Project: Implementation of Communication Protocol on FPGA

Ideation and stakeholder need analysis

I. Stakeholder Identification and Needs

Communication protocols like **UART, SPI, and I2C** are the building blocks of embedded systems, IoT devices, and automation. But in schools, colleges, and even early-stage industries, people often struggle to implement and test these protocols in practice.

The main stakeholders and their needs are:

- **Students and Learners**:
  Mostly learn protocols in theory. Without hands-on work, it's hard to understand timing, data flow, and synchronization. They need a **low-cost, practical platform** to practice on.
- **Colleges and Universities**:
  Need working lab setups that show real-time communication between devices. The issue is the lack of **flexible and reusable teaching tools** that cover many protocols.
- **Developers and Hobbyists**:
  When building IoT or embedded projects, they need prototypes to test communication between devices. Debugging becomes hard when there is no **structured test platform**.
- **Industry and Research Labs**:
  Require testing setups before going into large-scale production. They need **multi-protocol testing platforms** that can simulate and validate communication in a controlled way.

Reports from **IEEE (2024)**, **Gartner (2023)**, and **McKinsey (2024)** show that hands-on training and reliable communication setups are becoming essential to bridge the gap between classroom learning and real-world industry needs.

II. Problem Statement

"Students, educators, and developers lack an affordable and flexible platform to practically implement and test communication protocols like UART and SPI on FPGA. This creates a gap between theory and real-world ICT applications."

This is important because:

- Without practice, learners can't build the right skills for IoT or embedded jobs.
- Developers and professionals also struggle to debug device communication without reliable tools.

III. Solution Ideas

FPGA–Arduino Communication Platform (Implemented)

- Data flow: PC → UART → FPGA → Arduino (SPI) → Data increment → FPGA (SPI) → PC.
- FPGA = SPI master, Arduino = SPI slave.
- Shows how UART and SPI work together in real time.
- **Benefit**: Learners see the actual data transfer and changes.

Web Dashboard for Visualization (In Progress)

- Browser-based tool to detect ports and show communication.
- Planned feature: real-time waveforms of UART and SPI.
- **Benefit**: Makes learning easier and supports remote labs.

Extension to More Protocols (Future Work)

- I2C already tested in simulation.
- Future additions: CAN bus, Ethernet, etc.
- **Benefit**: Expands the platform for more industry use.

Conclusion

There is a clear gap between learning communication protocols in theory and applying them in practice. This project—by combining **UART and SPI on FPGA, Arduino integration, and a web dashboard**—fills that gap. It is aligned with current ICT trends like IoT, edge computing, and embedded systems, making it useful for **education, research, and industry applications**.