

Question 1. (4 points)

What are the main principles behind a direct mapped cache and a set associative cache? Illustrate with figures.

Question 2. (2 points)

List two design principles behind the MIPS instruction set design. Explain each with example.

Question 3. (4 points)

Reading from cache does not require a special strategy, however, writing to cache needs special strategies. Why?

Write through, write back and *write back* are three write strategies for cache. Explain each strategy and their relative advantages.

Question 4. (3 points)

Define the three types of pipeline hazards with examples.

Question 5. (2 points)

Reduction is a programming technique used in parallel programming. Explain the technique briefly. You may use summation of numbers in an array as an example. It is not required to write code.

Question 6. (2 points)

What is the role of a page table in virtual memory? What does it store?

Question 7. (2 points)

How does a branch history table work?

For the following questions, Q8 to Q13, it is mandatory to write couple of lines of explanation to discuss your approach to solve the problem.

Question 8. (4 points)

Consider two different implementations, M1 and M2, of the same instruction set. There are three classes of instructions (A, B, and C) in the instruction set. For a given program, the average number of cycles for each instruction class is shown below. The table also shows how many instructions of a given class are in the program, as a percentage. E.g., if there are 100 instructions in total, there are 60 Class A instructions.

Instruction Class	Computer M1 (Cycles Per Instruction Class)	Computer M2 (Cycles Per Instruction Class)	Percentage of total instruction
A	1	2	60%
B	2	3	30%
C	4	4	10%

Calculate the average **CPI** (Clocks Cycles per Instruction) for computer M1 and M2.

Question 9. (4 points)

The following problem deals with translating from MIPS to C/Java. Assume that the variables f, g, h, i, and j are assigned to registers \$s0, \$s1, \$s2, \$s3, and \$s4, respectively. Assume that the base address of the arrays A and B are in registers \$s6 and \$s7, respectively.

```
sub    $t0, $s3, $s4
slli   $t1, $t0, 2
add    $t2, $t1, $s6
lw     $t0, 0($t2)
sw     $t0, 32($s7)
```

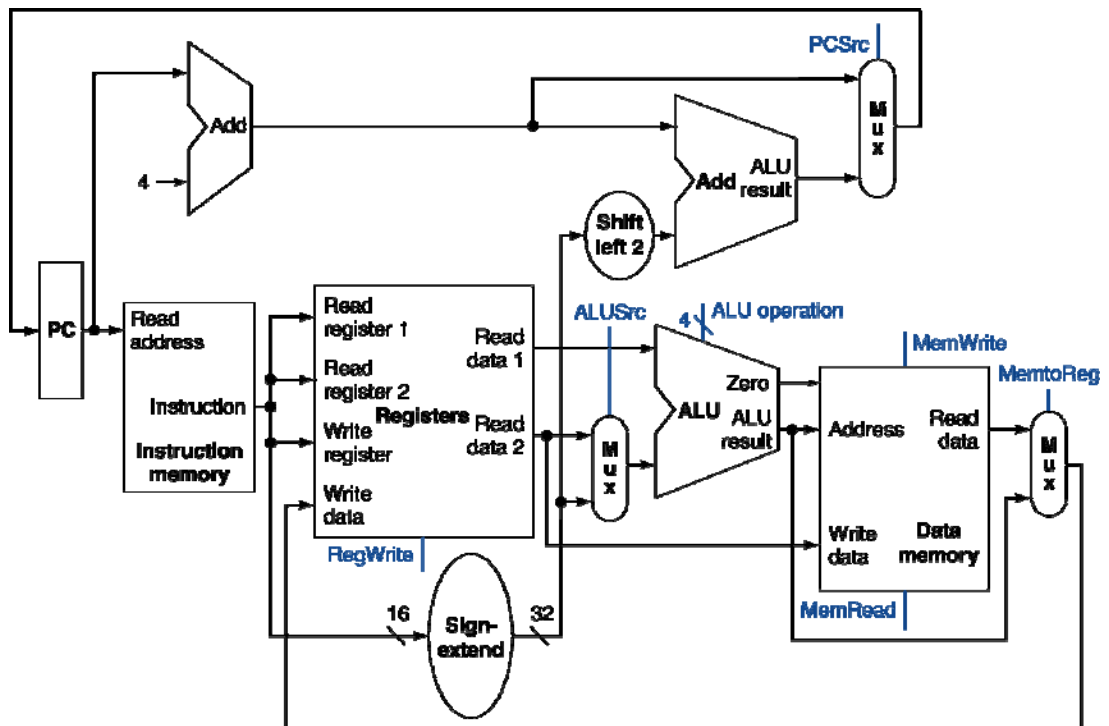
Write the final C/Java code. Show the intermediate steps you work out for each assembly statement. Hint: slli is the same as sll except that the second argument is a number.

Question 10. (4 points)

Consider the datapath shown in the figure. If we only have to support the load instruction (`lw $t0, offset($s0)`)

- Which components are utilized by the load instruction?
- What are the values of control signals (those signals that are shown in the figure) generated for this instruction?
- What is the clock cycle time? Assume the following latencies for each block in the datapath.

I-Mem	ADD	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-left-2
200 ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps



Question 11. (3 points)

In this question, assume that all branches are perfectly predicted (this eliminates all control hazards). Assume, we have only one memory (for both instructions and data), and there might be a structural hazard. To resolve this, the pipeline must be stalled in some cycles.

- (i) Show the pipelined execution for each instruction.
- (ii) Identify and explain where the pipeline stalls.

```
                SW  R2, 0(R3)
                OR  R1, R2, R3
                BEQ R2, R0, Label (Assume R2== R0)
                OR  R2, R2, R0
Label :         ADD R1, R4, R3
```

Question 12. (3 points)

Design a 8-way set associative cache that has 16 blocks and 32 bytes per block. Assume a 32 bit address. Calculate the following:

- (a) How many bits are used for the byte offset?
 - (b) How many bits are used for the set (index) field?
 - (c) How many bits are used for the tag?
-

Question 13. (3 points)

For the contents of registers \$s1 = 0000 ... 0001_b and \$s2 = 1111 ...1111_b, what is the value of \$t0 for the following assembly code? All registers are 32 bit long.

```
add  $t0, $s1, $s2
```

Is the result in \$t0 the desired result, or has there been overflow?
