ECE341 Homework No. 6

Due date: 11/19/2014

Problem No. 1 (6 points)

Consider the 5-stage RISC processor discussed in class. During stage-1 of instruction processing, a memory Read operation is started to fetch an instruction at location 0x34000. However, as the instruction is not found in the cache, the Read operation is delayed, and the MFC signal does not become active until the fourth clock cycle. Assume that the delay is handled as described in Section 5.6.2. Show the value of PC_enable control signal and the contents of PC during each of the four clock cycles of stage-1, and also during stage-2 of instruction processing.

Problem No. 2 (12 points)

Consider the following sequence of instructions being processed on the 5-stage RISC processor discussed in class:

Load R4, #200(R3)

Or R2, R4, R6

Store R2, #600(R3)

Call_Register R7

Write down the values of following control signals for **each of the four** instructions:

(a) *PC_enable* during stage 3 of instruction processing **(2 points)**
(b) *Mem_read* and *Y_select* during stage 4 of instruction processing **(4 points)**
(c) *RF_write* and *C_select* during stage 5 of instruction processing **(6 points)**

Note: Refer to Figure 5.12 for instruction encoding formats and Figures 5.18, 5.19 and 5.20 to understand the function of each control signal

Problem No. 3 (14 points)

(a) **(3 points)** What is the key idea behind pipelining? How does pipelining improve performance?
(b) **(5 points)** A program consisting of 150 instructions is executed on a 5-stage processor. How many cycles would be required to complete the program, (i) without pipelining, (ii) with pipelining? Assume *ideal* overlap in case of pipelining.
(c) **(3 points)** Referring to part(b) of the problem, calculate the speedup obtained with pipelining.
(d) **(3 points)** What is the upper bound on the speedup attainable by pipelining a 5-stage processor? Is the speedup obtained in part(c) less than the upper bound? If yes, why?

Problem No. 4 (19 points)

Consider the following sequence of instructions being processed on a **pipelined** 5-stage RISC processor discussed in class:

Load R5, #200(R2)

Add R3, R5, R6

Subtract R4, R3, R7

And R7, R3, R5

(a) **(5 points)** Identify all the data dependencies in the above instruction sequence. For each dependency, indicate the two instructions and the register that causes the dependency.

(b) **(10 points)** Assume that **the processor does not use operand forwarding** and simply stalls the pipeline to mitigate data hazards. Draw a diagram similar to Figure 6.1 that represents the flow of instructions through the pipeline during each clock cycle. Account for all the pipeline stalls caused by data hazards.

(c) **(4 points)** Calculate the speedup in execution time as compared to a 5-stage processor that does not use pipelining.

Problem No. 5 (14 points)

Refer to the pipelined processor and the instruction sequence discussed in Problem 4. Now, assume that **the pipeline uses operand forwarding** by means of forwarding paths to the ALU from registers RY and RZ of the datapath.

(a) **(10 points)** Draw a diagram similar to Figure 6.1 that represents the flow of instructions through the pipeline during each clock cycle. Indicate operand forwarding by arrows, similar to Figure 6.4.

(b) **(4 points)** Calculate the speedup in execution time by using operand forwarding as compared to the pipeline of Problem 4 which does not support operand forwarding.