Digital Design and Computer Architecture LU

Lab Protocol

Exercise III

Group 13

Bernhard Ruhm, Matr. Nr. 12019841

e12019841@student.tuwien.ac.at

Slavomir Slavtchev, Matr. Nr. 12021361

e12021361@student.tuwien.ac.at

Vienna, June 5, 2023

Task 1

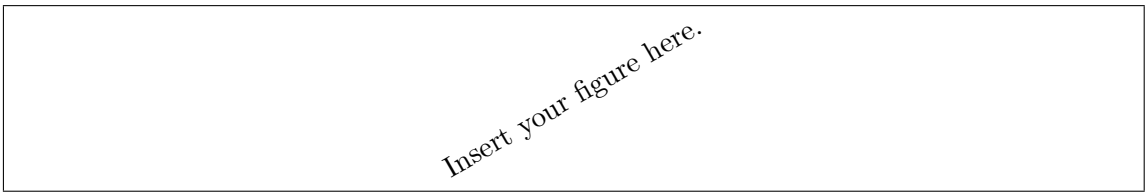

Figure 1: Simulation screenshot for Listing 1.

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.

Bernhard Ruhm is responsible for the regfile, fetch, decode, execute and pipeline and Slavomir Slavtchev is responsible for alu, memu, mem and wb. Beforehand, we tried to get an overview of the different stages and components and discussed it together. Next, we partitioned the work and began implementing. Especially more complex designs, such as the decode stage were implemented bit by bit and tested in between, to ensure correct behaviour and to simplify the debugging process. On the other hand, smaller designs were implemented first and then tested afterwards. Debugging and Testing definitely took longer than expected and we spend most of out time in creating testcases and searching for errors.