Design of Digital Circuits
Lab 8 Supplement

Prof. Onur Mutlu
ETH Zurich

Spring 2018

2 May 2018

Lab 8 Overview

- You will build a whole single-cycle processor and write assembly code that runs on the FPGA board

- Don't worry! You have 2 sessions for the lab and it will give you up to 6 points

- Each session will have an optional report so you can gain up to 2 bonus points

- The hand-in and deadline is the same as for the previous report (by 15.06.2018 via Moodle)

Lab 8 Session I: The MIPS Processor

- Download the Vivado project from the course website

- A lot of parts are already implemented for you!

- What you will have to implement
  - Compute the Instruction Memory address and read the instruction
  - Connect the ALU
  - Compute the Data Memory address and add the necessary wires
  - Instantiate the Control Unit

Lab 8 Session I: The MIPS Processor I/O

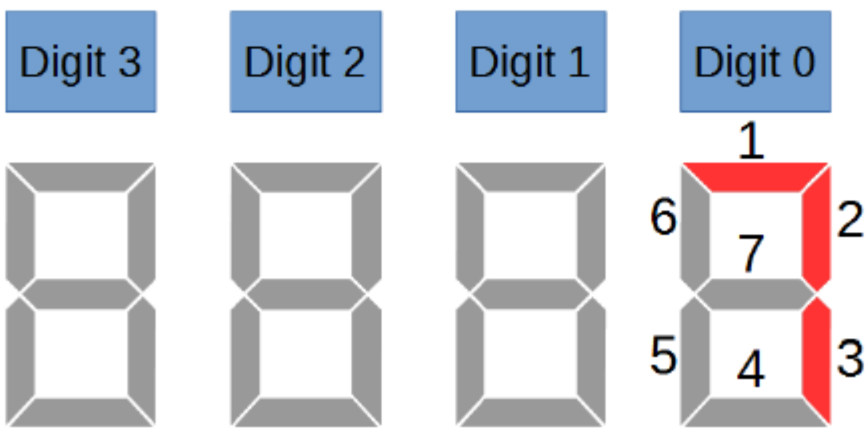
- Your goal is to control the 7-segment display with your assembly program

- You will need to complete the I/O controller so the output of the processor will be correctly mapped to the display

Lab 8 Session I : The Crawling Snake

- You learned how to write assembly code in Lab 7

- This time you will implement a crawling snake on the 7-segment display



- Write the code in the MARS simulator and export it

I/O in Assembly

- How do we communicate with the display?

- Memory Mapped I/O
  - We designate specific addresses for the I/O
  - We can read and write to those addresses
    - Example

```
# write contents of $t0 into memory at address 0x7FF0
# so that the I/O controller can send it to the display

                    sw $t0, 0x7FF0($0)
```

Lab 8 Session II: Speed Up The Snake

- You will modify your assembly code to accept inputs
  - The snake should crawl at different speeds for different inputs
  - The inputs will be controlled by switches on the FPGA board

- Modify the I/O controller to accept the inputs

- Challenges
  - Change the direction of the snake
  - Change the pattern of the snake

Last Words

- IMPORTANT! Use only the following instructions
  - Arithmetic: ADD, SUB, SLT, ADDI
  - Logic: XOR, AND, OR, NOR
  - Additionally, you can use: BEQ, J, SW, LW
- Other instructions are not supported by our processor
- Even if the code runs fine in MARS, the instructions will be skipped by the processor

Design of Digital Circuits
Lab 8 Supplement

Prof. Onur Mutlu
ETH Zurich
Spring 2018
2 May 2018