

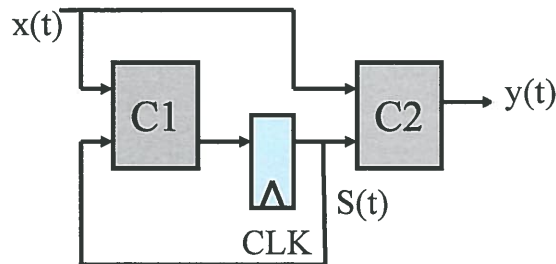
Mealy and Moore Machines: Canonical Form

Mealy Machine: $y_i(t) = f_i(X(t), S(t))$

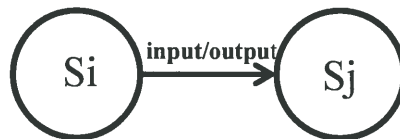
Moore Machine: $y_i(t) = f_i(S(t))$

$s_i(t+1) = g_i(X(t), S(t))$

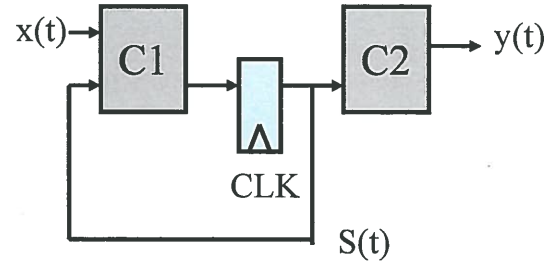
Mealy Machine



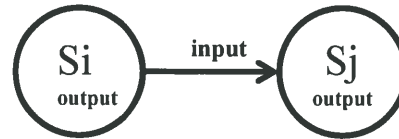
	Input
PS	NS, output



Moore Machine



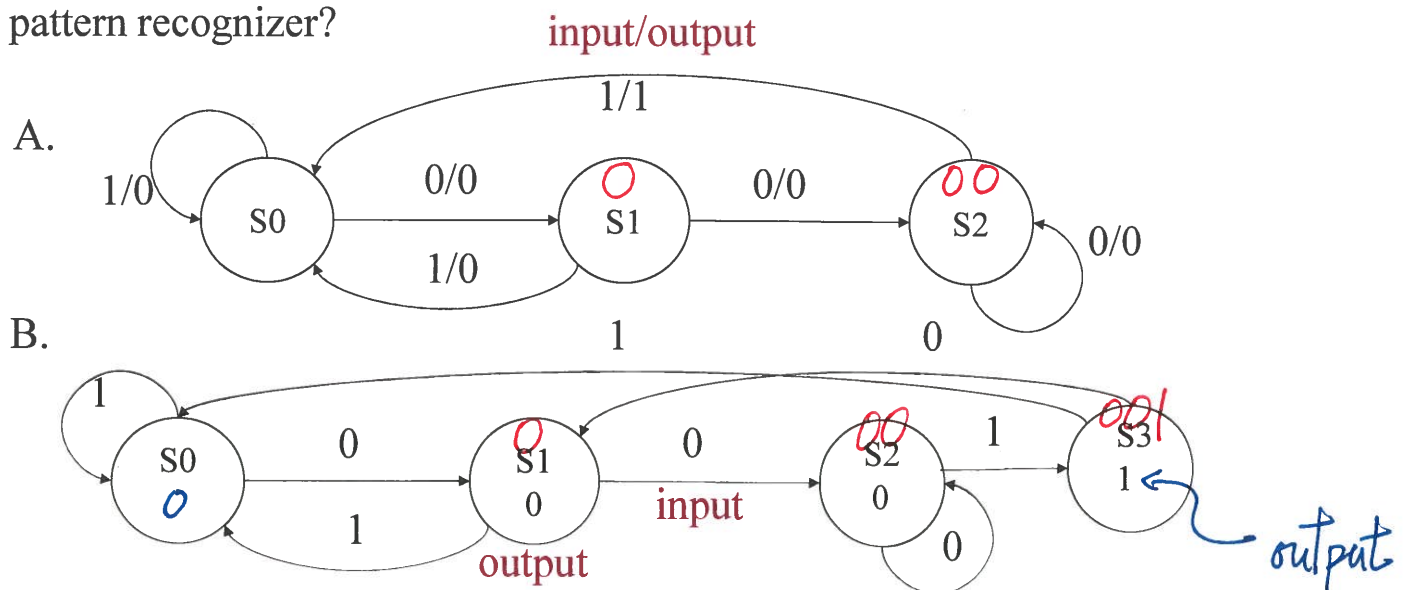
	Input	
PS	NS	Output



29

Mealy and Moore Machines: Mars Life Recognizer

Which of the following diagrams is a correct Moore solution to the '001' pattern recognizer?



C. Both A and B are correct

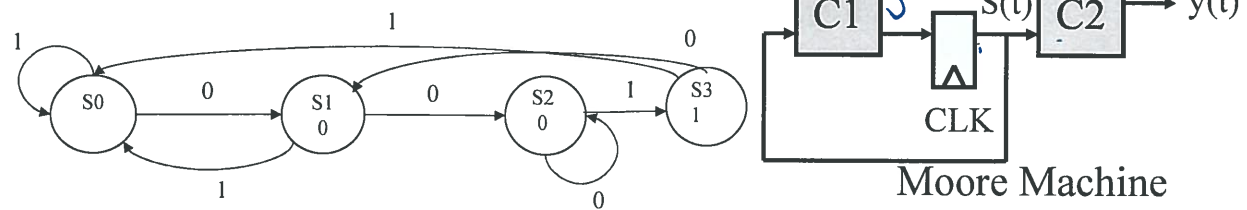
D. None of the above

30

$$DFFs \quad Q(t+1) = D(t)$$

Mealy and Moore Machines: Mars Life Recognizer

Pattern Recognizer '001'

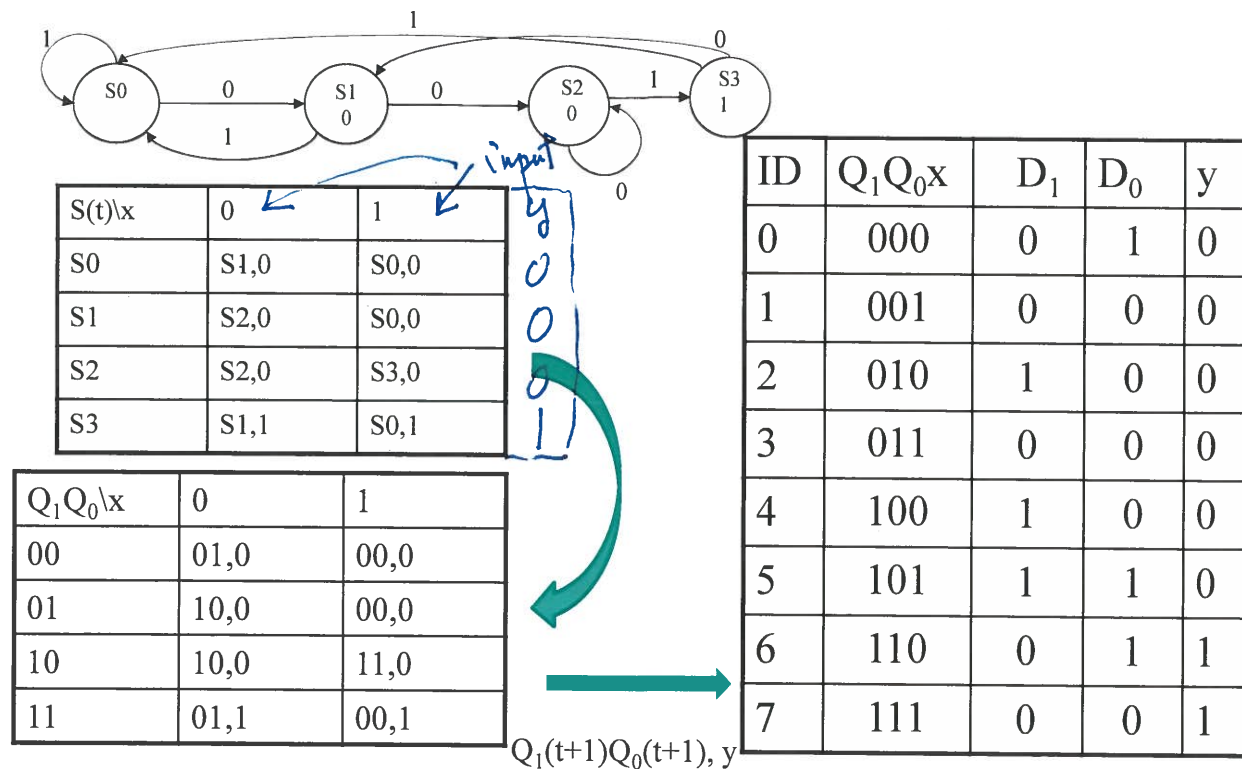


What does state table need to show to design controls of C2?

- A.(current input $x(t)$, current state $S(t)$ vs. next state, $S(t+1)$)
- B.(current input, current state vs. current output $y(t)$)
- C.(current state vs. current output $y(t)$ and next state)
- D.(current state vs. current output $y(t)$)
- E.None of the above

31

Mealy and Moore Machines: Mars Life Recognizer



Mealy and Moore Machines: Mars Life Recognizer

id	$Q_1 Q_0 x$	D_1	D_0	y
0	000	0	1	0
1	001	0	0	0
2	010	1	0	0
3	011	0	0	0
4	100	1	0	0
5	101	1	1	0
6	110	0	1	1
7	111	0	0	1

$D_1(t)$:

	Q_0			
	0	1	0	1
$x(t)$	0	0	0	1
	Q_1			

$D_0(t)$:

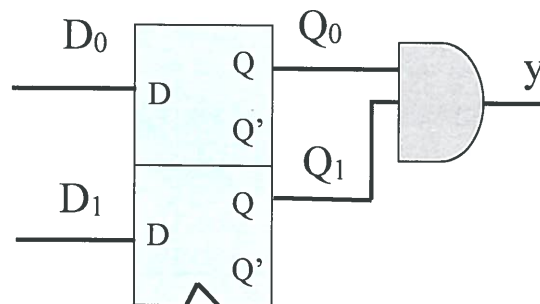
	Q_0			
	1	0	1	0
$x(t)$	0	0	0	1
	Q_1			

$y(t)$:

	Q_0			
	0	0	1	0
$x(t)$	0	0	1	0
	Q_1			

Mealy and Moore Machines: Mars Life Recognizer

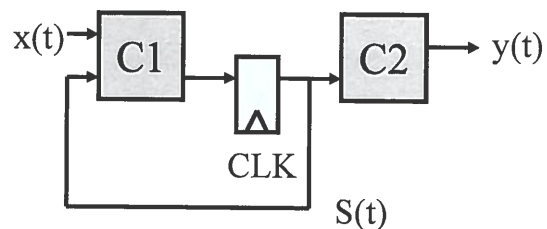
State Diagram $\Rightarrow$ State Table $\Rightarrow$ Excitation Table $\Rightarrow$ **Circuit**



$$D_1(t) = Q_1(t)Q_0(t)' + Q_1(t)'Q_0(t)x(t)$$

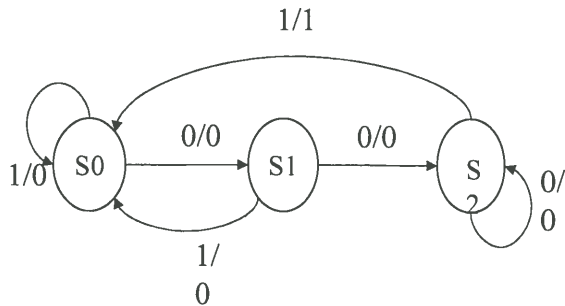
$$D_0(t) = Q_1(t)'Q_0(t)'x(t)' + Q_1(t)Q_0(t)x(t)' + Q_1(t)Q_0(t)x(t)$$

$$y(t) = Q_1(t)Q_0(t)$$



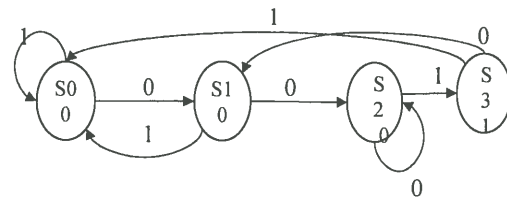
Moore Machine

Conversion from Mealy to Moore Machine



S(t)\x	0	1
S0	S1,0	S0,0
S1	S2,0	S0,0
S2	S2,0	S0,1

replace
S0,1



S(t)\x	0	1
S0	S1,0	S0,0
S1	S2,0	S0,0
S2	S2,0	S3,0
S3	S1,1	S0,1

Conversion from Mealy to Moore Machine

S(t)\x	0	1
S0	S1,0	S0,0
S1	S2,0	S0,0
S2	S2,0	S0,1



S(t)\x	0	1	y
S0	S1	S0	0
S1	S2	S0	0
S2	S2	S3	0
S3	S1	S0	1

Algorithm

1. Identify distinct (NS, y) pair
2. Replace each distinct (NS, y) pair with distinct new states
3. Insert rows of present state = new states

1. (NS, y): (S0, 0), (S0, 1) S3
 (S1, 0)
 (S2, 0)

Conversion from Mealy to Moore Machine

Mealy	S(t)\x	0	1
	S0	S1,0	S0,0
	S1	S2,0	S0,0
	S2	S2,0	S0,1

S(t)\x	0	1	y
S0	S1	S0	0
S1	S2	S0	0
S2	S2	S3	0
S3			

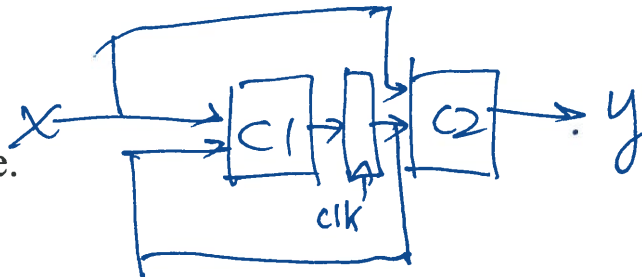
 Moore |

1. Find distinct NS, y
2. Add new states to represent distinct NS, y

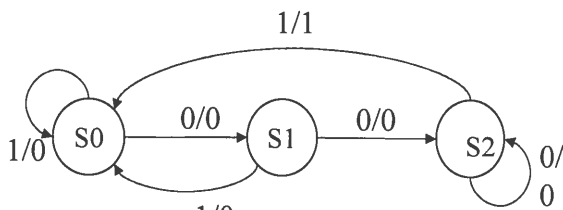
iClicker

For the above Moore machine, what are the next states with respect to present state S3?

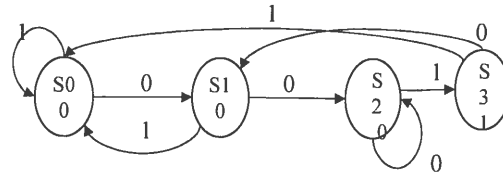
- A. S2, S3, 1
- B. S2, S0, 1
- C. S1, S0, 1
- D. S1, S0, 0
- E. None of the above.



Conversion from Mealy to Moore Machine



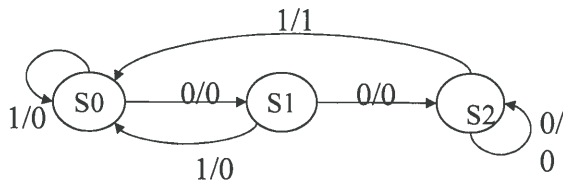
S(t)\x	0	1
S0	S1,0	S0,0
S1	S2,0	S0,0
S2	S2,0	S0,1



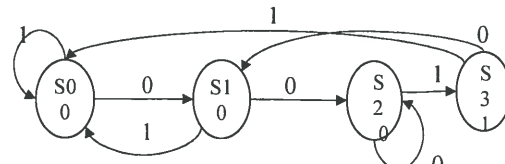
S(t)\x	0	1
S0	S1,0	S0,0
S1	S2,0	S0,0
S2	S2,0	S3,0
S3	S1,1	S0,1

Time	0	1	2	3	4	5	6	7	8
x	0	1	0	0	1	1	0	0	1
S _{mealy}	S0	S1	S0	S1	S2	S0	S0	S1	S2
y _{mealy}	0	0	0	0	1	0	0	0	1
S _{moore}	S0	S1	S0	S1	S2	S3	S0	S1	S2
y _{moore}	0	0	0	0	0	1	0	0	0

Conversion from Mealy to Moore Machine



S(t)\x	0	1
S0	S1,0	S0,0
S1	S2,0	S0,0
S2	S2,0	S0,1



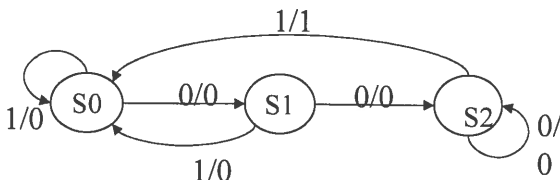
S(t)\x	0	1
S0	S1,0	S0,0
S1	S2,0	S0,0
S2	S2,0	S3,0
S3	S1,1	S0,1

Time	0	1	2	3	4	5	6	7	8
x	0	1	0	0	1	1	0	0	1
S _{mealy}	S0	S1	S0	S1	S2	S0	S0	S1	S2
y _{mealy}									
S _{moore}	S0								
y _{moore}									

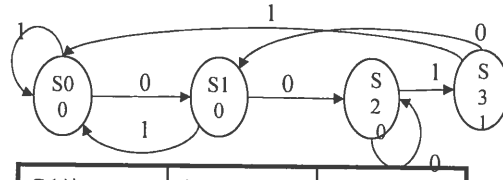
iClicker S_{moore}[0-5]

- A. S0,S1,S0,S1,S2,S3
- B. S0,S1,S0,S1,S2,S0
- C. S3,S1,S0,S1,S2,S3
- D. S3,S1,S0,S1,S2,S0
- E. None of the above

Conversion from Mealy to Moore Machine



S(t)\x	0	1
S0	S1,0	S0,0
S1	S2,0	S0,0
S2	S2,0	S0,1



S(t)\x	0	1
S0	S1,0	S0,0
S1	S2,0	S0,0
S2	S2,0	S3,0
S3	S1,1	S0,1

Time	0	1	2	3	4	5	6	7	8
x	0	1	0	0	1	1	0	0	1
S _{mealy}	S0	S1	S0	S1	S2	S0	S0	S1	S2
y _{mealy}									
S _{moore}	S0								
y _{moore}									

iClicker y_{moore}[0-5]

- A. 0,0,0,0,1,0
- B. 0,0,0,0,0,1
- C. 0,1,0,0,0,0
- D. 0,0,0,0,0,0,
- E. None of the above

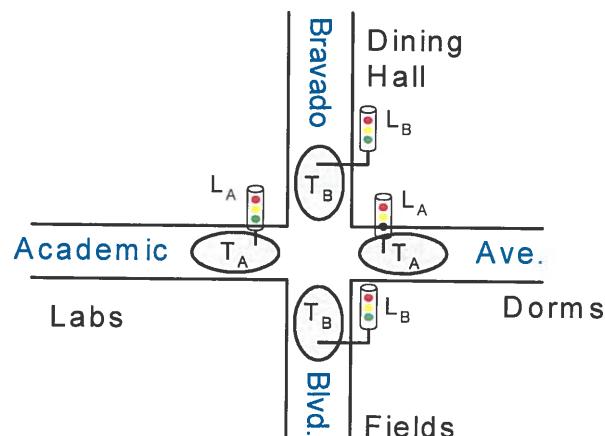
Conversion from Mealy to Moore Machine

Algorithm

1. Identify distinct (NS, y) pair
2. Replace each distinct (NS, y) pair with distinct **new states**
3. Insert rows of present state = **new states**
4. Append each present state with its output y

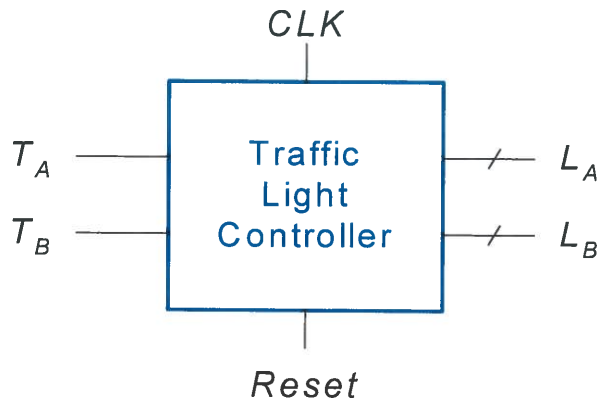
Example: Traffic Light Controller

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



Traffic Light Controller: FSM Black Box

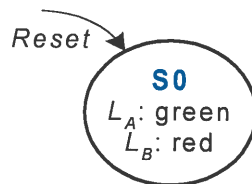
- Inputs: CLK , $Reset$, T_A , T_B
- Outputs: L_A , L_B



43

FSM State Transition Diagram

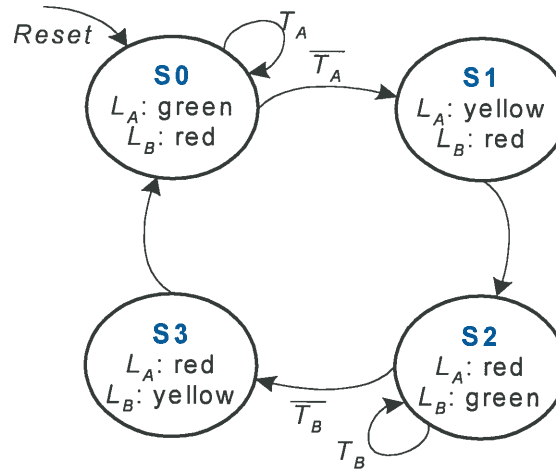
- Moore FSM: outputs labeled in each state
- States: Circles
- Transitions: Arcs



44

FSM State Transition Diagram

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



45

FSM State Transition Table

PS	Inputs		NS
	T_A	T_B	
S0	0	X	S1
S0	1	X	S0
S1	X	X	S2
S2	X	0	S3
S2	X	1	S2
S3	X	X	S0

46

State Transition Table

PS		Inputs		NS	
$Q_1(t)$	$Q_0(t)$	T_A	T_B	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	X	0	1
0	0	1	X	0	0
0	1	X	X	1	0
1	0	X	0	1	1
1	0	X	1	1	0
1	1	X	X	0	0

State	Encoding
S0	00
S1	01
S2	10
S3	11

$$Q_1(t+1) = Q_1(t) \oplus Q_0(t)$$

$$Q_0(t+1) = Q_1'(t)Q_0'(t)T_A' + Q_1(t)Q_0'(t)T_B'$$

47

Traffic Light Controller: FSM Output Table

PS		Outputs			
Q_1	Q_0	L_{A1}	L_{A0}	L_{B1}	L_{B0}
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	0	0
1	1	1	0	0	1

Output	Encoding
green	00
yellow	01
red	10

$$L_{A1} = Q_1$$

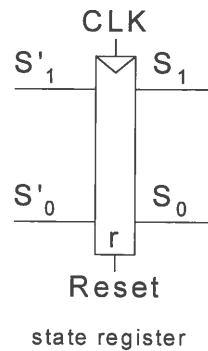
$$L_{A0} = Q_1'Q_0$$

$$L_{B1} = Q_1'$$

$$L_{B0} = Q_1Q_0$$

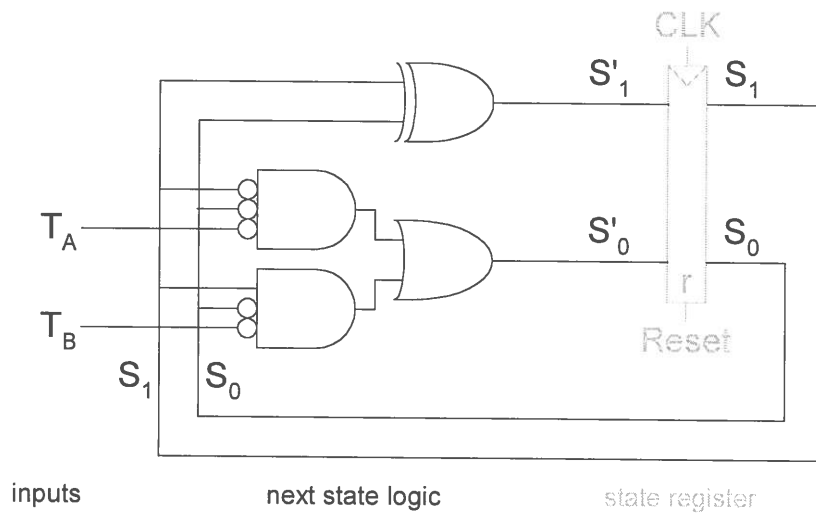
48

Traffic Light Controller: State Register



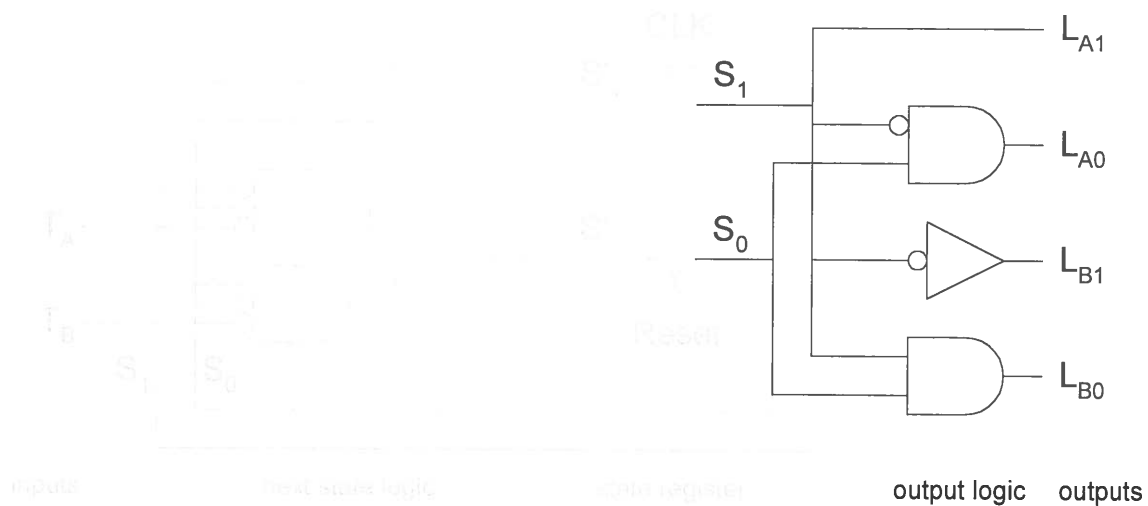
49

Traffic Light Controller: Logic Diagram



50

Traffic Light Controller: Output Logic



51

Summary: Implementation

- Set up canonical form
 - Mealy or Moore machine
- Identify the next states
 - state diagram $\Rightarrow$ state table
 - state assignment
- Derive excitation table
 - Inputs of flip flops
- Design the combinational logic
 - don't care set utilization

52

CSE 140: Components and Design Techniques for Digital Systems

Lecture 10: Sequential Networks: Timing and Retiming

CK Cheng

Dept. of Computer Science and Engineering
University of California, San Diego

1

Timing

- Motivation
- Basic Timing Characteristic
 - Gate Delay, Interconnect Delay
 - Flip-Flop Timing Window
- Two Timing Constraints: shortest and longest timing paths
 - Examples
- Clock Skews and Retiming
 - Examples

2

Timing: Motivation

- Clock specifies a precise time for the next state
 - In general, we allocate one clock period for signal propagation between registers. **Goldilocks** timing.
- Too late: Fail to reach for the **setup** of the next state.
- Too early: Race to disturb the **holding** of the next state.
- Analysis: Verify the timing of the system.
- Goal: A robust design.

3

The Story of Goldilocks and the Three Bears

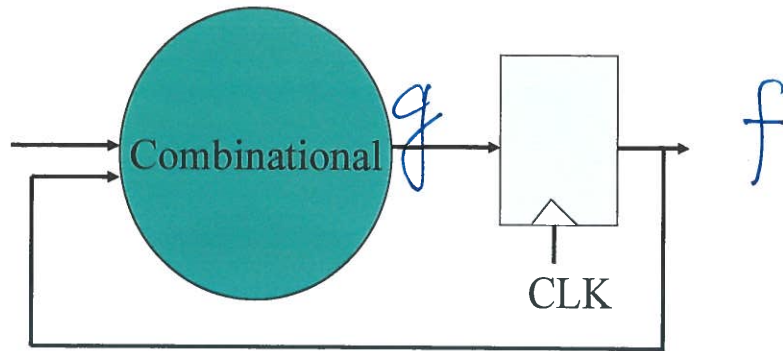
Once upon a time, there was a little girl named **Goldilocks**. She went for a walk in the forest. Pretty soon, she came upon a house. She knocked and, when no one answered, she walked right in. At the table in the kitchen, there were three bowls of porridge. Goldilocks was hungry. She tasted the porridge from the first bowl. "This porridge is **too hot**!" she exclaimed. So, she tasted the porridge from the second bowl. "This porridge is **too cold**," she said. So, she tasted the last bowl of porridge. "Ahhh, this porridge is **just right**," she said happily and she ate it all up.

DLTK's Crafts for Kids



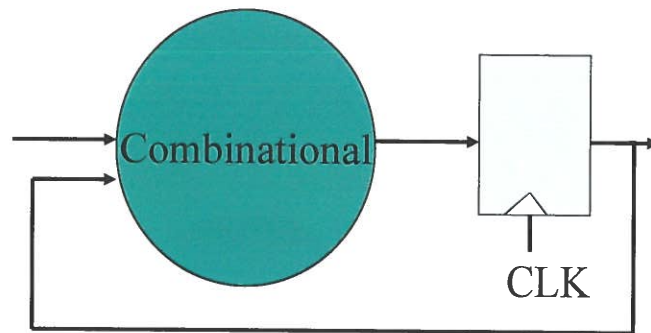
4

Motivation: So far



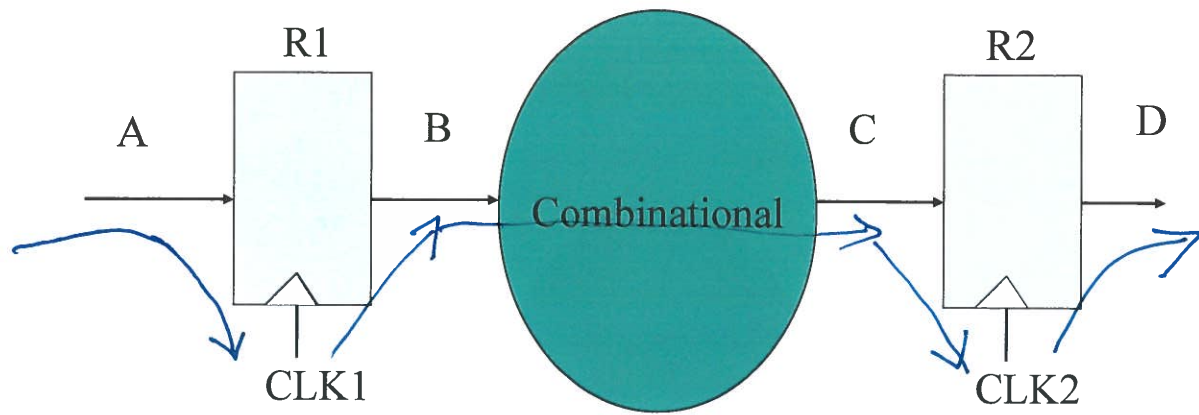
Logic-level analysis

Motivation: This lecture ...



- When does our (seemingly logically correct) design go wrong?
- How can we design a circuit that works under real constraints?
- Popular interview questions.

Motivation: Sequential Networks



A typical sequential network has combinational circuit between registers ($R1$ to $R2$).

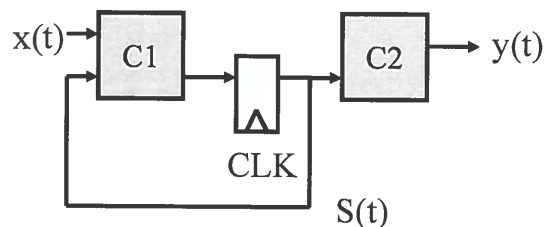
The registers are synchronized by clocks ($CLK1$ and $CLK2$).

Timing is set between clocks ($CLK1$ and $CLK2$).

The beauty of the synchronized design is that we need only to take care of the timing of the regions separated by the registers.

7

Timing of the System

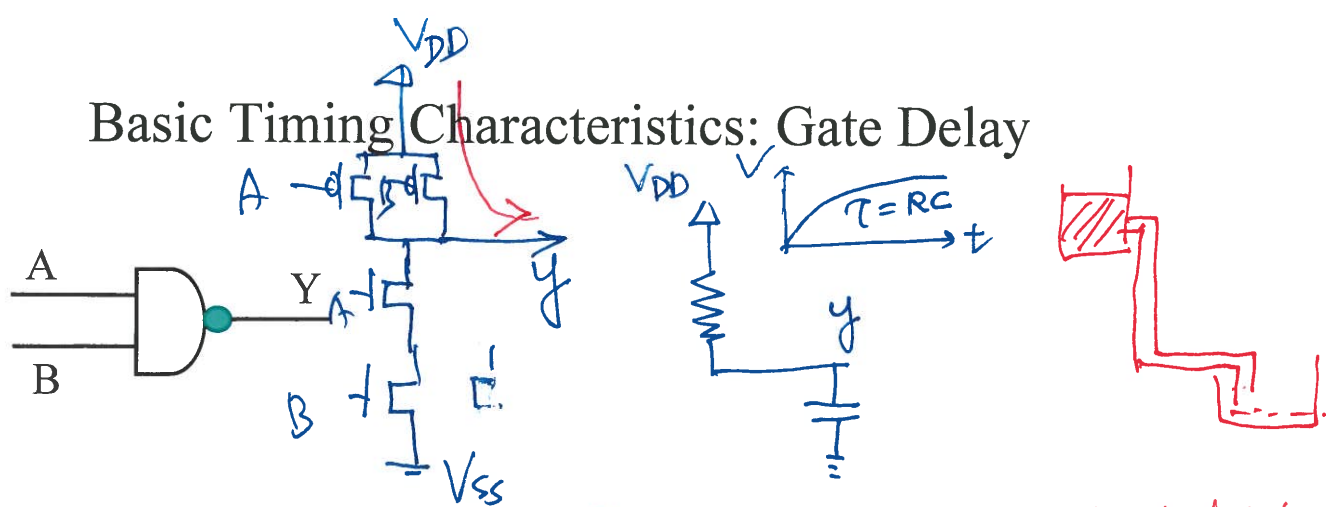


For a synchronized digital Moore machine, we take care of the timing of the following three groups of paths between:

- Every pair of registers,
- Each input and register, and
- Each register and output.

8

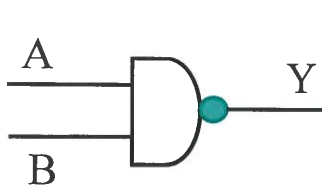
Basic Timing Characteristics: Gate Delay



- I. Min delay of a gate, also called **Contamination delay: t_{cd}** *shortest delay*
Minimum time from when an input changes until the output *starts* to change
- II. Max delay of a gate, also called **Propagation delay: t_{pd}** *longest delay*
Maximum time from when an input changes until the output *is* guaranteed to reach its final value (i.e., stop changing)

9

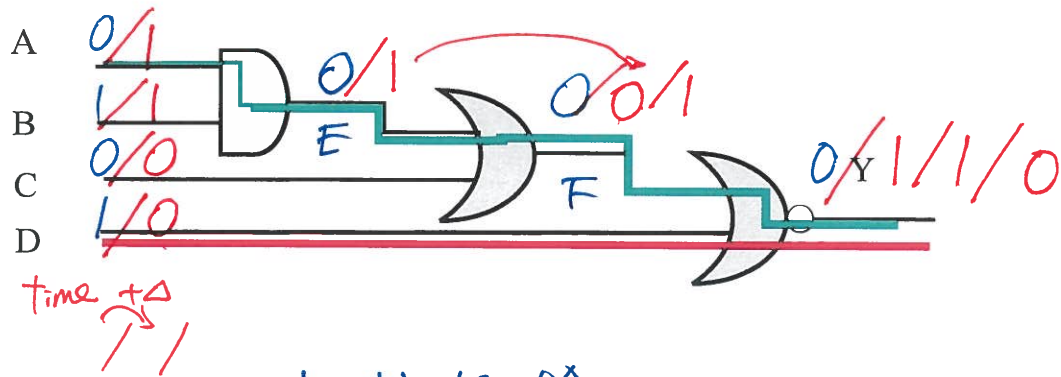
Gate Delay: Combinational Logic Timing



$A_t B_t$	$A_{t+1} B_{t+1}$	$Y = (AB)'$
00	11	$1 \Rightarrow 0$
01	11	$1 \Rightarrow 0$
10	11	$1 \Rightarrow 0$
11	11	$0 \Rightarrow 0$

Different situation causes different delay at output, e.g.
input patterns, process variations, noises, temperature, aging

Combinational Logic Delay



Different path causes $t_0 + \Delta + \Delta + \Delta$ different output transition delay.

A	0	1	1	1	1
B	1	1	1	1	1
C	0	0	0	0	0
D	1	0	0	0	0
E	0	0	1	1	1
F	0	0	0	1	1
Y	0	0	1	1	0

$t_{cd} = \Delta$
 $t_{pd} = 3\Delta$

11

Interconnect Delay



Speed of light: $C/\sqrt{\epsilon} \approx 1.5 \times 10^{10}$ cm/s

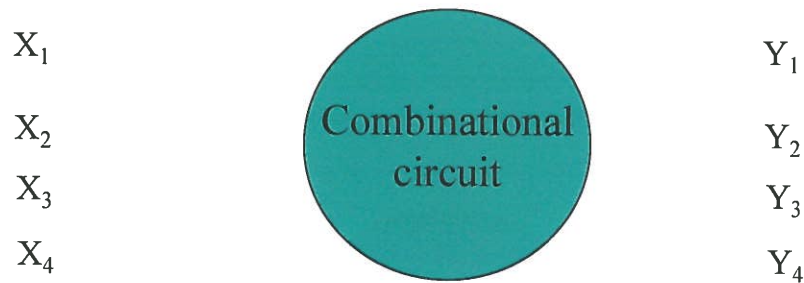
For 1cm, it takes 0.7×10^{-10} s = 70ps (picosecond)

for the light to reach from one end to the other end.

Chain of buffers: 5-40 times of speed of light, i.e. 350-2,800ps.

For 5GHz, the clock period is 200ps/cycle.

Combinational Logic: Output timing constraints



I. Contamination delay (shortest): t_{cd}

Minimum time from when an input changes until any output *starts* to change

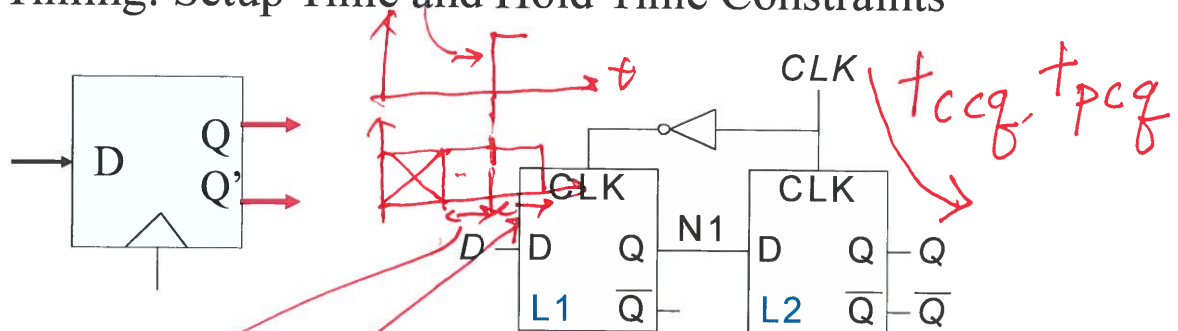
II. Propagation delay (longest): t_{pd}

Maximum time from when an input changes until the output *or* outputs of a combinational circuit are guaranteed to reach their final value (i.e., stop changing)

13

Flip-Flop Timing Window

Timing: Setup Time and Hold Time Constraints



Given a flip flop, its timing behavior is simulated/tested and specified with timing characteristics:

t_{setup} , t_{hold} timing relation between D and CLK

t_{ccq} , t_{pcq} timing relation between CLK and Q

No direct timing relation between input D and output Q

14