Aeronautics Institute of Technology – ITA
Computer Science Division – IEC

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

Laboratory 05
Last document update: October 2, 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

- Implement an Arithmetic Logic Unit (ALU).

2 Lab Development

2.1 Interpreting

- Based on the HDL given below, sketch the diagram representing the circuit, for bit y[3] only.

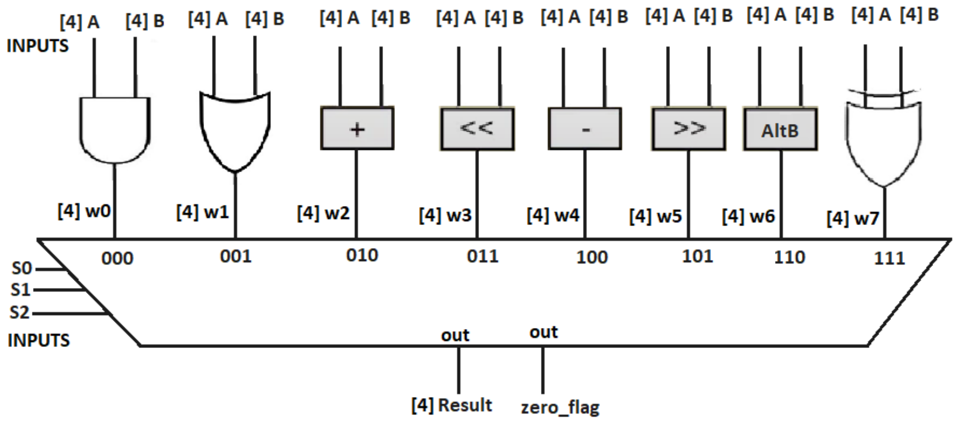
```
module ______ (
     // Inputs
     input [3:0] i0, i1, i2, i3,
     input [1:0] s,
     // Outputs
     output [3:0] y
);
     assign y[3] = i0[3] & ~s[0] & ~s[1]  |
                   i1[3] &  s[0] & ~s[1]  |
                   i2[3] & ~s[0] &  s[1]  |
                   i3[3] &  s[0] &  s[1];

     assign y[2] = i0[2] & ~s[0] & ~s[1]  |
                   i1[2] &  s[0] & ~s[1]  |
                   i2[2] & ~s[0] &  s[1]  |
                   i3[2] &  s[0] &  s[1];

     assign y[1] = i0[1] & ~s[0] & ~s[1]  |
                   i1[1] &  s[0] & ~s[1]  |
                   i2[1] & ~s[0] &  s[1]  |
                   i3[1] &  s[0] &  s[1];

     assign y[0] = i0[0] & ~s[0] & ~s[1]  |
                   i1[0] &  s[0] & ~s[1]  |
                   i2[0] & ~s[0] &  s[1]  |
                   i3[0] &  s[0] &  s[1];
endmodule
```

- The HDL represents which circuit?

- What is the main difference between this circuit and the one implemented in previous labs?
- Propose a simplified testbench just to simulate some inputs and validate the circuit operation.
- Adapt the above HDL code to obtain 8 channels of selection.
- Simulate and validate the operation of your proposed circuit. You will use it as part of your ALU.

2.2 Circuit analysis

An ALU is a combinational circuit responsible for most logic and arithmetic operations within processors. It is capable of performing operations (addition, subtraction, AND, OR, ...) on machine words that represent operands. For our project, the inputs are the operands A/B and the operation selector S0/S1/S2. The outcomes are Result and Zero flag.



A simple ALU.

```
module alu (
        src_a,
        src_b,
        operation,
        alu_result,
        zero_flag

        mux_8_1 mux ();

        assign zero_flag =
);
```

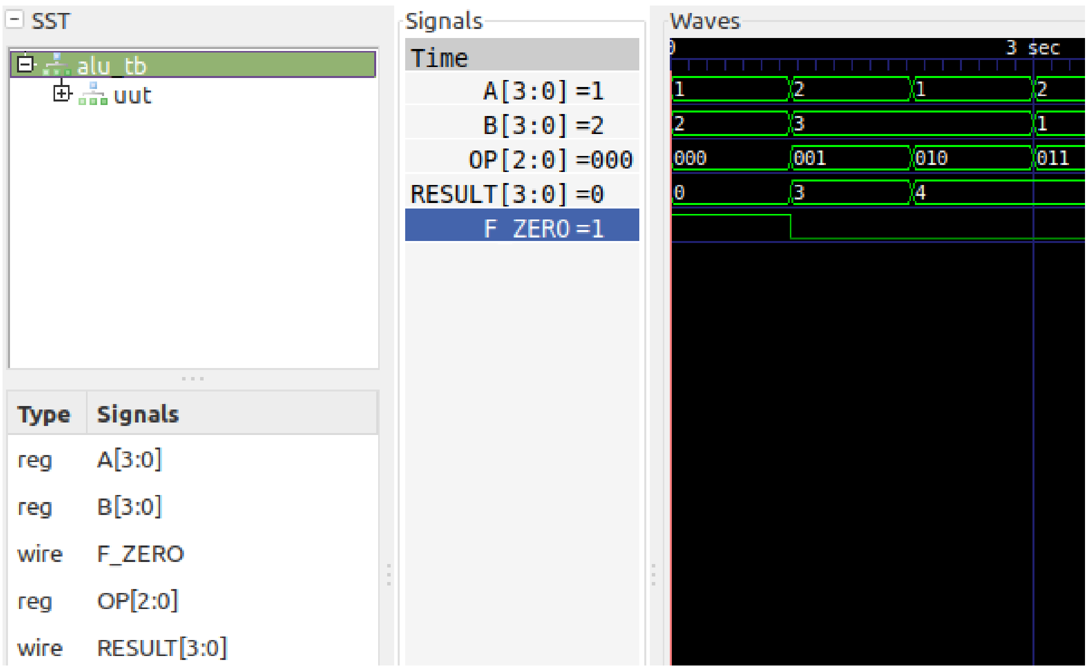
2.3 HDL coding and simulation

- Create a folder named "LAB-5".
- Based on the above incomplete HDL and diagram provided: create a file named "alu.v", this implementation has two 4-bit inputs: "src_a" and "src_b" that represent the operands, another input called "operation" that contains our 3-bit **opcode**, and output that stores the 4-bits operation results. We will also implement a flag called "zero_flag" which will be used to indicate when we get the result 0 in an operation.
- Notice that you must use the circuit obtained in the first item, to be able to select the output channel based on the requested operation.
- Use the testbench below as a basis to test all the operations in the developed ALU.

```

`include "alu.v"

module alu_tb ();
```

```

    parameter DATA_WIDTH = 4;
    parameter OPCODE_LENGTH = 3;

    reg [DATA_WIDTH - 1:0] A;
    reg [DATA_WIDTH - 1:0] B;
    reg [OPCODE_LENGTH - 1:0] OP;

    wire [DATA_WIDTH - 1:0] RESULT;
    wire F_ZERO;

    alu uut(    .src_a(A),
                .src_b(B),
                .operation(OP),
                .alu_result(RESULT),
                .zero_flag(F_ZERO)
            );

    initial begin
        A = 1;   B = 2;   OP = 4'b0000;
    end


        begin
            $dumpfile("alu_tb.vcd");
            $dumpvars(0, alu_tb);
        end

endmodule
```

- You should get a simulation similar to the image below:



Type	Signals
reg	A[3:0]
reg	B[3:0]
wire	F_ZERO
reg	OP[2:0]
wire	RESULT[3:0]

```
Time
A[3:0] =1
B[3:0] =2
OP[2:0] =000
RESULT[3:0] =0
F_ZERO =1
```

```
A       1    2    1    2
B       2    3         1
OP      000  001  010  011
RESULT  0    3    4
```

ALU simulation.

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.