Dr. A. Q. Khan Institute of Computer Science and

Information Technology

Department of Computer Engineering

**Lab Report-3**

Computer Architecture Lab

Submitted To

Mam Muneeba Mubarik

Submitted By

Muhammad Bilal

BSCS- V B

**Lab Rubrics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab Report** | **Coding** | **Logic** | **Total** |
| **4** | **4** | **4** | **12** |
|  |  |  |  |

**Half Subtractor**

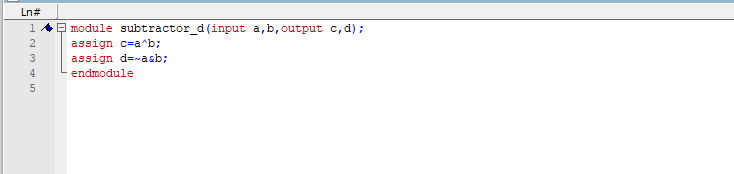
**Explanation:**  
A **Half Subtractor** subtracts one binary digit (B) from another (A).  
It produces two outputs:

* **Subtract = A ⊕ B**
* **Borrow = A'B**

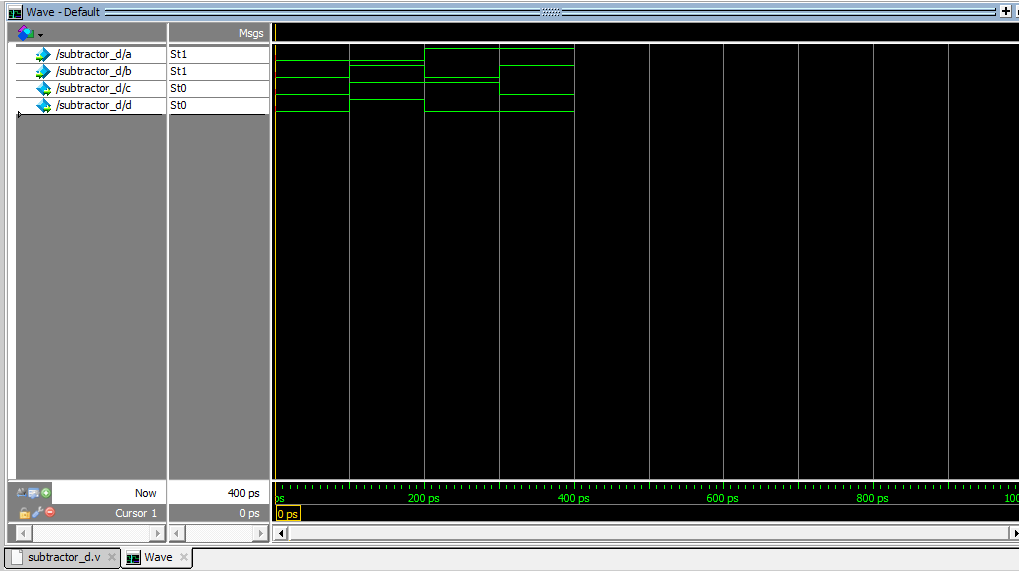
**Truth Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **Subtract (A ⊕ B)** | **Borrow (A'B)** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

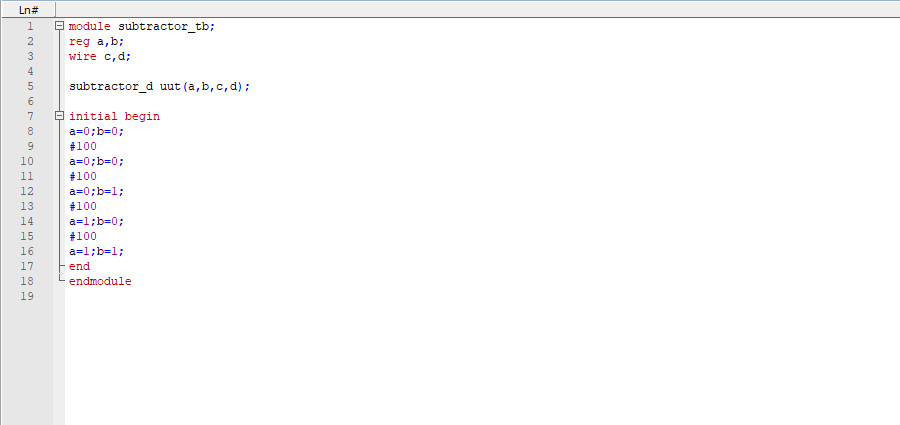
**Code**



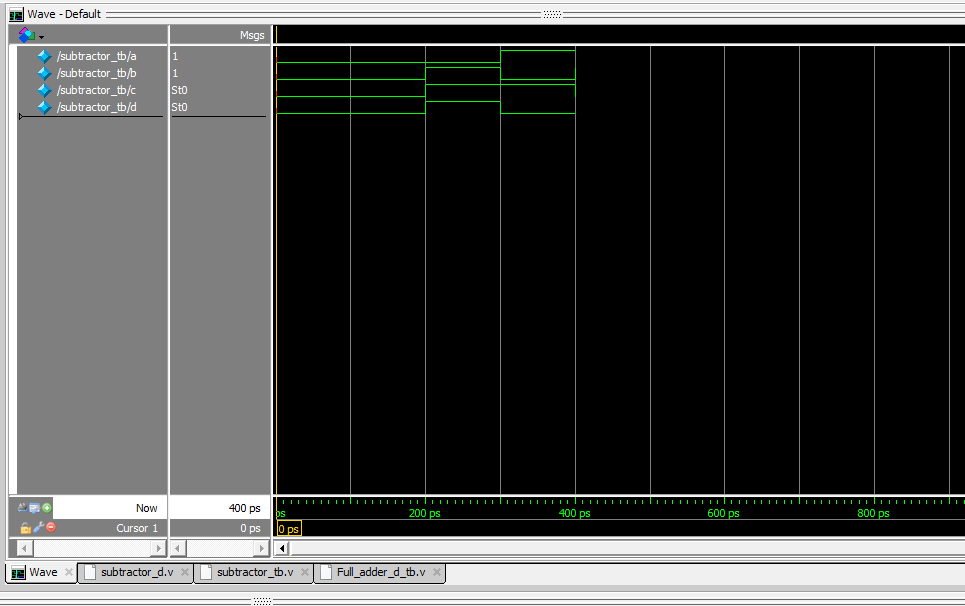
**Output:**



**Half Subtractor With Testbench**:



**Output:**



**Full Subtractor**

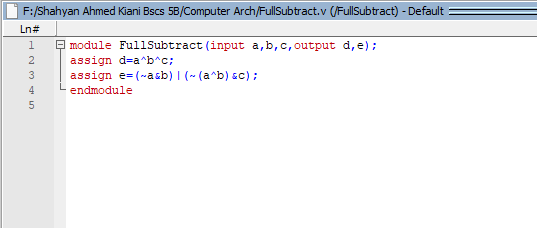
**Explanation:**  
A **Full Subtractor** subtracts two bits and a borrow from a previous stage.  
Outputs:

* **Subtract = A ⊕ B ⊕ C**
* **Borrow = A'B + A'C + BC**

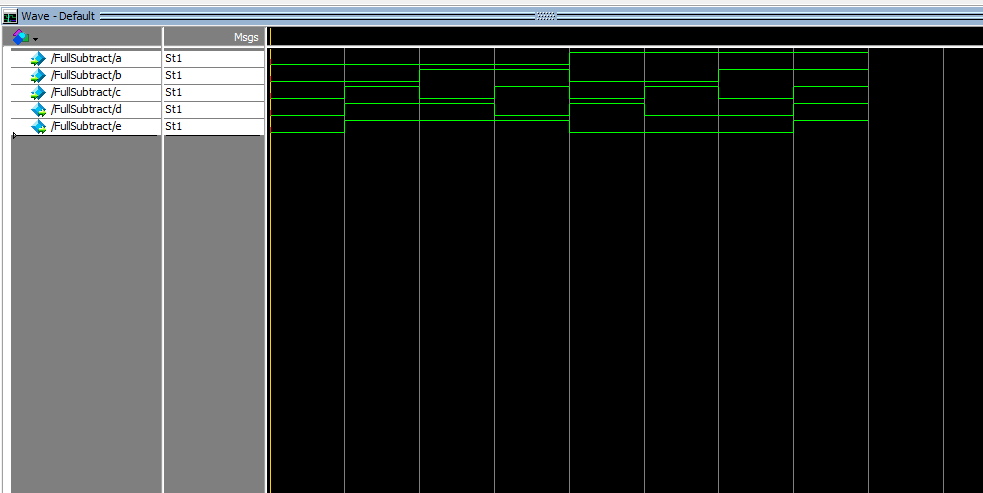
**Truth Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C (Borrow In)** | **Subtract (A ⊕ B ⊕ C)** | **Borrow Out (A'B + A'C + BC)** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

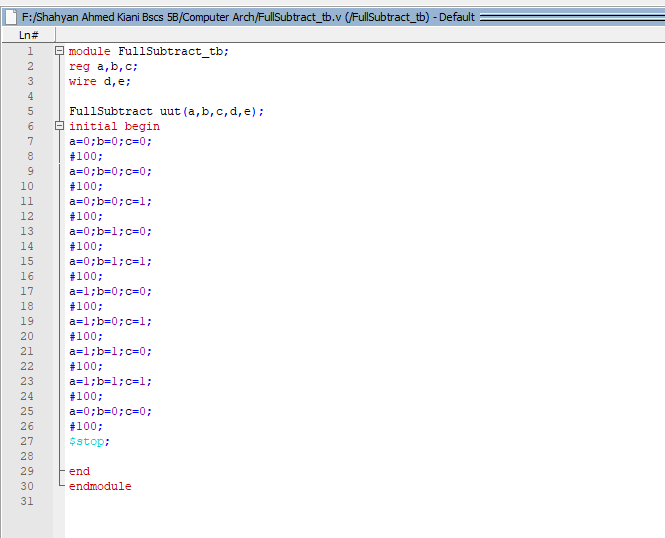
**Code**



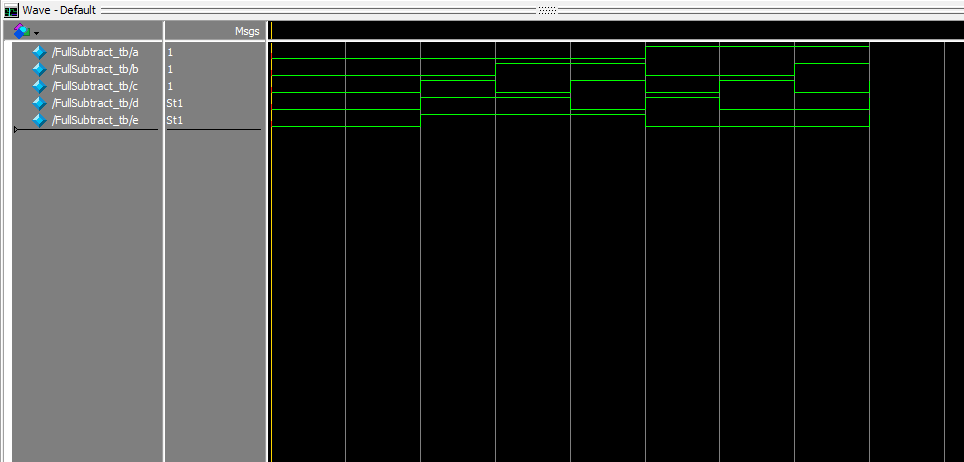
**Output:**



**Full Subtractor With Testbench**



**Output:**



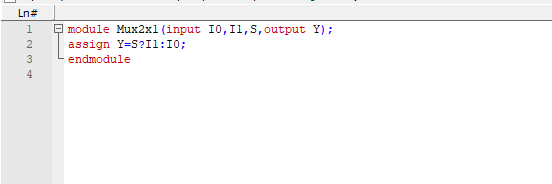
**Multiplexer 2x1**

**Explanation:**  
A **2x1 Multiplexer (MUX)** selects one of two input signals based on a **select line (S)**.  
Output equation:  
**Y = S'I0 + SI1**

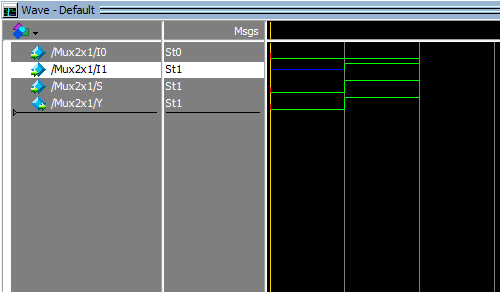
**Truth Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S** | **I0** | **I1** | **Y** |
| 0 | 0 | 0 | I0 |
| 1 | 0 | 1 | I1 |

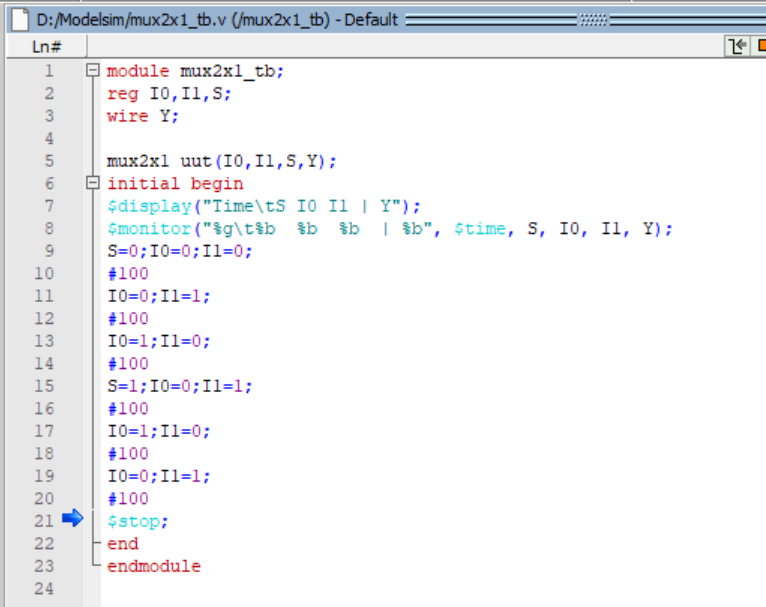
**Code**



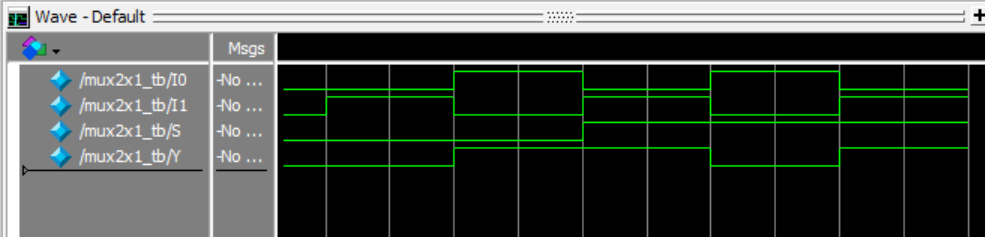
**Output:**



**Multiplexer With Testbench**



**Output:**



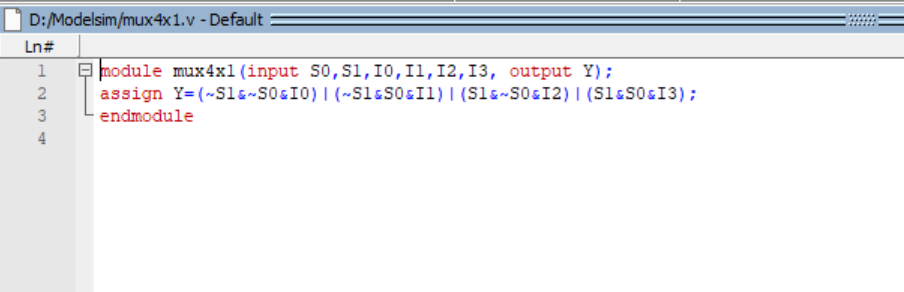
**Multiplexer 4x1**

**Explanation:**  
A **4x1 MUX** has four input lines and two select lines (S1, S0).  
Output equation:  
**Y = S1'S0'I0 + S1'S0I1 + S1S0'I2 + S1S0I3**

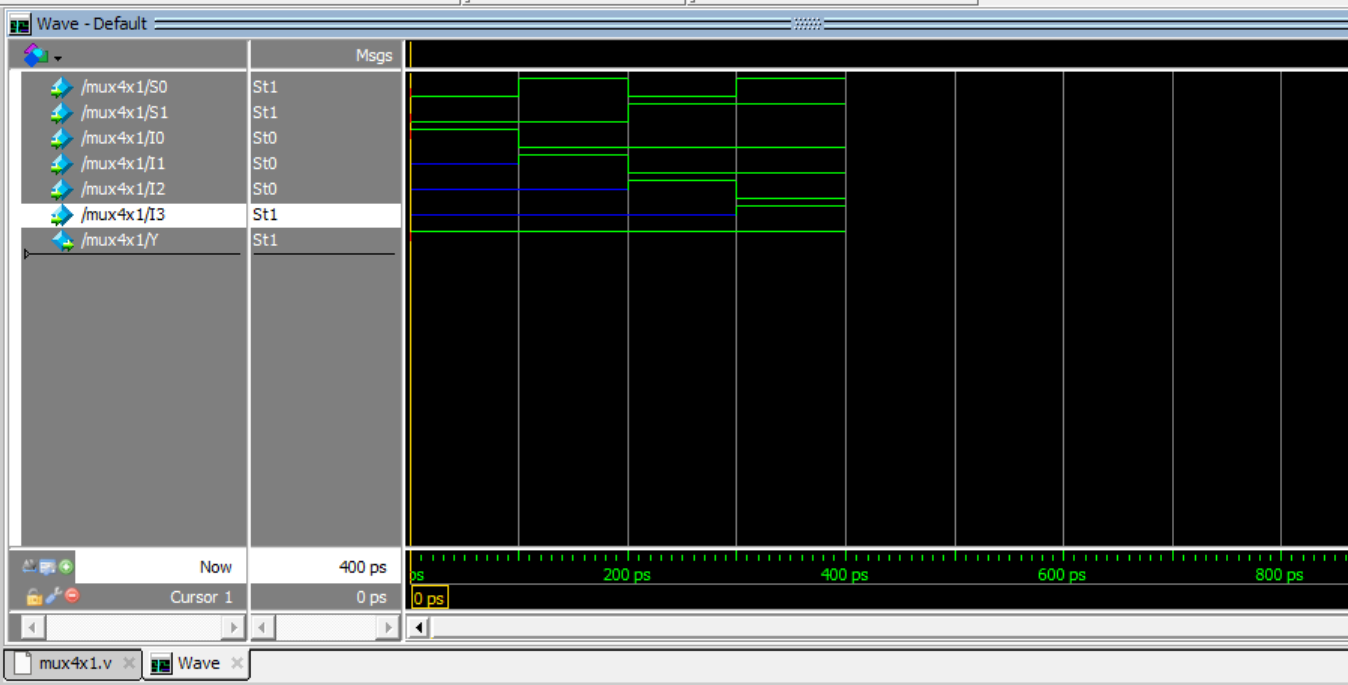
**Truth Table:**

|  |  |  |
| --- | --- | --- |
| **S1** | **S0** | **Y** |
| 0 | 0 | I0 |
| 0 | 1 | I1 |
| 1 | 0 | I2 |
| 1 | 1 | I3 |

**Code**

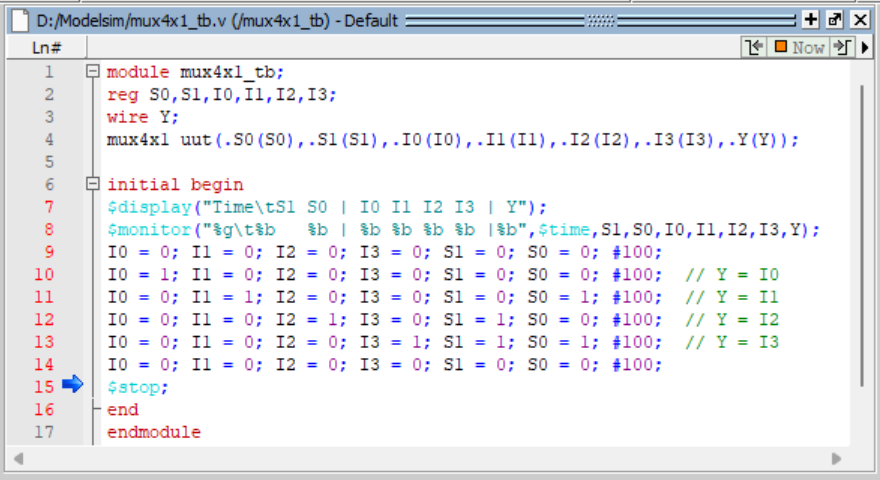


**Output:**

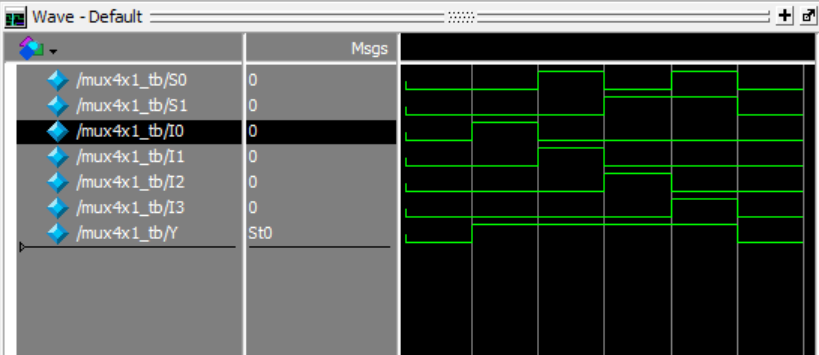


**Multiplexer With Testbench**

**Code:**

****

**Output:**



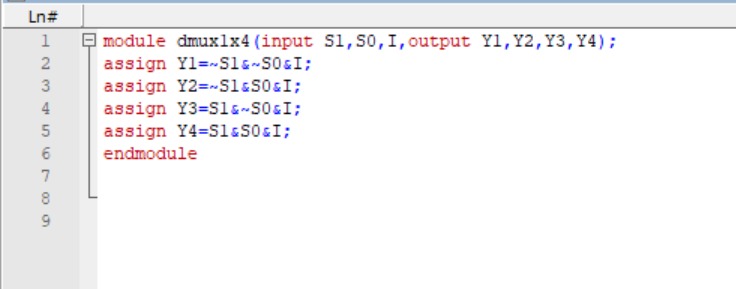
**Demultiplexer**

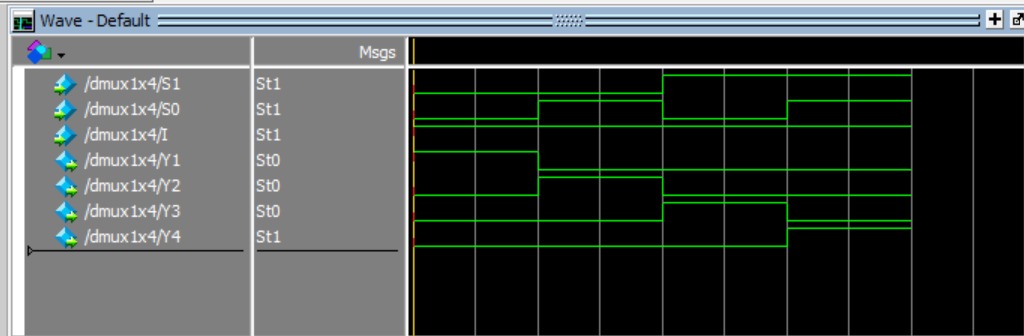
**Explanation:**  
A Demultiplexer (DEMUX) takes a single input and routes it to one of several outputs, depending on select lines.  
Example for 1x4:  
Y0 = S1'S0' D, Y1 = S1'S0 D, Y2 = S1S0' D, Y3 = S1S0 D

**Table:**

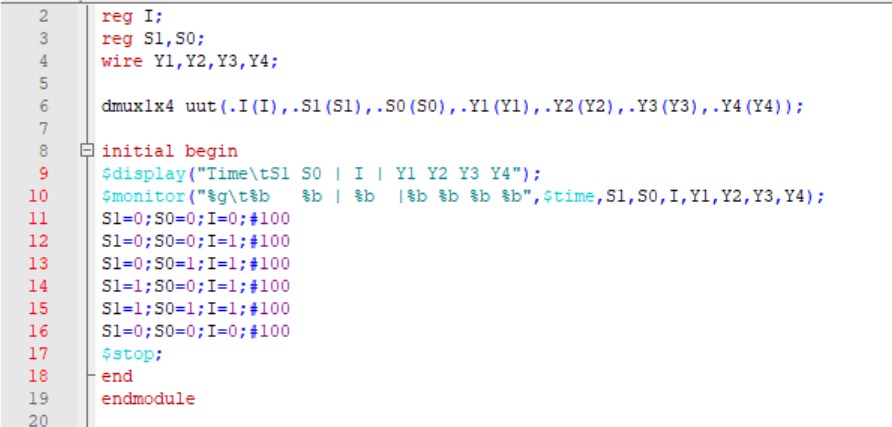
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S1 | S0 | Input | Y1 | Y2 | Y3 | Y4 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

**Code:**

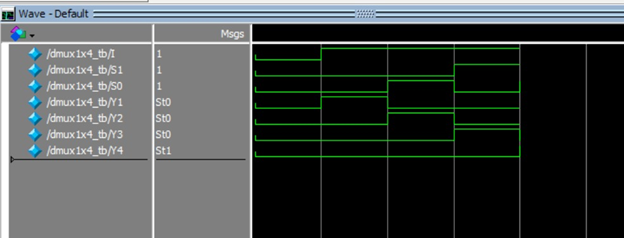


**Output**

**Demultiplexer With Techbench**



**Output**

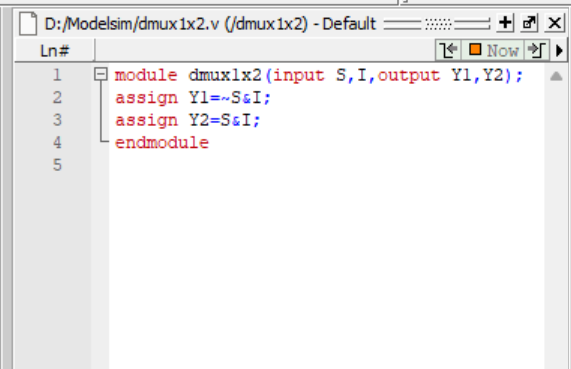


**DeMultiplexer 1x2**

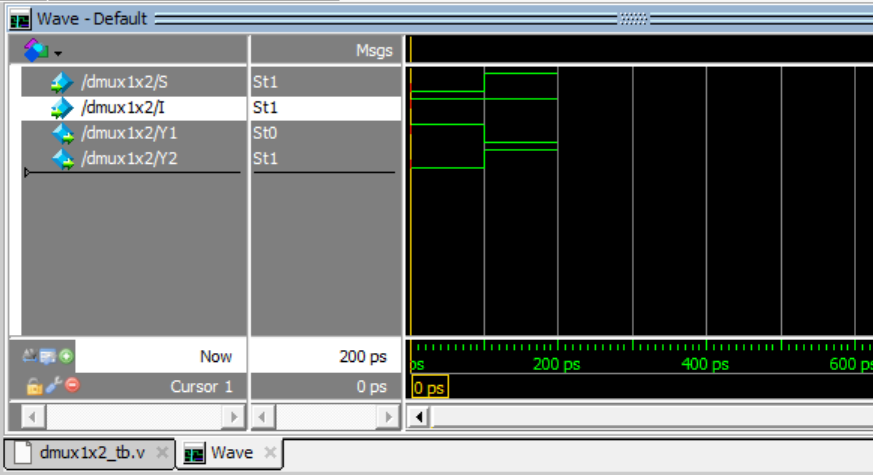
**Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| S | I | Y1 | Y2 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |

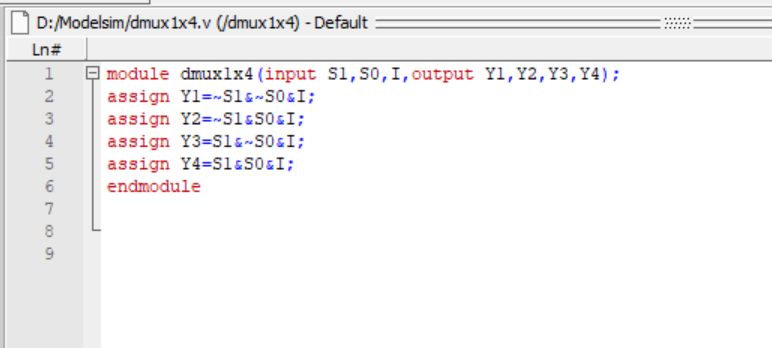
**Code:**

****

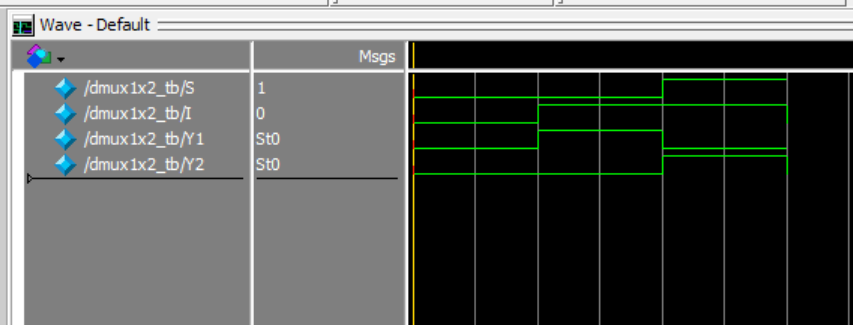
**Output:**

****

**Testbench Code**

****

**Output:**

****