Section A
Compulsory Questions (60marks)

Question 1

(a). Explain the the concept of data partitioning as it applies to parallel algorithm design.

SOLUTION

Data partitioning is the process of dividing the data structures used by a parallel algorithm among the processors. The aim of the data allocation process is to minimise sharing and thus synchronisation. The appropriate divisions of data structures may change from stage to stage in the algorithm.

[6]

(b). Explain how you could use two different data partitioning schemes to divide the image array when finding the largest square of ones in an array of ones and zeroes.

SOLUTION

Divide by rows and use a wave front, or divide into lists of discrete points and use a stochastic solution method.

[8]

(c). Comment on the level of synchronisation required by the two approaches you have described and comment on the implications of that synchronisation for performance.

SOLUTION

Wavefront requires high synchronisation and yeilds at best 50% processor use in a single stage, this tends to limit its performance dramatically as synch is expensive. The stochastic method is synchronisation free, and thus despite redundant computations will yield better performance.

[4]

Question 2

What problems inherent in multi-stage interconnection networks are addressed by combining?

SOLUTION

Congestion, hotspot formation, tree saturation

[3]

Describe what happens in a combining network when a read and a write to the same address arrive on the inputs to a switch in the same network cycle.

SOLUTION

The read proceeds no further and returns the value of the write. The write continues to the memory.

[4]

Question 3

Many programming models for shared memory architectures include a concept called *non-determinism.* What does non-determinism mean when applied to program execution? Illustrate the effects of non-deterministic execution using a simple example.

SOLUTION

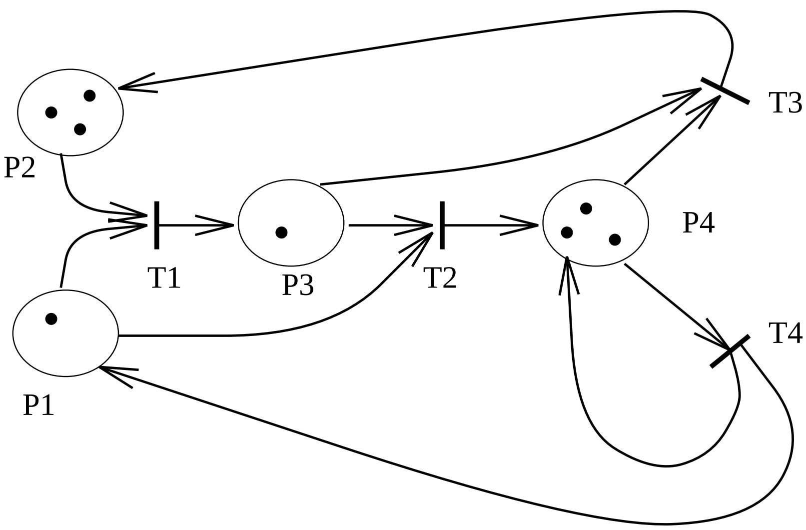
Non-determinism means that the same program run on the same data might not yeild the same result on two consecutive executions. This is possible because the global interleaving of memory accesses is not mandated by the programing model.

```
cobegin
    x:= x + 1;
□
    x:= x + 1;
coend
```

[8]

Question 4

Provide a formal definition of the following petri-net including its initial marking.



[10]

SOLUTION

$C=\{P,T,I,O,\mu\}$

$P = \{P1,P2,P3,P4\}$ $T = \{T1,T2,T3,T4\}$
$I(T1) = \{P1,P2\}$ $O(T1) = \{P3\}$
$I(T2) = \{P1,P3\}$ $O(T2) = \{P4\}$
$I(T3) = \{P3,P4\}$ $O(T3) = \{P2\}$
$I(T4) = \{P4\}$ $O(T4) = \{P1,P4\}$

$\mu = (1,3,1,3)$

Question 5

How does dataflow programming differ from conventional von Neumann style programming?

[4]

Why does this provide implicit fine grain parallelism?

[3]

Question 6

Write a Linda program which finds the largest value in a set of tuples already in the tuple-space. You should assume that the number of copies of your Linda process that execute concurrently and the number of values to be considered are stored in a tuple (D,<copies>,<num_vals>). The values are stored in tuples with the following structure (V,<val>).

HINT: Reduce the number of values in the tuple space by half in each parallel stage of the algorithm.

OVER/

[10]

SOLUTION

Write a program that iterates for $log_2(num\_vals)$ *stages. Each stage gets two values and creates an output V tuple containing the larger of them.*

Section B
Topic Questions
(Answer <u>TWO</u> questions from this section)

Question 7

Use the Modula-P language to write a k-parallel procedure which performs a rank sort on an input array of integer values. Your solution should deal correctly with duplicate values.

[30]

Question 8

Give the final petri-net model of signal and wait as derived in the lectures.

[10]

Generate the reachability tree for that network.

[15]

What formation in a reachability tree indicates that deadlock can occur? Using the tree you generated for signal and wait (or any other method) show that this model of signal and wait is deadlock free.

[5]

Question 9

Draw a diagram of the Manchester Dataflow architecture and write a paragraph describing each of its major components.

[20]

Draw a data flow graph that is equivalent to the following sequential code.

```
IF q > 12 THEN
      q = q + a;
      r := q + a + m;
ELSE
      m = 2 + q;
      r = q*4 + m;
END
```

[10]