Integrated Systems Architectures

Design of a RISC-V-lite processor
Description

1 Problem description

In this lab you have to design in VHDL a RISC-V-lite processor with 5 pipeline stages. The RV32I has a 32-bit fixed-width instruction set, where instructions are organized in six classes:

- R or register;
- I or immediate;
- S or store;
- SB or conditional branch;
- U or upper immediate;
- UJ or uncoditional jump

The full list of instructions is summarized in the RISC-V reference data card. More details can be found in the RISC-V Instruction Set Manual, Volume I: User-Level/Unprivileged ISA (available at `https://riscv.org/specifications/`). In particular, Table *RV32I Base Instruction Set* at page 146 (Document Version 20190608-Base-Ratified) summarizes opcode values and coding for each RV32I base instruction.

The instructions supported by the RISC-V-lite processor are the following ones, a subset of the whole RV32I:

- arithmetic `add, addi, auipc, lui`
- branches `beq`
- loads `lw`
- shifts `srai`
- logical `andi, xor`

- compare `slt`
- jump and link `jal`
- stores `sw`

Refer to the slides of the course (Part 3-A Processor architecture) for the details. **Note:** the RISC-V ISA does not explicitly include the `nop` instruction. So it is implemented as `addi x0,x0,0`.

2 Simulation and validation

In order to simulate and validate the architecture you need to translate the ASM to machine language. A tool, which can help in i) understanding instructions flow, ii) see ASM excecution and iii) translate the ASM is the free RARS simulator:
`https://github.com/TheThirdOne/rars/releases/tag/v1.3.1`.