

## Design & Implementation of I2C Master Controller Interfaced With RAM Using VHDL

Sansar Chand Sankhyan

M.tech(VLSI), Department of ECEITM University, Gurgaon, Haryana, India

### ABSTRACT

In most of the applications, the physical systems require a real-time operation to interface high speed constraints. The Inter Integrated Circuits (I2C) is a 2-wired communication bus. Physically, it consists of 2 active wires: SDA (Serial Data), SCL (Serial Clock) and a ground connection. All I2C-bus compatible devices incorporate an on-chip interface which allows them to communicate directly with each other via the I2C-bus.

This paper focuses on the software implementation for I<sup>2</sup>C Driver and its interfacing with RAM. Specifically, this paper describes in detail an I<sup>2</sup>C Master connected to I<sup>2</sup>C Slave using an I<sup>2</sup>C bus. The I<sup>2</sup>C protocol was given by Philips Semiconductors for faster devices to communicate with slower devices and each other without data loss. The complete module is designed in VHDL and simulated in Xilinx ISE 14.5.

**KEYWORDS:** SCL, SDA, I<sup>2</sup>C Bus, RAM

### I. INTRODUCTION:

The design of I<sup>2</sup>C bus controller is to interface between two devices for the purpose of communication. The presence of an on-chip RAM support for a data communication between the chip and a host computer to send data on the RAM. To minimize the system-level interconnect between integrated devices, I propose to transmit the contents of the RAM storing the results back to a host computer via a serial bus, the I<sup>2</sup>C Interface. The interconnect integrated circuit bus commonly known as the I2C bus which is a bi-directional, two-wire and serial communication standard protocol. The designed are simple but difficult for efficient integrated circuit (IC) control. The system is comprised of two bus lines, SCL (Serial Clock) and SDA (Serial Data). Standard I<sup>2</sup>C devices operate up to 100Kbps, while fast-mode devices operate at up to 400Kbps. Most of the I<sup>2</sup>C devices available today support 400Kbps operation.

### II. LITERATURE SURVEY:

There are many reasons for using serial interface design much more important application includes serial communication like sensors communication with personal computer. Many common embedded system peripherals, such as analog-to-digital and digital-to-analog converters, LCDs, and temperature sensors, support serial interfaces. Serial interface allow processors to communicate without the need for shared memory and the problems they can create. There are Serial communication protocols like UART, CAN, USB, SPI, Inter IC. USB, SPI and UARTS are all just one type to point type protocol. USB uses multiplexer to communicate with other

devices. Only I<sup>2</sup>C and CAN protocol uses software addressing. But only I<sup>2</sup>C is very simple to design and easy to maintain.

Table 1. Comparison of different protocols

	UART	CAN	USB	SPI	I <sup>2</sup> C
PRONS	Well known Simple	Secure fast	Secure, fast, plug and play	Fast, low cost, universally accepted, large portfolio	Simple, plug and play, cost effective, universally accepted
CONS	Limited functionality, Point to point	Complex, limited portfolio, Automotive oriented	Powerful master required, No plug and play software, Extra drivers required	No plug and play hardware, No fixed standard	Limited no. Of components due to capacitance effect

### III. I2C BUS TERMINOLOGY:

- **Transmitter:** The device which sends the data or message to the receiver.
- **Receiver:** The device which receives the data or message from the bus.
- **Master:** The device which generates clock signals, initiates a transfer i.e. start condition and terminates a transfer i.e. stop condition. It may also act as a transmitter/receiver.
- **Slave:** The device addressed by a master. Also it may act as a transmitter/receiver.
- **Multi master:** More than one master can attempt control the bus at the same time without demeaning the data/message.
- **Arbitration:** It is a procedure to ensure that, if more than one master simultaneously tries to

control the bus, only one is allowed to do so and the data is not corrupted.

- **Synchronization:** It is a procedure to synchronize the clock signals of two or more devices.

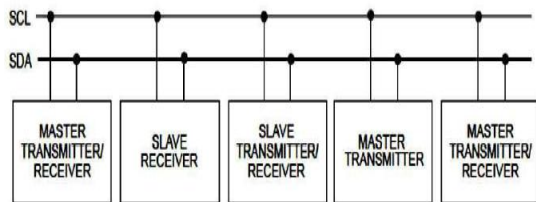


Figure 1. I<sup>2</sup>C bus configuration using masters and slaves

**3.1 Master and Slave:**

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. Both lines are connected to a positive supply via a pull-up resistor, and remain HIGH when the bus is not busy. Each device is recognized by a unique address—whether it is a microcomputer, LCD driver, memory or keyboard interface—and can operate as either a transmitter or receiver, depending on the function of the device. A device generating a message or data is a transmitter, and a device receiving the message or data is a receiver. Obviously, a passive function like an LCD driver could only be a receiver, while a microcontroller or a memory can both transmit and receive data. When a data transfer takes place on the bus, a device can either be a master or a slave. The device which initiates the transfer, and generates the clock signals for this transfer, is the master. At that time any device addressed is considered a slave. It is important to note that a master could either be a transmitter or a receiver; a master microcontroller may send data to a RAM acting as a transmitter, and then interrogate the RAM for its contents acting as a receiver in both cases performing as the master initiating the transfer. In the same manner, a slave could be both a receiver and a transmitter. The I<sup>2</sup>C is a multi master bus. It is possible to have, in one system; microcontroller may act as a master for one transfer, and then be the slave for another transfer, initiated by another processor on the network. The master/slave relationships on the bus are not permanent, and may change on each transfer.

**3.2 Data Validity:**

One data bit is transferred must remain stable during the HIGH period of the clock pulse in order to be valid. Changes in the data line at this time will be interpreted as eight bits long, is transferred serially

with the most significant bit first, and is followed by an acknowledge bit. The clock pulse related to the acknowledge bit is generated by the master.

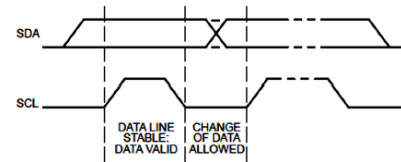


Figure 2 Data validity condition

**3.3 Byte format Data Transfer:**

Every byte put on the SDA line must be 8 bits transmitted per transfer is unrestricted. Each byte has to be followed by an Acknowledge bit. Data is transferred with the Most Significant Bit (MSB) first. If a slave cannot receive or transmit another complete byte of data until it has performed other function, for example servicing an internal interrupt, it can hold the clock line SCL LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL

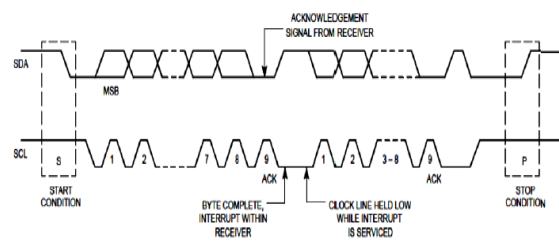


Figure 3 Data transfer on the I<sup>2</sup>C bus [7]

**IV. I2C Protocol:**

Normally, a standard communication protocol consists of four parts:

- 1) START signal generation
- 2) Slave address transfer
- 3) Data transfer
- 4) STOP signal generation

**4.1 Transmitting a byte to a slave device:**

After start condition has been sent, a byte can be transmitted to a slave by the master. This first byte after a start condition will identify the slave on the bus (address) and will select the mode of operation. The meaning of all following bytes depends on the slave.

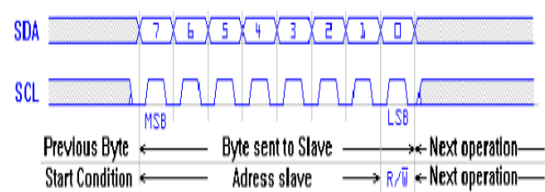


Figure 4 Transmission of a byte to a slave [8]

**4.2 Receiving a byte from a slave device:**

Once the slave has been addressed and the slave has acknowledged this, a byte can be received from the slave if the R/W bit in the address was set to READ (set to '1').

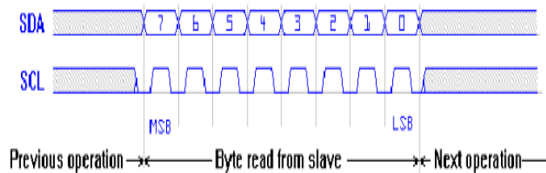


Figure 5 Reception of a byte from a slave [8]

**4.3 Design steps:**

The functional description of I<sup>2</sup>C master has to be described in the VHDL language. That is called design module / core. The test bench program has to be developed to test the design module. The test bench gives the input to the design module & verifies the outputs. The test bench has to be written in such way to check the design module in all possible conditions.

**V. SYSTEM ARCHTECTURE:**

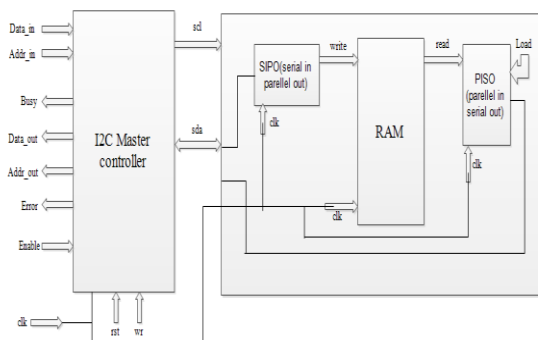


Figure 6 I2C and RAM interface system architecture

In this paper we are considering master as I<sup>2</sup>C controller and slave as SDRAM.

**5.1 Top-Level Signal Descriptions:**

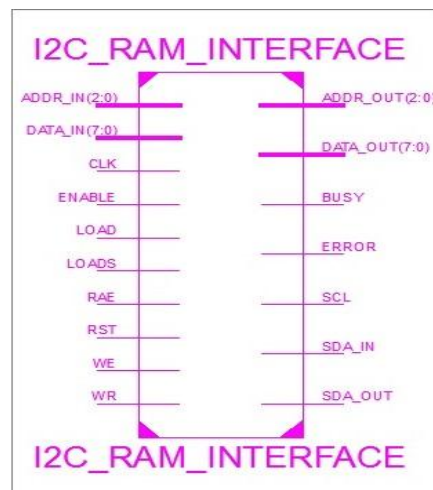


Figure 7 RTL schematic of designed I2C master controller

Table-1 provides descriptions of the input/output signals of the I<sup>2</sup>C bus Master controller. The address bus is a 3-bit input pin to the I<sup>2</sup>C master controller. The data input to the master controller is 8-bit and data output from the master controller is also 8 bit. Serial data line (SDA) and Serial clock line (SCL) both are out pins between the master controller and slave devices. RD and WR both are input pins to the master controller from microprocessor.

Table 2. Signal Description of I2C Master Controller [7]

Signal	Type	Description
ADDR[2:0]	Input	Address bus from microprocessor
DATA[7:0]	In-out	Microprocessor Data bus
SDA	In-out	I <sup>2</sup> C serial data
SCL	In-out	I <sup>2</sup> C serial clock
CS	Input	I <sup>2</sup> C chip select from microprocessor
RD	Input	I <sup>2</sup> C read signal from microprocessor
WR	Input	I <sup>2</sup> C write signal from microprocessor
CLK	Input	Input clock from microprocessor

**5.2 Process Algorithm:**

The complete process of data transfer from master to slave and vice-versa is shown below in Figure 7 in form of an algorithm.

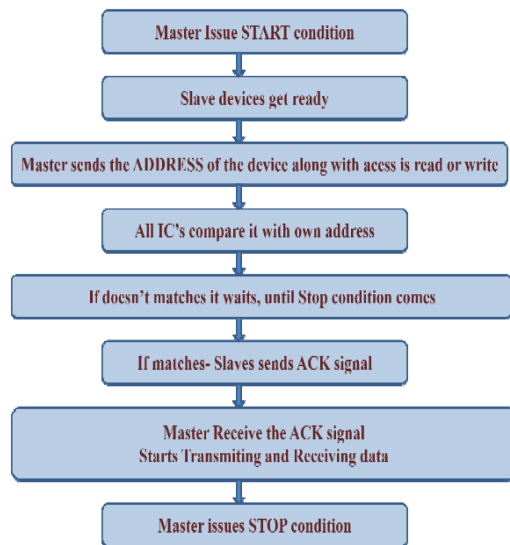


Figure 8 Process flow of I2C controller [8]

### 5.3 Advantages OF I<sup>2</sup>C

1. Used for security sensitive applications like sensor connections, RFID, biometric devices, etc
2. Common communication standards between microcontrollers and sensors
3. Each device is recognised by its unique address and can operate as either a transmitter or receiver, depending upon the function of the device.
4. It Provides enhance security system
5. Compatible with FPGA

## VI. SIMULATION RESULTS:

The VHDL code for I2C master controller is compiled in the Xilinx ISE 14.5 software tool. The test bench of this module is simulated and followings are the results.

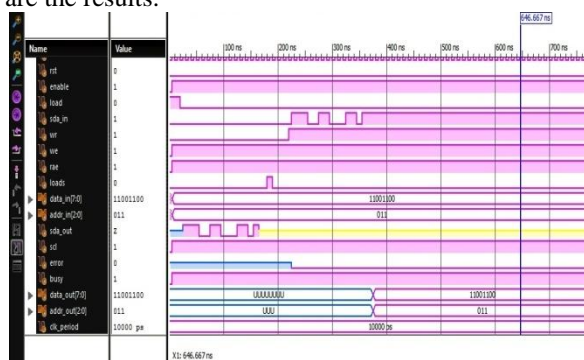


Figure 9 Simulation result of write and read operation in I2C and RAM interface

## VII. CONCLUSION:

The simulation results showed that all the designed functional blocks of the system design are giving desired results and the system itself also giving the desired results. The interfacing done

between the I2C master controller and RAM slave is successfully designed and is giving desired results. The design module is designed using is done using VHDL. If the designer wants to design the module for any new technology than he only needs give the design to the logic synthesis tool as input and in result it will create the new gate level net list and the tool itself optimize the timing and area for new technology.

## References:

- [1] The I2C -Bus specifications version 2.1 January,2000 <http://www.semiconductors.philips.com/buses/i2c>.
- [2] UM10204 I2C -bus specification and user manual Rev. 03 — 19 June 2007
- [3] I2C bus Inter Integrated Circuits bus by Philips Semiconductors.
- [4] Embedded Platform Excalibur Altera, <http://www.altera.com>.
- [5] PCF 8584, I2C bus controller datasheet, <http://www.semiconductor.philips.com>
- [6] [www.latticesemi.com](http://www.latticesemi.com), I2C Master Controller reference design
- [7] Arvind Sahu, Ravi Shankar Mishra, Puran Gour, "Design and Interfacing of High speed model of FPGAs using I2C protocol", Int.J.Com.Tech.Appl.vol2(3),531-536
- [8] Shoaib.Shah Sobhan, Sudipta. Das and Iqbalur. Rahman, "Implementation of I2C using System Verilog and FPGA", ICAEPE, 2011.
- [9] Jayant Mankar et. al, "Review of I2C Protocol," International Journal of Research in Advent Technology, vol. 2, issue 1, January 2014, pp.474-479.
- [10] A.K. Oudjida, M.L. Berrandjia, R. Tiar, A.Liacha, K. Tahraoui, with "FPGA Implementation of I2c & SPI Protocols: A Comparative Study" Electronics, Circuits, and Systems, 2009. ICECS.