CSIS 212 : Machine Organization & Assembly Language

Duy Nguyen, Ph.D.

dnguyen@palomar.edu

858.204.5232 (cell)

Contact Infomation

- Duy Nguyen, Ph.D.
- Email: dnguyen@palomar.edu
- 858.204-5232 (cell)
- Office Hours: Mon 10-11:00, Wed 10-11:00 & 1:30-2pm

- Julie Thurston-Donaghy, CSIS Academic Department Assistant
- Email: jthurston-donaghy@palomar.edu
- Phone: 760-744-1150 x2387
- Office: MD-275

- CSIS Department: http://www.palomar.edu/csis/
- Palomar College: http://www.palomar.edu
- Palomar College Academic Technology: http://www.palomar.edu/atrc/

Outline

- Syllabus discussion

- Homework & Lab submission policies

- Course Overview

- Computer Arithmetic

- Basic Data Type

- Raspberry Pi, Linux, text editor, compiler, makefile, gdb debugger, etc…

Assignments Submission

- **Each written homework assignment will be graded out of 2 points**
  - **If it looks as if you put an effort into the homework assignment and have mostly correct answers, you will receive a 2 on the assignment**
- **Grades are posted in the Grade Center and solutions are posted in the Homework Solutions link after the due date**
- **The deadlines for the homework assignments are set at Sunday, midnight**
  - **All assignments will be posted on Canvas and submitted via Canvas**

Assignments Submission Continued…

- **Be sure that your full name and Palomar ID number appear on your homework assignment**
- **Save your written homework file as a Word document**
- **Name your Word document LastFirst (e.g., NguyenDuy)**

Labs Submission

- Each Lab is worth 10 points
- Your full name and Palomar ID number must appear as a comment in the source file that contains main()
- Place the following files into a folder and be sure to include only those files I have requested:
  - all C and ASM source files required to build the project including all interface files
- Rename the folder LastFirst using your own name (i.e., NguyenDuy)
- Compress the folder into a zip archive

Lab Submission Continued…

- The deadlines for the computer lab assignments are set. When the deadline passes, you will have three days to submit the computer lab assignment late with a penalty of 1/10 points.
- Labs will not be accepted after the late submission deadline has passed. Your grade for the lab assignment will be 0/10 if it is not submitted
- The deadlines for the Labs are set at Sunday, midnight
  - All lab assignments will be posted on Canvas and submitted via Canvas

Grades

Tentative grading policy will be as follows:

- **15% - Homework**
- **15% - Midterm Exam**
- **15% - Final Exam**
- **55% - Computer Labs**

- **There will be 1 midterm (see syllabus)**
- **Final exam (last day of class, 11:00-13:20)**

Grades (Tentative Cutoffs)

- **A >= 90%**
- **B >= 80%**
- **C >= 70%**
- **D >= 60%**
- **F Below 60%**

Course Materials

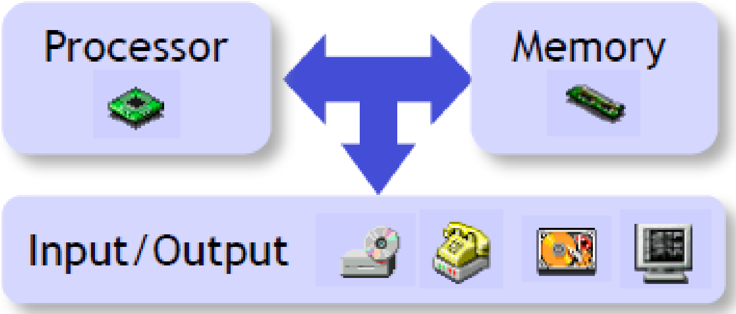
- **Textbook:** **http://bob.cs.sonoma.edu/IntroCompOrg-RPi/index-1.html**
- **Hardware: Raspberry Pi 3**
- **We will be using the GNU gcc and GNU as compilers**
- **Learn how to compress files into a zip archive. Each program that you write for this class will contain multiple source code files and you will need to send me the code as a zipped archive rather than as individual files. This is extremely easy to learn how to do**

Outline

- Syllabus discussion
- Homework & Lab submission policies
- Course Overview
- Computer Arithmetic
- Basic Data Type
- Raspberry Pi, Linux, text editor, compiler, makefile, gdb debugger, etc…

What is Computer Architecture About and Why Do We Care?

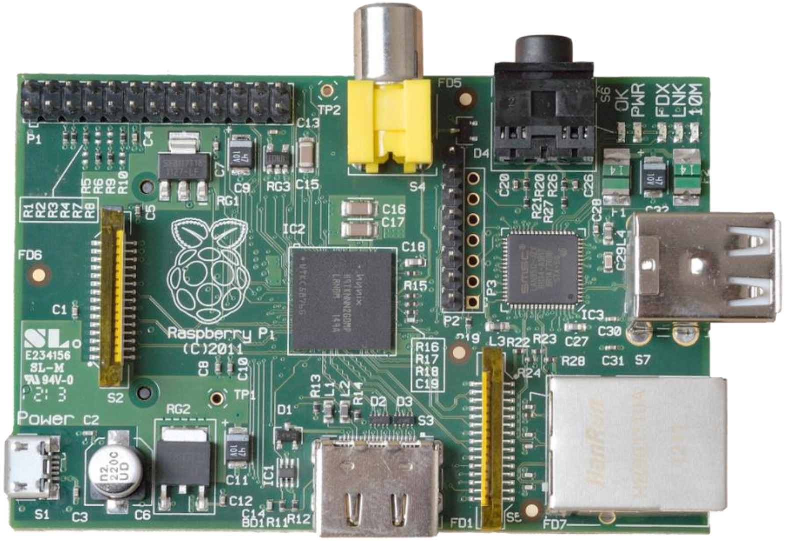
- Computer architecture is the study of building computer systems.



- **It's interesting???**
- **Helps you become a better programmer – understanding how your program is translated to assembly code lets you reason about correctness and performance and helps you demystify the seemingly arbitrary (e.g., bus errors, segmentation faults)**
- **Many interesting jobs require an understanding of computer architecture – the cutting edge is often pushing computers to their limits (e.g. supercomputing, games, portable devices, embedded processing, etc…)**

What is a Raspberry Pi and Why?

- University of Cambridge’s Computer Laboratory
  - Decline in skill level
  - Designed for education
- A credit card sized PC
- Plugs into a TV or monitor
- Inexpensive(ish) ~$35 each
- Capability:
  - Programming
  - Electronic Projects
  - Office
  - Play HD Videos



Programming Languages

- The Raspberry Pi Foundation recommends Python
- Any language which will compile for ARMv6 (Advanced RISC Machines, RISC – Reduced Instruction Set Computer) can be used
- Installed by default on the Raspberry Pi:
  - C
  - C++
  - Java
  - Scratch
  - Ruby

Raspberry Pi
Quick start

1 Insert SD card

See page 3 for how to prepare the SD card

2a Connect display

Plug in your digital TV or monitor

2b Connect display

If *not* using HDMI, plug in your analogue TV or display

3 Connect input

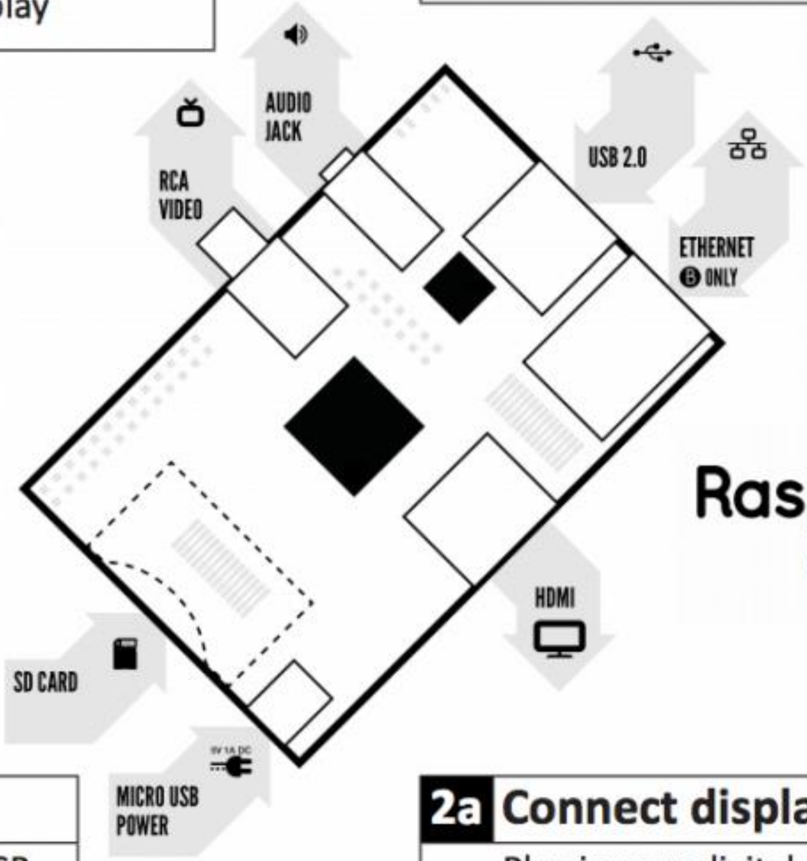
Plug in a USB keyboard and mouse

4 Connect network

Connect to your wired network [optional]

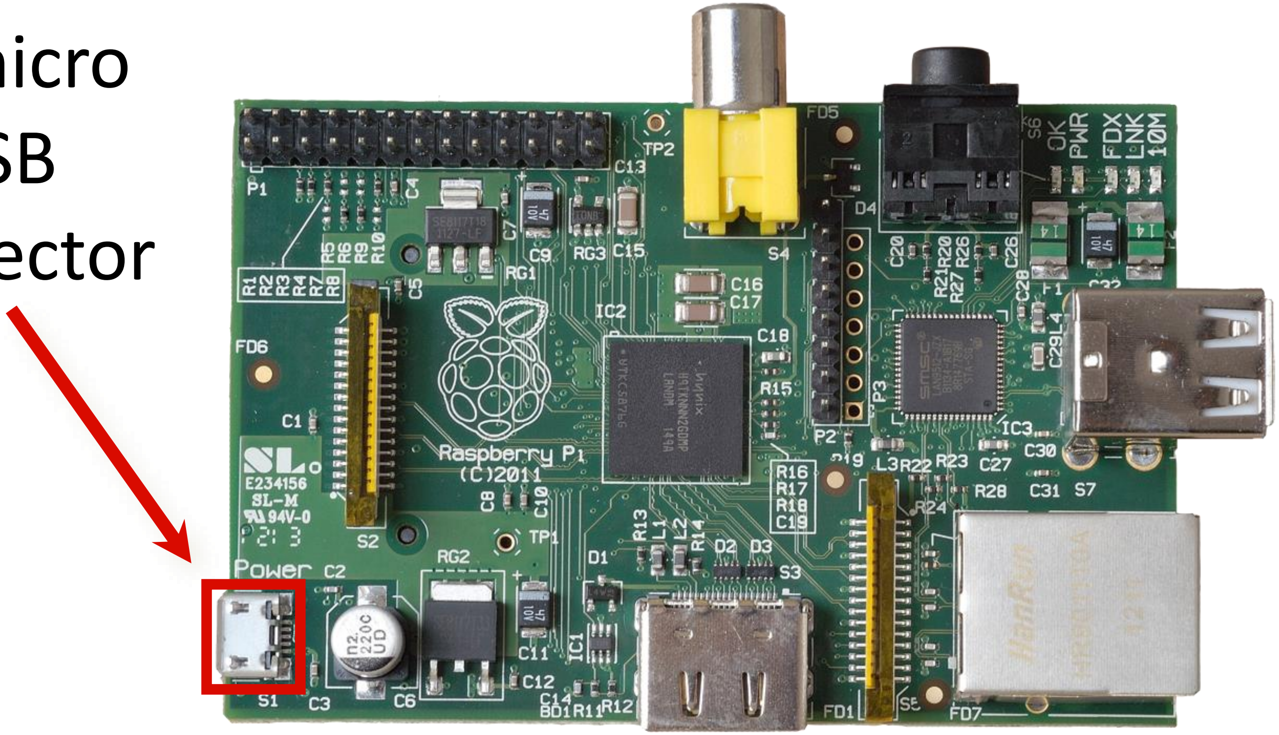
5 Power up

Plug in the micro USB power supply



Power

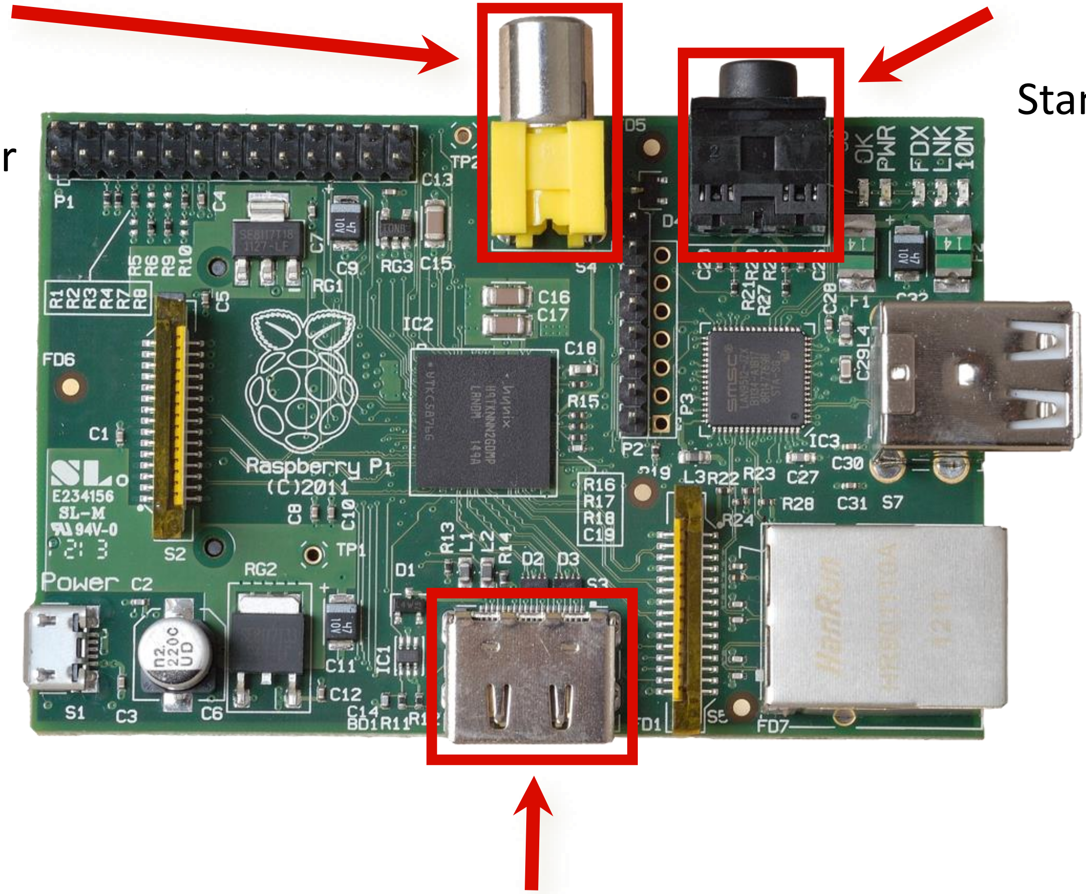
5v micro USB connector



(Similar to the one on a lot of mobile phones!)

A/V (Audio/Video)

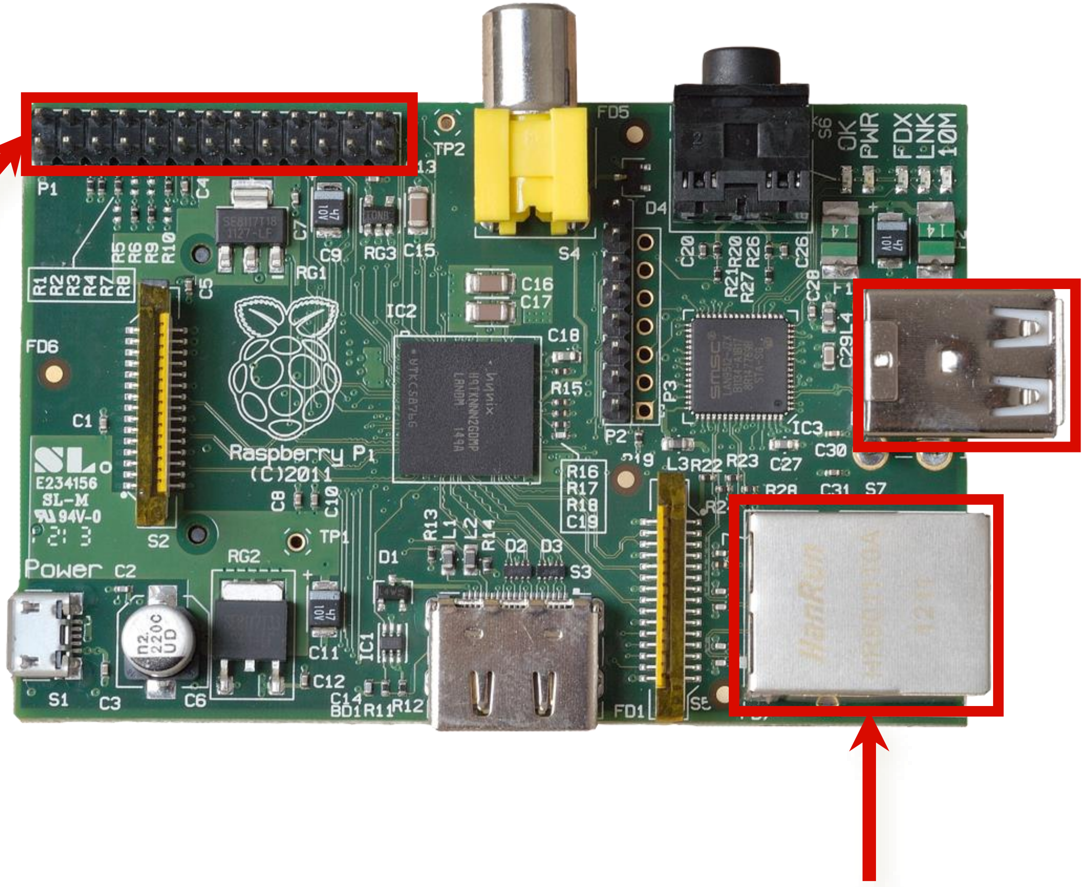
RCA Video
(works with most older TVs)



3.5mm Audio
Standard headphone socket

HDMI Audio & Video
(works with modern TVs and DVI monitors)

Connectivity

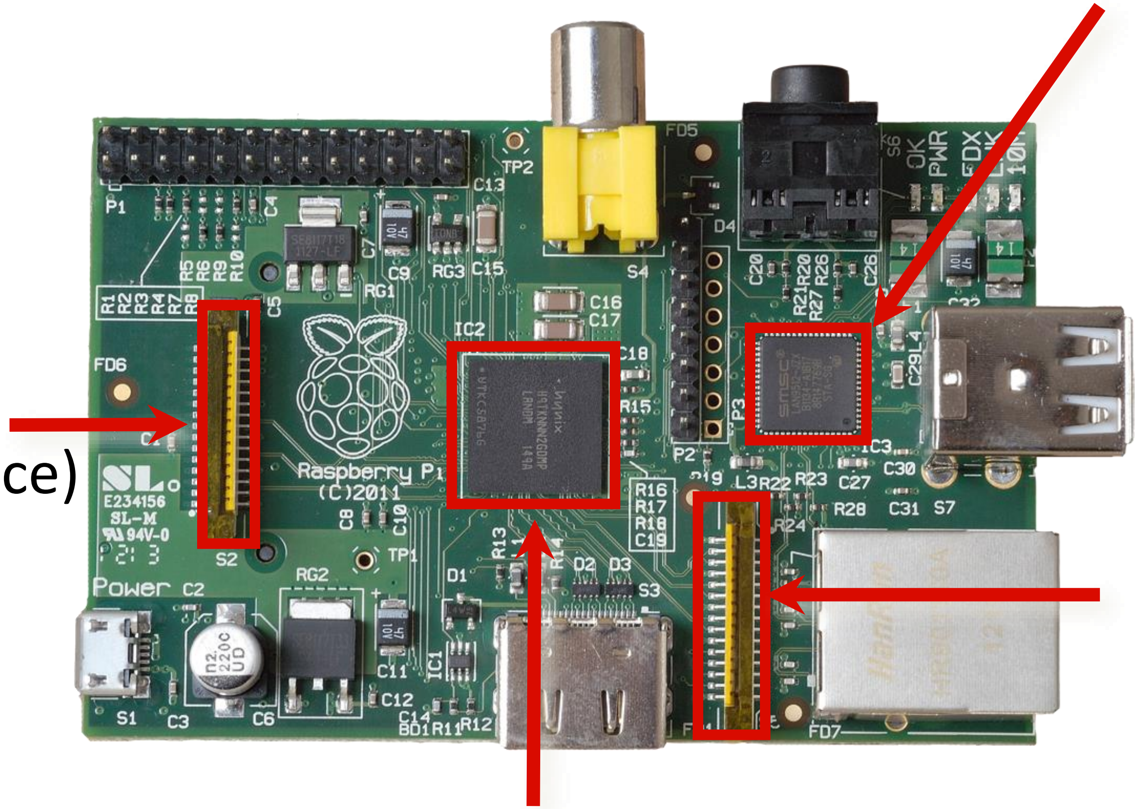


4 x USB 2.0 ports

GPIO (General Purpose Input & Output)

10/100Mb Ethernet

Internals



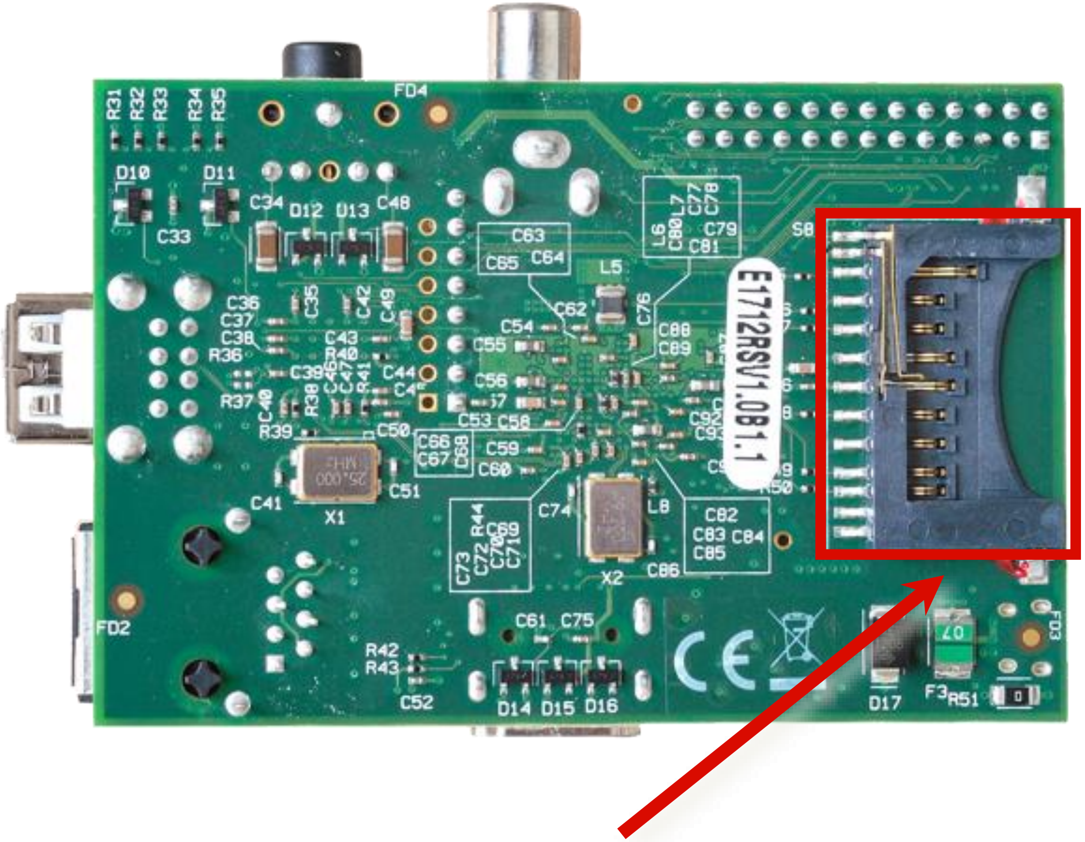
LAN Controller

DSI
(display interface)

CSI
(camera
interface)

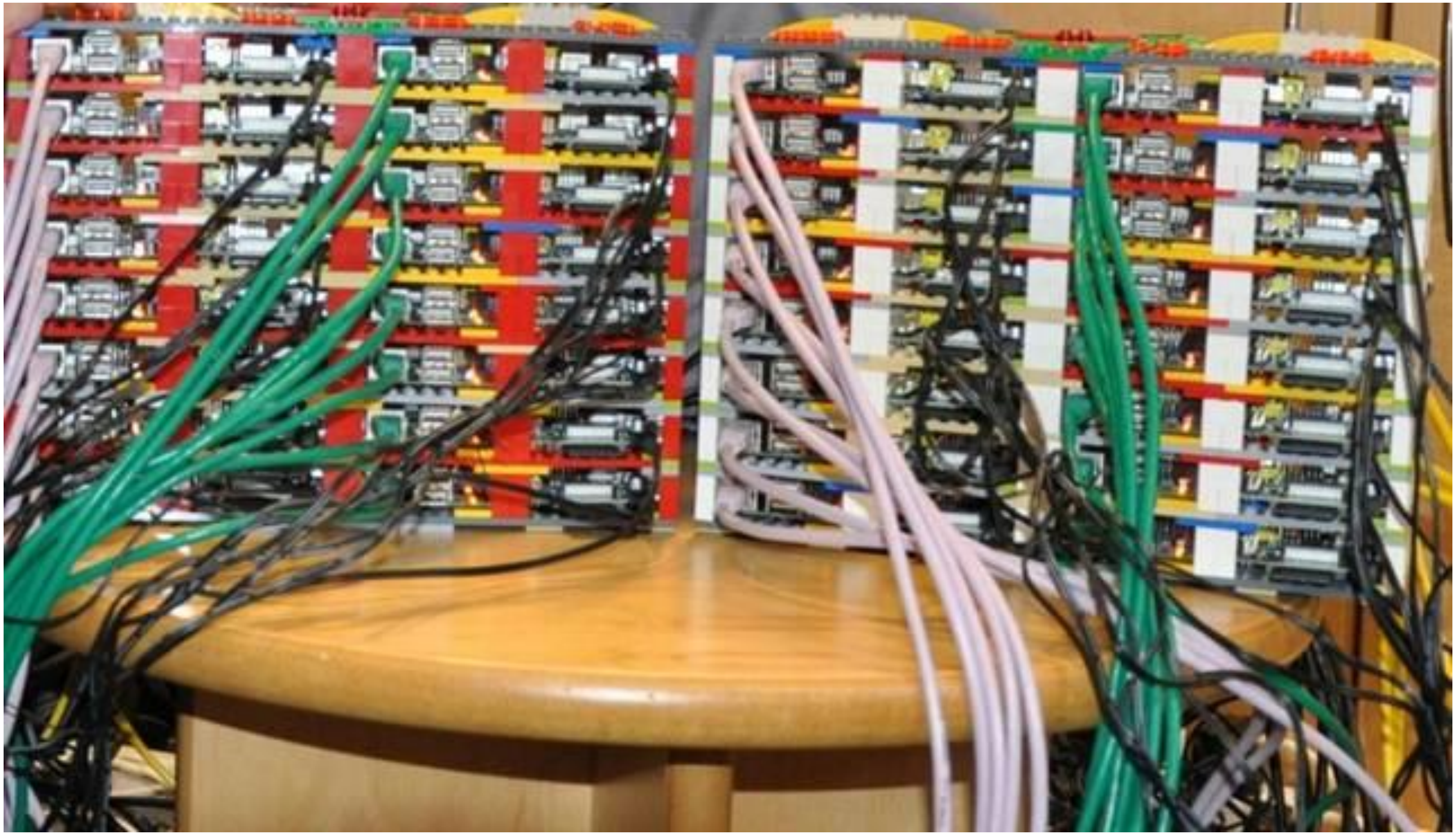
SOC (System On a Chip)
Broadcom BCM2835 700Mhz

Storage

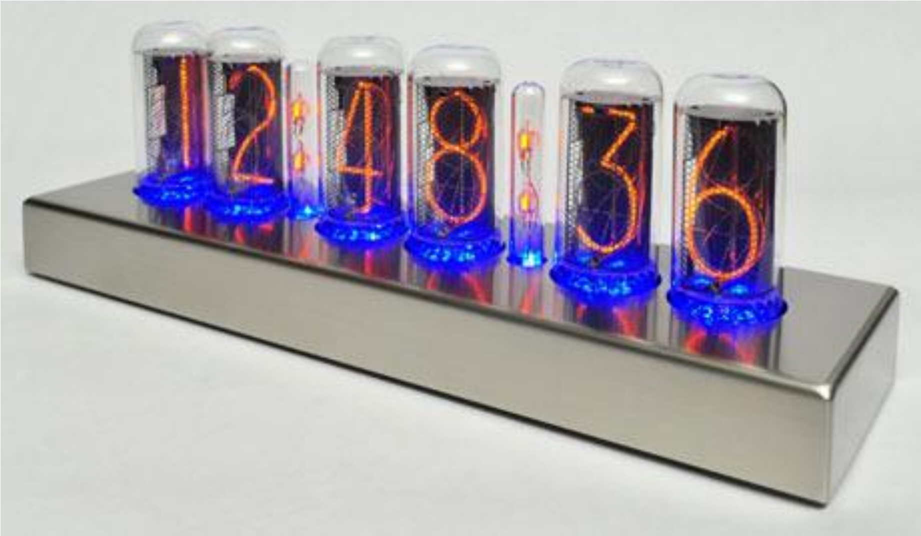


SD Card Slot
(supports SD cards up to 32GB)

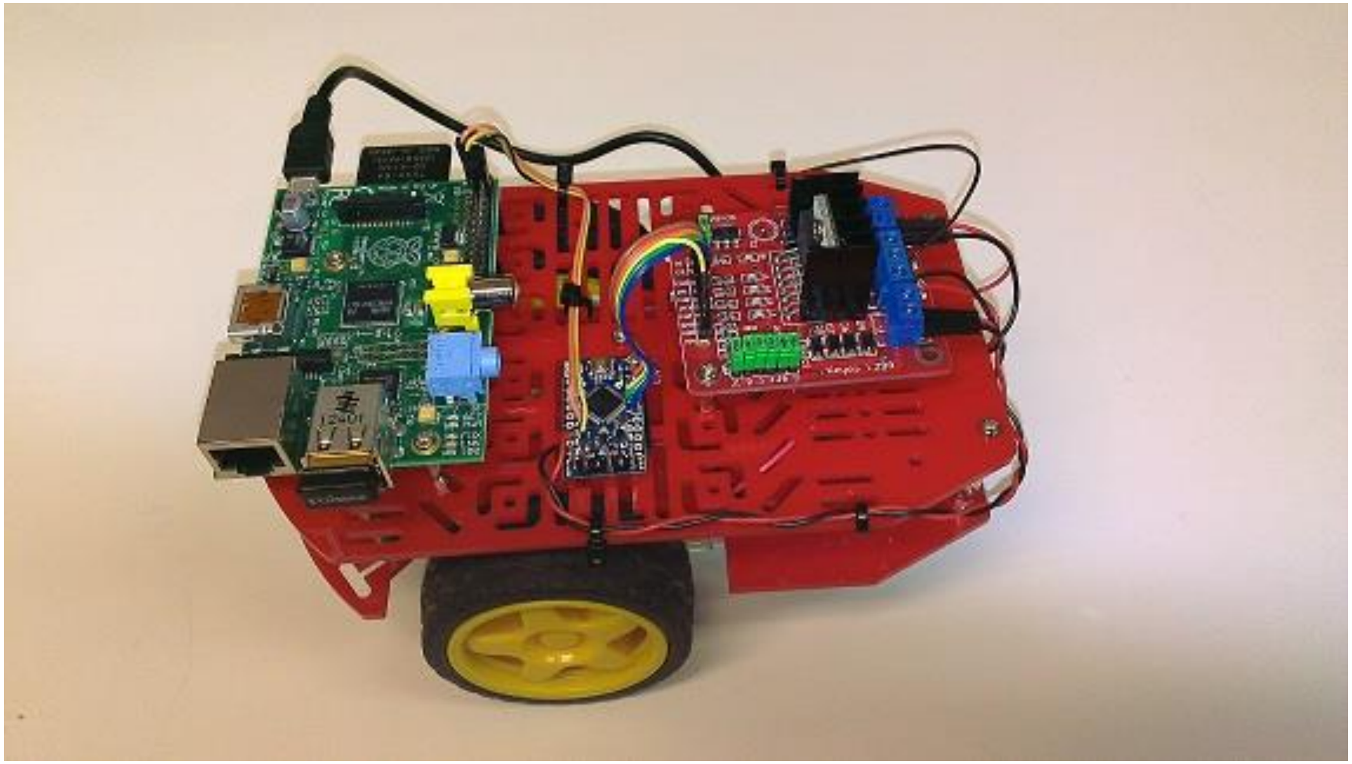
Supercomputer



Clock

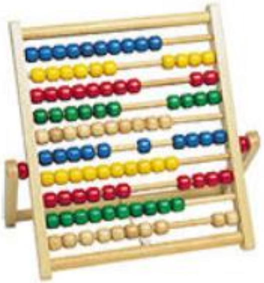


PiBot!

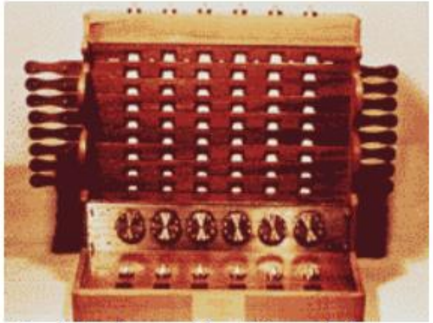


So Why The Raspberry Pi?
Practice programming for an ARM based embedded platform

The Evolution of Computing



Pascaline



Schickard’s Machine

2400 BC

17th Century

Big Idea Behind Early “Computers”

Fixed Program Model

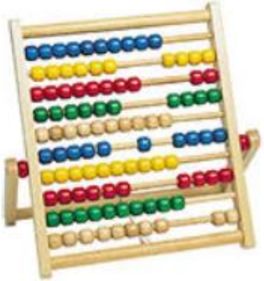
Specific (computation) Problem

↓

Circuit to solve it

- The ‘program’ was wired into the computing device

The Evolution of Computing



2400 BC

Pascaline

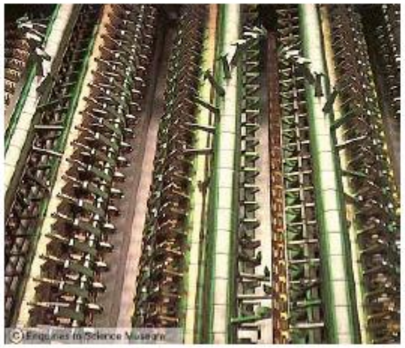
Schickard's Machine

17th Century

Automated textile looms

Jacquard's Loom

1804



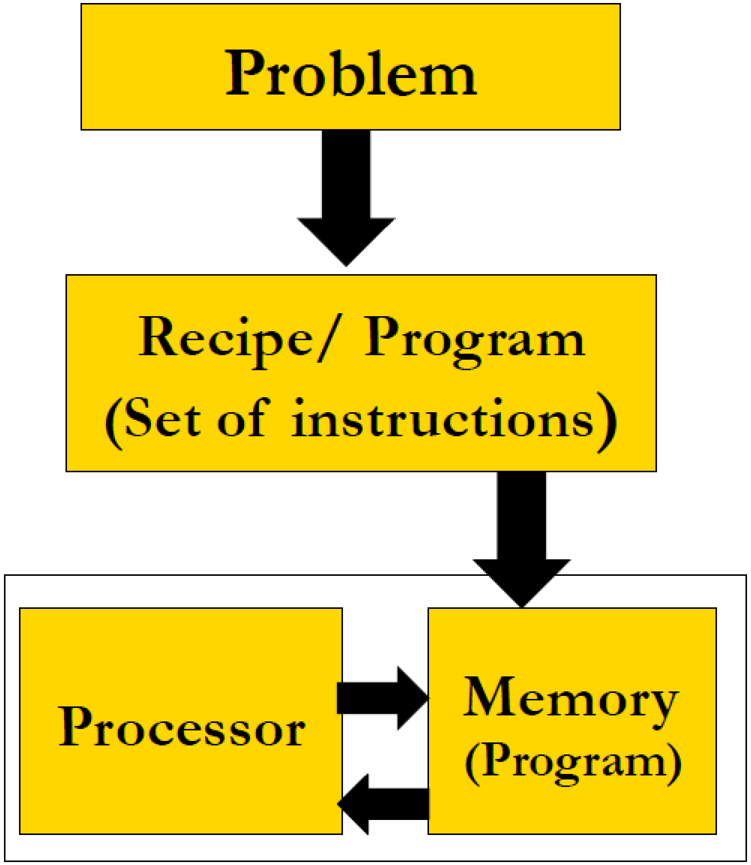
Analytical Engine

1822

Next Big Idea – The Stored Program Model

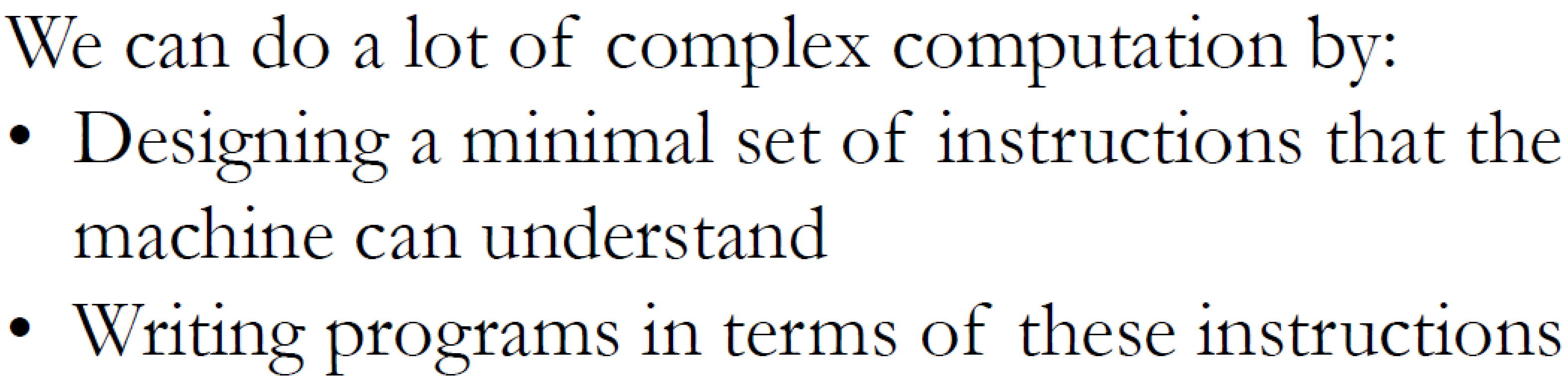
- Key Ideas:
  - Computer divided into two components: Processor and Memory
  - Program and data stored in the same place: memory

- Consequences
  - Programs easily fed into the computer
  - Avoid clumsy methods of programming



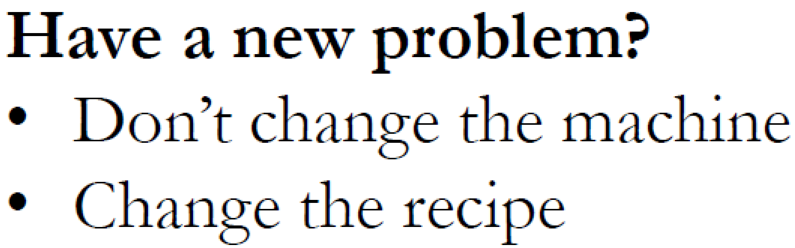
Stored Program Model
proposed by Jon Von Neumann

Stored Program – Big Idea!



We can do a lot of complex computation by:

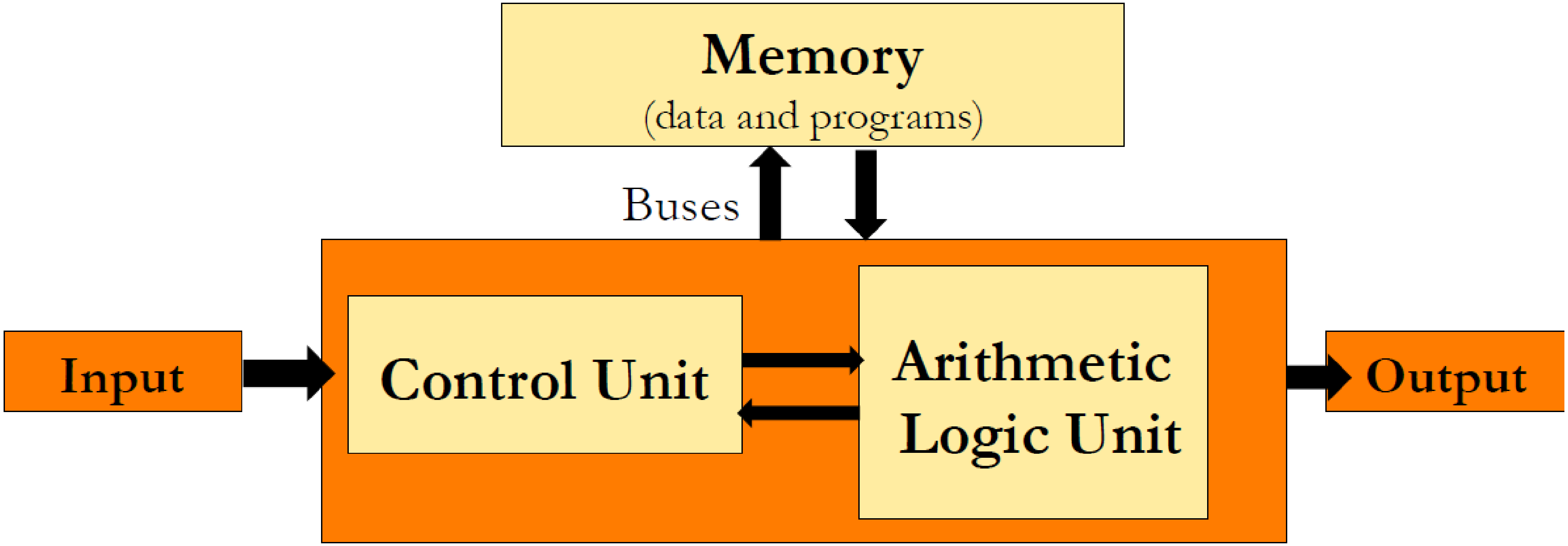
- Designing a minimal set of instructions that the machine can understand
- Writing programs in terms of these instructions



Have a new problem?

- Don't change the machine
- Change the recipe

Von Neuman Architecture



4 Basic Components of a Computer:

1. **Memory**: a long but finite sequence of cells (1D)
    - Each cell has a distinct address
    - Data in each cell: instruction, data or the address of another cell
2. **Control Unit**: Fetches instructions from memory and decodes them
3. **Arithmetic Logic Unit**: Does simple math operations on data
4. **Input/Output:** The connections with the outside world

In Lab Exercises

- **Practice writing, compiling, and executing codes on the Pi**

Reading Assignment

- Read chapters 1-3 in the Plantz online textbook