

# SPI DESIGN REPORT

## VLSI Internship

---



### Introduction

# 21BEC1085

SERIAL PERIPHERAL INTERFACE (SPI) IS A MASTER – SLAVE TYPE PROTOCOL THAT PROVIDES A SIMPLE AND LOW COST INTERFACE BETWEEN A MICROCONTROLLER AND ITS PERIPHERALS ICs SUCH AS SENSORS, ADC'S, DAC'S, SHIFT REGISTER, SRAM AND OTHERS. IN SPI PROTOCOL, THERE CAN BE ONLY ONE MASTER BUT MANY SLAVE DEVICES.

THE SPI BUS CONSISTS OF 4 SIGNALS OR PINS. THEY ARE

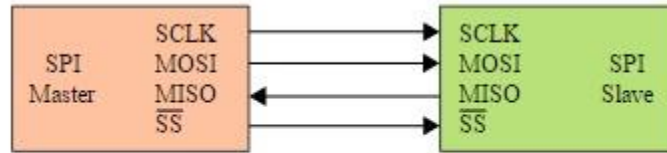
- MASTER – OUT / SLAVE – IN (MOSI)
- MASTER – IN / SLAVE – OUT (MISO)
- SERIAL CLOCK (SCLK) AND
- SLAVE SELECT (SS)

DATA IS EXCHANGED SIMULTANEOUSLY BETWEEN DEVICES IN A SERIAL MANNER

(SHIFTED OUT SERIALLY AND SHIFTED IN SERIALLY). SERIAL CLOCK LINE SYNCHRONIZES

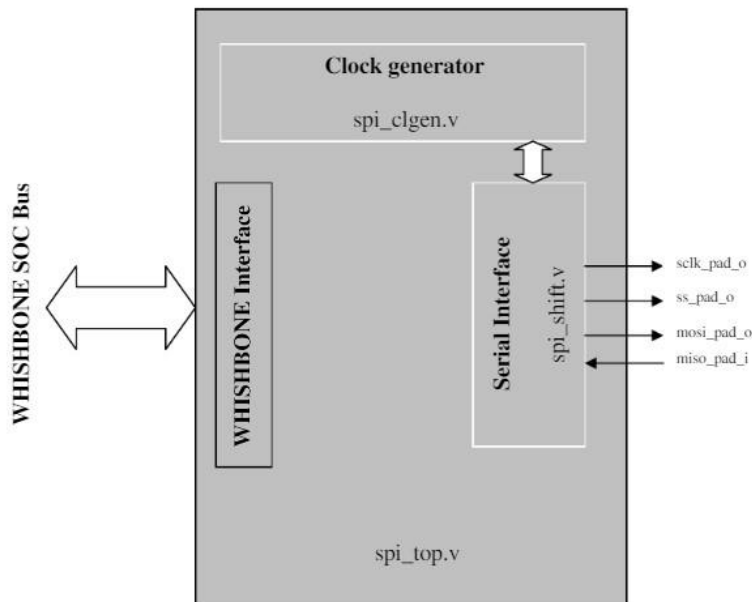
SHIFTING AND SAMPLING OF INFORMATION ON TWO SERIAL DATA LINES. A SLAVE

SELECT LINE ALLOWS INDIVIDUAL SELECTION OF SLAVE SPI DEVICES. 1



## ARCHITECTURE OF SPI MASTER CORE

THE SPI MASTER CORE CONSISTS OF THREE PARTS SHOWN IN THE FOLLOWING FIGURE:

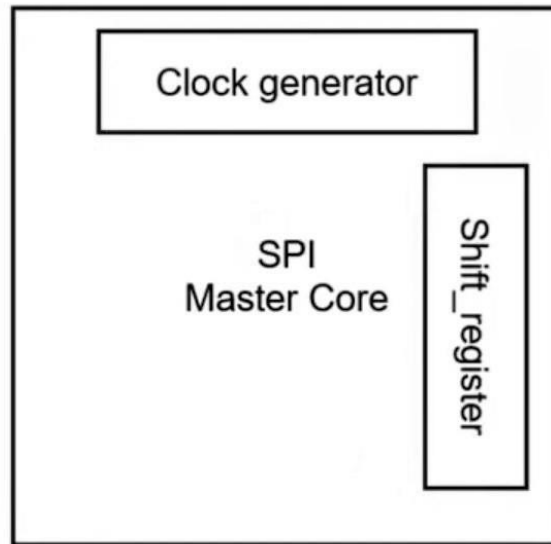


FEATURES OF SPI MASTER CORE:

- FULL DUPLEX SYNCHRONOUS SERIAL DATA TRANSFER
- VARIABLE LENGTH OF TRANSFER WORD UP TO 128 BITS
- MSB OR LSB FIRST DATA TRANSFER
- RX AND TX ON BOTH RISING OR FALLING EDGE OF SERIAL CLOCK INDEPENDENTLY
- 8 SLAVE SELECT LINES
- FULLY STATIC SYNCHRONOUS DESIGN WITH ONE CLOCK DOMAIN
- TECHNOLOGY INDEPENDENT VERILOG
- FULLY SYNTHESIZABLE

## SPI MASTER CORE INTERFACE

SPI MASTER CORE MODULE IS A COMBINATION OF CLOCK GENERATOR AND SHIFT REGISTER BLOCKS.



MODULE "SPI.DEFINES.V" IS USED TO INITIALIZE ALL THE VARIABLE VALUES RELATED TO REGISTERS ( CTRL, SS, DIVIDER AND SPI REGISTERS).

CONTROL AND STATUS REGISTER BUS IS AS FOLLOWS:

Bit#	31:14	13	12	11	10	9	8	7	6:0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Name	Reserved	ASS	IE	LSB	TX_NEG	RX_NEG	GO_BUSY	Reserved	CHAR_LEN

SLAVE SELECT REGISTER BUS IS AS FOLLOWS:

Bit#	31:8	7:0
Access	R	R/W
Name	Reserved	SS

DIVIDER REGISTER BUS IS AS FOLLOWS:

Bit#	31:16	15:0
Access	R	R/W
Name	Reserved	DIVIDER

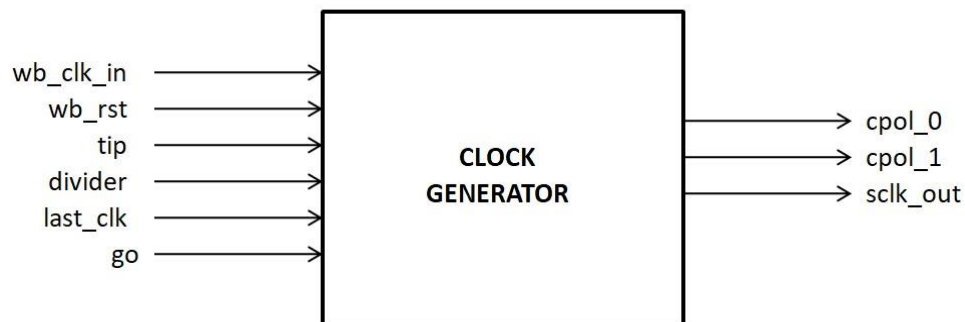
TXX AND RXXREGISTER BUS ARE AS FOLLOWS:

Data receive registers[RxX]	
Bit#	31:0
Access	R
Name	Rx

Data transmit register [TxX]	
Bit#	31:0
Access	R/W
Name	Tx

### CLOCK GENERATOR BLOCK



MODULE DECLARATION: DEFINES THE MODULE SPI\_CLGEN WITH INPUT AND OUTPUT PORTS.

- INPUTS:
  - ❖ WB\_CLK\_IN: INPUT CLOCK SIGNAL.
  - ❖ WB\_RST: RESET SIGNAL.
  - ❖ GO: SIGNAL INDICATING THE START OF THE TRANSFER.
  - ❖ TIP: SIGNAL INDICATING WHETHER A TRANSFER IS IN PROGRESS.
  - ❖ LAST\_CLK: SIGNAL INDICATING THE LAST CLOCK EDGE OF THE TRANSFER.
  - ❖ DIVIDER: INPUT VALUE DETERMINING THE CLOCK DIVIDER FOR GENERATING THE SPI CLOCK.

- OUTPUTS:
  - ❖ SCLK\_OUT: OUTPUT SPI CLOCK SIGNAL.
  - ❖ CPOL\_0: OUTPUT SIGNAL INDICATING THE PULSE MARKING THE POSITIVE EDGE OF THE SPI CLOCK.
  - ❖ CPOL\_1: OUTPUT SIGNAL INDICATING THE PULSE MARKING THE NEGATIVE EDGE OF THE SPI CLOCK.

THE CLOCK GENERATOR BLOCK GENERATES TWO CLOCK SIGNALS: RX\_CLK AND TX\_CLK. THESE CLOCK SIGNALS ARE USED TO CONTROL THE RECEIVE AND TRANSMIT OPERATIONS OF THE SPI SHIFT REGISTER. THE GENERATION OF THESE CLOCKS IS BASED ON THE INPUT SIGNALS RX\_NEGEDGE, TX\_NEGEDGE, LAST, SCLK, CPOL\_0, AND CPOL\_1.

THE VALUE GIVEN TO DIVIDER FIELD IS THE FREQUENCY DIVIDER OF THE SYSTEM CLOCK WB\_CLK\_I TO GENERATE THE SERIAL CLOCK ON THE OUTPUT SCLK\_PAD\_O. THE DESIRED FREQUENCY IS OBTAINED ACCORDING TO THE FOLLOWING EQUATION:

$$f_{sclk} = \frac{f_{wb\_clk}}{(DIVIDER+1)*2}$$

THE RX\_CLK SIGNAL IS GENERATED BY COMBINING THE RX\_NEGEDGE INPUT SIGNAL AND THE LOGICAL OR OF LAST AND SCLK. IT IS USED AS THE RECEIVE CLOCK ENABLE. THE TX\_CLK SIGNAL IS GENERATED BY COMBINING THE TX\_NEGEDGE INPUT SIGNAL AND THE LOGICAL AND OF LAST AND THE SELECTED CLOCK POLARITY (CPOL\_0 OR CPOL\_1). IT IS USED AS THE TRANSMIT CLOCK ENABLE.

THE CLOCK GENERATOR BLOCK ENSURES THAT THE CLOCK SIGNALS ARE GENERATED APPROPRIATELY BASED ON THE SPECIFIED CLOCK POLARITIES AND THE TRANSFER STATUS (LAST). THIS ENABLES PROPER SYNCHRONIZATION AND TIMING CONTROL FOR THE SPI SHIFT REGISTER MODULE DURING DATA TRANSMISSION AND RECEPTION.

The diagram shows a central rectangular block labeled "SHIFT REGISTER". On the left side, there are 14 input lines with arrows pointing into the block, labeled from top to bottom: wb\_clk\_in, wb\_rst\_in, latch, byte\_sel, lsb, sclk, cpol\_0, cpol\_1, len, miso, Pin, go, tx\_nedge, and rx\_nedge. On the right side, there are 4 output lines with arrows pointing out of the block, labeled from top to bottom: Pout, tip, mosi, and last.

- INPUT:
  - ❖ RX\_NEGEDGE: INPUT SIGNAL, INDICATES THE NEGATIVE EDGE OF THE RECEIVE CLOCK.
  - ❖ TX\_NEGEDGE: INPUT SIGNAL, INDICATES THE NEGATIVE EDGE OF THE TRANSMIT CLOCK.
  - ❖ BYTE\_SEL: 4-BIT INPUT SIGNAL USED FOR BYTE SELECTION.
  - ❖ LATCH: INPUT SIGNAL USED FOR LATCH CONTROL.
  - ❖ LEN: 32-BIT INPUT SIGNAL REPRESENTING THE LENGTH OF THE CHARACTER.
  - ❖ P\_IN: 32-BIT INPUT SIGNAL FOR THE INCOMING DATA.
  - ❖ WB\_CLK\_IN: INPUT CLOCK SIGNAL (SYSTEM CLOCK).

- ❖ WB\_RST: INPUT RESET SIGNAL.
- ❖ GO: INPUT SIGNAL TO INITIATE THE TRANSFER.
- ❖ MISO: INPUT SIGNAL FOR THE MASTER INPUT SLAVE OUTPUT.
- ❖ LSB: INPUT SIGNAL INDICATING THE LEAST SIGNIFICANT BIT.
- ❖ CPOL\_0: INPUT SIGNAL FOR THE PULSE MARKING THE POSITIVE EDGE OF THE SCLK\_OUT.
- ❖ CPOL\_1: INPUT SIGNAL FOR THE PULSE MARKING THE NEGATIVE EDGE OF THE SCLK\_OUT.
- OUTPUT:
  - ❖ P\_OUT: OUTPUT SIGNAL REPRESENTING THE SHIFTED DATA FROM THE SHIFT REGISTER. IT IS SPI\_MAX\_CHAR BITS WIDE.
  - ❖ LAST: OUTPUT SIGNAL INDICATING WHETHER THE LAST CHARACTER IS BEING TRANSMITTED OR RECEIVED.
  - ❖ MOSI: OUTPUT SIGNAL REPRESENTING THE SERIAL OUTPUT DATA FROM THE SHIFT REGISTER.
  - ❖ TIP: OUTPUT SIGNAL INDICATING IF A TRANSFER IS IN PROGRESS OR NOT.
  - ❖ RX\_CLK: OUTPUT SIGNAL REPRESENTING THE RECEIVE CLOCK ENABLE.
  - ❖ TX\_CLK: OUTPUT SIGNAL REPRESENTING THE TRANSMIT CLOCK ENABLE.
- INTERNAL REGISTERS:
  - ❖ CHAR\_COUNT: REGISTER USED FOR COUNTING THE NUMBER OF BITS IN A CHARACTER.
  - ❖ MASTER\_DATA: REGISTER REPRESENTING THE SHIFT REGISTER HOLDING THE DATA.
  - ❖ TX\_BIT\_POS: REGISTER INDICATING THE NEXT BIT POSITION FOR TRANSMISSION.
  - ❖ RX\_BIT\_POS: REGISTER INDICATING THE NEXT BIT POSITION FOR RECEPTION.

THE SHIFT REGISTER MODULE TAKES VARIOUS INPUTS SUCH AS CLOCK SIGNALS, DATA SELECTION, DATA INPUT, AND CONTROL SIGNALS. THE MODULE INCLUDES INTERNAL REGISTERS FOR STORING DATA, BIT POSITIONS FOR DATA SHIFTING, AND A COUNTER FOR CHARACTER BIT TRACKING.

IT CALCULATES THE TRANSFER IN PROGRESS AND THE LAST CHARACTER INDICATOR. IT ALSO CALCULATES THE SERIAL OUTPUT BASED ON THE CLOCK SIGNALS AND THE CURRENT BIT POSITION.

THE MODULE HANDLES THE CALCULATION OF THE BIT POSITIONS FOR BOTH TRANSMISSION AND RECEPTION BASED ON THE CONFIGURATION. THE OUTPUT IS THE SHIFTED DATA FROM THE SHIFT REGISTER. THE MODULE ALSO SUPPORTS DATA LATCHING.

## MISO



8



IT INTERFACES WITH A WISHBONE BUS AND PROVIDES COMMUNICATION WITH A MASTER DEVICE USING THE SPI PROTOCOL.

- INPUTS:
  - ❖ WB\_CLK\_IN: CLOCK INPUT FOR THE WISHBONE BUS.
  - ❖ WB\_RST\_IN: RESET INPUT FOR THE WISHBONE BUS.
  - ❖ WB\_ADR\_IN: ADDRESS INPUT FOR THE WISHBONE BUS.
  - ❖ WB\_DAT\_IN: DATA INPUT FOR THE WISHBONE BUS.
  - ❖ WB\_SEL\_IN: SELECT INPUT FOR THE WISHBONE BUS.
  - ❖ WB\_WE\_IN: WRITE ENABLE INPUT FOR THE WISHBONE BUS.
  - ❖ WB\_STB\_IN: STROBE INPUT FOR THE WISHBONE BUS. ❖ WB\_CYC\_IN: CYCLE INPUT FOR THE WISHBONE BUS.
  - ❖ MISO: MASTER INPUT SLAVE OUTPUT (MISO) SIGNAL
- OUTPUTS:
  - ❖ WB\_DAT\_O: DATA OUTPUT FOR THE WISHBONE BUS.
  - ❖ WB\_ACK\_OUT: ACKNOWLEDGE OUTPUT FOR THE WISHBONE BUS.
  - ❖ WB\_INT\_O: INTERRUPT OUTPUT FOR THE WISHBONE BUS.
  - ❖ SCLK\_OUT: CLOCK OUTPUT FOR THE SPI INTERFACE.
  - ❖ MOSI: MASTER OUTPUT SLAVE INPUT (MOSI) SIGNAL.
  - ❖ SS\_PAD\_O: SLAVE SELECT OUTPUT FOR THE SPI INTERFACE.

THE CODE INSTANTIATES TWO SUB-MODULES: "SPI\_CLGEN" AND "SPI\_SHIFT\_REG".

- "SPI\_CLGEN" GENERATES THE SPI CLOCK SIGNAL BASED ON THE INPUT CLOCK AND CONTROL SIGNALS.
- "SPI\_SHIFT\_REG" HANDLES THE DATA SHIFTING AND SYNCHRONIZATION FOR SPI COMMUNICATION.

IT DECODES ADDRESSES TO READ FROM SPECIFIC REGISTERS AND PROVIDES OUTPUT DATA AND ACKNOWLEDGMENT SIGNALS. IT ALSO SUPPORTS INTERRUPT GENERATION AND SLAVE DEVICE SELECTION. OVERALL, IT SERVES AS A COMPLETE SPI CONTROLLER FOR DATA TRANSFER BETWEEN A MASTER AND MULTIPLE SLAVE DEVICES.

## WISHBONE MASTER MODULE INTERFACE

SPI MASTER ACTS AS A SLAVE TO WISHBONE INTERFACE. WISHBONE MASTER COMMUNICATES WITH THE HOST. BUS SIGNALS OF WISHBONE MASTER ARE AS FOLLOWS:

Port	Width	Direction	Description
wb_clk_i	1	Input	Master clock
wb_rst_i	1	Input	Synchronous reset, active high
wb_adr_i	5	Input	Lower address bits
wb_dat_i	32	Input	Data towards the core
wb_dat_o	32	Output	Data from the core
wb_sel_i	4	Input	Byte select signals
wb_we_i	1	Input	Write enable input
wb_stb_i	1	Input	Strobe signal/Core select input
wb_cyc_i	1	Input	Valid bus cycle input
wb_ack_o	1	Output	Bus cycle acknowledge output
wb_err_o	1	Output	Bus cycle error output
wb_int_o	1	Output	Interrupt signal output

Table 1: Wishbone interface signals

ALL OUTPUT WISHBONE SIGNALS ARE REGISTERED AND DRIVEN ON THE RISING EDGE OF WB\_CLK\_I. ALL INPUT WISHBONE SIGNALS ARE LATCHED ON THE RISING EDGE OF WB\_CLK\_I.

MODULE DECLARATION: DEFINES THE MODULE WISHBONE\_MASTER WITH INPUT AND OUTPUT PORTS.

- INPUTS:
  - ❖ CLK\_IN: THE CLOCK INPUT FOR THE MODULE.
  - ❖ RST\_IN: THE RESET INPUT FOR THE MODULE.
  - ❖ ACK\_IN: THE ACKNOWLEDGMENT INPUT FROM THE SLAVE DEVICE.
  - ❖ ERR\_IN: THE ERROR INPUT FROM THE SLAVE DEVICE.
  - ❖ DAT\_IN: THE DATA INPUT FOR WRITE OPERATIONS.
- OUTPUTS:
  - ❖ ADR\_O: THE ADDRESS OUTPUT FOR THE WISHBONE MASTER.
  - ❖ CYC\_O: THE CYCLE OUTPUT FOR INITIATING A WISHBONE TRANSACTION.
  - ❖ STB\_O: THE STROBE OUTPUT FOR INDICATING THE START OF A WISHBONE TRANSACTION.
  - ❖ WE\_O: THE WRITE ENABLE OUTPUT FOR WRITE OPERATIONS.
  - ❖ DAT\_O: THE DATA OUTPUT FOR READ OPERATIONS.
  - ❖ SEL\_O: THE SELECT OUTPUT FOR SELECTING THE ACTIVE BYTE LANES.

THE MODULE CONTAINS A TASK CALLED “INITIALIZE” FOR INITIALIZING THE INTERNAL SIGNALS OF THE MODULE. IT ALSO CONTAINS A TASK CALLED SINGLE\_WRITE FOR PERFORMING A SINGLE WRITE OPERATION. IT SETS THE INTERNAL SIGNALS ADR\_TEMP, SEL\_TEMP, WE\_TEMP, DAT\_TEMP, CYC\_TEMP, AND STB\_TEMP BASED ON THE PROVIDED

INPUTS. IT WAITS FOR THE ACKNOWLEDGMENT SIGNAL (ACK\_IN) TO GO LOW BEFORE RESETTING THE INTERNAL SIGNALS.

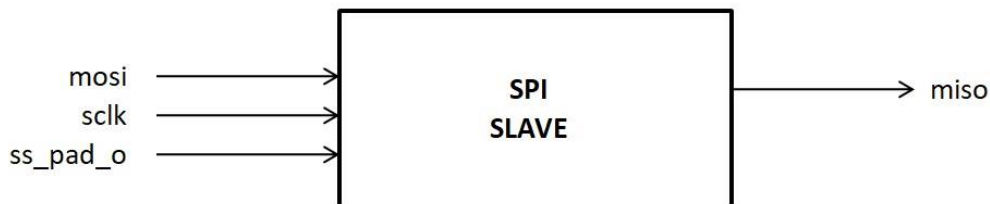
THE MODULE USES SEQUENTIAL BLOCKS (ALWAYS@(POSEDGE CLK\_IN)) TO ASSIGN VALUES TO THE OUTPUT SIGNALS BASED ON THE INTERNAL SIGNALS. THE RST\_IN SIGNAL IS USED TO HANDLE RESET CONDITIONS FOR CYC\_O AND STB\_O.

OVERALL, THE WISHBONE\_MASTER MODULE PROVIDES CONTROL AND DATA SIGNALS FOR INTERACTING WITH A WISHBONE SLAVE DEVICE, ALLOWING FOR READ AND WRITE OPERATIONS.

## SLAVE MODULE INTERFACE

THE MASTER CAN CHOOSE WHICH SLAVE IT WANTS TO TALK TO BY SETTING THE SLAVE'S SS LINE TO A LOW VOLTAGE LEVEL. IN THE IDLE, NON-TRANSMITTING STATE, THE SLAVE SELECT LINE IS KEPT AT A HIGH VOLTAGE LEVEL. MULTIPLE SS PINS MAY BE AVAILABLE ON THE MASTER, WHICH ALLOWS FOR MULTIPLE SLAVES TO BE WIRED IN PARALLEL. IF ONLY ONE SS PIN IS PRESENT, MULTIPLE SLAVES CAN BE WIRED TO THE MASTER BY DAISY-CHAINING.

SPI SLAVE MODULE HERE JUST BEHAVES AS A "SLAVE" TO TEST FOR THE CODE RESULTS. IT IS NOT AN ACTUAL SLAVE.



BUS SIGNALS FOR SPI SLAVE MODULE ARE AS FOLLOWS:

Port	Width	Direction	Description
/ss_pad_o	8	Output	Slave select output signals
sclk_pad_o	1	Output	Serial clock output
mosi_pad_o	1	Output	Master out slave in data signal output
miso_pad_i	1	Input	Master in slave out data signal input

Table 2: SPI external connections

MODULE DECLARATION: DEFINES THE MODULE SPI\_SLAVE WITH INPUT AND OUTPUT PORTS.

- INPUTS:
  - ❖ SCLK: THE SERIAL CLOCK INPUT FOR SYNCHRONIZATION.
  - ❖ MOSI: THE INPUT SIGNAL REPRESENTING THE DATA RECEIVED FROM THE SPI MASTER.
  - ❖ SS\_PAD\_O: A MULTI-BIT OUTPUT SIGNAL REPRESENTING THE CHIP SELECT LINES FOR THE SPI SLAVE.
- OUTPUT:
  - ❖ MISO: THE OUTPUT SIGNAL REPRESENTING THE DATA TO BE TRANSMITTED FROM THE SPI SLAVE.
- INTERNAL SIGNALS:
  - ❖ RX\_SLAVE: A REGISTER INDICATING WHETHER THE SPI SLAVE IS IN RECEIVE MODE.
  - ❖ TX\_SLAVE: A REGISTER INDICATING WHETHER THE SPI SLAVE IS IN TRANSMIT MODE.
  - ❖ TEMP1: A 128-BIT REGISTER USED FOR TEMPORARY STORAGE OF RECEIVED DATA.
  - ❖ TEMP2: A 128-BIT REGISTER USED FOR TEMPORARY STORAGE OF RECEIVED DATA.

ON THE POSEDGE OF SCLK, IF THE CHIP SELECT LINES ARE NOT ALL HIGH ( $SS\_PAD\_O \neq 8'b1111111$ ) AND THE SLAVE IS IN RECEIVE MODE (RX\_SLAVE IS HIGH) AND NOT IN TRANSMIT MODE (TX\_SLAVE IS LOW), THE RECEIVED DATA MOSI IS CONCATENATED WITH THE EXISTING DATA IN TEMP1.

ON THE NEGEDGE OF SCLK, IF THE SLAVE IS IN RECEIVE MODE (RX\_SLAVE IS HIGH) AND NOT IN TRANSMIT MODE (TX\_SLAVE IS LOW), THE VALUE IN TEMP1 IS ASSIGNED TO MISO1.

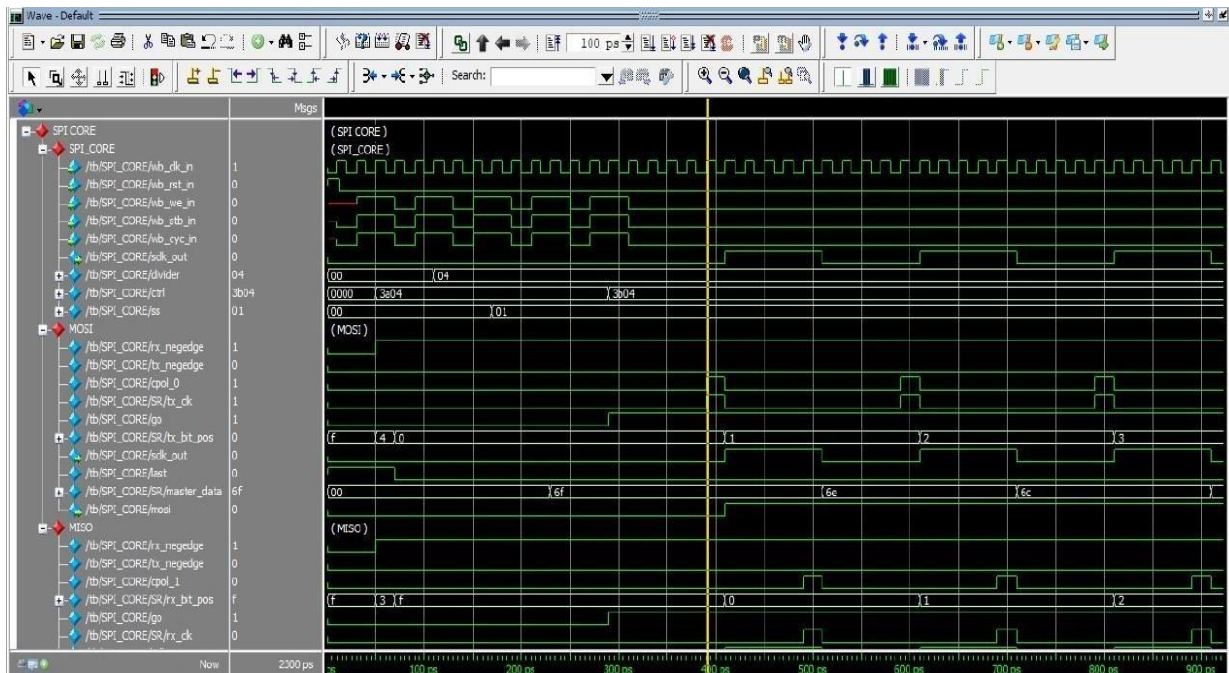
ON THE POSEDGE OF SCLK, IF THE SLAVE IS IN RECEIVE MODE (RX\_SLAVE IS HIGH) AND NOT IN TRANSMIT MODE (TX\_SLAVE IS LOW), THE VALUE IN TEMP2 IS ASSIGNED TO MISO2.

THE MISO SIGNAL IS ASSIGNED AS THE LOGICAL OR OF MISO1 AND MISO2, WHICH REPRESENTS THE DATA TO BE TRANSMITTED FROM THE SPI SLAVE.

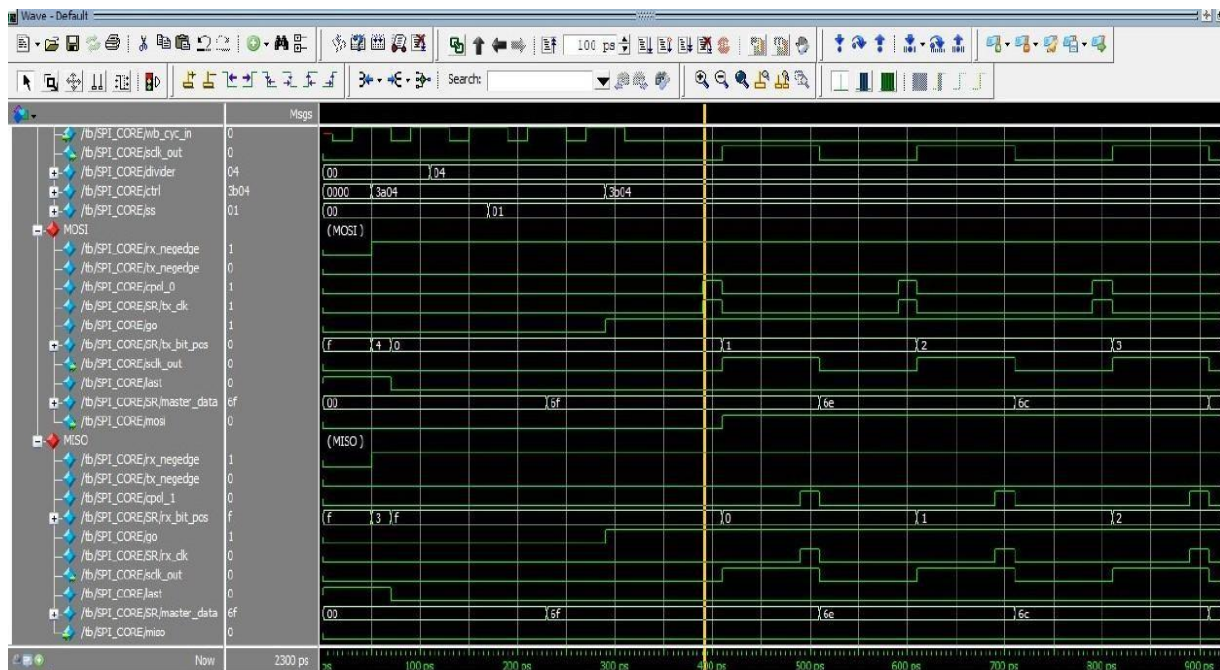
OVERALL, THE SPI\_SLAVE MODULE RECEIVES DATA FROM THE SPI MASTER ON THE MOSI LINE, PROCESSES AND STORES IT IN TEMP1 AND TEMP2, AND PROVIDES THE OUTPUT MISO FOR TRANSMITTING DATA BACK TO THE SPI MASTER. THE CHIP SELECT LINES (SS\_PAD\_O) ARE USED TO CONTROL THE SLAVE OPERATION.

## TEST CASE MODULE WAVEFORMS

A. RX NEGEDGE = 1 AND TX NEGEDGE = 0

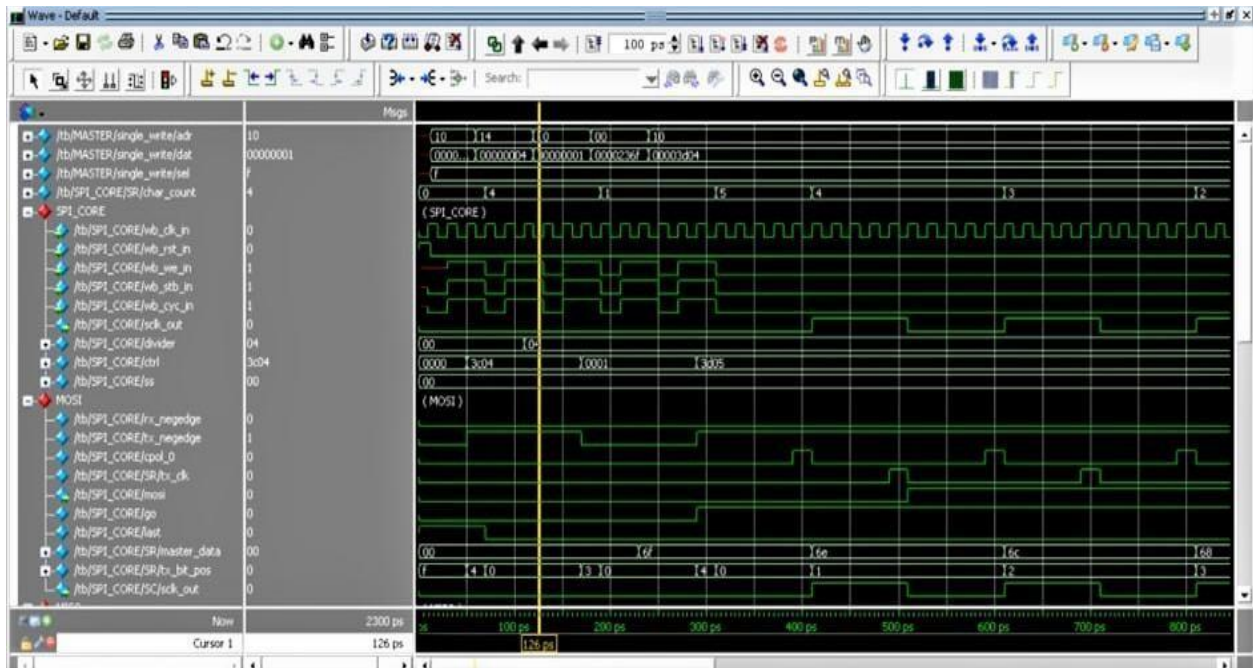


21BEC1085





B. RX NEGEDGE = 0 AND TX NEGEDGE = 1



21BEC1085

