KU_logo KU_soeng



**The University of Kansas**

**School of Engineering**

**Department of Electrical Engineering and Computer Science**

EECS 645 – Computer Architecture

Fall 2016

Homework 02 (Resource Sharing)

Student Name: Hao Luo Student ID:2737588

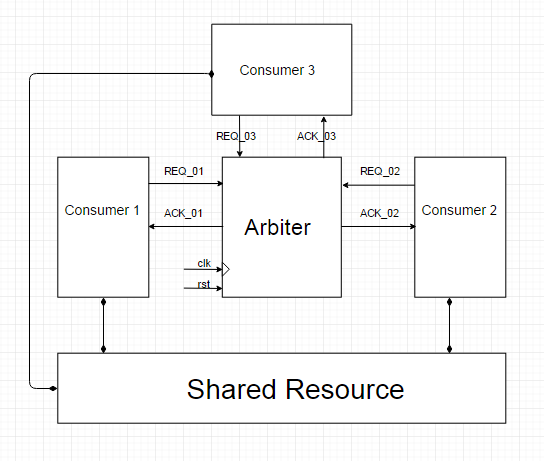
**Homework Problem:**

Given a resource that is to be shared by three consumers such that only one consumer has access to the resource at any given time. The policy of access is preemptive with descending priority. More specifically the policy is as follows:

* Priority is determined by the consumer ID (index), i.e. consumer 1 has the highest priority than all other consumers, and consumer 3 has the lowest priority than all other consumers
* When the resource is idle or being accessed by any of the consumers and the consumers simultaneously request the resource, the consumers are scheduled/preempted according to their priority
* Consumers are responsible for saving and restoring their work if preempted (i.e. the controller does not handle saving/restoring the consumer’s work)

Design a three-consumer arbiter/controller that controls the access to the shared resource and implements the above policy. In the design process, provide the following:

1. The system architecture (block diagram) showing the interface ports to the arbiter including the clock and reset signals.
2. Finite State Machine (FSM) diagram showing all possible states, transitions, and output values.
3. K-maps for internal state and output variables.
4. Boolean expressions for internal state and output variables.
5. Detailed logic diagram using synchronous memory elements showing internal connections and external interfaces.
6. Complete description of the arbiter using both *structural* and *behavioral* VHDL.
7. Simulation results.

****

**Figure 1: System architecture**

**Table 1: Input Code Assignment**

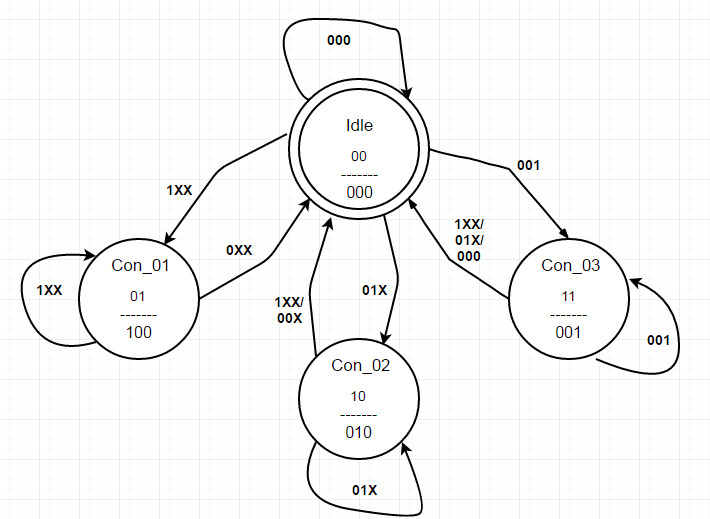
|  |  |  |  |
| --- | --- | --- | --- |
| Input Combinations (Input Codes) | | | Input Description |
| REQ\_01 | REQ\_02 | REQ\_03 |
| 0 | 0 | 0 | No requests |
| 1 | 0 | 0 | Consumers 1 request resource |
| 0 | 1 | 0 | Consumers 2 request resource |
| 0 | 0 | 1 | Consumers 3 request resource |
| 1 | 1 | 0 | Both consumer 1 and consumer 2 request resource |
| 1 | 0 | 1 | Both consumer 1 and consumer 3 request resource |
| 0 | 1 | 1 | Both consumer 2 and consumer 3 request resource |
| 1 | 1 | 1 | Both consumers request resource |

**Table 2: Output Code Assignment**

|  |  |  |  |
| --- | --- | --- | --- |
| Output Combinations (Output Codes) | | | Output Description |
| ACK\_01 | ACK\_02 | ACK\_03 |
| 0 | 0 | 0 | None granted access to resource |
| 1 | 0 | 0 | Consumer 1 granted access to resource |
| 0 | 1 | 0 | Consumer 2 granted access to resource |
| 0 | 0 | 1 | Consumer 3 granted access to resource |
| 1 | 1 | 0 | Unexpected output (Consumer 1 granted access) |
| 1 | 0 | 1 | Unexpected output (Consumer 1 granted access) |
| 0 | 1 | 1 | Unexpected output (Consumer 2 granted access) |
| 1 | 1 | 1 | Unexpected output (Consumer 1 granted access) |

**Table 3: State Code Assignment**

|  |  |  |
| --- | --- | --- |
| State Description | State Codes | |
| S\_01 | S\_02 |
| Resource is idle (No access) | 0 | 0 |
| Resource is required by consumer 1 (Con\_01) | 0 | 1 |
| Resource is required by consumer 2 (Con\_02) | 1 | 0 |
| Resource is required by consumer 3 (Con\_03) | 1 | 1 |

****

**Figure 2: State transition diagram**

**Table 4: State transition table**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Next State**  **(S1 , S2)** | | **Inputs**  **(REQ1 , REQ2 , REQ3)** | | | | | | | |
| **000** | **001** | **011** | **010** | **110** | **111** | **101** | **100** |
| **Current State**  **(S1 , S2)** | **00** | **00** | **11** | **10** | **10** | **01** | **01** | **01** | **01** |
| **01** | **00** | **00** | **00** | **00** | **01** | **01** | **01** | **01** |
| **11** | 00 | 11 | 00 | 00 | 00 | 00 | 00 | 00 |
| **10** | **00** | **00** | **10** | **10** | **00** | **00** | **00** | **00** |

**Table 5: K-map for the state variable *S1***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Next State**  **(S1)** | | **Inputs**  **(REQ1 , REQ2 , REQ3)** | | | | | | | |
| **000** | **001** | **011** | **010** | **110** | **111** | **101** | **100** |
| **Current State**  **(S1 , S2)** | **00** |  | **1** | **1** | **1** |  |  |  |  |
| **01** |  |  |  |  |  |  |  |  |
| **11** |  | 1 |  |  |  |  |  |  |
| **10** |  |  | **1** | **1** |  |  |  |  |



**Table 6: K-map for the state variable *S2***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Next State**  **(S2)** | | **Inputs**  **(REQ1 , REQ2 , REQ3)** | | | | | | | |
| **000** | **001** | **011** | **010** | **110** | **111** | **101** | **100** |
| **Current State**  **(S1 , S2)** | **00** |  | **1** |  |  | **1** | **1** | **1** | **1** |
| **01** |  |  |  |  | **1** | **1** | **1** | **1** |
| **11** |  | 1 |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |



**Table 7: Output transition table**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Outputs**  **(ACK1 , ACK2 , ACK3)** | | **Inputs**  **(REQ1 , REQ2 , REQ3)** | | | | | | | |
| **000** | **001** | **011** | **010** | **110** | **111** | **101** | **100** |
| **Current State**  **(S1 , S2)** | **00** | **000** | **000** | **000** | **000** | **000** | **000** | **000** | **000** |
| **01** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| **11** | 001 | 001 | 001 | 001 | 001 | 001 | 001 | 001 |
| **10** | **010** | **010** | **010** | **010** | **010** | **010** | **010** | **010** |

**Table 8: K-map for the output variable ACK1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Outputs**  **(ACK1)** | | **Inputs**  **(REQ1 , REQ2 , REQ3)** | | | | | | | |
| **000** | **001** | **011** | **010** | **110** | **111** | **101** | **100** |
| **Current State**  **(S1 , S2)** | **00** |  |  |  |  |  |  |  |  |
| **01** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| **11** |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |



**Table 9: K-map for the output variable ACK2**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Outputs**  **(ACK2)** | | **Inputs**  **(REQ1 , REQ2 , REQ3)** | | | | | | | |
| **000** | **001** | **011** | **010** | **110** | **111** | **101** | **100** |
| **Current State**  **(S1 , S2)** | **00** |  |  |  |  |  |  |  |  |
| **01** |  |  |  |  |  |  |  |  |
| **11** |  |  |  |  |  |  |  |  |
| **10** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |



**Table 10: K-map for the output variable ACK3**

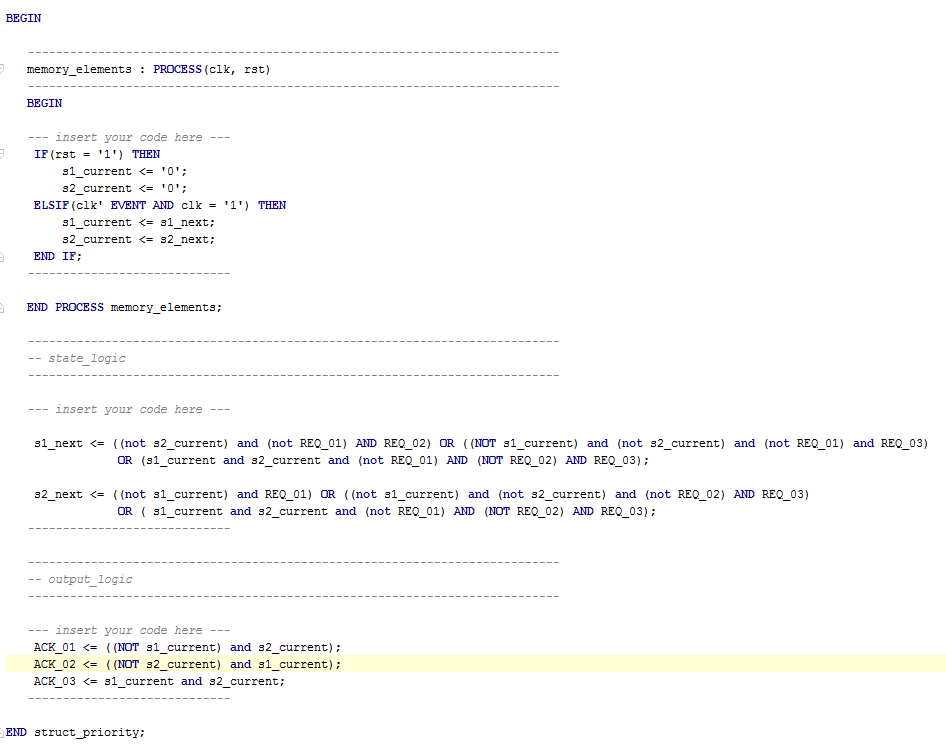
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Outputs**  **(ACK3)** | | **Inputs**  **(REQ1 , REQ2 , REQ3)** | | | | | | | |
| **000** | **001** | **011** | **010** | **110** | **111** | **101** | **100** |
| **Current State**  **(S1 , S2)** | **00** |  |  |  |  |  |  |  |  |
| **01** |  |  |  |  |  |  |  |  |
| **11** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **10** |  |  |  |  |  |  |  |  |



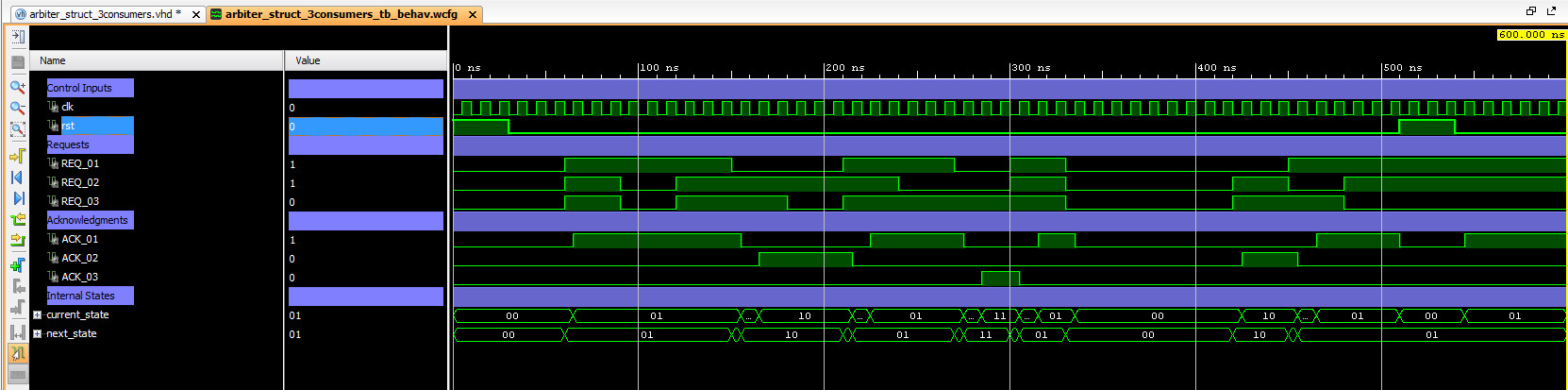
**Figure 3: Synchronous logic diagram**



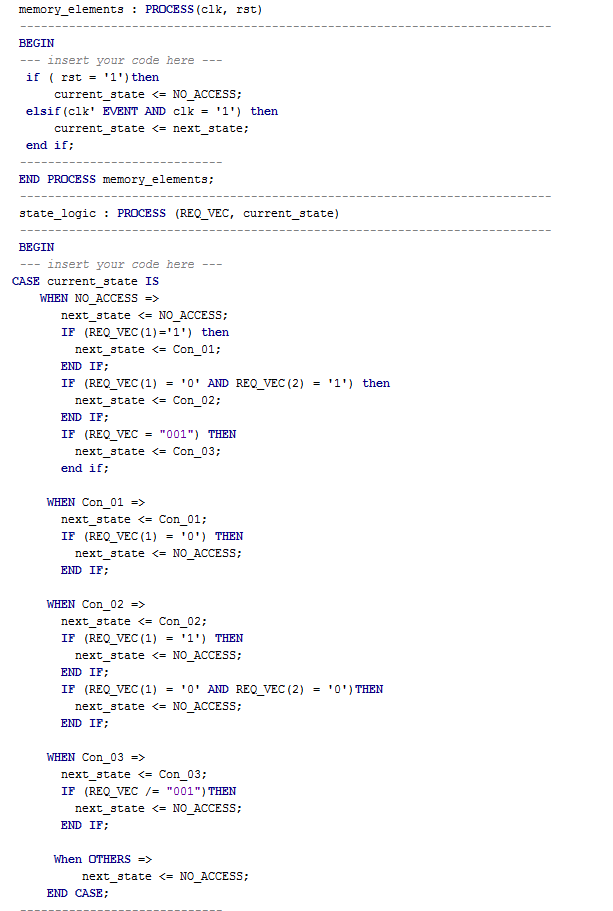
**Structural VHDL code**

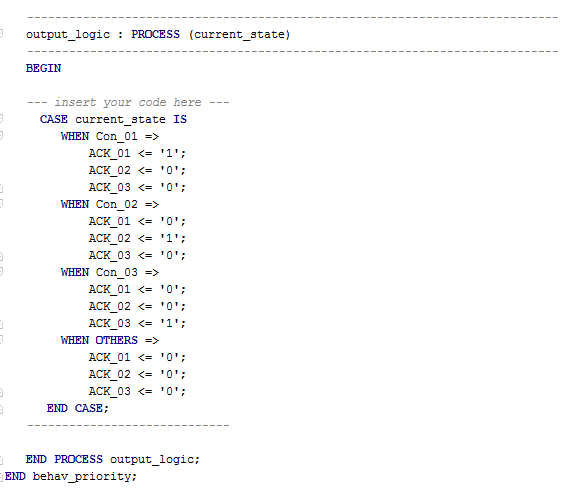
****

**Simulation Results**

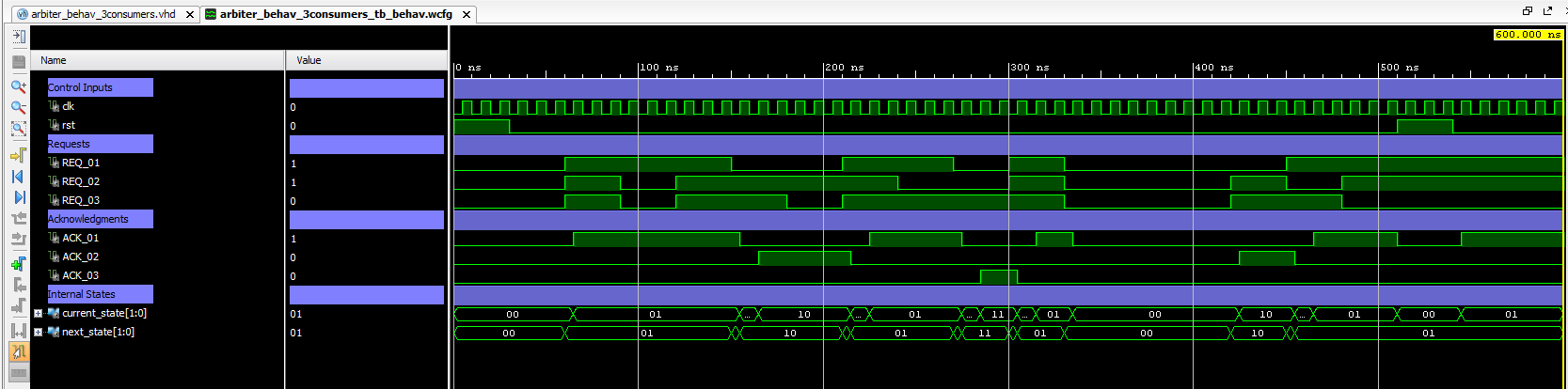
****

**Behavioral VHDL code**





**Simulation Results**

****