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

1. The program counter (PC) is used to indicate the address of the instruction to be executed next. In many computers, after this instruction is fetched, the program counter increases its value by 1, indicating that the next instruction to be executed is usually the one stored in the following memory address.

   a) In some computers, however, the PC increases its value by 4, after each instruction fetch. Why?

   b) During the execution of the current instruction, the next instruction address stored in the PC may also be changed. Why? Give two scenarios to show why the PC value needs to be changed during the instruction execution. Use a concrete example to illustrate each of these two scenarios.

   (3 p)

2. a) In a computer, the cache's access time is 2 ns, while the main memory's access time is 20 ns. Assume that the cache block size (line size) is 8, the cache hit ratio is 0.98, and the time needed to check for cache hit/miss is 0.5 ns. What is the average access time of this memory system?

   b) Discuss all the assumptions that must be valid for your above calculation to be correct.

   c) Describe a method that can be used to decrease the average access time of this memory system. Discuss also the implication on the power consumption of this memory of the method described.

   (4 p)

3. a) Explain the three types of hazards (data, structural, and control hazards) in pipelined architectures.

   b) For each type, provide a method used to mitigate the hazard and discuss the trade-offs associated with each method.

   (4 p)

4. A computer has an instruction pipeline that has sex pipeline stages. You are considering the possibility of increasing the number of pipeline stages in order to improve this computer's performance. It turns out that the stage with the longest execution time cannot be divided into two or several shorter stages. Does it make sense to divide the other five stages so that the total number of stages increases? Why?

   (2 p)

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

5. Consider the following instruction sequence:

```
L1: ADD R3,R5          Note: R3 <- R3 + R5
    MUL R1,R6          Note: R1 <- R1 * R6
    BLT R1,#16, L2     Note: Branch to L2, if R1 > 16
    SUB R2, R4         Note: R2 <- R2 - R4
    ...
L2: ...
```

Transform this sequence in order to make use of the delayed branch technique. Show how the original sequence and the transformed one are executed in a pipelined CPU with two pipeline stages, and illustrate the reduction of the delay (draw two figures to illustrate the corresponding pipelined executions of the original sequence and the transformed sequence, respectively).

(3 p)

6. a) Explain the concept of microprogramming in the design of a control unit.

b) Compare hardwired control units with microprogrammed control units, outlining the advantages and disadvantages of each approach in terms of flexibility, speed, and implementation complexity.

c) Given a system that requires frequent updates to the instruction set (e.g., a system supporting multiple instruction set architectures), which control unit design would you recommend? Justify your answer with respect to reconfigurability and performance.

(3 p)

7. a) Describe the VLIW architecture and compare it with the superscalar architecture. How does each approach handle instruction parallelism, and what are the implications for compiler design?

b) Compare superscalar architecture with VLIW architecture from a performance point of view. Which architecture gives usually better performance? Why?

c) Compare superscalar architecture with VLIW architecture from a power consumption point of view. Which architecture consumes usually less power? Why?

(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) In a SIMD architecture (e.g., a vector processor or an array processor), there is usually the masked execution mode. What does it mean by this. Why do we need this mode?

   b) What is the mechanism used in a vector processor to implement the masked operations?

   c) Give a concrete example to illustrate the use of the masked operations.

   (3 p)

10. a) Define the concept of loop unrolling. Why is loop unrolling very useful for a VLIW processor?

    b) Can the loop unrolling technique be used in a superscalar processor? Why?

    c) If a loop is unrolled completely (i.e., there is no need for loop control any longer), what will happen? Discuss the negative side effects of unrolling a loop completely?

    (3 p)

11. a) Describe and compare inclusive and exclusive cache architectures in a multi-level cache hierarchy, particularly for L1 and L2 caches.

    b) What are the key advantages and disadvantages of each organization in terms of cache coherence, hit/miss ratios, and overall system performance?

    (3 p)

12. a) What does it mean by the globally asynchronous, locally synchronous (GALS) design strategy?

    b) Describe all the advantages of using the GALS strategy. You should explain why GALS has each of the advantages you have described.

    (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)