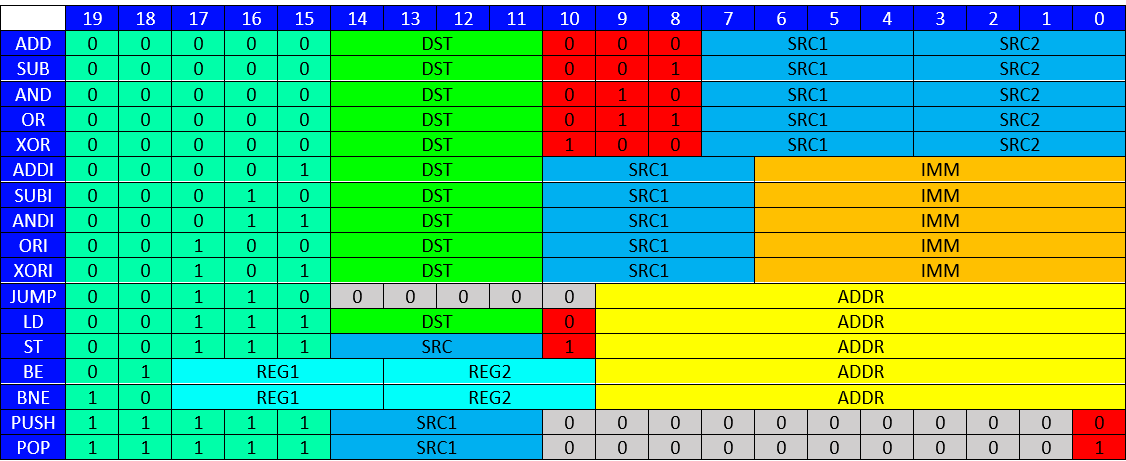
**Assembly Language:** Firstly, we designed an instruction set architecture (ISA) that is suitable for the instruction set given to us.



We have reserved 5 bits for the opcode in our table. Thus, our work would be much easier while making the assembler. Then we reserved 4 bits for registers such as DST, SRC, REG. We reserved a 10-bit field for ADDR as requested. For the IMM register, we reserved a 7-bit area, which is the maximum we can allocate.

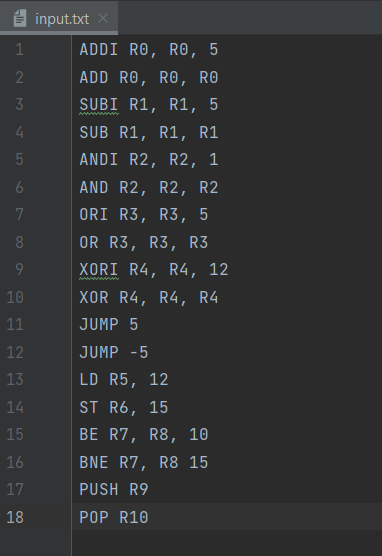
We have given the same opcode for the ADD, SUB, AND, OR and XOR instructions. Because these instructions have a 3-bit empty field, we could easily find which instruction it is by looking at this field. We gave the same opcode for the LD and ST instructions because they had 1-bit empty field so that we could more easily find which instruction it was.

After the Instruction set architecture was finished, we started making Assembler.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

We used java to write the assembly codes. The purpose of assembly code is to print the data in the input file in hexadecimal form. This output file will be our input in the Logisim design.

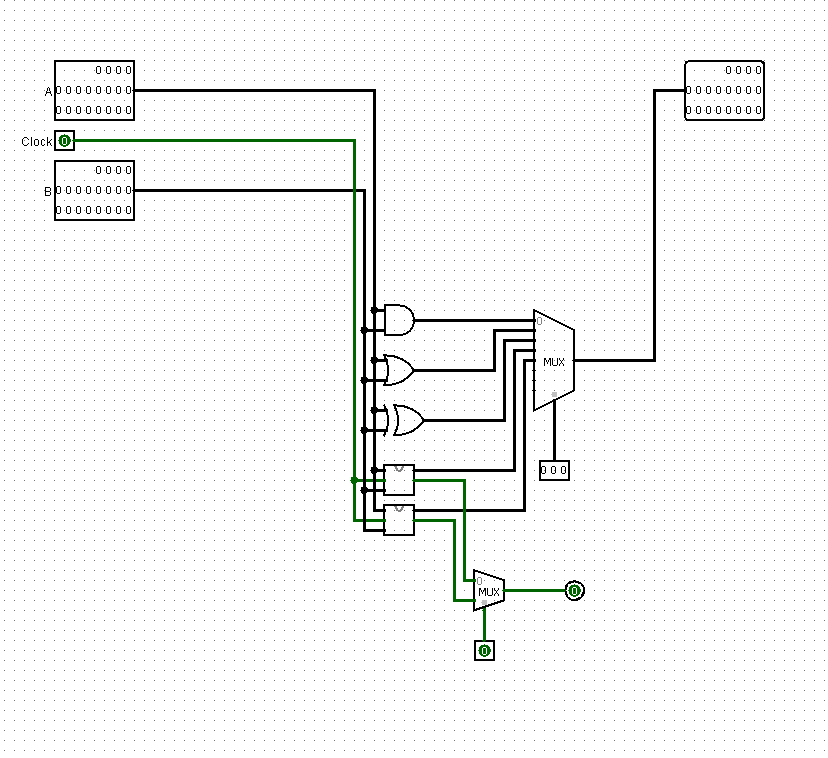
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Logisim Component Design:**

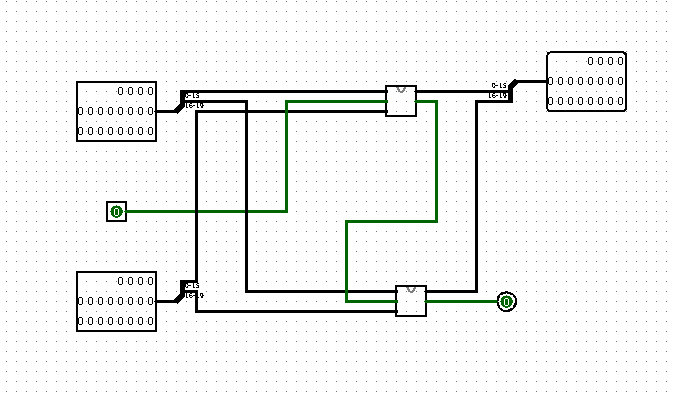
**ALU:**

Arithmetic logic unit (ALU) compute ADD, SUB, AND, OR, XOR, ADDI, SUBI, ANDI, ORI and XORI.



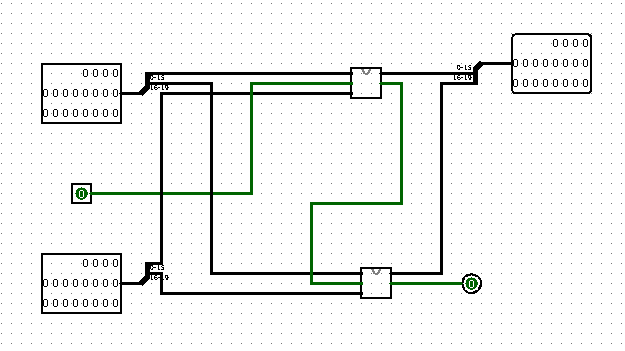
**20-Bit Adder:**

The 20-Bit adder adds the 2 given 20-Bit values ​​and 1 bit carry value.



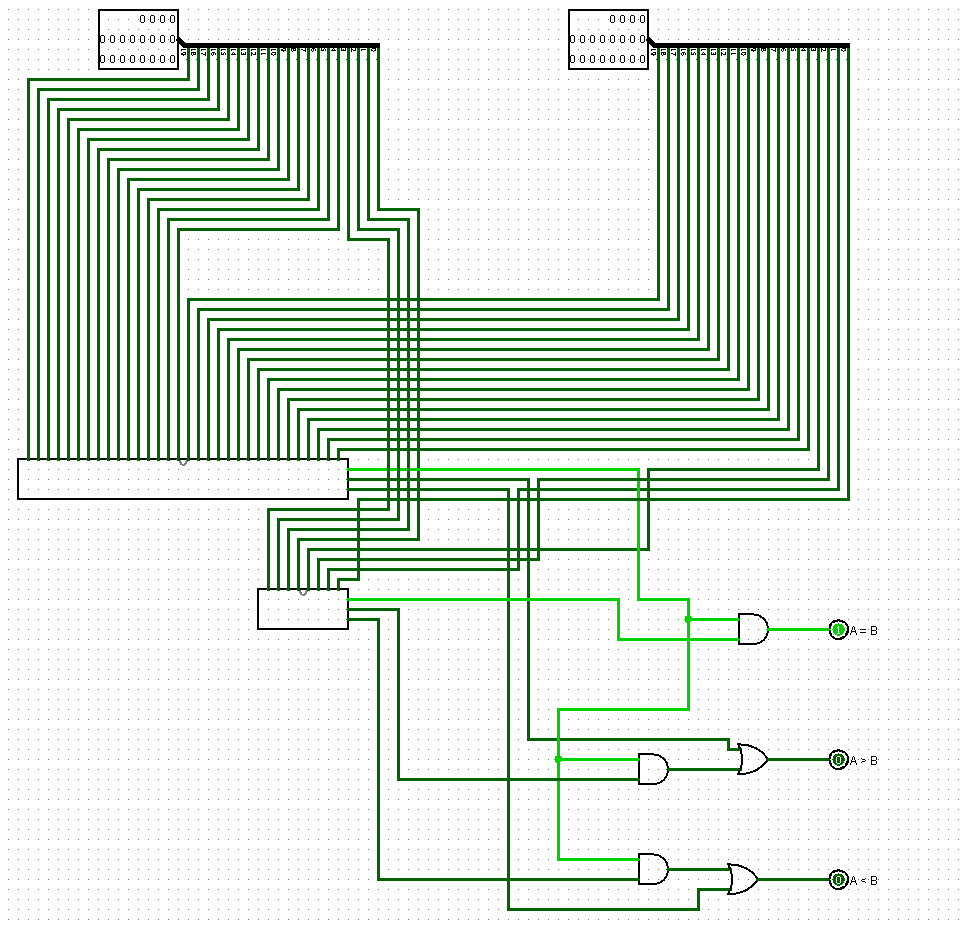
**20-Bit Subtractor:**

The 20-Bit subtractor subtracts the 2 given 20-Bit values and 1 bit carry value.



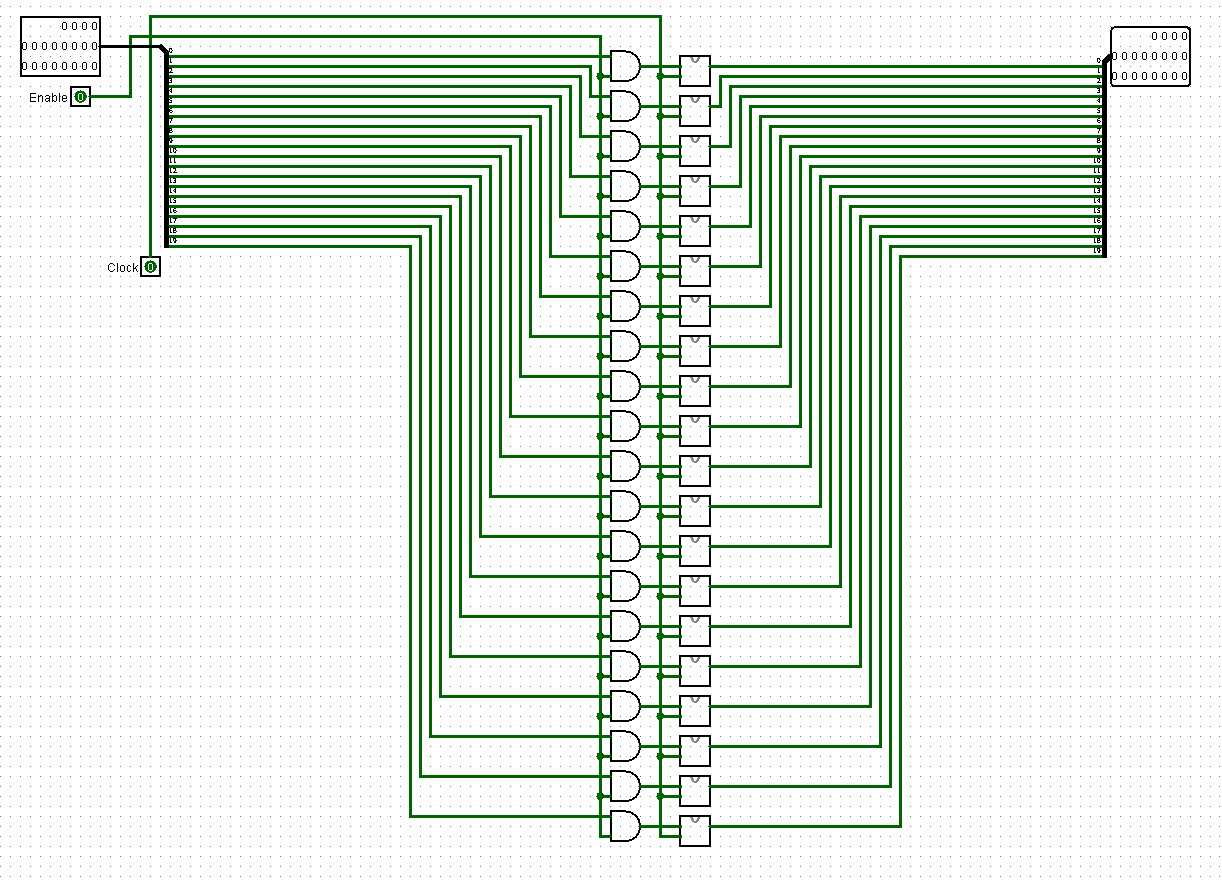
**20-Bit Comparator:**

The 20-Bit comparator compares 2 20-Bit values ​​with each other and shows whether these 2 values ​​are equal to each other.



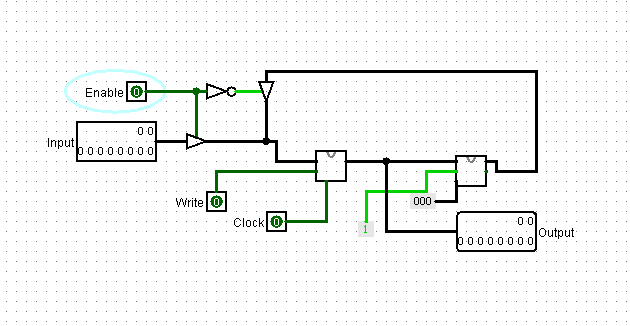
**20-Bit Register:**

The 20-Bit register is the data storage area of ​​the CPU.



**Program Counter:**

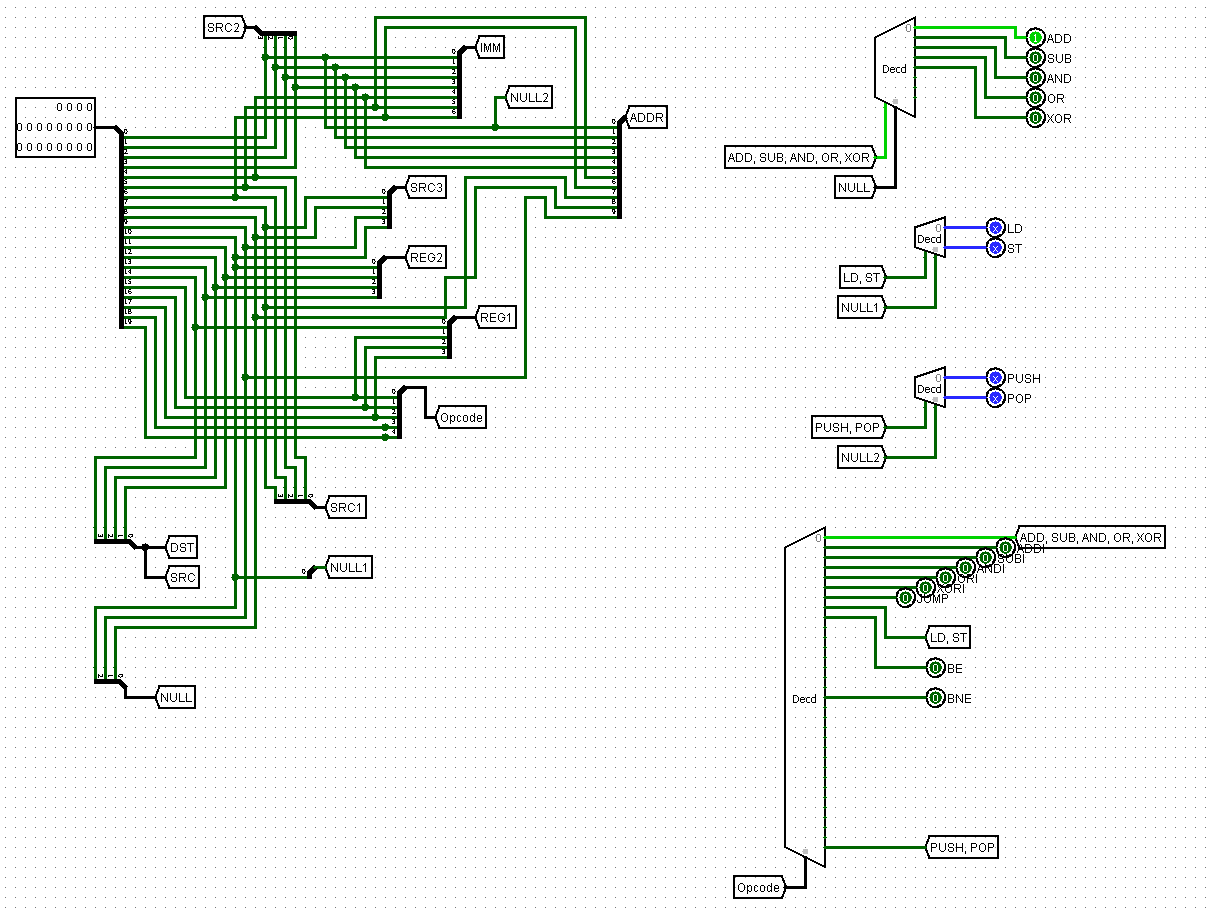
The program counter holds the address of the next instruction to be read inside the CPU. The program counter automatically increments after each instruction.



**Logisim Design with Control Unit:**

**Control Unit:**

The control unit is the circuit that manages the operations and timing of the CPU.



Emir Ege Eren 150119739

Burak Dursun 150119743

Haydar Taha Tunç 150119745

Mehmet Emre Şengüler 150119781