

Quiz #6 Answers

1. Give me two example pieces of code that involve what would typically called data hazards. For the first one it should be fixable with forwarding in the pipeline that we designed. For the second one show an example of commands where forwarding alone can't help. Each need only be two assembly lines long.

```
add $t0, $t1, $t2  
add $t3, $t4, $t0
```

Anything where the first is an ALU instruction and the second uses the register the first stores to counts.

```
lw $t0, 0($t2)  
add $t3, $t4, $t0
```

Here the first command is a load and the second command uses where it stores to. Because memory is read in the 4th step forward alone can't help here.

2. Describe in basic terms the role of cache memory in computer systems. Why are they important to have?

The cache memory is a smaller and faster storage space that sits close to the processor. It holds some of what we keep in real memory with the idea that when we access memory that can be found in the cache we don't spend as much time waiting for a reply as we would when we have to go all the way out to main memory. The cache is only effective if the program exhibits temporal and spatial locality. Because the delay in going to memory is significant, not having a cache would cause the CPI to be much larger than 1, even on a superscalar dynamic pipelined system that should be able to get well under 1 for its CPI.

Extra Credit: I/O performance is more complex than processor performance. Amdahl's law tells us that this complexity is worth dealing with though. Why is that?

The idea behind Amdahl's law is that there are diminishing returns from optimizing a single aspect of a computer's performance because the unoptimized aspects grow in the fraction of the total execution time they consume. In this case, if you don't try to optimize I/O performance, it will come to dominate total runtimes as other parts, the processor in particular, get faster.