It’s All (just) Bits

1) Numbers are bits
2) Text is bits
3) Formatted text is bits
4) Pictures are bits
5) Sound is bits
6) Programs
   (instructions on how to operate on bits)
   are also bits!
I CAME TO THE AUDIT OF YOUR NEW $150 MILLION SOFTWARE SYSTEM.

I RECOMMEND THAT YOU SCRAP THE ENTIRE THING.

WHAT?!! HOW COULD THE ENTIRE THING BE WORTHLESS?

WELL, YOUR NORMAL SOFTWARE SYSTEM WOULD BE A CLEVER COMBINATION OF ONES AND ZEROS.

YOURS IS ALL ONES.

THERE MUST BE SOME WAY TO TWEAK IT UNTIL IT WORKS.

MY COMPANY CAN SELL YOU ALL THE ZEROS YOU NEED, BUT YOU'LL HAVE TO ARRANGE THEM YOURSELF.

WHEN YOU HAVE A FEW MINUTES, I HAVE A LITTLE ASSIGNMENT FOR YOU.
Computer Architecture in 1 Slide

• A program is a sequence of **instructions**
• Both data and program sit in **memory** (RAM)
• At the heart of your computer is a **processor**, which simply executes instructions in order
• A processor has a few **registers** for storing intermediate results (like scratch paper)
• There are only a few possible instructions and a few registers; anything more complex is built up from there

R1  R2  R3  R4  R5  R6  R7  R8
Example Instructions

1. R1 ← 100
2. R2 ← 9
3. R3 ← R1+R2
4. R8 ← 218
5. R6 ← R8/2

Instructions can be encoded as bits:
need a code for each operation (e.g., +=1, −=2, /=3)
write down the operation code, the values,
and the register numbers

100  9  109  
R1  R2  R3  R4

109  218
R6  R7  R8
Memory

- But we have lots of data; it doesn’t all fit into the few registers. It sits in memory.
- Big idea: address it.
But we have lots of data; it doesn’t all fit into the few registers. It sits in memory.

Big idea: address it.

load into R1 the value in memory at address 153

(lots of) memory
multiplexer: chooses 1 bit out of 2
choosing 1 bit out of 4

Set selectors to: 00
choosing 1 bit out of 4

Set selectors to: 01 (bottom to top)
choosing 1 bit out of 4

Set selectors to: 10 (bottom to top)
choosing 1 bit out of 4

Set selectors to: 11 (bottom to top)
choosing 1 bit out of 8

Addresses:

000 001 010 011 100 101 110 111
How many layers of multiplexers?
Program using memory

1. load into R1 the value in memory at address 153

(lots of) memory
Program using memory

1. load into R1 the value in memory at address 153
2. R1 ← R1 – 32

(lots of) memory
Program using memory

1. load into R1 the value in memory at address 153
2. R1 ← R1 – 32
3. store the value in R1 into memory at address 153
Program using memory

1. load into R1 the value in memory at address 153
2. R1 ← R1 – 32
3. store the value in R1 into memory at address 153