

Lesson 10

Digital Logic

Junying Chen

Synchronous Sequential Logic Design

- Breaks cyclic paths by **inserting registers**
- Registers contain **state** of the system
- State changes at clock edge: system **synchronized** to the clock
- **Rules** of synchronous sequential circuit composition:
 - Every circuit element is either a register or a combinational circuit
 - At least one circuit element is a register
 - All registers receive the same clock signal
 - Every cyclic path contains at least one register
- Two common synchronous sequential circuits
 - Finite State Machines (FSMs)
 - Pipelines

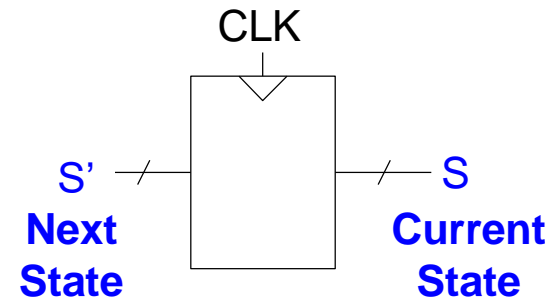


Finite State Machine (FSM)

- Consists of:

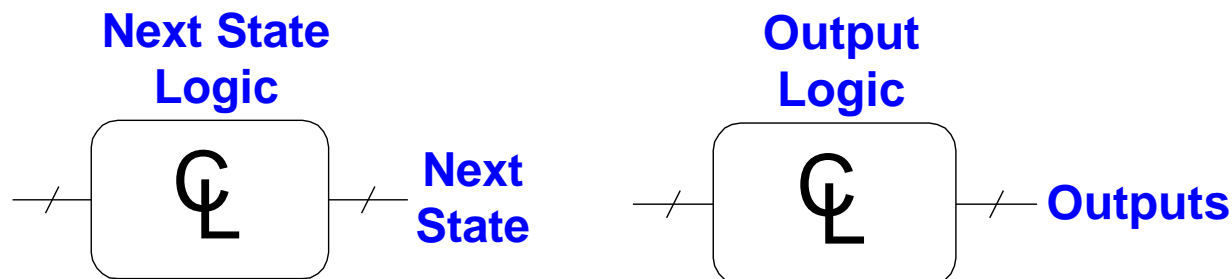
- State register**

- Stores current state
 - Loads next state at clock edge



- Combinational logic**

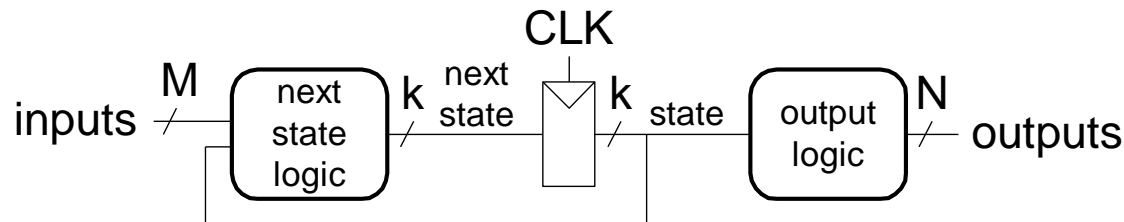
- Computes the next state
 - Computes the outputs



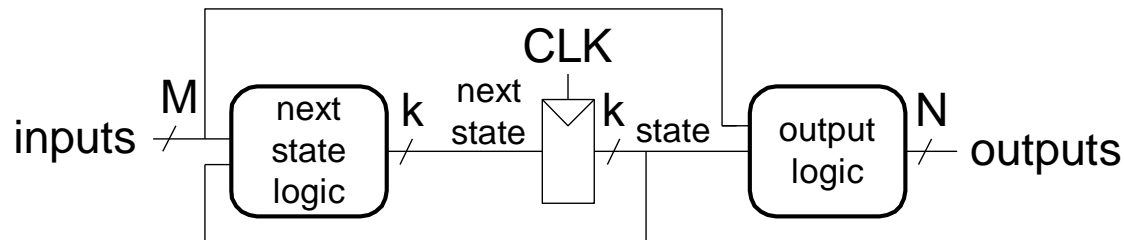
Finite State Machines (FSMs)

- Next state determined by current state and inputs
- Two types of finite state machines differ in output logic:
 - **Moore FSM:** outputs depend only on current state
 - **Mealy FSM:** outputs depend on current state *and* inputs

Moore FSM

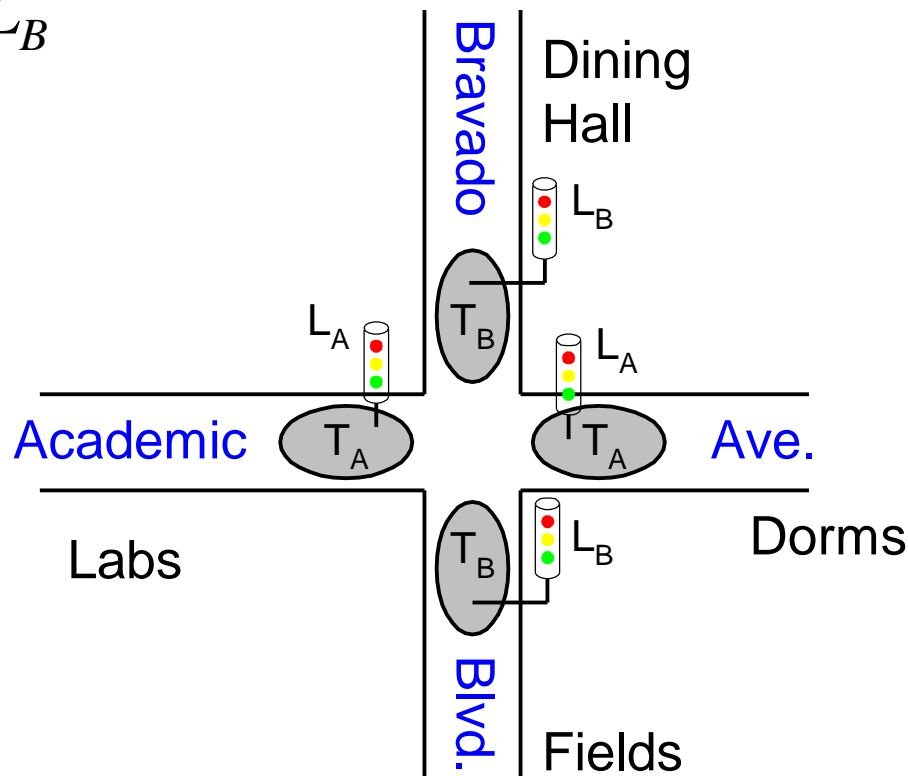


Mealy FSM



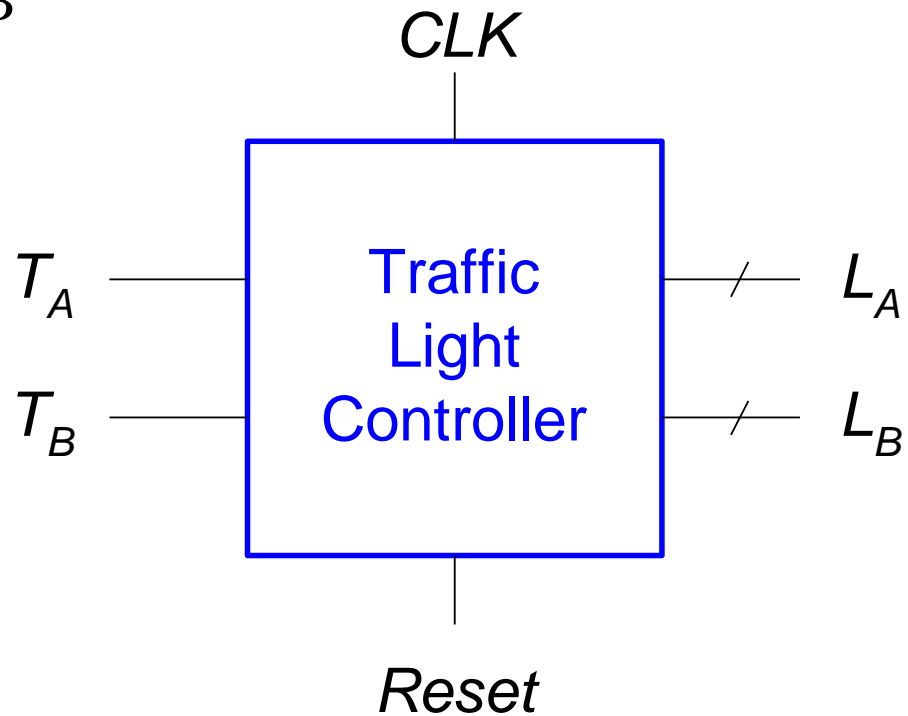
FSM Example

- Traffic light controller
 - Traffic sensors: T_A , T_B (TRUE when there's traffic)
 - Lights: L_A , L_B



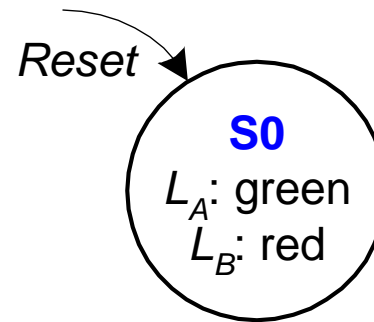
FSM Black Box

- Inputs: CLK , $Reset$, T_A , T_B
- Outputs: L_A , L_B



FSM State Transition Diagram

- **Moore FSM:** outputs labeled in each state
- **States:** Circles
- **Transitions:** Arcs



FSM State Transition Diagram

- **Moore FSM:** outputs labeled in each state
- **States:** Circles
- **Transitions:** Arcs

