The duration of this examination is 50 minutes

Please answer *all three* questions

This examination is partly based on the IBM Power6 processor, as described in the article "IBM POWER6 microarchitecture" (Le et al, IBM Journal of R&D, V51N6), 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 a Explain how the Power6 achieves a zero-cycle load-use penalty for FP operands.

b The Power6 processor provides some support for out-of-order execution of floating-point (FP) instructions. What might cause FP instructions to execute out-of-order?

c The Power6 is an example of a "decoupled" architecture, in which integer instructions execute ahead in order to enqueue memory operands for the floating-point instructions. A "loss-of-decoupling" (LOD) occurs when a conditional branch depends on a floating-point result. How do you think Power6 handles this case?

The three parts carry, respectively, 30%, 20%, and 50% of the marks.

2 a The Power6 is often described as an in-order architecture. How can the processor initiate several memory accesses concurrently?

b In contrast to the Power6, consider a wide-issue out-of-order processor architecture, such as the IBM Power5 or Intel Pentium 4. How can such a processor initiate several memory accesses concurrently?

c Intel's IA-64 instruction set, implemented in the in-order Itanium processor, provides instruction set support for initiating several memory accesses concurrently. How can an IA-64 processor initiate several memory accesses concurrently? What determines when the processor must actually stall?

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

3 In this question, consider the following loop:

```
declare float U[0:M, 0:N]

for t = 1 to M do
  for i = 1 to N-1 do
S:  U[t,i] = (U[t-1,i-1] + U[t-1,i+1]) * 0.5;
```

a Suppose M=4 and N=6. Draw the iteration space graph for the loop. Mark on the graph all the dependences present.

b Write down a unimodular transformation matrix which represents a valid interchange of the two loops.

c Draw the iteration space for the loop above after this transformation has been applied. Show the dependences.

d Suppose N=10,000,000 and M=20. Do you think the transformed version of this code will run faster than the original code? You should consider both memory system performance and instruction-level parallelism.

The four parts carry, respectively, 30%, 20%, 20%, and 30% of the marks.