FIFO

Single-Port Memory (4-bit × 256)

Objective

Design and simulate a **synchronous single-port memory** that can perform **read or write** operations using separate enable signals. The memory must use the same address bus for both operations.

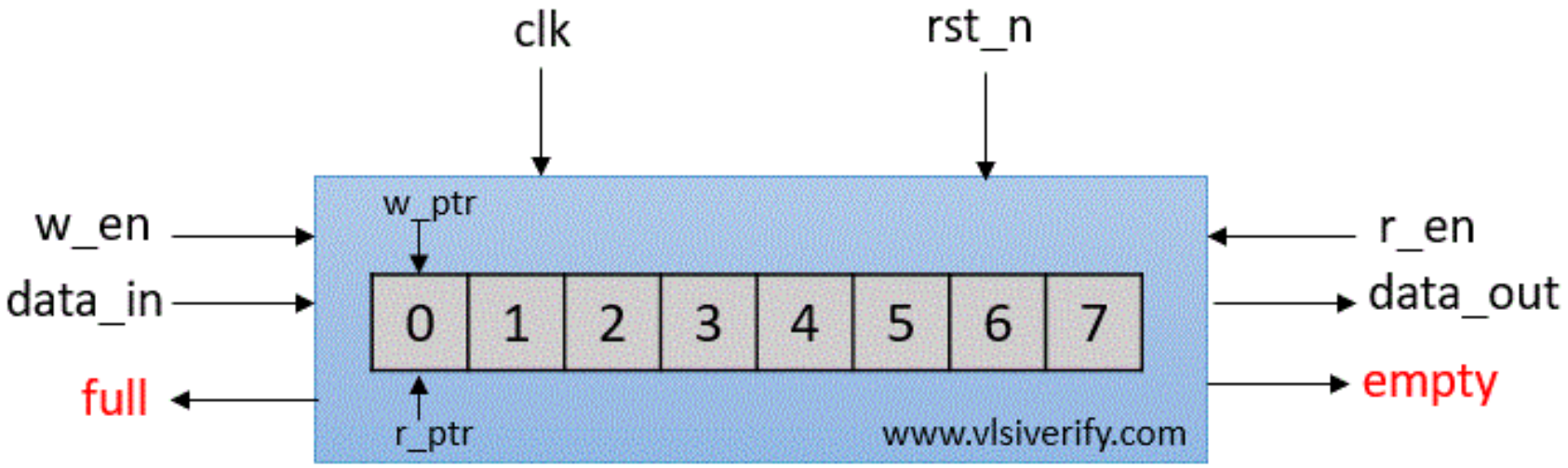
Functional Specifications

Item	Description
Data width	4 bits (`data_in` , `data_out`)
Address width	8 bits → 256 memory locations
Clock	Single synchronous clock (`clk`)
Reset	Active-high, synchronous reset clears all locations and output
Write enable (`wr_en`)	When 1, writes `data_in` to the location pointed by `addr`
Read enable (`rd_en`)	When 1 and `wr_en` = 0, reads data from `addr` into `data_out`
Simultaneous read & write	Not allowed (single port). If both `wr_en` and `rd_en` are 1, **write has priority**

Problem Statement — Synchronous FIFO Design and Verification

Objective

Design, write, and simulate a **Synchronous FIFO (First-In, First-Out) memory** using Verilog HDL.
The FIFO should support **4-bit data width** and **8-bit address lines** (total depth = 256 locations).
The design must allow **synchronous read and write** operations using a **single clock** and should indicate **full** and **empty** status flags.



Synchronous FIFO

Specifications

1. **Data width:** 4 bits
2. **Address width:** 8 bits (→ 256 memory locations)
3. **Clock:** Single synchronous clock for read and write
4. **Reset:** Active-high, synchronous reset (clears pointers, count, and output)

5. **Write operation:**
    - Occurs when `wr_en = 1` and FIFO is **not full**
    - Data is written to the location pointed by `wr_ptr`
6. **Read operation:**
    - Occurs when `rd_en = 1` and FIFO is **not empty**
    - Data from the location pointed by `rd_ptr` is output on `dout`
7. **Pointer logic:**
    - `wr_ptr` and `rd_ptr` increment after successful write/read
    - Both wrap around after the last address (circular buffer behavior)
8. **Flags:**
    - `full = 1` when all memory locations are filled
    - `empty = 1` when no valid data is present in FIFO
9. **Counter:** Tracks the number of elements currently stored.

Cycle	Action	FIFO Contents	wr_ptr	rd_ptr	full	empty
0	Reset	[- , - , - , -]	0	0	0	1
1	Write 0x11	[11, -, -, -]	1	0	0	0
2	Write 0x22	[11, 22, -, -]	2	0	0	0
3	Read → gets 0x11	[*, 22, -, -]	2	1	0	0
4	Write 0x33	[*, 22, 33, -]	3	1	0	0
5	Write 0x44	[*, 22, 33, 44]	0	1	1	0
6	Read → gets 0x22	[*, *, 33, 44]	0	2	0	0