Design and Implementation of I2C BUS Protocol on Xilinx FPGA.

Meenal Pradeep Kumar

Thesis Submitted in fulfillment of the requirements for the Master of Science

Jun 2017

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to the Electrical and Electronic Engineering Department of Universiti Sains Malaysia (USM) for providing me the chance to undertake this remarkable Master in Science Course. A very special thanks to Intel as part of project contribution was due to the working experience in Intel. I would like to express my gratitude to all my team members and coworkers for their encouragement and support.

Nevertheless, I would like to take this opportunity to thanks my supervisor, Dr. Mohamad Khairi Ishak, who was always willing to assist me and provided good support throughout the project completion. His excellent support, patience and effective guidance brought a great impact to my project as well as me.

Besides that, I would like to thanks USAINS and USM for arranging various workshop as support and knowledge to assist the student in the project, the workshop were indeed very helpful and insightful to me. I would like to thanks all lecturers from Universiti Sains Malaysia who had given me guidance throughout the period of the project.

Last but not least, I would also like to take this opportunity to express my deepest thanks to all relative third party member who had given me guidance indirectly to complete this thesis. My heartfelt gratitude goes to my family and friends for providing me continuous support throughout this thesis.

TABLE OF CONTENT

ACKNOWLEDGEMENT i

LIST OF FIGURES vii

LIST OF TABLES ix

ABSTRAK x

ABSTRACT xi

CHAPTER 1 1

INTRODUCTION 1

1.1 Background 1

1.2 Problem Statement 3

1.3 Objectives 4

1.4 Research Scope 5

1.5 Thesis Outline 6

CHAPTER 2 8

LITERATURE REVIEW 8

2.1 Introduction 8

2.2 Introduction to I2C Protocol System 8

2.4 Data Transfer 11

2.5 START and STOP Condition 12

2.6 Byte Format Data Transfer 13

2.7 Addressing 14

2.8 Acknowledge (ACK) and Not Acknowledge (NACK) 15

2.9 Read and Write Operation 16

2.9.1 Write Operation... 17

2.9.2 Read Operation.. 17

2.10 Addressing Mode.. 18

2.10.1 Seven bit Addressing mode.. 18

2.10.2 Ten bit Addressing mode 19

2.11 Combined Message.. 20

2.12 Speed.. 21

2.12.1 Standard Speed... 21

2.12.2 Fast Speed.. 21

2.12.3 High Speed 22

2.13 Pattern 23

2.13.1 Increment.. 23

2.13.2 Fibonacci 23

2.13.3 Gray Code... 23

2.14 Design of I2C Protocol 24

2.14.1 Design and Implementation of FPGA Based Interfaced Model for Scale-Free Network Using I2C Bus Protocol on Quartus11 6.0 24
2.14.2 Design and Modeling of I2C Bus Controller 25
2.14.3 Design and Implementation of I2C Communication Protocol on FPGA EEPROM 26
2.14.4 Design of I2C Master with Multiple Slave 27
2.14.5 Implementation of I2C Master Bus Controller on FPGA 28
2.14.6 Summary of Design and Implementation of I2C 29
2.14 Summary 33
CHAPTER 3 34
Methodology 34
3.1 Project requirements 38
3.2 Programming Flow 39
3.2.1 Programming Flow of I2C Master 39
3.2.2 Programming Flow of Test Card act as Slave 41
3.2.3 Flow Chart of I2C in Different Speed 43
3.2.4 Flow chart of Different Type Addressing Mode 46
3.3 Hardware Setup 47
3.3.1 Connection of FPGA and Test Card 48
3.3.2 Xilinx Tool 50

3.3.3 USB Blaster 52

3.3.4 Logic analyzer 52

3.4 Summary 54

CHAPTER 4 55

Result and Discussion 55

4.1 Introduction 55

4.2 Type of Mode of transfer 56

4.2.1 Write Transaction 56

4.2.2 Read Transaction 57

4.3 Combined Message 58

4.3.1 Write Read Transaction 58

4.3.2 Read Write Transaction 60

4.4 Addressing Mode 61

4.4.1 7 bit Addressing Mode 61

4.4.2 10 bit Addressing Mode 62

4.5 Randomize number of bytes 63

4.6 Randomize start address 64

4.8 Clock Frequency 65

4.8.1 Standard Speed 65

4.8.2 Fast Speed 66

4.8.3 High Speed 67

4.9 Pattern 68

4.9.1 Increment 68

4.9.2 Gray Code 69

4.9.3 Fibonacci 70

4.10 Automation Regression 71

4.11 Summary 73

CHAPTER 5 74

Conclusion and Future Works 74

5.1 Conclusion 74

5.2 Future Work 75

REFERENCES 77

LIST OF FIGURES

Figure 2.1: Relationship between Master and Slave 9
Figure 2.2: Data Validity Condition 12
Figure 2.3: Start and Stop condition 13
Figure 2.4 : Data Transfer of I2C 14
Figure 2.5: Slave Address 14
Figure 2.6: ACK of I2C Bus 16
Figure 2.7: Write Mode 17
Figure 2.8: Read Mode 18
Figure 2.9: 7 bit Addressing Mode 18
Figure 2.10: 10 bit addressing Mode 19
Figure 2.11: Combined Message 21
Figure 2.12: First 21 Fibonacci Numbers 23
Figure 2.13: Conversion of Binary to Gray Code 24
Figure 2.14: I2C Controller with master and multiple Slave 27
Figure 2.15: Pin Assignment of DS1307 28
Figure 3.1: State Diagram of Proposed Methodology 37
Figure 3.2: Software and Hardware Requirement 38
Figure 3.3: I2C Controller with Master Mode 39
Figure 3.4: Flow chart of Test Card act as Slave 41
Figure 3.5: Flow Chart of I2C Controller with Different Speed 44
Figure 3.6: Flowchart of I2C with different Addressing Mode 47

Figure 3.7 FPGA and Test Card Connection 48

Figure 3.8 Xilinx FPGA and FMC 49

Figure 3.9 Xilinx Software Tool 51

Figure 3.10 Xilinx Hardware Tool 51

Figure 3.11 USB Blaster 52

Figure 3.12: Logic analyzer Protocol Signal and Data Packet 53

Figure 3.13: Colour Coding for each Condition 53

Figure 4.2: Read Transaction 57

Figure 4.3: Write Read Transaction 58

Figure 4.4: Read Write transaction 60

Figure 4.6: 10 bit Addressing Mode Signal 62

Figure 4.7: Different number of byte 63

Figure 4.8: Different type of start address 64

Figure 4.9: Standard Speed 65

Figure 4.10: Fast Speed 66

Figure 4.11: High Speed Signal 67

Figure 4.12: Increment Pattern 68

Figure 4.13: Gray Code Pattern 69

Figure 4.14: Fibonacci Pattern 70

LIST OF TABLES

Table 2.1: Comparison with Other Research Papers 32

Table 3-1: XM105 Features 49

Table 3-2: Pin connection on XM105 50

Table 4.1 Gray Code in Decimal and binary 70

Table 4-2 Randomization Parameters 71

Table 4-3: Long Hour Regression Result 73

ABSTRAK

Tujuan utama penyelidikan ini adalah untuk mereka bentuk dan melaksanakan I2C protokol dengan ciri ciri I2C yang berbeza seperti mesej bergabung, mod address, pattern data yang berlainan, pemulaan address yang berbeza, kelajuan I2C protokol, dan mod yang berlainan Antara FPGA dan Kad Ujian. Melalui Kad Ujian, masalah signal integriti dapat dikurangkan kerana kad ujian boleh bertindak sebagai peranti sebenar. Selain itu, I2C merupakan satu IP yang boleh mengurangkan kos dan kerumitan kerana I2C mempunyai dua signal sahaja. Selain itu, ciri ciri ini akan dirawakan dan diuji untuk masa yang panjang. FPGA akan bertindak sebagai master dan kad ujian akan bertindak sebagai 'slave'. Master akan menjana 'START' bit dan semasa ini berlaku, transaksi dari tinggi ke rendah akan berlaku pada SDA dan SCL akan kekal tinggi sahaja. Di samping itu, Master juga akan menjana 'STOP' bit. Semasa ini, SCL akan berada pada tahap yang tinggi dan transaksi dari rendah ke tinggi akan berlaku pada SDA. I2C protokol mempunyai beberapa mod mesej. Anataranya ialah 'read' transaksi, 'write' transaksi, 'write-read' transaksi dan 'read-write' transaksi. Selain mod mesej yang berlainan, I2C protokol juga mempunyai mod address yang berbeza. Mod address yang telah dilaksanakan dalam penyelidikan ini ialah mod address 7 bit dan 10 bit. Penyelidikan ini juga bertumpu untuk merawakan bait data and permulaan address. Selepas operasi I2C protokol ini dilaksanakan, signal akan dianalisis mengunakan 'Logic analyzer'. Signal ini akan dibandingkan dengan I2C protokol. Di samping itu, ujian tekanan juga akan dilaksanakan dengan merawakkan ciri ciri I2C protokol dan diuji untuk masa yang panjang iaitu selama 4 jam. Ujian tekanan ini dilaksanakan untuk menguji I2C IP dan memastikan IP adalah sihat.

ABSTRACT

The focal point of this research is to design and implement the Inter-Integrated Circuit (I2C) protocol with different types of features such as combined message, addressing mode, different data pattern, different start address, clock frequency, and type of mode between the FPGA and Test card. By using test card, signal integrity issue will be able to reduce as test card will be able to replicate the actual device. I2C IP is also able to reduce the cost and complexity issue as it consists of two signal. All of this features will able to randomize and run for long hours. The field-programmable gate array (FPGA) will act as master and test card as slave. As the design architecture consists of master and slave, the master will generates a START condition and at this condition the serial data (SDA) will have a transaction between high level to low level and serial clock (SCL) will remain high. Besides that, Master will also generate STOP condition. At STOP condition, SCL is HIGH and SDA will have a transaction from LOW to HIGH. Additionally, there are a few type of messaging mode such as read transaction, write transaction, write-read transaction and read-write transaction. All this messaging mode will have its own protocol. On the other hand, master also transfers and received data to or from slave devices by different addressing mode. The addressing mode that is implemented are 7 bit addressing mode and 10 bit addressing mode. This thesis is also concerned by randomizing the data byte send and start address. The data send, read and write particularly these operations are carried out and stimulate by capturing signal using logic analyzer. The signal is then examined and compared with the actual I2C protocol format. A stress test is also done by randomizing all the features and running

for long hours which is 4 hours. This stress test is carried to stress the IP and make sure the IP is healthy.

CHAPTER 1

INTRODUCTION

I2C is commonly known as Inter Integrated Circuit. It is a synchronous bus protocol. Besides that, it also allows communication with less or even zero data loss. On the other hand, I2C is used for signal processing device to control interface. In this chapter, the background of I2C is explained in detail followed by the problems and drawbacks of I2C. Besides that, objective and scope for this thesis has been discussed. Last but not least, it summarize the outline of the thesis.

1.1 Background

This section provides a background overview for the whole research scope. I2C stands for Inter-Integrated Circuit. The I2C bus protocol was developed by Philips Electronics [2]. I2C allow communication between integrated circuits. Its communication are from different manufactures [1]. I2C is a synchronous bus protocol. This synchronous bus protocol will able the faster device to communicate without any data loss. Besides that, I2C is commonly used for signal processing device [3].

I2C protocol is a two-wire serial interface or even know a bidirectional wire interface. These bidirectional wire consists of serial data (SDA) and serial clock (SCL). Firstly, SDA line is where data goes on this line when it sent from one device to another device [3]. Secondly, SCL line is generated by the master device and control when data is sent and when it is in read operation. Both these wire, SDA and SCL line carry messages between the master and slave which is connected in the bus [4]. These master or slave is the devices that is connected on the bus. A unique address represent each device on the bus and these device has the functionality to act as transmitter or receiver [6].

Device on the I2C bus are able to act as master or slave. A node on the bus will act as master when this specific node generates clock. Besides that, this master node should also initiates a communication with the slave. On the other hand, a slave node will received the clock that is generated by the master. Besides that, this node will also respond when addressed by the master [5]. A bus device can act as four modes. A bus device can act as Master and Slave. Each of the Master and slave can act as Transmit and Slave. Therefore in total there will be four mode of operation which is Master Transmit, Master Received, Slave Transmit and Slave Received. When a data is send to the slave, the master will act as Master Transmit and slave will act as slave receiver. Besides that, when the slave node ending the data to the master, the master will act as master received while slave will be slave transmit [4].

There are various speed that an I2C IP can run. It can cover 3 speeds category. The speed category includes is the standard speed, fast mode speed and high speed. The standard speed will be operating with 100 kHz. Besides that, the fast mode speed will be operating at speed of 400 kHz and the high speed will be operating at speed of 3.4 MHz [6].

In this research, implementation of I2C bus protocol with different features such as combined message, different type of mode transfer, different type of speed, addressing mode, different pattern of data transmitted and number of bytes. It is also the best bus for control application where devices may have to be added or removed from the system.

1.2 Problem Statement

I2C is a very important on chip integration for the future technology purpose. Therefore noise and signal integrity issue should be eliminated [1]. This is important as there are many drawbacks on I2C validation in terms on signal integrity and noise [3]. Therefore a validation though a test card which will able to replicate the actual device will able to reduce the signal integrity and noise issue.

Besides that, cost and complexity issue is always be major issue within the customers. I2C device is one of the best solution to reduce cost and complexity. This is because I2C consists of only two signal that are SDA and SCL compared with other IP such as RS232, RS485, QEP and many more [5]. In another hand, when many devices is connected on the bus, I2C IP is most suitable to be used. This is because I2C IP able to cut the cost and scale down the complexity of the circuit. On the other hand, chip select line is commonly used in many other IP such as SPI IP. This chip select is to select one or more than one set of integrated circuit which is connected on the same bus.

By using chip select line, component and complexity will increased [4]. Therefore cost will also increase. This issue is eliminated by adding addressing feature which able to support 7 bit addressing feature and 10 bit addressing feature and combined message that able to support read-write and write-read transaction.

The most common problem is when the IP is only able to run with a selected feature that is implemented and without any stress test. Healthiness of an IP is very important. Therefore stress test is important to test the healthiness of the IP and investigate the performance of data transfer and packet loss. Stress test can be done by doing a randomization among all the features that is implemented and run for long hours [2].

Therefore thesis is able to increase the healthiness of the IP by running randomization with all the features that is implemented and for long hour. Besides that, noise and signal integrity issue is also being to minimize as the use of test card which is able to replicate the actual device. Last but not least, complexity and cost of production will able to reduce.

1.3 Objectives

The objectives of this research project are as the following

- To design and implement I2C protocol with different types of features such as combined message, different type of addressing mode, different type of pattern, speed and different slave address with multiple I2C controller and test card.

- To investigate the performance of data transfer and packet loss during long hours running by randomizing the features in an automation regression.

1.4 Research Scope

This research scope will focus on the design and implementation of I2C protocol with different type of features such as combined message such as write read transaction and read write transaction, different type of addressing mode such as 7 bit and 10 bit addressing mode, different type of pattern such as increment, speed and different slave address with multiple I2C controller and test card. I2C protocol is being simulated by logic analyzer and the signal is compared with the protocol format.

Firstly, understand the behavior of I2C protocol work and the features of I2C protocol. To understand I2C protocol work by understanding the behavior of I2C protocol and application of I2C IP will be used. Besides that, each features has a different behavior on I2C protocol. For example, for read transaction, the slave is I2C controller and master is test card. Another example is the bus protocol for 7 bit addressing mode and 10 bit addressing mode which is different. After understanding each features bus protocol, design and implementation process happened. Design and implementation of I2C controller is done using Verilog coding. The design that been implemented is changed to bit file and flash on Xilinx FPGA. For test card programming, it is also done using Verilog Coding.

I2C protocol is simulated using logic analyzer to capture signal and data packet. This is then is compared with the actual protocol format. The test is run for long hours with 3 I2C controller and randomize with all the I2C features and performance of data transfer and packet loss are being investigated.

1.5 Thesis Outline

This thesis contains overall 5 chapters with the following detail discuss as below. There are five chapters in this thesis. Firstly, in Chapter 1, an introduction of I2C IP has been introduced and followed by the background of I2C. Next, the problem statement and objective had been discusses. Last but not least, the research scope and thesis outline is discussed.

Chapter 2 is the literature review chapter. In this chapter, I2C bus protocol and I2C bus term is discussed in detail. Next, I2C features is also been discussed. Some of the features that being discussed are write transaction, read transaction, combined message, data byte, pattern, slave address and many more. Last but not least, a comparison with other research papers had been discussed.

In Chapter 3, the methodology of this thesis had been discussed. Strategy and methodology is being discussed to design and implement I2C protocol. This thesis involve of hardware and software requirement. Both the hardware and software requirement had been discuss in details. This hardware and software requirement includes the hardware setup, hardware that being used and flow chart of programming I2C Controller and test card which replicate the actual device.

The result and discussion is discussed in Chapter 4. In this chapter, the signal that is being simulate by the Logic analyzer is being captured and discussed. Besides that, the test is being run in long hours and randomization of all I2C features is done for stress test. This stress test is to increase the coverage of I2C protocol. This stress test able to make sure the performance of data packet loss is almost zero and able to investigate the healthiness of the IP.

Finally, Chapter 5 provides a conclusion and recommendation on future works. The conclusion is a short summary of the whole thesis. The recommendation for the future works includes some improvement for a better coverage of I2C IP. The Better coverage of I2C IP will be able to make sure I2C is healthy and reduce bug escape.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides research study about I2C and different features of I2C. Section 2.2 discussed the background of I2C protocol system. Section 2.3 discussed the I2C bus term that being used in I2C Protocol. Next, Section 2.5 till Section 2.13 discuss the different type of features of I2C that will be implemented such as different type of mode, combined message, different start address, different speed, different number of byte and many more. Lastly, Section 2.14 discussed on background of different features and techniques that being implemented on I2C protocol.

2.2 Introduction to I2C Protocol System

I2C is commonly known as Inter Integrated Circuit. It is a synchronous bus protocol. I2C also known as bi-directional, two-wire and serial communication standard protocol which consists of SDA and SCL wire. SCL is known as the serial clock and SDA which is Serial Data that carry information between the ICs connected to them [3].SDA line is where data goes on this line when it sent from one device to another device [7] while SCL line is generated by the master device and control when data is sent and when it is in read operation. Both these wire, SDA and SCL line carry messages between the master and slave which is connected in the bus [9]. These master or slave is the devices that is connected on the bus. Figure 2.1 shows the relationship between Master and Slave.

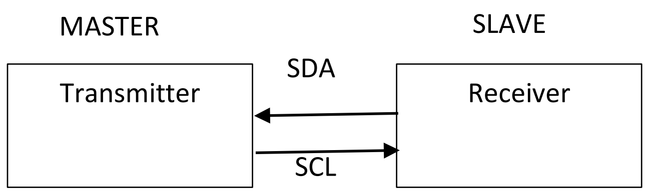

Figure 2.1: Relationship between Master and Slave [3]

IC Fabrication Process is also supported by the I2C bus. For example, NMOS, CMOS or bipolar. A unique address represent each device on the bus and these device has the functionality to act as transmitter or receiver [13]. For example, for a video decoder, the receiver will be the decoder. In other hand, memory will act as both receiver and transmitter. Device is not only the transmitter or receivers but this device can also be consider as master and slave [12].

2.2 I2C Bus Term

The following terms relate on how the I2C device and how it interacts with other I2C devices on the bus [13]. There are four terms in the I2C Bus. Firstly is the Transmitter. Transmitter is a device that send data to the I2C bus. This transmitter can act as master transmitter or slave transmitter [11]. A device that act as master transmitter trigger and begin the data transmission. Slave transmitter is a device that acknowledge request from the master. The request acknowledge is to send data to the I2C bus [12].

The second bus term is the receiver. It accept data from the I2C bus. Receiver can be either master receiver or slave receiver. Master receiver is a device that accept data on its own request. Slave receiver will acknowledge the request [11].

Besides that, another I2C bus term is master. Master will initiates a transfer with a START command. This Master will generates SCL signal. Besides that, master also terminate the transfer with a STOP command. This master are able to act as master transmitter or master receiver [12]. On the other hand, slave is also known as I2C Bus term. This is address by master. This address is known as slave address. This slave can be either receiver or transmitter [13].

2.3 Master and Slave

A device that able to initiate a data transfer will act as a master. Master will generate a clock signal and this will ensure the data that is generated on the bus is valid. On the other hand, the devices that is addressed will be known as slave [9]. A positive voltage supply is connected to both the line which is the SDA and SCL line. These line are connected by positive supply by a pull up resistor. Due to the pull up resistor, these lines SDA and SCL will remain high. The SDA and SCL line will remain high only when the I2C bus is not busy. A unique slave address is identified by an address. This device will able to act as a transmitter role or receiver role [14].

A device will act as a transmitter role when the device generate message or data on the I2C bus. On the other hand, receiving data or message from the I2C bus means the device is acting as a receiver. During the process of data and message transfer on the bus, the device can execute as a master or slave [15]. Master will initiates the data or message transfer on the I2C bus by generating clock signal and at the same point device addressed will be considered as a slave [13].Transmitter or receiver device could be master. For example, a master microcontroller will act as transmitter and send data to a RAM. Then the RAM is interrogate for its content. The RAM act as master receiver where it initiate the transfer. In other hand, a slave can also be both a receiver and a transmitter [16].

2.4 Data Transfer

In order for the data to be valid, SDA line will remain stable. SDA will remain stable when the clock pulse is at the HIGH period. At this moment, the data of the SDA line will remain stable. Changes of data is allowed when SCL is at LOW period. At this point, the change of data line will be known as control signal. Each byte is 8 bit long. Transmission of the 8 bit is serially [16]. The transmission is by transmitting the most significant bit is and followed with the acknowledge bit. The acknowledge bit is generated by the master. Figure 2.2 below show the data validity condition [18].

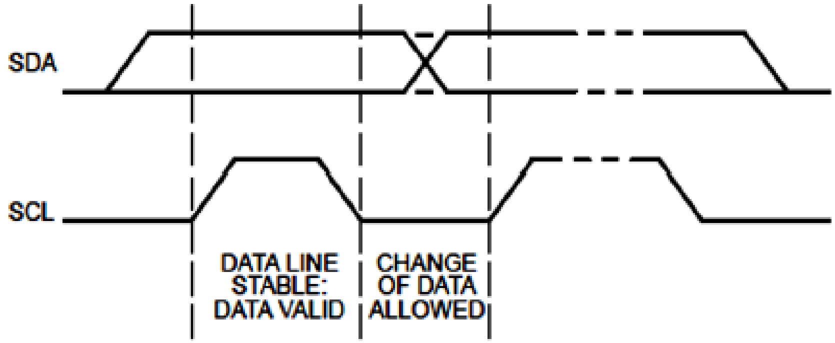

Figure 2.2: Data Validity Condition [18]

2.5 START and STOP Condition

A START and STOP always be as begin and terminate for I2C bus protocol. I2C transaction begin with a START and terminate with a STOP. A High to Low transition on the SDA line while SCL is high defines a START condition. A STOP condition is when there is a transaction of LOW to HIGH on the SDA line whereas SCL line is high [15]. Device which act as master will generate both the START and STOP condition. When the START condition is generated, the bus will be busy. On the other hand, the bus is not busy after certain time of STOP condition [16]. RESTART is also able to generate by the master instead of STOP. RESTART is also known as Repeated Start. This Repeated Start is generated instead of a STOP condition and if this happen, the bus will remain busy until a certain time of STOP condition. Figure 2.3 shows the graph of SDA and SCL signal during START and STOP condition [18].

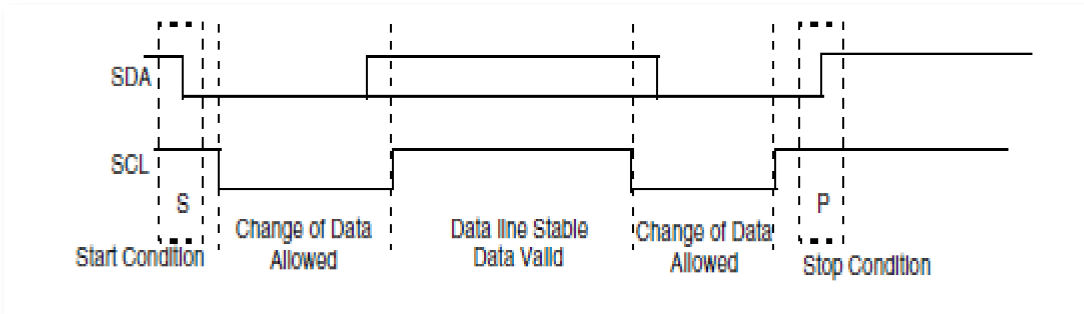

Figure 2.3: Start and Stop condition [18]

2.6 Byte Format Data Transfer

SDA line will be placed with bytes and every byte put on the SDA line must be 8 bits long [18]. In every transmission, the number of byte that can be transfer is unlimited but in one condition where every byte that being transfer should consists of Acknowledge bit. The transmission data is start with the most significant bit. Some other behavior will be happened if the slave is not able to do transmission or received. The behavior that it will perform is such as interrupt [19]. Figure 2.4 shows the signal for SDA and SCL when data transfer scenario happen on the I2C bus [17].

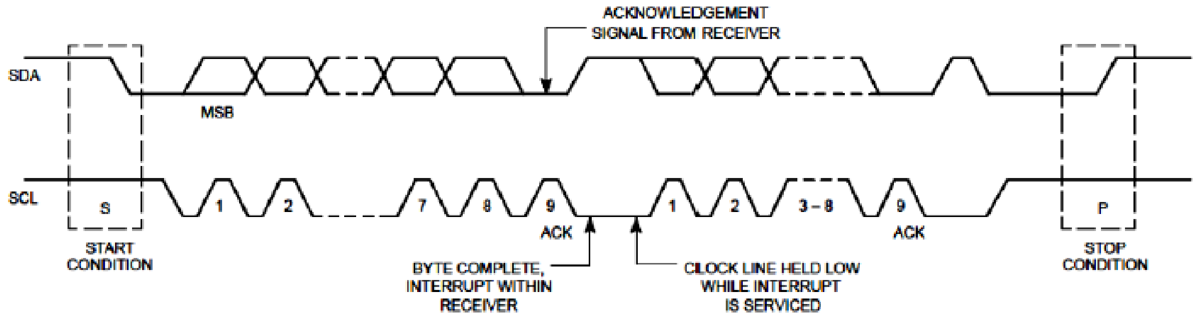

Figure 2.4 : Data Transfer of I2C [17]

2.7 Addressing

The addressing format for the I2C bus is unique. This is because the first byte after the START condition is selected by the master. This byte is known as slave address and this slave address able to determine which slave is selected. This slave is always select by the master. Figure 2.5 show the addressing for the slave address [19].

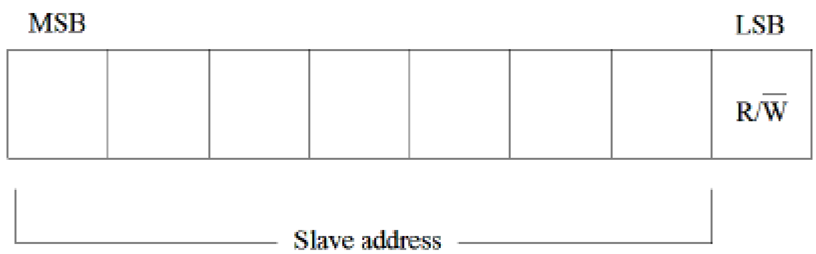

Figure 2.5: Slave Address [19]

The slave address is made up of the first seven byte. The eight bit of the byte which is also known as the least significant bit (LSB) arbitrate the command or even known as direction of the message [20]. A ‘zero’ is the LSB is ‘zero’, the direction is write. This means that the master will write information to a selected slave. On the other hand, if the LSB is ‘one’, it is a read direction. In a read direction, the master will read the message from the slave [19].

When the address is sent by the slave, the slave or which is known as the devices will compare the first seven bits after the START bit. The comparison is done with the device address [22]. If both the address from the device and address that is sent match, the device considers itself addressed by the master as a slave-receiver or slave-transmitter. Slave receiver or transmitter is depend on the LSB bits which determine read or write operation [20].

2.8 Acknowledge (ACK) and Not Acknowledge (NACK)

The ACK happens after every byte. The function of the master is to generate the clock pulses which include the ACK 9^{th} clock pulse. When the transmitter releases the SDA line during the ACK clock pulse and this causes the receiver of SDA line pull to low and will be stable. This happened during the ACK signal. This can be seen in detail in Figure 2.6 as below [21].

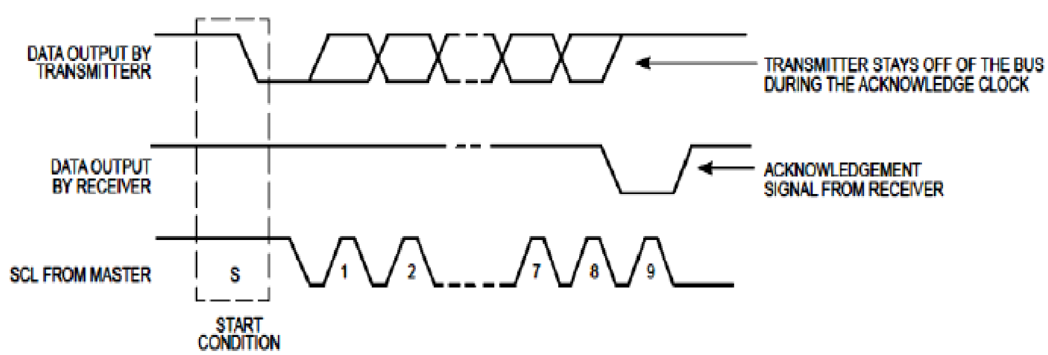

Figure 2.6: ACK of I2C Bus [21]

When SDA remains High during the 9^{th} clock pulse, this is defines as NACK signal. A STOP is generated by the master. This STOP bit to halt the transfer. Besides the STOP bit, master can also generate the REPEATED START bit instead of STOP bit. This REPEATED START is start a new transfer of data. To generate a NACK bit, there are a total of five situation [11]. The first condition is there should not be any receiver present on the I2C bus. The inexistence of the receiver will lead to no transmitter address and this cause no device to respond with an ACK [20]. Besides that, when the receiver is operating some real-time operation, the receiver should not able to receive or transmit. This is because, the receiver is not ready to start transmission with the master [19]. Thirdly, during the transfer, the receiver cannot receive any more data bytes. Last but not least, the master-receiver needs to send signal at the end of the transfer. This signal is send to the slave transmitter [22].

2.9 Read and Write Operation

A slave address is generated and send after the START condition. This slave address is 7 bits long followed by an eight bit, which is a data direction bit which is Read or Write. A ‘zero’ indicate a WRITE transmission and a ‘one’ indicates a READ transmission. A data transfer is always terminated by a STOP condition which is generated by a master. But, if a master still continue to communicate, master will generate a RESTART and another slave address.

2.9.1 Write Operation

Master transmitter transmit data to the slave receiver in this mode. Figure below shows a master transmit to slave receiver with a WRITE mode. Slave address is generated right after the START condition [20]. The last bit of slave address is zero for write operation. Addressed device then send ACK during the ninth pulse of SCL. After getting the ACK, master can transmit data of 8 bit continuously [23]. Each data byte also should be ACK. Then a STOP condition happen. The write operation can be observed in the Figure 2.7 [22].

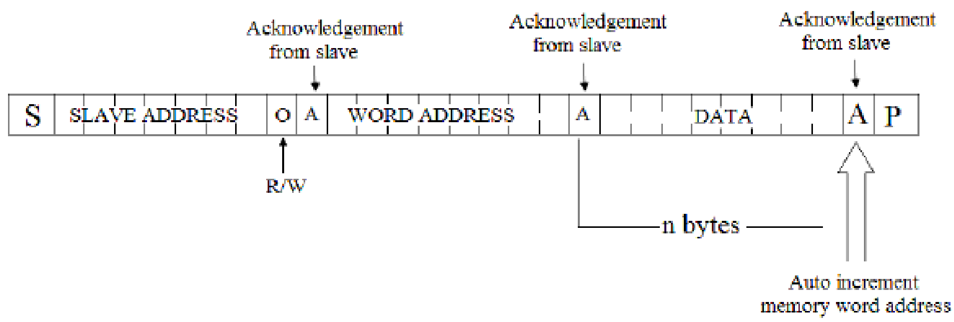

Figure 2.7: Write Mode [22]

2.9.2 Read Operation

For read mode, the last bit of slave address is 1. Master can read the slave data either by setting word address or directly by the first location [20]. Master write the word address then become receiver to read the data send by slave. Master is generating SCL while reading the data. The read operation can be seen as Figure 2.8 [21].

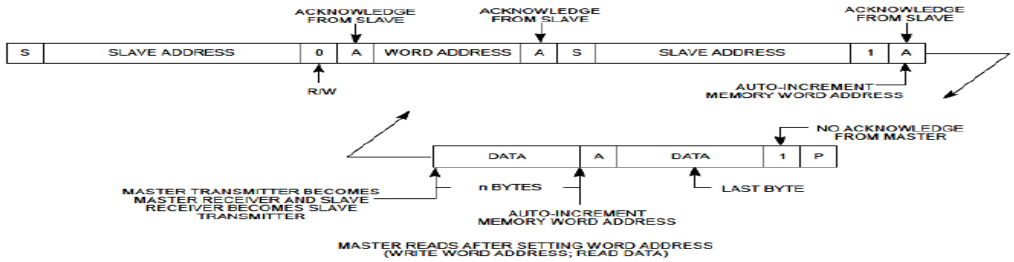

Figure 2.8: Read Mode [21]

2.10 Addressing Mode

2.10.1 Seven bit Addressing mode

The first seven bits of the first byte of the first byte set the slave address and the LSB bit 0 is the read or write (R/W) bit. Figure 2.9 shows the 7 bit addressing format [22].

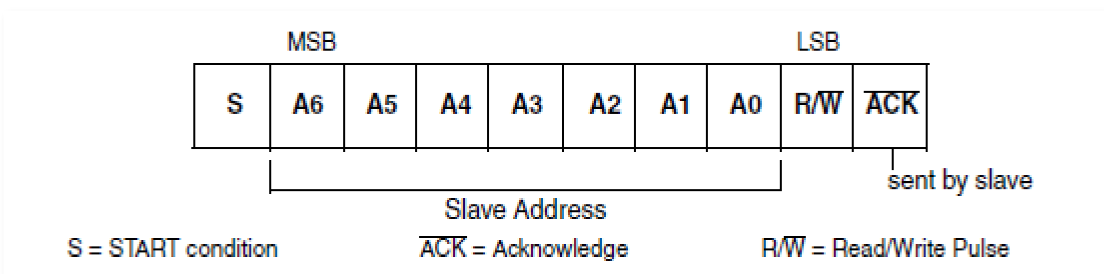

Figure 2.9: 7 bit Addressing Mode [22]

I2C will be in IDLE or not busy state when SDA and SCL are at high. If a Master want to transmit the data, Master will generate a Start Condition at bit 0. This will allow SDA line to go low and SCL will be high [20]. Then, the I2C bus will not be in idle state already. It will be at a busy state. Then Master will generate the slave address of 7 bits and then bit 8 to indicate whether is read or write. After the slave received the slave address bit and the 8th bit, slave will acknowledge by sending ACK to Master [24]. After the ACK bit, whoever is supposed to write the data, will start transmitting the data. After the data transmit, either slave or master who have to read the data will send an ACK pulse to the other side. When all data transmission is over, master will generate a STOP condition. When STOP condition happen, the CL signal will be high and SDA will go from Low to high [11].

2.10.2 Ten bit Addressing mode

The difference between 10 bit addressing mode and 7 bit addressing mode is in 10 bit addressing will be known by considering first two byte after START condition. During this mode, two bytes are transferred to set the 10 bit address [15]. When first five bits in first byte is 11110 then slave will come to know that it is 10 bit addressing mode. The 6th and 7th bit will be identify as the most significant bit (MSB) of the 10 bit address and as similar as 7 bit addressing mode, the 8th bit which is known as the least significant bit (LSB) will be either read or write. Figure 2.10 shows the 10 bit addressing mode [25].

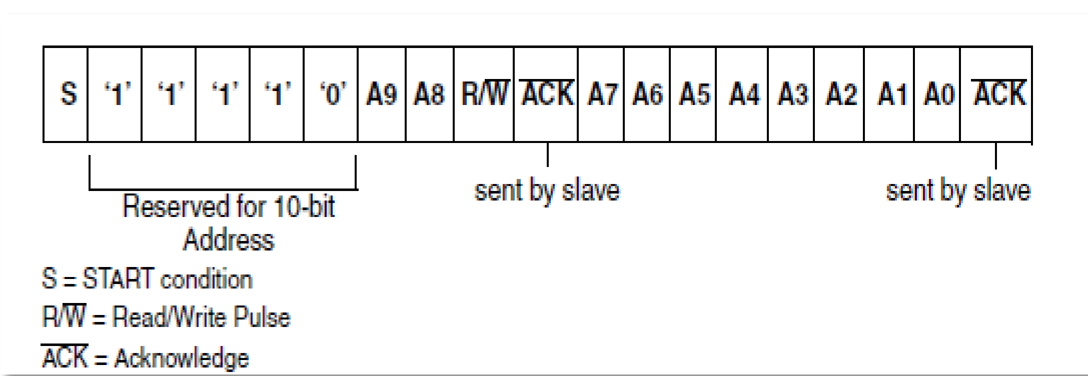

Figure 2.10: 10 bit addressing Mode [25]

For the LSB which is the 8th bit, when bit 0 is set to 0, it follow the write operation where master will write to slave whereas it the bit is set to 1, then master will read from slave. When the first byte is received by the slave, slave will generate an ACK to the master. Then the master will send register address and then the data transmission process happen similar to 7 bit addressing mode [22].

2.11 Combined Message

I2C support mixed read and write combined format transaction in both 7 bit and 10 bit addressing mode. Example of combined message are read write transaction or write read transaction. Combined message allows combined read or write operation to device without releasing the bus and thus with the guarantee that the operation is not interrupted. To initiate format transfer, a restart bit should be generated by the master [6]. With this bit generated by the master, the I2C controller complete the I2C transfer by checking the transmit FIFO and executes the next transfer. If the direction of this transfer differs from the previous transfer, the combined format is used to issue the transfer.

In the normal operation, after the slave address and the read or write bit, the master will send the data and followed with a STOP condition but for a combined message format, instead of generating a STOP condition, master will generate a RESTART Bit which is similar to the START bit. After the RESTART bit, it is followed with another slave address and read or write bit. After the read or write bit, data will be generated by the master. This scenario is explain in detail in the Figure 2.11 [28].

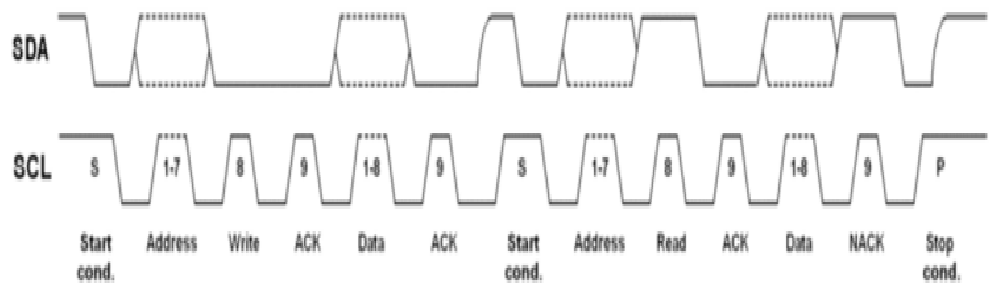

Figure 2.11: Combined Message [28]

2.12 Speed

The clock is transmitted by the sender and the receiver is always able to synchronize with that clock. I2C defines several speed such as standard speed, fast speed and high speed. These speeds are more widely used on embedded systems than on PCs.

2.12.1 Standard Speed

Standard mode refers to the initial transfer speed mode of the I2C specification which allow up to 100 kbit/s [4]. During this standard speed, the master will pull SCL to low. It will pull SCL to low when it observes the clock line going high with a minimal time of 4 μs.

2.12.2 Fast Speed

The protocol, logic levels and maximum capacitive load for the SDA and SCL lines are remain same for the fast speed. Besides that, during fast Mode speed device received and transmit at up to 400kbit/s [27]. Fast speed mode has a few features. One of the features is the bit rate will increase to the maximum which is 400 kbit/s. Besides that, the serial data (SDA) and serial clock (SCL) signals have timing which has been adapted. Compatibility with different bus such as CBUS is not needed. The only reason for this is they are able to operate at the increase bit rate [29]. Both signals of I2C which is the SDA and SCL signals must be variable. This is because the floating of both this IO pins will not able to disturb the other bus lines. Last but not least, to accommodate the short but maximum rise time, an external pull-up resistor is connected on the bus [19].

2.12.3 High Speed

The maximum for high speed device is 3.4Mbit/s. Device able to transfer information at the bit rate [4]. In order to allow a transmission of data up to 3.4Mbit/s. To achieve a bit transfer of up to 3.4Mbit/s, the following enhancements have been made to the regular I2C bus specification. Firstly, the high speed mode master device has an open drain output buffer and a combination of an open drain pull down and current source pull up circuit [26]. On the other hand, master device will produce a sequential clock signal with a HIGH to LOW transaction. This transaction is a ratio of 1 to 2. This ratio is able to dismiss the timing requirements for set-up and hold times. Next, the input of High speed mode devices include spike destruction and is Schmitt trigger at the SDA and SCL inputs [30]. Lastly, the output buffer of high speed mode integrate slope control. This slop control is the falling edge of the SDA and SCL signal [12].

2.13 Pattern

Data being transmitted from master to slave can be transmitted in various pattern. Some of the pattern that can be implemented are increment pattern, Fibonacci pattern and gray code pattern. All of this pattern has its own unique pattern.

2.13.1 Increment

The data that being transmitted in the increment data pattern is from ascending order to descending order. The data that being transmitted start from 0x0 followed by 0x1 and so on. The data increases by 0x1 after every data.

2.13.2 Fibonacci

Fibonacci numbers are the numbers in the following integer sequence called the Fibonacci sequence and characterized by the fact that every number after the first two is the sum of the two preceding ones. For example, the data will be 0, 1, 1, 2, 3, 5, 8, 13 and so on. The first 21 Fibonacci numbers F_n for n is from 0 to 20. Figure 2.12 shows the data of first 21 Fibonacci numbers [25].

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597	2584	4181	6765

Figure 2.12: First 21 Fibonacci Numbers [25]

2.13.3 Gray Code

Gray Code which is also known as reflected binary code. It is binary numeral system where two successive values differ in only one bit. The reflected binary code was originally designed to prevent spurious output from electromechanical switches. The

Gray Code for decimal 15 rolls over to decimal 0 with only one switch change [15]. This is called the "cyclic" property of a Gray Code. In the standard Gray Coding the least significant bit follows a repetitive pattern of 2 on, 2 off, 4 on, 4 off and so forth. Figure 2.13 shows the conversion of binary to gray code [11].

Decimal	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Figure 2.13: Conversion of Binary to Gray Code [11]

2.14 Design of I2C Protocol

2.14.1 Design and Implementation of FPGA Based Interfaced Model for Scale-Free Network Using I2C Bus Protocol on Quartus11 6.0

I2C bus consists of SDA and SCL lines and this is known as the main specification of I2C. Both this SDA and SCL lines are bi-directional. These lines are connected to a positive voltage with a pull-up resistor. The data on the SDA line in consider valid when the SCL line is low. Otherwise it is not consider valid. Besides that, data transfer is also another I2C specification. Every data is placed on the SDA line. This data bits are transfer after every start condition and end whenever a STOP condition is triggered.