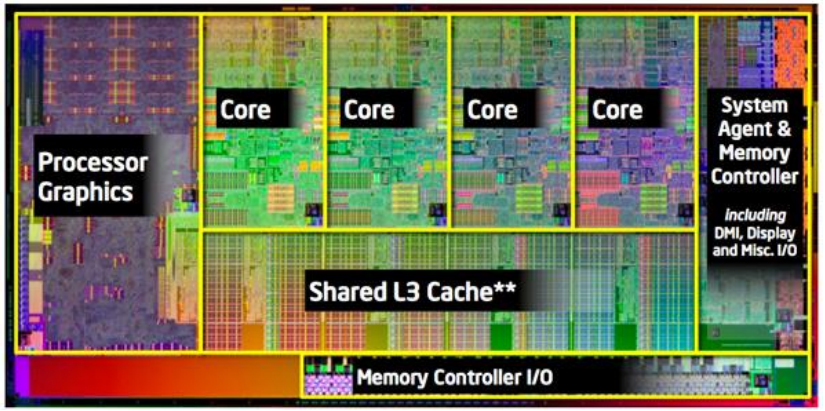
Computer Organization

Lecture 1 - Course Introduction

Reading: 1.1-1.3

Homeworks: No



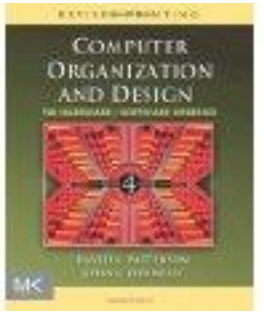
Intel “Sandy Bridge” Quad-Core Microprocessor
(Intel Corporation, 2011)

Course Introduction

- **Administrative Details**
- Computer Systems Overview
  - High-Level Organization - The “5 classic components”
  - High-Level Operations - the “fetch/execute cycle”
  - Common Abstractions
- Course Overview
  - Roadmap - subjects to be covered
  - Course Objectives
  - Project Details

Textbook and References

- **Textbook:**
  - **David A. Patterson and John L. Hennessy, *Computer Organization and Design, 4th Edition (revised),* Morgan-Kafumann, 2011**
    ((You can use the Asian Edition)
  - **Course Notes (ppt)**
    - **Do not distribute the lecture notes**
  - **Verilog Handout**



- **References (not required):**
  - **Reference: Verilog 2001 디지털시스템 설계, 홍릉과학출판사 이강 & 장경선**
  - **Simulator: Verilog simulator (uclass download)**
    - **Vrilogger**
    - **Quartus**

1st Half

Wk	Subject
1	- Course Overview (Ch. 1.1 ~ 1.3)
	- Performance (Ch. 1.4) - Advanced Performance (Ch. 1.7 ~ 1.9)
2	- Assembly Programming 1 (2.8-2.19) Introduction to PC-SPIM
3	- Assembly Programming 1 (2.8-2.19)
4	- Instruction Sets (Ch. 2.1 ~ 2.2) - Instruction Sets: MIPS (Ch. 2.3 ~ 2.7)
5	- Instruction Sets: Software Concerns (2.8-2.19)
6	- Integer Arithmetic & Logic (3.1-3.2, C.5) - Multiplication and Division (3.3 ~ 3.4)
7	- Floating Point (Ch.3.5 - 3.9)

2nd Half

Wk	Subject
9	- PI: Overview and Single-Cycle Design (Ch. 4.1-4.4, D.1, D.2) - Introduction to Verilog II (Hardware Design and Verification with Verilog (Ch.2 - Ch. 4)) - A Verilog Single-Cycle MIPS
10	- PI: Multi-cycle Processor Design
11	- PI: Pipelined Processor Design 1 (Ch 4.5 - 4.6)
12	- PI: Pipelined Processor Design II (Ch 4.7 ~ 4.9) - Pipelined MIPS Verilog Project
13	- Memory Hierarchy 1 (5.1 – 5.3)
14	-- Memory Hierarchy 2 (5.3 - 5.4)
15	– Make-up lecture (5 hours)

Grading Policy

Items	Points (%)
Middle Exam	35
Final Exam	35
Projects (1) ALU Design (2) Single cycle microprocessor design (3) Pipelined cycle microprocessor design	15
Homeworks	10
Class attendance	5
Total	100

Make-Up Schedule

법정(대체)공휴일		대체수업일	비 고
추석 연휴	09. 28.(목)	12. 14.(목)	
	09. 29.(금)	12. 08.(금)	
개천절	10. 03.(화)	12. 12.(화)	
한글날	10. 09.(월)	12. 11.(월)	

09.04(월) => ?
10.02(월) => 12.13 (수)?

Prerequisites

- **Digital design (논리회로)**
  - **Combinational logic (조합회로):**
    - **Gates (AND, OR, NAND, XOR, etc)**
    - **Common building blocks: multiplexers, adders, etc**
  - **Sequential logic (순차회로):**
    - **Flip-Flops**
    - **FSMs**
    - **Clocking and Timing**

Helpful but not Required

- **High-Level Language (Java) Programming**
  - **Variables, Primitive Data Types, and Expressions**
  - **Assignments, Conditionals, Loops, etc.**
  - **Classes, Objects, & Methods**
  - **Basic Input/Output**
  - **Arrays & Basic Data Structures**

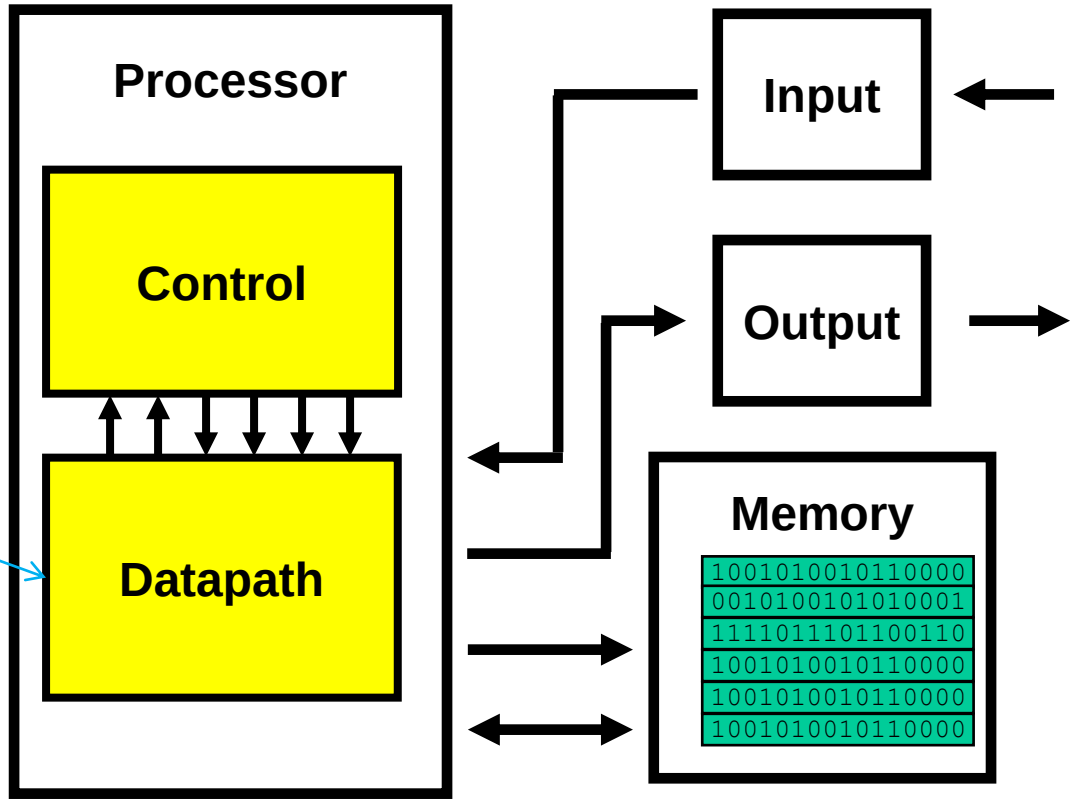
Course Introduction

- Administrative Details
- **Computer Systems Overview**
  - **High-Level Organization - 5 classic components**
  - **High-Level Operations - fetch/execute cycle**
  - **Common Abstractions**
- Course Overview
  - Roadmap - subjects to be covered
  - Course Objectives
  - Project Details

Computer System Organization

- **“Five classic components”**

A command is executed by moving through its own particular path



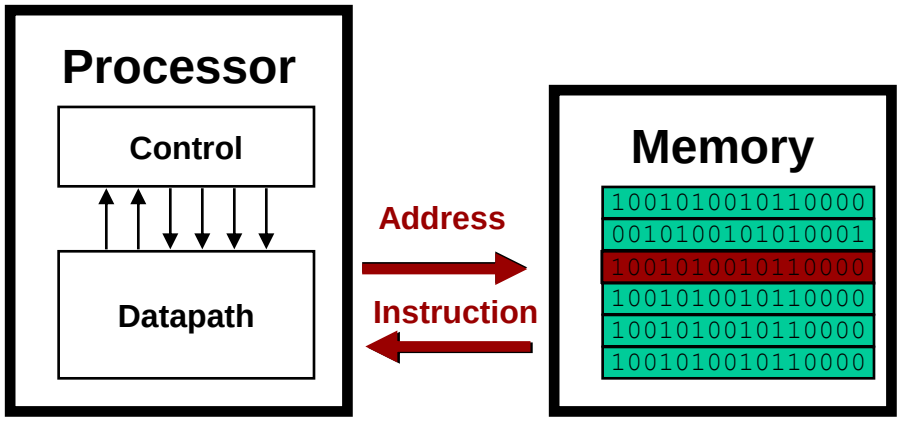
Computer System Operation

- **Executing Programs - the "fetch/execute" cycle**
  - **Processor fetches instruction from memory**

next instr

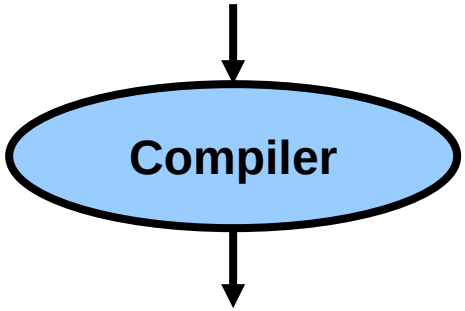
  - **Processor executes instruction (*machine language*)**

Load Data
Perform Calculation
Store Results

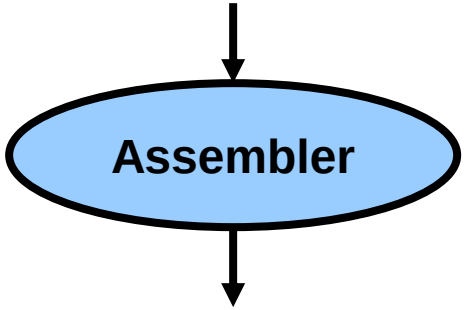


Software Abstractions - Languages

High-Level Language (C) `c = a + b;`



Assembly Language `add R8,R1,R2`



Machine Language `00000000010001001000000100000`

See Fig. 1.3 for a more detailed example

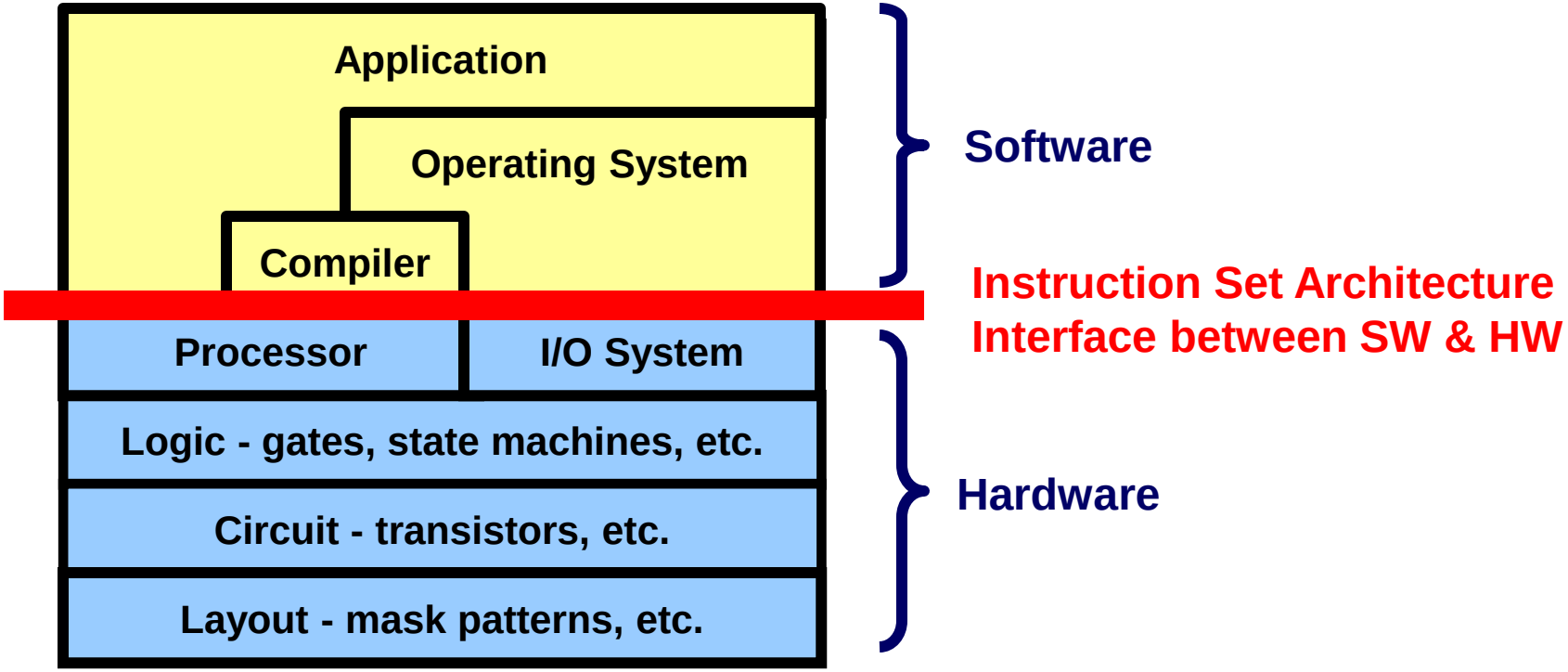
Software Abstractions - System Software

- **Operating system**
  - **Insulates programmer from low-level details**
    - **Manages system resources (자원관리)**
    - **Manages file system (파일시스템)**
  - **Coordinates operation of multiple programs (스케쥴링)**
  - **Protects from system or from damage by user programs (accidental or malicious)**
  - **Programs interact w/ OS through system calls**

- **Libraries**
  - **Provide programmer with access to high-level "primitives"**
  - **Programs access through well-defined interface (API)**

Instruction Set Architecture

- The **most important abstraction** of computer design



Architecture vs. Organization

- **Architecture: features visible to programmer**
  - **Registers and memory model**
  - **Data types**
  - **Instructions**

- **Organization: system implementation**
  - **Processor design: Datapath, Control, "microarchitecture"**
  - **System design: Processor + Memory, I/O**

Course Introduction

- Administrative Details
- Computer Systems Overview
  - High-Level Organization - The “5 classic components”
  - High-Level Operations - the “fetch/execute cycle”
  - Common Abstractions
- **Course Overview**
  - **Roadmap - subjects to be covered**
  - **Course Objectives**
  - **Design Projects**

Roadmap for the Term: Major Topics

Performance

Instruction Sets (and Software)

Logic and Arithmetic

Processor Implementation

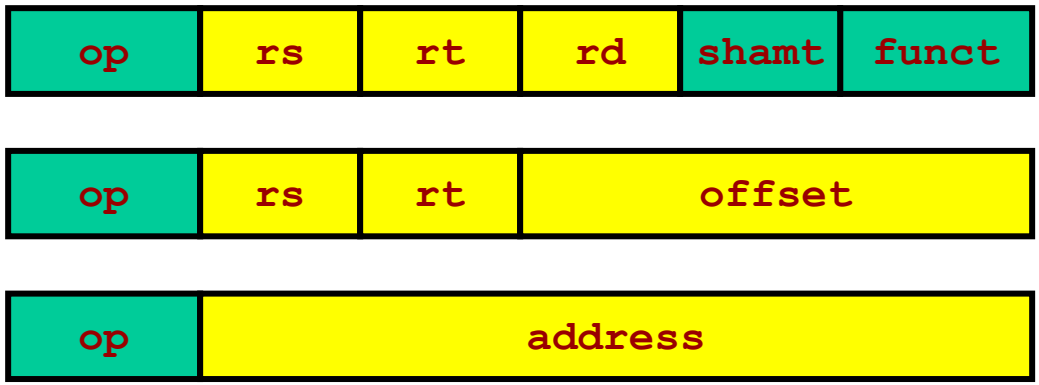
Memory Systems

Performance

- Response Time vs. Throughput
- Measuring performance using individual programs
- Combining measurements
- Benchmarks

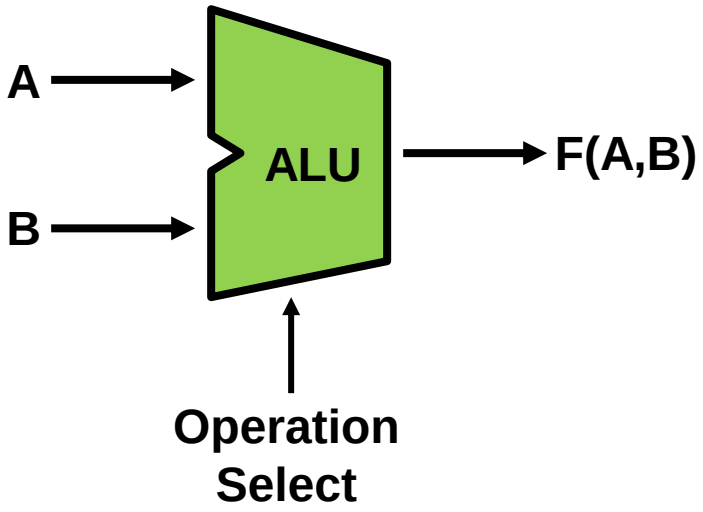
Instruction Sets (and Software)

- General principles of instruction set design
- The MIPS instruction set
- Software concerns: procedures, stacks, etc.



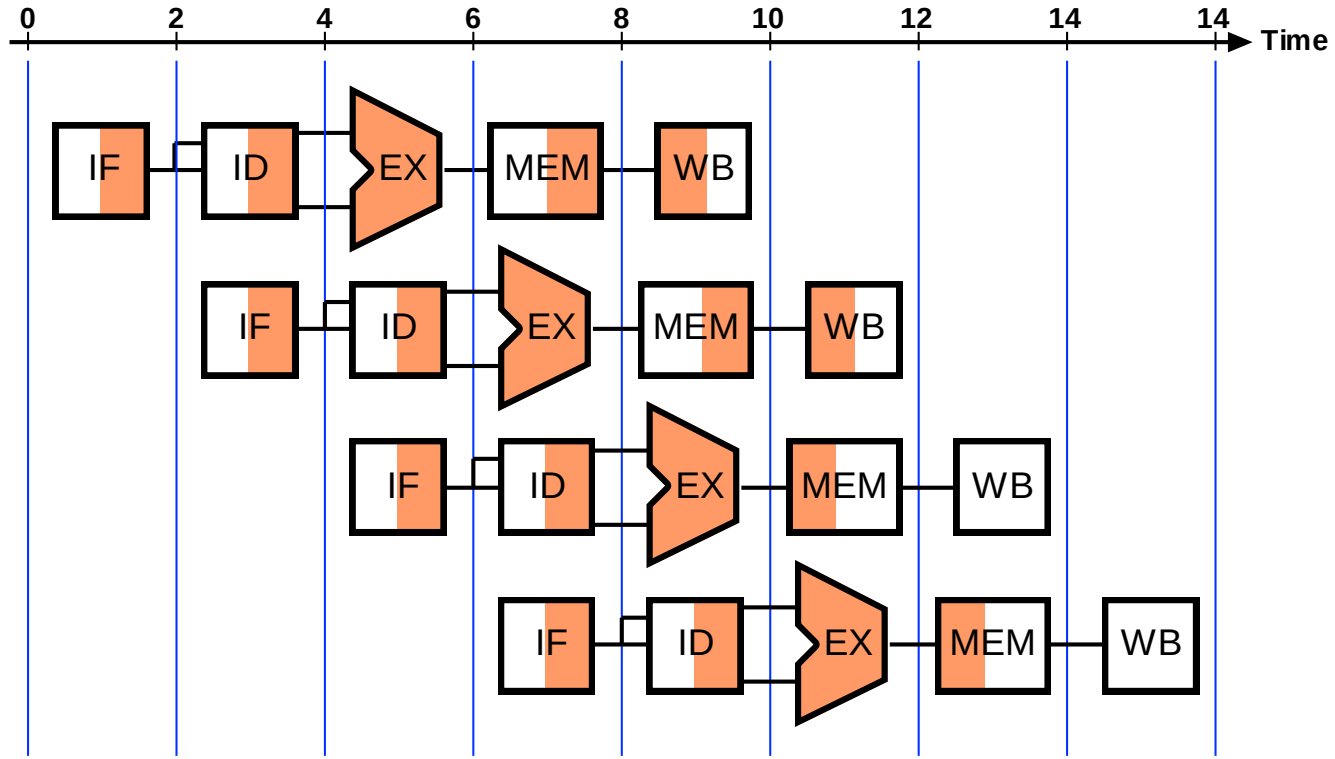
Logic & Arithmetic

- Quick review: binary numbers and arithmetic
- Adder & ALUs; multiplication & division
- Floating Point



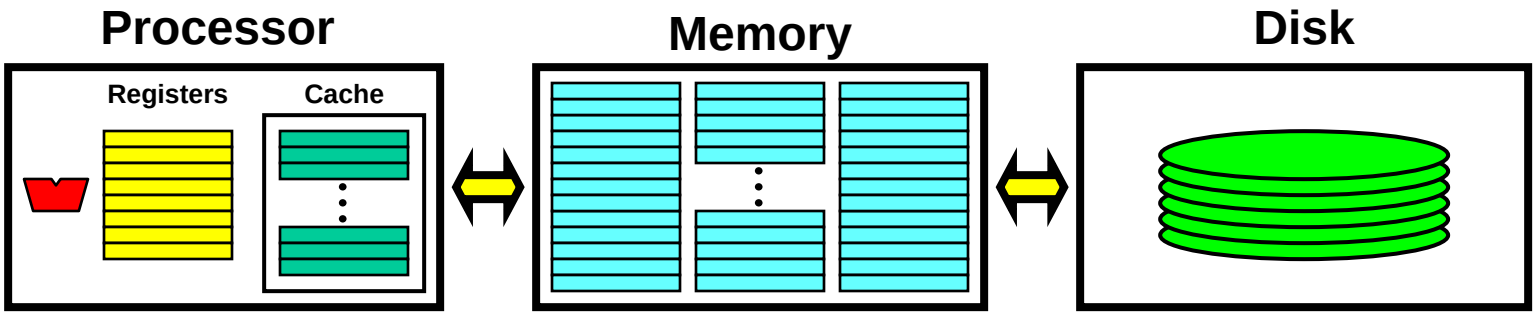
Processor Implementation

- Basic implementation
- Pipelined implementation



Memory Systems

- Memory Technology Overview
- Memory Hierarchy
  - Cache Memories - making access faster
  - Virtual Memory - making memory larger using disk



Input/Output

- I/O Overview
- Impact of I/O on Performance
- Buses
- Interfacing

Image Source:
Seagate Technolgy LLC
www.seagate.com

Course Objectives

- **You should be able to evaluate the performance of computers comparatively.**
- **You should be able to define a new Instruction Set Architecture (ISA)**
- **You should be able to design a new microprocessor according to the ISA you have designed**
- **You have to understand the memory architecture and how you can resolve the performance limitation of memory system**

Design Projects

- **Design modeling and simulation using Verilog HDL**
  - **(Design Project 1) Arithmetic/Logic Unit (ALU) design**
  - **(Design Project 2) Single-cycle processor design**
  - **(Design Project 3) Pipelined processor design (group project)**

- **Design tool of choice**
  - **Quartus**
  - **Verilog**

  etc