Σχεδιάζοντας με μικροεπεξεργαστές

- Αρχιτεκτονικές και περιεχόμενα:
  - λογισμικό
  - υλικό
- Debugging(διόρθωση-εκκαθάριση).
- Δοκιμή κατασκευής.

Πλατφόρμα αρχιτεκτονικής υλικού

Περιλαμβάνει διάφορα στοιχεία:

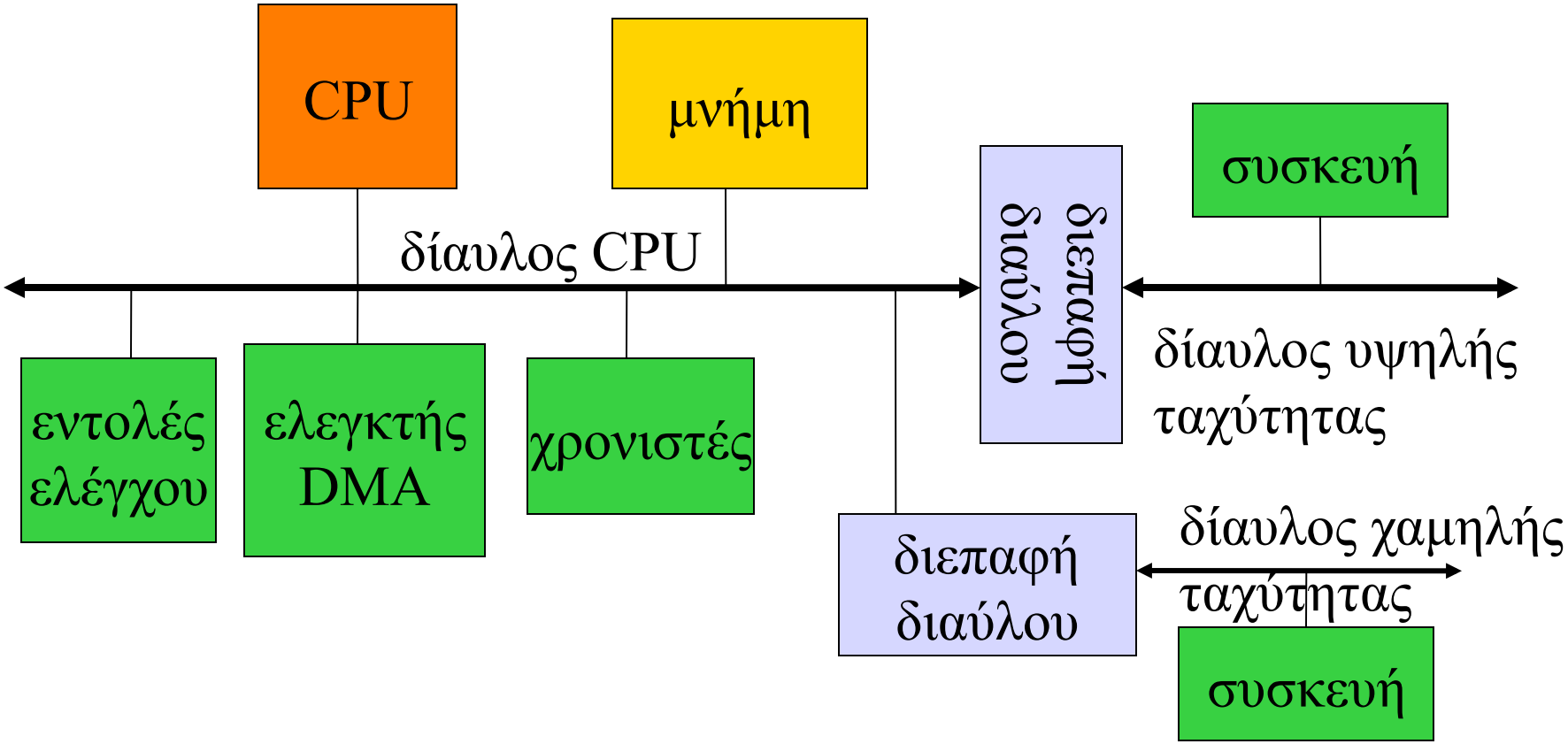
- CPU
- δίαυλο
- μνήμη
- συσκευές I/O: δικτυακές, αισθητήρες, ενεργοποιητές, κτλ.

Πόσο μεγάλο/γρήγορο πρέπει να είναι το καθένα;

Το PC σαν μια πλατφόρμα

- Πλεονεκτήματα:
  - φτηνό και εύκολο να το αποκτήσετε
  - πλούσιο και γνώριμο λογισμικό περιβάλλον
- Μειονεκτήματα:
  - απαιτεί πολλές πηγές υλικού
  - έχει προβλήματα σε εφαρμογές πραγματικού χρόνου

Τυπική πλατφόρμα υλικού PC



Τυπικοί δίαυλοι

- ISA (Industry Standard Architecture): αρχικός δίαυλος IBM PC, χαμηλής ταχύτητας για τα σημερινά πρότυπα.
- PCI: πρότυπα για υψηλής ταχύτητας διασύνδεση
  - 33 ή 66 MHz.
- USB (Universal Serial Bus), Firewire: σχετικά χαμηλού κόστους σειριακή διεπαφή με υψηλή ταχύτητα.

Πλατφόρμα λογισμικού PC

- Το IBM PC χρησιμοποιεί BIOS (Basic I/O System) για να εφαρμόσει λειτουργίες χαμηλού επιπέδου:
  - boot-up
  - ελάχιστοι οδηγοί συσκευών(drivers)
- Το BIOS έχει γίνει ένας γενικός όρος για το χαμηλότερο επίπεδο λογισμικού του συστήματος.

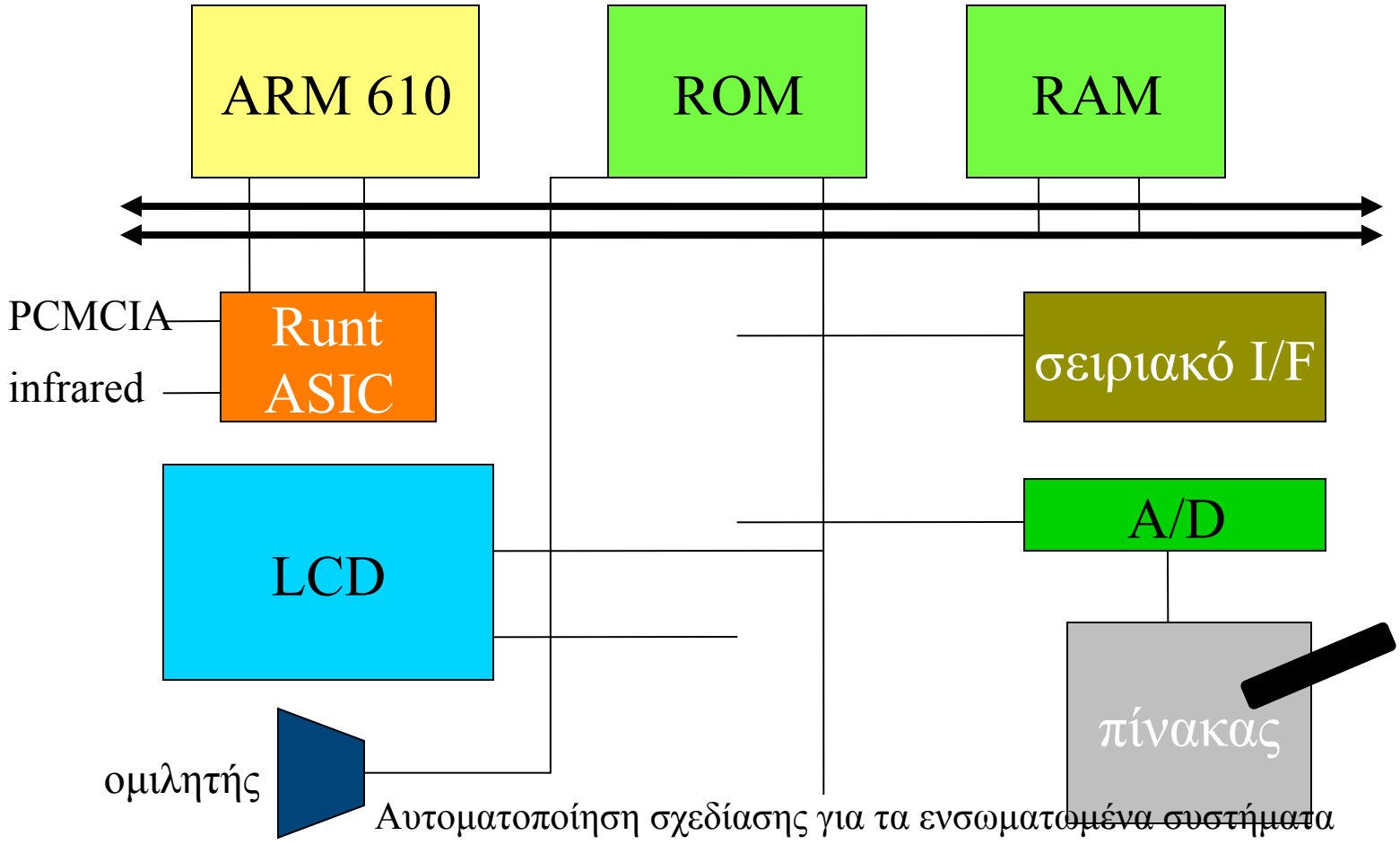
Παράδειγμα: StrongARM

- Το σύστημα StrongARM περιέχει:
  - CPU chip (ρολόι 3.686 MHz)
  - διαμόρφωση ελέγχου συστήματος (ρολόι 32.768 kHz).
    - Ρολόι πραγματικού χρόνου
    - Λειτουργικό σύστημα χρονομέτρου
    - I/O γενικού σκοπού
    - Ελεγκτής διακοπών
    - Ελεγκτής διαχείρισης ισχύος
    - Ελεγκτής επαναχρησιμοποίησης

Πλεονεκτήματα και μειονεκτήματα

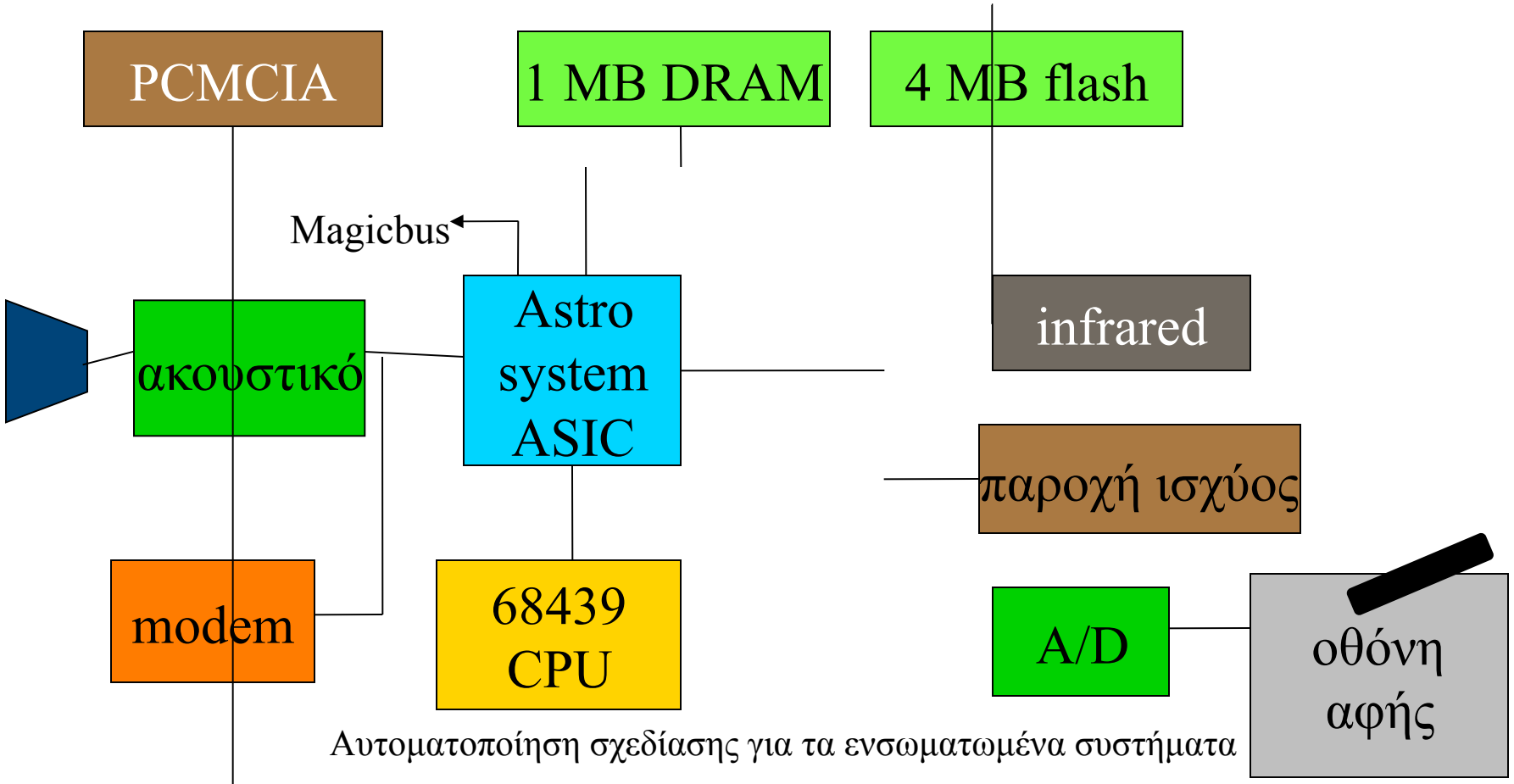
- Άφθονες επιλογές υλικού.
- Απλή σημασιολογία προγραμματισμού.
- Καλά περιβάλλοντα ανάπτυξης λογισμικού.
- Περιορισμένη απόδοση.

Αρχιτεκτονική υλικού Apple Newton

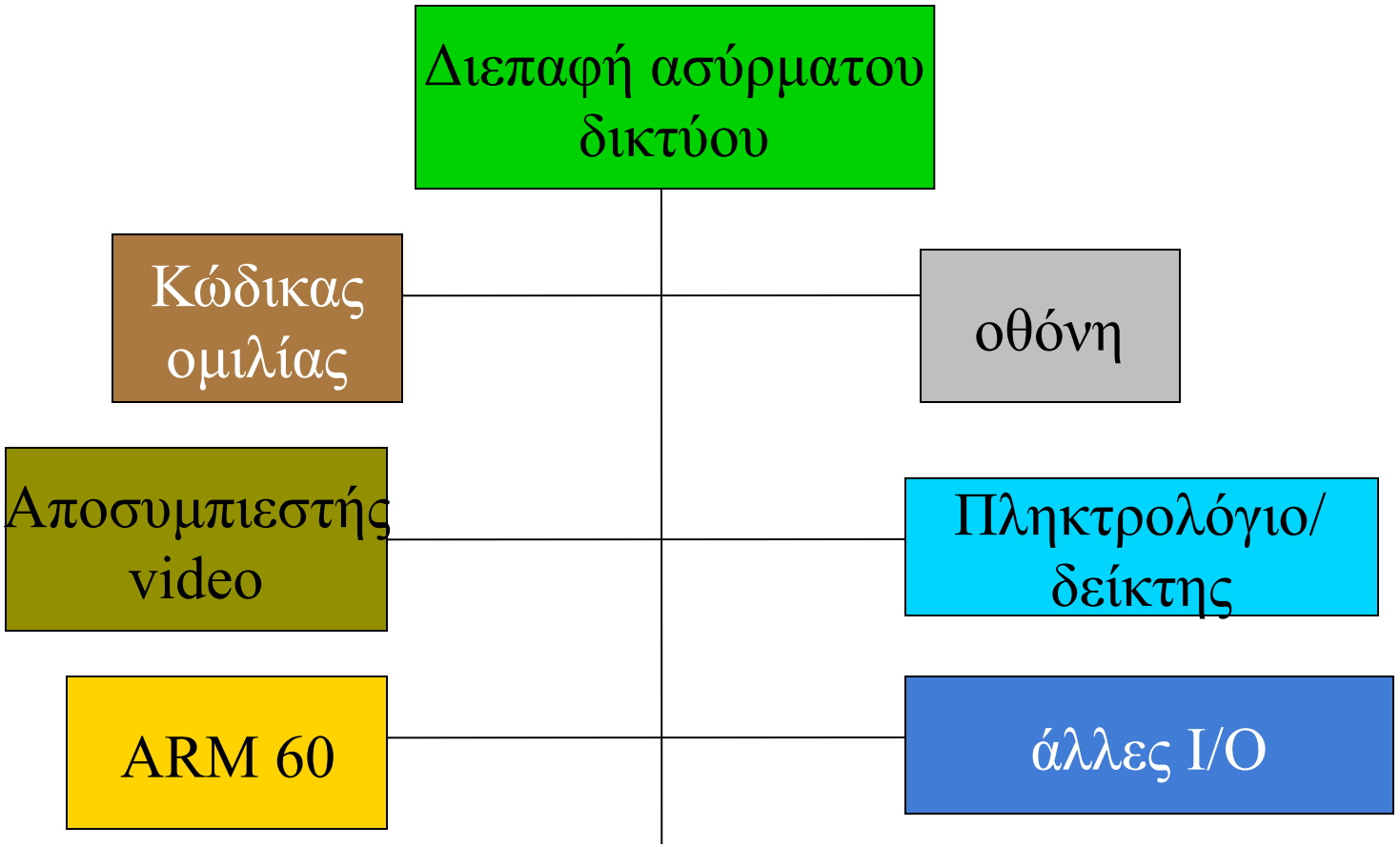


Αυτοματοποίηση σχεδίασης για τα ενσωματωμένα συστήματα

Αρχιτεκτονική υλικού Motorola Envoy



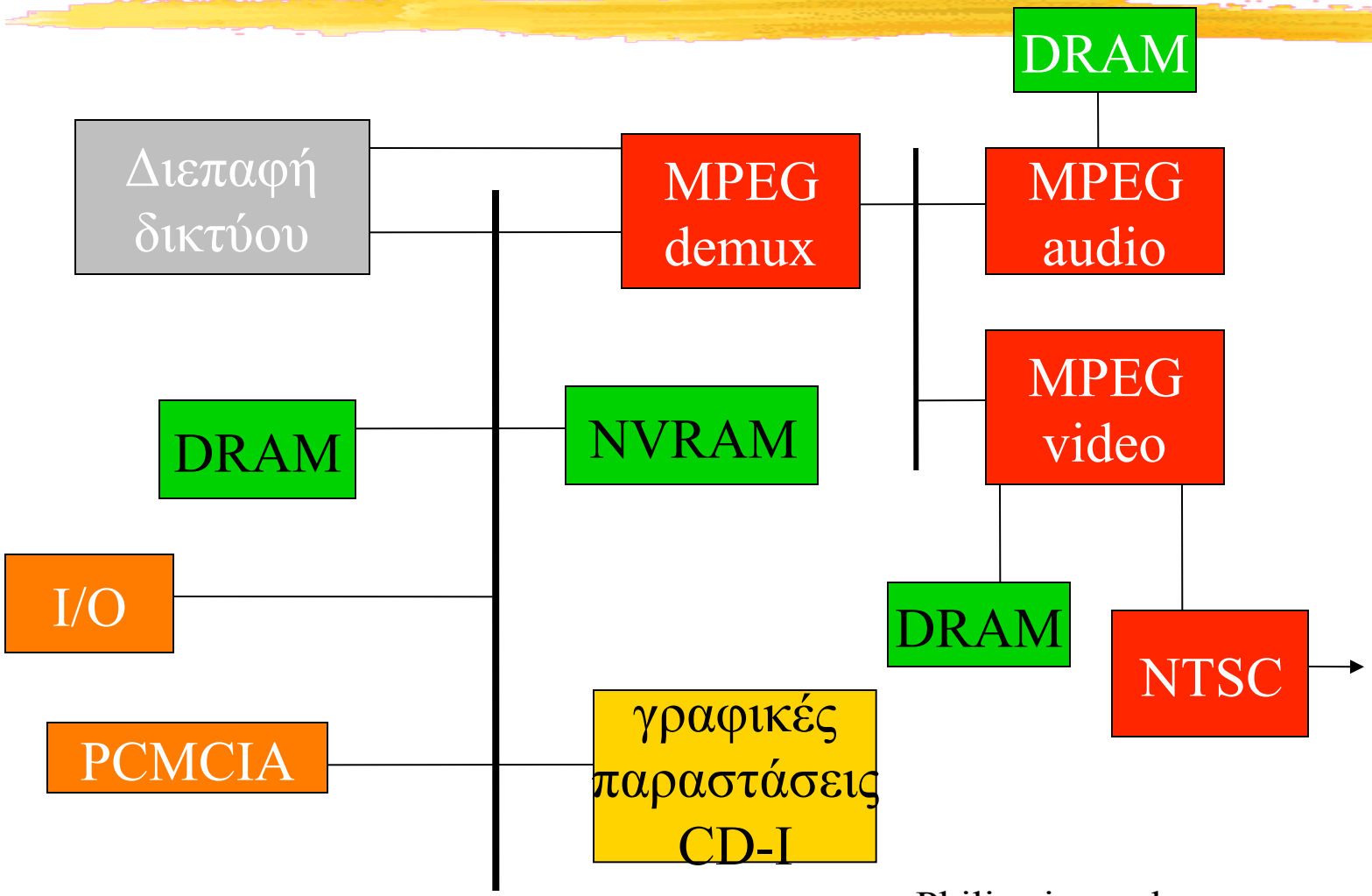
Αρχιτεκτονική υλικού InfoPad



Υλικό vs. λογισμικό

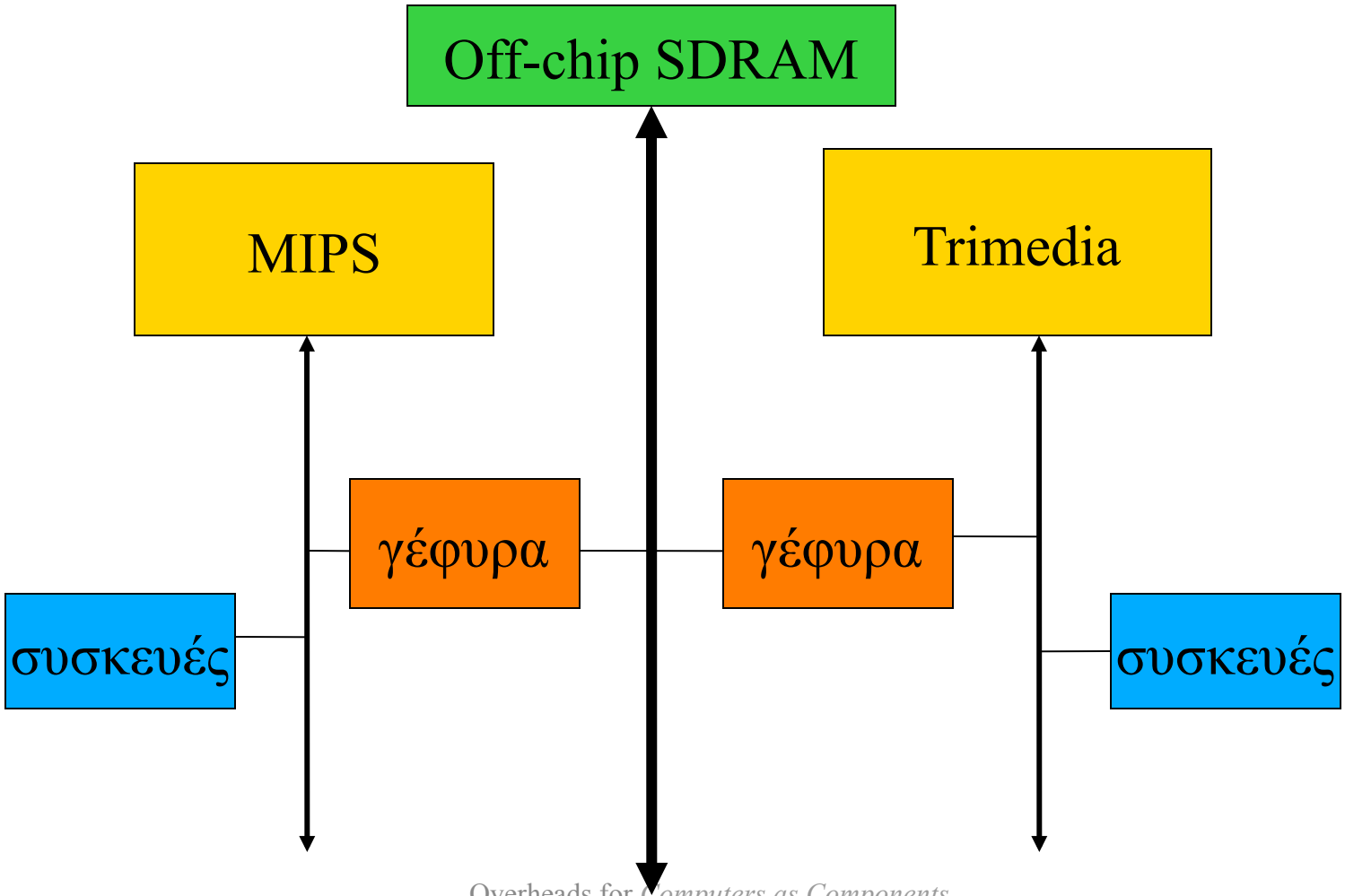
- Το ειδικής χρήσης υλικό καταναλώνει συχνά την λιγότερη ενέργεια.
- Χρειάζεται να σκεφτούμε την επικοινωνία μεταξύ των μονάδων, πολυεπεξεργασία.
- Οι επιταχυντές συχνά απαιτούν όρια στις παραμέτρους.
  - Μπορεί να είναι εντάξει εάν τα πρότυπα περιορίζουν τις παραμέτρους.

Αρχιτεκτονική υλικού αποκωδικοποιητή και μετατροπέα τηλεοπτικού σήματος Philips



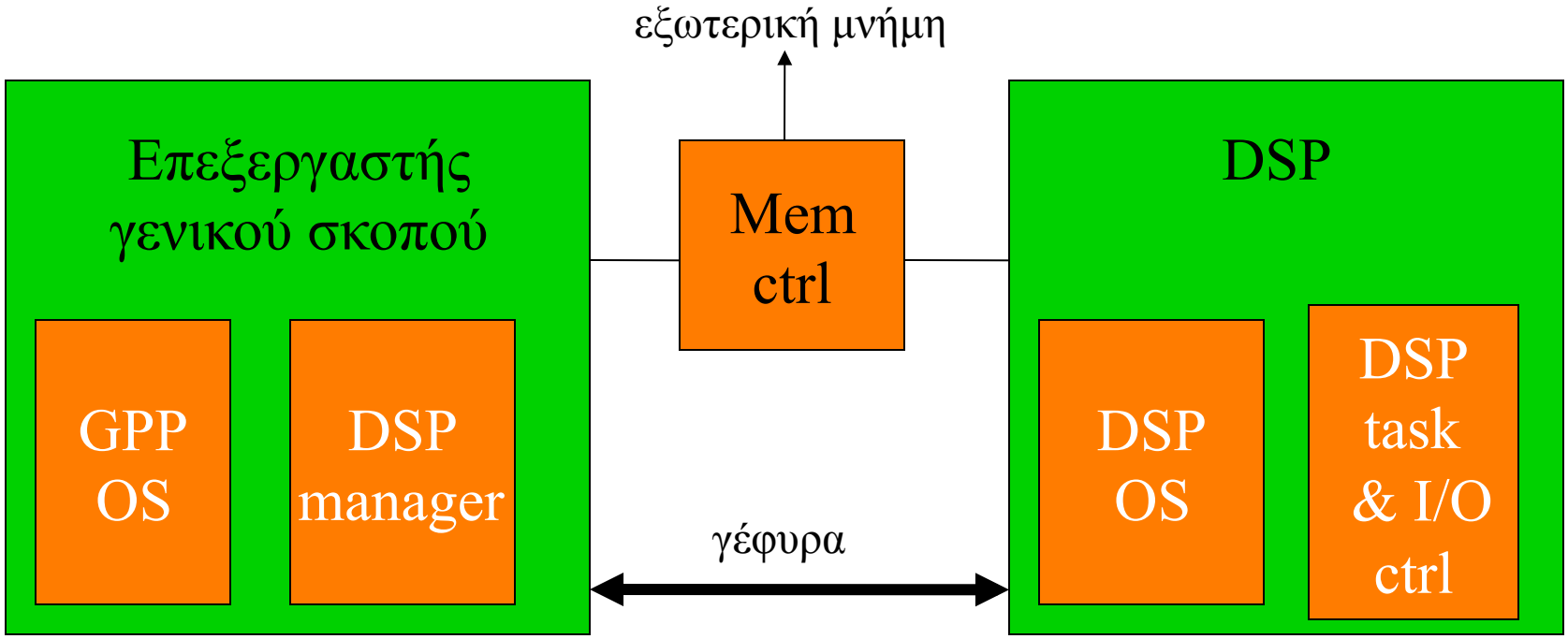
Philips journal

Chip αποκωδικοποιητή και μετατροπέα τηλεοπτικού σήματος Viper



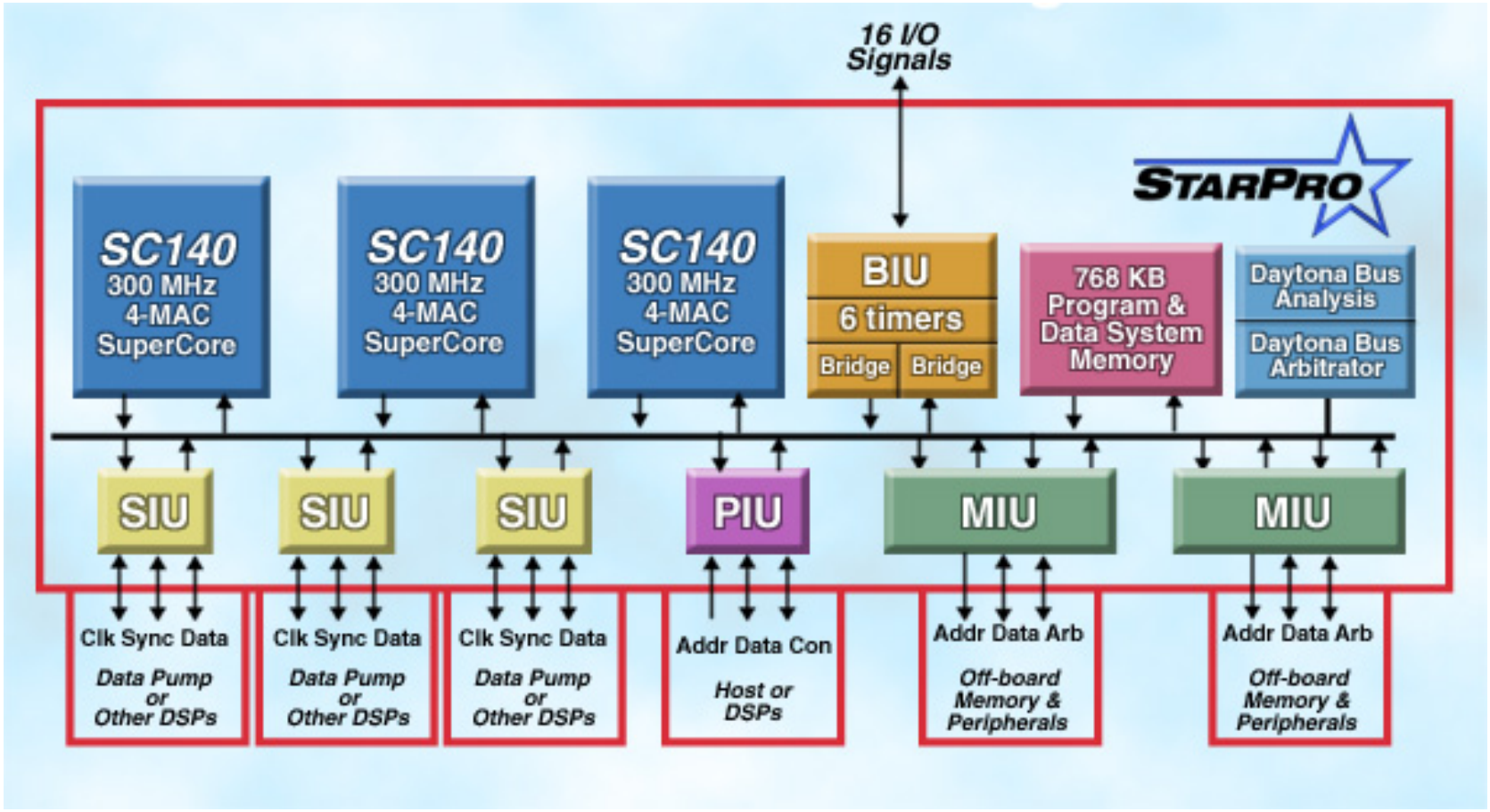
Ανοικτή πλατφόρμα εφαρμογών πολυμέσων TI

- Κοινό σύστημα μνήμης διπλού επεξεργαστή:



http://www.ti.com/sc/docs/apps/wireless/omap/overview.htm

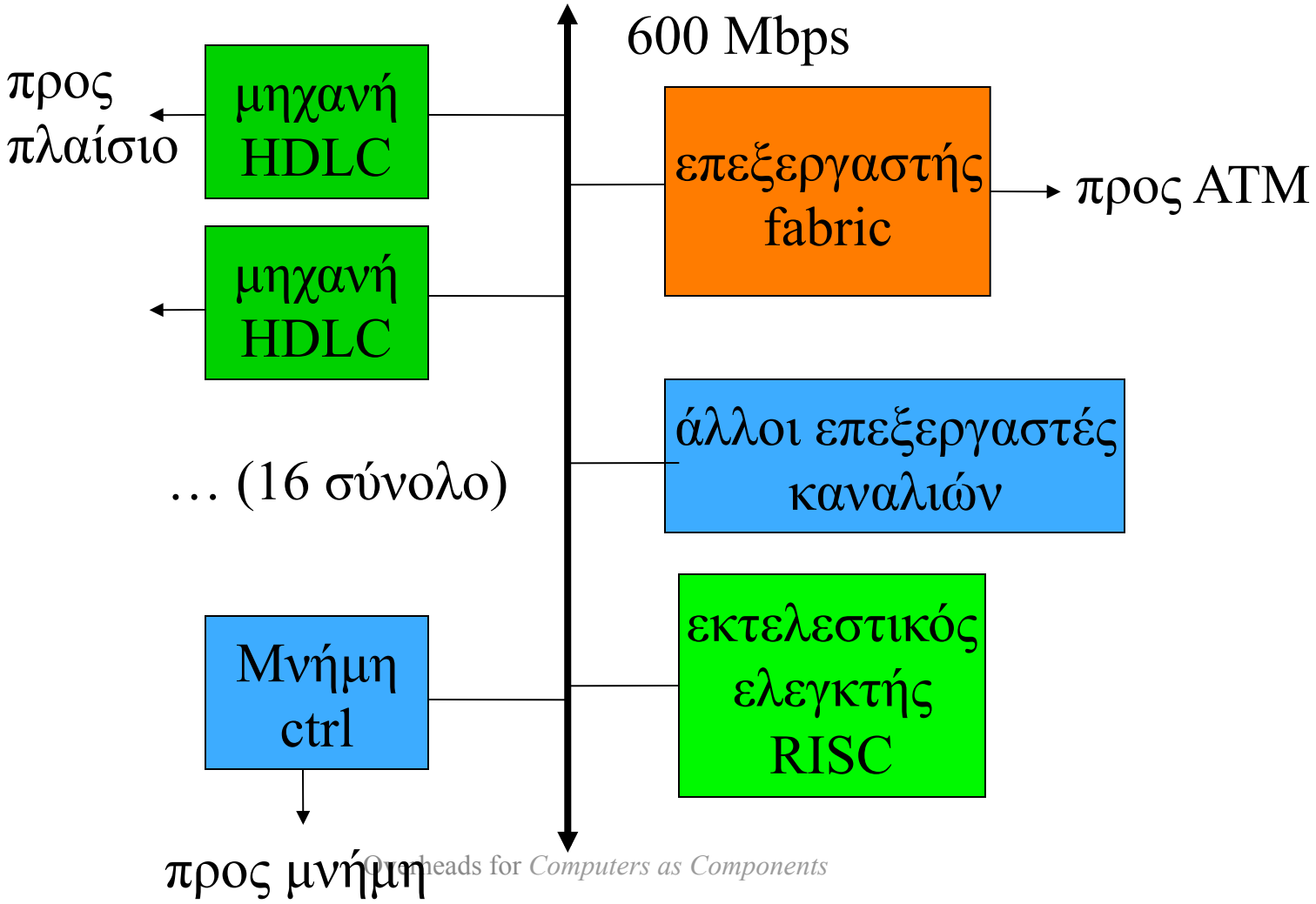
Πλατφόρμα Agere StarPro



http://www.lucent.com/micro/starpro/arch.html

Επεξεργαστής δικτύου C-Port C5

http://www.cportcorp.com/products/digital.htm



Αρχιτεκτονικές υλικού και λογισμικού

Το υλικό και το λογισμικό συσχετίζονται στενά:

- Το λογισμικό δεν εκτελείται χωρίς υλικό
- Το πόσο υλικό θα χρειαστούμε καθορίζεται από τις απαιτήσεις του λογισμικού:
  - ταχύτητα
  - μνήμη

Αρχιτεκτονική λογισμικού

Η λειτουργική περιγραφή πρέπει να σπάσει σε κομμάτια:

- διαίρεση μεταξύ ανθρώπων
- εννοιολογική οργάνωση
- απόδοση
- δυνατότητα δοκιμής
- συντήρηση

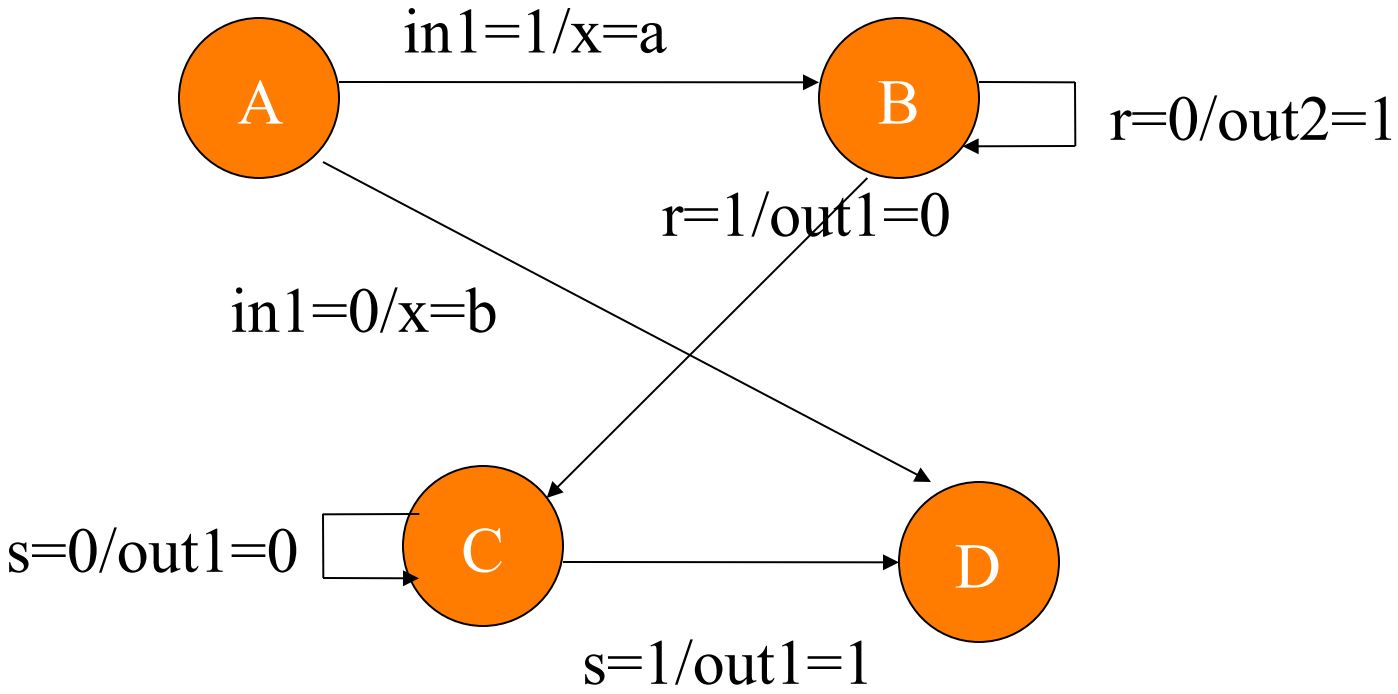
Περιεχόμενα λογισμικού

- Χρειάζεται να σπάσει το σχεδιασμό σε κομμάτια για να είναι διαθέσιμο να γράψει κώδικα.
- Συχνά εμφανίζονται μερικά περιεχόμενα της σχεδίασης.
- Ένα μοτίβο σχεδίασης είναι μια γενική περιγραφή ενός περιεχομένου που μπορεί να προσαρμοστεί και να χρησιμοποιηθεί στις διαφορετικές περιστάσεις.

Μηχανή κατάστασης λογισμικού

- Η μηχανή κατάστασης κρατά την εσωτερική κατάσταση σαν μια μεταβλητή, αλλάζει η κατάσταση βασισμένη στις εισόδους.
- Χρησιμοποιεί:
  - ελεγχόμενο κώδικα ελέγχου
  - συστήματα με αντίδραση

Προδιαγραφή μηχανής καταστάσεων



Δομή κώδικα στη C

- Η τρέχουσα κατάσταση δεσμεύεται σε μια μεταβλητή.
- Ο πίνακας κατάστασης εφαρμόζεται ως διακόπτης(switch).
  - Οι περιπτώσεις καθορίζουν τις καταστάσεις.
  - Οι καταστάσεις μπορούν να εξετάσουν τις εισόδους.
- Ο διακόπτης αξιολογείται επανειλημμένα σε έναν βρόχο στιγμής.

Δομή μηχανής καταστάσεων στη C

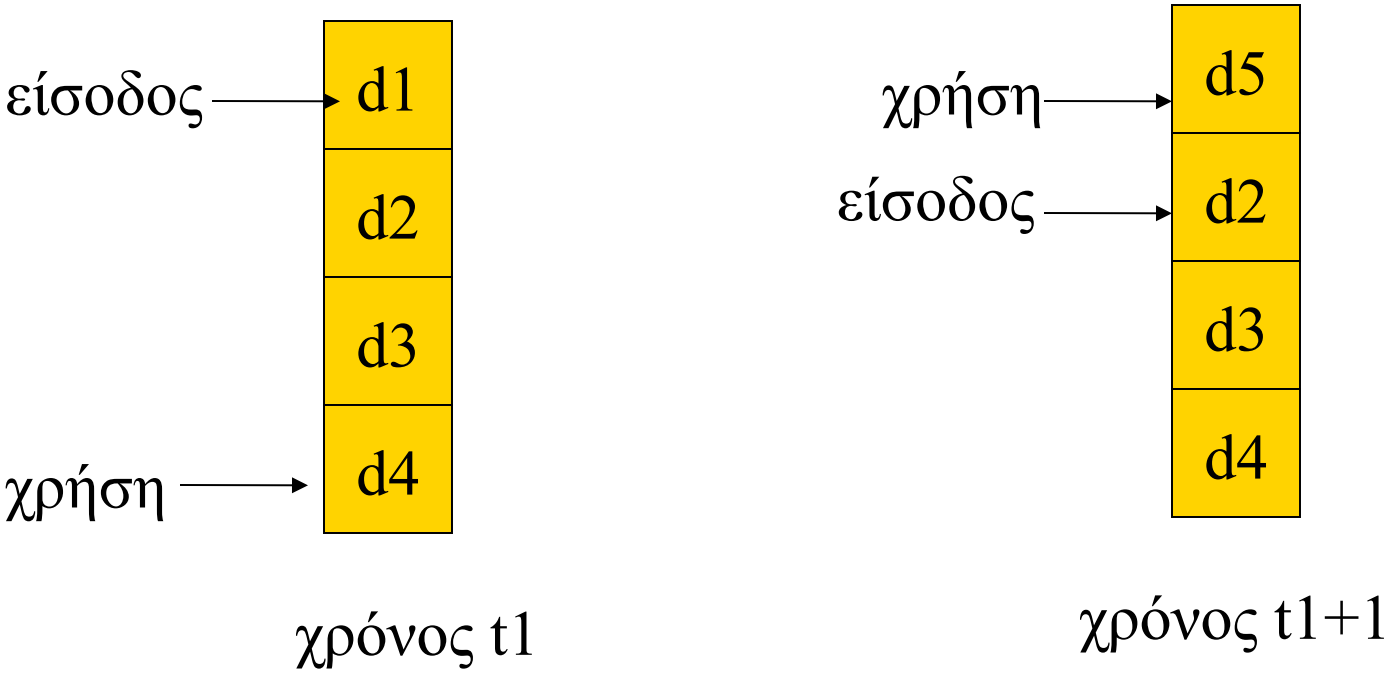
```
while (TRUE) {
      switch (state) {
            case state1: …
      }
}
```

Πίνακας κατάστασης στη C

```
switch (state) {
case A: if (in1==1) { x = a; state = B; }
  else { x = b; state = D; }
     break;
case B: if (r==0) { out2 = 1; state = B; }
  else { out1 = 0; state = C; }
     break;
case C: if (s==0) { out1 = 0; state = C; }
  else { out1 = 1; state = D; }
     break;
```

Κυκλικός buffers

- Οι δείκτες εντοπίζουν τα πρόσφατα χρησιμοποιημένα δεδομένα, τρέχοντα δεδομένα εισόδου:



Κυκλικός buffer στη C

Για να υπολογίσουμε την αξία f του φίλτρου FIR:

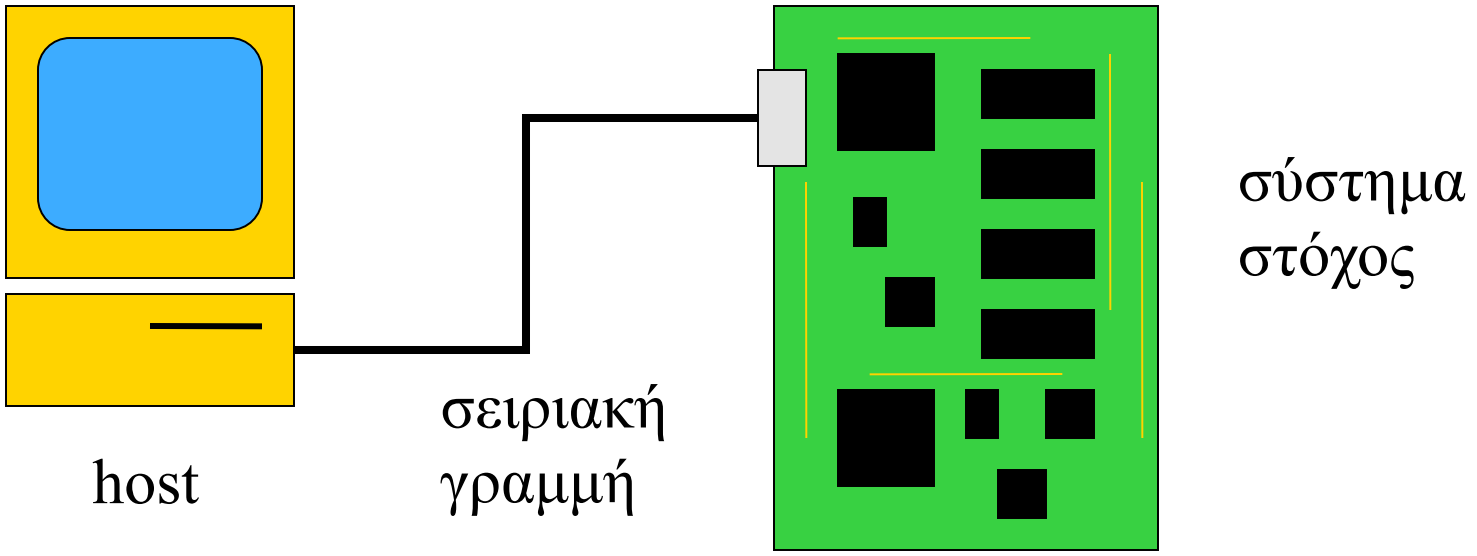
```
for (f=0, ic=0, ibuff = circ_buff_head;
     ic < N;
     ibuff = (ibuff = N-1 ? 0 : ibuff++) )
     f = f + c[ic] * circ_buff[ibuff]
```

Τεχνικές σχεδίασης λογισμικού

- Θέλει να αναπτύξει όσο το δυνατόν περισσότερο κώδικα σε μια τυποποιημένη πλατφόρμα:
  - πιο φιλικό περιβάλλον προγραμματισμού
  - ευκολότερη διόρθωση(debugging)
- Μπορεί να χρειάζεται να επινοήσει κάποια στελέχη για το λογισμικό για να επιτρέψει τη δοκιμή των στοιχείων του λογισμικού χωρίς την πλήρη πλατφόρμα υλικού/λογισμικού.

Σχυπηρεσία/στόχος host/ target

- Χρησιμοποιεί ένα σύστημα host για να προετοιμάσει λογισμικό για ένα σύστημα στόχου:



Εργαλεία βασισμένα στον host

- Cross μεταγλωτιστής:
  - συντάσσει τον κώδικα στον host για το σύστημα στόχων.
- Cross debugger:
  - επιδεικνύει την κατάσταση στόχου, επιτρέπει στο σύστημα στόχων να ελέγχεται.

Πίνακες αξιολόγησης

- Σχεδιασμένοι από τον κατασκευαστή της CPU ή άλλους.
- Περιέχει CPU, μνήμη, μερικές συσκευές I/O.
- Μπορεί να περιέχει τμήμα πρωτότυπης διαμόρφωσης.
- Ο κατασκευαστής της CPU συχνά παραδίδει netlist πίνακες αξιολόγησης - μπορεί να χρησιμοποιηθούν σαν η αφετηρία για τη σχεδίαση πινάκων της επιλογής μας.

Προσθήκη λογικής σε έναν πίνακα

- Προγραμματίσημες συσκευές λογικής (PLDs) παρέχει τη χαμηλή/μέση λογική πυκνότητα.
- Προγραμματιζόμενη στο πεδίο συστοιχία πυλών (FPGAs) παρέχει περισσότερη λογική και πολλαπλών επιπέδων λογική.
- Ολοκληρωμένα κυκλώματα οριζόμενα από εφαρμογή (ASICs) κατασκευάζονται για έναν απλό σκοπό.

Διορθώνοντας ενσωματωμένα συστήματα

- Προκλήσεις:
  - το σύστημα στόχου μπορεί είναι δύσκολο να παρατηρηθεί
  - ο στόχος μπορεί να είναι δύσκολο να ελεγχθεί
  - μπορεί να είναι δύσκολο να αναπαραχθούν πραγματικές είσοδοι
  - η ακολουθία οργάνωσης μπορεί να είναι σύνθετη

Διορθωτές λογισμικού

- Ένα πρόγραμμα ελέγχου συσκευών που βρίσκεται στο στόχο παρέχει τις βασικές λειτουργίες διορθωτών.
- Ο διορθωτής πρέπει να έχει ένα ελάχιστο ίχνος στη μνήμη.
- Ο χρήστης του προγράμματος πρέπει να είναι προσεκτικός να μην καταστρέψει το πρόγραμμα διορθωτών, αλλά, πρέπει να είναι σε θέση να ανακτηθεί από κάποια ζημία που προκαλείται από τον κώδικα του χρήστη.

Σημεία διακοπής

- Ένα σημείο διακοπής επιτρέπει στο χρήστη να σταματήσει την εκτέλεση, να εξετάσει την κατάσταση του συστήματος, και να αλλάξει την κατάσταση.
- Αντικαταστήστε την εντολή διακοπής με μια κλήση υπορουτίνας στο πρόγραμμα ελέγχου συσκευών.

Ενέργειες χειριστών σημείου παύσης

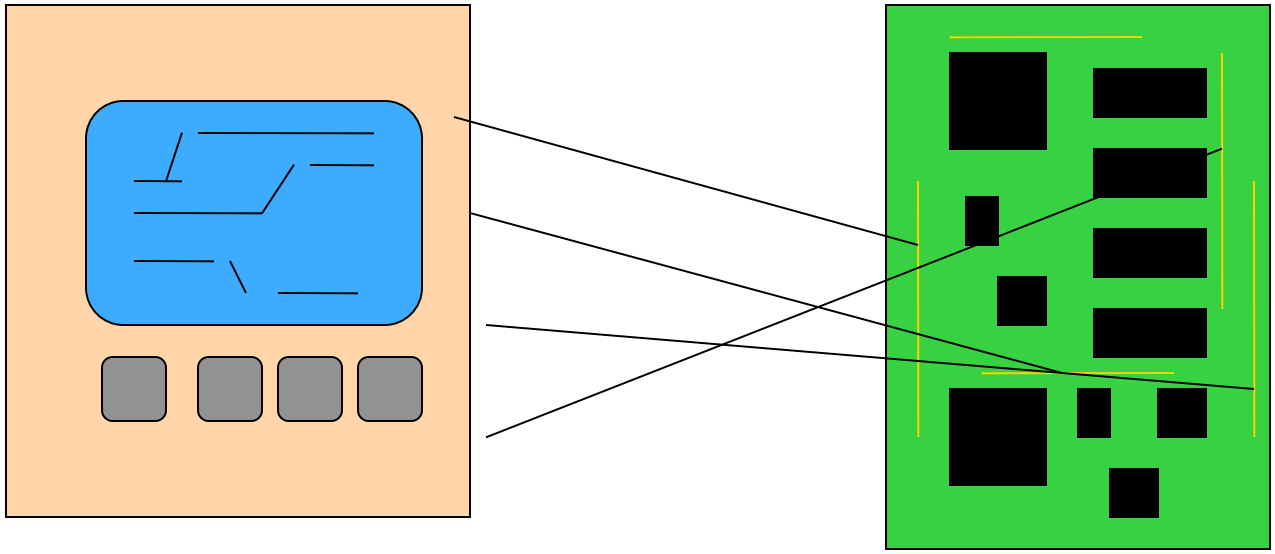
- Σώζουν τους καταχωρητές.
- Επιτρέπουν στον χρήστη να εξετάσει τη μηχανή.
- Πριν την επιστροφή, αποκαθιστούν την κατάσταση του συστήματος.
  - Ο ασφαλέστερος τρόπος να εκτελεσθεί η εντολή είναι να αντικατασταθεί και να εκτελεστεί πάλι όταν είναι έτοιμη.
  - Βάζουν ένα άλλο σημείο διακοπής μετά από το αντικατεστημένο σημείο διακοπής για να επιτρέψουν την επιδιόρθωση του αρχικού σημείου διακοπής.

Εξομοιωτές κυκλωμάτων (In-circuit emulators)

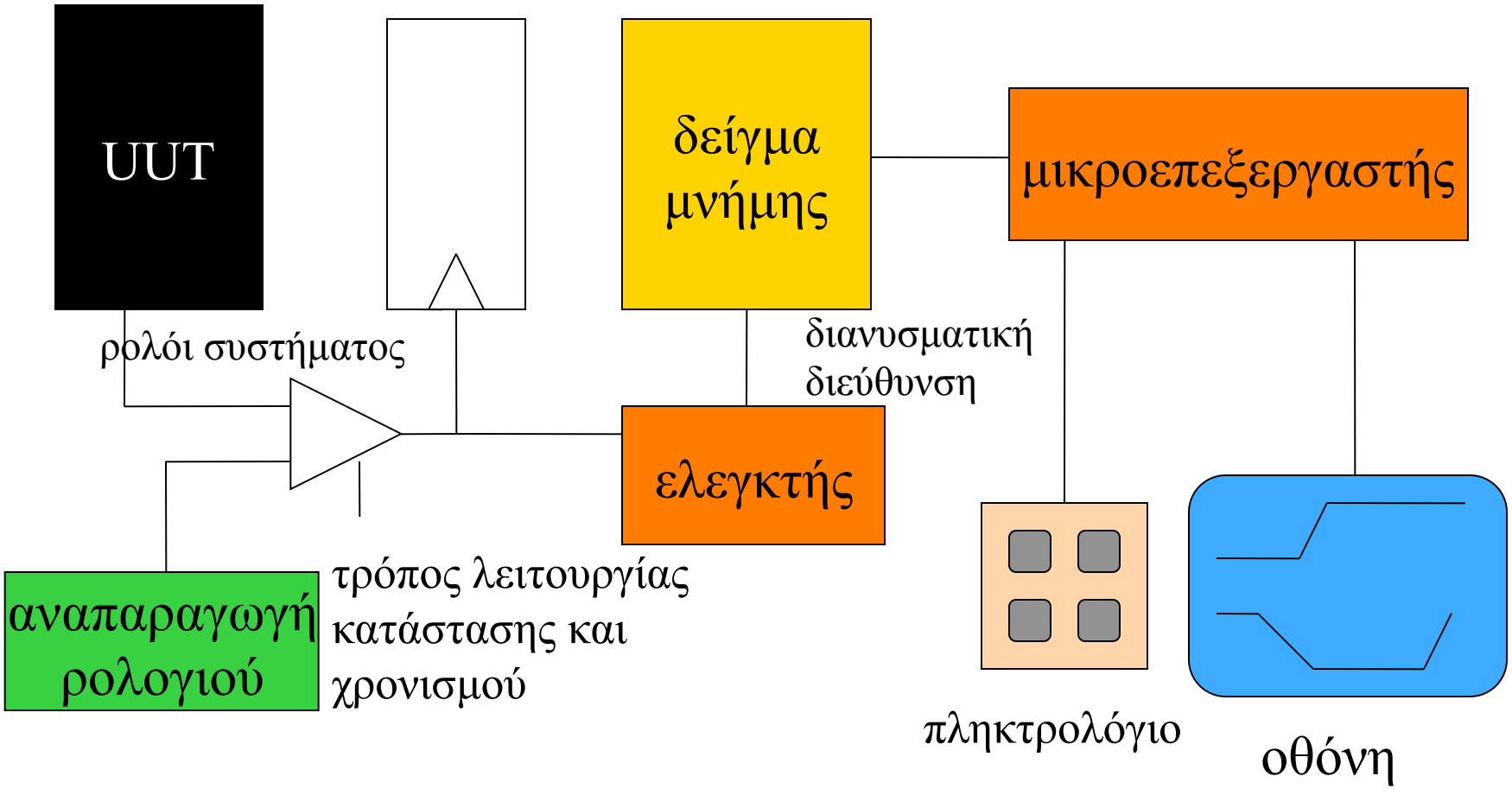
- Ένας μικροεπεξεργαστής εξομοιωτής κυκλωμάτων είναι ένας ειδικά οργανωμένος μικροεπεξεργαστής.
- Σου επιτρέπει να σταματήσεις την εκτέλεση, να εξετάσεις την κατάσταση της CPU, να τροποποιήσεις τους καταχωρητές.

Λογικοί αναλυτές

- Ένας λογικός αναλυτής είναι μια σειρά χαμηλής βαθμίδας παλμογράφων:



Αρχιτεκτονική λογικού αναλυτή



Πώς να εξετάσουμε τον κώδικα

- Εκτελούμε στο σύστημα host.
- Εκτελούμε στο σύστημα στόχου.
- Εκτελούμε σε προσομοιωτή επιπέδου εντολής .
- Εκτελούμε σε προσομοιωτή κύκλου ακριβείας.
- Εκτελούμε σε περιβάλλον συν-προσομοίωσης υλικού/λογισμικού.

Δοκιμή κατασκευής

- Στόχος: επιβεβαιώνει ότι η κατασκευή παράγει δωρεάν ατελή αντίγραφα του σχεδιασμού.
- Μπορεί να εξεταστεί από τη σύγκριση της μονάδας που εξετάζεται με την αναμενόμενη συμπεριφορά .
  - Αλλά η εκτέλεση των δοκιμών είναι ακριβή.
- Μεγιστοποιεί την εμπιστοσύνη ελαχιστοποιώντας το κόστος εξέτασης.

Έννοιες εξέτασης

- Απόδοση παραγωγής: ποσοστό των κατασκευασμένων συστημάτων που λειτουργούν.
  - Η κατάλληλη κατασκευή μεγιστοποιεί την απόδοση παραγωγής.
  - Η κατάλληλη δοκιμή υπολογίζει ακριβώς την απόδοση παραγωγής.
- Πεδίο επιστροφής: ατελής μονάδα που φεύγει από το εργοστάσιο.

Σφάλματα

- Τα κατασκευαστικά λάθη μπορεί να προκληθούν από πολλά πράγματα.
- Μοντέλο σφάλματος: το μοντέλο που προβλέπει τα αποτελέσματα ενός συγκεκριμένου τύπου σφάλματος.
- Κάλυψη σφάλματος: το ποσοστό των πιθανών σφαλμάτων που βρίσκονται από ένα σύνολο δοκιμών.
  - Η κατοχή ενός προτύπου σφάλματος μας επιτρέπει να καθορίσουμε την κάλυψη του σφάλματος.

Δοκιμή λογισμικού vs. υλικού

- Κατά τη δοκιμή του κώδικα, δεν έχουμε κανένα μοντέλο σφάλματος.
  - Πιστοποιούμε την εφαρμογή, όχι τη κατασκευή.
  - Απλές δοκιμές δουλεύουν καλά για να πιστοποιήσουν την κατασκευή του λογισμικού.
- Το υλικό απαιτεί τις δοκιμές κατασκευής επιπρόσθετα με την πιστοποίηση της εφαρμογής.

Συνδυαστική δοκιμή

- Κάθε πύλη μπορεί να είναι κολλημένη-στο-0, κολλημένη-στο-1.
- Συνήθως ελέγχουμε για απλά κολλημένα-στα-σφάλματα.
  - Ένα σφάλμα κάθε φορά.
  - Πολλαπλά σφάλματα μπορούν να καλύψουν το ένα το άλλο.
- Μπορούμε να αναπαράγουμε μία δοκιμή για μία πύλη:
  - ελέγχοντας την είσοδο της πύλης
  - παρατηρώντας την έξοδο της πύλης μέσω άλλων πυλών

Διαδοχική δοκιμή

- Μια μηχανή καταστάσεων είναι συνδυασμός λογικής + καταχωρητών.
- Η διαδοχική δοκιμή είναι αρκετά δυσκολότερη.
  - Ένα απλό κόλλημα-σε-σφάλμα επηρεάζει τη μηχανή σε κάθε κύκλο.
  - Η συμπεριφορά σφάλματος σε έναν κύκλο μπορεί να καλυφθεί από το ίδιο σφάλμα σε άλλους κύκλους.

Αλυσίδες σάρωσης

- Ένας καταχωρητής σάρωσης(scannable register) λειτουργεί σε δύο τρόπους:
  - κανονικός
  - σαρωτής - διαμορφώνει ένα στοιχείο σε έναν καταχωρητή μετατόπισης.
- Η χρήση αλυσίδων ανίχνευσης μειώνει τη διαδοχική δοκιμή σε συνδυαστική δοκιμή.
  - Η εκφόρτωση(unloading) της αλυσίδας σάρωσης είναι αργή.
  - Μπορεί να χρησιμοποιήσει μερική σάρωση.

Αναπαραγωγή

- Προγράμματα αυτόματη παραγωγή ακολουθιών δοκιμής (ATPG): παράγει ένα σύνολο δοκιμών δίνοντας τη λογική δομή.
- Μερικά σφάλματα μπορεί να μην είναι ελεγχόμενα – περιττά.
  - Η διακοπή σε ένα σφάλμα μπορεί να σημαίνει δύσκολο στη δοκιμή ή μη ελεγχόμενο.