332 Advanced Computer Architecture

Tutorial exercise 8

Refer to the the article you have been given entitled

- *The Microarchitecture of the Pentium 4 Processor* by Hinton, Sager et al (Intel Technology Journal, Q1 2001, henceforth "MPP").

1. Is the Pentium 4 suitable for use in a laptop computer?
2. What is a uop?
3. The Pentium 4 can execute instructions faster than it can decode them. How?
4. What does "Branch History Update" mean in Figure 1?
5. Figure 4 shows seven functional units. What is the maximum possible number of functional units can, in principle, be active simultaneously in one cycle? See page 7.
6. The front-end bandwidth and retirement bandwidth is, apparently, 3 uops per clock. How could more than 3 uops be executed in the same cycle?
7. What is a RAT? (see Figure 5 and page 6). When is the RAT used?
8. (Page 7). ROB entries track uop status and are allocated and deallocated sequentially. What happens when a uop is retired?
9. Consider the IA-32 instruction:

   ```
   addl 12(%ebp),%edx
   ```

   This takes the register ebp, adds 12 to it and uses this address to fetch its first operand, which it then adds to the register edx. Presumably, this instruction is translated into two uops - a load and an add. The load uop (and its address calculation) is executed by the Load AGU ("address generating unit") shown in Figure 4.

   How do you think the value read from location (ebp+12) is passed from the load uop to the add uop?
10. IA32 instructions must be executed atomically from the point of view of external interrupts (though not from the point of view of other CPUs or I/O devices). How do you think the Pentium 4 ensures that external interrupts are serviced at appropriate points?
11. What questions have been omitted from this list because they appear in the exam?

Paul Kelly, Imperial College, November 2009

Notes on the solutions to Exercise 8

Refer to the the article you have been given entitled

- *The Microarchitecture of the Pentium 4 Processor* by Hinton, Sager et al (Intel Technology Journal, Q1 2001, henceforth "MPP").

1. Is the Pentium 4 suitable for use in a laptop computer?

   The chip consumes 55 Watts at 1.5GHz. This is a major drain on battery power all by itself (most laptops have a peak overall power budget of 50W or less, and that has to include disk drive, screen and motherboard). Furthermore, the chip gets hot - compare with a 60W light bulb - and must be cooled using a fan, which in turn needs more power.

2. What is a uop?

   Instead of executing IA-32 instructions directly, the processor's instruction fetch mechanism translates each instruction on-the-fly into a sequence (sometimes very short) of simple instructions. This allows the microarchitecture to focus on implementing simple instructions very fast.

3. The Pentium 4 can execute instructions faster than it can decode them. How?

   The decoder can decode only one IA32 instruction per cycle. However, the decoded instructions are cached in the trace cache. A branch to an address which hits in the trace cache leads to a 6-uop trace cache block.

4. What does "Branch History Update" mean in Figure 1?

   When a conditional branch instruction is retired, we finally know whether it was taken or not-taken. So we need to update the branch prediction tables so that this outcome is taken into account in making future predictions. Note that there are actually two branch predictors in the Pentium 4 - the 4K-entry Front-end BTB (Branch Target Buffer) and the 512-entry Trace-Cache BTB. They both need to be updated.

5. Figure 4 shows seven functional units. What is the maximum possible number of functional units can, in principle, be active simultaneously in one cycle? See page 7.

   On Page 7 column 2, the paper states that up to six uops can be dispatched per cycle. However, some SSE instructions take two cycles, so all seven functional units can, in principle, be active simultaneously. On the other hand, the two fast ALUs complete in half a cycle, allowing two uops to be executed per cycle in each fast ALU. Although the machine can't issue 9 uops in one cycle, it has the resources to execute them.

6. The front-end bandwidth and retirement bandwidth is, apparently, 3 uops per clock. How could more than 3 uops be executed in the same cycle?

   If 6 uops (suitably distributed across dispatch ports) are all waiting for the same register result (perhaps EAX), then they could all start in the same cycle.

7. What is a RAT? (see Figure 5 and page 6). When is the RAT used?

   The RAT points to the current version of each of the architectural registers such as EAX.

   The RAT is used when a uop is issued, to find where to get the correct current instance of each of its operand registers.

   It is updated with the destination register allocated for each uop as it is issued.

8. (Page 7). ROB entries track uop status and are allocated and deallocated sequentially. What happens when a uop is retired?

Retirement occurs in order; retirement is blocked until uop results are available (this includes branch outcomes). Provided no misprediction is detected, the Retirement RAT entry for the instruction's destination register(s) is updated to point to the physical register where the result(s) is.

9. Consider the IA-32 instruction:

   ```
   addl 12(%ebp),%edx
   ```

   This takes the register ebp, adds 12 to it and uses this address to fetch its first operand, which it then adds to the register edx. Presumably, this instruction is translated into two uops - a load and an add. The load uop (and its address calculation) is executed by the Load AGU ("address generating unit") shown in Figure 4.

   How do you think the value read from location (ebp+12) is passed from the load uop to the add uop?

   It needs to be passed in a register - but it can't be passed in an IA-32 register. So the decoder needs to allocate a physical register for the purpose.

10. IA32 instructions must be executed atomically from the point of view of external interrupts (though not from the point of view of other CPUs or I/O devices). How do you think the Pentium 4 ensures that external interrupts are serviced at appropriate points?

    We have to speculate. It looks like the easy way to handle external ("asynchronous") interrupts is to inject an unconditional jump into the instruction fetch stream. The translation into uops adds a complication - we mustn't inject a jump between two uops that belong to the same IA32 instruction - we have to wait for a gap (some long-running IA32 instructions are an exception to this).

    At a pinch, we might get away with servicing interrupts only at branches (which are guaranteed to be IA32 instruction boundaries). However, the trace cache packs branches and their destinations into the same trace block. Loops, of course, must always include a residual real branch.

    An external interrupt can be serviced at a time of our choosing. In contrast, an exception, such as a page fault or a divide-by-zero, has to be serviced so that from the point of view of the handler routine, the interrupt occurs at well-defined point in program execution.

(Paul Kelly, Imperial College, November 2009)