CISC 101

Welcome!

Elements of Computing Science I

http://sites.cs.queensu.ca/courses/cisc101

- Most of the lecture notes and other material will be (and is) posted

Slides courtesy of Dr. Alan McLeod

CISC 101 Introduction

- Me: Sarah-Jane Whittaker
  - PhD Candidate, Queen's School of Computing
  - sarah@cs.queensu.ca
  - 550 Goodwin Hall
- TAs (thus far)
  - Muhammad Aboelfotoh
  - Mallory Ketcheson

Slides courtesy of Dr. Alan McLeod

Queen's School of Computing

SCHOOL OF COMPUTING

- www.cs.queensu.ca
  - Linked on main page of course web site
- Lots of information
  - Undergrad and graduate programs and courses
  - What's happening in the School
  - … and more!

Slides courtesy of Dr. Alan McLeod

Labs and Tutorials

- No scheduled labs or tutorials this week
  - They begin next week
- Tutorials
  - The period before your lab
  - In Jeffrey Hall – see the course web site
  - Gives you a chance to talk to your TA or for the TA to present material to you before the lab
  - You will write tests in your tutorial

Slides courtesy of Dr. Alan McLeod

Labs and Tutorials – Cont.

- Labs
  - Two hours in JEF155 or JEF157
  - Work, work, work! TAs are there to help
- You can attend more than one section
  - Good way to get extra help
  - If there are too many attendees, priority will be given to students in that section

Slides courtesy of Dr. Alan McLeod

Lectures

- Lectures will not go for three hours straight
  - There will be a break at the mid-point
  - May end early if I have covered all the material
- I will be programming examples in lectures
  - Feel free to bring your laptops and code as well
- Lecture slides will be posted on the website

Slides courtesy of Dr. Alan McLeod

Grading

- Marking scheme
  - 24% - three tests
    - Weeks 4, 7 and 10
  - 15% - three assignments
    - Due on Sundays before midnight in weeks 3, 6 and 9
  - 61% - <u>one</u> final exam
- Tests are written in your tutorial
  - 40 minutes in length, on paper, no aids
- Assignments are submitted in Moodle
  - One assignment submission per person!

Slides courtesy of Dr. Alan McLeod

Grading - Cont.

- Note that for the final exam:

If you obtain less than 50% on the final exam then that will be your final grade. In this case, your test and assignment marks will not count.

- *This is not meant to stress you out!*
- The best strategy is to learn the material as we go
  - Best accomplished by attending tutorials and labs

Slides courtesy of Dr. Alan McLeod

Assignment Rules

- Each assignment will have its own marking scheme
  - Marks and comments will be entered into Moodle by your TA
- See the *Assignment → Submission* page for instructions on uploading to Moodle
- No late submissions!
  - If you do not submit by the deadline, you will receive a grade of zero
  - Don't leave your assignment to the night before!!!

Slides courtesy of Dr. Alan McLeod

Assignment Rules - Cont.

- "Group Efforts"
  - I encourage you to discuss your difficulties with your peers, myself and your TAs
  - You may look at other people's code
  - **You cannot copy other people's code!**
  - Submissions will be electronically and physically checked for code duplication
  - If you are caught with duplicated code then it is considered a violation of academic integrity
    - More on that later …
  - You will not learn anything if you copy someone else's code!

Slides courtesy of Dr. Alan McLeod

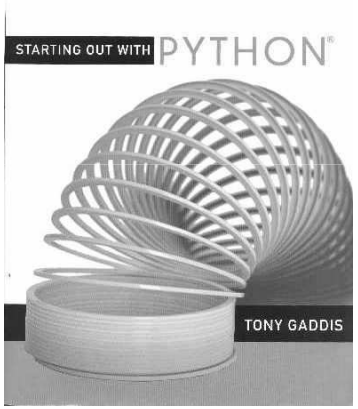
Moodle

- News and updates
- Assignment and test marks
- Question: would you like forums?
  - One for each assignment?
  - *Study group* forum already available
- Don't change your time zone
  - Always use the default: "Server's local time"
  - Assignment deadlines use this time

Slides courtesy of Dr. Alan McLeod

Resources

- Recommended textbook:



- Used last two years
- Refers to an older version of Python
- See *Resources → Textbook* on the course website

Slides courtesy of Dr. Alan McLeod

Resources - Cont.

- Older CISC 101 textbooks will not refer to the Python language
- Many other learning resources are available
  - See the *Resources* page on the course website
- "*Should I buy the textbook?*" (You ask!)

Slides courtesy of Dr. Alan McLeod

Some Policies

- Email
  - Complex questions can not be answered well via email
    - *e.g.*, code debugging
  - Both I and the TAs may ask that you see us in person
- Code of Conduct
  - Students are required to abide by the Queen's Code of Conduct and all established policies
    - *e.g.,* Computer User Code of Ethics
  - Behaviour that violates the code will not be tolerated
    - *e.g.,* harassment or discrimination
  - Sanctions will be pursued

Slides courtesy of Dr. Alan McLeod

Some Policies – Cont.

- Academic Integrity
  - "a commitment to the fundamental values of honesty, trust, fairness, respect and responsibility"
  - Offenses include plagiarism
  - Penalties are ...
    - First offense: a grade of zero on the assignment or test
    - Second offense: a grade of zero for the course
  - Please review the University's official policy
    - Link on the course website (*Policies → Academic Integrity*)

Slides courtesy of Dr. Alan McLeod

Some Policies – Cont.

- Academic Accommodations
  - Not a problem, but please contact both the instructor and Disability Services as soon as possible
    - May need advanced notice to accommodate you
- Late Submissions
  - Late submissions are not permitted
  - A missed assignment or test will be marked as zero
  - Extensions will be provided where warranted
    - *e.g.,* illness, personal crisis
  - Contact the instructor as soon as you can
    - Failure to do so will still result in a grade of zero

Slides courtesy of Dr. Alan McLeod

A Few More Things …

1. I am unlikely to be reached by calling my office
2. When e-mailing try to avoid using your `hotmail` or `gmail` or non-`queensu` address
3. Check to make sure you can access the CISC 101 Moodle space … in a few days
   - E-mail me if you can't
4. Update your Moodle profile (if necessary)
5. Please communicate with me!
   - I can't fix problems with the course if you don't tell me about them

Slides courtesy of Dr. Alan McLeod

Is This a Python Course?

- So, what is this course about, anyways?
- From the calendar description:

> *CISC101 is an "Introduction to algorithms: their definition, design, coding, and execution on computers. Intended for students who have no programming experience. All or most assignment work will be completed during lab time."*

Doesn't say anything about

Slides courtesy of Dr. Alan McLeod

CISC 101 or CISC 121?

- Do you already have some programming experience?
  - You may wish to take CISC 121 instead
- See the *About → 101 or 121* page on the website
- Contact me or the CISC 121 instructor
  - Margaret Lamb
  - `malamb@cs.queensu.ca`

Slides courtesy of Dr. Alan McLeod

Purpose of the Course

- The best way to learn about programming is to do it – *of course*!
- Programming is a creative process…

Slides courtesy of Dr. Alan McLeod

Purpose of the Course - Cont.

- I will teach you about …
  - The basic structure and operation of the hardware we are using to express ourselves
  - Some time-tested techniques used with the language to efficiently express algorithms
  - Some common algorithms
  - Good programming style and best practices
- But, can I teach you how to program?
  - I certainly could not teach you how to paint!!!

Slides courtesy of Dr. Alan McLeod

Purpose of the Course - Cont.

- You will …
  - Figure out how to write a program that does what you want
  - Have a chance to see if you like programming
  - Be exposed to other aspects of Computer Science
  - Have fun!!!

Slides courtesy of Dr. Alan McLeod

Playtime!

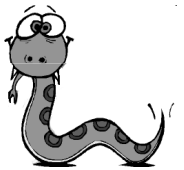
- Play with Pygame

- Do some GUI stuff with Tkinter and maybe PyQT
- Use other modules for math and physics, or sound and images
- And more …

Slides courtesy of Dr. Alan McLeod

Why Python?

- Why not?
- Python is a solid language like many others
  - C++, C#, VB, Java, Delphi, *etc.*
- It is:
  - Easy to learn
  - Powerful
  - Object-oriented
  - Available for many platforms
  - Structured the same as other modern languages
  - Not named after a snake!!!



Slides courtesy of Dr. Alan McLeod

Python

- Development tools for novice Python programmers can be downloaded from `www.python.org`
- See the course website's *Resources → Python Software* page for more information

Slides courtesy of Dr. Alan McLeod

What You Need to Do

- Decide to get the textbook or not
- Look over course web site
  - There is a great deal of information to review!
  - Install Python on your own computer?
- Let me know if you need to change your lab section from what is listed in your timetable
- Contact me if you have any problems or questions!

Slides courtesy of Dr. Alan McLeod

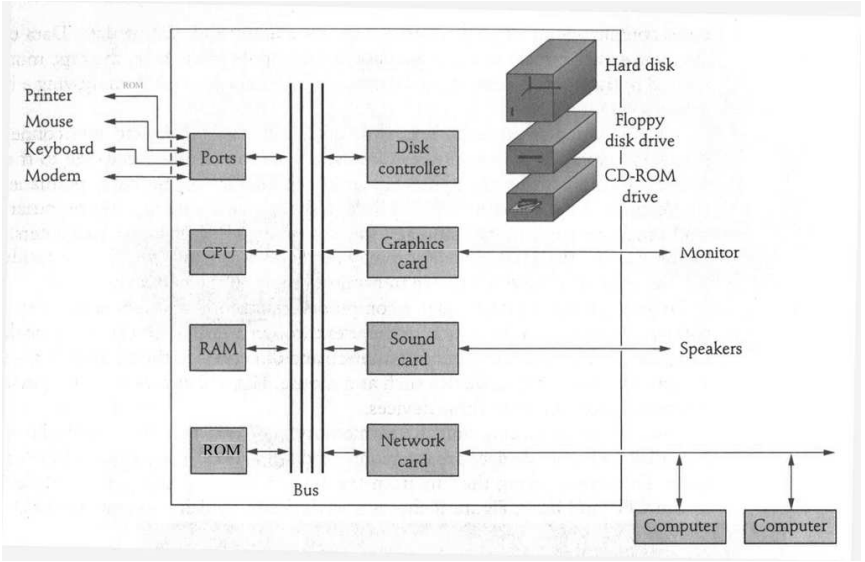
Getting Started

- Take some of the "MAGIC" out computers
  - An overview of computer architecture
  - How did the technology get to where it is today?
    - von Neumann Architecture
    - ENIAC
    - The transistor
    - Integrated circuits

`http://www.howstuffworks.com/microprocessor.htm`

Slides courtesy of Dr. Alan McLeod

Computer Architecture (PC)



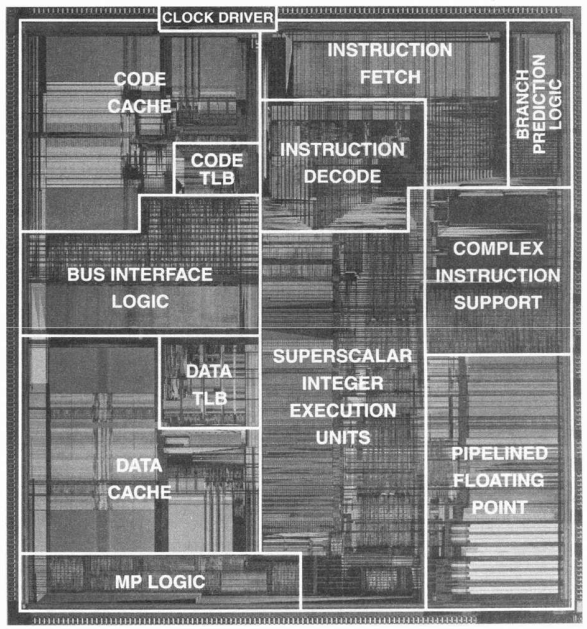
Slides courtesy of Dr. Alan McLeod

CPU

- Central Processing Unit
  - The heart of the computer
  - Consists of millions of transistors on a single chip

  - Next slide shows the physical layout of a Pentium chip

Slides courtesy of Dr. Alan McLeod



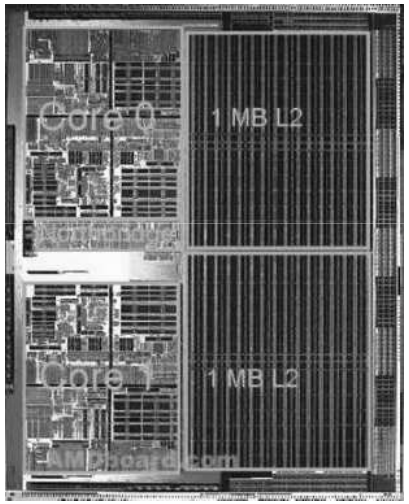
Slides courtesy of Dr. Alan McLeod

CPU - Cont.

- The latest is Twelve Core
  - Twelve processors on one chip!
- Characterized by their clock speed
  - Controls the number of operations per second that they can carry out
- The first PC contained an 8080 chip
  - Introduced in 1979
  - Had a clock speed of 2 MHz
    - That's 2,000,000 cycles per second
- Now Pentium IV's operate at at least 2 GHz!

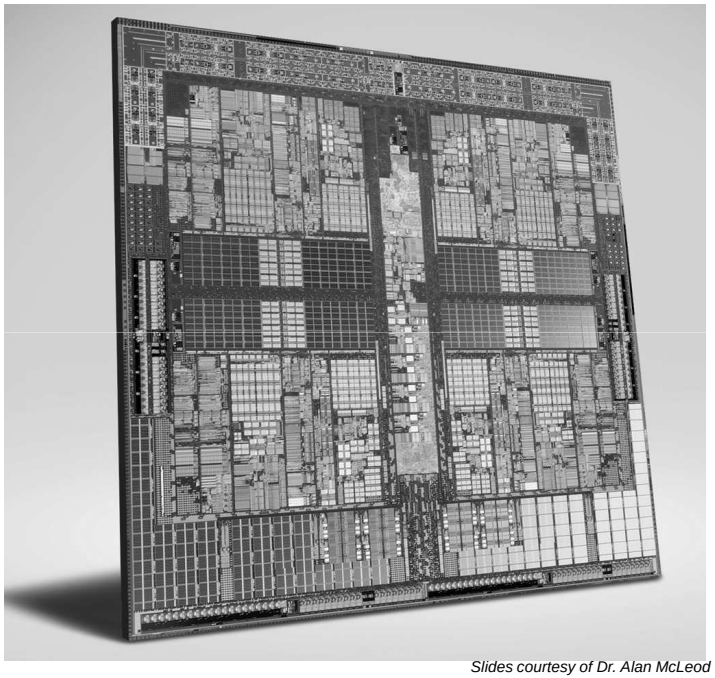
Slides courtesy of Dr. Alan McLeod

Dual Core Processor



Slides courtesy of Dr. Alan McLeod

Quad Core



Winter 2011

Slides courtesy of Dr. Alan McLeod

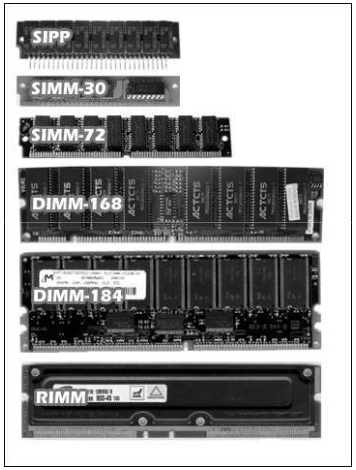
CPU - Cont.

- Responsible for …
  - Locating and carrying out program instructions from RAM
  - Carrying out arithmetic operations on data stored temporarily in a few “registers”
  - Moving data between RAM and other storage devices
    - … except for those devices that can use Direct Memory Access (DMA)

Slides courtesy of Dr. Alan McLeod

RAM and ROM

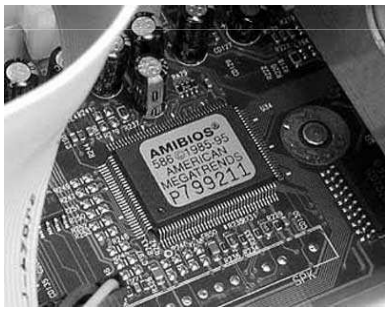
- Random Access Memory
  - Volatile memory
    - When the power goes off, the data disappears
  - RAM holds data and program instructions



Slides courtesy of Dr. Alan McLeod

RAM and ROM

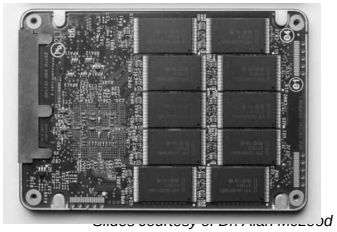
- Read Only Memory
  - Not volatile
  - Contains code and BIOS data used to start (“boot”) the computer



Slides courtesy of Dr. Alan McLeod

Other Components

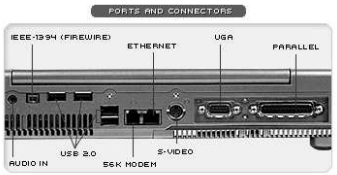
- Disk storage
  - Data stored on magnetic or optical media
  - Not volatile! (Hopefully!)
  - Disk read/write operations are much slower than operations carried out in RAM by the CPU
  - Program code is first loaded from the disk to RAM and then executed
  - Modern (but still rather pricey!) solid state drives just have banks of non-volatile RAM



Slides courtesy of Dr. Alan McLeod

Other Components

- Data ports
  - Serial (RS232), Parallel, USB, PS2, *etc.*
  - Provide input/output for the user
    - Via the keyboard, mouse, monitor, sound card, microphone, printer, scanner, joystick, webcam, *etc.* (you get the idea!)
- Modems, network cards and/or wireless adapters
  - Provide a means of connecting to other computers



Slides courtesy of Dr. Alan McLeod

Booting Your Computer

- Initial BIOS is loaded from ROM into RAM
  - Basic Input-Output System (BIOS)
  - Computer detects any attached hard/optical/flash drives
- Computer then looks at the MBR on the hard drive
  - Master Boot Record (MBR)
  - Small portion at the beginning of the disk
  - Describes the layout of the drive
  - Loads a boot loader
    - Knows how to tell the operating system (OS) to load

Content courtesy of Benjamin Hall

Slides courtesy of Dr. Alan McLeod

Booting Your Computer – Cont.

- Computer then loads the OS
  - Loads drivers necessary to control the hardware
- Last task is to load the Graphical User Interface
  - Provides you with your operating environment
- You can now log in and write your programs!

Content courtesy of Benjamin Hall

Slides courtesy of Dr. Alan McLeod

Processes and Process Switching

- When a program executes, the operating system does so in a *process*
  - Each program has its own process
  - Many complex programs launch additional processes
- A CPU can only manage one process at a time!
- The OS must *switch* between processes
  - Allow users to run several programs simultaneously
  - Done with the *scheduler*
    - Takes care of what process gets to use the CPU at given time
    - May also need to manage several CPUs

Content courtesy of Benjamin Hall *Slides courtesy of Dr. Alan McLeod*

Processes and Process Switching – Cont.

Less than a quarter of the processes running on a relatively idle Mac laptop:

```
UID PID PPID C STIME TTY TIME CMD
0 1 0 0 0:01.92 ?? 0:01.99 /sbin/launchd
0 10 1 0 0:02.70 ?? 0:18.84 /usr/libexec/kextd
0 11 1 0 0:00.67 ?? 0:01.50 /usr/sbin/DirectoryService
0 12 1 0 0:00.16 ?? 0:00.23 /usr/sbin/notifyd
0 13 1 0 0:00.09 ?? 0:00.19 /usr/sbin/syslogd
0 14 1 0 0:00.06 ?? 0:00.10 /usr/sbin/diskarbitrationd
0 15 1 0 0:01.06 ?? 0:01.80 /usr/libexec/configd
0 16 1 0 0:00.28 ?? 0:00.40 /usr/sbin/blued
1 17 1 0 0:00.10 ?? 0:00.25 /usr/sbin/distnoted
65 19 1 0 0:00.18 ?? 0:00.27 /usr/sbin/mDNSResponder -launchd
0 22 1 0 0:00.16 ?? 0:00.28 /usr/sbin/securityd -i
0 25 1 0 0:00.16 ?? 0:00.20 /usr/sbin/ntpd -c /private/etc/ntp-restrict.conf -
   n -g -p /var/run/ntpd.pid -f /var/db/ntp.drift
0 26 1 0 0:00.21 ?? 0:00.58 /usr/sbin/httpd -D FOREGROUND
0 27 1 0 0:00.05 ?? 0:00.07 /usr/sbin/krb5kdc -
```


Content courtesy of Benjamin Hall *Slides courtesy of Dr. Alan McLeod*

Binary Numeric System

- Computers store and use data with the *binary numeric system*
  - A single *bit* is a 0 or a 1
    - Magnetic orientation (*e.g.*, hard-disk drives)
    - Presence or absence of electrical charge (*e.g.*, RAM)
    - And more …
  - A *binary number* is composed of multiple bits

1 0 1 1 0 0 0 1 1 0 1 0

Slides courtesy of Dr. Alan McLeod

Binary Numeric System – Cont.

- Binary numbers can be easily converted to decimal values
  - Assign an index *i* to each bit from right to left
    - Start with 0 and increase each index by 1
  - Sum all values 2^i where the bit at index *i* contains a 1

$2^{11} + 2^9 + 2^8 + 2^4 + 2^3 + 2^1 = 2842$

11	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
1	0	1	1	0	0	0	1	1	0	1	0

Slides courtesy of Dr. Alan McLeod

Numerical Bases

- Decimal numbers have a *base* of 10
  - *e.g.*, $942 = (9 \times 10^2) + (4 \times 10^1) + (2 \times 10^0)$
- Binary numbers have a base of 2
  - *e.g.*, $1100 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)$
- Octal numbers have a base of 8
  - *e.g.*, $721 = (7 \times 8^2) + (2 \times 8^1) + (1 \times 8^0)$
- Hexadecimal values have a base of 16
  - Digits 0-9 and letters A-F for values 10-15
  - *e.g.*, A0C = $(10 \times 16^2) + (0 \times 16^1) + (12 \times 16^0)$

Boolean Values and Logical Operators

- The *Boolean domain* is {0, 1}
  - A *Boolean value* is either 0 or 1
    - 0 is often interpreted as false
    - 1 is often interpreted as true
- *Boolean logical operators*
  - Use Boolean values for input and output
  - Will compare or examine values and return a result
- Three primary operators
  - `AND`
  - `OR`
  - `NOT`

Boolean Values and Logical Operators

- Consider Boolean variables *a* and *b*
  - *a* and *b* can contain any Boolean value
  - There are four possible combinations of values for *a* and *b*
  - These combinations are shown to the right in a *truth table*

a	*b*
0	0
0	1
1	0
1	1

- AND ($a \wedge b$)
  - True only if both *a* and *b* are true
  - False otherwise

a	*b*	($a \wedge b$)
0	0	0
0	1	0
1	0	0
1	1	1

Boolean Values and Logical Operators

- OR ($a \vee b$)
  - True if at least one of *a* or *b* is true
  - False, otherwise

a	*b*	($a \vee b$)
0	0	0
0	1	1
1	0	1
1	1	1

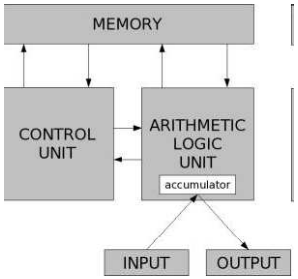
- NOT (! *a*)
  - True if *a* is false
  - False if *a* is true

a	(! *a*)
0	1
1	0

- Why is Boolean logic important?
  - You will be using these operators to test conditions in your algorithms!

von Neumann Architecture

- Created in 1944/5
- The structure we have been referring to thus far
  - Separate units for input and output
  - Data is stored in a separate memory location
  - The ALU carries out instructions on data items moved into the ALU
  - The Control Unit acts as a stage manager



Slides courtesy of Dr. Alan McLeod

von Neumann Cycle

- **Fetch**
  - The address of the next instruction is read from the instruction counter. The next instruction is read from this memory address to instruction register.
- **Decode**
  - The instruction is translated to a format that is usable for the execution unit by the decoder.
- **Fetch operands**
  - Depending to the actual instruction operands from a memory location have to be fetched to be accessible for the execution unit.
- **Execute**
  - The arithmetical logical unit performs the operations and writes the results to registers or memory according to the instruction.
- **Update instruction counter**
  - The instruction counter is incremented for the next cycle. Now the first step can start again.

Slides courtesy of Dr. Alan McLeod

von Neumann Architecture - Cont.

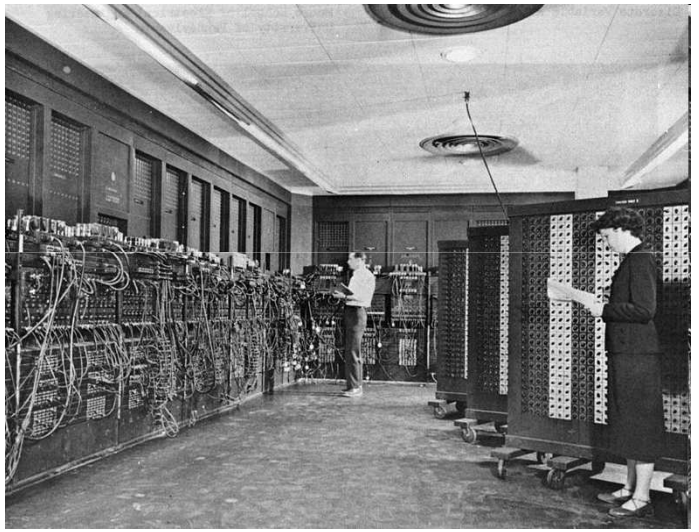
- Named after John von Neumann
  - Idea most likely developed by J. Presper Eckert and John Mauchly
  - All working on the ENIAC computer at the Moore School at the University of Pennsylvania at the same time

von Neumann

Mauchly (L) and Eckert (R)

Slides courtesy of Dr. Alan McLeod

ENIAC



Slides courtesy of Dr. Alan McLeod

ENIAC - Cont.

http://news.cnet.com/1606-2_3-29770.html

Slides courtesy of Dr. Alan McLeod

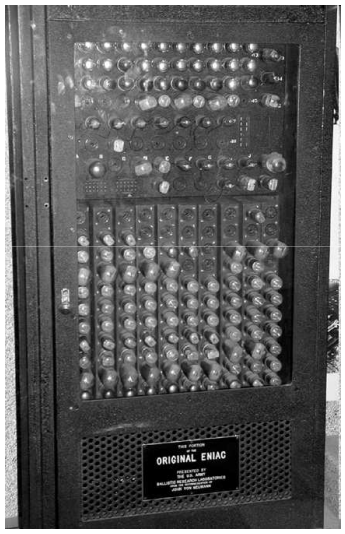
ENIAC - Cont.

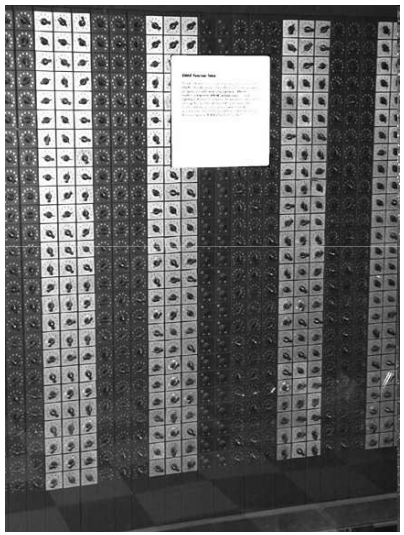
From Wikipedia

- "Electronic Numerical Integrator And Computer"
- First electronic computer, built in the late 1940's to calculate artillery firing tables
- ENIAC contained 17,468 vacuum tubes, 7,200 crystal diodes, 1,500 relays, 70,000 resistors, 10,000 capacitors and around 5 million hand-soldered joints
- It weighed 30 tons, took up 680 square feet (63 m^2), and consumed 150 kW of power

Slides courtesy of Dr. Alan McLeod

ENIAC - Cont.



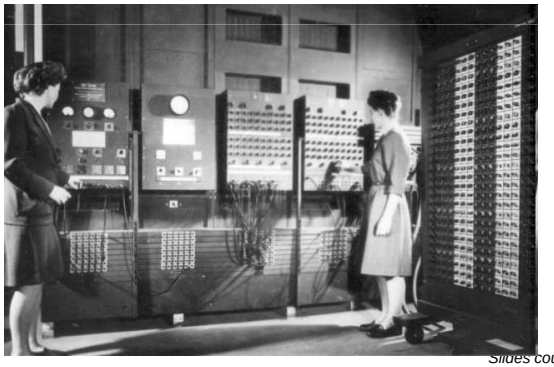


Slides courtesy of Dr. Alan McLeod

ENIAC - Cont.

- Six women (inducted into the Women in Technology International Hall of Fame) took several weeks to manually enter a single program into the machine

Kay McNulty, Betty Jennings, Betty Snyder, Marlyn Wescoff, Fran Bilas and Ruth Lichterman



Slides courtesy of Dr. Alan McLeod

Operating the “Differential Analyzer” (an analog mechanical calculator) in the basement of the Moore School (1942-45)

Slides courtesy of Dr. Alan McLeod

ENIAC - Cont.

- In 1942, their annual salary was $1,620
- The job title of all of the women was “computer”!

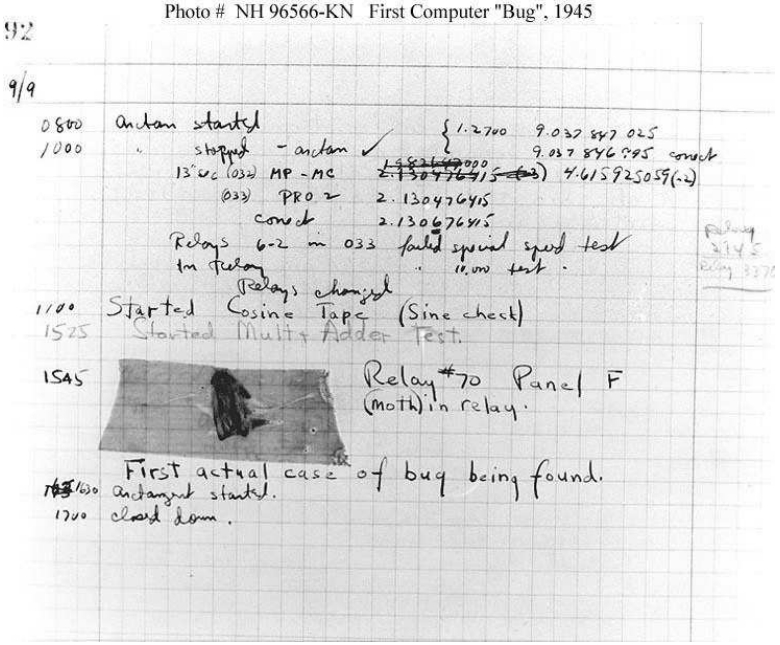
Slides courtesy of Dr. Alan McLeod

Aside – The First Computer Bug

- In 1947 Grace Murray Hopper was a technologist on a Mark II Aiken Relay Calculator at Harvard University
- She logged the following:

Slides courtesy of Dr. Alan McLeod

Photo # NH 96566-KN First Computer "Bug", 1945



Slides courtesy of Dr. Alan McLeod

The First Computer Bug - Cont.

- The word went out that she had "debugged" the machine
- Grace Murray Hopper became known as the "Mother of Cobol" and is one of the most important people in the history of computers

http://www.jamesshuggins.com/h/tek1/grace_hopper.htm

Slides courtesy of Dr. Alan McLeod

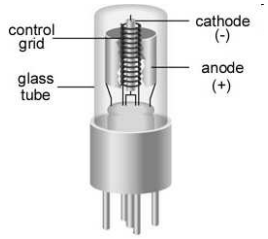
After the ENIAC

- Only this one ENIAC was ever built
- It was followed by the EDVAC in 1950
- The first commercial computer was the UNIVAC I
  - Delivered to the Bureau of the Census in the U.S. in 1951

Walter Cronkite at right…

Slides courtesy of Dr. Alan McLeod

Before Transistors…

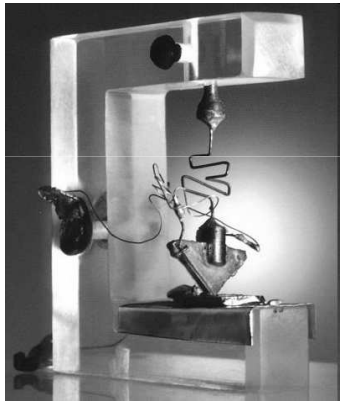


Vacuum tubes – large, lots of heat, lots of energy required …

Slides courtesy of Dr. Alan McLeod

Transistors

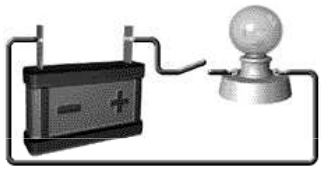
- First demonstrated in 1947, at Bell Labs by William Shockley



Slides courtesy of Dr. Alan McLeod

Transistors - Cont.

- Transistors work just like the following circuit:

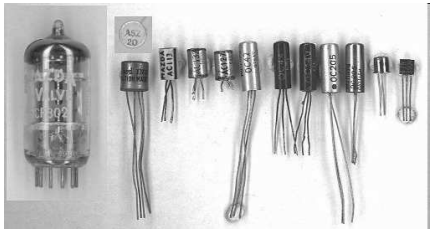


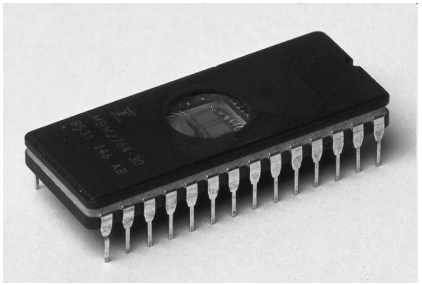
- Each wire in a computer has a signal that is either on or off for the duration of a single clock tick
  - A clock tick is a very short time!
  - Results in a 1 or 0 in binary

Slides courtesy of Dr. Alan McLeod

Transistors - Cont.

- Individual



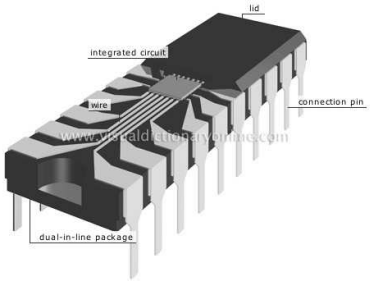


- Integrated Circuits
  - Started in1958

Slides courtesy of Dr. Alan McLeod

Integrated Circuits

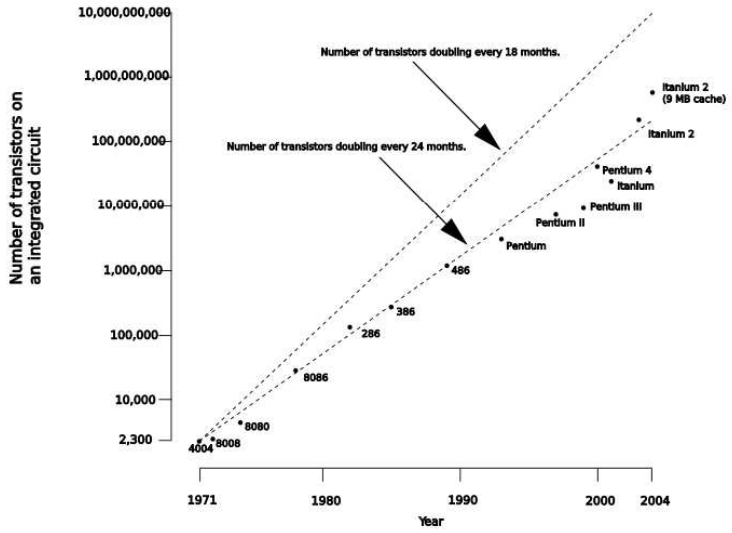
- An IC combines thousands, if not millions of transistors onto a single chip
  - Chips are flat crystals made of silicon
    - Silicon is a semiconductor material



Slides courtesy of Dr. Alan McLeod

Moore's "Law"

"The number of transistors in a processor will double every 24 months."



Slides courtesy of Dr. Alan McLeod

Moore's "Law" - Cont.

- First documented by Intel co-founder Gordon E. Moore in 1965
- But has the quality of software improved exponentially over the same time period?

Slides courtesy of Dr. Alan McLeod

Integrated Circuit Fabrication

See Intel Videos at:

`http://intelpr.feedroom.com/?fr_story=33c969acc3f25d0f702e56eb10b55c5692bd0f81&rf=bm`

`http://intelpr.feedroom.com/?fr_story=FEEDROOM195302&rf=bm`

Slides courtesy of Dr. Alan McLeod