Politecnico di Torino

Testing and Fault tolerance

Report Contest

Authors:

Grottesi Lorenzo 268978, Ostovar Iman 272561

The first approach to the contest has been the translation of groups 4 and 8 that had good results in the previous contest. All the store operation has been changed to access only the data section. Furthermore, to preserve the status a PUSH_ALL() and POP_ALL() functions were written in the PUSH_POP.h file. These two functions are used to preserve the content of the registers in a dedicated part of the data by handling a stack pointer. This section will be unused for the rest of the test. At the end of each test, the content of the register is restored. The figure below shows the memory map.

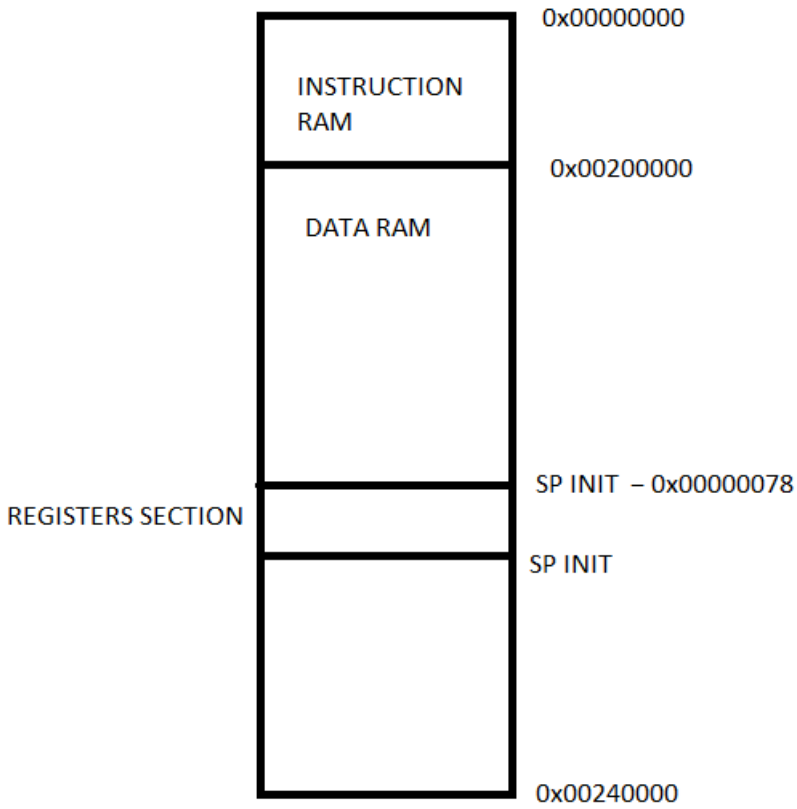

Figure 1 - Memory map

After the first simulation, we saw GR8 and GR4 weren't enough so after trying the effectiveness of the other group's codes according to not covered faults. Finally, we found the very good optimized combination of codes included below:

- **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".

However, the ***CSR*** unit is not part of the test, we decide to update and 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 concerning 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 90% covering the remaining percentage of the Load and Store unit.

The overall simulation result is reported in the figure below :

instance	faults	cumulative
/	162146	90%
/if_stage_i	17028	69%
/if_stage_i/prefetch_128_prefetch_buffer_i	10588	76%
/if_stage_i/prefetch_128_prefetch_buffer_i/L0_buffer_i	3780	70%
/if_stage_i/hwloop_controller_i	1628	64%
/if_stage_i/compressed_decoder_i	1504	86%
/id_stage_i	65582	91%
/id_stage_i/registers_i	39590	99%
/id_stage_i/registers_i/riscv_register_file_i	39590	99%
/id_stage_i/decoder_i	4136	80%
/id_stage_i/controller_i	2278	41%
/id_stage_i/int_controller_i	184	7%
/id_stage_i/hwloop_regs_i	3222	78%
/ex_stage_i	72788	96%
/ex_stage_i/alu_i	32094	94%
/ex_stage_i/alu_i/alu_popcnt_i	340	100%
/ex_stage_i/alu_i/alu_ff_i	416	100%
/ex_stage_i/alu_i/int_div_div_i	5280	96%
/ex_stage_i/mult_i	39780	97%
/load_store_unit_i	5608	78%
/cs_registers_i	30680	NAN
/RISCY_PMP_pmp_unit_i	76610	NAN

Figure 2 - Report faults hierarchy

Concerning the size the elf file requires about 60 kbytes considering all the sections.

text	data	bss	dec	hex	
56410	2116	2104	60630	ecd6	bytes

The duration of the simulation requires 5:15 minuntes in Det server and abou 2 min in DAUIN server.

Resuming the results:

- Test coverage 90,18%
- Code size 60 Kbytes.
- 5:15 minutes.

Due to an issue on DAUIN server was not possible to complete the last complete simulation, so the test coverage (figure 2) refers to the increamental one.