Multithreading on a core:

- Find a way to "hide" true data dependency stalls, cache miss stalls, and branch stalls by finding instructions (from other process threads) that are independent of those stalling instructions
- Multithreading – increase the utilization of resources on a core by allowing multiple processes (threads) to share the functional units of a single core
  - Processor must duplicate the state hardware for each thread – a separate register file, PC, instruction buffer, and store buffer for each thread
  - The caches, TLBs, BHT, BTB can be shared (although the miss rates may increase if they are not sized accordingly)
  - The memory can be shared through virtual memory mechanisms
  - Hardware must support efficient thread context switching

Possible options for Multithreading:

- Fine-grain – switch threads on every instruction issue
  - Round-robin thread interleaving (skipping stalled threads)
  - Processor must be able to switch threads on every clock cycle
  - Advantage – can hide throughput losses that come from both short and long stalls
  - Disadvantage – slows down the execution of an individual thread since a thread that is ready to execute without stalls is delayed by instructions from other threads
- Coarse-grain – switches threads only on costly stalls (e.g., L2 cache misses)
  - Advantages – thread switching doesn't have to be essentially free and much less likely to slow down the execution of an individual thread
  - Disadvantage – limited, due to pipeline start-up costs, in its ability to overcome throughput loss (pipeline must be flushed and refilled on thread switches)
- Simultaneous Multithreading (SMT) – A variation on multithreading that uses the resources of a multiple-issue, dynamically scheduled processor (superscalar) to exploit both program ILP and thread-level parallelism (TLP)
  - Most superscalar processors have more machine level parallelism than most programs can effectively use (i.e., than have ILP)
  - With register renaming and dynamic scheduling, multiple instructions from **independent threads** can be issued without regard to dependencies among them
    - Need separate rename tables (ROBs) for each thread
    - Need the capability to commit from multiple threads (i.e., from multiple ROBs) in one cycle
  - Intel's Pentium processors are SMT, called hyperthreading, that supports just two threads (doubles the architecture state)

Threading on a 4-way SS Processor Example