North South University

Computer Organization & Design

CSE 332

Project Report

Project Name: 10 Bit Single Cycle Processor Design

Course Instructor:
Khaleda Ali (KDA1)
Semester: Fall 18
Section: 12

Team Members:

Name	ID
Md .Nur-E-Azam	1512268042
Rakibuzzaman	1510390042
Fatiha Jahan	1420477042

Contents

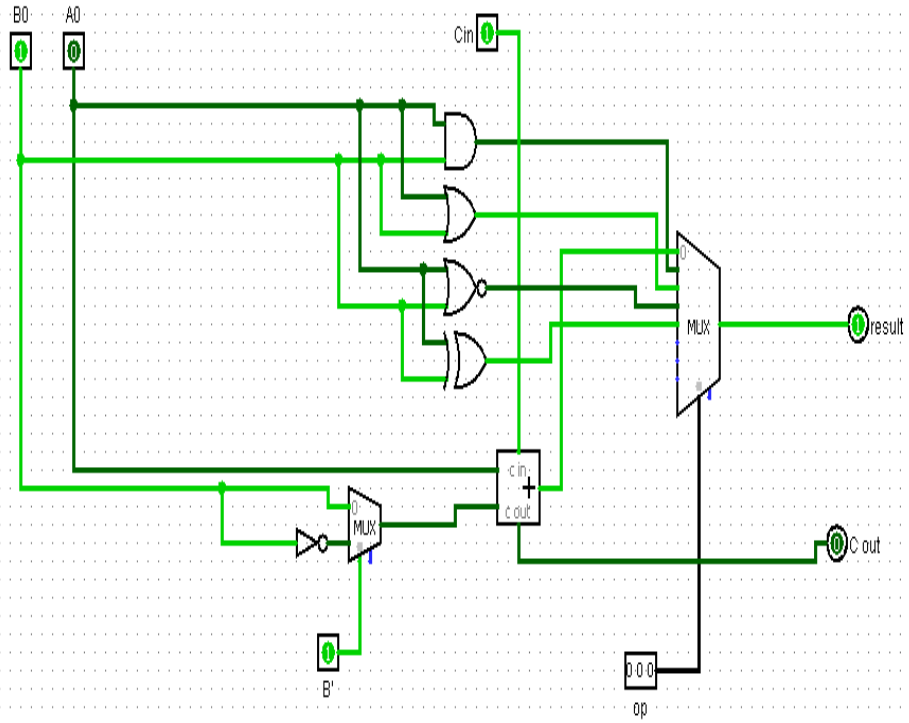
Topic ---------- Page No.

Introduction: ISA (instruction set architecture) decides how to represent an instruction of a processor in Instruction memory, how the instruction is oriented, what is the length of an instruction, in which way the instruction should be decoded so that the CPU gets appropriate data. We already have our ISA. The objective is to design our processor based on the ISA. In this section, we have designed a 10 bit ALU, 10 bit Register File & a Control Unit. Then connected all the component with Program counter, Instruction Memory, Data Memory etc so that our processor works. Our Goal is to execute the 10 instructions (without multiplication) described in ISA successfully in our processor.

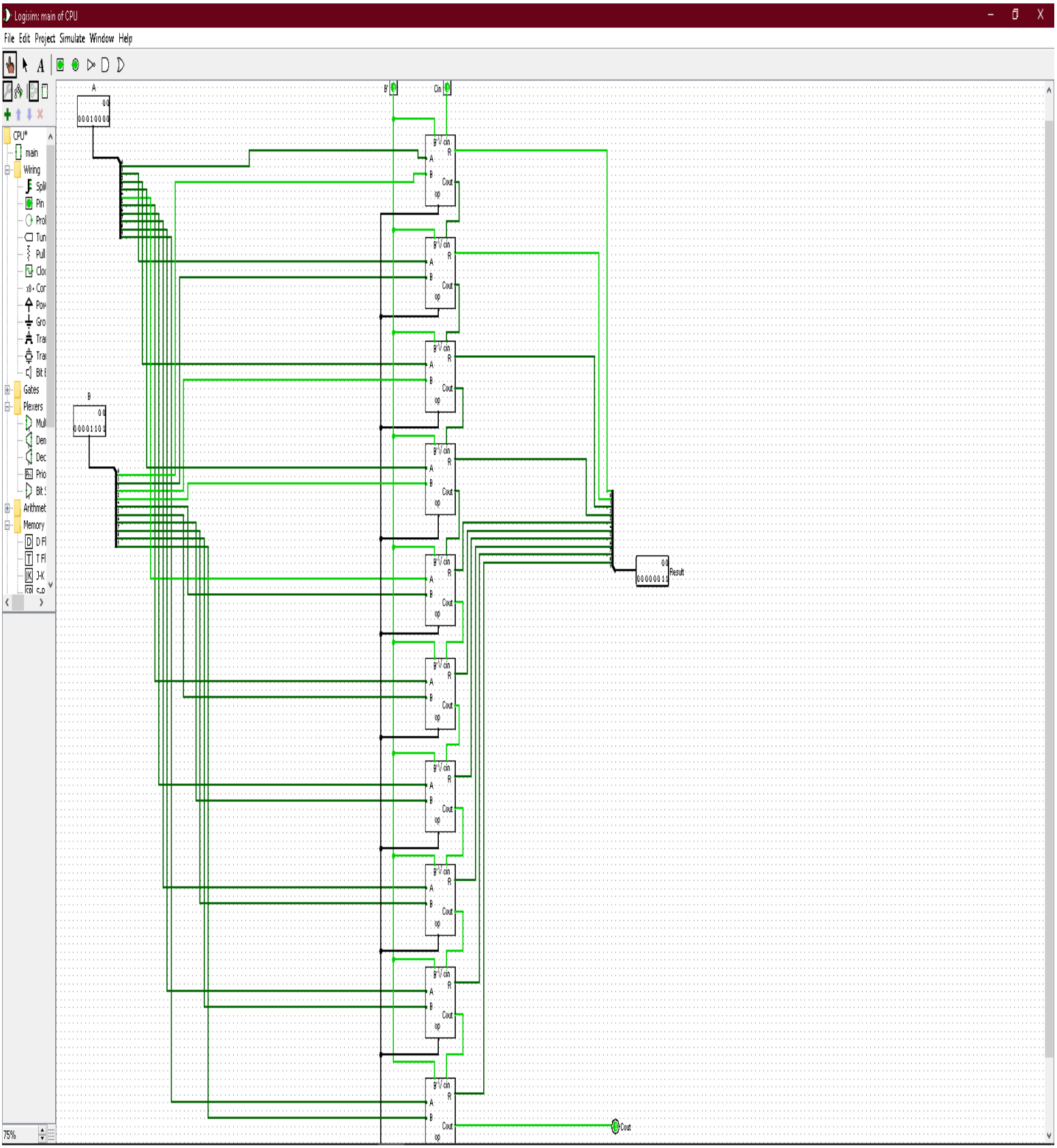
Arithmetic & Logic Unit

(ALU)

1 Bit ALU:



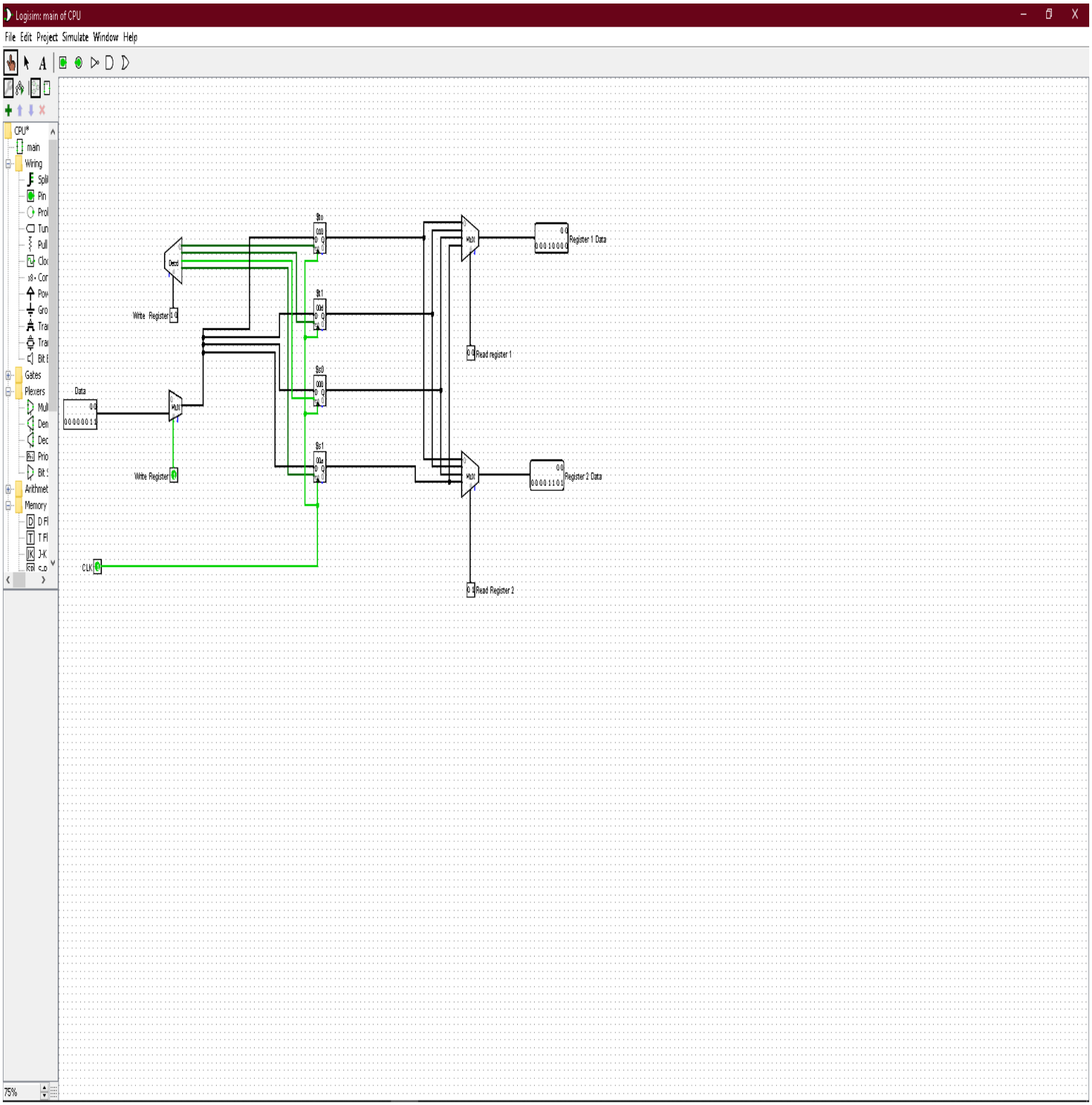
10 bit ALU:


Logisim: main of CPU
File Edit Project Simulate Window Help
CPU*
main
Wiring
Gates
Plexers
Arithmet
Memory
A
B
Cin
Result
Cout
B'V cin
A
B
Cout
op
R
75%

Register File

In processor, ALU gets data from a Register Bank / Register File. As we have designed our ISA, our register file should have 4 registers. And the design should be easy enough to choose **rs, rt, rd** values. The circuit below is our register file design. Note: each register is able to store 10 bit data.

Register File Design:



Control Unit

Control unit is the circuit in processor which controls the flow of data and operation of ALU. Initially an instruction is loaded in instruction memory. According to Opcode of that instruction, control unit gets understood what the instruction wants to execute. Then it flows the data in such a path so that ALU gets necessary data to execute.

Control unit design:
As our designed ISA, our CPU should be able to do 11 operations. The table below represents a demo of 11 operations, binary form of the operations following our ISA design & hex form of the operation.

Instructions Demo:

MIPS CODE	Binary String	Hex Code
add $s1 $t0 $s0	0000 00 10 11	00B
sub $s0 $t1 $s1	0001 01 11 10	05E
and $s1 $t0 $t1	0010 00 01 11	087
or $s1 $t0 $s0	0011 00 10 11	0CB
nor $s0 $t0 $t1	0100 00 01 10	106
addi $s1 $t1 2	1001 01 11 10	25E
andi $s1 $t1 3	1010 01 11 11	29F
ori $s0 $t0 2	1011 00 10 10	2CA
lw $s0 3($t0)	0101 00 10 11	14B
sw $s1 3($t0)	0110 00 11 11	18F
xor $s1 $t1 $t0	1110 01 00 11	393

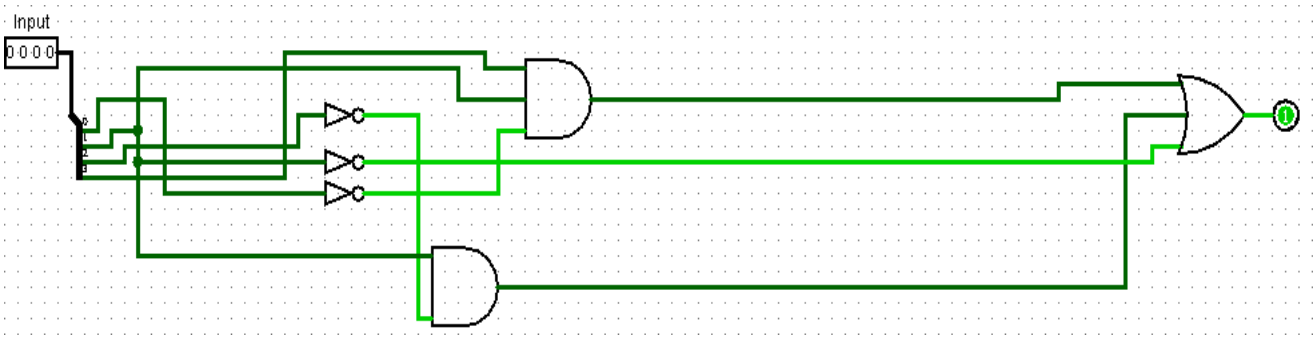
To execute these instructions some bits should be 0/1 according to necessity. The table below describes which instruction needs which bit to be 0/1.

Control Table:

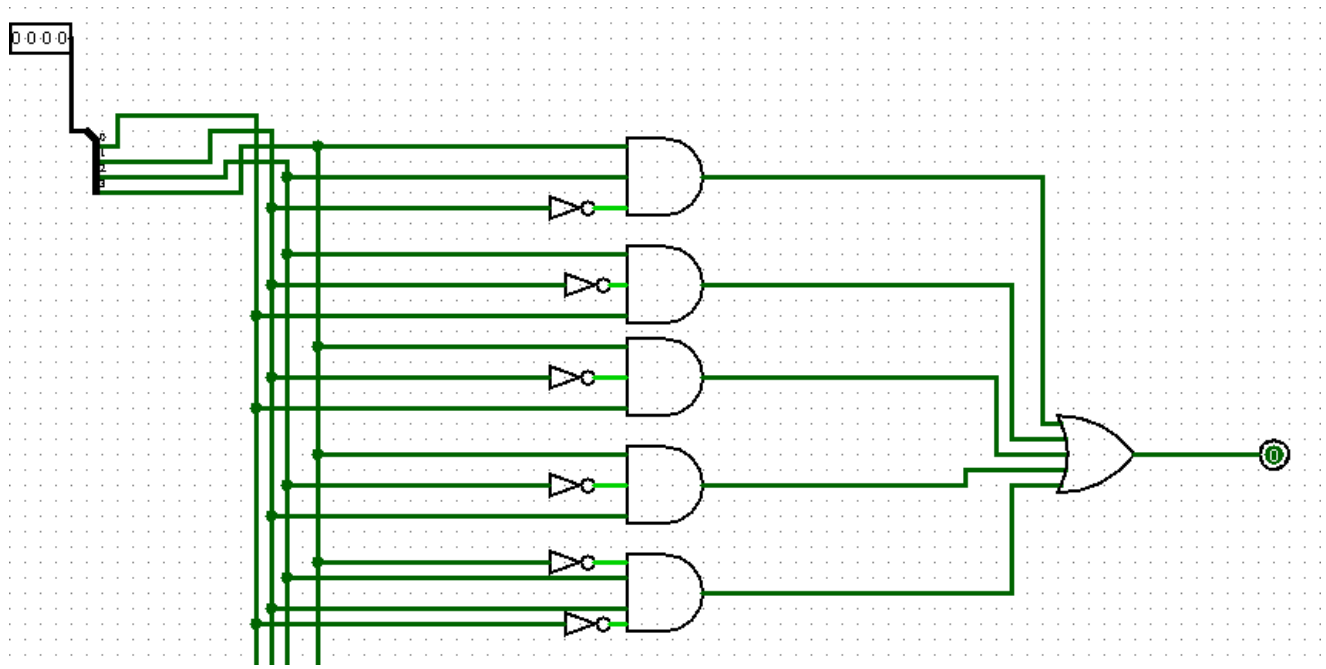
Operation	Op-Code	RegWrite	RegDst	ALUSrc	MemRead	ALUOp	Branch	Jump	Load	Store	A’	B’	Cin	L/R	Shift
add	0000	1	0	0	0	000	0	0	0	0	0	0	0	0	0
sub	0001	1	0	0	0	000	0	0	0	0	0	1	1	0	0
and	0010	1	0	0	0	001	0	0	0	0	0	0	0	0	0
Or	0011	1	0	0	0	010	0	0	0	0	0	0	0	0	0
nor	0100	1	0	0	0	011	0	0	0	0	0	0	0	0	0
lw	0101	1	1	1	1	000	0	0	1	0	0	0	0	0	0
sw	0110	0	1	1	1	000	0	0	0	1	0	0	0	0	0
beq	0111	0	0	0	0	000	1	0	0	0	0	1	1	0	0
mul	1000	0	0	0	0	000	0	0	0	0	0	0	0	0	0
addi	1001	1	1	1	0	000	0	0	0	0	0	0	0	0	0
andi	1010	1	1	1	0	001	0	0	0	0	0	0	0	0	0
ori	1011	1	1	1	0	010	0	0	0	0	0	0	0	0	0
sll	1100	1	1	0	0	000	0	0	0	0	0	0	0	0	1
srl	1101	1	1	0	0	000	0	0	0	0	0	0	0	1	1
xor	1110	1	0	0	0	100	0	0	0	0	0	0	0	0	0
jump	1111	0	0	0	0	000	0	1	0	0	0	0	0	0	0

Following this control table, we built circuits for **RegWrite, RegDst, ALUSrc, MemRead, ALUOp, Branch, Jump, Load, Store, A’, B’, Cin, L/R, Shift**. Minimized with K-map when needed.

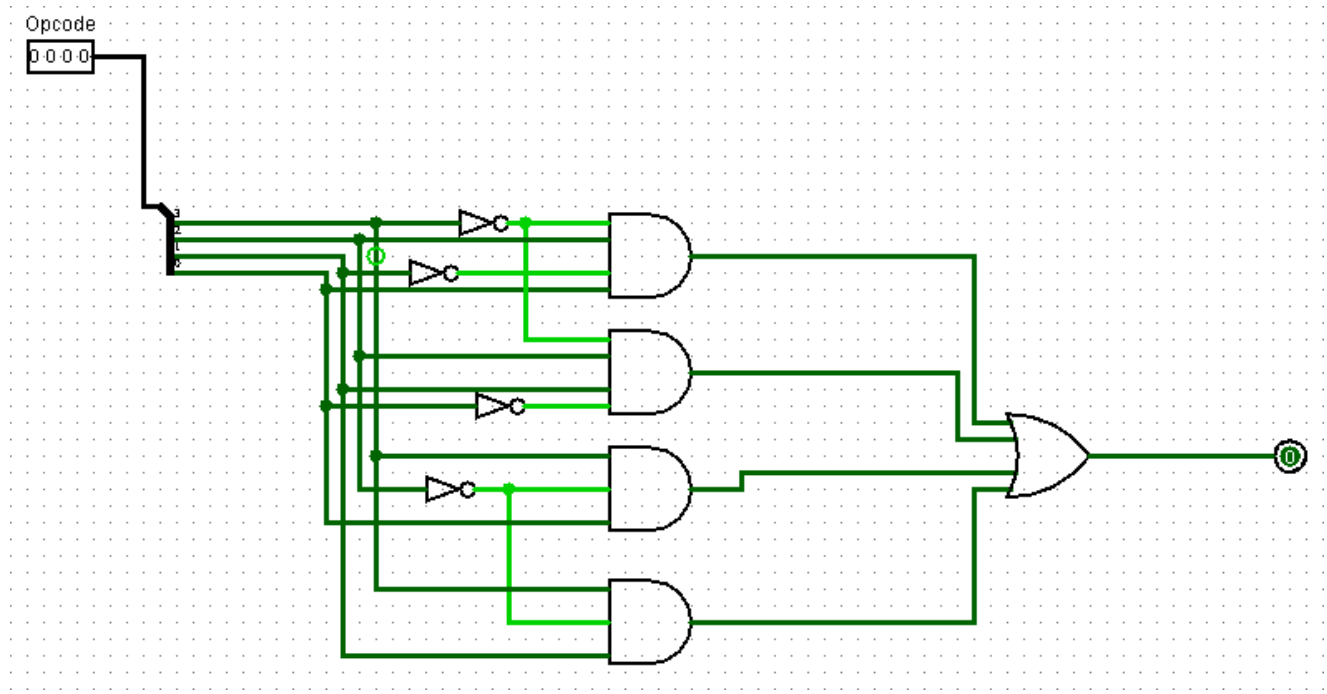
Regwrite:



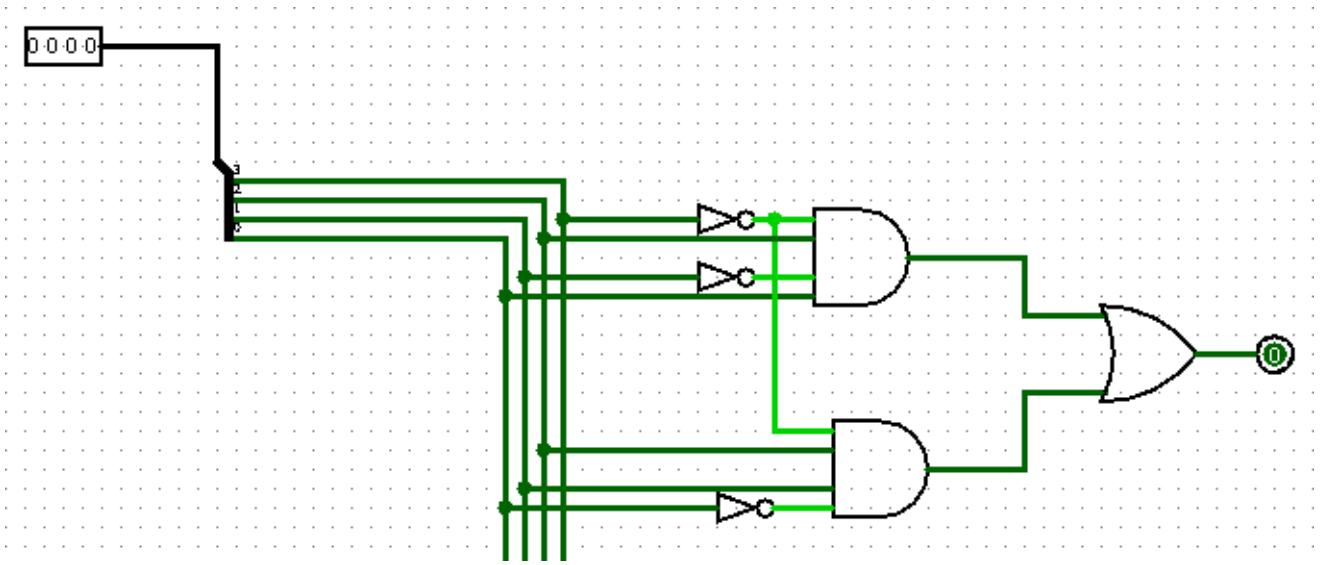
RegDst:



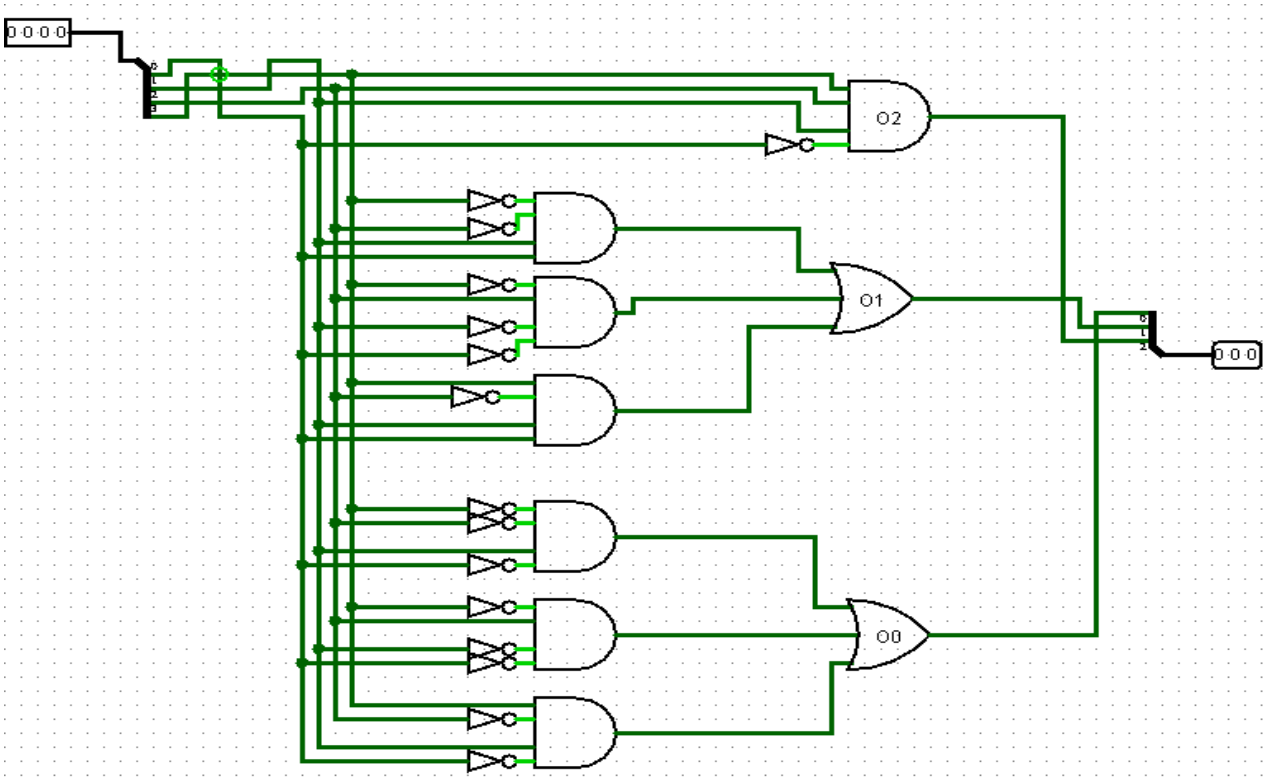
ALUSrc:



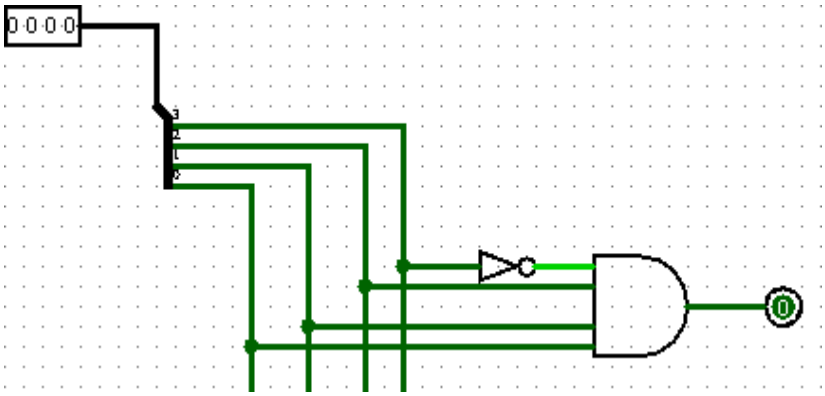
MemRead:



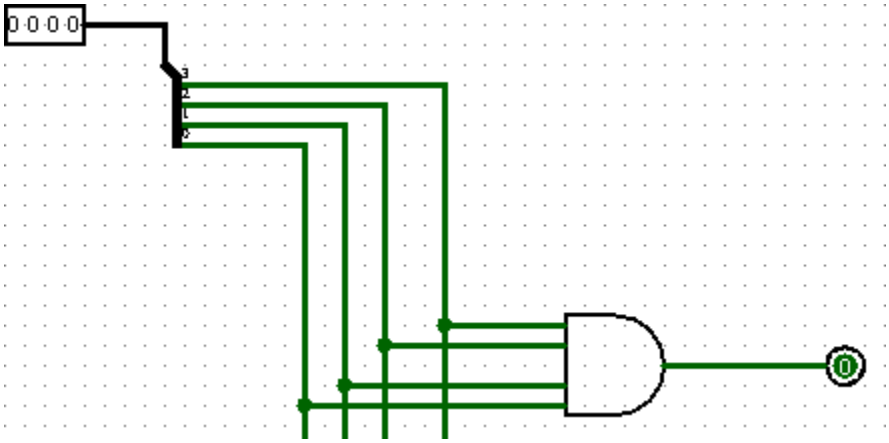
ALUOp:



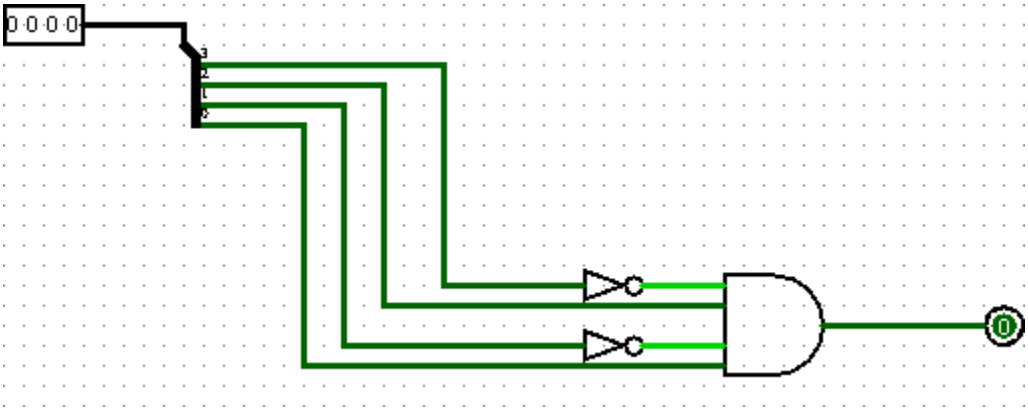
Branch:



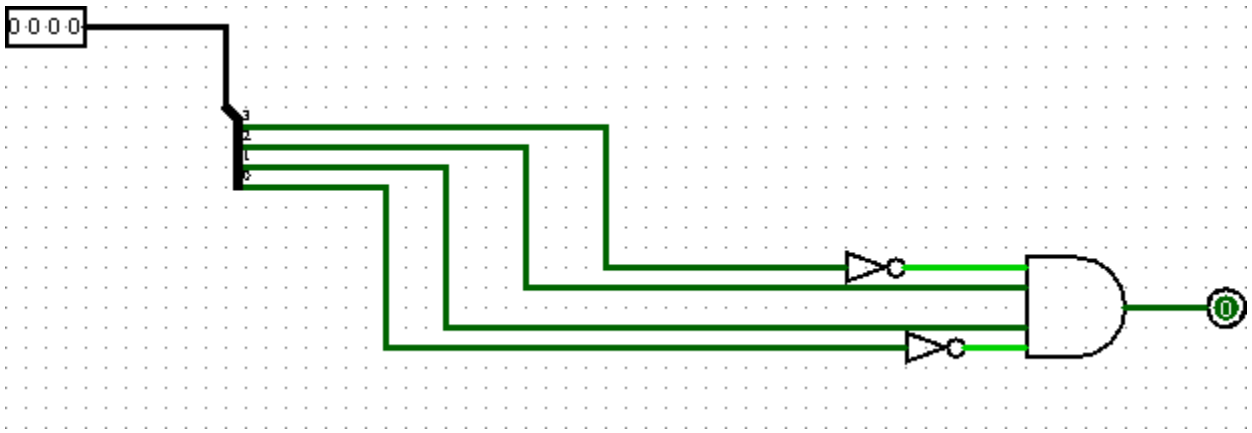
Jump:



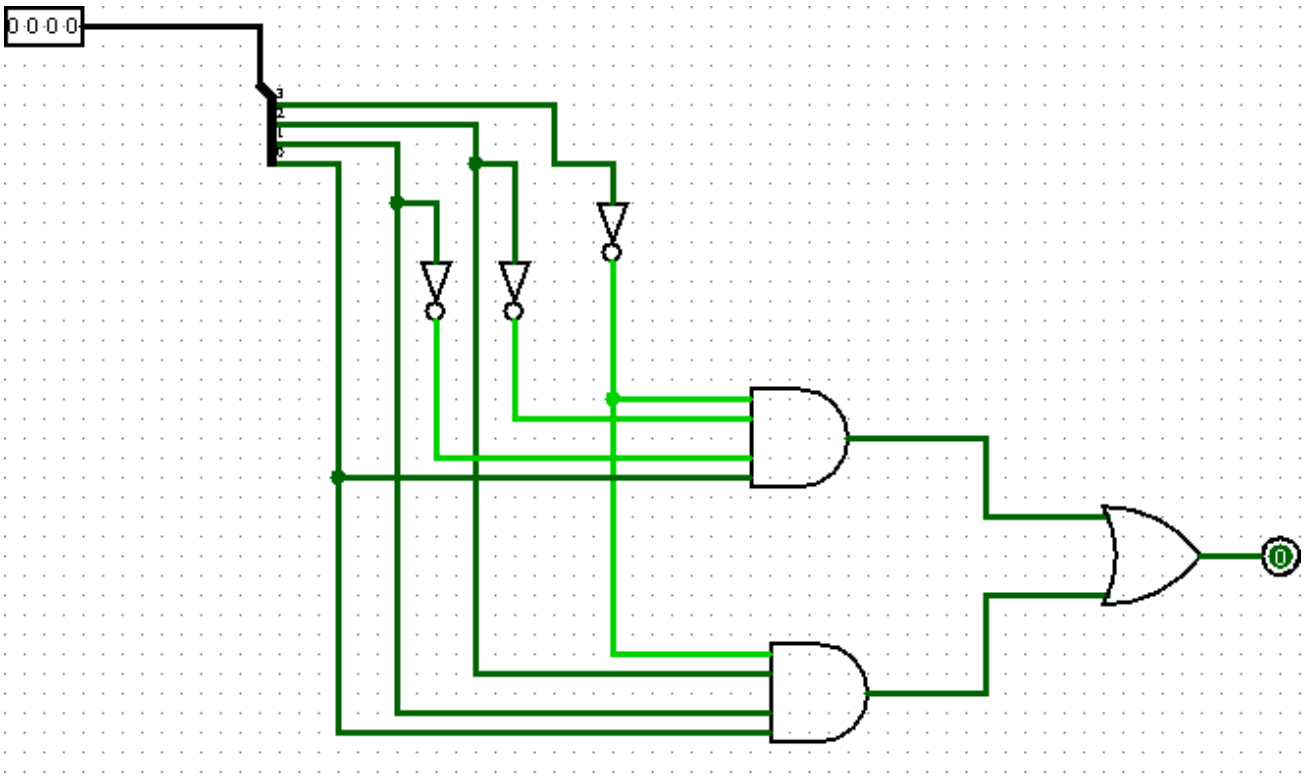
Load:



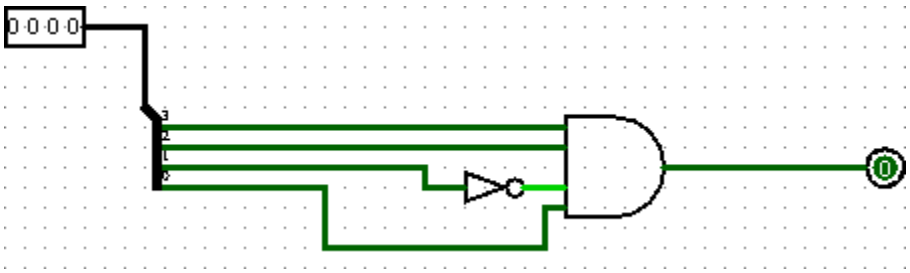
Store:



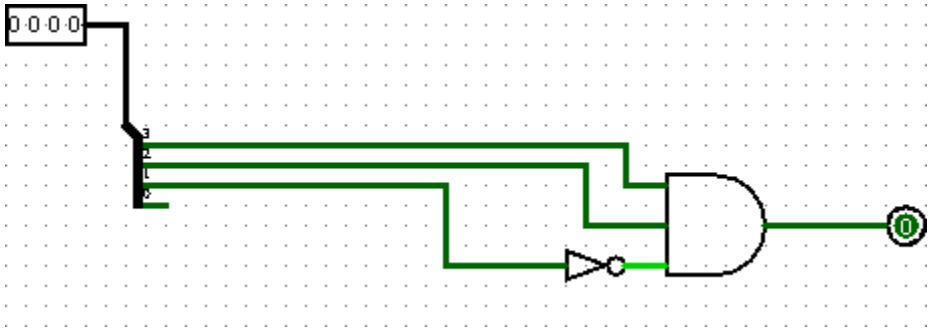
B'/Cin:



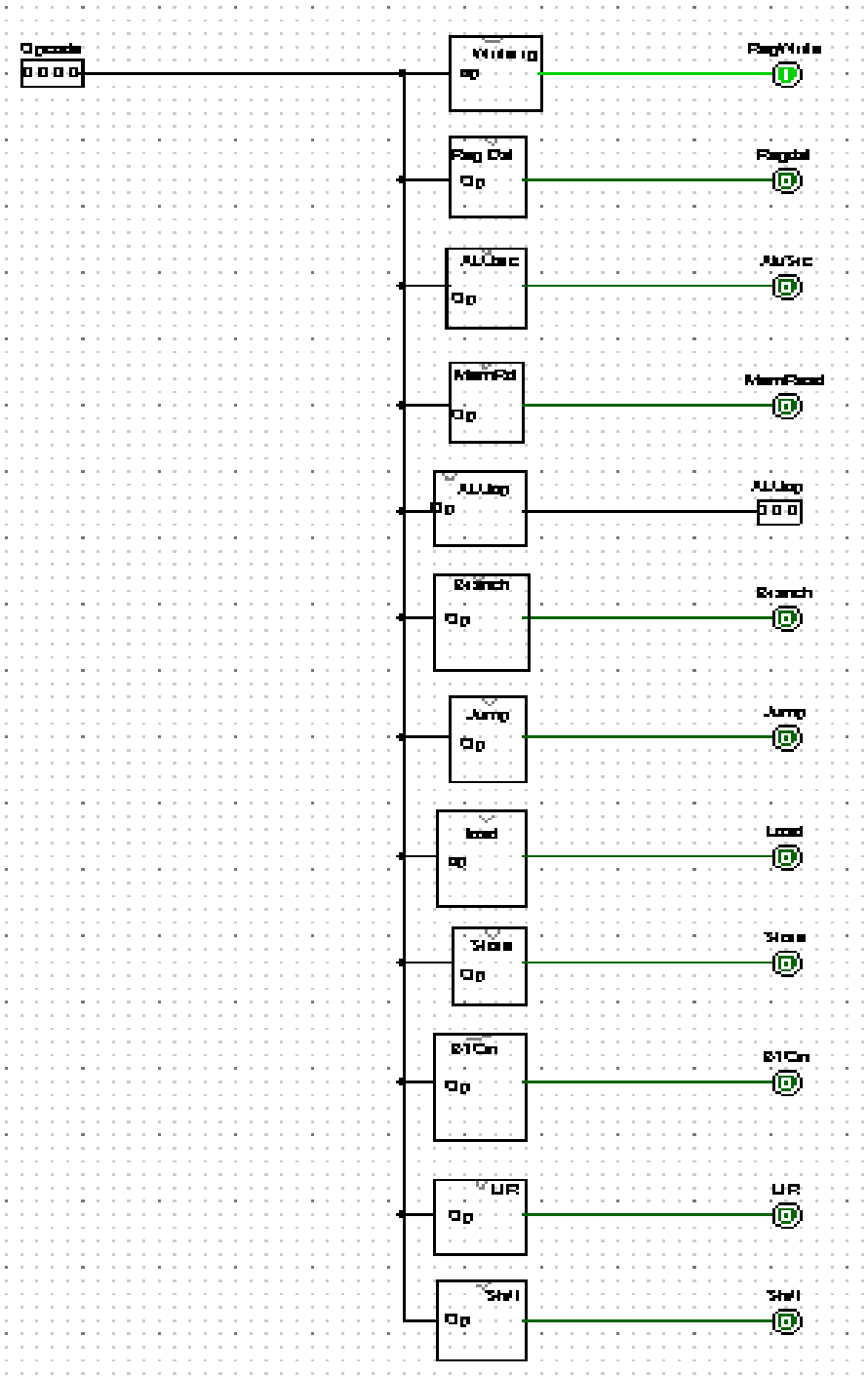
L/R:



Shift:



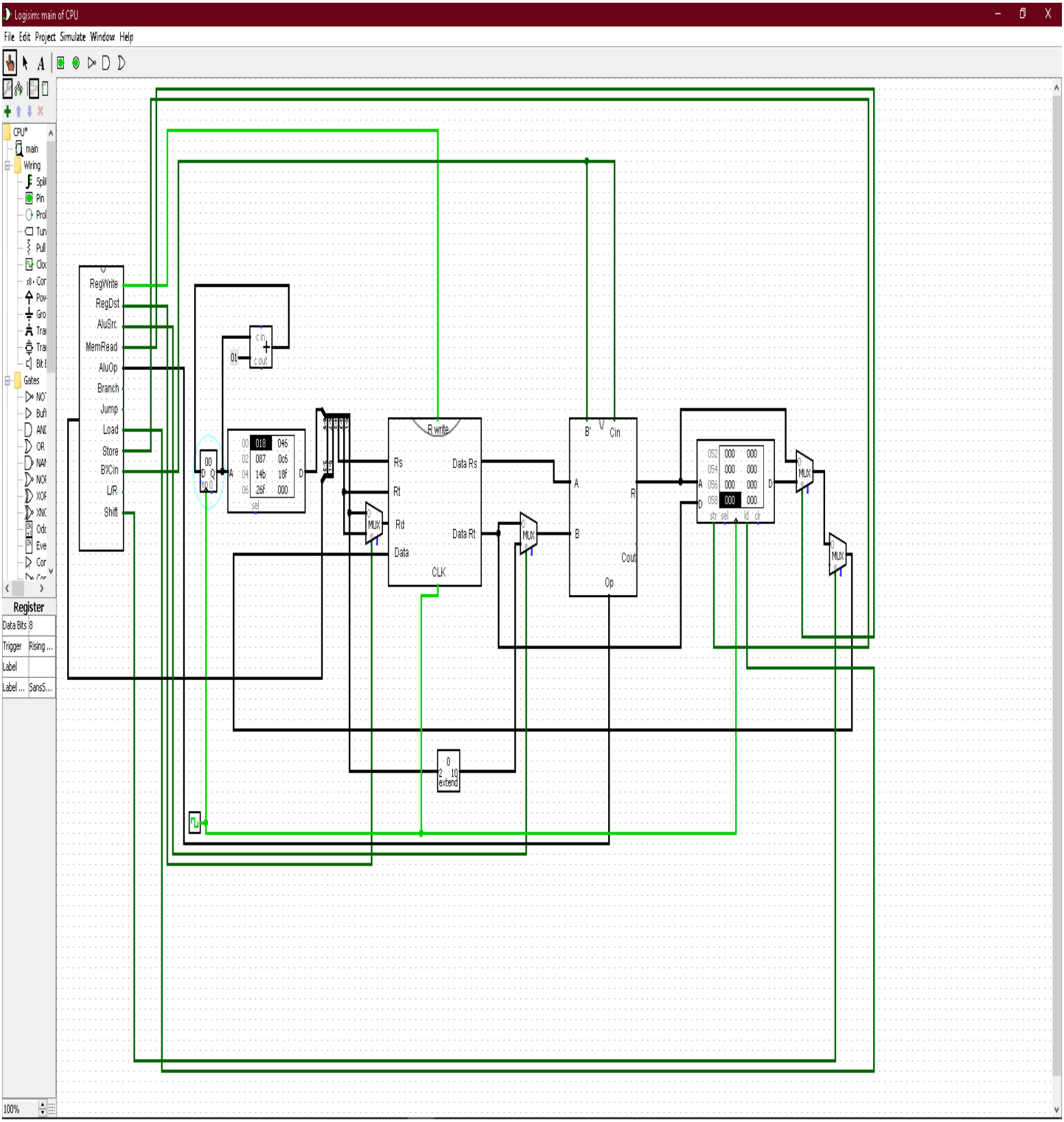
Control Unit Circuit:



Processor Circuit

Processor Circuit:

Now, connect ALU, Register file, Instruction Memory, Data Memory, control unit and all other hardwires to complete the design. The circuit below is our complete 10 bit processor according to our ISA design.



Conclusion:

The processor is able to successfully execute the 10 operations from ISA design. To execute any instruction, the syntax and the orientation of different type of instruction explained in ISA must be followed. Otherwise the CPU won’t work properly.