Lecture 0

or “Syllabus day! (without a syllabus…)”

Welcome!

You are in CSCE 212. If you aren’t in 212, you can’t leave, because I will miss you.

/$ whoami

My name is William Hoskins. About to finish my 2nd year of the PhD program here. Received my undergraduate degree here a few years ago.

My research

Serious game design:

- Games with a primary goal other than entertainment
- I work on various games, but the ones that I am most interested in are Autism therapy games

What is this class?

Well, this class is a lot of things.

We will discuss the basics of what makes a computer work, and some of the history behind how we got here

We will also be working in MIPS

Detailed course objectives

- Describe the microstructure of a processor
- Describe how conventional machine instructions operate in conjunction with the components of a computer
- Demonstrate the ability to program a microprocessor in assembly language
- Classify and describe the operation of parallel computer architectures
- Evaluate the performance of computers

What the plan?

I will be structuring my class in the same way Dr. Wang will be teaching his. Our schedule is as follows:

- General Overview of Computer Architecture
- MIPS Instruction Set Architecture- Assembly Language Paradigm
- Floating Point Algorithms
- Performance
- Processor Design
- Memory Hierarchy
- Multicore and multiprocessor architectures

Grades

Midterm 1:	15%
Midterm 2:	15%
Final Exam:	20%
Homework/Quiz:	20%
Projects:	30%

Extra credit?

Throughout the semester there will be various extra credit opportunities.

One will be attending a codeathon I will be doing sometime in February (There might be 2 in Feb, not sure)

The other will be a Smash Bros tournament

Super Smash Credit

Using Smash Wii U, because it’s new

The rules are simple, 1v1 pairings, best of 3, single elimination (Unless someone wants to let us use their system then we can do a losers bracket)

First prize: 5 points on the final

Second prize: 5 points on a midterm

Third prize: Free homework

There’s a catch

I am entering too

If I win, no one gets any points

There’s a catch

I am entering too

If I win, no one gets any points

lol, jk

Questions before we learn?

Gordon Moore

Who is he?

(Hint: His net worth is $7.2 billion)

Gordon Moore

Gordon Moore is a co-founder of Intel

He is also the author of Moore’s Law

Moore’s Law

Moore’s law is the observation that the number of transistors on an integrated circuit doubles every 2 years

His original quote stated that he believed this would remain consistent for about 10 years (1965-1975)

Moore’s Law

Moore’s Law held true until about 2013 when growth slowed to doubling every 3 years

Levels of Code

Levels of Code

High-Level Languages:

- Level of abstraction
- Provides ease of use and portability

Assembly:

- Textual representation of instructions

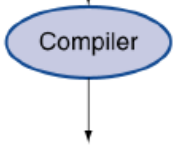
Hardware:

- Binary

High-level language program (in C)

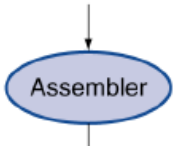
```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

f01-03-P374493



Assembly language program (for MIPS)

```
swap:
      muli $2, $5,4
      add  $2, $4,$2
      lw   $15, 0($2)
      lw   $16, 4($2)
      sw   $16, 0($2)
      sw   $15, 4($2)
      jr   $31
```



Binary machine language program (for MIPS)

```
00000000101000010000000000011000
00000000000011000000110000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

What are abstractions?

A higher level programming language has more abstraction. Basically abstraction makes code human readable

Abstraction helps us deal with complexity

- Hides low level details

Performance

Algorithms

- Determines number of operations executed

Programming language/Compiler/Architecture

- Determines # of machine instructions executed per operation

Processor and Memory

- Determines how fast instructions are executed

I/O System (including OS)

- Determines how fast I/O operations are executed

What’s inside a processor?

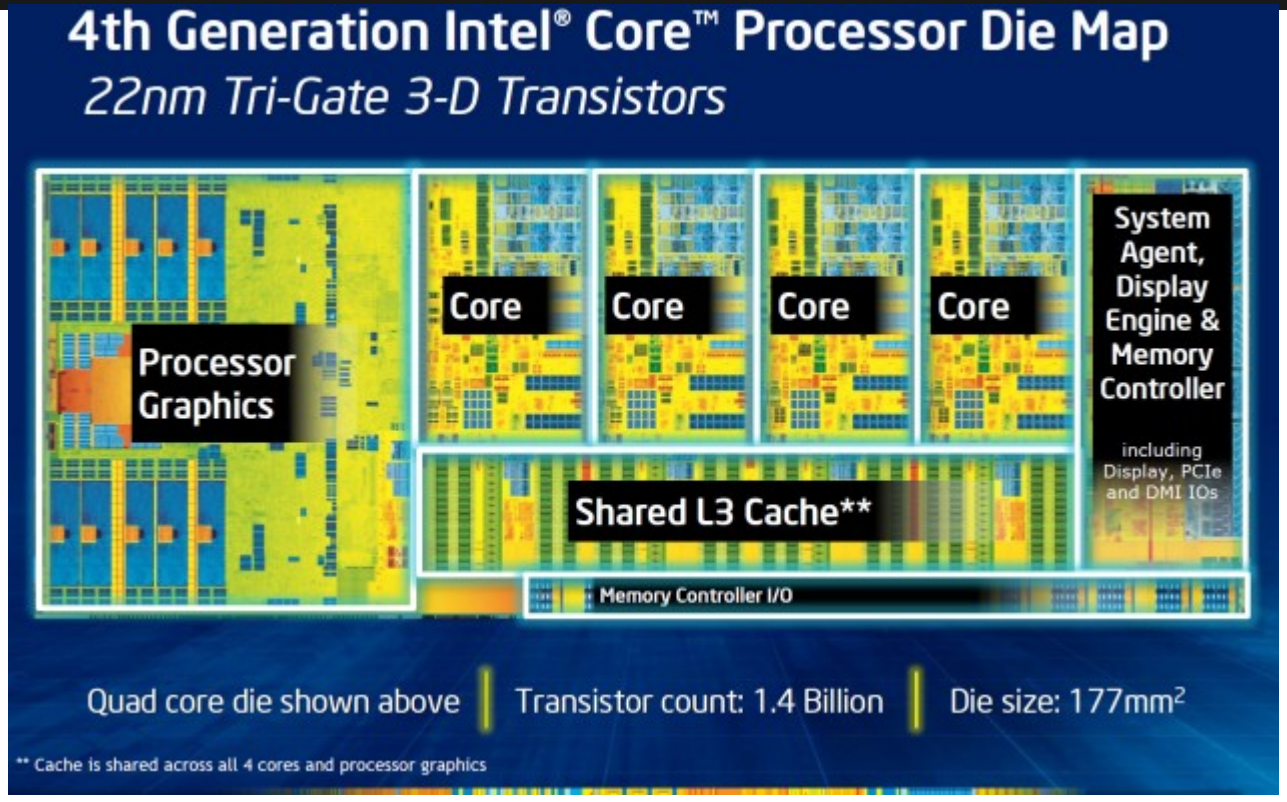
Data path: Part of CPU where data signals flow

Control unit: Guides data through data path

Cache memory: Small, fast memory

Other stuff that we will talk about more later...

i7 processor



Storage

While we have cache, we will mainly talk about 2 kinds of memory: Volatile main memory, and Non-volatile secondary memory

Storage

Volatile main memory (RAM): Loses data when power isn’t going to it. Very fast

Non-volatile secondary memory: Magnetic disk, flash memory, slow

Storage side note

There have been advancements in the non-volatile memory scene recently. SanDisk has something called ULLtraDIMM SSDs, which are Solid State Drives that slot into a RAM slot

Storage: RAM

RAM and Standard storage both evolve at a fast rate.

The increase in RAM capacity has been exponential, as well as the performance to cost ratio. This basically means you can get a lot of RAM for pretty cheap now

POWER WALL

In regards to CPU speed, we use Hz (Hertz)

Currently speeds range from around 2.9 GHz and 4.0 GHz

There is a reason that we haven’t gotten much faster than this

Power Wall

To make a processor faster we need to pull more power. The more power we pull the more heat the processor will produce. We can’t dissipate this heat very easily using fans so we don’t go any faster.

This is why we have overclocking