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 the larger (8 state) design example (Mustang turn signals)
    - Five-variable K-Maps
- Today’s Lecture
  - Continue analysis & design of clocked sequential circuits
    - A larger Mealy design example
    - Start another Mealy design example

Handouts and Announcements

- Announcements
  - Homework Problems: Homework 13-3
    - Posted on Carmen this morning
    - Due: 11:59pm Tuesday 4/4
  - Homework Reminder
    - HW 13-1 now past due
    - HW 13-2 Due: 11:59pm Thursday 3/30
  - Read for Friday: 473-479
  - Participation Quiz 12 available 12:25pm today
    - Due 12:25pm tomorrow
    - Available additional 24hr with late penalty

Moore or Mealy

- After class on Monday a student asked about choosing Moore or Mealy for implementing a design
- Reminder from several lectures ago:
- Finite State Machines
  - Special purpose hardware used to implement simple algorithms
  - Two most prevalent approaches: Mealy & Moore
    - Both define specific states
    - Use "inputs" to determine state changes
  - Mealy:
    - Output depends on both the present state and the inputs
    - If inputs have a "glitch," Mealy Machine output may change ☹
  - Moore:
    - Output depends only on present state
    - Usually has more states ☹ (see my comment in solution of HW 13-2)
    - Stable outputs between clock transitions ☺

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

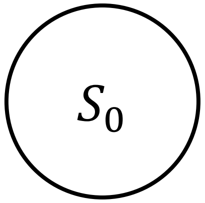
1. Start by identifying sample input and output sequences. Doing this also helps you understand the problem statement.

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

Z: 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 1 0 1 0

Mealy Design Example: Detector of 110 or 1010

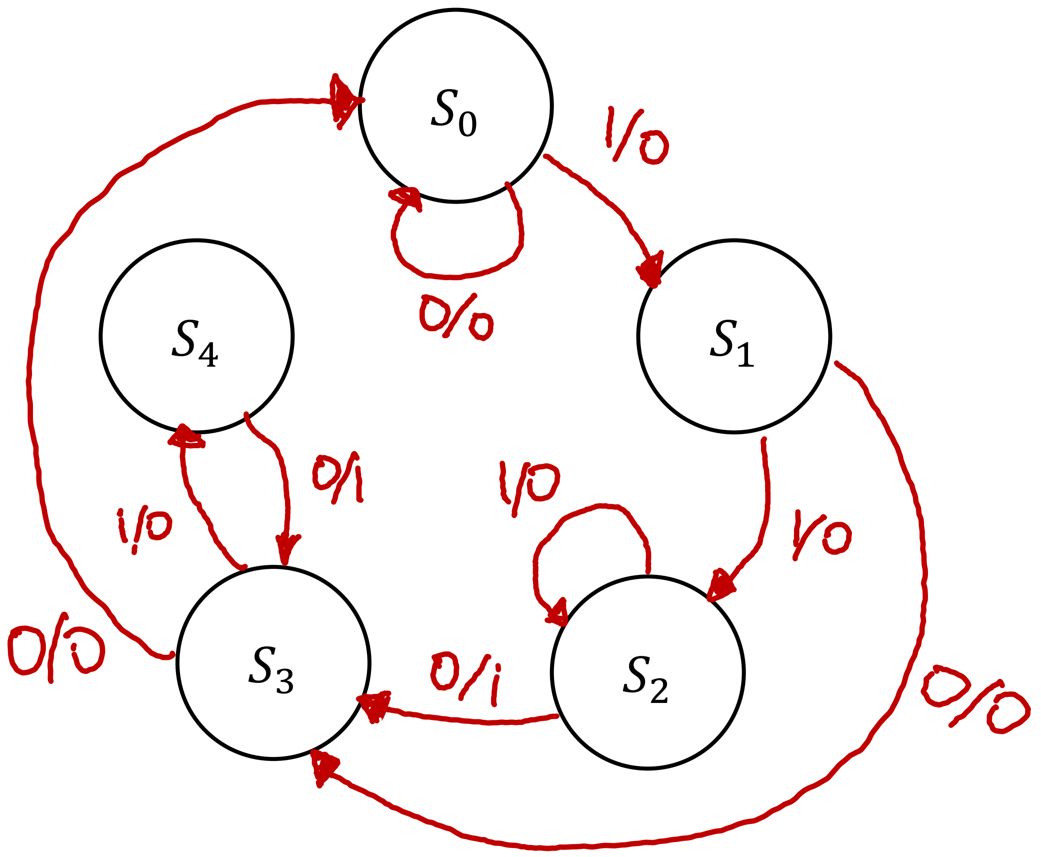
2. Determine an initial state and any condition that causes a reset to that state (if there are any)



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



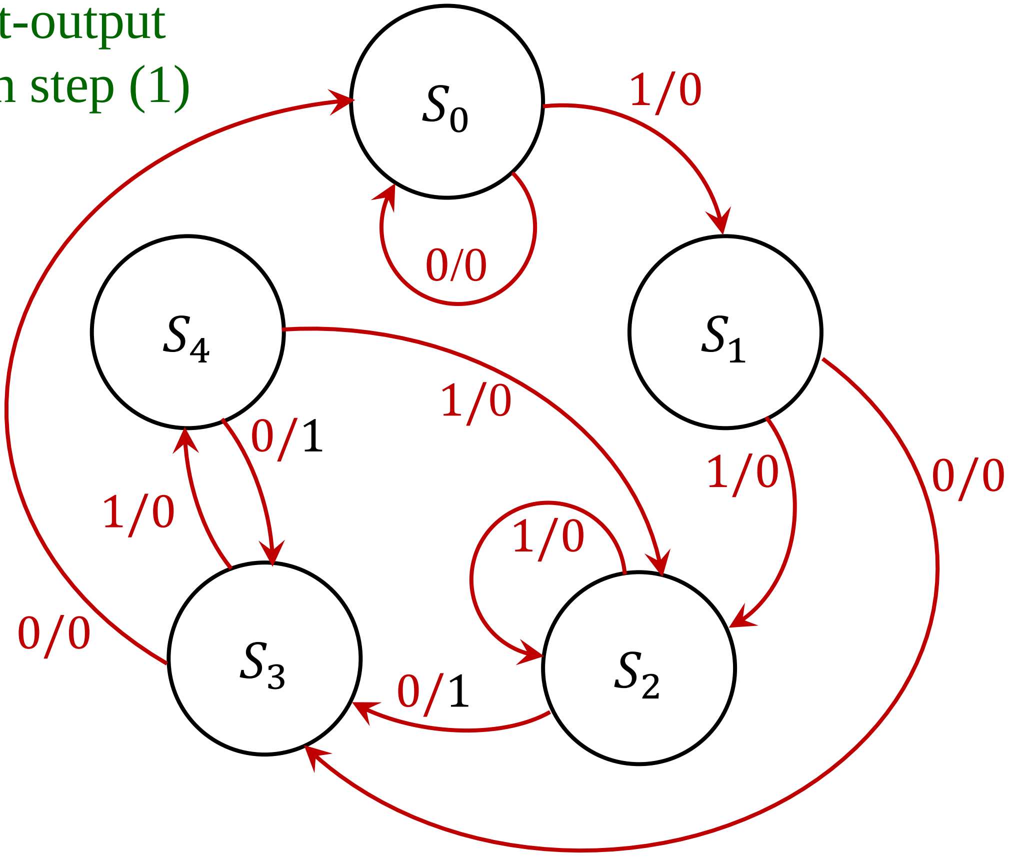
State	Sequence Received
S_0	Reset
S_1	1
S_2	11
S_3	110

State	Sequence Ending In
S_0	Reset or 0 (not 10)
S_1	1 (not 11 or 101)
S_2	11
S_3	10
S_4	101

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:		0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	1	0	1	0
State:	S_0	S_0	S_0	S_1	S_2	S_3	S_4	S_3	S_0	S_0	S_1	S_2	S_2	S_3	S_4	S_2	S_3	S_4	S_3	S_4	S_3	S_0

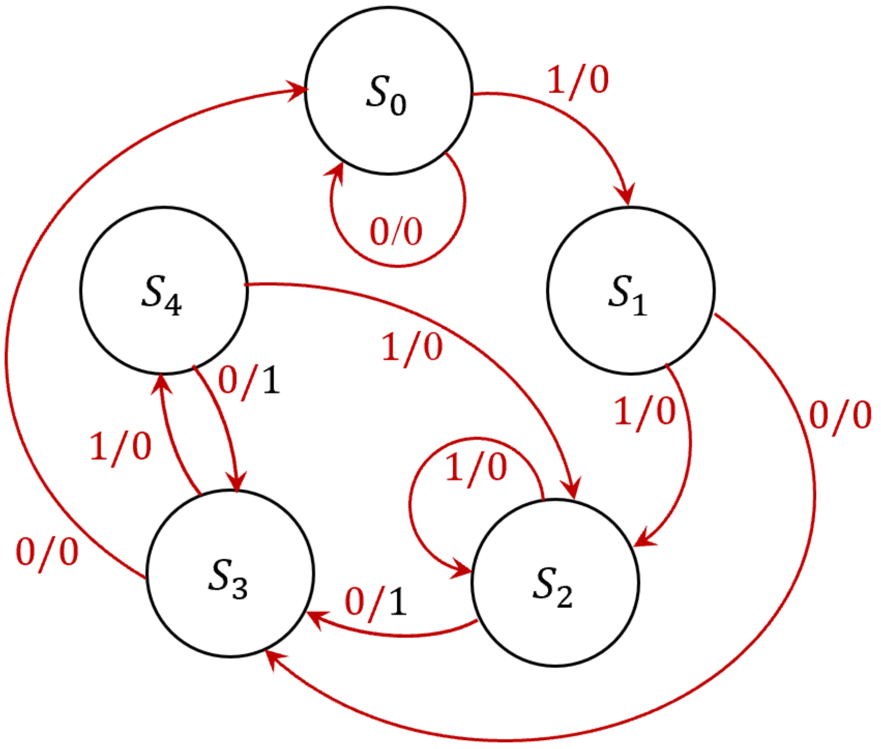
Review of Steps to Complete the Design

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

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

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

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

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

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

$B^+ = B'C'X$

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

$C^+ = A'X$

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

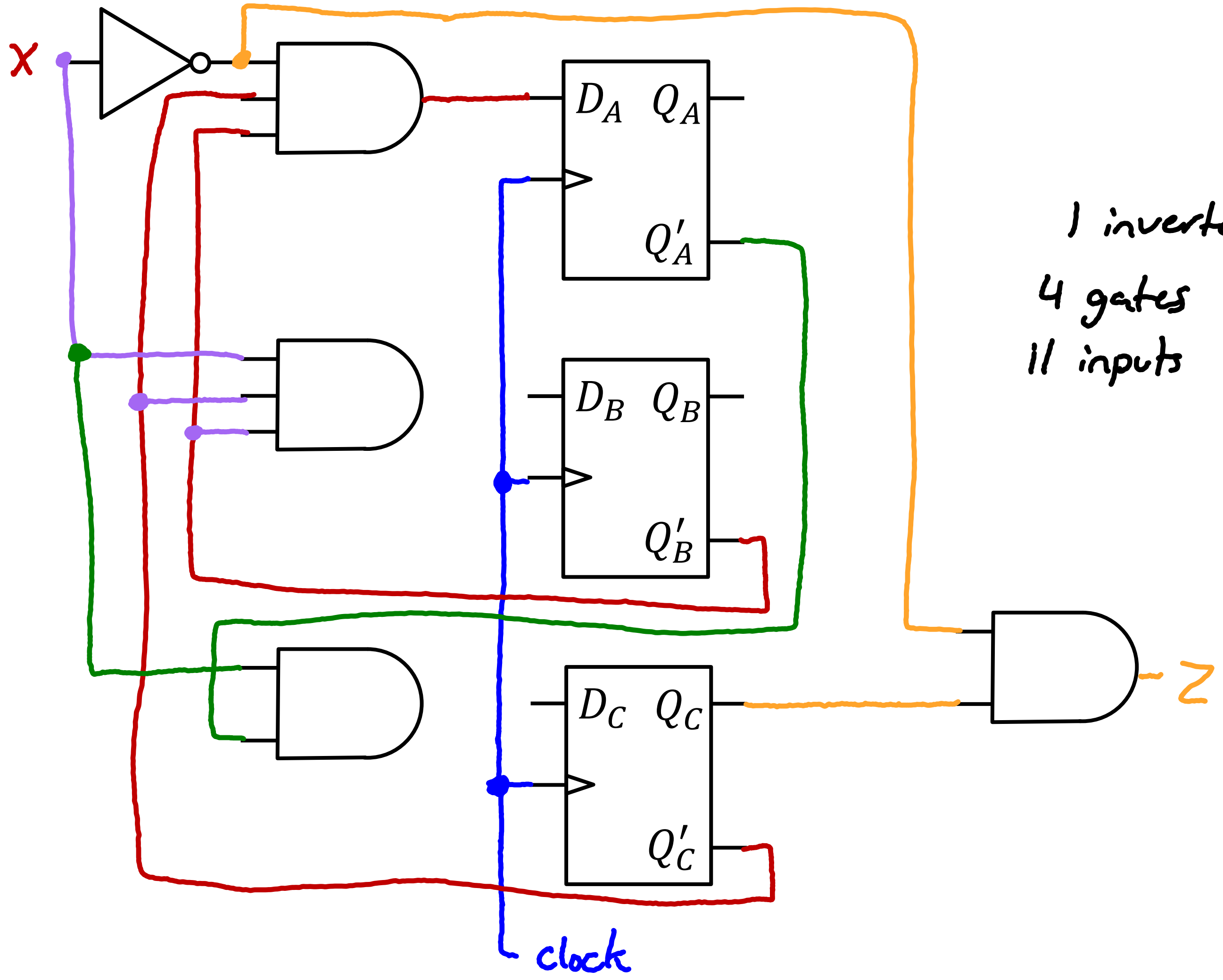
$Z = CX'$

Note: All don't cares realized as 0 in next-state maps $\rightarrow$ 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)

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



1 inverter
4 gates
11 inputs

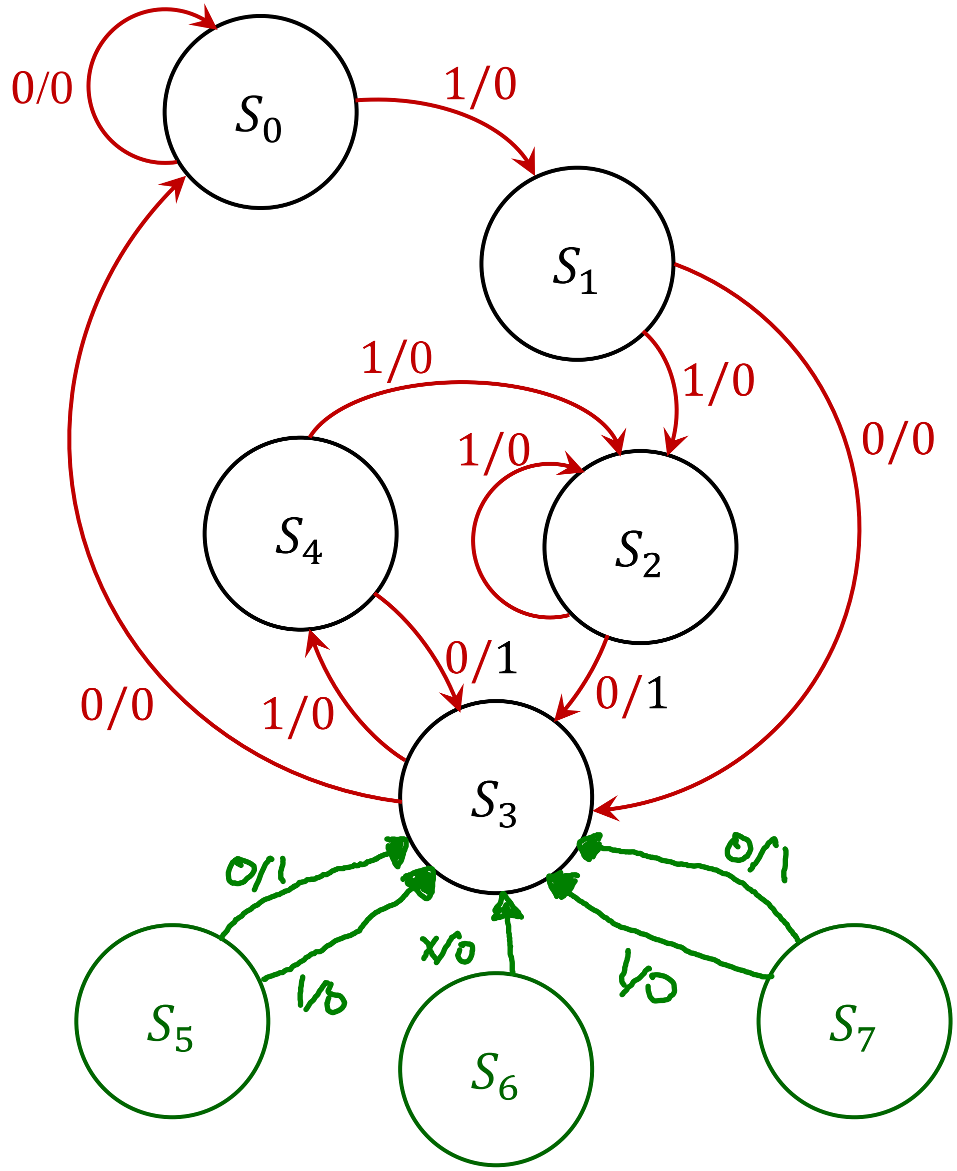
Steps to Analyze a State Machine Circuit

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

THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

ECE2060

Mealy Design Example: Detector of 110 or 1010



	ABC	$A^+B^+C^+$ $X=0$	$A^+B^+C^+$ $X=1$	Present Z $X=0$	Present Z $X=1$
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

$Z = CX'$

Mealy Design Example: Detector of 4-bit sequences 1010 or 0010

- 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 1010 or 0010
  - For this example, data is known to be in 4-bit words, reset after each word is received
  - From our guidelines

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

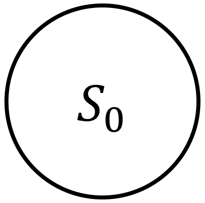
```
X:    0010 0100 1010 1101
Z:    0001 0000 0001 0000
```

THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

ECE2060

Mealy Design Example: Detector of 4-bit sequences 1010 or 0010

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



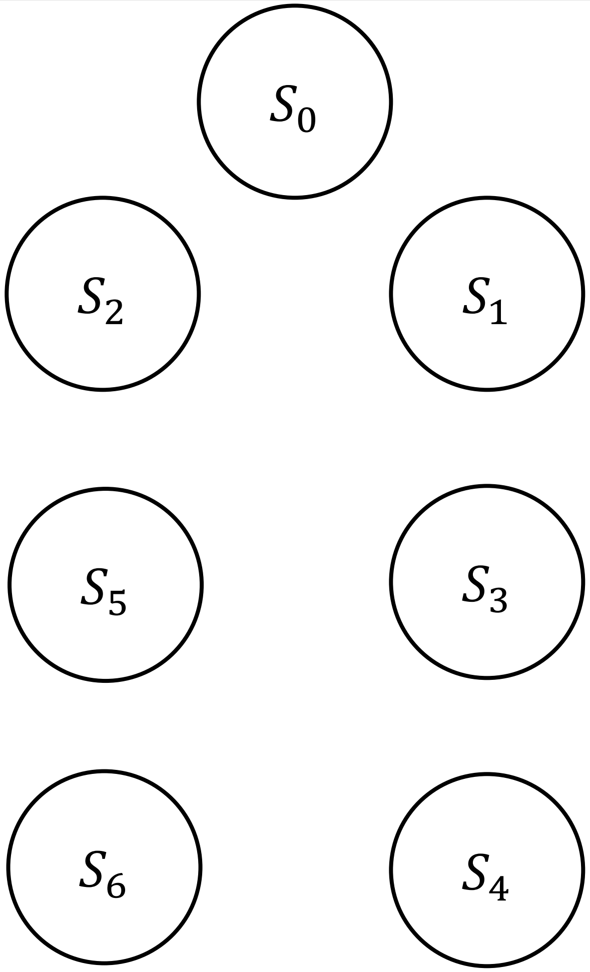
X: 0010 0100 1010 1011

Z: 0001 0000 0001 0000

State	Sequence Received

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

Mealy Design Example: Detector of 4-bit sequences 1010 or 0010

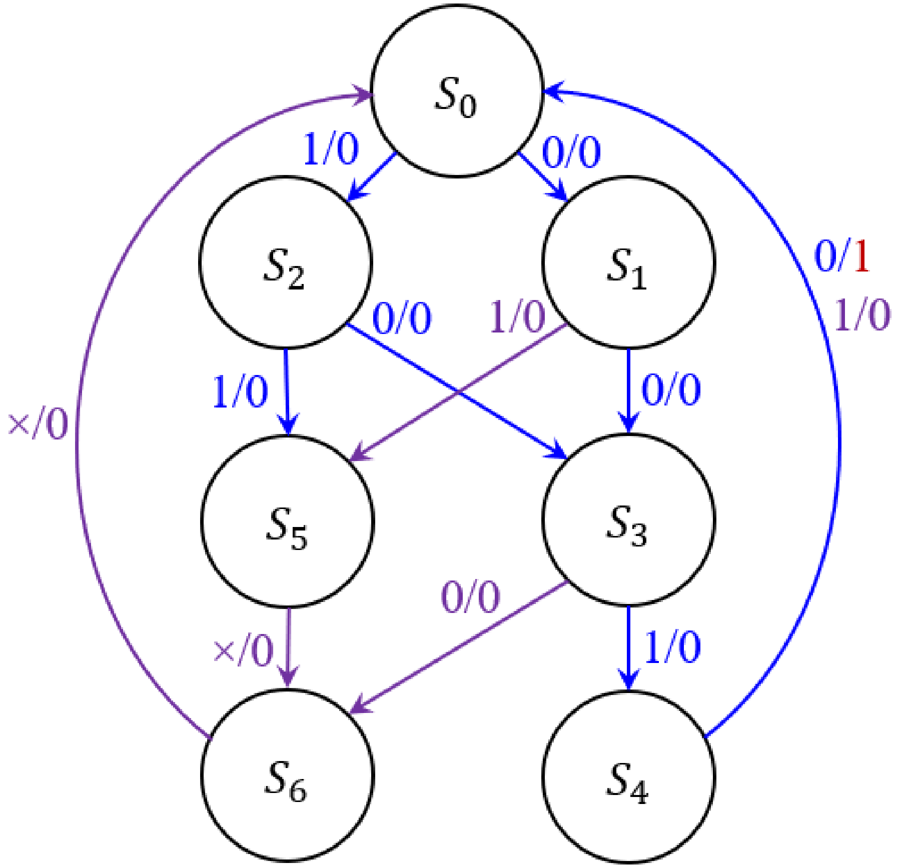


State	Sequence Received
S_0	Reset

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 4-bit sequences 1010 or 0010

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



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

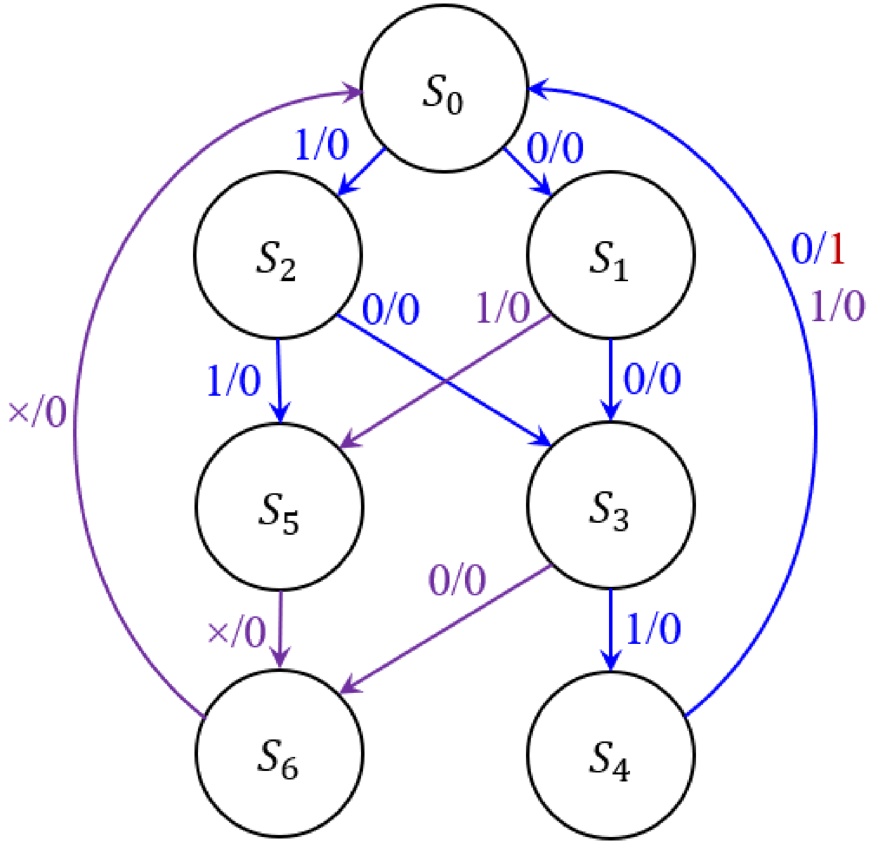
Z:

State: S_0

Mealy Design Example: Detector of 4-bit sequences 1010 or 0010

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				
S_5				
S_6				



Mealy Design Example: Detector of 4-bit sequences 1010 or 0010

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_1	S_2	0	0
S_1	S_3	S_5	0	0
S_2	S_3	S_5	0	0
S_3	S_6	S_4	0	0
S_4	S_0	S_0	1	0
S_5	S_6	S_6	0	0
S_6	S_0	S_0	0	0

	$A^+B^+C^+$		Present Z	
ABC	$X=0$	$X=1$	$X=0$	$X=1$
000			0	0
011			0	0
101			0	0
001			0	0
110			1	0
100			0	0
010			0	0

Mealy Design Example: Detector of 4-bit sequences 1010 or 0010

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 $ABCX = 111\times$

	$A^+B^+C^+$		Present Z	
ABC	$X = 0$	$X = 1$	$X = 0$	$X = 1$
000	0 1 1	1 0 1	0	0
011	0 0 1	1 0 0	0	0
101	0 0 1	1 0 0	0	0
001	0 1 0	1 1 0	0	0
110	0 0 0	0 0 0	1	0
100	0 1 0	0 1 0	0	0
010	0 0 0	0 0 0	0	0

CX \ AB	00	01	11	10
00				
01				
11				
10				

$A^+ =$

CX \ AB	00	01	11	10
00				
01				
11				
10				

$B^+ =$

CX \ AB	00	01	11	10
00				
01				
11				
10				

$C^+ =$

CX \ AB	00	01	11	10
00				
01				
11				
10				

$Z =$

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