

Course: ENCM 369

Lab Section: B03

Lab 9

Student Name: Mitchell Sawatzky

Date Submitted: March 22, 2016

Exercise A

Part I

- 1.) $T_C \geq t_{pcq} + t_{I-mem} + t_{setup} = 30ps + 330ps + 20ps = 380ps$
- 2.) $3.2GHz = 312ps$; $380 - 312ps = 68ps$
Therefor the t_{pd} of the Instruction Memory must be reduced by 68ps to have a maximum of 262ps.

Part II

- 1) The critical path is through ALUSrcE's mux into the ALU. $t_{pd} = t_{mux} + t_{ALU} = 35ps + 150ps = 185ps$
- 2) A clock of 3.2GHz corresponds to 312ps, which is more than 50ps above the t_{pd} of this stage already. ($50ps = t_{pcq} + t_{setup}$)

Part III

- 1) $t_{pd} = t_{pcq} + t_{adder} + t_{mux} = 30ps + 90ps + 35ps = 155ps$. This means that the computer could run as fast as 6.45 GHz if other elements could be made faster.
- 2) $T_C \geq t_{pcq} + t_{D-mem} + t_{setup} = 30ps + 350ps + 20ps = 400ps$

Part IV

- 1) $0.5T_C \geq t_{pcq} + t_{mux} + t_{setup} = 30ps + 35ps + 30ps = 95ps$; $T_C \geq 190ps$
- 2) $0.5T_C \geq t_{ctr} + t_{setup} = 130ps + 20ps = 150ps$; $T_C \geq 300ps$
- 3) No, a 3.2GHz clock corresponds to 312ps, which is slower than the max delays in the Writeback and Decode stages.

Part V

- 1) The stage with the highest possible delay is the Memory stage, with a 400ps max delay. Therefor the minimum possible clock period is 400ps.
- 2) The Instruction Memory must have a maximum t_{pd} of 262ps.
The Data Memory must also have a maximum t_{pd} of 262ps.

Exercise B

Part I

- 1.) The second hazard is the use of the lw result as a source in the add instruction of line 8. During the Execute stage of the add instruction, the Hazard Unit detects that RtE (01000_{two} for \$8) matches WriteRegW (also 01000_{two} for \$8) and that RegWriteW=1, so it sets ForwardBE=01 so that ResultW (the lw result) is passed into the "B" input of the ALU.
- 2.) The third hazard is the use of the use of the add result as a source in the addi instruction of line 9. During the Execute stage of the addi instruction, the Hazard Unit detects that RsE (10011 for \$19) matches WriteRegM (also 10011 for \$19) and that RegWriteM=1, so it sets ForwardAE=10 so that ALUOutM (the add result) is passed to the "A" input of the ALU.
- 3.) The fourth hazard is the use of the the addi result as a source in the sw instruction of line 10. During the Execute stage of the sw instruction, the Hazard Unit detects that RtE (10011 for \$19) matches the WriteRegM (also 10011 for \$19) and that RegWriteM=1, so it sets ForwardBE=10 so that ALUOutM (the addi result) is passed to the "B" input of the ALU.

Part II

The proper way to correct this hazard is to use a combination of stalling and forwarding. However, the circuit in Figure 7.50 is not equipped to stall the pipeline, so an outdated value will be stored instead.