Lecture Outline

Reminders to self:

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

- Last Lecture
  - Finished Analysis of a Mealy machine by Transition Tables & State Graphs
  - Timing charts from State Tables & Graphs
  - General models for clocked sequential circuits
  - Started design of clocked sequential circuits
- Today's Lecture
  - Continue analysis & design of clocked sequential circuits
    - Finish Mealy 101 sequence detector design started last lecture
    - Analysis example (of Mealy sequence detector just designed)
    - State Graph design guidelines
    - Start a larger (8 state) design example (Mustang turn signals)

Handouts and Announcements

- Announcements
  - Homework Problems: HW 13-2
    - Posted on Carmen yesterday morning
    - Due: 11:59pm Thursday 3/30
  - Homework Reminder
    - HW 13-1 Due: 11:25am Wednesday 3/29
  - Read for Monday: no new reading assignment
    previous assignment pages 463-472, 149-151
  - Participation Quiz 11 available 11:15am today
    - Due 11:15am tomorrow
    - Available additional 24hr with late penalty

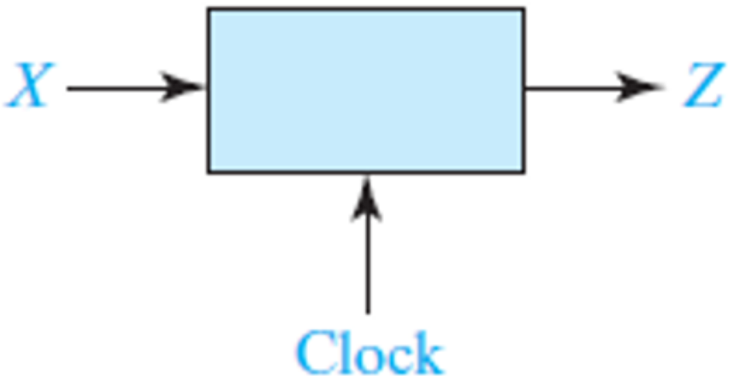
Handouts and Announcements

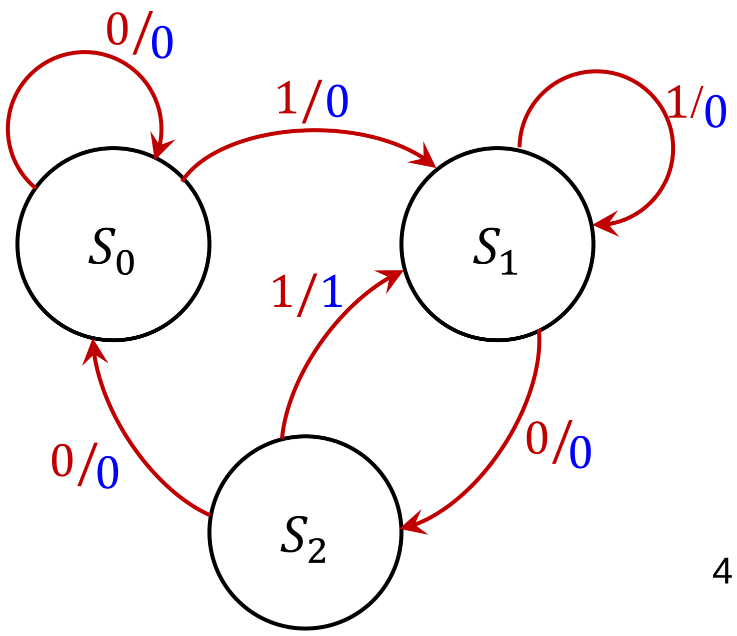
- Announcements
  - Mini-Exam 5 Reminder
    - Available 5pm Monday 3/27 through 5:00pm Tuesday 3/28
    - Due in Carmen PROMPTLY at 5:00pm on 3/28
    - Designed to be completed in ~36 min, but you may use more
    - When planning your schedule:
      - I recommend building in 10-15 min extra
      - 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

Mealy Design Example: Sequence Detector

- Sequence Detector Description:
  - Circuit that
    - Examines a serial string of 0's and 1's applied to the X input
    - Generates an output $Z = 1$ only when a prescribed input sequence occurs
  - For this example, the prescribed input sequence is 101

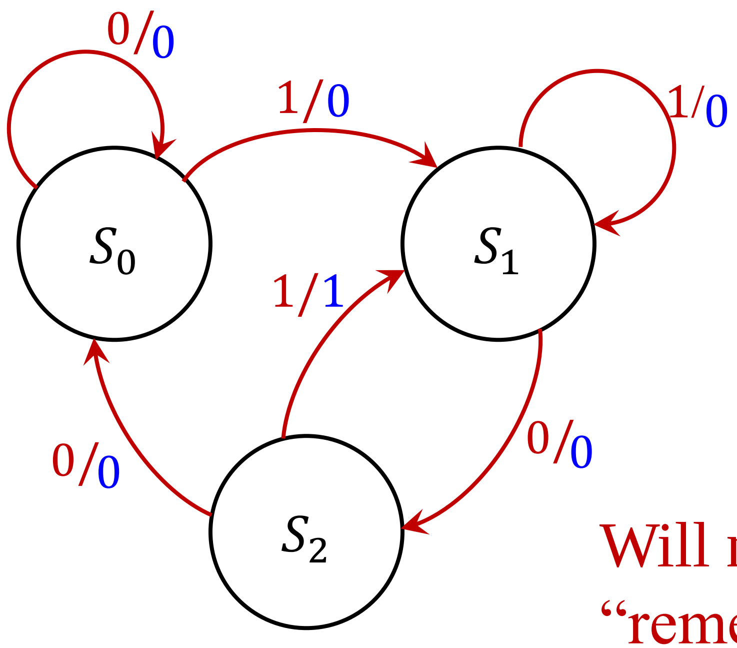
Block Diagram





Ran out of time last lecture just after converting word description into state graph →

Mealy Design Example: Sequence Detector



The State Table can be created from the State Graph

Present State	Next State		Present Output	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_0	S_1	0	0
S_1	S_2	S_1	0	0
S_2	S_0	S_1	0	1

Will need two flip-flops to "remember" three states

Remember, output is present as soon as X changes. Present before state change at active clock edge.

The Transition Table can be created from the State Table

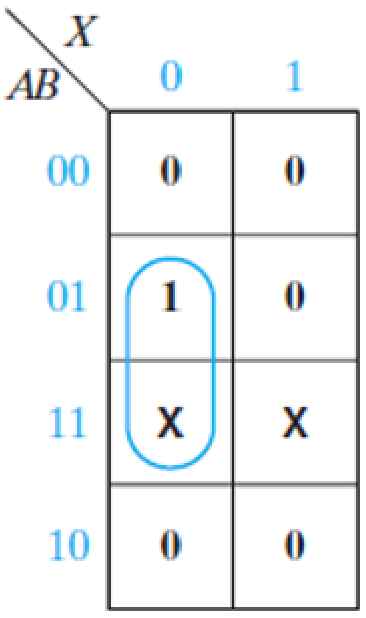
	A^+B^+		Z	
AB	$X = 0$	$X = 1$	$X = 0$	$X = 1$
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

Meally Design Example: Sequence Detector

D Flip-Flop Next-State Maps and the Output Map can be created from the Transition Table

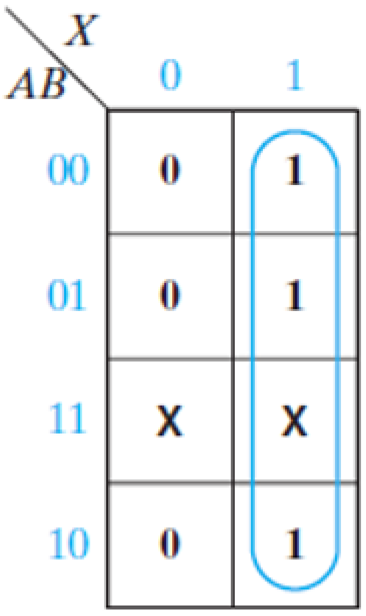
Transition Table

	A^+B^+		Z	
AB	$X = 0$	$X = 1$	$X = 0$	$X = 1$
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1



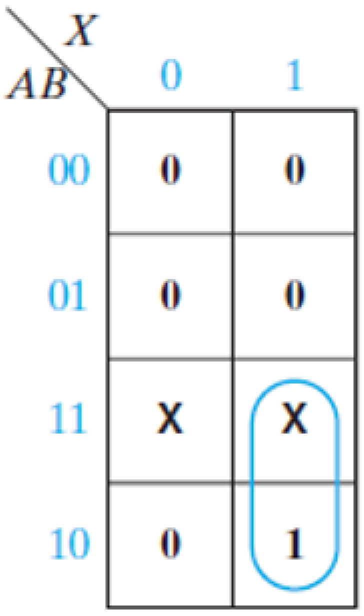
$AB \backslash X$	0	1
00	0	0
01	1	0
11	X	X
10	0	0

$A^+ = X'B$



$AB \backslash X$	0	1
00	0	1
01	0	1
11	X	X
10	0	1

$B^+ = X$

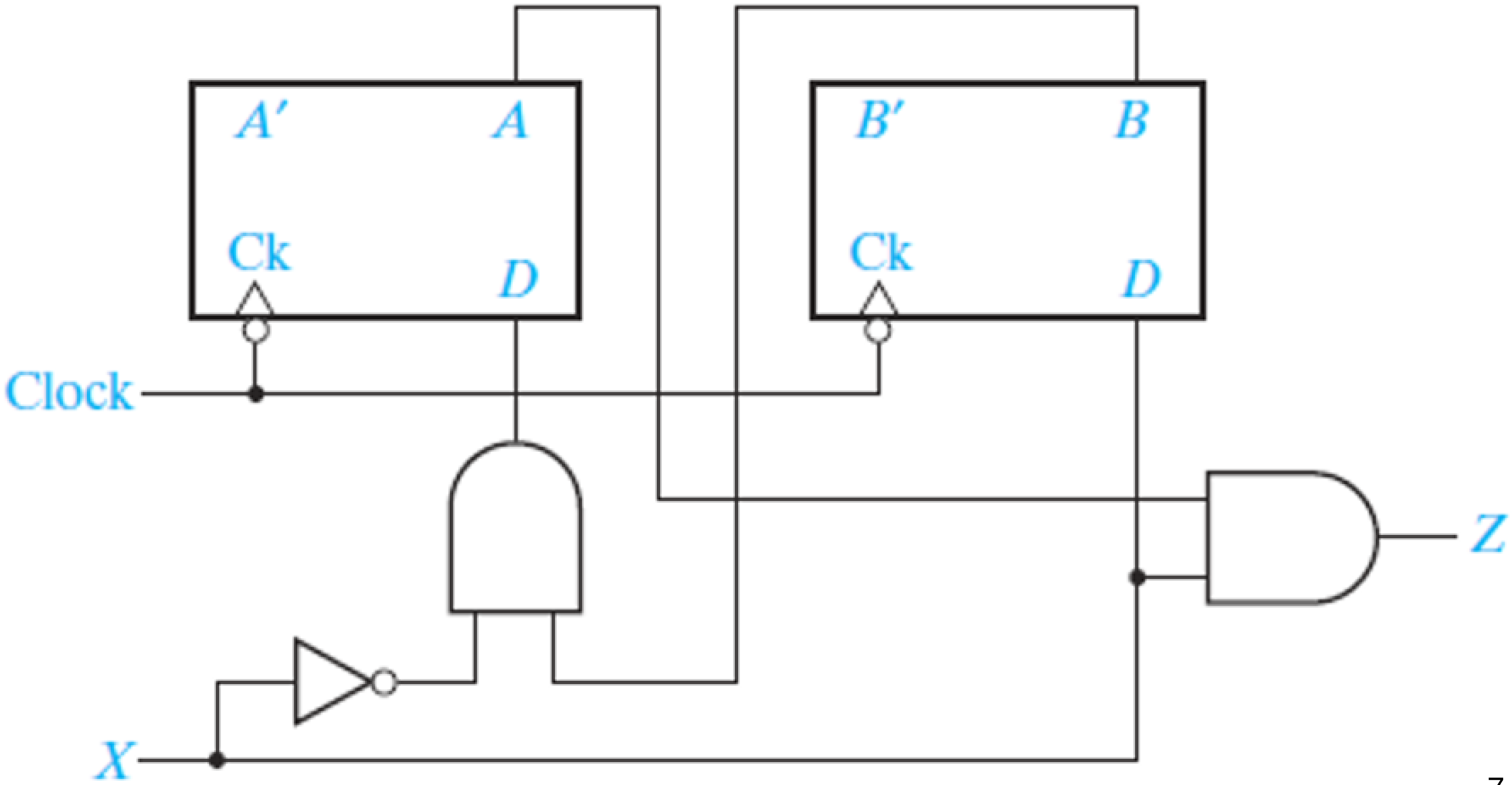


$AB \backslash X$	0	1
00	0	0
01	0	0
11	X	X
10	0	1

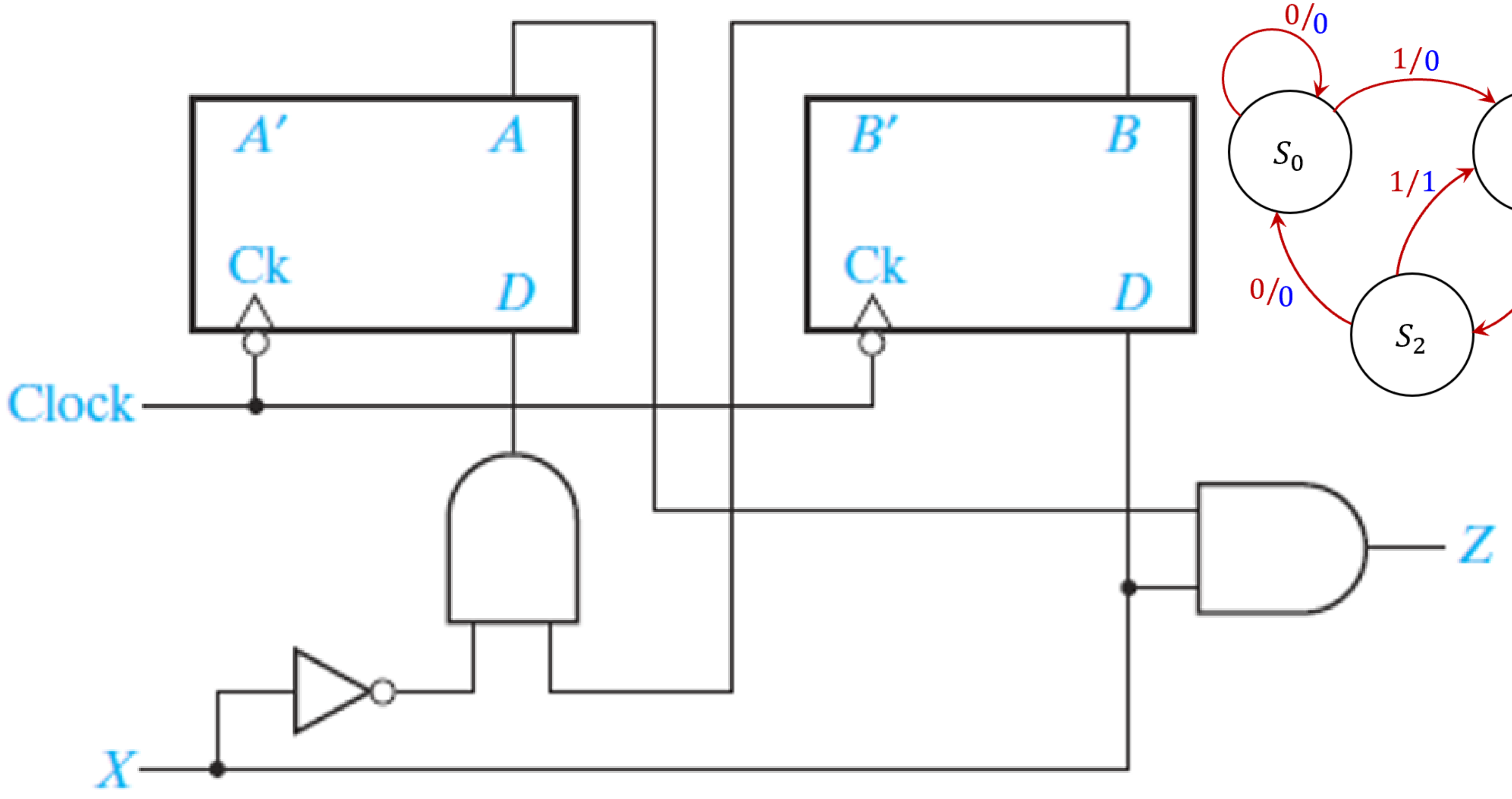
$Z = XA$

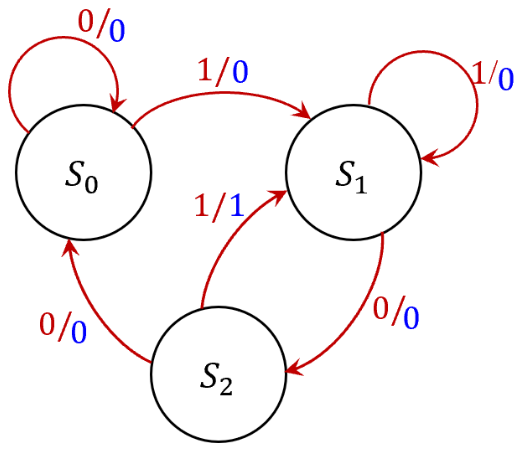
Mealy Design Example: Sequence Detector

$A^+ = X'B$ $B^+ = X$ $Z = XA$



Mealy Design Example: Sequence Detector





$A^+ = X'B$ $B^+ = X$ $Z = XA$

Sequence detector for "101" designed as a 3-state machine last lecture

- But with two flip-flops circuit realizes 4 states
- Let's analyze and compare to design goals at each step

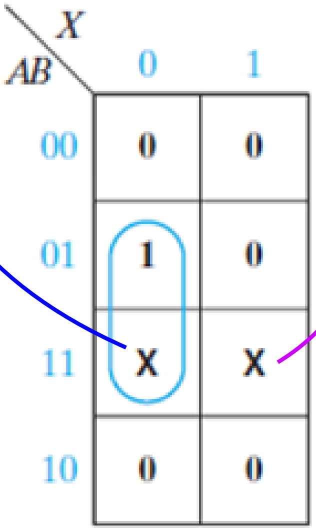
Mealy Analysis Example: Sequence Detector

$A^+ = X'B$

AB \ X	0	1
00	0	0
01	1	0
11	1	0
10	0	0

Don't Care realized as 1

Don't Care realized as 0



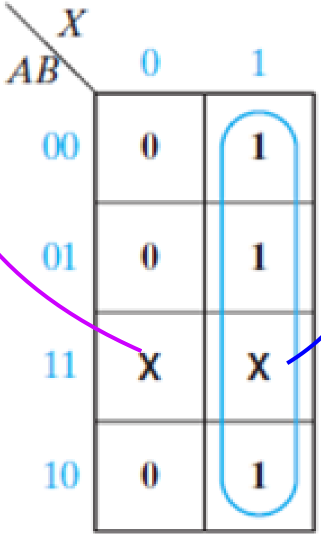
AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	0	0

$B^+ = X$

AB \ X	0	1
00	0	1
01	0	1
11	0	1
10	0	1

Don't Care realized as 0

Don't Care realized as 1



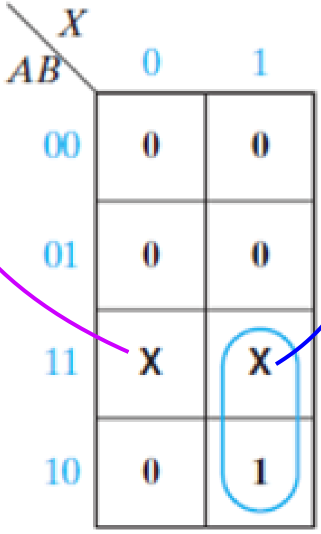
AB \ X	0	1
00	0	1
01	0	1
11	X	X
10	0	1

$Z = XA$

AB \ X	0	1
00	0	0
01	0	0
11	0	1
10	0	1

Don't Care realized as 0

Don't Care realized as 1



AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	0	1

Mealy Analysis Example: Sequence Detector

$A^+ = X'B$

$AB \backslash X$	0	1
00	0	0
01	1	0
11	1	0
10	0	0

$B^+ = X$

$AB \backslash X$	0	1
00	0	1
01	0	1
11	0	1
10	0	1

$Z = XA$

$AB \backslash X$	0	1
00	0	0
01	0	0
11	0	1
10	0	1

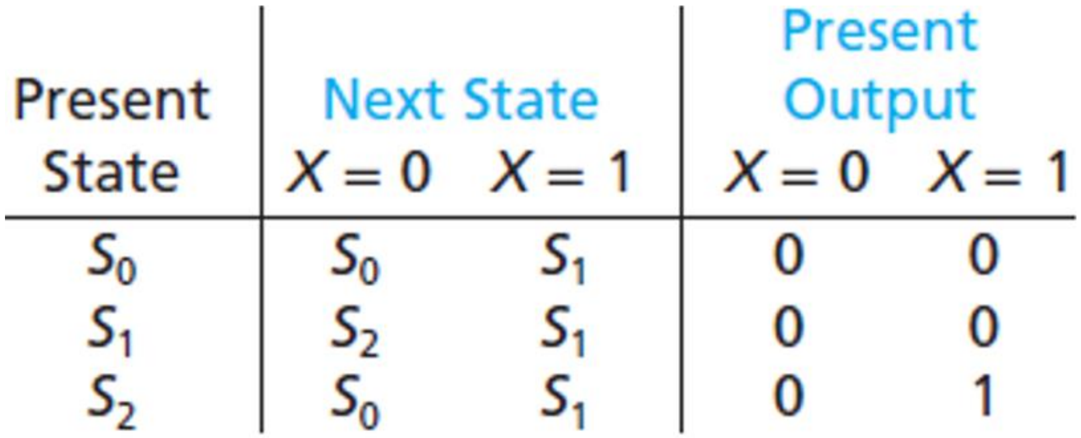
	A^+B^+		**Present Z**	
AB	$X = 0$	$X = 1$	$X = 0$	$X = 1$
00	00	01	0	0
01	10	01	0	0
11	10	01	0	1
10	00	01	0	1

	A^+B^+		Z	
AB	$X = 0$	$X = 1$	$X = 0$	$X = 1$
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1
11	xx	xx	x	x

Mealy Analysis Example: Sequence Detector

	A^+B^+		Present Z	
AB	$X = 0$	$X = 1$	$X = 0$	$X = 1$
00	00	01	0	0
01	10	01	0	0
11	10	01	0	1
10	00	01	0	1

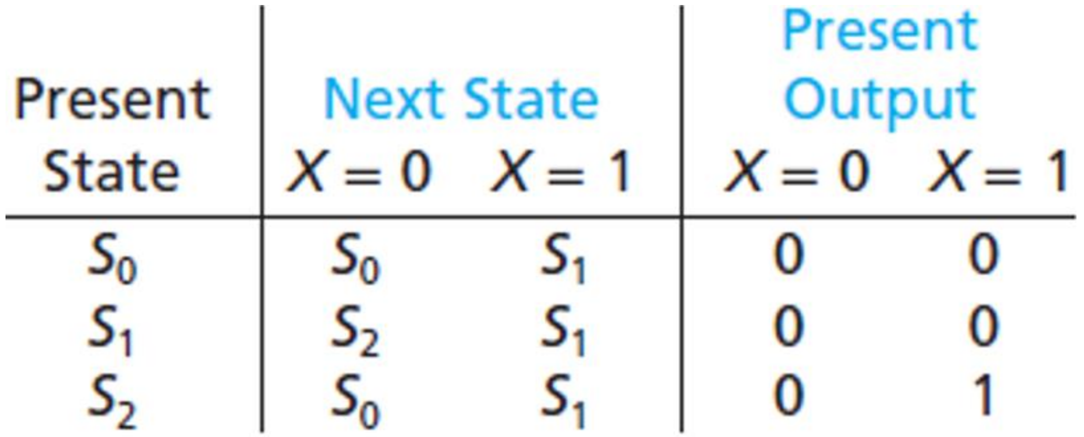
	A^+B^+		Present Z	
AB	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_0	S_1	0	0
S_1	S_2	S_1	0	0
S_3	S_2	S_1	0	1
S_2	S_0	S_1	0	1



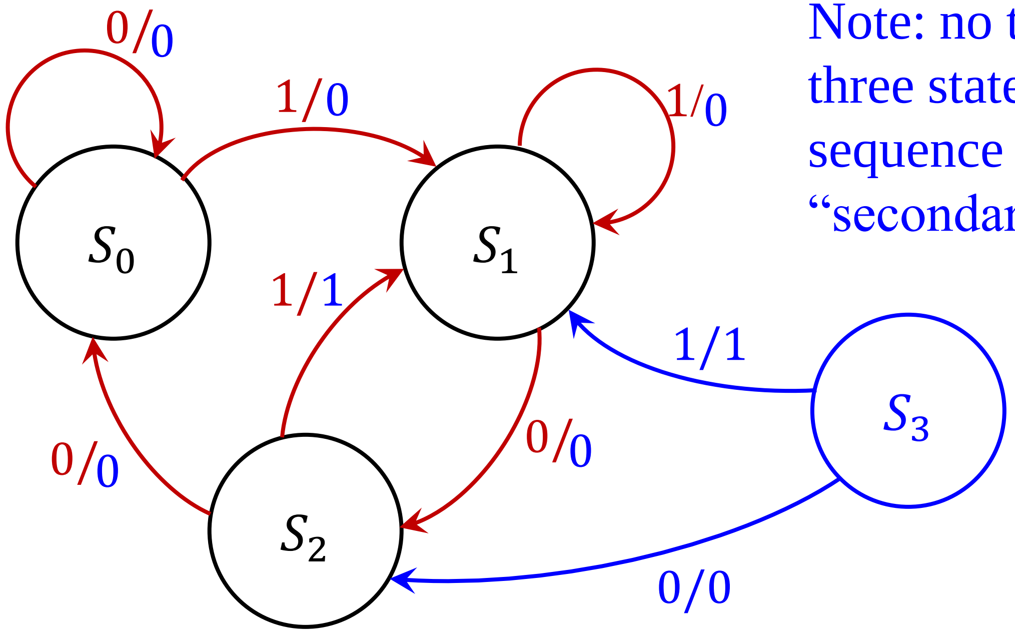
Present State	Next State		Present Output	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_0	S_1	0	0
S_1	S_2	S_1	0	0
S_2	S_0	S_1	0	1

Mealy Analysis Example: Sequence Detector

	A^+B^+		Present Z	
AB	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_0	S_1	0	0
S_1	S_2	S_1	0	0
S_3	S_2	S_1	0	1
S_2	S_0	S_1	0	1



Present State	Next State		Present Output	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_0	S_1	0	0
S_1	S_2	S_1	0	0
S_2	S_0	S_1	0	1



Note: no transitions from three states needed for sequence detector into "secondary" state S_3

State Graph Guidelines

Note that these are guidelines (helpful steps, not “The Law”)

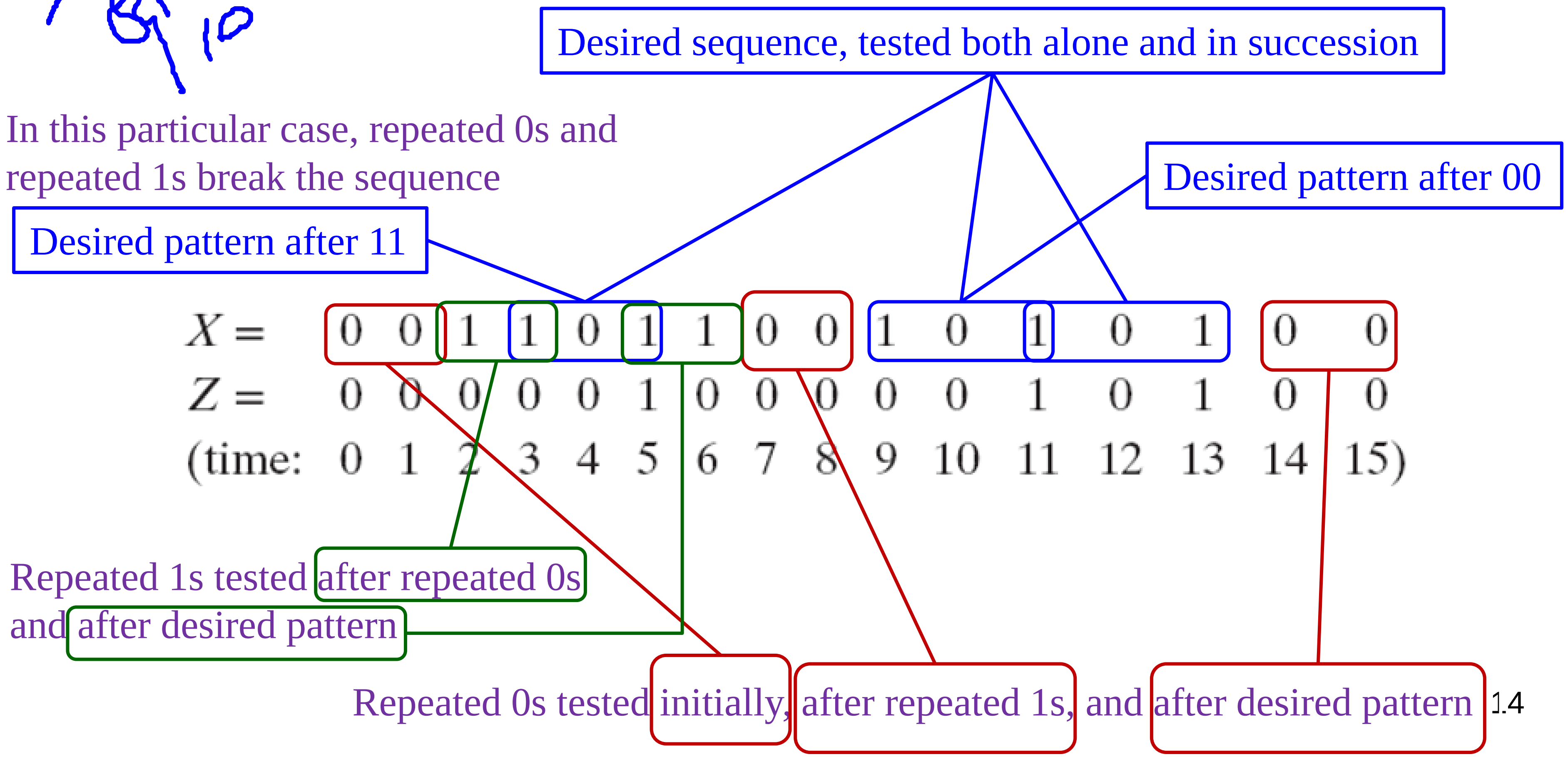
1. Start by identifying sample input and output sequences. Doing this also helps you understand the problem statement.
2. Determine an initial state and any condition that causes a reset to that state (if there are any)
3. If the output is mostly zero, identify the few states that cause non-zero output 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
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
7. Test graph using input-output combinations found in step (1)

State Graph Guidelines
101sequence detector

Note that these are guidelines (helpful steps, not “The Law”)

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

7 8 9 10



Desired sequence, tested both alone and in succession

In this particular case, repeated 0s and repeated 1s break the sequence

Desired pattern after 00

Desired pattern after 11

$X =$	0	0	1	1	0	1	1	0	0	1	0	1	0	1	0	0
$Z =$	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0
(time:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15)

Repeated 1s tested after repeated 0s and after desired pattern

Repeated 0s tested initially, after repeated 1s, and after desired pattern

State Graph Guidelines
101sequence detector

Note that these are guidelines (helpful steps, not "The Law")

1. Start by identifying sample input and output sequences. Doing this also helps you understand the problem statement.
2. Determine an initial state and any condition that causes a reset to that state (if there are any)

From the problem statement "Circuit will not reset when a 1 output occurs"

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

The sequence detector outputs 1 in only a single situation, but…

Also must remember sequence of serial inputs received in 101 pattern

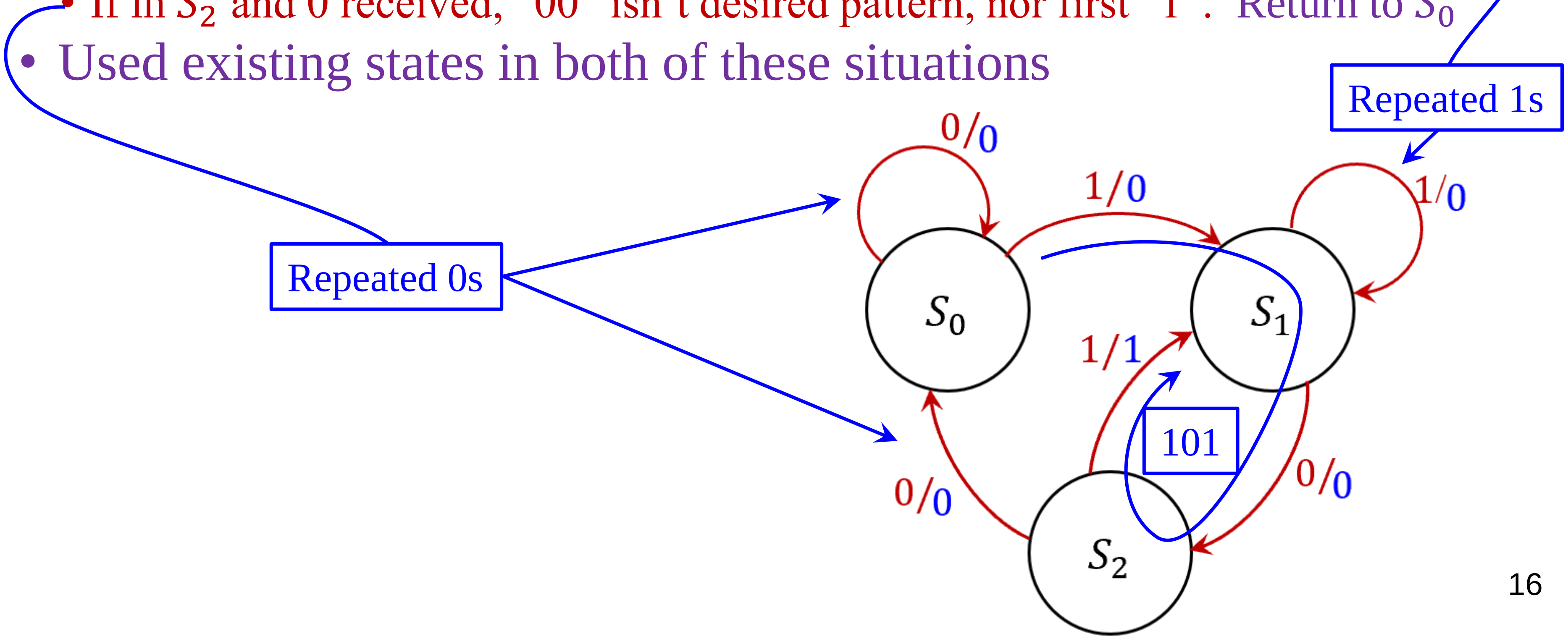
We used approach from statement 4 last lecture

State Graph Guidelines
101sequence detector

Note that these are guidelines (helpful steps, not "The Law")

5. Can transition arrows go to existing states? Add a new state only when you really have to.

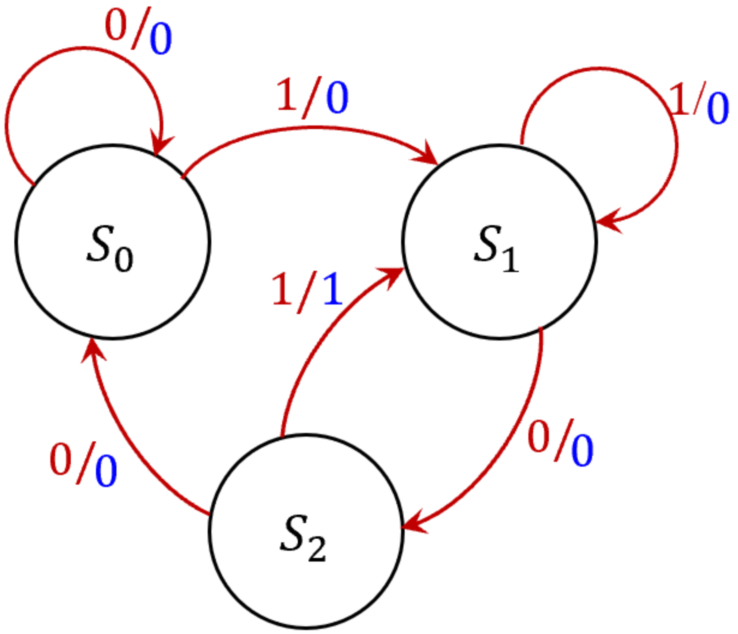
- Partway through development of state graph
  - If in S_1 and 1 received, still have only first "1" bit in "101" pattern. Stay in S_1
  - If in S_2 and 0 received, "00" isn't desired pattern, nor first "1". Return to S_0
- Used existing states in both of these situations



State Graph Guidelines
101sequence detector

Note that these are guidelines (helpful steps, not “The Law”)

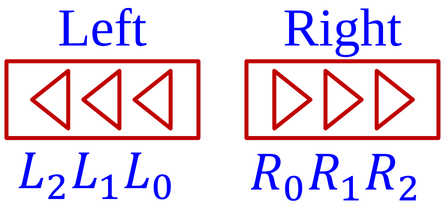
6. Once graph is complete, make sure each input combination leaves each state only once.
7. Test graph using input-output combinations found in step (1)



$X =$	0	0	1	1	0	1	1	0	0	1	0	1	0	1	0	0
$Z =$	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0
(time:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15)

Design Example: Mustang Sequential Turn Signals

- Left and right sequential signals
- Hazard lights



$$\begin{tikzcd}\text{When turning left:} & L_0 \arrow[r] & L_0L_1 \arrow[r] & L_0L_1L_2 \arrow[r] & \text{all off} \arrow[lll, bend left]\end{tikzcd}$$

$$\begin{tikzcd}\text{When turning right:} & R_0 \arrow[r] & R_1R_0 \arrow[r] & R_2R_1R_0 \arrow[r] & \text{all off} \arrow[lll, bend left]\end{tikzcd}$$

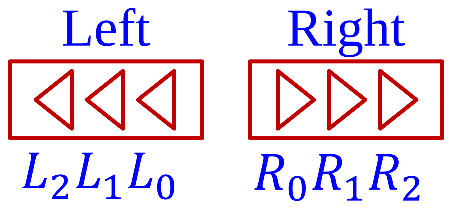
$$\begin{tikzcd}\text{Hazards on:} & L_2L_1L_0R_2R_1R_0 \arrow[r] & \text{all off} \arrow[l, bend left]\end{tikzcd}$$

$\Uparrow$

Highest priority: Leave any state → Hazard State Directly

https://www.americanmuscle.com/raxiom-sequential-harness-1014.html?utm_content=Lights%20-%20Sequential%7CRaxiom&T5_Var4=301042&utm_source=google-pla&utm_medium=shopping&T5_Var2=shopping&utm_campaign=AMM+Mustang+Vehicle+Medium&T5_Var3=blue&intl=0&dialogtech=ppc&gclid=EAIaIQobChMIuJbNosj-5QIVTR-tBh2BnQuTEAQYBSABEgIsTfD_BwE&gclsrc=aw.ds#product_pics-0

Design Example: Mustang Sequential Turn Signals



← The six outputs

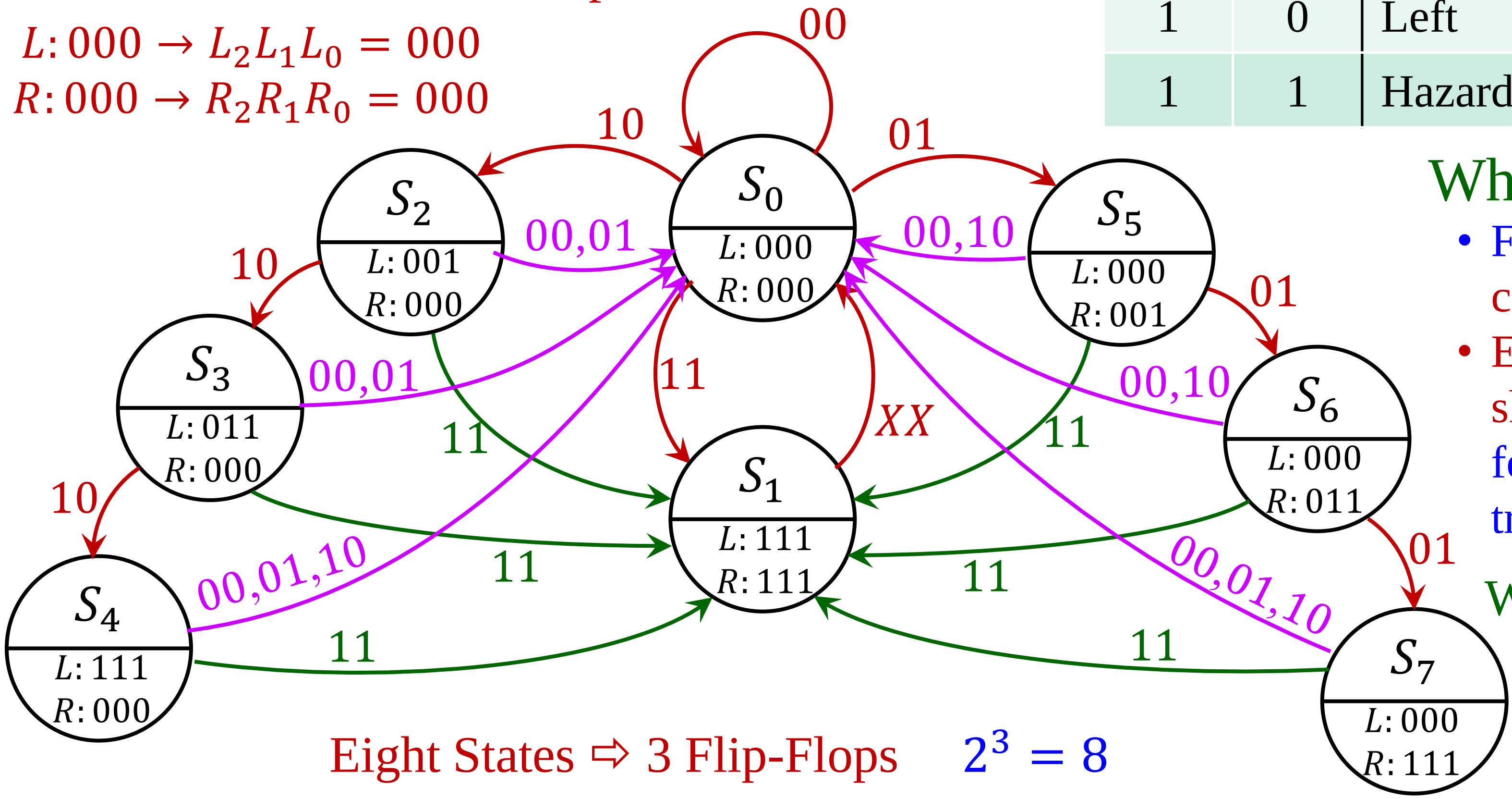
Inputs: 2		
L	***R***	
0	0	No lights
0	1	Right
1	0	Left
1	1	Hazard

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 $2^3 = 8$

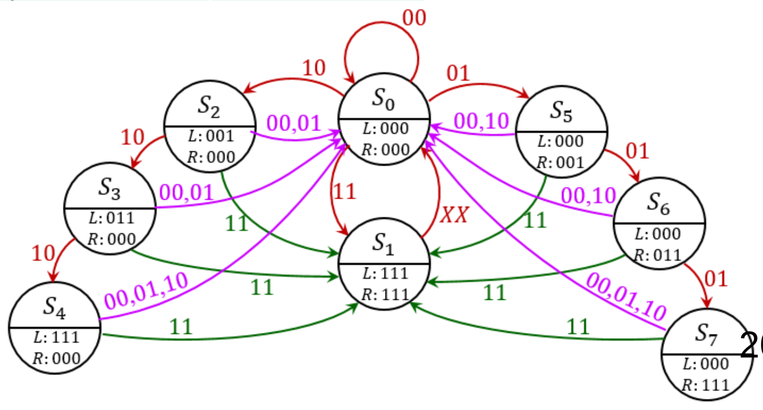
Design Example: Mustang Sequential Turn Signals

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?

Transition Table



THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

ECE2060

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

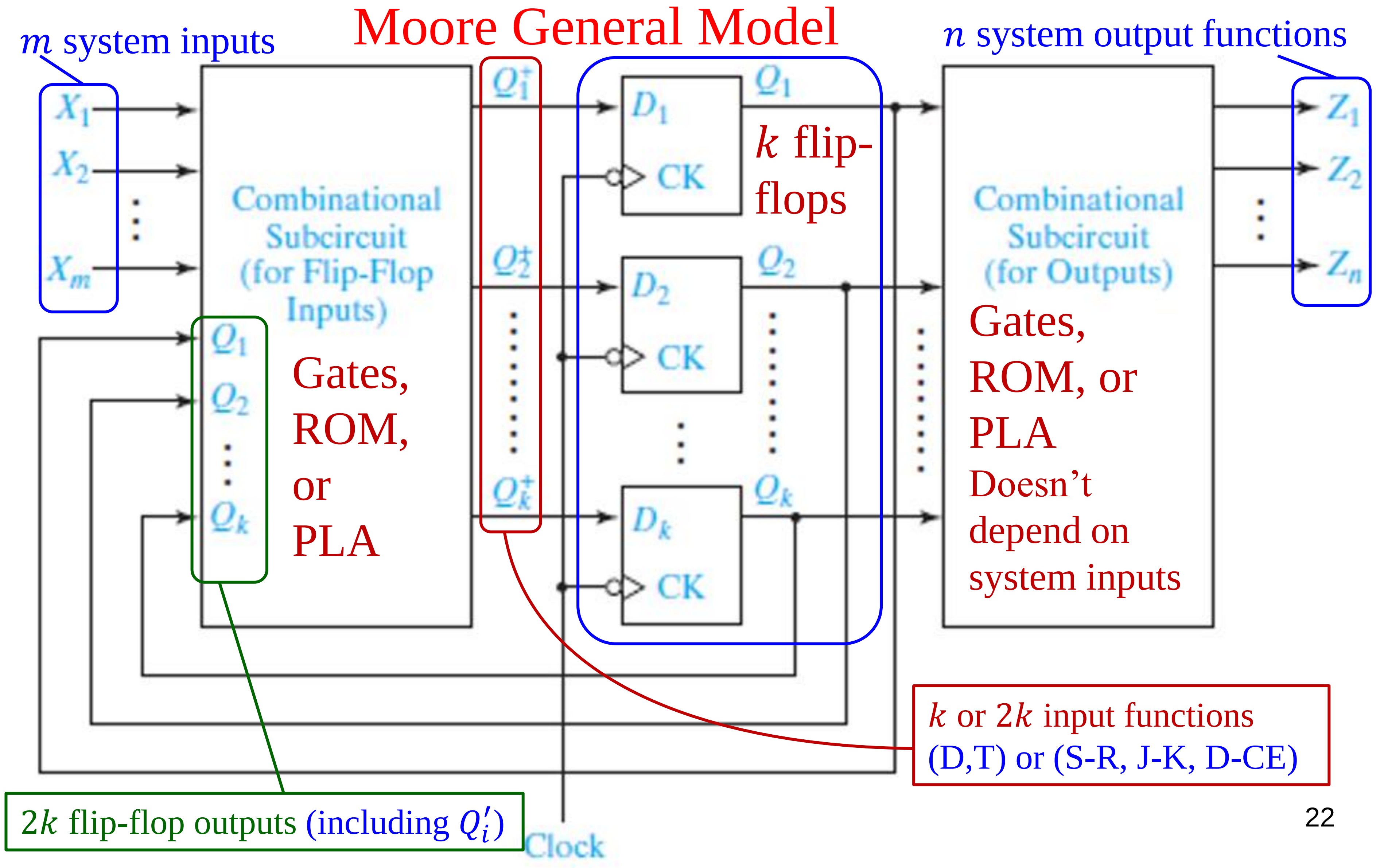
Present State	Next State $LR = 00$	$LR = 01$	$LR = 11$	$LR = 10$
S_0	S_0	S_5	S_1	S_2
S_1	S_0	S_0	S_0	S_0
S_2	S_0	S_0	S_1	S_3
S_3	S_0	S_0	S_1	S_4
S_4	S_0	S_0	S_1	S_0
S_5	S_0	S_6	S_1	S_0
S_6	S_0	S_7	S_1	S_0
S_7	S_0	S_0	S_1	S_0

Do you see a new challenge?

Five variables: A, B, C, L, R

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

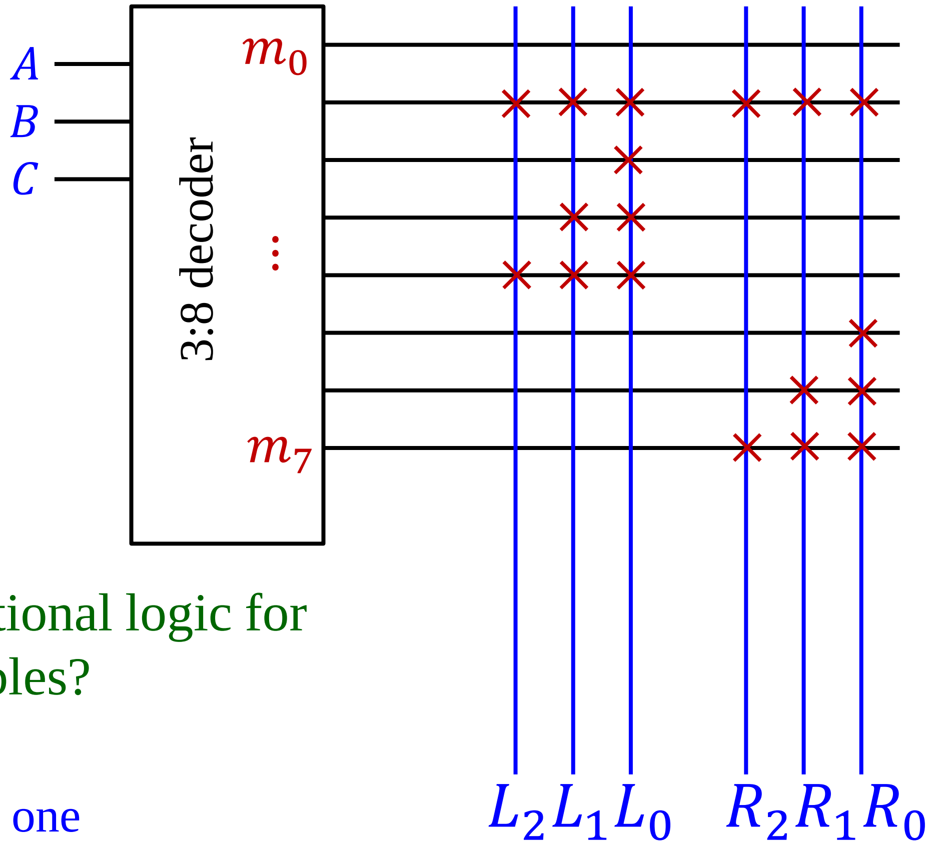
General Models for Sequential Circuits


Moore General Model
m system inputs
X_1
X_2
X_m
Combinational Subcircuit (for Flip-Flop Inputs)
Gates, ROM, or PLA
Q_1
Q_2
Q_k
Q_1^+
Q_2^+
Q_k^+
D_1
D_2
D_k
CK
k flip-flops
Combinational Subcircuit (for Outputs)
Gates, ROM, or PLA
Doesn't depend on system inputs
n system output functions
Z_1
Z_2
Z_n
Clock
k or $2k$ input functions
(D,T) or (S-R, J-K, D-CE)
$2k$ flip-flop outputs (including Q_i')

Design Example: Mustang Sequential Turn Signals

	Present Outputs	
ABC	$L_2L_1L_0$	$R_2R_1R_0$
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

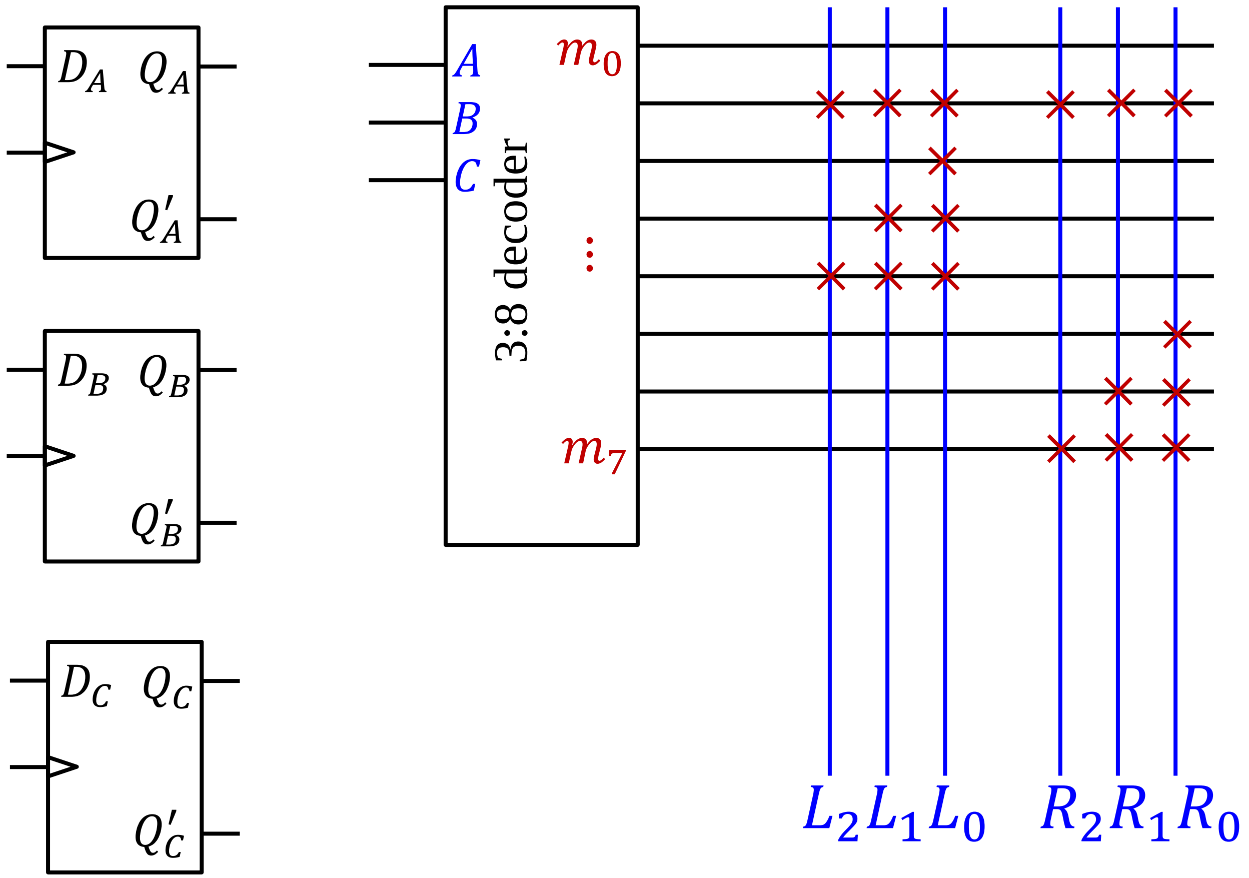


What about design of combinational logic for flip-flop inputs with five variables?

- Pick one variable as "outlier"
- Make two 4-variable K-maps, one for each of the "outlier's" values

Design Example: Mustang Sequential Turn Signals

Sketch overall system



THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

ECE2060

Design Example: Mustang Sequential Turn Signals

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

$A = 0$

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

$A = 1$

$BC \backslash LR$	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 – think layers, top & bottom

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

Design Example: Mustang Sequential Turn Signals

The design for B^+ and C^+, D Flip-flops will be completed next lecture

Before wrapping up for the day:
- Remember to complete Participation Quiz 11
- Available from 11:15am today through 11:15am 3/26
- Due at 11:15am tomorrow (late penalty starts then)