CSSE232
Computer Architecture

Introduction

Reading

- Better for you if done before class

- For today:
  - Ch 1 (esp 1.1-3, 10)
  - App. C
  - Sections 2.4, 3.1-2

Outline

- Introductions
- Class details
  - Syllabus, website, schedule
- History of computing
- Moore's Law
- Class outline
  - Parts of a computer
  - Program processing
  - Introduction to MIPS
  - Project

Introduction

- Introductions
  - Name/nickname
  - Location on campus
  - One thing you enjoy or are good at
- Student assistants
- Instructor

Class details

- Syllabus on course webpage
  - http://www.rose-hulman.edu/Class/csse/csse232/
- Submit homework hardcopies in class
- Submit labs through SVN
  - You will be given a repository
  - csse232-1314b-*yourusername*
- Project submission will be discussed later

Quick poll

- How many SE? CS? CPE?

- Anyone else?

Five classic components of a computer

- Same components for all kinds of computers (Desktop, server, embedded)

Five classic components of a computer

- Same components for all kinds of computers (Desktop, server, embedded)
  - Input
  - Output
  - Memory - stored program model (von Neumann)
  - Datapath – performs arithmetic operations
  - Control – tells the datapath, memory, and devices what to do

- Difference between hardware/software?
  - Both implement algorithms
- What is a computer?
  - Input, output, memory, processor
  - Processor : datapath, control
- How old is computing?
  - 1943, enigma, Alan Turing, Blechly Park, Colossus
  - 1830, Charles Babbage, Analytical engine
  - 1803, Jacquard loom
  - 1951, UNIVAC, 1st commercial computer

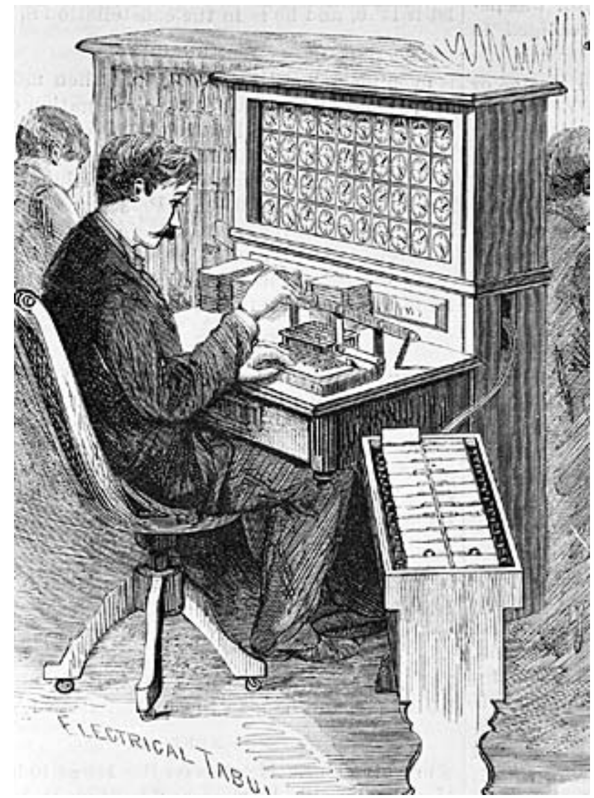
History of Computers

- Mechanical / Electromechanical
- Vacuum tube
- Transistor
- Integrated circuit
- Very Large Scale Integration (VLSI) / Microprocessor
- Ultra Large Scale Integration (ULSI) / Microprocessor

Mechanical/Electromechanical

Jacquard's Loom
1805

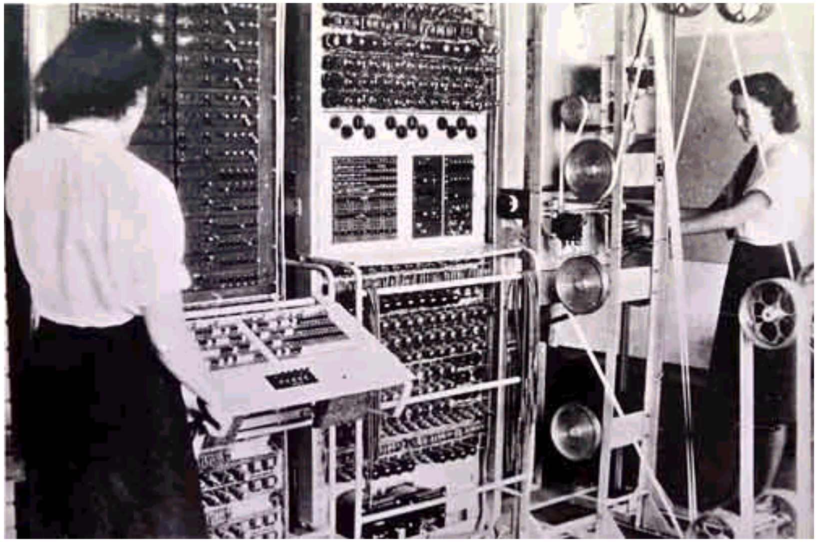
Babbage's engine
1833,1837,1853



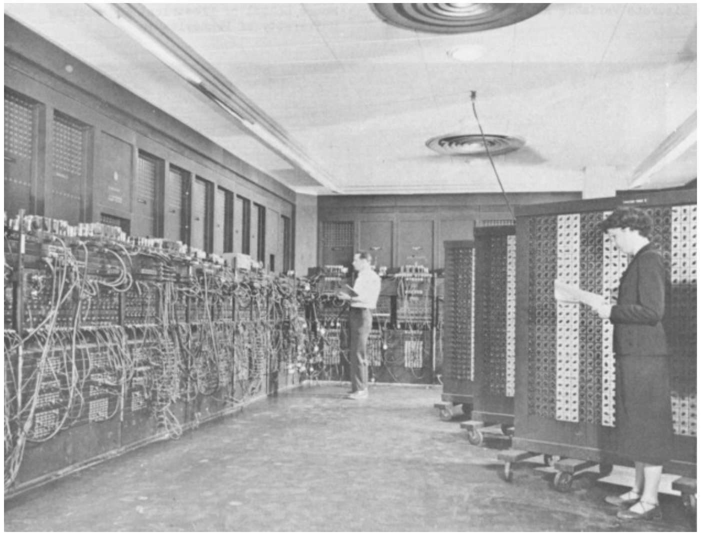
Hollerith's Census Tabulator
1890

http://www.history.rochester.edu/steam/hollerith/loom.htm
http://www.computersciencelab.com/ComputerHistory/History.htm

Vacuum Tubes



COLUSSUS
1943

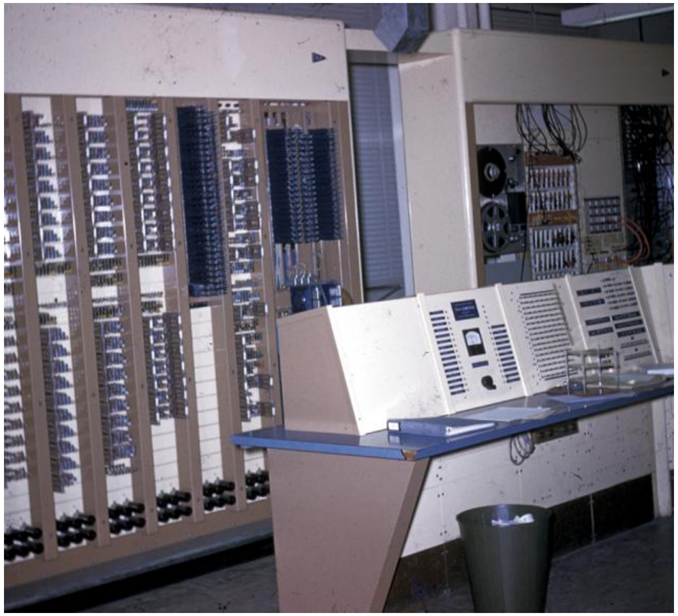


ENIAC
1946

http://www.computersciencelab.com/ComputerHistory/History.htm

Transistors

CDC 1604
1960



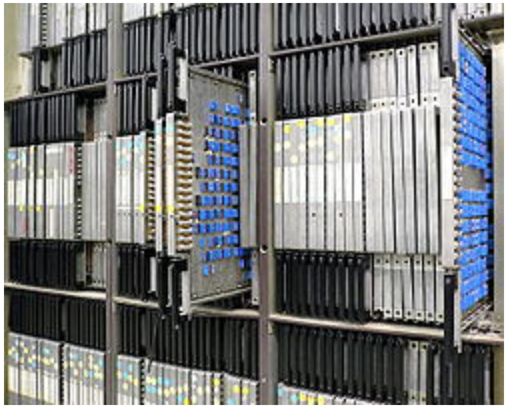
TX-0
1955

Replica of first working transistor

http://tixo.org/image/tx-0-console?size=preview
http://www.billdolson.com/history/ui/fluids/cdc1604.htm

Integrated Circuits

IBM 360
1965



Illiac IV
1976

http://en.wikipedia.org

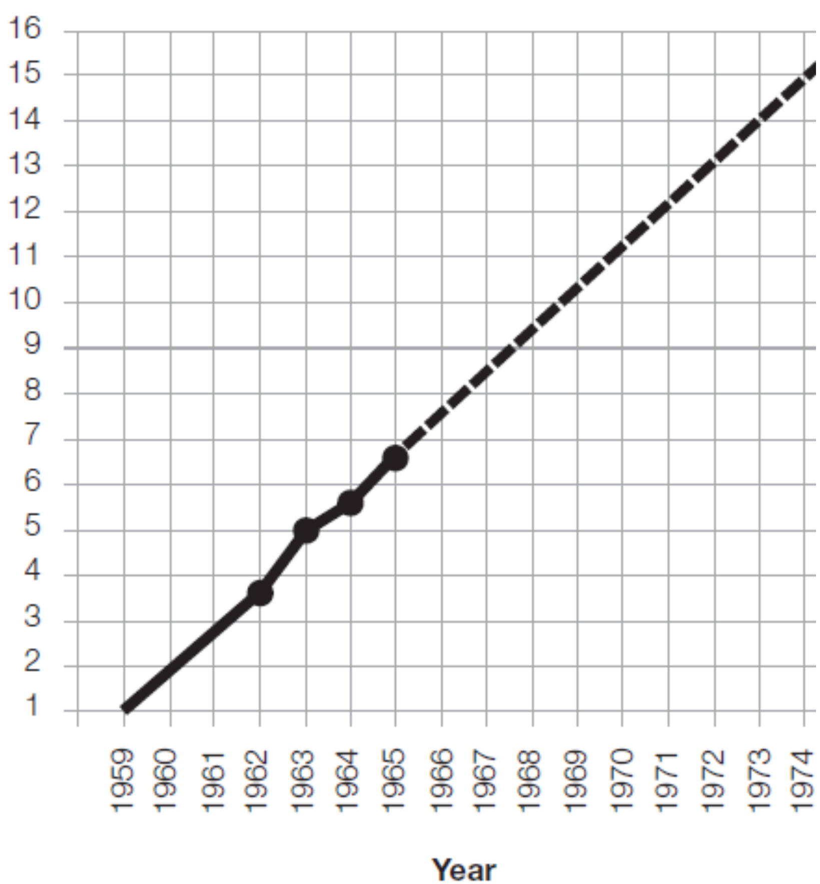
VLSI and ULSI

- Thousands of transistors on chip
- Entire system on chip
- Parallel processing

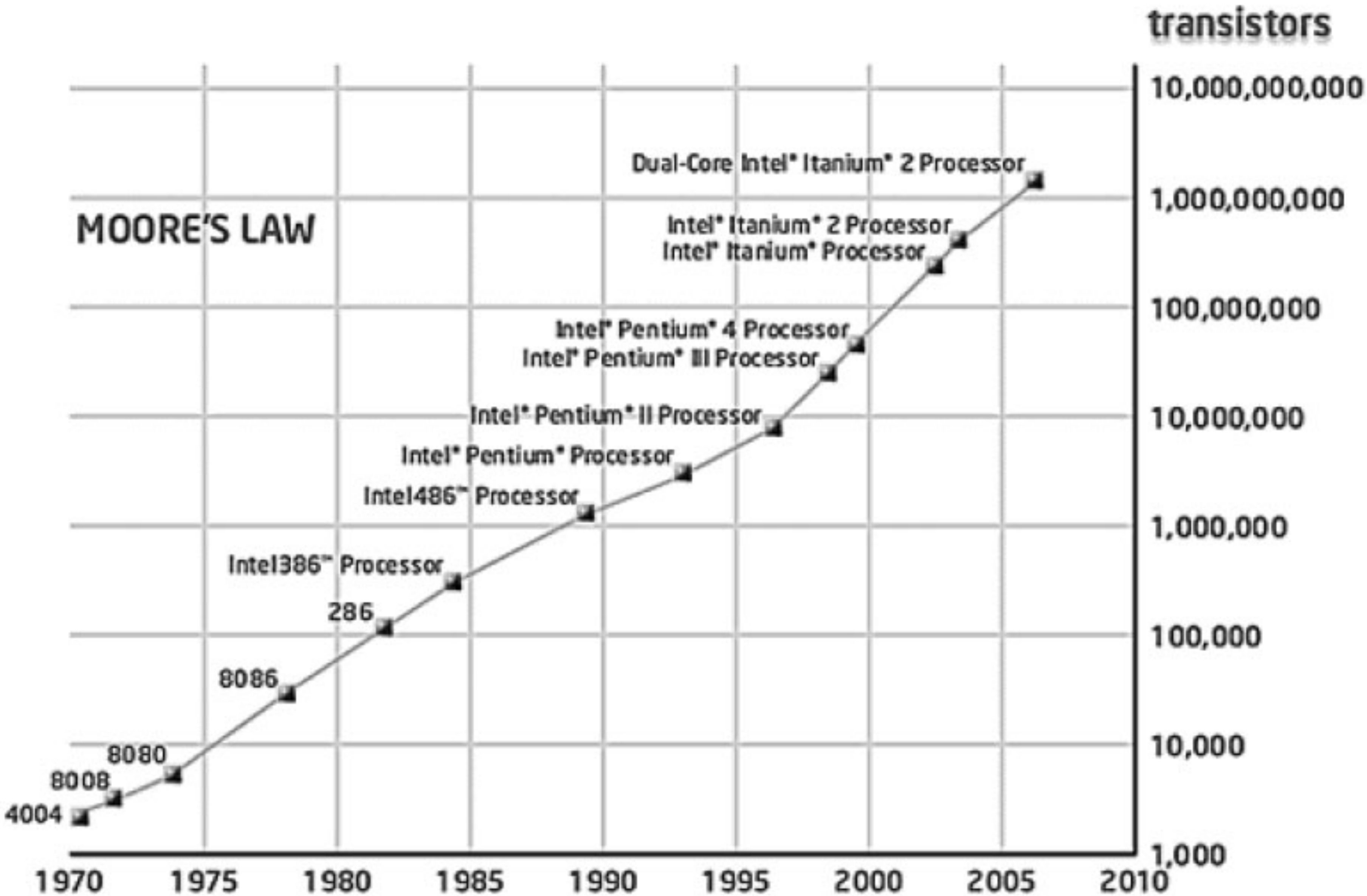
1982 vs 2010
28.75 lbs/ 0.3 lbs = ~100 times heavier
4MHz / 1 GHz = 250 times slower
$2500 / $500 = 5 times more expensive
(52cm x 23cm x 33cm)/(11.5cm x 5.86cm x 0.93cm) = 629 times as large

http://en.wikipedia.org/wiki/File:Osborne_Executive_with_iPhone_in_2009.jpg

Moore's Original Prediction

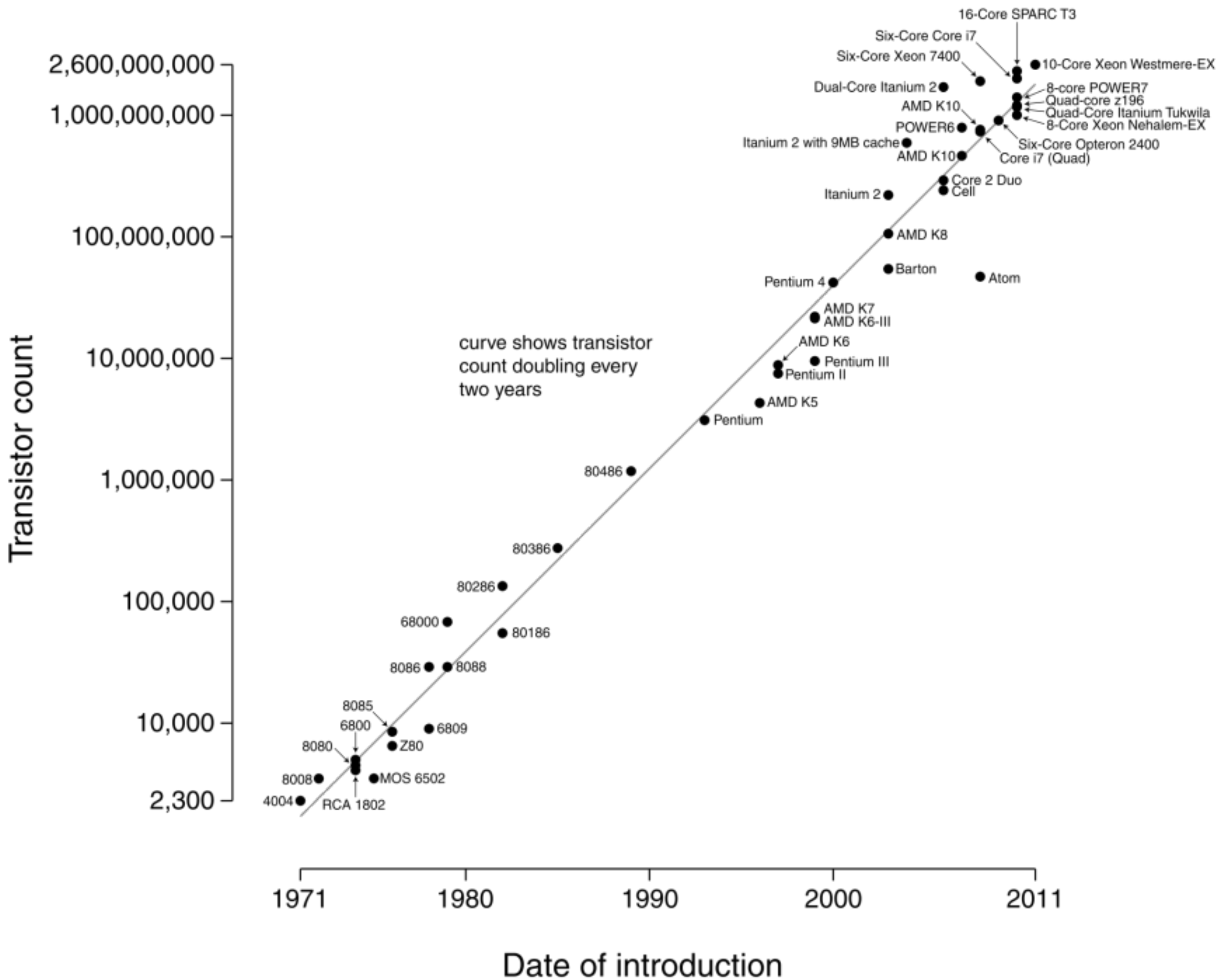


- Moore predicted that transistor counts would double every 2 years
- Not really a law, just an observation



http://www.isgtw.org/images/mooreslaw.jpg

Microprocessor Transistor Counts 1971-2011 & Moore's Law

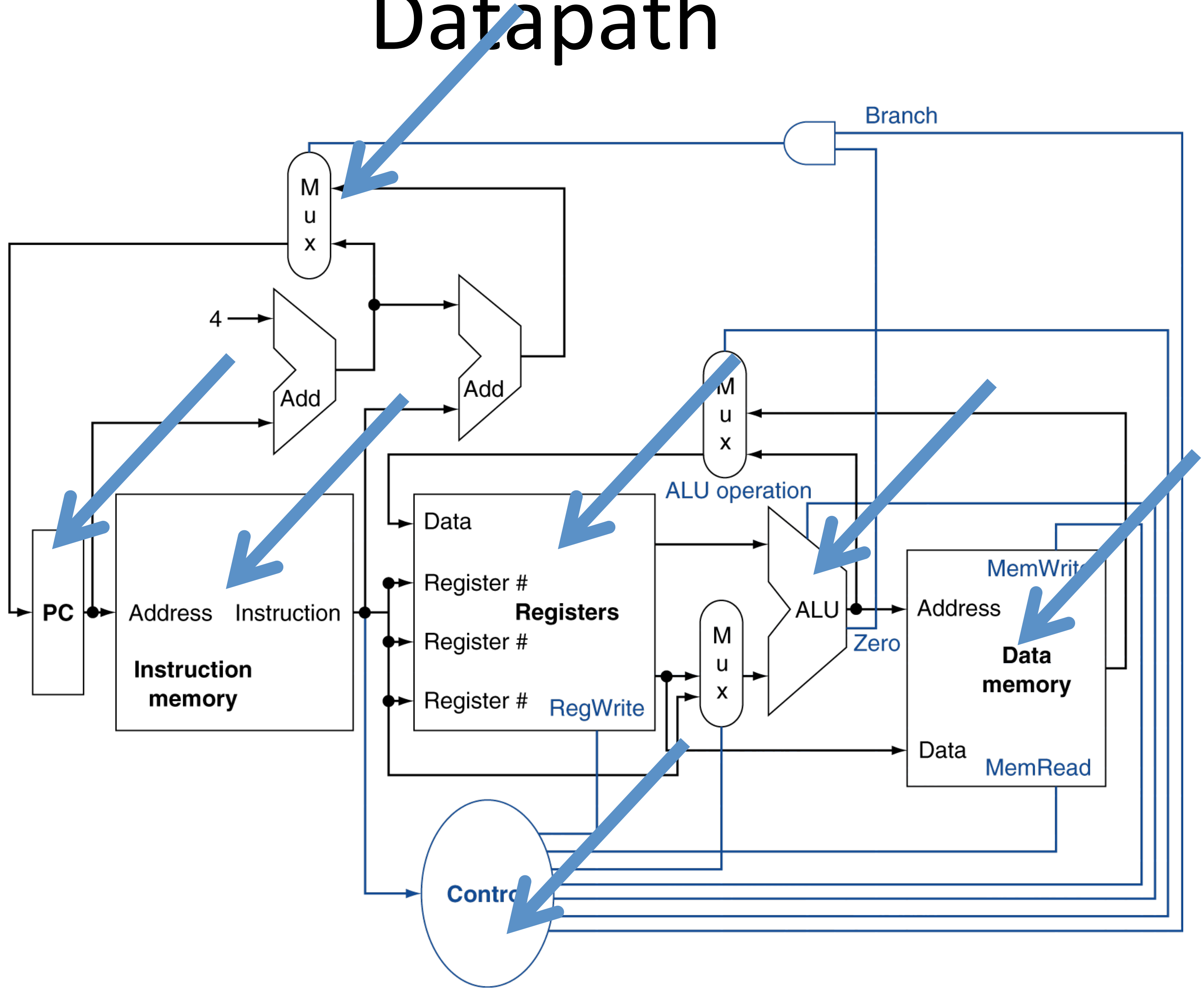


Computers for us

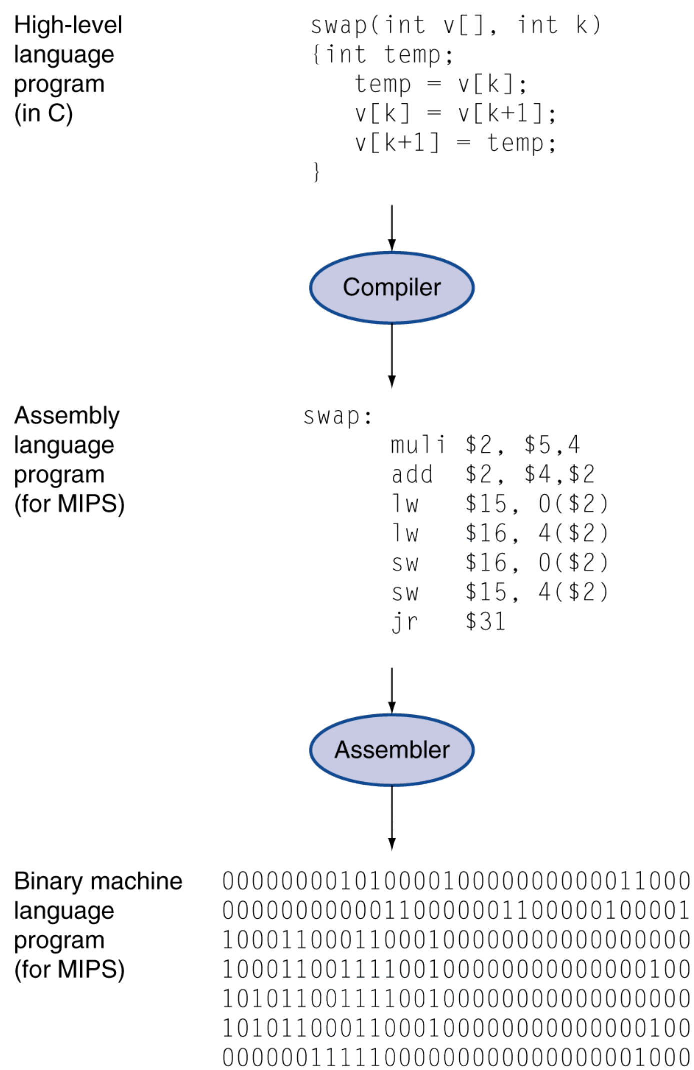
- Datapath + control
- I/O
- Memory

- Instructions to make drive the whole engine
  – Instruction set

Datapath



Branch
M
u
x
4
Add
Add
M
u
x
ALU operation
Data
Register #
Register #
Register #
Registers
RegWrite
PC
Address
Instruction
Instruction
memory
M
u
x
ALU
Zero
MemWrit
Address
Data
memory
Data
MemRead
Contr


High-level language program (in C)
swap(int v[], int k)
{int temp;
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
}
Compiler
Assembly language program (for MIPS)
swap:
muli $2, $5,4
add $2, $4,$2
lw $15, 0($2)
lw $16, 4($2)
sw $16, 0($2)
sw $15, 4($2)
jr $31
Assembler
Binary machine language program (for MIPS)
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000

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

What is an Instruction Set?

To command a computer, you must speak its language.

The words of a computer are called instructions, and its vocabulary is an instruction set.

The MIPS Instruction Set

- Used as the example throughout the book
- Stanford MIPS commercialized by MIPS Technologies (www.mips.com)
- Large share of embedded core market
  - Applications in consumer electronics, network/storage equipment, cameras, printers, …
- Typical of many modern ISAs
  - See MIPS Reference Data tear-out card, and Appendixes B and E

Android and MIPS

http://www.androidauthority.com/android-jelly-bean-mips-source-code-107257/

MIPS Design Principles

1. **Simplicity favors regularity**
   - All instructions single size
   - Always requires three register operands in arithmetic instructions
   - Register fields always in the same place
2. **Smaller is faster**
   - Only 32 registers
3. **Make the common case fast**
   - PC-relative addressing for conditional branches
   - Immediate addressing for larger constant operands
4. **Good design demands good compromise**
   - Compromise between providing for larger addresses and constants in instructions and keeping all instructions the same length

Course outline

- We will learn
  - CPU performance metrics
  - MIPS instruction set architecture
  - Assembly language programs
  - CPU datapath design
    - Types
    - Components
  - Project – build CPU

Project

- Teamwork (3 or 4)
- Design a "miniscule instruction set" general purpose processor that can execute programs stored in an external memory
- Model your design, test it, debug it, assess its performance, and possibly implement it on a Field Programmable Gate Array (FPGA) microchip
- Maintain current documentation
- Presentations

Project (cont.)

- Your processor must be capable of executing programs stored in an external memory with which it communicates using:
  - A 16-bit address bus, and
  - A 16-bit data bus.
- Further, your processor should support:
  - Interrupts from two input devices,
  - Reading from a 4-bit input port,
  - Reading from and writing to a special 16-bit display register, and
  - Displaying the contents of the display register on the LCD display via a 16-bit output port.

Project (cont.)

- Your instruction set:
  - Must be capable of performing general computations, and
  - Must support parameterized and nested procedures.

Lab0 - ioBlockPart

- Install Xilinx Tools (ISE 13.4)
- Run ISE 13.4
- Test your installation
  - Boards are in F217
- Modify the project
- Due date on website

HW0

- Review of CSSE132
- Material we will cover in the upcoming days
- Due date on website

Recap

- Introductions
- Class details
  - Syllabus, website, schedule
- History of computing
- Moore's Law
- Class outline
  - Parts of a computer
  - Program processing
  - Introduction to MIPS
  - Project