Specialized Subjects

15:00-17:30, Monday, August 18, 2014

Instructions

1. Do not open this booklet before the examination begins.
2. This booklet contains five problems. The number of pages is nine excluding this cover sheet and blank pages. If you find missing or badly printed pages, ask the proctor for exchange.
3. Answer three problems. You can select any three out of the five. Your answer to each problem should be written on a separate sheet. You may use the reverse side of the sheet if necessary.
4. Fill the top parts of your three answer sheets as instructed below. Before submitting your answer sheets, make sure that the top parts are correctly filled.

専　門　科　目

第　　問

↑ Write the problem No.

受験番号				

↑ Write the examinee No.

5. Submit all the three answer sheets with the examinee number and the problem number, even if your answer is blank.
6. Answer either in Japanese or English.
7. This booklet and the scratch paper must be returned at the end of the examination.
8. This English translation is supplemental and provided for convenience of applicants. The Japanese version is the formal one.

Problem 1

Figure 1 shows an equivalent circuit for an operational amplifier. An operational amplifier is ideal when its impedance Z_i between + and − input terminals, impedance Z_o of output terminal, and open loop gain A can be assumed as ∞, 0, and ∞, respectively. Answer the following questions.

(1) Find the gain $\frac{v_o}{v_r}$ of the inverting amplifier shown in Fig. 2. Assume that the operational amplifier is ideal.

(2) Find the gain $\frac{v_o}{v_r}$ for the above question (1), when A is finite.

(3) When resistance R_f is replaced by capacitance C_f in the circuit shown in Fig. 2, the circuit is called an integrator. Show the frequency characteristic of $\frac{v_o}{v_r}$ as a function of complex frequency s, when the open loop gain A of the operational amplifier has the frequency characteristic $A(s) = \frac{A_0\omega_0}{s+\omega_0}$.

(4) Assume that A_0 is sufficiently large in the above question (3). Show that the range of angular frequency ω, where the circuit acts as an integrator, is approximately restricted to $\frac{1}{A_0 C_f R_r} < \omega < A_0\omega_0$, by approximating frequency characteristic $\frac{v_o}{v_r}$, and considering two extremes $\omega \to 0$ and $\omega \to \infty$.

(5) Find the gain $\frac{v_o}{v_n}$ of the non-inverting amplifier shown in Fig. 3. Assume that the operational amplifier is ideal.

(6) Related to the question (5), find the input impedance Z_{in} of the non-inverting amplifier, when the operational amplifier is not ideal; Z_i and A in Fig. 1 take finite values. Assume that Z_o is approximated as 0.

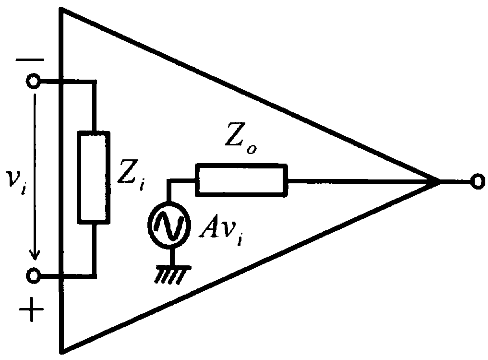


Fig. 1

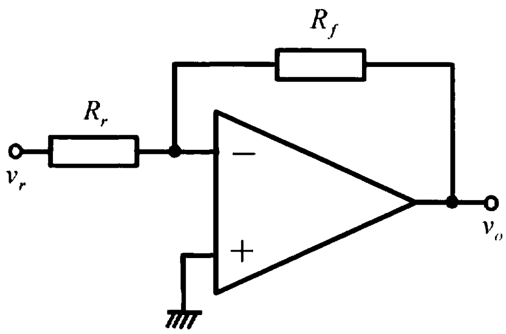


Fig. 2

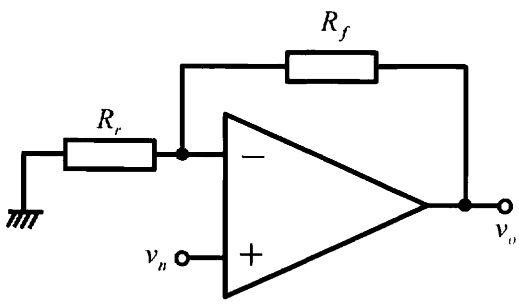


Fig. 3

Problem 2

Answer the following questions.

(1) Modern superscalar processors are equipped with several mechanisms to enhance memory access performance of programs, including:

(a) caches,

(b) hardware prefetchers, and

(c) non-blocking caches.

Briefly explain what each mechanism does and how it contributes to enhancing memory access performance of programs.

(2) Suppose we are given an array of structures and a program that repeatedly accesses all its elements. We examine how its performance is affected by the method with which to access elements and the array size. Specifically, consider the following methods 1 through 4.

Method 1: Sequentially access all elements from the beginning to the end.

Method 2: Generate array indices randomly and access elements at those indices in the order they are generated. Assume each valid index of the array is generated once and only once. Generating a random number approximately takes 7-8 processor cycles and needs no memory access instructions.

Method 3: Let each element have a pointer to its successor element, thereby connecting all the elements in a linear list, and access them by traversing these pointers. Here, an element points to the element immediately following it in the array. Elements are thus accessed in the same order as in Method 1.

Method 4: The same as Method 3, except that pointers are generated randomly in the manner of Method 2. That is, the pointer of each element points to the element that would be accessed next to it in Method 2. Elements are thus accessed in the same order as in Method 2.

Assume a single element is 16 bytes large and the processor has caches of level 1 through 3, whose sizes are 32KB, 256KB, and 4MB, respectively (1KB = 2^{10} bytes, 1MB= 2^{20} bytes).

With each method, we repeatedly and consecutively accessed the same array many times and measured the average time per access. We observed the time varies depending on the method used and the number of elements of the array (N), as shown in the table below. Each number represents an average time (processor cycles) per access.

	$N \approx 2^{10}$	$N \approx 2^{22}$
Method 1	1.1	8.0
Method 2	8.4	47.5
Method 3	7.0	(x)
Method 4	7.0	199.2

What is the mechanism most relevant to the performance difference between the two cases in each of the following? Choose from (a) through (c) above, or say none if none of them are relevant, and explain your reasoning.

(2-1) (Method 1, $N \approx 2^{10}$) and (Method 2, $N \approx 2^{10}$).

(2-2) (Method 4, $N \approx 2^{10}$) and (Method 4, $N \approx 2^{22}$).

(2-3) (Method 1, $N \approx 2^{22}$) and (Method 2, $N \approx 2^{22}$).

(2-4) (Method 2, $N \approx 2^{22}$) and (Method 4, $N \approx 2^{22}$).

(3) How large is (x) in the table? Choose the most conceivable one from the following and explain.

(a) In the vicinity of 8.0; that is, it's similar to Method 1.

(b) In the vicinity of 47.5; that is, it's similar to Method 2.

(c) In the vicinity of 199.2; that is, it's similar to Method 4.

Problem 3

Let $f = a_0 + a_1 x + \cdots + a_m x^m$ and $g = b_0 + b_1 x + \cdots + b_n x^n$ be polynomials of x (a_i and b_i are real. $a_m \neq 0$ and $b_n \neq 0$). We represent the leading terms of the polynomials f and g by $\mathrm{LT}(f) = a_m x^m$ and $\mathrm{LT}(g) = b_n x^n$, and their degrees by $\deg(f) = m$ and $\deg(g) = n$. The polynomial division, where f is divided by a non-zero polynomial g, is given by

$$f = qg + r.$$

Here quotient q and remainder r are polynomials of x satisfying $r = 0$ or $\deg(r) < \deg(g)$. In this case, we represent $r = \mathrm{remainder}(f, g)$ and $q = \mathrm{quotient}(f, g)$.

(1) Calculate $\mathrm{quotient}(f, g)$ and $\mathrm{remainder}(f, g)$ for $f = x^2 + 7x + 3$ and $g = x + 1$.

(2) Complete a pseudocode of the polynomial division algorithm by filling (a) with appropriate expressions. Note that the four arithmetic operations for monomial terms (expressions that contain only one term, e.g. $7x^3$ or $-5x^{10}$) and addition/subtraction operations for polynomials can be used as they are.

Input : f, g
Output : q, r
$q = 0,\ r = f$
while ($r \neq 0$ and $\deg(g) \leq \deg(r)$) {
  $q = q + \mathrm{LT}(r)/\mathrm{LT}(g)$
  $r =$ ____(a)____
}

(3) Prove that the algorithm introduced in (2) always terminates.

(4) The greatest common divisor (GCD) for polynomials f and g is a polynomial h which satisfies the following conditions.

- h divides f and g
- if a polynomial p divides f and g, then p also divides h

h satisfying these conditions is represented by $h = \mathrm{GCD}(f, g)$. $\mathrm{GCD}(f, g)$ is unique up to multiplication by nonzero numbers. Given $f = qg + r$, by using the following relations $\mathrm{GCD}(f, g) = \mathrm{GCD}(f - qg, g)$ and $\mathrm{GCD}(f, 0) = f$, $\mathrm{GCD}(f, g)$ can be calculated by the following procedure (without loss of generality, we assume $\deg(f) \geq \deg(g)$). Fill (b) and (c) with appropriate expressions.

Input f, g
Output: h
$h = f$
$s = g$
while ($s \neq 0$) {
    rem = remainder(h, s)
    h = ______(b)______
    s = ______(c)______
}

(5) Given arbitrary polynomials f and g ($\deg(f) \geq \deg(g)$), calculate an upper bound of the number of times that a function remainder is called inside the while-loop during the calculation of GCD(f, g). Also provide a reason for the obtained result.

Problem 4

Consider the system that uses the following algorithm to calculate transmission routes of IP packets over the network composed of nodes and links.

The transmission route calculation algorithm:

Every node sends a routing table that consists of vectors {Destination node, Next hop node, Number of hops} to all the neighbor node(s) connected by link(s) every 30 [sec]. Fig. 1 shows an example of a routing table of Node A. In the figure, the number of hops, $d(i,j)$, represents the minimum number of hops so as to reach Node j from Node i, and is calculated by the following expression.

$$d(i,j) = \min\{d(i,k) + d(k,j)\}, \qquad \text{for all neighbor node } k \text{ of node } i$$

Here, when there are multiple routes having the same number of hops, the route that has smaller node number of neighbor node is chosen as an appropriate route.

Answer the following questions.

(1) Regarding the network shown in Fig. 2, show the routing table at Node 1, when the routing tables are converged after enough routing table exchanges among nodes. Here, a number shown in a circle represents its node number and a link connecting nodes is shown by a line with Li (i is an integer).

(2) Using the routing table information, we can generate the spanning tree that represents the transmission route of IP packets from Node 1 (as a root node) to all the other nodes (Nodes 2, 3, 4 and 5). Show this spanning tree.

(3) Links L1 and L8 are disconnected simultaneously. Show the routing table at Node 1, and show the spanning tree representing the IP packet transmission route from Node 1 to all the other nodes (Nodes 2, 3, 4 and 5), when the routing tables are converged after enough routing table exchanges among nodes.

(4) As shown in Fig. 3, a node that has the same node identification number with Node 3 is connected to the network with links L10 (connecting to Node 1) and L9 (connecting to Node 5). This newly connected node (i.e., Node 3a) sends the routing table, whose node identification number is the same number used by Node 3 at the rightmost, to the neighbor nodes. Describe the spanning trees, whose roots are Node 3 and Node 3a, when the routing tables are converged after enough routing table exchanges among nodes.

(5) We sometimes intentionally put multiple nodes having the same node identification number in the Internet. Show a good usage and a bad usage of this system operation method.

Destination node	Next Hop node	Number of hops
A	*A*	*d*(*A*,*A*) = 0
B	*C*	*d*(*A*,*B*) = 3
C	*C*	*d*(*A*,*C*) = 1
⋮	⋮	⋮
Z	*B*	*d*(*A*,*Z*) = 4

Fig. 1

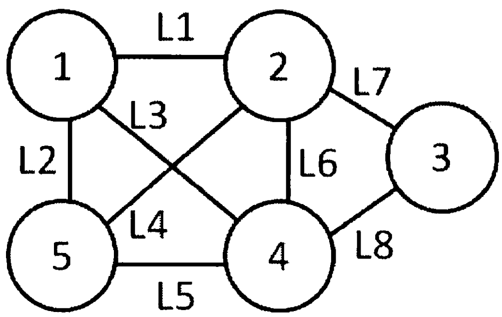


Fig. 2

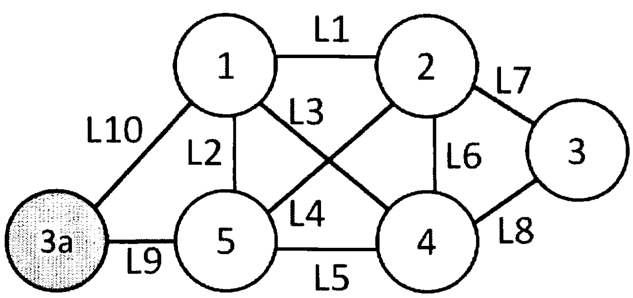
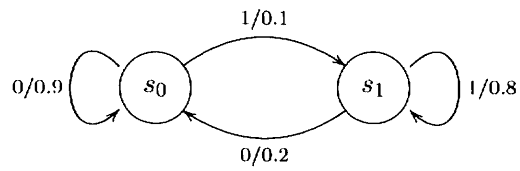


Fig. 3

Problem 5

Consider the binary first-order Markov information source described by the following state transition diagram.



Answer the following questions.

(1) Obtain the steady state probabilities w_0 and w_1 of the states s_0 and s_1, respectively.

(2) Obtain the steady state probability of the output one (i.e., 1).

(3) Assume that we observe from output sequences an arbitrary sequence of consecutive ones right after zero and right before zero (a run of ones). Obtain the probabilities that the observed run of ones has length 1, 2, and k, respectively.

(4) Obtain the average length of runs of ones.

(5) Obtain the entropy of this information source.

(6) Obtain the entropy of the information source which randomly outputs ones according to the probability obtained in (2). Discuss the difference between this value and the value obtained in (5).

(You may use $\log_2 3 = 1.58$ and $\log_2 5 = 2.32$ if necessary.)