COLLEGE OF COMPUTER SCIENCE & ENGINEERING

Informantion & Computer Science Department

COE301 PROJECT GROUP:4

Single Cycle and Pipelined Processor

ABSTRACT: -
Designing a Pipelined 32-bit processor with 16-bit instructions

Techer: Saleh AlSaleh

Students:
Abdulaziz Osama Bassam

Table of Contents

1.Design and Implementation

I. Design Detailed Description

- **ALU**, there are 4 type of operation used which are logical, arithmetic, Shift, and compare.
  - **Logical**, there is 4 choices. These are:
    - **AND**
    - **Complement AND**
    - **OR**
    - **XOR**
  - **Arithmetic**, there are 2 operation. These are:
    - **adding**
    - **negative adding** that subtract the value for the Rb value with Ra value.
  - **Shifting**, there are 3 shift 1 rotate instruction that are:
    - **Shift logical left and**
    - **Shift logical right**
    - **Shift right arithmetic**
    - **Right operation**
  - **Compare**, there are 2 operation used which are:
    - **Equal**
    - **Lower than**

- **Hazard Detect Forwarding & Stalling,** to make this circuit we use multiplexer to assign If Else statemen. 3 multiplexers used in sequence from the most important to the least important if statement as follow.
  - **First, Execute Register Write**
  - **Second, Memory Register Write**
  - **Third, Write Back Register Write**

II. Component Circuits Drawings

1- PC Control

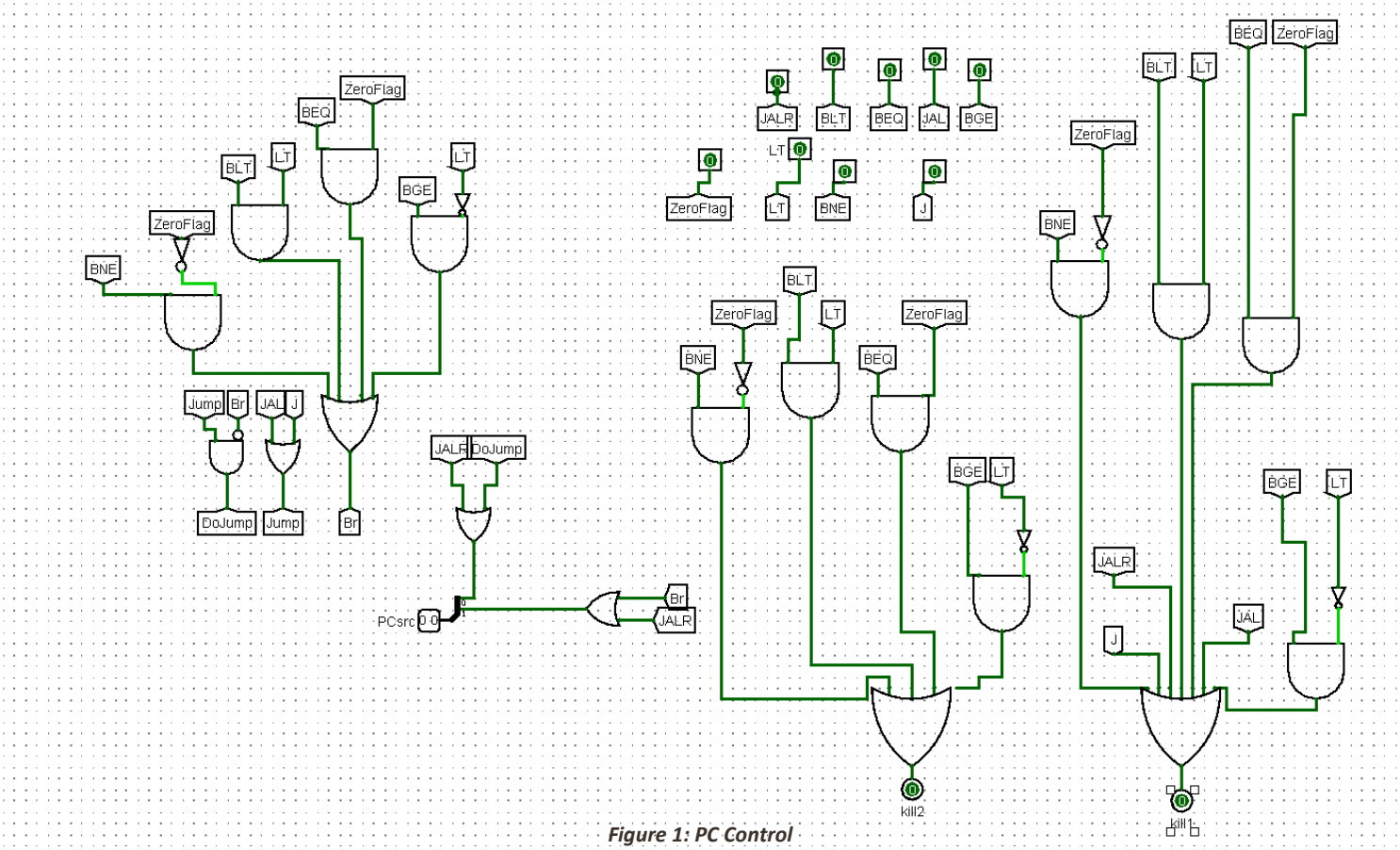


Figure 1: PC Control

2-Main_ALU_Control

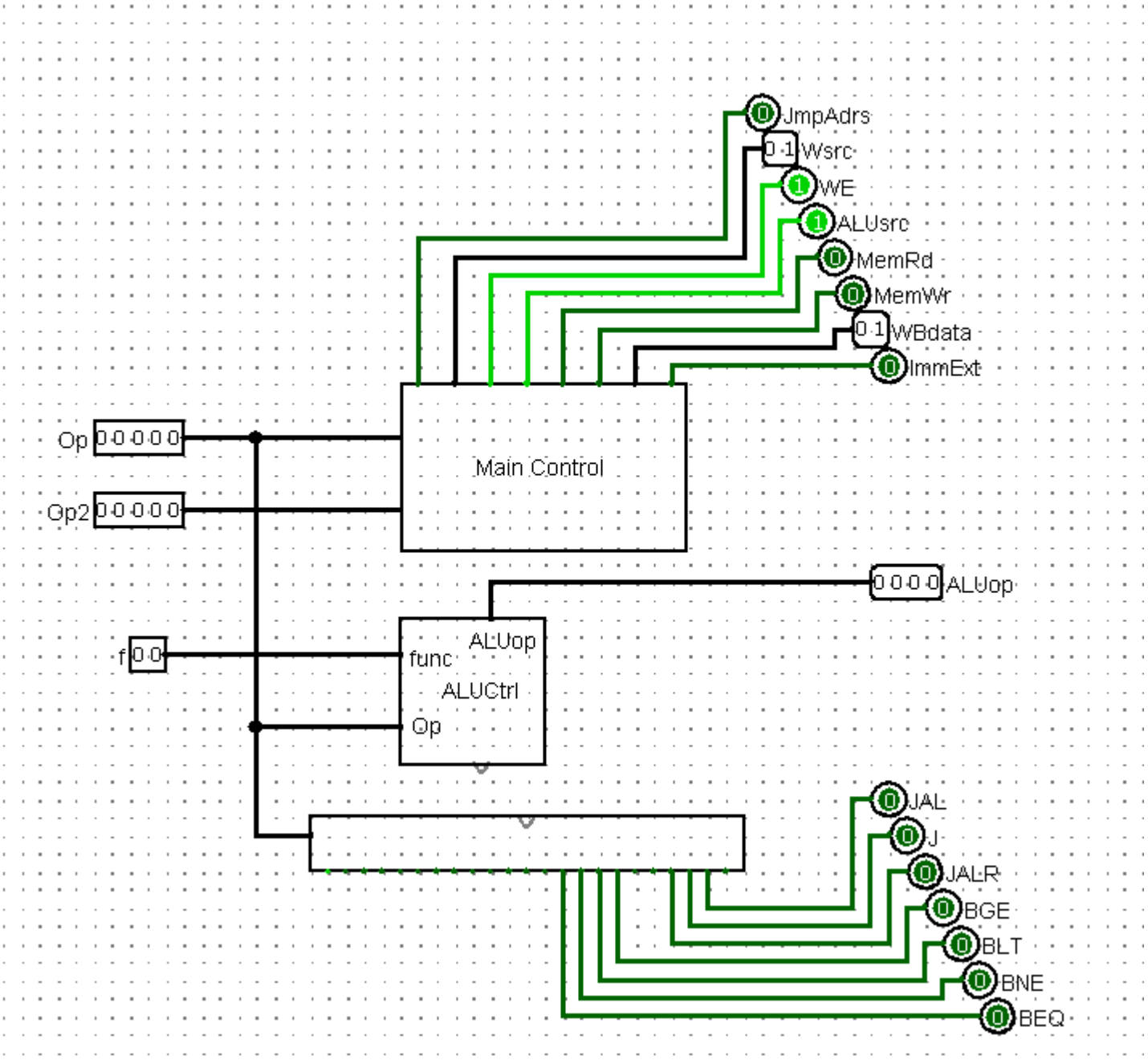


Figure 2: Main_ALU_Control

3-Register File

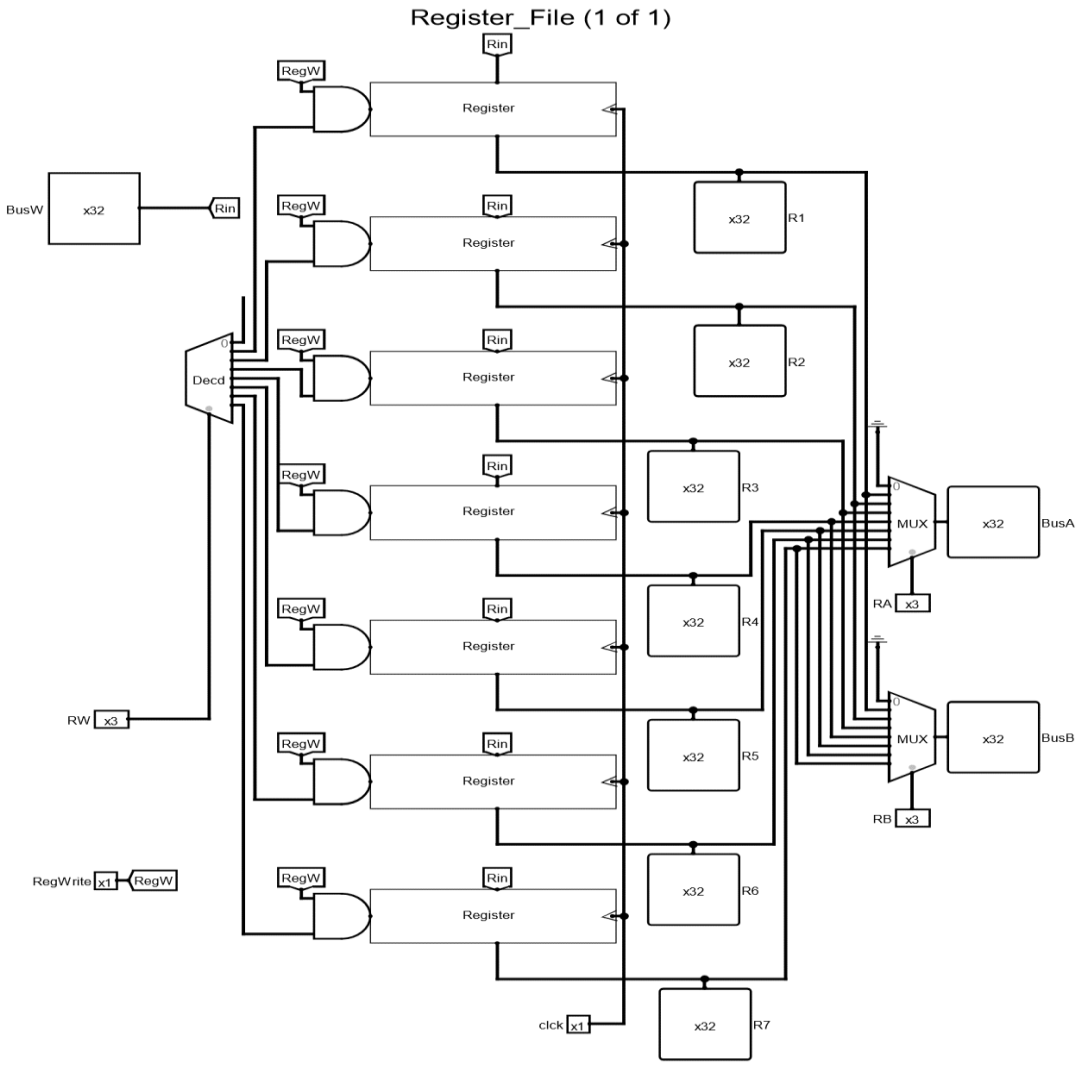


Figure 3: Register File

4-Register

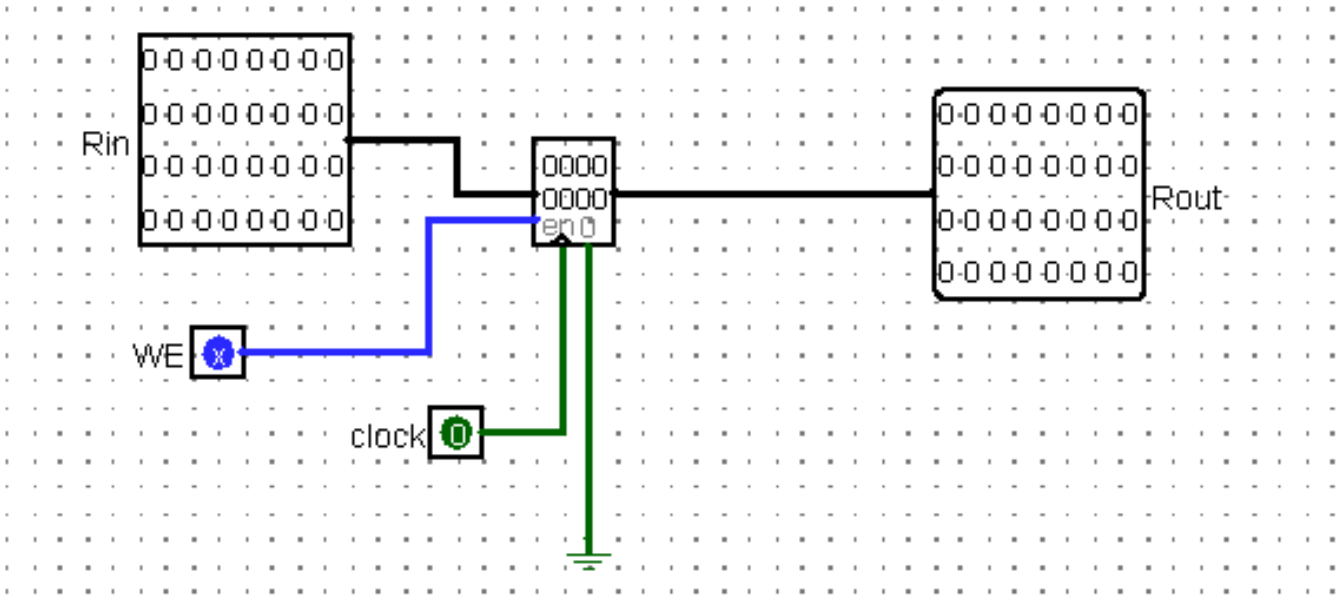


Figure 4: Register

5-PC Control

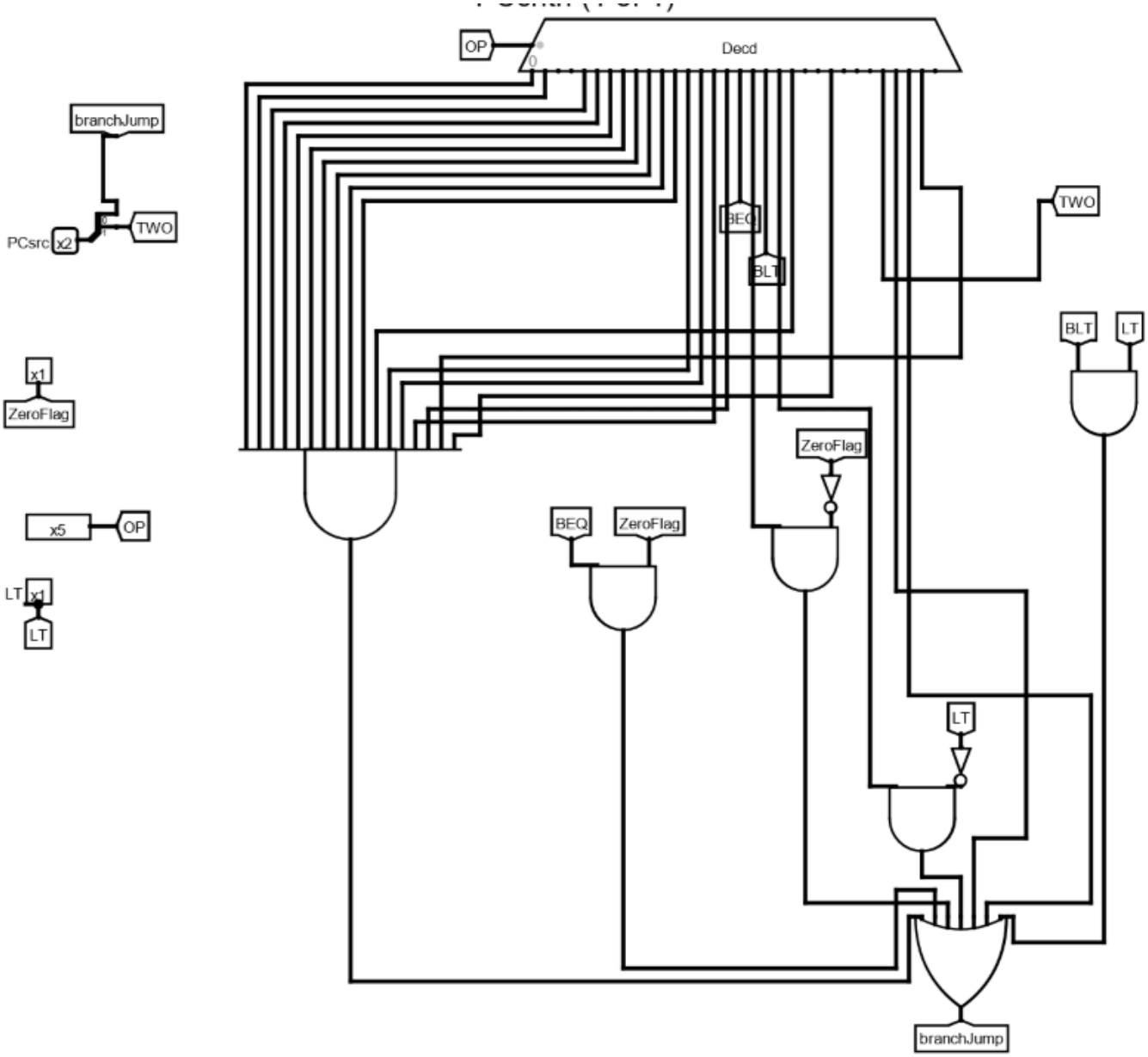


Figure 5: Pc Control

6-PC

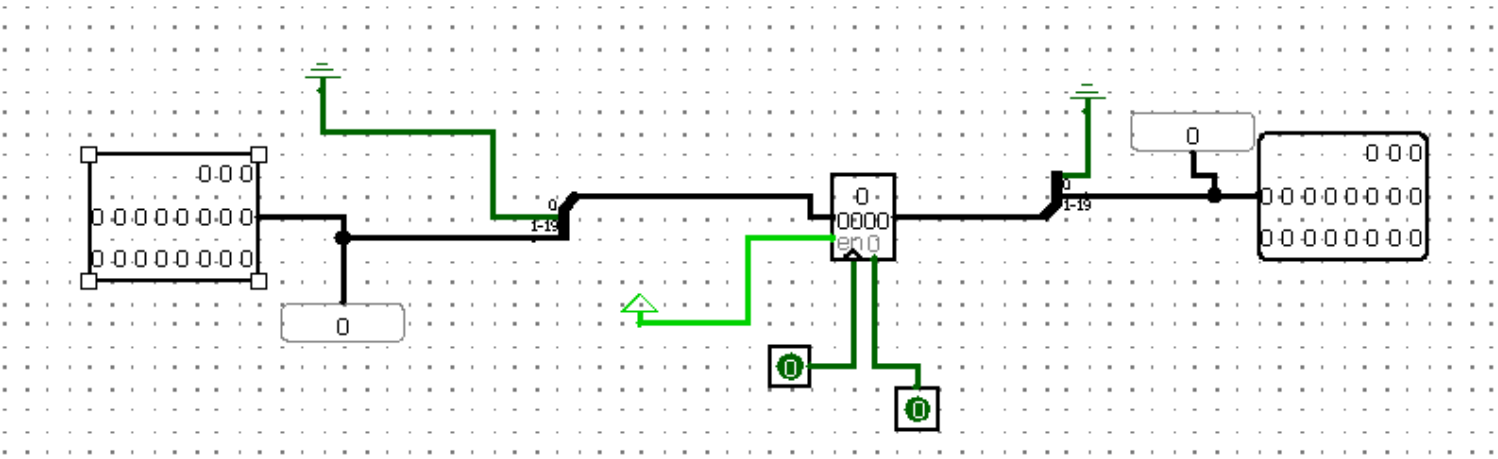


Figure6: PC

7-ALU Control

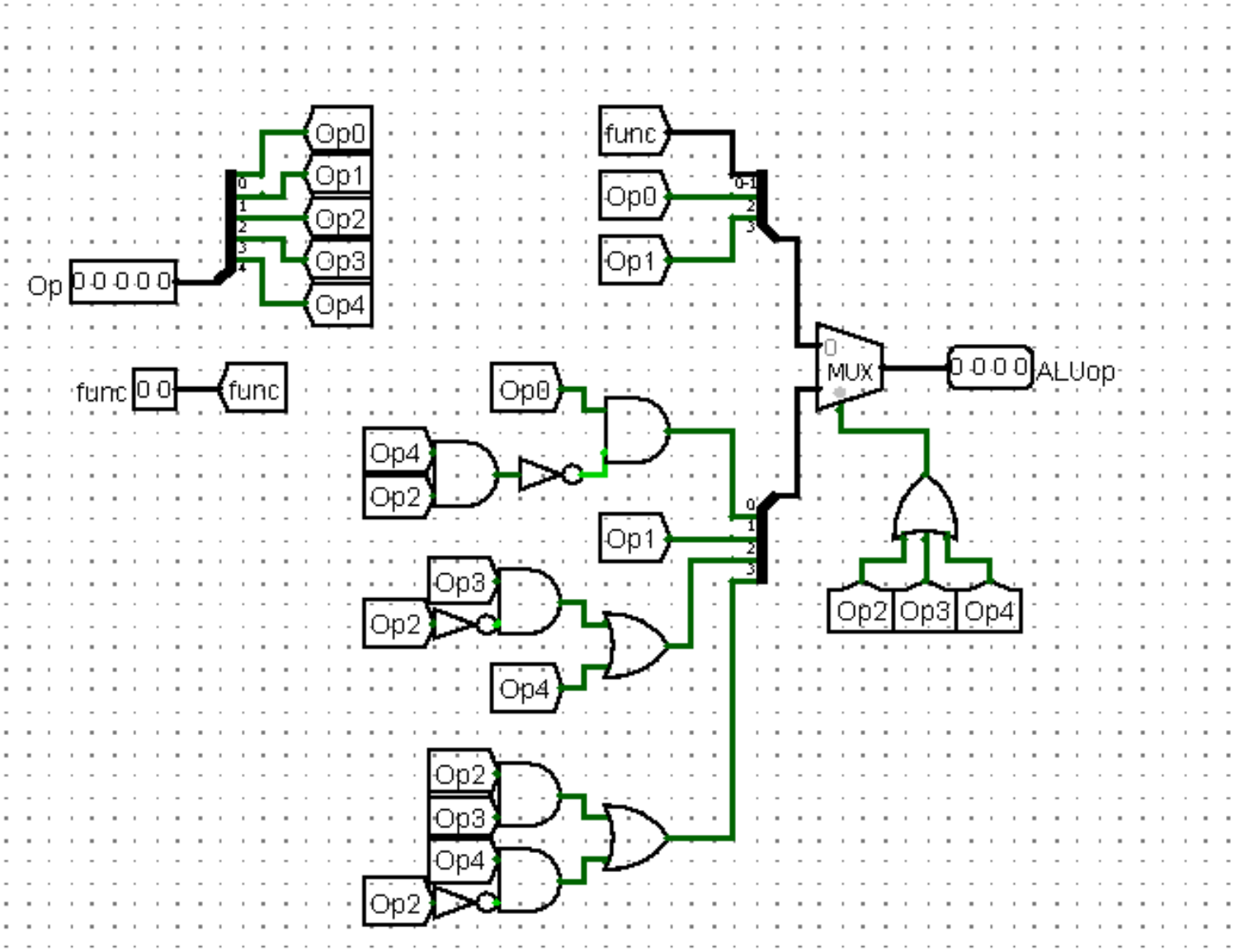


Figure 7: ALU Control.

8-ALU

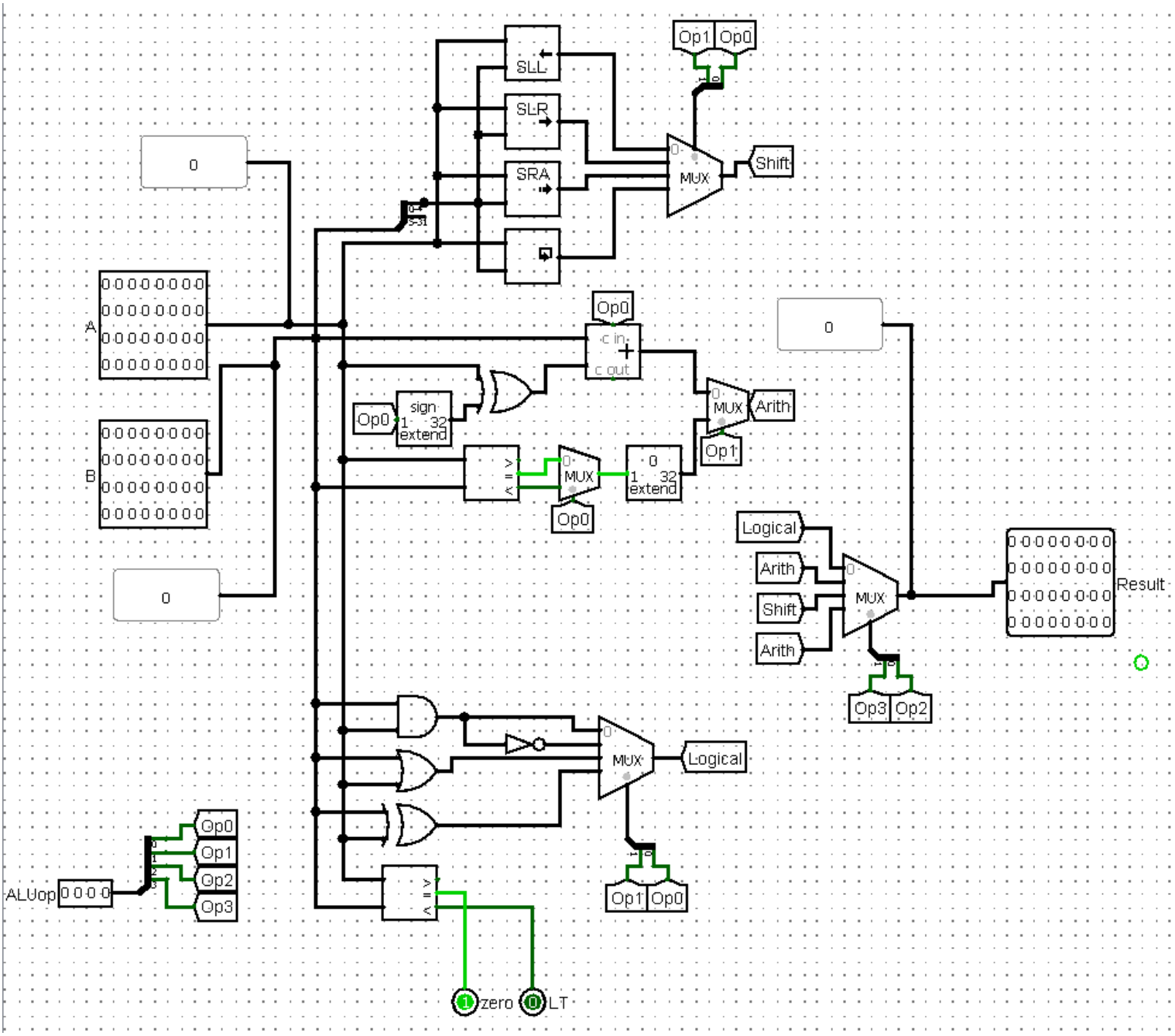
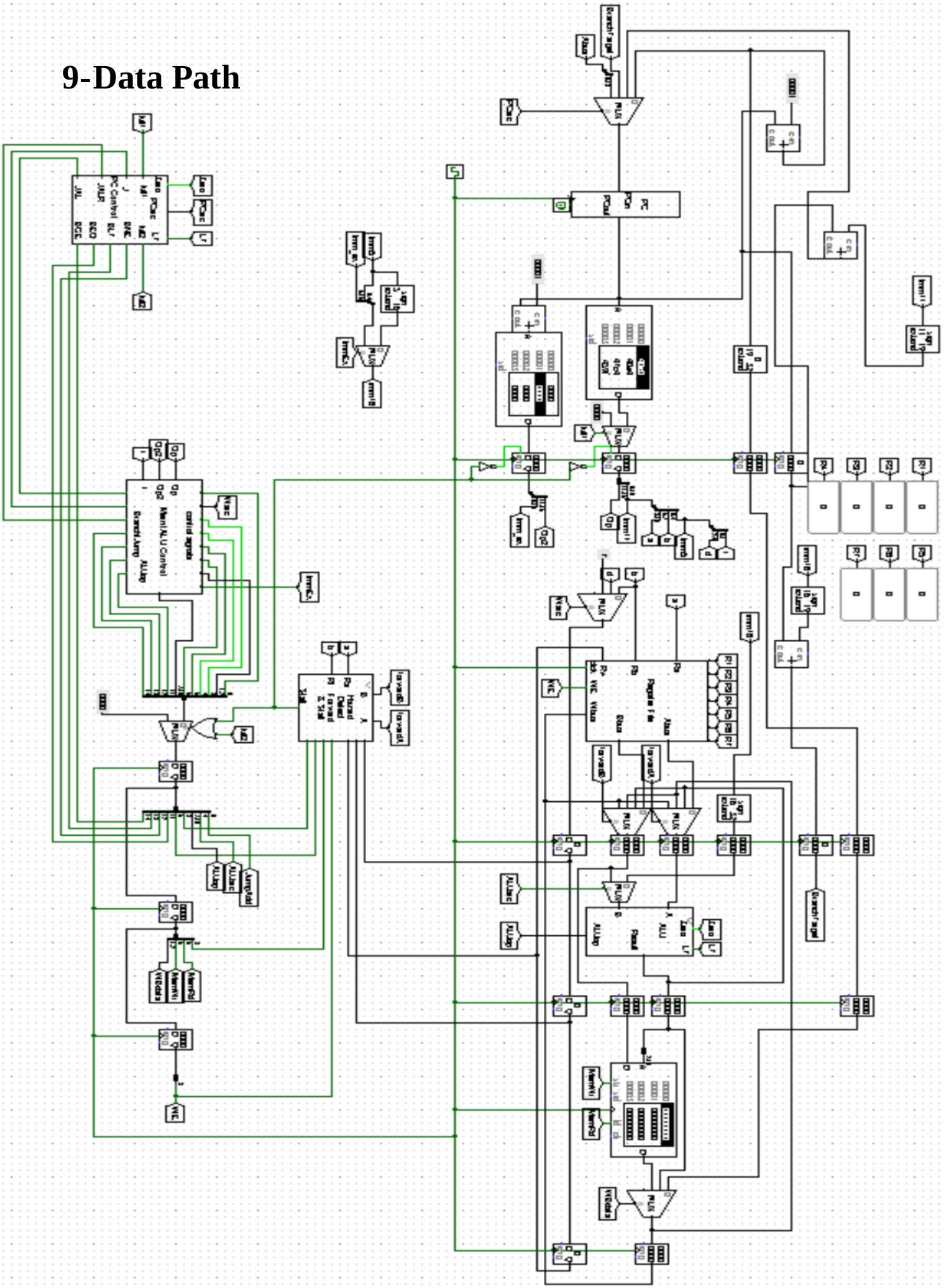


Figure 9: ALU.

9-Data Path



10- Main Control

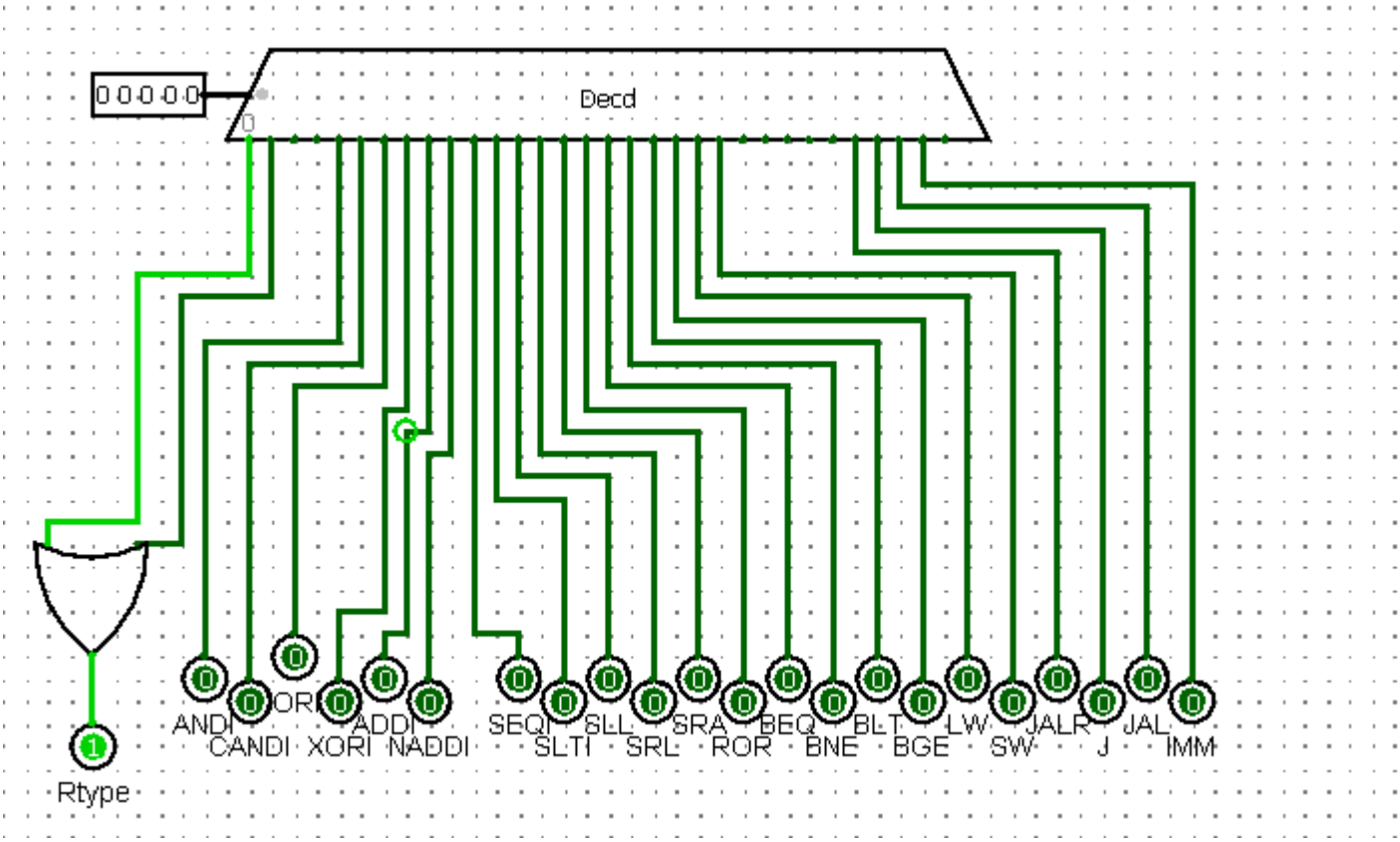


Figure10: Main Control

11- Main Control 2

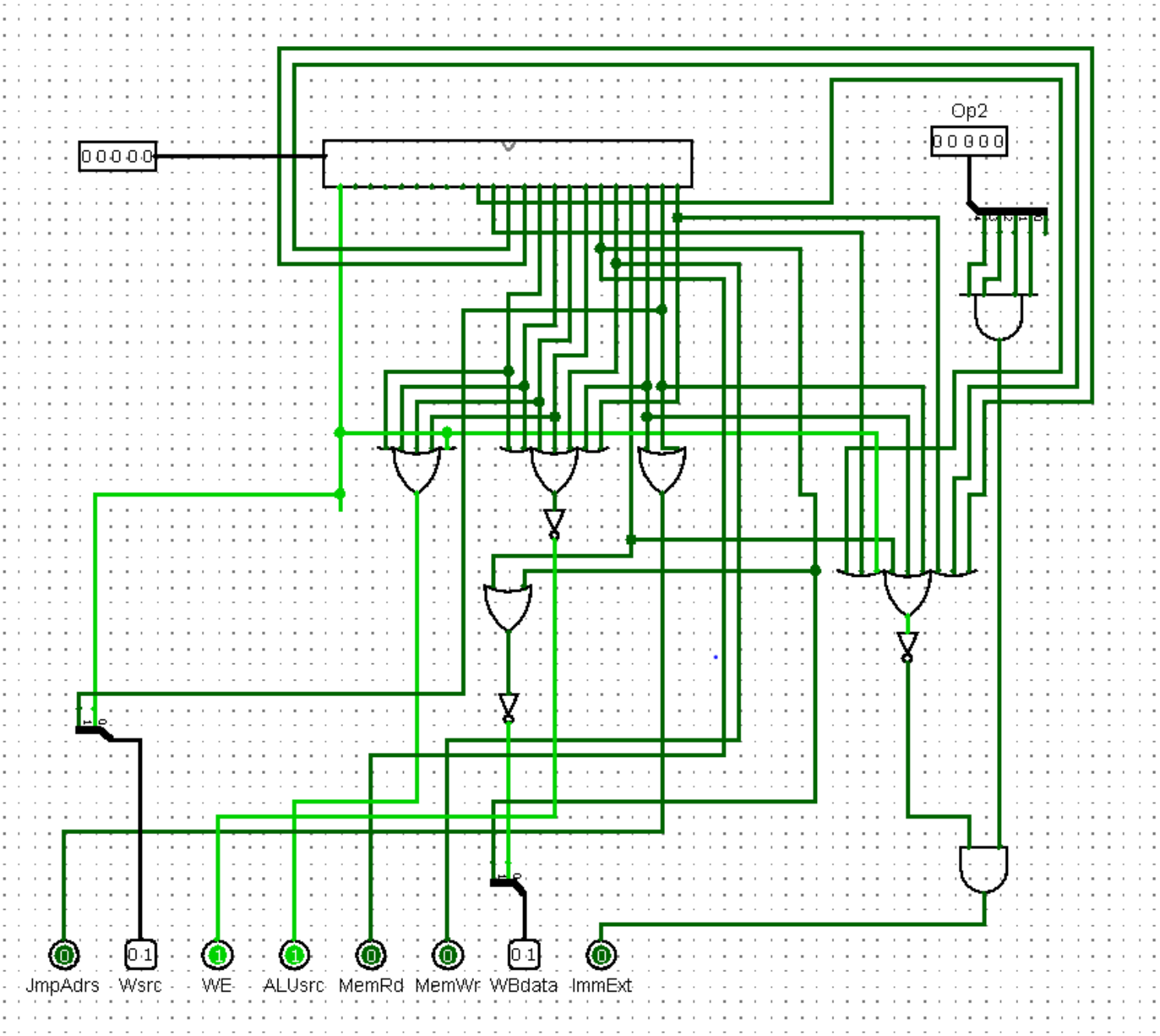


Figure11: Main Control.

12- Hazard Detect And Forwarding

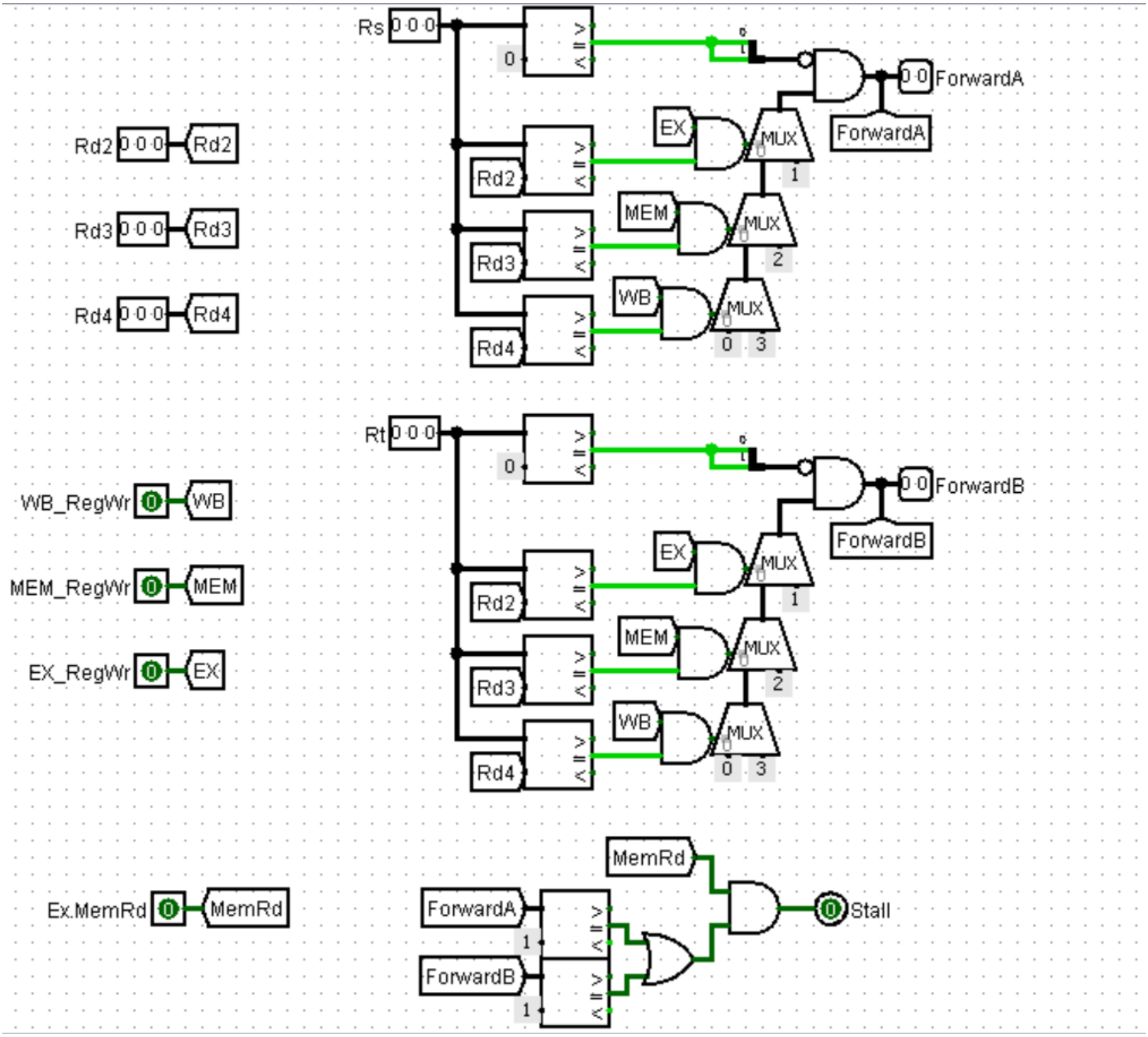


Figure 12: Hazard Detect & Forward.

Complete description of the control logic and control signals with a table and logic equations

Control Signals Table and Equation

Op	JmpAdrs	Wsrc	WE	ALUsrc	MemRd	Memwr	WBdata	ImmExt
R-type	x	1	1	1	0	0	1	x
ANDI	x	0	1	0	0	0	1	1
CANDI	x	0	1	0	0	0	1	1
ORI	x	0	1	0	0	0	1	1
XORI	x	0	1	0	0	0	1	1
ADDI	x	0	1	0	0	0	1	1
NADDI	x	0	1	0	0	0	1	1
SEQI	x	0	1	0	0	0	1	1
SLTI	x	0	1	0	0	0	1	1
shift	x	0	1	0	0	0	1	1
Branch	0	0	0	1	0	0	x	1
LW	x	0	1	0	1	0	2	1
SW	x	x	0	0	0	1	x	1
JALR	x	0	1	x	0	0	0	x
J	1	x	0	x	0	0	x	x
JAL	1	2	1	x	0	0	x	x
IMM	x	x	0	x	0	0	x	x
Equation	J+JAL	[JAL, Rtype]	!(Branch+SW+J+IMM)	Rtype+Branch	LW	SW	[LW, i(JALR+LW+JAL)]	
	ImmExt=	Op=!(Rtype+JALR+J+JAL+IMM)*op2=(IMM)						

2.Testing Simulation and Programs

A.Selection Sort

This programs sorts

#	Instruction	Hexadecimal
1.	**addi R7, R0, 16**	40e8
2.	**addi R5, R0, 4**	40a4
	Loopi:	
3.	**addi R6, R7, 4**	47C1
4.	**addi R4, R5, -1**	459F
	Loopj:	
5.	**lw R1, 0(R7)**	A720
6.	**lw R2, 0(R6)**	A640
7.	**blt R1, R2, skip:**	AF40
8.	**sw R2, 0(R7)**	AE20
9.	**sw R1, 0(R6)**	E003
	j update	
	skip:	
10.	**sw R1, 0(R7)**	AF20
11.	**sw R2, 0(R6)**	AE40
	update:	
12.	**addi R6, R6, 4**	46C1
13.	**addi R4, R4 -1**	449F
14.	**Blt R0,R4,loopj**	9097
15.	**addi R7, R7, 4**	47E1
16.	**addi R5, R5, -1**	45BF
17.	blt R0, R5, loopi	90B2

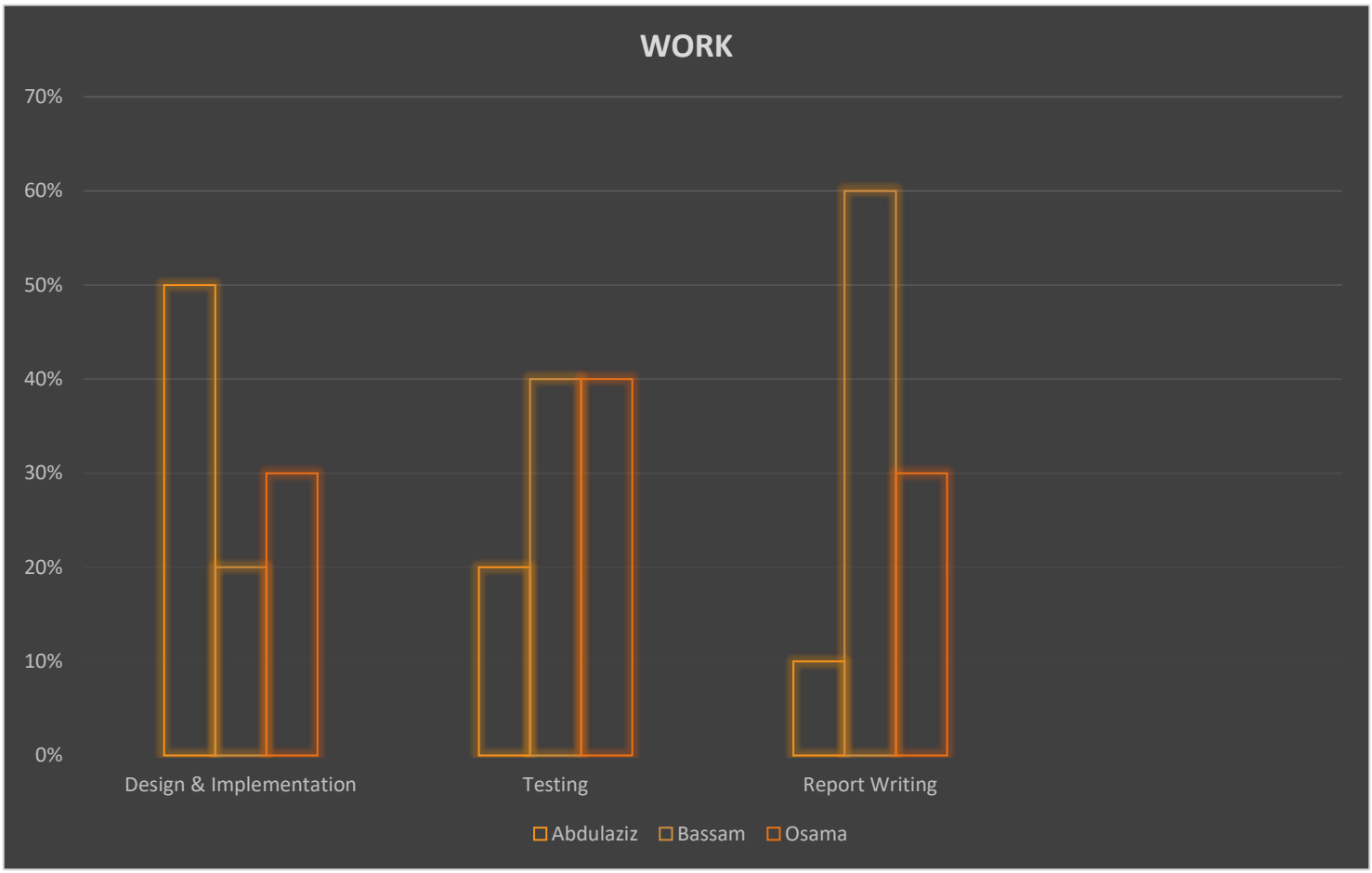
B. instruction

This program tests all instructions

#	instruction	Hexadecimal	Comment/Remark
1.	**addi R1 R0 3**	**4023**	**R1 will have the value = 3**
2.	**addi R2 R0 1**	**4041**	**R2 will have the value= 1**
3.	**add R3 R1 R2**	**094c**	**R3 will have the value= 4**
4.	**or R3 R1 R2**	**014e**	**R3 will have the value= 3**
5.	**xor R3 R1 R2**	**014f**	**R3 will have the value= 2**
6.	**add R3 R1 R2**	**094c**	**R3 will have the value= 4**
7.	**nadd R3 R1 R2**	**094d**	**R3 will have the value= -2**
8.	**seq R3 R1 R1**	**092e**	**R3 will have the value= 1**
9.	**slt R3 R2 R1**	**0a2f**	**R3 will have the value= 1**
10.	**andi R3 R1 5**	**2165**	**R3 will have the value= 1**
11.	**candi R3 R1 5**	**2965**	**R3 will have the value= = - 2**
12.	**ori R3 R1 5**	**3165**	**R3 will have the value= 7**
13.	**xori R3 R1 5**	**3965**	**R3 will have the value= 6**
14.	**addi R3 R1 5**	**4165**	**R3 will have the value= 8**
15.	**naddi R3 R1 5**	**4965**	**R3 will have the value= 2**
16.	**seqi R3 R1 5**	**5165**	**R3 will have the value= 0**
17.	**slti R3 R1 5**	**5965**	**R3 will have the value= 1**
18.	**sll R3 R1 1**	**6161**	**R3 will have the value= 6**
19.	**srl R3 R1 1**	**6961**	**R3 will have the value= 1**
20.	**sra R3 R1 1**	**7161**	**R3 will have the value= 1**
21.	**ror R3 R1 1**	**7961**	**R3 will have the value= -2147483647**
22.	**bne R1 R2 L2**	**8943**	**It will take branch to L2, because R1 (3) not equal to R2 (1).**
23.	**add R0 R0 R0**	**0800**	**Dummy instruction**

24.	**add R0 R0 R0**	**0800**	**Dummy instruction**
25.	**L2:**		
26.	**blt R1 R2 L3**	**9143**	**It will not take branch to L2, because R1 (3) not less than R2 (1).**
27.	**add R0 R0 R0**	**0800**	**Dummy instruction**
28.	**add R0 R0 R0**	**0800**	**Dummy instruction**
29.	**L3:**		
30.	**bge R1 R2 L4**	**9943**	**It will take branch to L2, because R1 (3) greater than R2 (1).**
31.	**add R0 R0 R0**	**0800**	**Dummy instruction**
32.	**add R0 R0 R0**	**0800**	**Dummy instruction**
33.	**L4:**		
34.	**sw R1 1(R2)**	**aa21**	**R1 will be stored in data memory**
35.	**lw R3 1(R2)**	**a261**	**R3 will have the value= 3**
36.	**addi R1 R1 1** **imm 100**	**4121** **f064**	**R1 will have the value = ١٦٠٠**
38.	**j L5**	**e002**	**New address= address +2**
39.	**add R0 R0 R0**	**0800**	**Dummy instruction**
40.	**L5:**		
41.	**jal L6**	**e802**	**R7= address** **New address= address +2,**
42.	**add R0 R0 R0**	**0800**	**Dummy instruction**
43.	**L6:**		
44.	jalr R1 R0	d900	**R1=address+1** **New address= 0**

3.Teamwork



Name	Job
Abdulaziz Alshehri	**Data Path, Arrange Pipelined Components in Data Path, Main Control**
Osama Bujwaied	**ALU, ALU Control, Hazard Detect Forward and Stall, Sort algorithm**
Bassam AlDossary	**PC control, Project Report Writing, Testing**