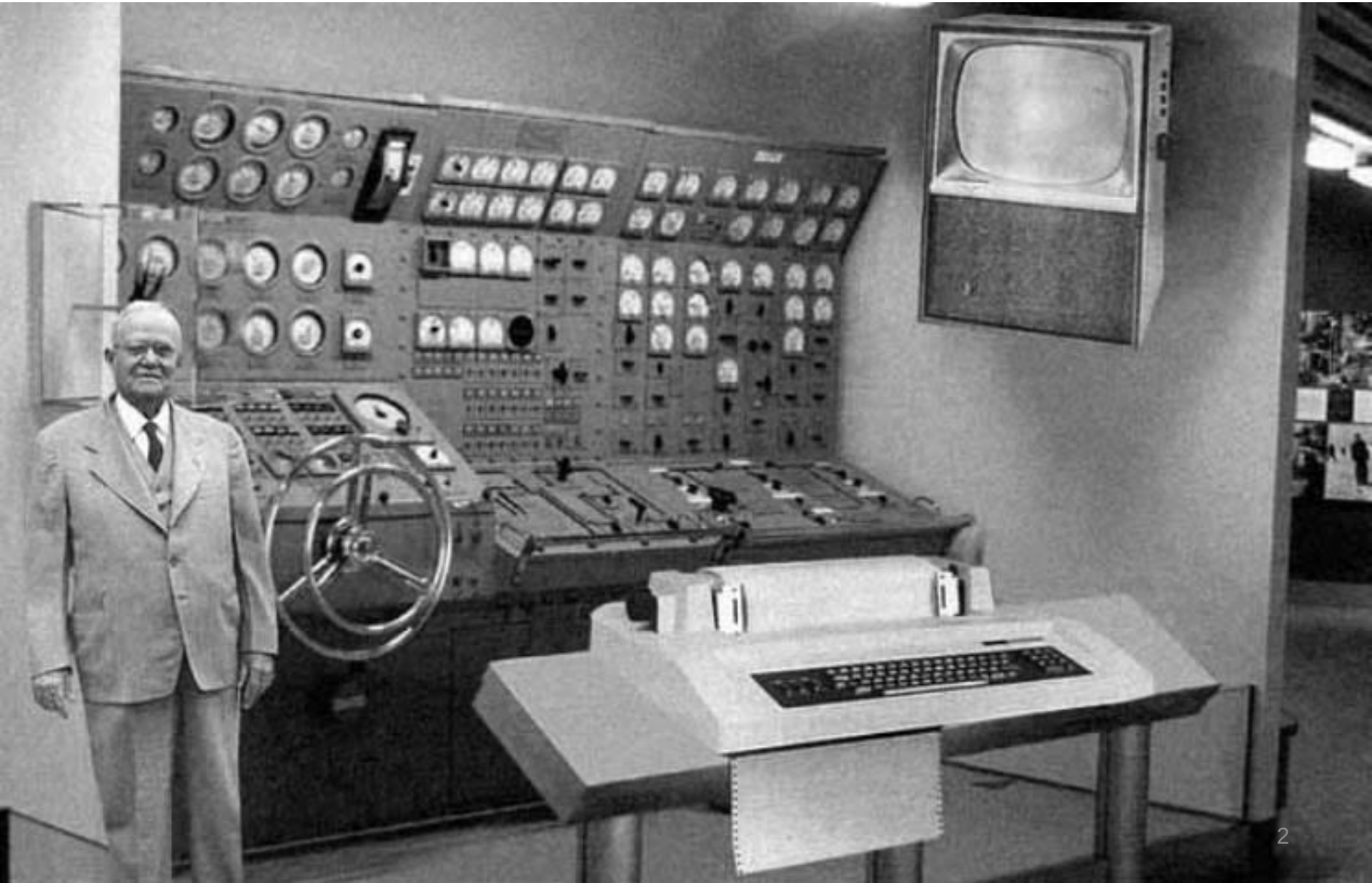
CS3350B
Computer Architecture
Winter 2015

Introduction

Marc Moreno Maza

www.csd.uwo.ca/Courses/CS3350b

What is a computer?



What is a computer?

What is a computer?

What is a computer?

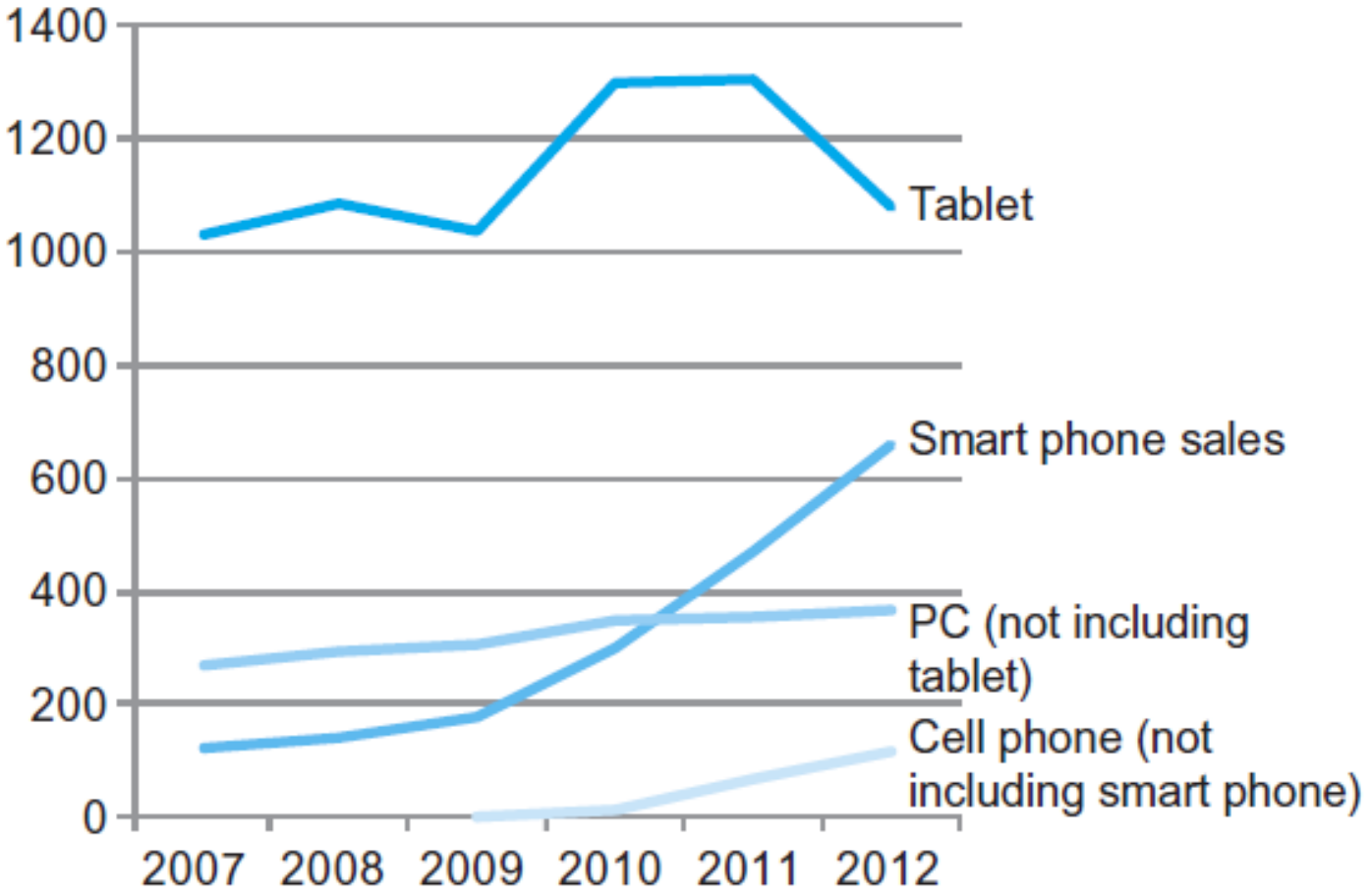
The Computer Revolution

- **Progress in computer technology**
  - Underpinned by Moore's Law
- **Makes novel applications feasible**
  - Computers in automobiles
  - Cell phones
  - Human genome project
  - World Wide Web
  - Search Engines
- **Computers are pervasive**

Classes of Computers

- **Personal computers**
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- **Server computers**
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized
- **Supercomputers**
  - High-end scientific and engineering calculations
  - Highest capability but represent a small fraction of the overall computer market
- **Embedded computers**
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

The PostPC Era

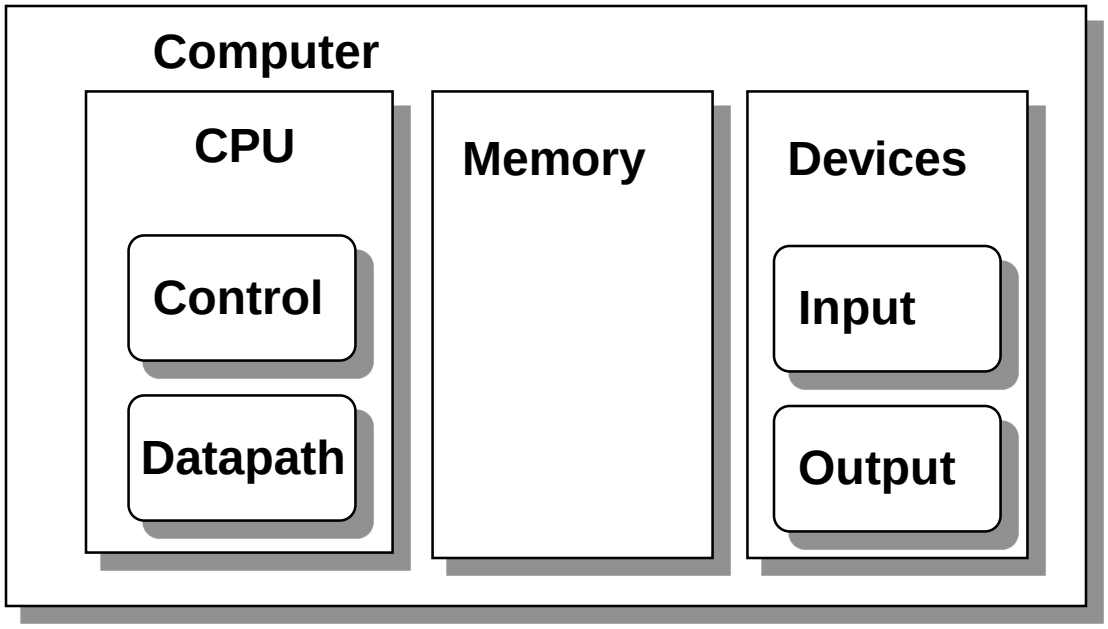


The PostPC Era

- **Personal Mobile Device (PMD)**
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses

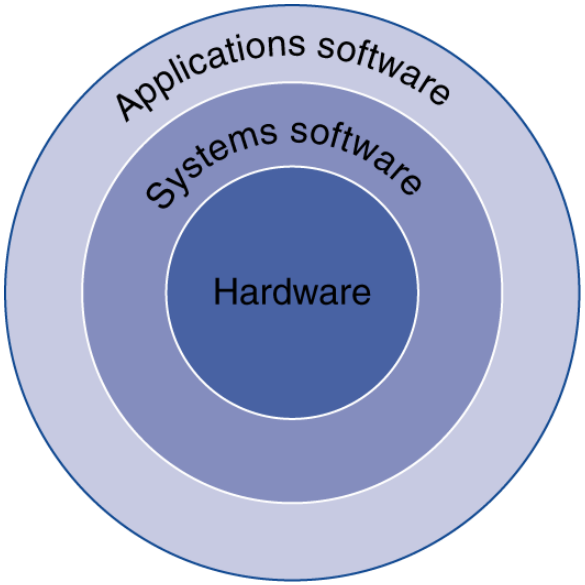
- **Cloud computing**
  - Warehouse Scale Computers (WSC)
  - Software as a Service (SaaS)
  - Portion of software run on a PMD and a portion run in the Cloud
  - Amazon and Google

Components of a Computer



- Same components for all kinds of computer
  - Desktop, server, embedded

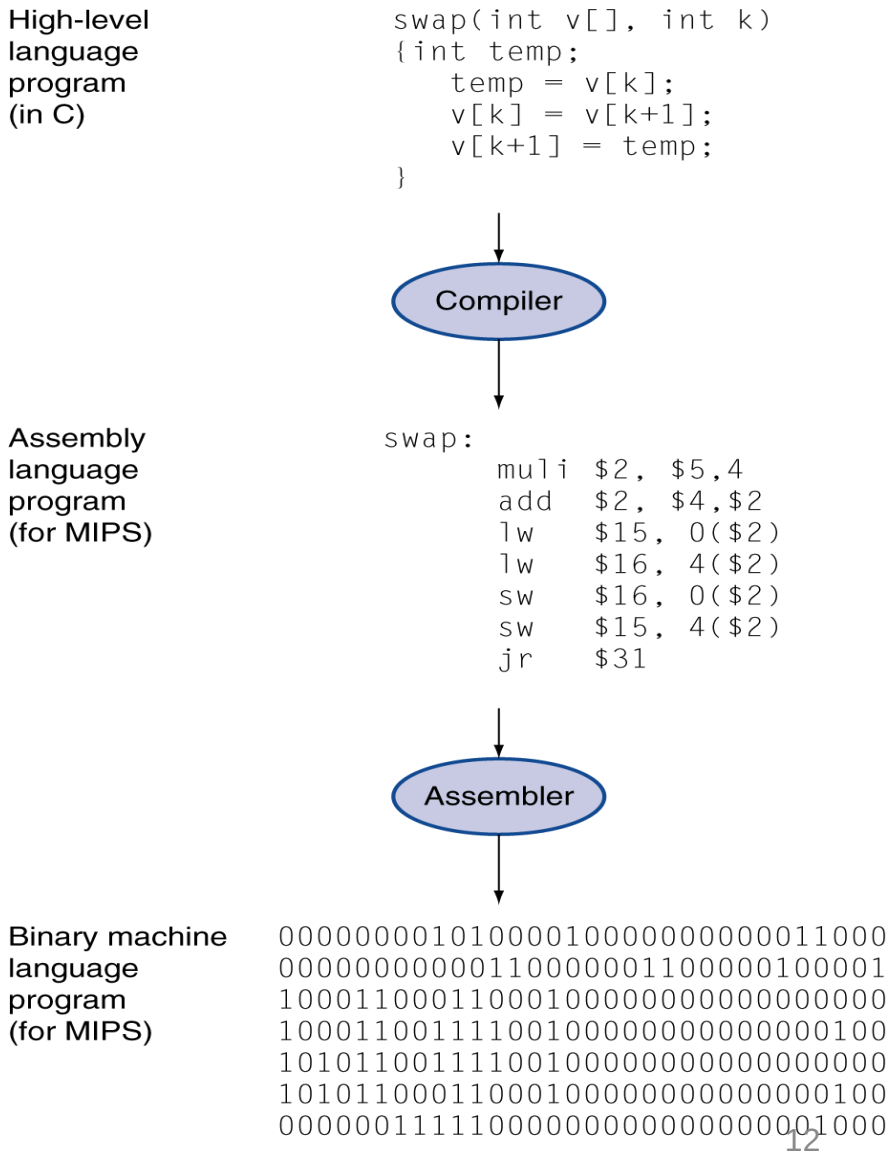
Below Your Program



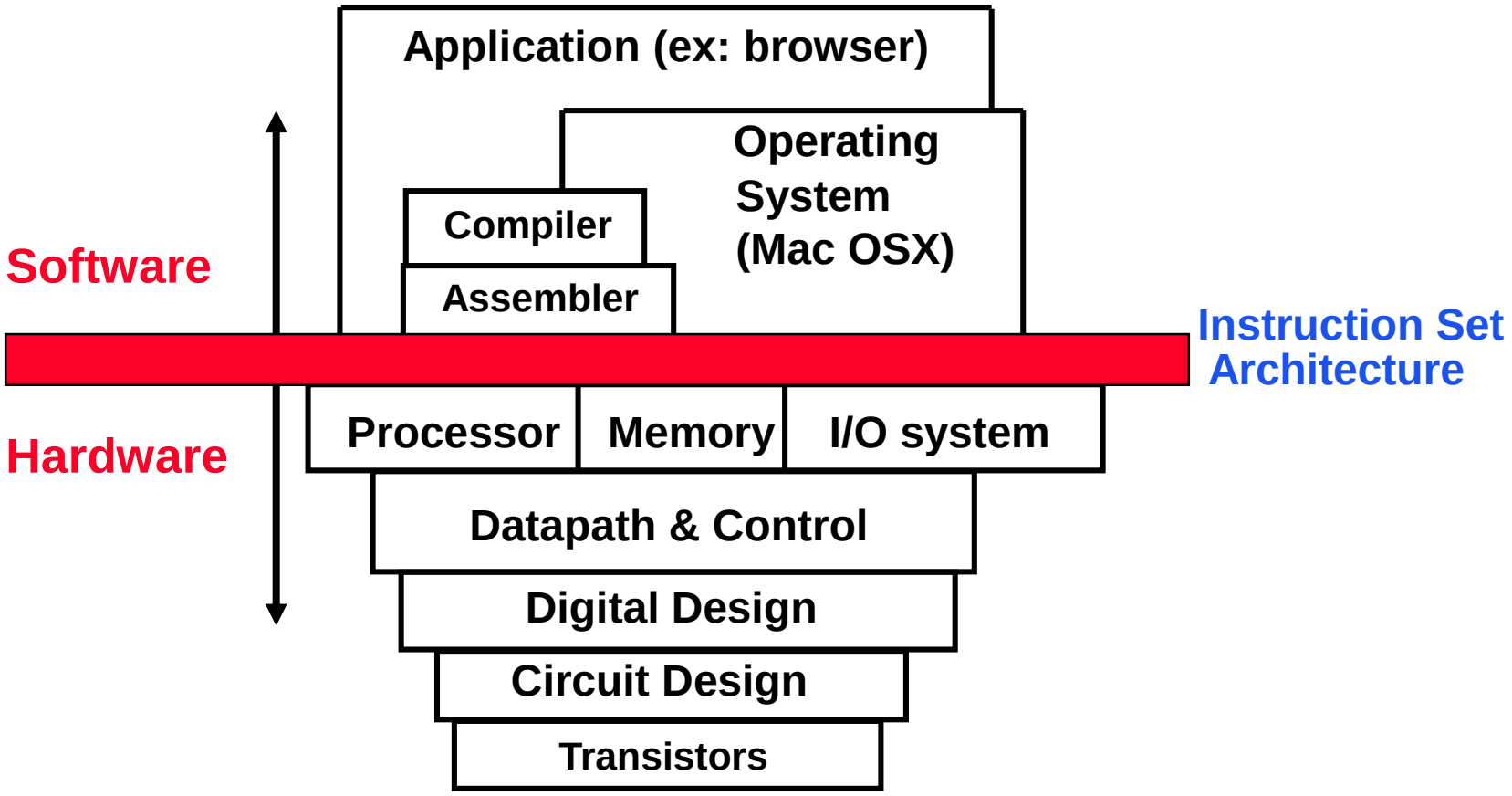
- Application software
  - Written in high-level language
- System software
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- Hardware
  - Processor, memory, I/O controllers

Levels of Program Code

- **High-level language**
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- **Assembly language**
  - Textual representation of instructions
- **Hardware representation**
  - Binary digits (bits)
  - Encoded instructions and data



Old School Machine Structures (Layers of Abstraction)



New-School Machine Structures

Software *Hardware*

- Parallel Requests
  - Assigned to computer
  - e.g., Search “Katz”
- Parallel Threads
  - Assigned to core
  - e.g., Lookup, Ads
- Parallel Instructions
  - >1 instruction @ one time
  - e.g., 5 pipelined instructions
- Parallel Data
  - >1 data item @ one time
  - e.g., Add of 4 pairs of words
- Hardware descriptions
  - All gates working in parallel
  - at same time

Harness Parallelism & Achieve High Performance

Warehouse Scale Computer

Smart Phone

Computer

Core … Core

Memory (Cache)

Input/Output

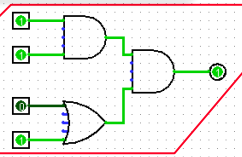
Core

Instruction Unit(s)

Functional Unit(s)

A_0+B_0 A_1+B_1 A_2+B_2 A_3+B_3

Main Memory



Logic Gates

Why do computers become so complicated?
Pursuing performance!

- **Eight Great Ideas**
  - 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

Understanding Performance

- **Algorithm**
  - Determines number of operations executed
- **Programming language, compiler, architecture**
  - Determine number of machine instructions executed per operation
- **Processor and memory system**
  - Determine how fast instructions are executed
- **I/O system (including OS)**
  - Determines how fast I/O operations are executed

What You Will Learn

- How programs are translated into the machine language
  - And how the hardware executes them
- The hardware/software interface
- What determines program performance
  - And how it can be improved
- How hardware designers improve performance
- What is parallel processing

Course Topics

1. Introduction
   - Machine structures: layers of abstraction
   - Eight great ideas
2. Performance Metrics I
   - CPU performance
   - *perf*, a profiling tool
3. Hierarchical Memory
   - The principle of locality
   - DRAM and cache
   - Cache misses
   - Performance metrics II: memory performance and profiling
   - Cache design and cache mapping techniques

Course Topics (cont'd)

4. MIPS Instruction Set Architecture (ISA)
   - MIPS number representation
   - MIPS instruction format, addressing modes and procedures
   - *SPIM* assembler and simulator
5. Introduction to Logic Circuit Design
   - Switches and transistors
   - State circuits
   - Combinational logic circuits
   - Combinational logic blocks
   - MIPS single cycle and multiple cycle CPU datapath and control
6. Instruction Level Parallelism
   - Pipelining the MIPS ISA
   - Pipelining hazards and solutions
   - Multiple issue processors
   - Loop unrolling, SSE

Course Topics (cont'd)

7. Multicore Architecture
   - Multicore organization
   - Memory consistency and cache coherence
   - Thread level parallelism
8. Parallel Performance Metrics III: Parallelism and Profiling
   - Amdahl's law, Graham and Brent's theorem
   - Parallelism, Speedup
   - Cilkview, a parallel performance analyzer
9. GPU Architecture
   - Memory model
   - Execution model: scheduling and synchronization

Student Evaluation

- Four assignments, each worth 10% of the final mark
  - Assignment 1 (memory hierarchy), due on Friday, Jan. 23rd
  - Assignment 2 (MIPS and circuits), due on Friday, Feb. 27th
  - Assignment 3 (ILP), due on Friday, March 13th
  - Assignment 4 (Multicore and TLP), due on Monday, April 6th
- Four quizzes (key concepts, 30-minute in class), each worth 5% of the final mark.
  - Quiz 1 (CPU/memory performance metrics and hierarchical memory), beginning of class on Thursday, Jan. 22nd
  - Quiz 2 (MIPS and logic circuits), Thursday, Feb. 26th
  - Quiz 3 (ILP), Thursday, March 12th
  - Quiz 4 (multicore and TLP), Thursday, April 2nd
- One final exam (covering all the course contents), worth 40% of the final mark

Recommended Text Book

Patterson & Hennessy (2011), "Computer Organization and Design: The Hardware/Software Interface“, revised 4th edition or 5th edition. ISBN: 978-0-12-374750-1

Instructor: Marc Moreno Maza

- email: moreno@csd.uwo.ca
- Office Room: MC327

- Office hours:

  Tuesdays 2:30pm - 4:30pm
  Thursdays 1:30pm - 3:30pm
  Otherwise by appointment

Teaching Assistants:

Xiaohui Chen (xchen422@csd.uwo.ca)
Ning Xie (nxie6@csd.uwo.ca)

Acknowledgements

- The lecturing slides of this course are adapted from the slides accompanied with the text book and the teaching materials posted on the WWW by other instructors who are teaching Computer Architecture courses.