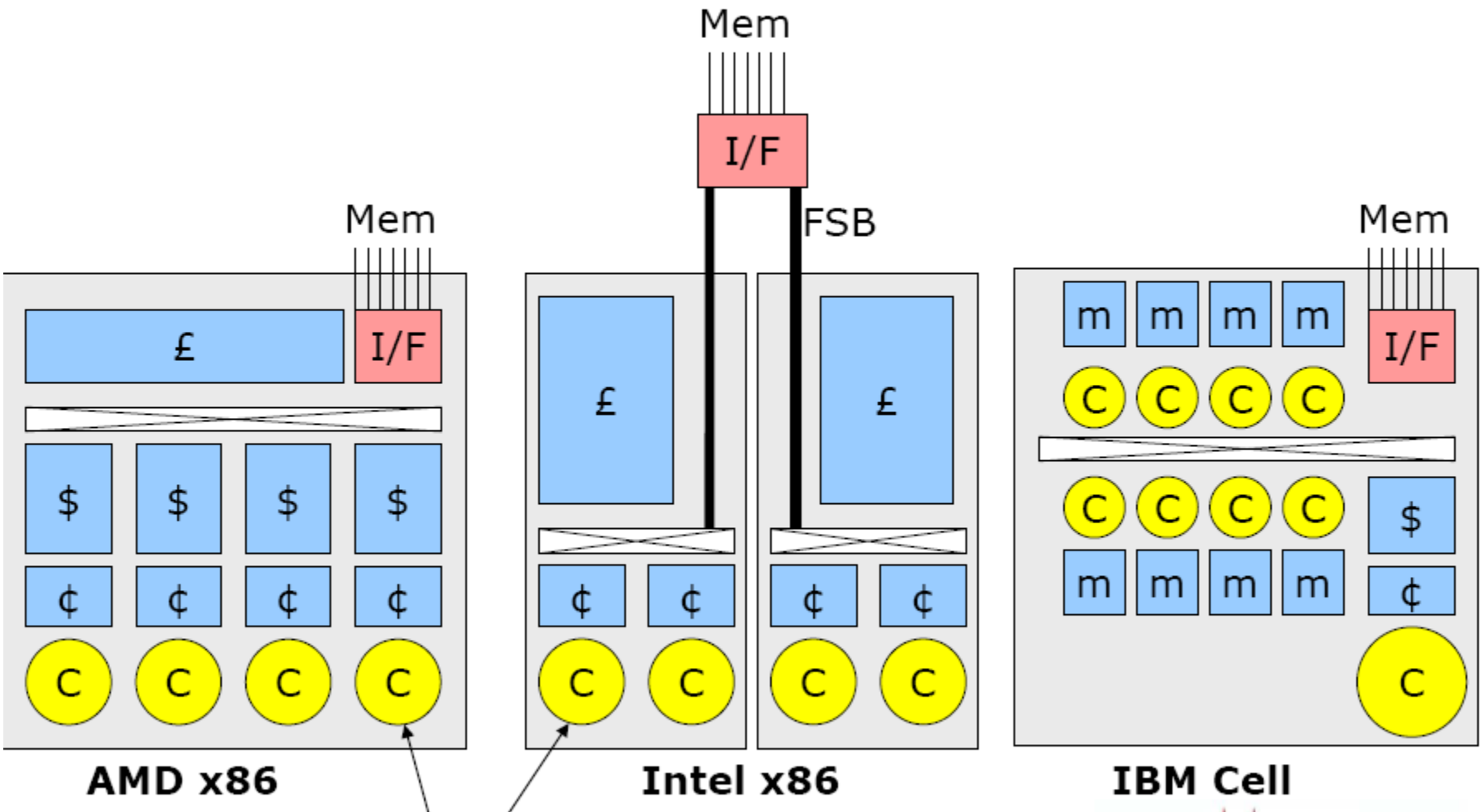
Σύγχρονες Προκλήσεις

Multi-core CPUs:

- Ageia PhysX, a multi-core physics processing unit.
- Ambric Am2045, a 336-core Massively Parallel Processor Array (MPPA)
- **AMD**
  - Athlon 64, Athlon 64 FX and Athlon 64 X2 family, dual-core desktop processors.
  - Opteron, dual- and quad-core server/workstation processors.
  - Phenom, triple- and quad-core desktop processors.
  - Sempron X2, dual-core entry level processors.
  - Turion 64 X2, dual-core laptop processors.
  - Radeon and FireStream multi-core GPU/GPGPU (10 cores, 16 5-issue wide superscalar stream processors per core)
- ARM MPCore is a fully synthesizable multicore container for ARM9 and ARM11 processor cores, intended for high-performance embedded and entertainment applications.
- Azul Systems Vega 2, a 48-core processor.
- Broadcom SiByte SB1250, SB1255 and SB1455.
- Cradle Technologies CT3400 and CT3600, both multi-core DSPs.
- Cavium Networks Octeon, a 16-core MIPS MPU.
- HP PA-8800 and PA-8900, dual core PA-RISC processors.
- **IBM**
  - POWER4, the world's first dual-core processor, released in 2001.
  - POWER5, a dual-core processor, released in 2004.
  - POWER6, a dual-core processor, released in 2007.
  - PowerPC 970MP, a dual-core processor, used in the Apple Power Mac G5.
  - Xenon, a triple-core, SMT-capable, PowerPC microprocessor used in the Microsoft Xbox 360 game console.
- IBM, Sony, and Toshiba Cell processor, a nine-core processor with one general purpose PowerPC core and eight specialized SPUs (Synergystic Processing Unit) optimized for vector operations used in the Sony PlayStation 3.
- Infineon Danube, a dual-core, MIPS-based, home gateway processor.
- **Intel**
  - Celeron Dual Core, the first dual-core processor for the budget/entry-level market.
  - Core Duo, a dual-core processor.
  - Core 2 Duo, a dual-core processor.
  - Core 2 Quad, a quad-core processor.
  - Core i7, a quad-core processor, the successor of the Core 2 Duo and the Core 2 Quad.
  - Itanium 2, a dual-core processor.
  - Pentium D, a dual-core processor.
  - Teraflops Research Chip (Polaris), an 3.16 GHz, 80-core processor prototype, which the company says will be released within the next five years[6].
  - Xeon dual-, quad- and hexa-core processors.
- IntellaSys seaForth24, a 24-core processor.
- **Nvidia**
  - GeForce 9 multi-core GPU (8 cores, 16 scalar stream processors per core)
  - GeForce 200 multi-core GPU (10 cores, 24 scalar stream processors per core)
  - Tesla multi-core GPGPU (8 cores, 16 scalar stream processors per core)
- Parallax Propeller P8X32, an eight-core microcontroller.
- picoChip PC200 series 200-300 cores per device for DSP & wireless
- Rapport Kilocore KC256, a 257-core microcontroller with a PowerPC core and 256 8-bit "processing elements".
- Raza Microelectronics XLR, an eight-core MIPS MPU
- **Sun Microsystems**
  - UltraSPARC IV and UltraSPARC IV+, dual-core processors.
  - UltraSPARC T1, an eight-core, 32-thread processor.
  - UltraSPARC T2, an eight-core, 64-concurrent-threa
- Texas Instruments TMS320C80 MVP, a five-core multimedi
- Tilera TILE64, a 64-core processor
- XMOS Software Defined Silicon quad-core XS1-G4

[source: Wikipedia]

Many shapes and forms...



May run more than one thread...

Εισαγωγή

- Οι CMP είναι πια πραγματικότητα
  - Intel Core 2, Quad, Nehalem
  - IBM Power5, Power6
  - Sun Niagara, Niagara2, Rock
  - Sony Cell
- ΄Υπαρξη πολλών threads – Τί τα κάνουμε;
  - Multithreaded environment
  - Multiprogrammed environment
- Νέες προκλήσεις

Homogeneous vs. Heterogeneous

- Τί είδους cores χρησιμοποιούμε;
- Homogeneous Systems
  - Χρήση πολλαπλών ίδιων πυρήνων
  - Ευκολία στο design
- Heterogeneous Systems
  - Χρήση πυρήνων με τελείως διαφορετικό ISA
  - Χρήση πυρήνων με “παρόμοιο” instruction set
  - Πολύπλοκο design
  - Πολύπλοκο scheduling
  - Καλύτερο ισοζύγιο απόδοσης – κατανάλωσης ενέργειας

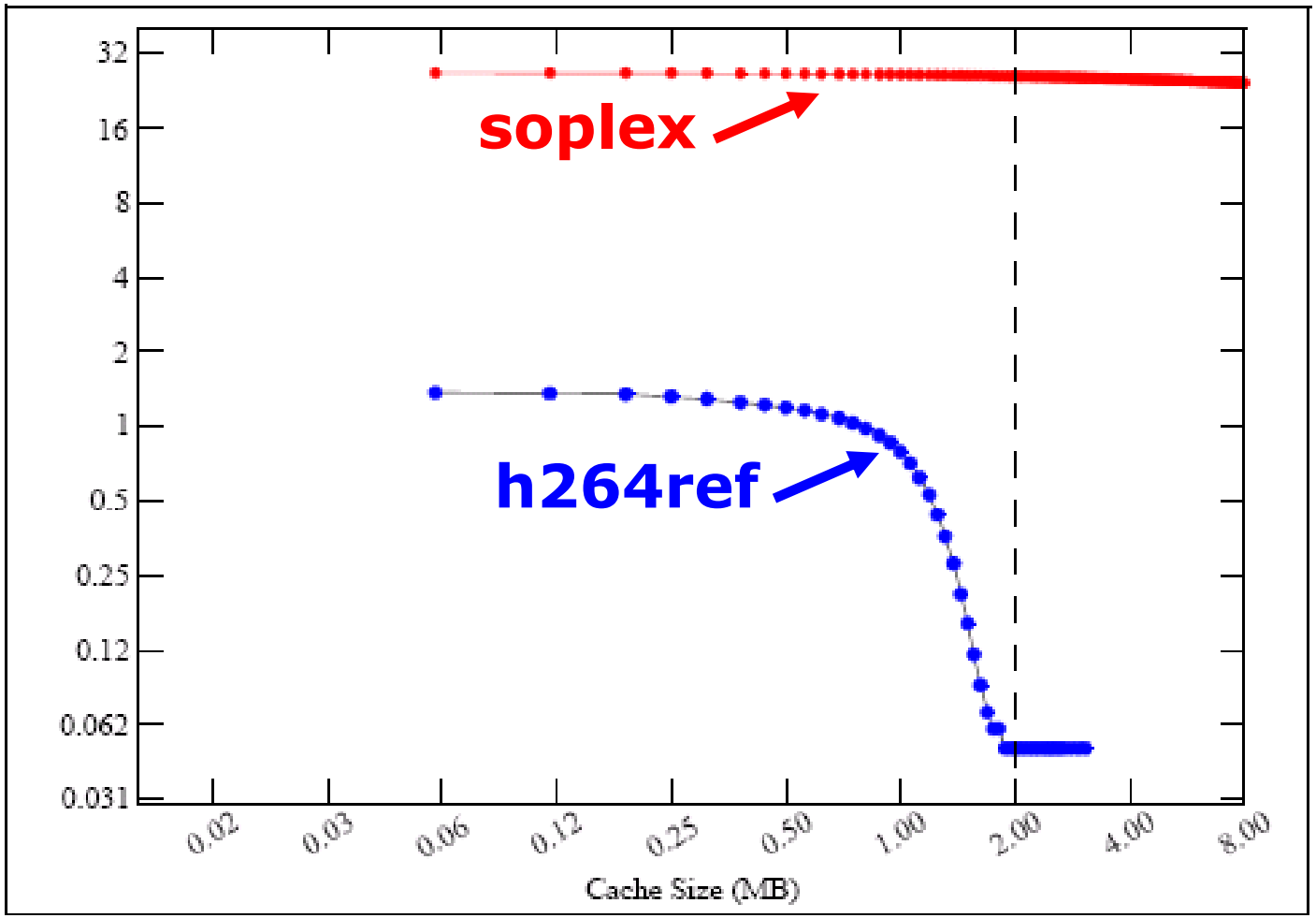
Ιεραρχία Μνήμης

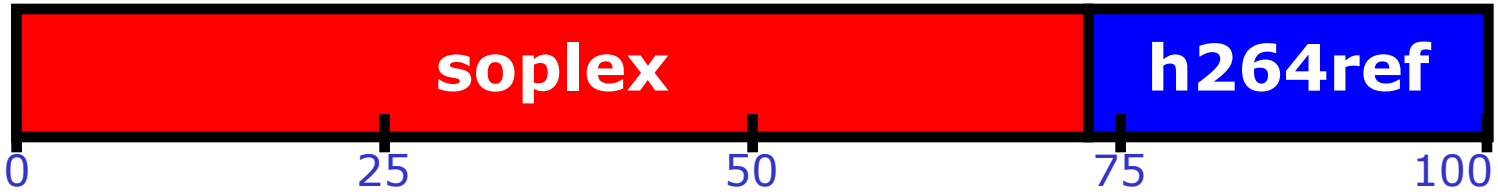
- Πολλές σχεδιαστικές επιλογές έχουν μεταφερθεί από το πεδίο των single-processor systems.
- Στους CMP λόγω των παράλληλων threads έχουμε πολύ μεγαλύτερη πίεση στην ιεραρχία μνήμης.
  - Παράλληλα threads που δουλεύουν στο ίδιο data set (π.χ. databases)
  - Παράλληλα threads που δουλεύουν σε ανεξάρτητα data sets(π.χ. servers)
  - Διαφορετικές εφαρμογές που εκτελούνται παράλληλα
- Ανάγκη επαναξιολόγησης των λύσεων που είχαν προταθεί παλιότερα.
  - Shared or private levels
  - Πολιτικές αντικατάστασης
  - Coherency protocols

Cache Partitioning

- Η LRU θεωρείται η καλύτερη πολιτική αντικατάστασης.
  - Χρησιμοποιείται στα περισσότερα συστήματα (ή προσεγγίσεις αυτής)
  - Έχει μεταφερθεί και στους CMPs
  - Κατανέμει την cache με βάση τη ζήτηση (demand)
- Πρόβλημα : thread-blind
  - Μπορεί να οδηγήσει σε starvation κάποιο thread (π.χ. αν ένα από τα threads εκτελεί ένα streaming application)
- Λύση : cache partitioning
  - Κάθε thread μπορεί να χρησιμοποιήσει ένα συγκεκριμένο κομμάτι/ποσοστό της cache
  - Καλύτερο utilization, βελτίωση απόδοσης
  - QoS

Cache Partitioning

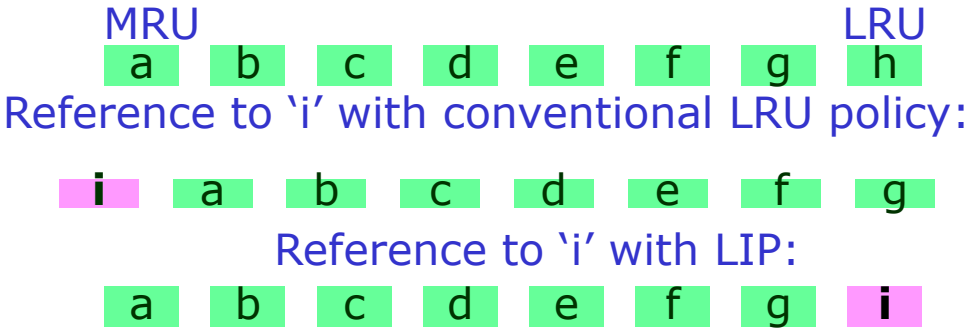




Cache Occupancy Under LRU Replacement
(2MB Shared Cache)

Cache Replacement Policies

- Victim Selection
  - Ποιο block θα αντικατασταθεί (LRU, Random κτλ)
- Insertion Policy
  - Ποια η ‘‘προτεραιότητα’’ του νέου block (π.χ. MRU) ?
- ISCA 2007
  - MRU policy
  - LRU policy
  - Bimodal Insertion Policy (BIP)

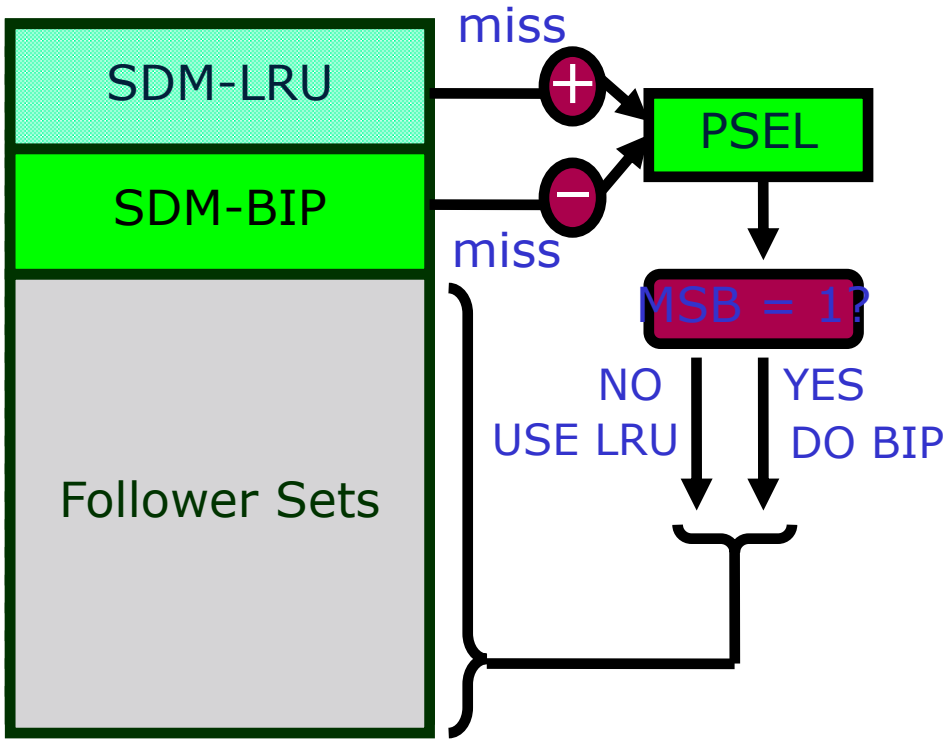


Reference to ‘i’ with BIP:

if(rand() < β)
Insert at MRU postion
else
Insert at LRU position

Dynamic Insertion Policy

- Set Dueling Monitors
- HW overhead
  - 10 bits
  - Combinational logic
- Επέκταση για shared caches
  - TADIP

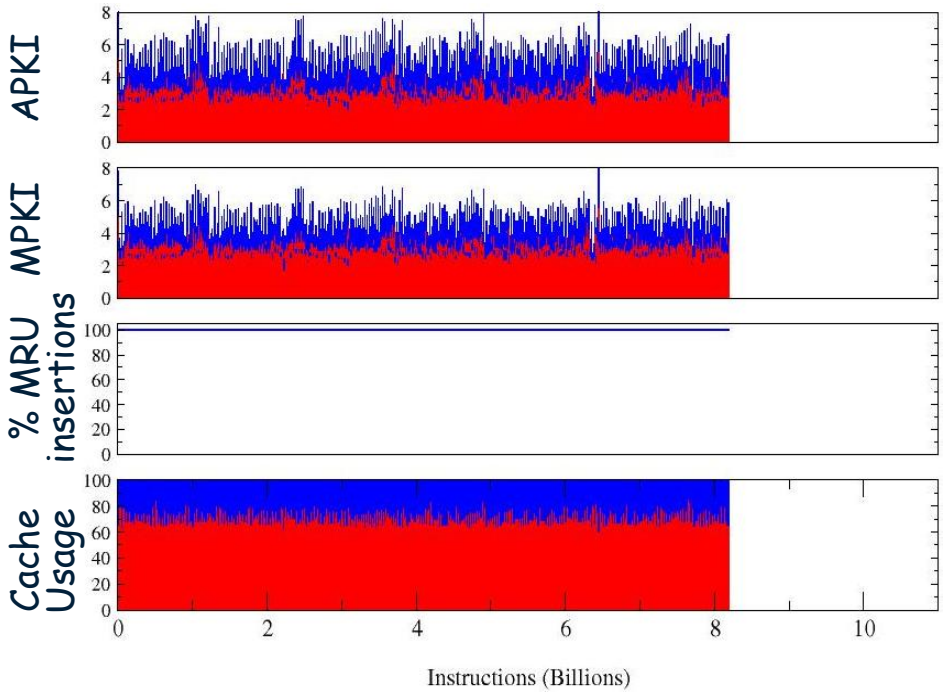


- Based on Analytical and Empirical Studies:
  32 Sets per SDM
  10 bit PSEL counter

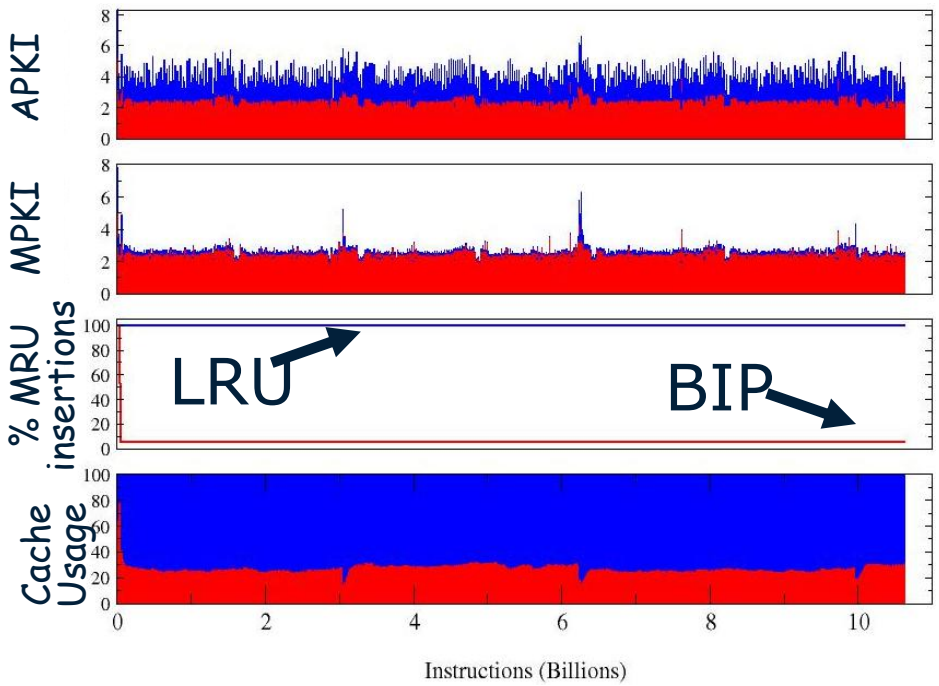
TADIP

SOPLEX

H264REF

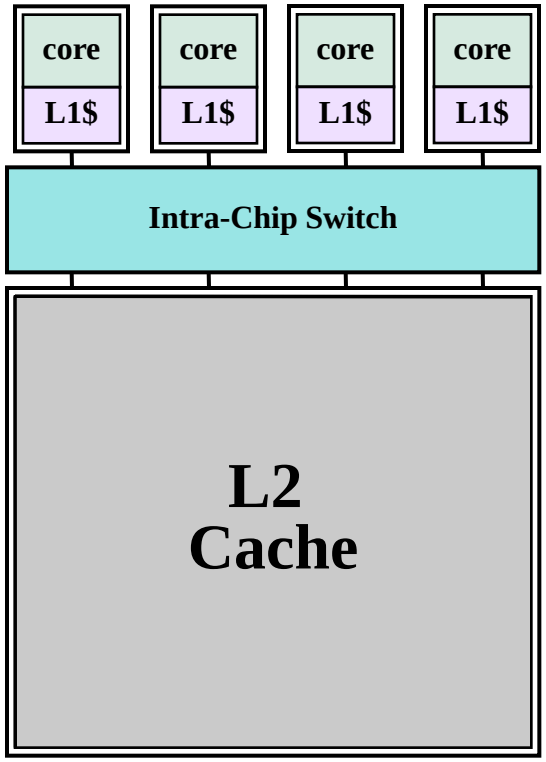


Baseline LRU Policy / DIP



TADIP

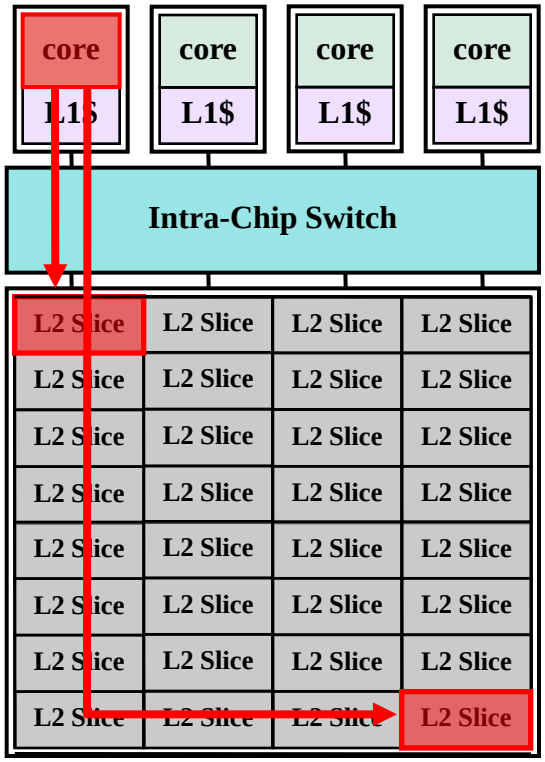
Non-Uniform Cache Access Latency (1)



“Dance-Hall” Layout

- Οι caches σχεδιάζονται με (μεγάλες) uniform acess latency
  - Best Latency = Worst Latency !!!
- Μικρές και γρήγορες L1
- Μεγάλη και αργή L2

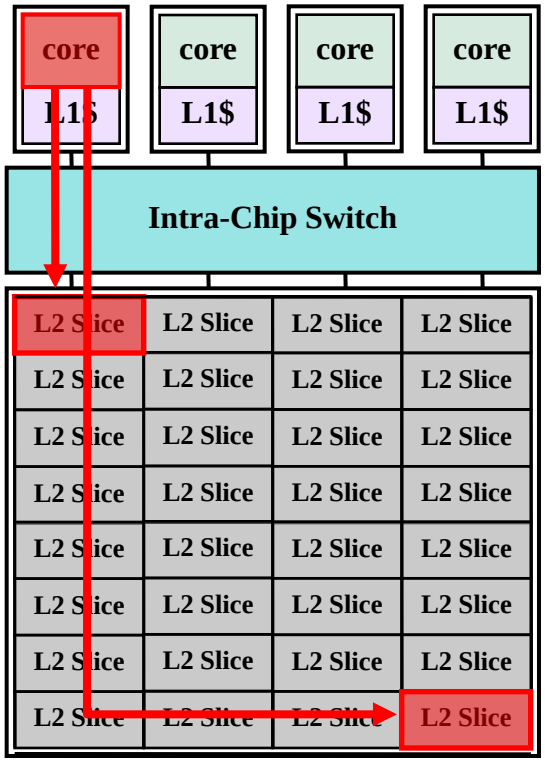
Non-Uniform Cache Access Latency (2)



“Dance-Hall” Layout

- Σπάσιμο της L2 σε μικρά κομμάτια για να μειωθεί ο χρόνος πρόσβασης και η κατανάλωση ενέργειας
- Best Latency < Worst Latency
- Στόχος :
  - Average Latency → Best Latency

Non-Uniform Cache Access Latency (3)



"Dance-Hall" Layout

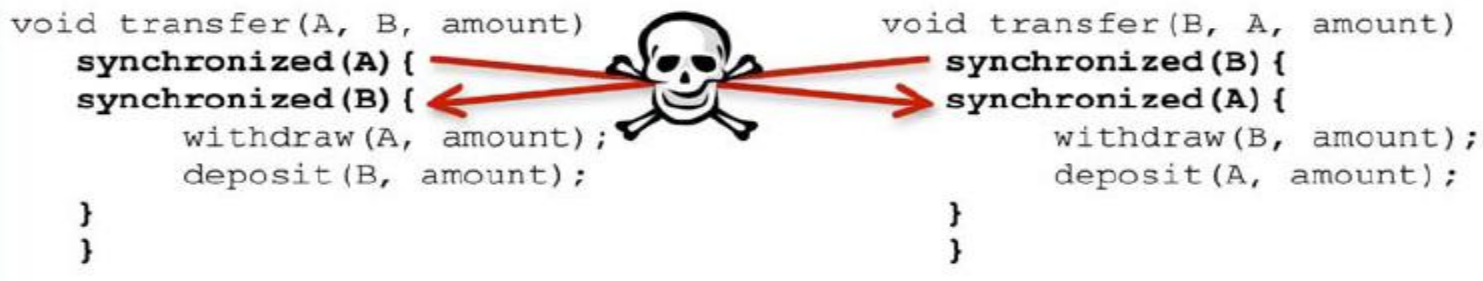
- Προκλήσεις :
  - Private vs Shared
  - Data Placement
  - Data Migration
  - Efficient search

Parallel Programming

- Μέχρι σήμερα :
  - Για όσους έχουν πρόσβαση σε μεγάλα παράλληλα συστήματα
  - Χρήση γλωσσών με low-level concurrency features
  - Δύσκολο να το γράψεις
  - Δύσκολο debugging
  - Δύσκολη η διατήρηση και η επέκταση του κώδικα
  - Δύσκολο να πετύχεις speedups!
- Σήμερα ο καθένας έχει ένα CMP
  - Πώς γράφουμε αποδοτικό παράλληλο κώδικα;

Locks

- Χρήση locks (enforce atomicity)
- Απαισιόδοξη θεώρηση των πραγμάτων
- Coarse grain
  - Εύκολη υλοποίηση
  - Περιορισμός του εκμεταλλεύσιμου παραλληλισμού
- Fine grain
  - Πολύπλοκη υλοποίηση (πολλαπλά locks για τα επιμέρους στοιχεία μιας δομής)
  - Περισσότερος παραλληλισμός, αλλά μεγαλύτερο overhead
- Πολύ δύσκολη η υπέρθεση των locks (composability)
- Ο προγραμματιστής ορίζει τι θα γίνει και πως



Transactional Memory (1)

- Ο προγραμματιστής ορίζει atomic sections και το σύστημα αναλαμβάνει να τα υλοποιήσει.

```
void deposit(account, amount){
   lock(account);
     int t = bank.get(account);
     t = t + amount;
     bank.put(account, t);
   unlock(account);
}
```

```
void deposit(account, amount){
   atomic {
     int t = bank.get(account);
     t = t + amount;
     bank.put(account, t);
   }
}
```

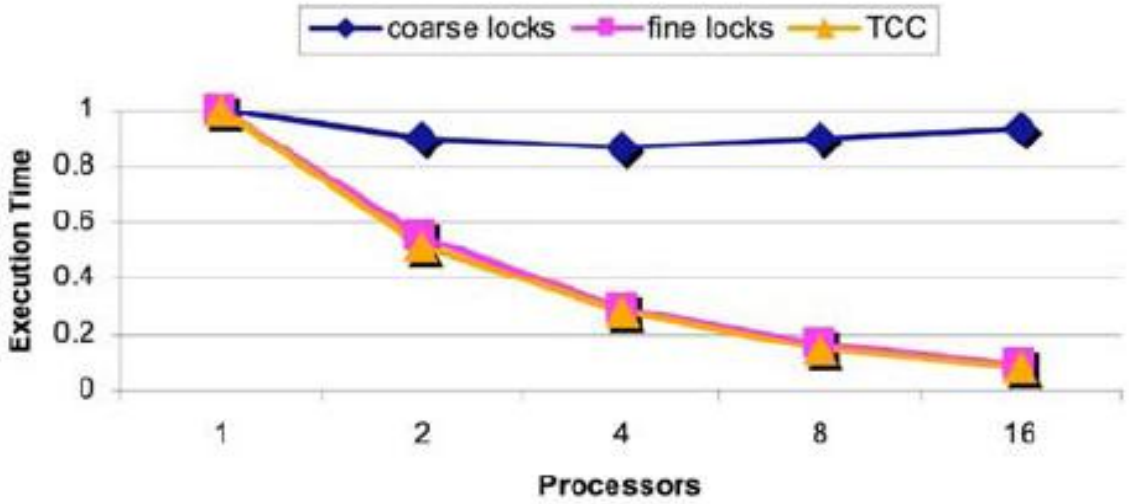
- Εμπνευσμένο από τις βάσεις δεδομένων
- Αισιόδοξη θεώρηση των πραγμάτων
- Αποδίδει τόσο καλά όσο και το fine-grain locking
- Composability

```
void transfer(A, B, amount)
   synchronized(bank){
        withdraw(A, amount);
        deposit(B, amount);
   }
```

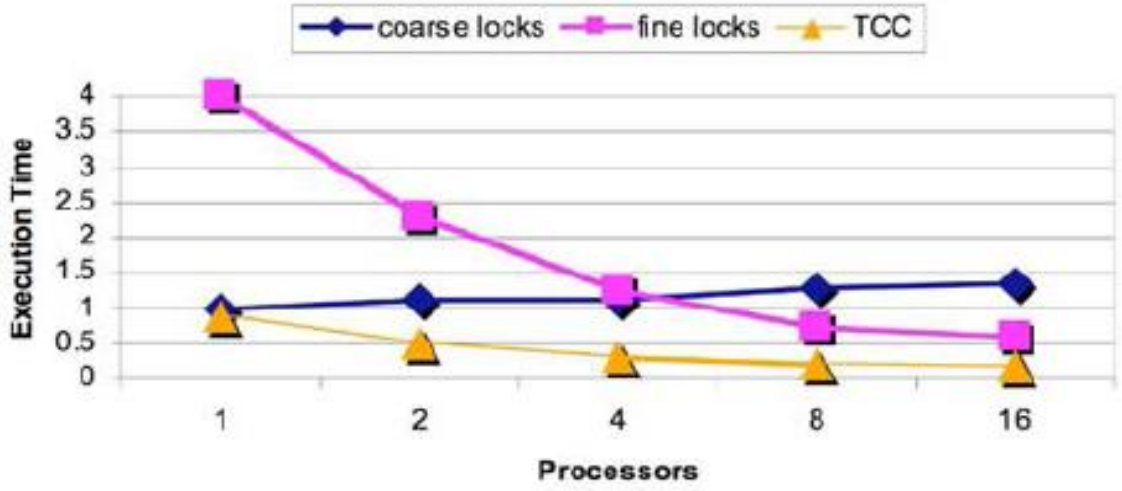
```
void transfer(C, D, amount)
   synchronized(bank){
        withdraw(C, amount);
        deposit(A, amount);
   }
```

Transactional Memory vs Locks

HashMap

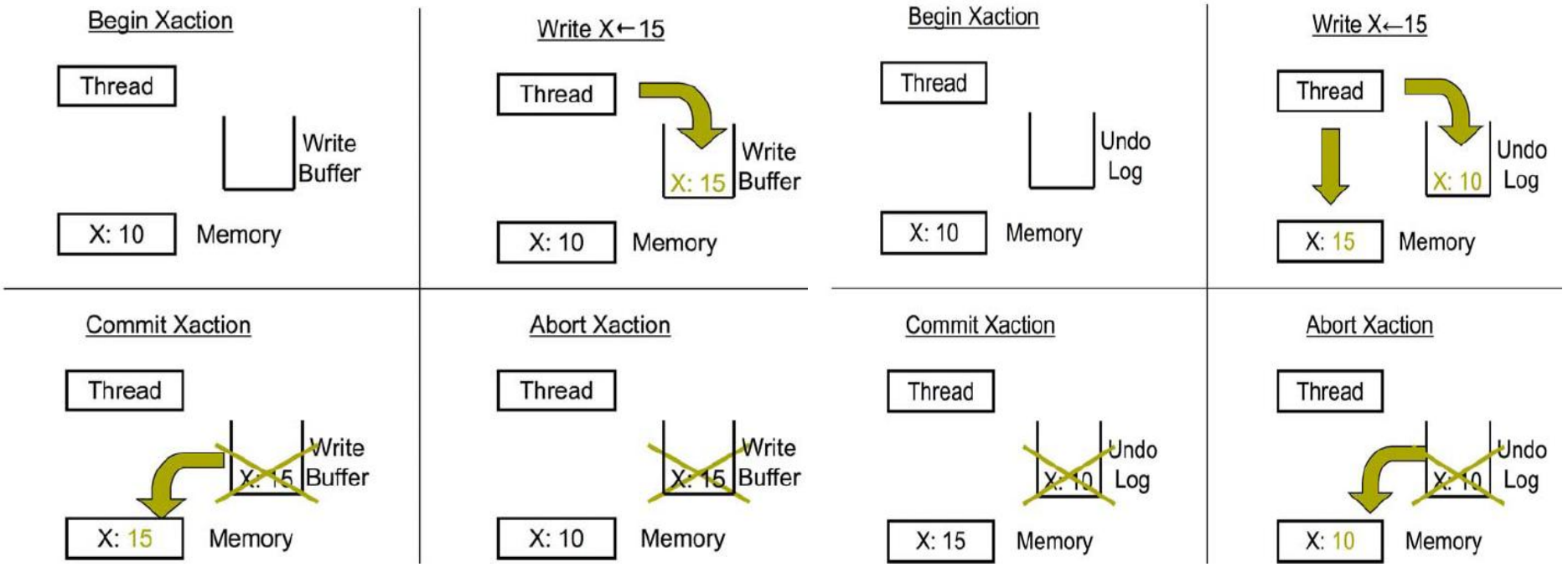


Balanced Tree



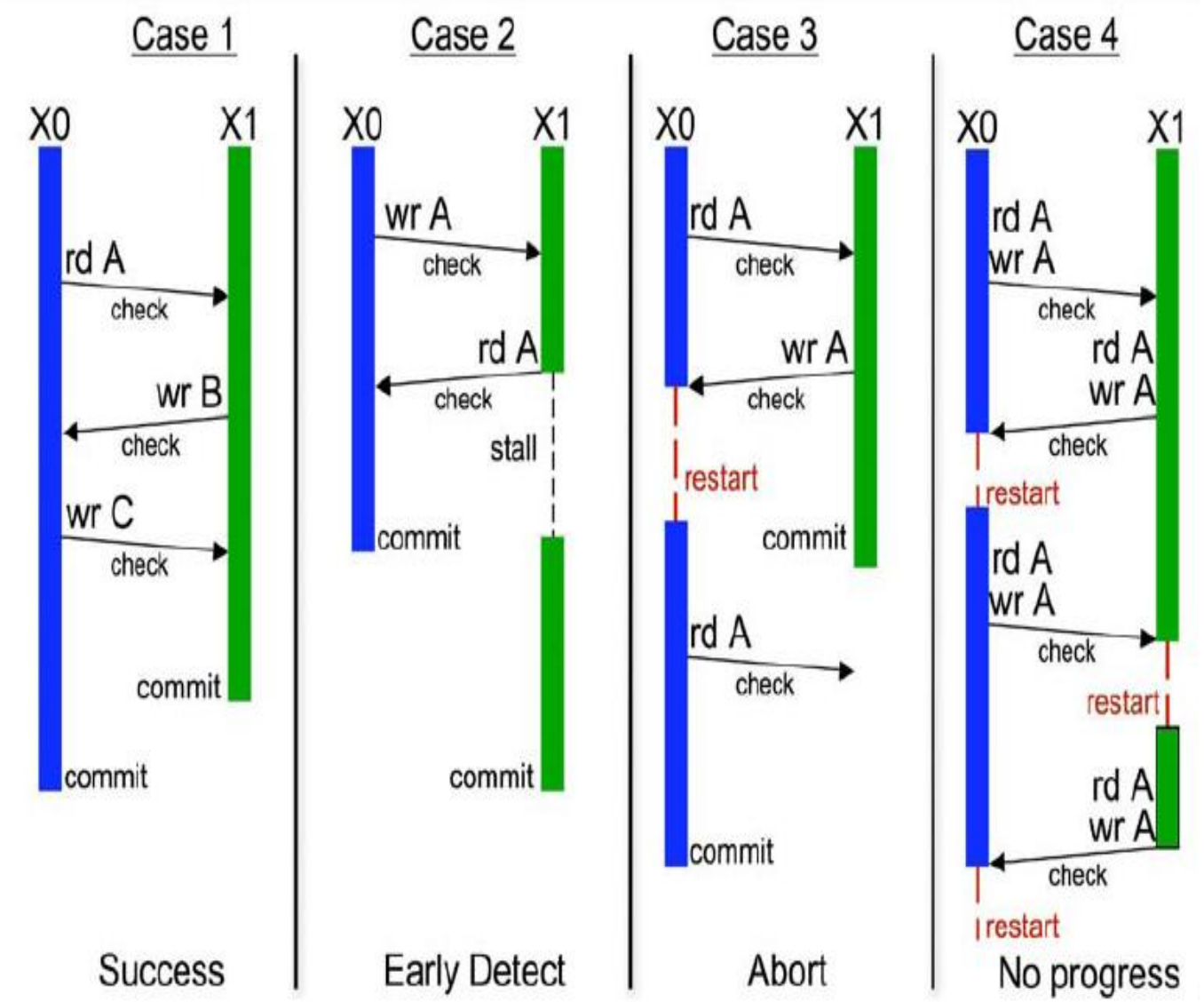
Transactional Memory(2)

- STM vs HTM (ή Hybrid TM)
- Data versioning
  - Lazy vs Eager



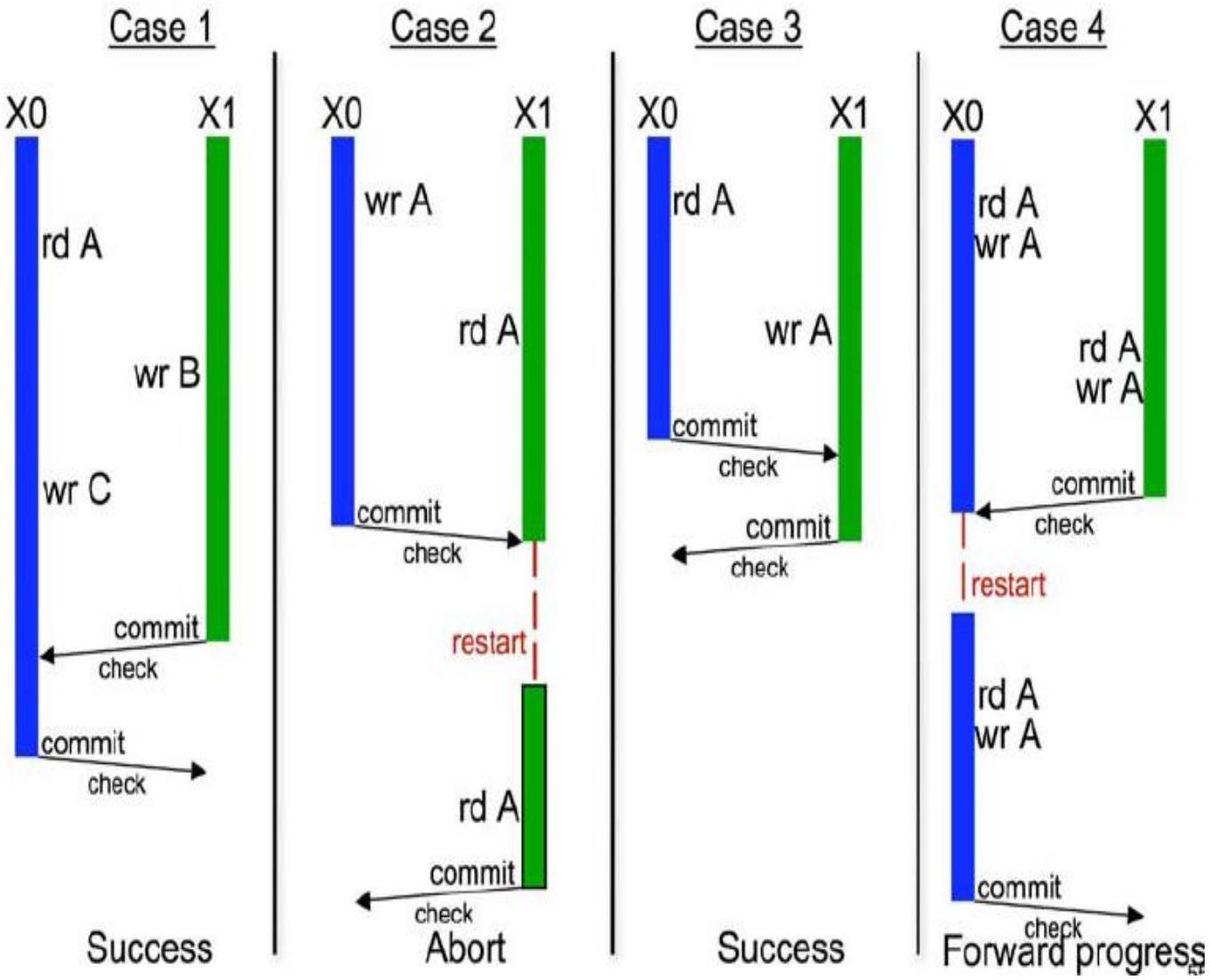
Transactional Memory(3)

- Conflict detection
- Pessimistic (Eager)
  - Εντοπισμός των conflicts νωρίς
  - Λιγότερη χαμένη δουλειά
  - Δεν εγγυάται forward progress



Transactional Memory(4)

- Conflict detection
- Optimistic (Lazy)
  - Εντοπισμός των conflicts στο τέλος
  - Fairness problems
  - Εγγυάται forward progress



TM Implementation Space

- Hardware TM Systems
  - Lazy + optimistic : Stanford TCC
  - Lazy + pessimistic : MIT LTM, Intel VTM
  - Eager + pessimistic : Winsconsin LogTM
- Software TM Systems
  - Lazy + optimistic (rd/wr) : Sun TL2
  - Lazy + optimistic (rd) / pessimistic (wr) : MS OSTM
  - Eager + optimistic (rd) /pessimistic (wr) : Intel STM
  - Eager + pessimistic (rd/wr) : Intel STM
- Optimal design ??????