Logic and Computer Design Fundamentals

State Machine Design

Charles Kime

(Hyperlinks are active in View Show mode)

Overview

- **Part 1 - Storage Elements**
- **Part 2 - Sequential Circuit Analysis**
- **Part 3 - Sequential Circuit Design**
- **Part 4 – State Machine Design**
  - **Issues with traditional state diagrams and table representations**
  - **The state machine diagram model**
  - **Constraint checking**
  - **State machine diagram application and design**

Finite State Machines

- **A finite state machine (FSM) consists of three sets I, O, and S and two functions f and g in which:**
  - **I is a set of input combinations,**
  - **O is a set of output combinations,**
  - **S is a set of states**
  - **f is the next state function f(I, S), and**
  - **g is the output function f(S) [Moore model] or the output function f(I, S) [Mealy model].**
- **The FSM is a fundamental mathematical model used for sequential circuits.**
- **The details of the traditional state diagrams and state tables as we have defined them are just two of many ways of representing FSMs.**

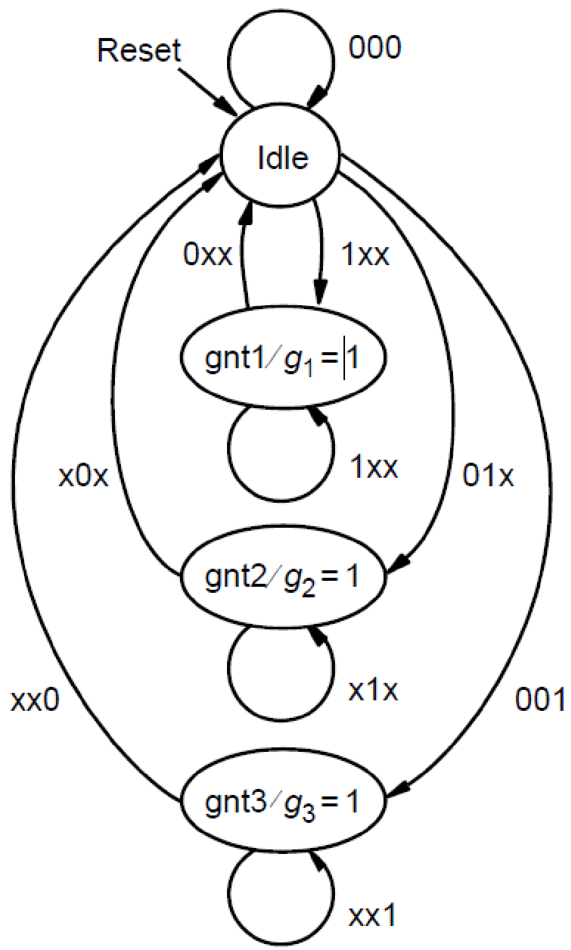
Issues with Traditional State Diagram and Table Representations

- **Both of these traditional representations require:**
  - **Enumeration of all input combinations for each state in defining next states**
  - **Enumeration of all input combinations for each state in defining Mealy outputs**
  - **Enumeration of all applicable output combinations for each state (Moore) and for each input combination-state pair (Mealy).**
- **For state diagrams, all Mealy outputs must be specified on transition arcs**

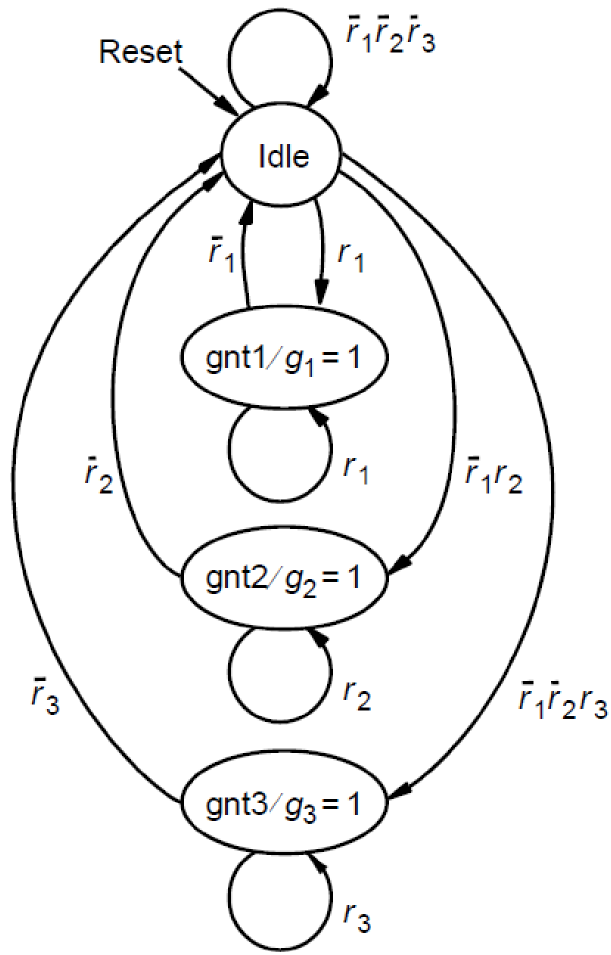
Issues with Traditional State Diagram and Table Representations

- **These requirements may be acceptable for sequential circuits with relatively few inputs, and outputs.**
- **For larger numbers of inputs and outputs both traditional representations become intractable.**
- **The specification of outputs only on transition arcs complicates the specification of outputs for Mealy circuits unnecessarily.**

Issues with Traditional State Diagram and Table Representations



Traditional State Diagram



State Machine Diagram (SMD)

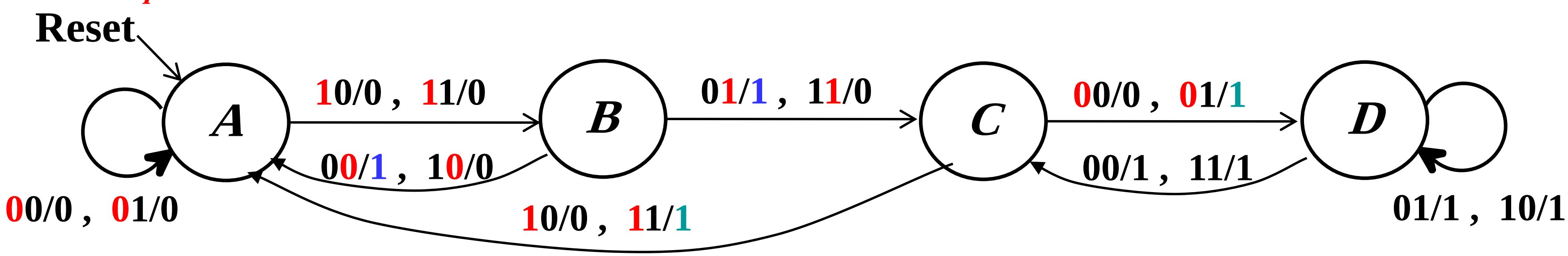
State Machine Diagram Model

- **In response to the issues listed, a broader state machine diagram (SMD) representation has been devised.**
- **Many other authors have used similar representations to overcome some of the issues we have listed.**
- **The SMD achieves the flexibility of the *Algorithmic State Machine* (ASM) (used in some previous editions of this text), without adopting the constraints of the ASM notation.**

Issues with Traditional State Diagram and Table Representations

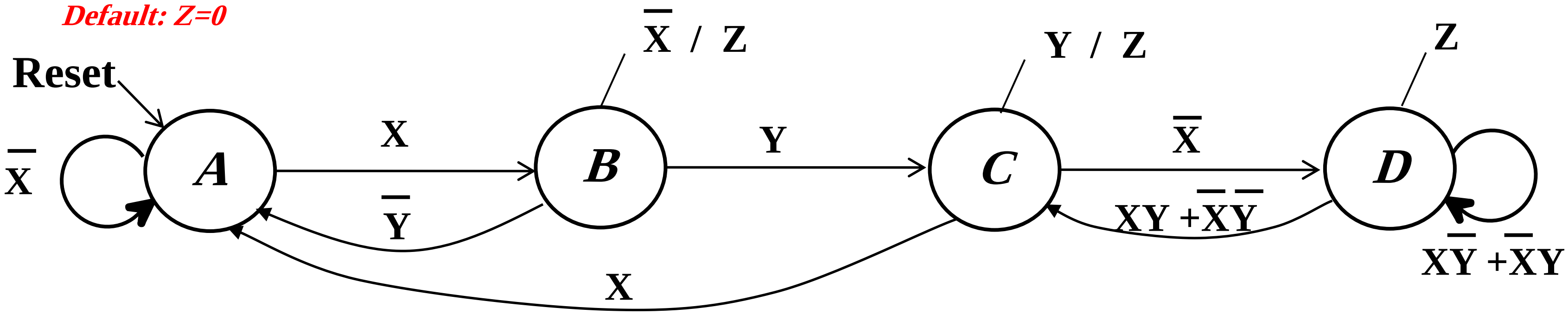
Traditional State Diagram:

Inputs: X, Y
Output: Z



State Machine Diagram (SMD):

Inputs: X, Y
Output: Z
Default: Z=0

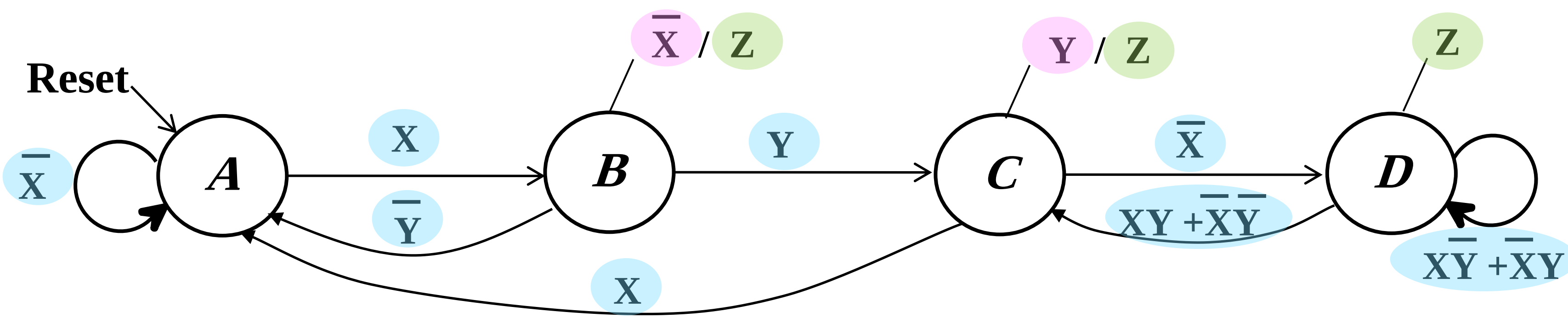


State Machine Diagram

- **Uses state nodes and transition arcs as in the traditional state diagram**
- **Adds notation for defining Mealy outputs on states as well as transitions**
- **Is based on input conditions, transition conditions, output conditions and output actions:**
  - ***Input condition*: a Boolean expression or equation which evaluates to either 0 or 1.**
  - ***Transition condition,* (TC): an input condition on a transition arc which evaluates to either 0 or 1.**
  - ***Output condition* (OC): a input condition that if equal to 1 causes an output action to occur and if 0 does not cause the output to occur.**

State Machine Diagram

Inputs: X, Y
Output: Z
Default: Z=0



Transition Condition

Output Condition ….

Output Action ………

State Machine Diagram

- **Output Action Examples**
  - **Single Variables:**
    - **Appearance of variable Z attached a state specifies that Z = 1. Z is implicitly 0 otherwise.**
    - **Appearance of variable Z attached to a transition condition (and possibly an output condition) from a state implies that Z = 1 for the condition(s) satisfied. Z is implicitly 0 otherwise unless Z is a Moore output (unconditional) attached to the state or is part of a TCI label attached to a state.**
    - **Separate default value statements may be used to explicitly specify by default Z = 0 or Z = 1.**

State Machine Diagram

- **Output Action Examples**
  - **Vector Variables:**
    - Appearance of an equation Z = vector value attached to a state specifies the value of Z for the state.
    - Appearance of an equation Z = vector value attached to a transition condition (and possibly an output condition) from a state specifies the value of Z for the state, transition condition and output condition. The value of Z attached to a transition may also be specified by a Moore output (unconditional) attached to the state or as part of a TCI label attached to a state. Otherwise, Z takes on a default value if one is specified . The default value for a vector must be specified (including possibly don't cares).
  - **Register Transfer Outputs**
    - Useful for describing controlled datapath operations (see Chapter 7)

State Machine Diagram Transition Conditions

- A **unconditional transition** has no transition condition (TC) on its arc or a transition condition consisting of the constant 1.

- A **conditional transition** has one or more transition conditions on its arc. If any one of the conditions evaluates to 1, the transition occurs.

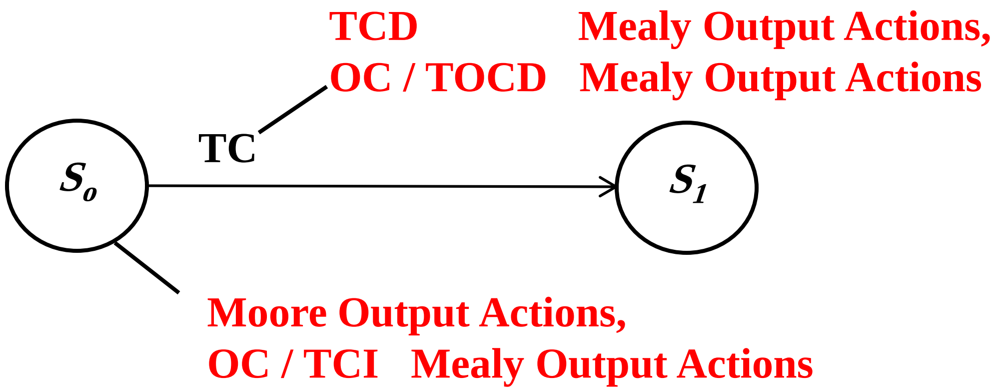
State Machine Diagram Output Actions

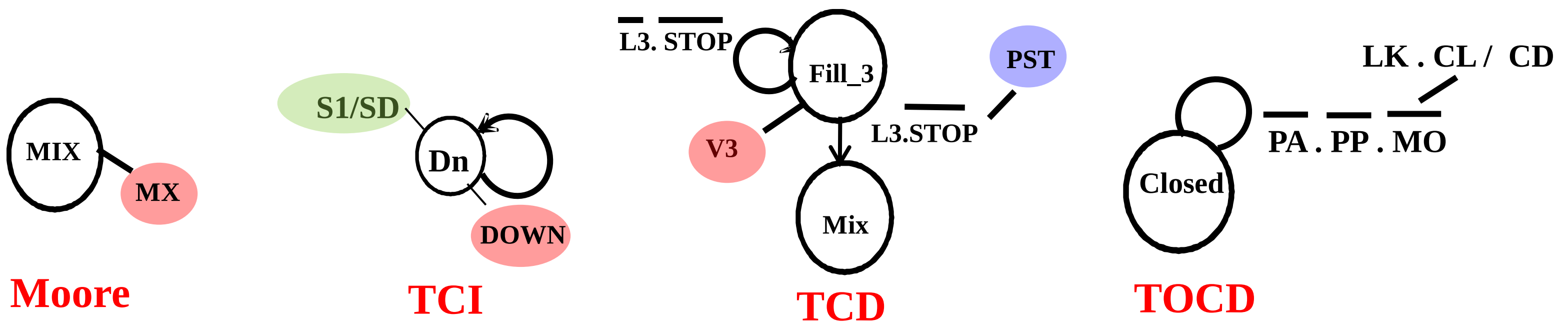
- ***Moore output actions***, are unconditional, depending only on the state, and are attached by a line to the respective state.
- ***Transition condition-independent* (TCI) *Mealy output actions*** are preceded by their output condition and a slash and are attached by a line to the respective state. The output action occurs if the output condition evaluates to 1.
- ***Transition condition-dependent* (TCD) Mealy output actions are** attached by a line to their respective transition condition. The output action occurs if the transition condition evaluates to 1.
- ***Transition and output condition-dependent* (TOCD) Mealy** output actions are preceded by an output condition and a slash and are attached by a line to their respective transition condition. The output action occurs if the transition condition and the output condition both evaluate to 1.

State Machine Diagram Output Actions

- To summarize, in a given state, an output action occurs if it is:
  - (a) unconditional (Moore)
  - (b) TCI and its output condition OC evaluates to 1
  - (c) TCD and its transition condition TC evaluates to 1
  - (d) TOCD and its transition condition TC and output condition OC both evaluate to 1.
- Moore and TCI output actions attached to a state, apply to *all* transitions from the state.

State Machine Diagram Output Actions





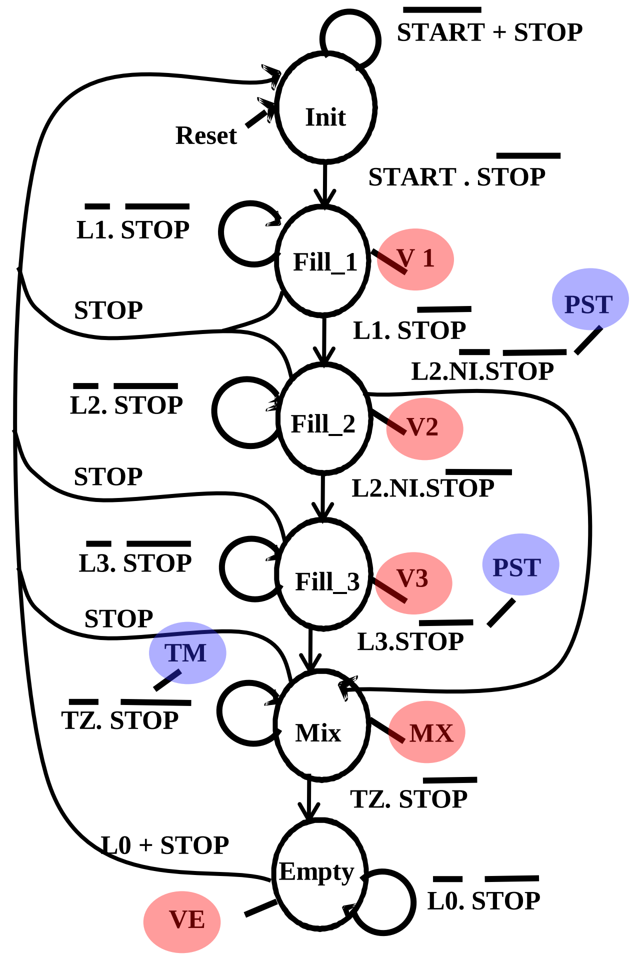
State Machine Diagram
Output Actions: Examples

Moore Output Actions

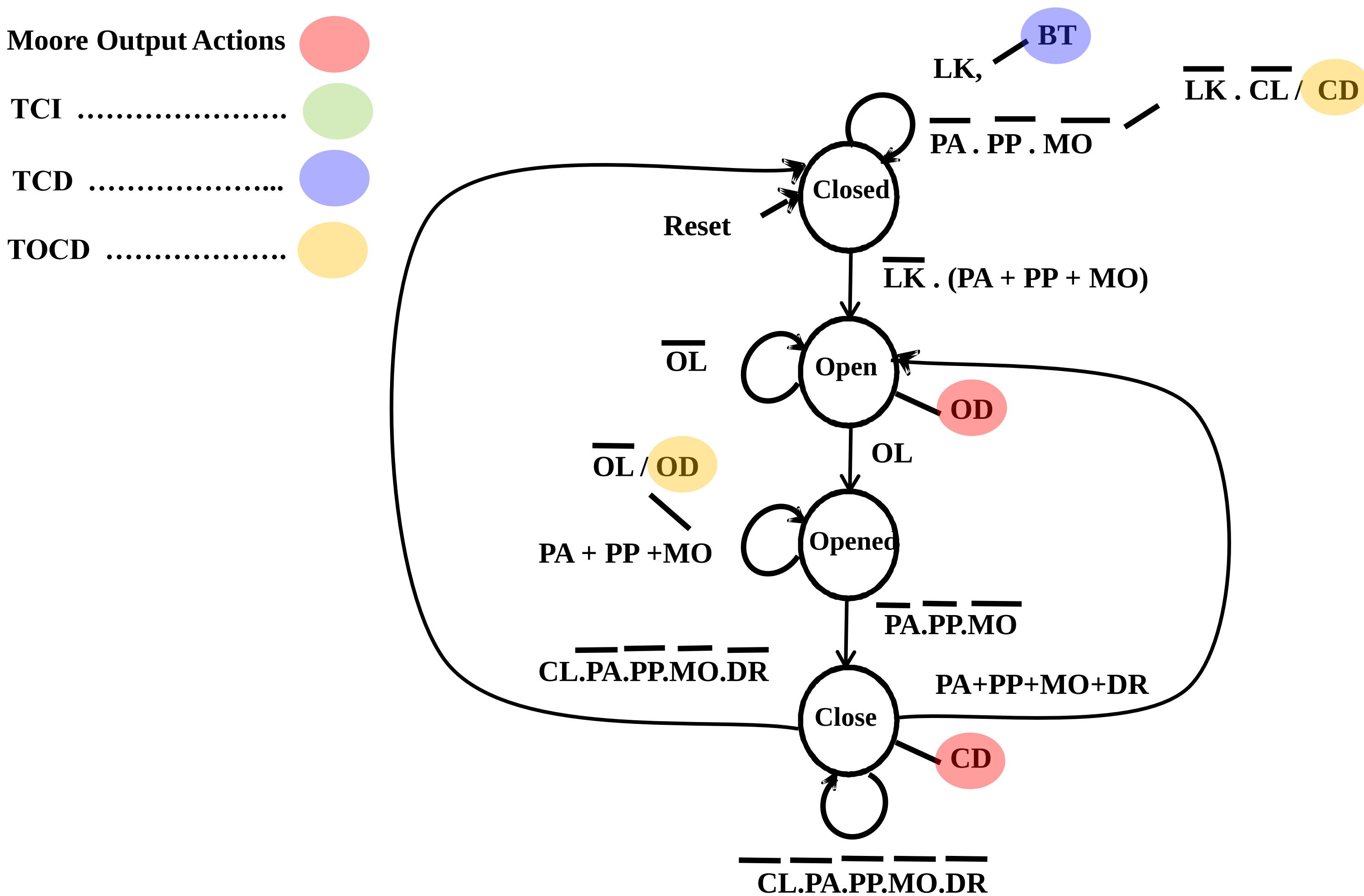
TCI ....

TCD ....

TOCD ....



State Machine Diagram
Output Actions: Examples



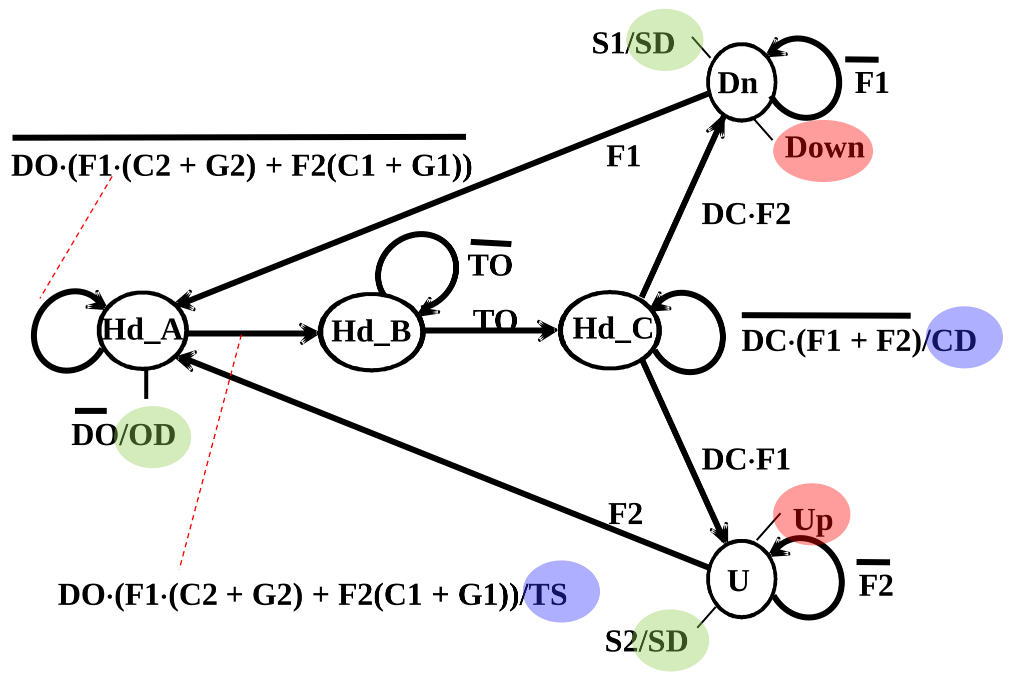
State Machine Diagram
Output Actions: Examples

Moore Output Actions

TCI ………………….

TCD ………………...

TOCD ……………….



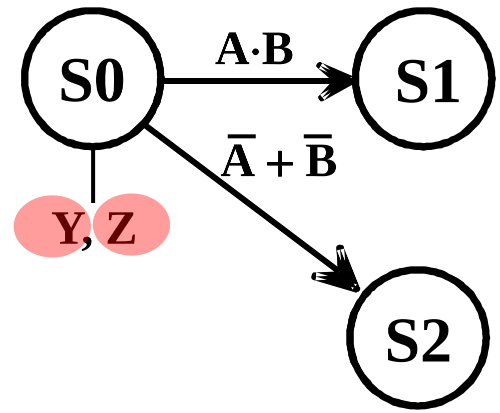
State Machine Diagram

- **This may seem complex, but note the following:**
  - **Only the unconditional output type applies to pure Moore machines**
  - **TCD outputs represents the traditional Mealy model and can be used exclusively at some potential cost in complexity including an increase in the number of states.**
  - **Mixing of Moore and Mealy types and the TCI and TOCD types provide optional opportunities to simplify the state diagram and state table and their specifications**

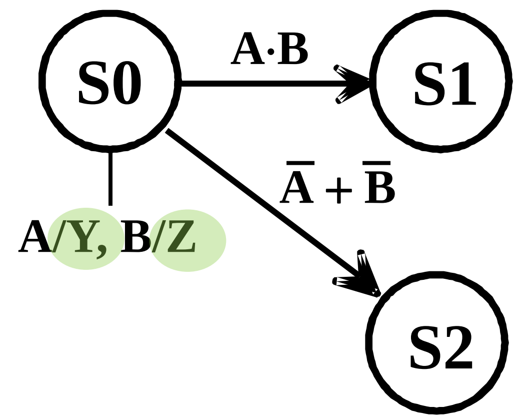
Examples Of Transition & Output Conditions

- **Input Variables A, B, C**
- **Output Variables Y, Z**
  Default: Y = 0, Z = 0

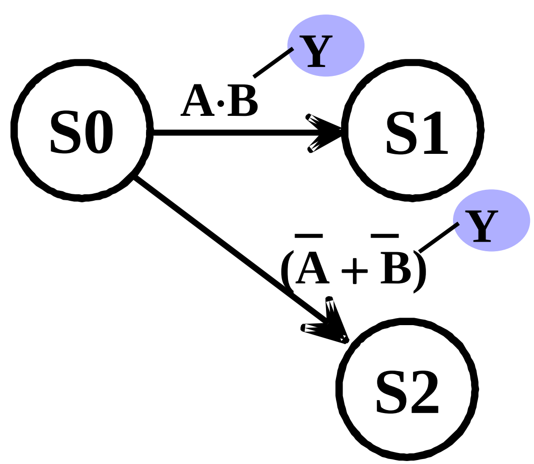
Moore Output Actions
TCI ....
TCD ....
TOCD ....



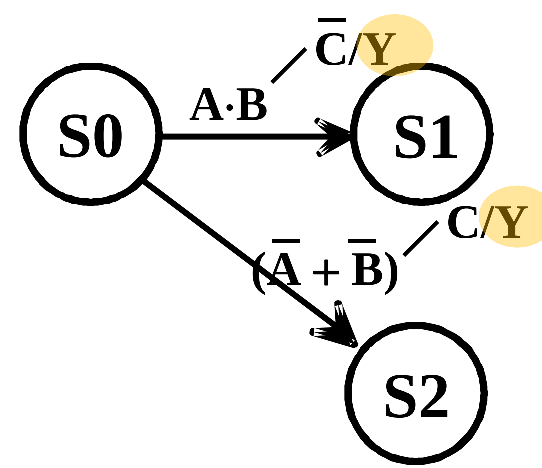
Ex. 1: Moore Outputs



Ex. 2: TCI Outputs



Ex. 3: TCD Outputs



Ex. 4: TOCD Outputs

Constraint Checking

- **TC Constraints**
  - **Constraint 1: In state S_i, for all possible TC pairs (T_{ij}, T_{ik}) on arcs to distinct next states from S_i,**

$$T_{ij} \cdot T_{ik} = 0$$

  - **Constraint 2: In state Si, for all possible TCs, T_{ij}**

$$\Sigma\, T_{ij} = 1$$

- **OC Constraints**
  - **Constraint 1: For every output action in state S_i or on its transitions having coincident output variables with differing values, the corresponding pair of output condition (O_{ij}, O_{ik}) must be mutually exclusive, i. e., satisfy**

$$O_{ij} \cdot O_{ik} = 0$$

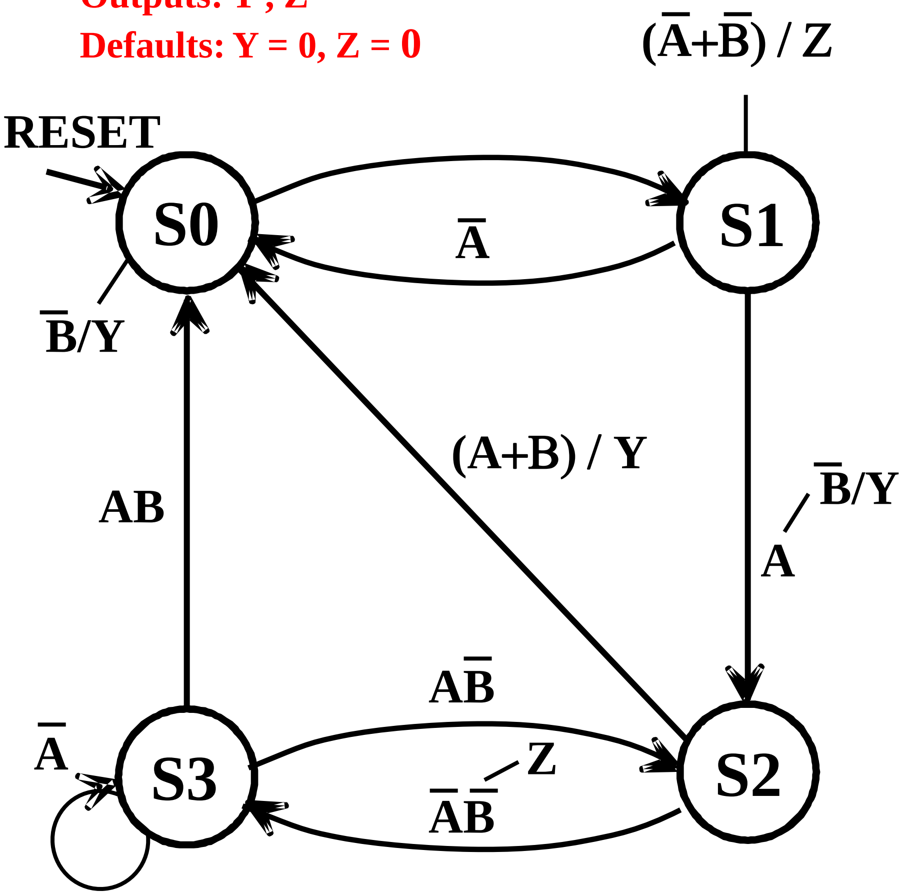
  - **Constraint 2:For every output variable, the output conditions for state S_i or its transitions must cover all possible combinations of input variables that can occur, i. e.,**

$$\Sigma\, O_{ij} = 1$$

  - **For both output constraints above, TCs must be used in evaluating O_{ij} for output actions of TCD and TOCD output action types**
  - **See text for using don't cares and defaults.**

Constraint Checking Example 1:

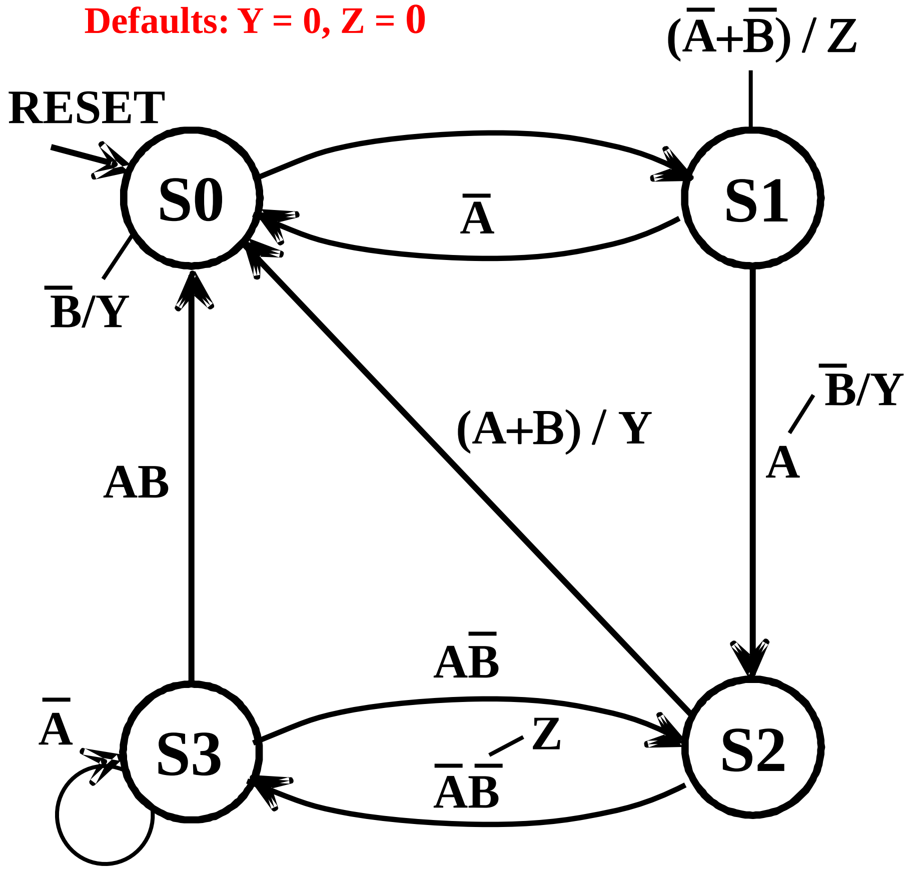
Inputs: A , B
Outputs: Y , Z
Defaults: Y = 0, Z = 0



- **Transition Constraints:**
  - S0: One unconditional TC
  - S1: $A \cdot \bar{A} = 0;$ $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$
    $A + \bar{A} = 1$
  - S2: $(A+B) \cdot (\bar{A}\,\bar{B}) = 0;$ $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$
    $(A+B) + (\bar{A}\,\bar{B}) = 1$
  - S3: $\bar{A} \cdot AB = 0;$ $\begin{bmatrix} 3 \\ 2 \end{bmatrix}$
    $\bar{A} \cdot A\bar{B} = 0;$
    $AB \cdot A\bar{B} = 0;$
    $\bar{A} + AB + A\bar{B} = 1$
- **Output Constraints:**
  - Satisfied for all four states by the given output conditions and values and the default constraints.

Constraint Checking Example 1:

Inputs: A , B
Outputs: Y , Z
Defaults: Y = 0, Z = 0



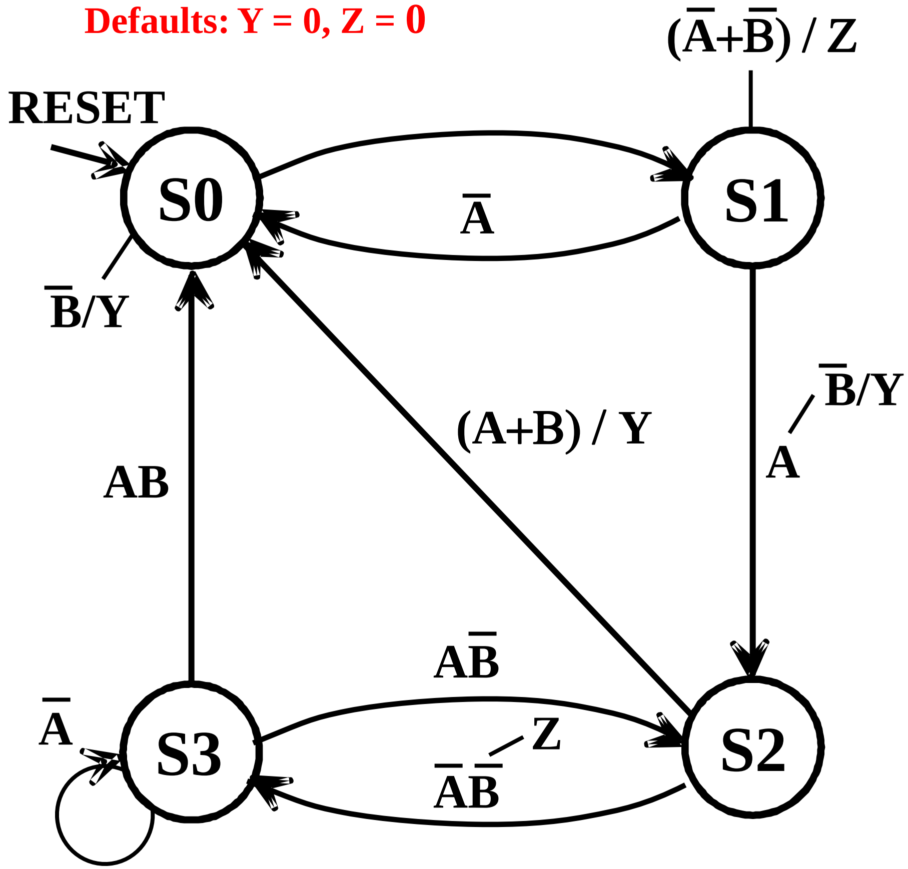
- **Output Constraints:**

Constraint 1:

- **S0: For $\overline{B}$ → Y = 1**
  - **By default, for B, Y = 0**
  - **$\overline{B}$. B = 0**
- **S1: For A$\overline{B}$ → Y=1**
  - **By default, for $\overline{A}$+B, Y = 0**
  - **A$\overline{B}$. ($\overline{A}$+B) = 0**
  - **For $\overline{A}$+$\overline{B}$ → Z=1**
  - **By default, for AB, Z = 0**
  - **($\overline{A}$ + $\overline{B}$) . AB = 0**
- **S2: For A+B → Y=1**
  - **By default, for $\overline{A}\overline{B}$, Y = 0**
  - **(A + B) . $\overline{A}\overline{B}$ = 0**
  - **For $\overline{AB}$ → Z=1**
  - **By default, for A+B, Z = 0**
  - **($\overline{AB}$) . (A+B) = 0**
- **S3: None**

Constraint Checking Example 1:

Inputs: A , B
Outputs: Y , Z
Defaults: Y = 0, Z = 0



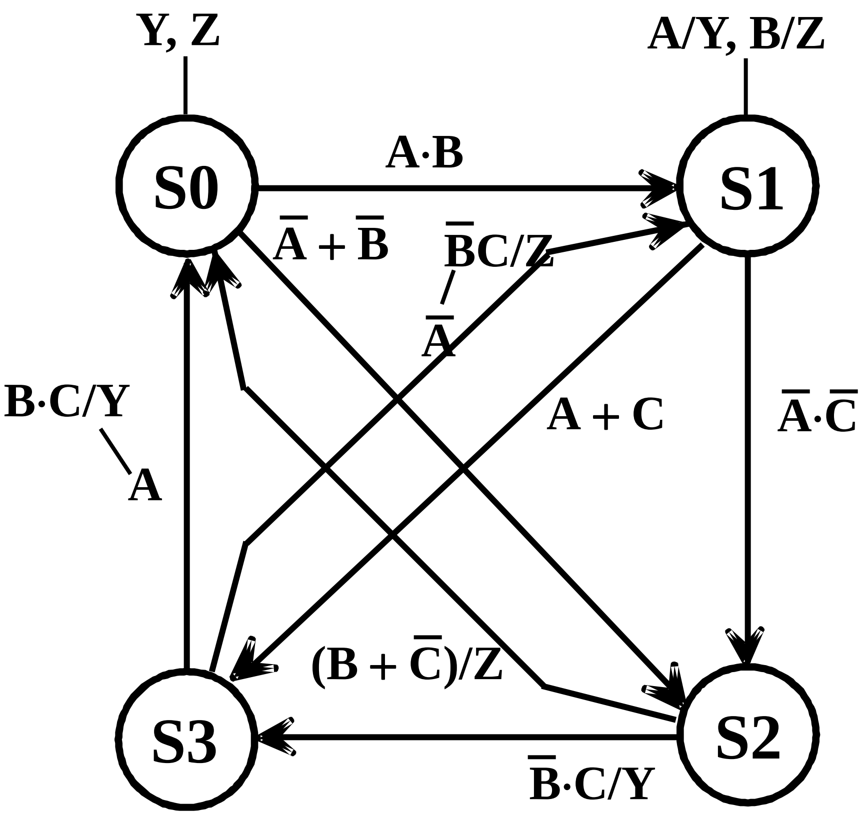
- **Output Constraints:**

  Constraint 2:

  - **S0: For $\bar{B}$ → Y = 1**
    - **By default, for B, Y = 0**
    - **$\bar{B}$ + B = 1**
  - **S1: For $A\bar{B}$ → Y=1**
    - **By default, for $\bar{A}$+B, Y = 0**
    - **$A\bar{B}$ + ($\bar{A}$+B) = 1**
    - **For $\bar{A}+\bar{B}$ → Z=1**
    - **By default, for AB, Z = 0**
    - **($\bar{A}$ + $\bar{B}$) + AB = 1**
  - **S2: For A+B → Y=1**
    - **By default, for $\bar{A}\bar{B}$, Y = 0**
    - **(A + B) + $\bar{A}\bar{B}$ = 1**
    - **For $\overline{AB}$ → Z=1**
    - **By default, for A+B, Z = 0**
    - **($\bar{A}\bar{B}$) + (A+B) = 1**
  - **S3: None**

Constraint Checking Example 2:

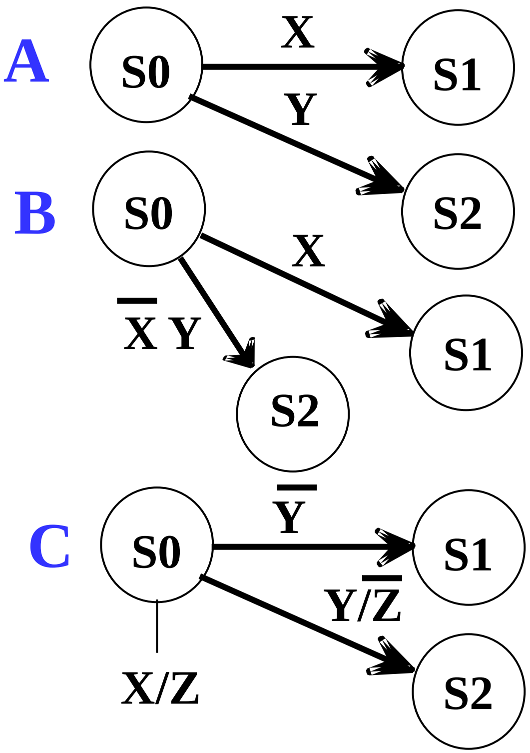
Inputs: A , B
Outputs: Y , Z
Defaults: Y = 0, Z = 0



- **Transition Constraints:**
  - **S0:** $A \cdot B \cdot (\bar{A} + \bar{B}) = 0;$ $\begin{bmatrix}2\\2\end{bmatrix}$
    $A \cdot B + (\bar{A} + \bar{B}) = 1$
  - **S1:** $\bar{A} \cdot \bar{C} \cdot (A + C) = 0;$ $\begin{bmatrix}2\\2\end{bmatrix}$
    $\bar{A} \cdot \bar{C} + (A + C) = 1$
  - **S2:** $\bar{B} \cdot C \cdot (B + \bar{C}) = 0;$ $\begin{bmatrix}2\\2\end{bmatrix}$
    $\bar{B} \cdot C + (B + \bar{C}) = 1$
  - **S3:** $A \cdot \bar{A} = 0;$ $\begin{bmatrix}2\\2\end{bmatrix}$
    $A + \bar{A} = 1$
- **Output Constraints:**
  - **Satisfied for all four states by the given output conditions and values and the default constraints.**

Constraint Violation Examples

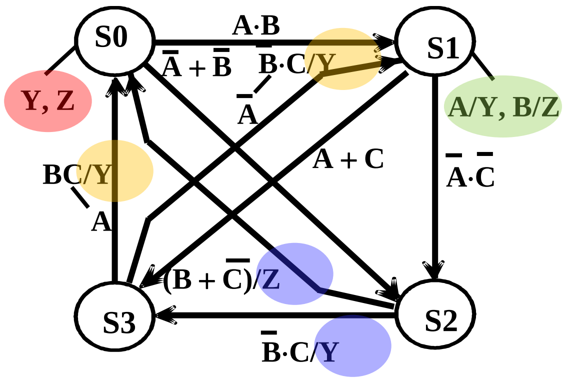
- **Transition Constraints**
  - **Example A:** $X \cdot Y \neq 0$ and $X + Y \neq 1$, so **two constraints are violated**
  - **Example B:** $X \cdot \overline{X}Y = 0$, but $X + \overline{X}Y \neq 1$. so **constraint 2 is violated**
- **Output Constraints**
  - **Example C:** For values Z = 1 and Z = 0, $X \cdot Y \neq 0$, so **constraint 1 is violated**
  - Constraint $X + Y + \overline{Y} = 1$, due to the default value of Z on $\overline{Y}$, so **constraint 2 is satisfied**
  - **Example D:** In general, for a given state, since the output condition for a Moore type output action is 1, no output action on a same output variable with a different value is permitted on the transitions.



State Machine Table Format

State	State Code	Transition Condition	Next State	Next State Code	Output Actions (and OCs)
State Name 1	State Code 1	Unused	Unconditional Next State 1	Next State Code 1	Moore or TCI Output (and OC)
		Transition Cond. 11	Next State 11	Next State Code 11	TCD or TOCD Output 11 (and OC)
		Additional Transition Conditions and Entries for State Name 1			
State Name i	Entries for State Names i, i = 2, …n				

State Machine Table for Constraint Checking Example



Moore Output Actions

TCI ....

TCD ....

TOCD ....

State	State Code Y_1Y_0	Transition Condition	Next State	Next State Code Y_1Y_0	Output Actions (OCs)
S0	00				Y , Z
		A·B	S1	01	
		$\overline{A} + \overline{B}$	S2	10	
S1	01				A/Y , B/Z
		$\overline{A}\cdot\overline{C}$	S2	10	
		A + C	S3	11	
S2	10				
		$\overline{B}$·C	S3	11	Y*
		B + $\overline{C}$	S0	00	Z*
S3	11				
		A	S0	00	$\overline{B}$·C / Y*
		$\overline{A}$	S1	01	B·C / Y*

** is reminder of an output action dependent on transition condition*

State Machine Design Procedure

- Define the input and output variables for the circuit or system and meaning of 0 and 1 values of each variable
- Draw the state machine diagram or formulate the state machine table for the circuit or system
- If a state machine diagram is used, convert it to a state machine table
- From the state machine table, derive optimized next state equations and output equations for the circuit or system

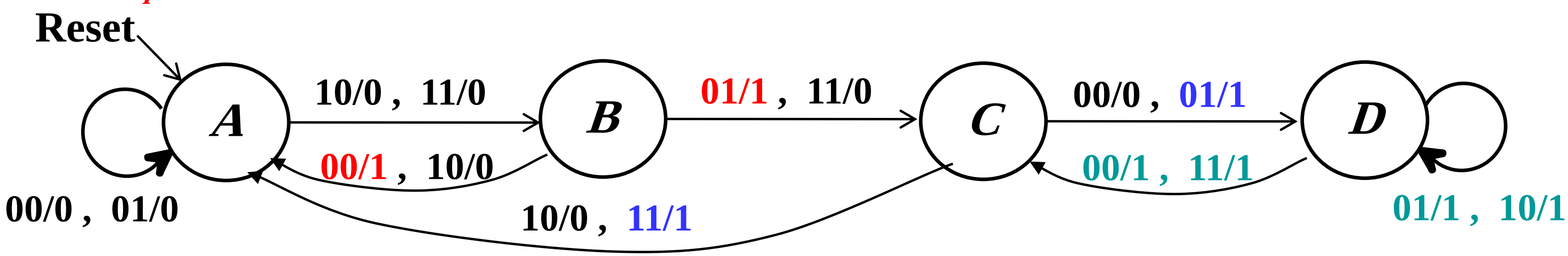
Example 1

Conversion of Traditional State Diagram to State Machine Diagram

Example 1: Conversion

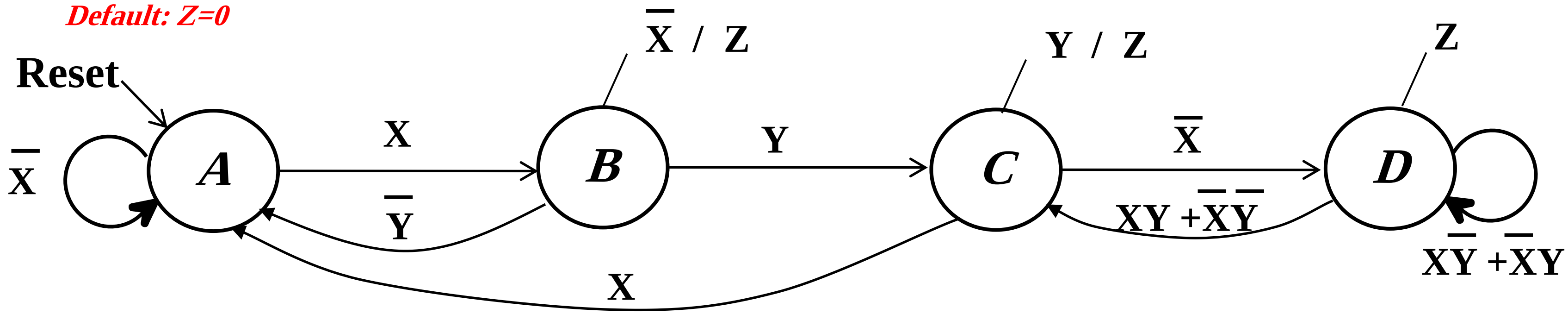
Traditional State Diagram:

Inputs: X, Y
Output: Z



State Machine Diagram:

Inputs: X, Y
Output: Z
Default: Z=0



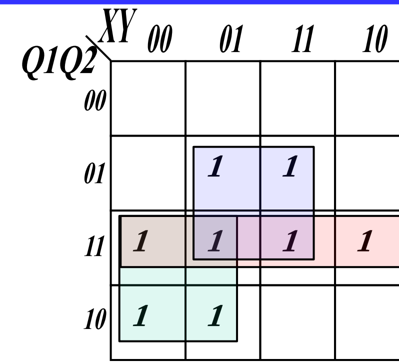
Example 1: Conversion State Machine Table (SMT)

State	State Code	Transition Condition	Next State	Next State Code	Output Actions (OCs)
	$Q_1 Q_2$			$Q_1 Q_2$	
A	00	$\overline{X}$	A	00	
	$Q_1'Q_2'$	X	B	01	
B	01				$\overline{X}$ / Z
		$\overline{Y}$	A	00	
	$Q_1'Q_2$	Y	C	10	
C	10				Y / Z
		$\overline{X}$	D	11	
	Q_1Q_2'	X	A	00	
D	11				Z
		$X\overline{Y} + \overline{X}Y$	D	11	
	Q_1Q_2	$XY + \overline{X}\,\overline{Y}$	C	10	

Example 1: Conversion Equations

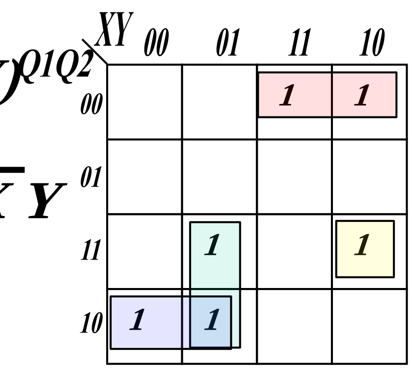
- **Flip-Flop Inputs:**
  - $Q_1(t+1) = \overline{Q_1}\, Q_2\, Y + Q_1\, \overline{Q_2}\, \overline{X} + Q_1\, Q_2$

    → $Q_1(t+1) = Q_2\, Y + Q_1\, \overline{X} + Q_1\, Q_2$



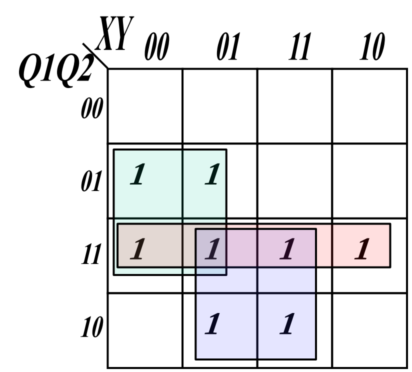
  - $Q_2(t+1) = \overline{Q_1}\, \overline{Q_2}\, X + Q_1\, \overline{Q_2}\, \overline{X} + Q_1\, Q_2\, (X\, \overline{Y} + \overline{X}\, Y)$

    → $Q_2(t+1) = \overline{Q_1}\, \overline{Q_2}\, X + Q_1\, \overline{Q_2}\, \overline{X} + Q_1\, \overline{X}\, Y + Q_1\, Q_2\, \overline{X}\, Y$

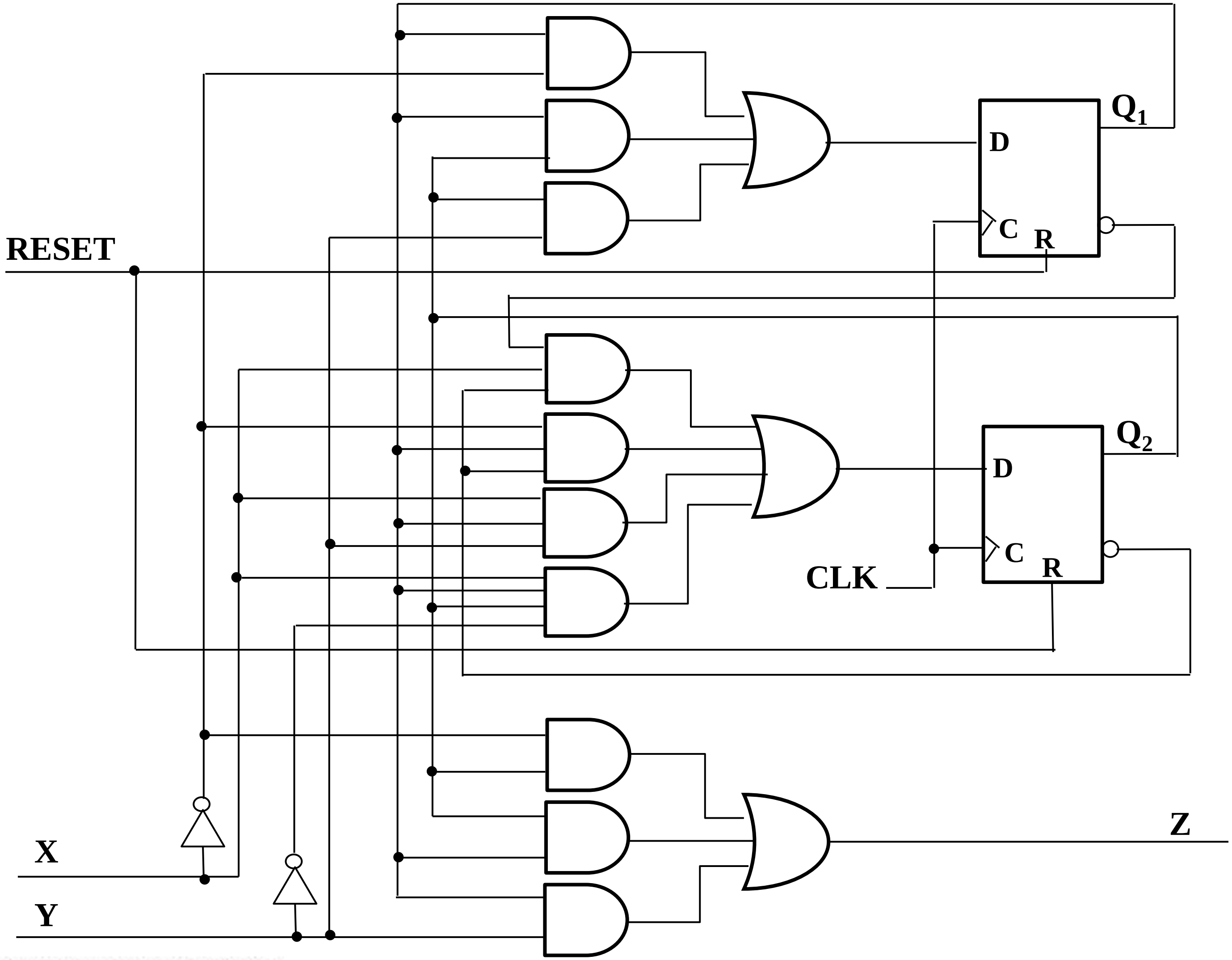


- **Outputs:**
  - $Z = \overline{Q_1}\, Q_2\, \overline{X} + Q_1\, \overline{Q_2}\, Y + Q_1\, Q_2$

    → $Z = Q_2\, \overline{X} + Q_1\, Y + Q_1\, Q_2$



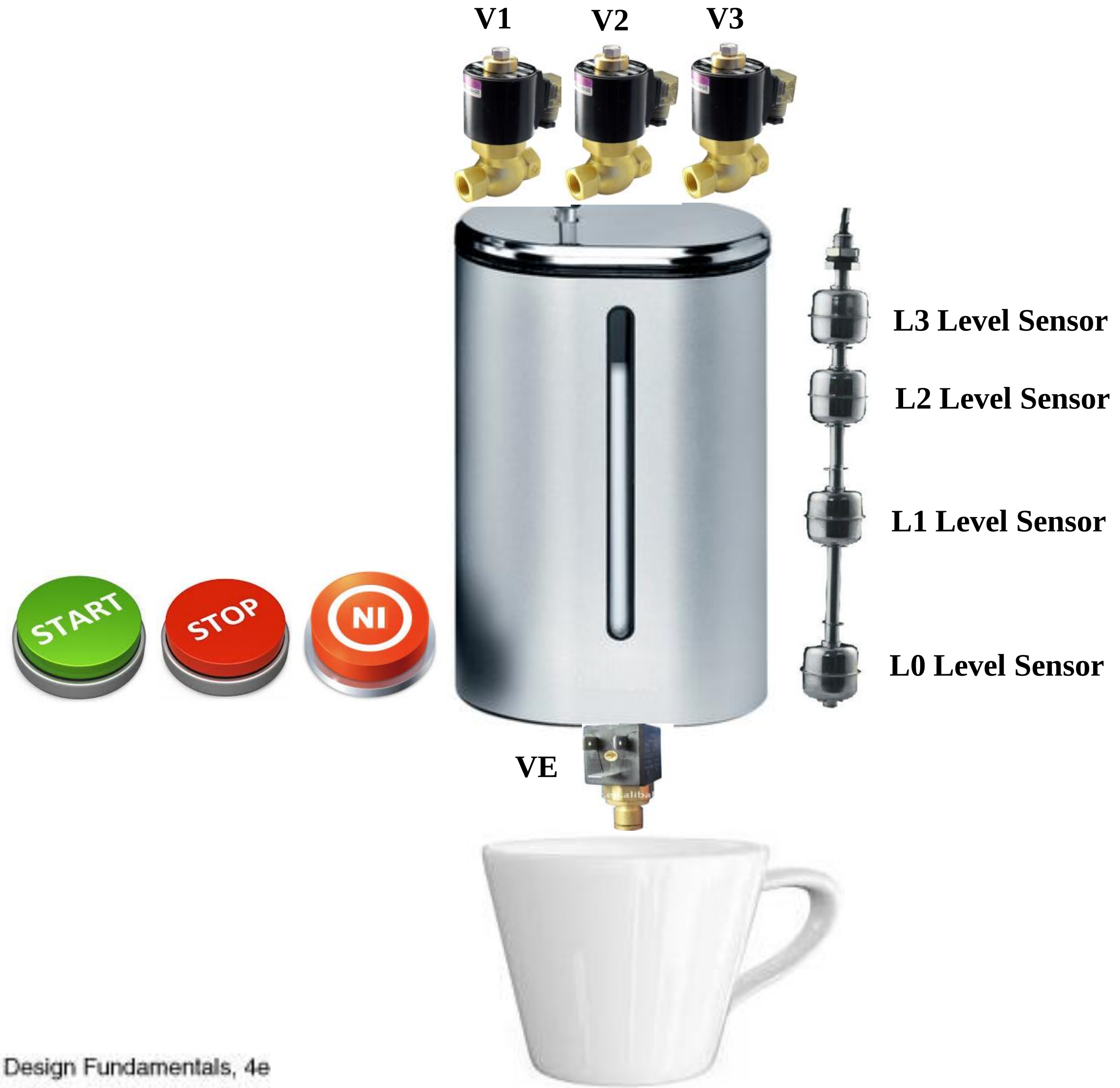
Example 1: Conversion



Example 2

Batch Mixing System

Example 2: Batch Mixing System



Example 2: Batch Mixing System Inputs

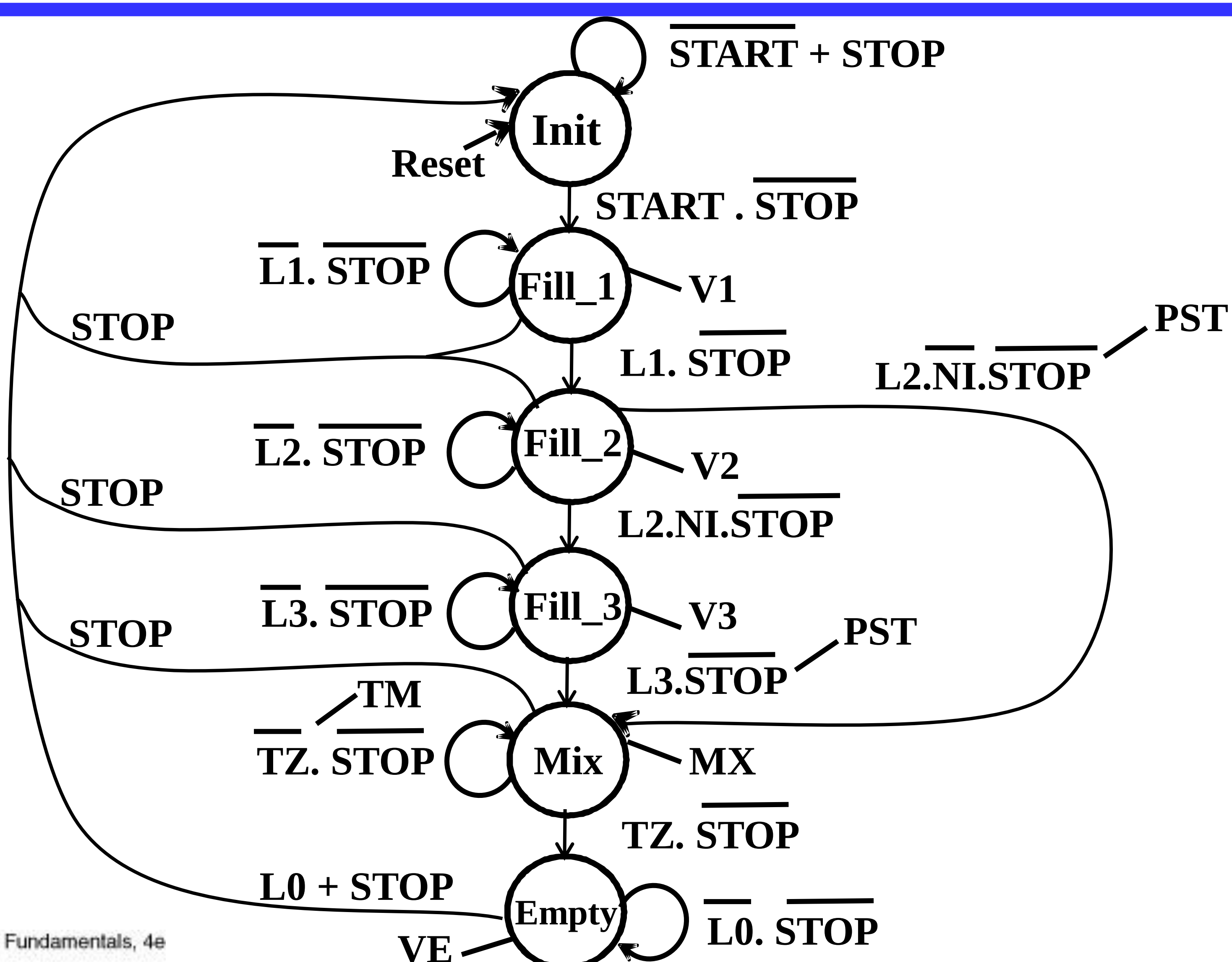
Input	Meaning for Value 1	Meaning for Value 0
NI	Three ingredients	Two ingredients
Start	Start a batch cycle	No Action
Stop	Stop an on-going batch cycle	No Action
L0	Tank empty	Tank not empty
L1	Tank filled to level 1	Tank not filled to level 1
L2	Tank filled to level 2	Tank not filled to level 2
L3	Tank filled to level 3	Tank not filled to level 3
TZ	Timer at value 0	Timer not at value 0

Example 2: Batch Mixing System Outputs

Output	Meaning for Value 1	Meaning for Value 0
MX	Mixer on	Mixer off
PST	Load timer with value from D	No Action
TM	Timer on	Timer off
V1	Valve open for ingredient 1	Valve closed for ingredient 1
V2	Valve open for ingredient 2	Valve closed for ingredient 2
V3	Valve open for ingredient 3	Valve closed for ingredient 3
VE	Output valve open	Output valve closed

Example 2: Batch Mixing System State Machine Diagram (SMD)

Defaults:
MX=0,
PST=0,
TM=0,
V1=0,
V2=0,
V3=0,
VE=0



Example 2: Batch Mixing System State Machine Table (SMT)

State	State Code	Transition Condition	Next State	Next State Code	Output Actions (OCs)
Init	100000	$\overline{START}$ + STOP	Init	100000	
	Init	START . $\overline{STOP}$	Fill_1	010000	
Fill_1	010000				V1
		STOP	Init	100000	
		$\overline{L1}$. $\overline{STOP}$	Fill_1	010000	
	Fill_1	L1 . $\overline{STOP}$	Fill_2	001000	
Fill_2	001000				V2
		STOP	Init	100000	
		$\overline{L2}$. $\overline{STOP}$	Fill_2	001000	
		L2. $\overline{NI}$. $\overline{STOP}$	Mix	000010	PST*
	Fill_2	L2 . NI . $\overline{STOP}$	Fill_3	000100	
Fill_3	000100				V3
		STOP	Init	100000	
		$\overline{L3}$. $\overline{STOP}$	Fill_3	000100	
	Fill_3	L3 . $\overline{STOP}$	Mix	000010	PST*
Mix	000010				MX
		STOP	Init	100000	
		$\overline{TZ}$. $\overline{STOP}$	Mix	000010	TM*
	Mix	TZ . $\overline{STOP}$	Empty	000001	
Empty	000001				VE
		$\overline{L0}$. $\overline{STOP}$	Empty	000001	
	Empty	L0 + STOP	Init	100000	

Example 2: Batch Mixing System Equations

- **Intermediate Variables:**
  - $X = Fill\_2 \,.\, L2 \,.\, \overline{N1} \,.\, \overline{STOP}$
  - $Y = Fill\_3 \,.\, L3 \,.\, \overline{STOP}$
  - $Z = Mix.\, \overline{TZ} \,.\, \overline{STOP}$

- **Flip-Flop Inputs:**
  - $Init(t+1) = Init \,.\, (\overline{START} + STOP\,) + STOP \,.\, (Fill\_1 + Fill\_2 + Fill\_3 + Mix) + Empty \,.\, (STOP + L0)$
  - $Init(t+1) = Init \,.\, \overline{START} + STOP + Empty \,.\, L0)$
  - $Fill\_1(t+1) = Init \,.\, START \,.\, \overline{STOP} + Fill\_1 \,.\, \overline{L1}.\, \overline{STOP}$
  - $Fill\_2(t+1) = Fill\_1.\, L1 \,.\, \overline{STOP} + Fill\_2 \,.\, \overline{L2}.\, \overline{STOP}$
  - $Fill\_3(t+1) = Fill\_2.\, L2 \,.\, NI \,.\, \overline{STOP} + Fill\_3 \,.\, \overline{L3}.\, \overline{STOP}$
  - $Mix(t+1) = X + Y + Z$
  - $Empty(t+1) = Mix \,.\, TZ \,.\, \overline{STOP} + Empty \,.\, \overline{L0} \,.\, \overline{STOP}$

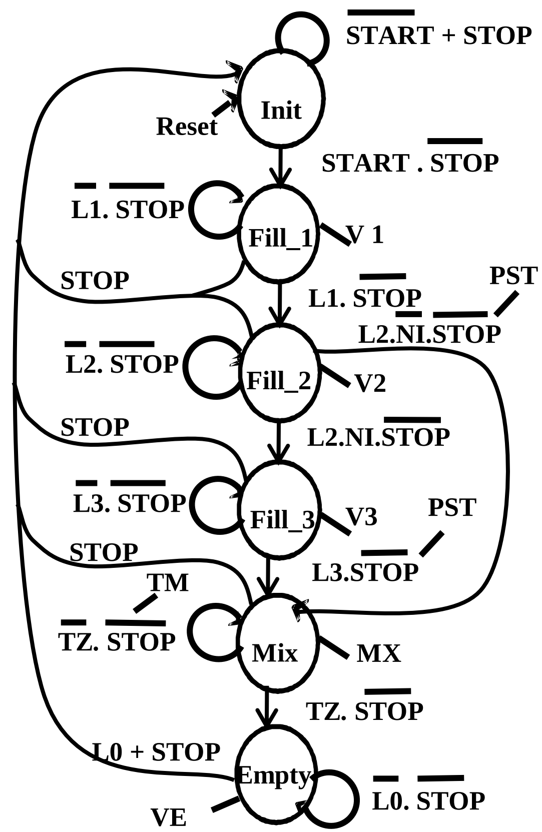
Example 2: Batch Mixing System Equations

- **Intermediate Variables:**
  - $X = Fill\_2 \cdot L2 \cdot \overline{N1} \cdot \overline{STOP}$
  - $Y = Fill\_3 \cdot L3 \cdot \overline{STOP}$
  - $Z = Mix \cdot \overline{TZ} \cdot \overline{STOP}$
- **Outputs:**
  - $V1 = Fill\_1$
  - $V2 = Fill\_2$
  - $V3 = Fill\_3$
  - $PST = X + Y$
  - $MX = Mix$
  - $TM = Z$
  - $VE = Empty$

Example 2: Batch Mixing System Constraint Checking

Constraint 1 Checking :

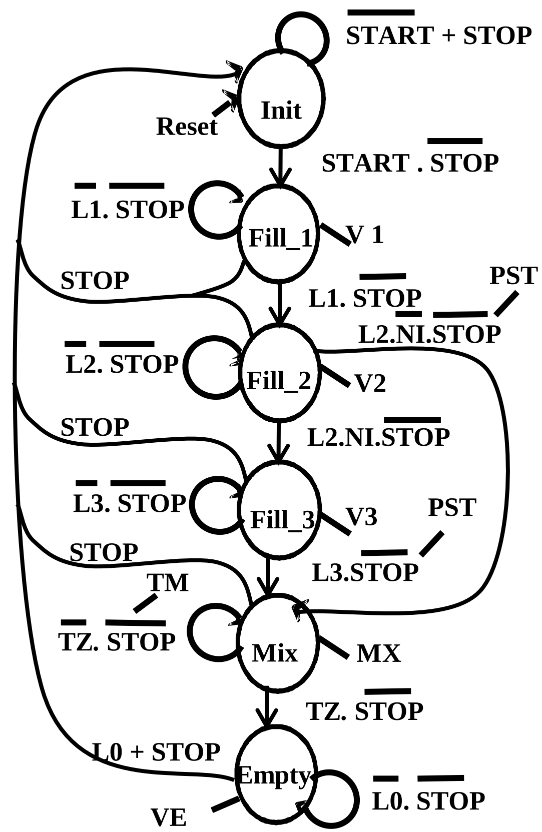
State	Constraint Checking
Init	$[(\overline{START} + STOP)] . [START . \overline{STOP}] = 0$
Fill_1	$[\overline{L1} . \overline{STOP}] . [STOP] = 0$
	$[\overline{L1} . \overline{STOP}] . [L1 . \overline{STOP}] = 0$
	$[STOP] . [L1 . \overline{STOP}] = 0$
Fill_2	$[L2 . \overline{NI} . \overline{STOP}] . [STOP] = 0$
	$[L2 . \overline{NI} . \overline{STOP}] . [\overline{L2} . \overline{STOP}] = 0$
	$[L2 . \overline{NI} . \overline{STOP}] . [L2 . NI . \overline{STOP}] = 0$
	$[STOP] . [L2 . \overline{STOP}] = 0$
	$[STOP] . [L2 . NI . \overline{STOP}] = 0$
	$[\overline{L2} . \overline{STOP}] . [L2 . NI . \overline{STOP}] = 0$



Example 2: Batch Mixing System Constraint Checking

Constraint 1 Checking (Continued):

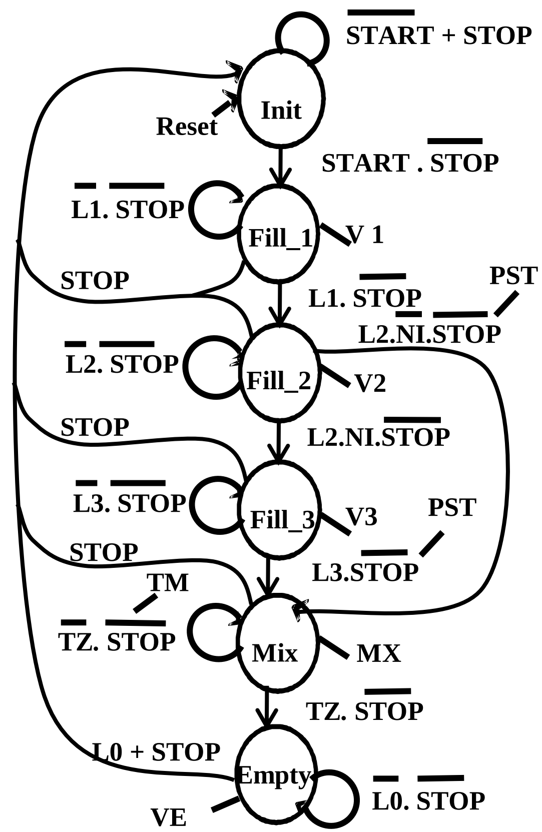
State	Constraint Checking
Fill_3	[$\overline{L3}$. $\overline{STOP}$] . [STOP] = 0
	[$\overline{L3}$. $\overline{STOP}$] . [L3 . $\overline{STOP}$] = 0
	[STOP] . [L3 . $\overline{STOP}$] = 0
Mix	[$\overline{TZ}$. $\overline{STOP}$] . [STOP] = 0
	[$\overline{TZ}$. $\overline{STOP}$] . [TZ . $\overline{STOP}$] = 0
	[STOP] . [TZ. $\overline{STOP}$] = 0
Empty	[$\overline{L0}$. $\overline{STOP}$] . [L0 + STOP] = 0



Example 2: Batch Mixing System Constraint Checking

Constraint 2 Checking (Continued):

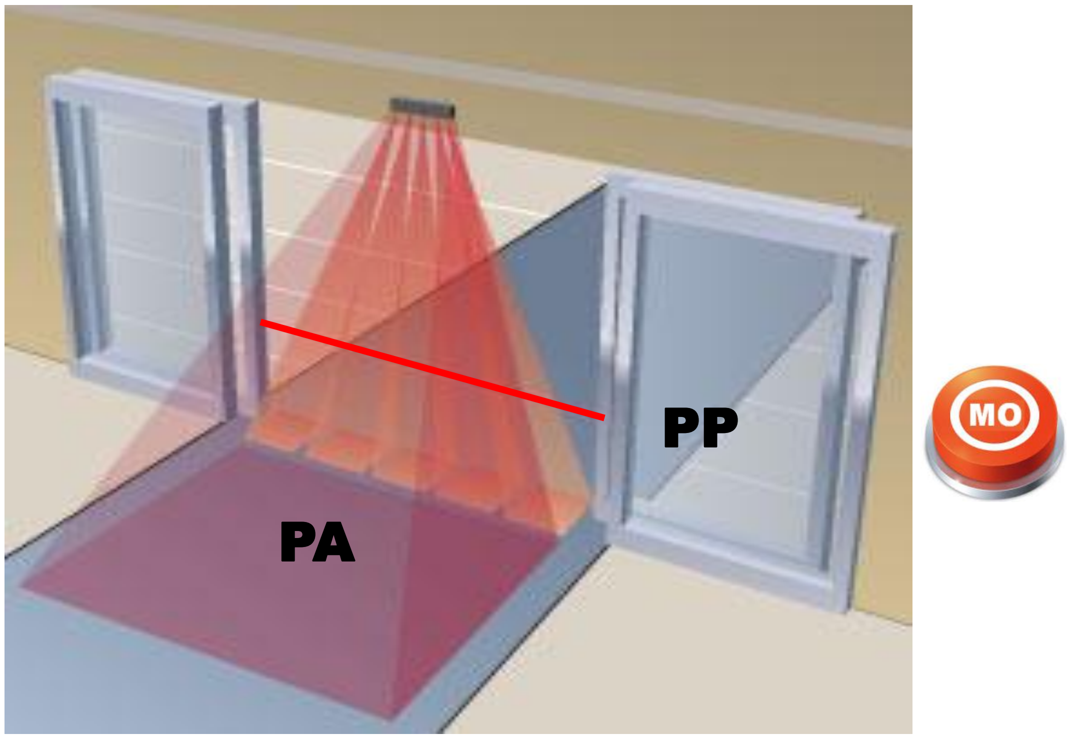
State	Constraint Checking
Init	[($\overline{START}$ + STOP)] + [START . $\overline{STOP}$] = 1
Fil_1	[$\overline{L1}$. $\overline{STOP}$] + [STOP] + [L1 . $\overline{STOP}$] = 1
Fil_2	[L2 . $\overline{NI}$. $\overline{STOP}$] + [STOP] + [$\overline{L2}$. $\overline{STOP}$] + [L2 . NI . $\overline{STOP}$] = 1
Fil_3	[$\overline{L3}$. $\overline{STOP}$] + [STOP] + [L3 . $\overline{STOP}$] = 1
Mix	[$\overline{TZ}$. $\overline{STOP}$] + [STOP] + [TZ . $\overline{STOP}$] = 1
Empty	[$\overline{L0}$. $\overline{STOP}$] + [L0 + STOP] = 1



Example 3

Sliding Door Control

Example 3: Sliding Door Control



Example 3: Sliding Door Control Inputs

Input	Name	Meaning for Value 1	Meaning for Value 0
LK	Lock with Key	Locked	Unlocked
DR	Door Resistance Sensor	Door resistance ≥ 15 lb	Door resistance < 15 lb
PA	Approach Sensor	Person/object approach	No person/object approach
PP	Presence Sensor	Person/object in door	No person/object in door
MO	Manual Open by Pushbutton	Manual Open	No Manual Open
CL	Close Limit Switch	Door fully closed	Door Not fully closed
OL	Open Limit Switch	Door fully open	Door Not fully open

Example 3: Sliding Door Control Inputs

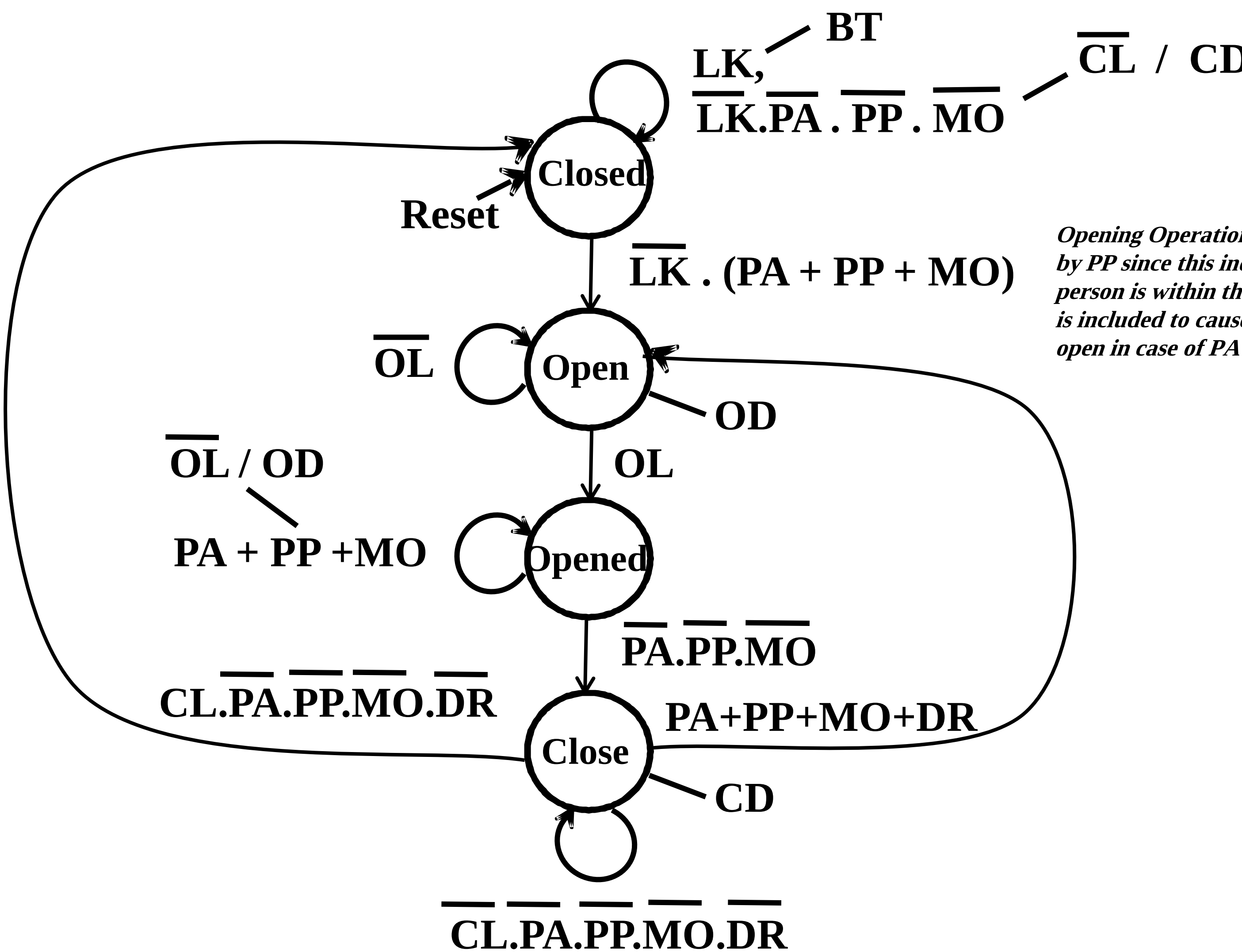
- The door opens in response to:
  - PA (Approach Sensor)
  - PP (Presence Sensor)
  - DR (Door Resistance Sensor)
  - Pushbutton MO (Manual Open)
- PA senses a person or object approaching the door.
- PP senses the presence of a person or object within the doorframe.
- DR senses a resistance to the door closing indicating that the door is pushing on a person or obstacle.
- MO is a manual pushbutton on the door control box that opens the door without dependence on the automatic control.

Example 3: Sliding Door Control Outputs

Output	Name	Meaning for Value 1	Meaning for Value 0
BT	Bolt	Bolt closed	Bolt open
CD	Close Door	Close door	No action
OD	Open Door	Open door	No action

Example 3: Sliding Door Control State Machine Design (SMD)

Defaults:
BT=0,
CD=0,
OD=0



Opening Operation is not initiated by PP since this indicates that a person is within the doorframe. It is included to cause the door to open in case of PA failure

Example 3: Sliding Door Control State Machine Table (SMT)

State	State Code	Transition Condition	Next State	Next State Code	Output Actions (OCs)
	$Y_1\ Y_2$			$Y_1\ Y_2$	
Closed	00	LK	Closed	00	BT*
		$\overline{LK} . \overline{PA} . \overline{PP} . \overline{MO}$	Closed	00	$\overline{CL}$ / CD *
	$Y_1'\ Y_2'$	$\overline{LK}$. (PA + PP + MO)	Open	01	
Open	01				OD
		$\overline{OL}$	Open	01	
	$Y_1'\ Y_2$	OL	Opened	11	
Opened	11	PA + PP + MO	Opened	11	$\overline{OL}$ / OD *
	$Y_1\ Y_2$	$\overline{PA} . \overline{PP} . \overline{MO}$	Close	10	
Close	10				CD
		$\overline{CL} . \overline{PA} . \overline{PP} . \overline{MO} . \overline{DR}$	Close	10	
		CL . $\overline{PA} . \overline{PP} . \overline{MO} . \overline{DR}$	Closed	00	
	$Y_1\ Y_2'$	PA + PP + MO + DR	Open	01	

Example 3: Sliding Door Control Equations

- **Intermediate Variables:**
  - $X = PA + PP + MO \rightarrow \overline{X} = \overline{PA}.\overline{PP}.\overline{MO}$

- **Flip-Flop Inputs:**
  - $Y_1(t+1) = \overline{Y_1} . Y_2 . OL + Y_1 . Y_2 + Y_1 . \overline{Y_2} . \overline{CL} . \overline{X} . \overline{DR}$
  - $Y_2(t+1) = \overline{Y_1} . \overline{Y_2} . \overline{LK} . X + \overline{Y_1} . Y_2 + Y_1 . Y_2 . X + Y_1 . \overline{Y_2} (X + DR)$

- **Outputs:**
  - $BT = \overline{Y_1} . \overline{Y_2} . LK$
  - $CD = Y_1 . \overline{Y_2} + \overline{Y_1} . \overline{Y_2} . \overline{LK} . \overline{CL} . \overline{X} = (Y_1 + \overline{LK} . \overline{CL} . \overline{X}) . \overline{Y_2}$
  - $OD = \overline{Y_1} . Y_2 + Y_1 . Y_2 . \overline{OL} . X = (\overline{Y_1} + \overline{OL} . X) . Y_2$

Simplification Theorem:
$x + x'y = x + y$

Example 4

Elevator Control

Elevator control for two-floor elevator
Warning: Does not include safety features or all user buttons!

Example 4: Elevator Control Inputs

Input	Name	Meaning for Value 1	Meaning for Value 0
C1(C2)	Call button (outside elevator) to floor 1(2)	Call for elevator	No action
G1(G2)	Go button (inside elevator) to floor 1(2)	Go to floor command	No action
F1(F2)	Senses elevator at floor 1(2)	Elevator at floor	Elevator not at floor
S1(S2)	Senses elevator approaching floor 1(2) (Controls slowdown of elevator)	Elevator approaching floor	Elevator not approaching floor
DO	Doors open?	Doors fully open	Doors not fully open
TO	End of time interval from button push to elevator movement starting	Time interval has ended	Waiting for time interval to end
DC	Doors closed?	Doors closed	Doors not closed

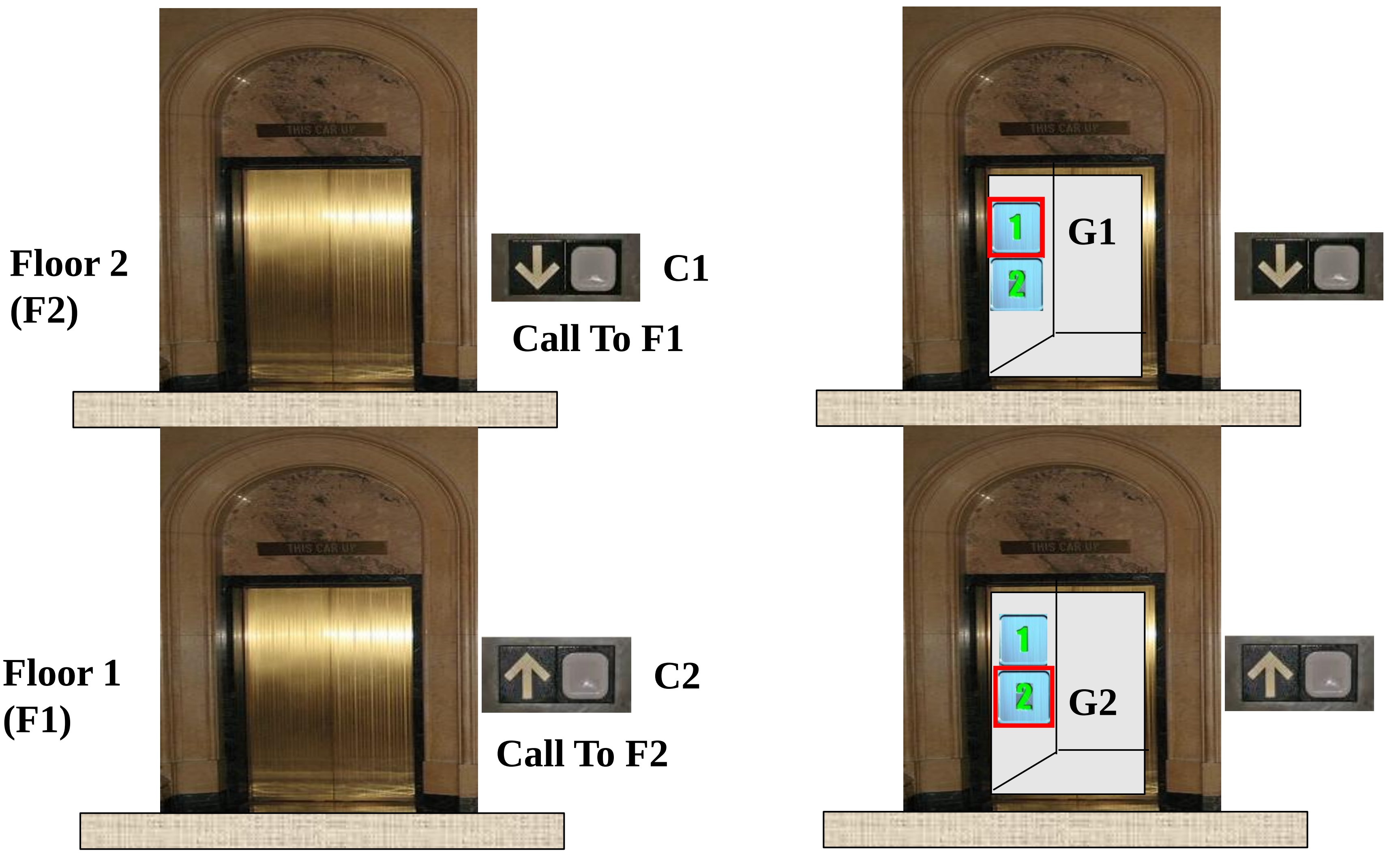
Example 4: Elevator Control Outputs

Output	Name	Meaning for Value 1	Meaning for Value 0
Up	Elevator to go up	Commands elevator to go up	No action
Down	Elevator to go down	Commands elevator to go down	No action
TS	Timer start	Initialize and start timer	No action
SD	Slow down	Elevator approaching target floor slows down	Elevator moves as normal speed
OD	Open doors	Open doors	No action
CD	Close doors	Close doors	No action

Example 4: Elevator Control Specifications

- **The elevator parks at the floor to which it has last taken passengers with doors open.**
- **Call button C_i calls elevator to a floor.**
- **If the elevator is not at the floor, TS is used to initialize and start the timer;**
- **After TO becomes 1, the doors close, and when DC is active, the Up or Down output is activated.**
- **The S_i sensor detects the floor approach and activates output SD to slow elevator.**
- **The F_i sensor detects the elevator at the floor, forces both Up and Dn to 0, and opens the doors.**
- **Passenger(s) enter elevator and push the G_i button.**
- **After TO becomes 1, the doors close, and when DC is active, the Up or Down output is activated.**
- **The S_i sensor detects the approach and activates output SD to slow elevator.**
- **The F_i sensor detects the elevator at the floor, forces both Up and Dn to 0, and opens the doors, permitting passengers to exit.**

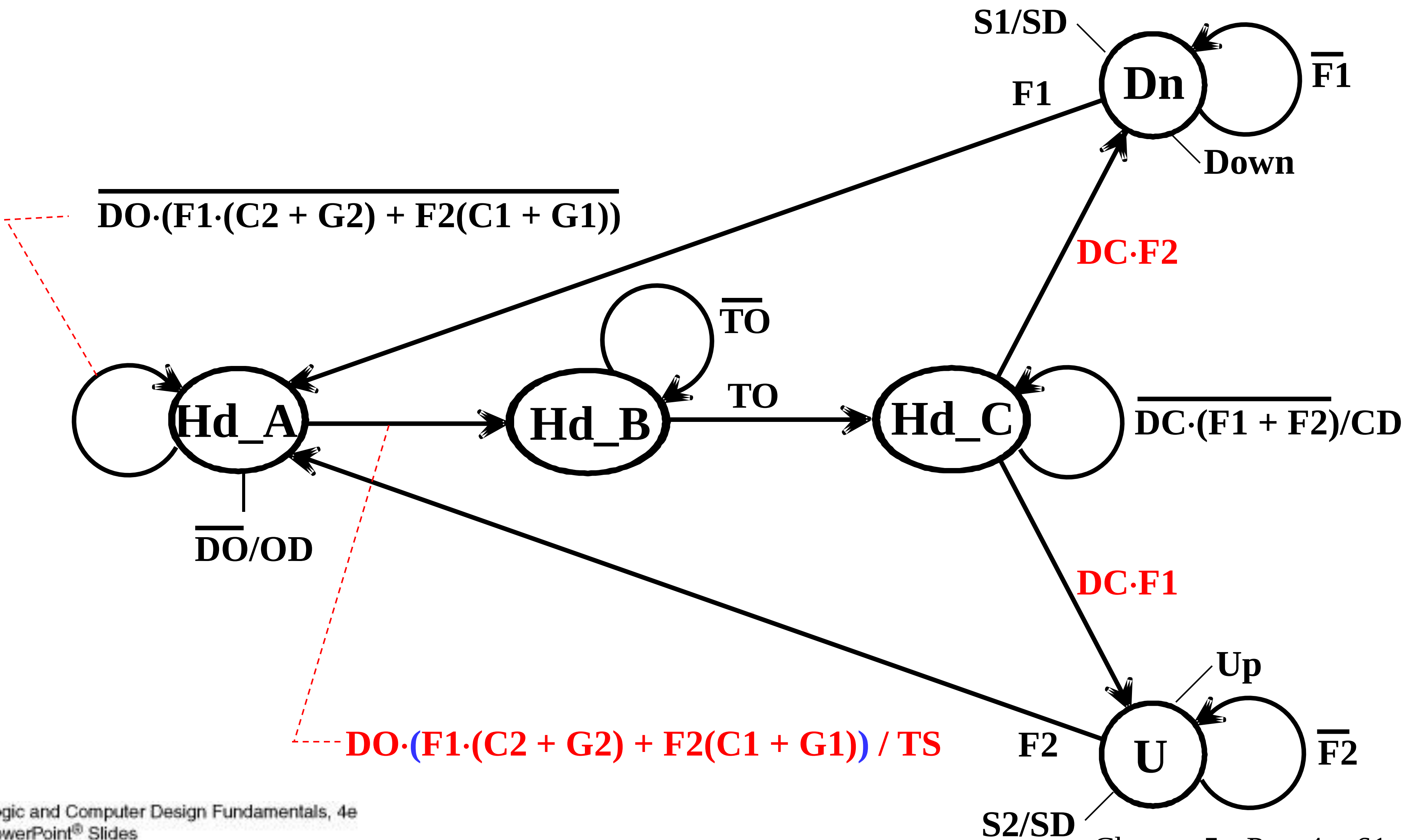
Example 4: Elevator Control



Example 4: Elevator Control States

- **Initial proposed states:**
  - **U (Up)**
  - **Dn (Down)**
  - **Hd (Hold)**
- **Series of actions required in Hd state:**
  - **Open doors (Hd_A)**
  - **Use timer to wait for passengers (Hd_B)**
  - **Close doors (Hd_C)**
- **Expand Hd to 3 states: Hd_A, Hd_B, Hd_C**
- **One-Hot State Vector:**
  (U, Dn, Hd_C, Hd_B, Hd_A)

Example 4: Elevator Control State Machine Diagram (SMD)



Example 4: – Elevator Control State Machine Table (SMT)

State	State Code	Transition Condition	Next State	Next State Code	Output Actions (OCs)
	U, Dn, Hd_C, Hd_B, Hd_A			U, Dn, Hd_C, Hd_B, Hd_A	
Hd_A	00001				$\overline{DO}$/OD
		$\overline{DO \cdot (F1 \cdot (C2 + G2) + F2 \cdot (C1 + G1))}$	Hd_A	00001	
	Hd_A	$DO \cdot (F1 \cdot (C2 + G2) + F2 \cdot (C1 + G1))$	Hd_B	00010	TS
Hd_B	00010	$\overline{TO}$	Hd_B	00010	
	Hd_B	TO	Hd_C	00100	
Hd_C	00100	$\overline{DC \cdot (F1 + F2)}$	Hd_C	00100	CD
		$DC \cdot F2$	Dn	01000	
	Hd_C	$DC \cdot F1$	U	10000	
Dn	01000				Down, S1/SD
		$\overline{F1}$	Dn	01000	
	Dn	F1	Hd_A	00001	
U	10000				Up, S2/SD
		$\overline{F2}$	U	10000	
	U	F2	Hd_A	00001	

Example 4: Elevator Control Equations

- **Flip-Flop Inputs:**
  - $X = DO \cdot ((F1 \cdot (C2 + G2) + F2 \cdot (C1 + G1))$
  - $Y = DC \cdot (F1 + F2)$
  - $D_{Hd\_A} = Hd\_A \cdot \overline{X} + Dn \cdot F1 + U \cdot F2$
  - $D_{Hd\_B} = Hd\_A \cdot X + Hd\_B \cdot \overline{TO}$
  - $D_{Hd\_C} = Hd\_B \cdot TO + Hd\_C \cdot \overline{Y}$
  - $D_{Dn} = Hd\_C \cdot DC \cdot F2 + Dn \cdot \overline{F1}$
  - $D_U = Hd\_C \cdot DC \cdot F1 + U \cdot \overline{F2}$
- **Outputs:**
  - $Down = Dn$
  - $Up = U$
  - $SD = Dn \cdot S1 + U \cdot S2$
  - $TS = Hd\_A \cdot X$
  - $OD = Hd\_A \cdot \overline{DO}$
  - $CD = Hd\_C \cdot \overline{Y}$

Terms of Use

- **All (or portions) of this material © 2008 by Pearson Education, Inc.**
- **Permission is given to incorporate this material or adaptations thereof into classroom presentations and handouts to instructors in courses adopting the latest edition of Logic and Computer Design Fundamentals as the course textbook.**
- **These materials or adaptations thereof are not to be sold or otherwise offered for consideration.**
- **This Terms of Use slide or page is to be included within the original materials or any adaptations thereof.**