Implementation Summary

FIFO Shift Register – Verilog Design

Board Used: Basys 3 Artix-7 (xc7a35tcpg236-1)

Objective

To design a FIFO shift register with a depth of at least 10 words, and a width of at least 16-bits. Also, to design a test bench, and generate synthesis and implementation designs for the same.

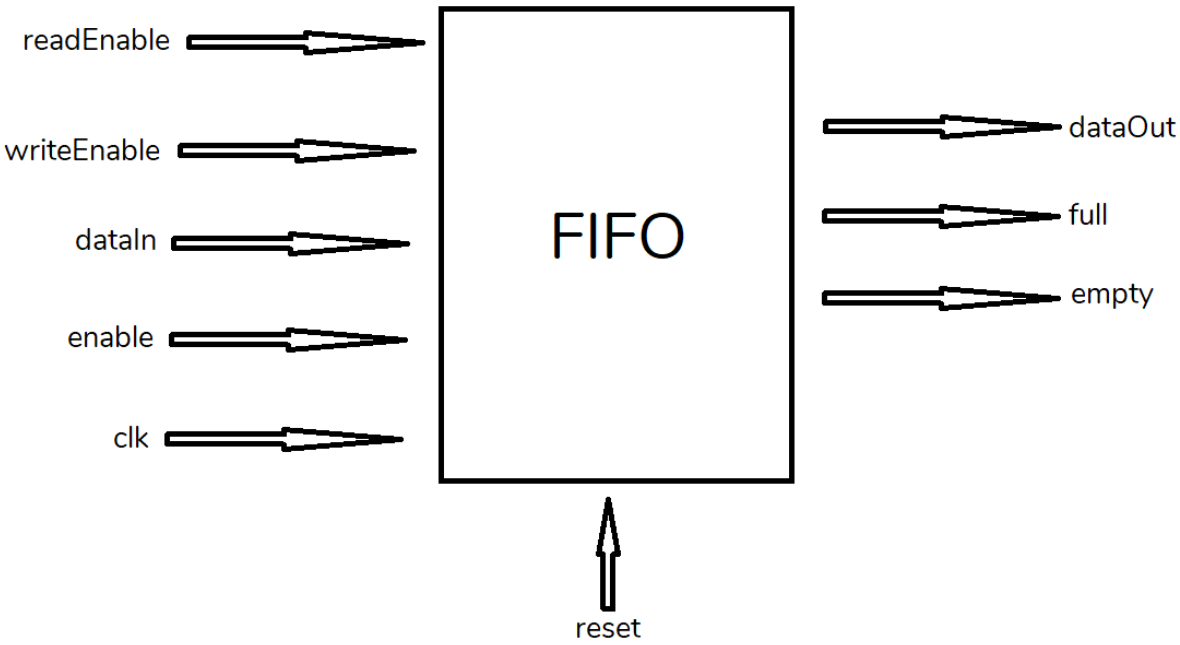
Description

A FIFO shift register is a register based on the First-In First-Out methodology of data organization and manipulation. It behaves like a queue of people, where the first element (person) to enter is also the first one to leave.

A FIFO can have different clocks for the read and write operations. This kind of FIFO is frequently known as an asynchronous FIFO, and is used to cross clock domains. A FIFO which uses the same clock for the read and write operations is often known as a synchronous FIFO, and is mainly used for data storage and buffering.

A queue behaves in a similar fashion as a FIFO, with the first person entering also being the first person to leave.

We are going to design a synchronous FIFO, with the following interface:



Interface for designed FIFO Shift Register

Pin Functions

Inputs:

- readEnable – Enables reading of the data at the *dataIn* terminal into the FIFO shift register
- writeEnable – Enables the writing of data at the *dataOut* terminal; lower priority than readEnable
- dataIn – Used to feed data into the FIFO shift register while writing
- enable – Enables the device; the device will not function if *enable* is 0
- clk – Used to provide a clock signal to the device
- reset – Resets the outputs and internal signals to 0; clears the register memory

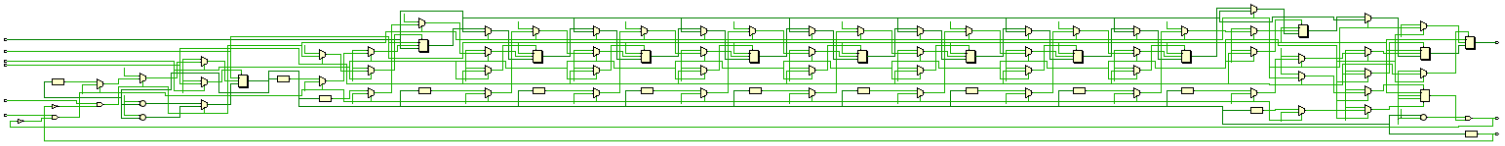
Outputs:

- dataOut – Holds the data to be read
- full – Indicates if the register memory is full, and prevents writing if so
- empty – Indicates if the register memory is empty, and prevents reading if so

All the signals used above are active-high, meaning that their value is 1 when they are active. The positive edge of the clock is used to trigger the device.

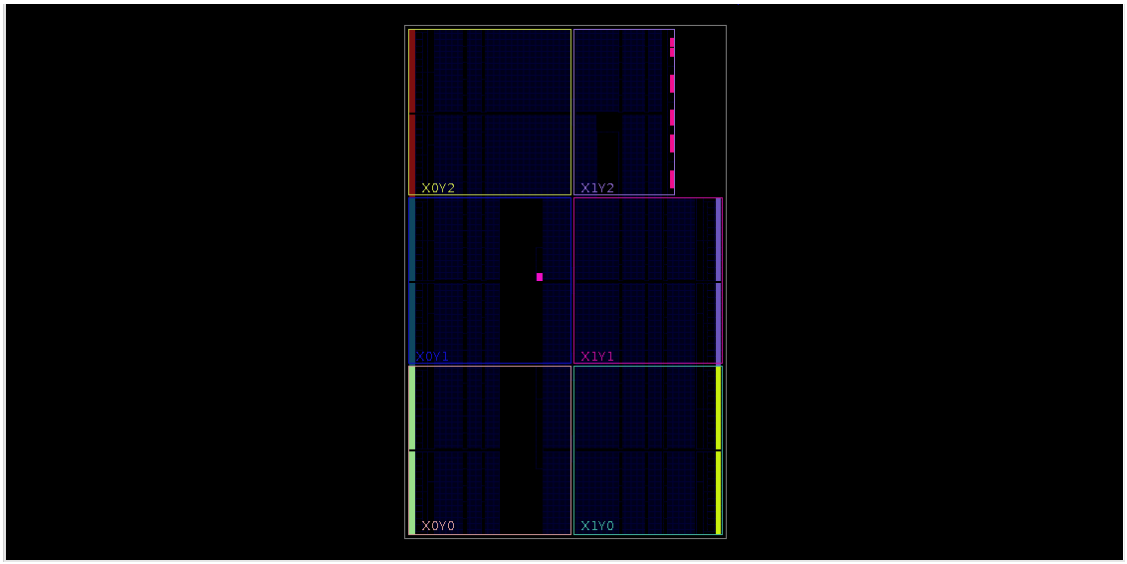
Designs

- **RTL Design**

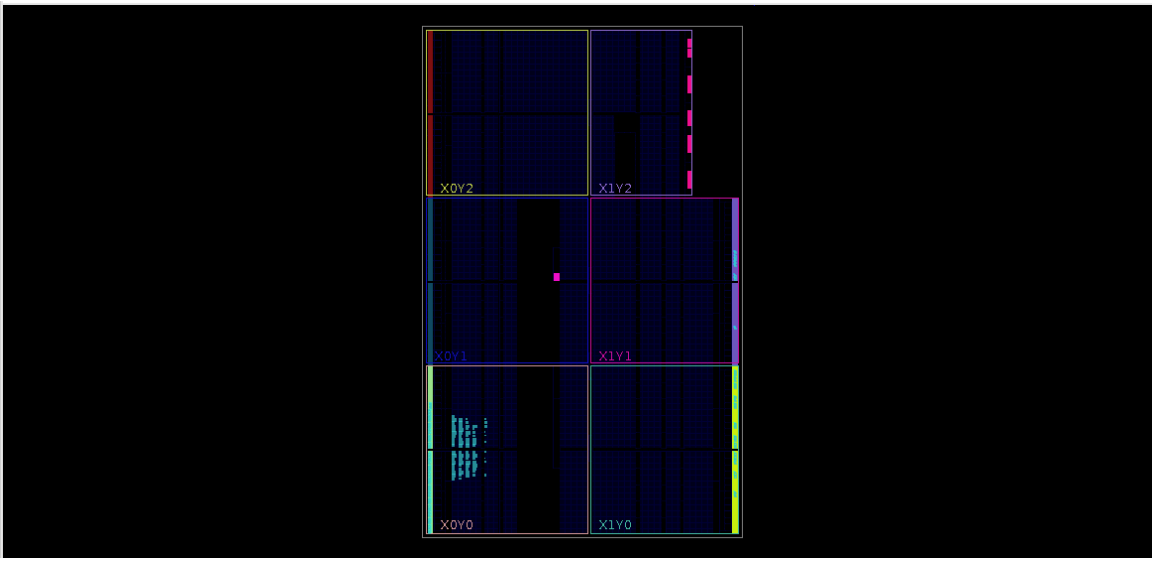


438 Cells 71 I/O Ports 765 Nets

- **Synthesized Design**

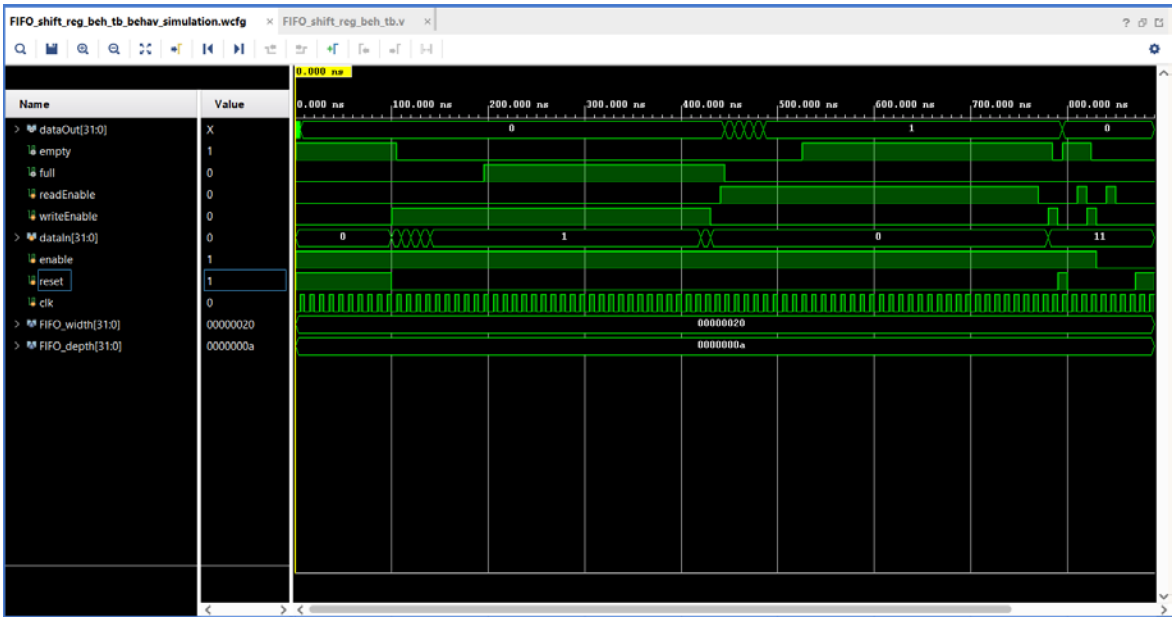


- **Implemented Design**



Reports

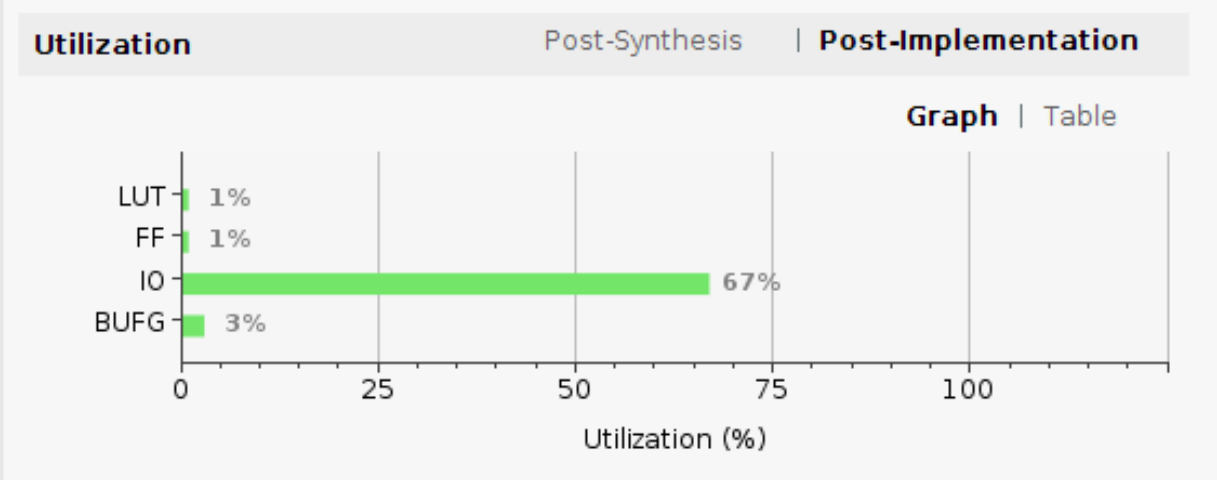
- **Simulation Results**



(refer to the `Simulation Results.wcfg` file to access the waveforms interactively)

The simulation results were as expected, and verified the logic design of our model.

- **Resource Utilization**



Our design utilizes a lot of input-output ports. This is mainly due to the selected FIFO width, which makes both *dataIn* and *dataOut* use 32 ports each, thereby utilizing a lot of the input-output ports.

- **Timing Analysis**

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.918 ns	Worst Hold Slack (WHS):	0.140 ns	Worst Pulse Width Slack (WPWS):	3.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	713	Total Number of Endpoints:	713	Total Number of Endpoints:	358

All user specified timing constraints are met.

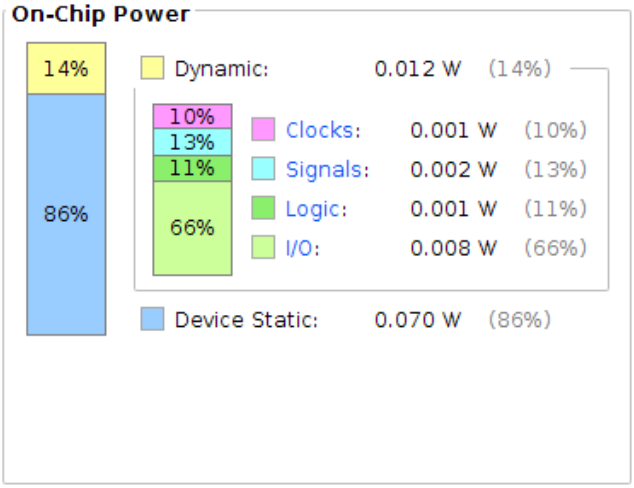
In the simulation, a clock with a time period of 10ns was used. However, this clock was yielding a WNS of greater than 5ns, therefore the synthesis was redone with a clock period of 8ns. This yielded an acceptable WNS of 3.918 ns. All the other delays were well below their maximum permissible values.

- **Power Utilization**

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**0.082 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**25.4°C**
Thermal Margin:	59.6°C (11.9 W)
Effective θJA:	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity



The power utilization of our device is quite low, as well as the junction temperature is permissible. Most of the contribution in power consumption comes from the input-output handling, which is reasonable due to the large number of input-output ports being utilized.