

# Chapter 13

## Algorithmic State Machines

### SKEE2263 Digital Systems

Mun'im Zabidi {munim@utm.my}  
Ismahani Ismail {ismahani@fke.utm.my}  
Izam Kamisian {e-izam@utm.my}

Faculty of Electrical Engineering, Universiti Teknologi Malaysia

April 30, 2018

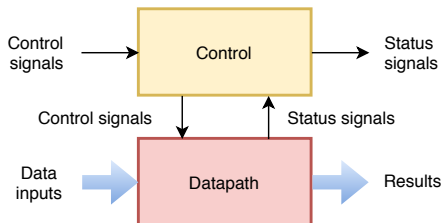
# Table of Contents

- 1 ASM
- 2 Traffic Light Controller
- 3 One-Hot State Assignment

# Large Digital Circuits

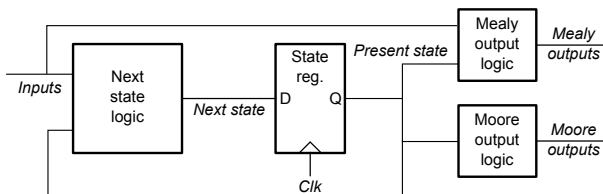
Large digital circuits:

- Consists of data path circuits and control circuits
  - Data path : store, process and move data
  - Control : determine flow of data
- May have Moore and Mealy outputs in the same circuit
- Better described using ASM charts



# Large Digital Circuits

- Having Moore and Mealy outputs in the same circuit makes the system more *efficient*



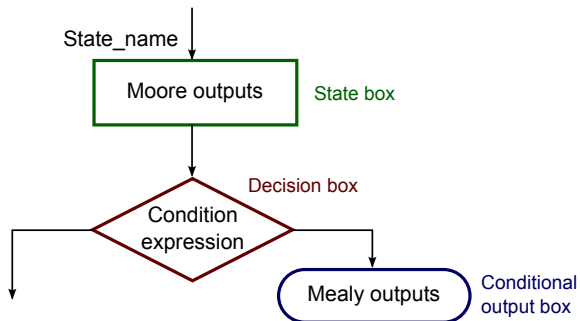
# ASM Charts

## ASM

Algorithmic State Machines is a method for designing large finite state machines.

- Looks like programming flow charts
- Self documenting
- Reduces design errors
- Easy to automate
- More suitable for complex digital systems with a large number of inputs and outputs
- Allows Moore and Mealy outputs in the same circuit

# ASM Elements



# ASM Elements

## State box :

Each state box has only one exit and is usually followed by a decision box. The Moore outputs are listed in this box.

## Decision box :

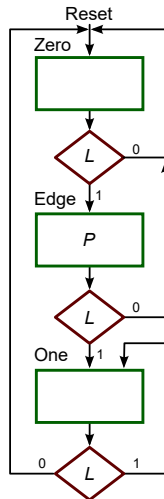
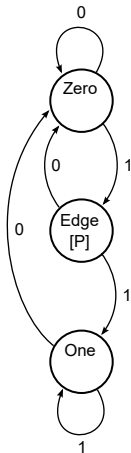
The decision box indicates a condition that is to be tested and the exit path that is to be chosen accordingly.

## Conditional output box :

A conditional output box can only follow decision boxes. Mealy outputs are listed in this box.

# State Diagram $\rightarrow$ ASM Chart

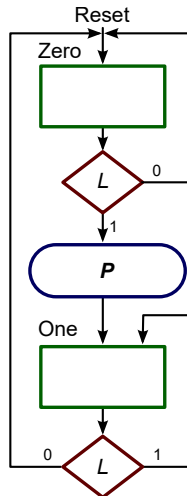
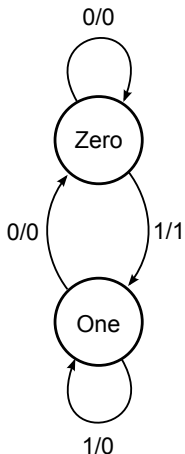
Level-to-Pulse Moore Type





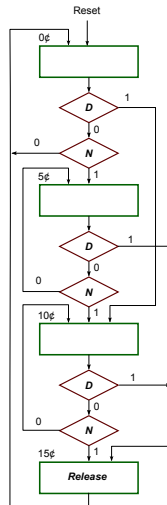
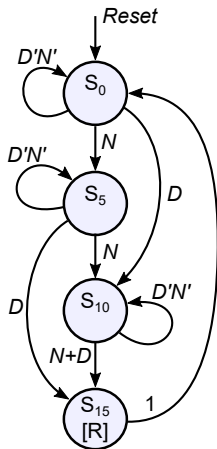
# State Diagram $\rightarrow$ ASM Chart

Level-to-Pulse Mealy Type



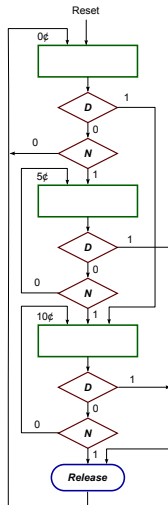
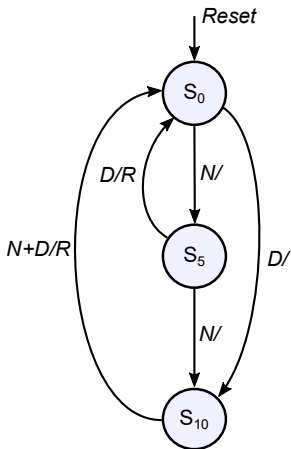
# State Diagram → ASM Chart

## Vending Machine Moore Type



# State Diagram → ASM Chart

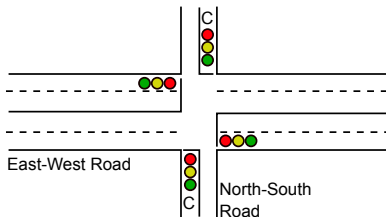
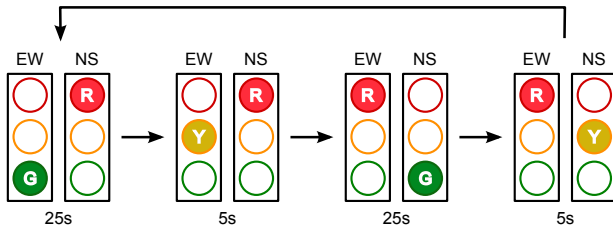
## Vending Machine Mealy Type



## Designing FSM with ASM Chart

- 1 Create an algorithm, using *pseudocode*, to describe the desired operation of the device.
- 2 Convert the pseudocode into an *ASM chart*.
- 3 Design the datapath based on the ASM chart.
- 4 Create a *detailed ASM chart* based on the datapath.
- 5 Design the control logic based on the detailed ASM chart.

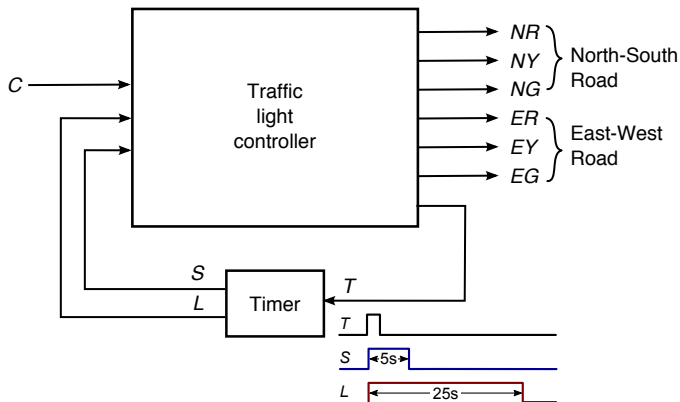
# Traffic Light Sequencing



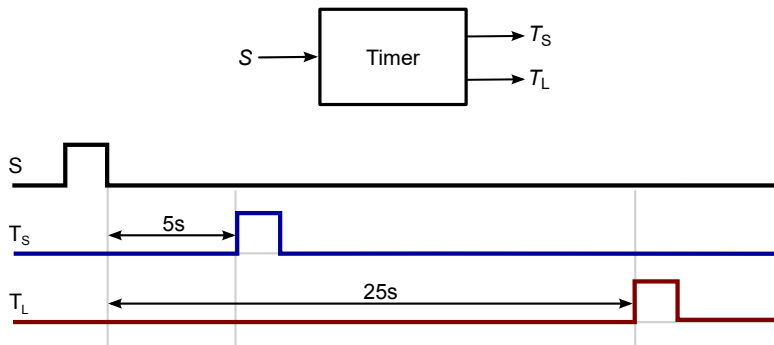
## Traffic Light Sequencing

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



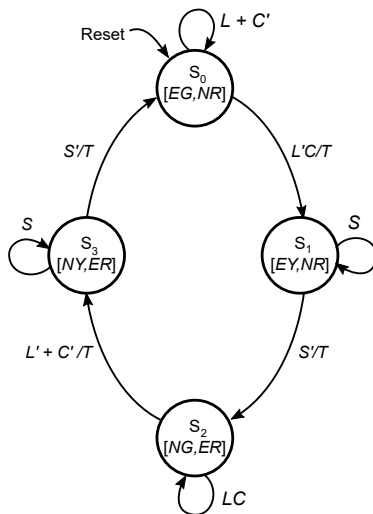
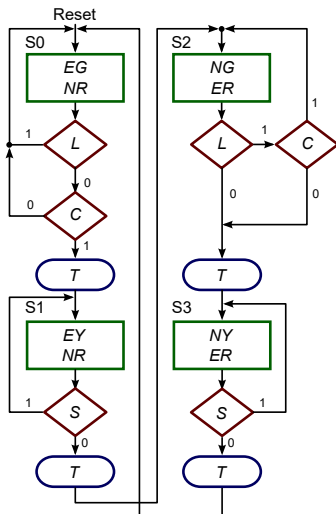
## Timer Module



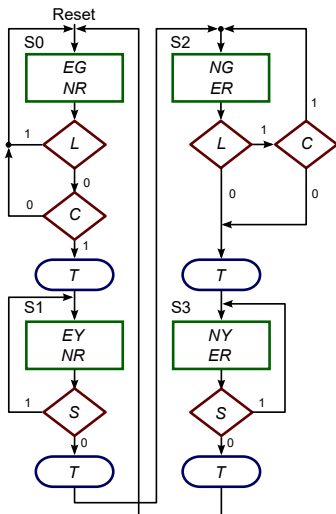
- This is a separate state machine
- Just use as is. Don't worry about the internals.



# TLC with Car Detection



# One-Hot Traffic Light Controller



## One-Hot State Assignment

Symbolic	Encoding
$S_0 = 0\text{¢}$	1000
$S_1 = 5\text{¢}$	0100
$S_2 = 10\text{¢}$	0010
$S_3 = 15\text{¢}$	0001

# One-Hot Traffic Light Controller

Next state equations:

$$S_0^+ = S_0(T'_L + C') + S_3T_S = S_0T'_L + S_0C' + S_3T_S$$

$$S_1^+ = S_0T_L C + S_1T'_S$$

$$S_2^+ = S_1T_S + S_2T'_L.C$$

$$S_3^+ = S_2(T_L + C') + S_3T'_S = S_2T_L + S_2C' + S_3T'_S$$

Output equations:

$$S = S_0T_L C + S_1T_S + S_2(T_L + C') + S_3T_S$$

$$EG = S_0$$

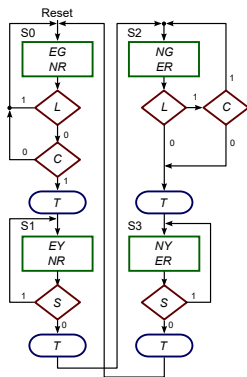
$$EY = S_1$$

$$ER = S_2 + S_3$$

$$NG = S_2$$

$$NY = S_3$$

$$NR = S_0 + S_1$$



# One-Hot Traffic Light Controller

