Cpr E 381 Homework 7

1. Problem 5.2 Describe the effect that a single stuck-at-0 fault (i.e., regardless of what it should be, the signal is always 0) would have for the signals shown below, in the single-cycle datapath in Figure 5.17 on page 307. Which instructions, if any, will not work correctly? Explain why.
    a. RegWrite = 0
    b. ALUop0 = 0
    c. ALUop1 = 0
    d. Branch = 0
    e. MemRead = 0
    f. MemWrite = 0

2. Problem 5.3 For the same signals in the previous problem, evaluate a stuck-at-1 fault (signal is always a 1). Which instructions, if any, will not work correctly? Explain why.

3. Problem 5.4 Suppose we have a floating-point unit that requires 400 ps for a floating-point add and 600 ps for a floating-point multiply, not including the time to get the instruction or read and write any registers, which take the same as for an integer instruction. Use the functional unit times from the example on page 315. In these exercises, we will find the performance ratio between an implementation in which the clock cycle is different for each instruction class and an implementation in which all instructions have the same clock cycle time. Assume the following:

    - All loads take the same time and comprise 30% of the instructions.
    - All stores take the same time and comprise 15% of the instructions.
    - R-format instructions comprise 25% of the mix.
    - Branches comprise 10% of the instructions, while jumps comprise 5%.
    - FP add and subract take the same time and together total 5% of the instructions.
    - FP multiply and divide take the same time and together total 10% of the instructions.
    - Find the time for the FP operations

4. Problem 5.5 For the datapath and instruction mix in Exercise 5.4 (previous problem), find the time for the processor with a single clock cycle length equal to the longest instruction.

5. Problem 5.6 For the datapath and instruction mix in Exercise 5.4, find the time for the processor with a varying length clock.

6. Problem 5.8 We wish to add the instruction jr (jump register) to the single-cycle datapath described in this chapter. Add any necessary datapaths and control signals to the single-cycle datapath of Figure 5.17 on page 307 and show the necessary additions to Figure 5.18 on page 308. You can photocopy these figures to make it faster to show the additions.

7. Problem 5.10 As in the previous problem, show the necessary additions to incorporate the lui (load upper immediate) instruction, which is described in Section 2.9.

8. Problem 5.14 MIPS choses to simplify the structure of its instructions. The way we implement complex instructions through the use of MIPS instructions is to decompose such complex instructions into multiple simpler MIPS ones. Show how MIPS can implement the instruction swap $rs, $rt, which swaps the contents of the registers $rs and $rt. Consider the case in which there is an available register that may be destroyed as well as the case in which no such register exists.

   If the implementation of this instruction in hardware will increase the clock period of a single-cycle implementation by 10%, what percentage of swap operations in the instruction mix would recommend implementing it in hardware?

9. Problem 5.23 Explain why it is not possible to implement the swap instruction in the previous problem within the single-cycle datapath without modifying the register file.

10. Problem 5.26 Consider the following idea: Let's modify the instruction set architecture and remove the ability to specify an offset for memory access instructions. Specifically, all load-store instructions with nonzero offsets would become pseudo- instructions and would be implemented using two instructions. For example:

    addi $at, $t1, 104 # add the offset to a temporary
    lw $t0, $at # new way of doing lw $t0, 104 ($t1)

    What changes would you make to the single-cycle datapath and control if this simplified architecture were to be used?