Introduction

The product modeled using the Logisim 2.7.1 tool consists of an 8-bit Pico processor and a 32 x 8 RAM memory. It includes the following components: "Program Counter", "Memory Address Register", "Memory Buffer Register", Instruction Register, the "Accumulator", and the "Arithmetic Logic Unit". Along with these, there are address and data buses, which enable communication between the CPU and the memory. The processor follows a basic set of instructions that include memory read and write operations, logical operations such as negation, arithmetic operations like addition, and system detection. Other instructions in the instruction set include "LOAD", "STORE", "ADD", "INC", "NOT", and "HLT". No conditional jump instructions were made. It operates with 8-bit words, as instructions and data are processed in 8-bit blocks. Communication within the CPU occurs through the two main buses: the "Address Bus" and the "Data Bus". The "Address Bus" carries the memory address, and the Data Bus transfers data between the CPU and the RAM. It is worth noting that the system mimics the Von Neumann architecture, where instructions and data are stored in a single memory that allows the immediate execution of instructions stored sequentially.

Operation of the System

Initiation of the system

LOAD *instruction example*

Como se muestra en la Figura 1, para iniciar el sistema se requieren realizar los siguientes pasos:

1. Activate the Enable and CLOCK.
2. Press PINS:
   i. MANUAL THICK
   ii. PROGRAM COUNTER
   iii. MEMORY ADDRESS REGISTER

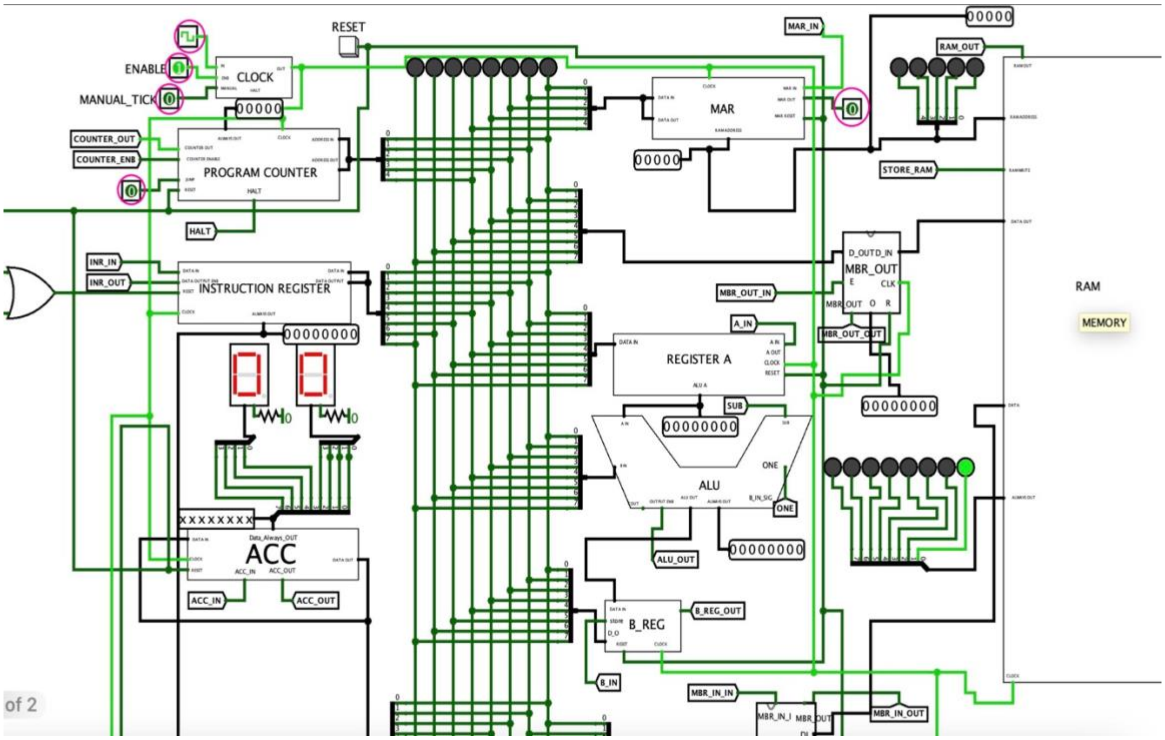


Figure 1: Schematics of step #1 and #2 in Logisim.

3. STORE in RAM:
   i. In ADDRESS 0: The value 01 = (LOAD 00001).
   ii. In ADDRESS 1: The value 03 = (LOAD value 3 in ACC)

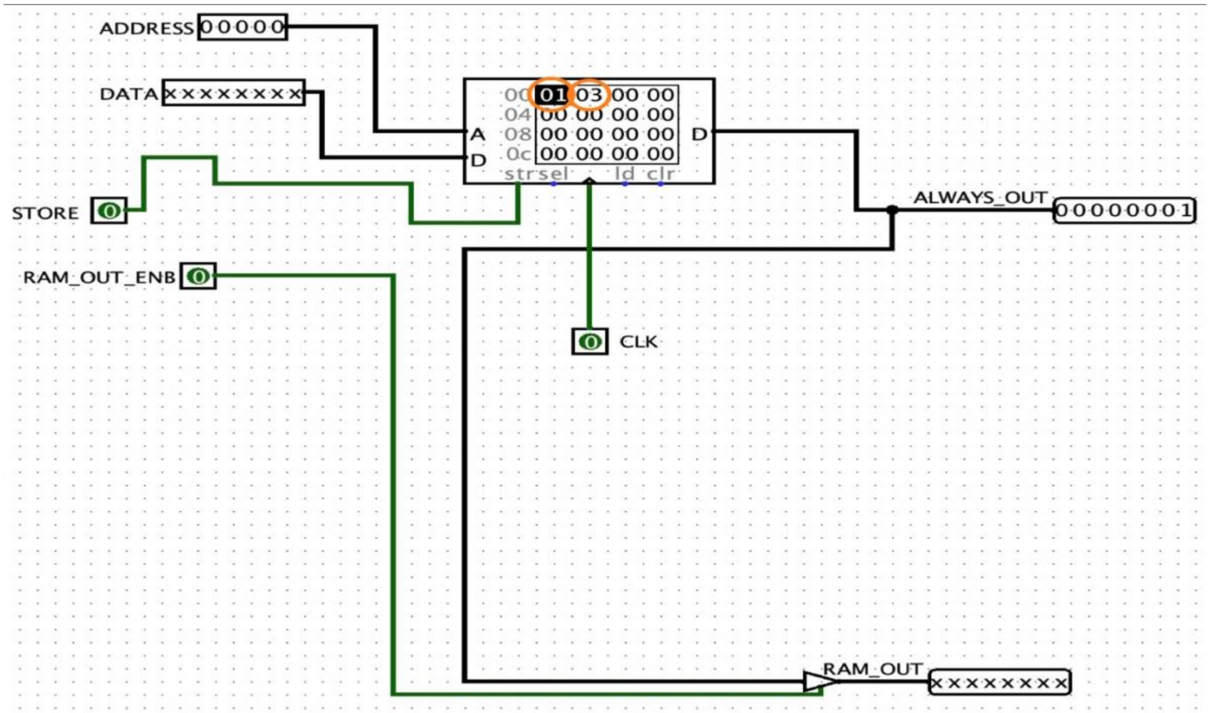


Figure 2: Schematics of step #3 in Logisim

4. Simulate the circuit.

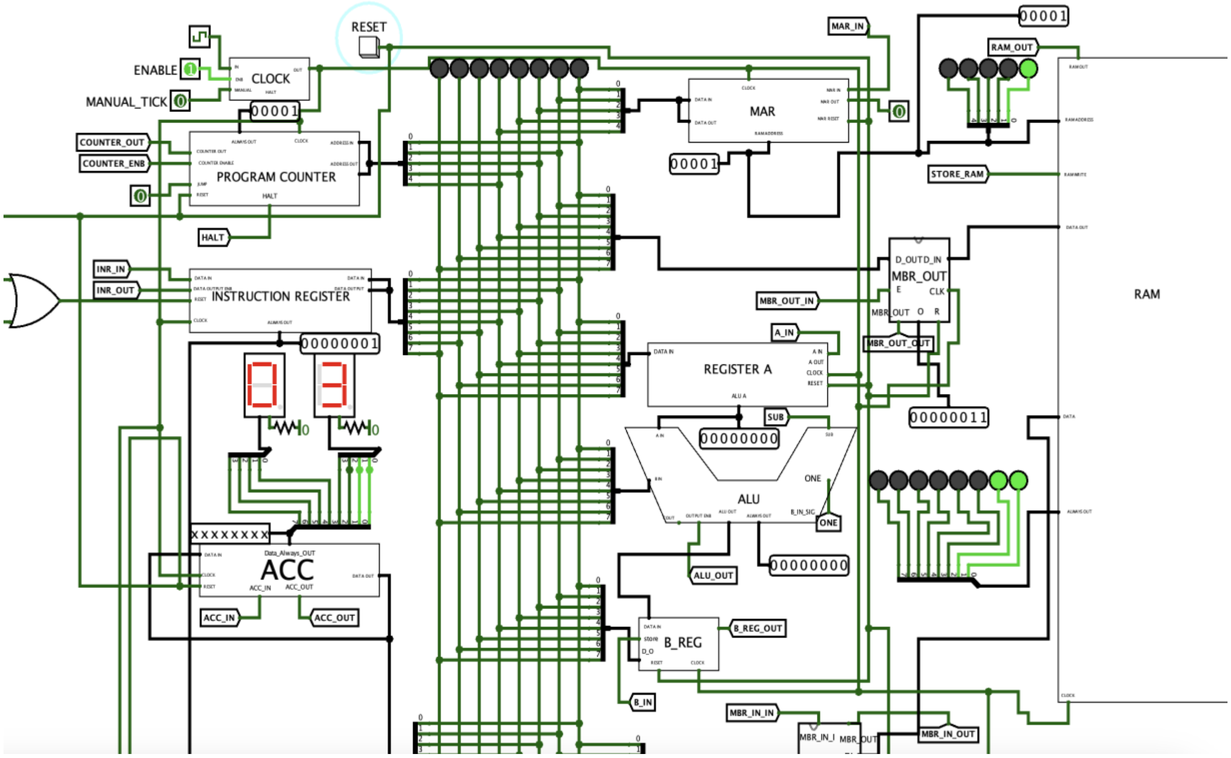
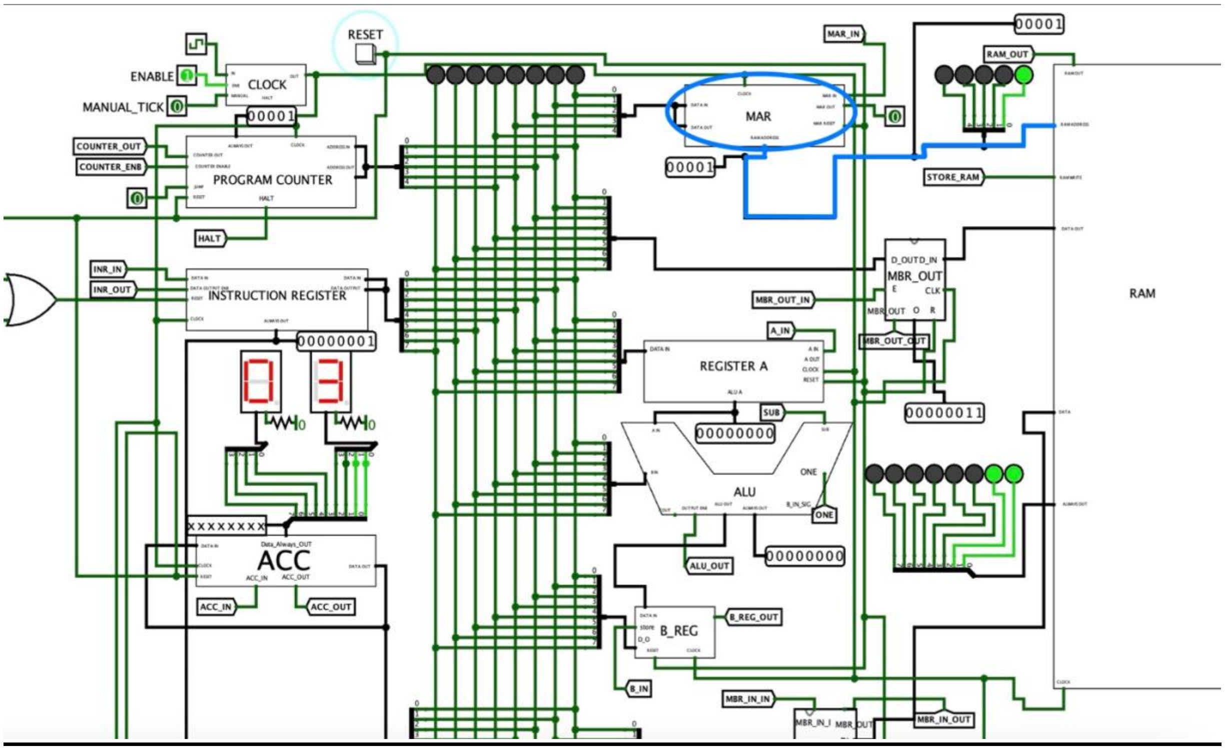


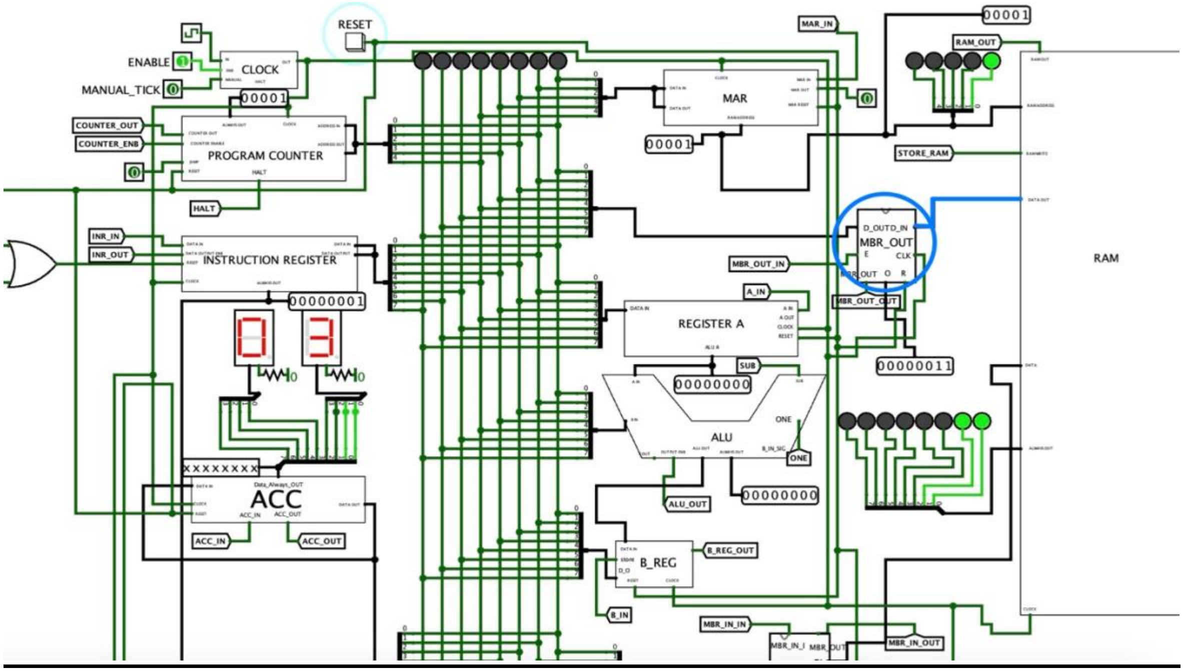
Figure 3: Simulation schematics for the CPU.

Simulation of the System

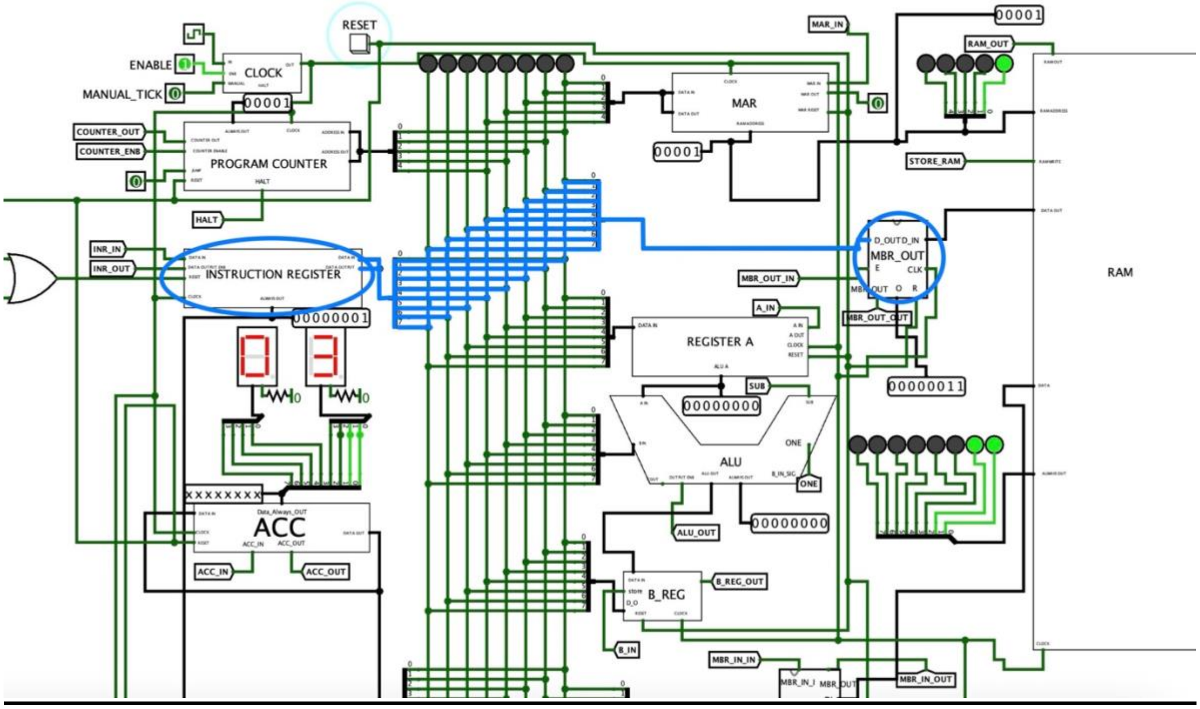
1. In the RAM, the address 0 contains the value 01.
2. On the first CLOCK pulse, the PC sends ADDRESS 0 to the MAR using the control signals COUNTER_OUT and MAR_IN.



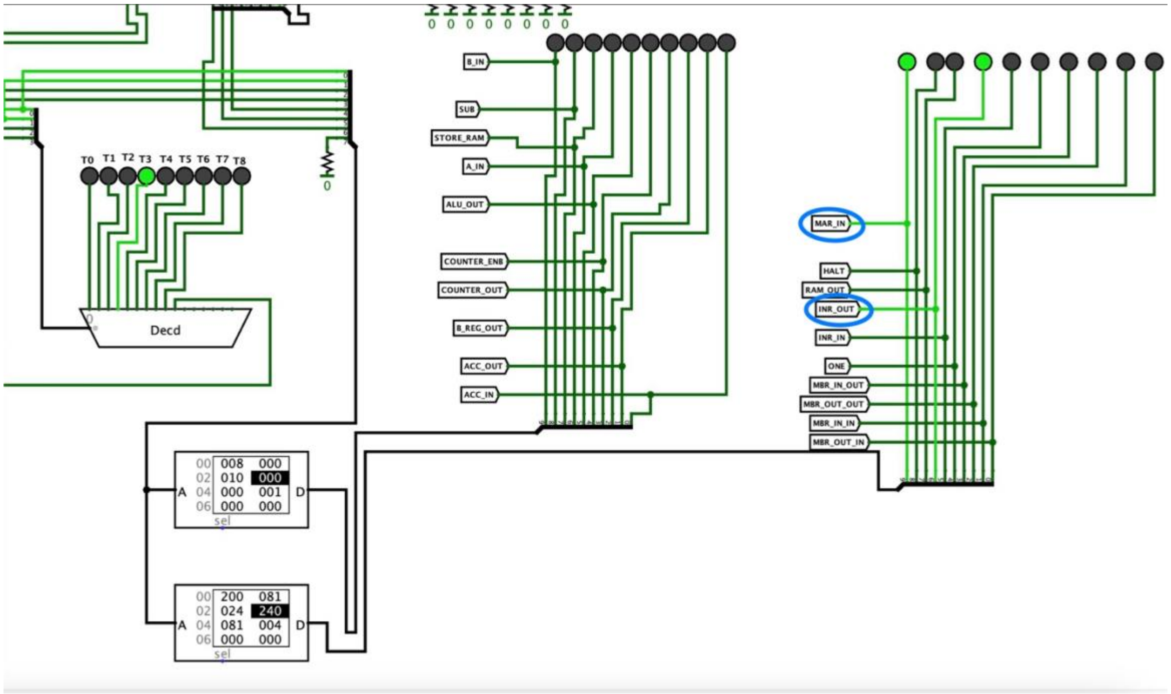
3. In the second CLOCK pulse, the RAM sends the contents of memory location 0 to the MBR using the control signals RAM_OUT and MBR_IN.



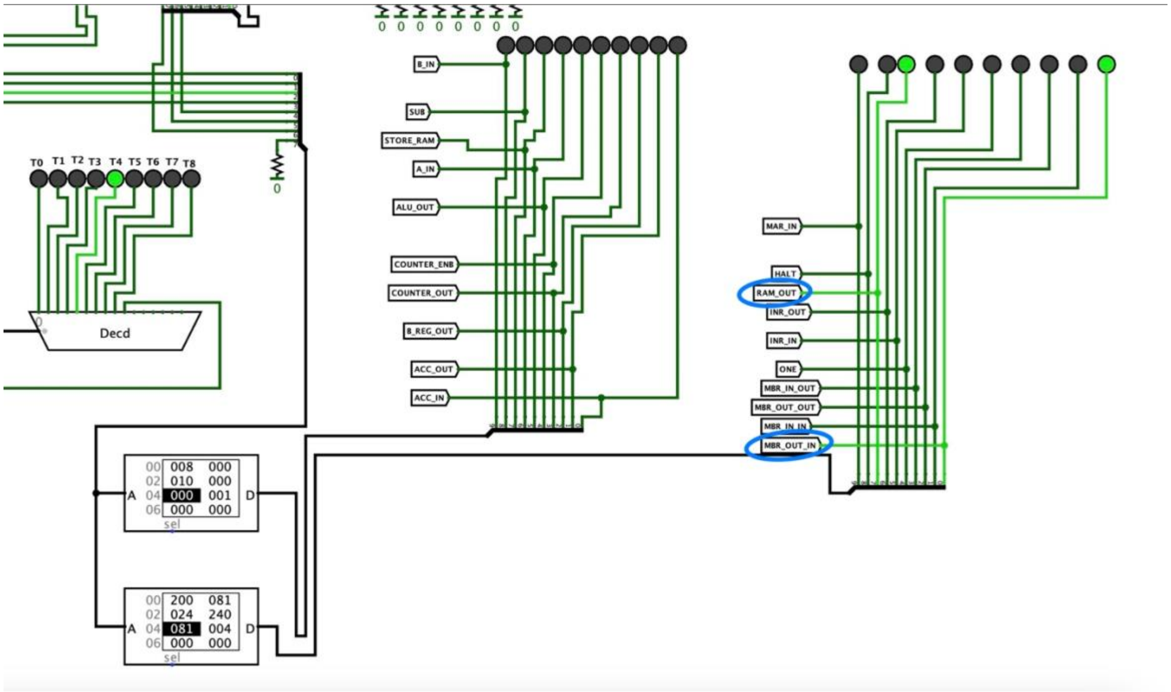
4. In the third CLOCK pulse, the MBR sends the content to the BUS, which enters the IR, and then the PC increments by one.



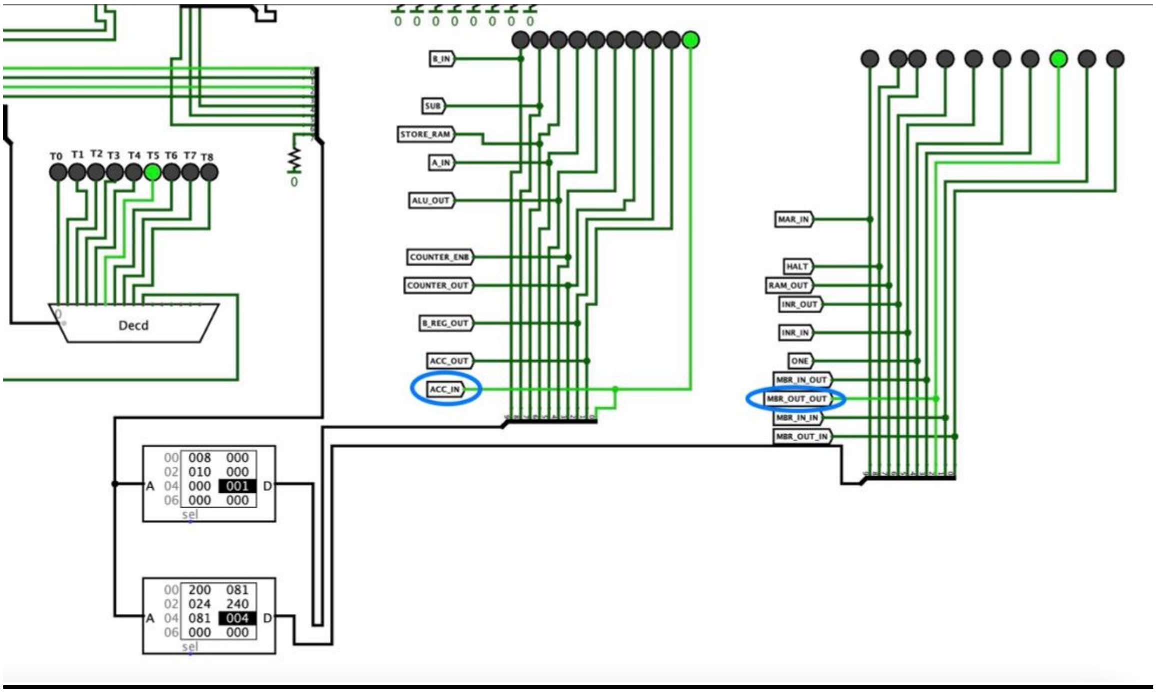
5. The instruction is decoded by the CU.
6. The CU activates the INR_OUT and MAR_IN signals. (The 5 least significant bits end up in the MAR).



7. Then, the CU activates the RAM_OUT and MBR_IN signals. (The content that was inside the ADDRESS goes to the MBR).



8. Finally, the CU activates the MBR_OUT signal and ACC_IN. (The content that was in the register enters the ACC, loading the indicated value).



```
00 008 000
02 010 000
04 000 001
06 000 000
```

```
00 200 081
02 024 240
04 081 004
06 000 000
```