Abstract--- This project paper introduce with uniform verification environment using system Verilog as well as gives appropriate direction about reusability of verification environment’s components. This project paper divides with five sections. First section contains overview of SPI interface protocol. Second section is about physical interface for SPI protocol. Third section leads towards verification environment using system Verilog and over view about verification components. Fourth section is about Master VIP using reusability of environment’s components of SPI protocol with simulation results and coverage report using VCS. Fifth section is about Slave VIP using reusability of environment’s components of SPI protocol with simulation results and coverage report using VCS.

I. OVERVIEW OF SPI INTERFACING PROTOCOL

The SPI is essentially a “three-wire plus slave selects” serial bus for eight or sixteen bit data transfer applications. The three wires carry information between devices connected to the bus. Each device on the bus acts simultaneously as a transmitter and receiver. Two of the three lines transfer data (one line for each direction) and the third is a serial clock. Some devices may be only transmitters while others only receivers. Generally, a device that transmits usually possesses the capability to receive data also. An SPI display is an example of a receive-only device while EEPROM is a receiver and transmit device. The devices connected to the SPI bus may be classified as Master or Slave devices. A master device initiates an information transfer on the bus and generates clock and control signals. A slave device is controlled by the master through a slave select (chip enable) line and is active only when selected. Generally, a dedicated select line is required for each slave device. The same device can possess the functionality of a master and a slave but at any point of time, only one master can control the bus in a multi-master mode configuration. Any slave device that is not selected must release (make it high impedance) the slave output line. The SPI bus employs a simple shift register data transfer scheme: Data is clocked out of and into the active devices in a first-in, first-out fashion. It is in this manner that SPI devices transmit and receive in full duplex mode. All lines on the SPI bus are unidirectional: The signal on the clock line (SCLK) is generated by the master and is primarily used to synchronize data transfer. The master-out, slave-in (MOSI) line carries data from the master to the slave and the master-in, slave-out (MISO) line carries data from the slave to the master. Each slave device is selected by the master via individual select lines. Information on the SPI bus can be transferred at a rate of near zero bits per second to 1 Mbits per second. Data transfer is usually performed in eight/sixteen bit blocks. All data transfer is synchronized by the serial clock (SCLK). One bit of data is transferred for each clock cycle. Four clock modes are defined for the SPI bus by the value of the clock polarity and the clock phase bits. The clock polarity determines the level of the clock idle state and the clock phase determines which clock edge places new data on the bus. Any hardware device capable of operation in more than one mode will have some method of selecting the value of these bits.

The SPI bus specifies four logic signals:
- SPI_CLK: Serial Clock.
- SPI_MOSI: Master Output, Slave Input.
- SPI_MISO: Master Input, Slave Output.
- SPI_SS: Slave Select.

Fig. 1: Spi Interface

II. PHYSICAL INTERFACE OF SPI PROTOCOL

Before developing of SPI VIP we need to analyse actual SPI communication between to ICs. In this paper implemented actual data transmission according to SPI protocol employs two PSoC kits. One of them is working as a master and another one is working as a slave.

Fig. 2: physical interpretation of SPI protocol

The figure shows there are two PSoc kits connected with each other and transfer data in full duplex mode. We can analyse this data transfer using DSO which is shown in figure 2.

Fig. 3 Starting of Transmission
Fig. 4 End of Transmission

Fig. 5 Simulation Result of SPI Protocol

According to these simulation results we can say that this data transfer is employed SPI communication protocol. Physical implementation of SPI data transmission protocol is useful for understanding of Data transmission fashion in SPI protocol Overview about Verification Environment using System Verilog

Verification is technique which is verify DUT and find out bugs in that DUT if there. DUT is connected with components of verification environment and using these components entire DUT can verify.

III. MASTER VIP

Master VIP contains Slave DUT and master verification environment. This VIP designed for single master and single slave. But this is a reusable verification environment so this environment is also employable for multi master multi slave design. There is no need to do change in environment’s components.

A. Report of Master VIP

Fig. 8: Compilation of Master VIP

Fig. 9: Assertion Which Checks All Pins Works Correctly

monitor. Monitor component is monitor data using interface. This data is transferred by monitor to scoreboard. In earlier generator puts data in to scoreboard and now monitor puts data in to scoreboard. There are two data available with Scoreboard component. Scoreboard compares these two data. If both data matched we can say that DUT is working with proper functionality.
IV. SLAVE VIP

SLAVE VIP of SPI contains master DUT and Slave environment. This environment is used to verify master DUT. If we got desirable output in scoreboard class or we can say if packets matched in to scoreboard than functionality of master DUT is up to date.

A. Report of slave VIP

Fig. 10: Matching of Packets

B. Simulation result of master VIP

Fig. 11: simulation Result of Master VIP

C. Functional coverage report of master VIP

Fig. 12: Coverage Report of Master VIP

D. Code coverage report of master VIP

Fig. 13: code coverage of Master VIP

Fig. 14: Compilation of Slave VIP

Fig. 15: Packet Matching

B. Simulation result of slave VIP

Fig. 16: Simulation Result of Slave VIP

C. Coverage report of slave VIP

Fig. 17 Coverage Report of Slave VIP
D. Code coverage of slave VIP

![Image of code coverage for slave VIP](image)

Fig. 18: Code Coverage of slave VIP

V. CONCLUSIONS

In the project, the verification environment of SPI Interface was developed using System Verilog. The idea was to achieve the properties like re-usability and robustness. The results shown here are limited to only one master and slave but if used for multiple slaves, the components are designed such that they can be reused easily which can be checked with the architectural diagram of the environment. We have designed parameterized components so that they can be configured as per requirement. Analysis of results shows that reusability can be achieved using the flow described in the report.

REFERENCES


