# **DESIGNING OF INTER INTEGRATED CIRCUIT USING VERILOG**

### **DISHA MALIK**

Masters of Technology Scholar, Department of Electronics & Communication Engineering, Jayoti Vidyapeeth Women's University, Jaipur- INDIA

## ABSTRACT

This paper primarily deals with the designing of inter integrated circuit (I2C) using verilog in Modelsim-Altera 6.4a(Quartus 2<sup>nd</sup> edition 9.0).The I2C Bus is a two-wire, bidirectional serial bus that provides a simple and efficient method of data transmission over a short distance between many devices. It provides good support for communication with various slow, on-board peripheral devices that are accessed intermittently, while being extremely modest in its hardware resource needs. It is a simple, low bandwidth, short distance protocol which supports multiple masters. Masters and slaves can receive and transmit data bytes. The operating speed modes of I2C are as follows:

- Standard-mode (Sm) up to 100 kbps
- Fast-mode (Fm) up to 400 kbps
- Fast-mode Plus (Fm+) up to 1 Mbps
- **High-speed mode (Hs)** up to 3.4 Mbps.

**KEYWORDS:** I2C bus, Master, Slave, Finite State Machine, Verilog ModelSim.

## **1. INTRODUCTION**

The Interconnect Integrated Circuit or I2C interface was originally developed by Philips Semiconductors Company for data transfer among ICs at the Printed Circuit Board (PCB) level in early 1980s. All I2C-bus compatible devices have an interface allowing them to communicate directly with each other via the I2C-bus. The concept provides an excellent solution for problems in many interfacing in digital design. I2C is now broadly adopted by many leading chip design companies like Intel, Texas Instrument, Analog Devices, etc.

The I2C Bus is a two-wire, bi-directional serial bus that provides a simple and efficient method of data transmission over a short distance between many devices. Prior to I2C, chip-to-chip communications used many pins in a parallel interface. Many of these pins were used for inter-chip addressing, selection, control, and data transfers. In a parallel interface, 8 data bits are typically transferred from a sender IC to a receiver IC in a single operation.I2C performs chip-to-chip communications using only two wires in a serial interface, allowing ICs to communicate with fewer pins. The two wires in the I2C Bus carry addressing, selection, control, and data, one bit at a time. The Data (SDA) wire carries the data, while the Clock (SCL) wire synchronizes the sender and receiver during the transfer. ICs that use the I2C Bus can perform the same function as their larger parallel interface counterparts, but with far fewer pins.

I2C provides good support for communication with various slow, onboard peripheral devices that are accessed intermittently, while being extremely modest in its hardware resource needs. It is a simple, lowbandwidth, short distance protocol. I2C is easy to use to link multiple devices together since it has a built-in address .The two I2C signals are serial data (SDA) and serial Clock (SCL).The device that initiates a transaction on the I2C bus is termed the master. The master normally controls the clock signal. A device being addressed by the master is called a slave.

The I2C protocol supports multiple masters, but most system designs include only one. There may be one or more slaves on the bus. Both masters and slaves can receive and transmit data bytes. Standard I2C devices operate up to 100Kbps, while fast-mode devices operate at up to 400Kbps. Most of the I2C devices available today support 400Kbps operation. Higher speed operation may allow I2C to keep up with the rising demand for bandwidth in multimedia and other applications.

In the figure below, there is one Master; the other devices are all Slaves. When a Master wants to initiate a communication, it issues a "START" condition. At that time, all devices, including the other Masters, have to listen to the bus for incoming data. After the "START" is issued, the Master sends the "ADDRESS" of the Slave that it wishes to communicate with along with a bit to indicate the direction of the data transfer (either read or write). All Slaves will then compare their addresses with the address received on the bus. If the addresses are identical, the Slave with the matching address will send an (ACK) "ACKNOWLEDGEMENT" to the Master. Slaves whose addresses do not match will not send an ACK. Once communication is established, the two lines are busy. No other device is allowed to control the lines except the Master and the Slave which was selected. When the Master wants to terminate communication, it will issue a "STOP" signal. After that, both SCL line and SDA line are released and free.



Block diagram of an i2c configuration. Multiple integrated circuits are aligned on a two-wire bus. Assigning unique addresses allows a master device to control multiple slave devices.

## 2. MASTER AND SLAVE

A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave. Both lines are connected to a positive supply via a pull-up resistor, and remain HIGH when the bus is not busy. Each device is recognized by a unique address whether it is a microcomputer, LCD driver, memory or keyboard interface and can operate as either a transmitter or receiver, depending on the function of the device. A device generating a message or data is a transmitter, and a device receiving the message or data is a receiver.

A passive function like an LCD driver could only be a receiver, while a Micro controller or a memory can both transmit and receive data. When a data transfer takes place on the bus, a device can either be a master or a slave. The device which initiates the transfer, and generates the clock signals for this transfer, is the master. At that time any device addressed is considered a slave. It is important to note that a master could either be a transmitter or a receiver; a master microcontroller may send data to a RAM acting as a transmitter, and then interrogate the RAM for its contents acting as a receiver in both cases performing as the master initiating the transfer. In the same manner, a slave could be both a receiver and a transmitter. The I2C is a multi master bus. It is possible to have, in one system, more than one device capable of initiating transfers and controlling the bus. A microcontroller may act as a master for one transfer, and then be the slave for another transfer, initiated by another processor on the network. The master/slave relationships on the bus are not permanent, and may change on each transfer.

As more than one master may be connected to the bus, it is possible that two devices will try to initiate a transfer at the same time. Obviously, in order to eliminate collisions bus and communications chaos, an arbitration procedure is necessary. The I2C design has an inherent arbitration and clock synchronization procedure relying on the wired-AND connection of the devices on the bus. In a typical multi system, а microcontroller master program should allow it to gracefully switch between master and slave modes and preserve data integrity upon loss of arbitration.

### **2.1 START AND STOP**

When a Master wants to initiate a data transfer, it issues a Start condition and when it wants to terminate the transfer, a Stop condition will be initiated. There can be multiple Starts during one transaction called a repeated Start. The Master can then release the Stop condition whenever it wants to. In Figure, a Start is issued by bringing the SDA line low while the SCL line is high. After that the Master controls the SCL line and can generate clock signals. A Stop condition is implemented by transitioning the SDA line high while the SCL line is high.



### **2.2 STARTING BYTES**

The I2C bus is a byte-oriented protocol. After signaling Slaves by the Start condition, the Master sends "starting bytes" to the Slave. There are two components that make us the "starting bytes": Slave address and data direction (Read or Write). The Master sends the MSB (Most Significant Bit) first and the LSB (Least Significant Bit) last. There are two addressing modes in the I2C protocol: the 7-bit and 10-bit address modes. We will first consider the 7-bit addressing mode. After the START condition (S), a Slave address is sent. This address is the first 7 bits, the eighth bit is a data direction bit (RnW). If the direction bit is '0', it indicates a transmission (or WRITE). If the bit is '1', it indicates a request for data (or READ). A data transfer is always terminated by a STOP condition (P) generated by the Master. However, a Master can generate a repeated START condition (Sr) and address another Slave without first generating a STOP condition.



When the I2C bus became more popular, it was recognized that the number of available addresses in the 7bit addressing mode is too small. Therefore, a new addressing mode (the 10-bit mode) was developed. The new addressing mode also supports the old one. Devices with 7-bit addresses can be connected with devices with 10-bit addresses on the same bus. In this mode, the first two bytes are dedicated for address and data direction. The format of the first byte is 11110xx; the last two bits of the first byte, combined with eight bits in the second byte form the 10-bit address.

### **2.3 ACKNOWLEDGEMENT**

Acknowledgement is obligatory in order to inform the transmitter that data has been successfully transmitted. Figure the acknowledgement illustrates mechanism. The Master generates the acknowledgement related clock pulse and the transmitter releases the SDA line (HIGH) during the acknowledge clock pulse so that the receiver can take control of the SDA line. If the receiver does not acknowledge, leaving the SDA line high, the transfer must be aborted. If it acknowledges by pulling the SDA line low, the transmitter knows that data has been successfully received, so it keeps sending data to the receiver.



### **2.4 A COMPLETE DATA TRANSFER**

All of the major aspects of I2C bus discussed so far are combined to create

a complete data transfer from the transmitter to the receiver as in Figure 2.7. The SCL signal, Start and Stop signals, and the first byte must be generated by the Master. The Acknowledgement of the first byte must be generated by the Slave when it recognizes its address on the bus. The other Acknowledgements are generated by the receiver.

Write Byte	Format
------------	--------

S	Address	WR	ACK	Command	ACK	Data	ACK	Р
	7 Bits		8 Bits			8 Bits		
Slave Address				Command Byte: se which register you writing to.	elects are	Data Byte: into the re by the con	: data goe gister set nmand by	es rte.

#### Read Byte Format

s	Address	WR	ACK	Command	ACK	S	Address	RD	ACK	Data	NACK	Р
	7 Bits			8 Bits			7 Bits			8 Bits		
Slave Address Command Byte: selects which register you are reading from.				s	Slave Address: repeated Da due to change in data- flow direction. cc			Data Byte the regise commane	e: reads f ter set by d byte.	rom the		

#### Receive Byte Format

S	Address	RD	ACK	Data	NACK	Р	
	7 Bits			8 Bits			
S = S	TART Conditio		Data Byte: reads data from				
P = S	TOP Condition		the register commanded by				
Shad	ed = Slave Tra	the last Read Byte or Write					

Byte transmission.

### **2.5 7 BIT ADDRESS**

SLAVE ADDRESS	R/W BIT	DESCRIPTION
000 000	0	General call address
000 000	1	START byte
0000 001	Х	CBUS address
0000 010	Х	Reserved for different bus format
0000 011	Х	Reserved for future purposes
0000 1XX	Х	Hs-mode master code
1111 1XX	Х	Reserved for future purposes
1111 0XX	Х	10-bit slave addressing

# **3. RESULTS**

## **3.1 SIMULATION AND TIMING DIAGRAM FOR READ OPERATION**



### SIMULATION AND TIMING DIAGRAM FOR WRITE OPERATION



## 4. CONCLUSION

In this paper designing of inter integrated circuit (I2C) using verilog in Modelsim- Altera 6.4a (Quartus 2<sup>nd</sup> edition 9.0) we have designed an inter integrated circuit (I2C). The I2C Bus is a two-wire, bi-directional serial bus that provides a simple and efficient method of data transmission over a short distance between many devices. Simulation is done with the help of verilog coding in Modelsim- Altera 6.4a edition.

## REFERENCE

1. ST24C02, user manual by ST MICROELECTRONICS.

I2C-Bus Specification, Version
 January 2000.

Frederic Leens. An Introduction
 to I2C and SPI Protocols, IEEE
 Instrumentation & Measurement
 Magazine February 2009.

4. J. M. Irazabel & S. Blozis, Philips Semiconductors, "I2C Manual," Application Note, ref. AN10216-0, March 24, 2003. 5. Abinesh R. Bharghava, M.B.Srinivas. Transition Inversion based Low Power Data Coding Scheme for Synchronous Serial Communication, 2009 IEEE Computer Society Annual Symposium on VLSI.

6. S. Sobhan, S. Das and I. Rahman, "Implementation of I2C Using System Verilog And FPGA", International Conference on advancement in electronics and power Engineering Bangkok December 2011.

7. Breckling A. Khan, A. Thakare,
"FPGA Based Design & Implementation of Serial Data Transmission Controller"
International Journal of Engineering
Science and Technology 2010, pp 5526
- 5533.