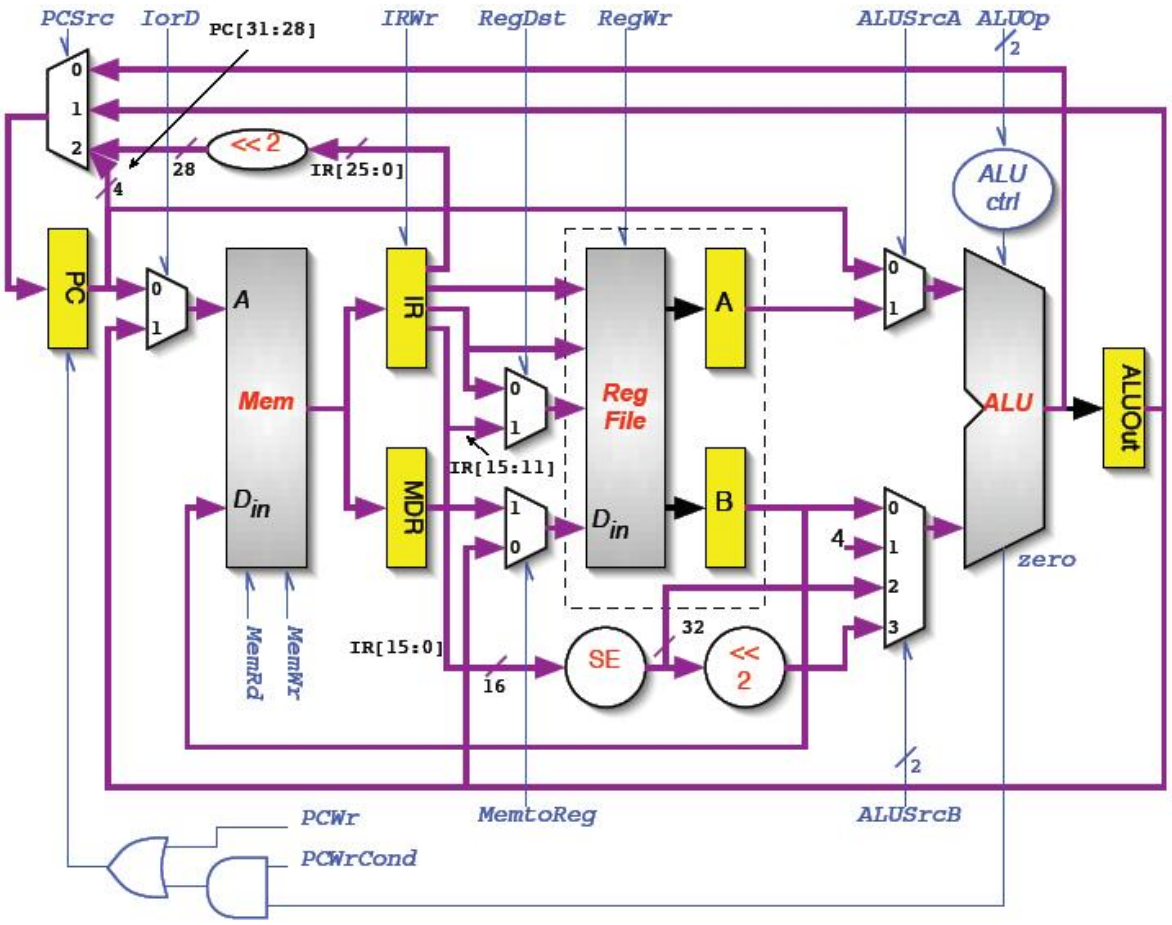
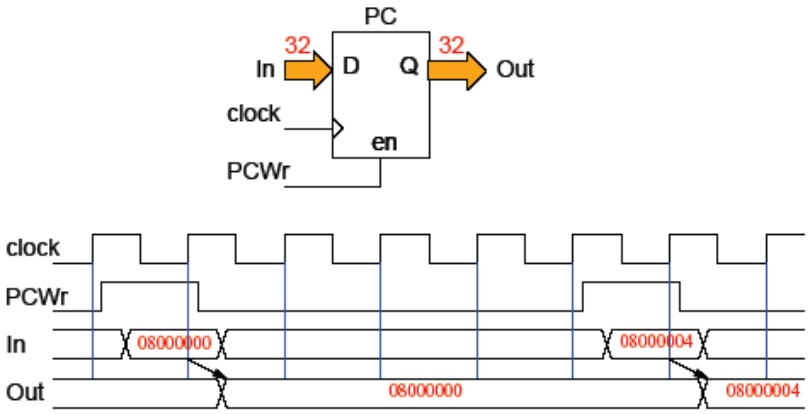
ECE 30
Introduction to Computer Engineering

Example Problems, Set #8
Spring 2014

1.



Shown above is a *multi-cycle* CPU. There are six registers in this datapath: `PC`, `IR`, `MDR`, `A`, `B`, and `ALUOut`. Of these, `PC` and `IR` are *enabled to change* when `PCWr` and `IRWr` are high (logic "1") respectively. These control signals `PCWr` and `IRWr` turn high shortly after the sampling (rising) edge of the clock, as the control-FSM changes its state. Therefore, the outputs of these registers do not change until the next sampling edge of the clock after the enabling control signals are asserted, as shown in the following timing diagram. [Note: The "in" and "out" in the timing diagram correspond to the input to and the output from the PC.]

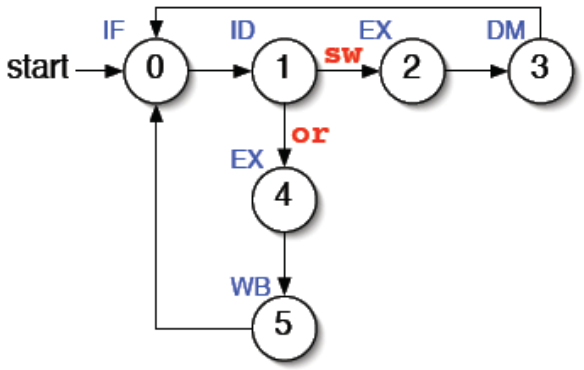


Assume that:

- `ALUOp` = 00 signals the ALU to *add* `A` and `B`.
  `ALUOp` = 01 signals the ALU to *subtract* `B` from `A`.
  `ALUOp` = 10 signals the ALU to do the operation specified by the R-format function code.

- `PC+4` and the *branch target* (`PC+4 + 4*offset`) are computed in the IF and ID/RF stages respectively by the ALU. *Note that these are computed regardless of the instruction type.*

Complete the state table below for the instructions `sw` and `or`. The state diagram for these instructions is also shown below.



	0	1	2	3	4	5
`IorD`						
`MemRd`						
`MemWr`						
`IRWr`						
`ALUSrcA`						
`ALUSrcB`						
`ALUOp`						
`PCSrc`						
`PCWr`						
`PCWrCond`						
`RegWr`						
`RegDst`						
`MemtoReg`						
	IF	ID/RF	EX	DM	EX	WB

2. Compute the number of clock cycles it takes to "execute" the following instructions in a *multi-cycle* implementation, assuming that each of the following phases takes exactly one clock cycle:

   IF: fetch instruction; compute `PC+4`
   ID/RF: decode instruction; fetch source operands; compute branch target
   EX: perform arithmetic / logic operation; compute data memory address; compute branch condition and determine the next PC; determine jump address
   DM: data memory access
   WB: write result to register file

   Complete the table below. Write the phases in the correct order.

Instruction type	No. of clock cycles	Phases (IF, ID/RF, EX, DM, WB)
`add`		
`lw`		
`sw`		
`or`		