

# Nova Documentation

## Creators

### *Johan Nguyen*

- Co-Designer (main, RegisterFile, and ALU circuits)
- Assembler.py Constructor
- Documentation Creator (Encodings)

### *Michael Preziosi*

- Co-Designer (main, DataControl, and Decode circuits)
- Documentation Creator (Assembler Usage, Registers/Functions)

## How to Use the Assembler with Nova

- 1.) Open the “Nova” folder
- 2.) Write assembly code in the “Assembly.txt” file
  - “Assembly.txt” initially contains a sample of a properly formatted assembly file; replicate this format when writing your assembly code
- 3.) Run the “Assembler.py” file
  - “Instructions.txt” will contain the output for the instructions written in “Assembly.txt”
  - “Data.txt” will contain the output for the data written in “Assembly.txt”
- 4.) In “Nova.circ”, right-click on either the instruction memory (“InstructionMem”) or the data memory (“DataMem”) and select “Load Image”
- 5.) Select “Instructions.txt” for the instruction memory or “Data.txt” for the data memory to load the assembler outputs to their corresponding memory in Logisim
  - Select “v3.0 hex” for the file format of each image, then click “OK”
- 6.) Still in “Nova.circ”, click on “Auto-Tick Enabled” under the “Simulate” dropdown menu and start the simulation
  - The values in each of the registers can be seen in the displays to the far right of the architecture

## Architecture

### *Registers*

Nova has four registers for general purpose, which can be referred to as X0, X1, X2, and X3.

*I pledge my honor that I have abided by the Stevens Honor System.*

## Functions

Nova can perform the following functions:

- ADD Function
  - Assembly Syntax: ADD Rd, Rn, Rm
  - Used to add the values in registers Rn and Rm, storing the sum in register Rd
    - Example: ADD X1, X2, X3 ( $X1 = X2 + X3$ )
    - Values represented by 8 bits
- SUB Function
  - Assembly Syntax: SUB Rd, Rn, Rm
  - Used to subtract the value in register Rm from the value in register Rn, storing the difference in register Rd
    - Example: SUB X1, X2, X3 ( $X1 = X2 - X3$ )
    - Values represented by 8 bits
- LDR Function
  - Assembly Syntax: LDR Xt, Xn
  - Loads the data at address Xn in memory to the target register Xt
    - Example: LDR X1, X2 ( $X1 = \text{Mem}(X2)$ )
- STR Function
  - Assembly Syntax: STR Xt, Xn
  - Stores the data in register Xt to the memory address Xn
    - Example: STR X1, X2 ( $\text{Mem}(X2) = X1$ )

## Encodings

Instructions	Opcode(7-6)	Rm(5-4)	Rn(3-2)	Rd(1-0)
ADD Rd, Rn, Rm	00	2nd Register	1st Register	Destination Register
SUB Rd, Rn, Rm	01	2nd Register	1st Register	Destination Register
LDR Xt, Xn	10	Not used	Address Register	Target Register
STR Xt, Xn	11	Source Register	Address Register	Not used