RES Homework 4

Marco Deuritz, Lev Likhanov

June 16, 2025

Task 2

(a) The state of out RISC-V processor can be fully described by the Program Counter (PC) and the 32 registers (x0 - x31) defined in the RISC-V architecture. In our case we aren't utilizing control and status registers (CSR) of floating point registers.

(b) The opcode is used to distinguish between different instruction formats. Although different opcodes like for LUI and AUIPC could still refer to the same instruction format (U-Type).

(c) A register file needs to contain 32, 32-bit wide registers. Reading is done through two input multiplexers (MUX) for *rs1* and *rs2* which will return *rd1* and *rd2* values respectively. Writing would be handled by an output multiplexer (DEMUX) that writes the value of *wd* into *wa*.

There would have to be a special case for *x0* which will always return a zero and can't be written to.

(d) U-Type, S-Type and I-Type immediate field are placed differently inside the 32 bit instruction in order to preserve the same bit positions for the opcode, rd, rs1, rs2, funct3 and funct7. This allows the decoder to be simpler, as the bit fields it outputs stay at the same position.

Differentiation of the instruction types is done by analyzing the opcode.

The sign extension in VHDL is done by using the resize function which takes the MSB or the input vector and extends it to the wanted length.

```
1  imm_temp <= resize(signed(instr(31) & instr(19 downto 12) &
       instr(20) & instr(30 downto 21) & '0'), 32);
```

Listing 1: Snippet from imm$_g$*enerator.vhd*

(e) The provided code has 444 lines of machine code with each line having a 32 bit instruction. This means our ROM needs to be a minimum of 14308 bits in size. This means practically a memory size of 16 Kbits or 2 KBytes is necessary to store the program code.

A Source Code

A.1 Task 1: Multiply Accumulate