IOSR Journal of Engineering (IOSRJEN) www.iosrjen.org
ISSN (e): 2250-3021, ISSN (p): 2278-8719 Special Issue || September 2021 || PP 16-21
7th National Conference On Advancements in Communication, Computing and Electronics Technology-ACCET 2021

Interfacng of Sensors with FPGA Board Using I2c Protocol

SangramSawant
Department of Electronics and Telecommunication,
Modern education society's college of Engineering, Pune
Bund Garden Rd, Pune, Maharashtra 411001, India

ShubhamMarathe
Department of Electronics and Telecommunication,
Modern education society's college of Engineering, Pune
Bund Garden Rd, Pune, Maharashtra 411001, India

SwapnilBagle
Department of Electronics and Telecommunication,
Modern education society's college of Engineering, Pune,
Bund Garden Rd, Pune, Maharashtra 411001, India

Prof.KanchanTiwari
Department of Electronics and Telecommunication,
Modern education society's college of Engineering, Pune
Bund Garden Rd, Pune, Maharashtra 411001, India

Abstract— *This paper implements serial data communication using I2C (Inter-Integrated Circuit) master bus controller using a field programmable gate array (FPGA).FPGA is an integrated circuit designed to be configured by user or designer after manufacturing.FPGAs are more often used in applications where huge amount of data must be processed and routed in real time. The I2C master bus controller is interfaced with TCN75A, HDC1080, TCS3472. These three sensors act as slaves. This module is designed in VHDL and simulated in Xilinx Vivado 2020.2. The design is synthesized using Xilinx Vivado 2020.2. I2C master initiates data transmission and in order slave responds to it.In this project, we are working on system which requires minimum ports so that the system becomes compact and reduce latency. This project has vast application in home automation and industrial automation.*
Keywords— *TCN75A, HDC1080, TCS3472, I2C Master, FPGA*

I. INTRODUCTION

In this era where everything is getting compact and faster, we should look forward towards the systems which could perform multiple applications with the same basic hardware just by changing few lines of code. So this need led us to work with the FPGA and I2C based system. After this first and most likely question arises what FPGA is? If you've talked on a cell phone or browser the internet, then you've more likely benefited from an FPGA. While not commonly known outside of technology circles used in applications where huge amount of data must be processed and routed in real time.

In the world of serial data communication, there are protocols like RS-232, RS-422, RS-485, SPI (Serial peripheral interface), Micro wire for interfacing high speed and low speed peripherals. These protocols require more pin connection in the IC(Integrated Circuit) for serial data communication to take place, as the physical size of IC have decreased over the years, we require less amount of pin connection for serial data transfer. USB/SPI/Micro wire and mostly UARTS are all just 'one point to one point' data transfer bus systems. They use multiplexing of the data path and forwarding of messages to service multiple devices. To overcome this problem, the I2C protocol was introduced by Phillips which requires only two lines for communication with two or more chips and can control a network of device chips with just a two general purpose I/O pins whereas, other bus protocols require more pins and signals to connect devices. In this project, we are implementing I2C bus protocol for interfacing low speed peripheral devices on FPGA. It is also the best bus for the control applications, where devices may have to be added or removed from the system.[4]

I2C protocol can also be used for communication between multiple circuit boards in equipment's with or without using a shielded cable depending on the distance and speed of data transfer. I2C bus is a medium for communication where master controller is used to send and receive data to and from the slave. The low speed peripheral, is interfaced with I2C master bus and synthesized on Basys3. The I2C bus system with the I2C master controller implemented o n a FPGA and the real time clock device acting as the slave. The synopsis of the paper is as follows: We discussed I2C protocol of our proposed design which also presents module description for our proposed system. We present the software implementation along with algorithm and flow chart. It holds the detailed description of hardware implementation of I2Cmaster bus controller in Basys3 FPGA Design kit using Vivado software. Finally, concluded with future scale up.

II. METHODOLOGY

A. *I2C Interface*

The FPGA used in this prototype study is ARTIX 7 embedded in BASYS 3 FPGA board. The I2C communication is established between the FPGA and the TCN75A, HDC1080, TCS3472 sensors.

It would be easier to interface the digital sensors with FPGA through I2C or SPI communication protocols. But here we recommend I2C since it provides some advantages over SPI Communication. In the I2C protocol, using two wires – SDA (Serial Data Line) and SCL (Serial Clock Line) we can control many slave devices. In case of SPI communication, connecting a single master to a single slave with an SPI bus requires four lines. Each additional slave requires one additional chip select pin on the master. Therefore large number of connections for each device could be challenge in a tight PCB layout [3]. I2C is a multi-master, multi-slave, single-ended, simple bidirectional two-wire serial computer bus invented by Philips Semiconductors (NXP Semiconductors) for efficient inter IC control. All I2C-bus compatible devices incorporate an on-chip interface that will allow them to communicate directly with each other via the I2C-bus. This design concept solves many interfacing problems encountered while designing digital control circuits.[3]

Both SDA and SCL lines of I2C bus are bidirectional lines, connected to a positive supply voltage via pull-up resistors. 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 (Figure 1). There are different data transfer modes available based on the transfer rates as Standard mode (up to 100 Kbit/s), Fast mode (up to 400 Kbit/s), Fast mode plus (up to 1 Mbit/s) and High.[3]

B. *Data Transfer*

Data transfers are initiated by a Start condition (Start), followed by a 7-bit device address and a read/write bit. An Acknowledge (ACK) from the slave confirms the reception of each byte. Each access must be terminated by a Stop condition (Stop).Repeated communication is initiated after tB-FREE.This device does not support sequential register read/ write. Each register needs to be addressed using the Register Pointer.This device supports the Receive Protocol. The register can be specified using the pointer for the initial read. Each repeated read or receive begins with a Start condition and address byte. The sensor retains the previously selected register. Therefore, it outputs data from the previously specified register.[2]

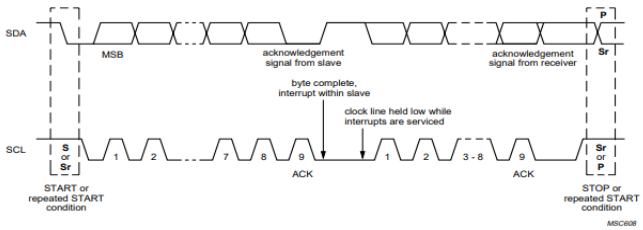

Figure 1: Data transfer on the I2C-bus

C. *Master/Slave*

The bus is controlled by a master device (typically a microcontroller) that controls the bus access and generates the Start and Stop conditions. The sensor is a slave device and does not control other devices in the bus. Both master and slave devices can operate as either transmitter or receiver. However, the master device determines which mode is activated.[2]

D. *Start/Stop Condition*

A high-to-low transition of the SDA line (while SCL is high) is the Start condition. All data transfers must be preceded by a Start condition from the master. If a Start condition is generated during data transfer, the sensor resets and accepts the new Start condition.

A low-to-high transition of the SDA line (while SCL is high) signifies a Stop condition. If a Stop condition is introduced during data transmission, the sensor releases the bus. All data transfers are ended by a Stop condition from the master.[2]

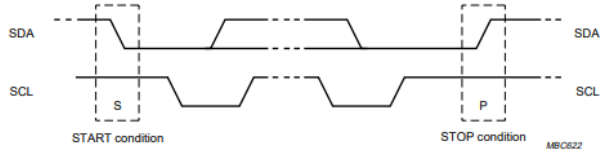

Figure2: START and STOP conditions.

E. *Address Byte*

Following the Start condition, the host must transmit an 8-bit address byte to the sensor. For Instance, the address for the sensor is '1001,A2,A1,A0' in binary, where the A2, A1 and A0 bits are set externally by connecting the corresponding pins to VDD '1' or GND '0'. The 7-bit address transmitted in the serial bit stream must match the selected address for the sensor to respond with an ACK. Bit 8 in the address byte is a read/write bit. Setting this bit to '1' commands a read operation, while '0' commands a write operation.[2]

F. *Acknowledge (ACK)*

Acknowledgement is obligatory in order to inform the transmitter that data has been successfully transmitted. Figure1 illustrates the acknowledgement mechanism. The Master generates the acknowledge-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.

G. *Data Validity*

After the Start condition, each bit of data in transmission needs to be settled for a time specified by data before SCL toggles from low-to-high.

III. SENSORS

H. *TCN75A Temperature Sensor*

- The TCN75A product comes with user programmable resistors that provide flexibility for temperature sensing applications.
- Continually outputs latest temperature data on a parallel interface
- Handles I2C communication and all data retrieval from the Temperature Sensor Pmod.
- Range: -40 to 125 ^{0} C.*[2]*

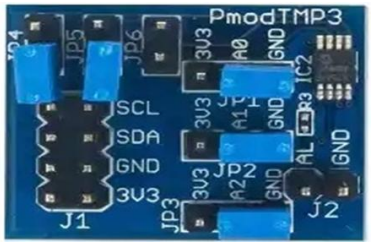

Figure 3: TCN75A Temperature Sensor

I. *HDC1080 Humidity Sensor*

- The HDC1080 is a digital humidity sensor with 1• Relative Humidity Accuracy ±2% (typical)
- Temperature Accuracy ±0.2°C (typical) integrated temperature sensor that provides excellent measurement accuracy at very low power.
- The Excellent Stability at High Humidity HDC1080 operates over a wide supply range, and is 14 Bit Measurement Resolution a low cost, low power alternative to competitive.[5]

Figure 4: HDC1080 Humidity Sensor

J. *TCS3472 Colour Sensor*

- The TCS3472 device provides a digital return of red, green, blue (RGB), and clear light sensing values.
- An IR blocking filter, integrated on-chip and localized to the colour sensing photodiodes, minimizes the IR spectral component of the incoming light and allows colour measurements to be made accurately.
- The high sensitivity, wide dynamic range, and IR blocking filter make the TCS3472 an ideal colour sensor solution for use under varying lighting conditions and through attenuating materials.

Figure 5: TCS3472 Colour Sensor

IV. STATE MACHINE

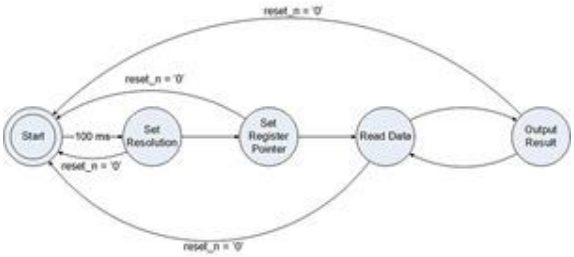

Figure 6: State Diagram

The design uses the state machine depicted in Figure 6 to implement its operation. Upon start-up the component immediately enters the start state. It remains in this state for 100ms to ensure the sensor module has ample time to power-up. It then proceeds to the set_resolution state, where it configures the sensor's resolution. Once complete, it enters the set_reg_pointer state to set up thesensor for data readings. In the following read_data state, it gathers the most recent data from the sensor. Finally, it outputs the data in the output_result state. It then continuously cycles between the read_data and output_result states to keep the data constantly updated. Resetting the component at any time returns it to the start state.

The above process is implemented to each sensor interfaced with the I2C master on FPGA board and the serial communication is carried out between Master and slave.

A.I2C Master

During the set_resolution, set_reg_pointer, and read_data states, the state machine controls an I2C Master component to communicate with the sensors.

B.System clock frequency

The generic parameter sys_clk_freq must be set to the frequency of the system clock provided to the sensor control on the clock port.

V. SOFTWARE IMPLEMENTATION

I2C master controller is designed VHDL based on Finite State Machine (FSM). FSM is a sequential circuit that uses a finite number of states to keep track of its history of operations, and based on history of operation and current input, determines the next state. There are several states in obtaining the result.

a. Algorithm

State1: An idle condition: I2C bus doesn't perform any operation. (SCL and SDA remainshigh).

State 2: Start condition: master initiates datatransmission by providing START (SCL is high and SDA is from high to low).

State 3: Slave address - write: master sends the slave address-write to each slave.

State 4: If the slave address matches with the one of the three slaves, the acknowledged slave sends an acknowledgement bit in response to the master.

State 5: Register Address will betransmitted to the slave. Again, acknowledgement is sent to the master by the slave.

State 6: Data to be transmitted is sent to the master by the slave. After receiving the data, master acknowledges the slave.

State 7: The data transfer is terminated for the first acknowledged slave and then the address for next slave is transmitted.

State 8: If the slave address matches with one of the remaining two slaves, the acknowledged slave sends an acknowledgement bit in response to the master.

State 9: Register Address will be transmitted to the slave. Again, acknowledgement is sent to the master by the slave.

State 10: Data to be transmitted is sent to the master by the slave. After receiving the data, master acknowledges the slave.
.
State 11: And the same process iterates for the third slave.

State 12: Then the data transfer of third slave isstoppedandThen it goes to state 3 and the process continues for all slaves.

VI. RESULTS AND DISCUSSIONS

The simulation for interfacing I2C master and three sensor slaves is demonstrated in Vivado 2020.2 software and synthesized on Basys 3 FPGA board.

The Figure shows the expected result for read operation of I2C master in which the output data of each sensor is read by the I2C master in FPGA.

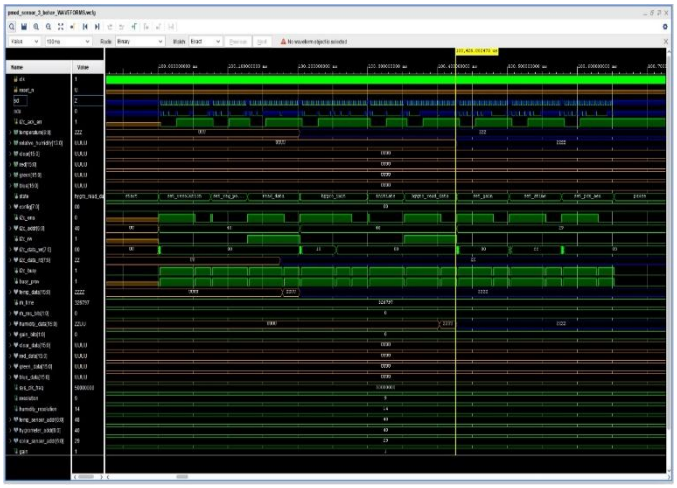

Figure 7: Simulation Result for all three slaves.

VII. CONCLUSION

In this project, the sensors are programmable logic components that are interfaced to the FPGA using I2C protocol. It handles all the serial communication between FPGA and sensors. Each sensor is called by the Master individually by transmitting the unique address of the slave. Due to this approach, we are successful in achieving minimum latency and serial communication between slave and master.

ACKNOWLEDGMENT

First and foremost, we would like to thank to our supervisor of this project, DR. KANCHAN TIWARI for the valuable guidance and advice. She inspired us greatly to work in this project. Her willingness to motivate us contributed tremendously to our project. We also would like to thank her for showing us some example that related to the topic of our project. Besides, we would like to thank the authority of MODERN EDUCATION SOCIETY'S COLLEGE OF ENGINEERING for providing us with a good environment and facilities to complete this project. Also, we would like to take this opportunity to thank to the MODERN EDUCATION SOCIETY'S COLLEGE OF ENGINEERING for offering this subject, Computing Project. It gave us an opportunity to participate and learn about the operation of Sensors..

REFERENCES

[1]. Philips Semiconductor "I2C Bus Specification" version 2. 1, January 2000
[2]. TCN75A 2-Wire Serial Temperature Sensor Datasheet, Microchip Technology Inc., 2010
[3]. A. Lokapure, S. Mandar N., B. Satyanarayana and P. C. M. Raees, "Interfacing of digital TPH sensors with FPGA using I2C interface," *2016 IEEE Bombay Section Symposium (IBSS)*, 2016, pp. 1-5, doi: 10.1109/IBSS.2016.7940205..
[4]. Sangepu Baby Dhivya "Implementation of I2C Single Master Multiple Slave Bus Controller on FPGA" IPASJ International Journal of Electronics & Communication (IIJEC), Volume 3, Issue 1, January 2015
[5]. HDC1080 Digital Humidity Sensor Datasheet, Texas Instruments, 2016
[6]. Shital Mahajan "Study of I2C protocol for implementation on FPGA", International Journal of Computer Application(0975-8887), NCETACT,2015
[7]. Digilent "Basys 3 FPGA Board Referance Manual" Revised 2016
[8]. SpandanaSunku and LanvyaLathaMutyala,"Design of I2C bus controller using VHDL", International Journal of Emerging Engineering research and Technology, Vol.3, issue8, 2015, PP125-129
[9]. Prof. Jai karanSingh,"Design and Implementation of I2C master controller on FPGA using VHDL", IJET, Vol.4 no.4 August-September2012