In-Class Activity 09-2

1 Activity 1 - R Format Instructions

In MIPS, there are different instruction formats. One of the formats is “R Format” - and it includes instructions which read from two registers, perform some simple ALU operation, and then write back to a register.

As a group, come up with all of the R-format instructions that you can think of.

1.1 I Format Instructions

In MIPS, an “I format” instruction is one that names two different registers (this might be for read or write), and includes a 16-bit constant value (the “immediate” value) as well.

As a group, come up with all of the I-format instructions that you can think of.

2 Activity 2 - Some I-Format Instructions

Write out pseudocode for each of the following I-format instructions. Be as specific as you can; include steps like “read $s0 from the register file” and “sign-extend the 16-bit immediate field.”

(If you don’t know all of the steps, that’s OK - I’ll show them soon. But do as much as you can!)

Once we understand what these instructions do in detail, we’ll look at how the hardware accomplishes them.

```
addi $s2, $s0,10
addi $t3, $t3,-1
lw $s1, 12($s4)
sw $s1, 12($s4)
beq $t0,$zero, SOME_LABEL
```

3 Activity 3 - lw vs. sw

Using the picture in the slides, identify all of the differences between lw and sw in the datapath. (It wouldn’t hurt to also note all of the things that are the same - there are a lot of them!)
4 Activity 4 - Fields

How many bits do we need to encode an R-format instruction (ignore opcode for now, we'll deal with that soon)?

How many bits do we need to encode an I-format instruction? (Assume that the immediate field is always exactly 16 bits.)

What parts of the instruction might be common between the R-format and I-format? (That is, what parts might be wired up to the same places in the CPU?)