Question 1: How is a machine instruction fetched from main memory to the CPU, and what determines the order of execution for instructions?

Answer: A machine instruction is fetched from main memory to the CPU by transferring it using a bus. The order of execution for instructions is determined by the CPU's program counter, which contains the address of the next instruction to be executed. Instructions are executed in the order they are stored in memory unless specified by a JUMP instruction.

Question 2: In the context of data manipulation, how is the AND operation typically used, and what is the significance of masking?

Answer: The AND operation in data manipulation is often used to place 0s in one part of a bit pattern while not disturbing the other part. This process is called masking, where one operand, called a mask, determines which part of the other operand will affect the result. For example, if you AND a byte with a mask of 00100000, it will produce a byte of all 0s if and only if the third bit from the high-order end of the bit map is itself 0. Masking allows you to selectively manipulate specific bits in a bit pattern.

Question 3: What is the purpose of rotation and shift operations, and how are they classified?

Answer: Rotation and shift operations are used to move bits within a register and are often employed to solve alignment problems. These operations are classified based on the direction of motion (right or left) and whether the process is circular. Circular shifts involve moving bits such that they wrap around, while logical shifts discard bits that fall off the edge and fill the hole with 0. Logical shifts can be used for multiplication and division by powers of two.

Question 4: How can subtraction be simulated using addition in 2's complement notation, and what are the key considerations when handling numbers in 2's complement notation?

Answer: Subtraction can be simulated by addition in 2's complement notation by adding the binary of the first number to the negation of the second number. When working with numbers in 2's complement notation, it is straightforward for addition. However, it's essential to consider sign extension when performing arithmetic operations to preserve the sign bit. Additionally, for floating-point notation, one needs to account for the mantissa, exponent, and sign bit to perform arithmetic operations correctly.