Aeronautics Institute of Technology – ITA
Computer Science Division – IEC

EEA-25 Programmable Digital Systems
Prof. André F. Ponchet
Teaching assistant: Felipe Ferreira

Laboratory 06
Last document update: October 23, 2023

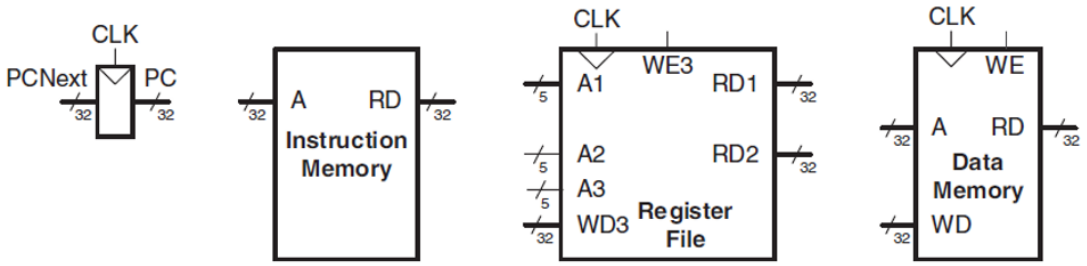
1 Objectives

1.1 General Objectives

- Familiarize yourself with the Lattice iCE40 development environment;
- Understand circuit design process using HDLs and FPGA implementation;
- Simulate and verify results with real FPGA boards.

1.2 Specific Objectives

- Start the development of the basic modules responsible for the RISC-V core architectural state;
- Implement a 32-bit Register File.



Architecture state.

2 Lab Development

2.1 Circuit analysis

A Register can be classified as a type of memory frequently used inside processors due to their high operation speed. Registers are made of flip-flops (FFs) that can store one bit of data.

A register file can read two registers and write to one register.

The RISC-V register file contains a total of 32 32-bit registers. Therefore, 5 bits are used to specify the register of interest.

- **Register File:** The register file places 32-bit values on RD1 and RD2 read data outputs related to the addresses A1 and A2, respectively. The A3 port receives a 5-bit address input and WD3 a 32-bit write data input. If write enable (WE3) is enabled, the register file writes data (WD3) to the specified register (A3) on the rising edge of the clock (CLK).
- The HDL provided below is incomplete, it must be completed to obtain the equivalent HDL that represents the register file.
- **Requested requirement:** The registers must be initialized with a predefined value corresponding to their number. For example:
  - x0 = 32h00

Name	Register Number	Use
zero	x0	Constant value 0
ra	x1	Return address
sp	x2	Stack pointer
gp	x3	Global pointer
tp	x4	Thread pointer
t0–2	x5–7	Temporary registers
s0/fp	x8	Saved register/Frame pointer
s1	x9	Saved register
a0–1	x10–11	Function arguments/Return values
a2–7	x12–17	Function arguments
s2–11	x18–27	Saved registers
t3–6	x28–31	Temporary registers

Register File.

- x1 = 32h01
- x2 = 32h02
- ⋮
- x31 = 32h1F

1. Create a folder named "LAB-6";
2. Create and complete the hardware description of a module called "reg_file":

```
module reg_file (
    rg_A1,
    rg_A2,
    rg_A3,
    rg_WD3,
    rg_RD1,
    rg_RD2,
    rg_WE3,
    clock,
    reset
);
```

3. Try your design with the incomplete test bench below. Show *reset* and *write* features working.

```
`include "reg_file.v"

module reg_file_tb ();

    reg [4:0] rg_A1, rg_A2, rg_A3;
    reg [31:0] rg_WD3;
    wire [31:0] rg_RD1, rg_RD2;
    reg rg_WE3;
    reg clock, reset;
    integer i;

    reg_file uut (
```

```
        rg_A1, rg_A2, rg_A3, rg_WD3,
        rg_RD1, rg_RD2, rg_WE3,
        clock, reset);

    initial begin
        $dumpfile("reg_file_output_wave.vcd");
        $dumpvars(0, reg_file_tb);
    end

    initial begin
        reset = 1;
        #15 reset = 0;
    end

    initial begin
        clock = 0;
        forever #20 clock = ~clock;
    end

    \\ Reading the values from the 32 registers
    initial begin
        for (i = 0; i < 31; i = i + 2) begin
            rg_A1 = i;
            rg_A2 = i + 1;
            #20;
        end
    end

    initial begin
        #700 $finish;
    end

endmodule
```

3 Grading

This lab does not require a report. When finishing it, make sure you enter the line for a direct assessment conducted by the professors or TAs during lab timeframe.