DESIGN SPEC DOCUMENT

ECE-593: Fundamentals of Pre-Silicon Validation

Maseeh College of Engineering and Computer Science

Winter, 2025

Project Name: Asynchronous FIFO

Members:

Jyothsna Tallada

Krishnavardhan Raju

Poojith Pogarthi

Sri Chandana Reddy Vanukuri

Date: 20/04/2025

Project Name	Asynchronous FIFO
Location	Portland
Start Date	20/04/2025
Estimated Finish Date	25/05/2025
Completed Date	

Prepared by: Team Number 11	
Prepared for: Prof. Venkatesh Patil	
Team Member Name	Email
Jyothsna Tallada	jtallada@pdx.edu
Krishnavardhan Raju Godugu	krgodugu@pdx.edu
Poojith Pogarthi	poojith@pdx.edu
Sri Chandana Reddy Vanukuri	vanukuri@pdx.edu

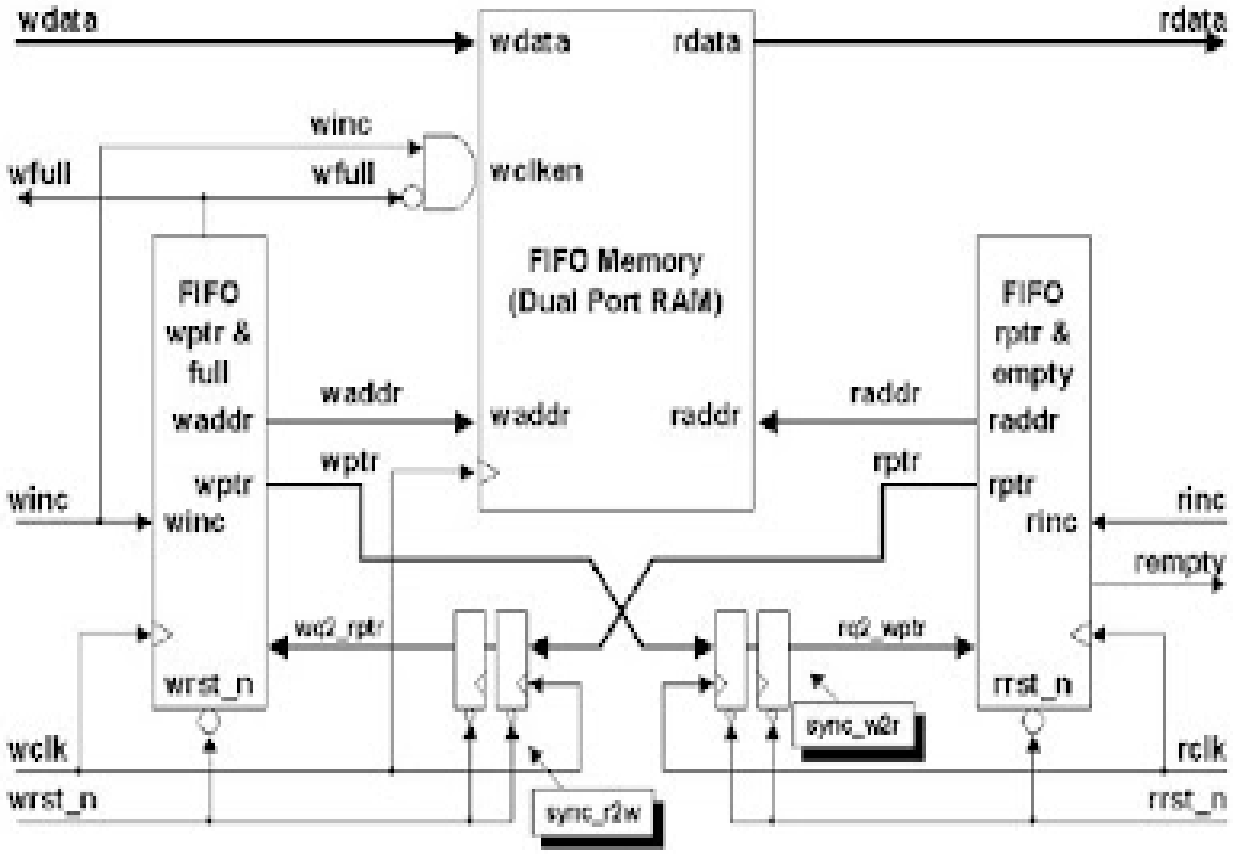
Design Features:
Two Clock Domains
FIFO Depth and Width Configurable
Data Buffering
Full and Empty Flags
Pointer Synchronization
Asynchronous Reset

Project Description:
This project involves designing an asynchronous FIFO using SystemVerilog with separate read and write clocks. The FIFO allows reliable data transfer between different clock domains using Gray code pointer synchronization. It supports configurable data width and depth, along with full and empty status flags. An asynchronous reset initializes the system safely. A simple SystemVerilog testbench is used to verify functionality by testing read/write operations, data integrity, and flag behavior.

Important Signals/Flags
clk_w and clk_r – Write clock and Read clock (Input) rst – reset (Input) wr_en and rd_en – Write enable signal and Read enable signal (Input) wdata – Data input to FIFO (Input) rdata – Data output from FIFO (Output) FULL – FIFO is full, no more writes allowed (flag) EMPTY – FIFO is empty, no data to read (flag) overflow – Indicates write attempted when FIFO is full (flag) underflow - Indicates read attempted when FIFO is empty (flag)

Design Signals
wptr and rptr – write and read pointers wptr_g and rptr_g – grey code write pointer and grey code read pointer wr_toggle and rd_toggle - write and read toggle wptr_rd_clk and rptr_wr_clk – write pointer read clock and read pointer write clock rd_toggle_wr_clk and wr_toggle_rd_clk – read toggle write clock and write toggle read clock

Block Diagram



References/Citations
https://asic-soc.blogspot.com/2007/12/new-asynchronous-fifo-design.html