

zachary_hightower_hwk_10

CSCI 223 Computer Org. & Assembly Language

Homework #9

4/9/2024

Due: 11:59pm on Wednesday 4/10/2024

Submission: Upload a file (.txt or .pdf) that has your answer computer-typed. No late submission will be accepted.

1. Explain instruction pipelining in one short sentence

Ans: Divide the execution of an instruction into a series of sequential steps, or stages, performed by different hardware units

2. What is the cause and improving techniques of structural hazard?

Ans: Cause is when a required source does not exist (or is busy). We could also say it is a Conflict for use (or lack) of a hardware resource.

Improvement Techniques

The L1 cache for instruction and the L2 cache for data represents an attempt to solve this problem. Avoiding the structural hazard requires separate instruction/data memories or separate instruction/data caches. So, we need to have more hardware resources dedicated to specific tasks so that we do not run into the issue of conflicts or lack.

3. What is the cause and improving techniques of data hazard?

Ans: Cause is when an instruction depends on completion of data by a previous instruction. Could be put another way as we need to wait for previous instruction to complete the data read/write

Improvement Techniques

The general idea for how to solve this, is to look for an independent instruction somewhere. If we put that independent instruction somewhere, we can use the time that would be taken up by the bubble.

In that way, we could completely fill the bubble caused by the instruction

Compilers usually do this, they shuffle the order of instructions to make the program the most efficient that it can be.

Another solution

- Use a result as soon as it is computed
 - Don't wait for it to be stored in a register
 - Requires extra connections in the hardware circuit

This means that as soon as we add, we have the results, and then we allow the next instruction to use those results. We overload one thing so that it can be both output for one instruction and input for the next instruction

This technique is used everywhere, it doesn't solve every data issue, but it does solve it for simple issues like the one presented in class like this

```
add $t0, $t1, $s0$  
sub $s0, $t3, $t2$
```

4. What is the cause and improving techniques of control hazard?

Ans: Cause is deciding on control-flow action (e.g conditional jump) depends on previous instruction.

We restate this in more detail below:

- Branch determines the flow of instruction execution
 - Fetching next instruction depends on branch outcome
 - For example, do we fetch the first instruction of if or else body? it depends on the branch outcome
 - Pipeline can't always fetch correct instruction
 - Still working on ID stage of branch

Improvement Techniques

- Predict outcome of branch
 - Only stall if prediction is wrong
- Say we have a loop, we can predict that the loop will continue, and that will be true most of the time

There are two ways of doing this shown below

- Static branch prediction
 - Based on typical branch behavior
 - Example: Loop branches

- Predict backward branches taken (higher probability)
- Dynamic branch prediction
 - Hardware measures actual branch behavior
 - e.g Record recent history of each branch
 - Assume future behavior will continue along the trend
 - When wrong, stall while re-fetching and updating the history