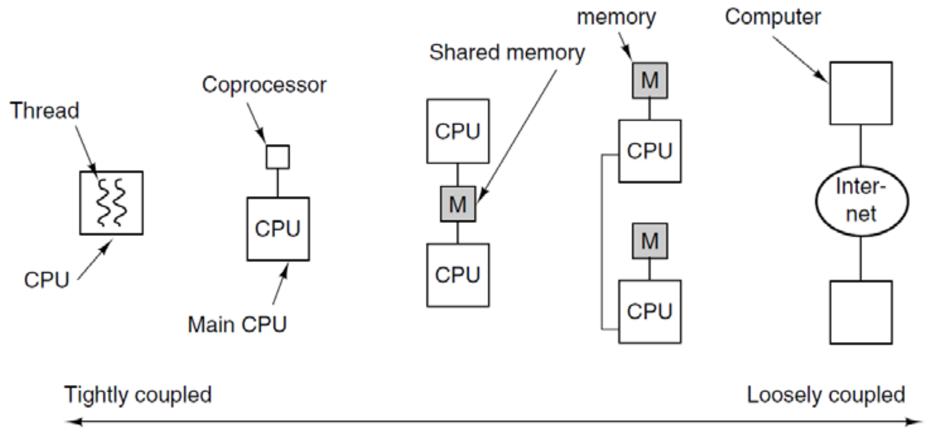
Multiprocessing and Parallelism

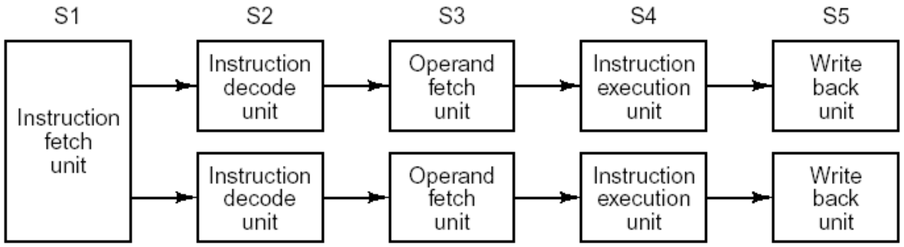
Ways to achieve parallelism



On-chip parallelism

1. **Instruction level parallelism**
   - Issue multiple instructions per clock cycle

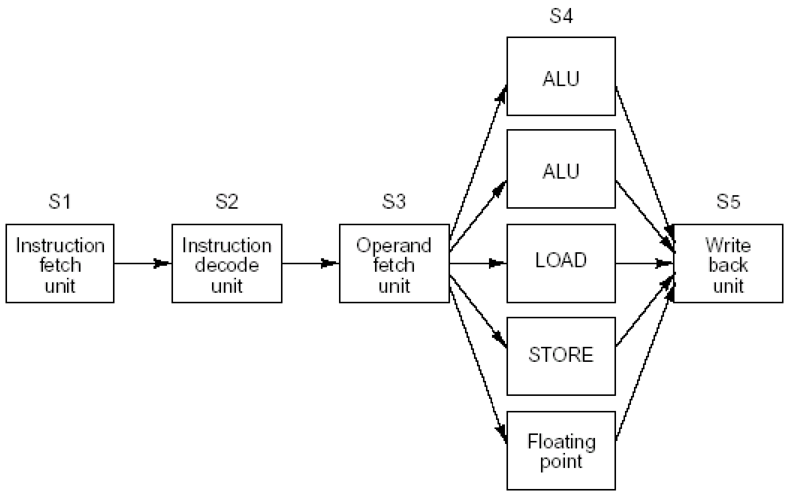
Example: Dual-pipeline architecture



- Instruction fetch unit issues two instructions, puts each into its own pipeline
- In case of a conflict, one pipeline is executed, the other is held, and paired with next instruction

Example: Superscalar architecture

- A single pipeline with multiple functional units
- VLIW (Very Long Instruction Word)(many opcodes and sets of operands)
- Can perform 5 operations at the same time (5 opcodes + 5 pairs of operands)



On-chip (hardware) multithreading

- In a pipelined CPU, pipeline stalls in a cache miss
- This allows the CPU to manage multiple threads at the same time in order to mask these stalls
  - Thread is a lightweight process that includes program counter, register state, stack (details will be discussed in OS course)
- Increases the utilization of a processor by switching to another thread when one is stalled

Variants of on-chip multithreading

✓ Fine-grained multithreading
✓ Coarse-grained multithreading
✓ Simultaneous multithreading

Fine-grained multithreading

- Switches between threads on each instruction
- Runs threads in round robin fashion with a different thread in consecutive cycles
- If an instruction stalls, subsequent instructions cannot be issued

Example:

- Assume a computer that can issue one instruction per clock cycle
- Three threads (A, B, C), 20 clock cycles
- Blank cells: dead cycles (2 cycles for miss in level 1 cache, 5 cycles for level 2, 50 cycles for last level cache)

A1	A2			A3						A4	A5								

B1			B2			B3	B4	B5											

C1	C2	C3	C4			C5	C6						C7			C8			

A1	B1	C1	A2	B2	C2	A3	B3	C3		B4	C4	A4	B5	C5	A5		C6		

- Advantage:
  - Hide throughput losses arising from stalls
- Disadvantage:
  - Thread that is ready execute without stalls will be delayed by instructions from other threads

Coarse-grained multithreading

- A thread starts and continues to issue instructions until it encounters an expensive stall (eg: last level cache miss)
- At this point a switch occurs and the next thread starts
- Threads are run in turn in this manner

A1	A2			A3	B1			B2			B3	B4	B5	C1	C2	C3	C4		

A1	A2			A3						A4	A5	B1			B2			B3	B4

- Less efficient than fine-grained
- Unable to overcome throughput losses from shorter stalls

Fine-grained multithreading in a dual-issue CPU

- CPU can issue 2 instructions per thread in a clock cycle
- When an instruction stalls, subsequent instructions cannot be issued

A1	B1	C1	A3	B2	C3		B3	C5	A4	B5				C7			C8		
A2		C2			C4		B4	C6	A5										

Coarse-grained multithreading in a dual-issue CPU

- CPU issues 2 instructions per thread in a clock cycle
- Threads are run in turn until one instruction stalls (larger) and then switch to the next thread in the next cycle immediately

A1			A3	B1			B2			B3	B5	C1	C3			C5	A4		
A2										B4		C2	C4			C6	A5		

A1			A3						A4	B1			B2			B3	B5	C1	C3
A2									A5							B4		C2	C4

Simultaneous multithreading in a dual-issue CPU

- Refinement to coarse-grained multithreading
- A single thread is issues two instructions per clock cycle as long as it can
- When stalls, instructions are immediately taken from the next thread in sequence

A1	B1	C2	C4	B2		C5	B3	B5	A4			C7			C8				
A2	C1	C3	A3			C6	B4		A5										

Hyper-Threading (HT)

- Allows a single processor to run two treads at once
- HT converts a single physical processor into two virtual processors
- An HT-enabled processor has 2 sets of general-purpose registers, control registers and other architecture components, but same cache, execution units and buses
- Provides 25% increase in performance for additional 5% increase in chip size for HT hardware

Single-chip multiprocessors

- Homogeneous multiprocessors on a chip
  - Having 2 or more CPUs on a single chip
  - Share same memory, last level of cache etc.
- Heterogeneous multiprocessors on a chip
  - Appears in audio-visual consumer electronics (TV, DVD players etc.)

Coprocessors

- A second specialized processor
- Coprocessors perform specialized tasks helping the main processor
- Come in as a part of the CPU package or in a plug-in board

Math coprocessor (Floating-Point Unit) (FPU)

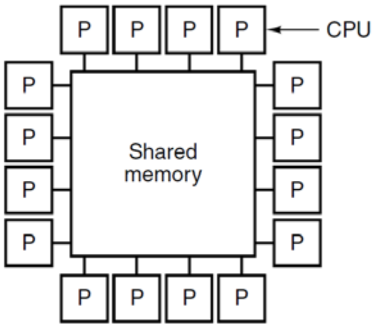
- Provides hardware for floating-point math
- Speeds computer's operations when running software designed to use the coprocessor
- Can perform high-level mathematical operations (trigonometric functions, roots, logarithms) many times faster
- Instruction set of math chip is different from that of main CPU

GPU (Graphics Processing Unit)

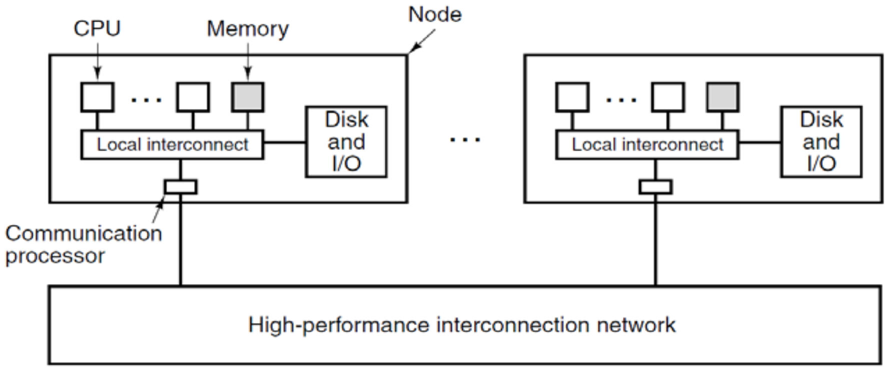
- Designed to handle high-resolution graphics processing
- Contains a large number of cores with a smaller instruction set
- Rely on hardware multithreading
- Comes in as
  - Plug-in cards (high performance)
  - On motherboard (high end video used in laptops)
  - In motherboard chipset (economical, shares system memory and other components, less powerful)
  - Integrated to processor (shares system memory, less powerful)

Shared Memory Multiprocessor (SMP)

- All CPUs share a common memory
- Processors communicate through shared variables in memory
- Allows old programs to run well on parallel hardware



Message-Passing Multiprocessors



- Each node consists of
  - One or more CPUs
  - RAM shared by CPUs within the node
  - Disk and I/O devices
  - Communication processor
- Communication processors are connected by a high-speed interconnection network
- Processors communicate by passing messages