Suggestions on ECE 486/586 Project Task Breakdown

Below is a suggestion on the ECE 586/686 project task breakdown to help you and your project team members to divide the work. Please note this is ONLY a suggestion. You can breakdown the tasks in your own way and/or in whichever way that makes sense to you.

Suggested Project Tasks Breakdown:

1. Trace Reader: Reads a memory image trace file which is a text file line by line correctly (with correct content).

2. Convert each line of memory image trace file from Hex format to Binary format, and passes it to the instruction decoder.

3. Instruction Decoder Development: Decode each line of trace file: figure out each line of trace file corresponds to what instruction (interprets instruction type, determines source and destination registers).
    a. There are 2 types of Instruction Formats: R-type format, I-type format
    b. There are 4 categories of instructions: Arithmetic Instructions, Logical Instructions, Memory Access Instructions, Control Flow Instructions.

4. Functional Simulator Development: captures the effect of running the simulated program on the simulated machine state. The machine state includes the following three components: (i) Program counter (PC), (ii) General purpose registers (R1 to R31), (iii) Memory. The initial state of the memory is contained in the memory image. You will assume that the PC and all the general purpose registers are initialized to "0". As you simulate each instruction in the program, you will keep track of the machine state (PC, GPRs and memory) and make necessary changes to the state based on the impact of the simulated instruction. You will continue the simulation until you encounter a "HALT" instruction.
5. Functional Simulator Validation: development testcases to validate.

6. Pipeline Simulator with No Forwarding Development: Develop Timing Simulator Assuming No Pipeline Forwarding Development
7. Pipeline Simulator with No Forwarding Validation: development of testcases to validate the Timing Simulator assuming no pipeline forward

8. Pipeline Simulator with Forwarding Development: Develop Timing Simulator Assuming Pipeline Forwarding Development
9. Pipeline Simulator with Forwarding Validation: development of testcases to validate the Timing Simulator assuming pipeline forwarding

Note: For Tasks 4 – 9 above, all instructions described in the MIPS Lite Spec document file "ece586_project_specs.pdf" posted on Canvas will need to be covered.