If some questions are unclear or not well defined to you, you can make your own assumptions and state them clearly in the answer sheet.

1. Short Questions (35%)

Answer and explain the following questions. Credit will be given only if explanation is provided.

1.1 What is the "addressing mode"? Enumerate and explain two addressing modes that are the frequently used in reduced instruction set computers (RISCs).

1.2 RISCs generally have poor code density (larger code size) compared with CISCs (complex instruction set computers). Please explain that RISCs how to reduce code size?

1.3 Pentium 4 has a much deeper pipeline than Pentium III. Please explain the advantage and disadvantage of deeper pipeline.

1.4 Is the floating-point addition performed in a computer associative? Why?

1.5 Derive the IEEE 754 binary representation for the floating-point number -10.75_{ten} in single precision.

1.6 Assume that multiply instructions take 10 cycles and account for 20% of the instructions in a typical program and that the other 80% of the instructions require an average of 5 cycles for each instruction. What percentage of time does the CPU spend doing multiplication?

1.7 Assume that in 1000 memory references there are 30 misses in the first-level (L1) cache and 6 misses in the second-level (L2) cache. The miss penalty from the L2 cache to memory is 100 clock cycles, the hit time of the L2 cache is 10 clock cycles, the hit time of L1 is 1 clock cycle, and there are 1.5 memory references per instruction. What is the average memory access time? Ignore the impact of writes.

2. Computer Datapath (25%)

2.1 (15%) There are three methods to implement the datapath: single-cycle (M1), multicycle (M2), and pipeline (M3). The operation times for the major functional units in these implementations are 2 ns for memory access, 2 ns for ALU operation, and 1 ns for register file read or write. Assuming that the multiplexors, control unit, PC accesses, sign extension unit, and wires have no delay. The other details of these three implementations are listed as follows:

M1: The critical path of single-cycle implementation for the different instruction classes is:

Instruction class	Functional units used by the instruction class				
ALU type	Instruction fetch	Register access	ALU	Register access	
Load word	Instruction fetch	Register access	ALU	Memory access	Register access
Store word	Instruction fetch	Register access	ALU	Memory access	
Branch	Instruction fetch	Register access	ALU		
Jump	Instruction fetch				

M2: Multicycle implementation uses the control shown in Fig. 1 (on page 4).

M3: For the pipelined implementation, assume that half of the load instructions are immediately followed by an instruction that uses the result, that the branch delay on misprediction is 1 clock cycle, and that one-quarter of the branches are mispredicted. Assume that jumps always pay 1 full clock cycle of delay, so their average time is 2 clock cycles.

If the instruction mix is 23% loads, 13% stores, 43% ALU instructions, 19% branches, and 2% jumps. Please calculate the average CPI (clock cycles per instruction) and the average instruction time for the three implementations.

2.2 (10%) Consider the five instructions in the following program. These instructions execute on the five-stage pipelined datapath of Fig. 2 (on page 4). The five stages are: instruction fetch (IF), instruction decode and fetch operand (ID), instruction execution (EX), memory access (MEM), and register write back (WB). Assume that each stage takes one cycle to complete its execution and the first instruction starts from clock cycle 1.

```
add      $1, $2, $3
add      $4, $5, $1
lw       $6, 50($7)
sub      $8, $6, $9
add      $10, $11, $8
```

(a) At the end of the fifth cycle of execution, which registers are being read and which register will be written? How many cycles will it take to execute this program?

(b) Explain what the forwarding unit and the hazard detection unit are doing during the fifth cycle of execution. If any comparisons are being made, mention them.

3. Performance Analysis (20%)

3.1 (6%) Suppose we have made the following measurements:

Frequency of floating-point (FP) instructions = 5%
Frequency of ALU instructions = 25%
Average CPI of FP instructions = 20.0
Average CPI of ALU instructions = 4.0
Average CPI of other instructions = 2.0

Assume that the two design alternatives are to decrease the average CPI of FP instructions to 8.0 or to decrease the average CPI of ALU instructions to 2.0. Compare the performance of these two design alternatives.

3.2 (14%) Consider the following three processors with the same instruction architecture:

1. A simple processor running at a clock rate of 1.2 GHz and achieving a pipeline CPI of 1.0. This processor has a cache system that yields 0.01 misses per instruction.

2. A deeply pipelined processor with slightly smaller caches and a 1.5 GHz clock rate. The pipeline CPI of the processor is 1.2, and the smaller caches yield 0.014 misses per instruction on average.

3. A speculative superscalar processor. This processor has the smallest caches and a 1 GHz clock rate. The pipeline CPI of the processor is 0.4, and the smallest caches lead to 0.02 misses per instruction, but it hides 20% of the miss penalty on every miss by dynamic scheduling.

Assume that the main memory time (which sets the miss penalty) is 100 ns. Determine the relative performance of these three processors.

4. Others (20%)

4.1 (8%) Suppose the branch frequencies (as percentages of all instructions) as follows:

Conditional branches	20%
Unconditional branches	1%
Conditional branches	60% are taken

Consider a five-stage pipelined machine where the branch is resolved at the end of the third cycle for conditional branches and at the end of the second cycle for unconditional branches. Assuming that only the first pipe stage can always be done independent of whether the branch goes and ignoring other pipeline stalls, how much faster would the machine be without any branch hazards?

4.2 (12%) A superscalar MIPS processor can issue two instructions per clock cycle. One of the instructions could be an integer ALU operation or branch, and the other could be a load or store.

(a) Enumerate the possible extra resources required for extending a simple MIPS pipeline into the superscalar pipeline so that it wouldn't be hindered by structural hazards.

(b) Unroll the following loop once under the assumption that the loop index is a multiple of two. How would this unrolled loop be scheduled on the superscalar pipeline for MIPS? Reorder the instructions to avoid as many pipeline stalls as possible and show your unrolled and scheduled code as the following table.

```
Loop:  lw    $t0, 0($s1)
       addu  $t0, $t0, $s2
       sw    $t0, 0($s1)
       addi  $s1, $s1, -4
       bne   $s1, $zero, Loop
```

	ALU or branch instruction	Data transfer instruction	Clock cycle
Loop:			1
			2
			3
			•••

Fig. 1

States described in the diagram:

0 — Instruction fetch
MemRead
ALUSrcA = 0
IorD = 0
IRWrite
ALUSrcB = 01
ALUOp = 00
PCWrite
PCSource = 00

1 — Instruction decode/register fetch
ALUSrcA = 0
ALUSrcB = 11
ALUOp = 00

2 — Memory address computation
(Op = 'LW') or (Op = 'SW')
ALUSrcA = 1
ALUSrcB = 10
ALUOp = 00

6 — Execution
(Op = R-type)
ALUSrcA = 1
ALUSrcB = 00
ALUOp = 10

8 — Branch completion
(Op = 'BEQ')
ALUSrcA = 1
ALUSrcB = 00
ALUOp = 01
PCWriteCond
PCSource = 01

9 — Jump completion
(Op = 'J')
PCWrite
PCSource = 10

3 — Memory access
(Op = 'LW')
MemRead
IorD = 1

5 — Memory access
(Op = 'SW')
MemWrite
IorD = 1

7 — R-type completion
RegDst = 1
RegWrite
MemtoReg = 0

4 — Write-back step
RegDst = 0
RegWrite
MemtoReg = 1

Start

Fig. 1

Fig. 2

Labels in the datapath diagram:

Hazard detection unit
ID/EX.MemRead
ID/EX
WB
M
EX
EX/MEM
WB
M
MEM/WB
WB
IF/IDWrite
Control
Mux
0
PCWrite
PC
Instruction memory
IF/ID
Instruction
Registers
Mux
ALU
Data memory
Mux
IF/ID.RegisterRs
IF/ID.RegisterRt
IF/ID.RegisterRt
IF/ID.RegisterRd
ID/EX.RegisterRt
Rt
Rd
Mux
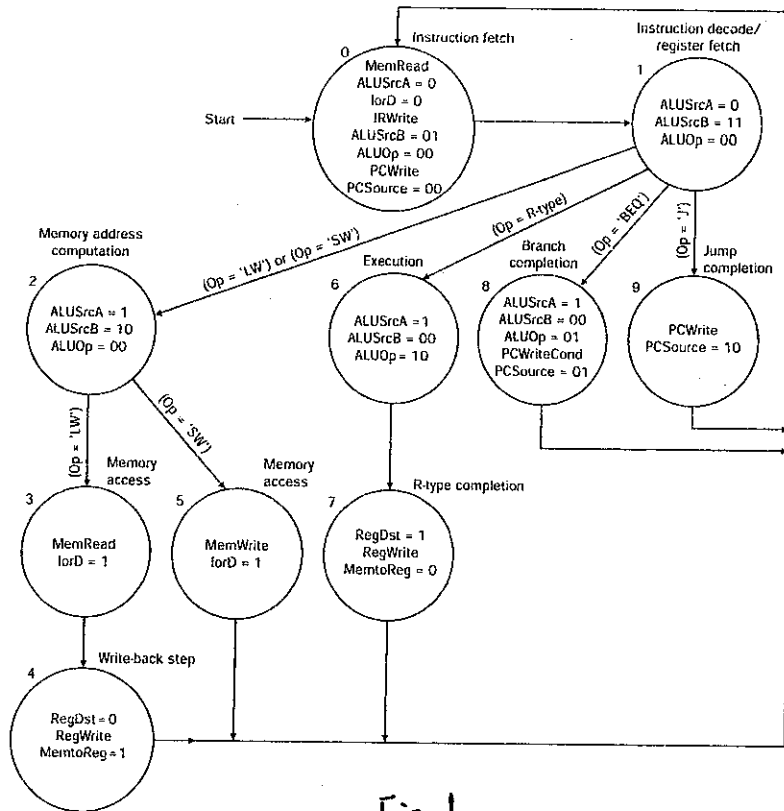Rs
Rt
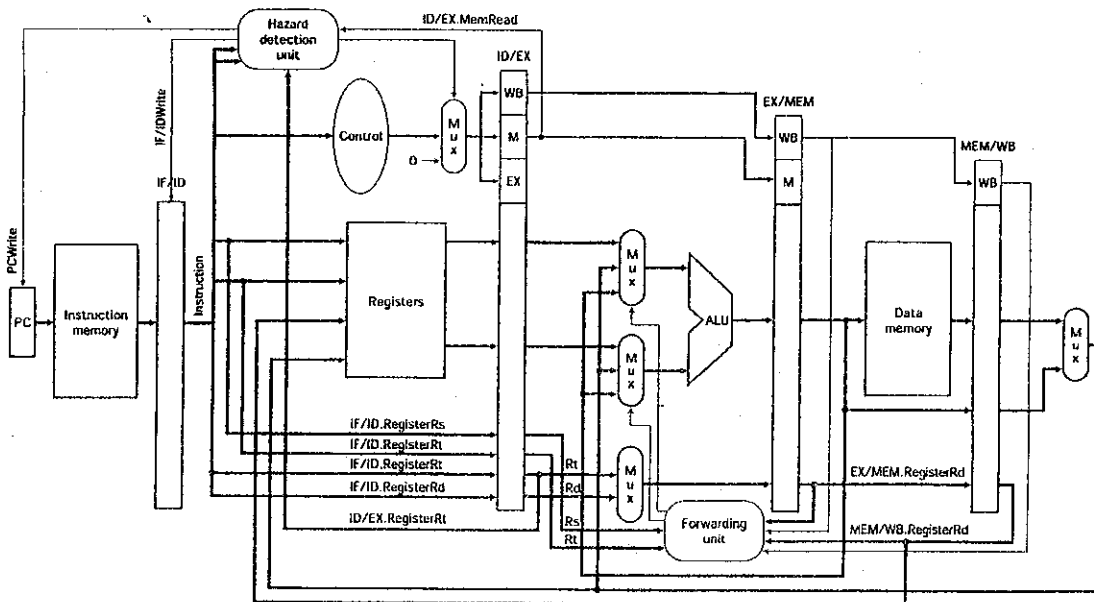Forwarding unit
EX/MEM.RegisterRd
MEM/WB.RegisterRd

Fig. 2

1. (15%) System D uses 512 byte pages and frames for its virtual memory scheme.

   (a). Program Q's logical address space is 20 pages long. Convert each one-dimensional logical address in the following stream to (p,d) format.
   1252, 2322, 528, 5280, 1550

   (b). Assume that the program from above is allocated five frames during execution and that demand paging with local replacement is used, decide which page reference from the stream cause page faults under each replacement strategy bellow

   (i)   FIFO

   (ii)   LRU

   (iii) LFU

   (c). Redo (b) where only 3 frames are allocated to program Q

2. (15%) Consider a computer system that runs 5,000 jobs per month with no deadlock-prevention or deadlock-avoidance scheme. Deadlocks occur about twice per month, and the operator must terminate and rerun about 10 jobs per deadlock. Each job is worth about \$2 (in CPU time), and the jobs terminated tend to be about half-done when they are aborted. A system programmer has estimated that a deadlock-avoidance algorithm (like the banker's algorithm) could be installed in the system with an increase in the average execution time per job of about 10 percent. Since the machine currently has 30-percent idle time, all 5,000 jobs per month could still be run, although turnaround time would increase by about 20 percent on average.

   (a). What are the arguments for installing the deadlock-avoidance algorithm?

   (b). What are the arguments against installing the deadlock-avoidance algorithm?

3. (10%) Consider a system where free space is kept in a free-space list.

   (a). Suppose that the pointer to the free-space list is lost. Can the system reconstruct the free-space list? Explain your answer.

   (b). Suggest a scheme to ensure that the pointer is never lost as a result of memory failure.

4. (10%) Demonstrate that monitors, conditional critical regions, and semaphores are all equivalent, insofar as the same types of synchronization problems can be implemented with them.

5. (5%) Transform the following predicate calculus into the prefix form

$$(\neg x \wedge y) \vee (\neg y \wedge z) \wedge (x \vee \neg z)$$

6. (5%) Is the *postfix* expression AB + C * DE - +FG * + - HI * - K L + M * a valid statement?

7. (10%) The depth-first spanning tree means that the spanning tree is derived from a depth-first search. For the following adjacent matrix, whose node sequence is given too, please derive the depth-first spanning tree and write your result in terms of graph and node sequence number as shown in the adjacent matrix. (Use node 1 as the root)

$$
\begin{array}{c c}
& \begin{array}{c c c c c c c c} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} &
\left[\begin{array}{c c c c c c c c}
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0
\end{array} \right]
\end{array}
$$

8. (20%) The following figure represents a two dimensional scene. Each labeled pixel represents a two dimensional object. They are labeled by alphabetical letters.

(a) Please draw a search tree with minimum level (or depth) to represent these objects. The degree of the search tree you can use is 2.

(b) Based on your search tree to write a procedure to retrieve the objects.
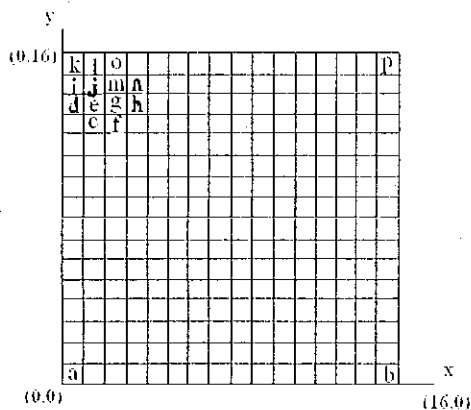
9. (10%) The odd-even transposition sort proceeds as follows. Pass through the file several times. On the first pass, compare x[i] with x[i+1] for all odd i. On the second pass, compare x[i] with x[i+1] for all even i. Each time that x[i] > x[i+1], interchange the two. Continue alternating in this fashion until the file is sorted.

(a) What is the condition for the termination of the sort?

(b) On the average what is the efficiency of the sort?

1. Please explain the following terms:
    (a) The Pigeonhole Principle　　　　　　　　(5 %)
    (b) Equivalence relation　　　　　　　　　　(5 %)
    (c) Partially ordered set　　　　　　　　　　(5 %)
    (d) Galois Field　　　　　　　　　　　　　　(5 %)
    (e) Planar graph　　　　　　　　　　　　　　(5 %)
    (f) Finite state machine　　　　　　　　　　(5 %)

2. Prove that: if G is a group of order n with a subgroup H of order m, then m divides n.　　　　　　　　　　(15 %)

3. Show that:
    (a) $\sum_{i=1}^{n} \dfrac{n}{i} = O(n \log n)$　　　　　　　　(13 %)
    (b) $\log(n!) = \Omega(n \log n)$　　　　　　　　(7 %)

4. Considering the associative law in matrix multiplication, we have 2 different ways to compute $A_3 = M_1 * M_2 * M_3$, i.e., $((M_1 * M_2) * M_3)$ and $(M_1 * (M_2 * M_3))$, and there are 5 different ways to compute $A_4 = M_1 * M_2 * M_3 * M_4$, i.e., $(((M_1 * M_2) * M_3) * M_4)$, $((M_1 * M_2) * (M_3 * M_4))$, $((M_1 * (M_2 * M_3)) * M_4)$, $(M_1 * ((M_2 * M_3) * M_4))$, and $(M_1 * (M_2 * (M_3 * M_4)))$.　　Let $A_n = M_1 * M_2 * M_3 \ldots \ldots * M_n$ where $M_1, M_2, M_3, \ldots, M_n$ are n given matrixes.
    (a) How many different ways are there to compute A_n ?　　　　　　　　(5 %)
    (b) Prove your answer in (a).　　　　　　　　(15 %)

5. Let p and q be two distinct primes and $n = pq$.
    (a) Please find a positive integer k such that $(m^k \bmod n) = 1$ for each positive integer m less than and relatively prime to n where k is a polynomial of p and q.　　　　　　　　(10 %)
    (b) By (a), please find $[m]^{-1}$ in Z_n.　　　　　　　　(5 %)