Logic Circuits Course

Ch. 8-4

Counters

Dr. Abul Kareem Alaloosy

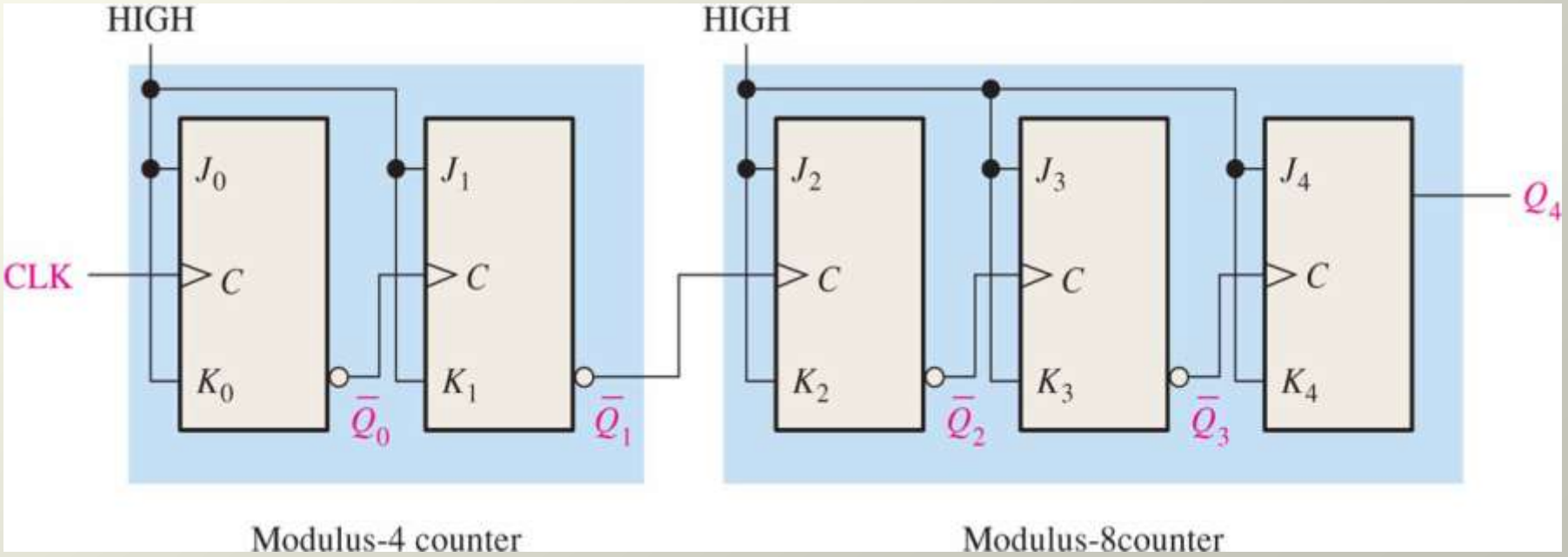
Counters

1- Asynchronous Counters
2- Synchronous Counters
3- Up/Down synchronous Counter
4- Design of synchronous Counter
5- Cascaded Counters
6- Counter Decoding
7- Counter Applications
8- Logic Symbols with Dependency Notation

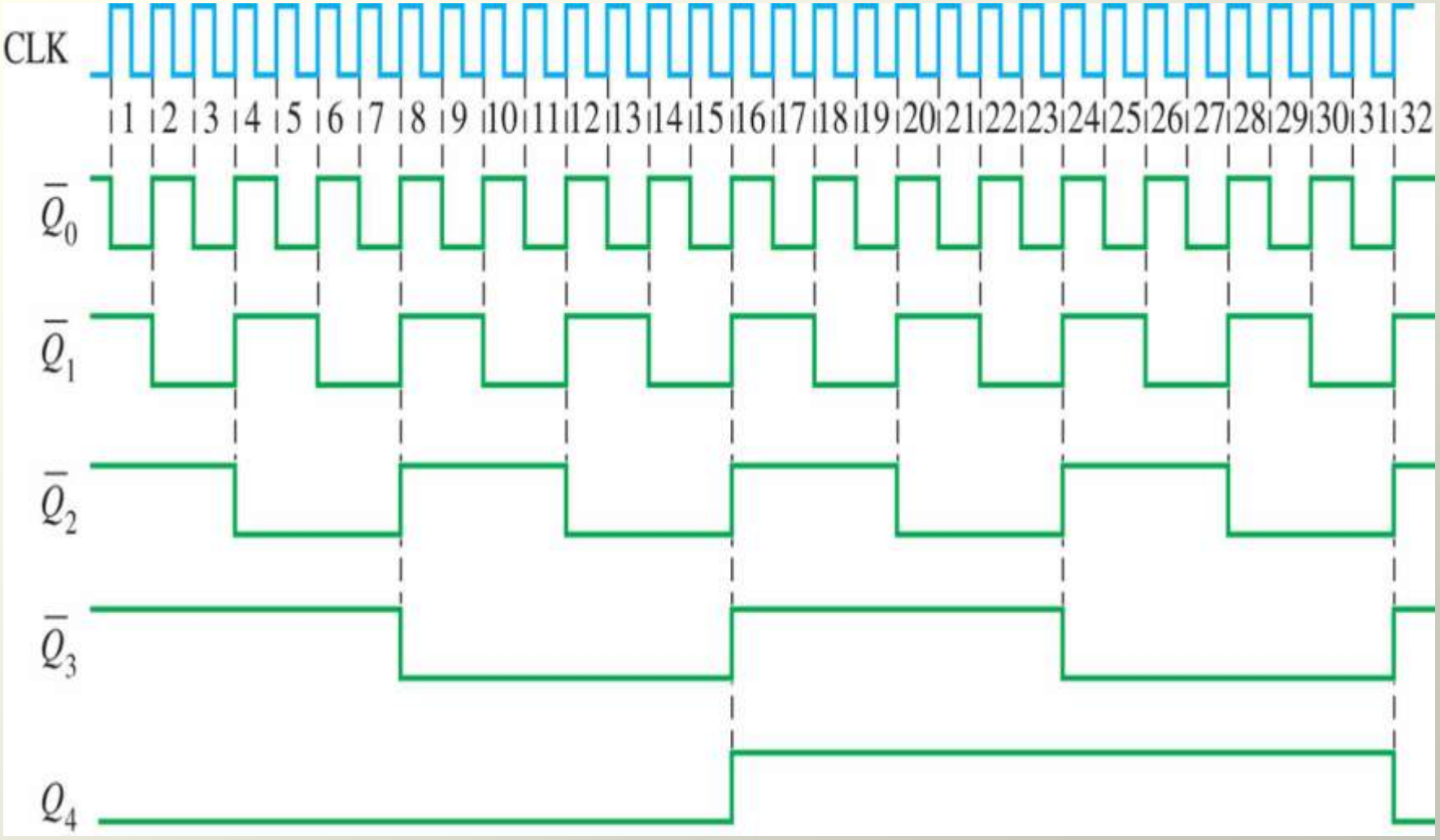
5. Cascade Counters

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

Two cascaded asynchronous counters (all *J* and *K* inputs are HIGH)
2-bit and 3-bit ripple counter



Timing diagram for the cascaded counter configuration

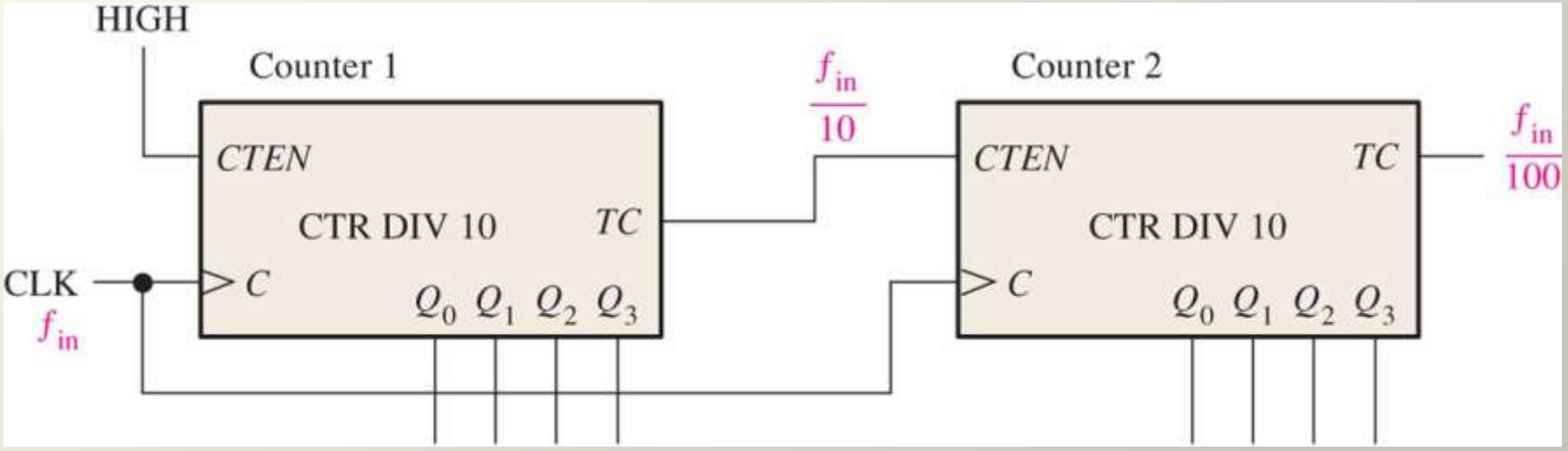


The overall modulus counter is 4 x 8= 32; they act as a divide by 32 counter

Synchronous cascading

The counter has output called terminal count TC (equal to RCO) and input called count enable CTEN.
TC is HIGH when counter reaches its last state, and is used for cascading and connected to CTEN input which enable counter for normal operation.

A modulus-100 counter using two cascaded decade counters.

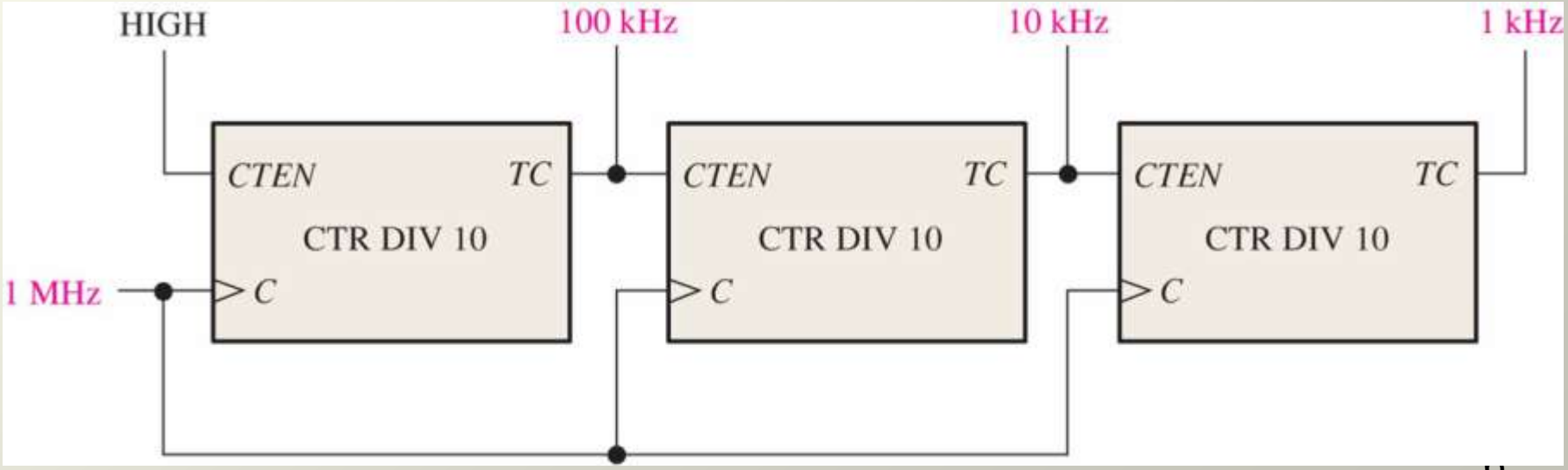


Frequency divider by 100

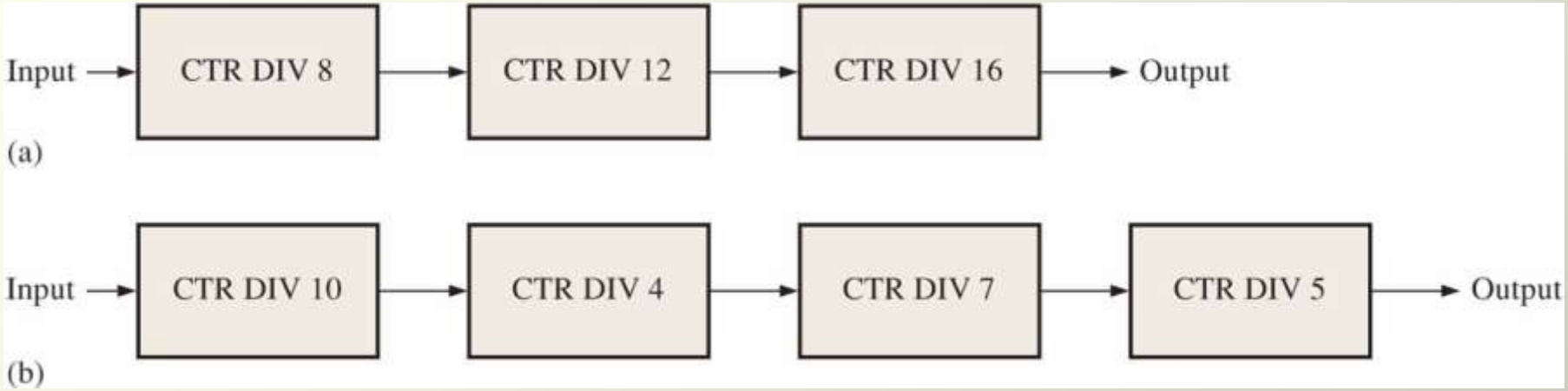
TC output of counter 1 is connected to CTEN input to counter 2.
Counter 2 is inhibited by the LOW on its CTEN input until counter 1 reaches its last state and its terminal count output goes HIGH. This HIGH enable counter 2.
At clock10 TC=HIGH and when clock11 comes Counter 2 start counting.
After completing each cycle of Counter 1, TC becomes HIGH and enables counter 2, then counter 2 advances to its second state.

Cascaded counters are often used to divide a high frequency clock signal to obtain highly accurate pulse frequency (countdown chain)

Three cascaded decade counters forming a divide-by-1000 frequency divider with intermediate divide-by-10 and divide-by-100 outputs.



Example: Determine the overall modulus of the two cascaded counter configuration

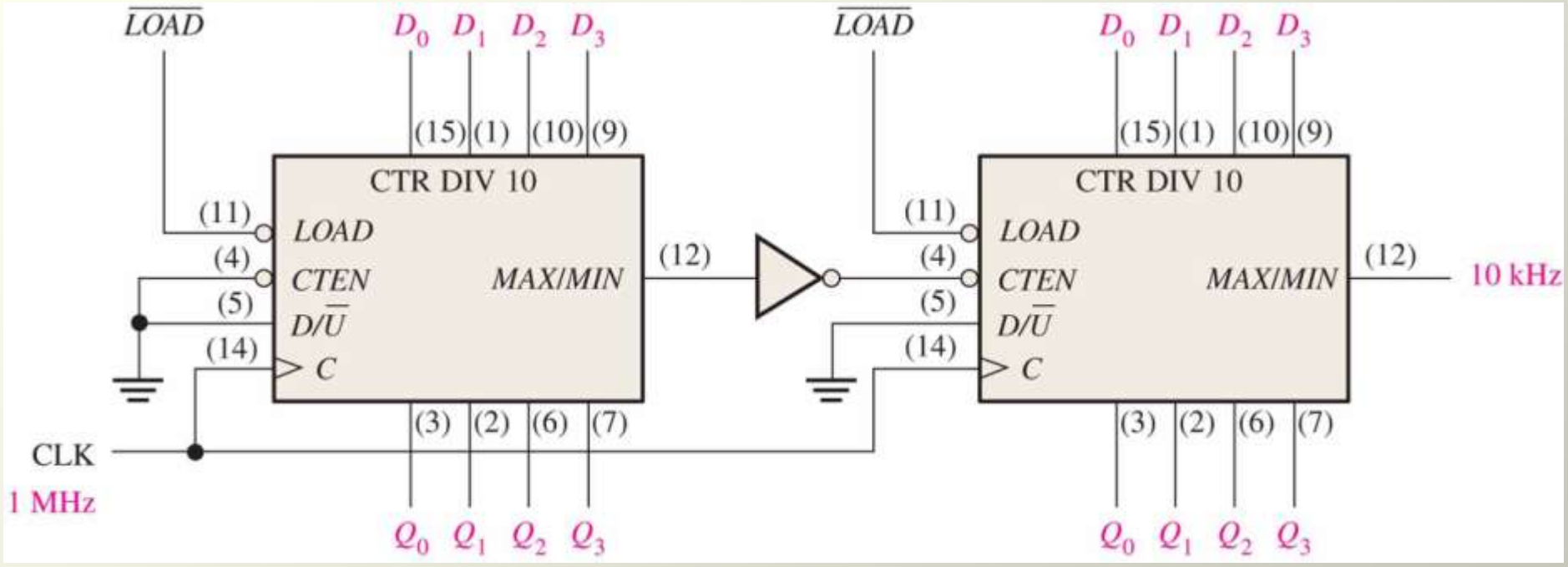


For (a) cascaded configuration
the overall modulus for the 3-counter is 8 x 12 x 16 = 1536

For (b) cascaded configuration
the overall modulus for the 4-counter is 10 x 4 x 7 x 5 = 1400

How many cascaded decade counters are required to divide a clock frequency by 100,000 ?

A divide-by-100 counter using two 74HC190 up/down decade counters connected for the up sequence.



The left counter produces a terminal count (MAX/MIN) pulse for every 10 clock pulses. The right counter produces a terminal count (MAX/MIN) pulse for every 100 clock pulses

Cascade counters with truncated sequences

A divide-by-40,000 counter using 74HC161 4-bit binary synchronous counters.

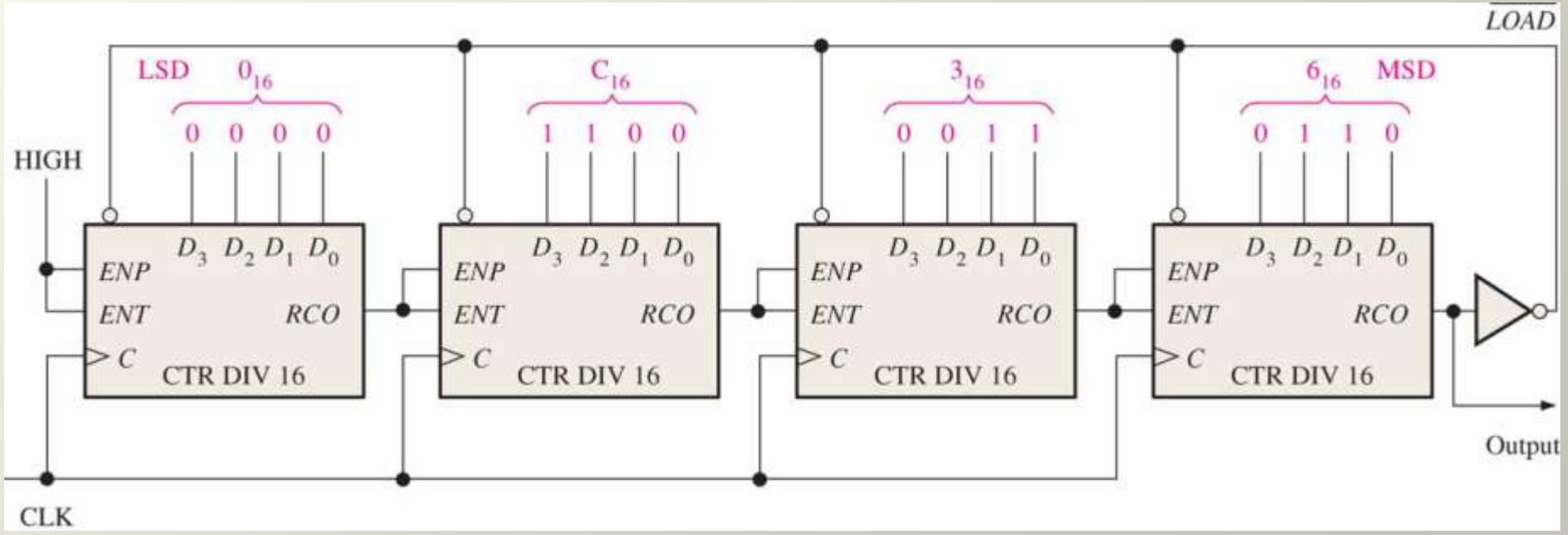
By using 4 cascaded counters (each divide by 16) we get 16x16x16x16=65,536 to divide by 40,000 , the counters must start from initial value of

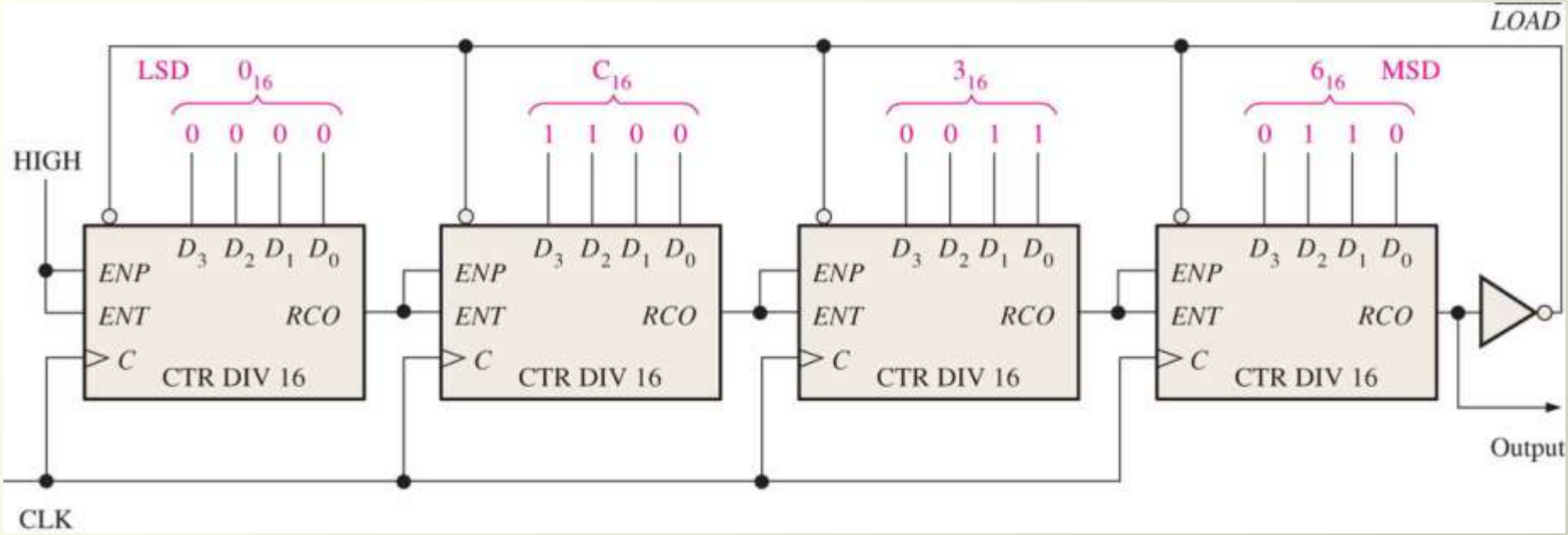
65,536-40,000=25,536 in binary 0110 0011 1100 0000 $63C0_{16}$

65,536 in binary 1111 1111 1111 1111

Note that each of the parallel data inputs is shown in binary order (the right-most bit D_0 is the LSB in each counter).

RCO is also called TC





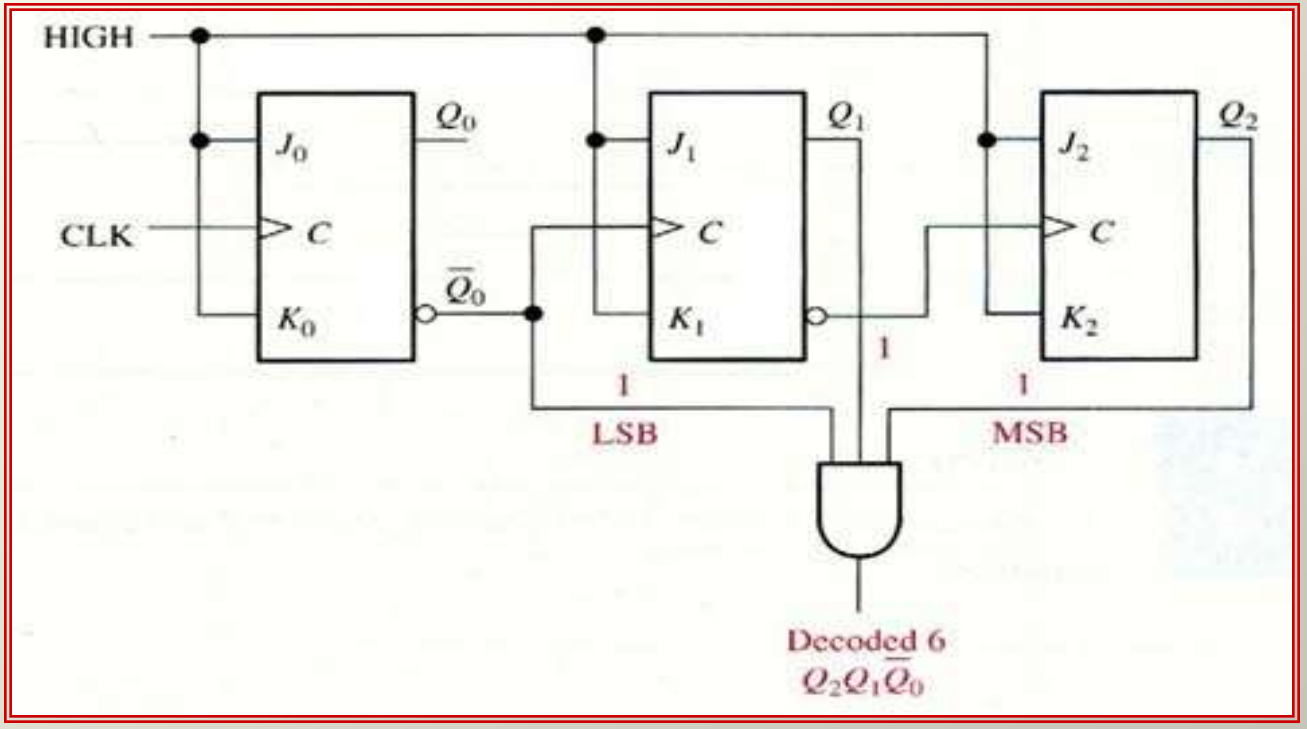
RCO of CTR 4 is inverted and applied to the $\overline{\text{load}}$ input of the of the 4 counters. Each time the counter reaches its terminal value of 65,535 , which is $1111\ 1111\ 1111\ 1111_2$, RCO goes HIGH and cause the number on the parallel data input ($63C0_{16}$) to be synchronously loaded into the counter with the clock pulse.

Thus, there is one RCO pulse from the right-most 4-bit counter for every 40,000 clock pulses.

With this technique any modulus can be achieved by synchronous loading of the counter to the appropriate initial state on each cycle.

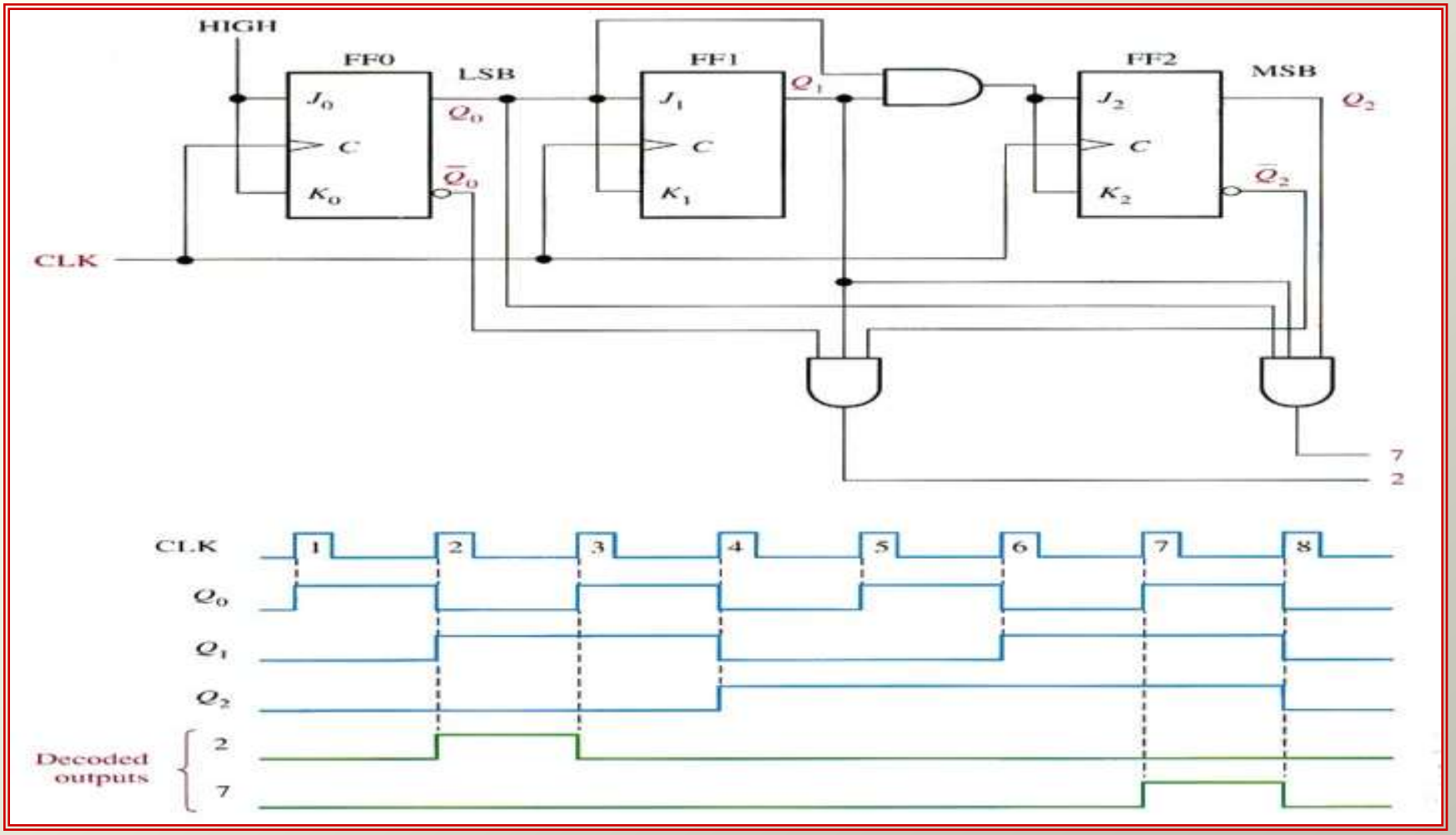
6. Counter Decoding

Suppose that you wish to decode binary state 6 (110) of a 3-bit binary counter. When Q_2=1, Q_1=1 and Q_0 = 0, a high appears on the output of the decoding gate, indicating that the counter is at state 6. This is called active High decoding.



Example :- Implement the decoding of binary state 2 and binary state 7 of a 3-bit synchronous counter.

Solution :-



You may only draw the counter block (CRT DIV 8) and use its output.

7. Counter Applications

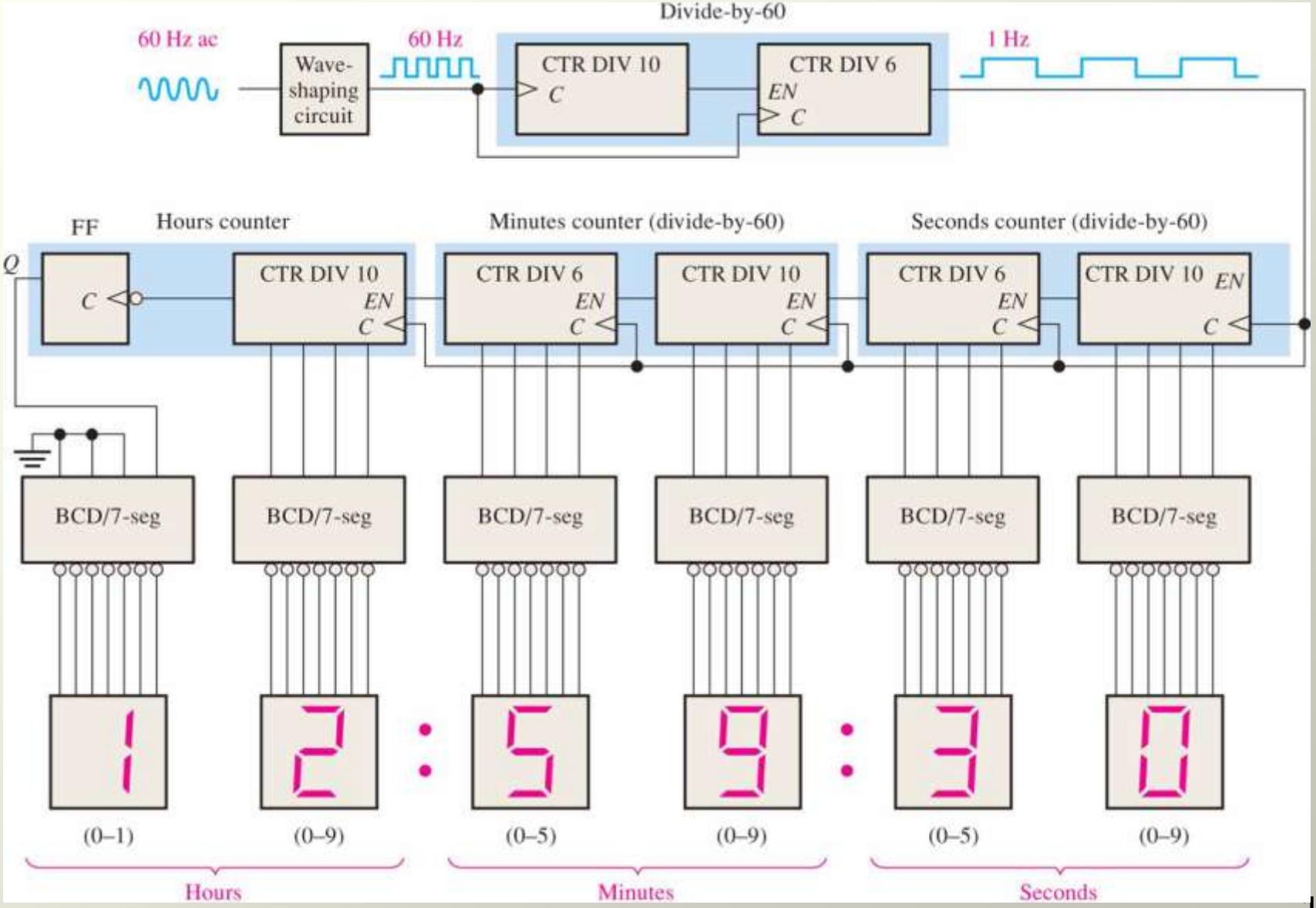
A Digital Clock:- a common example of a counter application is in timekeeping systems.

A divide-by-60 counter formed by a divided-by-10 counter followed by a divide-by-6 counter were used.

Both the seconds and minutes counts are also produced by divide-by-60 counters.

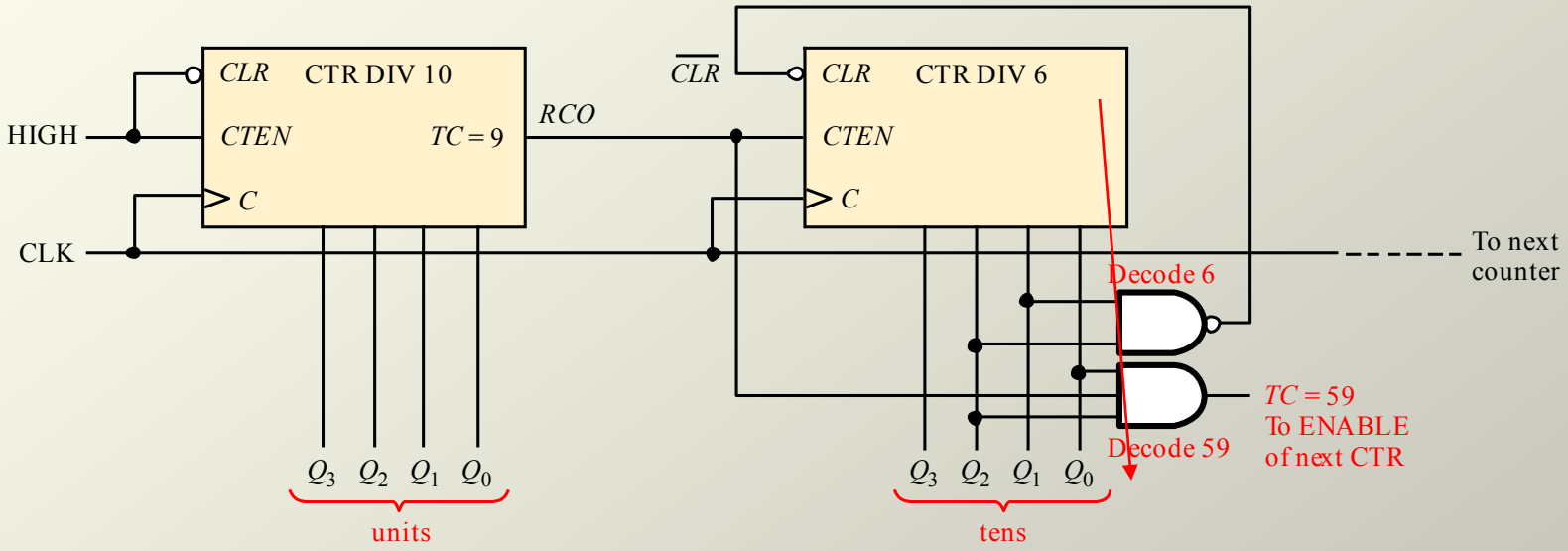
The hours are formed by a divide by 10 counter followed by a divide by two FF.

Simplified logic diagram for a 12-hour digital clock. Logic details using specific devices are shown in Figures 8–50 and 8–51.



Resetting the Count with a Decoder, divide by 60

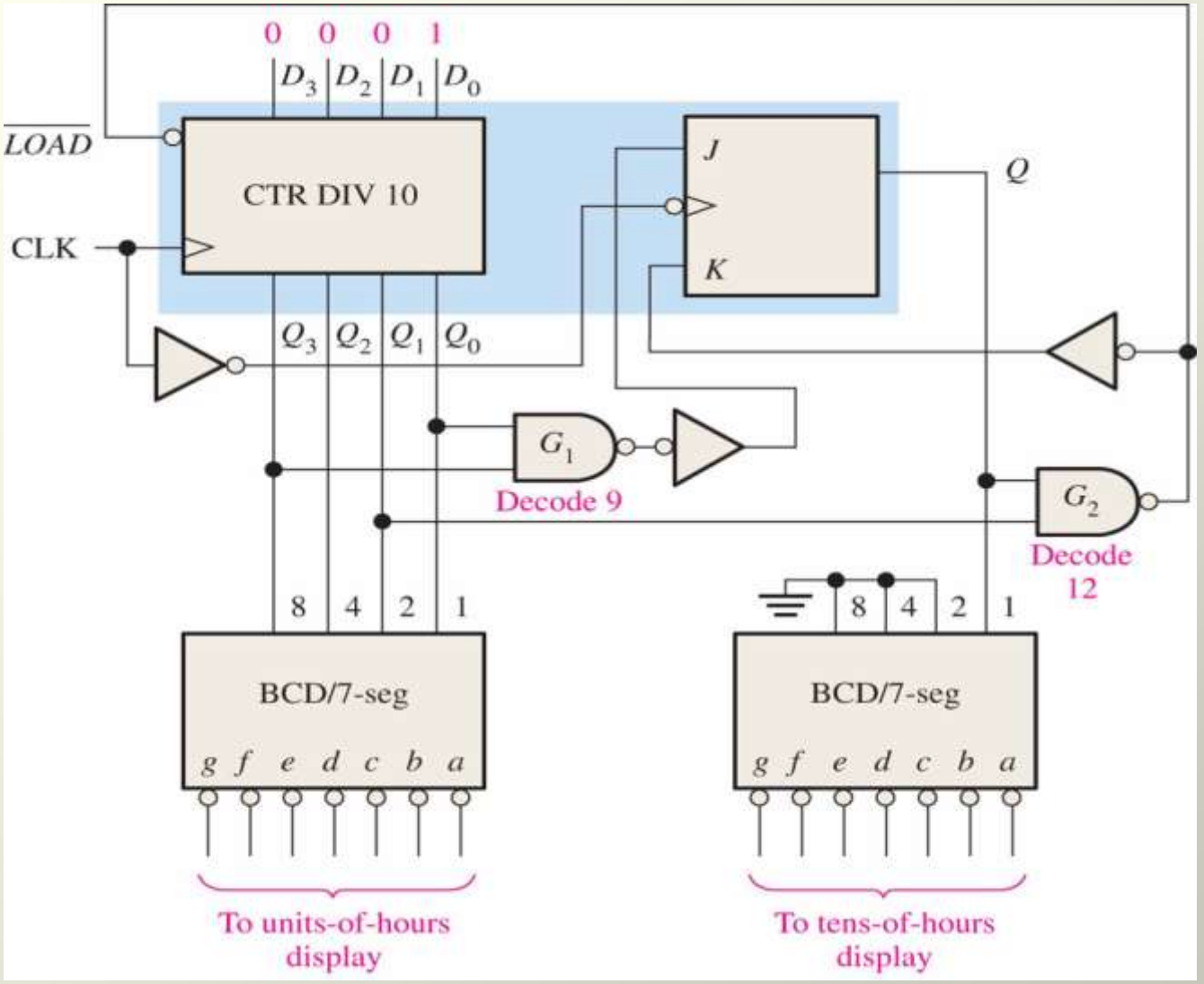
The divide-by-60 counter in the text also uses partial decoding to clear the tens count when a 6 was detected.



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.

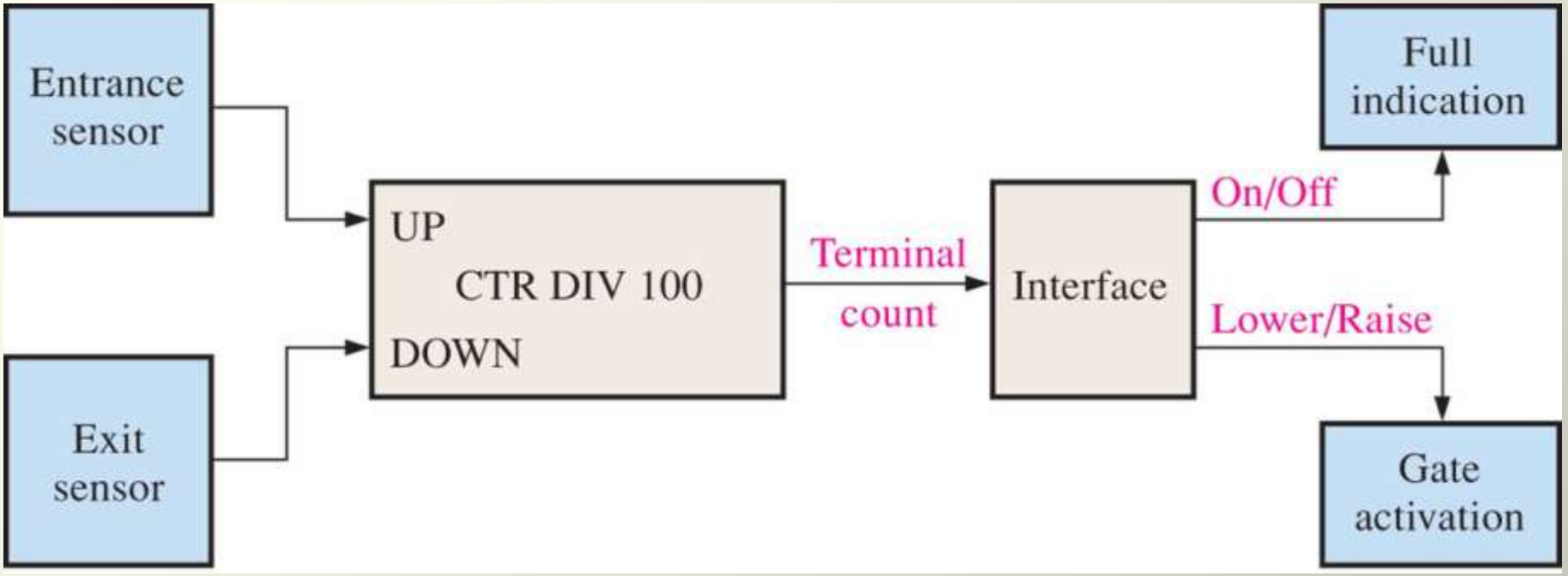
Logic diagram for hours counter and decoders.

In state 12 the Q2 output of the decade counter is HIGH, and the FF output is still SET, and the decoded-12 gate output is LOW.

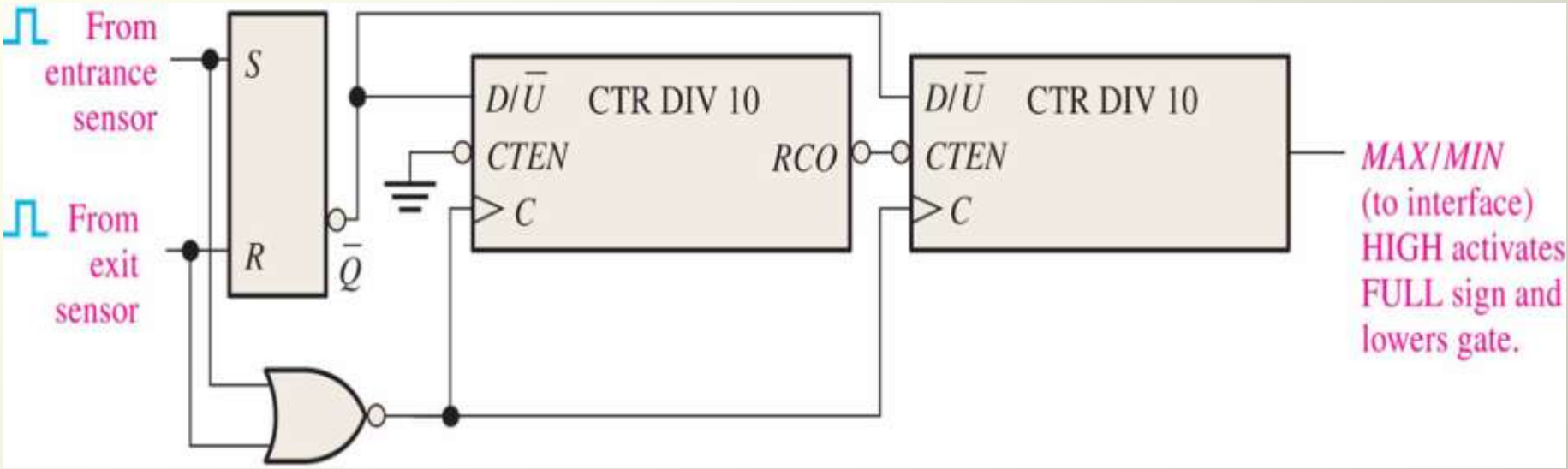


On the next clock pulse, decade counter is preset to 0001 from the data inputs, and the FF is RESET (J=0, K=1).

Functional block diagram for parking garage control.

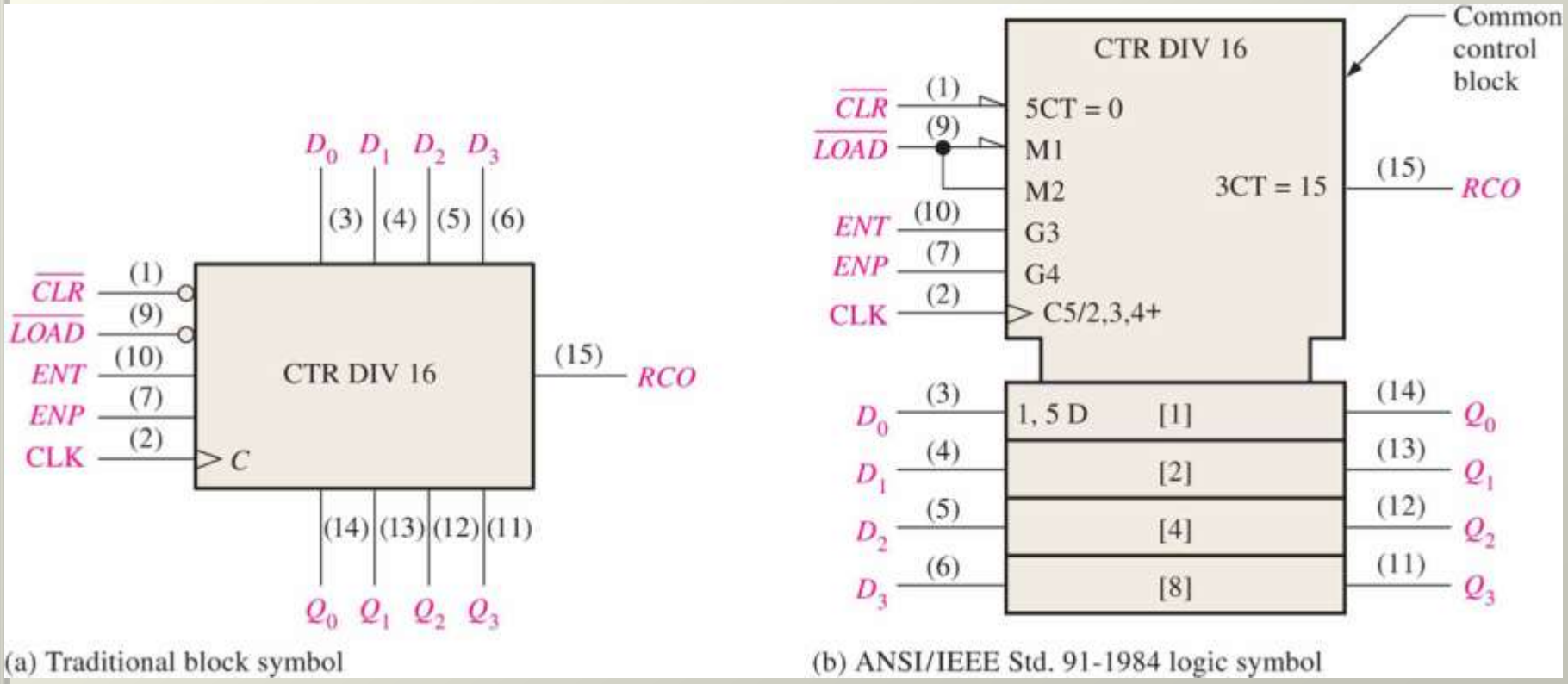


Logic diagram for modulus-100 up/down counter for automobile parking control.



8. Logic Symbols with Dependency Notation

Dependency notation is fundamental to ANSI/IEEE standard.
It is used in conjunction with the logic symbols to specify the relationships of inputs and outputs so that the logical operation of a given device can be determined entirely from its logic symbol.



Key Terms

Asynchronous : **Not occurring at the same time**

Modulus : **The number of unique states through which a counter will sequence.**

Synchronous : **Occurring at the same time.**

Terminal count : **The final state in a counter's sequence.**

State machine: **A logic system exhibiting a sequence of states or values.**

Cascade : **To connect "end-to-end" as when several counters are connected from the terminal count output of one to the enable input of the next counter.**

ANSI : The American National Standards Institute
IEEE: The Institute of Electrical and Electronics Engineers