

# Asynchronous FIFO

Name: Arnav Parkhi

Roll no.: 23M1138

An **Asynchronous FIFO (First-In, First-Out)** is a type of FIFO buffer that is used to transfer data between two systems or components that operate on different clock domains. This means that the data producer (writing to the FIFO) and the data consumer (reading from the FIFO) are running at different clock speeds or may not be synchronized at all. Asynchronous FIFOs are crucial in scenarios where you need to safely and efficiently pass data between these unsynchronized parts of a system.

## 1. Clock Domains:

- The write operations occur in the "write clock domain" and the read operations occur in the "read clock domain".
- These two clock domains can operate at different frequencies and can have arbitrary phase relationships.

## 2. Dual-Port Memory:

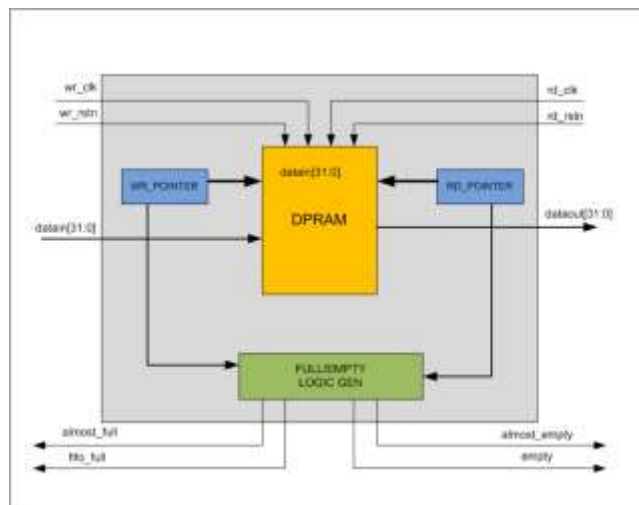
- Asynchronous FIFOs typically use dual-port memory, which allows simultaneous read and write operations from different clock domains.
- One port is used for writing data, and the other port is used for reading data, allowing the two operations to occur independently.

## 3. Read and Write Pointers:

- The FIFO uses two pointers: a write pointer and a read pointer.
- The write pointer advances when data is written to the FIFO, and the read pointer advances when data is read.

## 4. Full and Empty Flags:

- The FIFO generates "full" and "empty" status flags to indicate whether the FIFO is full or empty.
- These flags help prevent data corruption by stopping writes when the FIFO is full and reads when the FIFO is empty.

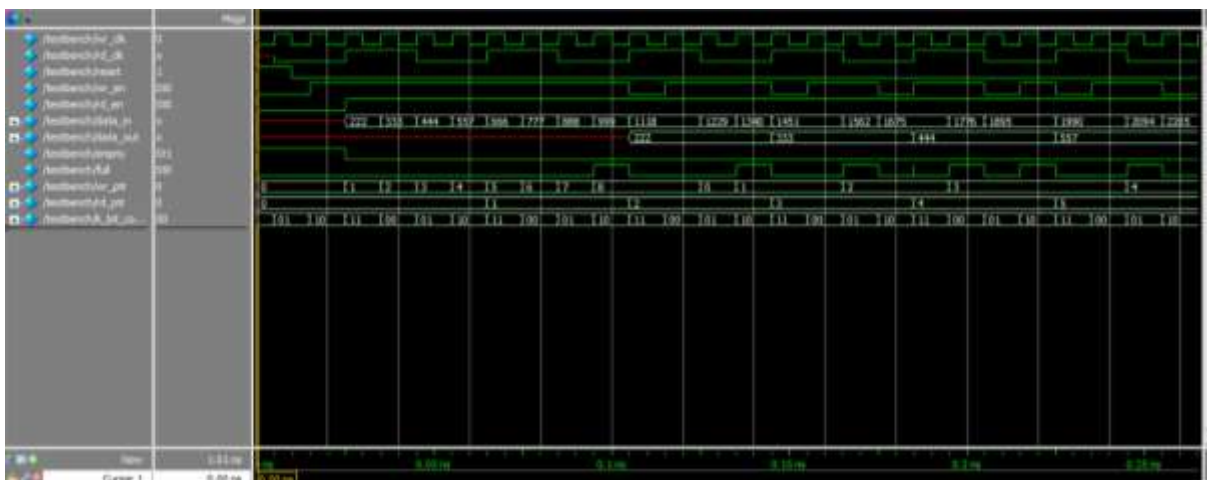




Read Pointer: 1, Data-out: 0100111001



Read Pointer: 5, Data-out: 0100110001



Simulated on ModelSim