
CS2100 Computer Organization

Tutorial #10: Pipeline

4 November – 8 November 2024

For this tutorial, you are provided some helper slides which contain the necessary datapaths. This is found in file *Tut10-Helper.pptx*.

1. Using Powerpoint animation (or just a number of slides), give the full details of the signals and data bits (highlighting those lines that are active) of the following 3 instructions are executed together in a pipelined MIPS.

- (a) `sub $1, $2, $3`
- (b) `lw $6, 4($7)`
- (c) `beq $4, $5, L2`

Assume that L2 at PC=0x100 instruction while the beq is at PC=0x1000.

2. Consider the following sequence of MIPS instructions.

- (a) `add $1, $2, $3 # PC = 0x100`
- (b) `add $1, $1, $3 # PC = 0x104`
- (c) `add $1, $1, $1 # PC = 0x108`

In the datapath with forwarding, trace the execution of these 3 instructions as they pass through the pipeline with attention paid especially to the forwarding of operands. Clearly identify which rule(s) of the forwarding (or hazard detection) was fired, if any.

3. Repeat Q2 for the following instruction sequence.

- (a) `add $1, $2, $3 # PC = 0x100`
- (b) `lw $1, 0($1) # PC = 0x104`
- (c) `add $1, $1, $1 # PC = 0x108`
- (d) `add $3, $2, $1 # PC = 0x10c`

4. Using the graphical notation for pipeline introduced in class, show how the following sequence of instructions would be executed in a datapath with forwarding, and branch resolution at the ID stage by the following instructions

```
add $1, $2, $3 # PC = 0x100
lw $1, 0($1) # PC = 0x104
beq $1, $0, L # PC = 0x108
add $3, $2, $1 # PC = 0x10c
```

- (a) assuming that the `beq` turns out to be not taken.
- (b) assuming that the `beq` turns out to be taken.

5. **For Fun** ☺: Do you want to know what some old hardware and computers look like? Have you had a chance to visit the wonderful “NUS Computing Gallery”. If not, time to explore :-)