Essential Skills: Assignment Computer Architecture (1) 2016-2017

## 1. Digital Circuit

a. Draw a digital circuit of an 4-bit incrementer, i.e., a circuit that satisfies the equation b = a + 1 mod 24. You can use the operations from the set {and, or, nand, nor, not, xor}.

b. Try to find an incrementer with the minimal number of operations needed.

c. Try to minimize the depth of the circuit, i.e. the maximal length of any path from input to output [Optional].

## 2. Experiments Performance measurement

This assignment has been designed by the UvA team in charge of delivering computer architecture course at both Ba and MSc levels. The lab was given as introductory lab for the course computer organization at IU[1]. The objective of the assignment is to give hands-on experimentation on the subject of performance as discussed in chapter 1 of Patterson and Hennessy. You will have to measure the performance of three different architectures (single cycle, multi-cycle, and pipelined) for three different program

a/ Quick start with SIM-PL and the steps needed to perform the required experiment are described [1]. (You do not have to following the instruction in slide 3 about the submission of the lab1)

- Download SIM-PL: https://staff.fnwi.uva.nl/t.r.walstra/innopolis/SIM-PL_2.3.0.zip
- Download lab components: https://staff.fnwi.uva.nl/e.h.steffens/Downloads/Arco/Innopolis/Components%20-%20Lab%201.zip
- Follow instructions presented in [1] to load and run the programs:
    - orloop.wasm
    - addition.wasm
    - squares.wasm

b/ For every program measure the following parameters. You can consult the lectures notes of the introductory lecture given by Taco Walstra [4]

- The number of instructions that have been executed for every architecture.
- the CPI
- the cpu time

c/Explain the following

- Which architecture performed the best for every executable and explain why
- A program counter is present in the architecture. Can you explain what the program counter function is?
- What do you notice about the behavior when running a simulation?

d/ In the programs there are different instructions. Explain for the following instructions what do they do and how the are used.

- ADDI
- NOP
- LI
- BNE

e/ Now change the first program to subtract two numbers, do not change the initialization of the variables can you explain what happens.

[1] Introductory lab of the course computer organization at IU given by Edven Steffens and Taco Walstra https://staff.fnwi.uva.nl/e.h.steffens/?page_id=101

f/ Finally explain why the clock rate (frequency) of a cpu is not a good benchmark for the performance of a cpu.

## 3. Write an assembly code to convert little-endian to big-endian (optional)

Write an assembler program (for the Harvard machine) that converts memory data from little endian to big endian. Assume that the address of the memory data that should be converted is stored in register 1. Test your assembly code using the SIM-PL

## Reference:

[1] SIM-PL presentation https://staff.fnwi.uva.nl/e.h.steffens/Downloads/Arco/Innopolis/Lab%201%20-%20Introduction.pdf

[2] SIM-PL Manual https://staff.fnwi.uva.nl/e.h.steffens/Downloads/Arco/Innopolis/SIM-PL%20instruction%20manual%202.0%20-%20UK.pdf

[3] Taco Walstra Lecture notes Computer Architecture Introduction https://staff.fnwi.uva.nl/t.r.walstra/innopolis/lect01.pdf