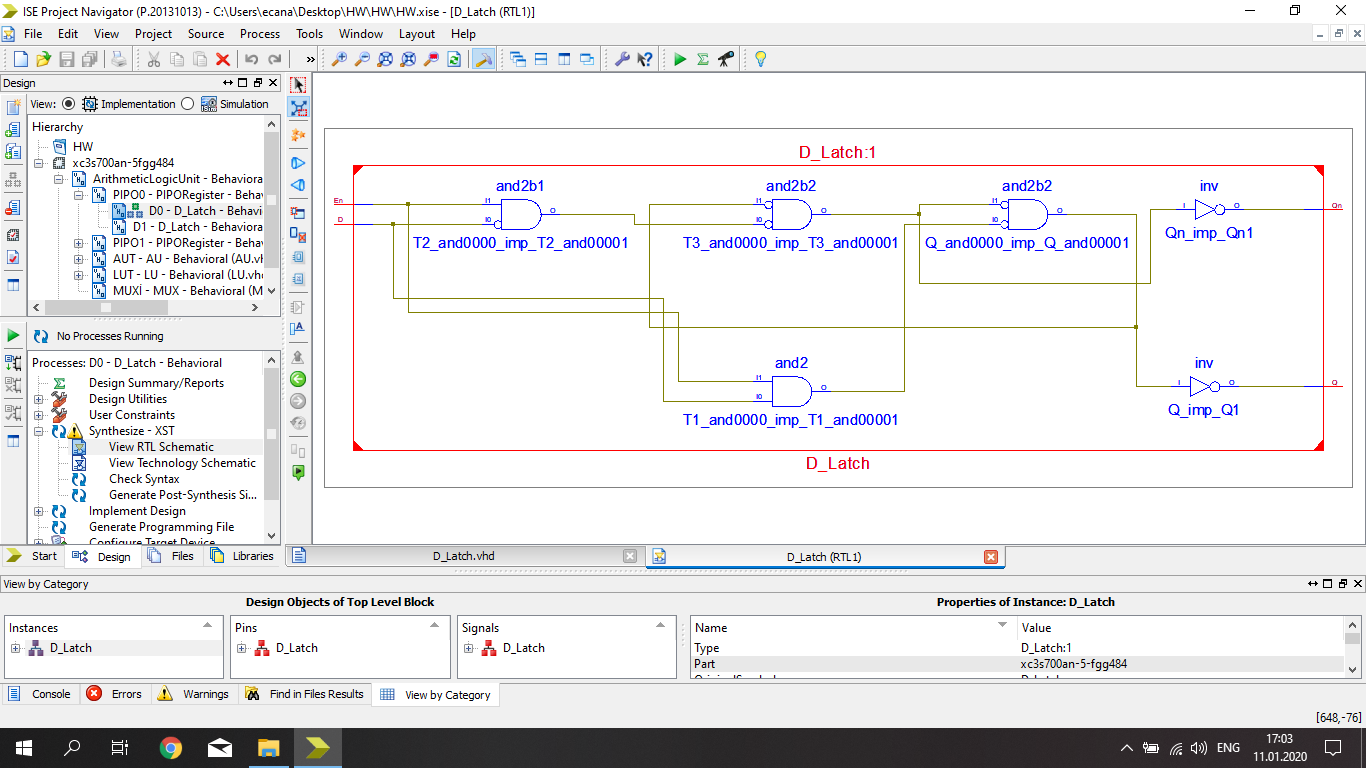
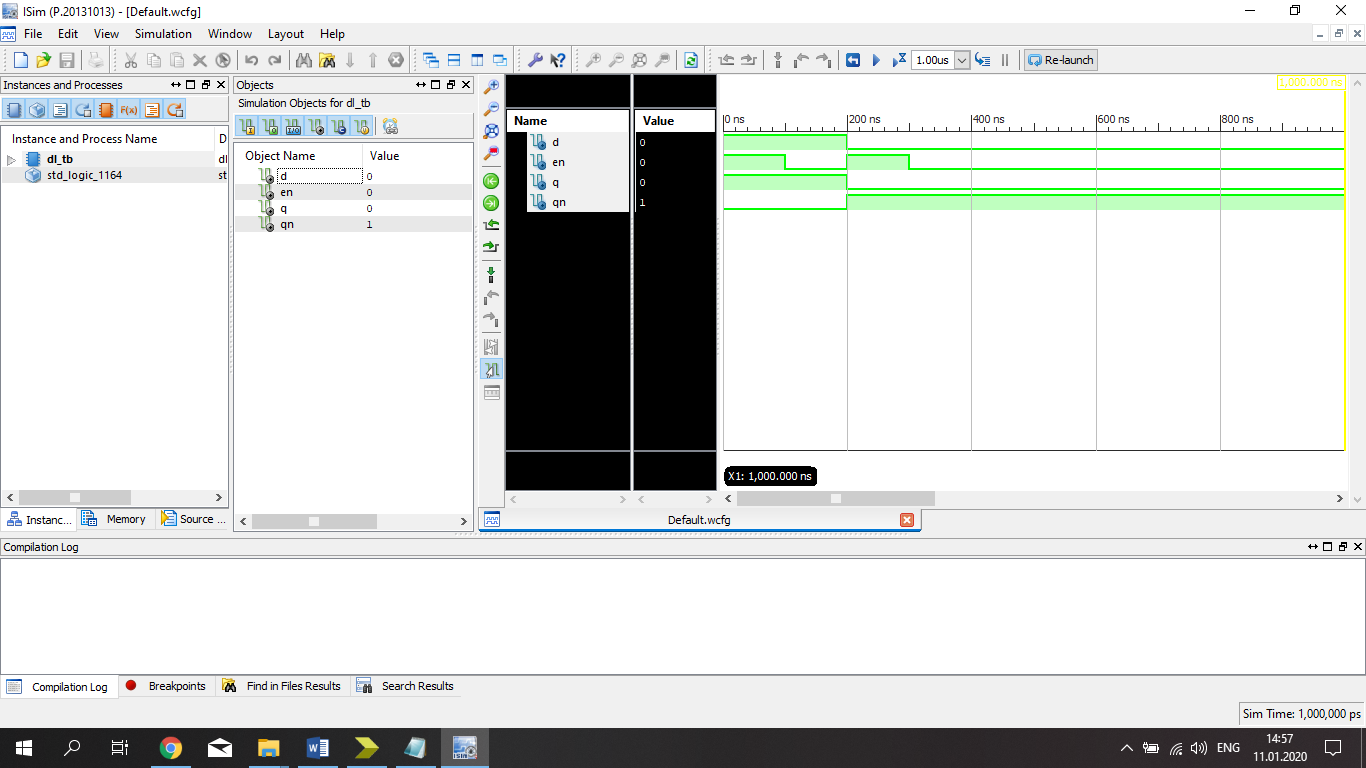
1. **D Lacth**

**RTL Schematic**



**Test Bench**

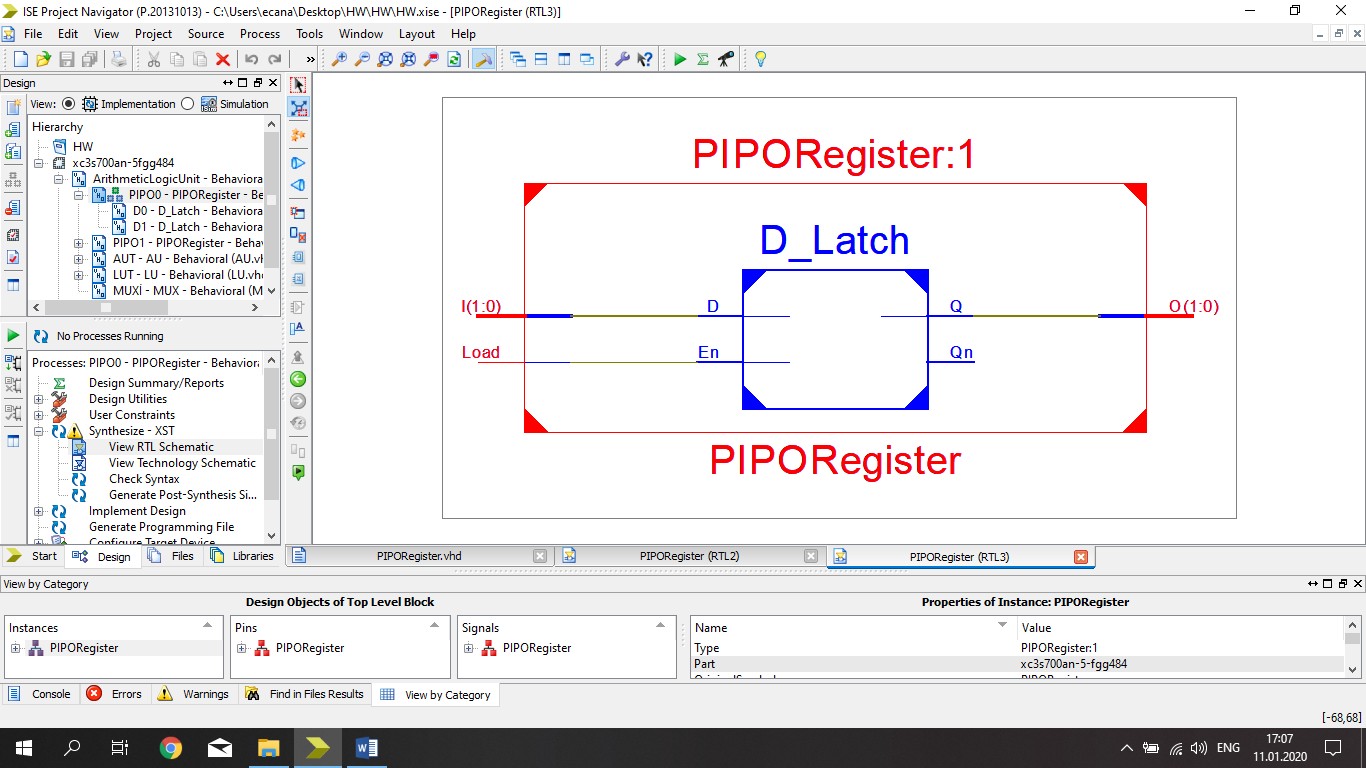


**D latch is given yourself when enable is 1. If enable is 0, Q is acting like memory.**

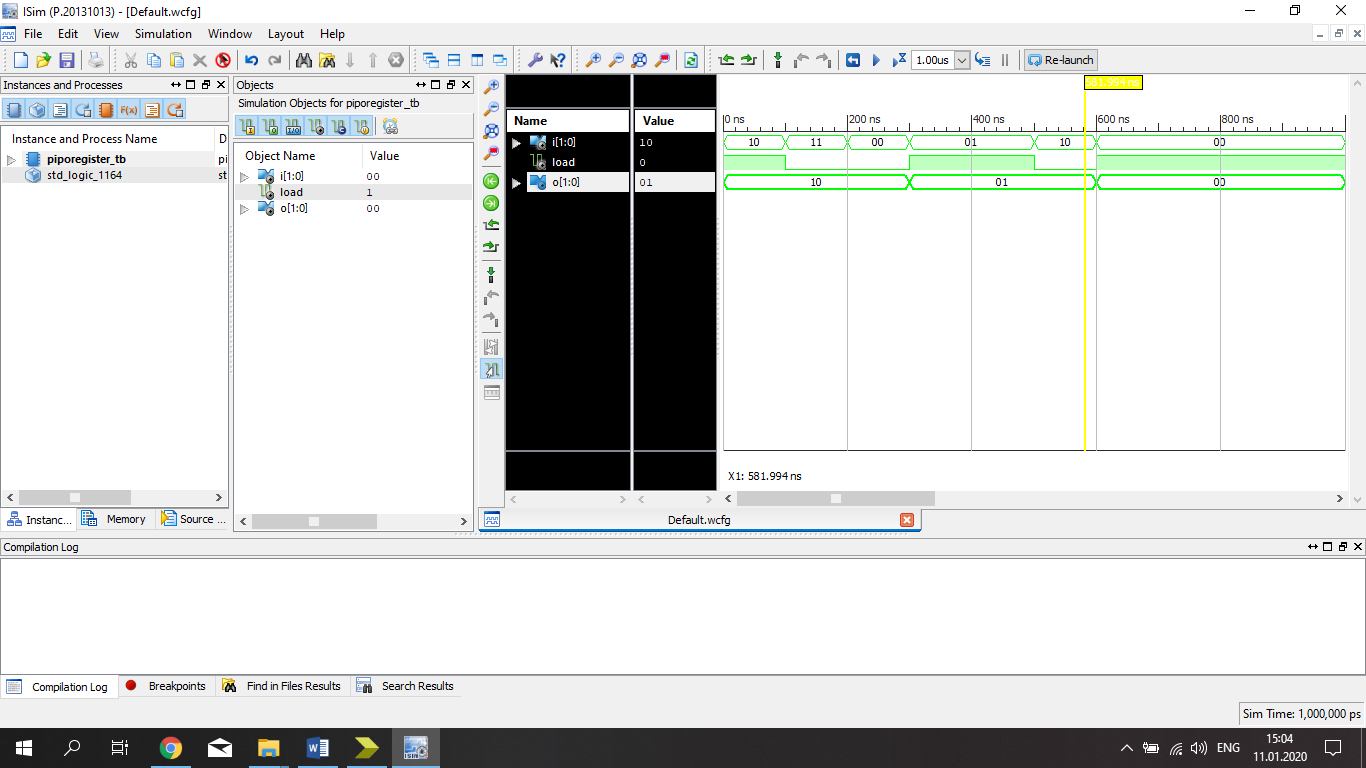
|  |  |  |  |
| --- | --- | --- | --- |
| **D** | **En** | **Q** | **Qn** |
| **1** | **1** | **1** | **0** |
| **1** | **0** | **1** | **0** |
| **0** | **1** | **0** | **1** |
| **0** | **0** | **0** | **1** |

1. **PIPO Register**

**RTL Schematic**



**Test Bench**

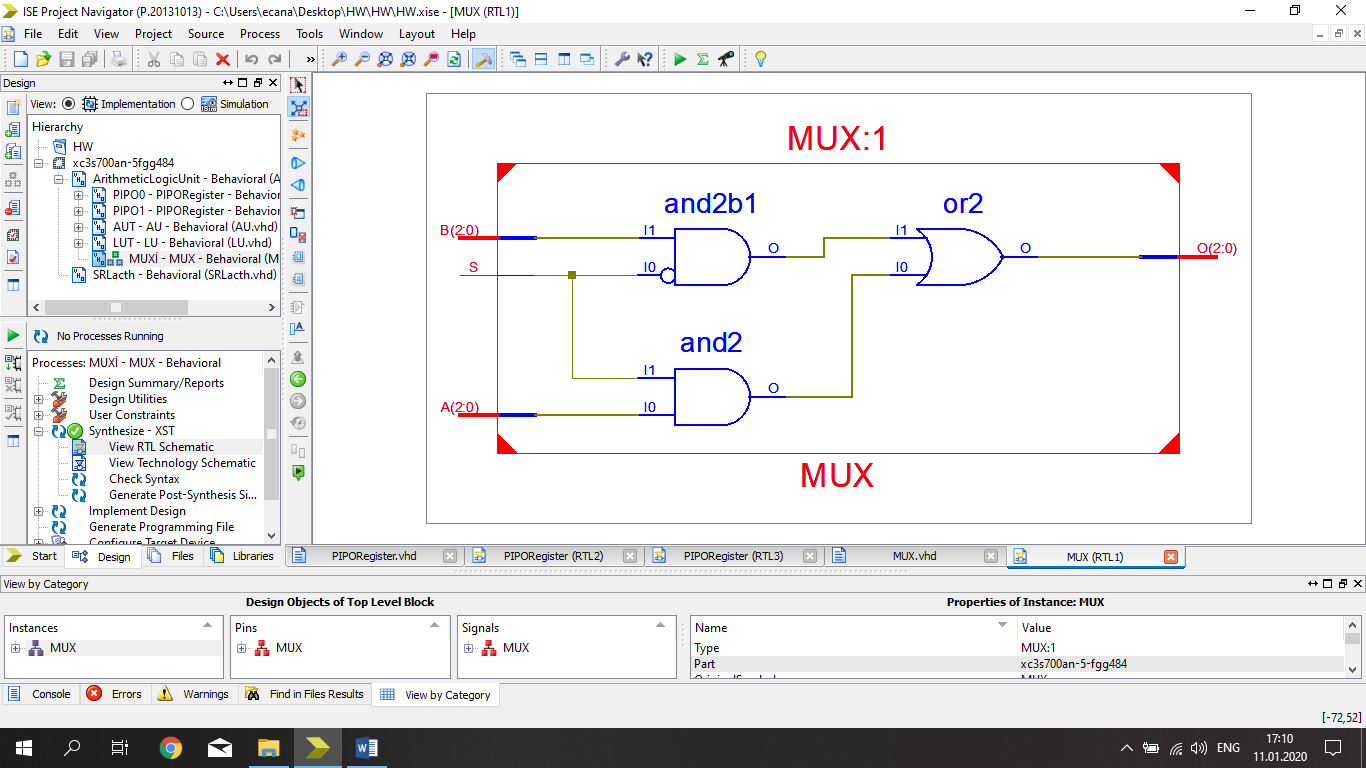


**PIPO Register is acting like D latch , It is given data when load is 1. If load is 0 , output will give the before result.**

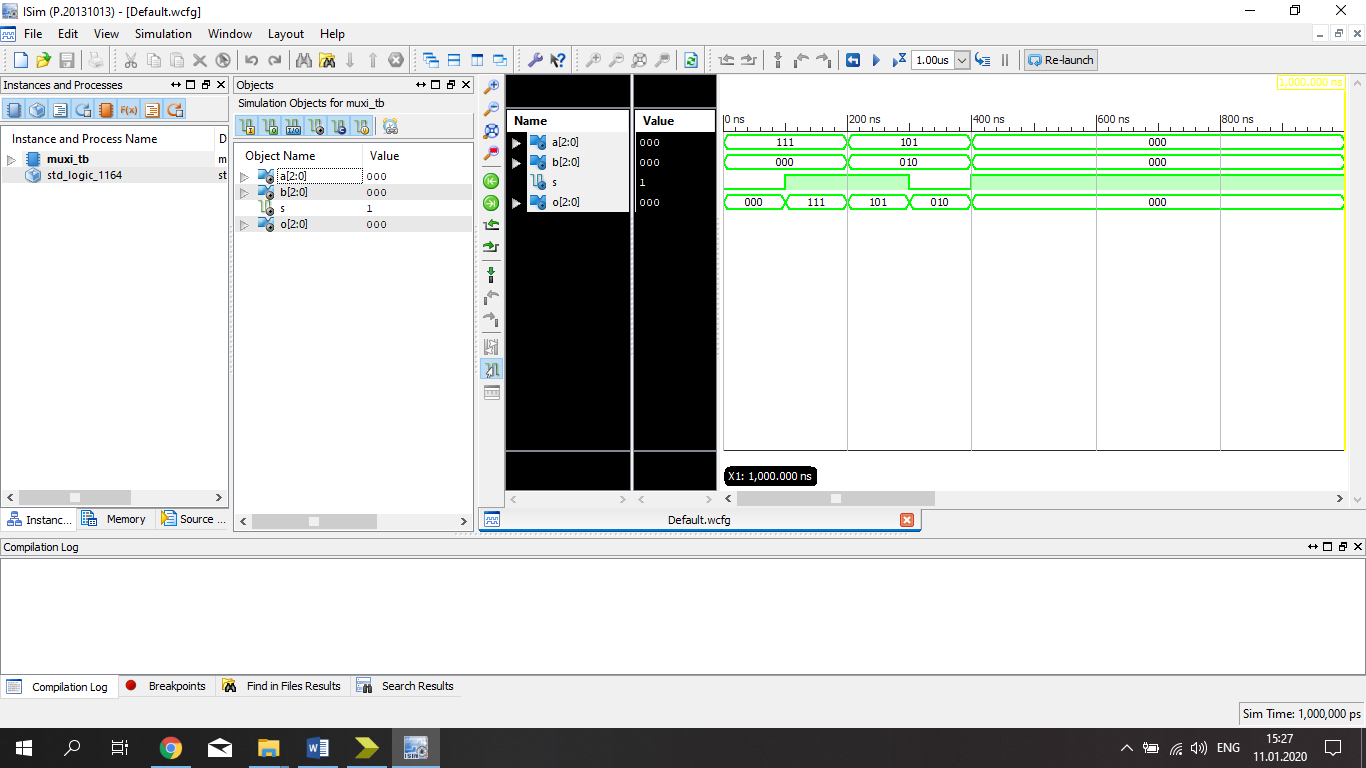
**FOR Example : Our input is 10 and load is 1 , our output will be 10 . After that we change ınput and load will be 0 , our output doesn’t change because when load is 0 , output shows before result.**

1. **MULTİPLEXER**

**RTL Schematic**



**Test Bench**



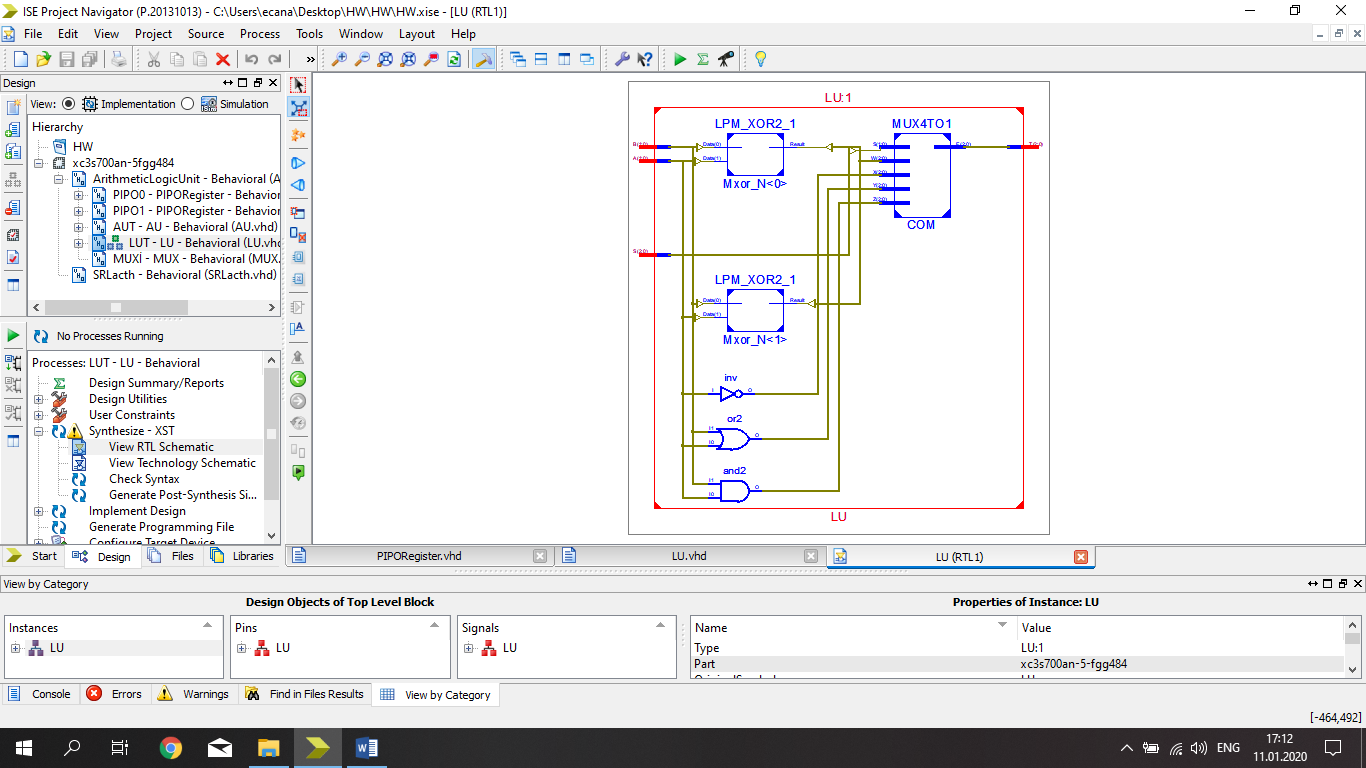
**We use multiplexer to transfer the selected value.**

**For example : We want to see input A . If S will be 1 , we can see it .**

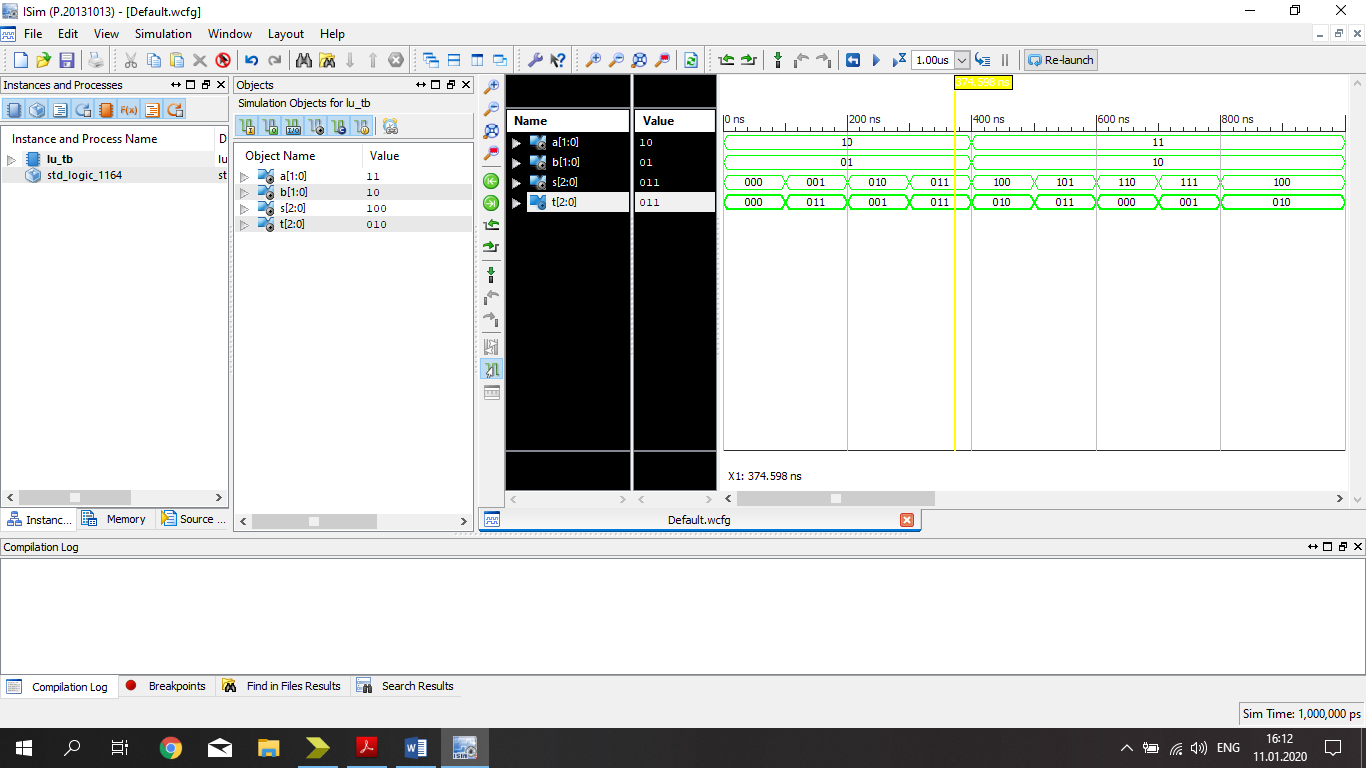
|  |  |  |
| --- | --- | --- |
| **INPUTS** | **SELECT LİNE** | **OUTPUT** |
| **X** | **1** | **A** |
| **X** | **0** | **B** |

1. **Logic Unit**

**RTL Schematic**



**Test Bench**



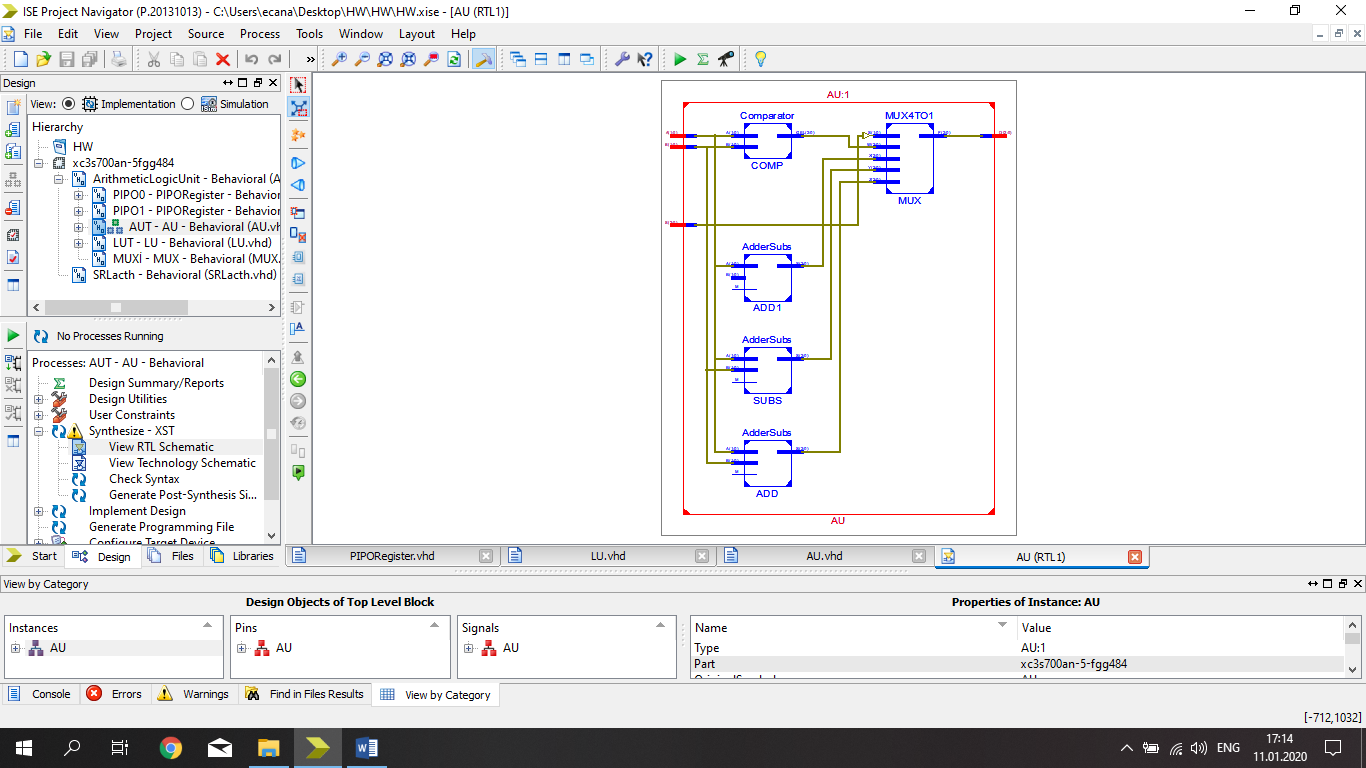
**We use 4 logic operations in this test bench.**

1. **When S = 000 , A and B ( A(1) and B(1) = T(1) , A(0) and B(0) = T(0) )**
2. **When S = 001 , A or B ( A(1) or B(1) = T(1) , A(0) or B(0) = T(0) )**
3. **When S = 010 , Not A**
4. **When S = 011 , A xor B ( A(1) xor B(1) = T(1) , A(0) xor B(0) = T(0) )**

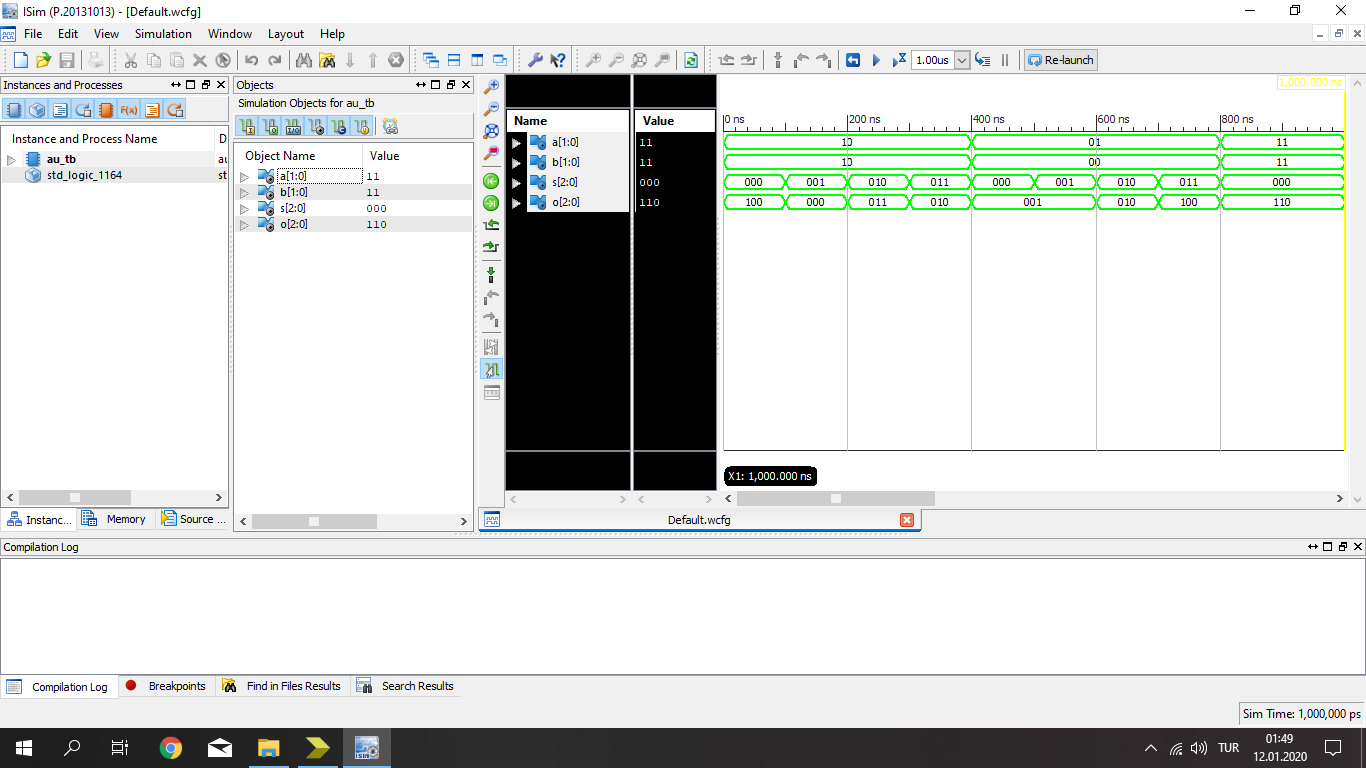
**S(2) doesn’t change anything.**

1. **Arithmetic Unit**

**RTL Schematic**



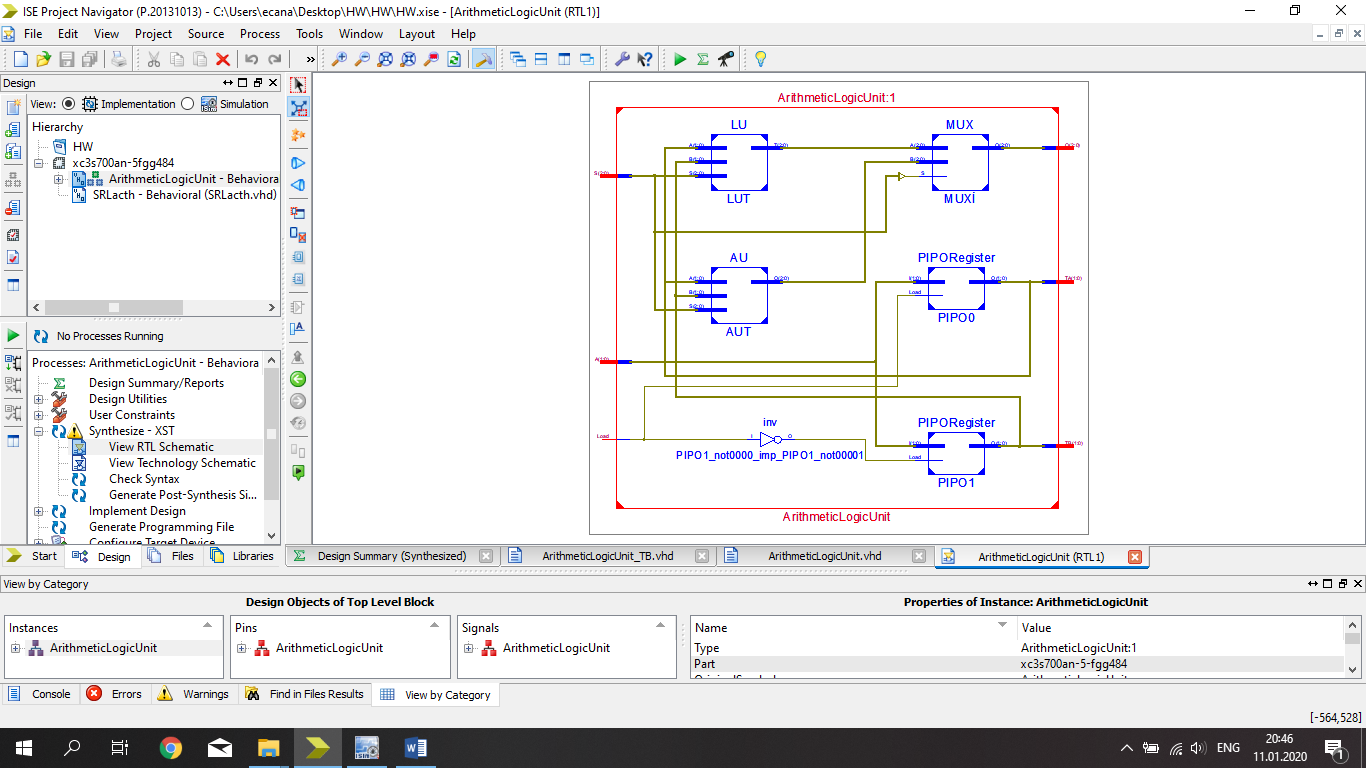
**Test Bench**



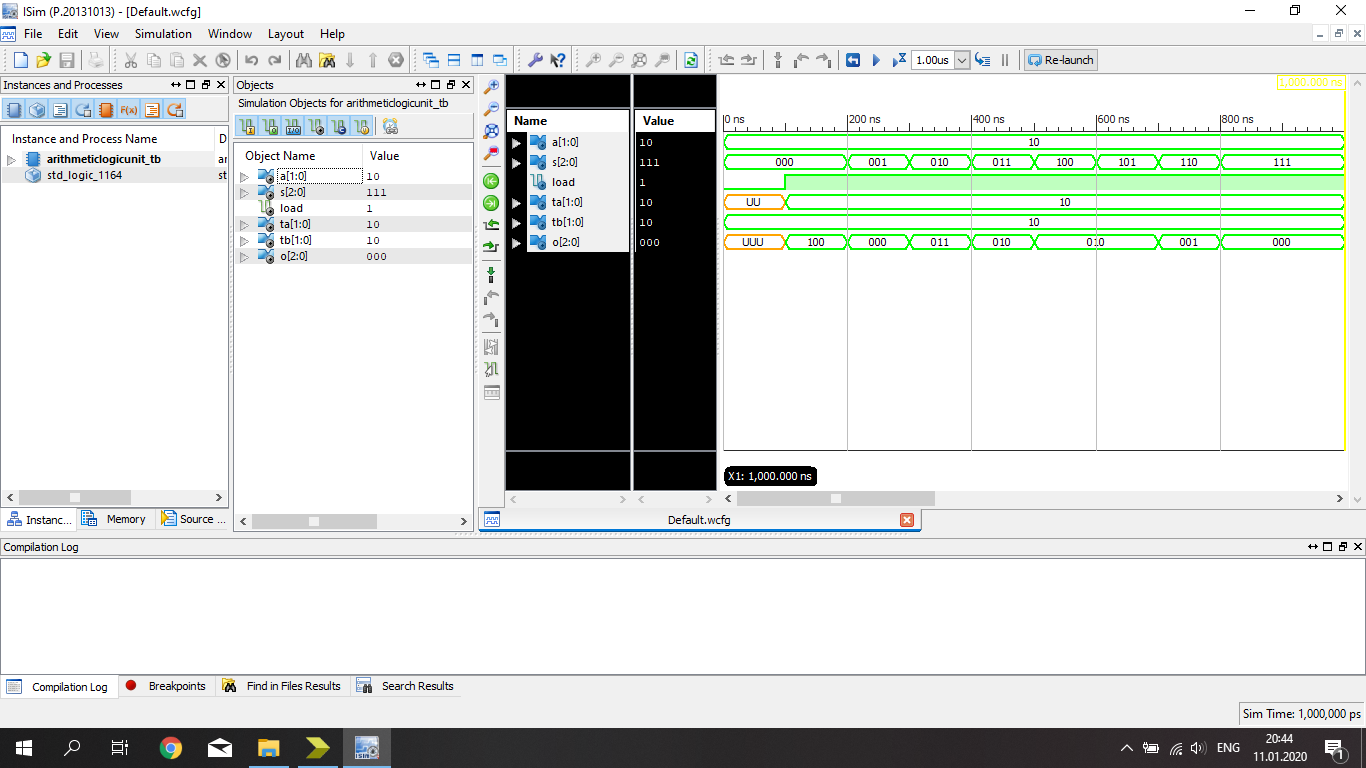
**We use arithmetic operations in this test bench.**

1. **When S = 000 , (A + B)**
2. **When S = 001 , (A - B)**
3. **When S = 010 , (A + 1)**
4. **When S = 011 , (A >=< B) ( In comparator , I designed output like GEL so , ıf A <B , output is given 001 , If A = B , output is given 010 , If A > B , output is given 100.)**
5. **Arithmetic Logic Unit**

**RTL Schematic**



**Test Bench**



**First quadruple is for Arithmetic Operation**

1. **When S = 000 (TA + TB)**
2. **When S = 001 (TA - TB)**
3. **When S = 010 (TA + 1)**
4. **When S = 011 (TA >=< TB)**

**Second quadruple is for Logic Operation**

1. **When S = 100 (TA and TB)**
2. **When S = 101 (TA or TB)**
3. **When S = 110 (Not TA)**
4. **When S = 111 (TA xor TB)**