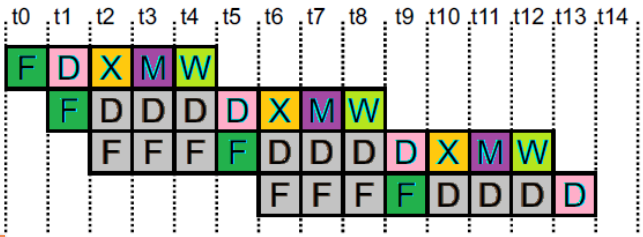
Multicore, SMT

2-Μαΐου-2023

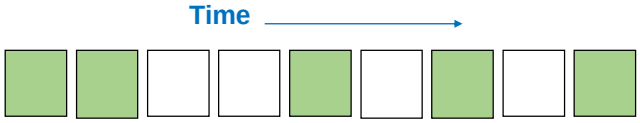
Pipelining…

LW r1, 0(r2)
LW r5, 12(r1)
ADDI r5, r5, #12
SW 12(r1), r5



Εξάρτηση μιας εντολής από το αποτέλεσμα της προηγούμενης εντολής

- Επεξεργαστές με pipelining εμφανίζουν υπο-χρησιμοποίηση των μονάδων υπολογισμών λόγω εξαρτήσεων



Επίλυση Εξαρτήσεων σε Pipeline

- Ποιές τεχνικές υπάρχουν για την βελτιστοποίηση επιδόσεων – χρησιμοποίησης?
- Προσπεράσματα (bypassing), speculation, εκτέλεση-εκτός-ακολουθίας (OoO)…
- Ο Παραλληλισμός σε επίπεδο εντολών (ILP) είναι περιορισμένος ⇒ δεν μπορεί να εκμηδενιστούν όλες οι καθυστερήσεις

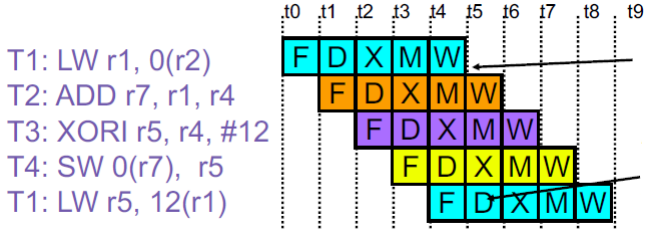
Η υπο-χρησιμοποίηση αυξάνεται σε superscalar επεξεργαστές

3

Επεξεργασία Πολλαπλών Ινών (Multithreading)

- Πως θα υπάρχουν εγγυήσεις ότι δεν υπάρχουν εξαρτήσεις ?
  - → ξεκίνα εντολές από διαφορετικά προγράμματα !

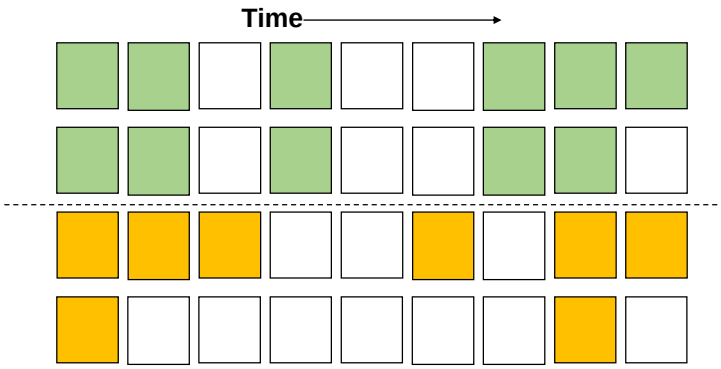
Εναλλαγή από 4 διαφορετικά νήματα (threads) : T1 – T4
Επεξεργαστής 5 σταδίων χωρίς πρόβλεψη για προσπεράσματα !



Το στάδιο WB σε ένα νήμα πρέπει να ολοκληρωθεί προτού η επόμενη εντολή στο ίδιο νήμα κάνει πρόσβαση στο αρχείο καταχωρητών

4

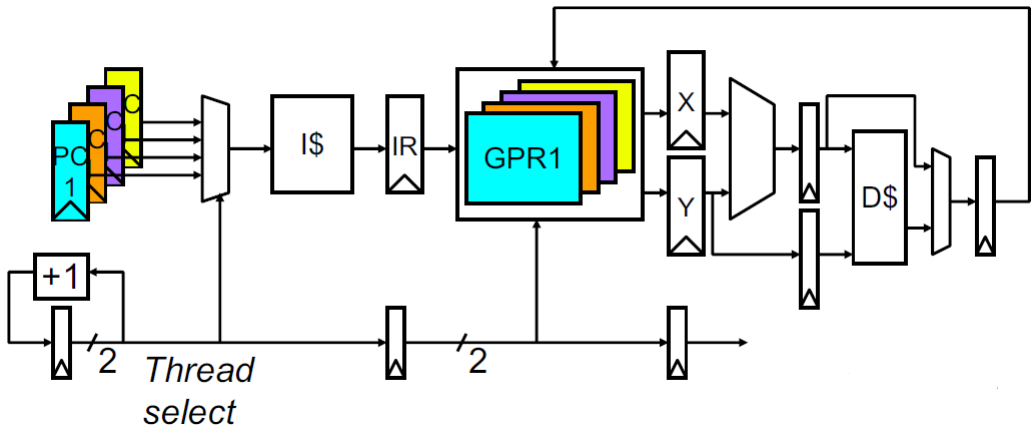
Πολυ-επεξεργαστής σε ένα Chip



- Καταμερισμός των υπολογιστικών μονάδων στους πυρήνες
- Ο παραλληλισμός είναι συγκεκριμένος, χωρίς εξαρτήσεις μεταξύ νημάτων

5

Απλό Pipelining Πολλαπλών Νημάτων



- Η πληροφορία κατάστασης για κάθε νήμα πρέπει να μεταφέρεται προσεκτικά σε κάθε στάδιο

6

Βελτιστοποίηση Ενός Επεξεργαστή Μονού Πυρήνα

+ Η εξέλιξη-ανάπτυξη ενός επεξεργαστή μονού πυρήνα (single-thread) βελτιώνει τις επιδόσεις ενός προγράμματος αυτόματα για τον προγραμματιστή και τον compiler

- Πολύ δύσκολο να σχεδιαστεί (οι αλγόριθμοι με ιδιότητες επεκτασιμότητας για την βελτίωση επιδόσεων ενός νήματος είναι απατηλοί)
- Μεγάλο κόστος σε χώρο και ισχύ (η βελτίωση των επεξεργαστικών στοιχείων για εκτέλεση out-of-order είναι συνήθως ανάλογη με $O(\text{issue width}^2)$)
- Ελάχιστη βελτίωση στις επιδόσεις
- Εφαρμογές που εξαρτώνται σημαντικά από την μνήμη δεν βλέπουν βελτιώσεις...

7

Βελτιστοποίηση με Χρήση Μεγαλύτερης Cache

- Η βελτίωση των επιδόσεων με αύξηση της κρυφής μνήμης ενός επεξεργαστή μονού πυρήνα (single-thread) βελτιώνει τις επιδόσεις ενός προγράμματος αυτόματα για τον προγραμματιστή και τον compiler
- Είναι εύκολο στη σχεδίαση

- Η συνεχής αύξηση της cache δεν δίνει γραμμικά καλύτερες επιδόσεις!
- Πολλά επίπεδα ιεραρχίας στην cache δυσκολεύουν την σχεδίαση

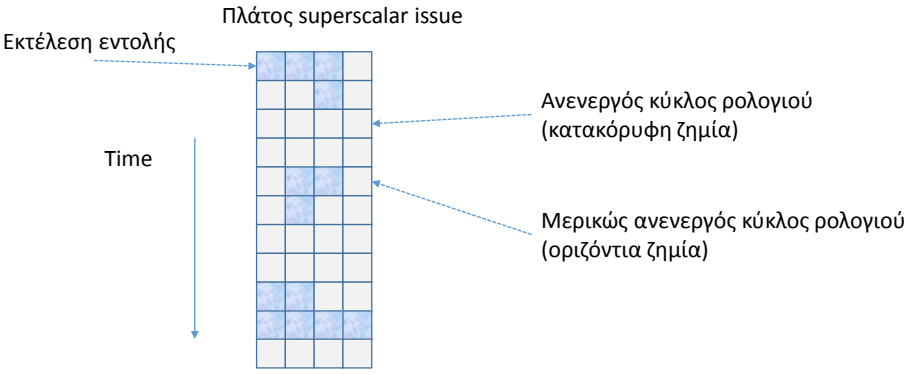
8

Επιλογές Σχεδίασης για Αρχιτεκτονική Πολλών Νημάτων

- Επεξεργασία πολλαπλών νημάτων fine-grain
  - Εναλλαγή νημάτων επεξεργασίας σε κάθε κύκλο ρολογιού
- Επεξεργασία πολλαπλών νημάτων coarse-grain
  - Εναλλαγή νημάτων επεξεργασίας ανά μερικούς κύκλους ρολογιού
    - Σε περίπτωση miss L1 cache
    - Σε περίπτωση miss L2 cache
    - Σε περίπτωση data hazard
- Η επιλογή εξαρτάται από το σχετικό κόστος
  - Κόστος εναλλαγής νημάτων (context switch), αρχικό κόστος
  - Αριθμός νημάτων που μπορούν να τρέξουν ταυτόχρονα
  - Αναμενόμενος παραλληλισμός νημάτων

9

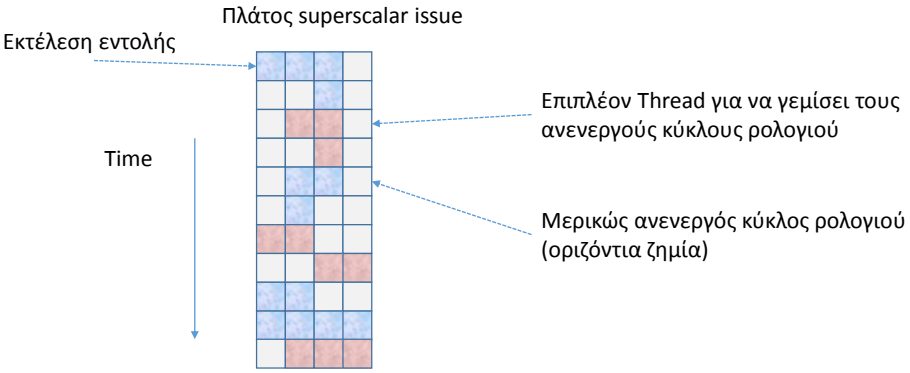
Αποδοτικότητα Μηχανής Superscalar



- Για ποιό λόγο υπάρχει η κατακόρυφη και για ποιο λόγο η οριζόντια ζημία?

10

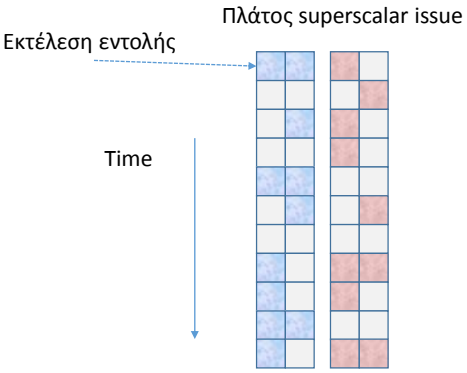
Βελτίωση Αποδοτικότητας Μηχανής Superscalar



- Η εναλλαγή με ένα επιπλέον νήμα μπορεί να εξαλείψει την κατακόρυφη ζημία

11

Πολυεπεξεργαστής Superscalar



- Διαχωρισμός σε ανεξάρτητες μηχανές superscalar
  - Ελατώνεται η οριζόντια ζημία
  - Εμφανίζεται κατακόρυφη ζημία
  - Περιορίζει το μεγιστο throughput του κάθε thread

12

Ιδανική Μηχανή Mutlithreading

Simultaneous Multithreading

Πλάτος superscalar issue

Time

Αλλαγές στο Pipeline:

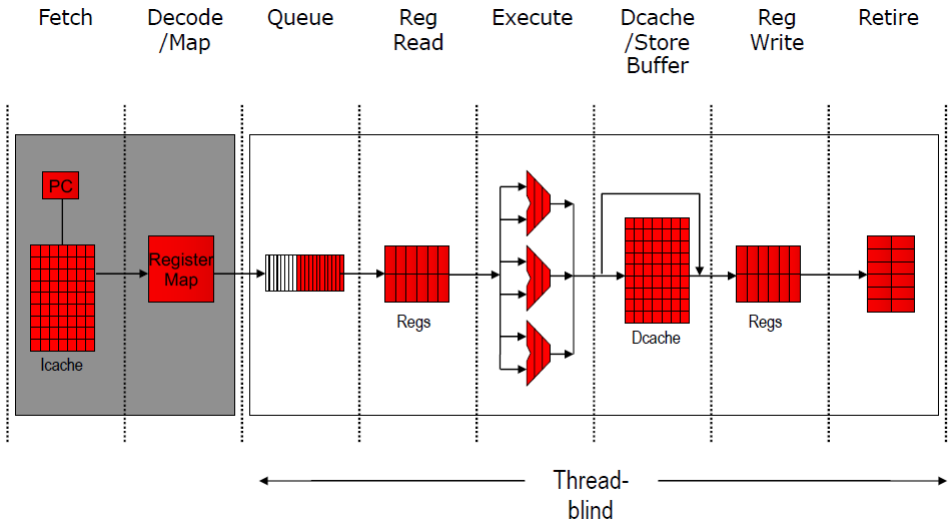
Replicated resources (front end)
- Program counter
- Register naming table
- Etc. (eg, branch predictor state)

Shared resources (back end)
- Physical register file (need more registers)
- Instruction queue (OOO scheduler)
- First and second level caches
- Execution units (memory and arithmetic)
- Etc. (eg, TLBs)

- Η εναλλαγή γίνεται χωρίς περιορισμούς με χρήση πολλών νημάτων
- Ο scheduler του επεξεργαστή superscalar, OoO πρέπει να ψάχνει σε ένα παράθυρο διαθέσιμων εντολών από όλα τα νήματα
  - Δυνατότητα για ανίχνευση εξαρτήσεων
  - Δυνατότητα για ανίχνευση προσβάσεων στη μνήμη

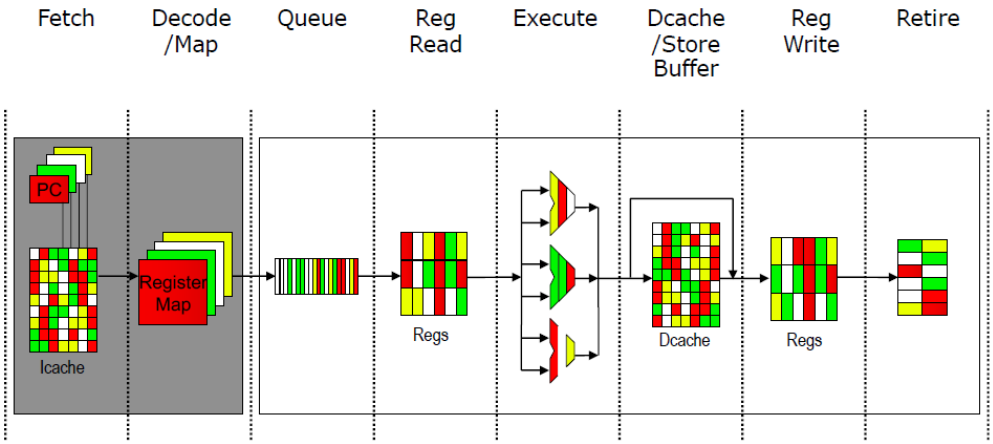
13

Βασική Αρχιτεκτονική Out-of-Order



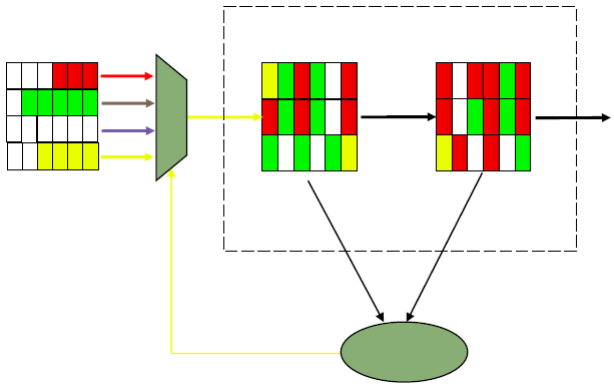
14

Pipeline Πολλαπλών Νημάτων (SMT)



15

Πολιτική Επιλογής ICOUNT



Επιλογή του νήματος με τις λιγότερες εντολές γενικά δίνει καλύτερο throughput

16

Πολλαπλά Νήματα - Συμπεράσματα

+ εκμεταλλεύεται τον παραλληλισμό που εμφανίζουν τα thread (όπως και ένας multi-core)
+ καλές επιδόσεις για ένα μοναδικό thread σε οργανώσεις SMT
+ αποδοτική οργάνωση: δεν χρειάζεται ένα επιπλέον ολόκληρο πυρήνα για ένα επιπλέον thread
+ η επικοινωνία είναι πιο γρήγορη με χρήση των κοινόχρηστων L1 caches (SAS model)

- Μειωμένη επεκτασιμότητα: για περισσότερα νήματα απαιτούνται:
  - μεγαλύτερες register files, more function units, larger issue width (and associated costs)
- Οι επιδόσεις παράλληλης εκτέλεσης υπόκεινται σε περιορισμό λόγω διαμοιρασμού του fetch bandwidth
- Εκτεταμένος διαμοιρασμός στο pipeline και στο σύστημα μνήμης μειώνει τις επιδόσεις και σε εφαρμογές single-thread και παράλληλες

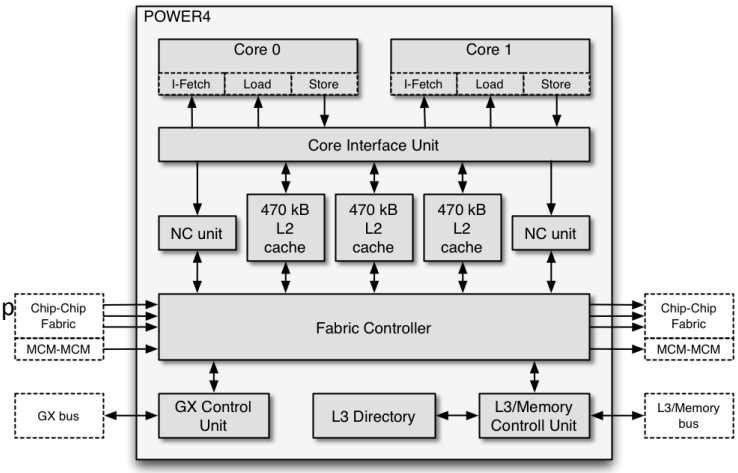
17

Παραδείγματα

18

IBM POWER4 (2001)

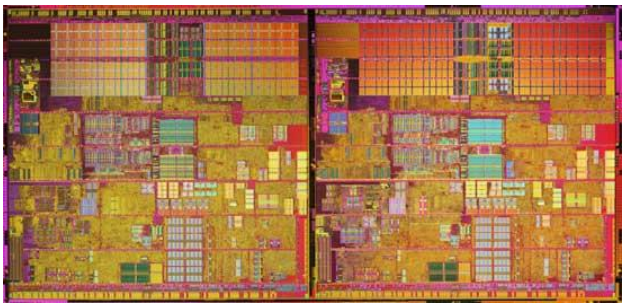
- 174M transistors @ 180nm
- 2 cores
  - 8-wide superscalar
  - Out-of-order
  - 1.3 GHz
- L2 is banked x3
- L3 control on-chip, memory off-chip
- Multi-chip module (MCM) config
  - 4 dies (8 cores) in single package
  - Shared bus “fabric”
  - 128MB combined L3



19

Intel Pentium D (2005)

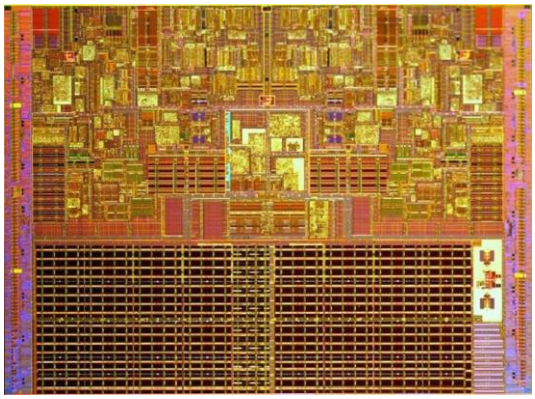
- 230M transistors @ 90nm
- Core
  - High frequency & low ILP
  - 3.8 GHz (designed for 10 GHz!!!)
  - 31-stage pipeline
  - 3-wide superscalar
  - Out-of-order
  - Trace cache
- Multi-chip module (MCM)
  - 2 dies in single package
- 2MB L2 cache
- 130 Watts!



20

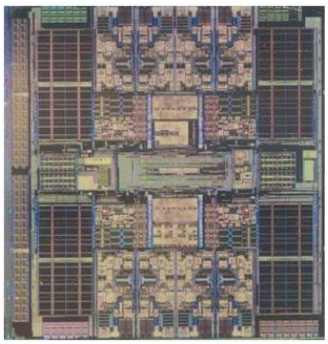
Example: Intel Core Duo (2006)

- Intel forced to use mobile designs derived from Pentium 3
- 151M transistors @ 65nm
- Core
  - 2.33 GHz
  - 4-wide issue
  - 12-stage pipeline
  - Out-of-order
- 2 cores on single die
- 2MB L2 shared cache
- 31 Watts

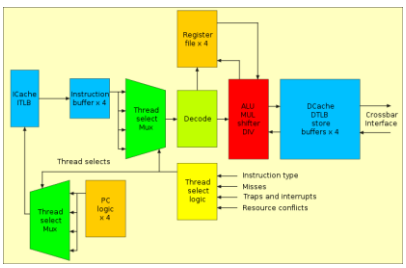


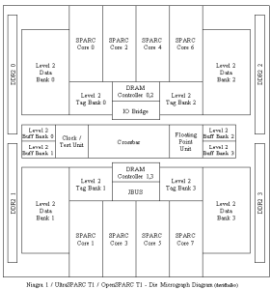
21

Example: Sun UltraSPARC T1 (2005)



- 279M transistors @ 90nm
  - 378mm^2 (!!)
- Multicore, multithreaded processor
  - 8 cores × 4 threads = 32 threads total
  - Maximize parallelism, sacrifice sequential perf.
- Core
  - 1.4 GHz
  - Fine-grain multithreading
  - In-order, simple 6-stage pipeline
- 74 Watts

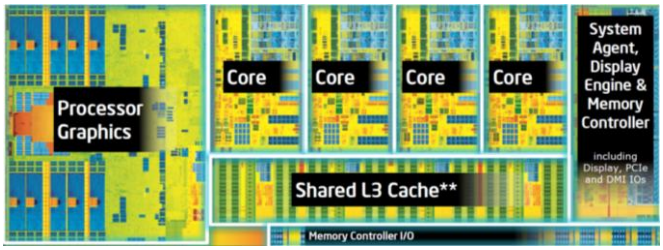




22

Example: Intel Core i7 (2013)

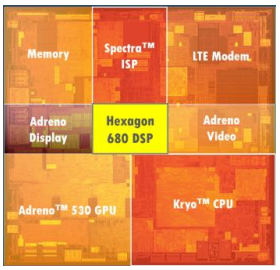
- 1.4B transistors @ 22nm
  - 177 mm^2
- Core
  - 3.5 GHz to 3.9 GHz
  - 14-stage pipelined datapath
  - 4-wide superscalar
  - 3 levels of large cache
- Four cores in single die



23

Example: Qualcomm Snapdragon 835 (2017)

- ?? Transistors @ 10nm
- ARM cores – heterogeneous "big.LITTLE" design
  - 4 "performance" cores – 2.45 GHz, 2MB L2 cache
  - 4 "efficiency" cores – 1.9 GHz, 1MB L2 cache
  - "Performance" cores are 20% faster; "efficiency" cores used 80% of the time
  - Graphics processing unit (GPU)
  - Digital signal processor (DSP)
  - Other custom accelerators (camera, modem, etc)



*Snapdragon 820 (only die shot I could find)

24