Department of Telecommunications Engineering

TC-490 Final Year Design Project

Proposal for the Final Year Design Project

Title	RISC-V System on Chip (SoC) for Communication

Domain	**Domain 1** Digital Logic Design	**Domain 2** Embedded Systems	**Domain 3** Integrated Circuits	**Domain 4** On-Chip Communication	**Domain 5** -	**Domain 6** -

1. Nature of Project [Tick all that applicable]

✓ New Project OR ☐ Extension of Existing Project	✓Industrial Collaboration	☐ Funded
☐ Other Department Collaboration (If yes) Department Name_____	☐ Other Academic Institution Collaboration (If yes) Institution Name_____	

2. Brief Outline (*Problem Identification and Significance*)

RISC V processor requires 32 bit data to be processed and a separate instruction set to perform any manipulation and processing on the data. In order to pass the data words to the processor, one can use a Communication protocol to establish a connection between the processor and user to send the data bits efficiently. Serial communication protocols such as I2C and UART can be used to send and receive data to and from the processor as shown in **Figure 1**. The processor only processes the data given to it but in order to store the data, it requires a separate data bank to store the real-time processed data, so an additional memory block is used to share with the processor (typically an external FPGA ROM) to hold the data while it's being processed [1], [4], [8].

1. **RISC-V Processor Design**:

The project begins with the design of a 32-bit RISC-V processor using System Verilog HDL adhering to the specifications of the RISC-V instruction set architecture (ISA). The design process will incorporate a five-stage pipeline architecture to optimize performance [3], [4], [6], [10].

2. Communication through UART Protocol:

The project will integrate a UART protocol for serial communication, based on existing designs that focus on efficient and reliable data transfer in FPGA environments [5], [7], [9]. It will be responsible for data communication between the user and the RISC V processor [2]. The project will explore adaptive baud rate control in the UART protocol to improve communication efficiency [13], [14], [15].

3. Data Retrieval through I2C Protocol:

The I2C protocol will be implemented to enable data retrieval between the RISC-V processor and the output. The I2C interface will be crucial for establishing a reliable connection with sensors and other peripherals in the SoC environment [10].

4. System Integration and Testing:

Finally, the project will integrate the RISC-V core with the UART and I2C communication modules, followed by rigorous testing to ensure system reliability and performance under various operating conditions. The integration will be validated through simulation and real-time FPGA implementation [1], [11].

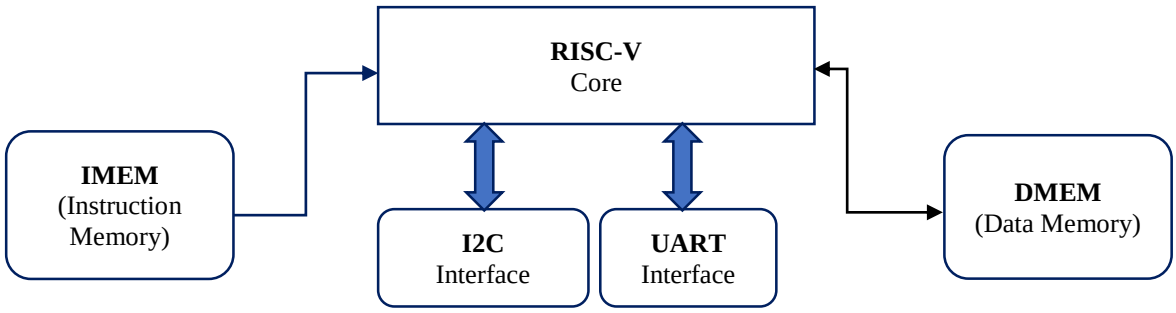

Figure 1: RISC-V Core Interface using I2C and UART Protocol

The data will be sent by the user (master) to the Processor through UART, processed according to the Instruction set fed in the processor, and finally the output generated by the processor will be brought back to user (slave) through I2C as shown in **Figure 2**. [10]

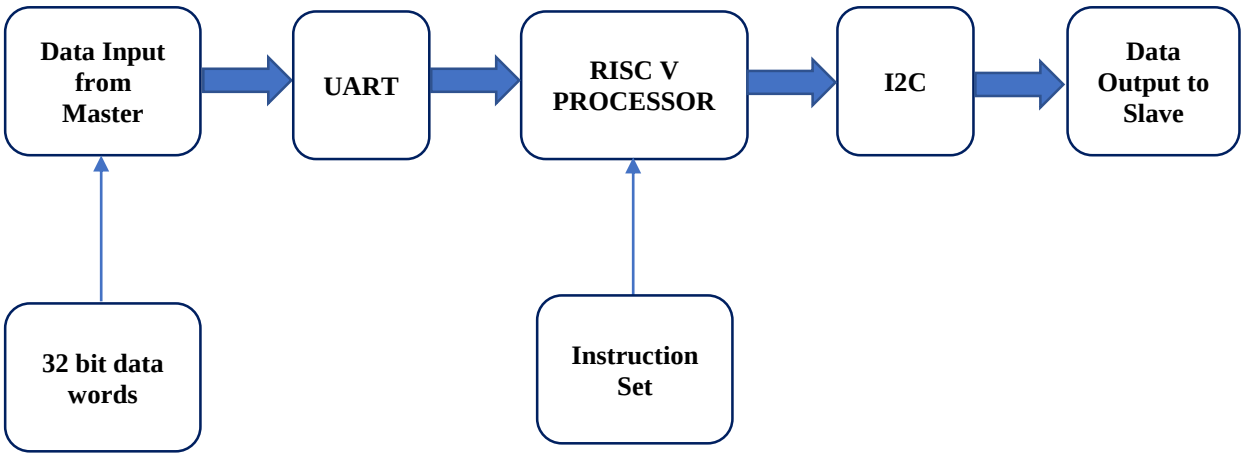

Figure 2: Complete Functional Block Diagram of the System

This system illustrates a data communication process which incorporated a RISC-V as the central processing unit from **Figure 2**.

1. Data Input from Master: The system first involves accepting 32-bit data words from user (master). These data words could be any type of data that have to be processed or sent to some other device for further processing.

2. UART (Universal Asynchronous Receiver/Transmitter): The received 32-bit data words are then passes to a storage bank (typically a continuous memory slot) by which packets of bits will be sent one-by-one to the processor's memory.

3. RISC-V Processor: The most central part of the system is the processor which is a RISC-V processor at the moment. Once the data is received by the processor, it will start to process the data according to the instructions fed in it by the user in its instruction set memory. The instruction set are operations that a RISC-V processor can perform. These may be arithmetic / logical operations, branching or controlling instructions. These instructions dictate to the processor on how to process and analyze the data that comes in. The RISC-V processor may also engage in some forms of processing including transformation of data, computation or making data ready for other action.

4. I2C (Inter-Integrated Circuit): The processed data is thereafter passed in the form of packets of bits to another storage bank through I2C bus so that they can be transferred to other slaves or fetched to the user as processed data.

5. Data Output to Slave: The last level of the system enlists the transmission of I2C formatted data generated by the RISC-V processor to a slave device. The slave device will often be another peripheral or another module that requires the processed data to be able to perform its functionality. Through the I2C protocol the data is transmitted in an efficient and proper manner to the slave device marking the end of this communication process.

3. Objectives

Our primary objective is to develop UART and I2C modules, complete with testbenches for each, and construct a fully operational RISC-V processor core, which will be pipelined to optimize task execution and efficiency. Integration of the UART and I2C protocols will facilitate communication between input/output devices. Additionally, a top-level controller module will be created to manage system parameters and monitor outcomes. To ensure comprehensive validation, we will separately test each module and then integrate them to evaluate the overall system's performance. This includes processing various data streams, generating results, and verifying the system's behavior.

4. Scope

The scope of this project is to design and implement a high-performance RISC-V processor core tailored for efficient instruction execution, ensuring that it meets system performance criteria. This includes the development of a robust architecture that supports the integration of UART and I2C protocols, enabling seamless communication between the processor core and user. The project will emphasize the optimization of these communication protocols to ensure reliable and efficient data transfer, which is critical for the overall system's functionality [8-11].

A key component of the project is the creation of a comprehensive testing and verification framework. This framework will be designed to rigorously test the system by sending data through the integrated communication protocols, processing it with the RISC-V core, and validating the output against expected results. The goal is to ensure that the system behaves correctly under various operating conditions, thereby verifying the reliability and accuracy of the processor core and communication interfaces.

In the final phase, the project will focus on system validation under real-world conditions, identifying and addressing any issues related to protocol communication, data processing, and output generation. This will ensure that the entire system operates as intended in a practical environment, meeting all functional and performance requirements.

5. Proposed Methodology

Our project will involve gathering and analyzing information from resources, as illustrated in **Figure [3].** We will focus on studying the design, complexity and performance of our communication platform through detailed computer simulation. This will enable us to understand how well the system performs in real world situations and it’s response for any potential improvements.

We’ll start by collecting the necessary information for implementing the RISC V processor, I2C and UART protocols. Through which, a big picture will be made for the complete communication system.

We start by designing the main module for our processor. Firstly, in designing the processor, the **ALU**, **decoder**, **mux** and **control unit** will be designed for the arithmetic and logical computation, then **register file**, **data memory** and **instruction memory** blocks will be designed. A **memfile** will be also designed to feed the instructions to be performed by the processor. Now, all the sub modules will be connected by each other through a main module so that the complete processor will behave like a black box.

After the processor is made, UART protocol will be designed. The complete **state machine** of the UART Tx will be designed. A **clock divider** will also be designed to divide the high speed system clock so that data bits transmitted from the user to the processor will be sampled at the exact moment at the receiver and reduce any kind of false sampling.

Now, the I2C will be designed. Almost same process will be executed for designing I2C. A **state machine** will be designed for the complete protocol and a **clock divider** will be made same as the UART here to synchronize the bits at the Tx and Rx.

Now the 3 main modules (Processor, I2C, and UART) will be connected to each other also having the IO ports now. The user will firstly transmit the data in to processor by sending it through UART, after the data is received at the processor, the computation will take place inside the processor through the commands fed in the processor implicitly (through the **memfile**). The output generated by the processor will be brought back to user through I2C completing the communication.

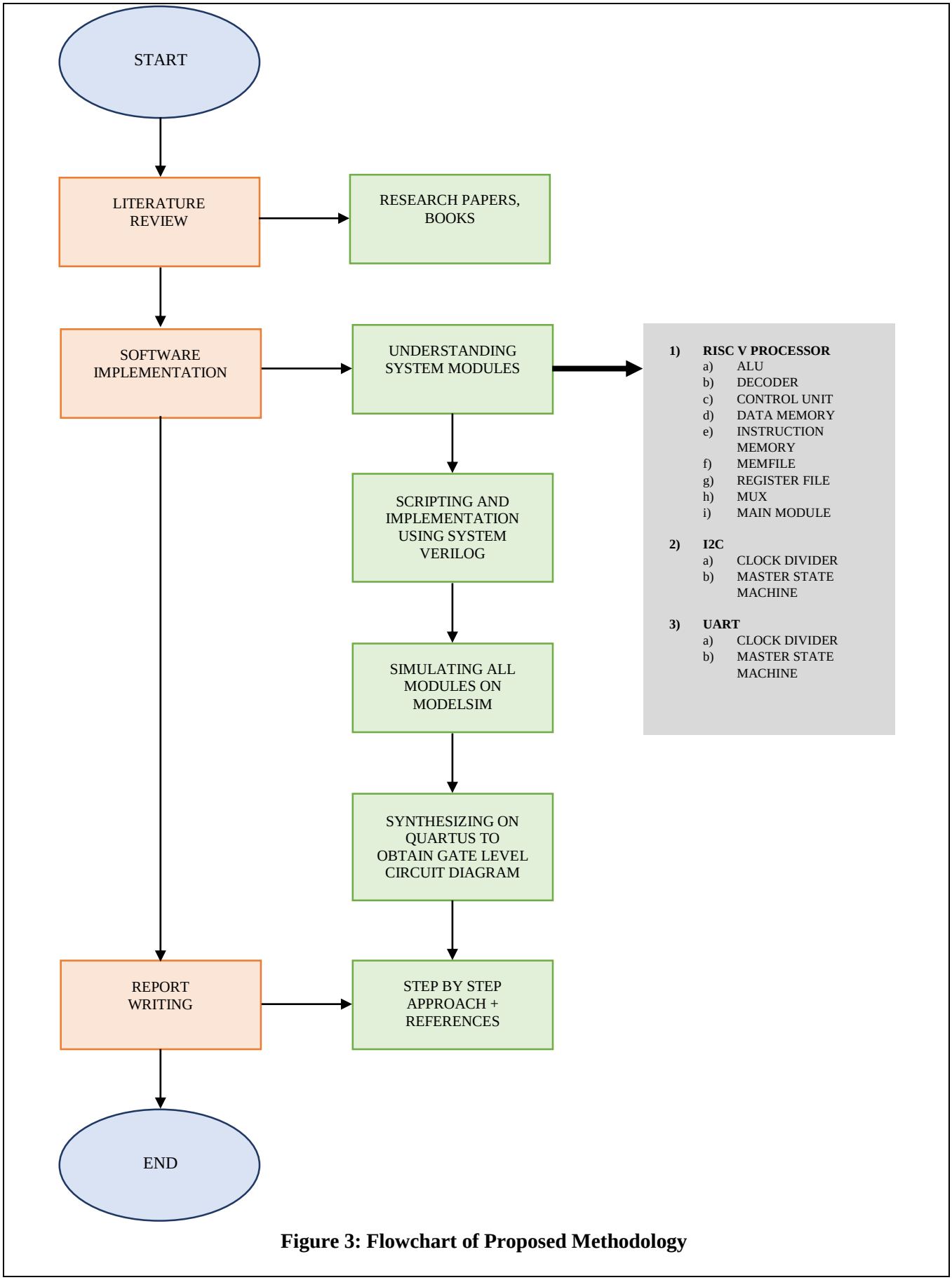

Figure 3: Flowchart of Proposed Methodology

6. Resources Involved

Included resources are:

PC with following applications installed:

i) Intel Quartus Prime (It's an environment that includes all the tools needed to design and test modules for FPGAs, SoC, and CPLDs). [17]

ii) ModelSim (It's a software designed by Seimens for HDL simulation using Verilog, System Verilog and System C). [17]

iii) GTKWave (It's a waveform viewing used to view and debug designs written in Verilog or VHDL simulation models). [18]

iv) Visual Studio Code (It's a widely used code editor to write and modify codes with its powerful integration with Microsoft's extensions). [19]

7. Description of Industrial Support (If any)

Not applicable

8. SDGs (If Applicable)

☐ No Poverty	☐ Zero Hunger
☐ Good Health and Well-Being	☐ Quality Education
☐ Gender Equality	☐ Clean water and Sanitation
☐ Affordable and Clean Energy	✓ Decent Work and Economic growth
✓ Industry, Innovations and Infrastructure	☐ Reduced Inequalities
☐ Sustainable Cities and Communities	✓ Responsible Consumption and Production
☐ Climate action	☐ Life Below Water
☐ Life on Land	☐ Peace, Justice and Strong Institutions
☐ Partnerships	

9. Gantt Chart

Year	2024 - 2025									
Months	Aug. 2024	Sep. 2024	Oct. 2024	Nov. 2024	Dec. 2024	Jan. 2025	Feb. 2025	Mar. 2025	Apr. 2025	May. 2025
TASKS										
Literature review for project	■	■			■	■		■	■	
Complete RISC V Single Cycle Processor			■	■	■					
Understanding and Simulation of modules				■	■	■	■			
Implementation of Protocols						■	■	■	■	
Verification using ModelSim							■	■	■	■
Pipeline and architecture									■	■
Report writing	■	■	■	■	■	■	■	■	■	■

10. Details of Project Team

i. Students

No.	Name	Seat No.	Signature (s)
1	Rao Muhammad Umer	TC-21073	
2	Huzaifa Hassan	TC-21060	
3	Muhammad Kashaf Khan	TC-21065	

ii. Supervisors / Advisors

	Name	Designation & Department	Address & Contact	Signature(s)
Supervisor	Dr. Muhammad Fahim-ul-Haque	Assistant Professor, Telecommunications Engineering department	NED University of Engineering and Technology, mfahim@cloud.neduet.edu.pk	
Co-Supervisor (If any)	Ms. Hafsa Amanullah	Lecturer, Electronics Engineering Department	NED University Of Engineering And Technology, hafsa@cloud.neduet.edu.pk	
Industrial Advisor (If any)	Fasahat Hussain	Technical Program Manager At DreamBig Semiconductor Inc.	NED University of Engineering and Technology, fasahat@dreambigsemi.com	

References:

[1] L. Poli, S. Saha, X. Zhai and K. D. Maier, "Design and Implementation of a RISC V Processor on FPGA," 2021 17th International Conference on Mobility, Sensing and Networking (MSN), Exeter, United Kingdom, 2021.

[2] J. Chen and S. Huang, "Analysis and Comparison of UART, SPI and I2C," 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 2023.

[3] J. -Y. Lai, C. -A. Chen, S. -L. Chen and C. -Y. Su, "Implement 32-bit RISC-V Architecture Processor using Verilog HDL," 2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Hualien City, Taiwan.

[4] F. Hussain and S. Sarkar, "Design and FPGA Implementation of Five Stage Pipelined RISC -V Processor," 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India.

[5] Gupta, A. K., Raman, A., Kumar, N., & Ranjan, R. (2020). Design and Implementation of High-Speed Universal Asynchronous Receiver and Transmitter (UART).

[6] Cui, E., Li, T., & Wei, Q. (2023). RISC-V Instruction Set Architecture Extensions: A Survey. IEEE Access, 11, 24696-24711.

[7] Dakua, B. K., Hossain, M. S., & Ahmed, F. (2019). Design and Implementation of UART Serial Communication Module Based on FPGA. In International Conference on Materials, Electronics & Information Engineering, ICMEIE-2019.

[8] Dennis, K., Borriello, G., Gahlinger, M., & Montrym, J. (2017). Single cycle RISC-V micro architecture processor and its FPGA prototype. In 2017 7th International Symposium on Embedded Computing and System Design (ISED) (pp. 1-5).

[9] Du, Z., Liu, Y., Qiu, C., & Zhang, X. (2023). Verilog implementation of configurable UART module. Proc. SPIE 12597, Second International Conference on Statistics, Applied Mathematics, and Computing Science (CSAMCS 2022).

[10] Harris, S. L., & Harris, D. M. (2021). Digital Design and Computer Architecture, RISC-V ed. United States of America: Elsevier.

[11] Qi, S., Jin, S., Xu, Y., & Dai, Y. (2022). A five-stage pipeline processor using the RISC-V instruction set architecture designed by System Verilog.

[12] Analysis and Comparison of Asynchronous FIFO and Synchronous FIFO. In 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA).

[13] N. RS, S. S, S. M. A, S. P. M and M. C, "Implementation Of I2C Protocol With Adaptive Baud Rate Using Verilog," 2024 7th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 2024.

[14] Huang, C., Yang, S. A mini I^2C bus interface circuit design and its VLSI implementation. *J Supercomput* **80**, 23794–23814 (2024).

[15] H. Wang and P. Qiao, "Design of IIC Interface Controller based on FPGA," 2024 9th International Conference on Electronic Technology and Information Science (ICETIS), Hangzhou, China, 2024.

[16] Visual Studio Code, "Visual Studio Code - Code Editing. Redefined".

[17] https://www.intel.com/content/www/us/en/collections/products/fpga/software

[18] https://gtkwave.sourceforge.net/

[19] https://code.visualstudio.com/download

For Office Use Only		
Project Serial No.: _____ Dated: _____	Signature Convener Steering Committee	Signature FYP Coordinator

☐ **Proposal Approved**	☐ **Not Approved**	☐ **Returned for Clarification / Modification**
Comments: (if any)		

_____

(Signature of Chairperson)

Date: _____