American International Journal of Research in Science, Technology, Engineering & Mathematics

Available online at http://www.iasir.net

ISSN (Print): 2328-3491, ISSN (Online): 2328-3580, ISSN (CD-ROM): 2328-3629

AIJRSTEM is a refereed, indexed, peer-reviewed, multidisciplinary and open access journal published by International Association of Scientific Innovation and Research (IASIR), USA

(An Association Unifying the Sciences, Engineering, and Applied Research)

An Efficient Designing of I2C Bus Controller Using Verilog

[1]Deepa Kaith, [2]Dr. Janakkumar B. Patel, [3]Mr. Neeraj Gupta

[1]Student M.Tech., [2]Professor, [3]Assistant Professor

ECE Dept., ASET

Amity University Haryana

***Abstract**: This paper focus on the efficient designing of Inter Integrated Circuit (I2C) master controller. The design follows I2C specifications for address sending and data transfer operations. The design is software based that provides easy up-gradation of the whole electronic system by simply clipping or unclipping of the desired IC. It is equipped to ensure no data loss and bus conflicts with multiple device connections on an I2Cbus. The paper demonstrates how I2C Master Controller can achieve appropriate data transmission without collapse. The master controller is designed in Verilog and is simulated in ModelSim. For synthesizability the design is further implemented in Xilinx XST 14.1.*

***Keywords**: Inter integrated circuit, master, slave, clock stretching, arbitration, serial clock, serial data*

I. Introduction

The different integrated electronic designs contain many similar modules even if they are not related. This includes an intelligent controller, data I/O ports, non-volatile and volatile memories, converters, tuning circuits and LCD drivers etc. The development of various wireless or wired communication mechanisms has made it possible to integrate different parts of design together [10]. The serial communication reduces the cost of connection and the number of IC pins. Inter Integrated Circuit (I2C) is one of the serial wired communication protocol developed by Philips semiconductors. I2C as the name suggest provides the efficient inter IC-control [1] with simple circuitry.

I2C is a simple two-wire bus having fewer IC connection pins that makes PCBs smaller, less expensive and also reduces the required area. It also provides various other benefits like on-chip bus interface, synchronization, software based addressing and data transfer, reusability of modules and easy fault diagnosis to ensure no data loss. I2C bus also allows easy clipping or unclipping of ICs for modification of the whole system. It supports every fabrication technology like bipolar, CMOS, NMOS etc [1]. Originally, I2C bus was limited to the speed of 100 kbps. Now, it has been upgraded to five speed categories using both bidirectional and unidirectional bus.

i. Using bidirectional bus:

a) Standard mode (SM): bit rate upto 100 kbit/sec.

b) Fast mode (FM): bit rate upto 400 kbit/sec.

c) Fast mode Plus (FM+): bit rate upto 1 Mbit/sec.

d) High speed mode (HS): bit rate upto 3.4 Mbits/sec.

ii. Using unidirectional bus:

a) Ultra Fast mode (UFM): bit rate upto 5 Mbit/sec and is not compatible with the previous versions due to unidirectional nature.

II. I2C PROTOCOL

I2C is a multi-master, short distance serial communication protocol. The communication among various ICs is carried using only two serial bi-directional lines: serial clock (SCL) and serial data (SDA). The pull-up resistors are only required additionally for each of the lines.

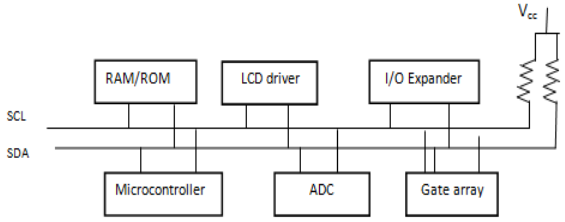


Fig: 1. Device connection environment using I2C bus configuration

Every device connected by the bus (as in fig.1) is recognized by a unique address and can act as a master or a slave depending on the functionality. The device generating the clock signals and initiating the data transfer is a *master* and the addressed device is a *slave* [6]. The devices can also be divided on the basis of data transfer. The data sending device is a *transmitter* and the data receiving device is a *receiver*. There can be multiple master devices connected to the bus and at a time only one is allowed to take control of the bus through *arbitration*.

The serial lines SCL and SDA are connected to the pull-up resistor or positive power supply through a current source. Both lines are high, when the bus is idle. The output of devices must have an open-drain/collector for wired AND function. The bus capacitance determines the number of interfaces connected to the bus which is upto 400 pF [1].

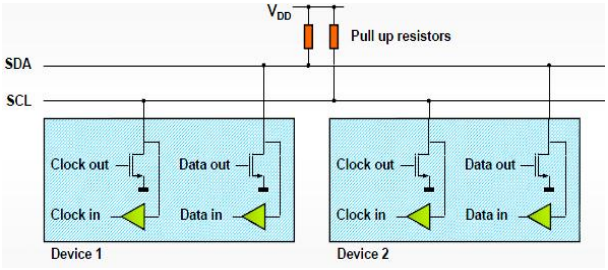


Fig: 2. Connection of devices to I2C bus in SM or FM

Pull-up resistor equation is given by:

R_p (max) = t_r /0.8473* C_b

C_b – bus capacitance,

t_r – rise time, max. rise time for 1 MHz, 400 kHz and 100 kHz I2C are 1us, 300 ns,120 ns respectively.

III. I2C SPECIFICATIONS

a) START (S) and STOP (P):

These are used to initiate and stop transactions on the I2C-bus. START is generated by pulling the SDA line low while SCL is high and a STOP is generated by pulling the SDA line high while SCL is high.

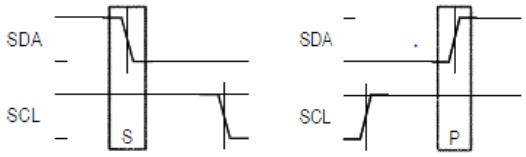


Fig: 3. Start and Stop condition

b) Data validity:

When SCL is HIGH data must be stable and can change when SCL is LOW.

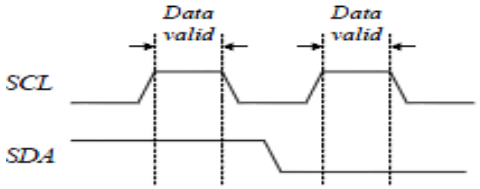


Fig: 4. Data validation timings

c) Acknowledgements:

It is given by the receiver by pulling down (LOW) the SDA line after transmitter releases the bus. If the SDA remains HIGH during this clock pulse, it is known as Not Acknowledge (NACK) signal. After NACK, the master can either generate a repeated START for a new transfer or a STOP to abort the data transfer [9].

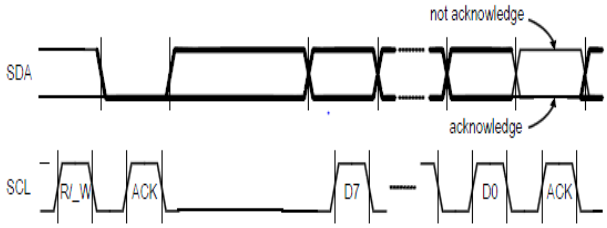
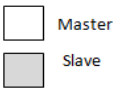


Fig :5. Acknowledge and not acknowledge signals in I2C

d) Write/Read:

The communication frame for write and read is shown in Fig: 6. The control bit (R/W) send along with the address bit decides the direction of transfer.



Transmit (0= write)

START	Slave Address [7]	R/W= 0	ACK	DATA [8]	ACK	DATA [8]	ACK	STOP

Receive (1= read)

START	Slave Address [7]	R/W= 1	ACK	DATA [8]	ACK	DATA [8]	ACK	STOP

Fig: 6. I2C communication frame for write and read operation

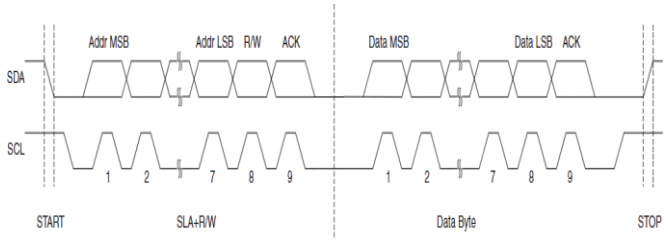


Fig: 7. Timing count of different bit transmission in I2C

IV. OPERATION

i. Master initiates the data transfer by sending START signal. It is as an attention signal to all the slave devices on the bus.
ii. Start is followed by the 7-bit ADDRESS of the slave it wants to communicate with.
iii. ADDRESS is followed by a R/W bit which acts as a control bit to decide the direction of data transfer.
iv. Master releases the bus and wait for the acknowledge (ACK) bit from the slave.
v. The desired slave pulls the SDA line LOW to send ACK signal.
vi. Depending on R/W bit value either master or slave sends the 8-bit data and acts as the transmitter.
vii. The other device acts as a receiver and sends ACK after receiving 8-bit data by pulling SDA LOW.
viii. After (N)ACK the master ends the transmission by sending STOP signal.
ix. Any master can now initiate a new data transfer by taking the control of bus.

A. Methods for efficient working of I2C

The various methods can be used in the designing of I2C to ensure that there is no bus conflict arising and thus, no data loss. These are discussed below.

a) Synchronization:

It is done to synchronize the clocks of more than one device. All the masters connected to the bus generate their own clock on SCL. The synchronization is done using wired-AND of I2C interfaces. When SCL goes LOW, it is hold low till the longest LOW period [1] of any device and the devices with less LOW period than this will go into HIGH wait state.

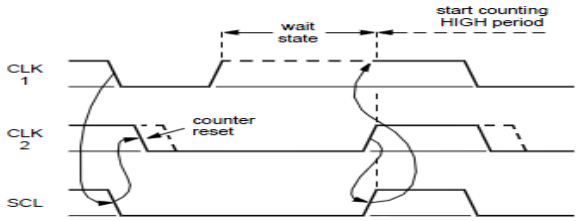


Fig: 8. Clock synchronization with multiple masters

b) Arbitration:

It includes master only and is done on SDA line while SCL is HIGH [6].

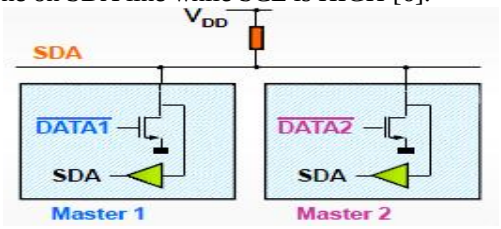


Fig: 9. Arbitration mechanism

When two master devices initiates a transfer at exactly same time, the one who sends "1" loses the control if other sends a "0". This is because it took the line to be used by other active master, it stops the transfer and waits for the STOP signal. The information on I2C bus is determined by winning master, there is no loss of bits in this process.

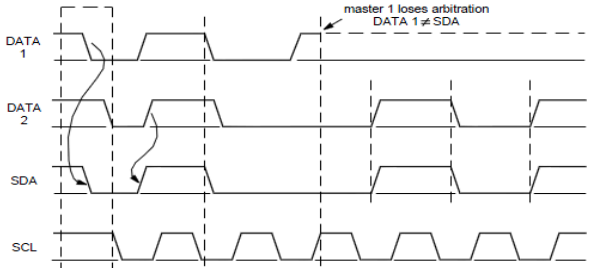


Fig: 10. Arbitration procedure in two masters

c) Clock stretching:

This mechanism gives an advantage in communication with the slow peripheral devices [3]. If the speed of the slave device is less than the clock generated by the master, it can hold the SCL line low to indicate that it is not ready yet. Then the master will wait till the clock is released by the slave before the next transfer.

d) High Voltage protection:

The series resistors (R_s) of few hundred Ω as shown in fig: 11 can provide protection from high-voltage spikes on two lines. The additional resistance should be included in bus capacitance (C_b) and parallel resistance (R_p) calculations.

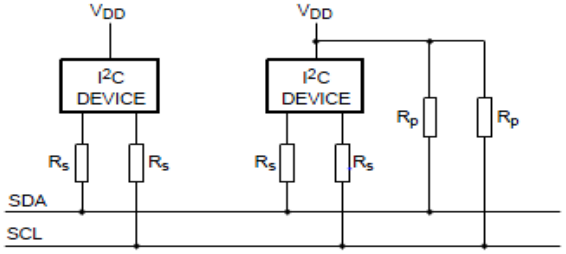


Fig: 11. High voltage protection using series resistance

V. I2C BUS ARCHITECTURE

The block diagrammatic representation of I2C is shown in Fig: 12. The Finite State Machine model acts as the main controller for the synchronized operation of the I2C. It is the sequential design that keeps the track of states and the inputs. The whole I2C design is made using verilog HDL in Modelsim.

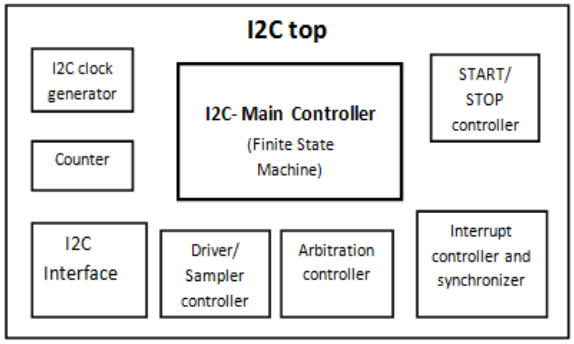


Fig: 12 Block diagram representation of I2C controller

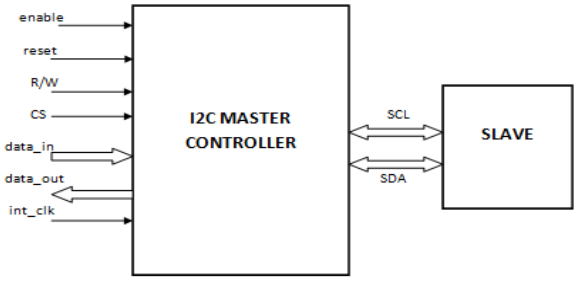


Fig: 13. RTL view of I2C bus Architecture

VI. RESULT

A. Simulation results:

The simulation is carried out in Modelsim Altera Starter 6.4a. The read and write operation is shown below

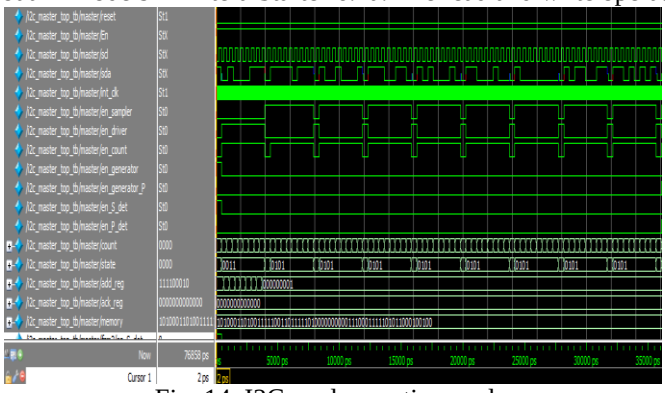


Fig: 14. I2C read operation cycle

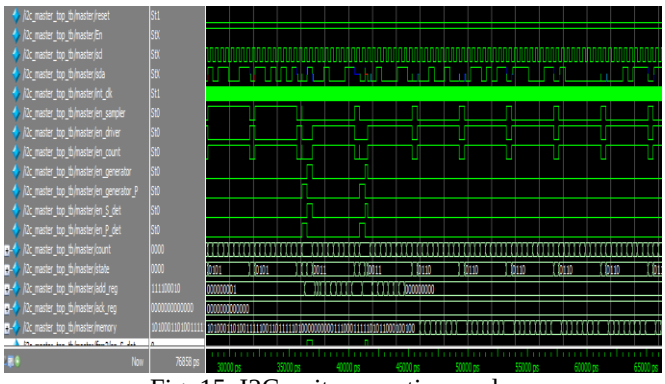


Fig: 15. I2C write operation cycle

B. Synthesis results:

The synthesis of I2C is carried out in XILINX XST 14.1 using Spartan3E-XC3S100E.The result of device utilization summary is shown below

```
Device utilization summary:
---------------------------

Selected Device : 3s100ecp132-4

 Number of Slices:                       83  out of     960     8%
 Number of Slice Flip Flops:             81  out of    1920     4%
 Number of 4 input LUTs:                155  out of    1920     8%
    Number used as logic:               151
    Number used as Shift registers:       4
 Number of IOs:                           5
 Number of bonded IOBs:                   4  out of      83     4%
    IOB Flip Flops:                       1
 Number of GCLKs:                         2  out of      24     8%
```

The technology tree schematic viewed from Xilinx is shown in Fig: 16.

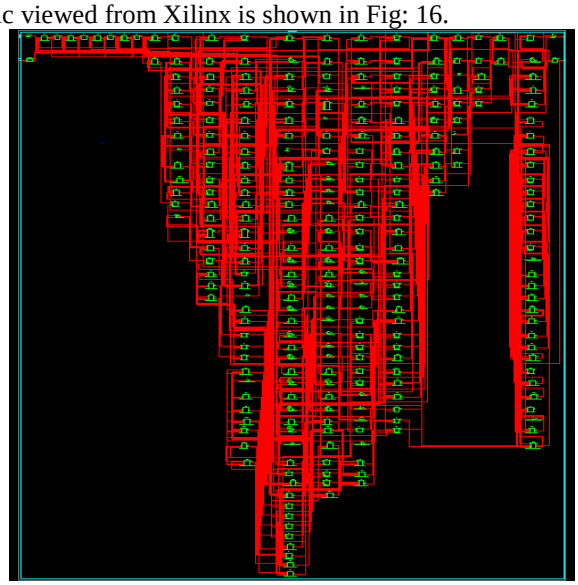


Fig: 16. Technology tree schematic of I2C bus Controller

VII. CONCLUSION

The I2C bus controller designed using verilog and simulated in ModelSim provides the efficient data transfer and synchronization by using acknowledgement, arbitration and clock stretching. Any low speed peripheral devices can be interfaced using I2C bus protocol as a master. The bus capacitance limits the number of devices attached to the bus. It supports various fabrication techniques which makes it possible for the designer to design without being much concerned of the specific technology. The synthesis tool itself optimizes the design as per requirement for the new technique. The design is meeting the timing constraints with the minimum utilization of the resources.

REFERENCES

[1] Philips Semiconductor, "I2C-bus specification and user manual", version 2.1 January 2000.

[2] J.M. Irazabel & S. Blozis, Philips Semiconductors, "I2CManual, Application Note, ref. AN10216-0" March 24, 2003.

[3] Richard Herveille, "I2C Master Core Specification" Open Cores, 2003

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

[5] F. Leens, "An Introduction to I2C and SPI Protocols", IEEE Instrumentation & Measurement Magazine, pp. 8-13, February 2009.

[6] NXP Semiconductors, I2C-bus Protocol and applications SASE, March 2010

[7] Cypress Semiconductor Corporation- I2C master/Multi-master/Slave 2.0, Document Number: 001-62887 Rev. A, Nov. 30, 2010.

[8] Bollam Eswari, N.Ponmagal, K.Preethi, S.G.Sreejeesh "Implementation of I2C Master Bus Controller on FPGA" International conference on Communication and Signal Processing, - IEEE- 978-1-4673-4866-9/13, April 3-5, 2013.

[9] UM 10204, I2C-bus-specification and user manual, Rev.6– 4 April, 2014.

[10] Tatiana Leal-del Río, Gustavo Juarez-Gracia, L. Noé Oliva-Moreno "Implementation of the communication protocols SPI and I2C using a FPGA by the HDL-Verilog language" Research in Computing Science 75 pp. 31–41; July, 2014.

AUTHORS AND AFFILIATIONS

1. **Deepa Kaith** is currently pursuing M.Tech. in Electronics and Communication Dept. ASET, Amity University Haryana. Her areas of interest are Digital System Design and verification.

2. **Professor Dr. JanakKumar B. Patel** is currently working as a Professor, Electronics & Communication Engineering Department, ASET, Amity University Haryana. He has done his Ph.D. from IIT Roorkee. He has 22 years of industrial and teaching experience in Engineering. College. His areas of research are image processing and VLSI Design

3. **Mr. Neeraj Gupta** is currently working as a Assistant Professor, Electronics & Communication Engineering Department, ASET, Amity University Haryana and currently pursuing Ph.D. from Amity University Haryana. He has 7 years of teaching experience in Engineering College. His areas of research are Device Modeling and VLSI Design.