Problems 2, 3 and 5 will be graded. There are 60 points on these problems. Note: You must do all the problems, even the non-graded ones. If you do not do some of them, half as many points as they are worth will be subtracted from your score on the graded problems.

**Problem 1.** (20 points) Chapter 8 of CS&G provides diagrams showing the network transactions for strict request-response, intervention forwarding, and reply forwarding for read operations in a flat, memory-based protocol like that of the SGI Origin (see Figure 8.12). Do the same for write operations.

**Problem 2.** (10 points) Consider a butterfly network with \( N = 2^n \) input nodes and the same number of output nodes. In this network, how many input nodes map to the same output nodes (that is, the input node number = the output node number)? The answer should be general for any value of \( N = 2^n \).

**Problem 3.** (20 points) A memory-consistency model for a shared address space specifies the constraints on the order in which memory operations can appear to execute with respect to one another, enabling programmers to reason about the behavior and correctness of their programs. The goal of this exercise is to help you practice your understandings of a few memory consistency models. Assume all values of all variables are initialized to 0.

(i) \( P_1 \) \hspace{1cm} \( P_2 \)

1. \( A = 1; \) \hspace{1cm} 1. \( \text{while } (\text{flag} == 0); \)
2. \( \text{flag} = 1; \) \hspace{1cm} 2. \( \text{print } A; \)

(ii) \( P_1 \) \hspace{1cm} \( P_2 \)

1. \( A = 1; \) \hspace{1cm} 1. \( \text{print } B; \)
2. \( B = 1; \) \hspace{1cm} 2. \( \text{print } A; \)

(iii) \( P_1 \) \hspace{1cm} \( P_2 \) \hspace{1cm} \( P_3 \)

1. \( A = 1; \) \hspace{1cm} 1. \( \text{while } (A == 0); \) \hspace{1cm} 1. \( \text{while } (B == 0); \)
2. \( B = 1; \) \hspace{1cm} 2. \( \text{print } A; \)
Problem 4. (20 points) Calculate the number of physical "wires" needed between nodes for a 4096-node system for each of the following network types. Give a general expression for the number of wires needed to connect that type of network.

(a) Hypercube network.

(b) Barrel shifter network.

(c) Calculate the diameter and average distance of each network using the formulae provided in the notes.

(d) Say we create a message-passing system that combines the mesh interconnect with the barrel shifter interconnect in a way that we now have a "hypercube of barrels"—i.e., we now have a hypercube network where each node is actually a barrel-shifter sub-network. The hypercube contains 512 nodes, and each barrel-shifter interconnect has 8 nodes. How many physical wires are required in this network?

(e) Comment on the advantages of this design versus the plain hypercube and barrel shifter designs.

Problem 5. (30 points) Answer the following questions about the eight-node shuffle-exchange network shown below.

(a) Is the shuffle-exchange network a blocking network? How do you know? Give an example based on the above eight-node network.

(b) If we set all the switches in a shuffle-exchange network to pass-through settings, the network reduces to a \((\log N)\)-stage perfect-shuffle network. After \(\log N\) shuffles, a message returns to the same node from which it came. This can be verified by the routing function of perfect shuffle interconnection as discussed in Lecture 25. An exchange permutation needs to be added to make the network a complete interconnection structure.

Given the routing function for perfect-shuffle and exchange as in Lecture 25,

- Perfect-shuffle (rotate bits circularly right): \(S((a_0, a_1, \ldots, a_n)_2) = (a_{n-1}, a_{n-2}, \ldots, a_0)_2\)
- Exchange (flip the least-significant bit): \(E((a_0, a_1, \ldots, a_n)_2) = (a_{n-1}, a_{n-2}, \ldots, a_1, \bar{a}_0)_2\)
use $N = 8$ and source-node number 6 ($110_2$) as an example. If the settings of the switches in stage 1, 2, and 3 are pass-through, exchange, and exchange, and the destination is 5 ($101_2$), discuss how the network routes the source to destination port, based on the above routing function. (*Hint:* First shuffle, then see whether an exchange is needed.)

(c) As in part (b), after a shuffle we check whether a bit needs to be flipped. So we can relate the operation of each stage of the switch to a bit position in the node number, expressed in binary. Is the following statement of routing in a shuffle-exchange network correct? Why or why not?

“Let each stage number (from 1 to $\log N$) be related to the corresponding bit position in the binary form of the source node number. Then the destination node number in binary form is determined from this rule: If the switch is set to exchange in a stage, flip the bit corresponding to that stage.”

(d) Now we want to set the switches along the path to reach a specified destination. For example, if we need to route source node 3 to destination 5, find the setting of the switch in each stage. This question is the reverse of part (c). Give a simple algorithm to do this.

(e) Find another algorithm to route from source to destination based only on the destination port number. You may change the meaning of the bits in the routing tag, if necessary (i.e., a bit may no longer have to specify whether to pass through or cross over at a given stage). *Hint:* As shown in the above figure, see the difference in routing from any source node to destination 0 ($000_2$) compared to routing a source node to destination 7 ($111_2$).