INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

TO DESIGN AN I^{2}C MASTER PROTOCOL IN VHDL FOR DESIRED AND CHANGING BUS CLOCK SPEED E.g. 100 KBPS AND 400 KBPS

Mudit Vaish[1], Twinkle Gupta[2], Rakesh Jain[3]

[1]Research Scholar, Suresh Gyan Vihar University
Jaipur, Rajasthan, India
muditvaish8191@gmail.com
[2] Research Scholar, Suresh Gyan Vihar University
Jaipur, Rajasthan, India
gupta.twinkle18291@gmail.com
[3]Assistant Professor, Suresh Gyan Vihar University
Jaipur, Rajasthan, India
100jaijeenendra@gmail.com

Abstract: *Now a days, every user wants maximum speed for data transfer between two devices. As the manufacturer is increasing the data transfer speed, its architecture is a matter of concern while trying to develop it. In 1982, Philips, a giant manufacturer developed the I^{2}C protocol for 100 Kbps speed. And this speed is increased in later versions. Nowadays, a speed of 3.4 Mbps and 5.0 Mbps is in vogue, at which different manufacturers are developing their devices. Here, in this work, I have found out how to work on two several speed, with a single code of VHDL module. By using this, we will be able to communicate at two speeds i.e. 100 Kbps and 400 Kbps via a single set of code. One code for two different speeds makes the architecture simpler to use and understand it. And in addition, its designing will be less complex and so more compact in size, which is what a big concern of manufacturers.*

Keywords: *I^{2}C Master Protocol, IIC, SDA and SCL Lines, I2C Advantages, I2C protocol code in VHDL.*

1. INTRODUCTION

The I2c bus also known as Inter- Integrated circuit and referred as I-squared-C, I-two-C or IIC is a multi-master serial single-ended computer bus designed by Phillips to connecting low speed peripherals to a mother-board, embedded system, cell phones or other electronic device. It is a 2-wire, half duplex data link designed for data transfer between a computer and other peripherals [3]. Its block diagram is shown in Fig. 1.

It was designed in 1982, and then it was able to only a maximum of 100 k-bit per second. Then, a Fast mode came into existence and it was able to transfer at the rate of 400 k-bit per second. And since, 1998, a 3.4M-bit per second option is available. And nowadays, 5000 k-bit per second is in vogue. Here, we are discussing and understanding the Fast mode that is being capable of a transfer rate of a maximum 400 K-bit per second. The program is designed in VHDL for both 100 Kbps and 400 Kbps and we can understand its working through its waveforms.

The two wires of I2C bus are SDA and SCL. SDA is Serial Data Line and SCL is Serial Clock Line. These lines are bi-directional and open drain and pulled up by resistors. The devices on the bus pulls a wire to GND to transmit a logical "0" (zero) and release a line (floating condition) to send a logical one [6].

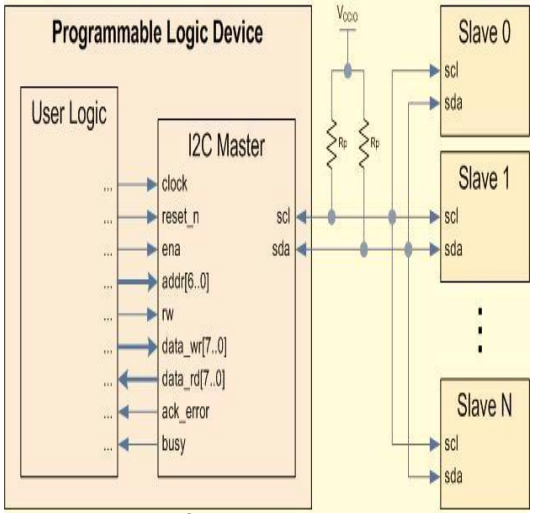


Figure 1: I^{2}C Master Block Diagram

The I2C-bus is also used in various control architectures such as System Management Bus (SM Bus), Power Management Bus (PM Bus), Intelligent Platform Management Interface (IPMI), Display Data Channel (DDC) and Advanced Telecom Computing Architecture (ATCA). The I2C bus is also available in various operating modes.

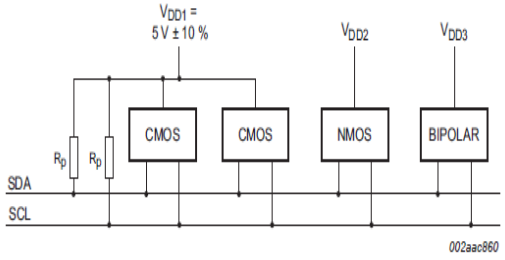
INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

2. LITRATURE REVIEW

(I2C bus protocols): Serial buses are used for data transfer between a device like computer and its peripherals. Different users use different protocol as for their purpose. Out of which I^2C bus protocol is the most efficient one. As it requires less complex architecture, at low cost, good transfer speed between the devices. And most important one is no other address or device in system can create a distraction or halt in first or running data transfer. As it is pulled up by master device only i.e. data is safe. The other big advantage here is that one can attach a lot of devices with one system. And it is Multi-master protocol, means, it is not necessary that only one master will act for the data transfer. And the receiving device, at particular time, is referred to a Slave device. Master can also act as slave device. And it is also compatible with other buses. So I^2C bus protocol is the protocol that is, for sure, most commonly used. The reason behind this is also there that its architecture is very simple and easy to understand. So other users found it simple and more effective to use this protocol in their system. And this led to so much development in I^2C protocol since its first version. And using same code for various speeds, it will make less complex architecture to develop.

a. SDA AND SCL SIGNALS:

Both SDA and SCL are bidirectional lines, connected to a positive supply voltage via a current-source or pull-up resistor (see Figure 2). When the bus is free, both lines are HIGH. The output stages of devices connected to the bus must have an open-drain or open-collector to perform the wired-AND function [1].



V_{DD2}, V_{DD3} are device-dependent (for example, 12 V).

Figure 2: SDA and SCL Lines

3. THEORY OF OPERATION

Fig. 2 shows the state machine of I^2C protocol. Upon start-up, the device enters the ready state. After ENA signal latches, the START signal generates the start condition. And then, through command state, the device communicates the address and RW commands to the bus. Then SLV_ACK1 checks the slave's acknowledge - ment. And after that, as per the RW command, device will come to know whether to receive (RD state) or send the data (WR state) [4]. After this, master look for slave's response (SLV_ACK2 state) in case of writing or sending the data and its own response (MSTR_ACK state) in case of reading or receiving the data from the slave. After sending or receiving, if ENA signal latches again, the master device immediately continue with another read or write operation in case of the same previous command. If commands are different than the previous one, then master issues a repeated start as per the I^2C specification. And if, after completion of a read or write, ENA does not latch again a new command, the master generates a STOP condition (STOP state) and returns at READY state as shown in figure 3.

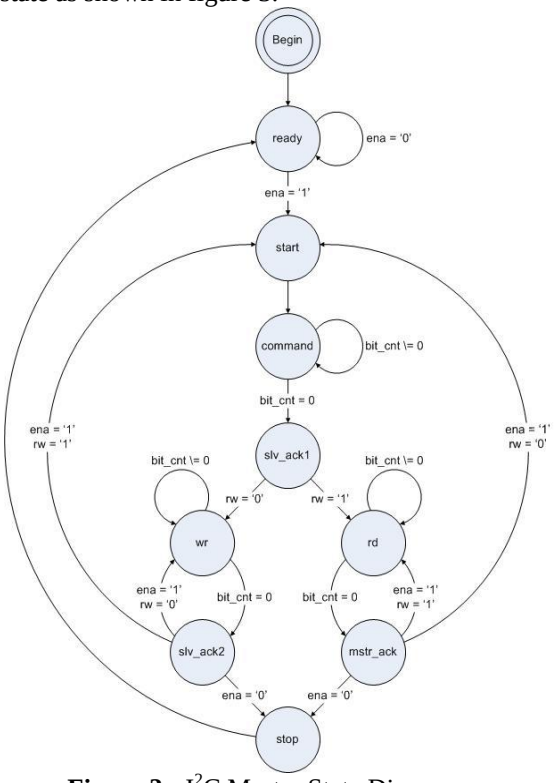


Figure 3: I^2C Master State Diagram

OTHER USES OF THE I2C BUS COMMUNICATION PROTOCOL: The I2C-bus protocol is also being used for other architectures as communication protocols. Different users are using this configuration for their own type of work but the protocol and physical interfaces are the same. The different types of buses and their compatibilities with I2C,

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

in which I2C-bus protocol is being used [2], are as follows-

a. **CBUS COMPATIBILITY:**

Standard mode I2C-bus can be connected with CBUS receivers, however, a third bus line called DLEN must be connected and the acknowledge bit omitted. And then, after transmission of the CBUS address, the communication is been done by CBUS, DLEN lines and some STOP conditions. And this STOP condition is been recognized by all devices.

b. **SYSTEM MANAGEMENT BUS:**

It is generally used in PC motherboards for power supply monitor, temperature monitor etc. SM Bus uses I2C hardware and addressing, but it needs second level software which includes an Address Resolution Protocol that can make dynamic address allocations. This dynamic reconfiguration of the system allow bus device to be 'hot-plugged' and used immediately, without restarting the system. This advantage results in a PLUG-and-PLAY user interface.

c. **POWER MANAGEMENT BUS:**

The Power Management Bus uses the I2C-bus protocol through SM Bus Version 1.1. It is a standard way to communicate between power converters and a system host over the SM Bus to provide more intelligent control of the power converters.

d. **INTELLIGENT PLATFORM MANAGEMENT INTERFACE:**

IPMI defines a standardized, abstracted, message-based interface for IPM hardware. IPMI increases reliability of systems by monitoring parameters such as temperatures, voltages, fans and chassis intrusion. It provides general system management functions such as automatic alerting, automatic system shutdown and restart, remote restart and power control.

e. **ADVANCED TELECOM COMPUTING ARCHITECTURE:**

ATCA is a follow-on to compact PCI (C-PCI), providing a larger card area, larger pitch and larger power supply for use in advanced rack-mounted telecom hardware. It includes a fault-tolerant scheme for thermal management that uses I2C-bus communication between boards. In ATCA, this requires some extra hardware and software to manage the dual I2C Buses. In Addition, on-board temperature reporting, tray capability reporting, fan turn-off capabilities, and non-volatile storage are required.

f. **DDC - DISPLAY DATA CHANNEL:**

The DDC allows a monitor or display to inform the host about its identity and capabilities. It allows bi-directional communication between the display and the host. It enables the controls of monitor functions such as image display and communication with other devices connected to the I2C-Bus.

4. DESINGING SCHEME

To design this protocol for two various speeds 100 Kbps and 400 Kbps, I have used the same code in VHDL. To design it at manufacturer level, only some small changes will be there in its architecture. And we can design it by using PLANAHEAD too.

In our program, we gave input to bus clock at 4,00,000 Hz, means 400 Kbps maximum transfer speed. We can also provide 100 KHz or even more than 400 KHz. This code is efficient at any speed. Don't forget that we'll also have to take a look as per its designing. So here, we can take its example with 400 Kbps. We have provided input clock of 50 MHz for creation of a divider = (input clock / bus clock)/4; this divider is number of cycles in quarter cycle of SCL.

The busy and other signals are also being used as shown in fig for showing that at this time, a particular data transfer is being occurring. So when, busy will be ON, no any other request will be taken.

Its further working can be understood by its theory of operation and its state diagram in previous section. The working of transfer of data from master to any selected slave device is same as its state diagram. These two acknowledgements are nothing but the acknowledgements between master and slave, after which only, a master device will transfer the data to a slave device.

5. CONCLUSION

We can see data transfer at this speed in the fig. 4. The data (DATA_WR) at the address (ADDR) is transmitted at the same address and received by the particular address. We can see this transmitted data in the figure as (DATA_TX). It can also be observed that state is being changed from READY to START. And other states cannot be shown as per its diagram. But after giving input as per its requirements in the system, this code is showing every state like ACKNOWLEDGEMENT and RD or WR states.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

As we have also discussed in previous section, we can also increase or decrease its speed just by changing the bus clock speed. But this thing is to be noted here that it will also decide clock cycles in quarter of a SCL. And according to that, the states will change after that particular period interval.

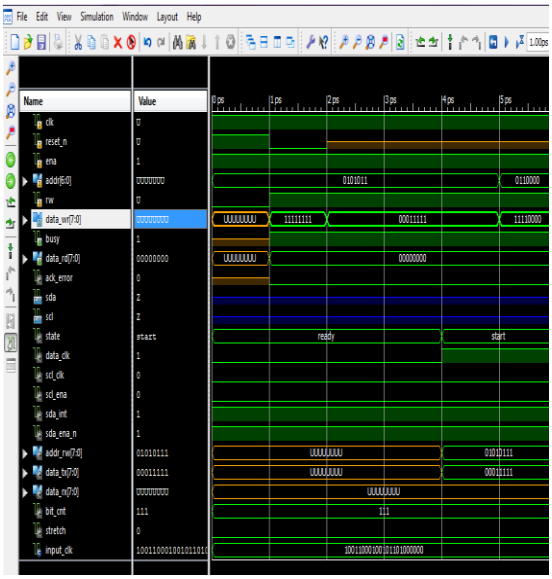


Figure 4: Final Results (Write Operation)

6. ADVANTAGES

Some basic advantages of I2C bus are as follows-

- This protocol required only two bus lines; a Serial Data Line (SDA) and a Serial Clock Line (SCL).
- Every peripheral connected to the bus is having a unique address and simple master/slave relationships exist at all times, in this way, masters can operate as master-transmitters or as master receivers
- Serial, 8-bit oriented, bidirectional data transfers can be made at up to 100 Kbps in the Standard-mode, up to 400 K-bit/s in the Fast-mode, up to 1 Mbit/s in Fast-mode Plus, or up to 3.4 Mbit/s in the High-speed mode.
- This multi-master bus includes collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate the data transfer.

At other levels [5], this protocol is having following advantages-

a. **AT DESIGNER LEVEL:**

The I2C compatible ICs allow a system design to progress rapidly directly from a functional block diagram to a prototype. This also allows a prototype system to be modified or upgraded simply by 'Clipping' or 'Unclipping' ICs to or from the bus.

I2C protocol allows connections of a wide variety of peripherals without the need for separate addressing or chip enable signals.

b. **AT MANUFACTURER LEVEL:**

Simple 2-wire Serial I2C-bus reduces interconnections, In this way, ICs have fewer pins and less PCB tracks. So it makes smaller and less expensive PCBs. The completely integrated I2C bus protocol doesn't need the address decoders and other 'glue logic'. It allows rapid testing and alignment of end-user equipment via external connections to an assembly line.

7. FUTURE WORK

For future considerations, we can see that this code is helpful in getting desired bus clock speed. But it is limited, as the hardware part is limited to design. As we will increase the speed of the bus, it will be harder in designing it.

Due to its limited speed, it can't be used in devices that need higher speed like using it in pen drives and other peripheral devices.

One more limitation is there that it can be used for a small distance as per its architecture and its speed. So that can also be overcome by designing for using for more distance.

REFERENCES

[1] Vincent Himpe, Mastering the I^{2}C Bus, 248 pages, 2011, ISBN 978-0-905705-98-9.

[2] Dominique Paret, The I2C Bus: From Theory to Practice, 314 Pages, 1997, ISBN 978-0-471-96268-7.

[3] UM 10204 I^{2}C Bus Specification and User Manual Rev. 4 – 13 February 2012, By NXP Semiconductors.

[4] I^{2}C Master (VHDL), By Scott Larson, 01 August 2013, http://eewiki.net/

[5] TMS 320X, 280X, 2801X, 2804X Inter-Integrated Circuit (I^{2}C) Module, By TEXAS INSTRUMENTS, Literature Number SPRU721C, November 2004 – Revised June 2011.

[6] AN10216-1, I^{2}C Manual, By Jean Marc Irazabal and Steve Blozis, Phillips Semiconductors, March 24, 2003.