The duration of this examination is 50 minutes

Please answer *all three* questions

This examination is partly based on the nVidia Tesla architecture, as described in the article “NVIDIA Tesla: a Unified Graphics and Computing Architecture” (Lindholm et al, IEEE Micro, March-April 2008), which you should have available to you in the examination. Where the article is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

1 The nVidia Tesla supports the CUDA programming model, in which each 32-wide warp executes a simple instruction operating on a single operand per thread, one from each of the warp’s 32 lanes.

a Compare the Tesla design with an alternative approach where each 32-wide warp executes a five-wide VLIW instruction consisting of up to two arithmetic and two memory operations, together with a branch.

b Compare the Tesla design with a single thread processor core augmented with a set of SIMD instructions operating on 32-word vector registers.

c Consider, instead of the CUDA SIMT model, a hybrid vector programming approach in which four lanes of each 32-way warp are controlled by a single SIMT thread.

The three parts carry, respectively, 30%, 35%, and 35% of the marks.

2 a In Intel's Core 2 Quad, there are four cores — but each L2 cache is shared by two cores. What might be the *benefit* of each of these alternatives:

(i) A larger L2 cache, shared by all four cores.

(ii) One smaller L2 cache for each core.

b According to the Tesla article (Figure 1 and page 49) the DRAM is partitioned into six banks, each of which owns 1/6 of the address space:

(i) What happens if multiple threads within a warp try to access the same DRAM address at the same time?

(ii) What happens in Intel's Core 2 Quad if all four cores read the same array of data?

The two parts carry, respectively, 60%, and 40% of the marks.

3 Consider the following program fragment, a row-wise image filter:

```
float A[1920][1080], B[1920][1080];

for (i = 0; i < 1920; ++i)
   for (j = 2; j < 1080-2; ++j)
     B[i][j] = A[i][j-2] - A[i][j+2];
```

a Show how strip mining can be applied to this code to choose an optimum number of threads for execution on the nVidia Tesla architecture.

b The innermost loop will be executed by each SIMT thread. Will the stores to `B` be coalesced?

c Would loop interchange improve the performance of this example?

The three parts carry, respectively, 30%, 35%, and 35% of the marks.