YOUR Group Details:

Group No:	20
Group Members (*Student IDs*):	Anay Praveen (20509910) Aravindh Palaniguru (20511833) Chee Jun Kit (20398665)

Justification of YOUR Circuitry Diagram Design:

(**NOTE**: No more than 300 words)

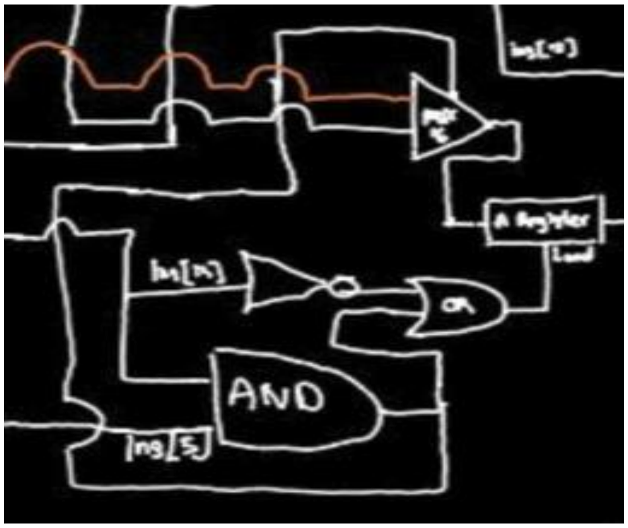

Figure1

Instruction decode(Figure1):

Instruction[15] -> A or C–instruction.

Load to register A(STORES ADDRESS):

(gates reasoning)

A register is loaded if it is an **Address instruction** since A register addresses memory location (**not** gate).

A register is loaded, if it is a **computation instruction and d1** is 1, for storing results in dest1.(**and** gate).

Either conditions above will load A register.(**or** gate)

From Mux:

A register is **loaded with ALU-output** from mux only if **C-instruction and instruction[5] (d1) is 1**(reason for **selector** for mux coming **from and gate**).

If **instruction[15] is 0(A-instruction) or d1 is not 1**, **instruction** taken into A register.

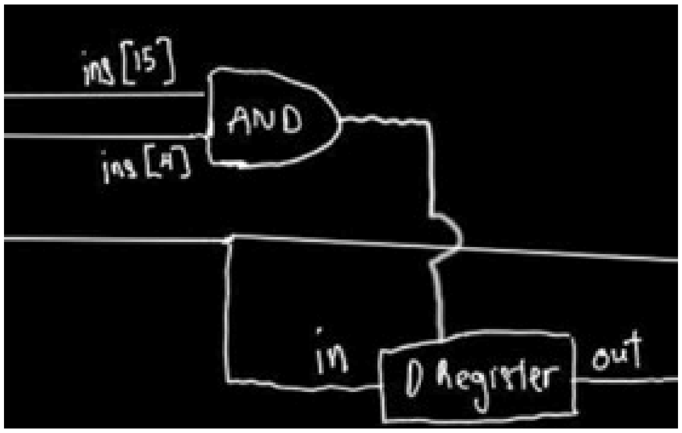

Figure2

D Register(Figure2):

If **instruction[15] and instruction[4] (d2) are 1**,**output** is given to **ALU as x** from D-register. (**AND** gate ensures that load to D register is 1 if it's a C-instruction and d2 is 1) .

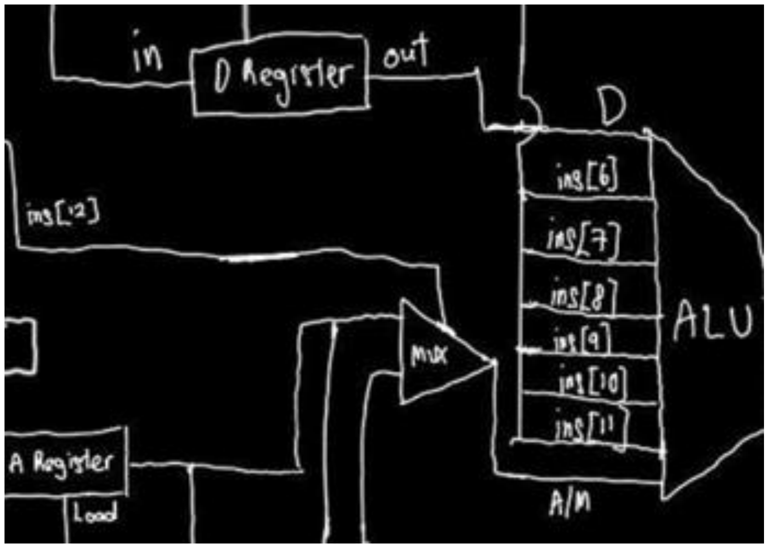

Figure3

MUX(Figure3):

Choose **inM as y to ALU**(perform computations in ALU) if **instruction[12](a)[comp] is 1**, **else** choose **output from A register.**

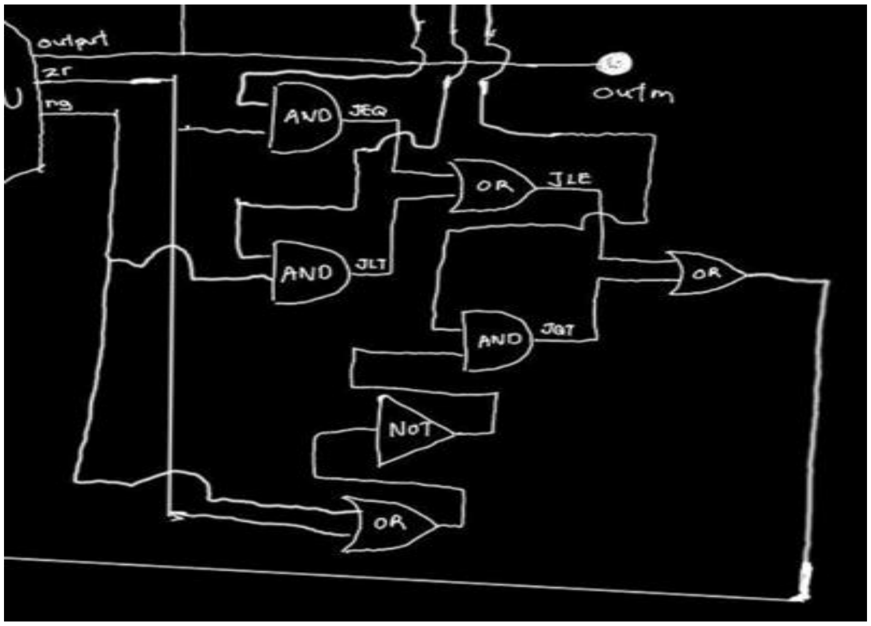

Figure4

JUMP(Figure4):

(jump to next instruction with address from A register in PC)

(reason for gates)

AND(JEQ)-if zr and instruction [1] (j2) are 1,output is 1(jump because ALUoutput is 0).

AND(JLT)-if ng is 1 and instruction 2(j1) is 1 output is 1(jump because ALUoutput is less than 0).

(JGT)-OR gate is used to check if either zr or ng is true. NOT gate is to invert the bit(jump if (not zr or not ng) and instruction[0](j1)is 1).

OR(JLE)-either JEQ or JLT is true.

OR(JLE, JGT)-jump if either JLE or JGT is true.

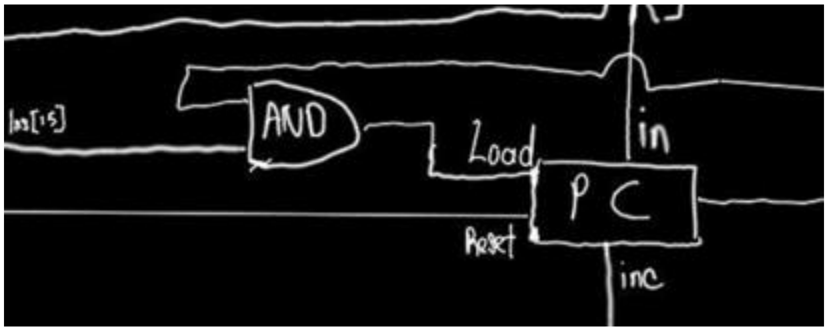

Figure5

PC:(Figure5)

Input into PC from A register loaded if both **C-instruction and jump is true**(**AND** gate reason).

If reset=1, PC=0(start over from address 0).

Inc always true in PC to go to next instruction.

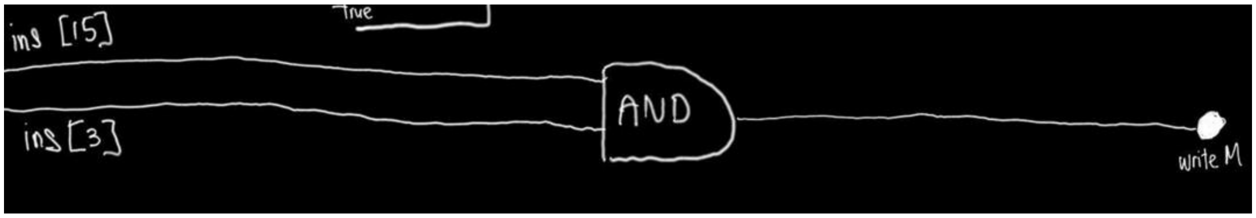

Figure6

writeM(Figure6):

Write to memory if **instruction[15] and instruction 3(d3) are 1**.