Laboratory 3

(Due date: October 21st)

OBJECTIVES

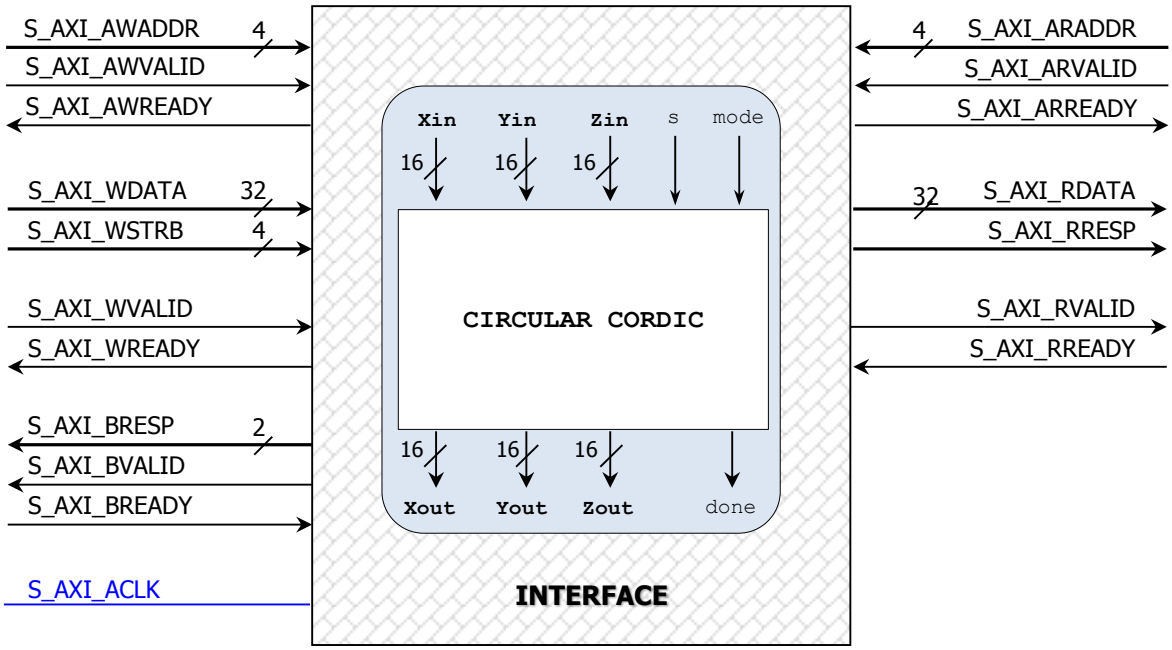
- ✓ Design an AXI4-Lite Interface for a custom VHDL peripheral.
- ✓ Integrate the custom VHDL peripheral in an embedded system in Vivado.
- ✓ Create a software application in SDK that can handle the custom peripheral.

VHDL CODING

- ✓ Refer to the Tutorial: VHDL for FPGAs for a tutorial and a list of examples.
- ✓ Refer to the Tutorial: Embedded System Design for Zynq SoC for information on how to create AXI interfaces for custom peripherals as well as embedded system integration in Vivado.

FIRST ACTIVITY (100/100)

- Using Vivado, create an AXI4-Lite Interface for the CIRCULAR CORDIC that you developed in Lab 2: Use the same format ([16 14]) for the inputs and outputs.
- AXI4-Lite Interface: Use as many Slave Registers as you wish. Your interface will most likely require a Finite State Machine. Draw a schematic of the circuit (including the FSM) that constitutes your AXI4-Lite Interface.



- Once you have your custom AXI4-Lite Peripheral, integrate it into an embedded system using the Block-Based Design approach in Vivado.

- SDK Software application: test it for the following cases by printing the hexadecimal results on the terminal (via UART).
  - ✓ Rotation Mode: $x_0 = 0, y_0 = 1/A_n, z_0 = \pi/6$.
  - ✓ Rotation Mode: $x_0 = 0, y_0 = 1/A_n, z_0 = -\pi/3$.
  - ✓ Vectoring Mode: $x_0 = y_0 = 0.8, z_0 = 0$
  - ✓ Vectoring Mode: $x_0 = 0.5, y_0 = 1, z_0 = 0$

- Download the hardware bitstream on the ZYNQ SoC.
- Launch your software application on the Zynq PS. The program should display the output results on the Terminal. **Demonstrate this to your instructor**.

- Submit (as a .zip file) the generated files: VHDL code, .c files to Moodle (an assignment will be created). DO NOT submit the whole Vivado Project.

Instructor signature: _____ **Date:** _____