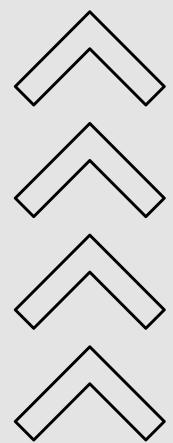
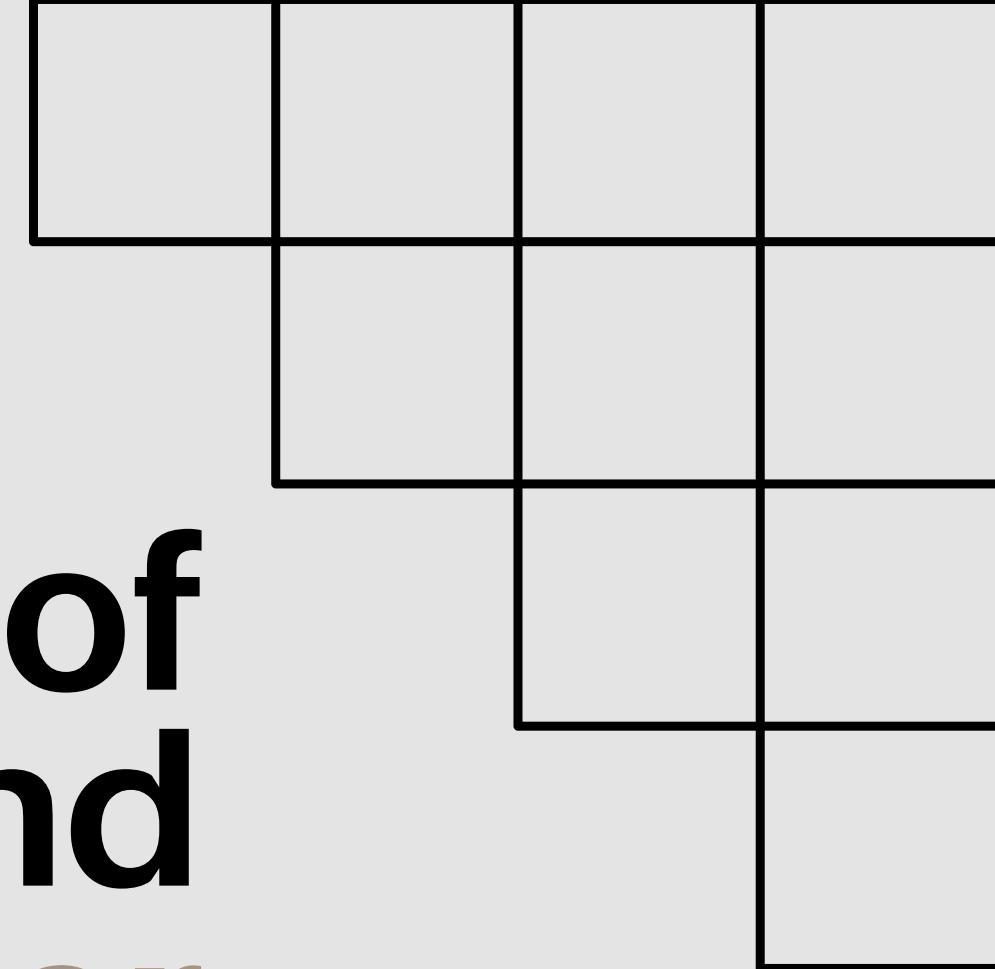
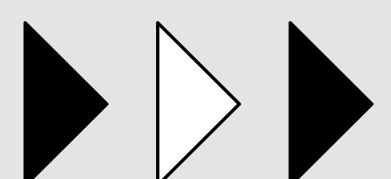
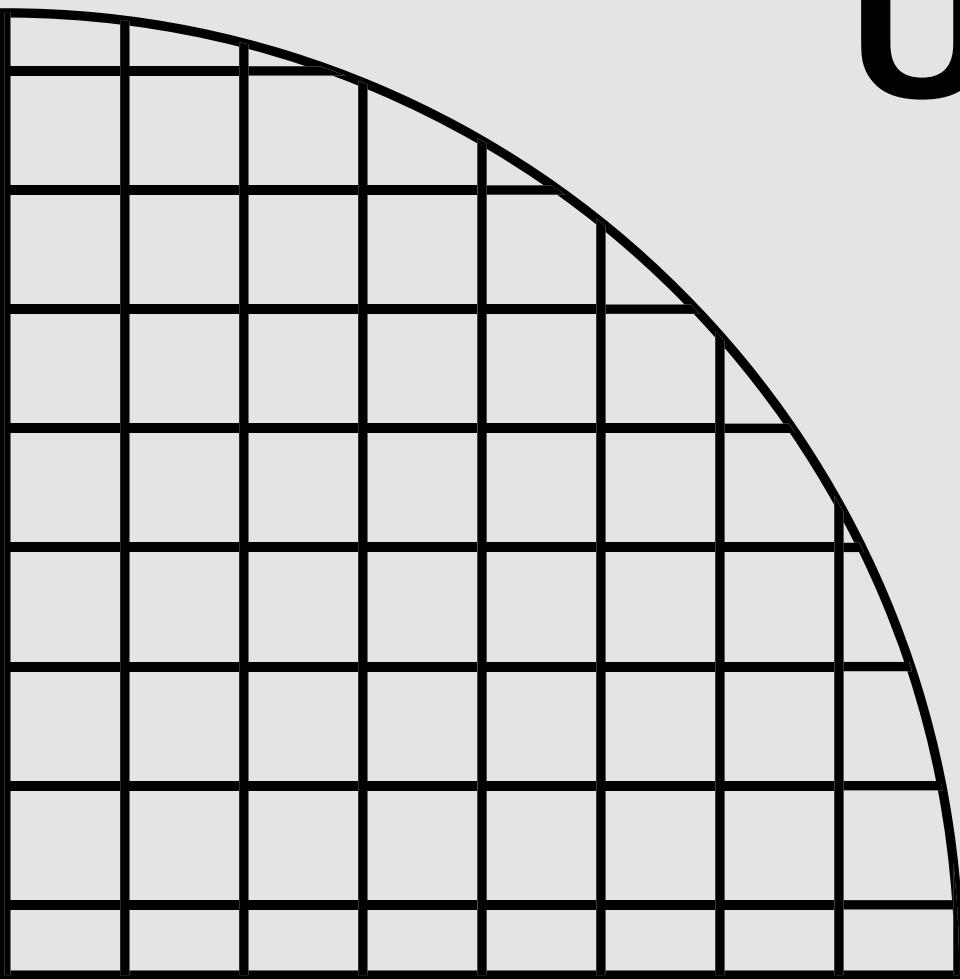


25VL682

RTL Design and FPGA Synthesis Lab
(Term Work)



Implementation of Round Robin and Priority-Based Arbiter Using Verilog HDL



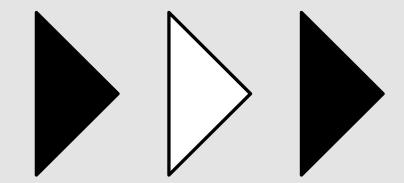
Presented by

Gadha Sanal -CB.EN.P2VLD25009

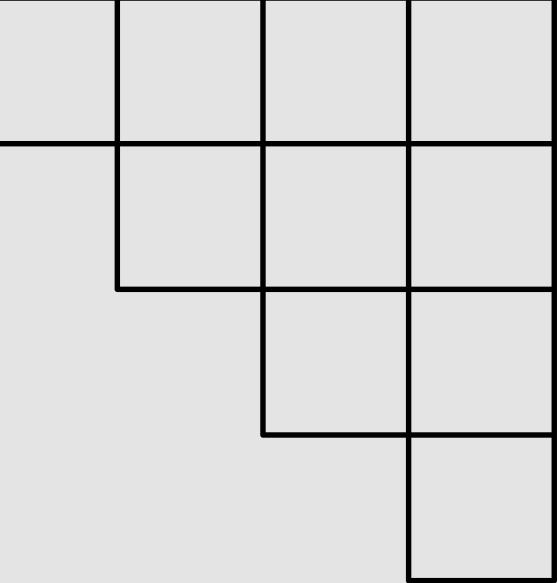
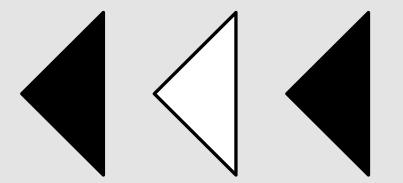
Neethu Jaisan-CB.EN.P2VLD25017

Sirigireddy Saikrupareddy-CB.EN.P2VLD25029

P Poornesh-CB.EN.P2VLD25021



Contents



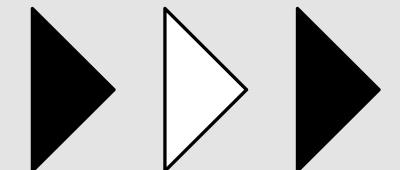
Introduction

Design Block
Diagram of RTL

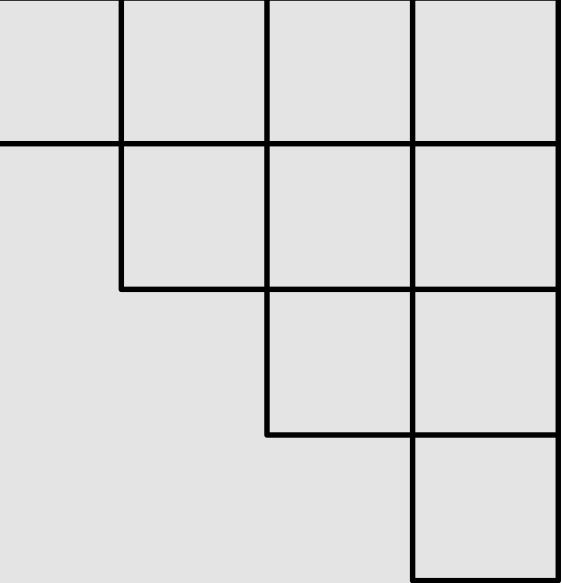
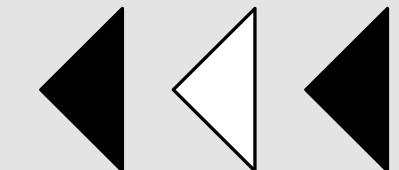
Scope of Design

Pin Diagram

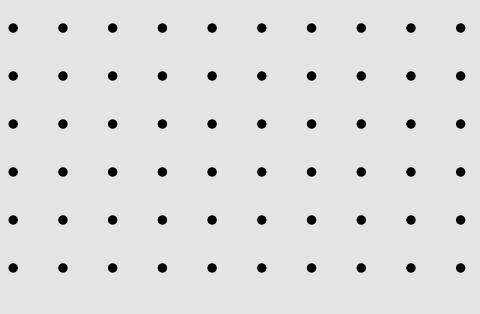
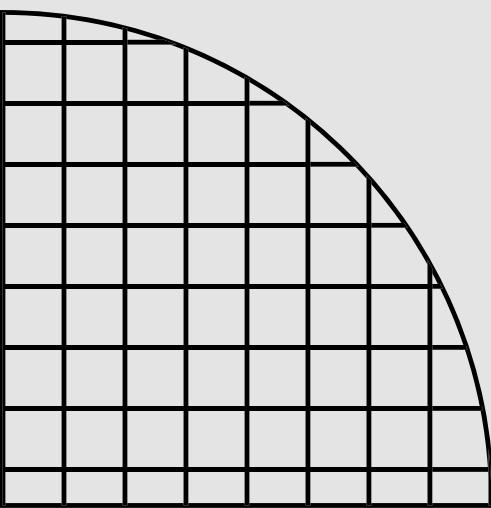
Timing Diagram

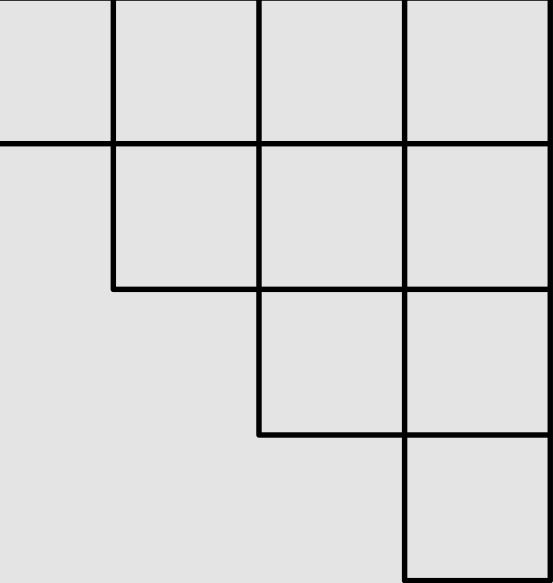
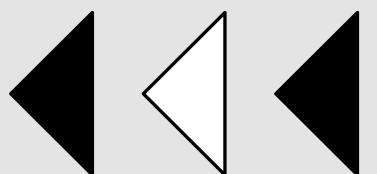
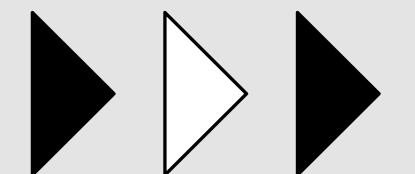


Introduction



- To design and implement a scalable Round Robin and Priority-Based Arbiter using Verilog HDL.
- To develop the arbiter as a Finite State Machine (FSM) for efficient request-grant control.
- To simulate and verify the design using Verilog testbenches in Vivado.
- To analyze and compare the fairness, latency, and performance of both arbitration schemes.





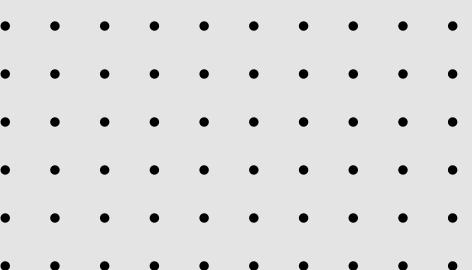
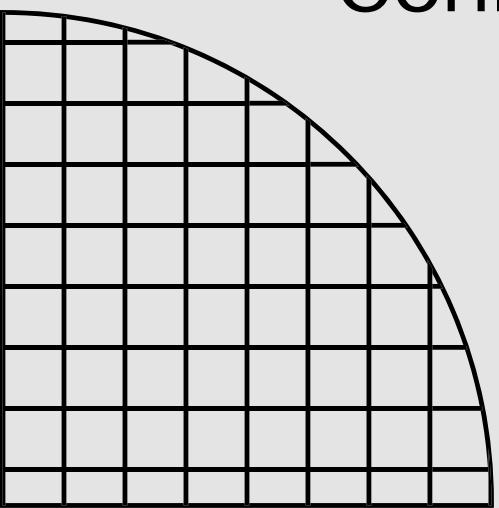
Introduction

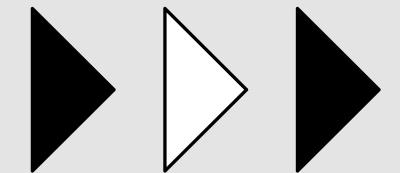
In digital systems, when multiple devices or modules request access to a common resource such as a bus or memory, a decision must be made about which request to serve first.

This decision-making process is done by a hardware block called an Arbiter.

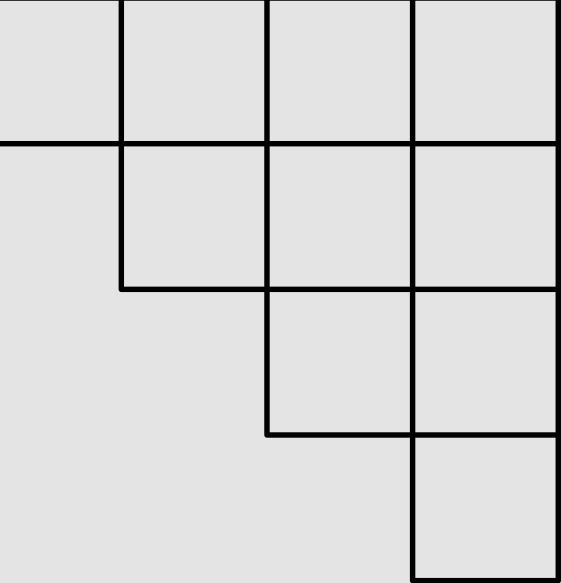
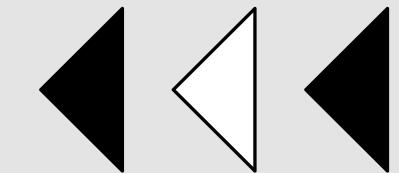
An Arbiter ensures that:

- Only one request is granted at a time.
- Resource sharing happens efficiently.
- Conflicts between multiple requests are resolved fairly.





Introduction

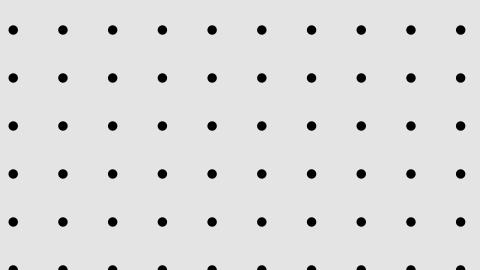


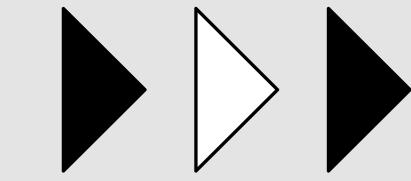
Priority Arbiter

1. Gives access based on fixed priority levels (some devices always have higher priority).
2. Fast decision-making, as priority is predefined.
3. Can cause starvation for low-priority requests if high-priority ones keep coming.

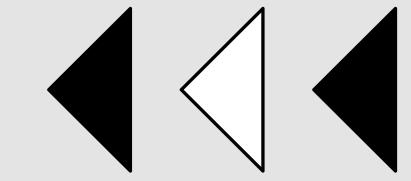
Round Robin Arbiter

-
- A quarter-circle shape filled with a grid pattern.
1. Rotates priority among all requesters to ensure fairness.
 2. Each requester gets a turn in circular order ($0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow$ back to 0).
 3. Prevents starvation by giving equal chance to all.





Scope of Design



Arbiter Implementation:

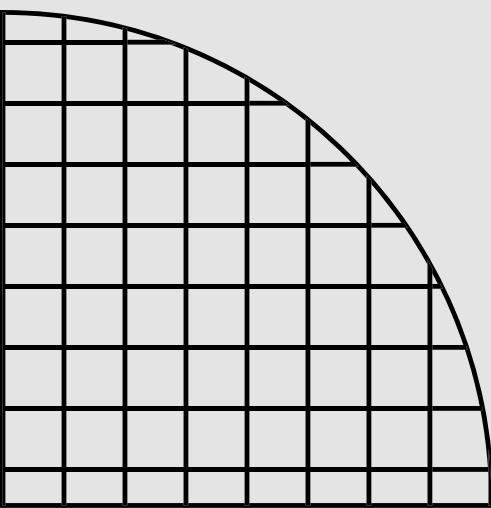
Design and implement a Round Robin and Priority-Based Arbiter in Verilog HDL, capable of handling multiple simultaneous requests fairly.

Scheduling Control:

The design includes a time quantum-based scheduling mechanism, ensuring each requester gets a defined time slot before rotation.

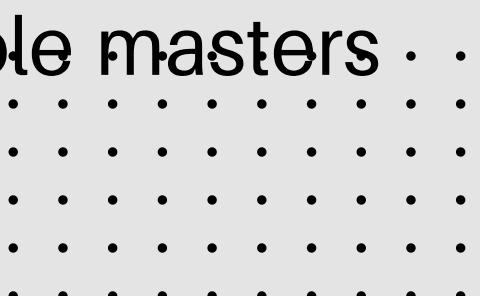
Simulation and Verification:

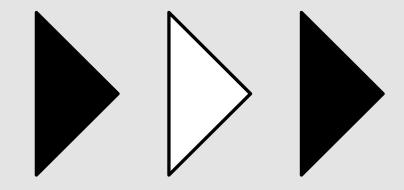
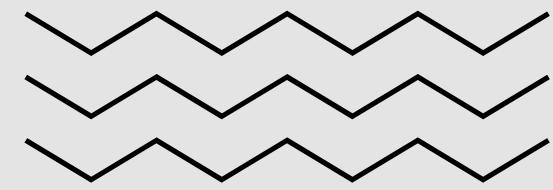
Simulate the design in Xilinx Vivado using testbenches to verify functionality, timing, and fairness under different request patterns.



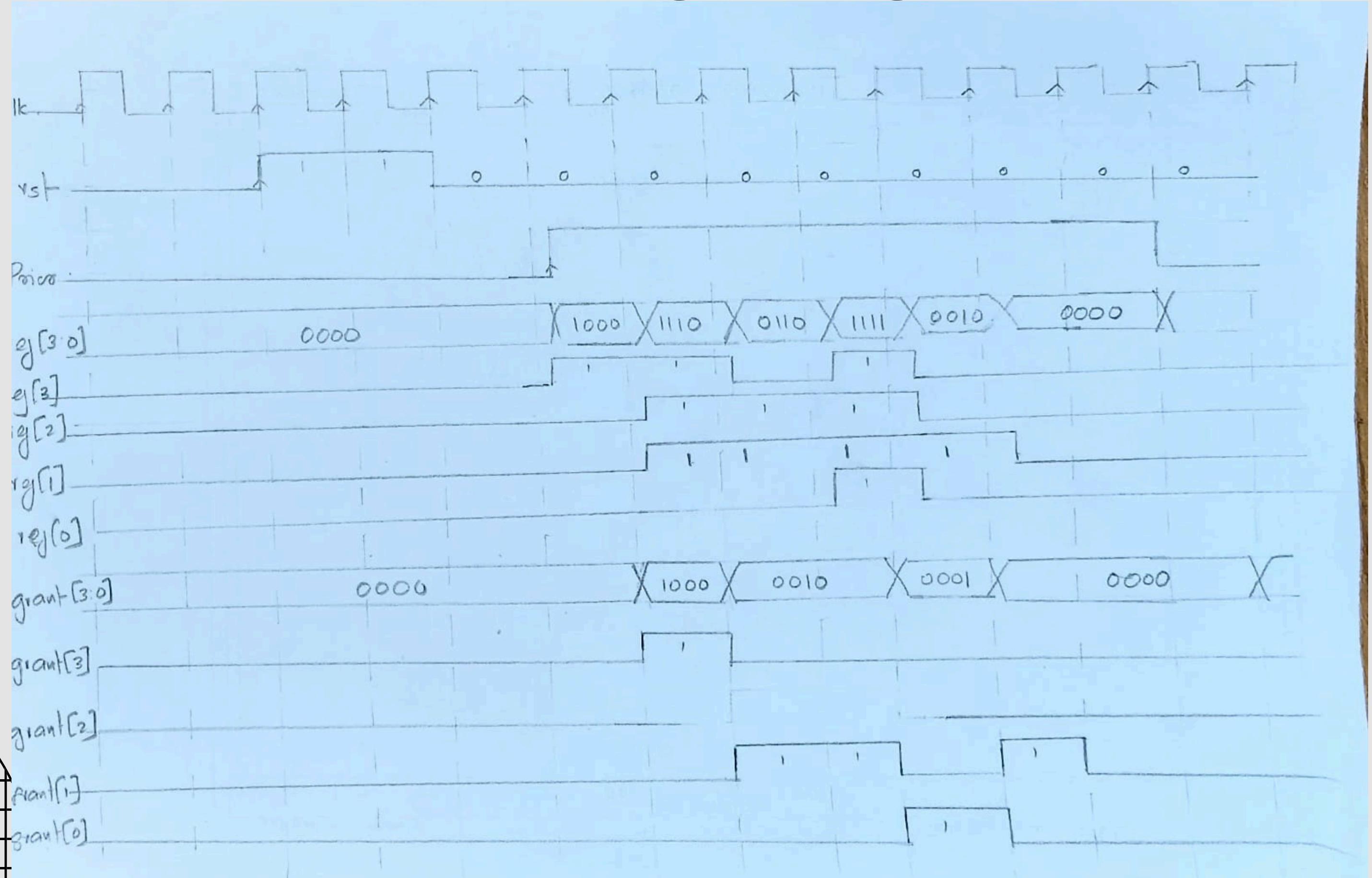
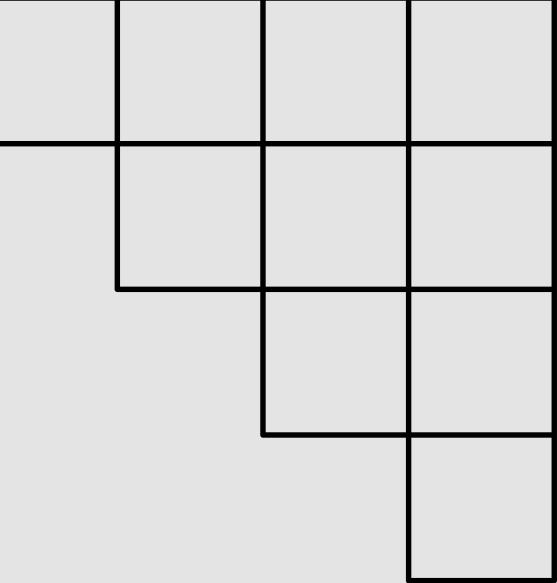
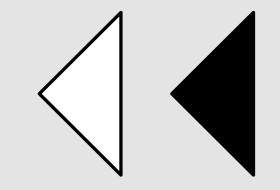
Scalability and Integration:

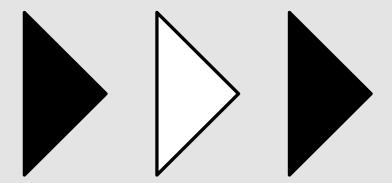
The arbiter can be integrated into larger SoC or bus systems, where multiple masters need shared resource access control.



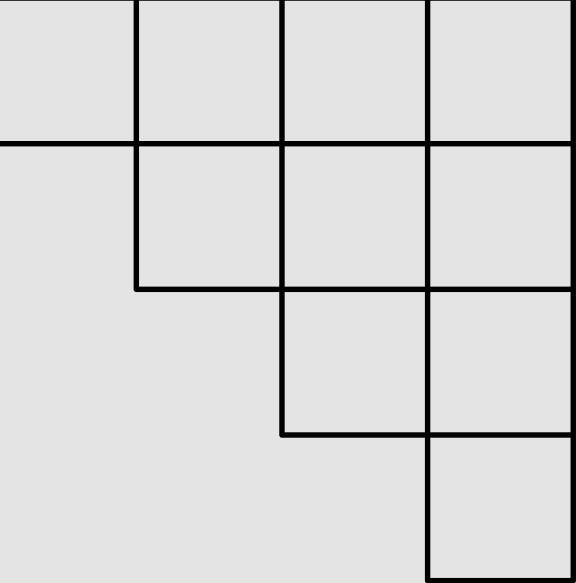
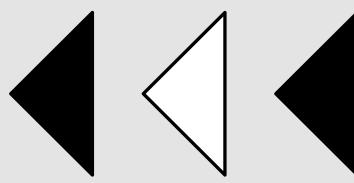


Timing Diagram



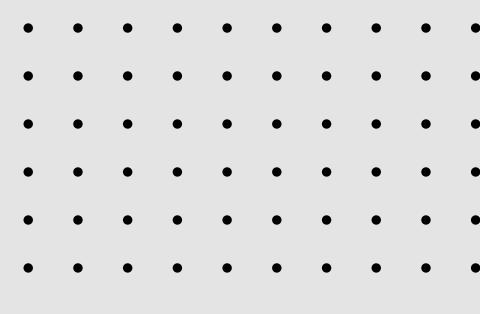
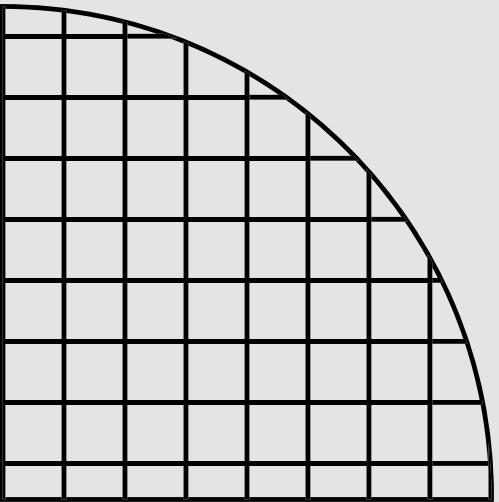


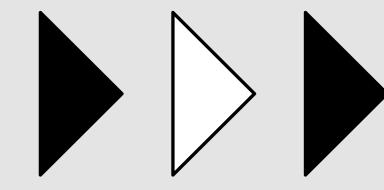
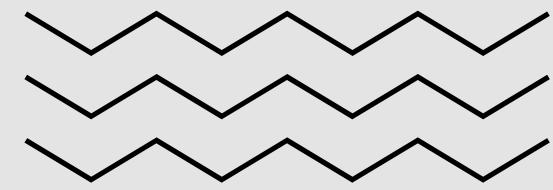
Timing Diagram



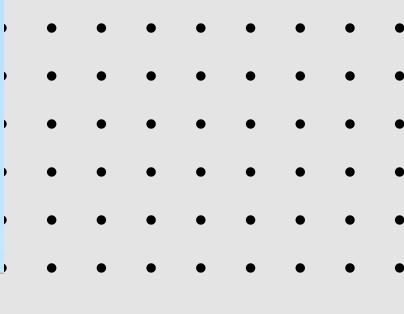
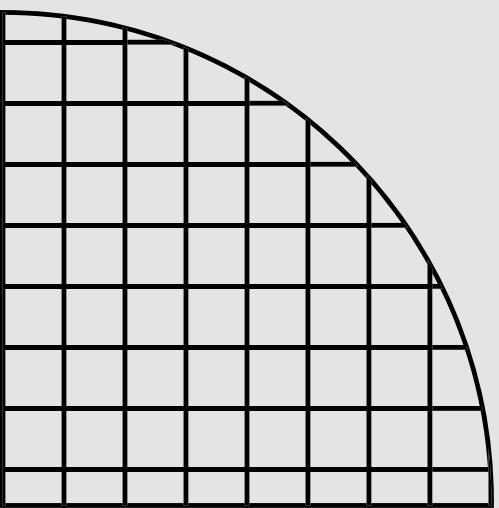
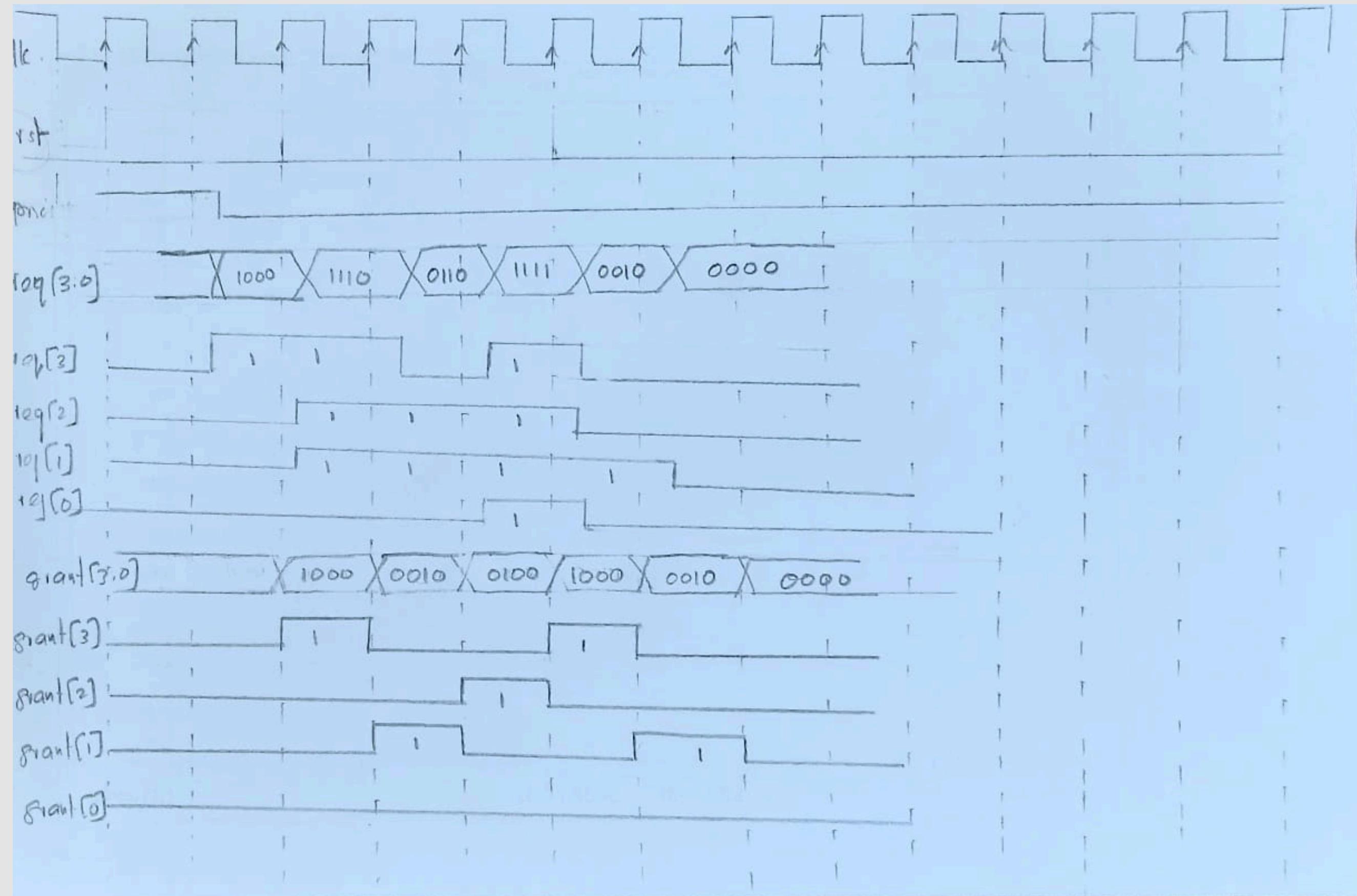
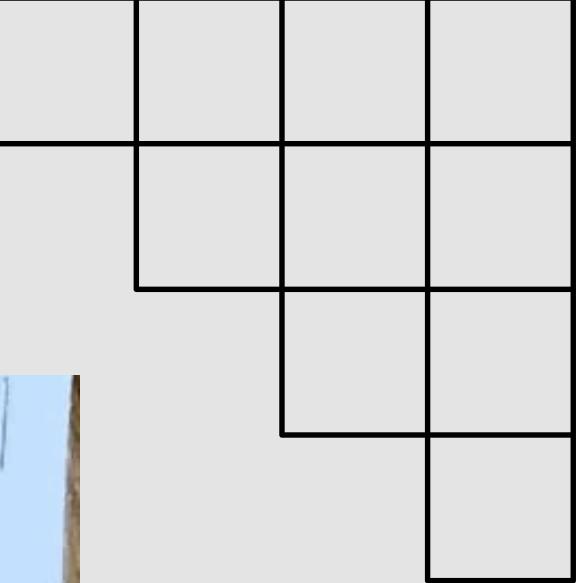
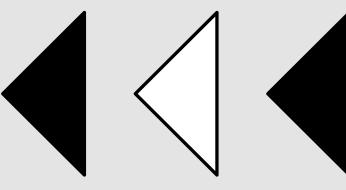
Prior	CLK	Request (4 bit)	Grant (4bit)
1	1	R3	R3
	2	R1,R2,R3	R1
	3	R1,R2	R1
	4	R0,R1,R2,R3	R0
	5	R1	R1
	6	R0,R1,R2,R3	R0
	7	R2	R2

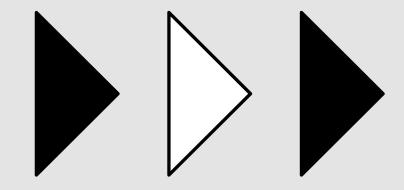
Priority Based Arbiter



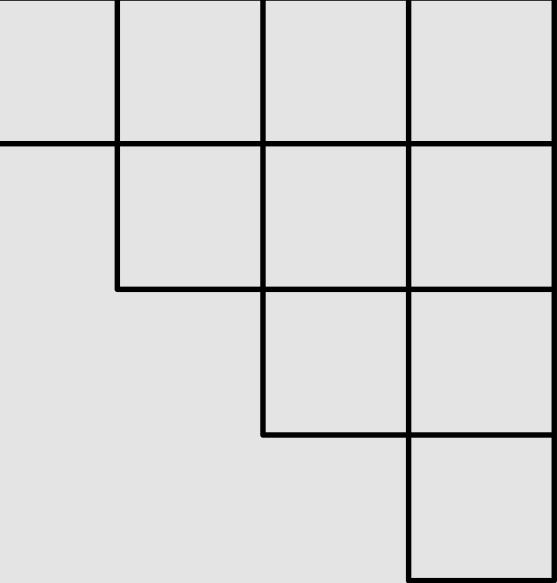
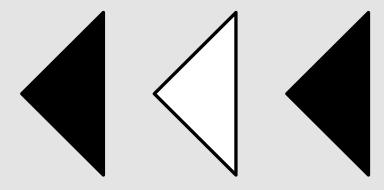


Timing Diagram



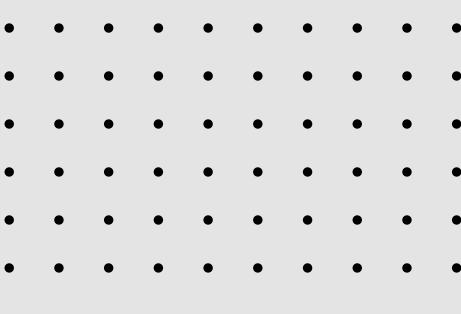
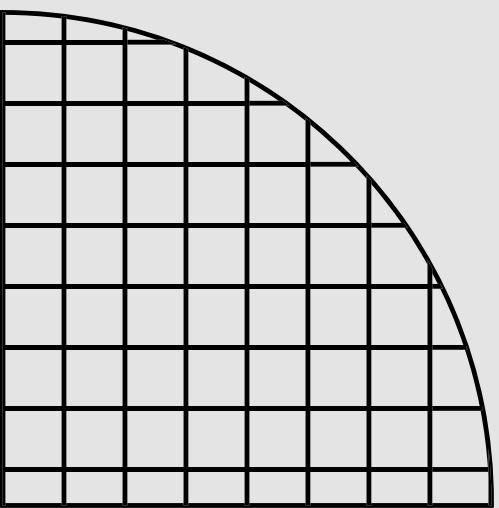


Timing Diagram

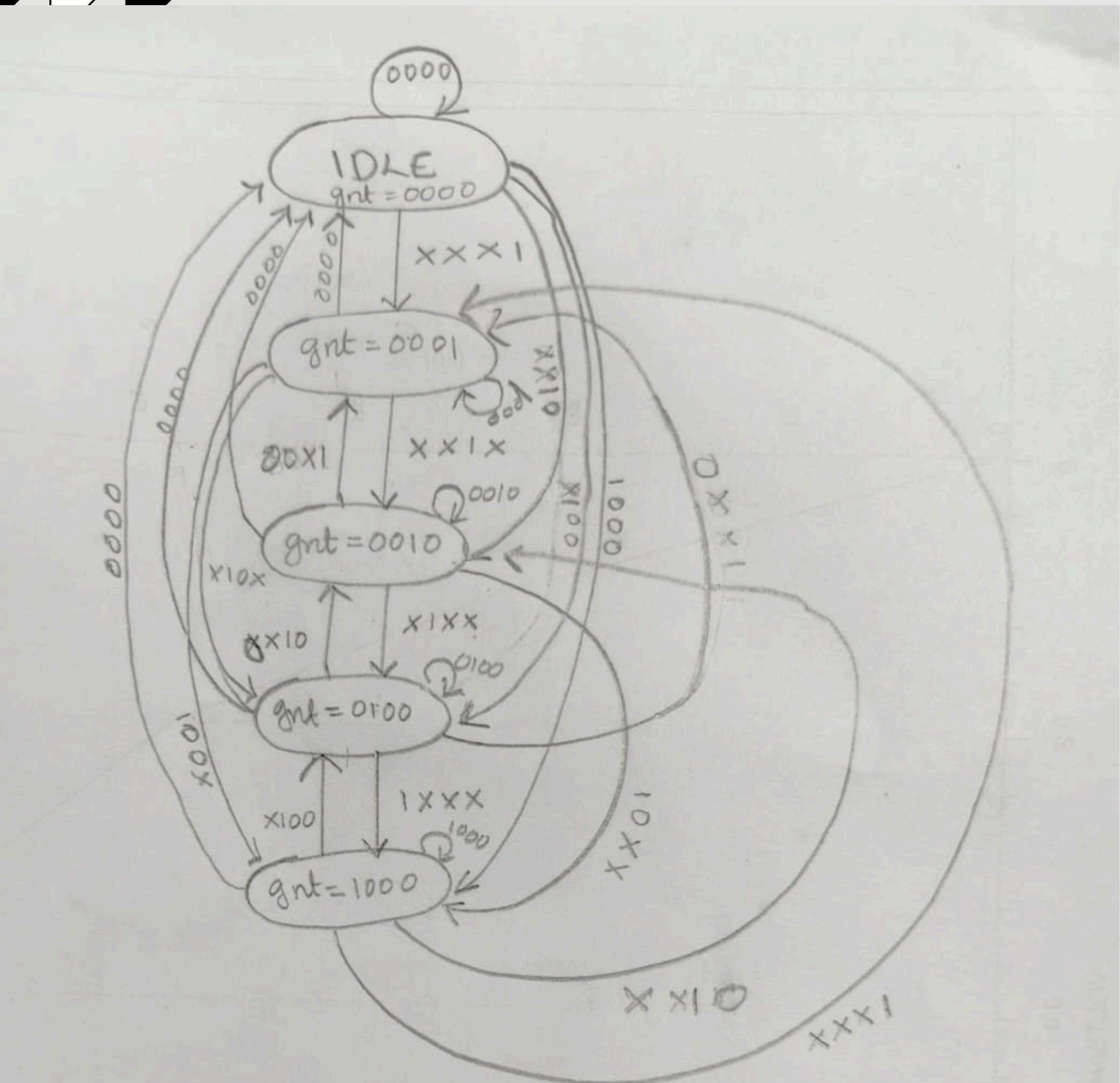


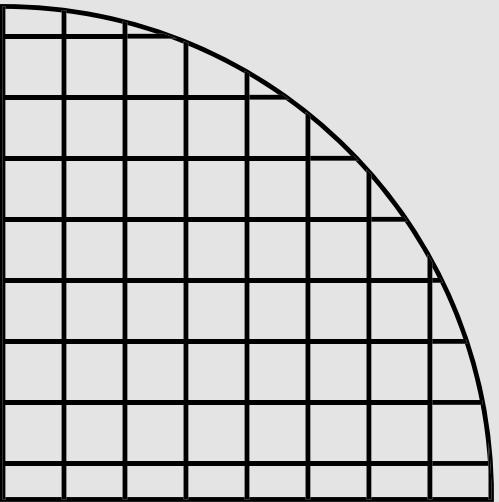
Prior	CLK	Request (4 bit)	Grant (4bit)
0	1	R3	R3
	2	R1,R2,R3	R1
	3	R1,R2	R2
	4	R0,R1,R2,R3	R3
	5	R1	R1
	6	R0,R1,R2,R3	R2
	7	R2	R2

Round Robin

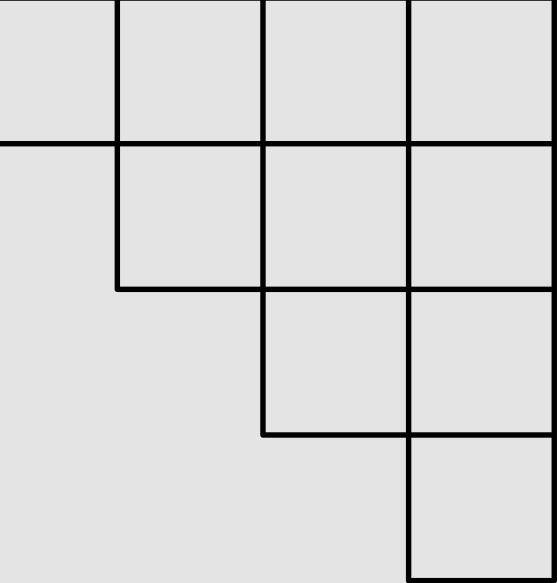
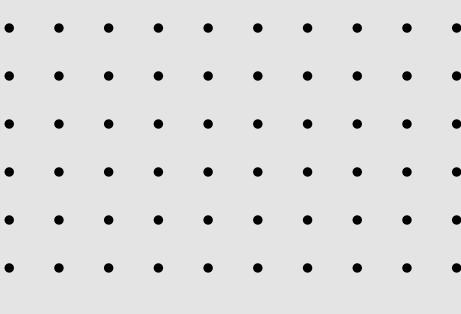
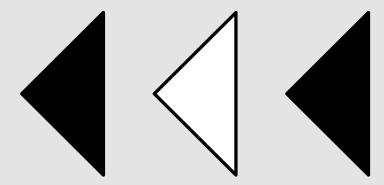
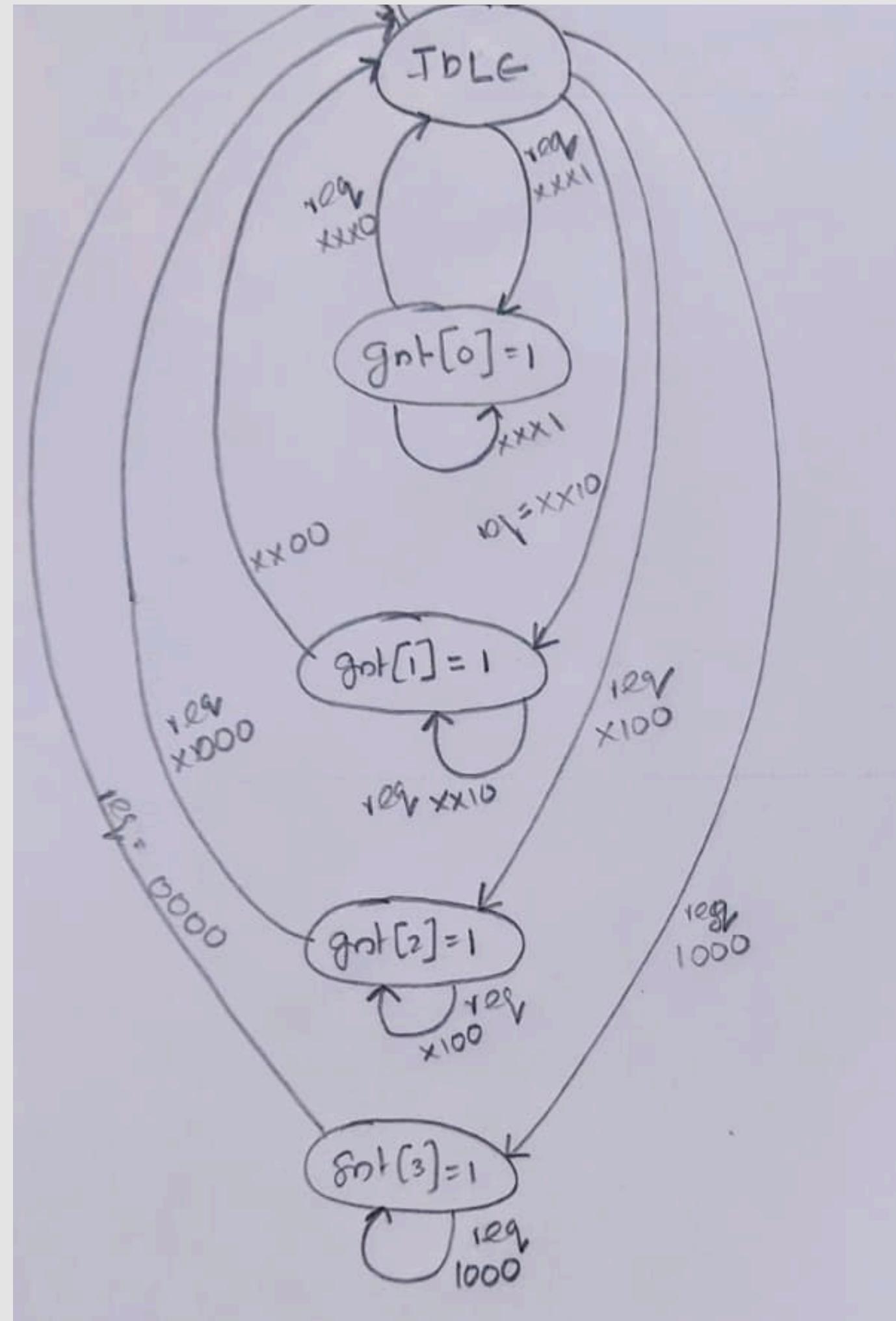
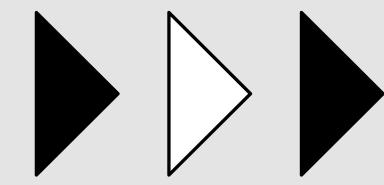


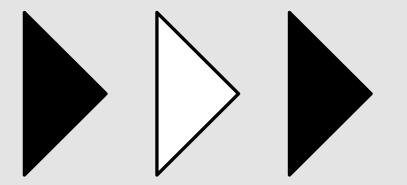
FSM- ROUND ROBIN



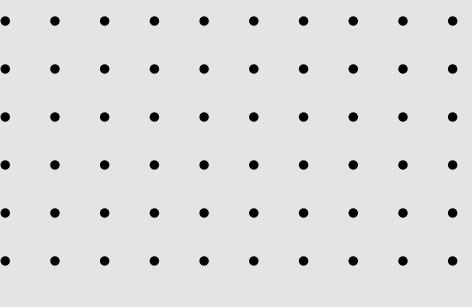
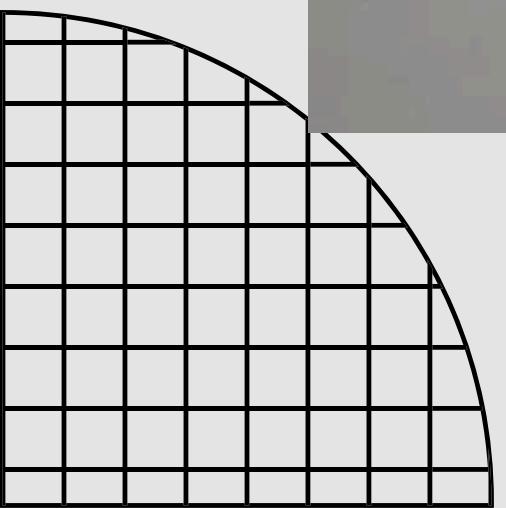
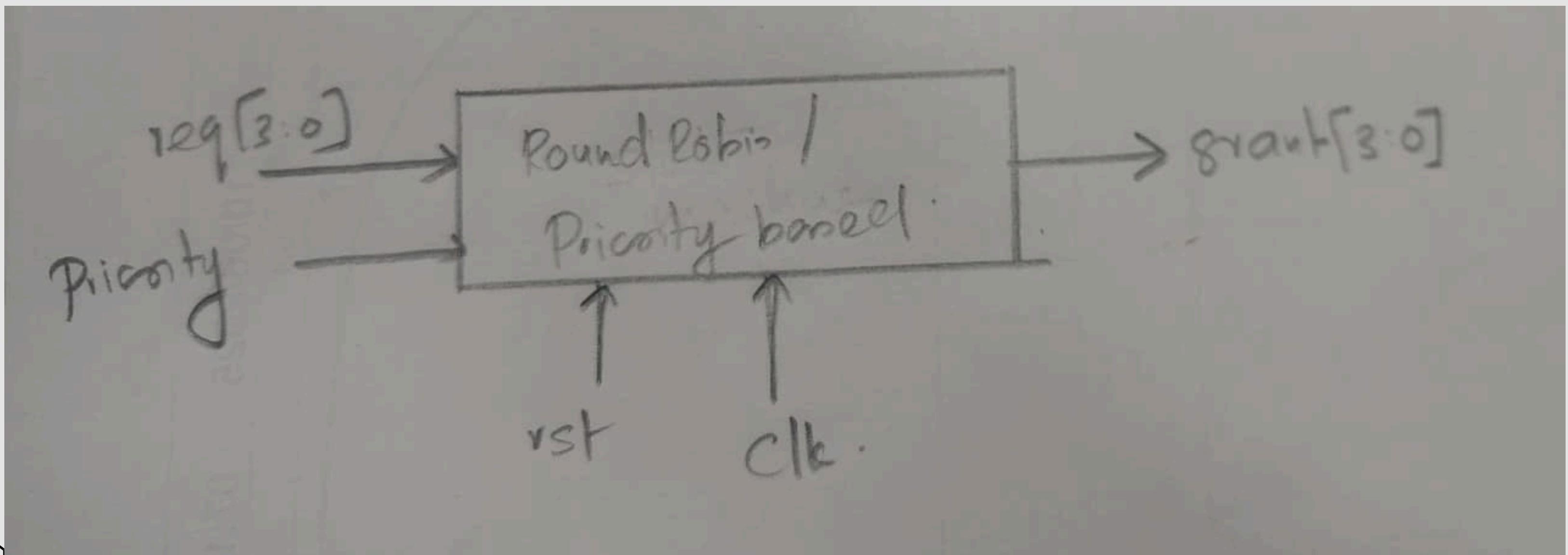
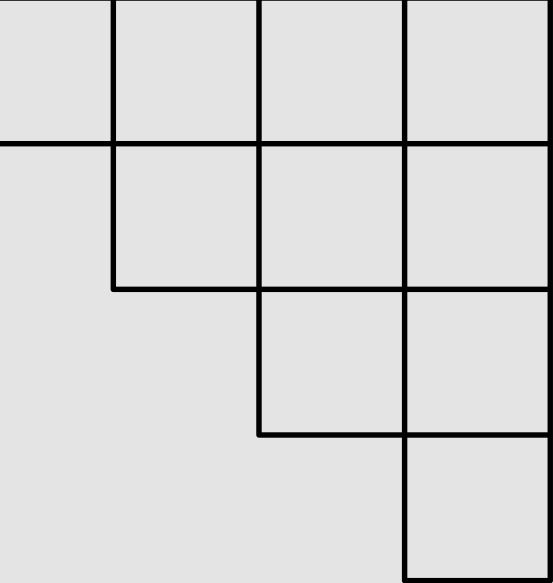
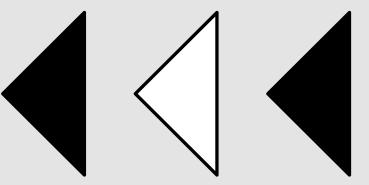


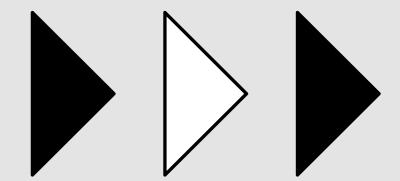
FSM-PRIORITY BASED ARIBIER



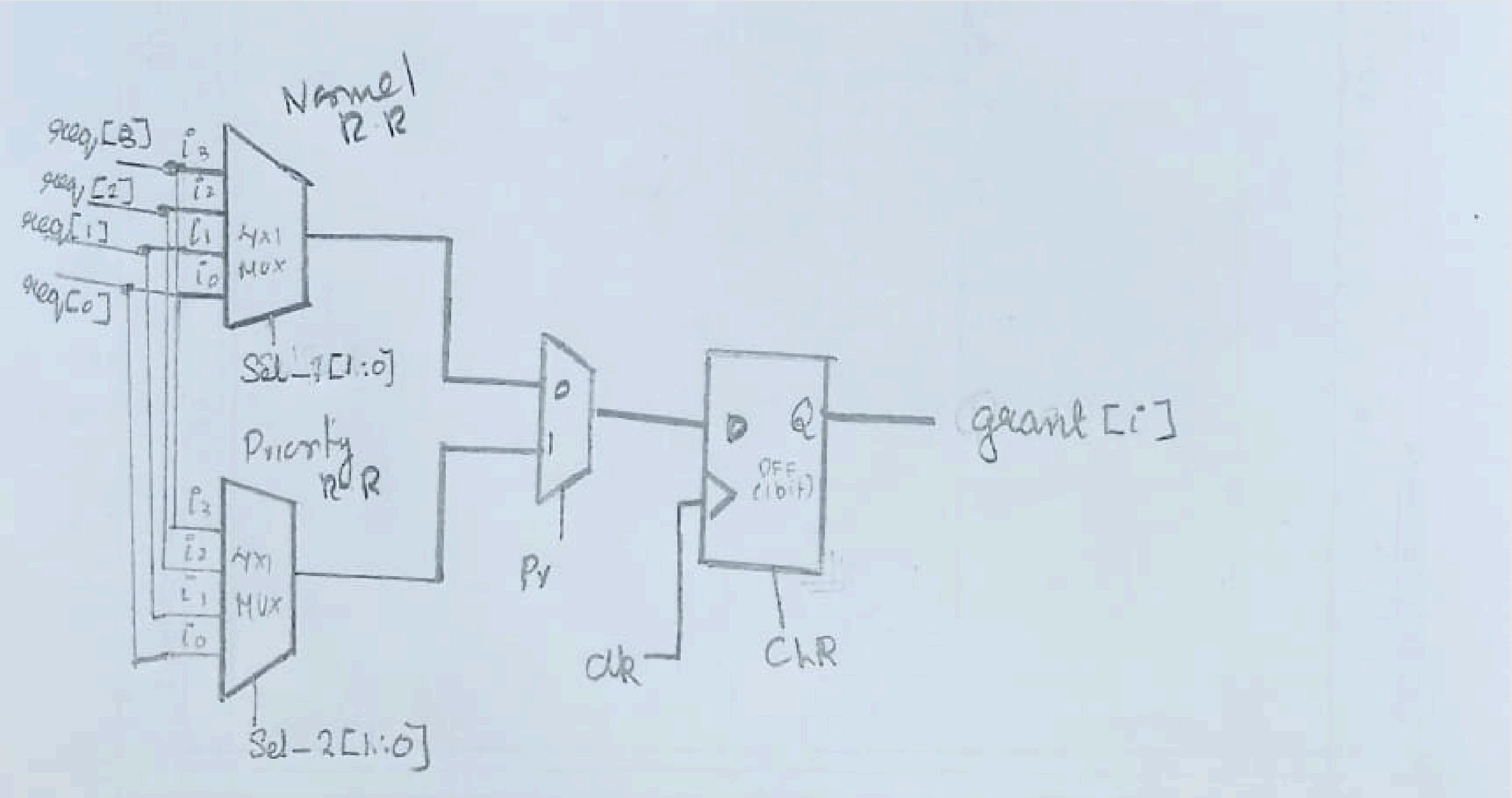
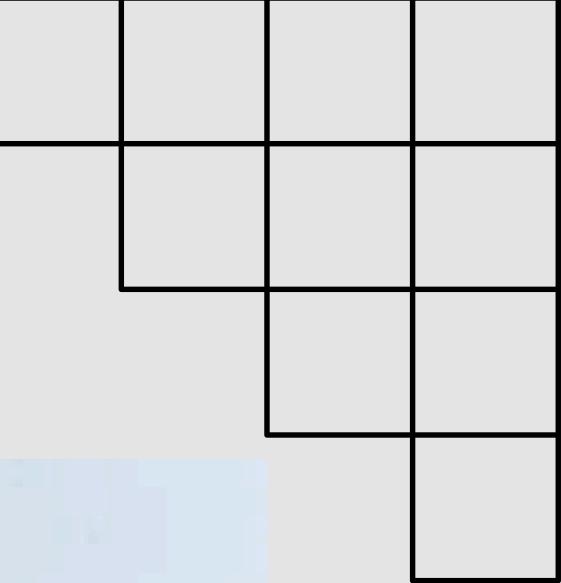
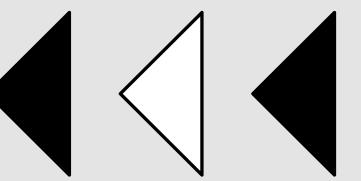


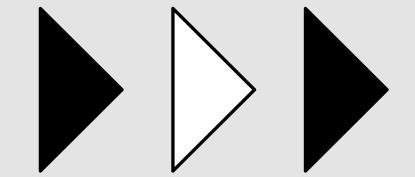
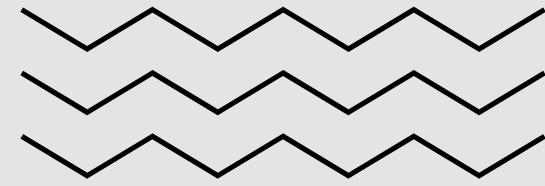
Design Block Diagram of RTL



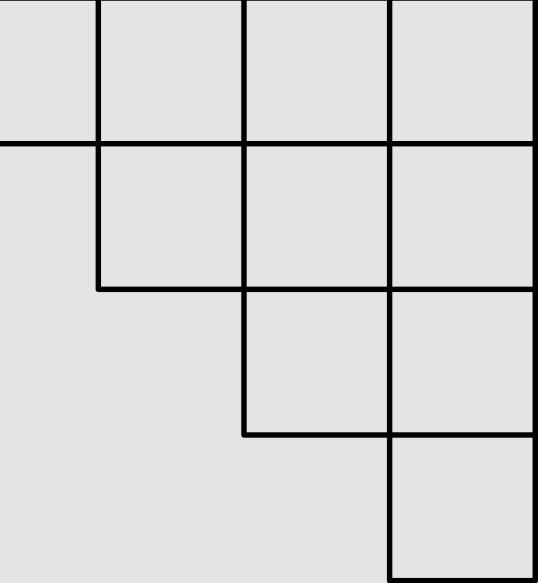
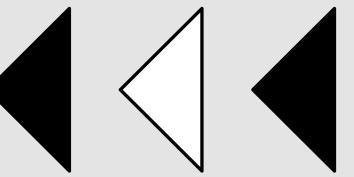


Pin Diagram





APPLICATIONS

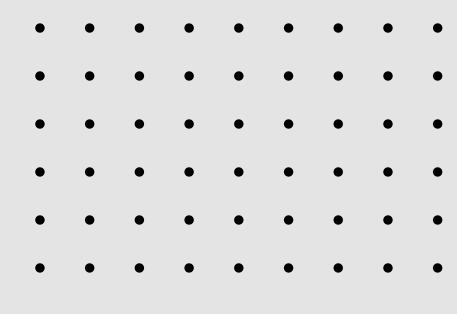
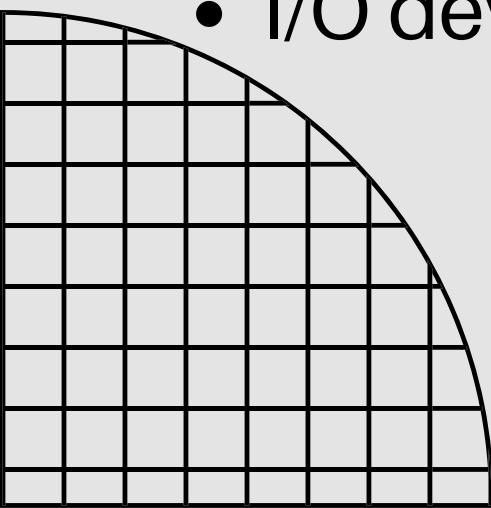


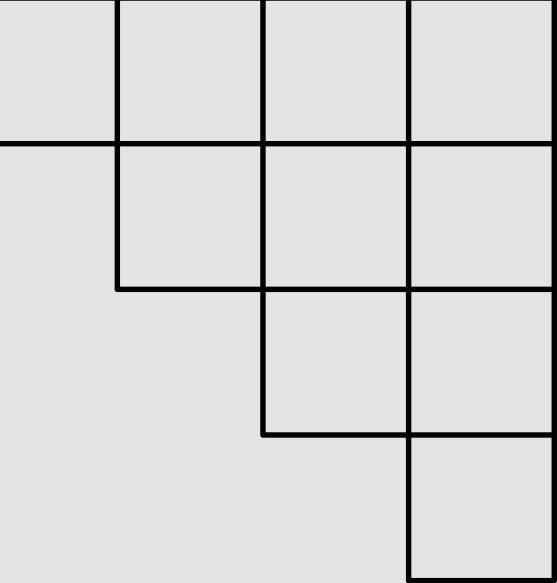
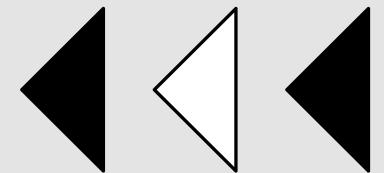
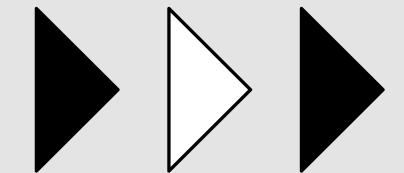
Priority Arbiter

- Used in CPU interrupt controllers
- Bus and memory access control for critical devices
- Real-time embedded systems requiring fast response.

Round Robin Arbiter

- Shared bus scheduling for fairness
- Multi-core or multiprocessor systems
- I/O device or network traffic management



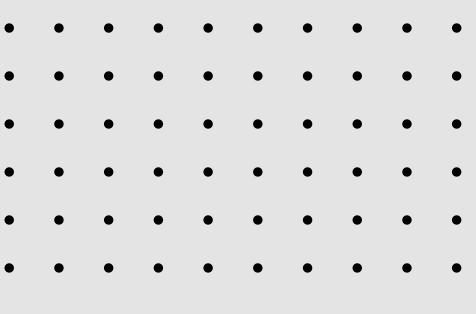
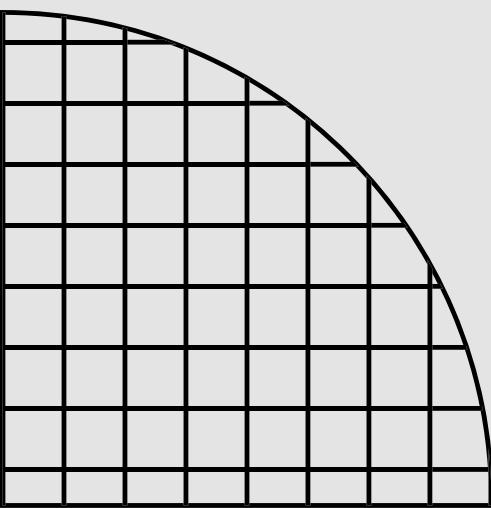


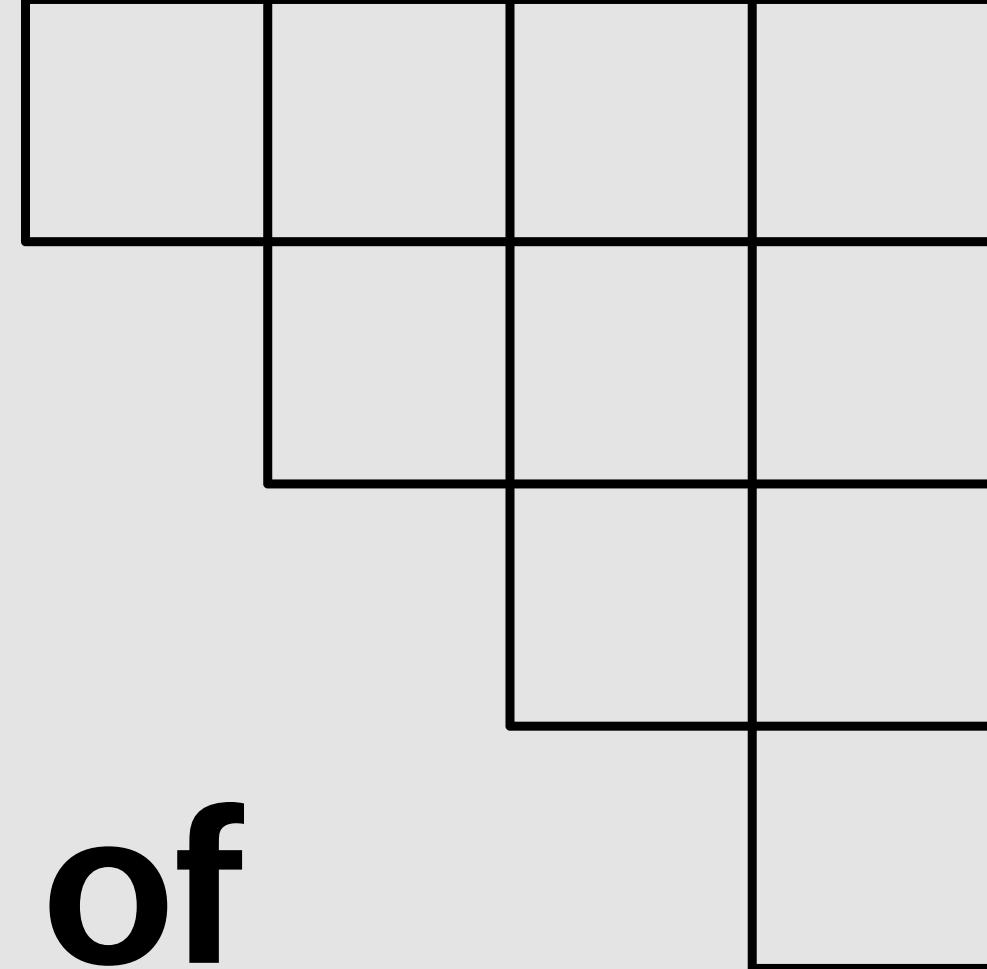
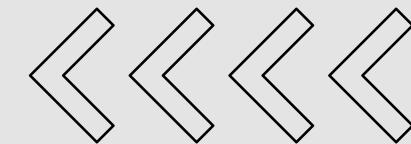
Reference

"RTL implementation and analysis of fixed priority, round robin, and matrix arbiters for the NoC's routers,"

R. Kamal and J. M. Moreno Arostegui,
2016 International Conference on Computing, Communication and
Automation (ICCCA), Greater Noida, India, 2016, pp. 1454-1459,

doi: 10.1109/ICCAA.2016.7813949.

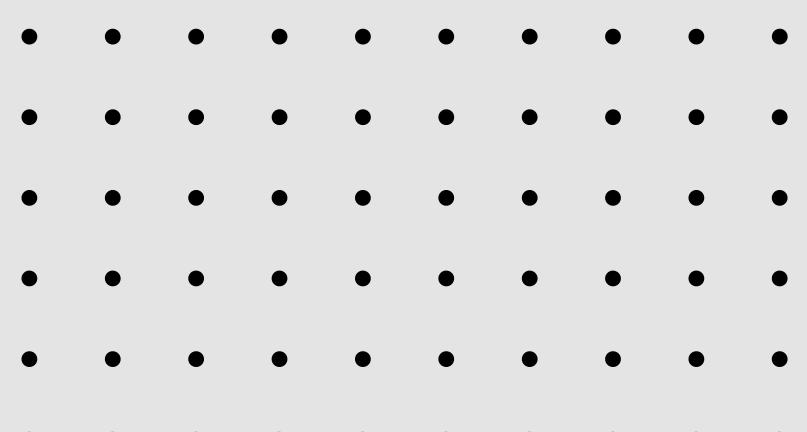
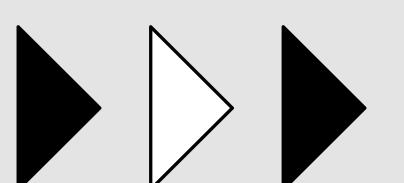
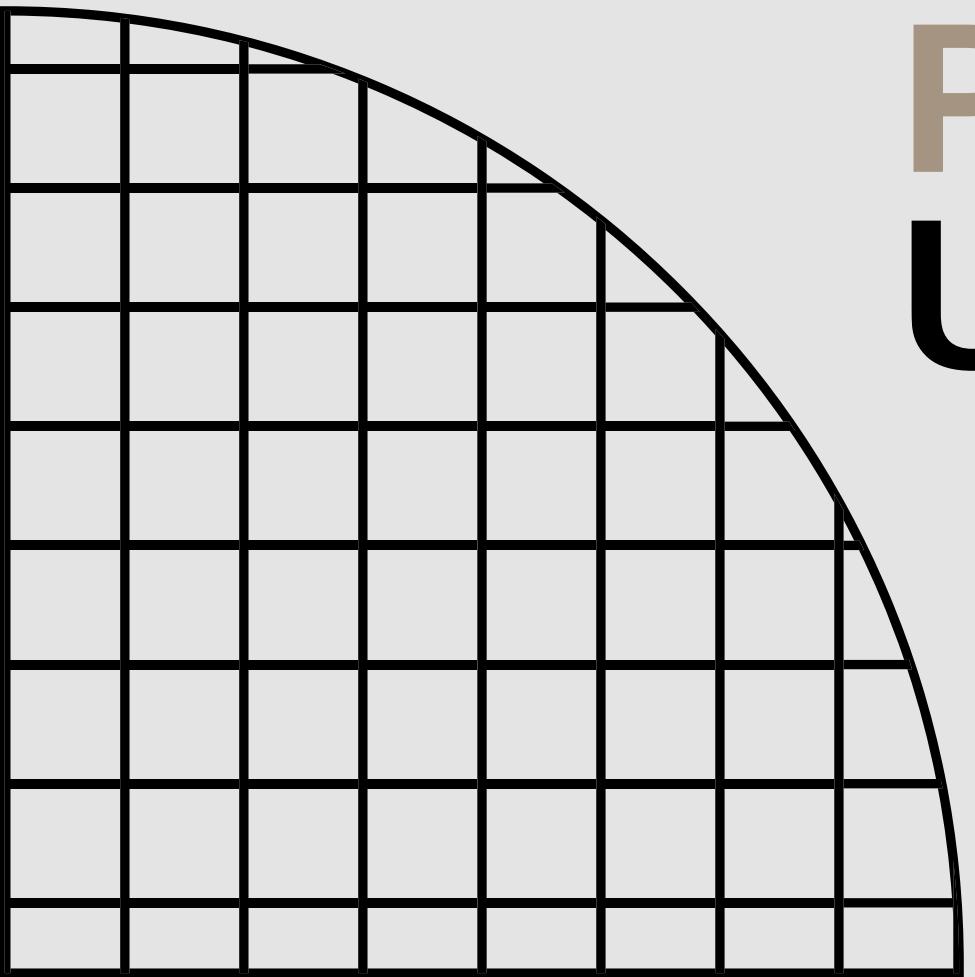




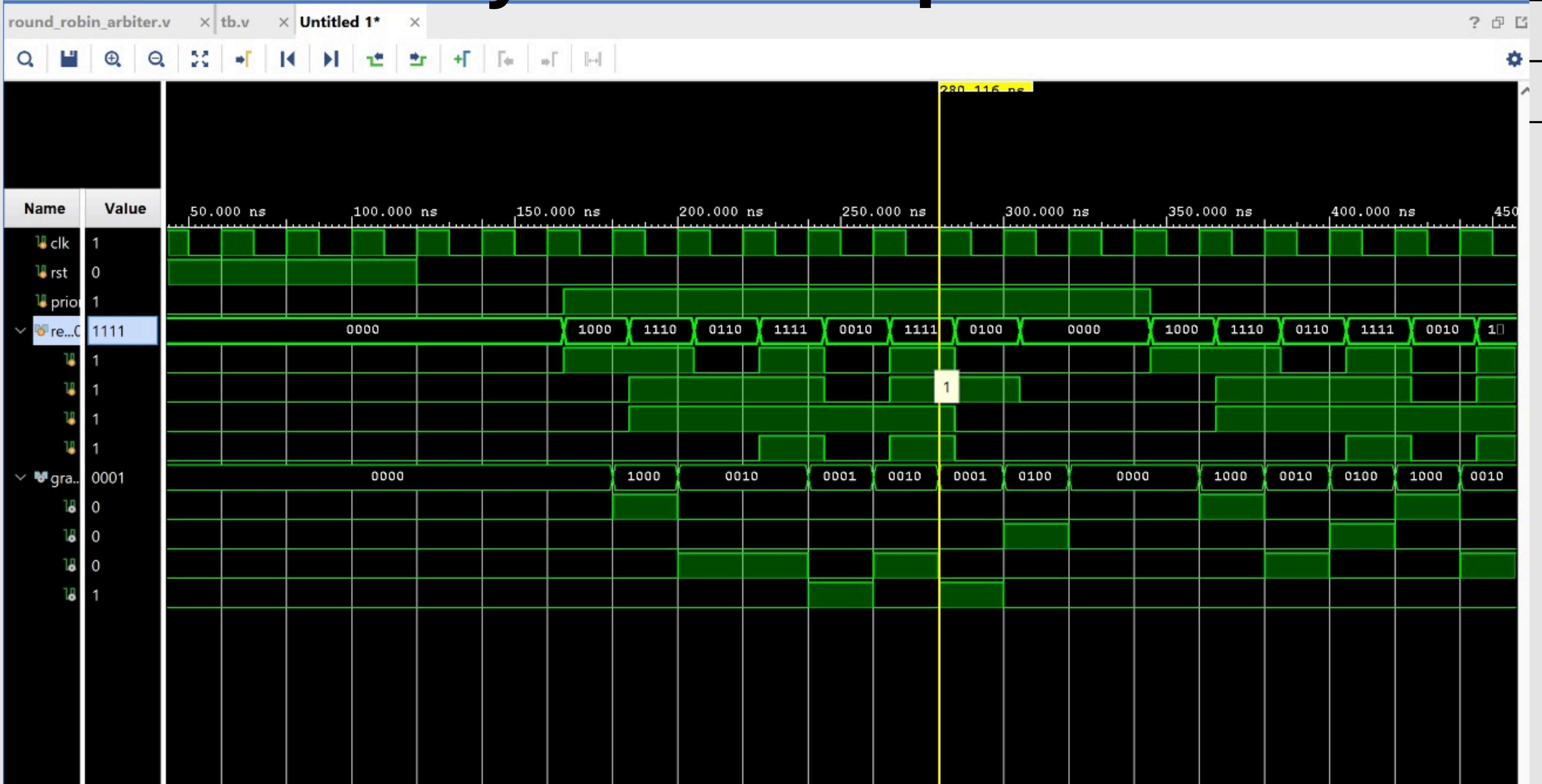
25VL682

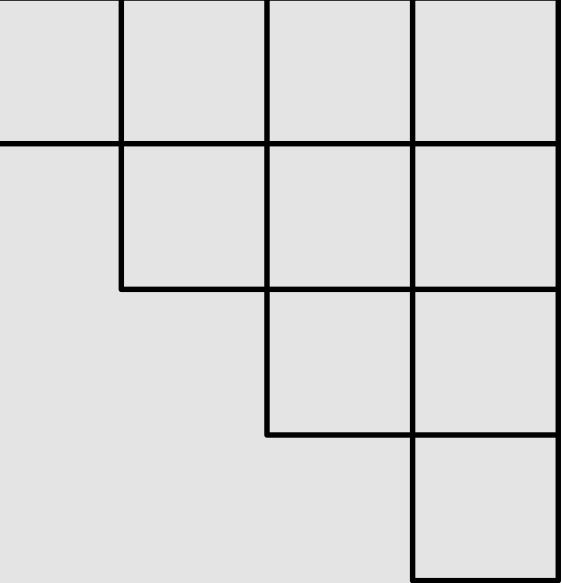
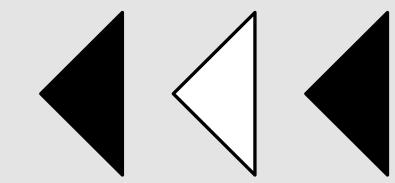
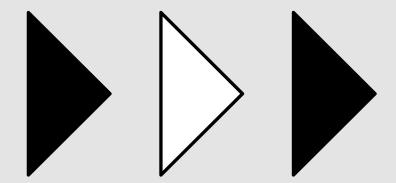
RTL Design and FPGA Synthesis Lab
(Term Work)

Implementation of Round Robin and Priority-Based Arbiter Using Verilog HDL



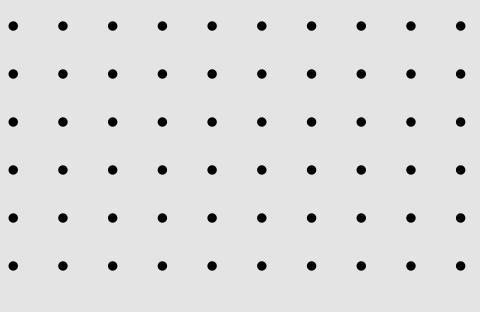
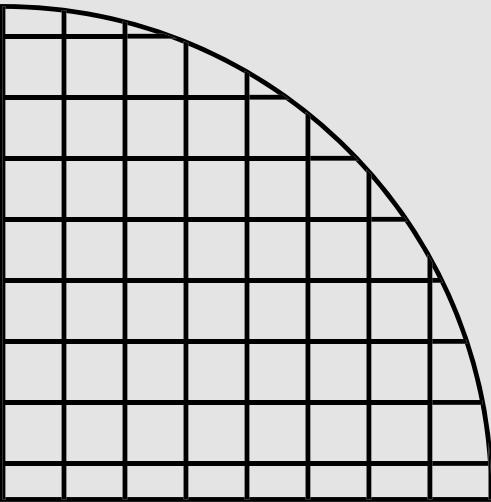
Synthesis Reports

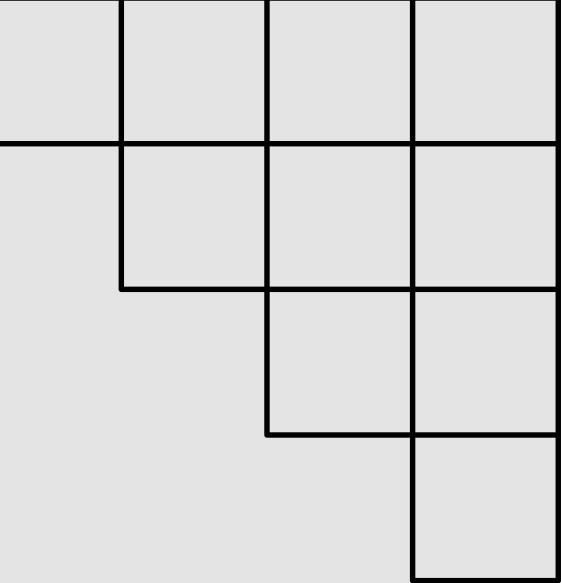
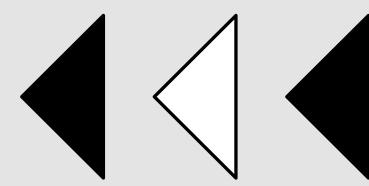
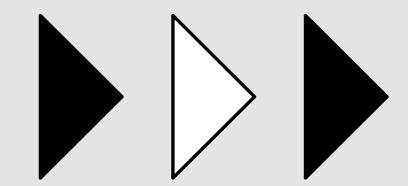




Prior	CLK	Request (4 bit)	Grant (4bit)
1	1	R3	R3
	2	R1,R2,R3	R1
	3	R1,R2	R1
	4	R0,R1,R2,R3	R0
	5	R1	R1
	6	R0,R1,R2,R3	R0
	7	R2	R2

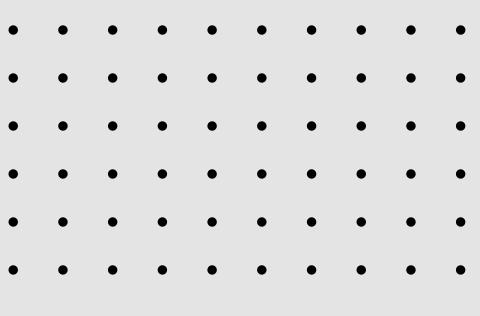
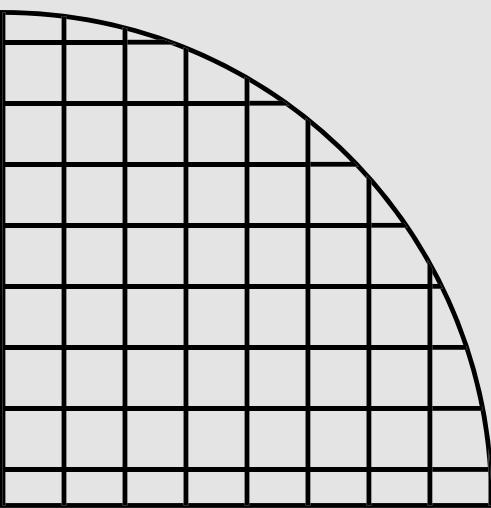
Priority Based Arbiter

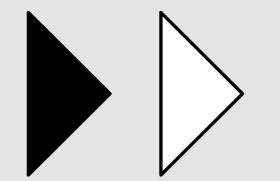




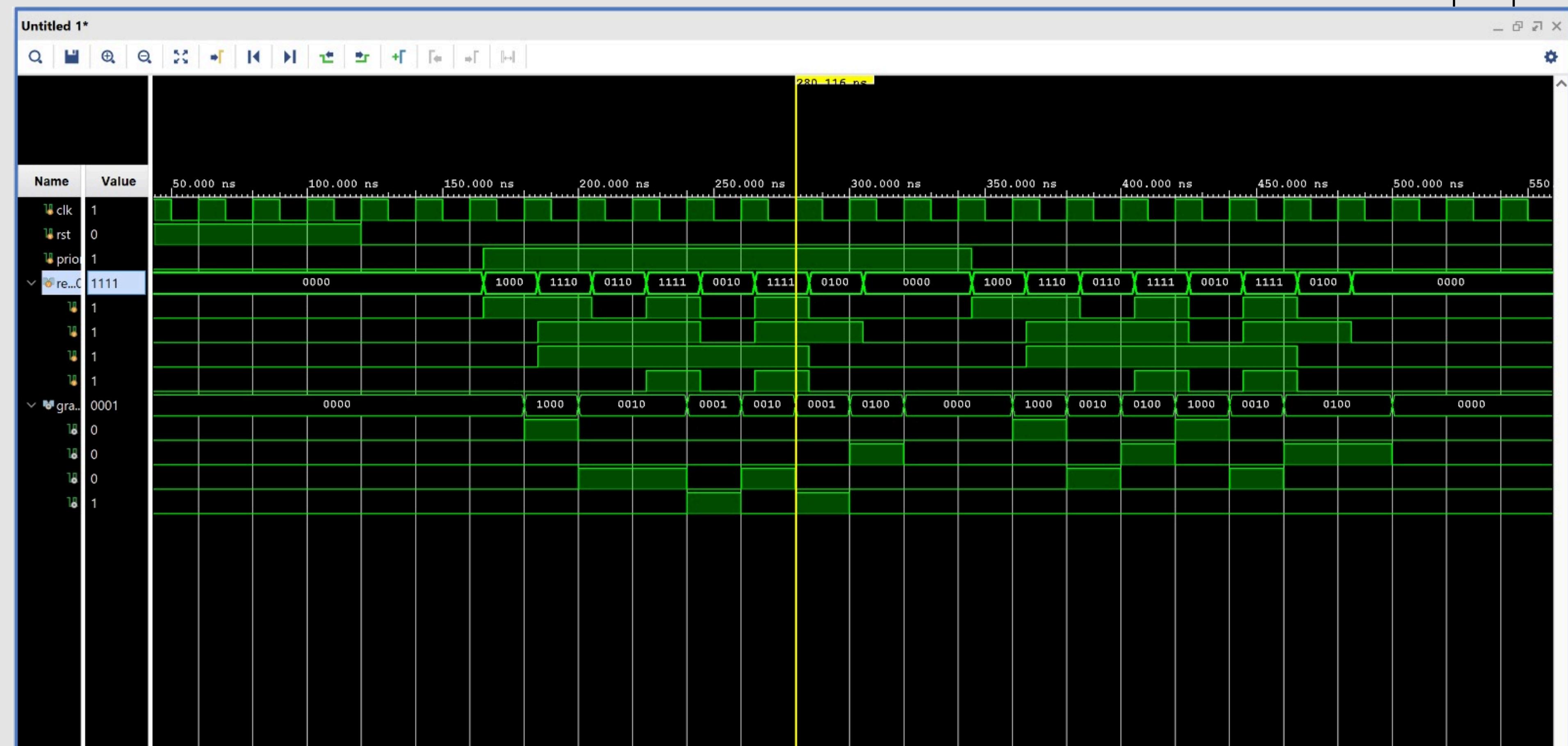
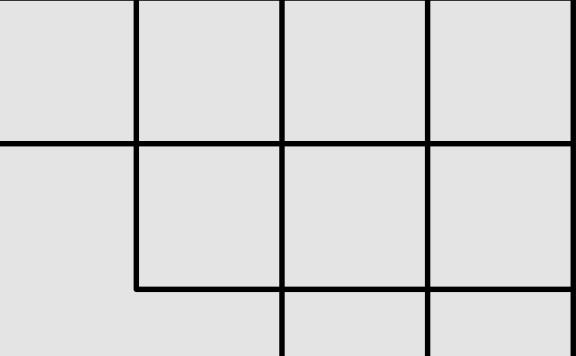
Prior	CLK	Request (4 bit)	Grant (4bit)
0	1	R3	R3
	2	R1,R2,R3	R1
	3	R1,R2	R2
	4	R0,R1,R2,R3	R3
	5	R1	R1
	6	R0,R1,R2,R3	R2
	7	R2	R2

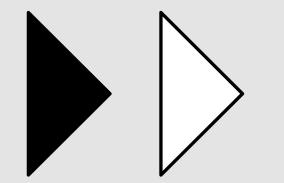
Round Robin



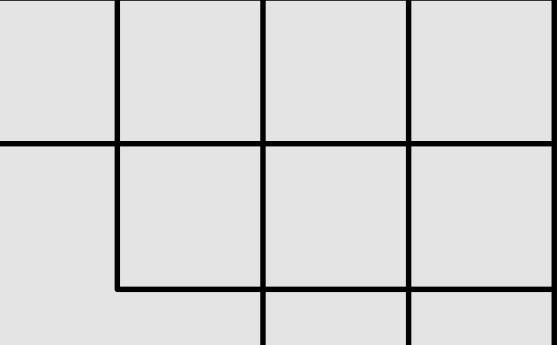
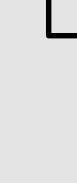
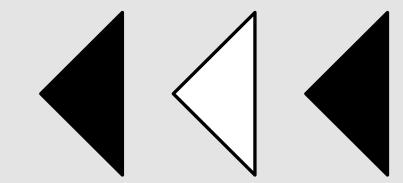


Synthesis Reports

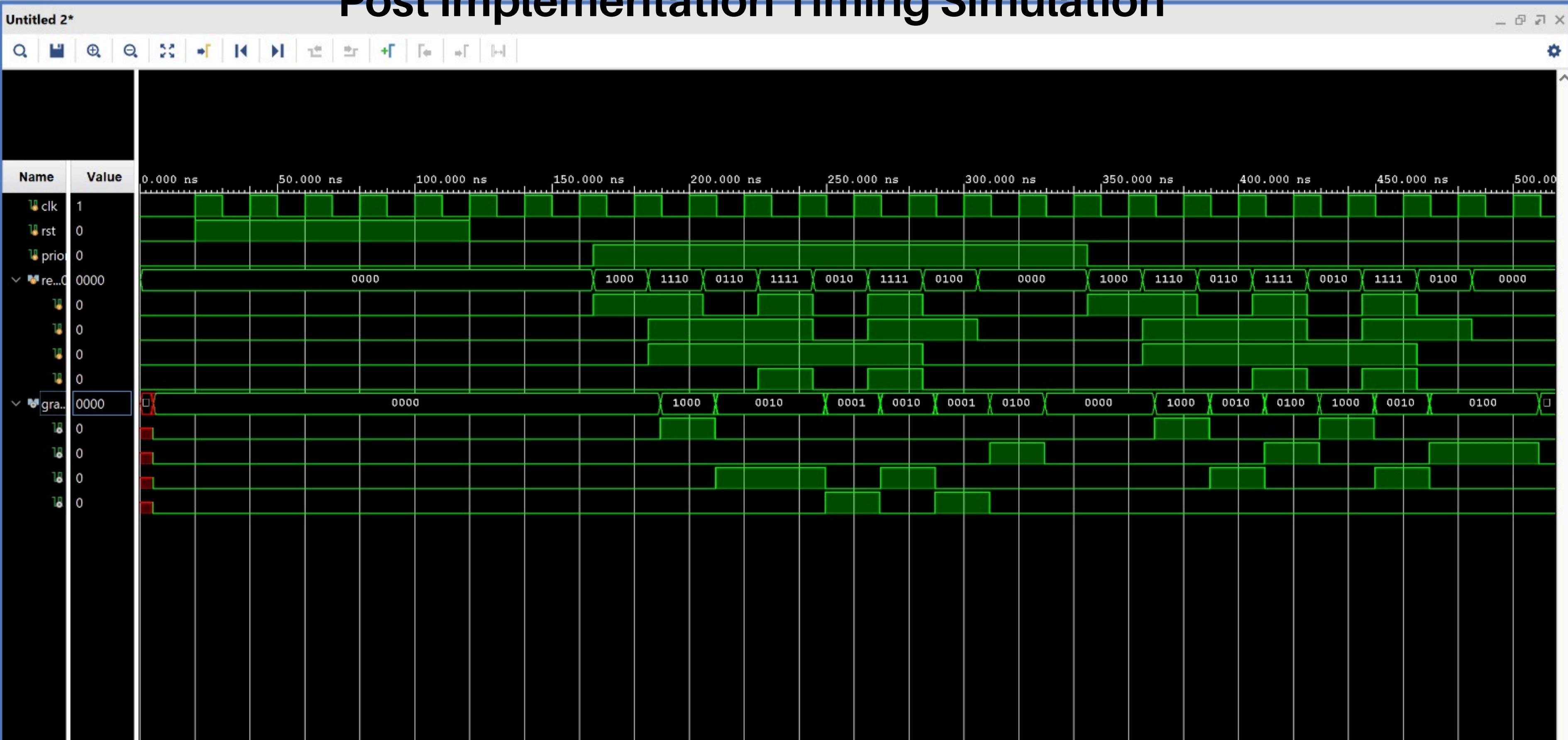




Synthesis Reports



Post Implementation Timing Simulation



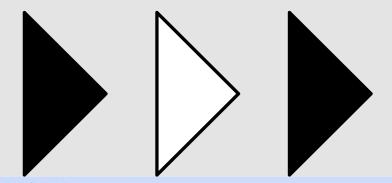
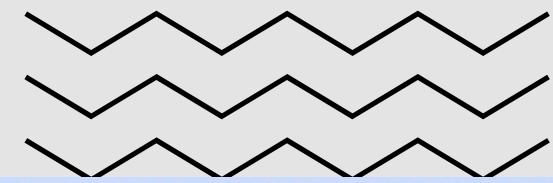
LATENCY

Synthesis: 21.3NS

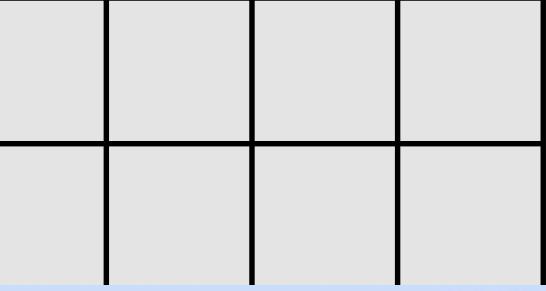
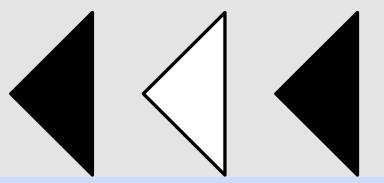
Implementation: 24.613NS

Grant Duration 20ns - 1 clk cycle

Signal	Description
clk	Clock input (period ~20 ns).
rst	Active-high reset signal.
prior	Tracks the previously served client.
req[3:0]	Request inputs from 4 clients.
grant[3:0]	One-hot grant outputs – indicates which request is currently served.

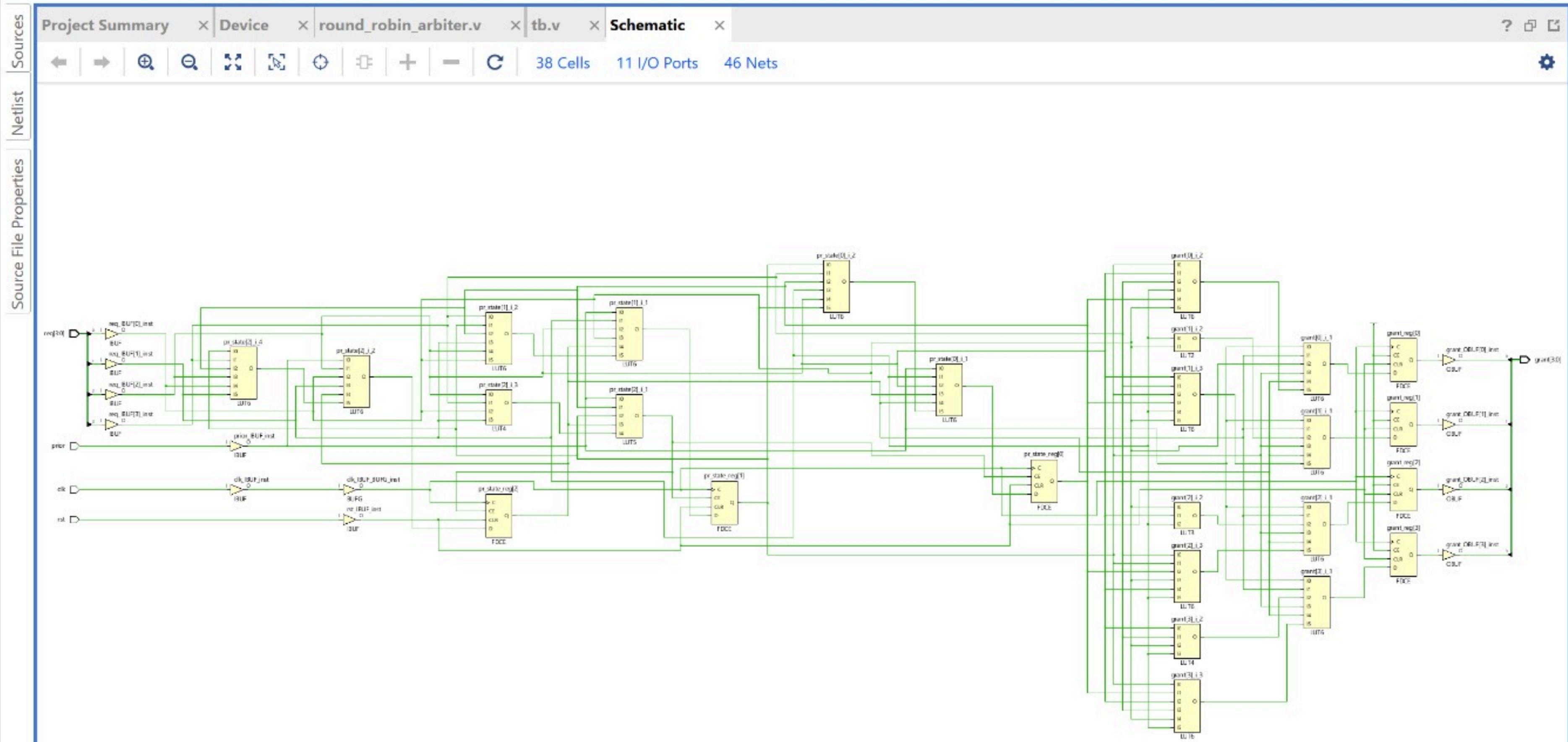


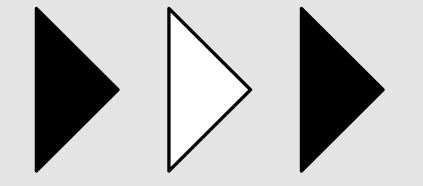
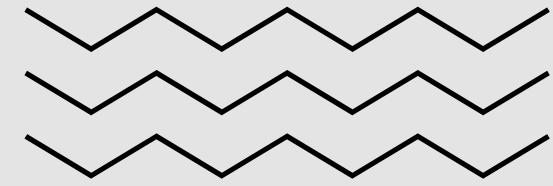
RTL Schematic



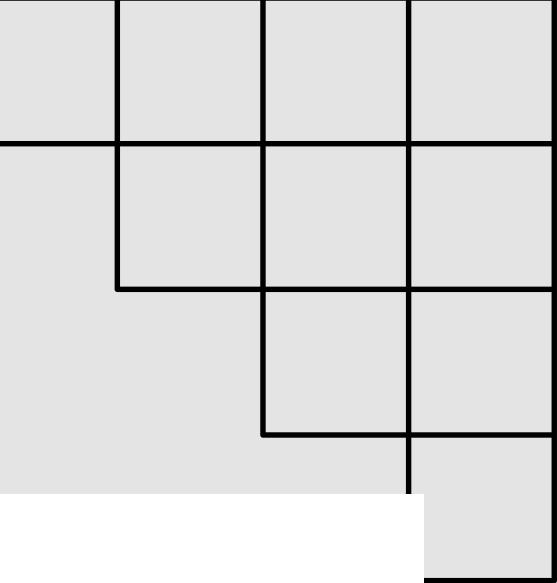
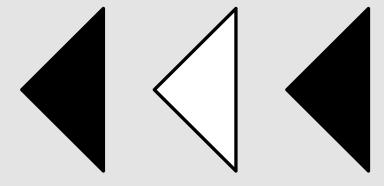
IMPLEMENTED DESIGN - xc7a35tcpg236-1

?



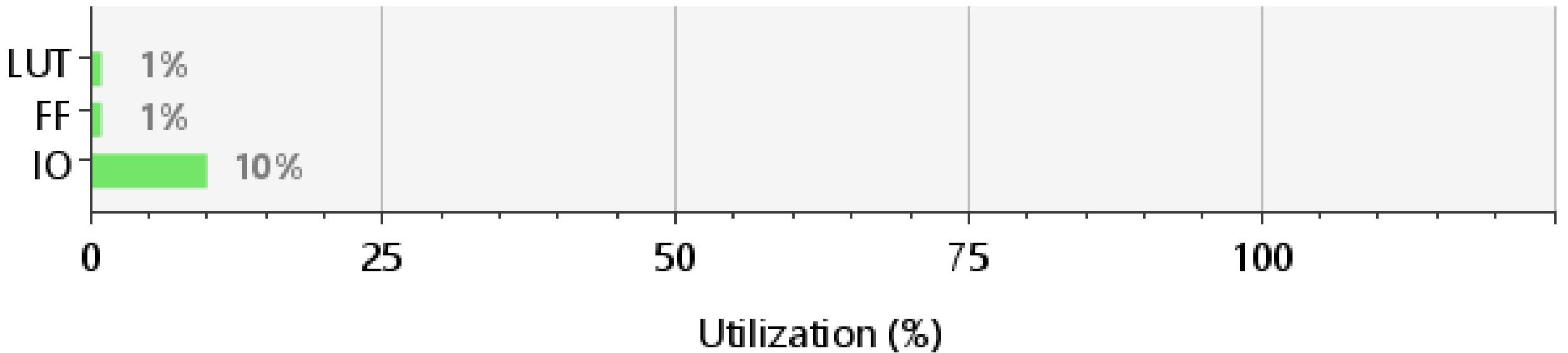


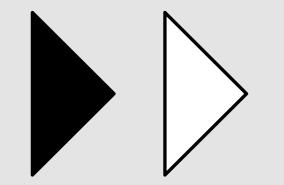
Report Utilization



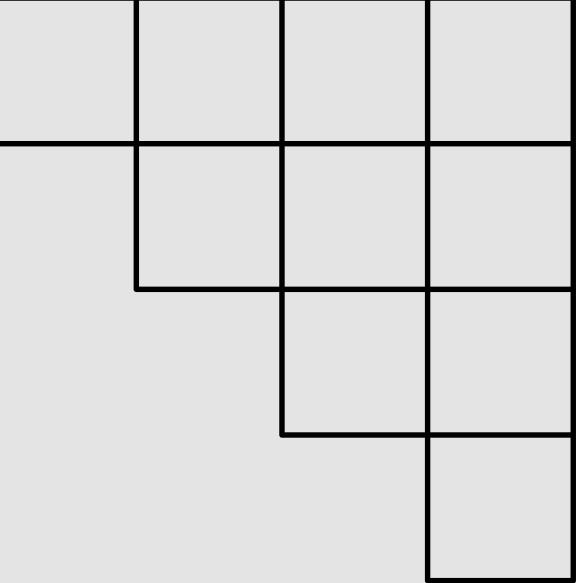
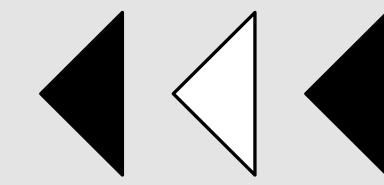
Summary

Resource	Utilization	Available	Utilization %
LUT	17	20800	0.08
FF	7	41600	0.02
IO	11	106	10.38





Power - Area Report



Settings

Summary (1.5 W, Margin: N/A)

Power Supply

Utilization Details

Hierarchical (1.427 W)

Signals (0.089 W)

Data (0.087 W)

Clock Enable (0.003 W)

Set/Reset (0 W)

Logic (0.092 W)

I/O (1.246 W)

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

Total On-Chip Power: 1.5 W

Design Power Budget: Not Specified

Power Budget Margin: N/A

Junction Temperature: 32.5°C

Thermal Margin: 52.5°C (10.4 W)

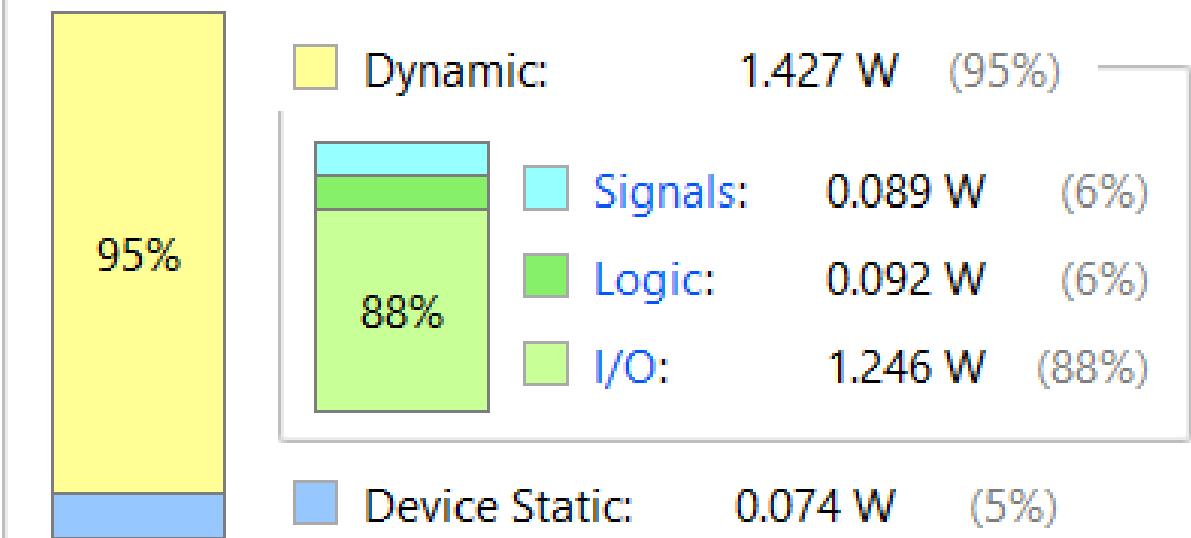
Effective θJA: 5.0°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



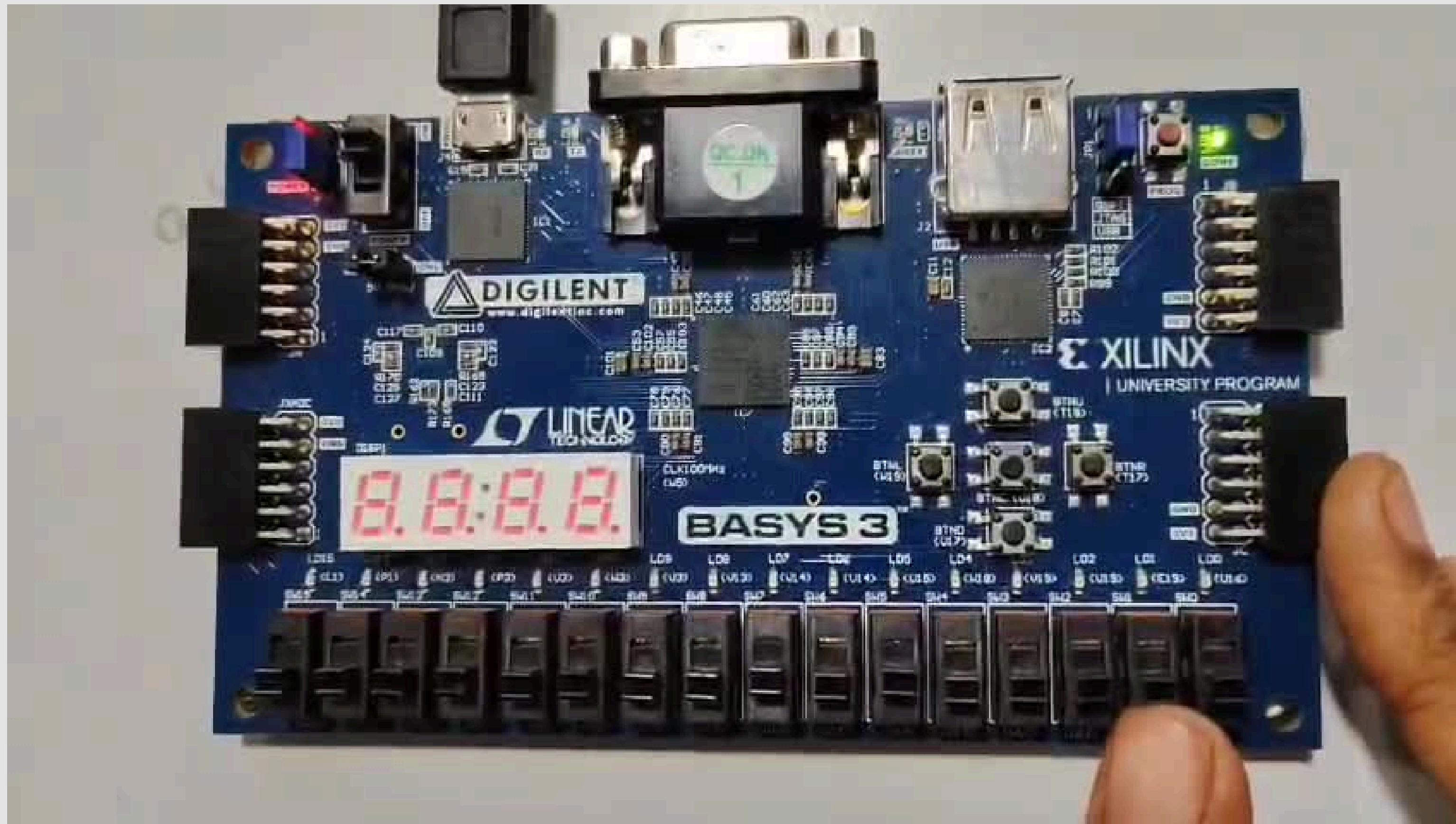
THROUGH PUT

$$\begin{aligned} \text{Total time} &= \text{Time(grant)} - \text{Time (req)} \\ &= 189.4 - 165 \\ &= 24.46 \\ &= 24.5 \text{ ms} \end{aligned}$$

$$\text{Throughput} = \frac{\text{Data per operation}}{\text{Time per operation}}$$

$$\begin{aligned} \text{Throughput} &= \frac{4}{24.5 \times 10^{-9}} \\ &= 163,265,306 \text{ bits/sec} \end{aligned}$$

$$\begin{aligned} \text{LNG} &= \text{LNG(GONG)} - \text{LNG(REQ)} \\ \text{Throughput} &= 163.3 \text{ Mbps} \\ \text{10fa LNG less operation} \end{aligned}$$



Thank You

