Digital Design LU

Protocol

Lab Exercise III

Group 7

Markus Reiß, Matr. Nr. 1126188

e1126188@student.tuwien.ac.at

Lorin Hillebrand, Matr. Nr. 0925917

e0925917@student.tuwien.ac.at

Petar Kosic, Matr. Nr. 1225761

petar.kosic@student.tuwien.ac.at

Vienna, December 15, 2015

Pipeline Simulation

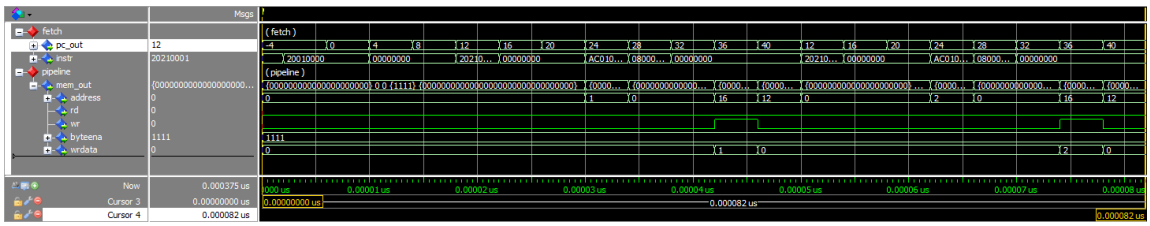

Figure 1: Simulation screenshot for Listing 1 - first two loop cycles. Note: Adress and wrdata in deciaml, instr in hex.

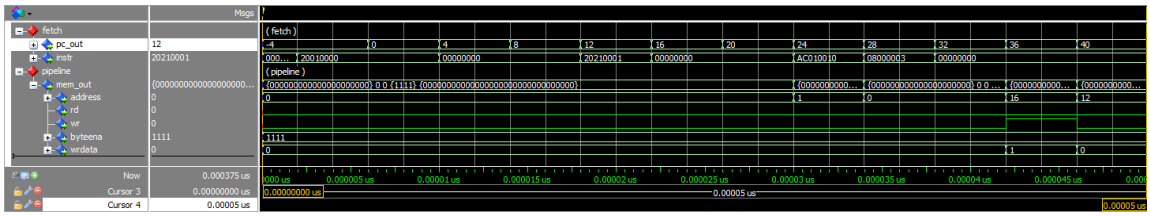

Figure 2: Simulation screenshot for Listing 1 - first loop cycle.

Make sure the following signals are visible in Figure 2: the program counter in the fetch stage, the instruction being fetched, and the fields `address`, `rd`, `wr`, `byteena`, and `wrdata` in the `mem_out` signal coming out of the pipeline.

Listing 1: Assembler example without forwarding

```
        addi $1, $0, 0
        nop
        nop
loop:
        addi $1, $1, 1
        nop
        nop
        sw $1, 16($0)
        j loop
        nop
        nop
        nop
```

Feedback & Comments

By answering the optional questions below you can give us feedback and help us to further improve this lab course. Your answers will not influence your grading!

Question: How many hours did you need to solve this lab exercise? Please give us a rough estimate.

Answer:

Question: Were there any annoying problems you encountered (e.g. bugs in tools, flaws in the task description or documentation, etc.)?

Answer:

Question: Other remarks?

Answer: