

For this assignment read the ARM Architecture Reference Manual linked to <http://www.ece.lsu.edu/ee4720/reference.html>. This assignment asks about the ARM A32 instruction set.

Problem 1: Show the encoding of the ARM A32 instruction that is most similar to MIPS instruction `add r1, r2, r3`.

Problem 2: ARM instructions can shift one of its source operands, something MIPS cannot. With this feature the code below can be executed with a single ARM add instruction. Show the encoding of such an ARM A32 add instruction.

```
sll r1, r2, 12
add r1, r4, r1
```

Problem 3: So, the ARM add instructions can shift one of its operands, something that MIPS would need two instructions to do. Since we have been working with MIPS for so long it would be natural for us to get protective of MIPS and defensive or jealous when hearing about wonderful features of other ISAs that MIPS doesn't have. To relieve these negative emotions lets add operand shifting to MIPS with a new `addsc` instruction. The `addsc` instruction will use MIPS' `sa` field to specify a shift amount. So instead of, for example, the following two instructions:

```
sll r1, r2, 12
add r1, r4, r1
```

We could use just

```
addsc r1, r4, r2, 12
```

where the "12" indicates that the value in `r2` should be shifted by 12 before the addition.

Modify our five-stage MIPS implementation so that it can implement this instruction. (See below for diagrams.)

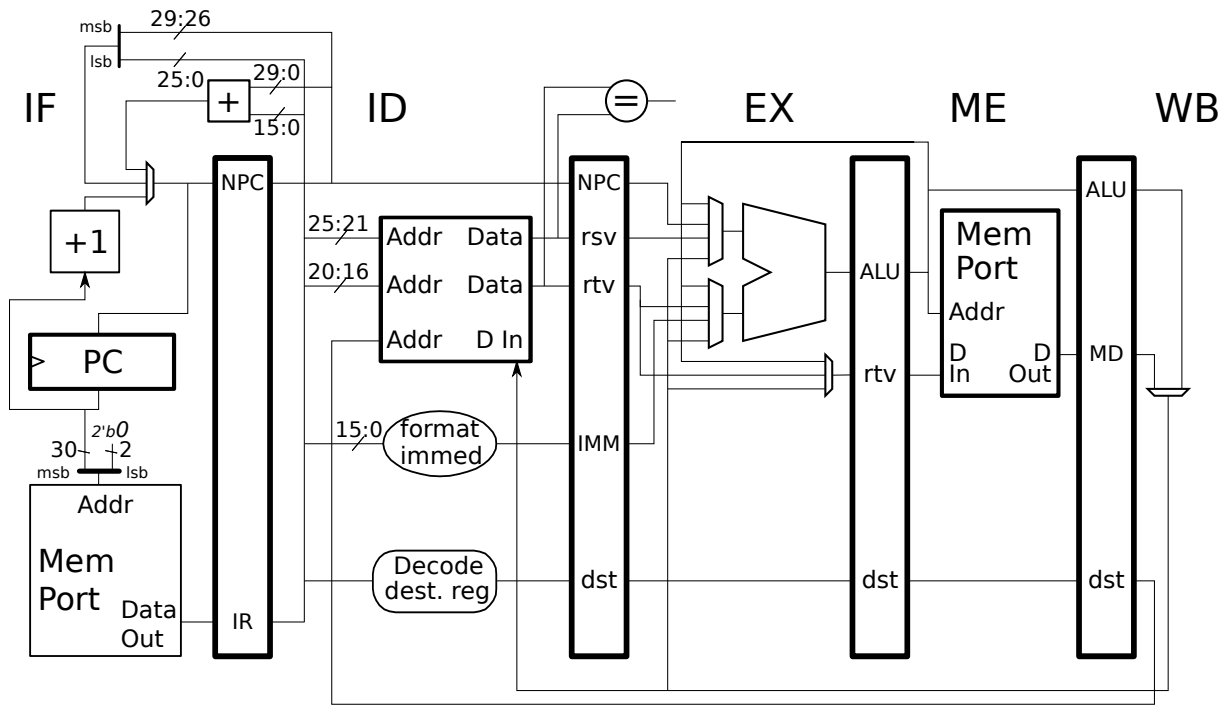
- The `addsc` should execute without a stall.
- Don't break existing instructions.
- Don't increase the critical path by more than a tiny amount.
- Keep an eye on cost.

Assume that both the ALU and shift unit take most of the clock period. **This means if the ALU and shifter are in the same stage and output of the shifter is connected to the ALU, the critical path will be doubled. (Of course, doubling the critical path would be disastrous for performance.)**

There are several ways to solve this, one possibility includes adding a sixth stage, another possibility uses a plain adder (not a full ALU) in the EX stage.

Add hardware to the implementation below. Source files for the diagram are at: <http://www.ece.lsu.edu/ee4720/2013/mpipei3.pdf>,

<http://www.ece.lsu.edu/ee4720/2013/mpipei3.eps>,
<http://www.ece.lsu.edu/ee4720/2013/mpipei3.svg>. The svg file can be edited using Inkscape.



To see how a shift unit can be added to MIPS see Fall 2010 Homework 3.