CSCI 524 – Computer Architecture
Homework 4

Due: October 6, 2016

Alexander Powell

1. **4.3**

   - **4.3.1. What is the clock cycle time with and without this improvement?**
     The original critical path involves I-mem, Regs, Mux, ALU, D-Mem and Mux. Adding this up, we get
     $$400 + 200 + 30 + 120 + 350 + 30 = 1130\text{ps}$$
     With the added multiplier, the new latency will be
     $$1130 + 300 = 1430ps$$
   - **4.3.2. What is the speedup achieved by adding this improvement?**
     There are 5% fewer instructions, but the cycle time is 1430ps. So, if we say a program takes n instructions to complete, then we can calculate the speedup by taking
     $$\frac{n \text{ instructions } \times 1130}{0.95 \times n \times 1430} \approx 0.8318$$
     Notice that this value is less than one; therefore this change actually decreases the speed by some amount.
   - **4.3.3. Compare the cost/performance ratio with and without this improvement.**
     First we need to calculate the total cost before and after the improvement. There are 2 Add and 2 Mux units, so the total cost originally is
     $$1000 + 30 \times 2 + 10 \times 3 + 100 + 200 + 2000 + 500 = 3890$$
     The cost after the improvement has been made is
     $$3890 + 600 = 4490$$
     Let's say there are baseline n instructions. The the cost/performance ratio can be written
     $$\frac{3890}{\frac{1}{n \times 1130}} = 3890 \times 1130 \times n = 4395700n$$
     Similarly, the improved version can be written
     $$4490 \times 0.95 \times 1430 \times n = 6099665n$$
     So, you can see that the cost to performance ratio increases when the multiplier is added. Therefore, this is not an improvement.

2. **4.4**

- **4.4.1. If the only thing we need to do in a processor is fetch consecutive instructions, what would the cycle time be?**
  The cycle time would only be as long as the instruction memory latency since they are on parallel paths. So, the cycle time is 200ps.
- **4.4.2. Consider a datapath like 4.11, but for a processor that only has one type of instruction: unconditional PC-relative branch. What is the cycle time?**
  Unconditional PC-relative branches rely on fetching from instruction memory, adding for the program counter, mux, sign-extension, and shift-left-2. Therefore, the cycle time can be calculated as $200 + 70 + 20 + 15 + 10 = 315$ps.
- **4.4.3. What is the cycle time for a conditional PC-relative branch?**
  Conditional PC-relative branches rely on fetching from instruction memory, the ALU, Regs, and Mux. Therefore, the cycle time can be calculated as $200 + 90 + 20 + 90 + 20 = 420$ps.
- **4.4.4. Which kinds of instructions require this resource?**
  Only PC-relative branches require the shift-left-2 resource.
- **4.4.5. For which kind of instructions (if any) is this resource on the critical path?**
  Only PC-relative unconditional branches have this resource on the critical path.
- **4.4.6. Assuming that we only support BEQ and ADD instructions, discuss how changes in the given latency of this resource affect the cycle time of the processor.**
  Since BEQ has a longer critical path than ADD, it will dictate the overall cycle time. Also, shift-left-2 is not on the critical path. The combined latency of Regs, Mux, and ALU is $90 + 90 + 20 = 200$ps and the combined latency of sign-extend, shift-left-2 and Add is $70 + 10 + 15 = 95$ps. Therefore, the latency for shift-left-2 has to increase by at least 105ps for it to affect the clock cycle time.

3. **Single Cycle Datapath Design**

(a) **Give the settings (making maximum use of dont cares) for the control signals for the single cycle datapath shown below when executing a lw and a slt instruction.**

Instr Name	RegDst	ALUSrc	MemReg	RegWr	MemRd	MemWr	Branch	ALUOp	Jump
lw	0	1	1	1	1	0	0	00	0
slt	1	0	0	1	0	0	0	00	0

(b) **What is the fastest clock speed for the single cycle machine when executing a lw instruction?**
A lw instruction requires PC access, instruction memory, register access, 3 muxes, ALU access, data memory, and another register access. This gives

$$200 + 400 + 200 + 30 \times 3 + 150 + 350 + 200 = 1590\text{ps}$$

(c) **What is the clock cycle time if the only type of instructions are ALU instructions?**
If the only instructions executed are ALU instructions, we will need to fetch from instruction memory, Regs, Mux, ALU, Mux, and Regs? That is

$$400 + 200 + 30 + 150 + 30 = 1010\text{ps}$$

(d) **What is the clock cycle time if we support SUB, BEQ, LW and SW instructions?**
If SUB, BEQ, LW and SW instructions are executed, we need I-mem, Reg, Mux, ALU, D-mem, Mux, and Regs. That is

$$400 + 200 + 30 + 150 + 350 + 30 + 200 = 1360\text{ps}$$

(e) **Now let us assume the ALU control delay is 50ps. Which control signal in the data path is the most critical to generate quickly? How much time does the control unit have to generate it to avoid being on the critical path?**

The Branch signal is the most critical signal to generate quickly in the data path. This requires the 2 muxes, reading a register file, and the ALU access, which is $50 + 30 \times 2 + 250 = 350$ps.