DAY 6-111 DAYS VERIFICATION CHALLENGE

Topic: FSM

Skill: Digital Electronics

DAY 6 CHALLENGE:

1. What is Finite state machine? What are different types of FSMs?

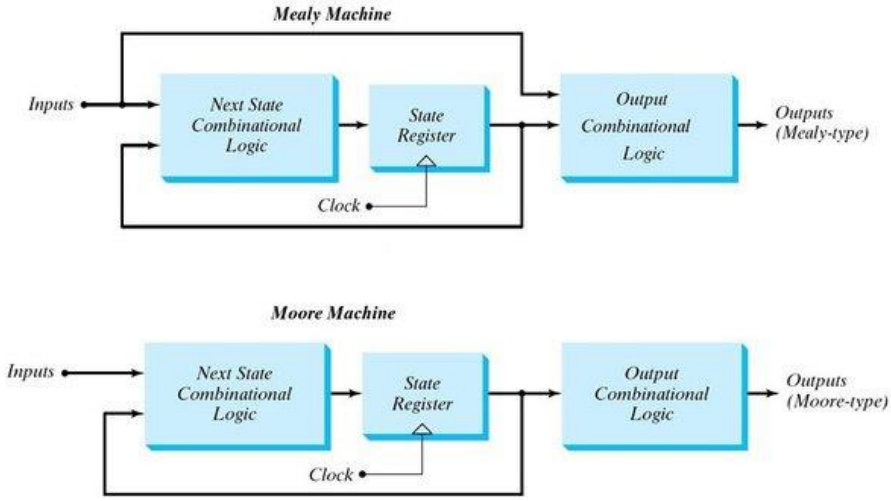
A Finite State Machine (FSM) is a computational model used to design algorithms and understand the behaviour of systems. It consists of a finite number of states and transitions between those states based on input events. An FSM can be in only one state at a time, and it changes from one state to another in response to external inputs. This model is widely used in computer science, electrical engineering, linguistics, and other fields to represent and control execution flow in systems and processes.

1. Deterministic Finite Automaton (DFA)
2. Nondeterministic Finite Automaton (NFA)
3. Mealy Machine
4. Moore Machine

2. What is the difference between Moore & Mealy machine?

Moore Machine	Mealy Machine
Output depends only upon the present state.	Output depends on the present state as well as present input.
More states are required.	Less number of states are required.
There is less hardware requirement for circuit implementation.	There is more hardware requirement for circuit implementation.
They react slower to inputs(One clock cycle later).	They react faster to inputs.

Synchronous output and state generation.	Asynchronous output generation.
Easy to design.	It is difficult to design.
If input changes, output does not change	If input changes, output also changes.



3. What are applications & limitations of FSM?

Applications of FSMs

- Control Units: Manage microprocessor operations.
- Counters & Registers: Design sequential circuits.
- Communication Protocols: Ensure data transmission.
- User Interfaces: Handle button debouncing and UI states.
- Signal Processing: Control digital filters and modulators.
- Memory Devices: Manage read/write operations.

Limitations of FSMs

- Scalability: Hard to manage many states.
- State Explosion: Exponential growth in states.
- Memory: Limited past state memory.
- Complexity: Can become cumbersome in large systems.
- Inflexibility: Challenging to modify for new requirements.

4. How do you handle exceptions or error states in a Finite State Machine?

- Define specific error states.
- Implement transitions to error states based on error conditions.
- Include mechanisms for error recovery.
- Communicate errors through outputs or notifications.
- Restore the FSM to a valid state after handling the error.

5. Explain below concepts in FSM:

I. Trigger

A trigger is an event or condition that causes a transition from one state to another in a Finite State Machine.

II. Transition

A transition represents a change of state in response to a trigger or input.
It defines the conditions under which the FSM moves from one state to another and may include actions or outputs associated with the transition.

III. State Diagram

A state diagram visually represents the states of an FSM, the transitions between states triggered by inputs or events, and the actions or outputs associated with each transition.

IV. Super state

A super state is a higher-level abstraction in state diagrams that groups a set of related states and transitions into a single entity.
It simplifies the representation and management of complex FSMs by organizing states into hierarchical structures.

V. Guard conditions

Guard conditions are Boolean expressions associated with transitions in FSMs that must be true for the transition to occur.
They allow FSMs to make decisions based on specific conditions, enhancing the flexibility and control flow of the system.

VI. Deadlock

Deadlock in FSMs occurs when the system enters a state where no further transitions are possible, preventing progress or recovery.

VII. Determinism

Determinism in FSMs means that for every state and input, there is exactly one next state.

VIII. epsilon transitions

Epsilon transitions (ε-transitions) are special transitions in FSMs that occur without consuming any input.