VLSI - Assignment 1

# Multiplexer

## Instructions

## Write the Verilog code for the following and simulate using the modelsim.

## Design a 2:1 Mux using gate level modeling. Write the test bench for this module as well.

## Design a 4:1 Mux using the 2:1 Mux designed in previous stage

## Design a 8:1 Mux using 4:1 Mux and 2:1 Mux

## Using the conditional assignment statements, write the Verilog code for 16:1 Mux. Write the test bench for this module.

## 1 - 2:1 Mux

### Code

**module** Mut2\_1**(**Out**,** In**,** sel**);**

**output** Out**;**

**input** **[**1**:**0**]** In**;**

**input** sel**;**

**wire** **[**1**:**0**]** W**;**

**and** C1**(**W**[**0**],** In**[**0**],** **(~**sel**));**

**and** C2**(**W**[**1**],** In**[**1**],** sel**);**

**or** C3**(**Out**,** W**[**0**],** W**[**1**]);**

**endmodule**

### Test Code

**module** Mut2\_1\_test**;**

**reg** **[**1**:**0**]** In**;**

**reg** sel**;**

**wire** out**;**

Mut2\_1 M1**(**out**,** In**,** sel**);**

**initial**

**begin**

$monitor**(**$time**,** "out=%b, In=%b, sel=%b\n"**,** out**,** In**,** sel**);**

**#**5 In **=** 2'b00**;** sel **=** 1'b0**;**

**#**5 In **=** 2'b01**;** sel **=** 1'b0**;**

**#**5 In **=** 2'b10**;** sel **=** 1'b0**;**

**#**5 In **=** 2'b11**;** sel **=** 1'b0**;**

**#**5 In **=** 2'b00**;** sel **=** 1'b1**;**

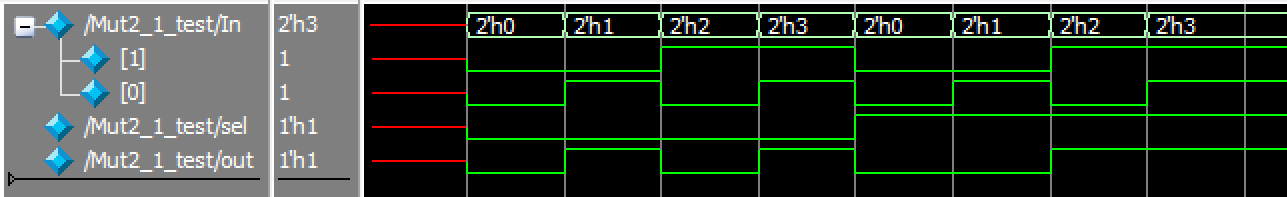
**#**5 In **=** 2'b01**;** sel **=** 1'b1**;**

**#**5 In **=** 2'b10**;** sel **=** 1'b1**;**

**#**5 In **=** 2'b11**;** sel **=** 1'b1**;**

**end**

**endmodule**



*Figure 1: Results Testing 2:1 Mux*

## 2 - 4:1 Mux

### Code

**module** Mut4\_1**(**Out**,** In**,** sel**);**

**output** Out**;**

**input** **[**3**:**0**]** In**;**

**input** **[**1**:**0**]** sel**;**

**wire** **[**1**:**0**]** W**;**

Mut2\_1 M1**(**W**[**0**],** In**[**1**:**0**],** sel**[**0**]);**

Mut2\_1 M2**(**W**[**1**],** In**[**3**:**2**],** sel**[**0**]);**

Mut2\_1 M3**(**Out**,** W**,** sel**[**1**]);**

**Endmodule**

## 3 - 8:1 Mux

### Code

**module** Mut8\_1**(**Out**,** In**,** sel**);**

**output** Out**;**

**input** **[**7**:**0**]** In**;**

**input** **[**2**:**0**]** sel**;**

**wire** **[**1**:**0**]** W**;**

Mut4\_1 M1**(**W**[**0**],** In**[**3**:**0**],** sel**[**1**:**0**]);**

Mut4\_1 M2**(**W**[**1**],** In**[**7**:**4**],** sel**[**1**:**0**]);**

Mut2\_1 M3**(**Out**,** W**,** sel**[**2**]);**

**endmodule**

## 4 - 16:1 Mux

### Code

**module** Mut16\_1**(**Out**,** In**,** Sel**);**

**output** Out**;**

**input** **[**15**:**0**]** In**;**

**input** **[**3**:**0**]** Sel**;**

**assign** Out **=** Sel**[**3**]** **?**

**(**Sel**[**2**]** **?**

**(**Sel**[**1**]** **?**

**(**Sel**[**0**]** **?** In**[**15**]** **:** In**[**14**])** **:**

**(**Sel**[**0**]** **?** In**[**13**]** **:** In**[**12**]))** **:**

**(**Sel**[**1**]** **?**

**(**Sel**[**0**]** **?** In**[**11**]** **:** In**[**10**])** **:**

**(**Sel**[**0**]** **?** In**[**9**]** **:** In**[**8**])))** **:**

**(**Sel**[**2**]** **?**

**(**Sel**[**1**]** **?**

**(**Sel**[**0**]** **?** In**[**7**]** **:** In**[**6**])** **:**

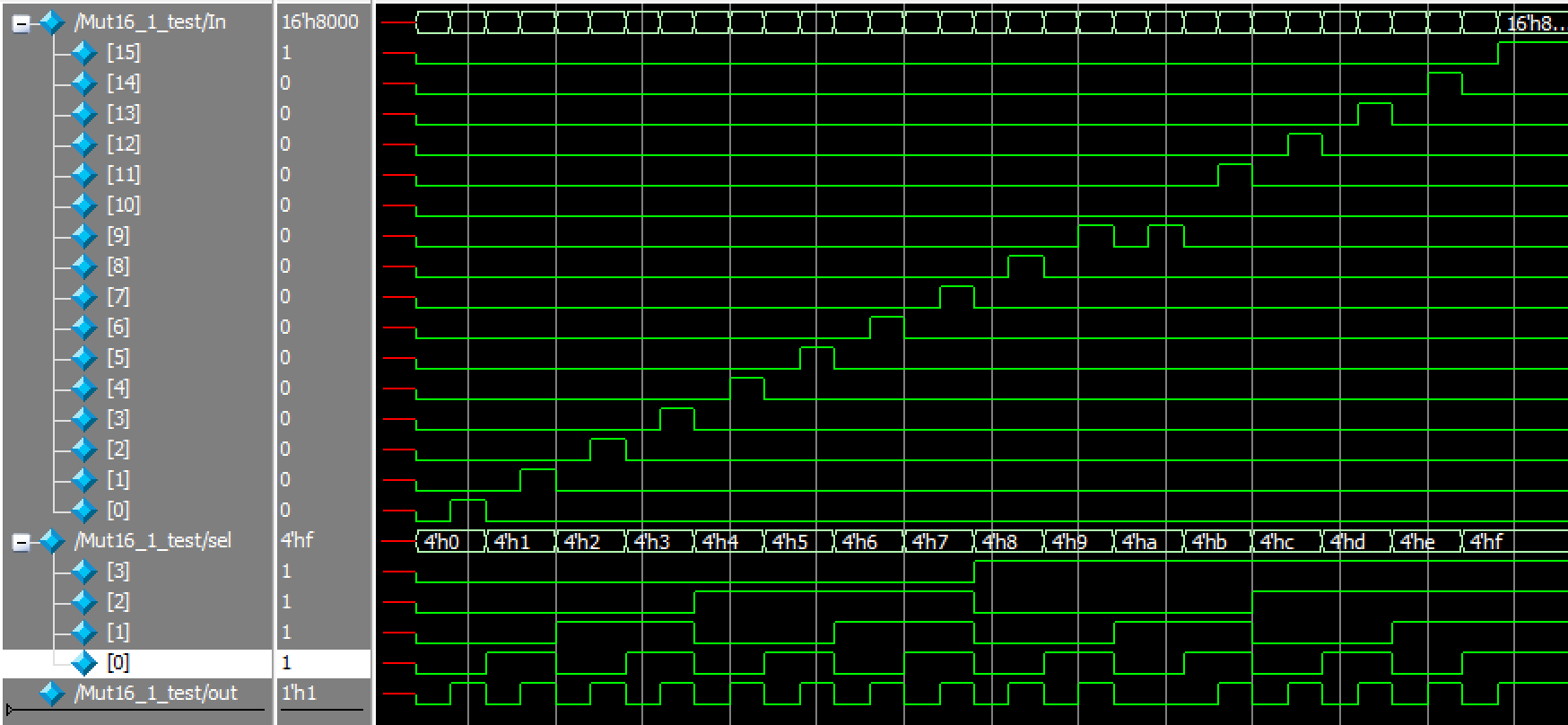
**(**Sel**[**0**]** **?** In**[**5**]** **:** In**[**4**]))** **:**

**(**Sel**[**1**]** **?**

**(**Sel**[**0**]** **?** In**[**3**]** **:** In**[**2**])** **:**

**(**Sel**[**0**]** **?** In**[**1**]** **:** In**[**0**])));**

**Endmodule**



*Figure 2: Results of Testing 16:1 Mux*

### Test Code

**module** Mut16\_1\_test**;**

**reg** **[**15**:**0**]** In**;**

**reg** **[**3**:**0**]** sel**;**

**wire** out**;**

Mut16\_1 M1**(**out**,** In**,** sel**);**

**initial**

**begin**

$monitor**(**$time**,** "out=%b, In=%b, sel=%b\n"**,** out**,** In**,** sel**);**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b0000**;**

**#**2 In **=** 16'b0000000000000001**;** sel **=** 4'b0000**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b0001**;**

**#**2 In **=** 16'b0000000000000010**;** sel **=** 4'b0001**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b0010**;**

**#**2 In **=** 16'b0000000000000100**;** sel **=** 4'b0010**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b0011**;**

**#**2 In **=** 16'b0000000000001000**;** sel **=** 4'b0011**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b0100**;**

**#**2 In **=** 16'b0000000000010000**;** sel **=** 4'b0100**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b0101**;**

**#**2 In **=** 16'b0000000000100000**;** sel **=** 4'b0101**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b0110**;**

**#**2 In **=** 16'b0000000001000000**;** sel **=** 4'b0110**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b0111**;**

**#**2 In **=** 16'b0000000010000000**;** sel **=** 4'b0111**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b1000**;**

**#**2 In **=** 16'b0000000100000000**;** sel **=** 4'b1000**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b1001**;**

**#**2 In **=** 16'b0000001000000000**;** sel **=** 4'b1001**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b1010**;**

**#**2 In **=** 16'b0000001000000000**;** sel **=** 4'b1010**;** // a repeat In, out is low

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b1011**;**

**#**2 In **=** 16'b0000100000000000**;** sel **=** 4'b1011**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b1100**;**

**#**2 In **=** 16'b0001000000000000**;** sel **=** 4'b1100**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b1101**;**

**#**2 In **=** 16'b0010000000000000**;** sel **=** 4'b1101**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b1110**;**

**#**2 In **=** 16'b0100000000000000**;** sel **=** 4'b1110**;**

**#**2 In **=** 16'b0000000000000000**;** sel **=** 4'b1111**;**

**#**2 In **=** 16'b1000000000000000**;** sel **=** 4'b1111**;**

**end**

**endmodule**