Lecture Outline

Reminders to self:

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

- Last Lecture
  - Continued analysis & design of clocked sequential circuits
    - Worked another larger Mealy design example
    - Serial data code conversion – as set up for another design example
- Today's Lecture
  - Continue analysis & design of clocked sequential circuits
    - Another Mealy design example with incompletely specified state table
    - Rework Mealy design example from last lecture with simplified state graph

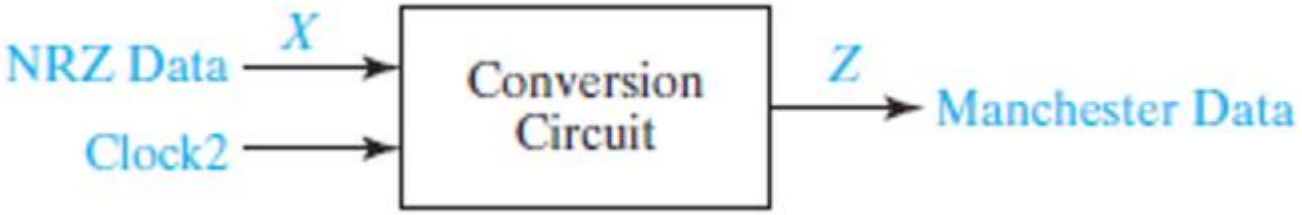
Handouts and Announcements

- Announcements
  - Homework Reminder
    - HW 13-3 Due: 11:59pm Tuesday 4/4
    - HW 13-4 Due: 12:30pm Wednesday 4/5
  - This is last lecture
  - Prof. Valco's office hours:
    - W 4/5, F 4/7, and M 4/10 11:30-12:25 in this lecture hall
    - I will also hold my regular office hours on those days
    - My last office hours for the semester will be on Monday 4/10
  - Mr. Subramanian's office hours:
    - Srini will hold his regular office hours this week
    - His last office hours for the semester will be Thursday 4/6

Handouts and Announcements

- Announcements
  - Mini-Exam 6 Reminder
    - Available 5pm Monday 4/10 through 5:00pm Tuesday 4/11
    - Due in Carmen PROMPTLY at 5:00pm on 4/11
    - 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 posted on Carmen
  - Sample Mini-Exams 8 and 9 from Au20 also posted

NRZ to Manchester Converter



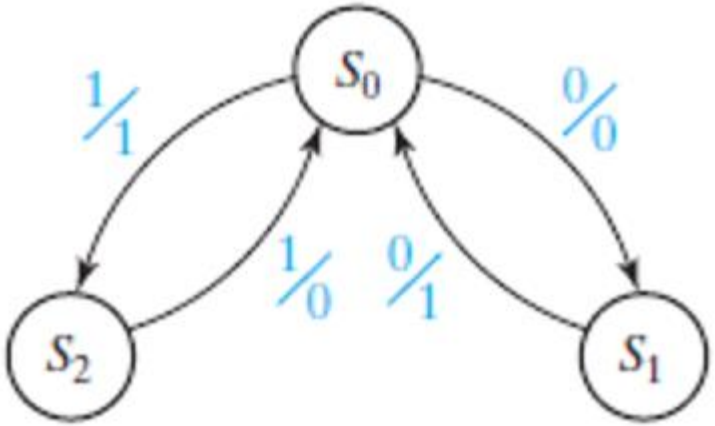
- Serial NRZ data in at input *X*
- Serial Manchester-coded data out at output *Z*
- To facilitate conversion implemented as transitions at middle of bit-time, use clock with double the frequency (half the period): Clock2
- Input serial data stream, at base Clock frequency, is (see previous slide)

↑ at middle of bit-time to encode 0

↓ at middle of bit-time to encode 1

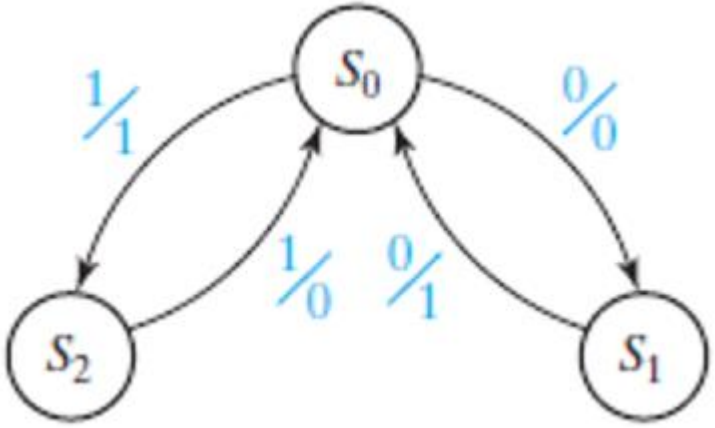
NRZ to Manchester Converter

- Input serial data stream, at base Clock frequency, is 01110010 (see previous slide)
- Output serial data stream, at Clock2 frequency:



- WRT Clock2, input data always in
- From S_0, with 00 input (Mealy)
  - Output 0 when
  - Output 1 when
- From S_0, with 11 input
  - Output 1 when
  - Output 0 when
  - State changes at edge of Clock2

NRZ to Manchester Converter

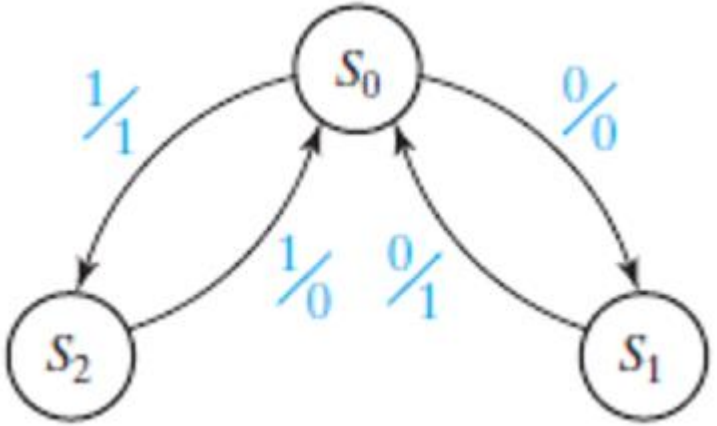


- In previous examples I verbalized, "make sure there is a transition out of each state for each input combination" – but not written in guidelines
- True in previous examples since each input combination existed
- But not here
- No 01 pair: no 1 input when in S_1
- No 10 pair: no 0 input when in S_2

Guidelines for construction of state graphs

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)

NRZ to Manchester Converter



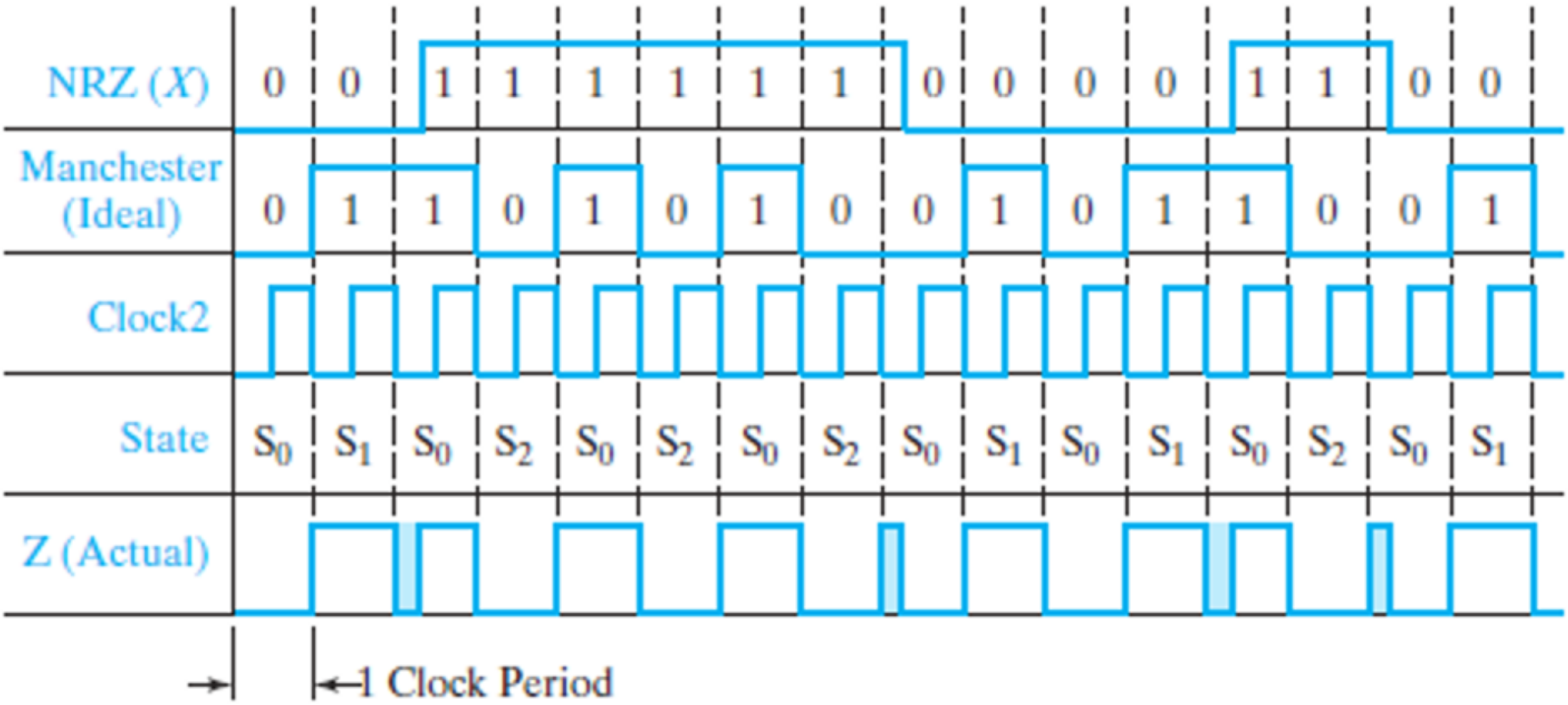
Incompletely specified State Table

- We previously had don't cares in the table and next state and output when entire states did not exist (e.g. State Graph had $< 2^n$ states where n = number of flip-flops)
- Now there will be don't cares in the State Table
- Associated with that do not exist, due to input combinations that

Present State	Next State		Output (Z)	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
S_0	S_1	S_2	0	1
S_1	S_0	–	1	–
S_2	–	S_0	–	0

X =0 does not occur when in S_1

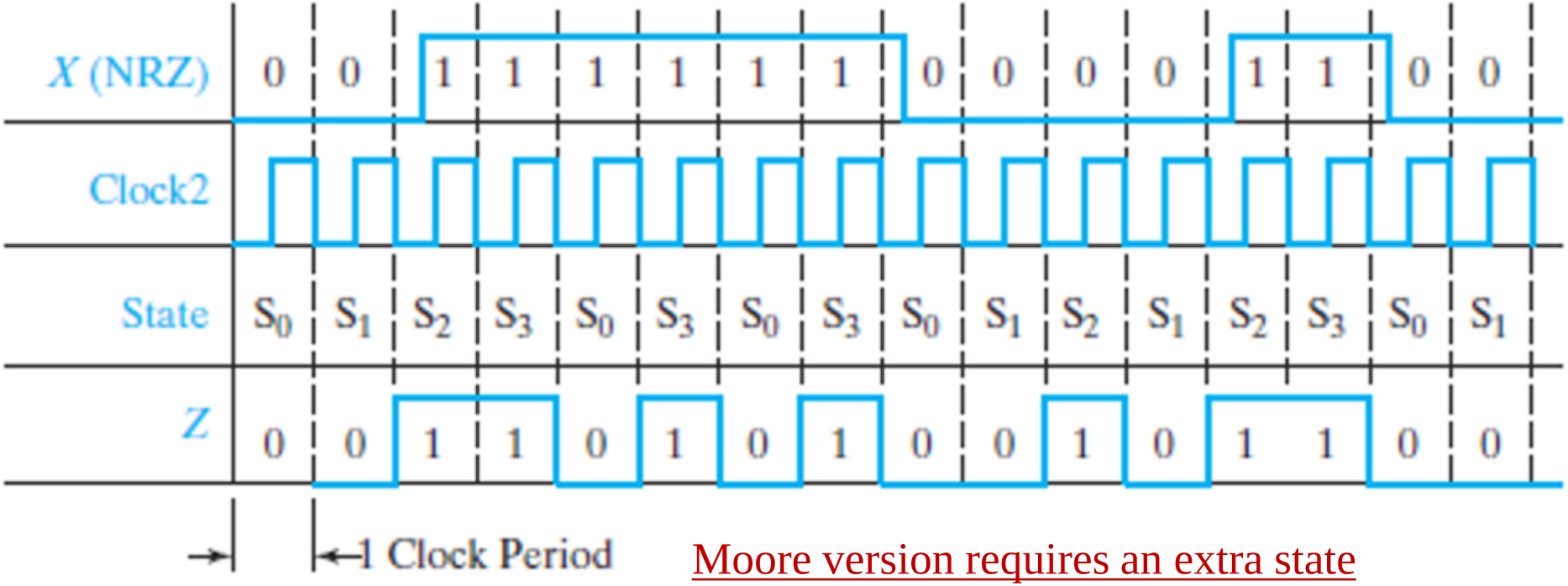
NRZ to Manchester Converter

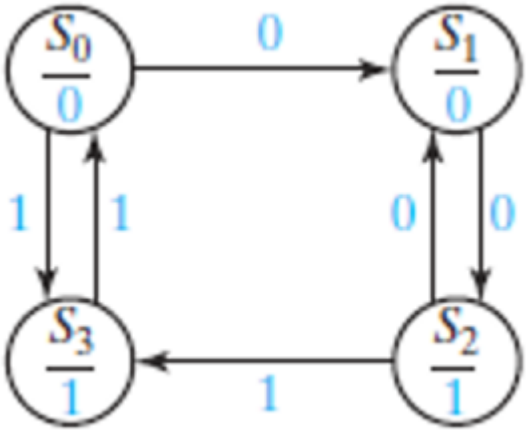


- Remember that Mealy Machines may have false output, between active clock edge and when input settles to new value
- May result in glitches in output
- So far: ideal output, but note NRZ data timing offset to demonstrate glitches
- Ignored if output read locally with synchronized clock, but don't want to transmit

Moore version: output tied to state
No false output glitches

NRZ to Manchester Converter

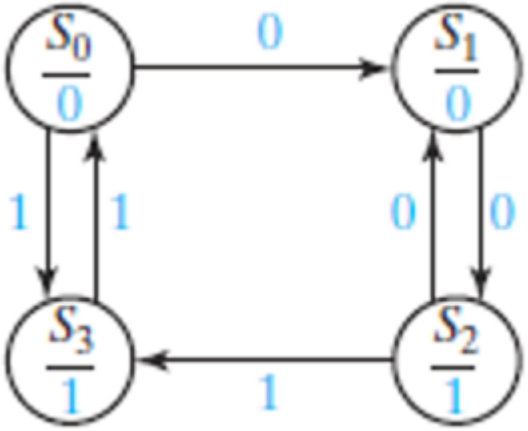




Moore version requires an extra state

- From S_0, with 00 input
  - Output 0 after
  - Output 1 after
- From S_0, with 11 input
  - Output 1 after
  - Output 0 when
- But no false outputs - output changes after clock edge

NRZ to Manchester Converter



Moore version still incompletely specified

- No transition out of
- Only for
- No transition out of
- Only for

Present State	Next State $X=0$	Next State $X=1$	Present Output (Z)
S_0	S_1	S_3	0
S_1	S_2	–	0
S_2	S_1	S_3	1
S_3	–	S_0	1

$X = 1$ does not occur when in S_1

$X = 0$ does not occur when in S_3

Review of Steps to Complete the Design

1. Develop a State Graph (Done)
2. Fill in a State Table (Done)
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)

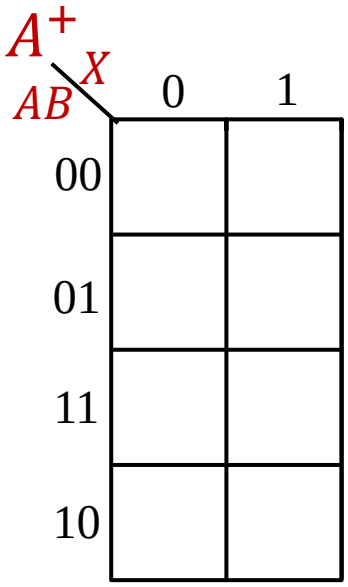
NRZ to Manchester Converter

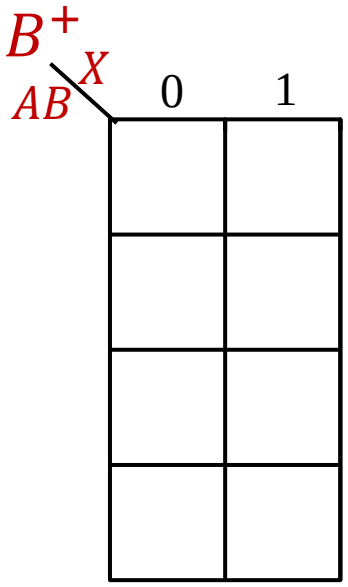
Present State	Next State $X = 0$	Next State $X = 1$	Present Output (Z)
S_0	S_1	S_3	0
S_1	S_2	–	0
S_2	S_1	S_3	1
S_3	–	S_0	1

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

NRZ to Manchester Converter

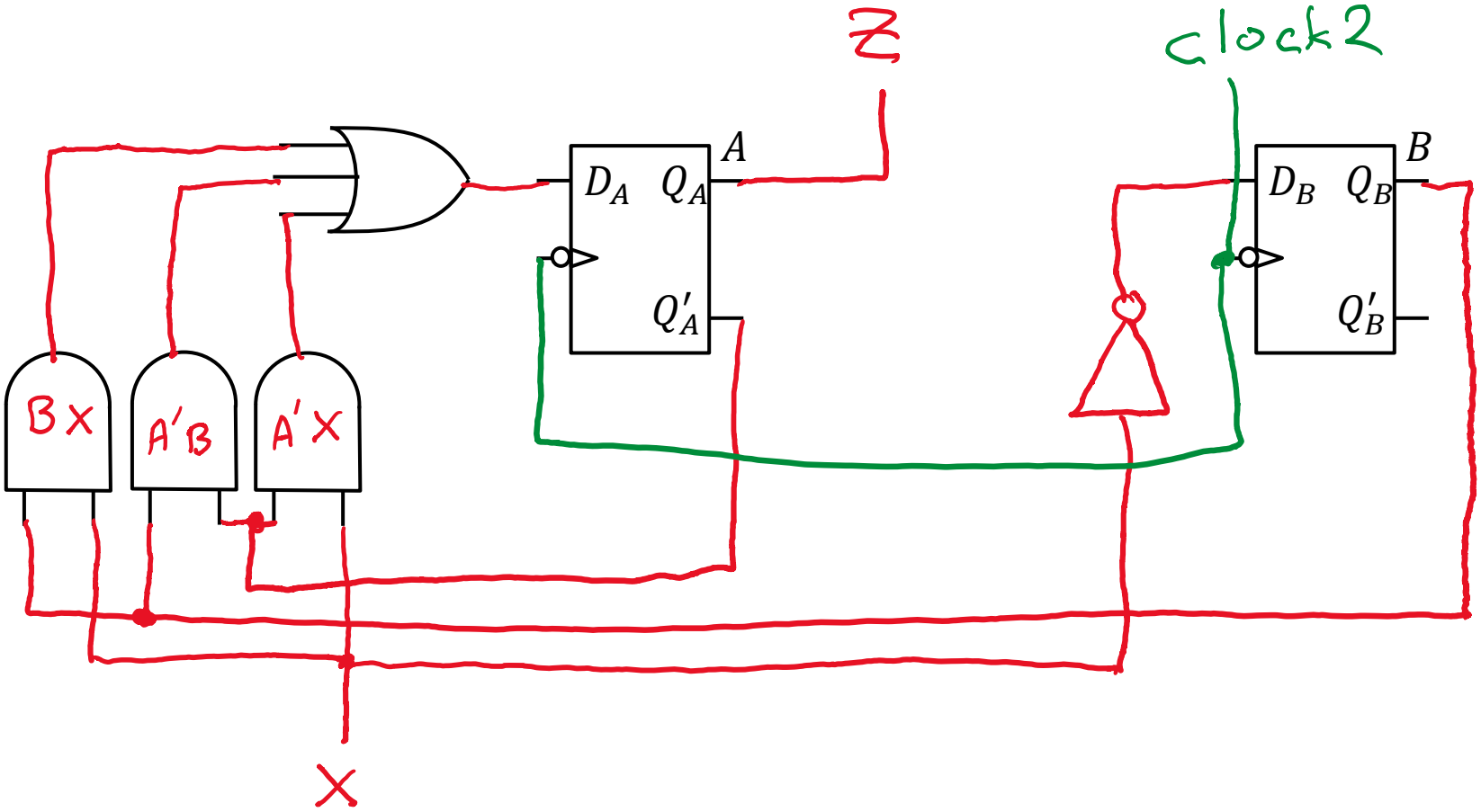
Present	Next A^+B^+		Output
AB	$X = 0$	$X = 1$	Z
S_0 00	0 1	1 0	0
S_1 01	1 1	**	0
S_2 11	0 1	1 0	1
S_3 10	**	0 0	1





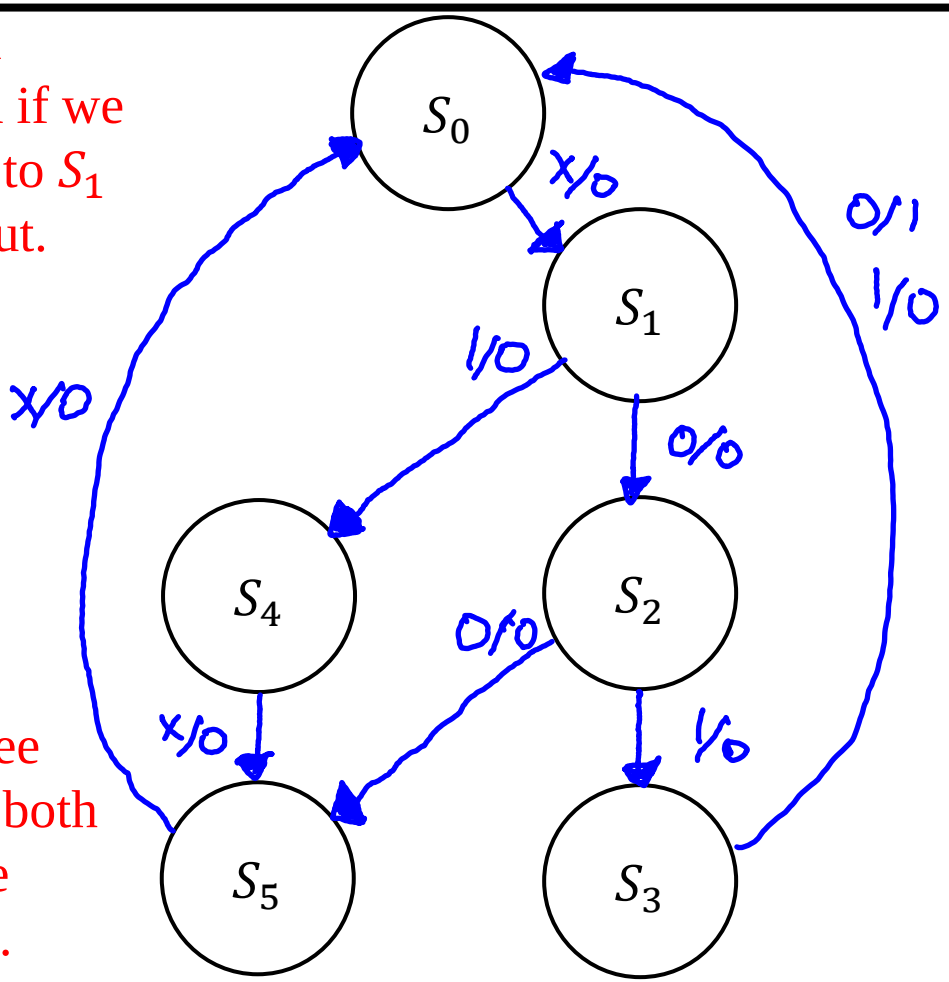
NRZ to Manchester Converter

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



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

Last lecture a student asked if we could just go to S_1 for either input.



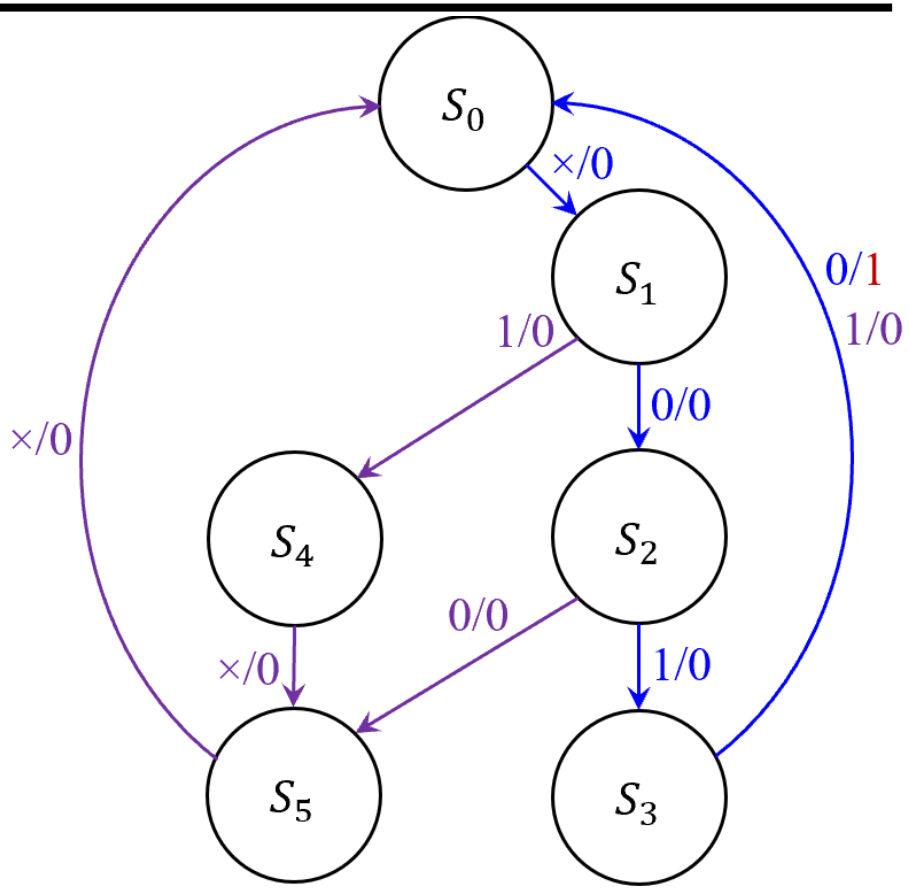
State	Sequence Received
S_0	Reset
S_1	1st bit received
S_2	00 or 10
S_3	001 or 101
S_4	2 inputs not in sequences
S_5	3 inputs not in sequences

Since last three bits same for both sequences the answer is yes.

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)

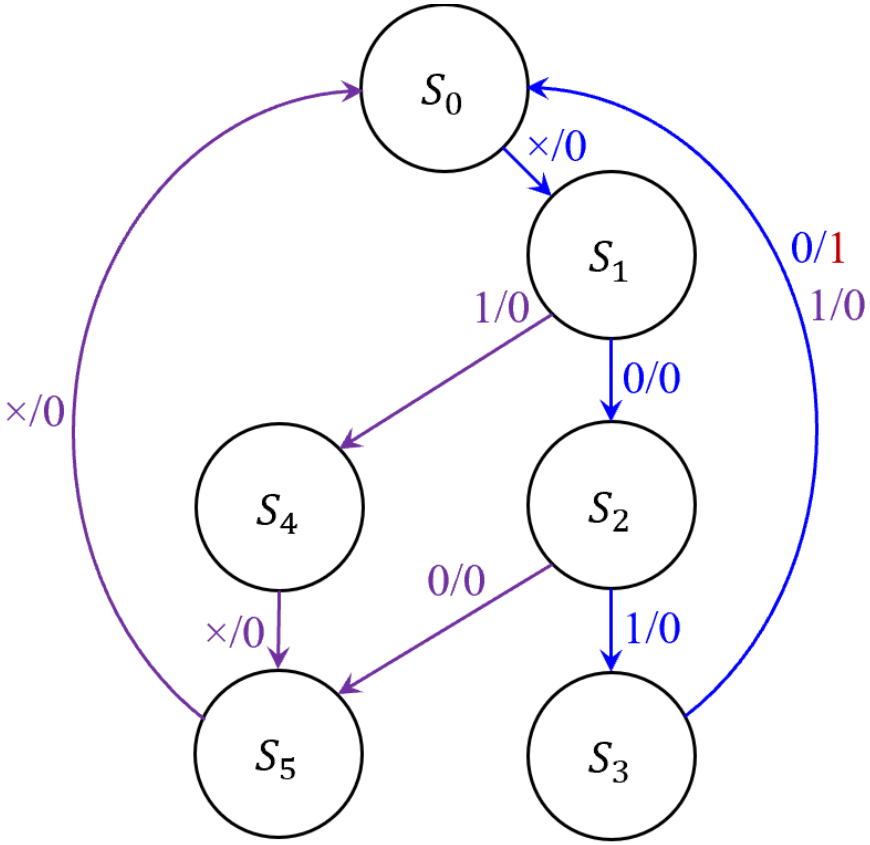


X:	0	0	1	0		0	1	0	0		1	0	1	0		1	0	1	1
Z:	0	0	0	1		0	0	0	0		0	0	0	1		0	0	0	0
State: S_0	S_1	S_2	S_3	S_0		S_1	S_4	S_5	S_0		S_1	S_2	S_3	S_0		S_1	S_2	S_3	S_0

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



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?

Trying to decrease number of 1s in Next-State Maps to reduce expressions

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

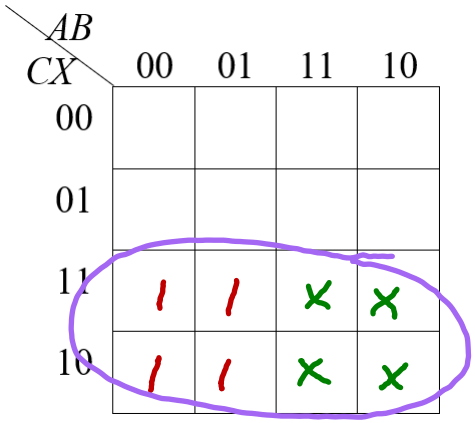
Present State	Next State		Present Z	
	$X=0$	$X=1$	$X=0$	$X=1$
S_0	S_1	S_1	0	0
S_1	S_2	S_4	0	0
S_2	S_5	S_3	0	0
S_3	S_0	S_0	1	0
S_4	S_5	S_5	0	0
S_5	S_0	S_0	0	0

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

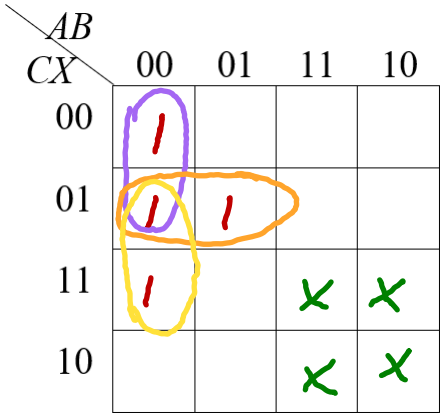
4. Generate Flip-Flop Next State and Output Maps
5. Determine SOP Expressions for circuit output(s) and flip-flop input(s)

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

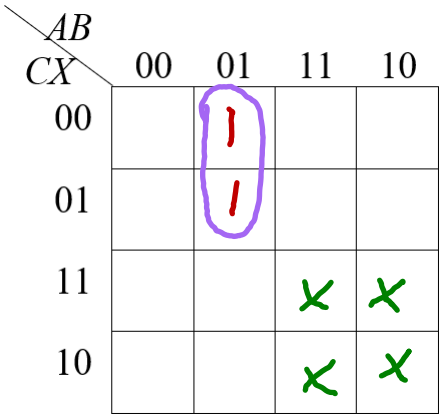
Don't Cares for $ABCX = 101\ , 111$



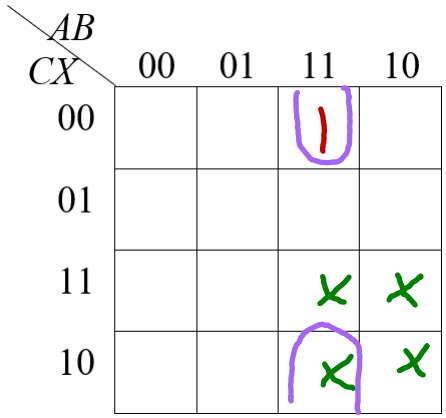
$A^+ = C$



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



$C^+ = A'BC'$



$Z = ABX'$

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

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

$A^+ = C$

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

$C^+ = A'BC'$

$Z = ABX'$