Digital Circuits

ECS 371

Dr. Prapun Suksompong

prapun@siit.tu.ac.th

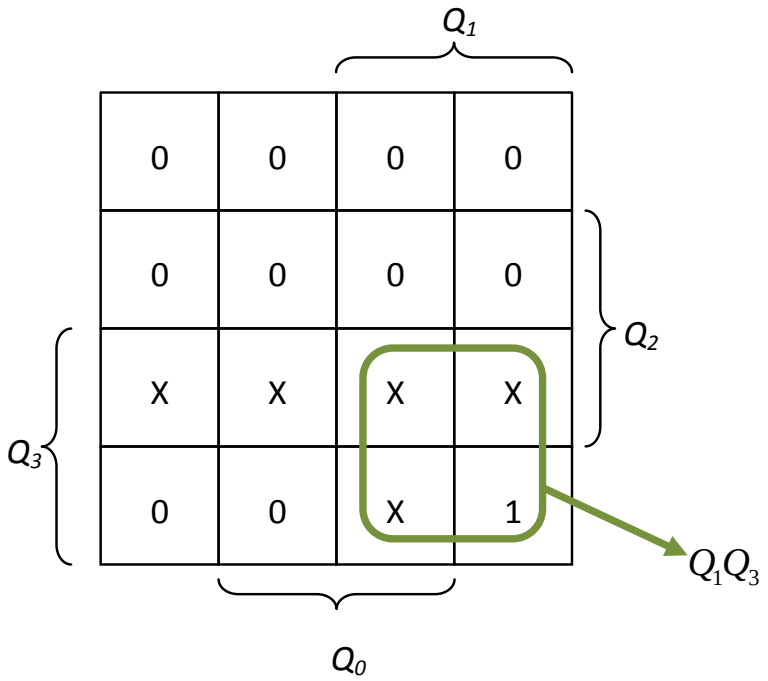
Lecture 19

Office Hours:
BKD 3601-7
Monday 9:00-10:30, 1:30-3:30
Tuesday 10:30-11:30

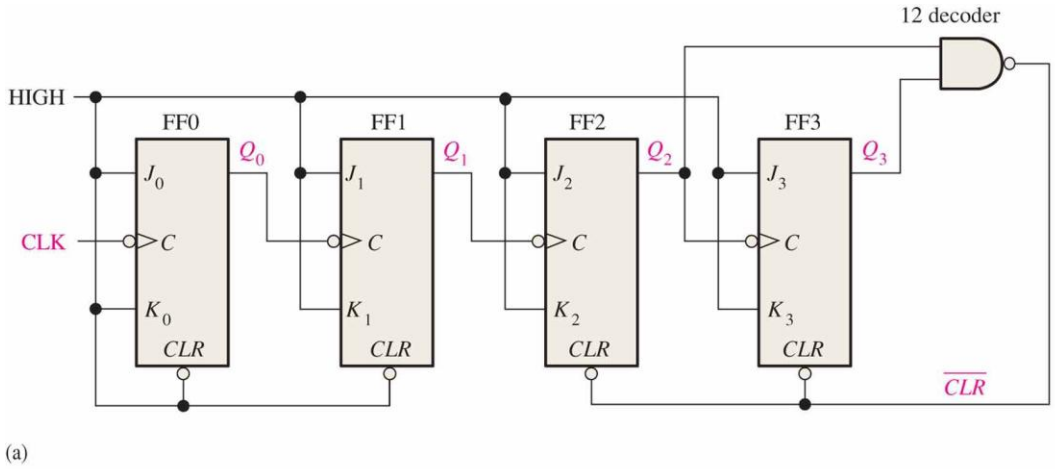
ECS371.PRAPUN.COM

K-Map Derivation of Partial Decoding

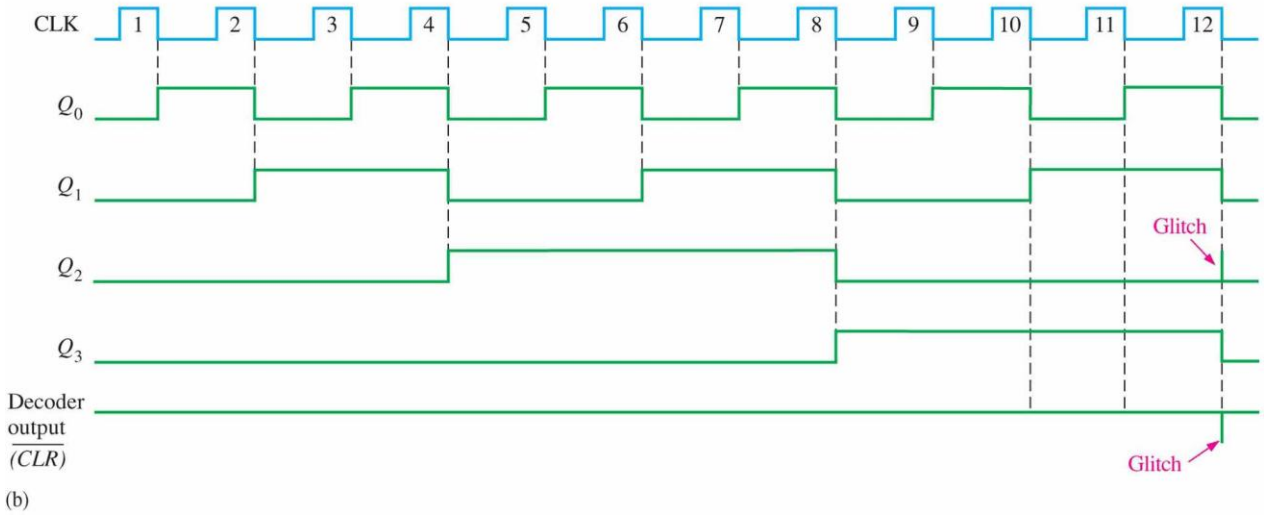
Q_3	Q_2	Q_1	Q_0	CLR
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X



Ex: MOD12 Counter



(a)

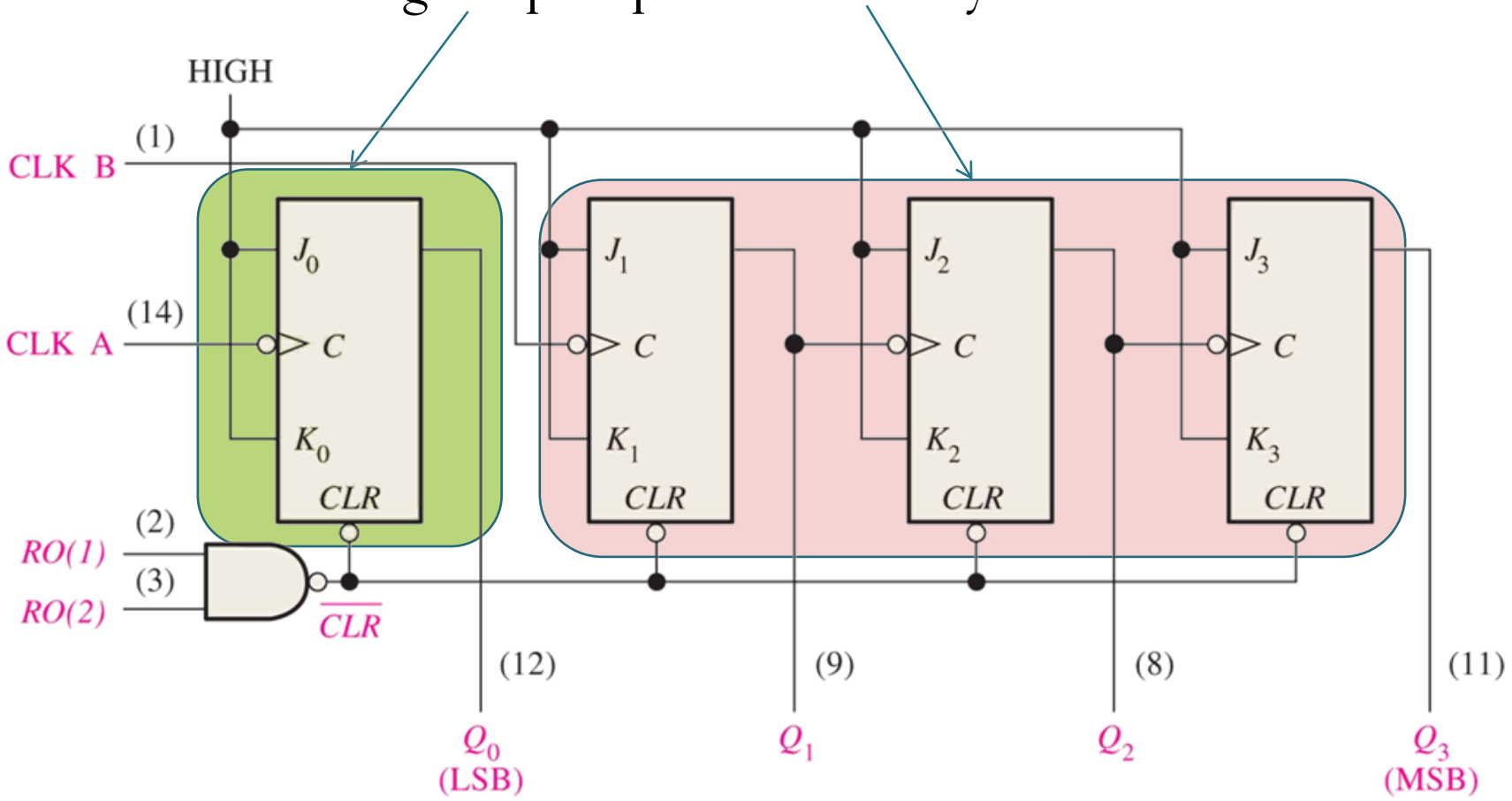


(b)

Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

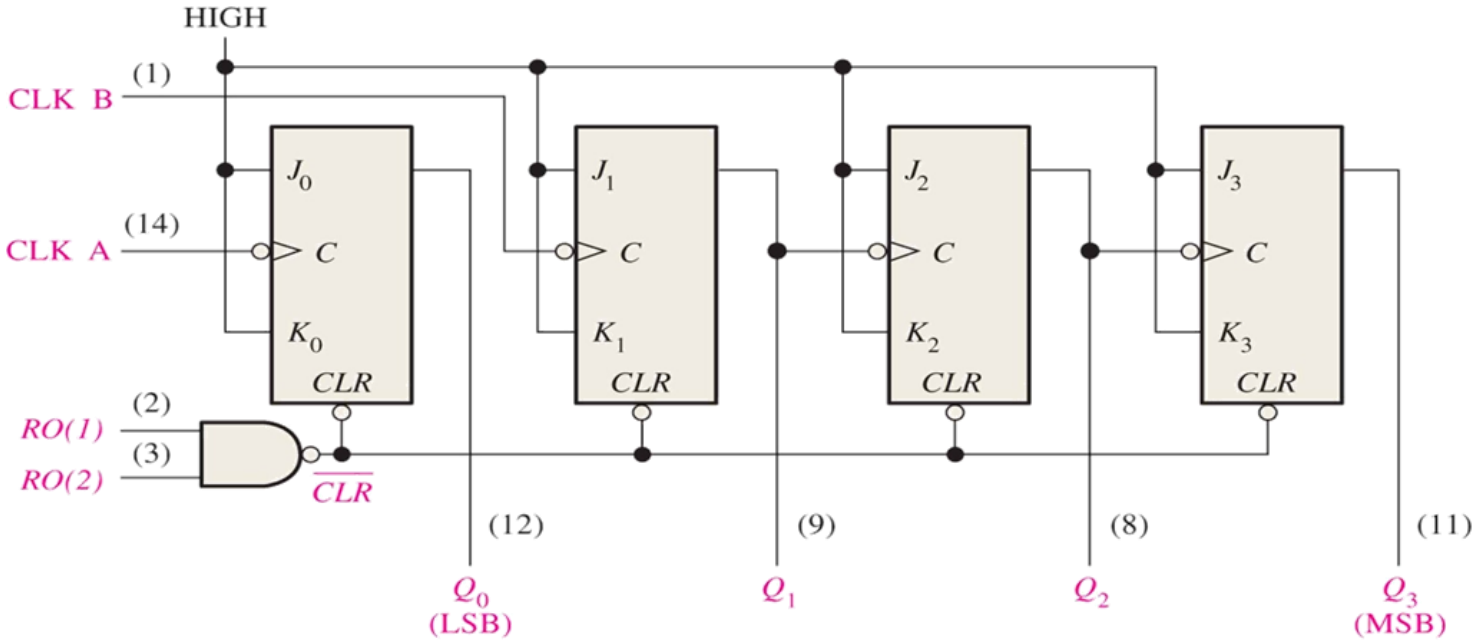
74x93

- 4-Bit Asynchronous Binary Counter
- Consist of single flip-flop and a 3-bit asynchronous counter.



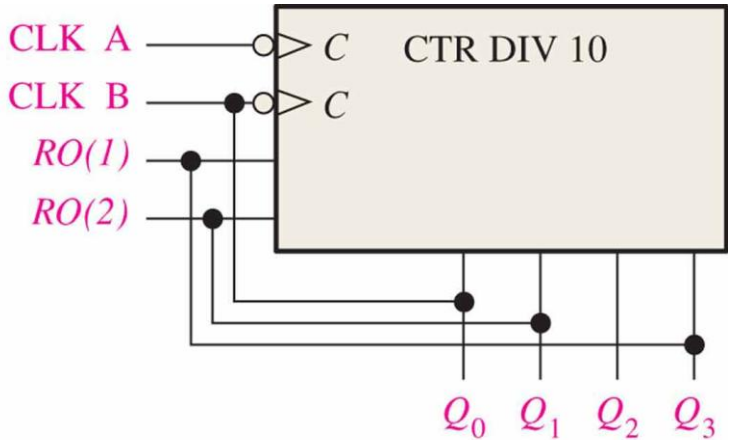
74x93

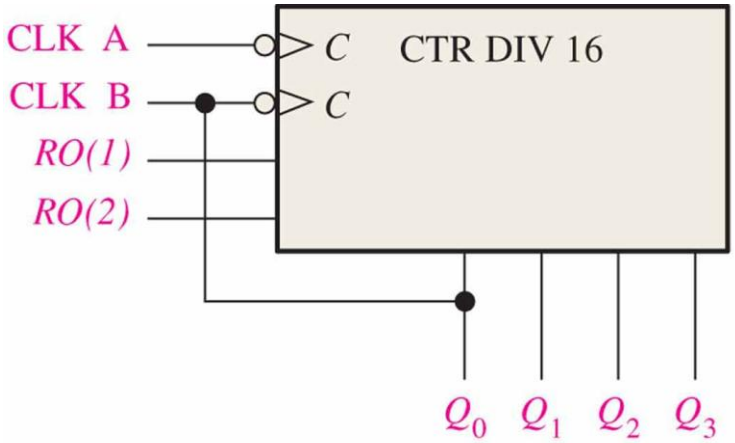
- The counter can be extended to form a 4-bit counter by connecting Q_0 to the CLK B input.
- Provide gated reset inputs, RO(1) and RO(2).
  - When both of these inputs are HIGH, the counter is reset to the 0000 state.



74x93 Applications

74x93 as a MOD10 counter





74x93 as a MOD16 counter

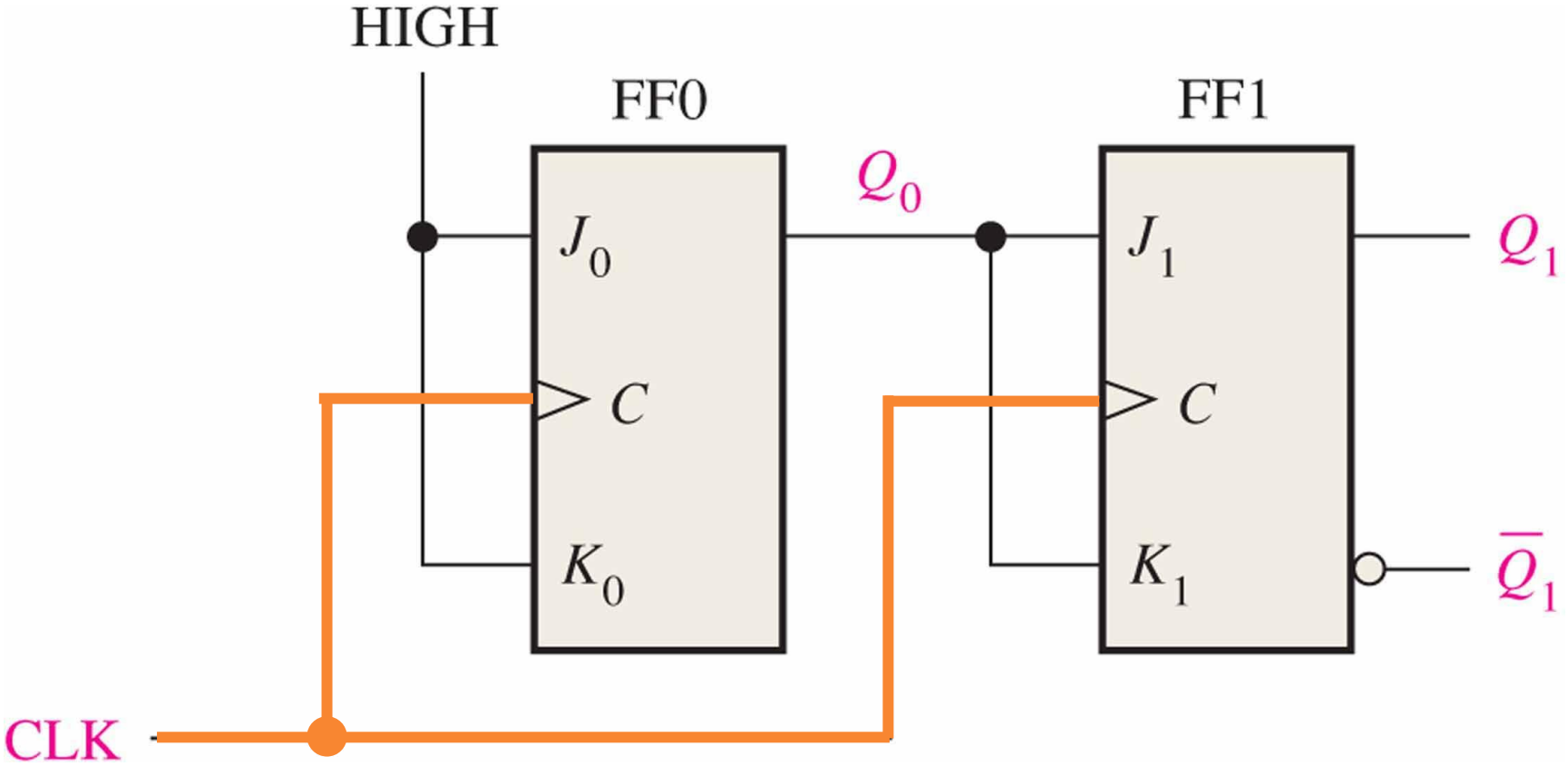
Exercise: Show how to connect a 74x93 4-bit asynchronous counter as a MOD12 counter.

The qualifying label "CTR DIV N" indicates a counter with N states.

Synchronous Counters

- The term **synchronous** refers to events that have a fixed time relationship with each other.
- A **synchronous counter** is one in which all the flip-flops in the counter are clocked at the same time by a **common** clock pulse.
- Synchronous counters overcome the disadvantage of accumulated propagation delays, but generally they require more circuitry to control states changes.

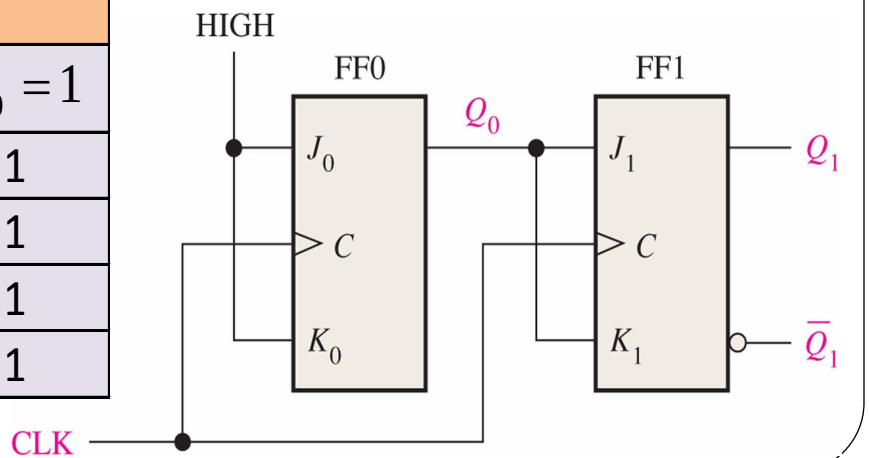
Ex: 2-Bit Synchronous Binary Counter



Summary of Analysis

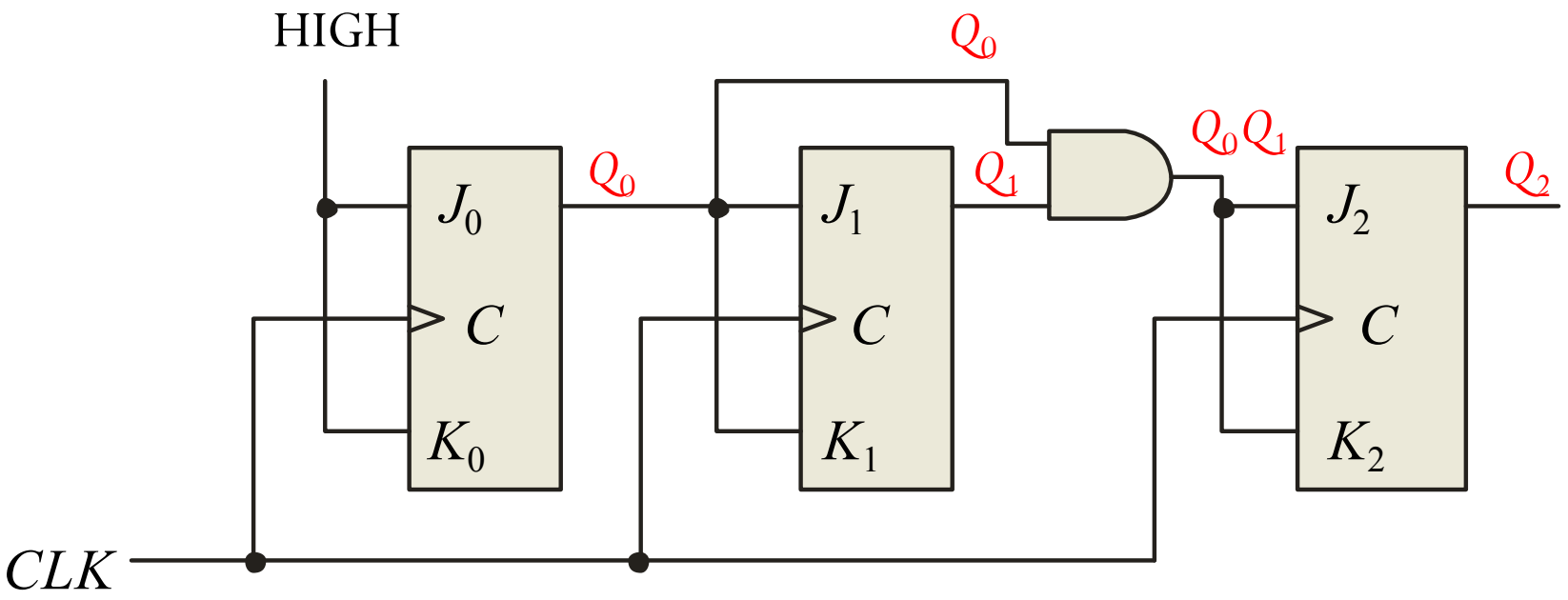
1. Find the equations that govern the relationship between the inputs (J_i, K_i) and the outputs (states: Q_i) of the FFs.
2. Start with a relevant arbitrary state (combination of outputs).
   - We will start with a state where every Q_i is 0.
3. Calculate the value of the inputs (J_i, K_i) of all FFs.
   - This tell us the modes of the FFs.
4. Find the next state (combination of outputs of the FFs) at the next clock pulse from the modes.
5. Repeat the last two steps until the counter recycles.

Outputs		J-K Inputs			
Q_1	Q_0	$J_1 = Q_0$	$K_1 = Q_0$	$J_0 = 1$	$K_0 = 1$
0	0	0	0	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	1	1	1	1	1



Ex: 3-bit Binary Synchronous Counter

- This 3-bit binary synchronous counter has the same count sequence as the 3-bit asynchronous counter shown previously.



Analysis of Synchronous Counters

A tabular technique for analysis is illustrated for the counter on the previous slide. (1) Write the logic equation for each input.

(2) Put the counter in an arbitrary state; then (3) determine the inputs for this state.	(4) Use the new inputs to determine the next state: Q_2 and Q_1 will latch and Q_0 will toggle.	(5) Set up the next group of inputs from the current output.

Outputs ← Logic for inputs →

$Q_2\ Q_1\ Q_0$	$J_2 = Q_0Q_1$	$K_2 = Q_0Q_1$	$J_1 = Q_0$	$K_1 = Q_0$	$J_0 = 1$	$K_0 = 1$
0 0 0	0	0	0	0	1	1
0 0 1	0	0	1	1	1	1
0 1 0						

(6) Q_2 will latch again but both Q_1 and Q_0 will toggle.

(7) Continue like this, to complete the table. The next slide shows the completed table…

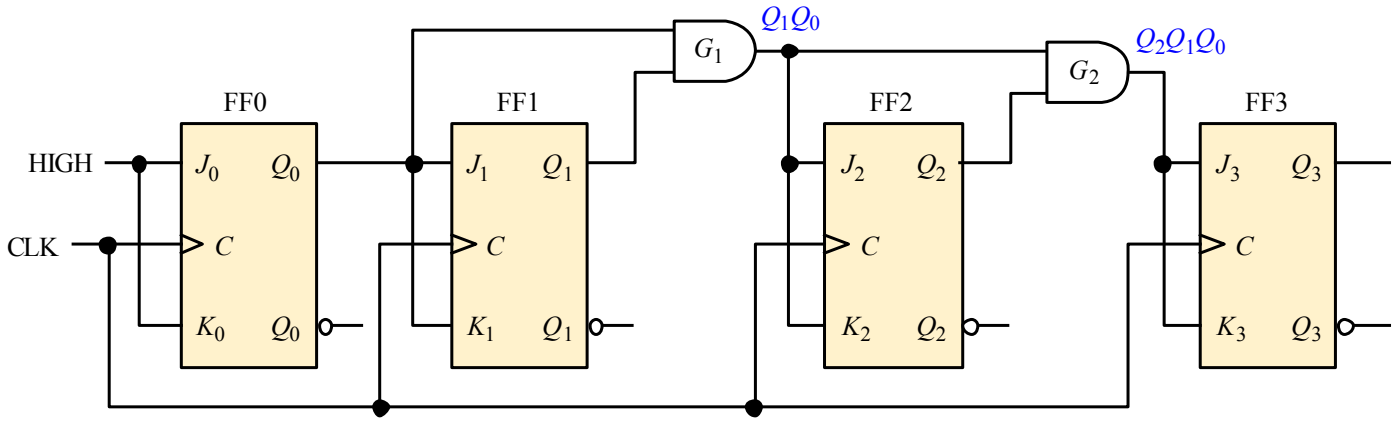
Analysis of Synchronous Counters

Outputs ← Logic for inputs →

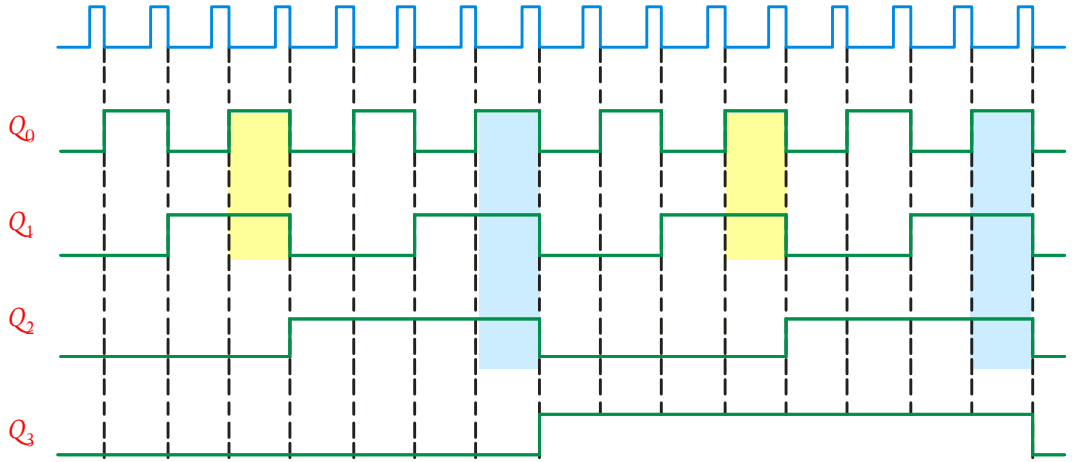
$Q_2\ Q_1\ Q_0$	$J_2 = Q_0Q_1$	$K_2 = Q_0Q_1$	$J_1 = Q_0$	$K_1 = Q_0$	$J_0 = 1$	$K_0 = 1$
0 0 0	0	0	0	0	1	1
0 0 1	0	0	1	1	1	1
0 1 0	0	0	0	0	1	1
0 1 1	1	1	1	1	1	1
1 0 0	0	0	0	0	1	1
1 0 1	0	0	1	1	1	1
1 1 0	0	0	0	0	1	1
1 1 1	1	1	1	1	1	1
0 0 0						

At this point all states have been accounted for and the counter is ready to recycle…

4-bit Synchronous Binary Counter

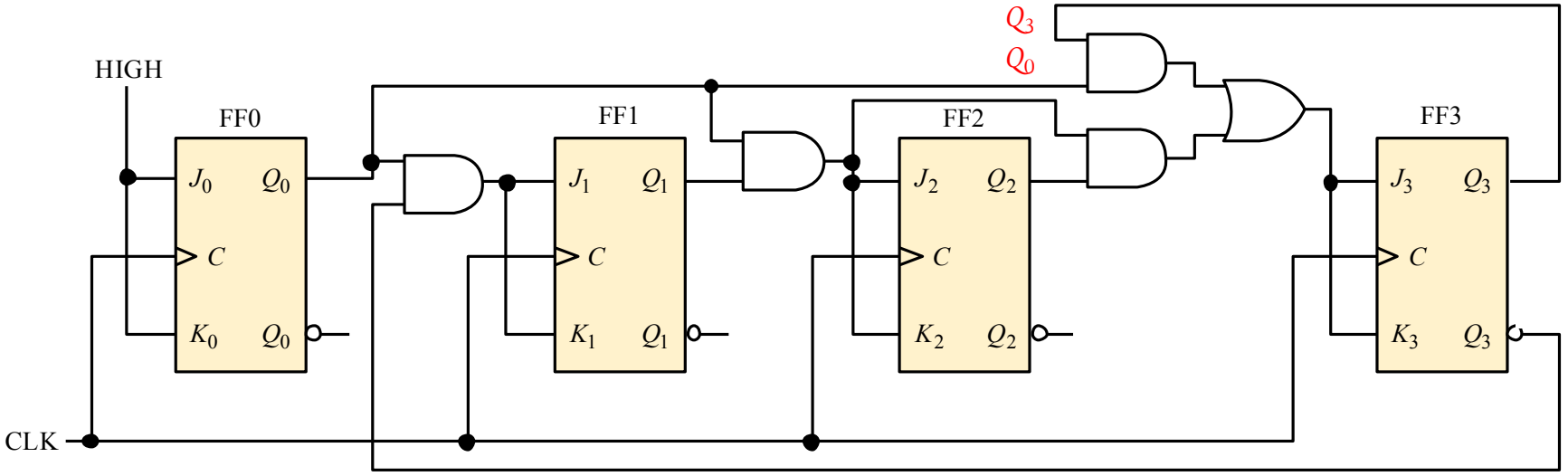


The 4-bit binary counter has one more AND gate than the 3-bit counter just described. The shaded areas show where the AND gate outputs are HIGH causing the next FF to toggle.



BCD Decade Counter

- With some additional logic, a binary counter can be converted to a BCD synchronous decade counter.
- After reaching the count 1001, the counter recycles to 0000.

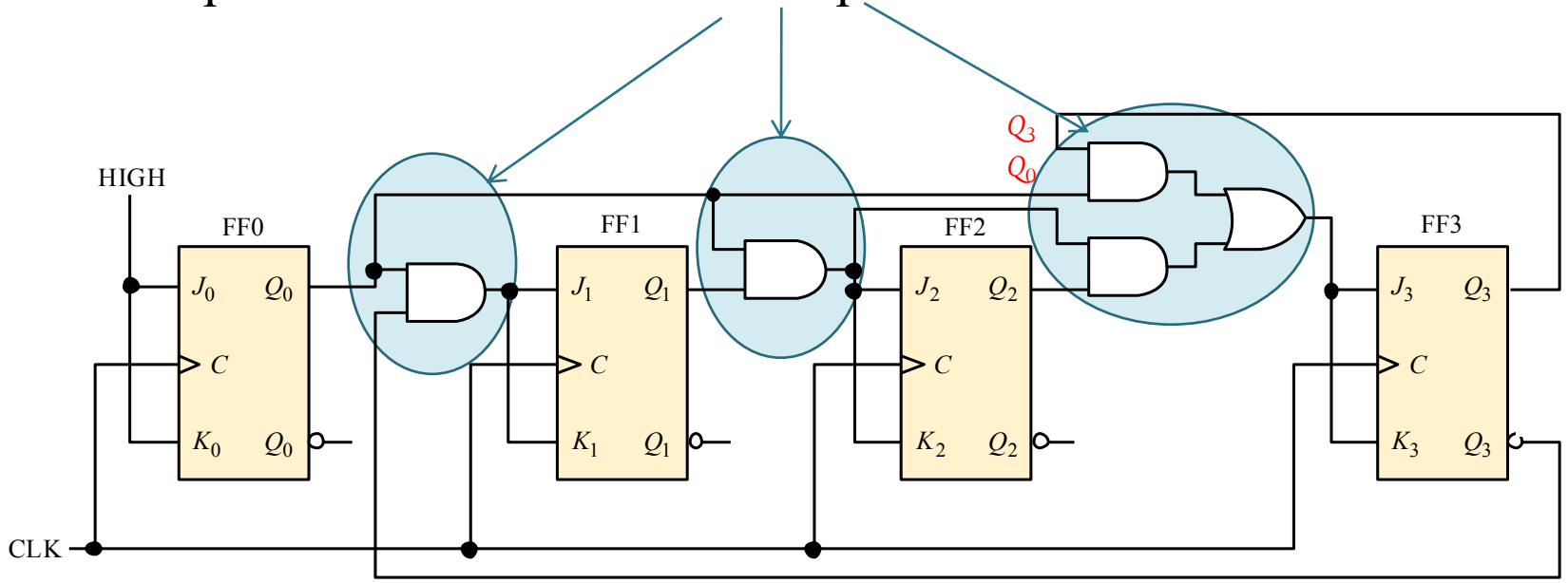


BCD Decade Counter: Tabular Analysis

Q_3	Q_2	Q_1	Q_0	$J_3 = K_3 = Q_0Q_3 + Q_0Q_1Q_2$	$J_2 = K_2 = Q_0Q_1$	$J_1 = K_1 = Q_0\overline{Q_3}$	$J_0 = K_0 = 1$
0	0	0	0				

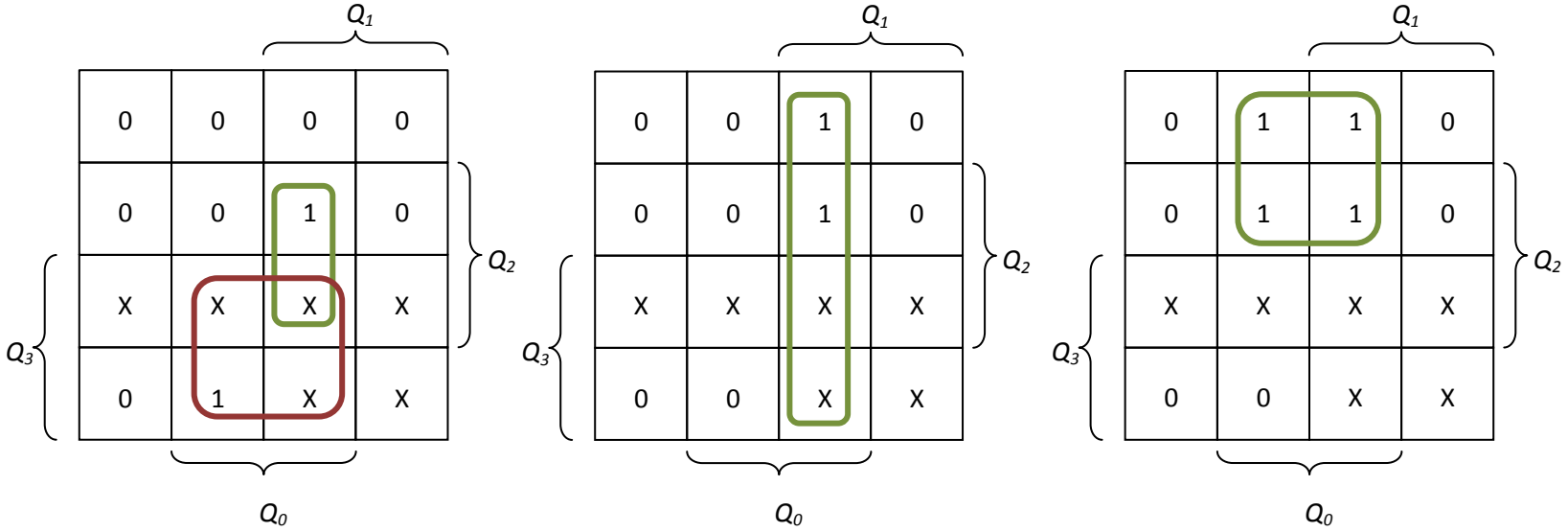
BCD Decade Counter Design

- The previous slide shows that the given design works.
- How can we get it in the first place?
- In particular, where do these parts come from?



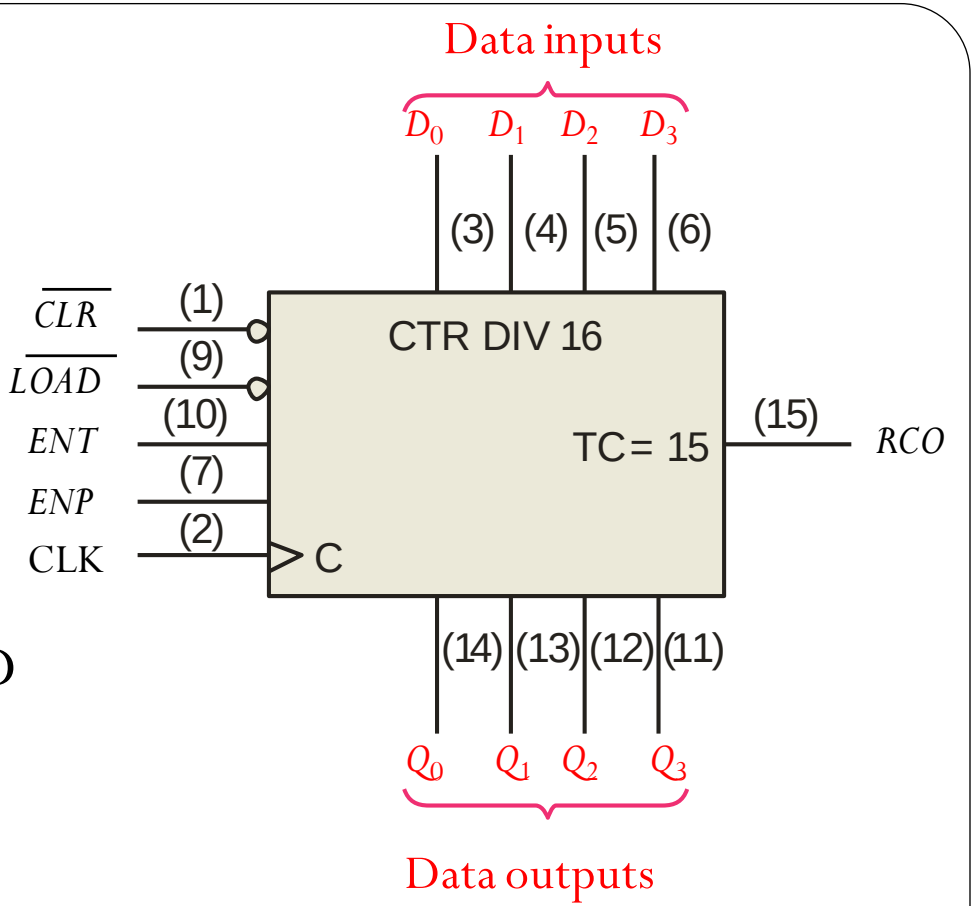
BCD Decade Counter: Derivation

Q_3	Q_2	Q_1	Q_0	$J_3 = K_3$	$J_2 = K_2$	$J_1 = K_1$	$J_0 = K_0$
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	1
0	0	1	0	0	0	0	1
0	0	1	1	0	1	1	1
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	1
0	1	1	0	0	0	0	1
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1
1	0	0	1	1	0	0	1
0	0	0	0				



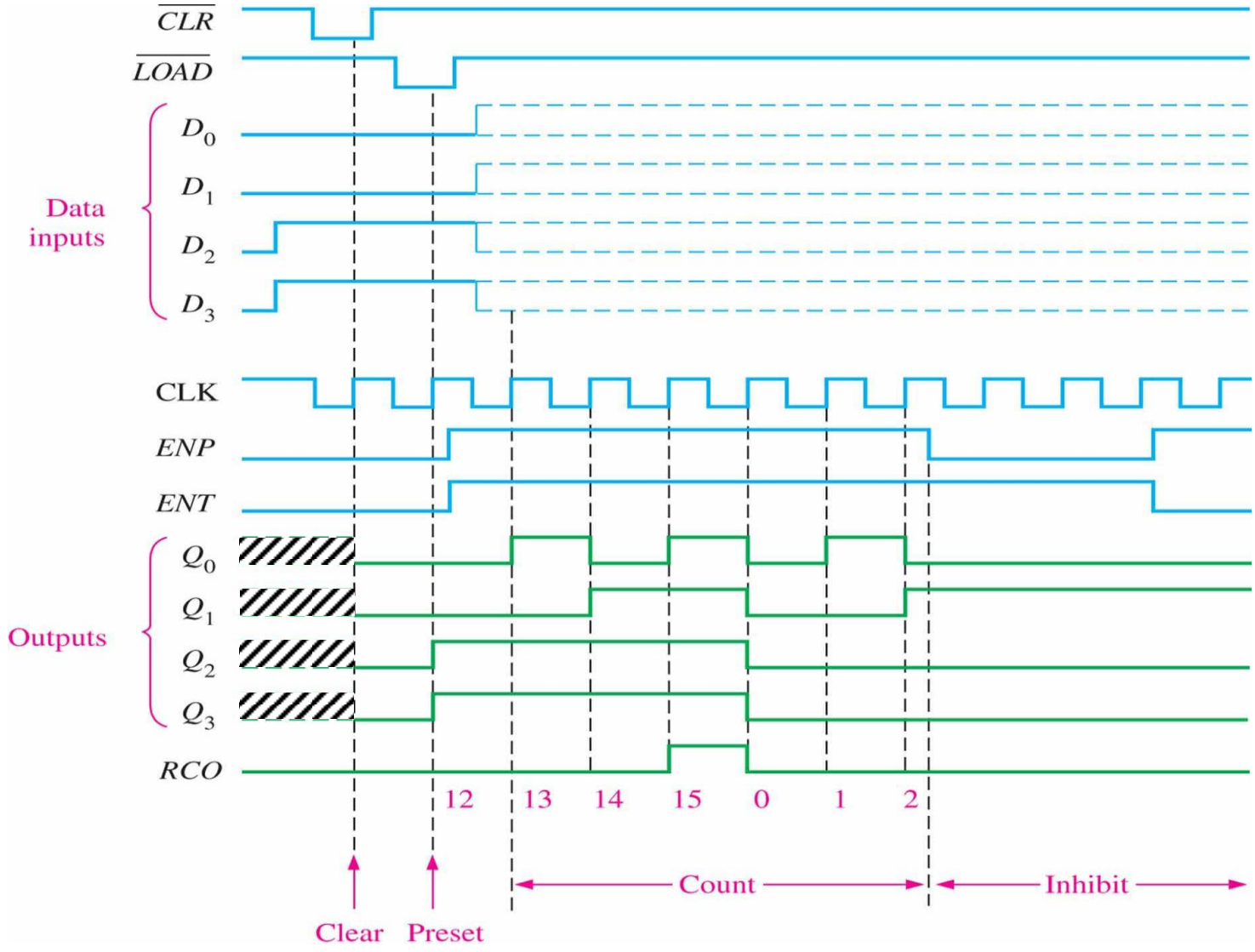
74x163

- 4-bit Synchronous Binary Counter
- Has several extra features
- Can be *synchronously preset* to any 4-bit binary number when the active-LOW LOAD input is activated.
- The active-LOW clear input (CLR) *synchronously resets* all four FFs.
- The two enable inputs, ENP and ENT must both be HIGH for the counter to sequence through its binary states.

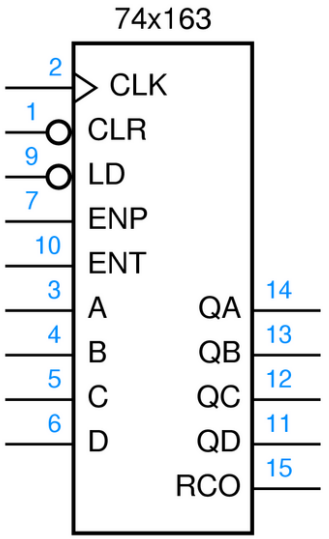


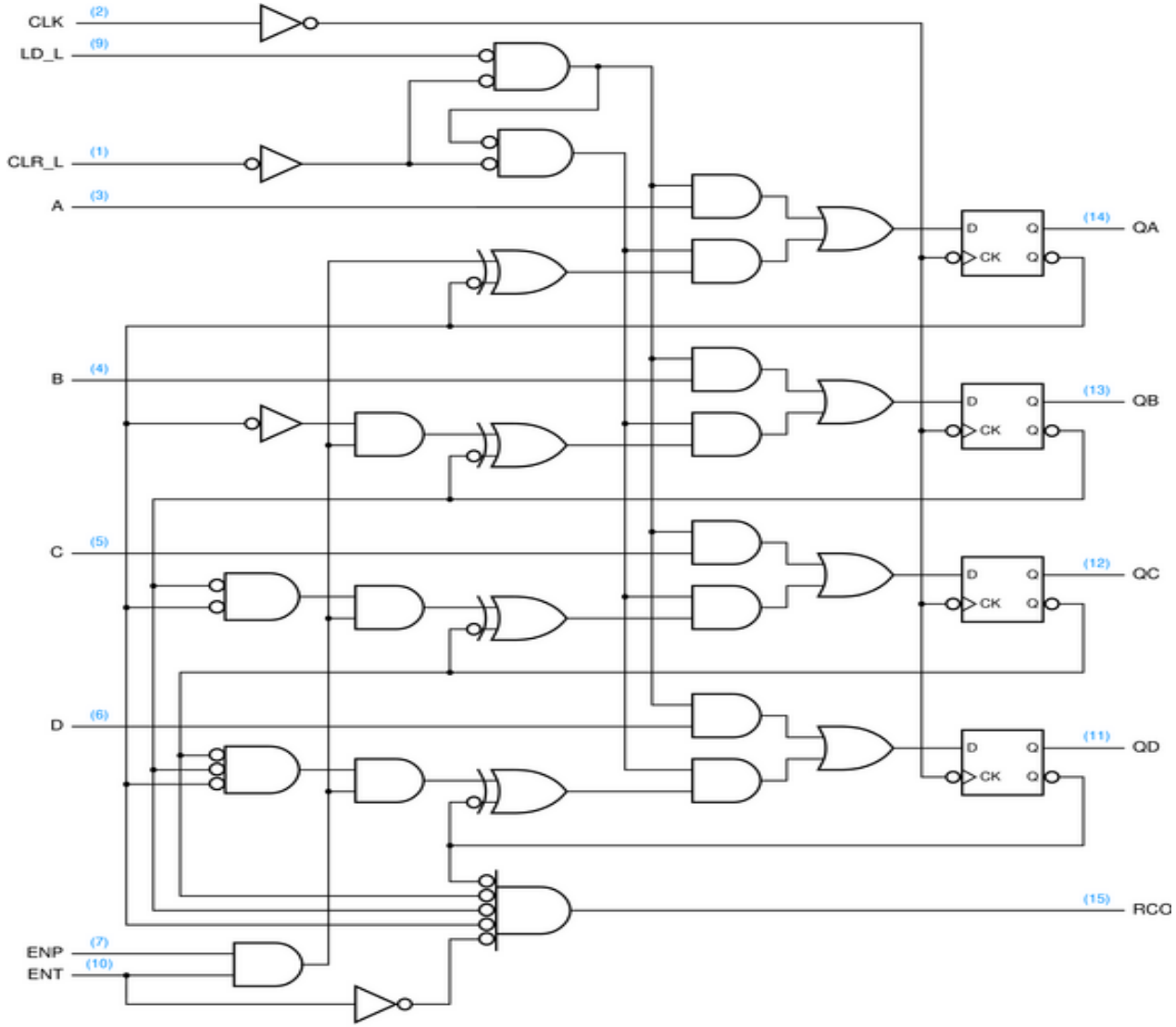
- The ripple clock output (RCO) goes HIGH when the counter reaches the last state in its sequence of fifteen, called the terminal count (TC = 15).

Example: 74x163



74x163: Logic Diagram





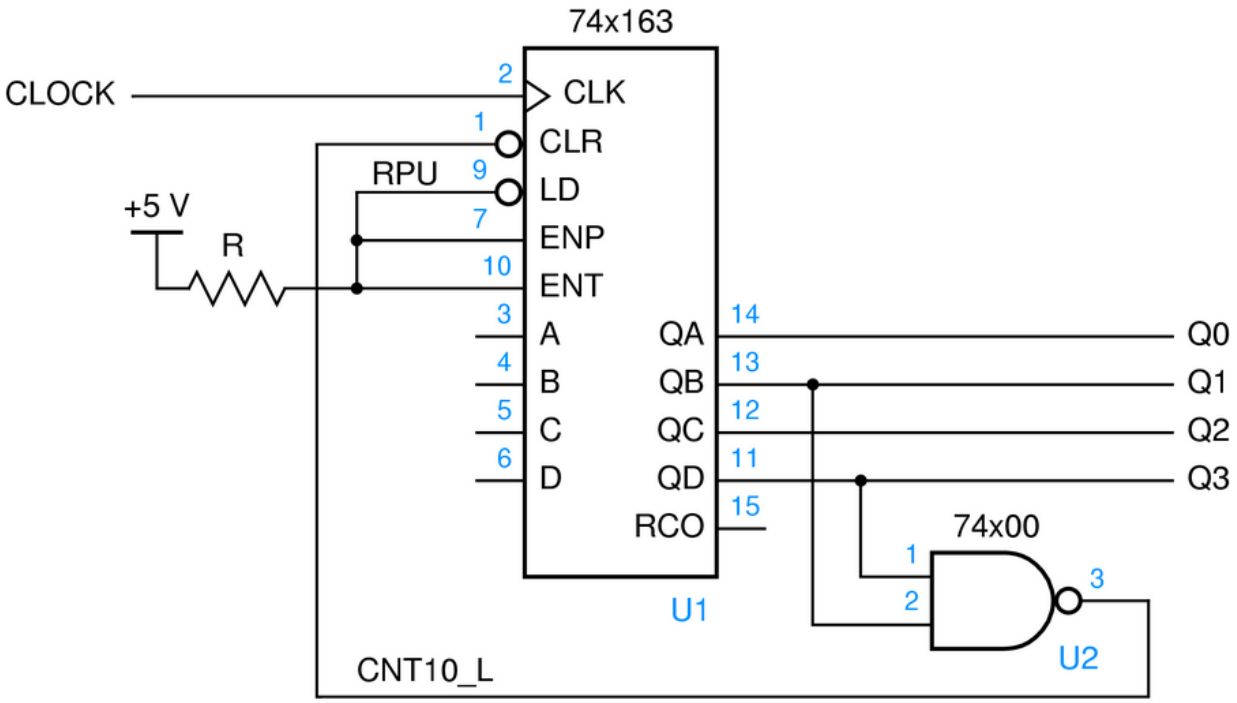
74x163: State Table

State table for a 74163 synchronous 4-bit binary counter

Inputs				Current State				Outputs				
CLR_L	LD_L	ENT	ENP	QD	QC	QB	QA	QD*	QC*	QB*	QA*	RCO*
0	x	x	x	x	x	x	x	0	0	0	0	
1	0	x	x	x	x	x	x	D	C	B	A	
1	1	0	x	x	x	x	x	QD	QC	QB	QA	
1	1	x	0	x	x	x	x	QD	QC	QB	QA	
1	1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	1	0	0	1	0	0
1	1	1	1	0	0	1	0	0	0	1	1	0
1	1	1	1	0	0	1	1	0	1	0	0	0
1	1	1	1	0	1	0	0	0	1	0	1	0
1	1	1	1	0	1	0	1	0	1	1	0	0
1	1	1	1	0	1	1	0	0	1	1	1	0
1	1	1	1	0	1	1	1	1	0	0	0	0
1	1	1	1	1	0	0	0	1	0	0	1	0
1	1	1	1	1	0	0	1	1	0	1	0	0
1	1	1	1	1	0	1	0	1	0	1	1	0
1	1	1	1	1	0	1	1	1	1	0	0	0
1	1	1	1	1	1	0	0	1	1	0	1	0
1	1	1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	0

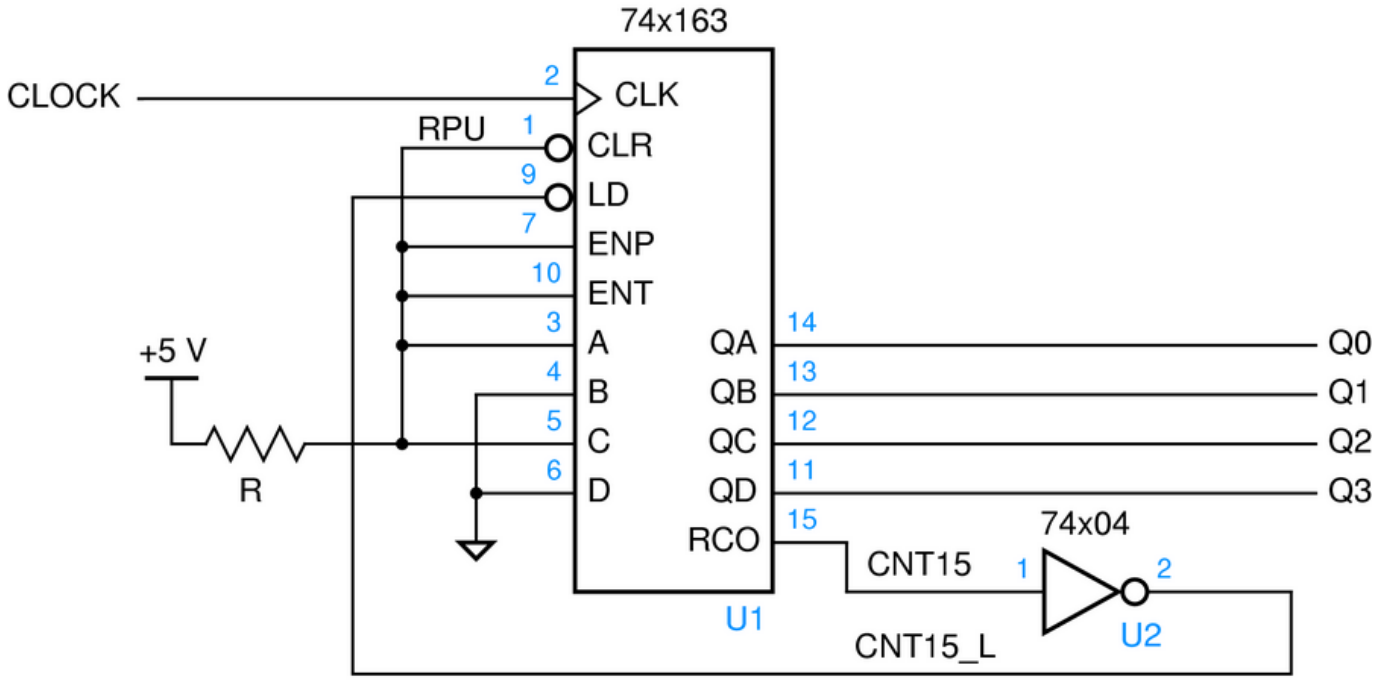
74x163: Application

- Count from 0 to 10. Then, repeat.



74x163: Application

- Count from 5 to 15. Then, repeat.



Exercise

What will this counter do?