Cairo University

Faculty of Engineering
Cairo University

First Year Computer Engineering Department

Advanced Logic Design Project

Spring 2021

Course Code: CMP 1030

Submitted To:

DR. Ihab El-Said Abdel Hadi Talkhan

Team Number:

#10

Names	**Section**	**Bench NO**
Abdelrahman Hamdy Ahmed Mekhaimar	1	36
Khaled Hesham Said Taha	1	21
Yasmine Ashraf Hassan Ghanem	2	36
Yasmin Abdullah Nasser Saeed Mahmoud Elgendi	2	37

Work Load

Name	Task
Abdelrahman Hamdy Ahmed Mekhaimar	Master Master Test Bench
Khaled Hesham Said Taha	Master Master Test Bench
Yasmine Ashraf Hassan Ghanem	Slave Slave Test Bench
Yasmin Abdullah Nasser Saeed Mahmoud Elgendi	Slave Slave Test Bench

Master Design

The Master module includes 10 parameters:

Reset ⟶ makes shift register in the master = 0.

MasterDataToSend ⟶ which is data to be sent to the Slave.

MasterDataReceived ⟶ which is data to be received from Slave.

Clk ⟶ Synchronous clock to operate the master.

Start⟶Indicates the start of data transmission.

Sclk ⟶ clock sent to the slave sent by Master.

CS ⟶ To know if the Slave is selected to operate or not (active-low).

MOSI ⟶ 1-bit output to the Slave.

MISO ⟶ 1- bit input from the Slave to be read

SlaveSelect ⟶ select which Slave to operate.

- ### *Design Process*

  We added a register for shifting data (Named: Register) and a boolean (Named: transmit) to allow data transmission. Assign sclk to clk. Integer counter initialized to zero as well.

  - For Each positive edge of Start or a positive edge of reset

    We check:

    1-if the Start is equal to one and transmit is zero then

    ⟶we assign data we need to send to the slave to the register.

    Initialize data received with don't cares.

    Initialize maxCount and set transmit to 1 (allow transmission)

    2-if the SlaveSelect is equal to zero or one or two then

    ⟶We assign the CS to the first ,second or third slave. Or we assign it to ones if no slave is selected.

    3-if reset is equal to one then

    ⟶We Reset the register to zero.

- For Each positive edge of Clk :
  we check :

  1- if transmit is equal to one then

  →We shift data in Register and output to the Slave in MOSI.
- For Each negative edge of Clk :
  we check :

  1-if Max count reached (All bits transferred) then

  →Disallow transmission and Unselect slave

  2-if transmit is equal to one then

  →We read from MISO.

  →We update Data Received from slave

  →Increment counter

Master Test Bench (Self-Checking)

The Master module includes 10 parameters:

Reset → makes shift register in the master = 0.

MasterDataToSend → which is data to be sent to the Slave.

MasterDataReceived → which is data to be received from Slave.

Clk → Synchronous clock to operate the master.

Start→Indicates the start of data transmission.

Sclk → clock sent to the slave sent by Master.

CS → To know if the Slave is selected to operate or not (active-low).

MOSI → 1-bit output to the Slave.

MISO → 1- bit input from the Slave to be read

SlaveSelect → select which Slave to operate.

Slave → Acts like the slave module

- *Design Process*

We define a localparam called PERIOD (=5).

First, assign the masterDataToSend and slave to any values.

- In the Initial Begin block:

  1- Set clk to zero

  2- Set reset to zero

  3- Set start to 1

  4- Select the first slave as an example (SlaveSelect = 0)

  Then, after a PERIOD delay, we set the start to zero as we already started the transmission.

  Next, we use $display to print the initial value of masterDataReceived and the slave reg.

  After a delay of (PERIOD*17), all bits have been transferred so we compare the final masterDataReceived with the expected value (original slave). If true, display success. Else, display false.

- In the always begin block:

  Keep toggling the clock after aPERIOD delay.

- In the always posedge clk block:

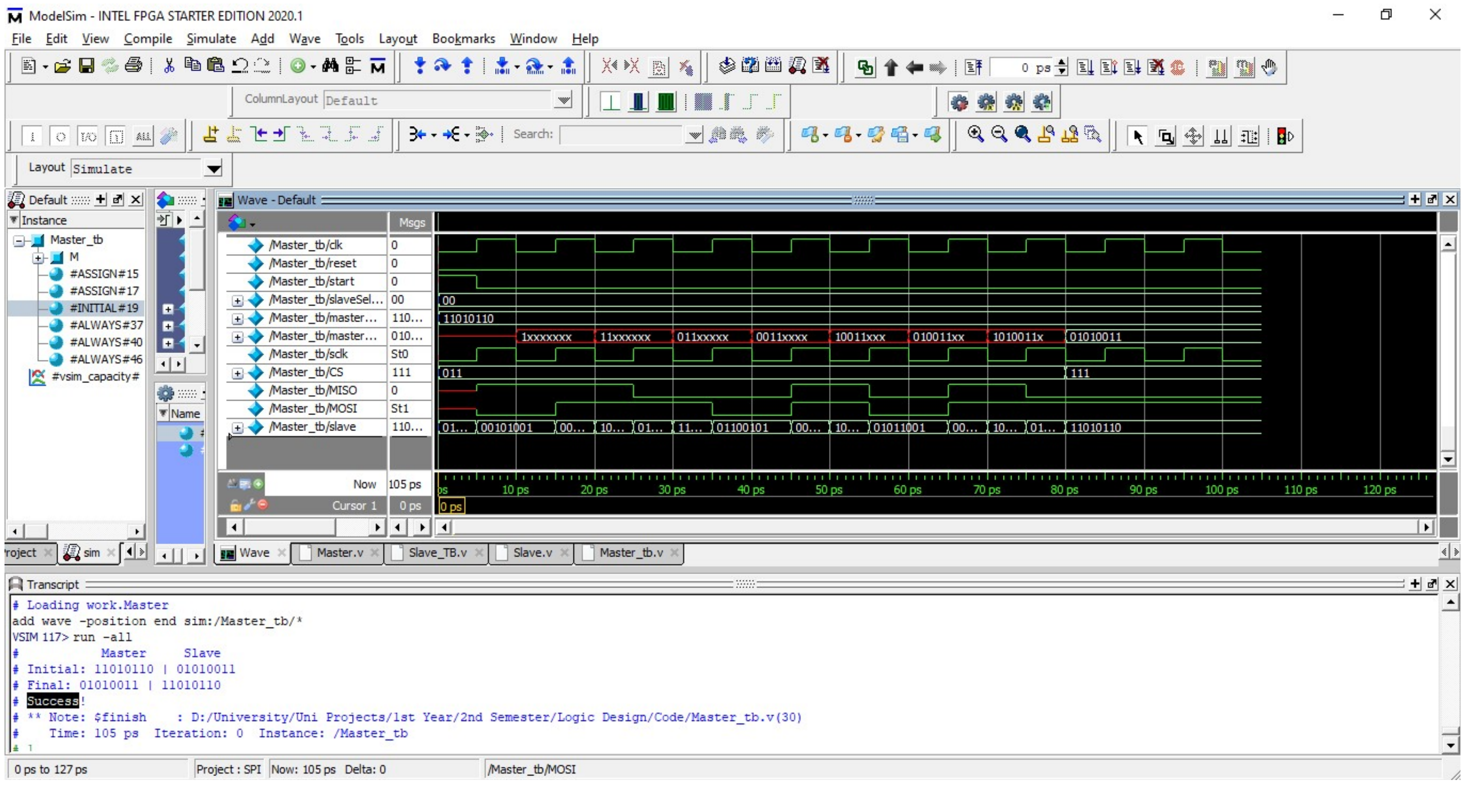
Check if slave is selected. If yes, write to MISO the first bit of the slave reg and shift it by 1 to the right.

- In the always negedge clk block:

  Check if slave is selected. If yes, read from the MOSI into the last bit of the slave reg.

- Connect DUT to testbench:

  Create an instance of the master module and pass all the parameters to it in order.

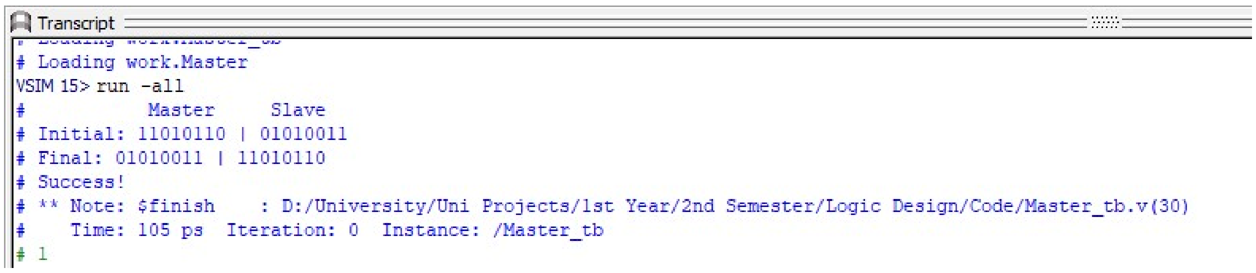

ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Layout Simulate
Wave - Default
Transcript

```
Instance
Master_tb
  M
  #ASSIGN#15
  #ASSIGN#17
  #INITIAL#19
  #ALWAYS#37
  #ALWAYS#40
  #ALWAYS#46
#vsim_capacity#
```

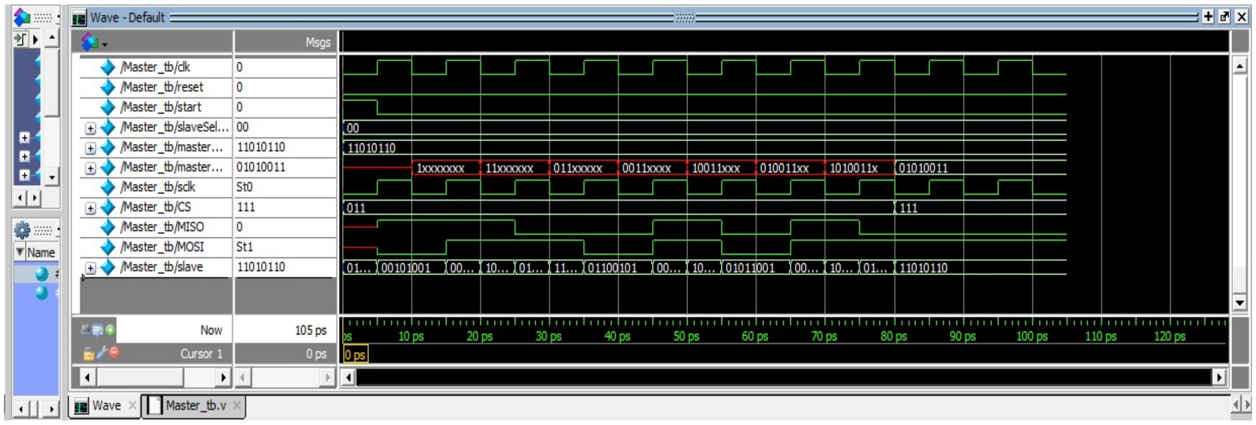
```
/Master_tb/clk          0
/Master_tb/reset        0
/Master_tb/start        0
/Master_tb/slaveSel...  00
/Master_tb/master...    110...
/Master_tb/master...    010...
/Master_tb/sclk         St0
/Master_tb/CS           111
/Master_tb/MISO         0
/Master_tb/MOSI         St1
/Master_tb/slave        110...
Now      105 ps
Cursor 1 0 ps
```

```
# Loading work.Master
add wave -position end sim:/Master_tb/*
VSIM 117> run -all
#            Master     Slave
# Initial: 11010110 | 01010011
# Final: 01010011 | 11010110
# Success!
# ** Note: $finish    : D:/University/Uni Projects/1st Year/2nd Semester/Logic Design/Code/Master_tb.v(30)
#    Time: 105 ps  Iteration: 0  Instance: /Master_tb
# 1
```

```
0 ps to 127 ps    Project : SPI    Now: 105 ps  Delta: 0    /Master_tb/MOSI
```



```
# Loading work.Master
VSIM 15> run -all
#            Master     Slave
# Initial: 11010110 | 01010011
# Final: 01010011 | 11010110
# Success!
# ** Note: $finish    : D:/University/Uni Projects/1st Year/2nd Semester/Logic Design/Code/Master_tb.v(30)
#    Time: 105 ps  Iteration: 0  Instance: /Master_tb
# 1
```



/Master_tb/clk	0
/Master_tb/reset	0
/Master_tb/start	0
/Master_tb/slaveSel...	00
/Master_tb/master...	11010110
/Master_tb/master...	01010011
/Master_tb/sclk	St0
/Master_tb/CS	111
/Master_tb/MISO	0
/Master_tb/MOSI	St1
/Master_tb/slave	11010110
Now	105 ps
Cursor 1	0 ps

Slave Design

The Slave module includes 7 parameters:

Reset ⟶ makes shift register in the slave = 0.

slaveDataToSend ⟶ which is data to be sent to the Master.

slaveDataReceived ⟶ which is data to be received from Master.

Sclk ⟶ clock of the slave sent by Master.

CS ⟶ To know if the Slave is selected to operate or not (active-low).

MOSI ⟶ 1-bit input from Master to be read.

MISO ⟶ 1- bit output to the Master to be read.

- ### *Design Process*

  We added a register for shifting data (Named: Register) and a boolean (Named: flag) to allow data transmission.

  - For Each negative edge of CS or a positive edge of reset

    We check:

    1- if CS equal zero then

    we assign data we need to send to the master to the register.

    Initialize data received with don't cares.

    2-If reset equal zero then

    Register data is reseted to zero.

  - For Each positive edge of CS:

    1- Disconnect slave by not allowing transmission (flag = 0).

  - For Each positive edge of sclk:

    1- We shift data in Register and output to the master in (MISO).

  - For Each negative edge of sclk:

    We check if we are transmitting data (flag = 1):

    then we sample data (Read from MOSI and add it to data received.)

Slave Test Bench (Self-Checking)

The Slave module includes 8 parameters:

Reset ⟶ makes shift register in the slave = 0.

slaveDataToSend ⟶ which is data to be sent to the Slave.

slaveDataReceived ⟶ which is data to be received from Slave.

Sclk ⟶ clock sent to the slave sent by Master.

CS ⟶ To know if the Slave is selected to operate or not (active-low).

MOSI ⟶ 1-bit input to the Slave.

MISO ⟶ 1- bit output from the Master to be read.

master ⟶ Acts like the Master module

Integer i ⟶ Counter that stops transmission when it reaches 8

- ***Design Process***

  We define a localparam called PERIOD (=5).

  First, assign the masterDataToSend and slave to any values.

  - In the Initial Begin block:

    1- Set sclk to zero.

    2- Set reset to zero.

    3- Set CS to one.

    Then, after a PERIOD delay, we set the CS to zero. (for negedge)

    Next, we use $display to print the initial value of master reg and the slaveDataToSend.

    We check if the counter reached 8. If yes, then set CS to 1 and stop transmission.

    After a delay of (PERIOD*17), all bits have been transferred so we compare the final slaveDataReceived with the expected value (original master). If true, display success. Else, display false.

  - In the always begin block:

    Keep toggling the clock after a PERIOD delay.

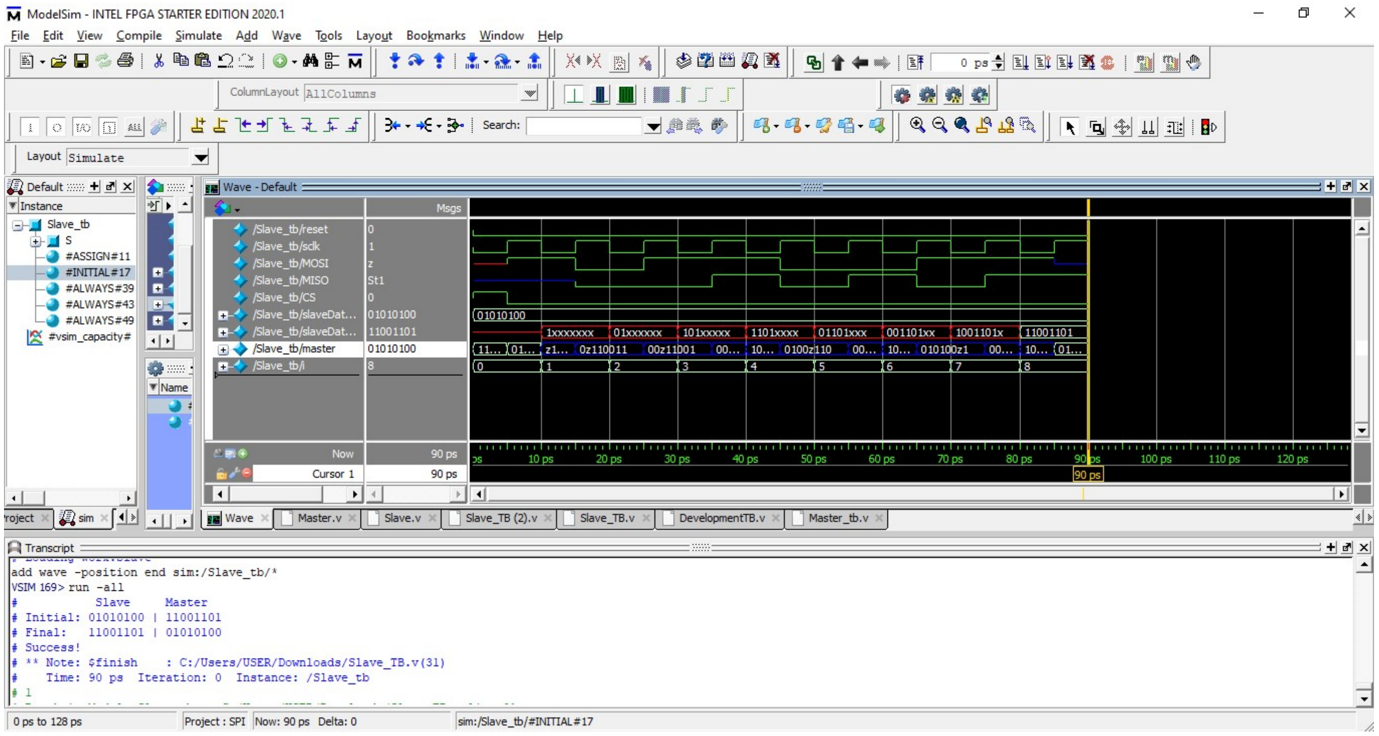
  - In the always posedge sclk block:

    Check if slave is selected. If yes, write to MOSI the first bit of the master reg and shift it by 1 to the right.

- In the always negedge sclk block:

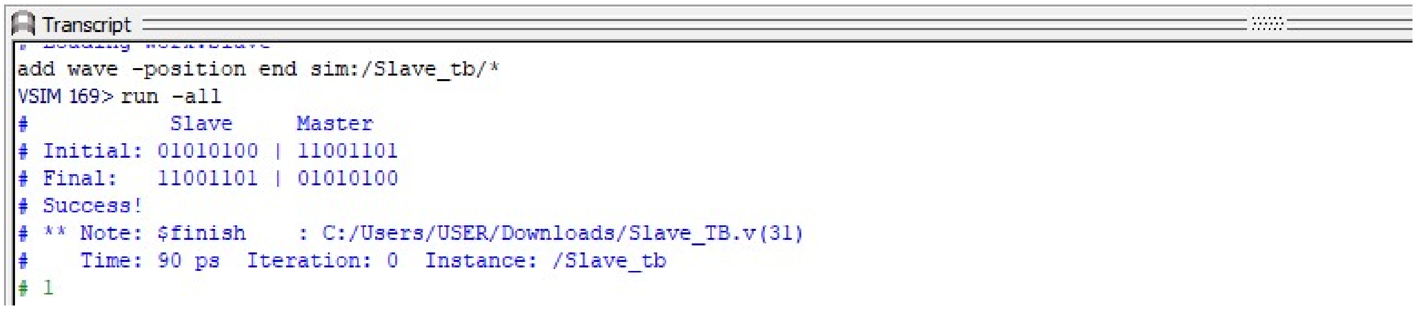
  Check if slave is selected. If yes, read from the MISO into the last bit of the master reg.
- Connect DUT to testbench:

  Create an instance of the master module and pass all the parameters to it in order.

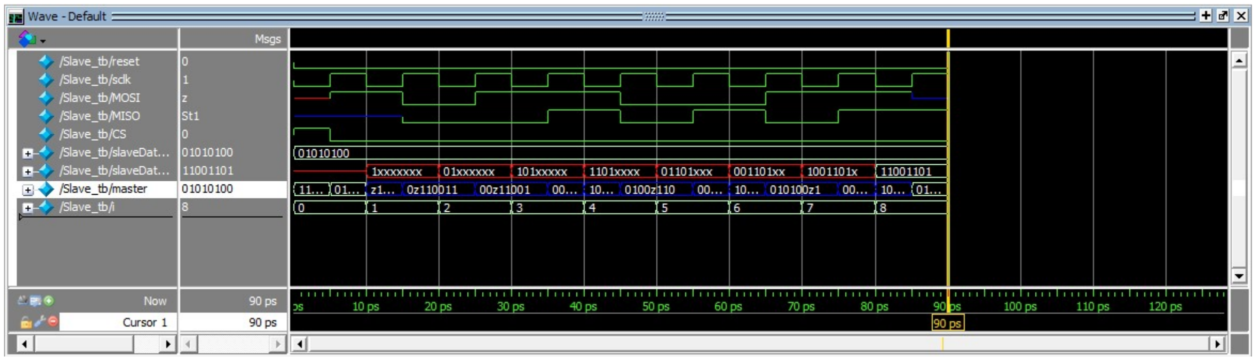


```
/Slave_tb/reset       0
/Slave_tb/sclk        1
/Slave_tb/MOSI        z
/Slave_tb/MISO        St1
/Slave_tb/CS          0
/Slave_tb/slaveDat... 01010100
/Slave_tb/slaveDat... 11001101
/Slave_tb/master      01010100
/Slave_tb/i           8
```

```
add wave -position end sim:/Slave_tb/*
VSIM 169> run -all
#           Slave     Master
# Initial: 01010100 | 11001101
# Final:   11001101 | 01010100
# Success!
# ** Note: $finish    : C:/Users/USER/Downloads/Slave_TB.v(31)
#    Time: 90 ps  Iteration: 0  Instance: /Slave_tb
# 1
```

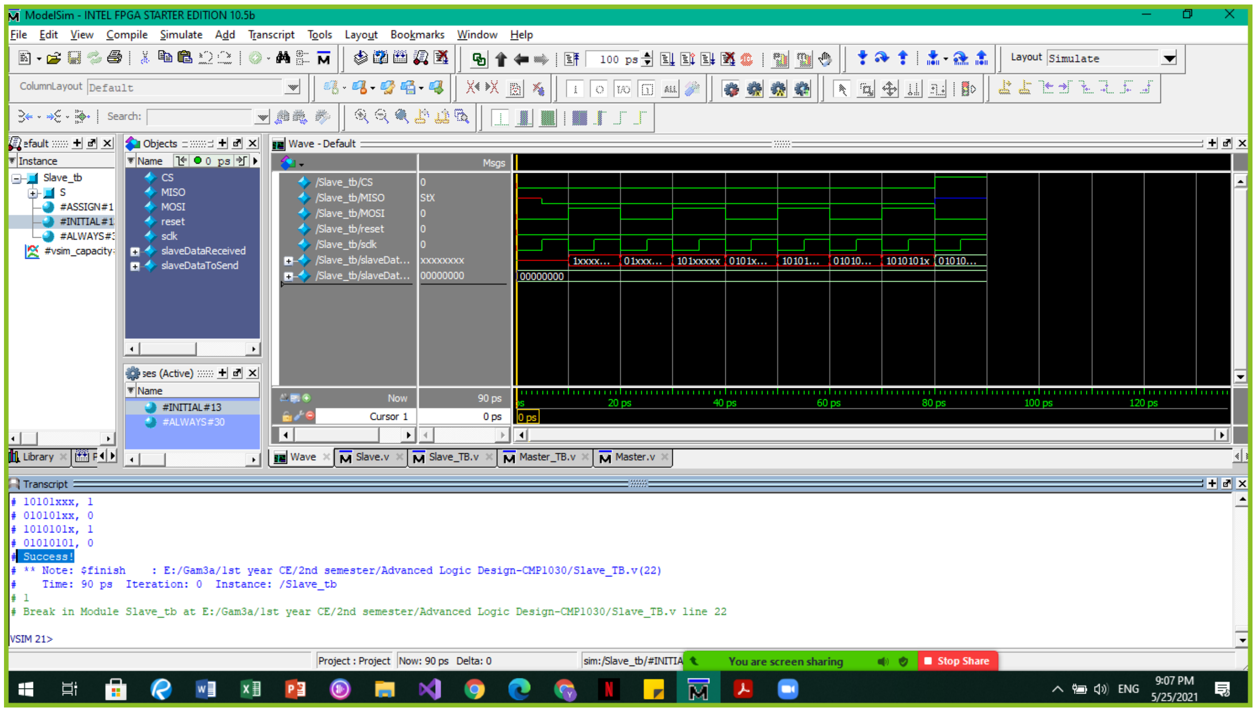


```
add wave -position end sim:/Slave_tb/*
VSIM 169> run -all
#           Slave     Master
# Initial: 01010100 | 11001101
# Final:   11001101 | 01010100
# Success!
# ** Note: $finish    : C:/Users/USER/Downloads/Slave_TB.v(31)
#    Time: 90 ps  Iteration: 0  Instance: /Slave_tb
# 1
```



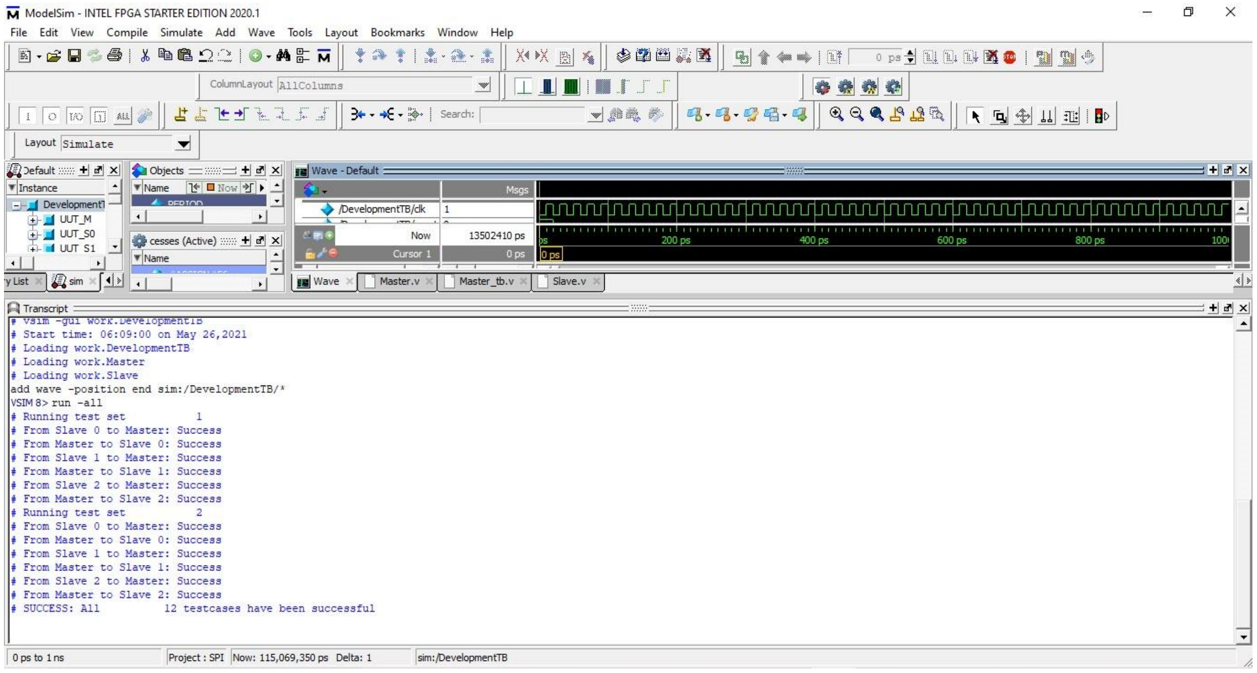
	Msgs
/Slave_tb/reset	0
/Slave_tb/sclk	1
/Slave_tb/MOSI	z
/Slave_tb/MISO	St1
/Slave_tb/CS	0
/Slave_tb/slaveDat…	01010100
/Slave_tb/slaveDat…	11001101
/Slave_tb/master	01010100
/Slave_tb/i	8
Now	90 ps
Cursor 1	90 ps

Development Test Bench (Self-Checking)



```
# 10101xxx, 1
# 010101xx, 0
# 1010101x, 1
# 01010101, 0
# Success!
# ** Note: $finish    : E:/Gam3a/1st year CE/2nd semester/Advanced Logic Design-CMP1030/Slave_TB.v(22)
#    Time: 90 ps  Iteration: 0  Instance: /Slave_tb
# 1
# Break in Module Slave_tb at E:/Gam3a/1st year CE/2nd semester/Advanced Logic Design-CMP1030/Slave_TB.v line 22

VSIM 21>
```



```
# vsim -gui work.DevelopmentTB
# Start time: 06:09:00 on May 26,2021
# Loading work.DevelopmentTB
# Loading work.Master
# Loading work.Slave
add wave -position end sim:/DevelopmentTB/*
VSIM 8> run -all
# Running test set           1
# From Slave 0 to Master: Success
# From Master to Slave 0: Success
# From Slave 1 to Master: Success
# From Master to Slave 1: Success
# From Slave 2 to Master: Success
# From Master to Slave 2: Success
# Running test set           2
# From Slave 0 to Master: Success
# From Master to Slave 0: Success
# From Slave 1 to Master: Success
# From Master to Slave 1: Success
# From Slave 2 to Master: Success
# From Master to Slave 2: Success
# SUCCESS: All          12 testcases have been successful
```