R31) και διαθέτουν το αντίστοιχο τμήμα μνήμης (Register file). Η ομάδα των καταχωρητών εργασίας χωρίζεται σε δύο τμήματα καθένα από τα οποία συνολικά διαθέτει 16 καταχωρητές, από τον R0έως τον R15 και από τον R16 στον R31. Όλες σχεδόν οι εντολές που χρησιμοποιούν ως τελεστές τους τους καταχωρητές εργασίας, έχουν άμεση πρόσβαση σε όλους τους καταχωρητές και ολοκληρώνονται σε διάρκεια μίας και μόνο περιόδου του ρολογιού.

Εντολές μεταφοράς δεδομένων					
MOV	Rd, Rr	Move Between Registers	Rd ß Rr	Καμία	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ß Rr+1:Rr	Καμία	1
LDI	Rd, K	Load Immediate	Rd ß K	Καμία	1
LD	Rd, X	Load Indirect	Rd ß (X)	Καμία	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ß (X), X ß X+ 1	Καμία	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ß X – 1, Rd ß (X)	Καμία	2

LDD	Rd, Z+q	Load Indirect with Displacement	Rd ß (Z+ q)	Καμία	2
LDS	Rd, k	Load Direct from SRAM	Rd ß (k)	Καμία	2
ST	X, Rr	Store Indirect	(X) ß Rr	Καμία	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ß Rr, X ß X+ 1	Καμία	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ß X – 1, (X) ß Rr	Καμία	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ß Rr	Καμία	2
STS	k, Rr	Store Direct to SRAM	(k) ß Rr	Καμία	2
LPM	Rd, Z	Load Program Memory	R0 ß (Z)	Καμία	3
SPM	Rr	Store Program Memory	(Z) ß R1:R0	Καμία	-
IN	Rd	In Port	Rd ß P	Καμία	1
OUT	Rd, Rr	Out Port	P ß Rr	Καμία	1
PUSH	Rd, Rr	Push Register on Stack	STACK ß Rr	Καμία	2
POP	Rd, K	Pop Register from Stack	Rd ß STACK	Καμία	2

Πίνακας 2.1: Εντολές μεταφοράς δεδομένων

Όπου Rr Καταχωρητής προέλευσης και Rd Καταχωρητής προορισμού του Register File, k μια σταθερή διεύθυνση και K μια σταθερά. Οι καταχωρητές X,Y,Z λέγονται καταχωρητές δείκτη και αντιστοιχούν στα ζεύγη X=R27:R26, Y=R29:R28 και Z=R31:R30. Χρησιμοποιούνται ως 16-bit καταχωρητές δείκτη για πρόσβαση προς θέσεις μνήμης SRAM ή προς θέσεις μνήμης προγράμματος (μονάχα ο Z). Τέλος, q (6-bit) είναι η μετατόπιση για έμμεση διευθυνσιοδότησης. Οι εντολές LD, ST εφαρμόζονται και για τους τρεις καταχωρητές δείκτη.

2. **Εντολές αριθμητικών και λογικών πράξεων**, όπου οι εντολές της ομάδας αυτής προϋποθέτουν τη χρήση της αριθμητικής λογικής μονάδας (ALU) σε αντίθεση με τις εντολές μεταφοράς δεδομένων. Εκτελούνται μόνο μεταξύ καταχωρητών εργασίας ή καταχωρητή εργασίας με σταθερά. Η εντολή διαβάζει τα περιεχόμενα του καταχωρητή, εκτελεί τις πράξεις με αυτά και αποθηκεύει το αποτέλεσμα στον ίδιο ή σε άλλο καταχωρητή. Οι εντολές αριθμητικών πράξεων διακρίνονται σε εντολές πρόσθεσης και αφαίρεσης με ή χωρίς κρατούμενο μεταξύ δύο καταχωρητών ή ενός καταχωρητή και μιας σταθεράς ή ζεύγους καταχωρητών και μιας σταθεράς, και εντολές πολλαπλασιασμού προσημασμένων ή μη.

α. Εντολές πρόσθεσης:

ADD Rd,Rr : Rd ß Rd + Rr

Περιγραφή: Πρόσθεση 2 καταχωρητών χωρίς χρήση κρατουμένου και αποθήκευση στον καταχωρητή προορισμού Rd.

ADC Rd, Rr **:** Rd R ßd + Rr + C

Περιγραφή: Πρόσθεση 2 καταχωρητών με χρήση κρατουμένου και αποθήκευση στον καταχωρητή προορισμού Rd.

ADIW Rd,K **:** Rd+1:Rd Rd ß+1:Rd + K

Περιγραφή: Πρόσθεση μιας τιμής K(0-63) σε ένα ζεύγος καταχωρητών και αποθήκευση στο ζεύγος καταχωρητών.

β. Εντολές αφαίρεσης:

SUB Rd,Rr **:** Rd ß Rd - Rr

Περιγραφή: Αφαίρεση 2 καταχωρητών χωρίς χρήση κρατουμένου και αποθήκευση στον καταχωρητή προορισμού Rd

SUBI Rd,K **:** Rd ßRd - K

Περιγραφή: Αφαίρεση μιας τιμής χωρίς χρήση κρατουμένου από έναν καταχωρητή και αποθήκευση στον καταχωρητή

SUBCI Rd,K **:** Rd ßRd – K-C

Περιγραφή: Αφαίρεση μιας τιμής με χρήση κρατουμένου από έναν καταχωρητή και αποθήκευση στον καταχωρητή

SBC Rd,Rr **:** Rd R ßd – Rr-C

Περιγραφή: Αφαίρεση 2 καταχωρητών με χρήση κρατουμένου και αποθήκευση στον καταχωρητή προορισμού Rd.

SBIWRd,Rd+1:Rd ßRd+1:Rd-K

Περιγραφή: Αφαίρεση μιας τιμής από ένα ζεύγος καταχωρητών και αποθήκευση στο ζεύγος

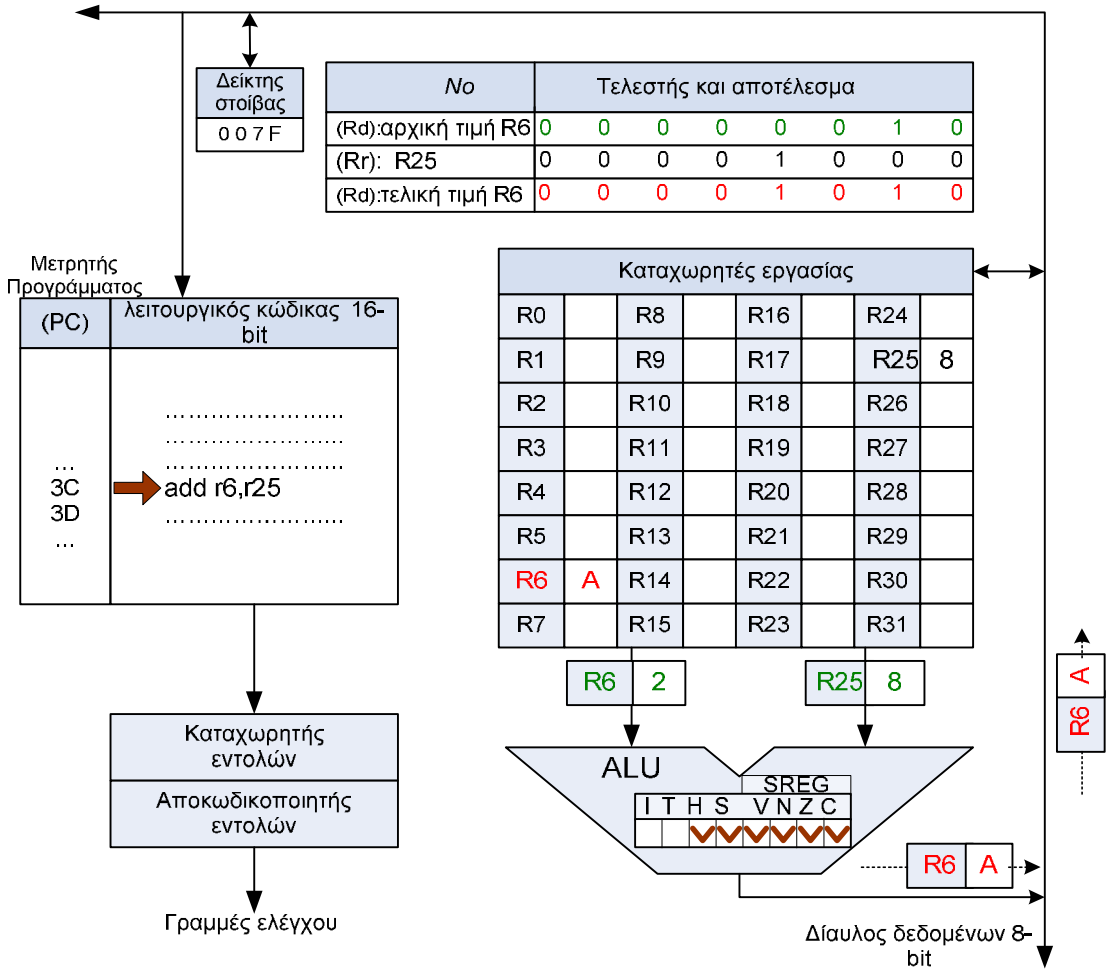
Καταχωρητών.

Υπάρχουν εντολές πολλαπλασιασμού μεταξύ προσημασμένων ή μη, περιλαμβανομένου του κλασματικού τμήματος ή όχι, αλλά δεν υποστηρίζονται από όλα τα μοντέλα μικροελεγκτή.

γ. Εντολές πολλαπλασιασμού:

MUL Rd,Rr **:** R1, R0 **ß** Rd **û**Rr

Περιγραφή: Μη προσημασμένος πολλαπλασιασμός μεταξύ των περιεχομένων των καταχωρητών Rd,Rr. Το αποτέλεσμα αποθηκεύεται στο ζεύγος καταχωρητών R1, R0.



Σχήμα 2.9: Παράδειγμα απευθείας διευθυνσιοδότησης δύο καταχωρητών.

Επίσης, εδώ ανήκουν οι εντολές αύξησης, ελάττωσης, μηδενισμού ή θεσίματος καταχωρητή ή bit καταχωρητή, συμπληρώματος του ένα και του δύο.

Οι εντολές λογικών πράξεων διακρίνονται σε εντολές λογικού ΚΑΙ, Η , ΑΠΟΚΛΕΙΣΤΙΚΟΥ Η:

AND Rd,Rr **:** Λογικό AND Rd ß Rd ŸRr

Περιγραφή: Λογικό AND μεταξύ των περιεχομένων των καταχωρητών Rd,Rr

OR Rd,Rr **:** Λογικό OR Rd ß Rd v Rr

Περιγραφή: Λογικό OR μεταξύ των περιεχομένων των καταχωρητών Rd,Rr

EOR Rd,Rr **:** Λογικό EXOR Rd ß Rd +Rr

Περιγραφή: Λογικό EXOR (αποκλειστικό ‘ή’) μεταξύ των περιεχομένων των καταχωρητών Rd,Rr.

Αντίστοιχα, υπάρχουν οι εντολές λογικού ΚΑΙ, Η μεταξύ κατ/τη και σταθεράς: ANDI Rd,K , ORI Rd,K.

Εντολές συμπληρώματος του 1 και 2:

COM Rd **:** Rd ß $FF - Rd

Περιγραφή: Αντιστροφή των περιεχομένων του καταχωρητή Rd με χρήση αριθμητικής συμπληρώματος του 1.

NEG Rd **:** Rd ß $00 - Rd

Περιγραφή: Αντιστροφή των περιεχομένων του καταχωρητή Rd με χρήση αριθμητικής συμπληρώματος του 2.

Τοποθέτηση λογικού ‘1’ ή ‘0’ στα bits καταχωρητή:

SBR Rd,K **:** Rd ß Rd v K

Περιγραφή: Τοποθέτηση λογικού ‘1’ στα bits του καταχωρητή Rd εκτελώντας λογικό ORI μεταξύ των περιεχομένων του καταχωρητή Rd και της σταθεράς K

CBR Rd,K **:** Rd ß Rd Ÿ($FFh -K)

Περιγραφή: Τοποθέτηση λογικού ‘0’ στα bits του καταχωρητή Rd εκτελώντας λογικό AND μεταξύ των περιεχομένων του καταχωρητή Rd και της σταθεράς K.

Αύξηση/ μείωση καταχωρητή:

INC Rd **:** Rd ß Rd + 1

Περιγραφή: Αύξηση των περιεχομένων του καταχωρητή Rd κατά μία μονάδα.

DEC Rd**:** Rd ß Rd - 1

Περιγραφή: Μείωση των περιεχομένων του καταχωρητή Rd κατά μία μονάδα.

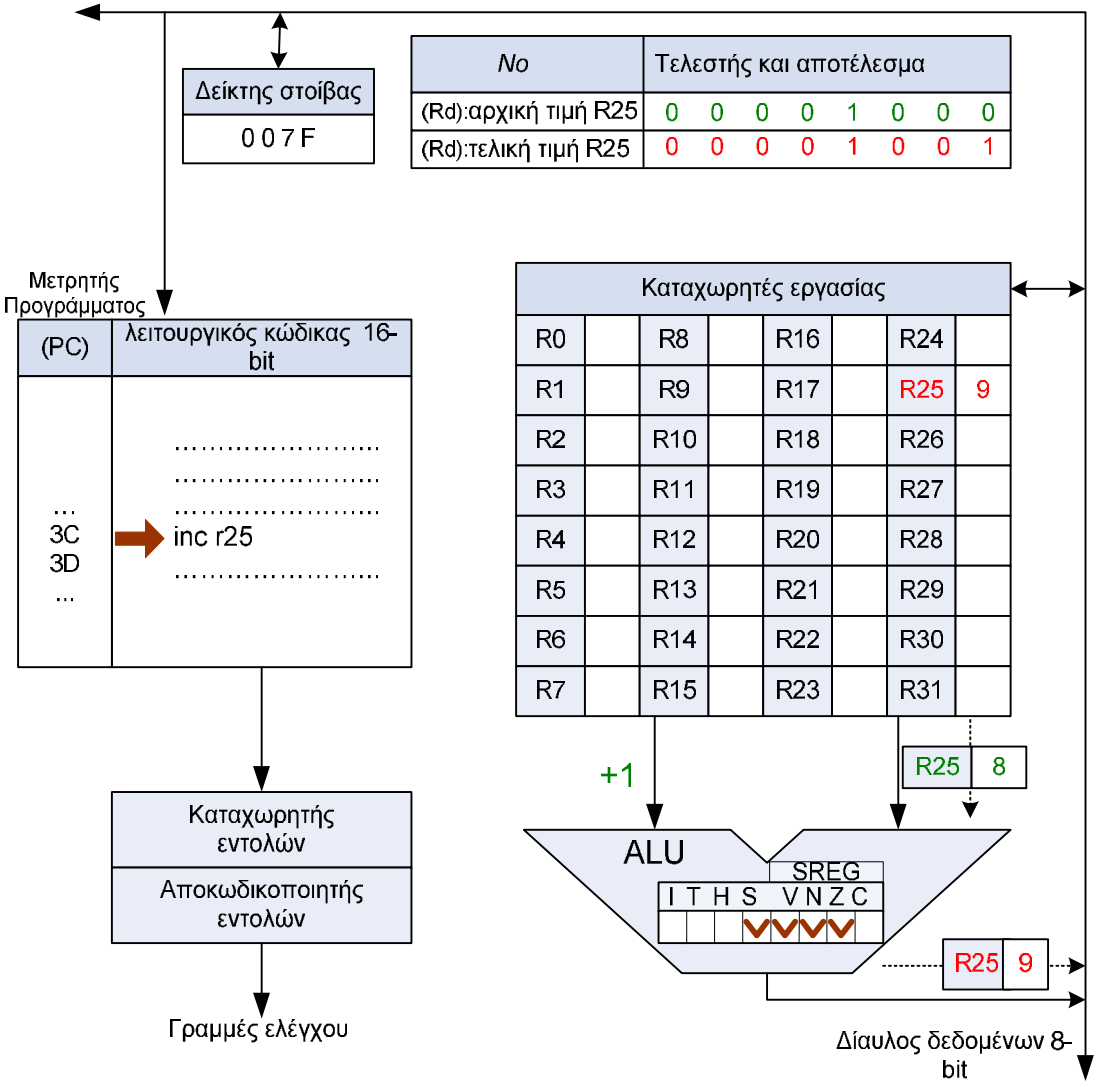
Μηδενισμός/ θέσιμο καταχωρητή:

CLR Rd **:** Rd ß Rd +Rd

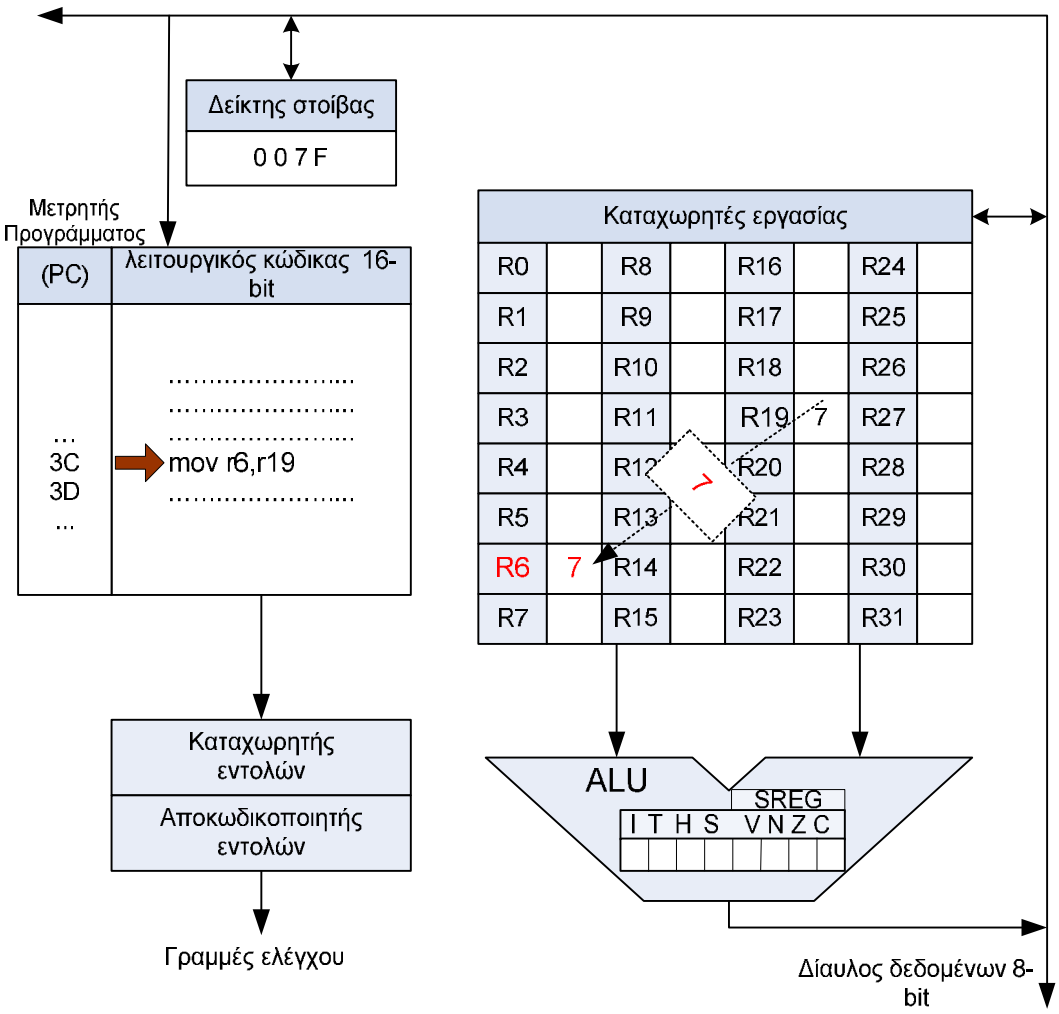
Περιγραφή: Μηδενισμός του καταχωρητή Rd με εκτέλεση αποκλειστικού ‘ή’ μεταξύ του καταχωρητή και του εαυτού του.

SER Rd**:** Rd ß $FF

Περιγραφή: Τοποθέτηση ‘1’ σε όλα τα bits του καταχωρητή με φόρτωση της τιμής $FF.



Σχήμα 2.10: Παράδειγμα άμεσης διευθυνσιοδότησης ενός καταχωρητή.



Σχήμα 2.11: Παράδειγμα μεταφοράς μεταξύ των καταχωρητών

3) **Εντολές σε επίπεδο bit και ελέγχου bit**

Οι εντολές αυτές μας παρέχουν τη δυνατότητα να θέσουμε ή να μηδενίσουμε σημαίες του καταχωρητή κατάστασης, διακοπές και συγκεκριμένα bit καταχωρητών ή θυρών. Επίσης, να ολισθήσουμε κάποιον καταχωρητή μέσω κρατουμένου ή όχι. Πρόκειται για τις εξής εντολές:

Εντολές σε επίπεδο bit και ελέγχου bit					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ß 1	Καμία	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ß 0	Καμία	2

LSL	Rd	Logical Shift Left	Rd(n+1) ß Rd(n), Rd(0) ß 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ß Rd(n+1), Rd(7) ß 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) ß C, Rd(n+1) ß Rd(n), C ß Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ß C, Rd(n) ß Rd(n+1), C ß Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ß Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3-0) ß Rd(7-4) Rd(7-4) ß Rd(3-0)	Καμία	1
BSET	S	Flag Set	SREG(s) ß 1	SREG(s)	1
BCLR	S	Flag Clear	SREG(s) ß 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ß Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ß T	Καμία	1
Sex		Set Flag	x ß 1	X	1
CLx		Clear Flag	x ß 0	X	1

Πίνακας 2.2: Εντολές σε επίπεδο bit και ελέγχου bit.

Όπου x μια από τις σημαίες του καταχωρητή κατάστασης C,N,Z,I,S,V,T,H και b το bit0-7 ενός καταχωρητή ή μιας θύρας (3-bit) , s το bit0-7 του καταχωρητή κατάστασης (3-bit) , k μια σταθερή διεύθυνση και P καταχωρητής εισόδου-εξόδου.

- **<u>Ολίσθηση καταχωρητή αριστερά ή δεξιά μέσω κρατουμένου ή όχι</u>**:

LSL Rd **:** Rd(n+1) □ Rd(n) , Rd(0) □ 0 ,C □Rd(7)

Περιγραφή: Ολίσθηση των bits του καταχωρητή Rd μία θέση προς τα αριστερά..Το bit0 μηδενίζεται και το bit7 φορτώνεται στη σημαία C του SREG.

ROL Rd **:** Rd(0) □ C , Rd(n+1) □ Rd(n) , C □Rd(7)

Περιγραφή: Περιστροφή των bits του καταχωρητή Rd μία θέση προς τα αριστερά μέσω της σημαίας κρατουμένου. Η σημαία C ολισθαίνει στο bit0 και το bit7 ολισθαίνει στη σημαία C.

Αντίστοιχα γίνεται η ολίσθηση προς τα δεξιά με τις εντολές **LSR** Rd, **ROR** Rd.

- **<u>Χειρισμός bit (σημαίας) καταχωρητή κατάστασης</u>**:

BSET s **:** SREG(s) **ß** 1

Περιγραφή: Ενεργοποίηση σημαίας του καταχωρητή κατάστασης.

BCLR s **:** SREG(s) **ß** 0

Περιγραφή: Μηδενισμός σημαίας του καταχωρητή κατάστασης.

- **<u>Χειρισμός bit καταχωρητή εισόδου-εξόδου</u>**:

SBI A,s **:** I/O(A,b) **ß** 1

Περιγραφή: Ενεργοποίηση του bit s ενός καταχωρητή εισόδου-εξόδου.

CBI A,s **:** I/O(A,b) **ß** 0

Περιγραφή: Μηδενισμός του bit s ενός καταχωρητή εισόδου-εξόδου.

- **<u>Χειρισμός bit κατ/τη μέσω σημαίας T</u>**:

BST Rd,b **:** T **ß** Rd(b)

Περιγραφή: Αποθήκευση του bit b του καταχωρητή Rd στη σημαία T του SREG.

BLD Rd,b **:** Rd(b) **ß** T

Περιγραφή: Αποθήκευση της σημαίας T του SREG στο bit b του καταχωρητή Rd.

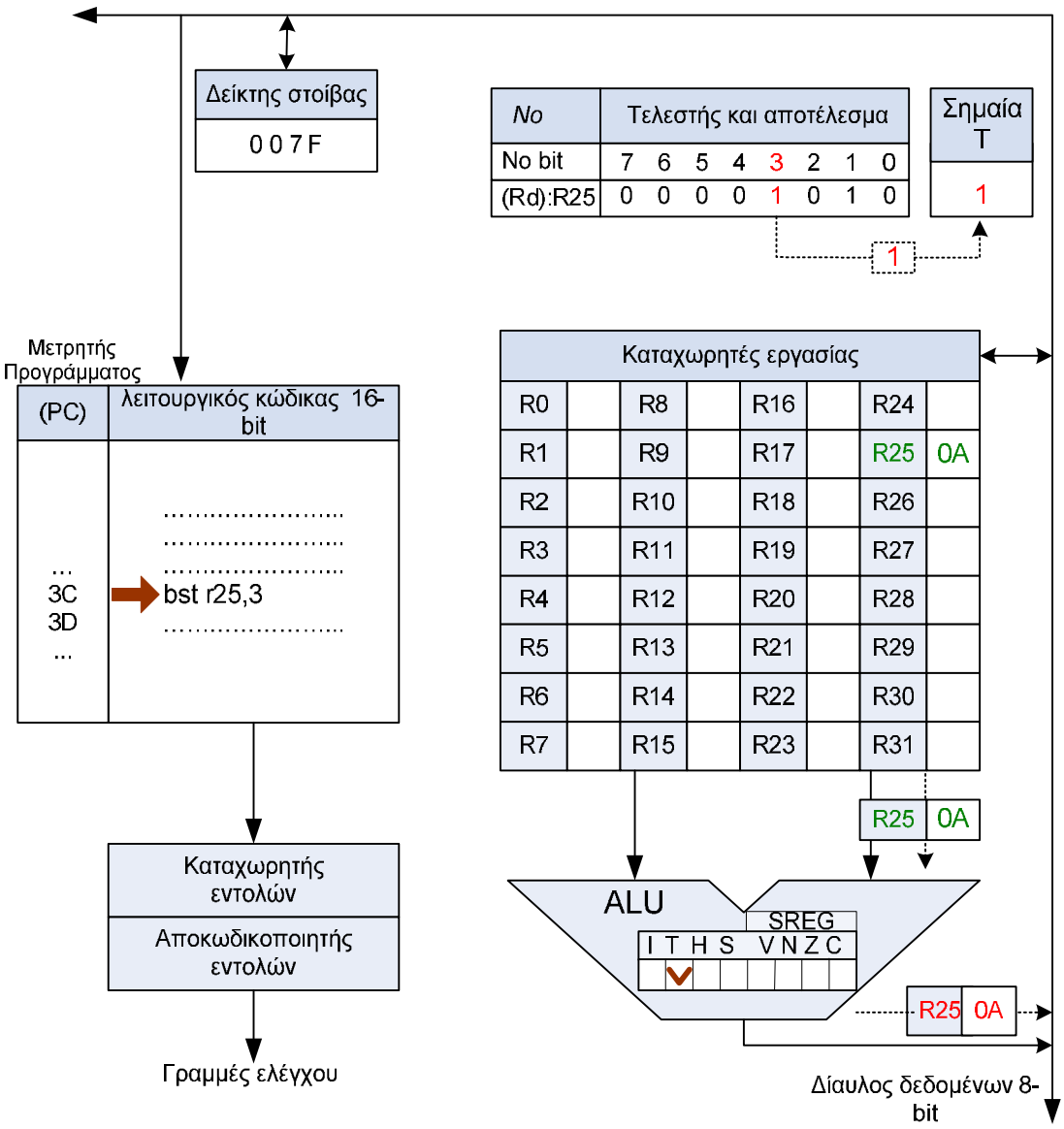
- **<u>Χειρισμός σημαίας καταχωρητή κατάστασης</u>**:

SEx : x **ß** 1

Περιγραφή: Ενεργοποίηση της σημαίας x του καταχωρητή κατάστασης.

CLx : x **ß** 0

Περιγραφή: Μηδενισμός της σημαίας x του καταχωρητή κατάστασης.



Σχήμα 2.12**:** Παράδειγμα εντολής

4)εντολές έλεγχου ροής του προγράμματος και διακλάδωσης

Σε αυτήν την κατηγορία ανήκουν οι εντολές άλματος, παράκαμψης, διακλάδωσης και κλήσης ρουτινών. Πρόκειται για εντολές που αλλάζουν την κανονική ακολουθιακή ροή του προγράμματος. Δηλαδή, όταν εκτελείται μια τέτοια εντολή ο έλεγχος μεταφέρεται σε άλλο σημείο του προγράμματος, αντί να εκτελεστεί η

επόμενη εντολή. Το σημείο αυτό μπορεί να είναι σε μια διεύθυνση της μνήμης προγράμματος, η οποία καθορίζεται από το όρισμα της εντολής. Συγκεκριμένα, ο μετρητής προγράμματος (PC-Program Counter) θα πάρει την επιθυμητή τιμή είτε από τον καταχωρητή Z είτε από μια ετικέτα (label).

Οι εντολές παράκαμψης και διακλάδωσης είναι υπό συνθήκη. Αυτό σημαίνει ότι εξετάζεται η τιμή κάποιου bit για να καθοριστεί αν πρέπει να μεταφερθεί ο έλεγχος ή όχι. Συγκεκριμένα, οι εντολές παράκαμψης εξετάζουν την τιμή ενός bit κάποιου καταχωρητή ή μιας θύρας εισόδου-εξόδου, ενώ οι εντολές διακλάδωσης εξετάζουν μία από τις επτά σημαίες του καταχωρητή κατάστασης. Ο μικροελεγκτής έχει μια πληθώρα εντολών διακλάδωσης, γεγονός που συμβάλλει στην αποδοτικότητα και ευελιξία του κώδικα. Παρακάτω παρατίθενται οι προαναφερθείσες εντολές ανά κατηγορία.

2.7.5.1 Εντολές Παράκαμψης

Εντολές παράκαμψης					
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	Καμία	1/ 2/ 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	Καμία	1/ 2/ 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	Καμία	1/ 2/ 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	Καμία	1/ 2/ 3

Πίνακας 2.5: Εντολές παράκαμψης

Οι εντολές παράκαμψης εξετάζουν την τιμή ενός bit κάποιου καταχωρητή ή μιας θύρας εισόδου-εξόδου για να καθορίσουν αν πρέπει να μεταφερθεί ο έλεγχος ή όχι. Οι σημαίες δεν επηρεάζονται.

SBRC Rr,b **:** Αν Rr(b) = 0 τότε PC ß PC + 2 (ή 3) αλλιώς PC ß PC + 1

Περιγραφή: Αν το bit (b) του καταχωρητή είναι 0 (Rr(b) = 0) τότε παρακάμπτεται η επόμενη εντολή.

SBIC A,b **:** Αν I/O(A,b) = 0 τότε PC ß PC + 2 (or 3) αλλιώς PC ß PC + 1

Περιγραφή: Αν το bit (b) του καταχωρητή εισόδου-εξόδου είναι 0 (A(b) = 0) τότε παρακάμπτεται η επόμενη εντολή.

2.7.5.2 Εντολές διακλάδωσης

Αντίστοιχα, οι εντολές SBRS, SBIS παρακάμπτουν την επόμενη εντολή αν το bit (b) είναι μονάδα.

Εντολές διακλάδωσης					
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC PC+k +1	Καμία	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC PC+k +1	Καμία	1 / 2
BREQ	K	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	Καμία	1 / 2
BRNE	K	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	Καμία	1 / 2
BRCS	K	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	Καμία	1 / 2
BRCC	K	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	Καμία	1 / 2
BRSH	K	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	Καμία	1 / 2
BRLO	K	Branch if Lower	if (C = 1) then PC ← PC + k + 1	Καμία	1 / 2
BRMI	K	Branch if Minus	if (N = 1) then PC ← PC + k + 1	Καμία	1 / 2
BRPL	K	Branch if Plus	if (N = 0) then PC ← PC + k + 1	Καμία	1 / 2
BRGE	K	Branch if Greater or Equal, Signed	if (NV= 0) then PC ← PC + k + 1	Καμία	1 / 2

BRLT	K	Branch if Less Than Zero, Signed	if (NV= 1) then PC ← PC + k + 1	Καμία	1 / 2
BRHS	K	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	Καμία	1 / 2
BRHC	K	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	Καμία	1 / 2
BRTS	K	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	Καμία	1 / 2
BRTC	K	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	Καμία	1 / 2
BRVS	K	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	Καμία	1 / 2
BRVC	K	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	Καμία	1 / 2
BRIE	K	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	Καμία	1 / 2
BRID	K	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	Καμία	1 / 2

Πίνακας 2.6: Εντολές διακλάδωσης

Οι εντολές διακλάδωσης εξετάζουν μία από τις επτά σημαίες του καταχωρητή κατάστασης για να αποφασίσουν αν πρέπει να μεταφερθεί ο έλεγχος ή όχι. Οι σημαίες δεν επηρεάζονται.

Παράδειγμα : BRBC s,k **:** Αν SREG(s) = 0 τότε PC ß PC + k + 1, αλλιώς PC ß PC + 1

Περιγραφή: Σχετικό άλμα υπό συνθήκη. Αν το bit s του καταχωρητή κατάστασης είναι 0 (SREG(s) = 0) τότε εκτελείται άλμα στην διεύθυνση της μνήμης προγράμματος που βρίσκεται k θέσεις μετά σε σχέση με τον μετρητή προγράμματος.

2.7.5.3 Εντολές σύγκρισης

Εντολές σύγκρισης					
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	Καμία	1/ 2/ 3

CP	Rd,Rr	Compare	Rd < Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd < Rr □ ±C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd < K	Z,N,V,C,H	1

Πίνακας 2.7: Εντολές σύγκρισης.

Οι εντολές σύγκρισης μας παρέχουν τη δυνατότητα να συγκρίνουμε δύο καταχωρητές ή έναν καταχωρητή και μια σταθερά.

Παράδειγμα: CPSE Rd,Rr **:** Αν Rd = Rr τότε PC□PC + 2 (ή 3) αλλιώς PC□PC + 1

Περιγραφή: Σύγκριση μεταξύ των καταχωρητών Rd,Rr. Σε περίπτωση ισότητας παρακάμπτεται η επόμενη εντολή.

2.7.5.4 Εντολές άλματος - ρουτινών

Εντολές άλματος- ρουτινών					
RJMP	K	Relative Jump	PC ← PC + k + 1	Καμία	2
IJMP		Indirect Jump to (Z)	PC ← Z	Καμία	2
JMP	K	Direct Jump	PC ← k	Καμία	3
RCALL	K	Relative Subroutine Call	PC ← PC + k + 1	Καμία	3
ICALL		Indirect Call to (Z)	PC ← Z	Καμία	3
CALL	K	Direct Subroutine Call	PC ← k	Καμία	4
RET		Subroutine Return	PC ← STACK	Καμία	4
RETI		Interrupt Return	PC ← STACK	I	4

Πίνακας 2.8: Εντολές άλματος ρουτινών.

Οι επόμενες εντολές μας παρέχουν τη δυνατότητα να εκτελέσουμε άλμα σε επιθυμητή διεύθυνση, να καλέσουμε μια ρουτίνα και να επιστρέψουμε από αυτήν.

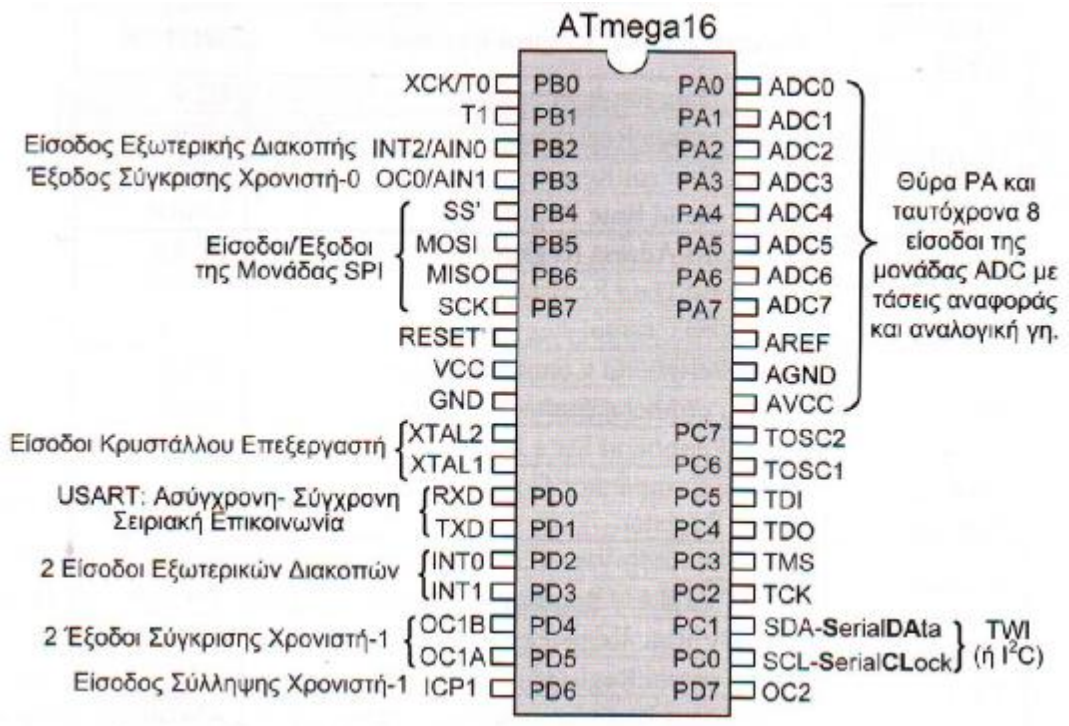
Παράδειγμα: RJMP k **:** PC ß PC + k + 1

4) **Περιγραφή**: Σχετικό άλμα σε μια διεύθυνση της μνήμης προγράμματος μεταξύ PC - 2K

ΚΕΦΑΛΑΙΟ 3 - ΠΕΡΙΦΕΡΕΙΑΚΑ ΤΩΝ ΜΙΚΡΟΕΛΕΓΚΤΩΝ AVR

Οι Μικροελεγκτές AVR διαθέτουν ένα μεγάλο πλήθος περιφερειακών μονάδων, το οποίο όμως πάντα εξαρτάται από τον ειδικότερο τύπο του μικροελεγκτή. Τα περιφερειακά στα οποία αναφερόμαστε είναι τα εξής:

- Μονάδα UART
- Χρονιστές (Timer 0, Timer 1, Timer 2) και Επιτήρησης (Watchdog Timer)
- Μονάδα EEPROM και μονάδα SPI
- Μονάδα Αναλογικού Συγκριτή
- Μονάδα Two – Wire Serial Interface
    - Μονάδα ADC (Analog to Digital Converter)



Σχήμα 3.1: Ακροδέκτες Περιφερειακών AT mega

Στο παραπάνω σχήμα εμφανίζονται οι ακροδέκτες που αντιστοιχούν στις εισόδους – εξόδους των περιφερειακών της νεότερης οικογένειας Mega των μικροελεγκτών AVR που περιλαμβάνει όλη την ποικιλία των περιφερειακών, ενώ ταυτόχρονα διαθέτει αρκετά μεγάλη on chip μνήμη προγράμματος και δεδομένων.

3.1 Η Μονάδα Ασύγχρονης Σειριακής Επικοινωνίας UART

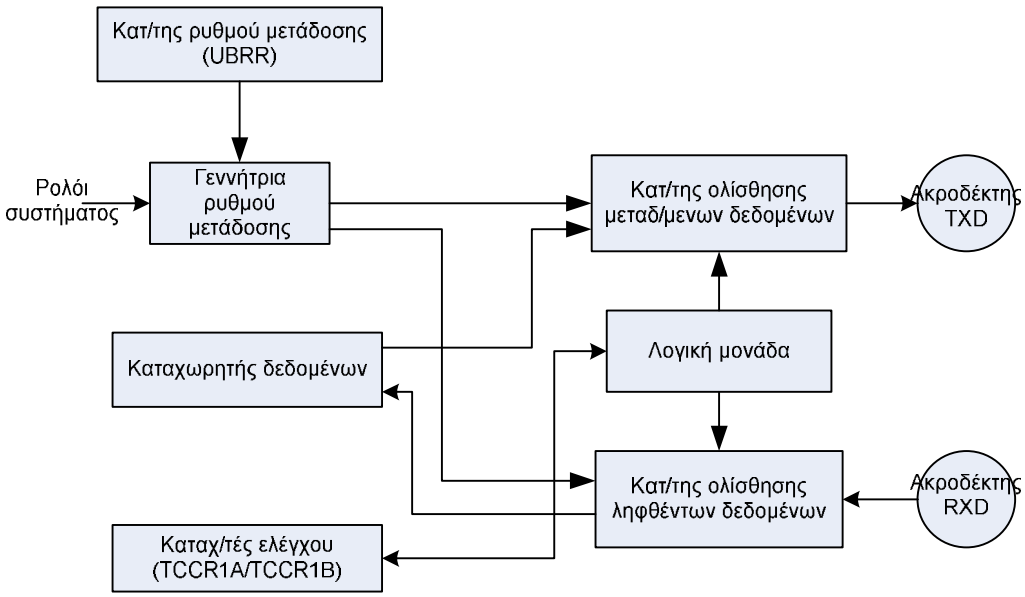
Πρόκειται για ένα πολύ ισχυρό και χρηστικό περιφερειακό, το οποίο χρησιμοποιείται προκειμένου να υλοποιηθεί με επιτυχία η αποστολή και λήψη δεδομένων από ένα PC, για επικοινωνία με το χρήστη, αποσφαλμάτωση κώδικα και άλλα.

Η μονάδα UART μπορεί να μεταδώσει τους εξής 30 συνδυασμούς πλαισίου:

- 1 bit εκκίνησης
- 5,6,7,8 ή 9 bits
- ένα ή δύο bit λήξης
- καμία, άρτια ή περιττή ισοτιμία.

Φιλτράρει τα ληφθέντα δεδομένα και ανιχνεύει σφάλματα πλαισίου και υπέρβασης. Διαθέτει τρία σήματα διακοπών και επιτρέπει μεγάλη ροή δεδομένων με απομονωτές υλοποιημένους από λογισμικό. Διαθέτει πομπό και δέκτη οι οποίοι μοιράζονται τη γεννήτρια του ρυθμού μετάδοσης και τους κατ/τες ελέγχου.

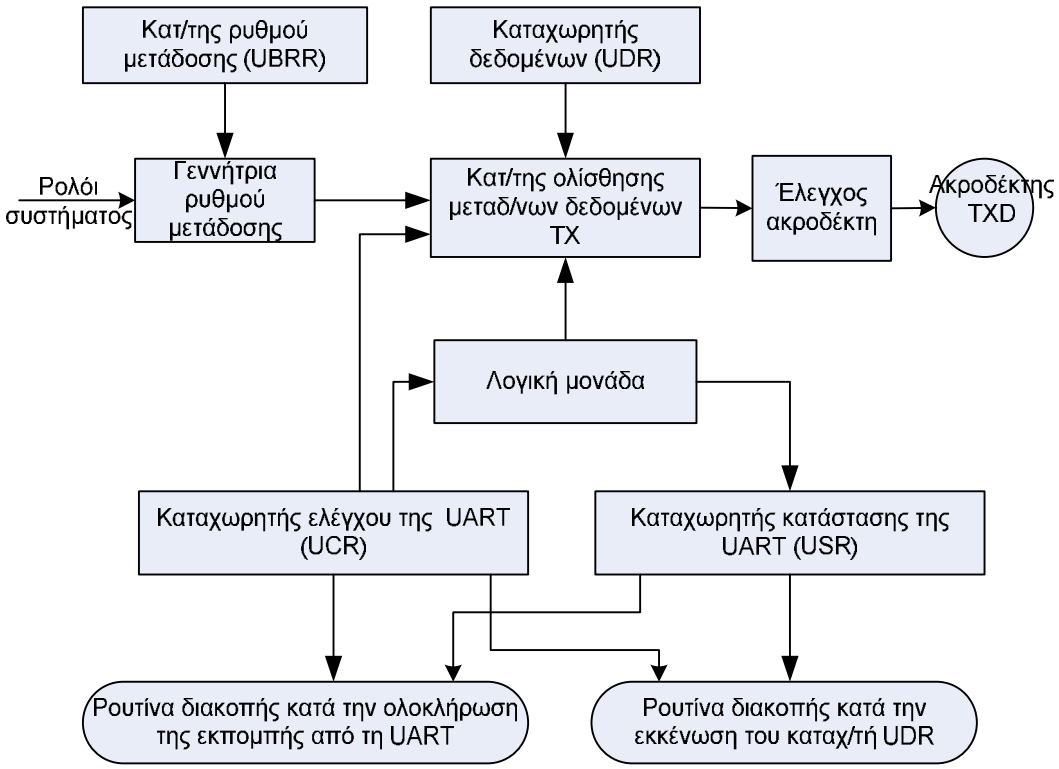
Η βασική δομή φαίνεται στο επόμενο σχήμα:



Σχήμα 3.2: Μονάδα UART

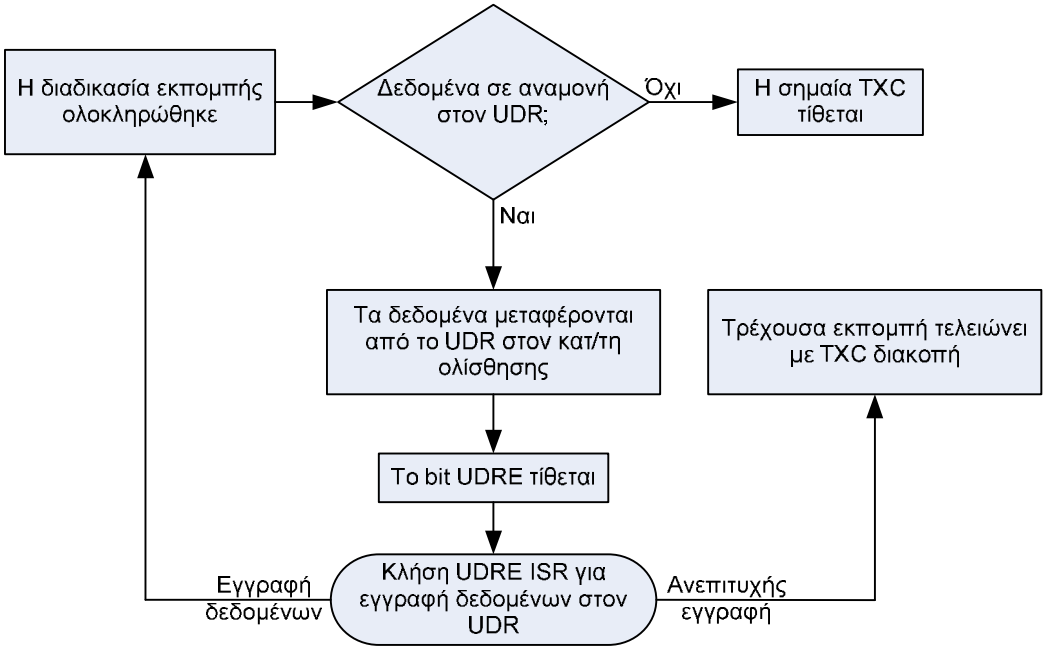
3.2 Πομπός UART

Ο πομπός UART στέλνει δεδομένα από τον μικροελεγκτή σε διάφορες συσκεύες (data logger, PC, κλπ) στον επιθυμητό ρυθμό μετάδοσης.



Σχήμα 3.3: Πομπός UART

Η εκπομπή αρχικοποιείται εγγράφοντας δεδομένα στον κατ/τη UDR. Έπειτα, τα δεδομένα μεταφέρονται στον κατ/τη ολίσθησης εκπομπής TX εφόσον το προηγούμενο byte έχει ολισθήσει πλήρως. Όταν ένα byte μεταφέρεται στον κατ/τη ολίσθησης εκπομπής TX, η σημαία UDRE τίθεται. Η ρουτίνα εξυπηρέτησης διακοπής UDRE ISR μπορεί να γράψει το επόμενο byte στον κατ/τη UDR χωρίς να αλλοιώσει την παρούσα διαδικασία εκπομπής. Όταν ένα byte έχει ολισθήσει πλήρως και δεν έχουν γραφτεί δεδομένα στον κατ/τη UDR από τη ρουτίνα εξυπηρέτησης διακοπής UDRE ISR, τότε η σημαία TXC τίθεται. Το επόμενο διάγραμμα ροής δείχνει πως συνεργάζονται οι σημαίες διακοπών για την μετάδοση:

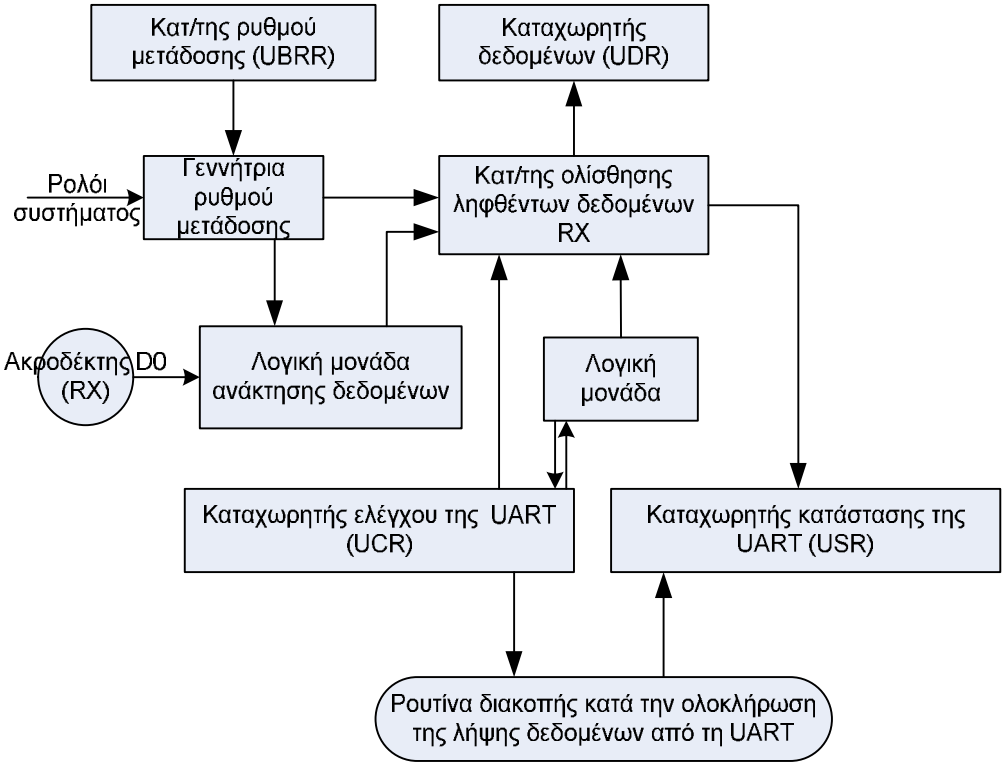


Σχήμα 3.4: Διάγραμμα ροής μετάδοσης

3.3 Ο δέκτης UART

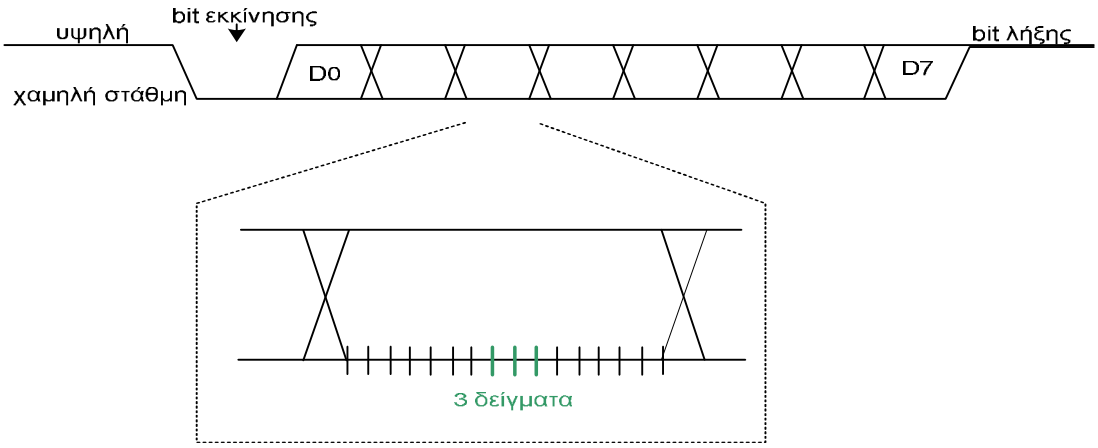
Ο δέκτης UART έχει παρόμοια δομή με τον πομπό, αλλά επιπλέον διαθέτει τα απαραίτητα κυκλώματα για λήψη δεδομένων : το λογικό κύκλωμα ανάκτησης δεδομένων και μία διακοπή για την ολοκλήρωση της λήψης δεδομένο.

.



Σχήμα 3.5: Δέκτης UART

Το δεδομένο δειγματοληπτείται με τον τρόπο που περιγράψαμε στην ενότητα της γεννήτριας ρυθμού μετάδοσης και φαίνεται στο ακόλουθο σχήμα. Διακρίνουμε τις μικρές κάθετες γραμμές που αντιστοιχούν στους παλμούς ρολογιού που παράγει η γεννήτρια ρυθμού μετάδοσης. Οι τρεις κόκκινες αντιστοιχούν στο 8^{o} ,9^{o} και 10^{o} δείγμα. Συνεπώς, ο ρυθμός μετάδοσης αρχικά είναι 16πλάσιος ώστε να δειγματολογήσει τα δεδομένα και έπειτα διαιρείται με το 16 για να ολισθήσει τα δεδομένα.

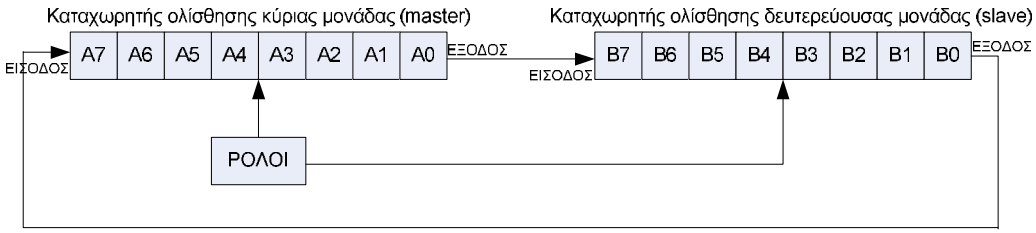


Σχήμα 3.6: Δειγματοληψία δεδομένων

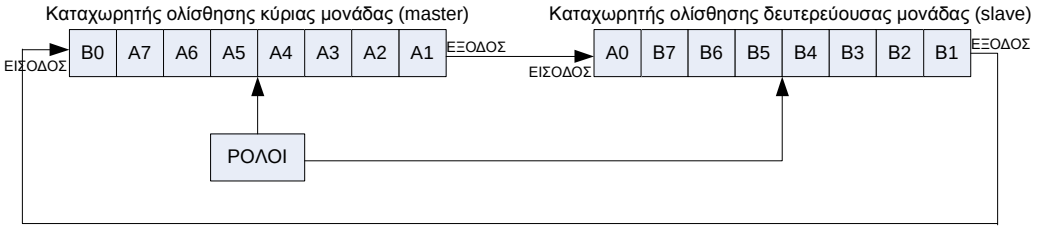
3.4 Μονάδα σύγχρονης σειριακής επικοινωνίας (SPI)

Η μονάδα σύγχρονης σειριακής επικοινωνίας SPI (Serial Peripheral Interface) είναι ένα περιφερειακό που χρησιμοποιείται για την επικοινωνία του μικροελεγκτή με διάφορες διατάξεις, όπως εξωτερικές μνήμες EEPROM, μετατροπείς DAC και ADC, άλλους μικροελεγκτές. Η μονάδα SPI δεν είναι διαθέσιμη σε όλα τα μοντέλα Παρόλα αυτά μπορούμε να χρησιμοποιήσουμε εναλλακτικά τις απλές θύρες εισόδου - εξόδου και να μιμηθούμε τη λειτουργία μιας πραγματικής θύρας SPI με κατάλληλο λογισμικό. Συγκεκριμένα, έχουμε μια κύρια συσκευή (master) που αρχικοποιεί και ελέγχει την επικοινωνία, και μία ή περισσότερες δευτερεύουσες συσκευές (slaves) που λαμβάνουν και εκπέμπουν στην κύρια.

Ο πυρήνας της μονάδας SPI είναι ένας 8-μπιτος Καταχωρητής ολίσθησης ευρισκόμενος σε κύρια και σε δευτερεύουσα συσκευή, καθώς επίσης και το σήμα χρονισμού παραγόμενο από την κύρια διάταξη. Έστω ότι η κύρια διάταξη (master) επιθυμεί να στείλει ένα byte δεδομένων στην δευτερεύουσα (slave) και συγχρόνως να λάβει δεδομένα από τη δευτερεύουσα. Προτού αρχίσει η επικοινωνία ο master και ο slave τοποθετούν τα δεδομένα τους στους κατ/τές ολίσθησης τους (σχήμα α). Τότε ο master παράγει 8 παλμούς ρολογιού και τα περιεχόμενα του κατ/τη ολίσθησης του μεταφέρονται στον κατ/τη ολίσθησης του slave και αντιστρόφως (σχήμα β ως ε). Η εκπομπή και η λήψη γίνονται συγχρόνως, οπότε έχουμε μια πλήρως αμφίδρομη μεταφορά δεδομένων. Η πρώτη εικόνα απεικονίζει τη κατάσταση των κατ/των πριν την μεταφορά, ενώ οι επόμενες απεικονίζουν τη διαδικασία μεταφοράς:

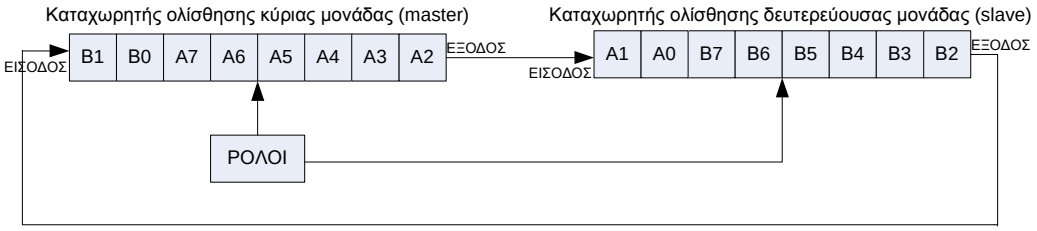


Σχήμα 3.7.1 (α): Μεταφορά δεδομένου μεταξύ master και slave διάταξης



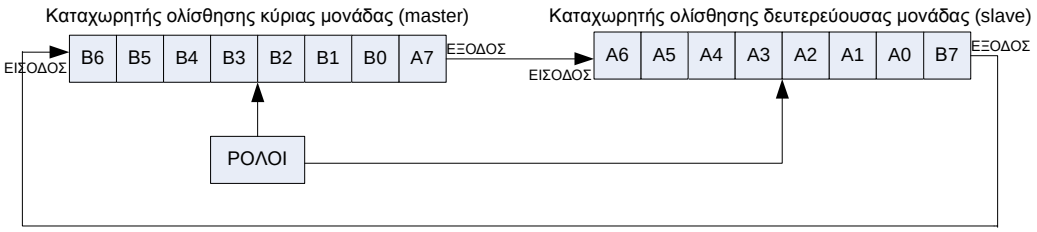
Η κύρια συσκευή (master) παράγει τον πρώτο παλμό ρολογιού:

Σχήμα 3.7.1 (β): Μεταφορά δεδομένου μεταξύ master και slave διάταξης



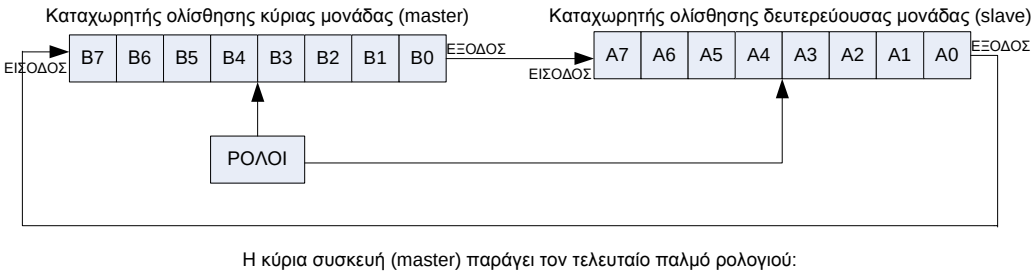
Η κύρια συσκευή (master) παράγει τον δεύτερο παλμό ρολογιού:

Σχήμα 3.7.1 (γ): Μεταφορά δεδομένου μεταξύ master και slave διάταξης



Η κύρια συσκευή (master) παράγει τον έβδομο παλμό ρολογιού:

Σχήμα 3.7.1 (δ): Μεταφορά δεδομένου μεταξύ master και slave διάταξης



Η κύρια συσκευή (master) παράγει τον τελευταίο παλμό ρολογιού:

Σχήμα 3.7.1 (ε): Μεταφορά δεδομένου μεταξύ master και slave διάταξης

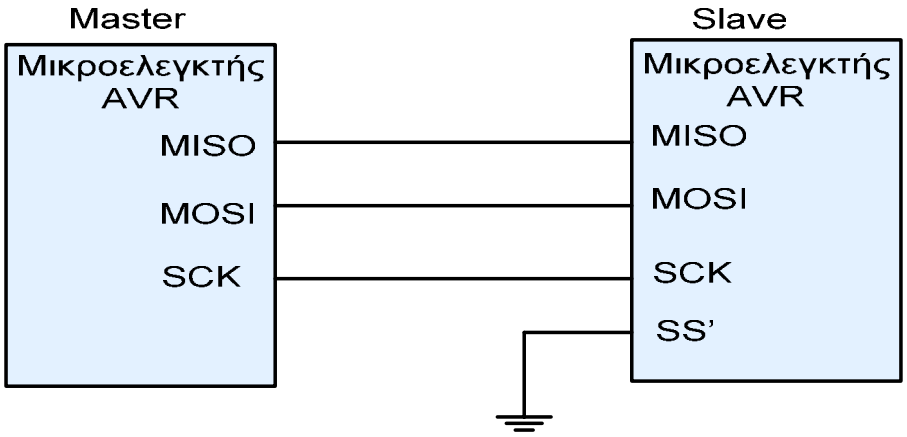
3.5 Περιγραφή διαύλου

Η επικοινωνία μέσω της μονάδας SPI προϋποθέτει τη συμφωνία κύριας και δευτερεύουσας διάταξης στις ρυθμίσεις σήματος χρονισμού. Τέσσερα σήματα (ακροδέκτες) χρειάζονται για την επικοινωνία: MISO, MOSI, SCK και SS'. Η λειτουργία κάθε σήματος είναι η εξής:

- MISO (Master In Slave Out): η είσοδος του κατ/τη ολίσθησης της κύριας διάταξης και η έξοδος του κατ/τη ολίσθησης της δευτερεύουσας διάταξης.
- MOSI (Master Out Slave In):): η έξοδος του κατ/τη ολίσθησης της κύριας διάταξης και η είσοδος του κατ/τη ολίσθησης της δευτερεύουσας διάταξης.
- SCK (Serial Clock): Στην κύρια διάταξη, το σήμα αυτό είναι η έξοδος της γεννήτριας χρονισμού. Στην δευτερεύουσα πρόκειται για την είσοδο του σήματος χρονισμού.
- SS' (Slave Select): Σε μια εγκατάσταση SPI μπορούν να συνδεθούν πολλές δευτερεύουσες διατάξεις. Το σήμα SS’ επιλέγει με ποια θα επικοινωνήσουμε. Όταν είναι σε υψηλή κατάσταση, όλοι οι ακροδέκτες των δευτερευουσών διατάξεων είναι είσοδοι και προφανώς δεν θα λάβουν δεδομένα μέσω της μονάδας SPI. Όταν είναι σε χαμηλή κατάσταση η μονάδα SPI ενεργοποιείται. Το λογισμικό της κύριας ελέγχει τη γραμμή SS’ κάθε δευτερεύουσας. Στη κύρια διάταξη, η συμπεριφορά του ακροδέκτη SS’ εξαρτάται από τη κατεύθυνση δεδομένων του ακροδέκτη. Αν ο SS’ ρυθμιστεί ως έξοδος, τότε δεν επιδρά στη μονάδα SPI. Αν ο SS’ ρυθμιστεί ως είσοδος, τότε πρέπει να διατηρηθεί σε υψηλή κατάσταση για να επιτύχουμε κύρια(master) λειτουργία. Σε χαμηλή κατάσταση το σύστημα ερμηνεύει το γεγονός ότι κάποια άλλη κύρια διάταξη χρησιμοποιεί την SPI για αποστολή δεδομένων σε

δευτερεύουσα. Αυτή η κατάσταση (δηλαδή να υπάρχουν δυο SPI masters) είναι ασυνήθιστη και δεν θα μας απασχολήσει. Αυτό που μας ενδιαφέρει είναι να ρυθμίζουμε τον ακροδέκτη SS' της κύριας διάταξης ως έξοδο.

Τα παρακάτω σχήματα δείχνουν μια τυπική εγκατάσταση μονάδας SPI:



Σχήμα 3.8: Εγκατάσταση μονάδας SPI με μια δευτερεύουσα διάταξη

3.6 Χρονιστής

Υπάρχουν διάφοροι τύποι χρονιστών διαθέσιμοι στους μικροελεγκτές (ενώ μερικά μοντέλα δεν διαθέτουν καθόλου χρονιστή). Οι χρονιστές βασικά μετρούν παλμούς ρολογιού. Οι AVR διαθέτουν έναν απλό χρονιστή των 8-bits (timer/counter0) και έναν πολύπλοκο χρονιστή των 16-bits (timer/counter1) με τέσσερις τρόπους λειτουργίας. Ο Χρονιστής δύναται να λειτουργεί ως Χρονιστής ή ως μετρητής. Ως μετρητής δέχεται στην είσοδο του παλμούς ενός εξωτερικού σήματος, ενώ το ρολόι του χρονιστή μπορεί να είναι ίσο με το κεντρικό ρολόι τους συστήματος (από τον κρύσταλλο) ή να επιβραδυνθεί με τον διαιρέτη συχνότητας (prescaler). Ο prescaler υπάρχει στον timer0 , αλλά και στον timer1. Με τον prescaler επιτυγχάνουμε μεγαλύτερες τιμές μέτρησης, αλλά μικρότερη ακρίβεια. Ο prescaler παίρνει τις τιμές 8, 64, 256 ή 1024 σε σύγκριση με το ρολόι του συστήματος. Ένας AVR στα 8 MHz και ένας Χρονιστής prescaler μπορεί να μετρήσει (όταν έχει 16-bit timer) : (0xFFFF + 1) * 1024 clock cycles = 67108864 clock cycles δηλαδή 8.388608 seconds.

3.6.1 Καταχωρητής μάσκας διακοπών χρονιστή (Timer Interrupt Mask Register -TIMSK)

Ο ρόλος του κατ/τη αυτού είναι να καθορίζει ποιες διακοπές είναι έγκυρες θέτοντας τα αντίστοιχα bits του.

Bit	7	6	5	4	3	2	1	0
Bits μάσκας	TOIE 1	OCIE1 A			TICIE 1		TOIE 0	

Η σημασία των bits του καταχωρητή είναι η εξής:

- Bit 7 (TOIE1- Timer Overflow Interrupt Enable/Timer 1): αν το bit αυτό έχει τεθεί και οι καθολικές διακοπές είναι ενεργοποιημένες, τότε εκτελείται άλμα στο διάνυσμα διακοπής υπερχείλισης του 16-bit χρονιστή timer1.
- Bit 6 (OCIE1A - Output Compare Interrupt Enable 1A): αν το bit αυτό έχει τεθεί και οι καθολικές διακοπές είναι ενεργοποιημένες, τότε εκτελείται άλμα στο διάνυσμα διακοπής του συγκριτή A
- Bit 3 (TICIE1 - Timer 1 Input Capture Interrupt Enable): αν το bit αυτό έχει τεθεί και οι καθολικές διακοπές είναι ενεργοποιημένες, τότε εκτελείται άλμα στο διάνυσμα διακοπής εισόδου σύλληψης
- Bit 1 (TOIE0 - Timer Overflow Interrupt Enable/Timer 0): ομοίως με TOIE1, αλλά για τον 8-bit χρονιστή.

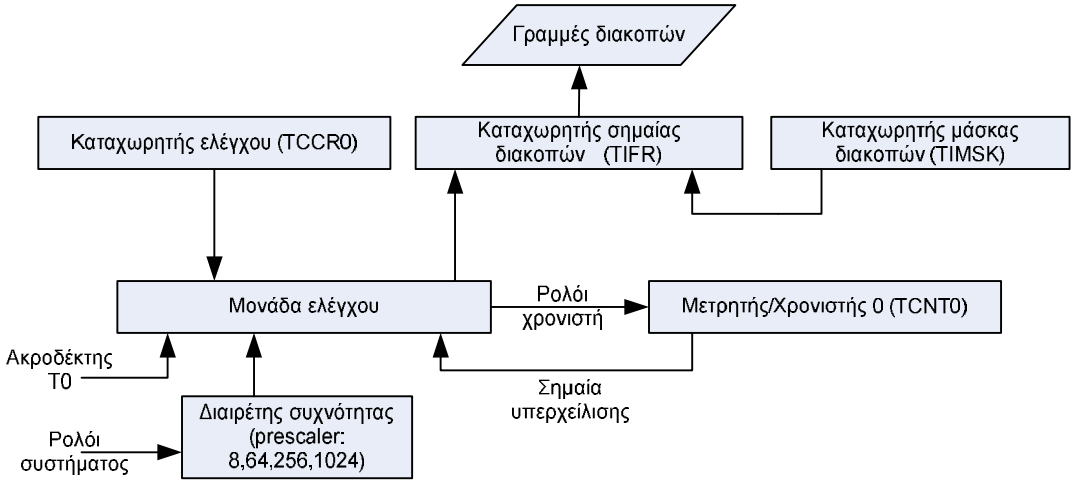
3.6.2 Καταχωρητής σημαίας διακοπών χρονιστή (Timer Interrupt Flag Register -TIFR)

Ο Καταχωρητής αυτός μας ενημερώνει ποιες διακοπές εκκρεμούν. Διατηρεί τις σημαίες διακοπής του χρονιστή που αντιστοιχούν στα bits του TIMSK. Αν μια διακοπή δεν έχει ενεργοποιηθεί ελέγχουμε τον TIFR για να καθορίσουμε εάν συνέβη και να μηδενίσουμε τις σημαίες.

Bit	7	6	5	4	3	2	1	0
Bits σημαίας	TOV1	OCF1A			ICF1		TOV0	

3.6.3 Ο 8-bit Χρονιστής (Timer0):

Η είσοδος χρονισμού του χρονιστή των 8-bits συνδέεται είτε με το κεντρικό ρολόι, είτε με ένα σήμα υποπολλαπλάσια συχνότητας του κεντρικού ρολογιού, είτε με κάποιο εξωτερικό σήμα σε κάποιον ακροδέκτη εισόδου-εξόδου (T0 Pin).Ακόμη, έχουμε τη δυνατότητα να σταματήσουμε τη λειτουργία του χρονιστή, τοποθετώντας τα σωστά bits στον κατ/τη ελέγχου TCCR0. Ο Χρονιστής αποτελεί βασικά έναν μετρητή προς τα πάνω. Παρακάτω φαίνεται ένα διάγραμμα με το υλικό του χρονιστή:



Σχήμα 3.9: 8-bit Χρονιστής και hardware

Η λειτουργία του είναι απλή: το ρολόι χρονισμού μετρά προς τα πάνω μέσω του κατ/τη Timer/Counter Register (TCNT0). Όταν αυτός υπερχειλίσει (από την τιμή 0xFF -> στην 0x00) τίθεται η σημαία υπερχείλισης του κατ/τη κατάστασης (Overflow Flag) και η σημαία διακοπής λόγω υπερχείλισης του χρονιστή. Αν το bit TOIE0 στον κατ/τη μάσκας διακοπών του χρονιστή TIMSK (Timer Interrupt Mask Register) είναι ‘1’ και οι καθολικές διακοπές είναι ενεργοποιημένες, τότε θα έχουμε άλμα στο αντίστοιχο διάνυσμα διακοπής.

3.6.4 Καταχωρητής ελέγχου του TIMER/COUNTER0 (TCCR0)

Αυτός ο Καταχωρητής ελέγχει τη λειτουργία του εσωτερικού μετρητή/χρονιστή, ο οποίος μετρά μέχρι την τιμή που του έχει φορτωθεί. Κάθε βήμα μέτρησης αντιστοιχεί σε ένα παλμό του σήματος διέγερσης. Οι πηγές χρονισμού φαίνονται στον πίνακα:

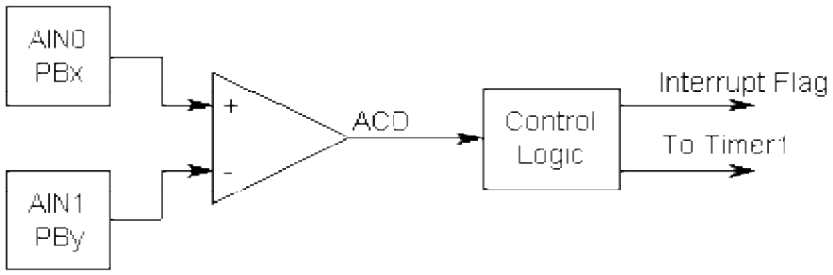
Πίνακας: Πηγές χρονισμού του TIMER/COUNTER0

CS02	**CS01**	**CS00**	**Σχόλιο**
0	0	0	Διακοπή λειτουργίας του χρονιστή
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	Αρνητικό μέτωπο παλμού στον ακροδέκτη T0
1	1	1	Θετικό μέτωπο παλμού στον ακροδέκτη T0

Bit	7	6	5	4	3	2	1	0
Bits ελέγχου						CS02	CS01	CS00

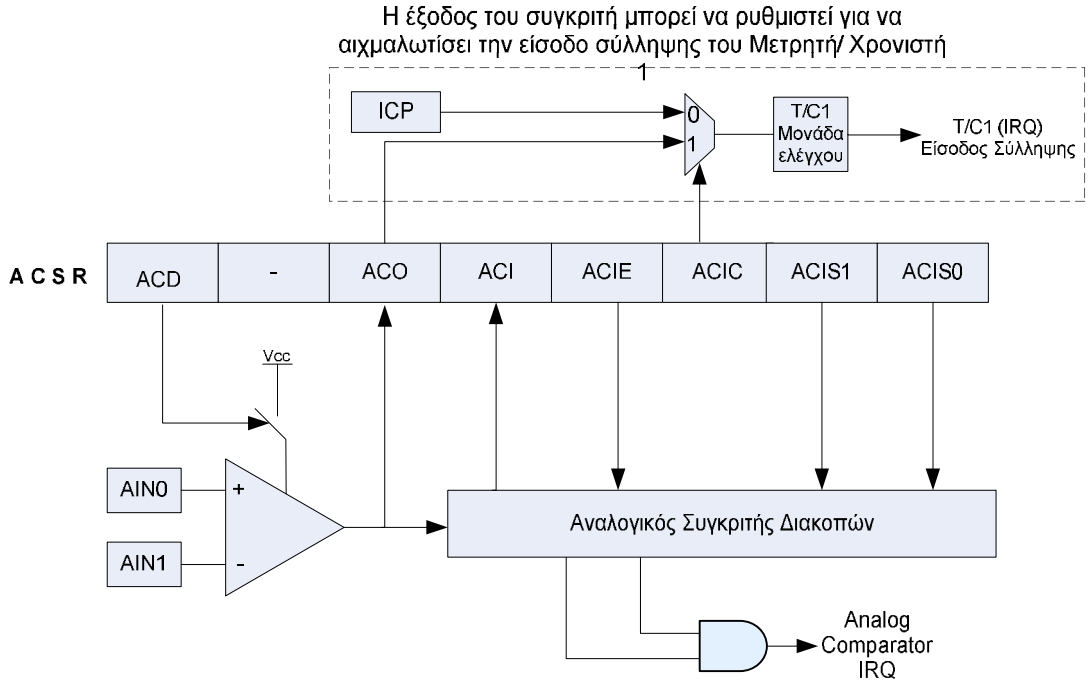
3.7 Αναλογικός συγκριτής

Ο αναλογικός συγκριτής είναι ένα χρήσιμο περιφερειακό για τη σύγκριση δύο αναλογικών σημάτων. Παραδείγματος χάριν, μπορούμε να συγκρίνουμε την έξοδο ενός αισθητήρα θερμοκρασίας με μια τάση αναφοράς, και να εκτελέσουμε κάποια λειτουργία όταν η θερμοκρασία υπερβαίνει το όριο που αντιστοιχεί στην τάση αυτή. Ο αναλογικός συγκριτής χωρίζεται σε 2 τμήματα. Το πρώτο είναι ο ίδιος ο συγκριτής, ο οποίος έχει δυο εισόδους: αναλογική είσοδο 0 (Analog Input 0 - AIN0) και αναλογική είσοδο 1 (Analog Input 1 - AIN0). Αν η είσοδος AIN0 είναι μεγαλύτερη από την AIN1, τότε η έξοδος του συγκριτή είναι υψηλή. Ενώ όταν η είσοδος AIN1 είναι μεγαλύτερη από την AIN0, τότε η έξοδος του συγκριτή είναι χαμηλή. Το δεύτερο τμήμα παίρνει την έξοδο του συγκριτή και θέτει τη σημαία διακοπής (Analog Comparator Interrupt Flag - ACI) και τη σημαία εξόδου του συγκριτή (Analog Comparator Output Flag - ACO). Ένα απλό σχέδιο του συγκριτή είναι:



Σχήμα 3.10: Σχέδιο συγκριτή

Πιο αναλυτικά, όπως θα δούμε στο παρακάτω σχήμα, οι είσοδοι AIN0 και AIN1 αντιστοιχούν στους ακροδέκτες μιας θύρας (π.χ. PORTB), οπότε ο Καταχωρητής κατεύθυνσης δεδομένων πρέπει να τεθεί κατάλληλα. Για το λόγο αυτό μηδενίζουμε τους κατ/τες κατεύθυνσης DDBx και DDBy ώστε οι ακροδέκτες να γίνουν είσοδοι. Ακόμη, μηδενίζουμε τους κατ/τες PORTx και PORTy ώστε να απενεργοποιήσουμε τις αντιστάσεις πρόσδεσης.



Σχήμα 3.11:Αναλογικός συγκριτής

Ο συγκριτής είναι σχετικά απλός και διαθέτει μόνο έναν καταχωρητή:

3.8 Η Μνήμη EEPROM

Οι μικροελεγκτές AVR διαθέτουν ενσωματωμένη μνήμη EEPROM. Το μέγεθος της κυμαίνεται από 64 bytes (στα μοντέλα AT Tiny10,AT Tiny12 και AT90S1200) έως 4096 bytes (στο μοντέλο AT Mega103). Υπάρχουν τρεις υπεύθυνοι καταχωρητές για την πρόσβαση στη μνήμη EEPROM. Αυτοί είναι: ο Καταχωρητής δεδομένων (EEDR), ο Καταχωρητής διευθύνσεων (EEAR) και ο Καταχωρητής ελέγχου.

(EECR). Σε μικροελεγκτές με μνήμη άνω των 256 bytes,ο Καταχωρητής διευθύνσεων χωρίζεται στους EEARH και EEARL. Η λειτουργία τους εξηγείται παρακάτω:

3.8.1 Καταχωρητής διευθύνσεων της μνήμης EEPROM (EEAR)

Ο Καταχωρητής αυτός περιέχει τη διεύθυνση της μνήμης EEPROM. Σε περίπτωση επανεκκίνησης ο Καταχωρητής διατηρεί τα δεδομένα του.

3.8.2 Καταχωρητής δεδομένων της μνήμης EEPROM (EEDR)

Για τη διαδικασία εγγραφής στη μνήμη EEPROM, ο Καταχωρητής αυτός περιέχει τα δεδομένα που πρέπει να γραφτούν στη διεύθυνση που δίνεται από τον καταχωρητή EEAR. Για τη διαδικασία ανάγνωσης από τη μνήμη EEPROM, περιέχει τα δεδομένα που διαβάστηκαν από τη διεύθυνση που δίνεται από τον καταχωρητή EEAR. Έχει μήκος 8-bits.

3.8.3Καταχωρητής ελέγχου της μνήμης EEPROM (EECR)

Ο Καταχωρητής αυτός ελέγχει τις διαδικασίες εγγραφής και ανάγνωσης στην EEPROM. Η σημασία των bits ελέγχου είναι η εξής:

- Bit 3 (EERIE- EEPROM Ready Interrupt Enable): Όταν το I-bit του SREG και το EERIE είναι ‘1’, τότε οι διακοπές της μνήμης είναι ενεργοποιημένες για την περίπτωση που το bit EEWE μηδενίζεται. Διαφορετικά απενεργοποιούνται.
- Bit 2 (EEMWE-EEPROM Master Write Enable): Όταν το bit αυτό τίθεται ίσο με ‘1’ και εφόσον θέσουμε και το bit EEWE τότε θα εγγραφούν δεδομένα στη EEPROM. Μετά από 4 περιόδους ρολογιού το υλικό μηδενίζει το bit αυτό.
- Bit 1 (EEME- EEPROM Write Enable): Η διαδικασία εγγραφής ενεργοποιείται, όταν αυτό το bit τίθεται ίσο με ‘1’ και εφόσον έχουμε θέσει και το bit EEMWE. Το περιεχόμενο του καταχωρητή EEDR εγγράφεται στη διεύθυνση της μνήμης που δείχνει ο EEAR. Η διαδικασία εγγραφής στην EEPROM είναι:

    i. αναμονή ωσότου το bit EEWE μηδενιστεί
    ii. εγγραφή της διεύθυνσης των δεδομένων στον EEAR
    iii. εγγραφή των δεδομένων στον EEDR
    iv. θέτουμε το bit EEMWE του EECR
    v. μετά από 4 περιόδους ρολογιού θέτουμε και το bit EEWE

- Bit 0 (EERE: EEPROM Read Enable): Η διαδικασία ανάγνωσης ενεργοποιείται, όταν αυτό το bit τίθεται ίσο με ‘1’. Συγκεκριμένα, ελέγχουμε αρχικά το bit EEWE ώστε να μην βρίσκεται διαδικασία εγγραφής σε εξέλιξη, τοποθετούμε την επιθυμητή διεύθυνση στον καταχωρητή EEAR και θέτουμε

το bit EERE. Έπειτα το υλικό μηδενίζει το bit EERE και τα δεδομένα μεταφέρονται αυτόματα στον καταχωρητή EEDR.

Bits	7	6	5	4	3	2	1	0
ελέγχου					EE RIE	EEM WE	EE WE	EE RE

ΚΕΦΑΛΑΙΟ 4^{Ο} - Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C

4.1 ΓΕΝΙΚΑ ΓΙΑ ΤΗΝ C

Η **C** (προφέρεται "*σι*") είναι μια διαδικαστική γλώσσα προγραμματισμού γενικής χρήσης, η οποία αναπτύχθηκε αρχικά, μεταξύ του 1969 και του 1973, από τον Ντενίς Ρίτσι στα εργαστήρια AT&T Bell Labs για να χρησιμοποιηθεί για την ανάπτυξη του λειτουργικού συστήματος UNIX. Όπως οι περισσότερες διαδικαστικές γλώσσες προγραμματισμού που ακολουθούν την παράδοση της ALGOL, η C έχει δυνατότητες δομημένου προγραμματισμού και επιτρέπει τη χρήση αναδρομής (αλλά όχι και εμφωλευμένων συναρτήσεων), ενώ, ο στατικός ορισμός του τύπου των μεταβλητών που επιβάλλει, προλαμβάνει πολλά σφάλματα κατά την χρήση τους. Ο σχεδιασμός της περιλαμβάνει δομές που μεταφράζονται αποδοτικά σε τυπικές εντολές μηχανής (machine Instruction) και εξ αιτίας αυτού χρησιμοποιείται συχνά σε εφαρμογές που παλιότερα γράφονταν σε συμβολική γλώσσα(assembly language). Αυτό ακριβώς το χαρακτηριστικό της, που έχει σαν συνέπεια και την αυξημένη ταχύτητα εκτέλεσης των εφαρμογών που γράφονται σε αυτή, καθώς και το γεγονός ότι είναι διαθέσιμη στα περισσότερα σημερινά λειτουργικά συστήματα, συνέβαλε κατά πολύ στην καθιέρωση της και την χρήση της για ανάπτυξη λειτουργικών συστημάτων και λοιπών προγραμμάτων συστήματος (system software), αλλά και απλών εφαρμογών.

Η C συγκαταλέγεται πλέον στις πιο ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού όλων των εποχών και πολλές νεώτερες γλώσσες έχουν επηρεαστεί άμεσα ή έμμεσα από αυτήν, συμπεριλαμβανομένων των C++, C#, D, GO, Java, JavaScript, Limbo, LPC, Perl, PHP, Python, καθώς και του κελύφους C(C shell) του Unix. Κάποιες από αυτές τις γλώσσες έχουν επηρεαστεί κυρίως στη σύνταξη τους, με το σύστημα τύπων, τα μοντέλα δεδομένων και το νόημα των εκφράσεων τους να διαφέρουν σημαντικά από την C. Η C++, ειδικά, ξεκίνησε σαν προεπεξεργαστής της C, αλλά έχει εξελιχθεί πλέον σε μια αντικειμενοστραφή γλώσσα, που αποτελεί υπερσύνολο της C.

4.2 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Υπάρχουν οι ακόλουθες γλώσσες προγραμματισμού στην αγορά σήμερα, οι οποίες παρατίθενται με αλφαβητική σειρά:

- ActionScript
- Ada
- Algol
- APL
- AWK
- BASIC
- C
- C++
- C#
- Cilk
- Clojure
- COBOL
- Datalog
- Eiffel
- Erlang
- Forth
- FORTRAN
- Haskell
- Java
- JavaScript
- Lisp
- Logo
- Lua
- Lucid
- Mathematica
- Matlab
- Miranda
- ML
- Modula-2
- OBJ / Σύστημα Maude
- Objective-C
- OCaml
- Pascal
- Perl
- PHP
- PostScript
- Prolog
- Python
- RPG
- Ruby
- Scala
- Scheme
- Simula
- Smalltalk
- SQL
- Tcl
- Visual Basic

4.3 ΜΙΑ ΣΥΓΚΡΙΣΗ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΑΤΙΣΜΟΥ

Η C θεωρείται γενικά γλώσσα μέσου επιπέδου και αυτό γιατί συνδυάζει στοιχεία των γλωσσών υψηλού επιπέδου (high level languages), όπως είναι η Cobol και η Pascal και στοιχεία των γλωσσών χαμηλού επιπέδου (low level languages), όπως είναι η Assembly.

Για να διευκρινίσουμε, πρέπει να πούμε ότι σαν **γλώσσα υψηλού επιπέδου** θεωρείται η γλώσσα εκείνη που είναι αρκετά περιγραφική και που είναι έτσι πιο κοντά στην ανθρώπινη γραπτή γλώσσα και σαν **γλώσσα χαμηλού επιπέδου** θεωρείται εκείνη που

είναι πιο κοντά στη μηχανή. Αυτός ο χαρακτηρισμός δεν έχει καμία απολύτως σχέση με τις δυνατότητες της γλώσσας.

Σίγουρα, η κάθε γλώσσα προγραμματισμού έχει κάποιες προτεραιότητες να εκπληρώσει. Η **Pascal**, για παράδειγμα, χρησιμοποιείται κυρίως για τη σωστή διδασκαλία των αρχών του προγραμματισμού, ενώ η ***Basic*** δημιουργήθηκε έτσι ώστε να δώσει τη δυνατότητα σε αρχάριους στον προγραμματισμό να κάνουν με άνεση και ευκολία τα πρώτα τους βήματα στον ιδιόμορφο αυτό χώρο. Ο Clipper και η Cobol είναι καθαρά επαγγελματικές γλώσσες προγραμματισμού.

Η C, όμως, μπόρεσε να φέρει τον προγραμματιστή *πιο κοντά στο* **hardware**, που με τις άλλες γνωστές γλώσσες προγραμματισμού κάτι τέτοιο θα ήταν πολύ δύσκολο να γίνει. Βέβαια, η κάθε γλώσσα προγραμματισμού κάνει και διαφορετική δουλειά και δεν θα ήταν σωστό να κάνουμε συγκρίσεις, για τον ίδιο λόγο που δεν μπορούμε να συγκρίνουμε ένα αεροπλάνο μ' ένα ποδήλατο καθώς το καθένα είναι προορισμένο να κάνει διαφορετική δουλειά.

4.4 Πλεονεκτήματα της Γλώσσας Προγραμματισμού C

Τα τελευταία χρόνια η C έχει καθιερωθεί ως μια από τις σημαντικότερες και δημοφιλέστερες γλώσσες προγραμματισμού. Τα σημαντικότερα πλεονεκτήματα που εξηγούν αυτήν την προτίμησή της, αναφέρονται παρακάτω.

- *Χαρακτηριστικά Σχεδίασης*, Η C έχει μοντέρνες δομές ελέγχου για να μπορούμε να κάνουμε επαναληπτικές εργασίες και για εύκολη επιλογή εναλλακτικών τρόπων δράσης. Με το πλήθος των δομών δεδομένων που διαθέτει, μπορεί να αναπαραστήσει ένα μεγάλο σύνολο από διαφορετικούς τύπους πληροφοριών. Έχει και το μεγάλο πλεονέκτημα ότι επιβάλλει τη διάσπαση του προγράμματος σε αυτοδύναμες ενότητες, τις συναρτήσεις.
- *Αποτελεσματική*, Η C είναι μια αποτελεσματική γλώσσα προγραμματισμού, που είναι τόσο συμπεριεκτική, ώστε να χρησιμοποιούμε σ' αυτήν πολύ λιγότερες λέξεις σε σχέση με άλλες γλώσσες. Έχει έναν συμπαγή και γρήγορο κώδικα.
- *Φορητή Γλώσσα*, Η C είναι μια φορητή γλώσσα, δηλ. τα προγράμματά της μπορούν να τρέξουν με λίγες ή και με καθόλου τροποποιήσεις και σε ένα άλλο σύστημα. Μάλιστα θεωρείται σαν η πιο φορητή γλώσσα.
- *Δυναμικότητα και Ευελιξία*, Η C είναι δυναμική και ευέλικτη, δύο ιδιότητες που είναι αρκετά δημοφιλείς στους υπολογιστές. Όπως ξέρουμε, το μεγαλύτερο μέρος του δυναμικού και ευέλικτου λειτουργικού συστήματος Unix είναι γραμμένο σε C. Αυτό ισχύει και για επεξεργαστές κειμένων,

μεταγλωττιστές (compiler) και ερμηνευτές (interpreters) γλωσσών προγραμματισμού. Η C διαθέτει μερικά από τα χαρακτηριστικά ελέγχου που συνήθως τα συναντάμε στη συμβολική γλώσσα (assembly language).

- ü *Προσανατολισμός προς τον Προγραμματιστή*, Η C είναι προσανατολισμένη προς τις ανάγκες του προγραμματιστή, ο οποίος και έχει άμεση πρόσβαση στο υλικό. Με τη C έχουμε τη σπουδαία δυνατότητα να μπορούμε να χειριζόμαστε μεμονωμένα τα *δυαδικά ψηφία* ***(bits)*** της μνήμης. Γενικά η C είναι πολύ λιγότερο περιοριστική στο να μας αφήνει να κάνουμε ό,τι θέλουμε σε σχέση με την Pascal για παράδειγμα.

Αυτή η ελευθερία είναι και πλεονέκτημα, αλλά είναι και επικίνδυνη όπως είναι φυσικό. Στη C τα πάντα (σχεδόν) επιτρέπονται. Δεν γίνεται έλεγχος των τύπων, άρα μπορεί κανείς να ανακατέψει ό,τι δεδομένα θέλει, κάτι που είναι πολύ χρήσιμο όταν προγραμματίζουμε σε επίπεδο συστήματος. Ακόμη, η C έχει μια τεράστια βιβλιοθήκη από χρήσιμες συναρτήσεις.

4.5 Τα Μειονεκτήματα της C

Η C έχει και μειονεκτήματα, γιατί όπως πολύ καλά ξέρουμε η πολύ ελευθερία βλάπτει. Για παράδειγμα, η *ελευθερία έκφρασης* που αναφέραμε παραπάνω ότι έχει η C, απαιτεί από τον προγραμματιστή μια αυξημένη επαγρύπνηση και υπευθυνότητα. Ακόμη, η λακωνικότητα της C σε συνδυασμό με τον πλούτο των τελεστών που έχει, έχει σαν αποτέλεσμα τη δημιουργία προγραμμάτων που είναι τόσο *δυσανάγνωστα*, ώστε να είναι δύσκολο να τα κατανοήσει κάποιος με την πρώτη ματιά και πολλές φορές ακόμα και αυτός που τα έγραψε. Επιπλέον, συχνά είναι πολύ δύσκολο να ανιχνευθούν και τα λογικά λάθη σ' ένα πρόγραμμα της C. Η C έχει τελικά τόσες πολλές δυνατότητες έκφρασης, ώστε να χρειαστεί πολύς καιρός για να μπορεί να πει κανείς με βεβαιότητα ότι την έμαθε καλά.

4.6 Αντικατάσταση της Assembly από τη C

Όπως ίσως ξέρουμε, η γλώσσα Assembly χρησιμοποιεί μια συμβολική αναπαράσταση του πραγματικού δυαδικού κώδικα που εκτελεί απευθείας ο υπολογιστής και η κάθε εντολή της Assembly αντιστοιχεί σε μία μόνο ενέργεια που θα πρέπει να εκτελέσει ο υπολογιστής.

Η γλώσσα Assembly είναι, όμως, δύσκολη στο γράψιμο προγραμμάτων και επειδή δεν είναι καθόλου δομημένη, κάνει το τελικό πρόγραμμα να είναι ανεπιθύμητο από τους προγραμματιστές. Ο ερχομός της C κάνει περιττή πλέον τη χρήση της Assembly.

Έτσι, με τη C μπορούμε να γράψουμε προγράμματα που να έχουν την ίδια αποτελεσματικότητα με άλλα ανάλογα προγράμματα που είναι γραμμένα στη γλώσσα Assembly, αλλά να έχουμε συγχρόνως και τα σπουδαία πλεονεκτήματα μιας γλώσσας προγραμματισμού ανωτέρου επιπέδου.

4.7 Δομή - Γράψιμο ενός προγράμματος στη γλώσσα C

Όπως είπαμε στα προηγούμενα, η C επιβάλλει τον καταμερισμό του προγράμματος σε ενότητες, που ονομάζονται *συναρτήσεις (functions)*. Εάν είναι απαραίτητο, οι συναρτήσεις μπορούν να χωριστούν και σε μικρότερες συναρτήσεις. Επίσης, στη C το κύριο πρόγραμμα είναι κι αυτό μια συνάρτηση, που ονομάζεται *main()*. Μια μέθοδος για το γράψιμο ενός προγράμματος στη C είναι να ξεκινήσουμε γράφοντας τη συνάρτηση main(), την ενότητα του πιο πάνω επιπέδου και μετά να ασχοληθούμε με τις συναρτήσεις των πιο κάτω επιπέδων. Η διαδικασία αυτή ονομάζεται *πάνω-προς-τα-κάτω προγραμματισμός (top-down programming)*.

Η αντίστροφη διαδικασία, δηλαδή το να ασχοληθούμε πρώτα με τις συναρτήσεις των κατώτερων επιπέδων και μετά να ανεβαίνουμε προς τα πάνω, ονομάζεται *κάτω-προς-τα-πάνω προγραμματισμός (bottom-up programming)*. Ένα πλεονέκτημα του πάνω-προς-τα-κάτω προγραμματισμού είναι ότι μπορούμε να χαράξουμε καλύτερα τη ροή του προγράμματος, μια και δεν ασχολούμαστε από την αρχή με τις λεπτομέρειες των επί μέρους συναρτήσεων.

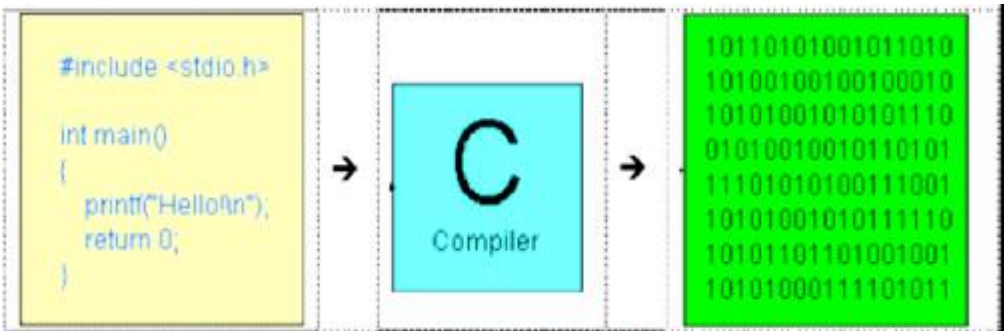
Για τη μεταγλώττιση και σύνδεση ενός προγράμματος στη γλώσσα C, ακολουθείται η εξής διαδικασία: ο *μεταγλωττιστής (compiler)* της C μετατρέπει τον *πηγαίο κώδικα (source program)*, δηλαδή το πρόγραμμα που γράφουμε σε C, σ' έναν

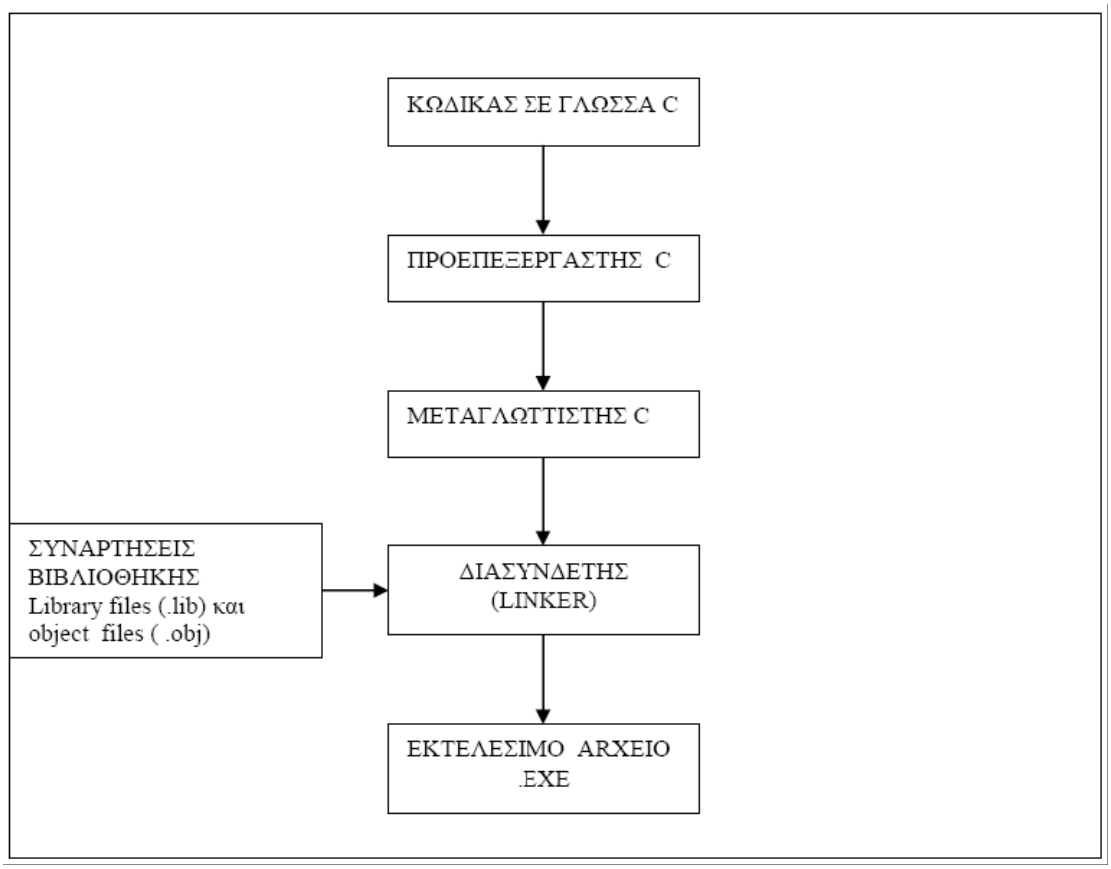
αντικειμενικό κώδικα ***(object program)*** και το *πρόγραμμα σύνδεσης* ***(linker)*** συνδυάζει αυτόν τον κώδικα με άλλους κώδικες και δημιουργείται έτσι το *εκτελέσιμο αρχείο* ***(executable file)***. Τα προγράμματα της C έχουν την επέκταση ***.c***.

Ο ρόλος του προγράμματος σύνδεσης είναι να ενώσει τον τελικό κώδικα, τον *κώδικα εκκίνησης* ***(start-up code)*** του συστήματός μας και τον *κώδικα βιβλιοθήκης* ***(library code)*** στο εκτελέσιμο αρχείο. Ο κώδικας εκκίνησης έχει σχέση με την επικοινωνία μεταξύ του προγράμματος και του λειτουργικού συστήματος και ο κώδικας βιβλιοθήκης περιέχει τον τελικό κώδικα για πολλές συναρτήσεις. Σε μερικά συστήματα πρέπει να τρέξουμε τα προγράμματα μεταγλώττισης και σύνδεσης ξεχωριστά, ενώ σε άλλα ο μεταγλωττιστής ενεργοποιεί το πρόγραμμα σύνδεσης αυτόματα μόνος του.

Η γλώσσα προγραμματισμού "C" είναι σχεδιασμένη για να τρέχει κάτω από όλα τα λειτουργικά συστήματα, και έτσι δεν βασίζεται σε κάποιες ειδικές ευκολίες που μπορεί να παρέχει το λειτουργικό σύστημα. Για να γράψουμε έτσι ένα πρόγραμμα C απαιτούνται:
1) ένας text editor στον οποίο θα γράψουμε το πρόγραμμά μας. Τέτοιοι editors υπάρχουν σε όλα τα λειτουργικά συστήματα,
2) ο μεταγλωττιστής της γλώσσας C που θα αναλάβει να μεταφράσει το πρόγραμμά μας σε κώδικα μηχανής. Ο μεταγλωττιστής της C αποτελείται από τον C preprocessor και τον C-compiler. Στα επόμενα με τον όρο "compiler" θα αναφερόμαστε και στα δύο αυτά τμήματα, εκτός αν καθορίζεται διαφορετικά. Ο C-preprocessor επεξεργάζεται τα αρχεία που περιέχουν κώδικα C, με τρόπο που θα αναλυθεί σε επόμενο κεφάλαιο, και παράγει ενδιάμεσα αρχεία, από τα οποία ο C-compiler παράγει αρχεία με κώδικα μηχανής (.obj) αρχεία και
3) ο linker, που είναι κομμάτι του λειτουργικού συστήματος, αν και κάποια υλοποίηση ενός C-compiler μπορεί να παρέχει τον δικό της. Ο linker αναλαμβάνει να συνδέσει τον κώδικα μηχανής του προγράμματος μας (αυτόν που βρίσκεται στα . obj αρχεία) με κώδικα που υπάρχει σε βιβλιοθήκες (αρχεία .lib) και άλλα αρχεία με κώδικα μηχανής (.obj), παράγοντας τελικά ένα εκτελέσιμο αρχείο. Ο κώδικας που βρίσκεται στις βιβλιοθήκες συμπεριλαμβάνει κώδικα που υλοποιεί υψηλού επιπέδου λειτουργίες μέσα από διαδικασίες πιο χαμηλού επιπέδου, απαλλάσσοντας μας έτσι από την υποχρέωση να κάνουμε εμείς την υλοποίηση αυτή.

Στα περισσότερα λειτουργικά συστήματα δεν είναι απαραίτητο να ασχοληθούμε με τη διαδικασία αυτή μια και υπάρχει κάποιο πρόγραμμα που την αναλαμβάνει . Σε μερικά μάλιστα συστήματα και ο text editor παρέχεται σε ένα ολοκληρωμένο περιβάλλον, στο οποίο μπορούμε να γράψουμε τα προγράμματά μας, να τα μεταφράσουμε και να τα συνδέσουμε παράγοντας έτσι το εκτελέσιμο αρχείο χωρίς να χρειαστεί να ασχοληθούμε καθόλου με την ενδιάμεση διαδικασία. Η όλη διαδικασία παραγωγής ενός εκτελέσιμου αρχείου από τα c, .obj και .lib αρχεία φαίνεται στο πιο κάτω σχήμα:





ΣΥΝΤΟΜΗ ΑΝΑΛΥΣΗ ΚΥΚΛΩΜΑΤΟΣ

Η πρώτη γραμμή του προγράμματος χρησιμοποιεί τα σύμβολα /* και */ για να συμπεριλάβουμε εκεί κάποια *σχόλια (comments),* που θα μας βοηθήσουν να κάνουμε το πρόγραμμά μας πιο ευανάγνωστο. Αυτά τα σχόλια αγνοούνται από τον υπολογιστή. Η δεύτερη γραμμή λέει στον υπολογιστή να συμπεριλάβει (include) τις πληροφορίες που υπάρχουν στο αρχείο stdio.h, το οποίο αποτελεί μέρος του πακέτου της γλώσσας C.

Τα προγράμματα της C αποτελούνται από μία ή περισσότερες συναρτήσεις, που είναι και οι βασικές ενότητες ενός προγράμματος C. Αυτό το πρόγραμμα αποτελείται από μία μόνο συνάρτηση που καλείται ***main()***. Οι παρενθέσεις υποδηλώνουν ότι το main() είναι ένα όνομα μιας συνάρτησης.

Η αγκύλη { δηλώνει την αρχή των προτάσεων που αποτελούν τη συνάρτηση και ο ορισμός της συνάρτησης τελειώνει με την αντίστοιχη αγκύλη }. Οι αγκύλες { και } είναι αντίστοιχες με τα begin και end της Pascal. Η πρόταση δήλωσης μάς λέει ότι θα χρησιμοποιήσουμε μια μεταβλητή με το όνομα num και ότι η num είναι ακέραια μεταβλητή (integer). Η πρόταση καταχώρησης δίνει την τιμή 10 στη μεταβλητή num.

Η επόμενη γραμμή είναι για την εκτύπωση της φράσης που βρίσκεται μεταξύ των εισαγωγικών (" "). Το *\n* λέει στον υπολογιστή να ξεκινήσει μια νέα γραμμή (σαν να πατούσαμε το πλήκτρο <enter>). Η επόμενη γραμμή είναι για την εκτύπωση της τιμής της num (που είναι το 10) ανάμεσα στη φράση που βρίσκεται μεταξύ των " ". Η εντολή %d λέει στον υπολογιστή πού και με ποια μορφή να εκτυπώσει την τιμή της num. Το πρόγραμμα τελειώνει με την αγκύλη }.

ΛΕΠΤΟΜΕΡΗΣ ΑΝΑΛΥΣΗ ΚΥΚΛΩΜΑΤΟΣ

1. Τα Αρχεία Επικεφαλίδας #include

Το αρχείο stdio.h αποτελεί μέρος του μεταγλωττιστή της C και περιέχει πληροφορίες σχετικές με συναρτήσεις εισόδου και εξόδου, όπως είναι η ***printf()***, που

χρησιμοποιεί ο μεταγλωττιστής. Το όνομά του προέρχεται από τις λέξεις ***standard input/output header***. Οι χρήστες της C αναφέρουν σαν επικεφαλίδα μια συλλογή πληροφοριών που βρίσκεται στην αρχή ενός αρχείου. Το αποτέλεσμα της εντολής ***#include <stdio.h>*** θα ήταν το ίδιο με το να αντιγράφαμε όλο το περιεχόμενο του αρχείου stdio.h στο δικό μας αρχείο, στη θέση όπου εμφανίζεται αυτή η γραμμή προγράμματος.

Στην πραγματικότητα, αυτή η γραμμή προγράμματος δεν είναι καν μια πρόταση της γλώσσας C. Το σύμβολο # σημαίνει ότι τη γραμμή αυτή τη διαχειρίζεται ο *προεπεξεργαστής* ***(preprocessor)*** της C, ο οποίος διαχειρίζεται κάποιες εργασίες πριν από τον μεταγλωττιστή.

2. Η Συνάρτηση main()

Η εκτέλεση ενός προγράμματος σε C αρχίζει πάντα με μια συνάρτηση που αποκαλείται main(). Είμαστε ελεύθεροι να επιλέξουμε τα ονόματα των άλλων συναρτήσεων που ίσως χρησιμοποιήσουμε, αλλά θα πρέπει υποχρεωτικά να υπάρχει η συνάρτηση main() στην αρχή του προγράμματος. Οι παρενθέσεις δηλώνουν ότι η main() είναι μια συνάρτηση. Γενικά, οι παρενθέσεις περιέχουν πληροφορίες (ορίσματα) που θα περάσουν μέσα στη συνάρτηση. Όταν, βέβαια, δεν υπάρχουν πληροφορίες για να περάσουν (μεταβιβασθούν), τότε οι παρενθέσεις είναι άδειες.

3. Τα Σχόλια

Όταν στο πρόγραμμά μας έχουμε σχόλια, τότε είναι πολύ ευκολότερο σε κάποιον άλλον, αλλά ακόμα και σε μας, να καταλάβει τι κάνει το πρόγραμμά μας. Οτιδήποτε υπάρχει ανάμεσα στο σύμβολο ανοίγματος /* και στο σύμβολο κλεισίματος */ των σχολίων αγνοείται από τον υπολογιστή και τα σχόλια στη C μπορούν να τοποθετηθούν οπουδήποτε, ακόμα και σε πολλές συνεχόμενες γραμμές.

4. Αγκύλες, Σώματα, Μπλοκ

Οι αγκύλες { και } δηλώνουν την αρχή και το τέλος του σώματος μιας συνάρτησης και μπορούν ακόμη να χρησιμοποιηθούν για να συμπεριλάβουν μαζί προτάσεις μέσα σε μια ομάδα ή σ' ένα μπλοκ του προγράμματος, κάτι δηλαδή παρόμοιο με τα begin και end της Pascal.

5. Οι Προτάσεις Δήλωσης

Η πρόταση δήλωσης είναι μια από τις σημαντικότερες προτάσεις της C. Η δήλωση αυτή σημαίνει ότι κάπου μέσα στη συνάρτηση χρησιμοποιούμε τη μεταβλητή που δηλώνουμε και ότι ο τύπος της είναι αυτός που δείχνουμε, π.χ. ακέραιος. Η λέξη ***int*** είναι μια λέξη-κλειδί της C, δηλ. δεν μπορεί να χρησιμοποιηθεί αλλού μέσα στο πρόγραμμα σαν όνομα μιας συνάρτησης ή μιας μεταβλητής. Το ερωτηματικό στο τέλος της γραμμής δηλώνει ότι η γραμμή αυτή αποτελεί μια πρόταση ή εντολή της C.

4.8 Οι Τύποι Δεδομένων και οι Μεταβλητές

Η C έχει διάφορα είδη (τύπους) δεδομένων : ακέραιους, χαρακτήρες, κινητής υποδιαστολής, αριθμούς κ.ά. Για το όνομα μιας μεταβλητής μπορούμε να χρησιμοποιήσουμε μικρά γράμματα, κεφαλαία γράμματα, ψηφία αριθμών και τον χαρακτήρα της υπογράμμισης (_). Ο πρώτος χαρακτήρας, όμως, πρέπει να είναι πάντα γράμμα. Πρέπει να είναι όλες οι μεταβλητές του προγράμματος συγκεντρωμένες μαζί για να είναι ευκολότερο για τον αναγνώστη να καταλάβει τι κάνει το πρόγραμμα και ακόμη πρέπει να υπάρχουν και σχόλια δίπλα στην κάθε μεταβλητή που να εξηγούν την αποστολή της. Ακόμη, η ονομασία που δίνουμε στις μεταβλητές πρέπει να μας βοηθάει να καταλαβαίνουμε και τη χρήση τους.

4.9 Δεδομένα, σταθερές και μεταβλητές

4.9.1 Δεδομένα

Τα δεδομένα (πληροφορίες-data) είναι απαραίτητα στοιχεία ενός προγράμματος , καθώς οι βασικές λειτουργίες ενός προγράμματος είναι η επεξεργασία αυτών των δεδομένων και η εξαγωγή αποτελεσμάτων (δηλαδή άλλα δεδομένα). Απαιτείται λοιπόν η δέσμευση κάποιων χώρων μνήμης για να αποθηκευτούν αυτά τα δεδομένα κατά τη διάρκεια της εκτέλεσης του προγράμματος. Αυτοί οι χώροι μνήμης που τους χρησιμοποιούμε για τη φύλαξη δεδομένων, όταν εκτελείται ένα πρόγραμμα, ονομάζονται ανάλογα με τη χρήση τους σταθερές ή μεταβλητές και κάθε γλώσσα προγραμματισμού μας δίνει τη δυνατότητα με διάφορους τρόπους να δηλώσουμε (καθορίσουμε) κάποιους τύπους μεταβλητών / σταθερών, τις οποίες θα χρησιμοποιήσουμε για την αποθήκευση δεδομένων, όταν εκτελείται ένα πρόγραμμα.

4.9.2 Οι σταθερές

Αρκετά προγράμματα απαιτούν ορισμένα δεδομένα που δεν αλλάζουν ποτέ κατά τη διάρκεια της εκτέλεσής τους. Αυτά τα δεδομένα συνήθως καθορίζονται μια φορά και χρησιμοποιούνται όσο συχνά επιθυμούμε κατά την διάρκεια λειτουργίας του προγράμματος. Για παράδειγμα, όταν γράφετε ένα πρόγραμμα, μπορείτε να καθορίσετε ότι η τιμή του π είναι 3.14159 και να χρησιμοποιείτε αυτή την τιμή όταν την χρειάζεστε, ξέροντας ότι είναι διαθέσιμη και σωστή. Οι σταθερές ορίζονται με εντολές του προεπεξεργαστής με την οδηγία # define και συνήθως χρησιμοποιούμε κεφαλαία γράμματα για να τις περιγράψουμε: # define PI 3.14159 (σημειώνουμε ότι αυτή η εντολή δεν τελειώνει με το ερωτηματικό ";")

4.9.3 Οι μεταβλητές

Αντίθετα με τις σταθερές, οι τιμές των μεταβλητών είναι δυνατόν να αλλάξουν κατά τη διάρκεια της εκτέλεσης ενός προγράμματος. Οι γλώσσες προγραμματισμού μας δίνουν τη δυνατότητα να καθορίσουμε μεταβλητές και στη συνέχεια να τους δώσουμε όποια τιμή επιθυμούμε, που έχει βέβαια σχέση με το πρόγραμμα. Η δήλωση των μεταβλητών γίνεται συνήθως στην αρχή του προγράμματος.

4.10 Έλεγχος της Ορθότητας των Προγραμμάτων

1. Συντακτικά Λάθη

Το συντακτικό λάθος στη C είναι κάτι ανάλογο με το γραμματικό λάθος στη γλώσσα που μιλάμε. Συντακτικά λάθη στη C μπορούν να γίνουν και με τη χρήση επιτρεπτών συμβόλων σε λανθασμένες θέσεις. Παραδείγματα τέτοιων λαθών μπορεί να είναι η μη σωστή χρήση των αγκυλών { και } ή ακόμη το να ανοίγουμε μια αγκύλη και να μην την κλείνουμε, το να ανοίγουμε κάπου σχόλια και να ξεχνάμε να τα κλείσουμε και τα λοιπά. Όπως ξέρουμε, μέρος της δουλειάς του μεταγλωττιστή είναι και η ανακάλυψη των συντακτικών λαθών του προγράμματος. Υπάρχουν, όμως, και περιπτώσεις όπου ένα λάθος παράγει, άθελά μας, και άλλα λάθη.

2. Εννοιολογικά Λάθη

Το εννοιολογικό λάθος είναι το λάθος στο νόημα των προτάσεων. Στη C εννοιολογικά λάθη μπορούμε να κάνουμε, όταν ακολουθούμε μεν σωστά τους κανόνες της γλώσσας, αλλά με λανθασμένο αποτέλεσμα. Τέτοιο λάθος μπορεί να γίνει, όταν π.χ. αντί να προσθέσουμε δύο μεταβλητές, τις πολλαπλασιάζουμε. Με τα λάθη αυτά βέβαια δεν έχει καμία σχέση ο μεταγλωττιστής. Είναι δική μας δουλειά να τα ανακαλύψουμε και να τα διορθώσουμε. Ο καλύτερος τρόπος για να ανακαλύψουμε τέτοια λάθη είναι να εξετάσουμε το πρόγραμμα βήμα-βήμα.

Μπορούμε ακόμα να χρησιμοποιούμε επιλεκτικά και τη συνάρτηση printf() μέσα στο πρόγραμμα, ώστε να ελέγχουμε τις τιμές κάποιων μεταβλητών του προγράμματος. Τις εντολές printf() τις απομακρύνουμε μετά όταν το πρόγραμμά μας λειτουργήσει κανονικά. Και η χρήση των σχολίων μπορεί να αποδειχθεί χρήσιμη εδώ, γιατί με τη βοήθειά τους μπορούμε να απομονώσουμε κάποιο κομμάτι του προγράμματος προσωρινά και να ελέγξουμε έτσι την ορθότητα του υπόλοιπου προγράμματος. Υπάρχουν και ειδικά προγράμματα που λέγονται *αποσφαλματωτές **(debuggers)*** και που μας επιτρέπουν να βλέπουμε τις τιμές των μεταβλητών του προγράμματος και ποια γραμμή του προγράμματος εκτελείται.

4.11 Τελεστές

Η C έχει τέσσερις τύπους τελεστών: **αριθμητικοί, σύγκρισης (συσχεσιακοί), λογικοί και τελεστές χειρισμού bits (bitwise operators).** Παρακάτω δίνεται ένας πίνακας με όλους τους τελεστές διατεταγμένους σύμφωνα με την προτεραιότητά τους.

προτεραιότητα	τελεστές
υψηλή	( ) [ ] ->
	! ~ ++ -- -(type) * & sizeof
	* / %
	- +
	<< >>
	< <= > >=
	== !=
	&
	^
	|
	&&
	||
	?
χαμηλή	= += -= *= /=

4.11.1 Αριθμητικοί τελεστές

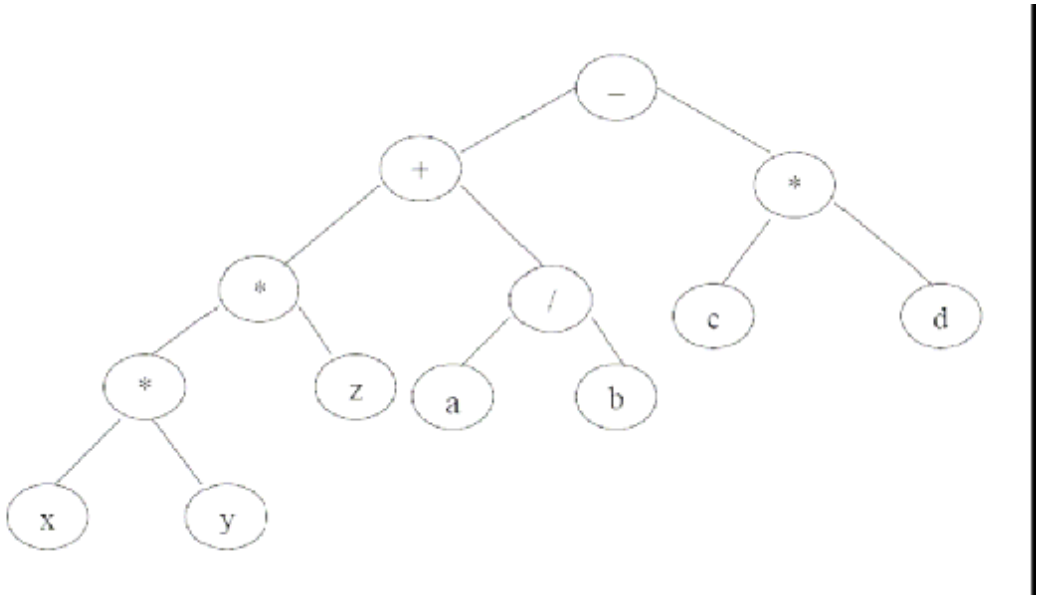
Η C ακολουθεί την αλγεβρική προτεραιότητα στους αριθμητικούς τελεστές. Δηλαδή, μεγαλύτερη προτεραιότητα έχουν οι παρενθέσεις, μετά το πρόσημο (δηλαδή το μείον), ακολουθούν ο πολλαπλασιασμός, η διαίρεση και το modulo και τέλος οι τελεστές πρόσθεσης και αφαίρεσης. Η εντολή καταχώρησης έχει μικρότερη προτεραιότητα από όλες τις πράξεις. Η φορά των πράξεων είναι από αριστερά προς δεξιά για όλους τους αριθμητικούς τελεστές, εκτός αν χρησιμοποιείται το μείον σαν πρόσημο, οπότε η φορά είναι από δεξιά προς αριστερά!

-	αφαίρεση, πρόσημο
+	πρόσθεση
*	πολλαπλασιασμός
/	διαίρεση
%	(mod)
--	ελάττωση μεταβλητής κατά 1
++	αύξηση μεταβλητής κατά 1

Οι τελεστές -- και ++ μπορεί να τοποθετηθούν μπροστά ή μετά ένα τελεσταίο. Η εντολή --x; ισοδυναμεί με την x:=x-1 αλλά η αφαίρεση εκτελείται πριν χρησιμοποιήσουμε την τιμή της x. Όμοια και η εντολή ++x; Η εντολή x--; Ισοδυναμεί με την x:=x-1 αλλά η αφαίρεση εκτελείται αφού χρησιμοποιήσουμε την τιμή της. x.x=3; x=3;y=++x; y=x++; αποτέλεσμα: y=4 (x=4) αποτέλεσμα: y=3 (x=4). Δυαδικοί τελεστικοί στην ίδια υποέκφραση και με την ίδια προτεραιότητα αποτιμούνται από τα αριστερά προς τα δεξιά – *αριστερή εταιρικότητα.*

Παράδειγμα

Η έκφραση x * y * z + a / b – c * d ερμηνεύεται ως: (((x * y) * z) + (a / b)) – (c * d)



Συντακτικά Ορθές	Συντακτικά Λανθασμένες
x + y / 2	((3 + z) / 2
(-x / y) * 2	-2x / y
((3))	5 +* 9
+(9 + z)	6 ^ 4
5 --x	

Σημείωση: Δεν υπάρχει τελεστής που υψώνει στη δύναμη. Η λειτουργία παρέχεται από τη συνάρτηση pow που ορίζεται στη βιβλιοθήκη math.

4.11.2 Οι τελεστές συσχετισμού

Πρόκειται για τελεστές που τους χρησιμοποιούμε για τη δημιουργία απλών λογικών εκφράσεων συσχετισμού (σύγκρισης). Αυτοί είναι:

<	Μικρότερο από (πχ. αν i<j επιστρέφει 1 αντο i είναι μικρότερο από το j)
>	Μεγαλύτερο από
<=	Μικρότερο από ή ίσο
>=	Μεγαλύτερο από ή ίσο
==	Λογικό ίσον
!=	Διάφορο (όχι ίσο)

4.11.3 Λογικοί τελεστές

&&	Λογικό «και» (AND) μας επιστρέφει 1 αν και δύο μεταβλητές είναι λογικό μηδέν, διαφορετικά μηδέν
II	Λογικό διαζευκτικό «ή» (είτε -OR)
I	Λογική άρνηση (NOT)

4.11.4 Ψηφιακοί τελεστές

τελεστής	περιγραφή
&	Ψηφιακό AND
|	Ψηφιακό OR
^	Ψηφιακό αποκλειστικό OR
~	Συμπλήρωμα ως προς ένα
&=	Σύνθετο ψηφιακό AND
|=	Σύνθετο ψηφιακό OR
^=	Σύνθετο ψηφιακό αποκλειστικό OR
<<	Ολίσθηση κατά ένα ψηφίο αριστερά
>>	Ολίσθηση κατά ένα ψηφίο δεξιά

4.12 ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C

```
/* prog01.c – αυτό είναι ένα απλό πρόγραμμα στη γλώσσα C */

#include <stdio.h>

main()

{

        int num;        /* ορίζεται μια ακέραια μεταβλητή με το όνομα num */

        num = 10 ;   /* καταχώρηση τιμής στη μεταβλητή num */

        printf("Ενα πολύ απλό πρόγραμμα σε C.\n"); /* η συνάρτηση printf() */

        printf("Ο αγαπημένος μου αριθμός είναι ο %d.\n", num);

}
```

Αφού μεταγλωττίσουμε και τρέξετε το παραπάνω πρόγραμμα, τότε θα πρέπει να εμφανισθούν στην οθόνη τα παρακάτω :

```
Ένα πολύ απλό πρόγραμμα σε C.
```

```
Ο αγαπημένος μου αριθμός είναι ο 10.

/* prog02.c – το πρόγραμμα αυτό μετατρέπει τα μέτρα σε εκατοστά */

#include <stdio.h>

main()

{

        int cm, metres;


        metres = 2;

        cm = 100 * metres;

        printf("Υπάρχουν %d εκατοστά σε %d μέτρα. \n", cm, meters);


}
```

Όπως βλέπουμε, το πρώτο σχόλιο του προγράμματος περιέχει το όνομά του και το τι ακριβώς κάνει. Ακόμη, το πρόγραμμα δηλώνει δύο ακέραιες μεταβλητές μαζί, τις οποίες και διαχωρίζει με κόμμα. Χρησιμοποιεί τον τελεστή του πολλαπλασιασμού, που είναι το * και εκτυπώνει πολλές μεταβλητές μαζί στη συνάρτηση printf(). Όταν τρέξει το πρόγραμμα, θα δώσει το εξής αποτέλεσμα :

```
Υπάρχουν 200 εκατοστά σε 2 μέτρα.
```

ΚΕΦΑΛΑΙΟ 5^{Ο} - ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΜΙΚΡΟΕΛΕΓΚΤΩΝ AVR

5.1) ΚΑΤΗΓΟΡΙΕΣ ΜΙΚΡΟΕΛΕΝΚΤΩΝ AVR

Οικογένεια	πλεονεκτήματα	εφαρμογές	τεχνικές	παράμετροι

32-bit AVR UC3	Παγκόσμια πιο αποτελεσματικός 32-bit μικροελεγκτής	Γενικού σκοπού	Pico Power	16-512KB Flash
			SleepWalking	48-144 pins
			FlashVualt	Up to 66 MHz
			DMA	1.5 MIPS/MHz
			Self Programming	1

Οικογένεια	πλεονεκτήματα	εφαρμογές	τεχνικές	Παράμετροι
AVR XMEGA microcontroller	Πολύ καλή 8-bit επίδοση	Γενικού σκοπού	Pico Power	16-384 KB Flash
			SleepWalking	32-100 Pins
			DMA	up to 32 MHz
			Event System	1.0 MIPS/MHz
			EEPROM	
			Self Programming	

Οικογένεια	πλεονεκτήματα	εφαρμογές	τεχνικές	Παράμετροι
megaAVR microcontroller	Περισσότερα περιφερικά και επιλογές	Γενικού σκοπού	QTouch	4-256KB Flash
		φωτισμός	Pico Power	28-10 pins
		LCD	SleepWalking	Up to 20 MHz
			EEPROM	1.0 MIPS/MHz
			Self Programming	

Οικογένεια	πλεονεκτήματα	εφαρμογές	τεχνικές	Παράμετροι
8-bit tinyAVR	Μικρό και ισχυρό	Γενικού σκοπού	QTouch	0.5-8KB Flash
		Περιορισμένος χώρος πίνακα	EEPROM	6-32 pins
			0.7V operation	Up to 20 MHZ
				1.0 MIPS/MHz

Οικογένεια	πλεονεκτήματα	εφαρμογές	τεχνικές	Παράμετροι
Battery management	li-ion Battery management	Μετρητής αεριού	1-4 cells	1.8-25V operation
		Εξισορρόπηση στοιχείων	High Side N-MOS	8-40 KB Flash
		πιστοποίηση	Μετρητής Coloump	28-48 pins
		Προστασία μικρών κυκλωμάτων	Αισθητήρας θερμοκρασίας	Up to 8 MHZ
		Προστασία θερμότητας	Self Programming	1.0 MIPS/MHz

Οικογένεια	πλεονεκτήματα	εφαρμογές	τεχνικές	Παράμετροι
Automotive AVR	Έξυπνος σχεδιασμός και Έλενος για απαιτούμενα περιβάλλοντα	Automotive	QTouch	Εύρωστη σχεδίαση για αυτοκινούμενα περιβάλλοντα
			Pico Power	Πλούσιο χαρτοφυλάκιο για ένα εύρος εφαρμογών

			SleepWalking	Εξαρτώμενος σχεδιασμός
			EEPROM	
			Self Programming	

5.2) Ο ΜΙΚΡΟΕΛΕΝΚΤΗΣ AT32UCEL016 ΤΗΣ ΟΙΚΟΓΕΝΕΙΑΣ 32-bit AVR UC3

Η μικροελεγκτής 32-bit AVR UC3 κατασκευάστηκε χάρη στην υψηλή επίδοση του για εφαρμογές που απαιτούν χαμηλή ηλεκτρική κατανάλωση, ενσωματωμένες εφαρμογές χωρίς υψηλό κόστος και εφαρμογές οι οποίες απαιτούν υψηλές επιδόσεις μνήμη προστασίας του μικροελεγκτή είναι γρήγορη, ευέλικτη και υποστηρίζει τα σύγχρονα λειτουργικά συστήματα συνδυασμός από τις τεχνικές ηλεκτρικού έλεγχου επιφέρει ενεργή ηλεκτρική μείωση ρεύματος κάτω από τα 165μA μόλις διατηρεί την ασφάλεια τον εναλλακτικών καταχωρητών.

5.2α) ΠΑΡΑΜΕΤΡΟΙ

Flash (Kbytes)	16 Kbytes
Pin count	48
CPU	32 bit-AVR
USB speed,interface	NO
SPI	5
UART	4
SRAM **(**Kbytes)	8
EEPROM (bytes)	0

5.2β) ΕΦΑΡΜΟΓΕΣ ΠΟΥ ΣΧΕΤΙΖΟΝΤΑΙ ΜΕ ΤΟΝ AT32UC3LO16

1)ΚΑΤΑΣΚΕΥΑΣΤΙΚΗ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗ

Α) Ασφάλεια και Έλεγχος

Β) Πυροπροστασία

2) ΟΙΚΙΑΚΕΣ ΕΡΓΑΣΙΕΣ

Α)κουζίνα

Β)Έλενος κινητήρα

Γ)ψυγείο

Δ)Πλυντήριο

3) ΒΙΟΜΗΧΑΝΙΚΗ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗ

Α)Βιομηχανική αυτοματοποίηση

Β)Αισθητήρες

4)ΦΩΤΙΣΜΟΣ

Α)Έλεγχος φωτισμού

5)ΚΙΝΗΤΗΡΙΕΣ ΗΛΕΚΤΡΙΚΕΣ ΣΥΣΚΕΥΕΣ

Α)Κάμερες

Β)Κινητά τηλεφωνά

Γ)Φορητές συσκευές αναπαραγωγής ήχου

Δ)Επιτραπέζιες κονσόλες παιχνιδιών

Ε) tablets και σημειωματάρια

6)ΠΕΡΙΦΕΡΙΑΚΑ ΥΠΟΛΟΓΙΣΤΗ

Α)Εξαρτήματα υπολογιστή

5.3) Ο ΜΙΚΡΟΕΛΕΝΚΤΗΣ ATxmega128A1U ΤΗΣ ΟΙΚΟΓΕΝΕΙΑΣ 32-BIT X-mega

Ο μικροελεγκτής AT mega128A1U προσφέρει στον χρήστη χαμηλή κατανάλωση ενέργειας, υψηλή επίδοση και κατασκευαστικά χαρακτηριστικά όπως επαναπρογραματιζομενη μνήμη, 8KB SRAM, 2048-Bite EEPROM,4 κανάλια Έλεγχου DMA.Η συγκεκριμένη συσκευή μπορεί να χρησιμοποιηθεί σε μια ευρεία κατάσταση εφαρμογών.

5.3α) ΠΑΡΑΜΕΤΡΟΙ

Flash (Kbytes)	128 Kbytes
Pin count	100
CPU	8 bit-AVR
USB speed	Full speed

USB interface	Device
UART	8
SPI	12
EEPROM (bytes)	2048

5.3β) ΕΦΑΡΜΟΓΕΣ ΠΟΥ ΣΧΕΤΙΖΟΝΤΑΙ ΜΕ ΤΟΝ ATxmega128A1U

1)ΚΑΤΑΣΚΕΥΑΣΤΙΚΗ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗ

Α) Ασφάλεια και Έλεγχος

Β) Πυροπροστασία

2) ΟΙΚΙΑΚΕΣ ΕΡΓΑΣΙΕΣ

Α)κουζίνα

Β)ψυγείο

Γ)Πλυντήριο

3) ΒΙΟΜΗΧΑΝΙΚΗ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗ

Α)Βιομηχανική αυτοματοποίηση

Β)Αισθητήρες

Γ) PLC and modules

Δ) αισθητήρες

4)ΦΩΤΙΣΜΟΣ

Α)Έλεγχος φωτισμού

5)ΚΙΝΗΤΗΡΙΕΣ ΗΛΕΚΤΡΙΚΕΣ ΣΥΣΚΕΥΕΣ

Α)Κάμερες

Β)Κινητά τηλεφωνά

Γ)Φορητές συσκευές αναπαραγωγής ήχου

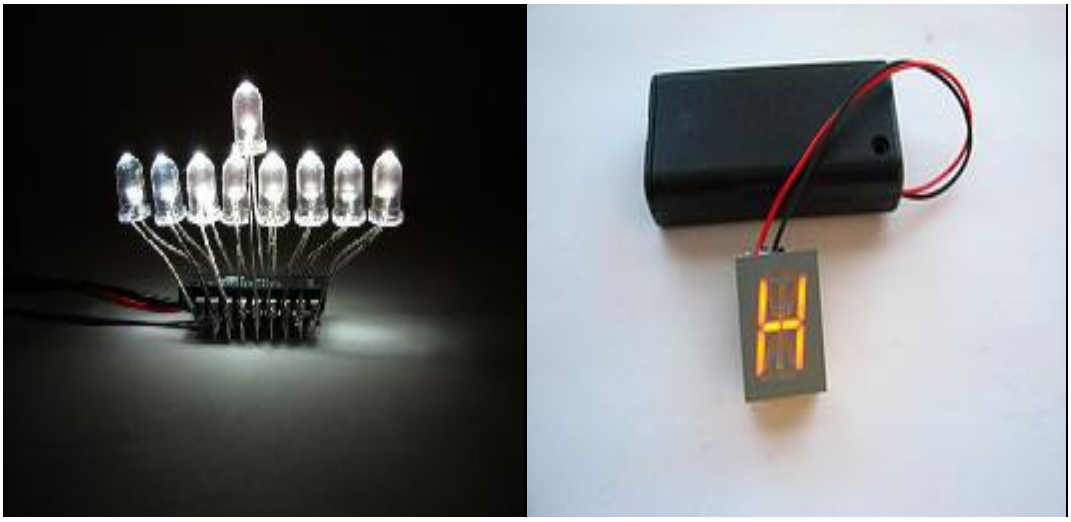
Δ)Επιτραπέζιες κονσόλες παιχνιδιών

Ε) tablets και σημειωματάρια

6)ΠΕΡΙΦΕΡΙΑΚΑ ΥΠΟΛΟΓΙΣΤΗ

Α)Εξαρτήματα υπολογιστή

5.4) Πώς να φτιάξουμε υψηλής τεχνολογικά διακόσμησης φωτιστικά στις διακοπές μας.



Εδώ σας παρουσιάζουμε δυο ελεγκτές για ηλεκτρονικές εργασίες τις οποίες μπορούμε να εφαρμόσουμε στις διακοπές μας. Ενας ηλεκτρονικός μικρός πινάκας που μπορούμε να τον προγραμματίσουμε να αναγράφει οποία λέξη επιθυμούμε από αυτές που έχουμε προγραμματίσει και ένα μικρό Led εννέα φωτιστικών.

5.4α) ΠΟΙΑ Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥΣ

Ο πινάκας παρουσιάζει την λέξη που έχουμε προγραμματίσει να διαβάσει. Είναι λέξη με λίγους χαρακτήρες ,μια λέξη την κάθε φορά .Ο συγκεκριμένος πινάκας είναι προγραμματισμένος να διαβάζει 36 λέξεις. Όπως, <στολίδι>

<καλή χρονιά>

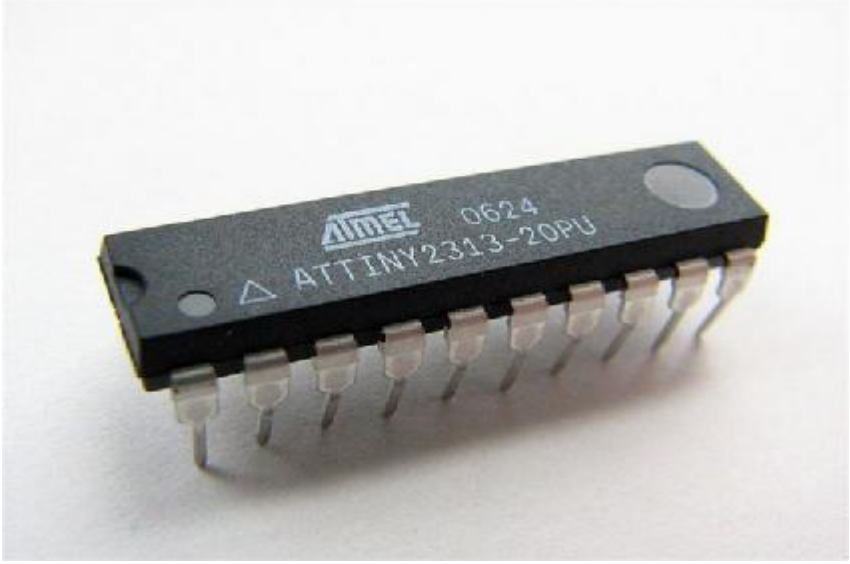
<όλα τα δώρα μας ανήκουν>

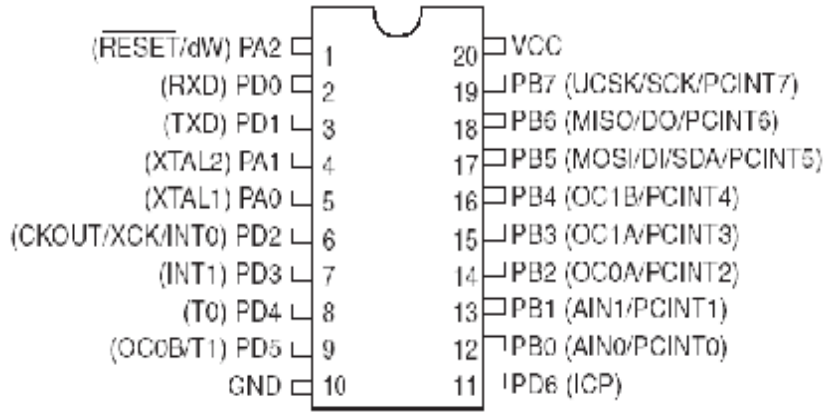
Όταν κάθε φορά απενεργοποιήσουμε τον πίνακα και τον ενεργοποιήσουμε πάλι αναγράφει μια από τις 36 αποθηκευμένες λέξεις που είναι τοποθετημένες στην μνήμη.

Όσο αναφορά το Led μας όπως διακρίνουμε και από την φωτογραφία έχει εννέα φωτιστικά τα όποια είναι σχεδιασμένα να την πρώτη μέρα να ανάβουν τα δυο πρώτα ,την δεύτερη τα Τρία έως ότου φτάσουμε την ογδόη ημέρα να ανάβουν όλα τα φωτιστικά. Όταν το απενεργοποιούμε την συσκευή και την ενεργοποιούμε πάλι, δείχνει ένα φωτιστικό περισσότερο σε σχέση με ην προηγούμενη φορά έκτος και αν την προηγούμενη φορά τα φωτιστικά ήταν και τα εννέα ανοίχτα.

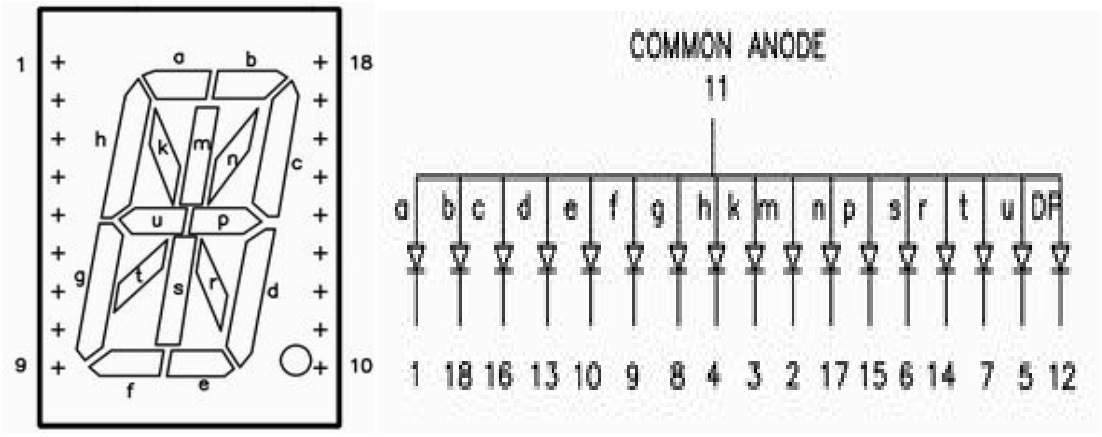
Καταρχήν, για να ξεκινήσουμε την εφαρμογή μας θα χρειαστούμε τον μικροελεγκτή ATtiny 2313 ο οποίος είναι ενας ισχυρός 8-bit μικροελεγκτής ο οποίος χρησιμοποιείται για εφαρμογές μικρής κλίμακας .Ο ATtiny 2313 είναι ένα φθηνό chip με τα εξής χαρακτηρίστηκα

Flash (Kbytes)	2Kbytes
Pin count	20
CPU	8-bit AVR
USB speed	No
UART	1
SPI	2
EEPROM (bytes)	128
SRAM (Kbytes)	0.12





Όπως έχουμε αναφέρει το συγκεκριμένο chip αποτελείται από 20 ακροδέκτες. Επίσης, έχει ρυθμιστεί οι 17 από τις 20 ακροδέκτες να χρησιμοποιούνται σαν λογικές εξόδους .Οι υπόλοιποι 3 ακροδέκτες είναι για την τροφοδοσία του συστήματος (2.7-5.5V DC),για την γείωση μας και ενας ακροδέκτης για την επαναφορά του συστήματος μας. Από την στιγμή που ο μικροελεγκτής είναι ενας μικρός υπολογιστης για να διεκπεραίωση οποιαδήποτε ενεργεία θα πρέπει να προγραμματιστεί. Γραφούμε τα προγράμματα μας στην γλωσσά C και κάνουμε αυτό που περιμένουμε. Δηλαδή, προγραμματίζουμε το chip πότε θέλουμε να κλείσουν τα φώτα και πότε θέλουμε να ανοίξουμε τα φώτα.



Ο μικροελεγκτής έχει 20 ακροδέκτες την ώρα οπού το Kingbright PSA08-11HWA έχει 18 ακροδέκτες για τον λόγο αυτό οι ακροδέκτες του μικροελεγκτή δεν συσχετίζονται κατευθείαν με τους ακροδέκτες του Kingbright PSA08-11HWA.

Ο ακροδέκτης 2 του ATtiny 2313 συνδέεται με τον ακροδέκτη 18

Ο ακροδέκτης 11 του ATtiny 2313 συνδέεται με τον ακροδέκτη 9

Ο ακροδέκτης 19 του ATtiny 2313 συνδέεται με τον ακροδέκτη 1

Για κάθε γράμμα που θα γράψουμε στην οθόνη του στολιδιού, αποφασίζουμε σε πιο τμήμα να το τοποθετήσουμε. Αυτά τα τμήματα είναι μαρκαρισμένα ελάχιστα με το δυαδικό αριθμό 1 και αυτά που είναι περισσότερο μαρκαρισμένα με τον αριθμό 0.Τα 16 δυαδικά ψηφία αποφασίζουμε σε πιο στάδιο θα τα τοποθετήσουμε από τα 16 τμήματα τα οποία περιλαμβάνουν δυο bite λέξεις δεδομένων . Για παράδειγμα, το τμήμα <α> της οθόνης είναι ελεγχόμενο από τον ακροδέκτη 1.Ο ακροδέκτης 1της οθόνης συνδέεται με τον ακροδέκτη 19 του μικροελεγκτή, η οποία έξοδος είναι η B7.Ακολουθοντας παρόμοια λογική έχουμε τον ακόλουθο πίνακα.

BYTE HIGH BYTE LOW

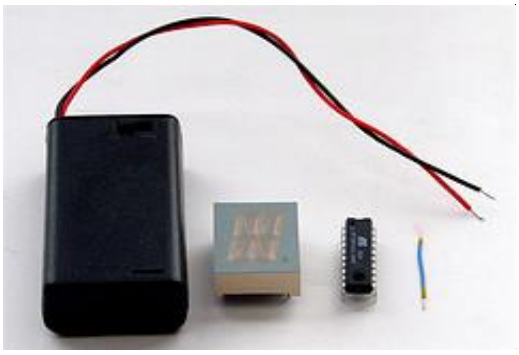
Seg.	c	P	f	e	d	r	n	b	a	m	k	h	u	s	t	g
I/O PIN	A1	A	D6	D5	D3	D2	D1	D0	B7	B6	B5	B4	B3	B2	B1	B0

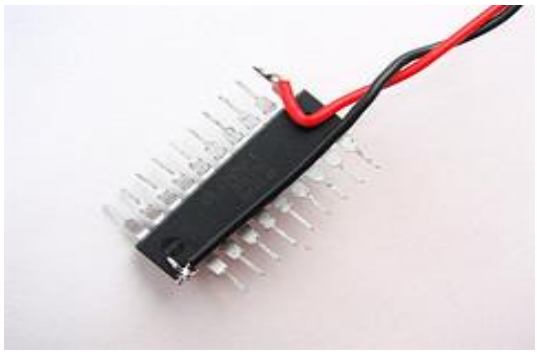
Με αυτόν τον ορισμό των δυο byte δεδομένων, είναι πιθανό να κατασκευαστεί ο πινάκας δεδομένων, με 16-bit δυαδική παρουσίαση από το κάθε γράμμα. Η δεκαδική τιμή αντιστοιχεί στο 2αδικο νούμερο που έχει δοθεί.

Μια μέθοδο την όποια ακολουθούμε ότι όταν έχουμε έναν ακέραιος αριθμός ο οποίος αντιπροσωπεύει το τρέχον μήνυμα στην οθόνη του στολιδιού έχει αποθηκεύει στην EEPROM.Χρησιμοποιούμε την συγκεκριμένη μνήμη προγράμματος για τον λόγο ότι έχει την ικανότητα να αφήνει το chip να αλλάζει την κατάσταση του από την προηγούμενη φορά που το είχαμε ανοιχτό. Όταν το πρόγραμμα έχει γραφτεί χρησιμοποιούμε κάποια εργαλεία (συμπεριλαμβανόμενου και του avr-gcc) να συντάξει τον κώδικα C σε αντικείμενο και έτσι να προγραμματιστεί με χαμηλό κόστος.

ΠΩΣ ΤΟ ΕΦΑΡΜΟΖΟΥΜΕ ΣΤΗΝ ΠΡΑΞΗ

Ξεκινώντας την εφαρμογή μας θα πρέπει να έχουμε στην διάθεση μας τα κατάλληλα εργαλεία για την πραγματοποίηση της εργασίας μας συμπεριλαμβανόμενου και του προγραμματισμένου μικροελεγκτή. Επίσης θα πρέπει να έχουμε στην διάθεση μας μια μπαταρία η άλλη πηγή τάσης και τα απαραίτητα LED.Στην αγορά υπάρχουν πολλά LED ωστόσο είναι κάπως πιο εύκολο να χρησιμοποιήσουμε led μικρού μήκους (3 mm).Για το στολίδι θα χρησιμοποιήσουμε το Kingbright PSA08-11HWA ένα κόκκινο 16 τμημάτων αλφαριθμητικό LED οθόνης. Είναι αρκετά μεγάλο 20mm (0.8 inch) και μια κοινή διαμόρφωση μονάδας δίσκου μειονέκτημα της συγκεκριμένης οθόνης είναι ότι έχει κάπως μια μεταβλητή διαθεσιμότητα.





Για να συναρμολογήσουμε ένα από τα στολίδια τοποθετούμε σε ένα μικρό κιτ την μπαταρία μας, η οθόνη του led μας θα πρέπει να είναι PSA08-11HWA η κάποια παρεμφερή κατάλληλη για την ορθή λειτουργία του συστήματος. Ακόμη , τον προγραμματισμένο μικροελεγκτή μας και ένα μικρό κομμάτι σύρματος μονωμένο με χαλκό. Μόλις πραγματοποιήσουμε τις απαραίτητες συνδέσεις του chip, έκτος από τον ακροδέκτη 10 ο οποίος θα πρέπει να τοποθετηθεί κάτω στην μέση του κιτ. Στην συνέχεια το κόκκινο καλώδιο από την μπαταρία θα τοποθετήσει στον ακροδέκτη 10 ο οποιος είναι η τάση μας Vcc και το μαύρο καλώδιο στον ακροδέκτη 10 ο οποιος είναι για την γείωση (GND).

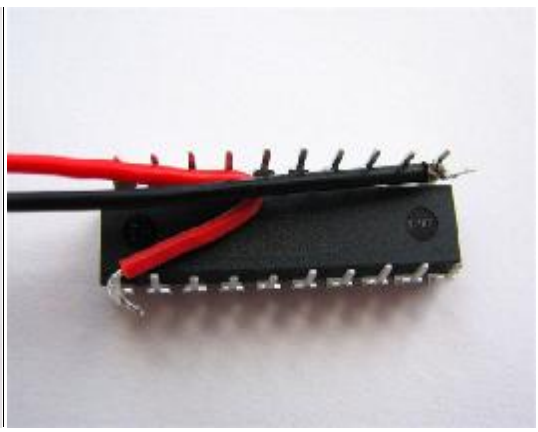
Στην συνεχεία θα πρέπει να αναγνωρίσουμε την κοινή άνοδο στην οθόνη του led,η οποία είναι ο δεύτερος ακροδέκτης στην γωνία του δεκαδικού σημείου. Το λυγίζουμε και συγκολλούμε το καλώδιο κάνοντας μια μικρή μόνωση. Λυγίζουμε το καλώδιο έτσι ώστε να πάει απέναντι στην γωνία της οθόνης. Ταιριάζουμε τον μικροελεγκτή στην οθόνη του led και βλέπουμε πως προσαρμόζονται μαζί ακροδέκτης 10 του μικροελεγκτή συνδέεται στην γωνία του δεκαδικού σημείου και ο εναπομένοντας ακροδέκτης του led στέκεται εκεί. Ο ακροδέκτης 1 και 20 του μικροελεγκτή δεν θα

πρέπει να βρίσκεται σε επαφή με τους ακροδέκτες του led όπως κάνουμε οι υπόλοιποι ακροδέκτες.

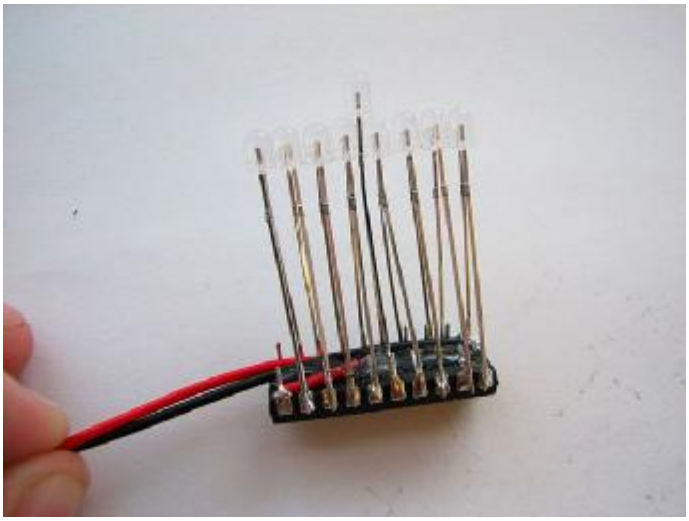
Οι ακροδέκτες 2-8 και 10-19 του μικροελεγκτή θα πρέπει να ακουμπανε στους ακροδέκτες της οθόνης του led?Μονώνουμε αυτές τις συνδέσεις .Μονώνουμε τον ακροδέκτη 9 του μικροελεγκτή στον μοναδικό ακροδέκτη του led,το οποίο ίσως χρειαστεί να το κάμψουμε. Συγκολλούμε το άλλο τέλος του καλωδίου από την κοινή κάθοδο της οθόνη του led στον ακροδέκτη 20 του μικροελεγκτή.

ΣΥΝΔΕΣΗ ΤΩΝ ΦΩΤΙΣΤΙΚΩΝ (ΜΕΝΟΡΑΗ)

Για να κάνουμε την συνδεσμολογία βάζουμε μαζί ένα μικρό κιτ μαζί με το κουτί τις μπαταριάς, τα εννέα led και τον προγραμματισμένο μικροελεγκτή. Έχοντας συγκολλήσει το κόκκινο καλώδιο του κουτιού της μπαταριάς στον ακροδέκτη 20 του chip (+Vcc) και στο μαύρο καλώδιο του ακροδέκτη 10 (GND).



Οι κάθοδοι από τα εννέα led θα συνδεθούν στους ακροδέκτες 2-10 και οι άνοδοι θα συνθεθουν απευθείας απέναντι τους,ακροδεκτες 11-19.Τα συγκεκριμένα φωτίστηκα είναι διαμέτρου μικρού τύπου (3mm).Ίσως θα χρειαστεί περισσότερη προσπάθεια για την παράκαμψη εάν χρησιμοποιούμε φωτιστικά μεγαλυτέρου διαμέτρου. Στην συνεχεία αφού συνδέσουμε τις καθόδους των οκτώ leds,συνδέσουμε τους ακροδέκτες 2-5 και 7-10.Στην συνεχεία, συνδέσουμε τους ακροδέκτες 11-14 και 16-19.Τωρα υπάρχει μόνο ενας ακροδέκτης να συνδέσουμε. Συνδεουμε την άνοδο του ακροδέκτη 15 και την κάθοδο στον ακροδέκτη 6.Καλο θα ήταν να τοποθετήσουμε τις συνδέσεις μας σε διαφορετικό ύψος από τις υπόλοιπες έτσι ώστε να μπορούμε να τα ξεχωρίζουμε ακόμα και όταν η συσκευή είναι απενεργοποιημένη.



ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Πεκμετζή Π., *Συστήματα Μικροϋπολογιστών Ι (1995)* Εκδόσεις Συμμετρία, Αθήνα 1995.
2. Πεκμετζή Π., *Συστήματα Μικροϋπολογιστών ΙΙ (2009)* Εκδόσεις Συμμετρία, Αθήνα 2009.
3. Εργαστηριακό φυλλάδιο *Μικροϋπολογιστών ΙΙ*
4. Ritchie Dennis & Brian Kernighan *Η γλώσσα προγραμματισμού C*, Εκδόσεις Κλειδάριθμος.
5. WWW. ATMEL AVR MICROCONTROLLER 32-BIT
6. WWW.EVIL MADSCIENTIST.COM
7. WWW.ATMEL.COM