International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: 6.614
Available Online at www.journalijcar.org
Volume 9; Issue 04 (A); April 2020; Page No.21855-21858
DOI: http://dx.doi.org/10.24327/ijcar.2020.21858.4309

Research Article

IMPLEMENTATION OF I2C COMMUNICATION PROTOCOL WITH RTC AND EEPROM ON FPGA

J. Sushmitha, J. Nikhil, G. Swaroop Kumar, L. Jaswanth, and Mr. J. Yeshwanth Reddy,

Department of Electronics &Communication

A R T I C L E I N F O

Article History:
Received 06th January, 2020
Received in revised form 14th February, 2020
Accepted 23rd March, 2020
Published online 28th April, 2020

Key words:
I2C, SDA, SCL, RTC (DS1307), EEPROM, VHDL, FPGA, Master, Slave.

A B S T R A C T

I^{2}C 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^{2}C bus compatible devices incorporate an on- chip interface which allows them to communicate directly with each other via the I^{2}C 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^{2}C master bus controller interfaced with slave devices Real Time Clock (DS1307) and EEPROM. This module was designed using VHDL. The design was synthesized using Xilinx ISE Design Suite 14.7 and implemented on SPARTAN 3E XC3S200 FPGA.

INTRODUCTION

I^{2}C 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^{2}C (Inter-IC) bus protocol was developed by Phillips Electronics to allow communication between integrated circuits (ICs) from different manufacturers. Applications that use the I^{2}C bus include microcontrollers, LCD, memory devices, PCs, cell-phones, Television, ADCs, DACs and other devices. It's I^{2}C (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 DS1307.

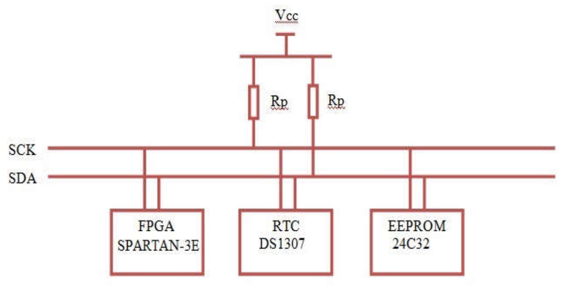

Figure 1 Block Diagram of Proposed System

Proposed Work

I2CProtocol

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.

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.

Corresponding author:* **J. Sushmitha
Department of Electronics &Communications

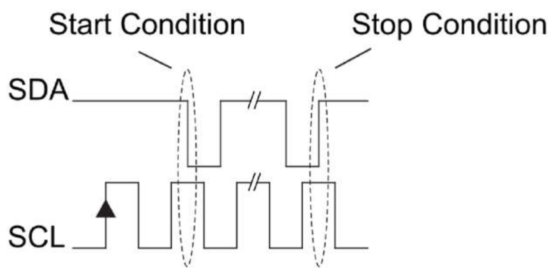

Figure 2 "**START" and** "**STOP"** Condition

The I2C is said to idle when both SCL and SDA are at logic 1 level. When the master wishes to transmit data to a slve it begins by issuing a start condition on the I2C bus, which is high to low transition on the SDA line while the SCL is high as shown in Fig-2. The bus is considered to be busy after the START condition. After the START condition, slave address which includes the read or write bit 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 condition. In Fig-2 which shows the STOP condition, where the SDA line is driven low while SCL line is high. This signals the end of the transaction with the slave device.

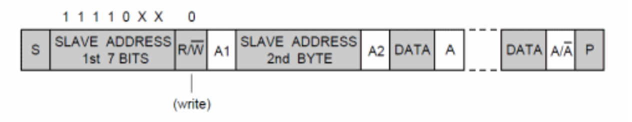

Figure 3 Data Transfer Format on I2C bus

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 2sends 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.

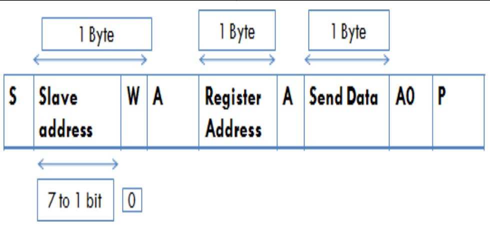

Figure 4 Master Transmission Mode

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-5). 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.

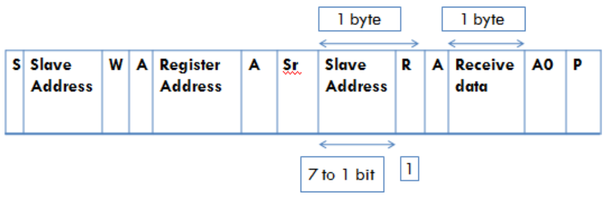

Figure 5 Master Receiver mode

Software Implementation

I^2C master protocol 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.

Algorithm

State 1: An idle condition: I^2C bus doesn't perform any operation. (SCL and SDA remains high).
State 2: Start condition: master initiates data transmission by providing START (SCL is high and SDA is from high to low).
State 3: Slave address - write: master sends the slave address-write (11010000) to the slave.
State 4: If the slave address matches with the slave, it sends an acknowledgement bit in response to the master.
State 5: 8 Bit Register Address[12] will be transmitted to the slave. Again acknowledgement is sent to the master by the slave.
State 6: Data to be transmitted is sent to the slave by the master. After receiving the data, slave acknowledges the master.
State 7: Stop condition: Slave sends a stop bit to the master to terminate the communication (SCL is high and SDA is from Low to high). For performing read operation, write operation is performed first and then read operation is done. Slave address for read is 11010001. (State 7 will not be performed for read operation)`

State 8: Master transmits slave address for read operation to the slave.
State 9: Master receives the data from the slave and acknowledges the slave.
State 10: Master sends a STOP bit to terminate the connection (SCL is high and SDA is from Low to high).

Fig.6 shows the flowchart for I^{2}C master bus communication with slave device. Fig-7 shows the simulation result for I^{2}C, a *dataout*, *sda, read* and *write* operations are observed in the I^{2}C Master. To read the written data from the slave, the write operation takes place first followed by the repeated start condition and sending the slave address read (11010001) in each state of FSM.

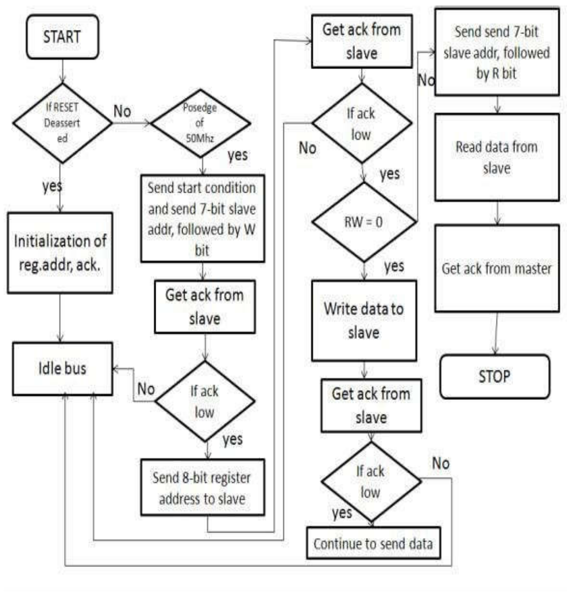

Figure 6 Flowchart for I^{2}C master bus communication

RESULTS AND DISCUSSION

Here the I^{2}C bus protocol is designed using VHDL and implemented in Spartan 3E XC3S200 using Xilinx ISE Design Suite14.7. The simulation results are shown below:

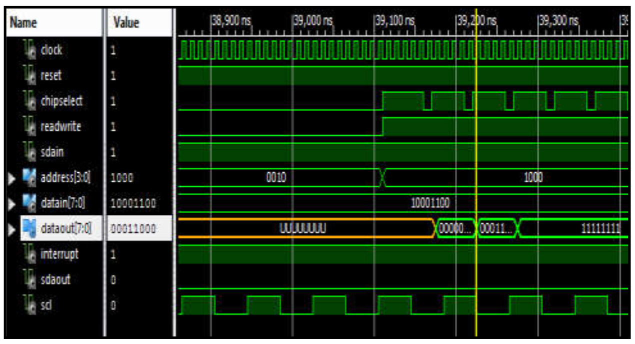

Figure 7 Simulation Results of I2C Master

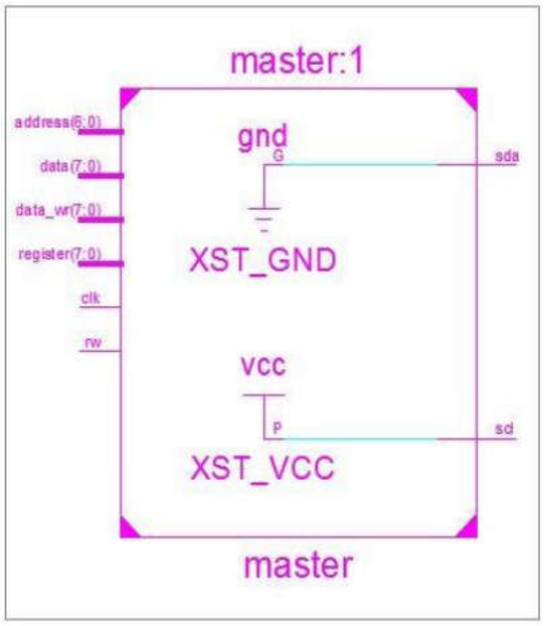

Figure 8 Synthesis Results of I2C Master

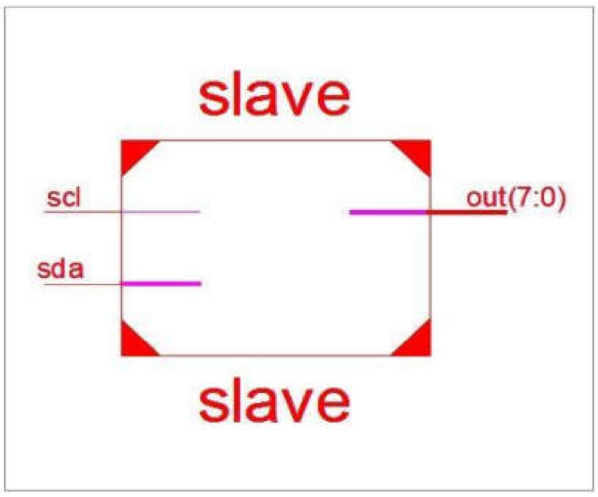

Figure 9 Synthesis Results of Slave

CONCLUSION

This project has shown up results of I2C communication protocol.I2C master controller communicates with slave 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.In future this can be implemented as real time in networks that contains multiple masters and multiple slaves to coordinate the entire system by clock synchronization techniques.

References

1. M.Mohankumar , N.Birundha ,T.Suganya, “Interfacing of I2C Master Bus Controller with FPGA” Asian Journal of Applied Science and Technology (AJAST) (Open Access Quarterly International Journal) Volume 2, Issue 2, Pages 172-180, April-June 2018

2. J. J Patel, Prof B. H. Soni "Design and implementation of I2C bus controller using Verilog" Journal of Information Knowledge and Research in Electronics and Communication Engineering ISSN: 0975– 6779, Volume–02, Issue-02, pp 520-522 Nov. 2012.
3. Shoaib. Shah Sobhan, Sudipta. Das and Iqbalur. Rahman "Implementation of I2C using System Verilog and FPGA" International Conference on Advancements in Electronics and Power Engineering (ICAEPE'2011), Bangkok, Dec 2011.
4. Madhuri Hanumanta Daware, Prof. A. S. Patil, "Implementation of I2 C Bus Protocol on Fpga" International Journal Of Current Engineering And Scientific Research (IJCESR) Issn (Print): 2393-8374, (Online): 2394-0697, Volume-2, Issue-8, 2015
5. Radha RC, Ravuri Aneesh Kumar, "Design and Implementation of I2C Communication Protocol on FPGA for EEPROM" International Journal of Scientific & Engineering Research, Volume 5, Issue 3, March-2014 ISSN 2229-5518.
6. Philips Semiconductor "I2C Specification" version 2.1 2000.

How to cite this article:

J. Sushmitha *et al* (2020) 'Implementation of i2c Communication Protocol With rtc and Eeprom on Fpga', *International Journal of Current Advanced Research,* 09(04), pp. 21855-21858. DOI: http://dx.doi.org/10.24327/ijcar.2020.21858.4309

****