ADVANCED LOGIC DESIGN

PROJECT REPORT

Team-1

MASTER:

Design Process:

1-Choose appropriate Port Connection and Type of Inputs and Outputs.

2-Thinking when the register will store the data so we made a counter to handle this condition.

3-Initializing the register data by using Asynchronous (reset).

4-Starting the operations by another Asynchronous (Start).

5-Choose which slave will receive and send to Master.

6-Since SPI MODE is 1 so CPOL=0, CPHA=1, Clock Polarity in Idle State is logic LOW.

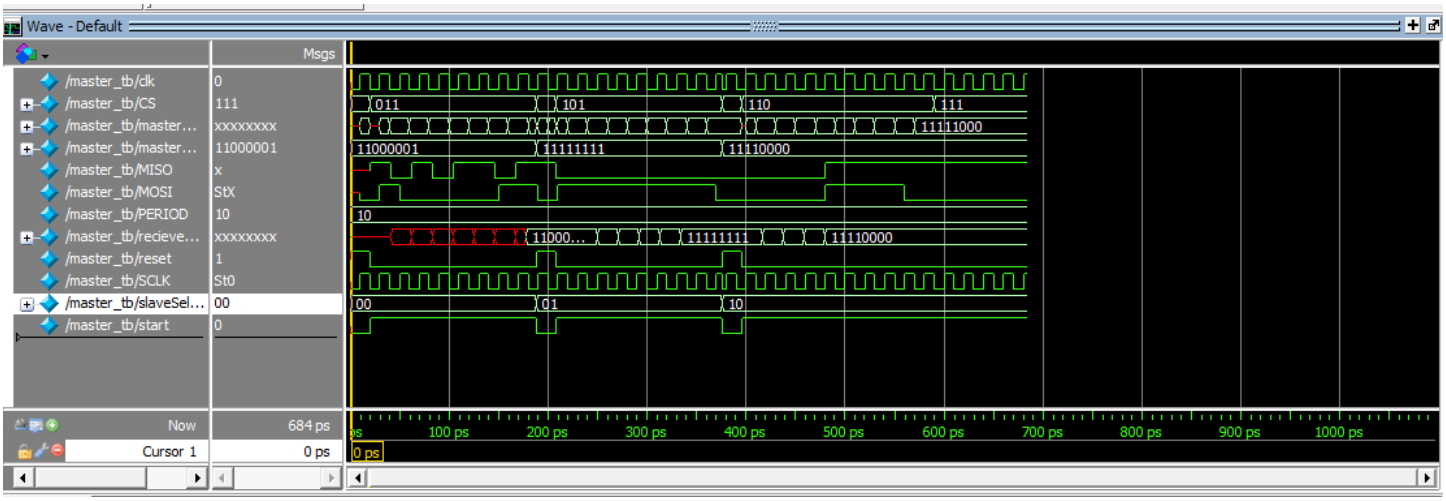
7-Making 2 Always Blocks

7.1: First Block is active rising edge, Shifting register and make MOSI is the LSB of the stored data in register.

7.2: Second Block is active falling edge, Sampling from MISO in MSB of the stored data in register

Master's Test Bench:

-Initializing Data and Test master by more test cases.



```
# Expected 11100000, Recieved 11100000
# Expected 01110000, Recieved 01110000
# Expected 10111000, Recieved 10111000
run
# Expected 01011100, Recieved 01011100
# Expected 10101110, Recieved 10101110
# Expected 11010111, Recieved 11010111
# Expected 01101011, Recieved 01101011
# Expected 10110101, Recieved 10110101
# First receiving test Succeeded :)
# First sending test Succeeded :)
run
# Expected 01111111, Recieved 01111111
# Expected 00111111, Recieved 00111111
# Expected 00011111, Recieved 00011111
# Expected 00001111, Recieved 00001111
run
# Expected 00000111, Recieved 00000111
# Expected 00000011, Recieved 00000011
# Expected 00000001, Recieved 00000001
# Expected 00000000, Recieved 00000000
# Second receiving test Succeeded :)
# Second sending test Succeeded :)

run
# Expected 01111111, Recieved 01111111
# Expected 00111111, Recieved 00111111
# Expected 00011111, Recieved 00011111
# Expected 00001111, Recieved 00001111
run
# Expected 00000111, Recieved 00000111
# Expected 00000011, Recieved 00000011
# Expected 00000001, Recieved 00000001
# Expected 00000000, Recieved 00000000
# Second receiving test Succeeded :)
# Second sending test Succeeded :)
run
# Expected 01111000, Recieved 01111000
# Expected 00111100, Recieved 00111100
# Expected 00011110, Recieved 00011110
# Expected 00001111, Recieved 00001111
run
# Expected 10000111, Recieved 10000111
# Expected 11000011, Recieved 11000011
# Expected 11100001, Recieved 11100001
# Expected 11110000, Recieved 11110000
# Third receiving test Succeeded :)
# Third sending test Succeeded :)
```

Slave:

Design Process:

1-Choose appropriate Port Connection and Type of Inputs and Outputs.

2-Getting Clock from Master (SCLK).

3-Making 3 Always Blocks:

3.1: First Block active at rising edge of SCLK:

If CS=0: MISO will be equal zero and shifting register data else it will make MISO in a high impedance state.

3.2: Second Block active at Falling edge of SCLK:

If CS=0: Sampling of MOSI to register data will happen.

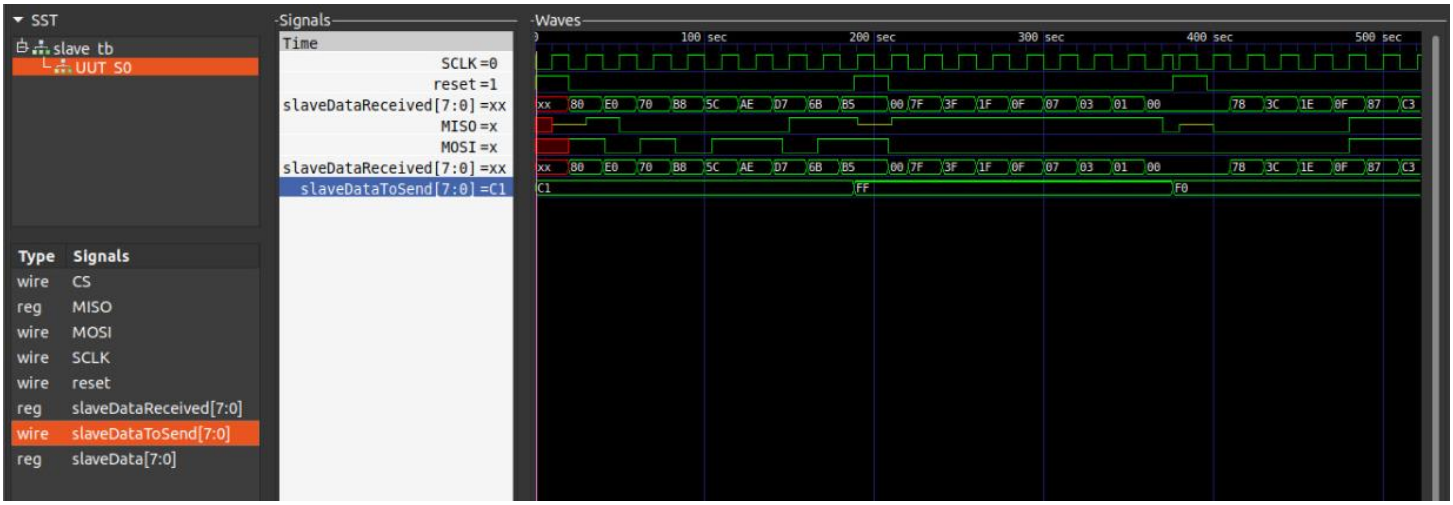
3.3: Third Block active at rising edge of reset or falling edge of CS:

3.3.1: When reset =1: register data will be equal zero bits.

3.3.2: When CS=0: the register data will sample the temporary data created before.

Slave's Test Bench:

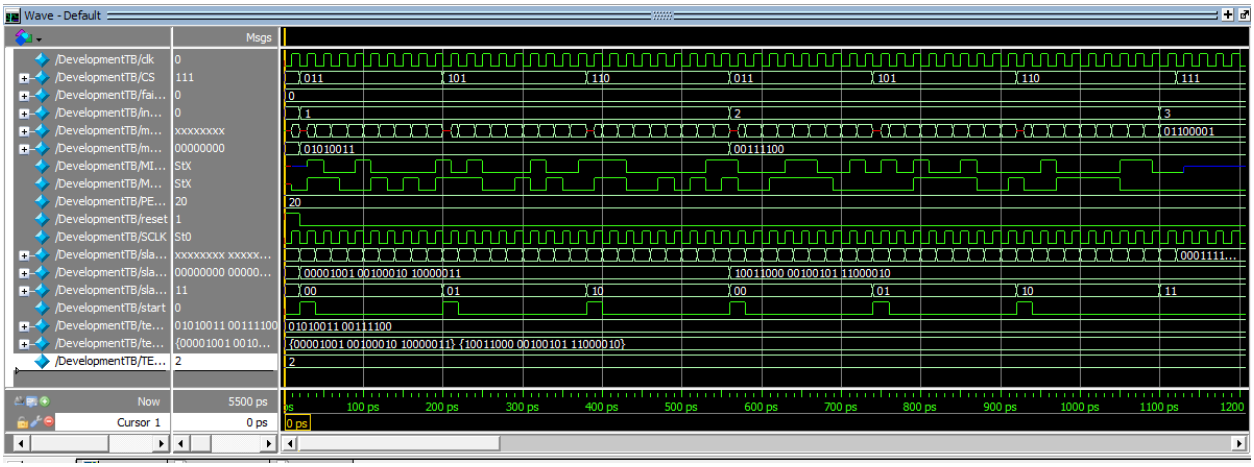
-Initializing data and testing slave using more test cases.



```
a_asaad22@ahmed-asaad22:~/Ahmed/college/CMP _1st/S_term/Logic/Project
$ vvp a.out
VCD info: dumpfile test.vcd opened for output.
Expected 11100000, Recieved 11100000
Expected 01110000, Recieved 01110000
Expected 10111000, Recieved 10111000
Expected 01011100, Recieved 01011100
Expected 10101110, Recieved 10101110
Expected 11010111, Recieved 11010111
Expected 01101011, Recieved 01101011
Expected 10110101, Recieved 10110101
First receiving test Succeeded :)
First sending test Succeeded :)
Expected 01111111, Recieved 01111111
Expected 00111111, Recieved 00111111
Expected 00011111, Recieved 00011111
Expected 00001111, Recieved 00001111
Expected 00000111, Recieved 00000111
Expected 00000011, Recieved 00000011
Expected 00000001, Recieved 00000001
Expected 00000000, Recieved 00000000
Second receiving test Succeeded :)
Second sending test Succeeded :)
Expected 01111000, Recieved 01111000
Expected 00111100, Recieved 00111100
Expected 00011110, Recieved 00011110
Expected 00001111, Recieved 00001111
Expected 10000111, Recieved 10000111
Expected 11000011, Recieved 11000011
Expected 11100001, Recieved 11100001
Expected 11110000, Recieved 11110000
Third receiving test Succeeded :)
Third sending test Succeeded :)
a_asaad22@ahmed-asaad22:~/Ahmed/college/CMP _1st
/S_term/Logic/Project$
```

Development test bench:

-Finally, this is the final output of integrating the master with 3 slave modules.




```
run
# From Slave 0 to Master: Success
# From Master to Slave 0: Success
run
run
# From Slave 1 to Master: Success
# From Master to Slave 1: Success
run
run
# From Slave 2 to Master: Success
# From Master to Slave 2: Success
# Running test set           2
run
run
# From Slave 0 to Master: Success
# From Master to Slave 0: Success
run
run
# From Slave 1 to Master: Success
# From Master to Slave 1: Success
run
# From Slave 2 to Master: Success
# From Master to Slave 2: Success
# SUCCESS: All          12 testcases have been successful
```