Tentamen i kursen Datorarkitektur - TDDI 68, 1998-12-18, kl. 14-18
Du kan skriva på svenska eller engelska!

9. What are the main characteristics of RISC architectures?

(2p)

9. Vad är karakteristiskt för en RISC arkitektur?

(2p)

10. What is register renaming? How does it improve the performance of superscalar computers?

(3p)

10. Vad är register renaming? Hur förbättrar det prestandan av superskalära datorer?

(3p)

11. Explain the following two policies for instruction execution in superscalar architectures:
    a) in-order issue with in-order completion;
    b) out-of-order issue with out-of-order completion.

(2p)

11. Förklara följande två principer för instruktions exekvering i superskalära ariktekturer.
    a) in-order issue med in-order completion;
    b) out-of-order issue med out-of-order completion.

(2p)

12. Enunciate Amdahl's law.

(2p)

12. Uttryck (beskriv) Amdahls lag.

(2p)

13. Both array processors and vector computers are used for speeding up computations with vectors. What is the main difference between them?

(2p)

13. Både array processors och vector computers används för att öka beräkningshastigheten vid vektor beräkning. Vilken är den huvudsakliga skillnaden mellan dem?

(2p)

14. What is the role of the mask register in a vector unit?

(2p)

14. Vilken roll har mask registret in en vektor enhet?

(2p)

Tentamen i kursen Datorarkitektur - TDDI 68, 1998-12-18, kl. 14-18
Du kan skriva på svenska eller engelska!

6. What is a data hazard in a pipelined unit? Illustrate by an example and show how penalties are produced (consider a 6 stages pipeline). How can this penalty be reduced with the *forwarding technique* (draw figures which illustrate the pipelined executions without and with forwarding).

(5p)

6. Vad är data hazard i en pipelinad enhet? Beskriv med hjälp av ett exempel och visa hur bestraffningar produceras (antag en 6 stegs pipeline). Hur kan denna bestraffning reduceras med *forwarding technique* (rita figurer som beskriver exekveringen av en pipeline utan och med forwarding).

(5p)

7. Consider the following sequence:

```
           ADD    R1,R2      R1 ← R1 + R2
           SUB    R4,R3      R4 ← R4 - R3
           MUL    R1,R5      R1 ← R1 * R5
           SUB    #2,R1      R1 ← R1 - 2
           BEZ    TARGET
           MOVE   R1,R6      R1 ← R6
           - - - - - - - - - - - - -
TARGET     - - - - - - - - - - - - -
```

Transform this sequence for a machine with delayed branching.

Show how the original sequence and the transformed one are executed in a six stages pipelined CPU, and illustrate the reduction of the delay (draw a figure which illustrates the corresponding pipelined execution).

(5p)

7. Sekvensen ovan är given.
Transformera sekvensen för en maskin med fördröjd branching.
Visa hur original sekvensen och den transformerade exekveras i en sex stegs pipelinad CPU, och beskriv reduktionen av fördröjning (rita en figur som beskriver motsvarande pipelinad exekvering).

(5p)

8. What is branch prediction? Why is it important?

(2p)

8. Vad är "branch prediction"? Varför är det viktigt?

(2p)

Tentamen i kursen Datorarkitektur - TDDI 68, 1998-12-18, kl. 14-18
Du kan skriva på svenska eller engelska!

1. Which are the main components in the structure of a von-Neumann architecture. Which are the primary functions of these components.

(3p)

1. Vilka är huvudkomponenterna i en von-Neumann arkitektur. Vilka är huvudfunktionerna i dessa komponenter.

(3p)

2. Unified caches and separate data and instruction caches: draw a picture for each of the two alternatives and comment on advantages and disadvantages.

(4p)

2. Sammanslagen cache och separat data och instruktions cache: rita en figur för vardera av de två alternativen och kommentera dess fördelar och nackdelar.

(4p)

3. Show a traditional memory hierarchy. What are the parameters according to which the hierarchy is organized.

(2p)

3. Beskriv en traditionell minneshierarki. Enligt vilka parametrar är hierarkin organiserad?

(2p)

4.

a) Why do we need special *write strategies* for cache memories?

b). We have discussed three write strategies: *write-through*, *write through with buffered write*, and *copy back*. How do they work? Which are their advantages and disadvantages?

(4p)

4.

a) Varför behöver vi speciella *skriv strategier* för cache minnen?

b). Vi har diskuterat tree skriv strategier: *skriv-genom, skriv-genom med buffrad skriv,* och *kopiera tillbaka*. Hur fungerar dom? Vilka är dess fördelar och nackdelar?

(4p)

5. *Demand paging*: how does it work?

(2p)

5. *Demand paging*: hur fungerar det?

(2p)

TEKNISKA HÖGSKOLAN I LINKÖPING
Institutionen för datavetenskap
Petru Eles

Tentamen i kursen

Datorarkitektur - TDDI 68

1998-12-18, kl. 14-18

Hjälpmedel:

Inga.

Poänggränser:

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

Resultat anslås:

Senast 1999-01-11 på IDAs anslagstavla för tentamensresultat.

Jourhavande lärare:

Petru Eles, tel 28 13 96

Good luck !!!