1 Introduction

During lab 2 and 3, we were introduced in , the ALU processor, which is a 16-bit architecture that can perform :
"Add", "Subtract", "and" "or" determined by their opcodes . It also has one-bit registers "Z", "Ovf" and "Op".
Before we implemented it using C and Python. this lab we did it using Quartus LMP form.

2 Discussion

We set two input register, then we use an add/sub. its output with the output of and, or are connected to a multiplexer.
However, and, or have also a multiplexer which determine which operation should be outputted. the second multiplexer determine between the two output.
The final answer will be saved in the output register connected to the multiplexer.

3 Results Analysis

3.1 add/sub parameters

The LPM of ADD/SUB IP core is set to take two outputs and output a result. the parameter are adjusted:

1. The Direction is set to "DEFAULT".
2. The Representation is set to unsigned.
3. The width to 4 for 2 output.
4. The pipeline is set to 0.
5. The MAXIMIZE SPEED is set to 10.
6. One input is constant is set to no.

3.2 Mux parameters

1. WIDTH: Specifies the width of the data[][] and result[] ports.
2. SIZE: Specifies the number of input buses to the multiplexer.
3. PIPELINE: Specifies the number of latency clock cycles associated with the result.

Adding two four-bit values A and B requires the use of multiple bits.
Multiple bits are grouped together and transported through a bus of the appropriate size.

4 Conclusion

During the lab 5, we designed a 4-bit adder/subtractor and a 4-bit multiplexer using the LPM component LP add/sub and Simulated the design and the ALU design.