LINKÖPINGS TEKNISKA HÖGSKOLA
Institutionen för datavetenskap
Petru Eles

Tentamen i kursen

Datorarkitektur - TDDI03

2018-08-24, kl. 14-18

Hjälpmedel:

Engelsk ordbok.

Supporting material:

English dictionary.

Poänggränser:

Maximal poäng är 40.
För godkänt krävs sammanlagt 21 poäng.

Points:

Maximum points: 40.
In order to pass the exam you need a total of minimum 21 points.

Jourhavande lärare:

Petru Eles 013281396

Good luck !!!

Tentamen i kursen Datorarkitektur - TDDI03, 2018-08-24, kl. 14-18
Du kan skriva på svenska eller engelska!

1. The Pentium 4 has an L1 instruction cache which is particular in several regards. In what consists the particularity and what is the reason behind it?

(3p)

2. A set-associative cache consists of 512 (=2^9) lines, divided into two-line sets. The main memory contains 128K (=2^{17}) blocks of 128 (=2^7) bytes each. Show the format of the main memory address (give the total number of address bits, what different groups of address bits indicate, and how long each of these groups is).

(3p)

3.

a) What is the role of the page table in a virtual memory system? What data does it store?

b) The page table is very large, usually too large to be stored in main memory. Such a large size, at the same time, makes access to the page table very slow. How is this solved in current microprocessor architectures.

(3p)

4. Consider a pipelined processor with *k* pipeline stages.

a) What is the theoretical acceleration (ignoring overheads) for a sequence of *n* instructions, compared to a similar but non-pipelined processor? Show how you obtain the formula!

b) What is the acceleration of a sequence of 90 instructions if the number of pipeline stages is 16?

c) What is the acceleration for an infinitely long sequence if the number of pipeline stages is 16?

(3p)

5. Define the three types of pipeline hazards. Give an example for each.

(3p)

6. Dynamic branch prediction with a two-bit scheme. How does it work?
Illustrate with the case of a loop like the one below. Compare with one-bit prediction.

```
        ----------------
LOOP    ----------------
        ----------------
        BNZ     LOOP
        ----------------
```

(3p)

7. Delayed load. Why do we need it with RISC architectures? How does it work?
Give an example.

(3p)

Tentamen i kursen Datorarkitektur - TDDI03, 2018-08-24, kl. 14-18
Du kan skriva på svenska eller engelska!

10. What is loop unrolling? How does it work? Why is it important with VLIW architectures?

(2p)

11.

a) What is branch predication (like in the Itanium architecture)?
b) Compare with ordinary branch prediction.

(3p)

12. Formulate Amdahl's law and comment.

(2p)

13.

a) What is hardware multithreading?
b) Why do multithreaded processors provide higher performance?
c) We have described three approaches to multithreading: interleaved, blocked, and simultaneous; what is the main characteristic of each of them?

(3p)

14. What is the basic idea with the MMX extensions (multimedia extension) to the INTEL architecture?

(2p)

Tentamen i kursen Datorarkitektur - TDDI03, 2018-08-24, kl. 14-18
Du kan skriva på svenska eller engelska!

8.

a) What is a superscalar architecture?

b) Draw a block-diagram of a superscalar unit.

(3p)

9.

Consider the following sequence of machine instructions:

```
 1:  R1  ←  R0 + R2
 2:  R7  ←  R1 + R12
 3:  R5  ←  R7 - 1
 4:  R1  ←  R2 * R4
 5:  R7  ←  R9 + 10
 6:  R12 ←  R4 - 25
 7:  R3  ←  R5 * 2
 8:  R4  ←  R1 + R3
 9:  R10 ←  R8 - 2
10:  R1  ←  R1 * 3
```

a) Indicate the data dependencies among instructions.

b) Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two units that execute addition&subtraction and one unit for multiplication. Produce a table, according to the model below, showing how instructions are executed in consecutive cycles, if we assume in order execution.

c) Rename the registers in the above sequence to prevent, where possible, dependency problems. Produce a second table, according to the model below, showing how instructions (after register renaming) are executed in consecutive cycles, if we assume out of order execution.

In the table cells indicate the sequence number (1, 2, .., 10) of the instruction executed in the corresponding cycle on the respective unit.

	ADD/SUB	ADD/SUB	MUL
Cycle 1			
Cycle 2			
Cycle 3			
...			

(4p)