DESIGN SPECIFICATION DOCUMENT

ECE-593: Fundamentals of Pre-Silicon Validation
Maseeh College of Engineering and Computer Science
Winter, 2026

Project Name: Design, Implementation, and Verification of Asynchronous FIFO

Members: Monesh Kareti, Senkathir Mutharasu

Date: 01/28/26

Project Name	Design, Implementation, and Verification of Asynchronous FIFO
Location	Portland, Oregon, United States of America
Start Date	Jan 15th 2026
Estimated Finish Date	March 15th, 2026
Completed Date	

Prepared by: 10	
Prepared for: Prof. Venkatesh Patil	
Team Member Name	Email
Monesh Kareti	monesh@pdx.edu
Senkathir Mutharasu	senkmuth@pdx.edu

Design Features:

We have two modules operating at different clock frequencies:

- **Sender (Module A): 500 MHz (fA)**
- **Receiver (Module B): 500 MHz (fB)**

Write and Read Conditions:

- **Writing:** After every write operation, Module A writes continuously (write idle cycles = 0). This means data is written once every 1 clock cycle.
- **Reading:** After each read operation, Module B waits 4 clock cycles before reading the next data. This means data is read once every 5 clock cycles.

Timing Calculations:

- **Time to write one data item:** 1 × 2 ns = 2 ns
- **Time to write an entire burst (450 data items):** 450 × 2 ns = 900 ns
- **Time to read one data item:** 5 × 2 ns = 10 ns
- **Total data read in 900 ns:** 900 ÷ 10 = 90 data items
- **Data remaining in FIFO:** 450 − 90 = 360

FIFO Depth Requirement:

Since the receiver is slower than the sender, some data will accumulate in the FIFO. To ensure smooth operation, the minimum FIFO depth should be at least 360 entries to store the extra data.

So we take the depth to be **512**, which is 2^9.

Project Description:
This project focuses on the design and verification of an Asynchronous FIFO (First-in First-Out) buffer. The FIFO is a critical digital design component used for safe data transfer between two independent clock domains. The primary goal is to ensure reliable communication by preventing data loss, duplication, and metastability when the write and read clocks operate at different frequencies or phases

Important Signals/Flags:

Write-side interface

- wclk – Write clock.
- wrst_n – Active-low reset for write domain.
- wr_en – Write enable; when high and not full, data is written to FIFO.
- wr_data[DATA_WIDTH-1:0] – Data input from sender.

Read-side interface

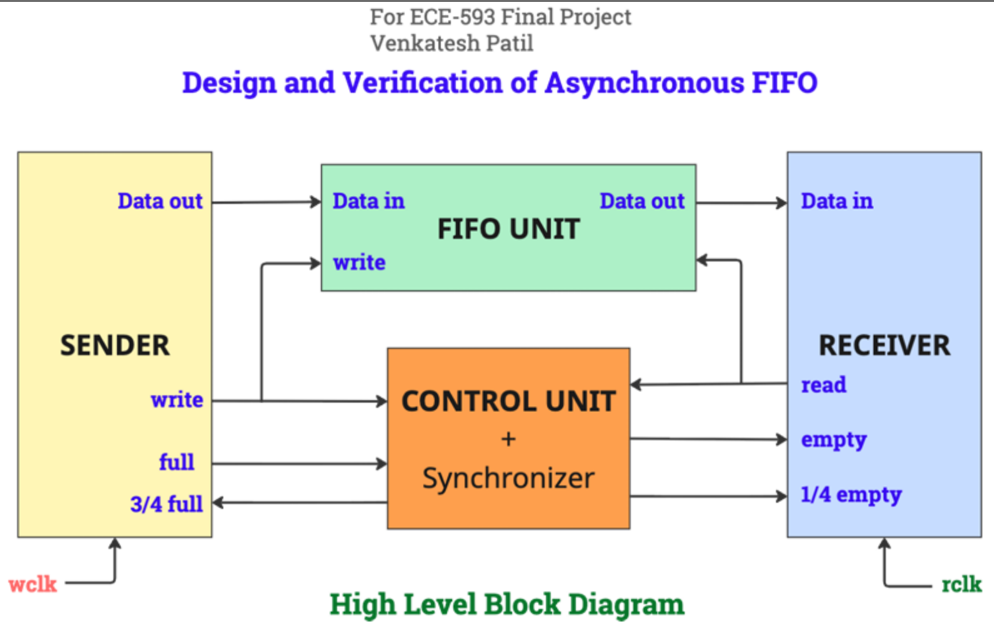
- rclk – Read clock.
- rrst_n – Active-low reset for read domain.
- rd_en – Read enable; when high and not empty, data is read from FIFO.
- rd_data [DATA_WIDTH-1:0] – Data output to receiver.

Status flags

- full – FIFO cannot accept more data.
- empty – FIFO has no valid data.
- almost_full – FIFO occupancy above 3/4 of depth. (to be implemented)
- almost_empty – FIFO occupancy below 1/4 of depth (to be implemented)

Design Signals

wr_en: write enable
wr_data: write data
full: FIFO is full
empty: FIFO is empty
rd_en: read enable
rd_data: read data
b_wptr: binary write pointer
g_wptr: gray write pointer
b_wptr_next: binary write pointer next
g_wptr_next: gray write pointer next
b_rptr: binary read pointer
g_rptr: gray read pointer
b_rptr_next: binary read pointer next
g_rptr_next: gray read pointer next
b_rptr_sync: binary read pointer synchronized
b_wptr_sync: binary write pointer synchronized

Block Diagram


References/Citations
https://vlsiverify.com/verilog/verilog-codes/asynchronous-fifo/#google_vignette