1. (10%) For single-cycle or multi-cycle implementation of CPU, which one is more efficient? Explain the reason why it is better.
2. (10%) Compute the clock cycle per instruction (CPI) for the following instruction mix. The mix includes 25% loads, 18% stores, 40% R-format operations, 15% branches, and 2% jumps. The number of clock cycles for each instruction class is listed as follows. 5 cycles for loads, 4 cycles for stores, 4 cycles for R-format instructions, 4 cycles for branches, 3 cycles for jumps.
3. (10%) Considering the following code sequence,

```
LOAD    R1, 45(R2)
SUB     R3, R1, R2
LOAD    R2, 50(R1)
ADD     R1, R2, 100
OR      R3, R2, R1
```

(1) What are the logic detecting the data hazard conditions of the load interlocks?

	Opcode field of ID/EX ($ID/EX.IR_{0..5}$)	Opcode field of IF/ID ($IF/ID.IR_{0..5}$)	Matching operand fields
(Rule 1)	Load	Register-register ALU	ID/EX.IR[rt] == IF/ID.IR[rs]
(Rule 2)	Load	Register-register ALU	ID/EX.IR[rt] == IF/ID.IR[rt]
(Rule 3)	Load	Load, store, ALU immediate, or branch	ID/EX.IR[rt] == IF/ID.IR[rs]

(A) Rule 1 and Rule 2
(B) Rule 2 and Rule 3
(C) Rule 1 and Rule 3
(D) Rule 3
(E) Rule 1

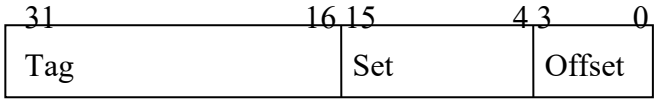
(2) Is forwarding necessary between the first LOAD and the second LOAD instruction? If yes, what is the logic detecting the forwarding condition between the first LOAD and t the second LOAD?
Detection Logic Options for (2):

	Pipeline register containing source instruction	Opcode of source instruction	Pipeline register containing destination instruction	Opcode of destination instruction	Destination of the forwarded result	Comparison (if equal then forward)
(A)	EX/MEM	Register-register ALU	ID/EX	Register-register ALU, ALU immediate, load, store, branch	Top ALU input	EX/MEM.IR[rd] == ID/EX.IR[rs]
(B)	EX/MEM	Register-register ALU	ID/EX	Register-register ALU	Bottom ALU input	EX/MEM.IR[rd] == ID/EX.IR[rt]
(C)	MEM/WB	Register-register ALU	ID/EX	Register-register ALU, ALU immediate, load, store, branch	Top ALU input	MEM/WB.IR[rd] == ID/EX.IR[rs]
(D)	MEM/WB	Register-register ALU	ID/EX	Register-register ALU	Bottom ALU input	MEM/WB.IR[rd] == ID/EX.IR[rt]
(E)	EX/MEM	ALU immediate	ID/EX	Register-register ALU, ALU immediate, load, store, branch	Top ALU input	EX/MEM.IR[rt] == ID/EX.IR[rs]
(F)	EX/MEM	ALU immediate	ID/EX	Register-register ALU	Bottom ALU input	EX/MEM.IR[rt] == ID/EX.IR[rt]
(G)	MEM/WB	ALU immediate	ID/EX	Register-register ALU, ALU immediate, load, store, branch	Top ALU input	MEM/WB.IR[rt] == ID/EX.IR[rs]
(H)	MEM/WB	ALU immediate	ID/EX	Register-register ALU	Bottom ALU input	MEM/WB.IR[rt] == ID/EX.IR[rt]
(I)	MEM/WB	Load	ID/EX	Register-register ALU, ALU immediate, load, store, branch	Top ALU input	MEM/WB.IR[rt] == ID/EX.IR[rs]
(J)	MEM/WB	Load	ID/EX	Register-register ALU	Bottom ALU input	MEM/WB.IR[rt] == ID/EX.IR[rt]

4. (10%) Computer A can execute more instructions within a second compared to computer B. Is it proper to conclude that the performance of computer A is better than computer B? Why or why not?

5. (10%) Suppose we would like to achieve a speedup of 10 with 40 processors. What fraction of the original computation can be sequential?

6. (38%) About memory hierarchy
   (1) Describe the reason why we need a memory hierarchy and how it can work. (6 pts.)
   (2) Given cache, TLB, page table, main memory, and the secondary storage (disk), describe the process of a memory access in a systematic way. (10 pts.)
   (3) Describe the possible positive and negative effects of increasing (a) cache size, (b) associativity, and (c) block size on the overall performance of memory access. (10 pts.)
   (4) Find the average memory access time with a 1 ns clock, a miss penalty of 25 clock cycles, a miss rate of 0.08 misses per instruction, and a cache access time (including hit detection) of 1 clock cycle. Assume that the read and write miss penalties are the same and ignore other write stalls. (6 pts.)
   (5) As in (4), suppose we can improve the miss rate to 0.04 misses per reference by doubling the cache size. This causes access time to increase to 1.8 cycles. Using the average memory access time as a metric, determine if this is a good trade-off. (6 pts.)
7. (12%) A 32-bit cache memory address is decomposed into 3 fields as shown below:



   What is the total size (including tag, data, and valid bits) of the cache memory in ***bits*** if the cache is
   (1) direct-mapped (6 pts.)
   (2) 2-way set associative? (6 pts.)