

## CSE596 Final Exam Solution

This take-home exam is to be complete and electronically submitted to Adam (carlson@cs.washington.edu) on or before 9:00 AM, Monday 16 December 2002.

**Ground Rules:** Take-home exams are conducted on the honor system. Pick a 4 hour block of time to take the test. (The actual effort required to answer the questions is under 2 hours.) Answer all questions within that time period. Do not return to it. You may use any materials and documents you wish (materials from class suffice), but please do not consult other people.

Points are shown in brackets. In all questions, show your work and explain your reasoning. Good luck!

1. [20] **ZPL Programming Exercise.** Write a ZPL code segment, including all declarations, that projects a 3D cube as follows: Assume an  $n \times n \times n$  array  $A$  (already filled with data), and compute the three 2D arrays,  $B_{nn}$ ,  $Bn_n$  and  $Bnn_$ , found by summing  $A$  along each dimension.

*Solution:*

```

config var      n = ...;
region          R1 = [1      , 1..n, 1..n];
                R2 = [1..n, 1      , 1..n];
                R3 = [1..n, 1..n, 1      ];
                R   = [1..n, 1..n, 1..n];
var            A      : [R ] integer;
                B_nn  : [R1] integer;
                Bn_n  : [R2] integer;
                Bnn_  : [R3] integer;

begin
... -- initialize A
[R1] B_nn := +<< [R] A;
[R2] Bn_n := +<< [R] A;
[R3] Bnn_ := +<< [R] A;
...
end

```

*B<sub>nn</sub>, B<sub>n</sub>, and B<sub>nn</sub> are logically two dimensional, but have been declared to be “collapsed” 3D arrays. This is because the partial reduce requires region conformance. If you wanted to have an actual 2D array for these variables, you could compute the 3d version in a Temp matrix and then copy the data by taking an array slice. While the permute operator would work, it is unnecessarily expensive.*

2. [20] **Network Routing Exercises.**

- (a) A packet is sent on a hypercube network from node [0000 0000] to node [1010 0101]. Give two different paths that the packet might take in a minimal adaptive network by listing the processors visited.

*Solution:*

*Recall that in hypercube routing, a minimal adaptive router simply selects any incorrect bit in the address and corrects it, forwarding the packet on to the appropriate node. So two possible routes are the one in which the most significant bit is corrected on each hop, and another in which the least significant bit is corrected. These routes would be as follows:*

<i>Route 1:</i>	<i>Route 2:</i>
[0000 0000]	[0000 0000]
[1000 0000]	[0000 0001]
[1010 0000]	[0000 0101]
[1010 0100]	[0010 0101]
[1010 0101]	[1010 0101]

- (b) A packet is sent on an 8x8 torus from node [0,0] to node [5,6] using Chaos router. Measuring from the time the first phit enters from the NIC to the time the last phit exits to the NIC, how many ticks are required for the transmission if in one node the packet fully enters the multiQ (but does not wait further in the MQ) and in all other nodes the packet encounters only the minimum delay?

*Solution:*

*This is an 8x8 torus, since the packet is never delayed in the MQ, it will never be derouted, so it will take a minimal path. One such minimal path is shown below:*

	0	1	2	3	4	5	6	7
0	$S \uparrow$							
1								
2								
3								
4								
5	$3 \leftarrow$						$D \leftarrow$	$4 \leftarrow$
6	$2 \uparrow$							
7	$1 \uparrow$							

*While there are only 5 hops between routers, the path length is actually 6, since the packet must go through both the initial and final nodes' routers.*

*Recall that a packet is 20 phits and the question asks for the total*

*travel time from the sending of the first phit to the arrival of the last. Since the data are pipelined through the system, we will measure the total travel time of the first phit and then add 19 ticks to allow for the streaming in of the rest of the packet at the destination node.*

*The latency for a phit to travel through the Chaos router when the MQ is not involved is 4 ticks total for the arrival of the phit, decoding the address, setting up the crossbar, moving the phit across the crossbar and sending it out the output frame. This is the case for 5 of the 6 hops. For the one router where the MQ is involved, the first phit is stalled until the entire packet arrives, that takes 20 ticks. The entire packet is then transferred to the MQ, taking another 20 ticks. Once the first phit leaves the MQ it no longer needs to wait for the rest of the packet. The overhead (crossbar setup, etc.) in this router will be overlapped with the arrival of the packet in the input frame and the movement of the packet to the MQ. The degree of this overlap depends on what else the router is doing, so we accepted any amount of overhead from 0 to 4 ticks. For our solution, let's say that only 1 tick of overhead is required. Finally, the first phit arrives in the output frame of the destination router. We must now wait another 19 ticks while the rest of the packet catches up. Totaling this all up we have:*

$$4 \text{ ticks per "normal" router} * 5 (= 20 \text{ ticks}) +$$
$$20 \text{ ticks to arrive in "holding" router} +$$
$$20 \text{ ticks to move to MQ in "holding" router} +$$
$$1 \text{ tick overhead in "holding router"} +$$
$$19 \text{ ticks for tail of packet to enter destination output frame} =$$
$$20 + 20 + 20 + 1 + 19 = 80 \text{ ticks.}$$

3. [20] **Ultracomputer Exercise.** It has been proposed that a revised Ultracomputer design can “save” the idea of shared-memory-through-combining by changing the network and memories so that there is just one location (double word) at each memory port. Each processor is also given a large private memory. Write a paragraph explaining what problems this solves and what problems it creates. Will it work?

*Solution:*

*First a note about questions 3 & 4 of the final. These questions were graded by Prof. Snyder, and while I have some understanding of what he was looking for, it's incomplete at best. If you have questions about your grade on these questions or about the answers, please contact him after the Christmas break. I apologize for not being able to give you a more complete answer. - Adam*

*In the Ultracomputer, shared memory was implemented by combining requests to similar addresses when forwarding them through the network, allowing multiple processors to efficiently address memory on other nodes. However this resulted in significant levels of false sharing, as partial addresses that didn't really match*

could be combined. One solution is to limit each memory port to a single word. This removes the possibility of any false sharing, as two requests that are addressed to the same port are guaranteed to be for the same memory location. However, it makes true sharing very unlikely and increases general contention on the network.

4. [40] **Model Exercise.** The concept of a “parallel machine model” has played an important role in this class. Explain in your own words – as if to another PMP student who hasn’t taken CSE596 – what the concept is, what is important about the idea, how it has been used, and its relationship to a “performance model.” (Your explanation can be no longer than a page, and is best if it is well organized.)

*Solution:*

*You were expected to cover four separate aspects of the parallel machine model and it’s relationship to parallel computing and the performance model (note, one could argue that some of these can be combined into one point and some should be split into two, but the basic gist is there.):*

- *The use of the parallel machine model as a basis for describing a theory of computation (analogous to the Von Nuemann model’s role in sequential computing)*
- *The ability to abstract away different physical machines characteristics such as topology or memory models (shared vs. message passing)*
- *As a basis for development of a portable high-level programming language*
- *The use of the model in developing a notion of parallel programming complexity and a model of the performance of parallel programs that takes into account important aspects such as communication costs and algorithmic complexity and allows for comparison of algorithms across the dimensions of parallel computations executed, memory accessed and communication required.*

Statistics:

The bin cutoffs are maxima, so the bin labeled 96 contains everyone who got more than a 93 and less than or equal to 96. The average grade in the class was a 3.88.

<b>Grade on final</b>	
<i>Avg</i>	92
<i>Bin</i>	<i>Frequency</i>
80	3
85	4
90	5
93	4
96	6
98	9
100	6