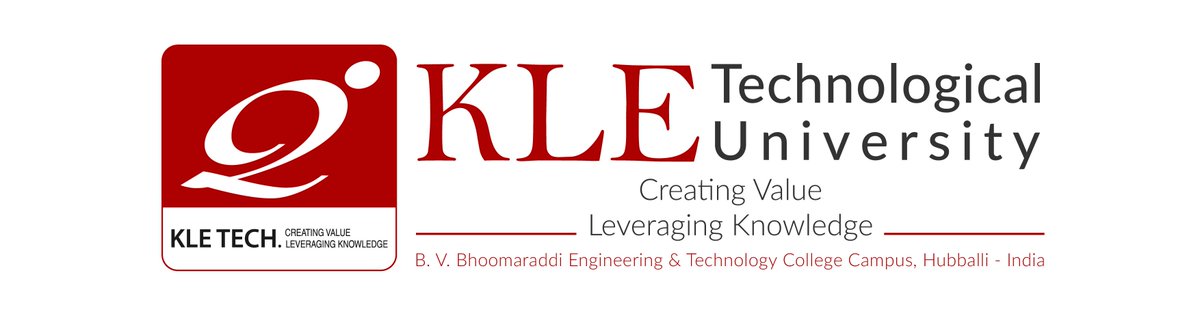
****

**4-BIT ROUND ROBIN ARBITER**

**COURSE PROJECT**

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Name** | **USN** |
| 1. | Sahana M. K | 01FE20BEC238 |
| 2. | Spandana S. B | 01FE20BEC243 |

**THEORY**

An **Arbiter** is used to provide access of data bus whenever there is more than one requester for the bus

An arbiter in which the priority of requesters is set in a way that every requester get resource equally. In **Round-Robin arbiter** the priority of every requester keeps changing with each allocation of resource.

**Uses and advantages**

The basic use of Round-Robin arbiter is to allocate resource to requesters uniformly. In Round-Robin arbiter there is no predefined priority of requesters. The main advantage of Round-Robin arbiter is that it does not lead the requesters to starvation.

**Disadvantages**

In Round-Robin arbiter every requester is treated equally. If we need to give importance to a specific unit we cannot. Even if we want to allocate resource to a particular unit frequently, between two successive allocations Round-Robin arbiter will allocate resource to all requesters once.

The key features of a round-robin arbiter are as follows:

**1. Inputs:**

**Request Signals:** There are four request signals (Req0, Req1, Req2, Req3) from four different requesters. Each signal indicates whether a requester is requesting access to the resource.

**2. Outputs:**

**Grant Signals:** There are four grant signals (Gnt0, Gnt1, Gnt2, Gnt3) corresponding to the request signals. Each grant signal indicates whether a requester is granted access to the resource.

**3. Register:**

A 4-bit register (Reg) is used to store the current state or position of the arbiter.

**4. Initialization:**

The register is initialized to a known state, such as all zeros.

**5. Round-Robin Logic:**

The arbiter follows a sequential round-robin algorithm to determine the next requester to grant access.

The register value represents the current position of the arbiter.

Initially, the register value is incremented by 1 (Mod 4) to determine the first requester to grant access.

**6. Grant Generation:**

The grant signals are generated based on the register value.

If a request signal corresponding to the current register value is active (high), the respective grant signal is asserted (high).

All other grant signals remain de-asserted (low).

**7. Request Priority:**

In a round-robin arbiter, all requesters have equal priority. The grant signals are activated in the order specified by the round-robin sequence.

**8. Grant Duration:**

Once a grant signal is asserted, it remains asserted until the corresponding request signal is de-asserted.

This ensures that the granted requester maintains access to the resource until it releases it.

**9. Register Update:**

After granting access to a requester, the register value is incremented by 1 (Mod 4) to determine the next requester in the round-robin sequence.

The register update occurs on the rising edge of a clock signal, allowing synchronization with other components.

By following this round-robin logic, the arbiter cycles through the requesters, granting them access to the shared resource in a fair manner. The requesters' order is determined by the register value, which is incremented for each grant. This approach ensures that all requesters have an equal opportunity to access the resource.

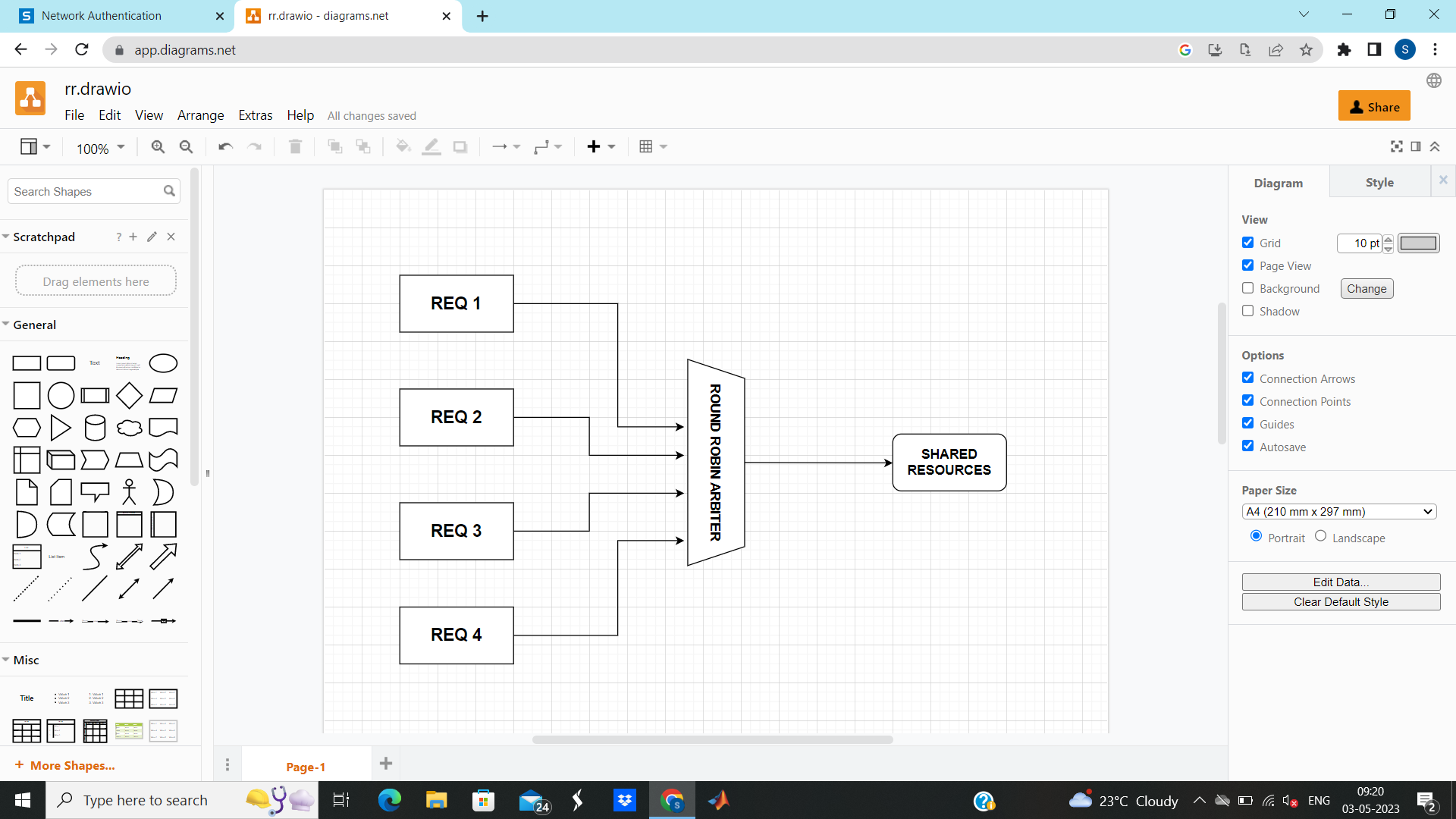
In this Verilog module, the input `Req` is a 4-bit signal representing the request signals from the four requesters. The output `Grant` is a 4-bit signal representing the grant signals for each requester.

The register `Reg` keeps track of the current position of the arbiter. It is incremented by 1 (modulo 4) on every rising edge of the clock signal `clk`. This determines the next requester to grant access.

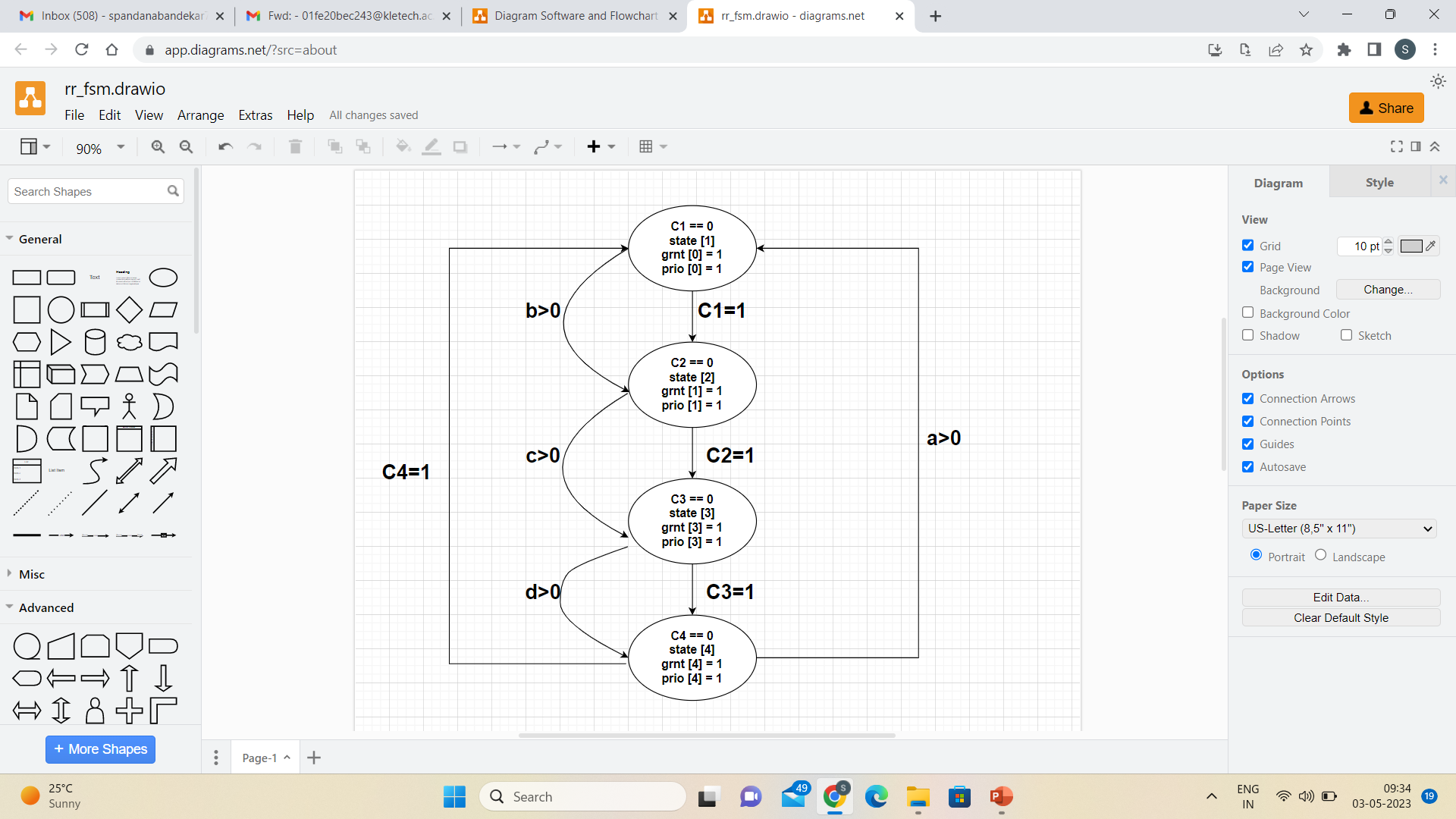
The grant signals are generated by bitwise ANDing the request signals with a bit mask corresponding to the current register value. If a request signal is active (high) for the corresponding requester, the grant signal for that requester is asserted (high) in the `Grant` output. Otherwise, the grant signal remains de-asserted (low).

Remember to provide appropriate testbench and simulate the module to observe the behavior and verify its functionality.

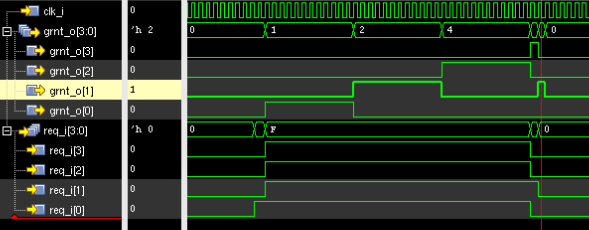
**BLOCK DIAGRAM**



**FINITE STATE MACHINE**



**RESULTS**



**CONCLUSIONS**

A 4-bit round-robin arbiter is a device used to allocate a shared resource among multiple requesters in a cyclic manner. It ensures fair access to the resource by sequentially granting access to each requester in a round-robin fashion.