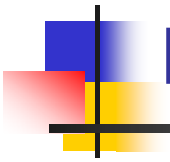
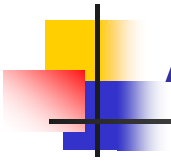


Modular Sequential Logic



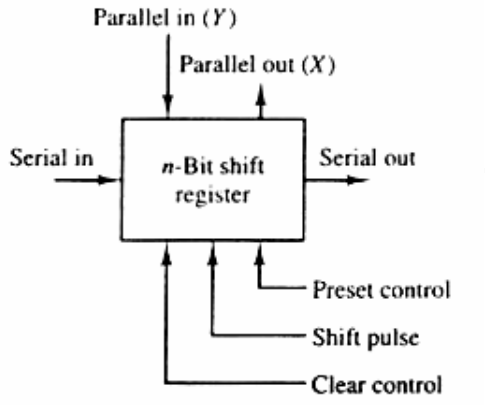
Agenda

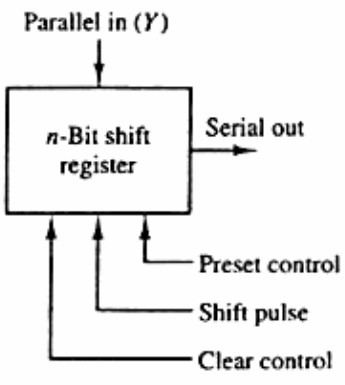
- Shift Register
- Design Examples Using Register
- Counters
- Modulo-N Counter

Shift Register

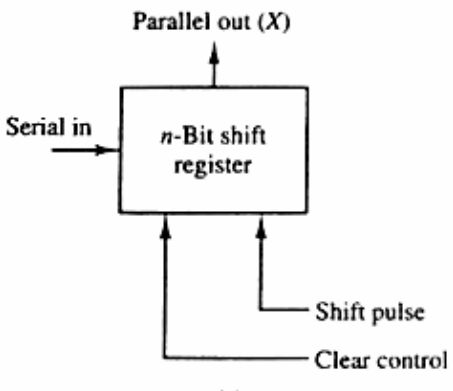
- A Shift Register is a sequential logic module constructed from flip-flop.
- That manipulates the bit position of binary data by shifting data bits to the left or right.

Shift Register

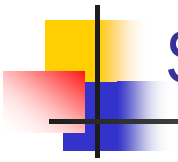




(b)

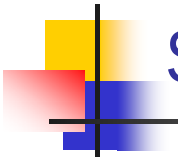


(c)



Shift Register

- **Parallel in** : one input line for each flip-flop with data to be entered into the register.
- **Parallel out** : one input line coming from each flip-flop Q terminal.
- **Shift pulse (CLOCK)** : a pulse on this control line makes the data in the register move over one cell.

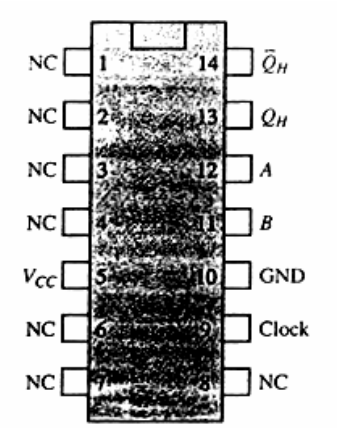


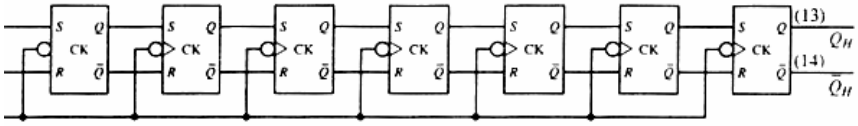
Shift Register

- **Serial in** : data line feeding the first cell in the shift register; a bit enters on each Shift pulse.
- **Serial out** : data line from the Q terminal of the last flip-flop in the register; a bit exits on each Shift pulse.
- **Clear** : a pulse on this line drives all the flip-flop in the register to logic 0.
- **Preset** : a pulse on this line drives all the flip-flop in the register to logic 1.

Shift Register

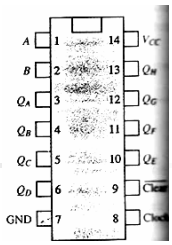
- 7491 (SISO)



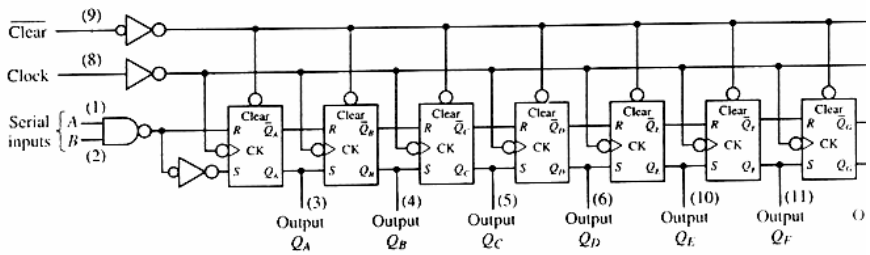


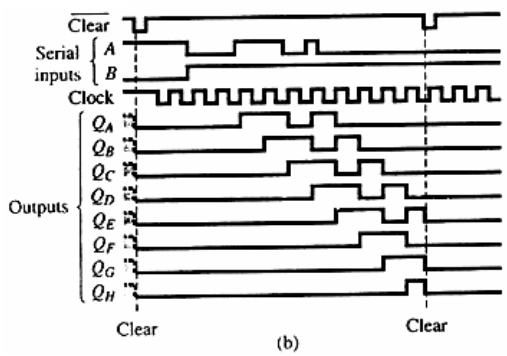
Inputs at t_n		Outputs at $t_n + 8$	
A	B	Q_H	$\bar{Q}_H$
H	H	H	L
L	×	L	H
×	L	L	H

Shift Register



- 74164 (SI S/P O)

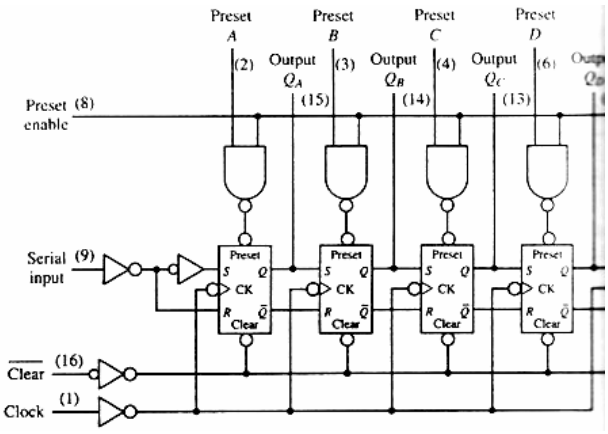




Inputs				Outputs			
$\overline{\text{Clear}}$	Clock	A	B	Q_A	Q_B	…	Q_H
L	×	×	×	L	L		L
H	L	×	×	Q_{A0}	Q_{B0}		Q_{H0}
H	↑	H	H	H	Q_{An}		Q_{Gn}
H	↑	L	×	L	Q_{An}		Q_{Gn}
H	↑	×	L	L	Q_{An}		Q_{Gn}

Shift Register

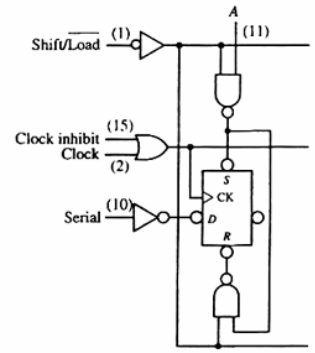
- 7496 (SI S/P O)

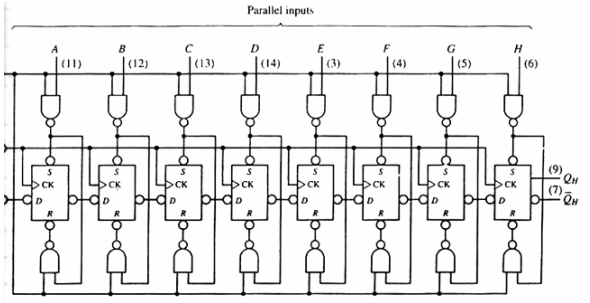


Inputs								Outputs				
		Preset										
$\overline{\text{Clear}}$	Preset Enable	A	B	C	D	E	Clock	Serial	Q_A	Q_B	Q_C	Q_D	Q_E
L	L	×	×	×	×	×	×	×	L	L	L	L	L
L	×	L	L	L	L	L	×	×	L	L	L	L	L
H	H	H	H	H	H	H	×	×	H	H	H	H	H
H	H	L	L	L	L	L	L	×	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}	Q_{E0}
H	H	H	L	H	L	H	L	×	H	Q_{B0}	H	Q_{D0}	H
H	L	×	×	×	×	×	L	×	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}	Q_{E0}
H	L	×	×	×	×	×	↑	H	H	Q_{An}	Q_{Bn}	Q_{Cn}	Q_{Dn}
H	L	×	×	×	×	×	↑	L	L	Q_{An}	Q_{Bn}	Q_{Cn}	Q_{Dn}

Shift Register

- 74165 (SI S/P O with Load)

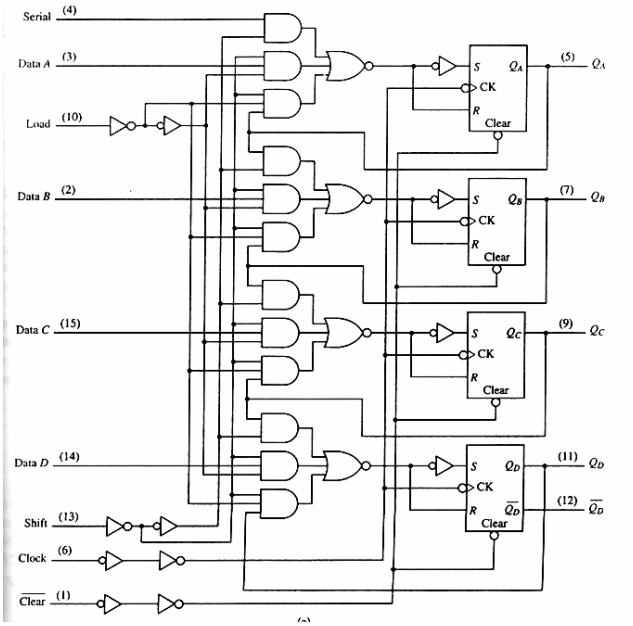




Inputs					Internal outputs		Output
Shift/$\overline{\text{load}}$	Clock inhibit	Clock	Serial	Parallel $A \ldots H$	Q_A	Q_B	Q_H
L	×	×	×	$a \ldots h$	a	b	h
H	L	L	×	×	Q_{A0}	Q_{B0}	Q_{H0}
H	L	↑	H	×	H	Q_{An}	Q_{Gn}
H	L	↑	L	×	L	Q_{An}	Q_{Gn}
H	H	×	×	×	Q_{A0}	Q_{B0}	Q_{H0}

Shift Register

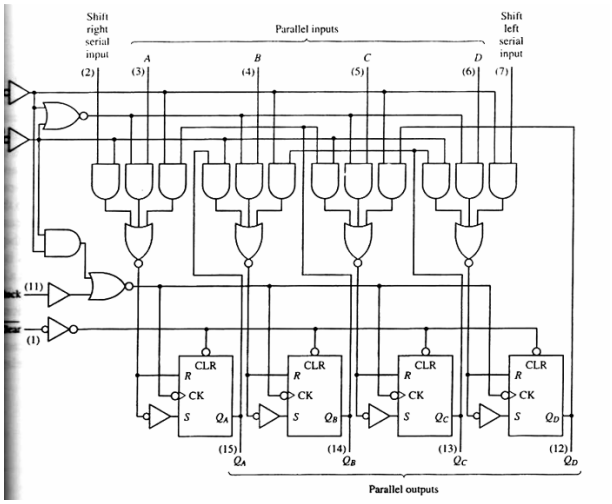
- 74179 (SI S/P O with Load)



Inputs			Mode
$\overline{\text{Clear}}$	Shift	Load	
L	×	×	Asynchronous clear
H	H	×	Shift
H	L	H	Synchronous load
H	L	L	Data hold

Shift Register

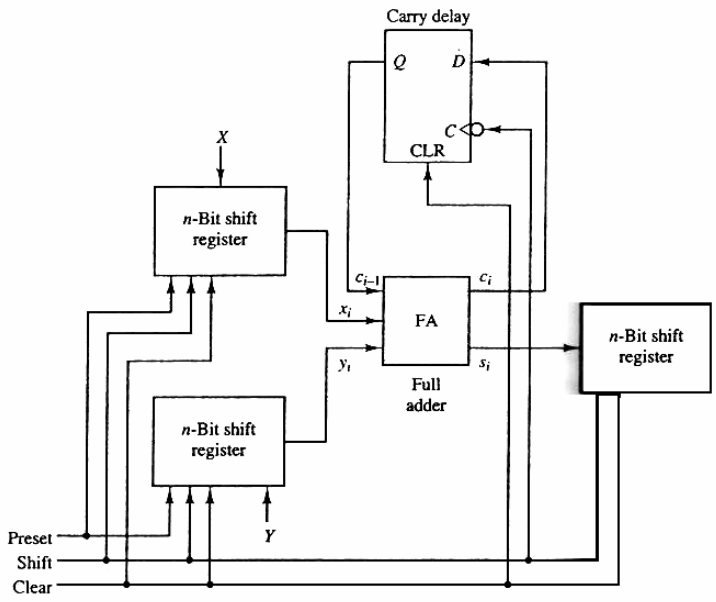
- 74165 (SI S/P O with Load)



Inputs				Internal signals ($i = A, B, C, D$)	Mode
$\overline{\text{Clear}}$	$S0$	$S1$	Clock	S_i	
L	×	×	×	×	Asynchronous clear
H	L	L	×	×	Clock inhibit (data hold)
H	L	H	↑	Q_{i+1}	Shift left
H	H	L	↑	Q_{i-1}	Shift right
H	H	H	↑	i	Parallel load

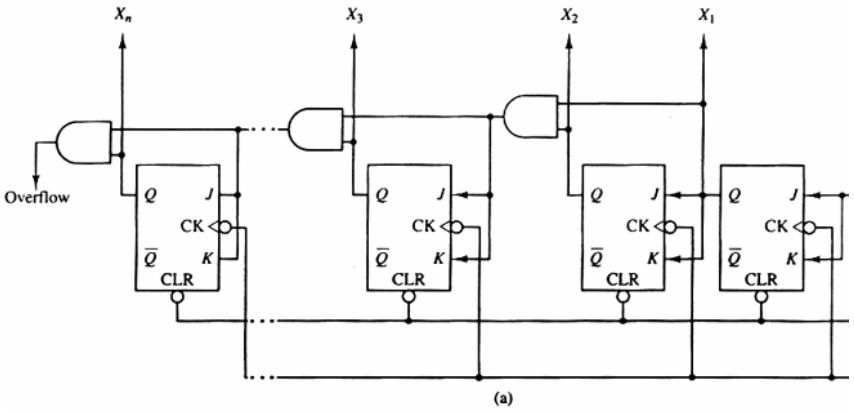
Design Examples Using Register

- Serial Adder Unit



Counters

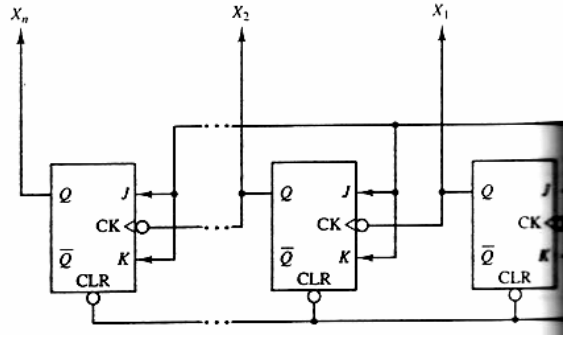
- Synchronous Binary Counters

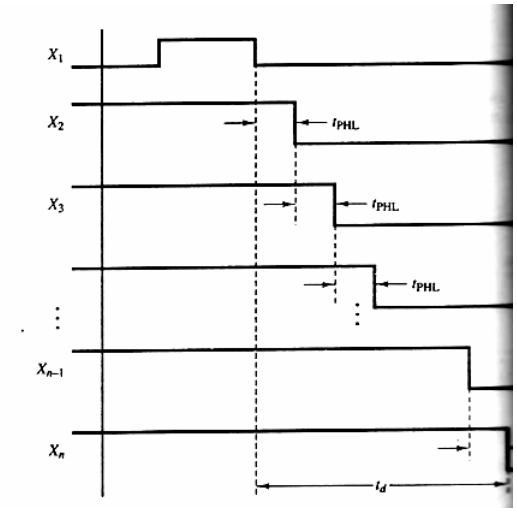


X_n	...	X_3	X_2	X_1	
0	...	0	0	0	
0	...	0	0	1	
0	...	0	1	0	
0	...	0	1	1	
0	...	1	0	0	
		⋮			
1	...	1	1	0	
1	...	1	1	1	← Recycles
0	...	0	0	0	
0	...	0	0	1	
0	...	0	1	0	

Counters

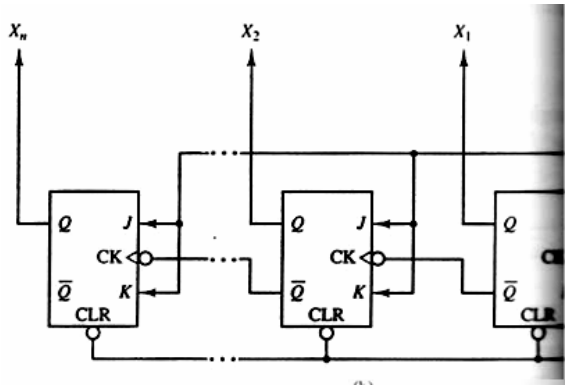
- Asynchronous Binary Counters





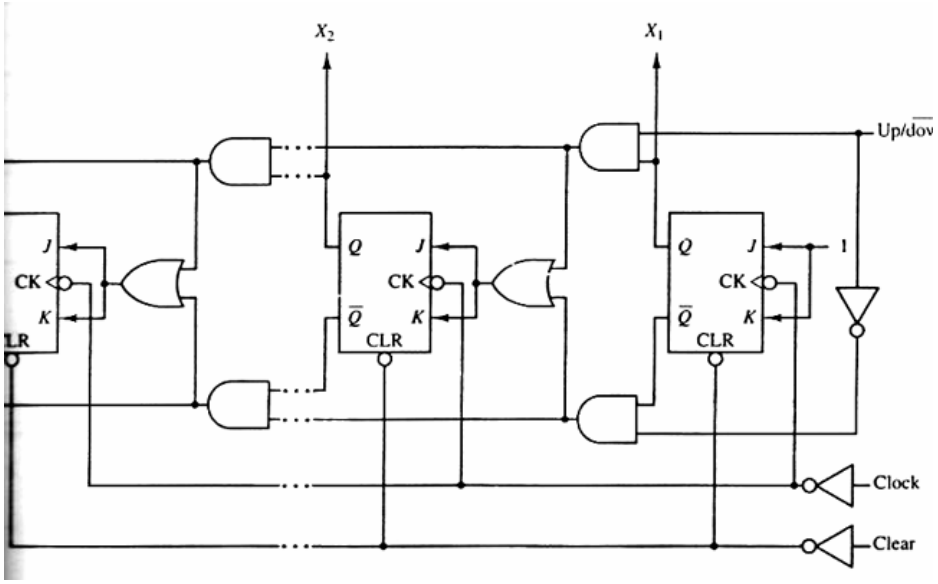
Counters

- Down Counters



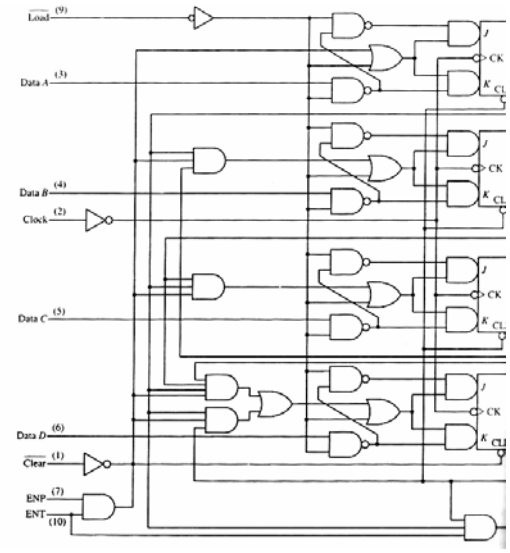
Counters

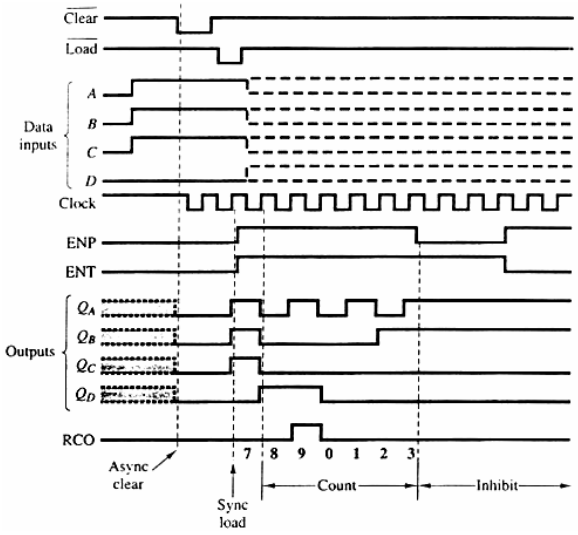
- Up/Down Counters



Modulo-N Counters

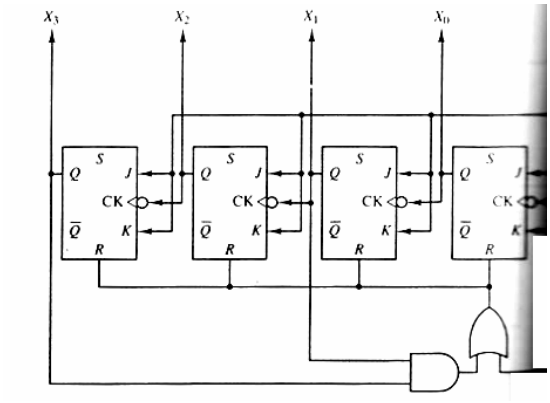
- Synchronous BCD Counters

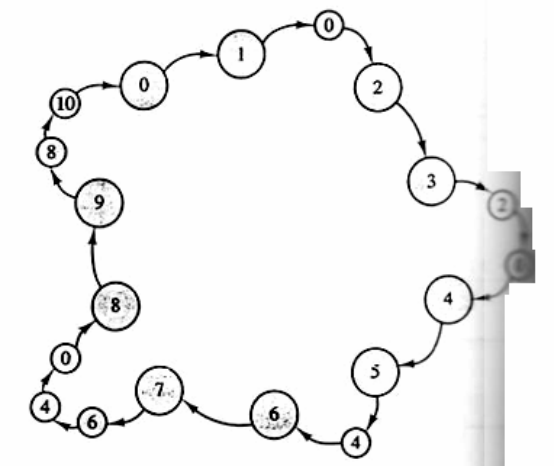




Modulo-N Counters

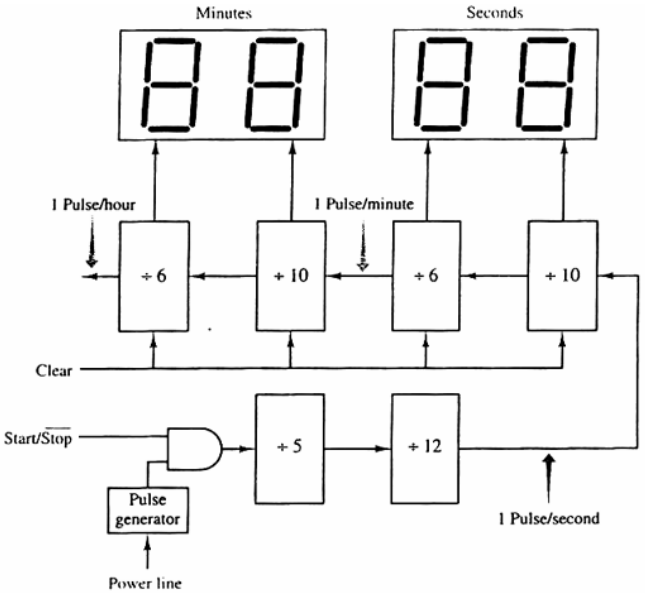
- Asynchronous BCD Counters





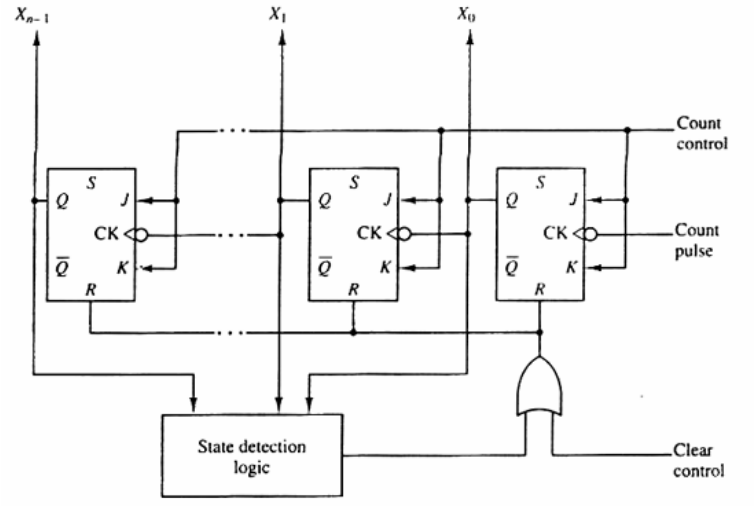
Modulo-N Counters

- Modulo-6 and 12 Counter



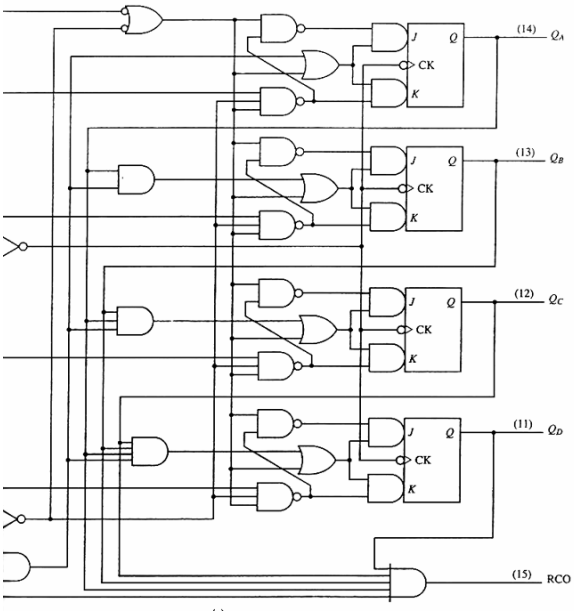
Modulo-N Counters

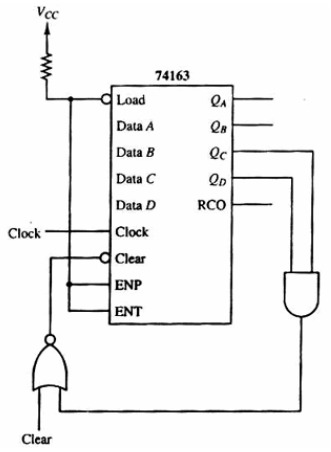
- Asynchronously Resetting Modulo-N Counters



Modulo-N Counters

- Synchronously Resetting Modulo-N Counters





Modulo-N Counters

Design a 13-state ring counter with active low outputs using a counter and a decoder. Since

$$2^4 > 13 > 2^3$$

the desired ring counter can be constructed from a modulo-13 counter and a 4-to-16 decoder.