Lecture Outline

Reminders to self:

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

- Last Lecture
  - Registers
    - Basic register, data transfer, accumulator register w/adder
    - Started shift registers
- Today's Lecture
  - Finish Shift Registers
  - Start Counters
    - Three-bit synchronous binary counter
    - Brief look at binary ripple counter (not synchronous)
    - Formal design procedures

Handouts and Announcements

- Announcements
  - Homework Problem 12-1
    - Posted on Carmen this morning
    - Due: 11:25am Wednesday 3/8
    - Data Sheet for CD40194B in Data Sheet Module on Carmen
  - Homework Reminder
    - HW 11-3 posted on Carmen 2/25
    - Due: 11:59pm Thursday 3/2
  - Read for Friday: pages 395-402
  - Mini-Exam 3 regrade in progress
    - Problem 2 & 3 rubric interpretation by GTA
    - Inconsistencies and errors I observed during requested regrades

Handouts and Announcements

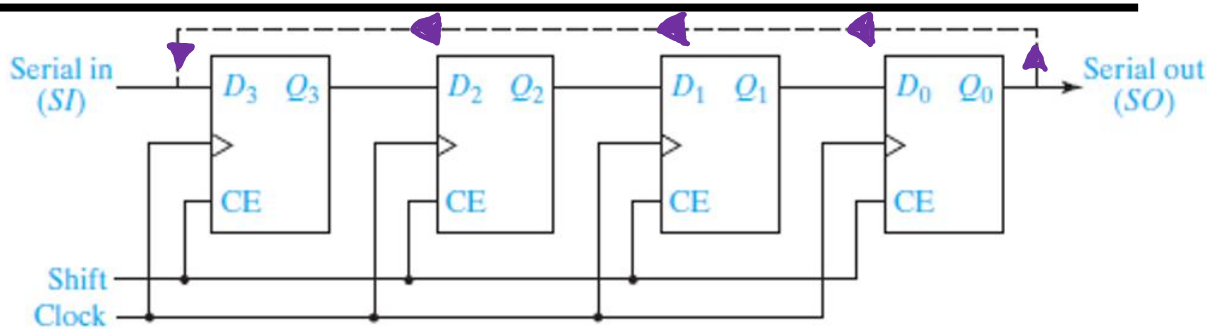
- Announcements
  - Mini-Exam 4 Reminder
    - Available 5pm Monday 3/6 through 5:00pm Tuesday 3/7
    - Due in Carmen PROMPTLY at 5:00pm on 3/7
    - Designed to be completed in ~36 min, but you may use more
    - When planning your schedule:
      - I recommend building in 10-15 min extra
      - To 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 5 and 6 from Au20 also available

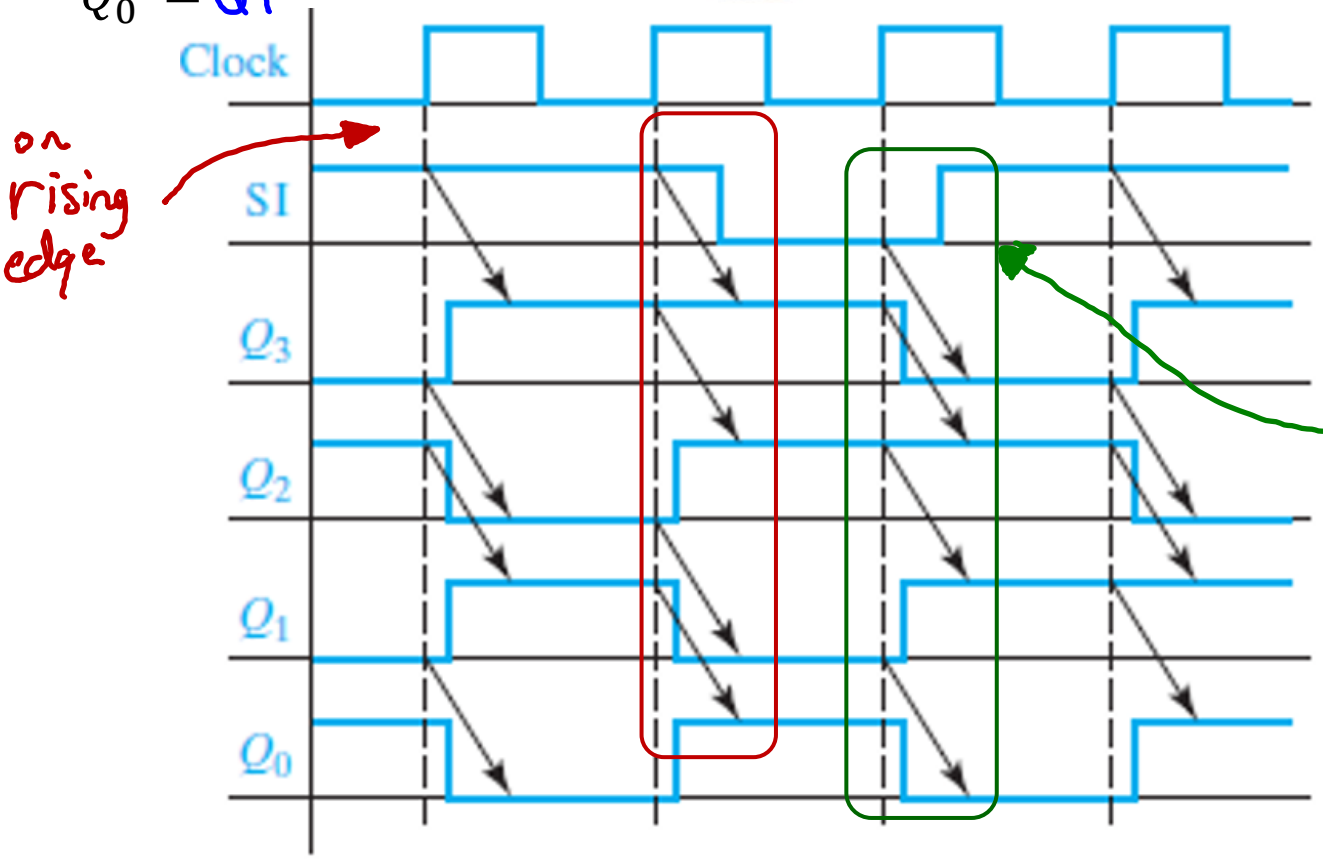
Shift Registers

$Shift = 1$ @ 1st $Clock \uparrow$

- $Q_3^+ =$ SI
- $Q_2^+ =$ Q3
- $Q_1^+ =$ Q2
- $Q_0^+ =$ Q1

0101 -> 1010





- Pattern repeats at 2nd $Clock \uparrow$

1010 -> 1101

- Note that $SI = 0$ before 3rd $Clock \uparrow$

1101 -> 0110

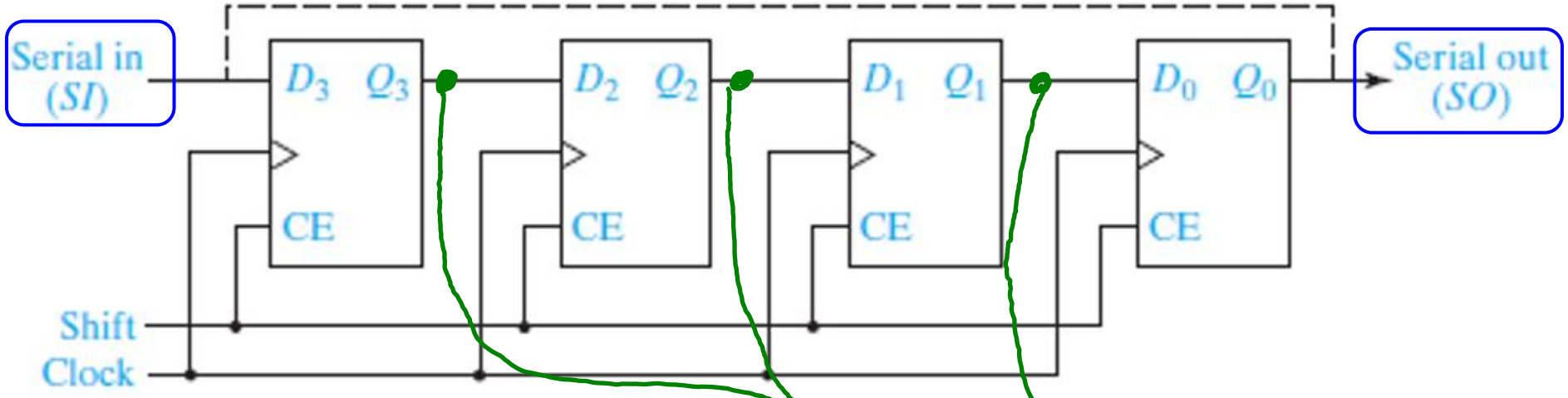
- Back to $SI = 1$ before 4th $Clock \uparrow$

0110 -> 1011

- If $Shift = 0$

(Hold) no change

Shift Registers



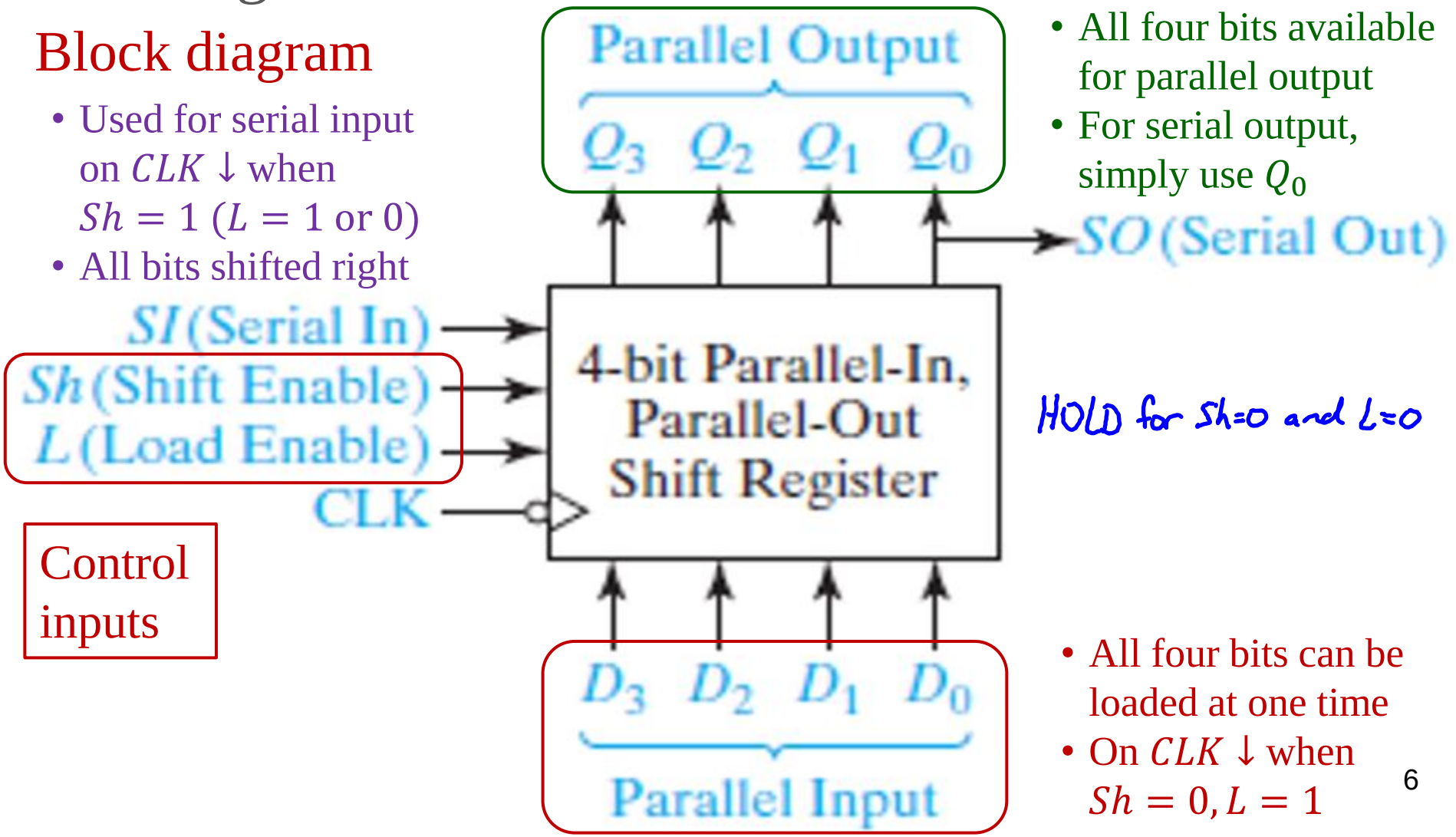
Serial In / Serial Out Registers:

- Serial in means data is
  - Shifted into 1st flip flop one bit at a time, and
  - Flip-flops cannot be loaded in parallel
- Serial out means data
  - Can only be read out of the last flip flop, and
  - Outputs from other flip-flops not connected outside register

Shift Registers with Parallel In / Parallel Out exist

Block diagram

- Used for serial input on $CLK \downarrow$ when $Sh = 1$ ($L = 1$ or 0)
- All bits shifted right



Control inputs

- All four bits available for parallel output
- For serial output, simply use Q_0

HOLD for Sh=0 and L=0

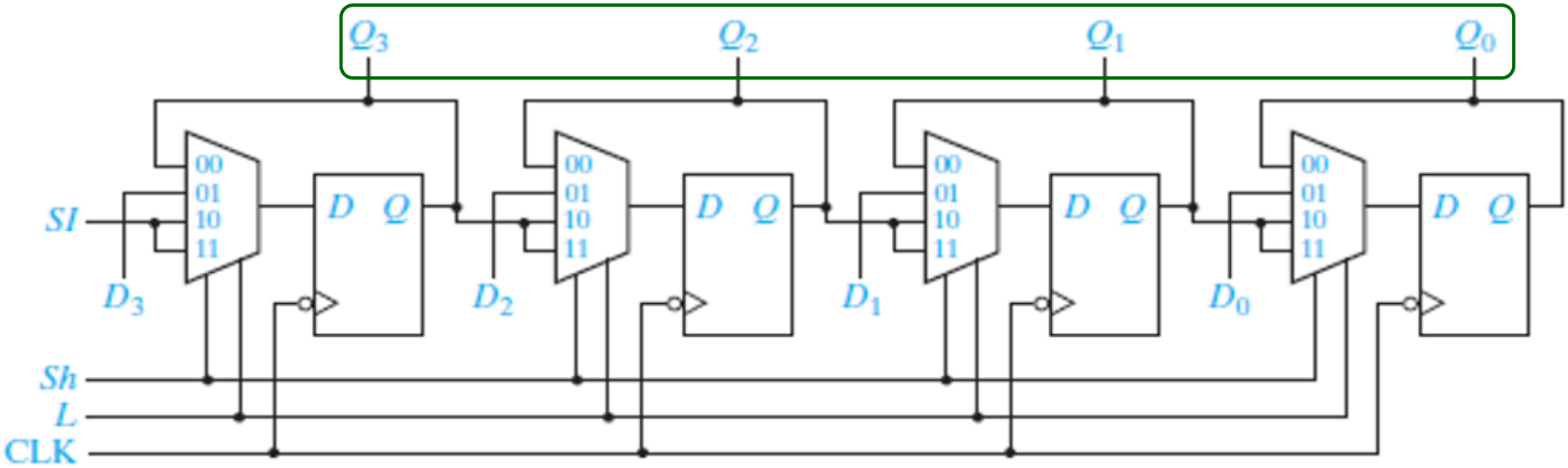
- All four bits can be loaded at one time
- On $CLK \downarrow$ when $Sh = 0, L = 1$

Shift Registers

Implementation with D Flip-Flops and MUXs

- HOLD for $Sh = 0, L = 0$
- All four Q_i fed back to D_i inputs through MUX

- All four bits available for parallel output
- For serial output, simply use Q_0

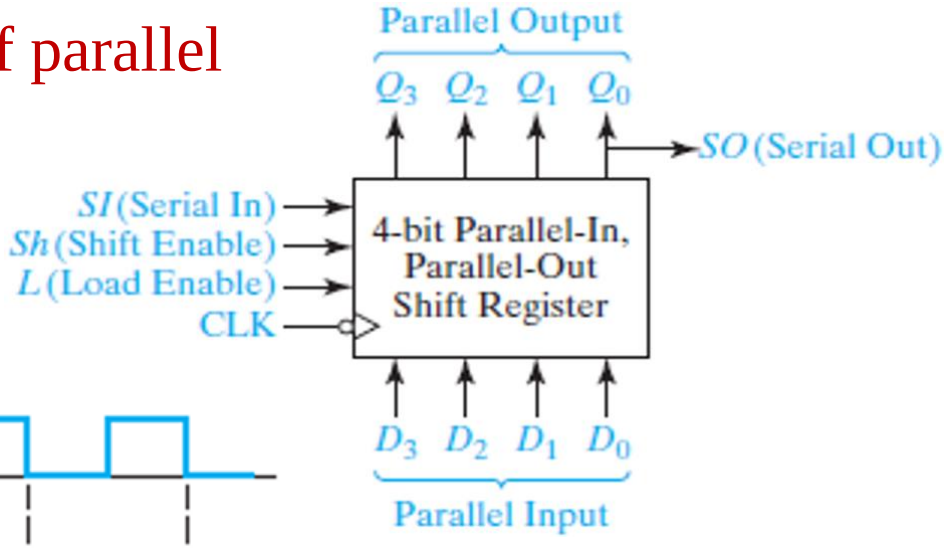


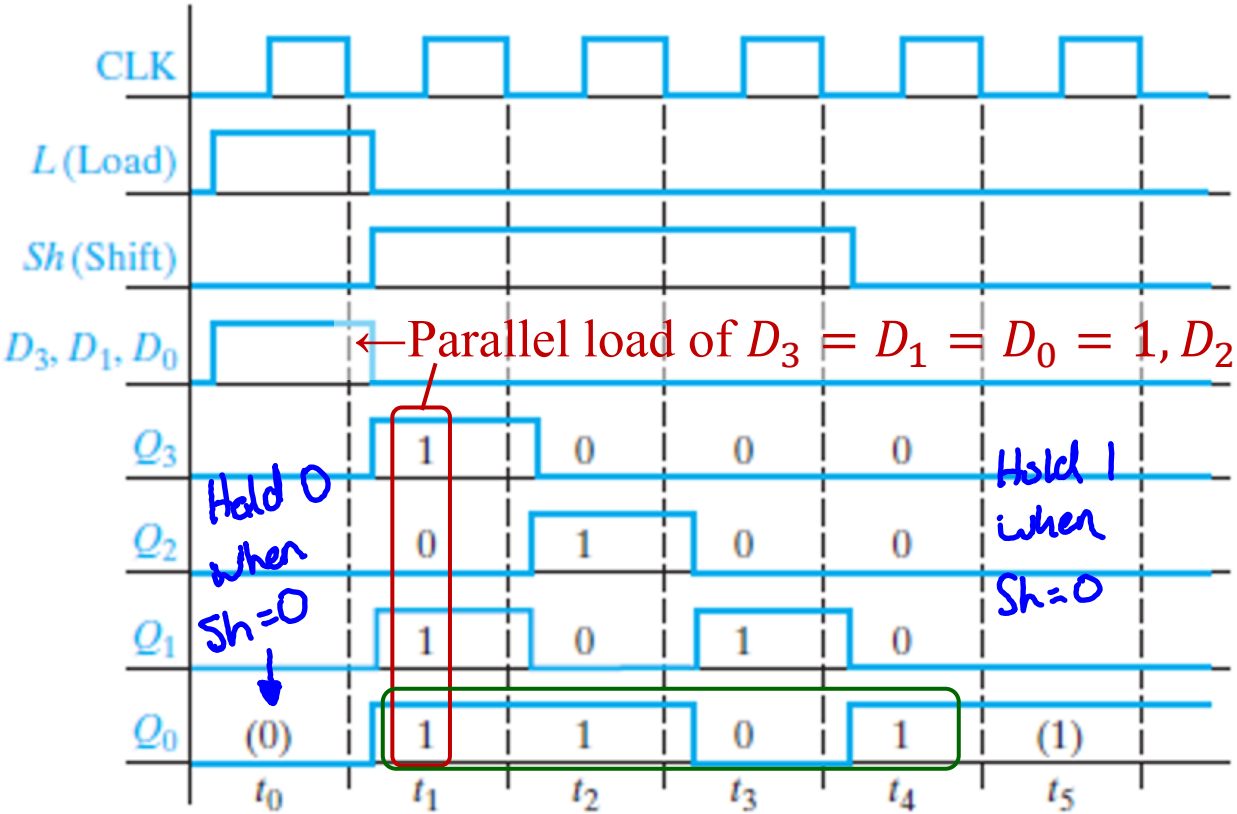
- Used for serial input on $CLK \downarrow$ when $Sh = 1$ ($L = 1$ or 0)
- All bits shifted right

- All four bits can be loaded at one time
- On $CLK \downarrow$ when $Sh = 0, L = 1$

Shift Registers

- Typical application is conversion of parallel data to a serial bit stream
- Timing diagram for this assumes "cleared" 0000 initial state, $SI = 0$
- Parallel load of 1011 at 1st $CLK \downarrow$

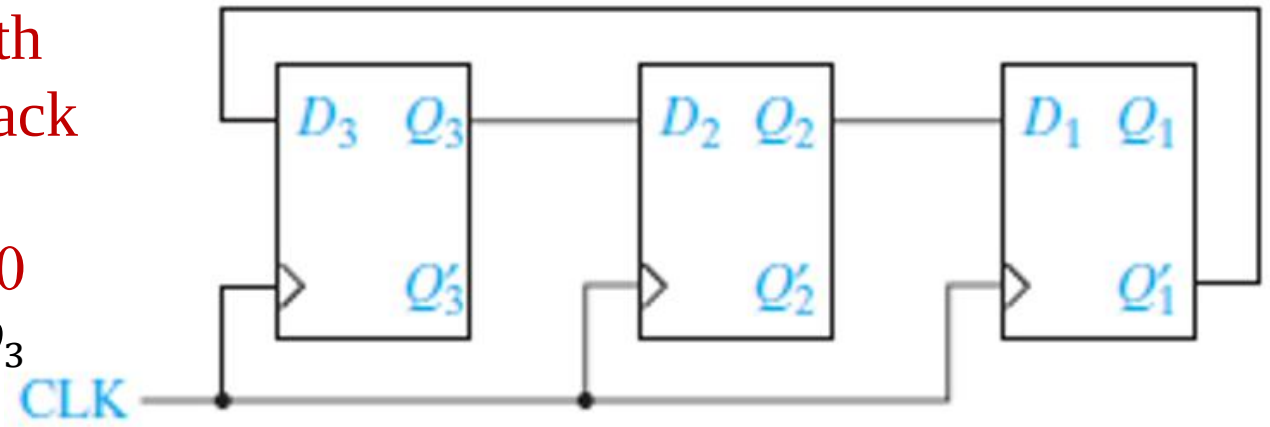


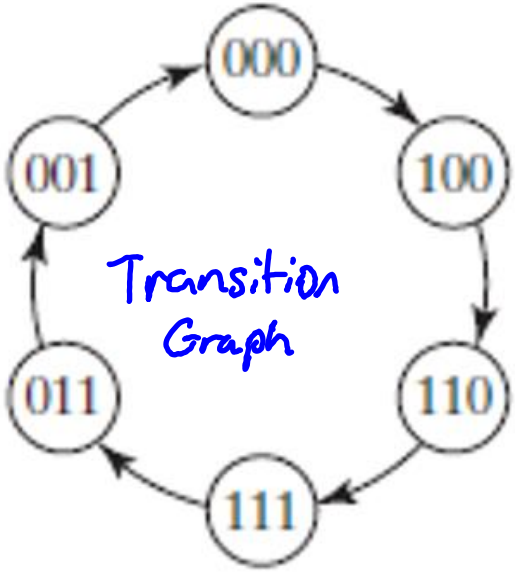


Serial 1101 bit stream at SO (Q_0)

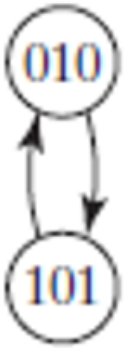
Shift Registers ⇨ Counters

- 3-bit shift register with inverted output fed back to input
- Starting "cleared" 000
  - 1 fed back from Q_1' to D_3
  - Progress around "transition graph" with each $CLK \uparrow$

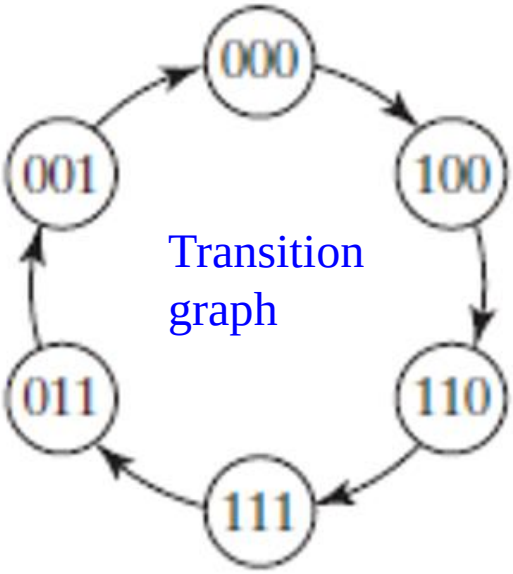




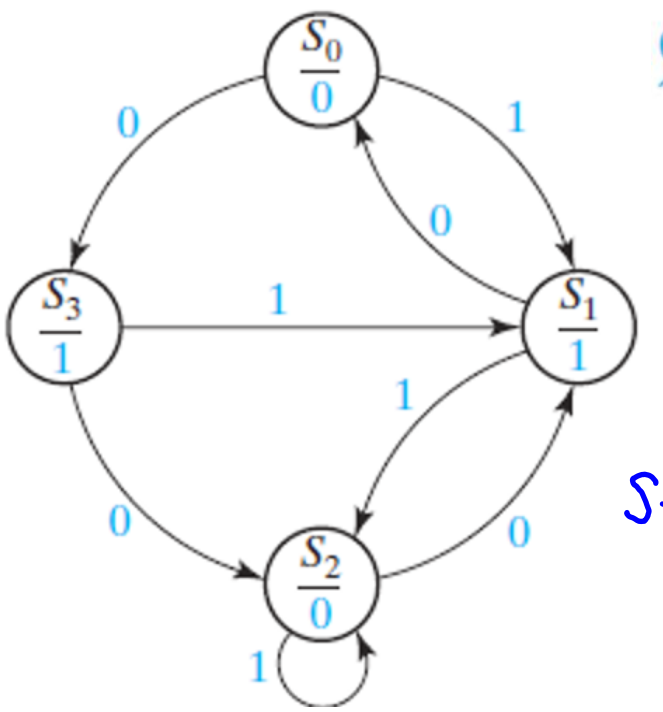
- Note that only six of the eight 3-bit combinations appear on this transition graph
- 010 and 101 are missing
- If register were loaded with 010 to start
  - Next state would be 101
  - And then back to 010
- This is a "secondary loop" transition graph for this counter
- Counter. A circuit that cycles through a fixed sequence of states

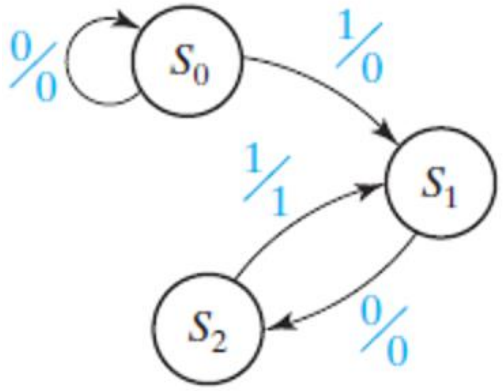


Shift Registers ⇨ Counters



- The combination of values of $Q_3Q_2Q_1$ in each of the circles represent a "state" of the counter
- Later this semester you will learn about "state machines"
  - At that time we will introduce two types of "state graphs"
  - State graphs contain more information

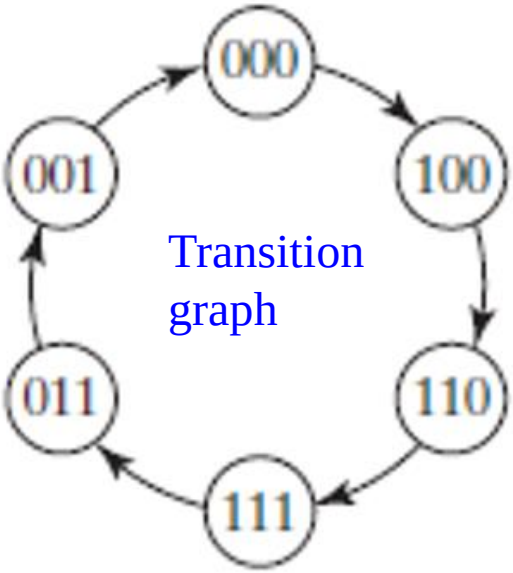


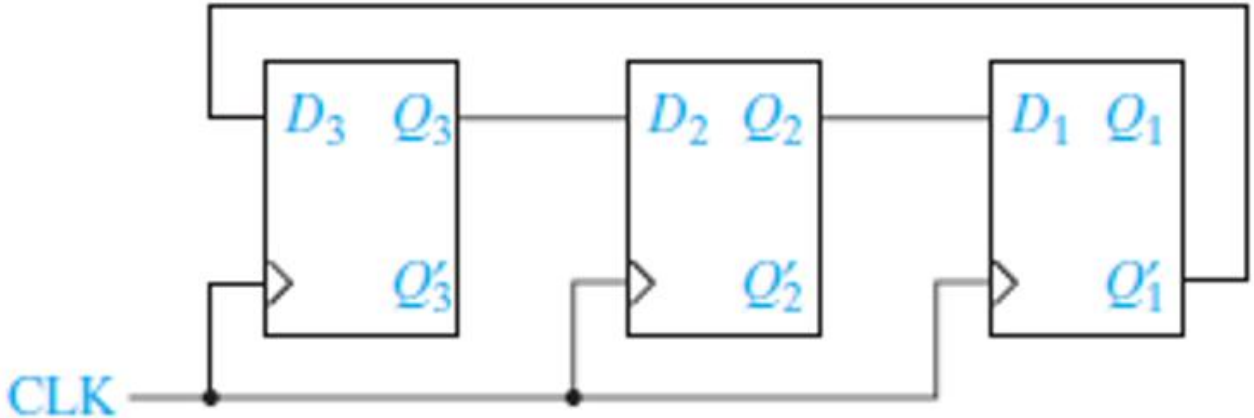


State graphs

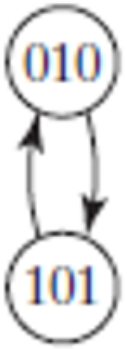
- When asked for a "state graph", do NOT draw a "transition graph"
- When asked for the "transition graph" of a counter, do not draw a "state graph"

Shift Registers ⇨ Counters





- A shift register with inverted feedback is called a "Johnson counter" or "twisted ring counter"
- If feedback is not inverted it is simply a "ring counter"



Secondary loop

Counters

3-Bit Binary Counter:

Transition table

Counting 0_{10} though 7_{10} in binary

Present State			Next State		
C	B	A	C^+	B^+	A^+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Fill in Next State from transition graph

To implement as a Synchronous counter, look for patterns of toggles by bit:

- A always toggles on $Clk\uparrow$ (or $Clk\downarrow$ if low-true clock)
- B toggles on $Clk\uparrow$ when $A=1$
- C toggles when both $B=1$ and $A=1$ on $Clk\uparrow$

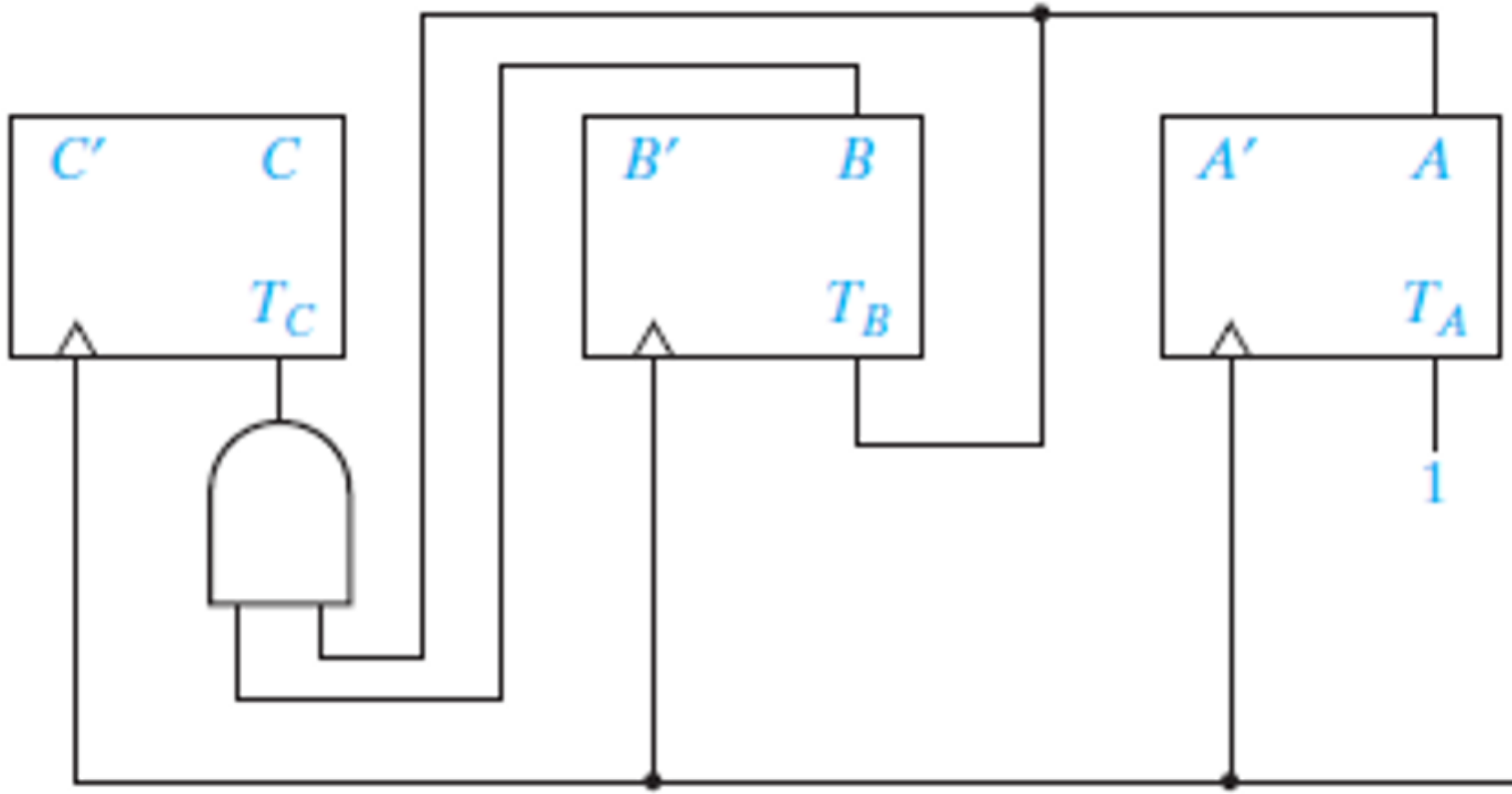
3-Bit Binary Counter: Transition Table

- A always toggles on $Clk \uparrow$
- B toggles on $Clk \uparrow$ when $A = 1$
- C toggles when both $B = 1$ and $A = 1$

Present State			Next State			Flip-Flop Inputs		
C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

3-Bit Binary Counter: Circuit

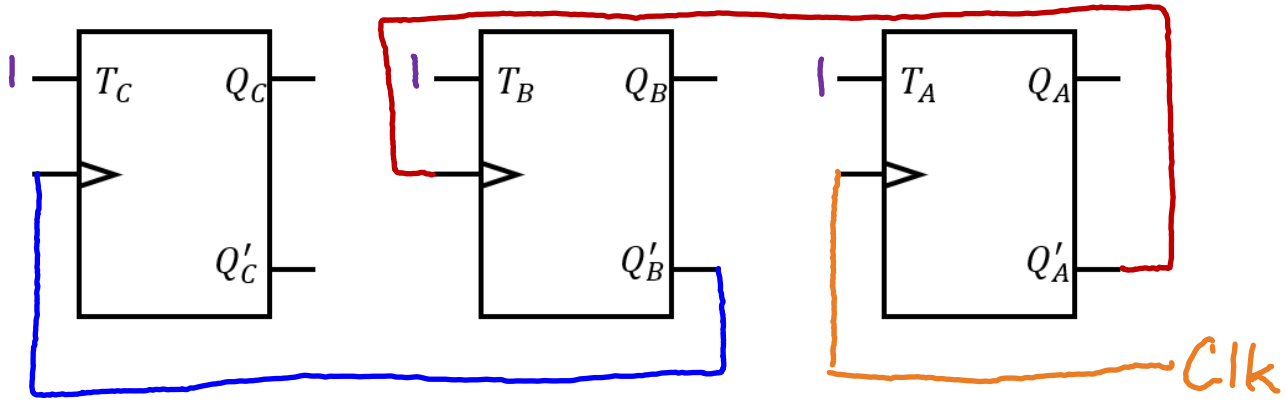
- A always toggles on $Clk \uparrow$
- B toggles on $Clk \uparrow$ when $A = 1$
- C toggles when both $B = 1$ and $A = 1$



Counters

Synchronous and Ripple Counters:

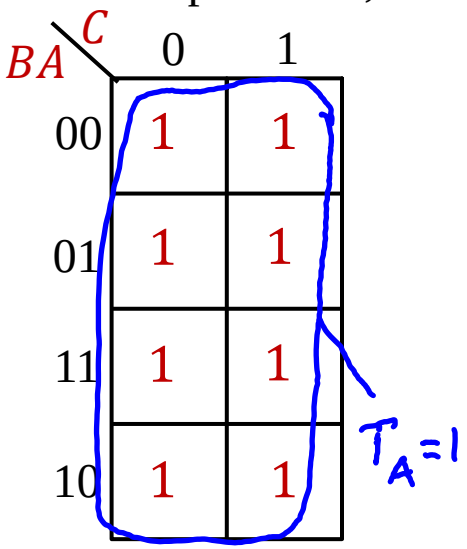
- For synchronous counters, operation of flip-flops synchronized by common clock pulse: when several flip-flops must change state, state changes occur simultaneously
- Ripple counters are those in which the state change of one flip-flop triggers another flip-flop
- Reviewing table two slides ago:
  - A always toggles
  - B toggles when A $1 \rightarrow 0$ or Q'_A $0 \rightarrow 1$. Use later for $\uparrow$ clock of B
  - C toggles when B $1 \rightarrow 0$ or Q'_B $0 \rightarrow 1$. Use later for $\uparrow$ clock of C

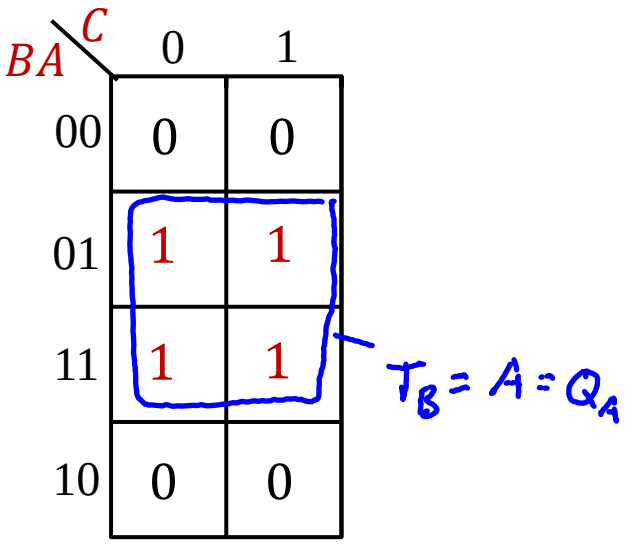


ECE2060

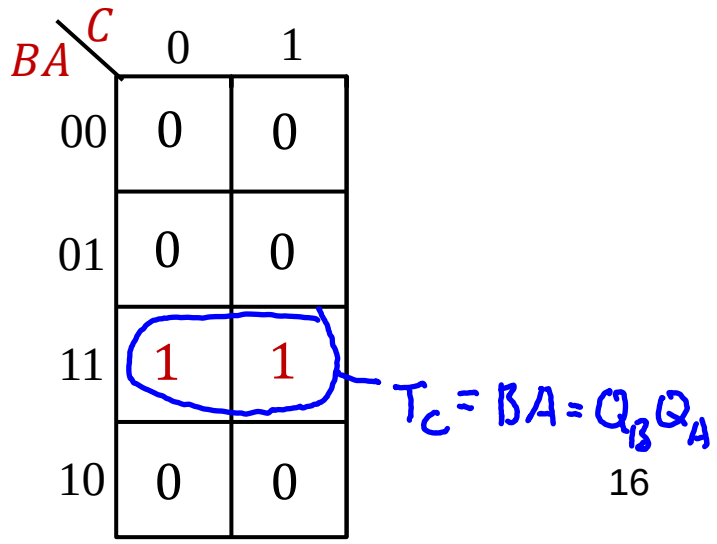
Counters

- Patterns in state table for T-implementation of the 3-bit binary counter were simple enough to recognize that we did not need K-maps
- For completeness, here are the K-maps





Present State			Next State			Flip-Flop Inputs		
C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1



Counters

- Implementation of the 3-bit binary counter using D flip-flops: $Q^+ = D$
- $D_A = A^+; D_B = B^+; D_C = C^+$
- Inspecting the transition table to fill the K-maps

Present State			Next State		
C	B	A	C^+	B^+	A^+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

BA \ C	0	1
00	1	1
01	0	0
11	0	0
10	1	1

BA \ C	0	1
00	0	0
01	1	1
11	0	0
10	1	1

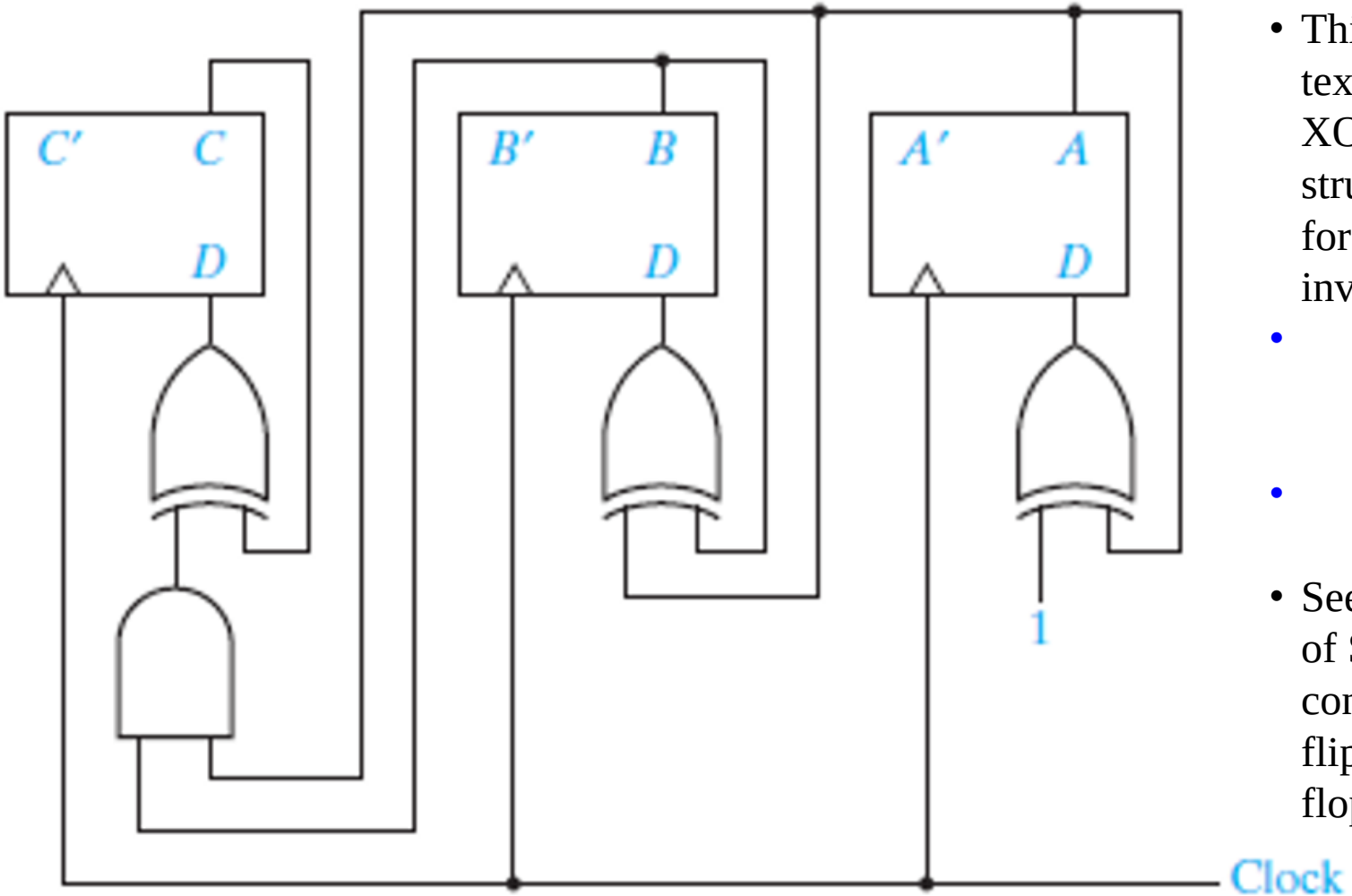
$D_B = AB' + A'B$
$= A \oplus B$

$D_C = A'C + B'C + ABC'$
$= C(A' + B') + ABC'$
$= C(AB)' + C'(AB)$
$= C \oplus AB$

BA \ C	0	1
00	0	1
01	0	1
11	1	0
10	0	1

Counters

- Implementation of the 3-bit binary counter using D flip-flops
- $D_C = C \oplus AB;$ $D_B = A \oplus B;$ $D_A = A' =$



- This figure from textbook uses same XOR input structure to D_A as for D_B and D_C to invert A
-
-
- See last paragraph of Section 11.7 for conversion of D flip-flop to T flip-flop using XOR

Counters

4-bit BCD counter using T flip-flops

D	C	B	A	D^+	C^+	B^+	A^+	T_D	T_C	T_B	T_A
0	0	0	0	0	0	0	1	0	0	0	
0	0	0	1	0	0	1	0	0	0		
0	0	1	0	0	0	1	1	0	0	0	
0	0	1	1	0	1	0	0	0			
0	1	0	0	0	1	0	1	0	0	0	
0	1	0	1	0	1	1	0	0	0		
0	1	1	0	0	1	1	1	0	0	0	
0	1	1	1	1	0	0	0				
1	0	0	0	1	0	0	1	0	0	0	
1	0	0	1	0	0	0	0		0	0	
1	0	1	0	-	-	-	-	X	X	X	X
1	0	1	1	-	-	-	-	X	X	X	X
1	1	0	0	-	-	-	-	X	X	X	X
1	1	0	1	-	-	-	-	X	X	X	X
1	1	1	0	-	-	-	-	X	X	X	X
1	1	1	1	-	-	-	-	X	X	X	X

Remember the requirement for counters with T flip-flops from last lecture:

$T_A(D, C, B, A) =$

$T_B(D, C, B, A) =$

$T_C(D, C, B, A) =$

$T_D(D, C, B, A) =$

4-bit BCD counter using T flip-flops

$$T_A(D,C,B,A) = 1$$
$$T_B(D,C,B,A) = \sum m(1,3,5,7) + \sum d(10,11,12,13,14,15)$$
$$T_C(D,C,B,A) = \sum m(3,7) + \sum d(10,11,12,13,14,15)$$
$$T_D(D,C,B,A) = \sum m(7,9) + \sum d(10,11,12,13,14,15)$$

T_D

BA \ DC	00	01	11	10
00				
01				
11				
10				

Counters

4-bit BCD counter using T flip-flops

$T_A(D,C,B,A) = 1$

$T_B(D,C,B,A) = \sum m(1,3,5,7) + \sum d(10,11,12,13,14,15)$

$T_C(D,C,B,A) = \sum m(3,7) + \sum d(10,11,12,13,14,15)$

$T_D(D,C,B,A) = \sum m(7,9) + \sum d(10,11,12,13,14,15) = AD + ABC$

The design of the rest of this BCD counter is left as homework