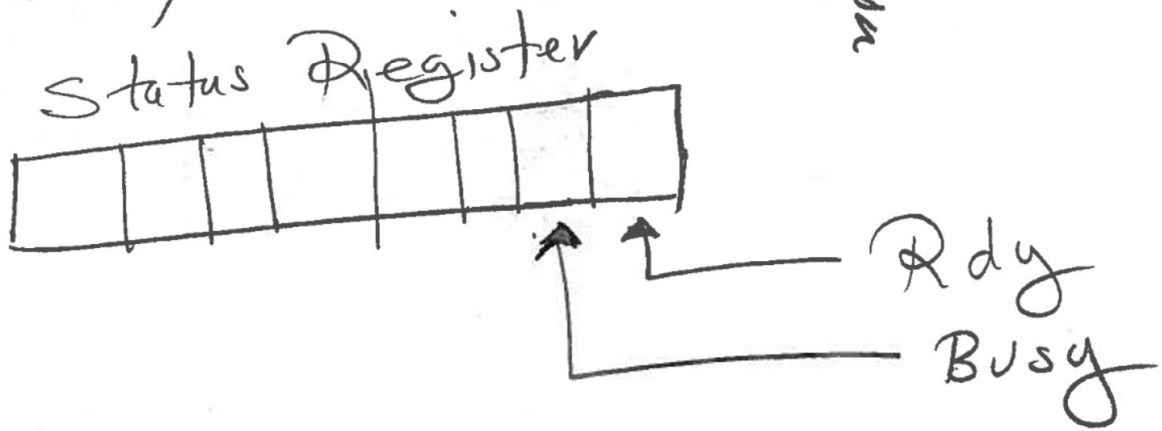
① Comb/Seq Logic
② Digital Design
③ RTL Implementation
④ Simulation.

SPI_slave Interface

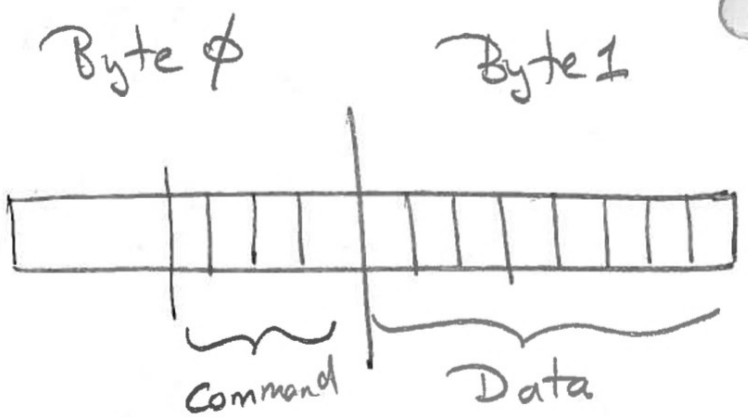
SCK — SPI clock
MISO — SPI Master Input Slave Output
MOSI — SPI Master Output Slave Input
SS — spi Slave Select Active low
EXTDATA [7:0] — Sample Data Input



Notes

① When Sample/Count Data is active, the Busy Bit should be set. And Cleared when Sample Data is complete.

Modes of Operation



① Write Address Register
AR ← SPI

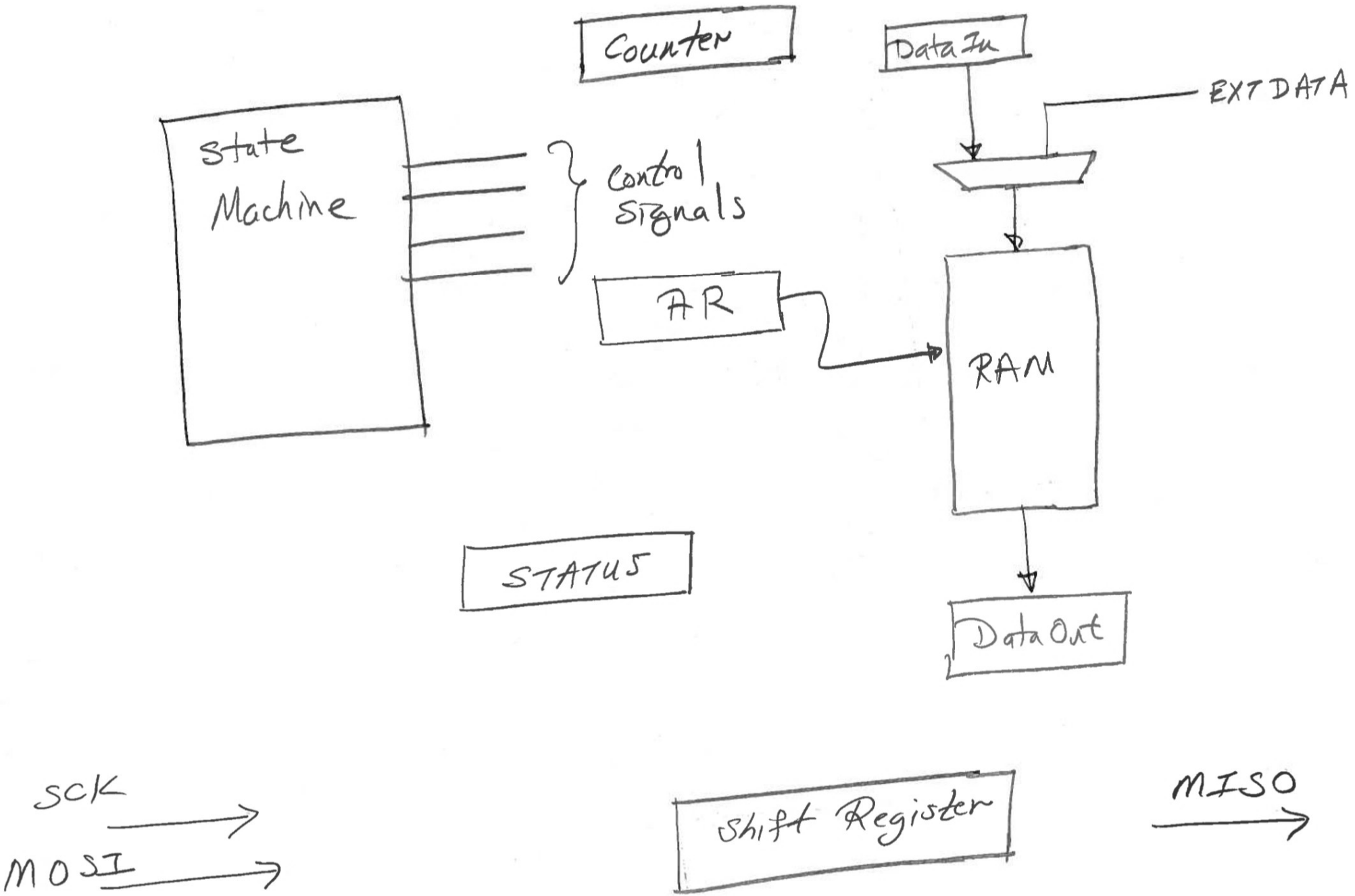
② Write DataIn Register
DataIn ← SPI

③ Read DataOut Register
SPI ← DataOut

④ Sample Data (Count = N bytes)
#) AR ← 0
Busy ← 1
Counter ← Count

→ RAM[AR] ← EXTDATA
Counter ← Counter − 1
AR ← AR + 1
Counter != 0
Busy ← 0

Description	command	Data
Write Address Register AR ← SPI	0001	<Address>
Write DataIn Register DataIn ← SPI	0010	<Data>
Read DataOut Register SPI ← DataOut	0011	<Data>
Sample / Count Counter ← SPI	0100	<Count>
Read Status Register SPI ← Status Register	0101	<STATUS>


Counter
Data In
EXT DATA
State Machine
control signals
AR
RAM
STATUS
Data Out
SCK
MOSI
SS
Shift Register
MISO

Suggestions

1. Develop both SPI_Master and SPI_Slave modules along with a testbench.
2. Generate SCK with the testbench.
3. Each Systemverilog module can have an initial block.
   So, you can add an initial block to the SPI_Master to generate test conditions.
4. Get the shift Register and Simple State Machines designed/coded first.
   Then add the various commands.

Submission Items

HC - Hard Copy
SC - Soft Copy

		HC	SC
(1)	System Verilog Code for SPI_Slave module.	✓	✓
(2)	State Machine Diagram (SPI_Slave)	✓	
(3)	Diagram of SPI_Slave with Control Signals	✓	
(4)	List of which SPI commands work and printout of waveforms showing each command	✓	
(5)			

Item	SPI_Slave.
Design	SPI Interface – Basic Functionality State Machine & Control Signals RAM {Read/Write} Registers {Read/Write} Commands – 0, ②, ③ ④
RTL Coding	State Machine Counter/RAM DataPath
Functionality	SPI Interface Commands 0, ②, ③, ④
Documentation	State Diagram & SPI_Slave digram with Control Signals