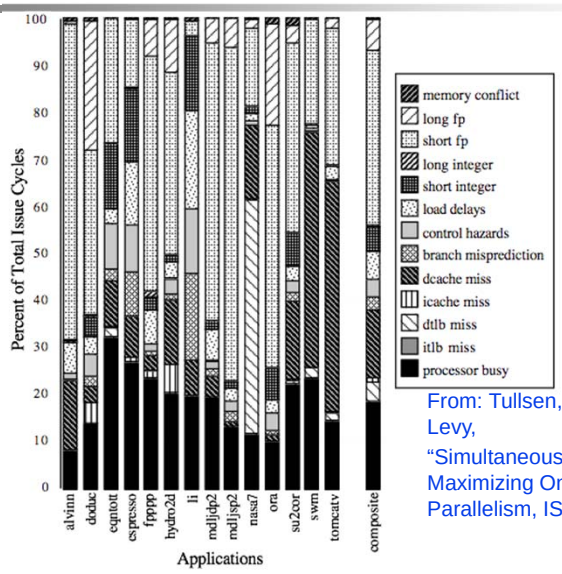
Thread level parallelism

- ILP is used in straight line code or loops
- Cache miss (off-chip cache and main memory) is unlikely to be hidden using ILP.
- Thread level parallelism is used instead.
- Thread: process with own instructions 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

Copyright © 2012, Elsevier Inc. All rights reserved.



From: Tullsen, Eggers, and Levy,

"Simultaneous Multithreading: Maximizing On-chip Parallelism, ISCA 1995.

Copyright © 2012, Elsevier Inc. All rights reserved.

Thread Level parallelism

- 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)

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved.

Fine-Grained Multithreading

- 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
- Used on Sun's T1

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved.

Coarse-Grained Multithreading

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

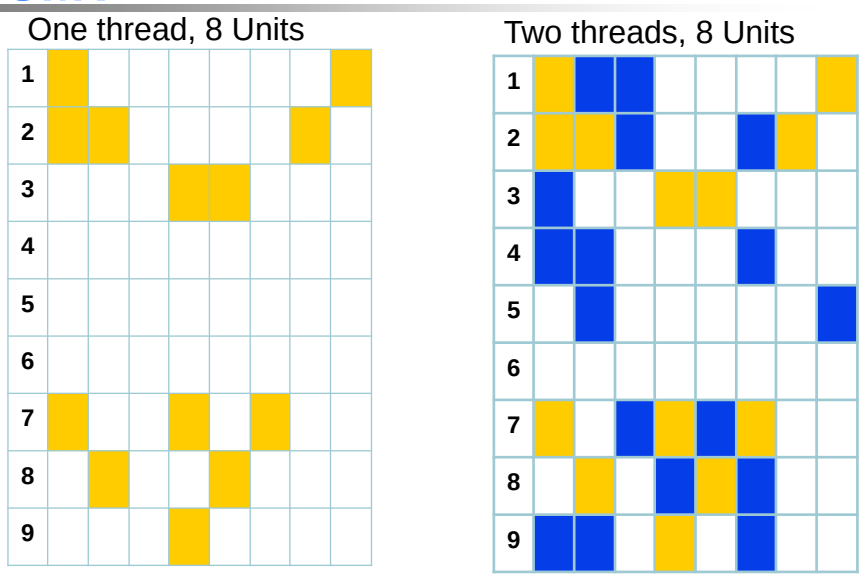
Copyright © 2012, Elsevier Inc. All rights reserved.

Simultanuous Multithreading SMT

- Fine-grained multithreading implemented on top of multiple-issued dynamically scheduled processor.
- Multiple instructions from different threads.

Copyright © 2012, Elsevier Inc. All rights reserved.

SMT



Copyright © 2012, Elsevier Inc. All rights reserved.

Multithreading

Copyright © 2012, Elsevier Inc. All rights reserved.

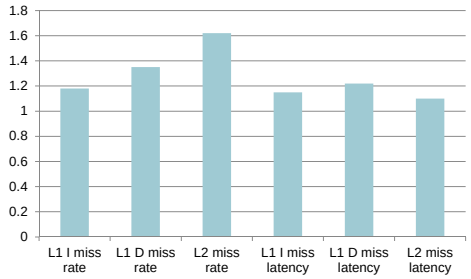
Sun T1

- Focused on TLP rather than ILP
- Fine-grained multithreading
- 8 cores, 4 threads per core, one shared FP unit.
- 6-stage pipeline (similar to MIPS with one stage for thread switching)
- L1 caches: 16KB I, 8KB D, 64-byte block size (misses to L2 23 cycles with no contention)
- L2 caches: 4 separate L2 caches each 750KB. Misses to main memory 110 cycles assuming no contention

Copyright © 2012, Elsevier Inc. All rights reserved.

Sun T1

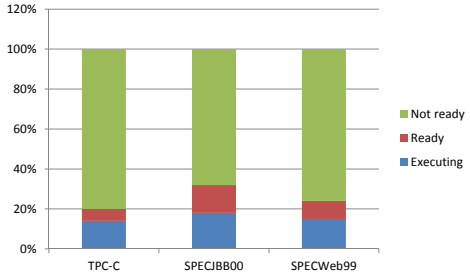
- Relative change in the miss rate and latency when executing one thread per core vs 4 threads per core (TPC-C)



Copyright © 2012, Elsevier Inc. All rights reserved.

Sun T1

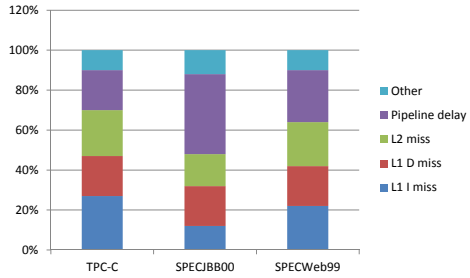
- Breakdown of the status on an average thread. Ready means the thread is ready, but another one is chosen – The core stalls only if all the 4 threads are not ready



Copyright © 2012, Elsevier Inc. All rights reserved.

Sun t1

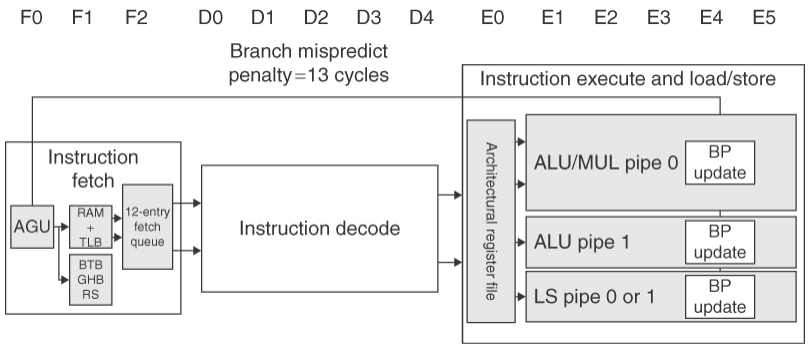
- Breakdown of the causes for a thread being not ready



Copyright © 2012, Elsevier Inc. All rights reserved.

The ARM Cortex-A8

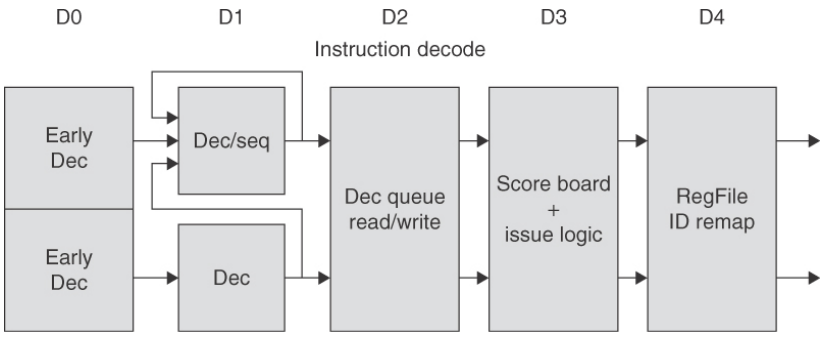
- Dual issue processor 13-stage pipeline



Copyright © 2012, Elsevier Inc. All rights reserved.

The ARM Cortex-A8

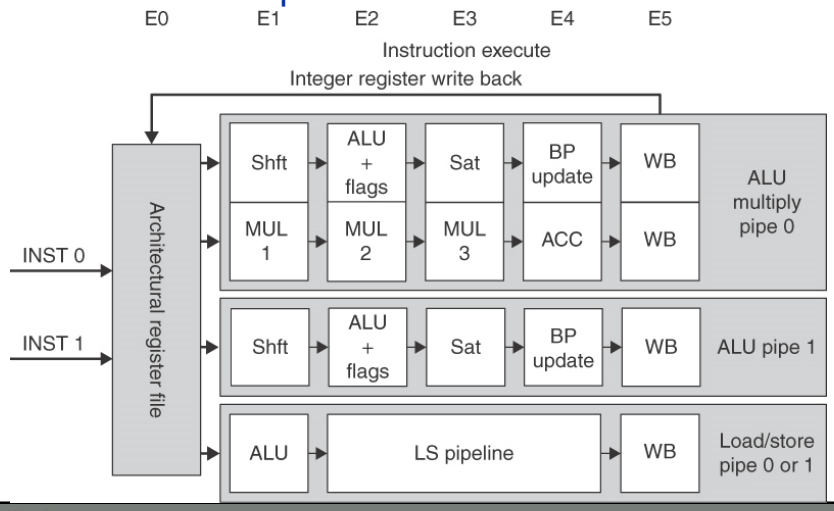
- Five stage instruction decode



Copyright © 2012, Elsevier Inc. All rights reserved.

ARM Cortex-A8

- Execution Pipeline



Copyright © 2012, Elsevier Inc. All rights reserved.