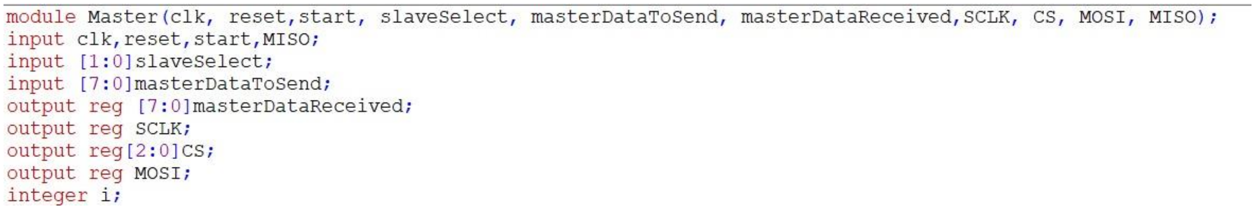
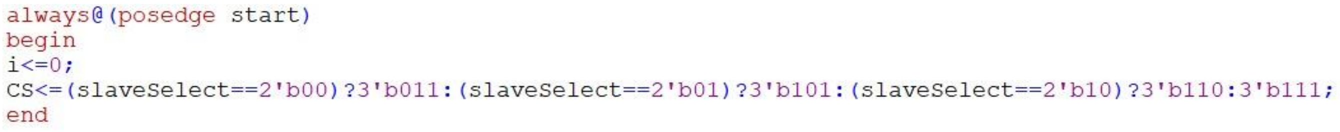
Master:

- Inputs and outputs ports and internal variables



```
module Master(clk, reset,start, slaveSelect, masterDataToSend, masterDataReceived,SCLK, CS, MOSI, MISO);
input clk,reset,start,MISO;
input [1:0]slaveSelect;
input [7:0]masterDataToSend;
output reg [7:0]masterDataReceived;
output reg SCLK;
output reg[2:0]CS;
output reg MOSI;
integer i;
```

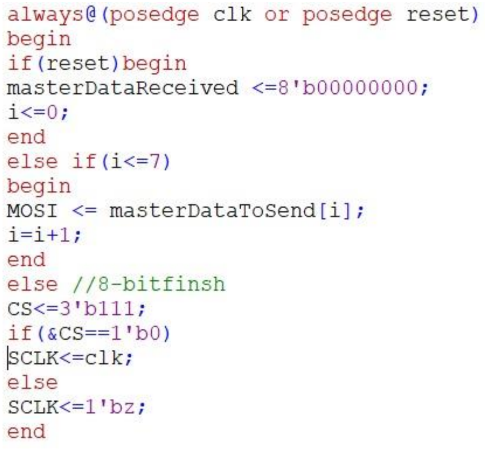
- Enabling Slaves



```
always@(posedge start)
begin
i<=0;
CS<=(slaveSelect==2'b00)?3'b011:(slaveSelect==2'b01)?3'b101:(slaveSelect==2'b10)?3'b110:3'b111;
end
```

According to slaveselect we enable the corresponding slave when slaveselect isn't 0,1,2, the all slaves are disabled

- Shifting (writing to MOSI@ posedge of clk)



```
always@(posedge clk or posedge reset)
begin
if(reset)begin
masterDataReceived <=8'b00000000;
i<=0;
end
else if(i<=7)
begin
MOSI <= masterDataToSend[i];
i=i+1;
end
else //8-bitfinsh
CS<=3'b111;
if(&CS==1'b0)
SCLK<=clk;
else
SCLK<=1'bz;
end
```

Case of reset assign masterDataReceived=0, initialize i(counter) by 0

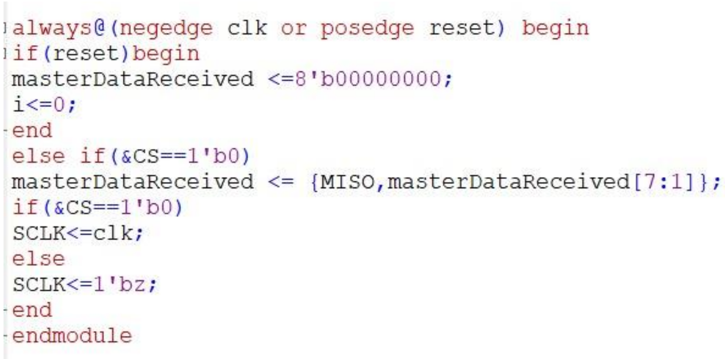
Assign MOSI by bit of masterDataToSend whose order is now (determined by i counter)

Case that 8 bits has already been sent disable the slaves(no need to send or receive data)

If there is a slave enabled assign SCLK to clk **[the positive 1/2 period of the clk]**

Else disconnect SCLK(no need for SCLK➔not used by any slave)

- Sampling (Master reads data from the MISO @ negedge of clk)



```
always@(negedge clk or posedge reset) begin
if(reset)begin
masterDataReceived <=8'b00000000;
i<=0;
end
else if(&CS==1'b0)
masterDataReceived <= {MISO,masterDataReceived[7:1]};
if(&CS==1'b0)
SCLK<=clk;
else
SCLK<=1'bz;
end
endmodule
```

Case of reset assign masterDataReceived=0, initialize i(counter) by 0

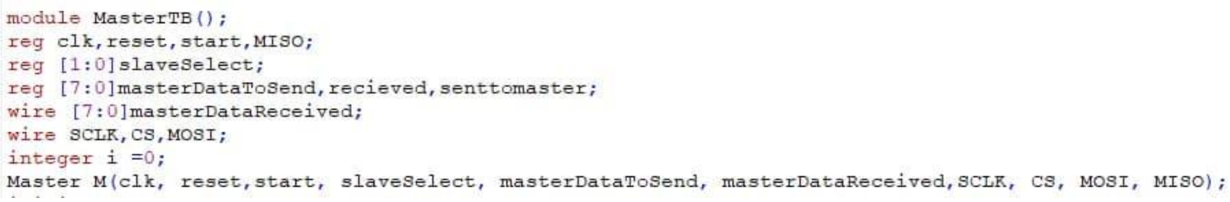
Case there is a slave enabled ➔receive MISO in the masterDataReceived

If there is a slave enabled assign SCLK to clk **[the negative 1/2 period of the clk]**

Else disconnect SCLK(no need for SCLK➔not used by any slave)

Master TB:

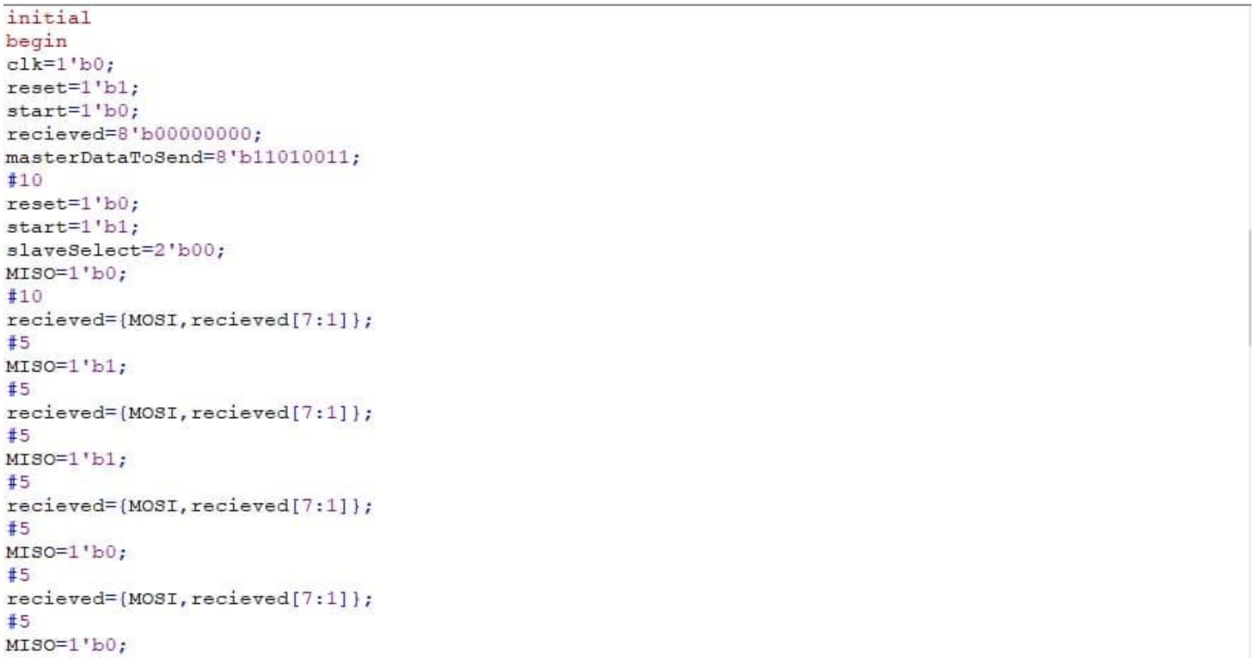
- Internal wires and regs



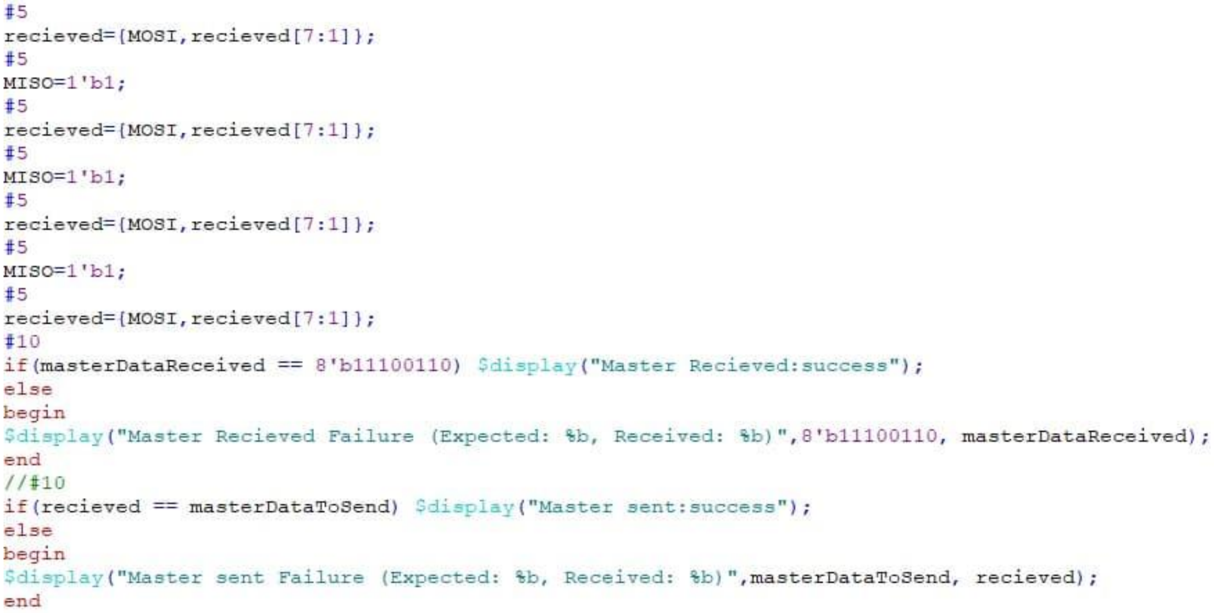
```
module MasterTB();
reg clk,reset,start,MISO;
reg [1:0]slaveSelect;
reg [7:0]masterDataToSend,recieved,senttomaster;
wire [7:0]masterDataReceived;
wire SCLK,CS,MOSI;
integer i =0;
Master M(clk, reset,start, slaveSelect, masterDataToSend, masterDataReceived,SCLK, CS, MOSI, MISO);
```

we have an instant of Master

- Initial block
  - ✓ First test case:



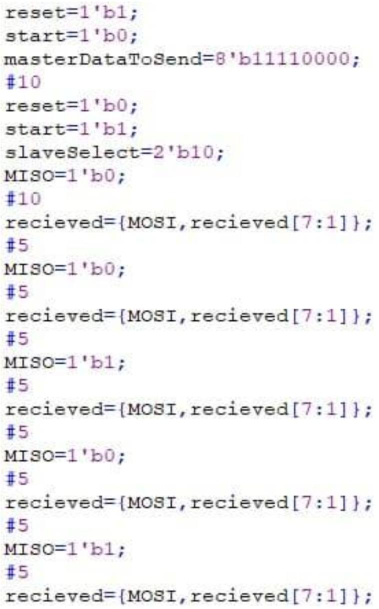
```
initial
begin
clk=1'b0;
reset=1'b1;
start=1'b0;
recieved=8'b00000000;
masterDataToSend=8'b11010011;
#10
reset=1'b0;
start=1'b1;
slaveSelect=2'b00;
MISO=1'b0;
#10
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b0;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b0;
```



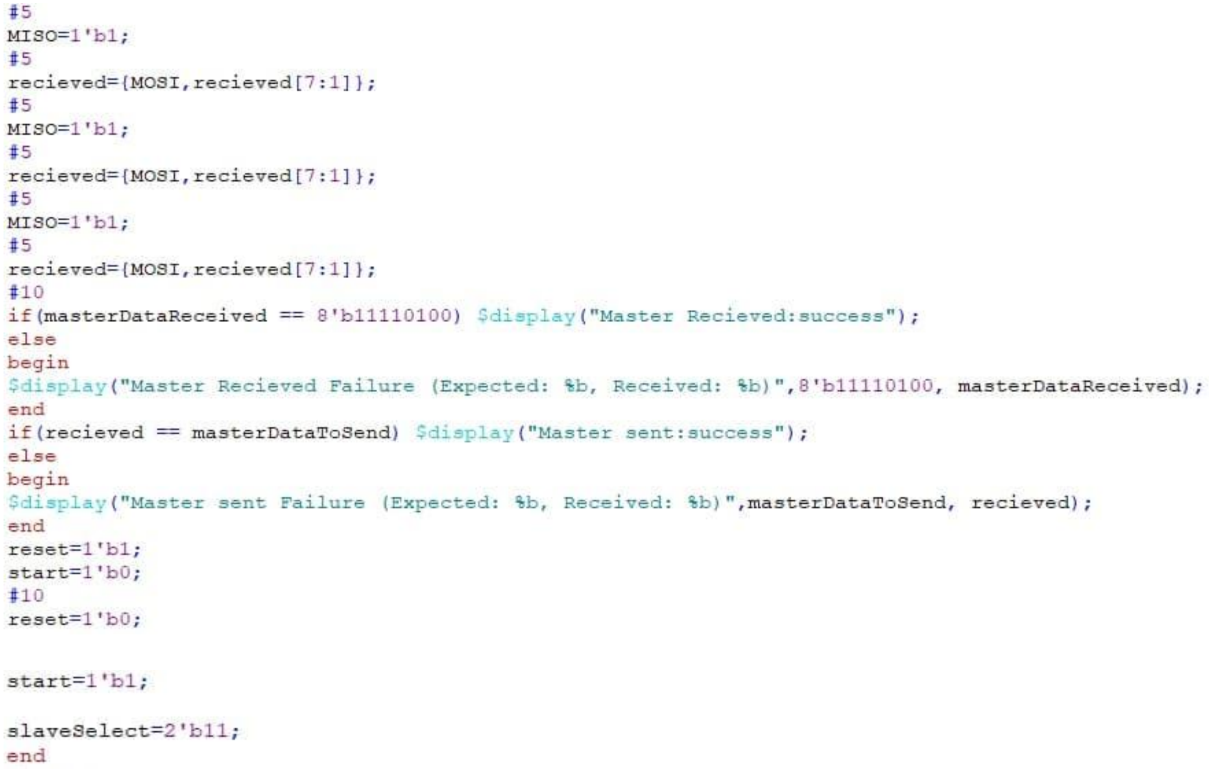
```
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
#10
if(masterDataReceived == 8'b11100110) $display("Master Recieved:success");
else
begin
$display("Master Recieved Failure (Expected: %b, Received: %b)",8'b11100110, masterDataReceived);
end
//#10
if(recieved == masterDataToSend) $display("Master sent:success");
else
begin
$display("Master sent Failure (Expected: %b, Received: %b)",masterDataToSend, recieved);
end
```

Testing if Successful receiving of master to 11100110 and sending 11010011

- ✓ Second test case:



```
reset=1'b1;
start=1'b0;
masterDataToSend=8'b11110000;
#10
reset=1'b0;
start=1'b1;
slaveSelect=2'b10;
MISO=1'b0;
#10
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b0;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b0;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
```



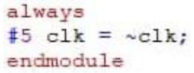
```
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
#5
MISO=1'b1;
#5
recieved={MOSI,recieved[7:1]};
#10
if(masterDataReceived == 8'b11110100) $display("Master Recieved:success");
else
begin
$display("Master Recieved Failure (Expected: %b, Received: %b)",8'b11110100, masterDataReceived);
end
if(recieved == masterDataToSend) $display("Master sent:success");
else
begin
$display("Master sent Failure (Expected: %b, Received: %b)",masterDataToSend, recieved);
end
reset=1'b1;
start=1'b0;
#10
reset=1'b0;

start=1'b1;

slaveSelect=2'b11;
end
```

Testing if Successful receiving of master to 11100110 and sending 11110000

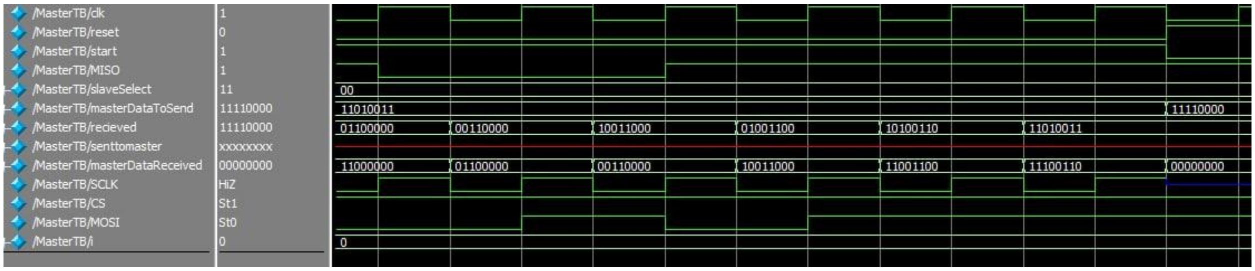
- Generating clk

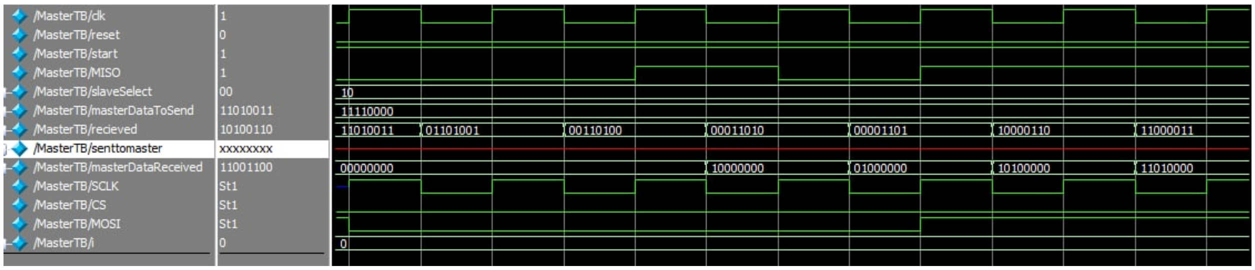


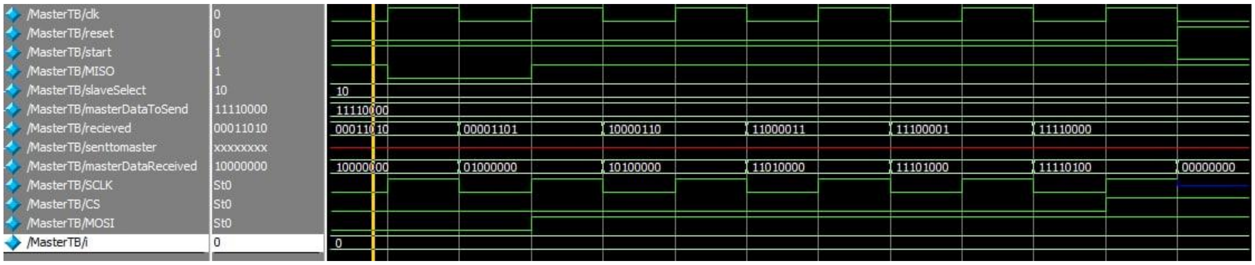
```
always
#5 clk = ~clk;
endmodule
```

Results:

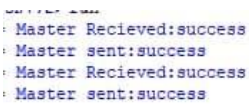
- ## Wave:







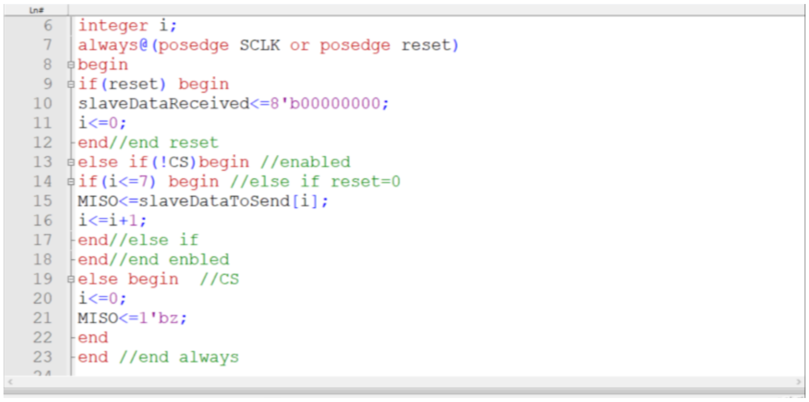
- ## Transcript:



```
Master Recieved:success
Master sent:success
Master Recieved:success
Master sent:success
```

SLAVE

- At positive edge of SCLK:

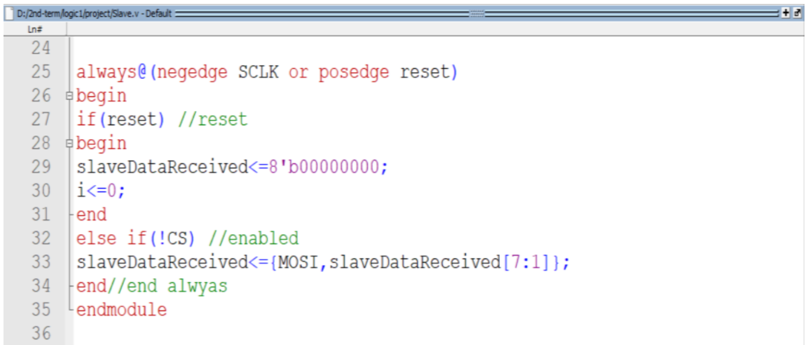


```
integer i;
always@(posedge SCLK or posedge reset)
begin
if(reset) begin
slaveDataReceived<=8'b00000000;
i<=0;
end//end reset
else if(!CS)begin //enabled
if(i<=7) begin //else if reset=0
MISO<=slaveDataToSend[i];
i<=i+1;
end//else if
end//end enbled
else begin  //CS
i<=0;
MISO<=1'bz;
end
end //end always
```

Shifting occur at which MISO read data from SlaveDataTosend (which is masters' input)

after master separated from slave counter i "which is used to send SlaveDataTosend to master bit by bit " return to zero again

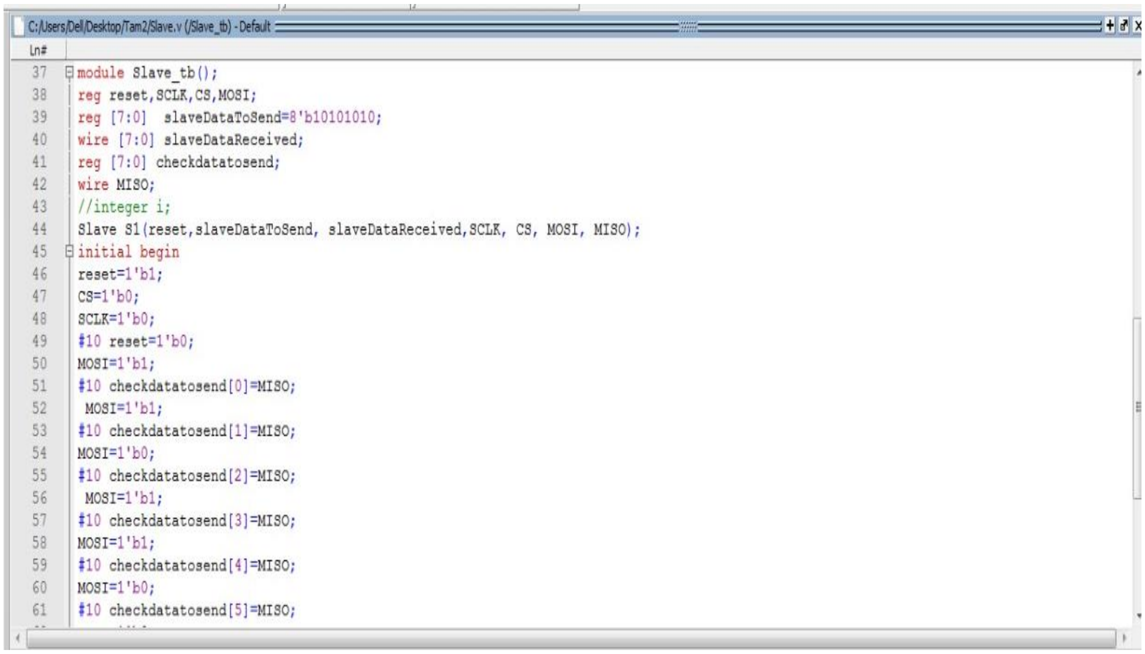
- At negative edge of SCLK:



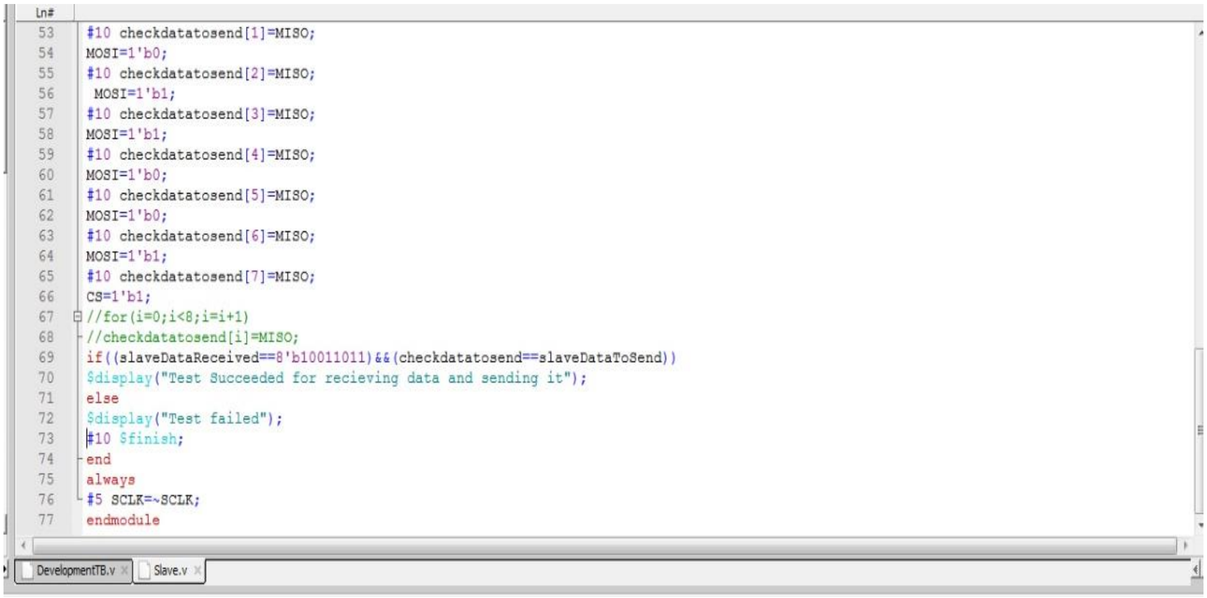
```
always@(negedge SCLK or posedge reset)
begin
if(reset) //reset
begin
slaveDataReceived<=8'b00000000;
i<=0;
end
else if(!CS) //enabled
slaveDataReceived<={MOSI,slaveDataReceived[7:1]};
end//end alwyas
endmodule
```

Sampling occurs at which slave "slaveDataReceived" is filled by MOSI bit by bit from left to right.

SlaveTB:



```
module Slave_tb();
reg reset,SCLK,CS,MOSI;
reg [7:0]  slaveDataToSend=8'b10101010;
wire [7:0] slaveDataReceived;
reg [7:0] checkdatatosend;
wire MISO;
//integer i;
Slave S1(reset,slaveDataToSend, slaveDataReceived,SCLK, CS, MOSI, MISO);
initial begin
reset=1'b1;
CS=1'b0;
SCLK=1'b0;
#10 reset=1'b0;
MOSI=1'b1;
#10 checkdatatosend[0]=MISO;
 MOSI=1'b1;
#10 checkdatatosend[1]=MISO;
MOSI=1'b0;
#10 checkdatatosend[2]=MISO;
 MOSI=1'b1;
#10 checkdatatosend[3]=MISO;
MOSI=1'b1;
#10 checkdatatosend[4]=MISO;
MOSI=1'b0;
#10 checkdatatosend[5]=MISO;
```



```
#10 checkdatatosend[1]=MISO;
MOSI=1'b0;
#10 checkdatatosend[2]=MISO;
 MOSI=1'b1;
#10 checkdatatosend[3]=MISO;
MOSI=1'b1;
#10 checkdatatosend[4]=MISO;
MOSI=1'b0;
#10 checkdatatosend[5]=MISO;
MOSI=1'b0;
#10 checkdatatosend[6]=MISO;
MOSI=1'b1;
#10 checkdatatosend[7]=MISO;
CS=1'b1;
//for(i=0;i<8;i=i+1)
//checkdatatosend[i]=MISO;
if((slaveDataReceived==8'b10011011)&&(checkdatatosend==slaveDataToSend))
$display("Test Succeeded for recieving data and sending it");
else
$display("Test failed");
#10 $finish;
end
always
#5 SCLK=~SCLK;
endmodule
```

reset is done to give i initial value =0

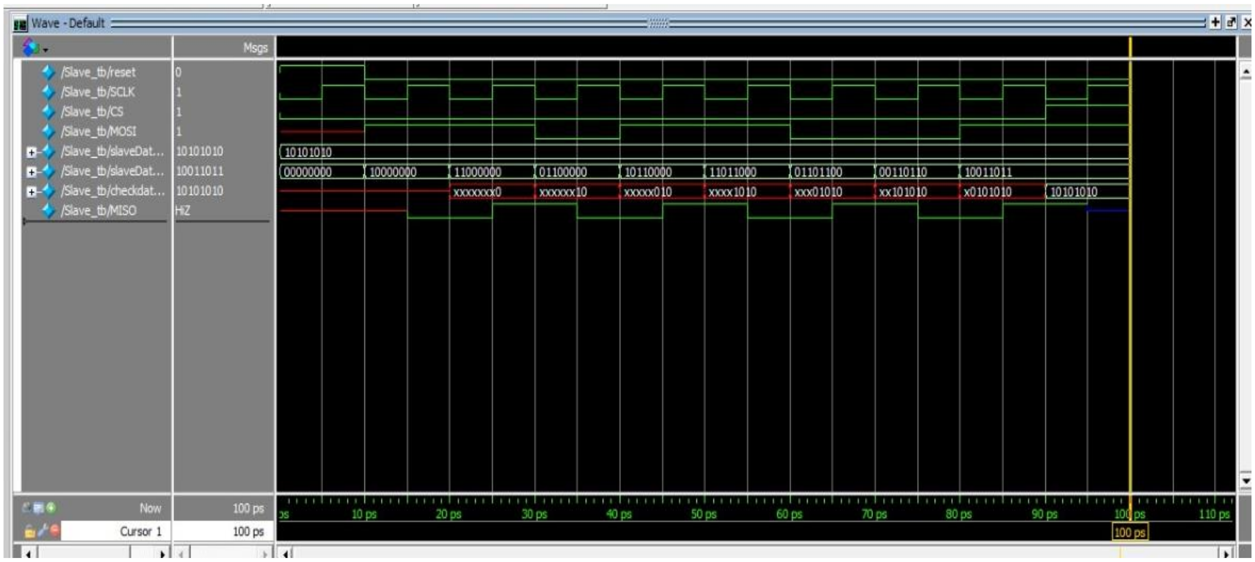
cycle of clock is 10 and value of MOSI change at every cycle and

store MISO in "checkdatatosend"

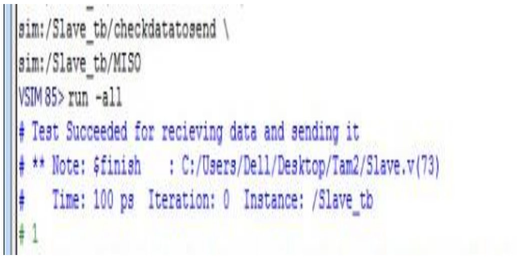
self-checking TB to check if the slave send and receive correct

Results:

- ## Wave:



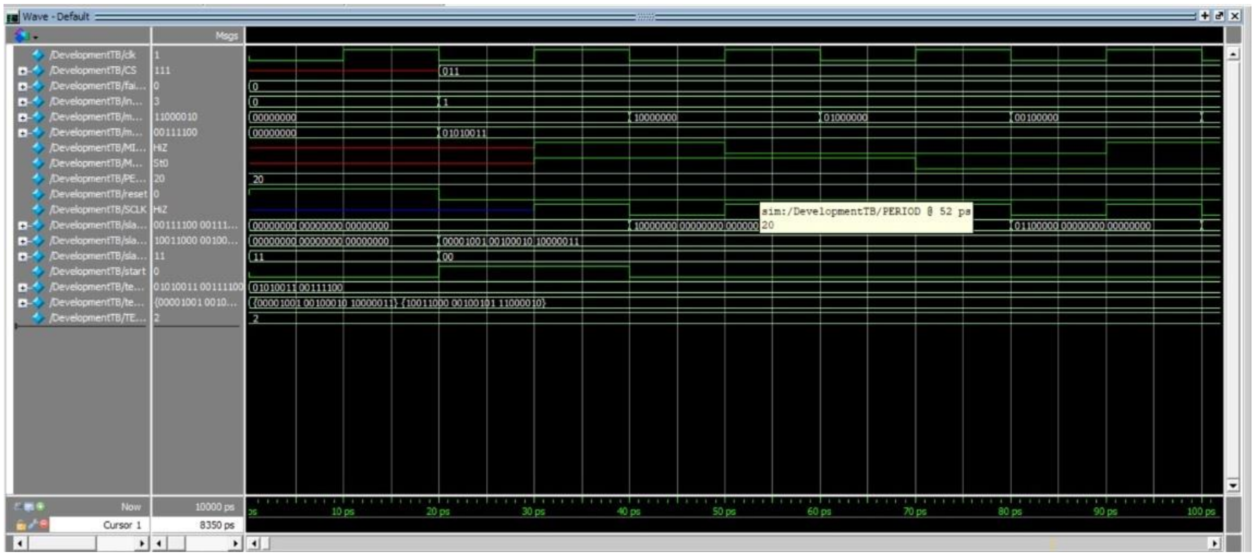
- ## Transcript:



```
sim:/Slave_tb/checkdatatosend \
sim:/Slave_tb/MISO
VSIM 85> run -all
# Test Succeeded for recieving data and sending it
# ** Note: $finish    : C:/Users/Dell/Desktop/Tam2/Slave.v(73)
#    Time: 100 ps  Iteration: 0  Instance: /Slave_tb
# 1
```

Development TB:

- Wave:

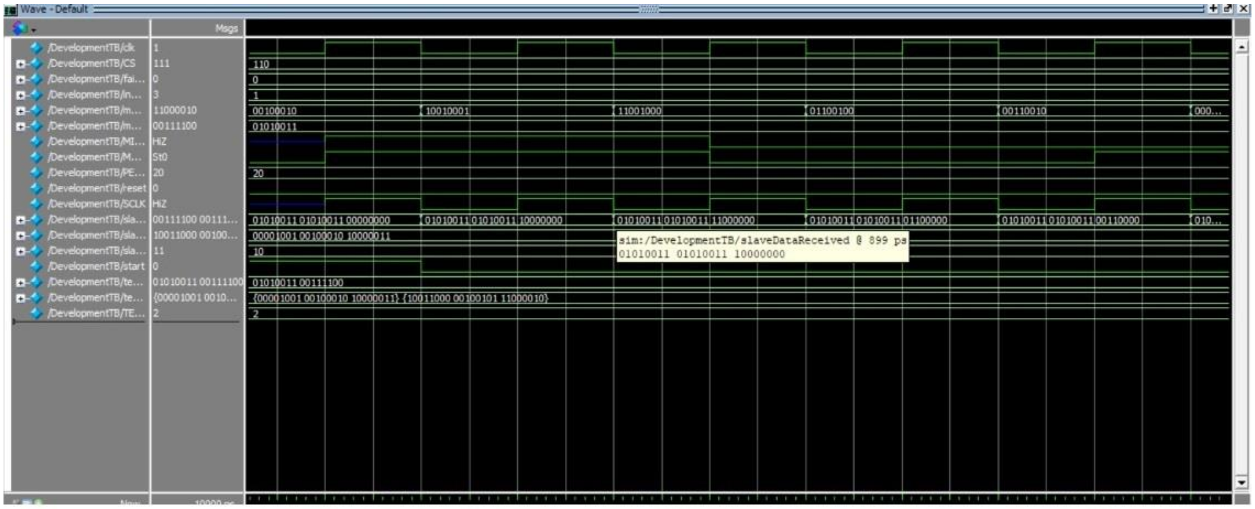


Wave - Default

	Msgs
/DevelopmentTB/clk	1
/DevelopmentTB/CS	111
/DevelopmentTB/fai...	0
/DevelopmentTB/in...	3
/DevelopmentTB/m...	11000010
/DevelopmentTB/m...	00111100
/DevelopmentTB/MI...	HiZ
/DevelopmentTB/M...	St0
/DevelopmentTB/PE...	20
/DevelopmentTB/reset	0
/DevelopmentTB/SCLK	HiZ
/DevelopmentTB/sla...	00111100 00111...
/DevelopmentTB/sla...	10011000 00100...
/DevelopmentTB/sla...	11
/DevelopmentTB/start	0
/DevelopmentTB/te...	01010011 00111100
/DevelopmentTB/te...	{00001001 0010...
/DevelopmentTB/TE...	2

Now 10000 ps
Cursor 1 8350 ps

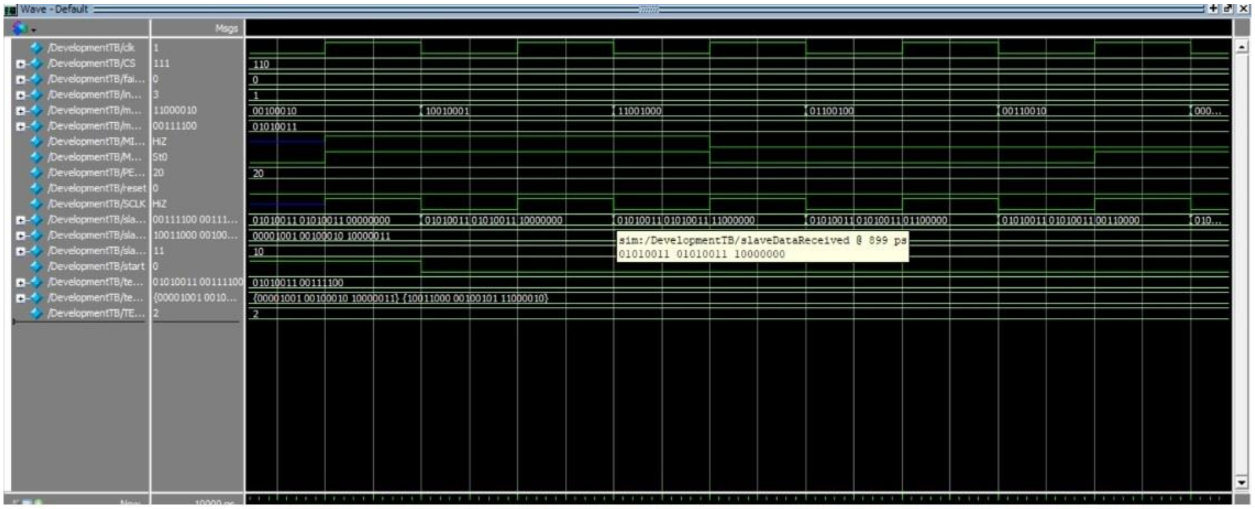
sim:/DevelopmentTB/PERIOD @ 52 ps
20



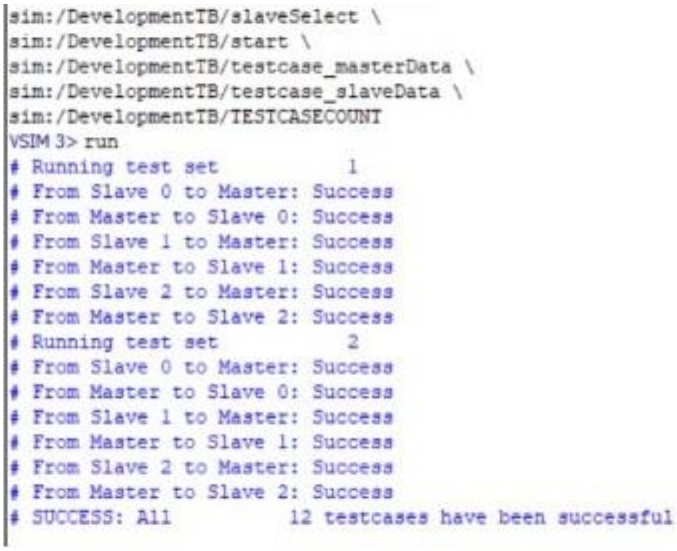
Wave - Default

	Msgs
/DevelopmentTB/clk	1
/DevelopmentTB/CS	111
/DevelopmentTB/fai...	0
/DevelopmentTB/in...	3
/DevelopmentTB/m...	11000010
/DevelopmentTB/m...	00111100
/DevelopmentTB/MI...	HiZ
/DevelopmentTB/M...	St0
/DevelopmentTB/PE...	20
/DevelopmentTB/reset	0
/DevelopmentTB/SCLK	HiZ
/DevelopmentTB/sla...	00111100 00111...
/DevelopmentTB/sla...	10011000 00100...
/DevelopmentTB/sla...	11
/DevelopmentTB/start	0
/DevelopmentTB/te...	01010011 00111100
/DevelopmentTB/te...	{00001001 0010...
/DevelopmentTB/TE...	2

sim:/DevelopmentTB/slaveDataReceived @ 899 ps
01010011 01010011 10000000


Wave - Default
Msgs
/DevelopmentTB/clk 1
/DevelopmentTB/CS 111 110
/DevelopmentTB/fai... 0 0
/DevelopmentTB/in... 3 1
/DevelopmentTB/m... 11000010 00100010 10010001 11001000 01100100 00110010 000...
/DevelopmentTB/m... 00111100 01010011
/DevelopmentTB/MI... HiZ
/DevelopmentTB/M... St0
/DevelopmentTB/PE... 20 20
/DevelopmentTB/reset 0
/DevelopmentTB/SCLK HiZ
/DevelopmentTB/sla... 00111100 00111... 01010011 01010011 00000000 01010011 01010011 10000000 01010011 01010011 11000000 01010011 01010011 01100000 01010011 01010011 00110000 010...
/DevelopmentTB/sla... 10011000 00100... 00001001 00100010 10000011
/DevelopmentTB/sla... 11 10
/DevelopmentTB/start 0
/DevelopmentTB/te... 01010011 00111100 01010011 00111100
/DevelopmentTB/te... {00001001 0010... {00001001 00100010 10000011} {10011000 00100101 11000010}
/DevelopmentTB/TE... 2 2
sim:/DevelopmentTB/slaveDataReceived @ 899 ps
01010011 01010011 10000000

- Transcript:



```
sim:/DevelopmentTB/slaveSelect \
sim:/DevelopmentTB/start \
sim:/DevelopmentTB/testcase_masterData \
sim:/DevelopmentTB/testcase_slaveData \
sim:/DevelopmentTB/TESTCASECOUNT
VSIM 3> run
# 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
```

All test cases passed