1. What is the difference between sequential and combinational circuits?
   Combinational circuits
   - The output only depends on the present input.
   - It is faster.
   - It has no memory element.
   - There is no feedback between input and output.
   - This is time-independent.

   Sequential circuits
   - The output depends on present previous inputs.
   - It is slower.
   - It has a memory element that stores the previous outputs.
   - There is a feedback between input and output.
   - They are time-dependent.
2. List applications of sequential and combinational circuits.
   Combinational circuits
   - They are used to implement the ALU in a CPU.
   - They are used in digital signal processing.
   - They are also used in digital calculators.
   - They are also used to access a particular location in a memory cell.

   Sequential circuits
   - They are used to implement different memory elements in a computer.
   - They are used to provide timings for any kind of delta required.
   - They are used in processors.
   - They are also used in various control systems.
3. Why does a sequential circuit have a memory element?
   It is used to store the past inputs and internal states to determine the next output. This lets the sequential circuits respond to both present and past inputs.
4. What is an asynchronous sequential circuit?
   An asynchronous sequential circuit is a digital logic circuit that changes its state and output values in response to input changes and doesn't use a clock signal to synchronise its components.
5. What are the two types of asynchronous sequential circuits?
   The two types are Mealy and Moore circuits.
6. What are the types of hazards in an asynchronous sequential circuit?
   In asynchronous sequential circuits, hazards refer to situations where the circuit might behave unpredictably or incorrectly due to timing issues
   - Static Hazard: This occurs when a signal that should remain constant changes due to glitches or transient pulses.
   - Dynamic Hazard: This type occurs when a signal changes multiple times during a transition.
   - Critical Race: A critical race occurs when the final state of the circuit depends on the order in which the input changes occur, leading to unpredictable results.
7. What is race condition and How to avoid it?
   A race condition in an asynchronous sequential circuit occurs when the final state of the circuit depends on the order or timing of input changes, leading to unpredictable behaviour. This happens because asynchronous circuits do not have a global clock to synchronize their operations, so the order in which signals propagate through the circuit can affect the outcome.
   To avoid race condition
   - We need to design the circuits with careful timing analysis and make sure such a state doesn’t occur.
   - Design and circuit without hazards to handle hazards.

- We can also synchronise the inputs to make sure they are stable and synchronised before reaching the circuit.

8. What is the difference between race condition and deadlock?
   Race condition
   A race condition occurs when the outcome of a process depends on the timing or order of uncontrollable events, such as the order in which threads or processes access shared resources. In a race condition, the system can exhibit unpredictable behaviour or produce incorrect results due to the non-deterministic sequence of operations.
   Deadlock
   A deadlock occurs when two or more processes or threads are each waiting for the other to release the resources they need, creating a situation where none of them can proceed. In a deadlock, the processes are effectively stuck in a cycle of waiting, preventing any of them from making progress.
9. How do you prevent deadlocks?
   - Ensure that at least one of the necessary conditions for deadlock (mutual exclusion, hold and wait, no preemption, and circular wait) is not satisfied.
   - Implement algorithms to detect deadlocks and take action to recover from them, such as terminating processes or rolling back operations.
   - Use techniques like resource ordering or the Banker's algorithm to manage resource allocation and avoid circular wait conditions.
10. What is a glitch? How to avoid and fix it?
    A glitch in sequential circuits, especially in digital electronics, refers to an unintended, brief change in the signal value that can cause temporary erroneous behaviour or incorrect outputs. Glitches are mostly caused by timing issues, propagation delays or multiple signal paths with different delays. To avoid glitches.

    - For circuits receiving signals from mechanical switches or similar sources, use debouncing techniques to eliminate glitches caused by bouncing contacts.
    - In synchronous sequential circuits, ensure that all flip-flops are driven by the same clock signal to synchronize the timing of state transitions.
    - Use two flip-flops in series to synchronize an asynchronous input signal. This technique reduces the chance of glitches propagating through the circuit.