A30844 No calculator permitted in this examination

UNIVERSITY OF BIRMINGHAM

School of Computer Science

Third Year - BSc Artificial Intelligence and Computer Science
Third Year – BEng Computer Systems Engineering
Third Year – BSc Computer Science
Third Year – MSci Computer Science
Third Year – MEng Computer Science/Software Engineering
Third Year – BSc Mathematics and Computer Science
Third Year – MSci Mathematics and Computer Science
Third Year – BSc Computer Science with Study Abroad
Third Year – BSc Computer Science with Business Management
Third Year – BSc Computer Science with Industrial Year
Third Year – MEng Computer Science/Software Engineering with Industrial Year
Third Year – BSc Computer Science with Business Management with Industrial Year
Third Year – MSci Computer Science with Industrial Year

06 26945

Distributed and Parallel Computing

Summer May/June Examinations 2016

Time allowed: 1 hour 30 minutes

[Answer ALL Questions]

1. This question is about GPU programming and the CUDA programming model.

   (a) The highly parallel processing that GPU processors specialise in compared to general purpose CPU processors leads to differing priorities for hardware architecture design choices. For the following design choices, identify whether they are typical for general purpose CPU or for highly parallel GPU processors and briefly explain why.

   (i) Large caches
   (ii) Long pipelines
   (iii) Complex ALUs

   [10%]

   (b) Explain the reasoning behind Amdahl's law and state the limitation on parallel computation that Amdahl's law imposes. **[10%]**

   (c) A CUDA GPU processor supports 8 blocks per streaming multiprocessor (SM), a maximum of 1024 threads per SM, a maximum of 512 threads per block and a warp size of 32. Consider a 2 dimensional matrix operation that is to be executed on this machine. Suppose a block size of 8×8 threads is chosen. Criticise this choice in terms of whether such a configuration would allow the kernel to launch and, if so, what, if any, performance problems it might have. **[10%]**

   (d) The following CUDA C code correctly executes a reduction in a one-dimensional configuration:

```
float A[];

for (uint stride = 1 ; stride < blockDim.x ; stride *= 2)
{
    __syncthreads () ;
    if (threadIdx.x % (2 * stride) == 0)
        A[threadIdx.x] += A[threadIdx.x + stride] ;
}
```

   However, it suffers from a significant divergence problem.

   (i) Explain, with the aid of a diagram, what the divergence problem is. **[10%]**

   (ii) Rewrite the code to minimise the divergence. **[10%]**

2. This question is about distributed algorithms

   (a) Draw the time-space diagram for a simplest possible example (i.e. smallest in number of events) that includes concurrent events a and b where $\mathrm{LC}(a) < \mathrm{LC}(b)$ for a Lamport Clock LC and indicate the clock values on the events in the diagram. **[10%]**

   (b) In a system with three processes, p, q, and r, and four messages m_1 to m_4, the following events occur in the following real time order:

   t_1: p sends m_1 to r

   t_2: r executes an internal event

   t_3: r sends m_2 to q

   t_4: p executes an internal event

   t_5: p sends m_3 to q

   t_6: q receives m_2 from r

   t_7: r receives m_1 from p

   t_8: q sends m_4 to p

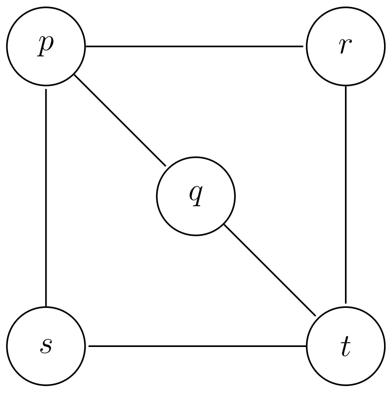
   t_9: p receives m_4 from q

   t_{10}: q receives m_3 from p

   Draw a space-time diagram of this execution and annotate each event with its vector clock value. **[10%]**

   (c) Explain how the Chandy-Lamport global snapshot algorithm on a FIFO network ensures that, for events a and b, if $a \prec b$, then a is pre-snapshot. **[10%]**

(d) Consider the following undirected FIFO network:



Taking p as the initiator, demonstrate a sample execution of the Tarry traversal algorithm by annotating the channel lines as follows:

- Add message numbers to show the order in which the messages are sent. Any total ordering compatible with the algorithm is acceptable.
- Add arrows with the message numbers to show the direction in which the messages are sent.
- Indicate the resulting spanning tree by adding arrowheads to the corresponding channel lines to show the parent relation.

[10%]

(e) Explain how the rules of the Tarry traversal algorithm guarantee that the token finally ends up at the initiator. **[10%]**