International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017), e-ISSN: 2455-2585
Volume 5, Issue 03, March-2019

Design of I2C Protocol

Dr. Sagar Patel[1], Prachi Talati[2], Saniya Gandhi[3]

[1]*Chandubhai S. Patel Institute of Technology, CHARUSAT University,Changa,Gujarat,India,*
[2]*Chandubhai S. Patel Institute of Technology, CHARUSAT University,Changa,Gujarat,India,*
[3]*Chandubhai S. Patel Institute of Technology, CHARUSAT University,Changa,Gujarat,India,*

Abstract— I2C (Inter IC) protocol is a simple two wire line protocol which is used to transfer data from one device to another device. I2C is simple, bidirectional, half Duplex protocol. This Serial Communication Protocol is developed by Philips Semiconductor (NXP Semiconductors) and now it is used by almost all major IC manufacturers.

Keywords— Master, Slave, SDA, SCL, Verilog HDL

I. INTRODUCTION

There are many protocols for transmission and reception of data from one device to another device, but I2c is a simplest protocol because it has only two wire lines SDA (Serial Data Line) and SCL (Serial Clock Line) and it has multi-master capability unlike SPI (Serial Peripheral Interface) protocol and also addressing of I2C is simple because it does not require any CS lines used in SPI and it is easy to add extra devices on the bus [1]-[6]. I2C is a Synchronous Protocol unlike UART (Universal Asynchronous Receiver/Transmitter).There is so many application of I2C protocol in real time such as micro controller, wireless communication and many more [3] [5] [6].

Both SDA (Serial Data Line) and SCL (Serial Clock Line) are bidirectional that provides a simple and efficient communication between devices. In I2C we can connect multiple masters and multiple slaves but single master and multiple slaves combination mostly used. Master is the device which generates clock signal, which initiates a transfer and which terminates the transfer. Slave is the device which addressed by a Master[1][2][4].

Here are some features of I2C Bus:

- Only two wire line : SDA and SCL
- I2C can be multi-master and multi-slave
- Serial, 8-bit oriented, bidirectional data transfers can be made at up to 100 Kbit/s in the Standard-mode
- Up to 400 Kbit/s in the Fast-mode
- Up to 1 Mbit/s in Fast-mode Plus,
- Up to 3.4 Mbit/s in the High-speed mode.
- Unidirectional data transfers up to 5 Mbit/s in Ultra Fast-mode
- 7 bit,10 bit and free data Addressing Modes

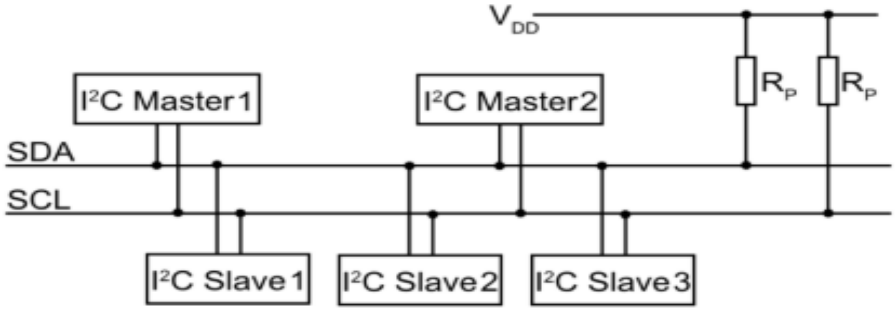


Fig.1. I2C Multimaster Multislave Configuration with Pull up Resistors [1]

Transmission or Reception is divided into following states:

- IDEAL: In Ideal Condition value of both lines SDA and SCL is 1.
- START: Transition of SDA from high to low while SCL is high defines START condition.
- Address and Read/Write: After START condition, Master sends address of slave to which it wants to write or from which it wants to read. In this there are many addressing modes like 7 bit, 10 bit, free data addressing mode but in this project we use 7 bit addressing mode.7 bit slave address transmitted bit by bit over SDA line and after this an additional Read/Write bit is transmitted which informs that if master wants to read from slave or write to slave.

If Read/Write bit is zero then it indicates that master wants to write to the slave device.
If Read/Write bit is one then it indicates that master wants to read from the slave device.
All transition occurs on SDA while SCL is low. Data must be stable while SCL is high.

- Data Transfer: If address of the slave matches with the address transmitted by the master then slave sends ACK to the master. After receiving an acknowledgement from the slave, data is transmitted bit by bit over SDA from master to slave(if Read/Write bit is zero) or slave to master(if Read/Write bit is one) depending upon Read/Write bit.ACK or NACK should be sent by slave(if write operation) or master(if read operation)after every 8 bits.
- STOP: After receiving ACK or NACK from slave (During writing operation), after last byte master terminates the transfer by transition of SDA from low to high while SCL is high. If master wants to address same slave again then instead of STOP condition, master can send REPEATED START condition. During reading operation, master sends NACK after last byte of data and then it terminates transfer.

II. LITERATURE REVIEW

In this project we design software implementation of I2C protocol using Verilog HDL language and simulation done in VCS.

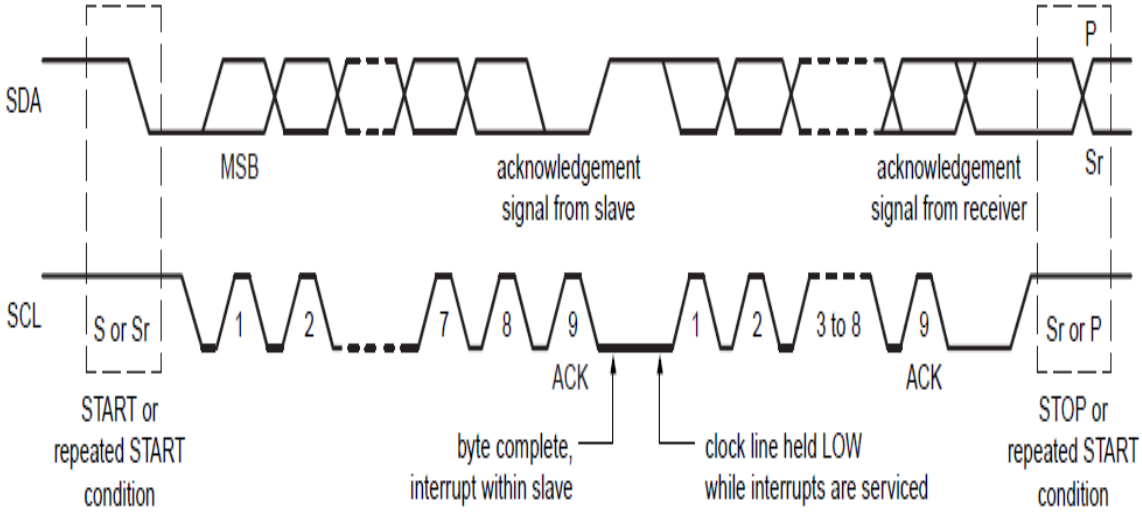
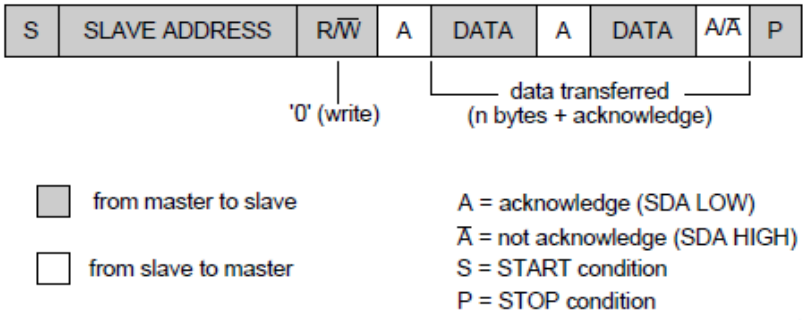
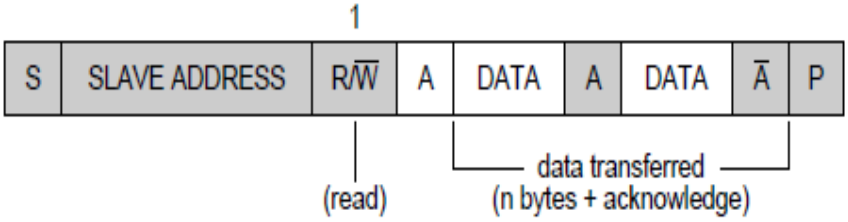


Fig. 2 Byte format of I2C Protocol [2]



A master-transmitter addressing a slave receiver with a 7-bit address (the transfer direction is not changed)

Fig. 3 writing data from master to slave [2]



A master reads a slave immediately after the first byte

Fig. 4 Reading data from slave to master [2]

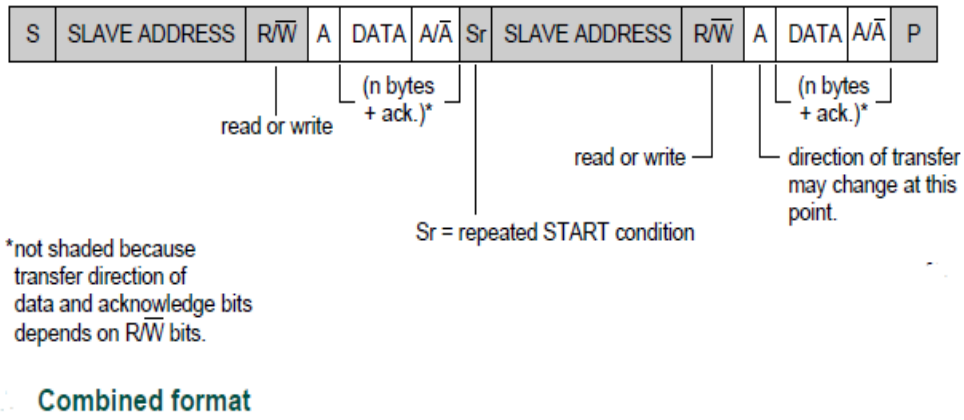


Fig. 5 Writing and Reading data [2]

In multi-master system, when two masters begin transmitting on a free bus at the same time then to prevent collision there must be a method for deciding which takes control of the bus and complete its transmission. This is done by Clock Synchronization and Arbitration method.

A. Clock Synchronization

Clock Synchronization is performed by using wired AND connection of I2C interfaces to the SCL line. This means a HIGH to LOW transition on the SCL line causes the masters concerned to start counting off their LOW period, once a master clock is LOW, it holds the SCL line in that state until the clock HIGH state is reached. If another clock is still within its LOW period, the LOW to HIGH transition of this clock may not change the state of the SCL line. The SCL line is held LOW by the master with the longest LOW period. During this time, master with a shorter LOW period enters in a HIGH wait state. A synchronized SCL clock is generated with its low period determined by a master which has longest clock low period, and its high period is determined by a master which has shortest clock high period.

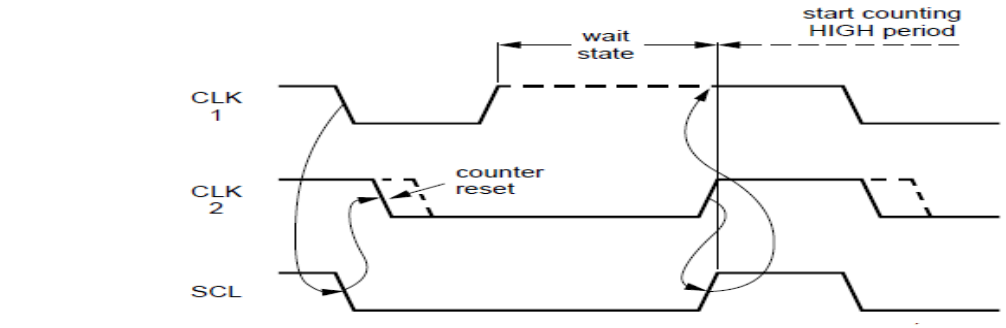


Fig. 6 Clock Synchronization between two masters [2]

B. Arbitration

Like Clock Synchronization, Arbitration is used when more than one master start data transfer at the same time. Arbitration is required to determine which master takes control over SDA line and completes its transfer. Arbitration performed bit by bit. At every bit, while SCL is high, each master checks if SDA level matches with what it has sent. The first time master sends high but it detects low level of SDA then that master looses arbitration and it goes into wait state.

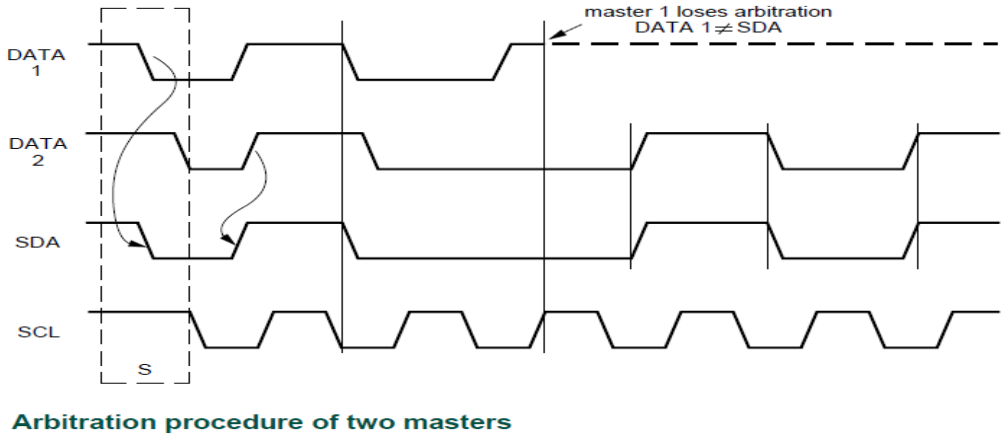


Fig. 7 Arbitration procedure of two masters [2]

C. Clock Stretching

During Communication, on a byte level, device may be able to receive data at fast rate but it needs more time to store a received byte or to prepare a next byte to be transmitted. Slaves can then hold the SCL line low which is known as Clock Stretching. During this time, master goes into a wait state.

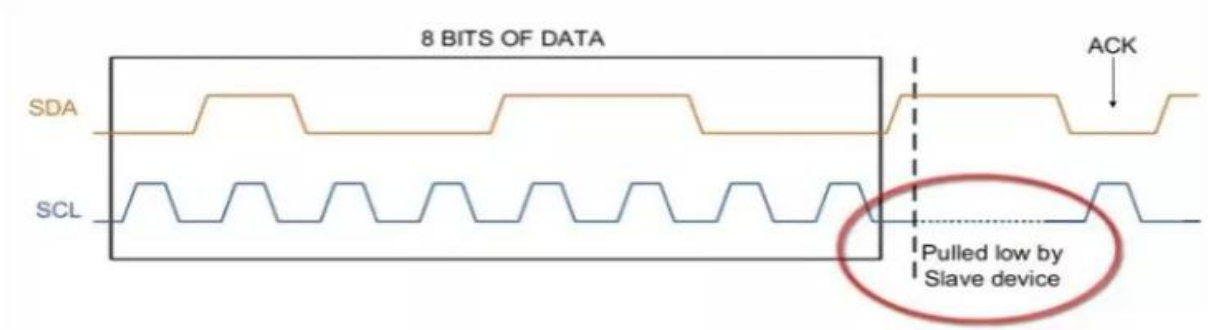


Fig. 8 Clock Stretching [4]

III. CONCLUSION

An I2C is the easy and cheap communication protocol, it can be multi-master or multi-slave. In I2C we get the acknowledgment (ACK) and not acknowledgment (NACK) bits after the each transmitted byte. I2C is an industry standard protocol which is used in many applications such as accessing EEPROM or accessing Real Time Clock. Some disadvantage also attaches with I2C, it is a half-duplex communication and slow as compared to SPI (serial peripheral communication) and it can cover shorter distance.

REFERENCES

[1] http://www.handsonembedded.com/stm32f103-spl-tutorial-6/
[2] https://aticleworld.com/i2c-bus-protocol-and-interface/
[3] S Patel, VV Dwivedi, YP Kosta, "A Parametric Characterization and Comparative Study of Okumura and Hata Propagation-loss-prediction Models for Wireless Environment", International Journal of Electronic Engineering Research, Vol. 2, No. 4, 2010
[4] NXP Semiconductors,UM10204, *i2c bus specification and user manual*, Rev. 6- 4 April 2014
[5] Sagar Patel, Rahul Patel and Jaymin Bhalani, "Performance Analysis & Implementation of different Modulation Techniques in Almouti MIMO Scheme with Rayleigh Channel", International Conference on Recent Trends in computing and Communication Engineering, April 2013
[6] Jigar Kumar, Arpita Patel and Sagar Patel, "Multiuser – MIMO Broadcast Channel Techniques", International Research Journal of Engineering and Technology, Vol. 3, No. 2, Feb. 2016