lecture 8

MIPS assembly language 1

- what is an assembly language?
- addressing and Memory
- instruction formats (R, I, J)

February 3, 2016

ASCII: 8 bit (one byte) code
In fact, it only uses 7 of the 8 bits.

Unicode is a 16 bit code which includes characters for most written natural languages. [Correction: It is more complicated than that.]

MIPS R2000 CPU (1985)

"Reduced Instruction Set Computer" (RISC)

MIPS is not so different from today’s ARM processors e.g. in your cell phone or tablet.

human readable &
readable machine
language

assembly language

machine code

C, Fortran, Java, etc

human writeable &
readable machine
code (in ASCII)

...although what we mean by “machine” here is subtle. e.g. Java Virtual Machine.

MARS simulator

http://courses.missouristate.edu/KenVollmar/MARS/index.htm

Edit mode

Execute mode

Addressing in MIPS

1 word = 4 bytes
= 32 bits
Each MIPS instruction is 32 bits.

Examples of MIPS instructions

```
add $10, $17, $18
```

Arithmetic and logic instructions

```
add $10, $17, $18
sub $19, $15, $19
and $17, $17, $16
or $10, $17, $18
nor $10, $17, $20
```

Memory transfer instructions

```
lw $16, 40($17)
```

Suppose the values of x and y are assigned to registers (say $18 and $20) but the value of z is stored in Memory. Then we need to bring z into a register (say $16) in order to do the addition.

```
lw $16, 40($17)
add $18, $20, $16
```

```
sw $16, 40($17)
```

```
lw $16, 40($17)
add $18, $20, $16
```

```
lw $16, 40($17)
add $18, $20, $16
```

```
lw $16, 40($17)
add $18, $20, $16
```

```
lw $16, 40($17)
add $18, $20, $16
```

```
lw $16, 40($17)
add $18, $20, $16
```

```
lw $16, 40($17)
add $18, $20, $16
```

```
lw $16, 40($17)
add $18, $20, $16
```
If you want to store the result of the sum in a Memory address (you might need it later) then you must use a register.

In MIPS, you cannot take the result of a summation from the ALU and put it directly in Memory.

$sw \quad \$16, \ 40 (\$17)$

Program Counter

The Program Counter (PC) register specifies the Memory address of the instruction that is currently being executed.

The default is that the computer advances to the next instruction. But there are also branches allowed (next slide)....

Branching Instructions

How is conditional branching done in MIPS?

e.g. "branch equals" (beq)

$\text{beg } \$17, \$18, \text{ Exit1}$

$a = g + h$

MIPS instructions (machine code)

- always 32 bits (one word)
- relatively few (RISC: "reduced instruction set computer")
MIPS instruction formats

R format instructions

I format instructions ("immediate")

J format instructions ("jump")

Announcements

A1
- get started by learning basics of Logisim (construct simple circuits shown in class, e.g. left shift register)
- specification with START and DONE is the last thing you should be concerned with

Quiz 2 is Monday (lectures 3-6)