A Capstone Project Report On

Implementation of Serial Communication Protocols using FPGA board

Submitted by: *Kirtan Makwana – 92200133031*
Guided by: Prof. Chandrasinh Parmar & Prof. Vijay Dubey

A Project Report Submitted to
Marwadi University in Partial Fulfillment of the Requirements for the B. Tech in Information Communication Technology

MARWADI UNIVERSITY
Rajkot- Morbi Road, At & Po. Gauridad, Rajkot-360003, Gujarat, India.

Contents

1. Abstract 1

2. Introduction 1

3. ICT Relevance: 2

4. System Design and Architecture 2

4.1 Hardware Platform 2

4.2 Modular Architecture of UART 3

4.3 Modular Architecture of SPI: 4

4.4 Modular Architecture of I2C: 5

5. Implementation and Outcomes 6

6. Implemented Hardware: 7

7. Key Outcomes 9

8. Conclusion 9

Marwadi University	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Capstone Project (01CT1718)	Capstone Project Final Report	
	Name: Kirtan A. Makwana	Enrolment No: 92200133031

1. Abstract

This project implemented fundamental communication protocols (UART, SPI, and I2C) on Field-Programmable Gate Array (FPGA) platforms using Verilog HDL. The primary objective was to move beyond software-driven microcontroller solutions to create robust, reconfigurable, and high-performance hardware implementations suitable for educational and industrial prototyping. The design used a modular architecture and was deployed on the Xilinx Nexys 4 DDR board, utilizing its built-in ADXL362 accelerometer (for SPI) and ADT7420 temperature sensor (for I2C) to validate real-world sensor interfacing. The successful deployment and testing confirm the system's reliability and its value as a practical learning tool for digital logic design, VLSI, and embedded systems.

2. Introduction

Reliable communication between system components like sensors, modules, and processors is critical in modern digital system design, especially within IoT and embedded systems. While protocols such as UART, SPI, and I2C are pervasive, their implementation is predominantly microcontroller-based.

This project addresses the gap by providing a hands-on, FPGA-based implementation, offering students and developers a practical platform to understand the underlying hardware complexity and realize the benefits of high-speed parallel processing and reconfigurability inherent to FPGAs. The project's reconfigurable nature, contrasting with fixed-function Application-Specific Integrated Circuits (ASICs), makes FPGAs highly versatile for prototyping and deployment.

3. ICT Relevance:

This project fits directly into ICT areas like Digital Logic Design, VLSI, and Embedded Systems. It also supports modern ICT trends:

- IoT: Most IoT devices rely on UART, SPI, and I2C for sensor communication.
- VLSI/Chip Prototyping: FPGAs are used as the first step before chip manufacturing.
- Industry Readiness: Students trained in FPGA communication protocols are better prepared for careers in hardware design and system integration.

In this project we design the basic requirement of decided topic with digital design flow and then according to flow of the definition we decide to use which kind of method to use in Verilog code to design the architecture of the definition then make the code such that it can be synthesizable and implementable on FPGA board. By following this process with different trials, we can accomplish the final definition requirements.

By solving this problem, the project not only helps students and educators but also strengthens the link between academic training and industry requirements in ICT.

4. System Design and Architecture

The project employs a modular design approach, where each communication protocol (UART, SPI, I2C) is implemented as an independent, reusable module in Verilog HDL. A single top-level module is responsible for integrating and controlling these separate protocol blocks.

4.1 Hardware Platform

The core hardware is the **Xilinx Nexys 4 DDR FPGA board** (featuring the XC7A100TCS0324A chip). This board was chosen for its widespread use in academia and its onboard peripherals that simplify integration testing.

- **UART:** Implemented for basic data transmission and receiving, typically using a loopback or USB-to-UART interface for monitoring.

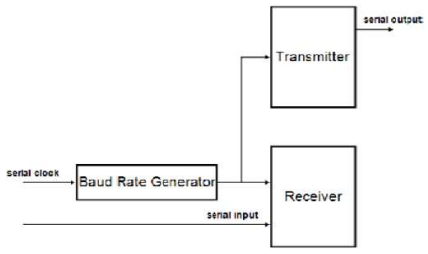


Fig. 1 UART Module

- **SPI Sensor:** The onboard ADXL362 3-axis accelerometer is used to demonstrate SPI communication. The FPGA acts as the Master, reading 12-bit acceleration data from the sensor's registers.

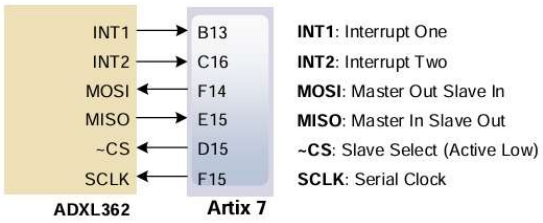
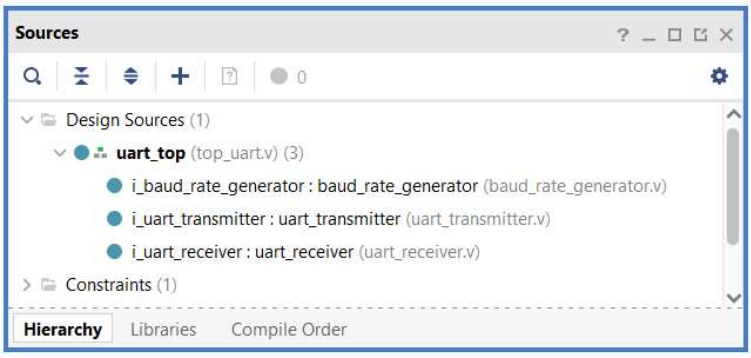


Figure 23. Accelerometer interface.

- **I2C Sensor:** The onboard ADT7420 temperature sensor is used to validate I2C communication, with the FPGA serving as the Controller/Master. The I2C bus address 0x4B was selected for interfacing.

4.2 Modular Architecture of UART

The UART protocol implementation is subdivided into three key sub-modules:



1. **Baud Rate Generator:** This module is critical for creating a serial clock pulse (baud_clk_en) from the much faster system clock (e.g., 100 MHz) to match standard baud rates (e.g., 9600 bps). The calculation for the counter limit is given by the formula:

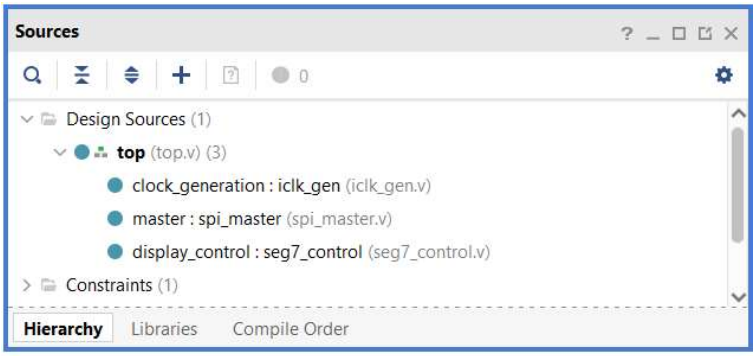
$$\mathbf{COUNTER\_LIMIT = (CLOCK\_FREQ + (BAUD\_RATE/2)/BAUD\_RATE) - 1}$$

2. **Transmitter:** Handles the parallel-to-serial conversion of data and adds start/stop bits.
3. **Receiver:** Handles the serial-to-parallel conversion of received data.

The overall flow for the UART protocol follows a sequence of checking for trigger and interrupt pending status for both transmission (TX) and reception (RX), followed by data availability or processing.

4.3 Modular Architecture of SPI:

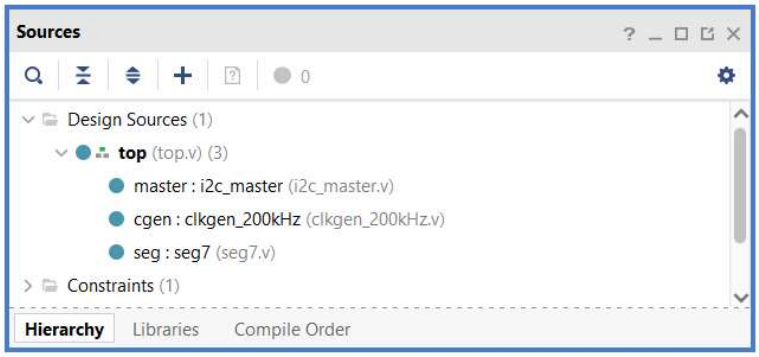
The SPI protocol implementation is subdivided into three key sub-modules to read data from ADXL362:



1. **Clock Generation**
   The FPGA's system clock is much faster (e.g., 100 MHz) than the communication protocol speed. This module divides the main clock to create a slower, synchronized Serial Clock (SCLK) for the SPI bus. In the context of UART, this is analogous to the Baud Rate Generator.
2. **SPI Master**
   This module contains the state machine logic for Asserting/deasserting the CS line. Sending register addresses and commands on the Master Out Slave In (MOSI) line. Receiving data from the peripheral on the Master In Slave Out (MISO) line, often receiving 1 or 2 bytes of data (e.g., 16 bits for the ADXL362).
3. **Display Control**
   This module receives the raw data (e.g., 16-bit acceleration data) from the spi_master.v module, processes it (e.g., converting binary to BCD for decimal display), and controls the segments of the display to show the final output.

4.4 Modular Architecture of I2C:

The I2C protocol implementation is subdivided into three key sub-modules:



1. **I2C master device:** This module is responsible for controlling the bus between the itself and peripheral/sensors. Which basically do data transmission with requirement of read or write. It synchronizes with the SDA line with SCL by implementing concept like setup and hold time while sampling the data on the bus
2. **Clock Generator:** The clock generator will generate the clock by taking the 100MHz clk frequency to map to 200kHz and send it to top module on SCL line.
3. **Segment Controller:** Using the same clock of 100MHz we can control each 7 segment to display digits converted from received data by taking modulus of it and display in each display changing its value every 10ns, which is not catchable for human eye.

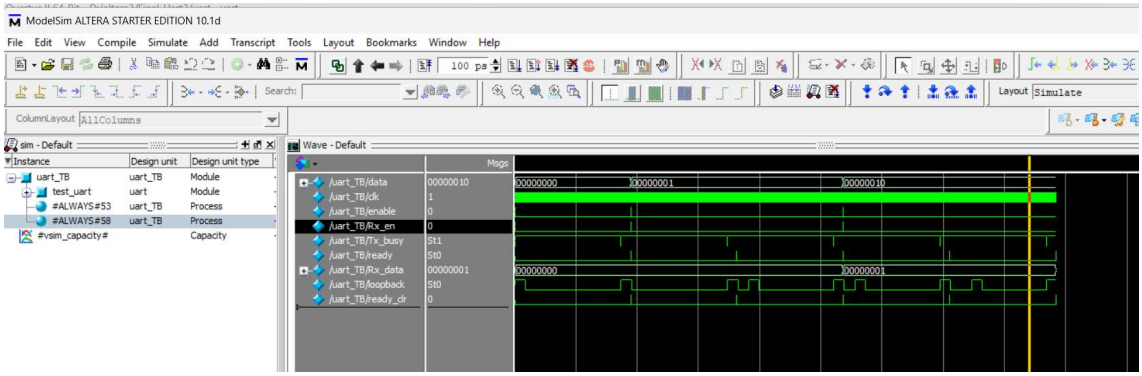
Overall the structure follows I2C protocol to read and write data and address on SDA line with SCL control we can also modify the mode of frequency in SCL line in it.

5. Implementation and Outcomes

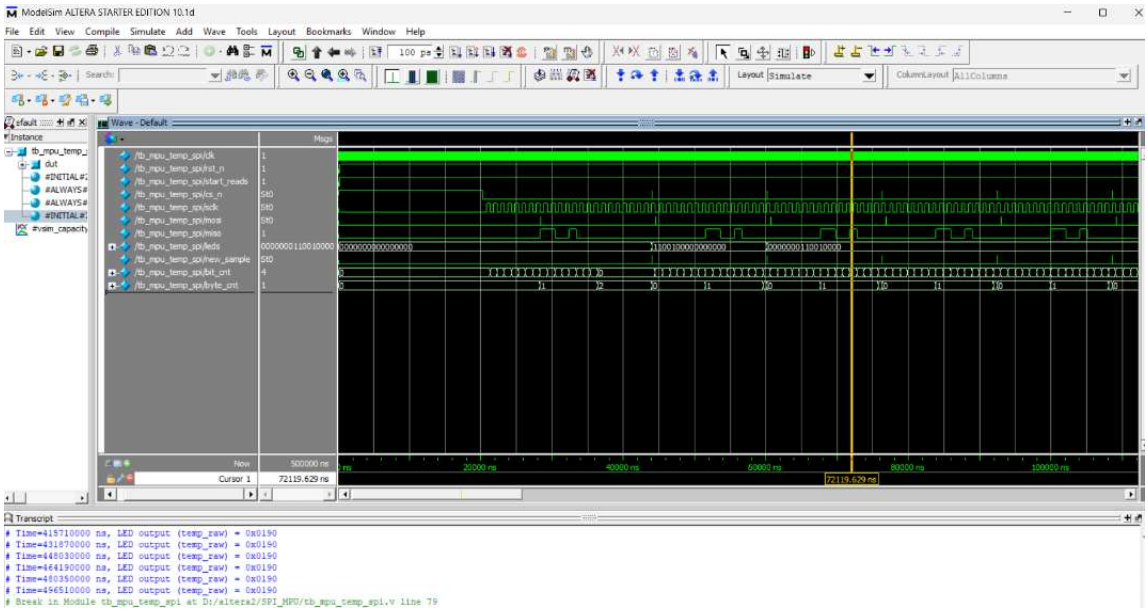
The project utilized Verilog HDL for all hardware designs, and Vivado (for Xilinx) and Quartus II (for Altera) for synthesis and implementation.

ModelSim was used extensively for simulation and verification via testbenches before deployment to the physical hardware.

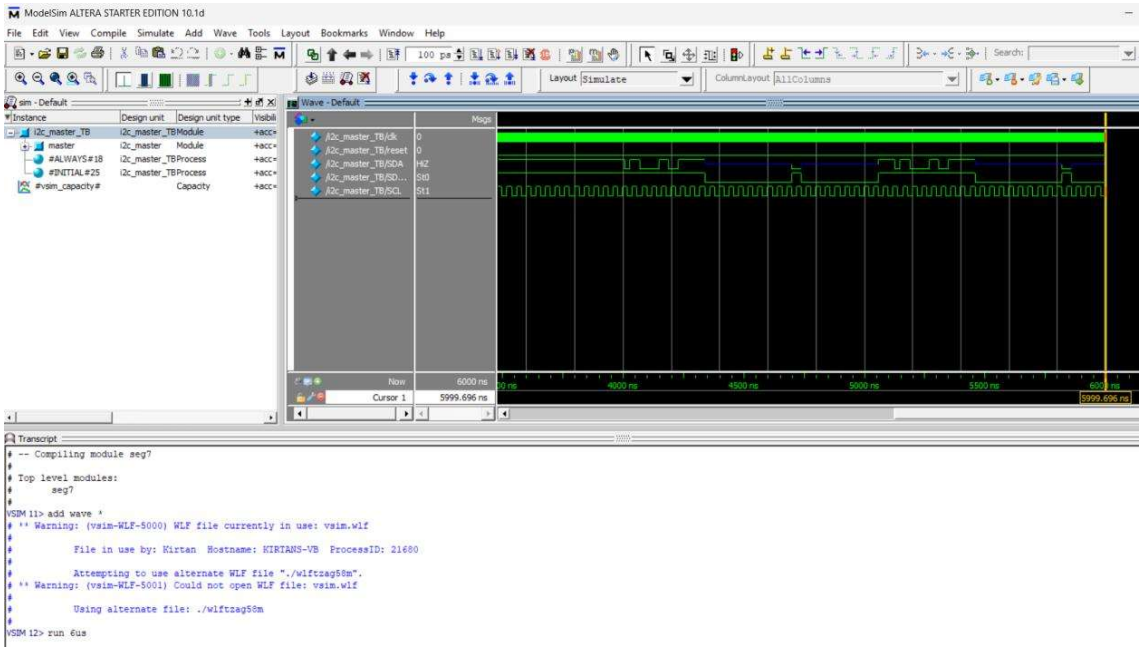
UART simulation testing :



SPI simulation testing :

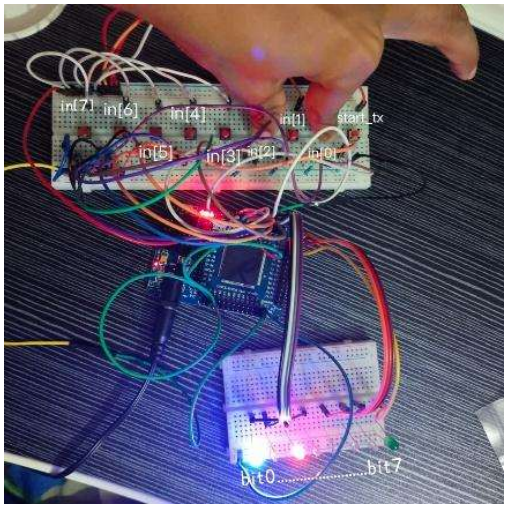


I2C simulation testing :

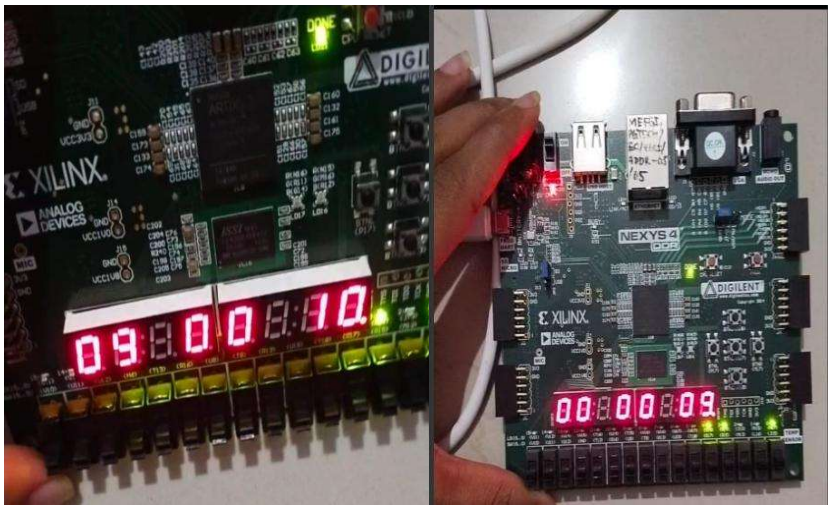


6. Implemented Hardware:

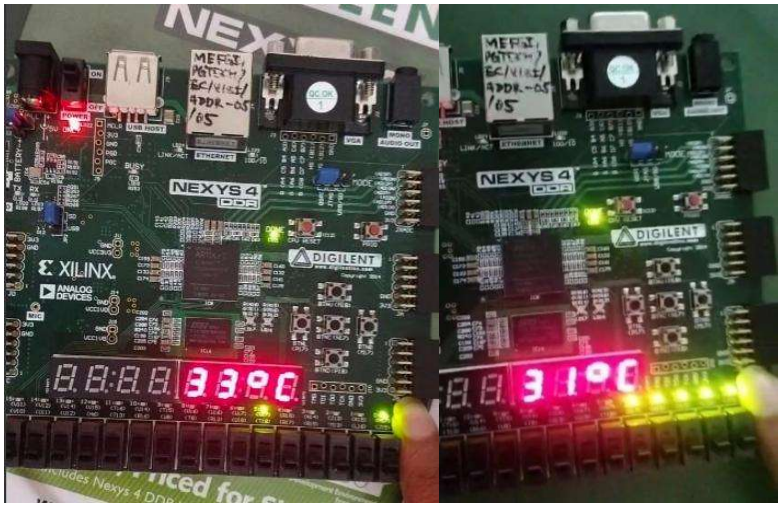
UART with RX TX: The below photo is test of UART on FPGA board of Altera Cyclone 2 whre while pressing start TX button we ca send desired data and see the same on LED by connecting RX and TX together.



SPI with 3 axis Acceleration reading: The SPI protocol was implemented to communicate with the ADXL362 accelerometer. The result, representing X, Y, and Z axis acceleration data, was successfully displayed on the Nexys 4 DDR's seven-segment display.



I2C with Temperature reading: The I2C Master module was designed to interact with the ADT7420 temperature sensor. The captured temperature data was successfully displayed on the seven-segment display.



7. Key Outcomes

The project successfully demonstrated the implementation of UART, SPI, and I2C on FPGA hardware. The outcomes include:

- A fully functional FPGA-Based Protocol Implementation in Verilog, complete with tested hardware demos.
- A clear pathway illustrating how these designs can be scaled and extended towards ASIC design flow.

8. Conclusion

This Capstone project successfully implemented and validated core communication protocols (UART, SPI, I2C) on the Xilinx Nexys 4 DDR FPGA platform using Verilog HDL. By interfacing with built-in, real-world sensors, the project delivers a highly practical and educational platform that meets the needs of students, educators, and hardware developers. The modular, Verilog-based design offers superior performance and flexibility compared to traditional microcontroller solutions, strengthening the link between academic training and industry requirements in the ICT domain.