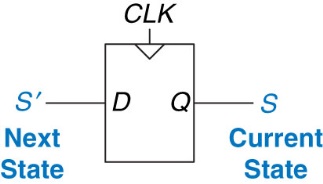
**Lesson 15 – Finite State Machines (FSM) - Design**

From Last Time:

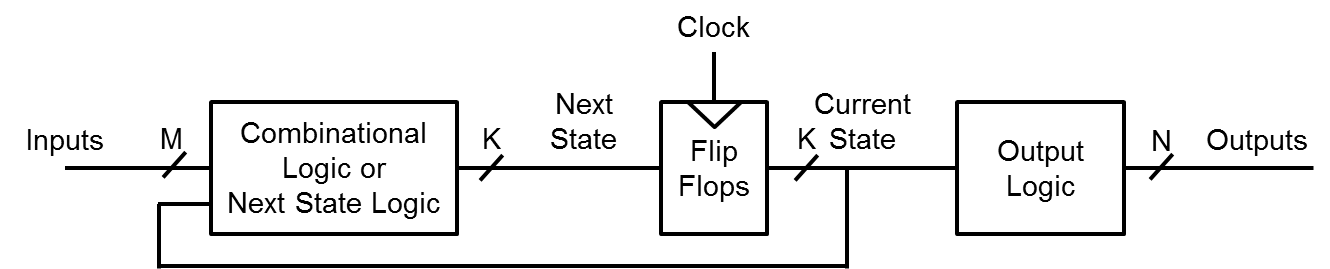
* What do Flip-Flops Create? Memory!
* When do they do this? On Clock Edge (rising vs falling)

**Finite State Machine (FSM) Makeup: M- inputs, N – Outputs, and K - bits of state**

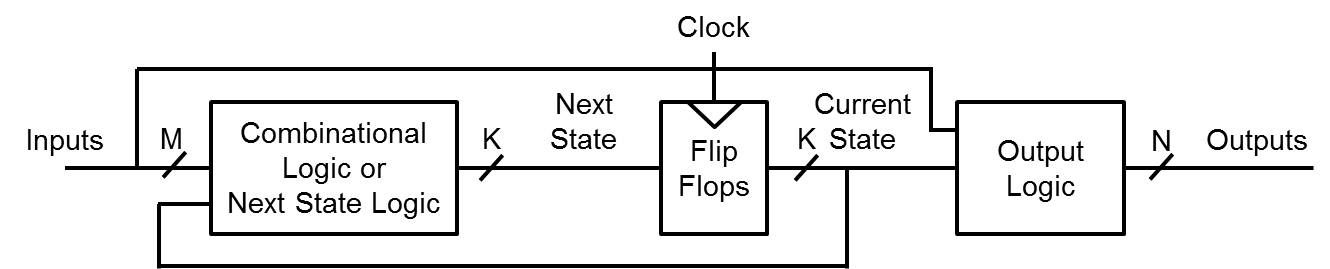
* FSMs have K registers that can be one of a finite number (2K) unique states

**Two Types of FSMs:**

**Moore Machine** – outputs depend only on current state of the machine.



**Mealy Machine** – outputs depend on both the current state and current inputs of the machine.

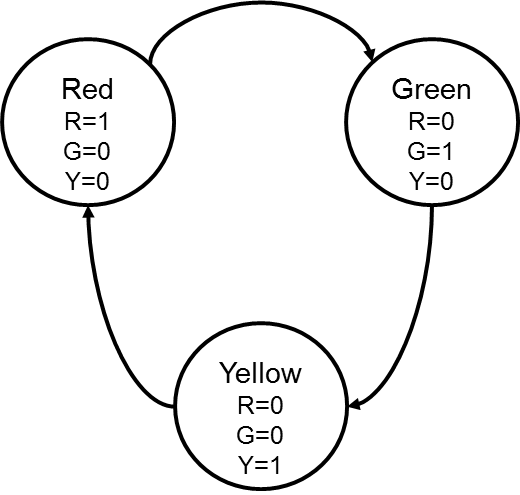


Real Life Examples of State Machines:

* Soda Machine
* Software Applications
* Spell Checker – Using Soundex
* Driving to School/Work

**How to Design a State Machine:**

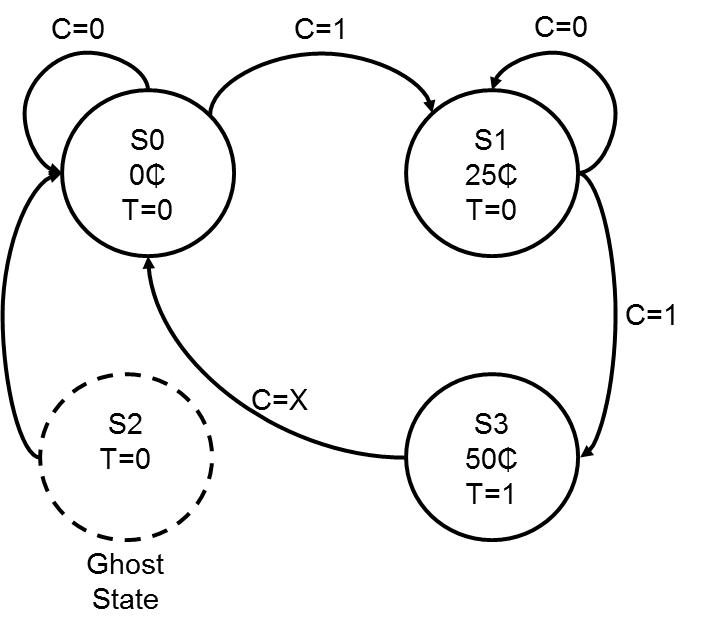
1. Description
2. State Transition Diagram
3. State Transition Table & Output Table

****

1. Next State and Output State Equations
2. Schematic

**FSM Example – Vending Machine:**

**Step 0) Description**

* **Only takes Quarters. Only sells (insert favorite drink).**
* **No change. 50₵ per Soda**
* **Inputs: C = Coin (also, there’s a clock)**
* **Outputs: S = Soda**

**Step 1) State Transition Diagram (Moore Machine)**

**- Transitions only occur with rising edge of clock.**

**- We have 3-4 states How many F/Fs do we need?**

**🡪 2 Lets name them A (MSB) and B (LSB) (i.e. 22)**

**\*\*\*Need to account for all states**

* **The ghost state could be our reset state**
* **When we turn on machine we need to account for all possible states**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| State Table: | | | | |
| Current State | | | Next  State | |
| A | B | C (Coin) | A\* | B\* |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |

**Step 2) State Transition Table & Output Table**

|  |  |  |
| --- | --- | --- |
| Output Table | | |
| Current State | | Output |
| A | B | S |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Step 3) Next State and Output State Equations**

**Step 4) Schematic**

****

**Step 1) State Transition Diagram (Mealy Machine)**

* **Potential for less states**
* **More difficult to design**
* **Outputs tied to arcs**

****

**How many Flip-Flops do we need now? 1**

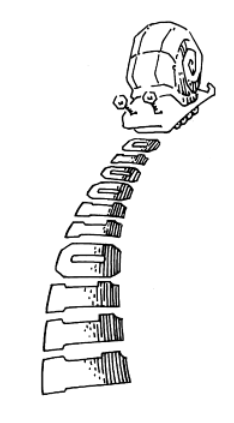
**Step 2) State Transition Table & Output Table**

|  |  |  |  |
| --- | --- | --- | --- |
| State/Output Table: | | | |
| Current State | | Next  State | Output |
| State | C (Coin) | State\* | Tea |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Step 3) Next State and Output State Equations**

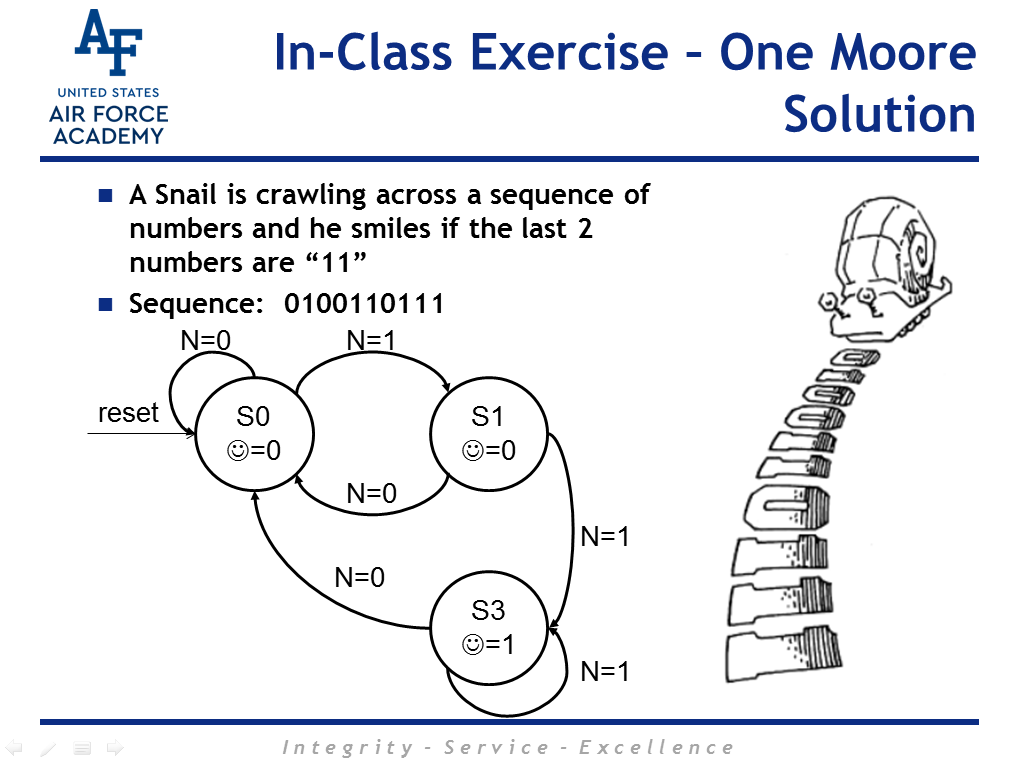
**Step 4) Schematic**

****

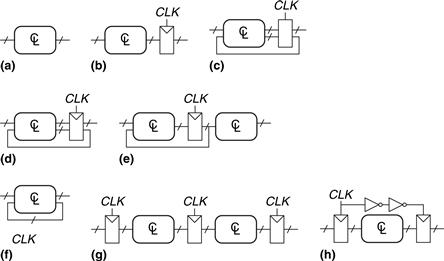
**Boardwork:**

* Draw me a picture of a Mealy or Moore State Transition Diagram that shows
* A Snail is crawling across a sequence of numbers and he smiles if the last 2 numbers are “11”
* Sequence: 0100110111

****



**Example 3.5: Synchronous Sequential Circuits**



**Solutions**

**Circuit:**

* (a) is combinational, not sequential, because it has no registers.
* (b) is a simple sequential circuit with no feedback.
* (c) is neither a combinational circuit nor a synchronous sequential circuit, because it has a latch that is neither a register nor a combinational circuit.
* (d) and (e) are synchronous sequential logic; they are two forms of finite state machines, which are discussed in Section 3.4.
* Same as (d)
* (f) is neither combinational nor synchronous sequential, because it has a cyclic path from the output of the combinational logic back to the input of the same logic but no register in the path.
* (g) is synchronous sequential logic in the form of a pipeline, which we will study in Section 3.6.
* (h) is not, strictly speaking, a synchronous sequential circuit, because the second register receives a different clock signal than the first, delayed by two inverter delays.