# HW #3

1. (a) We will change register B process of 4 entrances. We will connect the entrance 2 of the MUX to the output Read Data 2 to the single register. We will connect the control line jr, so the jr can decide which entrance goes into the PC.

(b) The values of the control lines for the new instructions will be:

RegDst = x

Branch = x

MemtoReg = x

ALUop = x

MemRead = 0

ALUSrc = x

RegWrite = 0

MemToRead = x

Jr = 1

(c) If we added a new control line, the value for all the other instructions will be jr = 0

2. (a) There is no need for any physical changes to be made

(b) The values of the control lines for the new instructions will be:

Reg dst = 0

Jump = 0

branch = 0

MemRead = 1

MemtoReg = 1

ALUop = 00

MemWrite = 0

ALUSrc = 0

RegWrite = 0

(c) no need to add a control line

(d) step 1: Fetch

step 2: Decode

step 3: the control line ALUSrc = 0 causes that the MUX will Pick

inside the Alt and will transfer it to the other entrance of the ALU according to the control chang ALUOp = 00, therefore ALUctrl will be 010. Add will be a connection between 2 registers.

step 4: the control line memRead is equal to 1, so data memory will perform operations with data it receives.

step 5: the control line memtoReg is equal to 1, so the Mux will choose the path indicated with the data memory (step 4). The data will be transmitted to the register. RegDst = 1, so the mux will change the register Rd and will transmit it to the control line Write Register (regWrite = 1). The data in WriteData will hence be written in the writeRegister.

(3) The value on the control lines for the new instruction will be:

RegDst - 0

branch - 0

MemtoReg - 0

ALUop - 0

MemWrite - 0

ALUsrc - 1

regwrite - 1

memread - 0

(4) (a) we need the Mux-A to have 4 entries, and one of them will have a path to the register 31-rg. We also need to add a new control line, MemToReg1.

(b) The values on the control lines for the new instruction will be:

regDst - 0

reg dst1 - 1

jump - 1

branch - 0

memToReg - 0

ALUop - xx

MemWrite - 0

ALUsrc - x

RegWrite - 1

memToReg1 - 1

MemRead - 0

(c) The value for new instruction will be 0 for jump, MemToReg1 = regDst 1.

(5) (a) It would be similar to the question 4, because we need to calculate the address that will be returned to -ra. Therefore, we will put 4 entries to Mux-D, and in one of the entries there would be PC+4 with a path to an add. We add a new control line that will be the MSB of memToReg1 = 1. The Mux-D will have 4 entries, with one of them being the new control line regDst 1 = 1. One of the entries will be the register 31 - ra.

(b) The values on the control lines for the new instruction will be:

regDst - 0
regDst 1 - 1
branch - 1
memToReg - 0
memToReg 1 - 1
ALU op - 01
MemWrite - 0
ALUsrc - 0
regWrite - 0
MemRead - 0

(6) (a) For example, beq (I type instruction) will not function, because the 0 will always be a 1 due to the flaw. Therefore, the 2 registers will not be able to correctly compare the data and the command will fail.

(b) Instructions such as add (J type) and jal (R type) would not be adversely affected by this flaw.

(c) without the flaw: $1 = 1, $2 = 2 $3 = 105
with the flaw: $1 = 1, $2 = 2, $3 = 105