ICS 233, Term 072

Computer Architecture & Assembly Language

HW# 6

Q.1. Describe the effect that a single stuck-at-0 fault (i.e., the signal is always 0 regardless of what it should be) would have for the signals shown below, in the singlecycle Datapath. Which instructions, if any, will not work correctly? Explain why. Consider each of the following faults separately:

(i) RegWrite = 0

(ii) RegDst = 0

(iii) ALUSrc = 0

(iv) MemtoReg = 0

Q.2. Consider the single-cycle datapath. A friend is proposing to modify this single-cycle datapath by eliminating the control signal MemtoReg. The multiplexor that has MemtoReg as an input will instead use either the ALUSrc or the MemRead control signal. Will your friend's modification work? Can one of the two signal (MemRead and ALUSrc) substitute for the other? Explain.

Q.3. We wish to add the following instructions to the single-cycle datapath. Add any necessary datapath and control signals needed for the implementation of these instructions. Show only the modified and added components to the datapath. Show the values of the control signals to control the execution of each instruction.

(i) sll

(ii) sllv

(iii) lui

(iv) bltz

(v) bgez

(vi) slti

(vii) jr

Q.4. We want to compare the performance of a **single-cycle CPU design** with a **multicycle CPU**. Suppose we add the multiply and divide instructions. The operation times are as follows:

Instruction memory access time = 190 ps, Data memory access time = 190 ps

Register file read access time = 150 ps, Register file write access = 150 ps
ALU delay for basic instructions = 190 ps, Delay for multiply or divide = 550 ps

Ignore the other delays in the multiplexers, control unit, sign-extension, etc.

Assume the following instruction mix: 30% ALU, 15% multiply & divide, 30% load & store, 15% branch, and 10% jump.

(i) What is the total delay for each instruction class and the clock cycle for the single-cycle CPU design?

(ii) Assume we fix the clock cycle to 200 ps for a multi-cycle CPU, what is the CPI for each instruction class and the speedup over a fixed-length clock cycle? Note that this implies that multiply and divide operations will be performed in multiple cycles.