**Course Project**

# Arithmetic and Logical Unit Design Project

It is required to design the ALU shown in Fig.1. This ALU can execute arithmetic and logical operations. The operation of the ALU is described by table 1. The output (arithmetic or logical) is selected by the MSB of the selection line, while the required operation is selected by the other 3 bits. It is also required to design flip-flops at the inputs and outputs of the ALU.

## Requirements

- Code for the ALU using VHDL or Verilog
- Signed i/p and o/p
- Testbench code testing all possible ALU operations
- Transistor-level/Block-diagram schematics
- Circuit-level simulations showing delay (at worst-case condition) and power consumption (at maximum possible frequency of operation)
- Transient simulations with the flip-flops included at your maximum frequency of operation showing proper sampling of the inputs and outputs.

## Bonus (2 Marks Maximum)

- Fastest and most-efficient architectures (1 Mark)
- Layout of the custom-digital implementation (1 Mark)
- FPGA implementation of the semi-custom implementation (1 Mark)

## Hints

- Use std_logic_unsigned package along with std_logic_1164 for addition/subtraction. (VHDL)
- You can use symbols in Cadence if you want to replicate any (previously made) gate many times in your design e.g inverter gate
- Use 130-nm CMOS technology.
- Use a supply of 1.2 V.
- You are free to choose any implementation style for the design, including (but not limited to): complementary CMOS, ratioed logic, DCVSL, pass-transistor logic, and dynamic logic. Feel free to mix the logic families in your design.
- Use a 2-pF load capacitance at the output of your ALU not your flip-flops.

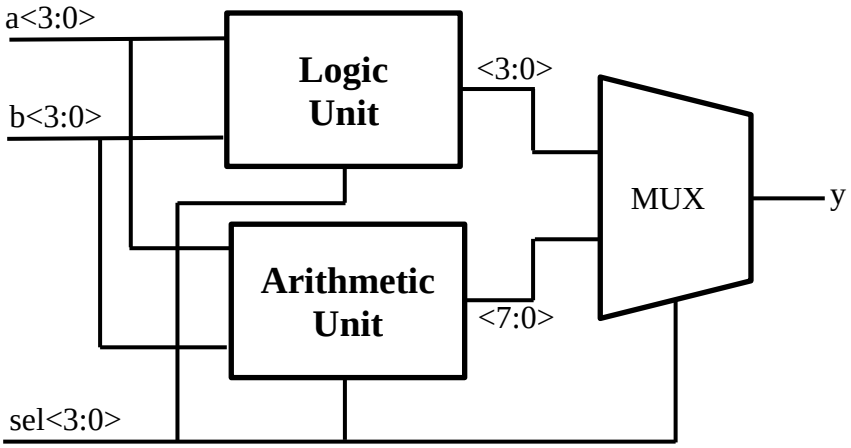


**Fig. 1: ALU Block Diagram**

**Table 1: ALU Operations**

| Sel | Operation | Unit |
|---|---|---|
| **0000** | **Increment a** | **Arithmetic** |
| **0001** | **Increment b** | |
| **0010** | **Transfer a** | |
| **0011** | **Transfer b** | |
| **0100** | **Decrement a** | |
| **0101** | **Multiply a and b** | |
| **0110** | **Add a and b** | |
| **0111** | **Subtract a and b (Assume a is always larger)** | |
| **1000** | **Complement a (1's complement)** | **Logic** |
| **1001** | **Complement b (1's complement)** | |
| **1010** | **AND** | |
| **1011** | **OR** | |
| **1100** | **XOR** | |
| **1101** | **XNOR** | |
| **1110** | **NAND** | |
| **1111** | **NOR** | |

**Delivery rules**

**Deliver one zipped file** containing all the deliverables of the project:

- A PDF file containing your code, testbench, output waveforms and all the circuits made.
- Source Code file, testbench code and the cadence schematics.

This is a group project (Number of students per group is **9 to 11**)

The project has to be submitted on the LMS

The project deadline is **19th of May at 11:59 PM.**

Finally, do not copy testbenches and reports from your colleagues. Partially or fully copied testbenches or reports get **ZERO** credit for both groups.