Homework 4 Answer Sheet

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

member	name	SID	email
#1 (contact person)	Syed Aahmad Adnan	58286436	saadnan2-c@my.cityu.edu.hk
#2	Muhammad Sarbuland	57271806	msarbulan2-c@my.cityu.edu.hk
#3	Muhammad Haris Mumtaz	56742254	mhmumtaz2-c@my.cityu.edu.hk

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

Yes all the members contributed equally.

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

2.7.1

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.

All 20 instructions are supported(r-type, i-type, j-type). The first 5 bits are used as the opcode in all the instructions. The next 3 bits are used as the index of the write/destination register except for ble and store. After that the next 3 bits become the index of the read register 1 except for branch or store. The next 5 bits are used for the index of the read register 2 except for the ble and in other situations can possibly be ignored. The proper implementation of each instruction has been explained below as in some instructions such as store and load there are variations in the bits being ignored or being used as the entire index of the register. It is to be noted whenever we mention below that we are using more than 3 bits to represent a register you will be writing the register number in those number of bits. Moreover, the binary mapping for each register is binary of register index = binary of register index -1, such that r1 is mapped to 000 and r2 is mapped to 001 all the way to r8 which is mapped to 111. Thus as mentioned in the store instruction below when we ask you to represent the register index in 8 bits and you want to represent the r2 you will be writing 00000001 in the machine code. Viewing the detail below and the machine code illustrations should suffice.

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

li	Opcode: 01100 next 3 bits: destination register rest of the bits: Immediate number to be loaded
add	Opcode: 00000 next 3 bits: Destination register next 3 bits: Source register 1 rest 5 bits: Source register 2
and	Opcode: 00111 next 3 bits: Destination register next 3 bits: Source register 1 rest 5 bits: Source register 2
or	Opcode: 01001 next 3 bits: Destination register next 3 bits: Source register 1 rest 5 bits: Source register 2

neg	Opcode: 00010 next 3 bits: Destination register next 8 bits: Source register
load	Opcode: 01011 next 3 bits: Destination register rest 8 bits: Source register
store	Opcode: 01101 next 3 bits: read register for data coming to be written to memory rest 8 bits: Write register containing the address of the memory to be written.
move	Opcode:01110 next 3 bits: destination register next 3 bits: Source register rest of the bits to be ignored
addi	Opcode: 00001 next 3 bits: destination register next 3 bits: source register remaining 5 bits: Immediate number to be added with source register

andi	Opcode: 01000 next 3 bits: destination register next 3 bits: source register remaining 8 bits: immediate number used to perform AND operation with source register.
ori	Opcode: 01010 next 3 bits: destination register next 3 bits: source register rest 5 bits: immediate number to be compared with source register using OR.
ble	Opcode: 00110 next 3 bits: Source1 register next 3 bits: Source1 register rest 5 bits: jump to this address if source1 is less than or equal to source2
slt	Opcode: 00101 next 3 bits: Destination register to be set to 1 or 0 next 3 bits: Source register 1 rest 5 bits: Source register 2
lsl	Opcode: 00011 next 3 bits: Destination register next 3 bits: Source register 1 rest 5 bits: Source register 2
lsr	Opcode: 00100 next 3 bits: Destination register next 3 bits: Source register 1 rest 5 bits: Source register 2

jump	Opcode: 01111 rest 11 bits: For immediate number to be added to PC
call	Opcode: 10000 rest 11 bits: For immediate number to be added to PC
rtn	Opcode: 10001 rest 11 bits to be ignored
reboot	Opcode: 10010 rest 11 bits: Ignored
halt	Opcode: 10011 rest 11 bits: Ignored

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	0110000000000001	6001
li $r2, 2	0110000100000010	6102
li $r3, 10	0110001000001010	620a
add $r2, $r1, $r2	0000000100000001	0101
ble $r2, $r3, -1	0011000101011111	315f
slt $r4, $r3, $r2	0010101101000001	2b41
halt	1001100000000000	9800

Test program 2:

instruction	machine code (binary)	machine code (hex)
li $r1, 3	0110000000000011	6003
li $r2, 5	0110000100000101	6105
andi $r3, $r1, 3	0100001000000011	4203
ori $r4, $r3, 8	0101001101001000	5348
neg $r5, $r4	0001010000000011	1403
lsl $r6, $r5, $r1	0001110110000000	1d80
lsr $r7, $r5, $r2	0010011010000001	2681
halt	1001100000000000	9800

Test program 3:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	0110000000000110	6006
li $r2, 5	0110000100000101	6105
and $r3, $r1, $r2	0011101000000001	3a01
li $r8, 0	0110011100000000	6700
store $r3, $r8	0110101000000111	6a07

or $r4, $r1, $r2	0100101100000001	4b01
li $r8, 1	0110011100000001	6701
store $r4, $r8	0110101100000111	6b07
li $r8, 1	0110011100000001	6701
load $r7, $r8	0101111000000111	5e07
reboot	1001000000000000	9000
halt	1001100000000000	9800

Test program 4:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	0110000000000110	6006
li $r2, 4	0110000100000100	6104
call 7	1000000000000111	8007
move $r4, $r3	0111001101000000	7340
li $r1, 7	0110000000000111	6007
li $r2, 8	0110000100001000	6108
call 3	1000000000000011	8003
move $r5, $r3	0111010001000000	7440
jump 3	0111100000000011	7803
add $r3, $r1, $r2	0000001000000001	0201
rtn	1000100000000000	8800
halt	1001100000000000	9800