

Lecture 8:
Sequential Networks and Finite State
Machines

CK Cheng

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

1

3. Implementation

- Specification:
 - Finite State Machine:
 - Behavior, State Diagram, State Table
 - Register Transfer Level Description
- Design Flow
- Excitation Table $\rightarrow y^{t+1} = f(x^t, s^t)$
- Mealy and Moore Machines
- Examples

Specification

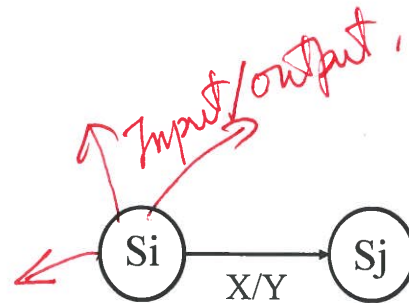
- Combinational Logic
 - Truth Table
 - Boolean Expression
 - Logic Diagram (No feedback loops)
- Sequential Networks: Finite State Machines
 - State Diagram, State Assignment, State Table
 - Excitation Table and Characteristic Expression
 - Register-Transfer Level Description
 - Logic Diagram (FFs and feedback loops)

3

FINITE STATE MACHINE: OVERVIEW USING AN EXAMPLE

State Table and State Diagram

<i>PS</i>	<i>X0</i>	<i>X1</i>	...	<i>Xm</i>
<i>S0</i>	<i>S_i, Y</i>			
<i>S1</i>				
<i>S_n</i>				



$$\begin{cases} s_i(t+1) = g_i(X(t), S(t)) \\ y_i(t+) = f_i(X(t), S(t)) \end{cases}$$

PS: Present State NS: Next State
Y: Output.

FINITE STATE MACHINE: OVERVIEW USING AN EXAMPLE

Not all sequences are created equal: A coin has two faces. A toss of the coin lands either Head or Tail with equal probability. A series of tossing creates a sequence. Given two patterns, say HHH and HTT, the pattern appears first wins.

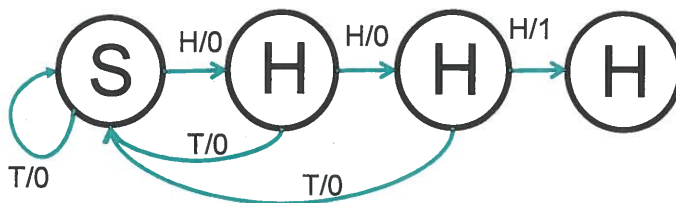
(H)(H)(H)(T)(H)(H)

- A. The two patterns have an equal chance to win
- B. HHH wins more times
- C. HTT wins more times
- D. None of the above

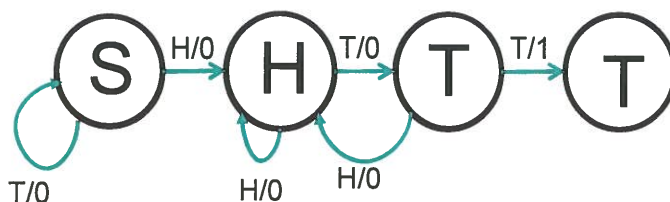


FINITE STATE MACHINE: NOT ALL SEQUENCES ARE CREATED EQUAL

State Diagram of HHH

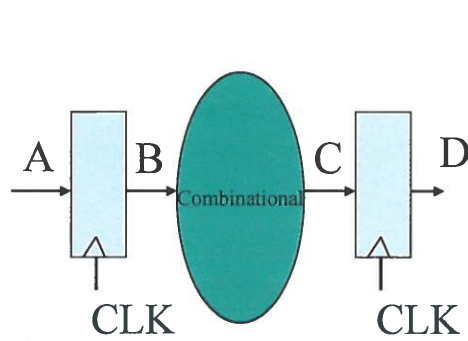


State Diagram of HTT



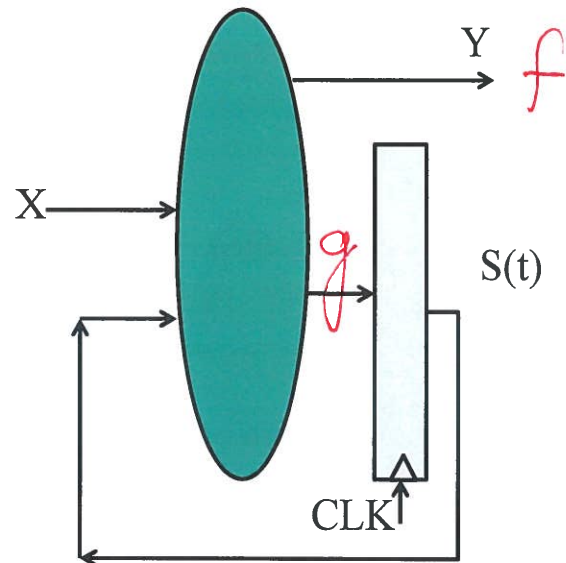
Different diagrams
yield different
expectations

Specification: Register-Transfer Level Description



RTL: Register-Transfer Level Description

- VHDL
- Verilog
- BVS



Conceptually, we can align all registers into one single column

7

Design Flow

- Input Output Relation
- State Diagram (Transition of states)
- State Assignment (Map states into binary code)
 - Binary code, Gray encoding, One hot encoding, Coding optimization
- State Table (Truth table of states)
- Excitation Table (Truth table of FF inputs)
 - K Map, Minimal Expression
 - Logic Diagram
- Verification: Simulation

8

Design Flow: Examples

- Example 1: a counter from description, state diagram, state table, netlist, to I/O behavior.
- Example 2: a pattern recognizer from state diagram to netlist to demonstrate the difference of two machines.

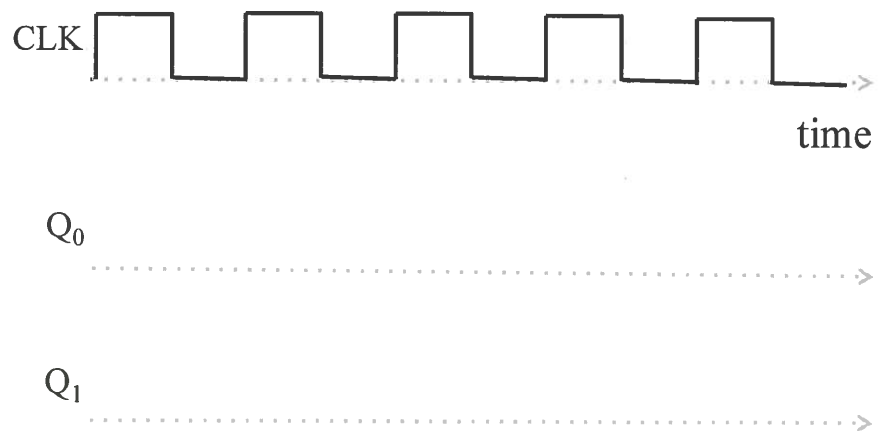
9

Example: A Counter

Symbol/ Circuit



Behavior over time



$$Q(t + 1) = (Q(t) + 1) \bmod 4$$

10

Counter: Input-output relation $\Rightarrow$ State diagram

Symbol/ Circuit

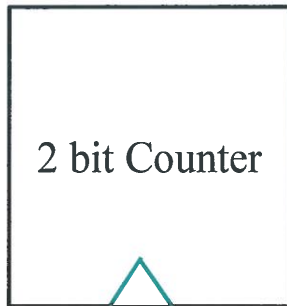
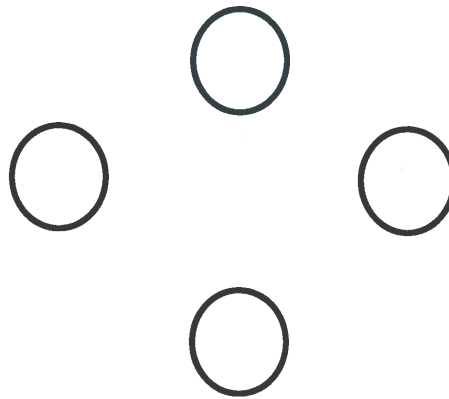


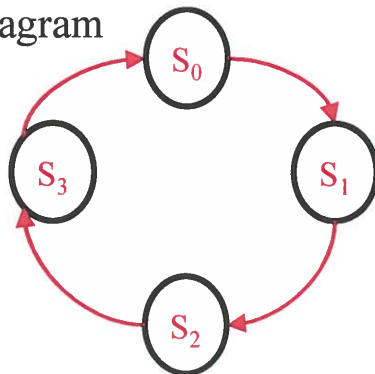
Diagram that depicts behavior over time



11

Counter: State diagram $\Rightarrow$ State assignment

State Diagram



State Table: Symbol

Current state	Next State
S_0	S_1
S_1	S_2
S_2	S_3
S_3	S_0

State Assignment

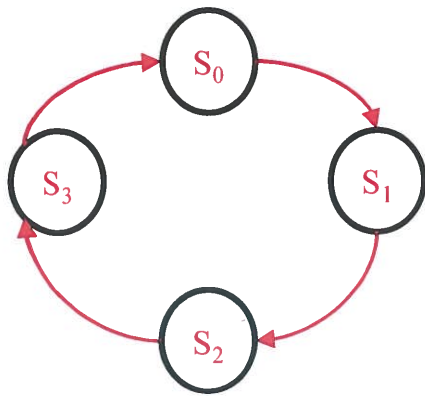
State	Q_1	Q_0
S_0	0	0
S_1	0	1
S_2	1	0
S_3	1	1

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$

State Table: Binary

12

Counter: State assignment $\Rightarrow$ Excitation table



State Diagram

Current state	Next State
S_0	S_1
S_1	S_2
S_2	S_3
S_3	S_0

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

State Table

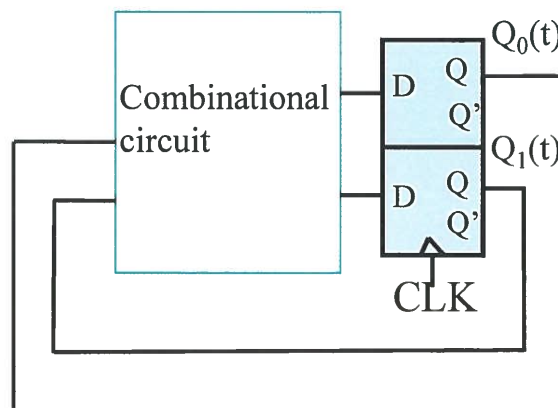
13

Counter: Excitation table $\Rightarrow$ K map $\Rightarrow$ Logic Diagram

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

$$D_0(t) = Q_0(t)'$$

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



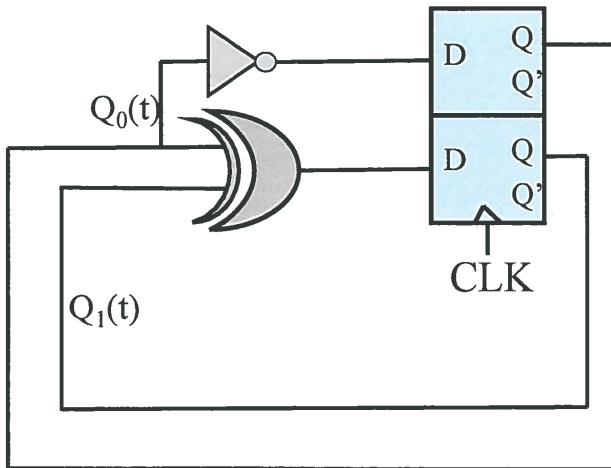
Circuit with 2 flip flops

14

Counter: Excitation table $\Rightarrow$ K map $\Rightarrow$ Logic Diagram

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

State Table



We store the current state using D-flip flops so that:

- Inputs to the combinational circuit don't change while the next output is being computed
- The transition to the next state only occurs at the rising edge of the clock

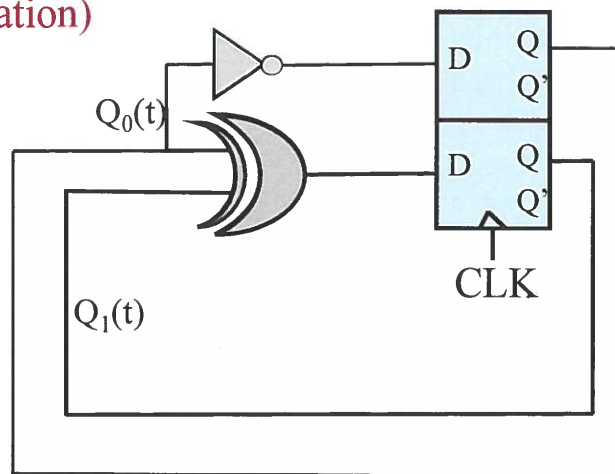
$$Q_0(t+1) = Q_0(t)'$$

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

15

Counter: Verification (Simulation)

Current state	Next State
S_0	S_1
S_1	S_2
S_2	S_3
S_3	S_0



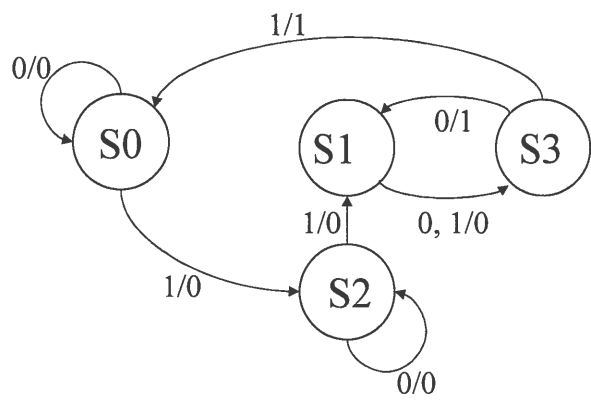
Example: State sequence

Time	0	1	2	3	4	5
Q_1	0					
Q_0	0					

16

Example: Simulation from state table/state diagram

PS\Input	X=0	X=1
S0	S0,0	S2,0
S1	S3,0	S3,0
S2	S2,0	S1,0
S3	S1,1	S0,1



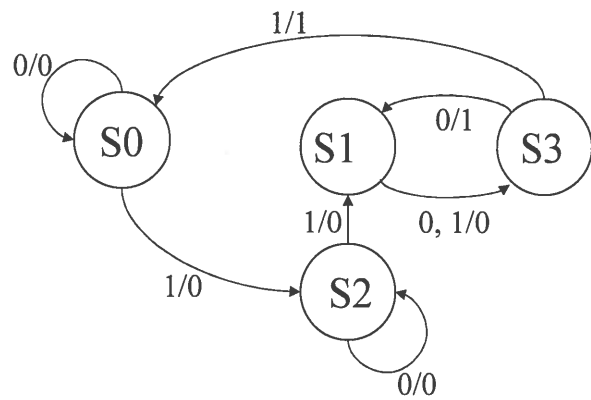
Example: Output sequence

Time	0	1	2	3	4	5
Input	0	1	1	0	1	-
State	S0					
Output						

17

Example: Simulation from state table/state diagram

PS\Input	X=0	X=1
S0	S0,0	S2,0
S1	S3,0	S3,0
S2	S2,0	S1,0
S3	S1,1	S0,1



Example: Output sequence

Time	0	1	2	3	4	5
Input	0	1	1	0	1	-
State	S0	S0	S2	S1	S3	S0
Output	0	0	0	0	1	0

18

Example: Life on Mars?

Mars rover has a binary input x . When it receives the input sequence $x(t-2, t) = 001$ from its life detection sensors, it means that it has detected life on Mars ☺ and the output $y(t) = 1$, otherwise $y(t) = 0$ (no life on Mars ☹).

Implement the Life-on-Mars
Pattern Recognizer!



19

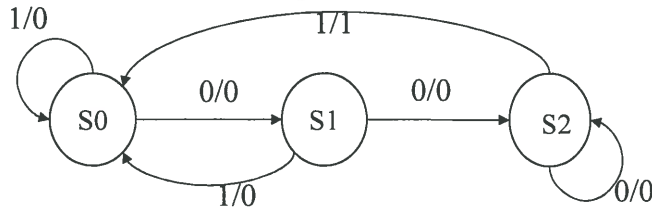
Mars Life Recognizer FSM

Which of the following diagrams is a correct Mealy solution for the 001 pattern recognizer on the Mars rover?

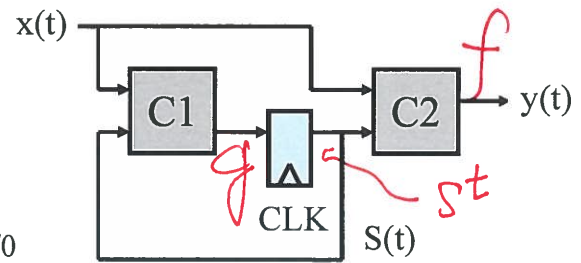
- A.
- B.
- C. Both A and B are correct
- D. None of the above

Mars Life Recognizer FFs

Pattern Recognizer '001'



Mealy Machine



What does state table need to show to design controls of C1?

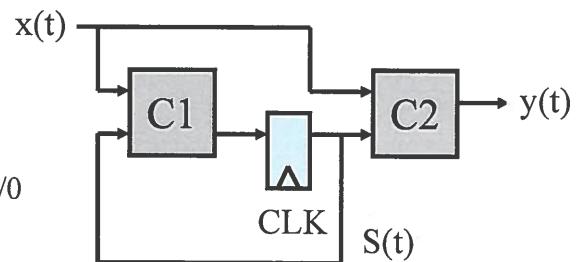
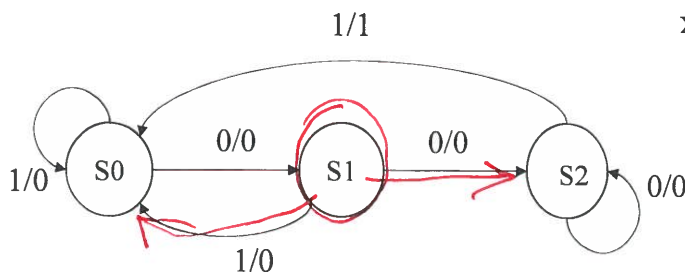
- A. next state $S(t+1)$ vs. input $x(t)$, and present state $S(t)$
- B. output $y(t)$ vs. input $x(t)$, and present state $S(t)$
- C. output $y(t)$ vs. present state $S(t)$
- D. None of the above

$$y = f(x^t, s^t)$$

$$s^{t+1} = g(x^t, s^t)$$

21

State Diagram => State Table with State Assignment



Mealy Machine

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

State Assignment

S0: 00

S1: 01

S2: 10

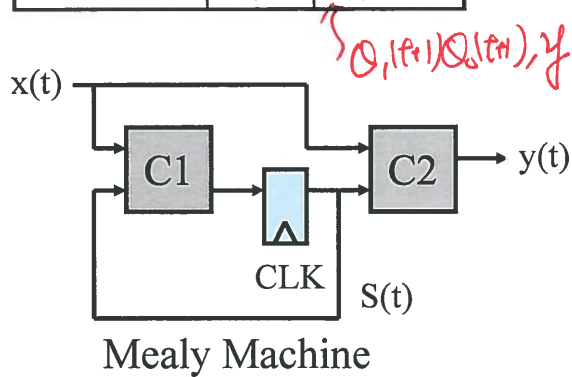
$S(t) \backslash x$	0	1
00	01,0	00,0
01	10,0	00,0
10	10,0	00,1

$Q_1(t+1)Q_0(t+1), y$

PS (Present State) and NS, y (Next State, output)

State Diagram => State Table => Excitation Table => Circuit

$Q_1(t) Q_0(t) \backslash x$	0	1
00	01,0	00,0
01	10,0	00,0
10	10,0	00,1

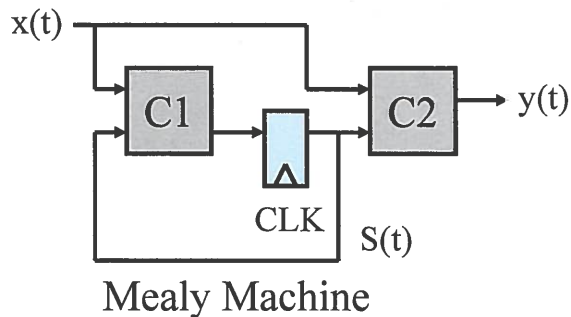


id	$Q_1 Q_0 x$	D_1	D_0	y
0	000	0	1	0
1	001	0	0	0
2	010	1	0	0
3	011	0	0	0
4	100	1	0	0
5	101	0	0	1
6	110	—	—	—
7	111	—	—	—

23

State Diagram => State Table => Excitation Table => Circuit

$Q_1(t) Q_0(t) \backslash x$	0	1
00	01,0	00,0
01	10,0	00,0
10	10,0	00,1



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	0	0	1
6	110			
7	111			

Example: What to fill in rows 6 and 7 of excitation table?

- A. All 0s
- B. All 1s
- C. All Don't Cares

24

State Assignment

K-map

State Diagram => State Table => Excitation Table => Circuit

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	0	0	1
6	110	X	X	X
7	111	X	X	X

$D_1(t)$:

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

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

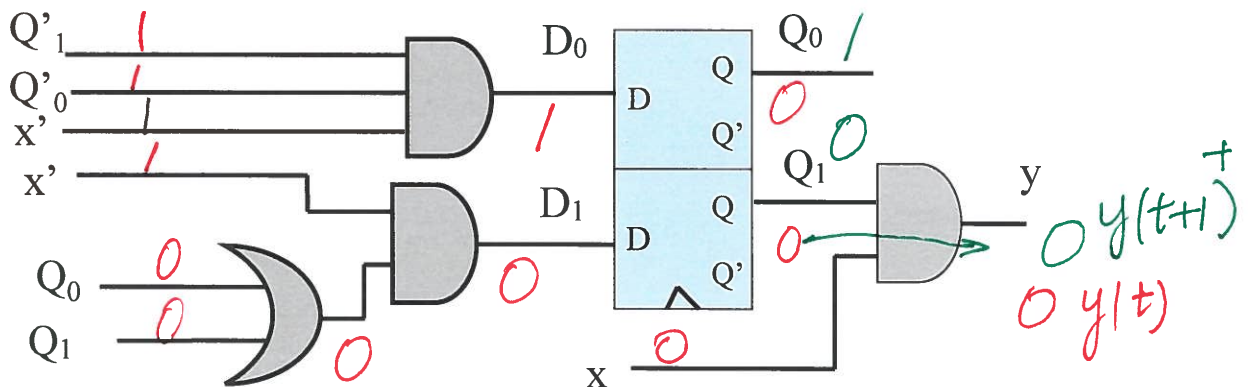
$$D_0(t) = Q_1'Q_0'x'$$

$$y = Q_1x$$

$\Sigma m(4,6): Q_1(t)X'(t)$
 $\Sigma m(2,6): Q_0(t)x'(t)$

25

State Diagram => State Table => Excitation Table => Circuit

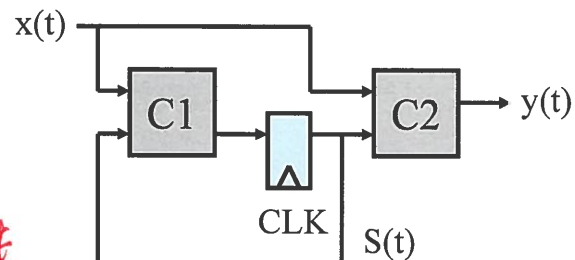


$$D_1(t) = x'Q_0 + x'Q_1$$

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

$$y = Q_1x$$

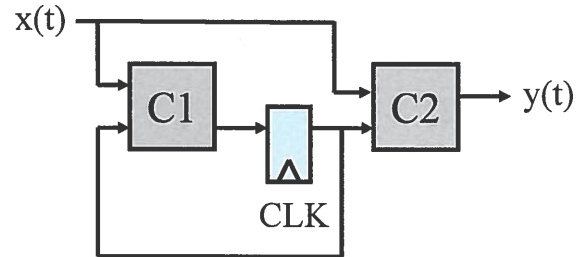
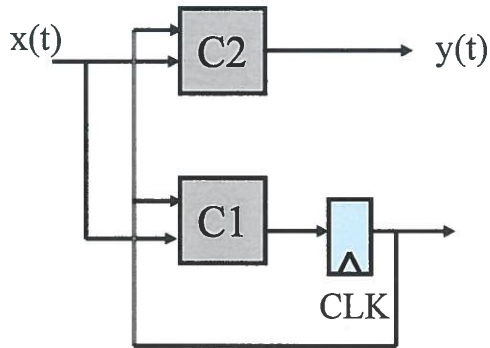
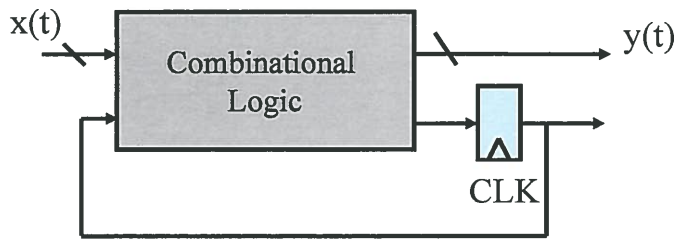
$Q_1 Q_0 x \rightarrow 0 0 0 \rightarrow 0 1 0$



Mealy Machine

26

Mealy and Moore Machines: Canonical Form



27

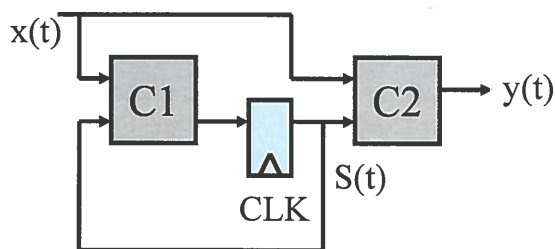
Mealy and Moore Machines: Canonical Form

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

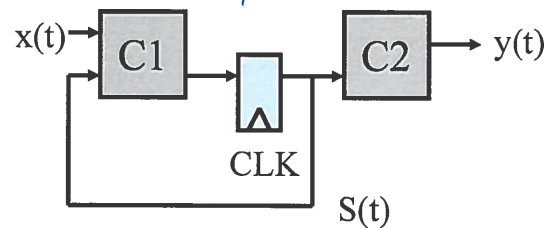
Moore Machine: $y_i(t) = f_i(S(t))$

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

*The timing of output y is determined by
 ↗ CLK + delay of C2*



Mealy Machine



Moore Machine

28