The Final Project for the Computer Organization Course (CMPE 344.01) Fall 2020

In the following, the final projects for the CMPE 344.02, Computer System Organization Course, have been listed and described. The students have an option to select one of these projects as their final Project. In the following Table, the rules and conditions for different projects have been shown. It should be noted that for the Projects I and II, there will be a bonus score which will be directly added to your final grade.

	Group Project	Bonus is Available	Bonus Score	Dead Line
Project I	Yes (2 students)	Yes	Extra 20%	17, Feb, 2021
Project II	Yes (2 students)	Yes	Extra 15%	10, Feb, 2021
Project III	Yes (2 students)	Yes	Extra 10%	10, Feb, 2021
Project IV	NO	NO	----	24, Jan, 2021

Project I: Design of a Processor using a Hardware Description Language

In this project, you should design a simple risc processor, the so called BU2020 Processor. The description of the project has been explained in **Appendix A** that can be found in Moodle.

Project II: Working with an open-source Simulator

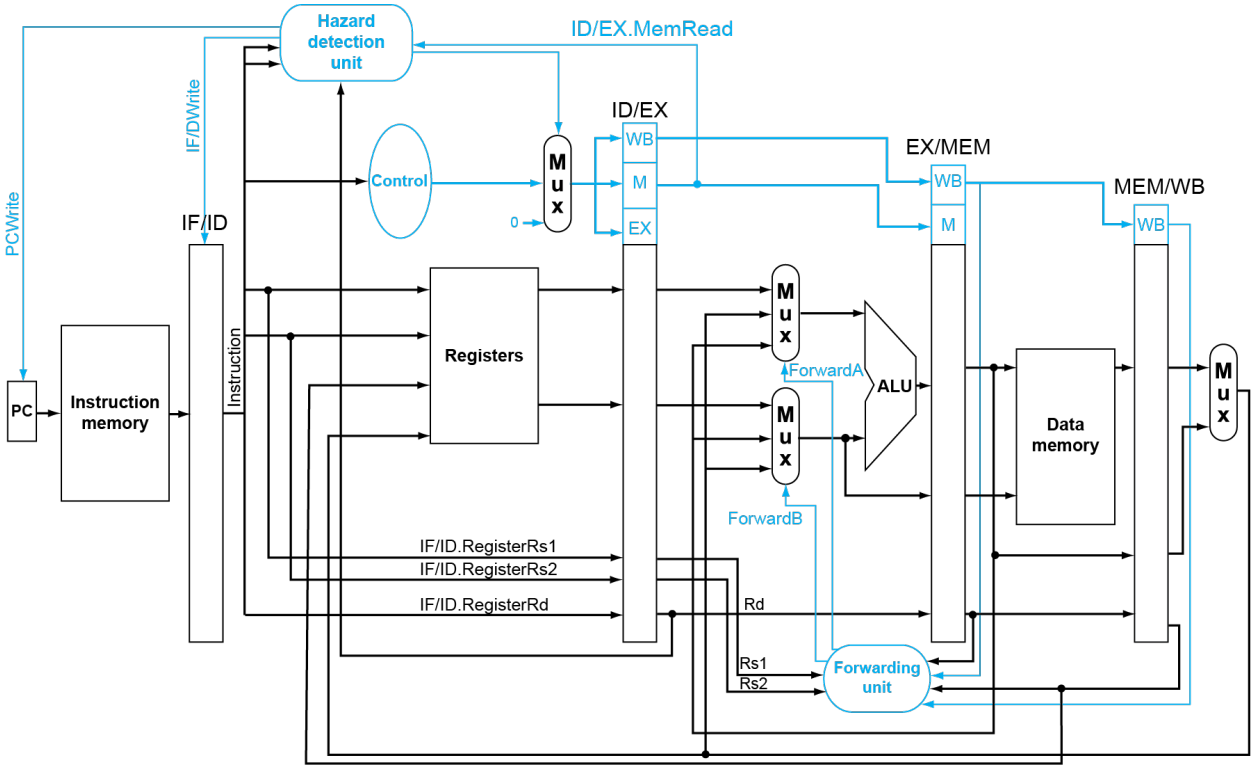
In the following table, you can see a list of open-source RISC-V simulators as well as the corresponding webpage. You should select one of the simulators and perform the following steps:

1- Install the Simulator.
2- Cross compile and Run two optional test programs.
3- Prepare a Report document for performance parameters extracted by the simulator.
4- A technical report on how to install the simulator and the corresponding packages.
5- Create a VM (Virtual Machine) in which the simulator has been installed. Please make sure if it works on another computer as well.

	Simulator Name	Abstraction Level	Webpage
1	Veriloator	Verilog/SystemVerilog simulator	https://www.veripool.org/projects/verilator/wiki/Manual-verilator
2	RISC-VP Virtual Prototyping	Software Level Simulator (C++)	https://github.com/agra-uni-bremen/riscv-vp A tutorial paper: https://past.date-conference.com/system/files/file/date19/ubooth/41824.pdf
3	Berkeley Out-of-Order Machine (BOOM)	synthesizable and parameterizable open source RV64GC	https://boom-core.org A technical Report: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-167.pdf
4	riscvOVPsim	Software Level Simulator	https://www.ovpworld.org/riscvOVPsimPlus/ User Guide: https://github.com/riscv-ovpsim/imperas-riscv-tests/blob/main/riscv-ovpsim/doc/riscvOVPsim_User_Guide.pdf
5	gem5-RISC V architecture	Software Level Simulator	https://gem5.googlesource.com/public/gem5/ A tutorial paper: https://carrv.github.io/2017/papers/roelke-risc5-carrv2017.pdf

Project III: Designing A Datapath Simulator for A RISC-V Architecture

In this project, you should develop a simulator using a high level programing language such as C++, JAVA, and Paython, etc. for the data and the control paths of the RISC-V case study I explained in the chapter 4 lectures (See the following Figure). The simulator gets a sequence of RISC-V instructions as a text file and then perform a cycle accurate simulation. It should have the ability to provide 1) CPI, 2) Total number of Clock Cycles, 3) total Number of Stalls, 4) a list of Instructions causing stalls and the corresponding number of stalls. (Note: the implementation of branch prediction unit is optional)



Project IV:

The description of the project has been explained in Appendix B that can be found in Moodle.