Homework 4 Answer Sheet

Please state the name, SID and email of each member of your group.

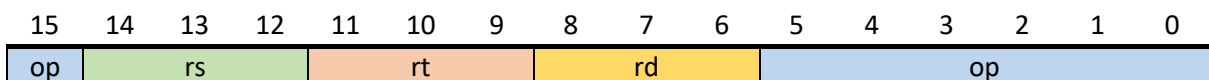| member	name	SID	email
#1 (contact person)	YUNG Kai Sen	59914603	kaisyung2-c@my.cityu.edu.hk
#2	YUN Tsz Sun	59376199	tsyun2-c@my.cityu.edu.hk
#3	TANG Wai Nok	59400192	waintang3-c@my.cityu.edu.hk

Logisim Version: <u>2.7.1</u>

OS (Window/MAC/Linux): <u>Windows 11 Pro 25H2</u>

A. Do all members make significant contributions to this homework? If not, please specify the details.

   <u>Yes</u>
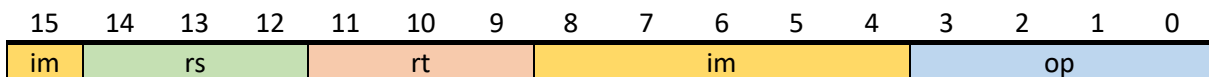
B. Please explain how many types of instructions are supported in your processor, and explain the format of each type of instructions (e.g., which bits are used as the operation or function code, which bits are used to index the 1st, 2nd or 3rd operand, and which bits are used to store the immediate number). You can draw figures to better explain your answer.
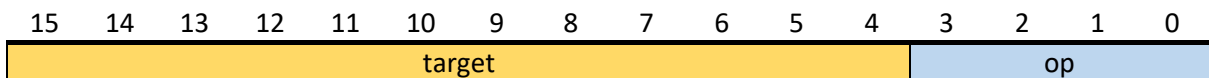
R-Type:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
op	rs	rs	rs	rt	rt	rt	rd	rd	rd	op	op	op	op	op	op

I-Type:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
im	rs	rs	rs	rt	rt	rt	im	im	im	im	im	op	op	op	op

J-Type:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
target												op	op	op	op

C. Please explain the format of each instruction (including the format of this instruction and its operation codes, and other information if needed).

li	Type: I-Type
	Opcode: 0000
	Format:
		Immediate 15	R2 14-12	R1 11-9	Immediate 8-4	Opcode 3-0	
	Example:
	li $r1, 1
	Binary: 0 001 000 00001 0000
	Hex: 1010
	li $r3, 10

Binary: 0 011 000 01010 0000
Hex: 30A0

add	Type: R-Type
	Opcode: 0001
	Format:

Opcode 15	R2 14-12	R1 11-9	Register Destination 8-6	Opcode 5-0

Example:
add $r2, $r1, $r2
Binary: 0 010 001 010 000001
Hex: 2281

and	Type: R-Type
	Opcode: 0010
	Format:

Opcode 15	R2 14-12	R1 11-9	Register Destination 8-6	Opcode 5-0

Example:

and $r3, $r1, $r2
Binary: 0 010 001 011 000010
Hex: 22C2

or	Type: R-Type
	Opcode: 0011
	Format:

Opcode 15	R2 14-12	R1 11-9	Register Destination 8-6	Opcode 5-0

Example:
or $r4, $r1, $r2
Binary: 0 010 001 100 000011
Hex: 2303

load	Type: I-Type
	Opcode: 0110
	Format:

Immediate 15	R2 14-12	R1 11-9	Immediate 8-4	Opcode 3-0

Example:
load $r6, $r7
Binary: 0 110 111 00000 0100
Hex: 6E04

store	Type: I-Type
	Opcode:0101
	Format:

Immediate 15	R2 14-12	R1 11-9	Immediate 8-4	Opcode 3-0

Example:
store $r3, $r7
Binary: 0 011 111 00000 0101
Hex: 3E05

move	Type: I-Type
	Opcode:0110
	Format:

Immediate 15	R2 14-12	R1 11-9	Immediate 8-4	Opcode 3-0

Example:
move $r4, $r3
Binary: 0 100 011 00000 0110
Hex:4606

addi	Type: I-Type
	Opcode: 0111
	Format:

Immediate 15	R2 14-12	R1 11-9	Immediate 8-4	Opcode 3-0

Example:
Addi $r2 $r1, 1
Binary: 0 010 001 00001 0111
Hex: 2217

andi	Type: I-Type
	Opcode:1000
	Format:

Immediate 15	R2 14-12	R1 11-9	Immediate 8-4	Opcode 3-0

Example:
andi $r3, $r1, 3
Binary: 0 011 001 00011 1000
Hex: 3238

ori	Type: I-Type Opcode:1001 Format: | Immediate 15 | R2 14-12 | R1 11-9 | Immediate 8-4 | Opcode 3-0 | Example: ori $r4, $r3, 8 Binary: 0 100 011 01000 1001 Hex: 4689
ble	Type: I-Type Opcode:1010 Format: | Immediate 15 | R2 14-12 | R1 11-9 | Immediate 8-4 | Opcode 3-0 | Example: ble $r2, $r3, -1 Binary: 1 011 010 11111 1010 Hex: B5FA
bne	Type: I-Type Opcode:1011 Format: | Immediate 15 | R2 14-12 | R1 11-9 | Immediate 8-4 | Opcode 3-0 | Example: bne $r2, $r3, -1 Binary: 1 011 010 11111 1011 Hex: B5FB
jump	Type: J-Type Opcode: 1100 Format: | Immediate 15-4 | Opcode 3-0 | Example: jump 3 Binary: 000000000011 1100 Hex: 003C
call	Type: J-Type Opcode: 1101

	Format:
		Immediate 15-4	Opcode 3-0	
	Example:
	call 3
	Binary: 000000000011 1101
	Hex: 003D
rtn	Type: J-Type
	Opcode: 1101
	Format:
		Immediate 15-4	Opcode 3-0	
	Example:
	rtn
	Binary: 000000000000 1110
	Hex: 003E
halt	Type: -
	Opcode: 1111
	Format:
		Immediate 15-4	Opcode 3-0	
	Example:
	halt
	Binary: 000000000000 1111
	Hex: 000F

D. Fill the following tables with the machine codes of each instruction of the testing programs:

Test program 1:

instruction	machine code (binary)	machine code (hex)
li $r1, 1	0001000000010000	1010
li $r2, 2	0010000000100000	2020
li $r3, 10	0011000010100000	30A0
add $r2, $r1, $r2	0010001010000001	2281
ble $r2, $r3, -1	1011010111111010	B5FA
halt	0000000000001111	000f

Test program 2:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	0001000001100000	1060
li $r2, 5	0010000001010000	2050

andi $r3, $r1, 3	0011001000111000	3238
ori $r4, $r3, 8	0100011010001001	4689
halt	0000000000001111	000F

Test program 3:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	0001000001100000	1060
li $r2, 5	0010000001010000	2050
and $r3, $r1, $r2	0010001011000010	22C2
li $r8, 0	0111000000000000	7000
store $r3, $r8	0011111000000101	3E05
or $r4, $r1, $r2	0010001100000011	2303
li $r8, 1	0111000000010000	7010
store $r4, $r8	0100111000000101	4E05
li $r8, 1	0111000000010000	7010
load $r7, $r8	0110111000000100	6E04
halt	0000000000001111	000F

Test program 4:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	0001000001100000	1060
li $r2, 4	0010000001000000	2040
call 7	0000000001111101	007D
move $r4, $r3	0100011000000110	4606
li $r1, 7	0001000001110000	1070
li $r2, 8	0010000010000000	2080
call 3	0000000000111101	003D
move $r5, $r3	0101011000000110	5606
jump 3	0000000000111100	003C
add $r3, $r1, $r2	0010001011000001	22C1
rtn	0000000000001110	000E
halt	0000000000001111	000F

Test program 5:

instruction	machine code (binary)	machine code (hex)
li $r1, 2	0001000000100000	1020
addi $r2 $r1, 1	0010001000010111	2217
halt	0000000000001111	000F