Homework 4 Answer Sheet

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

member	name	SID	email

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

Yes, everyone has made equal contribution to the homework.

B. Which version of Logisim was used for your design of the circuits?

Logisim-evolution v3.6.1 on Windows 11.

C. 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.

There are 3 types of instructions supported: R-type, I-type and J-type. Formats are listed below:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R-Type	opcode				rs			rt			rd			funct		
I-Type	opcode				rs			rt			immediate					
J-Type	opcode				target											

Note: **opcode** = operation code, **rs** = 1st operand, **rt** = 2nd operand, **rd** = 3rd operand, **funct** = function code (for arithmetic instructions), **immediate** = immediate number, **target** = immediate number as address.

Some specific instructions do not strictly follow the given format. They are listed below:

R-Type	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
move	opcode				*unused*			rt			rd			*unused*		

I-Type	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
li	opcode				*unused*			rt			immediate					
load	opcode				rs			rt			*unused*					
store	opcode				rs			rt			*unused*					

I-Type	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rtn	opcode				*unused*											
reboot	opcode				*unused*											
halt	opcode				*unused*											
jump	opcode				*unused*						immediate (target)					
call	opcode				*unused*						immediate (target)					

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

Note: **s** = operand 1, **t** = operand 2, **d** = destination, **i** = immediate, **x** = unused

	format	opcode	funct
li	1101 xxx ddd iiiiii	1101	
add	0000 sss ttt ddd 000	0000	000
and	0000 sss ttt ddd 010	0000	010
or	0000 sss ttt ddd 011	0000	011
neg	0000 sss xxx ddd 001	0000	001
load	1110 sss ddd xxxxxx	1110	
store	1111 sss ddd xxxxxx	1111	
move	0100 xxx sss ddd xxx	0100	
addi	1000 sss ddd iiiiii	1000	
andi	1010 sss ddd iiiiii	1010	
ori	1011 sss ddd iiiiii	1011	
ble	1100 sss ttt iiiiii	1100	
slt	0101 sss ttt ddd xxx	0101	
lsl	0000 sss ttt ddd 110	0000	110
lsr	0000 sss ttt ddd 111	0000	111
jump	0001 xxxxxx iiiiii	0001	
call	0010 xxxxxx iiiiii	0010	
rtn	0011 xxxxxxxxxxxx	0011	
reboot	0110 xxxxxxxxxxxx	0110	
halt	0111 xxxxxxxxxxxx	0111	

E. 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	1101 0000 0000 0001	D001
li $r2, 2	1101 0000 0100 0010	D042
li $r3, 10	1101 0000 1000 1010	D08A
add $r2, $r1, $r2	0000 0000 0100 1000	0048
ble $r2, $r3, -1	1100 0010 1011 1111	C2BF
slt $r4, $r3, $r2	0101 0100 0101 1000	5458
halt	0111 0000 0000 0000	7000

Test program 2:

instruction	machine code (binary)	machine code (hex)
li $r1, 3	1101 0000 0000 0011	D003
li $r2, 5	1101 0000 0100 0101	D045
andi $r3, $r1, 3	1010 0000 1000 0011	A083
ori $r4, $r3, 8	1011 0100 1100 1000	B4C8
neg $r5, $r4	0000 0110 0010 0001	0621
lsl $r6, $r5, $r1	0000 1000 0010 1110	082E
lsr $r7, $r5, $r2	0000 1000 0111 0111	0877
halt	0111 0000 0000 0000	7000

Test program 3:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	1101 0000 0000 0110	D006
li $r2, 5	1101 0000 0100 0101	D045
and $r3, $r1, $r2	0000 0000 0101 0010	0052
li $r8, 0	1101 0001 1100 0000	D1C0
store $r3, $r8	1111 1110 1000 0000	FE80
or $r4, $r1, $r2	0000 0000 0101 1011	005B
li $r8, 1	1101 0001 1100 0001	D1C1
store $r4, $r8	1111 1110 1100 0000	FEC0
li $r8, 1	1101 0001 1100 0001	D1C1
load $r7, $r8	1110 1111 1000 0000	EF80
reboot	0110 0000 0000 0000	6000
halt	0111 0000 0000 0000	7000

Test program 4:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	1101 0000 0000 0110	D006
li $r2, 4	1101 0000 0100 0100	D044
call 7	0010 0000 0000 0111	2007
move $r4, $r3	0100 0000 1001 1000	4098
li $r1, 7	1101 0000 0000 0111	D007
li $r2, 8	1101 0000 0100 1000	D048
call 3	0010 0000 0000 0011	2003
move $r5, $r3	0100 0000 1010 0000	40A0
jump 3	0001 0000 0000 0011	1003

add $r3, $r1, $r2	0000 0000 0101 0000	0050
rtn	0011 0000 0000 0000	3000
halt	0111 0000 0000 0000	7000