

# CS209P Project Phase 1

January 28, 2025

- Develop a simulator on the lines of Ripes. But note that the simulator should have four processors (or cores) and should be able to simulate a multi-core environment. In the later phases we will convert these cores to GPU cores.
- In this stage of the project, the four compute units do not have to communicate with each other. But they should be able to access the same memory.
- Find the RISC-V manual [here]. Be sure to read their commentary on design decisions (written in small font) for fun insights.

Please note that you do NOT have to read the entire manual. It is **NOT** a textbook. Your simulator should support the following RISC-V instructions: ADD/SUB, BNE, JAL(jump), LW/SW, and an instruction of your choice. You might want to add in some immediate-type instructions too!

Each core should have a special purpose register that will store the core number. For example, if you have 4 cores, core 0 will have 0 in this register, core 1 will have 1 in this register, and so on. This register should be read-only. An assembly file for the simulator will be provided. Each instruction takes 1 clock cycle. Example usage:

```

LABELx: add x0, x0, x1  #rd, rs1, rs2
        sub x0, x0, x1  #rd, rs1, rs2
        bne x0, x2, LABELy #rs1, rs2, offset (but label will
        suffice)
        sw x0, 0(x3) #rd, offset(rs1)
LABELy:  sub x0, x0, x2
        jal x0, LABELx  #rd, offset (again, label will work)
```

Note that rd and rs stand for destination and source registers. Comments start with #

If you want your simulator to support functions, you can implement jal and jalr opcodes. It is not a requirement for this phase.

- The simulator should support atleast 4kB of memory. Core 0 will have access to the first 1kB, core 1 will have access to the next 1kB, and so on. You do not have to store the code in the memory. You can directly read the code from the assembly file.
- The simulator should read in an assembly file and execute the same instructions on all cores and in the end display the contents of the registers of each processor, and the memory.
- Features like single step execution, graphical interface or any other feature you can think of, is not compulsory, and is left to your choice.
- Code should be maintained using git and checked into a private repository on github. You should add cs2024iitp as a collaborator to the repository.
- A README file in markdown format should be present in the repository. The README file should contain the following:

- Minutes of the meeting for each meeting held in reverse chronological order.
- The minutes should contain the date and any design decisions made in the meeting.
- The minutes can also contain the tasks assigned to each member and the deadline for the task, and what was accomplished by the member from the previous meeting.
- Example of a meeting minute:

```

Date: 10-Feb-2024
Members: S S Rajamouli, Dr. Sridhar, Prabhas, Alia and
Mogambo
Decisions: Decided to reduce the pitch of Prabhas'
dialogues in the scene following the interval.
Consuming too much energy. Elevate the background
score and add few dialogues for Mogambo. Dr. Sridhar
to work on the specific dialogue that Mogambo will
deliver when he destroys the watch. S S Rajamouli to
review the scene and give feedback by 12-Feb-2024.
Alia, Prabhas and Mogambo to practice the scene and
be ready for the shoot by 08-Feb-2024. Had samosa
for lunch.

```

- Any programming language can be used to develop the simulator.
- Your simulator should be able to run bubble sort on all cores.
- If you are unable to complete the project, do prepare a document detailing what you tried, and what did not work etc. The document along with the incomplete code will be evaluated.
- Any kind of malpractice will fetch you a straight **F**. Malpractice also includes sharing your code with your friends.
- Individual members of a team will be evaluated based on their contribution.
- This is “your” project, “you” might add it in “your” CV.
- **Deadline:** February 14<sup>th</sup> 11:59PM.
- Can you extend the deadline? Yes, definitely. New deadline: February 20<sup>th</sup> 11:59PM.
- Since the deadline has already been extended, there will be no further extensions.
- Note: We will discuss the project in the next class/lab. This is to give you a head start.
- Developed by one of your seniors, MIPSploration.
- Have fun!