DESIGN SPEC DOCUMENT

ECE-593: Fundamentals of Pre-Silicon Validation

Maseeh College of Engineering and Computer Science
Winter, 2025

Project Name:
Design and UVM Verification of Asynchronous FIFO in SystemVerilog

Members:
Aakash Siddharth
Sai Ganesh Reddy Charian
Satyajit Deokar
Siddesh Patil
Date: 01/31/2024

Project Name	Design and UVM Verification of Asynchronous FIFO in SystemVerilog
Location	https://github.com/Satya7733/Asyc_FIFO
Start Date	Jan 26, 2025
Estimated Finish Date	Mar 1, 2025
Completed Date	–

Prepared by: Team Number 5	
Prepared for: Prof. Venkatesh Patil	
Team Member Name	Email
Aakash Siddharth	aakashbh@pdx.edu
Sai Ganesh Reddy Charian	saiganes@pdx.edu
Satyajit Deokar	satyajit@pdx.edu
Siddesh Patil	patilsid@pdx.edu

Github link (Testing Branch): https://github.com/Satya7733/Asyc_FIFO.git

Design Features	
1	Theoretically unbounded, but in practice, limited by the simulator and hardware constraints supports data widths upto [2^64 -1] bits; Parameterized by ‘DATASIZE’
2	Theoretically unbounded, but in practice, limited by the simulator and hardware constraints supports memory depths upto [2^64 -1] bits; Parameterized by ‘DEPTH’
3	Fully synchronous, and independent clock domains for the Read and Write ports
4	Supports FULL and EMPTY status flags
5	Two optional output signals for indicating Write-Full and Read-Empty of the memory port.

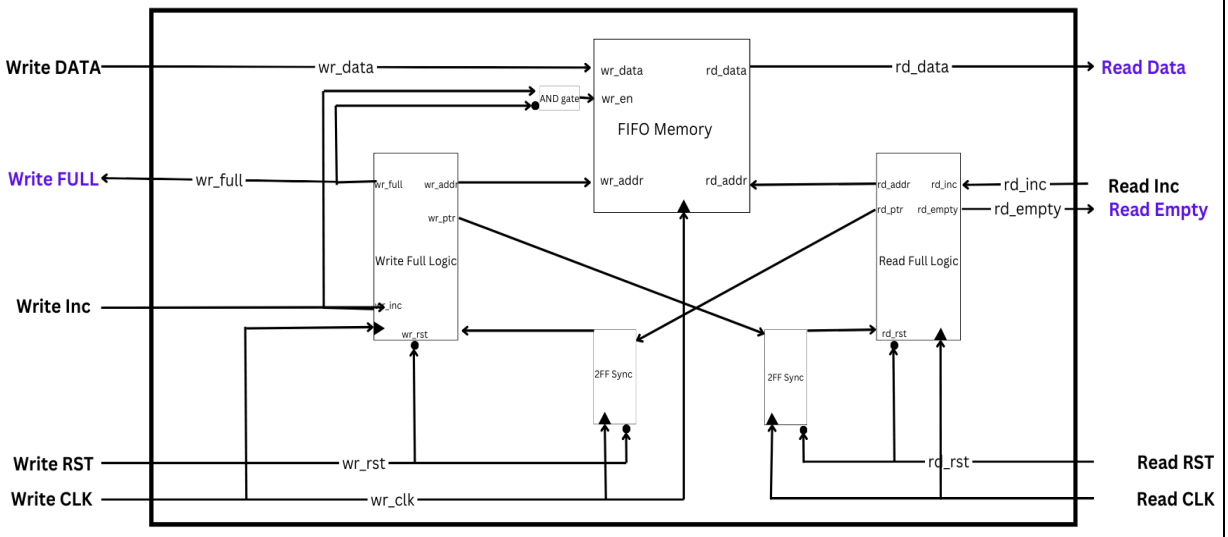
Project Description
Asynchronous FIFO, is a type of FIFO buffer in which separate clock domains govern the read and write operations. This indicates that the clocks that drive the writing and reading processes are not synced. Data is reliably transferred between these asynchronous clock zones using async FIFOs. Read and write operations are controlled by different clock zones in an asynchronous FIFO. The next word to be written is always shown by the write pointer. The write pointer is increased to point to the subsequent location to be written once the memory location it points to is written during a FIFO-write operation. The current FIFO word to be read is also always pointed to by the read pointer. Both points are set to zero upon reset. The read pointer, which is still addressing the contents of the first FIFO memory word, instantly pushes the first valid word onto the FIFO data output port so that the receiver logic can read it. This happens when the write pointer increments and the empty flag is cleared after the first data word is written to the FIFO. When the write and read pointers are equal, the FIFO is empty; when the write pointer has wrapped around and caught up to the read pointer, the FIFO is full.

Important Signals				
No	Signal name	Width	Purpose	Description
Memory Flags				
1	rd_empty	1	Read Empty	The FIFO is empty when the read and write pointers are both equal. This condition happens when both pointers are reset to zero during a reset operation, or when the read pointer catches up to the write pointer, having read the last word from the FIFO.
2	wr_full	1	Write Full	The FIFO is full when the write pointer has wrapped around and caught up to the read pointer. This means that the write pointer has incremented past the final FIFO address and has wrapped around to the beginning of the FIFO memory buffer.

Design Signals				
Memory Signals				
No	Design Signals	Width	Purpose	Description
1	rd_ptr	‘ASIZE+1’ =5	Read Pointer	This signal points the memory location to read that is read from the FIFO. It has a width of ASIZE+1 bits.
2	wr_ptr	‘ASIZE+1’ =5	Write Pointer	This signal points the memory location to write that is wrote from the FIFO. It has a width of ASIZE+1 bits.
3	rd_addr	‘ASIZE’=4	Read Address	This signal is the memory location to read that is read from the FIFO. It has a width of ASIZE bits.
4	wr_addr	‘ASIZE’=4	Write Address	This signal is the memory location to write that is wrote into the FIFO. It has a width of ASIZE bits.
Input Signals				
3	wr_data	‘DSIZE’=8	Write Data	This signal represents the data that is written into the FIFO. It has a width of DSIZE bits.
4	wr_clk	1	Write Clock	The clock signal for the write side of the FIFO, used to synchronize write operations.
5	wr_rst	1	Write Reset	A reset signal that resets the write-side

				logic of the FIFO when asserted.
6	rd_rst	1	Read Reset	A reset signal that resets the read-side logic of the FIFO when asserted.
7	rd_inc	1	Read Enable	A control signal that triggers a read operation when asserted.
8	wr_inc	1	Write Enable	A control signal that triggers a write operation when asserted.
9	rd_clk	1	Read Clock	The clock signal for the read side of the FIFO, used to synchronize read operations.
Output Signals				
10	rd_data	‘DSIZE’=8	Read Data	The data that is read from the FIFO. It has the same width as the write data (DSIZE bits).
11	rd_empty	1	Read Empty	A flag that indicates whether the FIFO is empty. If asserted, no data is available for reading.
12	wr_full	1	Write Full	A flag that indicates whether the FIFO is full. If asserted, no additional data can be written until space becomes available.

Block Diagram



References/Citations

1. https://youtu.be/0LVHPRmi88c?si=41ywg6sBar8KMudG
2. Microsoft Word - CummingsSNUG2002SJ_FIFO1_rev1_2.doc
3. Asynchronous FIFO - VLSI Verify

4. https://docs.amd.com/r/2021.1-English/ug953-vivado-7series-libraries/XPM_FIFO_ASYNC
5. ujjwal-2001/Async_FIFO_Design: This projects contains Veriolg code and timing analysis of a asynchronous FIFO. The README.md document is maintained, which explains every aspects of the code.