Library Progress International
Vol.44 No. 3, Jul-Dec 2024: P. 17384-17390

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

Model Of Fpga Enabled Body Monitoring System Using I2c Protocol

[1]Ravi Payal*, [2] Prof. Amit Prakash Singh

[1] Research Scholar ,University School of Information, Communication & Technology, Guru Gobind Singh Indraprastha University, New Delhi, India
[2] University School of Information, Communication & Technology, Guru Gobind Singh Indraprastha University, New Delhi, India

How to cite this article: Ravi Payal, Amit Prakash Singh (2024) Model Of Fpga Enabled Body Monitoring System Using I2c Protocol. *Library Progress International,* 44(3) 17384-17390.

Abstract
The paper talks about the Simulation model of I2C controller by using this a patient can measure a Heart pulse rate and do clinical check-ins if any problem detected. A sensor system was designed for patient self-monitoring with outputs displayed on LCDs. The proposed i2C interface model was prototyped in Sparton3 FPGA board. The proposed model comprised a I2C slave and I2C Master. The "master" collects results from the "slaves" and transmits the data to a personal computer for display.

Keyword : FPGA, I2C, Verilog, Sensor,Interfacing, Synthesis

Introduction
A digital sensor-based system is designed for patient self-monitoring and professional consultation, emphasizing pulse rate measurement. The system involves synchronized serial communication between the pulse sensor and an FPGA, displaying the pulse rate on the FPGA board's seven-segment display. An alarm triggers when the pulse rate exceeds a critical value, and inter-FPGA communication relays data from the patient to the clinical end via an I2C bus, with the patient-end FPGA operating in MASTER mode and the clinical-end FPGA in SLAVE mode. The output is displayed on a PC through a USB-UART interface, with host computer software establishing a serial port connection to display and store the received data at the clinical end.(Figure 1).

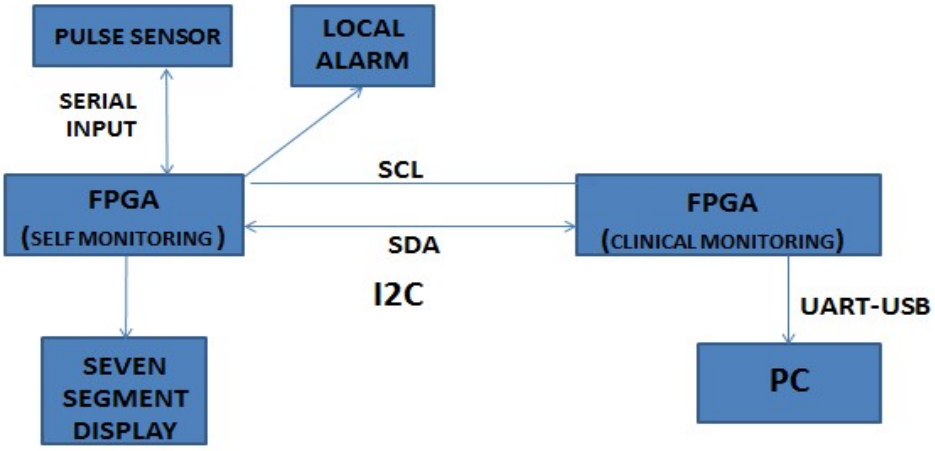

Figure 1. Overview of the Full system

1.1. THE I2C PROTOCOL

Philips Semiconductors' I2C bus, developed over 20 years ago, facilitates IC communication with minimal pins,

enabling direct communication through an on-chip interface [2]. Initially meant for single-card connections, the I2C bus had a maximum capacitance of 400 pF and a top speed of 100 kbps. The I2C protocol ensures straightforward, lossless communication, higher speeds, and supports various peripheral devices with minimal hardware. Widely adopted by major chip designers, the I2C interface allows data transfer among ICs on a PCB, solving many digital design interfacing issues. It uses only two bi-directional lines: SDA and SCL, with each device having a unique 7 or 10-bit address. The initiating device is the Master, and others are Slaves. Illustrated in Figure 2, the I2C bus is a simple, cost-effective two-wire serial bus requiring no chip select or arbitration logic.

The USB-UART PROTOCOL

The UART converts parallel bytes into a serial bit-stream, transmitting bits sequentially through a single wire per direction, and reconverts incoming bit streams to parallel bytes. Embedded systems commonly use RS-232 for communication, especially during debugging. As USB replaces RS-232, devices emulate RS-232 over USB, simplifying debugging without affecting data transmission. The USBUART device mimics a COM port via USB, requiring driver installation on Windows. The Nexys4 board's FTDI FT2232HQ USB-UART bridge enables communication using standard Windows COM port commands, converting USB packets into UART/serial data, exchanged with the FPGA through a two-wire serial port (TXD/RXD) and optional hardware flow control (RTS/CTS). After driver installation, PC I/O commands to the COM port generate serial data traffic on the C4 and D4 FPGA pins.

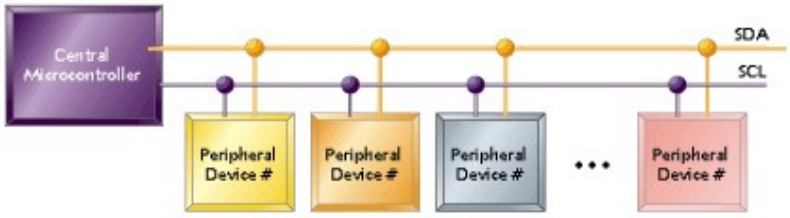

Figure 2. I2C-bus configuration [11]

The figure illustrates a Master-Slave configuration. The Master initiates communication by sending a "START" condition, causing all devices to listen. It then transmits the desired Slave's "ADDRESS" and a data transfer direction bit. The matching Slave acknowledges (ACK), while others remain silent. During communication, only the Master and selected Slave control the busy lines. The Master issues a "STOP" signal to end communication, releasing the SCL and SDA lines. The "START," "ADDRESS," "ACKNOWLEDGEMENT," and "STOP" signals, crucial to the bus's operation, will be further explained. Understanding the bus's physical connection is essential before proceeding.

1.1. START AND STOP CONDITIONS

A Master initiates data transfer by sending a Start condition and concludes with a Stop condition. Multiple Starts within a single transaction are termed repeated Starts, and the Master determines when to issue the Stop condition. Figure 3 illustrates a Start occurring when SDA is low while SCL transitions to high. The Master generates the clock to control timing. In a Stop condition (Figure 4), SDA transitions high when SCL is high.

Initiation of data

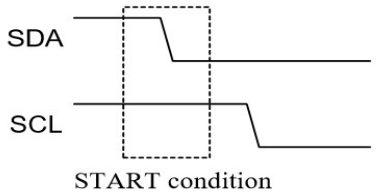

Figure 3. Start Condition

Termination of Data

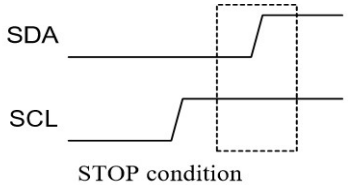

Figure 4. Stop Condition

The I2C bus uses 7-bit and 10-bit addressing modes in a byte-oriented protocol. In 7-bit mode, the Master sends a Slave address and data direction bit (0:WRITE, 1:READ) between START (S) and STOP (P) or repeated START (Sr) conditions. The 10-bit mode, addressing 7-bit limitations, uses two bytes for address and direction, with 11110xx formatting the first byte. Acknowledgement ensures data transmission, with the Master generating

the clock and the receiver controlling the SDA line. Data transfer involves SCL, Start/Stop signals, and the Master's first byte, with Slave and receiver acknowledgements (Figure 2.7). The Master sends "starting bytes" to the Slave, including address and data direction, transmitting MSB first and LSB last.

USB INTERFACE SPECIFICATION

A universal asynchronous receiver and transmitter (UART) is an adjustable circuit that converts parallel data to serial form for transmission, and vice versa at the receiving end, using shift registers (Axelson, 2007). Serial transmission via a single wire offers cost advantages over parallel transmission. Separate interface devices typically manage external signaling levels, allowing simplex, full duplex, or half duplex communication.

2. Results and Implementation

The prototype, using a top-bottom approach, has two modules: the Patient end (Master) and the Clinical end (Slave). The master module includes sensor interfacing, alarm, 7-segment, and I2C master transmission modules while the slave module contains I2C slave reception, 7-segment, and USB-UART modules. The HDL used for writing the RTL is Verilog and simulation was done on Questa Sim 10.1 tool . The Xilinx ISE 14.5 version was for the FPGA Synthesis.

2.1 MASTER IMPLEMENTATION ON FPGA (PATIENT END)

The prototype employs a Pulse Rate Sensor, interfaced with a Pulse Counter module to tally the total pulses over an 8-second sample. Additionally, it includes a module to produce a Control Signal for clinical operations and an Alarm Signal to alert when a critical threshold is surpassed. Detailed descriptions of these modules are provided in the subsequent sections.

1. PULSE COUNTER :This module employs a counter to tally pulses at a 50MHz rate while another segment of code concurrently counts pulses from the sensor. When the clock count hits "400000000" (indicating 8 seconds), the pulse-counting code stops. The pulse count is then extrapolated to a minute using the unitary method and stored in a register. Both the clock and pulse counters reset to zero, and the process repeats.

2. CONTROL SIGNAL GENERATION: Every 8 seconds, both counters reset to zero. From the second iteration onward, when the clock counter reaches "5000000," indicating 0.1 seconds have elapsed, a control signal prompts the clinical end to begin its computation.

3. BUZZER/ ALARM SIGNAL: As the calculated pulse rate for a minute exceeds a critical value(80; as used in the thesis), it enables a buzzer control signal which turns on the buzzer module. In the buzzer module a clock divider is used to generate a frequency of 2.5Hz. And during HIGH of the generated clock, buzzer is turned on and vice versa.

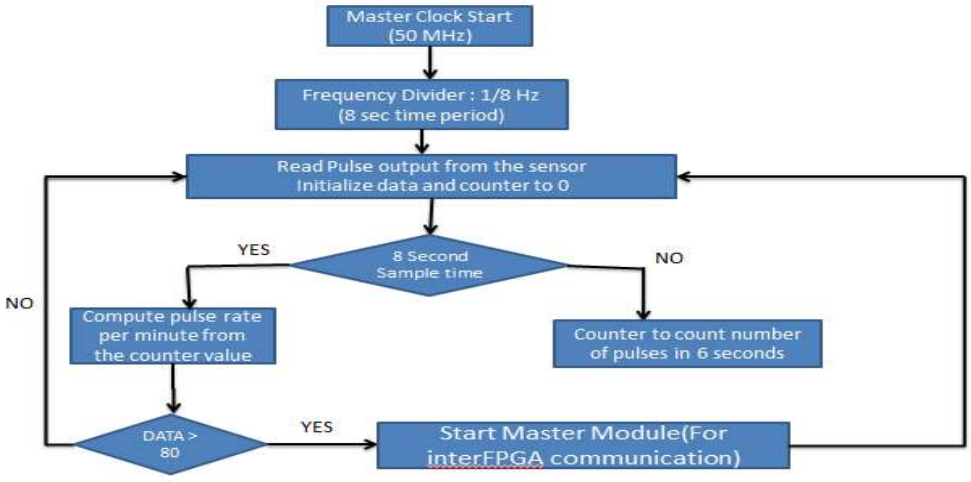

Figure 5. Flow Diagram for Sensor Interface.

2.1. I2C MASTER

The Start or Stop Detector module identifies the Start signal, prompting the master to sample the SDA line and retrieve the address and read/write signal through the Address Block. Upon receiving the slave address, the slave acknowledges the master. The ACK Block detects acknowledgments from the Master and generates responses.

The Data Block then loads data onto the SDA line. Simulation verifies the conditions of start/stop, SCL generation, address block checking, and acknowledgment checking. In Figure 6, Figure 7 and Figure 8 simaution result waveform for I2C master is there wrt to various different conditions.

Figure6. Simulation waveform for I2c Master- when reset and write mode is high

Figure7. Simulation waveform for I2c Master- when both write and reset is Low

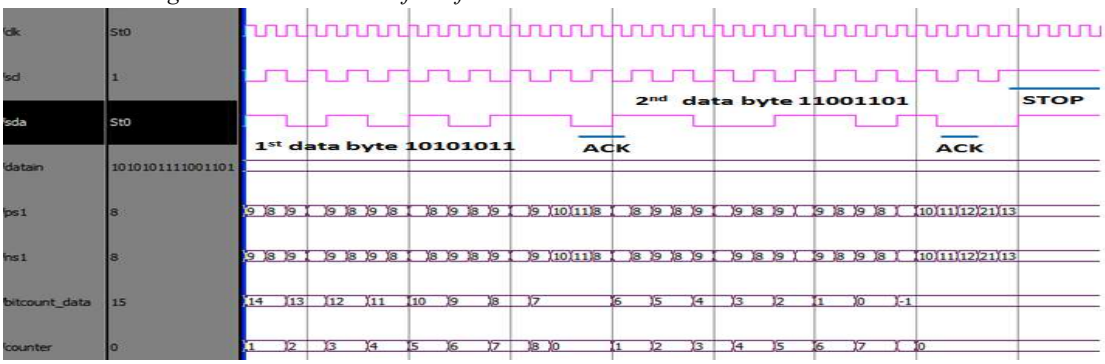

Figure8. Waveform for I2C Master after address has been acknowledged.

table 1 represent the synthesis result of I2C master controller which explains the Device utilization summary of I2C .

S.No	Logic utilization	used	Available	Utilization
1	Number of Slice FlipFlops	164	1920	8%
2	Number of LUT(4ip)	490	1920	25%
3	Number of Slices Used	310	960	32
4	Number of slice containing related logic	310	358	86%
5	Number of slice containing unrelated logic	0	358	0%
6	Total number of LUTS	550	1920	28%
7	Number of bonded IOBs	22	83	26%

Table1. Device utilization summary of Master Controller

2.1. I2C SLAVE

A Slave monitors the SDA and SCL lines for a START signal, unable to initiate data transfers independently. Upon detecting a START signal and matching its address with the received address, the Slave executes the Master's requested read or write operation. The Slave's design comprises six main functional blocks: SCL reception, Address Comparator, ACK Block, Data Receive Block, and ACK Block. The Start/Stop Detector module identifies Start signals, and the Address Comparator compares the sampled 7-bit address with the hardcoded Slave address. If the addresses match, an enable signal activates the Slave system. The ACK Block detects and generates acknowledgements to and from the Master. Table 2 represent the synthesis result of I2C Slave controller which explains the Device utilization summary of I2C

S.No	Logic utilization	Used	Available	Utilization
1	Number of Registers	110	126800	0 %
2	Number of LUTs	190	63400	0%
4	IOBs	22	210	10
5	BUF used	2	128	0

Table2. Device Utilization summary of Slave

3. THE PROTOTYPE SYSTEM

The prototype system will integrate the I2C Interface on both patient and clinical ends using FPGA, enabling USB-UART communication between the PC and FPGA. Xilinx's User Constraints File (UCF), an ASCII file generated by the user, links physical pin connections to TOP signals, influencing the logical design's implementation in the target device. The UCF file, added to the Xilinx project, details all physical connections of the prototype, including constraints and connections per the design specification.

The system necessitates connecting the SDA pins (designated pmod) of both FPGAs and shorting the ground to obtain valid results. Initially, the SCL lines remain unconnected. The connection occurs post-programming of both FPGAs, as the programming process sends an impulse on the PMODS, functioning as a parasitic clock impulse. The Figure 9 shows the Prototype of FPGA based system where one FPGA act as a Master and other one is Slave.

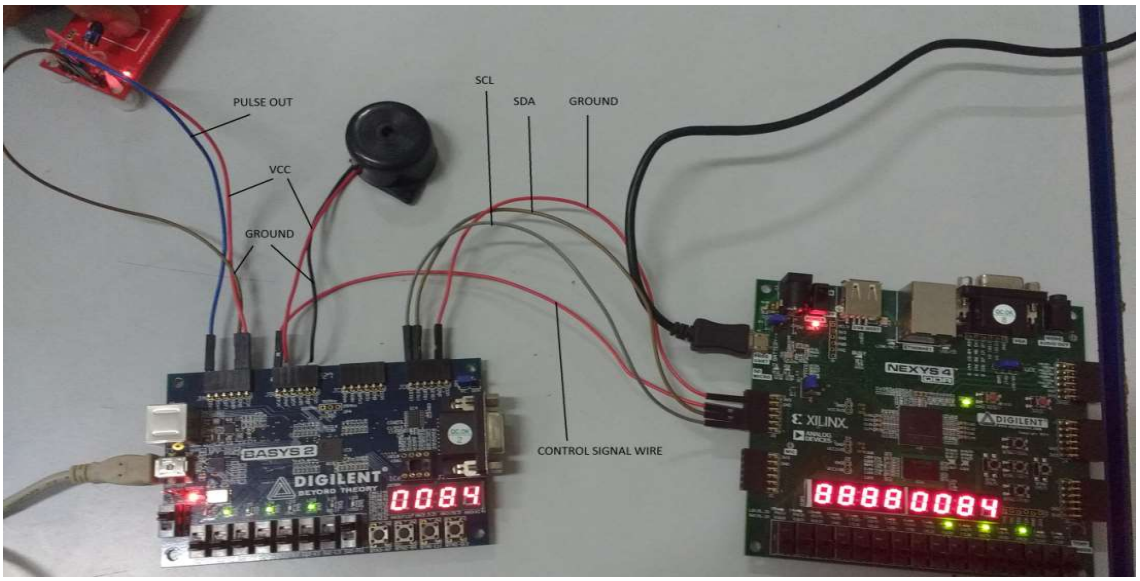

Figure9.Connections in the prototype

4. ***CONCLUSION :*** An I2C interface (in Verilog®) and a USB-UART interface were developed and tested on cost-effective Xilinx FPGA development boards, enabling rapid prototyping of biomedical sensors. The I2C interface operated at 100 KHz in 7-bit address mode, while the USB-UART interface worked at 9600 baud rate. Patient data received via the I2C protocol is stored in a register, displayed on a 7-segment display, and transmitted to a PC through a UART-USB/JTAG Bridge. An FTDI Driver converts the USB port into a Virtual COM port, and a hyper terminal YAT, configured according to standard UART protocol, allows the clinical end to log the received data. This type of system can be enhanced in future by integrating more number of sensors who can monitor different things and can be extended into IoT based monitoring system .

5. *REFERENCES*

[1] Srivastava, R., Prusty, C.K.K., Sahai, N., Tewari, R.P. and Kumar, B., 2022. An FPGA-based design for power efficient low delay rate adaptive pacemaker using accelerometer and heart rate sensor. *Journal of Medical Engineering & Technology*, *46*(1), pp.16-24.

[2] Abushukor, S.F., Syafalni, I., Mulyawan, R., Sutisna, N., Ahmadi, N. and Adiono, T., 2021, November. FPGA Implementation of IoT-Based Health Monitoring System. In *2021 15th International Conference on Telecommunication Systems, Services, and Applications (TSSA)* (pp. 1-5). IEEE.

[3] Udayashankar, D.K., 2023. Sampling Acceleration and Orientation Using SPI Between the iCE40UP FPGA and the LSM6DSOX IMU for In Situ Fall Detection and Classification (Master's thesis, San Diego State University).

[4] Akinwole, O.O., 2020. Design, simulation and implementation of an Arduino microcontroller based automatic water level controller with I2C LCD display. International Journal of Advances in Applied Sciences (IJAAS), 9(2), pp.77-84.

[5] Lin, K.H., Tseng, T.Y. and Wang, Q.T., 2023, October. LCD interface to I2C protocal conversion using FPGA. In *IET International Conference on Engineering Technologies and Applications (ICETA 2023)* (Vol. 2023, pp. 124-125). IET.

[6] Montsech Bravo, I., 2022. Design and Development of an FPGA-based Medical EIS System (Master's thesis, Universitat Politècnica de Catalunya)..

[7] Feng, S. and Tan, L., 2020. Simulation of sports and health big data system based on FPGA and Internet of Things. *Microprocessors and Microsystems*, p.103416.

[8] I2C-bus specification and user manual, NXP Semiconductor, Rev. 6 — 4 April 2014

[9] Ishak, M.K. and Kumar, M.P., Design and Implementation of I2C Bus Protocol on Master and Slave Data Transfer Based on FPGA. Makara Journal of Technology, 26(1), p.5.

[10] Yuan, B., 2023, June. A non-contact intelligent temperature acquisition and monitoring system based on

FPGA. In International Conference on Intelligent Systems, Communications, and Computer Networks (ISCCN 2023) (Vol. 12702, pp. 169-175). SPIE.

[11] Regu Archana, Mr. J.V.Rao, "Implementation of I 2C Master Bus Protocol on FPGA", Int. Journal of Engineering Research and Applications, ISSN : 2248-9622, Vol. 4, Issue 10 (Version 5), October 2014, pp.06-10

[12] Guo, J., Jiang, B. and Xu, H., 2021, June. Fpga-based uav heterogeneous computing platform architecture. In 2021 6th International Conference on Computational Intelligence and Applications (ICCIA) (pp. 252-256). IEEE.