

**Α.Ε.Ι. ΠΕΙΡΑΙΑ Τ.Τ.**

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ

**ΑΝΩΤΑΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ  
ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ**

**“ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ”**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**«Ενσωματωμένα Συστήματα»  
(Embedded Systems)**

**ΚΩΣΤΟΠΟΥΛΟΥ ΑΓΟΡΙΤΣΑ**

**Α.Μ.: 43277**



**ΥΠΕΥΘΥΝΗ ΚΑΘΗΓΗΤΡΙΑ: ΒΕΛΩΝΗ ΑΝΑΣΤΑΣΙΑ**

# ΠΕΡΙΕΧΟΜΕΝΑ

## ΕΙΣΑΓΩΓΗ

- A. Σκοπός της εργασίας
- B. Αντικείμενο της εργασίας
- Γ. Ευχαριστίες

## ΚΕΦΑΛΑΙΟ 1

### 1. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΚΑΙ ΟΡΙΣΜΟΙ

- 1.1 Ενσωματωμένα Συστήματα
- 1.2 Τι είναι τα Ενσωματωμένα Συστήματα;
- 1.3 Ιστορική Αναδρομή Ενσωματωμένων Συστημάτων
- 1.4 Η κατηγορία των Ενσωματωμένων Συστημάτων
- 1.5 Τα Ενσωματωμένα Συστήματα ως μέρος της καθημερινότητας
- 1.6 Λειτουργικά Συστήματα Ενσωματωμένων Συστημάτων
  - 1.6.1 Ενσωματωμένα Windows XP
  - 1.6.2 Ενσωματωμένο Linux
  - 1.6.3 Τι είναι η CPU και ποια η Χρησιμοποίηση της;
  - 1.6.4 Τι είναι το POSIX?
- 1.7 Ενσωματωμένα Συστήματα πραγματικού χρόνου (Real Time)
- 1.8 Πολυπύρηνια Ενσωματωμένα Συστήματα
- 1.9 Κατανεμημένα Ενσωματωμένα Συστήματα
  - 1.9.1 Χαρακτηριστικά Κατανεμημένων Συστημάτων
  - 1.9.2 Μειονεκτήματα Κατανεμημένων Συστημάτων
  - 1.9.3 Αρχιτεκτονικές Κατανεμημένων Συστημάτων
- 1.10 Χρήση των ενσωματωμένων συστημάτων

## ΚΕΦΑΛΑΙΟ 2

### 2. ΙΔΙΟΤΗΤΕΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ

- 2.1 Χαρακτηριστικά των ενσωματωμένων συστημάτων
- 2.2 Ιδιαιτερότητες των ενσωματωμένων συστημάτων
- 2.3 Μεθοδολογία Ανάπτυξης Ενσωματωμένων Συστημάτων
  - 2.3.1 Ανάπτυξη Ενσωματωμένων Συστημάτων
- 2.4 Φάσεις Ανάπτυξης ενός ενσωματωμένου συστήματος
- 2.5 Γλώσσα Assembly
- 2.6 Γλώσσα C/C++
  - 2.6.1 Γενικά Χαρακτηριστικά της C++
  - 2.6.2 Το σύστημα της C++
- 2.7 Παραδείγματα ενσωματωμένων συστημάτων
  - 2.7.1 Αναλυτικά Παραδείγματα Ενσωματωμένων Συστημάτων
- 2.8 Συσκευές με ενσωματωμένα συστήματα
- 2.9 Ανάπτυξη Υλικού- Λογισμικού Για Τα Ενσωματωμένα Συστήματα

## **ΚΕΦΑΛΑΙΟ 3**

### **3. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ**

- 3.1 Τι είναι Μοντελοποίηση;
- 3.2 Η ανάγκη για τη Μοντελοποίηση
- 3.3 Το NRE εξαρτάται από τη μοντελοποίηση
- 3.4 Η σημασία της Μοντελοποίησης
- 3.5 Οι 4 αρχές της Μοντελοποίησης
- 3.6 Μοντελοποίηση στο χαρτί
- 3.7 Γλώσσες Μοντελοποίησης
- 3.8 Τρόποι Μοντελοποίησης
- 3.9 Μοντελοποίηση με Γενικά Εργαλεία (π.χ. MATLAB)
- 3.10 Μοντελοποίηση με Προγράμματα (π.χ. C, Java)
- 3.11 Από Μοντέλο σε Ολοκληρωμένο Σύστημα- Σχεδιαστικές Ροές
- 3.12 Χρήση Εργαλείων CAD

## **ΚΕΦΑΛΑΙΟ 4**

### **4. ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΜΕΤΡΗΣΕΩΝ ΚΑΙ ΕΛΕΓΧΟΥ ΓΙΑ ΤΟ ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ (Internet of Things – IoT)**

- 4.1 Ο Ορισμός του IoT
- 4.2 Η Ιστορία του Internet of Things
- 4.3 Τάσεις και Στοιχεία του IoT
- 4.4 Cloud Networking
- 4.5 Cloud ή Cloud Computing
  - 4.5.1 Τύποι Cloud
  - 4.5.2 Κύρια Χαρακτηριστικά του Cloud Computing
  - 4.5.3 Προσβασιμότητα και Χρήση του Cloud
  - 4.5.4 Cloud και IoT
- 4.6 Εφαρμογές
  - 4.6.1 Εφαρμογές Για Τον Άνθρωπο
  - 4.6.2 Εφαρμογές Για Το Σπίτι
  - 4.6.3 Εφαρμογές Πόλης
  - 4.6.4 Εφαρμογές Στη Βιομηχανία Και Το Περιβάλλον

## **ΚΕΦΑΛΑΙΟ 5**

### **5. ΠΡΩΤΟΚΟΛΛΑ ΚΑΙ ΔΙΚΤΥΑ ΓΙΑ ΤΑ ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ**

- 5.1 Τι είναι Πρωτόκολλα;
- 5.2 Τι είναι Δίκτυα;
- 5.3 Τοπολογίες Δικτύων
- 5.4 IP(Internet Protocol)
- 5.5 Τι είναι το TCP/IP και τι κάνει;
- 5.6 Ασύρματες Ενσωματωμένες Συσκευές (Wireless Embedded Devices)
- 5.7 Ασύρματα Δίκτυα Αισθητήρων

## **ΚΕΦΑΛΑΙΟ 6**

### **6. ΣΧΕΔΙΑΣΜΟΣ ΤΩΝ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ**

- 6.1 Μεθοδολογία Σχεδίασης
  - 6.1.1 Παράδειγμα Σχεδίασης Ενσωματωμένων Συστημάτων
- 6.2 Μετρικές σχεδίασης ενσωματωμένων συστημάτων
- 6.3 Φάσεις Σχεδίασης
- 6.4 Ψηφιακή Σχεδίαση-Ενσωματωμένα Συστήματα με VHDL
- 6.5 Η Διαδικασία Σχεδίασης
- 6.6 Οικονομικά Θέματα Σχεδίασης Ενσωματωμένων Συστημάτων
  - 6.6.1 Χρόνος Εισαγωγής του προϊόντος στην Αγορά
  - 6.6.2 Χαρακτηριστικά & Προκλήσεις

## **ΚΕΦΑΛΑΙΟ 7**

### **7. ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ**

- 7.1 Θεμελιώδη στοιχεία της Αρχιτεκτονικής
- 7.2 Η μνήμη σε ένα προγραμματιζόμενο σύστημα
- 7.3 Μνήμη τυχαίας προσπέλασης- Random Access Memory, RAM
- 7.4 Μνήμη μόνο για Ανάγνωση – ROM
- 7.5 Electrically Erasable Programmable Memory - EEPROM
- 7.6 Η μνήμη FLASH
- 7.7 Τρόπος Αποθήκευσης των Byte
- 7.8 Κρυφή Μνήμη
  - 7.8.1 Τεχνικές Κρυπτογράφησης Κρυφής Μνήμης
  - 7.8.2 Πολιτική Αντικατάσταση Κρυφής Μνήμης
- 7.9 Ο Επεξεργαστής σε ένα Ενσωματωμένο Σύστημα
- 7.10 Περιγραφή της αρχιτεκτονικής – στόχου για ένα Ενσωματωμένο Σύστημα
  - 7.11 Βασική Αρχιτεκτονική FPGA
  - 7.12 Πολυεπεξεργασία σε Chip
    - 7.12.1 Συμμετρική Πολυεπεξεργασία
    - 7.12.2 Ετερογενής Πολυεπεξεργασία
  - 7.13 Βελτιστοποίηση Σχεδίασης Ενσωματωμένων Συστημάτων με χρήση δυναμικής μνήμης

## **ΚΕΦΑΛΑΙΟ 8**

### **8. ΤΡΟΠΟΙ ΥΛΟΠΟΙΗΣΗΣ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ**

- 8.1 Εισαγωγή
- 8.2 Οι πρώτες υλοποιήσεις
- 8.3 Κατηγορίες υλοποίησης
- 8.4 Τεχνολογίες ενσωματωμένων συστημάτων
  - 8.4.1 Τεχνολογίες επεξεργαστών
  - 8.4.2 Τεχνολογίες Ολοκληρωμένων Κυκλωμάτων
  - 8.4.3 Τεχνολογίες Σχεδίασης
- 8.5 Διασύνδεση και Είσοδος / Έξοδος

- 8.5.1 Διασύνδεση
- 8.5.2 Είσοδος και Έξοδος
- 8.5.3 Τυπικοί Διάυλοι Ενσωματωμένων Συστημάτων
- 8.6 Μικροεπεξεργαστές και Μικροελεγκτές
  - 8.6.1 Μικροεπεξεργαστές
  - 8.6.2 Μικροελεγκτές
    - a) Συνήθη Υποσυστήματα Μικροελεγκτών
    - b) Διαδεδομένες Κατηγορίες Μικροελεγκτών
    - c) Μικροελεγκτής PIC
- 8.7 Διαφορές Μικροεπεξεργαστή και Μικροελεγκτή

## **ΚΕΦΑΛΑΙΟ 9**

### **9. ΜΕΘΟΔΟΛΟΓΙΑ ΕΠΙΛΥΣΗΣ ΕΦΑΡΜΟΓΩΝ**

- 9.1 Περιγραφή Ενσωματωμένων Συστημάτων
- 9.2 Σχεδιασμός Εφαρμογών
  - 9.2.1 Σχεδίαση με μικρο-ελεγκτή 8bit - Η αναπτυξιακή πλατφόρμα ARDUINO
  - 9.2.2 Σχεδίαση σε επαναδιαμορφώσιμη Αρχιτεκτονική Altera FPGA
- 9.3 Μεθοδολογία Βελτιστοποίησης Αλγορίθμων για χαμηλή κατανάλωση ενέργειας και υψηλή απόδοση
- 9.4 Το Μέλλον των Ενσωματωμένων Συστημάτων

## **ΕΙΣΑΓΩΓΗ**

Η αυτοματοποίηση και η κυριαρχία των υπολογιστικών διατάξεων στην καθημερινή ζωή του ανθρώπου έχουν οδηγήσει τα ενσωματωμένα συστήματα (embedded systems) να αποτελούν τα τελευταία χρόνια σημαντική εξέλιξη στον τομέα της πληροφορικής και της εφαρμοσμένης μηχανικής.

### **A. Σκοπός της Εργασίας**

Σκοπός της Εργασίας είναι να κατανοήσουν οι αναγνώστες τον ορισμό του ενσωματωμένου συστήματος. Σημαντικό, επίσης, είναι να κατανοήσουν και να συνειδητοποιήσουν τα προβλήματα στη σχεδίαση συστημάτων ενσωματωμένης υπολογιστικής. Ακόμη, να αντιληφτούν την ανάπτυξη της μεθοδολογίας σχεδίασης των ενσωματωμένων συστημάτων και να είναι σε θέση να μπορούν να σχεδιάζουν ένα ενσωματωμένο σύστημα. Τέλος, η εισαγωγή του αναγνώστη στο αντικείμενο της συσχεδίασης ενσωματωμένων συστημάτων που περιλαμβάνει ανάπτυξη λογισμικού καθώς και σχεδίαση ή επιλογή του κατάλληλου υλικού.

### **B. Αντικείμενο της Εργασίας**

Αντικείμενο της εργασίας είναι η εισαγωγή στους φοιτητές του σχεδιασμού των ενσωματωμένων συστημάτων και η πρακτική τους εξάσκηση σε αυτό, η παρουσίαση των βασικών συστατικών στοιχείων των συστημάτων αυτών, του τρόπου λειτουργίας τους και της συνεργασία τους για την ανάπτυξη ολοκληρωμένων συστημάτων.

### **Γ. Ευχαριστίες**

Θα ήθελα να ευχαριστήσω θερμά την καθηγήτρια μου κα. Αναστασία Βελώνη. Η εμπιστοσύνη της, η καθοδήγηση και ο πολύτιμος χρόνος που αφιέρωσε ήταν τα πιο σημαντικά συστατικά, για την ολοκλήρωση της πτυχιακής μου εργασία με επιτυχία.

Ιδιαίτερες ευχαριστίες και απέραντη ευγνωμοσύνη θα ήθελα να εκφράσω για τους αγαπημένους μου γονείς, οι οποίοι με υποστήριξαν και με συμπαρασταθήκαν όλα αυτά τα χρόνια για την αποπεράτωση των σπουδών μου. Τα λόγια είναι περιττά, τους αγαπάω πολύ.

Τέλος ένα μεγάλο ευχαριστώ σε όλους τους καθηγητές μου, που όλα αυτά τα χρόνια αγωνίστηκαν για να μας μεταδώσουν γνώσεις και μόρφωση. Απώτερος σκοπός των κόπων τους, να με κάνουν καλύτερη, ικανότερη και να με προετοιμάσουν κατάλληλα για την αγορά εργασίας και στη ζωή γενικότερα.

# ΚΕΦΑΛΑΙΟ 1

## 1. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΚΑΙ ΟΡΙΣΜΟΙ

Σύστημα είναι μια λειτουργία (operation) ή επεξεργασία (process) ή διάταξη (device) για την οποία υπάρχει σχέση αιτίου και αποτελέσματος.

Ενσωματωμένο σύστημα θεωρείται ένα σύστημα που ελέγχεται από κάποια μορφή υπολογιστή, που μπορεί να είναι ένας κοινός υπολογιστής, ένας μικροελεγκτής ή ένας επεξεργαστής ψηφιακού σήματος. Τα πρώτα ενσωματωμένα συστήματα εμφανίστηκαν τη δεκαετία του '60. Σήμερα τα ενσωματωμένα συστήματα έχουν πλέον καθιερωθεί ως καινοτόμος και σημαντικός τομέας της τεχνολογίας και της πληροφορικής. Προκειμένου να κρατηθούν από άποψη κατασκευής απλά και από πλευρά κόστους φτηνά, κατασκευάζονται ως αυτόνομα συστήματα. Μερικά, πάντως, από τα συστήματα αυτά έχουν χαρακτηριστικά συστημάτων πραγματικού χρόνου, ώστε να είναι ασφαλή και εύχρηστα και να προσομοιάζουν με τα υπολογιστικά συστήματα γενικού σκοπού.

Επίσης, πολλά ενσωματωμένα συστήματα αποτελούν μικρά υποσυστήματα ενός μεγαλύτερου συστήματος. Για παράδειγμα, ένα ενσωματωμένο σύστημα σε ένα αυτοκίνητο επιτελεί μια συγκεκριμένη λειτουργία, που είναι μέρος της συνολικής λειτουργίας του αυτοκινήτου. Συγκεκριμένα, Ενσωματωμένα συστήματα είναι μικρά σε μέγεθος υπολογιστικά συστήματα, που ο άμεσος ρόλος τους δεν είναι η επεξεργασία δεδομένων. Εξυπηρετούν αυτοματισμούς και βρίσκονται πλέον σε κάθε τύπο ηλεκτρονικής συσκευής καθημερινής χρήσης, όπως κάμερες, οικιακές συσκευές, συσκευές γραφείου. Απαντώνται σε συστήματα μετρήσεων, σε παιχνίδια, σε βιομηχανικούς αυτοματισμούς, σε αυτοκίνητα και όλα τα μέσα μεταφοράς, σε ηλεκτρονικά όργανα και τηλεπικοινωνιακά συστήματα. Τα ενσωματωμένα συστήματα αποτελούν την ταχύτερη αναπτυσσόμενη βιομηχανία και αλλάζουν την καθημερινότητα, καθώς διεισδύουν παντού. Αποτελούν τη βάση της τεχνολογίας που αναφέρεται ως διαδίκτυο πραγμάτων (Internet of Things).

Σε γενικές γραμμές, ενσωματωμένο σύστημα είναι ένα σύστημα ειδικού σκοπού το οποίο επιτελεί μία προκαθορισμένη εργασία και έχει ως βασικά χαρακτηριστικά του την ύπαρξη στο βασικό σώμα της αρχιτεκτονικής του ενός τουλάχιστον μικροελεγκτή, το μικρό φυσικό του μέγεθος, το χαμηλό κόστος και την ικανοποίηση διαφόρων ειδών αυστηρών περιορισμών, κυρίως χρονικών. Ο πυρήνας ενός ενσωματωμένου συστήματος αποτελείται από έναν τουλάχιστον μικροεπεξεργαστή ή μικροελεγκτή ο οποίος είναι προγραμματισμένος έτσι ώστε να τρέχει μία συγκεκριμένη software εφαρμογή με την κατάλληλη παραμετροποίηση. Αυτή η «εξειδίκευση» των ενσωματωμένων συστημάτων δίνει τη δυνατότητα στους σχεδιαστές τους να τα βελτιστοποιήσουν έτσι ώστε να ικανοποιούνται οι προδιαγραφές τους, ενώ ταυτόχρονα να υπάρχουν κέρδη σε επίπεδο μεγέθους, κόστους, χρόνου εκτέλεσης ή κατανάλωσης ισχύος. Επίσης πολλές φορές υπάρχουν και οι λεγόμενοι real – time περιορισμοί (χρονικοί περιορισμοί για εκτέλεση σε

πραγματικό χρόνο). Η γενική λοιπόν λογική πίσω από αυτά τα συστήματα είναι να εκτελούν σωστά και αποδοτικά την εφαρμογή για την οποία έχουν σχεδιαστεί. Όσον αφορά τώρα στις αρχιτεκτονικές οι οποίες χρησιμοποιούνται στο σχεδιασμό ενσωματωμένων συστημάτων, ενδεικτικά αναφέρεται ότι ορισμένες από αυτές είναι οι: ARM, MIPS, PowerPC και X86.

Ένα ενσωματωμένο σύστημα είναι ένα υπολογιστικό σύστημα ειδικού σκοπού, σχεδιασμένο έτσι ώστε να εκτελεί μια ή και περισσότερες συναρτήσεις, συνήθως σε σταθερές πραγματικού χρόνου. Είναι συνήθως ενσωματωμένο σαν ένα μέρος μιας ολοκληρωμένης συσκευής, περιλαμβάνοντας hardware καθώς και μηχανικά μέρη. Σε αντίθεση, ένα υπολογιστικό σύστημα γενικού σκοπού, όπως ένας προσωπικός υπολογιστής, ανάλογα με τον προγραμματισμό που του έχει γίνει, μπορεί να κάνει πολλά διαφορετικά καθήκοντα. Τα ενσωματωμένα συστήματα ελέγχουν πολλές από τις κοινές καθημερινές συσκευές που χρησιμοποιούμε σήμερα. Από τη στιγμή που ένα σύστημα είναι προσανατολισμένο σε συγκεκριμένα καθήκοντα, οι μηχανικοί σχεδίασης μπορούν να το βελτιστοποιήσουν, μειώνοντας το μέγεθος και το κόστος του.

**ΓΕΝΙΚΟΣ ΟΡΙΣΜΟΣ:** Ενσωματωμένο Σύστημα αποτελεί συνδυασμό υλικού και λογισμικού υπολογιστών και, ίσως, επιπρόσθετων μηχανικών ή άλλων τμημάτων που έχει σχεδιαστεί για την εκτέλεση αφοσιωμένων λειτουργιών. Σε αρκετές περιπτώσεις, τα ενσωματωμένα συστήματα είναι μέρος μεγαλύτερου συστήματος ή προϊόντος, όπως στην περίπτωση του συστήματος ABS των αυτοκινήτων.

## 1.1 Ενσωματωμένα Συστήματα

Ζούμε στη εποχή της επανάστασης της πληροφορίας, με τους υπολογιστές να έχουν εκπληκτική υπολογιστική ισχύ, για κάθε μας χρήση. Οι υπολογιστές βρίσκουν το δρόμο για λύση, σε κάθε μας δραστηριότητα. Μερικοί σχεδιάζονται να είναι όσο το δυνατό ισχυρότεροι, χωρίς να υπολογίζουν το κόστος, για απαιτητικές εφαρμογές στη βιομηχανία και την έρευνα. Άλλοι έχουν σχεδιαστεί για το σπίτι και το γραφείο, λιγότερο ισχυροί αλλά και μικρό κόστος. Υπάρχει και μια ακόμη κατηγορία υπολογιστών που γνωρίζουν λιγότεροι και δεν φαίνεται. Αυτό το είδος των υπολογιστών έχει σχεδιαστεί ενσωματωμένο στη συσκευή, με σκοπό την παροχή ελέγχου. Αυτοί οι μικροί υπολογιστές είναι κρυμμένοι, από την άποψη ότι ο χρήστης συχνά δεν γνωρίζει την ύπαρξη τους. Αυτή η συσκευή είναι που ονομάζεται ενσωματωμένο σύστημα (embedded system), και η εργασία μιλάει για αυτή την τεχνολογία. Ο έλεγχος των λειτουργιών στα ενσωματωμένα συστήματα γίνεται από τους μικροελεγκτές (microcontrollers).

Η πλειοψηφία των ψηφιακών συστημάτων που σχεδιάζονται και κατασκευάζονται σήμερα είναι ενσωματωμένα συστήματα (embedded systems) στα οποία μεγάλο τμήμα της επεξεργαστικής εργασίας γίνεται από έναν ή περισσότερους υπολογιστές που αποτελούν μέρος του συστήματος. Στην πραγματικότητα, η τεράστια πλειοψηφία των υπολογιστών που χρησιμοποιούνται σήμερα βρίσκονται μέσα σε ενσωματωμένα



συστήματα, και όχι σε προσωπικούς υπολογιστές και άλλα συστήματα γενικού σκοπού. Οι πρώτοι υπολογιστές ήταν μεγάλα συστήματα από μόνοι τους, και σπάνια θεωρούνταν συστατικά στοιχεία μεγαλύτερων συστημάτων.

Ωστόσο, με την ανάπτυξη της τεχνολογίας, ειδικότερα της τεχνολογίας των ολοκληρωμένων κυκλωμάτων, έγινε πρακτικά εφικτό να ενσωματωθούν μικροί υπολογιστές σαν συστατικά ενός κυκλώματος και να προγραμματισθούν ώστε να υλοποιούν μέρος της λειτουργικότητας του κυκλώματος. Οι ενσωματωμένοι υπολογιστές συνήθως δεν έχουν την ίδια μορφή με τους υπολογιστές γενικού σκοπού, όπως οι επιτραπέζιοι και οι φορητοί προσωπικοί υπολογιστές. Αντίθετα, ένας ενσωματωμένος υπολογιστής αποτελείται από έναν πυρήνα επεξεργαστή (processor core), μαζί με στοιχεία μνήμης για την αποθήκευση του προγράμματος και των δεδομένων για το πρόγραμμα που θα εκτελεστεί στον πυρήνα επεξεργαστή, και άλλα συστατικά στοιχεία για τη μεταφορά δεδομένων μεταξύ του πυρήνα του επεξεργαστή και του υπολοίπου συστήματος.

Τα προγράμματα που εκτελούνται στους πυρήνες επεξεργαστών αποτελούν το ενσωματωμένο λογισμικό (embedded software) ενός συστήματος. Ο τρόπος με τον οποίο γράφεται το ενσωματωμένο λογισμικό έχει τόσο ομοιότητες όσο και διαφορές με την ανάπτυξη λογισμικού για υπολογιστές γενικού σκοπού. Αφού τα περισσότερα ψηφιακά συστήματα που χρησιμοποιούνται σήμερα είναι ενσωματωμένα συστήματα, το μεγαλύτερο μέρος της πρακτικής εφαρμογής της ψηφιακής σχεδίασης αφορά την ανάπτυξη κυκλωμάτων διασύνδεσης γύρω από τους πυρήνες επεξεργαστών και κυκλωμάτων εξειδικευμένων για εφαρμογές που εκτελούν εργασίες που δεν εκτελούν οι επεξεργαστές.

Τα ενσωματωμένα συστήματα στηρίζουν τη λειτουργία τους σε αισθητήρες, με τους οποίους συλλέγουν πληροφορία από τον περιβάλλον. Τέτοιοι αισθητήρες μετρούν φυσικά μεγέθη όπως θερμοκρασία, πίεση, υγρασία, ταχύτητα, επιτάχυνση. Σε πολλές περιπτώσεις είναι σύνθετοι αισθητήρες, όπως κάμερες ή υπερηχητικοί αισθητήρες απόστασης. Η τεχνολογία επιτρέπει πλέον τη δημιουργία αισθητήρων με τη μορφή σύνθετων ηλεκτρομηχανικών συστημάτων (MEMS-micro-electro-mechanical systems) που αναπτύσσονται σε chip πυριτίου και ενσωματώνουν και κυκλώματα μετατροπής του αναλογικού σήματος σε ψηφιακό ή και μονάδες επεξεργασίας. Τέτοια συστήματα ονομάζονται «έξυπνοι αισθητήρες».



Σχήμα 1.1 : (Εικόνα 1) Αισθητήρας επιτάχυνσης τύπου MEMS.

Τα ενσωματωμένα συστήματα αποτελούν πλέον σημαντικό μέρος των εκπαιδευτικών προγραμμάτων που είναι προσανατολισμένα στην τεχνολογία. Η προσέγγισή τους γίνεται μέσω εκπαιδευτικών μικροελεγκτών, όπως ο Arduino και με εκπαιδευτικά ρομποτικά αναπτύγματα, όπως VEX, Tetrrix, Lego Mindstorms κλπ.



Σχήμα 1.1: (Εικόνα 2) Η εκπαιδευτική διάσταση των ενσωματωμένων συστημάτων

## 1.2 Τι είναι Ενσωματωμένα Συστήματα;

Με τον όρο ενσωματωμένα συστήματα εννοούμε τα συστήματα στα οποία υπάρχει κάποιος επεξεργαστής ο οποίος λειτουργεί σαν μέρος ενός συστήματος, εκτελώντας ένα συγκεκριμένο έργο, και στο οποίο ο χρήστης δεν έχει πρόσβαση για να αλλάξει το πρόγραμμα ή την λειτουργικότητα του συστήματος. Γενικότερα, με τον όρο ενσωματωμένα συστήματα εννοούμε την ενσωμάτωση κάποιου μικροεπεξεργαστή στην λειτουργία ενός ολόκληρου συστήματος με ηλεκτρονικά, μηχανολογικά, και άλλα μέρη.



Σχήμα 1.2: Ένα Ενσωματωμένο Σύστημα

Συνήθως, με τον όρο ενσωματωμένο σύστημα εννοείται ένα υπολογιστικό σύστημα συγκεκριμένης λειτουργίας, το οποίο είναι σχεδιασμένο για μία ή κάποιες εξειδικευμένες εφαρμογές, συχνά με τον περιορισμό της επεξεργασίας σε πραγματικό χρόνο για λόγους ασφάλειας και χρησιμότητας. Ένα τέτοιο σύστημα είναι συνήθως ενσωματωμένο, σαν μέρος μιας συσκευής, η οποία περιλαμβάνει επιπλέον υλικό (hardware), ενίοτε και μηχανικά μέρη. Πρόκειται δηλαδή για μικρά υπολογιστικά μέρη που βρίσκονται εντός μιας μεγαλύτερης συσκευής που εξυπηρετεί έναν πιο γενικό σκοπό.

### **1.3 Ιστορική Αναδρομή Ενσωματωμένων Συστημάτων**

Η ραγδαία ανάπτυξη των υπολογιστικών συστημάτων έχει επηρεάσει κυρίως την κατηγορία των "μικρο-υπολογιστών". Αυτό έχει ως αποτέλεσμα οι διάφορες υποκατηγορίες όπως οι προσωπικοί υπολογιστές (PC) και τα ενσωματωμένα συστήματα να τείνουν όλο και περισσότερο να συγχωνευτούν μεταξύ τους. Έτσι, όσο κυλάει ο χρόνος γίνεται όλο και πιο δύσκολο να ξεχωρίσει κανείς τους προσωπικούς υπολογιστές από τα ενσωματωμένα συστήματα.

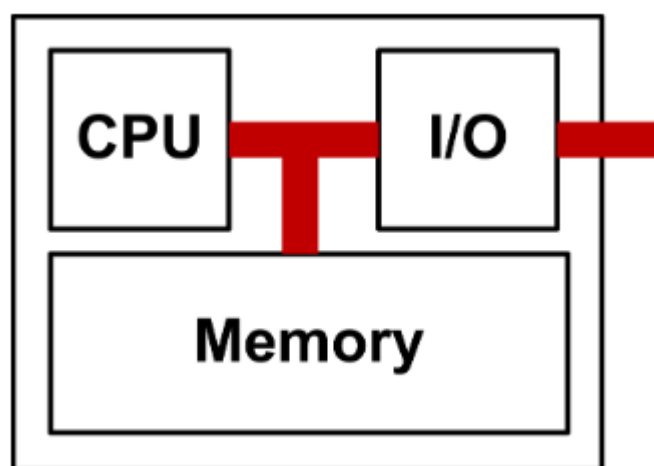
Μέχρι πριν λίγα χρόνια, ενσωματωμένο σύστημα θεωρούταν οποιοδήποτε σύστημα εκτελούσε μία μόνο λειτουργία, όπως για παράδειγμα ο έλεγχος της θερμοκρασίας σε ένα φούρνο, ο έλεγχος τη ταχύτητας ενός κινητήρα κ.τ.λ. Σήμερα όμως, υπάρχουν συσκευές όπως έξυπνα κινητά (smartphones), PDA's (personal digital assistants) κ.α., τα οποία έχουν την δυνατότητα να εκτελούν πολλές λειτουργίες μαζί ανήκοντας όμως στην κατηγορία των ενσωματωμένων συστημάτων. Έχουν παραμείνει σ' αυτή την κατηγορία και δεν έχουν συγχωνευτεί ακόμα με την κατηγορία των προσωπικών υπολογιστών (PC), γιατί τουλάχιστον μέχρι σήμερα, χρησιμοποιούν περιορισμένο υλικό (hardware), μεθόδους μείωσης της κατανάλωσης ενέργειας (λειτουργία με μπαταρίες), καθώς και ειδικό λογισμικό (firmware).

Έτσι, ένα ενσωματωμένο υπολογιστικό σύστημα εξυπηρετεί μία ή περισσότερες εφαρμογές οι οποίες δεν ξεπερνούν σε απαιτήσεις επεξεργαστικής ισχύ έναν προσωπικό υπολογιστή.

### **1.4 Η Κατηγορία των Ενσωματωμένων Συστημάτων**

Οι περισσότεροι χρήστες μπορούν να αναγνωρίσουν ένα υπολογιστικό σύστημα, όπως ένα επιτραπέζιο υπολογιστή, ένα διακομιστή ή ένα φορητό υπολογιστή. Όμως, υπάρχει και ένας άλλος τύπος υπολογιστικών συστημάτων που είναι αθέατος για αυτούς που δεν το γνωρίζουν και είναι ο κόσμος των ενσωματωμένων συστημάτων. Ο κόσμος αυτός είναι πολύ μεγαλύτερος από τον κόσμο των τυπικών υπολογιστικών συστημάτων, αφού κατασκευάζονται ετησίως δισεκατομμύρια μονάδες ενσωματωμένων συστημάτων σε σύγκριση με τις τυπικές υπολογιστικές μονάδες που κυμαίνονται σε κάποια εκατομμύρια. Έχει υπολογισθεί ότι σε ένα τυπικό σύγχρονο σπίτι υπάρχουν 1-2 υπολογιστές και πάνω από 50 ενσωματωμένα συστήματα.

Στα ενσωματωμένα συστήματα κατηγοροποιείται οποιαδήποτε συσκευή η οποία περιλαμβάνει έναν προγραμματιζόμενο επεξεργαστή, ο οποίος δεν είναι ένας επεξεργαστής γενικού σκοπού. Τα ενσωματωμένα συστήματα έχουν την ίδια αρχιτεκτονική με τα τυπικά υπολογιστικά συστήματα. Όπως κάθε προγραμματιζόμενο ψηφιακό σύστημα, έτσι και αυτά έχουν έναν ή περισσότερους επεξεργαστές, μια μνήμη και διεπαφές εισόδου εξόδου. Τα διαφορετικά στοιχεία των ενσωματωμένων συστημάτων είναι ότι χρησιμοποιούν άλλες κατηγορίες επεξεργαστών (συνήθως με χαμηλές δυνατότητες γι' αυτό και ονομάζονται και μικροεπεξεργαστές), με περισσότερες δυνατότητες διασύνδεσης εισόδου/εξόδου και πολύ λιγότερους πόρους (π.χ. μνήμη). Ένα ενσωματωμένο σύστημα μπορεί να θεωρηθεί ως μια μικρογραφία ενός τυπικού υπολογιστικού συστήματος. Οι διαφορετικές απαιτήσεις των ενσωματωμένων συστημάτων έχουν οδηγήσει στη διαφοροποίηση από τα τυπικά υπολογιστικά συστήματα.



Σχήμα 1.4 : Ένα ενσωματωμένο σύστημα, όπως και τα τυπικά υπολογιστικά συστήματα έχουν έναν επεξεργαστή, μία μνήμη και τουλάχιστον έναν τρόπο επικοινωνίας με το περιβάλλον (είσοδο/έξοδο).

Μια άλλη κατηγορία ενσωματωμένων συστημάτων είναι οι επεξεργαστές ψηφιακού σήματος (Digital Signal Processors ή DSP). Αυτά τα κυκλώματα είναι αφιερωμένα στην γρήγορη επεξεργασία σημάτων, συνήθως ήχου και εικόνας. Ο σκοπός τους είναι να επιτελέσουν μέσα σε συγκεκριμένη χρονική προθεσμία τις μαθηματικές πράξεις που απαιτούνται από τις ανάγκες επεξεργασίας του σήματος. Για το σκοπό αυτό, οι DSP επεξεργαστές διαθέτουν κατάλληλη αρχιτεκτονική και αριθμητικές μονάδες ώστε να εκτελούν με ταχύτητα πολλαπλασιασμούς και συσσωρεύσεις (Multiply Accumulate ή MAC). Εκτός από το κατάλληλο data path οι DSP επεξεργαστές διαθέτουν και περιφερειακά κυκλώματα για διακίνηση δεδομένων μεγάλου όγκου και με την κατάλληλη ψηφιακή κωδικοποίηση, σύμφωνα με πρωτόκολλα που χρησιμοποιούνται για δεδομένα ήχου και βίντεο. Οι DSP επεξεργαστές ανήκουν στην κατηγορία των συστημάτων επεξεργασίας σήματος και πολυμέσων.

## 1.5 Τα Ενσωματωμένα Συστήματα ως μέρος της καθημερινότητας

Κατά την τελευταία δεκαετία, τα ενσωματωμένα συστήματα, παρόλο που, αρχικά, είχαν εμφανιστεί ως συσκευές ή προϊόντα χαμηλών επιδόσεων (όπως ρολόγια και ηλεκτρονικοί υπολογιστές τσέπης). Σήμερα χρησιμοποιούνται αρκετά πολύπλοκες συσκευές με αυξημένες απαιτήσεις σε επιδόσεις όπως είναι τα φορητά τερματικά πολυμέσων ή οι ψηφιακοί προσωπικοί βοηθοί. Έτσι, τα ενσωματωμένα συστήματα έχουν γίνει πλέον μέρος της καθημερινής ζωής των περισσότερων ανθρώπων (στις τεχνολογικά αναπτυγμένες κοινωνίες) και μάλιστα, μερικές φορές τα χρησιμοποιούν χωρίς καν να το γνωρίζουν. Γι' αυτό το λόγο, αρκετοί ερευνητές έχουν χαρακτηρίσει τα ενσωματωμένα συστήματα ως επανάσταση των ηλεκτρονικών συσκευών.

Τα ενσωματωμένα συστήματα τα συναντάμε παντού. Αν και οι περισσότεροι από εμάς γνωρίζουμε ότι καθημερινά κατασκευάζονται εκατομμύρια υπολογιστές σε όλον τον κόσμο, αυτό που ίσως δε ξέρουμε είναι ότι κατασκευάζονται πολλά περισσότερα (δισεκατομμύρια) ενσωματωμένα συστήματα για ένα πλήθος διαφορετικών λειτουργιών. Αυτό οφείλεται στο γεγονός ότι τα ενσωματωμένα συστήματα δεν τα βλέπουμε αφού είναι “ενσωματωμένα” στις ψηφιακές συσκευές. Έχει υπολογιστεί ότι στις τεχνολογικά αναπτυγμένες κοινωνίες, αν και μόνον ένα ποσοστό 40% διαθέτει προσωπικό υπολογιστή, κάθε σπίτι έχει τουλάχιστον 30 ενσωματωμένα συστήματα. Συνήθως, τα ενσωματωμένα συστήματα, σε αντίθεση με τους υπολογιστές γενικού σκοπού, χρησιμοποιούνται για εξειδικευμένες εφαρμογές και έχουν περιορισμένη χρήση.

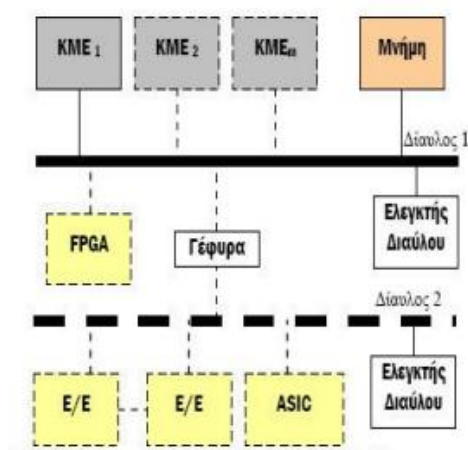
Επομένως, τα ενσωματωμένα συστήματα έχουν γίνει μέρος της σύγχρονης καθημερινότητας, όπως κινητά τηλέφωνα, κονσόλες βίντεο παιχνιδιών, ψηφιακές κάμερες, δέκτες GPS, εκτυπωτές, τηλεοράσεις, τηλεκοντρόλ, φούρνοι μικροκυμάτων, πλυντήρια ρούχων και πιάτων, διάφοροι αυτοματισμοί σπιτιών, όπως φωτισμού, ασφάλειας, κλιματισμού, ήχου και εικόνας, καθώς και δεκάδες άλλα παραδείγματα στον χώρο των παιδικών παιχνιδιών και αλλού. Προκειμένου να κρατηθούν από άποψη κατασκευής απλά και από πλευρά κόστους φτηνά, κατασκευάζονται ως αυτόνομα συστήματα, που βασίζονται σε μικροελεγκτές.

## 1.6 Λειτουργικά Συστήματα Ενσωματωμένων Συστημάτων

Τα Ενσωματωμένα Συστήματα (embedded systems) είναι υπολογιστικά συστήματα ειδικού σκοπού προσαρμοσμένα στο να εξυπηρετήσουν τις ανάγκες των συσκευών της σύγχρονης ζωής όπως κινητά τηλέφωνα, palmtops, ελεγκτές αεροσκαφών και αυτοκινήτων κλπ. Για το λόγο αυτό συνήθως χαρακτηρίζονται από το μικρό τους μέγεθος και τα ιδιαίτερα χαρακτηριστικά τους ως προς την κατανάλωση ισχύος, την απόδοση σε συγκεκριμένες εφαρμογές και το χαμηλό τους κόστος. Ένα «Ενσωματωμένο Σύστημα» (Ε.Σ.) (Embedded System) αποτελεί υπολογιστική μονάδα με αρχιτεκτονική και αρχές λειτουργίας παρόμοιες με αυτές

των συμβατικών υπολογιστών, η οποία ωστόσο προσαρμόζεται στις ανάγκες και απαιτήσεις της εκάστοτε εφαρμογής.

Έτσι, και στην περίπτωση των Ενσωματωμένων Συστημάτων βασικό δομικό στοιχείο αποτελεί ένας μικροεπεξεργαστής, ο οποίος βρίσκεται συνδεδεμένος μέσω μιας ιεραρχίας διαύλων με στοιχεία προσωρινής και μόνιμης αποθήκευσης (μνήμες RAM, EPRROM, Flash, non-Volatile). Παράλληλα, στα Ενσωματωμένα Συστήματα μπορεί να απαντώνται και στοιχεία εξειδικευμένου υλικού τα οποία επικοινωνούν με τα βασικά δομικά στοιχεία και καλούνται να επιτελέσουν συγκεκριμένες εργασίες ανάλογα με τις απαιτήσεις της εκάστοτε εφαρμογής σε απόδοση, κατανάλωση ισχύος, λειτουργίες Εισόδου/Εξόδου κ.α. Τα στοιχεία αυτά υλοποιούνται είτε σε μη προγραμματιζόμενο υλικό (VLSI, ASICs) είτε σε προγραμματιζόμενο υλικό (PLDs, FPGAs) και διασυνδέονται μέσω μιας ιεραρχίας (πιθανώς πολλών επιπέδων) διαύλων με τον μικροεπεξεργαστή και τη μνήμη. Το Σχήμα 1.6 περιγράφει το γενικό διάγραμμα αρχιτεκτονικής δομής Ενσωματωμένου συστήματος.



Σχήμα 1.6: Γενικό διάγραμμα αρχιτεκτονικής δομής Ενσωματωμένου Συστήματος.

Ένα ενσωματωμένο σύστημα είναι ένα οποιοδήποτε υπολογιστικό σύστημα ή συσκευή που εκτελεί μια ειδική λειτουργία ή έχει σχεδιαστεί για μια συγκεκριμένη εφαρμογή. Τα ενσωματωμένα συστήματα μπορεί να χρησιμοποιήσουν ένα λειτουργικό σύστημα βασισμένο στη μνήμη ROM ή ένα σύστημα βασισμένο στο δίσκο, όπως ο υπολογιστής. Αλλά ένα ενσωματωμένο σύστημα δεν είναι χρησιμοποιήσιμο σαν ένα PC.

Για να είναι χρήσιμο ένα ενσωματωμένο λειτουργικό σύστημα πρέπει να ικανοποιεί τα παρακάτω χαρακτηριστικά:

- Να είναι Αρθρωτό
- Να είναι Κλιμακωτό
- Να είναι Διαμορφώσιμο
- Να υποστηρίζει επεξεργαστή
- Να περιέχει οδηγούς σύσκευσης

Βασικά ένα ενσωματωμένο σύστημα χρειάζεται λίγα Kbytes της μνήμης ROM και λίγα της μνήμης RAM. Τυπικά, τα ενσωματωμένα λειτουργικά συστήματα μπορούν να ξεκινήσουν από τη ROM χωρίς την ανάγκη μιας συσκευής αποθήκευσης όπως έναν σκληρό δίσκο. Τα περισσότερα ενσωματωμένα λειτουργικά συστήματα έχουν όλα τα χαρακτηριστικά που έχουν και τα λειτουργικά συστήματα, π.χ. εκκίνηση συστήματος, συστήματα αρχείων, δίκτυο, διαχείριση μνήμης, γραφικά, κ.τ.λ.. Ένα σημαντικό χαρακτηριστικό των ενσωματωμένων συστημάτων είναι ότι γίνεται επιλογή μόνο των απαιτούμενων τμημάτων, έτσι αν το σύστημα δεν έχει γραφικά ή δίκτυο, δεν περιλαμβάνονται τα τμήματα του λειτουργικού συστήματος στο προϊόν. Οι εφαρμογές τους ποικίλλουν. Τα κινητά τηλέφωνα, τα ABS, τα TiVo, και ο Διεθνής διαστημικός σταθμός όλα τρέχουν με ενσωματωμένα λειτουργικά συστήματα.

Στον κόσμο των ενσωματωμένων λειτουργικών συστημάτων, οι πιο σημαντικές ιδιότητες είναι η ενέργεια, η εγκααιρότητα και η αξιοπιστία.

### ➤ **Ενέργεια**

Ειδικά για κινητά ή ανεξάρτητα ενσωματωμένα συστήματα (αλλά και για servers, για κεντρικούς υπολογιστές ή για τους υπερυπολογιστές), η ενέργεια είναι ένας πολύτιμος πόρος. Η κατανάλωση ενέργειας είναι μια μη λειτουργική ιδιότητα του προγραμματισμού των λειτουργιών, όπως αυτού της διαδικασίας, της μνήμης, ή ο I/O προγραμματισμός. Παραδείγματος χάριν, η στρατηγική τοποθέτηση της διαχείρισης της μνήμης μπορεί να έχει έμμεσο έλεγχο στην κατανάλωση της ενέργειας μιας τρέχουσας διαδικασίας εάν το χαρακτηριστικό hardware της (κυρίας) μνήμης, από άποψη απαίτησης ενέργειας και θερμότητας, διαφέρει με τις τράπεζες μνήμης ή τα υποσυστήματα. Προκειμένου να παρέχεται κάποιου είδους ένδειξης της ενέργειας, η εκτίμηση της σε διάφορα σημεία του λογισμικού του συστήματος γίνεται απαραίτητη.

### ➤ **Εγκααιρότητα**

Οι προθεσμίες χαρακτηρίζονται ως μαλακές, σταθερές ή σκληρές. Ανάλογα με το είδος της προθεσμίας, οι μέθοδοι όσον αφορά στην εγγύηση ότι κάποια προθεσμία τηρείται, διαφέρουν. Περιστασιακά, αποτρεπτικά σημεία πρέπει να παρεμβάλλονται σε κρίσιμες διαδρομές εκτέλεσης για να μειωθεί ο προγραμματισμένος μέσος χρόνος μεταξύ εντολής και εκτέλεσης. Η ύπαρξη, η τοποθεσία, καθώς επίσης και η συχνότητα ενός τέτοιου σημείου είναι ένα μη λειτουργικό ζήτημα της αντίστοιχης διαδρομής εκτέλεσης.

### ➤ **Αξιοπιστία**

Αυτή η ιδιότητα αναφέρεται στην αξιοπιστία ενός υπολογιστικού συστήματος η οποία επιτρέπει την δικαιολογημένη τοποθέτηση της εμπιστοσύνης στην υπηρεσία που παραδίδει. Αυτό καλύπτει τις πτυχές της αξιοπιστίας, της διαθεσιμότητας, της ασφάλειας, και της προστασίας. Η προσθήκη πλεονασμού σε ένα σύστημα είναι ένα

μέτρο ώστε να παρέχονται πολλές διαθέσιμες υπηρεσίες. Κατά συνέπεια, η απαίτηση μπορεί να προκύψει για να πολλαπλασιάσει ένα ενιαίο αίτημα συστημάτων και για να αντιμετωπίσει πολλές απαντήσεις.

Δυο από τα πιο διαδεδομένα ενσωματωμένα λειτουργικά συστήματα είναι τα ενσωματωμένα Linux και τα ενσωματωμένα Windows XP.

### 1.6.1 Ενσωματωμένα Windows XP

Τα ενσωματωμένα Windows XP ή αλλιώς XPe, είναι μια έκδοση των Microsoft Windows XP Professional και ο διάδοχος των ενσωματωμένων Windows NT 4.0. Τα XPe είναι βασισμένα στους ίδιους δυαδικούς κώδικες με τα XP Professional, αλλά πωλούνται σε αυτούς που αναπτύσσουν λογισμικό για OEMs, ISVs and IHVs, οι οποίοι θέλουν την πλήρη υποστήριξη του Win32 API των Windows αλλά χωρίς τα έξοδα του Professional. Τρέχει υπάρχουσες εφαρμογές των Windows και οδηγούς συσκευών σε συσκευές με τουλάχιστον 32MB Compact Flash, 32MB RAM και ένα P-200 μικροεπεξεργαστή.

#### ➤ Ιδιότητες

- Στοιχεία του λειτουργικού συστήματος

Τα XPe λειτουργούν με τον ίδιο κώδικα με τα XP Professional, αλλά με τα XPe ένα OEM είναι ελεύθερο να επιλέξει μόνο τα στοιχεία που απαιτούνται και με αυτόν τον τρόπο να μειώσει το ψηφιακό αποτύπωμα του λειτουργικού συστήματος και επίσης να μειώσει την περιοχή επίθεσης καθώς συγκρίνεται με τα XP Professional.

- Φίλτρα εγγραφής

Επίσης τα XPe περιλαμβάνουν χαρακτηριστικά στοιχείων γνωστά ως φίλτρα εγγραφής, τα οποία μπορεί να χρησιμοποιηθούν για να φιλτράρουν δίσκους εγγραφής. Οι τόμοι μπορούν να χαρακτηριστούν ως μόνο για ανάγνωση χρησιμοποιώντας αυτά τα φίλτρα και όλες οι εγγραφές σε αυτό μπορούν να επαναπροσανατολιστούν. Οι εφαρμογές στις ρυθμίσεις του χρήστη δεν είναι ενημερώνονται για αυτά τα φίλτρα εγγραφής.

- Εκκίνηση με USB

Άλλο ένα χαρακτηριστικό είναι ότι τα XPe προσθέτουν USB boot στα windows. Οι ενσωματωμένες XPe συσκευές μπορεί να διαμορφωθούν ώστε να γίνεται η εκκίνηση τους από ένα οδηγό USB.

- Εκκίνηση με CD-ROM

Μια συσκευή XPe μπορεί να διαμορφωθεί ώστε να γίνεται η εκκίνηση της μέσω ενός CD-ROM. Αυτό επιτρέπει στη συσκευή να ξεκινάει χωρίς την απαίτηση ύπαρξης ενός σκληρού δίσκου, παρέχει επίσης μια «φρέσκια εκκίνηση» κάθε φορά



που ξεκινάει η εικόνα (μια ιδιότητα που κληρονομείται από το γεγονός ότι το λειτουργικό σύστημα ξεκινάει από μέσο ανάγνωσης μόνο).

- Εκκίνηση μέσω δικτύου

Μια συσκευή XPe μπορεί επιπλέον να διαμορφωθεί ώστε να ξεκινάει η λειτουργία της από ένα κατάλληλα διαμορφωμένο δίκτυο. Όπως η CD εκκίνηση, η εκκίνηση από το δίκτυο αφαιρεί την απαίτηση ύπαρξης ενός σκληρού δίσκου και παρέχει επίσης μια «φρέσκια εκκίνηση».

## 1.6.2 Το Ενσωματωμένο Linux

Περιέχει τις βασικές λειτουργίες του λειτουργικού συστήματος Linux και χρησιμοποιείται σαν ενσωματωμένο λειτουργικό σύστημα σε υπολογιστικά συστήματα όπως κινητά τηλέφωνα, palms, ipod και σε πολλές άλλες ηλεκτρονικές συσκευές ευρείας κατανάλωσης. Το ενσωματωμένο Linux είναι επίσης κατάλληλο για άλλες ενσωματωμένες εφαρμογές όπως εξοπλισμός δικτύων, έλεγχος μηχανών, σε βιομηχανικούς αυτοματισμούς, σε εξοπλισμούς ναυσιπλοΐας, και ιατρικά όργανα και αλλά.

Το Ενσωματωμένο Linux είναι διαφορετικό από το Λειτουργικό σύστημα Linux που χρησιμοποιούμε στα PC και στις εκδόσεις του Linux για τους servers. Είναι σχεδιασμένο για τις συσκευές που χρησιμοποιούν περιορισμένους πόρους, όπως π.χ. μικρές RAM. Οι βασισμένες στο ενσωματωμένο Linux συσκευές περιλαμβάνουν μνήμες Flash. Είναι μια εναλλακτική λύση από τις συνήθειες γλώσσες σε assembly που γίνονταν κατά παραγγελία για την κάθε συσκευή ή ακόμα και σε γλώσσα C που χρησιμοποιείται κατά ένα μεγάλο μέρος στα ενσωματωμένα συστήματα.

Υπάρχουν πολλά πλεονεκτήματα του ενσωματωμένου Linux έναντι άλλων ενσωματωμένων λειτουργικών συστημάτων όπως ο Πηγαίος κώδικας μπορεί να τροποποιηθεί και να ανακαταμεμηθεί ανάλογα με τις απαιτήσεις της κάθε συσκευής.

Ακόμα η εγκατάσταση του λειτουργικού αυτού συστήματος καταλαμβάνει μικρό χώρο γεγονός πολύ θετικό για τις μικρές ηλεκτρονικές συσκευές, για παράδειγμα μια χαρακτηριστική εγκατάσταση μπορεί να απαιτήσει λιγότερο από δύο 2MB μνήμης. Επιπλέον δεν υπάρχει δαπάνη για χρήση αδειών γεγονός μεγάλης οικονομικής σημασίας.

Τα ενσωματωμένα συστήματα Linux περιέχουν εκτός από το λειτουργικό σύστημα Linux και επιπλέον ελεύθερο λογισμικό. Η GNU βιβλιοθήκη της C που είναι μια τυποποιημένη βιβλιοθήκη C είναι από τις εναλλακτικές λύσεις που χρησιμοποιούν λιγότερους πόρους όπως dietlibc, uClibc ή Newlib.

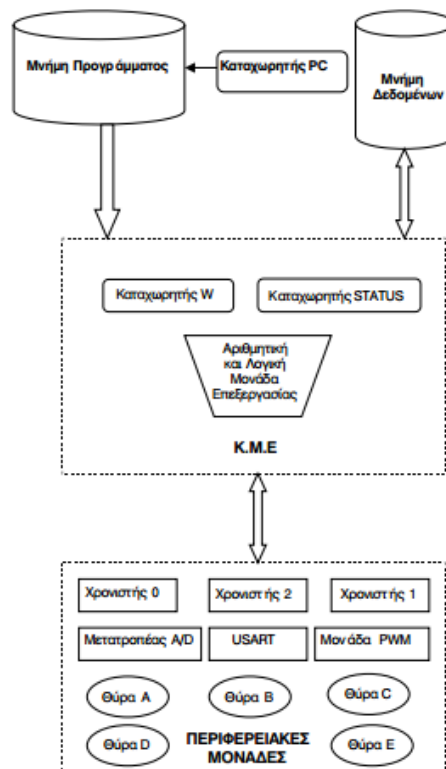
Πολλές βιομηχανίες έχουν διαμορφώσει και χρησιμοποιούν το ενσωματωμένο Linux για ενσωματωμένες εφαρμογές. Το forum CE Linux, που ιδρύθηκε το 2003 για να βοηθήσει στην ανάπτυξη εφαρμογών για τα ενσωματωμένα συστήματα. Ακόμα το Ίδρυμα Linux (αναπτύσσει εφαρμογές για το Linux ανοικτού λογισμικού).

Επιπλέον για την ανάπτυξη εφαρμογών στο Linux εμπλεκεται και το Forum τηλεφωνικών προτύπων που δημιουργήθηκε 2004 για την ανάπτυξη λογισμικών που αφορούν κινητά τηλέφωνα(ή ακόμα παρόμοιες συσκευές με αυτά και τα οποία χρησιμοποιούν το Linux σαν λειτουργικό σύστημα). Το LiMo ίδρυμα, ιδρύθηκε το 2006 από την Motorola, Nec, Panasonic, Samsung, DoCoMo, και Vodafone για να καθιερώσει ένα σύνολο τυποποιημένων τμημάτων του λειτουργικού συστήματος για την βελτίωση της λειτουργίας τη τρίτης γενναίας κινητών τηλεφώνων.

### 1.6.3 Τι είναι η CPU και ποια η Χρησιμοποίηση της?

Η Κεντρική Μονάδα επεξεργασίας (Κ.Μ.Ε.), γνωστή και ως CPU, εκτελεί τις εντολές του προγράμματος που έχουμε αποθηκεύσει σε μία μνήμη, η οποία ονομάζεται μνήμη προγράμματος. Από τη μνήμη αυτή, η κεντρική μονάδα επεξεργασίας φέρνει, με τη σειρά τις εντολές του προγράμματος, τις αποκωδικοποιεί και τις εκτελεί. Εδώ, πρέπει να σημειώσουμε ότι ο PIC αναγνωρίζει τριανταπέντε (35) εντολές.

Μέσα στην Κεντρική Μονάδα Επεξεργασίας, βρίσκεται και η αριθμητική και η λογική μονάδα. Το παρακάτω σχήμα παρουσιάζει την Κ.Μ.Ε. μαζί με τα άμεσα, με αυτήν, συνδεδεμένα στοιχεία του PIC. Οι αριθμητικές πράξεις που μπορεί να εκτελεί είναι η πρόσθεση και η αφαίρεση. Επίσης, έχει τη δυνατότητα να εκτελεί λογικές πράξεις (AND, NOR, XOR κτλ). Η μονάδα επεξεργάζεται δεδομένα μήκους οκτώ δυαδικών ψηφίων (8-bit).



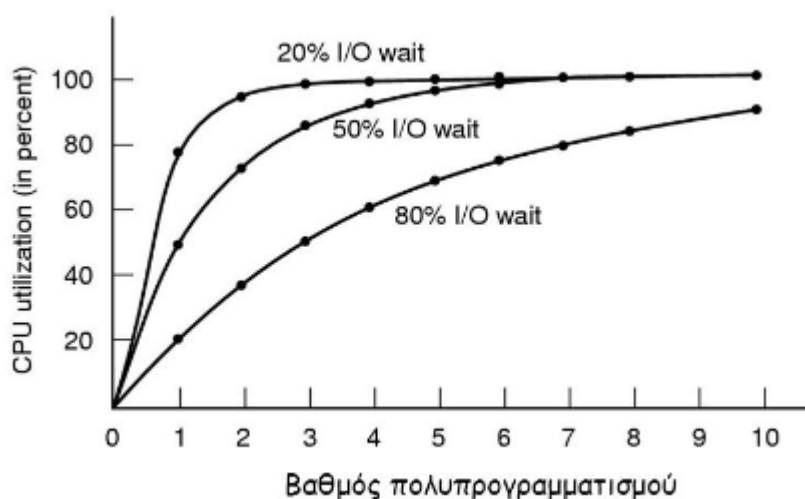
Σχήμα 1.6.3 (Εικόνα 1): Η Δομή μικροελεγκτή PIC

Η Κεντρική Μονάδα Επεξεργασίας - ΚΜΕ (Central Processing Unit - CPU) είναι το κεντρικό εξάρτημα που επεξεργάζεται δεδομένα σε έναν ηλεκτρονικό υπολογιστή, ελέγχει τη λειτουργία του και εκτελεί βασικές λειτουργίες διασύνδεσης και μεταβίβασης εντολών. Αν η ΚΜΕ αποτελείται από ένα μόνο ολοκληρωμένο κύκλωμα τότε ονομάζεται μικροεπεξεργαστής (microprocessor) ή μικροελεγκτής (microcontroller).

Οι επεξεργαστές δεν σχετίζονται αποκλειστικά με τους ηλεκτρονικούς υπολογιστές καθώς πλέον ενσωματώνονται και σε πολλές ηλεκτρονικές συσκευές όπως κινητά τηλέφωνα, ψηφιακές φωτογραφικές μηχανές, βιντεοκάμερες, κονσόλες ηλεκτρονικών παιχνιδιών και άλλα. Για την ακρίβεια, επεξεργαστές ενσωματώνονται σε κάθε είδους συσκευή στην οποία απαιτείται ύπαρξη υπολογιστικής ικανότητας.

#### ➤ Χρησιμοποίηση CPU:

- Ποσοστό της CPU που κάνει χρήσιμη δουλειά.
- Συχνά υπολογίζεται υποθέτοντας μηδενική χρονοπρογραμματιστική επιβάρυνση.



Σχήμα 1.6.3 (Εικόνα 2): Χρήση της CPU ως συνάρτηση του πλήθους διεργασιών στη μνήμη

#### 1.6.4 Τι είναι τα POSIX?

Το POSIX αποτελεί μια οικογένεια προτύπων παροχής κριτηρίων συμμόρφωσης για υπηρεσίες λειτουργικών συστημάτων και είναι σχεδιασμένο να επιτρέπει σε προγράμματα εφαρμογών να γράφουν εφαρμογές που μπορούν εύκολα να μεταφέρονται μεταξύ διαφορετικών λειτουργικών συστημάτων. Προσδιορίζει επίσης έναν καθορισμένο τρόπο για μια εφαρμογή να αλληλεπιδρά με το λειτουργικό σύστημα. Πιο συγκεκριμένα, το πρότυπο POSIX standard προσδιορίζει τις διεπαφές για βασικές λειτουργίες, όπως λειτουργίες αρχείων, διαχείριση εργασιών, σήματα και

συσκευές. Άλλες μετέπειτα εκδόσεις του POSIX προσδιορίζουν επεκτάσεις πραγματικού χρόνου και πολυνηματισμό με αποτέλεσμα τα συστήματα που είναι συμβατά με το POSIX να χρησιμοποιούνται ευρέως στις εφαρμογές πραγματικού χρόνου.

Περιλαμβάνει πάνω από 30 ανεξάρτητα πρότυπα που ποικίλουν από προδιαγραφές για βασικές εφαρμογές λειτουργικών συστημάτων μέχρι και προδιαγραφές για έλεγχο της συμμόρφωσης ενός λειτουργικού συστήματος σε ένα πρότυπο. Παρακάτω θα παρουσιαστούν εν συντομία μερικά από τα πρότυπα του POSIX.

Το POSIX (Portable Operating System Interface for Unix) δημιουργήθηκε για να εξασφαλίσει συμβατότητα στις εφαρμογές μεταξύ των διαφορετικών εκδόσεων. Το POSIX αποτελείται από ένα σύνολο προτύπων που καθορίζονται από το IEEE (Ίδρυμα Μηχανικών Ηλεκτρονικής και Ηλεκτρικής) και τον ISO (Διεθνή Οργανισμό Τυποποιήσεων). Το POSIX καθορίζει τον τρόπο που οι εφαρμογές επικοινωνούν με το λειτουργικό σύστημα.

### **1.7 Ενσωματωμένα Συστήματα Πραγματικού Χρόνου (Real Time)**

Η βασική ιδέα στα συστήματα πραγματικού χρόνου είναι ότι περιμένουμε να ανταποκριθούν στο περιβάλλον τους, την κατάλληλη χρονική στιγμή. Πολλοί άνθρωποι πιστεύουν ότι σε πραγματικό χρόνο σημαίνει παρά πολύ γρήγορα. Αυτό είναι λάθος. Σε πραγματικό χρόνο σημαίνει αρκετά γρήγορα στο περιβάλλον στο οποίο το σύστημα λειτουργεί. Αν μιλάμε για τον υπολογιστή που τρέχει στη μηχανή του αυτοκινήτου μας, αυτό είναι πραγματικά γρήγορο, πρέπει να λάβει αποφάσεις για την ροή του καύσιμου, του χρονισμού ανάφλεξης, κάθε φορά που ο κινητήρας κάνει μια περιστροφή.

Από την άλλη πλευρά, θεωρήστε ότι έχουμε ένα διωλιστήριο χημικό που ελέγχεται από έναν ή περισσότερους υπολογιστές. Το υπολογιστικό σύστημα είναι υπεύθυνο να ελέγχει την διαδικασία και δυσλειτουργίες. Αλλά οι χημικές διεργασίες έχουν μια σταθερά χρόνου από μερικά δευτερόλεπτα έως κάποια λεπτά. Το σύστημα του υπολογιστή θα πρέπει να είναι σε θέση να ανταποκριθεί σε οποιαδήποτε δυσλειτουργία σε επαρκή χρόνο, ώστε να αποφευχθεί μια καταστροφή. Αλλά ας υποθέσουμε ότι ο ηλεκτρονικός υπολογιστής (που ελέγχει την διεργασία) ήταν στη μέση μια εκτύπωσης μιας εκτενής έκθεσης σχετικά με την παραγωγή της περασμένης εβδομάδας όταν και συμβαίνει η βλάβη.

Τα Ενσωματωμένα Συστήματα Πραγματικού Χρόνου περιλαμβάνουν συστήματα που παρακολουθούν αισθητήρες, παίρνουν πολύπλοκες αποφάσεις και ελέγχουν ενεργοποιητές με συνθήκες πραγματικού χρόνου ή συστήματα τα οποία εκτελούν το ίδιο πρόγραμμα για όλη τη διάρκεια της λειτουργίας τους, όπως φορητά συστήματα

κινητής τηλεφωνίας, πλοήγησης (GPS), διασκέδασης (mp3), ψηφιακές φωτογραφικές συσκευές.

Η ουσία του πραγματικού χρόνου στους υπολογιστές, δεν είναι μόνο ότι ανταποκρίνονται αρκετά γρήγορα στο περιβάλλον τους, αλλά τόσο γρήγορα όσο και αξιόπιστα. Ο ελεγκτής ενός χημικού εργοστασίου πρέπει να είναι σε θέση να ανιχνεύσει και αποκριθεί σε μη φυσιολογικές συνθήκες σε επαρκή χρόνο για να αποφευχθεί μια καταστροφή. Αν δεν το κάνει, έχει αποτύχει. Έτσι, η τέχνη του προγραμματισμού και σχεδιασμού των συστημάτων πραγματικού χρόνου, είναι ένα περιβάλλον όπου η αξιοπιστία συναντά τους χρονικούς περιορισμούς αντίδρασης σε ασύγχρονα τυχαία γεγονότα. Καταλαβαίνουμε ότι είναι αρκετά σύνθετο, ο πραγματικός χρόνος σε ένα σύστημα και πόσο γρήγορα θα αποκρίνεται, Όπως υπάρχει και ειδική θεωρία για το σχεδιασμό τέτοιων συστημάτων.

Συχνά, τα ενσωματωμένα συστήματα είναι στενά συνδεδεμένα με το περιβάλλον τους. Αυτό μπορεί να προκαλέσει περιορισμούς πραγματικού χρόνου, οι οποίοι επιβάλλονται από την ανάγκη αλληλεπίδρασης με το περιβάλλον. Περιορισμοί, όπως απαιτούμενες ταχύτητες κίνησης, απαιτούμενη ακρίβεια μετρήσεων και απαιτούμενη χρονική διάρκεια υπαγορεύουν το χρονισμό λειτουργιών λογισμικού. Αν η διαχείριση πολλαπλών δραστηριοτήτων πρέπει να γίνει ταυτόχρονα, τότε επιβάλλονται πιο πολύπλοκοι περιορισμοί πραγματικού χρόνου.

## **1.8 Πολυπύρνα Ενσωματωμένα Συστήματα**

Πολύπλοκα πολυπύρνα ενσωματωμένα συστήματα είναι όλο και πιο διαδομένα σε σύγχρονα προϊόντα μικροηλεκτρονικής. Η ανάπτυξη αυτών των συστημάτων δημιουργεί νέες δυνατότητες για την εξέλιξη σε πόλους τομείς της επιστήμης και της τεχνολογίας. Η βελτίωση της αξιοπιστίας απόδοσης των ενσωματωμένων συστημάτων είναι συνεπώς, πολύ σημαντική για τις όλο και περισσότερο σύνθετες εφαρμογές που αναπτύσσονται σε αυτά.

Τα συστήματα MIMD διακρίνονται σε πολυεπεξεργαστές κοινής μνήμης, όπου πολλαπλοί επεξεργαστές επικοινωνούν με μία κοινή μνήμη ενιαίου χώρου διευθύνσεων, και σε πολυυπολογιστές κατανομημένης μνήμης, όπου πολλαπλά πακέτα επεξεργαστή-ιδιωτικής μνήμης, με τον δικό του χώρο διευθύνσεων το καθένα, διασυνδέονται και επικοινωνούν μεταξύ τους και στις δύο περιπτώσεις η επικοινωνία γίνεται μέσω ενός «δικτύου διασύνδεσης».

Τα μοντέλα παράλληλου προγραμματισμού για πολυεπεξεργαστές και πολυυπολογιστές είναι το μοντέλο κοινού χώρου διευθύνσεων (π.χ. πολλαπλές διεργασίες ή νήματα, OpenMP) και το μοντέλο μεταβίβασης μηνυμάτων (π.χ. PVM, MPI), αντιστοίχως. Στο πρώτο οι επεξεργαστικές μονάδες ανταλλάσσουν πληροφορίες προσπελώνοντας κοινόχρηστες μεταβλητές στην κοινή μνήμη, ενώ στο δεύτερο ανταλλάσσουν μηνύματα. Κάθε προγραμματιστικό μοντέλο μπορεί να εφαρμοστεί και σε σύστημα μιας αρχιτεκτονικής που δεν είναι η φυσική του (π.χ. πολυνηματικό πρόγραμμα σε πολυυπολογιστή ή πρόγραμμα MPI σε πολυεπεξεργαστή) αλλά συνήθως με χαμηλότερες επιδόσεις.

Στους πολυεπεξεργαστές εκτελείται ένα μόνο λειτουργικό σύστημα, το οποίο αναλαμβάνει να αναθέσει την εκτέλεση παράλληλων διεργασιών σε διαφορετικούς επεξεργαστές, ενώ στους πολυπολογιστές, οι οποίοι ουσιαστικά αποτελούν καταναμημένα συστήματα εφόσον είναι εφοδιασμένοι με το κατάλληλο λογισμικό, κάθε κόμβος εκτελεί τον δικό του πυρήνα. Μία υβριδική αρχιτεκτονική, η οποία επιτρέπει εύκολη εκτέλεση προγραμμάτων κοινού χώρου διευθύνσεων σε πολυπολογιστές, είναι η καταναμημένη κοινή μνήμη, στην οποία εξειδικευμένο υλικό (ελεγκτές μνήμης) παρέχει διαφανώς την ψευδαίσθηση ενιαίου χώρου διευθύνσεων στο λογισμικό που εκτελείται σε κάθε κόμβο.

Στα συστήματα αυτά, φυσικά καταναμημένης αλλά λογικά ενιαίας μνήμης, ένας κόμβος προσπελαύνει πολύ γρηγορότερα τα δεδομένα που είναι αποθηκευμένα στην τοπική του μνήμη απ' ό,τι τα δεδομένα που βρίσκονται σε κάποιον απομακρυσμένο κόμβο και γι' αυτό λέγονται συστήματα Ανομοιόμορφης Πρόσβασης Μνήμης (NUMA). Στους υπολογιστές αυτούς συνήθως εκτελείται ένας μόνο πυρήνας λειτουργικού συστήματος ο οποίος γνωρίζει, για λόγους βελτιστοποίησης των επιδόσεων, τις λεπτομέρειες της υποκείμενης αρχιτεκτονικής NUMA.

Στους πολυεπεξεργαστές η μνήμη συνήθως διασπάται σε πολλαπλά τμήματα προκειμένου να επιτρέπονται ταυτόχρονες προσπελάσεις σε αυτά από διαφορετικούς επεξεργαστές (χωρίς αυτό να αναιρεί φυσικά την ύπαρξη ενιαίου χώρου διευθύνσεων), ενώ το δίκτυο διασύνδεσης είναι στην απλούστερη περίπτωση ένας δίαυλος, δηλαδή ένα κοινό μέσο μετάδοσης το οποίο κάθε δεδομένη στιγμή μόνο ένας επεξεργαστής μπορεί να χρησιμοποιεί για επικοινωνία με κάποιο τμήμα μνήμης.

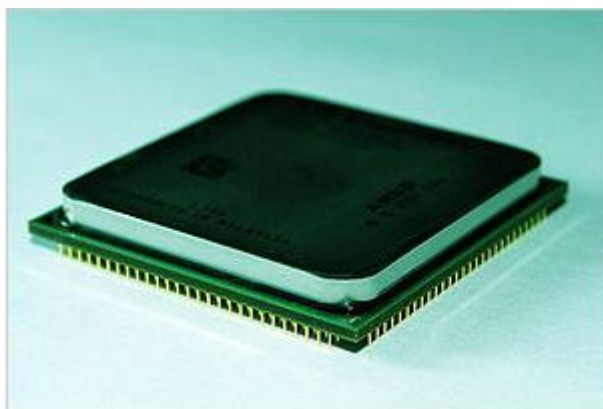
Ως εκ τούτου οι πολυεπεξεργαστές διαύλου δεν παρουσιάζουν επαρκή κλιμακωσιμότητα, καθώς ο δίαυλος δεν διαθέτει αρκετή χωρητικότητα ώστε να εξυπηρετεί ταυτόχρονα αιτήσεις από πολλούς διαφορετικούς επεξεργαστές. Αυτό συνεπάγεται ότι οι επιδόσεις δεν αυξάνονται αναλογικά με το πλήθος των επεξεργαστών, αφού με την προσθήκη νέων επεξεργαστών πολλαπλασιάζονται τα διαστήματα αναμονής του καθενός για κατάληψη του διαύλου. Η λύση του διαύλου πάντως είναι η απλούστερη καθώς οι συνήθεις σειριακοί υπολογιστές έτσι κι αλλιώς διαθέτουν έναν δίαυλο για την επικοινωνία του επεξεργαστή με τη μνήμη και τις μονάδες Εισόδου / Εξόδου. Αν ωστόσο χρειαζόμαστε ένα σύστημα με περισσότερες από 10-20 CPU πρέπει να καταφύγουμε σε δίκτυα μεταγωγής, όπου μεταξύ κάθε επεξεργαστή και κάθε τμήματος της μνήμης δημιουργείται μία μοναδική διαδρομή.

Ανεξάρτητα από το είδος της διασύνδεσης, για την αύξηση των επιδόσεων και την αποφυγή υπερφόρτωσης του δικτύου, κάθε CPU διαθέτει τη δική της τοπική κρυφή μνήμη. Στα παράλληλα συστήματα όμως παρουσιάζεται το λεγόμενο «πρόβλημα της συνεκτικότητας» (cache coherence): ένα κοινόχρηστο δεδομένο που έχει αντιγραφεί σε πολλές κρυφές μνήμες μπορεί να τροποποιηθεί ανεξάρτητα στην καθεμία χωρίς να το γνωρίζει η κύρια ή οι άλλες κρυφές μνήμες, με αποτέλεσμα να υπάρχει ασάφεια ως προς το ποια είναι η σωστή τιμή του κάθε στιγμής.

Το πρόβλημα αυτό επιλύεται στην περίπτωση ύπαρξης διαύλου με «πρωτόκολλα παρακολούθησης», όπου το υλικό κάθε επεξεργαστή παρακολουθεί όλη την πληροφορία που διακινείται στο κοινό μέσο και δρα ανάλογα, ενώ στην περίπτωση δικτύου μεταγωγής με «πρωτόκολλα καταλόγων», στους οποίους κάθε τμήμα της

μνήμης διατηρεί έναν πίνακα με πληροφορίες για το ποιοι επεξεργαστές έχουν κάθε δεδομένο στην κρυφή τους μνήμη ώστε όποτε χρειάζεται να γίνονται οι κατάλληλες ενημερώσεις. Σε κάθε περίπτωση πάντως τα πρωτόκολλα συνεκτικότητας επιβαρύνουν το σύστημα, αφού αυξάνουν την κίνηση στο δίκτυο διασύνδεσης, με το πρόβλημα να μειώνεται (όχι όμως να απαλείφεται) μόνο αν καταβληθεί προσπάθεια ώστε τα παράλληλα προγράμματα να μην τροποποιούν συχνά κοινόχρηστα δεδομένα. Τα παράλληλα συστήματα κοινής μνήμης με δίαυλο στα οποία όλες οι CPU είναι ισότιμες και μπορούν όλες να αναλάβουν την εκτέλεση του λειτουργικού συστήματος ονομάζονται «συμμετρικοί πολυεπεξεργαστές» (SMP).

Πάντως ακόμη και οι πολυεπεξεργαστές μεταγωγής στην πράξη έχουν περιορισμένη κλιμακωσιμότητα λόγω τετραγωνικά αυξανόμενου ως προς το πλήθος των επεξεργαστών κόστους κατασκευής του δικτύου διασύνδεσης, με αποτέλεσμα αν χρειαζόμαστε ένα σύστημα με παραπάνω από 64 CPU συνήθως να καταφεύγουμε σε πολυπολογιστές.



Σχήμα 1.8: Ένας διπύρηνος μικροεπεξεργαστής Athlon 64 X2 6400+. Πρόκειται για έναν πολυεπεξεργαστή διαύλου υλοποιημένο σε ένα μόνο μικροτσίπ.

## 1.9 Κατανεμημένα Ενσωματωμένα Συστήματα

Κατανεμημένα συστήματα είναι τα συστήματα στα οποία το λογισμικό του συστήματος εκτελείται σε ένα σύνολο συνεργαζόμενων επεξεργαστών που συνδέονται με χαλαρή σύζευξη μέσω δικτύου.

### 1.9.1 Χαρακτηριστικά Κατανεμημένων Συστημάτων

- Μερισμός πόρων
  - Κοινή χρήση πόρων υλικού και λογισμικού.
- Ανοιχτή λειτουργία
  - Χρήση εξοπλισμού και λογισμικού από διάφορους κατασκευαστές.
- Ταυτοχρονισμός
  - Ταυτόχρονη επεξεργασία για τη βελτίωση της απόδοσης.
- Επεκτασιμότητα
  - Αυξημένη διεκπεραιωτική ικανότητα με την προσθήκη νέων πόρων.
- Ανοχή ελαττωμάτων

-Η ικανότητα συνέχειας της λειτουργίας αφού ανακύψει κάποιο σφάλμα.

### **1.9.2 Μειονεκτήματα Κατανεμημένων Συστημάτων**

- Πολυπλοκότητα
  - Τα κατανεμημένα συστήματα είναι συνήθως πιο πολύπλοκα από τα συγκεντρωτικά συστήματα.
- Προστασία από εξωτερικούς κινδύνους
  - Πιο ευάλωτα σε εξωτερικές επιθέσεις.
- Διαχειρισιμότητα
  - Απαιτείται μεγαλύτερη προσπάθεια για τη διαχείριση του συστήματος.
- Απρόβλεπτες αποκρίσεις
  - Οι αποκρίσεις ενός κατανεμημένου συστήματος εξαρτώνται από την οργάνωσή του και από το φόρτο του δικτύου.

### **1.9.3 Αρχιτεκτονικές Κατανεμημένων Συστημάτων**

- Αρχιτεκτονικές πελάτη-διακομιστή
  - Κατανεμημένες υπηρεσίες που ζητούν οι εφαρμογές-πελάτες. Οι διακομιστές που παρέχουν τις υπηρεσίες αντιμετωπίζονται διαφορετικά από τους πελάτες που τις χρησιμοποιούν.
- Αρχιτεκτονικές κατανεμημένων αντικειμένων
  - Καμία διάκριση μεταξύ διακομιστών και πελατών. Οποιοδήποτε αντικείμενο του συστήματος μπορεί να παρέχει υπηρεσίες σε άλλα αντικείμενα και να χρησιμοποιεί υπηρεσίες άλλων αντικειμένων.

### **1.10 Χρήση των Ενσωματωμένων Συστημάτων**

Ο όρος Ενσωματωμένο Σύστημα (Embedded System) αναφέρεται στη χρήση ηλεκτρονικών και λογισμικού εντός κάποιου προϊόντος και αποτελεί διαφορετική έννοια από τους υπολογιστές γενικού σκοπού, όπως είναι οι επιτραπέζιοι και οι φορητοί υπολογιστές.

Τα ενσωματωμένα συστήματα αναφέρονται σε συσκευές, όργανα ή μεγάλων τεχνικών έργων / συστημάτων που έχουν κατασκευαστεί για να επεξεργάζονται ένα ή μερικά εκ των προτέρων καθορισμένα καθήκοντα. Ο υπολογιστής που ελέγχει το όλο θέμα είναι ενσωματωμένο ή «ενσωματωμένο» κατά τη συσκευή. Κινητά τηλέφωνα και προσωπικοί ψηφιακοί βοηθοί (PDAs) αποτελούν παραδείγματα των συσκευών με τα ενσωματωμένα συστήματα. Και οι δύο ασκούν ένα συγκεκριμένο αριθμό καθηκόντων που ελέγχονται από ένα ενσωματωμένο στο ηλεκτρονικό σύστημα. Αδρανειακής καθοδήγηση συστήματα που χρησιμοποιούνται για την πλοήγηση αεροφωτογραφίες ή ως μέρος του πακέτου της καθοδήγησης έξυπνες βόμβες και πυραύλους είναι παραδείγματα των μέσων με τα ενσωματωμένα συστήματα. Ένα σύστημα ελέγχου της κυκλοφορίας (η οποία ελέγχει από μακριά φανάρια, χωρίς την ανάγκη για ανθρώπινη παρέμβαση) είναι ένα παράδειγμα ενός μεγάλου μηχανική δομή, με ένα ενσωματωμένο σύστημα. Στην πραγματικότητα, η γραμμή μεταξύ «ενσωματωμένα συστήματα και τα συστήματα γενικής χρήσης (το οποίο μπορεί ή δεν



μπορεί να περιλαμβάνει ένα ενσωματωμένο σύστημα) γίνεται θολή καθώς η τεχνολογία εξελίσσεται.

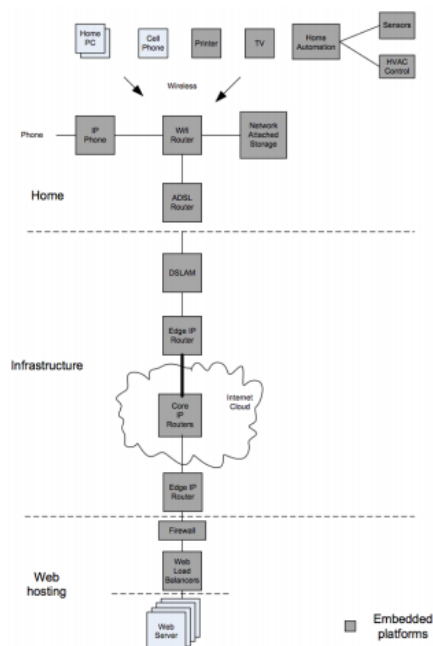
## ΚΕΦΑΛΑΙΟ 2

### 2. ΙΔΙΟΤΗΤΕΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ

Με ένα τέτοιο ευρύ φάσμα από υπολογιστικές πλατφόρμες, με ποικιλία από επιδόσεις και διαφορετικές δυνατότητες σε I/O, σε αυτή την ενότητα θα ορίσουμε κάποια βασικά χαρακτηριστικά των ενσωματωμένων συστημάτων.

Ένα βασικό χαρακτηριστικό όλων των ενσωματωμένων συστημάτων είναι ότι έχουν σχεδιαστεί για να εκτελούν μια συγκεκριμένη εργασία ή λειτουργία. Το λογισμικό που αναπτύσσετε για την εκάστοτε πλατφόρμα είναι συγκεκριμένο για τη λειτουργία συνολικά της συσκευής. Σε πολλές περιπτώσεις, οι υπολογιστικές δυνατότητες της συσκευής δεν είναι ορατές στον τελικό χρήστη, η λειτουργία που παρέχει η συσκευή είναι αυτό που έχει σημασία. Σε αντίθεση με το ταχέως αναπτυσσόμενο οικοσύστημα εφαρμογών (applications) και την ανάπτυξη τους για έξυπνα κινητά τηλέφωνα (smart phones), οι τελικοί χρήστες αυτών των συσκευών, γενικά δεν εγκαθιστούν πολλές εφαρμογές στα κινητά τους τηλέφωνα (αν και έχουν πολυάριθμες εφαρμογές, για να αναβαθμίσουν τις δυνατότητες του λογισμικού).

Τα ενσωματωμένα συστήματα μπορεί να τα συναντήσουμε σε ένα μεγάλο πλήθος από συσκευές και προϊόντα. Όπως φαίνεται στο Σχήμα πολλές καταναλωτικές συσκευές είναι ενσωματωμένες με υπολογιστικά συστήματα, όπως τηλεόραση μέσω ίντερνετ (Internet TV), συσκευή αναπαραγωγής βίντεο (Blue-ray Player), κονσόλες παιχνιδιών, κινητά τηλέφωνα, οι εκτυπωτές είναι μερικά μόνο από τις συσκευές που χαρακτηρίζονται από ένα αρκετά υψηλό επίπεδο απόδοσης.



Σχήμα 2 : Γενική ροή διασυνδέσεις των συσκευών στο διαδίκτυο

## 2.1 Χαρακτηριστικά Ενσωματωμένων Συστημάτων

Το βασικό χαρακτηριστικό ενός ενσωματωμένου συστήματος είναι ότι είναι να χειρίζονται μερικά απλά καθήκοντα, αν και τα βήματα που εμπλέκονται στη διακίνηση ή πραγματοποιώντας το καθήκον μπορεί να είναι τόσο περίπλοκο όπως κάθε προγράμματος ηλεκτρονικού υπολογιστή. Ένα videogame ελεγκτή, για παράδειγμα, μπορεί να λεχθεί και να έχουν απλά καθήκοντα - φορτίο και το παιχνίδι επιτρέπει στον παίκτη να ελέγχει μέσω εντολών που εγγράφονται μέσω του ακουστικού. Στην πραγματικότητα, όμως, ένα παιχνίδι ελεγκτή (ιδίως τα νεότερα παιχνίδια που κατασκευάζονται για το X-box ή PS3) περνά μέσα από μια σειρά από ενέργειες και δράσεις που χρειάζονται τόσο πολύ επεξεργαστική ισχύ ως μεμονωμένο υπολογιστή.

Μεταξύ των χαρακτηριστικών της σύγχρονων ενσωματωμένων συστημάτων είναι τα εξής: Αρχικά, ένα ενσωματωμένο σύστημα δεν είχε κανένα χρήστη - προγράμματα ενημέρωσης και είχαν ήδη ενσωματωθεί στο σύστημα (για παράδειγμα, το σύστημα καθοδήγησης για Intercontinental βαλλιστικών πυραύλων ή ICBM) και δεν υπήρχε ανάγκη για την ανθρώπινη αλληλεπίδραση ή παρέμβαση εκτός από την εγκατάσταση της συσκευής και δοκιμή αυτό.

Πολλά σύγχρονα ενσωματωμένα συστήματα ωστόσο, να έχουν πλήρη κλίμακα των διεπαφών χρηστών, αν και αυτές είναι μόνο για τις εισροές των δεδομένων, αλλά δεν πρέπει να παρέχει πρόσθετη λειτουργικότητα του συστήματος, π.χ. QWERTY πληκτρολόγια για τα PDA που έχει χρησιμοποιηθεί για ονόματα, διευθύνσεις, αριθμούς τηλεφώνου και τις σημειώσεις, ακόμη και η πλήρης μεγέθους των εγγράφων. Η στιγμή PDAs επιτύχουν την πλήρη επιφάνεια εργασίας του υπολογιστή λειτουργιών, ωστόσο, δεν μπορούν πλέον να θεωρούνται ενσωματωμένα συστήματα. Αρχικά, αυτό που αναφέρεται σε βασικά συστήματα όπως οι διακόπτες, οι μικρές χαρακτήρα-ή-ψηφία και εμφανίζει μόνο τα LED που προορίζονται για να δείτε την “υγεία” του ενσωματωμένου συστήματος αλλά αυτό έχει επίσης επιτύχει κάποιο επίπεδο πολυπλοκότητας.

Η ταμειακή μηχανή ή ένα ATM με οθόνη αφής τεχνολογία θεωρείται ένα ενσωματωμένο σύστημα, δεδομένου ότι έχει περιορισμένες χρήσεις, ακόμα και αν το χρήστη (η οθόνη αφής) είναι ένα πολύπλοκο σύστημα. Σε περιορισμένη έκταση αποτελεί το κλειδί για τον καθορισμό αυτών, όπως τα ενσωματωμένα συστήματα. Υπό μία έννοια τα BIOS chip είναι ένα ενσωματωμένο σύστημα, δεδομένου ότι έχει περιορισμένες αρμοδιότητες, καθώς και έργα αυτόματα (όταν ο υπολογιστής είναι να εκκινήσετε). Περιφερειακά όταν το USB μπορεί επίσης να θεωρηθεί ως το ενσωματωμένο σύστημα.

Τα απλά ενσωματωμένα συστήματα που διαθέτουν απλή λειτουργικότητα μπορούν να ελέγχονται από προγράμματα ειδικού σκοπού ή από σύνολο προγραμμάτων, χωρίς τη συνδρομή άλλου λογισμικού. Συνήθως, τα πιο πολύπλοκα ενσωματωμένα συστήματα είναι δυνατό να χρησιμοποιήσουν λειτουργικό σύστημα

γενικού σκοπού, όπως το LINUX, οι περιορισμοί του χώρου μνήμης, η κατανάλωση ισχύος και οι απαιτήσεις πραγματικού χρόνου συνήθως υπαγορεύουν τη χρήση λειτουργικού συστήματος ειδικού σκοπού, που έχει σχεδιαστεί για το περιβάλλον του ενσωματωμένου συστήματος.

Τα ακόλουθα αποτελούν μερικά από τα μοναδικά χαρακτηριστικά και τις μοναδικές απαιτήσεις σχεδίασης των ενσωματωμένων λειτουργικών συστημάτων:

- **Λειτουργία πραγματικού χρόνου (Real-time operation):** Σε αρκετά ενσωματωμένα συστήματα, η ορθότητα ενός υπολογισμού εξαρτάται εν μέρει από τη χρονική στιγμή που αυτός αποδίδεται. Συχνά, περιορισμοί πραγματικού χρόνου υπαγορεύονται από απαιτήσεις εξωτερικής E/E, καθώς και απαιτήσεις σταθερότητας ελέγχου.
- **Λειτουργία αντίδρασης (Reactive operation):** Το ενσωματωμένο λογισμικό μπορεί να εκτελείται ως αποτέλεσμα εξωτερικών γεγονότων. Αν τα γεγονότα αυτά δε συμβαίνουν περιοδικά ή σε προβλεπόμενα χρονικά διαστήματα, τότε το ενσωματωμένο λογισμικό μπορεί να χρειάζεται να λαμβάνει υπόψη συνθήκες χειρότερης περίπτωσης και να θέτει προτεραιότητες εκτέλεσης ρουτινών.
- **Δυνατότητα ρύθμισης (Configurability):** Εξαιτίας της υπερπληθώρας ενσωματωμένων συστημάτων, υπάρχουν μεγάλες αποκλίσεις απαιτήσεων, τόσο ποιοτικές όσο και ποσοτικές, όσων αφορά τη λειτουργικότητα των ενσωματωμένων λειτουργικών συστημάτων. Άρα τα ενσωματωμένα λειτουργικά συστήματα που προορίζονται για χρήση σε πληθώρα ενσωματωμένων συστημάτων πρέπει να διαθέτουν δυνατότητα ευέλικτης ρύθμισης, έτσι ώστε να παρέχεται μόνο η λειτουργικότητα που χρειάζεται ένας συγκεκριμένος συνδυασμός εφαρμογής και υλικού.
- **Ευελιξία συσκευής E/E (I/O device flexibility):** Δεν υπάρχει απολύτως καμία συσκευή που πρέπει να υποστηρίζεται από όλες τις εκδόσεις του λειτουργικού συστήματος και το πλήθος των συσκευών E/E είναι μεγάλο.
- **Λιτοί μηχανισμοί προστασίας (Streamlined protection mechanisms):** Συνήθως, τα ενσωματωμένα συστήματα σχεδιάζονται για να παρέχουν περιορισμένη και ορθά καθορισμένη λειτουργικότητα. Προγράμματα που δεν έχουν ελεγχθεί σπάνια προστίθεται στο λογισμικό. Αφότου το λογισμικό έχει ελεγχθεί και ρυθμιστεί, μπορεί να θεωρείται ότι είναι αξιόπιστο. Συνεπώς, εκτός από τα μέτρα ασφαλείας, τα ενσωματωμένα λειτουργικά συστήματα διαθέτουν περιορισμένους μηχανισμούς προστασίας.
- **Άμεση χρήση των διακοπών (Direct use of interrupts):** Λειτουργικά συστήματα γενικού σκοπού συνήθως δεν επιτρέπονται στις διεργασίες χρήστη να χρησιμοποιούν τις διακοπές άμεσα.

Υπάρχουν δύο γενικές προσεγγίσεις για την ανάπτυξη των ενσωματωμένων λειτουργικών συστημάτων. Η πρώτη προσέγγιση αφορά τη χρησιμοποίηση υφιστάμενου λειτουργικού συστήματος και την ανάλογη προσαρμογή του, ώστε να ανταποκρίνεται στις απαιτήσεις της ενσωματωμένης εφαρμογής. Η άλλη

προσέγγιση αφορά το σχεδιασμό και την υλοποίηση λειτουργικού συστήματος που προορίζεται μόνο για ενσωματωμένη χρήση.

## 2.2 Ιδιαιτερότητες των Ενσωματωμένων Συστημάτων

Ένας λόγος που εξηγεί γιατί τα ενσωματωμένα συστήματα είναι τόσο δημοφιλή, είναι το ότι αποτελούνται από υλικό υψηλών επιδόσεων σε συνδυασμό με ένα εξειδικευμένο λογισμικό, ένας συνδυασμός που έχει πολύ μικρό χρόνο σχεδιασμού και εισόδου στην αγορά. Η λέξη «ενσωματωμένο» υποδηλώνει ότι το σύστημα είναι ενσωματωμένο σε μια άλλη συσκευή και ότι η λειτουργία του δε θα τροποποιηθεί από τη στιγμή που θα γίνει διαθέσιμο στην αγορά.

Είναι σαφές ότι υπάρχει μια δυσκολία στη διατύπωση ενός ορισμού που θα κάλυπτε όλες αυτές τις περιπτώσεις. Το πλήθος των ορισμών για τα ενσωματωμένα συστήματα είναι ίσως ανάλογο του αριθμού των σχεδιαστών που ασχολούνται με αυτά. Η λέξη ενσωματωμένο υποδηλώνει ότι το σύστημα είναι 'ενσωματωμένο' σε μια άλλη συσκευή και ότι η λειτουργία του δε θα τροποποιηθεί από τη στιγμή που θα γίνει διαθέσιμο στην αγορά.

Ένας πιο αυστηρός ορισμός είναι ότι τα ενσωματωμένα συστήματα είναι εξειδικευμένα συστήματα υπολογιστών, τα οποία είναι αφοσιωμένα στην εκτέλεση μιας συγκεκριμένης λειτουργίας. Είναι συνδυασμός υλικού και λογισμικού μέρους. Μάλιστα, είναι κοινή πρακτική για τα ενσωματωμένα συστήματα η τοποθέτηση πάνω στην ίδια επιφάνεια πυριτίου επεξεργαστικών πυρήνων, μνημών, περιφερειακών, διασυνδέσεων εισόδου/εξόδου και μερικές φορές αναλογικών/ψηφιακών κυκλωμάτων, καταλήγοντας σε ολόκληρα συστήματα πάνω στο ίδιο υλικό πυριτίου (System-on-Chip, SOC). Συνήθως, τα ενσωματωμένα συστήματα είναι τμήμα ενός μεγαλύτερου συστήματος ή προϊόντος.

Πάντως, υπάρχουν μερικά σαφή χαρακτηριστικά σχετικά με τα ενσωματωμένα συστήματα. Έτσι, τα ενσωματωμένα συστήματα έχουν μια συγκεκριμένη λειτουργία. Ένα ενσωματωμένο σύστημα σχεδιάζεται για μια μόνο λειτουργία και την εκτελεί αδιαλείπτως. Γι αυτό και μερικές φορές ονομάζεται και 'αποκλειστικό' σύστημα. Ένα ενσωματωμένο σύστημα δεν μπορεί να χρησιμοποιηθεί για λειτουργίες διαφορετικές από αυτές που έχει σχεδιαστεί να επιτελέσει. Για παράδειγμα, δε θα μπορούσαμε να χρησιμοποιήσουμε τον ενσωματωμένο επεξεργαστή ενός έξυπνου πλυντηρίου, για να αναπαράγουμε πολυμέσα.

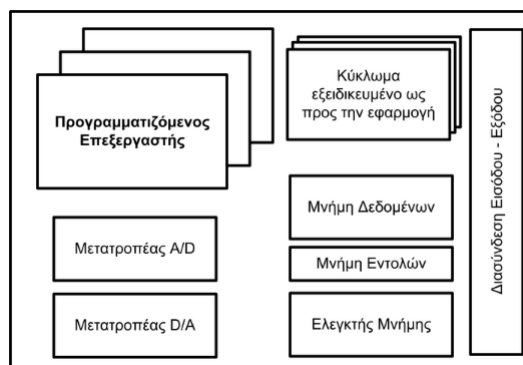
Επίσης, τα ενσωματωμένα συστήματα έχουν αυξημένους περιορισμούς. Αν και κάθε υπολογιστικό σύστημα έχει περιορισμούς στο σχεδιασμό, τα ενσωματωμένα συστήματα πρέπει να έχουν όσο το δυνατόν μικρότερο κόστος σχεδιασμού (αφού κατασκευάζονται σε μεγάλες ποσότητες και πρέπει να είναι φθηνά, ώστε να τα προτιμήσουν οι καταναλωτές), να έχουν υψηλές επιδόσεις (ώστε να ανταποκρίνονται αμέσως στις εντολές του χρήστη και να παρουσιάζουν μεγάλη διαδραστικότητα με αυτόν), να έχουν μικρό μέγεθος (αφού συχνά βρίσκονται σε φορητές συσκευές), να έχουν μικρή κατανάλωση ενέργειας (αφού από τη μια πρέπει να λειτουργούν

συνέχεια και από την άλλη να μην απαιτείται κάποια μέθοδος ψύξης), και να έχουν μεγάλη αξιοπιστία (δηλαδή να έχουν απρόσκοπτη λειτουργία).

Ακόμη, θα μπορούσαμε να πούμε ότι τα ενσωματωμένα συστήματα βασίζονται σε ένα πλήθος επεξεργαστών (από διάφορες εταιρείες) και αρχιτεκτονικών. Έτσι, ενώ ένας προσωπικός υπολογιστής συνήθως αποτελείται από επεξεργαστές της εταιρείας AMD, ή της INTEL, τα ενσωματωμένα συστήματα συναντώνται με ένα πλήθος επεξεργαστών. Ένα ακόμη χαρακτηριστικό των ενσωματωμένων συστημάτων είναι το λειτουργικό που έχουν, που σχεδόν πάντα είναι ένα λειτουργικό πραγματικού χρόνου (Real Time Operating System - RTOS).

Σε αντίθεση με τα κλασσικά λειτουργικά συστήματα (π.χ. Windows, Linux, FreeBSD) που συναντώνται σε προσωπικούς υπολογιστές, τα λειτουργικά των ενσωματωμένων συστημάτων λαμβάνουν πάντα σοβαρά υπόψη τις προτεραιότητες των εφαρμογών, και δίνουν όλη την επεξεργαστική ισχύ τους σε εφαρμογές με υψηλή προτεραιότητα. Ο σχεδιαστής είναι υπεύθυνος για να δώσει (ή να αφαιρέσει) προτεραιότητες στις εφαρμογές του συστήματος. Αν το κάνει σωστά, τότε τα λειτουργικά συστήματα των ενσωματωμένων αποδεικνύονται πολύ πιο σταθερά και αξιόπιστα, σε σύγκριση με τα λειτουργικά των προσωπικών υπολογιστών. Μια άλλη διαφοροποίηση είναι ότι το λειτουργικό σύστημα των ενσωματωμένων συστημάτων έχει πολύ μικρό μέγεθος (μερικά KB) και για αυτό τον λόγο δεν είναι και φιλικό προς τον χρήστη (user friendly) ή τον προγραμματιστή. Τέλος, το λειτουργικό σύστημα και οι εφαρμογές των ενσωματωμένων συστημάτων βρίσκονται τοποθετημένα σε μνήμη ROM, που ονομάζεται και μνήμη εντολών.

Αντιθέτως, το λειτουργικό σύστημα ενός προσωπικού υπολογιστή, ή ενός διακομιστή, βρίσκεται σε κάποια αποθηκευτική μονάδα δίσκου (λόγω του υπερβολικά μεγάλου μεγέθους που έχει) και είναι πολύ εύκολο να τροποποιηθεί. Για να αλλάξει το λογισμικό του ενσωματωμένου, πρέπει να αντικατασταθεί η μνήμη ROM με μια άλλη που έχει το καινούργιο πρόγραμμα, μια διαδικασία που απαιτεί εξειδικευμένο προσωπικό. Ασφαλώς, τα τελευταία χρόνια φαίνεται πως τροποποιείται το τοπίο ως προς τη μνήμη, και έτσι πολλά Ενσωματωμένα Συστήματα έχουν αντί για μνήμη ROM μνήμη FLASH ή EEPROM, η οποία επιτρέπει την εύκολη αναβάθμιση του συστήματος σε περίπτωση ανάγκης.



Σχήμα 2.2 : Ένα τυπικό ενσωματωμένο σύστημα αποτελείται από διακριτά δομοστοιχεία.

## 2.3 Μεθοδολογία Ανάπτυξης των Ενσωματωμένων Συστημάτων

Παρότι δεν υπάρχει «τυπικός τρόπος» ανάπτυξης ενσωματωμένων συστημάτων, υπάρχουν κάποιες γενικές αρχές. Συστήματα τα οποία βασίζονται στην απόλυτη εκμετάλλευση πόρων από μικρής υπολογιστικής ισχύος μικροελεγκτές (π.χ. 8-bit) συχνά αναπτύσσονται σε γλώσσες assembly. Η προσέγγιση αυτή γίνεται όλο και σπανιότερη διότι ακόμη και σε 8-bit μικροελεγκτές υπάρχουν compilers που κάνουν πάρα πολύ καλή δουλειά, και επιτρέπουν την ανάπτυξη σε υψηλές γλώσσες προγραμματισμού όπως η C. Όσο περισσότερο ισχυρός είναι ο χρησιμοποιούμενος επεξεργαστής τόσο η ανάπτυξη μοιάζει με αυτήν σε συμβατικές αρχιτεκτονικές. Οι διεργασίες τρέχουν υποβοηθούμενες από λειτουργικά συστήματα πραγματικού χρόνου, και η δυνατότητα συντήρησης και εξέλιξης του κώδικα οδηγεί ακόμη και σε ανάπτυξη με γλώσσες όπως η C++ ή η Java.

Επειδή όμως σε κάποιες από αυτές τις περιπτώσεις ο κώδικας δεν έχει ιδιαίτερα καλά χαρακτηριστικά όπως καλή ταχύτητα και κατανάλωση ισχύος, κρίσιμα μέρη του προγράμματος συνεχίζουν να αναπτύσσονται σε γλώσσες όπως C και assembly. Αυθαίρετο παράδειγμα είναι ότι ενδεχόμενα σε ένα κινητό τηλέφωνο η διεπαφή με τον χρήστη μπορεί να είναι σε Java αλλά ένα φίλτρο που κάνει αναίρεση ήχους και θορύβου να είναι αναπτυγμένα σε C. Εν γένει τα περιβάλλοντα ανάπτυξης δίνουν ολοκληρωμένες λύσεις για συγκεκριμένη κατηγορία επεξεργαστών. Αυτό περιλαμβάνει τόσο τα περιβάλλοντα λογισμικού όσο και την υποστήριξη του υπό ανάπτυξη συστήματος με προσομοιωτές και εντός κυκλώματος προσομοιωτές.

Τα περιβάλλοντα ανάπτυξης περιλαμβάνουν:

- **Cross compilers:** οι cross compilers λειτουργούν όπως και οι compilers αλλά βγάζουν κώδικα assembly για διαφορετική αρχιτεκτονική από αυτήν στην οποία τρέχουν οι ίδιοι. Για παράδειγμα μπορεί να έχουμε έναν C compiler που να τρέχει σε αρχιτεκτονική Intel x86 (π.χ. σε PC) και ο οποίος βγάζει κώδικα για αρχιτεκτονική Atmel AVR.
- **Cross assemblers:** Ίδια με τους cross compilers αλλά παίρνουν σαν είσοδο γλώσσα assembly αρχιτεκτονικής άλλης από αυτήν στην οποία τρέχουν και βγάζουν εκτελέσιμο κώδικα αντίστοιχο με αυτόν της assembly που δέχονται. Για παράδειγμα, όταν σε προσωπικό υπολογιστή PC τρέχουμε assembler για αρχιτεκτονική MIPS, αυτός είναι cross assembler.
- **Εντός κυκλώματος προσομοιωτές (in-circuit emulators):** Είναι προσομοιωτές που όχι μόνο δίνουν μία λειτουργική προσομοίωση ενός επεξεργαστή αλλά και αναπαράγουν τα χαρακτηριστικά του σε επίπεδο ακροδεκτών, δίνουν δηλαδή σε ένα συνδετήρα (connector) τα λειτουργικά χαρακτηριστικά κάποιου επεξεργαστή (π.χ. Intel 8086) αλλά με τις επί πλέον ιδιότητες του να βλέπει το σύστημα ανάπτυξης θέματα λειτουργίας όπως τα περιεχόμενα των καταχωρητών και της μνήμης. Επί πλέον ο χρήστης μπορεί να εισάγει στον κώδικα σημεία παρακολούθησης (breakpoints) να δημιουργήσει σήματα αντίστοιχα εξωτερικών σημάτων (όπως

διακοπές – interrupts) και να αλλάξει κάποια στοιχεία, π.χ. περιεχόμενα καταχωρητών. Οι εντός κυκλώματος καταχωρητές δεν λειτουργούν πάντοτε στην ίδια μέγιστη ταχύτητα όπως οι επεξεργαστές που προσομοιώνουν.

### **2.3.1 Ανάπτυξη Ενσωματωμένων Συστημάτων**

Τα προϊόντα που βασίζονται σε μικροελεγκτή ονομάζονται ενσωματωμένα συστήματα και δημιουργούνται μέσα από μια μεθοδική διεργασία, η οποία περιλαμβάνει κάποια συγκεκριμένα βήματα ή φάσεις. Τα βήματα αυτά είναι σημαντικά, και στον βαθμό που ακολουθούνται πιστά, οδηγούν με ασφάλεια στην παραγωγή ενός επιτυχημένου προϊόντος.

Η πλειοψηφία των ψηφιακών συστημάτων που σχεδιάζονται και κατασκευάζονται σήμερα είναι ενσωματωμένα συστήματα (embedded systems) στα οποία μεγάλο τμήμα της επεξεργαστικής εργασίας γίνεται από έναν ή περισσότερους υπολογιστές που αποτελούν μέρος του συστήματος. Στην πραγματικότητα, η τεράστια πλειοψηφία των υπολογιστών που χρησιμοποιούνται σήμερα βρίσκονται μέσα σε ενσωματωμένα συστήματα, και όχι σε προσωπικούς υπολογιστές και άλλα συστήματα γενικού σκοπού.

Οι πρώτοι υπολογιστές ήταν μεγάλα συστήματα από μόνοι τους, και σπάνια θεωρούνταν συστατικά στοιχεία μεγαλύτερων συστημάτων. Ωστόσο, με την ανάπτυξη της τεχνολογίας, ειδικότερα της τεχνολογίας των ολοκληρωμένων κυκλωμάτων, έγινε πρακτικά εφικτό να ενσωματωθούν μικροί υπολογιστές σαν συστατικά ενός κυκλώματος και να προγραμματισθούν ώστε να υλοποιούν μέρος της λειτουργικότητας του κυκλώματος. Οι ενσωματωμένοι υπολογιστές συνήθως δεν έχουν την ίδια μορφή με τους υπολογιστές γενικού σκοπού, όπως οι επιτραπέζιοι και οι φορητοί προσωπικοί υπολογιστές.

Αντίθετα, ένας ενσωματωμένος υπολογιστής αποτελείται από έναν πυρήνα επεξεργαστή (processor core), μαζί με στοιχεία μνήμης για την αποθήκευση του προγράμματος και των δεδομένων για το πρόγραμμα που θα εκτελεστεί στον πυρήνα επεξεργαστή, και άλλα συστατικά στοιχεία για τη μεταφορά δεδομένων μεταξύ του πυρήνα του επεξεργαστή και του υπολοίπου συστήματος. Τα προγράμματα που εκτελούνται στους πυρήνες επεξεργαστών αποτελούν το ενσωματωμένο λογισμικό (embedded software) ενός συστήματος. Ο τρόπος με τον οποίο γράφεται το ενσωματωμένο λογισμικό έχει τόσο ομοιότητες όσο και διαφορές με την ανάπτυξη λογισμικού για υπολογιστές γενικού σκοπού. Είναι μια ευρύτερη περιοχή από μόνη της και εκτός του σκοπού αυτού του βιβλίου. Ωστόσο, επειδή στο βιβλίο αυτό ασχολούμαστε με ενσωματωμένα συστήματα, χρειάζεται να ασχληθούμε και με το ενσωματωμένο λογισμικό τουλάχιστον σε ένα βασικό επίπεδο. Θα επανέλθουμε στο αντικείμενο αυτό σαν μέρος της συζήτησής μας για τη διασύνδεση (interfacing) με τους ενσωματωμένους πυρήνες επεξεργαστών.

Αφού τα περισσότερα ψηφιακά συστήματα που χρησιμοποιούνται σήμερα είναι ενσωματωμένα συστήματα, το μεγαλύτερο μέρος της πρακτικής εφαρμογής της



ψηφιακής σχεδίασης αφορά την ανάπτυξη κυκλωμάτων διασύνδεσης γύρω από τους πυρήνες επεξεργαστών και κυκλωμάτων εξειδικευμένων για εφαρμογές που εκτελούν εργασίες που δεν εκτελούν οι επεξεργαστές.

#### **2.4 Φάσεις Ανάπτυξης ενός ενσωματωμένου συστήματος**

Οι φάσεις που πρέπει να ακολουθούνται, όταν γίνεται η ανάπτυξη ενός ενσωματωμένου συστήματος, είναι οι παρακάτω:

- Φάση ορισμού του προβλήματος
- Φάση σχεδίασης
- Φάση ορισμού των ελέγχων
- Φάση κατασκευής και ελέγχου του πρωτοτύπου
- Φάση ολοκλήρωσης του συστήματος και ελέγχου του λογισμικού
- Φάση ελέγχου του συστήματος

#### **2.5 Γλώσσα Assembly**

Η γλώσσα Assembly είναι στενά συνδεδεμένη και εξαρτάται από την αρχιτεκτονική του επεξεργαστή καθώς και από το σετ εντολών του. Θα θυμόμαστε ότι έχουμε 2 τύπους από επεξεργαστές τους CISC και τους RISC. Ο κώδικας της γλώσσας αυτής θα πρέπει να επεξεργάζεται, ώστε να μετατρέπεται σε γλώσσα μηχανής. Τη γλώσσα μηχανής είναι που καταλαβαίνει το σύστημα. Το πρόγραμμα που κάνει αυτή τη μεταφράσει ονομάζεται Assembler. Η παραπάνω δυσκολία ξεπεράστηκε με το να χρησιμοποιείται για κάθε μία εντολή της γλώσσας μηχανής ένα συμβολικό όνομα, προερχόμενο από την αγγλική γλώσσα. Οι εντολές του 8051, γραμμένες με αυτά τα συμβολικά ονόματα, συνιστούν τη συμβολική γλώσσα ή γλώσσα assembly του 8051.

Ενώ στις γλώσσες ανωτέρου επιπέδου (π. χ. BASIC, C κ.λπ.) κάθε εντολή αυτών αντιστοιχεί σε αρκετές έως πολλές εντολές της γλώσσας μηχανής, οι εντολές της γλώσσας assembly αντιστοιχούν μία προς μία με τις εντολές της γλώσσας μηχανής. Πρόκειται για την ίδια γλώσσα, αλλά σε μορφή πολύ κατανοητή και εύχρηστη από τον χρήστη. Στη γλώσσα assembly, η εντολή που μεταφέρει στον συσσωρευτή A το περιεχόμενο του καταχωρητή R0 γράφεται ως MOV A,R0.

Επίσης, η εντολή πρόσθεσης στον A του περιεχομένου της θέσης μνήμης RAM που έχει διεύθυνση 29H γράφεται ως ADD A,29H. Η ευκολία που μας παρέχει η χρήση της γλώσσας assembly είναι εμφανής. Εμείς θα γράφουμε ένα πρόγραμμα στη γλώσσα assembly, χρησιμοποιώντας έναν κειμενογράφο (editor), το δίνουμε ως είσοδο στον συμβολομεταφραστή ή assembler, και αυτός μας παρέχει το ίδιο πρόγραμμα μεταφρασμένο σε γλώσσα μηχανής. Σε μια πρακτική εφαρμογή, το πρόγραμμα μηχανής θα το αποθηκεύουμε στη μνήμη EPROM χρησιμοποιώντας κατάλληλο προγραμματιστή (programmer) για το chip του 8051.

## 2.6 Γλώσσα C/C++

Περίπου το 1972 ο Dennis Ritchie στα εργαστήρια Bell, με αφετηρία την Β επινοεί την γλώσσα προγραμματισμού C με σκοπό να ξαναγράψει το Λειτουργικό Σύστημα Unix. Βασική καινοτομία στην γλώσσα C υπήρξε η ανάπτυξη και χρήση νέων τύπων δεδομένων. Ο στόχος της δημιουργίας της γλώσσας επέβαλλε τον εφοδιασμό της γλώσσας με εργαλεία, που θα επέτρεπαν τον προγραμματισμό σε επίπεδο μηχανής.

Η απλότητα της γλώσσας C και η αποτελεσματικότητα, όσον αφορά το χρόνο εκτέλεσης προγραμμάτων, κατέστησε την γλώσσα C σαν την πιο δυναμική γλώσσα προγραμματισμού. Αρχικά θεωρήθηκε σαν η γλώσσα προγραμματισμού λειτουργικών συστημάτων, γρήγορα όμως έγινε αντιληπτό ότι, είναι μια γενικής χρήσεως γλώσσα προγραμματισμού και γνώρισε τεράστια επιτυχία σε εφαρμογές πάσης φύσεως (από εμπορικές μέχρι εξελιγμένες επιστημονικές εφαρμογές). Αντικατέστησε σε πολλές περιπτώσεις τον επίπονο προγραμματισμό σε συμβολική γλώσσα προγραμματισμού (Assembly). Αρχικά τα πρώτα standards της C αποτέλεσαν τα προσδιοριζόμενα στο βασικό βιβλίο "The C Programming Language" των Brian Kernigham και Dennis Ritchie, που εκδόθηκε το 1978. Το 1983 συστάθηκε επιτροπή για ANSI standards της γλώσσας C, που ανακοίνωσε τα αποτελέσματα της το 1989.

Το 1980 επινοείται από τον Bjarne Stroustrup η γλώσσα C++. Η C++ διατηρεί όλα τα χαρακτηριστικά της C και επιπλέον προσφέρει την δυνατότητα ανάπτυξης προγραμμάτων με αντικειμενοστρεφή προγραμματισμό (Object Oriented Programming).

### 2.6.1 Γενικά Χαρακτηριστικά της C++

Σαν γενικά χαρακτηριστικά της C++ μπορούμε να εντοπίσουμε:

1) Η C++ είναι μια επέκταση της C .

Κάθε πρόγραμμα της C κατά κανόνα μπορεί να μεταγλωττιστεί στην C++, (με εξαίρεση μερικών μη ουσιαστικών τροποποιήσεων). Επομένως η C++ έχει τα γενικά χαρακτηριστικά της C όπως:

a) Συνδυάζει ιδιότητες γλωσσών υψηλού επιπέδου (ορίζονται συναρτήσεις για πολύπλοκες δραστηριότητες) και χαμηλού επιπέδου (προσπέλαση σε bits, bytes, addresses, ports).

b) Είναι δομημένη γλώσσα προγραμματισμού. Οι εντολές προσδιορίζονται σε μπλοκ (με τοπικές μεταβλητές), διατίθενται εντολές ανακυκλώσεων (do, while) και επιτρέπονται εσοχές (idents) κατά την γραφή των εντολών του προγράμματος.

c) Δεν είναι κατά μπλοκ δομημένη γλώσσα προγραμματισμού (όχι block structure). Δεν επιτρέπει ορισμό μιας συνάρτησης μέσα σε μια άλλη.

2) Η C++ υποστηρίζει Αντικειμενοστραφή Προγραμματισμό (Object Oriented Programming). Δηλαδή διαθέτει χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού όπως:

a) Η ενθυλάκωση (encapsulation). Τα δεδομένα μαζί με συναρτήσεις που τα αφορούν δένονται σε αντικείμενα (objects). Τα μέλη των αντικειμένων (δεδομένα ή συναρτήσεις) μπορεί να αποκρύβονται από εξωτερικές επιδράσεις.

b) Ο πολυμορφισμός (polymorphism), Διαφορετικές δράσεις (συναρτήσεις) αντιμετωπίζονται με τον ίδιο τρόπο (έχουν το ίδιο όνομα).

c) Η κληρονομικότητα (Inheritance) Αντικείμενα, που δημιουργούνται με βάση άλλα αντικείμενα, "κληρονομούν" (διαθέτουν) τα δεδομένα και τις συναρτήσεις των αντικειμένων από τα οποία δημιουργούνται.

3) Η C++ προσφέρεται για δημιουργία μεγάλων και πολύπλοκων προγραμμάτων. Ενδεικτικά αναφέρουμε ότι, η C++ χρησιμοποιείται σε δημιουργία Λειτουργικών Συστημάτων (Operating systems), Διερμηνευτών (Interpreters), Μεταγλωττιστών (Compilers), Επεξεργαστών κειμένου (Word Processors), Συστημάτων διαχείρισης Βάσεων Δεδομένων (DBMS), Πακέτων επεξεργασίας λογιστικών φύλλων (Spreadsheets) κ.α.

## 2.6.2 Το σύστημα της C++

Οι εντολές που δίνουμε είναι συμβολικά ονόματα κάποιων "οδηγιών" προς το μηχάνημα. Οι οδηγίες αυτές υποκρύπτουν ενότητες εντολών (ονομάζονται προγράμματα ή ρουτίνες), που έχουν γραφεί από ειδικούς με στόχο να ερμηνεύουν και εκτελούν τις συμβολικές εντολές μας.

Οι εντολές πλέον δεν είναι από το Λειτουργικό Σύστημα, αλλά από ένα προκαθορισμένο σύνολο συμβολικών εντολών μιας γλώσσας προγραμματισμού και με μια αυστηρά προκαθορισμένη σύνταξη (που αφορά την γλώσσα.). Ο προγραμματιστής βοηθούμενος από το Λειτουργικό Σύστημα δημιουργεί "ενότητες εντολών" (προγράμματα) σε αποθηκευτικές περιοχές "μνήμης.

Ειδικά προγράμματα (που έχουν γραφεί από τους επινοητές της γλώσσας προγραμματισμού), οι μεταγλωττιστές (compilers) αναλαμβάνουν, (όταν τα καλέσουμε προς εκτέλεση από το Λειτουργικό Σύστημα) να μεταφράσουν (μεταγλωττίσουν) τα "πηγαία προγράμματα" σε ενδιάμεσα προγράμματα, από τα οποία τελικά θα δημιουργηθούν εκτελέσιμα προγράμματα (executable programs). Ο προγραμματιστής για την δημιουργία του πηγαίου προγράμματος χρησιμοποιεί κάποιο πρόγραμμα, που ονομάζεται συντάκτης κειμένου (editor).

Το πηγαίο πρόγραμμα που είναι κατανοητό στον προγραμματιστή, δημιουργείται σε αρχείο που έχει την κατάληξη .cpp και στη συνέχεια μεταφράζεται (μεταγλωττίζεται) σε "γλώσσα μηχανής". Την μεταγλώττιση του πηγαίου

προγράμματος εκτελεί το εκτελέσιμο πρόγραμμα του συστήματος της C++, που ονομάζεται μεταγλωττιστής (compiler).

## 2.7 Παραδείγματα Ενσωματωμένων Συστημάτων

Μερικά παραδείγματα ενσωματωμένων συστημάτων είναι:

- **Τηλεοράσεις και Βίντεο:** Οι τηλεοράσεις έχουν ενσωματωμένους επεξεργαστές για να ελέγχουν την εικόνα, να ρυθμίζουν τα κανάλια, να εκτυπώνουν μηνύματα πάνω στην εικόνα, να απενεργοποιούν και να ενεργοποιούν τα κυκλώματα της τηλεόρασης.
- **Συσκευές τηλε-ελέγχου:** Οι συσκευές τηλε-ελέγχου(telecontrol) έχουν ενσωματωμένους επεξεργαστές για να μετατρέπουν τις εντολές του χρήστη σε σήματα υπερέθρων για να ελέγχουν άλλες συσκευές.
- **Κινητά/Ασύρματα Τηλέφωνα:** Τα κινητά τηλέφωνα έχουν ενσωματωμένους επεξεργαστές για να διεκπεραιώνουν τις διαδικασίες αποκωδικοποίησης/κωδικοποίησης της φωνής, για να υλοποιούν το πρωτόκολλο επικοινωνίας και να εκτελούν τις πολύπλοκες εφαρμογές του τηλεφώνου.
- **Οχήματα:** Όλα τα σύγχρονα αυτοκίνητα έχουν ένα πλήθος από ενσωματωμένους επεξεργαστές, είτε για την προστασία των επιβατών, είτε για βοηθητικές ενδείξεις, είτε για τη βελτίωση των συνθηκών οδήγησης.
- **Συσκευές Νοικοκυριού:** Οι σύγχρονες συσκευές νοικοκυριού (φούρνοι,πλυντήρια, ψυγεία κ.α.) έχουν ενσωματωμένους επεξεργαστές για να προστατεύουν τα τρόφιμα ή τα ρούχα από λανθασμένες ρυθμίσεις για να βελτιώσουν τις διαδικασίες συντήρησης ή μαγειρέματος και γενικά να αποτελούν τους βοηθούς της νοικοκυράς.
- **Αυτόματοι Πωλητές:** Οι αυτόματοι πωλητές έχουν ενσωματωμένους επεξεργαστές για να διεκπεραιώσουν τη λειτουργία της πώλησης αγαθών.
- **Εκτυπωτές/Φαξ:** Οι συσκευές αυτές έχουν ενσωματωμένα συστήματα για τον έλεγχο της εκτύπωσης, της μετατροπής των δεδομένων σε εντολές προς τους μηχανισμούς εκτύπωσης και της διασύνδεσης με άλλα συστήματα.
- **Φωτογραφικές Μηχανές/Μηχανές Βιντεολήψης:** Και αυτές οι συσκευές έχουν ένα ή παραπάνω ενσωματωμένα συστήματα για να ρυθμίζουν αυτόματα κάποια χαρακτηριστικά ή να αποθηκεύουν με βέλτιστο τρόπο τα δεδομένα που λαμβάνουν.



Σχήμα 2.7: Παραδείγματα Ενσωματωμένων Συστημάτων

### 2.7.1. Αναλυτικά Παραδείγματα Ενσωματωμένων Συστημάτων

Σε αυτή την ενότητα θα παρουσιάσουμε δύο σχεδιαστικά παραδείγματα για να γίνουν κατανοητές οι ανάγκες ελέγχου και υπολογισμών που διέπουν τα ενσωματωμένα συστήματα, με διάφορους βαθμούς πολυπλοκότητας. Ένα απλό Ενσωματωμένο σύστημα που βρίσκεται σε μια επαγωγική κουζίνα και ένα υψηλής πολυπλοκότητας Ενσωματωμένο Σύστημα που βρίσκεται μέσα σε ένα αυτοκίνητο.

#### ➤ Το Ενσωματωμένο Σύστημα για μια επαγωγική κουζίνα

Σε μια σύγχρονη οικεία, κάποιος μπορεί να διακρίνει ενσωματωμένα συστήματα σε πλήθος συσκευών. Μία από αυτές τις συσκευές είναι η επαγωγή εστία. Στο παρελθόν χρησιμοποιούνταν κεραμικές κουζίνες με περιστροφικούς διακόπτες, οι οποίες ασφαλώς δεν ήταν ενσωματωμένα συστήματα. Η ρύθμιση της θερμοκρασίας γίνονταν με κάποιο διμεταλλικό έλασμα που μόλις ζεσταίνονταν σε κάποια θερμοκρασία, έκοβε το ρεύμα που διέρρεε από την αντίσταση. Ο περιστροφικός διακόπτης ρύθμιζε την απόσταση από το διμεταλλικό έλασμα και την επαφή στην αντίσταση. Δεν υπήρχε κάποιο πρόγραμμα ή κάποια παραπάνω ανάγκη. Προκειμένου οι κατασκευαστές να βελτιώσουν το σχεδιασμό της κουζίνας και να προσθέσουν στυλ, αλλά ταυτόχρονα να δώσουν και παραπάνω δυνατότητες ξανασχεδίασαν το σύστημα με τη χρήση ενός ενσωματωμένου συστήματος.

Οι δυνατότητες που παρέχει το νέο ενσωματωμένο σύστημα:

- Τα κουμπιά δεν είναι περιστροφικά, αλλά αισθητήρια απόστασης τα οποία κρύβονται κάτω από το γυαλί της εστίας και εκπέμπουν μια υπεριώδη ακτινοβολία, όπου ανιχνεύουν την αντανάκλαση. Ο χρήστης αρκεί να τοποθετήσει το δάχτυλο πάνω από το αισθητήριο και μέσω της αντανάκλασης στο δάχτυλο του θα ανιχνευθεί η επιλογή. Αντικαταστάθηκαν

λοιπόν, οι περιστροφικοί διακόπτες με αυτούς τους οπτικούς αισθητήρες ανάκλασης, εμείς θα τους καλούμε διακόπτες ανάκλασης ή απλώς διακόπτες.

- Επιλογή του «ματιού» της κουζίνας που θα θερμανθεί.
- Επιλογή θερμοκρασίας μέσω κουμπιών αύξησης ή μείωσης (από 0 που είναι σβηστό, έως 9 που είναι το μέγιστο), για το συγκεκριμένο μάτι.
- Ηχητική ένδειξη για κάθε επιλογή με beeper (κάθε φορά που ο χρήστης επιλέγει κάτι).
- Φωτεινή ένδειξη με οθόνες 7 στοιχείων για κάθε μάτι που δείχνει τον αριθμό που έχει επιλεγεί, ενώ επίσης χρησιμοποιείται η τελεία για να σηματοδοτεί το μάτι που έχει επιλεγεί.
- Ρύθμιση της θερμοκρασίας μέσω PWM (διαμόρφωση πλάτους παλμού), όπου αναλόγως της επιλογής της θερμοκρασίας διαμορφώνεται κατάλληλα ο κύκλος ενεργοποίησης και απενεργοποίησης. Για παράδειγμα για θερμοκρασία στο επίπεδο 1, σημαίνει ότι η αντίσταση στο μάτι θα είναι 10% ενεργοποιημένη και 90% απενεργοποιημένη, ενώ στη μέγιστη βαθμίδα 9, θα είναι 100% ενεργοποιημένη.
- Κλείδωμα των επιλογών μέσω ειδικού διακόπτη, όπου απαιτείται παρατεταμένη επιλογή του κατάλληλου διακόπτη για να αρθεί το κλείδωμα. Το κλείδωμα χρησιμοποιείται για να ακυρώνει τις λειτουργίες των άλλων διακοπών, ώστε να μην πατηθούν κατά λάθος (π.χ. αν αφήσει κάποιος κάτι στην κουζίνα)

Οι απαιτήσεις του προγράμματος δε συνοδεύονται από αυστηρούς χρονικούς περιορισμούς (δεν απαιτείται να ενεργοποιηθεί η αντίσταση μέσα σε XX ns, αν ενεργοποιηθεί λίγο αργότερα δεν υπάρχει πρόβλημα), ούτε από ενεργειακούς περιορισμούς (η εστία τροφοδοτείται από εναλλασσόμενο ρεύμα 230V, και το 99% καταναλώνεται στις αντιστάσεις των ματιών), ούτε από θέμα κόστους (το κόστος των ηλεκτρονικών στοιχείων είναι αμελητέο ως προς την υπόλοιπη κατασκευή). Γενικά, ο σχεδιασμός ενός ενσωματωμένου συστήματος για αυτό το σύστημα είναι αρκετά απλός και εύκολος. Η μόνη απαίτηση είναι ο μικροεπεξεργαστής να έχει αρκετές εισόδους/εξόδους για να καλύψει τις ανάγκες.



Σχήμα 2.7.1 (Εικόνα 1) : Μια σύγχρονη επαγωγική εστία φέρει ένα ενσωματωμένο σύστημα.

Επειδή οι ανάγκες είναι αρκετά απλές , συνήθως οι κατασκευαστές επιλέγουν τον πιο απλό επεξεργαστή, έναν 8bit στα 16 Mhz (συγκρίνετε τους επεξεργαστές στους επιτραπέζιους ή φορητούς υπολογιστές που διαχειρίζονται 64 bit στα 1.5 με 2 Ghz). Ο επεξεργαστής θα πρέπει να έχει εσωτερική μνήμη FLASH για να μπορεί να διατηρεί το πρόγραμμα (32KB μνήμη FLASH είναι υπεραρκετά). Η μνήμη θα μπορούσε να είναι και εξωτερική, αλλά δεν υπάρχει διαφορά στο κόστος και προτιμάται ο επεξεργαστής να έχει εσωτερική μνήμη. Η μνήμη RAM του μικροεπεξεργαστή αρκεί να είναι κάποια Byte (π.χ. 1 ή 2KB είναι αρκετά).

Ως προς τους ακροδέκτες του μικροεπεξεργαστή, κάθε οθόνη 7 στοιχείων μαζί με την τελεία, απαιτεί 8 εξόδους, το beeper απαιτεί 1 έξοδο, κάθε διακόπτης απαιτεί 1 είσοδο, και κάθε αντίσταση έχει ένα ρελέ που ελέγχει τη ροή του ρεύματος, οπότε συνολικά ο μικροεπεξεργαστής για 2 μάτια πρέπει να υποστηρίξει  $2*8 + 1 + 2$  εξόδους και 6 εισόδους, συνολικά 23. Μάλιστα, για να μειωθούν οι απαιτήσεις σε ακροδέκτες, μπορούν να χρησιμοποιηθούν 2 των 8bit εξωτερικοί σειριακοί καταχωρητές παράλληλης εξόδου συνδεδεμένοι σε σειρά, που χρησιμοποιούν 3 ακροδέκτες από το μικροεπεξεργαστή (και όχι  $2*8$ ), οπότε οι συνολικές ανάγκες του συστήματος είναι  $3 + 1 + 2 + 6 = 12$  ακροδέκτες.

Εκτός από τις παραπάνω συνδέσεις, θα απαιτηθούν και οι κατάλληλες αντιστάσεις pull-up ή pull-down, ο σχεδιασμός του PCB (τυπωμένη πλακέτα κυκλώματος), η δρομολόγηση των ασθενών και ισχυρών ρευμάτων και η κατάλληλη θωράκιση από παρεμβολές για να είναι σύμφωνη με το πρότυπο της ηλεκτρομαγνητικής θωράκισης. Ασφαλώς δε χρειάζεται λειτουργικό σύστημα το ενσωματωμένο σύστημα, και αρκεί να γραφεί ένα πρόγραμμα είτε σε γλώσσα μηχανής, είτε στην πιο φιλική C, όπως συνηθίζεται. Το πρόγραμμα θα αποτελείται από αέναο βρόχο που θα κάνει διαδοχική σταθμοσκόπηση των επιλογών και διακοπών, θα ενημερώνει κατάλληλα τις οθόνες 7 στοιχείων, θα ενεργοποιεί το beeper, και θα διαμορφώνει μέσω PWM τη θερμοκρασία.

Ο κατασκευαστής της κουζίνας μπορεί να παρέχει για την ίδια κουζίνα πολλαπλές δυνατότητες. Εκτός από τα παραπάνω που είναι το βασικό μοντέλο, θα μπορούσε να προσθέσει ένα δομοστοιχείο Bluetooth (που απαιτεί δύο ακροδέκτες RX/TX, για σειριακή επικοινωνία με το μικροεπεξεργαστή), και να δίνει μια εφαρμογή για έξυπνα κινητά, ώστε η ρύθμιση της θερμοκρασίας να γίνεται ακόμη με πιο φιλικό τρόπο ή θα μπορούσε να τοποθετήσει ένα δομοστοιχείο επικοινωνίας wifi 802.11 για σύνδεση στο διαδίκτυο, ώστε να στέλνει ενημερώσεις με email ή να υπάρχει έλεγχος από κεντρικό website. Ασφαλώς, το κάθε μοντέλο επαγωγικής κουζίνας θα έχει και την ανάλογη τιμή, που συνήθως είναι πολλαπλάσια από το κόστος του υλικού (π.χ. ένα δομοστοιχείο Bluetooth κοστίζει 5 ευρώ, ενώ ο

κατασκευαστής της κουζίνας μπορεί να χρεώνει αυτό το μοντέλο 150 ευρώ παραπάνω από το βασικό).

### ➤ Το Ενσωματωμένο Σύστημα για έναν ψηφιακό καταγραφέα

Ένα αρκετά δημοφιλές Ενσωματωμένο Σύστημα είναι ο ψηφιακός καταγραφέας υψηλής ανάλυσης (High Definition DVR), ο οποίος τοποθετείται με μια βεντούζα στο παρμπρίζ του αυτοκινήτου και καταγράφει τη διαδρομή. Αυτή η καταγραφή μπορεί να χρησιμοποιηθεί ως αποδεικτικό στοιχείο σε περίπτωση τρακαρίσματος, για να καθοριστεί ο υπαίτιος. Σε σύγκριση με το προηγούμενο Ενσωματωμένο Σύστημα, αυτό έχει υψηλότερες απαιτήσεις, όπως φαίνονται από τη λίστα που ακολουθεί:

- Αποθήκευση VIDEO σε υψηλή ανάλυση σε κάρτα Micro SD.
- Συμπίεση VIDEO κατά H.234 σε πραγματικό χρόνο για μείωση των απαιτήσεων χώρου.
- Χαμηλή κατανάλωση ενέργειας (έως 1 A στα 12V, όσο παρέχει η σύνδεση του εσωτερικού αναπτήρα αυτοκινήτου).
- Πλήθος κουμπιών ελέγχου.
- Ρολόι πραγματικού χρόνου με διατήρηση της ημερομηνίας.
- Εσωτερική μπαταρία LiPO με κύκλωμα φόρτισης μέσω mini USB.
- Μενού με ποικίλες ρυθμίσεις ποιότητας και διαχείρισης της κάρτας SD.
- Μετατροπéας Αναλογικού σε Ψηφιακό, για την διασύνδεση με τον οπτικό αισθητήρα CMOS ή CCD.
- Υποστήριξη LCD (liquid crystal display) για την εμφάνιση του video και των ρυθμίσεων.
- Δημιουργία συμβατού συστήματος αρχείου FAT στην κάρτα SD, ώστε να μπορεί να διαβαστεί από ένα PC
- Υποστήριξη για USB Mass Storage Device όταν συνδέεται σε ένα PC
- Πολύ μικρό μέγεθος, ώστε να μην επηρεάζει το οπτικό πεδίο του χρήστη (αφού τοποθετείται στο μπροστινό παρμπρίζ).



Σχήμα 2.7.2 (Εικόνα 2) : Ένας ψηφιακός καταγραφέας αυτοκινήτου υψηλής ανάλυσης αποτελεί μεσαίας πολυπλοκότητας Ενσωματωμένου Συστήματος.



Όπως φαίνεται από τις απαιτήσεις, το σύστημα αυτό είναι πιο πολύπλοκο από το απλό Ενσωματωμένο Σύστημα που χρησιμοποιείται σε μια επαγωγική κουζίνα. Η πιο σημαντική απαίτηση είναι η ύπαρξη αυστηρών περιορισμών χρόνου και συγκεκριμένα η ανάγνωση μιας εικόνας υψηλής ανάλυσης (1080p, δηλαδή 1920x1080 στα 3x8=24 bit), σε ρυθμό 25 FPS (frames per second, εικονοστοιχεία το δευτερόλεπτο), συμπίεση και αποθήκευση σε ένα μέσο αποθήκευσης. Ο επεξεργαστής θα πρέπει να είναι 32bit (διαφορετικά με 8bit θα υπήρχε μεγάλη καθυστέρηση) και για να επιτευχθεί η χαμηλή κατανάλωση και ο μικρός όγκος, θα πρέπει να ενσωματώνει τις περισσότερες λειτουργίες πάνω στο ίδιο ολοκληρωμένο κύκλωμα (SoC, System On Chip - σύστημα πάνω στο ίδιο chip). Θα πρέπει να χρησιμοποιηθεί είτε ένας δυνατός 32bit επεξεργαστής, συνήθως της εταιρίας ARM, ή ένας άλλος 32bit επεξεργαστής, με έναν επιταχυντή υλικού για τη λειτουργία της συμπίεσης H.234 σε πραγματικό χρόνο. Θα πρέπει το ίδιο SoC να υποστηρίζει USB για mass storage device (υποστήριξη σύνδεσης σε υπολογιστή ως δίσκο), και να έχει έναν ελεγκτή για μνήμες SD.

Για να διατηρείται η ώρα, θα πρέπει να υπάρχει ένα κύκλωμα RTC (real time clock, ρολόι πραγματικού χρόνου) στο ίδιο το SoC (ή εκτός), το οποίο θα τροφοδοτείται από συγκεκριμένο ακροδέκτη του επεξεργαστή από τη μπαταρία της συσκευής και θα συνδέεται μέσω ενός διαύλου με τον επεξεργαστή. Κάθε φορά που ο επεξεργαστής ζητάει να ενημερωθεί για την τρέχουσα ώρα (π.χ. κατά την ενεργοποίηση του συστήματος), θα δίνεται αυτή η ενημέρωση, όπως επίσης θα υπάρχει δυνατότητα της ρύθμισης του RTC σε μια συγκεκριμένη ώρα. Το SoC θα πρέπει να έχει και τον ελεγκτή για το LCD, ενώ τα κουμπιά θα πρέπει να συνδεθούν είτε κατευθείαν σε ακροδέκτες του SoC, είτε σε ένα ξεχωριστό περιφερειακό I/O, που συνδέεται με τον επεξεργαστή. Επίσης, απαιτούνται πυκνωτές και αντιστάσεις για το κύκλωμα φόρτισης και προστασίας της μπαταρίας, όπως και για τη διασύνδεση των περιφερειακών.

Αυτό το Ενσωματωμένο Σύστημα μπορεί να υλοποιηθεί είτε αποκλειστικά σε C χωρίς λειτουργικό σύστημα, το οποίο είναι αρκετά πολύπλοκο αλλά όχι αδύνατο, αφού θα πρέπει να υλοποιήσει ο σχεδιαστής το λογισμικό οδήγησης για κάθε ελεγκτή (LCD, H.234, CCD/CMOS, κ.α.), να σχεδιάσει τη γραφική διεπαφή χρήστη και να υλοποιήσει το λογισμικό τμήμα του USB, για τη διασύνδεση με τον υπολογιστή, ώστε να φαίνεται ως μια συσκευή USB Mass Storage Device. Αυτή η σχεδίαση που ονομάζεται 'bare metal programming', είναι η πιο δύσκολη, αλλά έχει ως αποτέλεσμα την καλύτερη εκμετάλλευση των πόρων. Μια εναλλακτική σχεδίαση αποτελεί η χρήση ενός λειτουργικού συστήματος, π.χ. Linux, που έχει έτοιμες βιβλιοθήκες για πολλές λειτουργίες.

Το μειονέκτημα με το λειτουργικό σύστημα είναι ότι ίσως απαιτηθεί πιο γρήγορος επεξεργαστής και περισσότερη μνήμη, αφού το Λειτουργικό Σύστημα θα χρησιμοποιεί κάποιους από τους πόρους του συστήματος. Η ανάπτυξη του λογισμικού όταν υπάρχει ένα Λειτουργικό Σύστημα γίνεται πολύ πιο εύκολη, αφού το Λειτουργικό Σύστημα παρέχει στον προγραμματιστή μια αφαιρετική εικόνα για το hardware, και έτσι δε χρειάζεται να γνωρίζει λεπτομέρειες σχετικά με το υλικό, αρκεί να χρησιμοποιεί τις τυπικές διεπαφές του Λειτουργικού Συστήματος.

## ➤ Το Ενσωματωμένο Σύστημα σε ένα αυτοκίνητο

Ένα από τα πιο σύνθετα Ενσωματωμένα Συστήματα βρίσκεται σε ένα σύγχρονο αυτοκίνητο. Θα πρέπει να σημειωθεί ότι σε αυτή την περίπτωση δεν υπάρχει μόνο ένα ενσωματωμένο σύστημα, αλλά πλήθος ενσωματωμένων συστημάτων, που συνεργάζονται μεταξύ τους για την επίτευξη του επιθυμητού αποτελέσματος, το οποίο είναι η επίτευξη υψηλής ασφάλειας και άνεσης για τους επιβάτες. Επειδή όλα αυτά τα συστήματα συνεργάζονται μεταξύ τους, εμείς θα τα θεωρήσουμε ως ένα μεγάλο ενσωματωμένο σύστημα. Επιπρόσθετα, τα ενσωματωμένα συστήματα γνώρισαν μεγάλη ανάπτυξη κατά τη δεκαετία του 1970, όταν προέκυψε η πετρελαϊκή κρίση και έτσι οι κατασκευαστές αυτοκινήτων ερεύνησαν πως θα μπορούσαν να μειώσουν την κατανάλωση καυσίμου, και που το πέτυχαν με την τοποθέτηση ψηφιακών συστημάτων μέσα στο αυτοκίνητο που ρύθμιζε κάθε στάδιο λειτουργίας του κινητήρα και βελτιστοποιούσε την καύση.

Οι λειτουργίες του ενσωματωμένου συστήματος σε ένα σύγχρονο αυτοκίνητο είναι:

- Βελτιστοποίηση της καύσης καυσίμου
- Έλεγχος οδήγησης cruise
- Προειδοποίηση σύγκρουσης
- Προειδοποίηση αποχώρησης λωρίδας
- Σύστημα βοήθειας στο παρκάρισμα
- Σύστημα Παρακολούθησης πίεσης ελαστικών
- Ξεκλείδωμα χωρίς κλειδί
- Διασύνδεση με Bluetooth, wifi, GSM
- Πλοήγηση με GPS
- Έλεγχος Ραδιοφώνου (π.χ. όταν λαμβάνει μια κλήση μέσω Bluetooth να μειώνεται η ένταση)
- Σύστημα αντιμπλοκαρίσματος τροχών
- Έλεγχος ταχύτητας
- Κάμερα οπισθοπορείας
- Ηλεκτρικό τιμόνι για τη βοήθεια στο παρκάρισμα
- Έλεγχος αερόσακων
- Έλεγχος ζώνης
- Μπροστινό Radar για άμεση ενεργοποίηση φρένων
- Διαγνωστικά αισθητηρίων
- Multimedia System

Είναι εμφανές ότι ένα τέτοιο σύστημα έχει πολλές και πολύπλοκες απαιτήσεις, άλλες που απαιτούν απόκριση σε πραγματικό χρόνο (π.χ. το ABS), και άλλες που είναι πιο ανεκτικές ως προς τις αποκρίσεις (π.χ. GPS). Επίσης, είναι εμφανές ότι ένα τέτοιο σύστημα δε μπορεί να υποστηριχθεί από ένα μόνο επεξεργαστή, αλλά

απαιτούνται πολλαπλοί επεξεργαστές, διάσπαρτοι σε διάφορα σημεία του οχήματος, οι οποίοι επικοινωνούν με ένα ή περισσότερα δίκτυα διασύνδεσης. Το πιο δημοφιλές δίκτυο διασύνδεσης για τα οχήματα, είναι το CAN (controller area network, δίκτυο περιοχής ελεγκτών). Συνήθως, τα αυτοκίνητα έχουν ένα ή δύο δίκτυα CAN.

Σε περίπτωση που έχουν δύο δίκτυα CAN, αυτά χρησιμοποιούνται ως εξής: Στο πρώτο δίκτυο CAN τοποθετούνται τα αισθητήρια και οι συσκευές δράσης που έχουν αυστηρούς περιορισμούς χρόνου (δηλαδή, από τη στιγμή που ανιχνευτεί μια δυσλειτουργία, θα πρέπει ο επεξεργαστής να τη χειριστεί άμεσα και να δώσει την κατάλληλη εντολή), όπως ABS, έλεγχος πέδησης, έλεγχος τιμονιού, έλεγχος φώτων, έλεγχος κινητήρα. Στο δεύτερο δίκτυο CAN τοποθετούνται αισθητήρια και οι συσκευές δράσης (κυρίως ενδεικτικές λυχνίες) που έχουν πιο χαλαρούς περιορισμούς.

Για παράδειγμα, το αισθητήριο των θυρών (ανοιχτές/κλειστές), της πρόσδεσης της ζώνης, του air-condition και των αισθητήρων θερμοκρασίας. Με την πάροδο του χρόνου οι κατασκευαστές τοποθετούν όλο και περισσότερα αισθητήρια πάνω στους διαύλους CAN προκειμένου να κάνουν όσο το δυνατόν πιο φιλικά στη χρήση και στην οδήγηση, και πιο ασφαλή αυτοκίνητα, εμπλουτίζοντας τα με νέες δυνατότητες. Έτσι, ενώ στο παρελθόν τα αισθητήρια συνδέονταν άμεσα με τις ενδεικτικές λυχνίες στο ταμπλό, τώρα ενημερώνουν τον αντίστοιχο επεξεργαστή (που στην αυτοκινητοβιομηχανία ονομάζεται ECU, engine control unit- μονάδα ελέγχου μηχανής), και αυτός αποφασίζει τι θα εκτελεστεί.

Για ένα τέτοιο ενσωματωμένο σύστημα απαιτείται ένα λειτουργικό σύστημα που σέβεται τις αυστηρές προθεσμίες (μια τυπική επιλογή είναι το λειτουργικό σύστημα QNX Neutrino RTOS), επεξεργαστές 32bit με γρήγορη εξυπηρέτηση διακοπών (π.χ. ARM), με μονάδα διαχείρισης μνήμης (MMU) ή μονάδα προστασίας μνήμης (MPU), αφού θα υπάρχουν αρκετές διεργασίες και θα πρέπει να είναι απομονωμένες, ώστε η μία να μη «μολύνει» το χώρο της άλλης, αρκετά καλώδια για τα αισθητήρια, ένα λειτουργικό σύστημα για το 2ο δίκτυο (π.χ. Linux), που θα σχεδιαστούν οι γραφικές διεπαφές, και ο έλεγχος της τηλεματικής (τηλεματική είναι η αποστολή μέσω ασύρματου δικτύου δεδομένων σε απομακρυσμένη τοποθεσία). Στο όχημα μπορεί να χρησιμοποιηθεί για την αυτόματη αποστολή της θέσης στις πρώτες βοήθειες σε περίπτωση σύγκρουσης.

Επίσης, θα πρέπει κάθε ηλεκτρονικό στοιχείο και αισθητήριο να υποστηρίζει αυτοδιάγνωση και ενημέρωση λειτουργίας και σε περίπτωση προβλήματος να μπορεί να μεταβεί σε μια ασφαλή κατάσταση, ώστε να μην επηρεάζει το υπόλοιπο όχημα. Σύμφωνα με τη νομοθεσία αρκετών χωρών, είναι υποχρεωτικό να ενημερώνεται ο οδηγός μέσω κατάλληλης λυχνίας MIL (malfunction indicator lamp, λυχνία ένδειξης δυσλειτουργίας) για την ύπαρξη οποιουδήποτε προβλήματος. Συστήματα άκρως απαραίτητα για την ασφάλεια, θα πρέπει να έχουν πλεονάζουσες επικοινωνίες, ώστε αν παρουσιάσει πρόβλημα το ένα κανάλι επικοινωνίας, να μπορεί να χρησιμοποιηθεί μια εναλλακτική διαδρομή.

Το ενσωματωμένο σύστημα θα πρέπει να έχει επίσης μνήμη FLASH ή EEPROM, για να αποθηκεύσει σε μη πτητικό μέσο (δηλαδή, που δε χάνει τα δεδομένα αν διακοπεί το ρεύμα), εκτός από τις ρυθμίσεις του χρήστη και το αρχείο καταγραφής βλάβης, ώστε ο μηχανικός να μπορεί με το κατάλληλο μηχάνημα να ενημερωθεί με λεπτομέρειες για το πότε και υπό ποιες συνθήκες παρουσιάστηκε το σφάλμα. Τα αισθητήρια για το σύστημα των αυστηρά χρονικών προδιαγραφών, θα πρέπει να συνδέονται σε ξεχωριστή γραμμή διακοπής (interrupt line), ώστε να εξυπηρετούνται άμεσα αν χρειαστεί, ενώ τα αισθητήρια στο 2ο δίκτυο θα μπορούσαν να επικοινωνούν με κάποια διεργασία τύπου δαίμονα που εκτελείται στο λειτουργικό σύστημα.

Ασφαλώς, θα πρέπει να εξεταστεί ο χρόνος χειρίστης εξυπηρέτησης και να ελαχιστοποιηθεί, ώστε να μη διακυβεύεται η ασφάλεια. Η ανάπτυξη του λογισμικού θα πρέπει να γίνει σε μια γλώσσα υψηλού επιπέδου, εκτός από τις λειτουργίες που απαιτούν ελάχιστο χρόνο εξυπηρέτησης και μπορούν να προγραμματιστούν στη γλώσσα μηχανής του συγκεκριμένου επεξεργαστή.

## **2.8 Συσκευές με ενσωματωμένα συστήματα**

Ένας υπολογιστής γενικού σκοπού, όπως είναι ο προσωπικός υπολογιστής, μπορεί να εκτελεί πολλές διαφορετικές διεργασίες ταυτόχρονα, σε αντίθεση με ένα ενσωματωμένο σύστημα, το οποίο ανάλογα με τον προγραμματισμό του εκτελεί συγκεκριμένη λειτουργία. Όμως τα ενσωματωμένα συστήματα χρησιμοποιούνται για τον έλεγχο πολλών σύγχρονων συσκευών και η χρήση αυτών σε συγκεκριμένες εφαρμογές επιτρέπει στους μηχανικούς που τα σχεδιάζουν να τα βελτιστοποιήσουν, μειώνοντας το μέγεθος και το κόστος του προϊόντος, ή αυξάνοντας την αξιοπιστία και την απόδοση. Ένας κύριος επεξεργαστικός πυρήνας ελέγχει τα ενσωματωμένα συστήματα και στην ουσία αυτή τη λειτουργία την εκτελεί ένας μικροελεγκτής ή ένας επεξεργαστής ψηφιακού σήματος, αν και πρόσφατά χρησιμοποιούνται αρκετά συχνά συστήματα που βασίζονται σε μικροεπεξεργαστές (ARM, MIPS). Οι εντολές του προγράμματος που αναφέρονται στα ενσωματωμένα είναι γνωστές ως firmware και αποθηκεύονται στη ROM ή στη flash memory.

Τα περισσότερα ενσωματωμένα συστήματα αποτελούν προϊόντα μαζικής παραγωγής, τα οποία λόγω της μείωσης του μέσου κόστους παραγωγής επωφελούνται με την αύξηση του πλήθους των παραγόμενων μονάδων. Το εύρος των εφαρμογών οι οποίες έχουν σαν βασικό συστατικό τα ενσωματωμένα συστήματα, εκτείνεται από φορητές συσκευές, όπως είναι ψηφιακά ρολόγια χειρός και συσκευές αναπαραγωγής μουσικής, έως μεγάλες εγκαταστάσεις όπως είναι οι φωτεινοί σηματοδότες, ελεγκτές γραμμών παραγωγής και συστήματα που ελέγχουν τη λειτουργία πυρηνικών εργοστασίων παραγωγής ενέργειας. Η πολυπλοκότητα αυτών των συστημάτων ξεκινάει από χαμηλή διαθέτοντας έναν μόνο μικροελεγκτή μέχρι και πολύ υψηλή, με πολλές μονάδες, περιφερειακές συσκευές και δίκτυα τοποθετημένα σε μεγάλα πλαίσια.

Πολύ χαρακτηριστικά παραδείγματα ενσωματωμένων συστημάτων είναι το κινητό τηλέφωνο και τα φρένα ABS (Anti-lock Break System) των αυτοκινήτων. Σε αυτές τις δύο περιπτώσεις έχουμε την χρήση επεξεργαστή για να επιτελέσει συγκεκριμένες λειτουργίες. Και στις δύο περιπτώσεις ο χρήστης δεν μπορεί να έχει πρόσβαση στον επεξεργαστή, αν και στην περίπτωση κινητών νέας γενιάς με λειτουργικό σύστημα ορατό στον χρήστη υπάρχουν παραδείγματα παρέμβασης χάρη σε εφευρετικούς χρήστες.

Άλλα παραδείγματα ενσωματωμένων συστημάτων είναι οι εκτυπωτές, και οι φούρνοι μικροκυμάτων. Όπως αναφέραμε και σε προηγούμενη ενότητα παρατηρώντας την καθημερινή μας ζωή διαπιστώνουμε πως αλληλεπιδρούμε με ενσωματωμένα συστήματα σχεδόν συνέχεια: ο φούρνος μικροκυμάτων, αλλά πλέον και η συμβατική κουζίνα και το ψυγείο έχουν κάποιας μορφής επεξεργαστή. Τα αυτοκίνητα συχνά έχουν πολλούς επεξεργαστές για διάφορες λειτουργίες (ρύθμιση μίγματος καυσίμου, έλεγχος ψηφιακής οθόνης, σύστημα γεωγραφικού προσδιορισμού θέσης αυτοκινήτου, σύστημα ελέγχου κλιματισμού, κλπ.). Το κινητό μας τηλέφωνο αλλά και το σταθερό τηλέφωνο αν είναι ψηφιακής τεχνολογίας, τα συστήματα ελεγχόμενης στάθμευσης, τα φανάρια κυκλοφορίας αυτοκινήτων, μουσικά όργανα με διεπαφή MIDI, οι εκτυπωτές υπολογιστών, το σύστημα αυτόματης ανάληψης χρημάτων ATM της τράπεζας, το ηλεκτρονικό σύστημα με το οποίο παίρνονται οι παραγγελίες σε κέντρα κλπ.

Σε δυνατές τεχνολογικά χώρες υπάρχουν πιο εξειδικευμένες εφαρμογές, όπως συστήματα στα οποία καταχωρούνται τα στοιχεία για την παρακολούθηση μίας αποστολής από εταιρίες ταχυμεταφοράς, συστήματα παρακολούθησης κυκλοφορίας οχημάτων και έκδοσης κλήσεων σε παραβάτες, και συστήματα εντοπισμού και παρακολούθησης των λεωφορείων ώστε οι επιβάτες στην στάση να ξέρουν επακριβώς πότε θα έρθει το επόμενο λεωφορείο, συστήματα παρακολούθησης των στοιχείων του εδάφους για αγρότες (π.χ. θερμοκρασία και υγρασία για καθορισμό ποτίσματος).

Επίσης στην εποχή μας υπάρχουν παντού και χρησιμοποιούνται οι υπολογιστές, κάτι το οποίο συνεπάγεται την άμεση εξάρτηση από την τεχνολογία η οποία προφανώς σίγουρα αλλάζει το τρόπο ζωής μας, και όπως είναι αναμενόμενο έχει τόσο οφέλη όσο και μειονεκτήματα. Υπάρχουν πλέον ιατρικά συστήματα, όπως η αξονική τομογραφία, καρδιογράφοι, συστήματα μηχανικής υποστήριξης για άτομα με ειδικές ανάγκες, το οποία επιτρέπουν βελτίωση της ποιότητας της ζωής μας. Τα ενσωματωμένα συστήματα μπορούν να έχουν αμέτρητες εφαρμογές στο μέλλον. Ήδη η υπολογιστική ισχύς που μπορεί να έχει κανείς σε προσιτό κόστος είναι τόση ώστε να υπάρχουν συστήματα με αναγνώριση φωνής ως διεπαφή. Σε εργαστηριακή μορφή υπάρχουν ήδη συστήματα που λειτουργούν με εγκεφαλικά κύματα. Το πεδίο είναι ιδιαίτερα ανοιχτό για νέους επιστήμονες και μηχανικούς ώστε να κάνουν την δική τους συμβολή με τις ιδέες τους και αυτή η συμβολή μπορεί να έχει κοινωνικές προεκτάσεις.

Για να είναι εφικτή η επικοινωνία των ενσωματωμένων συστημάτων με τον «εξωτερικό κόσμο» χρησιμοποιούνται διάφορα περιφερειακά όπως:

- Επιφάνειες σειριακής επικοινωνίας RS-232
- Επιφάνειες σύγχρονης σειριακής επικοινωνίας I2C SPI
- Universal Serial Bus
- Multi Media Cards
- Networks->Ethernet
- General purpose I/O
- ADC/DAC
- Χρονιστές
- Debugging->JTAG, ISP

Όσο αφορά στη διεπαφή του χρήστη, ποικίλει από καμία στα ενσωματωμένα που εκτελούν μονάχα μία λειτουργία, έως σύνθετες γραφικές διεπαφές οι οποίες μοιάζουν με τους υπολογιστές. Απλές ενσωματωμένες συσκευές διαθέτουν buttons, LEDs, γραφικά LCD με ένα απλό μενού επιλογών. Τα ενσωματωμένα συστήματα συχνά ανήκουν σε μηχανισμούς που αναμένεται να λειτουργούν συνεχόμενα για χρόνια χωρίς σφάλματα και σε ορισμένες περιπτώσεις να ανακάμπτουν μόνα τους, σε περίπτωση που προκύψει κάποιο σφάλμα. Για το λόγο αυτό το λογισμικό συνήθως αναπτύσσεται και ελέγχεται πιο σχολαστικά από ότι συμβαίνει στη περίπτωση του λογισμικού προσωπικών υπολογιστών. Μη αξιόπιστα κινούμενα μηχανικά μέρη, όπως σκληροί δίσκοι και διακόπτες συνήθως αποφεύγονται.

Μερικές από τις σημαντικότερες παραμέτρους που έχουν σημασία στην σχεδίαση ενσωματωμένων συστημάτων είναι οι εξής:

- **Κόστος κατασκευής:** αφορά την μεγαλύτερη κατηγορία των ενσωματωμένων συστημάτων σε όγκο πωλήσεων. Με δεδομένο ότι υπάρχει επεξεργαστής κάποιας μορφής (ακόμη και ειδικής κατασκευής VLSI σε κάποιες περιπτώσεις) σε συσκευές από ρολόγια χειρός, αριθμομηχανές, κινητά τηλέφωνα, κλιματιστικά, φούρνους μικροκυμάτων, τηλεοράσεις, ψυγεία, και κάθε είδους μαζικά παραγόμενες συσκευές χαμηλού κόστους απόκτησης, το κόστος του ηλεκτρονικού μέρους αυτών είναι σημαντικός παράγοντας για την επιτυχία ενός προϊόντος.
- **Κατανάλωση ενέργειας:** Όλες περίπου οι φορητές ενσωματωμένες μικροηλεκτρονικές συσκευές είναι ευαίσθητες στο ζήτημα της κατανάλωσης ενέργειας (π.χ. προσωπικός ψηφιακός βοηθός – PDA, Personal Digital Assistant).
- **Υπολογιστική Ισχύς:** Κατά περίπτωση απαιτείται και αντίστοιχη υπολογιστική ισχύς. Εντυπωσιακό παράδειγμα είναι το ότι ενσωματωμένα συστήματα παιχνιδιών

όπως το Sony Playstation και το Nintendo έχουν ιδιαίτερα ισχυρούς επεξεργαστές με ειδικές δυνατότητες για γρήγορα γραφικά.

- **Αξιοπιστία:** Κάποιες εφαρμογές έχουν ιδιαίτερα υψηλές απαιτήσεις αξιοπιστίας λόγω είτε της αδυναμίας πρόσβασης στην συσκευή κατά την λειτουργία (π.χ. κάμουλες διαστημικής εξερεύνησης) είτε λόγω κινδύνου απώλειας ανθρώπινης ζωής σε περίπτωση αστοχίας (π.χ. ηλεκτρονικά αεροπλάνων).

- **Ευελιξία κατά την Χρήση:** Όταν μία εταιρία κατασκευής μεταγωγέων παραδίδει ένα προϊόν (π.χ. ένα δρομολογητή – router), αυτός έχει κάποιες ενσωματωμένες δυνατότητες, π.χ. εξέταση πακέτων προερχομένων από προγραμματιζόμενες διευθύνσεις που καταδεικνύουν ανεπιθύμητα πακέτα όπως spam. Κατά την λειτουργία ενδέχεται να αλλάξουν τα χαρακτηριστικά τέτοιων πακέτων, είναι επομένως απαραίτητη η αναβάθμιση στο πεδίο τόσο του λογισμικού του συστήματος, όσο και κάποιων χαρακτηριστικών του υλικού (hardware) αυτού και του ενδιαμέσου επιπέδου firmware. Αυτό μπορεί να οδηγήσει σε σχεδιαστικές λύσεις με αναδιατασσίμενη λογική που παρέχει τέτοιες δυνατότητες, και σίγουρα επηρεάζει την αρχιτεκτονική του συστήματος.

- **Λειτουργία σε Πραγματικό Χρόνο:** Είναι σαφές πως αν τα ηλεκτρονικά ενός αεροπλάνου δεν ελέγξουν τις επιφάνειες αυτού ορισμένες φορές το δευτερόλεπτο, αυτό θα πέσει, ενώ ένα κινητό τηλέφωνο από την στιγμή της εκκίνησής του μπορεί να χρειαστεί μερικά δευτερόλεπτα μέχρις ότου αρχικοποιηθεί, βρει το δίκτυο και είναι έτοιμο να δεχθεί και να στείλει τηλεφωνήματα, ή, μία ψηφιακή φωτογραφική μηχανή μπορεί να χρειαστεί αρκετά δευτερόλεπτα για να αποθηκεύσει μία φωτογραφία.

- **Συμβατότητα με Προγενέστερες Γενιές:** Παρότι ένα νέο φορητό παιχνίδι ενδεικτικής μάρκας μπορεί κάλλιστα να χρησιμοποιήσει ένα νέο επεξεργαστή με καλές δυνατότητες και κόστος ως προς απόδοση, ο τεράστιος όγκος των κασετών με παιχνίδια που υπάρχει ήδη στην αγορά βάζει συχνά περιορισμούς συμβατότητας γιατί διαφορετικά υφιστάμενοι χρήστες μπορεί να στραφούν προς ανταγωνιστικά προϊόντα. Το ίδιο ισχύει και με κινητά τηλέφωνα (πολλές από τις δυνατότητες και μενού των οποίων μπορούν να αναχθούν σε πολλές προγενέστερες γενιές της ίδιας εταιρίας), ψηφιακές φωτογραφικές μηχανές, κλπ.

## 2.9 Ανάπτυξη Υλικού- Λογισμικού Για Τα Ενσωματωμένα Συστήματα

Οι υπολογιστές χρησιμοποιούνται στον έλεγχο ενός μεγάλου φάσματος συστημάτων, από απλές οικιακές μηχανές και ελεγκτές ηλεκτρονικών παιχνιδιών μέχρι ολόκληρες βιομηχανικές εγκαταστάσεις. Το λογισμικό αυτών των συστημάτων πρέπει να αντιδρά σε συμβάντα που παράγονται από το υλικό και να παράγει σήματα ελέγχου αποκρινόμενο σε αυτά τα συμβάντα. Το λογισμικό σε αυτά τα συστήματα είναι ενσωματωμένο στο υλικό, συχνά στη μνήμη για ανάγνωση (ROM), και συνήθως αποκρίνεται σε πραγματικό χρόνο σε συμβάντα του περιβάλλοντος του συστήματος.

Ενσωματωμένο λογισμικό είναι το λογισμικό υπολογιστών, γραμμένο για να ελέγχει μηχανές ή συσκευές που συνήθως δε θεωρούνται ως υπολογιστές. Είναι συνήθως εξειδικευμένες για το συγκεκριμένο υλικό που τρέχει και να έχει το χρόνο και τη μνήμη τους σε περιορισμούς. Αυτός ο όρος μερικές φορές χρησιμοποιείται εναλλακτικά με firmware, αν και firmware μπορεί επίσης να εφαρμοστεί σε κωδικό ROM που βασίζεται σε έναν υπολογιστή, στην κορυφή του οποίου το λειτουργικό σύστημα τρέχει, ενώ το ενσωματωμένο λογισμικό είναι συνήθως το μόνο λογισμικό για την εν λόγω συσκευή.



## ΚΕΦΑΛΑΙΟ 3

### 3. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ

Η μοντελοποίηση είναι η πλέον σημαντική διαδικασία στην ανάπτυξη συστημάτων γενικά, ακόμη και συστημάτων λογισμικού. Με αυτό το σκεπτικό δεν είναι ειδικό θέμα της ανάπτυξης ενσωματωμένων συστημάτων. Σε ενσωματωμένα συστήματα η μοντελοποίηση ενέχει επί πλέον περιορισμούς επειδή τίθεται θέμα απεικόνισης της τελικής λύσης σε συστήματα με συγκεκριμένους πόρους.

Για παράδειγμα, ένα σύστημα λογισμικού μπορεί να υλοποιήσει δυναμικές δομές δεδομένων, καθότι με υποστήριξη ιδεατής μνήμης η κατάληψη μεγαλύτερης μνήμης είναι διαδικασία διαφανής στον προγραμματιστή, και σε μεγάλο βαθμό υποστηριζόμενη από μεταφραστές, λειτουργικό σύστημα, κλπ. Εάν όμως επιθυμούμε να υλοποιήσουμε κάτι αντίστοιχο σε ένα ενσωματωμένο σύστημα με επεξεργαστή 8-bits με μικρό χώρο διευθύνσεων και μνήμη (τυπικά 64KBytes), οι αντίστοιχες δομές που θα χρησιμοποιήσουμε θα πρέπει να είναι άμεσα ορισμένες (π.χ. πως θα ορίσουμε την στοίβα μας ώστε να μην έχουμε πρόβλημα ακόμη και στις χειρότερες συνθήκες).

#### 3.1 Τι είναι Μοντελοποίηση

Μοντελοποίηση είναι η διαδικασία δημιουργίας ενός προτύπου (μοντέλου) του συστήματος που θέλουμε να υλοποιήσουμε. Το πρότυπο αυτό μπορεί να είναι εκφρασμένο σε μεγαλύτερη ή μικρότερη λεπτομέρεια, αλλά σε κάθε περίπτωση πρέπει να είναι επαρκές για τα επόμενα βήματα της σχεδιαστικής διαδικασίας ούτως ώστε να οδηγηθούμε σε πλήρως λειτουργικά ολοκληρωμένα συστήματα. Η μοντελοποίηση μπορεί να είναι σε μία ευρεία κλίμακα από μεθόδους, π.χ. στο χαρτί, με γλώσσες προγραμματισμού, με γλώσσες περιγραφής υλικού, κλπ. Η κατά περίπτωση επιλογή είναι καθαρά θέμα του μηχανικού και είναι από τα πλέον δημιουργικά μέρη της διαδικασίας σχεδιασμού.

#### 3.2 Η ανάγκη για τη Μοντελοποίηση

Η ραγδαία ανάπτυξη των ενσωματωμένων συστημάτων δεν οφείλεται μόνο στην πρόοδο που επιτεύχθηκε στο υλικό μέρος (π.χ. αύξηση επιδόσεων των επεξεργαστών, αύξηση βαθμού ολοκλήρωσης) αλλά και από το λογισμικό μέρος. Το λογισμικό μέρος είναι αυτό που οργανώνει, κατευθύνει και ελέγχει το υλικό μέρος για τη σωστή και έγκυρη λειτουργία του συστήματος. Όπως τα χαρακτηριστικά του υλικού μέρους επηρεάζουν σημαντικά την ταχύτητα και την κατανάλωση ενέργειας του συστήματος (π.χ. η χρήση ενός πολύ αργού επεξεργαστή θα έχει ως συνέπεια το όλο σύστημα να λειτουργεί με μικρή ταχύτητα), έτσι και το λογισμικό μέρος παίζει έναν κυρίαρχο ρόλο στη διαμόρφωση όλων των μετρικών ενός συστήματος.

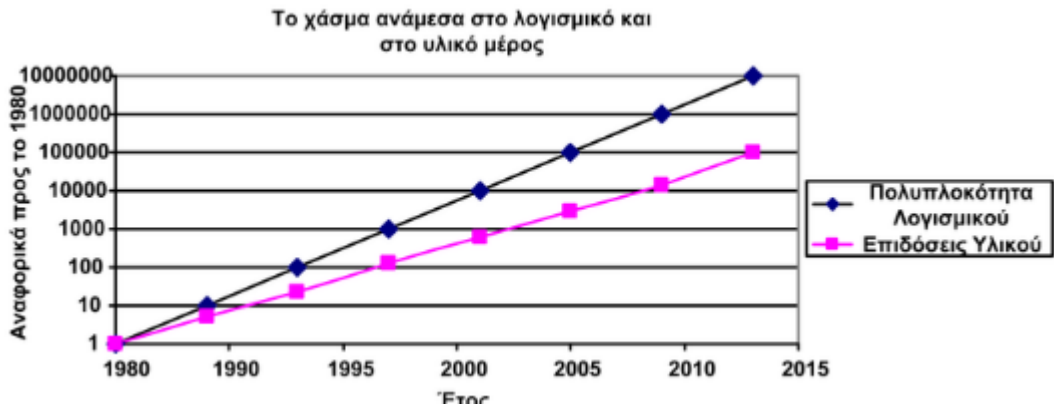
Ένα λογισμικό μέρος καλά προγραμματισμένο θα βοηθήσει το υλικό μέρος να αποδώσει τα μέγιστα των δυνατοτήτων του, ενώ σε αντίθετη περίπτωση, θα κάνει και το ενσωματωμένο σύστημα με τα πιο γρήγορα υλικά μέρη να μειονεκτεί αισθητά

έναντι παρόμοιων συστημάτων. Το λογισμικό μέρος συμπληρώνει το υλικό μέρος. Η σχέση που συνδέει αυτά τα δύο μέρη είναι παρόμοια (σε ένα αρκετό υψηλό επίπεδο αφαίρεσης), με τη σχέση που έχει ένα όχημα με τον οδηγό του. Αν και το όχημα έχει όλη τη λειτουργικότητα που απαιτείται, εντούτοις περιμένουμε από τον οδηγό να το εκκινήσει, να κάνει τις κατάλληλες ενέργειες, ώστε το όχημα να μετακινηθεί και να οδηγηθεί σε κάποια τοποθεσία που είχε τεθεί ως προορισμός. Μπορεί να θεωρηθεί ότι ο οδηγός παριστάνει το λογισμικό μέρος, ενώ το υλικό μέρος συνδέεται με το όχημα.

Έτσι, η συνύπαρξη και των δύο αυτών μερών αποτελεί το συνολικό σύστημα. Παρόλο που και τα δύο είναι αναπόσπαστα μέρη του συστήματος, κατά τα τελευταία χρόνια έχει δοθεί μεγαλύτερη έμφαση στο λογισμικό και, ιδιαίτερα, στους τρόπους καλύτερης μοντελοποίησης και προγραμματισμού των εφαρμογών, ειδικά για ενσωματωμένα συστήματα. Αυτό οφείλεται κυρίως στην τάση που επικρατεί να χρησιμοποιούνται τυπικά εξαρτήματα υλικού μέρους από το ράφι (off-the-shelf components), όπως για παράδειγμα επεξεργαστές των εταιρειών ARM, ή TI, ή μνήμες των εταιρειών Motorola, ή Kingston, κ.τ.λ. Όπως παρατηρείται παγκοσμίως, το πρόβλημα της ανάπτυξης ενός συστήματος, κατευθύνεται περισσότερο προς τη συγγραφή του λογισμικού μέρους, που θα ενεργοποιήσει το υλικό μέρος. Έτσι, η μερική τυποποίηση των υλικών μερών των ενσωματωμένων συστημάτων αποτελεί έναν παράγοντα που έχει μετατοπίσει το πρόβλημα της ανάπτυξης ενσωματωμένων συστημάτων προς το λογισμικό μέρος. Όμως, αυτός δεν είναι ο μόνος λόγος.

Τα τελευταία χρόνια, η ανάπτυξη των εφαρμογών των ενσωματωμένων συστημάτων έχει γίνει ο κυρίαρχος διαμορφωτής του κόστους για έναν ακόμη παράγοντα: αυτός είναι η πολυπλοκότητα των εφαρμογών. Οι πρώτες εκδόσεις των ενσωματωμένων συστημάτων είχαν ένα μικρό πρόγραμμα, το οποίο επιτελούσε κάποιες πολύ απλές λειτουργίες. Μαζί με την απαίτηση της αγοράς για ενσωματωμένα συστήματα με καλύτερες επιδόσεις στο υλικό μέρος, υπήρξε και η απαίτηση για πιο πολλές λειτουργίες, για προγράμματα φιλικά προς το χρήστη, και για εφαρμογές αρκετά απαιτητικές (όπως π.χ. η αναπαραγωγή πολυμέσων). Γι' αυτόν το λόγο υπήρξε και υπάρχει αρκετά μεγαλύτερη έμφαση στην ανάπτυξη των εφαρμογών ενσωματωμένων συστημάτων.

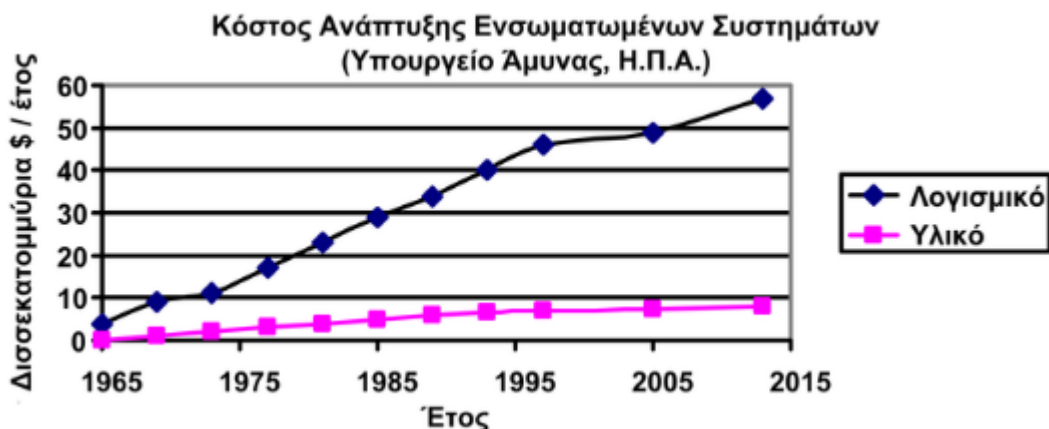
Ένας τρόπος μέτρησης της ανάπτυξης των εφαρμογών αποτελεί και το μέτρο της πολυπλοκότητας των εφαρμογών. Η πολυπλοκότητα μιας εφαρμογής εξαρτάται από τον αριθμό των εντολών και συναρτήσεων που την αποτελούν, από τις λειτουργίες που αυτή κάνει, από το μέγεθος κώδικα που αποθηκεύεται στη μνήμη εντολών, και από τον αποθηκευτικό όγκο σε μνήμη δεδομένων. Αν βγάλουμε χονδρικά ένα μέσο όρο της πολυπλοκότητας των εφαρμογών θα παρατηρήσουμε στο Σχήμα ότι ακολουθεί μια εκθετική αύξηση χρονολογικά. Αν στο ίδιο γράφημα τοποθετήσουμε και τις επιδόσεις των επεξεργαστών, θα δούμε ότι αυτές υστερούν, και δεν μπορούν να καλύψουν τις τρέχουσες απαιτήσεις των εφαρμογών. Βλέπουμε, λοιπόν, ότι υπάρχει ένα χάσμα ανάμεσα στο τι μπορεί να εκτελεστεί σε ένα σύστημα (επιδόσεις υλικού μέρους) και στο τι απαιτούν οι εφαρμογές για να εκτελεστούν (πολυπλοκότητα λογισμικού μέρους).



Σχήμα 3.2 (Εικόνα 1) : Η πολυπλοκότητα των εφαρμογών ενσωματωμένων συστημάτων αυξάνεται με πολύ μεγαλύτερο ρυθμό σε σχέση με τις επιδόσεις των επεξεργαστών.

Το πρόβλημα της δημιουργίας λογισμικού ενσωματωμένων συστημάτων είναι τόσο σημαντικό, ώστε μεγάλες εταιρείες, όπως η Microsoft και η Sun (εταιρείες που έχουν κυριαρχήσει στο λογισμικό μέρος των επιτραπέζιων συστημάτων, ή των διακομιστών), έχουν κατευθύνει μέρος του προϋπολογισμού τους στην έρευνα και ανάπτυξη εφαρμογών ενσωματωμένων συστημάτων. Για παράδειγμα, η Microsoft αναπτύσσει τα Windows Mobile, η Google το Android, και η Sun τη Java Virtual Machine, που όλα συμβάλλουν στην ανάπτυξη λογισμικού ενσωματωμένων συστημάτων.

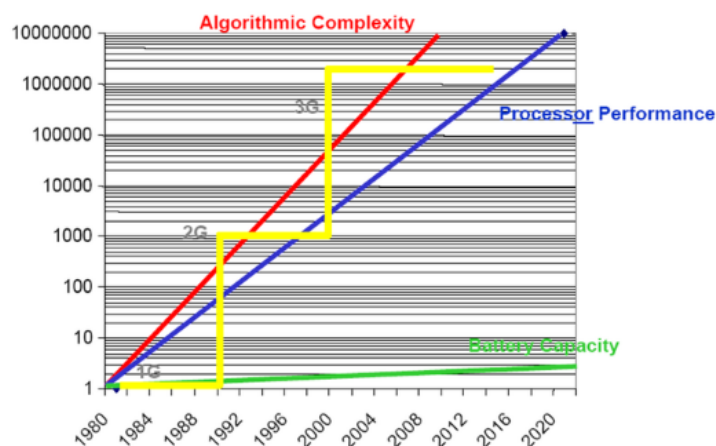
Το λογισμικό μέρος των ενσωματωμένων συστημάτων, είναι ένας ακμάζων τομέας, και αυτό φαίνεται στα οικονομικά μεγέθη που το συνοδεύουν. Για παράδειγμα, τα στοιχεία από το Υπουργείο Άμυνας των Η.Π.Α. – ένας από τους μεγαλύτερους παγκοσμίως αγοραστές ενσωματωμένων συστημάτων – (Σχήμα 4.2), δείχνουν ότι το κόστος ανάπτυξης του λογισμικού είναι έως και 6 φορές μεγαλύτερο από το κόστος ανάπτυξης των υλικών μερών. Είναι γνωστό, ότι το λογισμικό καταλαμβάνει περισσότερο από το 70% του κόστους ανάπτυξης πολύπλοκων ενσωματωμένων συστημάτων, όπως φορητά τερματικά, ηλεκτρονικά συστήματα αυτοκινήτων, συστήματα επικοινωνιών.



Σχήμα 3.2 (Εικόνα 2): Το λογισμικό μέρος κυριαρχεί στη διαμόρφωση του κόστους ενός ενσωματωμένου συστήματος.

Τέλος, η σπουδαιότητα του λογισμικού των ενσωματωμένων συστημάτων φαίνεται καθαρά και από τα στοιχεία που αναφέρονται στην έκθεση του Υπουργείου Εμπορίου και Βιομηχανίας της Ολλανδίας, στην οποία αναφέρεται ότι ο αριθμός των ατόμων ή εταιρειών που παίρνουν συμβόλαια για ανάπτυξη κώδικα ενσωματωμένων συστημάτων θα αυξηθεί από 2 εκατομμύρια το 1994 σε 13 εκατομμύρια το 2015, ενώ ο αριθμός των κατασκευαστών υλικού ενσωματωμένων συστημάτων θα αυξηθεί από 0,6 εκατομμύρια σε 1,1 εκατομμύρια, αντιστοίχως.

Είναι σημαντικό λοιπόν να αναλύεται η εφαρμογή, που θα εκτελεστεί στο ενσωματωμένο σύστημα, και να γίνεται συσχεδιασμός του υλικού και του λογισμικού μέρους. Η μεθοδολογία, σχεδιασμού Ενσωματωμένου Συστήματος αποτελείται από στάδια που βελτιώνουν το λογισμικό μέρος και από στάδια που βελτιώνουν το υλικό μέρος. Επίσης, μια σωστή μεθοδολογία έχει εφαρμοστεί με επιτυχία σε ένα πλήθος εφαρμογών, που εκτελούνται σε ενσωματωμένα συστήματα, γεγονός που δείχνει τη γενική χρήση που μπορεί αυτή να έχει.



Σχήμα 3.2 (Εικόνα 3): Η πολυπλοκότητα των εφαρμογών αυξάνεται λογαριθμικά.

### 3.3 Το NRE εξαρτάται από τη μοντελοποίηση

Το κόστος έρευνας και ανάπτυξης (non recurrent engineering cost, NRE) ενός ενσωματωμένου συστήματος είναι αποφασιστικής σημασίας για τη διαμόρφωση της τιμής του τελικού προϊόντος, αφού θα πρέπει να ενσωματωθεί στην τελική τιμή. Όσο μεγαλύτερο είναι αυτό το αρχικό κόστος, τόσο πιο πολύ θα αυξηθεί η τιμή των τελικών προϊόντων και ενδεχομένως θα επηρεάσει αρνητικά τον αριθμό πωλήσεων, προκαλώντας ίσως και ζημιά στην εταιρία που το προμηθεύει στην αγορά.

Η μείωση του κόστους NRE είναι λοιπόν ζωτικής σημασίας, αφού η ίδια η βιωσιμότητα μιας εταιρίας εξαρτάται από αυτό, και θα πρέπει να είναι όσο το δυνατόν μικρότερο. Το ίδιο προϊόν μπορεί σε δυο διαφορετικές εταιρίες να συνδέονται με NRE κόστη που έχουν πολύ μεγάλη διαφορά. Το κόστος NRE για το ίδιο προϊόν σε δυο διαφορετικές εταιρίες είναι διαφορετικό. Για να ελαχιστοποιηθεί αυτό το κόστος θα πρέπει να θυμηθούμε (1ο κεφάλαιο) τους παράγοντες που το διαμορφώνουν. Ο πρώτος παράγοντας είναι το κόστος μισθοδοσίας του προσωπικού. Μια εταιρία μπορεί να έχει υψηλό μισθολογικό κόστος λόγω νοοτροπίας ή

φορολογίας και ασφαλιστικού, ενώ μια εταιρία σε άλλη περιοχή του πλανήτη μπορεί να έχει υποπολλαπλάσιο κόστος.

Για αυτό αρκετές εταιρίες του Δυτικού Κόσμου, προβαίνουν σε εξωτερικές αναθέσεις (outsourcing) ή ανοίγουν παραρτήματα σε χώρες με χαμηλό μισθολογικό κόστος, όπως η Ινδία ή η Κίνα. Ο επόμενος παράγοντας είναι τα πάγια έξοδα των κτιρίων. Ομοίως, αυτό μπορεί να μειωθεί σε μετεγκατάσταση της εταιρίας είτε σε κατάλληλη οικοδομική ζώνη που παρέχει σε μικρότερο κόστος κτιριακή επιφάνεια, νερό, ρεύμα και διαδίκτυο, ή σε μετεγκατάσταση σε άλλη πόλη ή χώρα.

Ένας άλλος παράγοντας, είναι το κόστος των υλικών για τη δημιουργία του πρωτοτύπου. Αν και αυτό το κόστος είναι συνήθως μικρότερο από τα δυο προηγούμενα, εντούτοις διαμορφώνει και αυτό το NRE, ιδιαίτερα αν πρέπει να γίνουν πολλά και μακροχρόνια πειράματα, δοκιμάζοντας πολλά διαφορετικά σενάρια ή σχεδιασμούς. Για να μειωθεί αυτό το κόστος, εκτός από την προμήθεια των υλικών από πάροχο που έχει τις χαμηλότερες τιμές και ταυτόχρονα καλύπτει τα επίπεδα αξιοπιστίας, ποιότητας και ταχύτητας παράδοσης, θα πρέπει να ακολουθηθεί μια τεχνική που θα επιτύχει μείωση του χρόνου, μείωση των λανθασμένων σχεδιαστικών επιλογών και μείωση των προβληματικών πρωτοτύπων. Αυτή η τεχνική ονομάζεται μοντελοποίηση, και είναι πάντα το πρώτο βήμα σε οποιαδήποτε μεθοδολογία σχεδιασμού, είτε ενσωματωμένων συστημάτων είτε οποιοδήποτε άλλου προϊόντος.

### **3.4 Η σημασία της Μοντελοποίησης**

Η Μοντελοποίηση είναι μια τεχνική (ή για να είμαστε ακριβείς, ένα σύνολο από τεχνικές) που επιτρέπει ένα σχεδιαστή να πετύχει το στόχο του με την πρώτη προσπάθεια. Αν και έχει μεγάλη σημασία, αρκετές φορές παραλείπεται από τα επίσημα διαγράμματα σχεδίασης, λόγω χρονικής πίεσης, αφού κάποιιοι σχεδιαστές πιστεύουν ότι μπορούν να καταφέρουν το τελικό αποτέλεσμα (π.χ. τη ανάπτυξη ενός ενσωματωμένου συστήματος) με τη σύνδεση των δομοστοιχείων και τη συγγραφή του κώδικα. Αν και μπορεί να το πετύχουν και να σχεδιάσουν κάτι που να λειτουργεί, εντούτοις η μεγαλύτερη πιθανότητα είναι ότι δε θα το καταφέρουν και θα χάσουν το παράθυρο εισαγωγής στην αγορά με δυσάρεστες οικονομικές συνέπειες. Αλλά, ακόμη και να τα καταφέρουν, δε σημαίνει ότι το τελικό αποτέλεσμα θα είναι στιβαρό και θα λειτουργεί σε οποιαδήποτε συνθήκη.

Παραδείγματα που δείχνουν τη σημασία της μοντελοποίησης μπορούμε να βρούμε σε κάθε χώρο. Η μοντελοποίηση δεν ανήκει στα βήματα της μεθοδολογίας που θα ήταν καλό να γίνουν. Ανήκουν στα βήματα που πρέπει να γίνουν. Η σημαντικότητα αυξάνεται όσο περισσότερο αυξάνεται η πολυπλοκότητα και οι απαιτήσεις. Αν κάποιιοι αποφασίσουν να κατασκευάσουν μια αποθήκη με ξύλα, αρκεί να σημειώσουν σε ένα χαρτί τις διαστάσεις των τοίχων, τα εργαλεία και τα υλικά που θα χρησιμοποιήσουν. Σε περίπτωση που δεν ταιριάζουν κάποια υλικά μεταξύ τους μπορεί να γίνουν κάποια μπαλώματα για να τα καλύψουν. Αν όμως απαιτείται η κατασκευή ενός σπιτιού με πολλούς χώρους, τότε αυξάνεται η πολυπλοκότητα και θα πρέπει να ληφθούν υπόψιν πολλά πρισσότερα στοιχεία, όπως διελεύσεις των καλωδίων, των δικτύων ύδρευσης και αποχέτευσης, τα πάχη των τοίχων μαζί με τη

μόνωση, τα κουφώματα και πλήθος άλλων λεπτομερειών. Αν δε γίνει αυτό, στο τέλος μπορεί να μην καλύπτονται οι προδιαγραφές κατοίκησης και να μην έχει σωστή μόνωση ή μπορεί να ξεχάσουν να βάλουν βρύσες στο μπάνιο ή ακόμη αν είναι πολυκατοικία να υπάρχει κατάρρευση, αν δεν έχει γίνει μοντελοποίηση των στατικών φορτίων.

Για αυτό είναι απαραίτητο να μοντελοποιηθεί η κατασκευή ενός τέτοιου σύνθετου σχεδιασμού και μάλιστα, όχι μόνο με μια κάτοψη. Θα πρέπει να γίνει μια θερμική μελέτη, ηλεκτρολογική μελέτη, μελέτη ύδρευσης και αποχέτευσης, στατική μελέτη και άλλα. Σε περίπτωση που υπάρξει πρόβλημα σε μια τέτοια κατασκευή μπορεί να είναι δύσκολο να διορθωθεί, με αποτέλεσμα να υπάρχει σημαντική οικονομική καταστροφή.

Αν δεν υπάρξει η μοντελοποίηση, το τελικό αποτέλεσμα βασίζεται στην τύχη και σε υπερβολική εργασία της ομάδας να διορθώσει τα προβλήματα καθώς εμφανίζονται. Στις περισσότερες περιπτώσεις όμως, δεν υπάρχει τύχη, η ομάδα ασχολείται ήδη πολλές εργατοώρες με το προϊόν και δε μπορεί να διαθέσει άλλες ή πλησιάζει κάποια κρίσιμη καταληκτική ημερομηνία, και λόγω κακού προγραμματισμού δεν υπάρχουν όλες οι πρώτες ύλες, που είναι απαραίτητες για την κατασκευή του πρωτοτύπου.

Σχεδιαστικά project μεγάλων προϋπολογισμών (πάνω από 1.000.000 ευρώ) που χρηματοδοτούνται συμμετοχικά από ανώνυμους χρήστες (crowd-funding), έχουν αποτύχει λόγω έλλειψης μοντελοποίησης, όπως για παράδειγμα η εταιρία Pirate3D που μάζεψε \$1.5 εκατομμύρια δολάρια για την κατασκευή οικονομικών τρισδιάστατων εκτυπωτών, αλλά τελικά το κόστος τελικών προϊόντων ήταν πολύ μεγαλύτερο από το αναμενόμενο, ή η εταιρία Zano Drones που συγκέντρωσε \$3.5 εκατομμύρια δολάρια για την ανάπτυξη μικρών τετρακόπτερων που ελέγχονται από το smartphone, αλλά μετά από μήνες έρευνας χρεοκόπησε χωρίς να κατασκευάσει τίποτα. Σίγουρα για την ολοκληρωτική αποτυχία δεν έφταιγε μόνο η έλλειψη σωστής μοντελοποίησης των προϊόντων που ήθελαν να σχεδιάσουν, αλλά μια ελλιπής μοντελοποίηση ή λάθος εκτιμήσεις οδηγούν σε αυτό το αποτέλεσμα.

Ο σχεδιασμός και η κατασκευή ενός προϊόντος δεν πρέπει να βασίζεται στην τύχη. Θα πρέπει να έχει προηγηθεί μια σειρά μοντελοποιήσεων σε διάφορα επίπεδα, ώστε να βρεθούν από πριν τα προβλήματα, τα υλικά, τα εργαλεία και το κόστος που τα συνοδεύει. Ενώ το κάθε αποτυχημένο σχεδιαστικό έργο αποτυγχάνει με ποικίλους τρόπους, όλα τα σωστά σχεδιασμένα προϊόντα έχουν ένα κοινό στοιχείο: μια προσεκτική και σε βάθος περίοδο μοντελοποίησης.

Η μοντελοποίηση έχει αποδειχθεί ότι είναι απαραίτητο στοιχείο και εργαλείο των μηχανικών. Η μοντελοποίηση δεν αφορά μόνο τον κατασκευαστικό τομέα. Χρησιμοποιείται παντού: από την οικοδομική κατασκευή έως την κατασκευή νέων οχημάτων ή αεροπλάνων ή και δορυφόρων, από την κατασκευή νέων επεξεργαστών,

έως και μεγάλων συστοιχιών με δεκάδες χιλιάδες επεξεργαστές, από τη μοντελοποίηση ενός τοπικού δικτύου, ως και τη μοντελοποίηση του διαδικτύου κ.ο.κ.

Οποιοδήποτε σύστημα μπορεί να μοντελοποιηθεί, με πολλούς τρόπους και με πολλαπλά επίπεδα λεπτομέρειας. Μπορεί να χρησιμοποιηθεί ένα μαθηματικό μοντέλο, ένα δομικό μοντέλο, ένα ιεραρχικό μοντέλο, ένα μοντέλο συμπεριφοράς, ένα μοντέλο απεικόνισης, ένα μοντέλο επικοινωνίας κ.ο.κ. Γενικά, ένα μοντέλο είναι οτιδήποτε μπορεί να απλοποιήσει και να απεικονίσει την πραγματικότητα, και ο λόγος που τα κατασκευάζουμε είναι για την καλύτερη κατανόηση του συστήματος που σχεδιάζουμε.

Με τη χρήση των μοντέλων επιτυγχάνουμε τους παρακάτω στόχους:

- Απεικονίζεται το σύστημα όπως είναι ή όπως το σκεφτόμαστε.
- Προσδιορίζουμε τη δομή και την ιεραρχία ή τη συμπεριφορά.
- Μας οδηγεί και μας υπενθυμίζει τα βήματα προς την ολοκλήρωση του συστήματος.
- Καταγράφει λεπτομέρειες και αποφάσεις που έχουμε πάρει.
- Μας επιτρέπει να δημιουργήσουμε πολλαπλά αντίγραφα του ίδιου προϊόντος.
- Μας βοηθάει να επιβεβαιώσουμε τις λειτουργικές και άλλες προδιαγραφές, πριν ακόμη κατασκευαστεί.
- Μας επιτρέπει να κάνουμε μια γρήγορη σχεδιαστική αναζήτηση, αν υπάρχουν πολλές εναλλακτικές.
- Μας επιτρέπει να έχουμε μια κοινή εικόνα για το τελικό προϊόν σε όλα τα μέλη της ομάδας.

Όσο πιο πολύπλοκο είναι ένα σύστημα τόσο πιο σημαντική είναι η μοντελοποίηση, και αυτό γιατί υπάρχει αδυναμία να κατανοηθεί η έντονη πολυπλοκότητα και οι αλληλεξαρτήσεις από τους μηχανικούς. Ιεραρχικά και τμηματικά όμως μπορούμε να το επιτύχουμε, αφού η προσέγγιση 'διαίρεση-και-κατάκτηση' που είχε προτείνει ο Edsger Dijkstra, δηλαδή της διαίρεσης ενός σύνθετου προβλήματος σε πολλά επιμέρους βήματα, τα οποία ξεχωριστά μπορούν να επιλυθούν, δίνουν το καλύτερο αποτέλεσμα. Αν επιτευχθεί η επίλυση όλων των μικρών προβλημάτων, τότε θα επιτευχθεί και η επίλυση του μεγάλου πολύπλοκου προβλήματος. Η πρόταση του Dijkstra υλοποιείται με τη μοντελοποίηση.

Αφού έγινε κατανοητή η σημασία της μοντελοποίησης, το επόμενο βήμα είναι να βρεθεί ο σωστός τρόπος μοντελοποίησης. Κάποιοι σχεδιαστές Ενσωματωμένων Συστημάτων, μπορούν να μοντελοποιήσουν ένα σύστημα σε ένα κομμάτι χαρτιού, στον πίνακα, σε ένα σχεδιαστικό πρόγραμμα στο κινητό τους ή ακόμη και σε μια χαρτοπετσέτα.

Ασφαλώς, και μόνο που γίνεται η μοντελοποίηση, είναι ένα θετικό στοιχείο, αν μπορεί να το ακολουθήσει και να τον οδηγήσει σε μια σωστή υλοποίηση. Όμως, το πρόβλημα με αυτά τα αυθαίρετα μοντέλα είναι ότι τα καταλαβαίνει μόνο ο δημιουργός τους και από την άλλη υπάρχει μια δυσκολία διαμοιρασμού σε άλλους, και συνεπώς και δυσκολία επεξεργασίας και σχολιασμού από αυτούς. Οπότε, το σημείο κλειδί είναι η δημιουργία ενός μοντέλου σε προτυποποιημένη (formal)

γλώσσα, που την κατανοούν και άλλοι και μπορούν να τη χρησιμοποιήσουν για τις δικές τους ανάγκες.

Επομένως, θα πρέπει η σχεδιαστική ομάδα να επιλέξει μια ή παραπάνω γλώσσες μοντελοποίησης και να τις χρησιμοποιεί έως την κατασκευή του προϊόντος ή ακόμη και μετά για τη βελτιστοποίηση και αναθεώρηση.

### 3.5 Οι 4 αρχές της Μοντελοποίησης

Υπάρχουν τέσσερις βασικές αρχές στη μοντελοποίηση:

- Η επιλογή του μοντέλου επιδρά στον τρόπο που αντιμετωπίζεται το πρόβλημα και στη λύση που διαμορφώνεται.
- Κάθε μοντέλο μπορεί να εκφραστεί με διαφορετικά επίπεδα ακρίβειας.
- Τα καλύτερα μοντέλα είναι συνδεδεμένα με την πραγματικότητα.
- Δεν υπάρχει ένα μοντέλο για όλα. Υπάρχουν πολλαπλά μοντέλα και θα πρέπει να χρησιμοποιούνται παραπάνω από ένα σε κάθε πρόβλημα.

Η πρώτη αρχή σημαίνει ότι η σχεδιαστική ομάδα θα πρέπει να γνωρίζει αρκετούς τρόπους μοντελοποίησης για να επιλέξει το κατάλληλο εργαλείο. Τα σωστά μοντέλα θα διαφωτίσουν και θα φέρουν στην επιφάνεια σχεδιαστικά θέματα που ίσως δε θα είχαν γίνει κατανοητά, αν είχε επιλεγθεί κάποιο άλλο μοντέλο.

Η δεύτερη αρχή σημαίνει ότι το ίδιο στοιχείο μπορεί να εκφραστεί με πολλά επίπεδα λεπτομέρειας. Από ένα απλό εκτελέσιμο που δέχεται κάποιους εισόδους και εμφανίζει στην έξοδο κάποια αποτελέσματα, έως το επίπεδο τρανζίστορ που δείχνει πως συμπεριφέρεται κάθε πύλη. Βέβαια, όσες πιο πολλές λεπτομέρειες έχει ένα μοντέλο, τόσο πιο δύσκολη είναι η ανάπτυξη και βελτίωση του. Θα πρέπει να επιλέγεται το κατάλληλο επίπεδο λεπτομέρειας κάθε φορά.

Η τρίτη αρχή ερμηνεύεται ότι θα πρέπει τα μοντέλα που επιλέγονται να έχουν μεγάλη σχέση με την πραγματικότητα και να μη γίνονται αβάσιμες εκτιμήσεις λειτουργίας. Στα μοντέλα αυτά θα πρέπει να εξετάζονται και οι ακραίες συνθήκες και όχι μόνο οι τυπικές. Όλα τα μοντέλα απλοποιούν την πραγματικότητα. Θα πρέπει να προσέχουμε όμως, να μην υπερ-απλοποιούμε την πραγματικότητα καλύπτοντας σημαντικές πτυχές, που μπορεί να επηρεάσουν αρνητικά το προϊόν.

Η τέταρτη αρχή σημαίνει ότι για το ίδιο προϊόν απαιτούνται πολλά μοντέλα που δείχνουν μια διαφορετική εικόνα. Όπως σε ένα κατασκευαστικό έργο, υπάρχουν πολλά μοντέλα (ηλεκτρολογικό, στατικό, μηχανολογικό, θερμομονωτικό,...), έτσι και σε ένα έργο σχεδιασμού ενσωματωμένων συστημάτων, χρησιμοποιούνται πολλά μοντέλα που είναι ανεξάρτητα ως προς αυτό που εξετάζουν, αλλά συνδέονται με το τελικό προϊόν.

Τυπικά μοντέλα που συναντώνται στα Ενσωματωμένα Συστήματα είναι το μοντέλο καταστάσεων χρήσης (use-cases) που περιγράφει τις λειτουργικές προδιαγραφές του συστήματος, το μοντέλο σχεδιασμού που παρουσιάζει τα φυσικά και ηλεκτρονικά



δομοστοιχεία του κυκλώματος και πως συνδέονται, το μοντέλο υλοποίησης που δείχνει πως θα είναι το τελικό προϊόν, το μοντέλο λειτουργικής εμφάνισης, που δείχνει στον τελικό χρήστη πως θα επικοινωνεί μαζί του, και ασφαλώς μοντέλα περιγραφής προγράμματος, σχεδιασμού, και προσομοίωσης.

### **3.6 Μοντελοποίηση στο Χαρτί**

Μοντελοποίηση στο χαρτί είναι η ad hoc (δηλαδή χωρίς κάποιον κανόνα) διαδικασία με την οποία πολλές φορές παίρνουμε ένα λευκό χαρτί και «σχεδιάζουμε» ένα σύστημα. Πάρα πολλά συστήματα έχουν σχεδιαστεί με αυτόν τον τρόπο, όταν ειδικοί σε ένα αντικείμενο εκφράζουν με συνοπτικό τρόπο τα γενικά λειτουργικά χαρακτηριστικά ενός συστήματος. Επίσης πολλά συστήματα χαρακτηρίζονται με αυτόν τον τρόπο (π.χ. clean slate design δηλαδή σχεδίαση σε καθαρό πίνακα). Γενικά αυτό είναι μία πολυτέλεια που δεν έχουν οι μηχανικοί συχνά διότι θέματα συμβατότητας με υφιστάμενα συστήματα, σχεδιαστικοί περιορισμοί, εξωγενείς παράγοντες (π.χ. κάποιο παιχνίδι να είναι στην αγορά πριν τις γιορτές), κλπ.

Συχνά επιβάλλουν λύσεις με κριτήρια διαφορετικά από τους καθαρά τεχνολογικούς περιορισμούς. Συνεχίζοντας το παράδειγμα των κινητών τηλεφώνων, η συμβατότητα στην διεπαφή με τον χρήστη σε σχέση με προηγούμενα μοντέλα της εταιρίας είναι συχνά προκαθορισμένη προδιαγραφή και επομένως σε κάποιες τουλάχιστον διαστάσεις της σχεδίασης ένα κινητό τηλέφωνο δεν μπορεί να σχεδιαστεί σε λευκό χαρτί.

Η μοντελοποίηση στο χαρτί γενικά θέλει προσοχή διότι ένας άπειρος μηχανικός μπορεί εύκολα να υποτιμήσει κάποια διάσταση μίας σχεδίασης. Για παράδειγμα, αν σχεδιάσουμε ένα φορητό ενσωματωμένο σύστημα για την διεργασία X μπορεί η απαιτούμενη υπολογιστική ισχύς να οδηγήσει σε χρόνο αυτονομίας απαράδεκτα μικρό. Σε αυτή την περίπτωση δεν είναι πρόβλημα ότι το μοντέλο είναι λάθος αλλά ότι είναι ελλιπές.

Από την άλλη μεριά η μοντελοποίηση στο χαρτί συχνά δίνει μία καλή ιδέα για ένα σύστημα και τα κεντρικά του μέρη, και παρότι δεν υποκαθιστά αναλυτικές προδιαγραφές για ένα σύστημα μπορεί να αποτελέσει βάση για αυτές. Μοντελοποίηση στο χαρτί δεν σημαίνει απαραίτητα block diagram ενός συστήματος. Μπορεί να είναι η σύλληψη της βασικής ιδέας ενός συστήματος. Για παράδειγμα, αν έχουμε κάποια ιδέα για ένα διαφορετικό τρόπο οδήγησης μίας οθόνης προβολής video θα μπορούσε η μοντελοποίηση στο χαρτί να είναι ένα διάγραμμα χρονισμού, απαλλαγμένο από περιορισμούς υλοποίησης.

### **3.7 Γλώσσες Μοντελοποίησης**

Στο σχεδιασμό ενσωματωμένων συστημάτων, υπάρχουν πολλές γλώσσες μοντελοποίησης. Οι πιο σημαντικές γλώσσες ή εργαλεία μοντελοποίησης που χρησιμοποιούνται είναι οι παρακάτω:

- **VHDL**

Η VHDL είναι μία γλώσσα περιγραφής υλικού, που χρησιμοποιείται για την ανάπτυξη ολοκληρωμένων ψηφιακών κυκλωμάτων και συστημάτων. Αρχικά η δημιουργία της αποσκοπούσε στη μοντελοποίηση και στην προσομοίωση κυκλωμάτων, γι' αυτό πολλά χαρακτηριστικά της γλώσσας έχουν σκοπό την προσομοίωση των λειτουργιών. Αργότερα, η γλώσσα χρησιμοποιήθηκε και ως εργαλείο σύνθεσης. Κατά τη σύνθεση, ο μεταγλωττιστής συνθέτει ένα πραγματικό κύκλωμα που ανταποκρίνεται με ακρίβεια στη λογική και χρονική περιγραφή, την οποία μοντελοποιεί ο κώδικας.

Ο όρος VHDL είναι συντόμευση των λέξεων VHSIC Hardware Description Language, όπου VHSIC σημαίνει Very High Speed Integrated Circuit. Η γλώσσα αυτή αναπτύχθηκε στις αρχές της δεκαετίας του 1980 με χρηματοδότηση από το υπουργείο Άμυνας των ΗΠΑ και έγινε πρότυπη το 1987 από το ινστιτούτο IEEE ως IEEE 1076. Αργότερα, δημιουργήθηκε μια βελτιωμένη έκδοσή της, η IEEE 1164 (1993). Ακολούθησαν και νεότερες αναβαθμίσεις του προτύπου, όπως η IEEE 1076-2002 και IEEE 1076-2008.

Ο κώδικας VHDL αποτελεί τον βασικό τύπο αρχείου που δέχονται ως είσοδο τα λογισμικά ψηφιακής σχεδίασης κυκλωμάτων (Computer Aided Design ή CAD), για τη δημιουργία σύνθετων ολοκληρωμένων κυκλωμάτων. Η γλώσσα VHDL χρησιμοποιείται ευρύτατα για την περιγραφή και υλοποίηση ψηφιακών συστημάτων σε προγραμματιζόμενες λογικές διατάξεις, τύπου CPLDs (Complex Programmable Logic Devices-Σύνθετες προγραμματιζόμενες λογικές διατάξεις) και FPGAs (Field Programmable Gate Arrays-Διατάξεις πυλών προγραμματιζόμενες στο πεδίο).

Επίσης, έχει καθιερωθεί σαν ένα πρότυπο (standard) στη σχεδίαση ηλεκτρονικών κυκλωμάτων ASICs (Application Specific Integrated Circuits). Η καθιέρωση της ως πρότυπο μας διαβεβαιώνει ότι και οι επόμενες εκδόσεις εργαλείων σχεδίασης θα υποστηρίζουν το πρότυπο αυτό. Έτσι, ένα κύκλωμα που περιγράφηκε και αναπτύχθηκε με τα σημερινά εργαλεία σχεδίασης θα είναι μεταφέρσιμο μελλοντικά σε νέα εργαλεία σχεδίασης, με ελάχιστες ή καθόλου αλλαγές.

- **Verilog**

Αναπτύχθηκε ως γλώσσα μοντελοποίησης υλικού από την εταιρεία Gateway Design Automation το 1984 για ιδιωτική χρήση. Η εταιρεία Cadence Design Systems αγόρασε την Gateway το 1990 και είναι υπεύθυνη για την προώθηση της Verilog ως γλώσσα μοντελοποίησης & προσομοίωσης. Η εταιρεία Synopsys είναι υπεύθυνη για την προώθηση της Verilog ως γλώσσα σύνθεσης. Επιτρέπει την περιγραφή (μοντελοποίηση) ψηφιακών συστημάτων.

- **SystemC**

Η SystemC, μια επέκταση της γλώσσας προγραμματισμού C++, γνωρίζει αυξημένη χρήση.

- **SystemVerilog**

Η SystemVerilog είναι ο πιο πρόσφατος διάδοχός της Verilog.

- **UML**

Η ενοποιημένη γλώσσα σχεδιασμού (*unified modeling language*) (UML) είναι μια γραφική γλώσσα για την οπτική παράσταση, τη διαμόρφωση προδιαγραφών και την τεκμηρίωση συστημάτων που βασίζονται σε λογισμικό. Η UML στοχεύει στο σχεδιασμό αντικειμενοστρεφών συστημάτων. Το σχέδιο είναι μια απλοποιημένη παράσταση της πραγματικότητας.

Σχεδιάζουμε για να μπορέσουμε να καταλάβουμε το σύστημα που αναπτύσσουμε. Έτσι δημιουργώντας ένα σχέδια επιτυγχάνουμε τέσσερις στόχους:

- παριστάνουμε οπτικά το σύστημα που έχουμε ή θέλουμε να κατασκευάσουμε,
- προσδιορίζουμε τη δομή και τη συμπεριφορά του συστήματος,
- δημιουργούμε ένα πρότυπο για να βασίσουμε την κατασκευή του συστήματος,
- τεκμηριώνουμε τις αποφάσεις που λάβαμε.

Σε όλους τους τεχνολογικούς τομείς ο σχεδιασμός βασίζεται σε τέσσερις βασικές αρχές:

- η επιλογή του είδους του σχεδίου έχει επίπτωση στον τρόπο και την μορφή επίλυσης του προβλήματος,
- όλα τα σχέδια εκφράζονται σε διαφορετικές βαθμίδες ακρίβειας,
- τα καλύτερα σχέδια σχετίζονται με την πραγματικότητα,
- ένα είδος σχεδίων δεν είναι ποτέ αρκετό.

Η UML περιλαμβάνει τρία βασικά στοιχεία:

- Οντότητες
- Σχέσεις
- Διαγράμματα

Η UML είναι μια πλήρης και πλούσια γλώσσα με εξαιρετικά ευρύ πεδίο εφαρμογής.

Η UML ορίζει τα παρακάτω διαγράμματα:

- Διάγραμμα περιπτώσεων χρήσης (*use case diagram*)
  - (i) Διαγράμματα δομής
  - (ii) Διάγραμμα κλάσεων (*class diagram*)
  - (iii) Διάγραμμα αντικειμένων (*object diagram*)

- Διαγράμματα συμπεριφοράς
  - (i) Διάγραμμα καταστάσεων (*statechart diagram*)
  - (ii) Διάγραμμα δραστηριοτήτων (*activity diagram*)
  - (iii) Διαγράμματα αλληλεπίδρασης
    - 1) Διάγραμμα ακολουθίας (*sequence diagram*)
    - 2) Διάγραμμα συνεργασίας (*collaboration diagram*)
- Διαγράμματα δομής υλοποίησης
  - (i) Διάγραμμα εξαρτημάτων (*component diagram*)
  - (ii) Διάγραμμα ανάπτυξης (*deployment diagram*)

- **MATLAB**

Το MATLAB είναι ένα πρόγραμμα για όσους χρησιμοποιούν αριθμητικούς υπολογισμούς, βασισμένο στη γραμμική άλγεβρα (πίνακες). Ξεκίνησε ως ένα πρόγραμμα «Εργαστηρίου Πινάκων» και έχει αναπτυχθεί αρκετά, για να γίνει ένα ισχυρότατο εργαλείο στην υλοποίηση προσομοιώσεων και μοντελοποιήσεων συστημάτων, στον προγραμματισμό, στην έρευνα, στην επιστήμη των μηχανικών και στις επικοινωνίες.

Η ισχυρή, τεχνική γλώσσα που χρησιμοποιεί είναι συμπαγής και περιγραφική και επιτρέπει τη μοντελοποίηση των διαφόρων συστημάτων με τη χρήση κώδικα που μαθαίνεται εύκολα.

- **EDA TOOLS**

Είναι μια κατηγορία εργαλείων λογισμικού για το σχεδιασμό ηλεκτρονικών συστημάτων, όπως πλακέτες τυπωμένων κυκλωμάτων και ολοκληρωμένων κυκλωμάτων. Τα εργαλεία που εργάζονται μαζί σε μια ροή του σχεδιασμού ότι οι σχεδιαστές chip χρησιμοποιούν για να σχεδιάσουν και να αναλύσουν ολόκληρο το chip ημιαγωγών.

Παρόλο που αυτές οι γλώσσες έχουν πολλά κοινά βασικά χαρακτηριστικά, ποικίλουν στα πιο προηγμένα χαρακτηριστικά τους. Επιπλέον, όλες εξελίσσονται, και προστίθενται νέα χαρακτηριστικά σε επόμενες εκδόσεις ώστε να ικανοποιηθούν οι εξελισσόμενες σχεδιαστικές προκλήσεις. Η επιλογή μεταξύ των γλωσσών αυτών συχνά υπαγορεύεται από τη διαθεσιμότητα εργαλείων και την οργανωτική παράδοση, καθώς επίσης και από το είδος της σχεδιαστικής δουλειάς που πρόκειται να γίνει.

### 3.8 Τρόποι Μοντελοποίησης

Τα εργαλεία που έχει στη διάθεσή του ένας μηχανικός για να μοντελοποιήσει ένα σύστημα συνήθως δέχονται διαφόρους τρόπους για την εισαγωγή της περιγραφής. Μπορεί να είναι είτε μια ειδική μορφή κειμένου σαν μια τυπική γλώσσα προγραμματισμού, όπως για παράδειγμα η VHDL ή η Verilog, είτε μια σχηματική μορφή που μοιάζει με διάγραμμα, όπως η UML, είτε πολύπλοκα περιβάλλοντα που έχουν και γραφική είσοδο και είσοδο σε μορφή προγράμματος, όπως το Matlab. Τον τελευταίο καιρό έχουν παρουσιαστεί και ειδικά περιβάλλοντα μοντελοποίησης που

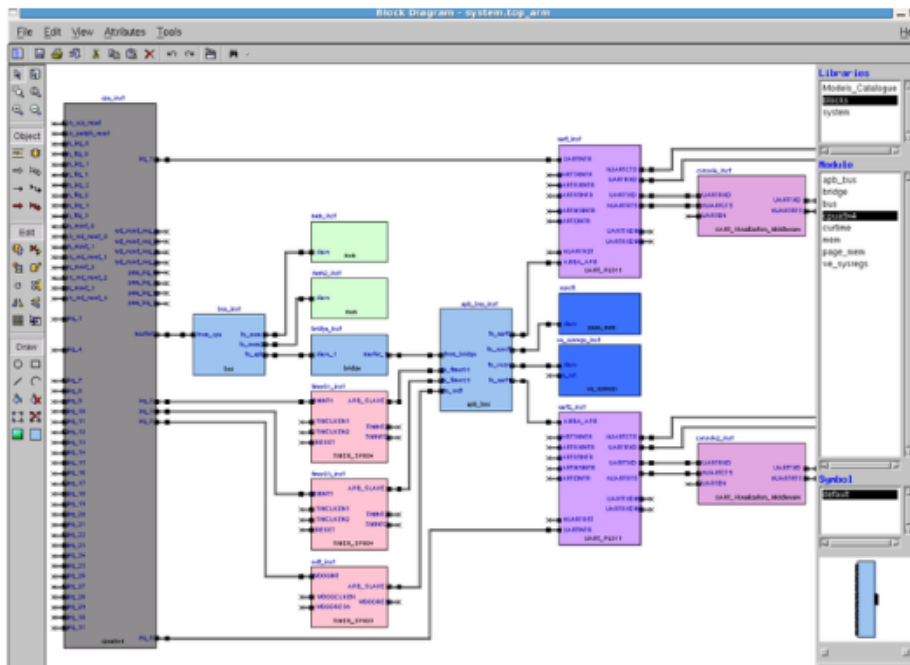
παρέχουν τη δυνατότητα στη σχεδιαστική ομάδα να δημιουργήσει ένα αφαιρετικό λειτουργικό μοντέλο πριν ακόμη σχεδιαστεί το σύστημα.

Αυτά τα περιβάλλοντα είναι ιδιαίτερα χρήσιμα σε αυτούς που σχεδιάζουν σύνθετα Ενσωματωμένα Συστήματα με διάφορους επεξεργαστές και δομοστοιχεία από πολλούς παρόχους, οπότε πριν αποφασίσουν και επιλέξουν αυτά τα στοιχεία, επιβεβαιώνουν τη συνεργασία τους. Ο σχεδιαστής μπορεί να περιγράψει τις οντότητες σε SystemC και να ορίσει τα πρότυπα επικοινωνίας. Το εργαλείο χρησιμοποιεί συνήθως μια υψηλού επιπέδου μοντελοποίηση, όπως το μοντέλο επιπέδου συναλλαγής (transaction level modeling, TLM), στο οποίο καθορίζονται τα δεδομένα που μεταφέρονται από μια οντότητα του συστήματος στην άλλη, και επίσης παρέχει μια βιβλιοθήκη από πολλά δομικά στοιχεία, όπως συγκεκριμένους επεξεργαστές, στοιχεία μνήμης, ελεγκτές, διαύλους και άλλα.

Αφού τοποθετηθούν όλα τα απαραίτητα στοιχεία με τις περιγραφές λειτουργίας και τις διασυνδέσεις, δημιουργείται ένα εκτελέσιμο πρόγραμμα, το οποίο μπορεί να επιτύχει υψηλές ταχύτητες προσομοίωσης (έχουν αναφερθεί προσομοιώσεις μεγάλης λεπτομέρειας, με ταχύτητες που προσεγγίζουν το 40% της πραγματικής εκτέλεσης του συστήματος). Αυτό το πρόγραμμα δέχεται εισόδους και δείχνει με γραφικό τρόπο τα αποτελέσματα, όπως για παράδειγμα το γράφο κλήσεων συναρτήσεων, τις πληροφορίες καθυστέρησης των συναρτήσεων, τους χρόνους εξυπηρέτησης διακοπών, την κίνηση δεδομένων στους διαύλους και άλλα.

Επειδή είναι εκτελέσιμο, ο μηχανικός μπορεί να επέμβει άμεσα και να τροποποιήσει παραμέτρους σε αυτό το μοντέλο μέσω μιας διεπαφής συμβατής με το GDB (gnu debugger) ή να αποσφαλματώσει και να διορθώσει τμήματα κώδικα με πρόσβαση μέσα στο εσωτερικό των μοντέλων και των διαύλων ή να δει σχετικά συνοπτικά γραφήματα εκτέλεσης και χρονισμού, όπως ρυθμό ολοκλήρωσης αποτελεσμάτων (throughput), καθυστέρηση επικοινωνίας (latency), κατανάλωση ενέργειας (δυναμική, στατική, ρολογιού), κατανομή ενέργειας (power distribution), χρήση διαύλου, μέγιστο, ελάχιστο και μέσο όρο του εύρους ζώνης διαύλων.

Όταν αποκτηθεί το πραγματικό hardware μπορεί να μεταφερθεί σε αυτό ο κώδικας που θα έχει αναπτυχθεί, αφού θα έχουν διορθωθεί τα σφάλματα του προγράμματος που βασίζονταν σε θέματα που είχαν μοντελοποιηθεί. Βέβαια, θα εμφανιστούν νέα σφάλματα τα οποία συνδέονται με το hardware, αλλά τουλάχιστον θα έχει μειωθεί σημαντικά ο χρόνος έρευνας και ανάπτυξης, αφού θα μπορεί να συνδέεται το μοντέλο με το αντίστοιχο hardware μέσω απομακρυσμένης αποσφαλμάτωσης (remote debugging) για να βρεθεί η πιθανή ασυνέπεια ή το πρόβλημα. Ένα αντιπροσωπευτικό παράδειγμα τέτοιου εργαλείου είναι το περιβάλλον εικονικής προτυποποίησης Vista Virtual Prototyping της εταιρίας Mentor όπως φαίνεται στο Σχήμα 4.8.



Σχήμα 3.8: Η μοντελοποίηση μπορεί να γίνει σε ιδεατά περιβάλλοντα, χωρίς πραγματικό hardware. Μια τέτοια πλατφόρμα είναι το Vista της Mentor

### 3.9 Μοντελοποίηση με Γενικά Εργαλεία (π.χ. MATLAB)

Ένα από τα προβλήματα που δύσκολα λύνονται είναι το πως θα μοντελοποιήσουμε ένα σύστημα με τρόπο που να κάνουμε σαν μηχανικοί καλύτερα την δουλειά μας στο να αναλύσουμε εναλλακτικές λύσεις πριν δεσμευθούμε σε μία από αυτές. Η μοντελοποίηση στο χαρτί κρύβει παγίδες γιατί δεν μπορούμε να διαστασιοποιήσουμε ένα σύστημα με σαφή τρόπο, δεν μπορούμε να αναλύσουμε το χρονικό και το χωρικό κόστος διαφορετικών αλγορίθμων υλοποίησης και δεν μπορούμε να δούμε, σε μη λειτουργική μορφή έστω, ολοκληρωμένο ένα σύστημα. Στο άλλο άκρο η μοντελοποίηση με γλώσσες περιγραφής υλικού μας δίνει «εκτελέσιμες προδιαγραφές», δηλαδή ένα μοντέλο το οποίο μέσα από την διαδικασία της σύνθεσης μπορεί να υλοποιηθεί πλήρως σε κάποια τεχνολογία (π.χ. VLSI, FPGA).

### 3.10 Μοντελοποίηση με Προγράμματα (π.χ. C, Java)

Οποιαδήποτε υψηλή γλώσσα προγραμματισμού είναι χρήσιμη σαν εργαλείο μοντελοποίησης στο βαθμό στον οποίο μας επιτρέπει να εξετάσουμε εναλλακτικά σενάρια (π.χ. αλγορίθμους υλοποίησης), να αναλύσουμε το πρόβλημα ποσοτικά (π.χ. βήματα/επαναλήψεις βρόχων που χρειάζονται για επίλυση του προβλήματος) και να συγκρίνουμε εναλλακτικές λύσεις (π.χ. ακρίβεια αποτελεσμάτων με αριθμητική 12 bit σε σχέση με κάποιο άλλο σύστημα που έχει ακρίβεια 32 bit). Οι συγκρίσεις αυτές είναι απαραίτητες ώστε να πάρουμε αποφάσεις γνωρίζοντας που ωφελούμαστε και πόσο από κάποια παρέμβαση, καθώς επίσης και κατά πόσον μία παρέμβαση είναι κρίσιμη ή όχι για μία σχεδίαση.

Καλή γνώση δομημένου προγραμματισμού είναι απαραίτητη για σχεδίαση ενσωματωμένων συστημάτων τόσο διότι χρειάζεται στην μοντελοποίηση ενός

συστήματος όσο και διότι η ανάπτυξη ενσωματωμένων συστημάτων σήμερα γίνεται με τέτοιες γλώσσες και εργαλεία (σπάνια χρησιμοποιείται η γλώσσα assembly).

### 3.11 Χρήση Εργαλείων CAD

Η έννοια του εργαλείου CAD είναι ιδιαίτερα παρανοημένη. Στην προηγούμενη ενότητα δώσαμε παραδείγματα από συγκεκριμένα εργαλεία CAD που κυκλοφορούν στο εμπόριο και χρησιμοποιούνται ευρύτατα. Θα ήταν αδύνατο να αρχίσουμε να απαριθμούμε εργαλεία, αλλά μπορούμε να επισημάνουμε μερικές γενικές αρχές.

Τα εργαλεία CAD δεν είναι απαραίτητο να έχουν γραφική είσοδο/έξοδο. Για παράδειγμα ο προσομοιωτής κυκλωμάτων Spice είναι ένα εργαλείο που κατ' αρχήν τρέχει σαν εντολή, δηλαδή από το command line. Για δική μας ευκολία υπάρχουν εκδόσεις που δέχονται γραφική αναπαράσταση των κυκλωμάτων και δίνουν σε γραφική μορφή τις κυματομορφές (π.χ. pSpice), αλλά αυτό δεν αλλάζει την βασική δομή του εργαλείου, δηλαδή την προσομοίωση ενός κυκλώματος που περιγράφεται με ένα netlist μορφής ASCII.

Εργαλεία CAD δεν είναι μόνο όσα πωλούνται σαν τέτοια από εταιρίες. Είναι οτιδήποτε αυτοματοποιεί μέρος της σχεδιαστικής διαδικασίας. Για παράδειγμα, αν θέλουμε να σχεδιάσουμε το τυπωμένο κύκλωμα για ένα σύστημα με εξαρτήματα που έχουν πάρα πολλούς ακροδέκτες (π.χ. 560) τότε το να συνδέουμε σε ένα γραφικό περιβάλλον τόσα σύρματα είναι χρονοβόρο. Μπορούμε να κάνουμε δύο Excel spreadsheets (xls) στα οποία να έχουμε στο μεν ένα στήλες με τον αριθμό κάθε ακροδέκτη και το όνομα του σήματος στο οποίο αντιστοιχεί, και στο άλλο να έχουμε στήλες για το σήμα της FPGA (με το όνομα του) που αντιστοιχεί σε κάποιο σήμα της σχεδίασής μας. Διαφορετικά μοντέλα κάθε οικογένειας έχουν εν γένει τα ίδια σήματα αλλά σε άλλους ακροδέκτες, κάτι που ισχύει ακόμη και για τροφοδοσία και γείωση. Η δημιουργία απλών αναφορών (reports) από το Excel είναι ακριβώς ένα netlist που δείχνει σε φυσικό επίπεδο την διασύνδεση ακροδεκτών με σήματα. Με αυτόν τον τρόπο χρειάζεται μία μόνο φορά να κάνουμε την είσοδο στην βιβλιοθήκη για κάθε διαφορετικό εξάρτημα και μία φορά για κάθε σχεδίαση. Η αλλαγή εξαρτήματος είναι πλέον απλή διαδικασία, με το να δημιουργήσουμε την κατάλληλη αναφορά και να την εισάγουμε σαν netlist στο εργαλείο για σχεδίαση τυπωμένου κυκλώματος.

Το να γράψουμε εμείς εργαλεία CAD είναι πολύ συνηθισμένο. Για παράδειγμα, αν θέλουμε να σχεδιάσουμε έναν full lookahead adder, μπορούμε από τις εξισώσεις να δημιουργήσουμε ένα πρόγραμμα (σε C/C++, PERL, Python, κλπ.) που να δημιουργεί ένα netlist πυλών, και μάλιστα για ευκολία αυτές μπορεί να έχουν απεριόριστο fan-in και fan-out. Κατόπιν, μπορούμε να δημιουργήσουμε ένα δεύτερο πρόγραμμα που μετατρέπει τις πύλες με απεριόριστο fan-in και fan-out σε πραγματικές πύλες δύο ή τριών εισόδων που εν γένει υπάρχουν σε βιβλιοθήκες. Η έξοδος του προγράμματος αυτού είναι το netlist του κυκλώματός μας που κατόπιν μπορούμε να εισάγουμε σε κάποιο εργαλείο για σύνθεση (προσέχοντας το format να είναι σωστό βέβαια).

Σε παλαιότερες εποχές οι σχεδιαστές hardware είχαν μεγάλη ευχέρεια στο να γράφουν εργαλεία CAD με προγράμματα που είχε το λειτουργικό σύστημα Unix (συνήθως sed/awk). Πιο πρόσφατα, γλώσσες όπως η PERL ή η Python βελτιώνουν περισσότερο τις δυνατότητες να αναπτυχθεί κάποιο εξειδικευμένο εργαλείο σε μικρό χρονικό διάστημα.

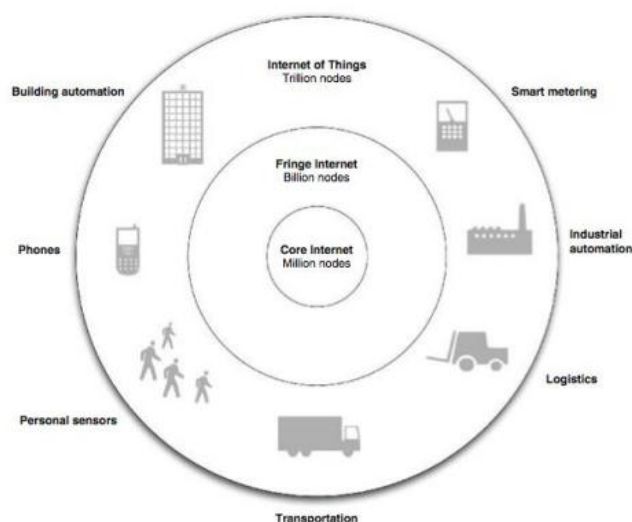


## ΚΕΦΑΛΑΙΟ 4

### 4. ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΜΕΤΡΗΣΕΩΝ ΚΑΙ ΕΛΕΓΧΟΥ ΓΙΑ ΤΟ ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ (Internet of Things – IoT)

Το Διαδίκτυο των πραγμάτων (IoT) είναι μια έννοια στην οποία ο εικονικός κόσμος της τεχνολογίας των πληροφοριών συνεργάζεται άψογα με τον πραγματικό κόσμο των πραγμάτων. Ο πραγματικός κόσμος γίνεται πιο προσιτός μέσω των υπολογιστών και στις επιχειρήσεις οι διαδικτυακές συσκευές σε καθημερινά σενάρια. Ωστόσο, το Διαδίκτυο των πραγμάτων είναι περισσότερα από ένα επαγγελματικό εργαλείο για πιο αποδοτικές και αποτελεσματικές διαδικασίες, θα μας επιτρέψει σε ένα πιο βολικό τρόπο ζωής.

Το διαδίκτυο των πραγμάτων είναι μια μελλοντική κατάσταση, κατά την οποία καθημερινά αντικείμενα – όπως κινητά τηλέφωνα, αυτοκίνητα, οικιακές συσκευές, ρούχα, ακόμη και τρόφιμα – θα συνδέονται ασύρματα στο διαδίκτυο μέσω έξυπνων μικροκυκλωμάτων και θα μπορούν να συλλέγουν και να ανταλλάσσουν δεδομένα. Σήμερα, ο μέσος άνθρωπος διαθέτει τουλάχιστον 2 αντικείμενα που συνδέονται με το διαδίκτυο, έως το 2015, ο αριθμός αυτός αναμένεται να αυξηθεί σε 7, με 25 δισεκατομμύρια συσκευές συνδεδεμένες ασύρματα παγκοσμίως. Έως το 2020, ο αριθμός αυτός μπορεί να διπλασιαστεί, φτάνοντας τα 50 δισεκατομμύρια. Για παράδειγμα, εάν ένας καθηγητής πανεπιστημίου ακυρώσει μια πρωινή διάλεξη λόγω ασθένειας, τα ξυπνητήρια των φοιτητών και οι καφετιέρες θα μπορούσαν να ρυθμιστούν αυτόματα, χαρίζοντάς τους μία επιπλέον ώρα ύπνου. Εάν ένα ηλικιωμένο άτομο ξεχάσει να πάρει ένα σημαντικό χάπι, θα μπορούσε να σταλεί ένα προειδοποιητικό μήνυμα σε κοντινό μέλος της οικογένειας, ή ακόμα και σε τοπικό κέντρο έκτακτης ανάγκης, έτσι ώστε κάποιος να μπορέσει να περάσει για ελέγξει την κατάσταση.



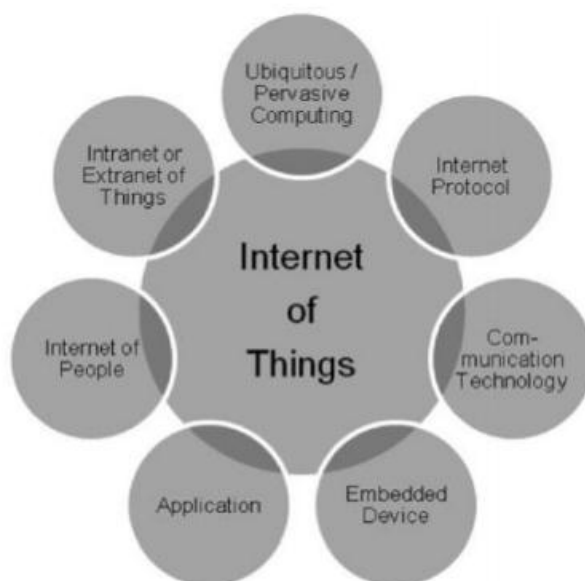
Σχήμα 4: Γενική Περιγραφή IoT

## 4.1 Ο Ορισμός του IoT

Ο όρος Ίντερνετ των πραγμάτων δεν σαφή και καθορισμένη έννοια. Ως εκ τούτου έχει χρησιμοποιηθεί ποικιλότροπος ως έννοια στον τομέα της επιστημονικής έρευνας, στις στρατηγικές μάρκετινγκ αλλά και τις πωλήσεις. Μέχρι σήμερα εξακολουθεί να είναι δύσκολο να καταλήξουμε σε ένα σαφή προσδιορισμό της έννοιας του.

Με τον όρο internet of things εννοούμε την διασύνδεση συσκευών και αισθητήρων με δυνατότητα πρόσβασης στο Internet με σκοπό την πραγματοποίηση μετρήσεων και συλλογή δεδομένων τα οποία στην συνέχεια θα επεξεργαστούν και θα χρησιμοποιηθούν άμεσα για ενημέρωση ενός ανθρώπου μίας επιχείρησης ή ενός άλλου μηχανήματος για την εκτέλεση κάποιας συγκεκριμένης λειτουργίας.

Χρησιμοποιείται ευρέως στην ιατρική στις μεταφορές και στις επικοινωνίες. Στο σύντομο μέλλον όλο και περισσότερες συσκευές θα συνδέονται μεταξύ τους με χαμηλό κόστος και θα σε βοηθούν στην καθημερινότητα σου χωρίς πολλές φορές να το γνωρίζεις.



Σχήμα 4.1: Οι διάφοροι τομείς έρευνας του Internet of Things

Ότι το Ίντερνετ των πραγμάτων είναι στενά συνδεδεμένα με περισσότερα του ενός πράγματα. Αυτά είναι ο άνθρωπος, οι εφαρμογές, τα ενσωματωμένα συστήματα, τα πρωτόκολλα του διαδικτύου, τα μικρής κλίμακας ή τοπικά δίκτυα, τα υπολογιστικά συστήματα και οι επικοινωνίες.

## 4.2 Η Ιστορία του Internet of Things

Το Internet of Things εξελίχθηκε με την γρήγορη διάδοση του ασύρματου internet & των ενσωματωμένων αισθητήρων και έτσι οι άνθρωποι άρχισαν να αντιλαμβάνονται ότι η τεχνολογία θα μπορούσε να είναι επαγγελματικό εργαλείο αλλά και προσωπικό.

Ο όρος “Internet of Things” (ή αλλιώς Διαδίκτυο των Πραγμάτων) επινοήθηκε στα τέλη της δεκαετίας του 1990 από τον επιχειρηματία Kevin Ashton. Ο Ashton, ο οποίος είναι ένας από τους ιδρυτές του Auto-ID Center στο MIT, ήταν μέρος μιας ομάδας που ανακάλυψε τον τρόπο να συνδέσει τα αντικείμενα με το διαδίκτυο μέσω μιας ετικέτας RFID. Έχει δηλώσει ότι χρησιμοποίησε πρώτη φορά τη φράση “Internet of Things” σε μια παρουσίαση που έκανε το 1999 και ο όρος αυτός έχει κολλήσει από τότε.

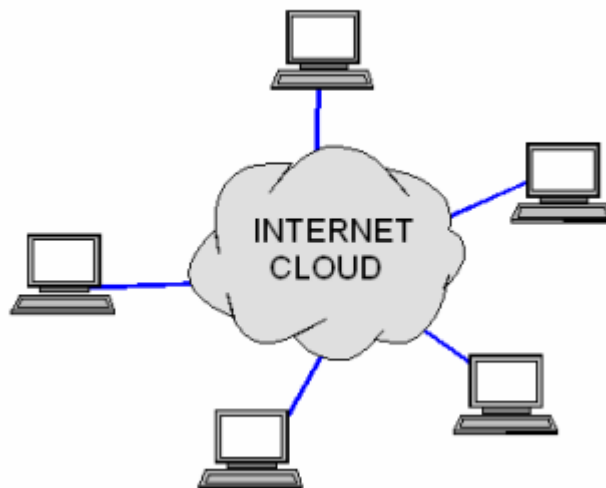
## 4.3 Τάσεις και Στοιχεία του IoT

Το διαδίκτυο των πραγμάτων έχει αναγνωριστεί ως μια από τις πιο αναδυόμενες τεχνολογίες στο τομέα της πληροφορικής. Έχει προβλεφθεί ότι για το IoT, θα χρειαστούν περισσότερο από 10 χρόνια για να υιοθετηθεί πλήρως από την αγορά.

Μια Ταξινόμηση που θα μας βοηθούσε να καθορίσουμε τα στοιχεία που απαιτούνται για το Διαδίκτυο των πραγμάτων είναι τα παρακάτω τρία. Ένα στοιχείο σίγουρα είναι υλικό εξοπλισμός (Hardware), δηλαδή οι αισθητήρες, οι ενεργοποιητές, τα ενσωματωμένα σύστημα επικοινωνιών. Το επόμενο στοιχείο που είναι απαραίτητο είναι η συλλογή, αποθήκευση, η επεξεργασία των δεδομένων, τα υπολογιστικά εργαλεία γενικά (Middleware). Τελευταίο χαρακτηριστικό είναι η απεικόνιση (ή παρουσίαση - Presentation) των στοιχείων και καταστάσεων, που θα πρέπει να είναι ευρέως προσβάσιμα από διαφορετικές πλατφόρμες.

## 4.4 Cloud Networking

Οι τοπολογίες οι οποίες μπορούν να δικτυωθούν τα αισθητήρια είναι πολλές. Μια τοπολογία από αυτές που μπορεί να εφαρμοστεί και είναι αποδεκτό ότι ίσως να αποτελέσει τη βασική δικτύωση στο μέλλον είναι το Cloud networking. Το Cloud networking είναι η διασύνδεση των στοιχείων που την απαρτίζουν ώστε να ανταποκριθεί στις απαιτήσεις που είναι συνυφασμένες με τον υπολογισμό cloud. Η Cloud δικτύωση επιτρέπει στους χρήστες να επιλέξουν ένα τεράστιο δίκτυο υπολογιστών που μπορεί να έχει πρόσβαση από μεγάλες αποστάσεις από ένα κινητό τηλέφωνο, φορητό υπολογιστή ή φορητή συσκευή για πληροφορίες ή δεδομένα.



**Σχήμα 4.4**

Επιπρόσθετα, η δικτύωση σύννεφου είναι η δικτύωση των υποδομών που απαιτούνται για την υποστήριξη του cloud computing, το οποίο απαιτεί βελτίωση του δικτύου σε επεκτασιμότητα, αξιοπιστία, και στην καθυστέρηση πέρα από ότι τα παραδοσιακά δίκτυα έχουν προσφέρει. Η διάσταση των υπολογιστικών αναγκών του σύννεφου υπολογιστών δικτύου είναι τουλάχιστον μία τάξη μεγέθους μεγαλύτερη από ότι για τα παραδοσιακά δίκτυα.

#### **4.5 Cloud ή Cloud Computing**

Υπάρχουν διάφοροι ορισμοί και ερμηνείες για το «Cloud» ή το «Cloud Computing». Ο έννοια του ορίζεται ανάλογα το πεδίο χρήσης. Εμείς θα δούμε ένα γενικό ορισμό και το συνδυασμό του με το Διαδίκτυο των πραγμάτων.

Ένας ευρύτερος τρόπος που μπορούμε να ορίσουμε είναι ο παρακάτω: Το Cloud ή Cloud Computing αποτελεί μια νέα προσέγγιση στο χώρο των καταναμημένων συστημάτων η οποία όμως χρησιμοποιεί και τεχνολογίες που προϋπήρχαν. Σκοπός του είναι η παροχή υπηρεσιών πόρων όπως, η υπολογιστική ισχύ και η αποθηκευτική ικανότητα δυνατότητα, στους χρήστες του συστήματος. Σχετικά με τον ορισμό της έννοιας έχουν γίνει πολλές προσπάθειες οι οποίες όμως δεν καλύπτουν όλες τις πτυχές του συστήματος με αποτέλεσμα την γενίκευση της έννοιας με τέτοιων τρόπο που να θεωρείται κάθε σύστημα το οποίο επιτρέπει ανάθεση υπολογιστικών και αποθηκευτικών υπηρεσιών εξωτερικά.

Με άλλα λόγια, τα Clouds είναι κυρίως πλατφόρμες που επιτρέπουν την εκτέλεση διάφορων μορφών εργασιών, σε συστήματα με πολλούς πόρους. Πρόκειται για μία τεχνολογία που χρησιμοποιεί το διαδίκτυο και κεντρικούς απομακρυσμένους εξυπηρετητές (servers) για τη διατήρηση μεγάλου όγκου δεδομένων και εφαρμογών.

Γενικά, Cloud computing είναι διαδικτυακή εφαρμογή και χρησιμοποιεί την τεχνολογία των υπολογιστών ( "υπολογισμούς"), σύμφωνα με την οποία δυναμικά

κλιμακωτοί , συχνά εικονικοί πόροι και υπηρεσίες παρέχονται μέσω του Διαδικτύου. Οι χρήστες δεν χρειάζεται να έχουν γνώση, εμπειρία και έλεγχο επί της τεχνολογικής υποδομής "στο σύννεφο" που υποστηρίζει την διαδικασία. Όπως βλέπουμε και στο παρακάτω Σχήμα 4.5.



Σχήμα 4.5

Η έννοια ενσωματώνει το λογισμικό ως υπηρεσία , Web 2.0 και άλλες πρόσφατες, με την οποία το κοινό 'θέμα' (εφαρμογή) είναι προσβάσιμο από το Διαδίκτυο για την ικανοποίηση των αναγκών των υπολογιστών των χρηστών. Ένα συχνά αναφερόμενο παράδειγμα είναι το Google Apps, το οποίο παρέχει κοινές επιχειρηματικές εφαρμογές online που είναι προσβάσιμο από τον web browser, ενώ το λογισμικό και τα δεδομένα είναι αποθηκευμένα σε διακομιστές της Google.

Επιτρέπει στους καταναλωτές και στις επιχειρήσεις να χρησιμοποιούν εφαρμογές χωρίς να χρειάζονται εγκατάσταση αλλά και να έχουν πρόσβαση σε όλα τους τα αρχεία οποιαδήποτε στιγμή και σε οποιοδήποτε μέρος και αν βρίσκονται έχοντας μόνο έναν υπολογιστή συνδεδεμένο στο διαδίκτυο. Με άλλα λόγια, το cloud ή cloud computing προσφέρει την Πληροφορική Τεχνολογία (Information Technology) ως υπηρεσία (IT-as-a-Service). Αντί να δημιουργήσει κανείς ολόκληρη τεχνολογική υποδομή για να φιλοξενεί βάσεις δεδομένων και λογισμικά, τρίτοι αναλαμβάνουν τη συγκεκριμένη φιλοξενία σε μεγάλες εκτάσεις με servers.

#### 4.5.1 Τύποι Cloud

Πάροχοι Cloud συνήθως επικεντρώνονται σε ένα είδος λειτουργικότητας. Τέτοιες κατηγορίες είναι: Υποδομή, πλατφόρμα ή εφαρμογών ως υπηρεσία, αν και δεν υπάρχει και κανένας περιορισμός για να προσφέρουν πολλούς τύπους ταυτόχρονα, η οποία μπορεί συχνά να παρατηρηθεί σε PaaS (Platform as a Service), πολλοί πάροχοι

που προσφέρουν επιπλέον εφαρμογές, όπως το Google App Engine σε συνδυασμό με το Google Docs.

Ορισμένες πεδία εφαρμογών αλληλοεπικαλύπτονται και είναι επομένως δύσκολο να διακριθούν. Ως ένα παράδειγμα, οι πλατφόρμες τυπικά πρέπει να παρέχουν πρόσβαση σε πόρους έμμεσα, και συνεπώς είναι μερικές φορές συγχέονται με τις υποδομές.

Η λίστα που ακολουθεί προσδιορίζει τα βασικά είδη των Clouds (επί του παρόντος σε χρήση):

- **(Cloud) Infrastructure as Service (IaaS)** που αναφέρεται επίσης ως σύννεφα πόρων, την παροχή πόρων σαν υπηρεσία για τον χρήστη. Με άλλα λόγια, παρέχουν ουσιαστικά ενισχυμένες δυνατότητες εικονικοποίησης (virtualization). Ως εκ τούτου, διαφορετικοί πόροι μπορούν να παρέχονται μέσω μιας διεπαφής της υπηρεσίας (δεδομένα και αποθήκευση). Αυτού το είδος το σύννεφο ασχολείται με τη αξιόπιστη πρόσβαση στα δεδομένα, σε δυναμικά μεγέθη πληροφορίας με ποιοτικά αποτελέσματα.
- **Compute Clouds**, παρέχουν πρόσβαση σε υπολογιστικούς πόρους (π.χ. CPUs). Μέχρι στιγμής όμως, το χαμηλό επίπεδο πόρων δεν μπορεί πραγματικά να το αξιοποιήσει, προς δικό του όφελος, έτσι ώστε να εκτίθενται ως μέρος ενός «εικονικού περιβάλλοντος».
- **(Cloud) Platform as a Service (PaaS)** παρέχει υπολογιστικούς πόρους μέσω μιας πλατφόρμας πάνω στην οποία εφαρμογές και οι υπηρεσίες μπορούν να αναπτυχθούν και να φιλοξενηθούν. Το PaaS κάνει συνήθως χρήση ειδικών APIs για τον έλεγχο της συμπεριφοράς ενός server hosting engine, που εκτελεί και αναπαράγει σύμφωνα με τα αιτήματα των χρηστών.
- **(Clouds) Software as a Service (SaaS)** επίσης μερικές φορές αναφέρεται ως Υπηρεσία. Τα Σύννεφα Εφαρμογών ως υπηρεσία προσφέρουν εφαρμογές με συγκεκριμένες δυνατότητες και διαδικασίες. Χρησιμοποιούν τις δυνατότητες του cloud, δηλαδή χρησιμοποιούν υποδομή PaaS ή IaaS για να λειτουργήσουν, αντί να παρέχουν οι ίδιοι δυνατότητες τέτοιες. Συχνά, το λογισμικό της εφαρμογής προσφέρεται μέσα σε ένα σύννεφο.

#### 4.5.2 Κύρια Χαρακτηριστικά του Cloud Computing

Για τους τελικούς χρήστες, υπάρχουν πέντε βασικά πλεονεκτήματα:

- 1) Πολύ μικρή επένδυση κεφαλαίων που απαιτούνται για την ανάπτυξη εφαρμογών σε κλιμακωτό επίπεδο.
- 2) Τα προγράμματα και τα δεδομένα διανέμονται για να μπορούν να γίνουν κλιμακωτά-επεκτάσιμα και ανεκτά σε βλάβες.
- 3) Τα προγράμματα μπορούν να προσπελαστούν από οποιαδήποτε συσκευή, συμπεριλαμβανομένων των ασύρματων και κινητών επικοινωνιών.
- 4) Καταργεί παραδοσιακούς server και θέματα διαχείρισης αποθήκευσης και Συντήρησης.

5) Επιτρέπει την πρόσβαση σε εφαρμογές και να διαμοιράσουν πληροφορίες.

Για τους προμηθευτές, υπάρχουν επίσης τέσσερα βασικά πλεονεκτήματα:

- 1) Η ανθεκτικότητα και η σταθερότητα των πλατφόρμων σύννεφων επιτρέπει το χαμηλότερο κόστος υλικού
- 2) Με τον διαμοιρασμό των χρήσεων του υλικού αυξάνεται το κέρδος και η απόδοση χρήσης.
- 3) Επιτρέπει νέα και πολύ οικονομικά αποδοτικά επιχειρηματικά μοντέλα.
- 4) Τα κέντρα δεδομένων μπορούν να βρίσκονται οπουδήποτε στη χώρα.

#### 4.5.3 Προσβασιμότητα και Χρήση του Cloud

Παρόμοια με P/I/SaaS, τα σύννεφα μπορούν να φιλοξενήσουν και να χρησιμοποιηθούν για διάφορους σκοπούς, ανάλογα με την περίπτωση χρήσης. Μέχρι στιγμής, υπήρχε μια τάση τα σύννεφα να εξελιχθούν κυρίως ιδιωτικά και για εσωτερικές λύσεις (private clouds), για τη διαχείριση των τοπικών υποδομών και συνολικά την εξυπηρέτηση των χρηστών. Με αυτό τον τρόπο εξασφαλίζεται η διαθεσιμότητα των δεδομένων στο σύνολο των χρηστών. Αυτό αύξησε τη παραγωγικότητα και την ευκολία στη χρήση. Η Φυσική εξέλιξη είναι τα δημόσια σύννεφα ανάλογα και του σκοπούς που εξυπηρετούν και πλέον που η τεχνολογία είναι αρκετά ώριμοι να υποστηριχθούν με ασφάλεια τα υβριδικά. Δηλαδή ο συνδυασμός δημόσιου και ιδιωτικού, με έλεγχο στην προσβασιμότητα, κατά περίπτωση.

Μπορούμε συνεπώς διακρίνει τους ακόλουθους τύπους:

- Οι **Private Clouds** τυπικά ανήκουν στη επιχείρηση και είναι ένα ιδιωτικό ή μισθωμένα σύννεφο. Οι λειτουργίες δεν είναι άμεσα εκτεθειμένες στον πελάτη, αν και σε ορισμένες περιπτώσεις, μερικά χαρακτηριστικά από τις υπηρεσίες cloud μπορεί να προσφέρονται - αυτό είναι παρόμοιο με (Cloud) SaaS από την άποψη του πελάτη.
- **Public Clouds** : Οι επιχειρήσεις προσφέρουν τις λειτουργίες του σύννεφου, ώστε να μπορούν να χρησιμοποιούν από όλους. Παρέχουν τις υπηρεσίες σε χρήστες εκτός της εταιρείας. Παρέχει στο χρήστη την ικανότητά να αξιοποιήσει τις δυνατότητες του cloud για δικούς του σκοπούς, επίσης, επιτρέπει σε επιχειρήσεις να χρησιμοποιήσουν τις υπηρεσίες cloud, μειώνοντας έτσι το κόστος και την προσπάθεια για τη δημιουργία της δικής τους υποδομής. Όπως σημειώνεται ότι οι τύποι σύννεφου, το πεδίο εφαρμογής και λειτουργιών μπορεί να διαφέρουν κατά πολυ.
- **Hybrid Clouds**: Αν και τα δημόσια clouds επιτρέπουν στις επιχειρήσεις να χρησιμοποιούν υποδομές δωρεάν από τους πάροχους, σε ορισμένες περίπτωση χάνουν τον έλεγχο από σημαντικές πληροφορίες ή ευαίσθητα δεδομένα. Αυτό δεν είναι πάντα επιθυμητό από την εκάστοτε επιχείρηση. Υβριδικά σύννεφα αποτελούνται ένα μίγμα λειτουργίας των ιδιωτικών και δημόσιων υποδομών cloud, έτσι ώστε να επιτευχθεί το μέγιστο της μείωσης του κόστους μέσω της

εξωτερικής ανάθεσης, διατηρώντας ταυτόχρονα τον επιθυμητό βαθμό ιδιωτικότητας.

Για παράδειγμα τα ευαίσθητα δεδομένα χρησιμοποιούν το ιδιωτικό τοπικό σύννεφο.

#### 4.5.4 Cloud και IoT

Υπάρχει μια διαφωνία για το αν το Διαδίκτυο των πραγμάτων συνδέεται με τα συστήματα Cloud σε όλες της περιπτώσεις. Ενώ το Διαδίκτυο των πραγμάτων ασχολείται με θέματα που σχετίζονται με την ελαστικότητα, την αξιοπιστία και διαχείριση των δεδομένων, ταυτόχρονα υπάρχει η παραδοχή ότι οι πόροι στο σύννεφο είναι αυτοί που υπολογίζουν, φιλοξενούν και διαχειρίζονται τα δεδομένα. Με αυτό τον τρόπο το σύννεφο στο Διαδίκτυο των πραγμάτων αναπαριστάτε με μια εικονική πλατφόρμα επεξεργασίας.

Τα σύννεφα μπορούν να προσφέρουν ζωτικής σημασίας υποστήριξη στο διαδίκτυο των πραγμάτων, μπορούν να ασχοληθούν ευέλικτα με μεγάλη ποσότητα δεδομένων που προέρχονται από ποικιλία αισθητήρων και πραγμάτων. Συνολικά, το Διαδίκτυο των πραγμάτων μπορούν να επωφελούνται από τα συστήματα Cloud, αλλά δεν υπάρχει άμεση σχέση μεταξύ των δύο. Ωστόσο, υπάρχουν σημεία επαφής που δεν πρέπει να αγνοηθούν. Η διαχείριση των δεδομένων και οι διεπαφές μεταξύ των αισθητήρων και των συστημάτων cloud δείχνουν ομοιότητες.



Σχήμα 4.5.4: Η τοποθέτηση του Cloud στο κέντρο του οικοσυστήματος.

Το όραμα του Διαδικτύου των πραγμάτων μπορεί να το δει κανείς από δύο οπτικές γωνίες, η πρώτη οπτική γωνία είναι το διαδίκτυο και η δεύτερη τα πράγματα. Η αρχιτεκτονική του διαδικτύου περιλαμβάνει τις διαδικτυακές υπηρεσίες ως κύριο σκοπό, ενώ τα δεδομένα συλλέγονται από τα αντικείμενα. Από την οπτική γωνία των αντικειμένων, ο κύριος ρόλος τους είναι να είναι έξυπνα. Στην Εικόνα φαίνεται μια γενική άποψη του πλαισίου που ενσωματώνει τις έξυπνες συσκευές με το διαδίκτυο.



Προκειμένου να αξιοποιηθεί πλήρως οι δυνατότητες του cloud computing, το σύννεφο πρέπει να τοποθετηθεί στο κέντρο του πλαισίου αυτού, όπου είναι και το πιο βιώσιμο. Αυτό όχι μόνο δίνει ευελιξία στη διαίρεση των δαπανών με λογικό τρόπο, αλλά είναι επίσης εξαιρετικά επεκτάσιμο.

Οι πάροχοι υπηρεσιών μετρήσεων μπορούν να ενταχθούν στο δίκτυο και να προσφέρουν τα δεδομένα τους, χρησιμοποιώντας το σύννεφο αποθήκευσης. Οι προγραμματιστές μπορούν να παρέχουν εργαλεία λογισμικό. Οι ειδικοί τεχνητής νοημοσύνης μπορούν να παρέχουν την μεθοδολογία για την χρήση της πληροφορίας σε γνώση, ενώ οι σχεδιαστές γραφικών να προσφέρουν εργαλεία και σχέδια για ολοκλήρωση των συστημάτων. Το cloud computing μπορεί να προσφέρει αυτές τις υπηρεσίες, όπως υποδομές, πλατφόρμες και λογισμικό όπου εξαρτάται από την ανθρώπινη δημιουργικότητα.

Η αξιοποίηση των δυνατοτήτων του Διαδικτύου των πραγμάτων μπορεί να γίνει σε διάφορα πεδία εφαρμογών. Εντούτοις, η ανάπτυξη εφαρμογών για το IoT χρησιμοποιώντας χαμηλού επιπέδου μοντέλα προγραμματισμού Cloud και διασυνδέσεων είναι πολύπλοκη. Για να ξεπεραστεί αυτός ο περιορισμός, χρειαζόμαστε μια IoT εφαρμογή ειδικού πλαισίου για την ταχεία δημιουργία εφαρμογών και η τοποθέτησή τους σε Cloud υποδομές.

Κάποια απλά παραδείγματα υπηρεσιών cloud για IoT είναι τα παρακάτω:

Το open.sen.se (<http://open.sen.se>), το οποίο είναι μια ανοιχτή πλατφόρμα συννέφου για το διαδίκτυο των πραγμάτων. Οπού μπορούμε να αποθηκεύσουμε δεδομένα ή να τα τραβήξουμε. Μπορούμε να προωθήσουμε σε άλλες υπηρεσίες ή μέσα. Στα δεδομένα και στοιχεία που έχουμε σώσει μπορούμε να κάνουμε υπολογισμούς και πράξεις, να βγάλουμε μέσους όρους ή να κρατήσουμε ιστορικό μετρήσεων.



Είναι ένας τόπος όπου ο άνθρωπος, οι συσκευές, το ίντερνετ και το περιβάλλον, αλληλεπιδρούν μεταξύ τους.

Μία άλλη ιστοσελίδα που προσφέρει υπηρεσίες σύννεφου για το διαδίκτυο των πραγμάτων, σε επαγγελματίες και ερασιτέχνες είναι το everything (<http://www.everything.com>).



Με μια μεγάλη ποικιλία από εργαλεία όπως αποθήκευση δεδομένων, ελέγχους πρόσβασης, υπολογιστικές ικανότητες, άλλα και πολλά έτοιμα APIs (Application Programming Interfaces), μπορούν να βρουν αμέτρητες εφαρμογές σε αντικείμενα τα φυσικού κόσμου.

#### **4.6 Εφαρμογές**

Υπάρχουν διάφορα πεδία εφαρμογών που θα επηρεαστούν από την αναδυόμενη αγορά του IoT. Οι εφαρμογές μπορούν να ταξινομηθούν με βάση του τύπου δικτύου που χρησιμοποιούν, την κάλυψη, την κλίμακα, την επαναληψιμότητα, την συμμετοχή των χρηστών και τον αντίκτυπο τους. Έτσι κάποιες από τις κατηγορίες είναι:

- Προσωπική ενός ατόμου δηλαδή, για παράδειγμα στο σώμα μας, αυτή που μπορεί να φορεθεί (wearable smart devices).
- Οικιακή, οι έξυπνες ηλεκτρονικές συσκευές που μπορούμε να συναντήσουμε και να έχουν εφαρμογή στο σπίτι μας.
- Κοινωνίας, εφαρμογές που μπορούν να έχουν εφαρμογές σε κλίμακας μια πόλης (όπως για παράδειγμα στις μεταφορές)
- Βιομηχανία και περιβάλλον, από εμπορικές εφαρμογές για την καλύτερη ποιότητα προϊόντων άλλα και καλύτερη διαχείριση τις διαδικασίας εφοδιασμού των προϊόντων, μέχρι συσκευές για περιβαλλοντικές μετρήσεις.

Οι κατηγορίες μπορεί να είναι περισσότερες, μπορεί να ταξινομηθούν με διαφορετικά χαρακτηριστικά, όπως και πολλές από τις εφαρμογές μπορούν να ενταχθούν σε περισσότερες από μια κατηγορίες.

##### **4.6.1 Εφαρμογές Για Τον Άνθρωπο**

Είναι όλες εκείνες οι εφαρμογές που σκοπό έχουν στην εξυπηρέτηση ή πληροφόρηση ενός άτομο. Τέτοιες στο διαδίκτυο μπορεί να βρει κανείς εκατοντάδες, υπάρχουν από ερασιτεχνικές μέχρι εμπορικές εφαρμογές, όλες τους πολλά υποσχόμενες. Τέτοιες είναι ατομικές ζυγαριές βάρους με σύνδεση wi-fi, μετρητές

καρδιακών παλμών, αισθητήρες πτώσης για κράνος, ρούχα που ελέγχουν την υγεία του ατόμου που τα φοράει, έξυπνα ρολόγια και ότι μπορεί να φανταστεί κάνεις.

#### **4.6.2 Εφαρμογές Για Το Σπίτι**

Σε αυτή την ενότητα είναι όλες εκείνες οι λύσεις που μας παρέχονται σε μια οικία γενικότερα, τόσο εντός όσο και στο περιβάλλον του χώρο. Κάνουν την καθημερινότητα μας ευκολότερη στο χειρισμό, στον έλεγχο αλλά και την πληροφόρηση που έχουμε ανά πάσα στιγμή από τις ηλεκτρονικές συσκευές, για την κατάσταση του σπιτιού. Τέτοιες εφαρμογές που σύνταμε σε έξυπνα σπίτια είναι ο φωτισμός, η θερμοκρασία, ο συναγερμός, έξυπνες οικιακές ηλεκτρικές συσκευές καθώς και κλειδαριές, σε όλες αυτές τις εφαρμογές, που είναι μόνο μερικές από αυτές που υπάρχουν διαθέσιμες στο εμπόριο, μπορούμε να πληροφορηθούμε από απόσταση (μέσω διαδικτύου), να αλλάξουμε την κατάσταση τους, να ενεργοποιήσουμε ή να απενεργοποιήσουμε όποια αυτές θέλουμε.

#### **4.6.3 Εφαρμογές Πόλης**

Σε κλίμακα πόλης οι εφαρμογές έχουν ένα προσανατολισμό, για την καλύτερη ζωή των πολιτών της σε αυτή. Οι εφαρμογές των ενσωματωμένων συστημάτων στο διαδίκτυο, έχουν να κάνουν κλιματολογικές συνθήκες που επικρατούν, την ποιότητα του αέρα που αναπνέουμε, τα επίπεδα θορύβου, την συλλογή των σκουπιδιών, μετρητές κινήσεις ανθρώπων και αυτοκινήτων, έξυπνες πινακίδες και πολλά ακόμη που προσαρμόζονται με τις τάσεις τις κοινωνίας.

#### **4.6.4 Εφαρμογές Στη Βιομηχανία Και Το Περιβάλλον**

Οι εφαρμογές σε αυτή την κατηγορία είναι πολλές, όπως και τα οφέλη για τους πολίτες ή τους καταναλωτές. Η βιομηχανία μπορεί να μειώσει το κόστος, να παρέχει καλύτερες και πιο αξιόπιστες υπηρεσίες προς τους πολίτες αυτού του κόσμου. Υπάρχει μια μεγάλη γκάμα οπου και μπορούμε να συναντήσουμε τέτοια έξυπνα ενσωματωμένα συστήματα που διασυνδέονται στο δίκτυο. Τέτοιες εφαρμογές μπορεί να είναι :

- Οι συνθήκες μεταφοράς εμπορευμάτων στην ναυτιλία
- Η μόλυνση στον αέρα που αναπνέουμε και τη θάλασσα.
- Έγκυρη ειδοποίηση για πυρκαγιές.
- Έξυπνο παρκάρισμα των αυτοκινήτων αλλά και έλεγχος της κυκλοφοριακή συμφόρησης.
- Αισθητήρες που θα ελέγχουν την ποιότητα των κατασκευών στο πέρας του χρόνου.
- Στις καλλιέργειες ή γενικότερα στο πρωτογενή τομέα με έξυπνα σύστημα έλεγχου της παραγωγής.

## ΚΕΦΑΛΑΙΟ 5

### 5. ΠΡΩΤΟΚΟΛΛΑ ΚΑΙ ΔΙΚΤΥΑ ΓΙΑ ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Στην Ιστορία των δικτύων έχουν αναπτυχθεί πολλά μοντέλα αναφοράς δικτυακής αρχιτεκτονικής τα οποία παρουσιάζουν διαφορές μεταξύ τους. Το πρώτο μοντέλο αναπτύχθηκε από τον Διεθνή Οργανισμό Τυποποίησης (International Standards Organisation, ISO) και αποτέλεσε το πρώτο μεγάλο βήμα προς την κατεύθυνση της τυποποίησης. Ονομάζεται μοντέλο αναφοράς OSI (Open System Interconnection) διότι ασχολείται με συνδέσεις ανοιχτών (προς τις διάφορες εταιρείες) συστημάτων.

Το μοντέλο OSI (Open Systems Interconnection) είναι μια ιεραρχική δομή επτά επιπέδων που καθορίζει τις προδιαγραφές επικοινωνίας μεταξύ δύο υπολογιστών, ορίζοντας επακριβώς τον σκοπό κάθε επιπέδου αλλά και τα χρησιμοποιούμενα πρωτόκολλα, και τυποποιήθηκε ως πρότυπο ISO 7498-1. Θεωρήθηκε ότι θα επέτρεπε τη λειτουργική συνεργασία μεταξύ ποικίλων ψηφιακών συσκευών που ήταν διαθέσιμες στην αγορά. Το μοντέλο επιτρέπει σε όλα τα στοιχεία ενός δικτύου να συλλειτουργούν, με κάθε στοιχείο να υλοποιεί ένα ή περισσότερα πρωτόκολλα δικτύωσης, ανεξάρτητα από το ποιος είναι ο κατασκευαστής τους. Περί τα τέλη της δεκαετίας του 1980 ο ISO συνιστούσε την εφαρμογή του μοντέλου OSI ως κοινώς αποδεκτού υποδείγματος σχεδιασμού δικτύων.

Ωστόσο εκείνη την εποχή η στοίβα πρωτοκόλλων TCP/IP, η οποία βασιζόταν σε ελαφρώς διαφορετική διαστρωμάτωση επιπέδων, ήταν ήδη επί πολύ καιρό σε ευρεία χρήση. Το TCP/IP ήταν θεμελιώδες για το δίκτυο ARPANET και τα άλλα δίκτυα που εξελίχθηκαν στο σημερινό Διαδίκτυο. Ως αποτέλεσμα το μοντέλο OSI παραμερίστηκε και σήμερα μόνο ένα υποσύνολό του χρησιμοποιείται ακόμη. Η επικρατούσα αντίληψη είναι ότι οι περισσότερες προδιαγραφές του είναι περίπλοκες και η πλήρης λειτουργικότητά του θα χρειαζόταν μεγάλο χρόνο κατασκευής, αν και συνεχίζουν να υπάρχουν σθεναροί υποστηρικτές του.

#### 5.1 Τι είναι Πρωτόκολλο;

Ως Πρωτόκολλο επικοινωνίας ορίζεται ένα σύνολο κανόνων συμφωνημένων και από τα δυο επικοινωνούντα μέρη και που εξυπηρετούν την μεταξύ τους ανταλλαγή πληροφοριών. Το πρωτόκολλο επικοινωνίας είναι δηλαδή μια δέσμη κανόνων στους οποίους στηρίζεται η επικοινωνία των συσκευών (συνήθως, αλλά όχι πάντα, υπολογιστών) σε ένα δίκτυο. Οι κανόνες αυτοί καθορίζουν τη μορφή, το χρόνο και τη σειρά μετάδοσης των πληροφοριών στο δίκτυο. Εκτελούν, επίσης, έλεγχο και διόρθωση σφαλμάτων στη διάρκεια μετάδοσης των πληροφοριών. Υπάρχουν διάφορα πρωτόκολλα επικοινωνίας, τα οποία προκαλούν πολλές φορές σύγχυση στους χρήστες. Ευτυχώς σήμερα, παρόλο που δεν υπάρχει κάποιο που να είναι καθιερωμένο πρότυπο, με την εξάπλωση των Windows και του Διαδικτύου, τα πρωτόκολλα που είναι περισσότερο διαδεδομένα είναι το TCP/IP, το NETBEUI και το IPX/SPX.

Στην καθημερινή μας ζωή, πρωτόκολλο είναι ένα σύνολο από συμβάσεις που καθορίζουν το πώς πρέπει να πραγματοποιηθεί κάποια διαδικασία. Στον κόσμο των δικτύων, πρωτόκολλο είναι ένα σύνολο από συμβάσεις που καθορίζουν το πώς ανταλλάσσουν μεταξύ τους δεδομένα οι υπολογιστές του δικτύου. Το πρωτόκολλο είναι αυτό που καθορίζει το πώς διακινούνται τα δεδομένα, το πώς γίνεται ο έλεγχος και ο χειρισμός των λαθών, κλπ. Το Internet δεν είναι ένα απλό δίκτυο, αλλά ένα διαδίκτυο. Χρειάζεται επομένως ένα σύνολο από συμβάσεις που να καθορίζουν το πώς ανταλλάσσουν μεταξύ τους δεδομένα υπολογιστές που μπορεί να είναι διαφορετικού τύπου και να ανήκουν σε διαφορετικά δίκτυα. Ακριβώς αυτό το σύνολο συμβάσεων προσφέρει το TCP/IP. Όλοι οι υπολογιστές που είναι συνδεδεμένοι στα χιλιάδες μικρότερα δίκτυα του Internet τρέχουν το πρωτόκολλο TCP/IP κι έτσι μιλούν μια κοινή γλώσσα που τους επιτρέπει να συνεννοούνται παρά τις διαφορές τους.

## 5.2 Τι είναι Δίκτυο;

Ένα δίκτυο υπολογιστών είναι ένα τηλεπικοινωνιακό σύστημα από αυτόνομους ή μη αυτόνομους διασυνδεδεμένους υπολογιστές. Οι υπολογιστές θεωρούνται διασυνδεδεμένοι όταν είναι σε θέση να ανταλλάξουν πληροφορίες μεταξύ τους και αυτόνομοι όταν δεν είναι δυνατό κάποιος υπολογιστής να ελέγξει τη λειτουργία (π.χ. εκκίνηση ή τερματισμό) κάποιου άλλου. Η επιστημονική μελέτη των δικτύων υπολογιστών γίνεται από τα υπολογιστικά συστήματα, έναν βασικό κλάδο της πληροφορικής. Το θεμελιώδες ηλεκτρονικό υλικό των τηλεπικοινωνιακών συσκευών μελετάται επίσης από την ηλεκτρονική μηχανική.

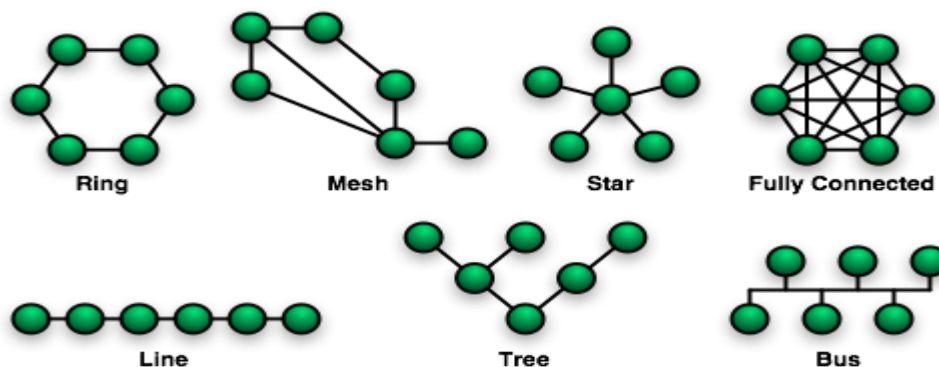
Σε ένα δίκτυο υπολογιστών μπορούν να διασυνδεθούν μεταξύ τους εκτός από τα παραδοσιακά επιτραπέζια PC και άλλου τύπου συσκευές όπως PDAs (Personal Digital Assistans), κινητά τηλέφωνα, τηλεοράσεις, εκτυπωτές, σαρωτές.

## 5.3 Τοπολογίες Δικτύων

Τοπολογία δικτύου ονομάζεται η μορφή της σύνδεσης μεταξύ των κόμβων ενός δικτύου. Οι τοπολογίες είναι είτε φυσικές είτε λογικές. Τα κυριότερα είδη τοπολογιών είναι η γραμμική, η τύπου διαύλου, δακτυλίου, αστέρα, η κατανεμημένη, η πλήρως κατανεμημένη και η τύπου δένδρου. Η επικοινωνία μεταξύ των υπολογιστών και γενικότερα η κοινή χρήση των πόρων τους είναι εφικτή μόνο όταν αυτοί είναι συνδεδεμένοι. Στα περισσότερα σύγχρονα δίκτυα οι υπολογιστές είναι συνδεδεμένοι μεταξύ τους με καλώδια. Ωστόσο, στη διασύνδεση ενός πλήθους υπολογιστών δεν παίζουν ρόλο μόνο τα καλώδια.

Οι διαφορετικοί τύποι καλωδίων, καρτών δικτύου, λειτουργικών συστημάτων καθώς και πολλών άλλων στοιχείων δικτύωσης απαιτούν και διαφορετικούς τρόπους διεύθετησης του δικτύου, ή αλλιώς, απαιτούν διαφορετικές δικτυακές τοπολογίες. Για τη σωστή διεύθετηση μιας τοπολογίας απαιτείται ιδιαίτερη προσοχή. Και αυτό γιατί μια συγκεκριμένη τοπολογία μπορεί να προσδιορίζει τον τύπο της καλωδίωσης που θα πρέπει να χρησιμοποιηθεί, τον τρόπο με τον οποίο τα διάφορα καλώδια θα περνούν μέσα από τα ταβάνια, τοίχους και πατώματα, τον τρόπο επικοινωνίας μεταξύ

των υπολογιστών καθώς και πολλά άλλα λειτουργικά στοιχεία. Όλα αυτά θα πρέπει να λαμβάνονται σοβαρά υπόψη κατά τη διευθέτηση της τοπολογίας, γιατί



προσδιορίζουν σε μεγάλο βαθμό το επίπεδο της απόδοσης ολόκληρου του δικτύου.

Σχήμα 5.3: Διάγραμμα διαφορετικών τοπολογιών δικτύων

#### 5.4 Τι είναι το IPv6 (Internet Protocol version 6);

Το **IPv6** (Internet Protocol version 6) είναι η πιο πρόσφατη αναθεώρηση του πρωτοκόλλου Internet (IP), του βασικού πρωτοκόλλου επικοινωνίας πάνω στο οποίο έχει χτιστεί ολόκληρο το διαδίκτυο. Το Πρωτόκολλο Διαδικτύου (Internet Protocol), αποτελεί το κύριο πρωτόκολλο επικοινωνίας για τη μετάδοση δεδομενογραμμάτων (datagrams), δηλαδή πακέτων δεδομένων, σε ένα διαδίκτυο. Το Πρωτόκολλο IP είναι υπεύθυνο για τη δρομολόγηση των πακέτων δεδομένων ανάμεσα στα διάφορα δίκτυα, ανεξάρτητα από την υποδομή τους, και αποτελεί το κύριο πρωτόκολλο πάνω στο οποίο είναι βασισμένο το Διαδίκτυο.

Το Πρωτόκολλο IP, ανήκει στο Επίπεδο Διαδικτύου, στο Μοντέλο Διαστρωμάτωσης TCP/IP. Καθορίζει τη μορφή των πακέτων που στέλνονται μέσω ενός διαδικτύου, καθώς και τους μηχανισμούς που χρησιμοποιούνται για την προώθηση των πακέτων από έναν υπολογιστή προς έναν τελικό προορισμό μέσω ενός ή περισσότερων δρομολογητών. Για αυτούς τους σκοπούς, το IP, χρησιμοποιεί συγκεκριμένες μεθόδους διευθυνσιοδότησης και δομές για την ενθυλάκωση των πακέτων δεδομένων. Το IPv6 (Internet Protocol version 6) είναι η πιο πρόσφατη αναθεώρηση του πρωτοκόλλου Internet (IP), του βασικού πρωτοκόλλου επικοινωνίας πάνω στο οποίο έχει χτιστεί ολόκληρο το διαδίκτυο. Πρόκειται να αντικαταστήσει το παλιότερο IPv4.

#### 5.5 Τι κάνει το TCP/IP και τι κάνει;

Το πρωτόκολλο IP είναι υπεύθυνο για το πέρασμα του πακέτου από υπολογιστή σε υπολογιστή μέσα από το “σύννεφο” των συνδέσεων. Καθώς το IP δρομολογεί το κάθε πακέτο μέσα στο δίκτυο, προσπαθεί να το παραδώσει, αλλά δεν μπορεί να εγγυηθεί ούτε ότι το πακέτο θα φτάσει στον προορισμό του ούτε ότι τα διάφορα

πακέτα που αποτελούν τα αρχικά δεδομένα θα φτάσουν με τη σειρά με την οποία στάλθηκαν ούτε ότι το περιεχόμενο των πακέτων θα φτάσει αναλλοίωτο.

Το TCP προσφέρει ένα αξιόπιστο πρωτόκολλο πάνω από το IP. Εγγυάται ότι τα πακέτα θα παραδοθούν στον προορισμό τους, ότι θα φτάσουν με τη σειρά με την οποία στάλθηκαν και ότι τα περιεχόμενα των πακέτων θα φτάσουν αναλλοίωτα (δηλ. όπως στάλθηκαν). Το TCP δουλεύει ως εξής: το κάθε πακέτο δεδομένων αριθμείται. Ο υπολογιστής - παραλήπτης και ο υπολογιστής - αποστολέας, αλλά όχι οι ενδιάμεσοι υπολογιστές, παρακολουθούν τους αριθμούς των πακέτων και ανταλλάσσουν μεταξύ τους πληροφορίες. Ο παραλήπτης λαμβάνει το πρώτο πακέτο, το δεύτερο, κλπ. Σε περίπτωση που παρουσιαστεί κάποιο πρόβλημα στο δίκτυο είτε χυθεί κάποιο πακέτο κατά τη διάρκεια της μετάδοσης, το ξαναζητάει και ο αποστολέας είναι υπεύθυνος για την αναμετάδοση του. Ο παραλήπτης ελέγχει επίσης αν το περιεχόμενο των πακέτων φτάνει σωστά. Η μέθοδος αυτή εξασφαλίζει αξιοπιστία και ταχύτητα διότι οι ενδιάμεσοι υπολογιστές δεν εκτελούν ελέγχους.

Ας υποθέσουμε ότι θέλουμε να μεταφέρουμε δεδομένα από έναν υπολογιστή που είναι συνδεδεμένος στο Internet και βρίσκεται π.χ. στην Αμερική, στο MIT, σε έναν άλλον που είναι επίσης συνδεδεμένος στο Internet και βρίσκεται π.χ. στην Ελλάδα, στο Πανεπιστήμιο Θεσσαλίας. Μεταξύ των δύο υπολογιστών παρεμβάλλεται το “σύννεφο” του Internet, δηλ. ένα πλέγμα από συνδέσεις και ενδιάμεσους υπολογιστές.

Τώρα λοιπόν που γνωρίσαμε το TCP/IP μπορούμε να δώσουμε έναν πιο “επίσημο” ορισμό του Internet: ένα δίκτυο αποτελούμενο από δίκτυα υπολογιστών που επικοινωνούν χρησιμοποιώντας το πρωτόκολλο TCP/IP.

## **5.6 Ασύρματες Ενσωματωμένες Συσκευές (Wireless Embedded Devices)**

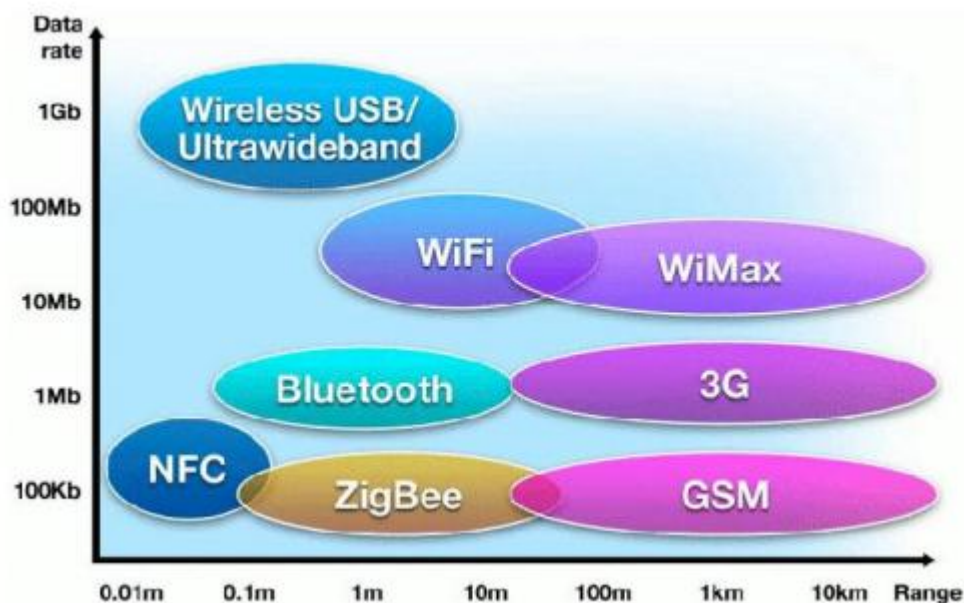
Ασύρματες ενσωματωμένες συσκευές αποτελούν μια από τις ταχύτερες αναπτυσσόμενες τεχνολογίες πληροφορικής και σήμερα. Από το συνολικό ποσό των μικροεπεξεργαστών που πωλούνται σήμερα, το 98% χρησιμοποιούνται σε ενσωματωμένα συστήματα. Ασύρματο ενσωματωμένα συστήματα κυμαίνονται από τα κλειδιά ασύρματο αυτοκίνητο για συστήματα ασφαλείας για τα εμπορευματοκιβώτια φορτηγό. Ασύρματες ενσωματωμένες συσκευές ανταποκρίνονται στις προκλήσεις από την άποψη του κόστους, φυσικού μεγέθους, και η κατανάλωση ενέργειας, η οποία έχει συνέπειες για τα λειτουργικά συστήματα που τρέχουν στις συσκευές.

## **5.7 Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks)**

Ένα ασύρματο δίκτυο αισθητήρων (ΑΔΑ / Wireless Sensor Network - WSN ) αποτελείται από διασκορπισμένους αυτόνομους αισθητήρες για την παρακολούθηση φυσικών ή περιβαλλοντολογικών συνθηκών, όπως η θερμοκρασία, ο ήχος, η ατμοσφαιρική πίεση κτλ και μέσω συνεργασίας να μεταφέρει τα δεδομένα μέσω του δικτύου σε μια συγκεκριμένη τοποθεσία. Τα πιο μοντέρνα δίκτυα είναι ικανά και να

δίνουν αλλά και να δέχονται πληροφορίες πράγμα που τους επιτρέπει να ελέγχουν την δραστηριότητα των αισθητήρων.

Το κίνητρο για την ανάπτυξη των ασύρματων δικτύων με αισθητήρες ήταν οι στρατιωτικές εφαρμογές όπως η παρακολούθηση των πεδίων μάχης. Σήμερα τέτοια δίκτυα χρησιμοποιούνται σε πολλές καταναλωτικές και βιομηχανικές εφαρμογές, η παρακολούθηση και ο έλεγχος της βιομηχανικής παραγωγής, την παρακολούθηση των μηχανημάτων υγείας και πολλά άλλα.



Σχήμα 5.7: Μερικές ασύρματες τεχνολογίες, με απόσταση κάλυψης και ρυθμό μεταφοράς

Το ασύρματο δίκτυο αισθητήρων αποτελείται από κόμβους - από μερικές σε αρκετές εκατοντάδες η ακόμα και χιλιάδες, όπου κάθε κόμβος συνδέεται σε έναν (η κάποιες φορές σε αρκετούς) αισθητήρες. Κάθε τέτοιος κόμβος του δικτύου αισθητήρων έχει χαρακτηριστικά μερικά κομμάτια : ένα ραδιοπομποδέκτη με μια εσωτερική κεραία η μια σύνδεση με μια εξωτερική κεραία, έναν μικροελεγκτή, ένα ηλεκτρονικό κύκλωμα για την διασύνδεση με τους αισθητήρες και μια πηγή ενέργειας, συνήθως μια μπαταρία η μια ενσωματωμένη μορφή συγκομιδής ενέργειας.

Ένας αισθητήριος κόμβος μπορεί να ποικίλει σε μέγεθος από εκείνο ενός κουτιού παπουτσιών μέχρι το μέγεθος ενός κόκκου σκόνης, αν και λειτουργικοί «κόκκοι» πραγματικά μικροσκοπικών διαστάσεων δεν έχουν ακόμα δημιουργηθεί. Το κόστος των αισθητήριων κόμβων ποικίλει, ξεκινώντας από μερικά και φτάνοντας σε εκατοντάδες δολάρια, αναλόγως την πολυπλοκότητα των μεμονωμένων αισθητήριων κόμβων. Οι περιορισμοί σε μέγεθος και κόστος έχουν ως αποτέλεσμα αντίστοιχους περιορισμούς σε πόρους όπως ενέργεια, μνήμη, υπολογιστική ταχύτητα και στο εύρος ζώνης των επικοινωνιών. Η τοπολογία των αισθητήρων μπορεί να διαφέρει από ένα



δίκτυο τοπολογίας αστέρος σε ένα αναπτυγμένο ασύρματο δίκτυο πλέγματος multi-hop. Η πολλαπλασιαστική τεχνική μεταξύ των λυκίσκων του δικτύου μπορεί να είναι η δρομολόγηση ή ο καταγισμός διακίνησης.

## ΚΕΦΑΛΑΙΟ 6

### 6. ΣΧΕΔΙΑΣΜΟΣ ΤΩΝ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ

Η διαδικασία σχεδιασμού ενσωματωμένων συστημάτων πρέπει να λάβει υπόψη της και με λεπτομέρεια τον σχεδιασμό και την απόδοση του υλικού του συστήματος.

- Μέρος της διαδικασίας σχεδιασμού δύναται να εμπλέκει τη λήψη απόφασης για το ποιες δυνατότητες του συστήματος να υλοποιηθούν στο λογισμικό και ποιες στο υλικό.
- Το υλικό αποδίδει καλύτερα αλλά χρειάζεται ενδεχομένως πιο μακρόχρονη ανάπτυξη και αφήνει μικρότερα περιθώρια για αλλαγές.
- Αποφάσεις χαμηλού επιπέδου για το υλικό, το λογισμικό υποστήριξης και τη χρονική συμπεριφορά του συστήματος θα πρέπει να εξετάζονται από την αρχή της διαδικασίας.
- Αυτοί οι παράγοντες περιορίζουν την ευελιξία των σχεδιαστών των συστημάτων και μπορεί να επιβάλλουν απαιτήσεις για επιπλέον λειτουργικότητα του λογισμικού, όπως διαχείριση μπαταριών και τροφοδοσίας.



Σχήμα 6: Σχεδιασμός Ενσωματωμένων Συστημάτων

Στην Ενότητα αυτή παρουσιάσαμε την ιδέα ενός ενσωματωμένου συστήματος, ενός ψηφιακού συστήματος στο οποίο ένας ή περισσότεροι υπολογιστές χρησιμοποιούνται σαν συστατικά στοιχεία. Κάθε ενσωματωμένος υπολογιστής αποτελείται από έναν πυρήνα επεξεργαστή (processor core), μνήμη και διασυνδέσεις με άλλα μέρη του συστήματος. Αφού οι υπολογιστές πρέπει να προγραμματιστούν για να υλοποιήσουν μέρος της λειτουργικότητας του συστήματος, πρέπει να εμπλουτίσουμε τη σχεδιαστική μας μεθοδολογία ώστε να συμπεριλάβει τη σχεδίαση ενσωματωμένου λογισμικού (embedded software).

Θυμηθείτε ότι οι αρχικές εισοδοι στη μεθοδολογία σχεδίασης είναι οι λειτουργικές απαιτήσεις και οι περιορισμοί από το σύστημα. Σαν μέρος των σκέψεων

μας για την αρχιτεκτονική σχεδίαση, μπορούμε να επιλέξουμε ποιες πτυχές της λειτουργικότητας μπορούν να υλοποιηθούν με ενσωματωμένο λογισμικό στον πυρήνα του επεξεργαστή, και ποια μέρη μπορούν να υλοποιηθούν σαν ψηφιακά υποκυκλώματα, δηλαδή στο υλικό. Η ταυτόχρονη σχεδίαση του υλικού και του λογισμικού ενός συστήματος ονομάζεται συσχεδίαση υλικού/λογισμικού (hardware/software codesign). Η απόφαση για το ποια τμήματα θα μπουν στο υλικό και ποια στο λογισμικό ονομάζεται διαμέριση (partitioning).

Υπάρχουν πολυάριθμοι συμβιβασμοί (trade-offs) που πρέπει να ληφθούν υπόψη. Η λειτουργικότητα που εμπεριέχει τον έλεγχο πολλών συνθηκών και την πραγματοποίηση εναλλακτικών ενεργειών μπορεί να είναι δύσκολο να υλοποιηθεί στο υλικό, αλλά είναι σχετικά απλή στο λογισμικό. Από την άλλη, η λειτουργικότητα που εμπεριέχει την εκτέλεση γρήγορων υπολογισμών σε μεγάλες ποσότητες δεδομένων ή σε δεδομένα που καταφθάνουν σε υψηλό ρυθμό μπορεί να απαιτεί έναν πυρήνα επεξεργαστή πολύ υψηλών επιδόσεων (άρα ακριβό και ενεργοβόρο), και συνεπώς ίσως είναι καλύτερο να υλοποιηθεί από προσαρμοσμένο (customized) υλικό.

Ένα πρόσθετο ζήτημα είναι ότι το ενσωματωμένο λογισμικό μπορεί να αποθηκεύεται σε κυκλώματα μνήμης που μπορεί να επαναπρογραμματίζονται αφού το σύστημα κατασκευαστεί ή εγκατασταθεί. Συνεπώς, το λογισμικό μπορεί να αναβαθμιστεί για να διορθώσει σχεδιαστικά λάθη ή για να προσθέσει λειτουργικότητα χωρίς να αναθεωρηθεί η σχεδίαση του υλικού ή να αντικατασταθούν εγκατεστημένα συστήματα. Για τις πτυχές του ενσωματωμένου λογισμικού που εξαρτώνται από το υλικό, μπορούν να χρησιμοποιηθούν τα αφηρημένα μοντέλα συμπεριφοράς από την εργασία σχεδίασης του υλικού ώστε να γίνει η επαλήθευση της σχεδίασης του λογισμικού.

Αυτό μπορεί να γίνει με χρήση ενός προσομοιωτή συνόλου εντολών (instruction-set simulator) για τον πυρήνα επεξεργαστή σε σύζευξη με έναν προσομοιωτή για το μοντέλο του υλικού. Μια παρόμοια προσέγγιση μπορεί να χρησιμοποιηθεί για την επαλήθευση τμημάτων του υλικού που επικοινωνούν απευθείας με τον πυρήνα του επεξεργαστή. Προγράμματα δοκιμής μπορούν να εκτελούνται στον προσομοιωτή του επεξεργαστή σε συνδυασμό με τον προσομοιωτή του υλικού.

Τα πλεονεκτήματα της ταυτόχρονης ανάπτυξης του υλικού και του λογισμικού περιλαμβάνουν την αποφυγή του επιπλέον χρόνου που απαιτείται για την σειριακή ανάπτυξη, και την έγκαιρη ανίχνευση σφαλμάτων που σχετίζονται με τη συνεργασία του λογισμικού και του υλικού.

## **6.1 ΜΕΘΟΔΟΛΟΓΙΑ ΣΧΕΔΙΑΣΗΣ**

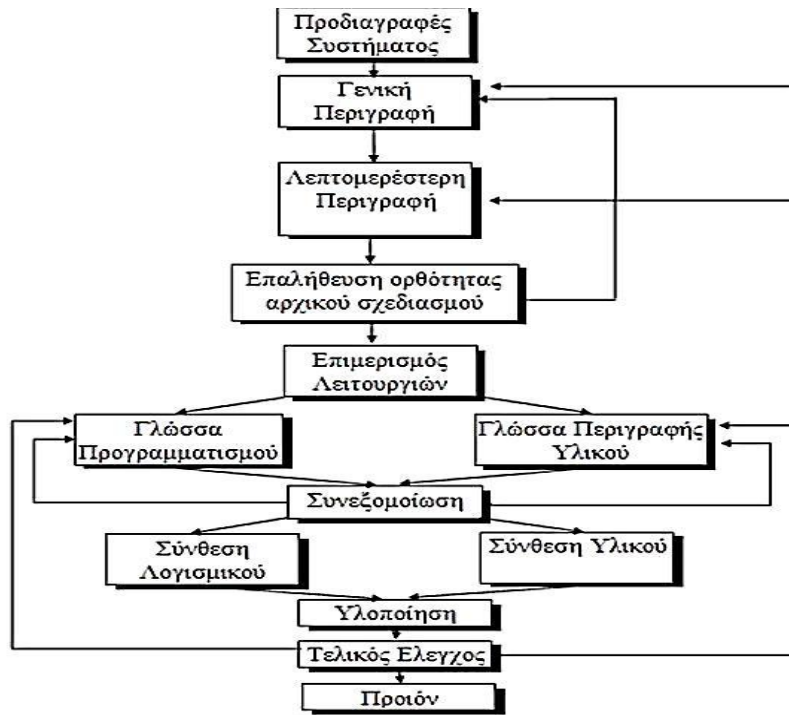
Η σχεδίαση ενός ψηφιακού συστήματος κάποιας πολυπλοκότητας είναι ένα μεγάλο έργο, που απαιτεί μια συστηματική προσέγγιση. Αυτό είναι ιδιαίτερα σημαντικό όταν πολλοί άνθρωποι συνεργάζονται σε μία σχεδίαση, όπως γίνεται συνήθως. Ανάλογα με την πολυπλοκότητα του προϊόντος, οι σχεδιαστικές ομάδες μπορεί να κυμαίνονται σε μέγεθος από μια μικρή ομάδα μηχανικών για ένα σχετικά

απλό προϊόν, μέχρι πολλές εκατοντάδες ανθρώπων για ένα πολύπλοκο ολοκληρωμένο κύκλωμα ή συσκευασμένο σύστημα. Χρησιμοποιούμε τον όρο μεθοδολογία σχεδίασης (design methodology) για να αναφερθούμε στη συστηματική διαδικασία της σχεδίασης, επαλήθευσης και προετοιμασίας της κατασκευής ενός προϊόντος.

Μια μεθοδολογία σχεδίασης καθορίζει τις εργασίες που αναλαμβάνουμε, την πληροφορία που απαιτείται και παράγεται σε κάθε εργασία, τις σχέσεις μεταξύ των εργασιών συ-μπεριλαμβανομένων των εξαρτήσεων και του καθορισμού της ακολουθίας τους, και τα εργαλεία CAD που χρησιμοποιούνται. Μια ώριμη μεθοδολογία σχεδίασης θα είναι επίσης ανακλαστική, δηλαδή θα προσδιορίζει τις μετρήσεις που θα γίνουν κατά τη σχεδιαστική διαδικασία, όπως η τήρηση του χρονοδιαγράμματος και του προϋπολογισμού, και τον αριθμό των σχεδιαστικών λαθών που ανιχνεύθηκαν ή διέφυγαν. Συσσωρευμένα δεδομένα από προηγούμενα έργα μπορούν να χρησιμοποιηθούν για τη βελτίωση της μεθοδολογίας σχεδίασης σε επόμενα έργα.

Το πλεονέκτημα μιας καλής μεθοδολογίας είναι ότι κάνει τη σχεδιαστική διαδικασία πιο αξιόπιστη και προβλέψιμη, και κατά συνέπεια μειώνει τους κινδύνους και το κόστος. Ακόμη κι ένα μικρό σχεδιαστικό έργο επωφελείται από μια σχεδιαστική μεθοδολογία, παρόλο που αυτό ίσως γίνεται σε μικρότερη κλίμακα. Θα αρχίσουμε δίνοντας ένα περίγραμμα μιας σχετικά απλής μεθοδολογίας, αφού ακόμη βρισκόμαστε στα πρώτα στάδια εκμάθησης της ψηφιακής σχεδίασης.

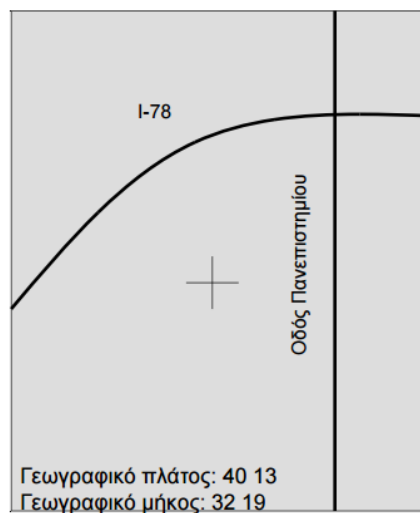
Το σημείο εκκίνησης είναι μια συλλογή απαιτήσεων και περιορισμών. Αυτές συνήθως δημιουργούνται εκτός της σχεδιαστικής ομάδας, για παράδειγμα από το διαφημιστικό τμήμα μιας εταιρείας ή από ένα πελάτη από τον οποίο γίνεται η ανάθεση της ανάπτυξης του προϊόντος. Συνήθως, περιλαμβάνουν λειτουργικές απαιτήσεις (τι πρόκειται να κάνει το προϊόν), απαιτήσεις απόδοσης (πόσο γρήγορα πρόκειται να το κάνει), και απαιτήσεις κατανάλωσης ισχύος, κόστους και συσκευασίας. Η σχεδιαστική μεθοδολογία καθορίζει τρεις εργασίες – τη σχεδίαση, τη σύνθεση και τη φυσική υλοποίηση – κάθε μία από τις οποίες ακολουθείται από μια εργασία επαλήθευσης. Οι εργασίες της σχεδίασης και της λειτουργικής επαλήθευσης έχουν μπει σε πλαίσιο για να δείξουμε ότι σχετίζονται λίγο περισσότερο από ότι φαίνεται στο διάγραμμα.



Σχήμα 6.1: Γενική Μεθοδολογία Σχεδιασμού Ενσωματωμένων Συστημάτων

### 6.1.1 Παράδειγμα σχεδίασης ενσωματωμένου συστήματος

Παράδειγμα: Απαιτήσεις ενός κινούμενου χάρτη GPS

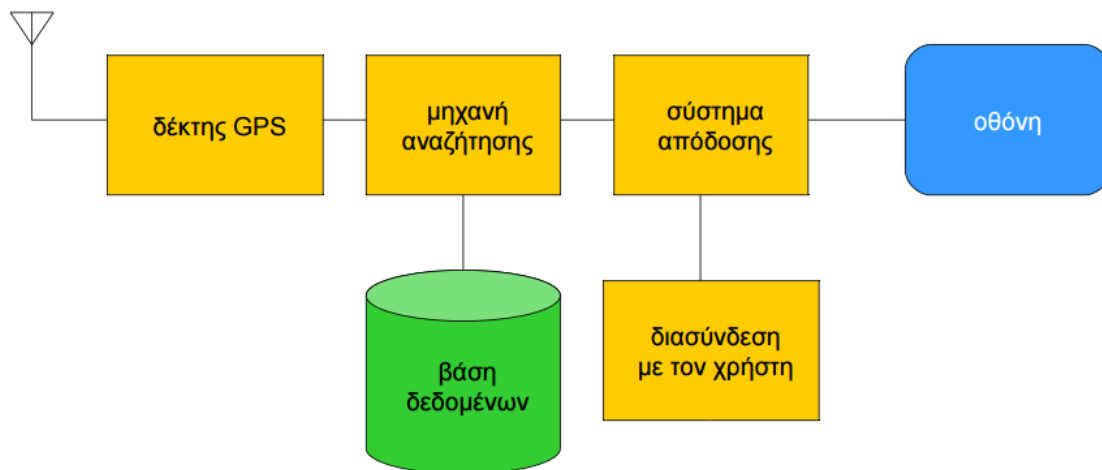


Σχήμα 6.1.1 (Εικόνα 1): Ο κινούμενος χάρτης λαμβάνει τη θέση του από το GPS, ο χάρτης σχεδιάζεται από την τοπική βάση δεδομένων.

Οι Ανάγκες κινούμενου χάρτη GPS:

- Λειτουργικότητα: Για χρήση στην αυτοκινητοβιομηχανία. Εμφάνιση μεγάλων οδικών αρτηριών και σημείων ενδιαφέροντος.
- Διασύνδεση με το χρήστη: Οθόνη τουλάχιστον 400x600 εικονοστοιχείων. Το πολύ τρία κουμπιά. Μενού pop-up.
- Απόδοση: Ο χάρτης θα πρέπει να κυλιέται ομαλά. Όχι περισσότερο από 1 δευτερόλεπτο για ενεργοποίηση. Εντοπισμός μέσα σε 15 δευτερόλεπτα.
- Κόστος: \$120 τιμή λιανικής πώλησης = προσεγγιστικά \$30 κόστους πωληθέντων αγαθών.
- Φυσικό μέγεθος/βάρος: Πρέπει να χωράει στην παλάμη του χεριού.
- Κατανάλωση ισχύος: Πρέπει να μπορεί να λειτουργεί τουλάχιστον για 8 ώρες με τέσσερις μπαταρίες AA.

Διαγράμμα 6.1.1 (Εικόνα 2): Μπλοκ διάγραμμα για τον κινούμενο χάρτη GPS



## 6.2 Μετρικές σχεδίασης ενσωματωμένων συστημάτων

Το ζητούμενο είναι να σχεδιάσουμε ένα σύστημα σύμφωνα με τις προδιαγραφές. Το σύστημα πρέπει να υλοποιεί την επιθυμητή λειτουργικότητα. Όμως, πρέπει να βελτιστοποιεί και κάποιες σχεδιαστικές μετρικές. Οι σχεδιαστικές μετρικές είναι μετρήσιμα χαρακτηριστικά του συστήματος. Αν δεν βελτιστοποιηθούν, το προϊόν καταδικάζεται σε αποτυχία:

- NRE cost (Non-Recurrent Engineering cost): το κόστος της σχεδίασης και ανάπτυξης του συστήματος
- Κόστος Μονάδας (Unit cost): το κόστος της κατασκευής μιας μονάδας μείον το κόστος NRE
- Μέγεθος (Size): το μέγεθος κάθε μονάδας

- Απόδοση (performance): ο χρόνος εκτέλεσης της μονάδας έργου (throughput)
- Ισχύς: Το ποσό της ισχύος που δαπανά το σύστημα
- Ευελιξία: πόσο εύκολα το σύστημα μπορεί να αλλάξει λειτουργικότητα
- Χρόνος πρωτοτυποποίησης (time to prototype)
- Χρόνος πρόσβασης στην αγορά (time-to-market)

Οι παραπάνω μετρικές συχνά είναι αντικρουόμενες. Έτσι, η αύξηση της απόδοσης, συχνά αυξάνει το μέγεθος του συστήματος, ενώ η μείωση της δαπανώμενης ισχύος αυξάνει το κόστος.

### 6.3 Φάση Σχεδίασης

Επειδή στην περίπτωση των ενσωματωμένων συστημάτων η ανάπτυξη του κυκλωματικού (hardware) και του λογισμικού (software) πρέπει να προχωρούν παράλληλα, στη φάση αυτή πρέπει να σχεδιαστούν τα κυκλώματα τα οποία θα γεμίσουν τις επιμέρους βαθμίδες των διαγραμμάτων βαθμίδων, που έγιναν στη φάση του ορισμού, και να κατασκευαστούν τα διαγράμματα ροής του λογισμικού.

Στη φάση αυτή δεν κατασκευάζεται κανένα πρότυπο και δε γράφεται κανένα πρόγραμμα. Απλώς γίνεται διερεύνηση των επιμέρους βαθμίδων των διαγραμμάτων βαθμίδων και σπάζουν σε μικρότερες βαθμίδες τέτοιες που να αποτελούνται από όσο το δυνατό μικρότερα επιμέρους κυκλώματα, τα οποία θα μπορέσουν να υλοποιηθούν και να ελεγχθούν για καλή λειτουργία.

### 6.4 Ψηφιακή Σχεδίαση-Ενσωματωμένα Συστήματα με τη γλώσσα VHDL

Σε αυτή την ενότητα θα δώσουμε έμφαση στη σχεδίαση των κυκλωμάτων και θα εξηγήσουμε πως υλοποιούνται τα κυκλώματα σε πραγματικά chip.

Η VHDL χρησιμοποιείται για να επιδείξει τον τρόπο ορισμού των βασικών δομικών βαθμίδων και των μεγαλύτερων συστημάτων σε μια γλώσσα περιγραφής υλικού, παράγοντας σχεδιάσεις που μπορούν να υλοποιηθούν με σύγχρονα εργαλεία σχεδίασης CAD.

Είναι μια περίπλοκη γλώσσα, για την οποία ορισμένοι εκπαιδευτές έχουν την αίσθηση ότι είναι πολύ δύσκολο να την καταλάβουν άτομα στο ξεκίνημα των σπουδών τους.

Οι σχεδιαστές ψηφιακών ολοκληρωμένων κυκλωμάτων ενδέχεται να έχουν ως αντικείμενο τη σχεδίαση λογικών κυκλωμάτων που πρέπει να τοποθετηθούν σε ένα ενιαίο chip ή, πιο πιθανό τη σχεδίαση κυκλωμάτων που περιλαμβάνουν περισσότερα από ένα ολοκληρωμένα κυκλώματα και τοποθετούνται σε μια πλακέτα τυπωμένου κυκλώματος. Συχνά μερικά από τα αναγκαία λογικά κυκλώματα μπορούν να υλοποιηθούν χρησιμοποιώντας κυκλώματα που ήδη υπάρχουν στην αγορά. Η δυνατότητα αυτή απλοποιεί τη σχεδιαστική δυσκολία και μειώνει το χρόνο που απαιτείται για την ανάπτυξη του τελικού προϊόντος. Προτού συζητήσουμε τη σχεδιαστική διαδικασία με περισσότερη λεπτομέρεια, θα πρέπει να περιγράψουμε τα

διάφορα είδη ολοκληρωμένων κυκλωμάτων. Τα ολοκληρωμένα κυκλώματα εκτείνονται από πολύ απλά κυκλώματα χαμηλής λειτουργικότητας έως εξαιρετικά περίπλοκα ολοκληρωμένα κυκλώματα διασύνδεσης προς συσκευές εισόδου και εξόδου. Τέτοια ολοκληρωμένα κυκλώματα διατίθενται στην αγορά από διάφορες κατασκευάστριες εταιρείες. Στα περισσότερα ψηφιακά προϊόντα είναι επίσης αναγκαία η σχεδίαση και κατασκευή κάποιων λογικών κυκλωμάτων από την αρχή.

Για την υλοποίηση τέτοιων κυκλωμάτων μπορούν να χρησιμοποιηθούν τρεις μορφές ολοκληρωμένων κυκλωμάτων: τα τυπικά ολοκληρωμένα κυκλώματα, οι διατάξεις προγραμματιζόμενης λογικής και τα ειδικά ολοκληρωμένα κυκλώματα.

- **Τυπικά Ολοκληρωμένα Κυκλώματα**

Σήμερα διατίθενται αναρίθμητα ολοκληρωμένα κυκλώματα, τα οποία υλοποιούν πολλά ευρέως χρησιμοποιούμενα λογικά κυκλώματα. Θα τα αναφέρουμε ως τυπικά ολοκληρωμένα κυκλώματα (standard chips), επειδή εν γένει ευρίσκονται σε συμφωνία με θεσπισμένα πρότυπα, όσον αφορά τη λειτουργία και τη φυσική τους οργάνωση. Το κάθε πρότυπο ολοκληρωμένο κύκλωμα περιέχει ένα μικρό κύκλωμα (συνήθως μικρότερο από 100 τρανζίστορες) και εκτελεί μια απλή εργασία. Για την κατασκευή κάποιου λογικού κυκλώματος, ο σχεδιαστής επιλέγει τα ολοκληρωμένα κυκλώματα που εκτελούν τις λειτουργίες που απαιτούνται και στη συνέχεια καθορίζει τον τρόπο με τον οποίο αυτά θα συνδεθούν μεταξύ τους για να σχηματίσουν ένα μεγαλύτερο λογικό κύκλωμα.

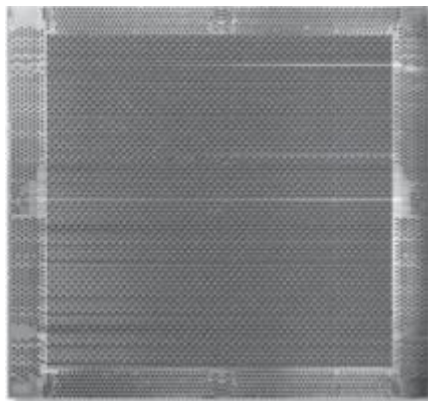
Τα τυπικά ολοκληρωμένα κυκλώματα ήταν δημοφιλή για τη δημιουργία λογικών κυκλωμάτων έως τις αρχές της δεκαετίας του 1980. Εντούτοις, καθώς εξελισσόταν η τεχνολογία κατασκευής ολοκληρωμένων κυκλωμάτων, κατέστη ανώφελη η διάθεση πολύτιμου χώρου στις πλακέτες (PCB) για την τοποθέτηση ολοκληρωμένων κυκλωμάτων με περιορισμένες δυνατότητες. Ένα άλλο μειονέκτημα των τυπικών ολοκληρωμένων κυκλωμάτων είναι ότι η λειτουργία του καθενός είναι συγκεκριμένη και δε μπορεί να μεταβληθεί.

- **Διατάξεις Προγραμματιζόμενης Λογικής**

Σε αντίθεση με τα τυπικά ολοκληρωμένα κυκλώματα που εκτελούν συγκεκριμένες λειτουργίες, είναι δυνατή η ανάπτυξη ολοκληρωμένων κυκλωμάτων που περιέχουν κυκλώματα τα οποία μπορούν να οργανωθούν από το χρήστη, ώστε να υλοποιούν διάφορα λογικά κυκλώματα. Τα κυκλώματα αυτά έχουν μία πολύ γενική δομή και περιλαμβάνουν ένα σύνολο προγραμματιζόμενων λογικών διακοπών (programmable logic switches), οι οποίοι επιτρέπουν τα εσωτερικά κυκλώματα του ολοκληρωμένου κυκλώματος να οργανώνονται με διάφορους τρόπους. Ο σχεδιαστής μπορεί να υλοποιήσει οποιοσδήποτε λειτουργίες επιθυμεί για την εκτέλεση μιας συγκεκριμένης εφαρμογής, επιλέγοντας την κατάλληλη οργάνωση των διακοπών. Οι διακόπτες προγραμματίζονται από τον τελικό χρήστη και όχι από την κατασκευάστρια εταιρεία, κατά το διάστημα κατασκευής του ολοκληρωμένου κυκλώματος. Τα ολοκληρωμένα αυτά κυκλώματα ονομάζονται διατάξεις προγραμματιζόμενης λογικής. Οι



περισσότερες μορφές PLDs μπορούν να προγραμματιστούν πολλές φορές. Αυτή η δυνατότητα είναι σημαντική, καθώς ένας σχεδιαστής μπορεί να προγραμματίσει ένα PLD ώστε να εκτελεί κάποια λειτουργία, αλλά αργότερα, όταν το τελικό ψηφιακό σύστημα θα ελέγχεται, μπορεί να επιφέρει διορθώσεις προγραμματίζοντας εκ νέου το PLD. Ο επαναπρογραμματισμός ενδέχεται να είναι απαραίτητος, για παράδειγμα, εάν κάποια λειτουργία δεν είναι αυτή που ακριβώς ζητείται και εάν απαιτούνται νέες λειτουργίες, οι οποίες δεν είχαν προβλεφθεί κατά την αρχική σχεδίαση. Τα PLDs διατίθενται σε διάφορα μεγέθη. Μπορούν να χρησιμοποιηθούν για την υλοποίηση πολύ μεγαλύτερων λογικών κυκλωμάτων, σε σύγκριση με αυτά που μπορεί να υλοποιήσει ένα τυπικό ολοκληρωμένο κύκλωμα. Λόγω του μεγέθους τους και του γεγονότος ότι μπορούν να ρυθμιστούν, ώστε να πληρούν ακριβώς τις προδιαγραφές της εκάστοτε ειδικής εφαρμογής, τα PLDs χρησιμοποιούνται ευρέως σήμερα. Μία από τις πλέον εξελιγμένες εκδοχές των PLDs είναι οι διατάξεις πυλών προγραμματιζόμενου πεδίου (field-programmable gate arrays, FPGAs). Διατίθεται FPGAs που περιέχουν αρκετές εκατοντάδες εκατομμύρια τρανζίστορες. Η φωτογραφία ενός FPGA που περιέχει 10 εκατομμύρια τρανζίστορες εικονίζεται στη σχήμα. Τα ολοκληρωμένο κύκλωμα αποτελείται από ένα μεγάλο αριθμό μικρών στοιχείων λογικών κυκλωμάτων που μπορούν να συνδεθούν μεταξύ τους με τη βοήθεια των προγραμματιζόμενων διακοπών. Τα στοιχεία των λογικών κυκλωμάτων είναι οργανωμένα σε κανονική δισδιάστατη δομή.



Σχήμα 6.4 : Ένα ολοκληρωμένο κύκλωμα σειράς πυλών προγραμματιζόμενου πεδίου.

- **Ειδικά Ολοκληρωμένα Κυκλώματα**

Τα PLDs διατίθενται στο εμπόριο και κατασκευάζονται από διάφορες εταιρείες. Επειδή είναι προγραμματιζόμενα, μπορούν να χρησιμοποιηθούν για την υλοποίηση των περισσότερων λογικών κυκλωμάτων που υπάρχουν στα ψηφιακά συστήματα. Εντούτοις, τα PLDs έχουν το μειονέκτημα ότι οι προγραμματιζόμενοι διακόπτες που περιέχουν καταλαμβάνουν πολύτιμο χώρο επάνω στο ολοκληρωμένο κύκλωμα και περιορίζουν την ταχύτητα λειτουργίας των κυκλωμάτων που υλοποιούνται με αυτά. Επομένως σε μερικές περιπτώσεις τα PLDs ενδέχεται να μην πληρούν τις επιθυμητές επιδόσεις ή το επιτρεπτό κόστος. Στις περιπτώσεις αυτές είναι προτιμότερη η σχεδίαση ενός ολοκληρωμένου κυκλώματος από το μηδέν. Έτσι σχεδιάζεται καταρχήν το λογικό κύκλωμα που πρέπει να τοποθετηθεί μέσα σε ολοκληρωμένο

κύκλωμα και στη συνέχεια επιλέγεται κάποια κατάλληλη τεχνολογία για την υλοποίηση του ολοκληρωμένου κυκλώματος. Τέλος, το ολοκληρωμένο κύκλωμα υλοποιείται από μια εταιρία που διαθέτει τα κατάλληλα μηχανήματα. Η προσέγγιση αυτή ονομάζεται ειδική σχεδίαση (custom design) και αυτά τα ολοκληρωμένα κυκλώματα ονομάζονται ειδικά ολοκληρωμένα κυκλώματα (custom chips). Τέτοια κυκλώματα που προορίζονται για χρήση σε συγκεκριμένες εφαρμογές ονομάζονται ολοκληρωμένα κυκλώματα ειδικής εφαρμογής (application-oriented integrated circuits, ASICs). Το κύριο πλεονέκτημα των ειδικών ολοκληρωμένων κυκλωμάτων είναι ότι η σχεδίασή τους μπορεί να βελτιστοποιηθεί προς την κατεύθυνση κάποιας λειτουργίας και αυτό οδηγεί συνήθως σε καλύτερες επιδόσεις. Είναι δυνατόν σε ένα ειδικό ολοκληρωμένο κύκλωμα να τοποθετηθούν περισσότερα κυκλώματα από όσα περιέχει κάποια άλλη μορφή ολοκληρωμένου κυκλώματος. Το κόστος παραγωγής τέτοιων ολοκληρωμένων κυκλωμάτων είναι υψηλό, αλλά, εάν το προϊόν πρόκειται να πωληθεί σε μεγάλες ποσότητες, το κόστος ανά τεμάχιο μπορεί να μειωθεί πολύ και να γίνει μικρότερο από το κόστος ενός εμπορικού τεμαχίου που εκτελεί την ίδια λειτουργία. Επιπρόσθετα, εάν μπορεί να χρησιμοποιηθεί για την εκτέλεση μιας λειτουργίας ένα ολοκληρωμένο κύκλωμα αντί για δύο ή περισσότερα, εξοικονομείται χώρος στην πλακέτα (PCB) του τελικού προϊόντος και έτσι το κόστος μειώνεται περαιτέρω. Ένα μειονέκτημα της προσέγγισης αυτής είναι ότι η κατασκευή ενός τέτοιου ολοκληρωμένου κυκλώματος διαρκεί μεγάλο χρονικό διάστημα. Αντίθετα, εάν χρησιμοποιηθεί ένα PLD, ο προγραμματισμός των ολοκληρωμένων κυκλωμάτων γίνεται από τον τελικό χρήστη και έτσι δεν υπαισέρχονται καθυστερήσεις λόγω της διαδικασίας κατασκευής.

## 6.5 Η Διαδικασία Σχεδίασης

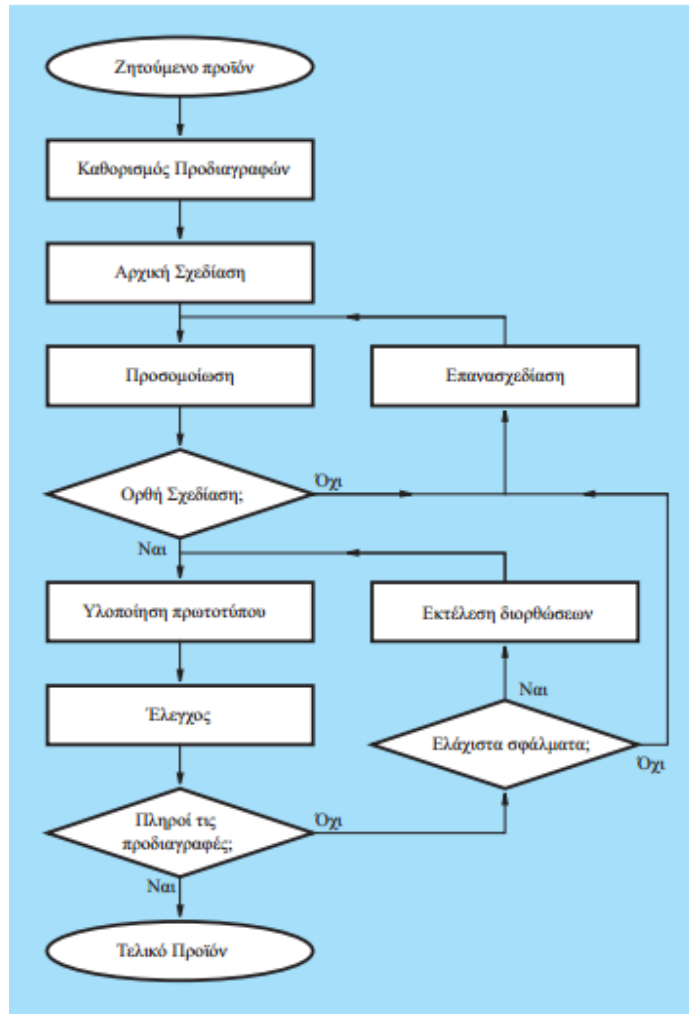
Η διαδικασία εργαλείων που στηρίζονται σε υπολογιστές έχει επηρεάσει σημαντικά τη σχεδιαστική διαδικασία σε διάφορα περιβάλλοντα σχεδίασης. Για παράδειγμα, η σχεδίαση ενός αυτοκινήτου μοιάζει με τη γενική διαδικασία σχεδίασης ενός φούρνου ή ενός υπολογιστή. Πρέπει να γίνουν συγκεκριμένα βήματα ανάπτυξης κατά τη διαδικασία αυτή εάν πρόκειται το τελικό προϊόν να πληροί κάποιες προδιαγραφές.

Το διάγραμμα ροής του σχήματος παρουσιάζει μια τυπική διαδικασία ανάπτυξης. Θεωρούμε ότι η διαδικασία στοχεύει στην ανάπτυξη ενός προϊόντος που πληροί κάποιες προδιαγραφές. Οι πιο προφανείς απαιτήσεις είναι ότι το προϊόν θα πρέπει να λειτουργεί σωστά και ότι θα πρέπει να έχει κάποιο συγκεκριμένο βαθμό απόδοσης, ενώ παράλληλα το κόστος του δε θα πρέπει να υπερβαίνει κάποια τιμή.

Η διαδικασία ξεκινά με τον καθορισμό των προδιαγραφών του προϊόντος. Καθορίζονται τα σημαντικά χαρακτηριστικά του προϊόντος και επιλέγεται μία αποδεκτή μέθοδος αξιολόγησης των χαρακτηριστικών αυτών στο τελικό προϊόν. Οι προδιαγραφές πρέπει να είναι ιδιαίτερα αυστηρές ώστε να διασφαλιστεί ότι το τελικό προϊόν θα πληροί τις γενικές προδιαγραφές, αλλά όχι απαραίτητα περιοριστικές (δηλαδή δε θα πρέπει να αποτρέπουν επιλογές που ενδέχεται να οδηγήσουν σε μελλοντικά οφέλη).

Με βάση το πλήρες σύνολο προδιαγραφών είναι αναγκαίο να καθοριστεί η γενική δομή και η αρχική μορφή που θα έχει το προϊόν. Αυτό το βήμα είναι δύσκολο να αυτοματοποιηθεί. Εν γένει εκτελείται από κάποιον άνθρωπο-σχεδιαστή, επειδή δεν υπάρχει συγκεκριμένη στρατηγική ανάπτυξης της γενικής δομής ενός αντικειμένου, καθώς για να γίνει αυτό απαιτείται σημαντική σχεδιαστική εμπειρία και διαίσθηση.

Αφότου αποφασιστεί η γενική δομή, χρησιμοποιούνται εργαλεία σχεδίασης CAD για την επεξεργασία των λεπτομερειών. Σήμερα διατίθενται πολλές μορφές προγραμμάτων CAD, οι οποίες εκτείνονται από αυτές που βοηθούν τη σχεδίαση επιμέρους τμημάτων έως αυτές που επιτρέπουν την παράσταση ολόκληρης της δομής του προϊόντος στον υπολογιστή. Όταν ολοκληρωθεί η αρχική σχεδίαση, τα αποτελέσματα πρέπει να ελεγχθούν ως προς τις αρχικές προδιαγραφές. Παραδοσιακά, προτού αναπτυχθούν τα προγράμματα σχεδίασης CAD, το βήμα αυτό περιελάμβανε την κατασκευή ενός φυσικού μοντέλου του σχεδιασμένου προϊόντος, το οποίο γενικά περιείχε μόνον τα τμήματα-κλειδιά. Σήμερα, σπάνια είναι απαραίτητη η κατασκευή ενός φυσικού μοντέλου. Τα προγράμματα CAD επιτρέπουν στους σχεδιαστές να προσομοιώνουν τη συμπεριφορά απίστευτα περίπλοκων προϊόντων και αυτές οι προσομοιώσεις χρησιμοποιούνται για να αποφασιστεί αν το σχέδιο που προέκυψε πληροί τις καθορισμένες προδιαγραφές. Αν διαπιστωθεί η ύπαρξη σφαλμάτων, τότε εκτελούνται οι αναγκαίες αλλαγές και το νέο σχέδιο ελέγχεται εκ νέου μέσω πάλι της διαδικασίας της προσομοίωσης. Παρότι μπορεί να μην ανιχνευθούν κάποια σχεδιαστικά σφάλματα, εν γένει η διάγνωση των περισσότερων προβλημάτων γίνεται σε αυτό το στάδιο. Όταν η προσομοίωση δείξει ότι το σχέδιο είναι σωστό, τότε κατασκευάζεται ένα πλήρες φυσικό πρωτότυπο του αντικειμένου. Το πρωτότυπο αυτό ελέγχεται εξονυχιστικά για να επιβεβαιωθεί ότι βρίσκεται σε συμφωνία με τις προδιαγραφές. Όλα τα σφάλματα που διαπιστώνονται κατά τον έλεγχο πρέπει να διορθωθούν. Τα σφάλματα αυτά ενδέχεται να είναι δευτερεύοντα και συχνά μπορούν να διορθωθούν επεμβαίνοντας άμεσα στο πρωτότυπο του προϊόντος. Στην περίπτωση σημαντικών σφαλμάτων είναι αναγκαία η σχεδίαση του προϊόντος εκ νέου και η επανάληψη των παραπάνω βημάτων. Όταν το πρωτότυπο περάσει όλους τους ελέγχους, τότε το προϊόν χαρακτηρίζεται ως ορθά σχεδιασμένο και μπορεί να οδηγηθεί στο τμήμα παραγωγής.



Σχήμα 6.5 : Η Διαδικασία Ανάπτυξης

## 6.6 Οικονομικά Θέματα Σχεδίασης Ενσωματωμένων Συστημάτων

Ένα σημαντικό στοιχείο για το σχεδιασμό των ενσωματωμένων συστημάτων, όπως και κάθε άλλου προϊόντος, είναι το κόστος. Το συνολικό είναι το άθροισμα δύο επιμέρους κόστων: το μη επαναλαμβανόμενο κόστος μηχανικής (non recurrent engineering cost, NRE) και το κόστος μονάδας (unit cost).

Το NRE κόστος SNRE περιέχει το κόστος που έχει δαπανήσει η εταιρία για την ανάπτυξη του συγκεκριμένου προϊόντος σε πληρωμές προσωπικού, σε σεμινάρια μηχανικών, σε αγορά ποικιλίας υλικών και δημιουργίας πρωτοτύπων, σε δοκιμές ποιότητας ή ηλεκτρομαγνητικής συμβατότητας, σε έξοδα των γραφείων (φως, νερό, θέρμανση, αναλώσιμα), σε ειδικό εξοπλισμό που απαιτήθηκε για τη δημιουργία του προϊόντος (π.χ. κάποιο εξειδικευμένο παλμογράφο), την αγορά δημοσιεύσεων ή πατεντών. Το κόστος αυτό συγκεντρώνει όλα τα παραπάνω και μόλις αποφασιστεί η μαζική κατασκευή του προϊόντος, τότε μπορεί να υπολογιστεί, ώστε να διευκρινιστεί και η τιμή πώλησης στο ράφι.

Το κόστος μονάδας  $S_u$  είναι ένα επαναλαμβανόμενο κόστος και εμπεριέχει όλα τα επιμέρους κόστη για τη αγορά των υλικών, την αποστολή στα κατάλληλα εργοστάσια, το τύπωμα των πλακετών, τη συναρμολόγηση, τη δοκιμή του κάθε προϊόντος, τη μισθοδοσία όσων ασχολούνται με την κατασκευή (ή σε περίπτωση που έχει γίνει εξωτερική ανάθεση(outsourcing) το κόστος αυτής, το πακετάρισμα και την αποστολή στις κεντρικές αποθήκες. Αυτό το κόστος πληρώνεται κάθε φορά που κατασκευάζεται ένα τέτοιο προϊόν. Το κόστος αυτό μεταβάλλεται κάθε φορά ανάλογα με την ισοτιμία των νομισμάτων (αν κάποια πληρωμή γίνεται σε άλλο νόμισμα, π.χ σε κινέζικα γουάν), ανάλογα με τον προμηθευτή (μερικές φορές επιλέγονται πιο οικονομικοί προμηθευτές για να μειώσουν το κόστος οι εταιρίες), ή ανάλογα και με την ποσότητα των μονάδων που κατασκευάζονται (άλλο κόστος έχει η κατασκευή 100 πλακετών και άλλο για  $10^6$ , σύμφωνα με την οικονομία κλίμακας).

Αφού υπολογισθούν αυτά τα δύο κόστη η εταιρία αποφασίζει το επιθυμητό περιθώριο κέρδους, έστω  $m$ . Κάποιες εταιρίες π.χ. αρκετά εξελιγμένων τηλεφώνων που τιμολογούν τα κινητά τους πάνω από 600 ευρώ, έχουν ένα περιθώριο κέρδους περίπου 50%. Ο υπολογισμός της τιμής πώλησης του προϊόντος υπολογίζεται από την εξίσωση:

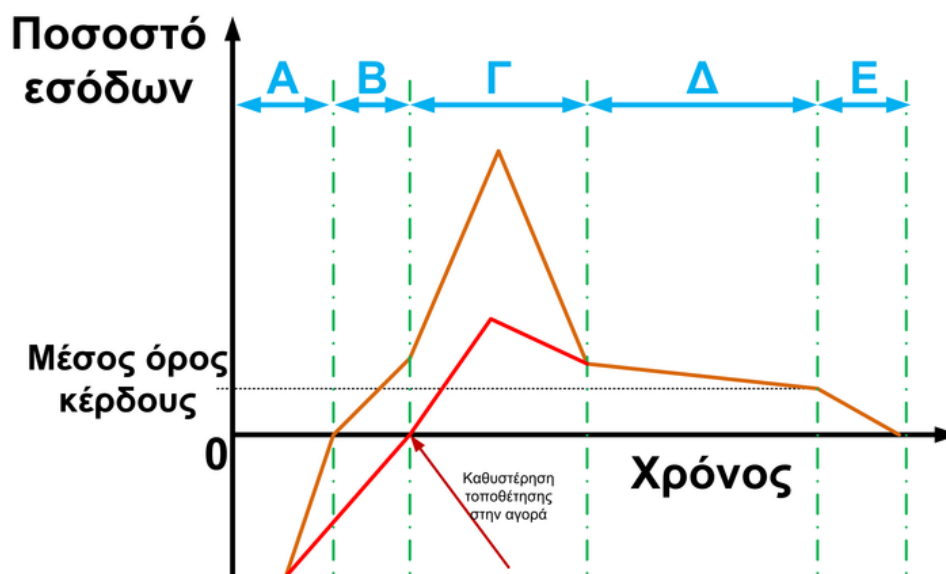
$$Cost = \frac{S_{NRE} + S_u}{1 - m}$$

Για να μειωθεί το κόστος NRE, μια πολύ χρήσιμη τακτική είναι η επαναχρησιμοποίηση αν όχι ολόκληρου του σχεδιασμού, όσο το δυνατόν μεγαλύτερου τμήματος, ώστε το NRE κόστος να διαιρεθεί σε πολλά προϊόντα. Έτσι, είναι πολύ συχνό το φαινόμενο της ίδιας βασικής αρχιτεκτονικής, να επαναχρησιμοποιείται σε πολλά προϊόντα με μόνη διαφορά την προσθήκη 1-2 νέων στοιχείων και τη συγγραφή κατάλληλου κώδικα. Ένα παράδειγμα είναι π.χ. μια επαγωγική κουζίνα με δυνατότητα εμφάνισης της ώρας σε μια οθόνη και χρονοπρογραμματισμό μαγειρέματος (η μόνη διαφορά του βασικού σχεδιασμού με αυτόν, είναι ότι έχει προστεθεί ένα σύστημα RTC με κόστος USD\$1, πέντε νέες οθόνες 7 στοιχείων για την εμφάνιση της ώρας, 2-3 κουμπιά και το κατάλληλο λογισμικό ). Έτσι, το NRE κόστος διαιρείται σε δύο προϊόντα και άρα είναι μικρότερο ανά προϊόν.

### 6.6.1 Χρόνος Εισαγωγής του προϊόντος στην Αγορά

Με τον όρο χρόνο εισαγωγής του προϊόντος στην αγορά (time-to-market), εννοούμε το χρονικό διάστημα από τη σύλληψη της ιδέας ως την τοποθέτηση του προϊόντος στο ράφι. Αυτό το χρονικό διάστημα καλύπτει όλες τις φάσεις έρευνας αγοράς, μοντελοποίησης, σχεδιασμού, κατασκευής και θα πρέπει να είναι αρκετά μικρό. Έχει υπολογιστεί ότι για τα ΕΣ θα πρέπει να είναι περίπου 6 μήνες, διαφορετικά υπάρχει σημαντική απώλεια εσόδων.

Κάθε προϊόν, έτσι και τα ΕΣ ακολουθούν συγκεκριμένες φάσεις κερδοφορίας. Η εικόνα παρουσιάζει αναλυτικά όλες αυτές της φάσεις. Στον κατακόρυφο άξονα υπάρχει το ποσοστό των εσόδων και ασφαλώς, όσο πιο μεγάλη είναι αυτή η τιμή τόσο περισσότερα έσοδα έχει η εταιρία, και στον οριζόντιο άξονα βρίσκεται ο χρόνος. Τα έσοδα μπορούν να υπολογιστούν αν υπολογιστεί το εμβαδόν του σχήματος που περικλείεται από την γραμμή του ΕΣ ως τον οριζόντιο άξονα και αφαιρεθεί το εμβαδόν του σχήματος κάτω από τον άξονα. Όσο μεγαλύτερο είναι το εμβαδόν τόσο μεγαλύτερα είναι τα έσοδα.



Σχήμα 6.6.1 : Κάθε προϊόν ακολουθεί συγκεκριμένες φάσεις κερδοφορίας. Μια καθυστέρηση τοποθέτησης στην αγορά μειώνει τα κέρδη.

Οι φάσεις που διέρχεται ένα προϊόν είναι πέντε. Μια σωστή τοποθέτηση του προϊόντος ακολουθεί την πορτοκαλί γραμμή (που έχει και τα περισσότερα έσοδα). Στην πρώτη φάση (φάση Α) δεν έχει αναπτυχθεί το προϊόν και έτσι η εταιρία δεν έχει έσοδα παρά μόνο έξοδα. Τα έξοδα αυτά κυρίως είναι τα ΝΡΕ, αφού βρίσκεται στη φάση της έρευνας και ανάπτυξης. Μόλις τελειώσει αυτή η φάση, το Ενσωματωμένο Σύστημα τοποθετείται στα ράφια (φάση Β) και με την κατάλληλη προώθηση υπάρχουν πωλήσεις και άρα έσοδα. Αν σε αυτό το σημείο δεν υπάρχουν έσοδα σημαίνει ότι κανείς δε θέλει αυτό το προϊόν και άρα είχε γίνει λάθος εκτίμηση και πρέπει να τροποποιηθεί κατάλληλα. Στη φάση Β βρίσκονται οι καταναλωτές που ονομάζονται early adopters (πρώιμοι καταναλωτές). Όταν συγκεντρωθεί μια κρίσιμη μάζα από αυτούς τους χρήστες και πουληθεί ένας ικανοποιητικός αριθμός ΕΣ, τότε θα προωθείται αυτόματα μέσα σε φόρουμ συζήτησης και περιοδικά, και έτσι οι πωλήσεις θα εκτοξευθούν (αν είναι ενδιαφέρον και καινοτόμο). Αυτή είναι η φάση Γ. Κάποια στιγμή τα έσοδα θα αρχίσουν να μειώνονται και αυτό γιατί είτε επειδή όσοι χρειάζονταν αυτό το προϊόν το έχουν αγοράσει, είτε επειδή άλλες εταιρίες έχουν αντιγράψει και κλωνοποιήσει και έτσι υπάρχουν και άλλοι προμηθευτές. Σιγά σιγά τα έσοδα δεν θα μειώνονται με τόσο μεγάλο ρυθμό και θα πλησιάζει το μέσο όρο κέρδους των προϊόντων της κατηγορίας. Το ΕΣ θα βρίσκεται ακόμη στα ράφια, αλλά οι πωλήσεις του θα είναι σταθερές (φάση Δ). Κάποια στιγμή θα αποσυρθεί το ΕΣ από

την αγορά, γιατί δε θα καλύπτει τις νέες ανάγκες, και η κατασκευάστρια εταιρία δε θα μπορεί να συντηρεί και να λειτουργεί μια γραμμή παραγωγής για ένα παρωχημένο προϊόν. Στη φάση E θα γίνει η απόσυρση (ονομάζεται EOL, end-of-life ή τέλος ζωής).

Αν ένα προϊόν καθυστερήσει να εισαχθεί στην αγορά (κόκκινη γραμμή), τότε είναι φανερό ότι τα έσοδα θα είναι αρκετά λιγότερα. Για αυτό θα πρέπει ο χρόνος εισαγωγής του προϊόντος στην αγορά να μη ξεπερνάει τους 6 μήνες, ενώ επίσης η τυπική διάρκεια ζωής είναι 18 μήνες.

## 6.6.2 Χαρακτηριστικά & Προκλήσεις

Ως τώρα έχει γίνει κατανοητό ότι το οικοσύστημα των Ενσωματωμένων Συστημάτων δεν είναι ομοιογενές, αφού κυμαίνεται από 8bit επεξεργαστές έως και 32bit, από διάφορες εταιρίες με πλήθος λειτουργικών συστημάτων, ή απλώς εκτέλεση σε 'γυμνό' υλικό, με ποικίλες απαιτήσεις. Μπορούμε όμως να σταχυολογήσουμε τα χαρακτηριστικά που συναντώνται σε Ενσωματωμένα Συστήματα:

- Εκτέλεση μιας λειτουργίας ή αρκετά περιορισμένων σε αριθμό διεργασίες: Όλα τα ΕΣ εκτελούν σε ένα αέναο βρόχο ένα πρόγραμμα που λαμβάνει δεδομένα από αισθητήρια και επιτελεί κάποιες ενέργειες.
- Περιορισμένων δυνατοτήτων: τα ΕΣ χρησιμοποιούν πολύ λιγότερους πόρους (π.χ. επεξεργαστικούς ή μνήμης) ως προς τα αντίστοιχα υπολογιστικά συστήματα των επιτραπέζιων υπολογιστών, και αυτό επειδή ο κατασκευαστής ενός ΕΣ γνωρίζει ακριβώς τους πόρους που χρειάζεται η εφαρμογή του και επλέγει το υλικό που ταιριάζει καλύτερα.
- Χαμηλή κατανάλωση ενέργειας και ισχύος: τα φορητά ΕΣ θα πρέπει να έχουν σχεδιαστεί με τέτοιο τρόπο, ώστε να έχουν χαμηλή στιγμιαία κατανάλωση ισχύος, αλλά και χαμηλό κατανάλωση ενέργειας σε μια συγκεκριμένη περίοδο. Αξίζει να σημειώσουμε τη διαφορά μεταξύ της κατανάλωσης ισχύος και της κατανάλωσης ενέργειας. Η κατανάλωση ισχύος αναφέρεται στο έργο που καταναλώνεται στη μονάδα του χρόνου, και στο σύστημα SI μετράται σε Joule / second, γνωστό και ως Watt. Η ενέργεια ομοίως που καταναλώνεται<sup>7</sup> σε ένα σύστημα μετράται σε killowatt-hour ή Joules, και αφορά τα watt που έχουν καταναλωθεί σε όλο το χρονικό διάστημα. Για παράδειγμα, ένα ΕΣ που καταναλώνει 1 Watt ισχύ σε 100 ώρες έχει συνολική κατανάλωση ενέργειας 100 watt-hours. Ένα άλλο ΕΣ που καταναλώνει 100 Watt ισχύ σε 1 ώρα, έχει ίδια συνολική κατανάλωση ενέργειας 100 watt-hours. Όμως το προηγούμενο ΕΣ έχει χαμηλότερη κατανάλωση ισχύος, και άρα είναι καλύτερο.
- Αντιδραστικά και Πραγματικού Χρόνου: Αρκετά ΕΣ χρησιμοποιούνται σε εφαρμογές που απαιτούν το σεβασμό αυστηρών χρονικών περιορισμών. Για αυτό το λόγο απαιτούνται επεξεργαστές με προβλέψιμο χρόνο εξυπηρέτησης διακοπών και ΛΣ (λειτουργικά συστήματα) που έχουν δημιουργηθεί ειδικά

για ανταπόκριση σε πραγματικό χρόνο (RTOS, real time operating systems-λειτουργικά συστήματα πραγματικού χρόνου).

- **Αξιοπιστία:** Τα ΕΣ κατασκευάζονται για να λειτουργούν χωρίς την παρέμβαση κάποιου ανθρώπινου χειριστή, που αν καταρρεύσει το ΛΣ μπορεί να πατήσει το reset και να ξεκινήσει η εκτέλεση από την αρχή. Τα ΕΣ μπορούν να τοποθετηθούν σε πολύ μεγάλα υψόμετρα (π.χ. δορυφόροι) ή σε πολύ χαμηλά (υποβρύχια αισθητήρια), που δε θα υπάρχει κάποιος κοντά. Θα πρέπει λοιπόν να λειτουργούν αξιόπιστα και αδιάκοπα.
- **Μικρό κόστος:** Το κόστος παραγωγής στα ΕΣ θα πρέπει να είναι ελάχιστο, αφού κατασκευάζονται σε πολύ μεγαλύτερες ποσότητες από ότι τα υπόλοιπα συστήματα. Για αυτό το λόγο στα ΕΣ, οι 8 bit επεξεργαστές καταλαμβάνουν ένα σημαντικό τμήμα της παγκόσμιας αγοράς επεξεργαστών (περίπου 25%), αν και σε όλες τις άλλες περιοχές τα 8bit συστήματα έχουν εξαφανιστεί. Ακόμη και σήμερα κατασκευάζονται και χρησιμοποιούνται ευρέως 8 bit επεξεργαστές που είχαν σχεδιαστεί το 1980 (π.χ. ο Intel 8051), και αυτό επειδή έχουν πολύ χαμηλό κόστος.
- **Μικρό μέγεθος και βάρος:** Τα ΕΣ είναι ενσωματωμένα μέσα σε άλλα συστήματα και έτσι πρέπει να καταλαμβάνουν όσο το δυνατόν μικρότερο όγκο. Αυτό σημαίνει και λιγότερο υλικό και άρα πιο περιορισμένους πόρους.
- **Συνεχή λειτουργία και μεγάλη διάρκεια ζωής:** Τα ΕΣ συνήθως από τη στιγμή που θα τεθούν σε λειτουργία θα λειτουργούν αδιάκοπα, και μάλιστα για πάρα πολλά χρόνια. Σε αντίθεση με έναν επιτραπέζιο υπολογιστή που έχει εκτιμώμενο χρόνο ζωής 2 με 4 χρόνια, τα ΕΣ έχουν εκτιμώμενο χρόνο ζωής πάνω από 15 χρόνια που σημαίνει ότι θα πρέπει να έχουν σχεδιαστεί κατάλληλα ώστε να 'αντέξουν' και να αντιμετωπίσουν με επιτυχία οποιοδήποτε πρόβλημα προκύψει.
- **Δυσκολία στην ενημέρωση λογισμικού:** Σε αντίθεση με τους τυπικούς υπολογιστές που μπορούν και ενημερώνουν το λογισμικό μέσω διαδικτύου (update), ένα ΕΣ μόλις βγει από το εργοστάσιο δε θα μπορέσει να ανανεωθεί το λογισμικό, κατά μεγάλη πιθανότητα και έτσι δύσκολα θα μπορέσουν να διορθωθούν προβλήματα (κυρίως ασφάλειας) σε διεργασίες που εκτελούνται, και άρα είναι πιο επιρρεπείς στη μη εξουσιοδοτημένη πρόσβαση.
- **Ασφάλεια:** Αρκετά ΕΣ τοποθετούνται σε κρίσιμες υποδομές, όπως π.χ. έλεγχο ενός πυρηνικού αντιδραστήρα. Θα πρέπει να υπάρχει μέριμνα ώστε αυτά τα ΕΣ να προστατεύονται όσο το δυνατόν καλύτερα, αφού αν αποκτήσει κάποιος κακόβουλος χρήστης πρόσβαση σε αυτά θα μπορέσει να τα απενεργοποιήσει ή να τα καταστρέψει. Πραγματικά σενάρια που είδαν το φως της δημοσιότητας αφορούν τη δημιουργία ενός ιού (STUXNET), που κατέστρεψε τα ενσωματωμένα συστήματα στους πυρηνικούς αντιδραστήρες του Ιράν [6], ή τον απομακρυσμένο έλεγχο των ΕΣ ενός οχήματος μέσω δικτύου κινητής τηλεφωνίας από κακόβουλους χρήστες.
- **Κατασκευασιμότητα (manufacturability):** Ένα ΕΣ θα πρέπει να κατασκευάζεται σε μεγάλες ποσότητες και για να υπάρχει μειωμένο κόστος



θα πρέπει να κατασκευάζεται αυτόματα ή σχεδόν αυτόματα. Με τον όρο κατασκευασσιμότητα εννοούμε τις τεχνικές και τα στοιχεία που πρέπει να επιλεγθούν, ώστε να μπορεί εύκολα να κατασκευαστεί σε μεγάλες ποσότητες. Σε αυτό τον όρο συγκαταλέγονται εκτός από το σχεδιασμό της τυπωμένης πλακέτας και του περιβλήματος (housing) και η επιλογή των προϊόντων και των προμηθευτών (ώστε να μπορούν απροβλημάτιστα να παρέχουν τα στοιχεία που απαιτούνται και να μην παρατηρηθεί κάποια έλλειψη, ένα παράδειγμα είναι ότι ο 8051 επεξεργαστής επειδή κατασκευάζεται από πολλές διαφορετικές εταιρίες είναι ένα προϊόν που αποτελεί μια ασφαλή επιλογή για 8bit επεξεργαστή, σε σύγκριση με έναν επεξεργαστή 8bit που κατασκευάζεται από μια μονό εταιρία και μπορεί να χρεοκοπήσει).

- Αυξημένη ανάγκη για μείωση του χρόνου εισαγωγής του προϊόντος στην αγορά (time-to-market): Τα ενσωματωμένα συστήματα προορίζονται κυρίως για καταναλωτικές συσκευές, οπότε θα πρέπει να έχουν ένα πολύ σύντομο χρόνο ολοκλήρωσης. Διαφορετικά, ο κατασκευαστής ενδέχεται να ζημιωθεί, όπως έχει συζητηθεί προηγουμένως.

Όπως φαίνεται από τις παραπάνω προδιαγραφές υπάρχει πολλές φορές σύγκρουση χαρακτηριστικών. Για παράδειγμα η επίτευξη της αξιοπιστίας έρχεται σε σύγκρουση με το μειωμένο χρόνο σχεδίασης. Δικαιολογημένα λοιπόν, ο σχεδιασμός των ενσωματωμένων συστημάτων θεωρείται αρκετά δύσκολος και αγχωτικός με απαίτηση ειδικών γνώσεων από πολλούς τομείς, και για αυτό οι μηχανικοί σχεδιασμού ενσωματωμένων συστημάτων έχουν αρκετά παχυλούς μισθούς.

## ΚΕΦΑΛΑΙΟ 7

### 7. ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ο σχεδιασμός των ενσωματωμένων συστημάτων μπορεί να επιτευχθεί σε διάφορα επίπεδα, ανάλογα με τις ανάγκες και τις προδιαγραφές που συνοδεύουν το εκάστοτε σύστημα. Από τη χρήση ενός μίνι έτοιμου υπολογιστή από το ράφι, που θα μπει μέσα σε ένα μηχάνημα ATM έως τη σχεδίαση εκ του μηδενός του επεξεργαστή και των περιφερειακών που θα χρησιμοποιηθούν. Ανάμεσα στις δύο αυτές ακραίες περιπτώσεις υπάρχουν πάρα πολλοί συνδυασμοί. Προκειμένου κάποιος μηχανικός να μπορεί να επιλέξει μια καλή υλοποίηση για το πρόβλημά του. Θα πρέπει να γνωρίζει θέματα αρχιτεκτονικής και τεχνολογίες υλοποίησης ενσωματωμένων συστημάτων, καθώς και τις προκλήσεις, προβλήματα και οφέλη που συνοδεύουν κάθε επιλογή.

#### 7.1 Θεμελιώδη στοιχεία της Αρχιτεκτονικής

Όπως έχει γίνει κατανοητό από την εισαγωγή, ένα ενσωματωμένο σύστημα βασίζεται στις ίδιες αρχές αρχιτεκτονικής που διέπουν κάθε προγραμματιζόμενο σύστημα. Τα δομικά του τμήματα είναι:

- (i) Μνήμη
- (ii) Επεξεργαστής
- (iii) Είσοδος / Έξοδος

Σε αυτό το κεφάλαιο συνοψίζουμε τις βασικές αρχές αρχιτεκτονικής, που βρίσκουν εφαρμογή στη συγκεκριμένη κατηγορία συστημάτων, περιγράφοντας βασικές γνώσεις για τα 3 δομικά τμήματα. Τα δεδομένα αποθηκεύονται ή ανακαλούνται από το (i), επεξεργάζονται και μετασχηματίζονται από το (ii)

Και μεταφέρονται μέσω διαύλων και δικτύων διασύνδεσης για το (iii).

#### 7.2 Η μνήμη σε ένα προγραμματιζόμενο σύστημα

Όλα τα ενσωματωμένα συστήματα φέρουν μια ποικιλία τεχνολογιών και ιδιοτήτων μνήμης. Ακόμη και το πιο απλό ενσωματωμένο σύστημα που βασίζεται σε ένα μικροεπεξεργαστή 8 bit, όπως το δημοφιλές ATMEGA328P, έχει διάφορες μνήμες, όπως μνήμη SRAM για τους καταχωρητές, 2KB μνήμη RAM για την εκτέλεση του προγράμματος, μνήμη FLASH για την αποθήκευση του προγράμματος, και μνήμη EEPROM 1KB για την αποθήκευση ρυθμίσεων. Στο κεφάλαιο αυτό περιγράφονται όλες αυτές οι τεχνολογίες (SRAM, EEPROM, FLASH) και πολλές άλλες που συναντάμε στα ενσωματωμένα συστήματα.

Η μνήμη χρησιμοποιείται για να διατηρεί τα δεδομένα (είτε μόνιμα είτε όσο διατηρείται η τροφοδοσία), και το πρόγραμμα εκτέλεσης του επεξεργαστή. Υπάρχει ποικιλία τεχνολογιών μνήμης, και συχνά χρησιμοποιείται ένα μίγμα μέσα σε ένα ενσωματωμένο σύστημα. Κάποιο είδος μνήμης θα διατηρήσει το περιεχόμενό της, ενώ δεν τροφοδοτείται καθόλου με ενέργεια (αυτή η κατηγορία μνήμης ονομάζεται non volatile-μη πτητική, ενώ η αντίθετη περίπτωση ονομάζεται volatile-πτητική),

όμως η πρόσβαση σε αυτή θα είναι αργή. Άλλες συσκευές μνήμης θα είναι μεγάλης χωρητικότητας, όμως θα απαιτούν στοιχεία κυκλώματος πρόσθετης υποστήριξης για την ανανέωση ή τον έλεγχο και θα είναι πιο αργές στην πρόσβαση.

Ακόμα άλλες συσκευές μνήμης θα θυσιάσουν την χωρητικότητα για την ταχύτητα, που παράγουν οι σχετικά μικρές συσκευές, όμως θα είναι σε θέση να αντεπεξέλθουν με το γρηγορότερο επεξεργαστή. Γενικά, υπάρχουν πολλές τεχνολογίες και κυκλώματα μνήμης, γιατί όπως ισχύει σε κάθε σχεδιασμό συστήματος, υπάρχουν πάρα πολλοί βαθμοί ελευθερίας και αναλόγως της βαρύτητας που δίνεται, προκύπτουν διαφορετικά χαρακτηριστικά. Για αυτό και έχει ειπωθεί ότι ενώ οι χρήστες θέλουν την ελευθερία της επιλογής (όπως π.χ. στα καταναλωτικά προϊόντα) οι σχεδιαστές συστημάτων την απεχθάνονται γιατί τους κάνει πολύ δύσκολο το έργο της επιλογής των κατάλληλων υποσυστημάτων.

Μια οποιαδήποτε μνήμη αποτελείται από κελιά, όπου το κάθε κελί αποθηκεύει ένα bit. Το bit αποθηκεύεται ως τάση, όπου η ύπαρξη ή απουσία τάσης καθορίζει αν θα είναι 1 ή 0. Υπάρχουν και κάποιες πιο σύνθετες υλοποιήσεις που χρησιμοποιούν πολλαπλές τάσεις στο ίδιο κελί (ονομάζονται MLC, multi-level cell ή κελί πολλαπλών επιπέδων τάσης [15]) και επιτρέπουν την αύξηση της χωρητικότητας στο ίδιο κελί δηλαδή αύξηση της χωρητικότητας στην ίδια επιφάνεια πυριτίου (συνήθως αποθηκεύονται 2 bit σε ένα κελί του 1 bit), που όμως αν και έχουν περάσει δεκαετίες από τότε που προτάθηκαν βρίσκονται σε ερευνητικό στάδιο, γιατί απαιτούνται πολύ ευαίσθητοι ανιχνευτές τάσης και είναι πολύ επιρρεπείς σε τροποποίηση και αλλοίωση των αποτελεσμάτων, ιδιαίτερα για τις πολύ μικρές τεχνολογίες υλοποίησης (π.χ. πλάτος τρανζίστορ 14nm).

Τα bit ομαδοποιούνται συνήθως σε λέξεις των 8 στοιχείων (1 Byte) ή σε κάποιες περιπτώσεις και σε άλλα μεγέθη όπως των 32 Byte. Έτσι λοιπόν, αν σε μια μνήμη υπάρχουν  $m$  λέξεις των  $n$  bit η κάθε μια, συνολικά η μνήμη θα φέρει  $m \cdot n$  bit. Αναφερόμαστε σε μια μνήμη ως μνήμη  $m \times n$  (' $m$ -από- $n$ '). Κάποιες μνήμες έχουν επιπρόσθετα bit για διάγνωση και επιδιόρθωση σφαλμάτων, όπως οι μνήμες με υποστήριξη ECC (error correcting code, κώδικα για διόρθωση σφαλμάτων). Οι ECC μνήμες χρησιμοποιούνται σε περιβάλλοντα ακραίων καταστάσεων λειτουργίας, όπως στο ΕΣ που θα τοποθετηθεί σε ένα δορυφόρο και θα βομβαρδίζεται συνεχώς από την κοσμική ηλεκτρομαγνητική ακτινοβολία, δημιουργώντας μαλακά σφάλματα (soft errors), δηλαδή προσωρινή αλλοίωση των τιμών των bit [16].

Μια μνήμη αποτελείται από  $m$  λέξεις των  $n$  bit, οπότε θα πρέπει να μπορεί κάποιος να διευθυνσιοδοτήσει αυτές τις  $m$  λέξεις, ώστε να διαβάσει στηνέξοδο δεδομένων της μνήμης, τη λέξη. Τα σήματα εισόδου διευθύνσεων  $\log_2(m)$  είναι απαραίτητα για να προσδιορίσουν μια συγκεκριμένη λέξη. Με άλλα λόγια, εάν μια μνήμη έχει  $k$  διευθύνσεις ως είσοδο, αυτή μπορεί να έχει μέχρι  $2^k$  λέξεις. Ο διάυλος των διευθύνσεων απαιτεί  $k$  bit. Αυτά μπορεί να βρίσκονται σε  $k$  διαφορετικές γραμμές ή σε λιγότερες με τη χρήση της πολύπλεξης. Η διευθυνσιοδότηση μιας μνήμης επιτρέπει την εγγραφή ή ανάγνωση μιας λέξης. Μερικές μνήμες μπορούν μόνο να προσπελαστούν (όπως η ROM), ενώ άλλες επιτρέπουν την προσπέλαση και την εγγραφή δεδομένων (όπως η RAM). Σε αντίθεση με τη μνήμη μόνο για ανάγνωση, δεν υπάρχει μνήμη μόνο για εγγραφή (δηλαδή, δε μπορεί να γίνει ανάγνωση), και ένα

φύλλο δεδομένων (data sheet) για μια υποτιθέμενη τέτοια συσκευή, αποτελεί κλασσικό ανέκδοτο των αρχιτεκτόνων της δεκαετίας του 1970.

Όλες οι μνήμες μπορούν να διευθυνσιοδοτούν λέξεις. Ένα στοιχείο που διαφοροποιεί τις μνήμες, είναι πόσα bit ή Byte έχει κάθε λέξη. Συνήθως κάθε λέξη είναι 8bit, ή 32bit, και αυτό σημαίνει ότι με μια διεύθυνση μνήμης ο επεξεργαστής αποκτά πρόσβαση σε 8 bit ή 32 bit αντίστοιχα. Άρα, σε περίπτωση που χρησιμοποιούνται 8 bit απαιτούνται 4 διαδοχικές διευθύνσεις για τη μεταφορά 32 bit. Υπάρχουν όμως και ειδικοί τρόποι πρόσβασης σε κάποιες μνήμες, που στέλνεται μόνο η αρχική διεύθυνση και το συνολικό μέγεθος των Byte που απαιτούνται για την ανάγνωση (burst transfer, μεταφορά ριπής).

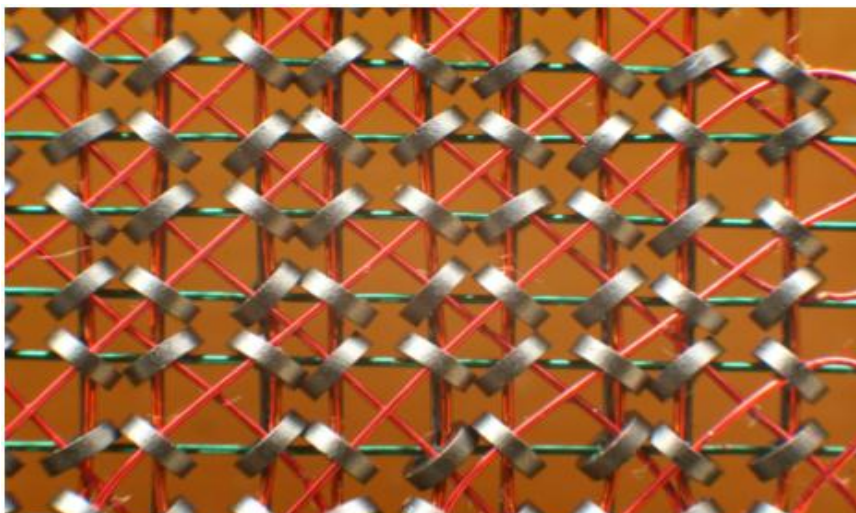
Τα ολοκληρωμένα τσιπ μνήμης έρχονται σε διαφορετικά μεγέθη, ως προς την οργάνωση. Παραδείγματος χάριν, ένα τσιπ DRAM (δυναμική RAM) μπορεί να περιγραφεί ως 4MBx1 (οργανωμένο σε bit), ενώ ένα τσιπ SRAM (στατική RAM) μπορεί να είναι 512KBx8 (οργανωμένο σε λέξεις). Και στις δύο περιπτώσεις, κάθε τσιπ έχει ακριβώς την ίδια ικανότητα αποθήκευσης, αλλά είναι οργανωμένο με διαφορετικούς τρόπους. Στην περίπτωση της DRAM, θα απαιτούνταν οκτώ τσιπ συνδεδεμένα όλα στις ίδιες γραμμές διευθύνσεων, όπου το πρώτο τσιπ θα έδινε το bit ελάχιστης σημαντικότητας (LSB), ενώ το τελευταίο τσιπ θα μετέφερε το μεγαλύτερης σημαντικότητας (MSB). Ασφαλώς, η πρόσβαση θα ήταν παράλληλη σε αυτά τα τσιπ και ταυτόχρονα θα τοποθετούνταν στην έξοδο και τα 8 bit μόλις αποκωδικοποιούνταν η διεύθυνση μνήμης.

Στην περίπτωση της SRAM θα απαιτούνταν μόνο ένα τσιπ. Εντούτοις, επειδή συνήθως τα τσιπ DRAM οργανώνονται παράλληλα, υποστηρίζουν ταυτόχρονη προσπέλαση και επιτυγχάνονται καλές ταχύτητες πρόσβασης. Το τελικό μέγεθος για τη DRAM είναι (4MBx1)x8 ή 32MB.

### **7.3 Μνήμη τυχαίας προσπέλασης – Random Access Memory, RAM**

Το ακρωνύμιο RAM σημαίνει μνήμη τυχαίας πρόσβασης ή προσπέλασης (Random Access Memory). Αυτό όμως είναι ένας ατυχής χαρακτηρισμός, δεδομένου ότι σχεδόν όλη η μνήμη των υπολογιστών μπορεί να θεωρηθεί ‘τυχαίας πρόσβασης’. Βέβαια πρόσφατα, έχουν προταθεί στα ενσωματωμένα συστήματα και μνήμες ‘σειριακές’ (serial memory), όπως το Hybrid Memory Cube [18] από μια κοινοπραξία μεγάλων εταιριών (Xilinx, Altera, Microsoft, Samsung, ARM κ.α.), με το όρο ‘serial’ όμως να αφορά τον τρόπο μεταφοράς των δεδομένων προς και από αυτή τη συσκευή και όχι την εσωτερική δομή, όπου πάλι είναι τύπου RAM.

Ο όρος RAM δημιουργήθηκε με τους πρώτους υπολογιστές με τρανζίστορ (δεκαετία 1960), αφού ως τότε υπήρχε ένα είδος μνήμης πυρήνα (magnetic core memory), όπως φαίνεται στην Εικόνα και η μνήμη ήταν αρκετά αργή στην ανάγνωση και εγγραφή. Επίσης, εκείνη την εποχή, υπήρχαν μαγνητικές ταινίες αποθήκευσης ως μνήμες, που υποστήριζαν μόνο τη σειριακή εγγραφή και ανάγνωση. Η επινόηση της γρήγορης μνήμης RAM, έδωσε μια σημαντική ώθηση στους υπολογιστές.



Σχήμα 7.3 (Εικόνα 1): Πριν από τη RAM χρησιμοποιούνταν μαγνητικοί δακτύλιοι μνήμης (1955-1975).

Η RAM είναι η μνήμη λειτουργίας στους ηλεκτρονικούς υπολογιστές. Είναι το μέρος όπου ο επεξεργαστής μπορεί εύκολα να γράψει δεδομένα για προσωρινή αποθήκευση (όσο τροφοδοτείται η μνήμη με ρεύμα). Η RAM είναι γενικά ευμετάβλητη (πτητική), χάνοντας το περιεχόμενό της όταν χάνει το σύστημα ενέργεια (ρεύμα). Οποιοσδήποτε πληροφορίες αποθηκεύονται στη RAM και πρέπει να διατηρηθούν, πρέπει να γραφτούν σε κάποια μορφή μόνιμης αποθήκευσης πριν το σύστημα χάσει την ισχύ του. Προκειμένου να αντιμετωπιστεί το μειονέκτημα της πτητικότητας των δεδομένων, που είναι κρίσιμης σημασίας σε διακομιστές, έχουν κατασκευαστεί μνήμες RAM με εφεδρικό σύστημα τροφοδότησης ενέργειας (Εικόνες 2 και Εικόνα 3), ώστε αν διακοπεί η τροφοδοσία να συνεχίσουν να τροφοδοτούνται με ρεύμα για να διατηρούν τα δεδομένα.



Σχήμα 7.3 (Εικόνα 2): Ένα ΕΣ (Gecko100) που φέρει μνήμη RAM με υποστήριξη τροφοδότησης από μπαταρία.



Σχήμα 7.3 (Εικόνα 3): Μνήμη DRAM με ενσωματωμένη μπαταρία.

Οι μνήμες RAM διαιρούνται σε δύο οικογένειες: στατική RAM (επίσης γνωστή ως Static RAM ή SRAM) και δυναμική RAM (επίσης γνωστή ως dynamic RAM ή DRAM). Οι μνήμες SRAM χρησιμοποιούν πύλες λογικής σε συνδεσμολογία μανδαλωτών ή flip-flop, ώστε να διατηρούν κάθε bit των στοιχείων. Αυτή η κατηγορία είναι η γρηγορότερη διαθέσιμη μορφή RAM, απαιτεί ελάχιστα εξωτερικά στοιχεία κυκλώματος υποστήριξης, και έχει σχετικά μικρή κατανάλωση ενέργειας. Στα μειονεκτήματά τους συμπεριλαμβάνονται η μειωμένη χωρητικότητα (αφού συνήθως απαιτούν 6 τρανζίστορ, περίπου 6 φορές περισσότερη επιφάνεια από τις DRAM), το αυξημένο κόστος και η χαμηλότερη πυκνότητα ολοκλήρωσης. Για αυτό, χρησιμοποιούνται μόνο σε ειδικού τύπου κυκλώματα που απαιτούν μεγάλη ταχύτητα πρόσβασης (όπως οι κρυφές μνήμες cache memorie). Η DRAM αντί για τρανζίστορ χρησιμοποιεί πυκνωτές για την αποθήκευση φορτίων, και αισθητήρια για την ανίχνευση ή όχι του φορτίου. Οι σειρές πυκνωτών θα κρατήσουν τη φόρτιση τους μόνο για μια μικρή χρονική περίοδο προτού η διαρροή εξαλείψει το φορτίο.

Επομένως, η DRAM χρειάζεται μια συνεχή αναζωογόνηση-ανανέωση (refresh) όπου διαβάζεται το κελί και ξαναγράφεται, κάθε λίγα χιλιοστά του δευτερολέπτου (για παράδειγμα η μνήμη DRAM τεχνολογίας DDR απαιτεί κάθε κελί να ανανεώνεται ανά 7.8  $\mu$ s). Αυτή η διαρκής ανάγκη ανανέωσης απαιτεί πρόσθετη υποστήριξη και μπορεί να καθυστερήσει την πρόσβαση σε αυτή, αν τη στιγμή που πρόκειται να γίνει μια ανάγνωση, απαιτείται ανανέωση του κελιού. Οι DRAM είναι συσκευές μνήμης υψηλής-χωρητικότητας διαθέσιμες και έρχονται σε μια ευρεία και διαφορετική ποικιλία υπό κατηγοριών. Λόγω της πολυπλοκότητας που έχουν μπορούν να διασυνδεθούν μόνο με κανονικούς επεξεργαστές, και δεν είναι πρακτική η σύνδεση με μικρούς μικροελεγκτές ενσωματωμένων συστημάτων. Οι περισσότεροι επεξεργαστές με δυνατότητα διευθυνσιοδότησης μεγάλων μεγεθών μνήμης, έχουν εγγενή υποστήριξη για DRAM (ή αν δεν έχουν μπορούν να συνδεθούν σε ένα ενδιάμεσο ελεγκτή μνήμης).

Ένα σημαντικό πρόβλημα με τη χρήση της DRAM, προκύπτει από το γεγονός ότι είναι εξωτερική μνήμη (δε βρίσκεται στο ίδιο τσιπ με τον επεξεργαστή) και άρα υπάρχει αυξημένη καθυστέρηση και ενεργειακή ανάγκη για κάθε πρόσβαση. Για να θεραπευτεί κάπως αυτό το μειονέκτημα κάποιοι επεξεργαστές έχουν κρυφές (μη ορατές) μνήμες εντολών και δεδομένων πάνω στο ίδιο πυρίτιο με τον επεξεργαστή και αποθηκεύουν τις πρόσφατες προσβάσεις μνήμης. Αυτές οι μνήμες είναι (συχνά, αλλά όχι πάντα) εσωτερικές στους επεξεργαστές και υλοποιούνται με γρήγορα στοιχεία μνήμης και μεγάλης-ταχύτητας διαύλους δεδομένων. Έτσι λοιπόν δεδομένα που επαναχρησιμοποιούνται μπορούν να αποθηκευτούν προσωρινά σε αυτή τη μνήμη μόλις διαβαστούν και έτσι την επόμενη φορά που θα ζητηθούν να βρίσκονται κοντά στον επεξεργαστή. Οι κρυφές μνήμης είναι τεχνολογίας SRAM.

Οι μνήμες DRAM παρουσιάζουν μια διαρκή εξέλιξη τεχνολογίας από το 1975 που είχαν πρωτοπαρουσιασθεί, την οποία θα συνοψίσουμε σε λίγες γραμμές: Μια αύξηση των επιδόσεων επήλθε όταν η μνήμη συγχρονίστηκε με το ρολόι του διαύλου που

συνέδεε τον επεξεργαστή με τη μνήμη. Βέβαια, η μνήμη θα έπρεπε να έχει αυστηρούς χρονισμούς και να υποστηρίζει παραμετροποιήσιμη καθυστέρηση (wait cycles), ώστε η όλη επικοινωνία να είναι συγχρονισμένη. Αυτή η μνήμη ονομάζονταν synchronous DRAM (SDRAM) και είχε ταχύτητες λειτουργίας (Mhz) SDRAM PC-66, SDRAMPC-100, SDRAM PC-133.

Μια βελτίωση στη SDRAM πραγματοποιήθηκε όταν άλλαξε η επικοινωνία με τη μνήμη, και από εκεί που η επικοινωνία γίνονταν με τις στάθμες της λογικής (1 και 0), στη συνέχεια γίνονταν με τις ανερχόμενες και κατερχόμενες ακμές του ρολογιού. Αυτό επέτρεψε να διπλασιαστεί ο ρυθμός επικοινωνίας με τη μνήμη (double data rate, DDR) για το ίδιο ρολόι του συστήματος. Οι ενδεικτικές ταχύτητες σε Mhz για αυτές τις μνήμες ήταν DDR-200, DDR-233, DDR-333, DDR-400. Μια ακόμη βελτίωση συντελέστηκε όταν αυξήθηκε η συχνότητα λειτουργίας, μειώθηκε η τάση τροφοδοσίας και διπλασιάστηκε η χωρητικότητα του διαύλου δεδομένων (από 128 bit σε 256 bit). Αυτές οι μνήμες ονομάστηκαν DDR2 SDRAM με συχνότητες λειτουργίας DDR2-400, DDR2-533, DDR2667, DDR2-800 και DDR2-1066.

Η επόμενη γενιά των μνημών (DDR3 SDRAM) διπλασίασε το εύρος του διαύλου σε 512bit, και την τάση λειτουργίας στα 1.5 Volt και συχνότητα λειτουργίας DDR3-1600. Η επόμενη γενιά που ακόμη δεν έχει καθιερωθεί από την αγορά γιατί είναι στα πρώιμα στάδια (παρουσιάστηκε το 2014), είναι η DDR4, με 1.2 Volt και με συχνότητες λειτουργίας DDR4-2133 και DDR4-3200. Οι μνήμες της τελευταίας γενιάς έχουν τόσο μεγάλες ταχύτητες που μπορούν να τις εκμεταλλευτούν μόνο εφαρμογές που κάνουν εντατική χρήση της μνήμης (εφαρμογές διακομιστών), για αυτό και η αύξηση της απόδοσης σε ένα σύστημα που αναβαθμίζεται από DDR3 σε DDR4 είναι 0 έως 5%, ενώ το κόστος αγοράς της μνήμης είναι 2 με 3 φορές περισσότερο.

Επίσης, υπάρχουν και τροποποιήσεις των DDR, όπως οι LPDDR(X) (X=2,3,4) με το LP να σημαίνει low-power (χαμηλή κατανάλωση), δηλαδή μνήμες που λειτουργούν με μειωμένη τάση τροφοδοσίας και χρησιμοποιούνται σε φορητές συσκευές (π.χ. σε κινητά), GDDR(X) (X=2,3,4,5) με το G να σημαίνει graphics (γραφικά) και είναι μνήμες που χρησιμοποιούνται σε κάρτες γραφικών και φέρουν κάποιες εξειδικευμένες αρχιτεκτονικές τροποποιήσεις, Rambus RDRAM και η εξέλιξή της ως XDRSDRAM, οι οποίες είναι κλειστές τεχνολογίες με πατέντες που κατέχει η Rambus και υποστηρίζουν μεγαλύτερες συχνότητες λειτουργίας από τις αντίστοιχες DDR, αλλά με αυξημένο κόστος.

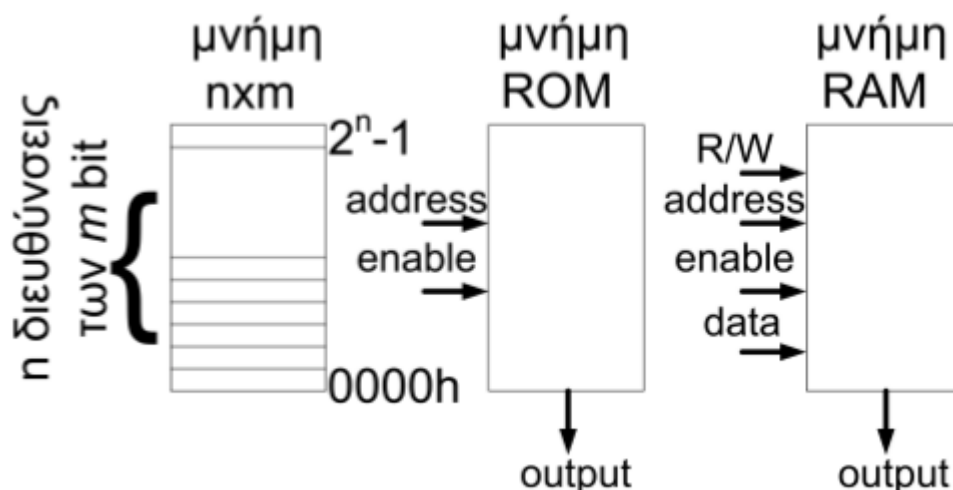
Σε ερευνητικό επίπεδο (μετά τη DDR4) υπάρχει η τεχνολογία Υβριδικού Κύβου Μνήμης (Hybrid Memory Cube, HMC) που συναγωνίζεται την τεχνολογία Μνήμη Υψηλού Εύρους ζώνης (High Bandwidth Memory, HBM) ως προς το ποια θα επικρατήσει. Οι τεχνολογίες αυτές χρησιμοποιούν την τρισδιάστατη ολοκλήρωση, δηλαδή την κατακόρυφη τοποθέτηση πολλών τσιπ και την κατάλληλη σύνδεση με οπές (vias), ενώ έχουν μειώσει τις γραμμές μεταφοράς δεδομένων χρησιμοποιώντας υψηλής ταχύτητας σειριακή αποστολή και λήψη δεδομένων από τον επεξεργαστή.

Όπως και σε άλλα δομικά στοιχεία, η τεχνολογία δε σταματάει να κάνει άλματα ανάπτυξης.

## 7.4 Μνήμη μόνο για Ανάγνωση – ROM

Η Read Only Memory (ROM), ή μνήμη μόνο για ανάγνωση, είναι μια μνήμη που μπορεί να προσπελαστεί για ανάγνωση, αλλά όχι να αποθηκεύσει δεδομένα κατά την τυπική λειτουργία. Φυσικά, υπάρχει τρόπος να γραφούν μια ή ελάχιστες φορές δεδομένα, αλλά αυτή η διαδικασία καλείται προγραμματισμός και όχι αποθήκευση. Τέτοιος προγραμματισμός γίνεται συνήθως εκτός σύνδεσης (off line), δηλαδή όταν δεν χρησιμοποιείται ενεργά η μνήμη ως μνήμη σε ένα ΕΣ. Μια ROM προγραμματίζεται συνήθως πριν τοποθετηθεί στην κατάλληλη υποδοχή του ενσωματωμένου συστήματος.

Το Σχήμα 7.4 παρέχει ένα διάγραμμα δομικών στοιχείων μιας ROM.



Σχήμα 7.4: Βασικές μνήμες (a) Λέξεις και Bits ανά λέξη  
(b) Διάγραμμα Τμήματος ROM  
(c) Διάγραμμα Τμήματος RAM

Μια ROM μπορεί να χρησιμοποιηθεί για διάφορους σκοπούς σε ένα ενσωματωμένο σύστημα. Για παράδειγμα, μπορεί να αποτελεί το πρόγραμμα εκκίνησης (bootloader), δηλαδή το πρώτο πρόγραμμα που θα εκτελέσει ένα ενσωματωμένο σύστημα μόλις τροφοδοτηθεί με ρεύμα. Το πρόγραμμα αυτό μπορεί στη συνέχεια να μεταφορτώσει το υπόλοιπο πρόγραμμα ή το λειτουργικό σύστημα από κάποιο μέσο μεγαλύτερης χωρητικότητας, όπως μια μνήμη Flash. Για μικρά ενσωματωμένα συστήματα μια μνήμη ROM μπορεί να φέρει όλο το πρόγραμμα και να μην απαιτείται κάτι παραπάνω.

Όμως, όπως θα δούμε παρακάτω, οι μνήμες RAM δε χρησιμοποιούνται πια, αφού έχουν κατασκευαστεί νέες οικογένειες μόνιμης αποθήκευσης με μεγαλύτερη ευελιξία. Μια ROM επίσης, μπορεί να χρησιμοποιηθεί για την υλοποίηση ενός συνδυαστικού



κυκλώματος. Μπορεί να υλοποιηθεί οποιαδήποτε συνδυαστική συνάρτηση  $k$  μεταβλητών με τη χρησιμοποίηση μιας  $2k \times 1$ ROM, και μπορούν να υλοποιηθούν  $n$  λειτουργίες των ίδιων μεταβλητών  $k$  χρησιμοποιώντας μια  $2k \times n$  ROM.

Η μνήμη ROM προγραμματίζεται μόνο στα εργοστάσια κατασκευής και δεν είναι δυνατός ο επαναπρογραμματισμός της. Η διαδικασία της φόρτωσης του λογισμικού σε μία ROM είναι γνωστή ως 'κάψιμο (burn) της ROM'. Αυτός ο όρος προέρχεται από το γεγονός ότι η διαδικασία προγραμματισμού εκτελείται περνώντας ένα αρκετά μεγάλο ρεύμα μέσω των κατάλληλων διακοπών που έχει το εσωτερικό πλέγμα της ROM για 'να τους καταστρέψει' ή να τους 'κάψει' και με αυτόν τον τρόπο δημιουργεί ένα μηδέν σε εκείνη την θέση bit.

Μια συσκευή γνωστή ως 'καυστήρας' ή προγραμματιστής ROM μπορεί να επιτελέσει αυτή τη λειτουργία ή αν η ROM είναι ειδικού τύπου να προγραμματιστεί εντός συστήματος (In-System Programming-ISP) ή εντός κυκλώματος (In-Circuit Programming ICP). Η απλή μνήμη ROM προγραμματίζεται μόνο μια φορά, γιατί κατά τον προγραμματισμό τροποποιείται μόνιμα το κύκλωμα. Αυτή η μνήμη χρησιμοποιείται σε ενσωματωμένο σύστημα που μόλις φύγουν από το εργοστάσιο δεν πρόκειται να μεταβληθούν (ή να ενημερωθούν).

Μια πιο ευέλικτη κατηγορία ROM, είναι η PROM (programmable ROM), η οποία μπορεί να προγραμματιστεί μια φορά από το χρήστη με ειδικό εξοπλισμό. Αυτές οι μνήμες ταιριάζουν καλύτερα για τη γρήγορη προτυποποίηση και στα ενσωματωμένα συστήματα χαμηλών απαιτήσεων.

Για να προγραμματιστεί μια συσκευή PROM, ο χρήστης παρέχει ένα αρχείο που καθορίζει το επιθυμητό περιεχόμενο για τη μνήμη. Ένα κομμάτι του εξοπλισμού αποκαλούμενο προγραμματιστής ROM (σημείωση: ο προγραμματιστής είναι εξοπλισμός υλικού, όχι ένα πρόσωπο που γράφει κώδικα λογισμικού) και έπειτα διαμορφώνει κάθε προγραμματίσιμη σύνδεση σύμφωνα με το αρχείο. Μόλις προγραμματιστεί η ROM καταστρέφονται με μεγάλο ρεύμα οι κατάλληλες διασυνδέσεις, και οι εναπομείναντες διασυνδέσεις υλοποιούν το κύκλωμα. Οι συνδέσεις που καταστρέφονται δε μπορούν ποτέ να επανεγκαθιδρυθούν. Για αυτόν τον λόγο, η βασική PROM αναφέρεται συχνά ως μιας φορές χρονοπρογραμματίσιμη συσκευή, ή OTP (One Time Programmable).

## **7.5 Electrically Erasable Programmable Memory – EEPROM**

Η EEPROM είναι ηλεκτρικά διαγράψιμη μνήμη μόνο για ανάγνωση, επίσης γνωστή ως EEPROM (ηλεκτρικά διαγράψιμη και προγραμματίσιμη μνήμη μιας ανάγνωσης). Πολύ σπάνια, καλείται επίσης ηλεκτρικά μεταβλητή μνήμη μιας ανάγνωσης (Electrically alterable read-only memory, EAROM).

Η EEPROM μπορεί να σβηστεί και να επαναπρογραμματιστεί μέσα σε ένα κύκλωμα, σε αντίθεση με τις προηγούμενες μνήμες που απαιτείται η απομάκρυνση από το κύκλωμα και η τοποθέτηση σε ειδική συσκευή. Η χωρητικότητα τους είναι

σημαντικά μικρότερη από το τυποποιημένο ROM (μόνο μερικά kilobyte), και έτσι δεν χρησιμοποιούνται για αποθήκευση κώδικα προγραμματισμού. Αντ' αυτού, χρησιμοποιούνται, για να κρατούν τις παραμέτρους των συστημάτων και τις πληροφορίες που διατηρούνται κατά την αποσύνδεση.

Είναι κοινό για πολλούς μικροελεγκτές (όπως ATMEGA 328P), να ενσωματώνουν ένα μικρό τσιπ EEPROM για την αποθήκευση των παραμέτρων του συστήματος. Αυτό είναι ιδιαίτερα χρήσιμο στα ενσωματωμένα συστήματα και μπορεί να χρησιμοποιηθεί για την αποθήκευση των διευθύνσεων δικτύων, τοποθετήσεις διαμόρφωσης, αύξοντες αριθμούς, που συντηρούν τα αρχεία, κτλ. Για παράδειγμα, σε ΕΣ διαδικτύου που έχει σχεδιάσει ο συγγραφέας του βιβλίου, έχει τοποθετήσει κώδικα που ενεργοποιείται και διαβάζει τις εφεδρικές ρυθμίσεις λειτουργίας από την EEPROM σε περίπτωση που το ΕΣ δε μπορεί να συνδεθεί σε κάποιο διακομιστή ή να λάβει τις αυτόματες ρυθμίσεις από DHCP.

## 7.6 Η μνήμη FLASH

Η FLASH είναι η νεότερη τεχνολογία ROM και είναι αρκετά δημοφιλής. Συνήθως, συνυπάρχει μαζί με τη μνήμη EEPROM ή σε κάποια ΕΣ την έχει αντικαταστήσει τελείως. Η μνήμη FLASH έχει τα πλεονεκτήματα της επαναπρογραμματισιμότητας εντός κυκλώματος (όπως η EEPROM) και τη μεγάλη χωρητικότητα όπως η ROM. Τα τσιπ FLASH αναφέρονται μερικές φορές ως 'flash ROMs' ή 'flash RAMs'. Δεδομένου ότι δεν είναι όπως τα τυποποιημένα ROMs ή όπως η τυποποιημένη RAM, προτιμούμε ακριβώς να τα λέμε 'FLASH' για αποφυγή σύγχυσης.

Η FLASH οργανώνεται κανονικά όπως οι τομείς σε ένα μαγνητικό δίσκο, και έχει το πλεονέκτημα ότι οι μεμονωμένοι τομείς μπορούν να διαγραφούν και να ξαναεγγραφούν χωρίς να επηρεάζουν το περιεχόμενο του υπολοίπου της συσκευής. Συγκεκριμένα, πριν μπορέσει να γραφτεί ένας τομέας, πρέπει να διαγραφεί. Μπορεί απλά να ξαναεγγραφτεί, όπως γίνεται και με ένα RAM.

Υπάρχουν αρκετές διαφορετικές τεχνολογίες FLASH, και οι απαιτήσεις διαγραφής και προγραμματισμού των συσκευών FLASH ποικίλλουν από κατασκευαστή σε κατασκευαστή. Οι δυο βασικές τεχνολογίες FLASH είναι η NOR FLASH και η NAND FLASH. Οι μνήμες FLASH είναι ιδιαίτερα δημοφιλείς και χρησιμοποιούνται σχεδόν σε όλα τα ΕΣ για την αποθήκευση δεδομένων και προγράμματος. Το μεγάλο πλεονέκτημα είναι ότι μπορούν να ενημερωθούν από τον κατασκευαστή εν ώρα λειτουργίας, χωρίς να χρειάζεται να μετακινηθεί η συσκευή (αν υπάρχει σύνδεση με το διαδίκτυο ή δυνατότητα σύνδεσης συσκευής USB που φέρει την ενημέρωση).

Η NOR FLASH επιτρέπει τυχαία πρόσβαση σε οποιοδήποτε byte αλλά έχει μεγάλους χρόνους εγγραφής και είναι ακριβό κύκλωμα. Επειδή επιτρέπει διευθυνσιοδότηση ανά byte μπορεί να αντικαταστήσει άμεσα τα υπόλοιπα είδη ROM σε οποιοδήποτε κύκλωμα (drop-in replacement), και μπορεί να περιέχει και κώδικα που θα εκτελείται άμεσα από αυτήν (π.χ. ROM). Η NAND FLASH έχει μεγαλύτερη πυκνότητα αποθήκευσης, μικρότερο κόστος, λιγότερα καλώδια και πιο γρήγορους χρόνους εγγραφής, αλλά το μειονέκτημα είναι ότι οι προσβάσεις γίνονται πάντα σε ομάδες από bits (π.χ. 4096).

Έτσι δεν υπάρχει άμεση τυχαία προσπέλαση σε οποιοδήποτε σημείο, αφού μεταφέρονται όλα τα bit σε κάθε πρόσβαση. Χρησιμοποιείται μόνο για αποθήκευση και όχι για άμεση εκτέλεση όπως η NOR FLASH. Αν απαιτηθεί να εκτελεστεί ο κώδικας που περιέχει, αντιγράφεται η αντίστοιχη ομάδα στη μνήμη RAM, και εκτελείται από εκεί. Εντούτοις, επειδή είναι αρκετά φθηνή χρησιμοποιείται σε πάρα πολλά ΕΣ. Όλες οι κάρτες SD για παράδειγμα που τοποθετούνται στα κινητά τηλέφωνα ή σε ΕΣ, είναι τεχνολογίας NAND FLASH.

## 7.7 Τρόπος Αποθήκευσης των Byte

Αναπόσπαστο στοιχείο κάθε υπολογιστή είναι η μνήμη. Στην πληθώρα των περιπτώσεων ένας επεξεργαστής έχει μνήμη εντός (on-chip) και εκτός (off-chip) ολοκληρωμένου κυκλώματος. Στην on-chip μνήμη συγκαταλέγονται οι φανεροί και μη φανεροί καταχωρητές, η κρυφή μνήμη, και η μνήμη FLASH ή EEPROM (αν υπάρχει). Η off-chip μνήμη ομοίως μπορεί να είναι FLASH ή κάποιο είδος RAM. Τις περισσότερες περιπτώσεις η off-chip μνήμη είναι διευθυνσιοδοτούμενη ανά Byte, δηλαδή κάθε διεύθυνση μνήμης αντιστοιχεί σε μια θέση μνήμης χωρητικότητας 8 bit.

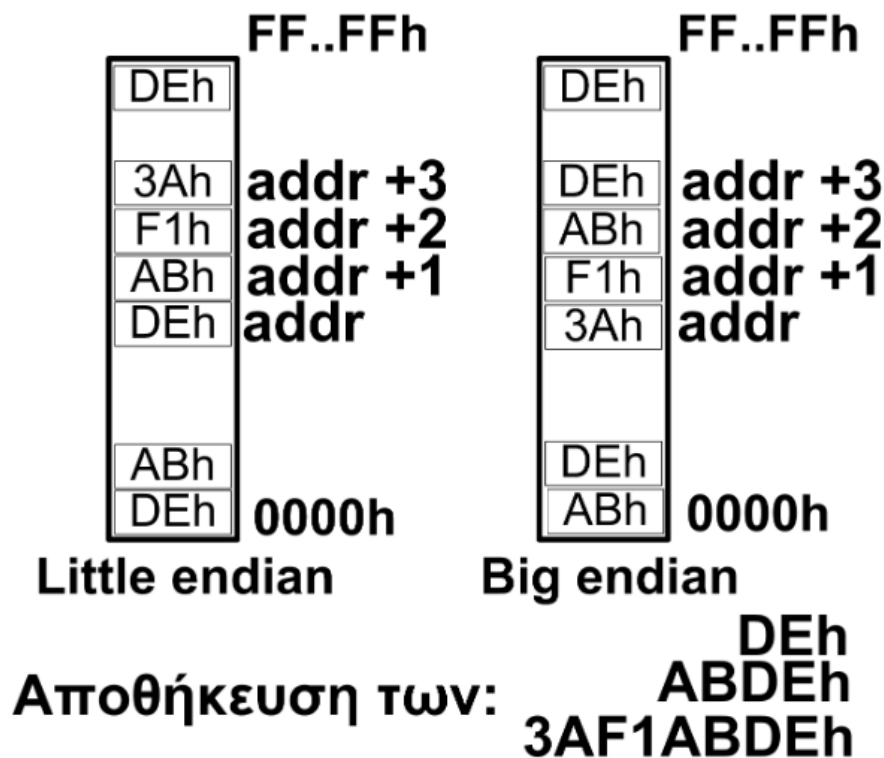
Σε περίπτωση που ο επεξεργαστής χρησιμοποιεί μεγέθη αποθήκευσης δεδομένων μεγαλύτερα του 1 Byte (multi Byte), υπάρχουν δυο τρόποι αποθήκευσης στην εξωτερική μνήμη· αυτό ονομάζεται endianness (σειρά αποθήκευσης των Byte): big endian και little endian. Ο κάθε επεξεργαστής σχεδιάζεται για να υποστηρίζει τον έναν ή τον άλλο τρόπο αποθήκευσης και δεν αλλάζει. Υπάρχουν περιπτώσεις που ο επεξεργαστής έχει δυνατότητα να υποστηρίζει και τους 2 τρόπους αποθήκευσης με χρήση επιπρόσθετων κυκλωμάτων, οπότε σε αυτή την περίπτωση ονομάζεται bi-endian, και μπορεί με την κατάλληλη εντολή κώδικα μηχανής να αποφασίζεται κατά την εκτέλεση ενός προγράμματος ποιον τρόπο θα χρησιμοποιήσει, από τους ανωτέρω δυο. Η endianness αφορά την αρχιτεκτονική του επεξεργαστή και όχι της μνήμης, οπότε όλες οι μνήμες μπορούν να χρησιμοποιηθούν είτε με big είτε με little endian.

Η endianness αφορά μόνο την αποθήκευση μιας λέξης που αποτελείται από πολλά Byte. Ένα μόνο Byte θα αποθηκευτεί με τον ίδιο τρόπο είτε υπάρχει little είτε big endian. Σε περίπτωση όμως που η λέξη προς αποθήκευση αποτελείται από πολλά Byte, τότε η endianness καθορίζει τι θα αποθηκευτεί και πού. Ένας big endian επεξεργαστής αποθηκεύει το σημαντικότερο Byte (Most Significant Byte, MSB) στη μικρότερη διεύθυνση και συνεχίζει να αποθηκεύει διαδοχικά τα επόμενα Byte, ως έως το χαμηλότερης σημαντικότητας Byte (Least Significant Byte, LSB), ενώ ένας little endian αρχιτεκτονικής επεξεργαστής αποθηκεύει το LSB στη μικρότερη διεύθυνση μνήμης και συνεχίζει διαδοχικά την αποθήκευση έως το MSB.

Το Σχήμα 7.7 παρουσιάζει ένα παράδειγμα αποθήκευσης μιας λέξης που αποτελείται από 1 Byte (η λέξη είναι DEh), μιας λέξης που αποτελείται από 2 Byte (η λέξη είναι ABDEh), και μιας λέξης που αποτελείται από 4 Byte (η λέξη είναι 3AF1ABDEh). Όταν η λέξη είναι 1 Byte μόνο, τότε θα αποθηκευτεί στη συγκεκριμένη διεύθυνση (π.χ. FFFFh) με τον ίδιο τρόπο και στις δυο endianness. Αν η λέξη είναι ABDEh, τότε το μεγαλύτερης σημαντικότητας Byte είναι το ABh, και το χαμηλότερης σημαντικότητας Byte είναι το DEh. Στη little endian θα αποθηκευτεί

στη χαμηλότερη διεύθυνση πρώτα το DEh, και στην επόμενη διεύθυνση το ABh, ενώ στη big endian θα αποθηκευτεί στη χαμηλότερη διεύθυνση πρώτα το ABh και στην επόμενη διεύθυνση το DEh.

Ομοίως, αν η λέξη είναι η 3AF1ABDEh, με MSB 3Ah και LSB DEh, και θέλουμε να το αποθηκεύσουμε στη διεύθυνση μνήμης addr, τότε στη little endian θα αποθηκευτεί πρώτα το LSB, δηλαδή το DEh, στην επόμενη διεύθυνση addr+1 το επόμενο Byte, δηλαδή το ABh, στην επόμενη διεύθυνση addr+2 το F1h και στην τελευταία addr+3 το 3Ah. Στη big endian η σειρά αποθήκευσης των Byte θα είναι αντίστροφη. Πρώτα το υψηλής σημαντικότητας Byte, δηλαδή το 3Ah, και στη συνέχεια τα υπόλοιπα με φθίνουσα πορεία, δηλαδή F1h, ABh και τέλος το DEh.



Σχήμα 7.7: Αρχιτεκτονική Μνήμης Big endian και Little endian.

Με τη little-endian αρχιτεκτονική, το λιγότερο σημαντικό στοιχείο μεταφέρεται πέρα από το λιγότερο σημαντικό μέρος των στοιχείων και αποθηκεύεται στη λιγότερη σημαντική θέση μνήμης. Για έναν προγραμματιστή, είναι ευκολότερο να γίνει κατανοητή η πορεία στοιχείων. Η δυσχέρεια της little endian αρχιτεκτονικής, είναι ότι το στοιχείο εμφανίζεται “προς τα πίσω” στη μνήμη του υπολογιστή και έτσι υπάρχει ένα τμήμα θέσεων που πρέπει να διαβαστεί για να καθοριστεί το μέγεθος. Η αποθήκευση της λέξης 3AF1ABDEh στη μνήμη οδηγεί στην αποθήκευση DEABF13Ah στη μνήμη.

Σημειώστε ότι ένας επεξεργαστής με little endian αρχιτεκτονική θα διαβάσει αυτό το στοιχείο ορθά, αφού υποστηρίζει αυτόν τον τρόπο αποθήκευσης, ασχέτως του πως αποθηκεύονται τα δεδομένα στη μνήμη. Εναλλακτικά, ένας επεξεργαστής με big

endian αρχιτετονική που αποθηκεύει την τιμή 3AF1ABDEh στη μνήμη καταλήγει να αποθηκεύσει τελικά την ίδια τιμή 3AF1ABDEh μέσα στο τσιπ μνήμης. Αυτό φαίνεται (για έναν προγραμματιστή) να έχει περισσότερο νόημα. Κανένα σχέδιο δεν έχει περισσότερο πλεονέκτημα από το άλλο, η λειτουργία τους είναι ίδια, είναι δύο διαφορετικοί τρόποι με το ίδιο αποτέλεσμα. Όταν κάνετε υψηλού επιπέδου προγραμματισμό σε ένα σύστημα, το “endian-ness” έχει λίγη διαφορά.

Ο μόνος χρόνος που εκτίθεστε πραγματικά σε κάτι τέτοιο, είναι όταν εξετάζετε τα στοιχεία πολλαπλών Byte άμεσα στη μνήμη και καλείστε να τα ερμηνεύσετε. Εντούτοις, όταν αναπτύσσετε και διορθώνετε το υλικό χαμηλού επιπέδου σταθερού λογισμικού, θα τα συναντάτε πολύ συχνά και έτσι μια κατανόηση big endian και little endian αρχιτεκτονικών στο σχεδιαστή των ενσωματωμένων συστημάτων είναι απαραίτητη.

Η endianess εμφανίζεται οπουδήποτε πρέπει να αποθηκευτεί ή να μεταφερθεί μια λέξη που αποτελείται από πολλαπλά Byte. Εκτός από την αποθήκευση λέξεων στη μνήμη, εμφανίζεται και στις επικοινωνίες δικτύου, δηλαδή κατά την αποστολή και λήψη δεδομένων μέσω δικτυακών πρωτοκόλλων, όπως το TCP/IP. Αυτό οφείλεται στο γεγονός, ότι μπορεί ο αποστολέας και ο παραλήπτης να έχουν διαφορετικά endianess και έτσι να ερμηνεύσουν διαφορετικά τα Byte. Για παράδειγμα, αν ένας υπολογιστής λάβει τη λέξη 3AF1ABDEh και είναι big endian θα ερμηνεύσει τα δεδομένα ως τον 32bit αριθμό 3AF1ABDEh, ενώ αν είναι little endian θα ερμηνεύσει τα δεδομένα ως DEABF13Ah.

Για αυτό το λόγο, έχουν δημιουργηθεί ειδικές συναρτήσεις που πρέπει να καλούνται πριν την αποστολή και μετά τη λήψη των δεδομένων, ώστε να μετατρέπουν τα δεδομένα σε σειρά αποθήκευσης δικτύου (network byte order, NBO), να στέλνονται, και στη λήψη να μετατρέπονται από τη NBO στην endianess του υπολογιστή. Ένα ζεύγος τέτοιων συναρτήσεων είναι οι htons()/ntohs() που υπάρχει στο πρότυπο IEEE Std 1003.1-2001 (‘POSIX.1’). Το πρόβλημα με το endianess εμφανίζεται και στην αποθήκευση δεδομένων σε αρχεία που μοιράζονται υπολογιστές με διαφορετικό endianess, και η συνηθισμένη λύση είναι η χρήση των συναρτήσεων htons()/ntohs(), ακόμη και αν δεν υπάρχουν δικτυακές επικοινωνίες.

Παραδείγματα επεξεργαστών που υποστηρίζουν big endian είναι οι Sun SPARC, Motorola 68K, η οικογένεια των PowerPC ή παλαιότερα οι IBM S/360, ενώ παραδείγματα επεξεργαστών αρχιτεκτονικής little endian είναι οι επεξεργαστές που υποστηρίζουν την οικογένεια x86, IA32 ή AMD64 εταιριών Intel, AMD και άλλων ή παλαιότερα οι DEC Vax. Η endianess είναι πολύ σημαντική στα ενσωματωμένα συστήματα, αφού σε αντίθεση με τους επιτραπέζιους υπολογιστές ή διακομιστές που όλα είναι συμβατά με x86 (little endian), υπάρχουν πολλές διαφορετικές αρχιτεκτονικές ή σχεδιασμοί νέων αρχιτεκτονικών και έτσι απαιτείται αυτή η γνώση σε ένα μηχανικό που ασχολείται με αυτά.

Η little endian μπορεί να διαβάσει την ίδια λέξη μερικές φορές σε διάφορα bit width, π.χ. το 4A 00 00 00 είτε διαβαστεί ως 8bit (4A), είτε ως 16 bit (004A) είναι το ίδιο (χρησιμοποιείται στους compilers). Από την άλλη, η big endian βοηθάει στο να

βρεθεί πόσος μεγάλο είναι ο αριθμός διαβάζοντας μόνο την πρώτη θέση μνήμης (που φέρει το MSB). Τέλος η little-endian απλοποιεί το hardware σε πράξεις πολλαπλών Byte (αφού απαιτείται μια μόνο εντολή αύξησης inc για να διαβαστεί το επόμενο Byte).

Η ορολογία Endianess, bin-endian ή little-endian προέρχεται από το λογοτεχνικό βιβλίο 'Τα ταξίδια του Γκιούλιβερ'. Μέσα στο βιβλίο περιγράφεται η διαμάχη δύο μυθικών λαών για το πως μπορεί κάποιος να σπάσει ένα βρασμένο αυγό, είτε κτυπώντας το στην μυτερή άκρη, είτε στην μέση (δηλαδή μεγάλη/big ή μικρή/little άκρη/endianess του αυγού). Στην διαμάχη αυτή (μέσα στο λογοτεχνικό έργο), όσοι υποστήριζαν την μία ή την άλλη τεχνική χαρακτηρίστηκαν ως 'big-endians' ή 'little-endians' και από εκεί η ορολογία υιοθετήθηκε στην πληροφορική.

## 7.8 Κρυφή Μνήμη

Κάθε πρόσβαση στην εξωτερική μνήμη επιφέρει μια μεγαλύτερη ενεργειακή και χρονική επιβάρυνση, παρά αν θα ήταν πρόσβαση σε μνήμη εντός ολοκληρωμένου κυκλώματος. Επίσης, έχει παρατηρηθεί ότι τα περισσότερα προγράμματα παρουσιάζουν μια χρονική και χωρική τοπικότητα, που μπορεί να την εκμεταλλευτεί κάποιος προς όφελος του. Η χρονική τοπικότητα στις προσβάσεις σημαίνει ότι αν ο επεξεργαστής έχει προσπελάσει μια διεύθυνση μνήμης, τότε σε σύντομο χρονικό διάστημα υπάρχει μεγάλη πιθανότητα να απαιτηθεί πάλι η προσπέλαση της ίδιας διεύθυνσης μνήμης. Η χωρική τοπικότητα στις προσβάσεις σημαίνει ότι αν ο επεξεργαστής έχει προσπελάσει μια διεύθυνση μνήμης, τότε σε σύντομο χρονικό διάστημα υπάρχει μεγάλη πιθανότητα να απαιτηθεί η προσπέλαση των γειτονικών διευθύνσεων της μνήμης.

Αν λοιπόν αντιγραφούν τα δεδομένα της εξωτερικής μνήμης (και οι γειτονικές διευθύνσεις) κατά την πρόσβαση σε κάποια εσωτερική μνήμη, τότε θα επιτευχθεί μεγάλο ενεργειακό και χρονικό κέρδος. Αν μάλιστα η μνήμη αυτή λειτουργεί κρυφά (χωρίς να φαίνεται ή να παρεμβαίνει στο πρόγραμμα και χωρίς να απαιτείται ειδική λειτουργία για να χρησιμοποιηθεί), τότε τα υπάρχοντα προγράμματα θα επωφεληθούν άμεσα. Έτσι λοιπόν, δημιουργήθηκε μια τέτοια μνήμη και ονομάζεται 'κρυφή μνήμη' (cache memory). Οι σημερινοί επεξεργαστές δεν έχουν μόνο μια τέτοια μνήμη, αλλά μια ιεραρχία από τέτοιες μνήμες, για λόγους που θα εξηγηθούν στη συνέχεια.

Η κρυφή μνήμη λειτουργεί ως εξής: Όταν θέλουμε ο επεξεργαστής να προσπελάσει (να διαβάσει ή εγγράψει) μια διεύθυνση κύριας μνήμης, πρώτα ελέγχουμε για την ύπαρξη αντιγράφου των δεδομένων της τοποθεσίας αυτής στην κρυφή μνήμη. Αν έχει γίνει αντιγραφή στην κρυφή μνήμη, τότε υπάρχει επιτυχία κρυφής μνήμης (cache hit) και μπορεί να ολοκληρωθεί άμεσα η πρόσβαση. Αν όμως δεν έχει γίνει αντιγραφή, τότε υπάρχει αστοχία κρυφής μνήμης (cache miss) και πρέπει πρώτα να διαβάσουμε τη διεύθυνση (και ίσως και κάποιες γειτονικές διευθύνσεις) και να μεταφέρουμε τα δεδομένα μέσα στην κρυφή μνήμη. Μόλις ολοκληρωθεί η περιγραφή, ο επεξεργαστής μπορεί να διαβάσει τα δεδομένα από την κρυφή μνήμη. Σημειώστε ότι αναφερόμαστε σε προσπελάσεις ανάγνωσης και όχι εγγραφής (δεν υπάρχει κέρδος ή νόημα να επαναχρησιμοποιούνται δεδομένα που γράφονται συνεχώς).

Η περιγραφή αυτή της λειτουργίας της κρυφής μνήμης οδηγεί σε διάφορες επιλογές σχεδιασμού κρυφής μνήμης: χαρτογράφηση κρυφής μνήμης, πολιτική αντικατάστασης της κρυφής μνήμης και τεχνικές εγγραφής στην κρυφή μνήμη. Αυτές οι επιλογές σχεδιασμού μπορεί να έχουν πολλαπλές επιδράσεις στο κόστος του συστήματος, την απόδοσή του καθώς και στην κατανάλωση ενέργειας και για αυτό θα έπρεπε να αξιολογηθούν προσεκτικά για μια δεδομένη εφαρμογή σε ένα ενσωματωμένο σύστημα.

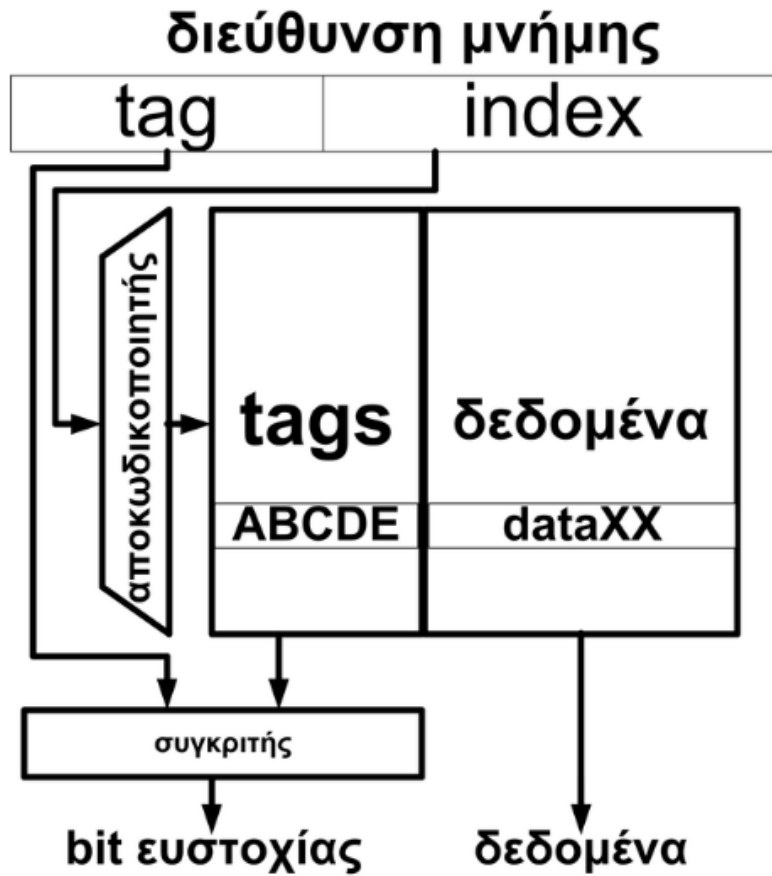
### 7.8.1 Τεχνικές χαρτογράφησης κρυφής μνήμης

Η χαρτογράφηση κρυφής μνήμης είναι μια μέθοδος για την εκχώρηση των διευθύνσεων της κύριας μνήμης στον πολύ μικρότερο αριθμό διαθέσιμων διευθύνσεων της κρυφής μνήμης, και για να προσδιοριστεί αν το περιεχόμενο της συγκεκριμένης διεύθυνσης κύριας μνήμης είναι στην κρυφή μνήμη. Μια κρυφή μνήμη μπορεί να έχει π.χ. 128 διευθύνσεις οι οποίες θα πρέπει να αντιστοιχηθούν σε χιλιάδες ή εκατομμύρια διευθύνσεις της εξωτερικής μνήμης RAM. Η χαρτογράφηση της κρυφής μνήμης μπορεί να επιτευχθεί με τη χρήση μιας από τις ακόλουθες τρεις βασικές τεχνικές:

- Άμεση χαρτογράφηση ή απευθείας απεικόνιση (direct mapped): Σε αυτή τη τεχνική, η διεύθυνση της κύριας μνήμης διαιρείται σε δύο περιοχές, τα bit ευρετηρίου (index) και τις ετικέτας (tag). Είναι η απλούστερη οργάνωση και για αυτό έχει και τη χαμηλότερη κατανάλωση ενέργειας λειτουργίας (αλλά και τη μικρότερη απόδοση). Το ευρετήριο αντιπροσωπεύει την διεύθυνση της κρυφής μνήμης και επομένως ο αριθμός των bits αυτού καθορίζεται από το μέγεθος της κρυφής μνήμης, με τη σχέση: μέγεθος ευρετηρίου =  $\log_2$ (μέγεθος κρυφής μνήμης). Για παράδειγμα, αν το μέγεθος της κρυφής μνήμης είναι 128 (από 0 έως 127) γραμμές, τότε ισχύει  $2^7 = 128$  και άρα απαιτούνται 7 bit. Τα υπόλοιπα bit της διεύθυνσης μνήμης αποτελούν την ετικέτα, αφού όπως γίνεται κατανοητό πολλές διαφορετικές διευθύνσεις κύριας μνήμης θα χαρτογραφηθούν στην ίδια γραμμή της κρυφής μνήμης.

Έτσι, όταν αποθηκεύουμε το περιεχόμενο μιας διεύθυνσης κύριας μνήμης στην κρυφή μνήμη αποθηκεύουμε επίσης και την αντίστοιχη ετικέτα. Για να καθορίσουμε αν μια επιθυμητή διεύθυνση κύριας μνήμης είναι στην κρυφή μνήμη, πηγαίνουμε στη διεύθυνση της κρυφής μνήμης, που υποδεικνύεται από τον δείκτη, και έπειτα συγκρίνουμε την ετικέτα αυτή με την επιθυμητή ετικέτα.

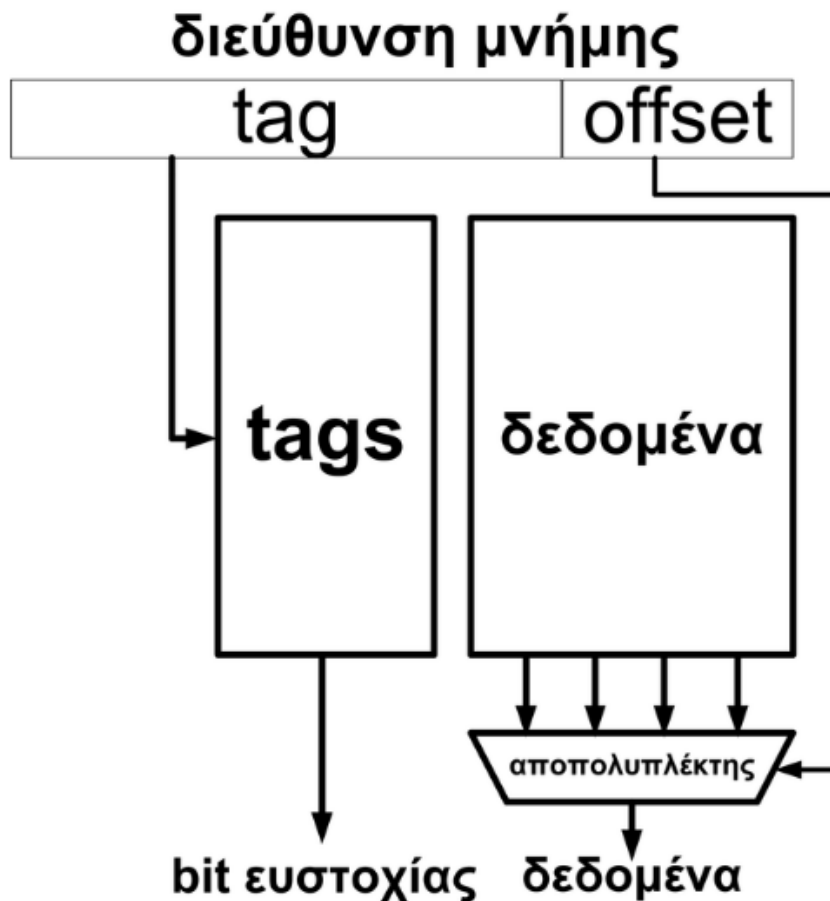
- Η κρυφή μνήμη απευθείας απεικόνισης για μέγεθος γραμμής 1 byte, φαίνεται στην Εικόνα . Συγκεκριμένα, φαίνεται ο διαχωρισμός της διεύθυνσης σε tag και index. Τα bit του index καθορίζουν τη γραμμή της κρυφής μνήμης που θα συγκριθεί το tag. Αν σε αυτή τη γραμμή το tag είναι το ίδιο με αυτό που προκύπτει από την αρχική διεύθυνση, τότε η κρυφή μνήμη φέρει έγκυρα δεδομένα για την τρέχουσα πρόσβαση και τα προωθεί στον επεξεργαστή.



Σχήμα 7.8.1 (Εικόνα 1): Οργάνωση της κρυφής μνήμης άμεσης απεικόνισης όπου κάθε γραμμή κρυφής μνήμης φέρει 1 Byte.

- Χαρτογράφηση (ή απεικόνιση) πλήρους συσχέτισης (fully associative): Σε αυτή τη τεχνική, κάθε διεύθυνση κρυφής μνήμης περιέχει όχι μόνο το περιεχόμενο της διεύθυνσης κύριας μνήμης αλλά και ολόκληρη τη διεύθυνση αυτής. Για να προσδιορίσουμε αν μια επιθυμητή διεύθυνση κύριας μνήμης είναι στην κρυφή μνήμη, συγκρίνουμε ταυτόχρονα (συσχετιζόμενα) όλες τις αποθηκευμένες στην κρυφή μνήμη διευθύνσεις με την επιθυμητή διεύθυνση. Σε αυτήν την περίπτωση οποιαδήποτε διεύθυνση της κρυφής μνήμης μπορεί να τοποθετηθεί σε οποιαδήποτε γραμμή της κρυφής μνήμης. Έχει πάρα πολύ αυξημένες ανάγκες σε υλικό (απαιτούνται πάρα πολλοί συγκριτές), μεγάλη κατανάλωση ενέργειας (λόγω των πολλών συγκρίσεων), αλλά πολύ καλό ποσοστό ευστοχίας. Εντούτοις, λόγω του μεγάλου κόστους δε χρησιμοποιείται. Η Εικόνα παρουσιάζει την οργάνωση μιας τέτοιας κρυφής μνήμης.

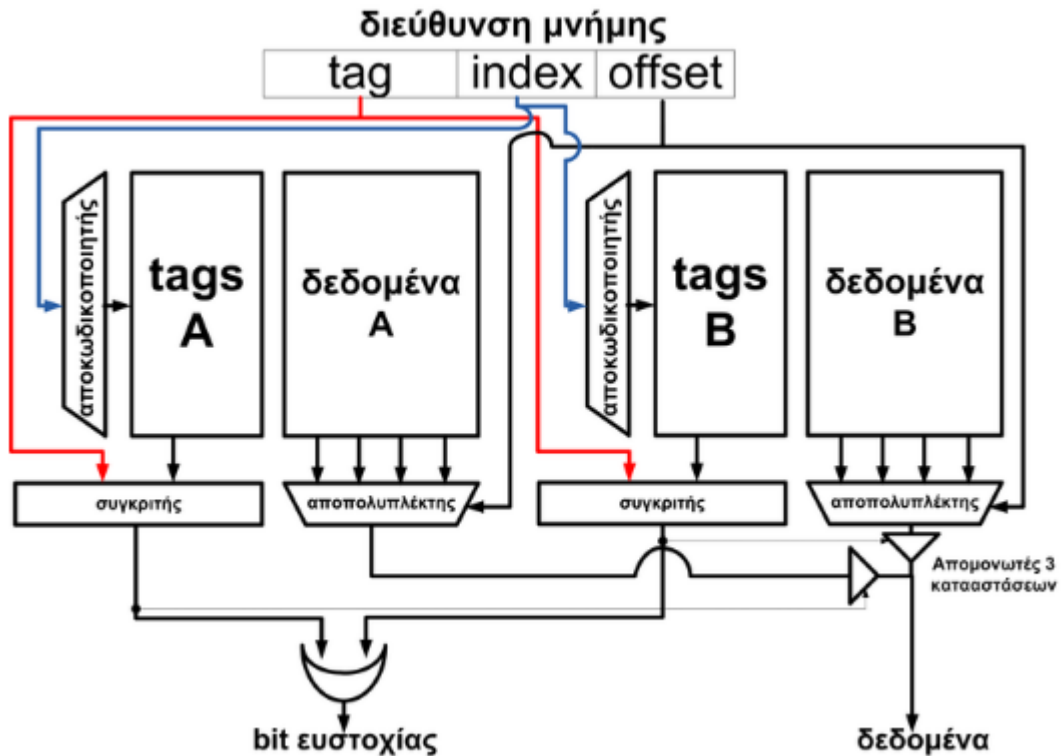




Σχήμα 7.8.1 (Εικόνα 2) : Οργάνωση της κρυφής μνήμης πλήρους συσχέτισης.

- Χαρτογράφηση (ή απεικόνιση) με συσχέτιση σε ομάδες (set associative): Η τεχνική αυτή αποτελεί έναν συμβιβασμό μεταξύ των δυο παραπάνω τεχνικών. Όπως στην άμεση χαρτογράφηση, έτσι και εδώ, το `index` χαρτογραφεί κάθε διεύθυνση κύριας μνήμης σε μια διεύθυνση κρυφής μνήμης, αλλά τώρα κάθε διεύθυνση κρυφής μνήμης περιλαμβάνει το περιεχόμενο και τις ετικέτες δύο ή περισσότερων τοποθεσιών μνήμης, που ονομάζονται ομάδες (sets). Για να προσδιορίσουμε αν μια επιθυμητή διεύθυνση κύριας μνήμης είναι στην κρυφή μνήμη, πηγαίνουμε στη διεύθυνση της κρυφής μνήμης που υποδεικνύεται από το `index`, και έπειτα ταυτόχρονα (συσχετιζόμενα) συγκρίνουμε όλες τις ετικέτες στην τοποθεσία με την επιθυμητή ετικέτα.

Η κρυφή μνήμη με ένα σύνολο  $N$  ονομάζεται  $N$ -δρόμων ομαδικά συσχετιζόμενη κρυφή μνήμη. Τυπικές δομές κρυφής μνήμης είναι 2-δρόμων, 4-δρόμων και 8-δρόμων. Το Σχήμα παρουσιάζει μια δομή κρυφής μνήμης δυο δρόμων. Όπως και στην κρυφή μνήμη απευθείας απεικόνισης, η διεύθυνση της εξωτερικής μνήμης χωρίζεται σε `tag|index|offset`. Το `index` σε αυτή την περίπτωση, δείχνει τη γραμμή και στα 2 σετ. Μόλις βρεθεί η γραμμή συγκρίνεται το αρχικό `tag` ταυτόχρονα και με τα `tag` από τα 2 σύνολα. Αν βρεθεί σε ένα από αυτά τότε προωθούνται τα δεδομένα προς τον επεξεργαστή από το αντίστοιχο σύνολο. Αν δε βρεθεί σε κανένα, τότε η κρυφή μνήμη θα μεταφέρει από την εξωτερική μνήμη τα δεδομένα και θα τα αποθηκεύσει σε ένα από τα δυο σύνολα. Η επιλογή θα γίνει με κάποιον αλγόριθμο, όπως θα αναφερθεί στη συνέχεια.



Σχήμα 7.8.1 (Εικόνα 3) : Οργάνωση της κρυφής μνήμης συσχέτισης κατά ομάδων δυο δρόμων.

Αυξάνοντας το επίπεδο της πολυπλοκότητας, η συσχέτιση κατά ομάδες στοχεύει να μειωθούν τα προβλήματα αντιστοίχισης που υπάρχουν στην απευθείας απεικόνιση μνήμη, δυνατότητα σε ένα συγκεκριμένο αντικείμενο της μνήμης να αποθηκευτεί σε περισσότερες από μια τοποθεσίες της κρυφής μνήμης. Μία διεύθυνση εξωτερικής μνήμης που παρουσιάζεται στην κρυφή μνήμη μπορεί να έχει δεδομένα σε οποιοδήποτε σύνολο. Σε περίπτωση που υπήρχε διαμάχη αντιστοίχισης (όπως συνέβαινε στην κρυφή μνήμη απευθείας απεικόνισης) από διευθύνσεις που απεικονίζονταν στην ίδια γραμμή, τώρα αυτή η διαμάχη εξαλείφεται αφού στην ίδια γραμμή υπάρχουν πολλαπλές επιλογές (σε άλλα σετ).

Σε σύγκριση με το προηγούμενο παράδειγμα, μια διεύθυνση των 32 bit θα μπορούσε να έχει 4 bit για το offset και 8 bit για το index (όχι 9 όπως στην απευθείας απεικόνιση), επειδή το ίδιο μέγεθος μνήμης διαιρείται σε 2 σετ, αφήνοντας 20 bit για το tag. Τα tag bit απαιτούν  $(20 \text{ bit} * 2^8 \text{ γραμμές} = 5120 \text{ bit})$  για κάθε set, και άρα συνολικά 10240 bit ή 1280 Byte, δηλαδή λίγο πάνω από ένα KB (η ίδια μεγέθους μνήμη απευθείας απεικόνισης απαιτούσε 1215 Byte). Παρατηρούμε λοιπόν ότι όσο αυξάνεται ο βαθμός συσχέτισης, αυξάνονται και οι αποθηκευτικές ανάγκες της κρυφής μνήμης. Στην ακραία περίπτωση της πλήρους συσχέτισης, το μέγεθος για τα tags είναι  $28 \text{ bit} * 2^9 \text{ γραμμές} = 14336$  ή 1792 Byte. Οι κρυφές μνήμες μερικής συσχέτισης, έχουν καλά ποσοστά ευστοχίας και για αυτό χρησιμοποιούνται τόσο σε ΕΣ όσο και στα υπόλοιπα υπολογιστικά συστήματα.

Ως προς το χρόνο, όσο πιο μεγάλη είναι η συσχέτιση, τόσο μεγαλύτερη καθυστέρηση έχει. Η αύξηση στο χρόνο γίνεται εξαιτίας της ανάγκης να πολλαπλασιαστούν τα δεδομένα από τα δύο σετ. Όταν ένα νέο δεδομένο τοποθετείται

στην κρυφή μνήμη, πρέπει να ληφθεί μια απόφαση, σε ποιο από τα δύο σετ θα τοποθετηθεί. Η πολιτική αντικατάστασης θα περιγραφεί στην επόμενη ενότητα.

Η κρυφή μνήμη κρατάει μια εγγραφή για να ξέρει ποιο ζευγάρι ποιας τοποθεσίας είχε τοποθετηθεί τελευταία ώστε να τοποθετήσει τα νέα δεδομένα σε μια άλλη τοποθεσία. Η συσχέτιση κατά ομάδες απαιτεί τουλάχιστον δύο επιπλέον δρόμους για κάθε βαθμό συσχέτισης, αλλά πρακτικά τα πλεονεκτήματα της συσχέτισης μετά από 4 δρόμους είναι λιγότερα και δεν δίνουν βάση για περαιτέρω αύξηση της συσχέτισης.

Στην άλλη ειδική συσχέτιση, είναι πιθανόν να σχεδιαστεί μια κρυφή μνήμη συσχέτισης στην τεχνολογία VLSI. Θα ήταν βέβαια προτιμότερο να διαιρεθεί η κρυφή μνήμη άμεσης απεικόνισης σε ακόμη μικρότερα συστατικά μέρη. Τα αποθηκευμένα δεδομένα σχεδιάζονται διαφορετικά χρησιμοποιώντας την κρυφή μνήμη διευθυνσιοδοτούμενη από δεδομένα (Content Addressable Memory - CAM). Ένα δομοστοιχείο της CAM είναι παρόμοιο με αυτό της RAM με έναν ενσωματωμένο συγκριτή, έτσι ώστε η βάση δεδομένων αποθηκευμένων στοιχείων της CAM να μπορεί να εκτελέσει μια παράλληλη έρευνα για να τοποθετηθεί μια διεύθυνση σε κάθε τοποθεσία (πλήρους συσχέτισης).

## 7.8.2 Πολιτική αντικατάστασης κρυφής μνήμης

Η πολιτική αντικατάστασης της κρυφής μνήμης είναι η τεχνική για την επιλογή της γραμμής της κρυφής μνήμης που θα αντικατασταθεί όταν μια συσχετιστική κρυφή μνήμη είναι πλήρης. Ασφαλώς η πολιτική αυτή δε χρειάζεται σε μια άμεσης απεικόνισης κρυφής μνήμης, αφού υπάρχει αντιστοίχιση πάντα 1 προς 1 (1 διεύθυνση εξωτερικής μνήμης πάντα αντιστοιχεί σε μια γραμμή της κρυφής μνήμης). Υπάρχουν τρεις συνηθισμένες πολιτικές αντικατάστασης. Μια πολιτική τυχαίας αντικατάστασης (random) επιλέγει τυχαία το σετ που θα αντικατασταθεί. Όσο απλή κι αν είναι στην υλοποίηση της, αυτή η πολιτική δεν κάνει τίποτα για να αποτρέψει την αντικατάσταση κάποιου στοιχείου που είναι πιθανό να χρησιμοποιηθεί σύντομα ξανά. Η πολιτική αντικατάστασης του λιγότερο χρησιμοποιούμενου στοιχείου (least recently used) αντικαθιστά το στοιχείο που δεν έχει προσπελαστεί για το μεγαλύτερο χρονικό διάστημα, υποθέτοντας βέβαια ότι αυτό σημαίνει δεν είναι πιθανό να προσπελαστεί στο άμεσο μέλλον.

Η πολιτική αυτή εξασφαλίζει ένα σημαντικό λόγο ευστοχίας έναντι αστοχίας, αλλά απαιτεί ακριβό υλικό για να παρακολουθεί τους χρόνους προσπέλασης των δομικών στοιχείων. Η πολιτική κατά την οποία αυτό που είχε έρθει πρώτο, θα αποχωρήσει και πρώτο (FIFO), χρησιμοποιεί μια ουρά μεγέθους  $N$  και σπρώχνει κάθε διεύθυνση στοιχείου στην ουρά, όταν η διεύθυνση αυτή προσπελαύνεται, και έπειτα επιλέγει από την κορυφή το στοιχείο που θα αντικαταστήσει μετακινώντας τα υπόλοιπα στοιχεία της ουράς.

## 7.9 Ο Επεξεργαστής σε ένα Ενσωματωμένο Σύστημα

Η καρδιά σε κάθε υπολογιστικό σύστημα είναι το δομοστοιχείο που κάνει τους υπολογισμούς. Στους υπολογιστές γενικού σκοπού δεν υπάρχει ευελιξία ως προς αυτό: υπάρχουν ένας ή περισσότεροι επεξεργαστές γενικής αρχιτεκτονικής 64 bit συμβατής με Intel x86-64, που σημαίνει ότι μπορούν να αντεπεξεχθούν σε

οποιαδήποτε φόρτο εργασίας τους ζητηθεί. Αντιθέτως, στα Ενσωματωμένα Συστήματα υπάρχει μια μεγάλη ποικιλία υλοποιήσεων, κάποιες από τις οποίες είναι αντιδιαμέτρου αντίθετες μεταξύ τους.

Για παράδειγμα τα Ενσωματωμένα Συστήματα μπορούν να χρησιμοποιήσουν έναν γενικής χρήσης επεξεργαστή (π.χ. Intel ATOM), ειδικούς επεξεργαστές ψηφιακής επεξεργασίας σημάτων (π.χ. Texas Instrument TIC6201), μικροεπεξεργαστές 8 bit (π.χ. AVR ATMEGA328P), σύνθετους επεξεργαστές 32 bit (π.χ. ARM Cortex A15), απλούς επεξεργαστές 32 bit (MIPS R3000), επεξεργαστές μαλακού πυρήνα σε επαναδιαμορφώσιμη λογική FPGA (soft-cores, π.χ. Altera NIOS II) ή ακόμη και εξειδικευμένα κυκλώματα υλικού (ASIC), τα οποία δεν έχουν καν επεξεργαστή, αλλά υλοποιούν μια συγκεκριμένη λειτουργία χρησιμοποιώντας μια μηχανή πεπερασμένων καταστάσεων (finite state machine).

Μάλιστα, οι επεξεργαστές δεν είναι συμβατοί μεταξύ τους, όπως στους υπολογιστές γενικού σκοπού ή στους φορητούς υπολογιστές ή στους διακομιστές (που όλοι είναι συμβατοί με τις προδιαγραφές της αρχιτεκτονικής x86-64, και έτσι ένα πρόγραμμα μπορεί να εκτελεστεί σε όλους τους συμβατούς επεξεργαστές). Το οικοσύστημα των Ενσωματωμένων Συστημάτων φαίνεται τόσο χαοτικό σε κάποιον που μόλις τώρα το γνωρίζει, και ο σχεδιασμός ενός Ενσωματωμένου Συστήματος μπορεί να παρομοιαστεί με ένα δαιδαλώδες λαβύρινθο με πολλά μονοπάτια, που κάθε μονοπάτι οδηγεί σε άλλες επιλογές.

## **7.10 Περιγραφή της αρχιτεκτονικής-στόχου για ένα Ενσωματωμένο Σύστημα**

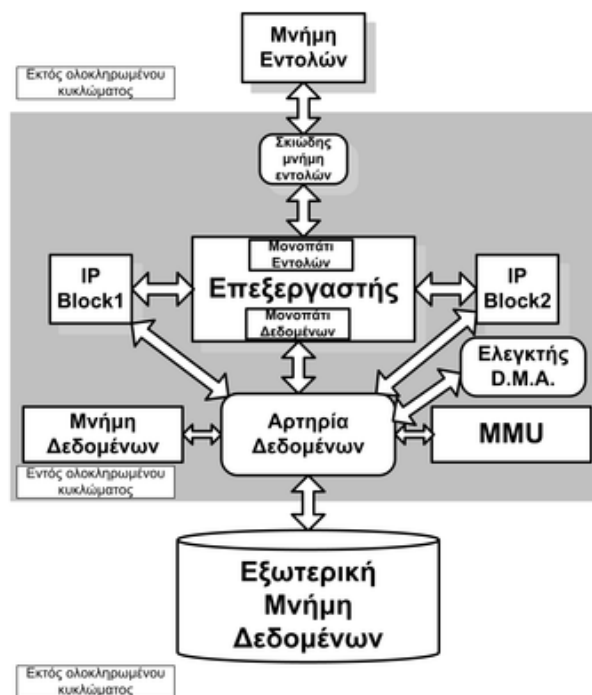
Κάθε μεθοδολογία σχεδιασμού βασίζεται σε κάποια αρχιτεκτονική. Η επιλογή της αρχιτεκτονικής πρέπει να γίνει προσεκτικά, διαφορετικά μπορεί να μην είναι πρακτικά υλοποιήσιμη. Η αρχιτεκτονική θα πρέπει να έχει όσο το δυνατόν χαμηλότερο κόστος, να μπορεί να υλοποιηθεί πραγματικά (δηλαδή να μην είναι η μεθοδολογία μόνο θεωρητική), και να είναι παραμετροποιήσιμη, δηλαδή να παρέχει στο σχεδιαστή ευελιξία στο να τροποποιεί κάποια χαρακτηριστικά που ίσως ταιριάζουν καλύτερα στο πρόβλημά του. Πριν αρχίσουμε την έρευνά μας, προβήκαμε σε μια ανάλυση των αρχιτεκτονικών των ενσωματωμένων συστημάτων διαφόρων κατασκευαστών, σημειώσαμε τα κοινά χαρακτηριστικά που έχουν, τα θετικά και αρνητικά σημεία τους, και αναπτύξαμε τη δικιά μας αρχιτεκτονική.

Έτσι, λοιπόν, η μεθοδολογία σχεδιασμού Ενσωματωμένου Συστήματος που παρουσιάζουμε βασίζεται σε μια γενική αρχιτεκτονική, η οποία συναντάται αρκετά συχνά στα ενσωματωμένα συστήματα. Όπως φαίνεται στην Εικόνα η αρχιτεκτονική αποτελείται από δομοστοιχεία εντός και εκτός ολοκληρωμένου κυκλώματος.

Η εξωτερική μνήμη δεδομένων και η μνήμη εντολών βρίσκονται εκτός ολοκληρωμένου κυκλώματος, ενώ η μνήμη δεδομένων (ή επιπλήνθια μνήμη), η μονάδα διαχείρισης μνήμης (Memory Management Unit, MMU), ο ελεγκτής

απευθείας πρόσβασης στη μνήμη (Direct Memory Access, DMA), η σκιώδης (ή κρυφή) μνήμη εντολών, ο επεξεργαστής και τα εξειδικευμένα στοιχεία για την εφαρμογή (IP-blocks) βρίσκονται πάνω στο ολοκληρωμένο κύκλωμα. Τα δομοστοιχεία αυτά θα τα αναπτύξουμε στη συνέχεια, μαζί με το λόγο για τον οποίον έχουν επιλεγεί.

Πρέπει να σημειωθεί βέβαια ότι ο τύπος του συγκεκριμένου δομοστοιχείου, για παράδειγμα αν ο τύπος της μνήμης θα είναι SRAM ή DRAM, δεν είναι ρητά καθορισμένος, αλλά καθορίζεται από το σχεδιαστή, ή από τη διαθεσιμότητα των υλικών.



Σχήμα 7.10: Η μεθοδολογία μας χρησιμοποιεί μια γενική αρχιτεκτονική ενσωματωμένων συστημάτων και δεν περιορίζεται σε συγκεκριμένα δομοστοιχεία.

## 7.11 Βασική Αρχιτεκτονική FPGA

Σε γενικές γραμμές τα FPGA μπορούν να θεωρηθούν ως προγραμματιζόμενες συσκευές που παρέχουν μια μεγάλη πυκνότητα ολοκλήρωσης και ευελιξία. Η βασική αρχιτεκτονική των FPGA αποτελείται από τρία συστατικά, ανεξαρτήτως της εταιρίας που τα κατασκευάζει:

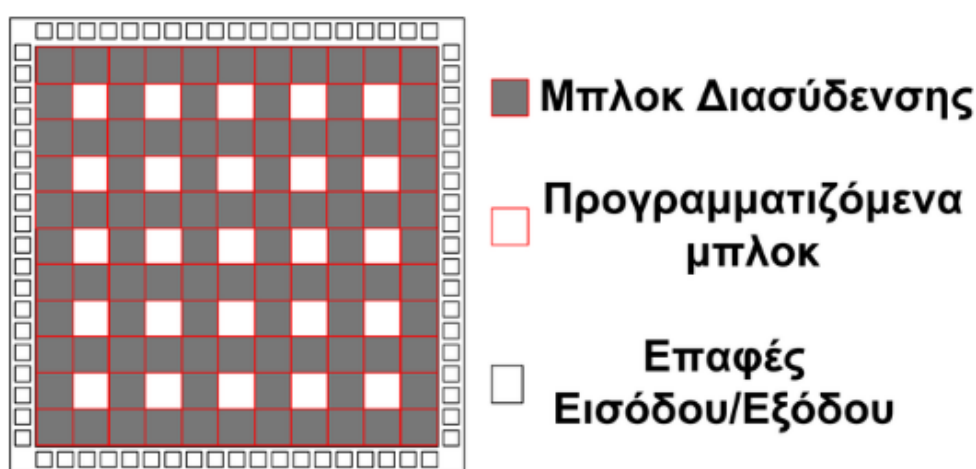
- Τη λογική μονάδα (logic block),
- τη διασύνδεση (routing) και
- τη μονάδα E/E (I/O block).

Τα FPGA αποτελούνται από ένα δισδιάστατο πλέγμα προγραμματιζόμενων λογικών μονάδων που μπορούν να διασυνδεθούν μεταξύ τους καθώς επίσης και με τις προγραμματιζόμενες μονάδες E/E μέσω κάποιου είδους προγραμματιζόμενης

αρχιτεκτονικής διασύνδεσης. Τα τελευταία χρόνια γίνονται προσπάθειες ανάπτυξης τρισδιάστατων ολοκληρωμένων κυκλωμάτων στα FPGA, τα οποία αποτελούνται από την κατακόρυφη συσσώρευση πολλαπλών IC που συνδέονται μέσω κατακόρυφων διαμπερών οπών. Το πρώτο τρισδιάστατο FPGA, που διατέθηκε για πώληση στους καταναλωτές υψηλών απαιτήσεων, ήταν το Virtex-7 2000T που παρουσιάστηκε το 2011. Είχε 6.8 δισεκατομμύρια τρανζίστορ με συνέπεια να διαθέτει πάνω από 2 εκατομμύρια λογικά κελιά ή το ισοδύναμο 20 εκατομμυρίων λογικών πυλών. Για την κατασκευή χρησιμοποιήθηκαν 4 ολοκληρωμένα κυκλώματα FPGA Virtex, το ένα πάνω από το άλλο, με πάνω από 10.000 κατακόρυφες οπές διασύνδεσης.

Εκτός από το Virtex-7 2000T, που είναι ομογενές τρισδιάστατο ολοκληρωμένο κύκλωμα, αφού και τα 4 επίπεδα φέρουν ίδια FPGA, η Xilinx παρουσίασε το 2012 μια ετερογενή αρχιτεκτονική που περιέχει 2 chip FPGA και ένα ολοκληρωμένο κύκλωμα πομποδέκτη 8 καναλιών και 28Gbps, με όνομα H580T. Όπως και τα τρισδιάστατα ολοκληρωμένα κυκλώματα ASIC, έτσι και τα FPGA έχουν πολύ υψηλό κόστος, εξαιτίας της δυσκολίας παραγωγής τους, αφού απαιτείται να ευθυγραμμιστούν χιλιάδες οπές διασύνδεσης σε όλα τα επίπεδα μεγέθους κάποιων χιλιάδων τρανζίστορ. Για αυτό και τα περισσότερα ολοκληρωμένα κυκλώματα είναι δισδιάστατα και αυτό δε φαίνεται να αλλάζει τα επόμενα χρόνια.

Όλα τα FPGA έχουν την ίδια τυπική αρχιτεκτονική, ανεξαρτήτως της εταιρίας που τα κατασκευάζει. Στην Εικόνα φαίνεται μια γενική αρχιτεκτονική FPGA.



Σχήμα 7.11 (Εικόνα 1): Τυπική Αρχιτεκτονική FPGA.

Για να γίνει η υλοποίηση ενός κυκλώματος σε ένα FPGA πρέπει να προγραμματιστεί η κάθε βασική λογική μονάδα του, ώστε να υλοποιεί ένα μικρό μέρος της λογικής, ενώ ταυτόχρονα κάθε μια από τις μονάδες E/E να ενεργεί είτε ως μονάδα εισόδου, είτε ως εξόδου, ανάλογα με τις απαιτήσεις του κυκλώματος. Η προγραμματιζόμενη διασύνδεση ρυθμίζεται για να υλοποιεί τις απαραίτητες συνδέσεις τόσο ανάμεσα στις βασικές λογικές μονάδες, όσο και ανάμεσα στις βασικές λογικές μονάδες και τις μονάδες E/E. Η λειτουργική πολυπλοκότητα των βασικών λογικών μονάδων μπορεί να ποικίλει από μια απλή λογική συνάρτηση 2-εισόδων έως πολύπλοκες αριθμητικές λειτουργίες πολλαπλών bit.

Η επιλογή της κοκκιότητας (granularity) της βασικής λογικής μονάδας εξαρτάται από το πεδίο της κάθε εφαρμογής. Η χρησιμοποιούμενη τεχνολογία προγραμματισμού καθορίζει και τη μέθοδο αποθήκευσης της πληροφορίας προγραμματισμού. Η επιλογή της συγκεκριμένης τεχνολογίας έχει σημαντική επίδραση τόσο στην καταλαμβανόμενη επιφάνεια όσο και στην απόδοση του FPGA. Όπως αναφέρθηκε, τα FPGA είναι προγραμματιζόμενα και επαναδιαμορφούμενα πολλών χρήσεων, δηλαδή μπορούν να επανδιαμορφώνονται συνεχώς. Θα πρέπει λοιπόν να αποθηκεύεται κάπου αυτό το πρόγραμμα, και για αυτό χρησιμοποιούνται διάφορες τεχνολογίες.

Οι κυριότερες και επικρατέστερες τεχνολογίες αποθήκευσης των δεδομένων προγραμματισμού, είναι η Στατική Μνήμη Τυχαίας Προσπέλασης (SRAM), η αντισφάλεια (antifuse) και η μη-πτητική (non-volatile). Η επιλογή μιας εκ των προαναφερθέντων τεχνολογιών βασίζεται κυρίως στο υπολογιστικό περιβάλλον, στο οποίο πρόκειται να χρησιμοποιηθεί το FPGA, και στις δυνατότητες του εκάστοτε ολοκληρωμένου κυκλώματος FPGA. Κάποιες τεχνολογίες επιτρέπουν τη διατήρηση του προγραμματισμού ακόμη και αν σταματήσει η παροχή και επανέλθει το ρεύμα (όπως π.χ. η Flash), ενώ κάποιες άλλες είναι πτητικές και απαιτούν σε κάθε εκκίνηση λειτουργίας τον επαναπρογραμματισμό του FPGA.

Οι προγραμματιζόμενες συσκευές FPGA που διατίθενται στην αγορά διαφέρουν στον τύπο τεχνολογίας προγραμματισμού, στην αρχιτεκτονική της βασικής λογικής μονάδας καθώς και την δομή της αρχιτεκτονικής διασύνδεσης. Στις επόμενες παραγράφους θα περιγράψουμε τα τρία συστατικά της αρχιτεκτονικής FPGA και τις τεχνολογίες προγραμματισμού.

Μια πιθανή υλοποίηση είναι ένα σύστημα βασισμένο σε H/Y με τον επιταχυντή να στεγάζεται σε μια κάρτα που συνδέεται με το PC bus .Η πλακέτα αυτή μπορεί να χρησιμοποιήσει ένα συνηθισμένο τσιπ ή έναν FPGA για να υλοποιήσει τον επιταχυντή. Αυτό το είδος του συστήματος είναι σχετικά ογκώδες και συχνότερα χρησιμοποιείται για ανάπτυξη ή για πολύ χαμηλής έντασης εφαρμογές. Ένας συνηθισμένος τυπωμένος πίνακας κυκλωμάτων, χρησιμοποιεί είτε ένα FPGA, είτε ένα συνηθισμένο ενσωματωμένο κύκλωμα για τον επιταχυντή. Η κάρτα επέκτασης FPGA απαιτεί περισσότερη σχεδιαστική εργασία από ένα σύστημα βασισμένο σε H/Y, αλλά έχει αποτελέσματα σε ένα χαμηλότερου κόστους και αρκετά πιο δυνατό επεξεργαστικά σύστημα.

Μια εναλλακτική και η πιο πρόσφατη υλοποίηση είναι μια πλατφόρμα FPGA που περιλαμβάνει μια ΚΜΕ και μια δομή FPGA σε ένα ενιαίο τσιπ, όπως η σειρά Zynq7000 της Xilinx (Εικόνα 4.28). Αυτά τα τσιπ είναι ακριβότερα από τα συνηθισμένα τσιπ αλλά παρέχουν μια εφαρμογή ενιαίου τσιπ με μία ή περισσότερες CPUs και μια τυπική επαναδιαμορφώσιμη λογική.



Σχήμα 7.11 (Εικόνα 2): Ο επιταχυντής σε ένα Ενσωματωμένο σύστημα, μπορεί να είναι μια κάρτα FPGA που συνδέεται στο διάλυλο του συστήματος. Εδώ φαίνεται η Virtex 6 FPGA κάρτα της Xilinx με διεπαφή PCI Express.



Σχήμα 7.11 (Εικόνα 3) : Στα ενσωματωμένα συστήματα μπορεί να χρησιμοποιηθεί κάποια αρχιτεκτονική FPGA που εμπεριέχει και έναν ή περισσότερους επεξεργαστές.

Εδώ φαίνεται η αναπτυξιακή πλατφόρμα Xilinx Zybo, η οποία περιέχει έναν διπύρνηνο επεξεργαστή ARM Cortex-A9 στα 650 Mhz και επαναδιαμορφώσιμη FPGA με 28000 λογικά κελιά.

Ένας επιταχυντής υλικού είναι ένα συνηθισμένο ολοκληρωμένο κύκλωμα, που υλοποιεί μια εξειδικευμένη λειτουργία, με όσο το δυνατόν μικρότερη κατανάλωση ενέργειας και υψηλές επιδόσεις. Πολλά ενσωματωμένα συστήματα σε τσιπ systems-on-chips (SoC) χρησιμοποιούν επιταχυντές για διάφορες λειτουργίες.



Ο συνδυασμός μιας ΚΜΕ με έναν ή περισσότερους επιταχυντές είναι η απλούστερη μορφή ετερογενούς πλατφόρμας ενσωματωμένων συστημάτων. Η ΚΜΕ καλείται συχνά host (ξενιστής). Η ΚΜΕ επικοινωνεί με τον επιταχυντή μέσω των δεδομένων και των καταχωρητών ελέγχου που βρίσκονται μέσα σε αυτόν. Αυτοί οι καταχωρητές επιτρέπουν στην ΚΜΕ να ελέγξουν τη λειτουργία του επιταχυντή και για να του δώσουν εντολές. Η ΚΜΕ και ο επιταχυντής μπορούν επίσης να επικοινωνήσουν μέσω της κοινής μνήμης. Εάν ο επιταχυντής πρέπει να λειτουργήσει σε έναν μεγάλο όγκο δεδομένων, είναι συνήθως αποδοτικότερο να αφήσει τα στοιχεία στη μνήμη και να έχει τον επιταχυντή να διαβάζει και να γράφει μνήμη άμεσα, παρά να έχει μεταφορά των δεδομένων της ΚΜΕ από τη μνήμη στους καταχωρητές του επιταχυντή και αντίστροφα. Η ΚΜΕ και ο επιταχυντής συγχρονίζουν τις ενέργειές τους. Περισσότερες και γενικότερες πλατφόρμες είναι επίσης δυνατές.

Μπορούμε να χρησιμοποιήσουμε διάφορες CPUs, σε αντίθεση με τον ενιαίο επεξεργαστή που είναι συνδεδεμένος με τους επιταχυντές. Μπορούμε να χρησιμοποιήσουμε πολλαπλούς διαύλους. Επιπλέον, μπορούμε να δημιουργήσουμε ένα πιο σύνθετο σύστημα μνήμης που να παρέχει διαφορετικούς τύπους προσβάσεων στα διαφορετικά μέρη του συστήματος. Ο συ-σχεδιασμός τέτοιων τύπων συστημάτων είναι περισσότερο δύσκολος, ιδιαίτερα όταν δεν κάνουμε τις υποθέσεις για τη δομή της πλατφόρμας.

## 7.12 Πολυεπεξεργασία σε Chip

Πολλοί σχεδιαστές, οι οποίοι είναι σχετικοί με την πληροφορική γενικού σκοπού σχετίζουν την πολυεπεξεργασία μόνο με τους διακομιστές υψηλών επιδόσεων, τις μηχανές βάσεων δεδομένων ή τους σταθμούς εργασίας για επιστημονικούς λόγους και τις διεργασίες heavy-duty ή μετατροπή σε 3D γραφικά ή πρόγνωση του καιρού. Στην πραγματικότητα η πολυεπεξεργασία στα ενσωματωμένα συστήματα είναι πολύ κοινή, παρόλο που συχνά παίρνει πολλές διαφορετικές μορφές. Η πιο επιτυχημένη μορφή της πολυεπεξεργασίας (SMP) στην οποία το σύστημα εξυπηρετεί πολλούς επεξεργαστές, είναι μέσω μια μορφής μνήμης, η οποία διαμοιράζεται (υπολογιστικά συστήματα διαμοιραζόμενης μνήμης). Στην SMP αρχιτεκτονική, οι διεργασίες χρονοδρομολογούνται ατομικά σε κάθε επεξεργαστή από το λειτουργικό σύστημα, το οποίο είναι συχνά υπεύθυνο για την μετανάστευση των διεργασιών κατά μήκος του επεξεργαστή, για να εξισορροπήσει την υπερχείλιση της μηχανής.

Στα ενσωματωμένα η πιο ευρέως υιοθετημένη μορφή πολυεπεξεργασίας είναι η ετερογενής, στην οποία καθεξοχήν διαφορετικοί επεξεργαστές βρίσκονται στο ίδιο σύστημα, αλλά ο καθένας μπορεί να θεωρηθεί ως ένα υποσύστημα (με μια ιδιωτική μνήμη), το οποίο δεν επηρεάζεται από τα άλλα στοιχεία του προγράμματος. Με άλλα λόγια, διαφορετικά λειτουργικά συστήματα (ή microkernels) ελέγχουν τους ατομικούς επεξεργαστές ο καθένας από τους οποίους τρέχει ένα προκαθορισμένο, καταναμημένο σύνολο διεργασιών. Η επικοινωνία μεταξύ των επεξεργαστών επιτυγχάνεται μέσα από μια καλά καθορισμένη μνήμη η οποία μοιράζεται και χειρίζεται τον συγχρονισμό των μηχανισμών, παρόμοια με τον τρόπο με τον οποίο χειριζόμαστε και άλλες μηχανές DMA. Σε αυτές τις διαμορφώσεις, είναι κοινό να ορίζουμε έναν επεξεργαστή ως τον κύριο επεξεργαστή του συστήματος. Ο κύριος αυτός επεξεργαστής ελέγχει την δραστηριότητα των διεργασιών πάνω στους άλλους

επεξεργαστές (σκλάβους), στέλνοντας τις κατάλληλες εντολές ελέγχου στους microkernels οι οποίοι τρέχουν πάνω στους σκλάβους.

Μια άλλη κοινή μορφή πολυεπεξεργασίας εκμεταλλεύεται τον έμφυτο παραλληλισμό των δεδομένων στην εφαρμογή. Αν μπορούμε να γράψουμε την εφαρμογή σε μορφή SPMD (το ίδιο πρόγραμμα σε πολλά δεδομένα, single program multiple data), μπορούμε να χρησιμοποιήσουμε πολλαπλούς ίδιους επεξεργαστές με ιδιωτικές μνήμες δεδομένων για να επεξεργασθούμε ανεξάρτητα περιοχές με προσωπικά δεδομένα. Μπορούμε να βρούμε παραδείγματα από τις πολλές εφαρμογές SPMD οι οποίες χειρίζονται δισδιάστατες εικόνες. (π.χ. ιατρική απεικόνιση) ή άλλες συνηθισμένες ροές δεδομένων (τηλεπικοινωνίες, ήχο και βίντεο). Αυτή η προσέγγιση δεν είναι καινούργια, αλλά χρονολογείται στα μέσα τις δεκαετίας του 1970. Συνεχίστηκε μέχρι και στην δεκαετία του 1980 και συμπεριλάμβανε την εκπληκτική ανάπτυξη των επεξεργαστών οι οποίοι σαφέστατα κατασκευάστηκαν για να καθορίζονται παράλληλα (όπως η Inmos's Transputers) και δοκιμάστηκαν για πολύ καιρό πριν εγκαταλειφθούν τελείως.

### **7.12.1 Συμμετρική Πολυεπεξεργασία**

Μέχρι πρόσφατα το SMP ήταν ακριβό και συνήθως ήταν αποτελεσματικό για συγκεκριμένα προβλήματα, και το οποίο το έκανε απαγορευτικά ακριβό για οτιδήποτε άλλο εκτός από ακριβούς σταθμούς υψηλών προδιαγραφών και διακομιστές. Τα τελευταία χρόνια οι επεξεργαστές SMP γίνονται πιο κοινοί και αρκετοί κατασκευαστές ενσωματωμένων συστημάτων προσφέρουν πολυπύρηννα συστήματα (όπως dual ή quad cores ARMS ή MIPS ή PPC ). Με αυτό το σενάριο τα ενσωματωμένα συστήματα γίνονται εμπορεύσιμα σαν τα SMP και τα ενσωματωμένα SMP συστήματα γίνονται οικονομικά πιο βιώσιμα.

Η ανάπτυξη του λογισμικού στα ενσωματωμένα πολυπύρηννα συστήματα είναι μια πρόκληση. Ήδη η βέλτιστη χρήση των παράλληλων διακομιστών που έχουν πολλούς πόρους είναι μια δύσκολη διαδικασία. Τα ΕΣ με τους πολύ λιγότερους πόρους και τις ακόμη περισσότερες απαιτήσεις κάνουν το έργο πολύ πιο δύσκολο, με λιγότερες δυνατότητες. Για παράδειγμα πολλές φορές απαιτείται η συνέπεια της μνήμης στα ΕΣ να υλοποιείται από μηχανισμούς λογισμικού ή υλικού των σχεδιαστών και όχι από κάποιο λειτουργικό σύστημα.

### **7.12.2 Ετερογενής Πολυεπεξεργασία**

Εκτός από τη συμμετρική πολυεπεξεργασία, στα ΕΣ συναντάμε και την ασύμμετρη πολυεπεξεργασία ή ετερογενής πολυεπεξεργασία. Αυτή είναι η πιο κοινή μορφή πολυεπεξεργασίας στα ενσωματωμένα SoCs σήμερα. Η πρώτη ερώτηση η οποία μας έρχεται στο μυαλό είναι η εξής: δεδομένου ότι ο γενικός σκοπός των επεξεργαστών είναι διαθέσιμος, γιατί θα πρέπει να χρησιμοποιήσουμε διαφορετικούς επεξεργαστές για κάθε μια διεργασία; Οι λόγοι είναι: οι εκτιμήσεις του κόστους, της απόδοσης και της προσαρμογής. Το να χρησιμοποιηθούν διαφορετικοί επεξεργαστές για διαφορετικές διεργασίες επιτρέπει στους σχεδιαστές να επιλέξουν την πιο αποτελεσματική ως προς το κόστος εναλλακτική για κάθε στάδιο της εφαρμογής.

Για παράδειγμα, το DSP και οι επεξεργαστές γενικού σκοπού έχουν ευδιάκριτα διαφορετικά σύνολα χαρακτηριστικών τα οποία τους κάνουν να ταιριάζουν καλύτερα σε διαφορετικού τύπου φόρτου εργασίας. Σε μια εφαρμογή, η οποία περιλαμβάνει και DSP και συστατικά υπολογισμού γενικού σκοπού, χωρίζοντας τις δύο διεργασίες και χρησιμοποιώντας ετερογενές σύστημα είναι πολύ αποτελεσματικότερο ως προς το κόστος από το να δοκιμάσουμε να επιβάλλουμε την ανάδραση σε ολόκληρη την εφαρμογή σε ένα παραδοσιακό γενικού σκοπού SMP (πολύ ακριβό) ή σε ένα μέρος ενός DSP (πολύ άκαμπτο).

Δυστυχώς, η ετερογενής πολυεπεξεργασία δεν είναι δωρεάν. Το μεγαλύτερο εμπόδιο έρχεται από την απουσία ενός διαισθητικού προτύπου προγραμματισμού. Οι διεργασίες θα πρέπει να είναι ρητά χωρισμένες για τους μεμονωμένους επεξεργαστές, η εσωτερική επικοινωνία και ο συγχρονισμός πρέπει να αντιμετωπιστούν χειροκίνητα, τα εργαλεία και τα δυαδικά συστήματα σύνταξης είναι διαφορετικά, και η αποσφαλμάτωση είναι αδέξια.

Παρά τις εγγενείς δυσκολίες στο πρότυπο προγραμματισμού, διάφορες τάσεις βιομηχανίας ωθούν έντονα στην κατεύθυνση της ετερογενούς πολυεπεξεργασίας. Πολλοί προμηθευτές και πρωτοβουλίες σε βιομηχανικό επίπεδο υποστηρίζουν μερικούς μικροελεγκτές και συνδυασμούς DSP που στοχεύουν σε μερικές από τις πιο ελπιδοφόρες περιοχές αγοράς. Παραδείγματος χάριν, και το PCA της Intel και οι γραμμές προϊόντων της Texas Instruments OMAP ανήκουν σε αυτήν την κατηγορία νέων πλατφορμών SoC. Μια από τις σημαντικότερες κατευθυντήριες δυνάμεις πίσω από την υιοθέτηση αυτών των μορφών αρχιτεκτονικής προέρχεται από τις αντιπαραβαλλόμενες απαιτήσεις.

Εκτιμώντας ότι οι εφαρμογές και τα λειτουργικά συστήματα παίρνουν έναν μακροχρόνιο χρόνο να μεταναστεύσουν στις νέες αρχιτεκτονικές, τα νέα προϊόντα ωθούν συνεχώς το σχεδιασμό με περισσότερες απαιτήσεις και ένα γρηγορότερο ποσοστό υιοθέτησης. Ο συνδυασμός ενός συμβατικού πυρήνα μικροεπεξεργαστή (όπως ARM) με έναν αποδοτικά προσανατολισμένο στον προγραμματισμό επιταχυντή (όπως πολλές σύγχρονες μηχανές DSP ή FPGA), υπόσχεται να παρέχει την ευελιξία που χρειάζεται για να αντιμετωπιστούν και τα δύο σύνολα απαιτήσεων. Πρέπει όμως να περάσουν πολλά στάδια μέχρι η ετερογενής πολυεπεξεργασία να γίνει κάτι συνηθισμένο και τυπικό.

### **7.13 Βελτιστοποίηση Σχεδίασης Ενσωματωμένων Συστημάτων με χρήση δυναμικής μνήμης**

Στο παρελθόν, οι περισσότερες εφαρμογές που ήταν στις ενσωματωμένες πλατφόρμες, παρέμειναν κυρίως στην κλασική περιοχή της επεξεργασίας σήματος, και απέφευγαν ενεργά τους αλγόριθμους που χρησιμοποιούν τα δυναμικά στοιχεία αποεκχώρησης στο χρόνο εκτέλεσης, που αποκαλείται επίσης και ως δυναμική μνήμη (dynamic memory, DM από εδώ και στο εξής). Πρόσφατα, τα πολυμέσα και οι ασύρματες εφαρμογές δικτύων για να μεταφερθούν στα ενσωματωμένα συστήματα έχουν δεχτεί μια πολύ γρήγορη αύξηση της ποικιλίας, πολυπλοκότητας και λειτουργίας.

Αυτές οι εφαρμογές (πχ. MPEG4 ή νέα πρωτόκολλα δικτύων) εξαρτώνται, με λίγες εξαιρέσεις, από το DM για τις διαδικασίες τους, λόγω της έμφυτης προβλεψιμότητας των δεδομένων εισόδου. Ο σχεδιασμός των τελικών ενσωματωμένων συστημάτων για το (στατικό) αποτύπωμα μνήμης στη χειρότερη περίπτωση<sup>1</sup> θα οδηγούσε να μην μπορεί να ικανοποιήσει τις απαιτήσεις αυτών των εφαρμογών. Ακόμα κι αν χρησιμοποιούνται οι μέσες τιμές των πιθανών εκτιμήσεων του αποτυπώματος μνήμης, αυτές οι στατικές λύσεις θα οδηγήσουν σε υψηλότερους αριθμούς αποτυπώματος μνήμης (δηλαδή, περίπου 25%) από τις λύσεις DM. Επιπλέον, αυτές οι ενδιάμεσες στατικές λύσεις δεν λειτουργούν σε ακραίες περιπτώσεις δεδομένων εισόδου, ενώ οι υλοποιήσεις με DM μπορούν να το κάνουν, δεδομένου ότι μπορούν να κλιμακώσουν το απαραίτητο αποτύπωμα μνήμης.

Κατά συνέπεια, πρέπει να χρησιμοποιηθούν στο σχεδιασμό ενσωματωμένων συστημάτων για δυναμικές εφαρμογές διαχειριστές δυναμικής μνήμης. Σήμερα, είναι πολλές οι γενικές διαθέσιμες πολιτικές διαχείρισης DM, και οι εφαρμογές τους, οι οποίες παρέχουν σχετικά καλή εκτέλεση και χαμηλό κατακερματισμό για συστήματα γενικής χρήσης [56]. Εντούτοις, για τα ενσωματωμένα συστήματα, τέτοιοι διαχειριστές πρέπει να εφαρμοστούν μέσα στο περιορισμένο λειτουργικό σύστημά τους (ΛΣ) και να λάβουν υπόψιν τους περιορισμένους διαθέσιμους πόρους.

Κατά συνέπεια, τα πρόσφατα ενσωματωμένα Λειτουργικά Συστήματα χρησιμοποιούν τους διαχειριστές DM σύμφωνα με την βασική ιεραρχία μνήμης και το είδος εφαρμογών που θα τρέξουν σε αυτούς. Συνήθως, οι εξατομικευμένοι διαχειριστές DM έχουν ως σκοπό να βελτιώσουν την απόδοση αλλά μπορούν επίσης να χρησιμοποιηθούν για να βελτιστοποιήσουν σε μεγάλο βαθμό το αποτύπωμα μνήμης έναντι των διαχειριστών γενικού σκοπού DM, το οποίο επηρεάζει την τελική κατανάλωση ενέργειας και την απόδοση στα ενσωματωμένα συστήματα, καθώς πρέπει να μοιραστούν την περιορισμένη διαθέσιμη μνήμη, εντός ολοκληρωμένου κυκλώματος με πολλές ταυτόχρονες δυναμικές εφαρμογές.

Για παράδειγμα, στους τρισδιάστατους οπτικούς αλγόριθμους, ένας κατάλληλα εξατομικευμένος σχεδιασμένος διαχειριστής DM μπορεί να βελτιώσει το αποτύπωμα μνήμης κατά 45% περίπου πέρα από τους συμβατικούς γενικής χρήσης διαχειριστές DM. Πόσο μάλλον, όταν χρησιμοποιούνται οι εξατομικευμένοι διαχειριστές DM, οι σχεδιασμοί των οποίων πρέπει να βελτιστοποιηθούν χειρωνακτικά από τον υπεύθυνο για την ανάπτυξη, εξετάζοντας μόνο έναν περιορισμένο αριθμό εναλλακτικών λύσεων σχεδιασμού και εφαρμογής. Αυτές καθορίζονται βασισμένες στην εμπειρία και την έμπνευση του σχεδιαστή.

Αυτή η εξερεύνηση είναι περιορισμένη λόγω της έλλειψης συστηματικών μεθοδολογιών, για να ερευνηθούν με συνέπεια το χώρο σχεδιασμού της διαχείρισης DM. Κατά συνέπεια, οι σχεδιαστές πρέπει να καθορίσουν, να κατασκευάσουν και να αξιολογήσουν χειρωνακτικά τις νέες εξατομικευμένες εφαρμογές των στρατηγικών DM, οι οποίες έχουν αποδειχθεί πολύ χρονοβόρες. Ο υπεύθυνος για την ανάπτυξη βρίσκεται αντιμέτωπος ακόμα, με τον καθορισμό της δομής του διαχειριστή DM και

πώς να σχεδιάσει το περίγραμμα ανά περίπτωση, ακόμα κι αν το ενσωματωμένο Λειτουργικό Σύστημα προσφέρει την ιδιαίτερη υποστήριξη για τις τυποποιημένες γλώσσες, όπως η C ή C++.

Σε αυτή την ενότητα, παρουσιάζουμε μια μεθοδολογία που επιτρέπει στους υπεύθυνους για την ανάπτυξη, να σχεδιάσουν τους εξατομικευμένους μηχανισμούς DM διαχείρισης, με το μειωμένο αποτύπωμα μνήμης που απαιτείται για αυτά τα δυναμικά πολυμέσα και τις ασύρματες εφαρμογές δικτύων. Καταρχήν, αυτή η μεθοδολογία οριοθετεί το σχετικό χώρο σχεδιασμού των διαχειριστικών αποφάσεων DM για ένα ελάχιστο αποτύπωμα μνήμης, στις δυναμικές ενσωματωμένες εφαρμογές. Μετά από αυτό, έχουμε μελετήσει τη σχετική επιρροή κάθε απόφασης του χώρου σχεδιασμού για το αποτύπωμα μνήμης, και έχουμε καθορίσει μια κατάλληλη διάταξη για να προσπελάσουμε αυτό το χώρο σχεδιασμού σύμφωνα με τη συμπεριφορά DM αυτών των δυναμικών εφαρμογών.

Κατά συνέπεια, οι κύριες συνεισφορές της μεθοδολογίας μας είναι διπλές:

- ο καθορισμός μιας συνεπούς ορθογωνιοποίησης του χώρου σχεδιασμού της διαχείρισης DM για τα ενσωματωμένα συστήματα και
- ο καθορισμός μιας κατάλληλης διάταξης των αποφάσεων σχεδιασμού, για τα δυναμικά πολυμέσα και τις ασύρματες εφαρμογές δικτύων (και οποιουδήποτε άλλου τύπου ενσωματωμένων εφαρμογών, που κατέχει τα ίδια δυναμικά χαρακτηριστικά αποεκχώρησης) για να ενισχυθούν οι σχεδιαστές και για να δημιουργήσουν προσαρμοσμένους διαχειριστές DM σύμφωνα με τη συγκεκριμένη δυναμική συμπεριφορά κάθε εφαρμογής.

## ΚΕΦΑΛΑΙΟ 8

### 8. ΤΡΟΠΟΙ ΥΛΟΠΟΙΗΣΗΣ ΕΝΣΩΜΑΤΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ

Η μεγάλη ποικιλία του οικοσυστήματος των ενσωματωμένων συστημάτων, που οφείλεται στο ευρύ φάσμα των απαιτήσεων που καλούνται να χρησιμοποιηθούν, αντικατοπτρίζεται και στις πάρα πολλές δυνατότητες υλοποίησης. Σε αντίθεση με τους τυπικούς προσωπικούς ή φορητούς υπολογιστές ή και διακομιστές, που οι μόνες επιλογές περιορίζονται στον τύπο του επεξεργαστή και της κεντρικής πλακέτας συστήματος. Τα Ενσωματωμένα Συστήματα μπορούν να υλοποιηθούν με πολλαπλούς τρόπους, ο καθένας από τους οποίους παρουσιάζει τα δικά του ιδιαίτερα χαρακτηριστικά και ταιριάζει καλύτερα σε συγκεκριμένες σχεδιαστικές απαιτήσεις.

#### 8.1 Εισαγωγή

Η επεξεργασία μέσα στο ενσωματωμένο σύστημα μπορεί να υλοποιηθεί μέσω των διαφόρων μονάδων εξειδικευμένου υλικού. Αυτά υλοποιούνται είτε σε μη προγραμματισμένο υλικό (ή αλλιώς σε ολοκληρωμένα κυκλώματα εξειδικευμένης εφαρμογής), είτε σε προγραμματιζόμενο υλικό (ή αλλιώς σε επί του πεδίου προγραμματιζόμενοι συστοιχίες πυλών). Ένας επεξεργαστής ειδικού σκοπού σε αντίθεση με έναν επεξεργαστή γενικού σκοπού είναι σχεδιασμένος σε ψηφιακό κύκλωμα, ώστε να εκτελεί μια συγκεκριμένη υπολογιστική εργασία ή εφαρμογή. Ο επεξεργαστής ειδικού σκοπού ονομάζεται αλλιώς και ως συνεπεξεργαστής ή επιταχυντής υλικού.

Ένας σχεδιαστής ενσωματωμένων συστημάτων μπορεί να επιτύχει διάφορα οφέλη επιλέγοντας να χρησιμοποιήσει ένα επεξεργαστή ειδικού σκοπού, ώστε να υλοποιήσει μια υπολογιστική εργασία. Πρώτον, η απόδοση των εξειδικευμένων ενσωματωμένων επεξεργαστών είναι ταχύτερη, εξαιτίας των λιγότερων κύκλων ρολογιού, ως αποτέλεσμα ενός εξατομικευμένου μονοπατιού δεδομένων. Επιπλέον, η απόδοση αυτή είναι βελτιωμένη, λόγω των μικρότερων κύκλων ρολογιού των απλούστερων λειτουργικών μονάδων και της απλούστερης λογικής ελέγχου.

Δεύτερον, το μέγεθος είναι μικρότερο, εξαιτίας του απλού μονοπατιού δεδομένων και ότι δεν απαιτείται μνήμη προγράμματος, εφόσον η υπολογιστική εργασία είναι υλοποιημένη σε ψηφιακό υλικό και υλοποιείται με συνδυαστικά ή ακολουθιακά κυκλώματα. Τρίτον, η κατανάλωση ενέργειας είναι λιγότερη, εξαιτίας του αποτελεσματικού υπολογισμού, χωρίς να υλοποιούνται περιττές συναρτήσεις. Όμως, υπάρχουν διάφορα μειονεκτήματα όπως το κόστος που ίσως είναι υψηλότερο, λόγω του υψηλού κόστους έρευνας και ανάπτυξης (NRE).

Ταυτόχρονα, ο χρόνος εισαγωγής στην αγορά είναι μεγαλύτερος και μειώνεται η ευελιξία σε σχέση με τους επεξεργαστές γενικού σκοπού. Από την άλλη πλευρά, η τεχνολογία επί του πεδίου προγραμματιζόμενοι συστοιχίες πυλών, προσπαθεί να διατηρήσει τα οφέλη που έχουν οι επεξεργαστές ειδικού σκοπού, αλλά και να βελτιώνει το κόστος, το χρόνο εισαγωγής στη αγορά και την ευελιξία.

## 8.2 Οι πρώτες υλοποιήσεις

Λίγα χρόνια μετά την ανακάλυψη του τρανζίστορ, η Texas Instruments κατασκεύασε μια σειρά διακριτών δομοστοιχείων, η οποία απαριθμούσε εκατοντάδες λογικά στοιχεία υλοποιώντας μια πληθώρα λειτουργιών. Η σειρά αυτή ήταν γνωστή ως 7400, από τον αριθμό που φέρει το πρώτο δομικό στοιχείο και αποτελείται από τέσσερις πύλες NAND. Αυτά τα στοιχεία ονομάζονται TTL, είχαν τυποποιημένες συσκευασίες και μπορούσαν να συνδεθούν εύκολα, ώστε να κατασκευάσουν μεγαλύτερα ψηφιακά συστήματα. Παρείχαν ένα πλήθος από λειτουργίες, όπως ψηφιακές πύλες, φλιπ-φλοπς, μετρητές, απομονωτές, αριθμητικές λογικές μονάδες.

Ενδεικτικό είναι το γεγονός, ότι τα περισσότερα μικρά και μεγάλα υπολογιστικά κέντρα του 1960 και 1970 χρησιμοποιούσαν αυτή την οικογένεια. Εκτός από την οικογένεια 7400 υπήρχε και η οικογένεια διακριτών στοιχείων 5400, η οποία είχε τα ίδια ακριβώς μέλη, αλλά παρείχε προδιαγραφές για στρατιωτικές εφαρμογές. Εκτός από την Texas Instruments που τα παρείχε αρχικά, και άλλες εταιρίες προσέφεραν συμβατά δομοστοιχεία με την ίδια αρίθμηση και έτσι υπήρχε μια αφθονία στοιχείων TTL. Με τον καιρό δημιουργήθηκαν και εκδόσεις που είχαν ελαφρώς διαφορετικά χαρακτηριστικά (αλλά παρείχαν τις ίδιες ψηφιακές λειτουργίες) εκδόσεις μειωμένης κατανάλωσης ενέργειας

## 8.3 Κατηγορίες Υλοποίησης

Για πέντε δεκαετίες, η αλματώδης πρόοδος της τεχνολογίας έχει βοηθήσει στην ανάπτυξη και υιοθέτηση των υπολογιστών σε όλους τους τομείς της καθημερινής μας ζωής. Έτσι, αν κοιτάξει κανείς γύρω του θα δει ότι περιτριγυρίζεται από ηλεκτρονικά συστήματα, που εμπεριέχονται για παράδειγμα στις τηλεοράσεις, στα αυτοκίνητα, στους υπολογιστές, στα κινητά τηλέφωνα, σε βομβητές κτλ. Τα περισσότερα από αυτά τα ηλεκτρονικά συστήματα έχουν κυκλώματα που ελέγχουν κάποιες ή όλες τις λειτουργίες της συσκευής.

Στα υπολογιστικά συστήματα γενικού σκοπού, ο σχεδιασμός ή η επιλογή του επεξεργαστή είναι μια ορθογώνια διαδικασία ως προς το λογισμικό και έτσι οι ομάδες είναι τελειώς ανεξάρτητες ή ακριβέστερα, συνεργάζονται μόνο οι ομάδες της αρχιτεκτονικής με αυτές που κατασκευάζουν τους μεταγωγτιστές και τους συμβολομεταφραστές. Οι υπόλοιποι προγραμματιστές, απλώς χρησιμοποιούν τα αναπτυξιακά εργαλεία που τους παρέχονται.

Στα ενσωματωμένα συστήματα όμως, πάντα υπάρχει ο συ-σχεδιασμός του υλικού και του λογισμικού, το οποίο περιγράφεται λεπτομερώς και σε επόμενη παράγραφο. Αρχικά, επιλέγεται ή σχεδιάζεται ο εξειδικευμένος επεξεργαστής που καλύπτει τις απαιτήσεις, ένα αρκετά δύσκολο εγχείρημα που δεν πρέπει να υποτιμάται, μαζί με τις αρχιτεκτονικές λεπτομέρειες που το συνοδεύουν όπως οι κρυφές μνήμες, οι μηχανισμοί συνοχής ή δυνατότητα πολυεπεξεργασίας κτλ.

Ο σχεδιασμός του υπόλοιπου συστήματος δεν είναι λιγότερο σημαντικός ή πιο δύσκολος. Υπάρχει αλληλεπίδραση των σχεδιαστικών αποφάσεων ανάμεσα στον επεξεργαστή και στα υπόλοιπα στοιχεία του ενσωματωμένου συστήματος. Για να

σχεδιαστεί ένας απλός επεξεργαστής θα πρέπει να εργάζονται δυο ή τρεις μηχανικοί, από έξι έως εννέα μήνες, προκειμένου να δημιουργήσουν μια περιγραφή επιπέδου μεταφοράς καταχωρητών (register transfer level, RTL). Το επόμενο βήμα είναι η κατασκευή, η δοκιμή, η ανάπτυξη λογισμικού κτλ. Κάποιοι πιστεύουν ότι ο πιο δύσκολος στόχος είναι να σχεδιαστεί ένας επεξεργαστής, για αυτό υποστηρίζουν ότι ο σχεδιασμός ενός πολύπλοκου ενσωματωμένου συστήματος ολοκληρώνεται με το σχεδιασμό του επεξεργαστή. Όμως, αυτό είναι λάθος, αφού το πιο σωστό είναι ότι ο ο σχεδιασμός ενός πολύπλοκου ενσωματωμένου συστήματος ξεκινάει με το σχεδιασμό του επεξεργαστή.

Στη συνέχεια τα ηλεκτρονικά κυκλώματα σχεδιάζονταν μόνο σε ολοκληρωμένα κυκλώματα εξειδικευμένα για την εφαρμογή (Application Specific Integrated Circuit, ASIC). Τα κυκλώματα αυτά εκτός από τις υψηλές επιδόσεις είχαν και υψηλό κόστος παραγωγής. Σήμερα, το κόστος αυτό δεν έχει μειωθεί, αλλά αυξάνεται συνεχώς εκθετικά ακολουθώντας το νόμο του Moore. Γι' αυτό το λόγο, ερευνητές έχουν υποστηρίξει ότι οι συνθήκες μας οδηγούν σε μια απομάκρυνση από τα ASIC και σε εντατικοποίηση της χρήσης των προγραμματιζόμενων υλοποιήσεων, της άλλης εναλλακτικής πορείας υλοποίησης συστημάτων. Οι δύο αυτές εναλλακτικές πορείες υλοποίησης έχουν θετικά και αρνητικά στοιχεία.

Συγκεκριμένα, τα ASIC έχουν πολύ χαμηλή ευελιξία (αφού σχεδιάζονται αποκλειστικά για μια εφαρμογή), έχουν πολύ χαμηλή κατανάλωση ενέργειας με υψηλές επιδόσεις και έχουν πολύ μεγάλο χρόνο ανάπτυξης. Στην ακριβώς αντίθετη θέση βρίσκονται οι επεξεργαστές γενικού σκοπού, οι οποίοι είναι αρκετά ευέλικτοι (αφού είναι γενικού σκοπού), αλλά υστερούν στις επιδόσεις και συνοδεύονται από μεγάλη κατανάλωση ενέργειας. Ανάμεσα στις δύο αυτές ακραίες θέσεις, υπάρχουν οι ψηφιακοί επεξεργαστές σήματος, οι οποίοι έχουν μεγάλη ευελιξία, χαμηλό χρόνο ανάπτυξης (αφού είναι προγραμματιζόμενοι και εύκολα μπορεί να μεταφραστεί μια εφαρμογή σε εκτελέσιμη μορφή για τον επεξεργαστή), αλλά η κατανάλωση ενέργειας και οι επιδόσεις υστερούν συγκρινόμενες με τα ASIC. Στις ενδιάμεσες θέσεις επίσης υπάρχουν τα FPGA, τα οποία έχουν μια παρόμοια ροή σχεδιασμού με τα ASIC, αλλά αρκετά πιο γρήγορη αφού παρέχουν τη δυνατότητα του προγραμματισμού, και του χαμηλότερου κόστους, γιατί παράγονται κατά μεγάλες ποσότητες.

Έως τώρα αναφερθήκαμε για υλοποιήσεις ενσωματωμένων συστημάτων που είτε φέρουν έναν επεξεργαστή, είτε αποτελούνται από εξειδικευμένο υλικό. Ως τώρα ήταν η εισαγωγή και δεν έχει δοθεί μια λεπτομερή ταξινόμηση. Είναι όμως ευκολονόητο ότι τα ΕΣ μπορούν να υλοποιηθούν με πάρα πολλούς τρόπους, και αυτό φαίνεται και από το μεγάλο πλήθος των ακρωνυμίων υλοποίησης, όπως CPLD, FPGA, ASIC, SOC, PSOC, DSP, ASIP, ASSP, PAL, CPU, MCPU, που θα περιγράψουμε λεπτομερώς στις επόμενες ενότητες.



## 8.4 Τεχνολογίες ενσωματωμένων συστημάτων

Υπάρχουν τρεις τεχνολογίες-κλειδιά που αφορούν στα ενσωματωμένα συστήματα:

- Τεχνολογία επεξεργαστών
- Τεχνολογία Ολοκληρωμένων κυκλωμάτων
- Τεχνολογία σχεδίασης

Οι τεχνολογίες αυτές αλληλοεπικαλύπτονται: εάν ολοκληρωμένο κύκλωμα μπορεί να περιέχει επεξεργαστή της μιας ή της άλλης τεχνολογίας και να είναι σχεδιασμένο με την μια ή την άλλη τεχνολογία σχεδίασης.

### 8.4.1 Τεχνολογίες επεξεργαστών

- a) Επεξεργαστές γενικού σκοπού: Προγραμματιζόμενες διατάξεις ικανές να ανταποκριθούν στις ανάγκες πολλών εφαρμογών. Έχουν μεγάλη RAM (εσωτερική ή εξωτερική), και συνήθως στηρίζονται σε αρχιτεκτονική Von-Neumann και ένα γενικό διάδρομο επεξεργασίας (data path). Παραδείγματα: Pentium, 8088, 68000.
- b) Εξειδικευμένοι επεξεργαστές (application-specific). Αποτελούν κατηγορία προγραμματιζόμενων επεξεργαστών που είναι βελτιστοποιημένοι για συγκεκριμένη κατηγορία εφαρμογών. Έχουν περιορισμένη RAM πάνω στο τσιπ, συνήθως υλοποιούν αρχιτεκτονική Harvard, η επεξεργασία τους είναι ειδικού σκοπού και έχουν ενσωματωμένα πολλά περιφερειακά κυκλώματα. Επίσης, έχουν μνήμη προγράμματος. Παραδείγματα: microcontrollers, DSP processors.
- c) Επεξεργαστές μοναδικού σκοπού (single purpose): Ψηφιακά κυκλώματα σχεδιασμένα να εκτελούν μόνον μία διεργασία. Περιλαμβάνουν μόνον τα απαραίτητα κυκλώματα που χρειάζονται ώστε να εκτελέσουν ένα μοναδικό πρόγραμμα. Δεν έχουν μνήμη προγράμματος, αλλά σε ορισμένες περιπτώσεις χρειάζονται λίγη RAM. Παραδείγματα: επιταχυντές υλικού, όπως video compressors, ψηφιακά φίλτρα, επεξεργαστές μηχανικής όρασης. Οι περιφερειακοί ελεγκτές (κυκλώματα επικοινωνίας με θύρες I/O και με ICs) θεωρούνται επίσης επεξεργαστές μοναδικού σκοπού.

Οι επεξεργαστές γενικού σκοπού και εξειδικευμένοι επεξεργαστές είναι προγραμματιζόμενοι, άρα για να λειτουργήσουν πρέπει να γράψουμε κώδικα γι' αυτούς. Από αυτή την άποψη αντιπροσωπεύουν το μέρος του συστήματος που ταυτίζεται με το ΛΟΓΙΣΜΙΚΟ.

Στους επεξεργαστές μοναδικού σκοπού, το κύκλωμα «είναι» το πρόγραμμα, άρα δεν γράφουμε κώδικα γι' αυτούς. Αντιπροσωπεύουν το ΥΛΙΚΟ μέρος του συστήματος. Στα ενσωματωμένα συστήματα, το υλικό και το λογισμικό σχεδιάζονται παράλληλα, κάτι που αναφέρεται ως συνσχεδιασμός υλικού-λογισμικού (hardware-software codesign).

### 8.4.2 Τεχνολογίες ολοκληρωμένων κυκλωμάτων

- a) Πλήρως προσαρμοσμένα (Full custom/VLSI). Όλα τα στρώματα του κυκλώματος σχεδιάζονται από την αρχή, από τα τραζίστορ μέχρι τις διασυνδέσεις.

- b) Μερικώς προσαρμοσμένα (Semi-custom ASIC). Τα χαμηλότερα στρώματα είναι μερικώς σχεδιασμένα. Η σχεδίαση γίνεται με τοποθέτηση και διασύνδεση προσχεδιασμένων βαθμίδων.
- c) Διαμορφούμενες διατάξεις PLD (CPLDs/FPGAs). Όλα τα στρώματα υπάρχουν εξαρχής μέσα στο τσιπ. Η σχεδίαση γίνεται με τη διαμόρφωση των πινάκων αναφοράς (LUT) και των προγραμματιζόμενων διασυνδέσεων.

### 8.4.3 Τεχνολογίες σχεδίασης

Για τη δημιουργία της επιθυμητής λειτουργικότητας έχουν αναπτυχθεί διάφορα εργαλεία σχεδίασης. Για το λογισμικό διαθέτουμε ολοκληρωμένα περιβάλλοντα για ανάπτυξη λογισμικού, όπως assemblers, μεταγλωττιστές, προσομοιωτές και προγραμματιστές. Για το υλικό μέρος διαθέτουμε εργαλεία σχεδίασης τύπου CAD και γλώσσες περιγραφής υλικού. Τα εργαλεία αυτά επιτρέπουν τη σχεδίαση σε διάφορα επίπεδα:

- Στο επίπεδο του ολοκληρωμένου συστήματος (system level)
- Στο επίπεδο της συμπεριφοράς του κυκλώματος (behavioral level)
- Στο δομικό επίπεδο της μεταφοράς σημάτων ανάμεσα σε πύλες και καταχωρητές (Register-transfer level).

## 8.5 Διασύνδεση και Είσοδος / Έξοδος

Όπως κάθε υπολογιστικό σύστημα, έτσι και τα ενσωματωμένα συστήματα έχουν ένα θεμελιώδη σκοπό ύπαρξης, ο οποίος είναι η επεξεργασία των δεδομένων και η λήψη αποφάσεων ή εκτέλεση ενεργειών. Αυτό επιτυγχάνεται με τους τρόπους εισόδου και εξόδου του επεξεργαστή από τα περιφερειακά, μέσω κατάλληλων θυρών. Τα δεδομένα μεταφέρονται από και προς τα περιφερειακά μέσω κατάλληλων δικτύων διασύνδεσης και διαύλων. Αν και τα ενσωματωμένα συστήματα χρησιμοποιούν τους περισσότερους προτυποποιημένους διαύλους που υπάρχουν στα τυπικά υπολογιστικά συστήματα και τους διακομιστές, εντούτοις χρησιμοποιούν και πολλούς διαύλους που τους συναντάμε μόνο σε αυτή την περιοχή, λόγω της ευρείας χρήσης τους. Σε αυτό το κεφάλαιο περιγράφουμε τους διαύλους και τη επικοινωνία και στη συνέχεια τις τυπικές λειτουργίες εισόδου εξόδου.

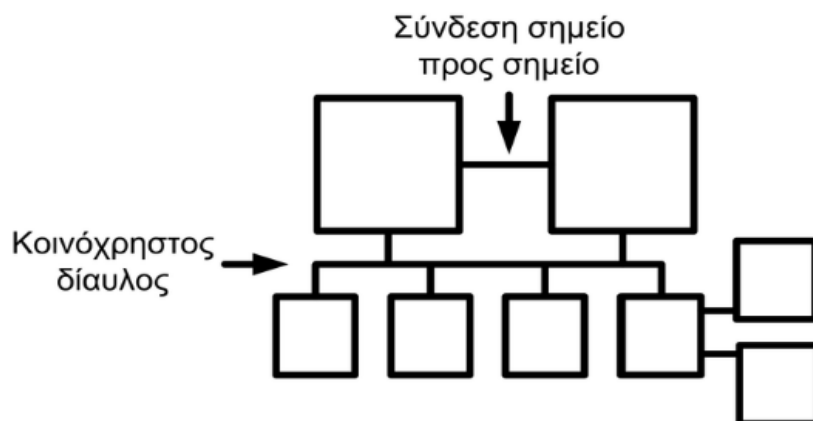
### 8.5.1 Διασύνδεση

Στη πιο βασική μορφή του, ένας υπολογιστής περιέχει ένα επεξεργαστικό στοιχείο (processing element, PE) είτε είναι επεξεργαστής, είτε είναι επαναδιαμορφώσιμη λογική, είτε ASIC, μνήμη και έναν ή παραπάνω τρόπους επικοινωνίας με το εξωτερικό περιβάλλον. Στις μέρες μας μπορεί ένας μηχανικός ενσωματωμένων συστημάτων να προμηθευτεί έναν πλήρη υπολογιστή σε μέγεθος όσο ένα πακέτο τσιγάρα, με κόστος κάτω από 10\$, με τους πιο γνωστούς αντιπρόσωπους τον υπολογιστή CHIP (κόστος 9\$, με 512MB RAM, 4MB Flash, ARM7, USB, HDMI, GPIO) ή τον υπολογιστή Raspberry Pi Zero (κόστος 5\$, με 512MB RAM, ARM6, USB, HDMI, GPIO).

Αυτοί οι υπολογιστές, που ονομάζονται υπολογιστές μιας πλακέτας (single board computers, SBC), μπορούν να χρησιμοποιηθούν με το κατάλληλο λογισμικό σε ενσωματωμένες εφαρμογές μέτριων υπολογιστικών απαιτήσεων, χωρίς κανένα

πρόβλημα. Όπως οι μεγαλύτεροι υπολογιστές, έτσι και αυτοί, αλλά και κάθε υπολογιστικό σύστημα, χρησιμοποιούν κανάλια διασύνδεσης ή διαύλους για να συνδέουν τα περιφερειακά και τα δομοστοιχεία, τόσο εντός του ολοκληρωμένου κυκλώματος όσο και εκτός. Υπάρχουν πάρα πολλοί δίαυλοι με διαφορετικά χαρακτηριστικά. Ο μηχανικός θα πρέπει να επιλέγει κάθε φορά το δίαυλο που θα χρησιμοποιήσει για την επικοινωνία, με βάση τη διαθεσιμότητα και την κάλυψη των αναγκών.

Αν το ενσωματωμένο σύστημα βρίσκεται ήδη στη διάθεσή του, τότε θα πρέπει να χρησιμοποιήσει τους διαύλους που αυτό υποστηρίζει, όπως για παράδειγμα το CHIP που υποστηρίζει το δίαυλο USB ή το GPIO και για να διασυνδεθεί θα πρέπει να επιλεγεί ένα περιφερειακό που να είναι συμβατό με το USB ή μπορεί να επικοινωνήσει μέσω του GPIO. Αν το ενσωματωμένο σύστημα βρίσκεται σε κατάσταση ανάπτυξης, τότε θα πρέπει να επιλεγθούν οι δίαυλοι που θα υποστηρίζει. Η σύνδεση του επεξεργαστή με το περιφερειακό γίνεται μέσω ενός δικτύου διασύνδεσης (Interconnection Network, IN), και περιλαμβάνει είτε αποκλειστικές γραμμές σημείο-προς-σημείο, ή διαμοιραζόμενα κανάλια πολλών συσκευών .

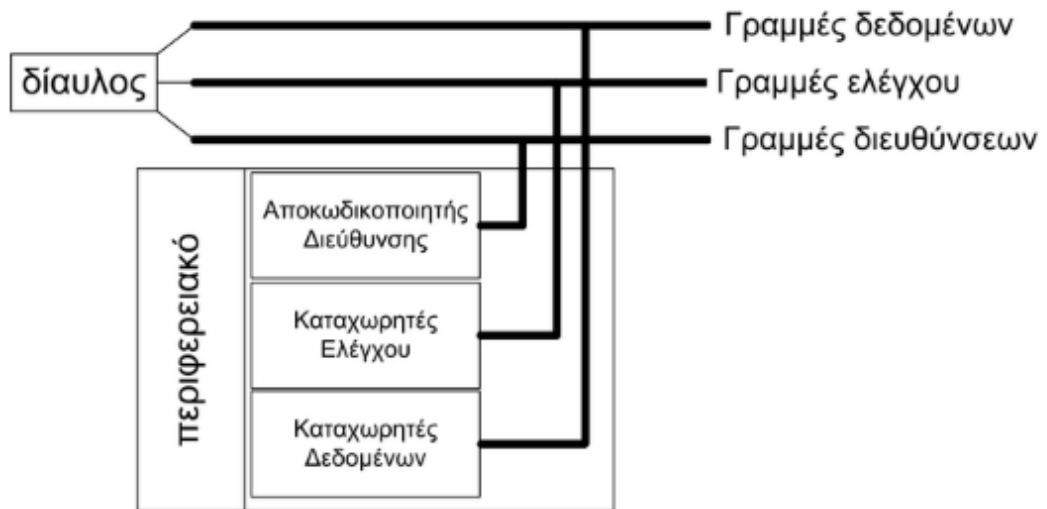


Σχήμα 8.5.1 (Εικόνα 1): Το δίκτυο διασύνδεσης ενός συστήματος αποτελείται από κοινόχρηστους διαύλους και αποκλειστικές διασυνδέσεις σημείο-προς-σημείο.

Για να επικοινωνήσει ο επεξεργαστής με ένα περιφερειακό θα πρέπει να του στείλει πληροφορίες που ανήκουν σε 3 κατηγορίες: (α) δεδομένα, (β) διεύθυνση, και (γ) έλεγχος. Τα δεδομένα είναι το ωφέλιμο φορτίο και μπορεί ο επεξεργαστής είτε να στείλει ή να ζητήσει να διαβάσει. Η διεύθυνση χρησιμοποιείται είτε στη περίπτωση που ένα περιφερειακό έχει πολλαπλές θέσεις μνήμης και ο επεξεργαστής αιτεί πρόσβαση σε μια από αυτές, είτε για να προσδιοριστεί ακριβώς το περιφερειακό, αν υπάρχουν περισσότερο από ένα, από τα οποία το καθένα έχει δικιά του μοναδική διεύθυνση.

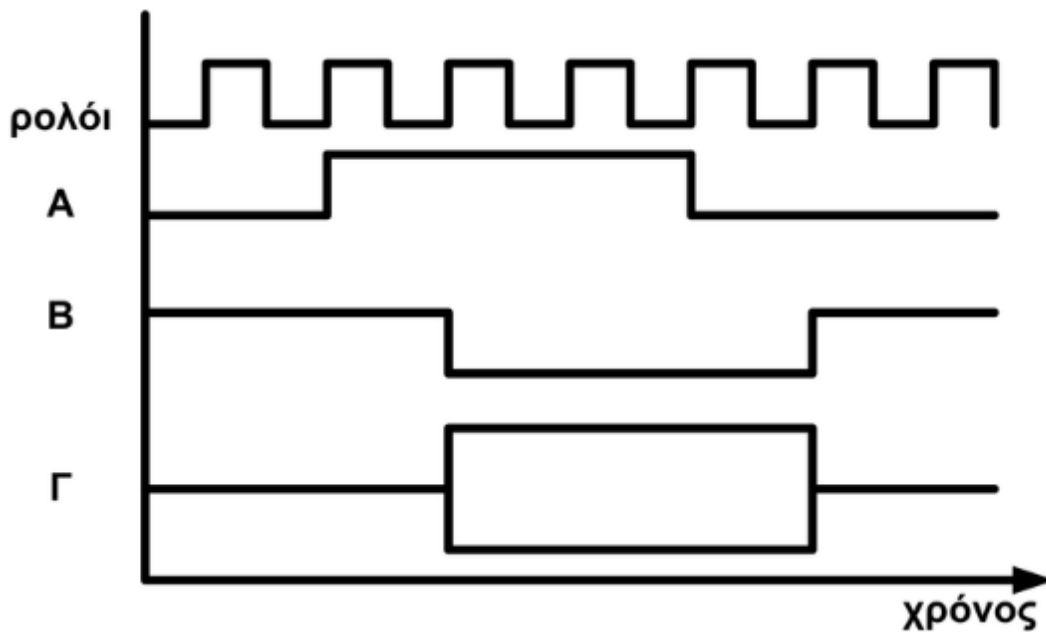
Ο έλεγχος προσδιορίζει κυρίως αν θα γίνει εγγραφή ή ανάγνωση, αλλά μπορεί να χρησιμοποιηθεί και για πιο προχωρημένες λειτουργίες όπως αίτηση για διακοπή, εξυπηρέτηση διακοπής, αίτηση για έλεγχο του διαύλου κ.α., που θα αναλύσουμε σε επόμενη ενότητα. Το κάθε περιφερειακό έχει μια διεπαφή, στην οποία υπάρχουν οι καταχωρητές ελέγχου, κατάστασης και δεδομένων. Ο καταχωρητής ελέγχου τροποποιείται από τον επεξεργαστή και καθορίζει τη λειτουργία που αιτείται, ο

καταχωρητής κατάστασης δείχνει την κατάσταση της συσκευής (αν είναι αδρανής, αν έχει δεδομένα προς αποστολή, αν είναι απασχολημένη με προηγούμενη επεξεργασία).



Σχήμα 8.5.1 (Εικόνα 2) : Τα περιφερειακά για να συνδεθούν στο δίαυλο χρησιμοποιούν μια διεπαφή που επεξεργάζεται τα δεδομένα, τα σήματα ελέγχου και τη διεύθυνση που έχει τοποθετήσει ο επεξεργαστής.

Η λειτουργία, η χρήση και η πρόσβαση στους προτυποποιημένους διαύλους καθορίζεται από αυστηρά πρωτόκολλα, ηλεκτρικών προδιαγραφών (π.χ. τι τάση θα πρέπει να υπάρχει και σε ποιους ακροδέκτες), χρονισμού (για πόσο χρονικό διάστημα θα πρέπει να εφαρμοστεί κάτι στο δίαυλο), και καταστάσεων (ποιες είναι οι καταστάσεις που διέρχεται μια μεταφορά). Τα πρωτόκολλα διαύλων καθορίζουν πως επικοινωνούν οι συσκευές και πρέπει όλες οι συνδεδεμένες συσκευές να τα σέβονται στην εντέλεια. Προκειμένου ένα περιφερειακό να είναι συμβατό με το συγκεκριμένο δίαυλο θα πρέπει να έχει πιστοποιηθεί ότι ακολουθεί όλες τις προδιαγραφές, διαφορετικά είτε δε θα λειτουργήσει, είτε θα προκαλέσει πρόβλημα επικοινωνίας ή και καταστροφής στις υπόλοιπες συσκευές. Συνοψίζοντας, ο δίαυλος είναι ένα κοινόχρηστο μέσο μετάδοσης που αποτελείται από καλώδια και προδιαγραφές.



Σχήμα 8.5.1 (Εικόνα 3) : Η λειτουργία κάθε διαύλου καθορίζεται από το διάγραμμα χρονισμού. Στην εικόνα φαίνεται το ρολόι, ένα σήμα A που από χαμηλή στάθμη εξυψώνεται 3 κύκλους ρολογιού, ένα σήμα B που βρίσκεται σε υψηλή στάθμη και χαμηλώνει στο 0 για 3 κύκλους ρολογιού, και ένα σήμα Γ που βρίσκεται για 3 κύκλους σε μια απροσδιόριστη κατάσταση (π.χ.είτε είναι 1 είτε 0, είτε πολλαπλά διαφορετικά bit).

Το πρώτο χαρακτηριστικό των διαύλων είναι το μέγεθος της ταυτόχρονης μεταφοράς των δεδομένων. Υπάρχουν δυο βασικές κατηγορίες: (α) σειριακό και (β) παράλληλο. Στη σειριακή επικοινωνία στέλνεται το κάθε bit ξεχωριστά, και ο δέκτης έχει τους κατάλληλους καταχωρητές που επιτελούν τη μετατροπή σειριακό σε παράλληλο, αφού τα δεδομένα χρησιμοποιούνται σε λέξεις των 8 ή 16 ή 13 bit. Στην παράλληλη επικοινωνία, υπάρχουν πολλαπλοί αγωγοί μεταφοράς που χρησιμοποιούνται ταυτόχρονα. Μέχρι το τέλος της δεκαετίας του 1990, υπήρχε μια προτίμηση στους παράλληλους διαύλους για τη μεταφορά μεγάλου όγκου δεδομένων, όπως π.χ. από το σκληρό δίσκο ή την κάρτα γραφικών.

Όμως, διαπιστώθηκαν αρκετά προβλήματα που εμπόδιζαν τα συστήματα να επιτύχουν μεγαλύτερες ταχύτητες επικοινωνίας και έτσι εγκαταλείφθηκαν οι παράλληλοι δίαυλοι. Το πρώτο πρόβλημα είναι ο όγκος των διαύλων. Ένας σειριακός δίαυλος καταλαμβάνει πολύ μικρότερη επιφάνεια από τον παράλληλο. Οι παράλληλοι δίαυλοι παρουσιάζουν διαφορετικές καθυστερήσεις σε κάθε αγωγό, επειδή είναι αδύνατο να κατασκευαστούν όλοι οι αγωγοί με μια ιδανική διαδικασία και να έχουν ίδιες αντιστάσεις. Η διαφορά της αντίστασης, όσο μικρή και να είναι, προκαλεί διαφορετική καθυστέρηση σε κάθε αγωγό και έτσι δημιουργείται πρόβλημα συγχρονισμού.

Οι σειριακοί δίαυλοι δεν έχουν τέτοιο πρόβλημα, αφού όλα τα bit διέρχονται από τον ίδιο αγωγό και έχουν την ίδια καθυστέρηση. Οι παράλληλοι δίαυλοι δεν έχουν εγκαταλειφθεί. Χρησιμοποιούνται μέσα στα ολοκληρωμένα κυκλώματα, αν και υπάρχει μια τάση να αποσυρθούν και εκεί. Εκτός ολοκληρωμένου κυκλώματος, χρησιμοποιούνται μόνο σειριακοί δίαυλοι σε κάθε σύγχρονο ψηφιακό σύστημα. Υπάρχουν όμως και υβριδικές υλοποιήσεις, στις οποίες χρησιμοποιείται ένας

σειριακός δίαυλος (όπως το PCI express) και πολλαπλές ανεξάρτητες παράλληλες κλωνοποιήσεις αυτού (όπως το PCI express x8, που αποτελείται από 8 σειριακούς διαύλους).

Το δεύτερο χαρακτηριστικό των διαύλων αφορά το μέγιστο αριθμό των συνδεδεμένων συσκευών. Αν είναι μόνο δυο συσκευές, ο δίαυλος αφορά την επικοινωνία σημείο-προς-σημείο (point-to-point), ενώ αν αφορά περισσότερες από δυο συσκευές, ο δίαυλος είναι κοινόχρηστος. Αυτό το χαρακτηριστικό είναι ανεξάρτητο ως προς το προηγούμενο. Δηλαδή, υπάρχουν σειριακοί δίαυλοι που ανήκουν στην κατηγορία σημείο-προς-σημείο (π.χ. το RS232) και άλλοι που ανήκουν στην κατηγορία του κοινόχρηστου (π.χ. το 1-wire). Επίσης, υπάρχουν παράλληλοι δίαυλοι που ανήκουν στην κατηγορία σημείο-προς-σημείο (π.χ. Accelerated Graphics Port, AGP) και παράλληλο δίαυλοι που ανήκουν στην κατηγορία της κοινόχρηστης επικοινωνίας (π.χ. PCI).

Το τρίτο χαρακτηριστικό αφορά το συγχρονισμό. Υπάρχουν σύγχρονοι και ασύγχρονοι δίαυλοι. Ο όρος σύγχρονος σημαίνει συγχρονισμένος με το ίδιο κοινό ρολόι επικοινωνίας. Ο επεξεργαστής και τα περιφερειακά που συνδέονται έχουν πρόσβαση στο ίδιο ρολόι του συστήματος, το οποίο είτε παρέχεται με ξεχωριστό κανάλι διαύλου, είτε μπορεί να εξαχθεί από την κωδικοποίηση των δεδομένων από τον παραλήπτη (με όφελος λιγότερους αγωγούς). Στην περίπτωση του σύγχρονου δίαυλου, η αίτηση τοποθετείται στο κανάλι ελέγχου μαζί με τη διεύθυνση για κάποιους συγκεκριμένους κύκλους ρολογιού και μετά γίνεται η ανάγνωση των δεδομένων από το περιφερειακό ή η εγγραφή των δεδομένων. Όταν ολοκληρωθεί το έργο, απομακρύνεται ο έλεγχος και η διεύθυνση. Η αλληλουχία των καταστάσεων ορίζεται σαφώς από τον επεξεργαστή και τα περιφερειακά στις ρυθμίσεις του συστήματος, με τα wait states (καταστάσεις αναμονής).

## 8.5.2 Είσοδος και Έξοδος

Με τον όρο είσοδο και έξοδο ενός ενσωματωμένου συστήματος, αναφερόμαστε στην τοποθέτηση δεδομένων που προέρχονται από κάποια εξωτερική συσκευή μέσα σε καταχωρητές και στην αντιγραφή των δεδομένων των καταχωρητών του επεξεργαστή, στους ακροδέκτες της συσκευασίας και στη συνέχεια, την κατάλληλη μεταφορά στο περιφερειακό δομοστοιχείο.

Το πως υλοποιείται η είσοδος/έξοδος (E/E) εξαρτάται από το συγκεκριμένο επεξεργαστή. Μικροί 8bit επεξεργαστές, μπορούν να υποστηρίξουν μόνο απλές λειτουργίες, ενώ σύνθετοι 32bit μπορούν να υποστηρίξουν περισσότερες τεχνικές. Για να υλοποιηθεί η E/E απαιτείται αφενός υποστήριξη από το υλικό, αφετέρου και προγραμματιστική υποστήριξη.

Υπάρχουν 2 δημοφιλείς τεχνικές που υποστηρίζουν τις E/E: (α) περιόδευση (polling), και (β) διακοπές (interrupts). Όπως αναφέραμε στην προηγούμενη ενότητα, υπάρχουν κατ ελάχιστον 3 καταχωρητές στο περιφερειακό. Ένας από αυτούς είναι ο καταχωρητής κατάστασης. Ο επεξεργαστής ελέγχει περιοδικά τον καταχωρητή κατάστασης κάθε περιφερειακού για να διαπιστώσει αν υπάρχουν δεδομένα που πρέπει να λάβει. Αν ο καταχωρητής σηματοδοτήσει ότι υπάρχουν δεδομένα (π.χ. από το πληκτρολόγιο), τότε διαβάζει τον καταχωρητή δεδομένων.

Είναι ευκολονόητο ότι τις περισσότερες φορές δεν υπάρχουν δεδομένα και έτσι σπαταλώνονται κύκλοι του επεξεργαστή. Αυτή η τεχνική είναι πολύ απλή να υλοποιηθεί και δεν απαιτεί καθόλου υποστήριξη από το υλικό. Αυτή η τεχνική μπορεί να παρομοιαστεί με κάποιον που βλέπει συνεχώς το κινητό τηλέφωνο του αν έχει λάβει SMS. Με τις διακοπές υποστηρίζεται και η διαιτησία του διαύλου. Μια συσκευή που θέλει να στείλει δεδομένα στον επεξεργαστή χρησιμοποιεί μια διακοπή για να τον ενημερώσει. Ο επεξεργαστής εξετάζει τη διακοπή και παρέχει τον έλεγχο, αφού χρησιμοποιήσει κατάλληλο σήμα.

Η δεύτερη τεχνική είναι η βελτιωμένη έκδοση E/E. Σε αυτή την τεχνική κάθε περιφερειακό συνδέεται με μια αποκλειστική γραμμή με τον επεξεργαστή, και όταν θελήσει να στείλει δεδομένα, αλλάζει τη λογική στάθμη της γραμμής, προκαλώντας μια διακοπή αίτησης εξυπηρέτησης (interrupt service request, IRQ). Ο επεξεργαστής σταματάει να εκτελεί αυτό που επεξεργάζονταν, ενημερώνει το περιφερειακό με κατάλληλο σήμα επιβεβαίωσης διακοπής (interrupt acknowledge) και αρχίζει την εκτέλεση ενός ειδικού κώδικα που ονομάζεται ρουτίνα εξυπηρέτησης διακοπής (interrupt service routine, ISR).

Αυτός ο κώδικας αναλαμβάνει τη μεταφορά των δεδομένων από το περιφερειακό στον επεξεργαστή. Σε περίπτωση που ο επεξεργαστής δεν έχει τόσες πολλές γραμμές διακοπών, τότε μπορεί να χρησιμοποιηθεί μια υβριδική διασύνδεση στην οποία συνδέονται πολλές συσκευές σε μια γραμμή διακοπής. Όταν ο επεξεργαστής διαπιστώσει ότι έχει συμβεί μια διακοπή, σταματάει το έργο που εκτελεί, και ελέγχει όλες τις συσκευές που είναι συνδεδεμένες στο κοινό κανάλι με περιόδευση. Επειδή η χρήση της διακοπής, σημαίνει την παρεμπόδιση εκτέλεσης του προγράμματος που εκτελείται ήδη, θα πρέπει να έχει μικρή καθυστέρηση (latency), ώστε να μη προκαλεί πρόβλημα. Κάποιοι επεξεργαστές ενσωματωμένων συστημάτων υποστηρίζουν και πολύ γρήγορες κλήσεις εξυπηρέτησης διακοπών, όπως ο ARM που παρέχει είτε την κανονική εξυπηρέτηση, είτε τη γρήγορη εξυπηρέτηση. Η επιλογή εξαρτάται από τον προγραμματιστή και τις δυνατότητες που θέλει να χρησιμοποιήσει, αφού η γρήγορη εξυπηρέτηση παρέχει μειωμένες λειτουργίες.

Στην απλή υλοποίηση των διακοπών, ο επεξεργαστής εκτελεί κάθε φορά την ίδια συνάρτηση εξυπηρέτησης διακοπής, ανεξαρτήτου του πια συσκευή την έχει δημιουργήσει. Μια βελτιωμένη υλοποίηση υποστηρίζει διαφορετικές ρουτίνες εξυπηρέτησης διακοπής για κάθε περιφερειακό. Σε αυτή την περίπτωση, όλες οι διακοπές είναι αριθμημένες και μοναδικές. Κατά την έναρξη λειτουργίας του συστήματος καταγράφονται οι αντιστοιχίσεις της κάθε διακοπής με τη διεύθυνση που ξεκινάει η αντίστοιχη ρουτίνα εξυπηρέτησης διακοπής. Οπότε, όταν εμφανιστεί μια συγκεκριμένη διακοπή, ο επεξεργαστής εκτελεί τον κώδικα που έχει καθοριστεί.

Αυτό το είδος της διακοπής ονομάζεται διανυσματική διακοπή (vectored interrupt), επειδή για κάθε διακοπή αντιστοιχεί ένα διάνυσμα που είναι μια διεύθυνση μνήμης κώδικα. Για λόγους διευκόλυνσης του επεξεργαστή, υπάρχει ένα ειδικό υλικό που ονομάζεται προγραμματιζόμενος ελεγκτής διακοπών (programmable interrupt controller, PIC), στο οποίο συνδέονται όλα τα κανάλια διακοπών, και όταν συμβεί μια διακοπή αυτό ενημερώνει τον επεξεργαστή μαζί με τη διεύθυνση του ISR. Η Intel ήταν η πρώτη που πρόσθεσε αυτό το ειδικό ολοκληρωμένο κύκλωμα το 1990 και του έδωσε την ονομασία 89C59A.

Μετά από 10 χρόνια παρουσιάστηκε μια βελτιωμένη έκδοση, ο ελεγκτής APIC που υποστηρίζει 255 προγραμματιζόμενες διακοπές, ταυτόχρονη είσοδο/έξοδο, παρέχει μετρητές υψηλής ακριβείας, μεταβίβαση διακοπής στον πρώτο διαθέσιμο επεξεργαστή και βελτίωση στην απόδοση επεξεργασίας των διακοπών. Στα σημερινά συστήματα ο προγραμματιζόμενος ελεγκτής διακοπών βρίσκεται πάνω στο ίδιο chip με τον επεξεργαστή. Το APIC απαιτεί επίσης την κατάλληλη υποστήριξη από το λειτουργικό σύστημα, προκειμένου να ρυθμιστεί.

Με την εισαγωγή των πολλαπλών διακοπών, δημιουργήθηκε η απαίτηση των προτεραιοτήτων. Κάποιες διακοπές είναι πιο σημαντικές από κάποιες άλλες, όπως για παράδειγμα η διακοπή που έρχεται από το ρολόι του συστήματος, είναι πιο σημαντική από τη διακοπή που έρχεται από το πληκτρολόγιο. Τα περισσότερα σημερινά συστήματα υποστηρίζουν προτεραιότητες ως εξής: μια ρουτίνα εξυπηρέτησης διακοπής μπορεί να σταματήσει προσωρινά να εκτελείται, αν συμβεί μια διακοπή μεγαλύτερης προτεραιότητας και μόνο τότε. Δηλαδή, οι διακοπές χαμηλότερης προτεραιότητας δεν ικανοποιούνται. Για να υποστηρίζουν αυτή τη δυνατότητα, θα πρέπει οι επεξεργαστές να χρησιμοποιούν εμφωλευμένες διακοπές (nested interrupts), και επίσης να μπορούν να απενεργοποιούν προσωρινά τις διακοπές.

Η δυνατότητα προσωρινής απενεργοποίησης των διακοπών χρησιμοποιείται σε περίπτωση που ο επεξεργαστής εκτελεί ένα κομμάτι κώδικα που δεν πρέπει να διακοπεί σε καμία περίπτωση, όπως π.χ. όταν γίνεται εξυπηρέτηση μιας διακοπής υψηλής προτεραιότητας. Για την υλοποίηση υπάρχει μια σημαία μέσα στον επεξεργαστή, η οποία τίθεται σε '0' όταν ο επεξεργαστής δε δέχεται διακοπές, και '1' όταν δέχεται. Η σημαία αυτή στην αρχιτεκτονική x86 είναι η IF και με τις εντολές assembly CLI (clear interrupts) και STI (set interrupts), μπορεί να απενεργοποιηθεί ή να ενεργοποιηθεί το χειρισμό διακοπών. Παρομοίως, στον ARM καλείται η εντολή MSR με τα κατάλληλα ορίσματα όπου γράφει στην κατάλληλη θέση του καταχωρητή κατάστασης που αντιστοιχεί στην αποδοχή των διακοπών, είτε τη τιμή 1 είτε την τιμή 0.

Όταν ο επεξεργαστής δεν εξυπηρετεί διακοπές, τότε απλώς τις αγνοεί. Το περιφερειακό που έχει ζητήσει μια διακοπή διατηρεί τη λογική στάθμη της γραμμής της διακοπής σε υψηλή τιμή, μέχρι να εξυπηρετηθεί, οπότε δεν χάνονται διακοπές. Βέβαια, αν ο επεξεργαστής διατηρεί για πολύ χρόνο απενεργοποιημένες τις διακοπές, τότε μπορεί να καθυστερήσει σημαντικά να εξυπηρετήσει μια διακοπή και να γίνει κάποια υπερχείλιση ή κάποιο άλλο μη επιθυμητό γεγονός.

Μια βελτίωση που έχει γίνει αφορά τη χρήση ενός ειδικού συνεπεξεργαστή άμεσης πρόσβασης στη μνήμη (direct memory access, DMA). Αυτός μπορεί να γίνει κυρίαρχος του διαύλου και να μεταφέρει από ή προς τη μνήμη από οποιαδήποτε συνδεδεμένη συσκευή. Ο επεξεργαστής έχει άμεση πρόσβαση και ρυθμίζει τους καταχωρητές του DMA, και συγκεκριμένα το μέγεθος της μεταφοράς, τη διεύθυνση μνήμης, τη διεύθυνση του περιφερειακού και αν είναι εγγραφή ή ανάγνωση, του παραχωρεί την κυριότητα του διαύλου και ξεκινάει αυτόνομα τη μεταφορά των δεδομένων. Ο επεξεργαστής μπορεί να συνεχίσει να εκτελεί τον ίδιο ή άλλον κώδικα προγράμματος, και όταν ολοκληρωθεί η αντιγραφή θα ενημερωθεί με διακοπή από τον ελεγκτή DMA.



Με την παραχώρηση του ελέγχου στον ελεγκτή DMA μπορεί να εμφανιστεί και ένα πρόβλημα που ονομάζεται αντιστροφή των προτεραιοτήτων. Συγκεκριμένα, μπορεί να ρυθμιστεί μια μεταφορά DMA από τον επεξεργαστή και να του δοθεί ο έλεγχος, και καθώς γίνεται η μεταφορά, να δημιουργηθεί μια διακοπή υψηλής προτεραιότητας στον επεξεργαστή, που όμως επειδή δεν έχει τον έλεγχο του διαύλου δε μπορεί να την ικανοποιήσει. Με την εισαγωγή ενός ατομικού καναλιού επικοινωνίας του επεξεργαστή με το DMA, το πρόβλημα λύνεται, αφού μπορεί να δοθεί εντολή από τον επεξεργαστή να σταματήσει τη μεταφορά.

Τα περιφερειακά που χρησιμοποιούνται για E/E, υποστηρίζουν δυο προγραμματιστικούς τρόπους πρόσβασης. Ο πρώτος τρόπος ονομάζεται προγραμματιζόμενη E/E και οι διευθύνσεις που αντιστοιχούν σε κάθε περιφερειακό βρίσκονται σε ένα ανεξάρτητο χώρο διευθύνσεων ως προς τη μνήμη. Σε περίπτωση που απαιτηθεί πρόσβαση σε μια τέτοια διεύθυνση ο επεξεργαστής χρησιμοποιεί ειδικές εντολές εισόδου και εξόδου. Για παράδειγμα, οι ενσωματωμένοι μικροεπεξεργαστές της Intel, χρησιμοποιούν τις εντολές *in* και *out*.

Ο δεύτερος τρόπος πρόσβασης ονομάζεται απεικόνιση στη μνήμη, και η διεύθυνση που αντιστοιχεί στο περιφερειακό, βρίσκεται μέσα στο χώρο διευθύνσεων μαζί με τις διευθύνσεις της μνήμης. Οι επεξεργαστές της Intel, υποστηρίζουν και αυτόν τον τρόπο πρόσβασης μέσω των τυπικών εντολών πρόσβασης διευθύνσεων μνήμης *mon*, δηλαδή δεν υπάρχουν άλλες εντολές, είτε γίνεται πρόσβαση στη μνήμη είτε σε ένα περιφερειακό. Σχεδόν όλοι οι επεξεργαστές χρησιμοποιούν αυτόν τον τρόπο πρόσβασης, και κάποιοι και τον πρώτο. Οι MIPS και οι ARM επεξεργαστές χρησιμοποιούν μόνο αυτόν τον τρόπο.

### 8.5.3 Τυπικοί Δίαυλοι Ενσωματωμένων Συστημάτων

Κατά τα 30 τελευταία χρόνια τα ενσωματωμένα συστήματα έχουν ακολουθήσει τις εξελίξεις των σταθερών υπολογιστών στην επικοινωνία και έχουν υιοθετήσει πολλές φορές διαύλους που αρχικά είχαν προταθεί για τους προσωπικούς υπολογιστές. Η πρώτη σοβαρή πρόταση προήλθε από την IBM το 1980, όπου αφορούσε τον πρότυπο διάυλο 8bit της βιομηχανίας, ISA (Industry Standard Association). Χρησιμοποιούνταν στους πρώτους προσωπικούς υπολογιστές της IBM και στη συνέχεια από όλους τους κατασκευαστές που παρείχαν συμβατούς υπολογιστές με IBM (IBM PC compatible). Ο διάυλος γρήγορα επεκτάθηκε σε 16bit και αργότερα σε 32bit το 1988 με την ονομασία επεκταμένος ISA (extended ISA). Η IBM προσπάθησε στη συνέχεια να χρησιμοποιήσει ένα νέο δικό της διάυλο, που δεν έδινε όμως δικαιώματα χρήσης στους άλλους το MCA (microchannel architecture).

Λόγω των δικαιωμάτων χρήσης, οι υπόλοιποι κατασκευαστές παρέμειναν με το EISA και πρότειναν δυο νέες εκδόσεις: το PCI (peripheral component interconnect) και το VESA local bus. Ο Πίνακας παρουσιάζει τους πιο δημοφιλείς διαύλους που έχουν χρησιμοποιηθεί από το 1980 έως σήμερα. Μπορεί κάποιος να παρατηρήσει ότι με την πρόοδο της τεχνολογίας αποσύρονται οι παράλληλοι διάυλοι και χρησιμοποιούνται οι σειριακοί. Το πιο χαρακτηριστικό γεγονός είναι η αντικατάσταση των παράλληλων διαύλων IDE (αργότερα μετονομάστηκαν σε PATA, Parallel ATA) που είχαν μέγιστη ταχύτητα μεταφοράς 133 MB/sec, από τους σειριακούς διαύλους SATA (serial ATA), με μέγιστη ταχύτητα μεταφοράς περίπου 2GB/sec.

### Τυπικοί Δίαυλοι Συστήματος

Δίαυλος	bits	Mhz	Ταχύτητα (MB/sec)
8-bit ISA	8	8	4
16-bit ISA	16	8	8
EISA	32	8.33	33.3
32-bit PCI	32	33	133
64-bit PCI	64	33	266
1x AGP	64	66	266
8x AGP	64	533	2100
VL BUS	33	50	132
SCSI I/II	8	5	40
FAST SCSI	8	10	80
Wide SCSI	16	10	60
Ultra SCSI	16	20	320
PCIe x1	1	2500	>500
RAMBUS	32	1066	4200
IDE / PATA	16	66	133
SATA 1.0	1	1500	150
SATA 2.0	1	-	300
SATA 3.0	1	-	600
SATA 3.2	1	-	1969

Οι δίαυλοι ISA και EISA ήταν από τους μακροβιότερους διαύλους, που οφείλονταν στην μεγάλη χρήση και ευκολία ανάπτυξης εκείνα τα χρόνια. Όμως, είχαν εγγενή προβλήματα που δεν επέτρεπαν την εύκολη χρήση τους, ιδιαίτερα από χρήστες που δε γνώριζαν εσωτερικές λεπτομέρειες. Προκειμένου να λειτουργήσει ένα περιφερειακό θα έπρεπε να αναφέρει τη γραμμή διακοπής, τη διεύθυνση επικοινωνίας και το κανάλι DMA που έπρεπε να λειτουργήσει, στοιχεία που χαρακτήριζαν μοναδικά το περιφερειακό. Αν ένα άλλο περιφερειακό τοποθετούνταν στον κοινό δίαυλο με ίδιες παραμέτρους, τότε δυσλειτούργουσε όλο το σύστημα. Επίσης, η χρήση της πολυπλεξίας διευθύνσεων και δεδομένων μείωνε το μέγιστο ρυθμό επικοινωνίας. Τέλος, η διευθυνσιοδότηση στο δίαυλο ήταν 20bit και άρα μπορούσαν να υπάρχουν διευθύνσεις από 1 έως 1MB. Αυτά τα προβλήματα ώθησαν τους σχεδιαστές να βρουν ένα νέο δίαυλο. Αυτός ήταν ο PCI.

Ο δίαυλος PCI προτάθηκε αρχές του 1990 με το πιο σημαντικό χαρακτηριστικό της αυτόματης ρύθμισης των παραμέτρων επικοινωνίας των περιφερειακών. Είναι ο πιο κοινός δίαυλος και βρίσκεται ακόμη στις κεντρικές πλακέτες των περισσότερων συστημάτων, αν και τείνει να αντικατασταθεί και να εξαιρεθεί από νέους διαύλους. Σε αντίθεση με το ISA που λειτουργούσε στα 8Mhz, αυτός ο 32bit δίαυλος λειτουργεί στα 33/66Mhz με υποστήριξη τάσεων 3.3V και 5V, ενώ επιτυγχάνονταν εύρος ζώνης 524MB/sec για 64bit μεταφορές. Όπως και στον ISA έτσι και εδώ υπάρχει πολυπλεξία των διευθύνσεων και των δεδομένων για εξοικονόμηση ακροδεκτών, αλλά λόγω της μεγάλης συχνότητας λειτουργίας δεν αποτελεί πρόβλημα. Κάθε συσκευή παρέχει κάποιους καταχωρητές κατάστασης και ελέγχου, όπου έχει

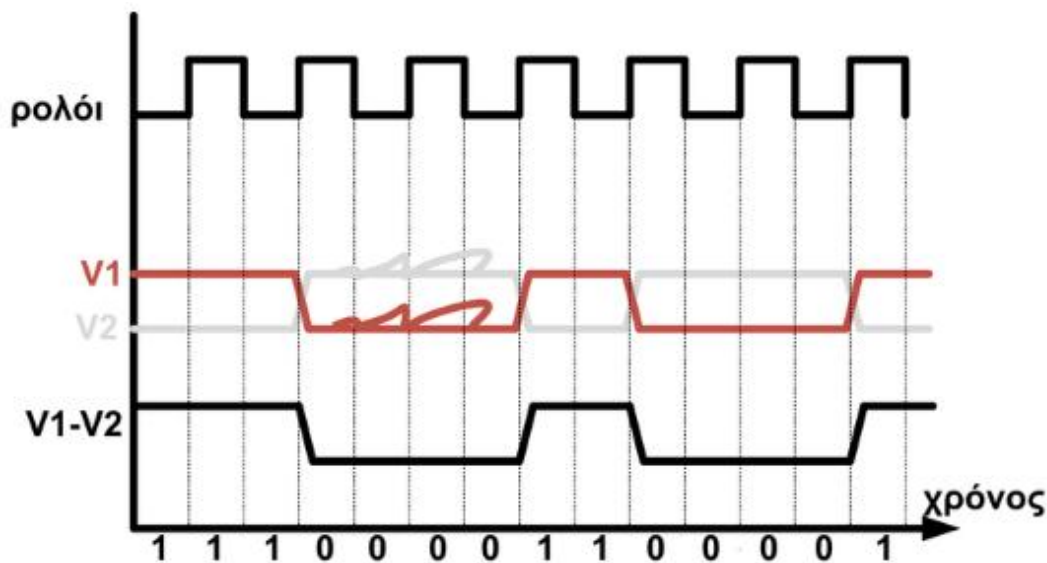
πρόσβαση ο επεξεργαστής τόσο για να κάνει την αυτόματη ρύθμιση, όσο και για να χειριστεί το περιφερειακό.

Προκειμένου να μειωθεί το πλάτος των αγωγών ο παράλληλος δίαυλος PCI εξελίχθηκε σε ένα σειριακό υβριδικό δίαυλο με δυνατότητα παραλληλίας με την ονομασία PCIe (PCI express) (Εικόνα 5.7). Συγκεκριμένα, αυτός ο δίαυλος αποτελείται από κανάλια επικοινωνίας (ονομάζονται lanes) και είναι τύπου σημείο-προς-σημείο (δηλαδή, μια συσκευή PCIe συνδέεται σε μια άλλη PCIe και δε γίνεται να συνδεθεί και τρίτη συσκευή. Το κάθε κανάλι επιτρέπει την αμφίδρομη αποστολή και λήψη δεδομένων από δυο γραμμές, με κατάλληλη διαφορική κωδικοποίηση, δηλαδή σειριακά. Ένα περιφερειακό μπορεί να σχεδιαστεί για να χρησιμοποιήσει πολλαπλά κανάλια (π.χ. 4 κανάλια και ονομάζεται x4, με μέγιστο τα 32 κανάλια), οπότε τα κανάλια χρησιμοποιούνται παράλληλα, αλλά η πρωταρχική επικοινωνία παραμένει σειριακή.

Ένα θετικό χαρακτηριστικό αυτού του διαύλου, είναι ότι περιφερειακά που χρησιμοποιούν λιγότερα κανάλια επικοινωνίας (π.χ. 4 κανάλια), μπορούν να συνδεθούν σε μεγαλύτερες υποδοχές επέκτασης πολλών καναλιών (π.χ. υποδοχή x8), και να λειτουργήσουν χωρίς κανένα πρόβλημα. Τα δεδομένα πριν σταλούν σε ένα κανάλι συσκευάζονται σε πακέτα. Οπότε, σε περίπτωση που υπάρχουν πολλαπλά κανάλια, το κάθε πακέτο προς αποστολή προωθείται σε οποιοδήποτε κανάλι είναι ελεύθερο. Συνήθως, όλα τα κανάλια από κάθε συσκευή καταλήγουν στις αντίστοιχες θύρες ενός διακόπτη μεταγωγής (switch), που συνδέεται στο κεντρικό δίκτυο (root complex), δημιουργώντας μια ιεραρχική δομή.

Οι δίαυλοι PCIe είναι αρκετά διαδεδομένοι και χρησιμοποιούνται σε κάθε υπολογιστικό σύστημα. Μπορεί να χρησιμοποιηθούν και σε ενσωματωμένα συστήματα, όταν υπάρχει ανάγκη για αυξημένο εύρος ζώνης, αλλά τις περισσότερες φορές επιλέγονται άλλοι δίαυλοι πιο οικονομικοί στους οποίους μπορούν να συνδεθούν άμεσα και εξωτερικά περιφερειακά. Ο δίαυλος PCIe χρησιμοποιείται κυρίως για τα εσωτερικά περιφερειακά ενός συστήματος, δηλαδή για αυτά που βρίσκονται άμεσα συνδεδεμένα στην κεντρική πλακέτα του συστήματος.

Ένα χαρακτηριστικό του διαύλου PCI express, είναι η υποστήριξη διαφορικής σηματοδότησης χαμηλής τάσης (low voltage differential signalling, LVDS), η οποία μπορεί να υποστηρίξει έως και 2 Gbit/s με κατανάλωση ισχύος κάποια mW. Η κωδικοποίηση LVDS αφορά το φυσικό επίπεδο, και έχει υιοθετηθεί από πολλά πρωτόκολλα διαύλων. Σε αυτή τη σηματοδότηση χρησιμοποιούνται 2 καλώδια για τη ταυτόχρονη σειριακή μετάδοση των bit, και η αποστολή των bit βασίζεται στη διαφορά δυναμικού ανάμεσα τους. Αν η διαφορά είναι μεγαλύτερη από κάποια τιμή, τότε ανιχνεύεται ως '1' διαφορετικά ως '0'. Η διαφορική σηματοδότηση επιτρέπει την μεγάλη ανοχή στο θόρυβο, αφού αν υπάρχει κάποια ηλεκτρομαγνητική ακτινοβολία θα επηρεάσει ταυτόχρονα και τους 2 αγωγούς που έχουν παράλληλη όδευση. Οι τάσεις που χρησιμοποιούνται είναι συνήθως 0.3 V και αυτό επιτρέπει την επίτευξη μεγάλης ταχύτητας αλλαγής από 0 σε 1 και 1 σε 0, με συνέπεια την επίτευξη εύρους ζώνης των Gbit.



Σχήμα 8.5.3: Η διαφορική σηματοδότηση LVDS, που χρησιμοποιείται στο PCIe, επιτρέπει την απροβλημάτιστη μεταφορά bit ακόμη και αν εμφανιστεί θόρυβος.

Προς τα μέσα της δεκαετίας του 1990, η εταιρία ARM εισήγαγε το δίαυλο Advanced Microcontroller Bus Architecture (AMBA), στις σχεδιάσεις εντός ολοκληρωμένου κυκλώματος για τη διασύνδεση των διαφόρων δομοστοιχείων που συνθέτουν ένα σύστημα πάνω στο chip. Ο AMBA χρησιμοποιείται σε κάθε σχεδίαση που χρησιμοποιεί τον επεξεργαστή ARM (π.χ. στο 90% των έξυπνων κινητών τηλεφώνων) και επιτρέπει πολλαπλούς επεξεργαστές και μεγάλο αριθμό περιφερειακών. Η ARM αν και κράτησε την κυριότητα, δημοσίευσε τις προδιαγραφές και έτσι υιοθετήθηκε από πολλούς κατασκευαστές και εκτός του οικοσυστήματος της ARM (π.χ. σε σχεδιασμούς ASIC). Η τρέχουσα έκδοση του AMBA είναι η 5 και παρουσιάστηκε το 2013. Θεωρείται το de facto standard για τη σχεδίαση ενσωματωμένων συστημάτων SoC, επειδή έχει πολύ καλή υποστήριξη από πολλούς παρόχους και επειδή έχει πολύ καλή τεκμηρίωση. Ο δίαυλος AMBA χωρίζεται σε υποκατηγορίες, αναλόγως του είδους των συσκευών που συνδέονται. Υπάρχει ο δίαυλος υψηλών ταχυτήτων AHB/AMBA που υποστηρίζει έως και 128bit, ο δίαυλος χαμηλών ταχυτήτων που συνδέονται περιφερειακά χαμηλών απαιτήσεων σε εύρος ζώνης, και δίαυλοι ειδικών προδιαγραφών, όπως αποσφαλμάτωσης, διατήρησης συνέπειας μνήμης κ.ο.κ. Από τις πρώτες εκδόσεις ο δίαυλος δημιουργήθηκε στοχεύοντας στην απόδοση· υποστηρίζει διασωληνωμένες μεταφορές, μεταφορές ριπής, αποδεσμευμένες μεταφορές και πολλαπλούς κυρίους των διαύλων.

Ένας αρκετά κοινός σειριακός δίαυλος που δημιουργήθηκε το 1994 από την κοινοπραξία 7 εταιριών (όπως IBM, Microsoft, και άλλες), είναι ο ενιαίος σειριακός δίαυλος επικοινωνίας (universal serial bus, USB). Στις προδιαγραφές του καθορίζονται τα καλώδια, οι συνδετήρες και τα πρωτόκολλα επικοινωνίας. Ως τώρα έχουν παρουσιαστεί αρκετές εκδόσεις, με την τελευταία να φέρει τον αριθμό 3.1 (Πίνακας 5.2). Η ιδιαιτερότητα αυτού του διαύλου και το πιο σημαντικό πλεονέκτημα που οδήγησε στην ευρεία αποδοχή του, είναι ότι εκτός από τα δεδομένα μεταφέρει

και ενέργεια (από 500mA έως 3A στα 5 Volt) από τον υπολογιστή προς τα περιφερειακά.

Στο δίαυλο μπορούν να συνδεθούν πολλαπλές συσκευές μέσω ενός USB διανομέα (hub). Στα αξιοσημείωτα χαρακτηριστικά του USB είναι η τοποθέτηση και άμεση χρήση (plug and play) χωρίς να απαιτείται η ρύθμιση παραμέτρων επικοινωνίας από το χρήστη, η τοποθέτηση και απομάκρυνση του περιφερειακού, ενώ ο υπολογιστής είναι σε λειτουργία (hot pluggable), η υποστήριξη πολλών ταχυτήτων, ώστε να μπορεί να χρησιμοποιηθεί τόσο από συσκευές χαμηλού εύρους ζώνης (όπως το πληκτρολόγιο), όσο και από συσκευές υψηλού εύρους ζώνης (π.χ. σκληρός δίσκος ή κάρτα γραφικών) και η χρήση καλωδίων διαφορετικών τύπων που δεν επιτρέπουν τη λανθασμένη συνδεσμολογία περιφερειακών USB στο διανομέα USB.

Ο δίαυλος USB χρησιμοποιεί την περιόδευση των περιφερειακών, και κάθε φορά μόνο ένα περιφερειακό μπορεί να επικοινωνήσει με τον υπολογιστή· έτσι, δε μπορούν να προκληθούν συγκρούσεις. Το μέγιστο μήκος σύνδεσης είναι 5m, επειδή μετά υπάρχει εξασθένηση των σημάτων ή αυξημένα επίπεδα θορύβου που αλλοιώνουν τα σήματα. Ο δίαυλος USB είναι πολύ δημοφιλής στα ενσωματωμένα συστήματα, επειδή δίνει τη δυνατότητα της διασύνδεσης με μιας πληθώρας περιφερειακών. Ακόμη και οι πιο οικονομικοί υπολογιστές μιας πλακέτας των 5\$, όπως το Raspberry pi zero, διαθέτουν διεπαφή USB, αν έχουν εξαλείψει άλλους ακροδέκτες που ως τώρα θεωρούνταν σημαντικοί (όπως του δικτύου LAN).

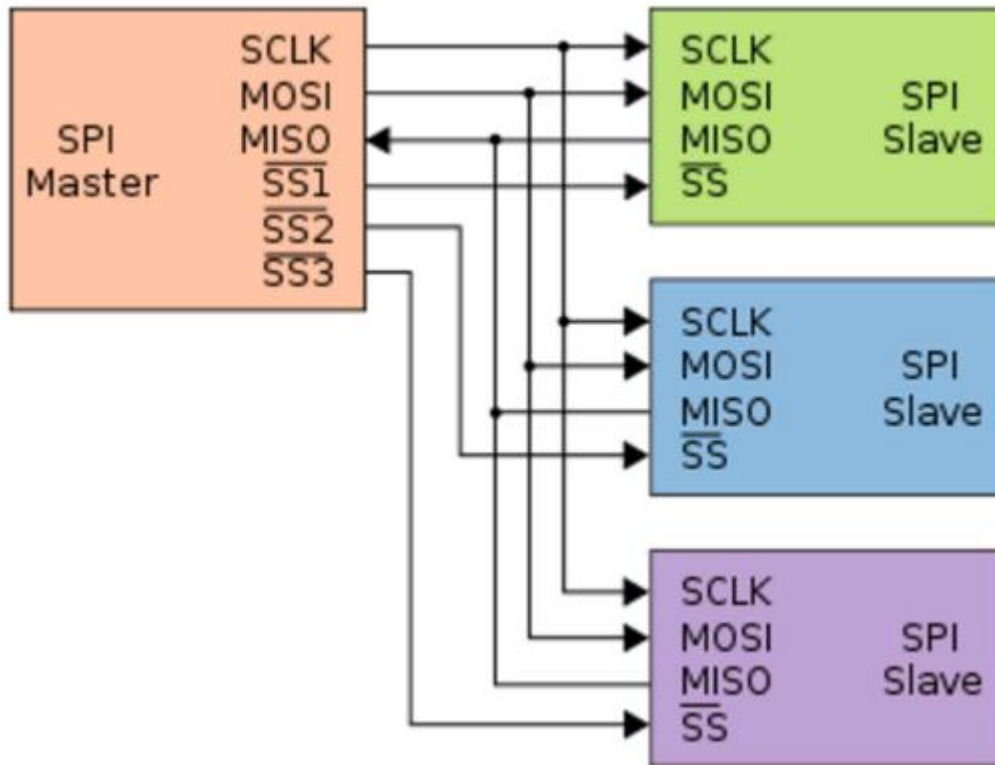
### Εκδόσεις του USB

Έκδοση	Ημερομηνία	Ταχύτητα
0.9	Απρίλιος 1995	1.5 Mbit/s
1.0	Ιανουάριος 1996	1.5 Mbit/s και 12 Mbit/s
2.0	Απρίλιος 2000	480 Mbit/s
3.0	Νοέμβριος 2008	5 Gbit/s
3.1	Ιούλιος 2013	10 Gbit/s

Ένα άλλο πρωτόκολλο που αναπτύχθηκε παράλληλα με το USB από την εταιρία Apple αρχές του 1990 είναι το Firewire, το οποίο έχει προτυποποιηθεί κατά IEEE 1394. Είναι ένα σειριακό πρωτόκολλο όπως το USB με τις παρακάτω διαφορές: (α) οι συσκευές συνδέονται στο Firewire σε αλυσίδα, δηλαδή η κάθε μια με τις διπλανές της, και μια μόνο συνδέεται στον υπολογιστή, (β) επιτρέπει τη χρήση του DMA, ώστε να μεταφέρει η μια συσκευή δεδομένα σε μια άλλη χωρίς τη διαμεσολάβηση του επεξεργαστή (όπως γίνεται στο USB), (γ) μπορεί και παρέχει πολύ περισσότερη ενέργεια στις συσκευές που διασυνδέονται (έως 30V σε 1.5A), (δ) κάθε έκδοση Firewire έχει διαφορετικό συνδετήρα (όλες οι εκδόσεις του USB έχουν συμβατούς συνδετήρες), (ε) η ταχύτητα της τελευταίας έκδοσης Firewire 800 είναι πολύ καλύτερη από την ταχύτητα του USB 2.0, αλλά είναι πιο χαμηλή από την ταχύτητα του USB 3.0. Το Firewire υποστηρίζεται κυρίως από την Apple, και η τελευταία έκδοση ονομάζεται thunderbolt, αλλά με την έλευση του USB 3.1 το μέλλον για αυτό το δίαυλο φαίνεται δυσοίωνο.

Στις τυπικές διεπαφές εισόδου εξόδου βρίσκουμε και τη σειριακή διασύνδεση RS232 με μέγιστη ταχύτητα 115200 bps σε 12V. Αυτή η διασύνδεση είναι πολύ χρήσιμη στην ανάπτυξη ενσωματωμένων συστημάτων, γιατί μπορεί να επιτευχθεί ακόμη και με 2 καλώδια αποστολής και λήψης. Χαρακτηριστικό είναι ότι κατά την ανάπτυξη προγραμμάτων σε FPGA ή σε μικροεπεξεργαστές, οι σχεδιαστές εκτυπώνουν στη σειριακή θύρα μηνύματα αποσφαλμάτωσης. Αν και οι σημερινοί σταθμοί εργασίας δεν έχουν θύρα RS232, εντούτοις υπάρχουν μετατροπείς USB-RS232 που καλύπτουν αυτή την έλλειψη. Η θύρα αυτή υπάρχει ακόμη στους διακομιστές που επιτρέπουν την εμφάνιση διαγνωστικών μηνυμάτων του βασικού συστήματος εισόδου εξόδου (Basic Input Output System, BIOS) πριν φορτωθεί το λειτουργικό σύστημα. Υπάρχουν και άλλα σειριακά πρωτόκολλα, όπως το RS422 για αποστάσεις έως 1.5 Km και χρησιμοποιείται για το βιομηχανικό αυτοματισμό, το RS485 που είναι παρόμοιο με το RS422 αλλά επιτρέπει πολλαπλούς κυρίους του διαύλου, το ps/2 που χρησιμοποιήθηκε στο παρελθόν για τη διασύνδεση πληκτρολογίων και ποντικών, αλλά αντικαθίσταται από τη διασύνδεση USB, και η παράλληλη θύρα 25 ακίδων IEEE1284 στην οποία συνδέονται εκτυπωτές, αλλά με την έλευση του USB αντικαταστάθηκε από μικρούς και ευέλικτους ακροδέκτες USB.

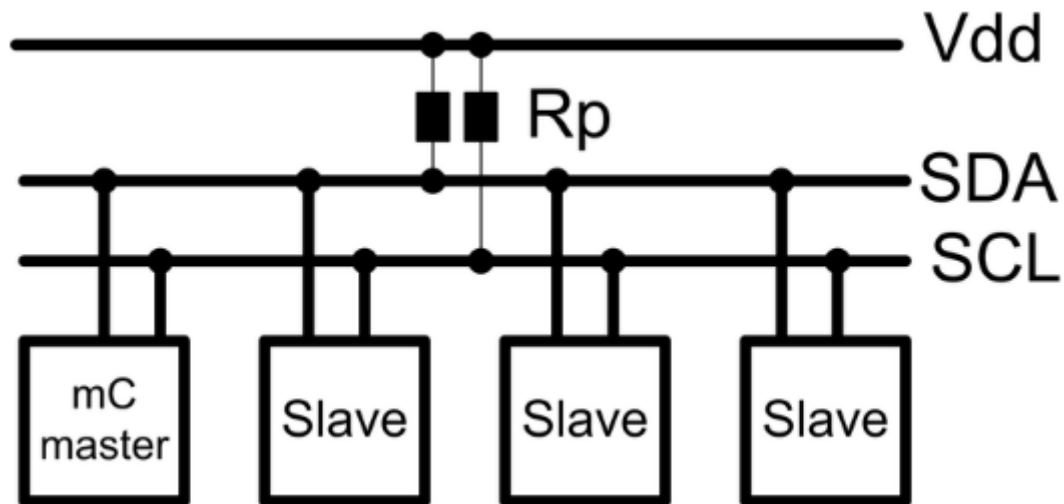
Στα ενσωματωμένα συστήματα χρησιμοποιούνται επίσης και πιο απλοί διάυλοι, που φέρουν ονομασίες όπως 4-wire, 3-wire, 2-wire, 1-wire ανάλογα με τον αριθμό των καλωδίων που χρησιμοποιούν για να συνδέσουν τα περιφερειακά στον επεξεργαστή. Ένας διάυλος 4-wire είναι ο SPI (Serial Peripheral Interface, σειριακή περιφερειακή διεπαφή) και χρησιμοποιεί 4 καλώδια: ρολόι, επιλογή περιφερειακού, εγγραφή και ανάγνωση (Εικόνα 5.9). Η μέγιστη συχνότητα λειτουργίας είναι 20Mhz, ταχύτητα η οποία ικανοποιεί τις απαιτήσεις μεταφοράς τιμών αισθητήρων σε κάποιον μικροεπεξεργαστή ή τον έλεγχο κάποιων περιφερειακών. Ένας ακόμη διάυλος 4 καλωδίων είναι ο SSI (Synchronous Serial Interface, σειριακή σύγχρονη διεπαφή), που βασίζεται στο RS-422 και χρησιμοποιεί 2 καλώδια για το ρολόι και 2 για τα δεδομένα. Μια παραλλαγή του SPI χρησιμοποιεί 3 καλώδια (3-wire) όπου υπάρχει μόνο μια γραμμή για την εγγραφή και ανάγνωση (αμφίδρομη επικοινωνία).



Σχήμα 8.5.3 (Εικόνα 1): Ο διάυλος SPI με 3 περιφερειακά.

Ο διάυλος με 2 καλώδια ονομάζεται I<sup>2</sup>C και υποστηρίζει πολλαπλούς κυρίους, πολλαπλά περιφερειακά, και έχει αναπτυχθεί από την εταιρία Philips. Ο διάυλος αυτός υποστηρίζει διάφορες ταχύτητες επικοινωνίας από 100 Khz έως 5 Mhz. Χρησιμοποιεί κατάλληλες αντιστάσεις pull-up που συνδέουν την κάθε γραμμή (ρολόι και δεδομένα) με την τροφοδοσία (Εικόνα 5.10). Πάνω στις προδιαγραφές αυτού του διαύλου έχει αναπτυχθεί ο διάυλος SMB με πολύ πιο αυστηρές οδηγίες για να επιτευχθεί σταθερότητα και αξιοπιστία. Οι σημερινοί υπολογιστές χρησιμοποιούν το SMB προκειμένου να διαβάσει ο επεξεργαστής τις τιμές από τα αισθητήρια πάνω στην κεντρική πλακέτα, όπως θερμοκρασίας και ταχύτητας ανεμιστήρα ψήκτρας. Το I<sup>2</sup>C είναι ένα αρκετά διαδεδομένο πρωτόκολλο στα ενσωματωμένα συστήματα που διαβάζουν τιμές από αισθητήρια, αφού χρησιμοποιεί ελάχιστα καλώδια και άρα έχει μειωμένο κόστος.

Ο διάυλος με 1 καλώδιο έχει προταθεί από την Dallas Semiconductor και χρησιμοποιεί το ίδιο καλώδιο για δεδομένα χαμηλού εύρους ζώνης, σηματοδότηση και ρολόι. Η απόσταση που υποστηρίζεται είναι πολύ μεγαλύτερη από τον I<sup>2</sup>C. Τα περιφερειακά του 1-wire έχουν ένα πυκνωτή που συγκεντρώνει το φορτίο και μπορεί να τα τροφοδοτεί με ενέργεια τα χρονικά διαστήματα που δεν παρέχεται ενέργεια από τη μια γραμμή. Συνήθως χρησιμοποιείται από αισθητήρια (π.χ. θερμοκρασίας ή υγρασίας) που έχουν ελάχιστες ανάγκες σε εύρος ζώνης.



Σχήμα 8.5.3 (Εικόνα 2): Ο διάυλος I<sup>2</sup>C χρησιμοποιεί μόνο 2 καλώδια για την επικοινωνία.

Ένας άλλος διάυλος είναι ο CAN (controller area network, περιοχή δικτύου ελεγκτών), ο οποίος έχει προτυποποιηθεί και χρησιμοποιείται στην αυτοκινητοβιομηχανία. Χρησιμοποιεί μηνύματα και επιτυγχάνει καλή απόδοση λόγω χρονικής πολυπλεξίας. Η ανάπτυξη του ξεκίνησε το 1983 από τη Bosch GmbH, και χρησιμοποιήθηκε για πρώτη φορά το 1988 στη BMW 8. Είναι ένα σειριακός διάυλος πολλαπλών κυρίων, πολλαπλών κόμβων, και υποστηρίζει από απλά περιφερειακά, έως σύνθετους σταθμούς επεξεργασίας δεδομένων. Όλοι οι κόμβοι συνδέονται με 2 καλώδια μεταξύ τους και έχουν τις κατάλληλες τερματικές αντιστάσεις των 120 Ω. Αν και δεν υπάρχει σήμα ρολογιού, εντούτοις όλοι οι κόμβοι είναι συγχρονισμένοι μεταξύ τους και έχουν ρυθμιστεί να δειγματοληπτούν τα bit όλοι, την ίδια στιγμή. Αυτός ο διάυλος χρησιμοποιείται μόνο στα ενσωματωμένα συστήματα των οχημάτων, πλοίων ή αεροπλάνων.

Εκτός από αυτούς τους διαύλους, υπάρχουν και πρωτόκολλα ασύρματης επικοινωνίας, όπως το bluetooth, Wifi, Wi-MAX, 3G και άλλα τα οποία συνδέονται τις περισσότερες φορές μέσω USB με τον επεξεργαστή, και έτσι δεν είναι ξεχωριστοί διάυλοι. Ιδιαίτερα, τον τελευταίο καιρό έχουν σχεδιαστεί πολύ οικονομικά περιφερειακά διασύνδεσης με το διαδίκτυο, όπως το ESP8266 που κοστίζει 5\$, απαιτεί τάση 3.3 V, είναι συμβατό με 802.11b και επικοινωνεί μέσω σειριακής σύνδεσης 2 αγωγών με κάποιον επεξεργαστή. Σε παρόμοια τιμή μπορεί κάποιος να προμηθευτεί ένα περιφερειακό Bluetooth ή το περιφερειακό nRF24L01 για επικοινωνία στα 2.4 Ghz σε μεγάλες αποστάσεις χωρίς οπτική επαφή, δημιουργώντας ένα ασύρματο καταναμημένο ενσωματωμένο σύστημα. Η προσθήκη ασύρματης επικοινωνίας στα ενσωματωμένα συστήματα, προσδίδει μια ελευθερία και ευελιξία, ιδιαίτερα όταν συνδέονται στο διαδίκτυο, κατασκευάζοντας το IoT (internet of things, διαδίκτυο των συσκευών). Όμως, η ελευθερία αυτή δεν έχει μόνο πλεονεκτήματα. Το πιο σοβαρό μειονέκτημα είναι η ασφάλεια, αφού δίνει τη δυνατότητα σε κακόβουλους χρήστες για απομακρυσμένη επίθεση στο ενσωματωμένο σύστημα. Η ασφάλεια των ενσωματωμένων συστημάτων είναι ένα πολύ σημαντικό θέμα που θα πρέπει να απασχολεί κάθε σχεδιαστή ΕΣ.



## 8.6 Μικροεπεξεργαστές και Μικροελεγκτές

Μερικές φορές ένα ενσωματωμένο σύστημα επιτελεί μια απλή λειτουργία που διαβάζει τιμές από αισθητήρες και ενεργοποιεί αντίστοιχες συσκευές δράσης. Τέτοια συστήματα δεν έχουν υψηλές υπολογιστικές απαιτήσεις και μπορούν να χρησιμοποιήσουν μικροελεγκτές ή μικροεπεξεργαστές. Ο μικροελεγκτής είναι ένα ολοκληρωμένο κύκλωμα που μπορεί να παρέχει αρκετά περιορισμένες υπολογιστικές ικανότητες, αφού συνήθως λειτουργεί σε ταχύτητες κάποιων Mhz και έχει κάποια KB μνήμης RAM, εντούτοις παρέχει αρκετά χαρακτηριστικά που είναι χρήσιμα σε συγκεκριμένες περιπτώσεις. Αυτό το στοιχείο είναι παρόμοιο με έναν επεξεργαστή (ή μικροεπεξεργαστή), δηλαδή έχει την ικανότητα να εκτελεί αριθμητικές και λογικές πράξεις, να μεταφέρει δεδομένα εντός και εκτός του ολοκληρωμένου κυκλώματος και να μπορεί να εκτελέσει ένα πρόγραμμα.

Τα σημαντικότερα χαρακτηριστικά του Μικροελεγκτή είναι τα παρακάτω:

- Έχει χαμηλό κόστος που είναι πολύ σημαντικό για ενσωματωμένα συστήματα που πρόκειται να κατασκευαστούν σε χιλιάδες ή εκατομμύρια κομμάτια.
- Περιέχει έναν αριθμό από διάφορες ενσωματωμένες λειτουργικές μονάδες (ή περιφερειακές συσκευές) στο ίδιο υπόστρωμα πυριτίου με τον επεξεργαστή.

Αυτές οι μονάδες παρέχουν χρήσιμες δυνατότητες για τα Ενσωματωμένα Συστήματα, όπως μέτρηση χρόνου με χρονιστές, μέτρηση συμβάντων με μετρητές, υποστήριξη διάφορων τρόπων επικοινωνίας, όπως σειριακή .

Οι επεξεργαστές που χρησιμοποιούνται στα ενσωματωμένα συστήματα μπορεί να είναι είτε οι κλασσικοί μικροεπεξεργαστές γενικού σκοπού είτε οι μικροελεγκτές, οι οποίοι διαθέτουν πάνω σε ένα ολοκληρωμένο κύκλωμα περισσότερα περιφερειακά, μειώνοντας έτσι το κόστος και το μέγεθος της κατασκευής.

Το παρακάτω σχήμα 2.3 δείχνει σχηματικά το βρόχο ελέγχου που εκτελεί ένας μικροελεγκτής, σε μια τυπική ενσωματωμένη εφαρμογή. Ο μικροελεγκτής λειτουργεί ως ψηφιακός ελεγκτής συλλέγοντας δεδομένα από τους αισθητήρες και ενεργοποιώντας εξωτερικά κυκλώματα, όπως κινητήρες, ηλεκτρονόμους, κυκλώματα ισχύος κλπ. Αντίστοιχα με όσα περιγράψαμε στην παράγραφο 1.2.1, ο μικροελεγκτής λαμβάνει ως είσοδο την τρέχουσα κατάσταση του συστήματος που ελέγχει, συνήθως με τη μορφή ενός αναλογικού σήματος, που προέρχεται από έναν ή περισσότερους αισθητήρες. Το σήμα αυτό ψηφιοποιείται, με τη βοήθεια ενός μετατροπέα αναλογικού σήματος σε ψηφιακό και συγκρίνεται με μια τιμή αναφοράς, που περιγράφει την επιθυμητή κατάσταση του συστήματος.

Με βάση την απόκλιση της τρέχουσας κατάστασης από την επιθυμητή κατάσταση αναφοράς, λαμβάνεται μια απόφαση και παράγεται ένα σήμα «οδήγησης», που επιδρά πάνω στο ελεγχόμενο σύστημα και μεταβάλλει την κατάστασή του. Για το σκοπό αυτό, ο μικροελεγκτής διαθέτει κατάλληλες θύρες εισόδου/εξόδου, με τις οποίες διασυνδέεται με τον εξωτερικό κόσμο. Επίσης διαθέτει περιφερειακά, όπως

κυκλώματα παραγωγής παλμών μεταβαλλόμενου εύρους (PWM) για οδήγηση κινητήρων, μετατροπής αναλογικού σήματος σε ψηφιακό (ADC), θύρες επικοινωνίας κ.ά. Ο ψηφιακός ελεγκτής είναι συνήθως σύστημα πραγματικού χρόνου.

### 8.6.1 Μικροεπεξεργαστές

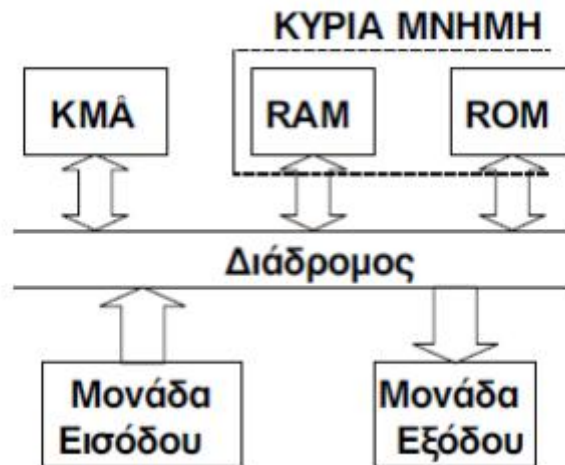
Οι πρώτοι μικροεπεξεργαστές εμφανιστήκαν στις αρχές τις δεκαετίας του 1970. Ένας μικροεπεξεργαστής περιλαμβάνει τις περισσότερες ή όλες τις λειτουργίες μιας κεντρικής μονάδας επεξεργασίας (ΚΜΕ ή CPU) ενός υπολογιστή σε ένα ενιαίο ολοκληρωμένο κύκλωμα (IC-Integrated Circuit). Για πρώτη φορά, είχαμε σημαντική επεξεργαστική ισχύ σε αρκετά χαμηλό κόστος και συγκριτικά μικρό χώρο. Αρχικά, χρησιμοποιήθηκαν σε ηλεκτρονικές αριθμομηχανές. Σταδιακά όμως, οι μικροεπεξεργαστές έγιναν πιο αυτόνομοι, για παράδειγμα, συμπεριλάμβαναν διάφορους τύπους μνήμης στο ίδιο τσιπ (chip) με τον επεξεργαστή. Την ίδια στιγμή, η CPU έγινε πιο ισχυρή, πιο γρήγορη και να προχώρησε ταχέως από 8-bit σε 16-bit και 32-bit συσκευές.



Σχήμα 8.6.1 (Εικόνα 1): Ο Πρώτος Μικροεπεξεργαστής (Intel 4004)

Η ανάπτυξη του μικροεπεξεργαστή οδήγησε ευθέως σε εφαρμογές όπως ο προσωπικός υπολογιστής (PC). Έτσι οι σύγχρονοι μικροεπεξεργαστές έχουν συχνότητα ρολογιού που κυμαίνεται από εκατοντάδες megahertz (MHz) έως αρκετά gigahertz (GHz). Παράλληλα, αυξήθηκε η πολυπλοκότητα και ο αριθμός των τρανζίστορ που αποτελούν ένα ολοκληρωμένο κύκλωμα. Ο ρυθμός αύξησης των τρανζίστορ περιγράφεται από τον νόμο του Μουρ (Moore's Law), που ισχύει μέχρι σήμερα και προβλέπει τον διπλασιασμό του αριθμού των τρανζίστορ, που ενσωματώνονται σε ένα ολοκληρωμένο κύκλωμα, κάθε 18 μήνες.

Με τον όρο υπολογιστικό σύστημα αναφερόμαστε σε ένα σύστημα το οποίο αποτελείται από μία κεντρική μονάδα επεξεργασίας (ΚΜΕ), μνήμη και συσκευές εισόδου και εξόδου. Στόχος όταν σχεδιάζουμε ένα τέτοιο σύστημα είναι να το χρησιμοποιήσουμε για την εκτέλεση υπολογισμών ή για να ελέγξουμε άλλες συσκευές.



Ένα υπολογιστικό σύστημα στο οποίο ως ΚΜΕ χρησιμοποιείται ένας μικροεπεξεργαστής ονομάζεται μικροϋπολογιστικό σύστημα. Ένας μικροεπεξεργαστής είναι ένα ολοκληρωμένο κύκλωμα γενικού σκοπού, το οποίο μπορεί να προγραμματιστεί. Η επεξεργασία των δεδομένων γίνεται με μια σειρά από εντολές. Οι εντολές που εκτελούνται από τον μικροεπεξεργαστή είναι εντολές σε γλώσσα μηχανής και είναι αποθηκευμένες στην κύρια μνήμη. Μια εντολή σε γλώσσα μηχανής είναι μια σειρά από δυαδικά ψηφία, όπου είναι κωδικοποιημένο το είδος της εντολής. Το σύνολο αυτών των εντολών χρησιμεύει ως μία διασύνδεση ανάμεσα στο λογισμικό (software) και το υλικό (hardware), δηλαδή ανάμεσα στα προγράμματα και στους επεξεργαστές. (Πεκμεστζή Κ.) Ένας μικροεπεξεργαστής είναι ένα ολοκληρωμένο κύκλωμα που μπορεί να προγραμματιστεί. Οι μικροεπεξεργαστές ξεκίνησαν ως μια ενδιάμεση φθηνή λύση στον έλεγχο συστημάτων μεταξύ υπολογιστών και του ειδικού υλικού που έπρεπε να κατασκευαστεί κατά περίπτωση. Η εμφάνισή τους άλλαξε τη σχεδίαση των υπολογιστικών συστημάτων απλοποιώντας την αρκετά. Το υλικό που βασίζεται σε μικροεπεξεργαστή μπορεί να χρησιμοποιηθεί χωρίς μετατροπές σε ποικιλία εφαρμογών, ενώ ένα ψηφιακό σύστημα που κατασκευάζεται με πύλες χρησιμοποιείται μόνο σε μία εφαρμογή. Επομένως με τους μικροεπεξεργαστές το πρόβλημα της σχεδίασης ψηφιακών συστημάτων μετατοπίστηκε από το υλικό στο λογισμικό και η σχεδίαση απαλλάχθηκε από την ακαμψία του υλικού. Ένας μικροεπεξεργαστής αποτελείται από την αριθμητική και λογική μονάδα, τη μονάδα ελέγχου και τους καταχωρητές. Η αριθμητική και λογική μονάδα εκτελεί πράξεις με δεδομένα τα οποία βρίσκονται σε καταχωρητές εκ των οποίων ο ένας ονομάζεται συσσωρευτής. Τα αποτελέσματα των πράξεων αποθηκεύονται στον συσσωρευτή. Η μονάδα ελέγχου ελέγχει και συγχρονίζει τη μεταφορά και την επεξεργασία των δεδομένων και αποτελεί το κύριο τμήμα του μικροεπεξεργαστή. Οι καταχωρητές χρησιμεύουν ως χώροι αποθήκευσης δεδομένων ή διευθύνσεων μνήμης. 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.

Επιπλέον ένας πιο σύγχρονος μικροεπεξεργαστής αποτελείται από τις ακόλουθες μονάδες:

- **Μονάδα αποκωδικοποίησης (Decoding Unit).**
- **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.
- **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.
- **Μονάδα ελέγχου (Control Unit):** Ελέγχει τη ροή δεδομένων από και προς την ALU, τους καταχωρητές, τη μνήμη και τις περιφερειακές μονάδες εισόδου/εξόδου.
- **Μονάδα προσκόμισης (Fetch Unit):** Μεταφέρει τις εντολές από τη μνήμη στον επεξεργαστή.
- **Μονάδα προστασίας (Protection Unit):** Εξασφαλίζει το αποδεκτό της κάθε διεργασίας που εκτελεί ο επεξεργαστής, ώστε να μη τροποποιούνται δεδομένα που δεν πρέπει ή να μην εκτελούνται μη αποδεκτές εντολές, όπως π.χ. διαίρεση αριθμού με το μηδέν. Τα κυριότερα χαρακτηριστικά των μικροεπεξεργαστών είναι η συχνότητα λειτουργίας, το μήκος της λέξης που αποθηκεύεται στους καταχωρητές και το ρεπερτόριο των εντολών, το οποίο δείχνει την ποικιλία των εντολών που μπορεί να δεχθεί και να επεξεργασθεί ένας μικροεπεξεργαστής. Ενδεικτικά όσες περισσότερες εντολές μπορεί να εκτελέσει τόσο πιο αργός είναι ο μικροεπεξεργαστής.

Οι μικροεπεξεργαστές είναι συχνά πολύ αποδοτικοί:

- Μπορούν να χρησιμοποιήσουν την ίδια λογική για την εκτέλεση πολλών διαφορετικών λειτουργιών.
- Κάνουν ευκολότερη τη σχεδίαση οικογενειών προϊόντων.
- Χρησιμοποιούν αποδοτικότερα τη λογική για την υλοποίηση μιας λειτουργίας από την προσαρμοσμένη λογική.
- Προσφέρουν δυνατότητες για να βοηθήσουν στον έλεγχο της κατανάλωσης ισχύος.

## 8.6.2 Μικροελεγκτές

Η επιθυμία των κατασκευαστών να δημιουργήσουν συστήματα με περισσότερες δυνατότητες και με μικρότερο μέγεθος οδήγησε στην ανάγκη για ενσωμάτωση όλων των λειτουργιών ενός υπολογιστή σε ένα ή μερικά ολοκληρωμένα κυκλώματα. Η ανάπτυξη της τεχνολογίας που χρησιμοποιήθηκε για την κατασκευή των κυκλωμάτων αυτών, είχε ως αποτέλεσμα τη βελτίωση των χαρακτηριστικών του μικροεπεξεργαστή και τη δημιουργία του μικροελεγκτή.

Ένα μικροϋπολογιστικό σύστημα αποτελείται από ένα μικροεπεξεργαστή, μνήμη και μονάδες εισόδου- εξόδου. Ένας μικροεπεξεργαστής μπορεί να εκτελεί υπολογισμούς και να επεξεργάζεται δεδομένα γρήγορα και αξιόπιστα και προκειμένου να επικοινωνήσει με το περιβάλλον του, πρέπει να συνδεθεί με κάποιες περιφερειακές συσκευές όπως μία μνήμη, έναν ελεγκτή διακοπών ή κάποιες θύρες επικοινωνίας με τις μονάδες εισόδου- εξόδου.

Επίσης ένα μικροϋπολογιστικό σύστημα μπορεί να χρησιμοποιηθεί για να ελέγξει ένα σύστημα. Σε αυτή την περίπτωση πρέπει να δεχθεί αναλογικά σήματα, όμως για να τα αξιοποιήσει ο μικροεπεξεργαστής πρέπει να παρεμβληθούν μετατροπείς από αναλογικό σήμα σε ψηφιακό. Επομένως όλες αυτές οι συσκευές χρειάζεται να είναι και μεταξύ τους συγχρονισμένες. Ένα τέτοιο κύκλωμα ονομάζεται μικροελεγκτής.

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.

Γενικά, ένας μικροελεγκτής είναι ένα μικρό υπολογιστικό κύκλωμα, σχεδιασμένο σε ένα και μόνο ολοκληρωμένο κύκλωμα υψηλής κλίμακας ολοκλήρωσης. Όπως κάθε υπολογιστικό κύκλωμα, περιέχει κεντρική μονάδα επεξεργασίας, έναν αριθμό καταχωρητών, κυκλώματα μνήμης και κυκλώματα ελέγχου περιφερειακών συσκευών. Κάθε μικροελεγκτής είναι λοιπόν ικανός να ανταλλάξει σήματα με το εξωτερικό περιβάλλον, να εκτελέσει πράξεις ανάμεσα σε μεταβλητές και να καταχωρήσει κάποιες τιμές στη μνήμη RAM που διαθέτει.

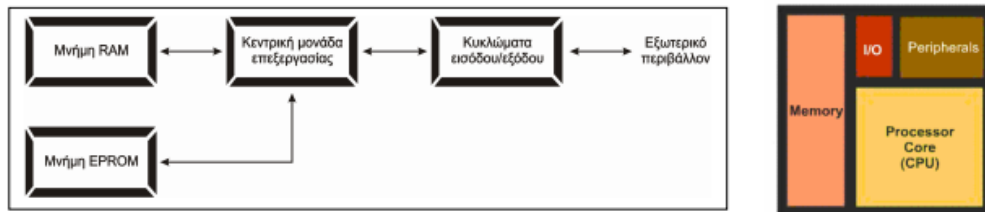
Κάθε μικροελεγκτής περιέχει μέσα σε ένα και μοναδικό ολοκληρωμένο κύκλωμα τα παρακάτω στοιχεία:

- Έναν αριθμό από καταχωρητές ειδικού σκοπού (συσσωρευτή, καταχωρητή κατάστασης, μετρητή προγράμματος, καταχωρητή εντολών, καταχωρητή δείκτη).
- Εσωτερικούς χρονιστές - απαριθμητές.
- Αριθμητική και λογική μονάδα (ALU).

- Μονάδα αποκωδικοποίησης εντολών.

Βασικά στοιχεία ενός μικροελεγκτή αποτελούν:

- Η μνήμη προγράμματος (ROM ή EPROM).
- Η μνήμη καταχωρητών / μεταβλητών (RAM).



Σχήμα 8.6.2: Βασική δομή ενός μικροελεγκτή

Στους μικροελεγκτές διακρίνουμε επίσης τα κυκλώματα χρονισμού και ελέγχου. Βασικά μέρη ενός μικροελεγκτή είναι

- παράλληλες θύρες εισόδου/εξόδου
- άλλα περιφερειακά κυκλώματα (UART, A/D μετατροπείς κλπ.)

Μέσα από τις θύρες I/O ένας μικροελεγκτής μπορεί να δέχεται σήματα εισόδου με τη μορφή λογικών ψηφιακών καταστάσεων, χαρακτήρες ή bytes δεδομένων με την τεχνική της ασύγχρονης ή της σύγχρονης σειριακής επικοινωνίας, σήματα διακοπών, ή σε ορισμένες περιπτώσεις και αναλογικά σήματα, τα οποία στη συνέχεια μετατρέπονται σε ψηφιακά. Επίσης μπορεί να αποστέλλει σήματα σε άλλες συσκευές μέσα από θύρες εξόδου, να οδηγεί ηλεκτρονόμους, διόδους LED και άλλα κατάλληλα κυκλώματα, που συνήθως περιλαμβάνονται σε κάθε μορφής αυτοματισμό.

Οι μικροελεγκτές χαρακτηρίζονται από ένα περιορισμένο ρεπερτόριο εντολών, οι οποίες μπορούν να γραφούν σε συμβολική μορφή (assembly), με τη βοήθεια μνημονικών ονομάτων. Στους μικροελεγκτές PIC μεσαίας τάξης (midrange), το μήκος της εντολής σε γλώσσα μηχανής είναι 14 bits, τα οποία καταχωρούνται στη μνήμη προγράμματος, τύπου EEPROM. Για τα εργαλεία που χρησιμοποιούνται για τον σκοπό αυτό θα μιλήσουμε σε επόμενη παράγραφο.

#### a) Συνήθη Υποσυστήματα Μικροελεγκτών

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.
- 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 οποίο 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 timercounter) 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).

#### **b) 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 π.χ. 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, I2C, 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 υποστηρίξει 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 τα μόντεμ.

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 μικροελεγκτών 8-bit έχουν πολύ βελτιωμένες επιδόσεις σε σχέση με το παρελθόν.

### **c) Μικροελεγκτής PIC**

Οι μικροεπεξεργαστές και οι μικροελεγκτές χωρίζονται σε δύο μεγάλες κατηγορίες, αυτών με την αρχιτεκτονική CISC και αυτών με RISC. Ο μικροελεγκτής που θα μελετήσουμε είναι ο PIC, της εταιρείας Microchip και είναι αρχιτεκτονικής RISC.

Στον πυρήνα του PIC ανήκουν:

- Κεντρική Μονάδα Επεξεργασίας
- Μνήμη
- Εντολές
- Λειτουργίες Διακοπών



Εδώ, πρέπει να σημειώσουμε λόγω της σημαντικότητας τους, έχουμε συμπεριλάβει και τις εντολές στον πυρήνα του PIC, παρόλο που πρόκειται μάλλον για κάποιο λόγο παρά υλικό στοιχείο του μικροελεγκτή.

Στις περιφερειακές μονάδες ανήκουν:

- Οι θύρες εισόδου/εξόδου γενικής χρήσης
- Οι μετρητές χρόνου (τρεις μονάδες)
- Η μονάδα διαμόρφωσης πλάτους
- Οι θύρες σειριακής επικοινωνίας (τρεις θύρες)
- Η θύρα παράλληλης επικοινωνίας
- Η μονάδα παραγωγής τάσης αναφοράς
- Οι συγκριτές
- Ο μετατροπέας αναλογικού σήματος σε ψηφιακό

Ο PIC είχε αρχικά σχεδιαστεί από την εταιρεία General Instruments. Η παραγωγή του είχε προσανατολισμό σε απλές εφαρμογές ελέγχου, από εκεί προέκυψε και το όνομα Peripheral Interface Controller. Στα τέλη της δεκαετίας παράγεται ο PIC 1650 και 1655 επεξεργαστής. Αν και ήταν σχετικά αργό, εν τούτης περιείχε κάποια πολύ σημαντικά και καινοτόμα χαρακτηριστικά. Η απλή CPU με δομή τύπου RISC, με ένα απλό καταχωρητή εργασίας και 30 μόνο εντολές. Οι ακροδέκτες του ολοκληρωμένου μπορούσαν να δουλέψουν με μεγαλύτερα ρεύματα στις εισόδους/εξόδους. Τα χαρακτηριστικά αυτά είναι που έγιναν και σήμα κατατεθέν για τον PIC – απλότητα, χαμηλό κόστος, υψηλή ταχύτητα και ανεξάρτητο.

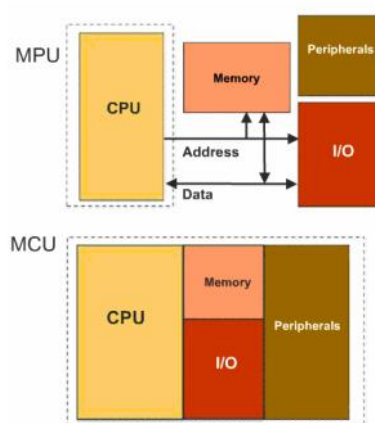
Σε πολλές περιπτώσεις οι μικροελεγκτές PIC βρισκόταν μπροστά από τον ανταγωνισμό. Παρά την εξελίξει μετά από τόσα χρόνια ακόμη, συναντάμε χαρακτηριστικά του παλιού γενικού μικροελεγκτή της Instruments. Σε αντίθεση με τους ανταγωνιστές της, η Microchip ανέπτυξε εργαλεία απλά στην χρήση, με χαμηλό κόστος ή και δωρεάν. Επιπλέον, για μεγάλο χρονικό διάστημα έμεινε σταθερά εδραιωμένη στο 8-bit κόσμο.

## **8.7 Διαφορές Μικροεπεξεργαστή Και Μικροελεγκτή**

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.

Ο μικροελεγκτής είναι ένα μικρό αυτόνομο υπολογιστικό σύστημα, προγραμματισμένο να εκτελεί μία συγκεκριμένη λογική ακολουθία εντολών, οι οποίες έχουν καταχωρηθεί στην προγραμματιζόμενη μόνιμη μνήμη του. Κάθε φορά που θα επανεκκινείται ο μικροελεγκτής, θα εκτελεί την ίδια λογική. Θα ανακαλεί τα δεδομένα, θα τα επεξεργάζεται και με βάση τα αποτελέσματα της επεξεργασίας θα ελέγχει το περιβάλλον του. Πρόκειται, δηλαδή, για σύστημα ειδικού σκοπού, αφιερωμένο (dedicated) στον έλεγχο και την εξυπηρέτηση ενός συγκεκριμένου αυτοματισμού.

Αντίθετα, ένας μικροεπεξεργαστής μετά την εκκίνησή του δεν είναι από μόνος του σε θέση να εκτελέσει κάποια λογική ακολουθία. Αν και μπορεί να συνδεθεί με μνήμες RAM και ROM, αυτές αποτελούν ξεχωριστές μονάδες, που συνήθως δεν ολοκληρώνονται μέσα στον ίδιο τον μικροεπεξεργαστή. Οι διαφορές αυτές φαίνονται στην παρακάτω εικόνα.



Σχήμα 8.7 : Διαφορές ανάμεσα σε σύστημα μικροεπεξεργαστή (MPU) και μικροελεγκτή (MCU).

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.

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 φον Νόιμαν.

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.
- Μεγάλο μέγεθος 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 χρονοστές-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, με διαφοροποιήσεις 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 9

### 9. ΜΕΘΟΔΟΛΟΓΙΑ ΕΠΙΛΥΣΗΣ ΕΦΑΡΜΟΓΩΝ

Ο κύριος στόχος είναι η εξοικείωση των εκπαιδευομένων με την τεχνολογία των μικροελεγκτών και των ενσωματωμένων συστημάτων καθώς και των ευρέων εφαρμογών τους. Επιπρόσθετα το μάθημα στοχεύει στην επιμόρφωση των εκπαιδευομένων σχετικά με τις αρχές λειτουργίας των ενσωματωμένων συστημάτων σε ένα σύνθετο περιβάλλον σε επίπεδο υλικού, λογισμικού και αρχιτεκτονικής συστήματος, και στην εξοικείωσή τους με τις τεχνολογίες που υποστηρίζουν την ανάπτυξη και τις διαδικασίες κατασκευής τους. Στόχος είναι να αποκτήσουν την ικανότητα να συμμετάσχουν αποτελεσματικά σε έργα σχεδίασης, ανάπτυξης και υλοποίησης ενσωματωμένων συστημάτων.

#### 9.1 Περιγραφή Ενσωματωμένων Συστημάτων

Ως ενσωματωμένο σύστημα ορίζεται κάθε συσκευή η οποία εμπεριέχει ένα προγραμματιζόμενο επεξεργαστή, αλλά δεν είναι από μόνο του ένας γενικού σκοπού υπολογιστής. Ο προγραμματισμός των ενσωματωμένων συστημάτων μπορεί να πραγματοποιηθεί σε γλώσσα μηχανής (assembly) ή σε κάποια υψηλότερη γλώσσα προγραμματισμού αν διατίθεται ο compiler της αντίστοιχης γλώσσας για το συγκεκριμένο επεξεργαστή.

#### 9.2 Σχεδιασμός Εφαρμογών

Οι εφαρμογές που σχεδιάζονται από τον προγραμματιστή δεν είναι πάντα καλογραμμένες εφαρμογές. Όταν ένας προγραμματιστής προσπαθεί να υλοποιήσει έναν αλγόριθμο, η σκέψη του επικεντρώνεται στην ανάλυση της εφαρμογής και στην επίλυση του προβλήματος και όχι στον τρόπο με τον οποίο θα τρέχει η εφαρμογή πάνω στον προγραμματιζόμενο επεξεργαστή. Το γεγονός αυτό έχει ως συνέπεια τη μη-βέλτιστη υλοποίηση μιας εφαρμογής αφού δε λαμβάνει υπόψη τα συγκεκριμένα χαρακτηριστικά ενός ενσωματωμένου συστήματος.

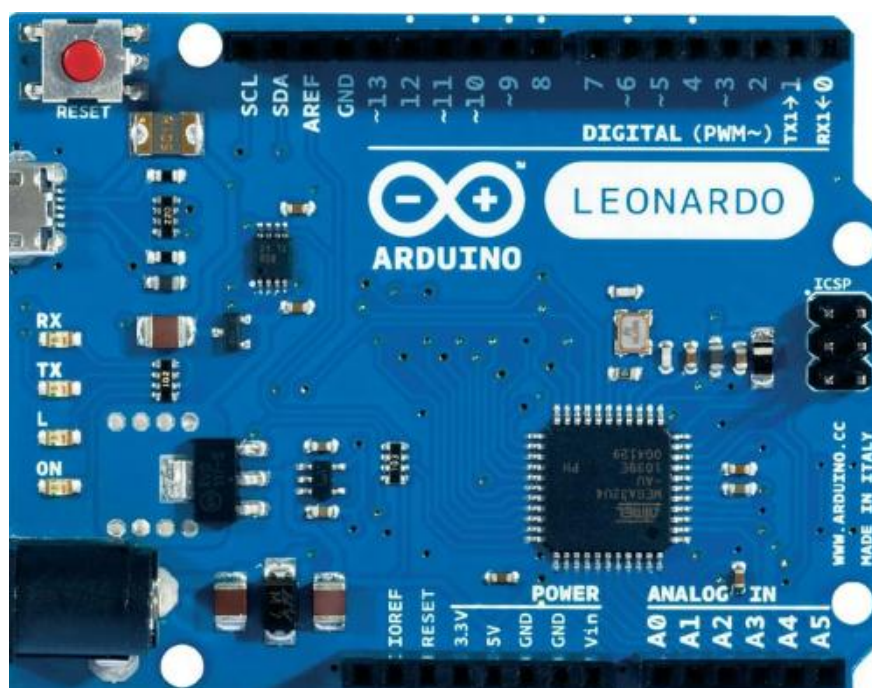
Για να επιτευχθεί μια βέλτιστη υλοποίηση θα πρέπει να μειωθούν οι περιττές εκτελέσεις εντολών και οι άσκοπες προσπελάσεις στη μνήμη, οι οποίες έχουν σαν αποτέλεσμα την αργή εκτέλεση της εφαρμογής, για να πετύχουμε εκτέλεση πραγματικού χρόνου (real-time). Επίσης σε εφαρμογές που πρόκειται να χρησιμοποιηθούν σε φορητές συσκευές, πρέπει να περιοριστεί η κατανάλωση ενέργειας στο ελάχιστο ώστε να αυξηθεί η αυτονομία του συστήματος, καθώς και οι δύο παραπάνω παράγοντες πρέπει να λαμβάνονται σοβαρά υπόψη κατά το σχεδιασμό εφαρμογών με τη χρήση ενσωματωμένων συστημάτων.

##### 9.2.1 Σχεδίαση με μικροελεγκτή 8bit – Η αναπτυξιακή πλατφόρμα Arduino

Αν και οι επεξεργαστές που χρησιμοποιούνται στους τυπικούς σταθμούς εργασίας ή στους διακομιστές ή στα έξυπνα τηλέφωνα, είναι αρχιτεκτονικής 32 bit ή 64 bit, στα ενσωματωμένα συστήματα χρησιμοποιούνται αρχιτεκτονικές από 8 bit έως και

64bit. Ένας αρκετά δημοφιλής, αξιόπιστος και οικονομικός επεξεργαστής 8 bit είναι ο ATmega328P, που αποτελεί την καρδιά της αναπτυξιακής πλατφόρμας Arduino UNO. Σε αυτό το παράρτημα, θα γίνει η παρουσίαση του Arduino UNO και στη συνέχεια θα ακολουθήσουν οι ασκήσεις εμβάθυνσης σε αυτή την αρχιτεκτονική.

Στην ενότητα αυτή, παρουσιάζεται ο μικροελεγκτής Arduino που χρησιμοποιήθηκε, τα χαρακτηριστικά του και μερικά από τα διαθέσιμα εξαρτήματα που μπορούν να συνδεθούν απ' ευθείας με αυτό. Το Arduino είναι μία ανοιχτού λογισμικού πλατφόρμα πρωτοτύπων ηλεκτρονικών συσκευών που βασίζονται στην ευελιξία και στην ευκολία χρήσης υλικού και λογισμικού. Το Arduino μπορεί να αλληλεπιδρά με το περιβάλλον κάνοντας λήψη σημάτων μέσα από μια ποικιλία αισθητήρων. Το arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη διαλογικών λειτουργιών, με είσοδο από μια πληθώρα πηγών (διακόπτες, αισθητήρες,..) και έλεγχο φυσικών αντικειμένων (φώτα, κινητήρες,..). Το arduino μπορεί να είναι αυτόνομο ή να επικοινωνεί με άλλα arduino ή υπολογιστικά συστήματα. Τα έργα που βασίζονται σε αυτόν τον μικροελεγκτή, μπορούν να είναι αυτόνομα ή μπορούν να επικοινωνούν με το λογισμικό που τρέχει σε έναν υπολογιστή (π.χ. Flash, Processing, MaxMSP). Το arduino είναι ένα εργαλείο που μας επιτρέπει να κατασκευάσουμε υπολογιστικά συστήματα που μπορούν να αισθανθούν και να ελέγξουν το φυσικό κόσμο πολύ πιο εύκολα από ότι αν χρησιμοποιούσαμε έναν τυπικό υπολογιστή γραφείου. Είναι μια αρχιτεκτονική που βασίζεται σε ανοιχτό κώδικα, μια πλακέτα μικρο-επεξεργαστή και ένα αναπτυξιακό περιβάλλον για τη συγγραφή προγράμματος για την πλακέτα. Η οικογένεια Arduino αποτελείται από πολλές αναπτυξιακές πλακέτες, διαφορετικών χαρακτηριστικών. Η πιο δημοφιλής αναπτυξιακή πλακέτα είναι η Arduino UNO.



## 9.2.2 Σχεδίαση σε επαναδιομόρφωση Αρχιτεκτονική Altera FPGA

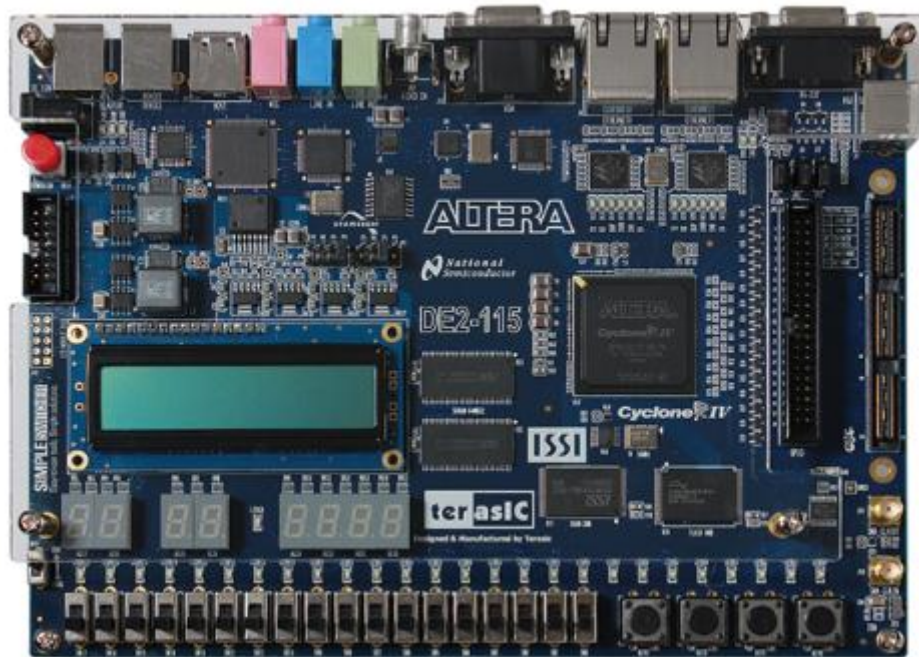
Όπως έχει γίνει κατανοητό στα προηγούμενα κεφάλαια, το οικοσύστημα των ενσωματωμένων συστημάτων καλύπτει ένα πολύ μεγάλο εύρος απαιτήσεων και για αυτό το λόγο χρησιμοποιούνται πλήθος τεχνολογιών υλοποίησης. Εκτός από τους επεξεργαστές των 8bit και των 32bit, αρκετά συχνά χρησιμοποιούνται και οι επαναδιαμορφώσιμες αρχιτεκτονικές, όπως Altera και Xilinx. Οι αρχιτεκτονικές αυτές χρησιμοποιούνται τόσο στα πρώτα στάδια του σχεδιασμού για τη γρήγορη προτυποποίηση και τη δημιουργία του πρωτοτύπου, όσο και στην τελική υλοποίηση, σε προβλήματα που δε μπορούν να υλοποιηθούν οικονομικά και αξιόπιστα με άλλες επιλογές. Σε αυτό το παράρτημα, θα γίνει η παρουσίαση της αναπτυξιακής πλακέτας της Altera DE2-115 και στη συνέχεια θα ακολουθήσουν οι ασκήσεις εμβάθυνσης σε αυτή την πλατφόρμα.

Η αναπτυξιακή πλακέτα FPGA που θα χρησιμοποιήσουμε είναι η Altera DE2-115, η οποία αν και είναι σχεδιασμένη κατά βάση για ακαδημαϊκούς σκοπούς (διδασκαλία και έρευνα), εντούτοις δεν υπολείπεται ως προς τη λειτουργικότητά της και μπορεί να χρησιμοποιηθεί ακόμη και σε πραγματικές εφαρμογές ενσωματωμένων συστημάτων. Εξάλλου, η διαδικασία ανάπτυξης ενός ενσωματωμένου συστήματος σε μια πλακέτα FPGA είναι η ίδια σε πλακέτες χαμηλής ή υψηλής πολυπλοκότητας ακόμη και σε διαφορετικές εταιρίες.

Τα παρακάτω στοιχεία περιλαμβάνονται στην πλακέτα DE2-115:

- FPGA
  - Cyclone V SoC 5CSEMA5F31 / EPCQ256 256-Mbit serial configuration device
- I/O Διεπαφές
  - Εσωτερική USB-Blaster για την επαναδιαμόρφωση του FPGA
  - Είσοδος/Εξοδος γραμμής ήχου, είσοδος μικροφώνου (24-bit)
  - Έξοδος Video (VGA 24-bit DAC)
  - Είσοδος Video (NTSC/PAL/SECAM)
  - Θύρα υπερέθρων
  - Σειριακή θύρα RS232 DB9
  - 10/100/1000 Ethernet (2 θύρες)
  - Δυο θύρες USB 2.0 Host (Type A/B)
  - PS/2 πληκτρολόγιο και mouse
  - Αναμονές επέκτασης (40-pin headers)
- Μνήμη
  - 128MB SDRAM, 2MB SRAM, 8MB Flash, 32Kbit EEPROM
  - Υποδοχή για Micro SD κάρτα μνήμης
- Απεικόνιση
  - 8 οθόνες 7-segment
  - 1 οθόνη LCD 2x16
- Switches and LEDs
  - 18 διακόπτες
  - 18 LEDs
  - 4 διακόπτες με αποκλιδωνισμό

- Ρολόγια
  - 50 MHz clock (3 ρολόγια)



Σχήμα 9.2.2: Η αναπτυξιακή πλακέτα που χρησιμοποιείται στα εργαστήρια είναι η Altera DE2-115 που μπορεί να την προμηθευτεί κάποιος σε χαμηλή ακαδημαϊκή τιμή από το διαδίκτυο.

Για να αναπτυχθεί ένα ενσωματωμένο σύστημα σε οποιαδήποτε πλακέτα της Altera χρησιμοποιείται το λογισμικό Quartus. Υπάρχουν 2 εκδόσεις του προγράμματος, που μπορούν να μεταφορτωθούν δωρεάν από τον ιστοχώρο της Altera ([www.altera.com](http://www.altera.com)). Η δωρεάν έκδοση (web), η οποία είναι μεν πλήρης αλλά φέρει κάποιους περιορισμούς ως προς τις δυνατότητες των εργαλείων (π.χ. δεν υποστηρίζεται η παράλληλη εκτέλεση, και όλα εκτελούνται σε έναν μόνο πυρήνα), και η επαγγελματική έκδοση (subscription), η οποία απαιτεί ένα αρχείο με άδειες που έχει αγοράσει κάποιος από την Altera και οι οποίες καθορίζουν ακριβώς τα εργαλεία, τα διαθέσιμα δομοστοιχεία (IP) και τις δυνατότητες που έχει στη διάθεσή του ο σχεδιαστής. Στα πλαίσια των εργαστηριακών ασκήσεων θα χρησιμοποιήσουμε τη web έκδοση, η οποία μας καλύπτει πλήρως. Η τελευταία έκδοση του Quartus (Ιούλιος 2015) είναι η 15, αλλά στα πλαίσια αυτών των ασκήσεων χρησιμοποιούμε την έκδοση 13, η οποία είναι διαθέσιμη για μεταφόρτωση και δεν έχει πολλές διαφοροποιήσεις σε σχέση με την 15. Όμως, το πιο σημαντικό στοιχείο είναι ότι αυτή είναι η τελευταία έκδοση που υποστηρίζει 32bit και 64bit συστήματα, από Windows XP και άνω, και έτσι μπορεί να χρησιμοποιηθεί σε όλους τους υπολογιστές που κατασκευάστηκαν από το 2001 και μετά. Κατά την αγορά της πλακέτας Altera DE2-115 υπάρχει το λογισμικό σε CD και έτσι δε χρειάζεται να μεταφορτωθεί από το Internet. Μαζί με το λογισμικό, υπάρχουν και αρκετά αρχεία επίδειξης των δυνατοτήτων της πλακέτας.

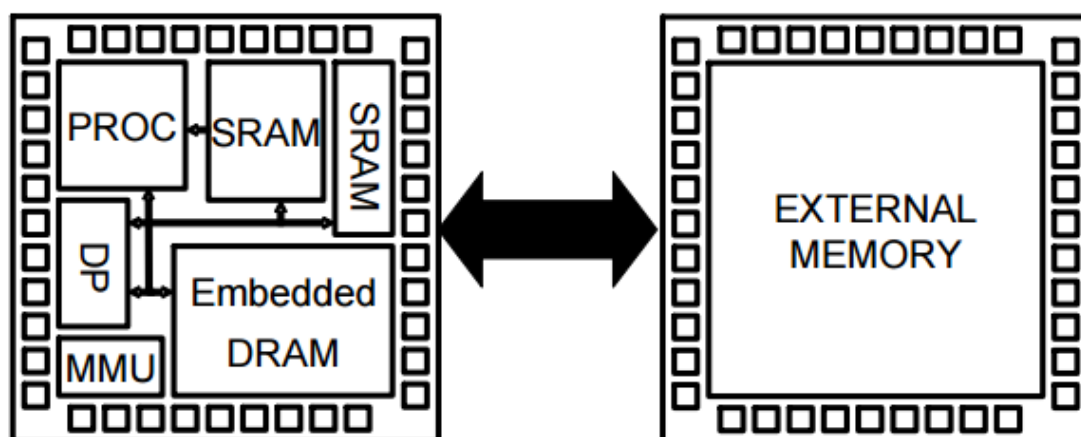


### 9.3 Μεθοδολογία Βελτιστοποίησης Αλγορίθμων για χαμηλή κατανάλωση ενέργειας και υψηλή απόδοση

Επειδή ο προγραμματιστής, κατά στην υλοποίηση αλγορίθμων σε μια γλώσσα υψηλού επιπέδου δε λαμβάνει υπόψη του το υπολογιστικό σύστημα που θα εκτελέσει την εφαρμογή, δεν εκμεταλλεύεται τα χαρακτηριστικά του συστήματος που του προσφέρονται. Ο κύριος λόγος της αργής εκτέλεσης σε εφαρμογές πολυμέσων και δικτύων οφείλεται στο μεγάλο όγκο μεταφοράς δεδομένων από και προς τις μνήμες δεδομένων. Για τη βελτιστοποίηση αλγορίθμων με βάση την αρχιτεκτονική υλοποίησης έχει αναπτυχθεί μια Μεθοδολογία Εξερεύνησης Μεταφορών και Αποθήκευσης Δεδομένων.

Η μεθοδολογία αυτή βασίζεται σε αλγοριθμικούς μετασχηματισμούς, με στόχο τη μείωση των προσπελάσεων στη εξωτερική μνήμη. Η μείωση αυτή επιτυγχάνεται αποθηκεύοντας σε μικρού μεγέθους μνήμη προσωρινά τμήματα των δεδομένων προς επεξεργασία. Οι μικρές προσωρινές μνήμες μπορούν να τοποθετηθούν πλησιέστερα στον υπολογιστικό πυρήνα που έχουν μικρότερο χρόνο προσπέλασης αλλά και χαμηλότερη κατανάλωση ενέργειας ανά προσπέλαση. Τα στάδια της βελτιστοποίησης μιας εφαρμογής είναι τρία.

Το πρώτο στάδιο είναι ο εντοπισμός των σημείων της εφαρμογής όπου εμφανίζονται οι μεγαλύτερες καθυστερήσεις και εκείνων των σημείων όπου γίνονται οι περισσότερες προσπελάσεις στη μνήμη. Η ανάλυση αυτή ονομάζεται profiling και σε μικρού μεγέθους εφαρμογές ο εντοπισμός μπορεί να γίνει εύκολα, αντιθέτως σε μεγάλο μεγέθους αλγόριθμους η χρήση εργαλείων είναι απαραίτητη. Ένα τέτοιο εργαλείο που βοηθά το σχεδιαστή είναι το ATOMIUM, το οποίο μετρά τον αριθμό των προσπελάσεων σε κάθε πίνακα δεδομένων της εφαρμογής και κατευθύνει το σχεδιαστή να επικεντρώσει την προσπάθεια και τη βελτίωση αυτών των σημείων.



Σχήμα 9.3 : Ενσωματωμένο Σύστημα με προσωρινές μνήμες, αποθήκευση πάνω στο ολοκληρωμένο κύκλωμα του επεξεργαστή για την προσωρινή αποθήκευση μικρού μεγέθους μεταβλητών, μειώνοντας τον αριθμό των προσπελάσεων στην εξωτερική μνήμη.

## 9.4 Το Μέλλον των Ενσωματωμένων Συστημάτων

Στη πτυχιακή μου εργασία έχω επικεντρωθεί σε ενσωματωμένα συστήματα, όπως είναι στις μέρες μας. Ωστόσο, όπως συμβαίνει με όλους τους τομείς της επιστήμης και της τεχνολογίας, οι μηχανικοί πρέπει να κοιτάζουν προς το μέλλον, ώστε τα προϊόντα και οι τεχνολογίες που σχεδιάζουν, να έχουν διάρκεια. Η τεχνολογία των ενσωματωμένων λειτουργικών συστημάτων και κατά επέκταση και των ενσωματωμένων συστημάτων αρχείων αναπτύσσεται με γοργούς ρυθμούς. Ήδη όπως προαναφέρθηκε ετοιμάζονται καινούρια ενσωματωμένα συστήματα αρχείων που θα είναι πιο αποδοτικά και θα πλεονεκτούν κατά πολύ των σημερινών συστημάτων αρχείων. Κορυφαίες εταιρίες που έχουν αναπτύξει συστήματα αρχείων και λειτουργικά συστήματα για PC εργάζονται προς αυτήν την κατεύθυνση.

Τα Ενσωματωμένα Συστήματα εμφανίστηκαν ως ξεχωριστός τομέας, για πρώτη φορά στο 6ο Πρόγραμμα Πλαίσιο και οι κύριες κατευθύνσεις ήταν οι μεθοδολογίες σχεδιασμού και η διασύνδεση συστημάτων, συμπεριλαμβανόμενων των ασύρματων δίκτυων αισθητήρων. Στον τελευταίο τομέα έχει σημειωθεί ικανοποιητική πρόοδος τα τελευταία 3-4 χρόνια και, ενώ ήταν μια αναδυόμενη περιοχή χωρίς εκτενή ευρωπαϊκή δραστηριότητα, υπάρχουν σήμερα πολλές πιλοτικές εφαρμογές. Είναι μια τεχνολογία με μέλλον. Ένας άλλος τομέας, ο οποίος ξεκίνησε “περιθωριακά” στο προηγούμενο ΠΠ, αλλά θα συνεχιστεί περαιτέρω στο 7ο ΠΠ, είναι ο τομέας των υπολογιστικών αρχιτεκτονικών. Έχουν σημειωθεί σημαντικές εξελίξεις, αλλά θεωρούμε ότι υπάρχουν μεγάλα περιθώρια ανάπτυξης στο μέλλον λόγω της μετάβασης προς τους πολυκεντρικούς (multicore) επεξεργαστές.

Η θέση της Ευρώπης στα ενσωματωμένα συστήματα είναι ισχυρή και μπορώ να πω πως πρωτοπορούμε διεθνώς. Ωστόσο, έχουμε να υπερνικήσουμε πολλές σημαντικές δυσκολίες, όπως, για παράδειγμα, το πρόβλημα της πολυπλοκότητας και του "χάσματος παραγωγικότητας". Οι επιδόσεις αλλά και η πολυπλοκότητα του hardware και των επεξεργαστών (chips) αυξάνονται με γρηγορότερο ρυθμό από ό,τι μπορεί να παρακολουθήσει το λογισμικό. Δηλαδή, σήμερα, το hardware είναι 100 φορές καλύτερο από ό,τι ήταν 10 χρόνια πριν. Τα εργαλεία σχεδιασμού και ανάπτυξης λογισμικού, όμως, δεν είναι 100 φορές καλύτερα, οπότε, οι εταιρείες ανταποκρίνονται στις αυξανόμενες ανάγκες προγραμματισμού των πολύ πιο αποδοτικών επεξεργαστών και πολυπλοκότερων συστημάτων προσλαμβάνοντας περισσότερους προγραμματιστές. Αυτό, βέβαια, δεν είναι ένα μοντέλο που είναι βιώσιμο μακροπρόθεσμα, διότι και ο αριθμός των προγραμματιστών είναι περιορισμένος και η πιθανότητα σφάλματος αυξάνεται γεωμετρικά όσο αυξάνεται η πολυπλοκότητα των συστημάτων. Η περιοχή αυτή παρουσιάζει λοιπόν μεγάλες δυσκολίες που θα πρέπει να αντιμετωπιστούν όσο το δυνατόν πιο γρήγορα.

Ένα δεύτερο πρόβλημα που αντιμετωπίζουμε στην Ευρώπη είναι ότι, ενώ οι ερευνητικές μας ομάδες στα ενσωματωμένα συστήματα είναι από τις καλύτερες στον κόσμο, η υπεροχή αυτή βασίζεται στην πρόοδο που έχει γίνει σε κάθε κλάδο μεμονωμένα, δηλαδή στον κλάδο των ηλεκτρονικών για αεροπλάνα, ή στον κλάδο της κινητής τηλεφωνίας, κ.λπ. Σήμερα, οι μεμονωμένοι κλάδοι δεν μπορούν να

ανταποκριθούν στις ανάγκες, απαιτούνται συνεργασίες και συνέργειες. Οι ξεχωριστές τεχνικές και τα πρότυπα που έχει αναπτύξει ο κάθε κλάδος θα πρέπει, λοιπόν, να συνδυαστούν.

## BIBΛΙΟΓΡΑΦΙΑ

- [http://arch.icte.uowm.gr/courses/embedded/01ES\\_oc.pdf](http://arch.icte.uowm.gr/courses/embedded/01ES_oc.pdf)
- <https://www.cs.ucy.ac.cy/courses/EPL362/Notes/Enothta-7.pdf>
- <http://www.ece.ncsu.edu/research/cas/ecs>
- [http://newtech-pub.com/wpcontent/uploads/2013/10/vhdl\\_per.pdf](http://newtech-pub.com/wpcontent/uploads/2013/10/vhdl_per.pdf)
- <https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/2247/1/book.compress.pdf>
- <http://arch.icte.uowm.gr/mdasyg/book/embedded/chapter08.pdf>
- [http://teachers.teicm.gr/kalomiros/Mtptx/ebooks/Embedded\\_PIC\\_new.pdf](http://teachers.teicm.gr/kalomiros/Mtptx/ebooks/Embedded_PIC_new.pdf)
- [https://arch.icte.uowm.gr/mdasyg/pdfs/sozubek\\_thesis.pdf](https://arch.icte.uowm.gr/mdasyg/pdfs/sozubek_thesis.pdf)
- <https://www.cs.ucy.ac.cy/courses/EPL362/Notes/Enothta-7.pdf>
- [http://newtech.j.scaleforce.net/wp-content/uploads/2013/11/vhdl\\_kef.pdf](http://newtech.j.scaleforce.net/wp-content/uploads/2013/11/vhdl_kef.pdf)
- [http://teachers.teicm.gr/kalomiros/Mtptx/ebooks/Essential\\_VHDL\\_Kalomiros.pdf](http://teachers.teicm.gr/kalomiros/Mtptx/ebooks/Essential_VHDL_Kalomiros.pdf)
- <http://nefeli.lib.teicrete.gr/browse/stef/epp/2009/MaurokostasKonstantinos,KenisKonstantinos/attached-document-1258542739-61226-25799/Maurokostas2009.pdf>
- <http://www.ece.ncsu.edu/research/cas/ecs>
- <http://repository.library.teimes.gr/xmlui/bitstream/handle/123456789/2386/CIED%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20-%20.pdf?sequence=1>
- [http://nemertes.lis.upatras.gr/jspui/bitstream/10889/4402/1/Siourounis\\_K\\_t\\_hesis.pdf](http://nemertes.lis.upatras.gr/jspui/bitstream/10889/4402/1/Siourounis_K_t_hesis.pdf)