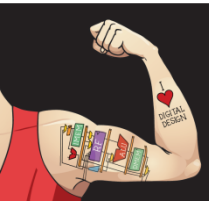
E85 Digital Design & Computer Engineering

Lecture 23:
Advanced Microarchitecture

HARVEY MUDD COLLEGE

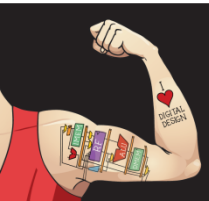
Advanced Microarchitecture

- Deep Pipelining
- Micro-operations
- Branch Prediction
- Superscalar Processors
- Out of Order Processors
- Register Renaming
- SIMD
- Multithreading
- Multiprocessors



Deep Pipelining

- 10-20 stages typical
- Number of stages limited by:
  - Pipeline hazards
  - Sequencing overhead
  - Power
  - Cost



Micro-operations

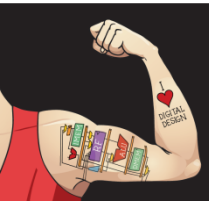
- Decompose more complex instructions into a series of simple instructions called *micro-operations* (*micro-ops* or *μ-ops*)
- At run-time, complex instructions are decoded into one or more micro-ops
- Used heavily in CISC (complex instruction set computer) architectures (e.g., x86)
- Used for some ARM instructions, for example:

Complex Op	Micro-op Sequence
`LDR R1, [R2], #4`	`LDR R1, [R2]` `ADD R2, R2, #4`

Without u-ops, would need 2nd write port on the register file

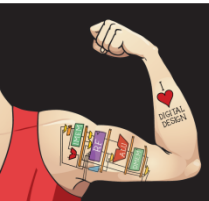
Micro-operations

- Allow for dense code (fewer memory accesses)
- Yet preserve simplicity of RISC hardware
- ARM strikes balance by choosing instructions that:
  - Give better code density than pure RISC instruction sets (such as MIPS)
  - Enable more efficient decoding than CISC instruction sets (such as x86)



Branch Prediction

- Guess whether branch will be taken
  - Backward branches are usually taken (loops)
  - Consider history to improve guess
- Good prediction reduces fraction of branches requiring a flush



Branch Prediction

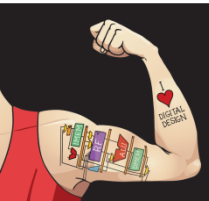
- Ideal pipelined processor: CPI = 1
- Branch misprediction increases CPI
- **Static branch prediction:**
  - Check direction of branch (forward or backward)
  - If backward, predict taken
  - Else, predict not taken
- **Dynamic branch prediction:**
  - Keep history of last several hundred (or thousand) branches in *branch target buffer*, record:
    - Branch destination
    - Whether branch was taken

Branch Prediction Example

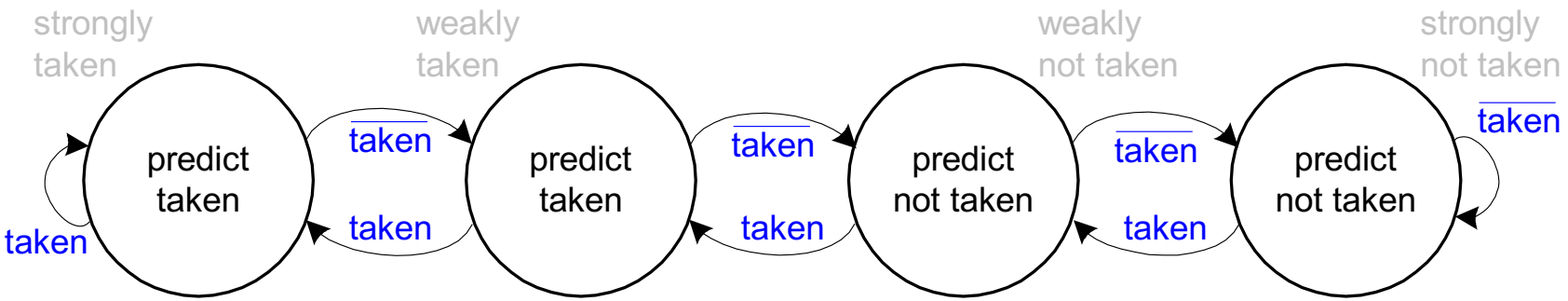
```
    MOV R1, #0              ; R1 = sum
    MOV R0, #0              ; R0 = i

FOR                         ; for (i=0; i<10; i=i+1)
    CMP R0, #10
    BGE DONE
    ADD R1, R1, R0          ; sum = sum + i
    ADD R0, R0, #1
    B   FOR

DONE
```

1-Bit Branch Predictor

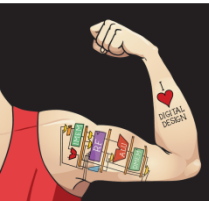
- Remembers whether branch was taken the last time and does the same thing
- Mispredicts first and last branch of loop



2-Bit Branch Predictor

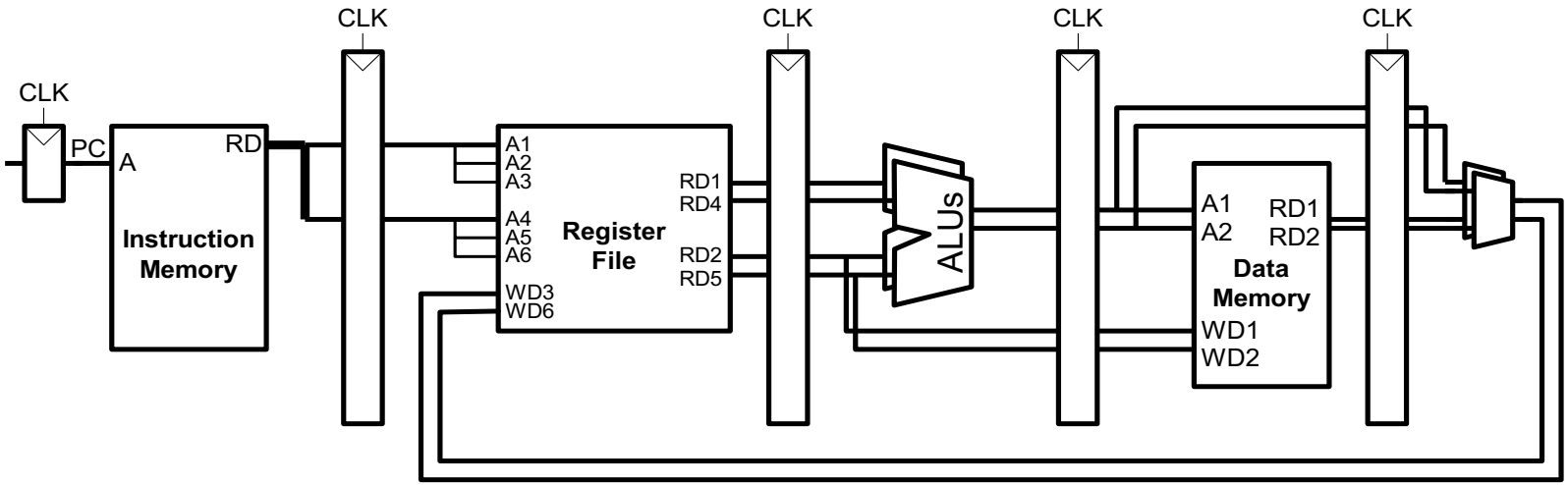


Only mispredicts last branch of loop



Superscalar

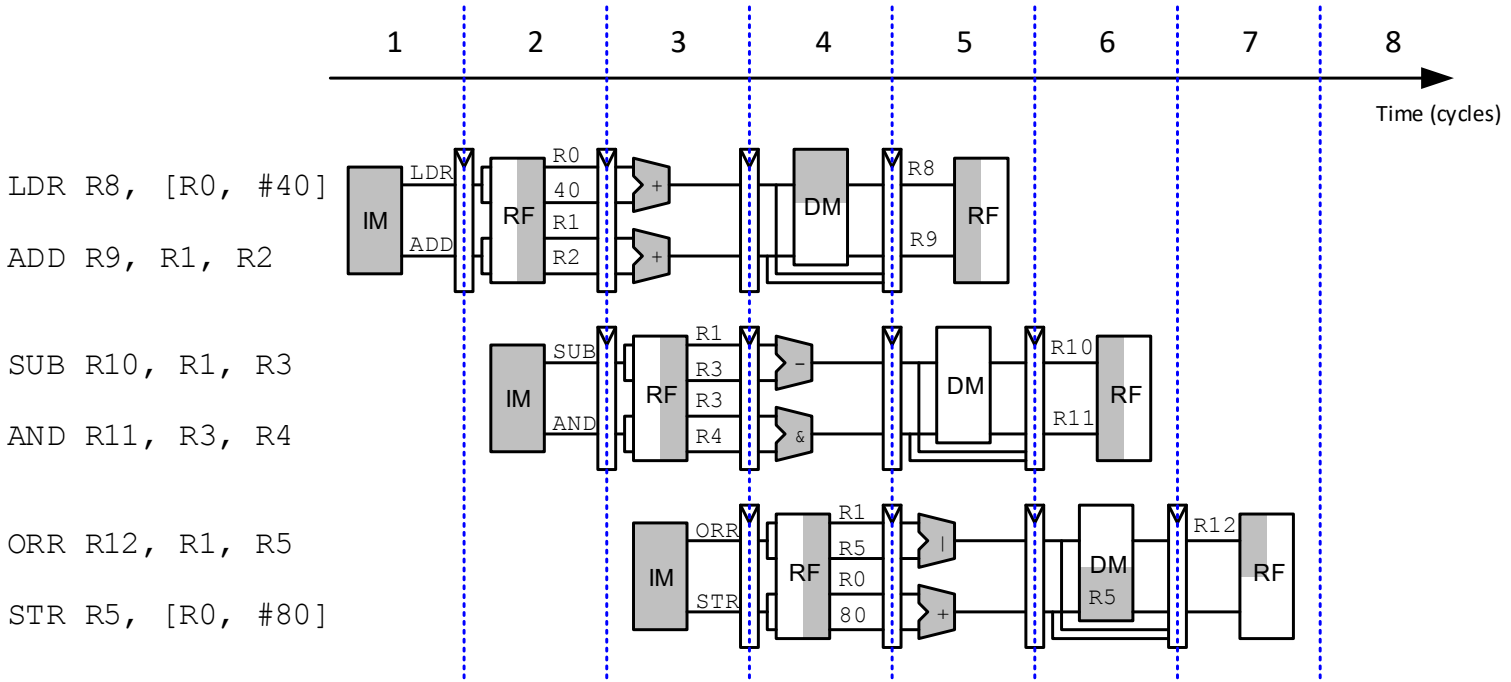
- Multiple copies of datapath execute multiple instructions at once
- Dependencies make it tricky to issue multiple instructions at once



Superscalar Example

Ideal IPC: 2

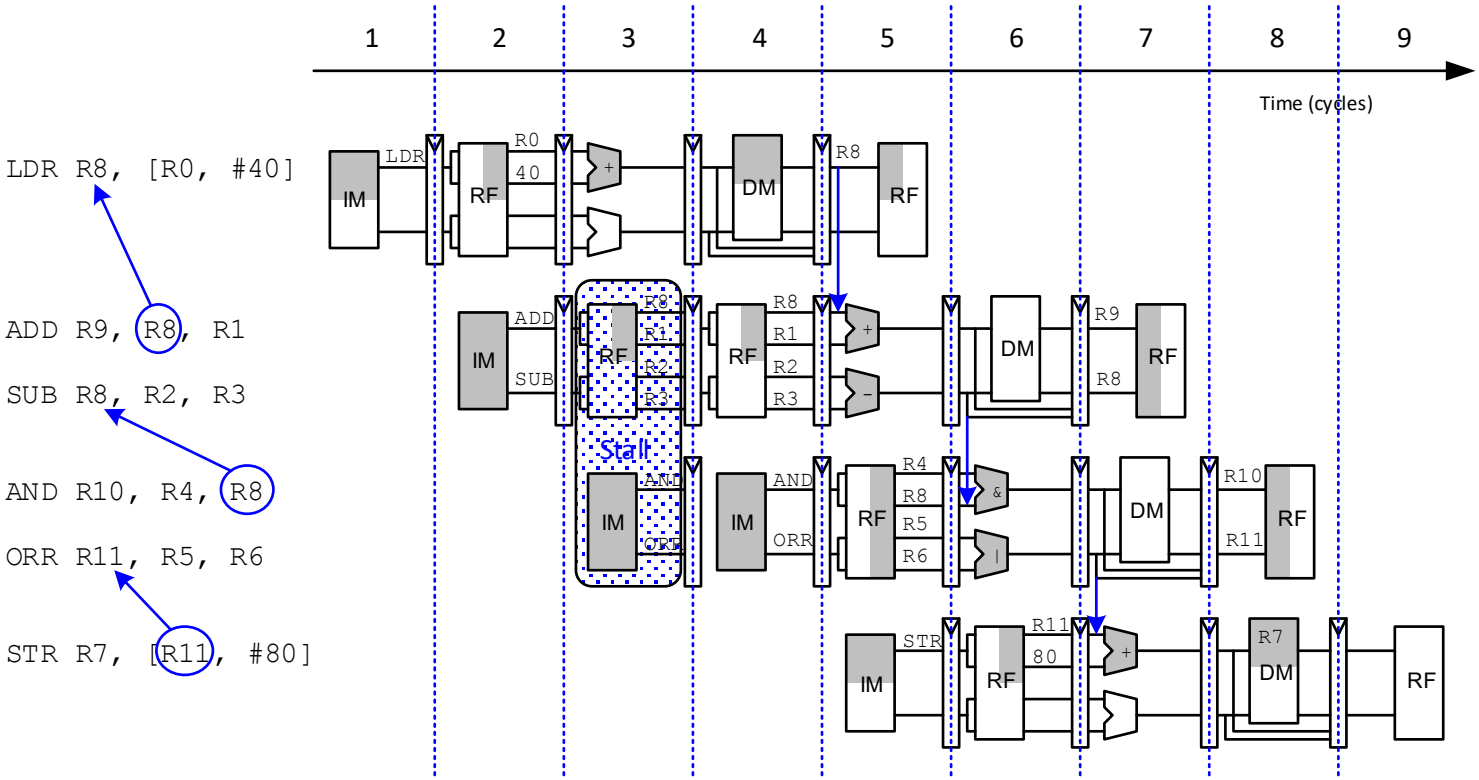
Actual IPC: 2



Superscalar with Dependencies

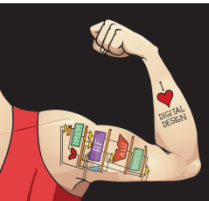
Ideal IPC: 2

Actual IPC: 6/5 = 1.2



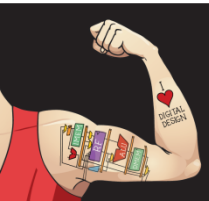
Out of Order Processor

- Looks ahead across multiple instructions
- Issues as many instructions as possible at once
- Issues instructions out of order (as long as no dependencies)
- **Dependencies:**
  - **RAW** (read after write): one instruction writes, later instruction reads a register
  - **WAR** (write after read): one instruction reads, later instruction writes a register
  - **WAW** (write after write): one instruction writes, later instruction writes a register



Out of Order Processor

- **Instruction level parallelism (ILP):** number of instruction that can be issued simultaneously (average < 3)
- **Scoreboard:** table that keeps track of:
  - Instructions waiting to issue
  - Available functional units
  - Dependencies

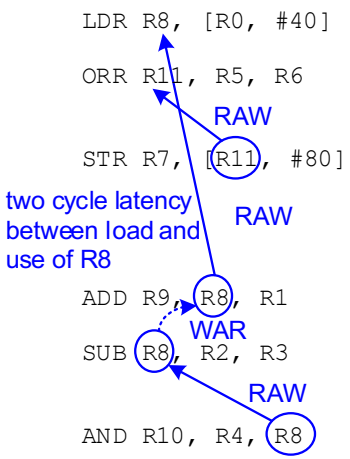


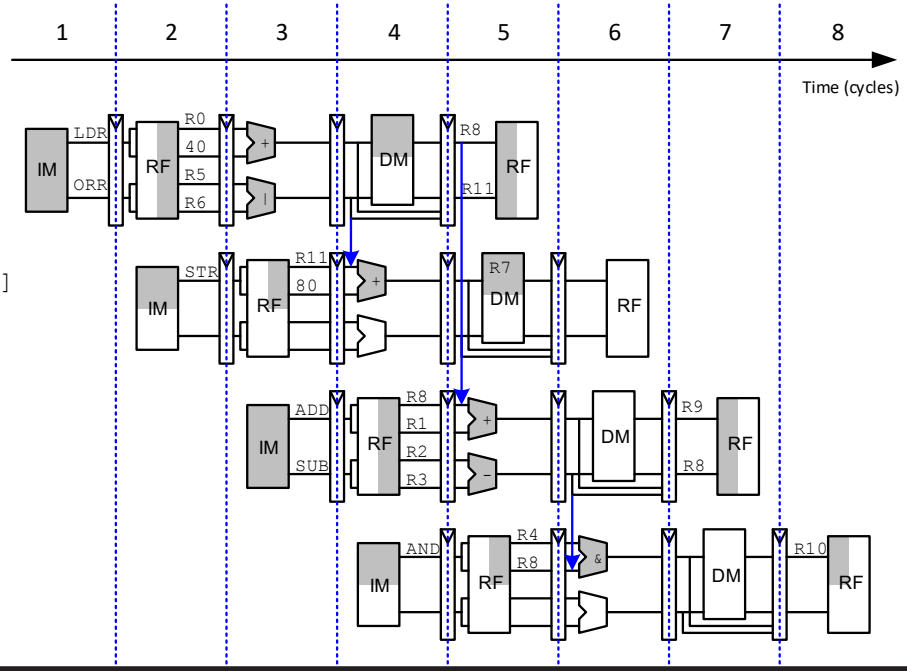
Out of Order Processor Example

```
LDR  R8, [R0, #40]
ADD  R9,  R8, R1
SUB  R8,  R2, R3
AND  R10, R4, R8
ORR  R11, R5, R6
STR  R7, [R11, #80]
```

Ideal IPC: 2

Actual IPC: 6/4 = 1.5



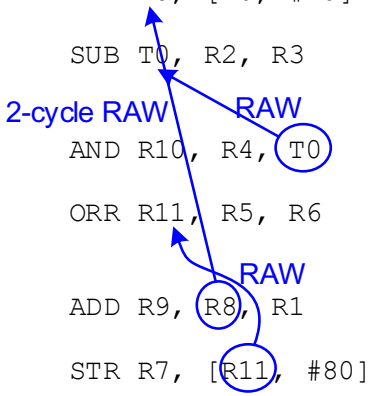


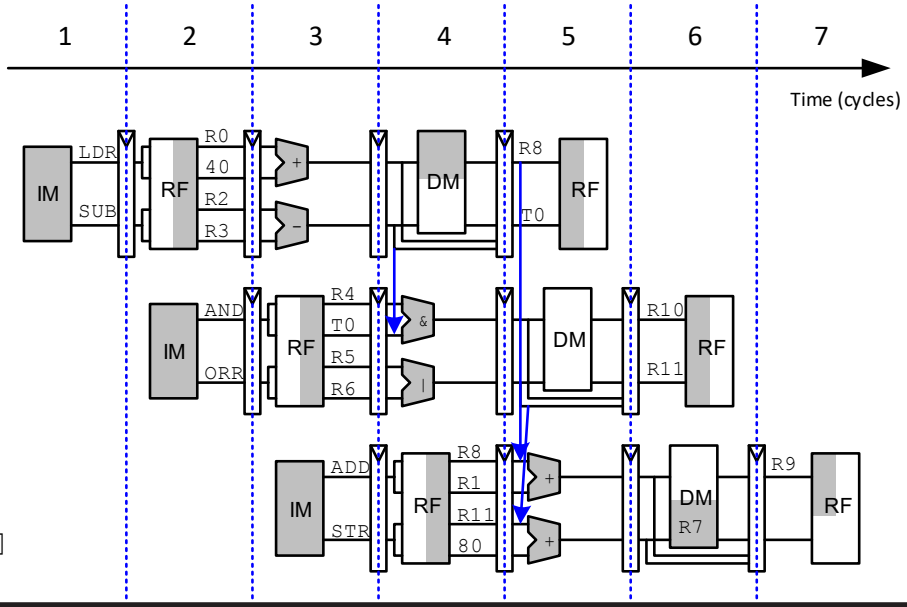
Register Renaming

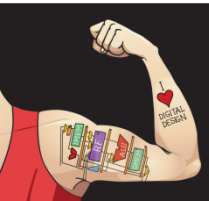
```
LDR  R8, [R0, #40]
ADD  R9,  R8, R1
SUB  R8,  R2, R3
AND  R10, R4, R8
ORR  R11, R5, R6
STR  R7, [R11, #80]
```

Ideal IPC: 2

Actual IPC: 6/3 = 2

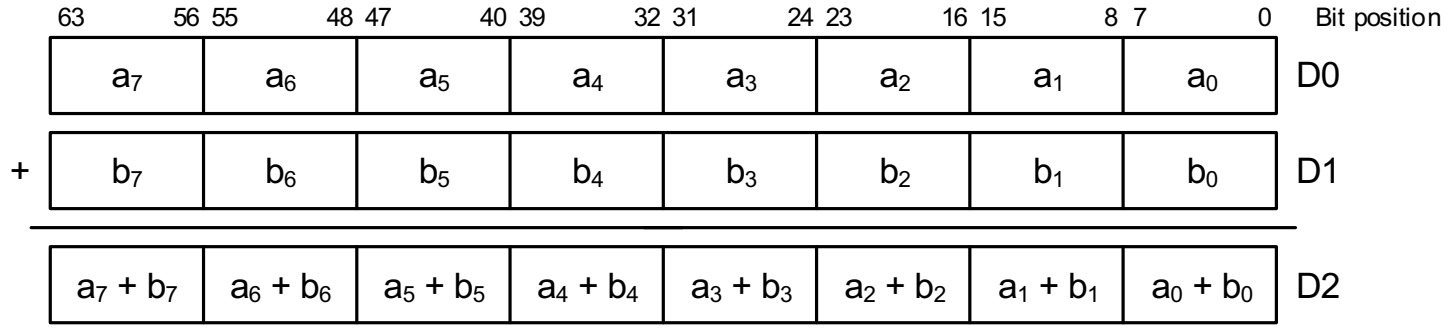


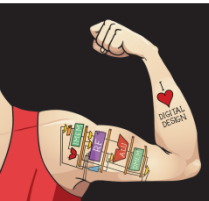




SIMD

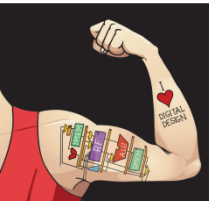
- Single Instruction Multiple Data (SIMD)
  - Single instruction acts on multiple pieces of data at once
  - Common application: graphics
  - Perform short arithmetic operations (also called *packed arithmetic*)
- For example, add eight 8-bit elements





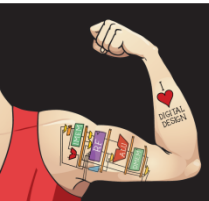
Advanced Architecture Techniques

- **Multithreading**
  - Word processor: thread for typing, spell checking, printing
- **Multiprocessors**
  - Multiple processors (cores) on a single chip



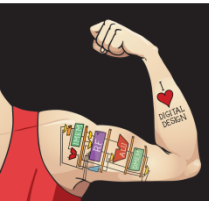
Threading: Definitions

- **Process:** program running on a computer
  - Multiple processes can run at once: e.g., surfing Web, playing music, writing a paper
- **Thread:** part of a program
  - Each process has multiple threads: e.g., a word processor may have threads for typing, spell checking, printing



Threads in Conventional Processor

- One thread runs at at at a time
- When one thread stalls (for example, waiting for memory):
  - Architectural state of that thread stored
  - Architectural state of waiting thread loaded into processor and it runs
  - Called **context switching**
- Appears to user like all threads running simultaneously



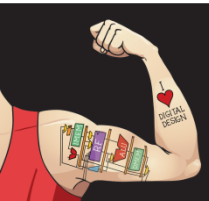
Multithreading

- Multiple copies of architectural state
- Multiple threads **active** at once:
  - When one thread stalls, another runs immediately
  - If one thread can't keep all execution units busy, another thread can use them
- Does not increase instruction-level parallelism (ILP) of single thread, but increases throughput

Intel calls this "hyperthreading"

Multiprocessors

- Multiple processors (cores) with a method of communication between them
- Types:
  - **Homogeneous:** multiple cores with shared main memory
  - **Heterogeneous:** separate cores for different tasks (for example, DSP and CPU in cell phone)
  - **Clusters:** each core has own memory system



Other Resources

- Patterson & Hennessy's: *Computer Architecture: A Quantitative Approach*
- Conferences:
  - www.cs.wisc.edu/~arch/www/
  - ISCA (International Symposium on Computer Architecture)
  - HPCA (International Symposium on High Performance Computer Architecture)