IMPLEMENTATION OF I^2C BUS PROTOCOL ON FPGA

Madhuri Hanumanta Daware[1], Prof. A. S. Patil[2]
Department of Electronics & Telecommunications, PVPIT, Bavdhan
Pune, India
Email: [1]madhuridaware.21@gmail.com,[2] patilalaknanda@yahoo.com

Abstract- **I^2C bus defined by Philips providing a simple way to talk between IC's by using a minimum number of pins. This bus is called the Inter IC or I2C bus. All I^2C bus compatible devices incorporate an on-chip interface which allows them to communicate directly with each other via the I^2C bus. This design concept solves the many interfacing problems encountered when designing digital control circuits. This paper implements I2C (Inter IC Communication) master bus controller for interfacing low speed peripheral devices using field programmable gate array. The I^2C master bus controller interfaced with slave devices Real Time Clock (DS1302) and EEPROM. This module was designed using VHDL. The design was synthesized using Xilinx ISE Design Suite 14.7 and implemented on Altera Cyclone IV FPGA.**

Keywords- I2C, SDA, SCL, RTC (DS1302), EEPROM, VHDL, FPGA, Master, Slave.

I.INTRODUCTION

I^2C stands for Inter IC Communication or Inter-Integrated Circuit. There are different protocols to achieve serial communication like RS-232, RS-422, RS-485, SPI, Microwire. These protocols require more pin connection in the integrated circuit to achieve serial communication. The I^2C (Inter-IC) bus protocol was developed by Phillips Electronics to allow communication between integrated circuits (ICs) from different manufacturers. Applications that use the I^2C bus include microcontrollers, LCD, memory devices, PCs, cell-phones, Television, ADCs, DACs and other devices. It's I^2C (Inter-Integrated Circuit, referred to as I-squared-C, I-two-C or IIC. The I2C bus uses two bidirectional signals, one as the serial clock (SCL) line and other as the serial data (SDA) line. Each device connected to the bus has a unique address used to identify the device in communication. The protocol is comprised of a set of conditions to establish or terminate communication.

In this paper implementing I2C bus protocol for interfacing low speed peripheral devices on FPGA. It is also the best bus for the control applications, where devices may have to be added or removed from the system. I2C protocol can also be used for communication between multiple circuit boards in equipments with or without using a shielded cable depending on the distance and speed of data transfer. I2C bus is a medium for communication where master controller is used to send and receive data to and from the slave DS1302.

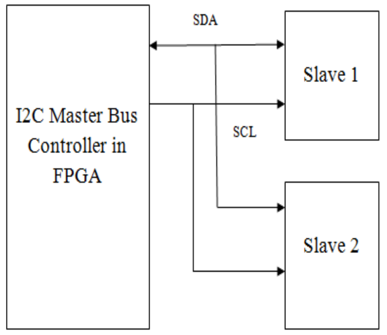

Figure 1: Block Diagram of Proposed System

II.PROPOSED WORK

A. I2C Protocol

I2C is a two wire, bidirectional serial bus that provides effective data communication between two devices. I2C bus supports many devices and each device is recognized by its unique address.

B. SCL, SDA Lines

The I2C bus physically consists of 2 active wire connections. The active wires called SDA and SCL. SDA is Bi-directional signal, SCL is uni-directional. SCL is the clock line. It is used to synchronize all data transfers over the I2C bus. SDA is the data line. The SCL and SDA lines are connected to all devices on the I2C bus.

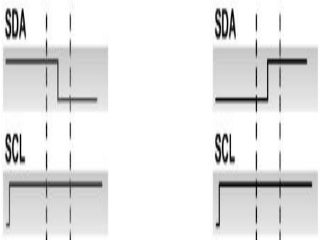

Figure 2: (a) “**START**” Sequence
(b) “**STOP**” Sequence

The I2C bus is said to be idle when both SCL and SDA are at logic 1 level. When the master (controller) wishes to transmit data to a slave it begins by issuing a start sequence on the I2C bus, which is a high to low transition on the SDA line while the SCL line is high as shown in Fig– 2(a).The bus is considered to be busy after the START condition. After the START condition, slave address is sent by the master. The slave device whose address matches the address that is being sent out by the master will respond with an acknowledgement bit on the SDA line by pulling the SDA line low. Data is transferred in sequences of 8 bits. The bits are placed on the SDA line starting with the MSB (Most Significant Bit). For every 8 bits transferred, the slave device receiving the data sends back an acknowledge bit, so there are actually 9 SCL clock pulses to transfer each 8 bit byte of data this is shown in Fig-3. If the receiving device sends back a low ACK bit, then it has received the data and is ready to accept another byte. If it sends back a high then it is indicating it cannot accept any further data and the master should terminate the transfer by sending a STOP sequence. In Fig-2(b) which shows the STOP sequence, where the SDA line is driven low while SCL line is high. This signals the end of the transaction with the slave device.

C. Serial Data Communication

The I2C bus has two modes of operation: master transmitter and master receiver. The I2C master bus initiates data transfer and can drive both SDA and SCL lines. Slave device is addressed by the master. It can issue only data on the SDA line. In master transmission mode, after the initiation of the START sequence, the master sends out a slave address. The address byte contains the 7 bit address followed by the direction bit (R/ w). After receiving and decoding the address byte the device outputs acknowledge on the SDA line. After the slave device acknowledges the slave address + write bit, the master transmits a register address to the slave device this will set the register pointer on the slave. The master will then begin transmitting each byte of data with the slave acknowledging each byte received. The master will generate a stop condition to terminate the data write.

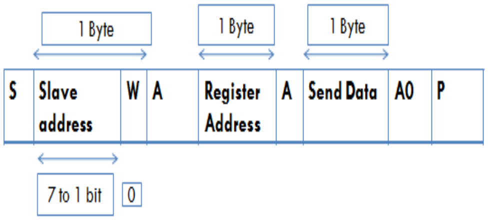

Figure 3: Master Transmission Mode

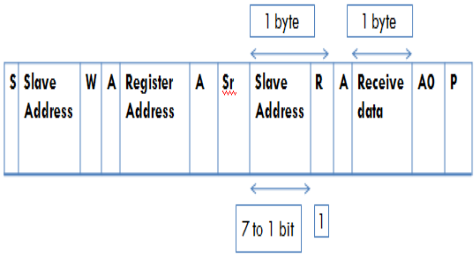

Figure 4: Master Receiver Mode

In master receiver mode, the first byte is received and handled as in the master transmission mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the slave device while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (Fig-4). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7-bit slave address followed by the direction bit (R/ *w*). After receiving and decoding the address byte the device inputs acknowledge on the SDA line. The slave then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written before the initiation of a read mode, the first address that is read is the last one stored in the register pointer. The slave must receive a “not acknowledged” to end a read.

III. SOFTWARE IMPLEMENTATION

I2C master controller is designed using VHDL based on Finite State Machine (FSM). FSM is a sequential circuit that uses a finite number of states to keep track of its history of operations, and based on history of operation and current input, determines the next state. There are several states in obtaining the result. Figure 5 shows write operation on EEPROM. The low speed peripherals RTC (DS1302) and EEPROM are interfaced with I2C master bus through Nios-II Softcore Processor and implemented on Cyclone IV FPGA.

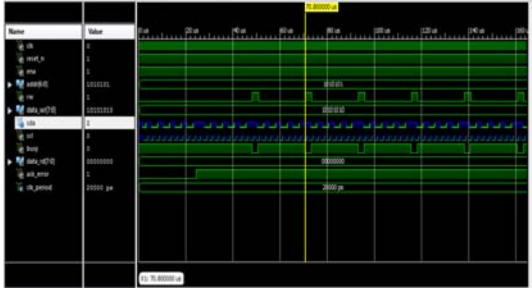

Figure 5: Simulation Results of Write Operation on EEPROM

IV. HARDWARE IMPLEMENTATION

The hardware implementation is done by interfacing Slave devices RTC (DS1302) and EEPROM with the I^{2}C master present in Altera Cyclone IV FPGA through Nios-II Softcore Processor.

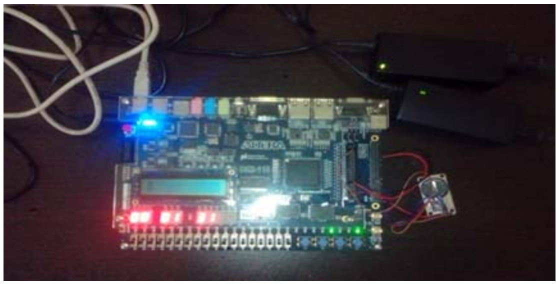

Figure 6: Hardware implementation in Altera Cyclone IV FPGA

V. ADVANTAGE OF COMMUNICATION WITH I2C

FPGAs are particularly well suited to meet the requirements of many data processing applications. FPGAs have the following characteristics that make them very appealing

1) High performance: HD processing can be implemented in a single FPGA.

2) Flexibility: FPGAs provide the ability to upgrade architectures quickly to meet evolving requirements, while scalability allows use of FPGAs in low-cost and high-performance systems.

3) Low development cost: Data development kits from start as low as US$1,095 and include the software tools required to develop a data system using FPGAs.

4) Obsolescence proof: FPGAs have a very large customer base on ship products for many years on. In addition, FPGA designs are easily migrated from one process node to the next.

5) Plan for lower production costs: offers several ways to help plan for the time when products move from lower unit volumes to much higher volumes.

VI. CONCLUSION

This project has shown up results of hardware implementation of I2C communication protocol on Cyclone IV FPGA interfaced with slaves RTC (DS1302) and EEPROM. I2C master controller communicates with slave devices using Nios-II Soft-Core Processor. I2C bus is used by many integrated circuits and is simple to implement. Any FPGA/Microcontroller can communicate with I2C devices even if it has no special I2C interface. I2C specifications are flexible, can communicate with slow devices and can also use high speed mode to transfer large amount of data. Because of many advantages, I2C bus will remain as one of the most popular serial interfaces to connect integrated circuits on board.

ACKNOWLEDGMENT

The authors would like to thank the researchers as well as publishers for making their resources available and teachers and guide for their guidance. We are thankful to the authorities of Savitribai Phule University,Pune. We also thank the college authorities for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

REFERENCES

[1] Bollam Eswari, N. Ponmagal, K. Preethi, S. G. Sreejeesh , “ Implementation of I^{2}C Master Bus Controller on FPGA,” International

Conference on Communication and Signal Processing, India, April 3-5, 2013, pp. 678-681.

[2] Prof. Jai Karan Singh, "Design and Implementation of I2C master controller on FPGA using VHDL," IJET, Vol. No. 4, Aug-Sep 2012.

[3] Philips Semiconductor, "I2C Bus Specification" version 2.1, January 2000.

[4] Arvind Sahu,Ravi Shankar Mishra,Puran Gour, "Design and Interfacing of High speed model of FPGA using I2C protocol," Int. J. Comp. Tech. Appl., Vol 2 (3), 531-536.

[5] Trupti D. Shingare, R. T. Patil, "SPI Implementation on FPGA," International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-2, Issue-2, January 2013.

[6] Pankaj Kumar Mehto, Pragya Mishra , Sonu Lal, "Design and Implementation for Interfacing Two Integrated Device Using I2C Bus, " International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 3, March 2013.

[7] J. J Patel, B. H. Soni, "Design and Implementation of I2c Bus Controller Using Verilog," Journal of Information, Knowledge and Research in Electronics and Communication Engineering Nov 12 To Oct 13, Vol. 02, Issue – 02, page no. 520-522.

[8] A.Sahu, R. Mishra, P.Gour, "An Implementation of I2C using VHDL for Data Surveillance", International Journal on Computer Science and Engineering, pp‒ 1857‒1865, May 2011.