

# CS251: Intro to Computer Organization and Design

Graham Cooper

May 4th, 2015

## Guiding Principals

- Abstraction to simplify design
- Moores Law: Expect Rapid change in technology
  - IC resources doubles every 18-24 months
  - # of transistors can fit on a circuit board will double
- Improvement via Parallelism
- Improve Performance via Pipelining
- Improve Performance via Prediction

## Big Picture:

- Computer
- Control -i registers
- Datapath
- Processor
- Memory
- Input
- Output

## Instruction Set Architecture

To connect to the hardware you must speak its language (machine language/byte-code)

## Basic MIPS

\$s1: f \$s2: g \$s3: h \$s4: i \$s5: j \$s6 = TEMP \$s7 = TEMP

$$f = (g + h) - (i + j)$$

```
add $s6 , $s2 , $s3
add $s7 , $s4 , $s5
sub $s1 , $s6 , $s7
```

## Registers

Mips: \$0 does not write

## Instructions

- R-format: results and registers
- I-format: use immediate values as well
- J-format: used for branching or jumps

```
beq $s1 , $s2 , 15
```

This will jump 15 ahead if  $\$s1 == \$s2$ , also I-format with the 15

**Program Counter (PC)** This is the line number you are on and it will always increase by 4 bytes for each instruction.