Design and verification of SPI bus IP core

Dawei Wang¹, Jiang Yan²* and Ying Qiao³

¹ School of Information Science and Technology, North China University of Technology, Beijing, 100144, China
² School of Information Science and Technology, North China University of Technology, Beijing, 100144, China
³ Digital IC-Technology Dept, Tsinghuaic, Beijing, 100084, China

the first author’s e-mail: 2019311030102@mail.ncut.edu.cn
*Corresponding author’s e-mail: yanjiang@ncut.edu.cn

Abstract. With the development of chip design technology, the complexity of the chip is getting bigger and bigger, and various IP (Intellectual Property) is integrated to form SoC (System on Chip). In order to improve the efficiency of designing SoC, IP reuse technology is often used, so the development of IP has a wide range of significance and use value. The SPI bus is a bus protocol often used in chip communication and has many application scenarios. Therefore, this research mainly develops the SPI bus IP. During the development process, the SPI bus design is carried out, then based on the UVM verification methodology the designed IP has been functionally verified, and finally, a fully functional SPI bus design is obtained.

1. Introduction
SoC products are widely used in all aspects of life, such as consumer electronics, computers, communications and other fields, bringing great convenience to scientific and technological life. And SoC integrates a variety of IP, and the development of SoC products depends on the development of IP technology. Applying the developed IP can improve the efficiency of developing chips, and authorizing and selling the developed IP can also bring in revenue, so many companies directly reuse the IP of other companies, and some companies realize huge profits by developing IP. The development of IP needs to follow certain specifications and the bus protocol, which is more conducive to the reuse of IP. The AMBA bus has a large number of applications on system chips based on the ARM processor core. In addition, the ARM processor has an absolute market position in the embedded field. Therefore, the AMBA bus protocol has been widely supported at home and abroad and is the mainstream on-chip bus architecture used in the industry. Therefore, the development of AMBA bus communication protocols has important prospective significance and use value. In the AMBA bus architecture, SPI is a commonly used communication protocol, which has the advantages of simplicity and ease of use, resource saving, reliability and stability, and wide application range. Therefore, this research has developed the SPI bus, and carried out front-end design and functional verification on it to ensure it can be used for IP reuse.

2. SPI bus introduction
SPI is a synchronous, high-speed serial communication bus invented by Motorola. It is mainly used for communication between CPU control devices and various peripheral devices. There are 4 signal
lines for SPI, which are SCLK (Serial Clock), MOSI (Master Output and Slave Input), MISO (Master Input and Slave Output), and SS (Slave Selection) [1]. Only 4 communication lines are used, so it saves the number of pins of the chip and saves layout space. The communication protocol stipulates two working modes: master mode and slave mode. The device working in master mode is responsible for outputting the synchronous serial clock SCLK and initializing data transmission.

SPI protocol stipulates 4 kinds of transmission timings, which are distinguished by different values of clock polarity CPOL and clock phase CPHA[2]. We can see these 4 kinds of transmission timings from figure1-4. CPOL=0 indicates that the clock idle state is low, and CPOL=1 indicates that the clock idle state is high; CPHA=0 indicates that the data is sampled on the first edge of SCLK, and CPHA=1 indicates that the data is sampled on the second edge of SCLK[3]. Each transmission sequence requires 1 SCLK cycle to send and receive one bit of data, and several 8-bit data can be transmitted each time.

![Figure 1. SPI timing when CPOL=0 and CPHA=0](image1)

![Figure 2. SPI timing when CPOL=0 and CPHA=1](image2)

![Figure 3. SPI timing when CPOL=1 and CPHA=0](image3)
3. SPI bus design

SPI_TOP is the top-level module of the entire design. Figure 5 is the top-level module of the SPI bus. As you can see from figure 5, under the control of the register, SPI_TRNSNSMITER and SPI_RECEIVER receive instructions from the bus, operate on the data, and finally output from the SPI port signal.

The SPI_APBIF module converts the signal from the APB bus into a signal for register control to read and write, and complete the operation of reading the APB bus, which is shown in figure 6.

The SPI_REGS module receives the control signal and SPI operation signal from APB_IF, and outputs the data written in the SPI register, which is shown in figure 7.
The SPI_CONTROLLER module converts the SPI control register data into a control signal, and converts the output signal of other modules into a control signal written into the register.

As is shown in figure 8, the SPI_TRANSMITER module is a module for SPI to send data. It generates clocks for the sending buffer and the receiving buffer, and converts the data to be sent into serial data according to the protocol. It calculates the CRC check value during data transmission, and send the CRC check value through the sending buffer.

The SPI_MSTSCLKGEN module can generate the SCLK signal in SPI. By configuring the BR bit in CR1 for frequency division, the SCLK clock of the corresponding baud rate can be generated.

The SPI_TRANSMITER_SHIFTER module is the shift register of the sending module, which can serially output the sending data.

The SPI_CRC16A8 module completes the calculation of the CRC according to the polynomial.

We can see from figure 9, the SPI_RECEIVER module is the data receiving module of SPI, which can receive the serial data of the data line.

The SPI_RECEIVER_SHIFTER module is the shift register of the receiving module, which receives the serial data of the data line.

The SPI_IOMUX module can convert external interface signals into internal signals, and internal control signals into external interface signals.

The SPI_TRQGEN module can generate various interrupts related to SPI.

4. SPI bus verification

4.1. SPI bus verification plan

According to the functions of the SPI bus, detailed verification plans are listed. The first is the verification of the system clock and reset. The second is the access check of the registers, including
reset value check, read-write check, and the working status check of the read-only register. The last is the verification of the SPI functional characteristics: the configuration process of master-slave mode, 4 modes of data transmission in master and slave mode (full-duplex mode, unidirectional receive-only mode, bidirectional mode when transmitting, bidirectional mode when receiving), 4 transmission timings composed of clock polarity and clock phase, and 8 baud rate clocks frequency division, 3 status flags (BUSY, TXE, RXNE), 3 error flags (MODF, OVR, CRCERR), 5 interrupt conditions (TXE, RXNE, MODF, OVR, CRCERR), and CRC check calculation when sending and receiving data in master and slave mode.

4.2. SPI bus verification platform
In this study, a verification platform for the SPI bus was built based on the UVM verification platform architecture. The verification platforms based on UVM generally instantiate DUT and testcase to complete the connection between DUT and testcase[4]. The same verification environment is maintained in different test cases, but different configurations and input stimuli will be used in different test cases to cover different test scenarios. Therefore, during the simulation regression, the simulator will dynamically instantiate test case, and only compile the static verification environment once. As you can see from figure 10, the UVM verification architecture is layered. At the bottom of the environment, there are scoreboard, reference model, adapter, agent, etc., and at the bottom of the agent, there are sequencer, driver, and monitor. The agent bundles these components that process the same transaction together, reflecting the encapsulation characteristics of object-oriented languages[5]. The sequencer controls the sending of input stimuli or the sending of the sequence. The adapter is responsible for the conversion between the register signal and the bus signal. The driver and the monitor do not produce content, but are responsible for the conversion of the content. The driver can convert transaction-level stimuli that are easy for engineers to control into signal-level stimuli that can be recognized by the DUT, while the monitor is the opposite, which can convert the DUT signal level output into transaction-level data which is easy to understand. The output of the reference model is the ideal value of the data. What the monitor receives is the actual value of the data, and finally, both of these 2 values are passed to the scoreboard for comparison. The APB master agent is the input agent, which receives the signal from the APB bus, and is generally the master device. SPI agent is the output agent which is matched with DUT, one is the master device and the other is the slave device.

![SPI bus verification platform](image)

4.3. Verification results
The research uses the built verification platform to verify the SPI design module according to the verification plan. Through continuous improvement of the design code and verification code, the final coverage result obtained after a certain number of regression tests is shown in figure 11 below. It can
be seen from figure 11 that the coverage rate of each module in the SPI bus has reached a high level, indicating that the verification function points are relatively complete.

| Name          | Score | Line | Module | FSM | Condition | Branch | Overall |
|---------------|-------|------|--------|-----|-----------|--------|---------|
| SPI Controller| 98.00%| 99.90%| 98.67% | 98.67% | 97.61% | 96.21% | 100.00% |

Figure 11. SPI bus verification coverage results

5. Conclusion
IP reuse is a popular technology in SoC design, and SPI bus is also a popular IP. This research has carried out front-end design on SPI bus IP core. The designed IP has advantages of complete functions, flexible use, and has engineering value as an IP core for chip design. At the same time, functional verification of the design IP was performed according to UVM. The coverage rate after verification met the requirements, and the verification environment can also be reused, finally achieving a complete SPI bus IP with high reusability.

References
[1] A. N. G. Joseph S. S. Oommen and R. Dhanabal. 2014."Design and implementation of a high speed Serial Peripheral Interface" 2014 International Conference on Advances in Electrical Engineering (ICAEE) pp. 1-3.
[2] H. H. Lu. 2020. "Design and realization of SPI driver based on VxWorks" 2020 IEEE 4th Information Technology Networking Electronic and Automation Control Conference (ITNEC) pp. 200-205.
[3] ST company. 2021. RM0008 Reference manual. http://www.st.com/stonline/products/literature/ds/13587.pdf
[4] Q Zhang, (2014) UVM Combat. Machinery Industry Press, Beijing
[5] B Liu, (2018) A Walking Guide to SoC Verification, Publishing House of Electronics Industry, Beijing