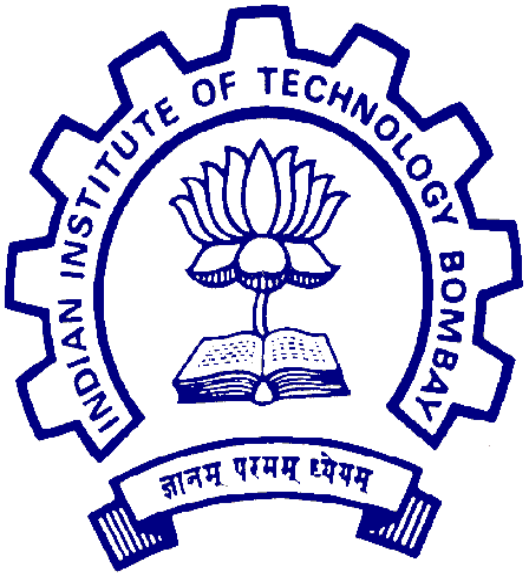
SELF PROJECT REPORT ON
ASYNCHRONOUS FIFO



DEPARTMENT OF ELECTRICAL ENGINEERING
IIT BOMBAY

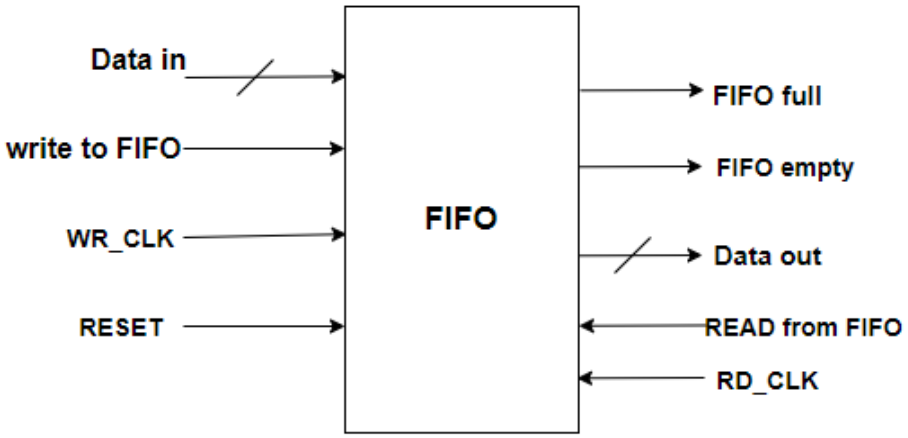
T.SIMMANAIDU(203070061)

ACADEMIC YEAR(2021-2022)

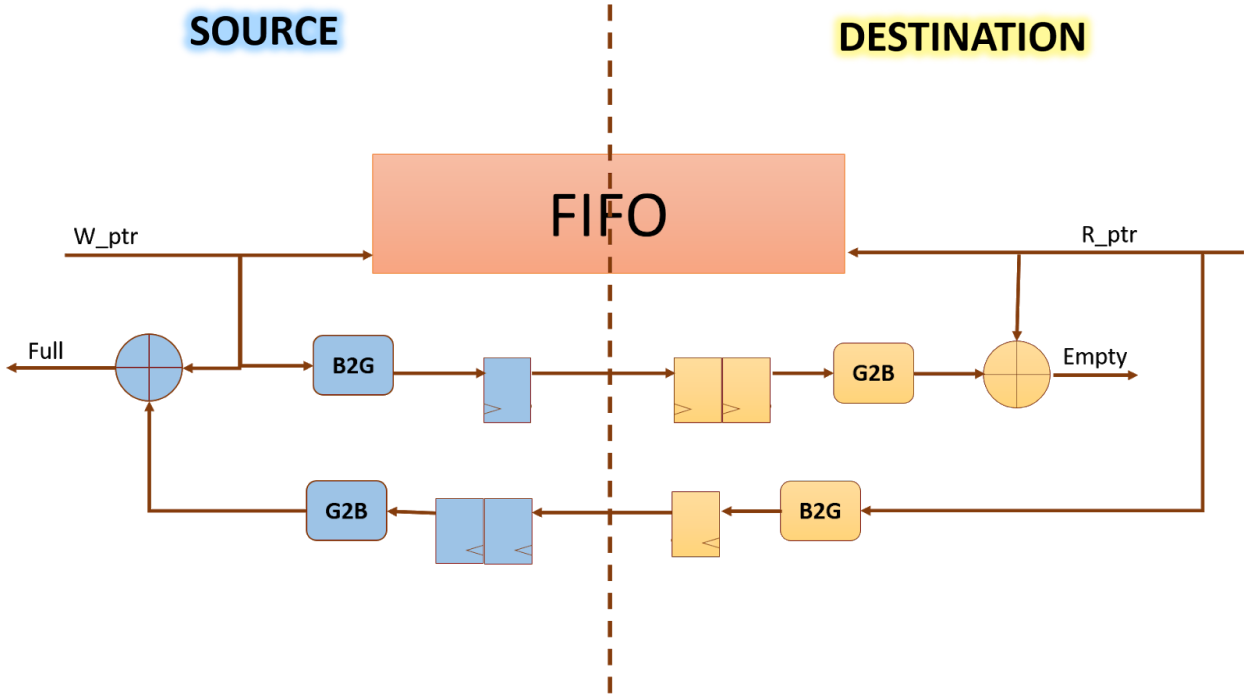
INTRODUCTION:

An Asynchronous FIFO refers to a FIFO design where data values are written sequentially into a FIFO buffer using one clock domain , and the values are sequentially read from the FIFO buffer using another clock domain,where the two clock domain are asynchronous to each other.

One common technique for designing an asynchronous FIFO is to use Gray code pointers that are synchronized into the opposite clock domain before generating Synchronous FIFO full or empty status signals.



Full diagram:



Asynchronous FIFO pointers:

Write pointer:
The write pointer always points to the next byte to be written,therefore on reset, both pointers are set to zero, which also happens to be the next FIFO byte location to be written.

Read pointer:
The read pointer always points to the current FIFO byte to read.

The generation of FIFO pointers is of utmost importance because they are clocked at different clocks so they need to be synchronized if they are used in other clock domains. These synchronized pointers are then compared to generate full and empty conditions.

Generation of full and empty conditions:
Empty condition arises when both read pointer and write pointer both are equal. This condition can happen when both pointers are reset to zero or when the read pointer catches up to the write pointer that is the last byte of data from FIFO has been read.

Full condition arises when the pointers are again equal but this time write pointers catch up to the read pointer that is the last location of the FIFO memory has been written. To distinguish between these two conditions,one extra bit is added to each pointer. Thus if all the bits except Msb of the both the pointers are equal then FIFO is full. Also if all the bits including MSB of the both the pointers are equal then FIFO is empty.

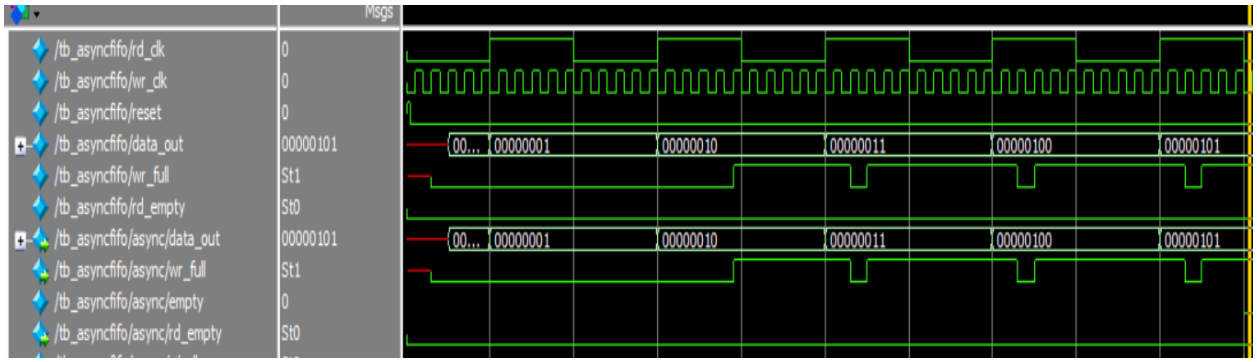
To generate FIFO full and empty conditions, the two pointers are compared in different clock domains i.e read pointer should be synchronized with the write clock domain to be compared with the write pointer and vice versa. Gray coded values are used instead of binary pointers. Thus synchronizing binary values is problematic.Gray codes allow only one bit to change for each clock transition, eliminating the problem of synchronising multiple bits at the same time.

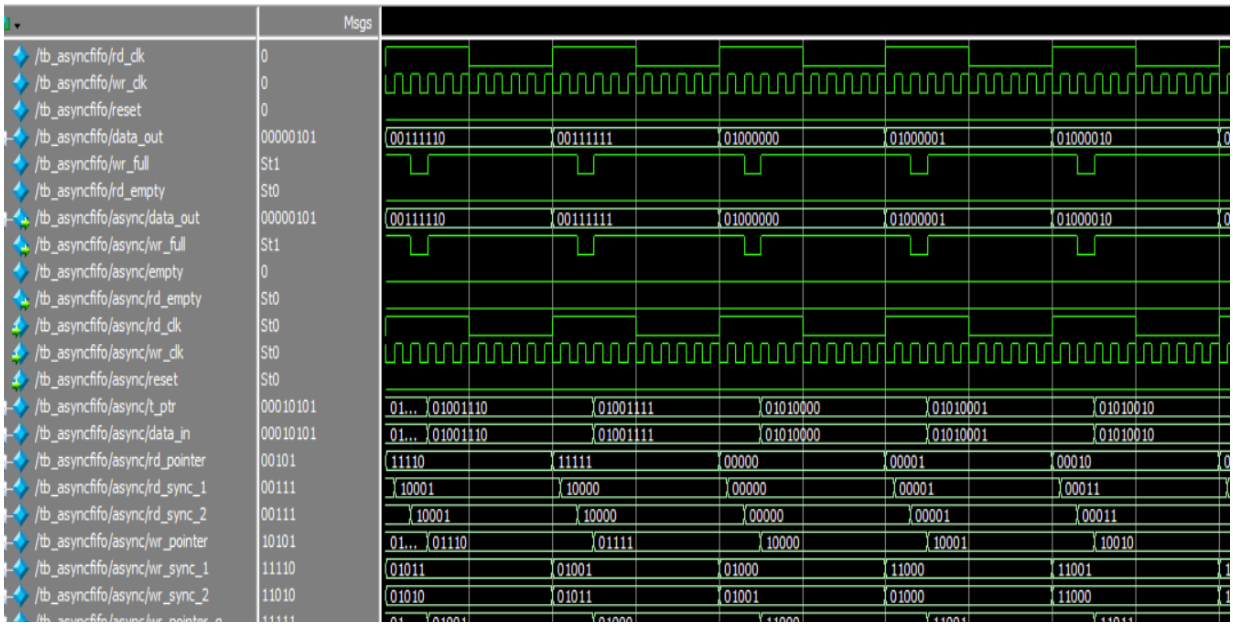
FIFOs are used in designs to safely pass multi-bit data words from one clock domain to another. Data words are placed into a FIFO buffer memory array by control signals in one clock domain, and the data words are removed from another port of the same FIFO buffer memory array by control signals from a second clock domain

Given signals in simulation:
reset,data_out, wr_full, rd_empty, rd_clk, wr_clk
Wr_clk=50MHZ
Rd_clk=5MHZ
FIFO depth is 16 and width is one byte.

Data coming from a memory which has 256 locations and each location has data with 8 bits.

Simulation results:





Conclusion:

The asynchronous FIFO is successfully implemented in verilog and simulated in Quartus and all flags like empty,full are included in the design to make the design safer. The write clock frequency given is 50MHZ and the read clock frequency is 5MHZ.

References:

1.Shruti Sharma, “Implementation of an RTL synthesizable asynchronous FIFO for conveying data by avoiding actual data movement via FIFO”.

2.H. Ashour, "Design, simulation and realization of a parameterizable, configurable and modular asynchronous FIFO," *2015 Science and Information Conference (SAI)*, 2015