Ενσωματωμένα Συστήματα

Ενότητα 8: Απόδοση κεντρικής μονάδας επεξεργασίας. Μηχανισμοί Συστημάτων Μνήμης. Κατανάλωση Ενέργειας CPU.

Δρ. Μηνάς Δασυγένης
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.

Σκοπός ενότητας

- Η κατανόηση της σημαντικότητας της κρυφής μνήμης στα ενσωματωμένα συστήματα.
- Η κατανόηση των πλεονεκτημάτων και των μειονεκτημάτων των μονάδων διαχείρισης μνήμης και προστασίας μνήμης.
- Η παρουσίαση της διασωλήνωσης στα ενσωματωμένα συστήματα.

Περίγραμμα διάλεξης

- Κρυφές Μνήμες.
- Μονάδες διαχείρισης μνήμης και μετάφραση διεύθυνσης.
- Απόδοση CPU.
- Διοχέτευση.
- Υπερβαθμωτή Εκτέλεση.
- Κατανάλωση ενέργειας CPU.

Μνήμη & επεξεργαστές

- Δεν υπάρχει πια μόνο μια μονολιθική μνήμη.
- Υπάρχουν πολλά επίπεδα μνήμης *(ιεραρχία μνήμης).*
- Αυτό οφείλεται:
  - Η απόδοση των επεξεργαστών αυξάνεται με πολύ γρηγορότερο ρυθμό από την απόδοση της μνήμης.
  - Η πρόσβαση στην off-chip μνήμη είναι ακριβή *(ενέργεια & χρόνος).*
- Στην ιεραρχία μνήμης, όσο πιο κοντά βρίσκεται ένα επίπεδο στο CPU, τόσο πιο μικρό σε μέγεθος, πιο μεγάλο κόστος $ /bit, πιο μικρό κόστος πρόσβασης, πιο μικρό κόστος ενέργειας.

Κρυφή μνήμη & MMU

- Οι σύγχρονοι επεξεργαστές έχουν κρυφή μνήμη σε κάποια επίπεδα, ώστε να κερδίσουν σε ενέργεια και χρόνο.
- Επίσης, αν η μνήμη είναι μεγάλη σε μέγεθος, ή απαιτείται ιδιαίτερος έλεγχος *(π.χ. προστασία),* τότε χρησιμοποιείται η MMU για τη μετάφραση των σχετικών διευθύνσεων του επεξεργαστή, σε ένα μεγαλύτερο σύνολο από απόλυτες διευθύνσεις της φυσικής μνήμης.

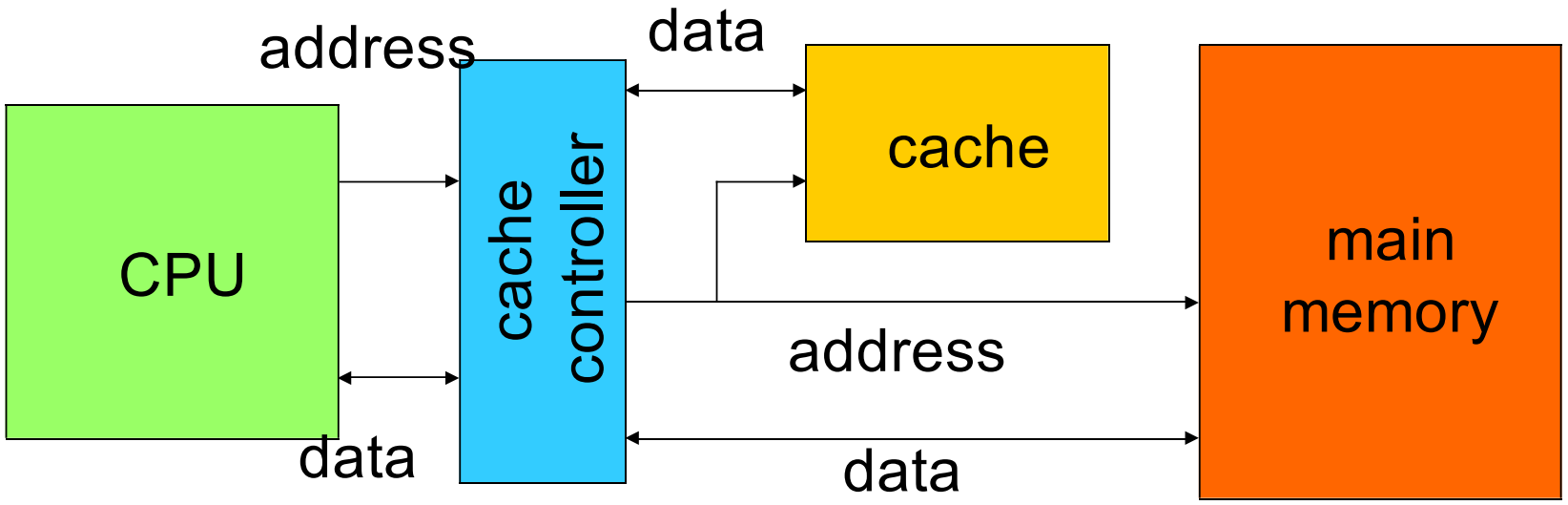
Γενικά για την κρυφή μνήμη

- Η cache επιταχύνει το μέσο χρόνο προσπέλασης της μνήμης όταν χρησιμοποιηθεί κατάλληλα.
- Η cache είναι μνήμη SRAM εντός του IC.
- Έχει μεγάλο κόστος ανά bit
  (6T ανά bit + κυκλώματα σύγκρισης, ενώ η DRAM έχει 1T/bit).
- Αυξάνει τη μεταβλητότητα των χρόνων προσπέλασης της μνήμης.
  - Πιο γρήγορη πρόσβαση σε δ/νσεις μνήμης που βρίσκονται στη cache *(ως προς ανυπαρξία cache).*
  - Πιο αργή πρόσβαση σε δ/νσεις μνήμης που δε βρίσκονται στη cache *(ως προς ανυπαρξία cache).*
- Σε συγκεκριμένες περιπτώσεις *(mission critical)* αποφεύγεται η cache, εξαιτίας της μεταβλητότητας.

Η κρυφή μνήμη

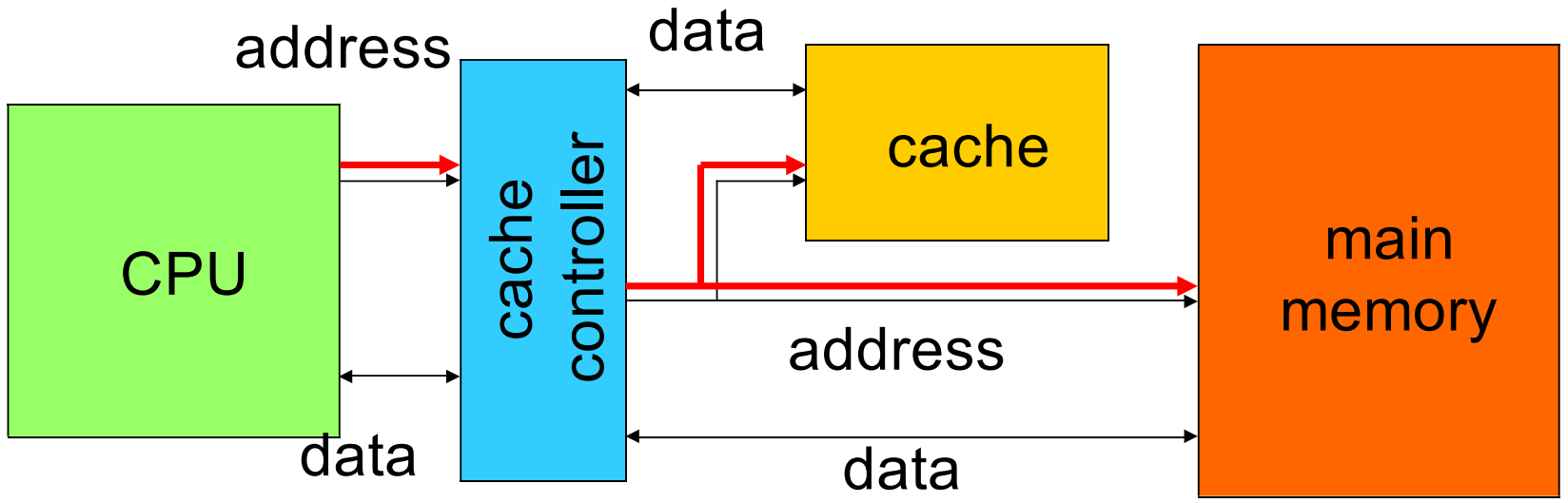
- Μικρή.
- Γρήγορη.
- Κρατά αντίγραφα από κάποια περιεχόμενα της off-chip.
- Κρατάει αντίγραφα για περιορισμένα δεδομένα.
- Η χρήση της cache έχει οφέλη, αν η CPU χρησιμοποιεί ένα σχετικά μικρό σύνολο θέσεων μνήμης *(ονομάζεται working set).*
- Η αρχιτεκτονική της cache, περιέχει:
- Την κρυφή μνήμη.
- Τον ελεγκτή κρυφής μνήμης.

Caches and CPUs (1/4)



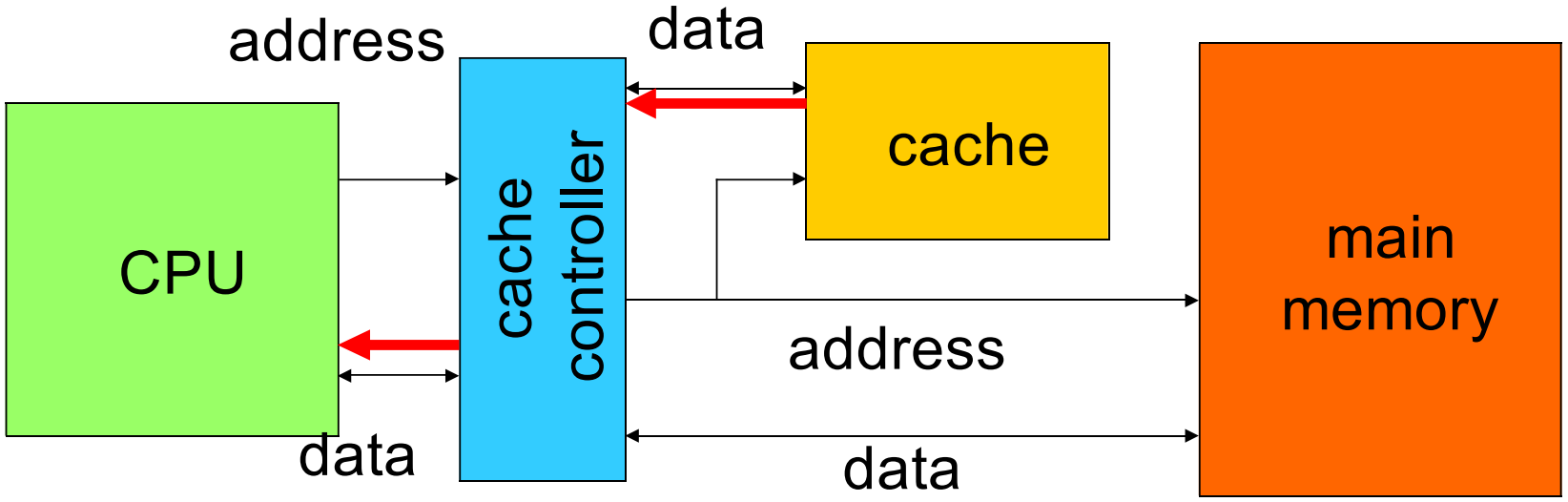
Caches and CPUs (2/4)

- Ο ελεγκτής προωθεί την αίτηση πρόσβασης στην κρυφή μνήμη *(και σε κάποιες αρχιτεκτονικές στην εξωτερική μνήμη).*



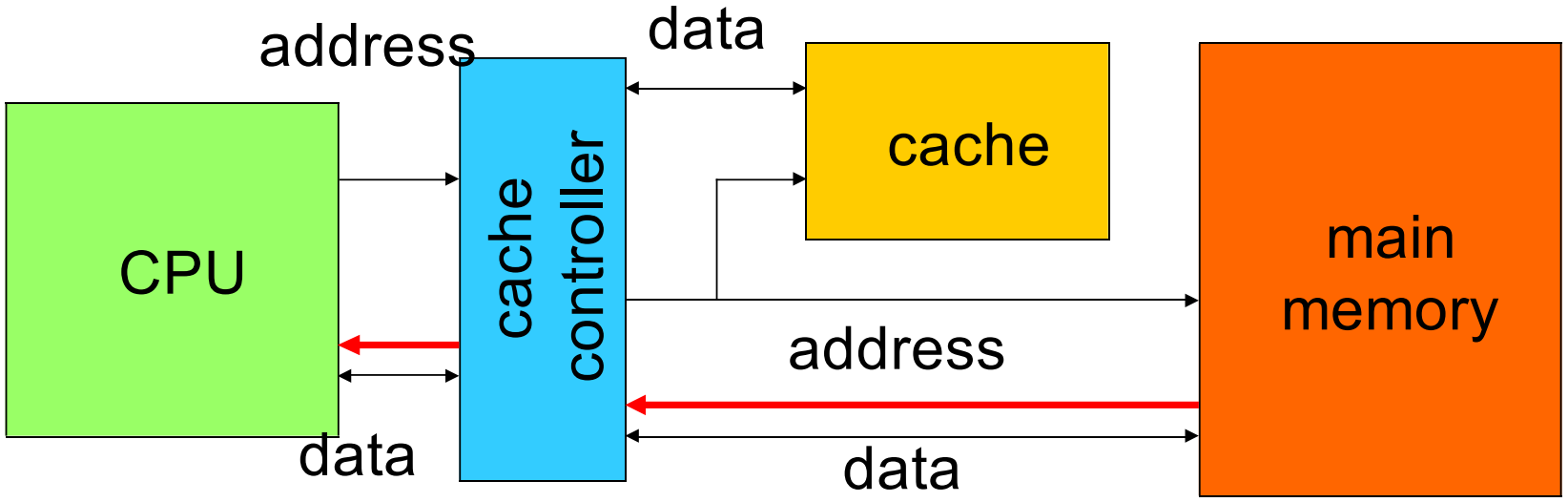
Caches and CPUs (3/4)

- Αν η διεύθυνση βρίσκεται στη cache, *“ευστοχία”*



Caches and CPUs (4/4)

- Αν η διεύθυνση δε βρίσκεται στη cache, *"αστοχία"*



Λειτουργία της cache

- Πολλαπλές δ/νσεις μνήμης απεικονίζονται σε μια γραμμή στη cache.
- Υπάρχουν caches για:
  - instructions;
  - data;
  - data + instructions (**unified**).
- Ο χρόνος πρόσβασης στη μνήμη έχει μεγάλη μεταβλητότητα.
- Η cache έχει τα βέλτιστα οφέλη όταν κάθε χρονική περίοδο το πρόγραμμα χρησιμοποιεί ένα σχετικά μικρό σύνολο θέσεων μνήμης *(working set)* που χωράει στη cache.

Όροι που συνδέονται με την κρυφή μνήμη

- **Cache hit** *(ευστοχία)*: η δ/νση που απαιτείται βρίσκεται στην κρυφή μνήμη.
- **Cache miss** *(αστοχία)*: η δ/νση που απαιτείται δε βρίσκεται στην κρυφή μνήμη.
- **Working set** *(σύνολο εργασίας)*: η ομάδα δ/νσεων που χρησιμοποιεί ένα πρόγραμμα σε ένα συγκεκριμένο χρονικό διάστημα.

Είδη αστοχιών

- **Αναγκαστική - Compulsory** *(cold)*:
  η δ/νση δεν έχει προσπελαθεί ποτέ
  (πρώτη φορά μόνο).
- **Χωρητικότητας - Capacity:** υπερβολικά μεγάλο σύνολο εργασίας.
- **Σύγκρουσης - Conflict:** πολλαπλές θέσεις απεικονίζονται στην ίδια θέση μνήμης.

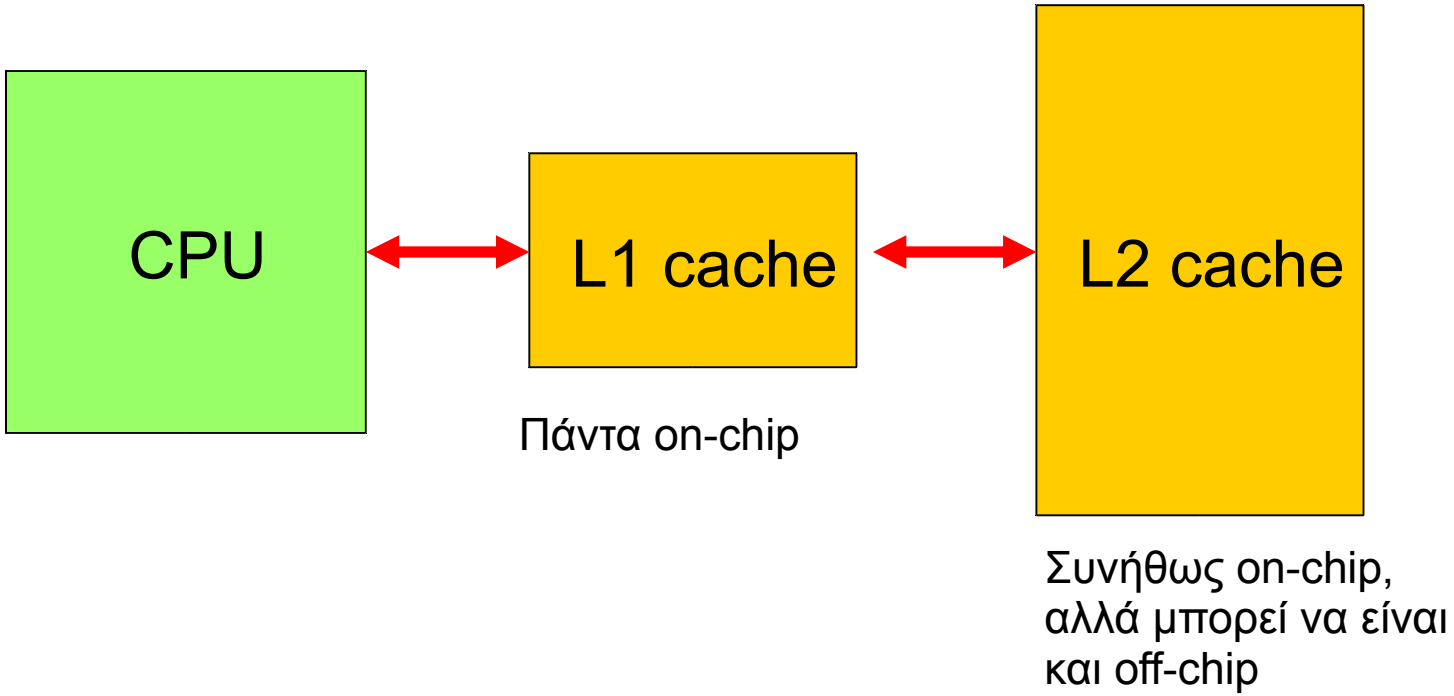
Απόδοση συστήματος μνήμης

- h = ρυθμός ευστοχίας
  (εξαρτάται από το πρόγραμμα, μπορεί να εκτιμηθεί).
- t_{cache}= χρόνος πρόσβασης cache *(1-5ns)*
- t_{main}= χρόνος πρόσβασης κυρίως μνήμης *(50-60ns).*
- Μέσο χρόνο προσπέλασης μνήμης:
  - $t_{av} = ht_{cache} + (1-h)t_{main}$

T_{main} 50-60ns, ενώ 2-3 ns το t_{cache}

Πολλαπλά επίπεδα κρυφής μνήμης

- Χρησιμοποιείται ιεραρχία μνήμης.
- L1 μικρή + πιο γρήγορη, L2 μεγαλύτερη + πιο αργή.



Χρόνος πρόσβασης κρυφής μνήμης πολλαπλών επιπέδων

- h_1 = ρυθμός ευστοχίας L1.
- h_2 = ρυθμός ευστοχίας L2.
- Μέσος χρόνος πρόσβασης μνήμης:
  - $t_{av} = h_1 t_{L1} + (h_2 - h_1) t_{L2} + (1 - h_2 - h_1) tmain$

Οργάνωση κρυφής μνήμης

- **Άμεσα απεικονιζόμενη** *(Direct-mapped)*: κάθε διεύθυνση μνήμης απεικονίζεται σε μια μόνο γραμμή στην κρυφή μνήμη.
- **Συνειρμικότητα συνόλου** *(N-way set-associative)*: κάθε διεύθυνση μνήμης απεικονίζεται σε N γραμμές στην κρυφή μνήμη.
- **Πλήρης συσχετιστική**: κάθε διεύθυνση μνήμης απεικονίζεται σε όλες τις γραμμές στην κρυφή μνήμη *(δεν υλοποιείται συνήθως).*

Πολιτικές αντικατάστασης

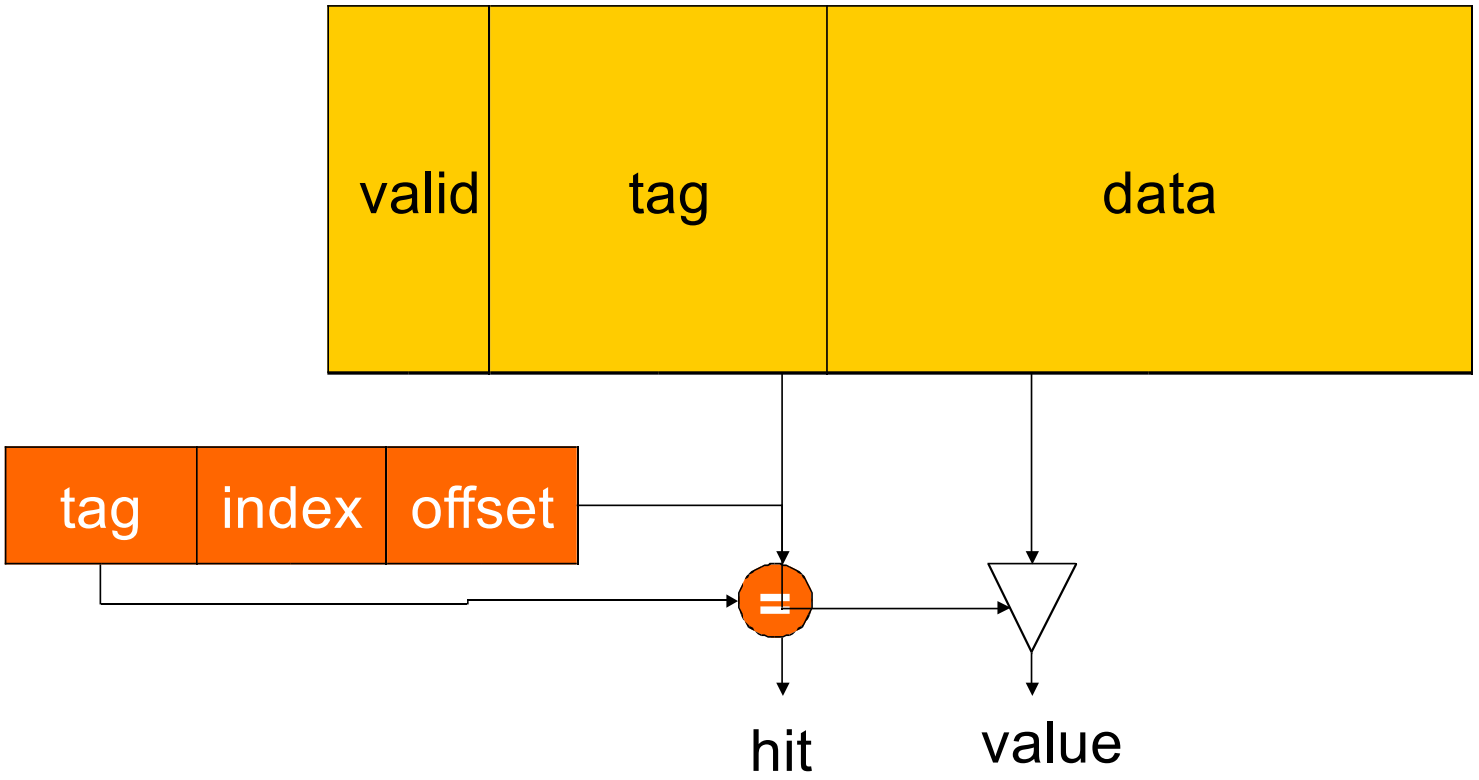
- **Πολιτική Αντικατάστασης** *(Replacement policy)*: η στρατηγική για την επιλογή της γραμμής που θα εκδιωχθεί για να δημιουργηθεί χώρος για μια νέα εγγραφή στη cache.
- Δεν υπάρχει αυτό το πρόβλημα στις κρυφές μνήμες απευθείας απεικόνισης.
- Δυο δημοφιλείς στρατηγικές:
  - Τυχαία.
  - Λιγότερο πρόσφατα χρησιμοποιούμενου *(LRU).*

Πλεονεκτήματα χρήσης κρυφής μνήμης

- Διατήρηση των προσφάτων προσβάσεων στη cache.
- Η κρυφή μνήμη μπορεί να μεταφέρει περισσότερες από μια λέξεις.
  - Οι ακολουθιακές προσβάσεις είναι πιο γρήγορες μετά την πρώτη πρόσβαση.

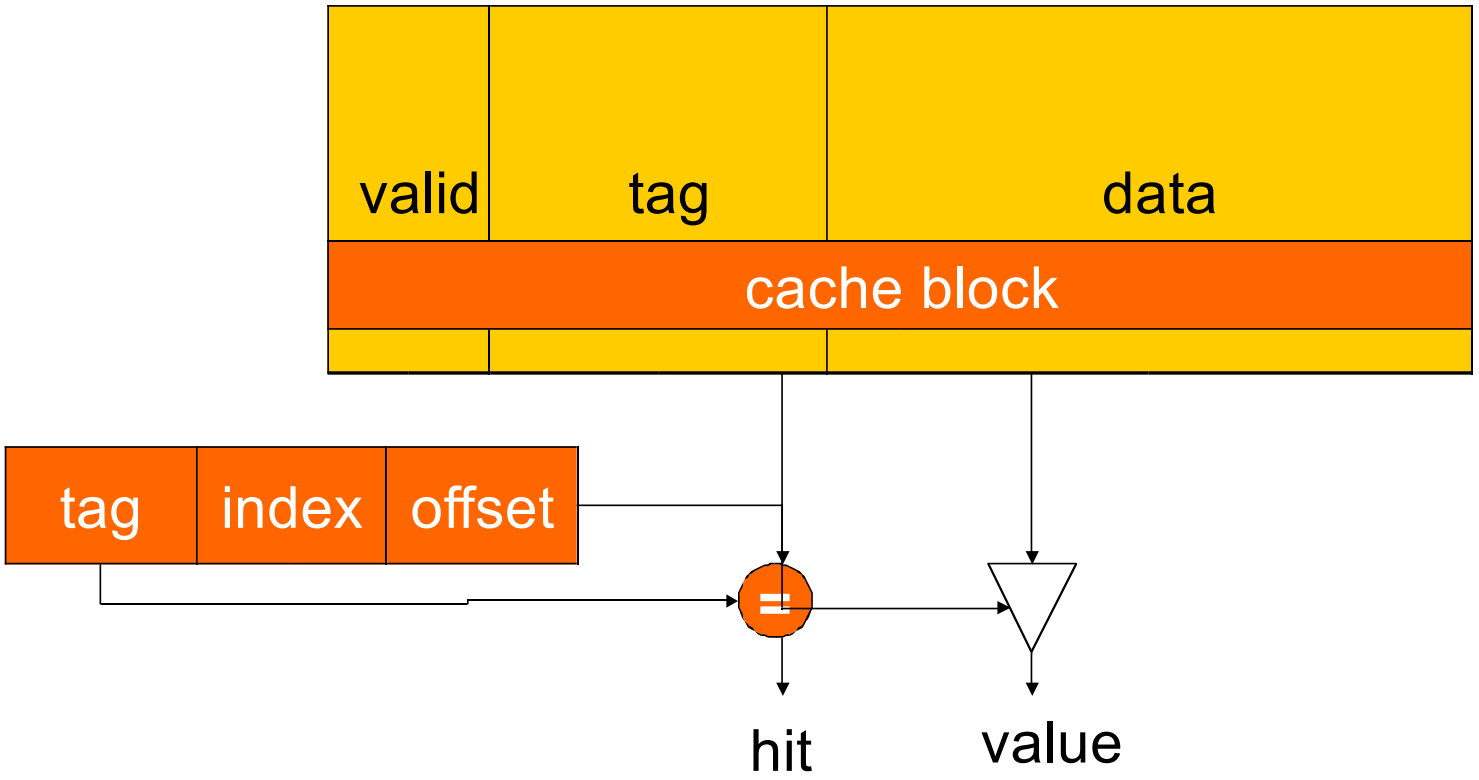
Direct-mapped cache (1/3)

- Κάθε γραμμή της κρυφής μνήμης έχει:



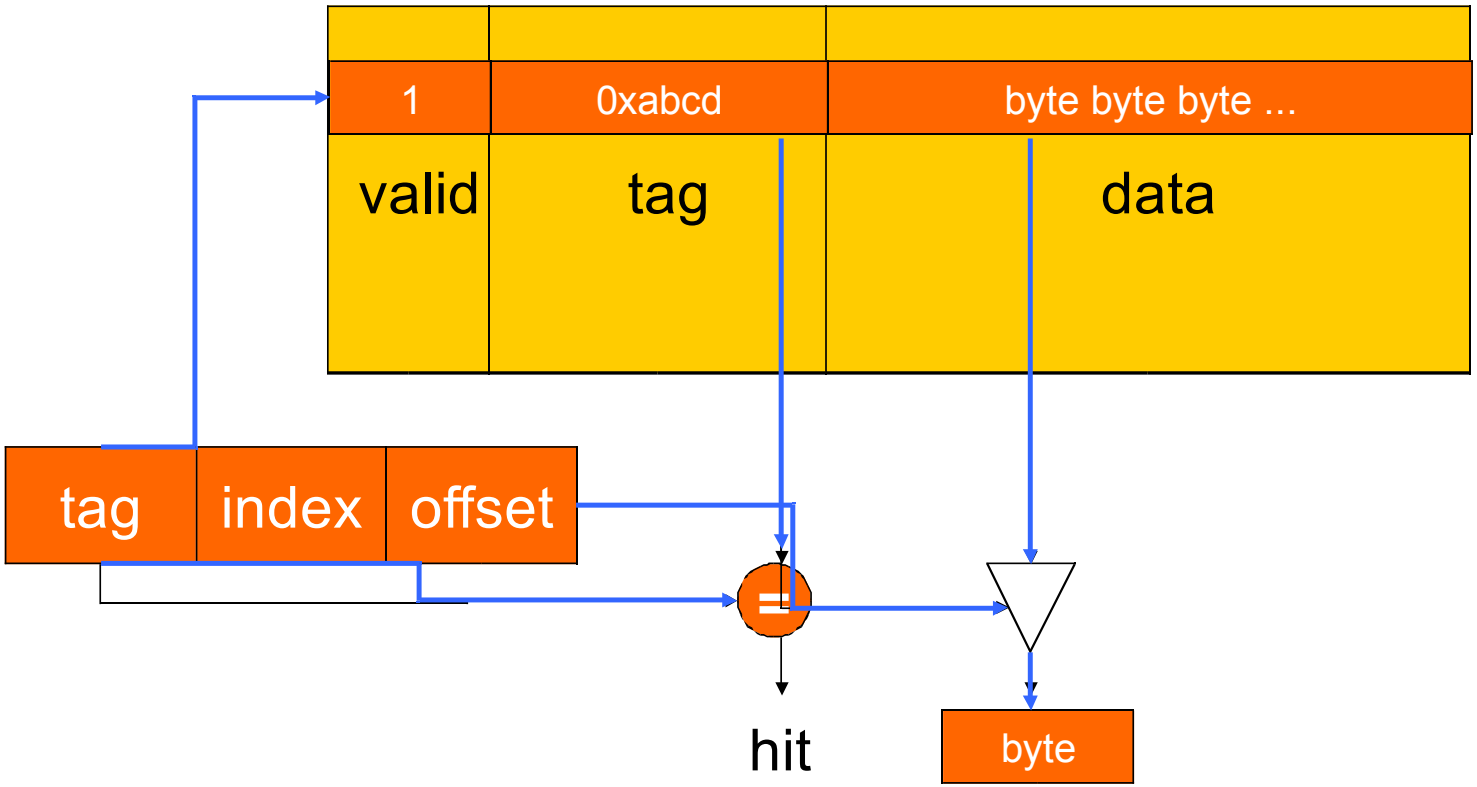
Direct-mapped cache (2/3)

- Υπάρχει μόνο μια γραμμή που ελέγχεται.



Direct-mapped cache (3/3)

- Αν στο data αποθηκεύονται πολλές λέξεις, τότε τα χαμηλότερα bit της διεύθυνσης χρησιμοποιούνται ως σχετική απόσταση.



Λειτουργίες Εγγραφής

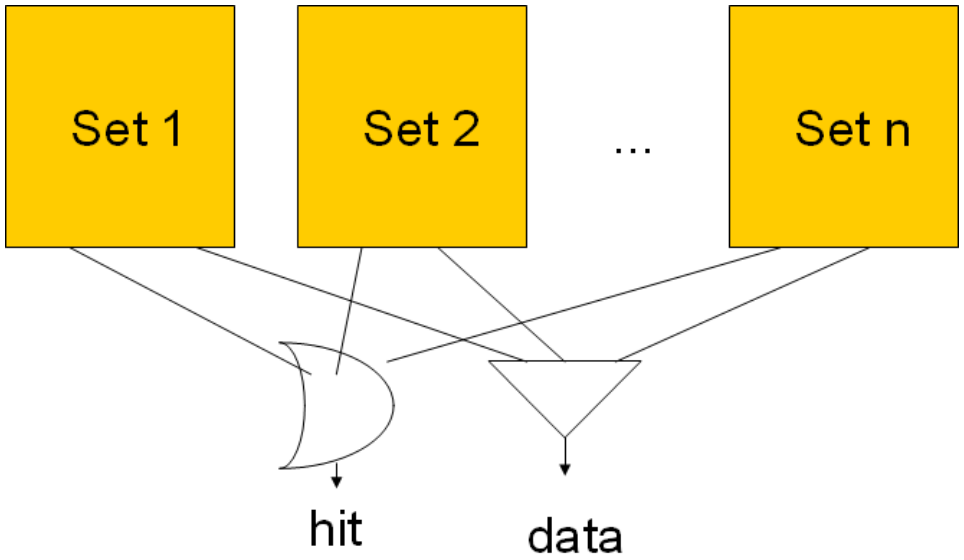
- **Άμεσης εγγραφής** *(Write-through):*
  κάθε εγγραφή αλλάζει τόσο την κρυφή μνήμη, όσο και την αντίστοιχη θέση της κύριας μνήμης.
  - πάντα συνεπείς οι μνήμες, αλλά επιπλέον κυκλοφορία στο δίαυλο.
- **Ετερόχρονης εγγραφής** *(Write-back):*
  η εγγραφή στην κύρια μνήμη γίνεται μόνο αν απομακρυνθεί η αντίστοιχη γραμμή.
- Αναλόγως το πρόγραμμα και τις απαιτήσεις πλεονεκτεί είτε το πρώτο σχήμα, είτε το δεύτερο.

Η κρυφή μνήμη άμεσης απεικόνισης.

- Πολλαπλές δ/νσεις απεικονίζονται στην ίδια γραμμή.
- Γρήγορη, χαμηλού κόστους, μικρότερο hit rate, ενδεχόμενα conflicts.
- Πολύ εύκολα δημιουργούνται συγκρούσεις:
  - 1024 cache lines
  - Array a[] uses locations 0, 1, 2, …
  - Array b[] uses locations 1024, 1025, 1026, …
  - Operation a[i] + b[i] generates conflict misses.

Η κρυφή μνήμη συνειρμικότητας

- Αντιμετώπιση των conflict misses.
- Υπάρχουν n διαφορετικά μπλοκ για κάθε σύνολο θέσεων, υψηλότερο ρυθμό ευστοχίας.
- Υλοποιείται ως μια ομάδα κρυφών μνημών άμεσης απεικόνισης:



Η κρυφή μνήμη συνειρμικότητας: πλεονεκτήματα/μειονεκτήματα

- Επιπρόσθετη επιβάρυνση, λόγω πολλαπλών συγκρίσεων.
- Μεγαλύτερο κόστος πρόσβασης, λόγω πολλαπλών προσβάσεων.
- Υψηλότερος ρυθμός ευστοχίας.
- Πιο δύσκολη η ανάλυση της απόδοσης.
- Πιο δύσκολη η προβλεψιμότητα.

Παράδειγμα (1/6)

Μνήμες άμεσης απεικόνισης προς κρυφές μνήμες συνειρμικότητας συνόλου

- Υποθέτουμε ότι: 1 bit διεύθυνσης tag, 2 MSB bit διεύθυνσης για καθορισμό γραμμής

address	data
000	0101
001	1111
010	0000
011	0110
100	1000
101	0001
110	1010
111	0100

Παράδειγμα (2/6) - Άμεση απεικόνιση

- **After 001 access:**

block	tag	data
00	-	-
01	0	1111
10	-	-
11	-	-

- **After 010 access:**

block	tag	data
00	-	-
01	0	1111
10	0	0000
11	-	-

Παράδειγμα (3/6) - Άμεση απεικόνιση

- **After 011 access:**

block	tag	data
00	-	-
01	0	1111
10	0	0000
11	0	0110

- **After 100 access:**

block	tag	data
00	1	1000
01	0	1111
10	0	0000
11	0	0110

Παράδειγμα (4/6) - Άμεση απεικόνιση

- **After 101 access:**

block	tag	data
00	1	1000
01	1	0001
10	0	0000
11	0	0110

- **After 111 access:**

block	tag	data
00	1	1000
01	1	0001
10	0	0000
11	1	0100

Παράδειγμα (5/6) - Κρυφή μνήμη 2 δρόμων

- **Τελική εικόνα**
  (αν υπάρχει διπλάσιο μέγεθος κρυφής μνήμης, πολιτική LRU):

set	blk 0 tag	blk 0 data	blk 1 tag	blk 1 data
00	1	1000	-	-
01	0	1111	1	0001
10	0	0000	-	-
11	0	0110	1	0100

Παράδειγμα (6/6) - Κρυφή μνήμη 2 δρόμων

- **Τελική εικόνα**
  (αν υπάρχει ίδιο μέγεθος κρυφής μνήμης):

set	blk 0 tag	blk 0 data	blk 1 tag	blk 1 data
0	01	0000	10	1000
1	10	0111	11	0100

Ενοποιημένη κρυφή μνήμη

- Η CPU γνωρίζει πότε προσκομίζει μια εντολή ή δεδομένα *(λόγω ο μετρητής προγράμματος).*
- Μπορούν να αποθηκευτούν στην κρυφή μνήμη, είτε δεδομένα *(κρυφή μνήμη δεδομένων)*, είτε εντολές *(κρυφή μνήμη εντολών)*, είτε και τα δυο *(ενοποιημένη κρυφή μνήμη, unified cache).*
- Σε ενοποιημένη κρυφή μνήμη το ποσοστό του χώρου που κατανέμεται σε εντολές ή δεδομένα, μπορεί να είναι είτε σταθερό είτε μεταβλητό.
- Οι εντολές έχουν καλύτερο ποσοστό ευστοχίας.

Παραδείγματα κρυφών μνημών σε ενσωματωμένα

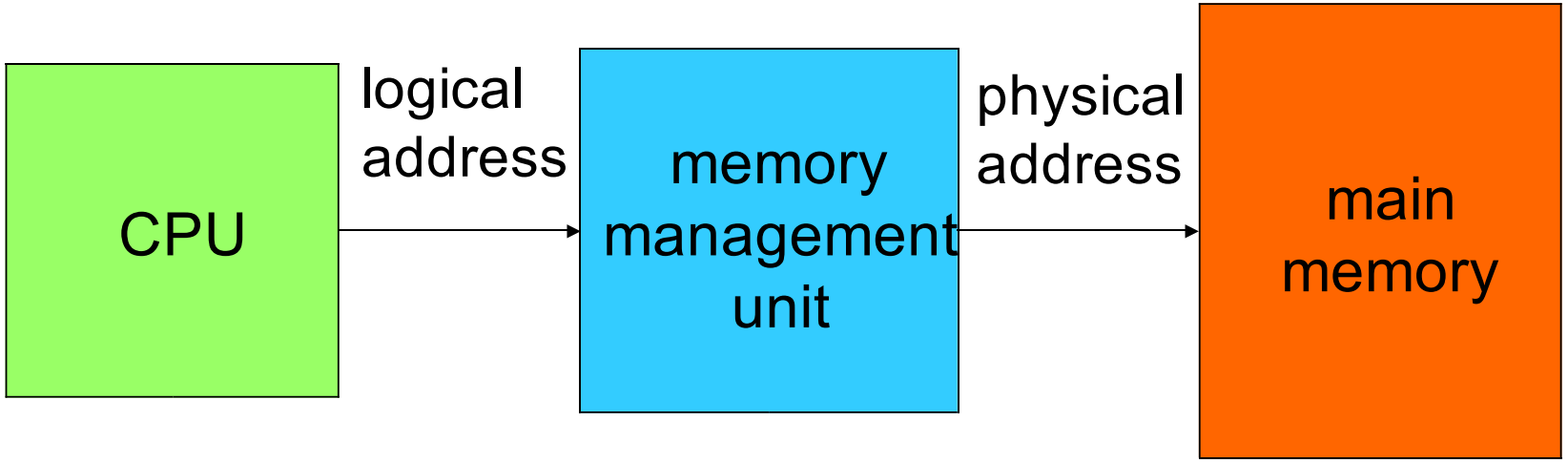
- StrongARM:
  - 16 Kbyte, 32-way, 32-byte block instruction cache.
  - 16 Kbyte, 32-way, 32-byte block data cache *(write-back).*
- C55x:
  - Various models have 16KB, 24KB cache.
  - Can be used as scratch pad memory.

Μνήμες ελεύθερης πρόσβασης *(scratch pad)*

- Εναλλακτικές προς την κρυφή μνήμη.
- Ο προγραμματιστής καθορίζει τι αποθηκεύεται.
- Δεν υπάρχει ελεγκτής.
- Βρίσκεται on-chip.
- Παρέχει προβλέψιμη συμπεριφορά, με κόστος τον προσεκτικό σχεδιασμό του προγράμματος.
- Η κρυφή μνήμη του C55x μπορεί να ρυθμιστεί ως scratch pad.

Μονάδες διαχείρισης μνήμης *(memory management unit)*

- Μεταφράζει διευθύνσεις μεταξύ της CPU *(σχετικές διευθύνσεις)* και της φυσικής μνήμης *(απόλυτες διευθύνσεις).*
- Ονομάζεται και απεικόνιση μνήμης *(memory mapping).*
- Παρέχει και επιπρόσθετες λειτουργίες, όπως προστασία μνήμης.
- Απαραίτητο στοιχείο για τα σύγχρονα πολυδιεργασιακά ΛΣ.



Μονάδες διαχείρισης μνήμης στους πρώτους υπολογιστές

- Οι παλαιότεροι υπολογιστές χρησιμοποιούσαν MMU για να αντισταθμίσουν τον περιορισμένο χώρο διευθύνσεων *(address space)* στα σύνολα εντολών τους *(δε μπορούσαν να διευθυνσιοδοτήσουν μεγάλες μνήμες).*
- Δεν υπάρχει πια αυτή η ανάγκη, όμως χρησιμοποιούνται για νέες λειτουργίες.
  - Προστασία.
  - Χρήση εικονικής μνήμης *(και αρχείου επέκτασης φυσικής μνήμης).*
  - Θεματική εναλλαγή *(αλλαγή διεργασίας)*, αφού απλοποιείται με μια αλλαγή πινάκων αντιστοίχησης.
- Υπάρχουν ενσωματωμένοι επεξεργαστές χωρίς MMU, γιατί η εικονική μνήμη απαιτεί δευτερεύουσα συσκευή αποθήκευσης, κάτι που απουσιάζει στα ΕΣ.

Οι λειτουργίες της μονάδας διαχείρισης

- Επιτρέπει τα προγράμματα να μετακινούνται *(relocation)* μέσα στη μνήμη *(αλλαγή φυσικής διεύθυνσης)* καθώς εκτελούνται *(τα προγράμματα χρησιμοποιούν σχετικές διευθύνσεις, οποίες δεν αλλάζουν).*
- Επιτρέπει τη χρήση εικονικής μνήμης *(virtual memory):*
  - Τμήματα μνήμης αντιγράφονται σε δευτερεύουσα μνήμη;
  - Τμήματα της μνήμης που βρίσκονται στη δευτερεύουσα μνήμη αντιγράφονται στην κυρίως μνήμη αν ζητηθούν *(αν γίνει page fault....).*

Σφάλμα σελίδας (page fault)

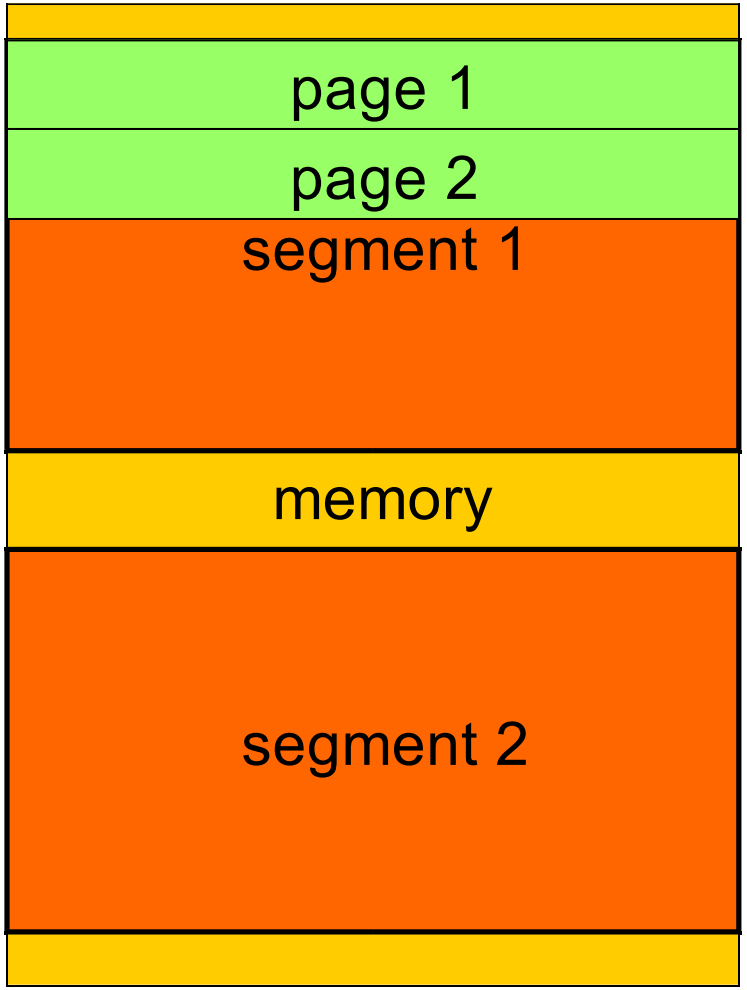
- Αίτηση για πρόσβαση σε διεύθυνση μνήμης που δε βρίσκεται στη φυσική μνήμη *(γιατί έχει αντιγραφεί σε δευτερεύουσα μνήμη, όπως π.χ. swap file στο σκληρό δίσκο).*
- Αν συμβεί σφάλμα σελίδας τότε η διεργασία σταματάει *(blocked)*, και συνεχίζει από εκείνο το σημείο, μόνο όταν έχουν μεταφερθεί τα αντίστοιχα τμήματα στη μνήμη, και έχουν ενημερωθεί οι πίνακες της διεργασίας στη MMU.
- **ΠΡΟΣΟΧΗ:** Αν συμβεί page fault στο ΛΣ για κώδικα που εκτελείται σε kernel mode, τότε συνήθως το σύστημα κολλάει *(bsod, panic, halt)* γιατί το page fault χρησιμοποιεί IRQ μεγαλύτερο από το scheduler, και άρα δε μπορεί να χρονοδρομολογηθεί η μεταφορά της σελίδας από το swap στη RAM ή έχουν απενεργοποιηθεί τα interrupts (π.χ. FreeBSD: `Fatal trap 12: page fault while in kernel mode, fault code = supervisor read, page not present, kernel trap 12 with interrupts disabled`).

Μετάφραση Δ/νσεων

- Απαιτεί καταχωρητές ή πίνακες για να επιτρέψει την αντιστοίχηση διευθύνσεων λογικών σε φυσικών.
- Δυο βασικά στυλ μετάφρασης *(διαφορετικά πλεονεκτήματα για την κάθε αρχιτεκτονική)*:
  - **Κατατμημένη** (Segmented) (αυθαίρετο μέγεθος);
  - **Σελιδοποιημένη** (συγκεκριμένο μέγεθος).
- Η κατάτμηση και η σελιδοποίηση μπορούν να συνδυαστούν (π.χ. x86).

Τμήματα και σελίδες (1/2)

- Με τις σελίδες επιτρέπεται η δυνατότητα του κατακερματισμού μνήμης *(μη συνεχόμενης δέσμευσης χώρου).*
- Με την κατάτμηση χρησιμοποιείται μια αυθαίρετα τυχαίου μεγέθους περιοχή της μνήμης.

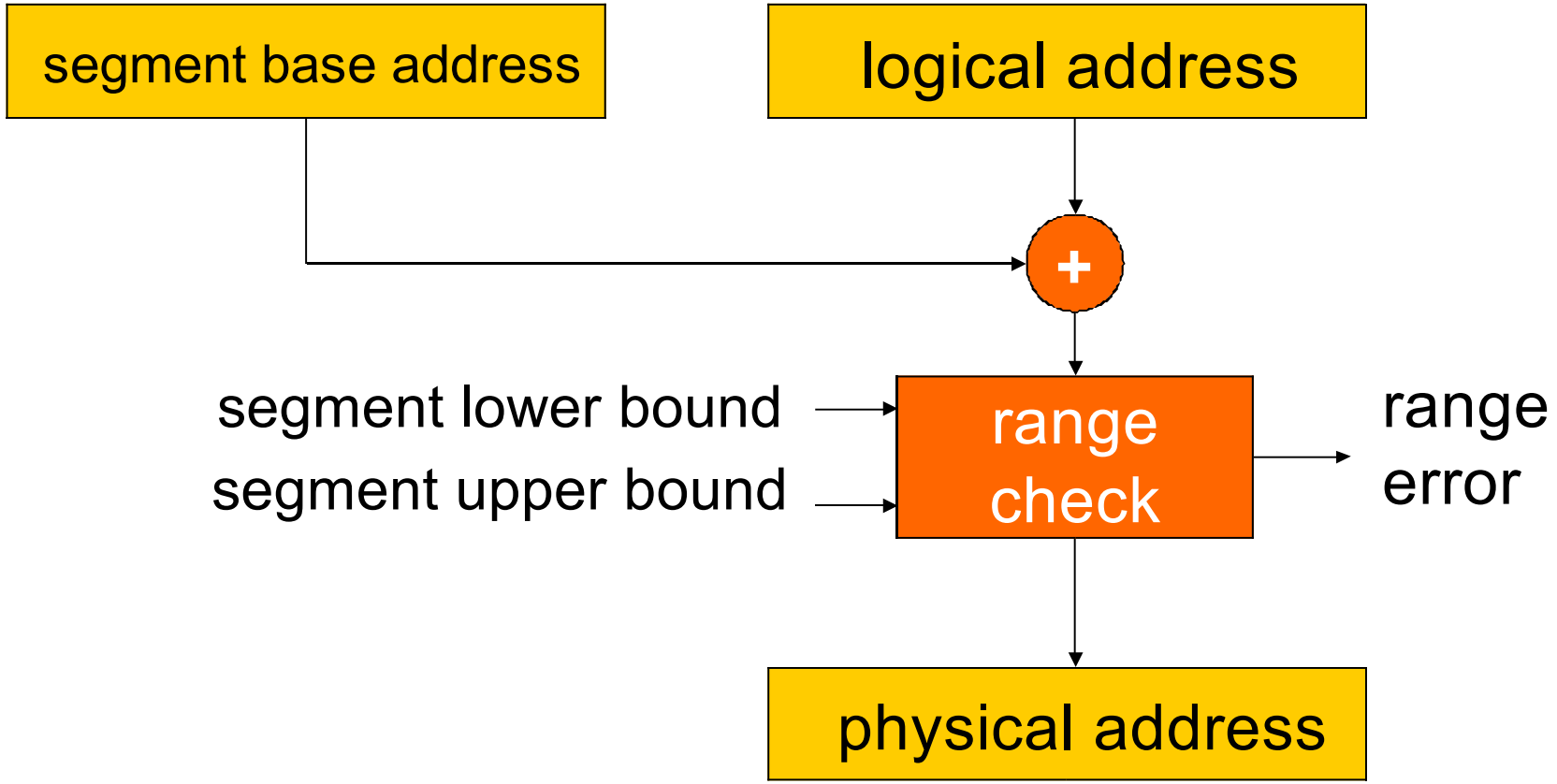


Τμήματα και σελίδες (2/2)

- Το τμήμα περιγράφεται από την αρχική διεύθυνση και το μέγεθος του.
- Οι σελίδες είναι ομοιόμορφου μεγέθους.
- Με το συνδυασμό: χρησιμοποιεί τμήματα και κάθε τμήμα είναι διαιρεμένο σε σελίδες *(μετάφραση δυο βημάτων).*
- Οι σελίδες επιτρέπουν τον κατακερματισμό, αφού μπορούν να είναι διεσπαρμένες στη φυσική μνήμη.

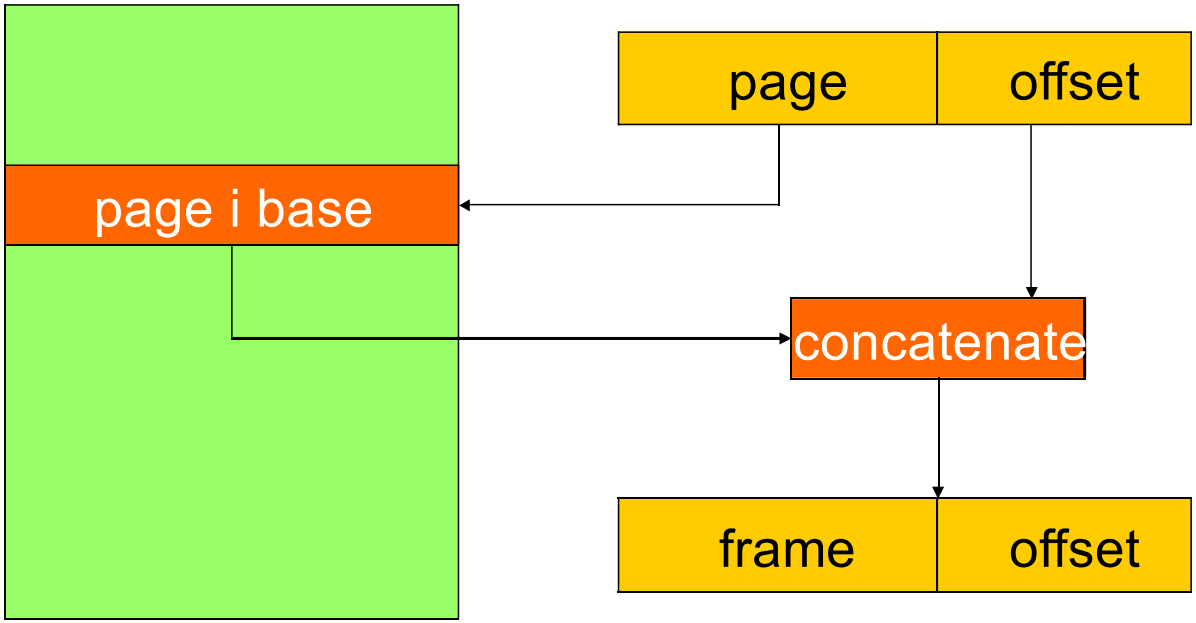
Μετάφραση διεύθυνσης για ένα τμήμα

Χρησιμοποιείται ο καταχωρητής τμήματος και ανώτερου ορίου.



Μετάφραση διεύθυνσης για μια σελίδα

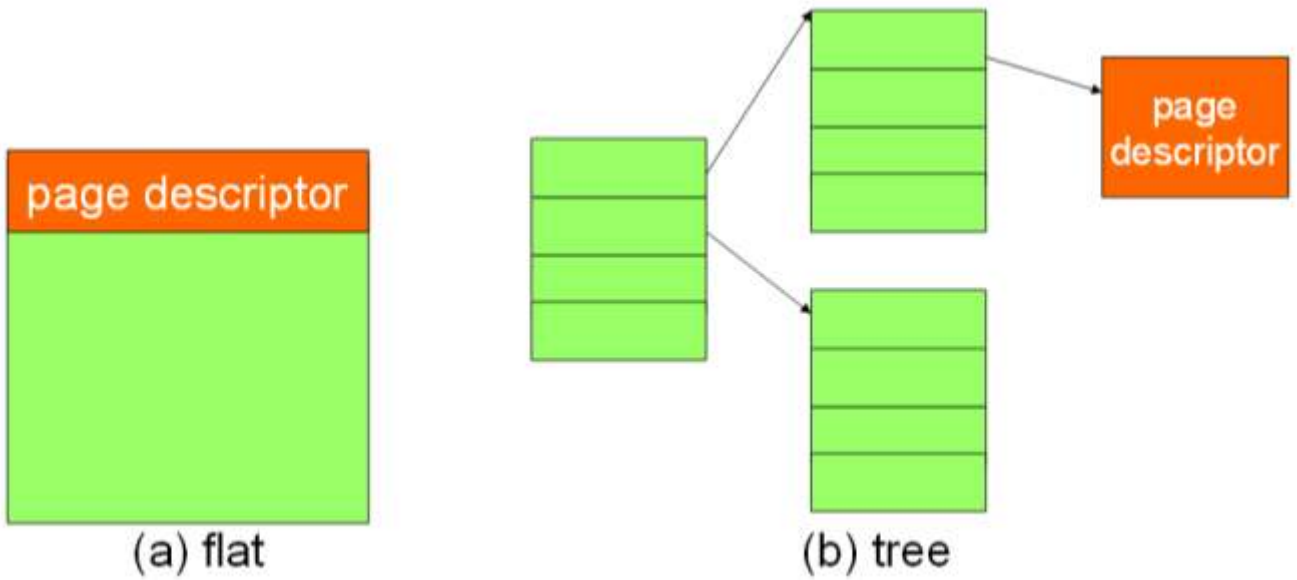
- Η σχετική διεύθυνση χωρίζεται σε page και offset. Από το page γίνεται η αντιστοίχηση στον πίνακα σελίδων και εξάγεται το frame number. Το frame number μαζί με το offset δημιουργούν την απόλυτη διεύθυνση μνήμης.
- Ο πίνακας σελίδων είναι μεγάλος *(βρίσκεται off-chip).*



Page size: 512 - 4KB

Οργάνωση πίνακα σελίδων

- Επιτρέπονται πολλαπλές διοργανώσεις.
- Επίπεδος *(flat)* ή δένδρο *(tree).*
- Το δένδρο επιτρέπει να δημιουργήσουμε μερικώς κατοικημένο *(sparsed populated)* δένδρο *(που δεν υπάρχουν κάποιες σελίδες).*



Μετάφραση διευθύνσεων και κρυφή μνήμη

- Μεγάλοι πίνακες σελίδων βρίσκονται στην κεντρική μνήμη.
- Απαιτείται πρόσβαση σε off-chip μνήμη.
- Μπορεί να χρησιμοποιηθεί ειδική κρυφή μνήμη που ονομάζεται TLB *(Translation Lookaside Buffer).*
  - Κάθε φορά που ζητείται η μετάφραση, η MMU κοιτάει το TLB μήπως υπάρχει η συγκεκριμένη αντιστοίχηση.

Η υποστήριξη εικονικής μνήμης απαιτεί επιπρόσθετα bits

- Η εικονική μνήμη στα σύγχρονα ΛΣ, χρησιμοποιεί σελιδοποίηση ή κατατμημένη σελιδοποίηση, ώστε σε περίπτωση σφάλματος να είναι απαραίτητο να μεταφερθούν μόνο συγκεκριμένες σελίδες.
- Επιπρόσθετα bit
  - **Παρουσία** (Present) bit *(αν η σελίδα βρίσκεται στη RAM ή όχι).*
  - **Ακάθαρτο** (Dirty) bit *(μας δείχνει αν η σελίδα/τμήμα έχει τροποποιηθεί και δεν έχει γραφεί στο δίσκο).*
  - **Άδεια** (Permission) bit *(προστασία σελίδας, ανάγνωση, εγγραφή, εκτέλεση, όχι εκτέλεση, κοινόχρηστη, user page, supervise mode page, κτλ).*

Μονάδα προστασίας μνήμης (MPU, Memory protection Unit)

- Η MPU είναι μια απλοποιημένη MMU.
- Ανιχνεύει άκυρες προσβάσεις σε μνήμη *(μόνο).*
- Η MPU υποστηρίζει:
  - Περιοχές με προστασία.
  - Περιοχές με υπερκάλυψη *(overlapping).*
  - Προστασία πρόσβασης.
  - Ενημέρωση των ιδιοχαρακτηριστικών από και προς το σύστημα.
  - Υποστήριξη ποικίλων χαρακτηριστικών ανά τμήμα *(Non-cacheable, Write back, write through, write allocate, read allocate, full access, no access, read/write, priviledge use only).*

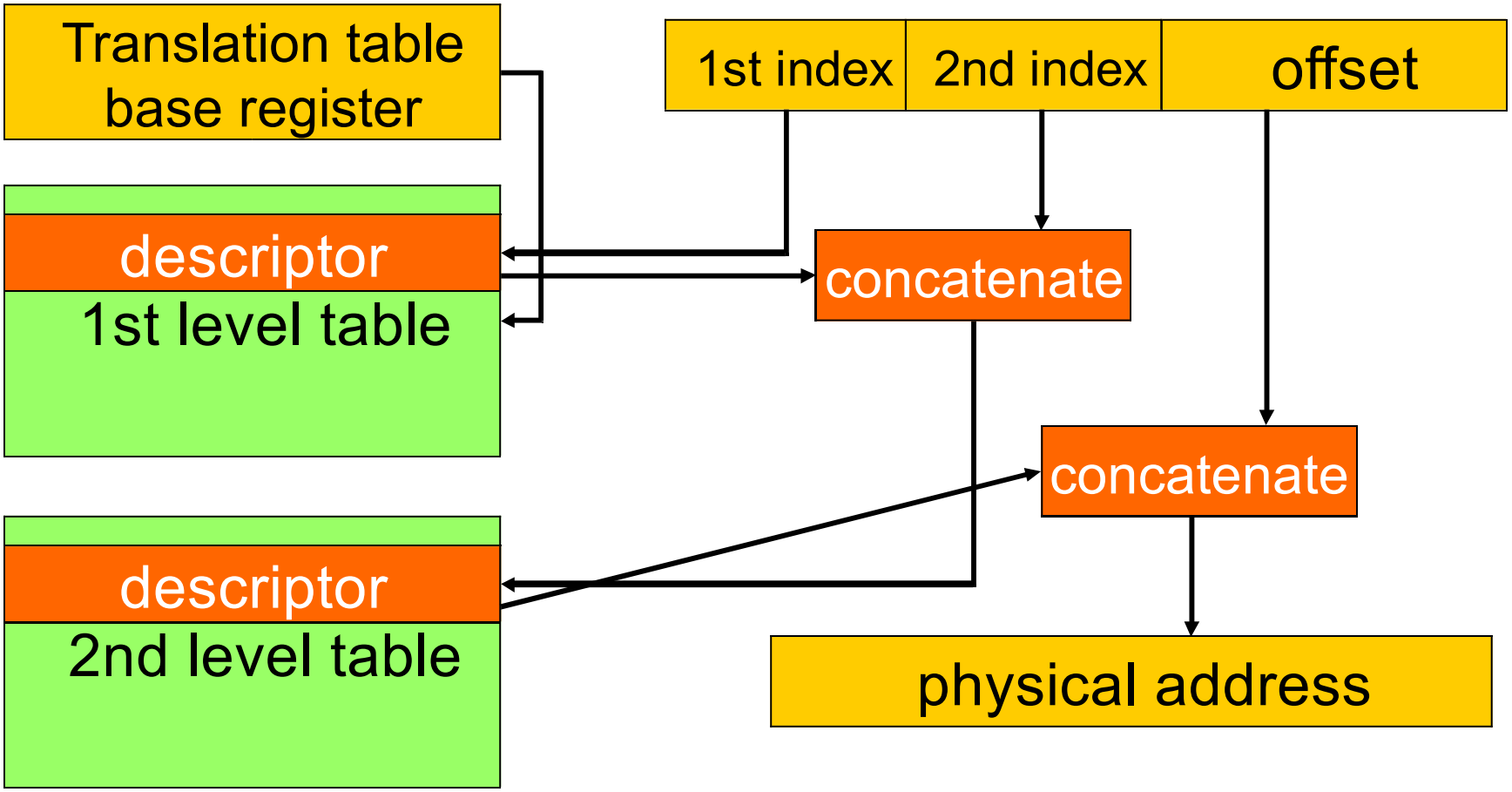
Η MMU δε χρησιμοποιείται συνήθως στα ενσωματωμένα σε αντίθεση με τη MPU

- http://www.embeddedinsights.com/channels/2010/07/21/to-mmu-or-not-to-mmu/
- “Memory protection is needed for security/stability if you have multiple processes running in your system. An MMU is useless for deeply embedded systems running only a single multi threaded process”.
- “MMU slows down access to the physical memory and local bus, at least on the “large” processors like PowerPC or Intel.”
- “If you have a system that is mission critical, the use of an MPU is wise, and having an RTOS that provides API’s to control it makes life easier. This way, each thread can have some guaranteed sections of memory”.

MMU @ ARM

- Προαιρετικό μέρος της αρχιτεκτονικής.
- Παρέχει: μετάφραση εικονικών δ/νσεων & προστασία.
- Υποχρεωτικό αν υπάρχει κρυφή μνήμη.
- Τύποι περιοχών μνήμης:
  - section: 1 Mbyte block;
  - large page: 64 kbytes;
  - small page: 4 kbytes.
- Μια διεύθυνση είναι απεικονισμένη σε τμήμα ή σε σελίδα.
- Χρησιμοποιείται σχήμα δυο επιπέδων.

ARM address translation



Απόδοση κεντρικής μονάδας επεξεργασίας

3 παράγοντες που επηρεάζουν σημαντικά την απόδοση ενός προγράμματος

- Κύκλος ρολογιού *(Cycle time).*
- Διοχέτευση *(CPU pipeline).*
- Σύστημα μνήμης.

Διοχέτευση (ή διασωλήνωση)

- Όλες οι σύγχρονες CPU σχεδιάζονται ως διοχετευμένες μηχανές.
- Πολλές εντολές εκτελούνται παράλληλα σε διαφορετικά στάδια ολοκλήρωσης.
- Μια διοχέτευση λειτουργεί καλύτερα όταν τα περιεχόμενα ρέουν ομαλά.
- Ποικίλες καταστάσεις μπορεί να δημιουργήσουν φυσαλίδες διοχέτευσης *(pipeline bubbles)*, που μειώνουν την αποδοτικότητα:
  - Αλλαγή ροής εκτέλεσης;
  - Καθυστερήσεις συστήματος μνήμης;
  - etc.

Μετρικά Κεντρικής Μονάδας Επεξεργασίας

- **Καθυστέρηση ή λανθάνων χρόνος** *(Latency)*: χρόνος που απαιτείται για την πλήρη εκτέλεση μιας εντολής.
- **Ρυθμοαπόδοση** *(Throughput):* αριθμός των εντολών που εκτελούνται στη μονάδα του χρόνου.
- Η διοχέτευση αυξάνει τη ρυθμοαπόδοση χωρίς να μειώνεται η καθυστέρηση.
- Η υπερ-διοχέτευση αυξάνει την καθυστέρηση, λόγω τοποθέτησης πολλαπλών flip-flops.

Διοχέτευση ARM7

- Το ARM 7 έχει διοχέτευση 3 σταδίων:
  - **προσκόμιση** *(fetch)* η εντολή μεταφέρεται από την εξωτερική μνήμη;
  - **αποκωδικοποίηση** *(decode)* κωδικού λειτουργίας *(opcode)* και των παραμέτρων *(operands);*
  - **εκτέλεση** *(execute).*

Υποθέτοντας ότι το κάθε στάδιο της διασωλήνωσης απαιτεί ένα κύκλο ρολογιού, τότε μπορούμε να υπολογίσουμε:

- latency 3cycles,
- throughput 1instruction/cycle

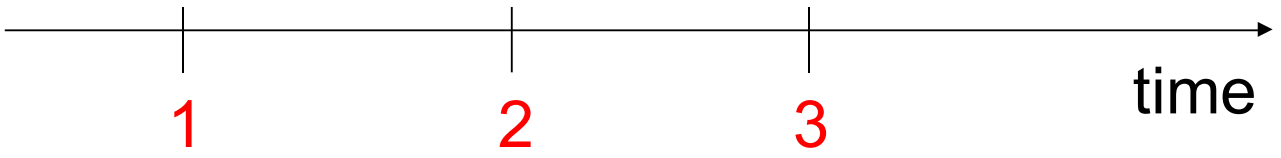
Διοχετευμένη εκτέλεση εντολών στον ARM (1/6)

- Στα σχήματα σε μια κάθετη στήλη, φαίνονται όλες οι εντολές που βρίσκονται στη διοχέτευση σε ποιο στάδιο βρίσκονται.

```
add r0,r1,#5
```

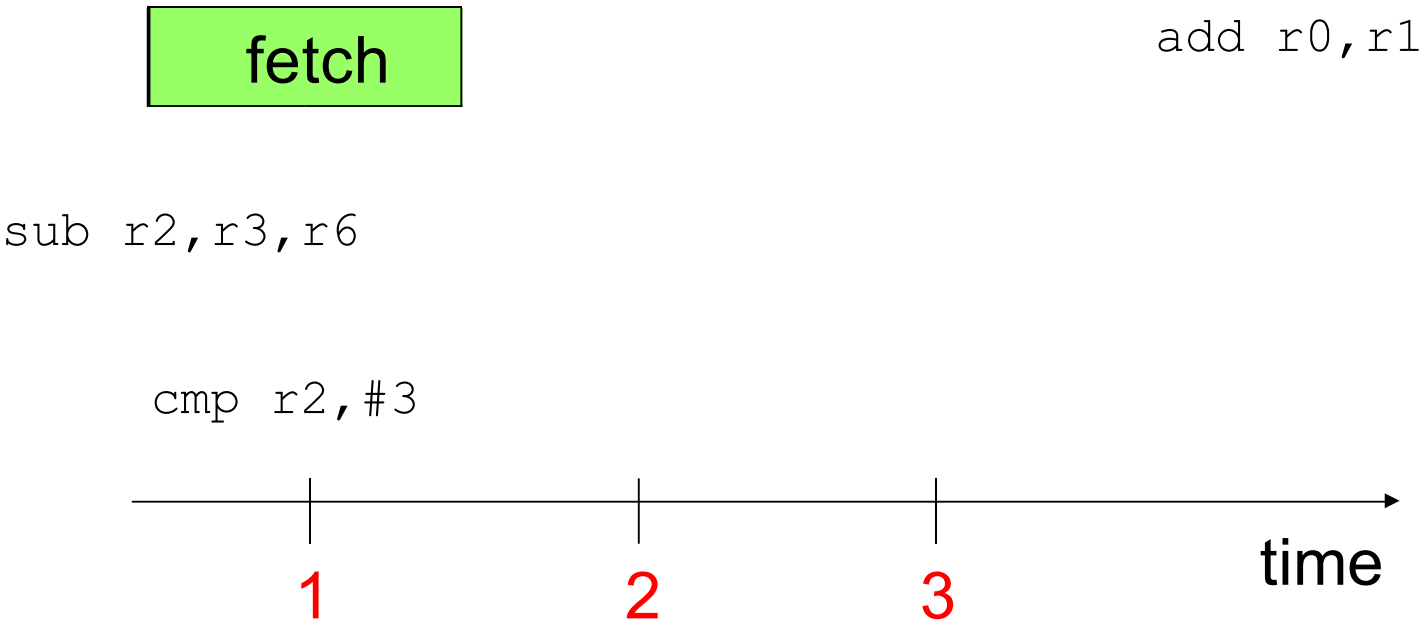
```
sub r2,r3,r6
```

```
cmp r2,#3
```



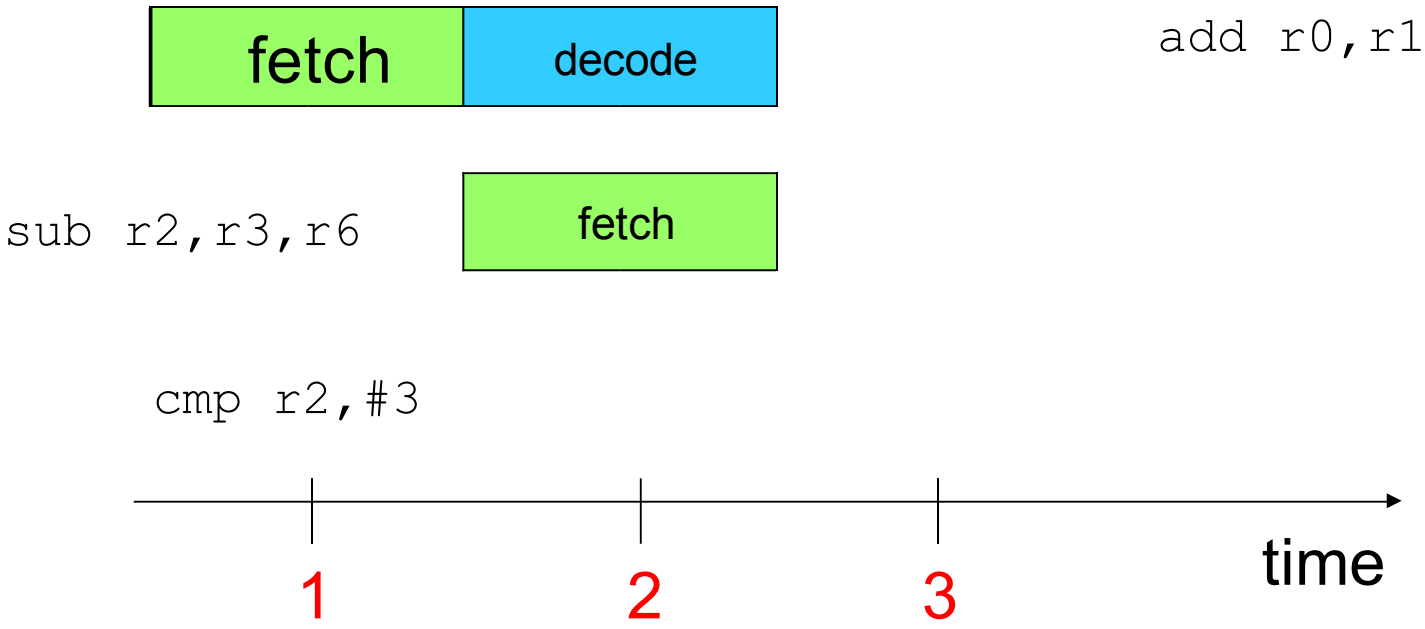
Διοχετευμένη εκτέλεση εντολών στον ARM (2/6)

- Ο ARM είναι RISC και σχεδιάστηκε να κρατάει απασχολημένη τη διοχέτευση. Εμφανίζουν λίγους κινδύνους διοχέτευσης.

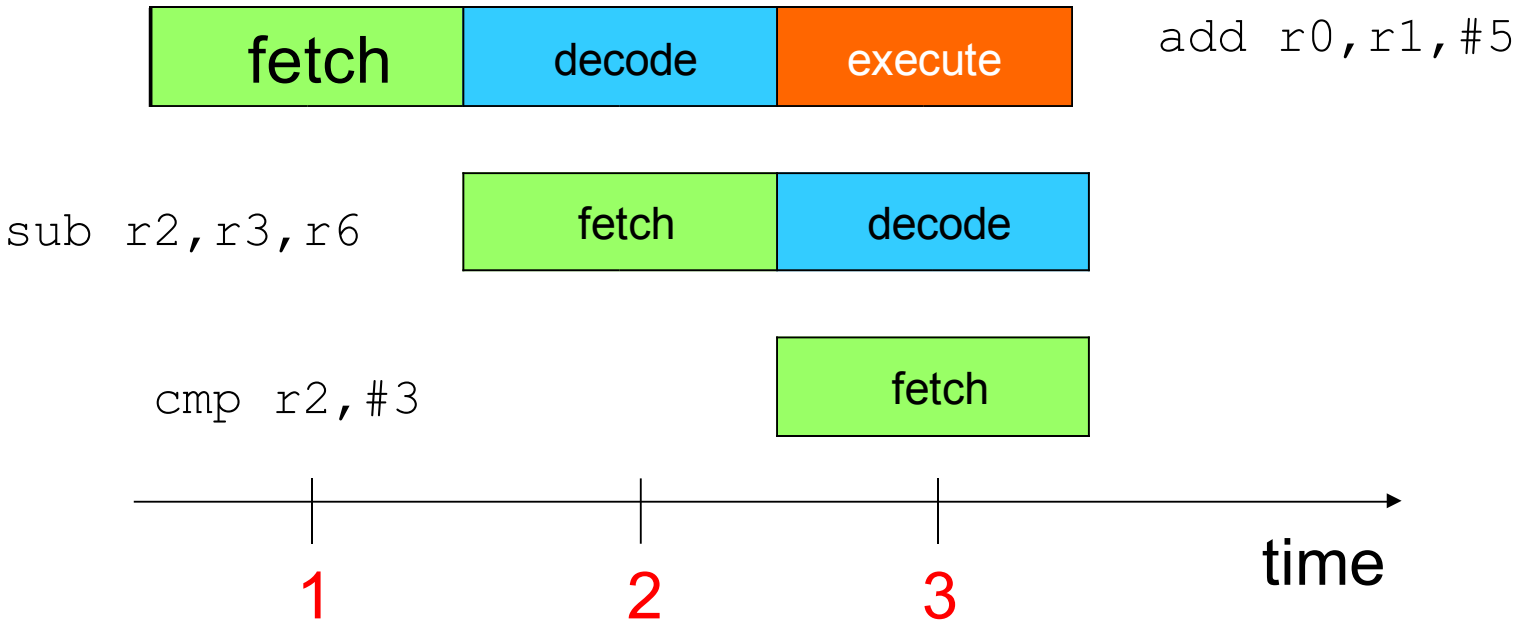


Διοχετευμένη εκτέλεση εντολών στον ARM (3/6)

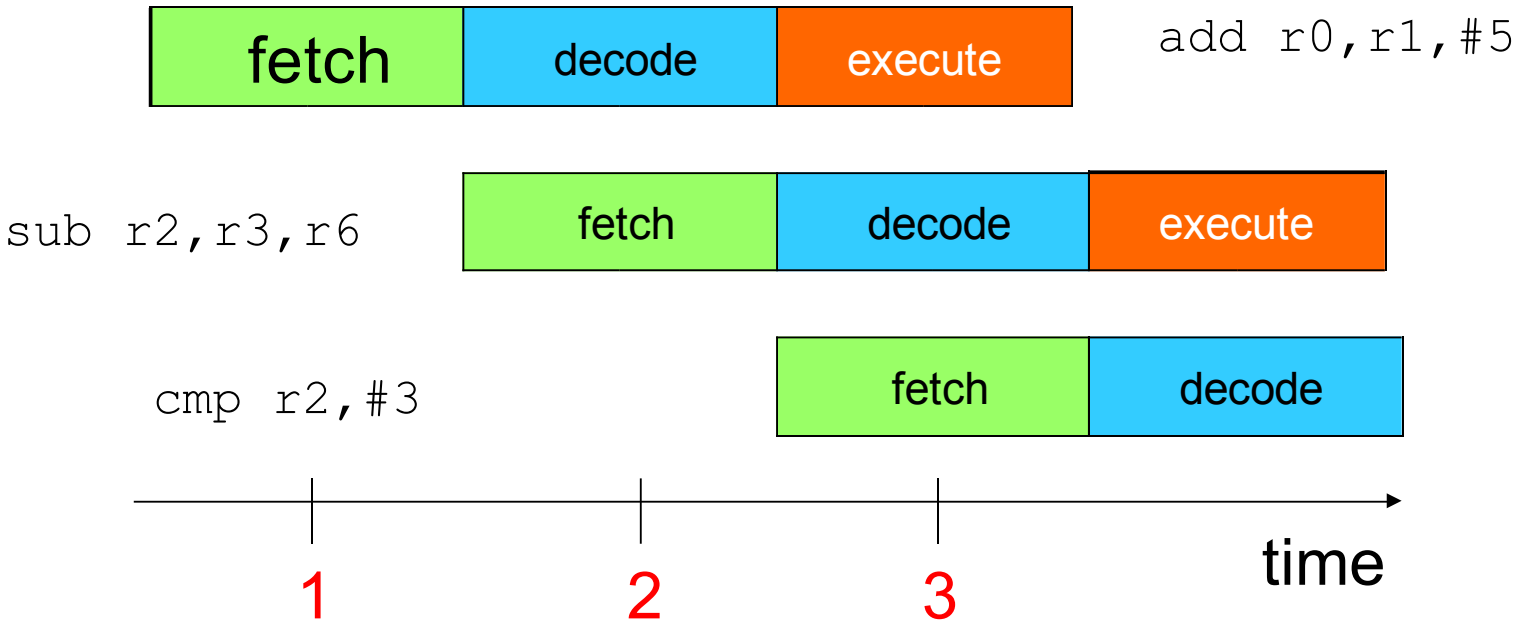
- Οι CISC έχουν ευρεία διακύμανση στο χρονισμό των εντολών.



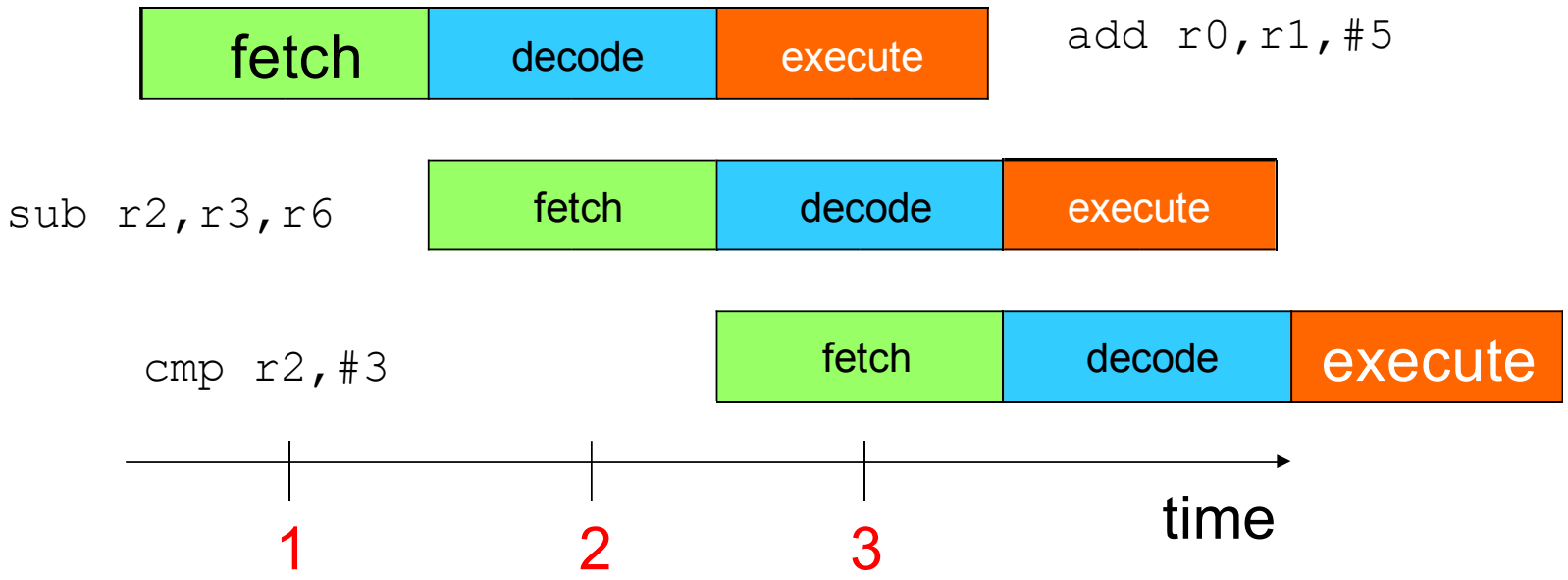
Διοχετευμένη εκτέλεση εντολών στον ARM (4/6)



Διοχετευμένη εκτέλεση εντολών στον ARM (5/6)



Διοχετευμένη εκτέλεση εντολών στον ARM (6/6)

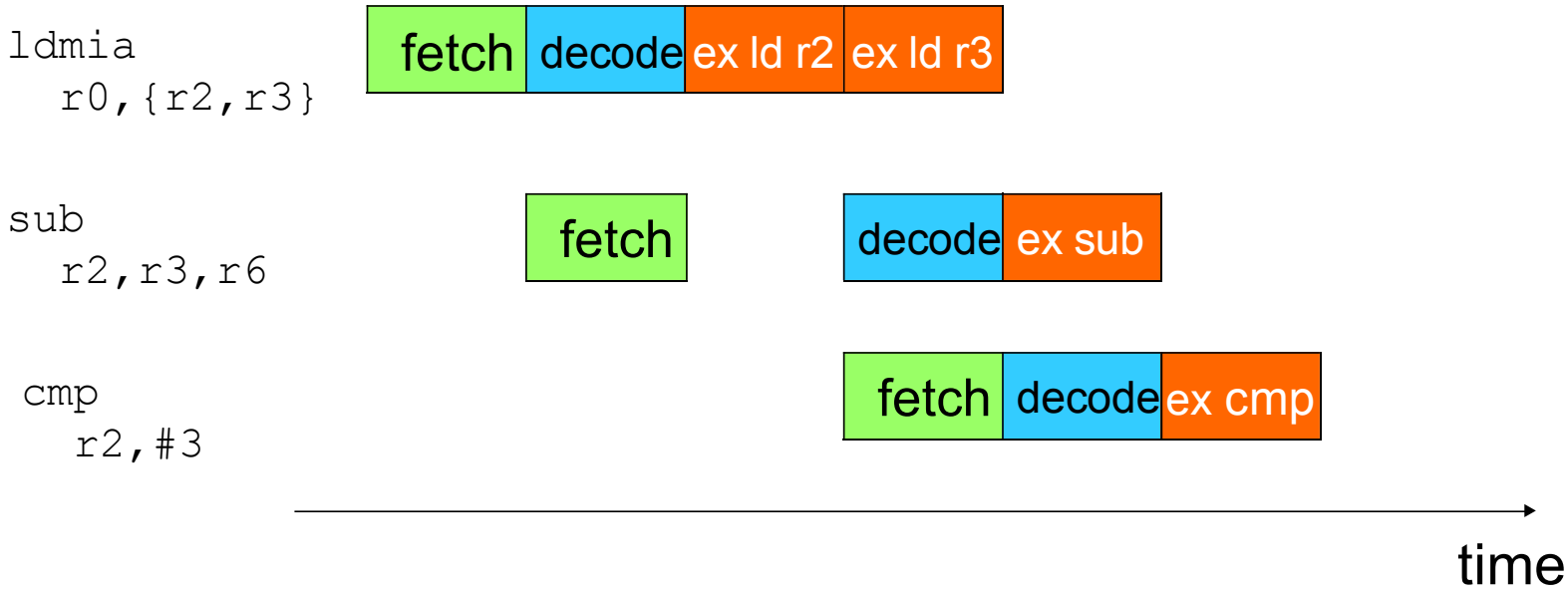


Καθυστερήσεις διασωλήνωσης

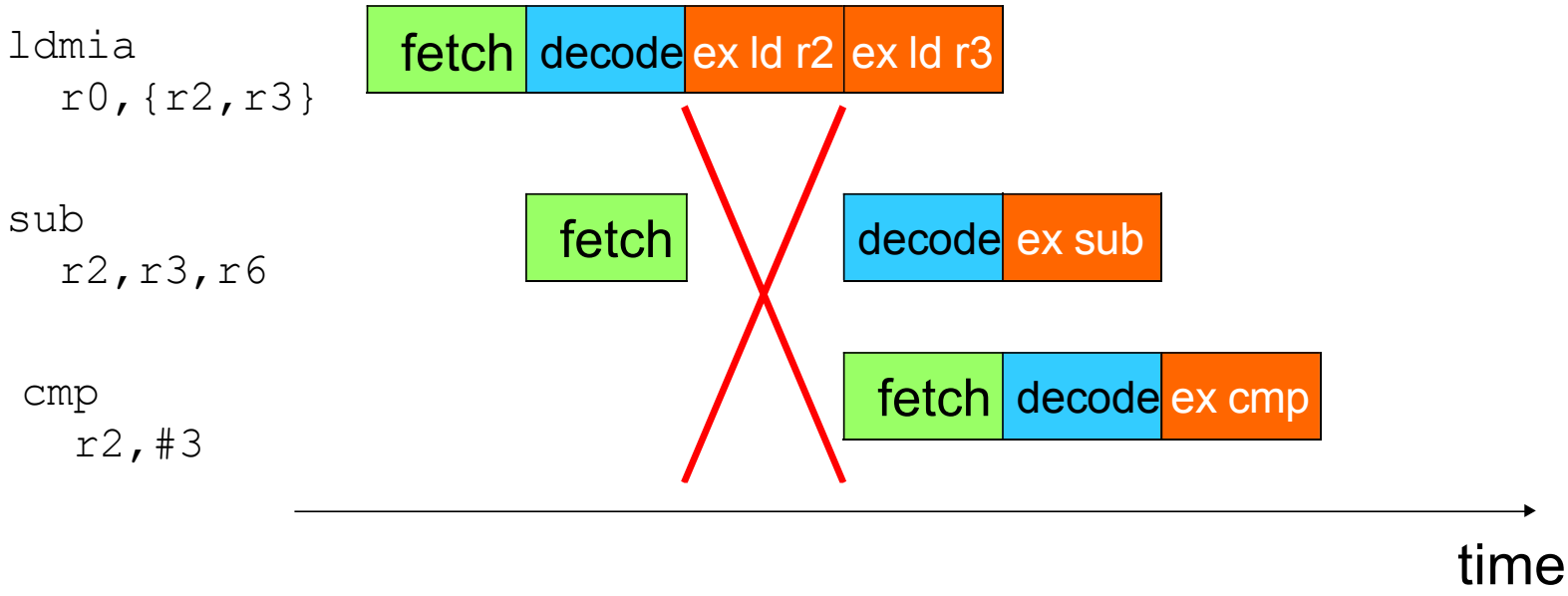
- Αν υπάρχουν τμήματα που δε μπορούν να ολοκληρωθούν στον ίδιο χρόνο, τότε η διασωλήνωση σταματάει προσωρινά *(καθυστερεί).*
- Οι φυσαλίδες διασωλήνωσης που εισάγονται αυξάνουν τον λανθάνων χρόνο και μειώνουν την απόδοση.

Εντολές στον ARM που δεν ολοκληρώνονται σε 1 κύκλο (1/2)

- Ακόμη και σε RISC κάποιες εντολές απαιτούν περισσότερους κύκλους, όπως η εντολή πολλαπλής φόρτωσης.



Εντολές στον ARM που δεν ολοκληρώνονται σε 1 κύκλο (2/2)

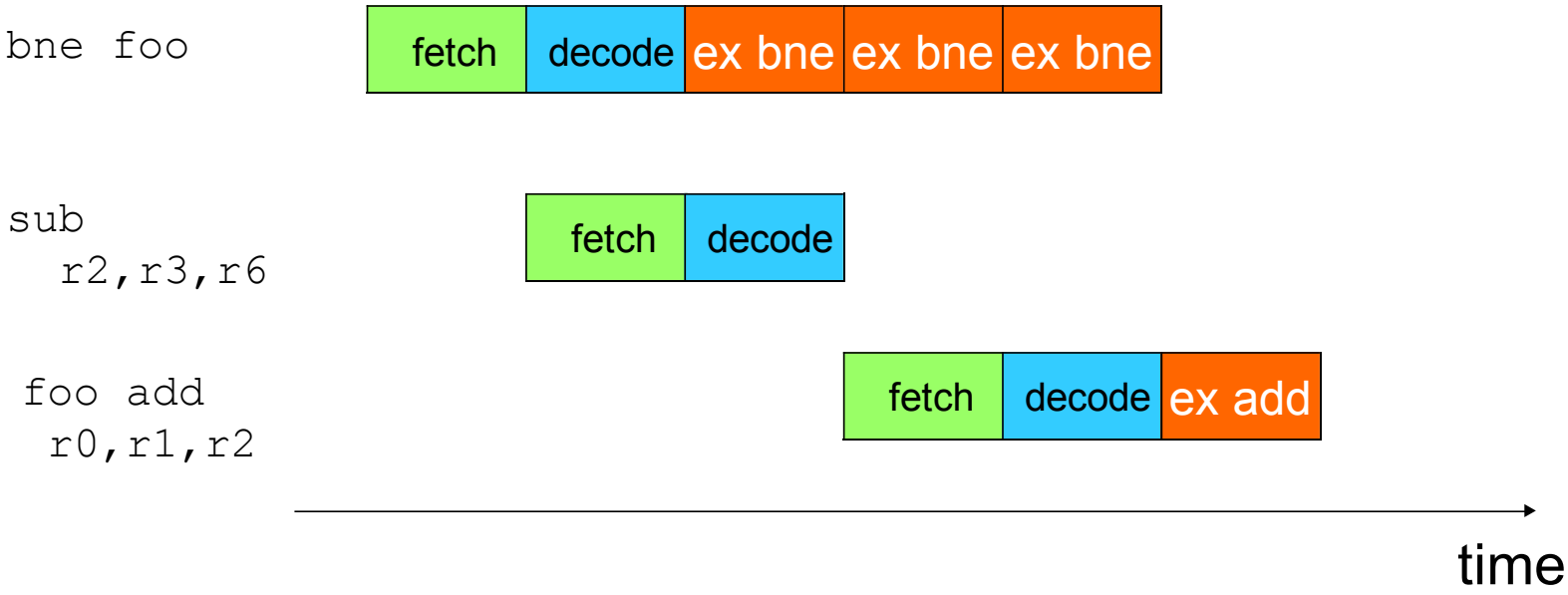


Καθυστερήσεις λόγω ελέγχου

- Οι διακλαδώσεις μπορεί να εισάγουν καθυστέρηση
(ποινή διακλάδωσης, branch penalty).
  - Η ποινή εξαρτάται από το αν ακολουθήθηκε η διακλάδωση ή όχι.
- Αν ακολουθηθεί η διακλάδωση, θα πρέπει να απομακρυνθούν οι εντολές που εισήχθησαν στη διασωλήνωση.
- Αν είναι διακλάδωση υπό συνθήκη, τότε θα υπάρχει καθυστέρηση έως να αξιολογηθεί η συνθήκη.

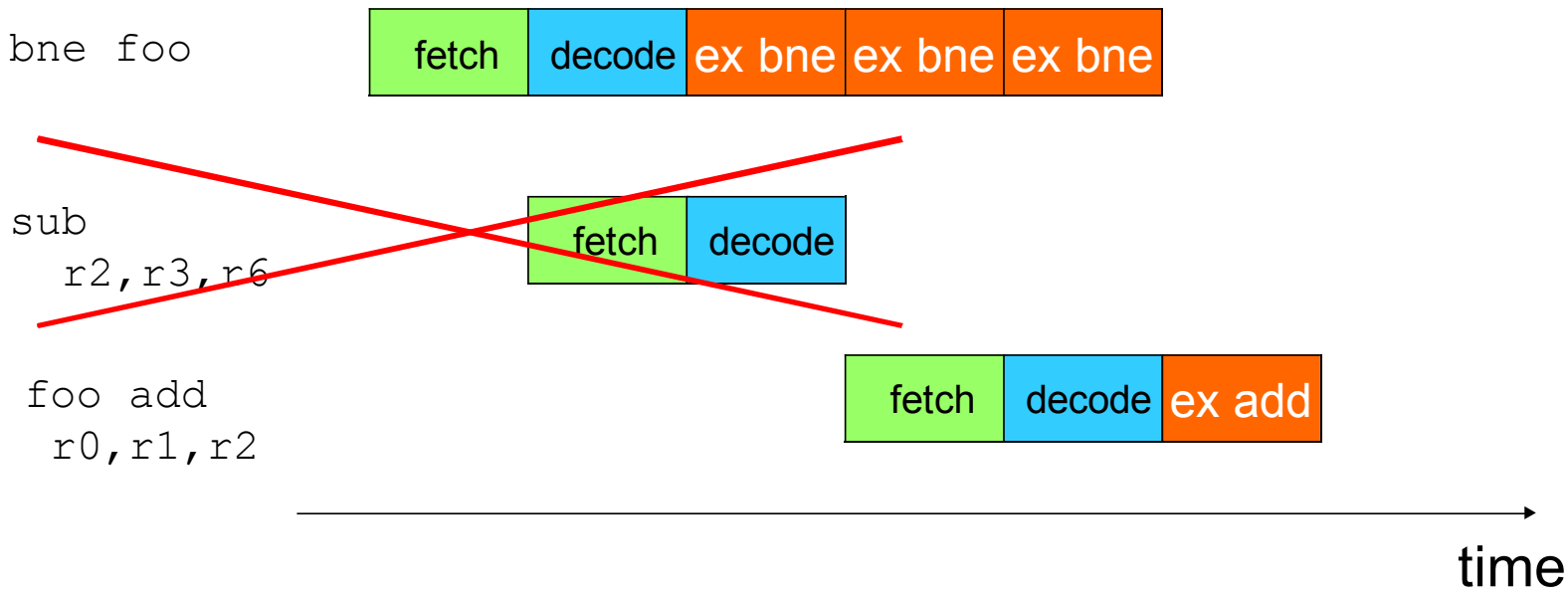
Διοχετευμένη εκτέλεση μιας διακλάδωσης (1/2)

- Γίνεται αξιολόγηση της bne, και εισέρχεται η επόμενη εντολή προς εκτέλεση.



Διοχετευμένη εκτέλεση μιας διακλάδωσης (2/2)

- Επειδή ακολουθείται η διασωλήνωση, θα πρέπει να αδειάσει από τις εντολές που δεν έπρεπε να εκτελεστούν *(εδώ: sub)*



Καθυστερημένη διακλάδωση

- Για να αυξηθεί η απόδοση της διασωλήνωσης, χρησιμοποιείται η καθυστερημένη διασωλήνωση.
- Έτσι, κάποιος αριθμός εντολών αμέσως μετά τη διακλάδωση εκτελείται πάντα, είτε ληθφεί είτε δε ληφθεί η αλλαγή ροής.
- Οι εντολές που τοποθετούνται πρέπει να είναι έγκυρες, είτε ληφθεί είτε δε ληφθεί η διασωλήνωση.
- Η διασωλήνωση διατηρείται γεμάτη.
- Στη χειρότερη περίπτωση τοποθετούνται 1-2 εντολές nop.

Παράδειγμα 3.11: Χρόνος εκτέλεσης βρόχου στο ARM

- Να βρεθεί ο χρόνος εκτέλεσης του παρακάτω βρόχου:

```
for (i=0; i<N; i++)
    f = f + c[i]*x[i];
```

- Αναλύοντας την assembly *(επόμενη διαφάνεια)*, παρατηρούμε ότι η μόνη εντολή που απαιτεί παραπάνω από 1 κύκλο είναι η εντολή αλλαγής ροής εκτέλεσης.
  - BLT loop απαιτεί 1 κύκλο στην καλύτερη περίπτωση και 3 στη χειρότερη.

Ο κώδικας του FIR φίλτρου σε ARM assembly

```
;loop initiation code
MOV r0,#0 ;use r0 for i, set to 0
MOV r8,#0 ;use a separate index for
arrays
ADR r2,N  ;get address for N
LDR r1,[r2] ;get value of N
MOV r2,#0 ;use r2 for f, set to 0
ADR r3,c ;load r3 with address of base
of c
ADR r5,x ;load r5 with address of base
of x

;loop body
loop LDR r4,[r3,r8] ;get value of c[i]
LDR r6,[r5,r8] ;get value of x[i]
MUL r4,r4,r6 ;compute c[i]*x[i]
ADD r2,r2,r4 ;add into running sum

;update loop counter and array
index
ADD r8,r8,#4 ;add one to array index
ADD r0,r0,#1 ;add 1 to i

;test for exit
CMP r0,r1
BLT loop ;if i < N,
continue loop
loopend  ...
```

Η μόνη εντολή που μπορεί να πάρει περισσότερους από ένα κύκλους είναι η BLT

Ανάλυση εκτέλεσης του FIR

Block	Variable	# instructions	# cycles
Initialization	t_{init}	7	7
Body	t_{body}	4	4
Update	t_{update}	2	2
Test	t_{test}	2	[2, 4]

$$t_{loop} = t_{init} + N(t_{body} + t_{update}) + (N-1)\, t_{test,worst} + t_{test,best}$$

Υπερβαθμωτή Εκτέλεση

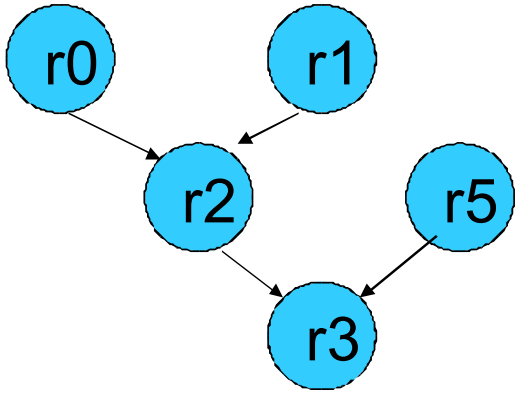
- Βελτίωση της απόδοσης με την υπεβαθμωτή *(superscalar)* εκτέλεση.
- Οι υπερβαθμωτοί επεξεργαστές μπορούν να εκτελούν περισσότερες από μια εντολές σε κάθε κύκλο.
  - Χρησιμοποιούν πολλαπλά διασωληνωμένα μονοπάτια εκτέλεσης.
- Τα προγράμματα εκτελούνται πιο γρήγορα.
- Αυξάνεται η πολυπλοκότητα υλικού.
- Δεν είναι δυνατή η παράλληλη εκτέλεση οποιοδήποτε εντολών. Απαιτείται έλεγχος εξαρτήσεων.

Εξαρτήσεις δεδομένων

- Ο χρόνος εκτέλεσης εξαρτάται από τις παραμέτρους και όχι μόνο από τον opcode. Δύσκολα να εκτιμηθεί με το χέρι. Απαιτείται προσομοίωση.
- Η CPU δε μπορεί να εκδώσει *(issue)* τόσες εντολές όσες θα ήθελε για να είναι πλήρως απασχολημένες όλες οι μονάδες.
- Η Superscalar CPU ελέγχει δυναμικά τις εξαρτήσεις—κατάτ ο χρόνο εκτέλεσης *(σε αντίθεση με VLIW, που οι εξαρτήσεις ελέγχονται από το compiler):*

data dependency

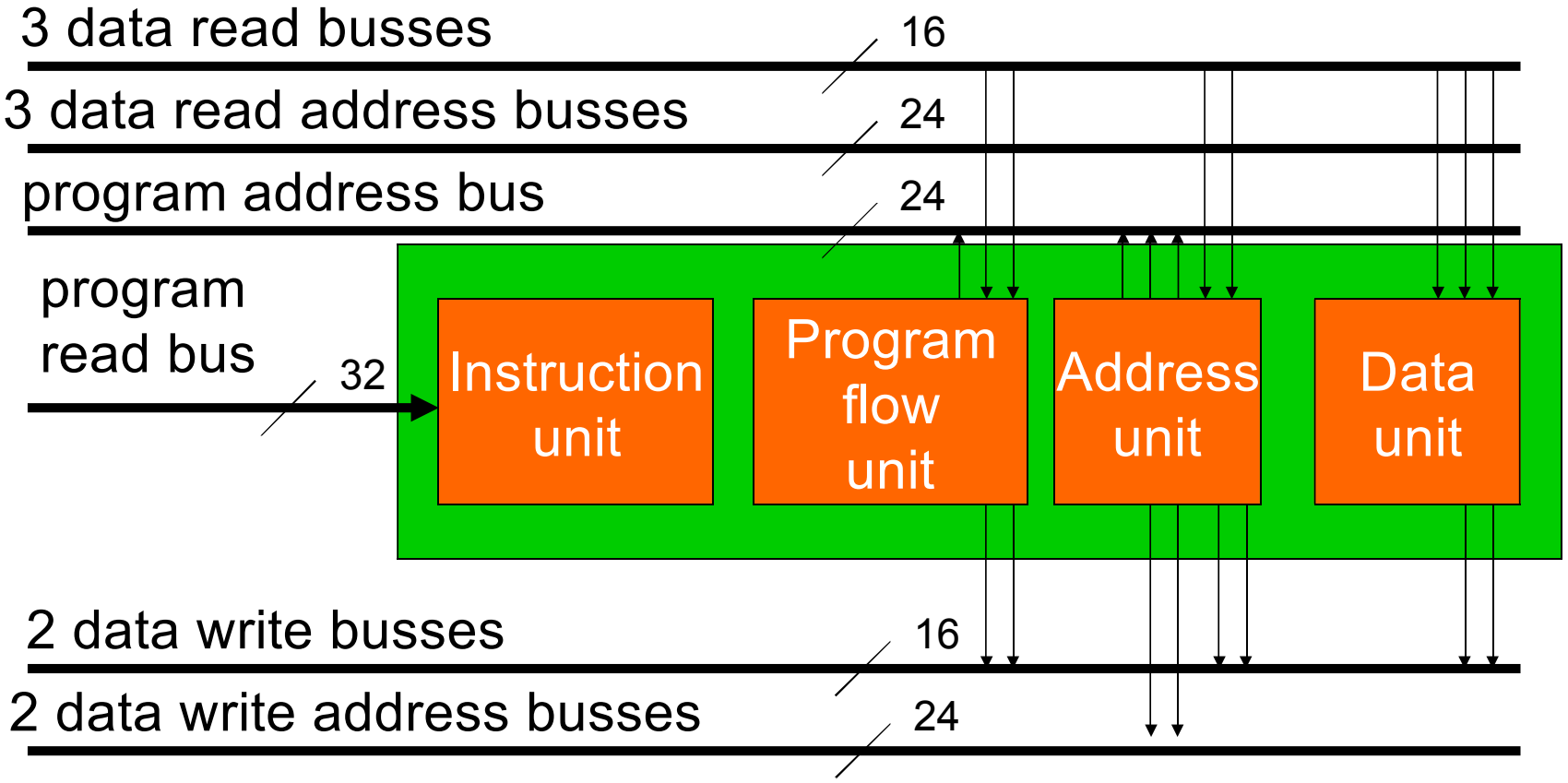
```
add r2,r0,r1
add r3,r2,r5
```



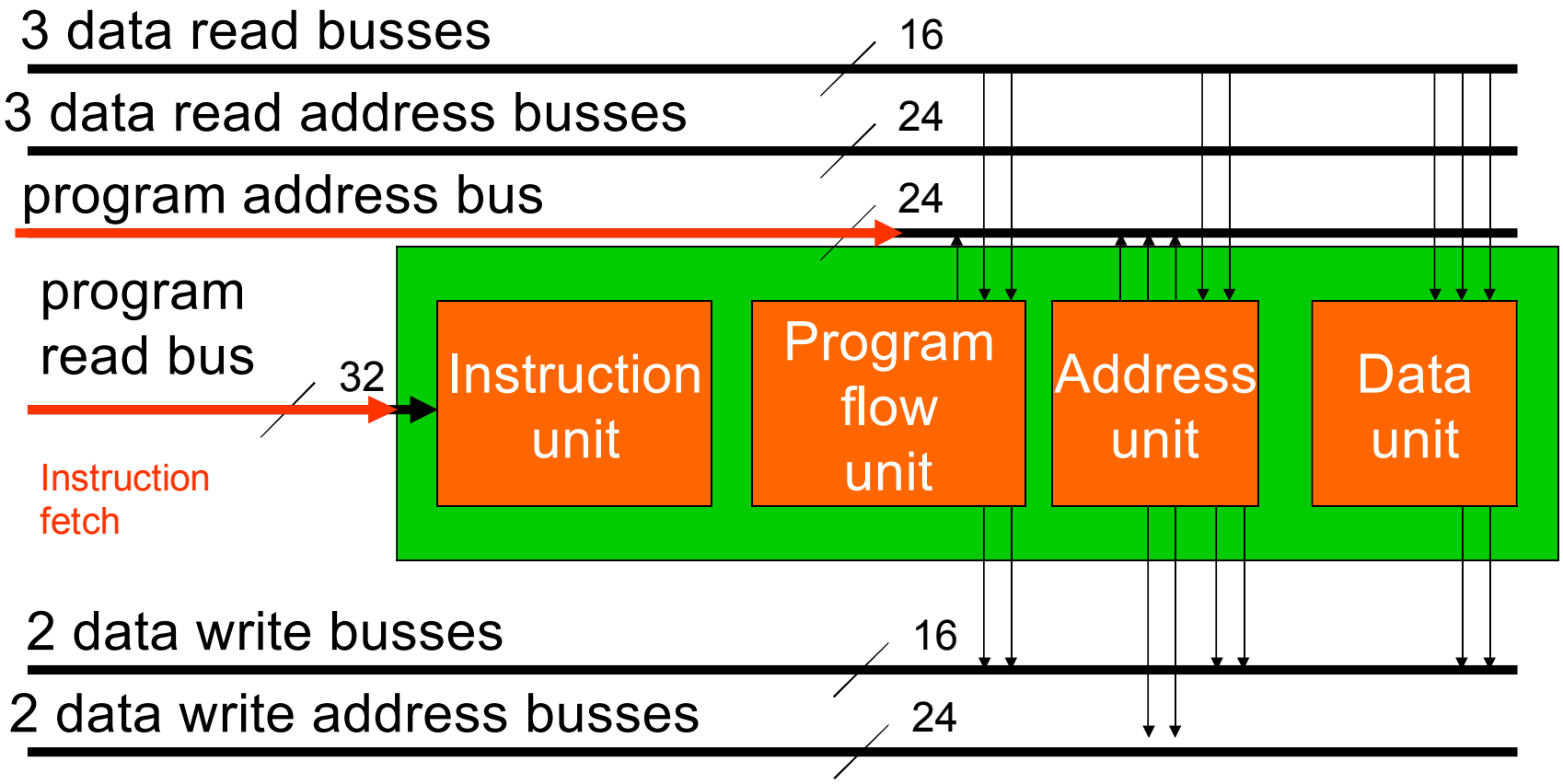
Η διασωλήνωση στο C55x

- 7 στάδια:
  - **fetch**;
  - **decode**;
  - **address**: computes data/branch addresses;
  - **access 1**: reads data;
  - **access 2**: finishes data read;
  - **read stage**: puts operands on internal busses;
  - **execute**.

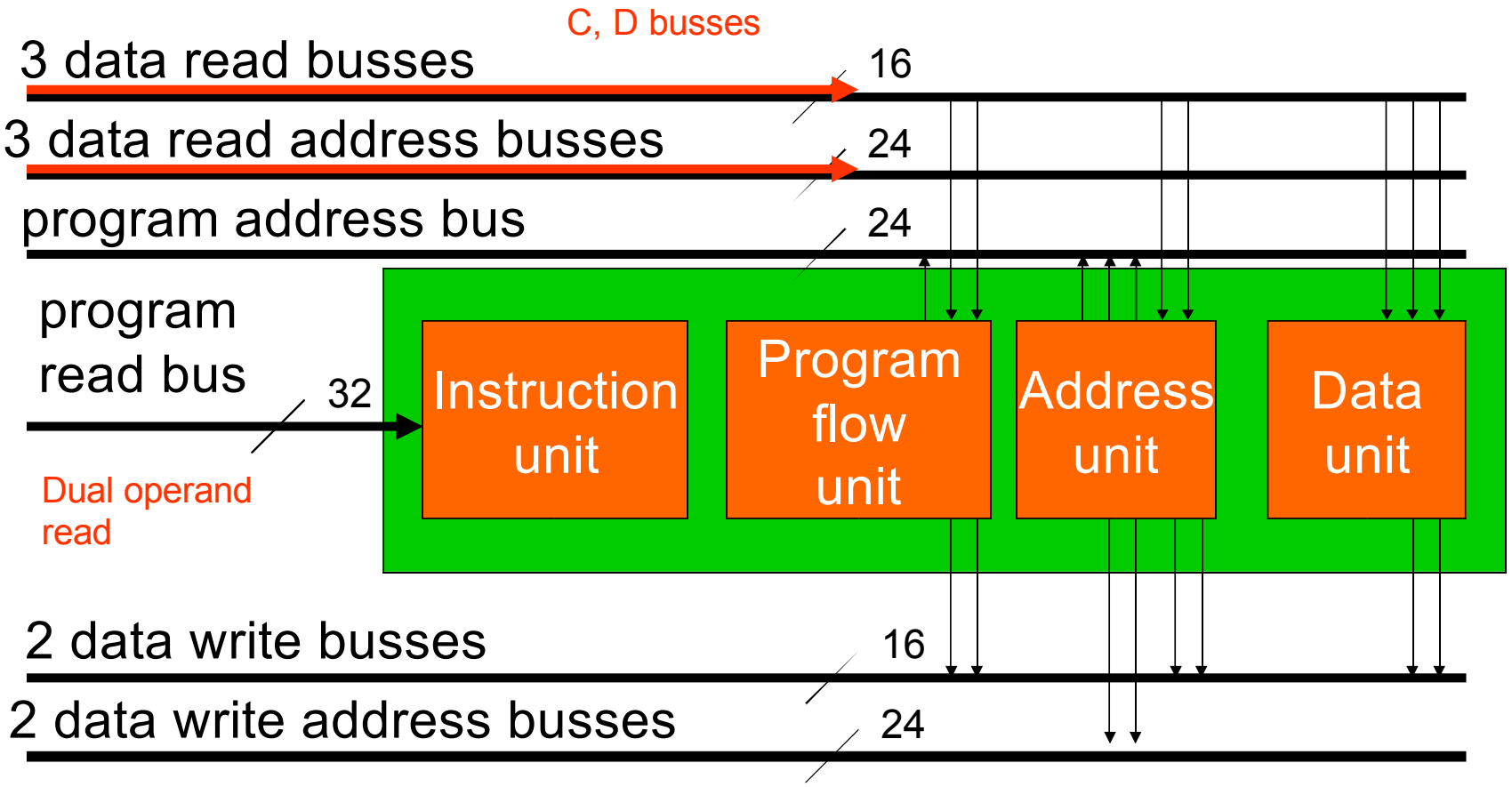
Οργάνωση του C55x



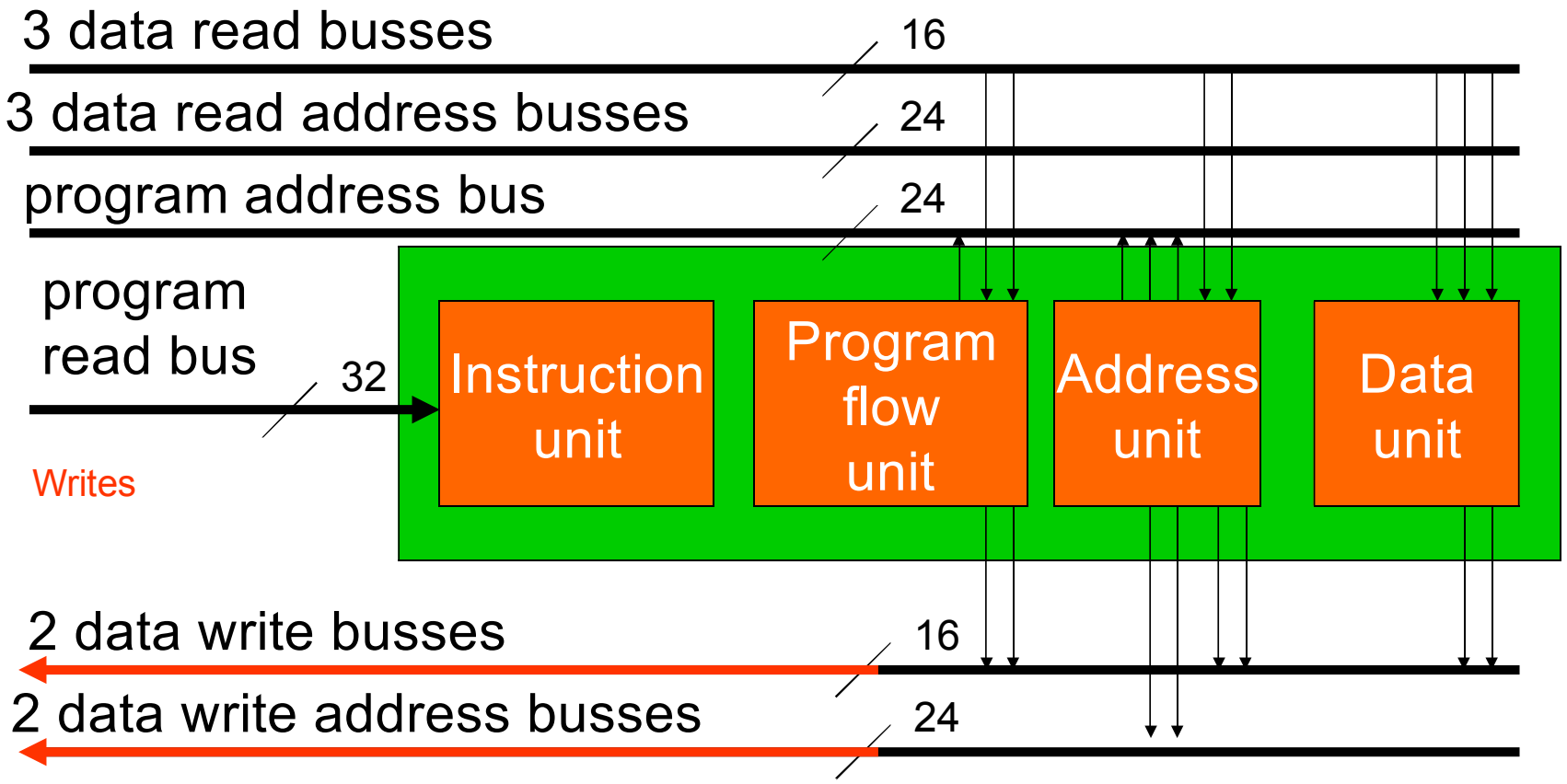
Οργάνωση του C55x: fetch



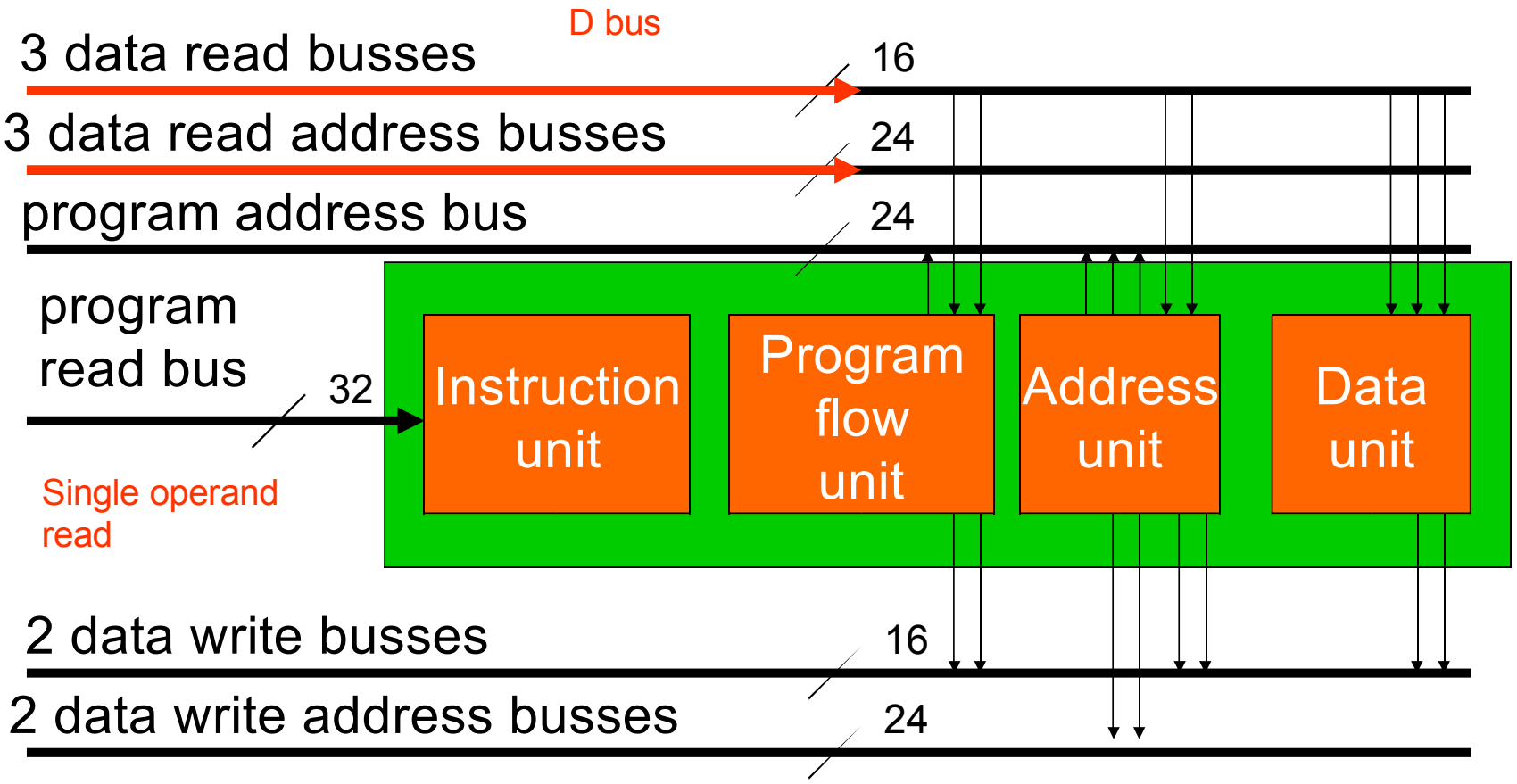
Οργάνωση του C55x: fetch operands



Οργάνωση του C55x: write



Οργάνωση του C55x: read single operand



Δομή του C55x

- Δομή του επεξεργαστή:
  - 3 μονάδες υπολογισμού.
  - 14 λειτουργίες.
- Μπορεί να επιτελέσει δυο λειτουργίες ανά εντολή.
- Κάποιοι συνδυασμοί λειτουργιών δεν είναι έγκυροι.

Οι κίνδυνοι διασωλήνωσης στο C55x

- A-unit ALU/A-unit ALU.
- A-unit swap/A-unit swap.
- D-unit ALU,shifter,MAC/D-unit ALU,shifter,MAC.
- D-unit shifter/D-unit shift, store.
- D-unit shift, store/D-unit shift, store.
- D-unit swap/D-unit swap.
- P-unit control/P-unit control.

Απόδοση του συστήματος μνήμης

- Αν και είναι αόρατες στο μοντέλο προγραμματισμού έχουν βαθιά επίδραση στην απόδοση.
- Οι κρυφές μνήμες εισάγουν μη προβλεψιμότητα στο χρόνο εκτέλεσης.
- Εξαρτάται από τη σειρά εκτέλεσης.
- **Ποινή αστοχίας της Cache** *(miss penalty):* επιπρόσθετος χρόνος εξαιτίας μιας αστοχίας *(συνήθως αρκετοί κύκλοι ρολογιού)*.
- Αρκετοί λόγοι για αστοχία: compulsory, conflict, capacity.

Κατανάλωση Ισχύος CPU

- Εξίσου σημαντική με την απόδοση και η κατανάλωση ισχύος.
- Οι σημερινές CPU σχεδιάζονται λαμβάνοντας υπόψιν την κατανάλωση ισχύος.
- Ενέργεια έναντι ισχύος:
  - Ισχύς είναι η κατανάλωση ενέργειας ανά μονάδα χρόνου.
  - Η παραγωγή θερμότητας εξαρτάται από την κατανάλωση ισχύος.
  - Η διάρκεια ζωής της μπαταρίας εξαρτάται από την κατανάλωση ενέργειας.
- Συνήθως χρησιμοποιείται η ισχύς (power) και για τα δυο.

Κατανάλωση ισχύος σε CMOS

- Όλα τα ψηφιακά συστήματα χτίζονται με κυκλώματα CMOS *(Complementary metal oxide semiconductor {MOS}).*
- Βασικές πηγές κατανάλωσης ισχύος:
  - **Πτώσεις τάσης** *(voltage drops)*:
    η κατανάλωση ισχύος είναι ανάλογη με V^2
    (βελτίωση με μείωση τάσης τροφοδοσίας, ίσως με προσθήκη παράλληλου υλικού για να διατηρηθεί η απόδοση).
  - **Αλλαγή κατάστασης εξόδων** *(toggling, dynamic):* υπάρχει κατανάλωση όταν αλλάζει η τιμή της εξόδου
    (βελτίωση με εξάλειψη περιττών εναλλαγών ή glitches).
  - **Διαρροή** *(leakage, static)*: υπάρχει διαρροή μέσω υποστρώματος ακόμη και αν δεν αλλάζει κάτι); μπορεί να βελτιωθεί με την αποσύνδεση από τη τροφοδοσία *(αλλά απαιτείται σημαντικό χρονικό διάστημα για την επανασύνδεση και εκ νέου αρχικοποίηση της εσωτερικής κατάστασης)*.

Στρατηγικές που ακολουθούνται για την εξοικονόμηση ενέργειας

- Μειωμένα επίπεδα τάσης (5V→3.3V, $5^2/3.3^2=2.29$).
- Χαμηλότερη συχνότητα ρολογιού *(μείωση ισχύς, όχι ενέργειας).*
- Απενεργοποίηση λειτουργικών μονάδων με σήματα ελέγχου, όταν δε χρειάζονται.
- Αποσύνδεση των τμημάτων από την τροφοδοσία όταν δε χρειάζονται.

Διαχείριση ισχύος από το σύστημα

- **Στατική** *(static):* καλείται από το χρήστη, και δεν εξαρτάται από τις δραστηριότητες CPU.
  - *Π.χ. Εντολή για shutdown/hibernate.*
- **Δυναμική** *(dynamic):* η CPU προβαίνει σε ενέργειες για να ελέγξει την ισχύ βασιζόμενη στη δυναμική δραστηριότητα.
  - *Π.χ. Απενεργοποίηση τμημάτων.*

Στοιχεία χαμηλής κατανάλωσης ισχύος στο C55x

- Παράλληλες μονάδες εκτέλεσης---μεγαλύτερους χρόνους αδράνειας δια να απενεργοποιηθούν.
- Πολλαπλοί δίαυλοι δεδομένων:
  - 16-bit ALU vs. 40-bit ALU.
- Η χρήση κρυφής μνήμης εντολών ελαχιστοποιεί τις προσβάσεις στη μνήμη.
- Διαχείριση ισχύος:
  - Ανίχνευση αδρανών λειτουργικών μονάδων.
  - Ανίχνευση αδράνειας στη μνήμη.
  - Ρυθμιζόμενη από το χρήση/προγραμματιστή.

Στοιχεία χαμηλής κατανάλωσης για IBM PowerPC 603 (32bit) 1/3

- Ειδικά σχεδιασμένος για ενσωματωμένα σύστημα με λειτουργία χαμηλής ισχύος, διατηρώντας παράλληλα την υψηλή απόδοση *(2.2 Watt @ 80Mhz).*
- 4 stage pipeline
- 5 execution units *(integer unit, floating unit, branch prediction unit,load/store unit, registry/control unit)*
- 1.6 million transistors
- 2 issue
- Out of order

Στοιχεία χαμηλής κατανάλωσης για IBM PowerPC 603 (32bit) 2/3

- 3 στατικές καταστάσεις λειτουργίας
- Provides doze, nap, sleep modes (2mW).
- Τεχνικές δυναμικής διαχείρισης ισχύος:
  - Χρησιμοποιεί στατική λογική.
  - Μπορεί να απενεργοποιήσει μονάδες εκτέλεσης.
  - Η κρυφή μνήμη οργανώνεται σε υπο-πίνακες για να ελαχιστοποιείται η προσπέλαση. Γίνεται προσπέλαση μόνο 1-2 από τους 8.

Στοιχεία χαμηλής κατανάλωσης για IBM PowerPC 603 (32bit) 3/3

- Ο επεξεργαστής έχει δυο τροφοδοσίες:
- Κυρίως VDD 3.3V *(μπορεί να κλείσει για low power).*
- Βοηθητικό VDDX 1.5V.
- Τρεις καταστάσεις λειτουργίας
  - **Run**: κανονική λειτουργία.
  - **Idle**: σταματάει το ρολόι. Λειτουργούν μόνο: χρονιστής,έλεγχος διακοπών,I/O, διαχειριστής ισχύος. Αν έρθει διακοπή τότε πηγάινει σε RUN.
  - **Sleep**: απενεργοποιεί τα περισσότερα τμήματα; χρησιμοποιείται ρολόι χαμηλής συχνότητας. Γίνεται σε 3 βήματα, κάθε βήμα30 μs; η ενεργοποίηση απαιτεί > 10 ms.
- Υπάρχουν καταχωρητές ορατοί στα προγράμματα για να ελέγχουν και να αλλάζουν τις καταστάσεις.

Το Strongarm, ARM V4 ISA, κατασκευάστηκε από DEC, αγοράστηκε το 1999 από την Intel και μετατράπηκε σε Xscale.

Πόσοστό χρόνου αδράνειας

Unit	Specint92	Specfp92
D cache	29%	28%
I cache	29%	17%
load/store	35%	17%
fixed-point	38%	76%
floating-point	99%	30%
system register	89%	97%

- Οι αδρανείς μονάδες κλείνονται αυτόματα με το σβήσιμο των ρολογιών
- Τα διάφορα στάδια της διοχέτευσης ανοίγουν και κλείνουν.

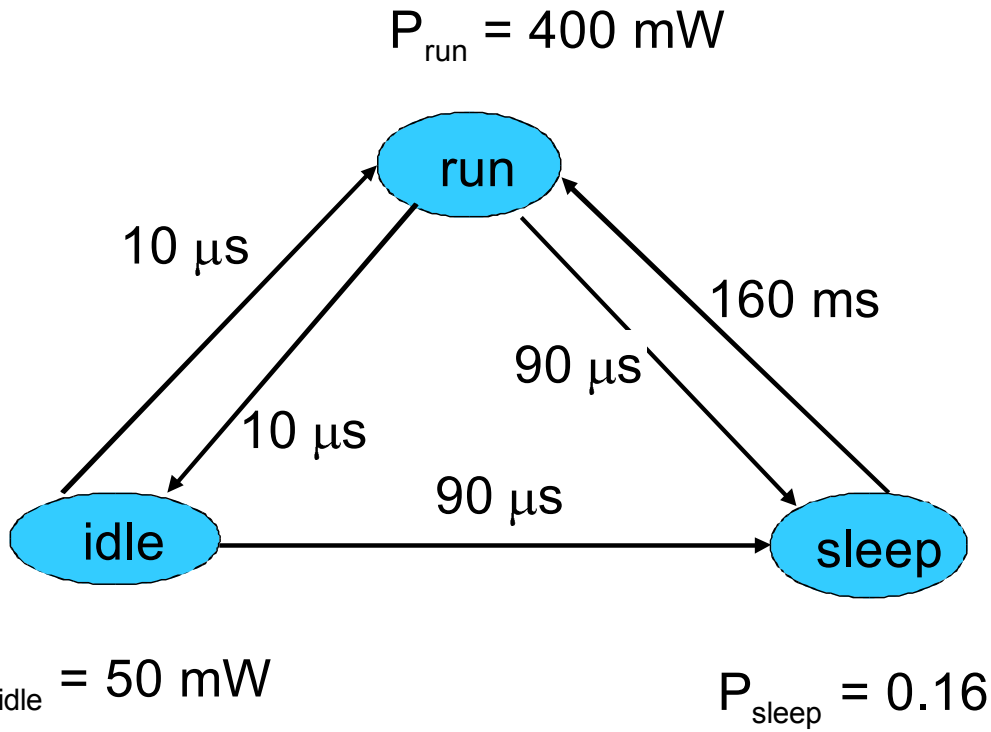
Κόστος απενεργοποίησης

- Η μεταγωγή σε κατάσταση χαμηλής κατανάλωσης κοστίζει *(για να ελεγχθεί η εσωτερική κατάσταση, ειδικά σε διοχέτευση):*
  - Χρόνο.
  - Ενέργεια.
- Πρέπει να αποφασιστεί αν αξίζει η αλλαγή κατάστασης.
- Μπορεί να χρησιμοποιηθεί διάγραμμα καταστάσεων ενέργειας *(power state machine)*. Κάθε κατάσταση συνδέεται στο διάγραμμα, συνδέεται με μια κατάσταση λειτουργίας του CPU.

Η αλλαγή κατάστασης κοστίζει, γιατί απαιτείται να ελεγχθεί κατάλληλα η εσωτερική λογική της CPU *(για να μην αλλοιωθούν τα δεδομένα).*

Η εκκίνηση πρέπει να γίνει προσεκτικά για να αποφευχθούν υπερτάσεις που θα προκαλέσουν δυσλειτουργία.

Μηχανή κατάστασης ισχύος του SA-1100



Παραμένουν λειτουργικές οι:

Real-time clock, timer, interrupt control, general purpose I/O, power manager

Κλείνουν σχεδόν όλες οι μονάδες. Μόνο η βοηθητική τροφοδοσία υπάρχει (με σήμα σε κατάλληλο pin απενεργοποιείται η βασική τροφοδοσία, βασικό ρολόι), ρολόι χαμηλής ταχύτητας για τη λειτουργία του διαχειριστή ενέργειας.

Βιβλιογραφία

Χρησιμοποιήθηκε υλικό από παρουσιάσεις των:

- Wayne Wolf, Overheads for Computers as Components 1st,2nd ed. ,2008, *(παράγραφοι 3.5, 3.6, 3.7)*

Τέλος Ενότητας