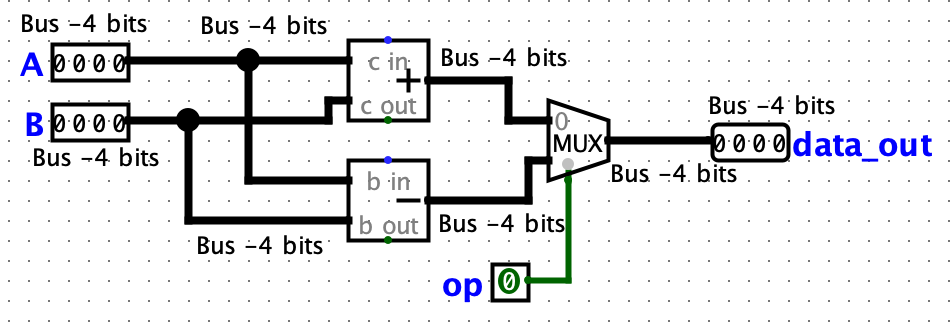
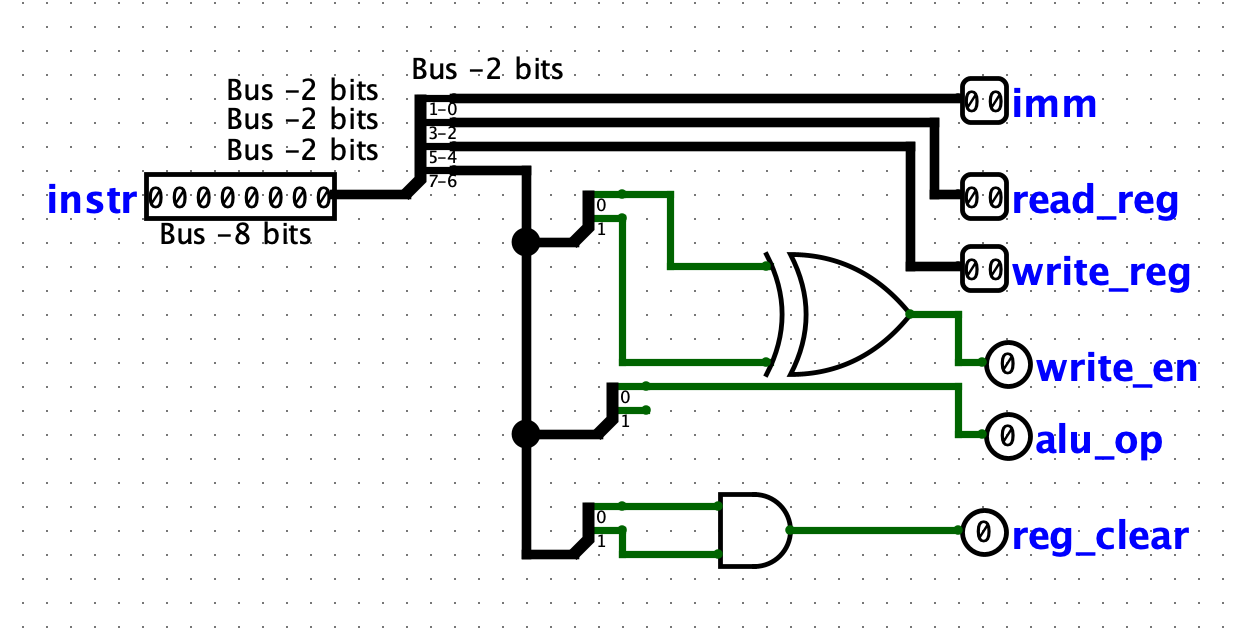
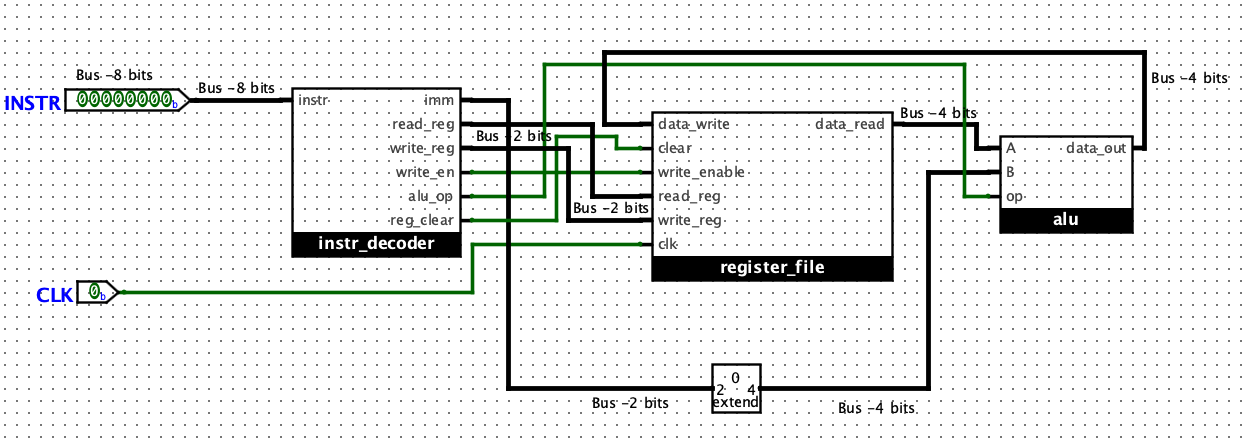
**Lab 7:**

**ALU Diagram:**

**Instruction Decoder:**

****

**High-level Logisim Circuit Diagram:**

**Example of Working Processor:**

**The circuit is reset, all registers have value 0000.**

**Instruction Code: 10100101**

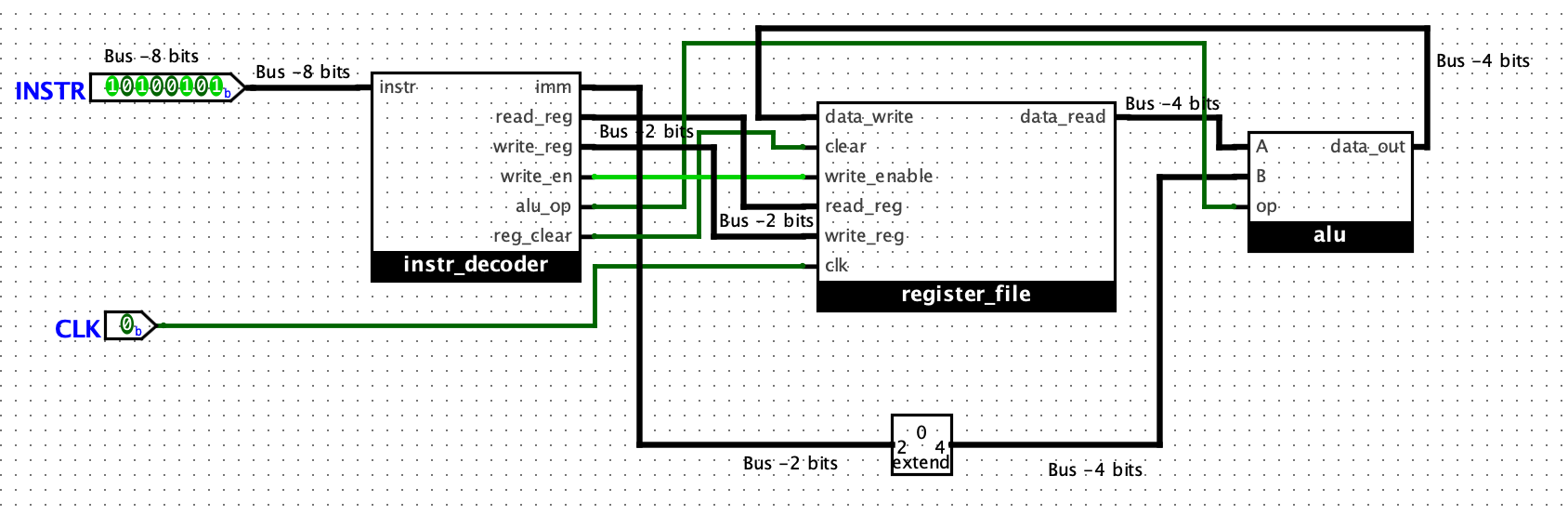
write\_reg = register 2

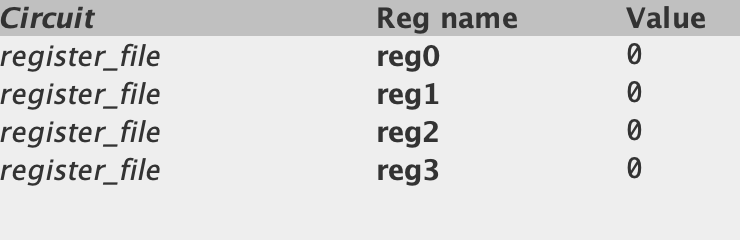
read\_reg = register 1

imm = 01

op\_code = 10 (operation A – Addition)

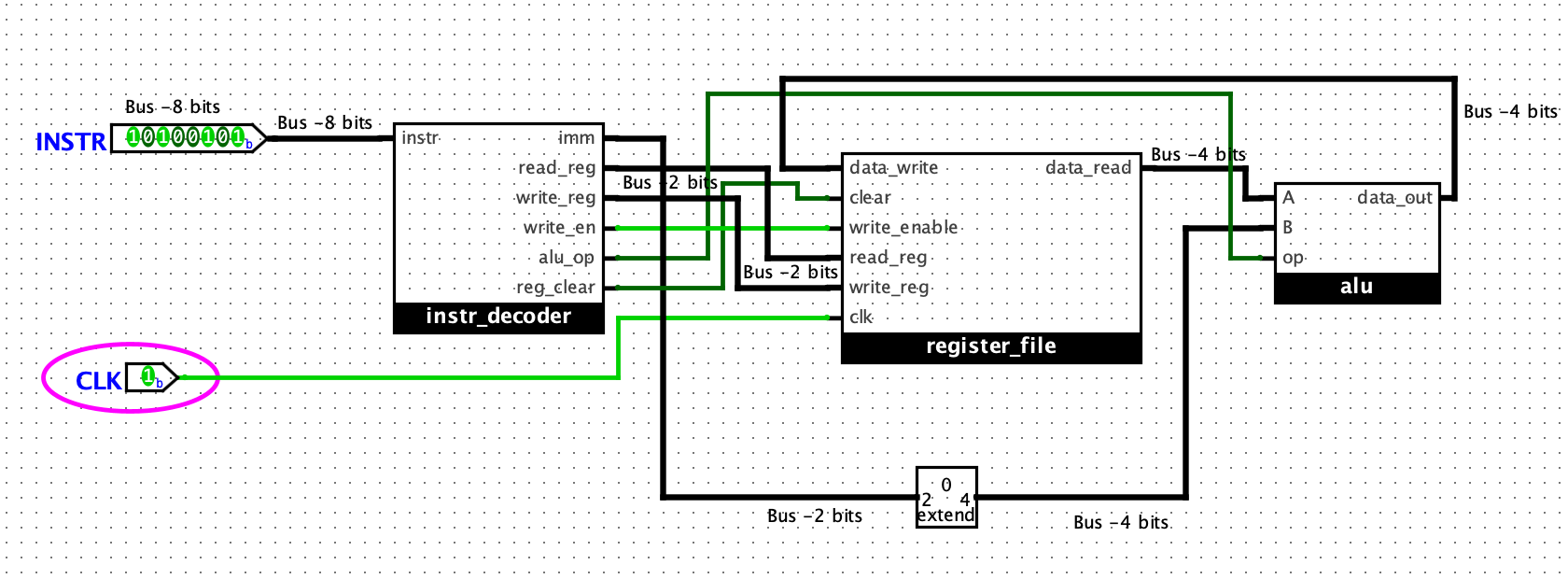
🡪 At first the instruction code 10100101 goes through the instr\_decoder and it is dissected. The instruction decoder outputs values: imm = 01, read\_reg = 01, write\_reg = 10, alu\_op = 1, reg\_clear = 0, write\_en = 1 (an operation will be taking place). This information is then sent to the components that require it to function. The write\_enable tells the register file that a value can be written to a register on a positive clock edge. The value imm is sent through a bit-extender (from 2 to 4 bits) so that it can be utilized in the ALU, where it is the value for the B-input (0001). Data is extracted from the read\_reg, that value is outputted through data\_read and sent to the ALU A-input (0000). The value of the alu\_op is used in the ALU to determine which operation is used; in this case alu\_op is 0, so the value from Operation A (addition, 0000 + 0001 = 0001) will be outputted through data\_out and sent to data\_write to be written to register 2. The value is only written to register 2 when the there is a positive edge (clock goes from 0 to 1).

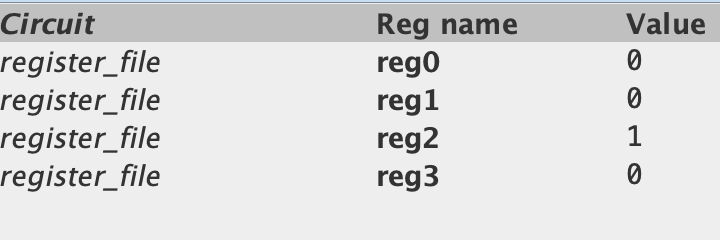
**When Clock is 0:**

****

**Value of register 2 is still 0.**

**When clock turns to 1:**

****



**Hence, we can see that the value of register 2 has changed accordingly.**