



North South University

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

PROJECT REPORT

---

## Complete Datapath based on IAS with assigned bits and operations

---

COMPUTER ORGANIZATION AND ARCHITECTURE  
CSE332 (Section 5)  
SUMMER 2024

**Group-11**

<b>Member Name</b>	<b>Student ID</b>
Saif Mohammed	2121913042
Humayra Rahman Nipa	2121128042

**Course Faculty:** Tanjila Farah  
**Lab Instructor:** Shuvashish Chakraborty

**Submission Date:** 23 December, 2024

# Table of Control Unit & ALU control

## 1. ALU Control Table

Instruction Opcode	ALUOp	Instruction Operation	Function Field	Desired ALU Action	ALU Control Input
000	10	SLL	000	Shift Left Logical	000
000	10	SLT	001	Set Less Than	001
001	00	SW	xxx	Add	011
000	10	SRL	010	Shift Right Logical	010
010	01	BNE	xxx	Subtract	111
000	10	ADD	011	Add	011
000	10	OR	100	Bitwise OR	100
011	01	BEQ	xxx	Subtract	110
000	10	NOR	101	Bitwise NOR	101
100	00	NOP	xxx	Shift Logical Left	000
101	00	ADDi	xxx	Add	011
110	00	LW	xxx	Add	011
000	10	AND	110	Bitwise AND	110
111	00	JMP	xxx	No ALU Action	xxx
000	10	SUB	111	Subtract	111

## 2. Control Unit Signal Table

Instr. uction	Reg Dst	ALU Src	Mem Reg	Reg Write	Mem Read	Mem Write	Bra-nch	Jump	ALU Op1	ALU Op0
LW	0	1	1	1	1	0	0	0	0	0
SW	X	1	X	0	0	1	0	0	0	0
ADDi	0	1	0	1	0	0	0	0	0	0
BEQ	X	0	X	0	0	0	1	0	0	1
BNE	X	0	X	0	0	0	1	0	0	1
R-Type	1	0	0	1	0	0	0	0	1	0
NOP	X	X	X	0	0	0	0	0	1	0
JMP	X	X	X	0	0	0	0	1	X	X

### 3. ALU Control Truth Table

ALUOp1	ALUOp0	Funct2	Funct1	Funct0	O2	O1	O0
0	0	x	x	x	0	1	1
0	1	x	x	x	1	1	1
1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	1	0
1	0	0	1	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	1	0
1	0	1	1	1	1	1	1

# Instruction Set Architecture

## R-Type

[19:17]	[16:13]	[12:9]	[8:5]	[4:3]	[2:0]
<b>op</b> (3-bit)	<b>rs</b> (4-bit)	<b>rt</b> (4-bit)	<b>rd</b> (4-bit)	<b>shamt</b> (2-bit)	<b>funct</b> (3-bit)

## I-Type

[19:17]	[16:13]	[12:9]	[8:0]
<b>op</b> (3-bit)	<b>rs</b> (4-bit)	<b>rd</b> (4-bit)	<b>immediate(9-bit)</b>

## J-Type

[19:17]	[16:0]
<b>op</b> (3-bit)	<b>target address(17-bit)</b>

## 1. Number of Operands

### R-type instructions

The R-type instructions in the instruction set architecture employ three distinct operands, consisting of

- Two source registers (rs and rt)
- One destination register (rd)

### I-type instructions

The I-type instructions utilize two operands, where one is consistently

- One source register (rs) and
- 9-bit immediate value or a destination register (rd)

### J-type instructions

The J-type instructions operate with a single operand.

- 17-bit target address

**Total Number of Operands: 7**

## 2. Types of Operands

### Register-Based Operands

- Used in R-Type instructions, where all operands ( $rs$ ,  $rt$ ,  $rd$ ) are registers.
- Also used in I-Type instructions for arithmetic and logical operations (e.g.,  $rs$  and  $rd$ ).

### Memory-Based Operands

- Found in I-Type instructions for *load* (LW) and *store* (SW) operations, where the *immediate* field specifies an offset, and  $rs$  points to a memory address.

### Address-Based Operands

- Found in J-Type instructions, where the operand is a memory address used for branching or jumping.

Types of Operands: 3

## 3. List of Operations

Number	Operation	Description	Operation Format
1	SLL	Shift Left Logical	$\$rd = \$rs \ll shamt$
2	SLT	Set Less Than	$\$rd = (\$rs < \$rt) ? 1 : 0$
3	SW	Store Word	$Memory[\$rs + immediate] = \$rt$
4	SRL	Shift Right Logical	$\$rd = \$rs \gg shamt$
5	BNE	Branch if Not Equal	$if (\$rs != \$rt) PC += immediate$
6	ADD	Addition	$\$rd = \$rs + \$rt$
7	OR	Logical OR	$\$rd = \$rs   \$rt$
8	BEQ	Branch if Equal	$if (\$rs == \$rt) PC += immediate$
9	NOR	Logical NOR	$\$rd = (\$rs   \$rt)$
10	NOP	No Operation	No operation performed
11	ADDi	Add Immediate	$\$rd = \$rs + immediate$
12	LW	Load Word	$\$rd = Memory[\$rs + immediate]$
13	AND	Logical AND	$\$rd = \$rs \& \$rt$
14	JMP	Jump	PC = target address
15	SUB	Subtraction	$\$rd = \$rs - \$rt$

## 4. Types of Operations

Operation Type	Operation Name	Opcode	Function Bits
Arithmetic	ADD	000	000
Arithmetic	ADDi	101	101
Arithmetic	SUB	000	111
Logical	AND	000	110
Logical	OR	000	100
Logical	NOR	000	101
Logical	SLL	000	000
Logical	SRL	000	010
Logical	SLT	000	001
Logical	NOP	100	xxx
Branch	BEQ	011	xxx
Branch	BNE	010	xxx
Branch	JMP	111	xxx
Memory	LW	110	xxx
Memory	SW	001	xxx

## Summary of Operations

Operation	Count
Arithmetic	3
Logical	7
Branch	3
Memory	2

## 5. Number of Instruction Formats

There are total 3 types of Instructions Formats used in the Instruction Set Architecture.

- R-Type
- I-Type
- J-Type

## 6. Instruction Format Description

### R-Type (Register-Type)

- **Purpose:** Used for arithmetic and logical operations.

- **Field Description:**

- $op$  (3-bit): Operation code.
- $rs$  (4-bit): Source register 1.
- $rt$  (4-bit): Source register 2.
- $rd$  (4-bit): Destination register.
- $shamt$  (2-bit): Shift amount (used in shift operations).
- $funct$  (3-bit): Function code to specify the exact operation.

- **Field Lengths:** Total: 20 bits.

## I-Type (Immediate-Type)

- **Purpose:** Used for immediate value operations, memory access, and branching.

- **Field Description:**

- $op$  (3-bit): Operation code.
- $rs$  (4-bit): Source register.
- $rd$  (4-bit): Destination register.
- $immediate$  (9-bit): Immediate value or offset for memory.

- **Field Lengths:** Total: 20 bits.

## J-Type (Jump-Type)

- **Purpose:** Used for jump instructions.

- **Field Description:**

- $op$  (3-bit): Operation code.
- $target\ address$  (17-bit): Address for the jump.

- **Field Lengths:** Total: 20 bits.

## 20-bit Single Cycle CPU Screenshots

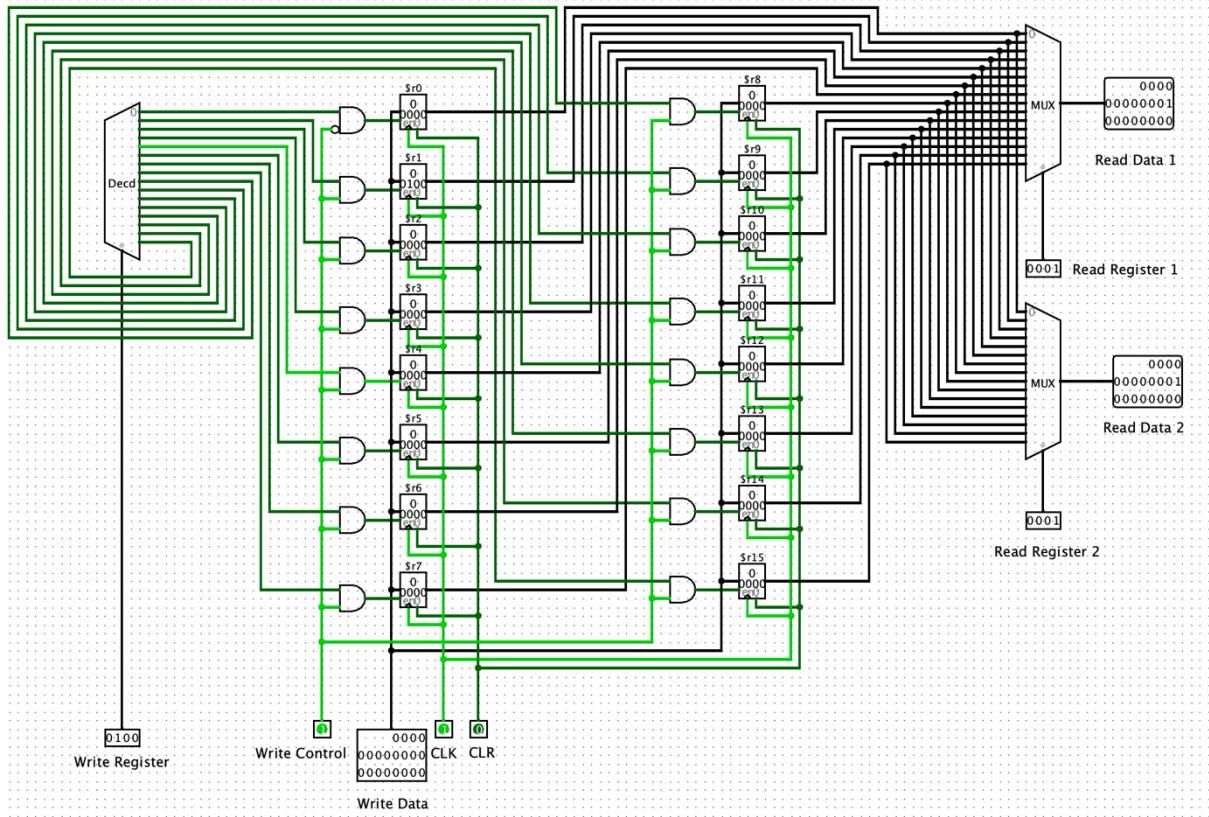


Figure-1: Register File

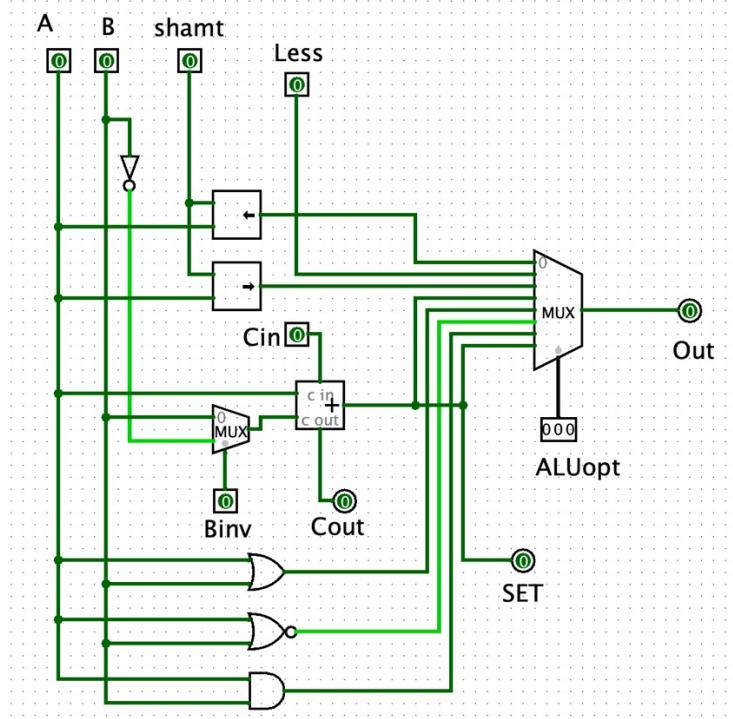
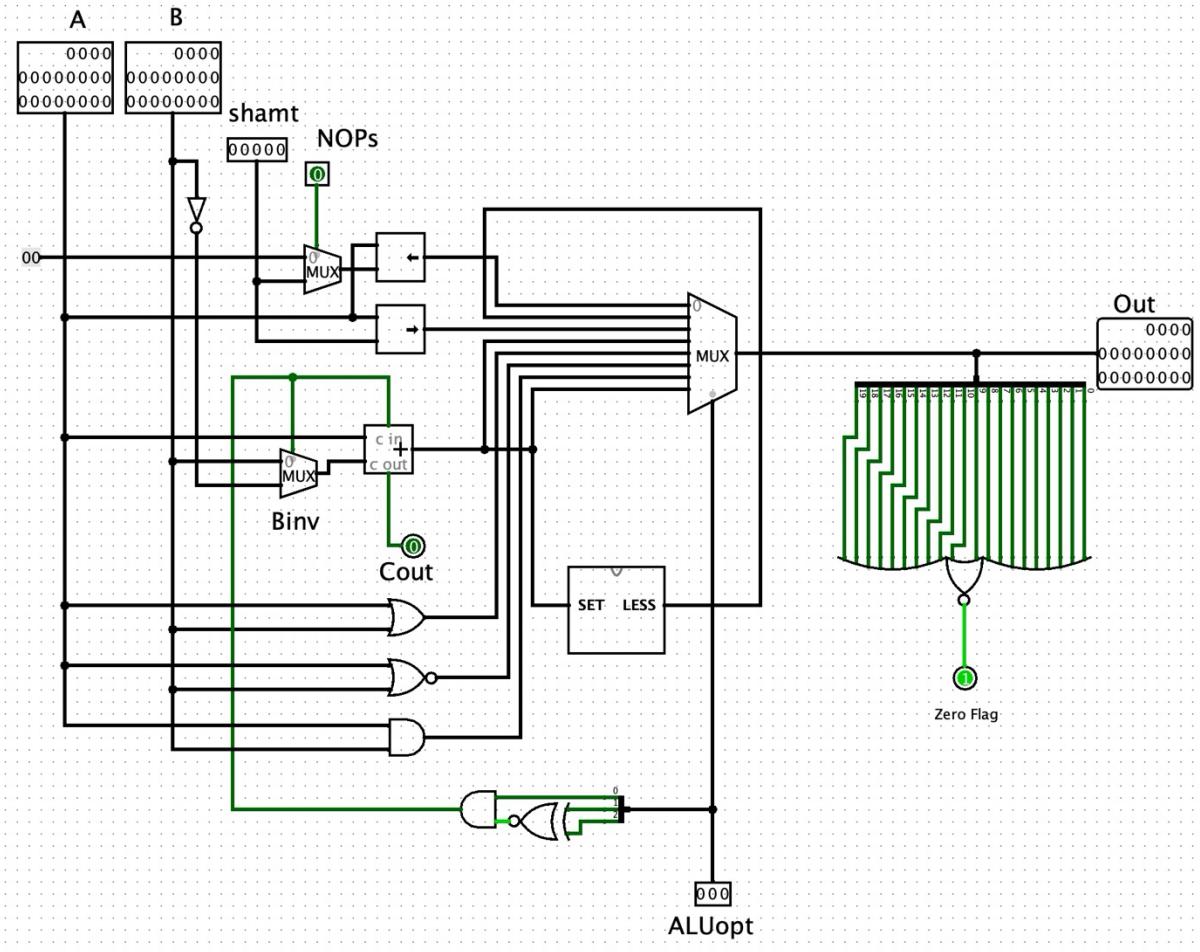
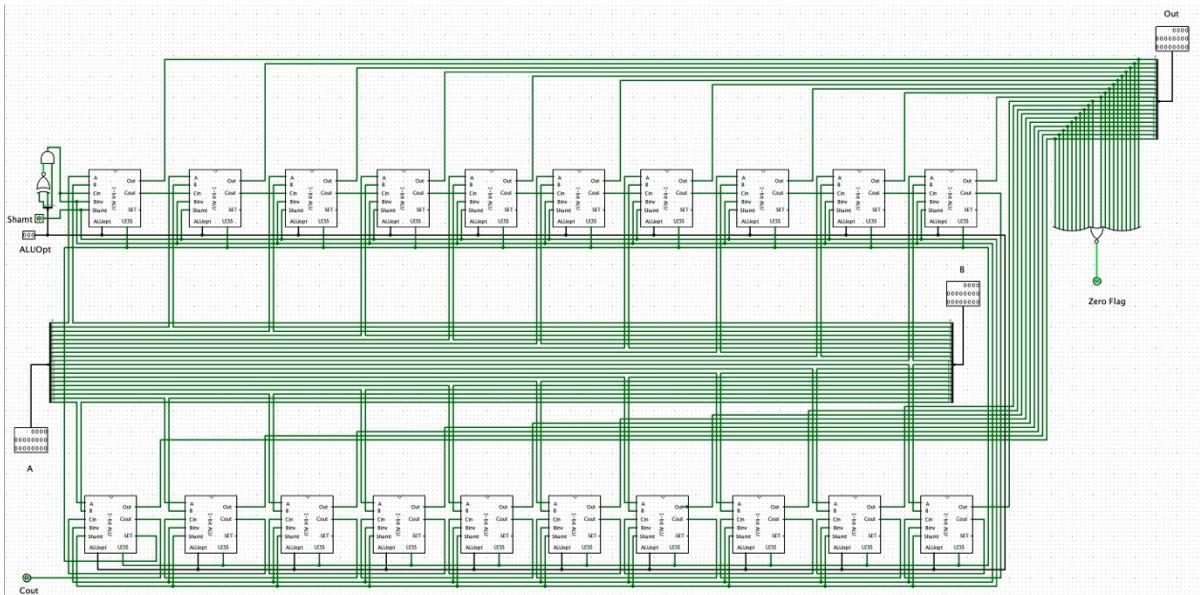


Figure-2: 1-bit ALU

## 20-bit Single Cycle CPU Screenshots

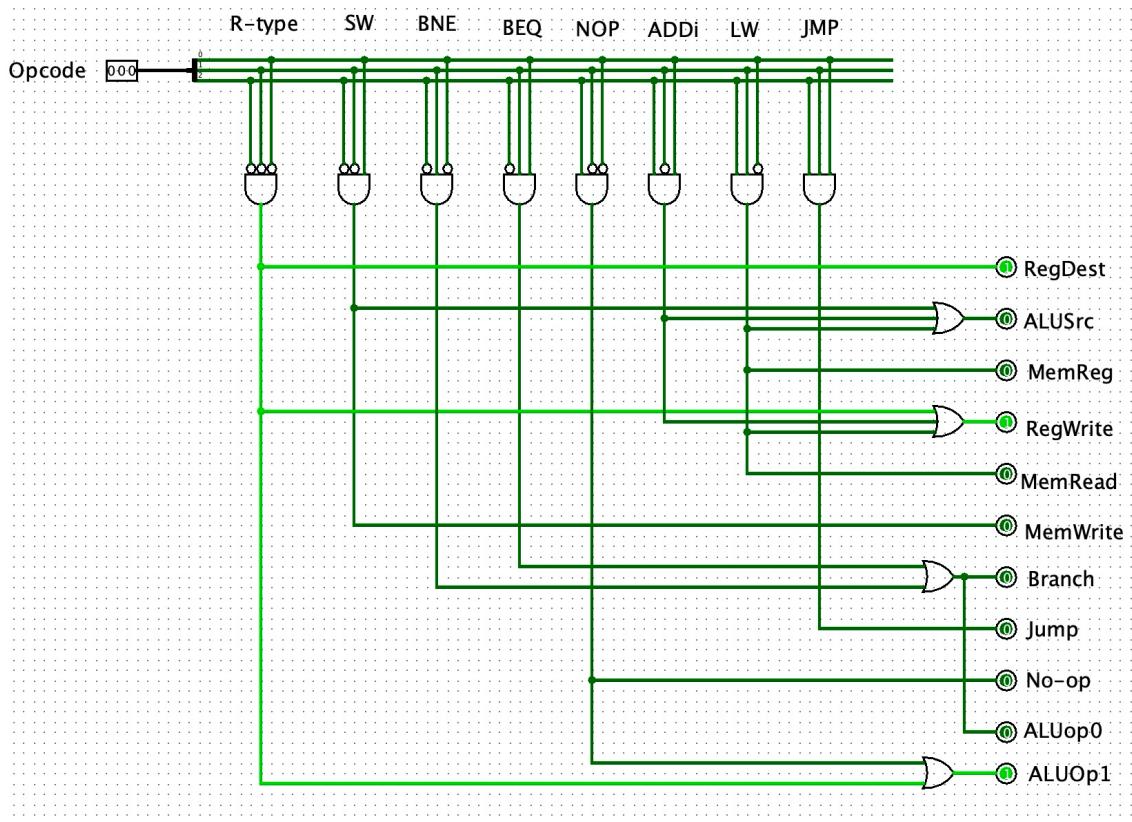


**Figure-3:** 20-bit ALU Control (Minimized)

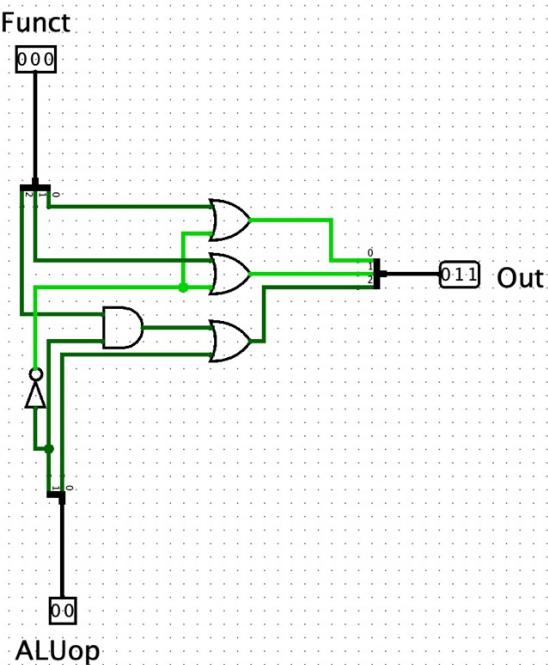


**Figure-4:** 20-bit ALU using 1-bit ALU Sub-Circuit

## 20-bit Single Cycle CPU Screenshots

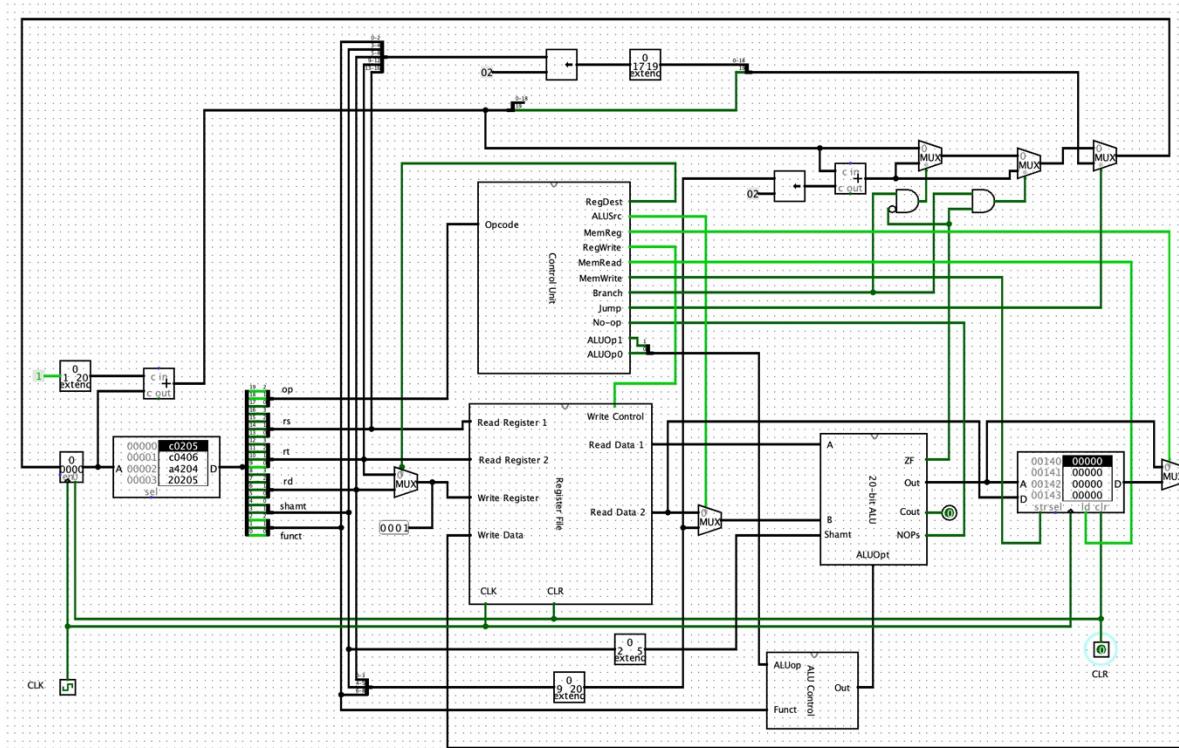


**Figure-5:** Control Unit



**Figure-6:** ALU Control

## 20-bit Single Cycle CPU Screenshots



**Figure-7: Single Cycle Datapath**