Proceedings of CONF-MSS 2026 Symposium: Mechanical Control and Automation
DOI: 10.54254/2755-2721/2026.BJ31080

FPGA Implementation of Low Power IIC Controller with EEPROM and Multi-Protocol Performance Analysis

Junnan Qiu

School of Design and Architecture, Zhejiang University of Technology, Hangzhou, China
202201050314@zjut.edu.cn

Abstract. As the embedded system in intelligent devices is progressing, hardware implementation of low-power communication protocols has become a significant research direction in the field of electronic and information engineering. This study presents an overall design technique for a low-power inter-integrated circuit (IIC) protocol master/slave device controller in Verilog hardware description language (HDL) and achieves full read-write operation in FPGA board-level simulation setting, electrically erasable programmable read-only memory (EEPROM) as the verification device. This study employs a finite state machine (FSM) technique to model the hardware implementation of IIC communication process, followed by the integration of a clock gating strategy to mitigate dynamic power consumption. Vivado is utilized for synthesis, power analysis, and hardware implementation on a field-programmable gate array (FPGA) board. The results indicate that the optimized IIC controller exhibits relatively low energy consumption. Simultaneously, to assess the performance level of the proposed IIC interface, the study constructs Serial Peripheral Interface (SPI) and Universal Asynchronous Receiver/Transmitter (UART) protocols, synthesizes, implements them with ModelSim and Vivado, and compares them with IIC protocol in resource usage, power consumption and structure. Experiments show that IIC offers the lowest power consumption which is 0.093W and excellent scalability, SPI achieves the highest data transfer rate which can be over 10 Mb/s, while UART has a simple structure and moderate resource usage. This provides a clear and feasible path for the hardware implementation of an appropriate and low-power communication interface in an embedded system.

Keywords: Verilog HDL, FPGA, IIC, UART, SPI

1. Introduction

Inter-Integrated Circuit (IIC) protocol is a widely-used interface between microcontroller (such as the Field Programmable Gate Arrays, FPGA), analog-digital converter (ADC), temperature sensor, electrically erasable programmable read-only memory (EEPROM), camera configuration, etc [1]. IIC has a wide range of electronic applications with chip-to-chip bidirectional serial communication bus because of the low consumption of I/O (In/Out) resources and its convenience in connecting multiple IIC devices to a single bus. Serial communication protocol is a topic of crucial importance in the realms of embedded systems and electronic applications within intelligent devices. Besides

DOI: 10.54254/2755-2721/2026.BJ31080

IIC protocol, Serial Peripheral Interface (SPI) and Universal Asynchronous Receiver/Transmitter (UART) are also one of the common serial communication interfaces between chips, devices, microcontrollers/microprocessors, etc. FPGA is a digital integrated circuit often considered as a development board which is equipped with a relatively large number of on-chip electronic units and can easily accomplish and validate real-time data updates through inputting bitstreams into a chip or unit [2]. FPGA is also a relatively economically accessible chip board for designing integrated circuit (IC) and prototyping chip [2]. However, FPGA requires users and engineers to set up a hardware description language (HDL) framework that conforms to the transfer principles of digital signal communication chips/devices, such as EEPROM and ADC. In this study, the establishment and implementation of IIC communication bus between EEPROM and FPGA is presented based on Verilog HDL. For the power consumption optimization section, finite state machine (FSM) technique will be used to optimize the structure [3]. Adding clock gating to communication bus will be demonstrated. Furthermore, the structure of SPI and IIC compared with UART will be investigated in order to obtain deeper insights into resource usage, communication efficiency, reliability and power consumption within the three communication protocols, evaluated and validated by Modelsim, Xilinx Vivado and Zynq-7000 All Programmable System on Chip (APSoC).

2. Inter-Integrated Circuit (IIC) protocol

Fig. 1 illustrates the universal structure of IIC master-slave communication bus, taking one master device connecting to multiple slave devices as the example. It is consisted of a serial clock line (SCL) and a serial data line (SDA) both linked with multiple master and slave devices. Slave devices can be some integrated circuit (IC) products that support the IIC protocol, such as microcontrollers, flash memory, EEPROM, analog-to-digital converter (ADC), digital-to-analog converter (DAC), real-time clock circuit (RTC), temperature sensor, accelerometer, etc. The SCL is unidirectional and generated by master device (typically a microcontroller), while the SDA is bidirectional. Both the SCL and SDA lines are implemented as open-drain outputs operating in push-pull logic configuration, so pull-up resistors must be connected between these two signal lines and the VDD power supply. IIC protocol allows more than one masters to be connected to the bus at the same time and has a bus arbitration mechanism [4]. All kinds of slaves are connected in parallel on the IIC bus and identified through device addresses (specific addresses can be found in their own device manual). The data transmission rate can reach 100 Kbps in standard mode, 400 Kbps in fast mode, and 3.4 Mbps in high-speed mode [5].

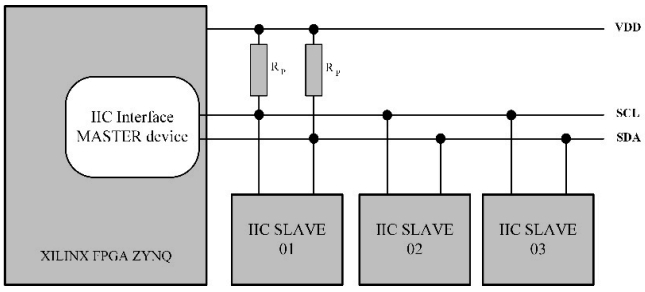

Figure 1. The universal structure of IIC master-slave communication bus

Data transmission on the bus is a series of bytes (8 bits). Generally, there is an acknowledge bit after every 8-bit data from transmitter side to receiver side regardless of masters or slaves. Fig. 2 illustrates the principle of IIC for a master device transmitting data to slave devices. The start condition generated by master is SDA changes to active low from high voltage level when SCL

DOI: 10.54254/2755-2721/2026.BJ31080

stays on high voltage level. The stop condition also generated by master is defined as the transition of SDA from active low to active high while SCL remains at high voltage level. This essentially indicates that during the data transmission process, the SDA must remain stable while SCL is in the period of high voltage level. Otherwise, a start or stop signal will be generated. Furthermore, any transition on the SDA is only permitted when SCL is at low level. The acknowledge bit from a slave device after one byte data must be low, that is “0”. In the case that a master device reads data from a slave device, that is, after a master receives an 8-bit data sent by a slave on SDA, the master also releases an acknowledge bit which must be "1". However, when all the slaves on the IIC bus are not sending data, that is, they are not pulling down the SDA, the IIC bus will be automatically pulled back to high level due to the existence of the pull-up resistor. Therefore, the acknowledge bit “1” by the master after receiving 8 bits of data is actually the result of the bus being automatically pulled back to the high level by the pull-up resistor. Therefore, the precise academic discourse would state that the master generates a high-level non-acknowledgement bit signal upon completion of data reception.

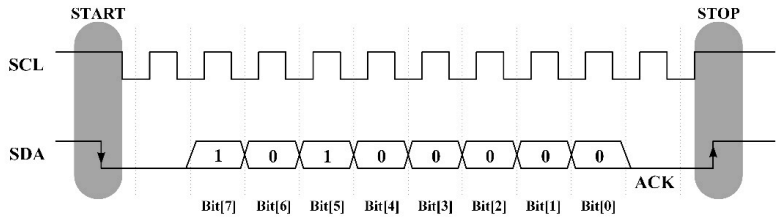

Figure 2. The principle of a master device transmitting data to slave devices within IIC communication protocol

Regardless of the write or the read operation for IIC communication, the higher-position bit, like Bit, in the transmission process is passed first, and the lower-position bit comes after.

3. EEPROM memory: AT24C64

Electrically Erasable Programmable Read-Only Memory (EEPROM) is one of the memory chips commonly used in FPGA board. In this study, the AT24C64 EEPROM, which belongs to the AT24C series manufactured by ATMEL Corporation, is utilized. The AT24C64 features a storage capacity of 64 Kbit. Internally, it is divided into 256 pages, with each page containing 32 bytes, resulting in a total storage of 8192 bytes.

IIC bus supports both multi-master and master-slave operation modes. This study adopts the master-slave operation mode, where there is only one master device initiating data transmission process and generating clock signals while others are peripheral slave devices. Fig. 3 illustrates the pin structure of EEPROM AT24C64.

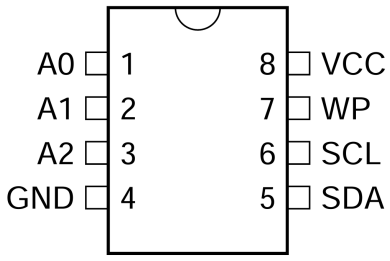

Figure 3. The pins of EEPROM AT24C64

DOI: 10.54254/2755-2721/2026.BJ31080

Fig. 4 illustrates the device address of AT24C64. Fig. 5 illustrates the IIC byte write and read operations for AT24C64 in timing diagrams.

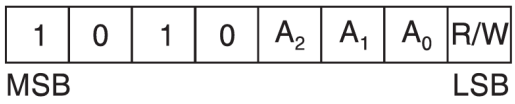

Figure 4. The device address of AT24C64

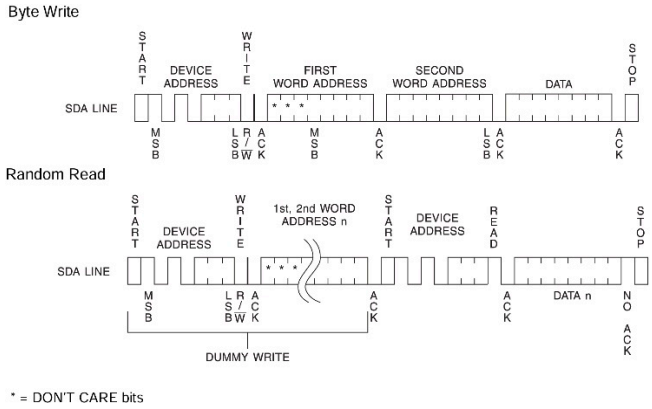

Figure 5. The timing diagrams of IIC write and read operation for AT24C64

4. Serial Peripheral Interface (SPI) protocol

SPI is a synchronous full-duplex communication protocol using four main signals which are Master Out Slave In (MOSI), Master In Slave Out (MISO), Serial Clock (SCLK), and Chip Select/Slave Select (CS/SS). MOSI and MISO sample data bits at the same edge of SCLK. The common SPI transmission rate can reach several megahertz (MHz), and its data transmission rate is usually higher than that of IIC, with no limit on the bit width of data transmission [6], making it suitable for high-speed data exchange scenarios such as video processing, high-speed analog-to-digital converter and SD card interface [7]. SPI is compatible with data transfers at different data bit lengths such as 8-bit, 16-bit, and 32-bit to for enhanced efficiency of data transmission [8]. The CS/SS is used to select a slave device to start transmitting data after being pulled down. However, SPI interface requires more pins than IIC, and to connect multiple slave devices, the master needs to assign independent CS/SS pins correspondingly to each slave device, which makes the design more complex when the system contains more peripheral devices [9]. Fig. 6 illustrates the connection between slave devices and a master device under SPI protocol. The SCLK, MOSI, and MISO signal lines are shared, while the master has every single CS/SS signal line for each slave device. This feature, to a certain extent, limits the scalability and flexibility of SPI in multi-device environments.

DOI: 10.54254/2755-2721/2026.BJ31080

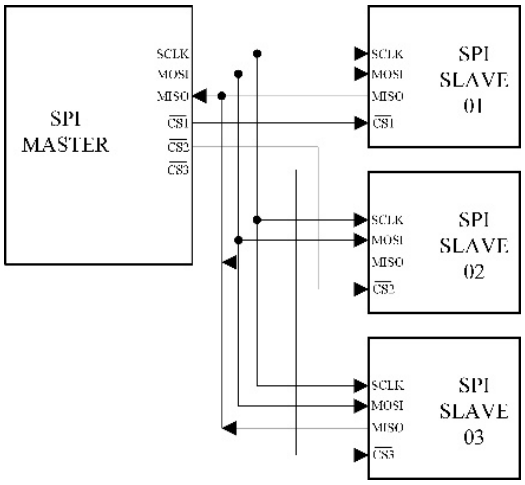

Figure 6. The connection between slave devices and master device under SPI protocol

Fig. 7 illustrates the timing diagram of four SPI modes depended by clock polarity (CPOL) and clock phase (CPHA). SCLK are generated by master and data are sampled at the rising or falling edge of each clock signal. The CPHA, which corresponds to the edge at which the data is sampled, 0 corresponds to the first coming edge, and 1 corresponds to the second edge following. Whether the sampling corresponds to the rising edge or the falling edge still needs to be determined based on the value of the corresponding CPOL. The way to judge the value of CPOL is to see whether the level state of SCLK is 0 or 1 when it is idle, which determines whether CPOL is 0 or 1.

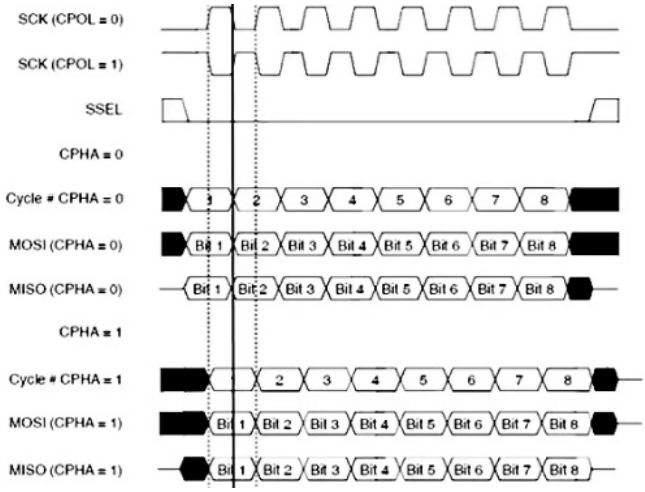

Figure 7. The timing diagram of four SPI modes depended by CPOL and CPHA

This study specifically explains the working principle and process of SPI interface through the use of an analog-to-digital converter ADC128S022. ADC128S022 uses the fourth transmission mode of SPI, that is, the level state of SCLK is high when it is idle (CPOL=1), and the data sampling time is at the second following edge of the SCLK (CPHA=1). Fig. 8 illustrates the connection between master device and ADC128S022 through SPI.

DOI: 10.54254/2755-2721/2026.BJ31080

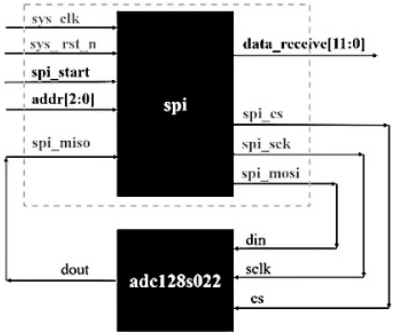

Figure 8. The connection between master device and ADC128S022 through SPI

5. Universal Asynchronous Receiver/Transmitter (UART) protocol

In addition to the baud rate generation module, which is essential for both transmitter and receiver, UART, as an asynchronous full-duplex serial communication protocol, mainly consists of two core modules: UART Transmitter (Tx) and UART Receiver (Rx). The input of the UART transmitter is parallel data and it will convert the parallel data into serial data. The converted serial data is sent to the UART receiver and converted back to the parallel data format. The UART transmitter and receiver operate independently and do not share the same clock signal. The rate of data transmission and reception is configured at the same rate, controlled by the corresponding baud rate generator set in the Rx and Tx. By setting UART transmitter and receiver in both host and peripheral, bidirectional data transmission can be carried out at the same time. When it comes to the communication between the microcontroller and external devices such as RFID, GPS, Bluetooth and GPRS, UART can be used as a communication bridge between these different devices [10]. Due to the features of point-to-point simple architecture, UART is widely utilized for short-distance, low-speed, and low-cost communication between computers and peripheral devices [5]. In the context of internal data transmission on an FPGA board, UART requires two core serial signal lines: one is serial receiver and the others is serial transmitter, as shown in Fig. 9.

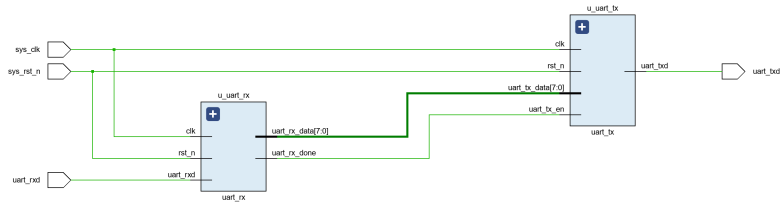

Figure 9. I/O diagram of UART Receiver interfaced with UART Transmitter within FPGA board

Fig. 10 illustrates the timing diagram of one data frame within UART Receiver. If the parity bits are used, the word data length can be from 5 to 8 bits. If the parity bits are not used, the word data length can be 9 bits. 8-bit word data length is commonly used [10]. For UART, the least significant bit (LSB) is transmitted first, the most significant bit (MSB) is transmitted last, and parity bits are used to determine whether any data has been wrongly transmitted [11]. Electromagnetic interference, inconsistent or unstable baud rates, and long-distance data transmission may potentially alter the word data bits.

DOI: 10.54254/2755-2721/2026.BJ31080

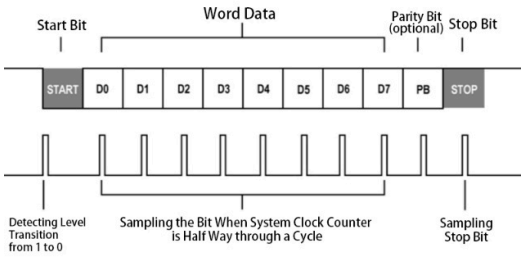

Figure 10. The timing diagram of one data frame within UART Receiver

6. Proposed design

6.1. Finite State Machine (FSM) design of IIC interface

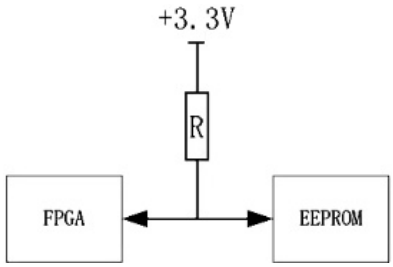

Figure 11. SDA (A bidirectional serial data line between master and slave devices)

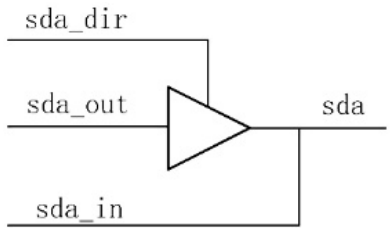

Figure 12. Schematic diagram of the internal three-state gate structure of FPGA

Fig. 11 illustrates that SDA is a bidirectional serial data line between master and slave devices. The three-state gate structure can be used inside the FPGA board in order to avoid the master and slave operating the SDA at the same time. Fig. 12 illustrates the internal three-state gate structure of FPGA. The value of sda_dir represents the IIC data direction. When it is 1, it means that the master outputs signals on sda line. When is 0, the FPGA output high impedance state, which is equivalent to the disconnection of sda_out and sda, indicating the release of control. At this time, sda_in obtains the value of sda line, and sda line inputs data to the master device.

The study divides the whole read and write controller of IIC communication into four main modules, which are IIC Driver (i2c_dri), EEPROM Read-Write (e2prom_rw), Result in LED (rw_result_led), and Top of EEPROM (top_e2prom). Fig. 13 illustrates the I/O block diagram of IIC Protocol. Specifically, IIC Driver module executes read-write operation on the EEPROM in IIC protocol. EEPROM Read-Write module initiates IIC operations, provides word addresses and corresponding data for byte writing, temporarily stores data retrieved from EEPROM through IIC protocol, and verifies the correctness of data read out from written in. Result-in-LED module

DOI: 10.54254/2755-2721/2026.BJ31080

displays the corresponding outcome of read-write operation. Top of EEPROM module is the top-level module that instantiates all sub-functional modules and interconnects their I/O signal pins.

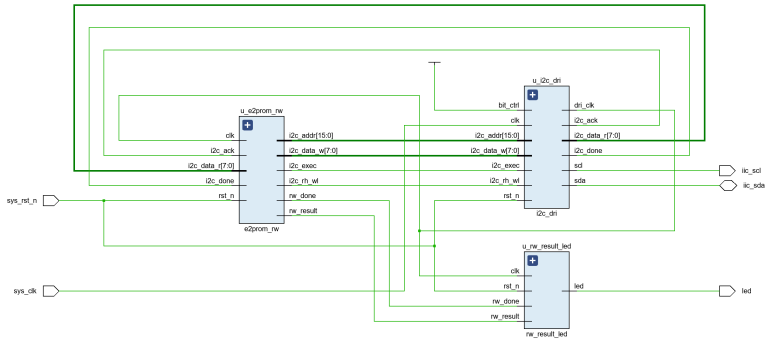

Figure 13. I/O block diagram of IIC protocol

The frequency of system clock of the FPGA board is 50 MHz, while the frequency of the clock that AT24C64 uses is 250 KHz. Therefore, there is a clock signal designated as dri_clk in the code of Verilog, which the FPGA board utilizes to drive the SCL and SDA pins of the EEPROM. To ensure that the level transitions of SDA occur exclusively during the low level of SCL, this study configures the SDA level transitions to take place at the midpoint of the low-level duration of SCL. Consequently, the frequency of dri_clk is four times the frequency of SCL employed by AT24C64.

The architecture of IIC protocol is based on Verilog HDL and is designed through Finite State Machine (FSM) technique which is a cyclic control mechanism that utilizes a finite number of states to record the previous operation state and execute the next operation based on the historical states and current information of action [5]. Fig. 14 illustrates the architecture of FSM for IIC interface controller.

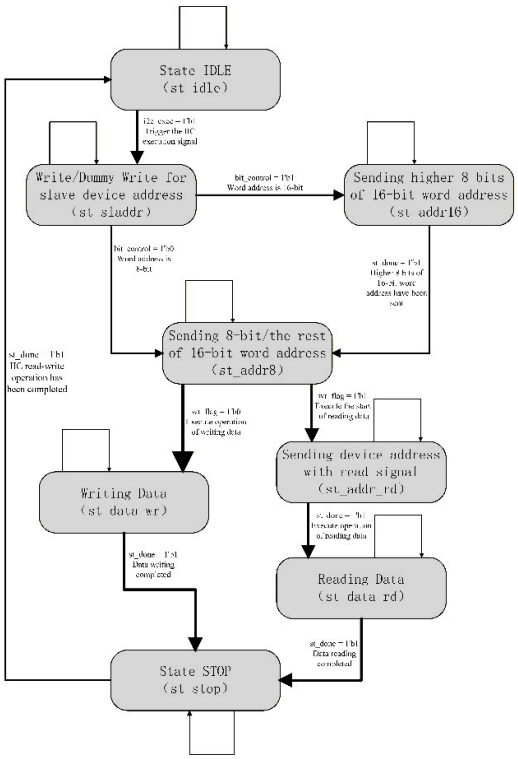

Figure 14. The architecture of FSM for IIC interface controller

DOI: 10.54254/2755-2721/2026.BJ31080

6.2. Clock gating technique design

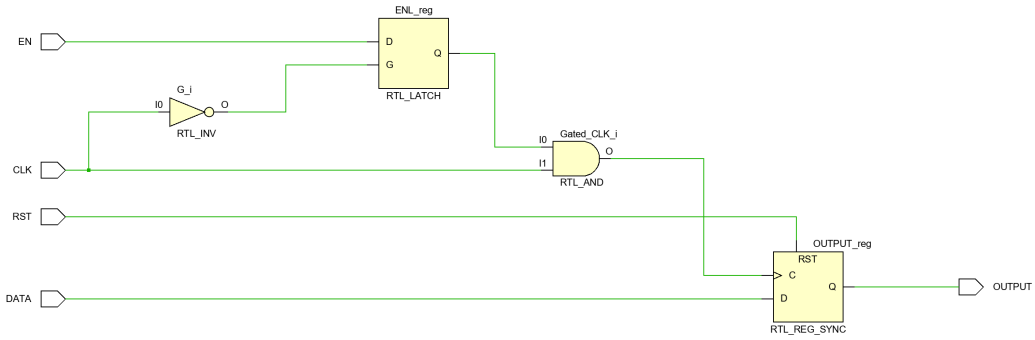

Figure 15. The structure of proposed latch clock gating

Fig. 15 illustrates the structure of proposed latch clock gating. Fig. 16 illustrates the timing diagram of latch clock gating. In the condition that the CLK signal is high, the EN signal of the latch is low, and the latch saves the previous value which is 1 at the midpoint of fig. 16, the value of the Q and GCLK will be 1. It can be seen that despite the presence of glitches in the EN signal, the GCLK is not affected by glitches. When the EN signal settles down to low for a relatively long period, Q will eventually become low, so that GCLK is constant 0 and the clock as a whole part is turned off. Consequently, the glitches and their effects are greatly reduced, and signal become stable and controllable. At the same time, only one latch is needed instead of two, which saves resources and energy.

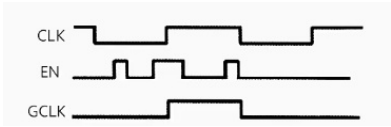

Figure 16. The timing diagram of latch clock gating

6.3. Clock gating technique design of UART transmitter

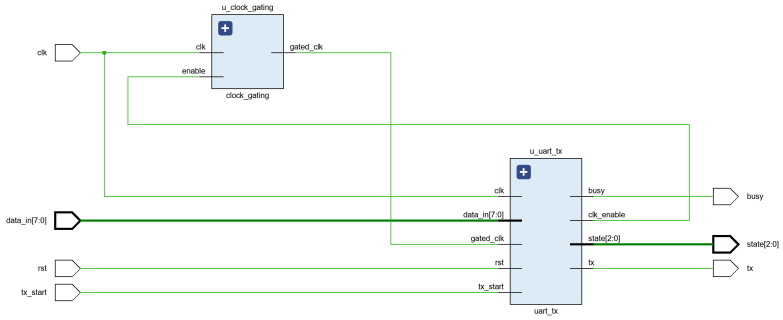

Figure 17. Clock gating technique design of UART Transmitter

Fig. 17 illustrates the clock gating technique that this study utilizes to design the UART Transmitter with a clock gating.

DOI: 10.54254/2755-2721/2026.BJ31080

7. Results and analysis

Figure 18. Simulation of IIC master-slave controller on Vivado simulator

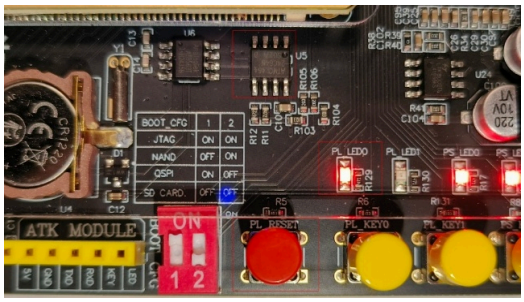

Figure 19. The experimental result of IIC communication between master on FPGA board and EEPROM AT24C64

Fig. 18 illustrates the simulation of IIC master-slave controller on Vivado simulator. The IIC Driver module initiates the write operation, and initiates read operation after the completion of data writing. The read operation reads the data from EEPROM address 0 to address 255 respectively, then EEPROM Read-Write module determines whether the read-out value is consistent with the written value. Before the whole EEPROM read-write operation is completed, the LED light is in the off state. If the whole operation is successfully done, the LED light is in the constant-on state. If the operation fails, the LED will blink continuously. As Fig. 19 shows, the result of this implementation on FPGA board is that the LED light stays on and still remains in a constant-on state after repeated read-write operations, indicating that the IIC communication between master and slave experiment is successful.

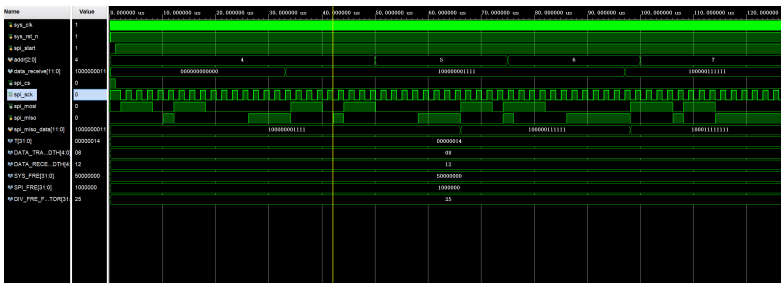

Figure 20. Simulation of SPI protocol on Vivado simulator

Fig. 20 illustrates the simulation of SPI protocol on Vivado simulator. Fig. 21 illustrates the simulation of UART transmission on Modelsim simulator. If a baud rate of 115200 is utilized, the bit

DOI: 10.54254/2755-2721/2026.BJ31080

rate of the serial port is 115200Bps×1bit = 115200bps. It is calculated that the time for the serial port to transmit or receive 1bit data is one baud, that is, 1/115200s. If a system clock of 50MHz (50,000,000 cycles per second) is used to count, the parameter cnt is the number of clock cycles required for each bit transmission. Therefore, cnt = 50000000/115200, which corresponds to approximately 434 system clock cycles. Essentially, the interval between bits exists for about 434 system clock cycles at a clock frequency of 50MHz, so that a baud rate counter of at least 9-bit length is required for counting, but can also be set to 16-bit length for other baud rates. Sampling a data bit is theoretically most stable when the system clock counter is half way through a cycle. UART Transmitter works the same way as UART Receiver. UART Receiver is serial-to-parallel, while UART Transmitter is parallel-to-serial.

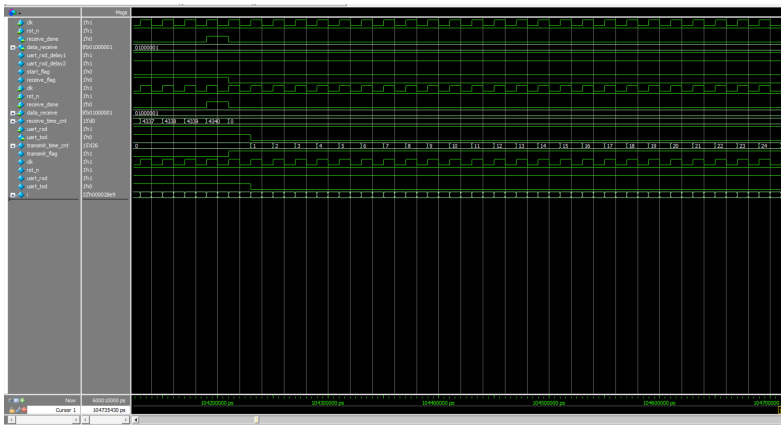

Figure 21. Simulation of UART transmission on Modelsim simulator

Table. 1 shows the specific total power, LUT and FF amount of IIC, UART and SPI protocol synthesized by Vivado in this study.

Table 1. IIC, UART and SPI comparison of total power and general structure synthesized by Vivado

Serial Communication Protocol	Total Power	Look-Up Table (LUT)	Flip-Flop (FF)
IIC	0.093W	148	136
UART	0.107W	61	70
SPI	0.161W	231	205

8. Discussion

In terms of low power design strategy, by introducing clock gating and optimizing finite state machine in protocol controller, the overhead of invalid flips and redundant logic is effectively reduced. The feasibility and effectiveness of the method for the low power design of communication interface are substantiated and demonstrated. In the aspect of practical value, this study is not limited to the simulation verification, but realizes the read-write operation through the EEPROM peripheral on board to ensure that the design can be operated stably in practical application scenarios. This verification method makes up for the general lack of peripheral verification in the existing literature, and makes the research closer to engineering practice. The comparison of IIC, SPI and UART provides a reference for the selection of protocols in embedded system design and for the exploration of Verilog design optimization to improve system performance.

DOI: 10.54254/2755-2721/2026.BJ31080

Notwithstanding, this study is subject to certain limitations. Firstly, the power consumption is estimated based on synthesis and simulation on Vivado, and lack of measurement in real hardware environment. Second, the SPI and UART modules implemented in this study adopt core-functional models and therefore exclude several advanced protocol features. For SPI, these include multi-slave chip-select management, full CPOL/CPHA configurability, variable word-length support, and FIFO buffering. For UART, omitted functions include hardware flow control, auto-baud detection, extended framing formats, and comprehensive error-handling mechanisms. Consequently, the comparative results reflect baseline implementations, and incorporating these advanced features may lead to different power, latency, and resource profiles. In the future research, the current and voltage measurement circuit can be introduced on the FPGA board to further verify the effect of power consumption optimization. At the same time, multi-protocol collaborative optimization strategies can also be explored to support more complex requirements of SoC system.

9. Conclusion

This study proposed a type of advanced clock gating technique and managed to apply it to the UART interface, which can lower the power consumption while ensuring efficient data transmission. Through the finite state machine technique for data transmission control, IIC, SPI and UART communication can be realized on FPGA. Through parametric design, each communication protocol can easily adapt to the peripheral devices with the same protocol but different transmission requirements. This study utilizes Verilog HDL and Xilinx Vivado to evaluate the application scenarios of IIC, SPI and UART through the analysis of power consumption and protocol structure.

The results show that IIC has the lowest power consumption and is suitable for multi-device, medium-speed transmission and low-power scenarios. SPI has the highest transmission rate, which is suitable for high-speed, low-latency and huge data transmission, but its power consumption is relatively high, while UART has the lowest rate but a clear structure, making it suitable for point-to-point communication.

Lastly, the design and verification methodology for the low-power IIC protocol controller proposed in this paper not only fills an academic research gap in the integration of low-power design and peripheral verification but also provides a valuable referential implementation path for engineering applications. This study, to a certain extent, contributes to advancing the development of low-power embedded communication interfaces and lays a foundation for future multi-protocol integration and system optimization.

References

[1] Wang H B, Qiao P W. Design of IIC interface controller based on FPGA. In: 2024 9th International Conference on Electronic Technology and Information Science (ICETIS), Hangzhou, China, May 17-19, 2024: 177-180.
[2] Bagdalkar P, Ali L. Hardware implementation of I2C controller on FPGA and validation through interfacing with low-cost ADC. In: 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, Jan 8-10, 2020: 887-891.
[3] Huang C X, Yang S. A mini I²C bus interface circuit design and its VLSI implementation. Journal of Supercomputing, 2024, 80(16): 23794-23814.
[4] Anagha A, Mathurakani M. Prototyping of dual master I²C bus controller. In: 2016 IEEE International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, Apr 6-8, 2016: 2124-2129.
[5] Sung G M, Tung L F, Huang C J, Yu C P. Front-end gateway system with serial communication protocol conversion and edge computing platforms. IEEE Access, 2023, 11: 93193-93203.
[6] Trehan M, Kumar P, Gaur N. Multi-protocol conversion unit implementation on a FPGA. In: 2024 2nd IEEE World Conference on Communication and Computing (WCONF), Raipur, India, Jul 12-14, 2024.

DOI: 10.54254/2755-2721/2026.BJ31080

[7] Muneeswaran V, Kumar K N, Manikandan P, Kumar M N, Chaithanya C V, Priyadarshini M G. Performance analysis of communication protocols (I2C, SPI) for embedded applications with IMU sensor using Verilog. In: 2025 5th International Conference on Trends in Material Science and Inventive Materials (ICTMIM), Chennai, India, Jan 9-10, 2025: 685-689.
[8] Maity A, Samanta P K, De B P, Dash S K, Bhowmik W, Bakshi A. FPGA implementation of serial peripheral interface transceiver module. In: 2024 International Conference on Computer, Electrical & Communication Engineering (ICCECE), Kolkata, India, Jan 25-26, 2024: 1-5.
[9] Hobden P, Srivastava S. Low cost FPGA implementation of a SPI over high speed optical SerDes. In: 2018 4th IEEE International Symposium on Smart Electronic Systems (iSES), Hyderabad, India, Dec 17-19, 2018: 146-151.
[10] Jeevan B, Sahithi P, Samskruthi P, Sivani K. Simulation and synthesis of UART through FPGA Zedboard for IoT applications. In: 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, Jan 28-29, 2022.
[11] Sahay N, Gajjar S. Design and UVM based verification of UART, SPI, and I²C protocols. In: 2024 5th International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, Sep 18-20, 2024: 330-335.