International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:9, No:10, 2015

I²C Master-Slave Integration

Rozita Borhan, Lam Kien Sieng

Abstract**—**This paper describes I²C Slave implementation using I²C master obtained from the OpenCores website. This website provides free Verilog and VHDL Codes to users. The design implementation for the I²C slave is in Verilog Language and uses EDA tools for ASIC design known as ModelSim from Mentor Graphic. This tool is used for simulation and verification purposes. Common application for this I²C Master-Slave integration is also included. This paper also addresses the advantages and limitations of the said design.

Keywords**—**I²C, master, opencores, slave, verilog, verification**.**

I. Introduction

I²C consists only of two bidirectional serial lines or buses that can support both multi master and multi slave designs. These 2 lines are known as Serial Data Line (SDA) and Serial Clock Line (SCL). However, for this particular I2C master design obtained from the OpenCores website, the I2C Master can only support single master with no bus arbitration process [1]. Typical voltages used are +5V/+3.3V.

This I²C Master Design also supports 7 bit address mode which can continuously perform read and write operations. It also provides an interrupt signal, buffer empty flag during transmission and buffer full flag during receiving mode [1]. SCL frequency for this I²C Master Design is the output frequency of ¼.

Every I²C transmission begins with a START and ends with a STOP command. Single message can be where a master writes data to a slave or where master reads data from a slave. For combined messages, master can issue at least 2 reads and/or writes to one or more slaves depending on the individual designs.

In a combined message, each read or write begins with a START and the slave address. After the first START in a combined message, there is also a signal called 'Repeated START' bit. Repeated START bits are not preceded by STOP bits, which is how slaves know the next transfer is part of the same message [2]. Any given slave will only respond to certain messages, as specified in its documentation.

During transmit mode, the master will initialize the START bit followed by the 7 –bit address of the slave the master wishes to communicate with. A single bit indicating whether the master wishes to write or read from the slave will be sent as well.

Corresponding slave exists on the bus, will respond with an ACK bit (active low). The master will then continue in transmit or receive mode according the WRITE/READ bit sent. The slave will then act respectively. The address and the data bytes are sent with the most significant bit first.

Rozita Borhan and Lam Kien Sieng are with MIMOS Bhd, Teknologi Park Malaysia, 57000 Kuala Lumpur, Malaysia (Phone: +603-89955000; Fax: +603-89914234; e-mail: ita@mimos.my, lksieng@mimos.my).

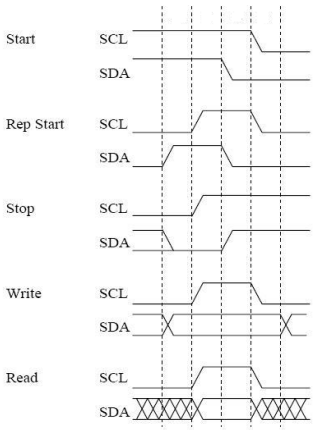

Fig. 1 I²C Timing Operation [1]

Fig. 1 shows the waveform for the START bit. The SDA will be a high to low transition with SCL high. For STOP bit, the SDA will be a low to high transition of SDA with SCL high as well. Except for these 2 conditions, all other transitions of SDA, SCL will be low. The data is being read while SCL is high.

II. Top Level Design

Fig. 2 shows the block diagram for the integration of I²C slave with the master. It also shows the input, output pins and the interconnection between them.

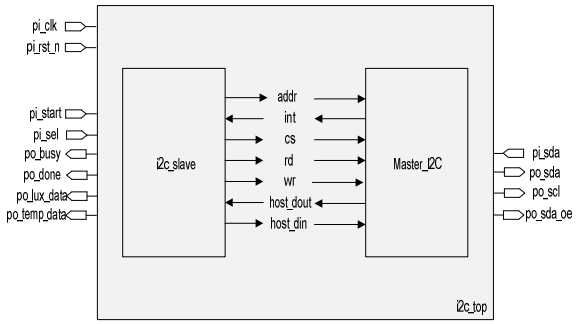

Fig. 2 I²C Top Level Block Diagram

A. Pin Descriptions

Table I shows the input and output pins of the Slave Design. The width, direction and the functions of the signals are also shown. Table II describes about the internal signals between I²C master and slave modules.

TABLE I
I²C PIN DESCRIPTION

Pin Name	Width	Direction	Function
pi_clk	1	Input	Clock
pi_rst_n	1	Input	Active low reset
pi_start	1	Input	Pulse signal to trigger the start of the operation
po_done	1	Output	Pulse signal indicates done of the operation
po_rdata	16	Output	Pulse signal indicates read data bus
pi_sda	1	Input	Serial data in
po_sda	1	Output	Serial data out
po_scl	1	Output	Serial clock out
po_sda_oe	1	Output	Serial data enable

TABLE II
I²C INTERNAL SIGNAL BETWEEN I²C MASTER AND SLAVE

Pin Name	Width	Function
cs	1	Chip select
addr	3	Address input
rd	1	Read function
wr	1	Write function
host_din	8	Data in to Master
host_dout	8	Data out from Master
int	1	Interrupt signal

III. THE IMPLEMENTATION

Fig 3 shows a flowchart on how the I²C Slave is designed in response to the I²C Master. The state starts at IDLE state. It will then need to set the register address according to their application needs. State OP_NUM refers to number of data bytes to be transferred. After the data bytes are set it goes to the WR_DATA or SET_CTRL_REG depending on the requirement. If data needs to be written then it will go to WR_DATA state else it will go straight to the SET_CTRL_REG state. After that, the slave will wait for the INT signal from the I²C Master. The slave will wait for another cycle to so that the I²C Master can complete its operation before checking the status and reading out the data.

IV. THE VERIFICATION

The testbench and stimulus were applied to the Design Under Test (DUT) which are I²C Master and I²C Slave. These testbenches are written in Verilog Language. The simulation was done using a tool from Mentor Graphic which is called ModelSim. The internal signal includes clk, rst_n, scl, sda_i, sda_o, sda_oe, done signal, rdata bus and start pin to trigger the start of the simulation. This is shown in Fig. 4. Verilog language is used to design I²C slave in an embedded application together with the Field Programming Gate Array (FPGA) board.

A. Result of the Verification

As can be seen from Fig. 5, the 'write' and 'read' operations are performed for the Design under Test (DUT) which consists of I²C Master and Slave. The pi_start signal indicates the start of the operation. The po_rdata signal is read out and the po_done signal will be triggered indicating the 'reading' operation is completed. The other waveforms are for the other pins as described in Table I.

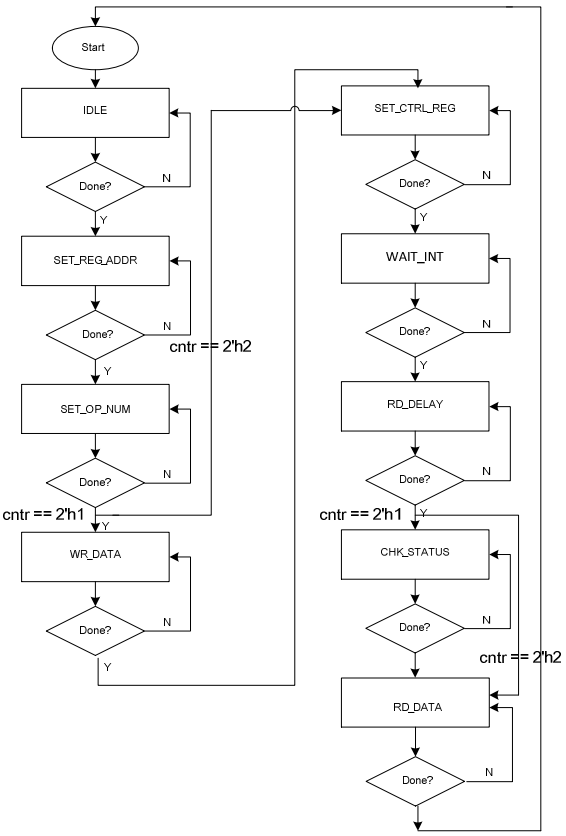

Fig. 3 I²C Slave Flowchart

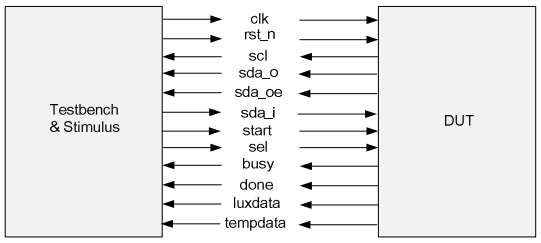

Fig. 4 Block Diagram for Verification

V. EXAMPLES OF APPLICATION

One of the application examples for this I²C Master and slave integration is the use of the Fully Integrated Proximity and Ambient Light sensor chip developed by Vishay Semiconductor (VCNL4000) [2]. This chip together with the I²C master-slave can be used to collect data from the changing of temperature.

Fig. 6 shows the VCNL chip used for I²C application. The dedicated bidirectional input-output 4 for SDA and 5 for SCL are already given as shown.

This VCNL4000 chip can be used in many applications which used I²C for its communication protocol.

The application includes proximity sensor for mobile devices i.e. smartphones, tablets for touch screen locking and power saving [2].

Other application examples are for controlling and dimming the light for keypad and computer display [2].

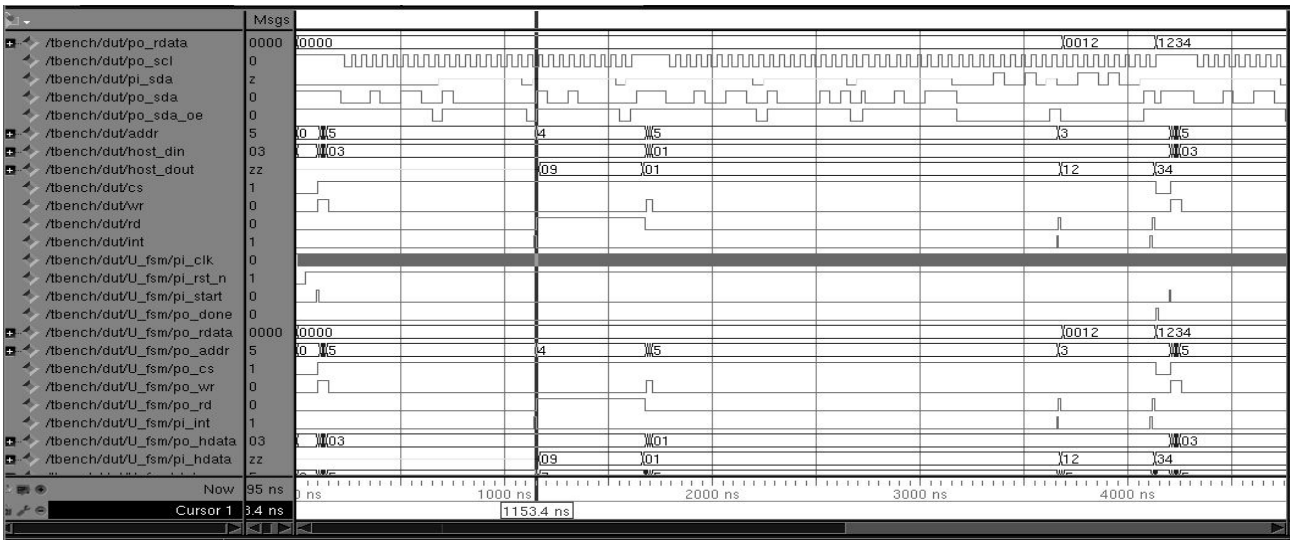

Fig. 5 Signal Waveform of DUT

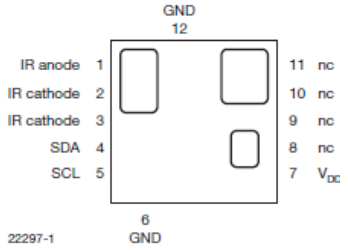

Fig. 6 VCNL4000 Chip

Method used to control this VCNL4000 chip is also described in the datasheet provided by the manufacturer. One important method is to set the slave address according to the datasheet. The predefined 7 bit I²C bus address is set at 13h for the slave address. Writing is set at the bus address 26h and reading is 27h. This address is already fixed by the manufacturer. This is followed strictly to ensure the success of the data transferred in this I²C communication protocol [2].

Fig. 7 shows a diagram for I²C application together with VCNL4000 chip taken from its datasheet. The diagram is an example on how the chip and I²C are integrated together.

VI. Advantages and Limitations

The most popular advantage of I²C is because of its simplicity, I²C is using only 2 signal lines. Data transmission rates are very flexible ranging from 100kbps to 3.4Mbps. For any device attached to the bus, the devices can have their own addresses and be independently addressable. I²C Master-Slave protocol is simple [3]. In this case, we are using START, STOP and ACK signal to control between Master and Slave relationship.

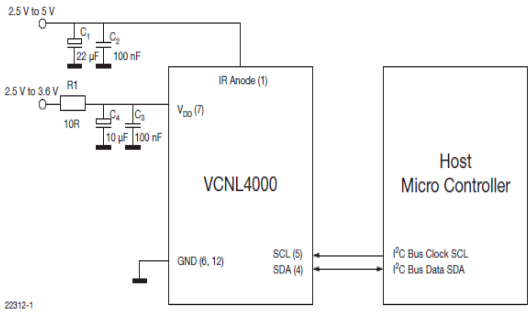

Fig. 7 Application example for I²C [3]

I²C protocol can handle multi master and multi slave design depending on the needs. This can be done by having arbitration capability of the Master to avoid collision on the bus. The design discussed in this paper however, only used single master to operate. Sharing of address is possible for devices on the same bus in order to utilize the only 7 bit address available.

Compared to Serial Parallel Interface (SPI) protocol, I²C has a longer distance for communication. However I²C draws more power than other serial communication busses. This is due to the open drain communication lines topology [4].

Another limitation that needs to be taken into consideration is that the devices' speeds are compromised by the I²C speed. Whether it is faster or slower, the devices need to follow the I²C data transmission rate to avoid glitches.

VII. Conclusion

I²C Implementation is the most common and can be a perfect choice for application which demands simplicity due to its only 2 wires communication protocol. It is also a typical alternative for those who prefer lower cost over high speed.

Most popular applications for I²C communication protocol are for examples, an application to read certain memory ICs, application to read data from hardware sensors of which this paper described. Other applications include accessing DACS and ADCs and application to communicate with multiple microcontrollers. It is also used for security sensitive applications such as sensor connections, RFID and biometric devices [5].

References

[1] Liang, "I2C Master Core Specification," http://www.opencores.org, Rev：1.0 2004-7-28 , 2004
[2] Vishay Semiconductor datasheet, "Fully Integrated Proximity and Ambient Light Sensor with Infrared Sensor and I2C Interface", http://vishay.com, 2011, page 12.
[3] Yifan L, and Fei M, " Principle of I2C bus and its Application in IC Design"2007
[4] B. Matthew, "Overview of I2C", http://components.about.com/od/Theory/a/Overview-Of-I2c.htm, date visited: 25 April 2015.
[5] C. S. Sansar, "Design & Implementation of I2C Master Controller Interfaced With RAM Using VHDL", Journal of Engineering Research and Applications, Vol 4, Issue 7, July 2014, page 67-70