

Politecnico di Torino

# Testing and Fault tolerance

# Report Contest

Author: Grottesi Lorenzo Ostovar Iman

The first approach to the contest has been the translation of groups 4 and 8 that was had good results in the previous contest. All the store operation has been changed in order to access only the data section. Furthermore, in order to preserve the status a PUSH\_ALL() and POP\_ALL() functions were written in the PUSH\_POP.h file. These two function are used to save the content of the registers in a dedicated part of the data. This section will be unused for rest of the test. At the end of each test the content of the register is restored. Figure below shows the memory map.

Table

Description automatically generated

Figure 1 - Memory map

After first simulation we seen GR8 and GR4 weren’t enough so we add

* GR20\_ALU.s
* GR20\_INTEGRATION.S
* GR19\_test\_ls.S
* GR19\_test\_if.S
* GR14\_TEST\_CSR.S

Running a complete simulation for these groups. In particular even if CSR unit is not part of the test we decide to add the GR14\_TEST\_CSR.S to trigger as much instruction as possible in the fetch and decode stage. Lastly we add the remaining percentage performing an incremental simulation using the simple\_hwl.S and my\_ls\_test.S files, in particular the first is used to improve the hardware loop, we noticed that the CSR related to this unit changed with respect the 2017 one. So we changed the CSR from 0x7B- to 0x7C-. Lastly my\_ls\_test.S is a test written by us to reach the 90% covering the remaining percentage of the Load and Store unit.

![Text

Description automatically generated]()The overall simulation result is reported in figure below

Figure 2 - Report faults hierarchy