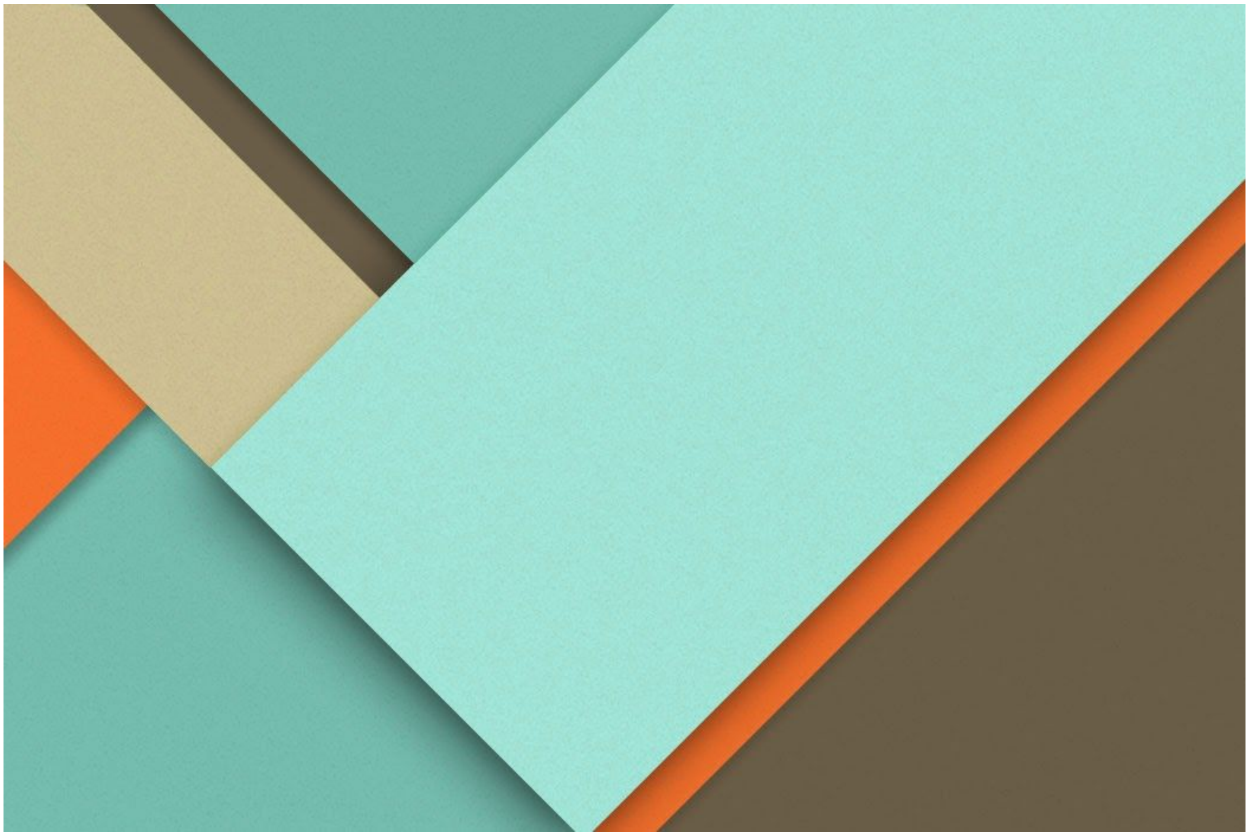


Computer Architecture Innovative Assignment II

Instruction Cycle

Arshit Babariya(18bce018)

Aditi Bhatt(18bce004)

CONTENTS

1 . Specifications and Processes

- ❖ Processes
    1. Fetch data from memory
    2. Decode instruction
    3. Execution of instruction
- ❖ Specifications
    - ➢ 16-bit instructions, 8-bit bus.
    - ➢ Instruction Format

— First 8 bits are

2 bits opcode

2 bits destination register

2 bits source register 1 (operand 1)

2 bits source register 2 (operand 2)

— Last 8 bits :

Show the value to be loaded into the register if loading occurs.

— Opcode:

1. 00 XXXXXX - Load register
2. 1X XXXXXX - ALU OPERATION

ALU SUB	ALU FUNCTION	OPERATION
1	00	SUBTRACTION
0	00	ADDITION
0	01	MULTIPLICATION
0	10	AND
0	11	0R

1. FETCHING:

The current program counter is used to index into memory.

An instruction is fetched, and stored in the Instruction Register.

At the same time, the next instruction address (current program counter plus one) is stored into the PC.

2. DECODING & EXECUTION:

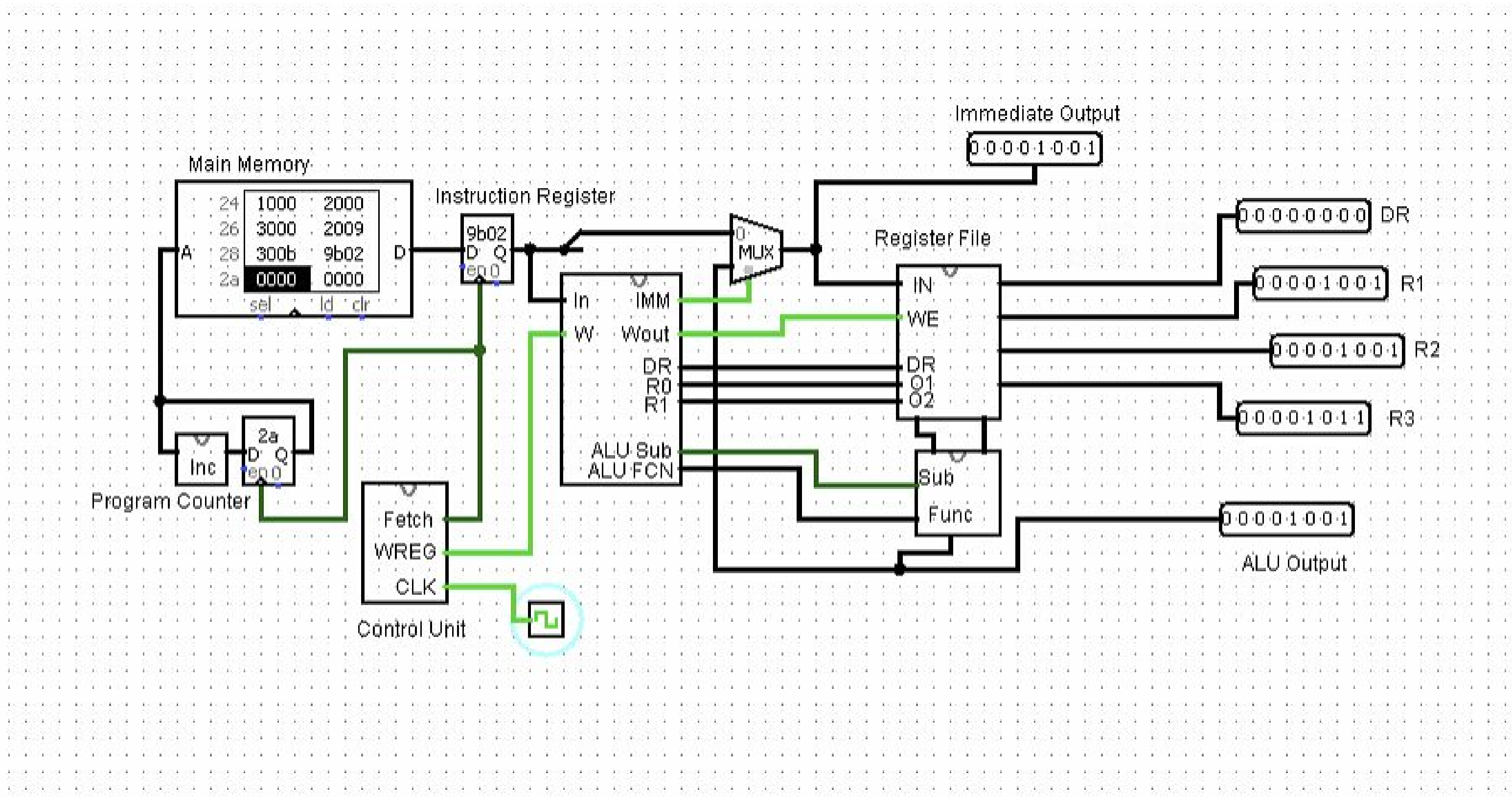
The decoder extracts the bits from an instruction, and the ALU does the computation.

3. STORE DATA:

Enable write for the register file, storing either an immediate value, or the result computed by the ALU.

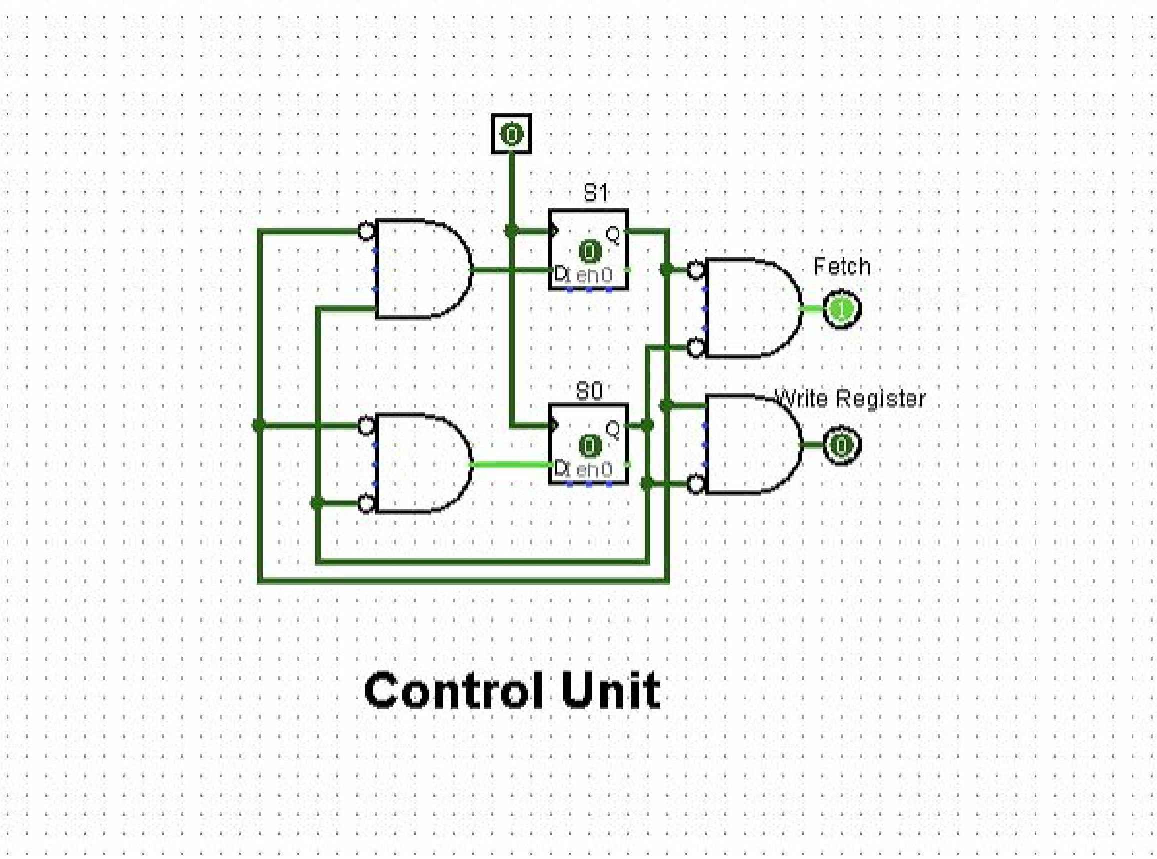
2. Working of the circuit with examples

1. MAIN CIRCUIT:



- The main circuit shows all the connected components together.

2. CONTROL UNIT:

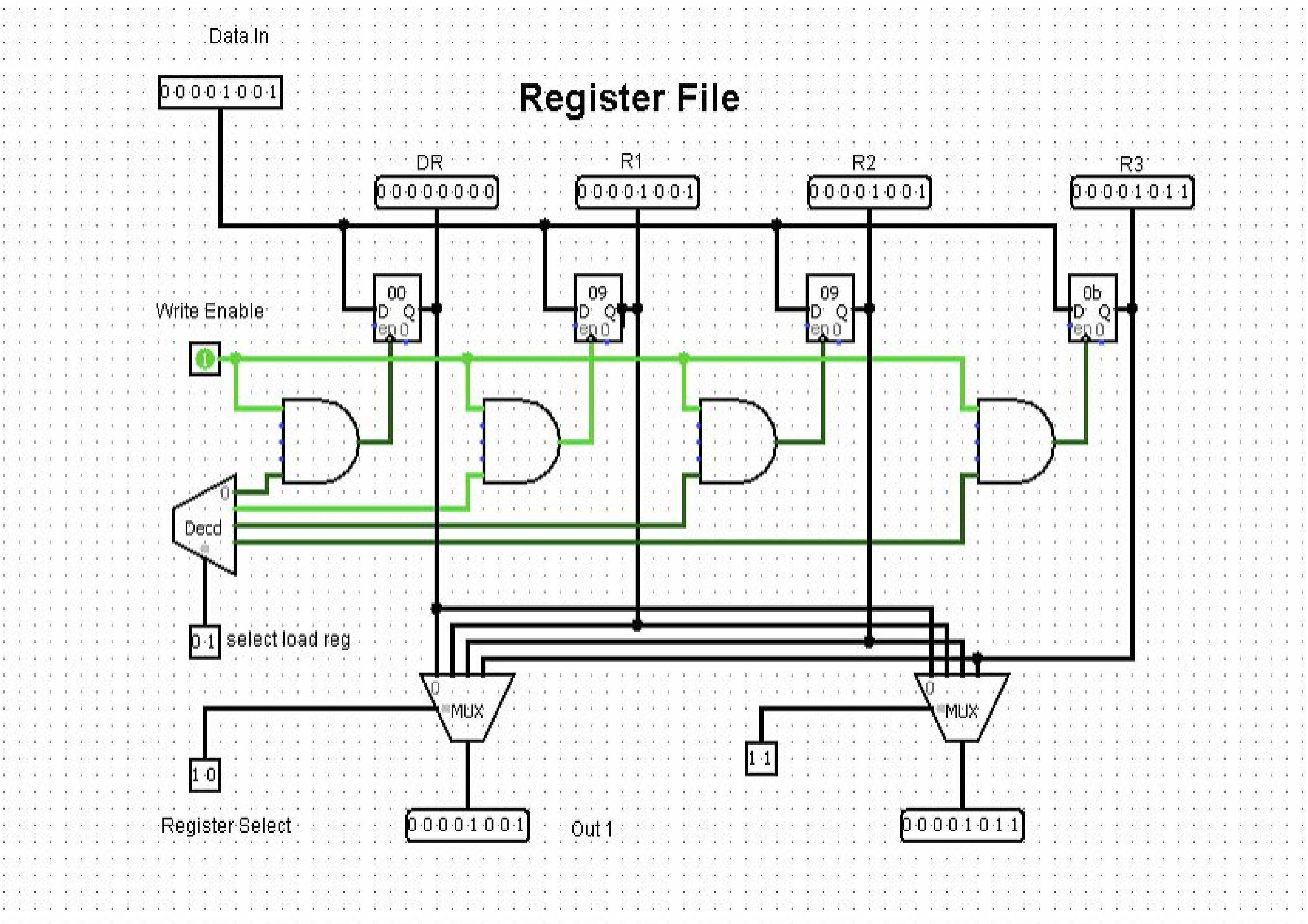


- The control unit works using the clock.

  It controls when the fetching and writing starts and ends.

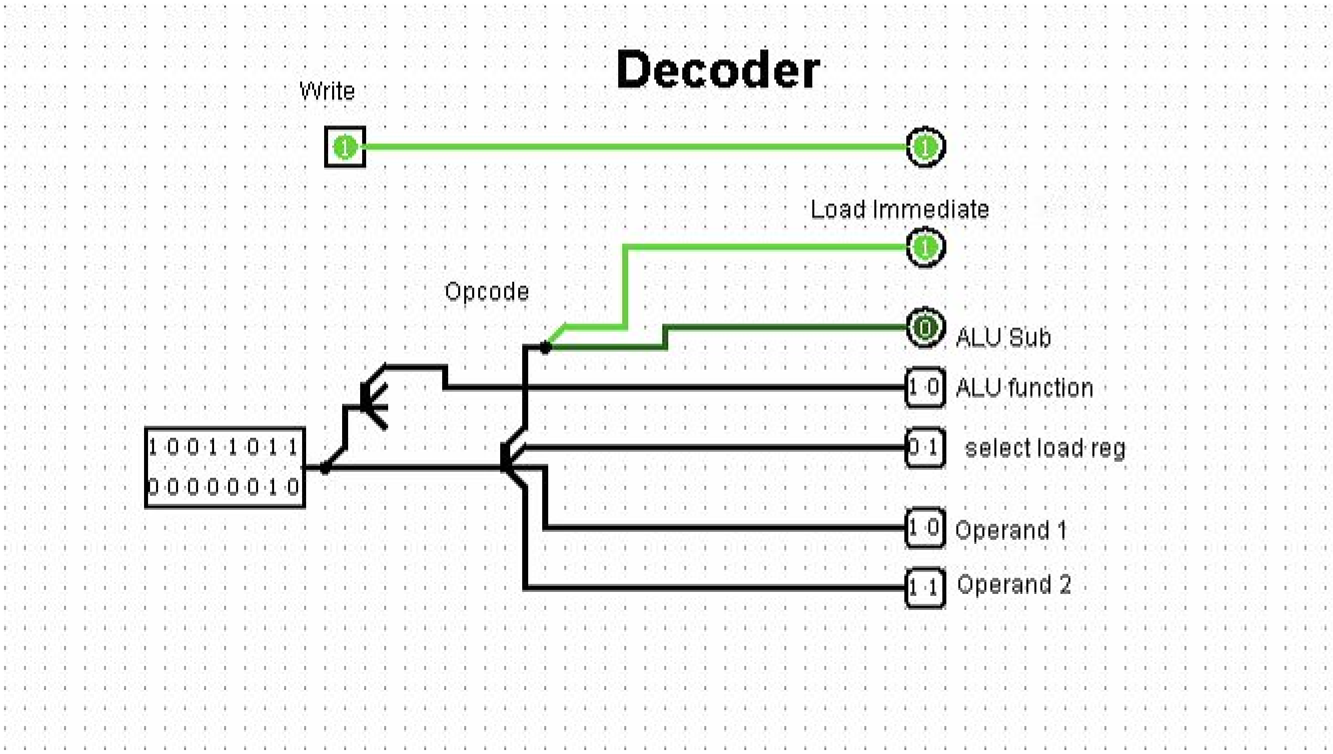
  It is implemented using two D flip flops and 4 AND gates.

3. REGISTER FILE:



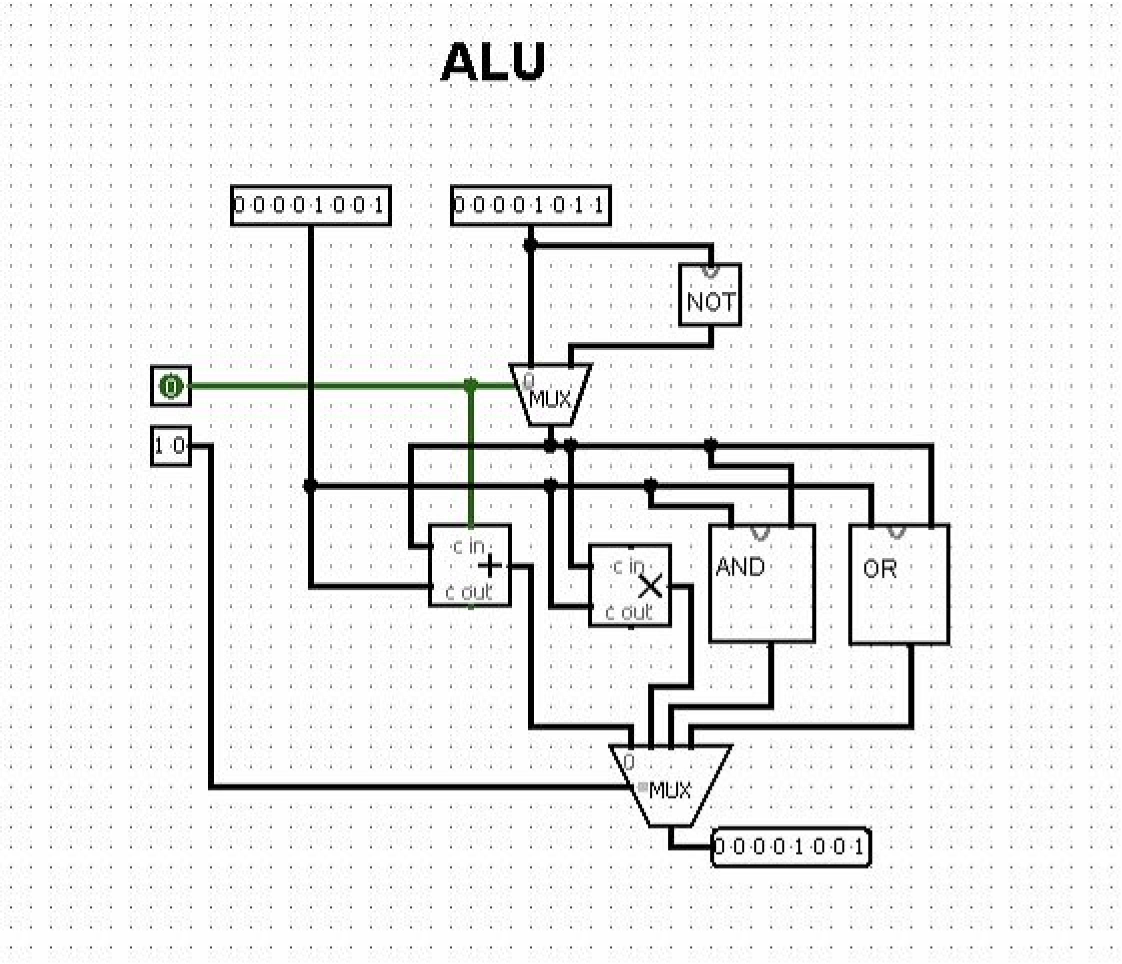
- The register files shows 4 registers - DR,R1,R2,R3,into which data is stored ,and from which data is retrieved according to selected opcodes.

4. DECODER:



- The Decoder decodes the 16 bits of the instruction and selects registers, and operations accordingly.

5. ALU:



The ALU made performs 5 operations : Addition, Subtraction, Multiplication,AND,OR.

EXAMPLES:

1. Load register 2 with value 9

   00 10 00 00: (Opcode, R2, source registers ignored)

   00001001 : (9 in binary) Convert to hex:

   opcode = 2009

2. Load register 3 with value 11

   00 11 00 00: (Opcode, R3, source registers ignored)

   00001011 : (11 in binary) Convert to hex:

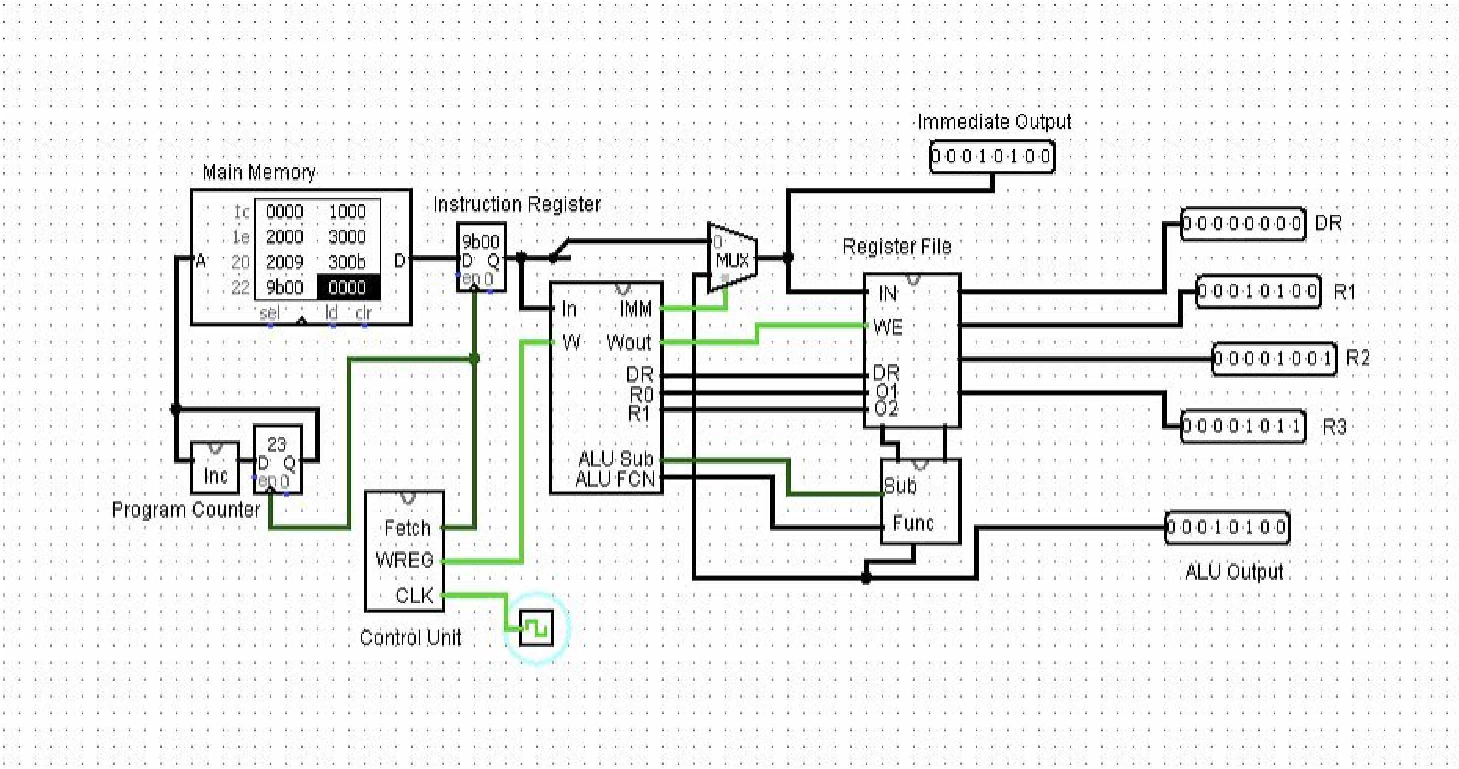
   opcode = 300b

3. Add R2 to R3, result to R1

   10 01 10 11: (Opcode, R1, R2, R3)

   00000000: (Immediate is ignored)

   Convert to hex: 9b00



1c	0000	1000
1e	2000	3000
20	2009	300b
22	9b00	0000

4. Multiply r2 and r3, result to DR

   10 00 10 11: (Opcode, DR, R2, R3)

   00000001: (Immediate is ignored)

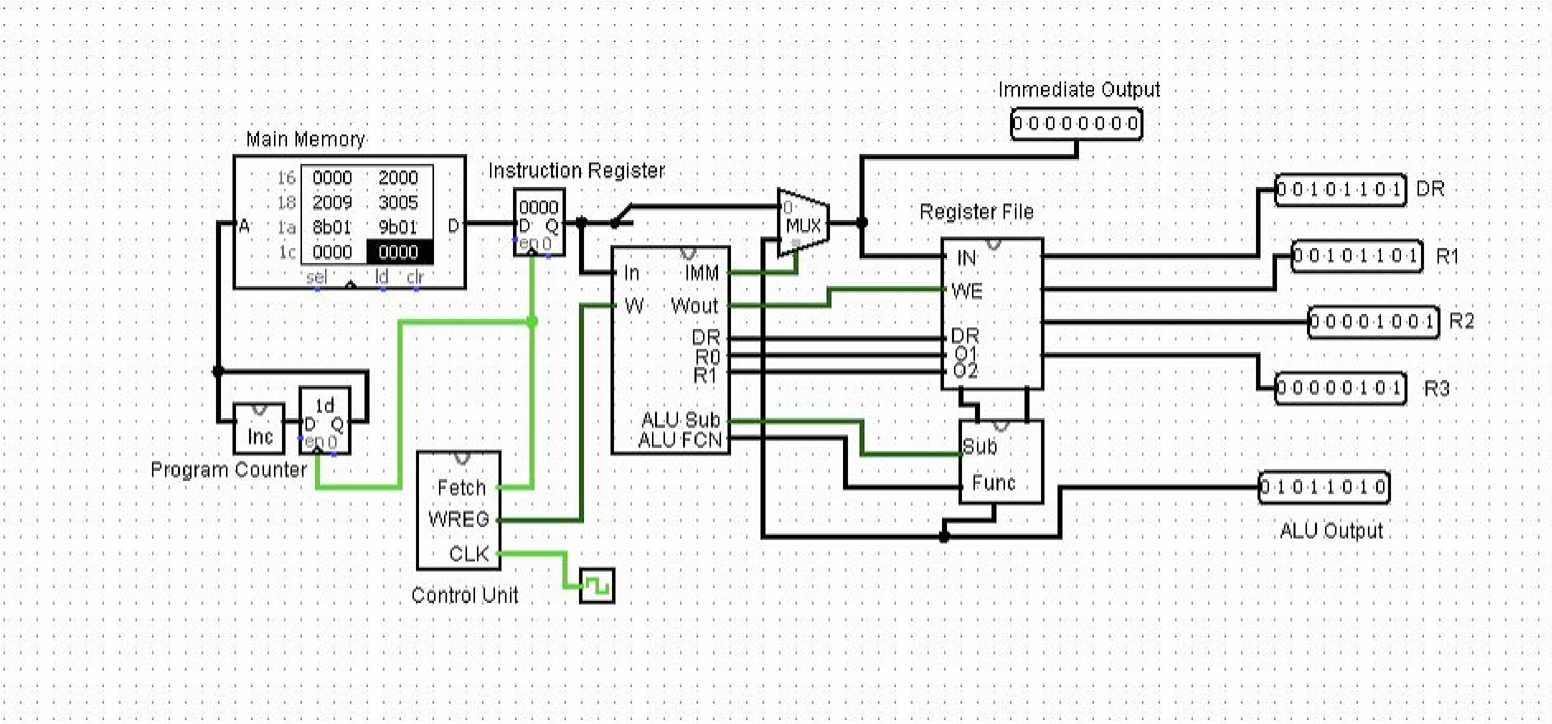
   Convert to hex: 8b01

5. Multiply r2 and r3, result to r1

   10 01 10 11: (Opcode, R1, R2, R3)

   00000001: (Immediate is ignored)

   Convert to hex: 9b01

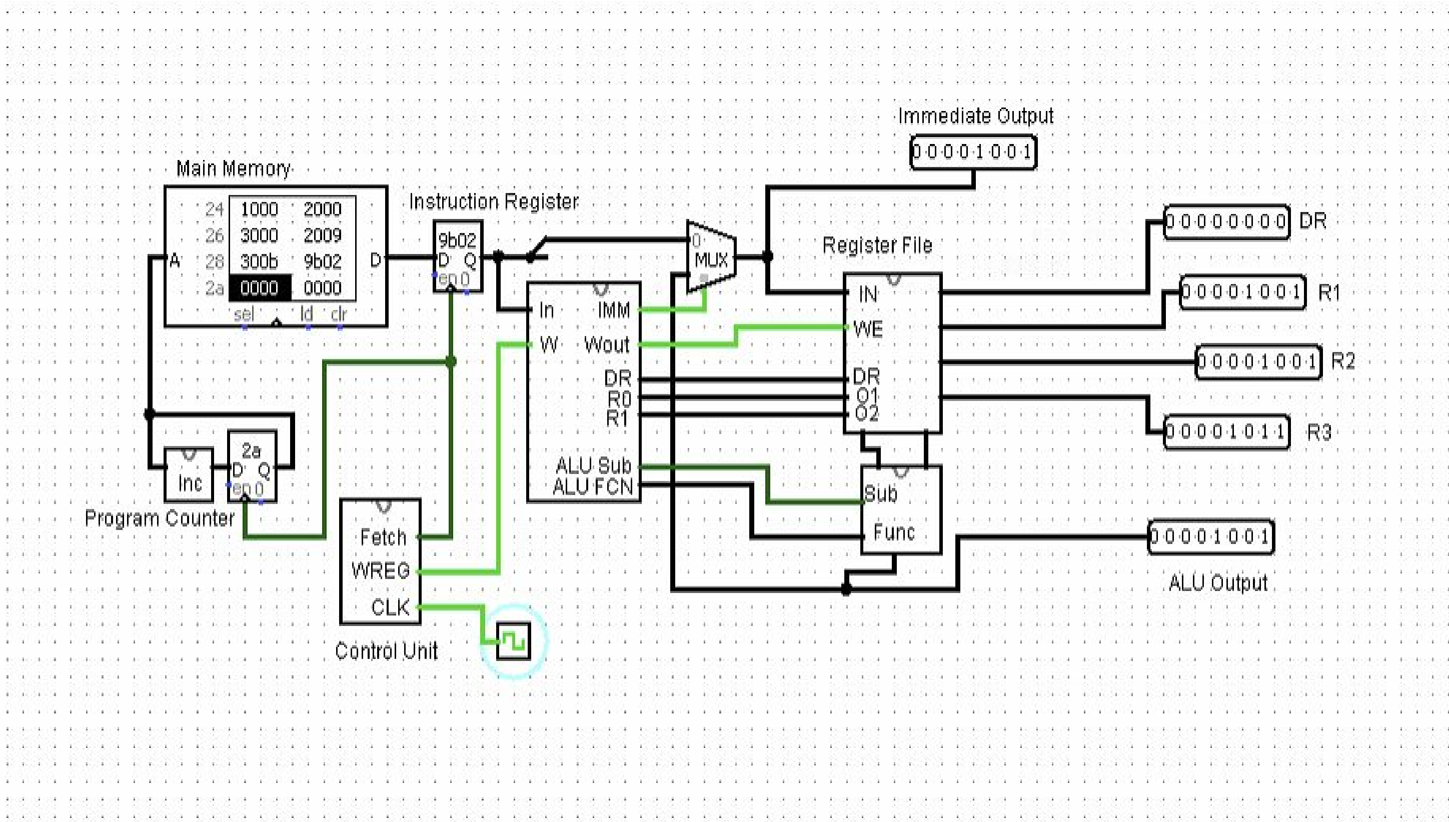


6. AND r2 and r3, result to r1:

   10 01 10 11: (Opcode, R1, R2, R3)

   00000010: (Immediate is ignored)

   Convert to hex: 9b02



3.Citations

1. https://cupola.gettysburg.edu/cgi/viewcontent.cgi?article=1002&context=oer
2. https://www.robots.ox.ac.uk/~dwm/Courses/2CO_2014/2CO-N2.pdf
3. https://youtu.be/XM4lGflQFvA