Multithreading

Superscalar and VLIW systems are based on ILP

- Instruction level parallelism
- The processor fetches and executes bundles of instructions
- Hardware executes as many instructions as possible in parallel
- Compilers combine instructions into long words on VLIW systems

Thread level parallelism overlaps streams of instructions

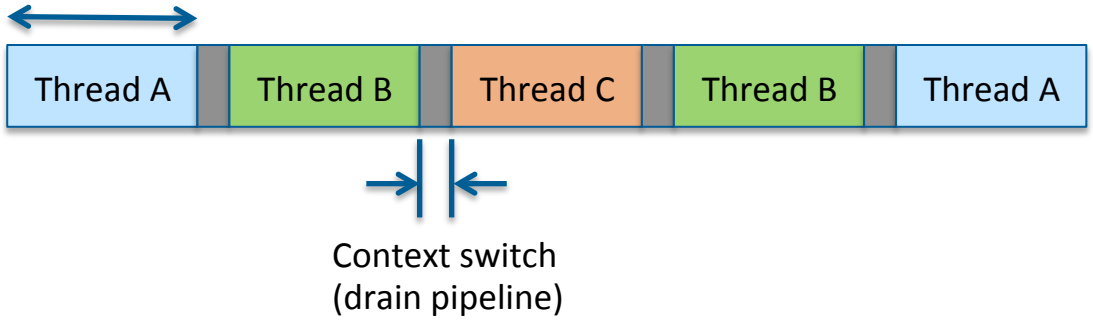
- Processes or tasks are split into separate threads
- When one thread stalls, control is switch to another
- This hides latencies due to cache misses or I/O
- All threads appear to be running at the same time

Tasks versus Threads

- Tasks & threads differ in scale or granularity
  - Tasks involve longer streams of instructions than do threads
    - Multitasking involves context switches and more overhead
    - Must complete pending instructions, save registers & flush cache
    - Triggered, for example, by a page fault
- Thread switching is more efficient
  - Hardware has multiple register sets
    - no saving or restoring is needed when switching
- Multithreading can be coarse-grained or fine-grained

Coarse-grained

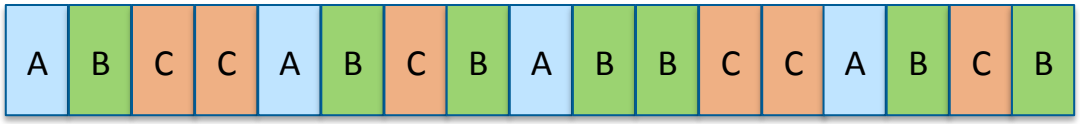
- Switch occurs after a group of instructions are executed
  - Pipeline is drained each time a switch takes place



Expiration of a time slice or having to wait for I/O to complete trigger the switching

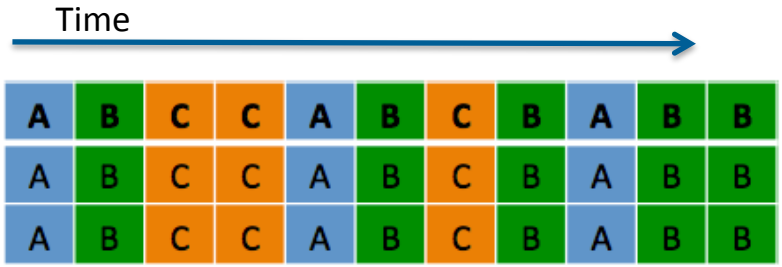
Fine-grained

- Switch occurs at the end of each cycle
  - Each thread has its own set of registers
  - No need to drain the pipeline



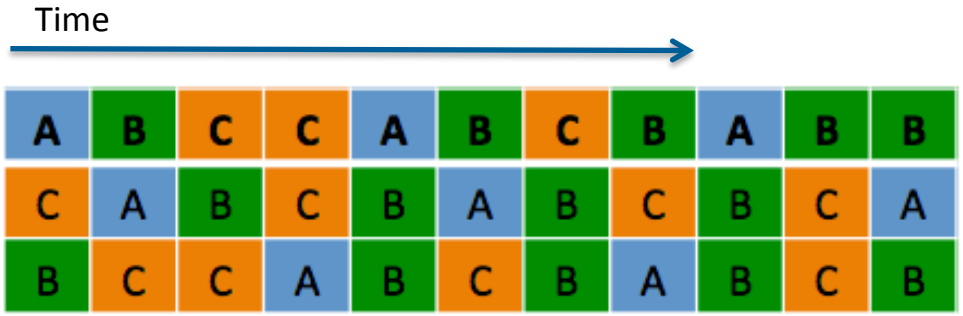
- Instructions from multiple threads are interleaved within the pipeline
- Having 2 or more processors allows several threads to execute in parallel
- To be efficient, there must be enough threads to keep the processors busy

Multiprocessors may also be superscalar



(each column corresponds to a clock cycle)

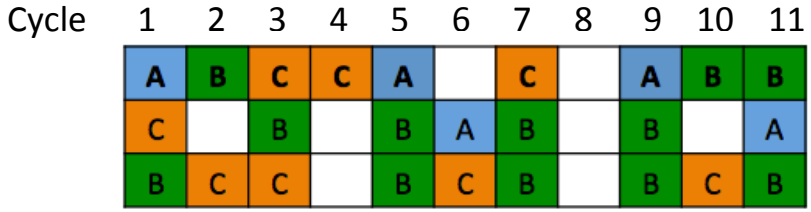
Fine-grained multithreading on a processor with three execute units which are available to only one thread at a time



Fine-grained simultaneous multithreading (SMT)
Execute units are available to all threads

Some execute units may not be used in some cycles

Data hazards and stalls may prevent the use of some units
Lack of proper instruction type may also cause idle units



(empty boxes represent idle units)

For example, with 2 integer units and 1 floating point unit, a bubble occurs if there is no floating point instruction available (cycle 2)

The need to stall may idle all 3 units (cycle 8)

SMT requires that each thread have its own register set and PC
Each thread must have its own resources
Instructions are tagged with the thread number or thread ID

Thread Level Parallelism