ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 9, December 2013

The Performance and Optimization of I2C Protocol using Verilog on Xilinx Tool

Narendra Gupta[1], Vipin Gupta[2]

PG scholar, Electronics & Communication Engineering Department, S.G.V University Jaipur, Rajasthan, India[1]

Assistant Professor, Electronics & Communication Engineering Dept, S.G.V University Jaipur, Rajasthan, India[2]

Abstract: The aim of this paper to optimize the I2C protocol which is one of the most popular serial data transfer protocol and used widely, design by Philips Semiconductors. The I2C protocol has many qualities such as controllability, less wire connection and low power consumption. By employing the I2C protocol in a design, much of the auxiliary support circuitry such as address decoders and standard logic gates needed for other communication methods can be removed.

I. INTRODUCTION

In the world of multiple application based product it is very much a mandatory to have multiple devices connected to a system, this includes peripherals following different communication protocols as well. This requirement gives rise to the need for an intermediate system which can act as a bridge between 2 devices following different communication protocols. This is where I2C master controller design is very useful which is developed for bus by Philips to maximize hardware efficiency and circuit simplicity. I2C has several operation modes but to specific application only parts of functions will be useful. If all functions were included, the cost and power dissipation are increase. Today systems is connected to a number of devices and make the communication smooth and faster, I2C protocol is considered as one of the very best. For the controllability, interface of each device is designed dedicatedly. The I2C is appropriate for interfacing to devices on a single board, and can be stretched across multiple boards inside a closed system, but not much further.

II. CONCEPT OF I2C PROTOCOL

The I2C is a two-wire serial bus, as shown in Figure 1. There's no need for chip select or arbitration logic, making it cheap and simple to implement in hardware.The two I2C signals are serial data (SDA) and serial clock (SCL). Together, these signals make it possible to support serial transmission of 8-bit of data-7-bit device addresses plus one control bit over the two-wire serial bus. The device that initiates a transaction on the I2C bus is termed the master. The master normally controls the clock signal.

A device being addressed by the master is called a slave. In a bind, an I2C slave can hold off the master in the middle of a transaction using what's called clock stretching (the slave keepsSCL pulled low until it's ready to continue). Most I2C slave devices don't use this feature, but every master should support it.

The I2C protocol supports multiple masters, but most system designs include only one. There may be one or more slaves on the bus. Both masters and slaves can receive and transmit data bytes. Each I2C-compatible hardware slave device comes with a predefined device address, the lower bits of which may be configurable at the board level. The master transmits the device address of the intended slave at the beginning of every transaction. Each slave is responsible for monitoring the bus and responding only to its own address. This addressing scheme limits the number of identical slave devices that can exist on an I2C bus without contention, with the limit set by the number of user-configurable address bits (typically two bits, allowing up to four identical devices).

III. IMPLEMENTATION OF I2C PROTOCOL

In this chapter discuss final results of the I2C protocol, which is used to design hardware of the below mention figure2. In I2C protocol, we are design hardware which is containing flowing section

1. Bit Command Controller
2. Byte Command Controller

Bit Command Controller: The Bit command controller are used to transmit serial data from parallel data and serial clock, it is also generate status signals like busy, command acknowledgement etc. The bit command controller generates serial data based on input command. The main purpose of bit command controller is to handle actual transmission of data and the generation of control signal to control SCL & SDA line. The Byte command controller tells bit command controller which operation has to be performed. Byte Command Controller: The Byte Controller is transmitting parallel data to bit controller. It does also generate busy signal, acknowledgement signal and other status signal. The byte controller handles I2C traffic at the level. It takes data from command register & translates; it's into sequence based on transmitter of a single bit.

IV. DESIGN OF I2C MASTER CONTROLLER

The Master Controller contains Bit controller, Byte controller, Interfacing Registers, Clock divider etc. The I2C Top module generates serial clock, serial data and status signals. It is receive data and commands from external source. Shift Register are contain some registers like Data I/O shift register, Command Register, Status Register, Transmitter Register, Receive Register, Prescaler Register.

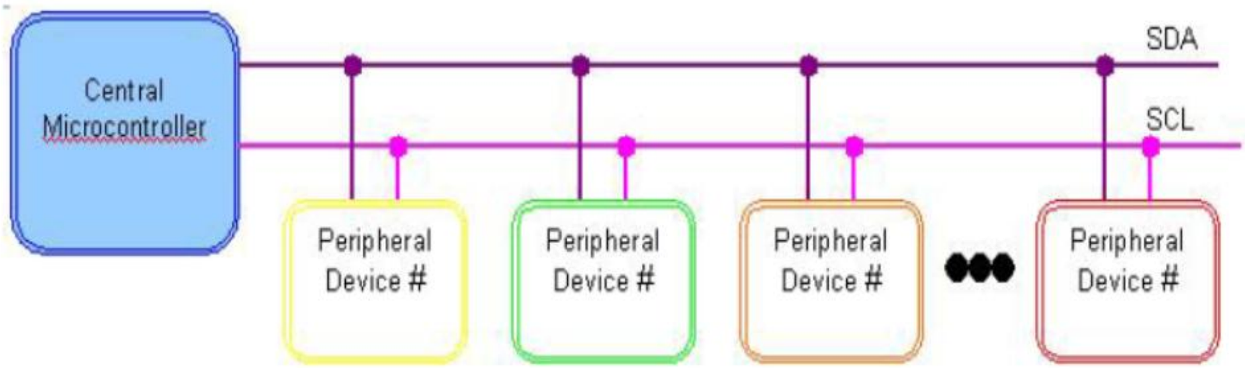


Figure 1 I2C has two lines in total

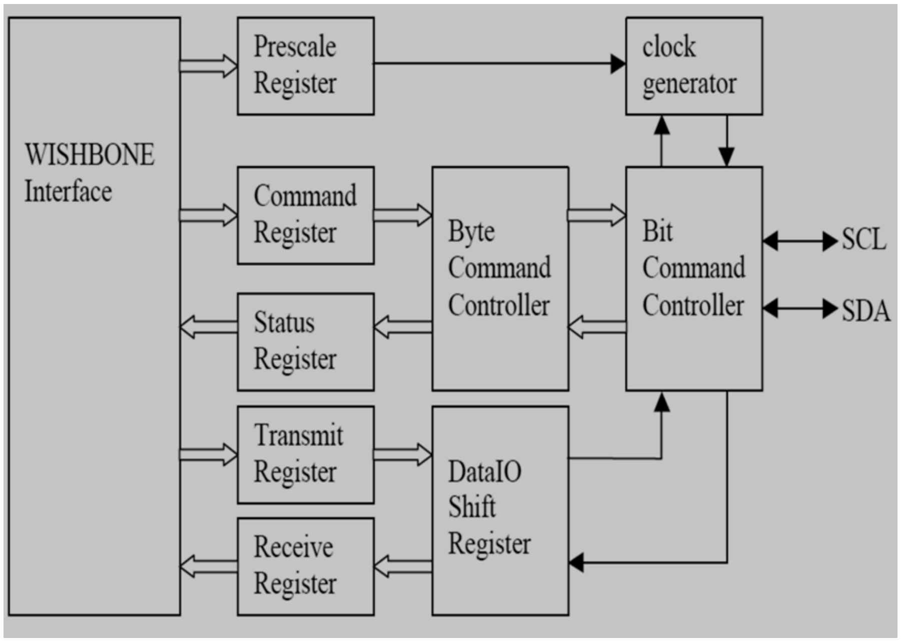


Figure2 Internal structure of I2C Master Core

These register except data I/O shift register are used for interfacing or for storing temporary data. Data I/O Shift Registers contain the data associated with the current transfer during read operation the data is shift from the SDA line. After a byte has been read contains are copied into the receive register. During a write action the transmitter registers contain are copied into the data I/O shift register and are than transmitted on two the SDA.

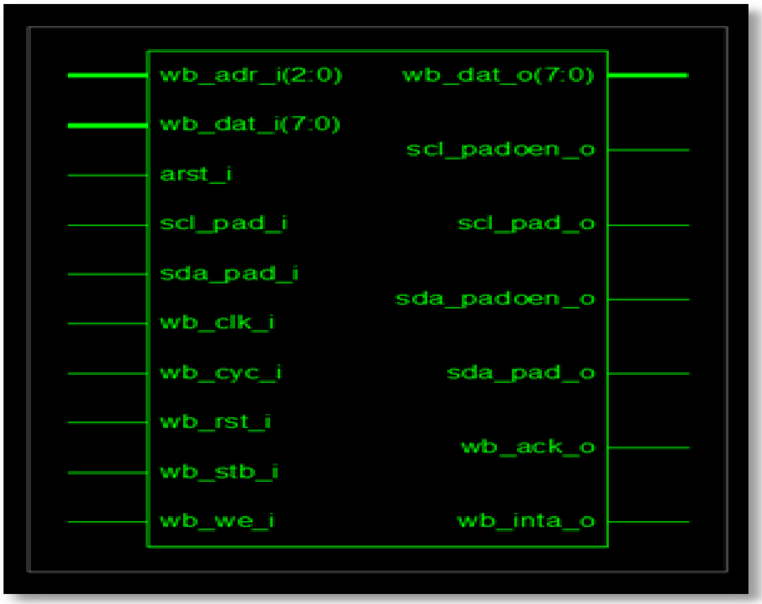


Figure 3 Top Module for I2C Bus

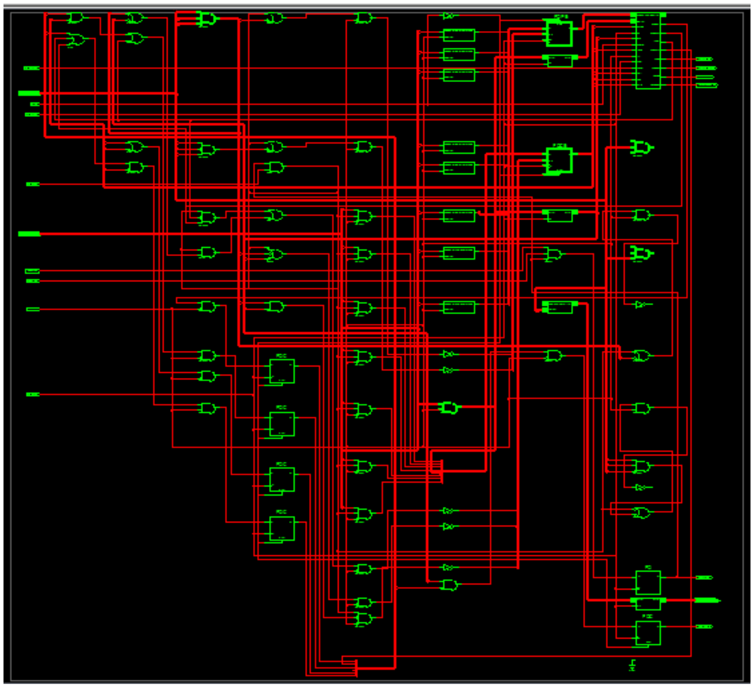
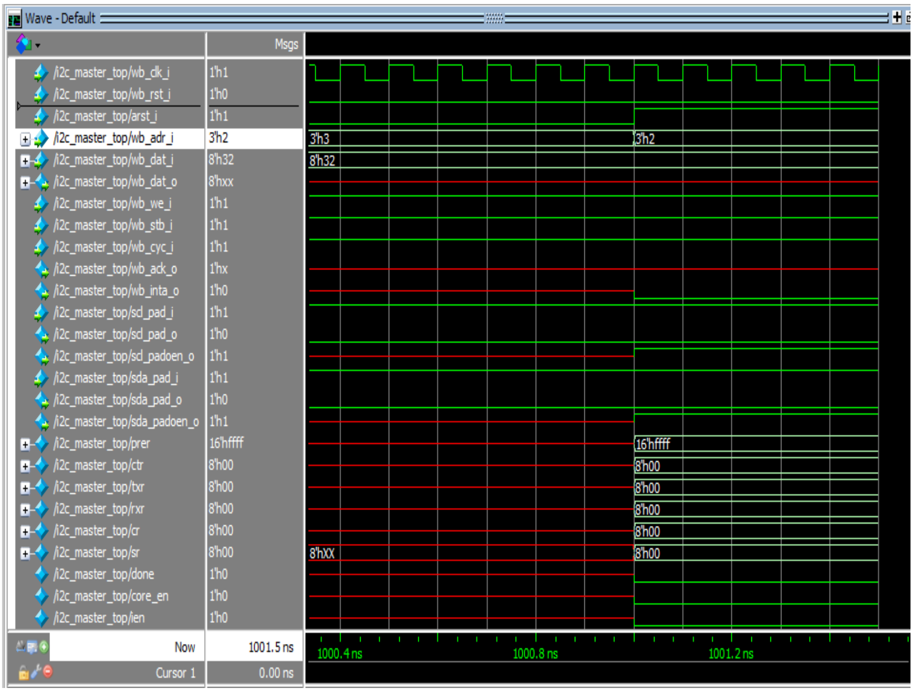


Figure 4 RTL of Top Module

Perscaler type Register is used to generate clock signal. Command Register is used to receive status signal. The transmit register are used to transmit data to data I/O. Shift register the receive register are used to receive register from data I/O shifter register. The prescaler register is used to synchronous system clock & SCL. The shift register also used to interface external parallel data in I2C (WISHBONE compliant core).

The Top Module of I2C Master controller is simulated in Mentor Graphics ModelSim. The results are obtained using I2C master Core Top Module. The result shows serial clock and serial data at output. It’s also generating status signal for next data processing. Its means that it provide busy and ideal condition of the network.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 9, December 2013


Wave - Default

	Msgs
/i2c_master_top/wb_clk_i	1'h1
/i2c_master_top/wb_rst_i	1'h0
/i2c_master_top/arst_i	1'h1
/i2c_master_top/wb_adr_i	3'h2
/i2c_master_top/wb_dat_i	8'h32
/i2c_master_top/wb_dat_o	8'hxx
/i2c_master_top/wb_we_i	1'h1
/i2c_master_top/wb_stb_i	1'h1
/i2c_master_top/wb_cyc_i	1'h1
/i2c_master_top/wb_ack_o	1'hx
/i2c_master_top/wb_inta_o	1'h0
/i2c_master_top/scl_pad_i	1'h1
/i2c_master_top/scl_pad_o	1'h0
/i2c_master_top/scl_padoen_o	1'h1
/i2c_master_top/sda_pad_i	1'h1
/i2c_master_top/sda_pad_o	1'h0
/i2c_master_top/sda_padoen_o	1'h1
/i2c_master_top/prer	16'hffff
/i2c_master_top/ctr	8'h00
/i2c_master_top/txr	8'h00
/i2c_master_top/rxr	8'h00
/i2c_master_top/cr	8'h00
/i2c_master_top/sr	8'h00
/i2c_master_top/done	1'h0
/i2c_master_top/core_en	1'h0
/i2c_master_top/ien	1'h0
Now	1001.5 ns
Cursor 1	0.00 ns

Figure 5 Simulated Results of Top Module

V. CONCLUSION

The I2C protocol is in order to allow faster devices to communicate with slower devices and also allow devices to communicate with each other over a serial data bus without data loss. The I2C protocol has many merits such as controllability, less wire connection. I2C protocol also can be shared by more than one device. In this paper the I2C protocol can try to design a chip using Verilog code and optimize the requirement of the overall performance like Delay, Complexity, Power and Area.

REFERENCES

[1] The I2C-bus,THE I2C-BUS SPECIFICATION, VERSION 2.1, JANUARY 2000, Philips Semiconductors, document order number: 9398 393 40011.

[2] Zheng-wei HU, “I2C Protocol Design for Reusability”, Third International Symposium on Information Processing, 978-0-7695-4261-4/10 © 2010 IEEE.

[3] Xavier Righetti, Daniel Thalmann, “Proposition of a Modular I2C-Based Wearable Architecture”, 978-1-4244-5795-3/10 ©2010 IEEE.

[4] Application Note AN2318/D Rev. 0.8/2002 Using I2c Bus With HCS12 Microcontrollers , by Micro Grant M More Applications Engineering Freescale, East Kilbride.

[5] Jason Oberg, Wei Hu,Ali Irturk,Mohit Tiwari and Ryan Kastner, "Information Flow Isolation in I2C and USB",ACM 978-1-4503-0636-2/11/06.

[6] Prof. Jai Karan Singh,Prof. Mukesh Tiwari, "Design and Implementation of I2c master controller on FPGA using VHDL",International Journal of Engineering and Technology, ISSN : 0975-4024 Vol. 4 No 4 Aug-Sep 2012 162.