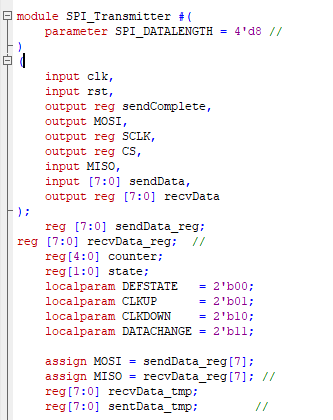
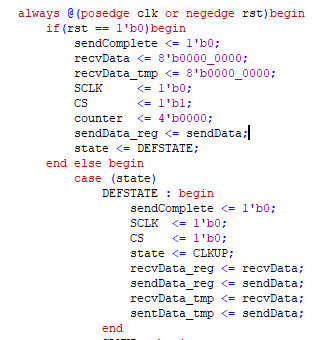
**The process of the master in transmitting**

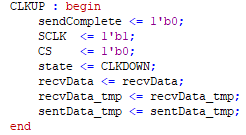
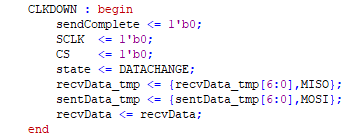
* **Declaration of some parameters**
* **reg to keep the data to be sent to the slave, and another one to keep the data to be received from the slave.**
* **MOSI is the output form master to the slave.**
* **MISO is the output from the slave to the master.**
* **There are 4 states for the master.**

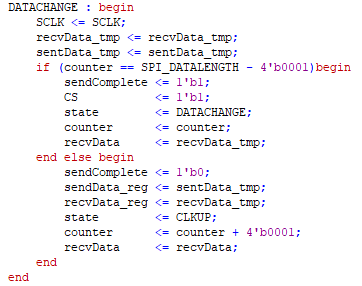
1. **Default**
2. **The clock is positive edge.**
3. **The clock is negative edge.**
4. **Data change is the last state.**

**Explanation of the four states: -**

****

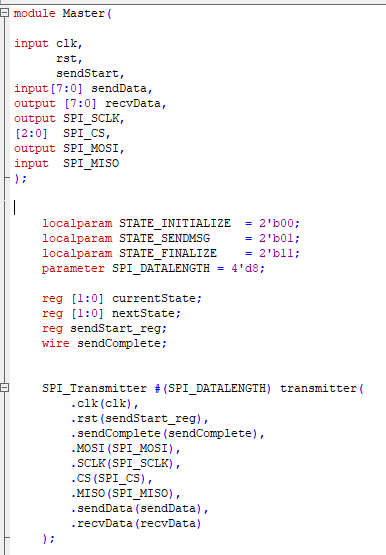
**Firstly, if we want to reset the modules, we will revert them to the default values ​​and the status will be the default state.**

1. **DEFSTATE**
   1. **Slave clock will be at the negative edge.**
   2. **Chip select will be low to make the slave start working.**
   3. **Reg of receiving data and send data will remain as it.**
   4. **State will be the next one which is Clock up.**
2. **CLKUP** 
   1. **Slave clock will be positive edge.**
   2. **Chip select will be low to make the slave start working.**
   3. **State will be the next one which is Clock down.**
3. **CLKDOWN**
   1. **Slave clock will be negative edge.**
   2. **Chip select will be low to make the slave start working.**
   3. **The data in the master will be shifted left by input sent from the slave.**
   4. **The data to be sent to the slave will be shifted left by the output from the master.**
   5. **State will be the next one which is data change.**

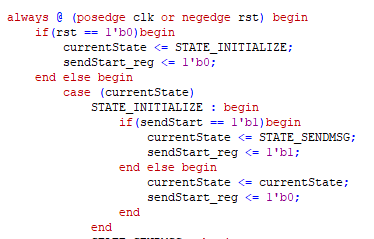
****

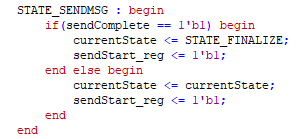
1. **DATACHANGE**
   1. **Slave clock will remain as it.**
   2. **If all the data in the master is transmitted to the slave (counter = 8), complete will be set as 1 (true) and chip select of the slave will be high to stop receiving.**
   3. **If not, we will copy data from the temporary register of sentData to the main reg.**
   4. **State will be the next one which is CLKUP.**
   5. **Counter will be incremented.**

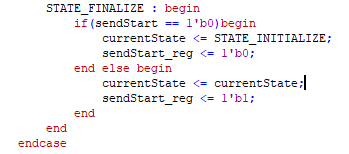
**Master Module**

* **Declaration of some parameters**
* **Reg for the current state and the next state.**
* **Reg for the starting signal**
* **Reg to determine which transmitting process completed or not.**
* **There are 3 states for the master.**
  + - 1. **Initialization.**
      2. **Sending data to the slave.**
      3. **Final state at the final clock.**
* **Calling for the transmitting module to transmit data from master to slave.**

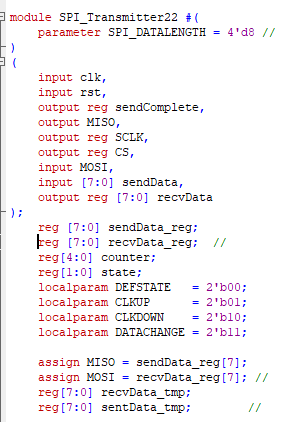
**Explanation of the three states: -**

**Firstly, if we want to reset the modules, we will revert them to the default values ​​and the status will be Initialization.**

1. **STATE\_INTITIALIZE**
   * **If the signal is high, we will go to the next state.**
   * **If not, we will stay in the current state.**
2. **STATE\_SENDMSG**
   * **If the start signal is high, we will go to the final state and set the register of the signal high.**
   * **If not, we will remain at the current state.**

****

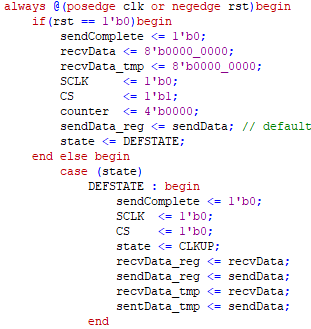
1. **STATE\_FINALIZE.**
   * **If the start signal is low, the state will be the first on (Initialization), and the signal remains as it.**
   * **If not, the signal will be high.**

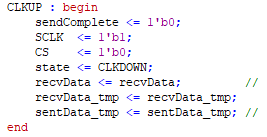
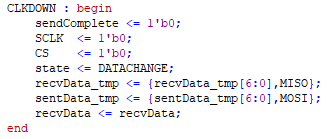
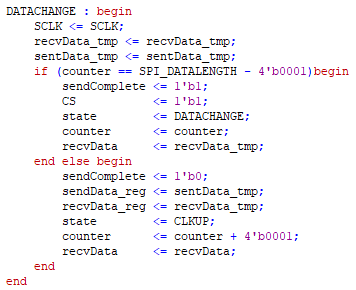
**The process of the slave in receiving.**

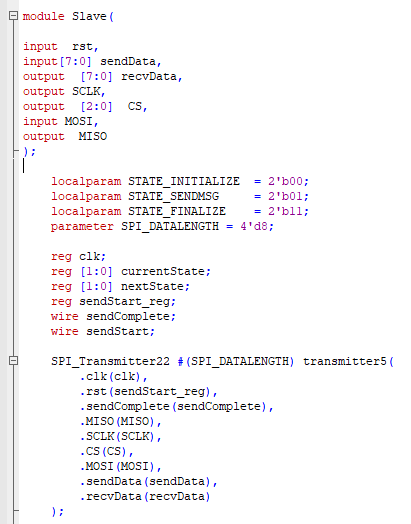
* **Declaration of some parameters**
* **reg to keep the data to be sent to the Master, and another one to keep the data to be received from the master.**
* **MOSI is the output form master to the slave.**
* **MISO is the output from the slave to the master.**
* **There are 4 states for the slave.**

1. **Default**
2. **The clock is positive edge.**
3. **The clock is negative edge.**
4. **Data change is the last state.**

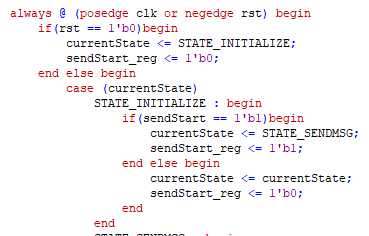
**Explanation of the four states: -**

**Firstly, if we want to reset the modules, we will revert them to the default values ​​and the status will be the default state.**

1. **DEFSTATE**
   1. **Slave clock will be at the negative edge.**
   2. **Chip select will be low to make the slave start working.**
   3. **Reg of receiving data and send data will remain as it.**
   4. **State will be the next one which is Clock up.**
2. **CLKUP** 
   1. **Slave clock will be positive edge.**
   2. **Chip select will be low to make the slave start working.**
   3. **State will be the next one which is Clock down.**
3. **CLKDOWN**
   1. **Slave clock will be negative edge.**
   2. **Chip select will be low to make the slave start working.**
   3. **The data in the slave will be shifted left by input sent from the master.**
   4. **The data to be sent to the master will be shifted left by the output from the slave.**
   5. **State will be the next one which is data change.**
4. **DATACHANGE**
   1. **Slave clock will remain as it.**
   2. **If all the data in the slave is transmitted to the master (counter = 8), complete will be set as 1 (true) and chip select of the slave will be high to stop receiving.**
   3. **If not, we will copy data from the temporary register of sentData to the main reg.**
   4. **State will be the next one which is CLKUP.**
   5. **Counter will be incremented.**

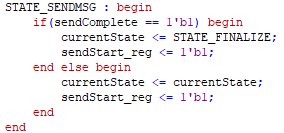
**Slave Module**

* **Declaration of some parameters**
* **Reg for the current state and the next state.**
* **Reg for the starting signal**
* **Reg to determine which transmitting process completed or not.**
* **There are 3 states for the master.**
  + - 1. **Initialization.**
      2. **Sending data to the master.**
      3. **Final state at the final coc.**
* **Calling for the transmitting module to transmit data from slave to master.**

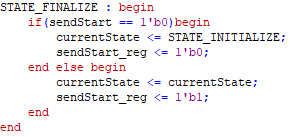
**Explanation of the three states: -**

**Firstly, if we want to reset the modules, we will revert them to the default values ​​and the status will be Initialization.**

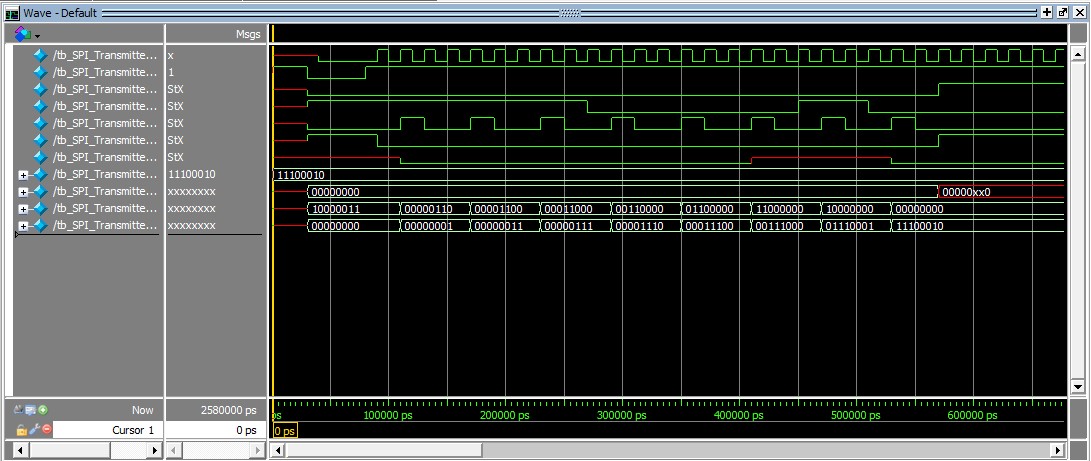
1. **STATE\_INTITIALIZE**
   * **If the signal is high, we will go to the next state.**
   * **If not, we will stay in the current state.**

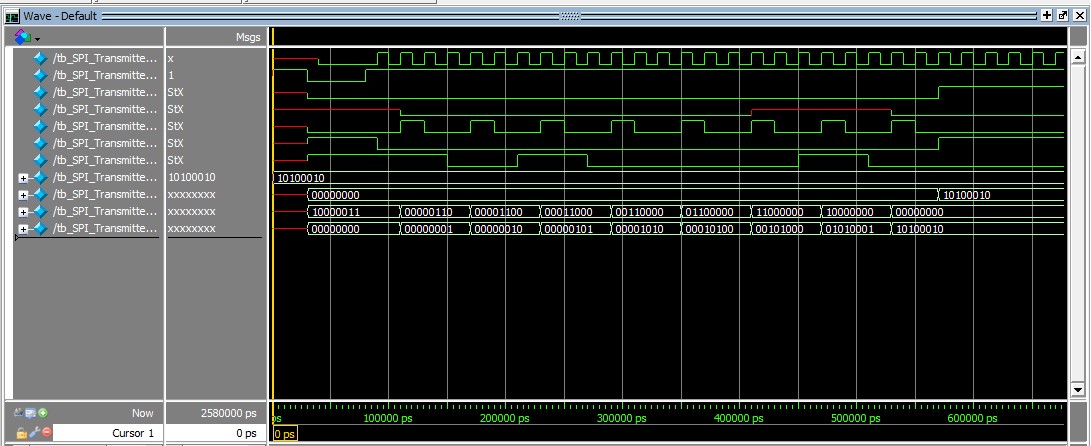
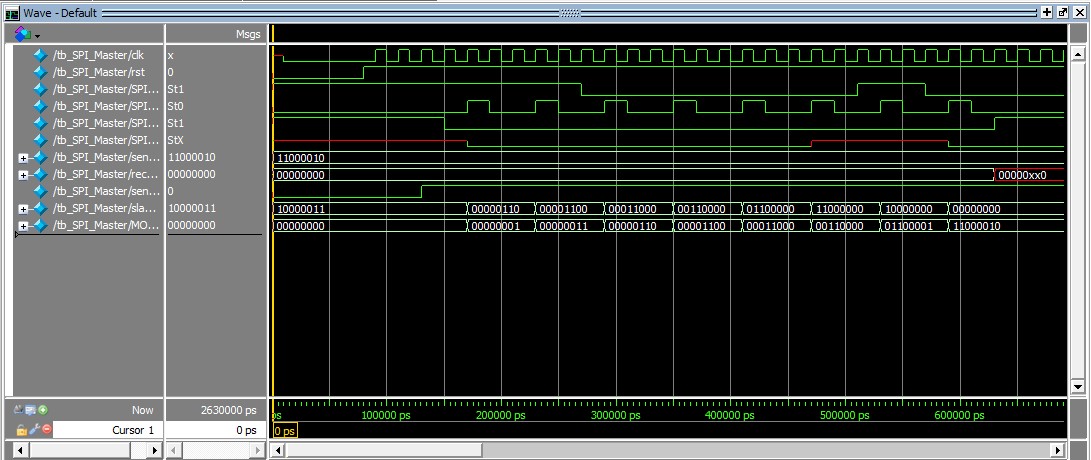
**2) STATE\_SENDMSG**

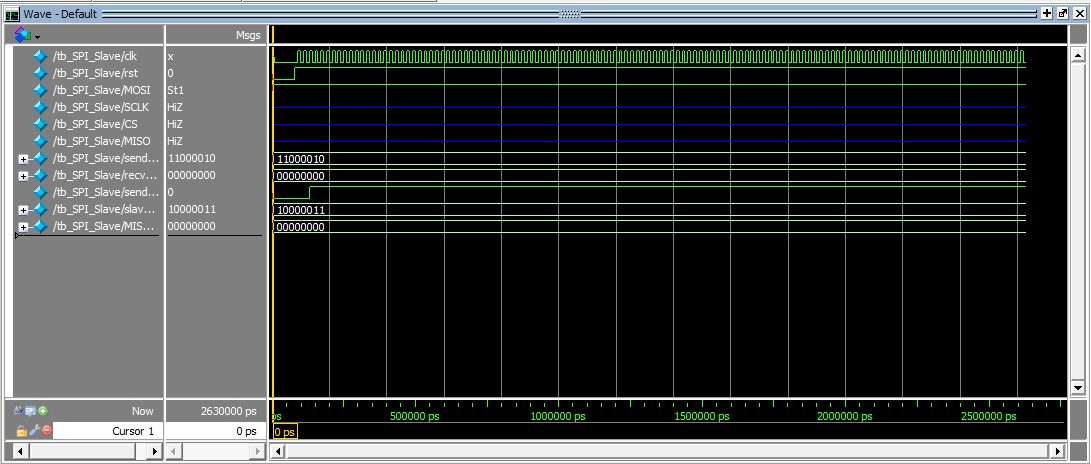
* + **If the start signal is high, we will go to the final state and set the register of the signal high.**
  + **If not, we will remain at the current state.**

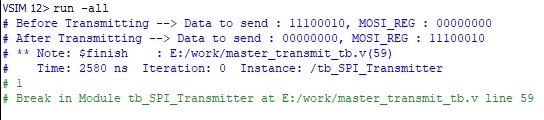
**3-STATE\_FINALIZE.**

* + **If the start signal is low, the state will be the first on (Initialization), and the signal remains as it.**
  + **If not, the signal will be high.**

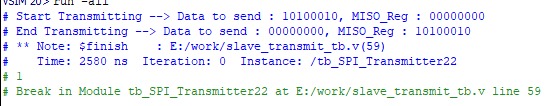
**Transmitter from Master to Slave Test Bench**

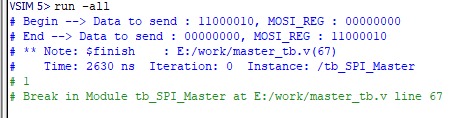
**Transmitter from Slave to Master Test BenchMaster Test Bench**

**Slave Test Bench**

**Transmitter from Master to Slave Output**

**Transmitter from Slave to Master Output**



**Master Output**