

Implementation Of I2c Multi Task And Multi Slave Bus Controller Using Verilog

Shaik.Fazil Ahmed Y.Murali,,M.Tech

Department of Electronics and communication Engineering
 Audisankara College of Engineering & Technology, Gudur
 (Autonomous)

Abstract - This paper implements serial data communication using I²C (Inter-Integrated Circuit) master bus controller using a field programmable gate array (FPGA). The I²C master bus controller was interfaced with MAXIM DS1307, which act as a slave. This module was designed in Verilog HDL and simulated in Modelsim 10.1c. The design was synthesized using Xilinx ISE Design Suite 14.2. I²C master initiates data transmission and in order slave responds to it. It can be used to interface low speed peripherals like motherboard, embedded system, mobile phones, set top boxes, DVD, PDA's or other electronic devices.

Keywords – DS1307, FPGA, I²C, master, Modelsim, serial data communication, slave, Spartan 3AN, Xilinx.

I. INTRODUCTION

In the world of serial data communication [8], there are protocols like RS-232, RS-422, RS-485, SPI (Serial peripheral interface), Microwire for interfacing high speed and low speed peripherals. These protocols require more pin connection in the IC(Integrated Circuit) for serial data communication to take place, as the physical size of IC have decreased over the years, we require less amount of pin connection for serial data transfer. USB/SPI/Microwire and mostly UARTS are all just 'one point to one point' data transfer bus systems.

They use multiplexing of the data path and forwarding of messages to service multiple devices. To overcome this problem, the I²C [6] protocol was introduced by Phillips which requires only two lines for communication with two or more chips and can control a network of device chips with just a two general purpose I/O pins whereas, other bus protocols require more pins and signals to connect devices.

In this project, we are implementing I²C bus protocol for interfacing low speed peripheral devices on FPGA [3]. 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 [9] is used to send and receive data to and from the slave DS1307. The low speed peripheral, DS1307 is interfaced with I²C master bus and synthesized on Spartan 3AN. Fig-1 shows the I²C bus system with the I²C master controller implemented on a FPGA and the real time clock device acting as the slave.

The synopsis of the paper is as follows: In section 2, we discussed I²C protocol of our proposed design which also presents module description for our proposed system. In section 3, we present the software implementation along with algorithm and flow chart. In section 4, holds the detailed description of hardware implementation of I²C master bus

controller in Spartan 3AN FPGA Design kit using Xilinx software. Finally, concluded with future scaleup in section 5.

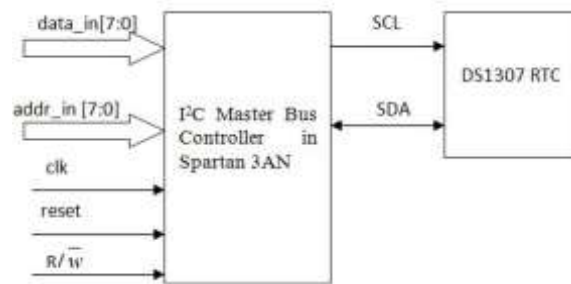


Fig-1 I/O Diagram of I²C Master Controller interfaced with DS1307 RTC slave device

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. In Fig -1 data_in and addr_in are the 8 bit address given as an input. Clk and reset are the input lines used to initiate the bus controller process. The R/w signal is given as an input to indicate whether master or slave acts as a transmitter in the data transmission.

The physical I²C bus consists of just two wires, called SCL and SDA. 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. As both SCL and SDA lines are "open drain" drivers they are pulled up using pull up resistors [1].

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 (DS1307) 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,

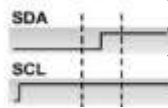
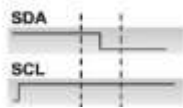


Fig- 2 (a) “START” Sequence. (b) “STOP” Sequence

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.

B. 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.[4] Slave device (DS1307) 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 DS1307 address, which is 1101000, followed by the direction bit (R/ w). After receiving and decoding the address byte the device outputs acknowledge on the SDA line. After the DS1307 acknowledges the slave address + write bit, the master transmits a register address to the DS1307 this will set the register pointer on the DS1307. The master will then begin transmitting each byte of data with the DS1307 acknowledging each byte received. The master will generate a stop condition to terminate the data write.

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 DS1307 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 DS1307 address, which is 1101000,

followed by the direction bit (R/ w). After receiving and decoding the address byte the device inputs acknowledge on the SDA line. The DS1307 then begins to transmit data starting with the register address pointed to by the

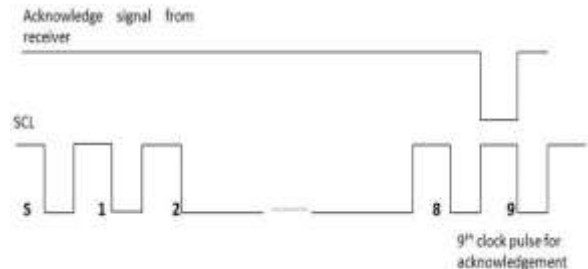


Fig-3 Acknowledgement on the I²C Bus register pointer.

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 DS1307 must receive a “not acknowledged” to end a read.

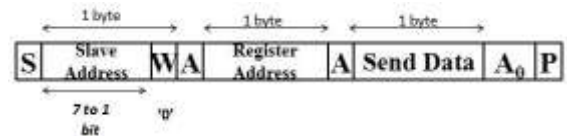


Fig -4 Master Transmission Mode

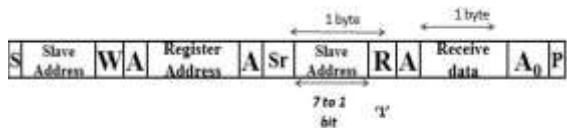


Fig- 5 Master Receiver Mode

C. MAXIM DS1307

The DS1307 supports a bi-directional, 2-wire bus and data transmission protocol. The pin assignment of DS1307 is given in Fig-6. The DS1307 operates as a slave on the 2- wire bus Fig-7

shows the interface connection of I²C bus in Spartan 3AN with the DS1307 RTC chip [2].

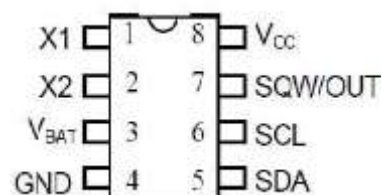


Fig-6 Pin Assignment of DS1307

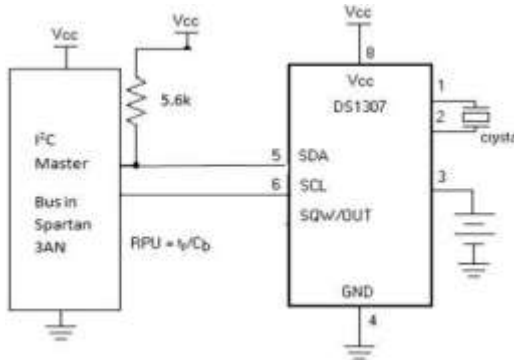


Fig-7 DS1307 connected to two wire data bus

III. SOFTWARE IMPLEMENTATION

Algorithm

State1 : An idle condition: I²C 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).

I²C master controller is designed using Verilog HDL [5] based on Finite State Machine (FSM) [13]. 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.



Fig-8 Flowchart for I²C master bus communication with slave device

Fig-8 shows the flowchart for I²C master bus communication with slave device. Fig-9 shows the Modelsim simulation result for write operation, the given data input is written in to slave register address in each state of FSM programmed in Verilog HDL. Fig-10 shows the Modelsim simulation result for read operation, 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.

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).

The simulated Verilog coding is synthesized on Spartan 3AN through Xilinx ISE Design Suite14.2 [7]. The design is analysed using ChipScope Analyzer in Spartan 3AN platform.

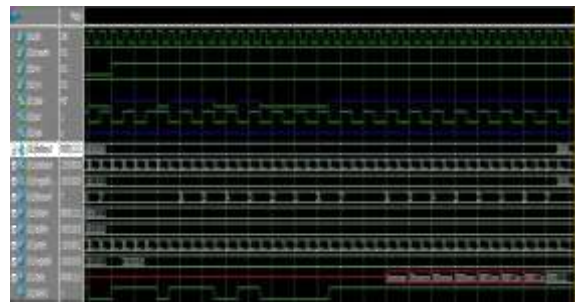


Fig-9 Modelsim Simulation Result for Write operation

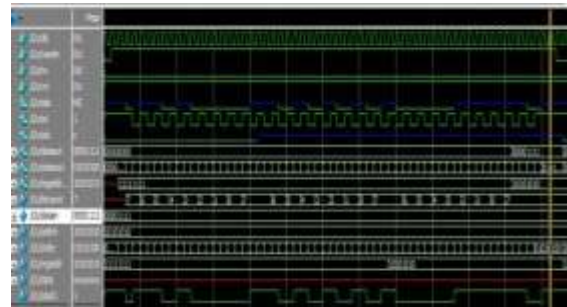


Fig-10 Modelsim Simulation Result for Read operation

IV. EXPERIMENTAL RESULTS

The Design is simulated using Xilinx 14.3. The Controller is designed using Verilog HDL. Figures below shows the simulation results,RTL schematic and Technology schematics of i2c multi task and multi slave bus controller



Fig.11 Modelsim Simulation Result for Master Transmitter



Fig:12 Modelsim Simulation Result for Master receiver



Fig:13 Modelsim Simulation Result for Start Stop Bit Generator

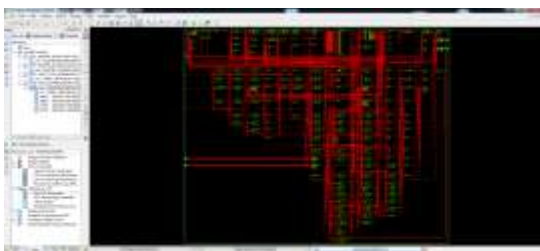


Fig:14master master RTL Schematics

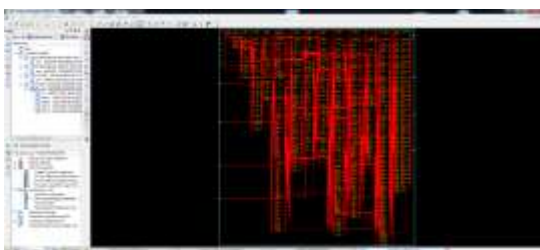


Fig:15 master master Rtech sch copy



Fig:16 master_reciever_synthesis_RTL - Copy



Fig:17 master_reciever_synthesis_TECHSCH



Fig:18 MASTERTRANSMITTERSYNTHESIS_R

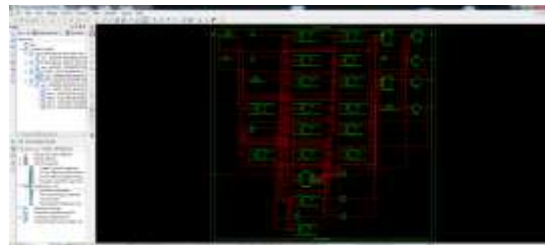


Fig:19 START_STOP_BLOCK_RTL



Fig:20 START_STOP_BLOCK_TECH SCH

The result shows that minimal resources are utilized in designing the I²C master as only 2 % slices, 0% flip flops and 2% LUTs are utilized.

Device Utilization Summary(estimated values)



Fig-22 Hardware implementation in Spartan 3AN design kit

Logic Utilization	Used	Available	Utilization
Number of slices	169	5888	2%
Number of slice flip flops	83	11776	0%
Number of 4 input LUTs	325	11776	2%
Number of bonded	31	372	8%
Number of GCLKs	1	24	4%

Fig-21 Xilinx Device Utilization Summary

V. HARDWARE IMPLEMENTATION

The hardware implementation is done by interfacing DS 1307 with the I²C master present in Spartan 3AN [10] kit. It is an 8 pin DIP. A power supply of 5V is given to the slave. Pin 4 is connected to the ground (GND) and pin 8 is connected to the V_{CC}. Pins 5 and 6 are connected to SDA and SCL of I²C master bus in Spartan 3AN. Both SDA and SCL are open drained lines. When multiple masters are connected, both SDA and SCL are in need to be pulled up with a 5.6kΩ resistor to the V_{CC}. Since only one master is connected, there is no need to pull up SCL. Red wire indicates V_{CC}, Black wire indicates ground (GND), yellow wire indicates SCL line and Green wire indicates SDA line connected with Spartan 3AN FPGA kit.

After completing the coding in Verilog HDL, it is downloaded to the Spartan 3AN kit using Xilinx software. And corresponding input is given according to the I²C protocol.

VI CONCLUSION

This project demonstrates how I²C Master Controller (Master) transmits and receives data to and from the DS 1307 (Slave). So that any low speed peripheral devices can be interfaced using I²C bus protocol as master. In future, this can be implemented as real time clock in networks that contains multiple masters and multiple slaves to co-ordinate the entire system by clock synchronization techniques.

REFERENCES

- [1] Philips Semiconductor "I²C Bus Specification" version 2. 1, January 2000.
- [2] Maxim integrated "DS1307 64 x 8, Serial, I²C Real Time Clock", 2008.
- [3] Prof. Jai Karan Singh "Design and Implementation of I²c master controller on FPGA using VHDL," IJET, Vol 4 No 4 Aug-Sep 2012.
- [4] Raj kamal "Embedded system: Architecture programming and Design", Tata McGraw Hill, 2008.
- [5] Stuart Sutherland, "Verilog® HDL Quick Reference Guide", IEEE Std 1364-2001.
- [6] Tim Wilmshurst, "Designing Embedded Systems with PIC Microcontrollers: Principles and Applications" Elsevier Ltd publications, 2007.
- [7] Xilinx "Spartan-3A/3AN FPGA Starter Kit Board User Guide," version 1.1, 2008.
- [8] A.P.Godse, D.A.Godse, "Microprocessor, Microcontroller & Applications" Technical publications, 2008.
- [9] Vincent Himpe, "Mastering the I²C bus" Elektor Verlag publications, 2011.
- [10] Pong P.Chu, "FPGA Prototyping By Verilog Examples: Xilinx Spartan - 3 Version" Wiley, 2008.
- [11] Mano, Ciletti, "Digital Design" Pearson Education India, edition 4, 2008.
- [12] M.Morris Mano, "Digital Design" EBSCO publishing, Inc., 2002.
- [13] Frank Vahid, "Digital Design with RTL Design, Verilog and VHDL" VP and Executive publisher, 20

Author's Profile:



1) Shaik.Fazil Ahmed received her B.Tech degree in Electronics and Communication Engineering from Audisankara Institute Of Technology, Gudur, S.P.S.R. Nellore District, affiliated to JNTU Anantapur. He is currently pursuing M.Tech VLSI in Audisankara college of Engineering and Technology, Gudur (Autonomous), SPSR Nellore (Dist), affiliated to JNTU Anantapur.



Y.Murali received his M.Tech in VLSI from Vathsalya Institute of Science and Technology, Bhongir, Nalgonda (Dist). He has 6 years and 4 months teaching experience. He is presently working as Assistant Professor in the department of ECE, Audisankara College of Engineering and Technology, Gudur (Autonomous), Affiliated to JNTU, Anantpur