CS305: Computer Architecture

Hyperthreading

https://www.cse.iitb.ac.in/~biswa/courses/CS305/main.html

https://www.cse.iitb.ac.in/~biswa/

Beyond ILP

- There can be much higher natural parallelism in some applications
  (e.g., Database or Scientific codes)

- Explicit Thread Level Parallelism or Data Level Parallelism

- Thread: instruction stream with own PC and data
  - thread may be a process part of a parallel program of multiple processes, or it may be an independent program
  - Each thread has all the state (instructions, data, PC, register state, and so on) necessary to allow it to execute

- Data Level Parallelism: Perform identical operations on data, and lots of data

Computer Architecture

TLP: Thread Level Parallelism

- ILP exploits implicit parallel operations within a loop or straight-line code segment

- TLP explicitly represented by the use of multiple threads of execution that are inherently parallel

- Goal: Use multiple instruction streams to improve
  1. Throughput of computers that run many programs
  2. Execution time of multi-threaded programs

- TLP could be more cost-effective to exploit than ILP

Computer Architecture   3

Multithreading (Hardware)

- Multithreading: multiple threads to share the functional units of 1 processor via overlapping
  - processor must duplicate independent state of each thread e.g., a separate copy of register file, a separate PC, and for running independent programs, a separate page table
  - memory shared through the virtual memory mechanisms, which already support multiple processes
  - HW for fast thread switch; much faster than full process switch ≈ 100s to 1000s of clocks

- When switch?
  - Alternate instruction per thread (fine grain)
  - When a thread is stalled, perhaps for a cache miss, another thread can be executed (coarse grain)

Fine grained

- Switches between threads on each instruction, causing the execution of multiples threads to be interleaved

- Usually done in a round-robin fashion, skipping any stalled threads

- CPU must be able to switch threads every clock

- Advantage is it can hide both short and long stalls, since instructions from other threads executed when one thread stalls

- Disadvantage is it slows down execution of individual threads, since a thread ready to execute without stalls will be delayed by instructions from other threads

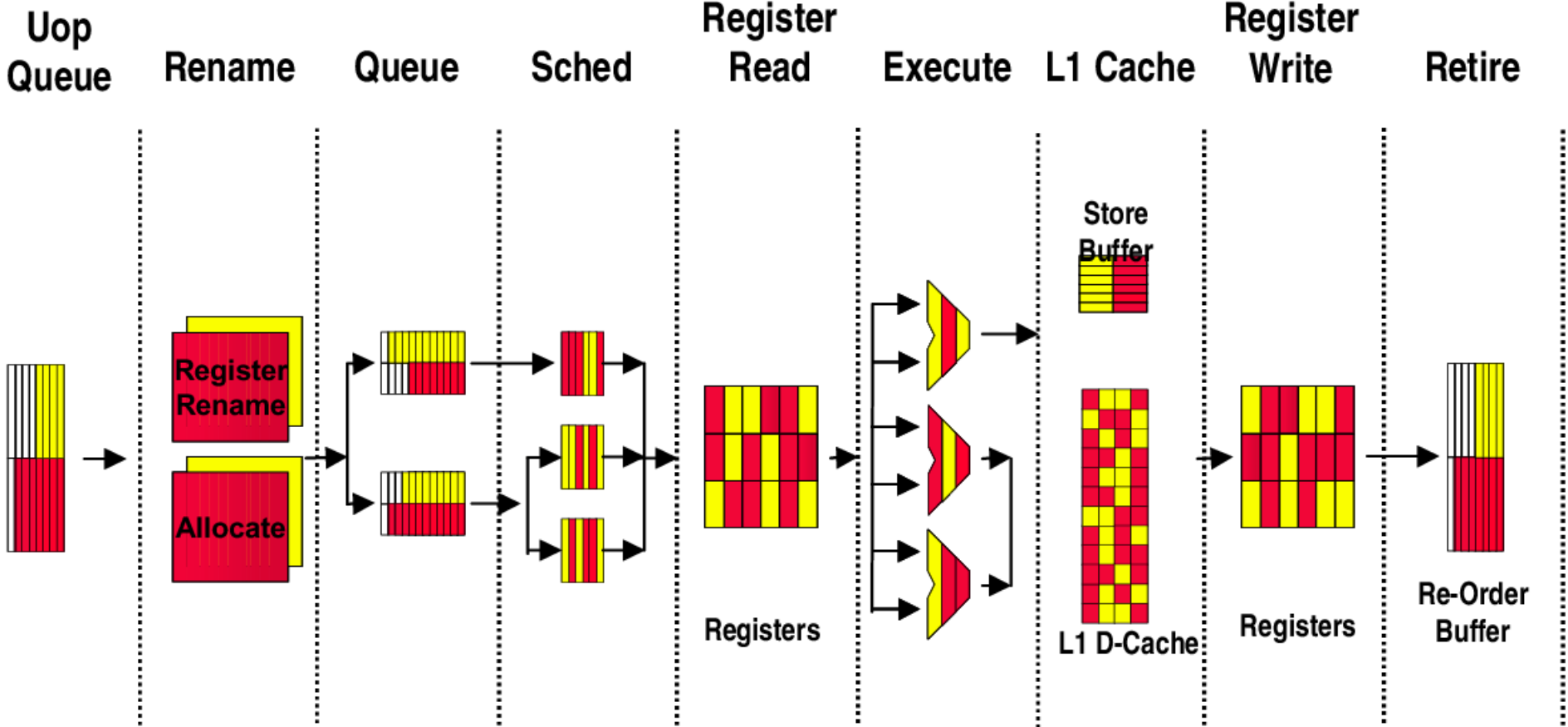Computer Architecture   5

Coarse grained

- Switches threads only on costly stalls, such as L2 cache misses

- Advantages
  - Relieves need to have very fast thread-switching
  - Doesn't slow down thread, since instructions from other threads issued only when the thread encounters a costly stall

- Disadvantage is hard to overcome throughput losses from shorter stalls, due to pipeline start-up costs
  - Since CPU issues instructions from 1 thread, when a stall occurs, the pipeline must be emptied or frozen
  - New thread must fill pipeline before instructions can complete

- Because of this start-up overhead, coarse-grained multithreading is better for reducing penalty of high cost stalls, where pipeline refill << stall time

Computer Architecture 6

SMT (Simultaneous Multithreading, hyper-threading)

- Simultaneous multithreading (SMT): insight that dynamically scheduled processor already has many HW mechanisms to support multithreading

  – Large set of virtual registers that can be used to hold the register sets of independent threads

  – Register renaming provides unique register identifiers, so instructions from multiple threads can be mixed in data path without confusing sources and destinations across threads

  – Out-of-order completion allows the threads to execute out of order, and get better utilization of the HW

- Just adding a per thread renaming table and keeping separate PCs

  – Independent commitment can be supported by logically keeping a separate reorder buffer for each thread

Computer Architecture

Intel SMT

ਧੰਨਵਾਦ