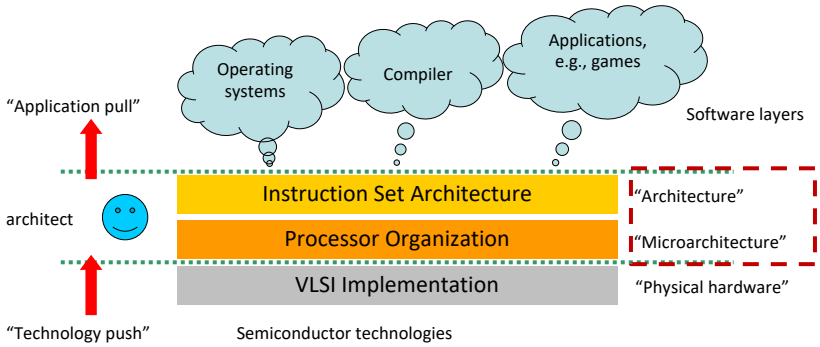
Review

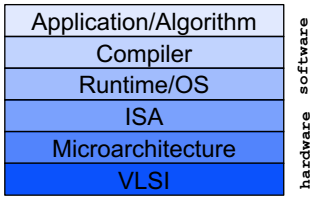
- Briefly go over the major contents we have discussed during this Spring term
- Note that, this is **NOT** a comprehensive summary of final exam coverage.
- All the contents we have discussed in the classes will be potential questions for the final exam, regardless in this review or not
- Final exam schedule
  - Wednesday, April 20th, 2022. 11:00am to 12:15pm
  - In class, in person, close-note
  - No calculator needed
- Do not forget the project submission and demo after the final exam!

What is Computer Architecture? (§1.3)



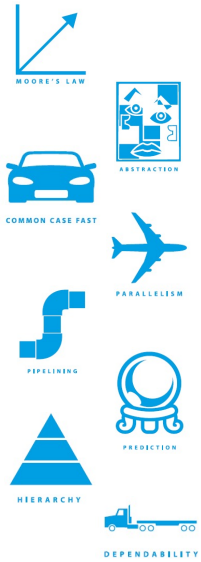
Models of Parallel executions:

- Instruction Level parallelism (ILP)
- Data-level parallelism (DLP)
- Thread-level parallelism (TLP)
- Request-level parallelism (RLP)
- Memory-level parallelism (MLP)



Eight Great Ideas in Computer Architecture

- Design for ***Moore's Law***
- Use ***abstraction*** to simplify design
- Make the ***common*** *case fast*
- Performance *via* ***parallelism***
- Performance *via* ***pipelining***
- Performance *via* ***prediction***
- *Hierarchy* of **memories**
- ***Dependability*** *via* redundancy



Review

classic von Neumann

SISD **Single instruction stream** **Single data stream** `Single Core`	**(SIMD)** **Single instruction stream** **Multiple data stream** `GPUs, Vector machines`
MISD **Multiple instruction stream** **Single data stream** `TPUs`	**(MIMD)** **Multiple instruction stream** **Multiple data stream** `Multi-core`

Does it make sense?

Yes, systolic array.

Review

- Moore's Law
- Dennard Scaling
- Technology improvement vs. architecture improvement
- Performance equations
- Amdahl's Law

$$CPU\ time = Cycle\ time * \sum_{j=1}^{n} (CPI_j \ \mathbf{x}\ I_j)$$

Where I_j is the number of instructions of type *j*, and

Cycle time is the inverse of the *clock rate*.

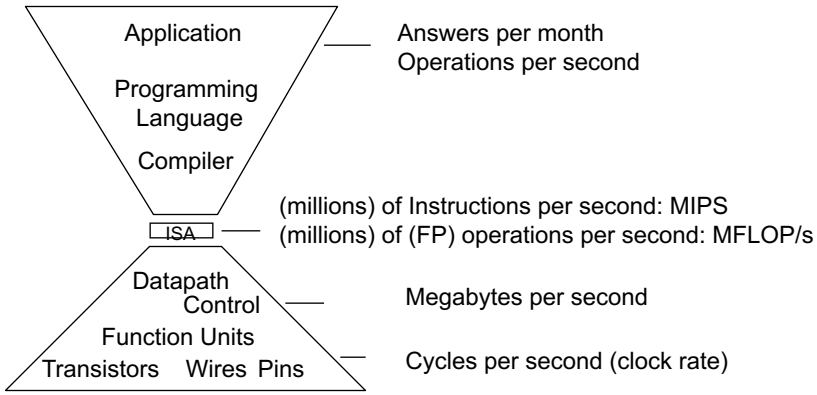
$$CPU\ time = \text{total \# of instructions} \ \mathbf{x}\ CPI \ \mathbf{x}\ Cycle\ time$$

Example: For some programs,

Machine A has a clock cycle time of 10 ns. and a CPI of 2.0
Machine B has a clock cycle time of 20 ns. and a CPI of 1.2

What machine is faster for this program, and by how much?

Performance



Review

- CISC vs RISC
  - Design philosophy
  - Different types
  - Different fields: Opcode, operands, etc.
  - Big/little Endian
- Pipeline execution
  - Pipeline registers/buffer
  - Multi-cycle pipelines
- Hazards
  - Structural
  - Data
  - Control

Review

- Hardware forwarding
- Register renaming
- Branch prediction
- Static parallelism
  - VLIW
- Dynamic parallelism
  - Scoreboarding
  - Tomasulo
  - Speculative Tomasulo
- Multi-threading
  - FGMT, CGMT
  - SMT

Review

- Memory hierarchy
  - Basics, bandwidth vs. latency
  - Cache hierarchy: direct, set-associative, fully-associative
  - 3C cache misses and their differences
  - AMAT, cache hit rates
  - Virtual memory and address translation basics
  - Translation + cache
- 13 cache optimizations and 6 TLB optimizations
- Memory organization
  - Bank level parallelism: pros and cons
  - Row-buffer locality: pros and cons
- Non-volatile memory

Review

- Memory hierarchy
  - Basics, bandwidth vs. latency
  - Cache hierarchy: direct, set-associative, fully-associative
  - 3C cache misses and their differences
  - AMAT, cache hit rates
  - Virtual memory and address translation basics
  - Translation + cache
- 13 cache optimizations and 6 TLB optimizations
- Memory organization
  - Bank level parallelism: pros and cons
  - Row-buffer locality: pros and cons
- Non-volatile memory

Review

- Thread level parallelism and multi/many core
  - SMP and chip-multiprocessors
  - Address space sharing and physical memory sharing
- Cache coherence
  - Protocols: vi, msi, mesi, etc.
  - Implementations: bus, directory
- Synchronization
  - ll and sc
  - deadlocking
- Memory consistency

Review

- Data parallelism
- Vectorization
- GPUs. Basics in architecture and programming
- GPU optimizations. Three lessons studies