# Chapter 10
# Finite State Machines Controller
## SKEE2263 Digital Systems

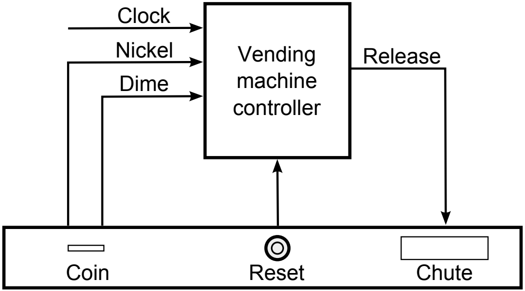
Mun'im/Ismahani/Izam

FEE, Universiti Teknologi Malaysia

April 27, 2016

# Table of Contents

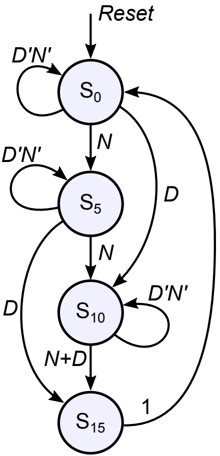
# Vending Machine Abstract View



# Vending Machine Specs

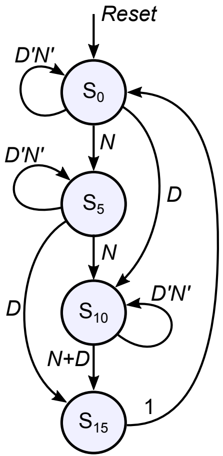
- Release gum after 15¢is received
- Machine has single coin slot
- Accepts 5¢(nickel) and 10¢(dime), one at a time
- The controller's output causes gum to be dispensed

# Vending Machine Symbolic State Diagram




- Each state represents how much money has been deposited
- Variables:
  - N = nickel, 10¢ detected
  - D = dime, 5¢ detected
  - R = release

# Vending Machine Symbolic State Table




| Present State | Inputs D | Inputs N | Next State | Output Release |
|---|---|---|---|---|
| 0¢ | 0 | 0 | 0¢ | 0 |
| | 0 | 1 | 5¢ | 0 |
| | 1 | 0 | 10¢ | 0 |
| | 1 | 1 | × | × |
| 5¢ | 0 | 0 | 5¢ | 0 |
| | 0 | 1 | 10¢ | 0 |
| | 1 | 0 | 15¢ | 0 |
| | 1 | 1 | × | × |
| 10¢ | 0 | 0 | 10¢ | 0 |
| | 0 | 1 | 15¢ | 0 |
| | 1 | 0 | 15¢ | 0 |
| | 1 | 1 | × | × |
| 15¢ | × | × | 15¢ | 1 |

# Vending Machine Encoded State Table

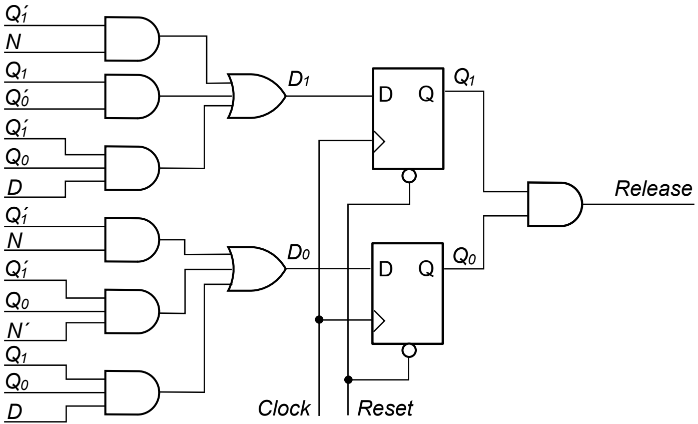
| State Encoding | |
|---|---|
| Symbolic | Encoded |
| 0¢ | 00 |
| 5¢ | 01 |
| 10¢ | 10 |
| 15¢ | 11 |

| Present State | Inputs | | Next State | Output |
|---|---|---|---|---|
| $Q_1 Q_0$ | *D* | *N* | $D_1 D_0$ | *Release* |
| 0 0 | 0 | 0 | 0 0 | 0 |
| | 0 | 1 | 0 1 | 0 |
| | 1 | 0 | 1 0 | 0 |
| | 1 | 1 | × × | × |
| 0 1 | 0 | 0 | 0 1 | 0 |
| | 0 | 1 | 1 0 | 0 |
| | 1 | 0 | 1 1 | 0 |
| | 1 | 1 | × × | × |
| 1 0 | 0 | 0 | 1 0 | 0 |
| | 0 | 1 | 1 1 | 0 |
| | 1 | 0 | 1 1 | 0 |
| | 1 | 1 | × × | × |
| 1 1 | × | × | 0 0 | 1 |

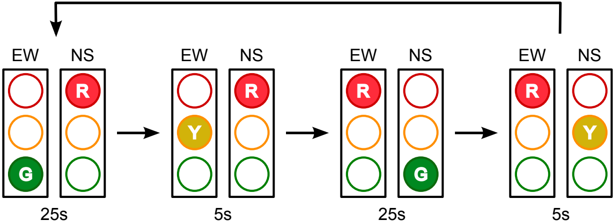
# Vending Machine Equations

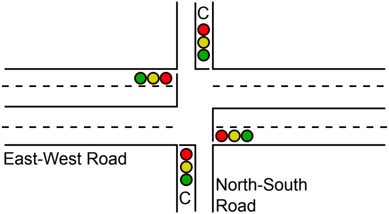
$$D_1 = Q_1'D + Q_1Q_0' + Q_1'Q_0N$$
$$D_0 = Q_1'N + Q_1'Q_0N' + Q_1Q_0D$$
$$Release = Q_1Q_0$$

# Vending Machine Gate Level Schematic



# Traffic Light Sequence

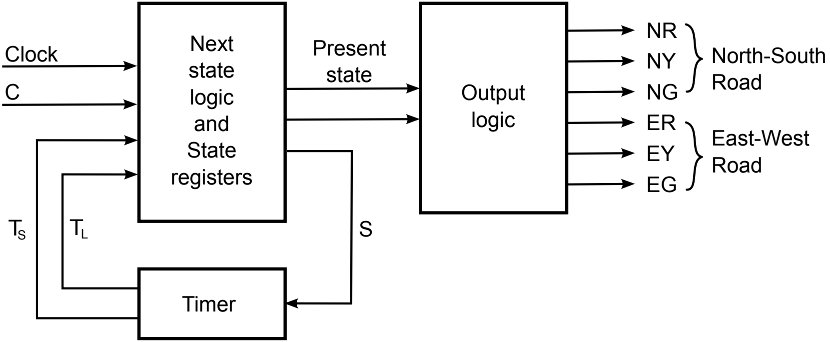


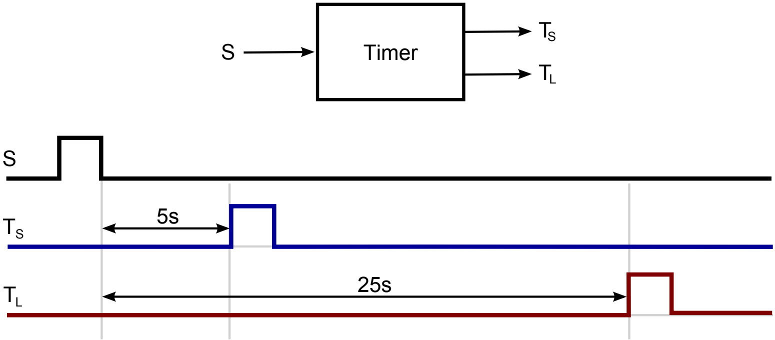
# Traffic Light Sequencing

| State | East-West | North-South | Delay |
|---|---|---|---|
| EG | Green | Red | 25 sec. |
| EY | Yellow | Red | 5 sec. |
| NG | Red | Green | 25 sec. |
| NY | Red | Yellow | 5 sec. |

# TLC Architecture

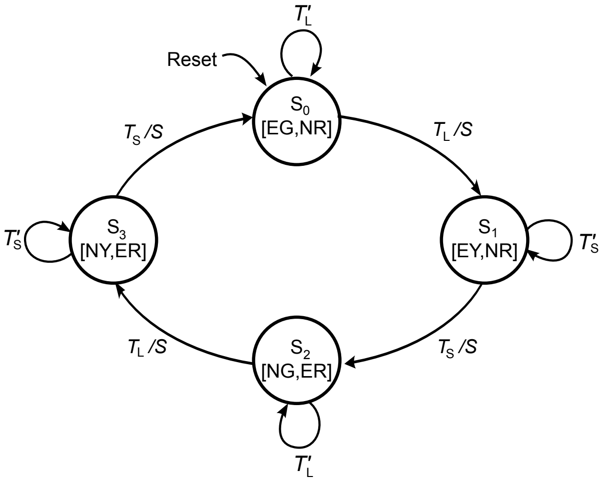


# Timer Module




- This is a separate state machine
- Just use as is. Don't worry about the internals.

# TLC Basic State Diagram

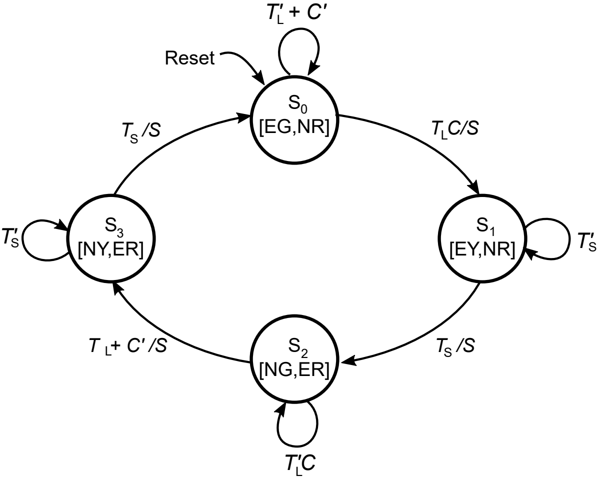



- Constant delay. Car detect (C) input not used.

# TLC Basic State Table

| Present State | Inputs | | Next State | Outputs | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $Q_1Q_0$ | $T_L$ | $T_S$ | $Q_1^+Q_0^+$ | NR | NY | NG | ER | EY | EG | S |
| EG | 0 | X | EG | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| EG | 1 | X | EY | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| EY | X | 0 | EY | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| EY | X | 1 | NG | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| NG | 0 | X | NG | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| NG | 1 | X | NY | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| NY | X | 0 | NY | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| NY | X | 1 | EG | 0 | 1 | 0 | 1 | 0 | 0 | 1 |

# TLC with Car Detection

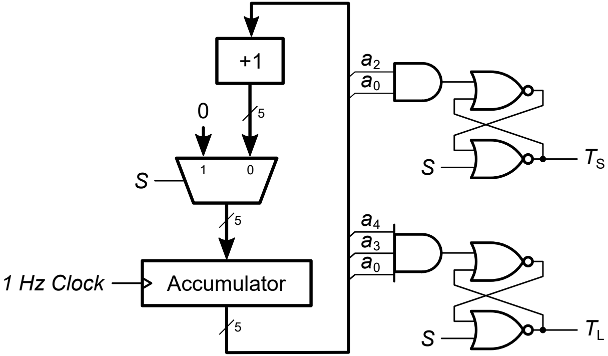



- If C = 0, wait in state EG.

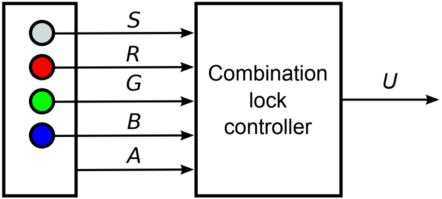
# TLC Car Detection State Table

| Present State | Inputs | | | Next State | Outputs | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_1Q_0$ | $C$ | $T_L$ | $T_S$ | $Q_1^+Q_0^+$ | NR | NY | NG | ER | EY | EG | S |
| EG | 0 | X | X | EG | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| EG | X | 0 | X | EG | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| EG | 1 | 1 | X | EY | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| EY | X | X | 0 | EY | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| EY | X | X | 1 | NG | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| NG | 1 | 0 | X | NG | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| NG | 0 | X | X | NY | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| NG | X | 1 | X | NY | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| NY | X | X | 0 | NY | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| NY | X | X | 1 | EG | 0 | 1 | 0 | 1 | 0 | 0 | 1 |

# TLC Timer Design



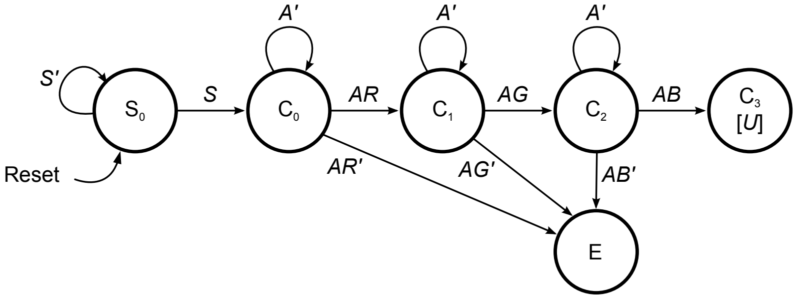
# Combination Lock Top Level




- The controller unlocks the door when buttons are pressed in this sequence:

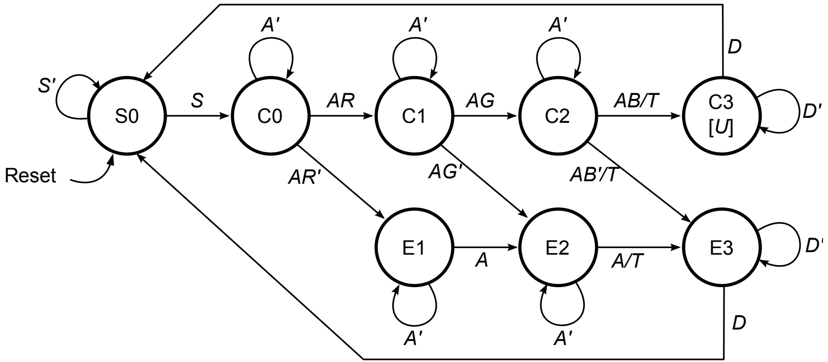
Start → Red → Green → Blue

# Combination Lock 1st Try State Diagram




- It works but lock is easily breakable
- Why?

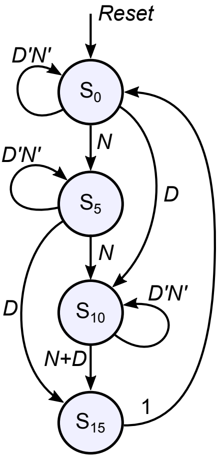
# A More Robust Combination Lock



# One-Hot Encoding

- One state = one flip-flop
  - example: 8 states = 3 FF for straight binary or Gray encoding
  - 8 states = 8 FF for one hot encoding
- Only 1 FF high for any given state, all other low
- The high FF is called the hot FF
- Wastes FF, but on FPGAs FF are plenty
- **Simpler to generate next state logic**

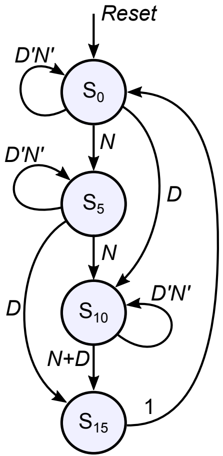
# One-Hot Moore Vending Machine




One-Hot State Assignment

| Symbolic | One-Hot Encoding |
|---|---|
| $S_0$ = 0¢ | 1000 |
| $S_5$ = 5¢ | 0100 |
| $S_{10}$ = 10¢ | 0010 |
| $S_{15}$ = 15¢ | 0001 |

# One-Hot Moore Vending Machine



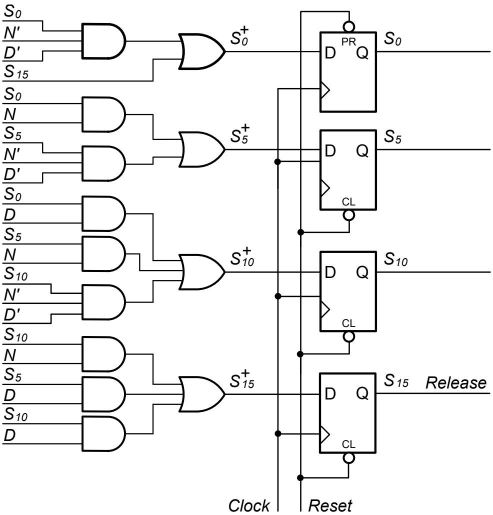

Next state equations:

$$S_0^+ = S_0 D' N' + S_{15}$$
$$S_5^+ = S_0 N + S_5 D' N'$$
$$S_{10}^+ = S_0 D + S_5 N + S_{10} D' N'$$
$$S_{15}^+ = S_5 D + S_{10} N + S_{10} D$$

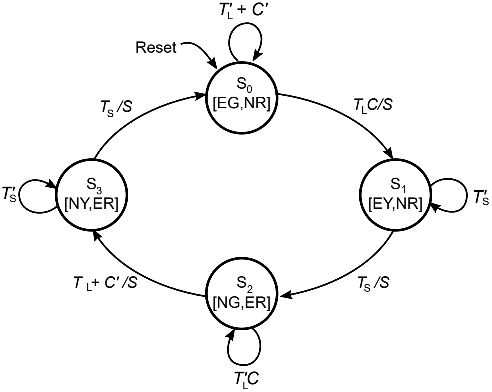
Output equation is:

$$Release = S_{15}$$

# One-Hot Moore Vending Machine



# One-Hot Traffic Light Controller




One-Hot State Assignment

| Symbolic | Encoding |
|---|---|
| $S_0$ = 0¢ | 1000 |
| $S_1$ = 5¢ | 0100 |
| $S_2$ = 10¢ | 0010 |
| $S_3$ = 15¢ | 0001 |

# One-Hot Traffic Light Controller

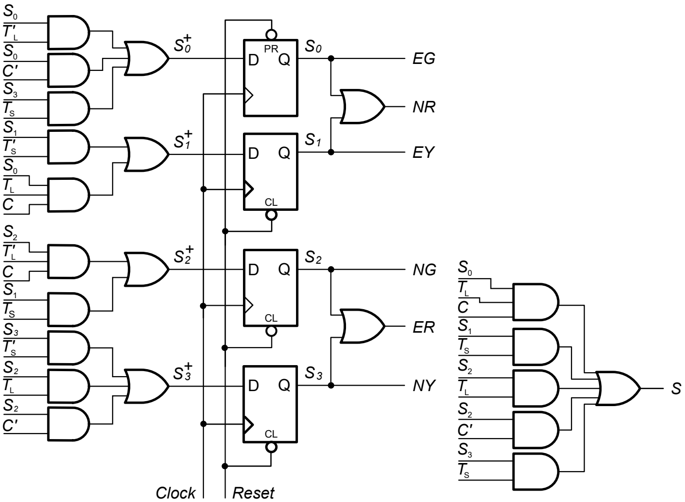
Next state equations:

$$S_0^+ = S_0(T_L' + C') + S_3T_S = S_0T_L' + S_0C' + S_3T_S$$
$$S_1^+ = S_0T_LC + S_1T_S'$$
$$S_2^+ = S_1T_S + S_2T_L'.C$$
$$S_3^+ = S_2(T_L + C') + S_3T_S' = S_2T_L + S_2C' + S_3T_S'$$

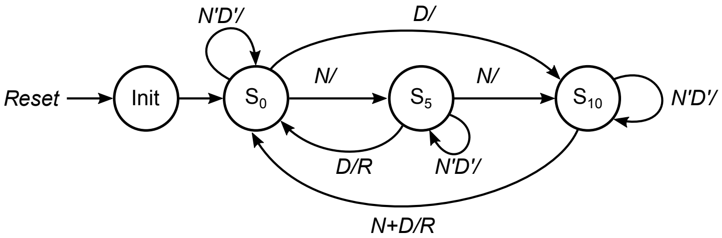
Output equations:

$$S = S_0T_LC + S_1T_S + S_2(T_L + C') + S_3T_S$$
$$EG = S_0$$
$$EY = S_1$$
$$ER = S_2 + S_3$$
$$NG = S_2$$
$$NY = S_3$$
$$NR = S_0 + S_1$$

# One-Hot Traffic Light Controller



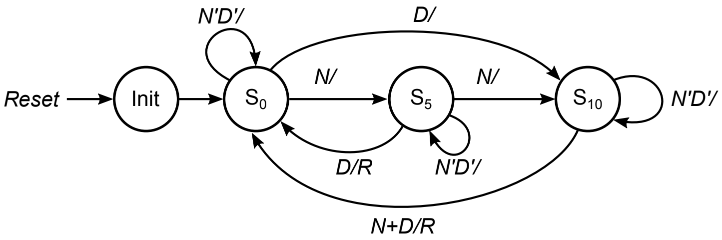
# Almost One-Hot Mealy-type Vending Machine




One-Hot State Assignment

| Symbolic | One-Hot Encoding |
|---|---|
| *Init* = Initialization | 000 |
| $S_0$ = 0¢ | 100 |
| $S_5$ = 5¢ | 010 |
| $S_{10}$ = 10¢ | 001 |

# Almost One-Hot Mealy-type Vending Machine




Next state & output equations:

$$S_0 = Init + S_0 N' D' + S_5 D + S_{10} N + S_{10} D$$
$$S_5 = S_0 N + S_5 N' D'$$
$$S_{10} = S_0 D + S_5 N + S_{10} N' D'$$
$$R = S_5 D + S_{10} D + S_{10} N$$

There is no next state equation for *Init*. Why?

# Almost One-Hot Mealy-type Vending Machine