RISC-V SoC Project Proposal

**Document Control**

Document Information

|  |  |
| --- | --- |
|  | **Information** |
| Document Id | *-* |
| Document Owner | *Abdelrahman Mohamed Khalil* |
| Issue Date | *10/8/2024* |
| Last Saved Date | *10/8/2024* |
| File Name | *RISC-V\_SoC\_PP* |

Document History

|  |  |  |
| --- | --- | --- |
| **Version** | **Issue Date** | **Changes** |
| *[1.0]* | *10/8/2024* | *Document created* |
| *[2.0]* | *18/8/2024* | *Changed the phases and phases descriptions* |
|  |  |  |
|  |  |  |

Document Approvals

|  |  |  |
| --- | --- | --- |
| **Role** | **Name©** | **Date** |
| Project Sponsor | - | - |
| Project Review Group | Peer Review | - |
| Project Manager© | Abdelrahman Mohamed Khalil | 10/8/2024 |

**Background**   
The main aim of the project is to **learn more about processors specially the RISC-V** based processors. This includes mainly the architecture and micro architecture and implements hardware for the architecture chosen. **This will increase our personal experience in digital design, computer architecture and in project management as well.**  
**Objectives**

* Deliver high performance RISC-V Based SoC with basic peripherals (GPIO, UART, I2C)
* Finish the learning plan which is parallel to the project

**Scope**

The project will be a complete HDL code for multi-core RISC-V SoC with only GPIO, UART and I2C peripherals (these peripherals could be third part or outsourced). With UVM testbench and any required tests related to such processor. No ASIC or FPGA implementations will be done in this project; yet it will be included in the next versions of the project.

**Timeframe**

The project will start in 11/8/2024 with a maximum total duration of 2 months

|  |  |  |
| --- | --- | --- |
| **Project Phases** | | |
| **Title** | **Description** | **Duration** |
| Preparation Phase | This phase includes all the learning required for the project | 11/8/2024 – 17/8/2024 (1 week) |
| Single cycle & Multi cycle RV32I with exceptions and testbench | This phase includes designing the basic RV32I in single and multi-cycle microarchitectures and handle exceptions. In addition, all the required tools for testing are installed. | 18/8/2024 – 24/8/2024 (1 week) |
| Writing program for RV32I | Using the GNU toolchain, produce ready-to-upload object file from C code. | Estimated (3 days) |
| Pipelined and deep pipelined RV32I | This phase includes designing the pipelined version of the RV32I with exploring the deep pipelining | Estimated (5 days) |
| Hazard control (including branch prediction) | Adding the hazard control that does forwarding, stalling and simple branch prediction (even if it requires instruction cache) | Estimated (1 week) |
| Advanced microarchitecture (out-of-order, super scalar) | Making the super scalar and out-of-order version of the pipelined architecture made previously | Estimated (1 weeks) |
| Pipelined RV32IMC | Adding M and C extensions | Estimated (5 days) |
| Adding cache | Adding all the necessary hardware to add cache to the system (at least 1 cache is added) | Estimated (2 weeks) |
| Adding AHB protocols | Add AHB master to the processor and AHB slave for the peripherals | Estimated (2 weeks) |

**Deliverables**

|  |  |  |
| --- | --- | --- |
| **Project Deliverables** | | |
| **Title** | **Description** | **Notes** |
| HDL Source code | All files that include the HDL code of the project | - |
| UVM Testbench | UVM testbench for the design in each stage | This may include any tests other that UVM |
| Documentation | Detailed documentation for the architecture | It is required for further development |

**Project Budget**

No budget for the project.