

# Ενσωματωμένα Συστήματα

## Ενότητα 11: Επιταχυντές Υλικού & Πολυπύρρηνα ενσωματωμένα συστήματα.

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto:mdasyg@ieee.org)

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.icte.uowm.gr/mdasyg>



# Άδειες Χρήσης

---

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Σκοπός ενότητας

---

- Η κατανόηση της ανάγκης των επιταχυντών και της διαφορετικότητας τους από τους συνεπεξεργαστές.
- Η παρουσίαση των πολυπύρηνων ενσωματωμένων αρχιτεκτονικών.



---

# Κεντρικές μονάδες επεξεργασίας και επιταχυντές.



# Επιταχυντές υλικού

---

- Παρέχουν μεγάλες αυξήσεις απόδοσης για εφαρμογές με υπολογιστικούς πυρήνες που ξοδεύουν μεγάλο μέρος του χρόνου σε μικρό τμήμα του κώδικα.
- Καθοριστικές επιταχύνσεις για **λειτουργίες I/O** χαμηλού λανθάνοντα χρόνου.
- Κλασσικό παράδειγμα συ-σχεδίασης υλικού/λογισμικού (*hardware/software co-design*)
  - (κάτι που θα μπορούσε να υλοποιηθεί με *software* υλοποιείται με *hardware*).



# Συστήματα με επιτάχυνση

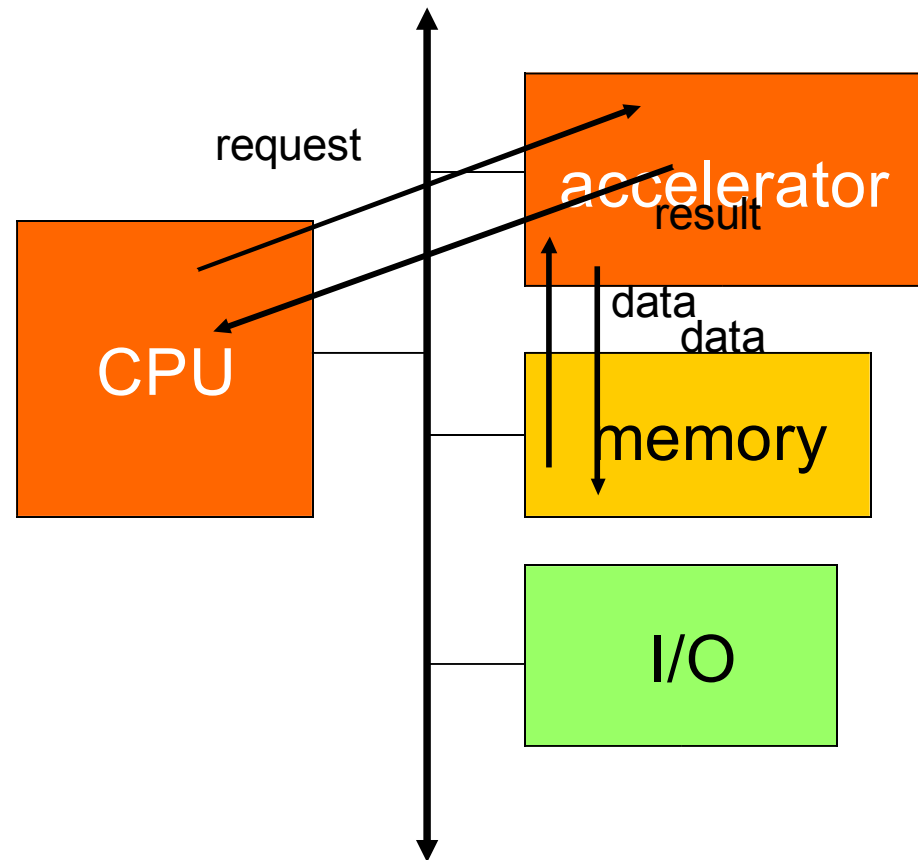
---

- Χρησιμοποιήστε μια επιπλέον υπολογιστική μονάδα αφιερωμένη σε ορισμένα καθήκοντα
  - Hardwired logic.
  - Extra CPU.
- **Hardware/software co-design:** προσάρμοσε το σχέδιο της αρχιτεκτονικής του υλικού και του λογισμικού.



# Επιταχυνόμενο σύστημα αρχιτεκτονικής

- Ο επιταχυντής συνδέεται στο δίαυλο της CPU.
- Η CPU επικοινωνεί με τον επιταχυντή μέσω **καταχωρητών** δεδομένων και ελέγχου.
- Η επικοινωνία μπορεί να γίνεται και μέσω **κοινής μνήμης** (ιδιαίτερα αν υπάρχουν πολλά δεδομένα)
  - {Απαιτείται συγχρονισμός}





# Επιταχυντής vs. συνεπεξεργαστής

---

## Ο επιταχυντής ΔΕΝ είναι συνεπεξεργαστής.

- Ένας συνεπεξεργαστής εκτελεί οδηγίες.
  - Οι οδηγίες αναθέτονται από τη CPU.
- Ένας επιταχυντής εμφανίζεται ως μία συσκευή πάνω στο bus.
  - Ο επιταχυντής ελέγχεται από καταχωρητές.



# Εφαρμογές Επιταχυντή

---

- Ολοκληρωμένα κυκλώματα ειδικού σκοπού εφαρμογών (*ASIC*).
- Προγραμματιζόμενες συστοιχίες πυλών στο πεδίο (*FPGA*).
- Προσαρμοσμένο Ολοκληρωμένο κύκλωμα (*VLSI*) {για μεγάλες ποσότητες}.
- Τυπικό υποσύστημα (*off the shelf*).
  - **Παράδειγμα:** επεξεργαστής γραφικών.
- Από τον κατάλογο (*IP core*).

(Τελικά, δε μας ενδιαφέρει που θα γίνει η επεξεργασία.)

Ονομάζονται:

**PE (processing elements)**



# Καθήκοντα σχεδιασμού συστήματος

---

- Σχεδίασε έναν ετερογενή πολυεπεξεργαστή αρχιτεκτονικής.
  - **Επεξεργασία στοιχείου (PE):**  
CPU, επιταχυντής κτλ.
- Προγραμματίσει το σύστημα.



# Γιατί επιταχυντές; (1/4)

---

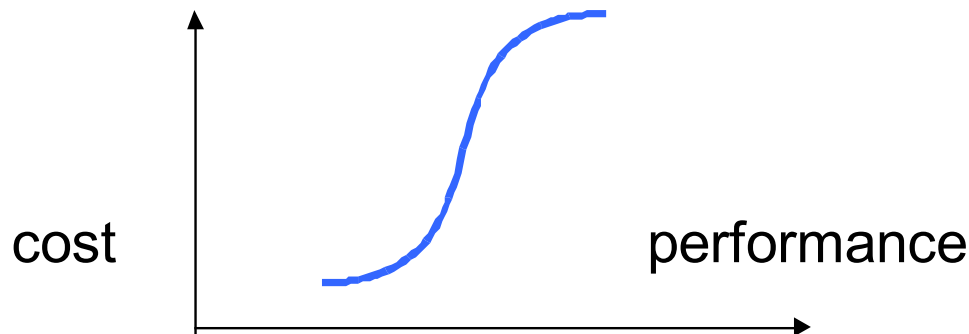
- Απαιτείται σημαντική προσπάθεια *(περισσότερη από μόνο σχεδίαση λογισμικού)*.
- Πρέπει το κέρδος να είναι σημαντικό.
- Καλύτερο κόστος/απόδοση.
  - Η προσαρμοσμένη λογική ίσως είναι ικανή να εκτελέσει τη λειτουργία πιο γρήγορα από τη CPU ισοδύναμου κόστους.
  - Το κόστος της CPU είναι μια μη γραμμική λειτουργία εκτέλεσης.



# Γιατί επιταχυντές; (2/4)

---

- Όσο πιο high performance τόσο πιο ακριβό εκθετικά.
- Ίδιες επιδόσεις με περισσότερα μικρότερα στοιχεία επεξεργασίας (*υψηλότερο όμως κόστος μηχανικού για ανάπτυξη και συναρμολόγηση*).



# Γιατί επιταχυντές; (3/4)

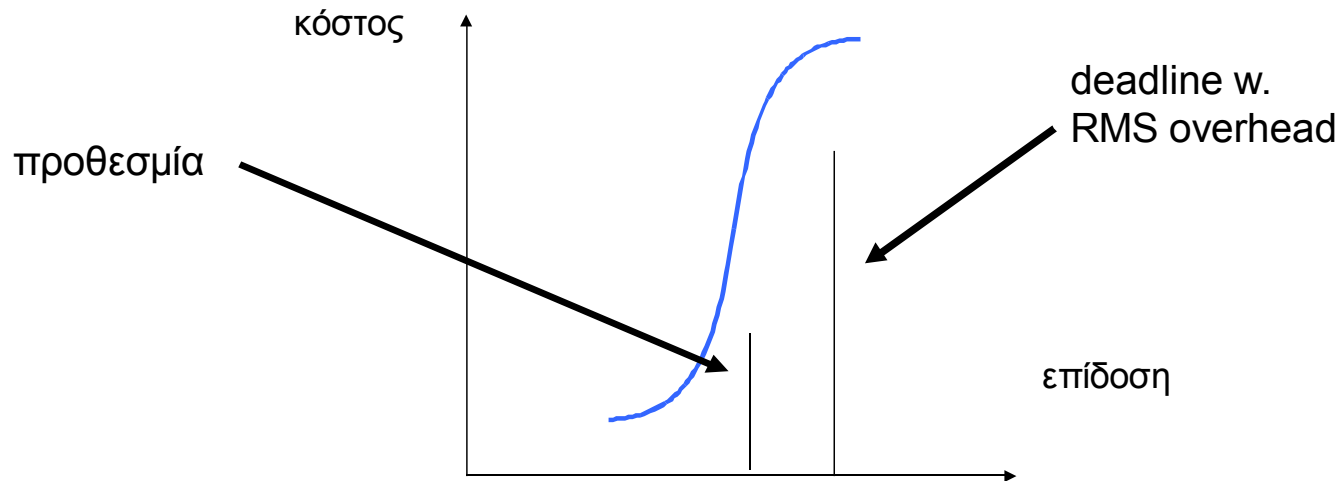
---

- Καλύτερη απόδοση σε πραγματικό χρόνο.
  - Βάλτε τις λειτουργίες σημαντικού χρόνου σε λιγότερο φορτωμένο στοιχεία επεξεργασίας.
  - Θυμηθείτε χρήση RMS (προγραμματισμού) --- Οι επιπλέον κύκλοι της CPU πρέπει να διατηρούνται για την τήρηση των προθεσμιών.



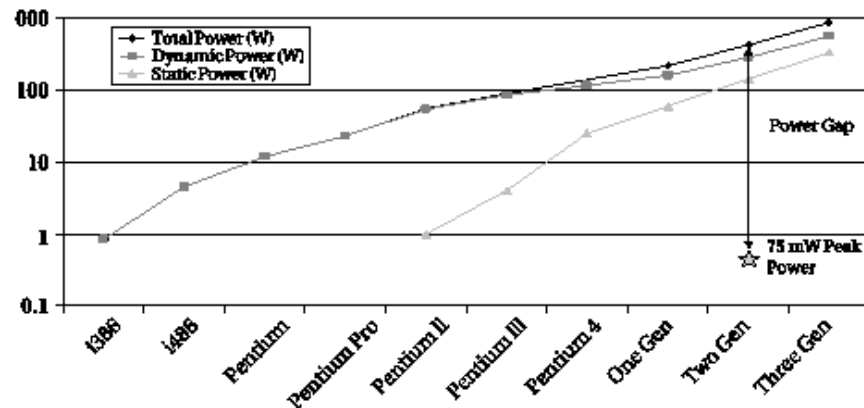
# Γιατί επιταχυντές; (4/4)

Κρίσιμες διεργασίες σε PE χωρίς  
ή με ελάχιστο time sharing  
(μείωση επιβάρυνσης  
χρονοπρογραμματισμού)



# Γιατί πολυεπεξεργαστές;

- Χρησιμοποιώντας εξειδικευμένους επεξεργαστές ή προσαρμοσμένη λογική, εξοικονομούμε ενέργεια.
- Οι επεξεργαστές των σταθμών εργασίας δεν είναι αρκετά ενεργειακά αποδοτικοί για εφαρμογές που λειτουργούν με μπαταρίες.



[Aus04] © 2004 IEEE Computer Society





# Γιατί επεξεργαστές;

- Καλό για επεξεργασία I/O σε πραγματικό χρόνο.
- Ίσως καταναλώσουν λιγότερη ενέργεια.
- Μπορεί να είναι καλύτεροι σε ροή δεδομένων.
- Μπορεί να μην είναι σε θέση να κάνει όλη τη δουλειά, ακόμα και τη μεγαλύτερη ενιαία CPU (πλήρη επεξεργασία βίντεο HD) {χρησιμοποίησε multi PEs}.

## Οι επεξεργαστές είναι καλοί για:

- Bit-wise λειτουργίες μπορεί να είναι δύσκολο για τον επεξεργαστή.
- I/O with strict deadlines.
- Ροή δεδομένων (δεν είναι καλό για την cache).



# Σχεδίαση επιταχυντή

---

- Πρώτον, προσδιόρισε ότι το σύστημα χρειάζεται πραγματικά να επιταχυνθεί.
  - Εκτίμησε τη χρησιμότητα (αν CPU απλός ελεγκτής, τότε η PE είναι καλύτερη, αλλά αν CPU χρειάζεται για υψηλής ποιότητας επεξεργασία τότε δεν μπορούμε εύκολα να αποφασίσουμε).
  - Πόσο γρηγορότερος είναι ο επιταχυντής στη βασική λειτουργία;
  - Πόσα δεδομένα μεταφέρει γενικά;
- Σχεδίασε τον ίδιο επιταχυντή {καλή κατανόηση του αλγορίθμου}.
- Σχεδιασμός της διασύνδεσης του επιταχυντή {χειραψία διαύλου, πως το software θα το χρησιμοποιεί, πως θα μεταφέρονται τα δεδομένα, αν θα υπάρχει λογική παραγωγής διευθύνσεων μνήμης για πολλά δεδομένα, πως θα γίνεται ο συγχρονισμός}.



---

Σχεδίαση συστημάτων με επιτάχυνση.



# Φάσεις Σχεδίασης

---

- Καθορισμός αρχιτεκτονικής.
- Σχεδίαση των συστατικών στοιχείων.
- Ολοκλήρωση του συστήματος.

*Η προδιαγραφή του συστήματος  
δεν αλλάζει, ο χρήστης αγνοεί την  
παρουσία του επιταχυντή.*



# Ανάλυση Απόδοσης

---

Σοβαρή παράμετρος είναι η επιτάχυνση:  
πόσο γρηγορότερο είναι το σύστημα με τον  
επιταχυντή;

*(συνήθως πρώτα εξετάζεται η απόδοση και μετά η κατανάλωση)*

- Πρέπει να ληφθεί υπόψη:
  - Χρόνος εκτέλεσης επιταχυντή.
  - Χρόνος μεταφοράς δεδομένων.
  - Συγχρονισμός με την κατοχή της CPU.



# Χρόνος εισαγωγής/εξαγωγής δεδομένων.

---

- Συναλλαγές bus περιλαμβάνουν:
  - Έξαψη τιμών καταχωρητή/μνήμη cache στην κεντρική μνήμη.
  - Χρόνος που απαιτείται για την cpu να δημιουργήσει συναλλαγές;
  - Επιβάρυνση της μεταφοράς δεδομένων από πακέτα bus, χειραψία, κλπ.



# Μονό εναντίον Πολλαπλού σπειρώματος

---

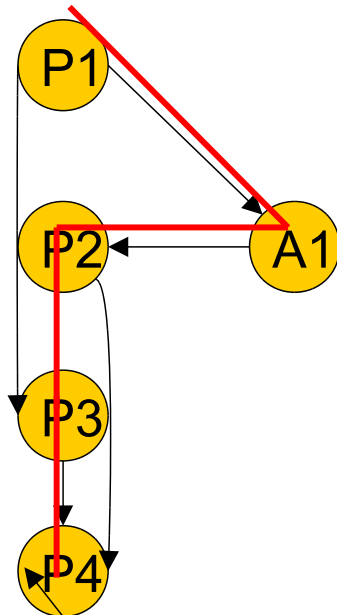
- Ένας σημαντικός παράγοντας είναι ο διαθέσιμος παραλληλισμός:
  - **μονονηματικό/κλείδωμα:**  
η CPU περιμένει τον επιταχυντή;
  - **πολυνηματικό/χωρίς κλείδωμα:**  
η CPU συνεχίζει την εκτέλεση ταυτόχρονα με τον επιταχυντή.
- Στο πολυνηματικό, η CPU πρέπει ασφαλώς να έχει κάποια χρήσιμη εργασία να κάνει.
  - Όμως, το λογισμικό πρέπει να υποστηρίζει τα πολλαπλά νήματα.
- Ισοδύναμη περιγραφή: **blocking** και **non-blocking**.

Ακολουθεί διάγραμμα...

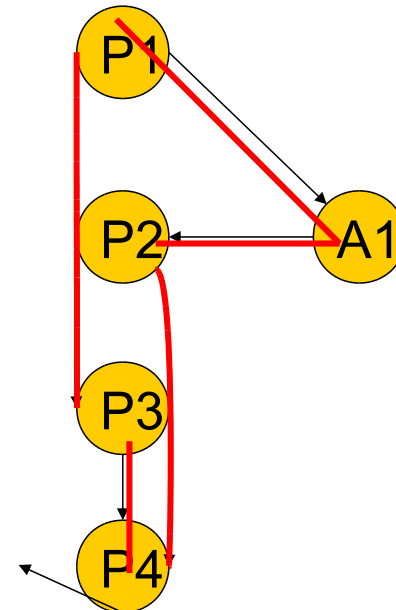


# Single/Multi νήματα

Μονονηματικό



Πολυνηματικό





# Χρόνος εκτέλεσης επιταχυντή

- Χρόνος εκτέλεσης επιταχυντή:
  - Χρόνος εκτέλεσης λειτουργίας.
  - Χρόνος μετάδοσης δεδομένων (προς και από).
- Συνολικός χρόνος εκτέλεσης επιταχυντή:
  - $t_{\text{accel}} = t_{\text{in}} + t_x + t_{\text{out}}$ 
    - ✓  $t_{\text{in}}$  ...δεδομένα εισόδου
    - ✓  $t_x$  ... ταχύτητα υπολογισμού
    - ✓  $t_{\text{out}}$  ... δεδομένα εξόδου

Πρέπει να λαμβάνεται η WCET αφού η μεταφορά στο δίαυλο μπορεί να καθυστερήσει (λόγω διενέξεων).



# Βελτίωση I/O

- Ένας εξελιγμένος επιταχυντής θα μπορούσε να επικαλύψει την είσοδο και την έξοδο με υπολογισμό (~"διασωλήνωση").
  - Διαβάζει αρχικές τιμές.
  - Αρχίζει υπολογισμούς με αυτές τις τιμές.
  - Ταυτόχρονα διαβάζει τις υπόλοιπες τιμές.
  - Τότε το  $t_{in}$ ,  $t_{out}$  είναι οι μη επικαλυπτόμενοι χρόνοι ανάγνωσης/εγγραφής.  
Παράδειγμα: digital filtering.

Δημιουργία αποτελεσμάτων εν κινήσει (*on-the-fly*) παρά την αποθήκευση όλων και την πραγματοποίηση υπολογισμών.



# Πηγές παραλληλισμού

---

- Επικάλυψη I/O και επιτάχυνση υπολογισμού.
  - Εκτέλεση εργασιών κατά ομάδες, διάβασμα δεδομένων στη δεύτερη παρτίδα, ενώ ο υπολογισμός βρίσκεται στην πρώτη ομάδα.
- “Βρες άλλη δουλειά να κάνεις στη CPU”.
  - Μπορεί να αναδιαρθρώσει ενέργειες για να μετακινηθεί το έργο μετά την έναρξη του επιταχυντή.



# Χρόνος εισόδου/εξόδου δεδομένων

---

- Οι συναλλαγές bus περιλαμβάνουν:
  - Αποθήκευση καταχωρητή/τιμή cache στη κύρια μνήμη.
  - Χρόνος που απαιτείται για την cpu να δημιουργήσει συναλλαγές;
  - Επιβάρυνση της μεταφοράς δεδομένων από πακέτα bus, χειραψία, κλπ.



# Επιτάχυνση επιταχυντή

- Ας υποθέσουμε ότι ο βρόχος εκτελείται  $n$  φορές.
- Σύγκριση επιταχυνόμενου συστήματος με μη-επιταχυνόμενο σύστημα:

$$\begin{aligned} \checkmark \quad S &= n(t_{\text{CPU}} - t_{\text{accel}}) \\ &= n[t_{\text{CPU}} - (t_{\text{in}} + t_x + t_{\text{out}})] \\ &\quad - \text{Όπου } t_{\text{CPU}} \dots \text{ Execution time on CPU} \end{aligned}$$

Όσες πιο πολλές φορές καλείται ο επιταχυντής τόσο περισσότερο είναι το όφελος.



# Ανάλυση χρόνου εκτέλεσης

---

Τελικά, μας ενδιαφέρει όχι μόνο η επιτάχυνση του επιταχυντή αλλά η επιτάχυνση για το πλήρες σύστημα.

- **Μονονηματικό:**

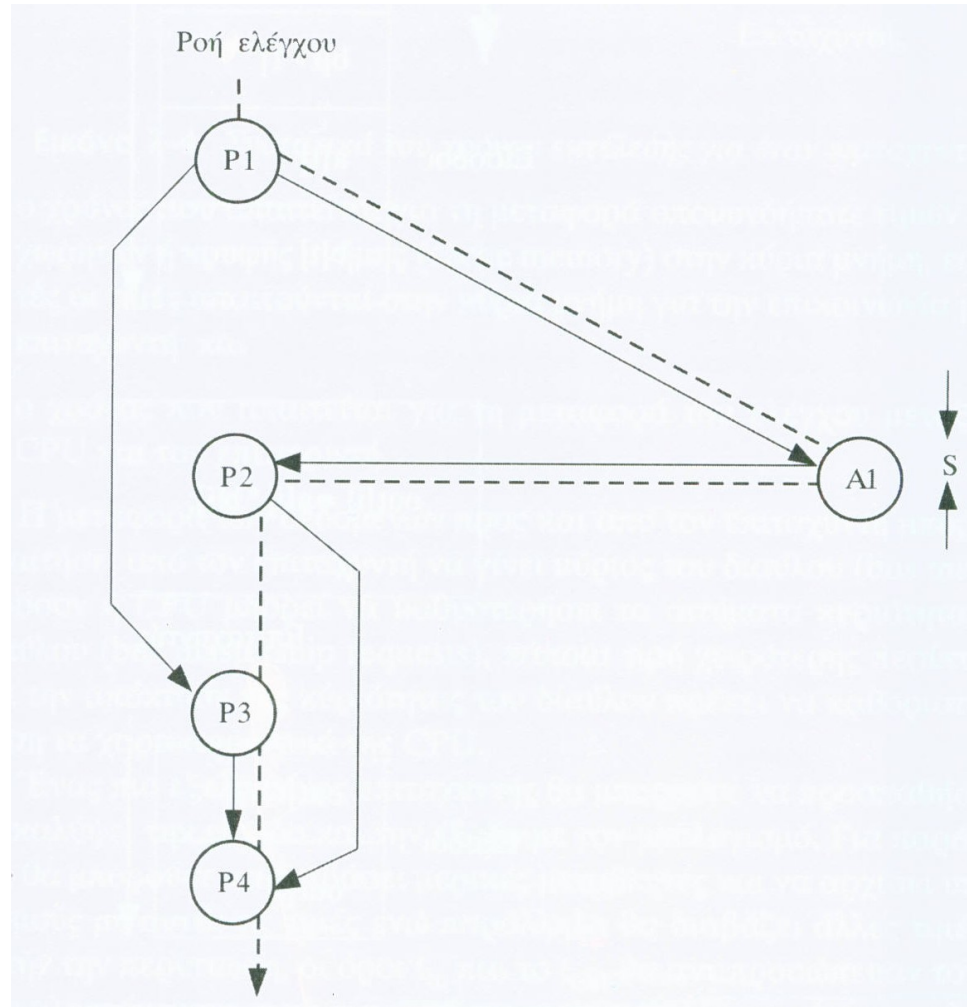
- Υπολογίζουμε το χρόνο εκτέλεσης όλων των διαδικασιών της συνιστώσας.

- **Πολυνηματικό:**

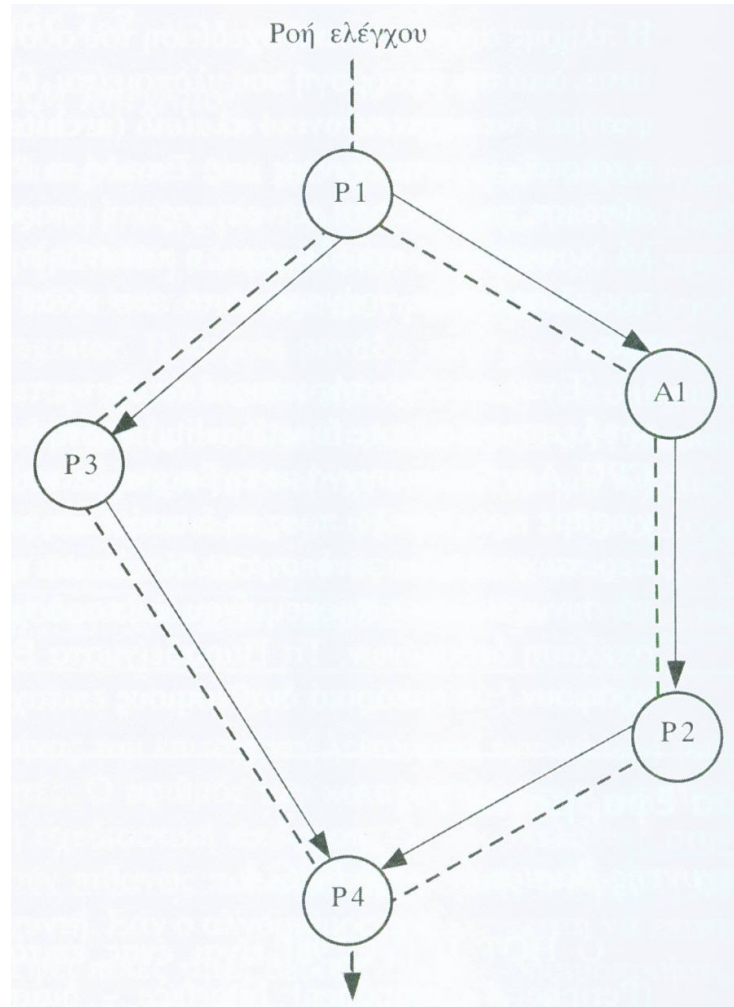
- Βρίσκουμε το μακρύτερο μονοπάτι εκτέλεσης.



# Αξιολόγηση Επιτάχυνσης σε μονονηματική υλοποίηση



# Αξιολόγηση Επιτάχυνσης σε πολυνηματική υλοποίηση





# Επιταχυνόμενο σύστημα πλατφορμών (1/2)

---

- Αρκετές έτοιμες πλακέτες είναι διαθέσιμες για την επιτάχυνση σε υπολογιστές:
  - FPGA-based core;
  - PC bus interface.



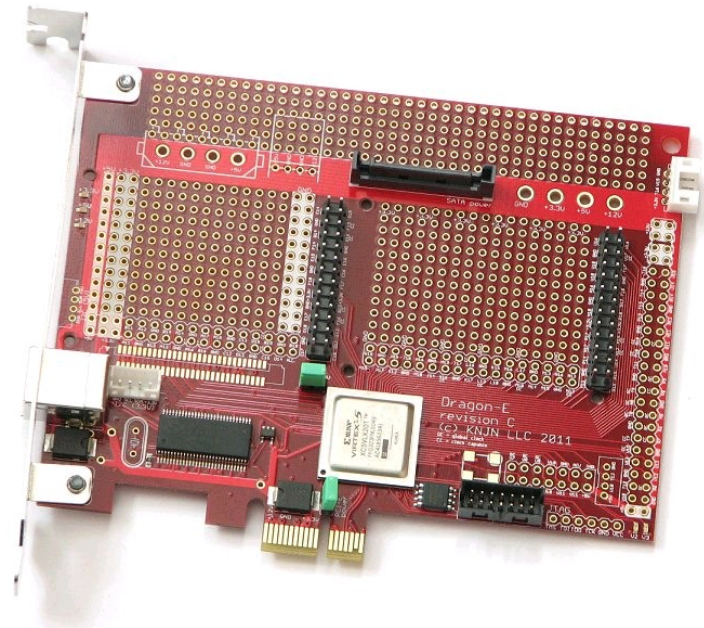
# Επιταχυνόμενο σύστημα πλατφορμών (2/2)

---

**FPGA in PCI**



**FPGA in PCI express**



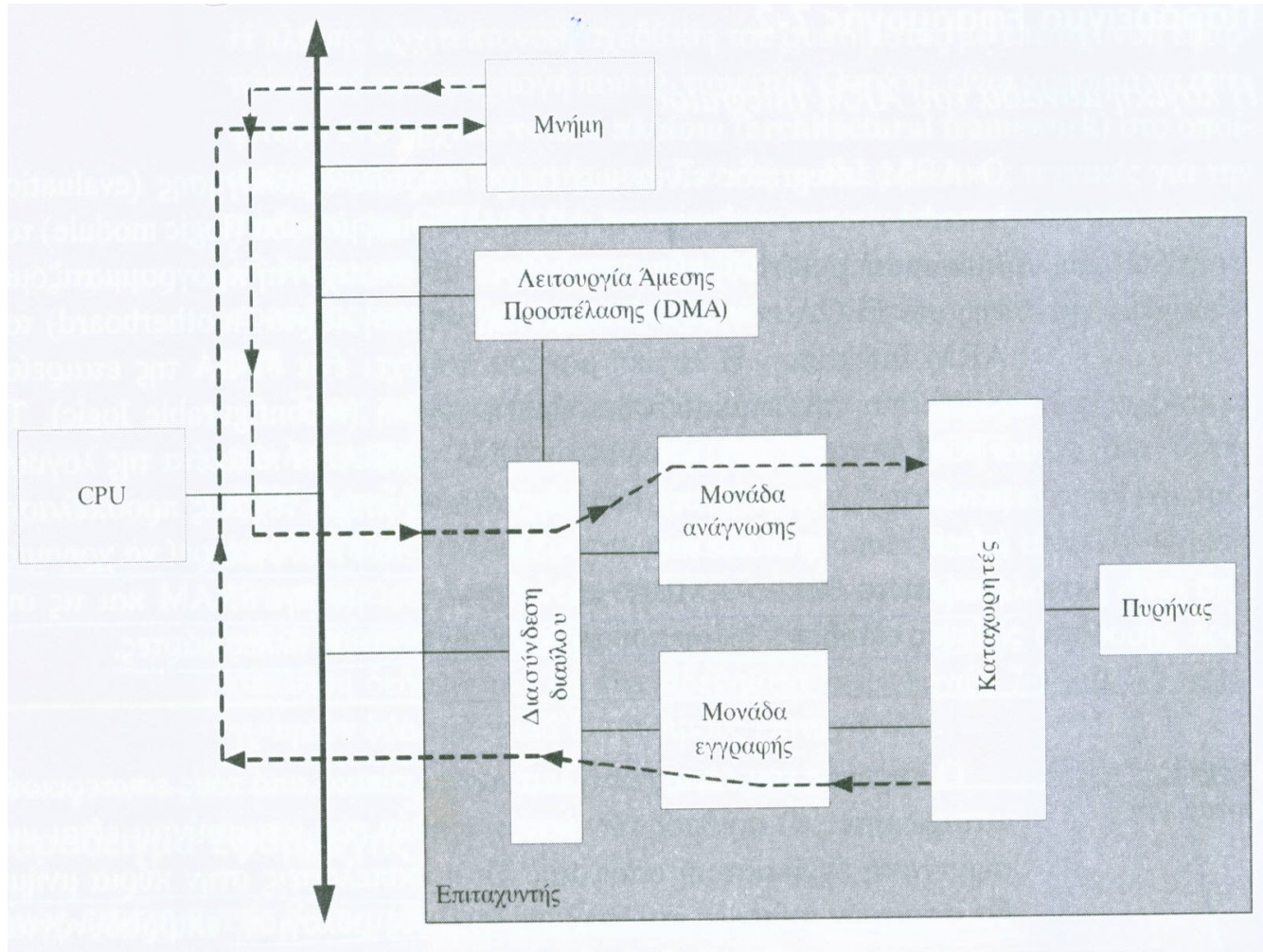
# Επιταχυντής/CPU διασύνδεση

---

- Οι καταχωρητές επιτάχυνσης παρέχουν καταχωρητές ελέγχου για την CPU.
- Οι καταχωρητές δεδομένων μπορούν να χρησιμοποιηθούν για μικρά αντικείμενα δεδομένων.
- Μπορεί να προσκομίζονται μεγάλου όγκου δεδομένα.
- Οι επιταχυντές μπορεί να περιλαμβάνουν read/write λογική για ειδικούς σκοπούς.
  - Ιδιαίτερα πολύτιμη για μεγάλες μεταφορές δεδομένων.



# Διασύνδεση



# Ολοκλήρωση του συστήματος και τον εντοπισμό σφαλμάτων

---

- Προσπαθήστε να αποσφαλματώσετε τον CPU/επιταχυντή διεπαφής ξεχωριστά από τον πυρήνα του επιταχυντή.
- Κατασκευάστε μοντέλα για να ελέγξετε τον επιταχυντή.
- Συν-προσομοίωση υλικού/λογισμικού μπορεί να είναι χρήσιμη.



# Προσωρινής αποθήκευσης προβλήματα

---

- Η κύρια μνήμη παρέχει το μηχανισμό μετάδοσης των κυρίων δεδομένων προς τον επιταχυντή.
- Τα προγράμματα πρέπει να διασφαλίζουν ότι η προσωρινή αποθήκευση δεν αναιρεί την κύρια μνήμη.
  - Η CPU διαβάζει την τοποθεσία S
  - Ο επιταχυντής γράφει την τοποθεσία S.
  - Η CPU γράφει την τοποθεσία S. **BAD**

**Μπορεί να απαιτείται συγχρονισμός σε προσβάσεις κοινών δεδομένων.**



# Συγχρονισμός (1/3)

- Όπως με την μνήμη cache, η κύρια μνήμη γράφει στην κοινή μνήμη και αυτό μπορεί να προκαλέσει ακύρωση:
  - CPU διαβάζει S.
  - Ο επιταχυντής γράφει S.
  - CPU διαβάζει S.

Απαιτείται ενημέρωση από τον επιταχυντή προς την κρυφή μνήμη για να ακυρώσει (invalidate) την αντίστοιχη γραμμή.



# Συγχρονισμός (2/3)

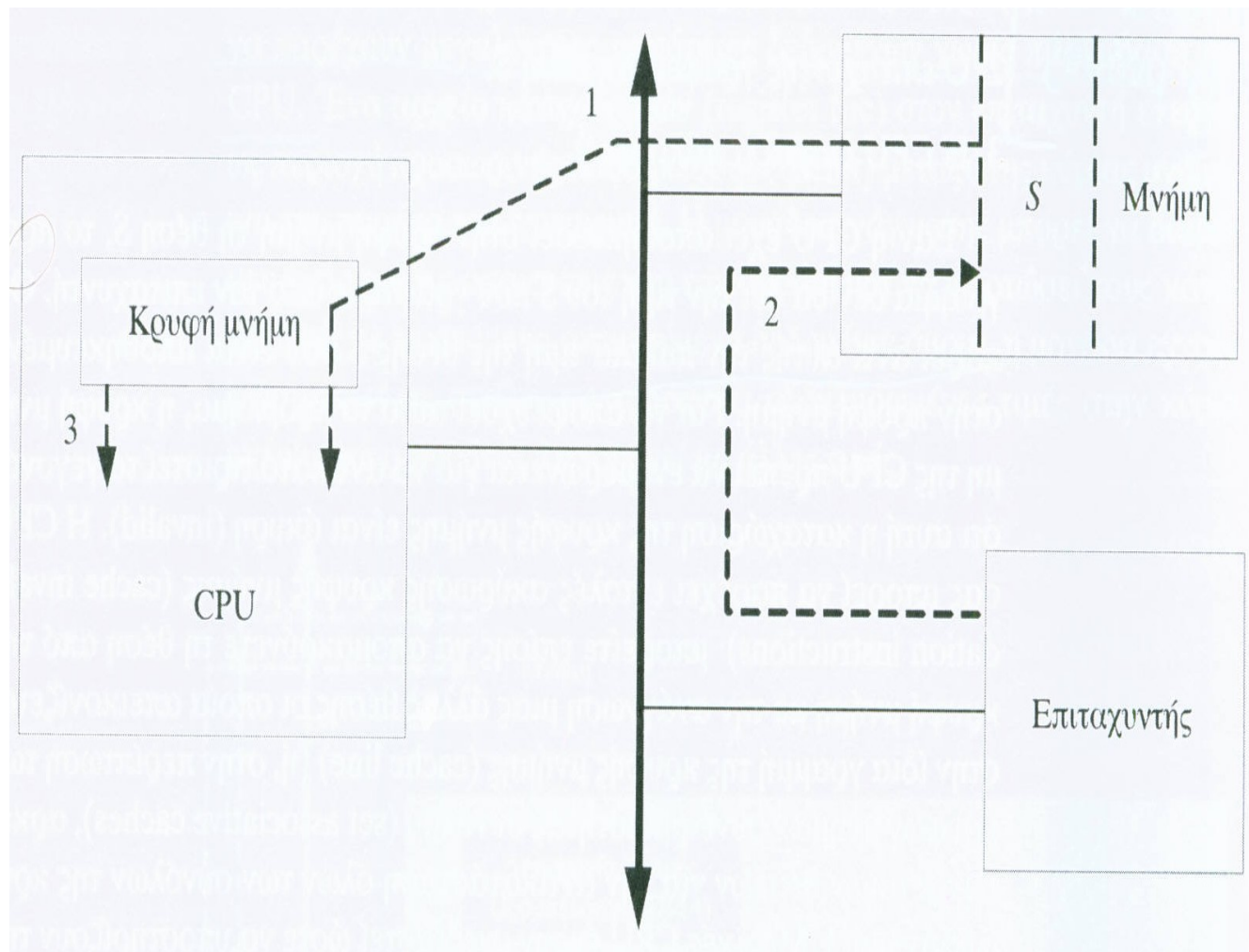
---

- Ένας απλός συγχρονισμός μπορεί να επιτευχθεί με τη χρήση συγκεκριμένης διεύθυνσης κοινής μνήμης (ως σημαφόρος).
- Η CPU διαβάζει μια περιοχή κρυφής μνήμης και όταν ο επιταχυντής ολοκληρώσει το έργο του, γράφει κάποια τιμή σε αυτή τη θέση μνήμης (*one-way synchronization*).
- Μπορεί να γίνει και συγχρονισμός και του επιταχυντή προς τη CPU, αν ο διάυλος υποστηρίζει ατομικές λειτουργίες Test-And-Set, οπότε, αναλόγως του ποια τιμή είναι γραμμένη σε αυτή τη θέση, το αντίστοιχο PE έχει πρόσβαση.





# Συγχρονισμός (3/3)



# Κατανομή (1/2)

---

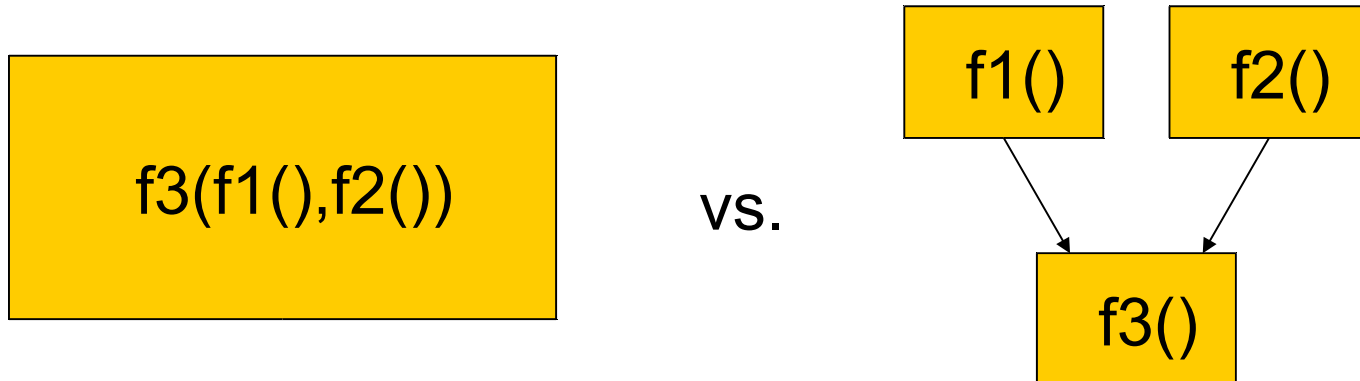
- Χωρίστε λειτουργικές προδιαγραφές σε μονάδες.
  - Χάρτες μονάδων σε PEs, CPUs.
  - Οι μονάδες μπορούν να γίνουν διαδικασίες.
- Σημαντικές λόγω:
  - **(α)** αποτελεσματικότητας
  - **(β)** εντοπισμού σφαλμάτων.



# Κατανομή (2/2)

---

- Καθορίστε κατάλληλο επίπεδο παραλληλισμού:



- ✓ 3 διεργασίες με εξαρτήσεις δεδομένων.
- ✓  $f1()$ ,  $f2()$  όμως μπορούν να γίνουν παράλληλα.

# Μεθοδολογία καταμερισμού

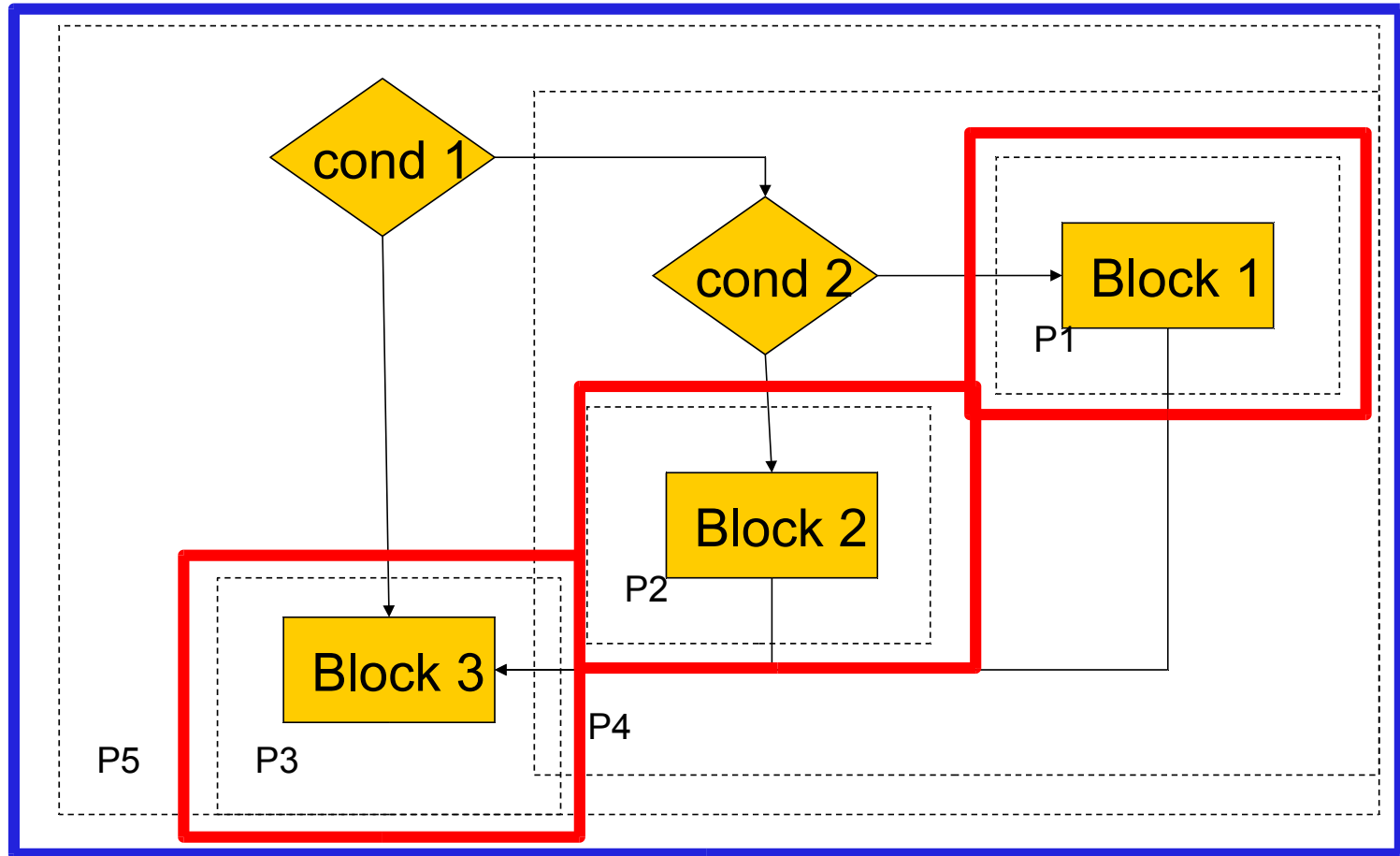
---

Χώρισε το CDFG σε κομμάτια, μοίρασε τις λειτουργίες ανάμεσα στα κομμάτια.

- Ιεραρχικά αποσύνδεσε το CDFG για να αναγνωρίσεις πιθανούς καταμερισμούς.
- Η αξιολόγηση της ποιότητας απαιτεί την αξιολόγηση της απόδοσης.
- Το πρόβλημα της διαμέρισης είναι η διαίρεση των γραφημάτων ροής ελέγχου/δεδομένων.



# Παράδειγμα Κατάτμησης



Μπορεί να αξιολογηθεί σε διαφορετικά επίπεδα αναλυτικότητας



# Προγραμματισμός και κατανομή

- Απαραίτητες ενέργειες:
  - Προγραμματίστε τις λειτουργίες εγκαίρως *(επικοινωνία + υπολογισμούς, αν υπάρξει καθυστέρηση ολοκλήρωσης τότε θα καθυστερήσει η επικοινωνία και ίσως έρθει σε διένεξη με άλλο PE)*;
  - Αναθέστε τους υπολογισμούς στα στοιχεία επεξεργασίας *(καθορίζει ποιες επικοινωνίες απαιτούνται)*.
- Προγραμματισμός και ανάθεση αλληλεπιδρούν, αλλά ο διαχωρισμός τους βοηθά το σχεδιαστή.
  - Εναλλακτικά κατένειμε, μετά προγραμματίσε.

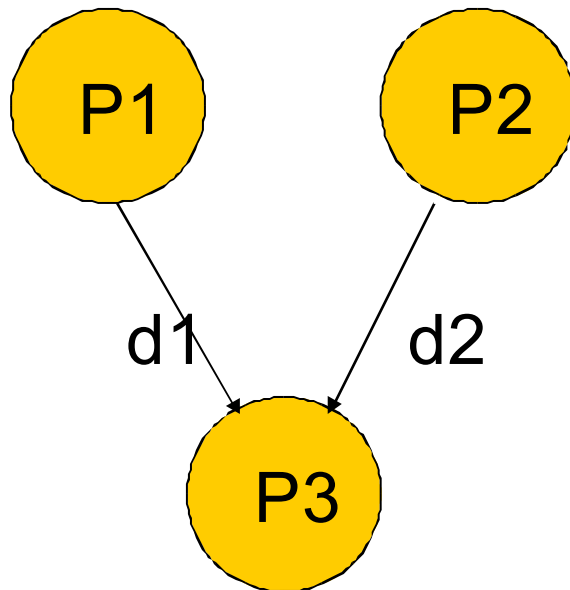


# Παράδειγμα: Προγραμματισμός και κατανομή

---

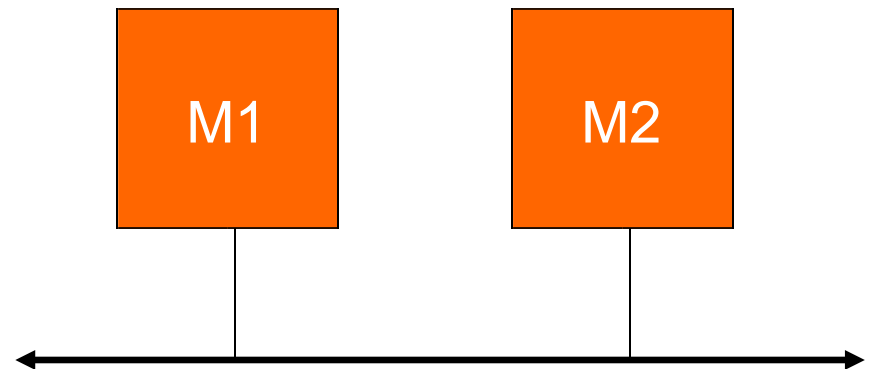
## Task graph:

- **P**: διεργασίες.
- **d**: επικοινωνία.



## Hardware platform:

- Κοινός δίαυλος.
- Διαφορετικές δυνατότητες ανά M.



# Παράδειγμα: επεξεργασία χρόνου εργασίας

---

	M1	M2
P1	5	5
P2	5	6
P3	-	5

- Η – σημειώνει ότι δε μπορεί να εκτελεστεί σε αυτό (π.χ. δεν έχει αρκετή μνήμη, ή δε μπορεί να κάνει αυτή τη λειτουργία).
- Διαφορετικοί χρόνοι εκτέλεσης γιατί διαφορετική ISA ή άλλη συχνότητα ρολογιού.





# Παράδειγμα επικοινωνίας μοντέλου

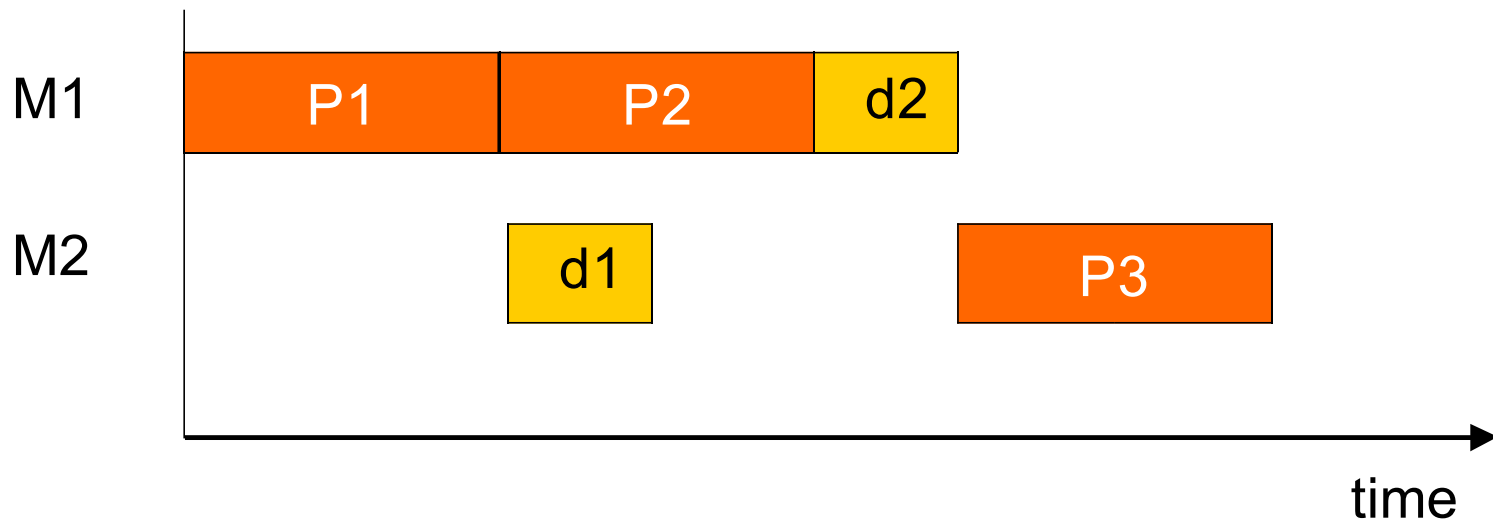
---

- Υπολόγισε ότι η επικοινωνία στο ΡΕ είναι ελεύθερη.
- Κόστος επικοινωνίας από Ρ1 μέχρι Ρ3 είναι  $d_1=2$ ; κόστος του Ρ2→Ρ3 επικοινωνία είναι  $d_2= 4$ .



# Πρώτη σχεδίαση

- Κατανέμω
  - P1, P2 → M1;
  - P3 → M2.

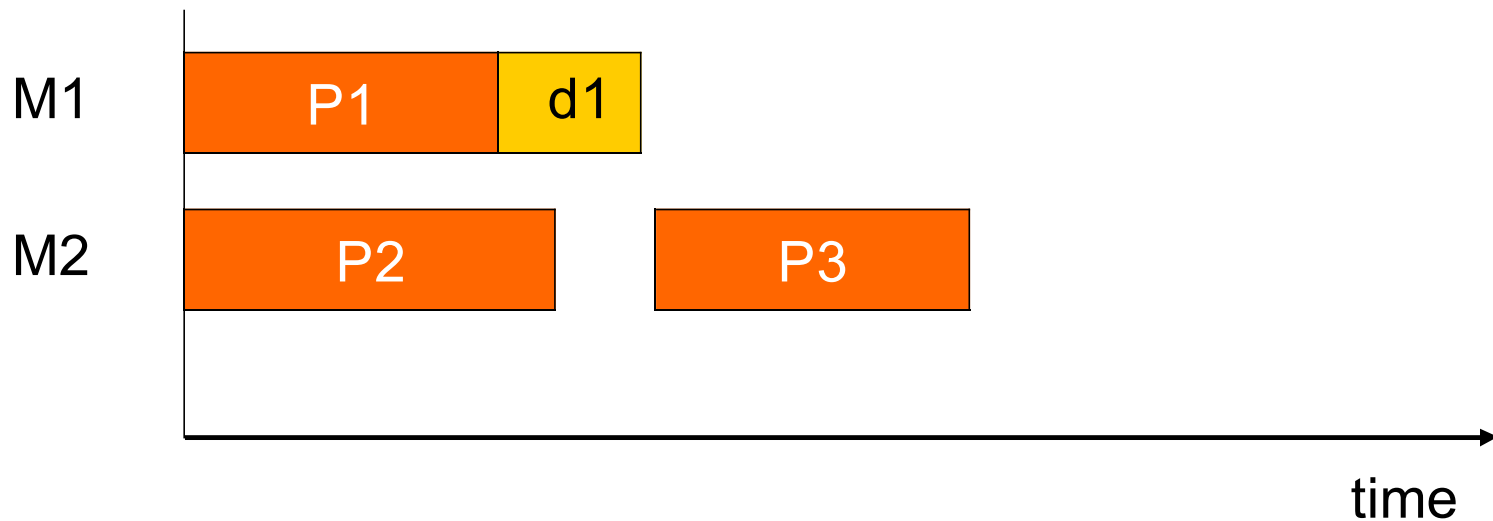


**Κόστος:  $5+5+4+5=19$**



# Δεύτερη σχεδίαση

- Κατανέμω
  - P1 → M1;
  - P3, P2 → M2.



Η P2 εκτελείται πιο αργά,  
αλλά δε μας προκαλεί πρόβλημα.

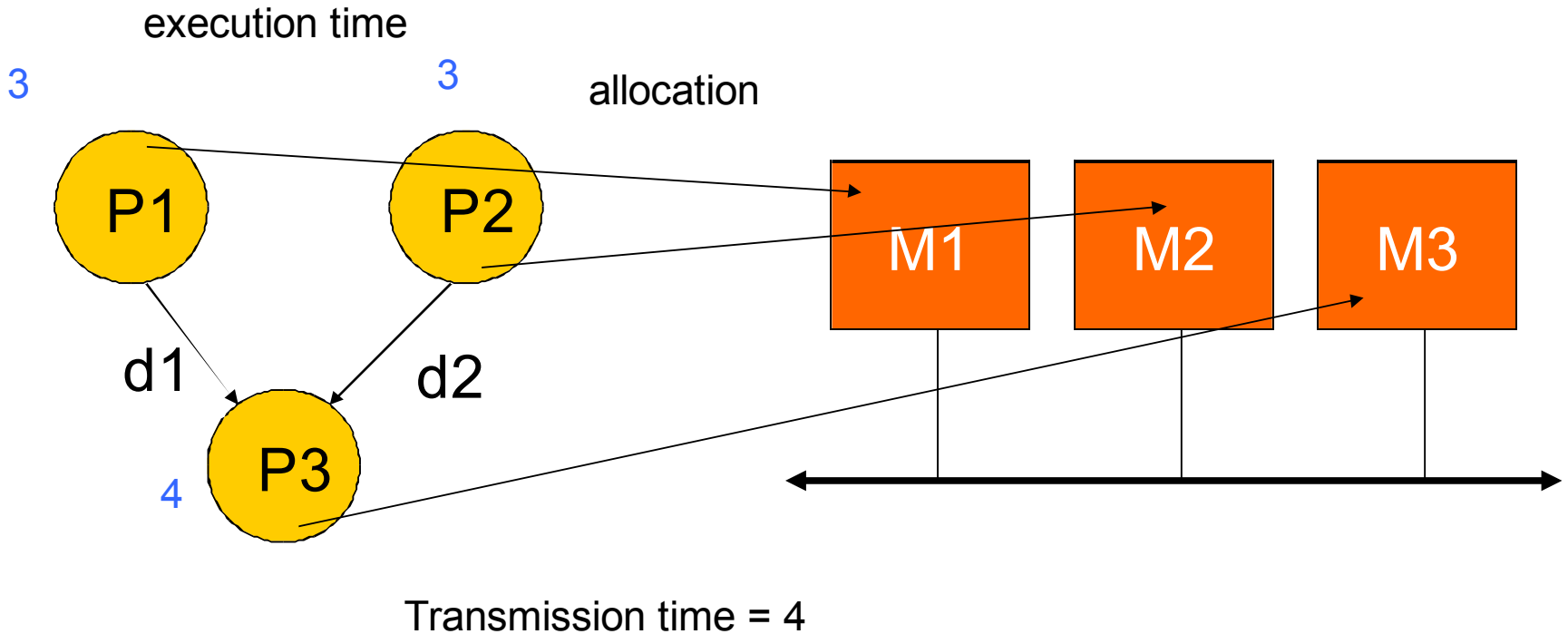
Κόστος:  $5+2+5=12$



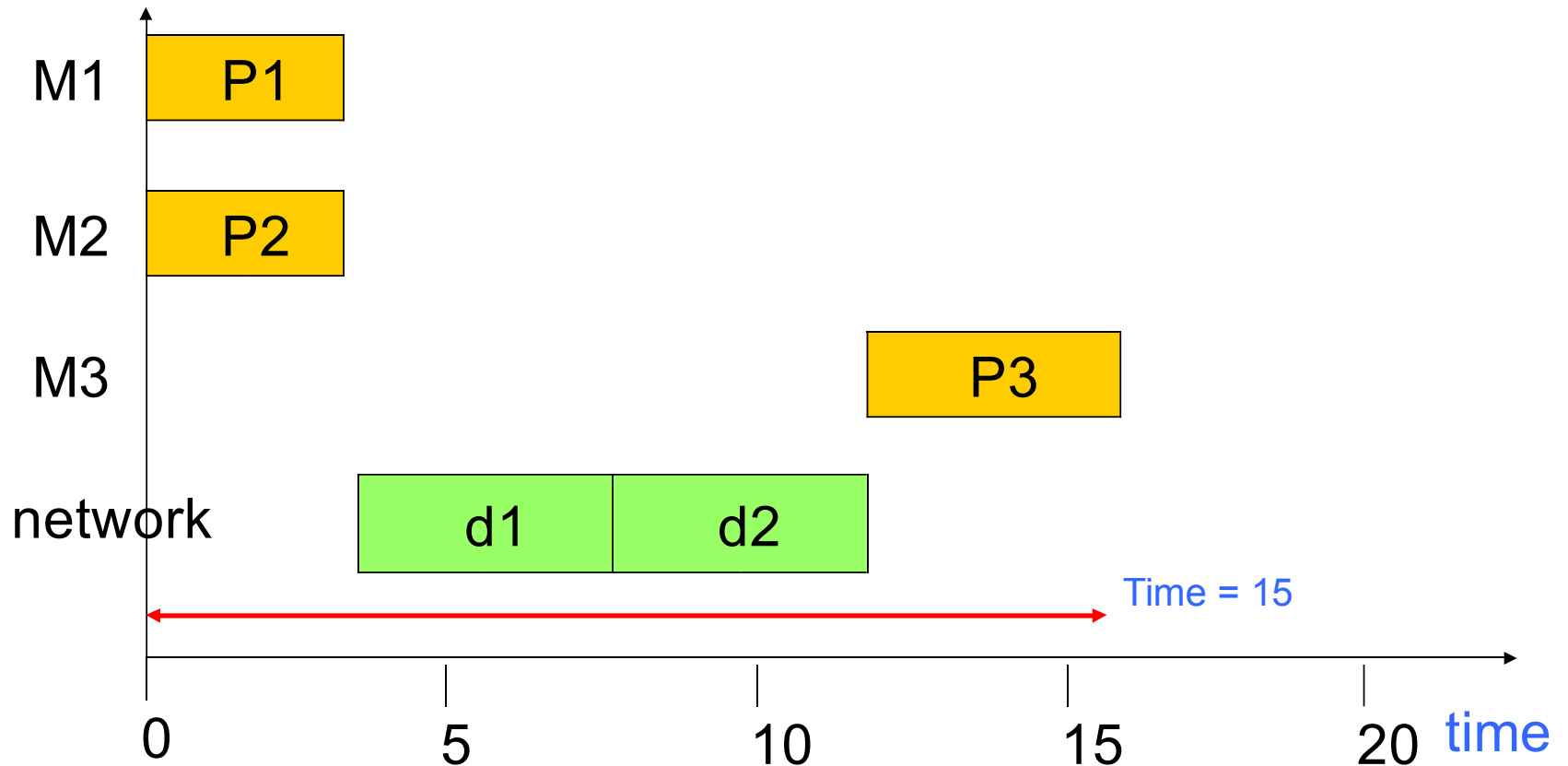
# Παράδειγμα: Προσαρμογή μηνυμάτων για ελάττωση καθυστέρησης

**Task graph:**

**Network:**



# Αρχικό Πρόγραμμα



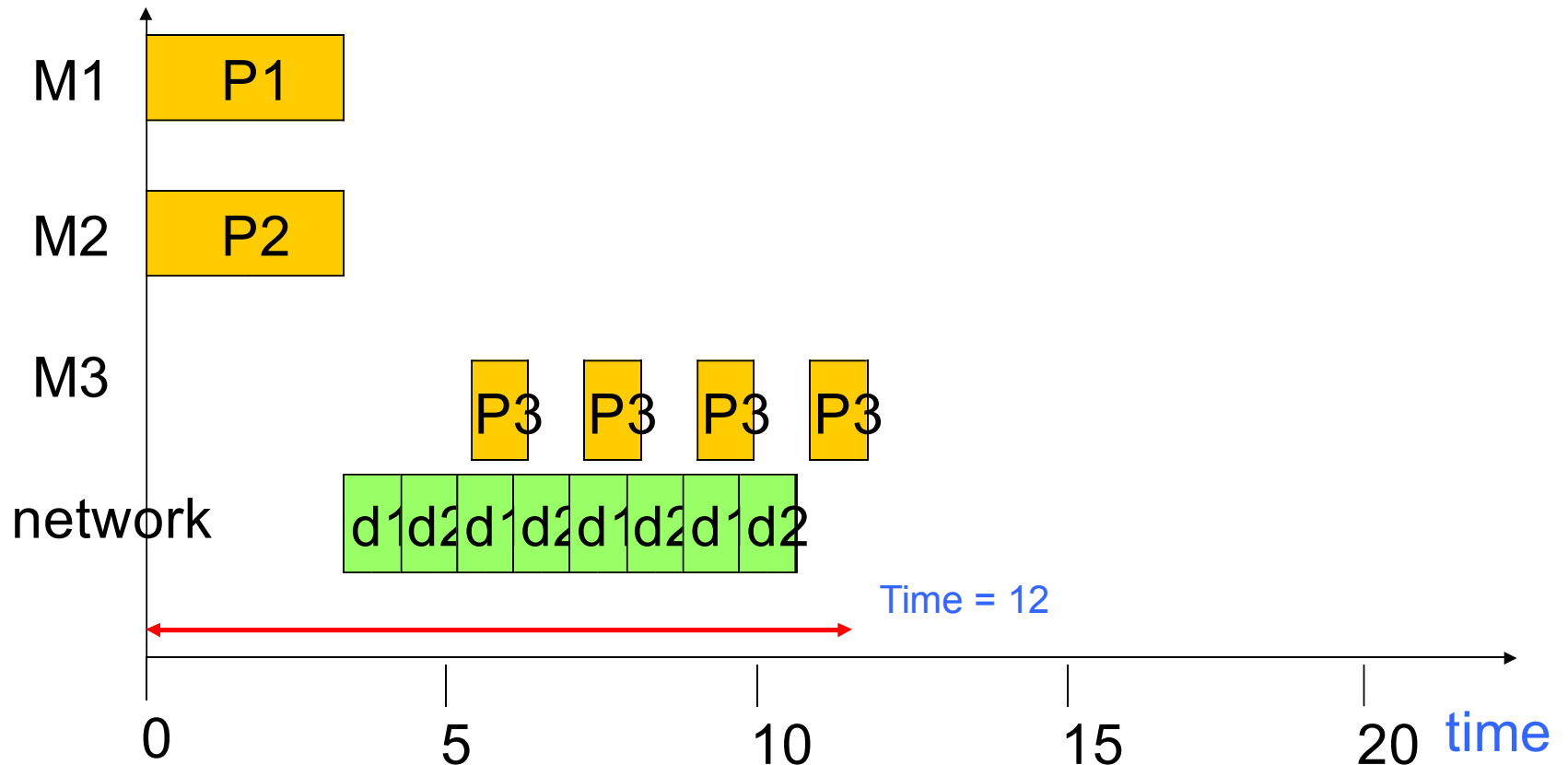
# Νέα σχεδίαση (1/2)

---

- Τροποποιώ P3:
  - Διαβάζει ένα πακέτο του  $d1$ , ένα πακέτο του  $d2$ .
  - Υπολογίζει το μερικό αποτέλεσμα.
  - Συνεχίζει στο επόμενο πακέτο.



# Νέα σχεδίαση (2/2)



# Ολοκλήρωση του συστήματος και εντοπισμός σφαλμάτων

---

- Προσπαθήστε να εντοπίσετε τα σφάλματα της CPU/επιταχυντή διασύνδεσης, χωριστά από τον πυρήνα του επιταχυντή.
- Χτίζετε μια αρχιτεκτονική σκελετού (προσωρινό μοντέλο συστήματος) για να ελέγξετε τον επιταχυντή.
- Η συν-προσομοίωση υλικού/λογισμικού μπορεί να είναι χρήσιμη.





# Ρύθμιση και επίδοση

---

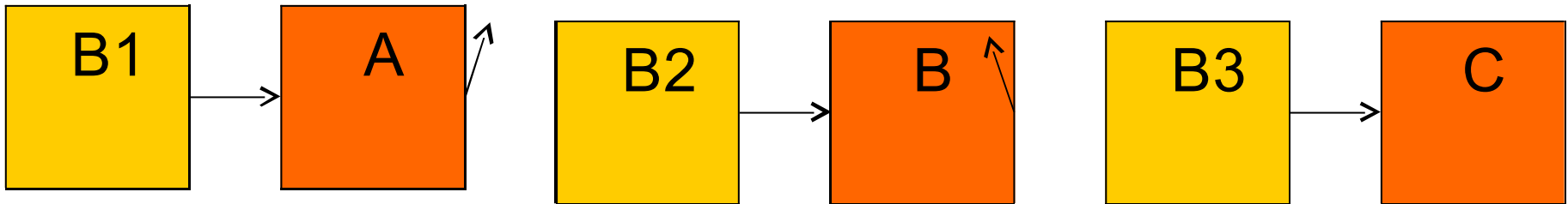
- Η χρήση ενδιάμεσης μνήμης (*buffer*) ίσως βάζει στη σειρά κάποιες λειτουργίες.
  - Η επόμενη διαδικασία πρέπει να περιμένει τα δεδομένα να εισέλθουν στο *buffer* για να μπορέσει να συνεχίσει.
- Η πολιτική του *buffer* (*queue*, *RAM*) επηρεάζει το διαθέσιμο παραλληλισμό.



# Buffers και καθυστέρηση

---

- Τρεις διαδικασίες χωρισμένες με buffer:



# Buffers και χρονοπρογραμματισμός καθυστέρησης

---

## Χωρίς buffer

A[0]

A[1]

... ← Must wait for all  
of A before getting any B

B[0]

B[1]

...

C[0]

C[1]

...

## Με buffer

A[0]

B[0]

C[0]

A[1]

B[1]

C[1]

...



---

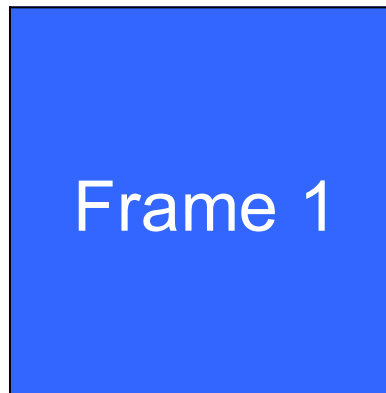
**Παράδειγμα: επιταχυντής βίντεο.**



# Κεντρική ιδέα (1/4)

---

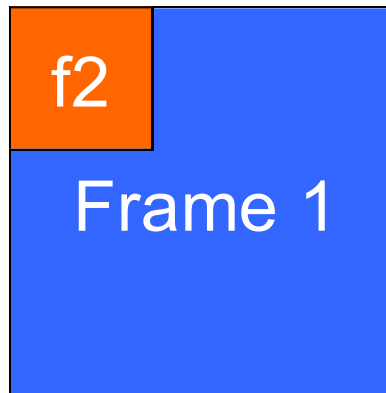
- Κατασκευάζουμε τον επιταχυντή για την εκτίμηση κίνησης μπλοκ, ένα τμήμα της συμπίεσης βίντεο.
- Εκτελούμε τη δισδιάστατη συσχέτιση:



# Κεντρική ιδέα (2/4)

---

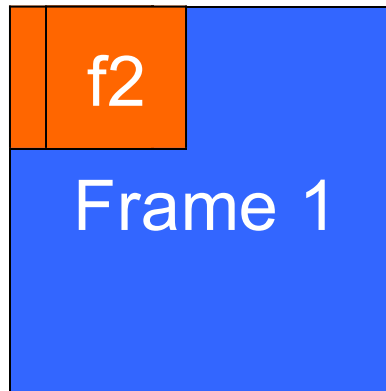
- Κατασκευάζουμε τον επιταχυντή για την εκτίμηση κίνησης μπλοκ, ένα τμήμα της συμπίεσης βίντεο.
- Εκτελούμε τη δισδιάστατη συσχέτιση:



# Κεντρική ιδέα (3/4)

---

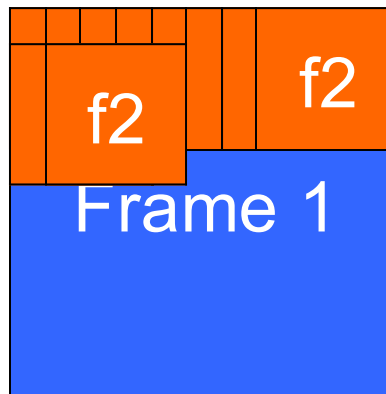
- Κατασκευάζουμε τον επιταχυντή για την εκτίμηση κίνησης μπλοκ, ένα τμήμα της συμπίεσης βίντεο.
- Εκτελούμε τη δισδιάστατη συσχέτιση:



# Κεντρική ιδέα (4/4)

---

- Κατασκευάζουμε τον επιταχυντή για την εκτίμηση κίνησης μπλοκ, ένα τμήμα της συμπίεσης βίντεο.
- Εκτελούμε τη δισδιάστατη συσχέτιση:





# Εκτίμηση κίνησης μπλοκ

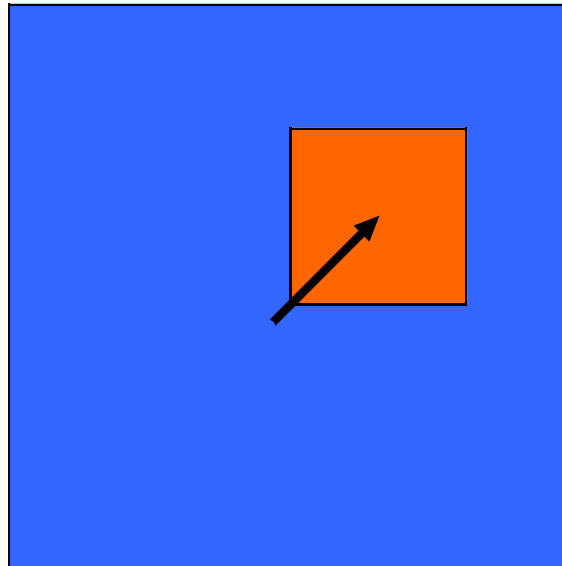
- Το MPEG χωρίζει το πλαίσιο σε 16 x 16 μακροομάδων για εκτίμηση της κίνησης.
- Αναζήτηση για το καλύτερο ταίριασμα σε ένα εύρος της αναζήτησης.
- Μετρά την ομοιότητα με το άθροισμα της απόλυτης διαφοράς(SAD) :
  - $\sum | M(i,j) - S(i-o_x, j-o_y) |$ 
    - ✓  $M(i,j)$  ένταση του μακρομπλοκ στο pixel  $i,j$
    - ✓  $S(i,j)$  ένταση της περιοχής αναζήτησης
    - ✓ Επιλέγεται το block που έχει το μικρότερο άθροισμα των διαφορών, και δημιουργείται το διάνυσμα κίνησης.



# Το καλύτερο ταίριασμα

---

- Το καλύτερο ταίριασμα παράγει το διάνυσμα κίνησης για το μπλοκ:



# Αλγόριθμος full search

---

```
bestx = 0; besty = 0;
bestsad = MAXSAD;
for (ox = - SEARCHSIZE; ox < SEARCHSIZE; ox++) {
    for (oy = -SEARCHSIZE; oy < SEARCHSIZE; oy++) {
        int result = 0;
        for (i=0; i<MBSIZE; i++) {
            for (j=0; j<MBSIZE; j++) {
                result + = iabs(mb[i][j] – search[i-ox+XCENTER]
[j-oy-YCENTER]);
            }
        }
        if (result <= bestsad) { bestsad = result; bestx = ox; besty = oy; }
    }
}
```



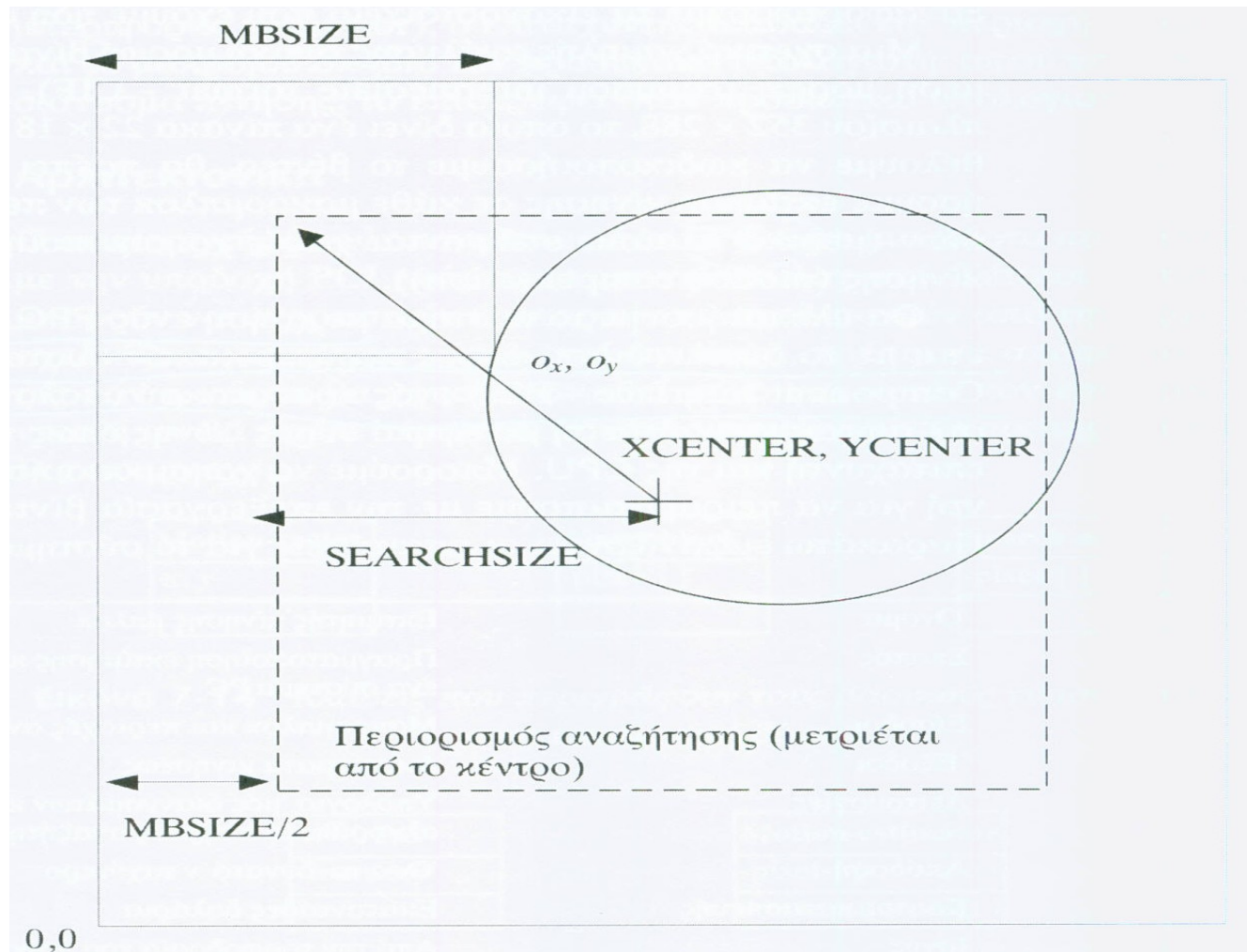
# Υπολογιστικές απαιτήσεις (1/2)

---

- Let MBSIZE = 16, SEARCHSIZE = 8.
- Περιοχή αναζήτησης είναι  $8 + 8 + 1$  σε κάθε διάσταση.
- Πρέπει να εκτελέσετε:
- $n_{ops} = (16 \times 16) \times (17 \times 17) = 73984$  ops
- CIF μορφή έχει  $352 \times 288$  pixels  $\rightarrow$   $22 \times 18$  macroblocks.



# Υπολογιστικές απαιτήσεις (2/2)



# Αριθμός λειτουργιών

---

- Mbsize=16
- Searchsize=8
- Απόσταση αναζήτησης 8+1+8
- $N_{ops} = (16 \times 17) \times (17 \times 17) = 73894$  λειτουργίες απόλυτης διαφοράς σε κάθε macroblock
- Πρέπει να εκτελεστεί σε κάθε macroblock (π.χ. σε CIF 352x288, 22x18 macroblock).
  - Συνολικά για κάθε frame: 31.638.024
  - Αν θέλουμε 25fps, τότε: 790.950.600 λειτουργίες/sec



# Απαιτήσεις επιταχυντή

---

<b>name</b>	block motion estimator
<b>purpose</b>	block motion est. in PC
<b>inputs</b>	macroblocks, search areas
<b>outputs</b>	motion vectors
<b>functions</b>	compute motion vectors with full search
<b>performance</b>	as fast as possible
<b>manufacturing cost</b>	hundreds of dollars
<b>power</b>	from PC power supply
<b>physical size/weight</b>	PCI card



# Τύποι δεδομένων επιταχυντή, Βασικές τάξεις

---

Motion-vector

x, y : pos

Macroblock

pixels[] : pixelval

Search-area

pixels[] : pixelval

PC

memory[]

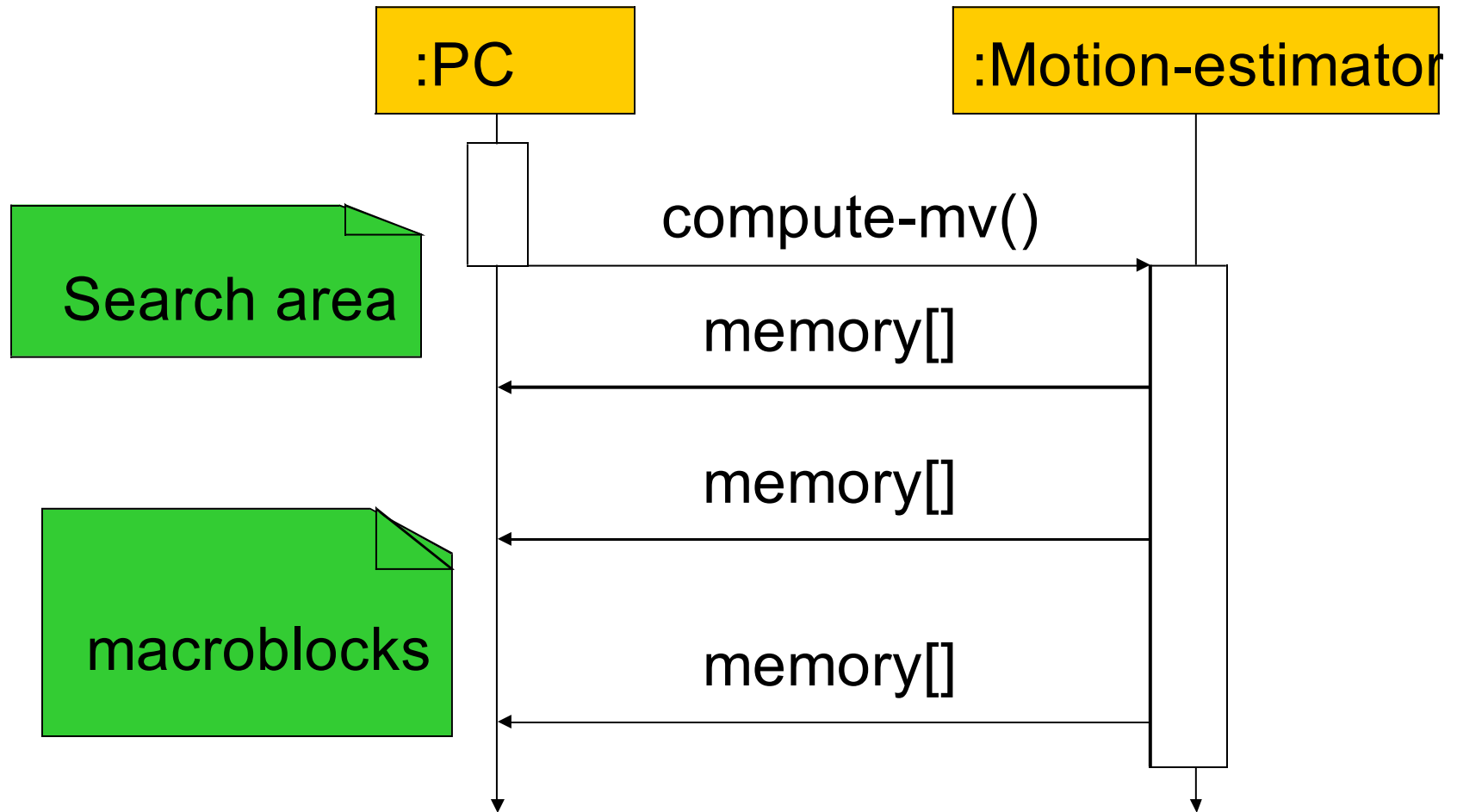
Motion-estimator

compute-mv()





# Σειρά διαγράμματος



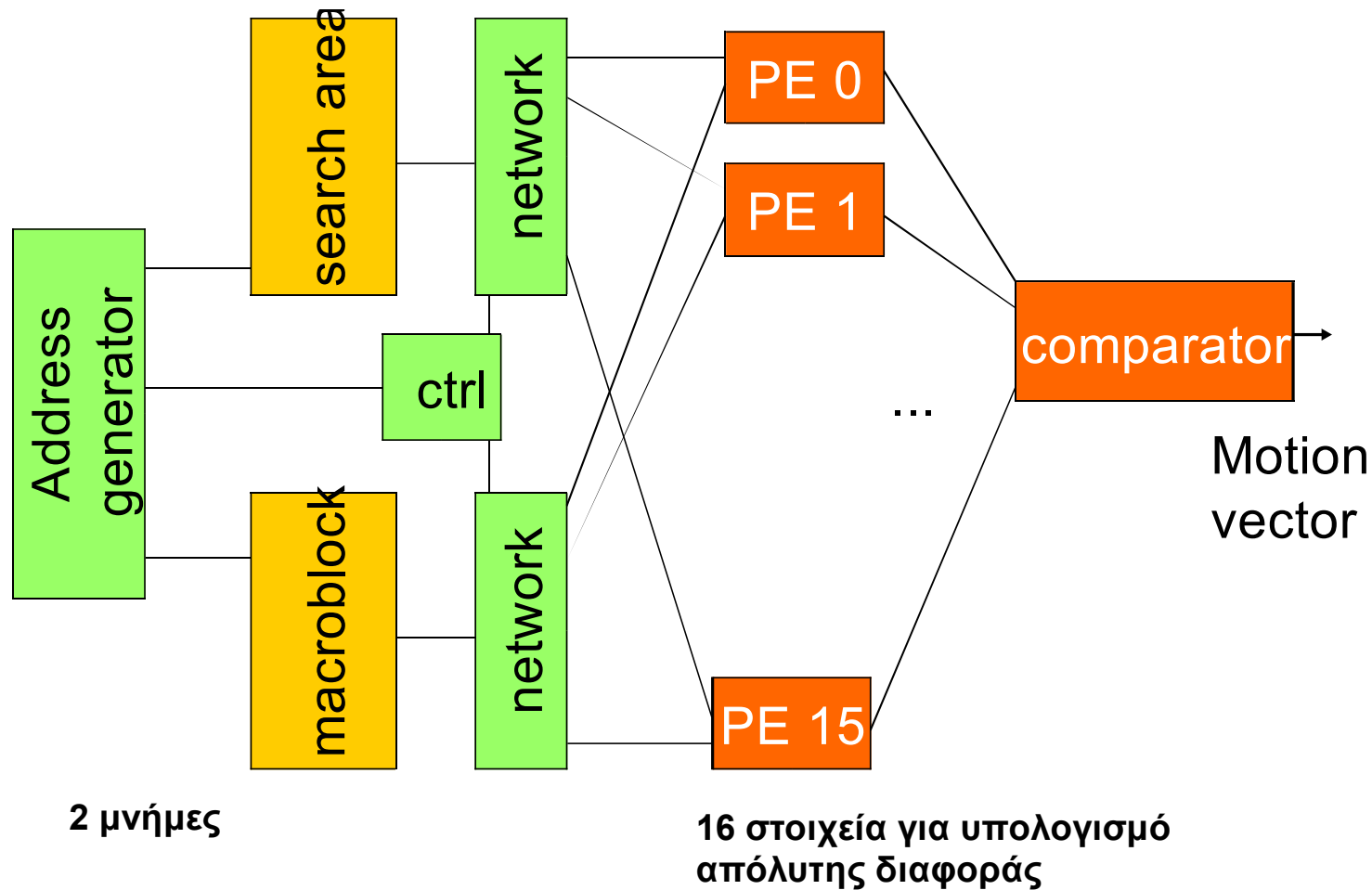
# Αρχιτεκτονικοί υπολογισμοί

---

- Απαιτεί μεγάλο ποσοστό μνήμης:
  - macroblock has  $16 \times 16 = 256$  pixels;
  - περιοχή αναζήτησης έχει  $(8+8+1+8+8)^2 = 1,089$  pixels.
- Δηλαδή, 1089 τιμές των 8bit.
- Ίσως χρειαστεί εξωτερική μνήμη (ιδιαίτερα αν ρυθμίζονται πολλαπλά *macroblocks* /περιοχές έρευνας).



# Οργανισμός κίνησης της ρύθμισης



# Χρονοδιαγράμματα pixel

PE 0

PE 1

PE 2

$|M(0,0)-S(0,0)|$

$M(0,0)$

$|M(0,1)-S(0,1)|$

$|M(0,0)-S(0,1)|$

$|M(0,2)-S(0,2)|$

$|M(0,1)-S(0,2)|$

$|M(0,0)-S(0,2)|$

$S(0,2)$

Χρονοδιάγραμμα (ανά κύκλο ρολογιού):

- ένα pixel από μνήμη μακρομπλοκ
- δυο pixel από τη μνήμη αναζήτησης



# Σύστημα δοκιμής

---

- Η δοκιμή απαιτεί ένα μεγάλο ποσοστό δεδομένων.
- Χρησιμοποίησε απλά σχέδια με προφανείς απαντήσεις για αρχικές δοκιμασίες.
- Αποσπάστε δείγματα δεδομένων από εικόνες JPEG για πιο ρεαλιστικές δοκιμές.



---

# Πολυεπεξεργαστές στα ενσωματωμένα συστήματα



# Πολυεπεξεργαστές

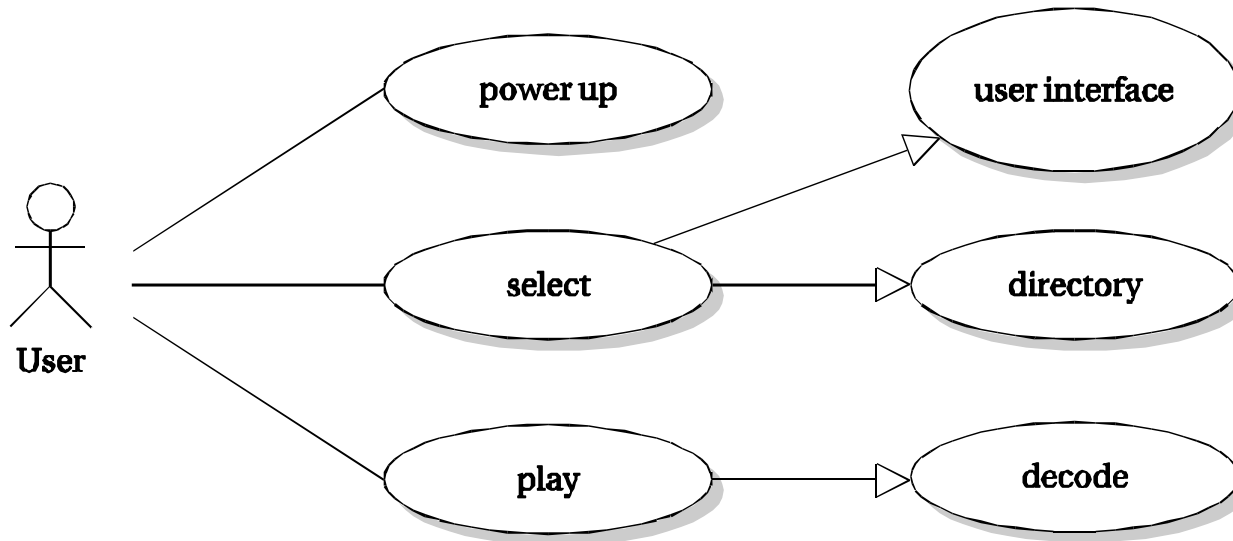
---

- Καταναλωτικά ηλεκτρονικά (**CE**) συστήματα.
- Κινητά τηλέφωνα.
- CDs και DVDs.
- Αναπαραγωγή ήχου.
- Ψηφιακές φωτογραφικές μηχανές.



# Περιπτώσεις χρήσης ηλεκτρονικών ειδών ευρείας κατανάλωσης

- Πολυμέσα: αποθηκεύονται σε συμπιεσμένη μορφή, στη προβολή φαίνεται πως δεν είναι συμπιεσμένα.
- Η αποθήκευση των δεδομένων και διαχείριση τους: παρακολούθηση, κλπ.
- Επικοινωνία: λήψη, αποστολή, chat





# Μη λειτουργικές απαιτήσεις για CE

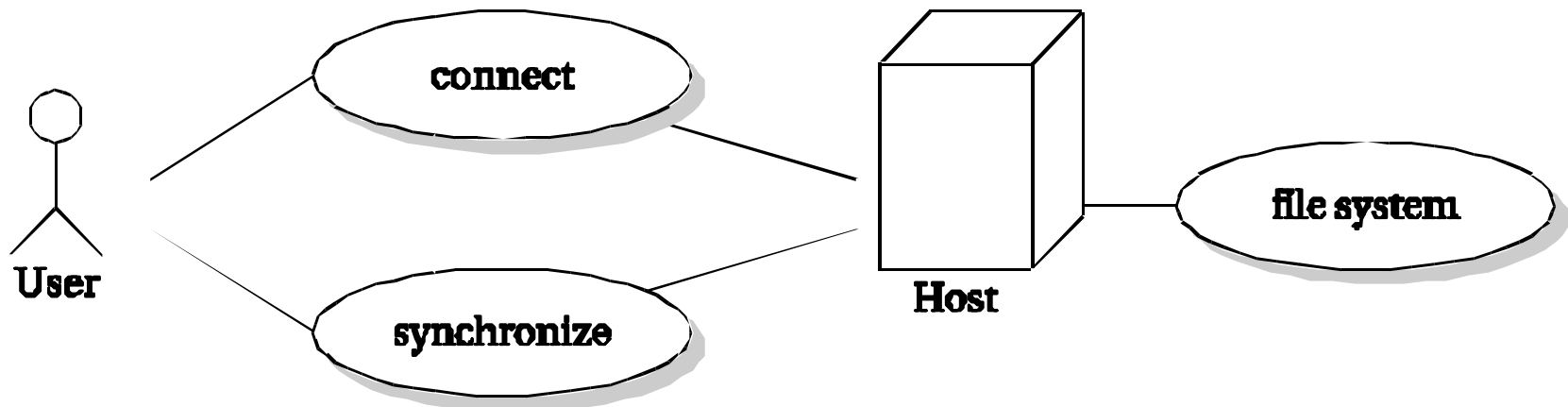
---

- Συχνά λειτουργούν με μπαταρία, αυστηρό προϋπολογισμό ενέργειας.
- Πολύ φθηνά.
- Η διασύνδεση του χρήστη θα πρέπει να υπάρχει, αλλά να είναι χαμηλού κόστους.



# CE συσκευές και ξενιστής (host)

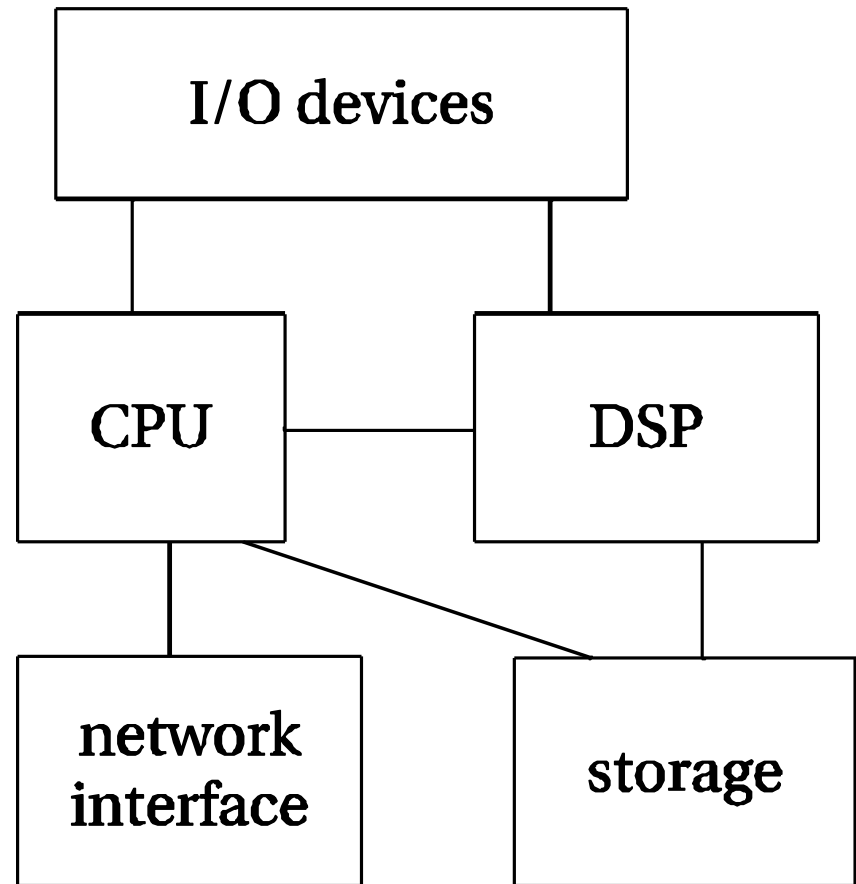
- Πολλές συσκευές επικοινωνούν με συστήματα host.
- Το PC ξενιστής κάνει πράγματα που είναι δύσκολο να το κάνετε με τη συσκευή.
- Όλο και περισσότερες, CE συσκευές επικοινωνούν απευθείας μέσω του δικτύου, αποφεύγοντας το ξενιστή για πρόσβαση.



# Πλατφόρμες και συστήματα λειτουργίας

---

- Πολλές συσκευές CE χρησιμοποιούν DSP για την επεξεργασία σήματος και RISC CPU για άλλες εργασίες.
- I/O περιλαμβάνουν κουμπιά, οθόνη, usb



# Flash συστήματα αρχείων

---

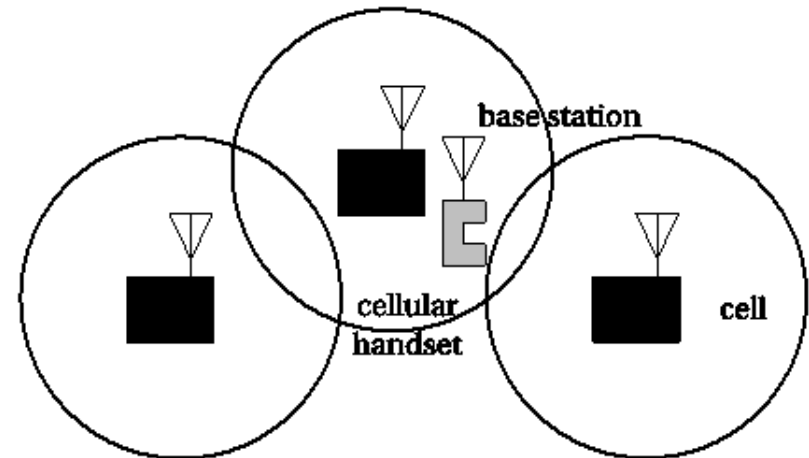
- Τα Flash χρησιμοποιούνται ευρέως για μαζική αποθήκευση
- Τα Flash φθείρονται στο γράψιμο (μέχρι 1 εκατ. κύκλους).
  - Ο φάκελος είναι πιο συχνά γραμμένος, φθείρεται πρώτος.
- Τα Flash συστήματα αρχείων έχουν στρώματα που κινούν περιεχόμενα με ανάλογη φθορά επιπέδου εγγραφής.
  - Κρύβει φθορά που αντιμετωπίζεται αποτελεσματικά από το firmware.



# Κινητά τηλέφωνα

---

- Πιο δημοφιλής συσκευή CE στην ιστορία; Πιο ευρέως χρησιμοποιούμενη υπολογιστική συσκευή.
- 1 δισεκατομμύριο πωλούνται ανά έτος.
- Η συσκευή συνδέεται με την κυψέλη.
- Η κυψέλη φροντίζει για τη μεταφορά της επικοινωνίας σε γειτονική κυψέλη, κατά την μετακίνηση.



# Πλατφόρμες κινητών (1/2)

---

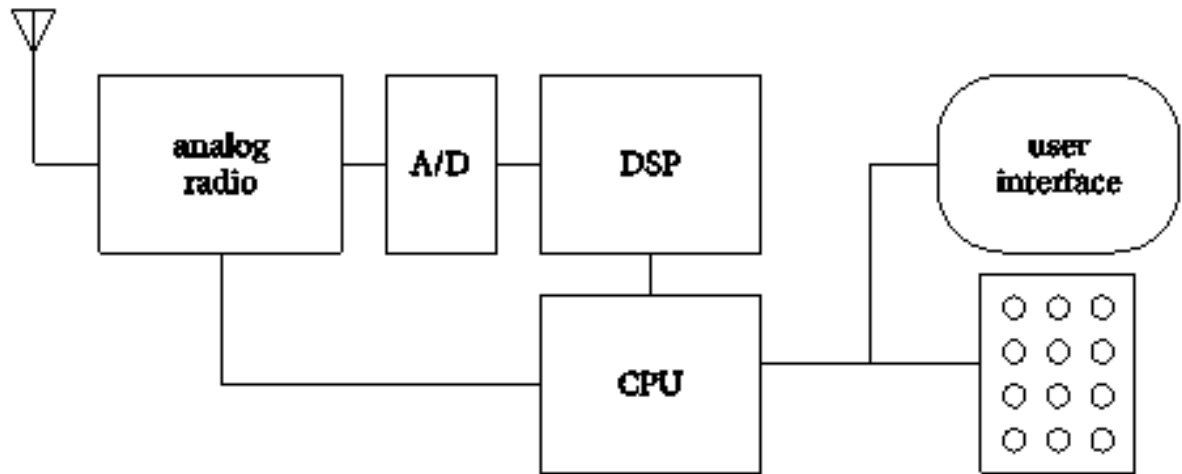
- Τα Κινητά τηλέφωνα σήμερα χρησιμοποιούν αναλογικό front-end, ψηφιακή επεξεργασία βασικής ζώνης.
- Τα Μελλοντικά κινητά τηλέφωνα θα εκτελούν IF επεξεργασία με DSP.
- Επεξεργασία βασικής ζώνης στο DSP:
- Συμπίεσης φωνής.
- Πρωτόκολλο δικτύου.



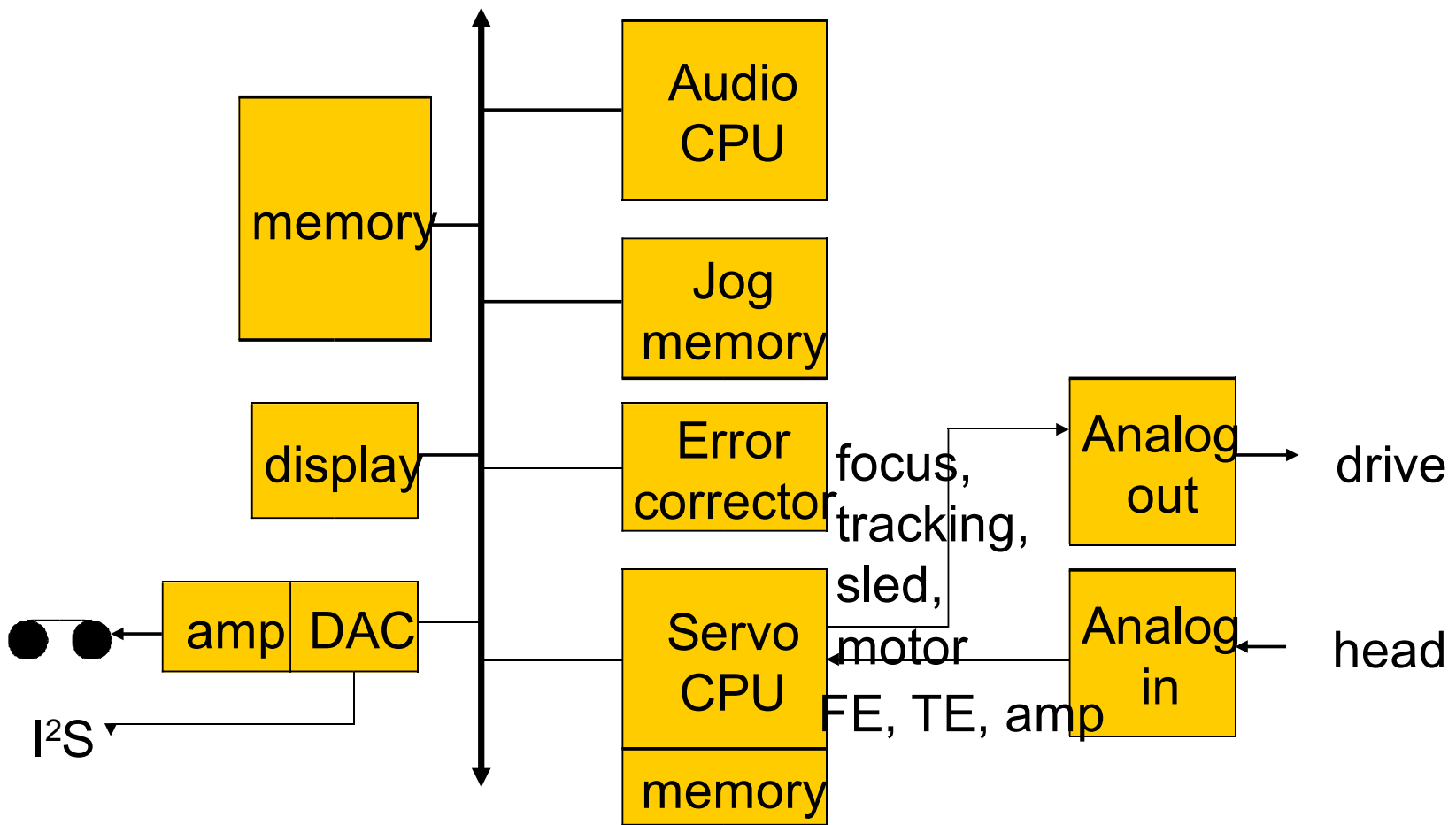
# Πλατφόρμες κινητών (2/2)

---

- Άλλη επεξεργασία:
  - Λειτουργίες πολυμέσων.
  - Διασύνδεση χρήστη.
  - Το σύστημα αρχείων.
  - Εφαρμογές (επαφές, κλπ.)



# CD/MP3 player





# CD medium

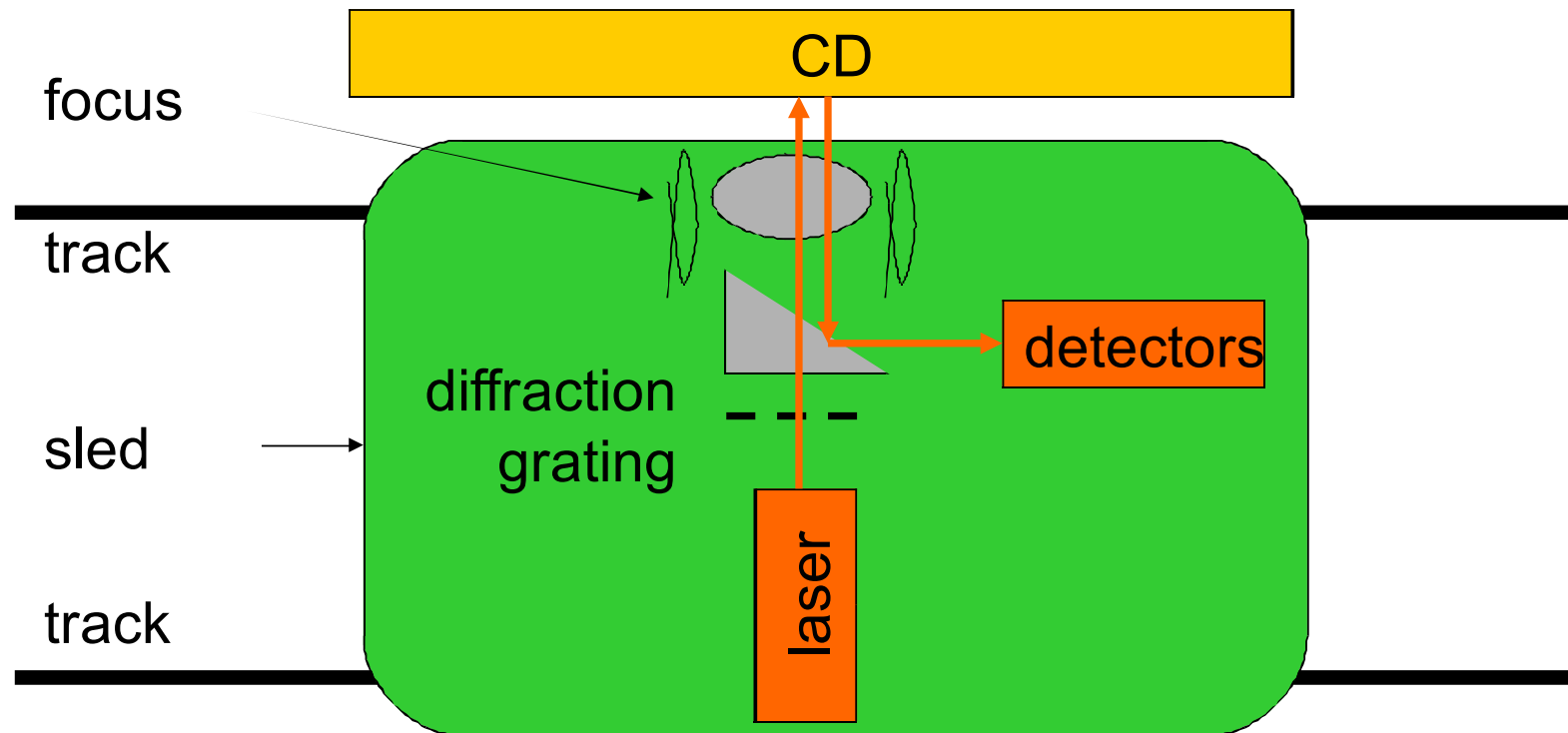
---

- Ταχύτητα περιστροφής: 1,2-1,4 m / s (CLV).
- Track pitch: 1.6 microns.
- Διάμετρος: 120 mm.
- Μήκος Pit: 0,8 -3 microns.
- Βάθος Pit: .11 microns.
- Πλάτος Pit: 0.5 microns.
- Μήκος κύματος Laser: 780 nm.



# Μηχανισμός CD

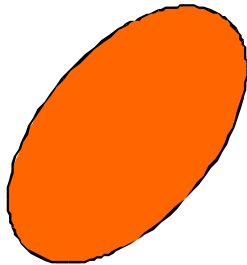
- Laser, lens, sled:



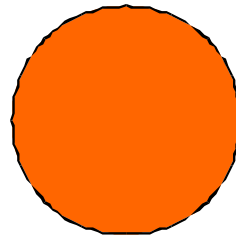
# Laser εστίαση

---

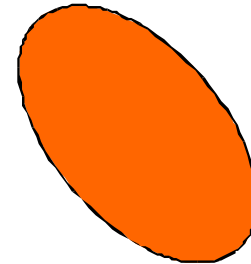
- Η εστίαση ελέγχεται από κάθετη θέση του φακού.
- Αόριστη πορεία προκαλεί ακανόνιστη θέση:



Out of focus



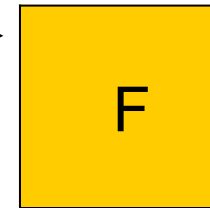
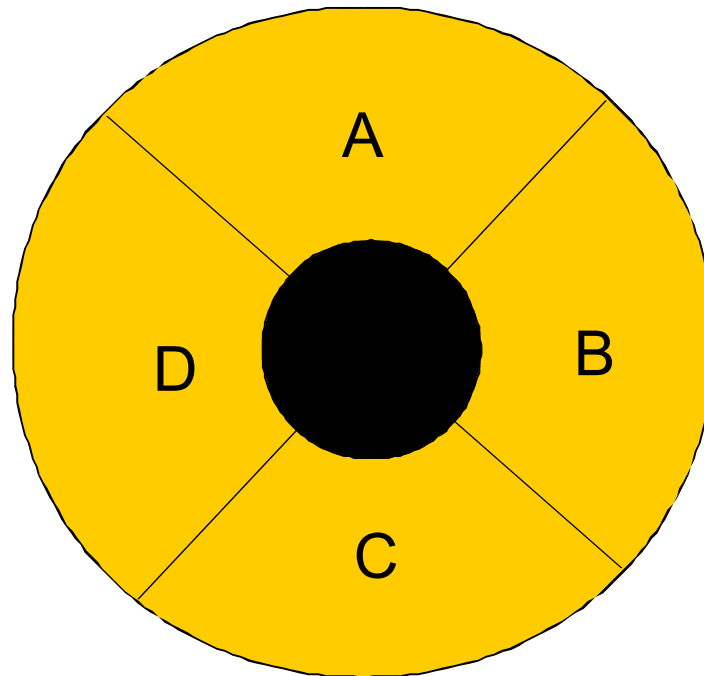
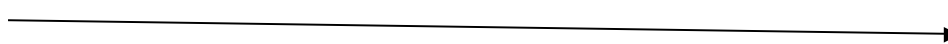
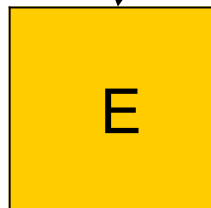
In focus



Out of focus

# Laser pickup

Side spot detectors



Level:

$$A+B+C+D$$

Focus error:

$$(A+C)-(B+D)$$

Tracking error:

$$E-F$$



# Servo control

---

## Τέσσερα είναι τα κύρια σήματα:

- εστίαση(laser) @ 245 kHz; → **Optical pickup**
- παρακολούθηση(laser) @ 245 kHz; ↗
- sled (motor): @ 800 Hz;
- Disc motor.



# EFM

- Eight-to-fourteen διαμόρφωση (κωδικοποίηση δεδομένων):
  - 14-bit κώδικα, εγγυάται τη μέγιστη απόσταση μεταξύ των μεταβάσεων (χρησιμοποιώντας ένα LUT).
  - Δυαδική '1' πάντοτε διαχωρίζεται (τουλάχιστον δύο '0' μεταξύ '1').

**00000011 → 00100100000000**



# Διόρθωση λάθους

- CD χωρητικότητα 6.99 GB σειρά, 700 MB μορφοποιημένο.
- Reed-Solomon Κωδικός:
  - $g(x) = (x-\alpha) (x-\alpha^2) \dots (x-\alpha^{n-k-1}) (x-\alpha^{n-k})$
- Παράγει δεδομένα, διαγράφει bits.
- Ο χρόνος να λυθούν ποικίλα προβλήματα εξαρτάται από το θόρυβο.
- CD ενδιάμεσες Reed-Solomon μπλοκ για να μειωθούν οι επιπτώσεις των μεγάλων κενών στα δεδομένα.



# Έλεγχος και διόρθωση λαθών

---

- Παραλείψεις προκαλούνται από φυσικές ενοχλήσεις.
  - Περιμένετε για την ενόχληση να υποχωρήσει.
  - Επανάληψη.
- Προβλήματα ανάγνωσης που προκαλούνται από το δίσκο/servo.
  - Εντοπισμός σφαλμάτων.
  - Επιλέξτε περιοχή για να ξαναπροσπαθήσετε.
  - Επανάληψη.
  - Αποτυχία και παρεμβολή





# MPEG πρότυπα ήχου

---

- **στρώμα 1:**
  - Χωρίς απώλειες συμπίεση των υποζωνών + προαιρετικό απλό μοντέλο συγκάλυψης.
- **στρώμα 2:**
  - Πιο προηγμένο μοντέλο συγκάλυψης.
- **στρώμα 3:**
  - Πρόσθετη επεξεργασία για τα χαμηλότερα ποσοστά bit.



# MPEG ηχητικές τιμές

---

- Ποσοστά δειγματοληψίας εισόδου:
  - 32, 44.1, 48 kHz.
- Ρυθμός bit εξόδου:
  - 23, 48, 64, 96, 112, 128, 192, 256, 384 kbits/sec.
- Έξοδος μπορεί να είναι μόνο, δύο καναλιών (δίγλωσσο, κλπ.), στερεοφωνικό



# Άλλα πρότυπα

---

- Dolby Digital (AC-3):
  - Χρησιμοποιεί τροποποιημένο διακριτό μετασχηματισμό συνημίτονου.
- ATRAC (*MiniDisc*):
  - Χρησιμοποιεί υποζώνη + τροποποιημένο DCT.
- MPEG-2 AAC.



# MPEG στρώμα 1

---

- 384 samples/block σε όλες τις συχνότητες.
- Ισούται 8 ms σε 48 kHz.
- Προαιρετικό μοντέλο συγκάλυψης.
- Καθοδηγείται από ξεχωριστές FFT για την καλύτερη ακρίβεια.



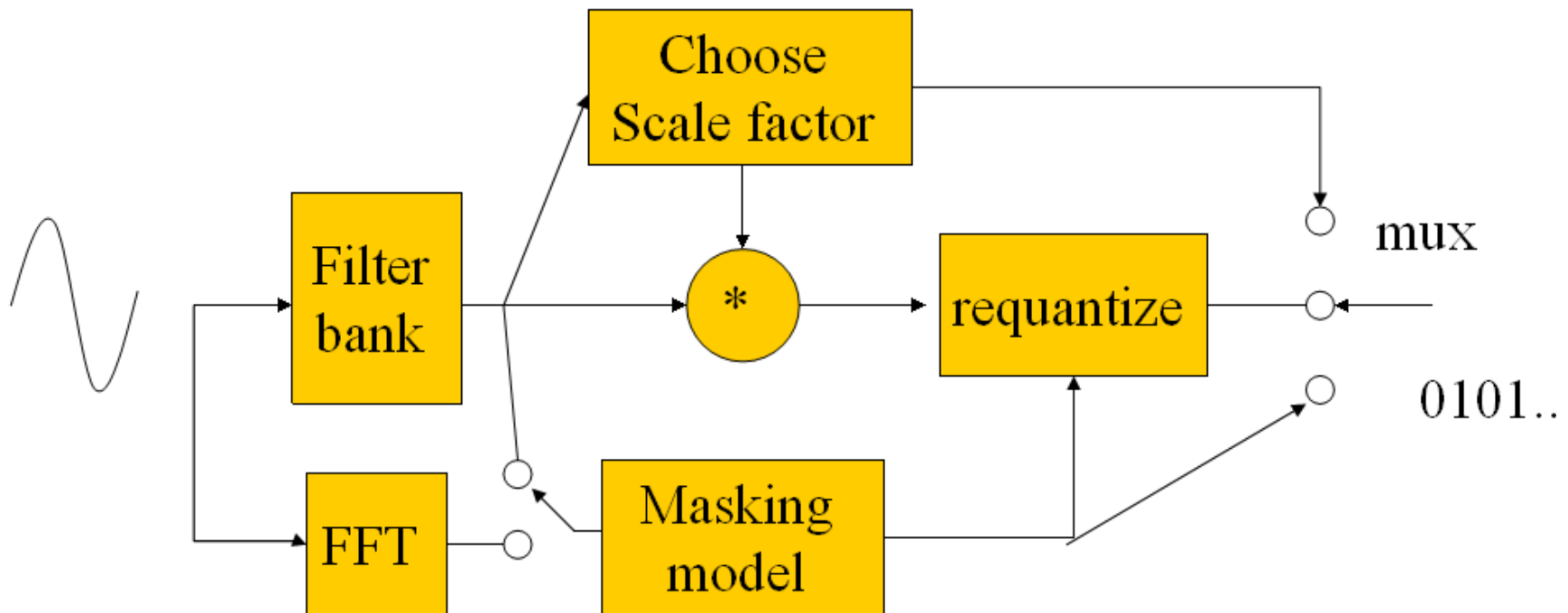
# MPEG στρώμα 1 πλαίσιο δεδομένων

---

- Οι κωδικοί κατανομής bit καθορίζουν το μήκος λέξης σε κάθε υποπεριοχή.
- Οι παράγοντες κλίμακας προσδιορίζουν το κέρδος για κάθε επίπεδο.



# MPEG Layer 1 decoder



# MP3

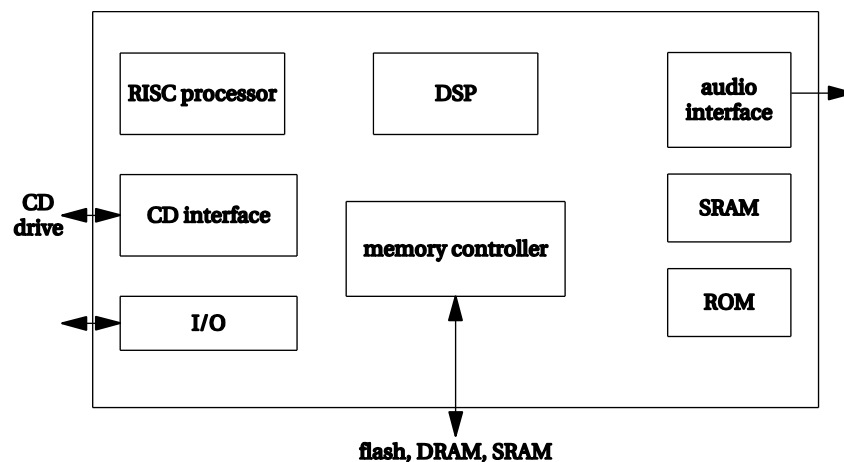
---

- Η Αποκωδικοποίηση είναι ευκολότερη από την κωδικοποίηση αλλά απαιτεί:
- Αποσυμπίεση.
- φιλτράρισμα.
- Βασικό πρότυπο CD για δίσκους δεδομένων.
- Δεν υπάρχουν πρότυπα για MP3 δομή αρχείων του δίσκου: η συσκευή αναπαραγωγής πρέπει να καταλάβει τα Windows, Mac, Unix δίσκους.



# Αναπαραγωγές ήχου

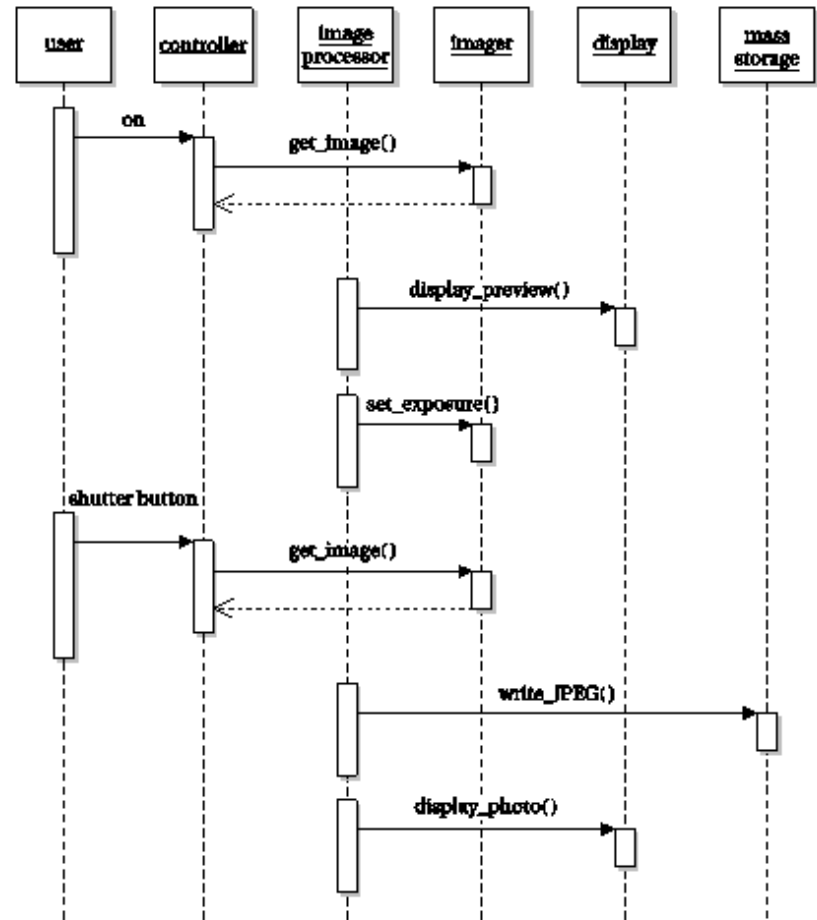
- Οι αναπαραγωγές ήχου μπορούν να χρησιμοποιούν flash, σκληρό δίσκο, CD για μαζική αποθήκευση.
- Η Αποσυμπίεση απαιτεί μικρή ποσότητα CPU:
- 10% των ARM7.
- Το σύστημα αρχείων πρέπει να είναι συμβατό (FAT).





# Ψηφιακές φωτογραφικές μηχανές

- Το DSC πρέπει να καθορίσει την έκθεση πριν από τη λήψη εικόνας.
- Μετά τη λήψη εικόνας:
- Βελτιώστε την ποιότητα της εικόνας.
- Συμπίεση.
- Αποθήκευση ως αρχείο.



# Αρχιτεκτονική ψηφιακής φωτογραφικής μηχανής (1/2)

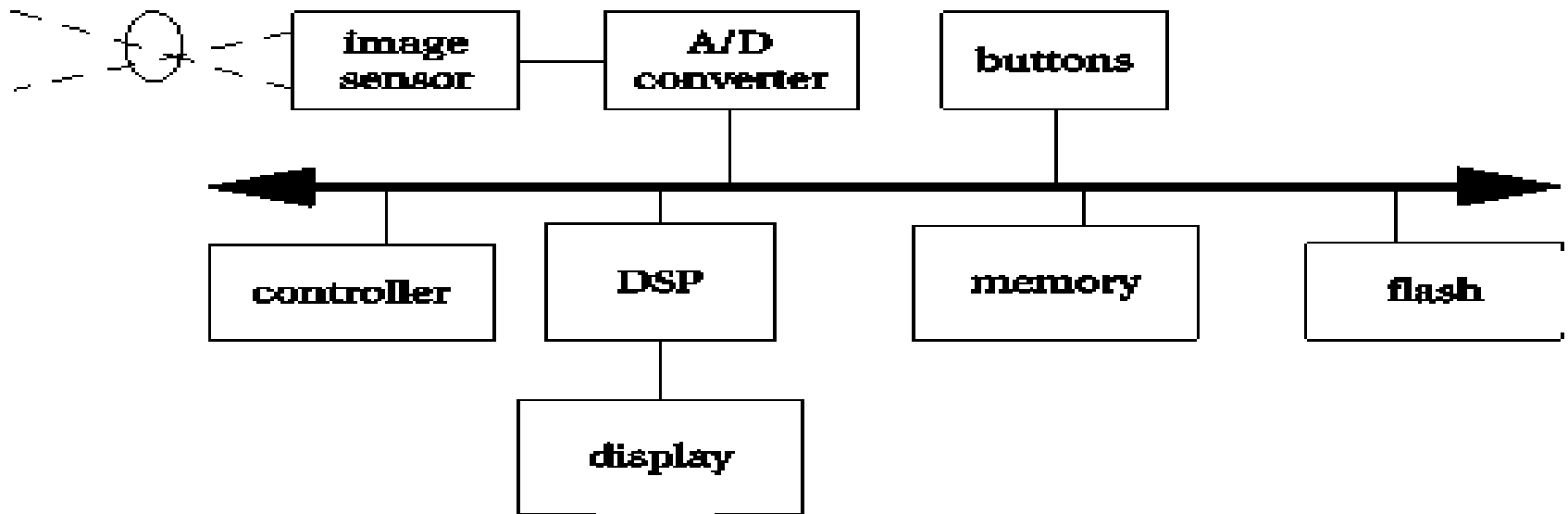
---

- Η DSC χρησιμοποιεί CPU για γενικούς σκοπούς επεξεργασίας και DSP για την επεξεργασία εικόνας.
- Η εσωτερική μνήμη αποθηκεύει προσωρινά τα περάσματα στην εικόνα.
- Η οθόνη είναι χαμηλότερης ανάλυσης από τον αισθητήρα εικόνας.
- Ίσως απαιτηθεί
- downsample.



# Αρχιτεκτονική ψηφιακής φωτογραφικής μηχανής (2/2)

---



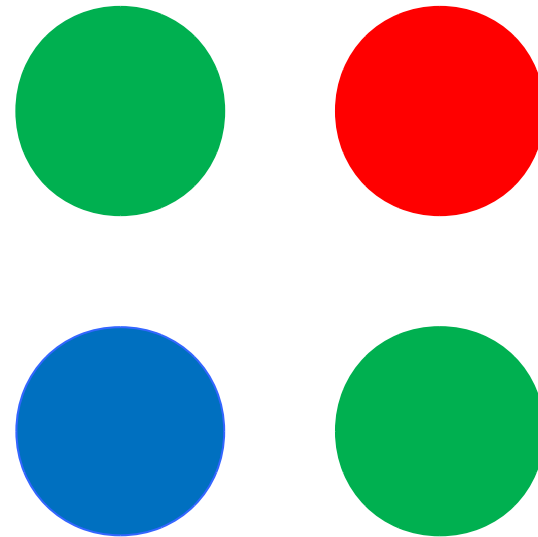
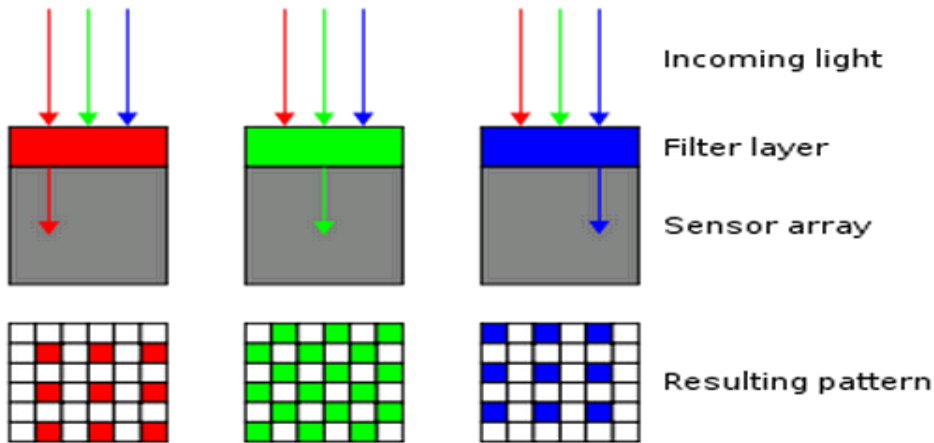
# Καταγραφή εικόνας (1/2)

---

- **Πριν από τη λήψη εικόνας:**
  - Καθορίστε την έκθεση.
  - Καθορίστε την εστίαση.
  - Βελτιστοποίησε την ισορροπία λευκού.



# Καταγραφή εικόνας (2/2)



Bayer pattern

# Επεξεργασία εικόνας

---

Πρέπει να εκτελέσετε βασικές επεξεργασίες για να χρησιμοποιηθεί η εικόνα:

- Bayer → RGB παρεμβολή.

DSCs εκτελεί πολλές λειτουργίες που άλλοτε εκτελούνταν από photoprocessors για την ταινία:

- Καθάρισμα εικόνας.
- Εξισορρόπηση των χρωμάτων.



# Διαχείριση αρχείων

---

- Το EXIF πρότυπο για τις ψηφιακές εικόνες ορίζει:
  - Μορφή των δεδομένων σε ένα αρχείο.
  - Δομή καταλόγου.
- Το EXIF αρχείο περιλαμβάνει:
  - Εικόνα (*JPEG, κλπ*)
  - Μικρογραφία.
  - Μεταδεδομένα  
(*τύπος φωτογραφικής μηχανής, ημερομηνία/ώρα, κλπ.*)



# Βιβλιογραφία

---

Χρησιμοποιήθηκε υλικό από παρουσιάσεις των:

- *Wayne Wolf, Overheads for Computers as Components 2nd ed. ,2008 (chapter 7)*





---

# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

