

# EECC 756 - Spring 1999

## Homework Assignment #2, Due April 29

1. A barrel shifter is a static point-to-point network topology obtained from a ring by adding extra links from each node to those nodes having a distance equal to an integer power of 2. Consider an Illiac-like (8 X 8) mesh, a binary hypercube, and a barrel shifter, all with 64 nodes, labeled  $N_0, N_1, \dots, N_{63}$ . All network links are bidirectional.
  - a) Find the bisection width for each of the three networks.
  - b) List all the nodes reachable from Node  $N_0$  in exactly three steps for each of the three networks.
  - c) Indicate for each case the tightest upper bound on the minimum number of routing steps, and the average number of routing steps needed to send data from any node  $N_i$  to any node  $N_j$ .
  
2. Topologically equivalent networks are those whose graph representations are isomorphic with the same interconnection capabilities. Prove the topological equivalence among the Omega and baseline networks (use 16 node networks to show this).
  
3. Network embedding is used to implement the topology of a network A on another network B. Explain how to perform the following network embeddings:
  - a) Embed a two-dimensional torus on an n-dimensional hypercube with  $N = 2^n$  nodes where,  $r^2 = 2^n$ .
  - b) Embed a complete balanced binary tree with maximum height on a mesh of  $r \times r$  nodes.
  
4. Estimate the effective MIPS rating of a bus-connected SMP multiprocessor system under the following assumptions. The system has 16 processors, each connected to an on-board private cache which is connected to a common bus. Globally shared memory is also connected to the bus. The private cache and the shared memory form a two-level memory access hierarchy. For a specific benchmark, each processor has a rating of 10 MIPS if a 100% cache hit ratio is assumed. On the average each instruction needs 0.20 memory access. The read access and write access are assumed equally probable. Consider only the penalty caused by shared memory and ignore all other overheads. The cache is targeted to maintain a hit ratio of 0.95. A cache access on a read hit takes 20 ns; that on a writ hit takes 60 ns with a write back scheme, and 400 ns with a write through scheme. When a block is replaced, the probability that it is dirty is estimated as 0.1. An average block transfer time between the cache and shared memory via the bus is 400 ns.
  - a) Derive the effective memory access times per instruction for the write-through and write-back searately.
  - b) Calculate the effective MIPS rate for each processor running this benchmark. Determine an upper bound on the effective MIPS rate of the 16-processor system. Discuss why the upper bound cannot be achieved by considering memory penalty alone.

5. Consider the simultaneous execution of the following three programs on three processors:

Processor 1	Processor 2	Processor 3
a. A := 1	c. B := 1	e. C := 1
b. Print B, C	d. Print A, C	f. Print A,B

Assume A, B, C, are shared writable variables in memory (initially A = B = C = 0)  
Assume atomic memory access operations. Answer the following with reasoning or supported by computer simulation results:

- List the 90 execution interleaving orders of the six instructions {a, b, c, d, e, f} which will preserve the individual program orders. The corresponding output patterns (6-tuples) should be listed accordingly.
  - Can all 6-tuple combinations be generated out of the 720 non-program-order interleavings? Justify the answer with reasoning and examples.
  - We have assumed atomic memory access in this exercise. Explain why the output 011001 for the above is not possible in an atomic memory multiprocessor system if individual orders are preserved.
- 6.
- A uniprocessor uses separate instruction and data caches with hit ratios  $h_i$  and  $h_d$ , respectively. The access time from the processor to either cache is  $c$  clock cycles, and the block transfer time between the caches and main memory is  $b$  clock cycles. Among all memory references made by the CPU,  $f_i$  is the percentage of references to instructions. Among blocks replaced in the data cache,  $f_{dir}$  is the percentage of dirty blocks. Assuming a write-back policy, determine the effective memory access time in terms of  $h_i$ ,  $h_d$ ,  $c$ ,  $b$ ,  $f_{dir}$  for this system.
  - The processor-memory system described in (a) is used to construct a bus-based shared-memory multiprocessor. Assume that the hit ratio and access times remain the same as in part (a). However, the effective memory access time will be different because every processor must now handle cache invalidation in addition to reads and writes. Let  $f_{inv}$  be the fraction of data references that cause invalidation signals to be sent to other caches. The processor sending the invalidation signal requires  $i$  clock cycles to complete the invalidation operation. Other processors are not involved in the invalidation process. Assuming a write-back policy again, determine the effective memory access time for this multiprocessor system.
7. Comment on the following choices in the design of multicomputers:
- Why were low-cost off-the-shelf processors chosen over custom-designed processors chosen as processing nodes?
  - Why was distributed memory chosen over global shared memory?
  - Why was MIMD, MPMD, or SPMD control chosen over SIMD data parallelism?

8.
  - a) Draw a 16-input Omega network using 2 x 2 switches as building blocks.
  - b) Show the switch settings for routing a message from node 1011 to node 0101 and from node 0111 to node 1001 simultaneously. Does blocking exist in this case?
  - c) Determine how many permutations can be implemented in one pass through this Omega network. What is the percentage of one-pass permutations among all permutations?
  - d) What is the maximum number of passes needed to implement any permutation through the network?
  
9. Comment on the advantages/disadvantages of constructing a system that is a hybrid of a message-passing multicomputer and a shared memory multiprocessor over a purely message-passing system or a purely shared memory system and on how this is achieved.
  
10. Using PVM: Problem 4-12 page 134 in "Parallel Programming: Techniques .." textbook.
  
11. Using PVM: Problem 4-17 page 135 in "Parallel Programming: Techniques .." textbook.