DESIGN SPEC DOCUMENT

ECE-593: Fundamentals of Pre-Silicon Validation
Maseeh College of Engineering and Computer Science
Winter, 2025

Project Name: Team12_Asynchronous FIFO_S25_ECE593
Members: Aravindh Nanjaiya Latha,Rishi Gunda,Rohit Bonigala, Shreya Umesh Shetty
Date: 20th April 2025

Project Name	Team12_Asynchronous FIFO_S25_ECE593
Location	Portland,Oregon
Start Date	8th April 2025
Estimated Finish Date	26th May 2025
Completed Date	

Prepared by: Team Number	
Prepared for: Prof. Venkatesh Patil	
Team Member Name	Email
Aravindh Nanjaiya Latha	aravindh@pdx.edu
Rishi Gunda	rgunda@pdx.edu
Rohit Bonigala	rohit@pdx.edu
Shreya Umesh Shetty	shreyau@pdx.edu

Design Features:	
FIFO Depth	Configurable, default 8 entries (parameter DEPTH=8).
Data Width	Configurable, default 8 bits (parameter DATA_WIDTH=8).
Pointer Width	Dynamically calculated as PTR_WIDTH=$clog2(DEPTH), default 3 bits for 8 entries, plus an extra bit for full/empty detection.
Clock Domains	Asynchronous write (wclk) and read (rclk) clocks for independent operation.

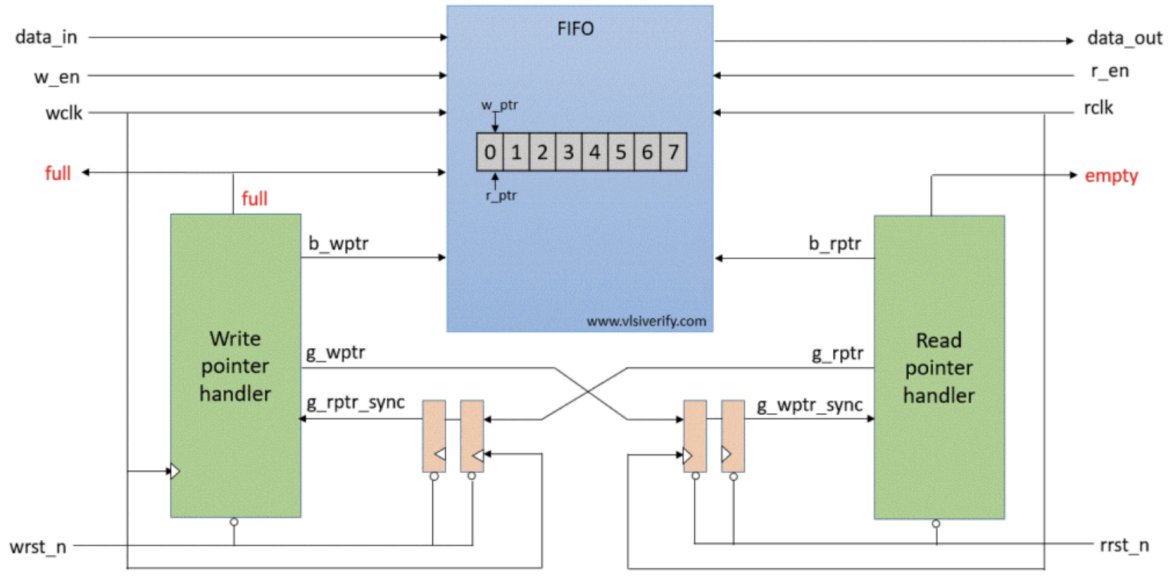
Synchronization	Gray code pointers synchronized across domains using two-stage flip-flop synchronizers to prevent metastability.
Full Flag	Indicates FIFO cannot accept writes (set when write pointer catches synchronized read pointer with wrap-around).
Empty Flag	Indicates no data to read (set when read pointer equals synchronized write pointer).
Reset	Active-low synchronous resets (wrst_n for write domain, rrst_n for read domain).
Memory Type	Dual-port RAM as a register array, supporting simultaneous write (on wclk) and combinational read.
Write Control	Write pointers (b_wptr, g_wptr) incremented when w_en is high and FIFO is not full.
Read Control	Read pointers (b_rptr, g_rptr) incremented when r_en is high and FIFO is not empty.
Implementation	SystemVerilog with modular design: wptr_handler (write pointer, full flag), rptr_handler (read pointer, empty flag), fifo_mem (data storage), synchronizer (pointer synchronization).

Project Description:
The Asynchronous FIFO (First-In-First-Out) design is a critical component for data transfer between two clock domains operating at different frequencies. This project aims to design, implement, and validate an asynchronous FIFO module for use in pre-silicon validation. The FIFO will handle data buffering, ensuring reliable data transfer across asynchronous clock domains while preventing data loss, metastability. The design will include separate write and read ports, each operating in its respective clock domain, with synchronization mechanisms to manage pointer updates. For the first milestone, the focus is on developing the basic design architecture, including the FIFO memory, read/write pointers, synchronization logic, and status flags. The design will be implemented in System Verilog, targeting a depth of 8 entries and a data width of 8 bits. Subsequent milestones will involve developing a class-based testbench using SystemVerilog for comprehensive verification.

Important Signals/Flags
wr_clk
rd_clk
rst_n
wr_en
rd_en
data_in
data_out
full
empty

Design Signals
wr_clk
rd_clk
rst_n
wr_en
rd_en
data_in
data_out
full
empty

Block Diagram



Asynchronous FIFO

References/Citations
https://vlsiverify.com/verilog/verilog-codes/asynchronous-fifo/ https://www.researchgate.net/publication/383129354_Design_and_implementation_of_asynchronous_FIFO