Advanced Computer Architecture
MEng3 Test

Thursday 13th December

Answer two *questions*
You have one-and-half hours

1 This question concerns dynamic instruction scheduling in the Intel Pentium 4 processor, as described in the paper "The Architecture of the Pentium 4 Processor" (Hinton et al, Intel Technology Journal Q1 2001), which you should have available to you in the examination. **See, in particular, pages 6 and 7.** Where the paper is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

a Explain how there could be several distinct instances of a given register (such as EAX) in the processor at the same time. Illustrate your answer with reference to a sample assembly code sequence (in an instruction set of your choice).

b At what point in instruction processing is the Front-end RAT (Register Alias Table) updated? What does the new entry mean?

c At what point in instruction processing is the Retirement RAT (Register Alias Table) read?

d Under what circumstances is a physical register de-allocated?

Now, Consider the following code fragment (this is the inner loop of a matrix multiply, compiled using Microsoft Visual Studio 6.0):

```
$L170:
   fld     ST(0)            ; duplicate value at top of FP register stack
   add     ecx, 4
   fmul    DWORD PTR [ecx-4] ; multiply top-of-stack by data at location [ecx-4]
   add     eax, 4
   dec     edx
   fadd    DWORD PTR [eax-4] ; add top-of-stack and data at location [eax-4]
   fstp    DWORD PTR [eax-4] ; store top-of-stack to location [eax-4]
   jne     SHORT $L170
```

e *Estimate* the number of clock cycles between the start of execution of the first uop of the loop, and execution of the final store (`fstp`). Assume cache hits and correct branch prediction wherever necessary. Note that floating-point loads have a 6-cycle latency. Assume that the floating-point multiply latency is 4 cycles, and the floating-point add latency is 2 cycles.

f A dynamically-scheduled processor like the Pentium 4 should be able to execute two or more iterations of this loop in parallel. List four architectural constraints which limit the amount of parallelism which can be exploited from this loop. Give a very brief explanation in each case.

(The six parts carry, respectively, 15%, 15%, 20%, 20%, 10% and 20% of the marks).

2 This question concerns caching and memory access in the Intel Pentium 4 processor, as described in the paper "The Architecture of the Pentium 4 Processor" (Hinton et al, Intel Technology Journal Q1 2001), which you should have available to you in the examination. **See, in particular, pages 9, 10 and 11.** Where the paper is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

a Which address bits are used to index the level-1 (L1) cache?

b Consider the following loop:

```
for i = 1 to N
  for j = 1 to M
    B[i] = B[i] + A[j]
```

Assume that `A` and `B` are arrays of 64-bit floating-point numbers. What is the maximum value of `M` which allows the elements of `A` to be level-1 cache hits?

c Consider the following loop:

```
for i = 1 to N
  for j = 1 to M step 1024
    B[i] = B[i] + A[j]
```

Assume that `A` and `B` are arrays of 64-bit floating-point numbers. What is the maximum value of `M` which allows the elements of `A` to be level-1 cache hits?

d Consider the following loop:

```
for i = 1 to N
  for j = 1 to M step 8
    B[i] = B[i] + A[j]
```

Assume that `A` and `B` are arrays of 64-bit floating-point numbers. What is the maximum value of `M` which allows the elements of `A` to be level-1 cache hits?

e Consider the following loop, where `M` is very large:

```
for i = 1 to N
  for j = 1 to M step 16
    B[i] = B[i] + A[j]
```

Estimate the performance (in MFlops) achieved with this loop. What *simple* code transformation could dramatically improve its performance?

(The five parts carry, respectively, 20%, 15%, 15%, 15% and 35% of the marks).

3 This question concerns instruction caching and branch prediction in the Intel Pentium 4 processor, as described in the paper "The Architecture of the Pentium 4 Processor" (Hinton et al, Intel Technology Journal Q1 2001), which you should have available to you in the examination. **See, in particular, pages 4 and 5.** Where the paper is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

a Under what circumstances is the Frontend BTB used?

b How is the prediction from the Frontend BTB used?

c There could be two or more different trace cache blocks corresponding to the same IA-32 instruction address. Why?

Now, Consider a loop such as the following:

```
j = 0
for (i=1; i<N; i++) {
  j = j+1;
  if (j<M) {
    A[i] = B[j];
  } else {
    j=0;
  }
}
```

d Suppose `M=2`. Assuming a one-bit branch predictor, what branch misprediction rate would you expect? Why?

e Suppose `M=2`. Assuming a two-bit bimodal branch predictor, what branch misprediction rate would you expect? What is the worst-case behaviour?

f Suppose `M=2`. Now assume a two-level correlating $(2, 2)$ "gselect" branch predictor. What branch misprediction rate would you expect? Why?

(The six parts carry, respectively, 10%, 20%, 20%, 10%, 20% and 20% of the marks).

End of Paper