EE 306, Introduction to Computing

Jacob Abraham

Department of Electrical and Computer Engineering
The University of Texas at Austin

Introduction to Computing
Fall 2018

August 29, 2018

EE 306, Introduction to Computing: The first course in computing for EE and CE majors

Course materials on **Canvas:**
https://utexas.instructure.com/courses/1101483

Course syllabus and Lecture Notes:
http://www.cerc.utexas.edu/~jaa/ee306/

Discussions

Discussion forum for course using **Piazza:**
piazza.com/utexas/fall2018/ee306abraham

Course Information

Instructor

Jacob A. Abraham, EER 4.874, 512-471-8983,
jaa@cerc.utexas.edu

Office Hours: 5:00 – 6:30 pm, EER 4.704

Teaching Assistants

- Yuxin Wang
- Hyunsu Chae

Textbook

Introduction to Computing Systems: from bits and gates to C and beyond, Yale N. Patt and Sanjay J. Patel; Mc-Graw Hill, 2004, 2nd edition

Course Topics

First 10 chapters in textbook

- Bits, Data Types, and Operations
- Digital Logic Structures
- The Von Neumann Model
- The LC-3
- Programming
- Assembly Language
- I/O
- TRAP Routines and Subroutines
- And, Finally...The Stack

Course details

Class meets MW 3:00 – 4:30 pm, EER 1.516

No attendance taken, but lectures are crucial for understanding the material covered in the class

Discussion sessions on Thursday (attend session of your choice, collect graded assignments from TAs for your section)

Prerequisites

- None, and no programming experience is assumed
- Students need to be highly motivated
- Need to put in the time and energy to support motivation

Homework Policy

- 6 Homework assignments
- Students encouraged to form study groups
  - Turn in only one copy for the group

Course details (continued)

5 programming assignments

- May discuss structure of the program with other members of group
- However, the detailed algorithm and actual coding must be individual
- See handout, Overview of EE 306 for more information on what constitutes cheating

Exams

- Two exams in class, October 31, November 3
- Final Exam, to be announced by registrar

Exams are closed book. You may bring in three sheets of paper, on which is written anything you wish, in your own handwriting

Course Grade

- Homework (problem sets) (2% each, times 6 assignments): 12%
- Programming assignments (5% each, times 5 assignments): 25%
- Two in-class exams (19% each): 38%
- Final exam: 25%

Problem sets must be turned in before class on the due date
Programming assignments are due by 11:59 pm on the due date

About the Instructor

- Born in Kerala, India
- Bachelor's degree in Engineering, University of Kerala, 1970
- M.S. Stanford, 1971
- Wrote production code in Assembly Language for Hewlett-Packard as part of a summer job
- Ph.D., Stanford, 1974
  - Electrical Engineering, with a minor in Computer Science
- Taught at the University of Illinois, 1975 – 1988
- At UT Austin since 1988
- Academic interests: VLSI design, test and verification; fault tolerance; security, neuroscience

Home page: `http://www.cerc.utexas.edu/~jaa/`

Objectives of the Course

- Serious introduction to the fundamental underpinnings of computing
- Bottom-up approach to understand how computers work
- Understanding of computer hardware
- Sufficient programming methodology to get a computer to do useful work
- Student should gain a deeper understanding of why and how computers work

What is Expected?

Of You

- Read the textbook
- Lectures, discussion sessions, textbook and homework assignments are all different ways to help learn the material
- Work in groups to do the homework problems – learn from each other
- Learn the fundamentals – this will help you greatly in future classes

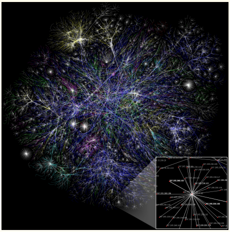
Of the Instructor

- Help you learn the material – lectures, office hours, special sessions
- Help you get through this class
- Help you to learn to **think** – a skill which will be useful

Computers are Everywhere

Google's autonomous car

Internet map
(source: Wikipedia)



Universal Computing Device

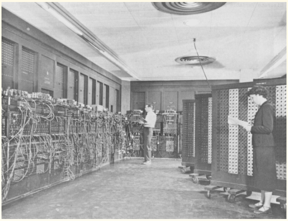
PDA

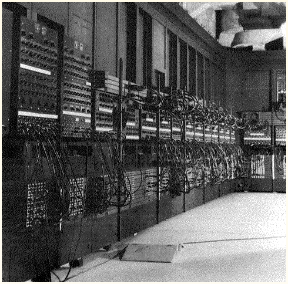
Workstation

UT Ranger Supercomputer

One of the First Computers

ENIAC





“Smarter” than Humans?

IBM Watson

Jeopardy Champion

Integrated Circuits of the Future?

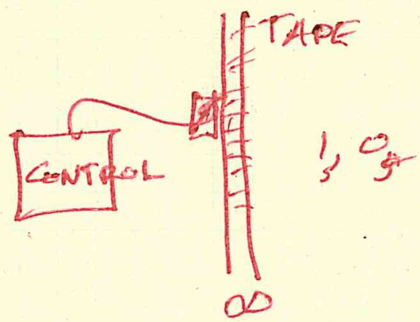
Star Trek Tricorder

Star Fleet Tricorder, 2383

X-Prize Challenge

ABSTRACTIONS

FOR A COMPUTER – "TURING MACHINE"



MODEL OF A MACHINE THAT CAN PERFORM ANY COMPUTATION

COMPUTER – UNIVERSAL TURING MACHINE

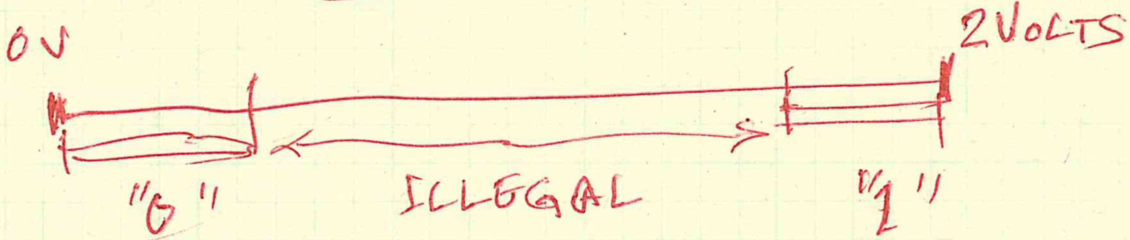
NUMBERS IN THE PHYSICAL WORLD

VOLTAGE (CONTROL ELECTRONS)

↳ SOME VOLTAGE (EX: 2V) — "1"

NO VOLTAGE (0V) — "0"

↳ BINARY



BIT - "BINARY DIGIT"

2 BITS: 00, 01, 10, 11

3 BITS: (OCTAL) - 2^3 COMBINATIONS

000, 001, 010, 011, 100, 101, 110, 111

n BITS ⇒ 2^n VALUES

CAN REPRESENT -

NUMBERS

TEXT

IMAGES

SOUND

LOGIC

DECIMAL: 0, 1, 2, ... 8, 9 [BASE 10]

2145

10^3 10^2 10^1 $\times 10^0$

WHAT IS 2145_{10} IN BASE 6?

2145

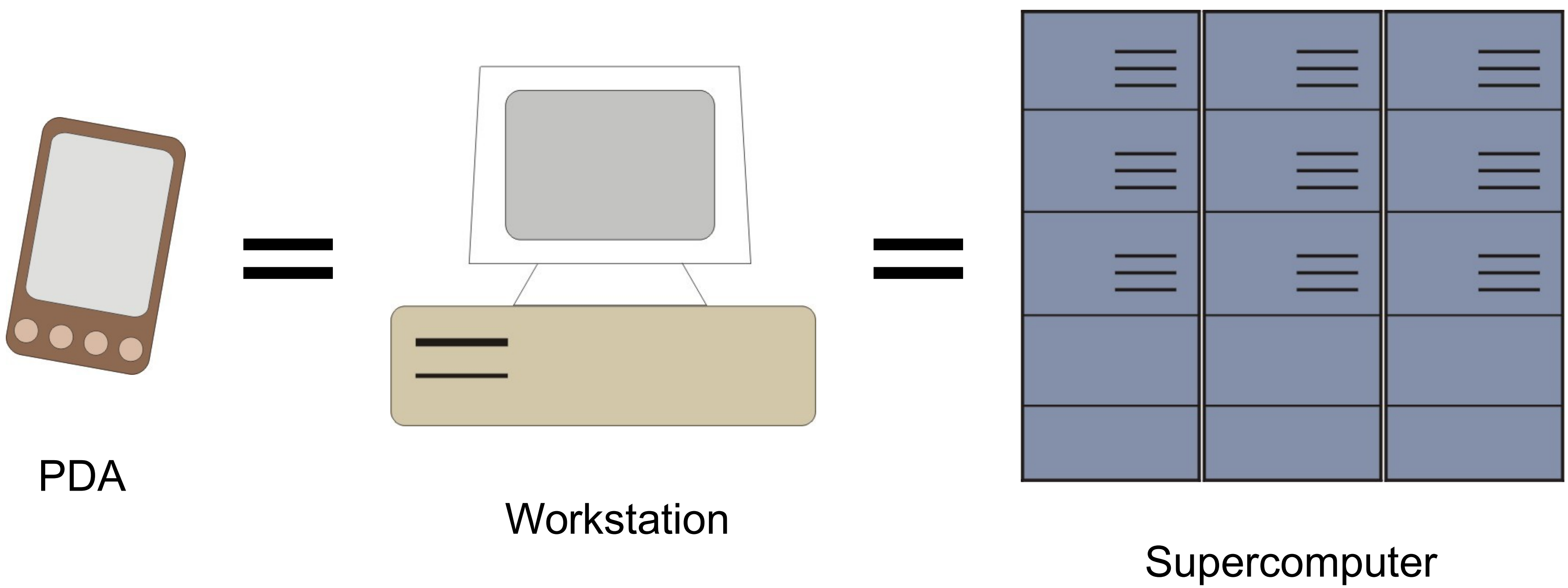
6^3 × 6^2 × 6^1 × 6^0 ×

UNARY: (4_{10}) – ||||

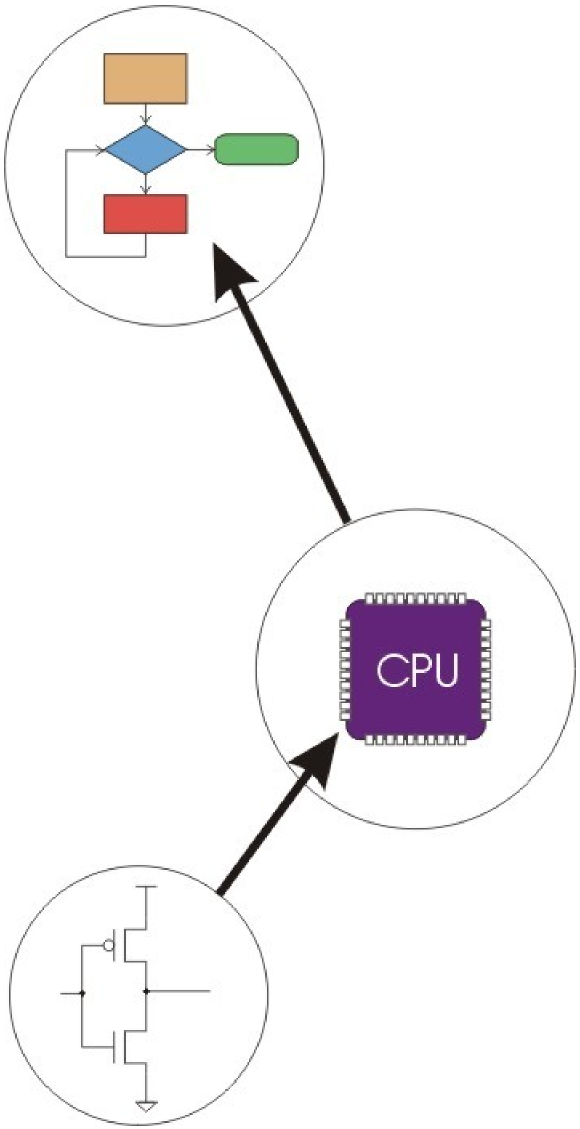
(ROMAN) XLV = 45_{10}

2018: MMXVIII

Universal Computing Device



Transformations Between Layers



Problems

Algorithms

Language

Instruction Set Architecture

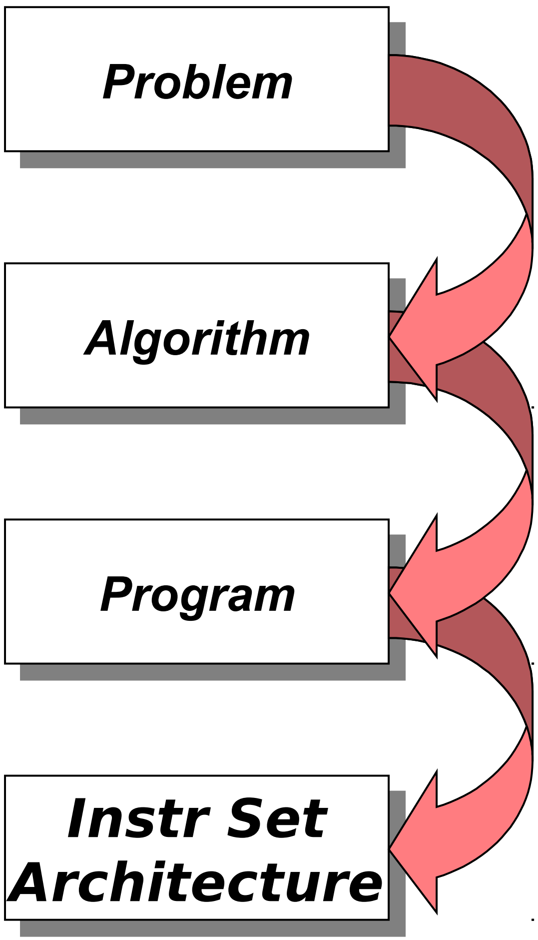
Microarchitecture

Circuits

Devices

How do we solve a problem using a computer?

A systematic sequence of transformations between layers of abstraction.

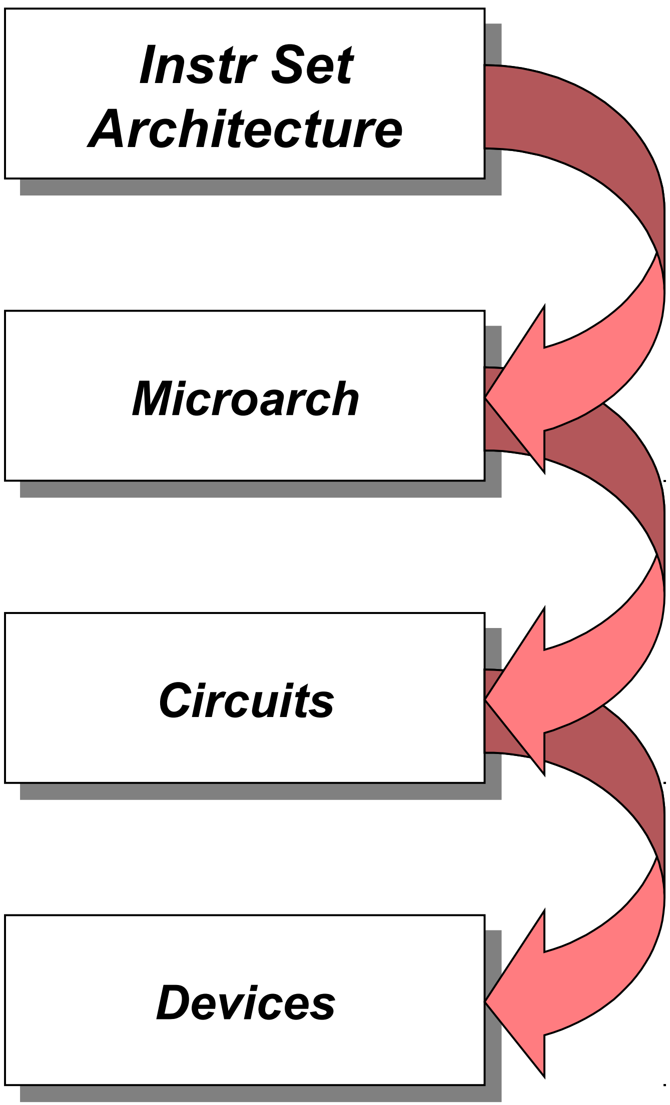


Software Design:
choose algorithms and data structures

Programming:
use language to express design

Compiling/Interpreting:
convert language to
machine instructions

Deeper and Deeper…



Processor Design:
choose structures to implement ISA

Logic/Circuit Design:
gates and low-level circuits to implement components

Process Engineering & Fabrication:
develop and manufacture lowest-level components

Descriptions of Each Level

Problem Statement

- stated using "natural language"
- may be ambiguous, imprecise

Algorithm

- step-by-step procedure, guaranteed to finish
- definiteness, effective computability, finiteness

Program

- express the algorithm using a computer language
- high-level language, low-level language

Instruction Set Architecture (ISA)

- specifies the set of instructions the computer can perform
- data types, addressing mode

Descriptions of Each Level (cont.)

Microarchitecture

- detailed organization of a processor implementation
- different implementations of a single ISA

Logic Circuits

- combine basic operations to realize microarchitecture
- many different ways to implement a single function (e.g., addition)

Devices

- properties of materials, manufacturability