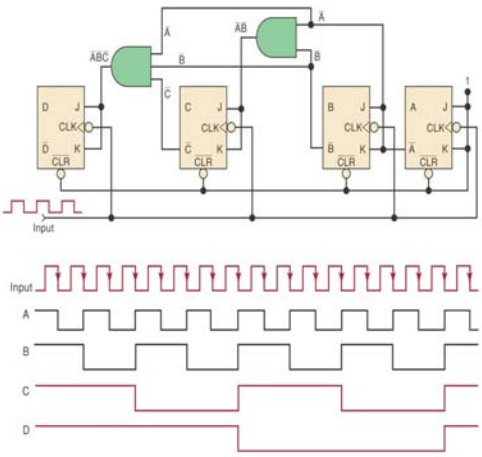
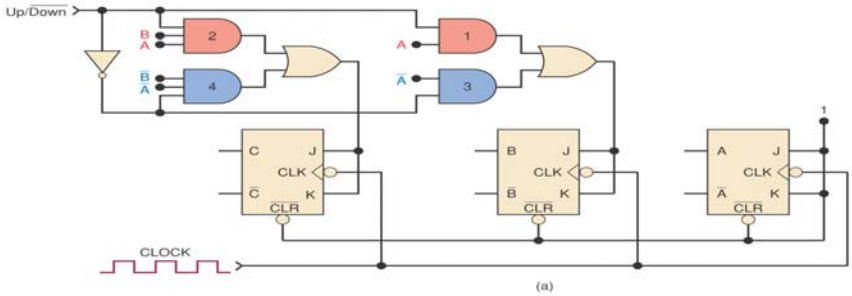
7-5 Synchronous Down and Up/Down Counters

- The synchronous counter can be converted to a down counter by using the inverted FF outputs to drive the JK inputs.
- FF B toggles when A = 0 on the next NGT CLK.
- The count sequence is 15, 14, 13, …, 1, 0, 15.



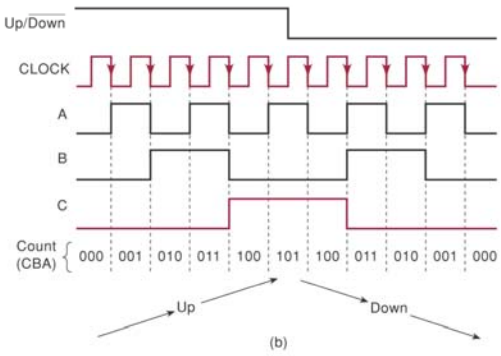
7-5 Synchronous Down and Up/Down Counters



- A synchronous counter can be made an up/down counter by connecting as illustrated.
- The control input Up/Down controls whether the normal FF outputs or the inverted FF output are fed to J and K inputs of the successive FFs.

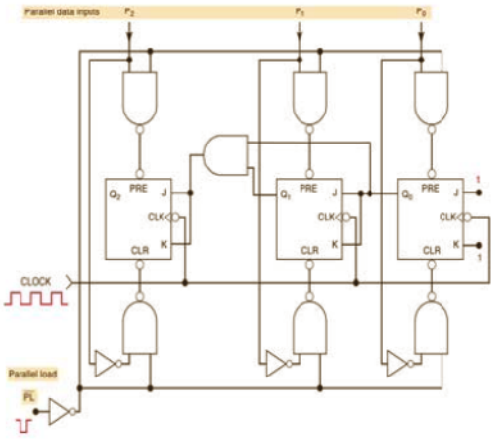
7-5 Synchronous Down and Up/Down Counters

- When the input Up/$\overline{\text{Down}}$ is High, the counter is UP, if it is Low, it is configured as DOWN counter.
- The UP/$\overline{\text{Down}}$ shown is High for 5 pulses (UP) and Low for 5 pulses (DOWN).



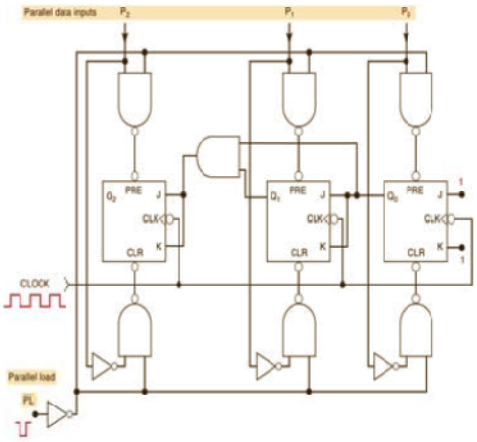
7-6 Presettable Counters

- A presettable counter can be set to any desired starting point either asynchronously or synchronously.
- The preset operation is also called parallel loading the counter.
- J,K, and CLK are wired as UP counter.
- The counter is loaded with any desired value P_2, P_1, P_0



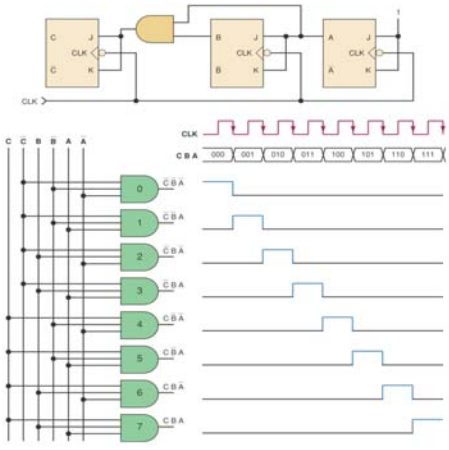
7-6 Presettable Counters

- A low pulse on PL* will perform asynchronous transfer into FF Q_2, Q_1, Q_0. The effect of CLK will be disabled as long as PL* is active, when PL* is inactive, counter is Synchronous.
- If P_2=1, P_1=0, P_0=1 and PL* is active, Q2 will be set, Q1 will be cleared, and Q0 will be set which will be held as long as PL* is active,
- J,K, and CLK are wired as UP counter.
- The counter is loaded with any desired value P2, P1, P0



7-8 Decoding a Counter

- Decoding is the conversion of a binary output to a decimal value.
- The active high decoder could be used to light an LED representing each decimal number 0 to 7.
- Each AND gate produces a HIGH output for one particular state, AND 0 represents State 0.
- The 8 AND gates can be used to control 8 LED
- Active low decoding is obtained by replacing the AND gates with NAND gates.

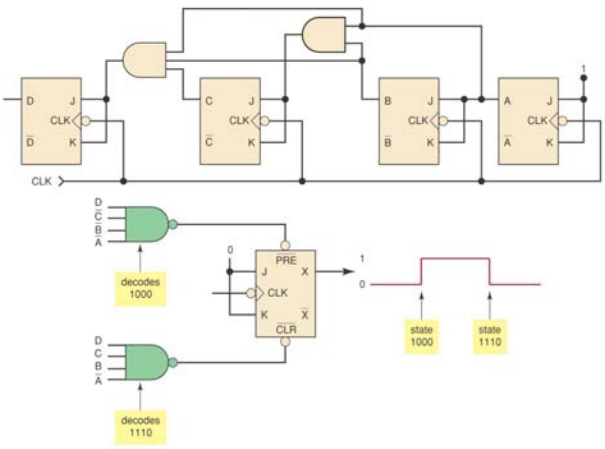


Example 7 -14

- How many AND gates are required to decode completely all the states of a MOD-32 binary counter? What are the inputs to the gate that decodes for the count of 21?
- MOD-32 has 5 FFs, decoder requires 32 AND gates with 5-input, $21_{10}=10101_2$, inputs are E, D*, C, B*, and A , E is the MSB

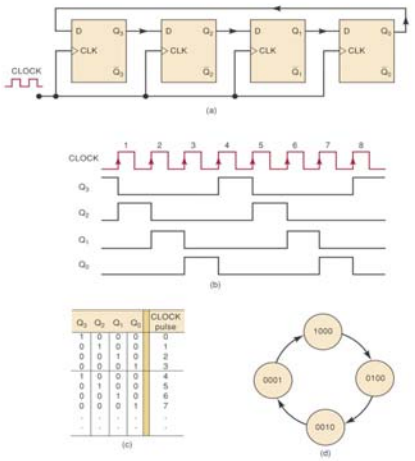
Example

- Explain the operation of the following counter?
- X will be HIGH between the count of 8 and 14
- X is a control waveform used to control devices such as a motor



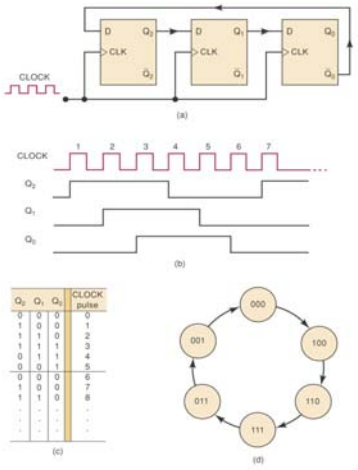
7-20 Shift Register Counters

- Ring Counter
  - Last FF shifts its value to first FF
  - Uses D-type FFs (JK FFs can also be used)
  - In most instances, start with only one FF in the 1 state and all others in the 0 state. The waveforms, state diagram and sequence table is shown.
  - Information shifts from left to right and back around from Q0 to Q3
  - Can function as MOD-4 counter since it has 4 distinct states. How many FFs are needed to construct MOD-8 ring counter.
  - Can be decoded without decoing logic.



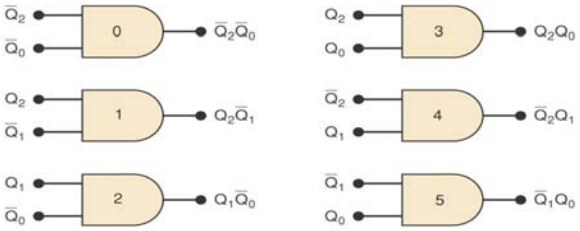
7-20 Shift Register Counters

- Johnson counter
  - Also called a twisted ring counter
  - Same as ring counter but the inverted output of the last FF is connected to input of the first FF
  - On each positive CLK, Q2 is shifted to Q1, Q1 is shifted to Q0, and inverse of Q0 is shifted into Q2
  - Assuming all FFs are initially 0s, the counter has 6 states – 000, 100, 110, 111,011, 001. it is MOD-6 (not normal binary count)
  - The waveform of each FF is a square wave (50% duty cycle) and 1/6 of the frequency.
  - The MOD number is twice the number of FFs



7-20 Shift Register Counters

- Johnson counter
  - For a given MOD number, A Johnson counter requires ½ FFs compared to ring counter. On the other hand, it does require decoding gates.
  - Decoding logic requires only 2-input since in each count, two of the three FFs are in a unique combination of states as shown.



Q_2	Q_1	Q_0	Active gate
0	0	0	0
1	0	0	1
1	1	0	2
1	1	1	3
0	1	1	4
0	0	1	5