Digital Design and Computer Architecture LU

Lab Protocol

Exercise III

Group 5

Simon Boehm, 12123460

e12123460@student.tuwien.ac.at

Mihai-Andrei Dancu, 12123694

e12123694@student.tuwien.ac.at

Vienna, June 5, 2023

Task 1

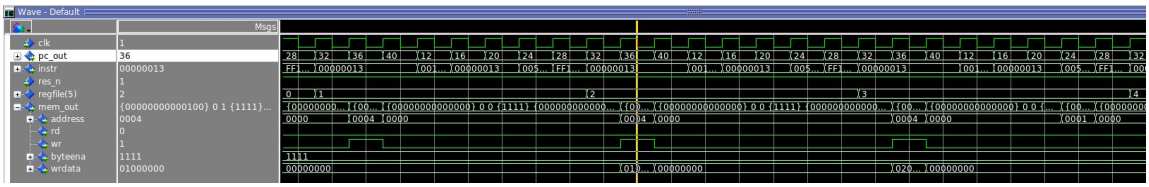

Figure 1: Simulation screenshot for Listing 1.

Make sure the following signals are visible in Figure 1 and the signal values are readable: the program counter in the fetch stage, the instruction being fetched, the content of register `x5` and the fields `address`, `rd`, `wr`, `byteena`, and `wrdata` in the `mem_out` signal coming out of the pipeline in the memory stage.

Listing 1: Assembler example without forwarding

```
        addi x5, x0, 0
        nop
        nop
loop:
        addi x5, x5, 1
        nop
        nop
        sw x5, 16(x0)
        jal x0, loop
        nop
        nop
        nop
```

Task 2

Describe how you organized the work within the group on the way to the submission, i.e., summarize how you partitioned the work, which group member was (mainly) responsible for which subtask(s), how you organized testing and integration, how well your time plan worked, which re-adjustments needed to be done, and how well your collaboration worked.

We split up level one: Simon Boehm: ALU and Regfile ;Mihai-Andrei Dancu: Memu. We did randomized testing using python scripts to verify these components as much as possible and make sure no faults will persist in these components in level 1.

For the second level we split the work as follows: Simon Boehm: Decode and Execute; Mihai-Andrei Dancu: Fetch, Memory and Writeback These stages were also tested using randomized test vectors generated by a python script.

The pipeline and subsequent debugging was performed by both participants. At the pipeline stage some troubleshooting was needed but all in all the collaboration worked quite well.