

Laboratory 4

(Due date: November 2nd)

OBJECTIVES

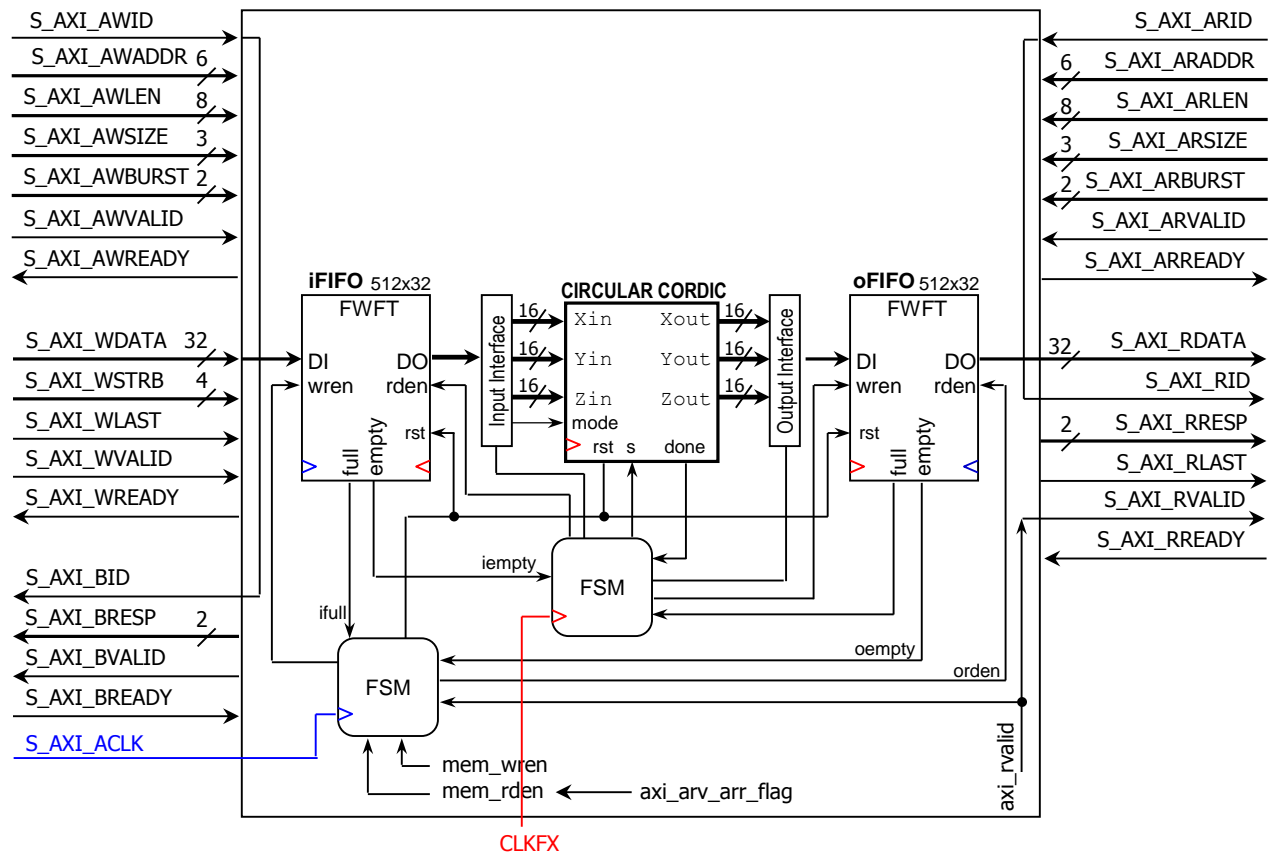
- ✓ Design an AXI4-Full Interface for a custom VHDL peripheral.
- ✓ Integrate the custom VHDL peripheral in an embedded system in Vivado.
- ✓ Create a software application in SDK that can handle the custom peripheral.

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-Full Interface for the iterative CIRCULAR CORDIC that you developed in Lab 2: Use the same format ([16 14]) for the inputs and outputs.
- AXI4-Full Interface: You can use the iFIFO/oFIFO approach used for the Pixel Processor (See Notes – Unit 5):
 - ✓ FSM @ [S_AXI_ACLK](#): It is the same as the one used for the Pixel Processor.
 - ✓ FSM @ [CLK_FX](#): This FSM controls the Input and Output interfaces to the FIFOs as well as FIFOs' signals.
 - ✓ Input and Output Interfaces: Since AXI bus size is 32 bits wide, we need to properly route data in and out of the CORDIC hardware that requires more than 32 bits. This circuit runs @ [CLK_FX](#).
- Draw a schematic of the circuit that runs at FSM @ [CLK_FX](#) (Input/Output Interfaces to the FIFO and the FSM). Connect [CLK_FX](#) to [S_AXI_CLK](#).
- If you decide to use a different approach, provide a detailed schematic of your AXI4-Full interface.



- Once you have your custom AXI4-Full 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 writing all the input data (one right after the other), and then retrieving all the output data. Print the results (via UART).
 - ✓ Rotation Mode: $x_0 = 0, y_0 = 1/A_n, z_0 = -5\pi/8$.
 - ✓ Rotation Mode: $x_0 = 0, y_0 = 1/A_n, z_0 = \pi/4$.
 - ✓ Rotation Mode: $x_0 = 0, y_0 = 1/A_n, z_0 = -\pi/2$.
 - ✓ Rotation Mode: $x_0 = 1/A_n, y_0 = 0, z_0 = -\pi/7$.
 - ✓ Vectoring Mode: $x_0 = y_0 = 0.125, z_0 = 0$
 - ✓ Vectoring Mode: $x_0 = 0.35, y_0 = -0.35, z_0 = 0$
 - ✓ Vectoring Mode: $x_0 = 0.5, y_0 = 0.3, z_0 = 0$
 - ✓ Vectoring Mode: $x_0 = 0.5, y_0 = -0.3, z_0 = 0$
- 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: _____