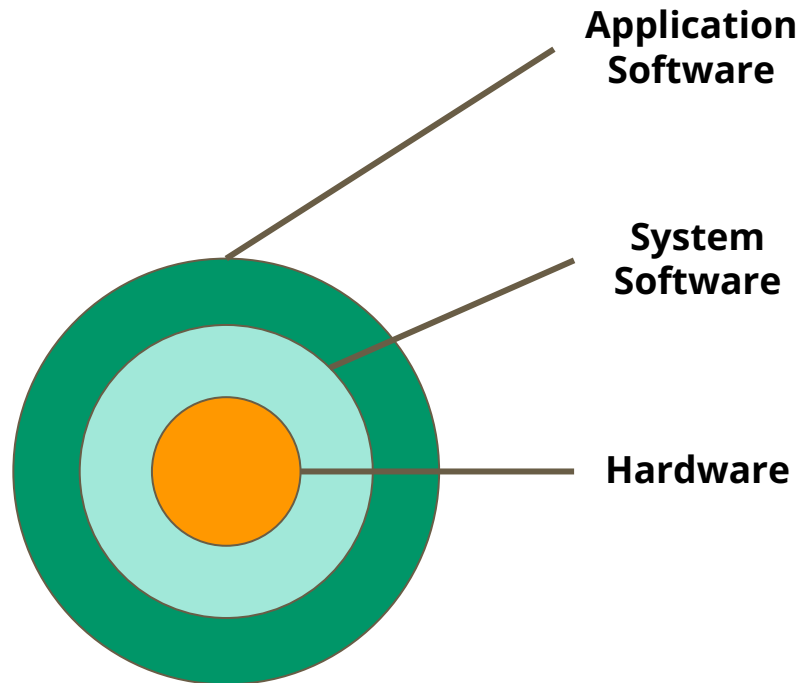

Bits of Architecture

— Core Ideas in Architecture —

Core Ideas In Architecture (From Patterson & Hennessy)

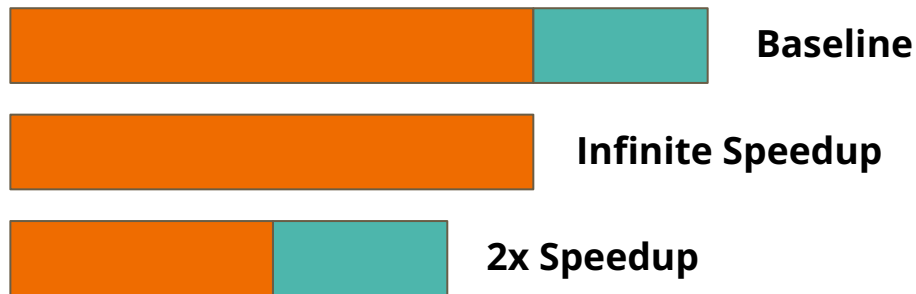
Abstraction

- Abstractions can improve productivity
 - E.g., assembly vs high-level languages
- We rely on abstractions throughout architecture
 - What do we expose to programmers?
 - How do we think about hardware structures?



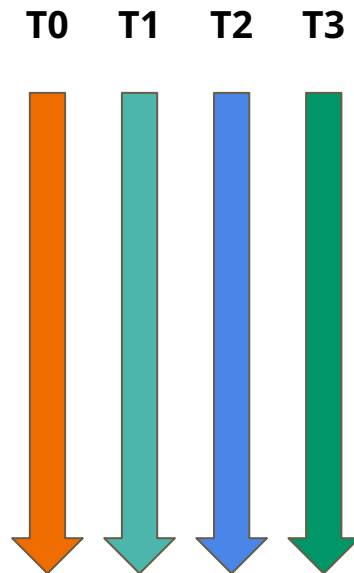
Making the Common Case Fast

- How do we prioritize optimizations?
 - Where do we spend our time and silicon?
- A moderate improvement to the common case can be more impactful than a large improvement to an infrequent case



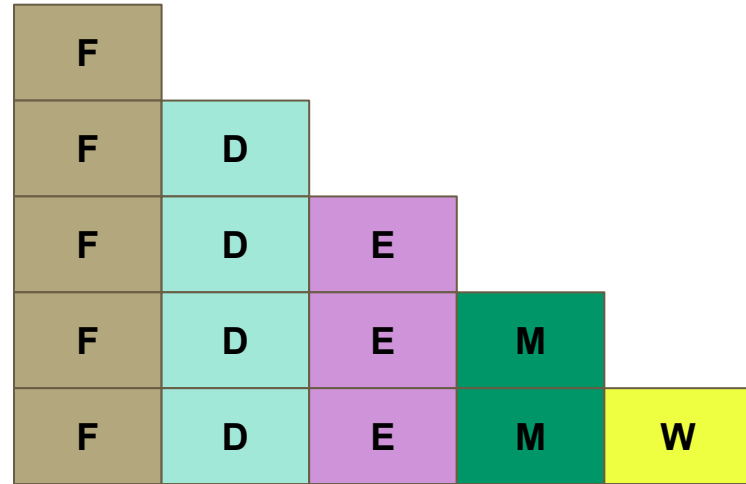
Performance via Parallelism

- There are limits to how far we can optimize serial execution
- However, some tasks can be worked on in parallel
- Improve performance by working on multiple things at once



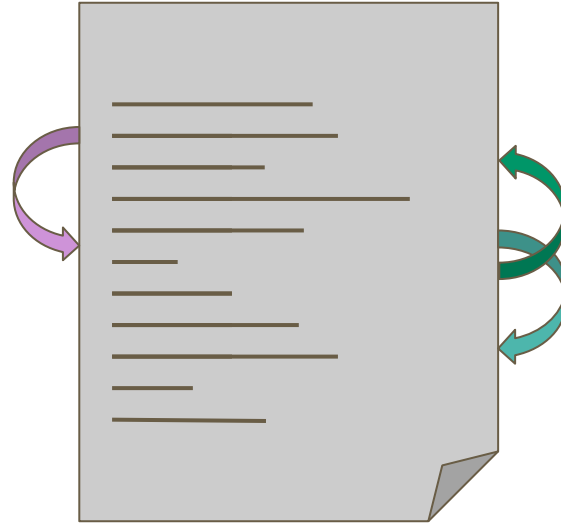
Performance via Pipelining

- Some operations cannot be completely overlapped
- However, we might be able to partially overlap them
 - Another form of parallelism
- Improve performance by overlapping pipeline stages



Performance via Prediction

- Waiting on results wastes time
- Programs are often very dynamic
 - Accessing different parts of memory
 - Jumping to different lines of code
- Improve performance by prediction
 - Works if recovery from misprediction is cheap



Hierarchies of Memory

- We would ideally want infinite fast memory
 - A very old idea!
- Programs often go through phases
 - Not all data is accessed at once
- A hierarchy can help give us the illusion of infinite fast memory

Fast

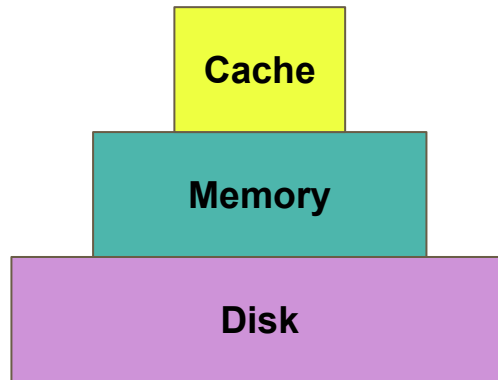


Slow

Expensive

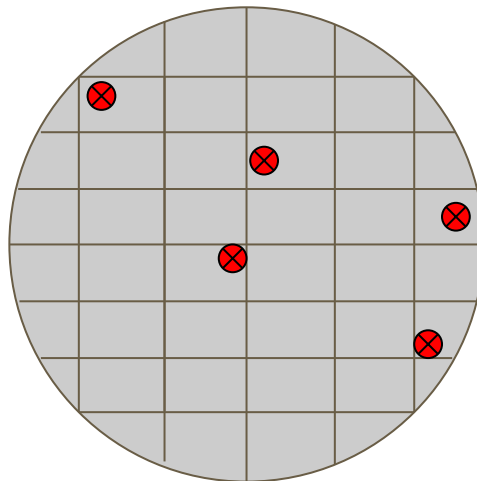


Cheap



Dependability via Redundancy

- Computers need to be dependable
- Redundancy exists for many reasons
 - Handle manufacturing defects
 - Handle errors at runtime



Designing for Moore's Law (deprecated)

- Integrated Circuit resources doubled every 1.5-2 years
 - Now? Not so much...
- How are we getting around this?
 - New architectures
 - Heterogeneous chips
 - Larger chips
 - Chiplets

