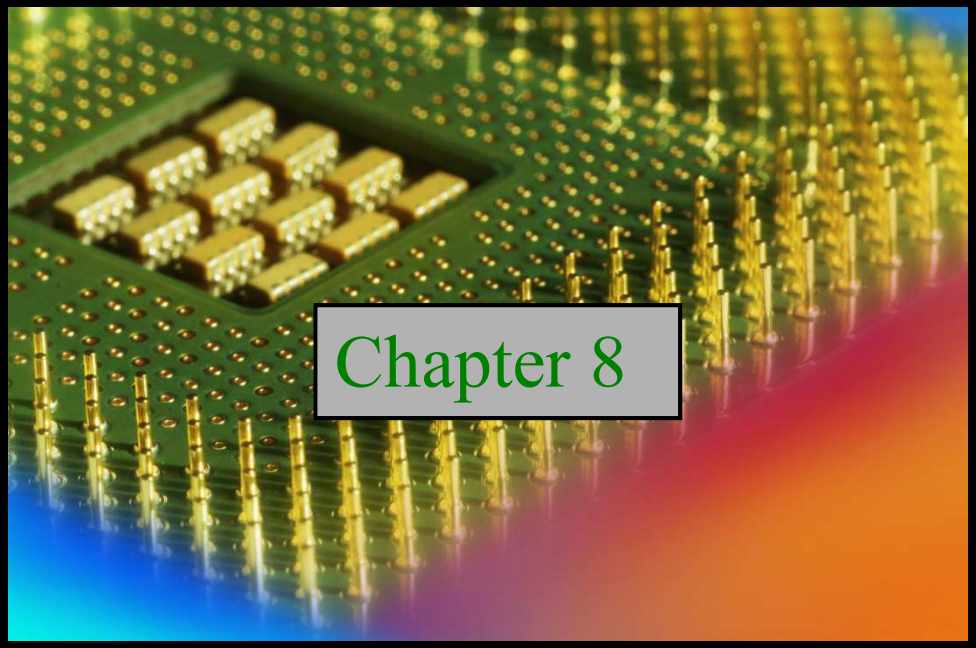
Digital Fundamentals

Tenth Edition

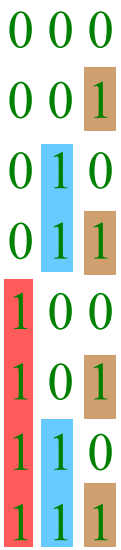
Floyd



Summary

Counting in Binary

As you know, the binary count sequence follows a familiar pattern of 0's and 1's as described in Section 2-2 of the text.



LSB changes on every number.

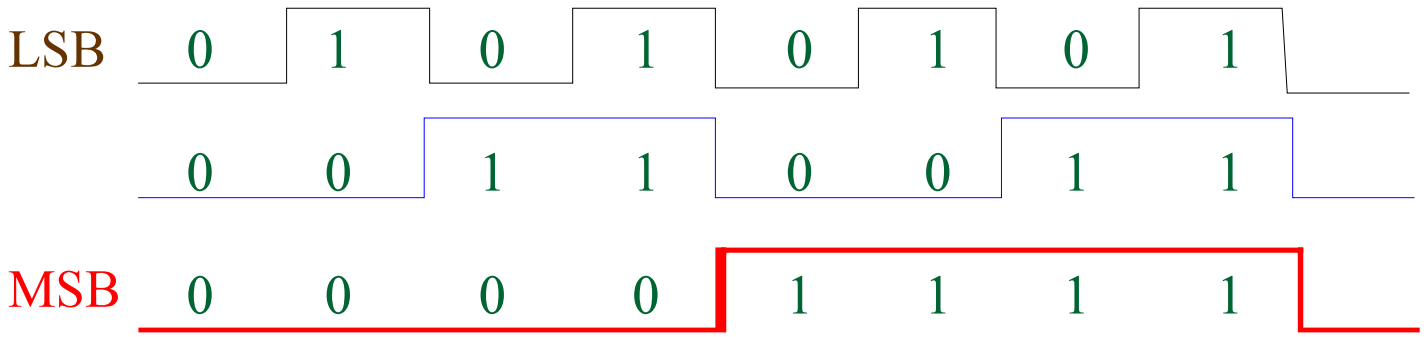
The next bit changes on every other number.

The next bit changes on every fourth number.

Summary

Counting in Binary

A counter can form the same pattern of 0's and 1's with logic levels. The first stage in the counter represents the least significant bit – notice that these waveforms follow the same pattern as counting in binary.

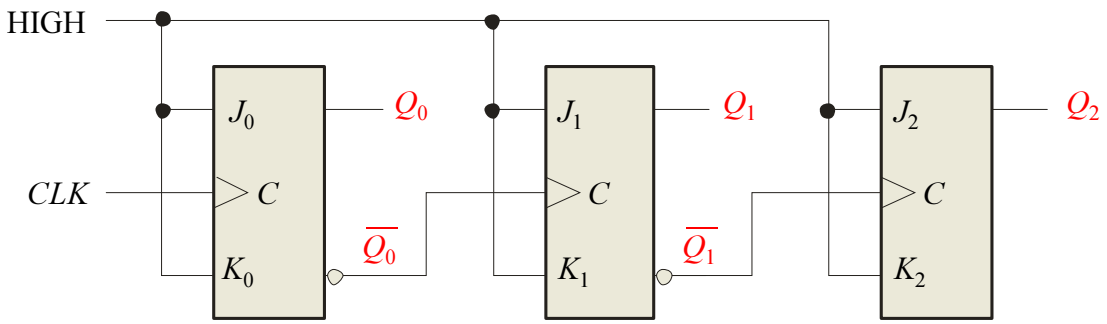


Summary

Three bit Asynchronous Counter

In an asynchronous counter, the clock is applied only to the first stage. Subsequent stages derive the clock from the previous stage.

The three-bit asynchronous counter shown is typical. It uses J-K flip-flops in the toggle mode.



Waveforms are on the following slide…

Summary

Three bit Asynchronous Counter

Notice that the Q_0 output is triggered on the leading edge of the clock signal. The following stage is triggered from $\overline{Q_0}$. The leading edge of $\overline{Q_0}$ is equivalent to the trailing edge of Q_0. The resulting sequence is that of an 3-bit binary up counter.

CLK

Q_0

Q_1

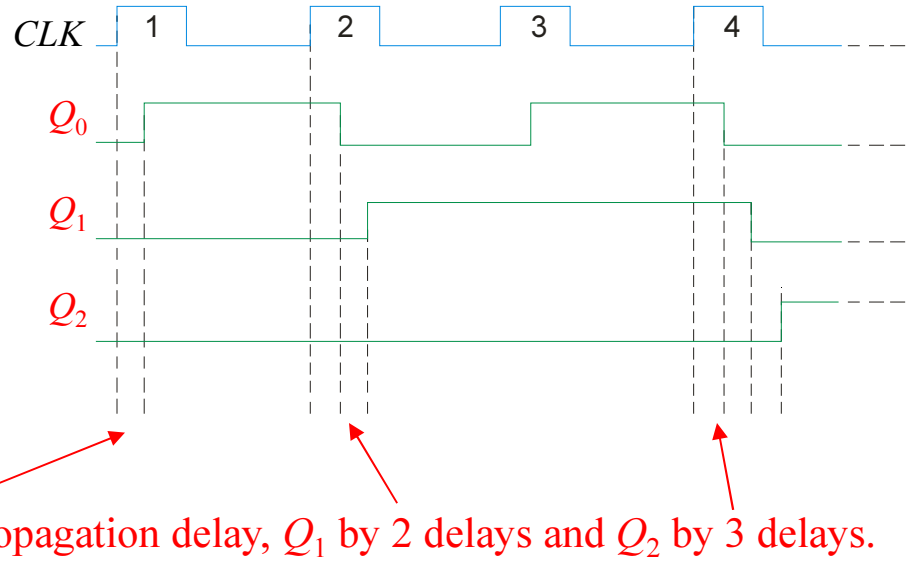
Q_2

Summary

Propagation Delay

Asynchronous counters are sometimes called **ripple** counters, because the stages do not all change together. For certain applications requiring high clock rates, this is a major disadvantage.

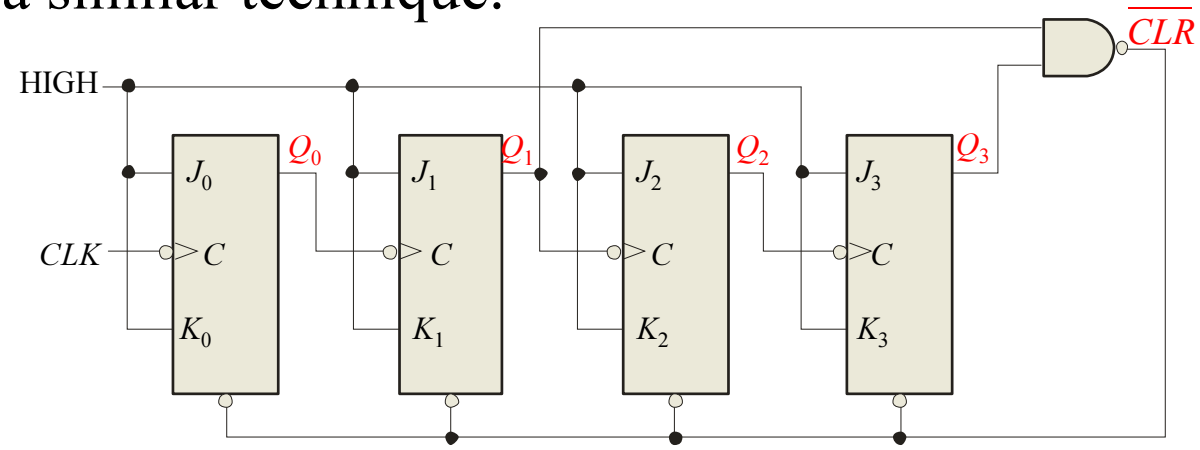
Notice how delays are cumulative as each stage in a counter is clocked later than the previous stage.



Summary

Asynchronous Decade Counter

This counter uses partial decoding to recycle the count sequence to zero after the 1001 state. The flip-flops are trailing-edge triggered, so clocks are derived from the *Q* outputs. Other truncated sequences can be obtained using a similar technique.



Waveforms are on the following slide…

Summary

Asynchronous Decade Counter

When Q_1 and Q_3 are HIGH together, the counter is cleared by a "glitch" on the $\overline{CLR}$ line.

CLK

Q_0

Q_1

Q_2

Q_3

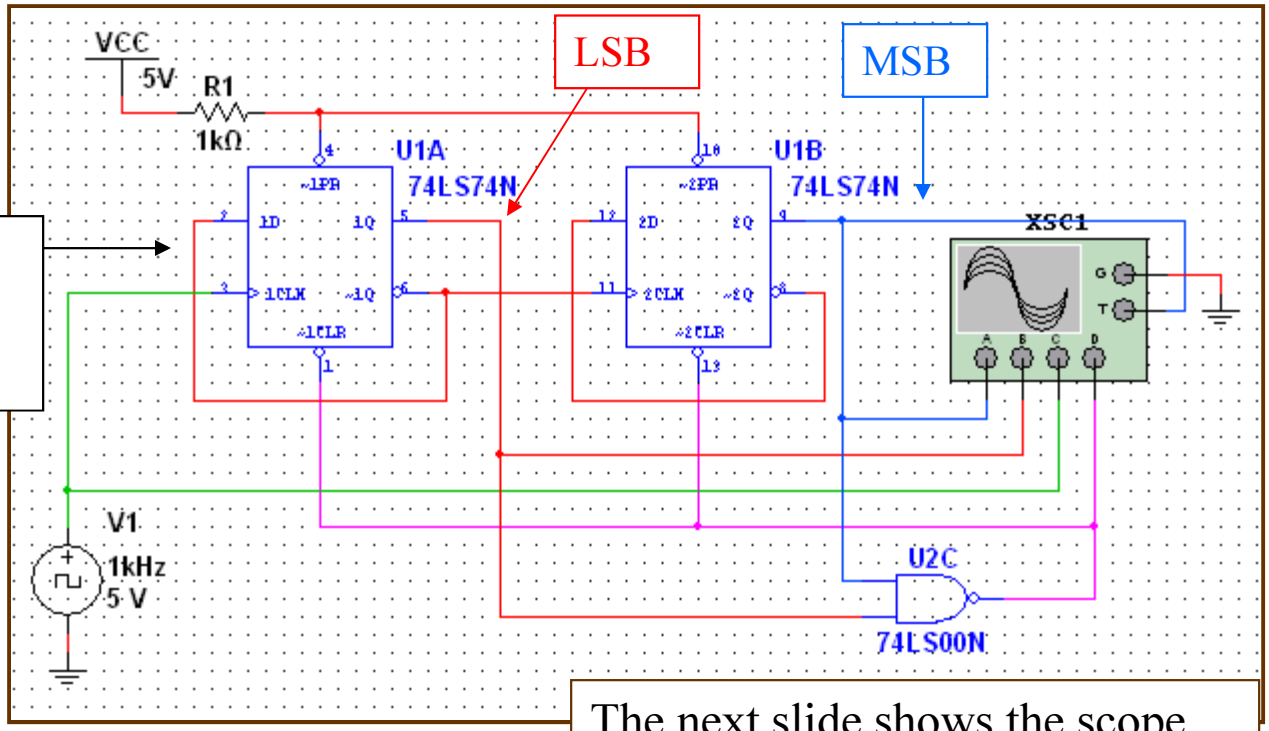
$\overline{CLR}$

Summary

Asynchronous Counter Using D Flip-flops

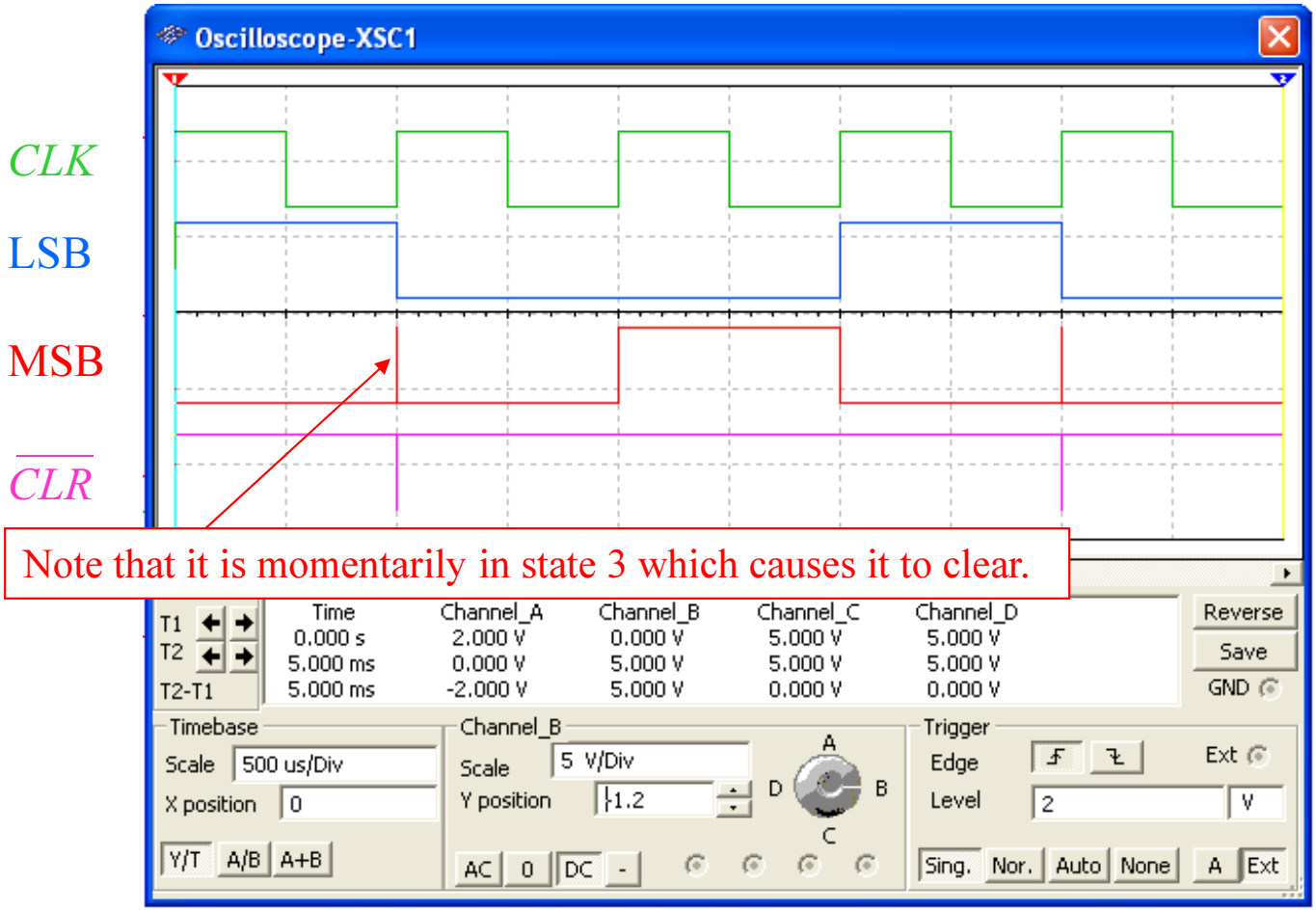
D flip-flops can be set to toggle and used as asynchronous counters by connecting $\overline{Q}$ back to D. The counter in this slide is a Multisim simulation of one described in the lab manual. Can you figure out the sequence?

$\overline{Q}$ to D puts D flip-flop in toggle mode



The next slide shows the scope…

Summary



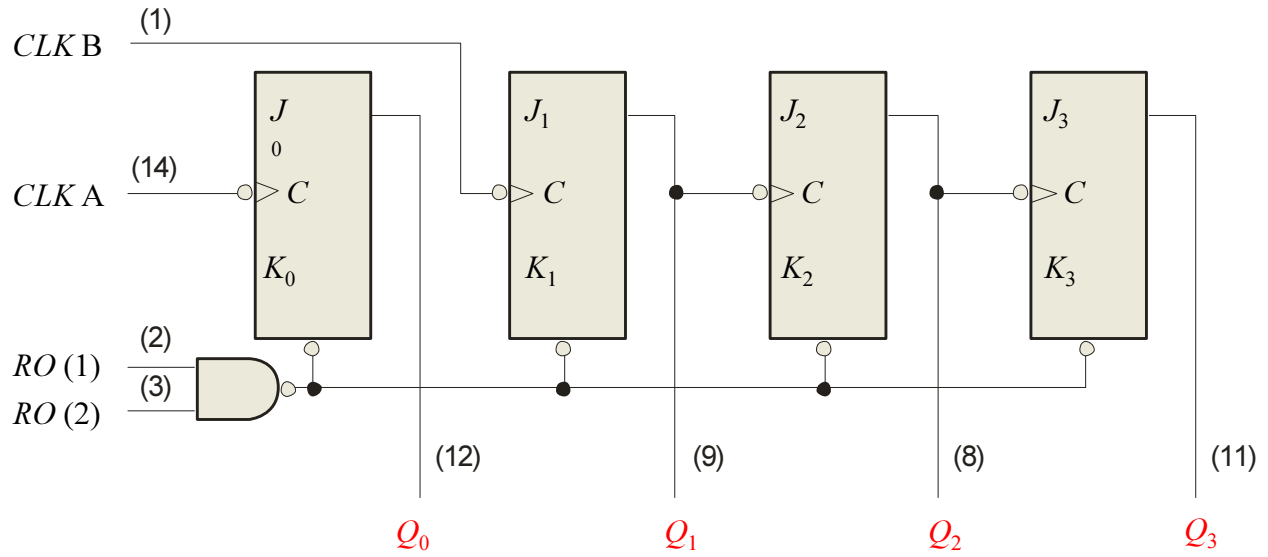
The sequence is 0 – 2 – 1 – ($\overline{CLR}$) (repeat)…

Summary

The 74LS93A Asynchronous Counter

The 74LS93A has one independent toggle J-K flip-flop driven by *CLK* A and three toggle J-K flip-flops that form an asynchronous counter driven by *CLK* B.

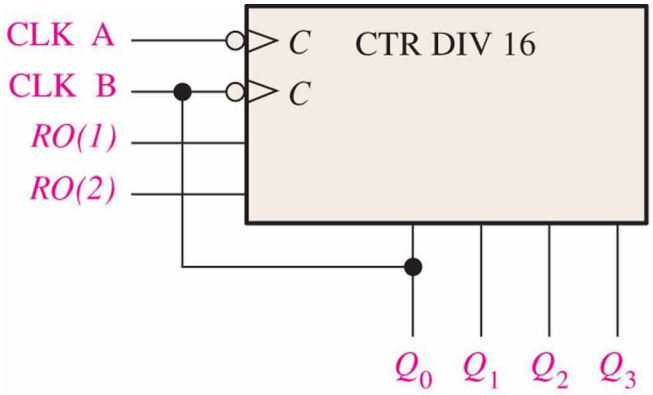
The counter can be extended to form a 4-bit counter by connecting Q_0 to the *CLK* B input. Two inputs are provided that clear the count.



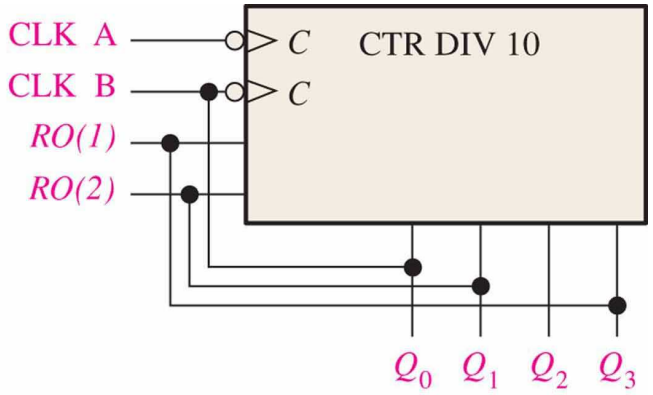
All J and K inputs are connected internally HIGH

Summary

The 74LS93A Asynchronous Counter



(a) 74LS93 connected as a modulus-16 counter

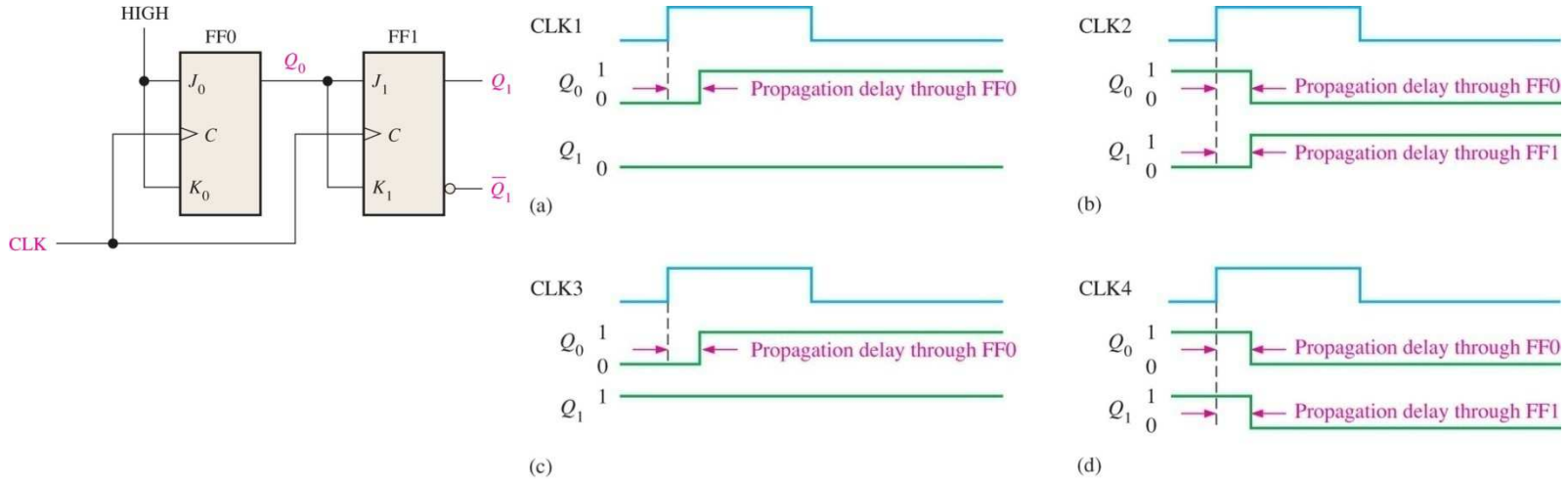


(b) 74LS93 connected as a decade counter

Summary

Synchronous Counters

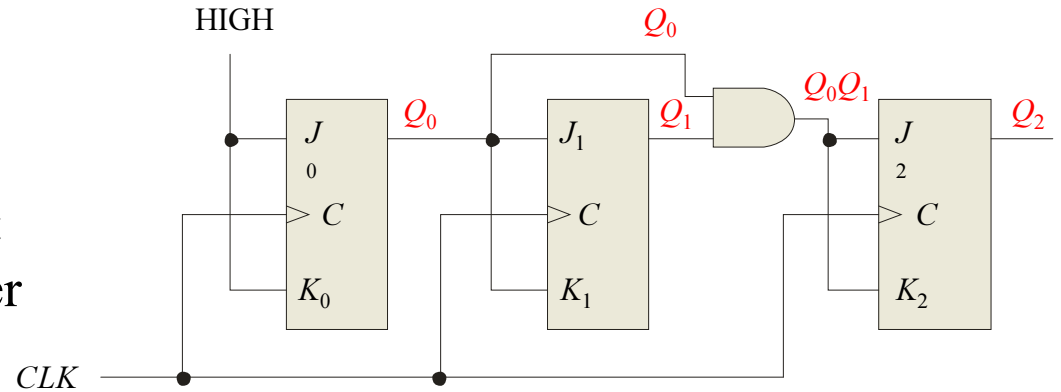
In a **synchronous counter** all flip-flops are clocked together with a common clock pulse. Synchronous counters overcome the disadvantage of accumulated propagation delays, but generally they require more circuitry to control states changes.



Summary

Synchronous Counters

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



The next slide shows how to analyze this counter by writing the logic equations for each input. Notice the inputs to each flip-flop…

Summary

Analysis of Synchronous Counters

A tabular technique for analysis is illustrated for the counter on the previous slide. Start by setting up the outputs as shown, then write the logic equation for each input. This has been done for the counter.

1. Put the counter in an arbitrary state; then determine the inputs for this state.	2. Use the new inputs to determine the next state: Q_2 and Q_1 will latch and Q_0 will toggle.	3. 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						

4. Q_2 will latch again but both Q_1 and Q_0 will toggle.

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

Summary

Analysis of Synchronous Counters

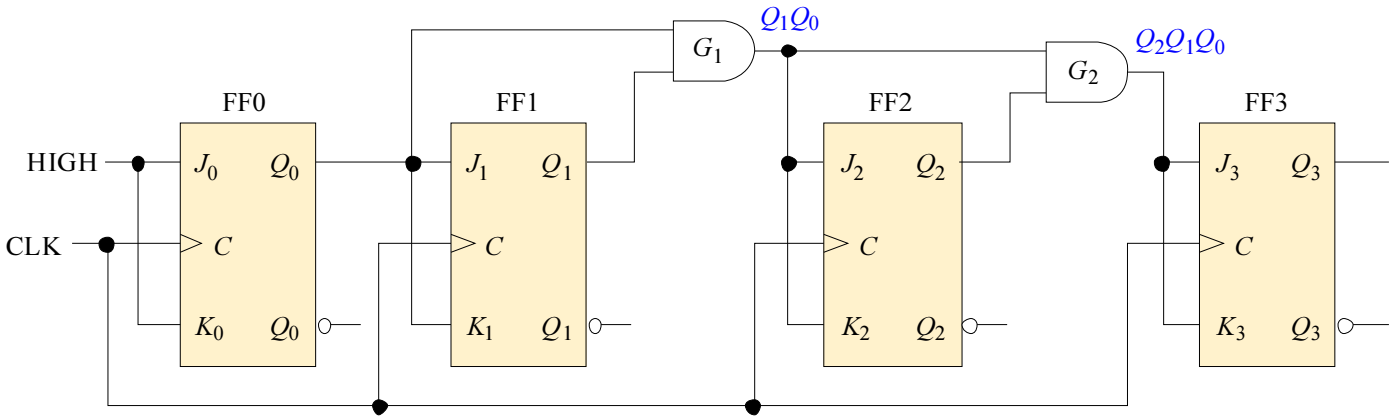
Outputs ← Logic for inputs →

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

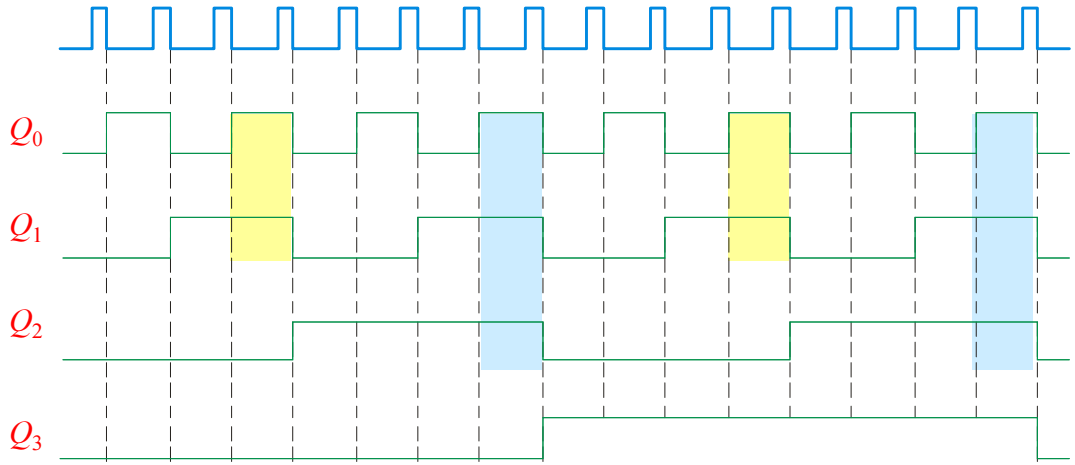
At this points all states have been accounted for and the counter is ready to recycle…

Summary

A 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.

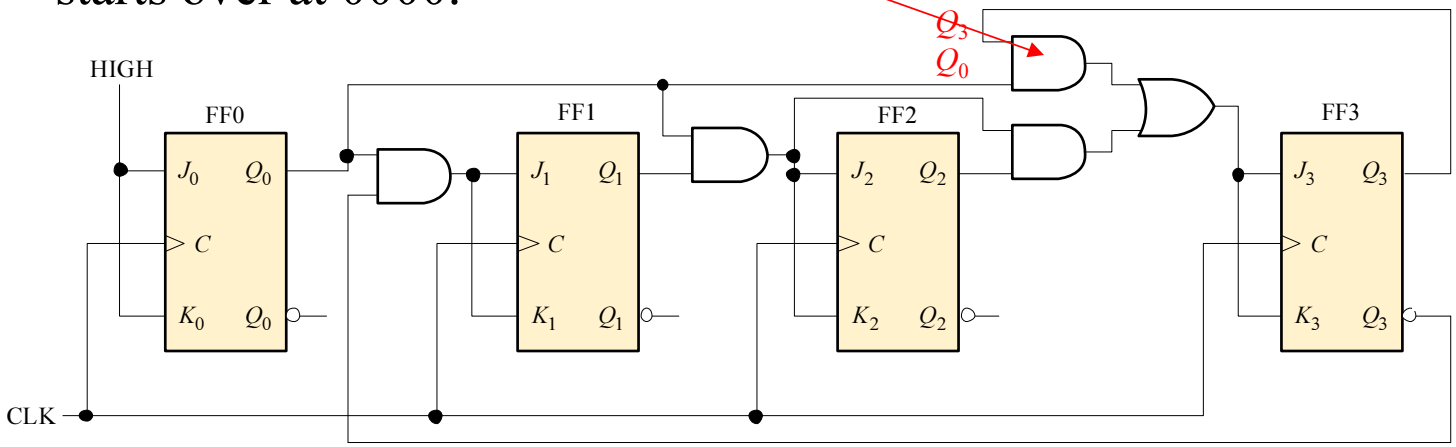


Summary

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.

This gate detects 1001, and causes FF3 to toggle on the next clock pulse. FF0 toggles on every clock pulse. Thus, the count starts over at 0000.

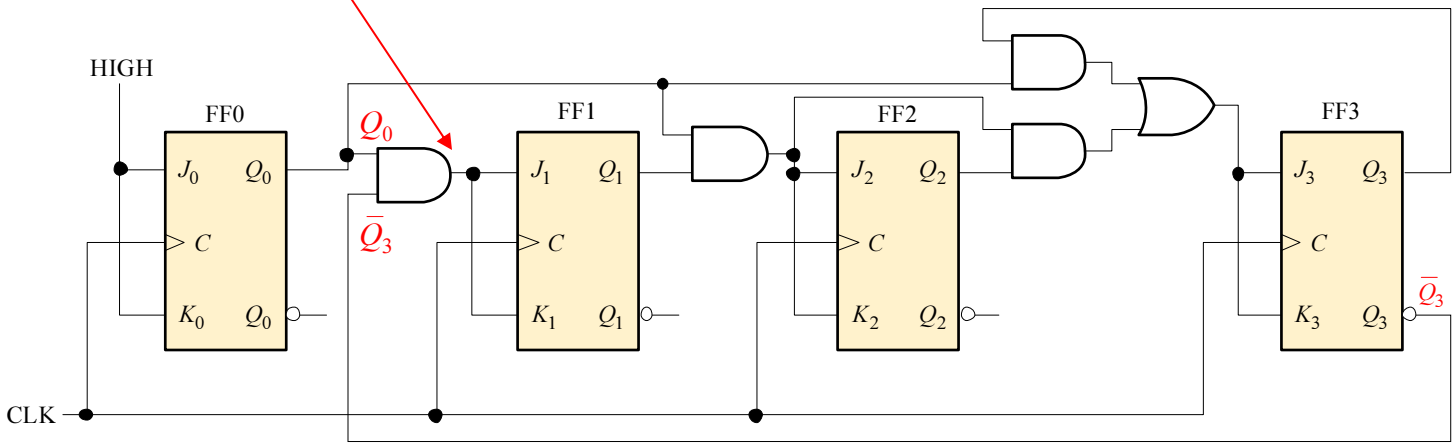


Summary

BCD Decade Counter

CLOCK PULSE	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 (recycles)	0	0	0	0

FF1 (Q1) toggle on the next clock pulse only if Q0 = 1 and Q3 = 0



Summary

BCD Decade Counter

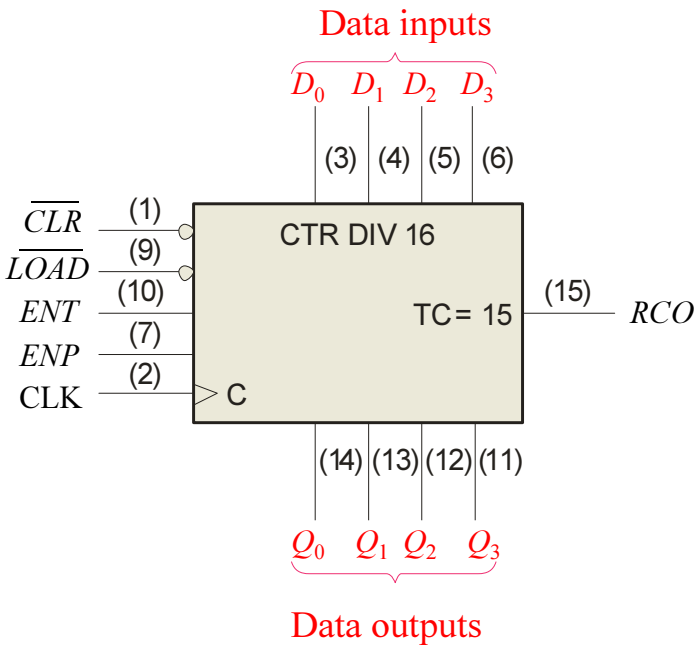
Waveforms for the decade counter:

These same waveforms can be obtained with an asynchronous counter in IC form – the 74LS90. It is available in a dual version – the 74LS390, which can be cascaded. It is slower than synchronous counters (max count frequency is 35 MHz), but is simpler.

Summary

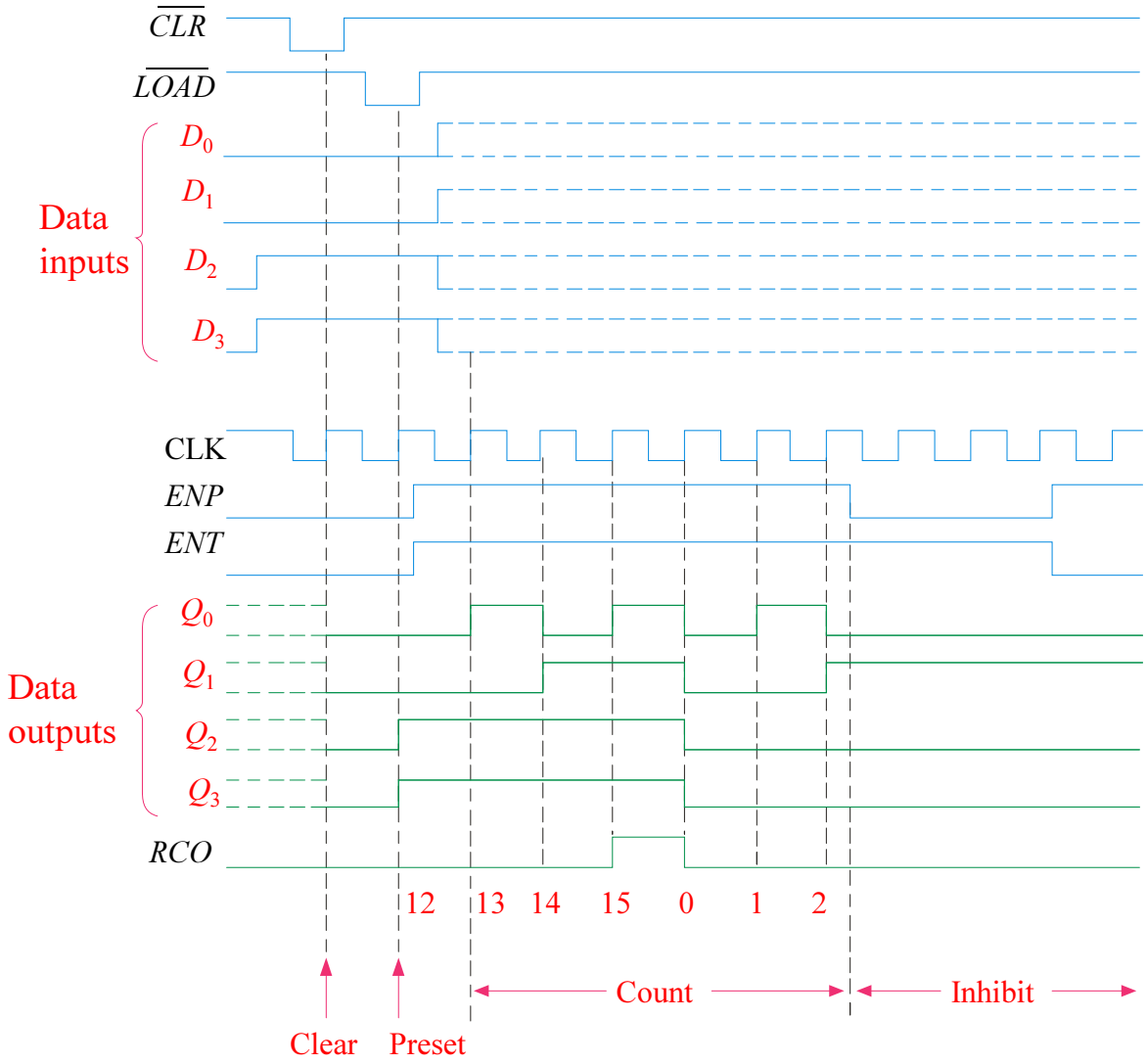
A 4-bit Synchronous Binary Counter

The 74LS163 is a 4-bit IC synchronous counter with additional features over a basic counter. It has parallel load, a $\overline{CLR}$ input, two chip enables, and a ripple count output that signals when the count has reached the terminal count.



Example waveforms are on the next slide…

Summary



Summary

Up/Down Synchronous Counters

Q0 toggles on every clock pulse for both UP and DOWN

Q1 toggles when Q0 = 1 for UP AND when Q0 = 0 for DOWN

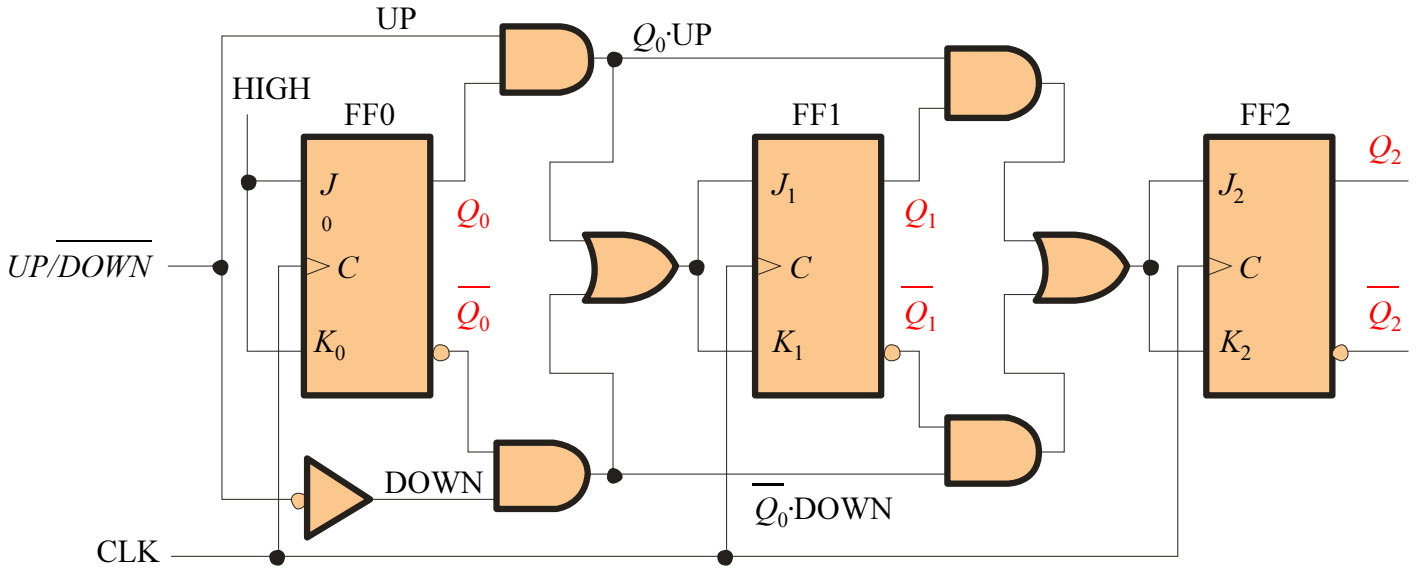
Q2 toggles when Q0=Q1=1 for UP AND when Q0=Q1=0 for DOWN

CLOCK PULSE	UP	Q_2	Q_1	Q_0	DOWN
0		0	0	0	
1		0	0	1	
2		0	1	0	
3		0	1	1	
4		1	0	0	
5		1	0	1	
6		1	1	0	
7		1	1	1	

Summary

Up/Down Synchronous Counters

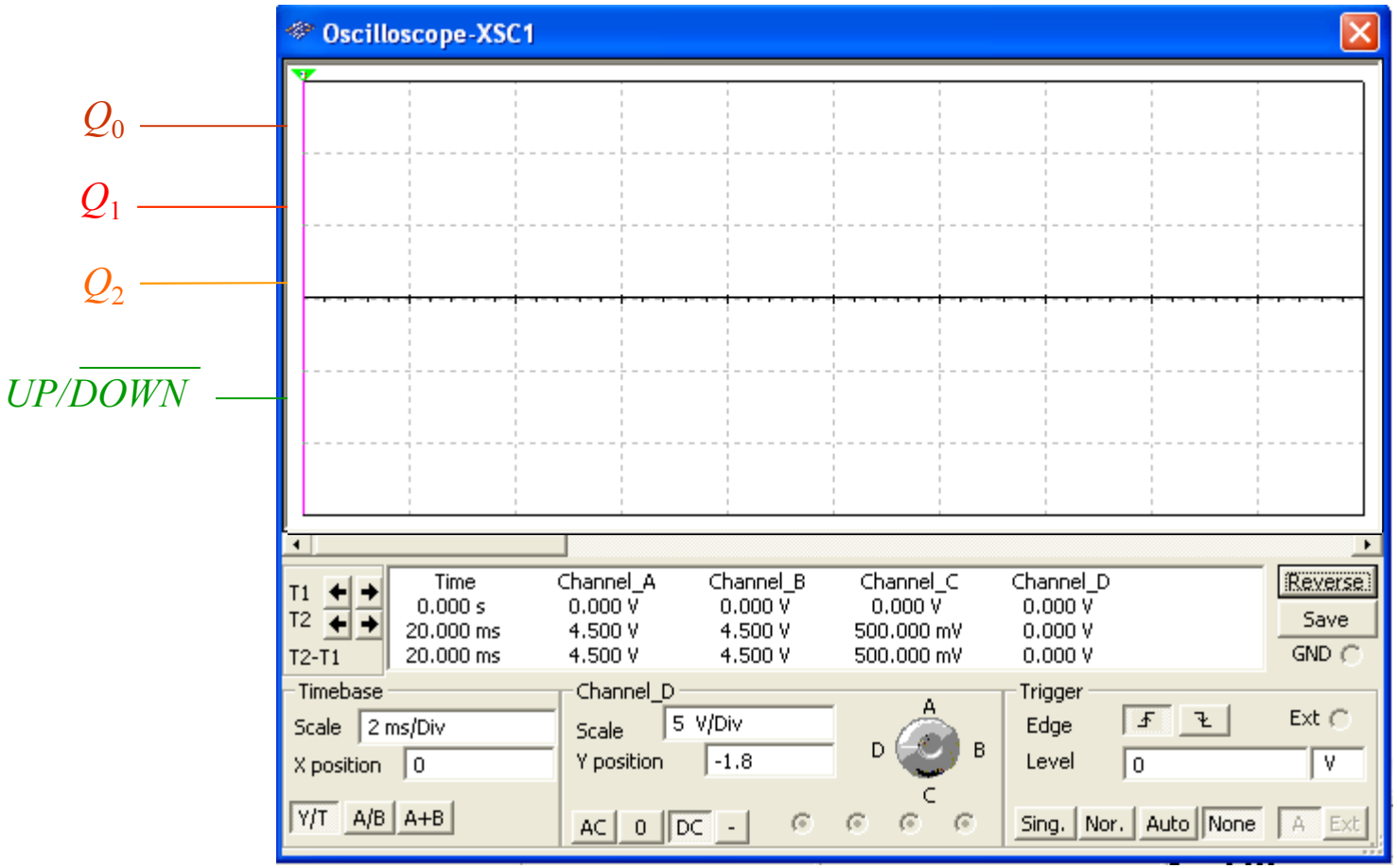
An up/down counter is capable of progressing in either direction depending on a control input.



Example waveforms from Multisim are on the next slide…

Summary

Up/Down Synchronous Counters

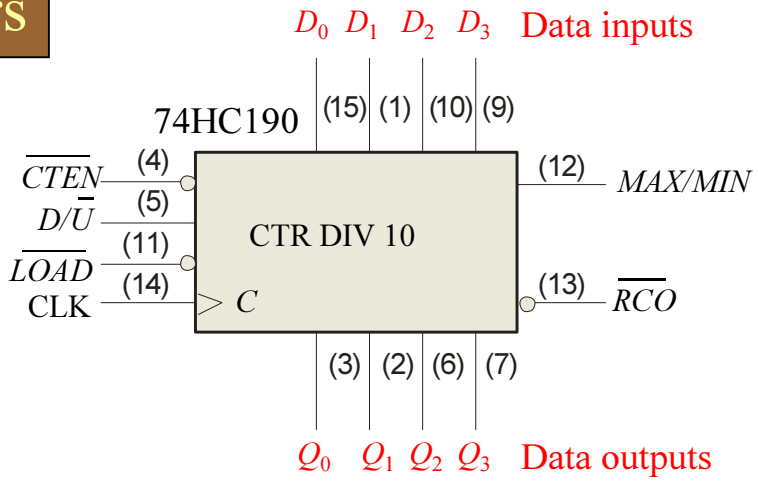


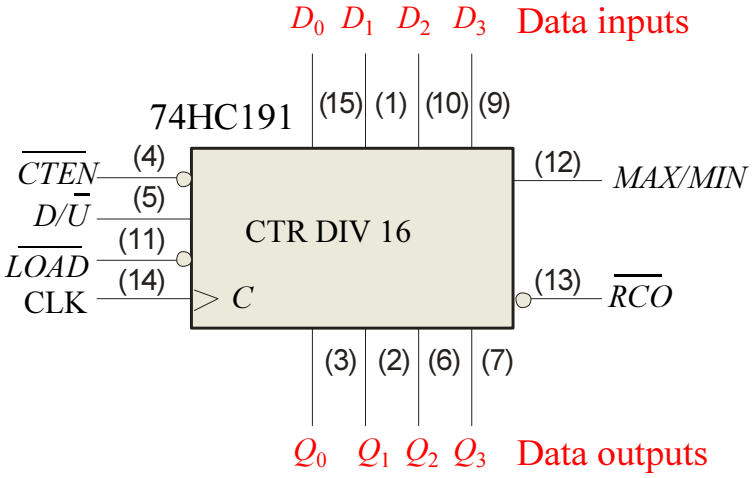
Summary

Up/Down Synchronous Counters

The 74HC190 is a high speed CMOS synchronous up/down decade counter with parallel load capability. It also has a active LOW ripple clock output ($\overline{RCO}$) and a *MAX/MIN* output when the terminal count is reached.

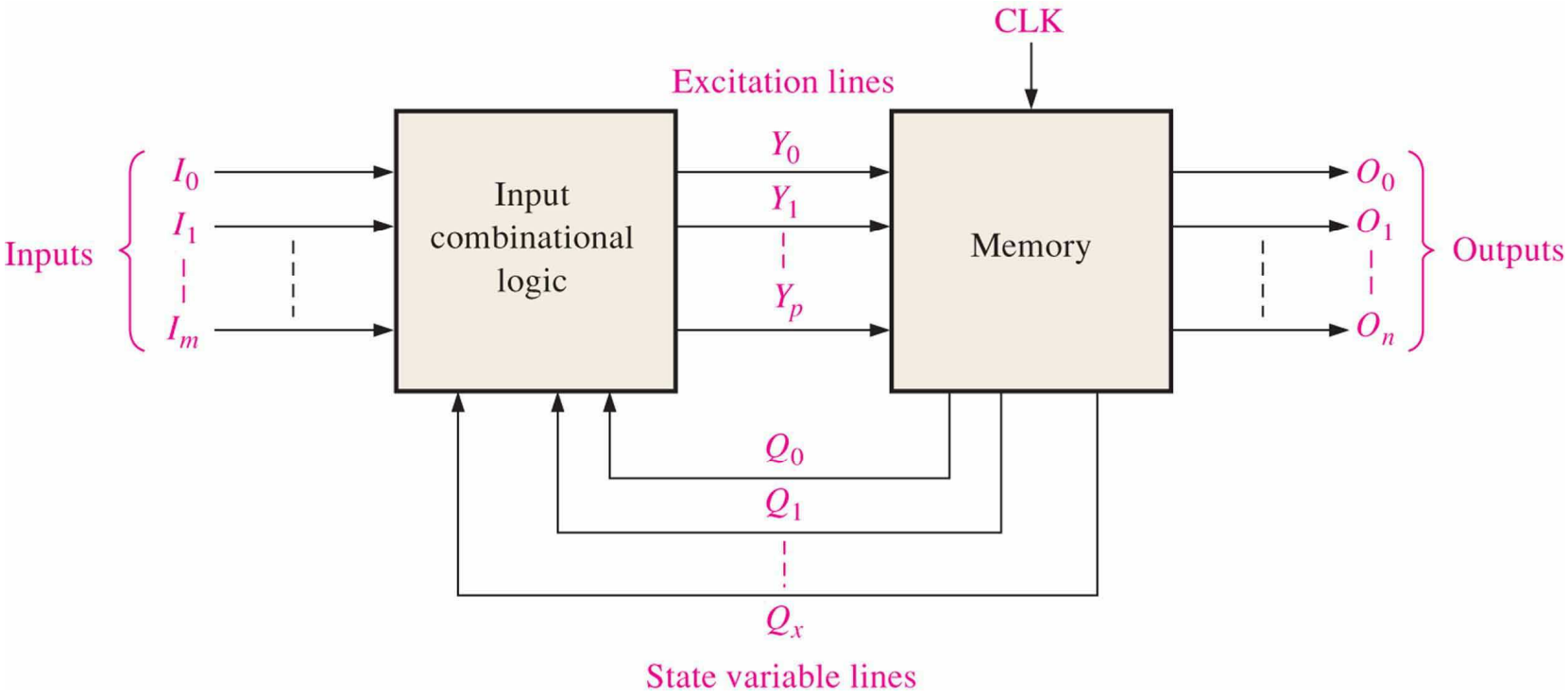
The 74HC191 has the same inputs and outputs but is a synchronous up/down binary counter.





Summary

Synchronous Counter Design



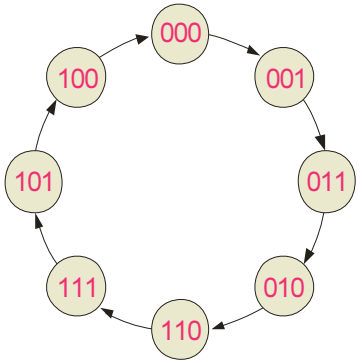
Summary

Synchronous Counter Design

Most requirements for synchronous counters can be met with available ICs. In cases where a special sequence is needed, you can apply a step-by-step design process.

The steps in design are described in detail in the text and lab manual. Start with the desired sequence and draw a state diagram and next-state table. The gray code sequence from the text is illustrated:

State diagram:



Next state table:

Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Summary

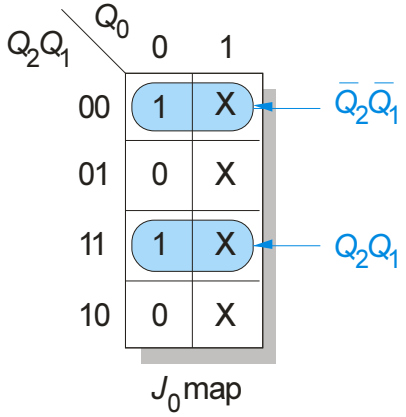
Synchronous Counter Design

The J-K transition table lists all combinations of present output (Q_N) and next output (Q_{N+1}) on the left. The inputs that produce that transition are listed on the right.

Each time a flip-flop is clocked, the J and K inputs required for that transition are mapped onto a K-map.

Output Transitions Q_N	Q_{N+1}	Flip-Flop Inputs J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

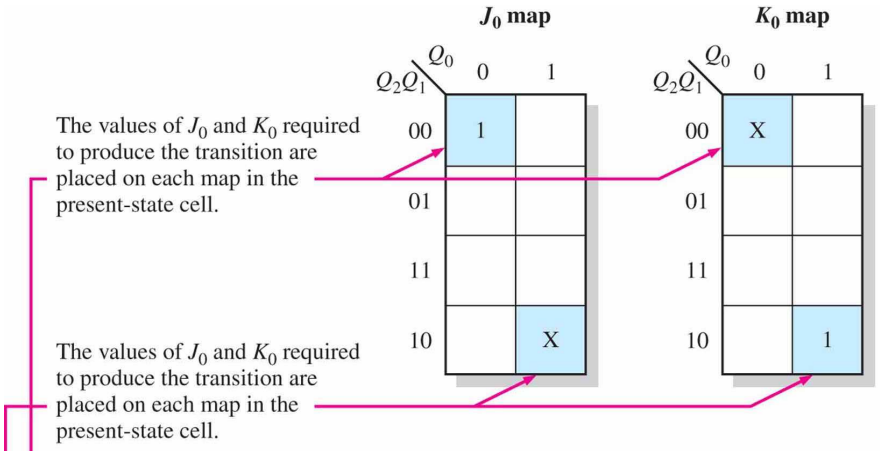
An example of the J_0 map is:

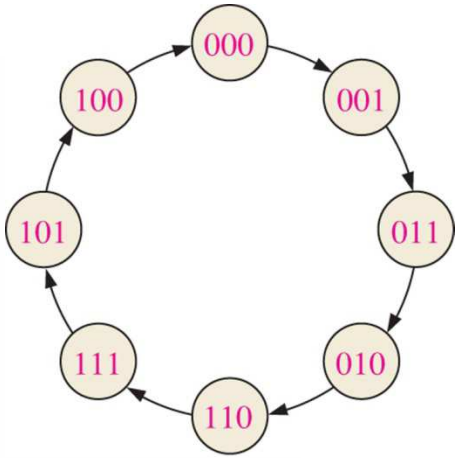


The logic for each input is read and the circuit is constructed. The next slide shows the implementation details of the circuit for the gray code counter…

Summary

Synchronous Counter Design





OUTPUT TRANSITIONS Q_N	Q_{N+1}	FLIP-FLOP INPUTS J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Flip-flop transition table

PRESENT STATE Q_2	Q_1	Q_0	NEXT STATE Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

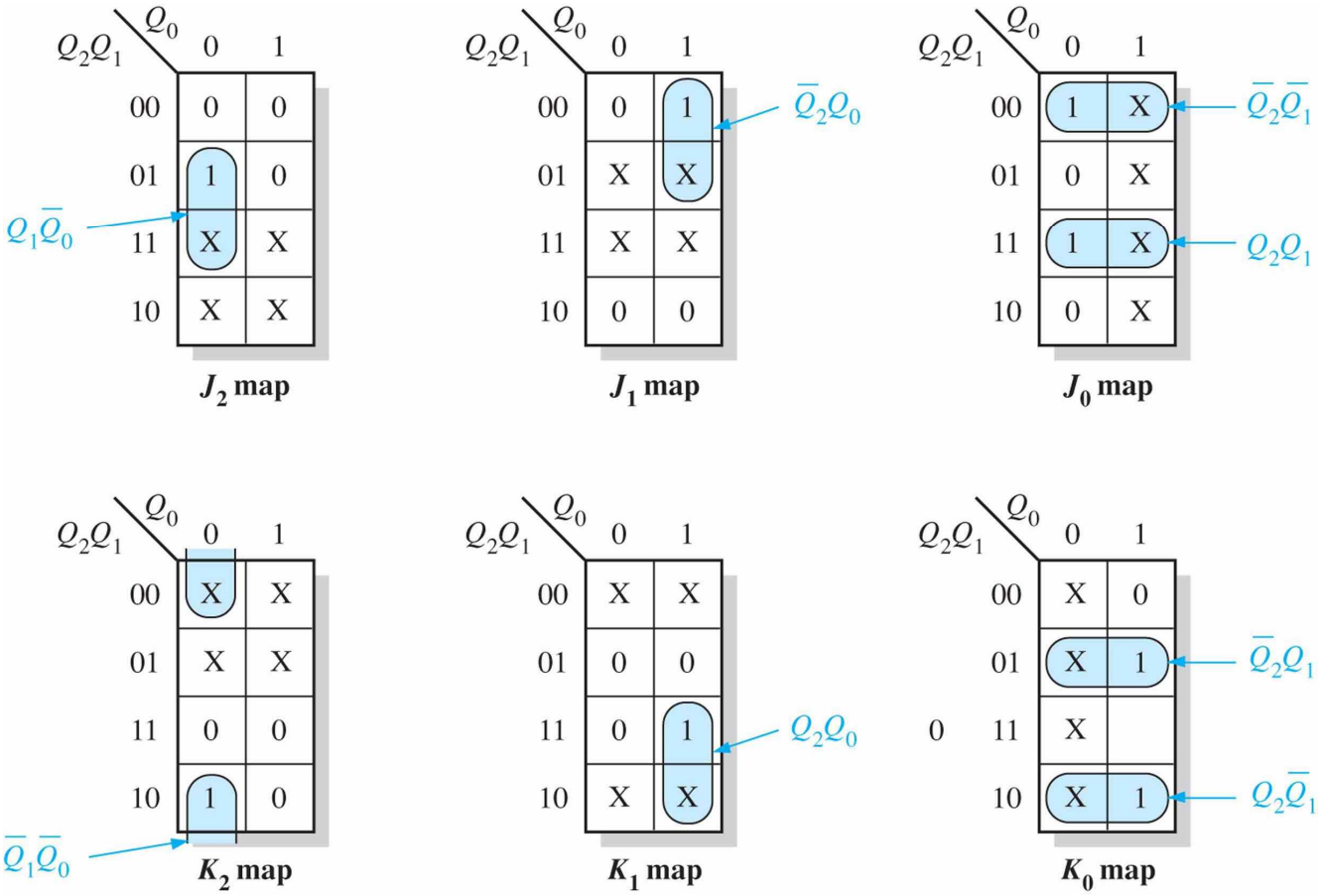
For the present state 000, Q_0 makes a transition from 0 to 1 to the next state.

For the present state 101, Q_0 makes a transition from 1 to 0 to the next state.

Next-state table

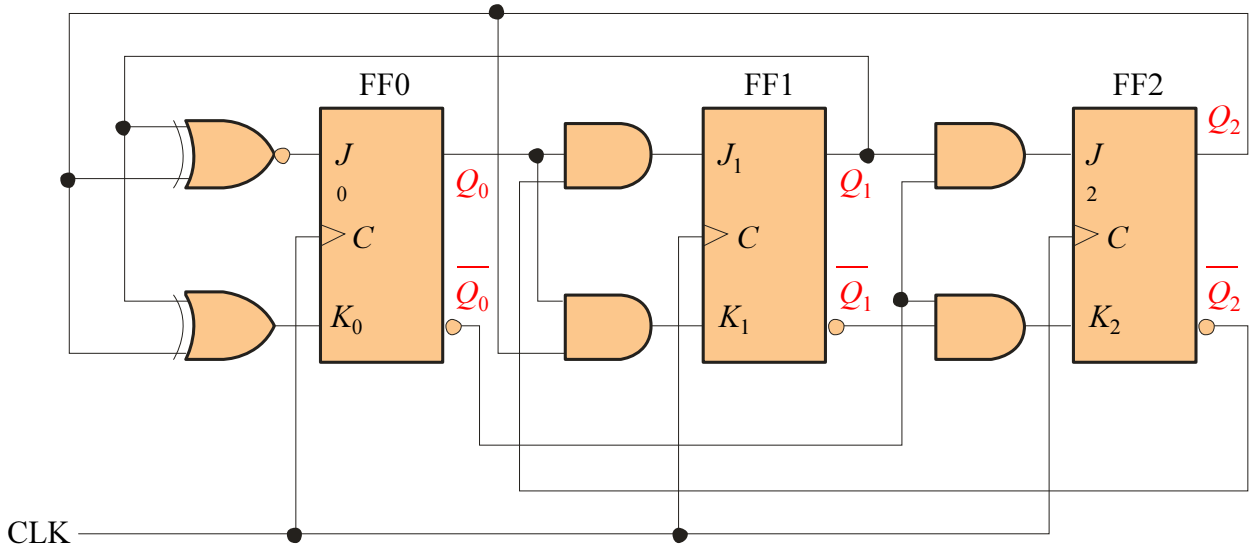
Summary

Synchronous Counter Design



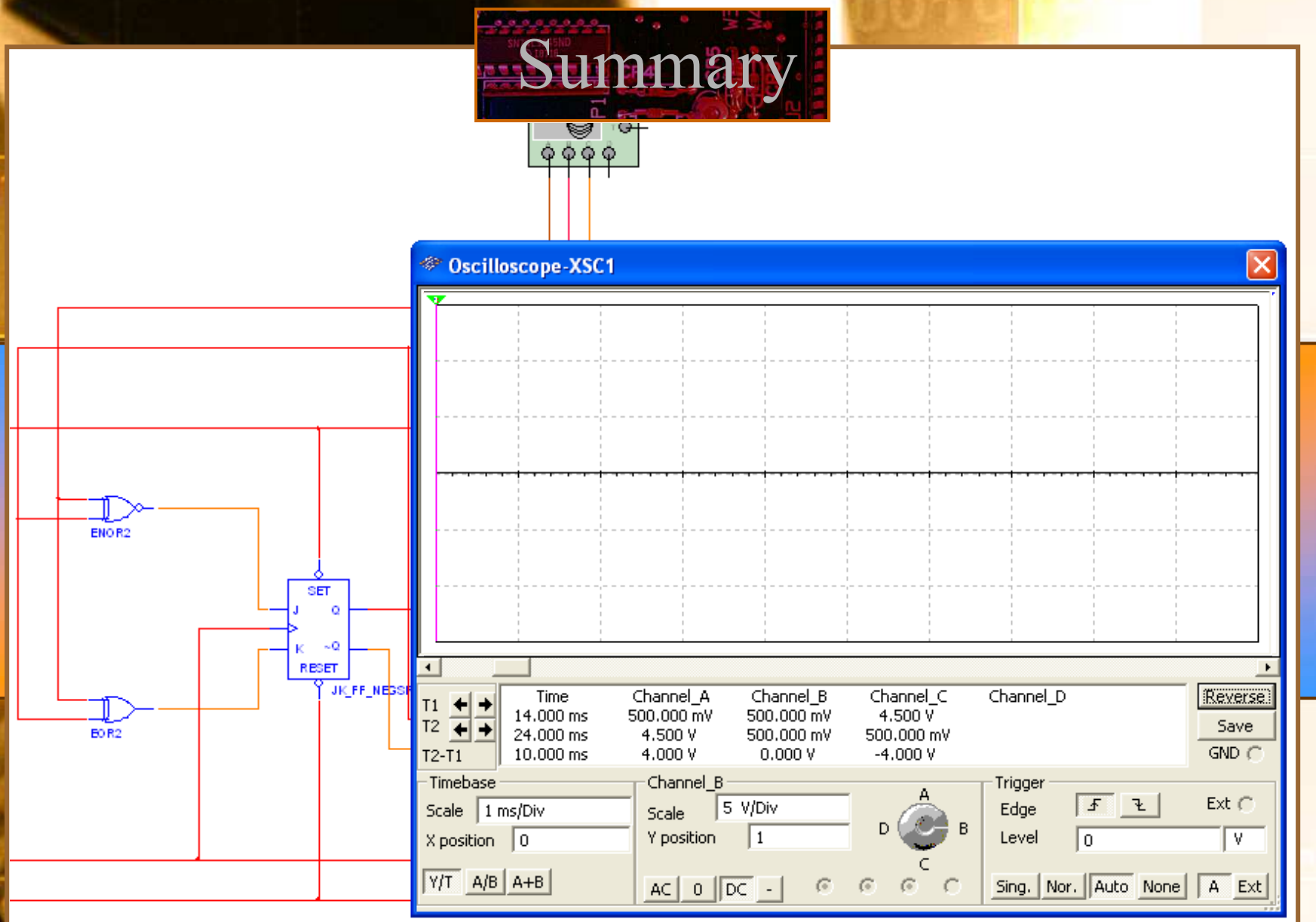
Summary

Synchronous Counter Design



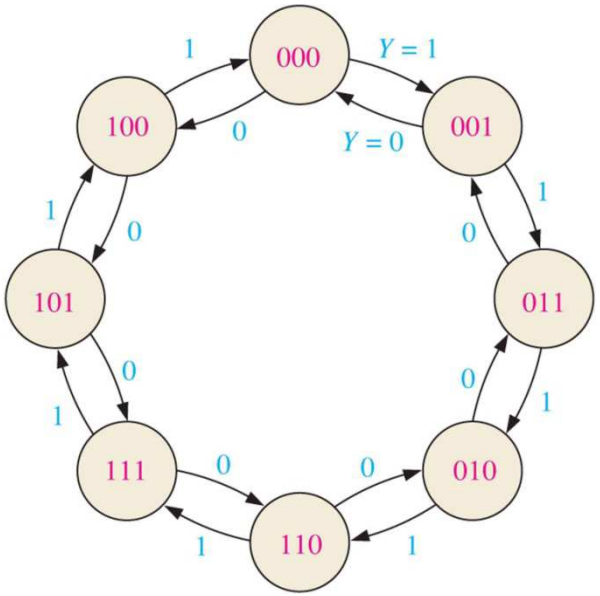
The circuit can be checked with Multisim before constructing it. The next slide shows the Multisim result…

Summary



Summary

Synchronous 3-bit Up/Down Gray Code Counter

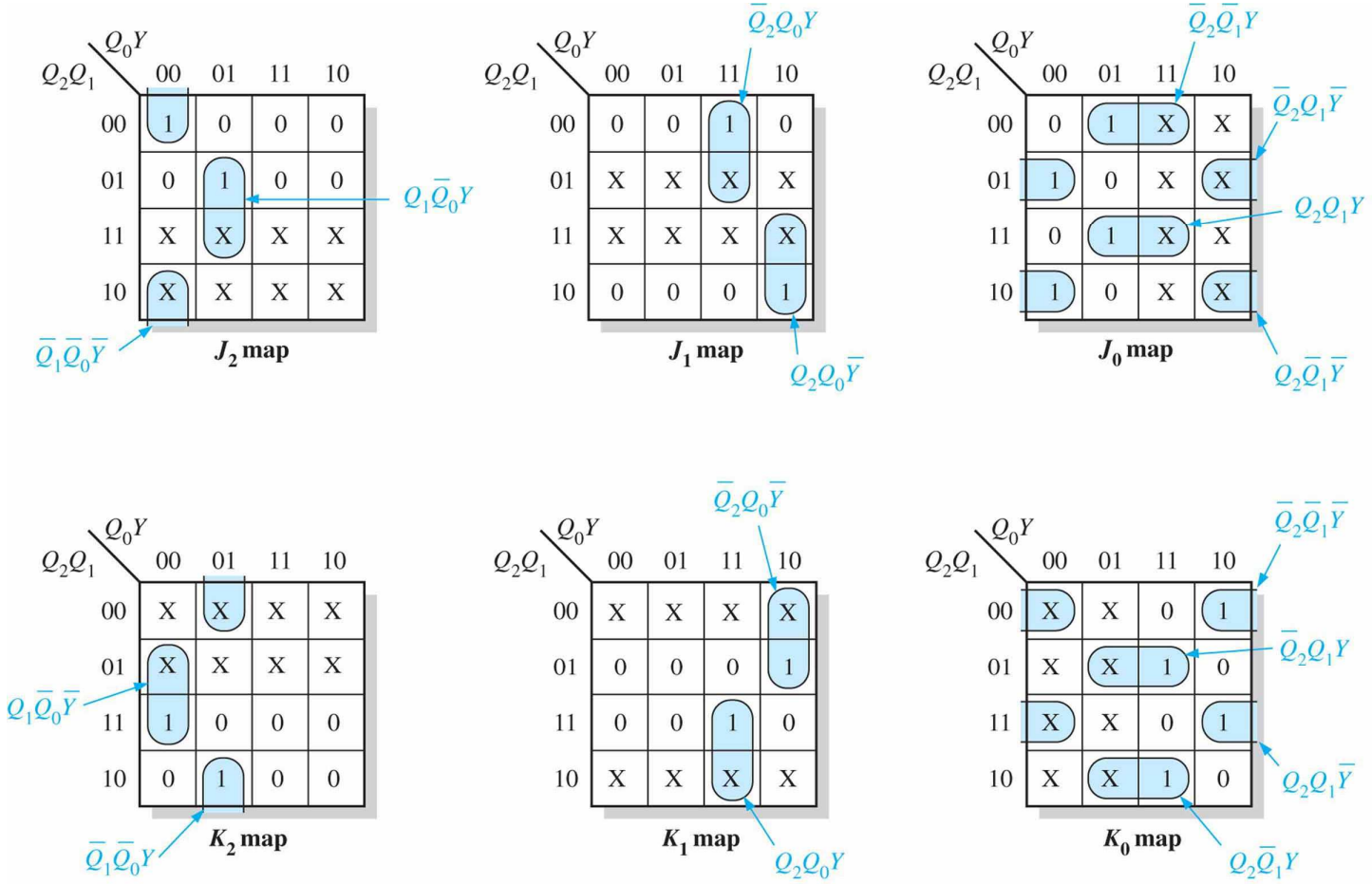


PRESENT STATE			NEXT STATE Y = 0 (DOWN)			NEXT STATE Y = 1 (UP)		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	1
0	1	1	0	0	1	0	1	0
0	1	0	0	1	1	1	1	0
1	1	0	0	1	0	1	1	1
1	1	1	1	1	0	1	0	1
1	0	1	1	1	1	1	0	0
1	0	0	1	0	1	0	0	0

Y = UP/$\overline{\text{DOWN}}$ control input.

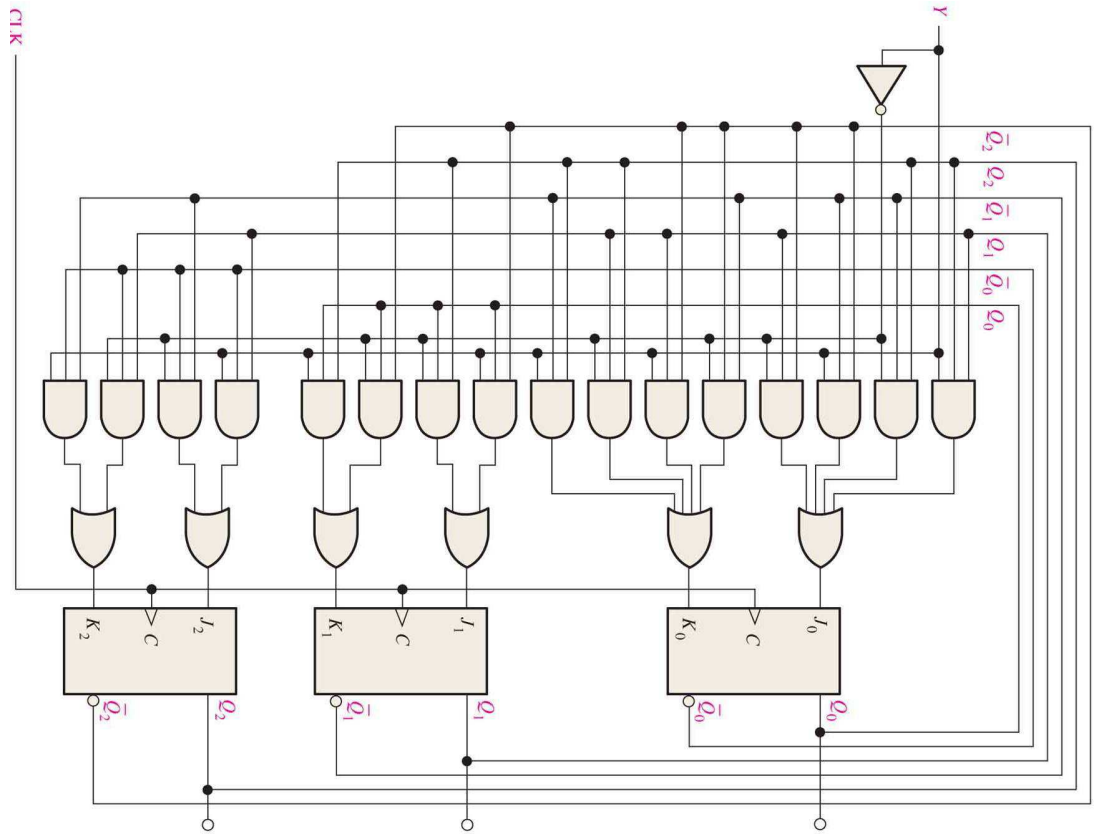
Summary

Synchronous 3-bit Up/Down Gray Code Counter



Summary

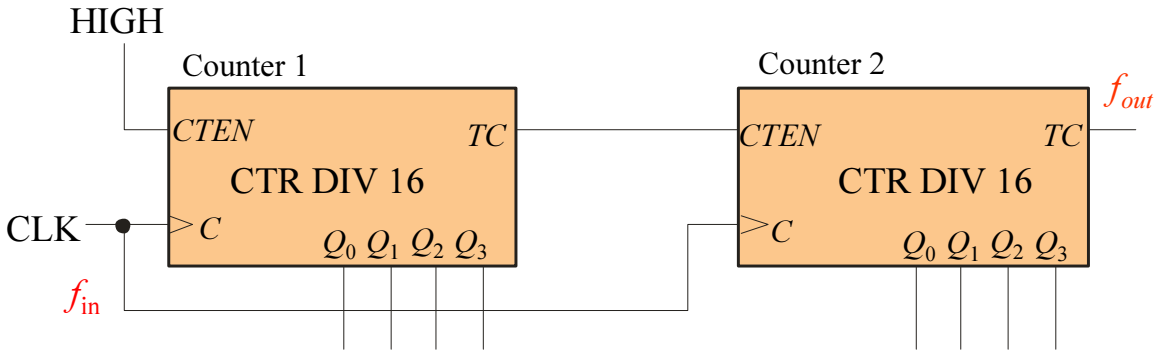
Synchronous 3-bit Up/Down Gray Code Counter



Summary

Cascaded counters

Cascading is a method of achieving higher-modulus counters. For synchronous IC counters, the next counter is enabled only when the terminal count of the previous stage is reached.



Example

a) What is the modulus of the cascaded DIV 16 counters?

b) If f_{in} =100 kHz, what is f_{out}?

Solution

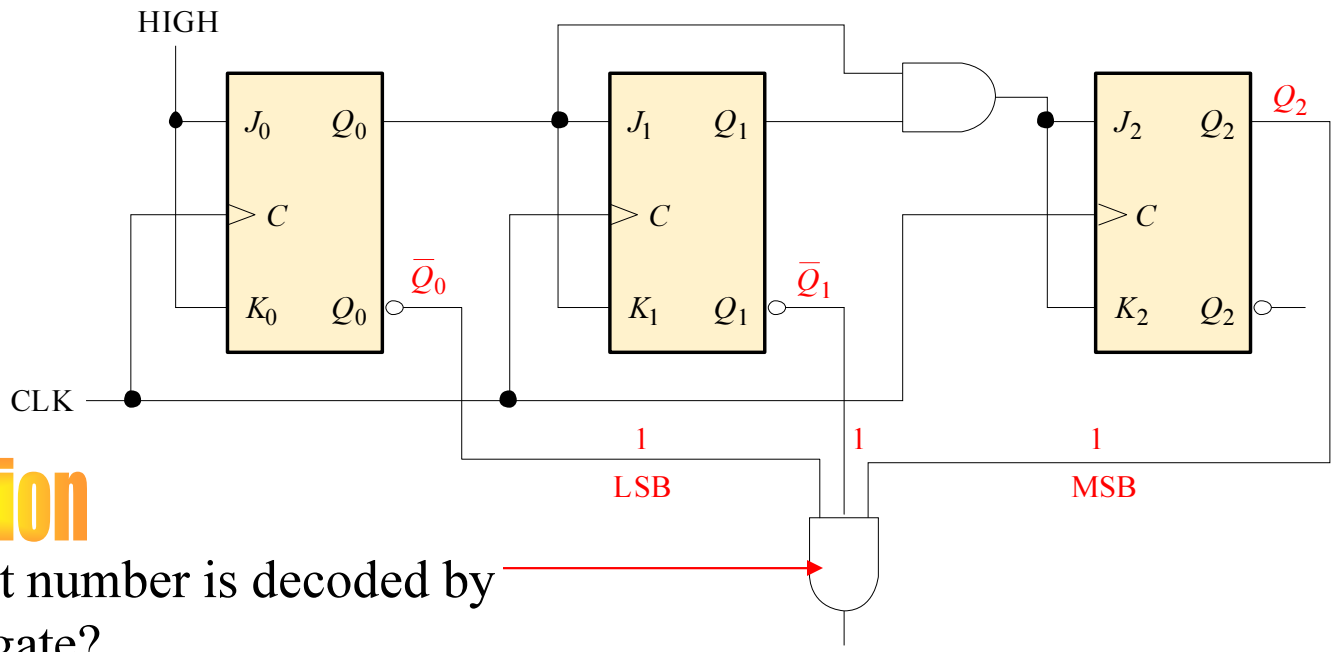
a) Each counter divides the frequency by 16. Thus the modulus is $16^2 = 256$.

b) The output frequency is 100 kHz/256 = 391 Hz

Summary

Counter Decoding

Decoding is the detection of a binary number and can be done with an AND gate.



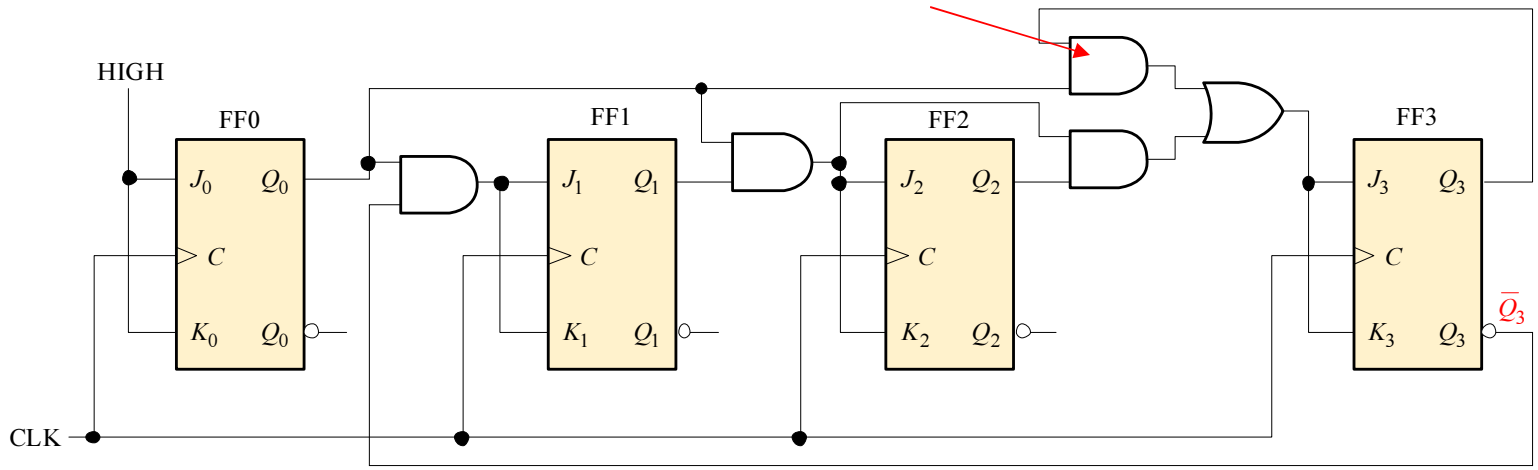
Question

What number is decoded by this gate?

Summary

Partial Decoding

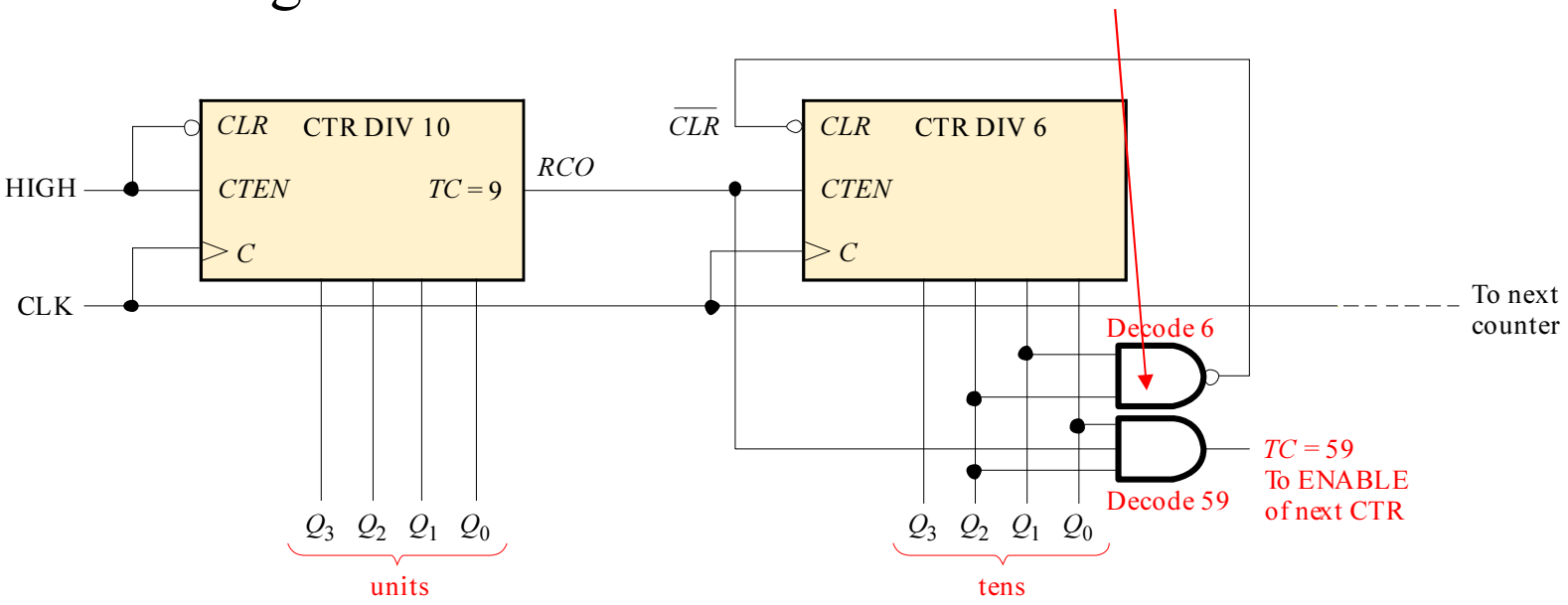
The decade counter shown previously incorporates *partial decoding* (looking at only the MSB and the LSB) to detect 1001. This was possible because this is the first occurrence of this combination in the sequence.



Summary

Resetting the Count with a Decoder

The divide-by-60 counter in the text also uses partial decoding to clear the tens count when a 6 was detected.



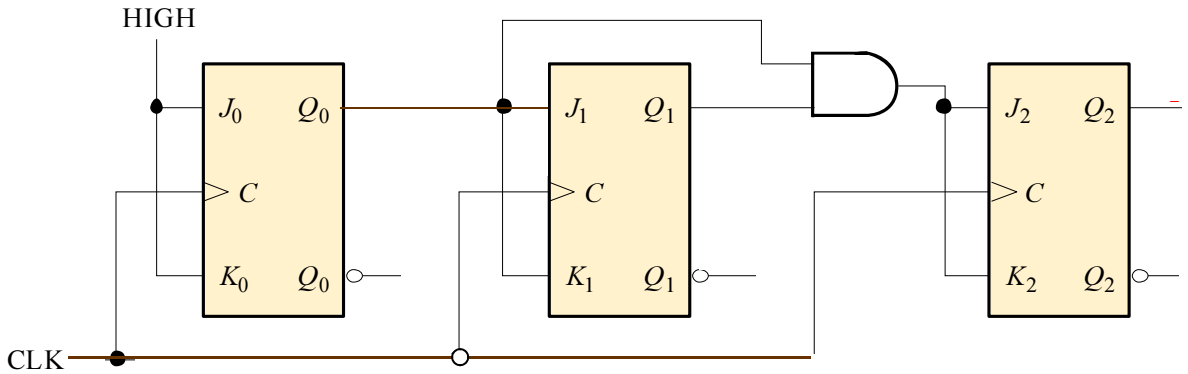
The divide characteristic illustrated here is a good way to obtain a lower frequency using a counter. For example, the 60 Hz power line can be converted to 1 Hz.

Summary

Counter Decoding

Example

Show how to decode state 5 with an active LOW output.



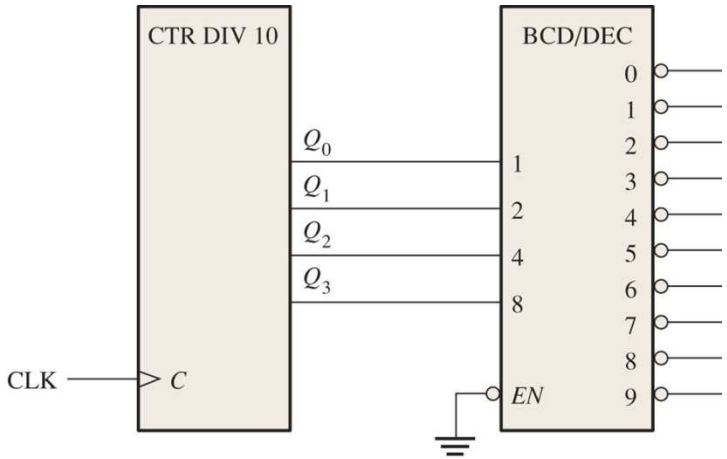
Solution

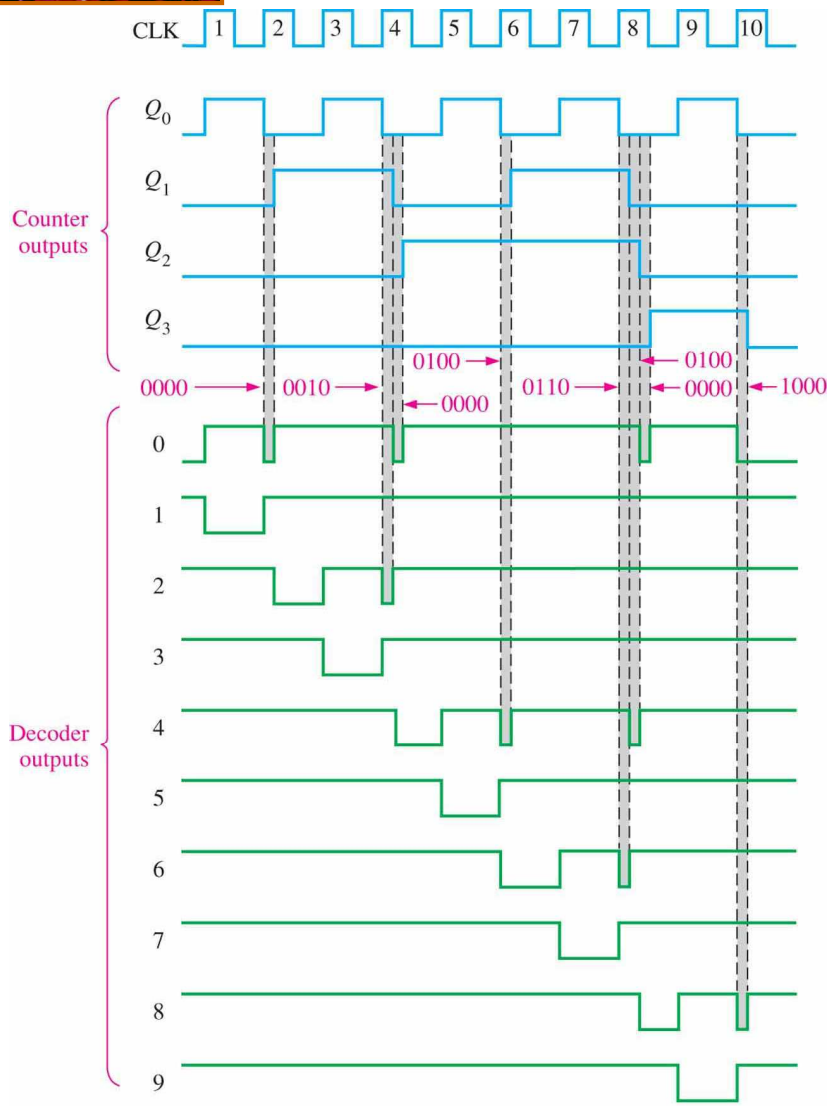
Notice that a NAND gate was used to give the active LOW output.

Summary

Decoding Glitches

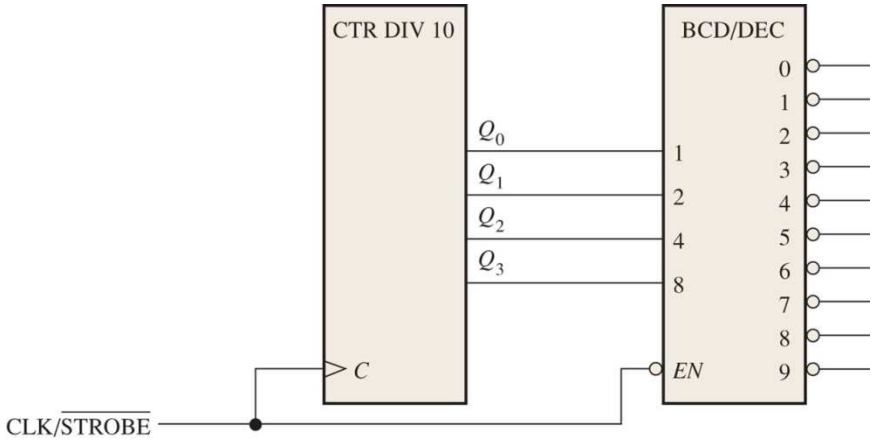
Glitches was produced due to the propagation delays from the decoding process

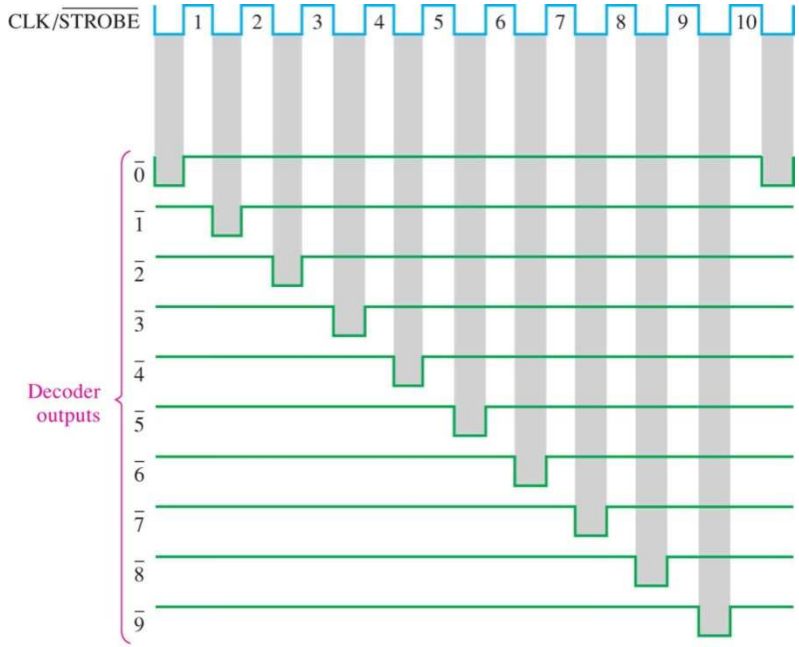




Summary

Decoding Glitches





Summary

Counter Applications