MIPS data path and control 1

- single cycle model
- fetch vs execute
- examples: add, lw, sw, bne, j

February 22, 2016
Q: Did A2 take you more than 10 hours?

Do you receive notifications of news items on mycourses?

Are you subscribed to mycourses Discussion boards (for Forum Assignments)?

Facebook is great, but not everyone uses it, and I'm not posting there.
Did you use breakpoints to examine registers and Memory as your program runs?

If not, then you are programming BLIND.
See end of lecture 10 slides.

[ADDED Feb 21]

max length string

array of 5 string references

25 words for the strings

the prompts

Checking this may be useful
Digital Logic (lectures 1-7)
- number reps
- combinational logic
- sequential logic

MIPS Assembly (lectures 8-12)
- instructions
- directives, syscalls
- functions
- floats, kernel

what's the Connection?
Instruction Fetch

where is the next instruction?

where is the current instruction?

various parts of instruction (depends on R, I, J format)

What is the current instruction?
What is the current instruction and where is it in memory?

Naive idea of how this works

We will see how this works later in course
What is the current instruction?

Suppose it is:

```
add  $16, $17, $18
```

<table>
<thead>
<tr>
<th>op</th>
<th>rs</th>
<th>rt</th>
<th>rd</th>
<th>shamt</th>
<th>funct</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
add $16, $17, $18

Source

Destination
Single Cycle Model

Each instruction takes one clock cycle!
\[ \text{lw $16, 40($17$)} \]

<table>
<thead>
<tr>
<th>op</th>
<th>rs</th>
<th>rt</th>
<th>immediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>5</td>
<td>16</td>
</tr>
</tbody>
</table>

Binary: 010011 10001 10000 00000000000101000
lw $16, 40($17)
sw $16, 40($17)
bne $18$, $17$, Exit!

<table>
<thead>
<tr>
<th>op</th>
<th>rs</th>
<th>rt</th>
<th>immediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>5</td>
<td>16</td>
</tr>
</tbody>
</table>

000101 10010 10001 offset

(number of words from current instruction)
bne $18, $17, Exit1
```
test 1 :
    addi $s0, $0, 0
    bne $s0, $s1, test 1
    bne $s3, $s4, test 2
    addi $s5, $0, 0

test 2 :

What are the value in the immediate field of the bne instructions?

<table>
<thead>
<tr>
<th>op</th>
<th>rs</th>
<th>rt</th>
<th>immediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>5</td>
<td>16</td>
</tr>
</tbody>
</table>
```

```
test 1  -2  0xffffffff
        1  0x000001
```
J format (jump)

Exit 2

<table>
<thead>
<tr>
<th>op</th>
<th>address</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>26</td>
</tr>
</tbody>
</table>

0 0 0 0 1 0

31, 30, 29, 28

PC

always

0 0