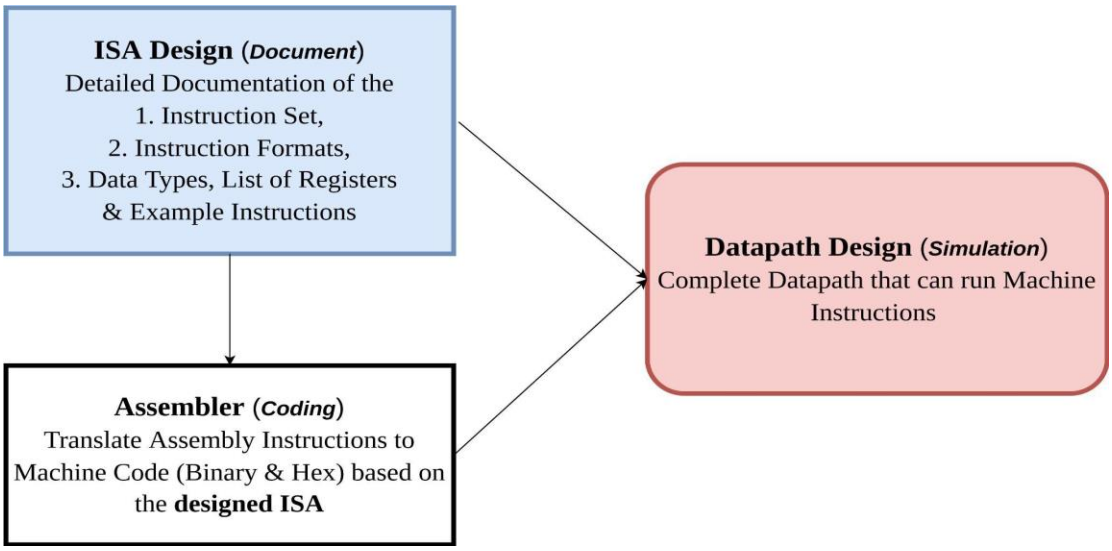
CSE332 Term Project (Section-6)
Department of Electrical and Computer Engineering
North South University- Summer 2022
Instructor: Dr. Mainul Hossain

PROJECT GROUP LINK: [Click Here]

Design a 12-bit Custom RISC-V Microprocessor

Part 1: It starts with a task where students design an ISA on paper. Next comes the assembler, small software capable of translating assembly language into machine format.

Part 2: The small individual components of a computing system such as **Register Files**, **ALU**, **R-type, I-Type**, and **Load-Store** Type Datapaths, etc. will be designed using Logisim.



Part 1.1: ISA Design (Document) Due: Aug 1, 2022

Consider yourself a computer architect and you are going to design a new 12-bit RISC type of CPU. As a designer, your job is to propose a detailed design of the ISA. You need to consider a few things.

1. How many **types** of **instruction** (R-Type, I-Type, J-Type, etc.)?
2. Describe each of the **formats** (fields and field length)
3. How many **operands**? (3 operands, 2 operands)
4. How many **operations**?
5. Types of operations? (Arithmetic, logical, branch type?? How many from each category? List the opcodes and respective binary values)

You must answer those with your reasoning. While you are deciding on the above issues, you might consider some sample high-level programs that can be run on this **CPU using your ISA.** The design might vary from one group to another and there might be multiple possible solutions. You will be scored based on your clear reasoning.

Part 1.2: Assembler (Coding)

It is difficult and error-prone to manually write machine code. The problem can be addressed by writing an assembler, which can automatically **generate a machine code (binary and hex)** from an assembly file. In this project, you should write an assembler for your ISA. The assembler **reads** a program written using assembly language in **a text file**, then translates it **into binary code** and generates an **output file(.txt)** containing **machine code**. The generated output files will later be useful to run a program while you run your instructions in logisim using machine language: **You can use any programming language.**

You should focus on your own ISA that you designed.

I/O format:

The input code will be written in a text file in assembly format following your ISA. There will be one instruction per line. The output will be generated in Hexadecimal format instead of binary. this will be helpful for us to later transfer this code into the RAM block of the logisim circuit.

Documentation: Each group must prepare your own documentation.

Part 2: 12-bit RISC-V Processor (Logisim Simulation) Due: Aug 25, 2022

Here students will be implementing their own CPU based on the 12-bit ISA submitted by individual groups. The CPU must be able to execute the instructions stored in the memory one after another.

1. Complete Single Cycle Datapath Processor based on **12-bit ISA -** Students will be designing different parts such as **ALU, Register Files, and connect them into a Single logisim file.**
2. Implement datapath and Control Unit – With the Main Control Unit, students can execute instructions automatically based on the Opcode

Testing the Processor

This information explains to you how I am going to test your **12-bit processo**r once you have finished building it completely and submit. So before submission, you can test the design yourself using the same method.

Step 1. Start Assembler and run the **input.txt** containing example Assembly Codes and generate the machine code **output.txt.** The machine code must be generated in **Hex format**. The values in the text file must be written in the text file in the following format:

```
v2.0 raw                    // must write on top
a9 b2 27 14
29 cb 10 11
```

Step 2. Start Logisim and load your **12-bit RISC processor**. Now, load the program (**output.txt**) into the **instruction memory** of your processor. The file containing instructions in Hexa format will update the instruction memory of your own datapath. Now start Executing all the Instructions one by one and verify the results in your **register file** and **Data memory.**

HOW TO SUBMIT THE PROJECT REPORT

1.Github: upload all documentation in Github and provide the link in the excel file.

2. YouTube:

- Record a video of the project presentation and upload in YouTube.
- In the Google sheet, include video link
- Clearly explain the Logisim circuit and demonstrate an operation.
- In the video, each student in the group should introduce himself/herself and present a part of the project.
- During presenting, the camera should be on, so that each student`s face is visible.

In case of any help needed, you may contact the lab instructor and the TA.