Design of I2C Single Master Using Verilog

Article *in* International Journal of Science and Research (IJSR) · May 2015

CITATIONS
0

READS
7,121

2 authors, including:

Shivani Mehrotra
Amity University
2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Design of I2C Single Master Using Verilog

Shivani Mehrotra[1], Nisha Charaya[2]

[1]M.Tech (ECE), Amity University Gurgaon (Haryana), India

[2]Assistant Professor, Amity University Gurgaon (Haryana), India

Abstract: ***This paper focuses on the design of I2C single master which consists of a bidirectional data line i.e. serial data line (sda) and serial clock line (scl).This protocol can support multiple masters.I²C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices and is used for faster devices to communicate with slower devices and each other without data loss. It requires only two lines for communication with two or more chips and can control a network of device chips with just two general purpose I/O pins whereas, other bus protocols require more pins and signals to connect devices. The complete module is designed in Verilog and simulated in ModelSIM.***

Keywords: Verilog, ModelSIM, I2C bus, Master, Slave, SDA, SCL.

1.Introduction

In the world of serial data communication, 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 connections in the IC (Integrated Circuit) for serial data communication to take place. But as the physical size of IC have decreased over the years, a less amount of pin connection for serial data transfer is required. 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 protocol was introduced by Phillips[2] [3] [10]

2.I2C Protocol

It is most suitable for applications requiring occasional communication over a short distance between many devices. The I2C standard is a communication protocol including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously. The component reads from and writes to user logic over a parallel interface. Resource requirements depend on the implementation.

The two lines of the I2C-bus, SDA and SCL, are bi-directional and open-drain, pulled up by resistors.SCL is a Serial Clock line, and SDA is a Serial Data line. Devices on the bus pull a line to ground when a logical zero is to be sent and release a line when a logical one is to be sent. Each device is recognized by a unique address — whether it's a microcontroller, LCD driver, memory or keyboard interface — and can operate as either a transmitter or receiver, depending on the function of the device [2].Obviously an LCD driver is only a receiver, whereas a memory can both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers. A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

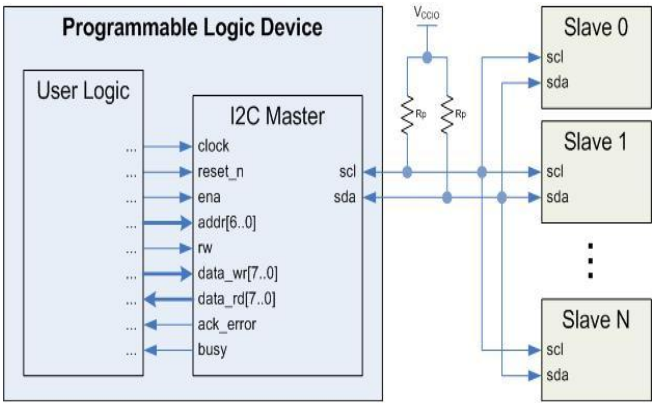

Figure 1: Logical Circuit Diagram of I2C Master – Slave

Features

- data transfers: serial, 8-bit oriented, bi-directional
- master can operate as transmitter or receiver
- bit transfer (level triggered)
- SCL = 1 , SDA = valid data
- one clock pulse per data bit
- stable data during high clock
- data change during low clocks [6]

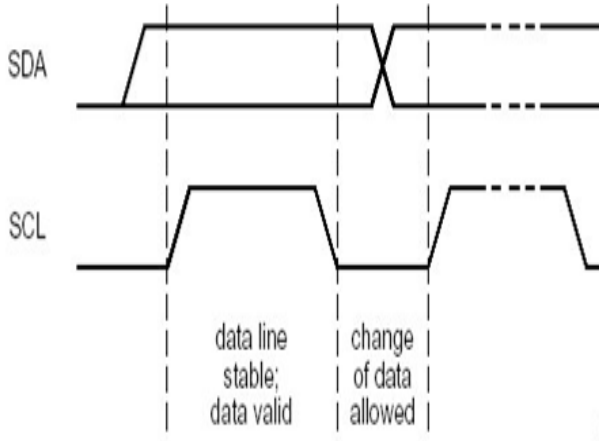

Figure 2: Change of word occuring during low clock

- start condition (S)
  SDA 1 to 0 transition when SCL = 1

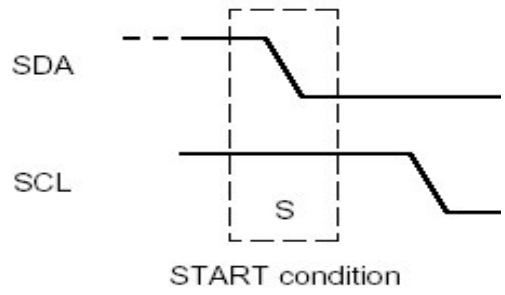

Figure 3: Start Condition

- stop condition (P)
  SDA 0 to 1 transition when SCL = 1

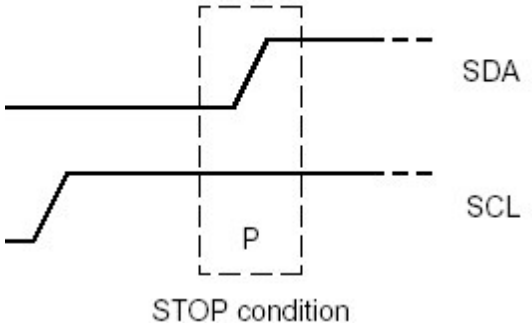

Figure 4: Stop Condition

I2C Master:

- controls the SCL
- starts and stops data transfer
- controls addressing of other devices

I2C Slave

- Device addressed by master I2C single Master works as a transmitter or a receiver.
- Master as transmitter sends data to slave-receiver (RW=0 i.e write state)
- Master as receiver requires data from slave-transmitter(RW=1 i.e read state) [6]

3. Design Methodology

Finite State Machine (FSM) that describes the design of single Master is shown in Figure 5.

Algorithm:

State 1: ***Ready condition:*** I2C bus doesn't perform any operation.(SCL and SDA remains high) and enable is low. If *ena* becomes HIGH it enters into next state.

State 2: ***Start condition:*** When *ena* is HIGH, Master initiates data transmission by entering into the next state *adr*.

State 3: ***adr*** **state:** In this next *adr* state, master sends the slave address serially (11010000) to the slave. *bit_cnt* is used as counter to count the bits of address transferred and as it becomes 0,it enters into next state.

State 4: ***ack*** **state:** If the slave address matches with the slave(here single slave is considered hence no need to match it as it is taken as state) it sends an acknowledgement bit in response to the master.

Now R/W bit is checked if it is LOW, it enters *write* state else *read* state.

State 5: ***Write state:*** The 8 bit data to be transmitted is sent to the slave by the master. After receiving the data, slave acknowledges the master.

State 6: ***Read state:*** The 8 bit data is read from the slave by the master. After reading the data, acknowledgement is sent.

State 7: ***Stop condition*****:** After the transmission of the data, STOP bit is sent.(SCL is high and SDA is from Low to high).

Master sends a STOP bit to terminate the connection. Again *ena* is checked if it is still LOW it remains in STOP state else it enters the READY state. For performing read operation, write operation is performed first and then read operation is done. Slave address used is of 3 bit (010).

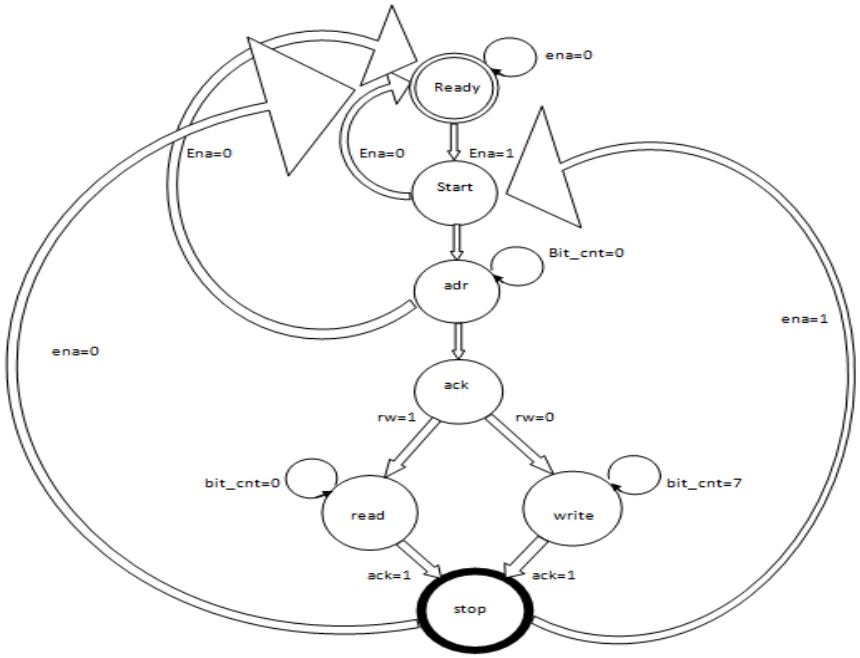

Figure 5: Finite State Machine For Design of Single Master

4.Functional Description

The functional description of I2C master is described in the Verilog HDL. That is called design module. The test bench program is developed to test the design module. The test bench gives the input to the design module & verifies the output. The test bench is written in such way that the design module can be checked in all possible conditions. The signals given from the test bench to test the working of the prototype design of the I^2C master are:

- ena
- ack
- en
- in_data
- sda_r

5. Simulation Results

Results

The I2C single master has been designed in Verilog and simulated in simulator tool Modelsim which is used to verify the design functioning. This design works for both read and write cycle.

1) Figure 6, shows the data is written from the register 'in_data' serially through the 'sda' to register 'out_data'
2) Figure 7, shows the data that is written is now read serially through the 'sda' and is shown at the output through register 'read_out'
3) Figure 8, shows the combined simulation result of write and read cycle.

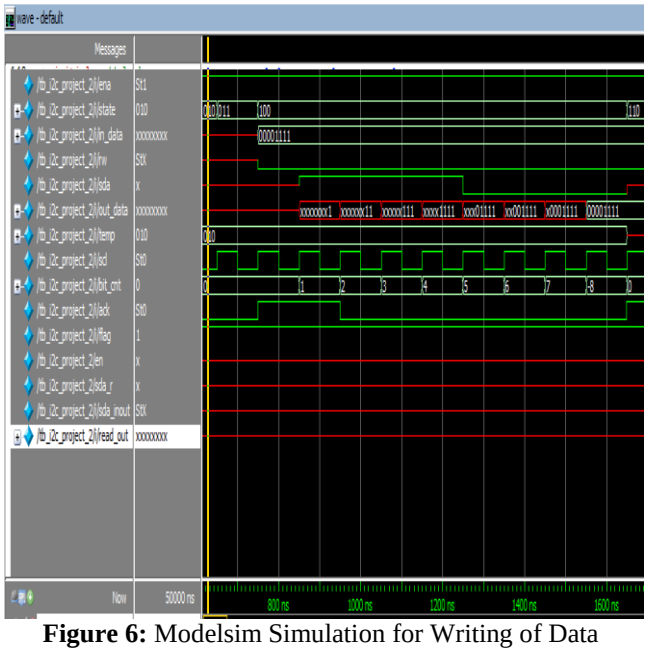

Figure 6: Modelsim Simulation for Writing of Data

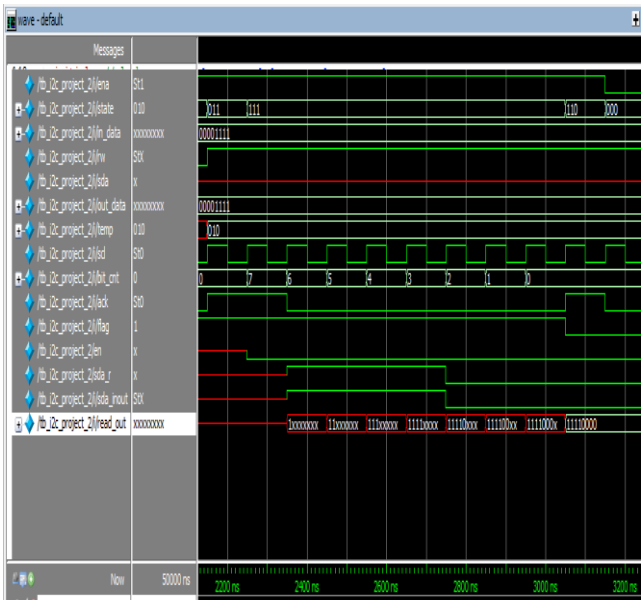

Figure 7: Modelsim Simulation for Reading of Data

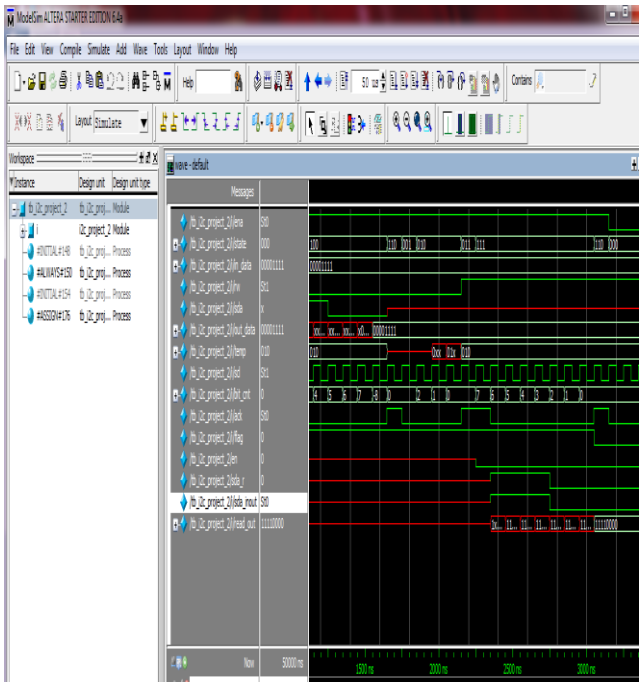

Figure 8: Modelsim Overall Simulation

6. Results

The data transmitted by master is successfully stored and read by it.

7. Conclusion & Future Scope

I2C Single Master is successfully designed and simulated. As the number of devices connected to a system is going to increase, there is a need for a system which supports multiple protocols. This project can be further extended to design for multiple masters.

References

[1] Samir Palnitkar ,"Verilog HDL A guide to Digital Design and Synthesis" SunSoft Press,1996

[2] Stuart Sutherland, "Verilog HDL Quick Reference Guide", IEEE Std 1364-2001
[3] M.Morris Mano, "Digital Design" EBSCO publishing. Inc., 2002
[4] Philips Semiconductor "I2C Bus Specification", April 1995
[5] Philips Semiconductor "I2C Bus Specification" version 2. 1, January 2000
[6] Tomáš ášMatoušek "I2C bus Inter Integrated Circuits bus Integrated Circuits bus by Philips Semiconductors"
[7] O. Romain, T.Cuenin & P.Garda: "Design &modeling of an I2c Bus Controller", FDL 0'3,Frankfurt, Deutschland, Sept 23-26,2003
[8] Prof. Jai Karan Singh, Prof. Mukesh Tiwari,Vishal Sharma "Design and Implementation of I2c master controller on FPGA using VHDL" International Journal of Engineering and Technology (IJET),ISSN : 0975-4024 Vol 4 No 4,Aug-Sep 2012
[9] Pankaj Kumar Mehto, Pragya Mishra, Sonu Lal "Design and Implementation for Interfacing Two Integrated Device Using I2C Bus" IJRICCE, ISSN(Online): 2320-9801, Vol. 2, Issue 3, March 2013
[10] Bollam Eswari, N.Ponmagal, K.Preethi, S.G.Sreejeesh "Implementation of I2C Master Bus Controller on FPGA" in IEEE, International conference on Communication and Signal Processing, April 3-5, 2013
[11] Mr. J. J Patel, Prof B. H. Soni, "Design And Implementation Of I2c Bus Controller Using Verilog" in Proc. Journal Of Information, Knowledge And Research In Electronics And Communication Engineering ISSN: 0975 – 6779, VOLUME – 02, ISSUE – 02, NOV 12 TO OCT 13
[12] Ashwini s. Tadkal, Padmapriya Patil, "DESIGN OF DUAL MASTER I2C BUS CONTROLLER" IJRET, Volume: 03,Special Issue:03, May-2014