**MULTIPLEXER IN VERILOG**

**LAB # 07**



**Fall 2023**

**CSE-304L Computer Organization and Architecture Lab**

Submitted by: **Ali Asghar**

Registration No.: **21PWCSE2059**

Class Section: **C**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Dr. Bilal Habib**

Date:

**7th December 2023**

**Department of Computer Systems Engineering**

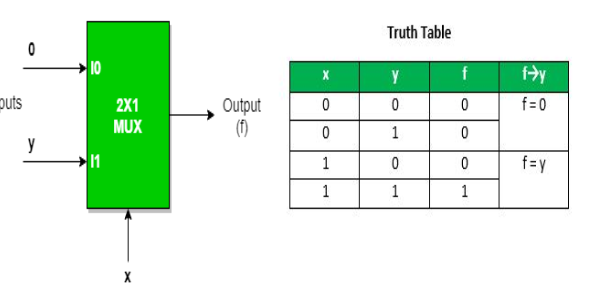
**University of Engineering and Technology, Peshawar**

**ASSESSMENT RUBRICS COA LABS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LAB REPORT ASSESSMENT** | | | | |
| **Criteria** | **Excellent** | **Average** | **Nill** | **Marks Obtained** |
| 1. **Objectives of Lab** | All objectives of lab are properly covered  [Marks 10] | Objectives of lab are partially covered  [Marks 5] | Objectives of lab are not shown  [Marks 0] |  |
| 1. **MIPS instructions with**   **Comments and proper indentations.** | All the instructions are well written with comments explaining the code and properly indented  [Marks 20] | Some instructions are missing are poorly commented code  [Marks 10] | The instructions are not properly written  [Marks 0] |  |
| 1. **Simulation run without error and warnings** | The code is running in the simulator without any error and warnings  [Marks 10] | The code is running but with some warnings or errors.  [Marks 5] | The code is written but not running due to errors  [Marks 0] |  |
| 1. **Procedure** | All the instructions are written with proper procedure  [Marks 20] | Some steps are missing  [Marks 10] | steps are totally missing  [Marks 0] |  |
| 1. **OUTPUT** | Proper output of the code written in assembly  [Marks 20] | Some of the outputs are missing  [Marks 10] | No or wrong output  [Marks 0] |  |
| 1. **Conclusion** | Conclusion about the lab is shown and written  [Marks 20] | Conclusion about the lab is partially shown  [Marks 10] | Conclusion about the lab is not shown[Marks0]  [Marks 0] |  |
| 1. **Cheating** |  |  | Any kind of cheating will lead to 0 Marks |  |
| Total Marks Obtained:\_\_\_\_\_\_\_\_\_\_  Instructor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | |

**Task 1:**

Implement 2X1 MUX in Verilog.



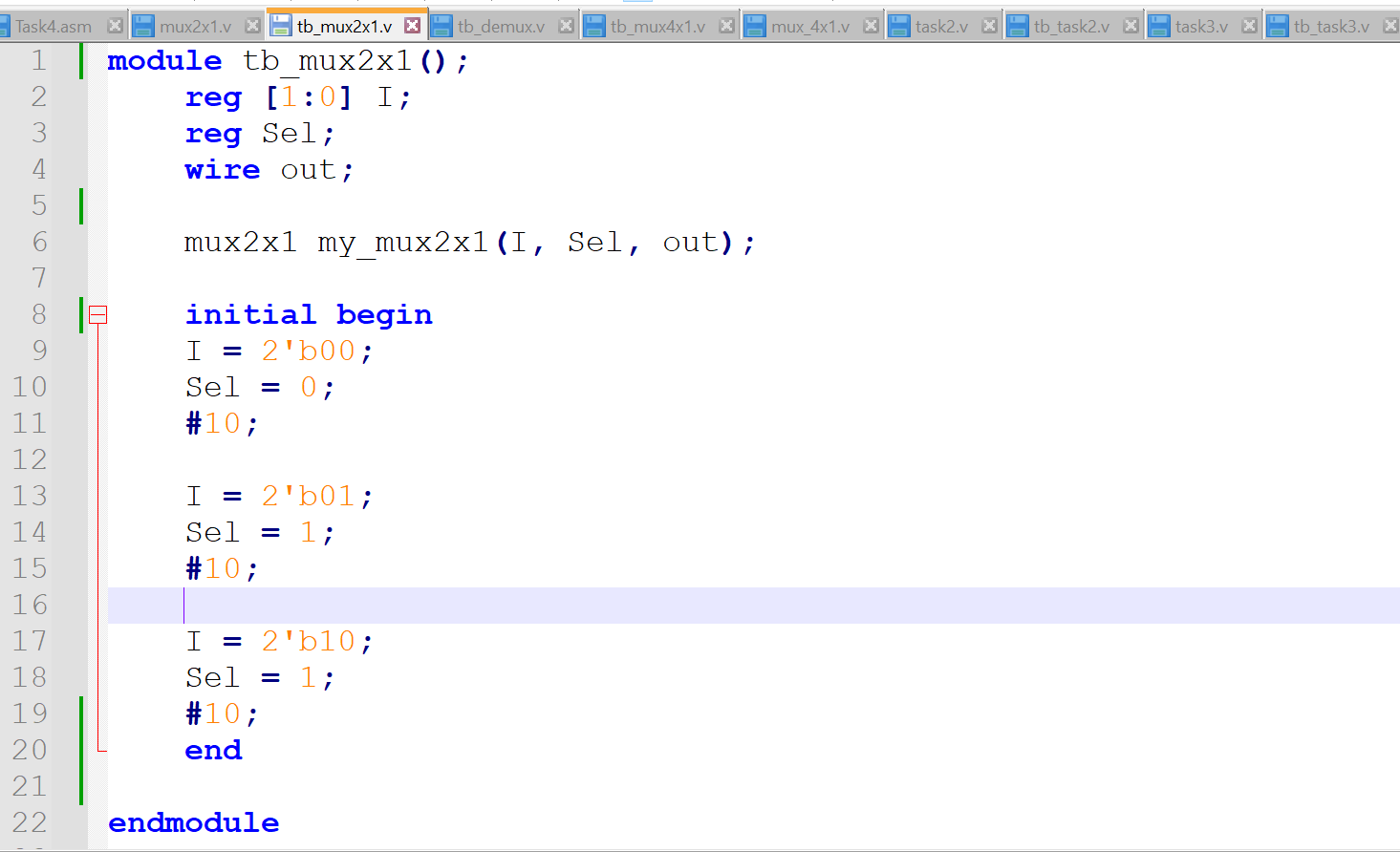
**Code:**

**DUT Code:**

**A screenshot of a computer program

Description automatically generated**

**Test bench Code:**

****

**Output:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Task 2:**

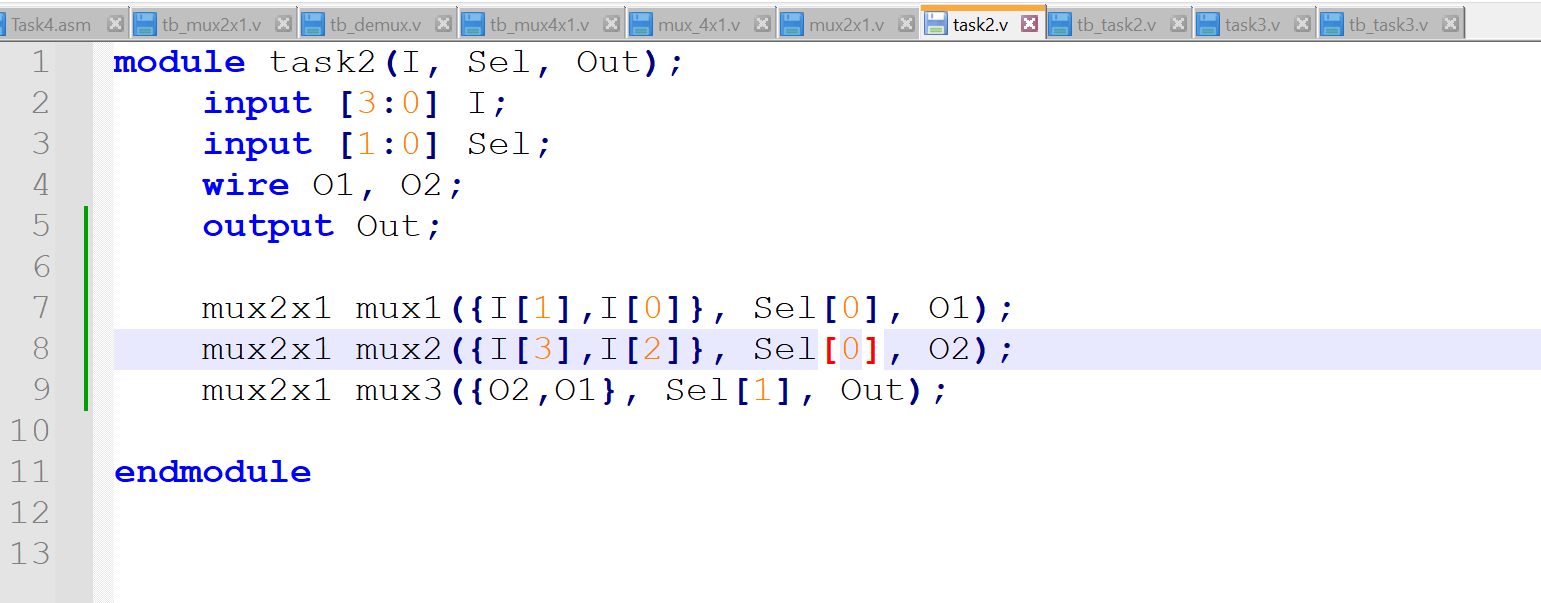
Implement 4X1 MUX using 2X1 MUX.

A diagram of a computer

Description automatically generated

**Code:**

**DUT Code:**

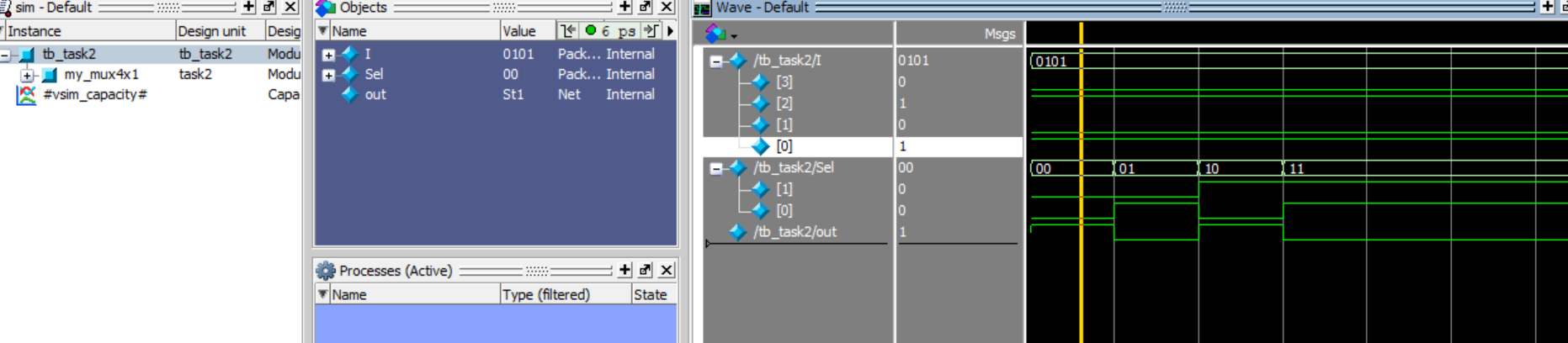
****

**Test bench Code:**

**A screenshot of a computer

Description automatically generated**

**Output:**

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

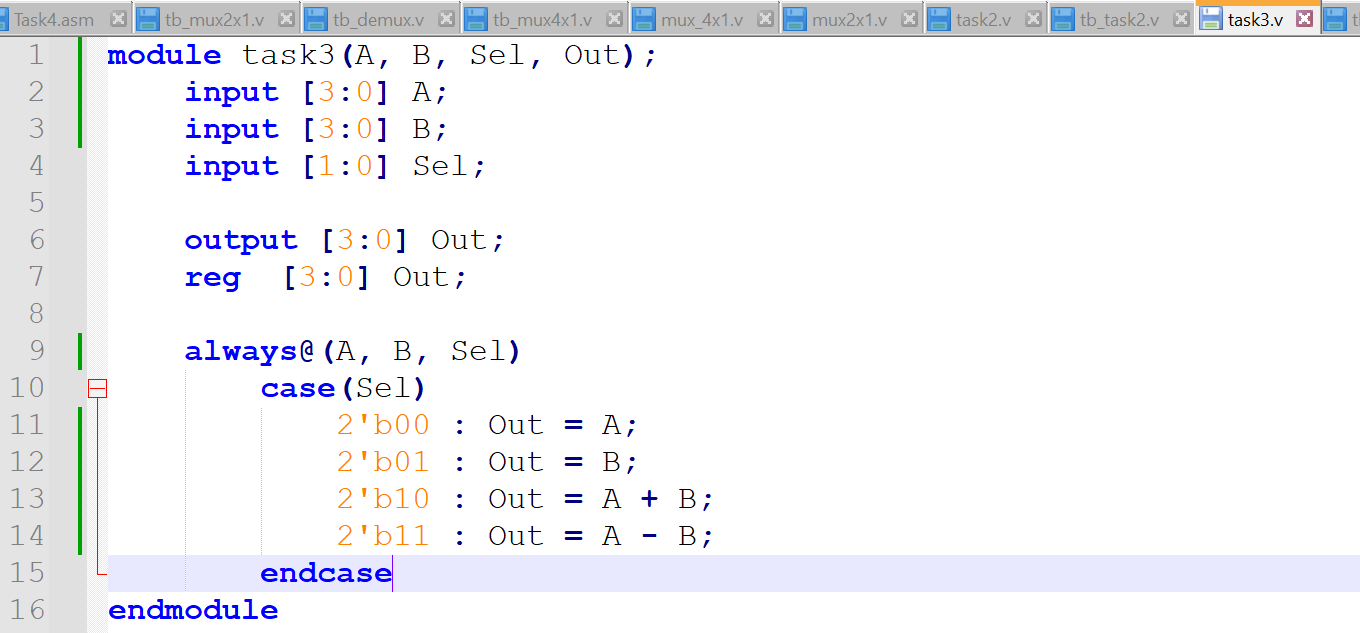
Description automatically generated**

**Task 3:**

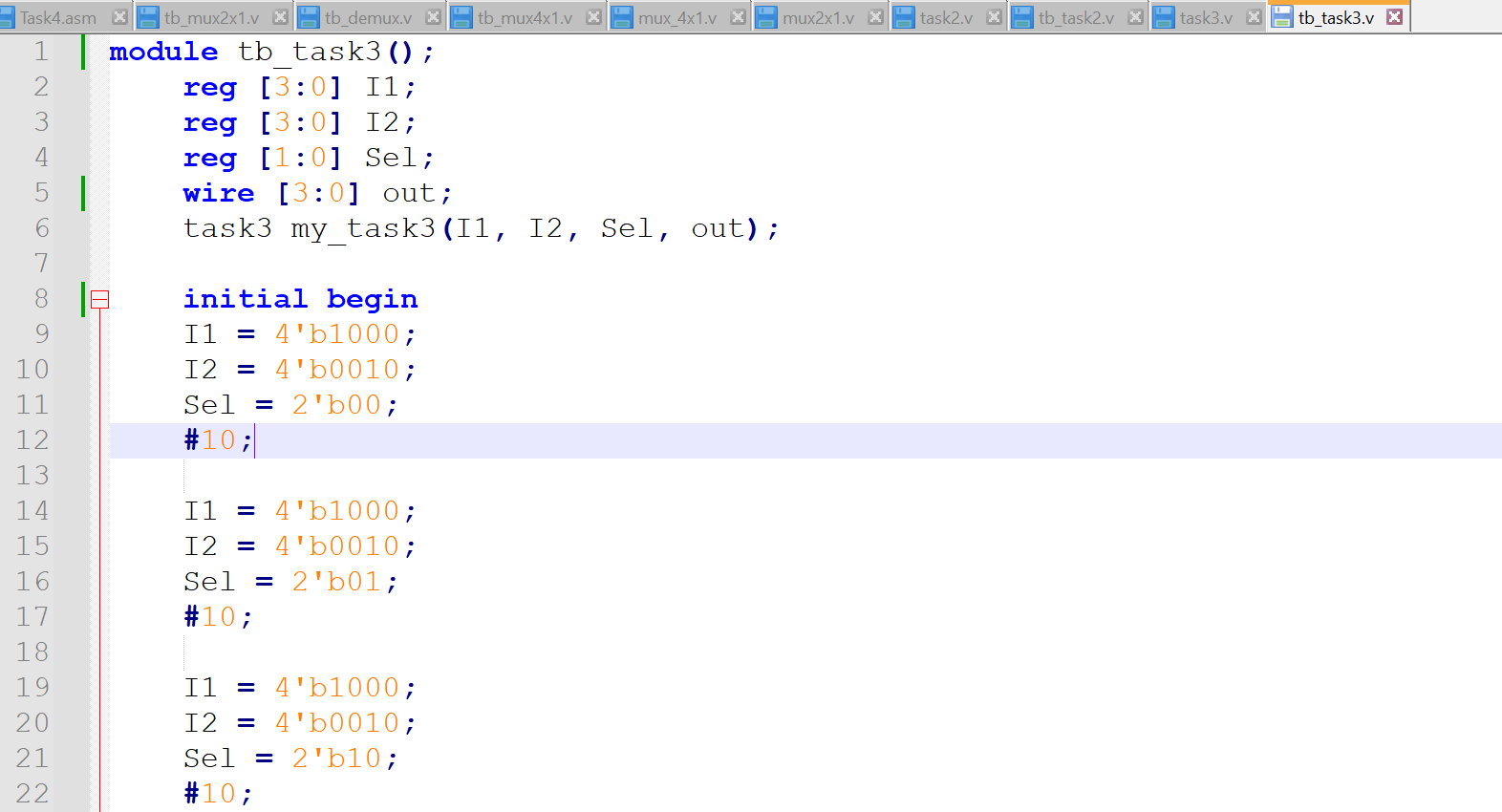
Design MUX where if the select to a MUX is 00 it will output A , 01 the output will be B , for 10 the output will be A+B and for 11 the output will be A-B.

**Code:**

**DUT Code:**

****

**Test bench Code:**

****

**A close-up of a white background

Description automatically generated**

**Output:**

**A screenshot of a computer

Description automatically generated**

**Conclusion:**

In this lab, I learned how to implement Multiplexers using always statement (behavior level modelling) in Verilog.