TEKNISKA HÖGSKOLAN I LINKÖPING
Institutionen för datavetenskap (IDA)
Zebo Peng

Tentamen i kursen
TDTS10 Datorarkitektur

Examination of the course
TDTS10 Computer Architecture

2017-04-19, 8:00-12:00

Hjälpmedel:

Engelsk ordbok.

Poänggränser:

Maximal poäng är 40.
För godkänt krävs 20 poäng.

Supporting material:

English dictionary.

Points:

Maximum points: 40.
You need 20 points to pass the exam.

Jourhavande lärare (Teacher on duty)

Arian Maghazeh, tel. 013-28 1407

Note: You can give the answers in English or Swedish.

1 Describe the instruction execution cycle (machine cycle). What are the two main phases of instruction execution? Which CPU components are used in each of these two phases?

(2 p)

2. a) What does it by mean by a memory of sequential access type?

b) Give an example of a sequential access memory.

c) Can a memory of sequential access type be used as the main memory of a computer system? Why?

(3 p)

3. Given a one-address computer with an accumulator, and the following memory values:

- Word 10 contains 40;
- Word 20 contains 60;
- Word 40 contains 15;
- Word 50 contains 20;

What values will the following instructions load into the accumulator? (Note: the keyword after LOAD in the following instructions denotes the *addressing mode*, and the number indicates the memory word-address).

a) LOAD Direct 20.

b) LOAD Indirect 10.

c) LOAD Indirect 50.

(3 p)

4. a) What is a cache memory?

b) Discuss how a cache is used to improve the performance of the memory system of a computer without greatly increasing its cost.

c) Describe the associative mapping technique used for a cache. What are the advantages and disadvantages of the associative mapping technique?

(3 p)

5. a) Why do we need a replacement algorithm in a cache memory?

b) We have discussed four replacement algorithms that can be used in a cache memory. Describe briefly each of these algorithms.

(3 p)

Note: You can give the answers in English or Swedish.

6. a) What are the motivations for using a virtual memory?
   b) How is a virtual memory organized?

(2 p)

7. a) What does it mean by interrupt-driven I/O? What are the advantages and disadvantages of this technique?

   b) Define the concept of multiple interrupts? What are the two main approaches used to handle multiple interrupts?

(3 p)

8. a) What is a data hazard in a pipelined unit? Illustrate this problem by an example and show how penalties are produced (consider a 6-stage pipeline as an example).
   b) How can this penalty be reduced with the forwarding (bypassing) technique? Draw figures to illustrate the pipelined executions without and with forwarding.

(3 p)

9. a) Discuss how the bimodal prediction technique work for branch prediction.
   b) Why does the bimodal prediction technique give better performance than the one-bit prediction method? Give an concrete example to support your argument.

(3 p)

10. a) What is microprogramming? How does it work?

    b) Microprogramming can be used to implement the control functions of a computer. What can it be used for other purposes? Please describe two other applications.

(3 p)

11. A superscalar makes use of instruction-level parallelism to improve the performance of instruction execution.
    a) How is this parallelism detected in a superscalar computer?
    b) Is this parallelism detection done by hardware or software?
    c) What mechanism is used to increase the parallelism in a superscalar architecture?

(3 p)

Note: You can give the answers in English or Swedish.

12. a) In what way an instruction set can influence the overall performance and implementation cost of a computer?

    b) Do you agree with the statement that "the more instructions a computer has, the more powerful it will be"? Why?

    (3 p)

13. What are the basic features of CICS and RISC computers, respectively? Discuss the differences the RISC and CISC machines, and the arguments for each of these two different computers. (3 p)

14. The design of RISC architectures is based on certain characteristics of program execution.

    a) What are the characteristics related to procedure calls and returns?

    b) What is the mechanism used in a RISC architecture to make procedure calls and returns efficient? How does this mechanism work?

    (3 p)