Question 3 ISA vs Microarchitecture

Fundamentally speaking, if a property belongs to the ISA, it is something that the programmer needs to be aware of when writing code and can be said to belong to microarchitecture otherwise. Microarchitecture is the implementation of the ISA and refers to anything done in hardware without exposure to software i.e. not something the programmer can explicitly control.

1. Memory is little-endian: ISA as programmer needs to know how data is stored in memory to access and manipulate it.
2. Processor frequency adjusting automatically based on workload: Microarchitecture as the programmer does not need to know how the processor executes instructions, as long as it does so in a generally expected manner i.e. sequentially.
3. 101010 represents an ADD instruction: ISA as the programmer needs to know the instruction encoding to write or understand assembly code.
4. 32 general-purpose registers: ISA as the programmer needs to know how many registers are available for use when writing code.
5. Machine is 64-bit: ISA as the programmer needs to know the word size for data types and operations.
6. Machine uses direct addressing mode: ISA as the programmer needs to understand how to access memory locations.
7. L1 cache size is 32KB: Microarchitecture as how the cache is being implemented and accessed is opaque to the programmer.
8. Processor dynamically adjusts clock frequency to lower heat: Microarchitecture as this is likely managed through hardware mechanisms.
9. Processor has 5-stage pipeline: Microarchitecture as pipelining is also handled by the hardware without programmer intervention.
10. SUB instruction only takes memory address as inputs: ISA as the programmer needs to know the instruction's operand requirements when writing code.