

Συσχεδίαση Συστημάτων Υλικού Λογισμικού

Νικόλαος Βώρος

Περιεχόμενα

- Εισαγωγή
- Πρακτικές σχεδίασης συστημάτων στη βιομηχανία
- Εννοιολογικό μοντέλο σχεδίασης συστημάτων
- Βασικές αρχές γλωσσών σχεδίασης συστημάτων
- Ανάλυση απόδοσης
- Επαναχρησιμοποίηση συστημάτων
- Παράδειγμα χρήσης του εννοιολογικού μοντέλου σχεδίασης συστημάτων

Ανθρωποειδές ρομπότ σε ταξίδι στο Διάστημα

Νοέμβριο θα εκτοξευθεί το Robonaut 2

ΑΚΡΩΤΗΡΙΟ ΚΑΝΑΒΕΡΑΛ, ΦΛΟΡΙΝΤΑ.

Το ανθρωποειδές ρομπότ, με κωδική ονομασία Robonaut 2, δεν θα ταξιδέψει πρώτη θέση αφού εκτοξευθεί στο Διάστημα στα αρχές Νοεμβρίου. Αντίθετα, ο μηχανοκίνητος «βοηθός αστροναύτη» εγκιβωτίσθηκε σε ειδικό κιβώτιο μεταφοράς, με παχιά στρώση προστατευτικού αφρού για το ταξίδι του μέχρι τον Διεθνή Διαστημικό Σταθμό.

Ο Νικ Ράντφορντ, μέλος του προγράμματος Robonaut 2 ή R2 για

Προορισμός του μηχανήματος ο Διεθνής Διαστημικός Σταθμός, όπου θα υποκαθιστά τους αστροναύτες σε επικίνδυνες εργασίες.

συντομία, εξηγεί πως το ρομπωτικό μηχανήμα θα γίνει ο πρώτος ανθρωποειδής μόνιμος κάτοικος του ΔΔΣ. Το R2 διαθέτει ακόμη και δικό του λογαριασμό στην ιστοσελίδα ανταλλαγής μηνυμάτων Twitter, όπου οι άνθρωποι - διαχειριστές του δημοσιεύουν ειδήσεις και πληροφορίες γύρω από την αποστολή. «Μόλις έκλεισα την πηγή ενέργειάς μου. Την επόμενη φορά που θα την ανάψω θα είμαι στο ΔΔΣ!» έγραφε το R2 την Παρασκευή.

Το επδέξιο ρομπότ μοιάζει με αν-

θρωπο και είναι σχεδιασμένο κατά τέτοιο τρόπο, ώστε να λειτουργεί σαν άνθρωπος. Αποτελείται από κεφαλή και κορμό με δύο βραχίονες και δύο χέρια με πέντε δάχτυλα. Προηγμένη τεχνολογία ελέγχου και πολυάριθμοι αισθητήρες επιτρέπουν στο R2 να λειτουργεί ακριβώς όπως θα έκανε και ένας άνθρωπος, βοηθός των αστροναυτών του Σταθμού. Ο Ρον Ντίφτλερ, επικεφαλής του προγράμματος «ρομπωτικού αστροναύτη», λέει ότι το R2 είναι πιο επιδέξιο από τους ανθρώπους σε συνθήκες μηδενικής βαρύτητας. Σε αυτό το στάδιο, το R2 έχει περιορισμένη ικανότητα κίνησης. Με την ανάπτυξη των μηχανικών του ποδιών, όμως, το ρομπότ θα είναι ικανό να υποκαθιστά τους αστροναύτες σε επικίνδυνες εργασίες, όπως για τους διαστημικούς περιπάτους ή τις εργασίες συντήρησης στο εξωτερικό κέλυφος του Σταθμού.

Απώτερος στόχος της ΝΑΣΑ είναι η δημιουργία ενός εξ ολοκλήρου ρομπωτικού αστροναύτη, που θα μπορεί ακόμη και να ταξιδέψει εκτός πλανητικού συστήματος, στέλνοντας στοιχεία πίσω στη Γη. Η αμερικανική διαστημική υπηρεσία ευελπιστεί να ενισχύσει τις λειτουργίες του R2, που θα περιλαμβάνουν αυτόνομο σύστημα όρασης, αντί του σημερινού που επιτρέπει στο ρομπότ να μεταδίδει όσα βλέπει στο Κέντρο Ελέγχου της ΝΑΣΑ, αλλά και σύστημα φωνητικής έκφρασης.

A.P.

Ανθρωποειδές ρομπότ θα συντροφεύει τους αστροναύτες του ΔΔΣ

ROBONAUT-2

(Σχεδιασμένο από τη NASA και την GM)

Αισθητήρες: 350

Βάρος: 150 κιλά

Υψος: 102 ΕΚΑΤ. (από τη μέση μέχρι το κεφάλι)

Πλάτος ώμων: 79 ΕΚΑΤ.

Βραχίονες: Διπλάσια ευκαμψία από τους ανθρώπινους

Το Robonaut-2, που θα γίνει μέλος του πληρώματος του Διεθνούς Διαστημικού Σταθμού, δεν μοιάζει μόνο με άνθρωπο, αλλά είναι σχεδιασμένο να λειτουργεί σαν άνθρωπος. Το R2 είναι ικανό να χρησιμοποιεί εργαλεία και να αντικαθιστά αστροναύτες για τους επικίνδυνους διαστημικούς περιπάτους.

Κάτω σώμα

Τα πόδια βρίσκονται ακόμη στο στάδιο ανάπτυξης - ο R2 τοποθετήθηκε αρχικά σε κινητή βάση



Σακίδιο

Σύστημα μετατροπής επιτρέπει στις μπαταρίες του R2 να φορτίζονται από ενεργειακή πηγή στη Γη ή τον ΔΔΣ

Υλικό: Αλουμίνιο, ασάμι, κέβλαρ, τεφλόν

Όραση:

Πέντε κάμερες προσφέρουν στερεοσκοπική και υπέρυθη όραση

Τα μέρη μπορούν να αναβαθμιστούν ή να αλλάξθούν

Στολή πτήσης

Κάθε δάχτυλο έχει ισχύ 2 κιλών

Κορμός

Περιέχει 38 επεξεργαστές PowerPC που λειτουργούν χάρη στο λειτουργικό σύστημα VxWorks

Ισχύς

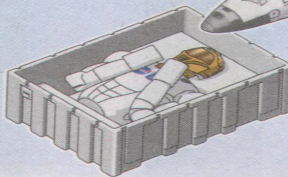
Μπορεί να σηκώσει 9,5 κιλά

Ελεγχός

Από τη Διεύθυνση Πτήσης ή σε τοπικό δίκτυο από το χειριστήριο του ΔΔΣ

Ο R2 ΣΤΟ ΔΙΑΣΤΗΜΑ

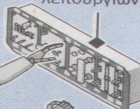
Το διαστημικό λεωφορείο Ντισκάβερι θα παραδώσει τον R2 σε ειδική συσκευασία πολλαπλών χρήσεων



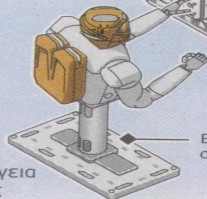
Οι δοκιμές

του R2 θα γίνουν στο εργαστήριο Ντέστιν και θα προσφέρουν στοιχεία για τις δονήσεις, τη μηδενική βαρύτητα, την έκθεση στη ραδιενέργεια και τις ηλεκτρομαγνητικές παρεμβολές

Πίνακας λειτουργιών



Βάση στήριξης



ΠΗΓΗ: NASA, GRAPHIC NEWS

Η ΚΑΘΗΜΕΡΙΝΗ

Εισαγωγή

Σήμερα, υπάρχουν σημαντικές προκλήσεις και προβλήματα στον αποδοτικό των σύγχρονων συστημάτων, εξαιτίας:

- Της γρήγορης εξέλιξης των χαρακτηριστικών των προϊόντων
 - Περισσότερη λειτουργικότητα
 - Μεγαλύτερη ποικιλία χαρακτηριστικών
 - Η διαχείριση οικογενειών προϊόντων
 - Απαίτηση για περισσότερο χρόνο για το σχεδιασμό νέων, πιο σύνθετων προϊόντων
 - Πολυπληθείς σχεδιαστικές ομάδες
- Γρήγορη εξέλιξη των τεχνολογιών
 - Συχνά απαιτούν προσαρμογή των χρησιμοποιούμενων μεθοδολογιών
- Η εμφάνιση νέων περιοχών ενδιαφέροντος
 - Απαιτούν νέες αντιλήψεις σχεδίασης π.χ. Cyberphysical Systems
- Η μείωση του TTM (time-to-market) παρά την ολοένα αυξανόμενη
 - Λειτουργικότητα
 - Ποικιλία
 - Πολυπλοκότητα
- Μειωμένο κόστος, σε μια αγορά όπου οι τιμές διαρκώς μειώνονται

Πρακτικές σχεδίασης συστημάτων στη βιομηχανία

Βασικά στάδια ενός βιομηχανικού περιβάλλοντος σχεδίασης

- Οι εταιρίες που σχεδιάζουν ενσωματωμένα συστήματα, διαθέτουν συνήθως δικές τους μεθόδους και πρακτικές και χρησιμοποιούν συγκεκριμένες γλώσσες για το σχεδιασμό συστημάτων
- Στις περισσότερες περιπτώσεις, οι πρακτικές σχεδίασης αντανακλούν το τεχνολογικό υπόβαθρο και την εμπειρία της κάθε εταιρίας

Μη τυπικός προσδιορισμός απαιτήσεων

- Ο σχεδιασμός ενός ενσωματωμένου συστήματος περιέχει τόσο τη σχεδίαση **υλικού** όσο και τη σχεδίαση **λογισμικού**
 - Συνήθως ξεκινά με ένα σύνολο *μη τυπικών απαιτήσεων*, π.χ. ένα έγγραφο που περιγράφει τις βασικές απαιτήσεις του υπό ανάπτυξη συστήματος
 - Αυτές οι μη τυπικές περιγραφές, μετατρέπονται σε τυπικές στα επόμενα στάδια της διαδικασίας σχεδίασης

Τυπικές προδιαγραφές του συστήματος

- Οι τυπικές προδιαγραφές του συστήματος δημιουργούνται χρησιμοποιώντας κατάλληλες τυπικές γλώσσες σχεδίασης
- Βασίζονται στη μη τυπική περιγραφή του συστήματος
- Σε αρκετές περιπτώσεις γίνεται συνδυασμένη χρήση πολλαπλών φορμαλισμών, ώστε να περιγράψουμε πιο αποδοτικά διαφορετικά τμήματα του ίδιου συστήματος
- Αυτό γίνεται επειδή:
 - ένας συγκεκριμένος φορμαλισμός μπορεί να περιγράψει πιο αποδοτικά συγκεκριμένα μέρη του συστήματος, ή επειδή
 - Υπάρχουν υλοποιημένα τμήματα του συστήματος τα οποία μπορούν να επαναχρησιμοποιηθούν

Διερεύνηση αρχιτεκτονικής και επιμερισμός υλικού/λογισμικού

- Η διερεύνηση της αρχιτεκτονικής του συστήματος και ο επιμερισμός υλικού/λογισμικού βασίζονται συνήθως στην εμπειρία των σχεδιαστών
- Βασίζονται συνήθως σε μη τυπικές περιγραφές του συστήματος, π.χ. block diagrams
- Η τελική αρχιτεκτονική του συστήματος προκύπτει ως αποτέλεσμα επαναλαμβανόμενων κύκλων δοκιμών και σφαλμάτων (trial and error iterations)
- Η χρήση μη τυπικών περιγραφών, συνήθως οφείλεται σε
 - έλλειψη τυπικών περιγραφών ή
 - σε κακή αφάιρηση (*bad abstraction*) της περιγραφής του συστήματος (όχι αμιγώς λειτουργικές περιγραφές αλλά περιγραφές με πολλές λεπτομέρειες υλοποίησης)

Παράλληλη ανάπτυξη υλικού και λογισμικού

- Το επόμενο βήμα είναι η παράλληλη (και σχεδόν ανεξάρτητη) ανάπτυξη υλικού και λογισμικού
- Η απαιτούμενη αλληλεπίδραση μεταξύ της ομάδας σχεδίασης υλικού και της ομάδας σχεδίασης λογισμικού γίνεται μέσω μη τυπικών προδιαγραφών
- Τίποτα δεν εξασφαλίζει την έλλειψη παρανοήσεων ανάμεσα στις δύο ομάδες σχεδίασης
- Η ανάπτυξη του υλικού γίνεται με τη χρήση HDLs (Hardware Description Languages) και εμπορικών εργαλείων σύνθεσης
- Η ανάπτυξη του λογισμικού γίνεται με τη χρήση κάποιας τυπικής γλώσσας, και στη συνέχεια μεταφράζεται είτε σε C είτε σε κάποια άλλη γλώσσα υλοποίησης λογισμικού

Παράλληλη ανάπτυξη υλικού και λογισμικού(2)

- Το επόμενο βήμα είναι η ανάπτυξη λογισμικού χρησιμοποιώντας
 - Αλγοριθμική προσομοίωση των τμημάτων του λογισμικού χρησιμοποιώντας compilers, assemblers κλπ
 - Η ανάπτυξη προγραμμάτων οδήγησης (driver development) και εξακρίβωση της λειτουργικότητάς τους
 - ✓ Πραγματοποιείται συνήθως με χρήση instruction set simulators, οι οποίοι έχουν επεκταθεί με εικονικά μοντέλα υλικού.
- Στόχος της παράλληλης ανάπτυξης υλικού και λογισμικού είναι
 - Να παραχθεί λογισμικό το οποίο θα μπορεί να εκτελείται σε έναν συγκεκριμένο μικροεπεξεργαστή και
 - Το εξειδικευμένο υλικό του συστήματος να είναι διαθέσιμο ως SoC (System-on-a-Chip).

Ανάλυση της ροής σχεδίασης

- Ο ρόλος του αρχιτέκτονα σχεδίασης είναι κομβικός για τη διαδικασία σχεδίασης
- Απόκτηση τεχνογνωσίας σχεδίασης
- Τεχνογνωσία για το πως παράγονται υψηλής ποιότητας σχέδια του συστήματος με βάση τις ροές σχεδίασης που ακολουθούνται
- Παραδείγματα
 - Scripts
 - Tools
 - Δραστηριότητες σχεδίασης
 - Ανάμιξη των ίδιων ατόμων (με διαφορετικούς ρόλους) σε διαφορετικές δραστηριότητες

Επαναχρησιμοποίηση IP

- Η επαναχρησιμοποίηση IP (Intellectual property), συμβαίνει, αλλά δεν είναι εγγυημένη
- Η έννοια της επαναχρησιμοποίησης υπάρχει στους σχεδιαστές συστημάτων, αλλά σπάνια υπάρχει σε επίπεδο επιχείρησης
 - Συνεπώς οι ευκαιρίες για επαναχρησιμοποίηση δεν μπορούν να αξιολογηθούν σε επίπεδο εταιρίας, ούτε ποσοτικά ούτε ποιοτικά
- Στις περισσότερες περιπτώσεις, τα IPs είναι τα αποτελέσματα παλαιότερων έργων ή προϊόντων, τα οποία έχουν σχεδιαστεί χωρίς την προοπτική της επαναχρησιμοποίησης
- Οι πολιτικές που καθορίζουν εάν ένα IP πρέπει να συμπεριληφθεί στην αποθήκη με τα IPs της εταιρίας δεν είναι πάντα προφανείς ή διαθέσιμες

Επαναχρησιμοποίηση IP (2)

Τυπικά παραδείγματα IPs είναι τα ακόλουθα:

- Αλγόριθμοι DSP, ή πιο γενικά, προδιαγραφές λειτουργικών τμημάτων του συστήματος
- Υποσυστήματα με σαφώς ορισμένες διεπαφές
- Μικροελεγκτές
- DSPs
- Μνήμες, κλπ

Διεπαφές

- Κατά τη δομική ανάλυση (structured-analysis) και το σχεδιασμό της αρχιτεκτονικής του, οι διεπαφές των συστατικών του είναι βαρύνουσας σημασίας
- Στην πλειοψηφία των περιπτώσεων, αρκετά σφάλματα στις διεπαφές εντοπίζονται κατά τη φάση της ολοκλήρωσης του συστήματος (Plug & Pray 😊)
- Η χρήση τεχνικών συν-εξομοίωσης υλικού λογισμικού συμβάλλει σημαντικά στη μείωση του κόστους διόρθωσης των συγκεκριμένων σφαλμάτων

- Συνεπώς, οι διεπαφές πρέπει να είναι από τις πρώτες προτεραιότητες στη ροή σχεδίασης του συστήματος, πρέπει να είναι σαφείς και να προσδιορίζονται όσο το δυνατόν νωρίτερα

Επικύρωση του συστήματος

- Είναι σημαντικό να επικυρώνονται οι λειτουργίες των συστημάτων (system validation) όσο το δυνατόν νωρίτερα στη ροή σχεδίασης του συστήματος διότι
 - Τα σφάλματα που προκύπτουν κατά την ολοκλήρωση του συστήματος ελαχιστοποιούνται
 - Τα εννοιολογικά σφάλματα εντοπίζονται σχετικά νωρίς
- Τα εκτελέσιμα μοντέλα του συστήματος σε υψηλό επίπεδο επιτρέπουν να γίνει επικύρωση των λειτουργιών του συστήματος αρκετά νωρίς στη ροή σχεδίασης, δηλαδή *πολύ πριν την έναρξη της υλοποίησης*
 - Τα συγκεκριμένα μοντέλα πρέπει να είναι εύκολα στο σχεδιασμό τους και να αναπτύσσονται γρήγορα

Παράλληλη Μηχανική

- Η ροή και η διαδικασία σχεδίασης είναι από-επάνω-προς-τα-κάτω (top-down)
 - Αλλά περιέχει σε μεγάλο βαθμό παράλληλη μηχανική (concurrent engineering)
- Το κλασικό παράδειγμα είναι η παράλληλη ανάπτυξη υλικού και λογισμικού
- Οι φάσεις της σχεδίασης του συστήματος και της αρχιτεκτονικής σχεδίασης εκτελούνται παράλληλα με τη φάση σχεδίασης υλικού και τη φάση σχεδίασης λογισμικού
- Σε αρκετές περιπτώσεις, ο σχεδιασμός υλικού μπορεί να ξεκινήσει αρκετά νωρίτερα
 - για παράδειγμα ο σχεδιασμός μιας πλατφόρμας ανάπτυξης

Επιμερισμός υλικού/λογισμικού

- Στην ανάλυση της ροής σχεδίασης, ο σχεδιασμός υλικού και ο σχεδιασμός λογισμικού είναι ξεχωριστές δραστηριότητες σχεδίασης και ανάπτυξης
 - και συνήθως εκτελούνται από από διαφορετικές ομάδες
- Ωστόσο, κατά τα αρχικά στάδια του σχεδιασμού του συστήματος και της αρχιτεκτονικής, ο επιμερισμός υλικού/λογισμικού δεν είναι μια σαφώς ορισμένη δραστηριότητα
- Ο ορισμός της λειτουργικής και της φυσικής αρχιτεκτονικής είναι ιδιαίτερα σημαντικές για το σχεδιασμό του συστήματος
 - Είναι ωστόσο ξεχωριστές η μια από την άλλη

Επιμερισμός υλικού/λογισμικού (2)

- Ο επιμερισμός υλικού/λογισμικού είναι μια *μη τυπική* και *διαισθητική* διαδικασία που εξαρτάται
 - από τη σχεδιαστική κουλτούρα της ομάδας σχεδίασης
 - και από το είδος του συστήματος που αναπτύσσεται
- Ο επιμερισμός υλικού/λογισμικού είναι περισσότερο προφανής στη διαδικασία σχεδίασης όταν αναφερόμαστε σε πλατφόρμες υλικού στις οποίες θα εκτελεστεί το λογισμικό του υπό ανάπτυξη
- Συνήθως ο επιμερισμός υλικού/λογισμικού δεν είναι επαρκώς λεπτομερής
 - Κυρίως ενδιαφέρουν τα χαρακτηριστικά που θα οδηγήσουν στον ορισμό της πλατφόρμας ανάπτυξης και των επεξεργαστών που θα χρησιμοποιηθούν
- Ο ακριβής ορισμός των διεπαφών υλικού/λογισμικού μπορεί να θεωρηθεί ως ένα επιπλέον σχεδιαστικό βήμα, που μπορεί, εν μέρει, να θεωρηθεί τμήμα της διαδικασίας επιμερισμού υλικού/λογισμικού

Έλεγχος και Αποσφαλμάτωση

- Ο **σχεδιασμός-για-έλεγχο** (design-for-test) και ο **σχεδιασμός-για-αποσφαλμάτωση** (design-for-debug) θα πρέπει να λαμβάνονται υπόψη σε όλες τις φάσεις της σχεδίασης, μέχρι την υλοποίηση
- Ο έλεγχος και η αποσφαλμάτωση είναι ορθογώνιες με τις διάφορες σχεδιαστικές φάσεις
- Τόσο ο έλεγχος όσο και η αποσφαλμάτωση έχουν σημαντικό ρόλο στη διαδικασία σχεδίασης
 - Αν και σε αρκετές περιπτώσεις δεν εκτιμάται σωστά η σημασία και η πολυπλοκότητά τους

Βελτιώσεις

- Η αρχιτεκτονική του συστήματος (σε όλες τις παραλλαγές της) είναι η βασική περιοχή που χρειάζονται βελτιώσεις
- Τα συστήματα πρέπει να επικυρώνονται για την ορθότητα της λειτουργίας τους και την απόδοσή τους πολύ νωρίτερα απ' ότι γίνεται σήμερα
 - Οι λύση βρίσκεται στις **προδιαγραφές με δυνατότητα εκτέλεσης (executable specifications)**
- Η επαναχρησιμοποίηση πρέπει να γίνεται με σαφώς καθορισμένα κριτήρια, και σε επίπεδο εταιρίας (όχι σε επίπεδο ομάδας ή τμήματος)
 - Αυτό συνήθως απαιτεί οργανωτικές αλλαγές στην εταιρία και επενδύσεις σε υποδομές
 - Είναι επίσης σημαντικό να υπάρχει δυνατότητα **έκφρασης και αξιολόγησης** της ποιότητας των σχεδίων που επαναχρησιμοποιούνται

Βελτιώσεις (2)

- Το εύρος των υποσυστημάτων που χρησιμοποιούνται πρέπει να αυξηθεί
 - Σήμερα επαναχρησιμοποιούνται μόνο υποσυστήματα που σχετίζονται με την υλοποίηση
- Θα υπάρχει μεγαλύτερο κέρδος στην παραγωγικότητα της ομάδας σχεδίασης, σε σημαντική μείωση του TTM, εάν η επαναχρησιμοποίηση εισαχθεί και στα πρώτα σχεδιαστικά στάδια
- Οι σύγχρονες γλώσσες σχεδιασμού είναι αρκετά εκφραστικές
 - σε αρκετές περιπτώσεις όμως θέτουν περιορισμούς στο τι μπορεί να εκφραστεί και πως
- Θα πρέπει να οριστούν οδηγίες, συμβάσεις και υποσύνολα γλωσσών τα οποία μπορούν να χρησιμοποιηθούν στις διάφορες φάσεις της σχεδίασης

Παραδείγματα σχεδιασμού συστημάτων

Ψηφιακοί τηλεφωνικοί μεταγωγείς

- Οι περίπλοκοι μεταγωγείς είναι ιδιαίτερα αξιόπιστοι:
 - διακοπή 30 δευτερολέπτων ανά χρόνο.
- Οι επιχειρήσεις, τα σπίτια εγκαθιστούν ιδιωτικά τηλεφωνικά κέντρα (PBXs):
 - χαρακτηριστικά γνωρίσματα εσωτερικού συστήματος επικοινωνιών
 - διαχείριση των δαπανών μεγάλης απόστασης.

Συστήματα τηλεφωνικής μεταγωγής

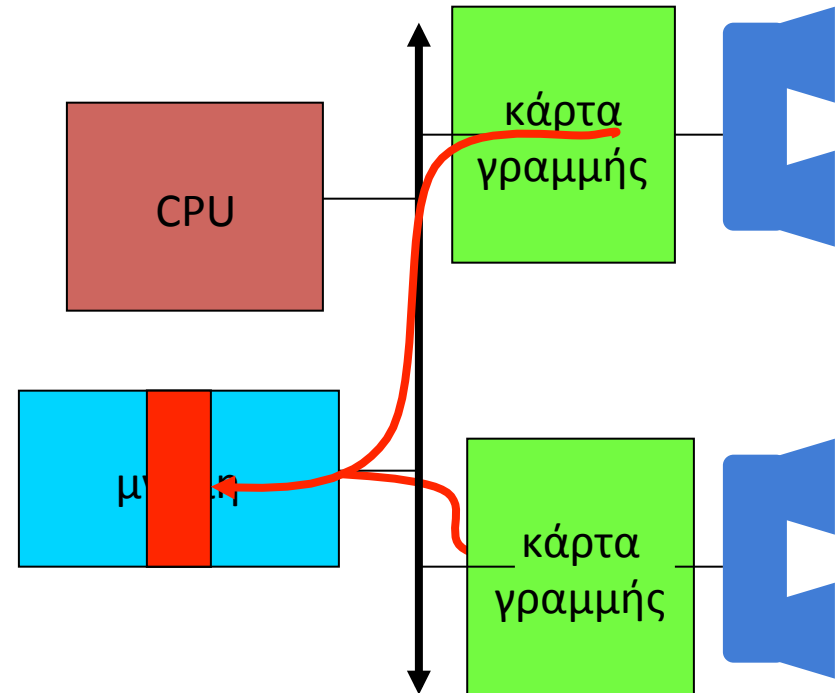
- Αποκαθιστούν τις κλήσεις:
 - μέσα στο μεταγωγέα, βρίσκουν άλλη τηλεφωνική γραμμή
 - έξω από τη μεταγωγή, βρίσκουν διαδρομή σε άλλη γραμμή.
- Δρομολογούν δείγματα φωνής μεταξύ των τηλεφώνων.
- Μετρούν το χρόνο κλήσης για την τιμολόγηση.
- Παρέχουν πρόσβαση συντήρησης στο μεταγωγέα.

Τηλεφωνική ορολογία

- **Γραμμή**: ευδιάκριτη τηλεφωνική σύνδεση.
- **Κάρτα γραμμής**: PBX/ διεπαφή γραμμής συνδρομητή.
- **Off-hook**: ενεργό τηλέφωνο.
- **On-hook**: ανενεργό τηλέφωνο.
- **Γραμμή κορμού**: τηλεφωνικές γραμμές μεταξύ μεταγωγέων.
- **POTS**: παλιά τηλεφωνική υπηρεσία (αναλογική, όχι κάποια σπουδαία υπηρεσία).

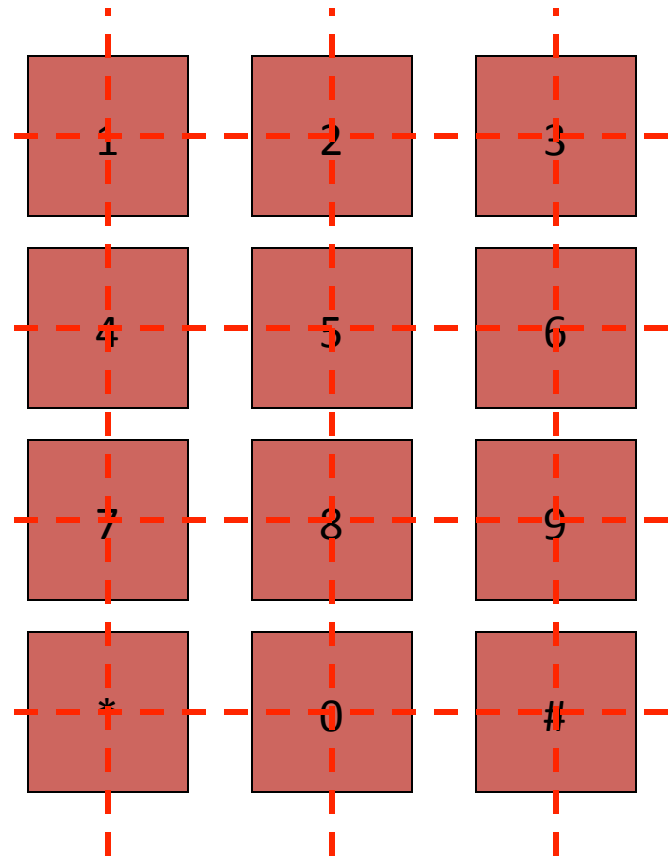
Τηλεφωνική μεταγωγή ελεγχόμενη από υπολογιστή

- Ποσοστά δεδομένων φωνής:
 - Δείγμα 8 bits (μ ή A law)
 - 125 μ s περίοδος (8 kHz).
- Τα τηλέφωνα είναι συσκευές I/O.
- Ο δίαυλος του υπολογιστή είναι μεταγωγέας.

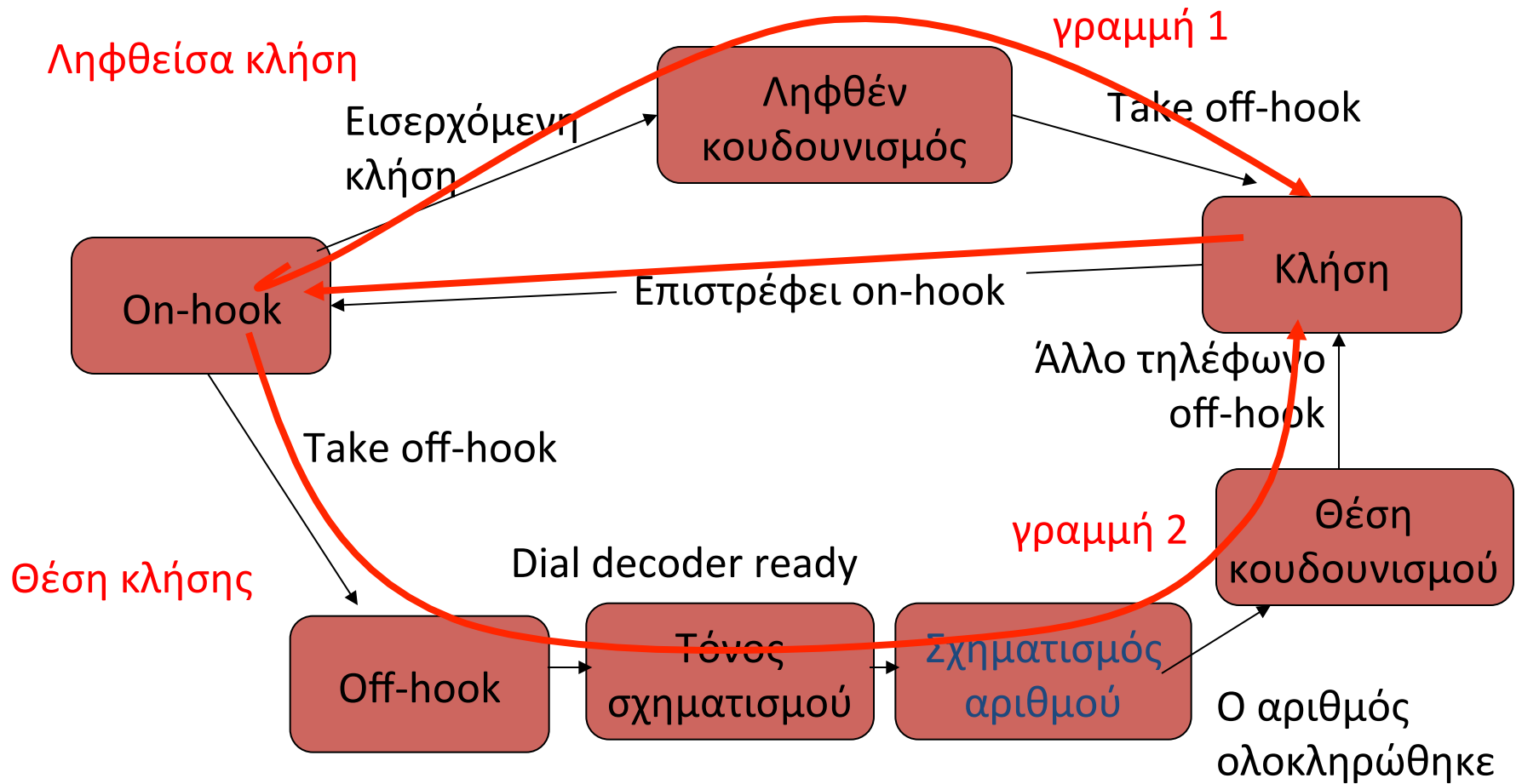


Σχηματισμός αριθμού

- Διπλός τόνος πολλαπλής συχνότητας (DTMF): οι τόνοι καθορίζουν τη σειρά, τη στήλη του πλήκτρου.
- Πρέπει να μείνει πατημένο για 0.1 δευτερόλεπτα.

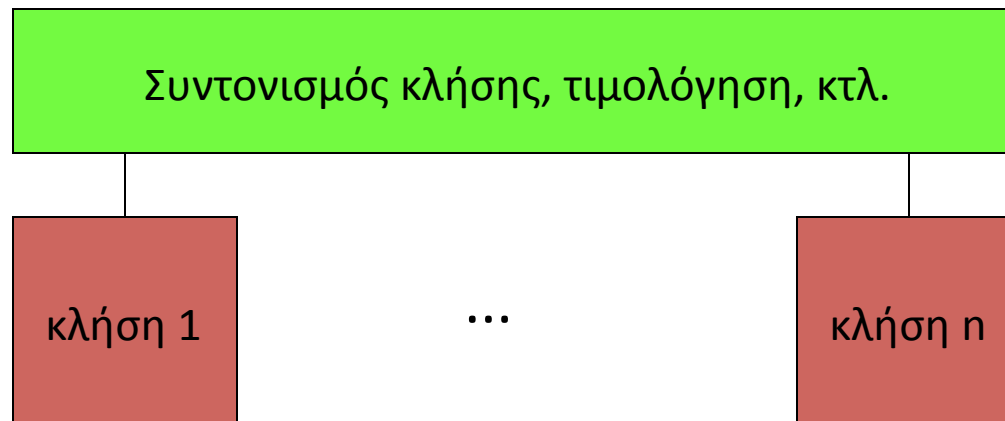


Καταστάσεις κλήσεων



Μοντέλο PBX

- Οι κλήσεις είναι παράλληλες διεργασίες:

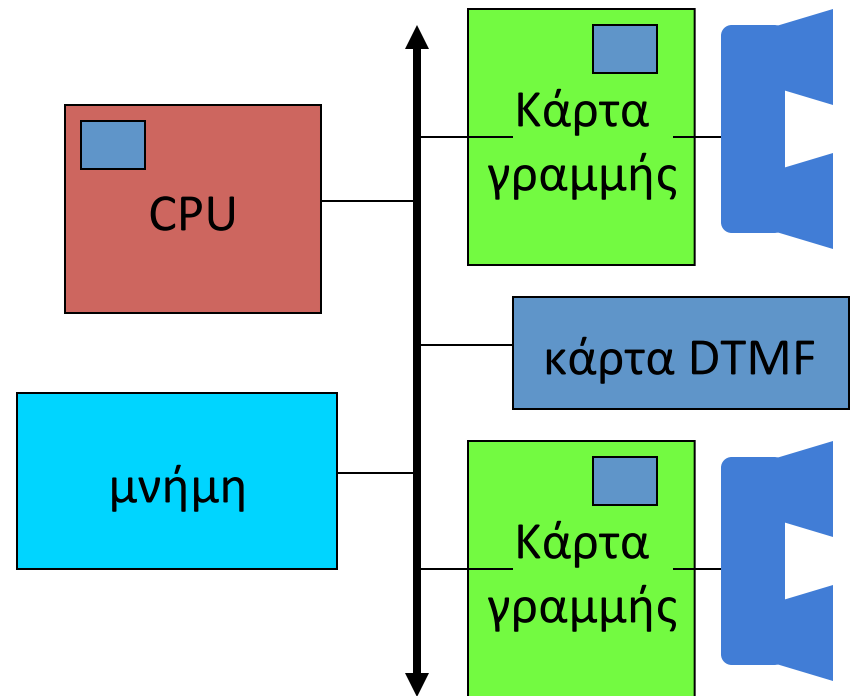


Αρχιτεκτονική συστήματος Tigerswitch

- Πλατφόρμα PC
 - Αλλάζει τις κλήσεις μέσω διαύλου ISA.
- Οι κάρτες γραμμής είναι παραδοσιακές συσκευές.

Εντοπισμός του DTMF

- Διεργασία λογισμικού στη CPU.
 - Χρησιμοποιεί πολύ από τον χρόνο της CPU.
- Το αναλογικό φίλτρο στηρίζεται στις κάρτες γραμμής.
 - Ακριβό στον όγκο.
- DSP σε ξεχωριστή κάρτα.
 - Απαιτεί καινούρια σχεδίαση.



Μεταγωγή

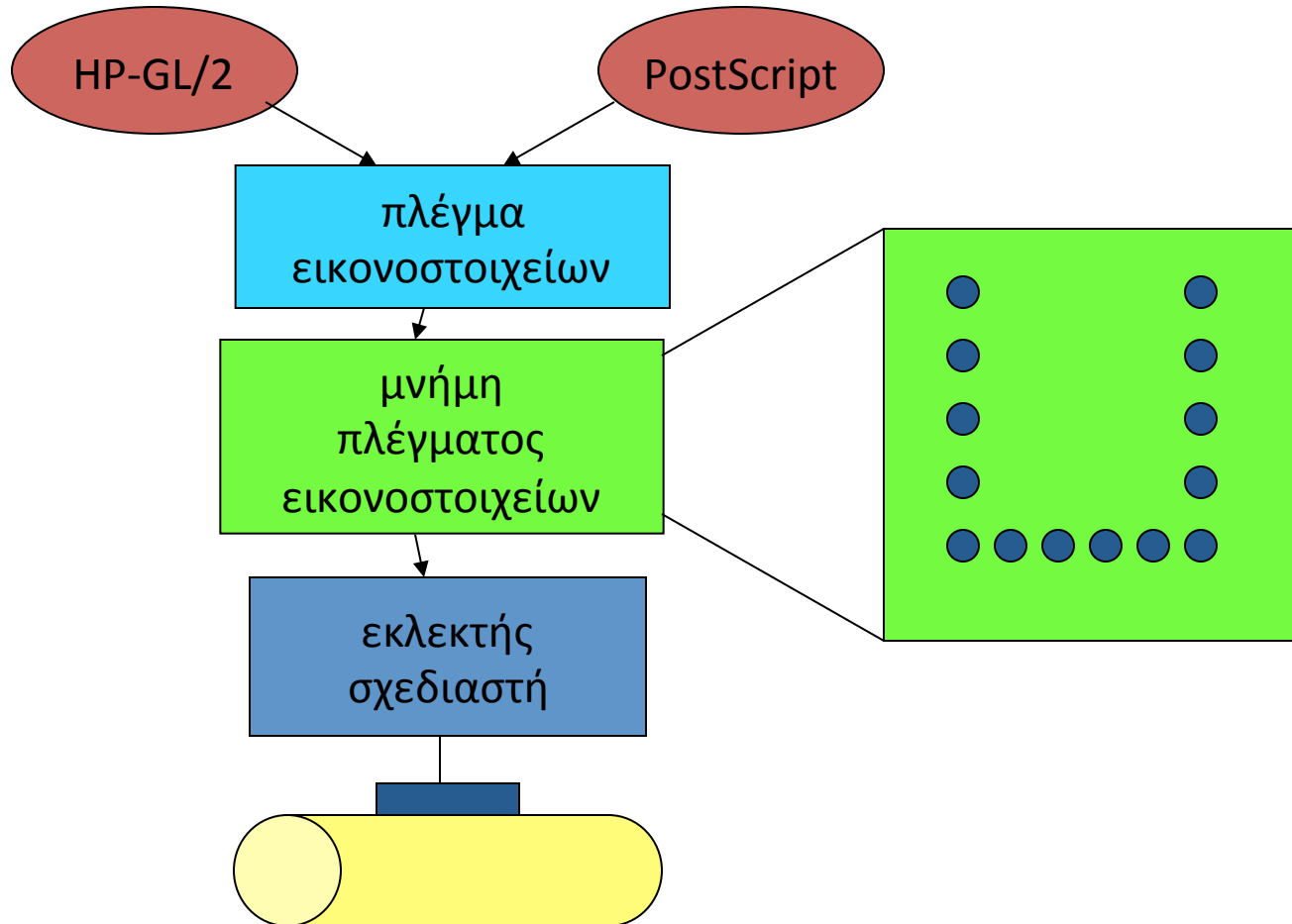
- Η διεργασία ανά κλήση είναι κακή εφαρμογή.
 - Το περιεχόμενο αλλάζει ανά κλήση ανά 125 μ s.
- Ξετυλίγει τις κλήσεις σε έναν βρόγχο.
 - Κάθε επανάληψη βρόγχου είναι μια κλήση για ένα δείγμα.

```
for (i=0; i<n_calls; i++) {  
    /* from 1 to 2 */  
    data = read(call[i].line1);  
    write(call[i].line2,data);  
    /* from 2 to 1 */  
    data = read(call[i].line2);  
    write(call[i].line1,data);  
}
```

Σχεδιασμός HP DesignJet

- Σχεδιάζει μέχρι 36 cm πλάτος σε 300 DPI.
- Συνδυάζει μία ποικιλία εργασιών:
 - επικοινωνία υπηρεσιών
 - γλωσσική ερμηνεία γραφικής παράστασης
 - μετατροπή εικόνας σε οθόνη
 - έλεγχος συσκευής.

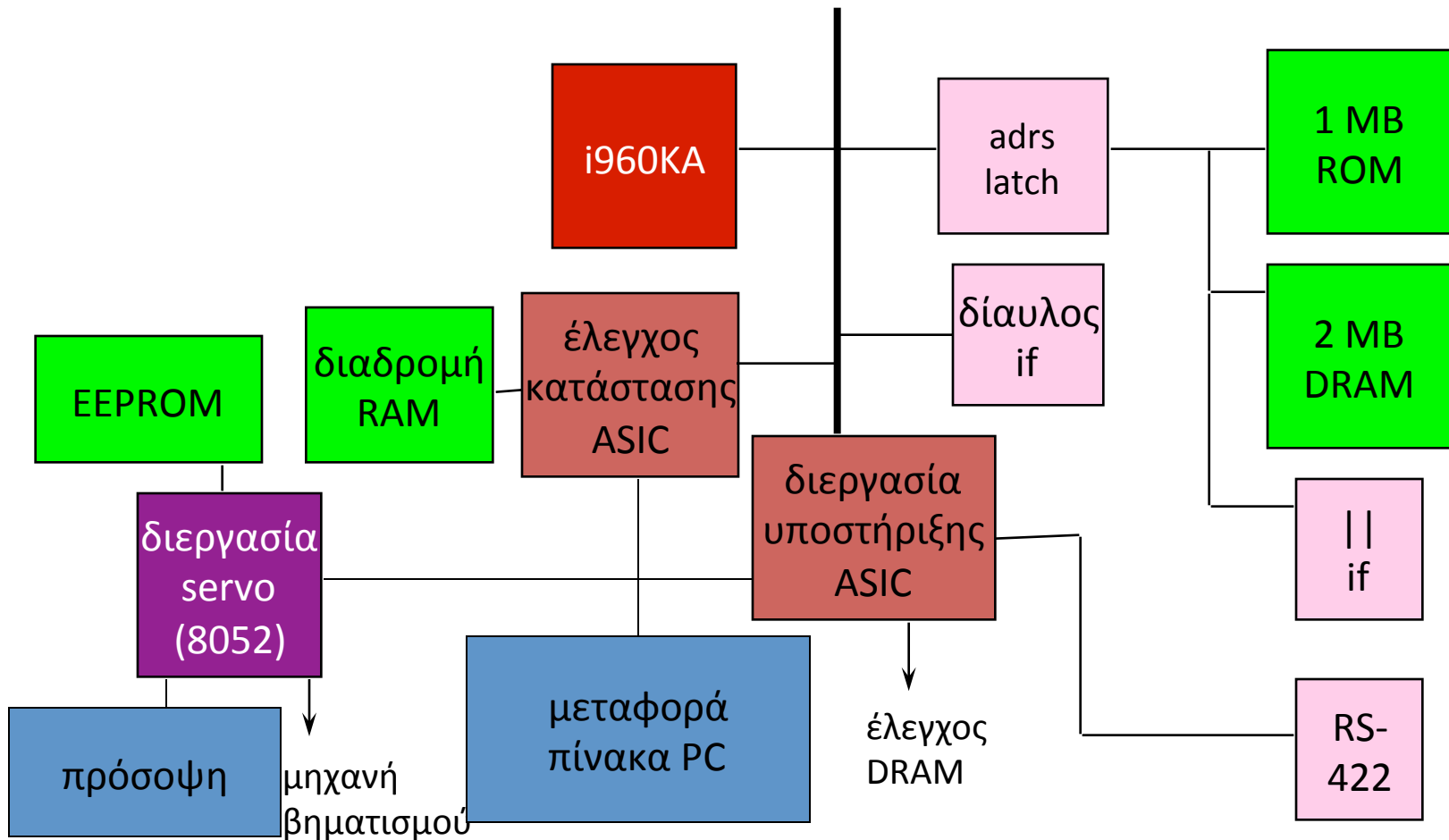
Η δομή της διεργασίας



Εκτιμήσεις σχεδίασης

- Η χρησιμότητα της μνήμης είναι σημαντική.
 - 36 cm X μέγεθος X 300 DPI X n bits/pixel είναι πολύ μνήμη.
 - Απαιτεί έξυπνους αλγορίθμους για να ελαχιστοποιήσει τις απαιτήσεις μνήμης πλέγματος εικονοστοιχείων.
- Απαιτεί έλεγχο σε πραγματικό χρόνο.
- Απαιτεί πολλές ταυτόχρονες λειτουργίες: να διαβάζει τα νέα δεδομένα, να δημιουργεί πλέγμα εικονοστοιχείων, να ελέγχει την κεφαλή της εκτύπωσης.

Αρχιτεκτονική υλικού του HP DesignJet



Πρόωρες αρχιτεκτονικές αποφάσεις

- Επιλέγουμε τον Intel 80960KA ως κύριο επεξεργαστή.
 - Καλή ανάλυση, έλεγχος μετατροπής εικόνας σε οθόνη, έλεγχος μηχανής εκτύπωσης.
 - Δίαυλος πολλαπλών λειτουργιών που μειώνει το πλήθος των τερματικών.
 - Εάν είναι απαραίτητο, θα μπορούσε να αναβαθμιστεί σε ώστε να υπολογίζει με αριθμητική κινητής υποδιαστολής.
- Χρησιμοποιούμε διαμορφωμένες I/O
- Δεν χρησιμοποιούμε δίσκο για τοπική αποθήκευση.

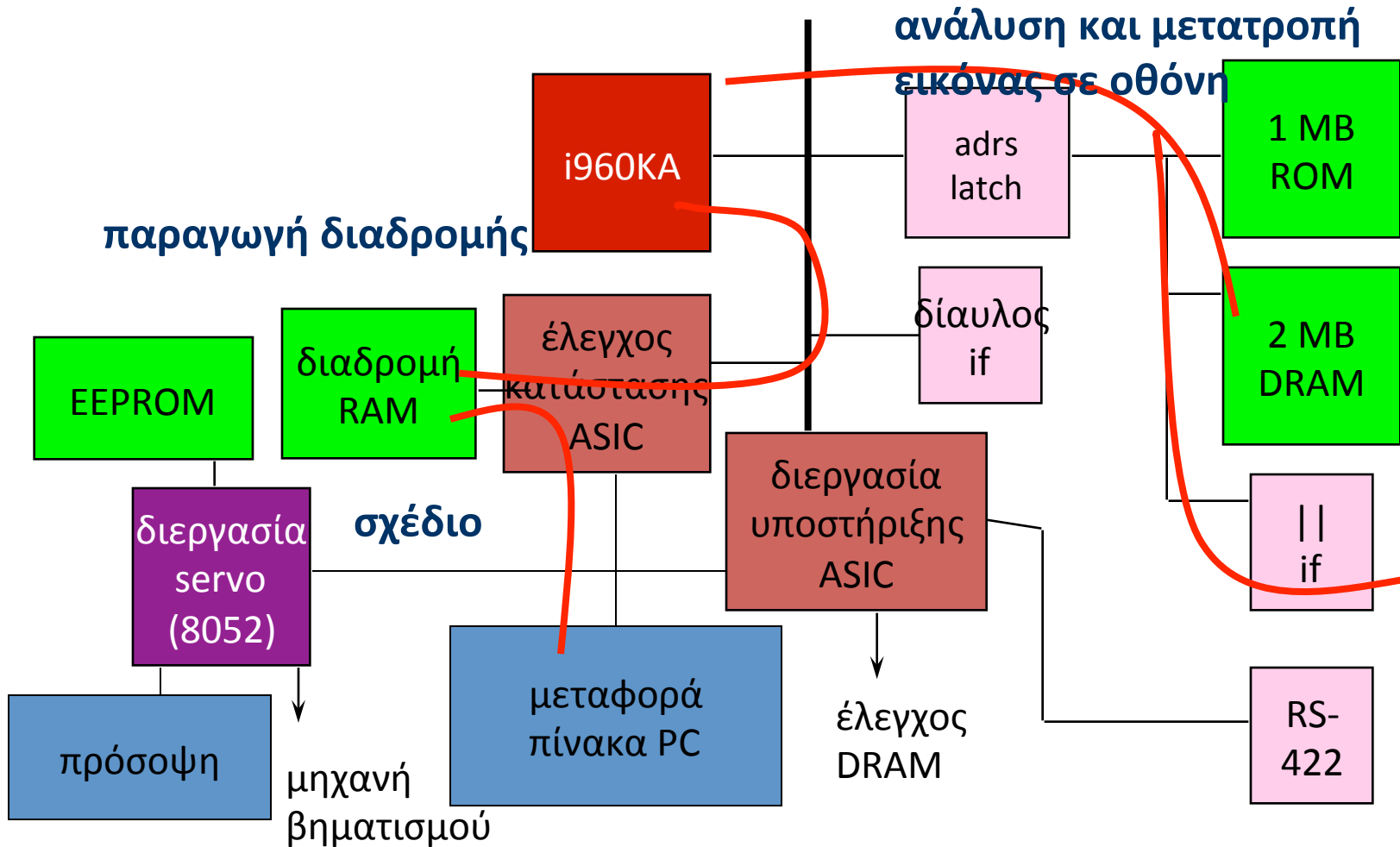
Περιεχόμενα συστήματος

- 2 MB RAM (SIMM για περισσότερες υποδοχές).
- Τρία ASICs:
 - διεπαφή κατάστασης
 - υποστήριξη επεξεργαστή
 - μεταφορά.
- Η επεξεργασία Servo εκτελείται από τον μικροελεγκτή 8052.

Μετατροπή εικόνας σε οθόνη

- Η σχεδίαση παράγεται στις **διαδρομές**.
 - Ξεχωρίζει τη μνήμη της διαδρομής.
- Τα pixels παράγονται σε γραμμές από τον κύριο επεξεργαστή.
- Τα pixels μετατρέπονται σε στήλες.
- Η διεπαφή κατάστασης ASIC μετασχηματίζει τη γραμμή σε στήλη.

Ροές δεδομένων



Λειτουργίες

- Ο επεξεργαστής Servo ελέγχει τη μηχανή βηματισμού.
- Ο επεξεργαστής μεταφορών πρέπει να γράψει, να διαβάσει τα σημάδια ευθυγράμμισης της διαδρομής.
- Η υποστήριξη των ASIC παρέχει πολλαπλές λειτουργίες: διακοπή και επικοινωνία.
- Ο ελεγκτής κινήσεως αποκωδικοποιεί τη θέση εκτύπωσης της μεταφοράς και του εγγράφου.

Διεπαφή διαδρομής του ASIC

- Διεπαφές στον δίαυλο i960, μνήμη διαδρομής, μεταφορά του ASIC.
- Η διεπαφή της διαδρομής διαβάζει τα pixels από τη διαδρομή σε προκαθορισμένη ακολουθία χρησιμοποιώντας γεννήτρια ακολουθίας διεύθυνσης.
- Πρέπει να υποστηρίξει την αμφίδρομη εκτύπωση δεδομένου ότι η κεφαλή τυπώνει και τους δύο τρόπους.

Μεταφορά του ASIC

- Διεπαφές στο δίαυλο του επεξεργαστή, διεπαφή της διαδρομής του ASIC, ελεγκτής servo.
- Διαβάζει τους καταχωρητές ελέγχου χρονισμού χρησιμοποιώντας τον δίαυλο της CPU.
- Οι καταχωρητές καθυστέρησης προσθέτουν τη διόρθωση για την ευθυγράμμιση των διαδρομών.

Διεργασία ανάπτυξης

- Ο αλγόριθμος μετατόπισης του pixel για τη διεπαφή της διαδρομής/ της μεταφοράς των ASIC ήταν αρχικά υλοποιημένος σε C.
- Κατασκευάζονται εξομοιωτές για τα ASIC για να επιτρέψουν την **παράλληλη** ανάπτυξη του λογισμικού και του υλικού του 1960.

Περιβάλλον ανάπτυξης υλικού

- Το λογισμικό των σχεδιαστών θα μπορούσε να υλοποιηθεί στον τερματικό σταθμό του Unix ή την τελική πλατφόρμα.
 - Διαφοροποιημένο I/O στα υποσυστήματα μηχανών εκτύπωσης.
 - Η μηχανή εκτύπωσης μιμείται στον host με τη διεπαφή παραθύρου X παρουσιάζοντας την κατάσταση της διαδρομής.
- Χρησιμοποιεί εσωτερικό RTOS.
- Ο αναλυτής HP-GL/2 είναι κώδικας που **επαναχρησιμοποιείται.**

Περιβάλλον ανάπτυξης υλικού (2)

- Ξαναγράφει το μετατροπέα διανύσματος/ πλέγματος εικονοστοιχείων από τη γλώσσα assembly στη C στη θύρα στο 1960.
- Η διεπαφή αναπτύσσεται στο PC, εξετάζεται από τους σχεδιαστές της διεπαφής του χρήστη και από το μάρκετινγκ.
- Το έγγραφο φόρτωσης σχεδιάζεται από τους μηχανικούς.

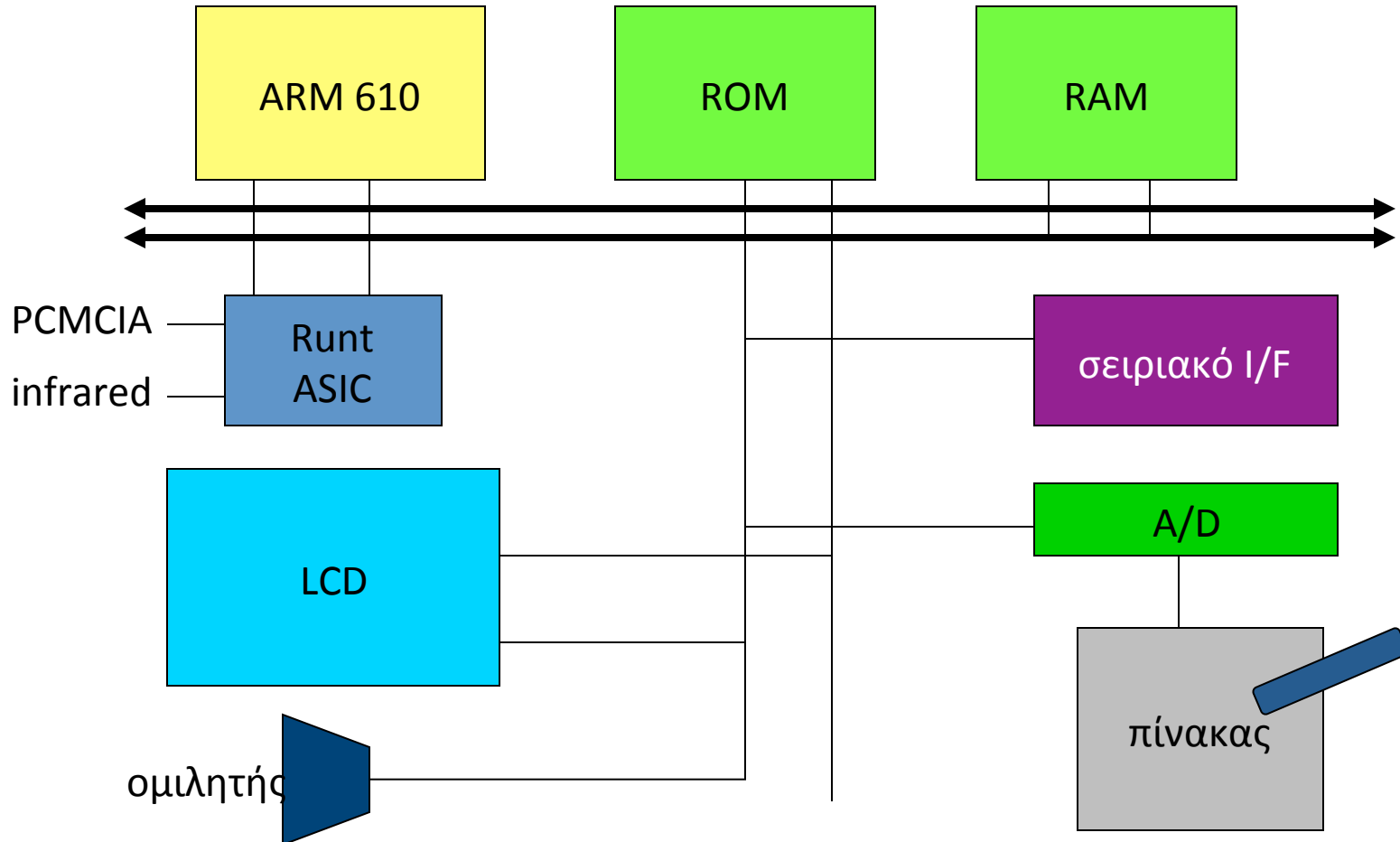
Προσωπικός ψηφιακός βοηθός

- **PDA**: φορητός, εξειδικευμένη συσκευή πληροφοριών.
- Χαρακτηριστικά:
 - χαμηλότερο κόστος για τη λιανική αγορά
 - φυσικά μικρό
 - τροφοδοτούμενο από μπαταρία
 - πλούσιο σε λογισμικό.

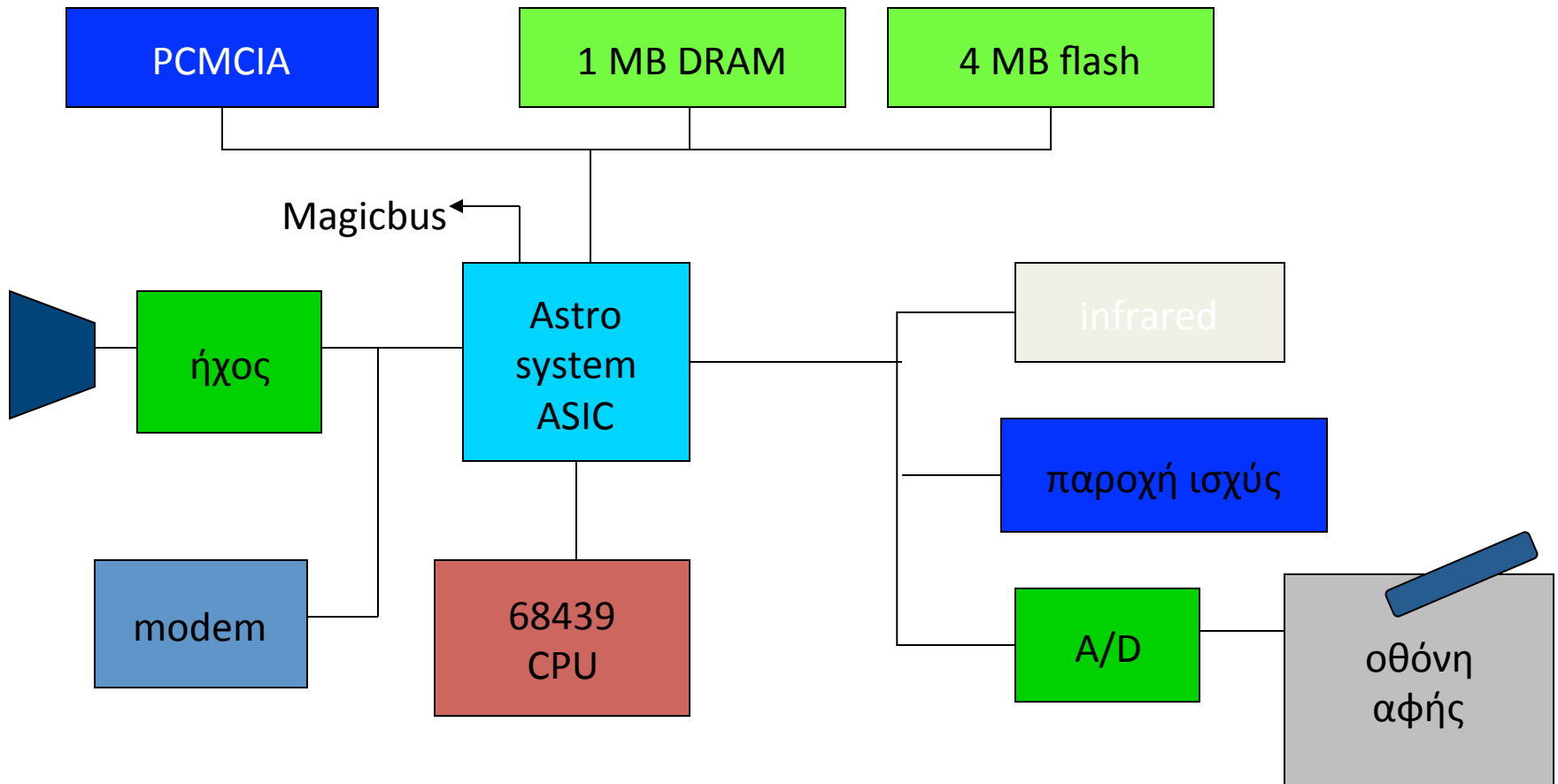
Apple Newton

- Το πρώτο μοντέρνο PDA.
- Αρχικά χρησιμοποιούσε τον ARM 610, πιο πρόσφατη έκδοση χρησιμοποιούσε τον StrongARM.
- Υποστηρίζει διεργασίες σε Runt ASIC: DMA, ρολόι σε πραγματικό χρόνο, διεπαφή video, ήχο, PCMCIA.
- Το λογισμικό είναι γραμμένο σε γλώσσα NewtonScript.

Αρχιτεκτονική υλικού Newton



Αρχιτεκτονική υλικού Motorola Envoy



Αλλαγές στη σχεδίαση ...

- Οι σχεδιαστές τείνουν να προσθέτουν χαρακτηριστικά γνωρίσματα στο σύστημα κατά τη διάρκεια της σχεδίασης.
 - Αυξάνει την κατανάλωση ισχύος.
 - Αλλάζει τη μηχανική σχεδίαση.
 - Κάνει τη σχεδίαση του λογισμικού πιο σύνθετη.

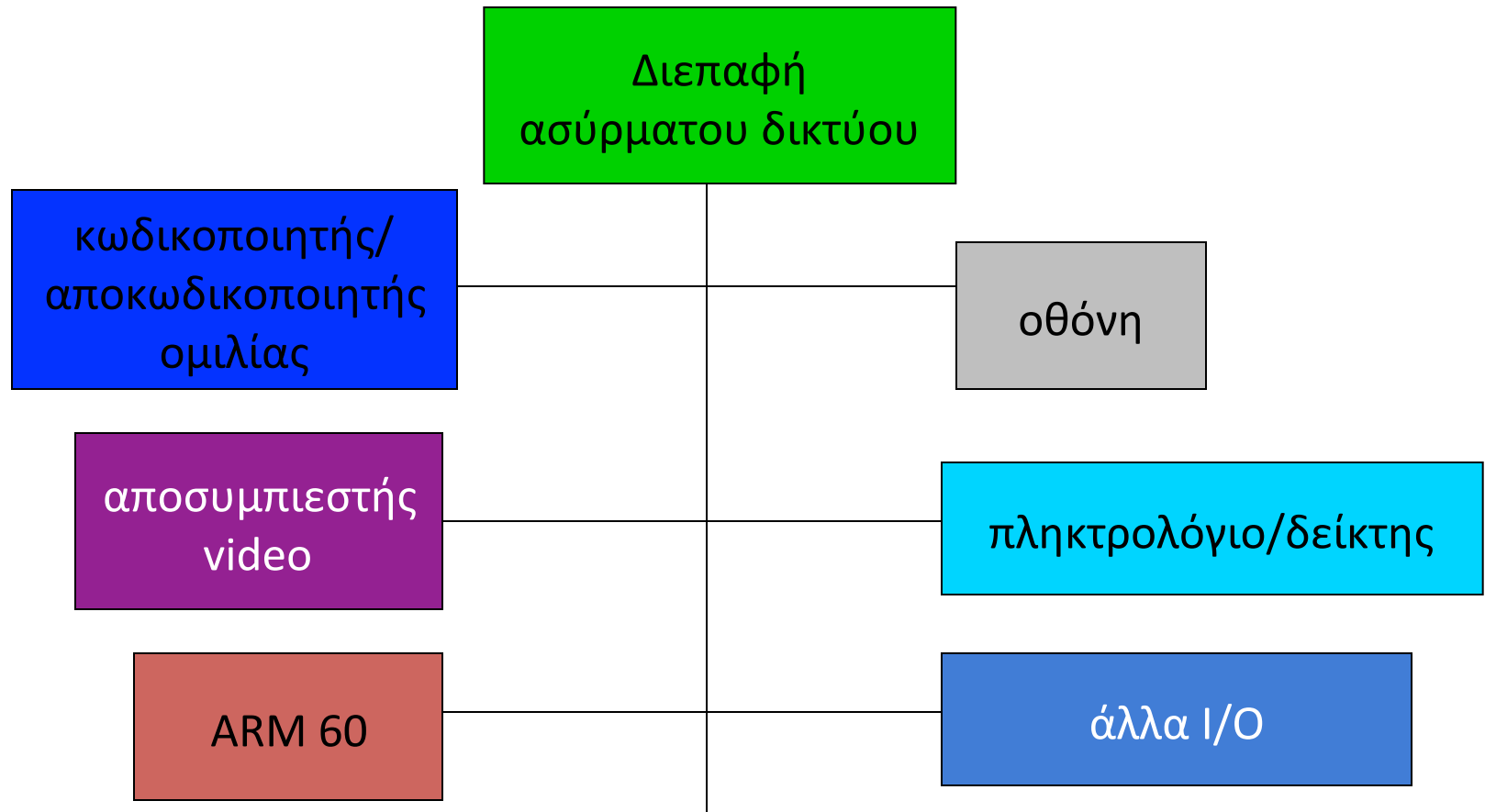
Παροχή ισχύος του PDA

- Το σύστημα πρέπει να σχεδιαστεί για να χειριστεί τη χαμηλή ισχύ της μπαταρίας.
 - Η απότομη απώλεια ισχύος μπορεί να καταστρέψει πολλά δεδομένα στη RAM.
- Το Έξυπνο Σύστημα Μπαταρίας βάζει ηλεκτρονικά συστήματα στην μπαταρία για να καταμετρήσει την απόδοσή της.

InfoPad

- Brodersen et al: Προηγμένη δικτυωμένη συσκευή πληροφοριών πολυμέσων.
- Το σύστημα εκτελεί πολλές λειτουργίες στα απομακρυσμένα συστήματα για να αυξήσει τη διάρκεια ζωής της μπαταρίας.
- Χρησιμοποιεί εξειδικευμένες μονάδες υλικού για να μειώσει την κατανάλωση ισχύος πέρα από την εφαρμογή λογισμικού.

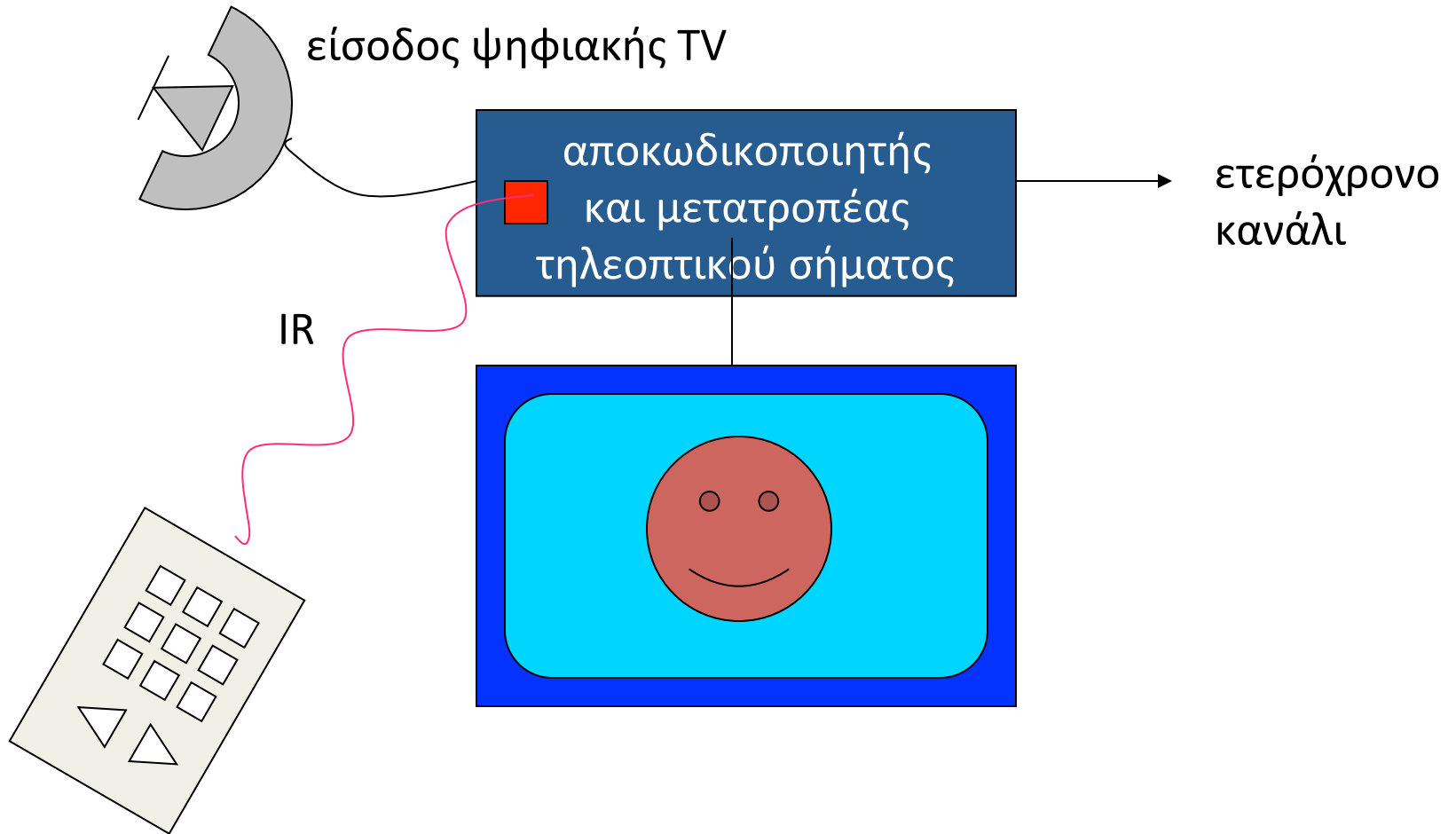
Αρχιτεκτονική υλικού InfoPad



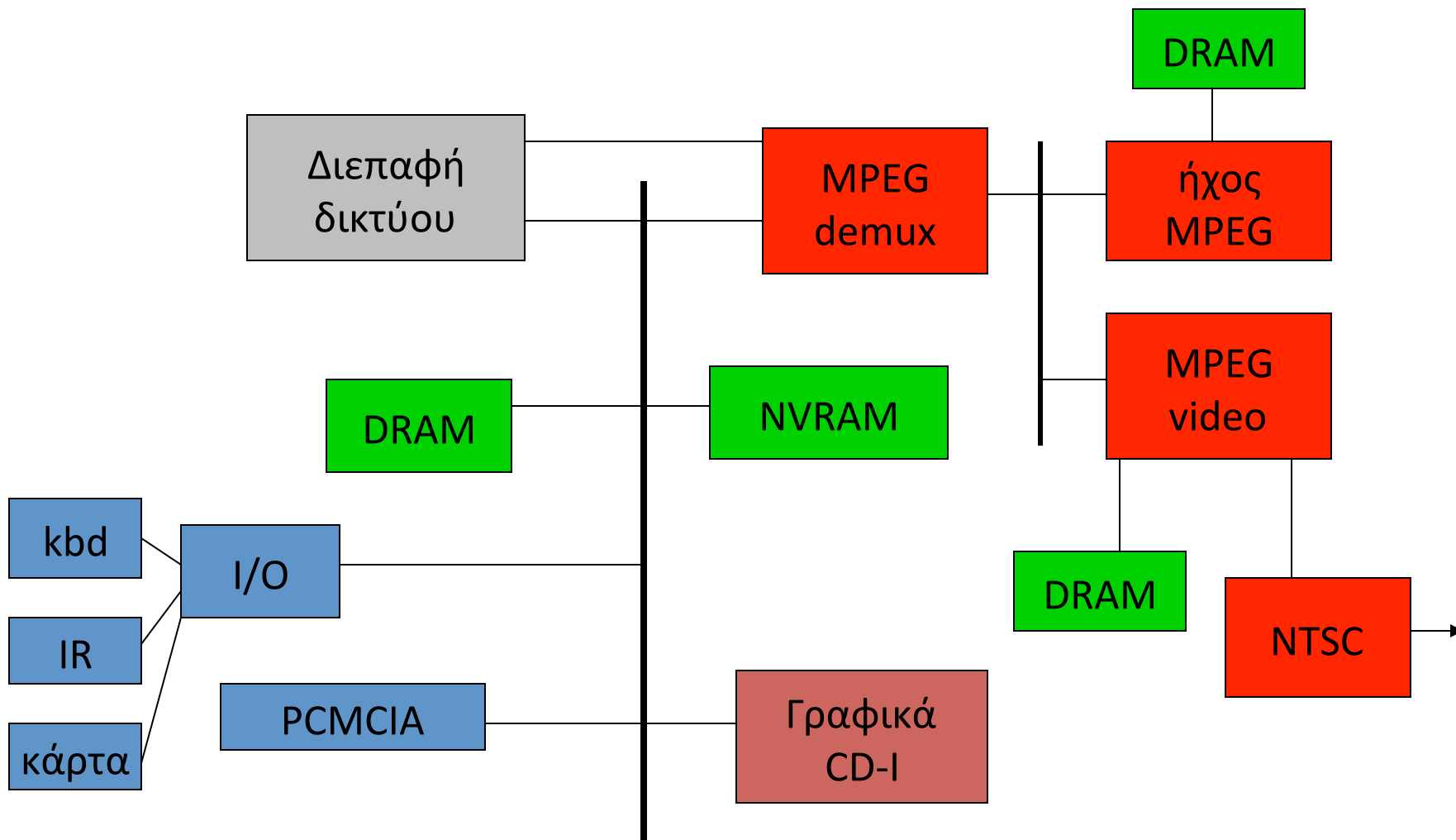
Αποκωδικοποιητές και μετατροπείς τηλεοπτικού σήματος

- Διεπαφή μεταξύ καλωδιακής/ δορυφόρου και της τηλεόρασης:
 - ψηφιακή τηλεόραση
 - διεπαφή χρήστη
 - μπορεί να περιλαμβάνει ετερόχρονο κανάλι για συναλλαγές, κτλ.
- Πολύ οικονομικά ευαίσθητη αγορά.

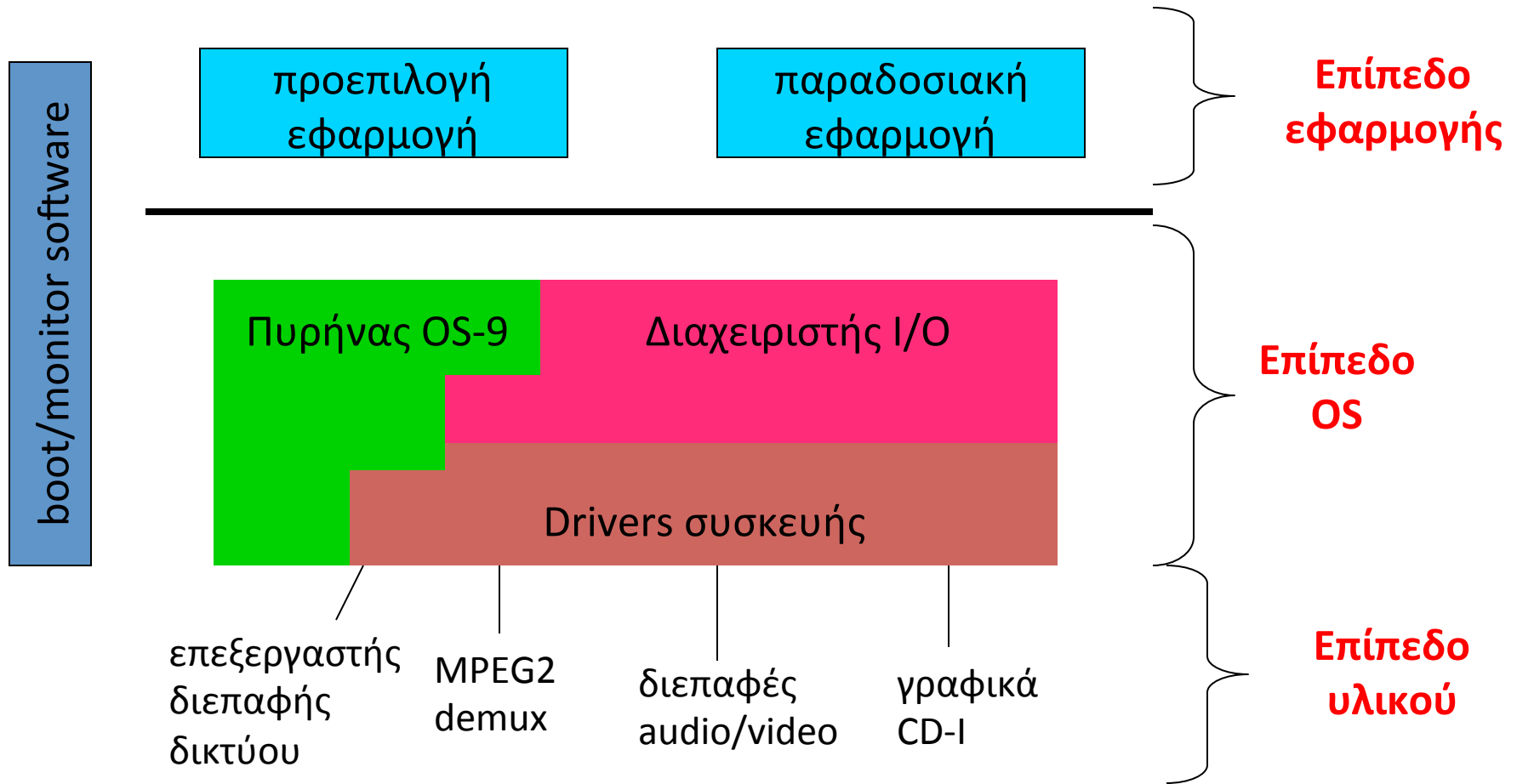
Αποκωδικοποιητής και μετατροπέας τηλεοπτικού σήματος σε σύστημα



Υλικό της Philips fiber-to-curb box



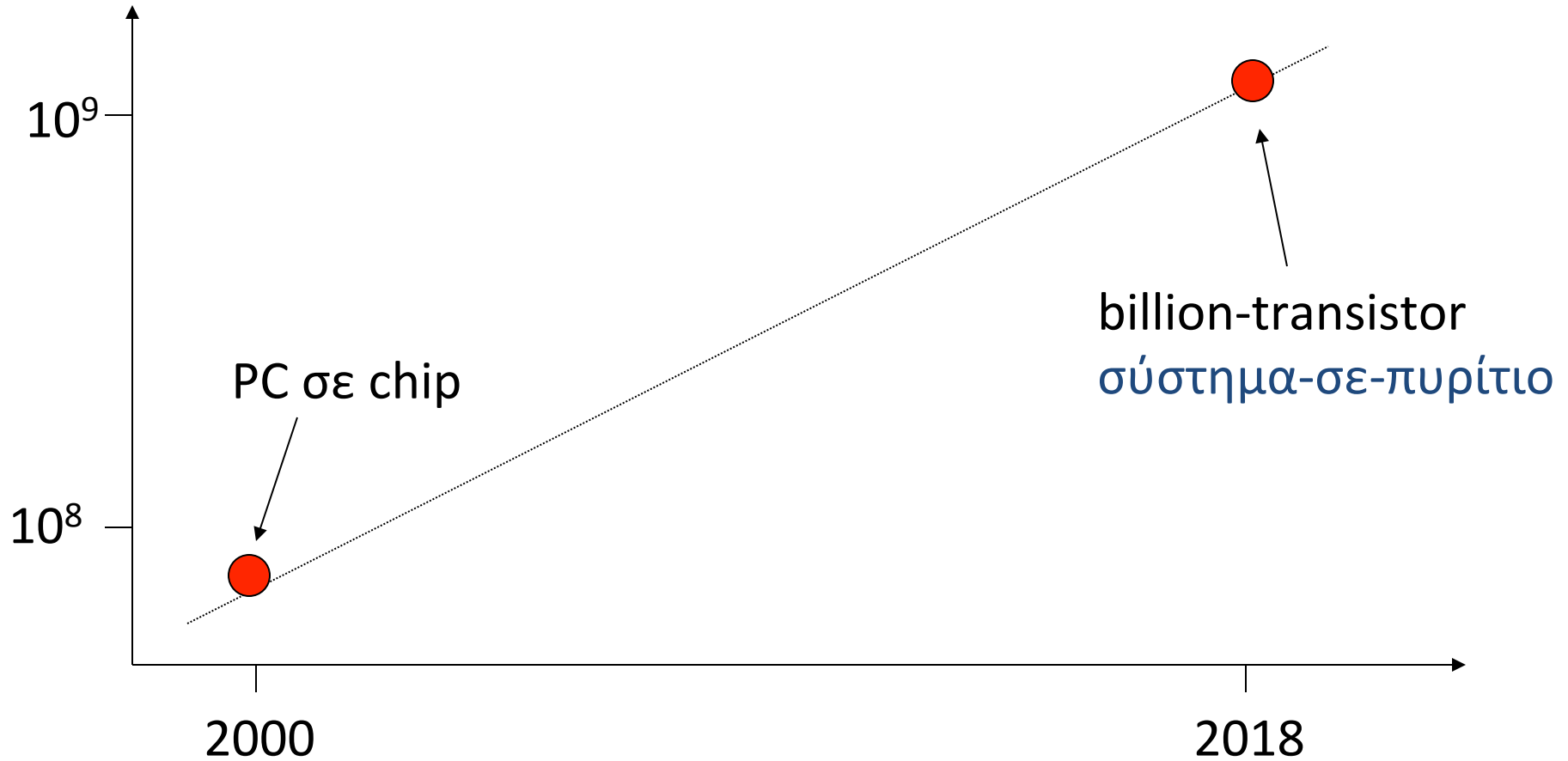
Λογισμικό του Fiber-to-curb box



Σύστημα-σε-πυρίτιο

- Μπορεί να κατασκευάσει σημαντικά ενσωματωμένα συστήματα σε απλό chip:
 - μία ή περισσότερες CPU υψηλής απόδοσης
 - συσκευές I/O
 - μνήμη
- Πλεονεκτήματα:
 - υψηλότερη απόδοση και χαμηλότερη ισχύς
 - χαμηλότερο κόστος

Ο νόμος του Moore



Προκλήσεις σχεδίασης

- Σχεδίαση βασισμένη σε πυρίτιο: μπορεί να πάρει περισσότερο χρόνο για να σχεδιαστεί το σύστημα από το να υλοποιηθεί σε λογισμικό.
- Παραδοσιακές αρχιτεκτονικές:
 - ετερογενείς, πολυεπεξεργαστές
 - παραδοσιακά συστήματα μνήμης.
- Πιστοποίηση:
 - μεγάλος χρόνος ανάδρασης
 - Δεν μπορεί να εξετάσει άμεσα το εσωτερικό του συστήματος.

Σχεδιασμός Συστημάτων

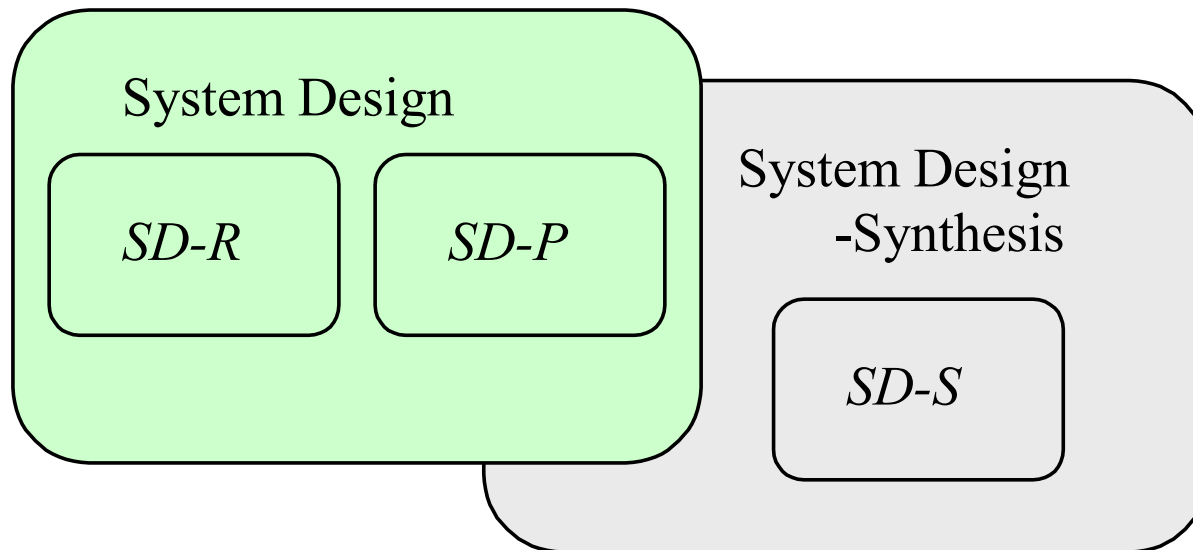
Μια μη τυπική προσέγγιση

Στόχος του σχεδιασμού συστημάτων

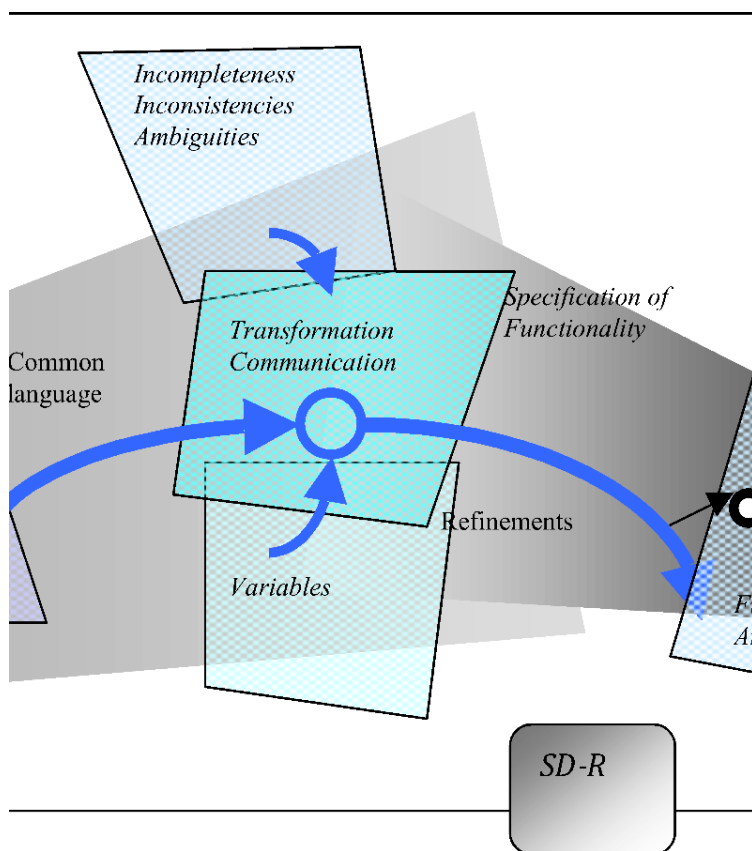
- Η ιδέα ενός νέου συστήματος γεννάται βάσει κάποιων αναγκών και ερεθισμάτων
 - Ευκαιρίες στην αγορά
 - Εμπνεύσεις
 - Ικανοποίηση υπαρχουσών αναγκών
 - Τεχνολογική πρόοδος
- Όλα τα παραπάνω συνθέτουν τις αρχικές προδιαγραφές του συστήματος ως σύνολο απαιτήσεων του χρήστη και της περιοχής ενδιαφέροντος (**user and domain requirements**)
- Το επόμενο στάδιο είναι η διαδοχική εκλέπτυνση των απαιτήσεων και η μετατροπή τους σε τεχνικές προδιαγραφές (**technical requirements specification**)

Στόχος του σχεδιασμού συστημάτων (2)

- *Εκλέπτυνση* του σχεδιασμού συστήματος (**SD-R** : System Design Refinement)
- *Επιμερισμός* του σχεδιασμού συστήματος (**SD-P**: System Design Partitioning)



Εκλέπτυνση του συστήματος (SD-R)

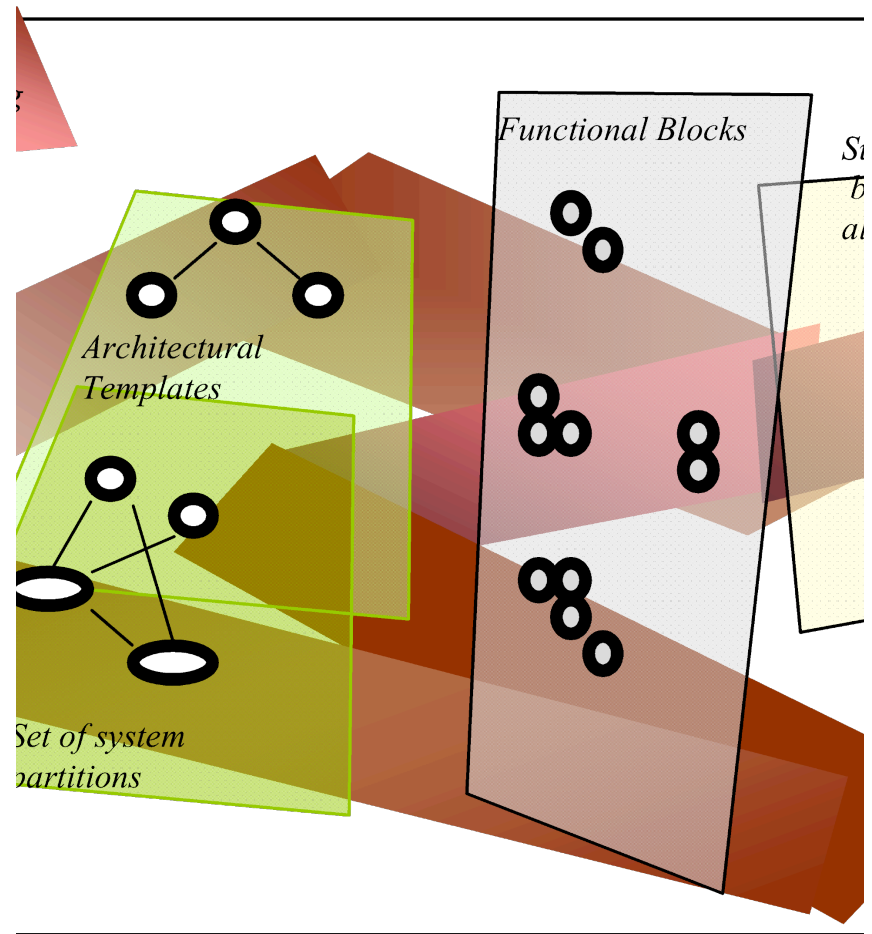


Η εκλέπτυνση του συστήματος (SD-R), συνήθως γίνεται βάσει κάποιων αρχικών ιδεών:

- Τις σημασιολογικές έννοιες που είναι ήδη γνωστές στους σχεδιαστές
- Η εμπειρία από το σχεδιασμό προηγούμενων συστημάτων
- Η εκλέπτυνση συνήθως βασίζεται σε όρους που καθορίζουν το ρόλο επιμέρους τμημάτων του συστήματος:
 - fixed vs programmable,
 - hardware vs software
 - control vs processing
- Το αποτέλεσμα της σχεδιαστικής φάσης SD-R είναι μια λειτουργική αναπαράσταση του συστήματος

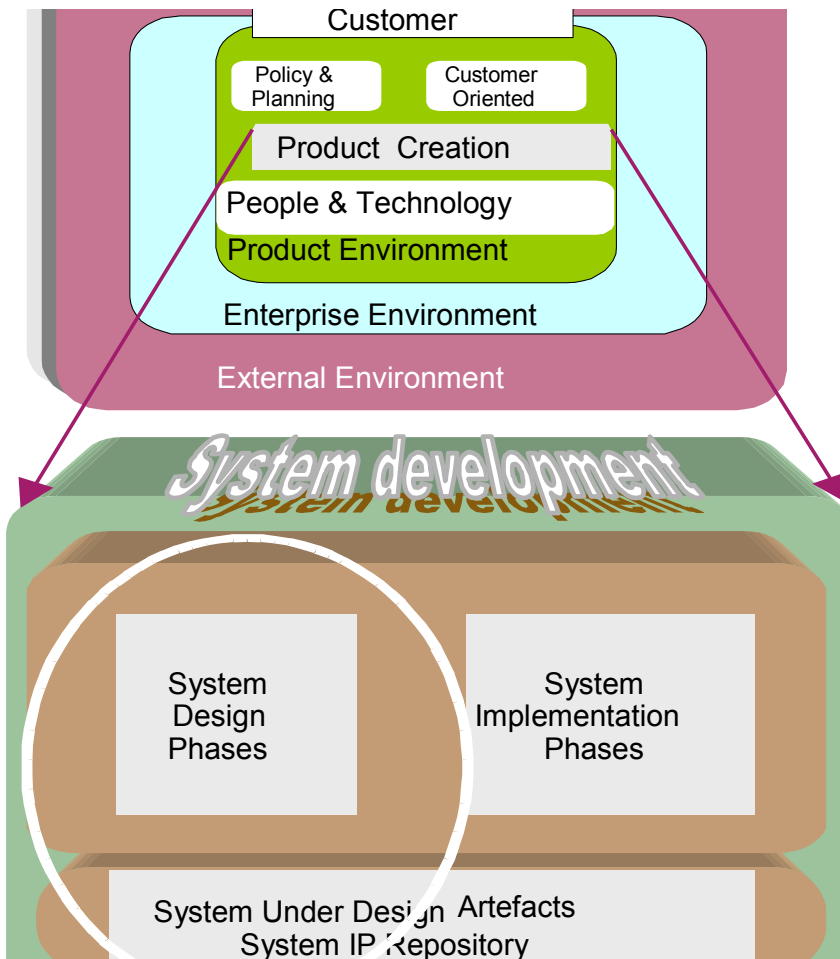
Επιμερισμός συστήματος (SD-P)

- Οι προδιαγραφές αντιστοιχίζονται σε αρχιτεκτονικές
- Οι λειτουργίες της αρχιτεκτονικής για υπολογισμούς και επικοινωνία μπορούν να δημιουργηθούν χρησιμοποιώντας είτε αντικείμενα είτε λειτουργικά μπλοκ
- Αυτά μπορούν να αντιστοιχιστούν είτε σε υλικό είτε σε λογισμικό, οδηγώντας σε ένα σύνολο περιγραφών για την αρχιτεκτονική του συστήματος



Εννοιολογικό μοντέλο σχεδιασμού συστημάτων

Σύνοψη του SDCM



- Το εννοιολογικό μοντέλο σχεδιασμού συστημάτων (**SDCM System Design Conceptual Model**) παρέχει μια ενιαία, συνολική θεώρηση του σχεδιασμού συστημάτων
- Συστήματα IP (System IP) που σχετίζονται με τη διαδικασία σχεδίασης μπορεί να είναι:
 - Τμήματα της διαδικασίας ανάπτυξης
 - Εμπειρία σε μεθόδους (ανάλυση, σύνθεση, κλπ) που εμπεριέχονται σε κανόνες και οδηγίες για τη σχεδίαση του συστήματος (πιθανότατα υλοποιημένα από εργαλεία και μοτίβα σχεδίασης)
 - Εμπειρία σε στυλ σχεδίασης (μοντελοποίηση, επιβεβαίωση, κλπ) που εμπεριέχεται σε ελεγκτές (checkers)
 - Εμπειρία σε εργαλεία που εμπεριέχεται σε scripts που έχουν αναπτυχθεί

IP Συστημάτων που σχετίζονται με τη διαδικασία σχεδίασης

- Αλγοριθμική γνώση
- Υποσυστήματα εφαρμογών
- Αρχιτεκτονικές (υλικού και/ή λογισμικού) συστημάτων
- Υποσυστήματα που ανήκουν με μια συγκεκριμένη, παγιωμένη (για την εταιρία) περιοχή ενδιαφέροντος, για τα οποία η πιθανότητα επαναχρησιμοποίησης είναι υψηλή
- IPs που έχουν αναπτυχθεί και συντηρούνται από τρίτους κατασκευαστές για λογαριασμό της εταιρίας (outsourcing)

Δομές του Εννοιολογικού Μοντέλου

- Το εννοιολογικό μοντέλο «συλλαμβάνει» το νόημα μιας περιοχής εφαρμογής όπως αυτή γίνεται αντιληπτή από κάποιον, δηλαδή περιγράφει τη γνώση για μια περιοχή ενδιαφέροντος του πραγματικού κόσμου
- Ο πραγματικός κόσμος γίνεται αντιληπτός μέσω:
 - **πραγμάτων (things)**, που συχνά αναφέρονται ως **οντότητες (entities)**
 - **συσχετίσεων (associations)**, που συχνά αναφέρονται ως **σχέσεις (relationships)**
- Ο κόσμος αποτελείται από πράγματα που έχουν ιδιότητες (**properties**) – οι ιδιότητες υπάρχουν είτε τις γνωρίζουμε είτε όχι
- Ωστόσο, αντιλαμβανόμαστε τα πράγματα ως **μοντέλα πραγμάτων (models of things)**
- Αυτά τα μοντέλα, ονομάζονται **εννοιολογικά πράγματα (conceptual things)**
 - Οι ιδιότητες ενός εννοιολογικού πράγματος ονομάζονται **ιδιότητες (attributes)**

Οντολογία

- Μια οντολογία είναι μια περιγραφή **εννοιών (concepts)** και **σχέσεων (relations)** που υπάρχουν σε μια συγκεκριμένη περιοχή
- Το πλεονέκτημα μιας οντολογίας είναι ότι εξαλείφει προβλήματα που σχετίζονται με τα λεξιλόγια των φυσικών γλωσσών
- Υπάρχουν τρία είδη πληροφορίας (ή επίπεδα) σε μια οντολογία:
 - **Το επίπεδο της ορολογίας (terminology level):** Είναι το βασικό σύνολο εννοιών και σχέσεων που αποτελούν μια οντολογίας
 - ✓ Ονομάζεται επίσης και **επίπεδο ορισμού (definition layer)** της οντολογίας
 - **Το επίπεδο ισχυρισμών (assertion level):** Είναι ένα σύνολο ισχυρισμών που αφορούν στις βασικές έννοιες και σχέσεις της οντολογίας
 - ✓ Ονομάζεται επίσης και **επίπεδο αξιωμάτων (axioms layer)** της οντολογίας
 - **Πραγματικό επίπεδο (pragmatic level):** Περιέχει πραγματικές πληροφορίες που δεν μπορούν να ενταχθούν σε κάποιο από το προηγούμενα επίπεδα
 - ✓ Ονομάζεται επίσης και **επίπεδο εργαλειοθήκης (toolbox layer)** της οντολογίας

Οντολογία (2)

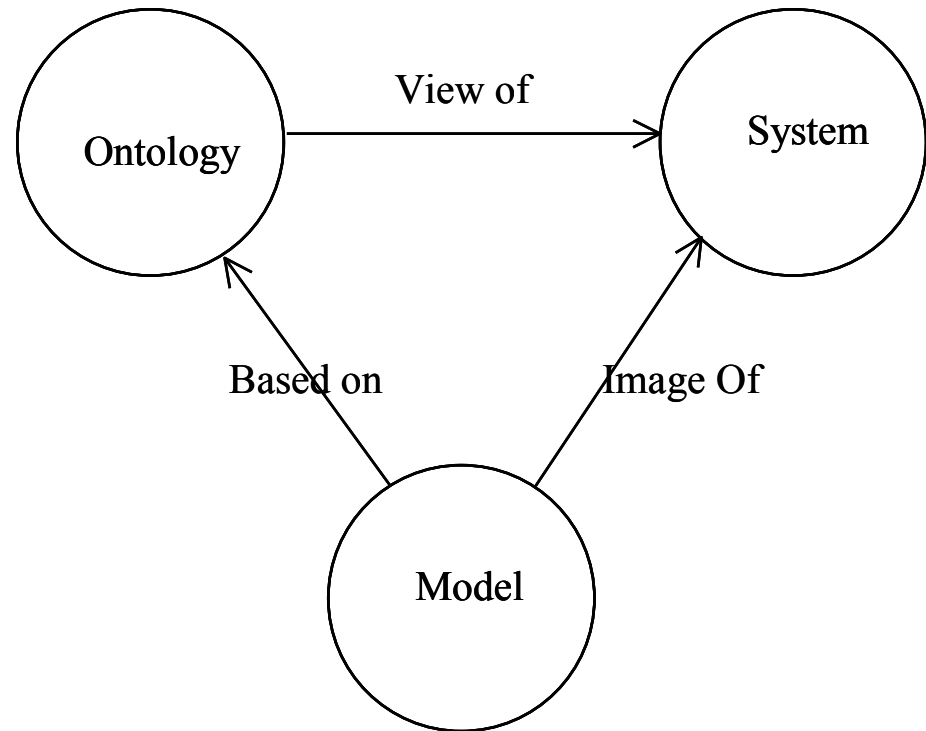
- Οι βασικές ιδιότητες μιας οντολογίας είναι:
 - **Διαμοιρασμός (sharing)**: Συμφωνία η οποία βασίζεται στην αποδοχή μια οντολογίας αναφορικά με την κοινή κατανόηση μιας δεδομένης ιδέας
 - **Φιλτράρισμα (filtering)**: Μια συνδεδεμένη αφαίρεση μοντέλων που λαμβάνει υπόψη μόνο ένα μέρος της πραγματικότητας
- Η οντολογία ορίζει τι πρέπει να εξαχθεί από ένα σύστημα ώστε να μπορέσουμε να δημιουργήσουμε ένα μοντέλο του συστήματος

Οντολογία (3)

- Μία οντολογία είναι μια περιγραφή **εννοιών** και **σχέσεων** που υπάρχουν για έναν συγκεκριμένο τομέα
- Η βασική χρήση μιας οντολογίας είναι ότι διευκολύνει τον διαχωρισμό εννοιών
- Για ένα συγκεκριμένο σύστημα, μπορούμε να παρατηρήσουμε και να εργαστούμε με **διαφορετικά μοντέλα του ίδιου συστήματος**, κάθε ένα από τα οποία χαρακτηρίζεται από μια συγκεκριμένη **οντολογία**

Οντολογία (4)

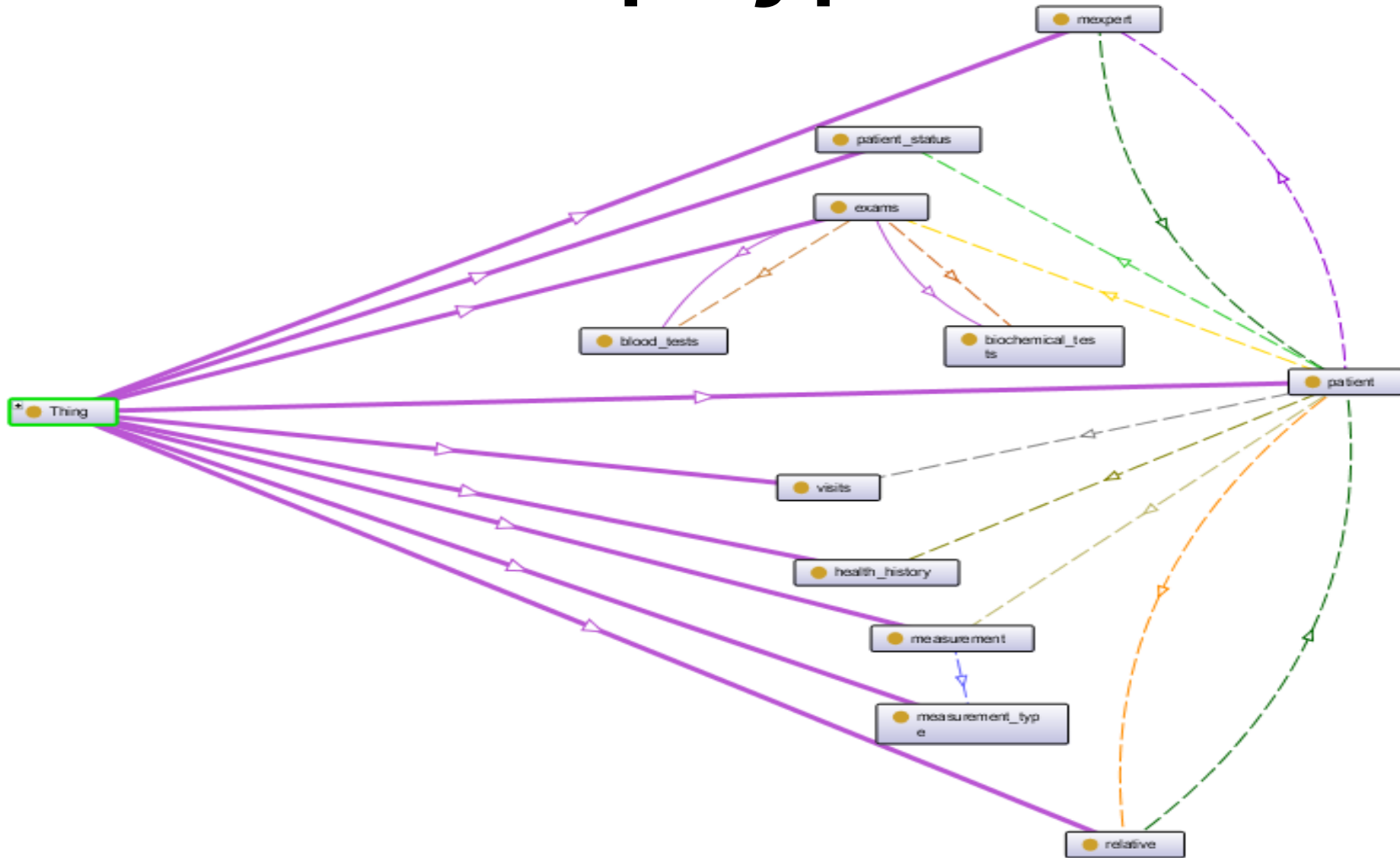
- Ένα **μοντέλο** δημιουργείται για ένα συγκεκριμένο **σκοπό**, συνήθως για να κατανοήσουμε συγκεκριμένες έννοιες του υπό ανάπτυξη συστήματος
- Ο σκοπός πρέπει να ορίζεται ξεκάθαρα και να σχετίζεται με την οντολογία
- Μια οντολογία θεωρεί τα εξής συστατικά {**σύστημα, οντολογία, μοντέλο**}



Γλώσσες Περιγραφής Οντολογιών

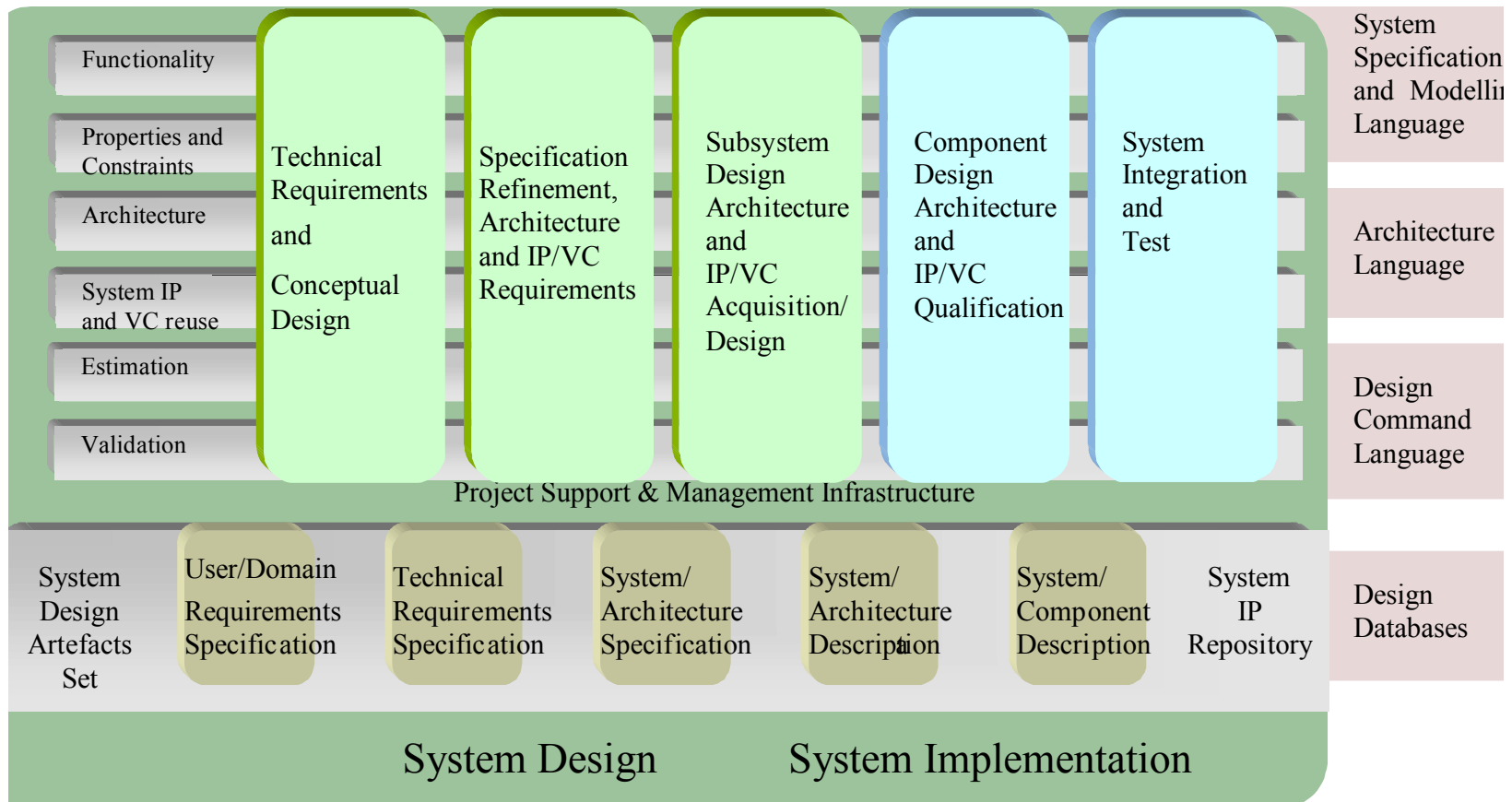
- Υπάρχουν διάφορες γλώσσες οντολογιών για τη σχεδίαση των αντίστοιχων μοντέλων όπως η
 - RDFS (DAML site),
 - η DAMN+OIL (DAML site) και η
 - OWL (W3C, 2005) η οποία εξάγεται από τις προηγούμενες δύο.
- Η OWL έχει σημαντικό πλεονέκτημα έναντι των υπολοίπων γλωσσών λόγω της αυξημένης της εκφραστικότητας.
- Η OWL είναι στενά συνδεδεμένη με το Πλαίσιο Περιγραφής Πόρων (Resource Description Framework – RDF), το οποίο και χρησιμοποιείται από την κοινοπραξία του παγκόσμιου ιστού (World Wide Web Consortium) ως βάση του σημασιολογικού ιστού. Η OWL χρησιμοποιεί τη σύνταξη της RDF που στηρίζεται στην XML (RDF/XML).

Παράδειγμα περιγραφής οντολογίας με OWL



Μοντέλο διαδικασίας σχεδιασμού συστήματος

System design process model (SDPM)



Μοντέλο διαδικασίας σχεδιασμού συστήματος (2)

Οι γενικές οπτικές γωνίες του σχεδιασμού συστημάτων και τα αντίστοιχα «δημιουργήματά» τους που αφορούν στο SDPM είναι:

- Οι **τεχνικές προδιαγραφές** και ο **εννοιολογικός σχεδιασμός**, ως αποτέλεσμα της ανάλυσης απαιτήσεων
- Η **εκλέπτυνση των προδιαγραφών**, της **αρχιτεκτονικής** και των **απαιτήσεων για IP/VC (Intellectual Property/Virtual Component)**, που καταλήγουν στον προσδιορισμό των **προδιαγραφών της αρχιτεκτονικής του συστήματος**
- Ο **σχεδιασμός υποσυστημάτων**, της **αρχιτεκτονικής** και της **απόκτησης και/ή σχεδιασμού IP/VC** που οδηγούν στην περιγραφή της αρχιτεκτονικής του υπό ανάπτυξη συστήματος (*SUD* - System Under Design)

Μοντέλο διαδικασίας σχεδιασμού συστήματος (3)

Μέσω το SDPM, έχουμε τις ακόλουθες θεωρήσεις του υπό ανάπτυξη συστήματος (SUD – System Under Design)

- **Λειτουργικότητα**, η οποία εκλεπτύνεται και συσχετίζεται με τη συμπεριφορά του συστήματος
- **Ιδιότητες και περιορισμοί**, που είτε παραμένουν αμετάβλητες κατά η σχεδίαση είτε εκλεπτύνονται και συσχετίζονται με το σύστημα και το περιβάλλον του
- **Αρχιτεκτονική**, η οποία συμπεριλαμβάνει διαφορετικά είδη αρχιτεκτονικών που χρησιμοποιούνται κατά τη διάρκεια της εκλέπτυνσης
- **System IP** και επαναχρησιμοποίηση **VC**, που αφορά στη χρήση υπάρχουσας γνώσης ή στην επαναχρησιμοποίηση συστατικών κατά την εκλέπτυνση και το σχεδιασμό του SUD
- **Εκτίμηση**, είναι μια τεχνική που αποσκοπεί στο να παρέχει εκτιμήσεις για τις συνέπειες των σχεδιαστικών αποφάσεων, και αφορά όλες τις φάσεις σχεδίασης του SUD
- **Επικύρωση και επιβεβαίωση**, είναι τεχνικές που αποσκοπούν στο να αποδείξουν ότι το σύστημα που σχεδιάζεται είναι το σωστό σύστημα και ότι παράγει το σωστό αποτέλεσμα (αφορούν επίσης όλες τις φάσεις σχεδίασης του SUD)

Επικύρωση (Validation): Are we building the **right system**?

Επιβεβαίωση (Verification): Are we building the **system right**?

Βασικές αρχές γλωσσών Ανάλυσης Απαιτήσεων & Σχεδιασμού Συστημάτων

Εισαγωγή στις γλωσσών Ανάλυσης Απαιτήσεων & Σχεδιασμού Συστημάτων

- Το **ενοιολογικό μοντέλο σχεδιασμού συστημάτων (SDCM)** αναφέρεται στα αρχικά στάδια σχεδίασης του συστήματος
- ... αλλά δεν αναφέρει συγκεκριμένες τεχνολογίες
- Θεωρεί το **υπό σχεδίαση σύστημα (SUD - System Under Design)** και τη **διαδικασία σχεδίασης του συστήματος (SDP System Design Process)** ως «μετά-μοντέλα» (meta-models)
- Οι **προδιαγραφές ενός συστήματος** και η **γλώσσα σχεδίασης** περιγράφουν το SUD σε όλα τα απαιτούμενα επίπεδα αφαίρεσης και αποσκοπούν:
 - στο μετασχηματισμό,
 - στην επικύρωση και
 - στην ανάλυση των εργασιώνενός συγκεκριμένου επίπεδου αφαίρεσης βάσει της διαδικασίας σχεδίασης που εφαρμόζεται

Βασικές έννοιες συστημάτων

- **Παραλληλισμός και ταυτοχρονισμός (Parallelism & concurrency)**
αναφέρονται στον διαμοιρασμό λειτουργιών σε διαφορετικούς πόρους
 - Ο ταυτοχρονισμός είναι ένα τρόπος να υλοποιήσουμε τον παραλληλισμό και επιτυγχάνεται με την επικάλυψη ή την ταυτόχρονη εκτέλεση δύο ή περισσότερων νημάτων (threads)
- Ο παραλληλισμός μπορεί να εκφραστεί είτε με **ροές ελέγχου (control flow)** είτε με **ροές δεδομένων (data flow)**
- **Ροές ελέγχου:** αφορούν μοντέλα του συστήματος στα οποία καθορίζεται η ακολουθία εκτέλεσης των τμημάτων του συστήματος. Τα
 - CSPs (Communicating Sequential Processes) και
 - FSMs (Finite State Machines)Είναι κλασικές προσεγγίσεις έκφρασης παραλληλισμού με βάση τις ροές ελέγχου
- **Ροές δεδομένων:** Η ροή εκτέλεσης των εντολών καθορίζεται από τις εξαρτήσεις των δεδομένων, οι οποίες εκφράζονται μέσω γράφων ροών δεδομένων (data flow graphs)

Βασικές έννοιες συστημάτων (2)

- **Ιεραρχική ανάπτυξη:** Επιτρέπει τον ιεραρχικό σχεδιασμό σύνθετων συστημάτων σύμφωνα με την οποία οι σχεδιαστές επιμερίζουν τη λειτουργικότητα σε υποσυστήματα, τα οποία είναι πιο εύκολο να σχεδιαστούν. Υπάρχουν δυο κατηγορίες ιεραρχικής ανάπτυξης:
 - η ιεραρχία συμπεριφοράς (*behavioural hierarchy*) και
 - η δομική ιεραρχία (*structural hierarchy*)
- **Επικοινωνία:** επιτρέπει στα υποσυστήματα να ανταλλάσσουν δεδομένα και πληροφορίες ελέγχου. Υπάρχουν δύο βασικά μοντέλα:
 - το πέρασμα μηνυμάτων (*message passing*) και
 - Η διαμοιραζόμενη μνήμη (*shared memory*)
- **Συγχρονισμός:** ορίζει τις αρχές της επικοινωνία ανάμεσα στα διάφορα υποσυστήματα. Υπάρχουν δύο τεχνικές:
 - οι ουρές μηνυμάτων (*message queues*) και
 - το ραντεβού (*rendezvous*)

Ταξινόμηση γλωσσών σχεδίασης με βάση το υπολογιστικό τους μοντέλο

- Σύμφωνα με τον D. Gajski υπάρχουν 5 βασικές κατηγορίες γλωσσών:
 - A) Προσανατολισμένες σε καταστάσεις (state oriented)*
 - B) Προσανατολισμένες σε ενέργειες (activity oriented)*
 - Γ) Προσανατολισμένες στη δομή (structure oriented)*
 - Δ) Προσανατολισμένες στα δεδομένα (data-oriented), και*
 - E) Ετερογενείς (heterogeneous)*
- Οι γλώσσες που ανήκουν στις κατηγορίες A & B επιτρέπουν την περιγραφή του συστήματος μέσω μηχανών καταστάσεων και μετασχηματισμών
- Οι γλώσσες της κατηγορίας της κατηγορίας Γ δίνουν έμφαση στη δομή του συστήματος
- Οι γλώσσες της κατηγορίας της κατηγορίας Δ δίνουν έμφαση στην περιγραφή των διαδικασιών του συστήματος που επεξεργάζονται την πληροφορία

Ταξινόμηση γλωσσών σχεδίασης με βάση το υπολογιστικό τους μοντέλο (2)

- Μια αντικειμενική ταξινόμηση των γλωσσών σχεδίασης είναι αυτή βάσει του υπολογιστικού του μοντέλου που υποστηρίζουν
- Η περιγραφή του συστήματος αντανακλά τη συντακτική δομή της γλώσσας που χρησιμοποιείται και όχι το υπολογιστικό μοντέλο της γλώσσας
- Το **υπολογιστικό μοντέλο** σχετίζεται με το θεωρητικό υπόβαθρο στο οποίο βασίζεται το μοντέλο εκτέλεσης της γλώσσας
- Το υπολογιστικό μοντέλο μπορεί να θεωρηθεί ως ορθογώνιος συνδυασμός του **μοντέλου επικοινωνίας (communication model)** και του **μοντέλου ελέγχου (control model)**

Ταξινόμηση γλωσσών σχεδίασης με βάση το υπολογιστικό τους μοντέλο (3)

- Μια γλώσσα μπορεί να υποστηρίζει τα ακόλουθα μοντέλα επικοινωνίας:
 - *σύγχρονο (ή απλού νήματος) και*
 - *κατανεμημένο όπου το μοντέλο επικοινωνίας μεταξύ των νημάτων θα πρέπει να είναι σαφώς ορισμένο*
- Τα μοντέλα ελέγχου που μπορεί να υποστηρίζει μια γλώσσας είναι:
 - *προσανατολισμένα σε ροές ελέγχου (control flow oriented) ή*
 - *προσανατολισμένα σε ροές δεδομένων (data flow oriented)*

Ταξινόμηση γλωσσών σχεδίασης με βάση το υπολογιστικό τους μοντέλο (4)

- Τα περισσότερα εργαλεία συ-σχεδίασης χρησιμοποιούν επίσης εσωτερικές γλώσσες αναπαράστασης που διευκολύνουν την εκλέπτυνση των μοντέλων
- Τυπικές γλώσσες για την εσωτερική αναπαράσταση των μοντέλων είναι οι:
 - SDL (Specification and Description Language)
 - C
 - VHDL
 - JAVA
- Υπάρχουν δύο βασικές κατηγορίες ενδιάμεσων αναπαραστάσεων
 - προσανατολισμένες σε γλώσσες (*language oriented*) και
 - προσανατολισμένες σε αρχιτεκτονική (*architectural oriented*)

Ταξινόμηση γλωσσών σχεδίασης με βάση το υπολογιστικό τους μοντέλο (5)

- Οι αναπαραστάσεις προσανατολισμένες σε γλώσσες βασίζονται σε:
 - Γράφους ροής δεδομένων (Data flow graphs – DFG)
 - Γράφους ροής ελέγχου (Control Flow Graphs – CFG)
- Οι αναπαραστάσεις προσανατολισμένες σε αρχιτεκτονική βασίζονται σε:
 - Μηχανές Πεπερασμένων Καταστάσεων (Finite State Machines – FSMs)
- Οι αναπαραστάσεις προσανατολισμένες σε αρχιτεκτονική αναφέρονται κυρίως στην αρχιτεκτονική του συστήματος και όχι στην αρχική περιγραφή του:
 - FSM για δεδομένα (FSM for Data - FSMD) ή
 - FSM με συνεπεξεργαστές (FSM with Coprocessors - FSMC)

Ταξινόμηση γλωσσών σχεδίασης

- Οι γλώσσες σχεδίασης προέρχονται από την περιοχή του σχεδιασμού λογισμικού
 - Όπου το συνεχώς αυξανόμενο κόστος ανάπτυξης και συντήρηση του λογισμικού οδήγησε το σχεδιαστές στο να δώσουν έμφαση στις *προδιαγραφές και στην ανάλυση απαιτήσεων των συστημάτων*
- Το ίδιο ισχύει και σήμερα στο σχεδιασμό συστημάτων, έτσι ώστε να χειριστούμε τη συνεχώς αυξανόμενη *λειτουργικότητά και πολυπλοκότητά τους*
- Το αποτέλεσμα είναι
 - Τελικά συστήματα υψηλότερης ποιότητας μέσω της διαδικασίας της «**διαδοχικής εκλέπτυνσης (gradual refinement)**»
 - Επιβεβαίωση του συστήματος στα αρχικά στάδια της σχεδίασης

Ταξινόμηση γλωσσών σχεδίασης (2)

Γλώσσες Περιγραφής Αρχιτεκτονικής

- Οι γλώσσες περιγραφής αρχιτεκτονικής (ADLs - Architecture Description Languages) είναι *τυπικές γλώσσες (formal)* που χρησιμοποιούνται για την αναπαράσταση της αρχιτεκτονικής του συστήματος
- Αρχιτεκτονικής σημαίνει:
 - τα συστατικά/υποσυστήματα που απαρτίζουν το υπό ανάπτυξη σύστημα
 - τις προδιαγραφές της συμπεριφοράς τους
 - Τα μοντέλα και τους μηχανισμούς της μεταξύ τους αλληλεπίδρασης
- Μια ADL πρέπει να μοντελοποιεί ρητά τα συστατικά/υποσυστήματα, τις διασυνδέσεις τους και τη διαμόρφωσή τους

Ταξινόμηση γλωσσών σχεδίασης (3)

Γλώσσες Περιγραφής Αρχιτεκτονικής

- Παραδείγματα ADLs:
 - Η PMS (Processor, Memory, Switch από τους Bell & Newell) θεωρείται ως η πρώτη ADL
 - ACME, Aesop, C2, Darwin, MetaH, Rapide, SADL, UniCon, Weaves, and Wright, είναι επίσης μερικές από τις γλώσσες που εντάσσονται στη συγκεκριμένη κατηγορία

Ταξινόμηση γλωσσών σχεδίασης (4)

Γλώσσες Περιγραφής Υλικού

- Τι γλώσσες προδιαγραφών της μηχανικής λογισμικού (software engineering specification languages) ακολούθησαν οι **Γλώσσες Περιγραφής Υλικού (Hardware Specification Languages)**
- Παραδείγματα Γλωσσών Περιγραφής Υλικού:
 - CASSANDRE, DDL and CONLAN of the '60s and '70s
 - HardwareC
 - SpecCharts
 - SpecC
- Και οι πιο δημοφιλείς:
 - VHDL
 - Verilog
- Υπάρχουν επιπλέον γλώσσες αυτής της κατηγορίας που χρησιμοποιούνται σε πιο εξειδικευμένα εργαλεία (όπως DSP) όπως η SPW και η COSSAP.

Ταξινόμηση γλωσσών σχεδίασης (5)

Γλώσσες Σχεδίασης Πρωτοκόλλων

- Υπάρχουν αρκετές γλώσσες που χρησιμοποιούνται για το σχεδιασμό τηλεπικοινωνιακών συστημάτων και πρωτοκόλλων επικοινωνίας
- Οι γλώσσες αυτής της κατηγορίας βασίζονται στις επονομαζόμενες **τυπικές περιγραφές του συστήματος (called formal system description)** και περιλαμβάνουν:
 - **FDT** (Formal Description Technique),
 - **LOTOS** (Logical Temporal Ordering Specification- η οποία είναι OSI στάνταρντ) - Η LOTOS αποτελείται από δύο μέρη: την *περιγραφή της συμπεριφοράς του συστήματος (behavioural part)* η οποία βασίζεται σε *άλγεβρα διαδικασιών (process algebra)* και την *περιγραφή των δεδομένων που γίνεται με αφηρημένους τύπους δεδομένων abstract data types*)

Ταξινόμηση γλωσσών σχεδίασης (6)

- **SDL** (Specification and Description Language): Χρησιμοποιεί Μηχανές Πεπερασμένων Καταστάσεων (FSMs - Finite State Machines) για την περιγραφή των λειτουργιών του συστήματος. Η περιγραφή του συστήματος, εκτός από τη δυναμική συμπεριφορά των επιμέρους τμημάτων του , περιλαμβάνει τον ορισμό του υπό ανάπτυξη συστήματος ως σύνολο καθώς και την επικοινωνία μεταξύ των επιμέρους τμημάτων του.
- **ESTELLE**: είναι μια τυπική διαδικασιακή (procedural) γλώσσα με χαρακτηριστικά όμοια με αυτά της PASCAL. Είναι περισσότερο γλώσσα προγραμματισμού παρά γλώσσα προδιαγραφών.

Ταξινόμηση γλωσσών σχεδίασης (7)

Γλώσσες Προγραμματισμού

- Η πλειοψηφία των γλωσσών προγραμματισμού χρησιμοποιείται τόσο για την περιγραφή όσο και για το σχεδιασμό του συστήματος
- Σε αυτή την κατηγορία ανήκουν η **C**, η **C++** και η **Java**. Το βασικό τους μειονέκτημα είναι το ότι στερούνται της δυνατότητας περιγραφής του **χρόνου** και του **παραλληλισμού**.
- Η **HardwareC** είναι μια επέκταση των παραπάνω γλωσσών, η οποία καλύπτει τις προαναφερθείσες αδυναμίες τους
- Η **SystemC** βασίζεται στη C++ για τις περιγραφές του συστήματος σε όλα τα στάδια σχεδίασής του. Τα βασικά δομικά μπλοκ μιας περιγραφής σε SystemC είναι οι διαδικασίες:
 - ✓ Μια διαδικασία είναι παρόμοια με μία συνάρτηση της a C/C++ που υλοποιεί συμπεριφορά
 - ✓ Μια πλήρης περιγραφή του συστήματος αποτελείται από πολλές παράλληλες διαδικασίες
 - ✓ Λόγω του ότι η SystemC βασίζεται σε επέκταση της C++ για παράλληλη επεξεργασία και περιγραφή υλικού, δεν ανήκει αποκλειστικά σε αυτή την κατηγορία

Ταξινόμηση γλωσσών σχεδίασης (8)

Γλώσσες Αλληλεπίδρασης

- Μια μεγάλη κατηγορία γλωσσών σχεδίασης συστημάτων είναι αυτή που αφορά στις γλώσσες αλληλεπίδρασης (reactive languages), η οποίες περιγράφουν την αλληλεπίδραση του συστήματος με το περιβάλλον του σε πραγματικό χρόνο.
- Παραδείγματα γλωσσών αλληλεπίδρασης:
 - **ESTEREL**: Υποστηρίζει παραλληλισμό και έχει τη δυνατότητα να περιγράψει το σύστημα με τυπικό τρόπο (formal) σε όλα τα σχεδιαστικά στάδια, από τις προδιαγραφές μέχρι την τελική υλοποίηση. Βασίζεται στην έννοια των *σύγχρονων γεγονότων (synchronous event)*
 - **LUSTRE**: είναι μια γλώσσα ανάπτυξης και προγραμματισμού που βασίζεται σε αυτόματα (automata)

Ταξινόμηση γλωσσών σχεδίασης (9)

- **SIGNAL:** Διαφοροποιείται από τη LUSTRE στο ότι χρησιμοποιεί πολλαπλά ρολόγια στο ίδιο πρόγραμμα, κάτι το οποίο μπορεί να συνδυαστεί μέσω προσωρινών τελεστών (temporal operators)
- **Statecharts:** Είναι ένας γραφικός φορμαλισμός που χρησιμοποιείται για την περιγραφή αλληλεπιδραστικών συστημάτων. Βασίζεται στην έννοια της κατάστασης (state). Επεκτείνει τις παραδοσιακές μηχανές πεπερασμένων καταστάσεων (FSMs) προσθέτοντας:
 - ✓ Ιεραρχία,
 - ✓ Παραλληλισμό και
 - ✓ επικοινωνία (βασίζεται στο broadcasting και στο σύγχρονο μοντέλο εκτέλεσης)
- **Petri Nets:** Αναπαριστούν το υπό ανάπτυξη σύστημα χρησιμοποιώντας διακριτά γεγονότα (discrete events)

Ταξινόμηση γλωσσών σχεδίασης (10)

Γλώσσες Παράλληλου Προγραμματισμού

- Έχουν χρησιμοποιηθεί για την περιγραφή συστημάτων υλικού λόγω των κοινών απαιτήσεων που υπάρχουν μεταξύ του σχεδιασμού συστημάτων υλικού και του παράλληλου προγραμματισμού
- Το βασικό του μειονέκτημα τους είναι η *έλλειψη της έννοιας του χρόνου*
- Η **OCCAM** και η **Unity** είναι δύο γλώσσες αυτής της κατηγορίας που έχουν χρησιμοποιηθεί τόσο για προδιαγραφές όσο και για ανάπτυξη συστημάτων

Ταξινόμηση γλωσσών σχεδίασης (11)

Γλώσσες Λειτουργικού Προγραμματισμού

- Οι γλώσσες λειτουργικού προγραμματισμού (functional programming languages) μπορούν να χρησιμοποιηθούν μαζί με αλγεβρικές αναπαραστάσεις (algebraic notations) για το σχεδιασμό συστημάτων υλικού.
- Παραδείγματα τέτοιων γλωσσών είναι οι:
 - **Haskell**
 - **VDM**
 - **Z**
 - **B**
 - ✓ Η B είναι μια τυπική γλώσσα (formal) η οποία χρησιμοποιείται για την τυπική περιγραφή του συστήματος
 - ✓ Μοντελοποιεί τόσο το σύστημα όσο και το περιβάλλον του
 - ✓ Παρέχει τη δυνατότητα μαθηματικά επιβεβαιωμένης εκλέπτυνσης του συστήματος (proven system refinement) για συστήματα υλικού/λογισμικού