Design and Simulation of IIC based on FPGA

Weiting Xu *

College of Communication Engineering, Jilin University, Changchun, China

* Corresponding Author Email: xuwt2020@jlu.edu

Abstract. The Inter-Integrated Circuit (IIC) bus has been widely used in the industry these years. Because it is very simple and bidirectional, IIC has been widely developed in the field of communication engineering and electronic engineering. It is simple and bidirectional. Developed by Philips, IIC just requires two wires to transfer information. The Field Programmable Gate Array (FPGA) is the product of further development of programmable devices. IIC work out the shortcomings of custom circuits and the problem of the restricted number of gates of the primitive programmable devices. This paper introduces the knowledge of FPGA, IIC bus system, and simulation software, and rewrites the IIC bus using FPGA, and finally successfully derives the simulation waveform to enhance the promotion of IIC technology. There is no doubt that the research of this paper further strengthens the development of IIC bus technology, promotes the application of IIC bus technology and promotes the progress of communication industry.

Keywords: FPGA, IIC, Stimulation.

1. Introduction

In this paper, we corrected the above problem and redesigned the IIC using FPGA, and finally obtained the simulation experimental results. Three literatures are needed, the supplementary literature section. Dvijen designs a protocol conversion unit that enables seamless communication between Serial Peripheral Interface (SPI) and IIC protocols, thus making better use of full duplex communication, more pin-saving SPI protocol, and ultimately to support a large number of peripheral devices [1]. Vishakh and Khwaja design an attendant program using IIC protocol automated system, which saves labor cost by using smart device mobile text, which is then transmitted to the master node in the kitchen by IIC protocol [2]. Chhikara etc. take advantage of the flexibility of IIC to achieve protocol compatibility between IIC and Advanced Peripheral Bus (APB), and can transfer data from IIC-supported modules to APB-supported modules, forming a bidirectional excuse between IIC and APB-supported modules [3].

This paper innovatively uses FPGA to redesign the IIC. This paper first introduces the theoretical basis of FPGA and I2C, and gives a detailed introduction to its background and internal working principle. Immediately after the code simulation experiments, the research in this paper will help the further promotion of IIC technology.

2. Basic theory analysis

2.1. Basic Knowlagent of FPGA

The full name of FPGA is Field-Programmable Gate Array.

The essence of FPGA is to design a chip. Its development process is compiled, integrated, laid out and wired by EDA tools through Verilog and other hardware description languages, and finally loaded into the FPGA device to complete the realized functions [4]. The hardware description language describes the combinatorial logic and sequential logic circuit, the degree and logic are the circuit composed of and, not, or, and the sequential circuit is the trigger. In FPGA, combinational logic becomes the work of lookup table, so all digital circuits are transformed into lookup table and register, which is the basis of FPGA, lookup table is responsible for logic implementation, register store circuit state. The fact that it is programmable means that its logic circuitry can be changed, which is so

erasable, and it is field programmable signify that it is not subject to harsh erasure conditions [5]. The flexibility and high speed of FPGAs make it uniquely suited for digital signal processing applications.

2.2. Basic Knowlagent of IIC

IIC is a very common bus protocol. IIC uses two lines to communicate data between the master controller and the slaves. One is Serial Time Line (SCL) and the other is Serial Data Line (SDA). These two data lines need pull-up resistors, and SCL and SDA are at high level when the bus is idle [6].

IIC supports multiple slaves, that is, an IIC controller can hang multiple IIC slave devices, these different IIC slave devices have different device addresses, so that the IIC master controller can access the specified IIC device through the device address of the IIC device [7]. The IIC bus has many advantages, such as low cost, simple hardware and software implementation, flexibility and so on. Because it is a two-way communication protocol, it can realize multi-master device control. In addition, because it is an open standard, many chips have implemented the IIC bus interface so that they can easily communicate with other chips. The structure of IIC is shown in figure 1.

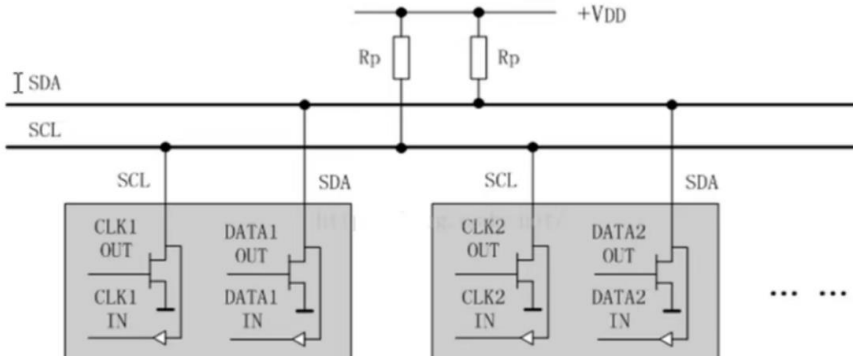


Fig 1. The schematic diagram of IIC

The IIC bus uses two signal lines: the Serial data line (SDA) and the serial clock line (SCL). During transmission, there is also an optional, low-level active reply signal (ACK). Specifically, the three signals in the transmission process are as follows: Serial Data Line (SDA): Used to transmit data in both directions. Serial Clock Line (SCL): Used for data synchronization and transmission in one direction. ACK (Acknowledge): optional to acknowledge whether data is transmitted successfully [8].

Before data is transferred, the master device sends a Start Signal, which indicates that data transfer on the IIC bus is about to begin. The master sends a low-level signal to the SDA line and then a clock signal to the SCL line to initiate data transmission. Next, the data to be sent by the master device is written bit by bit to the SDA line. The transmission of each data bit requires a clock pulse to be sent on the SCL line. When the data transfer is complete, the master device sends a Stop Signal to the SDA line, indicating that the data transfer on the IIC bus has ended.

During data transmission, the slave device may need to send an ACK to the master device. If the SDA line remains high and no reply is received after the master device has sent a byte, the master device considers the data transfer to have failed and stops the transmission.

2.3. Basic Knowlagent of ModelSim

ModelSim is a kind of simulation software for digital circuit design. Developed by Mentor Graphics, it is designed for FPGA design provides efficient simulation and verification tools [9].

ModelSim has full Verilog and VHDL support, and provides a variety of simulation models and debugging tools, including single-step execution, signal tracing, and waveform analysis. With ModelSim, users can create and edit circuit design files, simulate, and verify circuits, and generate waveform diagrams for analysis and debugging. In addition, ModelSim supports automated testing and code coverage analysis, which can help users improve the reliability and performance of circuit

design [10]. Because of its powerful function, easy to use and a wide range of applications, ModelSim has been widely used in digital circuit design.

The workflow is as follows: First, Set the work path. Second, create a project. Then, Write Verilog file and Testbench simulation file. After that, Compile (full compile and separate compile). Then Start the simulator and load the design top layer. After that, Run the simulation. Finally, after we execute the simulation, the ModelSim software will generate waveforms based on our design file and simulation file

3. Design of IIC BASED ON FPGA

3.1. Design process

IIC is a serial, half-duplex bus, mainly used for "close, low-speed" wired data transmission between chips. IIC has two buses: SCL clock bus and SDA data transceiver bus. The roles of the two buses are: SCL is responsible for clock synchronization between the two communication chips, and SDA is used for data transmission and reception between the chips. The IIC structure is simple, so it is not suitable for long-distance transmission, but the simple structure corresponds to the increased complexity of the protocol.

The IIC has only one data transceiver line, but it can still communicate with both sides. For one SDA data line, we can't transmit data between any one master and any one slave at the same time, but we follow the "IIC bus protocol" until the devices on the IIC bus, whether they are slaves or master devices.

We can achieve this through the IIC bus protocol: "At any given moment, we have only one master and one slave on the bus, and the transmission properties (transmission direction, i.e. who is the receiver and who is the sender of the data) between the master and the slave have been determined. The flow path is shown in figure 2.

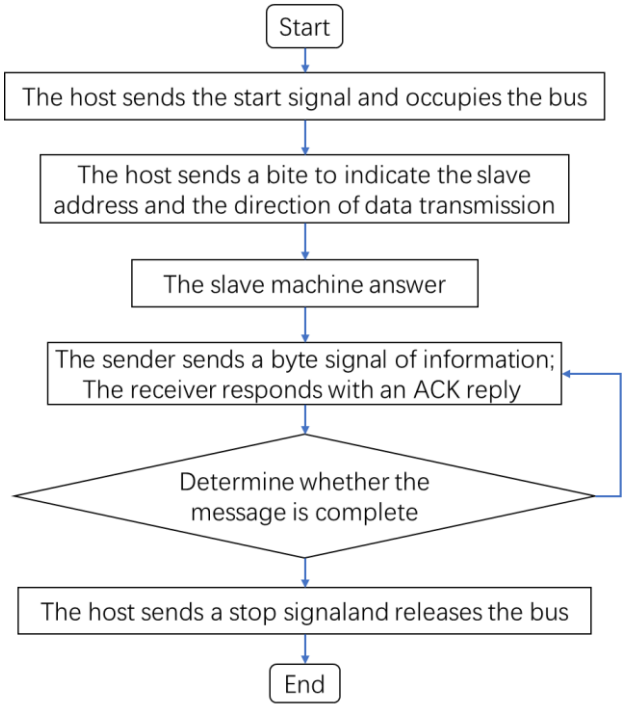


Fig 2. The flow path of IIC

(1) The master sends the start signal to enable the bus, which is already occupied by the bus, because all the masters and slaves are connected to the same SCL clock line and the same SDA data line, so all the masters except the master "shut up".

(2) The master sends a byte of data which means the "address for slave" and the "direction for transfer".

(3) The addressed slave responds to the host with an answer signal (ACK), after which the data transfer begins.

(4) The transmitter sends one byte of data (remember: our IIC bus can only send information in whole bytes).

(5) the receiver sends one byte of data and the receiver responds to the sender with an answer signal (ACK).

(6) Repeat steps (4), (5).

(7) After the communication is completed, the host sends a stop signal and releases the bus.

3.2. Simulation

In order to further verify the rationality and effectiveness of the proposed scheme, this paper simulates the IIC based on FPGA by ModelSim software, and the results are shown in the figure below.

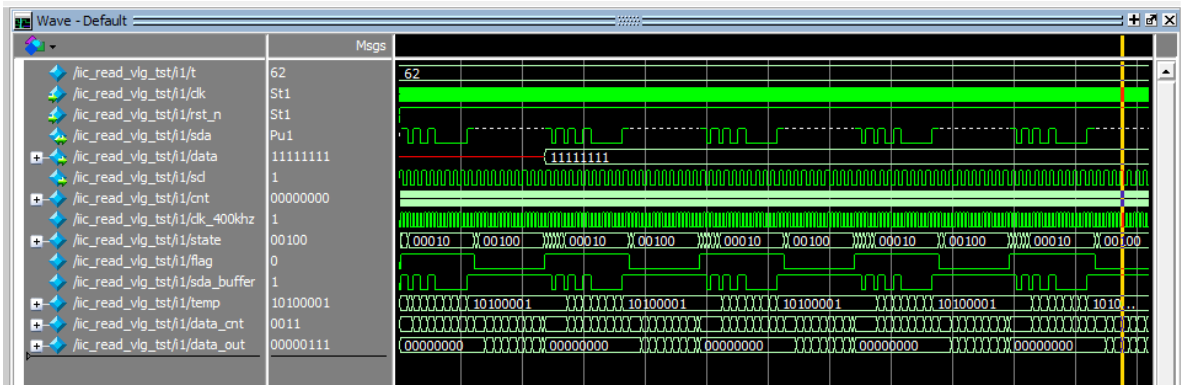


Fig 3. The simulation result of IIC read

The simulation result of IIC read is shown in figure 3. First, send the start bit. Then, send the address of the slave device and the read/write select bits. After that, release the bus and wait until the Electrically Erasable Programmable Read Only Memory (EEPROM) pulls the bus low to answer. If the EEPROM receives successfully, it will answer; if there is no successful handshake or the EEPROM does not generate an answer if the data sent is wrong, it will ask to resend or terminate. Then, send the address of the internal register to be written. After that, the EEPROM replies to it. Then, send data. At last, send stop bit. The simulation results show that the write data function runs smoothly and conforms to the principle of IIC bus.

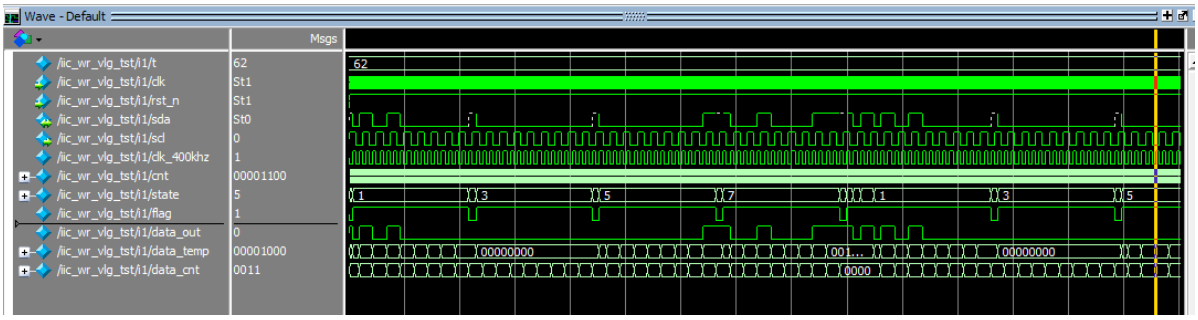


Fig 4. The simulation result of IIC write

The simulation result of IIC write is shown in figure 4. At first, send start bit. Then, send slave address + write bit set. After that, send internal register address. Then, resending the start bit, i.e. restart. Then, resend slave address + read bit set. After that, reading data. The host receiver does not send an ACK signal even after receiving the last byte. Thus, the slave transmitter releases the SDA

line to allow the host to send the P signal to end the transmission. At last, ending stop bits. As shown in the figure, the IIC read function works normally and data is read successfully

4. Conclusion

In this paper, the following conclusions were obtained from the above study: 1. Understanding the use of FPGA and the knowledge related to the IIC bus, and using FPGA to implement IIC read and write, and finally using the simulator to derive simulation results. 2. As shown in the figure, the simulation results of the experiment are finally derived and the feasibility of the code is proved. From the results, there is no doubt that the research in this paper will contribute to the further.

References

[1] D. Trivedi, A. Khade, K. Jain, and R. Jadhav, "SPI to I2C Protocol Conversion Using Verilog," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Aug. 2018.

[2] B. V. Vishakh, M. K. Khwaja, and C. M. Vidhyapathi, "Comprehensive automated device for hotel management using I2C protocol," 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Dec. 2015.

[3] J. Chhikara, R. Sinha, and S. Kaila, "Implementing Communication Bridge between I2C and APB," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Feb. 2015.

[4] L. Xiang, P. Yang, T. Wu, and M. Zhou, "Ultra compact pulse shrinking TDC on FPGA," Measurement, vol. 203, p. 111874, Nov. 2022.

[5] Z. Zhang, A. Xu, H. T. Ren, G. Liu, and X. Cheng, "Reconfigurable multivalued memristor FPGA model for digital recognition," International Journal of Circuit Theory and Applications, vol. 50, no. 11, pp. 3846–3860, Jul. 2022.

[6] D.-C. Mo, J.-F. Huang, P.-H. Luo, L. Chen, and S.-X. Huang, "Distant metastasis-free survival with adjuvant pembrolizumab for resected stage IIB or IIC melanoma," The Lancet Oncology, vol. 24, no. 1, p. e7, Jan. 2023.

[7] G. V. Long et al., "Pembrolizumab versus placebo as adjuvant therapy in resected stage IIB or IIC melanoma (KEYNOTE-716): distant metastasis-free survival results of a multicentre, double-blind, randomised, phase 3 trial," The Lancet Oncology, vol. 23, no. 11, pp. 1378–1388, Nov. 2022.

[8] E. Hindié, "Adjuvant therapy in stage IIB and IIC melanoma: is sentinel biopsy needed?," The Lancet, vol. 400, no. 10352, p. 559, Aug. 2022.

[9] Y.-S. Kung, J.-M. Lin, Y.-J. Chen, and H.-H. Chou, "ModelSim/Simulink Cosimulation and FPGA Realization of a Multiaxis Motion Controller," Mathematical Problems in Engineering, vol. 2015, pp. 1–17, 2015.

[10] S. Das, J.-F. Li, A. Nayak, M. Assaf, S. Biswas, and E. Petriu, "Circuit Architecture Test Verification Based on Hardware Software Co-design with ModelSim," IETE Journal of Research, vol. 59, no. 2, p. 132, 2013.