

Universidade de Aveiro

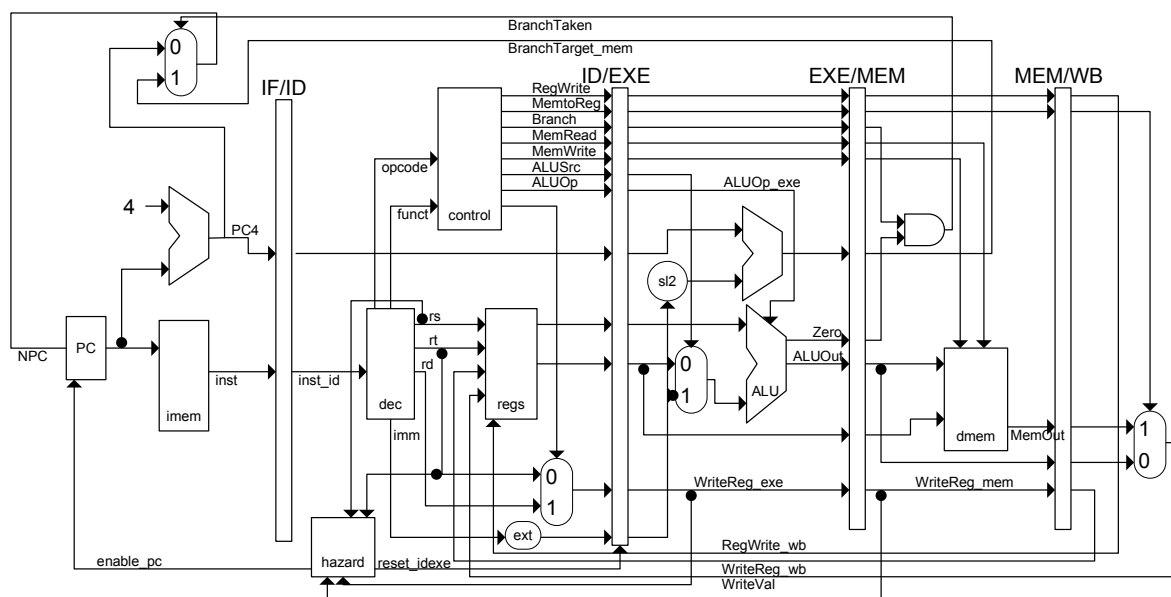
Mestrado Integrado em Engenharia de Computadores e Telemática

Arquitectura de Computadores Avançada

DLX – Pipelining 1

Academic year 2020/2021 Adaptation of exercise guide by Nuno Lau/José Luís Azevedo

1. The `DLX.zip` archive contains a DLX/MIPS compatible graphical pipeline simulator written in Java that will be used to run a few simple code examples that illustrate some features of pipelining.
 - 1.1. Download and decompress the `DLX.zip` archive to your workspace under Linux.
 - 1.2. Read the `README` file to learn how you can run it.
2. Consider the programs `ex_1.s` and `ex2.s` located in the directory `./DLX/apps/dlx_apps` in turn. Read them carefully, trying to understand what they do.
 - 2.1. Determine the values that should be stored in the registers at the end of the execution.
 - 2.2. Sketch the clock cycle diagram of the execution in a processor with the five-stage pipeline architecture depicted below.



- 2.3. Run the code in the DLX simulator with the *forwarding* option turned off. Check if the execution is correct. If not, explain what has happened?
- 2.4. Intersperse `nop` instructions in the code so that the program now behaves as expected. What is the new clock cycle count for running the program?
- 2.5. Run the original code in the DLX simulator with the *forwarding* option turned on. Compare the results with the previous runs.

3. Write a program that adds up the values of an integer array stored in memory. Run the code in the DLX simulator with the *forwarding* option turned off and then on. Take also into account in your runs the *branch predictor* alternatives: *none*, *static – predict always not taken* and *static – predict always taken*, and *initial predictor state* equal to *no initial state*. Discuss the results.