International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 9 Sep 2021, pp: 1058-1060 www.ijaem.net ISSN: 2395-5252

FSM implementation of I2C protocol and its verification using verilog

Tapaswi SJ[1], Dr.Kiran V[2]

[1]Student, Department of ECE, RV College of Engineering, Bangalore, Karnataka
[2]Associate Professor, Department of ECE, RV College of Engineering, Bangalore, Karnataka

Submitted: 10-09-2021 Revised: 19-09-2021 Accepted: 23-09-2021

ABSTRACT: Present day's technology has reached a goal where an entire system can be implemented on a single chip which is nothing but called system on chip (SOC). It involves microcontrollers and various peripheral devices with each peripheral device having its own intellectual property (IP) named as IP cores. Serial Communication is established between these IP cores using various protocols like RS232, RS422 and UART etc. These protocols perform point to point communication which requires huge wiring connections, multiplexing of all the bus connections to deliver the information to the IP Cores. To overcome this I2C protocol is developed by Philips, which is a two line communication. Here only two pins i.e., SCL, SDA establish connection between various devices considering one as master and other as slave. These two pins communicate using particular commands like start, address, read/write, acknowledgement and stop commands. Both 7-bit and 10-bit addressing formats can be used, 10-bit addressing supports 1024 devices and 7-bit addressing supports 127 devices.

Keywords: I2C Communication, Verilog HDL, FSM Model, Timing Diagram, Synchronous communication.

I. INTRODUCTION

The inter integrated Circuit Bus is synchronous bi-directional half duplex (one directional communication at a given time) two wire serial interface bus. The concept of I2C bus was developed by 'Philips semiconductor' in early 1980's. The original intension of I2C was to provide an easy way to connection between a microprocessor/microcontroller and peripheral chips. The I2C bus comprise of two bus lines, namely Serial Clock – SCL and Serial Data – SDA. SCL line is responsible for generating synchronous clock pulses and SDA is responsible for transmitting the serial data across devices. I2C bus is shared bus system to which many number of I2C devices can be connected, Devices connected to the I2C bus can act as either 'Master' device or 'Slave' device. The 'Master' device is responsible for controlling the communication by initiating/terminating data transfer, sending data and generating synchronous clock pulses. 'Slave' devices wait for the commands from the master and respond upon receiving the commands. 'Master' and 'Slave' devices can act as either transmitter or receiver. Regardless whether a master is acting as transmitter or receiver, the synchronization clock is generated by the 'Master' device only [3].

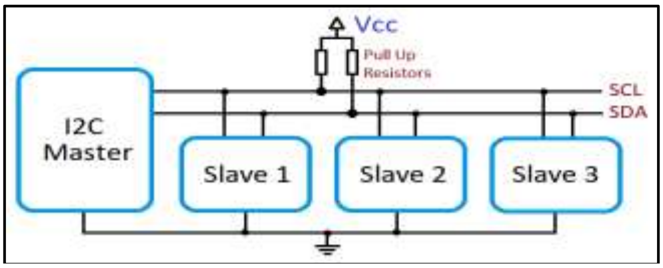

Figure 1:I2C Bus interfacing diagram

FEATURES

- Only two common bus lines (wires) are required to control any device/IC on the I2C network
- No need of prior agreement on data transfer rate like in UART communication. So the data transfer speed can be adjusted whenever required

- Simple mechanism for validation of data transferred
- Uses 7-bit addressing system to target a specific device/IC on the I2C bus
- I2C networks are easy to scale. New devices can simply be connected to the two common I2C bus line

II. LITERATURE SURVEY

I2C uses only two pins SCL, SDA to establish connection between various devices considering one as master and other as slave. These two pins communicate using particular commands like start, address, read/write, acknowledgement and stop commands. Both 7-bit and 10-bit addressing formats can be used. 10-bit addressing supports more addressing lines i.e., 1024 compared to 127 addressing lines in 7-bit mode [1].

The 'bus' wires are named SDA (Serial data) and SCL (Serial clock), these two bus wires have the same configuration. They are pulled-up to the logic 'high' level by resistors connected to a single positive supply, usually +3.3 V or +5 V but designers are now moving to +2.5 V and towards 1.8 V in the near future. All the connected devices have open-collector (open drain for CMOS - both terms mean only the lower transistor is included) driver stages that can transmit data by pulling the bus low, and high impedance sense amplifiers that monitor the bus voltage to receive data. Unless devices are communicating by turning on the lower transistor to pull the bus low, both bus lines remain 'high'. To initiate communication a chip pulls the SDA line low. It then has the responsibility to drive the SCL line with clock pulses, until it has finished, and is called the bus 'master' [2].

III. METHODOLOGY

The designed I2C protocol is based on FSM (Finite State Machine) model. The different states are explained as follows.

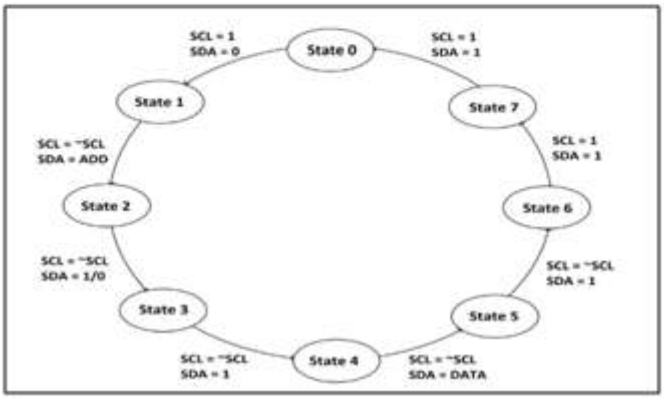

Figure 2:FSM model of I2C

- **State 0 - Idle:** No operation. Both SCL and SDA lines are pulled up to high level through pull up resistors as shown in Figure 1.
- **State 1 – Start:** The SDA line switches from a high voltage level to a low voltage level before the SCL line switches from high to low.
- **State 2 – Address:** Master will send a 7 bit unique address to the slave when the master wants to communicate with it.
- **State 3 – Read/Write bit:** Master will send a single bit specifying whether the master is sending data to the slave (low voltage level) or requesting data from it (high voltage level).
- **State 4 – Acknowledgment:** If an address frame was successfully received by the slave, an ACK bit is returned to the master from the slave.
- **State 5 – Data:** A 8 bit data will be sent to the slave from the master or 7 bit data will be received by the master from the slave depending upon read/write bit sent in state 3.
- **State 6 – Acknowledgment:** If a data frame was successfully received by the slave/master, an ACK bit is returned to the master/slave from the slave/master
- **State 7 – Stop:** The SDA line switches from a low voltage level to a high voltage level after the SCL line switches from low to high.

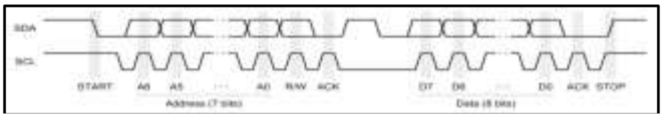

Figure 3:Timing diagram for Read/ Write Operations

IV. RESULT

In this work, I2C communication protocol is demonstrated using FSM modeling approach and the functionality of the designed FSM is verified using Verilog HDL using the Xilinx ISE tool. The timing wave formsof the simulation is shown in figure 4.

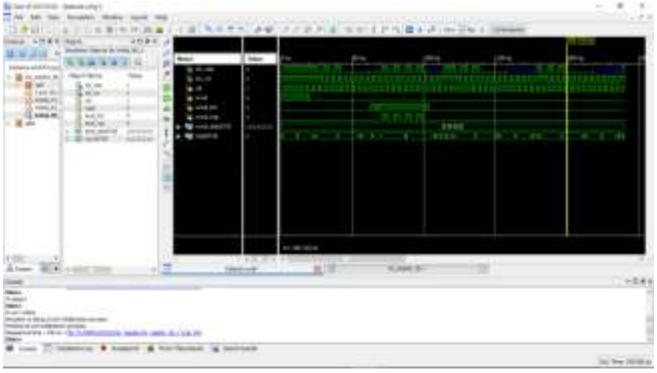

Figure 4:timing waveforms of simulation

V. SUMMARY

I2c is one of the most widely used on chip communication protocols. This paper aims to provide the understanding of this protocol and the basic hardware implementation of the protocol using Verilog HDL.

REFERENCE

[1]. Lakshmi ManasaKappaganthu, Durga Prakash M, “I2C Protocol and its Clock Stretching Verification using System Verilog and UVM”. International Conference on Inventive Communication and Computational Technologies (ICICCT 2017)

[2]. "I2C Licensing Information". nxp.com. Archived from the original on 2017-01-10. Retrieved 2018-04-29.

[3]. Imran Ali, Sung Hun Cho, Dong Gyu Kim, Muhammad Riaz Ur Rehman and Kang-Yoon Lee, “A Design of Ultra Low Power I2C Synchronous Slave Controller with Interface Voltage Level Independency in 180 nm CMOS Technology” 978-1-5386-2285-8/17/$31.00 ©2017 IEEE

[4]. Zheng-wei HU, “I2C Protocol Design for Reusability,” Third International Symposium on Information Processing, North China Electric Power University, Baoding, China.

[5]. J. W. Bruce, “Personal Digital Assistant (PDA) Based I2C Bus Analysis,” IEEE Transactions on Consumer Electronics, Vol. 49, No. 4, Nov. 2003.

[6]. Peter Corcoran, “Two Wires and 30 Years: A Tribute and Introductory Tutorial to the I2C Two-Wire Bus” U.S. Patent 4 689 750, Aug. 25, 1987