Second Midterm Examination

Due Date: 11/30/09.

Problem 1. (10 pts.) Consider the following R-Type variant of the *lw* instruction:

\[ \text{lwr } $t0, $t1, $t2 \]

which takes the contents of $t1, adds it to the contents of $t2 and uses this as the address in memory, whose contents are loaded into $t0.

Can we support the *lwr* instruction in our single clock-cycle datapath (Figure 4.24) “as-is”? That is, no new state elements, or combinational logic components (which also includes mux’es). The only way one can implement this new instruction is by defining the control signals appropriately so as to implement this new instruction.

If your answer is yes, indicate what ALL the control signals should be. If your answer is no, indicate what new components are needed and indicate what ALL the control signals should be (including any new ones you define).

Problem 2. (10 pts.) Using the pipelined datapath of Figures 4.65/4.57/4.51 describe in detail how one might implement the

\[ \text{jal } label \]

instruction. Indicate the value of ALL control lines, including any new ones you may elect define. The goal is to implement this new instruction with a minimum of new components.

Problem 3. (6 pts.) Consider the following code, executing on a pipelined datapath:

\[
\begin{align*}
\text{add } &\quad $t0, $t1, $t2 \\
\text{add } &\quad $t2, $t0, $t1 \\
\text{sw } &\quad $t0, 100($t1) \\
\text{lw } &\quad $t2, 100($t1) \\
\text{add } &\quad $t3, $t0, $t2
\end{align*}
\]

1. Identify the data dependencies that are hazards in the above code.
2. Of the hazards identified in part a, identify the ones that can be resolved by forwarding.
3. How many cycles will this code take to execute?

Problem 4. (4 pts.) Can all data hazards be eliminated? If yes, explain, if no, indicate which specific data hazards cannot be avoided and why they cannot be avoided/eliminated.
Problem 5. (5 pts.) A control hazard can lead to a pipeline flush. A data hazard can lead to a pipeline bubble. Can both happen simultaneously? If so, give an example. If not explain why not. (Don’t forget to provide any assumptions you are making.)

Problem 6. (10 pts.) Consider a program consisting of five conditional branches. The program will be executed thousands of times. Below are the outcomes of each branch for one execution of the program (T for taken, N for not taken).

- Branch 1: T-T-T
- Branch 2: N-N-N-N
- Branch 3: T-N-T-N-T-N
- Branch 4: T-T-T-N-T
- Branch 5: T-T-N-T-T-N-T

Assume the behavior of each branch remains the same for each program execution. For dynamic schemes, assume that each branch has its own prediction buffer and each buffer is initialized to the same state before each execution. List the predictions for the following branch prediction schemes:

a: Always taken
b: Always not taken
c: 1-bit predictor, initialized to predict taken
d: 2-bit predictor, initialized to weakly predict taken

What are the prediction accuracies for each?

Problem 7. (5 pts.) A single-cycle datapath usually leads to superior instruction latency, yet all modern datapaths are organized as pipelined datapaths! Explain. Don’t forget to define instruction latency in your answer.

Problem 8. (10 pts.) You are a new architect at “Chips from MikeyG.” For a new chip being designed you have been asked to defend the design of a long pipeline (say 15+ stages). Provide a succinct argument in favor of this position; i.e. describe the advantages of such a design as well as the disadvantages of the alternative - a short pipeline (say 5-7 stages).

Now reverse your role and provide a succinct argument against this position; i.e. describe the disadvantages of the long pipeline design as well as the advantages of the short pipeline design.

Finally, indicate which you would select; the long pipeline approach or the short pipeline approach. Describe why.