

A Peer Reviewed Open Access International Journal

Design and Implementation of Serial Port Communication by Using I2c Master Controller

B. Kanniram

Master of Technology VLSI Design, TKR College of Engineering and Technology, Medbowli, Meerpet, Saroornagar, Hyderabad.

Abstract:

I²C (Inter-Integrated Circuit), is a multi-master, multislave, single-ended, serial computer bus invented by Philips Semiconductor. It is typically used for attaching lower-speed peripheral ICs to processors and microcontrollers. The focus of this paper is on I2C (Inter-Integrated Circuit) protocol interface between Master Bus protocol and slave. Here we are interfacing between micro-controller and DS1307. I 2C bus protocol sends 8 bit data from micro-controller to DS1307. This module was designed in VHDL and simulated and synthesized using Xilinx ISE Design Suite 14.2. I2C and optimized for area and power. This concept is widely applicable from any high speed device or low speed device to any low speed device or high speed device. This module acts as a slave for the DS1307 at the same time acts like a master for the micro-controller device which can be considered as a slave. 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:

Integrated Cicuits, Xilinx, Micr Controller, VLSI, VHDL.

1.INTRODUCTION:

I²C uses only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), pulled up with resistors. Typical voltages used are +5 V or +3.3 V although systems with other voltages are permitted. The I²C reference design has a 7-bit or a 10-bit (depending on the device used) address space.[3] Common I²C bus speeds are the 100 kbit/s standard mode and the 10 kbit/s low-speed mode, but arbitrarily low clock frequencies are also allowed.

Dr. D. Nageshwar Rao

Professor & HOD, Department of ECE, TKR College of Engineering and Technology, Medbowli, Meerpet, Saroornagar, Hyderabad.

Recent revisions of I²C can host more nodes and run at faster speeds (400 kbit/s Fast mode, 1 Mbit/sFast mode plus or Fm+, and 3.4 Mbit/s High Speed mode). These speeds are more widely used on embedded systems than on PCs. There are also other features, such as 16-bit addressing.



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 International conference on Communication and Signal Processing 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 .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.

Volume No: 2 (2015), Issue No: 6 (June) www.ijmetmr.com



A Peer Reviewed Open Access International Journal

After the START condition, "START" Sequence, "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.

2.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 (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. 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 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.





Volume No: 2 (2015), Issue No: 6 (June) www.ijmetmr.com



A Peer Reviewed Open Access International Journal



Figure 3 Master Receiver Mode

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-shows the interface connection of I2C bus in Spartan 3AN with the DS1307 RTC.



Figure 4 Pin Assignment of DS1307



Figure 5 DS1307 connected to two wire data bus

3.SOFTWARE IMPLEMENTATION:

I2C master controller is designed using Verilog HDL 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 input, determines the next state. There are several states in obtaining the result.

Volume No: 2 (2015), Issue No: 6 (June) www.ijmetmr.com

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).

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 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.

State7: 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).



A Peer Reviewed Open Access International Journal



Figure 6 Flowchart for I2C master bus communication with slave device

4.SIMULATION AND SYNTHESIS RESULTS

The simulated waveforms of the I2C master controller are shown below.







A Peer Reviewed Open Access International Journal

This wave form represents the controlling nature of I2C bus protocol in which the data output would become 1111111 which indicates that the transmitter is completed its data transmission to the slave bus with no erros or misleading of data transmission.in this wave form we can find the control signals would become 1 after each and every test case to indiacate the successful transmission of data.



Figure 8 control signals for I2C master

In the above two diagrams, we can observe the successful data transmission (here oCF) from master to slave in respective to the address given in the master control finite state machine with the name m addr.



Figure 9 Master control waveforms

Here we could observe the wave forms of two signals namely SDA and SCL which are playing major role in data transmission in which both combine as" 01" indicates the start of the data transmission and "10 "indicates the end of data transmission.

5.CONCLUSION:

The result shows that minimal resources are utilized in Spartan 3E. The design process is simplified using VHDL to design the I2C bus protocol, I2C Master (Master) transmits and receives data to and from the DS1307 (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 coordinate the data transmission.

REFERENCES:

[1]Bollam Eswari, N.Ponmagal, K.Preethi, S.G.Sreejeesh, Implementation of I2C Master Bus Controller on FPGA, International conference on Communication and Signal Processing, April 3-5, 2013

[2] I2C Bus Specification, PhilipsSemiconductor, version 2.1, January 2000.

[3] DS1307 64 x 8, Serial, I2C Real Time Clock, Maxim integrated, 2008.

[4] Prof. Jai Karan Singh et al "Design and Implementation of I2C master controller on FPGA using VHDL," IJET, Aug-Sep 2012.

[5] Raj Kamal, "Devices and Communication Buses for Devices Network," in Embedded system: Architecture programming and Design, Shalini Jha Ed. New Delhi, India:Tata McGraw-Hill Education, 2008, pp.160-165.

[6] Tim Wilmshurst, "Starting with Serial," in Designing Embedded.

[7] Spartan-3A/3AN FPGA Starter Kit Board User Guide, Xilinx, version 1.1, 2008.

[8] A.P.Godse, D.A.Godse, "Bus Standards," in Microprocessors and its Applications, 3rd Ed. Pune, India: Technical publications, 2008.

[9] Systems with PIC Micro-controllers: Principles and Applications, 2nd Ed. Burlinton: Newnes, 2009, pp.307-327.



A Peer Reviewed Open Access International Journal

[10] Vincent Himpe, "Historical background of I2C," in Mastering the I2C Bus, Aachen, Germany: Elektor Verlag publications, 2011.

[11] Pong P.Chu, "I/O Modules," in FPGA Prototyping by Verilog Examples: Xilinx Spartan – 3 Version, New Delhi, India: Wiley, 2008.

About Author's:



B. Kanniram

Master of Technology VLSI Design, TKR College of Engineering and Technology, Medbowli, Meerpet, Saroornagar, Hyderabad.

Dr. D. Nageshwar Rao

Professor & HOD, Department of ECE, TKR College of Engineering and Technology, Medbowli, Meerpet, Saroornagar, Hyderabad.