高雄大學資訊工程系　計算機組織　期末考

學號：　　姓名：

1. (12%) Please explain the three types of hazards in pipelining:
   (a) Structural hazards
   (b) Data hazards
   (c) Control hazards

- Structural hazards: Hardware cannot support this combination of instructions - two instructions need the same resource.
- Data hazards: Instruction depends on result of prior instruction still in the pipeline
- Control hazards: Pipelining of branches & other instructions that change the PC

2. (12%) Please write the conditions for detecting data hazards and the control signals in order to forward data for (1) EX hazard and (2) MEM hazard.

   – At EX stage:
   if (EX/MEM.RegWrite
   and (EX/MEM.RegRd≠0)
   and (EX/MEM.RegRd=ID/EX.RegRs)) ForwardA=10

   if (EX/MEM.RegWrite
   and (EX/MEM.RegRd≠0)
   and (EX/MEM.RegRd=ID/EX.RegRt)) ForwardB=10

   – At MEM stage:
   if (MEM/WB.RegWrite
   and (MEM/WB.RegRd≠0)
   and (MEM/WB.RegRd=ID/EX.RegRs)) ForwardA=01

   if (MEM/WB.RegWrite
   and (MEM/WB.RegRd≠0)
   and (MEM/WB.RegRd=ID/EX.RegRt)) ForwardB=01

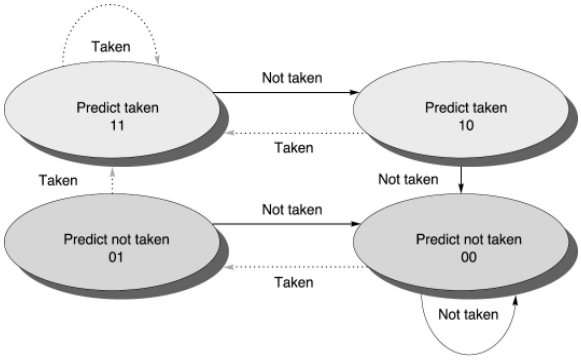
3. (6%) In pipelining design, a hazard detection unit operates during the ID stage so that it can stall between the load and its use. Please write the condition for load instruction.

   if (ID/EX.MemRead and
   ((ID/EX.RegisterRt = IF/ID.RegisterRs) or
   (ID/EX.RegisterRt = IF/ID.registerRt))
   stall the pipeline for one cycle

4. (8%) In order to reduce delay of taken branch, we must move branch execution earlier in the pipeline. Please describe how to let branch execution earlier.

   – Move up branch address calculation to ID
   – Check branch equality at ID (using XOR) by comparing the two registers read during ID
   – Branch decision made at ID => one instruction to flush
   – Add a control signal, IF.Flush, to zero instruction field of IF/ID => making the instruction an NOP

5. (12%)
   A. Which stage is a branch prediction buffer implemented as a small cache accessed during?
   B. Please explain how to determine the branch is taken or not when the branch prediction buffer where each entry is 2 bits.
   C. Please draw how to use the branch prediction buffer to predict.



A. IF stage
B. If (counter >= 2), predict taken, else predict not taken

6. (6%) A direct-mapped cache can contain 16KB data. Each block is 2-word. Assume that the address is 32 bits. How many total bits are required for this cache?

(2^14)/(2*4)= 2^11
(2^11)*(64+(32-11-1-2)+1) = (2^11)*83 = 169984 bits = 21248 bytes = 166 Kbits

7. (12%) Assume
   - 1 memory bus clock cycle to send the address
   - 54 memory bus clock cycles for each DRAM access initiated
   - 1 memory bus clock clock to send a word of data

   If each cache block is four words, please compute the miss penalty for
   1. one-word-wide memory organization
   2. two-word-wide memory organization
   3. interleaved memory organization

1. 1+4*54+4*1 = 221
2. 1+2*54+2*1 = 111
3. 1+1*54+4 = 59

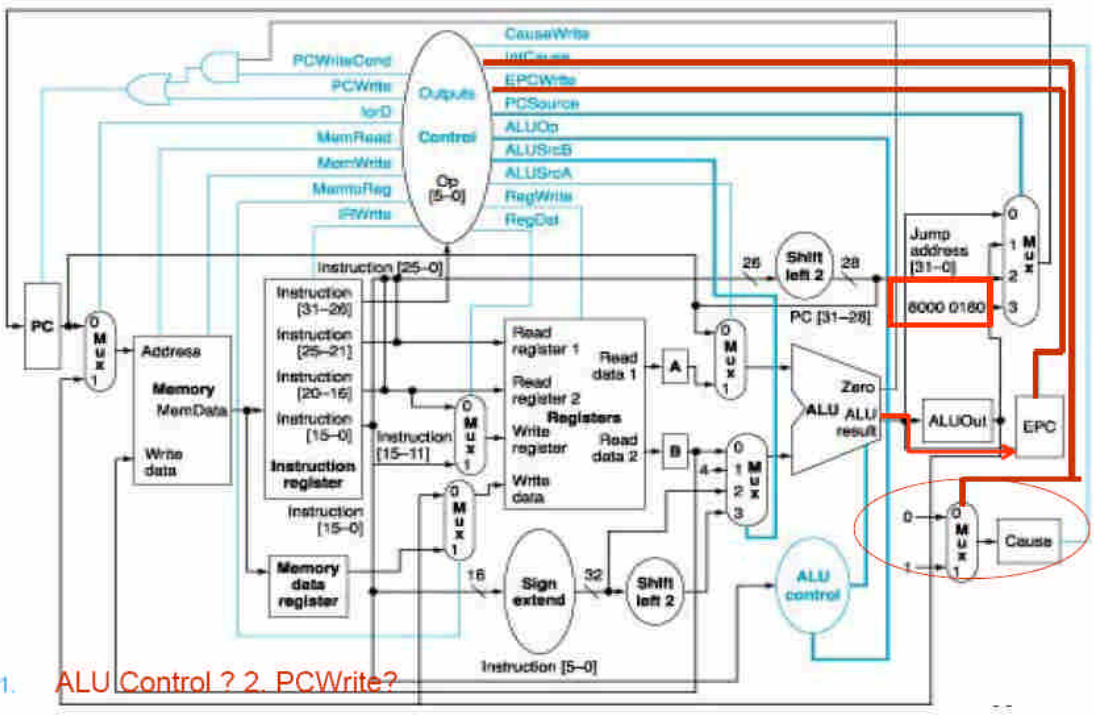
8. (8%) The access time of different components are shown as follows:
Memory units: 200 ps
ALU and adders: 100 ps
Register file (read or write): 50 ps
In a program, there are load, store, R-type, branch, and jump instructions. Their ratios are: 25% loads, 10% stores, 45% R-type instructions, 15% branches, and 5% jumps. Please compare the average time of per instruction under single-cycle implementation and multicycle implementation.

Instruction Class	Functional Units used by the instruction class				
R-Type	Inst Fetch	Register Access	ALU	Register Access	
Load Word	Inst Fetch	Register Access	ALU	Memory Access	Register Access
Store word	Inst Fetch	Register Access	ALU	Memory Access	
Branch	Inst Fetch	Register Access	ALU		
Jump	Inst Fetch				

- For single-cycle implementation
  - The clock cycle for each instruction is determined by the longest instruction, load, which is 600 ps(200+50+100+200+50).
- For multiple cycle implementation
  - The average time per instruction with a variable clock is 600 ×25% + 550 ×10% + 400 ×45% + 350 ×15% + 200 ×5% = 447.5 ps

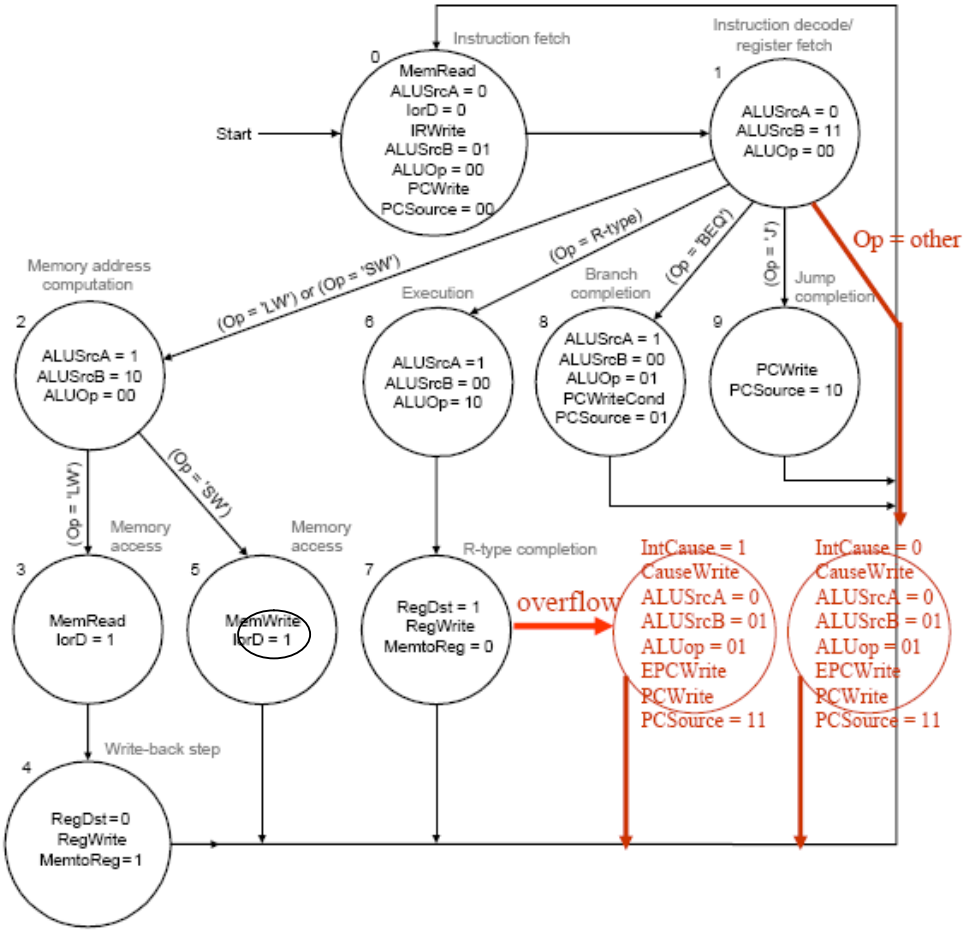
The multiple cycle implementation would be faster by 1.34 (600/447.5).

9. (20%) Please describe the behavior in the different stages for the following instruction types under multiple-cycle implementation. Besides, please finish the control signals of the finite state machine.



1. ALUControl ? 2. PCWrite?

Step name	Action for R-type instructions	Action for memory-reference instructions	Action for branches	Action for jumps
Instruction fetch	IR = Memory[PC] PC = PC + 4			
Instruction decode/register fetch	A = Reg [IR[25-21]] B = Reg [IR[20-16]] ALUOut = PC + (sign-extend (IR[15-0]) << 2)			
Execution, address computation, branch/ jump completion	ALUOut = A op B	ALUOut = A + sign-extend (IR[15-0])	if (A ==B) then PC = ALUOut	PC = PC [31-28] II (IR[25-0]<<2)
Memory access or R-type completion	Reg [IR[15-11]] = ALUOut	Load: MDR = Memory[ALUOut] or Store: Memory [ALUOut] = B		
Memory read completion		Load: Reg[IR[20-16]] = MDR		



10. (6%) Please explain temporal locality and spatial locality.

- Temporal locality: if an item is referenced, it will tend to be referenced again soon
- Spatial locality: if an item is referenced, items whose addresses are close by tend to be referenced soon

11. (4%) Please describe how to update data items in write-through strategy and write-back strategy.

    A. Write-through: write to cache and memory at same time => but memory is very slow!
    B. Write-back: write to cache only (write to memory when that block is being replaced)

12. (8%) Please explain the three types of cache misses, i.e., Compulsory, Conflict, and Capacity. Please propose two solutions to reduce Conflict and Capacity misses.

    A. Compulsory (cold start, process migration):
        (1) First access to a block, not much we can do
        (2) Note: If you are going to run billions of instruction, compulsory misses are insignificant
    B. Conflict (collision):
        (1) >1 memory blocks mapped to same location
        (2) Solution 1: increase cache size
        (3) Solution 2: increase associativity
    C. Capacity:
        (1) Cache cannot contain all blocks by program
        (2) Solution: increase cache size

13. (6%) Please explain how we can improve cache performance. (Hint: Give thee factors of average memory access time.)
    A. Decreasing the miss ratio
    B. Reduce the time to hit in the cache
    C. Decreasing the miss penalty