Design of Multi – Master Bus Controller

S.Ahmadunnisa
UCEK, Kakinada

B.Leela Kumari
Asst. Professor
UCEK, Kakinada

K.Padma Raju
Professor
UCEK, Kakinada

***Abstract*—This paper presents design and implementation of Multi – Master Inter – Integrated Circuit (often written as or IIC) Bus Controller. The Multi – Master C Bus Interface is a circuit to perform serial communication based on data format transfer. The arbitration lost detect function makes multi master communication possible. The communication is done on four modes of data transfer depending on the application. Ihe module was designed in Verilog HDL. It is simulated and synthesized using Xilinx Design Suite 13.2**

***Index Terms*-C, Master, Serial data communication, Slave, Xilinx.**

I.INTRODUCTION

In serial data communication, there are many protocols like RS-232, RS-422, RS-485, SPI (Serial peripheral interface), Micro wire for interfacing high speed and low speed peripherals. These protocols requires more pin connection in IC(Integrated Circuit) for serial data communication to take place, as the physical size of IC have decreased over the years, we require less number of pin connections for serial data transfer to take place. USB/SPI/Microwire and mostly UARTS are all just point to point data transfer bus systems. These use techniques like multiplexing of the data paths and forwarding of messages to service multiple devices. To overcome this problem, the C protocol was introduced by Phillips. This protocol requires only two lines for communication with two or more chips and can control a network of device chips which has a two general purpose I/O pins whereas, other bus protocols require more pins and signals to connect devices.

Bollam Eswari et al[1].,discussed the implementation of C Protocol on FPGA, it is given that C Master Controller transmits data to and from slave. Any low speed peripherals can be interfaced by using C Master Controller.

J.J.Patel et al[2]., discussed the design and the implementation of C Bus Controller using Verilog, it gives an idea of C Bus Controller design which consists of start/stop control, Counter, Arbitration Unit, Microprocessor Interface, State machine, Interrupt Controller, Clock Generator and synchronizer.

P.K.Mehto et. Al[3] discussed about the design and the implementation for interfacing two integrated devices using C Bus. It gives an idea about C Master Controller pin level architecture.

In this paper, we are implementing Multi – Master C Bus Controller in various speed modes. Among all the speed modes throughput is found to be good in the High Speed Mode.

The synopsis of the paper is as follows: In section II, we discussed C Bus Specifications, in Section III, we had given proposed work which also includes various I2C Bus Characterstics and Bus Architectures, in Section IV, we di scussed about the designing of C Master Controller, in Section V, we posted the simulation and synthesis results, in Section VI, conclusions are drawn based on the results and future scope is discussed in brief.

II.C BUS SPECIFICATIONS

The C Controller Bus is a two-wire, bi-directional serial bus that provides a simple and it gives an efficient method of data transmission over a short distance between many devices. C provides good support for communication with various slow, on-board peripheral devices which are accessed intermittently, being extremely modest in its hardware resource needs. It has an advantage of being a simple, low-bandwidth, short distance protocol. C is easy to use to link multiple devices together since it has a built-in address. The two C signals are serial data (SDA) and serial Clock (SCL) as shown in Figure 1.The device which gives an initiation to a transaction on the C bus is termed as the master. The master normally controls the clock signal and the device being addressed by the master is called as a slave.

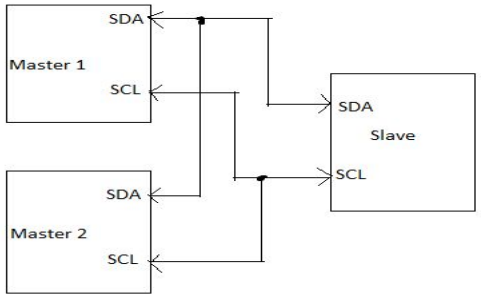

Figure 1: Multi – Master C devices

The C protocol supports multiple masters, though most of the system designs includes only one master. There may be one or more slaves on the bus. Both masters and slaves can

receive data bytes and also can transmit data bytes. There are four modes of data transfer: Standard Mode, Fast Mode, Fast Mode Plus and High Speed Mode. Standard C devices operate up to 100Kbps, fast-mode devices operates up to 400Kbps, fast-mode plus devices operate upto 1Mbps and High Speed Mode operate upto 4Mbps. Most of the C devices available today support 400 Kbps operation. Higher speed of operation allows C to keep up with the rising demand for bandwidth in multimedia and also in several other applications.

III. Proposed Work

A. Data Transfer

The SDA and SCL lines are two bi-directional lines. These are connected to a positive voltage supply through a pull up resistor. The bus is free when both of these lines are 'high'. The data on the SDA line is valid only when the SCL line is 'high'. Change of the data is allowed when SCL line becomes 'low'. During data transfer, the master generates the START and STOP conditions, both of which are unique conditions.

HIGH to LOW transition on the SDA line when SCL is HIGH is one such unique condition. This situation indicates a START condition. A LOW to HIGH transition on the SDA line when SCL is HIGH defines a STOP condition. START and STOP conditions are always generated by the Master. The bus is considered to be busy after the START condition. The bus is considered to be free again for a certain time after the STOP condition. This is shown in Figure 2.

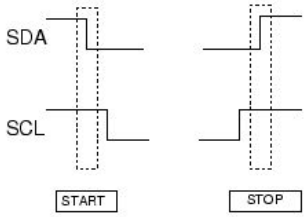

Figure 2: Start and Stop Conditions on C Bus

The data on the SDA line must be stable during the HIGH period of the clock. The change of data line is allowed only when the clock signal on the SCL line is LOW. It is as shown in Figure 3.

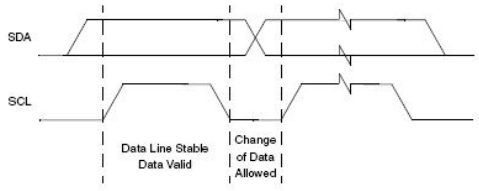

Figure 3: Data Validity

B. Byte Format

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is not restricted. Each byte has to be followed by an acknowledge bit. The byte-wise transfer is as shown in Figure 4.

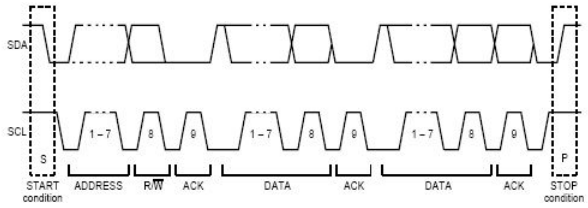

Figure 4: Byte-wise transfer

For every byte of transfer on the I²C bus, whether it is slave address or data always an MSB is sent first and LSB last. The byte format is as shown in Figure 5.

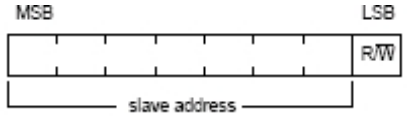

Figure 5: Byte Format

C. Acknowledge

The acknowledge-related clock pulse is generated by the master. The receiver must pull down the SDA line during the acknowledge clock pulse. It is so that it remains stable LOW during the HIGH period of this clock pulse. The acknowledge on I²C Bus is as shown in Figure 6.

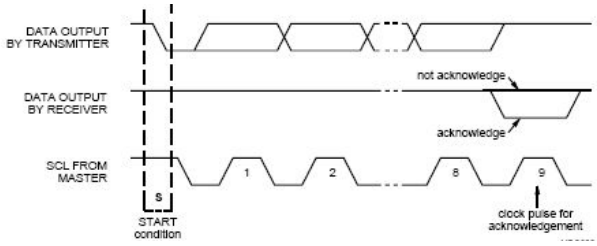

Figure 6: Acknowledge on C Bus

D. Frame Format

Data transfers follow the format in which, after a START condition a slave address is sent. This address is 7 bits long, it is followed by an eighth bit which is a data direction bit (R/W) which decides the direction of data transfer - a `zero' bit that indicates a transmission (TRANSMIT) and a `one' bit indicates a request for data (RECEIVE). A data transfer is always terminated by a STOP condition which is generated by the master. The frame format is as shown in Figure 7.

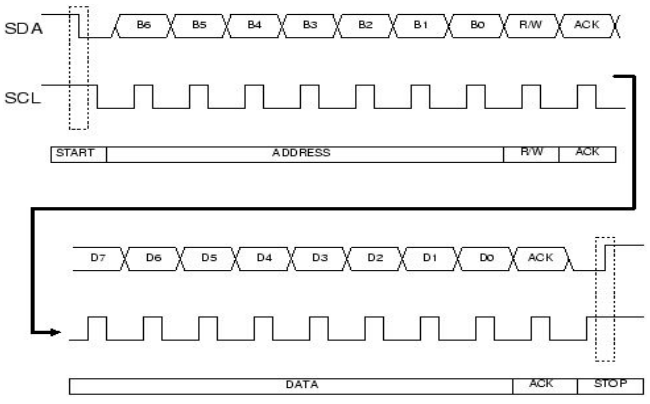

Figure 7: Frame format

E. Basic Architecture

The basic architecture of a bus controller is designed by the Microprocessor or Microcontroller interface and I²C master or slave interface to which the SDA and SCL lines are connected. The Bus Controller has communication with the microcontroller or microprocessor through the Address, Data and Control lines. The basic architecture is as shown in Figure 8 and the symbol of I²C Master Controller is as shown in Figure 9.

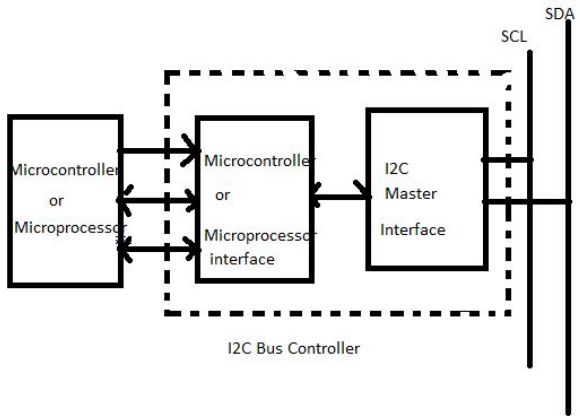

Figure 8: Basic Architecture of I2C Master Bus Controller

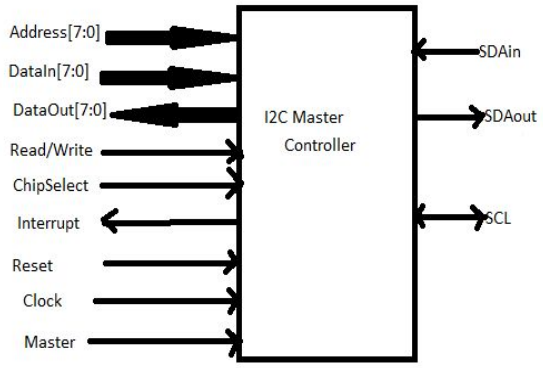

Figure: Symbol of I2C Master Controller

IV.DESIGNING I²C MASTER CONTROLLER

Designing the Master Controller is done by using a finite state machine (FSM). Implementation of finite state machine is done by writing Verilog code

Finite State machine

A finite state machine is a sequential circuit which uses a finite number of states and keeps track of its history of operations, and based on this history and its current inputs, it determines what to do next. A sequential circuit is a circuit where the outputs of this circuit are dependent on its history of operation and its current inputs.

V. RESULT ANALYSIS

I²C Bus Controller is designed in Verilog and the simulation results are obtained. The device utilization summary is as given below in Table 1.

Table 1: Device Utilization Summary

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	366	4656	7%
Number of Slice Flip Flops	550	9312	5%
Number of 4 input LUTs	670	9312	7%
Number of bonded IOBs	49	232	21%
Number of GCLKs	4	24	16%

The simulated result is as shown below in Figure 10 and RTL View is as shown in Figure 11

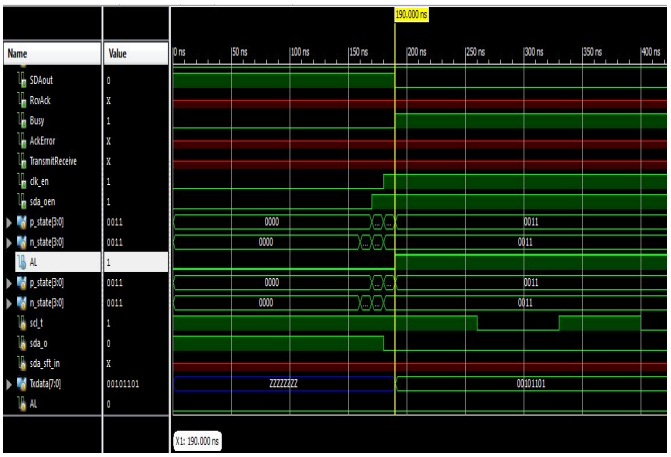

Figure 10: Simulation Waveform for Multi – Master I2C Bus Controller

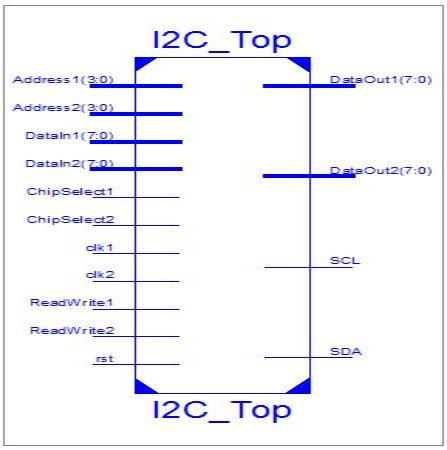

Figure 11: RTL View

The C Master Bus Controller is operated on four modes of data transmission. The throughput of various modes is as shown in Table 2

Table 2: Throughput in various speed modes

Standard Mode	77.92 μs
Fast Mode	20.48 μs
Fast Mode Plus	8.32 μs
High Speed Mode	2.24μs

VI. Conclusion and Future Scope

The I²C Master Controller has been implemented, simulated and synthesized for four modes of the operation. The designed controller is well suited for on-board applications. The controller can be used for embedded microprocessor boards, various low-power applications, communication systems, several cost - effective reliable automotive systems.

High Speed Mode devices remain fully downward compatible with the Fast or Standard- mode (F/S-mode) devices for bidirectional communication in a mixed-speed bus system. The throughput is also found to be best in the High Speed Mode when compared to all the modes of operation. Depending on the application, new devices may have a Fast or High speed mode I²C bus interface, although High speed mode devices are preferred as they can be designed in a greater number of applications.

The design of C Master Controller has good applications in the near future as the number of devices connected to a particular system are going to increase day by day. So there is always a need for a system which supports multiple protocols. In all these situations, C master controller acts as a great support and it will be a key in the future design to support multiple parallel functions.

VII. Acknowledgement

We express our profound sense of gratitude to our project coordinator Dr. K. Babulu, for his systematic guidance and valuable advices. We would like to express our sincere gratitude to Dr. A. M. Prasad, Professor, for providing us with all the necessary facilities for the work.

VIII. References

[1] Bollam Eswari, N.Ponmagal, K.Preethi, S. G. Sreejeesh, "Implementation of I2C Master Controller on FPGA", *International conference on Communication and Signal Processing, April 3-5, 2013,*pp.1113-1116

[2] J.J.Patel, Prof B. H. Soni, "Design and Implementation of I2C Bus Controller using Verilog", *Journal of Information, Knowledge and Research in Communication Engineering,* Vol. 2, pp. 520 – 522

[3] P.K.Mehto, P.Mishra and S.Lal, " Design and Implementation for interfacing two integrated device using I2C bus", *International Journal of Innovative Research in Computer and Communication Engineering,* Vol. 2, March 2013, pp. 3423-3426

[4] M. Alassir, J. Denoulet, O. Romain & P. Garda, " A SystemC AMS Model of an I2C Master Bus Controller", *International Conference,*2006, pp. 154 – 158

[5] J.K.Singh, M.Tiwari, V.Sharma, " Design and Implementation of I2C Master Controller on FPGA using VHDL", *International Journal of Engineering and Technology (IJET),*Vol. 4, 2012, pp. 162 -166

