# How exam-questions can look like?

1. Why is the memory system of a computer organized as a hierarchy?
   What are the basic elements of a memory hierarchy?

2. Consider 7 instructions each with execution time $T_{ex}$. They are executed by a 6 stage pipeline. Pipeline overheads are ignored. How long does it take to execute the 7 instructions by the pipelined CPU (we suppose that there are no hazards)? Draw a figure.

3.
   a) What are pipeline hazards?
   b) Enumerate and briefly present the three types of pipeline hazards.

4. Data hazards in pipelines can sometimes be avoided by a technique called *forwarding*. How does this technique work? Give an example in which forwarding produces an acceleration (draw a figure which illustrates the corresponding pipelined execution).

5. Consider the following sequence:

```
           MUL    R3,R4      R3 ← R3 * R4
           SUB    #1,R2      R2 ← R2 - 1
           ADD    R1,R2      R1 ← R1 + R2
           BEZ    TARGET
           MOVE   #10,R1     R1 ← 10
           - - - - - - - - - - - - -
TARGET     - - - - - - - - - - - - -
```

   Transform this sequence in order to use delayed branching.
   Show how the original sequence and the transformed one are executed in a pipelined CPU, and illustrate the reduction of the delay (draw a figure which illustrates the corresponding pipelined execution).

6. Static branch prediction:
   How does it work? Show three alternative approaches.

7. What are the main characteristics of RISC architectures?

8. Data dependencies:
   Enumerate the three types of data dependencies and give an example for each.

9. Flynn’s classification of computer architectures:
   Give the definitions of the alternative architectures.

10. Which are the main characteristics of the two mapping algorithms for cache memories: *direct* and *associative*? Illustrate with a figure for each one.

11. How does the ratio of sequential computations influence the speedup obtained with a parallel computer? Amdahl's law.

12. Consider a sequence of eight instructions and show how it passes through a four and six stage pipeline respectively (draw a figure for each of the two situations). How many clock cycles are needed in each case? In which case is the execution time of the sequence shorter? Can we conclude that an increasing number of stages always provides increasing performance? Bring arguments. (Consider that no hazards occur).

13.

a) Describe the strategy for static branch prediction depending on the branch direction.

b) Describe the strategy for dynamic branch prediction with an one-bit scheme.

c) Compare how the two approaches work in the case of a loop like the one below:

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

14.

a) Give an example with o*utput dependency* and another one with *antidependency.* Show how they can be solved by *register renamimg*.

b) Which data dependencies have to be considered by a superscalar CPU using:
   - in-oder issue with in-order completion?
   - out-of-order issue with out-of-order completion?

15. Both *vector processors* and *array processors* are specialized to operate on vectors. What is the main difference between them?

16. Explain how virtual memory functions. Illustrate by a figure.

17. What is the purpose of instruction pipelining? How does a pipelined unit work?

18.

a) What is a data hazard in a pipelined unit? Give also an example.

b) Explain how some data hazards can be avoided by *forwarding* (*bypassing*)?

19.

a) Why is branch prediction important?

b) Illustrate your answer by showing how a certain instruction sequence, of your choice, passes a pipelined unit, in the case of a correct and in that of an incorrect prediction.

20. The design of RISC architectures is based on certain characteristics of currently used programs. Enumerate at least five such characteristics.

21. 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.

22. Flynn’s classification of computer architectures: give the definition of the alternative architectures and draw for each one a block diagram.

23. Can you always improve the efficiency of parallel execution by decomposing the computation in a very large number of small processes which are executed each one on a processor? What is very important to be taken into consideration? Motivate.