Concordia University

Lab 3

Designing a Combinational Sign-Magnitude to Two's Complement Converter

COEN 313

Lab Section: FK-X

Digital System Design II

Penoelo Thibeaud

40212017

Lab Instructor: Ted Obuchowicz

Performed on : November 3, 2023

Submitted on : November 17, 2023

I certify that this submission is my original work and meets the Faculty’s Expectations of Originality

# Objective:

The aim of this lab is to design a combinational circuit using VHDL that converts a 4-bit input representing a signed integer in sign-magnitude notation to its equivalent 4-bit two's complement representation.

# Procedure (Methods):

1. Algorithm and Hardware Description:
   1. An algorithm was described that conditionally negates the magnitude bits, adds "001", and preserves the sign bit for negative numbers.
   2. The hardware consists of a 3-bit wide inverter, a 3-bit parallel adder, and a 2-1 3-way multiplexer.
2. VHDL Design:
   1. A VHDL entity named converter was designed with an input port sign\_mag for the sign-magnitude value and an output port twos\_comp for the two's complement value.
   2. Hints were provided to use vector slices and concatenation operators to manipulate bits.
3. Compiler Warnings and Synthesis Reports:
4. The importance of addressing compiler warnings and synthesis reports was emphasized, as these can indicate potential issues in the design.
5. Requirements and Questions:
6. The lab required Modelsim simulation results, RTL schematic diagrams, synthesis and implementation log files, VHDL code, and a demonstration on an FPGA Results and Discussion:

## Questions:

**1.** What will result (during synthesis) if a signal appears on both sides of the signal assignment operator (<=) within a combinational VHDL process such as:

library IEEE;

use IEEE.std\_logic\_1164.all;

entity is\_this\_good\_ou\_mauvaise is

port( mick : in std\_logic;

end;

keith : out std\_logic);

architecture rtl of is\_this\_good\_ou\_mauvaise is

signal stone : std\_logic;

begin

process(mick)

begin

stone <= mick and stone;

end process ;

keith <= stone ; -- tout le monde sais que Keith == stone

end ;

**2.** What will happen during simulation if a signal is read from within a combinational process but does not appear in the process sensitivity list?

**3.** If you made use of variable in your combinational process, rewrite the VHDL code such that the process makes use of only signals. If you originally made use of only signals, rewrite your VHDL code such that it makes use of variable(s). Simulate your new VHDL code to show that it gives the same simulation results. You do not have to re-synthesize. Comment on the salient dif- ferences between the code which uses only signals and the code which makes use of a variable.

# Conclusions: