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

1. In a von Neumann architecture, we use a program counter (PC) to indicate the address of the instruction to be executed next. After this instruction is fetched, the program counter increases its stored value by 1, indicating that the next instruction to be executed is usually the one stored in the following memory address. Sometimes, however, the address stored in the PC will be changed during the execution of the current instruction. Why? Please give two scenarios to show the change of the PC value to something else than the previous value plus 1. Use a concrete example to illustrate each of these two scenarios. (2 p)

2. a) Reading from a cache memory does not require a special strategy; however, writing to a cache needs special policies. Why?

   b) We have discussed three write policies for cache. Describe briefly each of these policies and discuss the advantages and disadvantages of each of them, respectively.

   c) Describe the problems when applying these write policies in a multiprocessor system.

   (3 p)

3. a) Assume that we have a 8-way set associative cache for a byte-addressable memory. The cache has 128 cache blocks in total and each block consists of 16 bytes. Show how to break the following address into three fields for tag, set index, and offset:

   1000 0101 1000 0101 0001 0011 1001

   b) Describe how these three fields are used to locate the content stored in the address in a step by step manner.

   (3 p)

4. a) What does it mean by virtual memory? Describe how a virtual memory works.

   b) Why is it not needed to have all of the pages of a program to be in the main memory while the program is being executed?

   c) How is a logical (virtual) address converted into a physical address of the main memory?

   (3 p)

5. a) Why is instruction pipeline *widely* used in a modern computer?

   b) What are the three main types of hazards that can reduce the performance of an instruction pipeline?

   c) Give an example of each hazard type and use it to illustrate the impacts of each of these hazards.

   (4 p)

6. 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?

c) Give a concrete example to support your argument for b).

(3 p)

7. a) Define the concepts of superscalar and superpipelined architectures.

b) What are the differences between these two architectures?

c) Compare these two architectures to each other, and discuss their advantages and disadvantages, respectively.

(3 p)

8. a) Identify all the different types of data dependencies in the following code. Indicate the type of dependency you have identified for each one, and give the reasons for your answers.

```
L1: move r3,r4        Note: r3 <- r4
    load r8,(r3)      Note: r8 <- memory location pointed by r3
    add r4,r3,4       Note: r4 <- r3 + 4
    load r9,(r4)      Note: r9 <- memory location pointed by r4
    ble r8,r9,L1      Note: branch to L1 if r8 <= r9
```

b) Which of the identified dependencies can be eliminated? How?

(3 p)

9. a) What is the purpose of the window of execution in a superscalar architecture?

b) Discuss the impact of the size of the window of execution with respect to performance and cost in details.

(3 p)

10. a) Why is the placement of the “load from memory” operations an important issue?

b) Describe the speculative loading technique used in VLIW processors. What are the advantages of this technique?

c) Illustrate the speculative loading technique with a simple example.

(3 p)

11. a) There are two basic approaches to implement a snoopy protocol: write-invalidate and write-update. How do they work, respectively?

    b) Describe the situation when the write-invalidate approach works better, and the situation when the write-update works better, respectively.

    c) Both these approaches suffer from false sharing overheads. What does it mean by false sharing here?

    (4 p)

12. a) Describe the different multithreading approaches and discuss how they are applied in the context of a superscalar architecture. What are the advantages and disadvantages of these different approaches, respectively?

    b) Why does multithreading improve system performance even in the case when there is only a single scalar processor in your computer?

    (3 p)

13. a) Which features of a graphics processing unit (GPU) have contributed to its high performance? Why?

    b) Discuss the concept of divergent execution in a GPU processor. What is the main impact of such divergent execution?

    c) Discuss one technique that can be used to address the divergent execution problem.

    (3 p)