Some Basic Computer Concepts

Binary

- **Decimal is base 10, with symbols 0–9**

- **1907:**

7	1s digit	7 x 10^0
0	10s digit	0 x 10^1
9	100s digit	9 x 10^2
1	1000s digit	1 x 10^3

- **Binary is base 2, with symbols 0, 1**

- **1100:**

0	1s digit	0 x 2^0
0	2s digit	0 x 2^1
1	4s digit	1 x 2^2
1	8s digit	1 x 2^3

Binary

- Figure out the binary number 1101

1 1s digit 1 x 1 = 1
0 2s digit 2 x 0 = 0
1 4s digit 4 x 1 = 4
1 8s digit 8 x 1 = 8
TOTAL: 13

- Just as in decimal, leading zeros don't mean anything.

002341 = **2341**

001010 = **1010**

Binary

- **Practice**

110101	11010110
0001101	01110100
111	11110000
1000	01101101
1111111	
10000000	
1000001	

- How many unique numbers can be expressed with four binary digits?

Answer: 16

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

6-digit Binary Numbers

000000	010000	110000	100000
000001	010001	110001	100001
000010	010010	110010	100010
000011	010011	110011	100011
000100	010100	110100	100100
000101	010101	110101	100101
000110	010110	110110	100110
000111	010111	110111	100111
001000	011000	111000	101000
001001	011001	111001	101001
001010	011010	111010	101010
001011	011011	111011	101011
001100	011100	111100	101100
001101	011101	111101	101101
001110	011110	111110	101110
001111	011111	111111	101111

Some Questions

- If you have n binary digits, how many different numbers can you represent (as a function of n) ?
- How many binary digits must be used to represent the number 1017?
- In general how many binary digits must be used to represent a number m?
- How do you represent -14?

Bits, Bytes, Words, Etc.

- Computers use strings of binary to represent values.

Length	Example	Java Name	Combinations
1	0	boolean (Bit)	2
8	01001010	byte	256
16		short	65536
32		int	4294967296
64		long	18446744073709551616

- The amount a CPU fetches at a time is known as a word.

- On most CPUs, a word is 32 bits (or 4 bytes!).

- What is 4 bits called?

Signed and Unsigned

- If you use the 256 combinations to represent the numbers 0 through 255, you are using the byte **UNSIGNED**.

- If you use the 256 combinations to represent the numbers -128 through 127, you are using the byte **SIGNED**.

Length	Java Name	Lowest	Highest
8	byte	-128	+127
16	short	-32768	+32767
32	int	-2147483648	+2147483647
64	long	-9223372036854775808	+9223372036854775807

- Are these signed or unsigned?

Computer Architecture

- **Most computers are organized around a BUS**
  A network that allows various devices to talk to one another

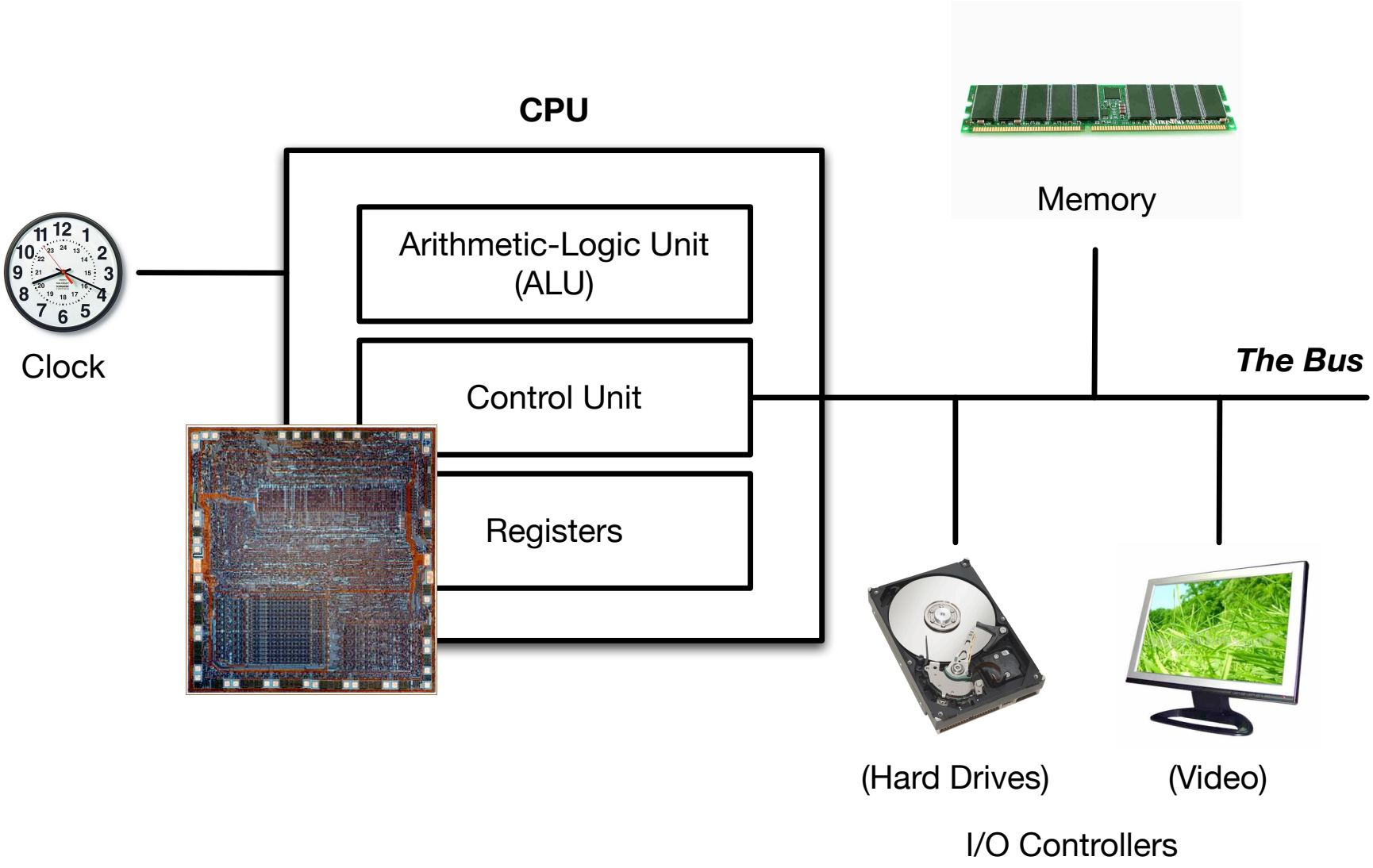
- **The primary device on the bus is the Central Processing Unit or CPU**
  The CPU is pulsed by a clock which keeps things in sync. A 2GHz CPU is pulsed by a clock 2 billion times a second.

- **The CPU communicates with other devices on the bus:**

  Memory

  Peripheral Controllers: hard drive controller, DVDdrive controller, video card, USB controller, etc.

Architecture



Memory

- **Computer memory is a very long string of bits**
- **There are two common kinds of memory:**

  “Random Access” (read/write) memory

  “Read-Only” memory.

    - Cache? FLASH?

The CPU

- The CPU contains **registers**: a few internal slots of superfast memory in which it may store things temporarily.
- One register is the **program counter**, which says where to get the next instruction.
- The CPU performs the following loop, which takes a few clock cycles: the **Decode-Execute Cycle:**
    - Request the instruction from memory as indicated by the program counter.
    - Decode the instruction, increment the program counter.
    - Perform the instruction

An Instruction

- **An instruction may be stored as a 32-bit number, like this:**

  00000000 0101 00100 00011 0100 01101000

  01101000 Instruction name: “Add two registers”

  0100 Register 4

  0011 and Register 3

  0010 put the sum in a register

  0101 put it in Register 5

  00000000 padding to make an even 32 bits

Some Instructions

- Read 32 bits from a location of memory into a register
- Divide one register into another, storing in a third register
- Do the cosine of a register, storing in another register
- Write a register's bits to a location in memory
- Write 32 bits to a controller (to video memory maybe)
- Change the program counter to a very different location
- Copy a chunk of memory from one location to another

CPUs

- **There are many many CPU designs**
  Intel 8086 family (Pentium, etc.)
  PowerPC family (G5, Cell, etc.)
  Sun SPARC family
  ARM family (Intel XScale, StrongARM, ARM7)
  Motorola 680x0 family
  ... on and on

- Each CPU family has a different set of instructions

- Each CPU family has different capabilities

- It’s complicated

Different Ways of Writing Code

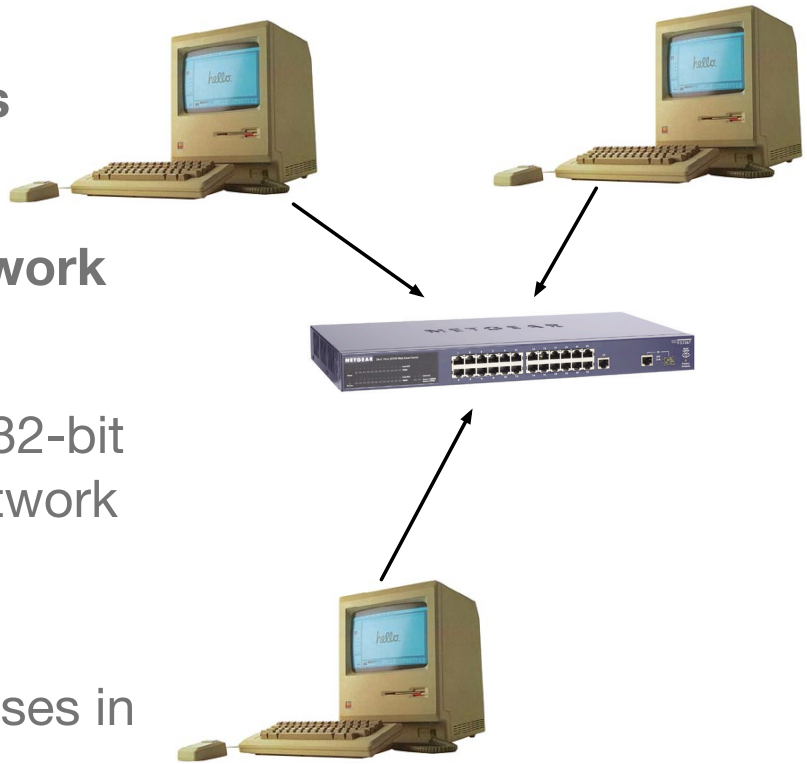
- Hand-write binary CPU instructions (**machine code**) (yeesh, are you nuts?)
- Write in a human-readable form of the instructions (**assembly code**)
    - **Assembler:** changes your assembly code into CPU instructions
- Write in a **high-level programming language** like C/C++/Fortran
    - **Compiler:** translates the language into CPU instructions

Different Ways of Writing Code

- Write in a **high-level bytecode-compiling language** like Java
  - **Bytecode Compiler**: translates Java into instructions for a made-up CPU ("**bytecode**")
  - **Virtual Machine:** automatically translates the bytecode into instructions for your CPU when you run the program
- Write in a **portable high-level *online* language** like Python (or Lisp)
  - **Python Interpreter:** translates the Python on-the-fly and performs it. Can also take input as you type it.
  - **Bytecode Compiler:** translates the Python into a byte code like Java's byte code. The bytecode can then be used in the interpreter faster because it doesn't have to spend as much energy translating it.
  - **Compiler:** translates the Python into the machine code of your computer so you can run it directly as an executable program.

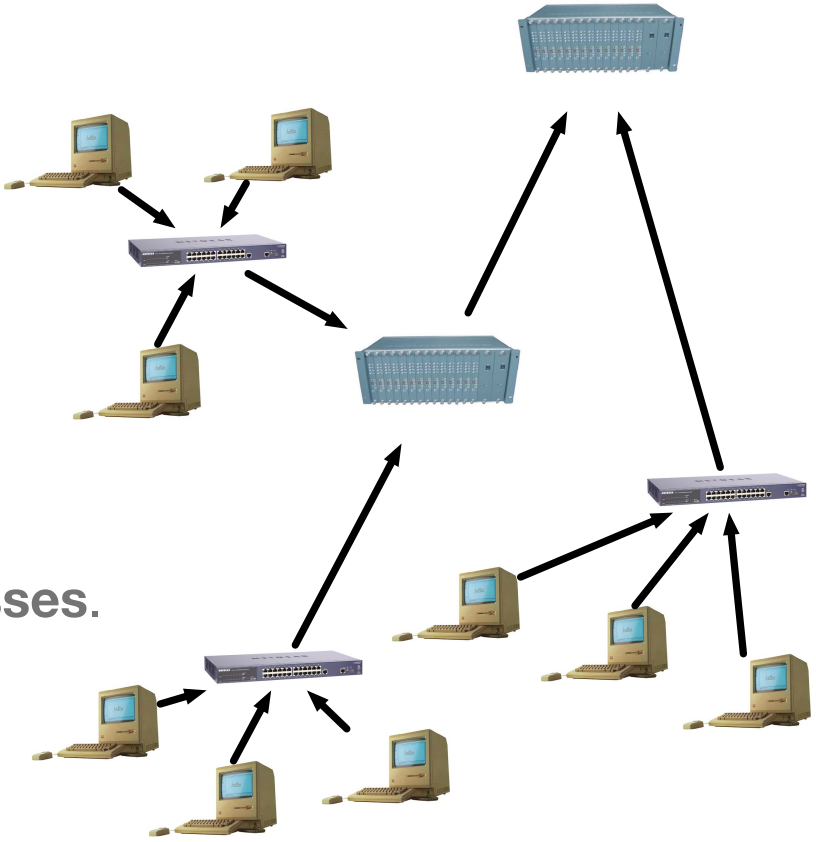
Networks

- A local area network is a **larger, slower bus** that connects immediate computers
- Computers communicate by sending **packets** (chunks of binary) to one another
- Usually computers are all connected to a **network switch**
- Each computer is assigned an **IP address** (a 32-bit number) which uniquely identifies it on the network
  - Example: 129.174.241.198 is zeus
- Computers on the same network have addresses in the same numerical range (called a **domain**)
  - Zeus is in GMU's domain (129.174)



Networks

- Local area networks may themselves be connected to a larger wide area network, and be part of its domain
- Wide area switches may be called **routers** or **gateways**
- Wide area networks can be part of even bigger wide area networks
- The **internet** is a big wide-area network
- It's inconvenient to use IP addresses. So we make up names for the computers and their domains, called **Internet Addresses**.
  - Internet Address: **zeus.ite.gmu.edu**
  - Domain: **gmu.edu**
  - "Top-Level" Domain: **edu**



Protocols

- Streams of packets from one computer to another on a network can take various standard formats, known as **protocols**.
- Sending Email uses the protocol **SMTP: Simple Mail Transfer Protocol**
- Connecting to zeus via SSH or SFTP uses the protocols **SSH: Secure Shell** and **SFTP: Secure File Transfer Protocol**
- Connecting to a World-Wide Web server to download a web page uses the protocol **HTTP: Hyper-Text Transfer Protocol**

URLs

- A **URL (Unique Resource Locator)** is a way to uniquely locate a piece of information on the world-wide web

- **URLs often contain three parts:**

  The **protocol** to use to access the information
  The **internet address** of the server providing the information
  The location of the information **on the server**

- **http://cs.gmu.edu/~eclab/projects/robots/flockbots/**

http://	The protocol to use
cs.gmu.edu	The address of the server
/~eclab/projects/robots/flockbots/	Where the information is on the server

Undergraduate Computer Resources at GMU

- **Set up Email**
  (via MasonLive) I doubt you need instruction on this one.

- **Get a UNIX Computer Account**
  mason.gmu.edu is a Sun Solaris server available for you. Several CS classes (330, 480, etc.) will require usage of this machine.
  http://itusupport.gmu.edu/STG/masonaccount.asp

- **Set up a Personal Website**
  You'll use the UNIX computer account above.
  http://itusupport.gmu.edu/STG/webpage.asp

- **Read Journal Articles / use protected Library facilities from OFF campus**
  https://login.mutex.gmu.edu/login

- **Find the GMU Computer Labs**
  https://classtech.gmu.edu/lablocations.cfm

Undergraduate Computer Resources at IT&E

- **Get a Linux Computer Account *(more apropos to CS students)***
  You will first need to get the UNIX Computer Account from GMU to get this additional account from IT&E. The machine is called **zeus.ite.gmu.edu.** Various CS classes may require this machine.
  http://labs.ite.gmu.edu/index.php/FAQ/ClusterAccount

- **Find the IT&E Computer Labs**
  These labs have machines which boot into Linux or Windows. When in Linux, the machines share a filesystem with the Linux Cluster above.
  http://labs.ite.gmu.edu/

- **Use VPN, print, FTP, access databases (Oracle etc.), and more...**
  http://labs.ite.gmu.edu/index.php/FAQ/FAQ

Russian Peasant Multiplication

- To multiply two numbers A and B
- Iteratively divide A by two (taking the floor if it's odd) and multiply B by 2 until A = 1
- Example: 22 x 12 = 264

22	x	12
11	x	24
5	x	48
2	x	96
1	x	192

Why it works?
(2k+1)*n = k*(2n) + n

- Add all of the "B" values for which the corresponding "A" values were **odd.**
- Example:

22	x	12	10110	x	1100
11	**x**	**24**	**1011**	**x**	**11000**
5	**x**	**48**	**101**	**x**	**110000**
2	x	96	10	x	1100000
1	**x**	**192**	**1**	**x**	**11000000**

24 + 48 + 192 = 264

11000+110000+11000000 = 100001000

The *Nim* Game

A simple version of the ***nim*** game is played as follows: Two players alternate in removing stones from three piles initially containing two, two, and three stones, respectively. The player who picks up the last stone wins. At any given turn a player can pick one or more stones from a single pile; at least one stone has to be picked every time.

Examples:

How would you play? Would you like to play first?

A Solution for the Nim Game

- Write all numbers in binary and XOR them

1	1	11	1
10	11	101	101
11	101	111	1001
00	111	001	1101

- If the result is 0 whoever play first loses, if the result is non-zero the first player needs to pick as many stones as needed to make it zero

First player to play loses

1	11	1
11	101	101
10	110	100
00	000	000