1 Representation of numbers (4 points)

There are several ways to represent signed integers using bits. In computer systems, the two most frequently encountered are sign and magnitude and two’s complement.

A is an integer which $n$ bits sign and magnitude representation is $a_{n-1}a_{n-2} \ldots a_1a_0$, where $a_0$ is the Least Significant Bit (LSB) and $a_{n-1}$ is the sign bit.

B is an integer which $n$ bits two’s complement representation is $b_{n-1}b_{n-2} \ldots b_1b_0$, where $b_0$ is the Least Significant Bit (LSB) and $b_{n-1}$ is the Most Significant Bit (MSB).

1. [1/2 point] Under which condition can the $n$ bits sign and magnitude representation of $A$ be reduced to $n - 1$ bits without changing the value it represents?

2. [1/2 point] How would you change the $n$ bits sign and magnitude representation of $A$ to divide it by 8 (integer division, rounding towards 0), but still representing it on $n$ bits?

3. [1/2 point] How would you modify the $n$ bits sign and magnitude representation of $A$ to obtain its $n$ bits two’s complement representation?

4. [1/2 point] If $n = 10$ what is the range of values that $A$ can take?

5. [1/2 point] How can the $n$ bits two’s complement representation of $B$ be extended to $2 \times n$ bits without changing the value it represents?

6. [1/2 point] How would you change the $n$ bits two’s complement representation of $B$ to divide it by 16 (integer division, rounding towards $-\infty$, but still representing it on $n$ bits?)
7. [1/2 point] If \( n = 9 \) what is the range of values that \( B \) can take?

8. [1/2 point] Is the usual decimal representation of integers (the one you use every day) closer to sign and magnitude or two’s complement? Why?

2 Hardware support for Operating Systems (3 points)

2.1 Memory Management Unit (MMU)

1. [0.5 point] Briefly explain what a Memory Management Unit (MMU) is and what it does.

2. [0.5 point] Is a MMU absolutely mandatory to run a full-featured Operating System like GNU/Linux or is it there for performance reasons only? Explain.

2.2 Hardware support for Operating Systems: timers

1. [0.5 point] Explain what a timer is.

2. [0.5 point] In order to run a full-featured Operating System (OS) like GNU/Linux, a timer is mandatory. Why? Provide examples of services that an OS cannot offer without a timer.

3. [1 point] Give at least two other examples of hardware supports without which a timer alone cannot be fully exploited by the OS. Explain.

3 Caches (9 points)

3.1 Cache coherence

1. [1 point] Briefly explain what cache coherence is

2. [1 point] What kind of computer systems do not need cache coherence?

3. [1 point] Give an example of a cache coherence protocol, list its different states and for each state briefly explain to which situation it corresponds

3.2 Replacement policy

An engineer responsible for the hardware design of a 3-ways set-associative cache is asked to implement a Least Recently Used (LRU) replacement policy. He decides to use 3 bits per set for the LRU management.

1. [0.5 point] Do you think 3 bits are enough? Why?

2. [0.5 points] Assuming you are asked to do the same how many bits per set would you use?

3. [0.5 point] How would you initialize your LRU management information?

4. [0.5 point] What would happen to your LRU management information when a line is filled for the first time?
5. [0.5 point] What would happen to your LRU management information when a line is accessed?

6. [0.5 point] What would happen to your LRU management information when a line is evicted?

3.3 Cache architecture

We consider a computer system based on a 64-bits CPU with a unique (quite) large cache where:

- Addresses are byte addresses
- The bit-width of addresses is 48 bits
- The bit-width of the words stored in memory and cache is 64 bits
- The cache is 3-ways set-associative, with 32 words per block
- The cache is write-back, write-allocate
- There are $2^{16} = 65536$ sets

Ignore the replacement policy and answer the following questions:

1. [1 point] What is the breakdown of a 48-bits address?

2. [0.5 point] What is the breakdown of a cache line?

3. [0.5 point] What is the breakdown of a cache set?

4. [1 point] What is the total size (in bits) of the cache?

4 Pipe-line hazards (4 points)

4.1 Pipeline data hazards

1. [1 points] Briefly explain what pipeline data hazards are.

2. [1.5 points] In order to mitigate pipeline data hazards several techniques can be used. Some aim at guaranteeing the correct behavior of the system, on a pure functional point of view. Give an example of such a technique, briefly explain how it works and why it contributes guaranteeing the correct behavior.

3. [1.5 points] In order to mitigate pipeline data hazards several techniques can be used. Some aim at improving the performance. Give an example of such a technique, briefly explain how it works and why it contributes improving the performance.

---

1 The breakdown of a data structure is the partitioning of the data structure in individual fields and, for each field, its bit-width and its role.