Homework 4 – Datapath Intro, Instruction Encoding, Single Cycle CPU

Out: 3.8.21
Due: 3.17.21

1. [Datapath]
Figure B8.8 in the textbook (also on the 'Reading From the Register File' slide) illustrates the implementation of the register file for the MIPS datapath. Pretend that a new register file is to be built, but that there are only two registers and only one read port, and that each register has only 2 bits of data. Redraw this figure so that ever wire in your diagram corresponds to only 1 bit of data. Redraw the registers using D flip-flops, and use multiplexors.

2. [Datapath]
Given the circuit shown in the slide titled 'MUX Controlled R-to-R Transfer', fill in the rest of the table.

	T0	T1	T2	T3	T4	T5	T6
IN_R0	3	5	7	9	11	13	15
IN_R1	0	2	4	6	8	10	12
OUT_R0							
OUT_R1							
K1	0	0	1	0	0	1	1
K2	1	0	0	0	1	0	1
ENABLE (LOAD)							
IN_R2							
OUT_R2							

3. [Instruction Encoding/Decoding]
What is the machine language encoding for the following instructions? If you do not have enough information, explain what is needed.
a) lw $7,48($12)
b) jal procedure
c) addi $21,$15,-1000

4. [Instruction Encoding/Decoding]
   What is the machine language encoding of the following branch instruction (only)?

```
done: lw     $7,48($12)
        sll    $10,$5,3
        addi   $21,$15,-1000
        beq    $10,$11,done
```

5. [Instruction Encoding/Decoding]
   What are the mnemonic equivalents of the following machine language instructions?
   a) 0x0810000a
   b) 0x8d4c0114

6. [Converting MIPS AL to Bare MIPS]
   In class we showed how the MIPS assembly language instruction BLT could be emulated using the bare MIPS instructions SLT and BNE. Show the emulations for the following instruction (i.e., implement using SLT, BNE, and/or BEQ).

   BGT $4,$5,17

7. [Converting MIPS AL to Bare MIPS]
   Bare MIPS only allows 16 bit immediates. Show how you would emulate the MIPS assembly language instruction using only bare MIPS instructions.

   OR $8,$9,0xEFEFEFEF

8. [Converting MIPS AL to Bare MIPS]
   Bare MIPS only allows 16 bit displacement. Show how you would emulate the MIPS assembly language instruction using only bare MIPS instructions.

   LW $9,0xEFEFEFEF($10)

9. [Single Cycle CPU]
   Consider the following instruction: *AND Rd, Rs, Rt*
   With the following interpretation: *Reg[Rd] = Reg[Rs] AND Reg[Rt]*
   1. What are the values of control signals generated by the single-cycle main control unit?
   2. Which resources (blocks) perform a useful function for this instruction?
   3. Which resources (blocks) produce outputs, but their outputs are not used for this instruction? Which resources produce no outputs for this instruction?

10. [Single Cycle CPU]
    The basic single-cycle MIPS implementation can only implement some instructions. New instructions can be added to an existing Instruction Set Architecture (ISA), but the decision whether or not to do that depends, among other things, on the cost and complexity the proposed addition introduces into the processor datapath and control.

The following problems in this exercise refer to the new instruction:
Instruction: *LWI Rt, Rd(Rs)*
Interpretation: *Reg[Rt] = Mem[Reg[Rd]+Reg[Rs]}*

1. Which existing blocks (if any) can be used for this instruction?
2. Which new functional blocks (if any) do we need for this instruction?
3. What new signals do we need (if any) from the control unit to support this instruction?

11. [Single Cycle CPU]

When processor designers consider a possible improvement to the processor datapath, the decision usually depends on the cost/performance trade-off. In the following three problems, assume that we are starting with a basic single-cycle MIPS datapath, where I-Mem, Add, Mux, ALU, Regs, D-Mem, and Control blocks have latencies of 400 ps, 100 ps, 30 ps, 120 ps, 200 ps, 350 ps, and 100 ps, respectively, and cost of 100, 30, 10, 100, 200, 2000, and 500, respectively.

Consider the addition of a multiplier to the ALU. This addition will add 300 ps to the latency of the ALU and will add a cost of 600 to the ALU. The result will be 5% fewer instructions executed since we will no longer need to emulate the MUL instruction.

What is the clock cycle time with and without this improvement?

12. [Single Cycle CPU]

Problems in this exercise assume that logic blocks needed to implement a processor's datapath have the following latencies:

I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2
200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps

1. If the only thing we need to do in a processor is fetch consecutive instructions, what would the cycle time be?
2. Consider a single-cycle MIPS datapath with branch support, but for a processor that only has one type of instruction: unconditional PC-relative branch. What would the cycle time be for this datapath?
3. Repeat (2), but this time we need to support only conditional PC-relative branches.

The following three problems refer to the datapath element Shift-left-2:

4. Which kinds of instructions require this resource?
5. For which kinds of instructions (if any) is this resource on the critical path?
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. Assume that the latencies of other resources do not change.