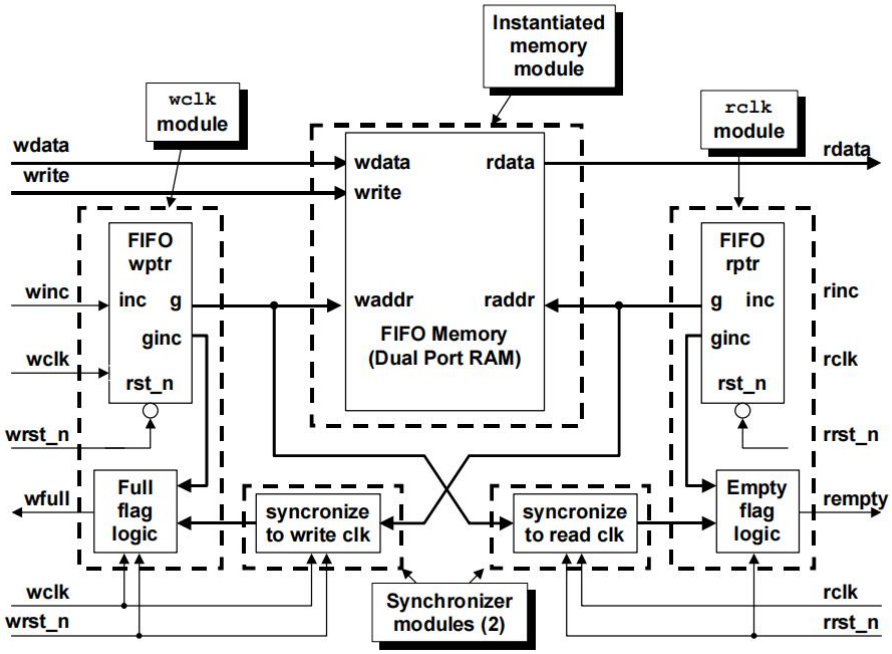
Report on
Dual Clock Asynchronous FIFO
By: Yogesh Kumar

Introduction:
The dual clock asynchronous FIFO is a critical component in digital systems, enabling data transfer between two clock domains with different frequencies. This report presents the design principles and operation of a dual clock asynchronous FIFO, addressing the challenges of clock domain crossing. The design incorporates dual synchronizers to ensure reliable data transfer and prevent metastability issues during asynchronous data exchanges.

Purpose and Motivation:
The primary purpose of this design is to establish a reliable data transfer mechanism between two clock domains with distinct frequencies. The motivation behind this project lies in the need to prevent metastability issues and ensure data integrity during asynchronous data exchanges.

Dual Clock Asynchronous FIFO Architecture:
The dual clock asynchronous FIFO comprises two main sections: the write (producer) side and the read (consumer) side. Each section operates with its independent clock signal, referred to as Clock Write (wclk) and Clock Read (rclk), respectively.
The FIFO is configured as a 16x8 memory bank, providing storage for up to 16 data entries, each being 8 bits wide.
Key components include:

- Write pointer (wptr) and read pointer (rptr) for tracking data positions within the FIFO.
- Data input (wdata) and data output (rdata) ports for seamless data transfer.
- Status signals indicating FIFO full (wfull) and FIFO empty (rempty) conditions.
- Control signals for enabling write (winc) and read (rinc) operations.



FIFO Design Implemented in Verilog

FIFO Operation:

The dual clock asynchronous FIFO operates as follows:

- The producer side (using wclk) writes data into the FIFO when it is not full, controlled by the write-enable (winc) signal.
- The consumer side (using rclk) reads data from the FIFO when it is not empty, controlled by the read-enable (rinc) signal.
- The write pointer (wptr) increments upon data writing, and the read pointer (rptr) increments upon data reading to maintain orderly data flow.
- Status signals (wfull and rempty) update based on the FIFO's occupancy level.

Synchronization Techniques:

Asynchronous FIFOs require synchronization techniques to handle data transfer between clock domains. Some commonly used methods include:

Gray Code: Implementing Gray code counters to prevent metastability during pointer synchronization.

Gray-to-Binary Conversion: Converting Gray code pointers to binary format for proper read and write operations.

Handshaking: Utilizing handshaking protocols between clock domains to ensure data integrity.

Double Synchronizer: A double synchronizer can be used to safely transfer data from the write (source) clock domain to the read (destination) clock domain. Here's how the double synchronizer technique is typically implemented:

Write (Source) Clock Domain: The data to be transferred is driven by the source clock domain. A single flip-flop is used as the first synchronizer. It captures the data and transfers it to the next clock domain.

Between Clock Domains: The output of the first synchronizer is passed through the logic boundary (where the two clock domains meet) to the second synchronizer.

Read (Destination) Clock Domain: The second synchronizer, located in the destination clock domain, captures the output of the first synchronizer. The output of the second synchronizer is considered safe to use in the destination clock domain.

This arrangement increases the likelihood of capturing the correct data value in the destination clock domain and provides an additional layer of protection against metastability. By introducing two levels of synchronization, the double synchronizer technique helps reduce the probability of erroneous data transfers and related issues.

Advantages of Dual Clock Asynchronous FIFO with Dual Synchronizers:

The dual clock asynchronous FIFO design offers several advantages, particularly when augmented with dual synchronizers:

- Facilitates data transfer between clock domains with different frequencies.
- Prevents metastability issues through the implementation of dual synchronizers.
- Ensures reliable and orderly data transfer without data loss.

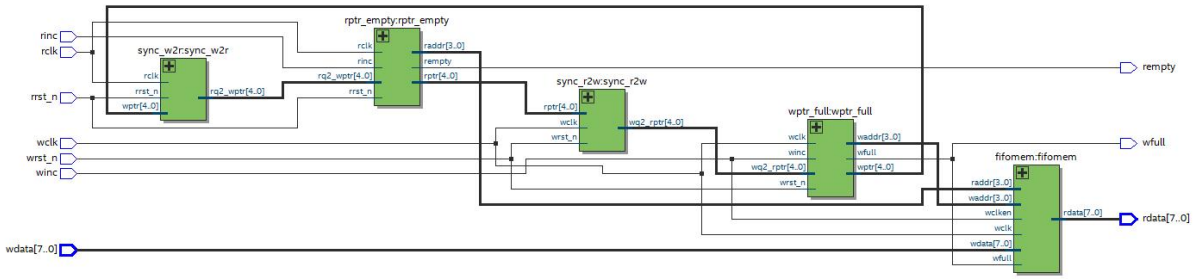
Conclusion:

In conclusion, the design of a dual clock asynchronous FIFO with dual synchronizers is crucial for seamless data transfer in digital systems with multiple clock domains. By incorporating dual synchronizers at the input and output ports, this FIFO design guarantees data integrity and efficient data exchange between clock domains operating at different frequencies. The choice to configure the FIFO as a 16x8 memory bank underscores its ability to handle a substantial volume of data while maintaining robust synchronization and data transfer. Engineers must carefully analyze system requirements and clock relationships during the design and

implementation of a dual clock asynchronous FIFO with dual synchronizers to achieve optimal performance and reliability.

Results:

RTL View of FIFO-



Output Waveform with different test cases:

References/Sources used for the work :

1. https://www.sunburst-design.com/papers/CummingsSNUG2002SJ_FIFO1.pdf

2. https://asic-soc.blogspot.com/2007/12/new-asynchronous-fifo-design.html

3. ChatGPT