Lecture Outline

Reminders to self:

- ❑ Turn on lecture recording to Cloud
- ❑ Turn on Zoom microphone

- Last Lecture
  - Continued analysis & design of clocked sequential circuits
    - Finished Mealy 101 sequence detector design started last lecture
    - Analysis example (of Mealy sequence detector just designed)
    - State Graph design guidelines
    - Started a larger (8 state) design example (Mustang turn signals)
- Today's Lecture
  - Continue analysis & design of clocked sequential circuits
    - Finish the larger (8 state) design example (Mustang turn signals)
    - Five-variable K-Maps
    - Start larger Mealy design example

Handouts and Announcements

- Announcements
  - Homework Problems: No new assignment
  - Homework Reminder
    - HW 13-1 Due: 11:25am Wednesday 3/29
    - HW 13-2 Due: 11:59pm Thursday 3/30
  - Read for Wednesday: no new reading assignment
    previous assignment pages 463-472, 149-151

Handouts and Announcements

- Announcements
  - Mini-Exam 5 Reminder
    - Available 5pm Monday 3/27 through 5:00pm Tuesday 3/28
    - Due in Carmen PROMPTLY at 5:00pm on 3/28
    - Designed to be completed in ~36 min, but you may use more
    - When planning your schedule:
      - I recommend building in 10-15 min extra
      - To allow for downloading exam, signing and dating honor pledge, saving solution as pdf, and uploading to Carmen
    - I also recommend not procrastinating
  - Exam review topics available on Carmen
  - Sample Mini-Exams 6 and 7 from Au20 also available

Design Example: Mustang Sequential Turn Signals

Left $L_2L_1L_0$ Right $R_0R_1R_2$

Where we left off last lecture

← The six outputs

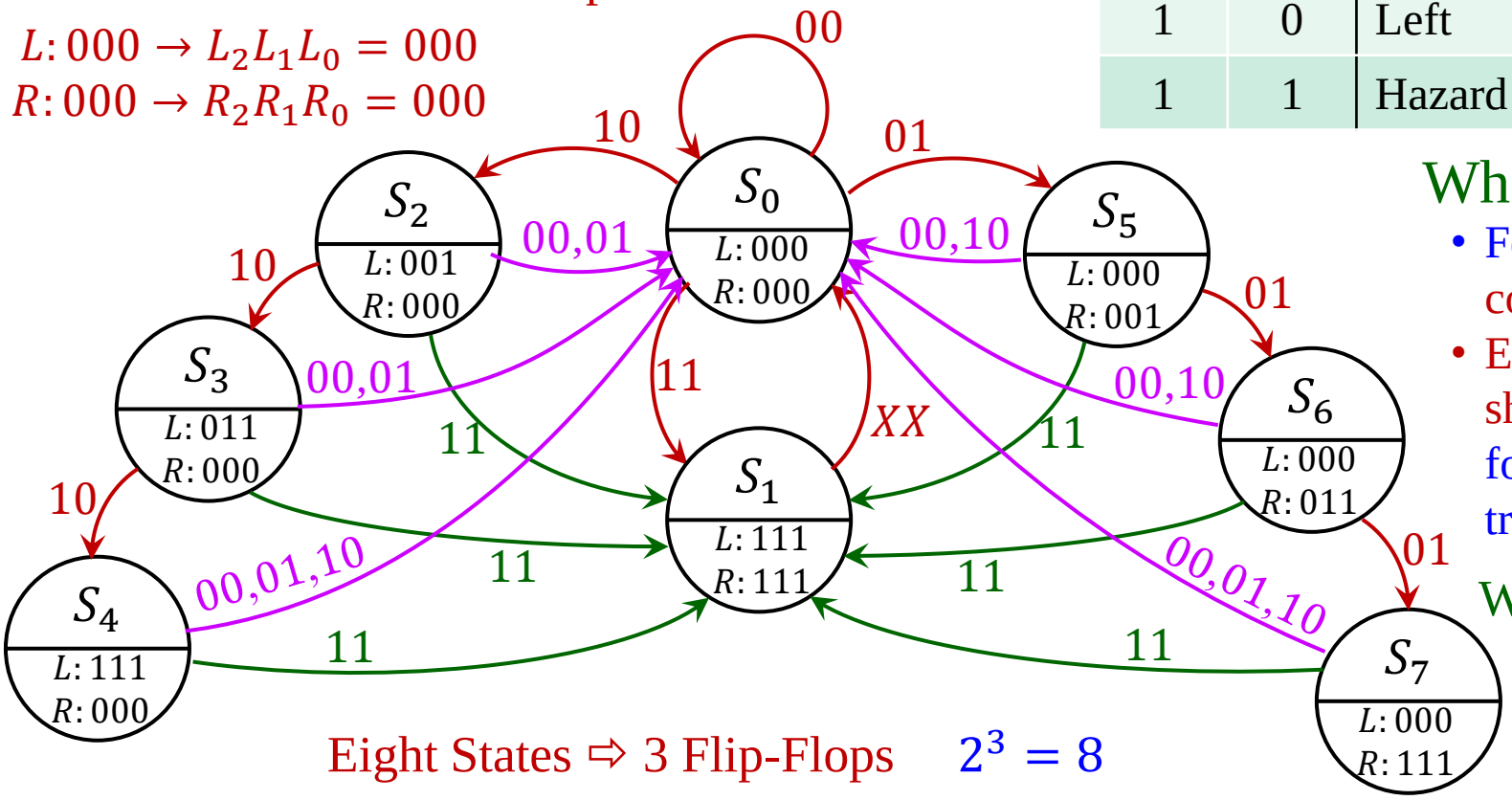
Inputs: 2		
L	*R*	
0	0	No lights
0	1	Right
1	0	Left
1	1	Hazard

Initial State: All Off

$L: 000 \rightarrow L_2L_1L_0 = 000$

$R: 000 \rightarrow R_2R_1R_0 = 000$

Inputs labeled in *LR* order



What Next?

- Four input combinations
- Each state should have four output transitions

What Next?

State Table

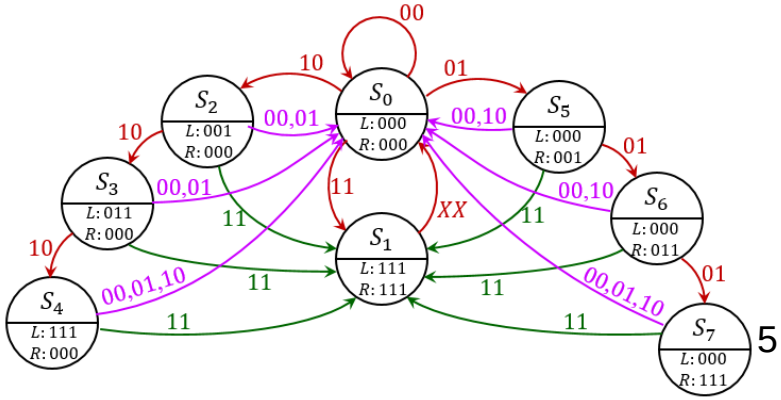
Eight States ⇨ 3 Flip-Flops $2^3 = 8$

Design Example: Mustang Sequential Turn Signals

Present State	Next State				Present Outputs	
	$LR = 00$	$LR = 01$	$LR = 11$	$LR = 10$	$L_2L_1L_0$	$R_2R_1R_0$
S_0	S_0	S_5	S_1	S_2	000	000
S_1	S_0	S_0	S_0	S_0	111	111
S_2	S_0	S_0	S_1	S_3	001	000
S_3	S_0	S_0	S_1	S_4	011	000
S_4	S_0	S_0	S_1	S_0	111	000
S_5	S_0	S_6	S_1	S_0	000	001
S_6	S_0	S_7	S_1	S_0	000	011
S_7	S_0	S_0	S_1	S_0	000	111

State Table

What Next?



Design Example: Mustang Sequential Turn Signals

ABC	$LR = 00$ $A^+B^+C^+$	$LR = 01$ $A^+B^+C^+$	$LR = 11$ $A^+B^+C^+$	$LR = 10$ $A^+B^+C^+$	Present Outputs $L_2L_1L_0$	$R_2R_1R_0$
S_0 000	000	101	001	010	000	000
S_1 001	000	000	000	000	111	111
S_2 010	000	000	001	011	001	000
S_3 011	000	000	001	100	011	000
S_4 100	000	000	001	000	111	000
S_5 101	000	110	001	000	000	001
S_6 110	000	111	001	000	000	011
S_7 111	000	000	001	000	000	111

Transition Table

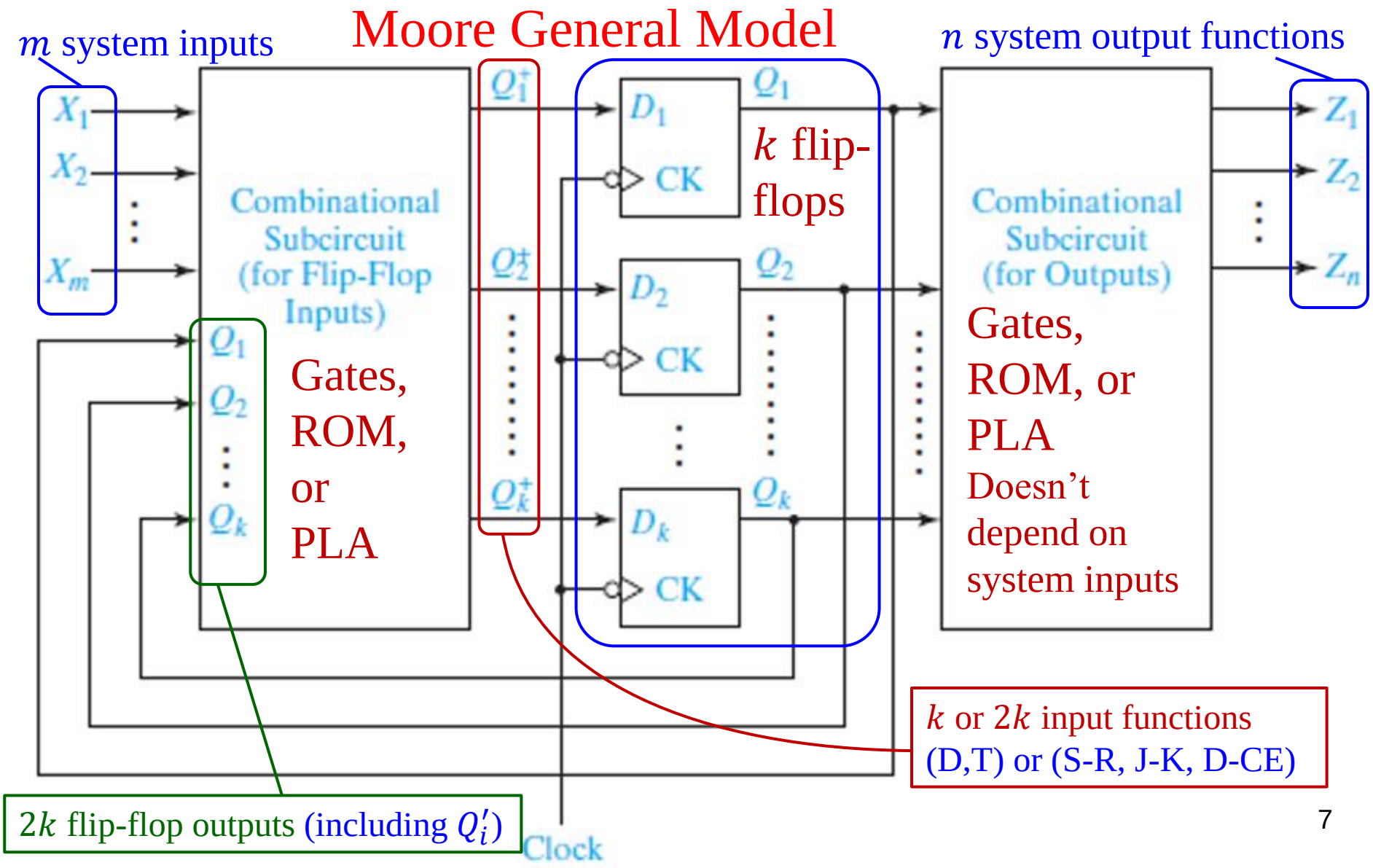
What Next?

Do you see a new challenge?

We will come back to that later. Do outputs first.

Present State	Next State $LR = 00$	$LR = 01$	$LR = 11$	$LR = 10$
S_0	S_0	S_5	S_1	S_2
S_1	S_0	S_0	S_0	S_0
S_2	S_0	S_0	S_1	S_3
S_3	S_0	S_0	S_1	S_4
S_4	S_0	S_0	S_1	S_0
S_5	S_0	S_6	S_1	S_0
S_6	S_0	S_7	S_1	S_0
S_7	S_0	S_0	S_1	S_0

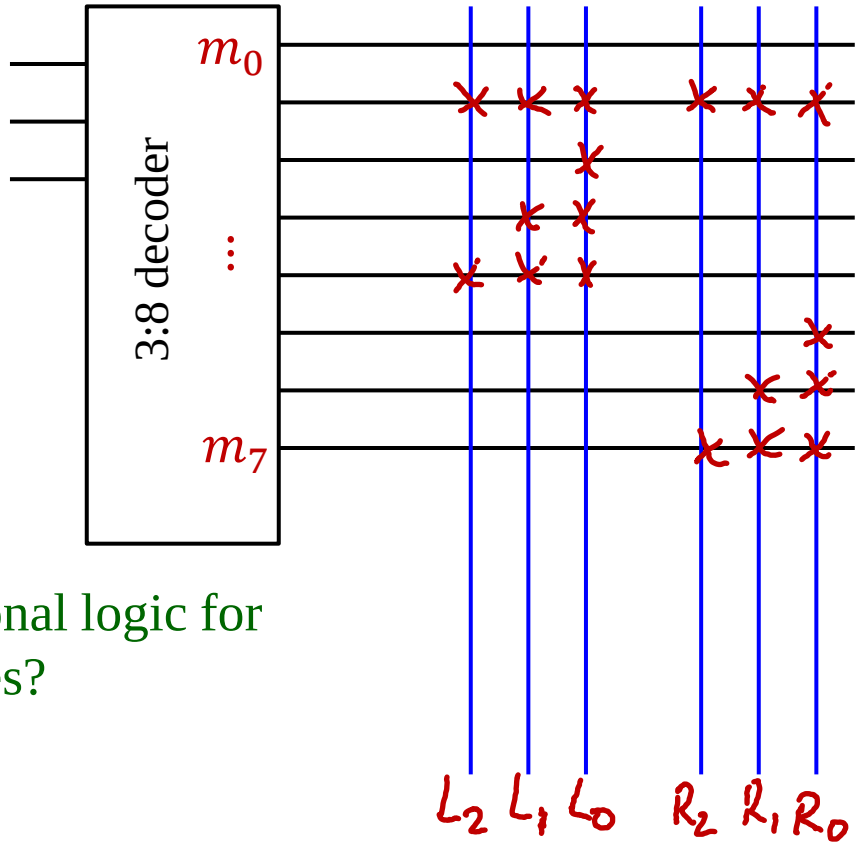
General Models for Sequential Circuits



Design Example: Mustang Sequential Turn Signals

ABC	Present Outputs $L_2L_1L_0$	$R_2R_1R_0$
S_0 000	000	000
S_1 001	111	111
S_2 010	001	000
S_3 011	011	000
S_4 100	111	000
S_5 101	000	001
S_6 110	000	011
S_7 111	000	111

Outputs: Decoder and ROM

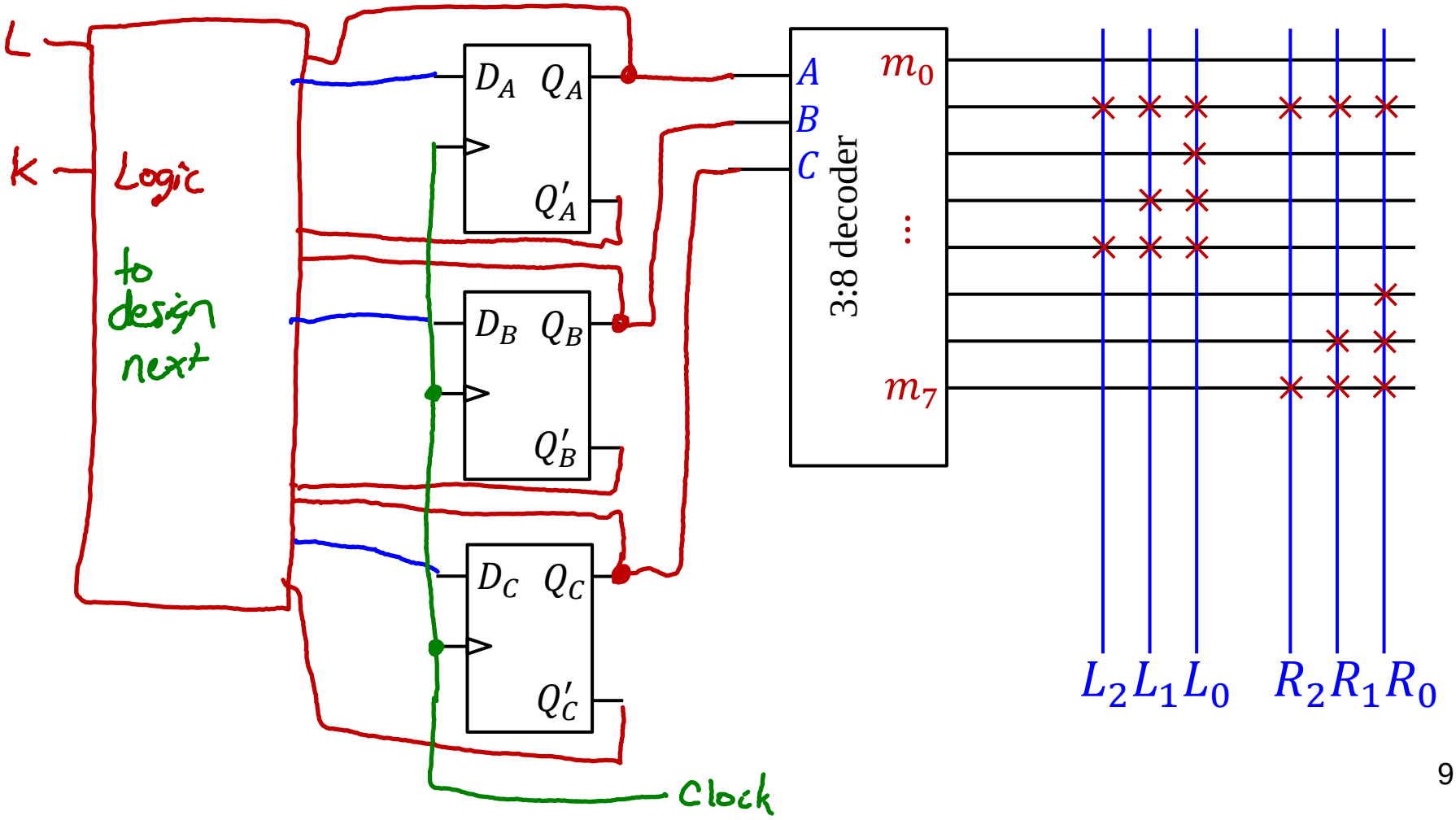


What about design of combinational logic for flip-flop inputs with five variables?

-
-

Design Example: Mustang Sequential Turn Signals

Sketch overall system



Design Example: Mustang Sequential Turn Signals

Picking A as the outlier; Demonstrating for A^+, D Flip-flops

A = 0

BC \ LR	00	01	11	10
00		1		
01				
11				1
10				

A = 1

BC \ LR	00	01	11	10
00				
01		1		
11				
10		1		

$A^+ = B'C'L'RA' + BCLR'A' + B'CL'RA + BC'L'RA$

K-maps: Try to group neighboring 1s

4-variable: Each cell can have 4 neighbors

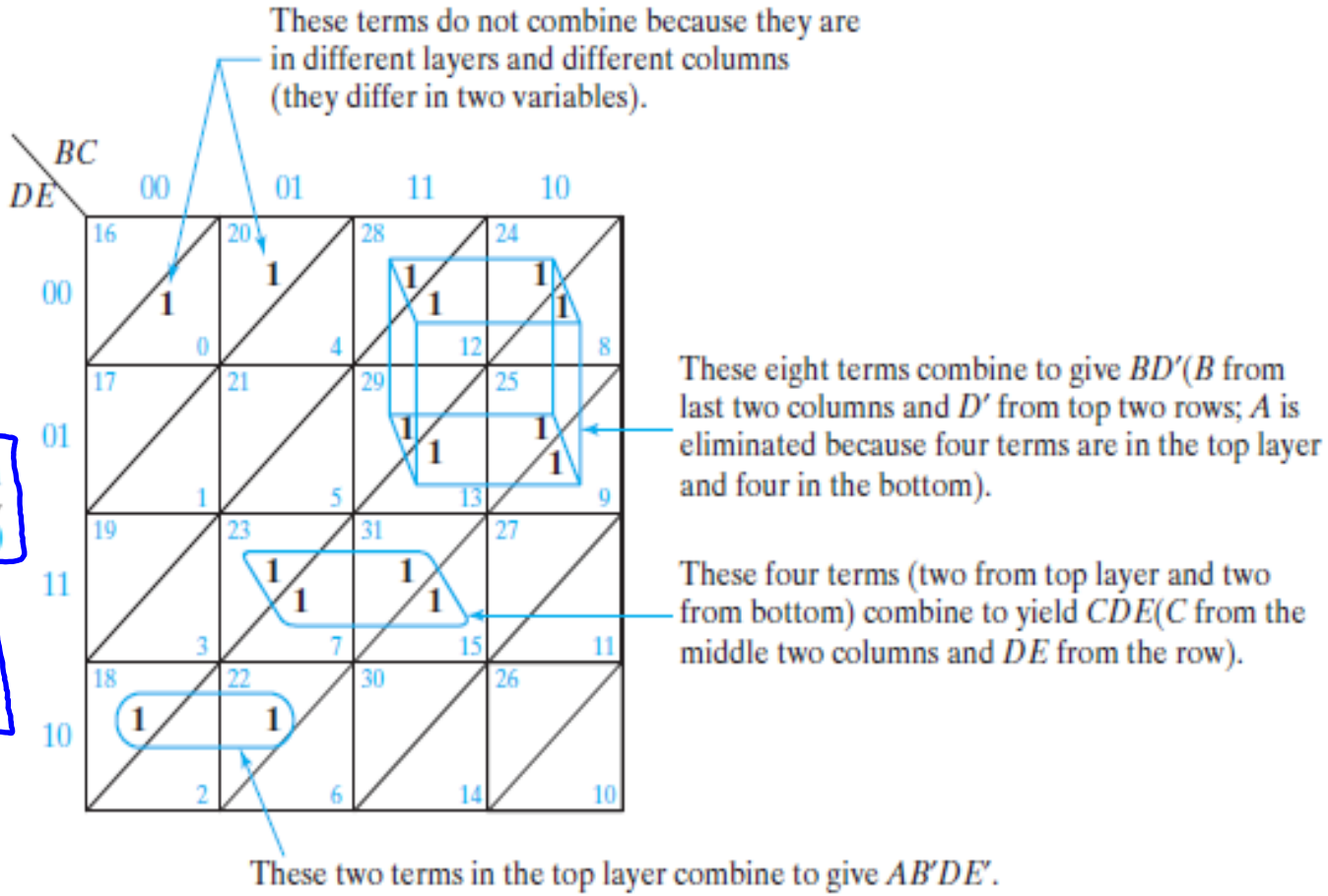
3-variable: Each cell can have 3 neighbors

5-variable: Each cell can have 5 neighbors – Layered vertically

This case: none of the 1s overlay – no further reduction

Design Example: Mustang Sequential Turn Signals

FIGURE 5-21
A Five-Variable Karnaugh Map



- Add diagonals across each cell for top and bottom layers
- Top layer:
  $A = 1$
  $m_{16} - m_{31}$
- Bottom layer:
  $A = 0$
  $m_0 - m_{15}$

Design Example: Mustang Sequential Turn Signals

ABC	$LR = 00$ $A^+B^+C^+$	$LR = 01$ $A^+B^+C^+$	$LR = 11$ $A^+B^+C^+$	$LR = 10$ $A^+B^+C^+$	Present Outputs $L_2L_1L_0$	$R_2R_1R_0$
S_0 000	000	101	001	010	000	000
S_1 001	000	000	000	000	111	111
S_2 010	000	000	001	011	001	000
S_3 011	000	000	001	100	011	000
S_4 100	000	000	001	000	111	000
S_5 101	000	110	001	000	000	001
S_6 110	000	111	001	000	000	011
S_7 111	000	000	001	000	000	111

Transition Table

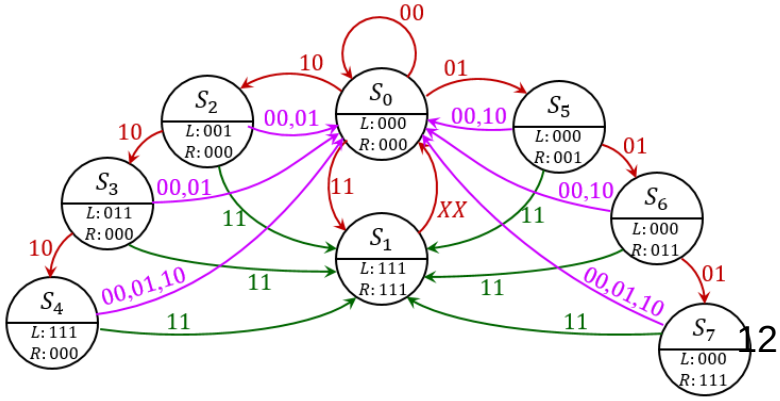
What Next?

- Next State
- Maps for each flip-flop

A^+: Already done

B^+: Four 1s

C^+: Ten 1s



Design Example: Mustang Sequential Turn Signals

B^+, D Flip Flops

A	BC \ LR	00	01	11	10
1	00				1
	01		1		
	11				
	10		1		1
0	00				
	01				
	11				
	10				

A	BC \ LR	00	01	11	10
1	00		1	1 1	
	01			1	
	11			1 1	
	10		1	1 1	1
0	00				
	01				
	11				
	10				

$$B^+ = B'CL'RA + BC'L'RA + C'LR'A'$$

$$C^+ = B'C'RA' + BC'LA' + BC'RA + LRA + BLR$$

Design Example: Mustang Sequential Turn Signals

$$A^+ = B'C'L'RA' + BCLR'A' + B'CL'RA + BC'L'RA$$
$$B^+ = B'CL'RA + BC'L'RA + C'LR'A'$$
$$C^+ = B'C'RA' + BC'LA' + BC'RA + LRA + BLR$$

A	B	C	L	R	A^+	B^+	C^+
0	0	0	0	1	1	0	0
0	1	1	1	0	1	0	0
1	0	1	0	1	1	1	0
1	1	0	0	1	1	1	0
0	-	0	1	0	0	1	0
0	0	0	-	1	0	0	1
0	1	0	1	-	0	0	1
1	1	0	-	1	0	0	1
1	-	-	1	1	0	0	1
-	1	-	1	1	0	0	1

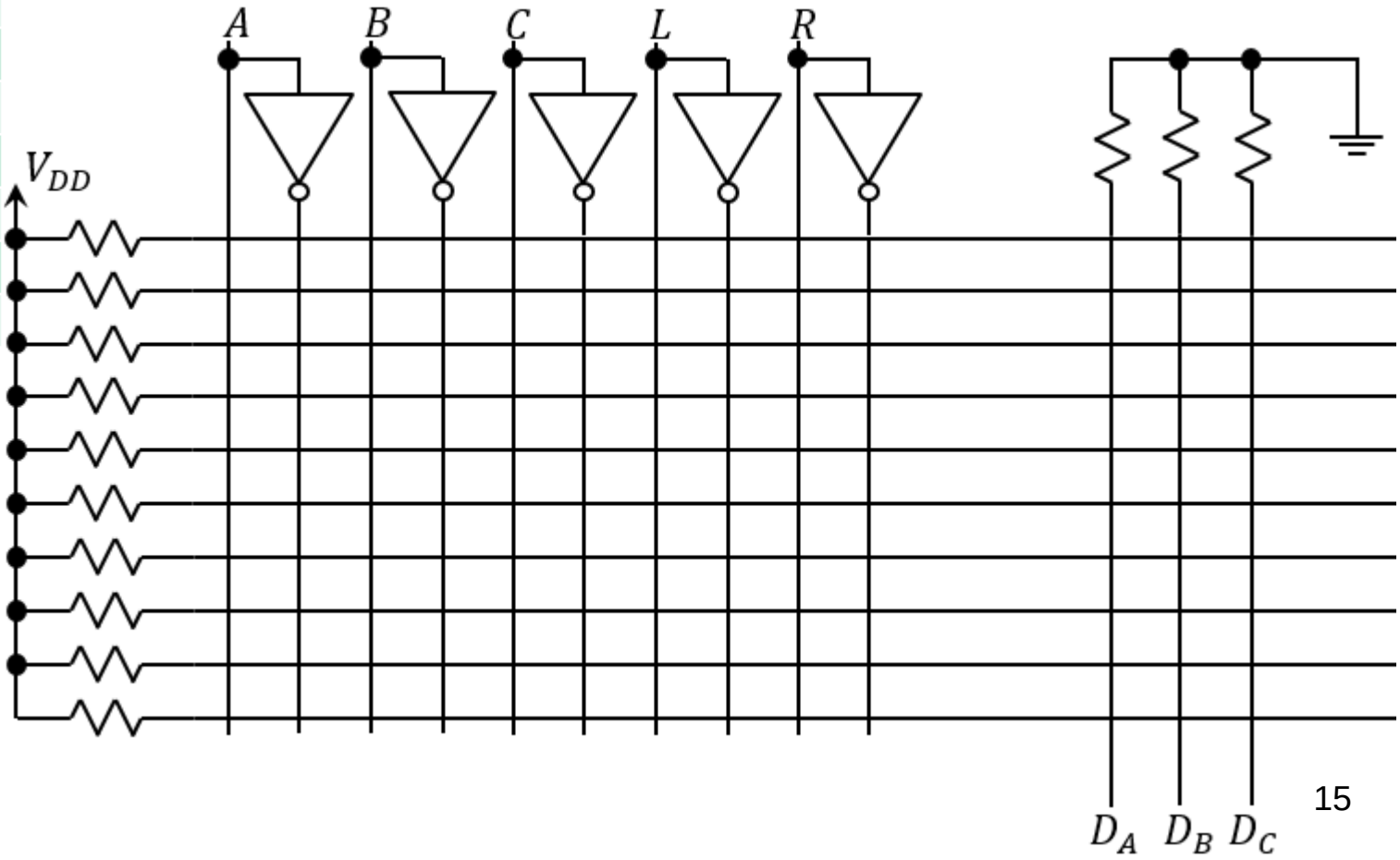
What is this table for?

Design Example: Mustang Sequential Turn Signals

A	B	C	L	R	A^+	B^+	C^+
0	0	0	0	1	1	0	0
0	1	1	1	0	1	0	0
1	0	1	0	1	1	1	0
1	1	0	0	1	1	1	0
0	-	0	1	0	0	1	0
0	0	0	-	1	0	0	1
0	1	0	1	-	0	0	1
1	1	0	-	1	0	0	1
1	-	-	1	1	0	0	1
-	1	-	1	1	0	0	1

Present state outputs from flip-flops – to PLA inputs

Inputs to the system



Mealy Design Example: Detector of 110 or 1010

- Sequence Detector Description:
  - Circuit that
    - Examines a serial string of 0's and 1's applied to the *X* input
    - Generates an output $Z = 1$ when either of two prescribed input sequences occur
  - Input *X* synchronized with clock pulses
  - For this example, the prescribed input sequences are 110 or 1010
  - Circuit will not automatically reset when a 1 output occurs
  - From our guidelines

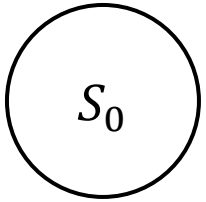
1. Start by identifying sample input and output sequences. Doing this also helps you understand the problem statement.

X: `00110100011101101010 0`

Z: `00001010000010010101 0`

Mealy Design Example: Detector of 110 or 1010

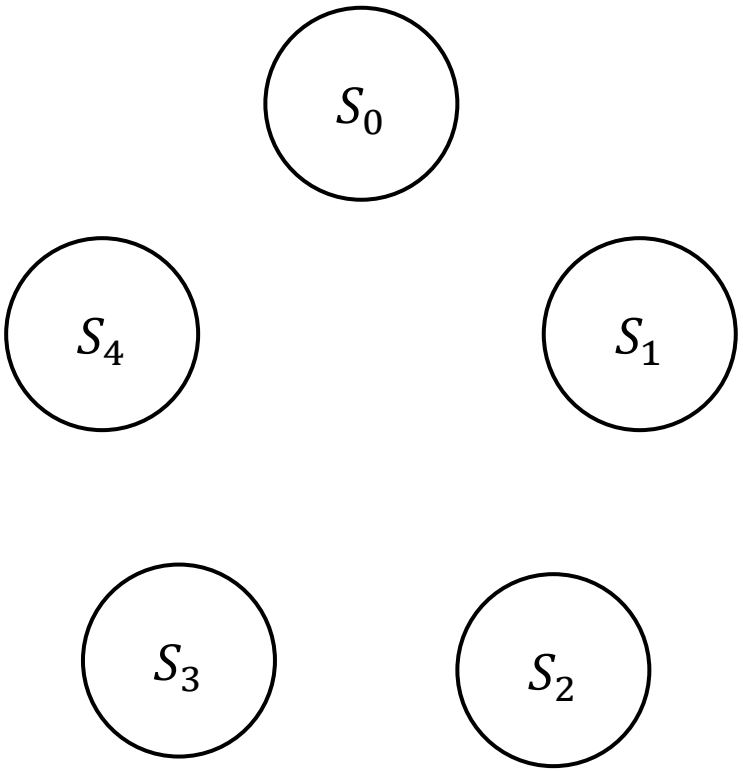
2. Determine an initial state and any condition that causes a reset to that state (if there are any)



3. If the output is mostly zero, identify the few states that cause non-zero output and start with those (partial state graph)
4. Another way to start is to determine sequences or groups of sequences that must be remembered by the circuit, and set up states for them

State	Sequence Received

Mealy Design Example: Detector of 110 or 1010



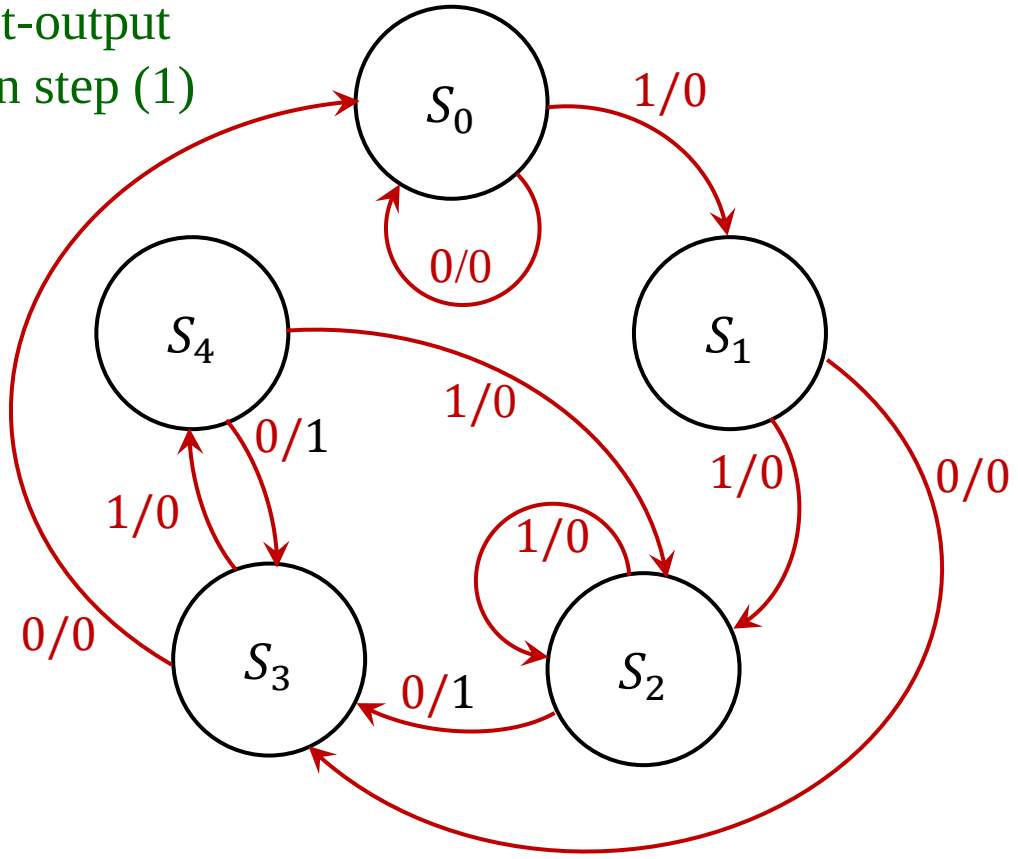
State	Sequence Received

State	Sequence Ending In

5. Can transition arrows go to existing states? Add a new state only when you really have to.
6. Once graph is complete, make sure each input combination leaves each state only once.

Mealy Design Example: Detector of 110 or 1010

7. Test graph using input-output combinations found in step (1)



Input X: 0 0 1 1 0 1 0 0 0 1 1 1 0 1 1 0 1 0 1 0 0

Output Z:

State: S_0

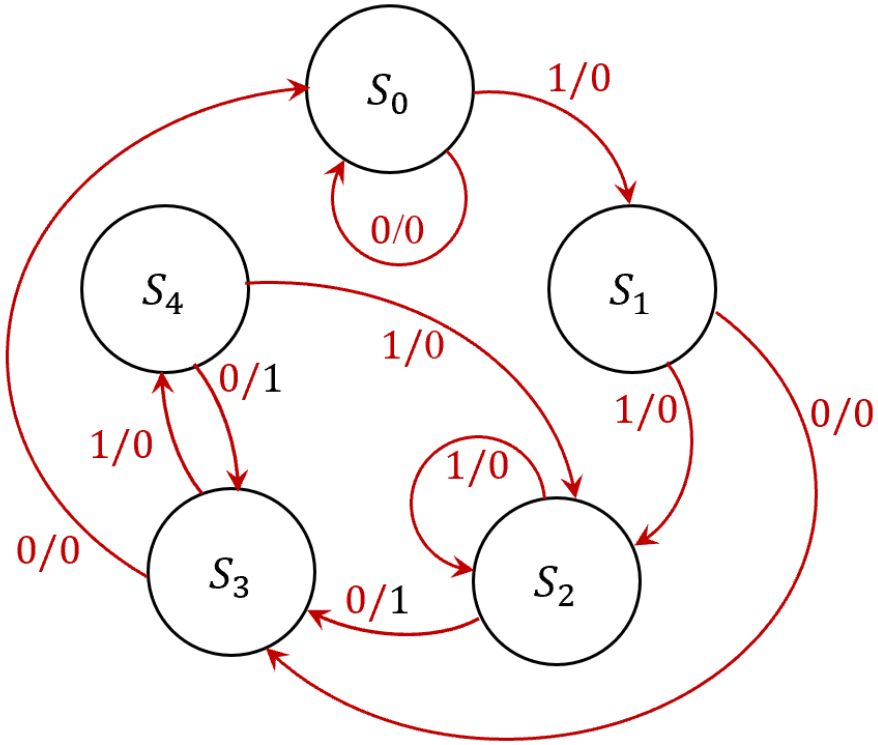
Review of Steps to Complete the Design

1. Develop a State Graph (Done)
2. Fill in a State Table
3. Fill in a Transition Table
4. Generate Flip-Flop Next State and Output Maps
5. Determine the SOP Expressions for the circuit output(s) and flip-flop input(s)
6. Design the logic circuits for the circuit output(s) and flip-flop input(s)

Mealy Design Example: Detector of 110 or 1010

2. Fill in a State Table

Present State	Next State		Present Z	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0				
S_1				
S_2				
S_3				
S_4				



Mealy Design Example: Detector of 110 or 1010

3. Fill in a Transition Table

Why did I choose these ABC codes for the states, instead of the obvious binary values of decimal subscripts?

Present State	Next State		Present Z	
	$X=0$	$X=1$	$X=0$	$X=1$
S_0	S_0	S_1	0	0
S_1	S_3	S_2	0	0
S_2	S_3	S_2	1	0
S_3	S_0	S_4	0	0
S_4	S_3	S_2	1	0

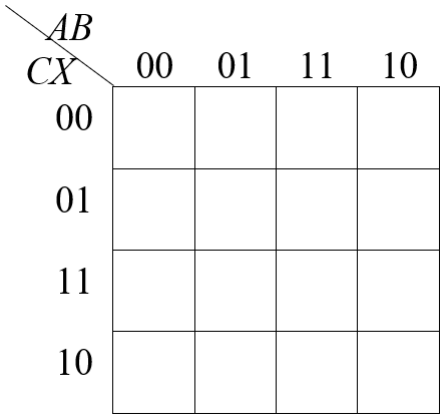
	$A^+B^+C^+$		Present Z	
ABC	$X=0$	$X=1$	$X=0$	$X=1$
100			0	0
010			0	0
001			1	0
000			0	0
011			1	0

Mealy Design Example: Detector of 110 or 1010

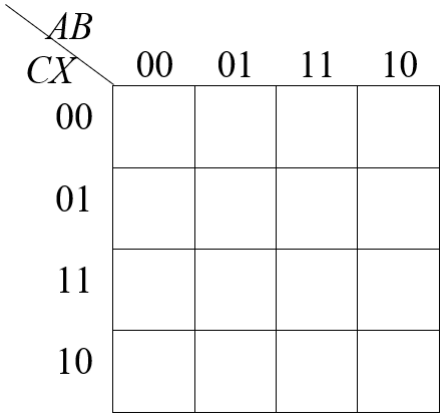
4. Generate Flip-Flop Next State and Output Maps
5. Determine SOP Expressions for circuit output(s) and flip-flop input(s)

Don't Cares for $ABCX = 101$

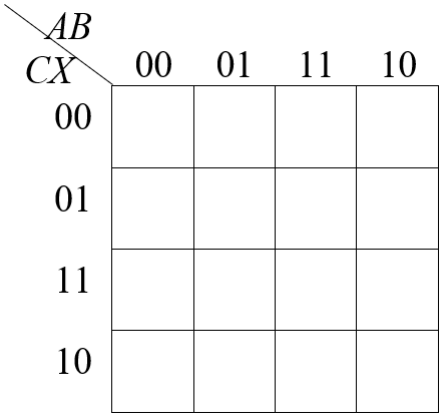
	$A^+B^+C^+$		Present Z	
ABC	$X = 0$	$X = 1$	$X = 0$	$X = 1$
100	100	010	0	0
010	000	001	0	0
001	000	001	1	0
000	100	011	0	0
011	000	001	1	0



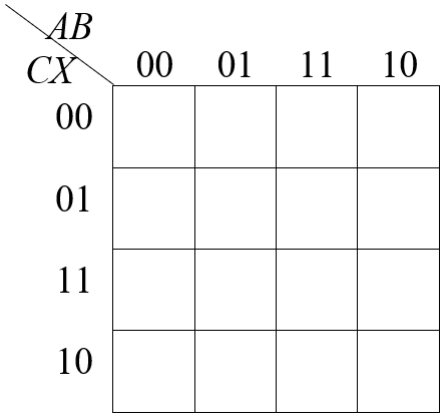
$A^+ =$



$B^+ =$



$C^+ =$



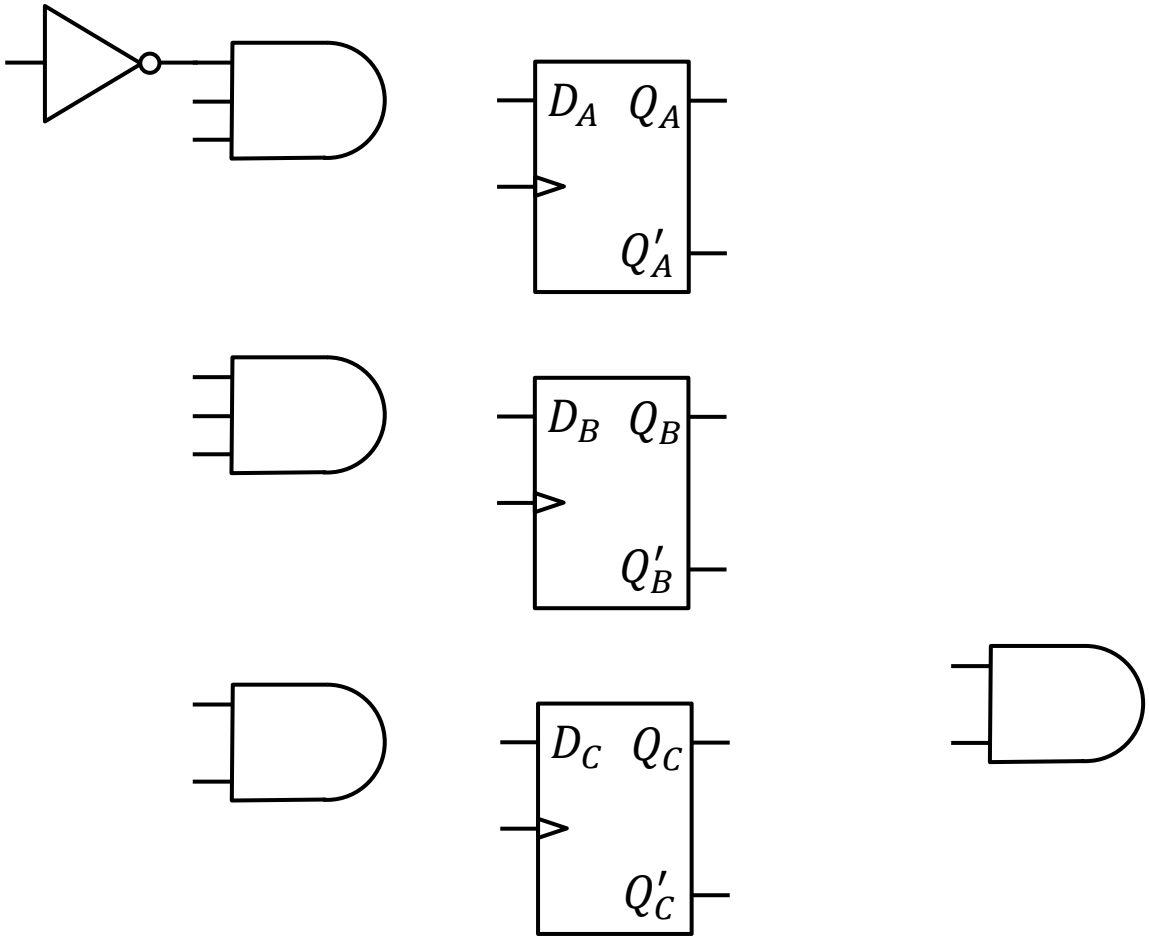
$Z =$

Note: All don't cares realized as 0 in next-state maps → extra states S_5, S_6, and S_7 all transition to $S_3 (ABC = 000)$ for $X = 0$ or $X = 1$ in full 8-state graph

Mealy Design Example: Detector of 110 or 1010

6. Design the logic circuits for circuit output(s) and flip-flop input(s)

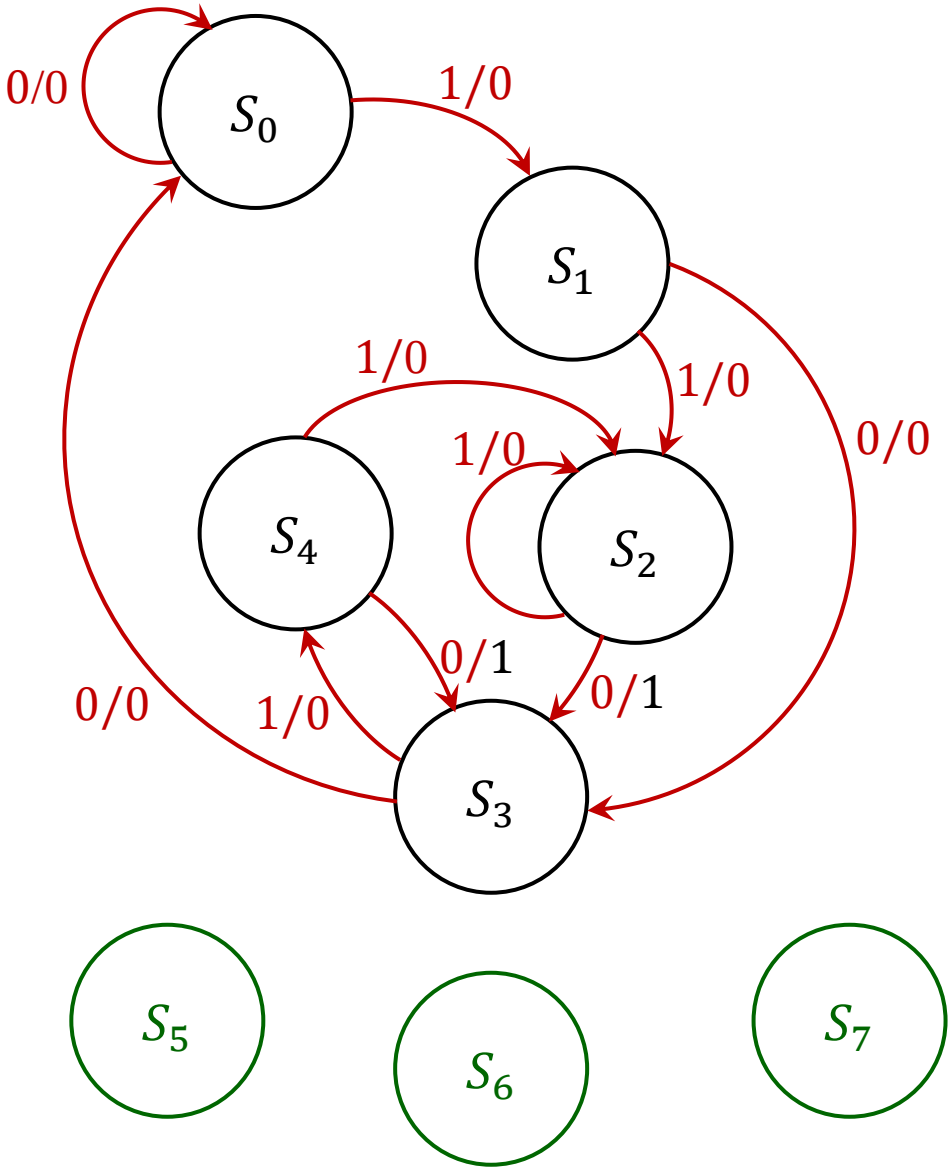
$A^+ = B'C'X'$ $B^+ = B'C'X$ $C^+ = A'X$ $Z = CX'$



Steps to Analyze a State Machine Circuit

1. Study circuits used to generate the Flip-Flop input(s) and circuit output(s). Write out their Boolean expressions (e.g. in SOP form).
2. Use the expressions to fill in Flip-Flop Next State and Output Maps
3. Use them to fill in a Transition Table
4. Fill in a State Table based on the Transition Table
5. Draw the State Graph based on the State Table

Mealy Design Example: Detector of 110 or 1010



		$A^+B^+C^+$		Present Z	
	ABC	$X=0$	$X=1$	$X=0$	$X=1$
S_0	100	1 0 0	0 1 0	0	0
S_1	010	0 0 0	0 0 1	0	0
S_2	001	0 0 0	0 0 1	1	0
S_3	000	1 0 0	0 1 1	0	0
S_4	011	0 0 0	0 0 1	1	0
S_5	101	0 0 0	0 0 0	1	0
S_6	110	0 0 0	0 0 0	0	0
S_7	111	0 0 0	0 0 0	1	0

$Z = CX'$