International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 462 - 469

Design and Implementation of I2C Master Controller for Serial Communication using VHDL

Sujata. S. Chiwande[1], Riya Wankhede[2], Diksha Dhawale[3], Aishwarya Chayande[4], Bharati Pahune[5]

[1]Asst. Prof. in ET dept. Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India,
[2]B.E in ET dept., Yeshwantrao Chavan College of Engineering, India,
[3]B.E in ET dept., Yeshwantrao Chavan College of Engineering, India
[4]B.E in ET dept., Yeshwantrao Chavan College of Engineering, India,
[5]B.E in ET dept., Yeshwantrao Chavan College of Engineering, India,
sujata_chiwande28@yahoo.co.in
wankhede.riya98@gmail.com
chandeaishwarya@gmail.com
dikshadhawale428@gmail.com
bharatipahune11@gmail.com

Abstract

The paper focuses on the design and implementation of I2C that is inter integrated circuit master controller for single master buses. The program is written in VHDL (Very high speed hardware descriptive language) for the use in FPGA and it is simulated by using the Quartus II software. The I2C consists of simple two wire bus interface which includes a bidirectional serial data line and serial clock line that will read and write to the given user logic. The method is simple and efficient for the serial communication between the devices without any data loss. To analyze the results and the operation of I2C master controller the RTC maxim DS1307 (real time clock) as a slave device is used which is connected to the I2C bus. Using I2C bus it gets very simple to interface as many as devices to a system and to communicate to send and receive the messages or data between the devices.

Keywords**:** *VHDL, I2C, SDA, SCL, FPGA, Master, Slave, HDL.*

1. Introduction

Inter-Integrated Circuit (I2C) bus protocol is a serial communication protocol designed by Philip semiconductors. This paradigm has been proliferating its use in serial communication. There are a number of standard protocols that are developed for communicating with different devices successfully such as UART, USART, SPI, I2C, CAN, etc with its advantages and disadvantages. Among all these protocols the I2C bus protocol increases the efficiency of the hardware and also the simplicity to connect as various devices can simply communicate with only two signals (SDA and SCL) regardless of how many devices are on the bus. It allows multiple masters and multiple slaves communication for various fast and slow devices. The connection between the slave devices is identified through a particular 7- bit address for each deice. I2c provides extensible, flexible and low pin count serial communication with devices [1].

ISSN: 2233-7857 IJFGCN

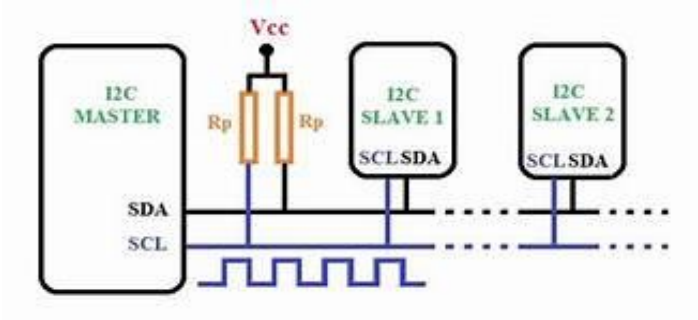

Figure 1. I2C bus configuration

1.1 Overview

As mentioned I2C bus protocol is known as Inter Integrated Circuit which is a bi-directional, two-wire synchronous protocol used as a serial communication standard for connecting various devices. It consists of SDA (serial data) which carries data for transmission and reception between devices connected on it and SCL (serial clock). These two lines are connected by pull-up resistors to the positive voltage supply. The pull-up resistors keep the SDA and SCL lines high. The master device generates the clock signals and also starts and stops the data transmission. Master device can also restart the process and it is called as repeated start. It is as same as the START condition. The slave device can be a transmitter and a receiver for reading and writing conditions and each device has their unique address. The data can be changed when the SCL is at low. The pull up resistor for SCL is need when number of master wanted to start data transmission at once called as arbitration. The I2C interface is much simpler and flexible as there are various peripheral devices are available.[2]

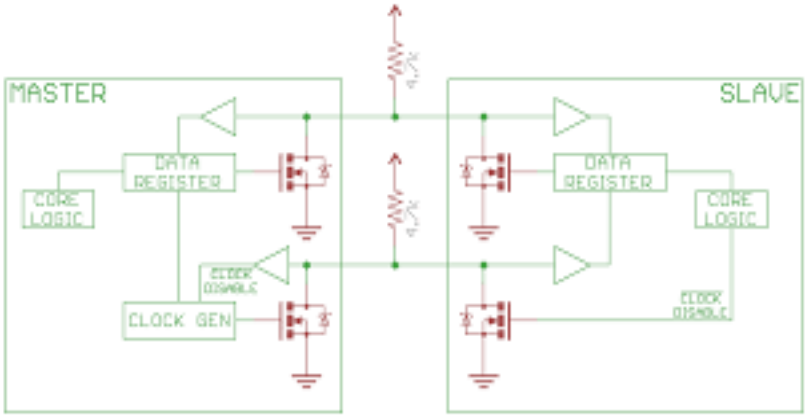

Figure 3. I2C Circuit Diagram with Pull up Resistors

I2C bus protocol

I2C bus protocol is two wired serial protocol, having bidirectional SDA (serial data line) and SCL (serial clock line) to transfer data between the devices which are attached to the bus. And each one of the slave device is identified by their addresses to connect with the master. The slave device can be any microcontroller, sensor, memory or I/O interfaces. I2C bus protocol is simple to use and allows more than one master devices with the two wires with pull up resistors to connect multiple devices with each other. At first I2C features included 100 kHz clock frequency and later it was improved with 400 kHz as fast mode, 3.4 MHz as high speed mode and 5 MHz as ultra-high speed mode. In I2C the master device initiates the data transfer with the START and STOPS bits and also generates clock signals for the slave device.

SDA and SCL signal

SDA (serial data) and SCL (serial clock) both are bidirectional signals which allow the communication from a system with various devices. These both lines are connected with a voltage supply and pull-up resistors. The I2C bus protocol provides different speeds for transferring data with different devices, in which the standard speed is up to 100 kbit/sec, fast speed is 400kbit/sec, fast mode plus is 1m M bits/sec and high speed mode is up to 3.4Mbits/sec.

SDA and SCL logic levels

With a wide variety of devices that can be connected to I2c bus protocol with different technologies, the logic levels i.e. HIGH and LOW are not fixed at all and also it is depended on the VDD level. So generally the VIL is up to 0.3vdd and the VIH is 0.7vdd. Some of the devices have the fixed logic levels for VIH and VIL that is 3V and 1.3V.

START and STOP conditions

The transmission between master and slave device initiates with the START bit and ends with the STOP bit. And both the operations are controlled by the master device.
For the START condition, the SDA line must be high to low and SCL as high while for STOP condition the SDA must be at low to high transition while SCL remain high. Always the 8-bit data is transferred on the I2C bus followed with the acknowledgement bit which indicated when the slave device is ready to receive and transmit the data. There is no limitation to send the data bytes to the device.

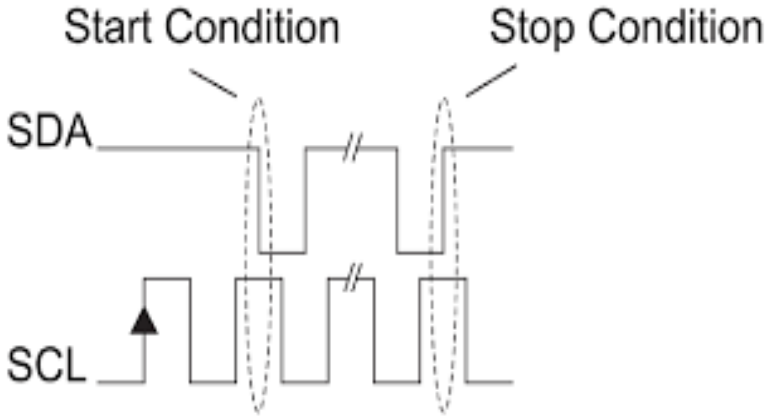

Figure 4. START and STOP condition

1.2 Byte Format

Every byte (8-bit) that is sent on the I2C bus is followed by an acknowledgement bit that indicates:

1. The address of the slave device is identified and the transmission can start.
2. The data that is sent is successfully received by the slave device and the next byte can be accepted.
3. The slave has received the data and the process can terminate.
4. After the 8-bit transmission, the transmitter pulls the SDA low so that the slave device can acknowledge or not acknowledge the data that is sent. If the SDA line is high, then the data is not acknowledged and in that case the transmitter can stop the process or else it can be repeated with a START condition. And if the SDA line is transitioned to low, it indicates the data has been acknowledged.

ISSN: 2233-7857 IJFGCN

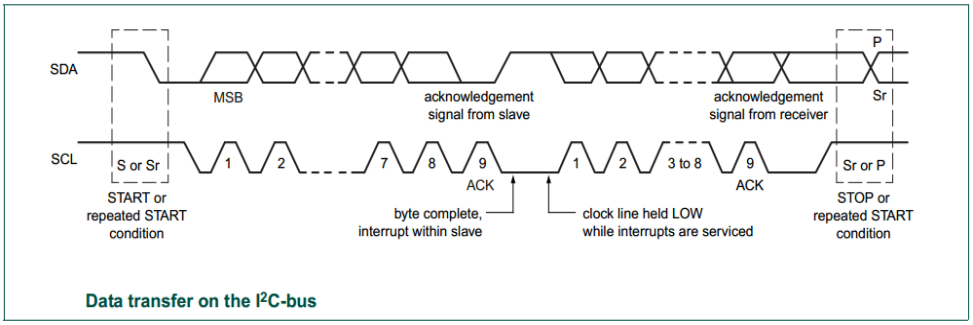

Figure 5. Data Format

2. Serial Communication

In I2C serial communication the data is transferred bit wise between master and slave devices through a single line i.e. serial data line with clock signals. The master initiates the connection with the slave device with their specific 7-bit address with high to low voltage level followed by the read or write bit. After receiving the address byte slave sends an acknowledgement bit by giving SDA low for a bit and the master will transmit a slave register address pointed by the register pointer. According to the read/write bit the master conveys or fetches the data frame. After transferring the data frame, the device which receives the data will return another acknowledgement bit indicating successful reception. After transmission is completed we have to stop the communication, for this the master gives stop condition to slave. By giving SCL to high before converting SDA high.[3]

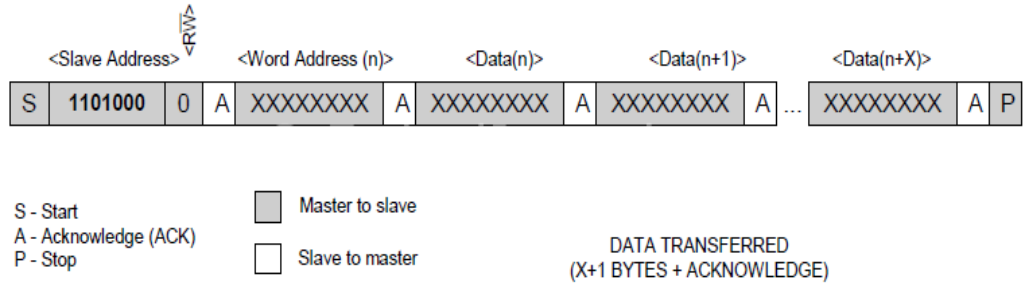

Figure 6. Write Process

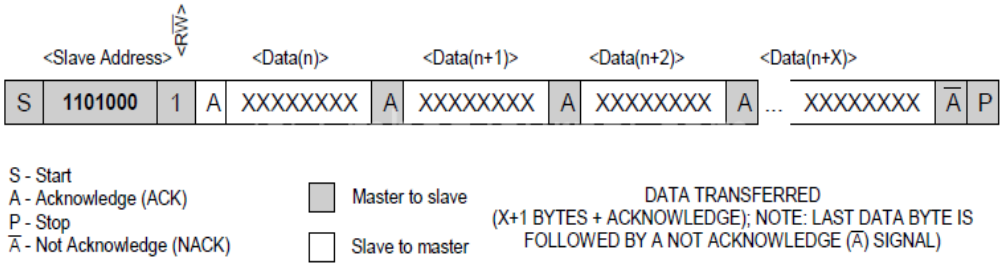

Figure 7. Read Process

2.1 Clock synchronization

Since I2C bus protocol allows multiple master controls, the master devices can initiate the data sending at the same instant. So for that, decide whether which master device can take control and transmit data there are two methods named as clock synchronization or clock stretching and arbitration. When a single master is used in a system there is no need of clock synchronization and arbitration. In situations where the slave device is unable to

operate with the clock speed of the master device and needs to slow down or hold the process for a while, then the clock stretching method is used. The master device always controls the clock speed. This is done by stretching the serial clock line, as the slave device can hold process when SCL is in low transition to slow down the clock speed of the serial clock line.

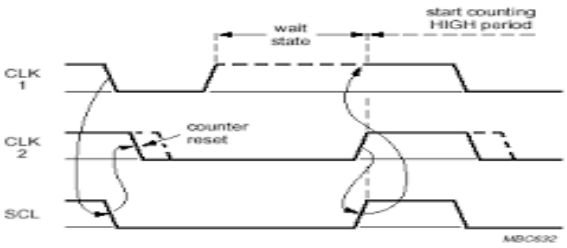

Figure 8. Clock Synchronization

2.2 Arbitration

Several I2C multi-masters can attached to same I2C bus and operate concurrently. By constantly monitoring SDA and SCL for START and STOP conditions, they can determine whether the bus is currently idle or not. If the bus is busy, masters delay pending I2C transfers until a stop condition indicates that the bus is free.

However, it may happen that two master's start a transfer at the same time. During the transfer, the masters constantly monitor SDA and SCL. If one of them detects that SDA is low when it should actually be high, it assumes that another master is active and immediately stops its transfer. This process is called *arbitration*.

MAXIM DS1307

RTC DS1307 is a module which is used to show the accurate time and date and it consists of a battery setup in the module to keep running the module in absence of the external power supply. The module maintains the information for seconds, minutes, hours, date, day and month. It is a low cost and quite accurate chip that has simple two wire I2C interface feature for connections to any microcontroller.[4]

1. Maximum voltage at SDA, SCL: VCC + 0.3V
2. Operating temperature: -45ºC to +80ºC
3. 400Khz I2C interface
4. Low power consumption
5. Portable size

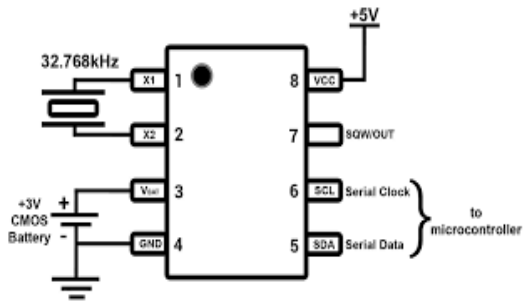

Figure 9. Pin Diagram of DS1307

3. Work Done

Algorithm

Step 1: The first step for idle condition is that SDA and SCL becomes high so that I2C doesn't perform any operation.

Step 2: When we start data transmission the master gives START so that SCL is at H (high) and SDA becomes H to L (Low), this is start condition.

Step 3: After start condition the master convey the address WR (write) to that of slave (SL).

Step 4: If SL address send by the master is parallel to the slave then it gives response as a acknowledgement bit.

Step 5: We can transmit 8-bit Address to the SL by master then again the acknowledgement is given by the SL to the master.

Step 6: The overall operation performed by them is that the data transmitted by the master to slave and then slave gives acknowledgement bit in a response.

Step 7: After data transmission, now we have to stop the transmission and for that slave convey a stop bit to the master to discontinue the processing between master and slave. By this the SCL is H and SDA becomes L to H, this is the STOP condition. The read operation is performed by first writ operation then read process is accomplished. For read operation the STOP condition will not be execute.

Step 8: We know that master send slave address to the slave and this task is performed for the read operation.

Step 9: As I2C is a bidirectional bus so that master can receive the data from slave and master also gives acknowledge to the slave.

Step 10: After data transmission from slave to master, the master redirect STOP bit to discontinue the communication so that SCL is H and SDA becomes L to H.

RTL View

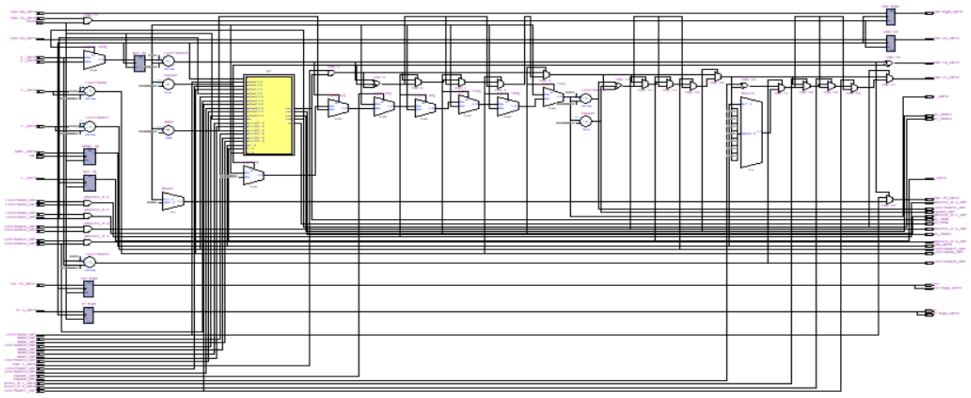

Figure 10. RTL View

4. Result and Conclusion

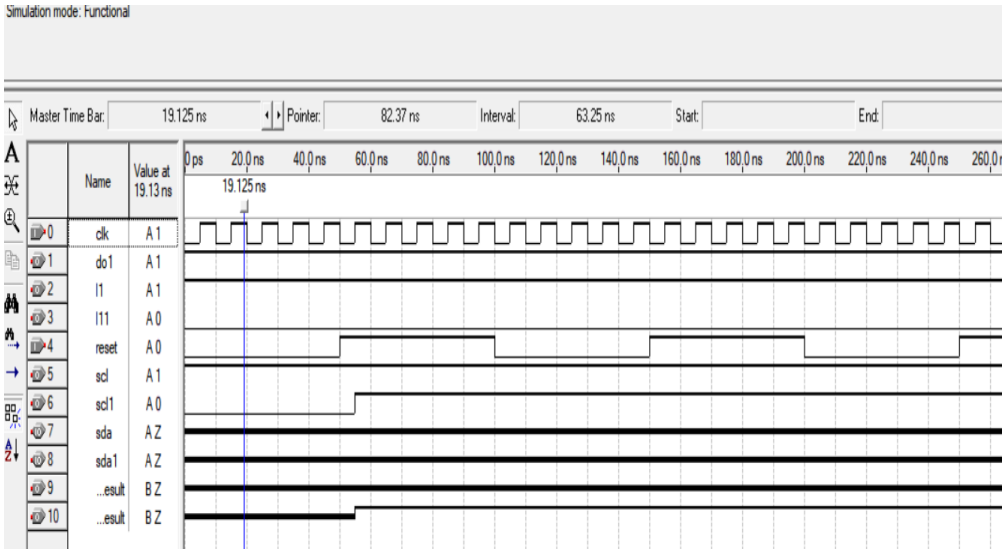

Figure 11. Output Waveform

The above figure shows the waveform presenting the output of the read and write process from master controller with RTC.

As the I2C bus has many advantages such as high performance, flexibility and low cost. The method is programmed and designed in VHDL and is simulated by using Quartus II software, so it will allow rapid prototyping of the interface model for large scale communication. The I2C bus has so many applications that can be used for various purposes in future as it can simply connect as many as slave devices to a system.

References

[1] Prof. Jai Karan Singh, Prof. Mukesh Tiwari, Vishal Sharma "Design and Implementation of I2c master controller on FPGA using VHDL" International Journal of Engineering and Technology (IJET)Vol 4 No 4 Aug-Sep 2012 .

[2] Madhuri Hanumanta Daware, Prof. A. S. Patil "IMPLEMENTATION OF I2C BUS PROTOCOL ON FPGA " ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-2, ISSUE-8, 2015.

[3] Pavan Kumar K, Daniel Raju Paga, V. Bagyaveereswaran "I2C Master Protocol Implementation in VHDL for Data Acquisition From CDC" (IJSCE) ISSN: 2231-2307, Volume-6 Issue-1, March 2016

[4] M. Mohankumar, N. Birundha and T. Suganya "Interfacing of I2C Master Bus Controller with FPGA" AJAST, Volume 2, Issue 2, Pages 172-180, April-June 2018 .

[5] Bhushan A. Sawalkar1 "A Review on Analysis and Reusability of FPGA Implementation for I2C Protocol", International Journal of Scientific & Engineering Research, Volume 7, Issue 2, February-2016 ISSN 2229-5518.

[6] Shoaib Shah, Sobhan, Sudipta Das and Iqbalur Rahman, "Implementation of I2C using System Verilog and FPGA", ICAEPE, 2011

[7] Spandana Sunku, Lavanyalatha Mutyala "Design of I2C BUS Controller using VHDL" International Journal of Emerging Engineering Research and Technology Volume 3, Issue 8, August 2015, PP 125-129 ISSN 2349-4395 (Print) & ISSN 2349-4409.

[8] M .Mohankumar, N. Birundha and T. Suganya "Interfacing of I2C Master Bus Controller with FPGA" Asian Journal of Applied Science and Technology (AJAST) Volume 2, Issue 2, Pages 172-180, April-June 2018.

Author's Profile

Sujata S. Chiwande completed M.Tech in Electronics Engg. from Y.C.C.E, Nagpur in year 2010. She is Currently working as Asst. Prof. in Electronics & Telecommunication department, in Y.C.C.E, Nagpur. She has 15 years of teaching experience. Her research of Interest includes Low Power VLSI Design. She has published number of papers in various reputed international Journals & conferences.

Riya Wankhede pursuing B.E. in Electronics and telecommunication Engg. from Y.C.C.E, Nagpur.

Aishwarya Chayande pursuing B.E. in Electronics and telecommunication Engg. from Y.C.C.E, Nagpur.

Bharati Pahune pursuing B.E. in Electronics and telecommunication Engg. from Y.C.C.E, Nagpur.

Diksha Dhawale pursuing B.E. in Electronics and telecommunication Engg. from Y.C.C.E, Nagpur.

ISSN: 2233-7857 IJFGCN