# Homework VII

Tom Kelliher, CS 220

Due Dec. 8, 2011

Remember that the due date is a final due date — late assignments will not be accepted, except for emergencies. Your solutions to these problems should be collated in the order listed below. Make sure that you show work, that you write your solutions in such a way that I will be able to understand how you arrived at them, and that your solutions are legible. Please leave enough whitespace on your solutions so that I can write comments. *This assignment is due at 5:00 pm on the 8th.*

1. We wish to add the instruction `jr` (jump register) to the single-cycle datapath. Add any necessary datapaths and control signals to the single-cycle datapath figure on page 3 of this assignment and show the necessary additions to the figure on page 4 of this assignment. (Make copies of the figures.)

2. We wish to add the instruction `lui` (load upper immediate) to the single-cycle datapath. Add any necessary datapaths and control signals to the single-cycle datapath figure on page 3 of this assignment and show the necessary additions to the figure on page 4 of this assignment. (Make copies of the figures.)

3. (5 points) Consider a variant of the `lw` (load word) instruction, `lwi` (load word and increment), in which the base register is incremented by one as a side effect. Explain why it is not possible to modify the single-cycle datapath to implement `lwi`.

4. (5 points) If the time for an ALU operation can be shortened by 25% (compared to the description in Figure 4.26 on page 333);

   (a) Will it affect the speedup obtained from pipelining? If yes, by how much? Otherwise, why?

   (b) What if the ALU operation now takes 25% more time?

5. A computer architect needs to design the pipeline of a new microprocessor. She has an example workload program core with $10^6$ instructions. Each instruction takes 100 ps. to finish.

   (a) How long does it take to execute this program core on a nonpipelined processor?

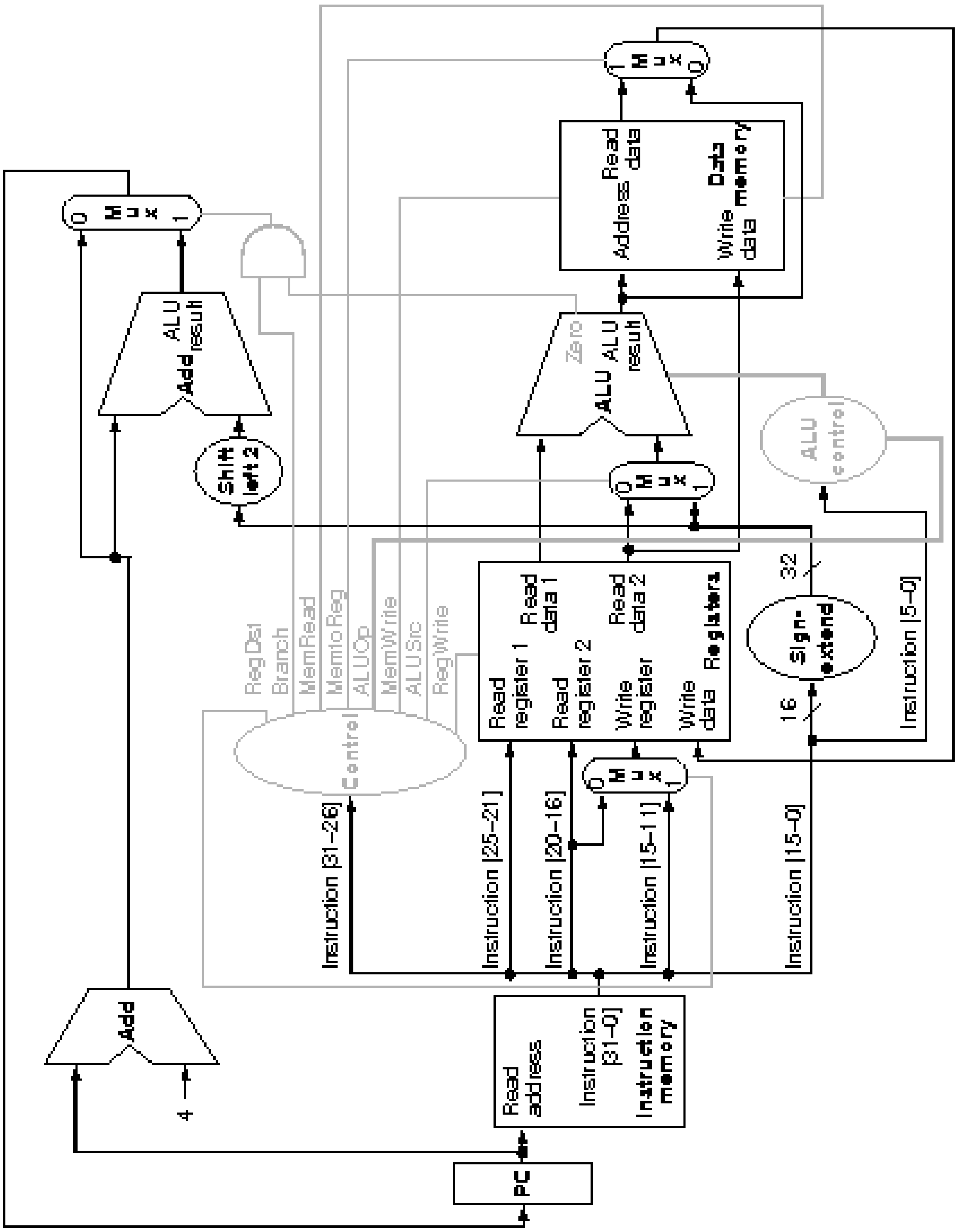
   (b) The current state-of-the-art microprocessor has about 20 pipeline stages. Assume it is perfectly pipelined. How much speedup will it achieve compared to the nonpipelined processor.

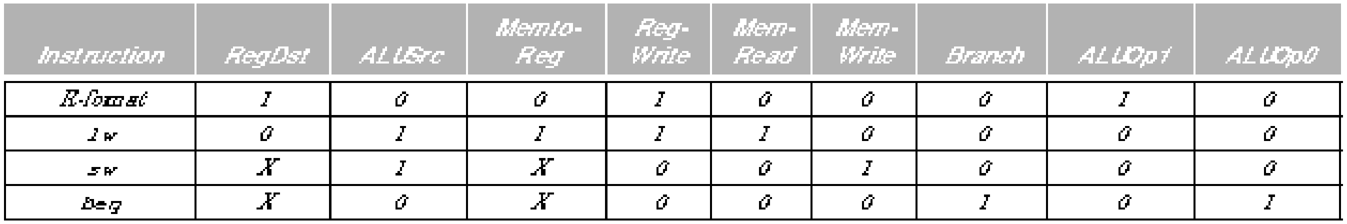
   (c) Real pipelining isn't perfect, since implementing pipelining introduces some overhead per pipeline stage. Will this overhead affect instruction latency, instruction throughput, or both?

6. Identify *all* of the data dependencies in the following code. Which dependencies are data hazards that will be resolved by forwarding? Which dependencies are data hazards that will cause a stall?

```
add $3, $4, $2
sub $5, $3, $1
lw $6, 200($3)
add $7, $3, $6
```

Unless otherwise stated, each problem is worth 10 points.


PC
Read address
Instruction [31–0]
Instruction memory
Add
4
Add
ALU result
Shift left 2
Mux
0
1
Control
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Mux
0
1
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
32
Sign-extend
Instruction [5–0]
Mux
0
1
ALU control
ALU
Zero
ALU result
Address
Write data
Read data
Data memory
Mux
1
0



| Instruction | RegDst | ALUSrc | MemtoReg | RegWrite | MemRead | MemWrite | Branch | ALUOp1 | ALUOp0 |
|---|---|---|---|---|---|---|---|---|---|
| R-format | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| lw | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| sw | X | 1 | X | 0 | 0 | 1 | 0 | 0 | 0 |
| beq | X | 0 | X | 0 | 0 | 0 | 1 | 0 | 1 |