HW3 Solution & Common Mistakes

1. ALUSrc: 1
   RegDst: 0
   ALUOp: X or 11 or 10
   MemWrite: 0
   MemRead: 0
   Branch: 0
   MemtoReg: 0

2. Adder for Branch target calculation

3. Data Memory

4. Load instruction (5')
   1150ps 1/1150ps or about 869.6MHz (5')
   The instruction with longest latency is lw. For lw, the critical path in this question is ***Inst => Reg => ALU => Data =>Mux11T0ps*** *(For the first input of ALU, it comes from* ***Inst => Reg = 400+180****; for the second, it comes from max{****Inst => Control => MUX = 400+100+80****,* ***Inst => SignExt => MUX = 400+50+80****} =* ***Inst => Control => MUX = 400+100+80****, and the control sign comes from* ***Inst => Control => ALU Control = 400+100+POO****.*
   Therefore, the time for ALU to get all input is 600ps. In addition, in Reg the decoder's output is ready, so the data is stored as long as the write-back data is ready. We consider that process as zero latency.)
   You will not get deduction if your solution is 1330ps.

5. Sample answer:

   addi $t0, $0, 0x0 8000, $t0 will stay 0 if there is a stuck-at-0 fault

   no example -5'

6. Mux "RegDst" (before reg): x
   Mux "ALUSrc" (after reg): 0000 0000 0000 0000 0000 0000 0001 0100 or 0x14 or 20 Mux "Mem2Reg" (after data memory): x
   Mux "Branch" and mux "Jump": PC+4
   ALU Read data 1: 0x4
   ALU Read data 2: 0000 0000 0000 0000 0000 0000 0001 0100 or 0x14 or 20
   Add after PC: PC, 4
   Add ALU result: PC+4, 0000 0000 0000 0000 0000 0000 0101 0000 or 0x50 or 80 Registers
   Read register 1: 00011 or 3
   Registers Read register 2: 00010 or 2

Registers Write register: x
Registers Write data: x
Registers RegWrite: 0

Each 1'

Check if you miss any answer. Each cost 1'.

7. Components need to be added:
   A mux before PC, one input PC value from the mux choose whether to use value from J instruction(only have this get 2'), the other input of readdata 1 from the register (5'), control by some jr signal (3')

8. 220ps (5')
   ALUOp (5') 50ps (5' if the former is correct)
   Assume other control signals don not lengthening the critical path. MemWrite is only used by sw and lw. The critical path between Inst and Data in this question is ***Reg => ALU = 220ps*** *(for the address, it comes from max{****Reg => ALU = 100+120****,* ***SignExt => MUX => ALU = 10+30+100****} =* ***Reg => ALU = 100+120****; for the write data, it comes from* ***Reg = 100****)*
   The question is about "which control signal should be generated most quickly to avoid being on any critical path". We only need to compare ALUOp and ALUSrc. Also, assume the instruction is lw or sw since they do not waste time on passing ***MUX => Reg****. Notice that Read Data 1 takes 100ps to get ready on ALU input, which means ALUOp can only take 50ps (and then pass the 50ps ALU Control) and ALUSrc can only take 70ps (and then pass the 30ps MUX). ALUOp takes the shortest time.*

9. 360ps (4')
   360*5=1800ps (4')
   (2000+4)*360=7214400ps (4')
   1.002 (3')