Com S 321　　Homework 8　　Due: October 30, 2017 before midnight

1. For the MIPS single cycle Datapath, complete the following Control Table with entries 0, 1, or D for Don't care. Be very careful when using a Don't care control signal to avoid unintended consequences. The instructions appear in the column headings and the control variables are in the rows of the table. Assume that the BEQ instruction is a branch that is taken. The last row for the ALUOp has been completed for you with the name of the operation done in the ALU.

	Add	**Sub**	**LW**	**SW**	**BEQ** (Taken)	**Addi**
RegDst						
ALUSrc						
MemToReg						
RegWrite						
PCSrc						
MemWrite						
MemRead						
ALUOp	Add	Sub	Add	Add	Sub	Add

2. The format of a proposed **NewLW** instruction in an R-type-like format is as follows:

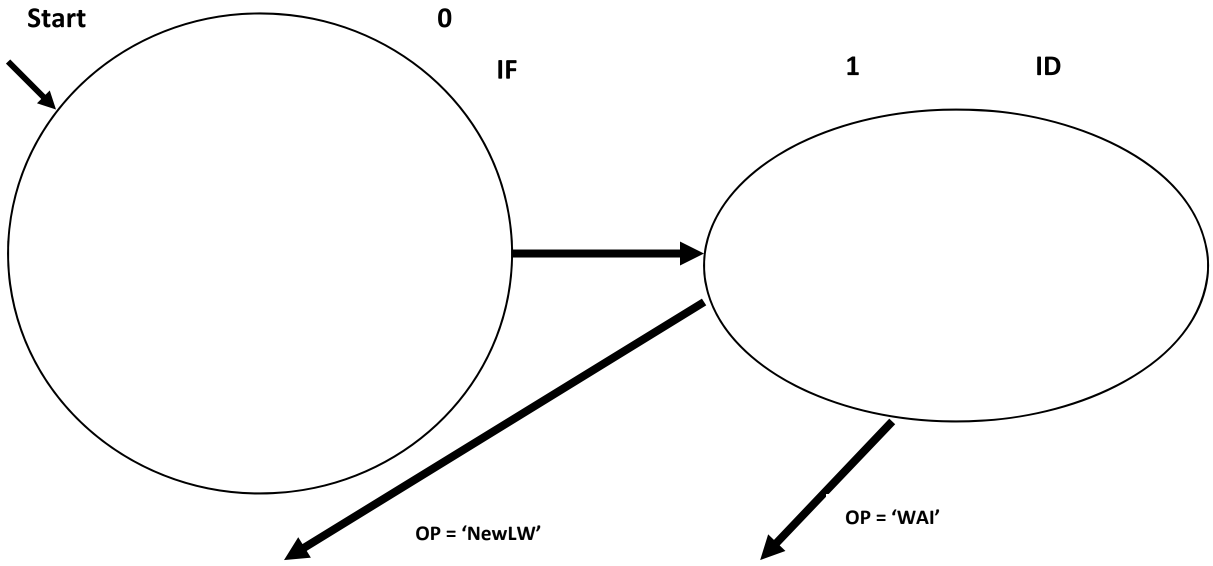
NewLW　rd, rs (rt)　　Opcode　rs　rt　rd　..　# R[rd] ← MEM [R[rs] + R[rt]]
6 bits　5 bits 5 bits　5 bits

What changes, if any, would you make to the **single cycle** datapath to execute the **NewLW** instruction?

Indicate the necessary changes and also specify the values of the multiplexor variables RegDst, AluSrc, PCSrc, and MemToReg, and the values of the RegWrite, MemRead, and MemWrite control variables for this instruction.

3. For the MIPS **multi-cycle** datapath, draw the finite state machine that represents the control necessary for the execution of the instructions **NewLW** described in problem **2,** and the instruction **WhereAmI** (WAI) which does the following: It puts the instruction's location, that is, the value of the PC when the instruction was called into a register specified by the **rt** field (i.e., bits [20…16]).

You should **number** all the states, **label** the states with actions taken from the set {IF, ID, EX, MEM, WB} and also indicate in each state the values of all the relevant multiplexor variables and control signals. Complete the diagram that has been started below.



4. We wish to add a new instruction **ADD3** to the MIPS multicycle datapath. This instruction adds the contents of 3 registers instead of two. The format of the instruction is as follows:

ADD3 rd, rs, rt, ru # R [rd] ← R [rs] + R [rt] + R [ru]

Registers **rs**, **rt**, and **rd** are specified as in the R-type Add instruction. Register **ru** is specified in bits [4–0] of the instruction word.

(a) Describe the changes you would make to the multicycle datapath to execute the **ADD3** instruction. You are not allowed to add a new ALU and you are not allowed to add new read ports to the register file. Your implementation should execute this instruction in the fewest number of clock cycles.

(b) Draw the complete finite state machine that represents the control necessary for the execution of the instruction **ADD3**.