COMP 273
Introduction to Computer Systems

COMP 273

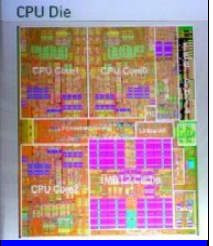
Introduction to Computer Systems

Prof. Joseph Vybihal

Warning

This course may remove all fantasies you have about computers...

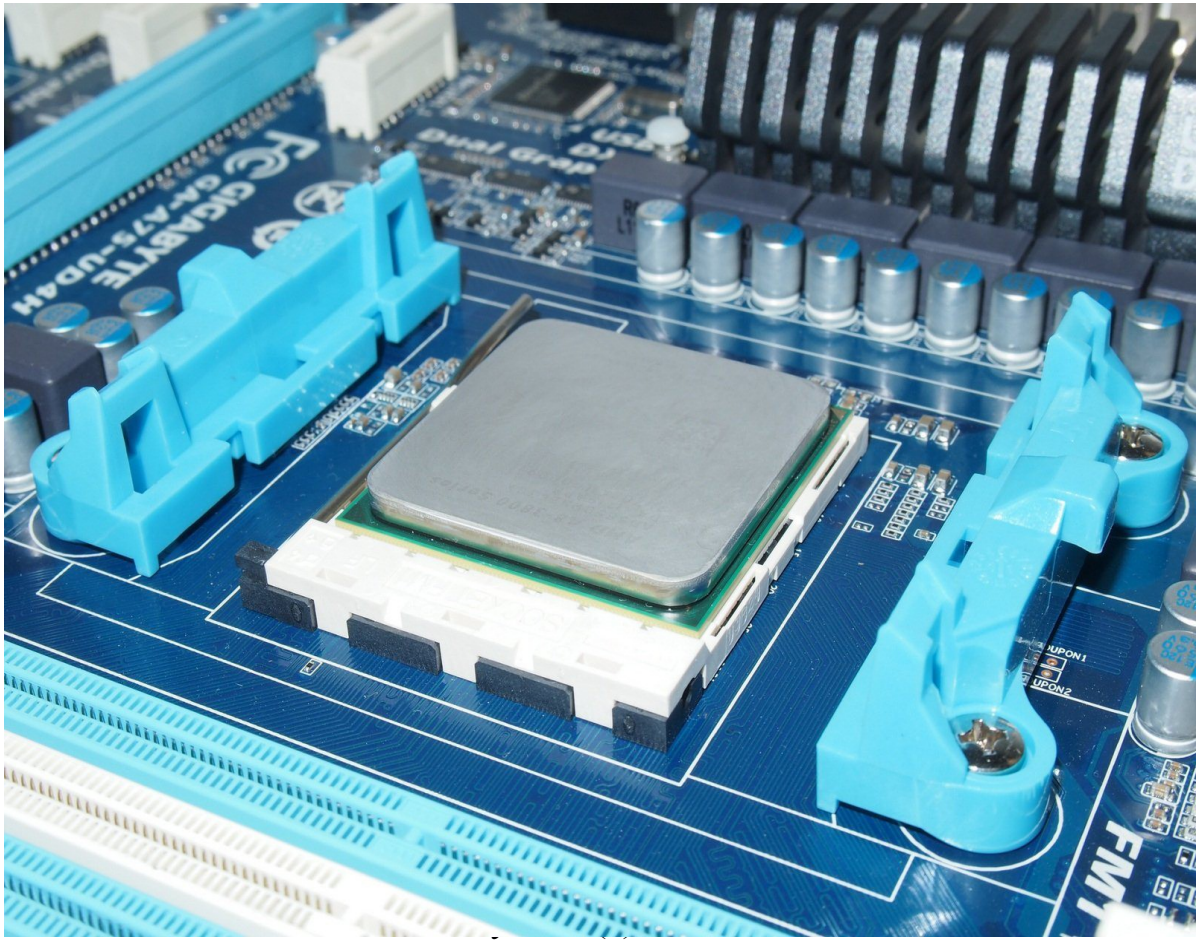
... you may end up loving it!

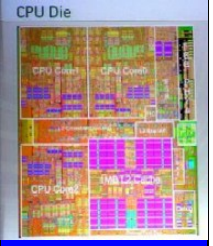


CPU

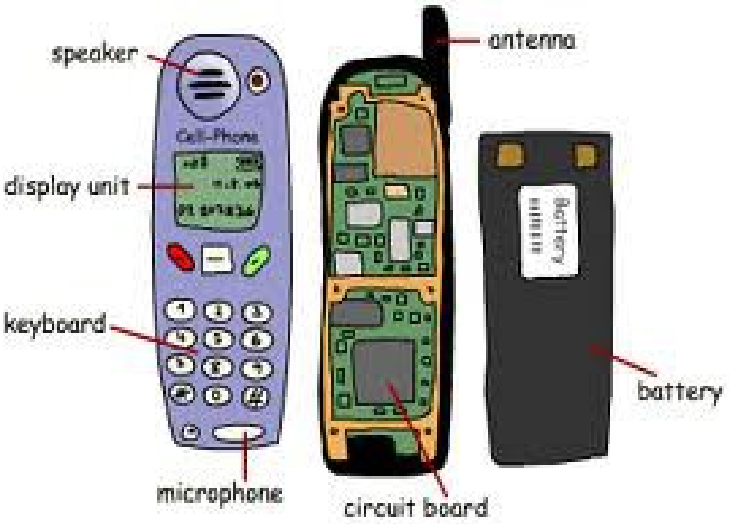
It exists almost everywhere.

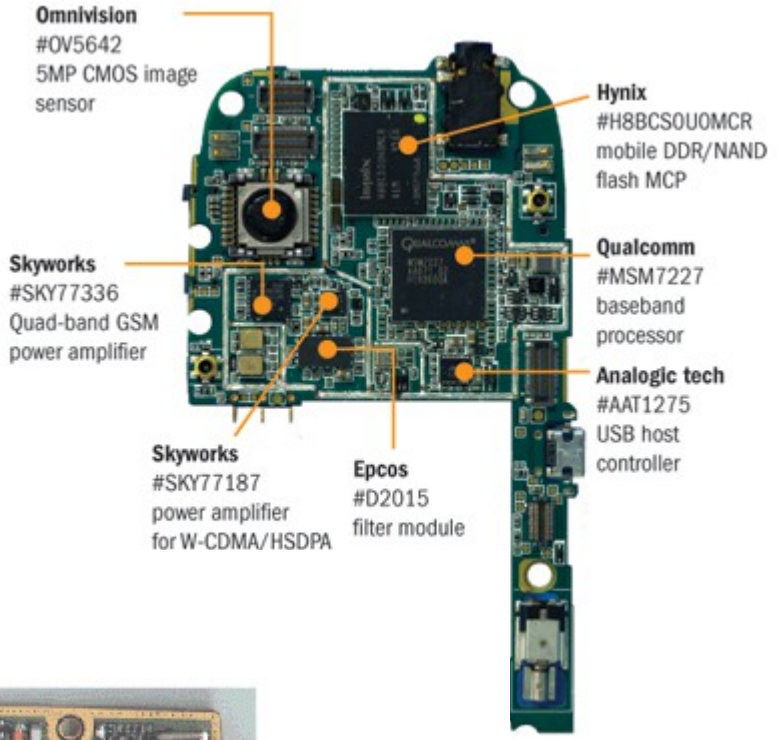
Name some places....

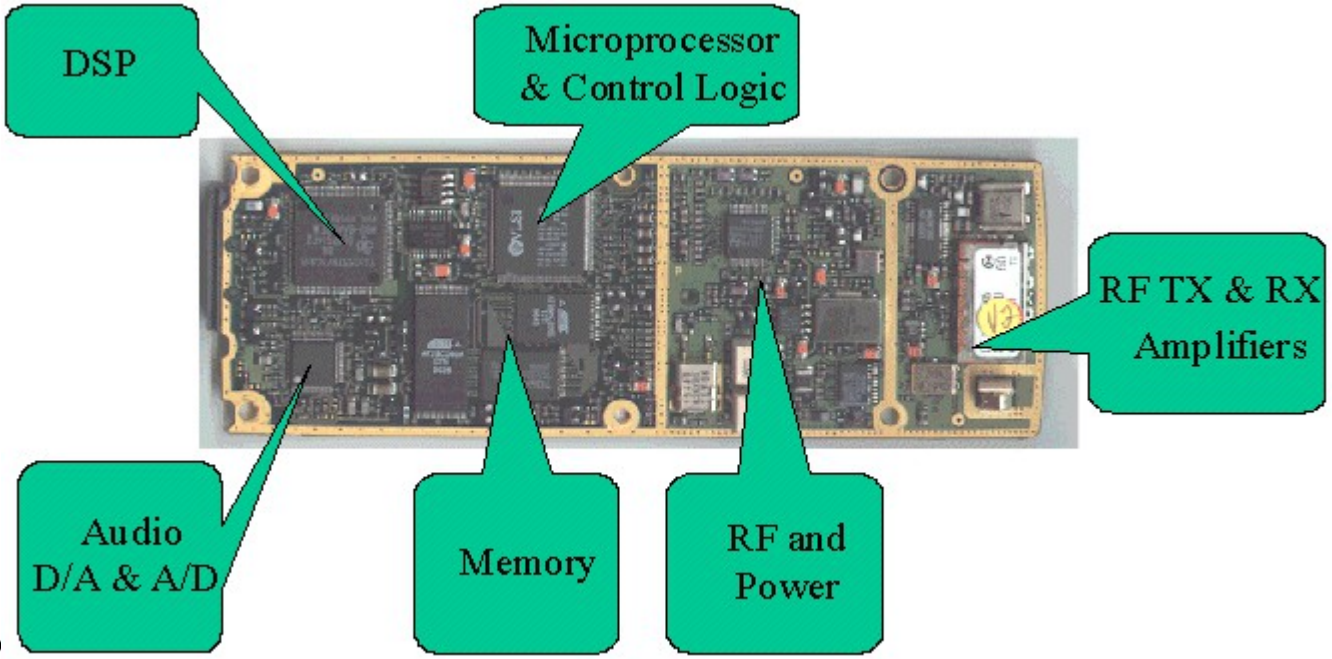



CPU Die

Cell Phones

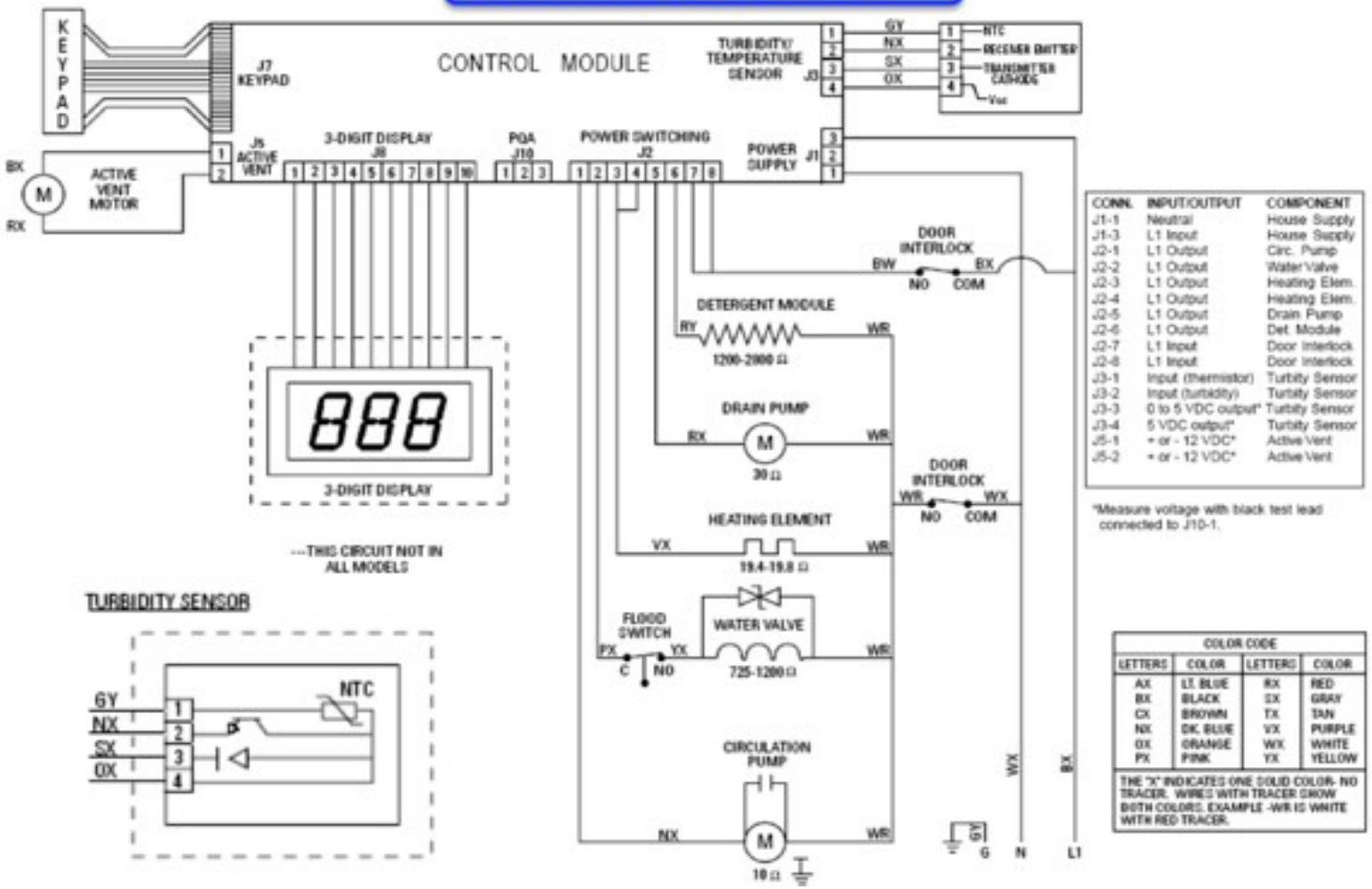

speaker
antenna
display unit
keyboard
battery
microphone
circuit board


Omnivision
#OV5642
5MP CMOS image sensor
Hynix
#H8BCS0UOMCR
mobile DDR/NAND flash MCP
Skyworks
#SKY77336
Quad-band GSM power amplifier
Qualcomm
#MSM7227
baseband processor
Analogic tech
#AAT1275
USB host controller
Skyworks
#SKY77187
power amplifier for W-CDMA/HSDPA
Epcos
#D2015
filter module


DSP
Microprocessor & Control Logic
RF TX & RX Amplifiers
Audio D/A & A/D
Memory
RF and Power

COMP 273
Introduction to Computer Systems

Dishwasher



CONN.	INPUT/OUTPUT	COMPONENT
J1-1	Neutral	House Supply
J1-3	L1 Input	House Supply
J2-1	L1 Output	Circ. Pump
J2-2	L1 Output	Water Valve
J2-3	L1 Output	Heating Elem.
J2-4	L1 Output	Heating Elem.
J2-5	L1 Output	Drain Pump
J2-6	L1 Output	Det. Module
J2-7	L1 Input	Door Interlock
J2-8	L1 Input	Door Interlock
J3-1	Input (thermistor)	Turbity Sensor
J3-2	Input (turbidity)	Turbity Sensor
J3-3	0 to 5 VDC output*	Turbity Sensor
J3-4	5 VDC output*	Turbity Sensor
J5-1	+ or - 12 VDC*	Active Vent
J5-2	+ or - 12 VDC*	Active Vent

*Measure voltage with black test lead connected to J10-1.

COLOR CODE			
LETTERS	COLOR	LETTERS	COLOR
AX	LT. BLUE	RX	RED
BX	BLACK	SX	GRAY
CX	BROWN	TX	TAN
NX	DK. BLUE	VX	PURPLE
OX	ORANGE	WX	WHITE
PX	PINK	YX	YELLOW

THE "X" INDICATES ONE SOLID COLOR- NO TRACER. WIRES WITH TRACER SHOW BOTH COLORS. EXAMPLE -WR IS WHITE WITH RED TRACER.

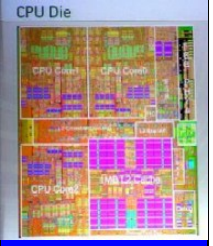
A house of cards

- Simple units
  - Build wonderful things
- Engineered
  - One layer built above a previous layer
- So simple, but not.

COMP 273
Introduction to Computer Systems

This course is about understanding...

- The Computer
  - The internal operation of the computer
- Programming Languages
  - What an instruction really is and how it works
- The capabilities of a programmer
  - How to control machines
  - Creating drivers
  - Communicating with peripherals



COMP 273
Introduction to Computer Systems

How it fits in

Programming, C, Unix

273

310 Operating Systems

308 Graphics Cards

535 Networks

520 Compiler Design

417 Robotics

How does this course fit in?

Programming

COMP273

Assembler, Circuits

OS | Networks, Compilers | Robots Drivers | Devices | Computers

Systems Programming

X-Domain

Circuit Building

Computer Science

Engineering

Instructor

- Prof
  - Joseph Vybihal, ENGMC 323
    - www.cs.mcgill.cs/~jvybihal
  - Hours:
    - M 13:30 & W 9:00 or by appointment
  - Discussion boards via MyCourses
  - Email: jvybihal@cs.mcgill.ca
- Class participation

Course Outline

- Part 1
  - About circuits
  - Building circuits
  - Internal CPU construction
    - Classical
    - Pipeline
    - Multi Core
- Part 2
  - Assembler programming
  - Low-level peripheral interfacing
- Part 3
  - System hardware support
    - Circuits that support the OS

Project

- Team project
- Build your own CPU
- Create your own programming language
- Write a program in your language
- Run it on your CPU !!

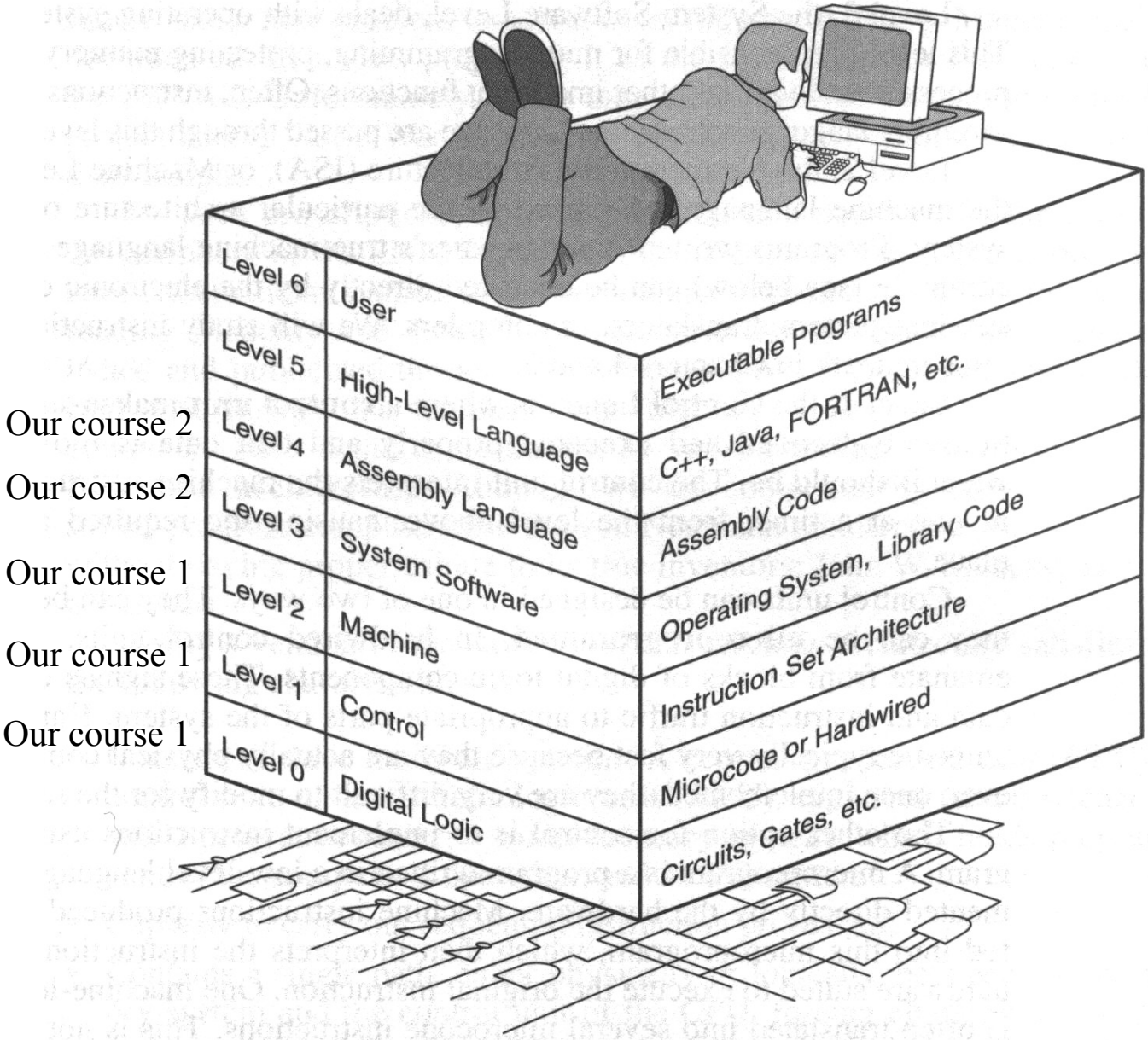
Course Syllabus

- Organization:
  - 3 assignments → 10% of grade
  - CPU Project → 20% of grade
  - Midterm → 20% of grade (TBD)
  - Final exam → 50%
- Co-requisites: COMP 206
- Textbooks:
  - The Soul Of A New Machine (optional, bonus pts)
  - Computer Organization and Design

My Courses

- Course outline
- Lectures
- Assignment pick-up and drop-off
- All grades displayed there
- Discussion boards
  - Organized by topic
  - All welcome to ask and answer questions
    - No code! (pseudo-code okay)

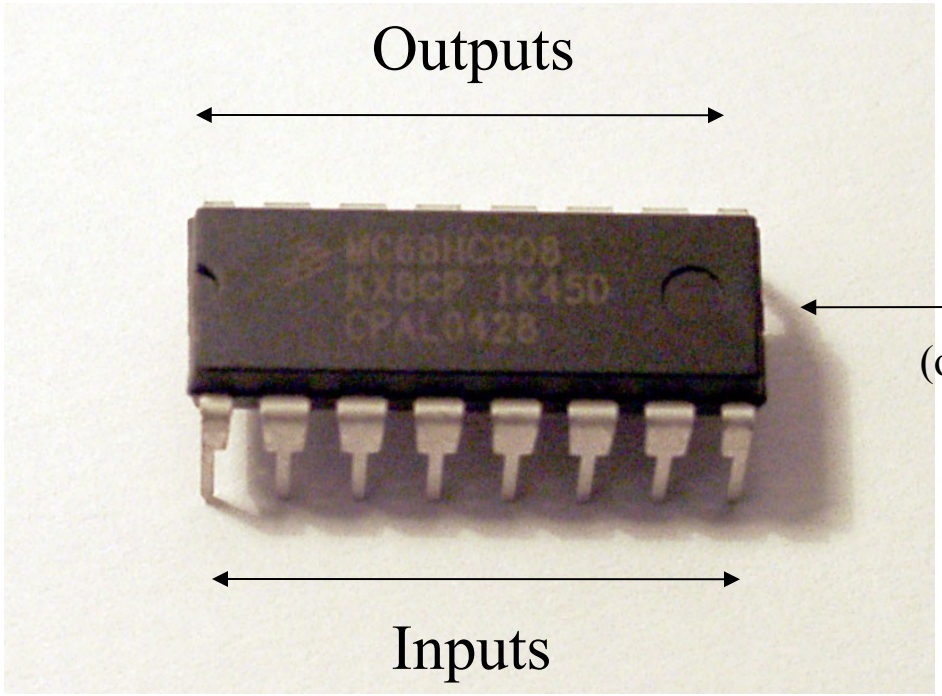
COMP 273
Introduction to Computer Systems



Our course 2
Our course 2
Our course 1
Our course 1
Our course 1

COMP 361
COMP 202/206
COMP 206/310

What is this course about?



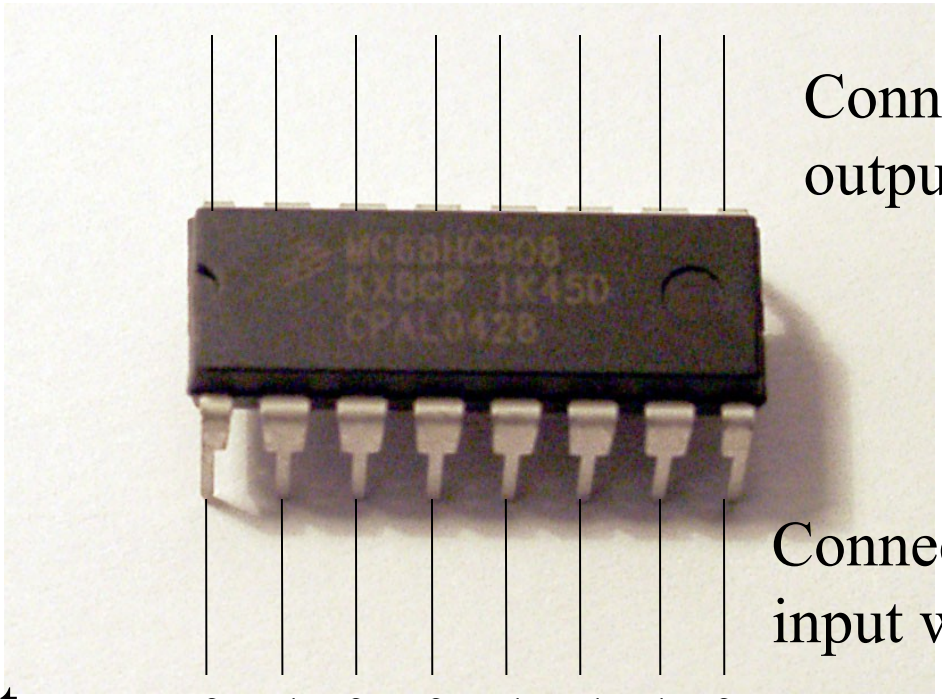
Note: 8 inputs and 8 outputs…. A byte!

What is this course about?

Controls ⟷ Solution

Overflow
Negative

0 0 0 0 0 1 1 1



Connected to output wires

Connected to input wires

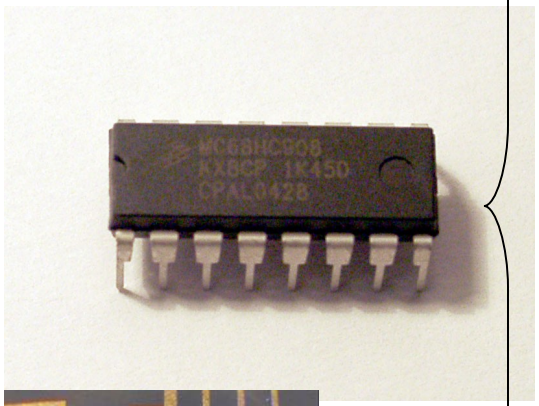
Hi or Low volt signal to chip ⟶

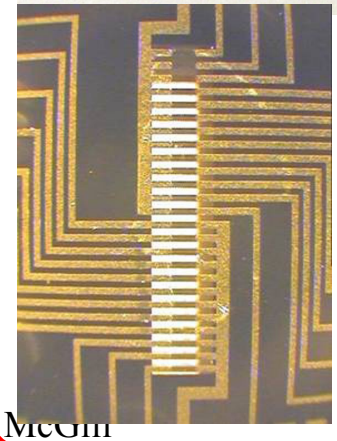
0 1 0 0 1 1 1 0

Input is an instruction:
01 is add
001 is number 1
110 is number 6

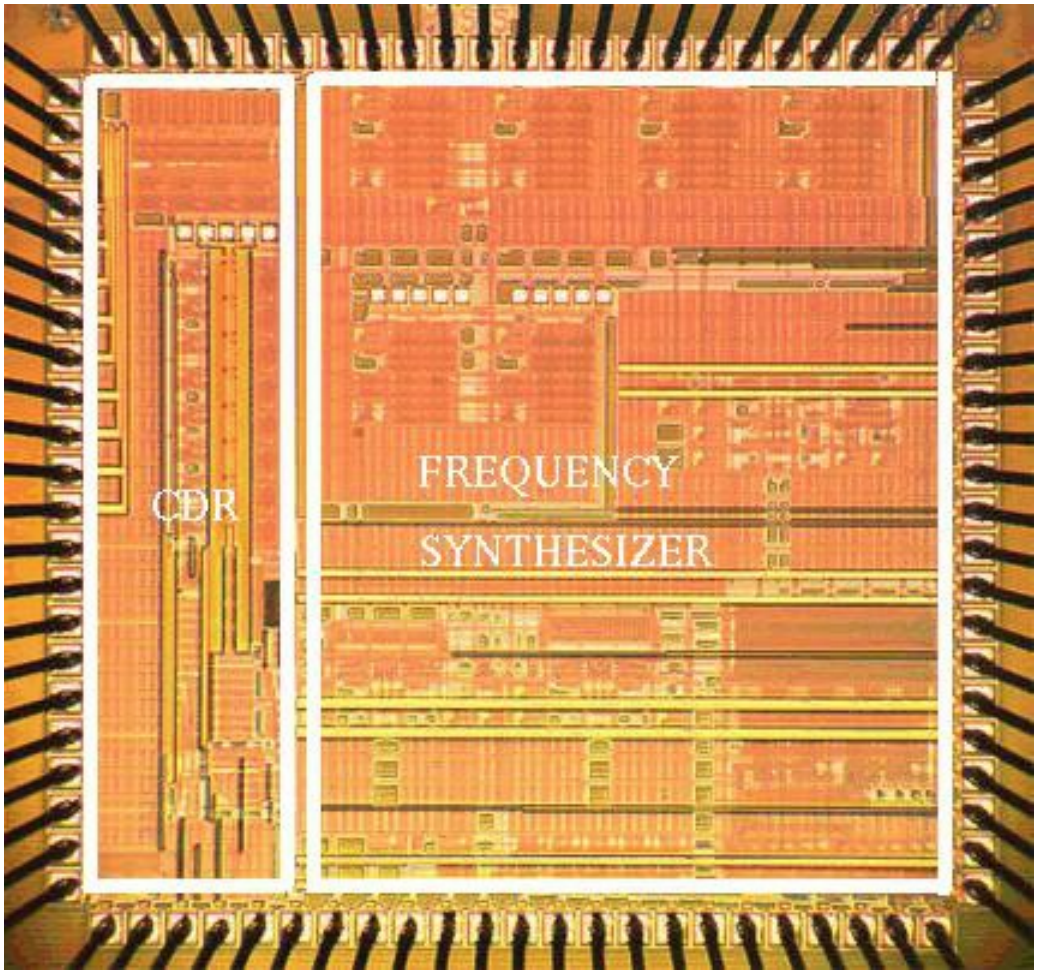
What is this course about?

Outputs





Golden Wires



Inputs

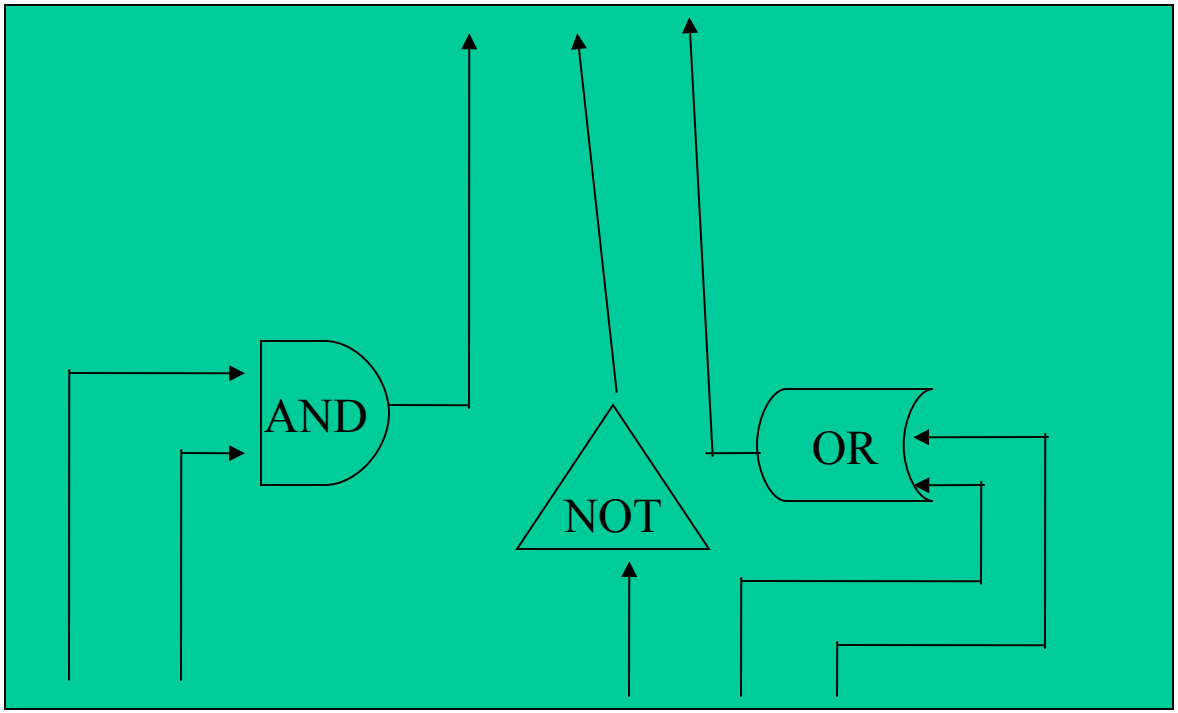
COMP 273
Introduction to Computer Systems

What is this course about?

Output wires

A Chip

Circuits & Gates



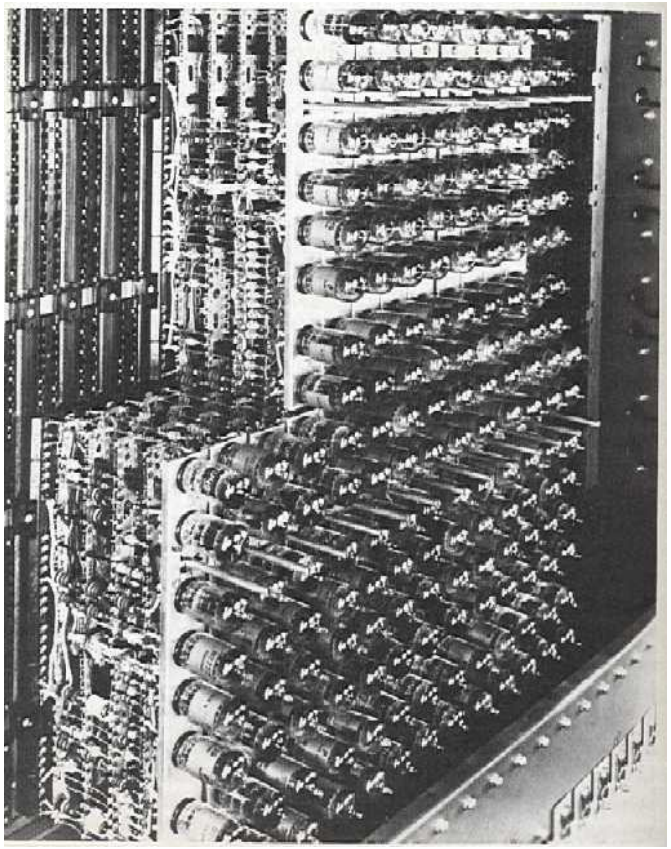
Input wires

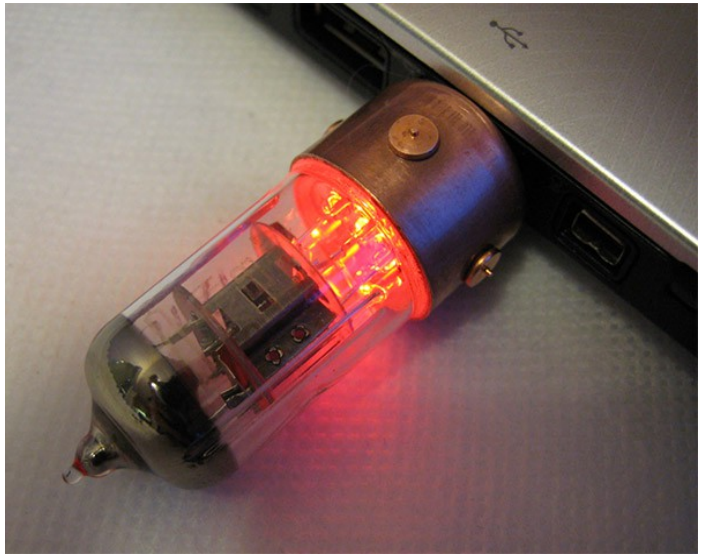
AND	OR	NOT
1 and 1 = 1	1 or 1 = 1	1 not = 0
1 and 0 = 0	1 or 0 = 1	0 not = 1
0 and 1 = 0	0 or 1 = 1	
0 and 0 = 0	0 or 0 = 0	

Boolean expressions

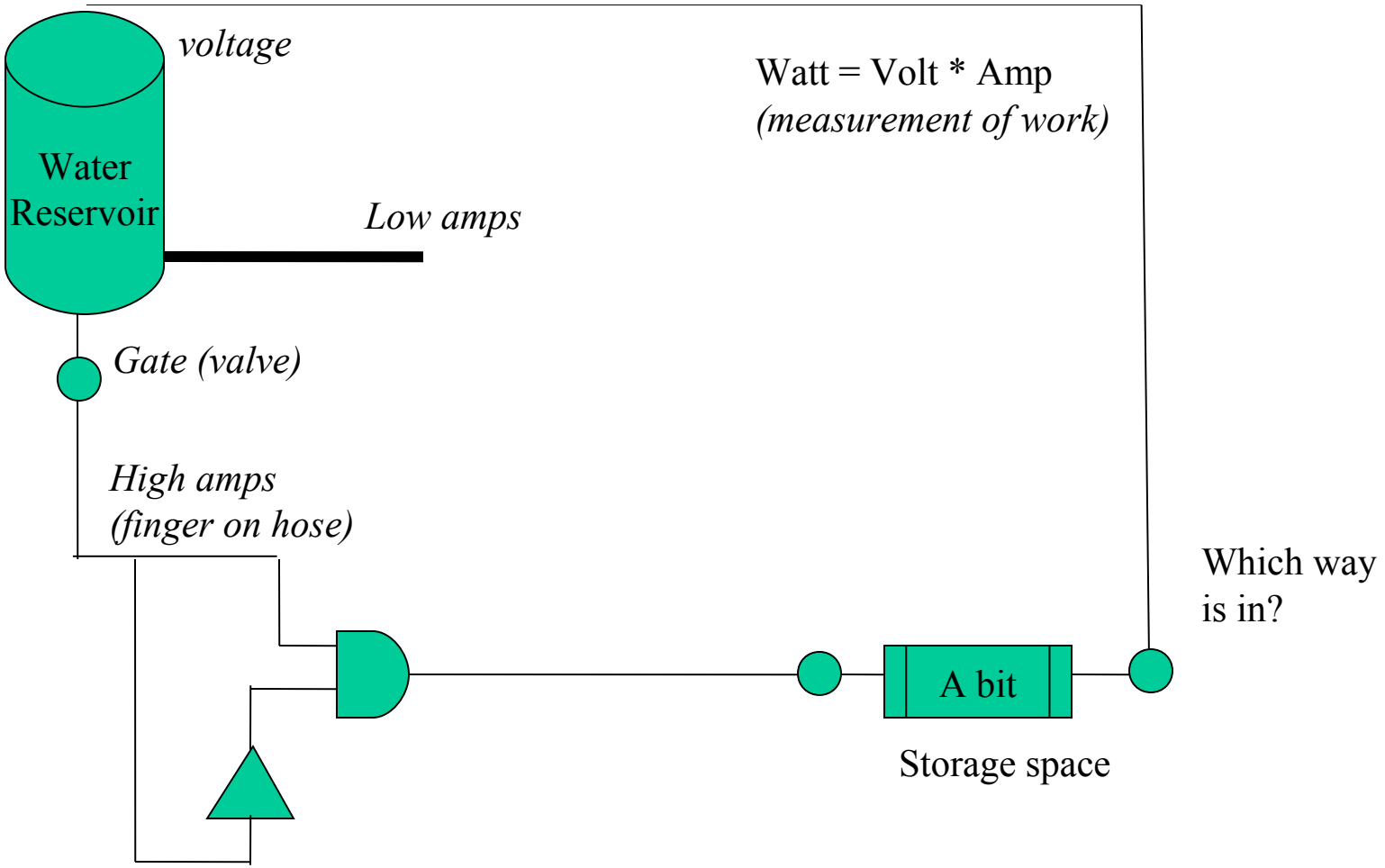
Electricity and Gates

- The meaning of "Live" data
  - Based on fickle electricity

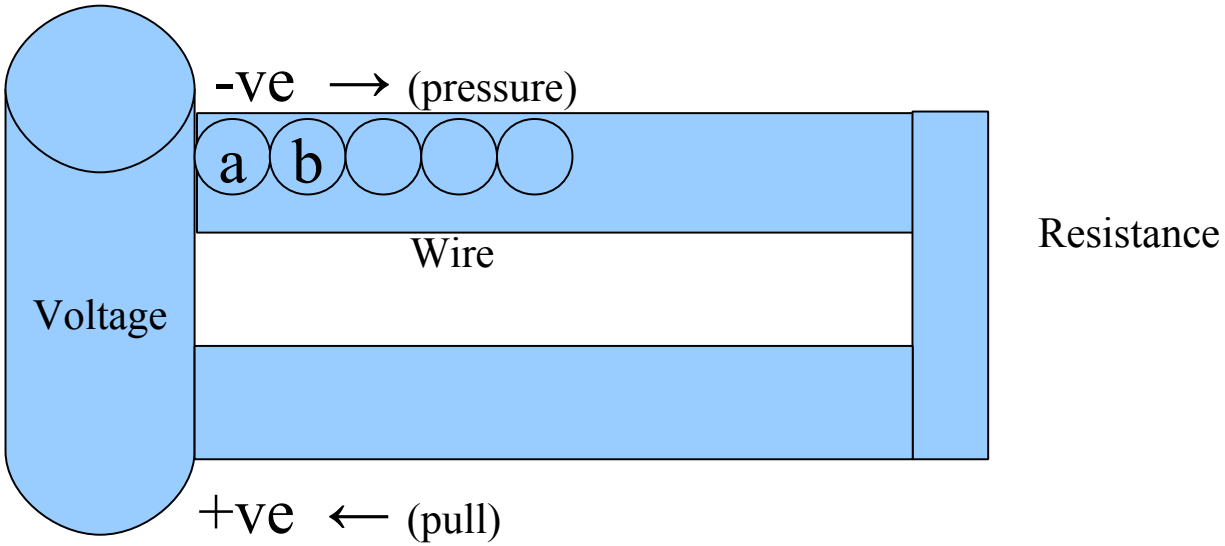




The Water Flow Analogy

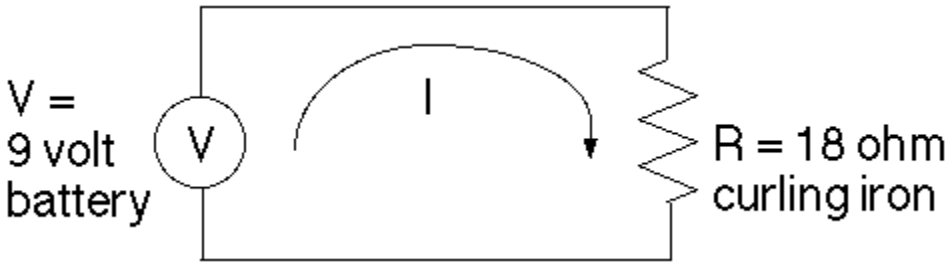


The Balls in a Pipe Analogy



- Each atom is beside each other like a chain of balls.
- Extra electrons jump out of the voltage source to the adjacent atom 'a'.
- Atom 'a' has an extra electron that jumps to atom 'b', etc until +ve terminal.
- Wires are “conductors”, they let atoms hop from atom to atom easily.
- Resistors are “semiconductors” that have a harder time with electrons hopping from one atom to the next. Since this is a chain, then the entire circuit moves at the speed of the slowest resistor.
- Generally speaking: everything causes resistance, even wires.

Ohm's Law: V = I * R



A nine volt battery supplies power to a curling iron with a resistance of 18 ohms. How much current is flowing through the curling iron?

V = I R

I = V / R

I = 9 / 18

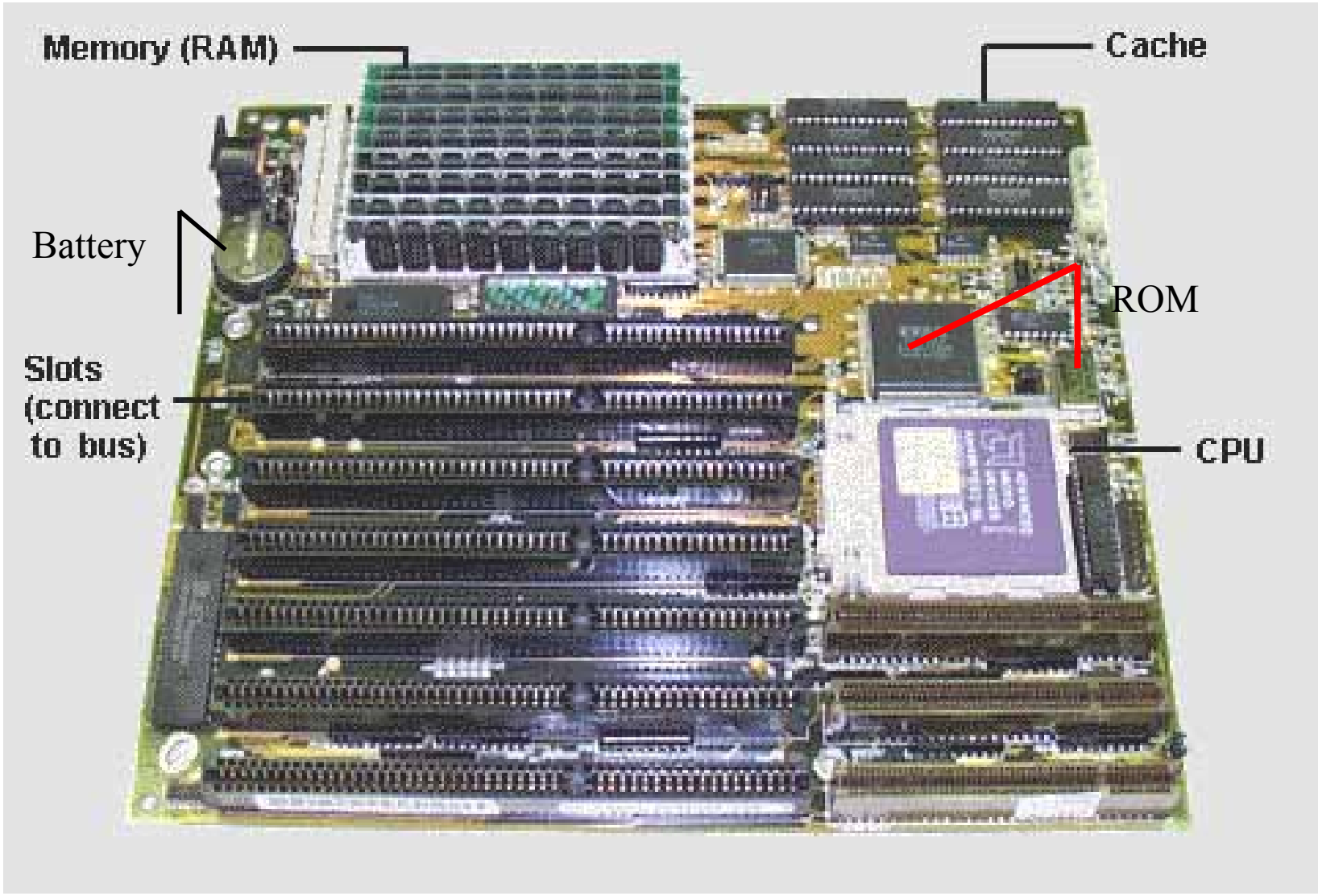
I = 0.5 Amps

How many volts are needed to run a CD player that uses 0.1 Amps of current and has a resistance of 1000 ohm?

The Architecture of the ~~Motherboard~~ System Board

Old style (big)

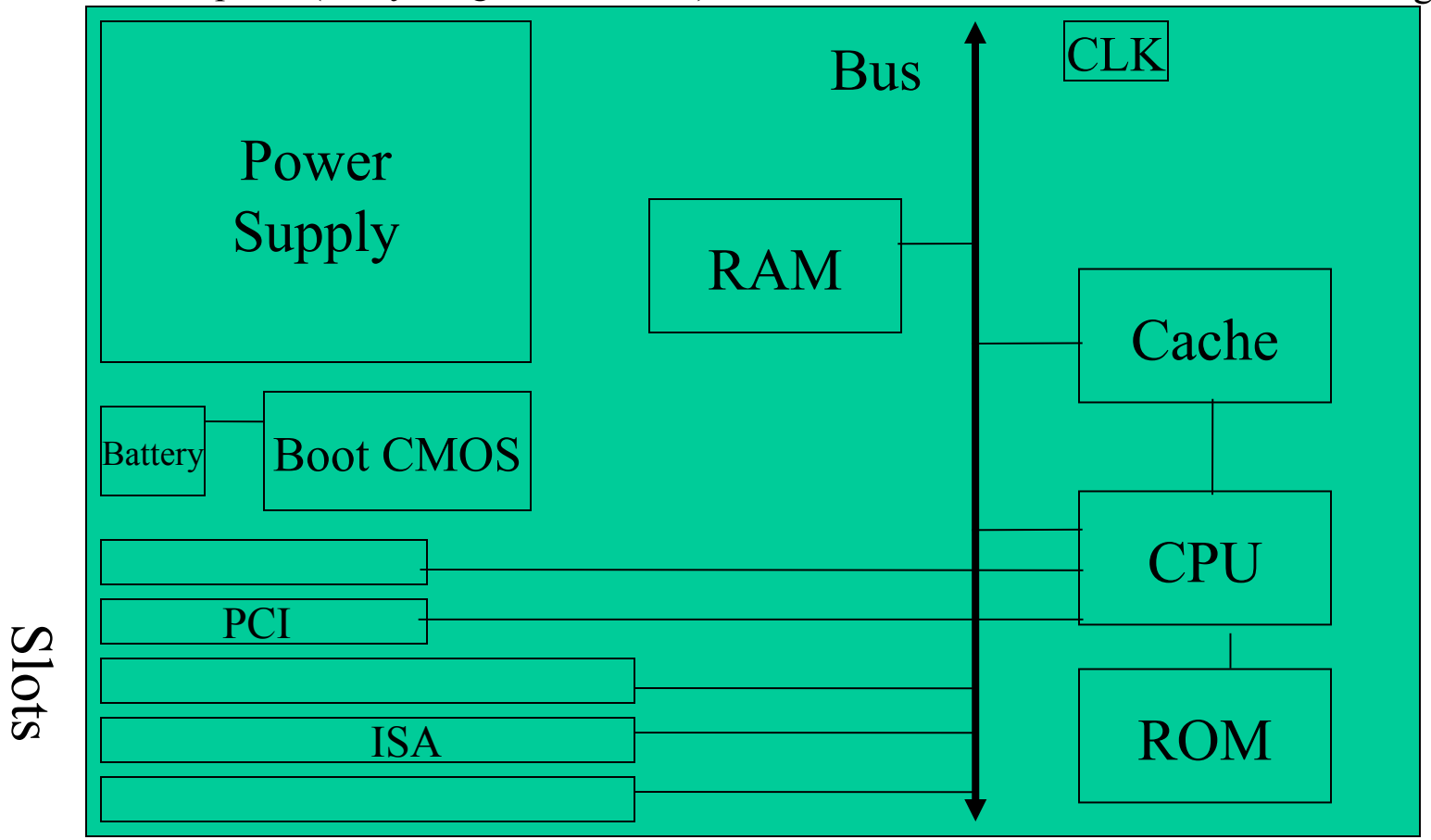
The Motherboard/System Board



System Board Schematic

Data paths (everything is connected)

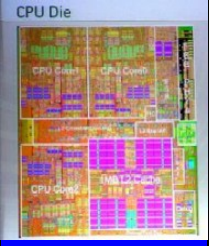
British Galleon Drum Analogy



How do all the parts work together? Its all about the Bus...

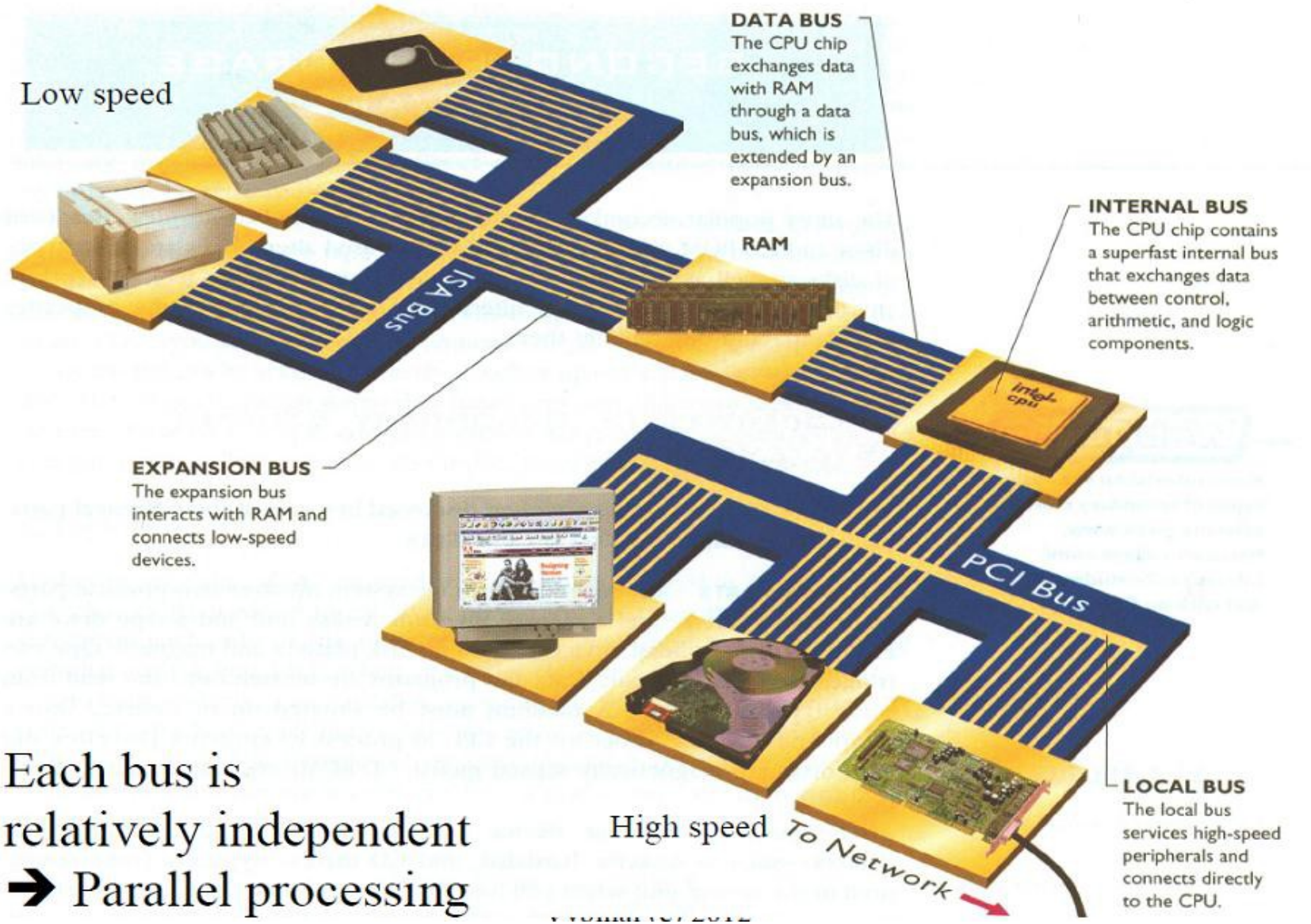
System Board Parts

- Power Supply: Converts the AC/DC from the home into the steady current needed in the PC.
- CPU: Central Processing Unit (The Brain) – Math, Logic, Data movement, loops.
- ROM: Read Only Memory – Used to store built-in instructions (like the CMOS), additional instructions for the CPU.
- Battery: Used to help keep the CMOS parameters (including the time).
- RAM: Random Access Memory – Volatile main memory bank, large and slow.
- Cache: A very fast type of memory (pipeline) directly connected to RAM.
- Bus: A common road for data that interconnects all devices on the motherboard
- CLK: Clock – Beats the processing cycle (2 of them)
- Slot – Connects to devices external to the motherboard through cards



The 4 Buses

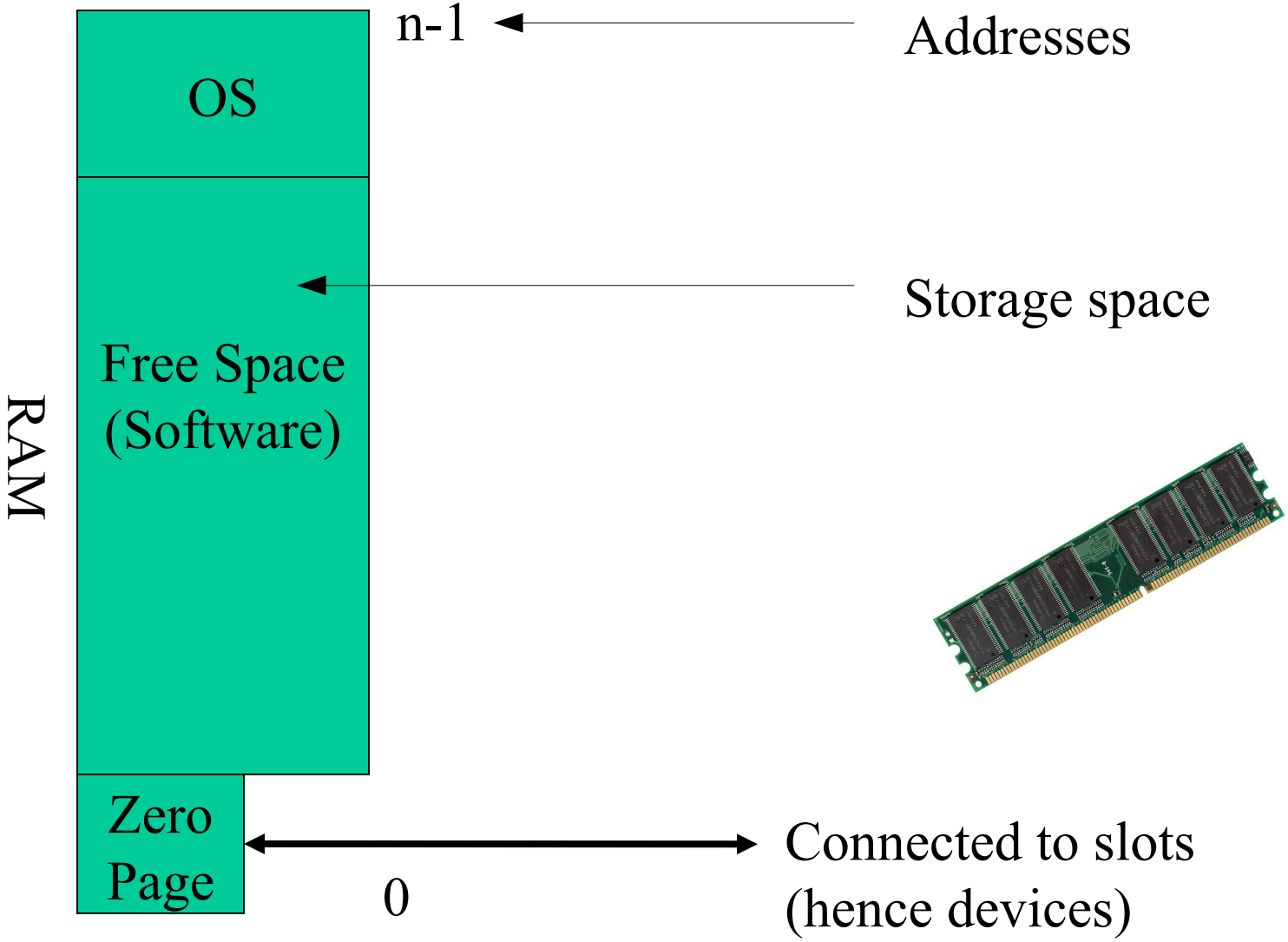
(Industry Standard Architecture vs. Peripheral Component Interconnect)



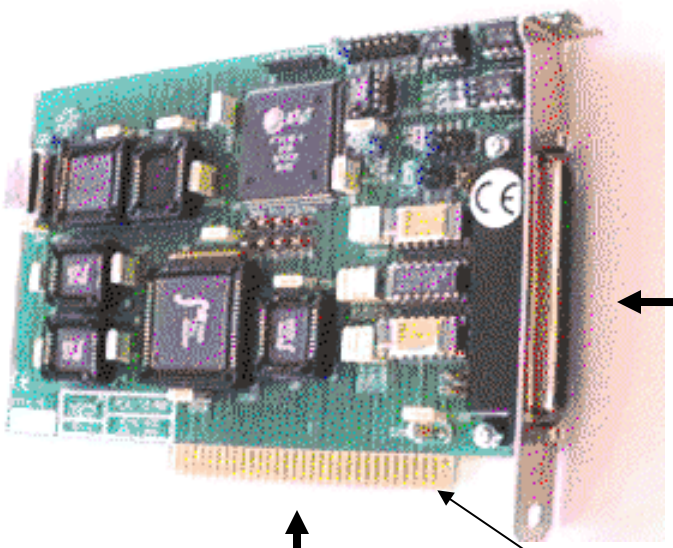
Each bus is relatively independent

➔ Parallel processing

ISA Bus and RAM



Slots, Cards & Ports



Printer Port

One PIN of data (bit)

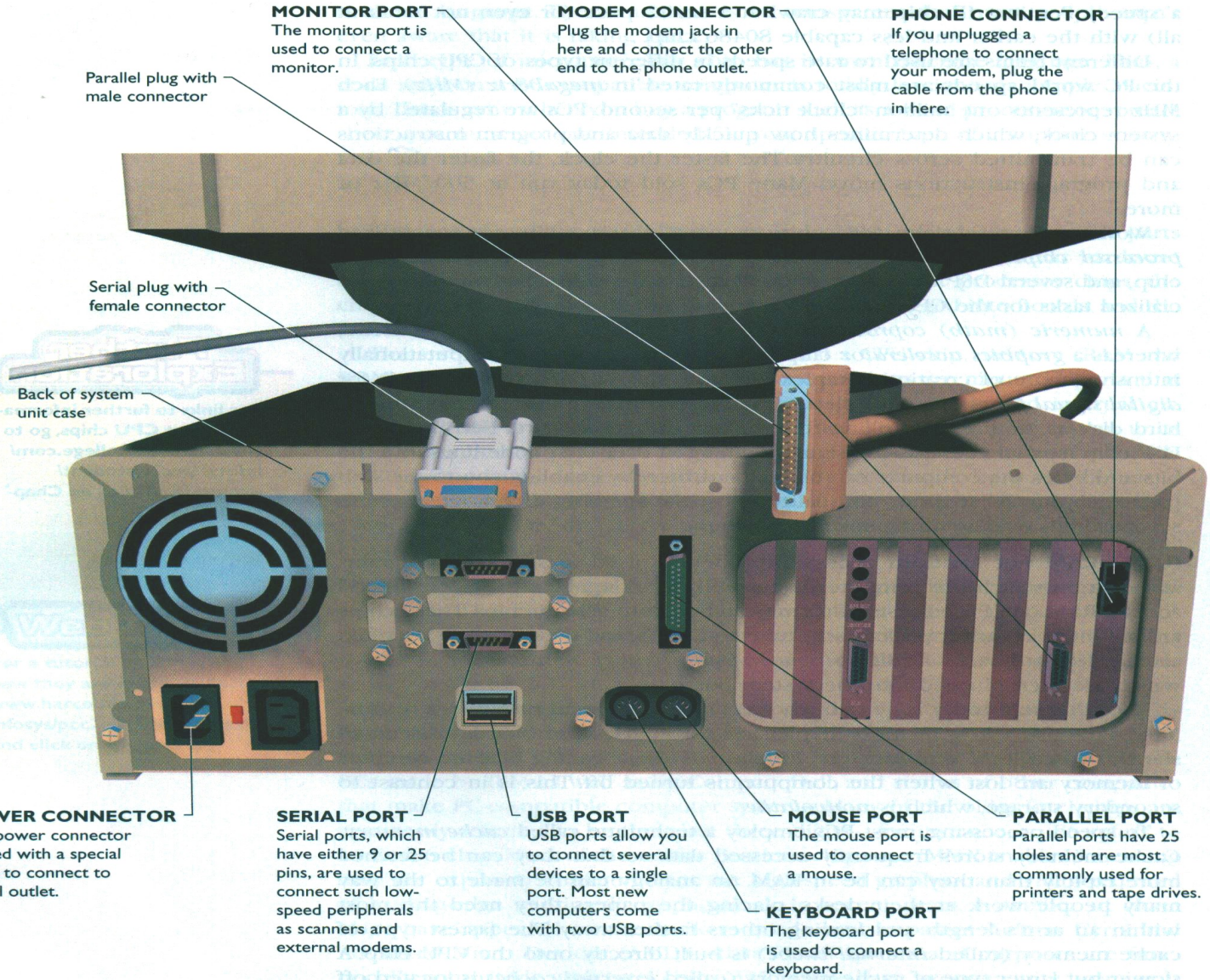
Interconnected

ISA Bus

An ISA Slot

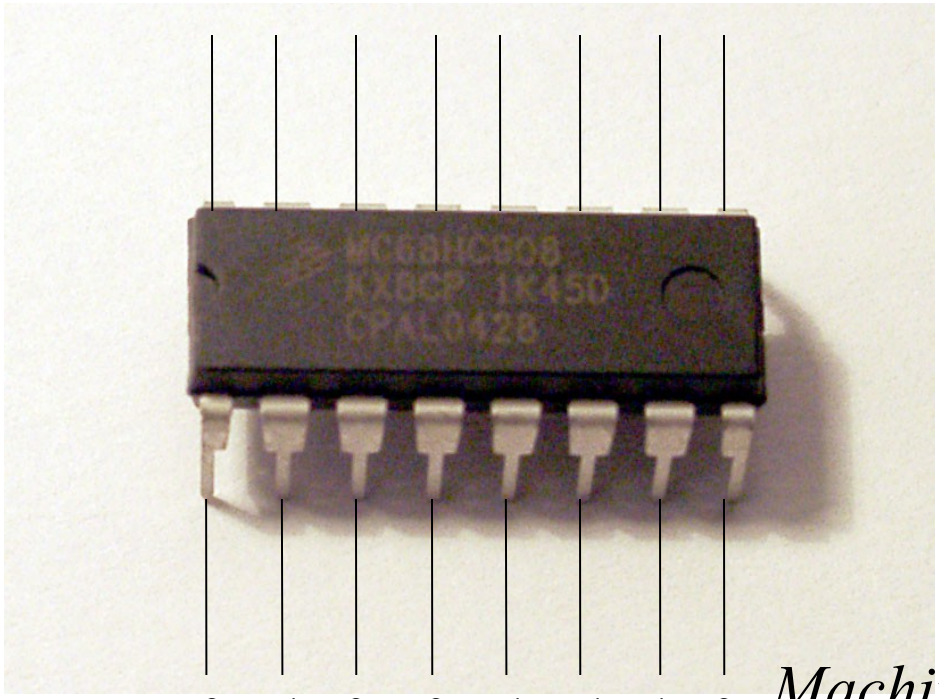
COMP 273
Introduction to Computer Systems

Computer Ports



The Machine's Language

The Machine's Language



Machine Language

Assembler

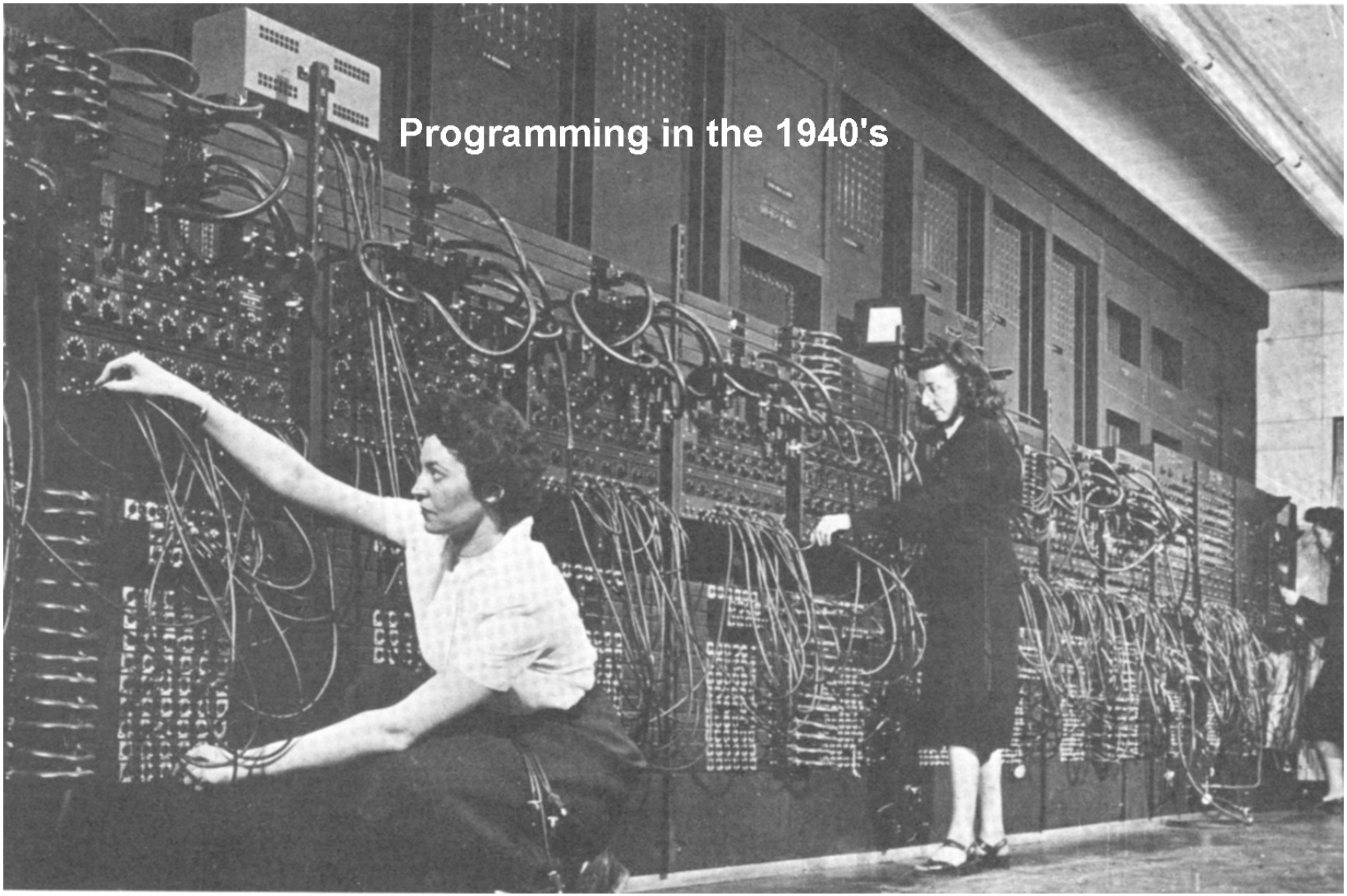
ADD 1, 6

ADD = 01
One = 001
Six = 110

Coded instructions

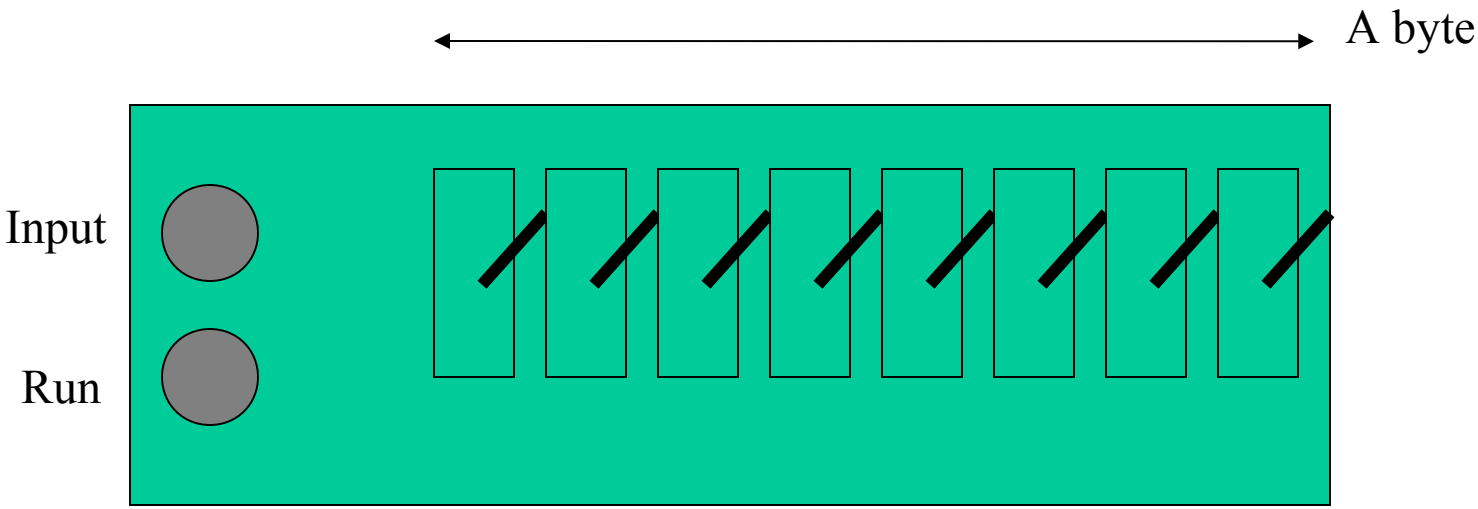
COMP 273
Introduction to Computer Systems

Early Devices



Plug and play machine language – must be something better!

Early Devices

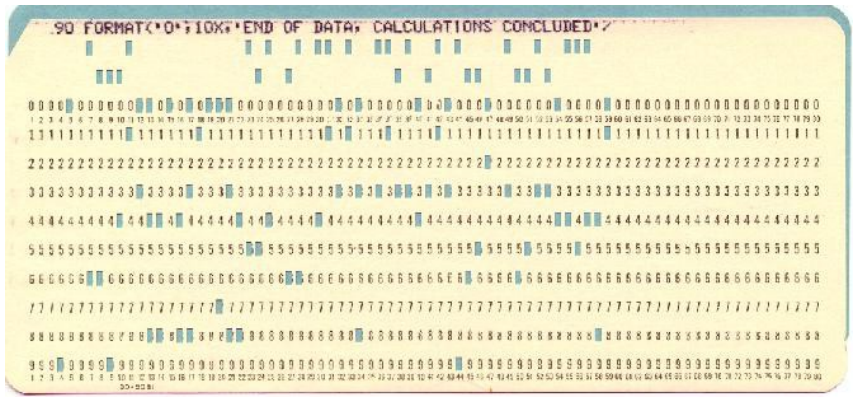


1. Select address first
2. Press input
3. Select command
4. Press input
5. Loop until done
6. Press run

Direct input keyboards – must be something better!

COMP 273
Introduction to Computer Systems

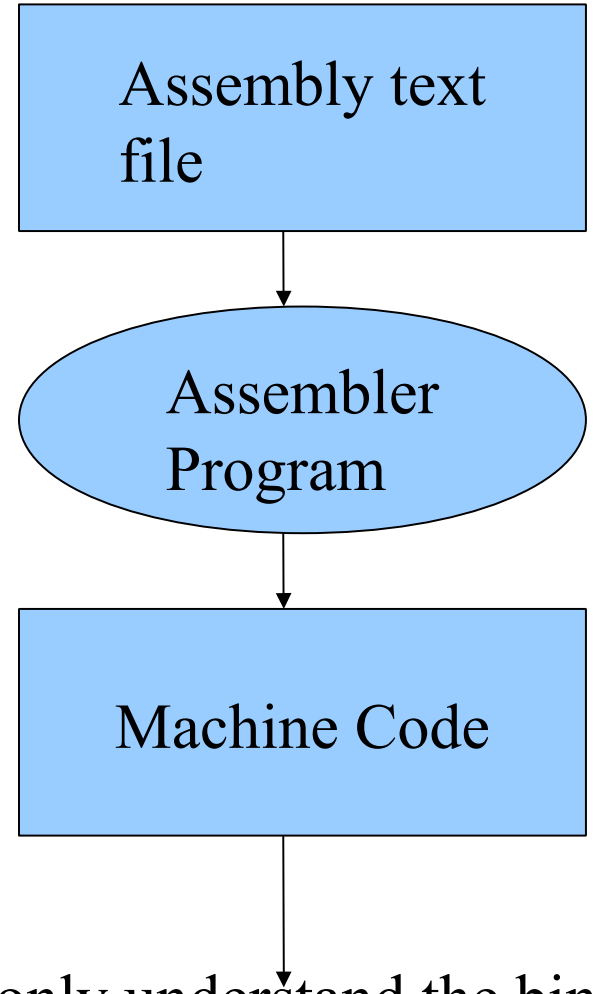
Early Devices



Binary and text combined

Punch card!

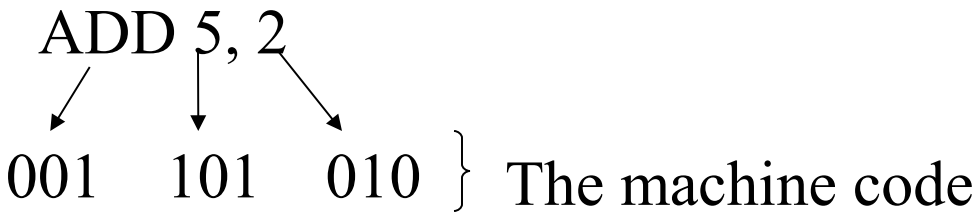
The Assembly Language



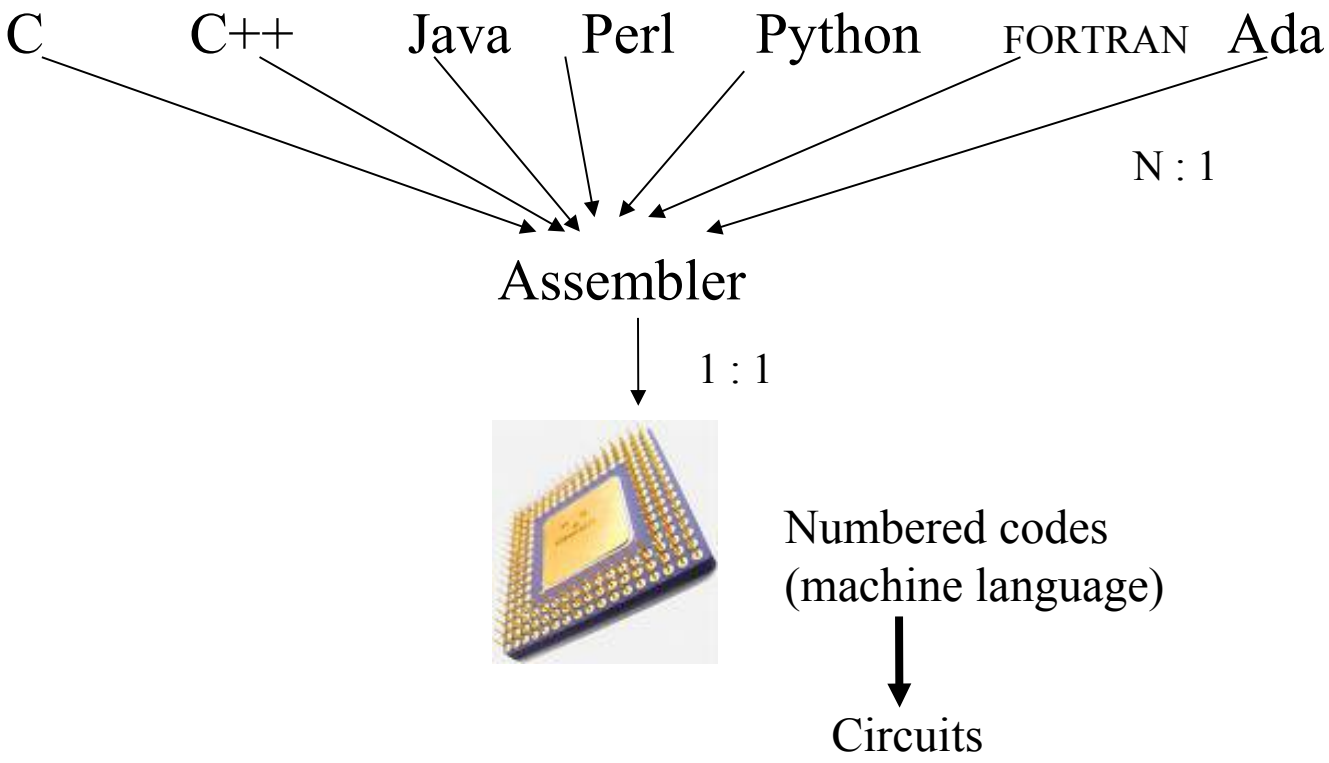
The chips only understand the binary codes

The Assembler Language

- Writing in binary (machine language) is a lot of work so the assembly language was created
- The assembler program (written in machine code) was built to convert assembly into machine code
- Assembly and machine code have a 1:1 mapping



It's all about machine language!



Everything in the end is the same...

Try This Out At Home

- Open your computer and identify:
  - RAM (read the chip markings – how much RAM?)
  - CPU (read the chip markings – brand, speed, ID)
  - Slots (ISA or PCI?)
  - Cards (try to name the different ones)
  - Bus (how many can you identify?)
- Notice when you compile a C or C++ program the source file size is not the same size as the executable (machine language), why?… try to open the executable (a text editor won't help since it is not in text any longer) – What program did help you? (Google will help)

COMP 273
Introduction to Computer Systems