Digital Design
Lectures 11 & 12

Shift Registers and Counters

Register with Parallel Load

- Register: Group of Flip-Flops
- Ex: D Flip-Flops
- Holds a Word (Nibble) of Data
- Loads in Parallel on Clock Transition
- Asynchronous Clear (Reset)

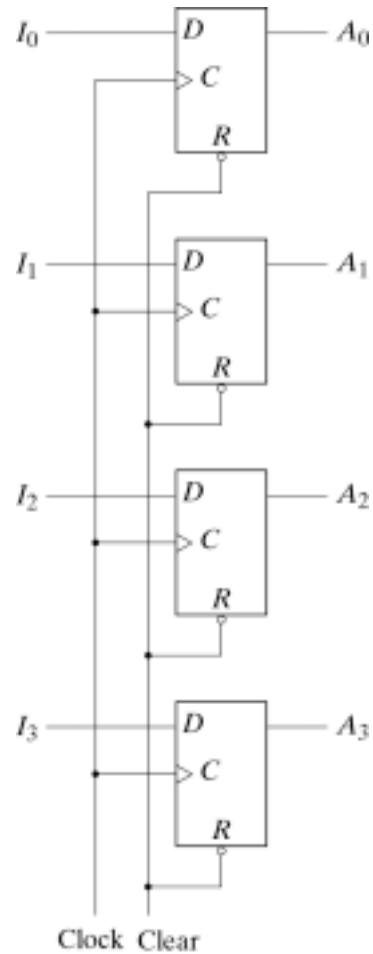


Fig. 6-1 4-Bit Register

Register with Load Control

- Load Control = 1
  - New data loaded on next positive clock edge
- Load Control = 0
  - Old data reloaded on next positive clock edge

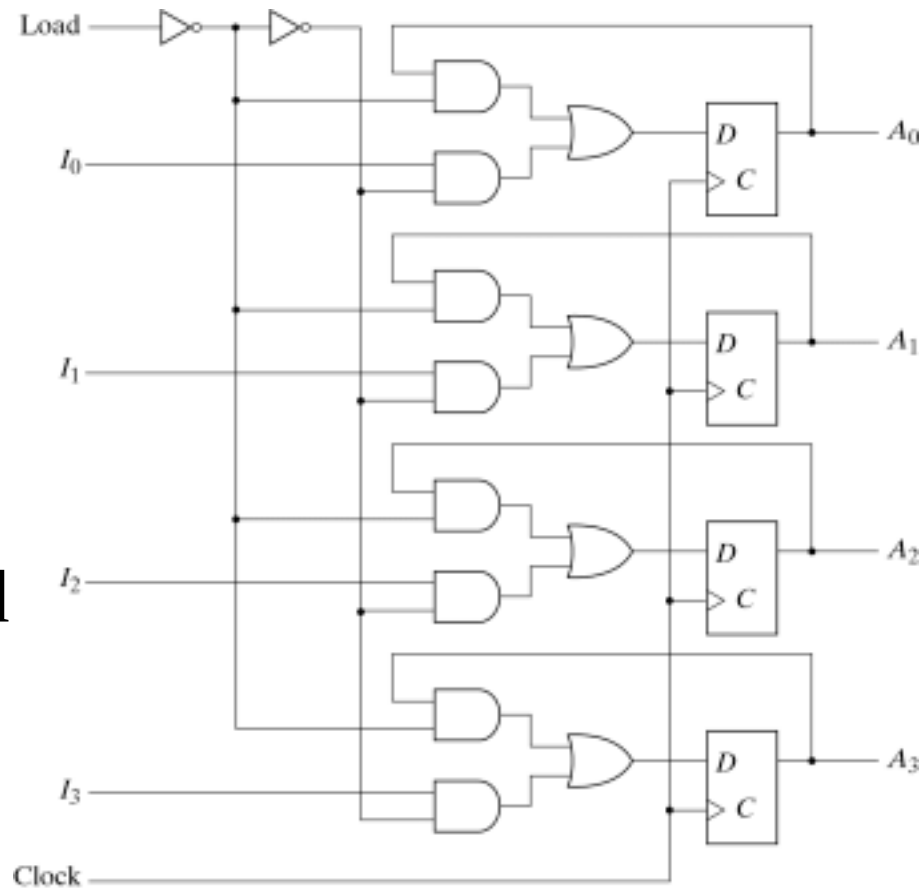


Fig. 6-2 4-Bit Register with Parallel Load

Shift Registers

- Cascade chain of Flip-Flops
- Bits travel on Clock edges (Bucket Brigade)
- Serial in – Serial out, can also have parallel load / read

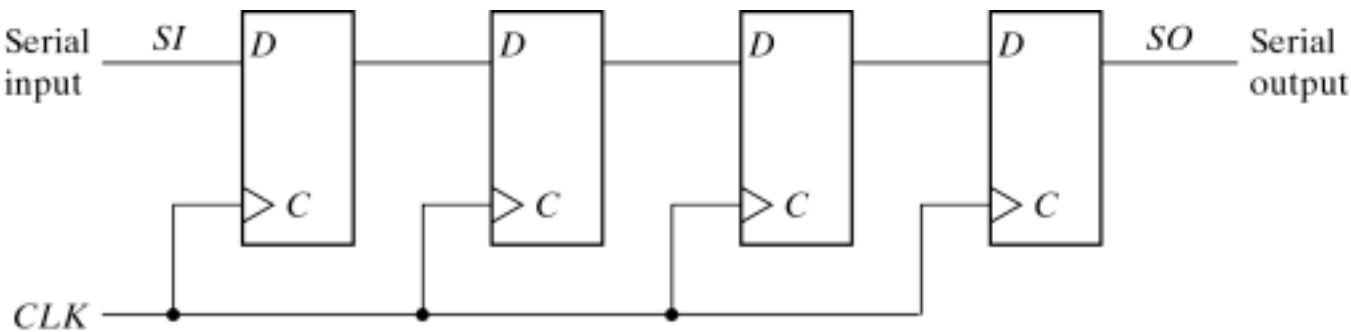


Fig. 6-3 4-Bit Shift Register

Serial Transfer

Time	Reg A	Reg B
T0	1011	0011
T1	1101	1001
T2	1110	1100
T3	0111	0110
T4	1011	1011

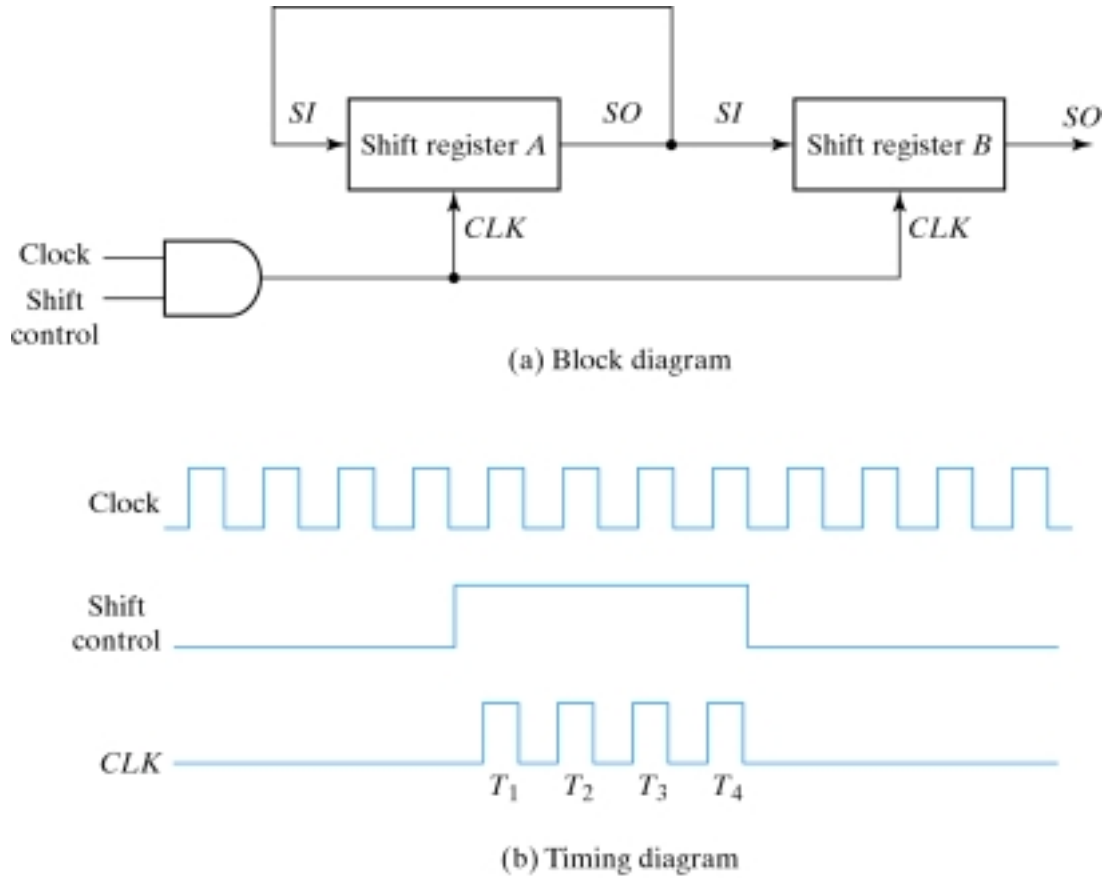


Fig. 6-4 Serial Transfer from Register *A* to register *B*

Serial Addition (D Flip-Flop)

- Slower than parallel
- Low cost
- Share fast hardware on slow data
- Good for multiplexed data

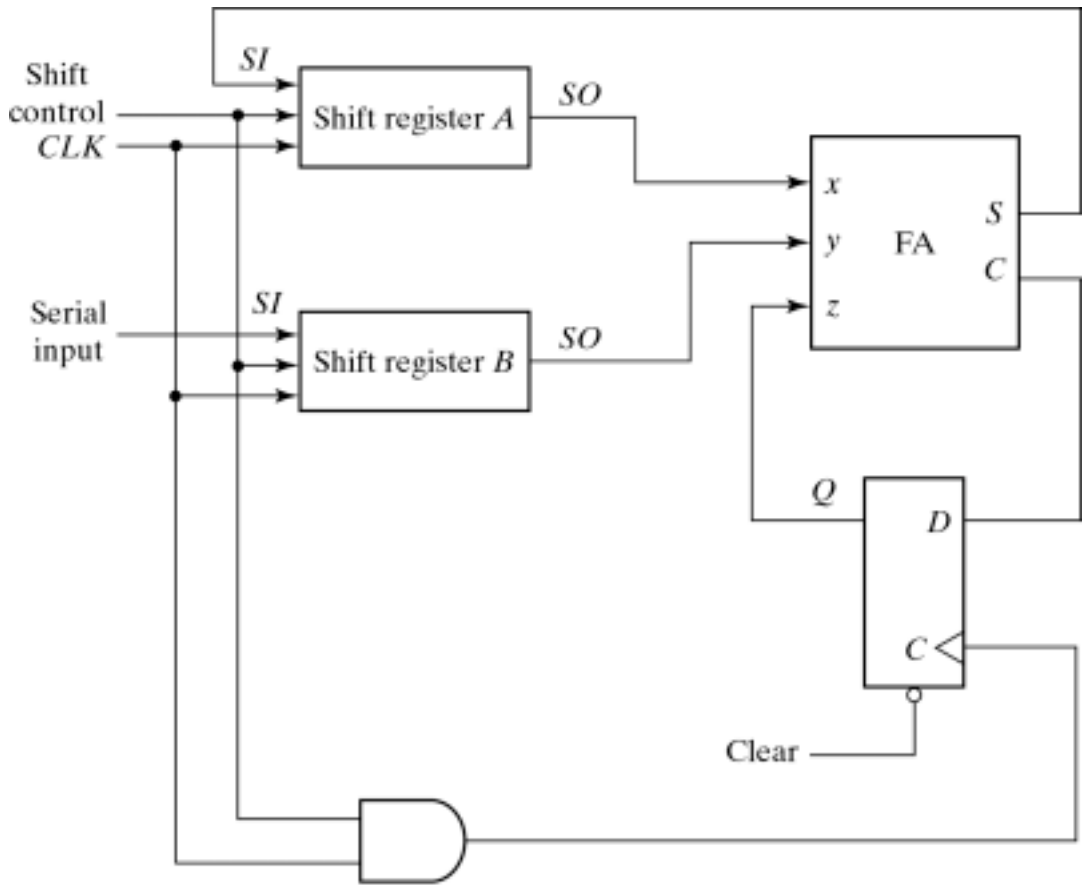


Fig. 6-5 Serial Adder

Designing a JK Serial Adder

- $J_Q = x \cdot y$, $K_Q = x' \cdot y' = (x+y)'$, $S = x \oplus y \oplus Q$

Present State	Inputs		Next State	Output	Flip_Flop Inputs	
Q	x	y	Q	S	J_Q	K_Q
0	0	0	0	0	0	x
0	0	1	0	1	0	x
0	1	0	0	1	0	x
0	1	1	1	0	1	x
1	0	0	0	1	x	1
1	0	1	1	0	x	0
1	1	0	1	0	x	0
1	1	1	1	1	x	0

Table 6-2

New Serial Adder

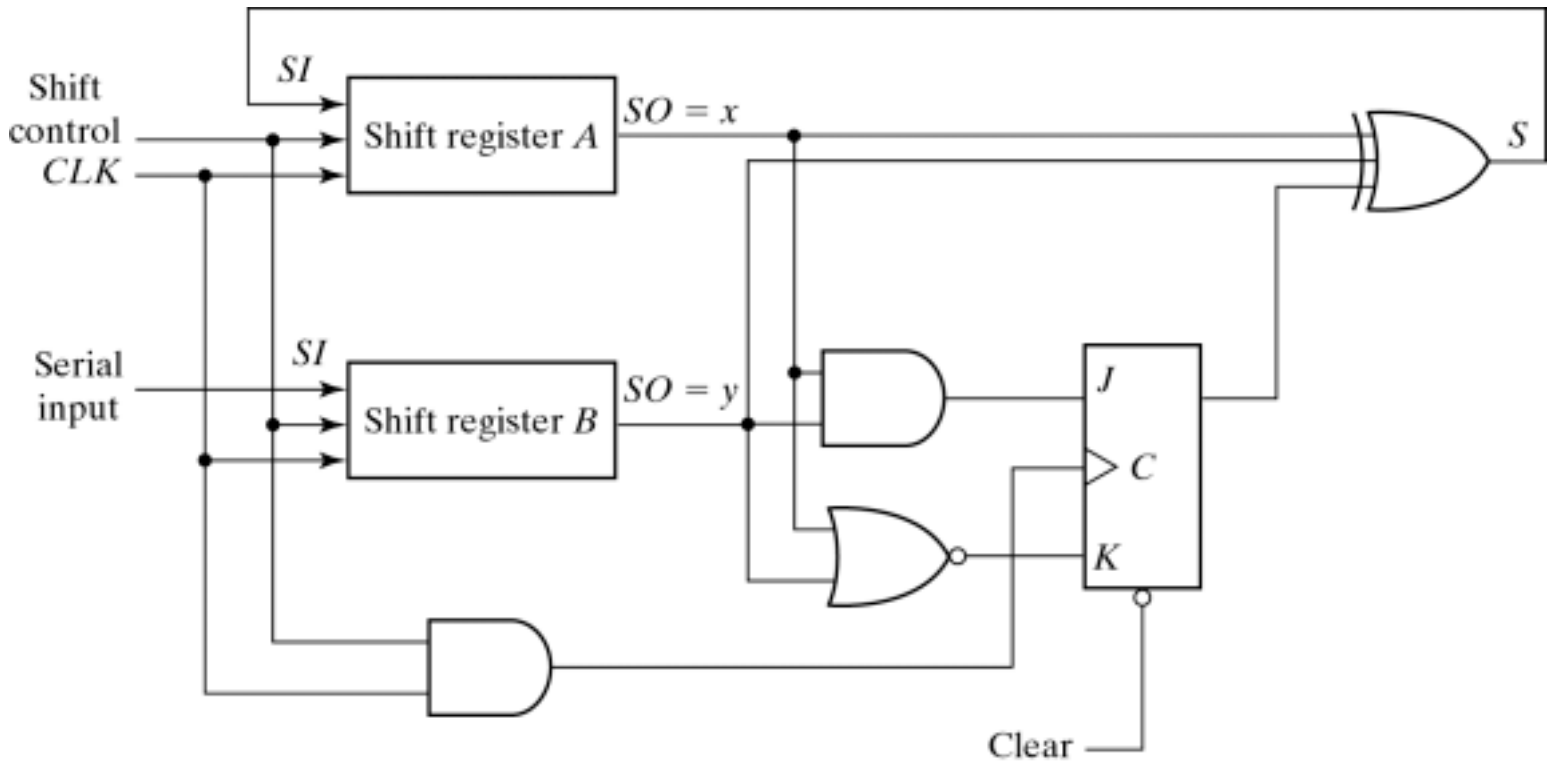


Fig. 6-6 Second form of Serial Adder

Universal Shift Register

- Clear
- Clock
- Shift
  - Right
  - Left
- Load
- Read
- Control

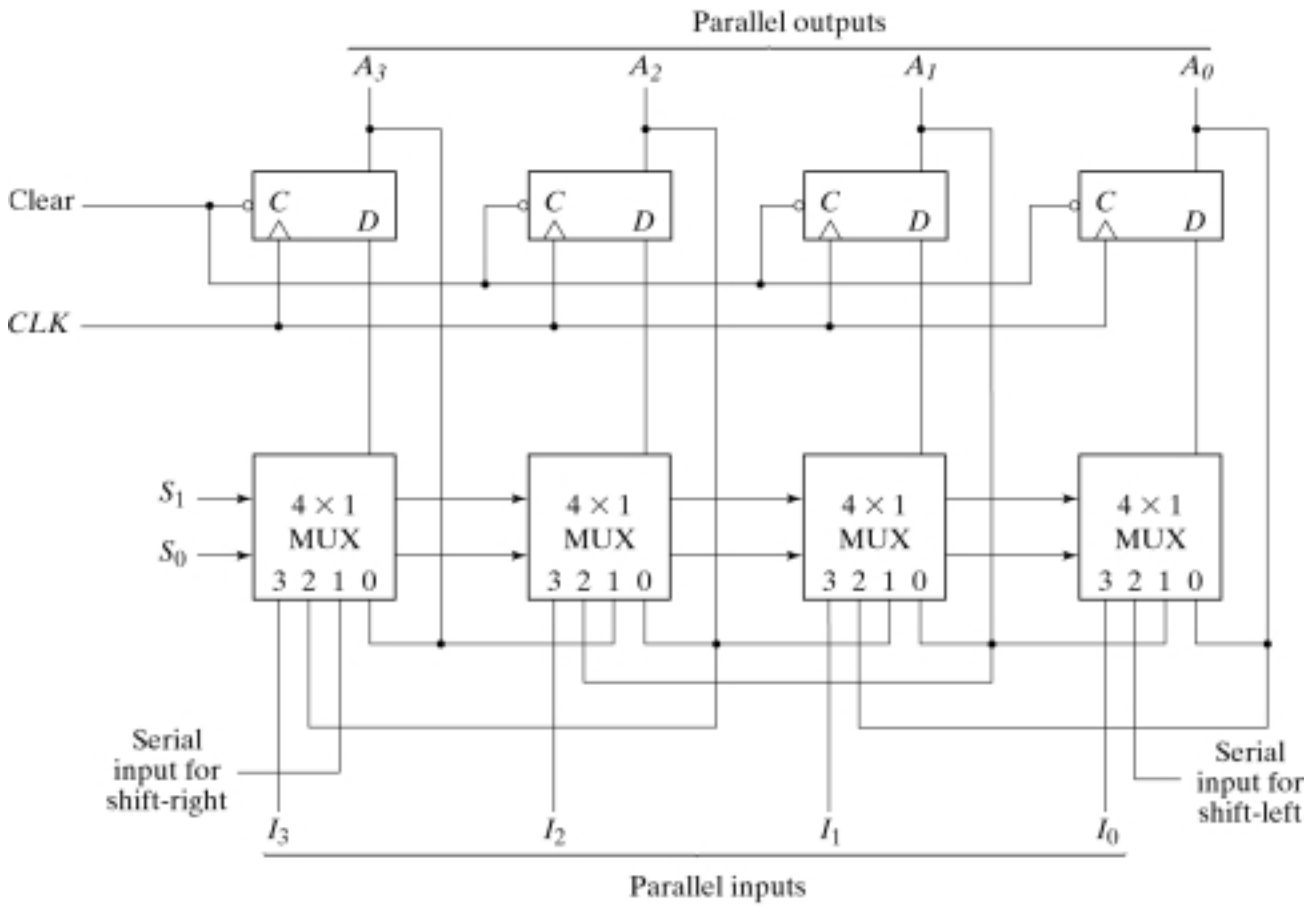


Fig. 6-7 4-Bit Universal Shift Register

S1
0
0
1
1

Binary Ripple Counter

- Asynchronous counter
- Changes “ripple” through the stages

Table 6-4

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

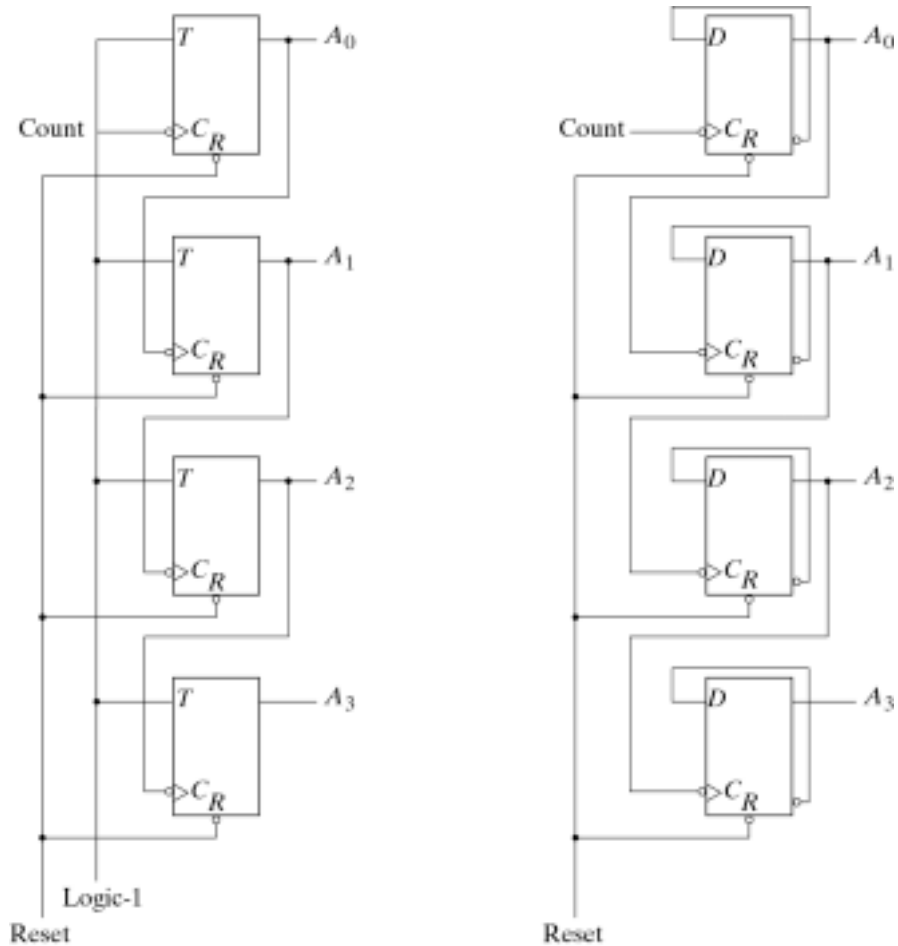


Fig. 6-8 4-Bit Binary Ripple Counter

Decimal (BCD) Counter

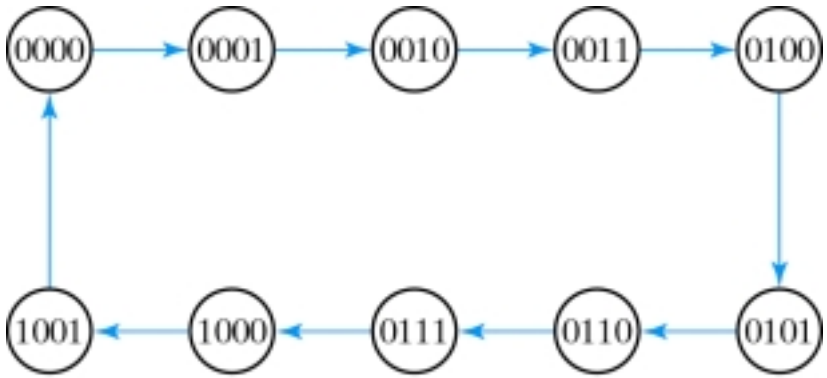


Fig. 6-9 State Diagram of a Decimal BCD-Counter

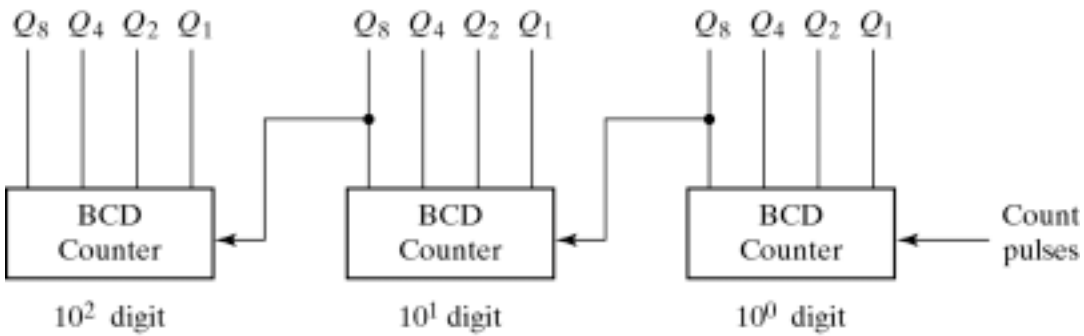


Fig. 6-11 Block Diagram of a Three-Decade Decimal BCD Counter

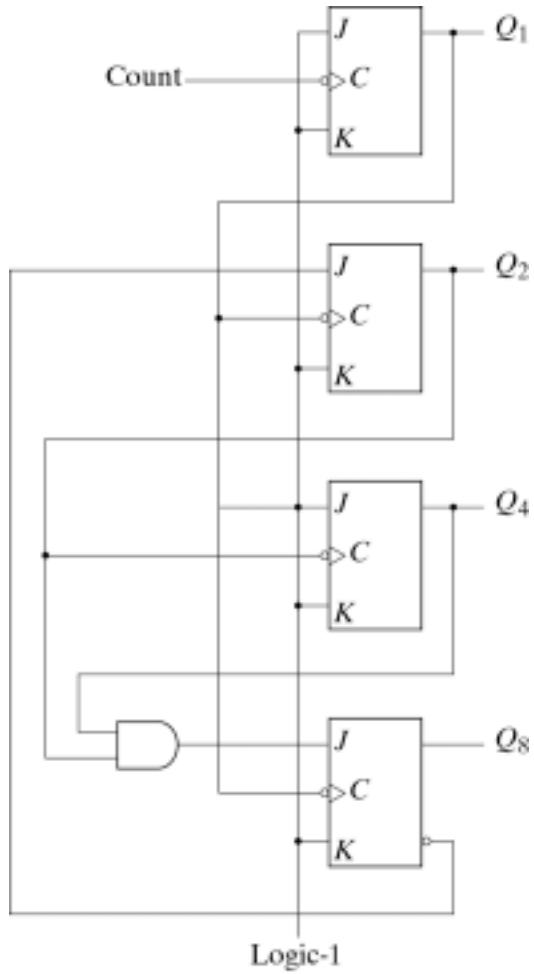


Fig. 6-10 BCD Ripple Counter

Synchronous Binary Counter

- All transitions occur on clock pulse edges in parallel
- Faster results than ripple counters

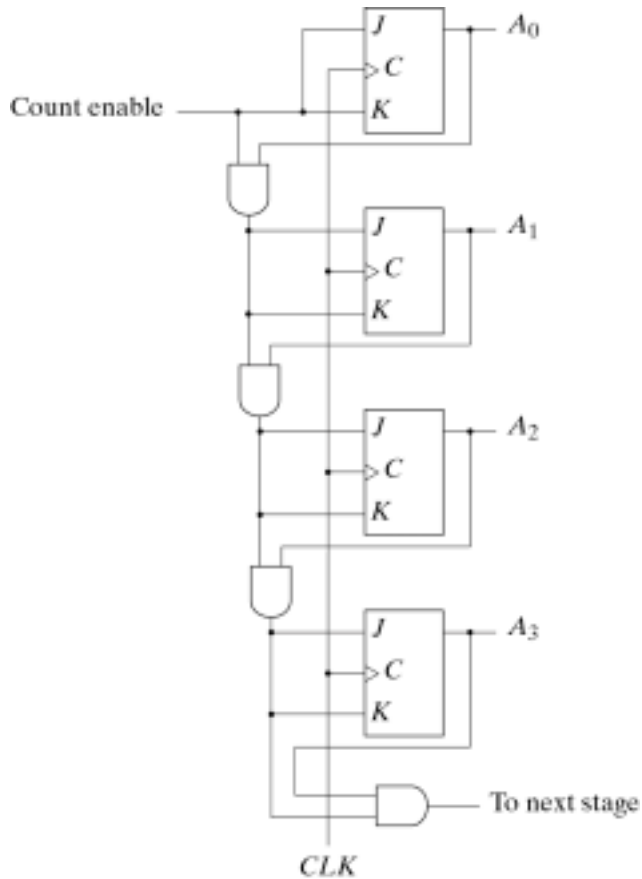


Fig. 6-12 4-Bit Synchronous Binary Counter

Up-Down Synchronous Counter

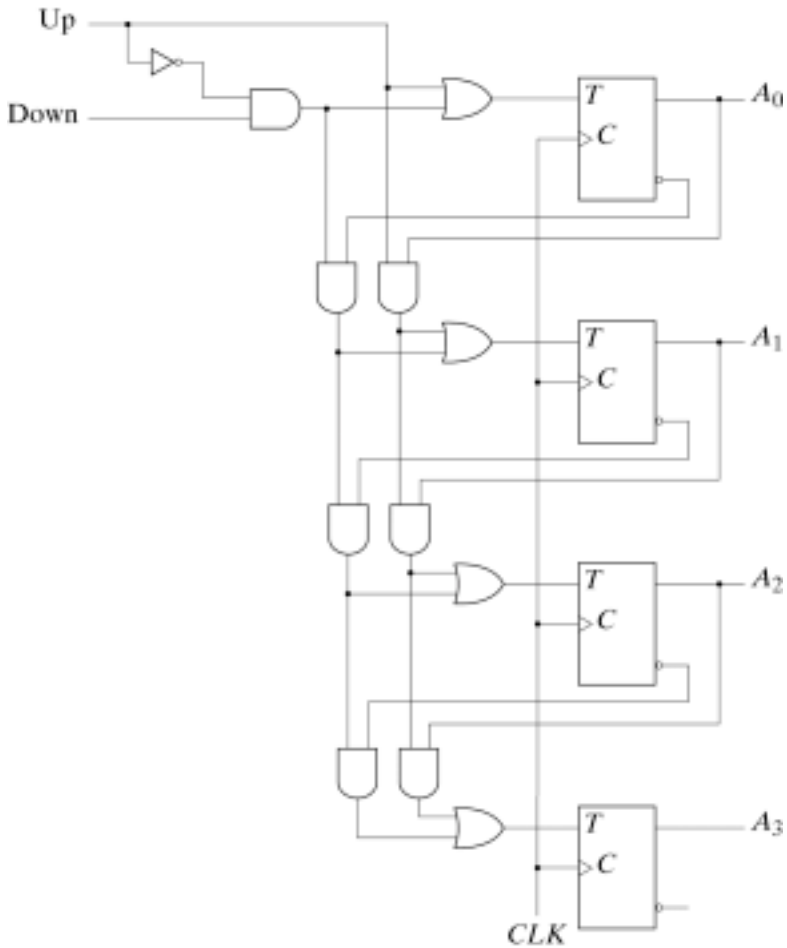


Fig. 6-13 4-Bit Up-Down Binary Counter

Designing a Synchronous BCD Counter

Present State				Next State				Output	Next State			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	TQ_8	TQ_4	TQ_2	TQ_1
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	1	0	0	1	1	0	0	0	0	1	1
0	1	0	1	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1.	1	1
1	0	0	0	1	0.	0	1	0	0	0	0	1
1	0.	0	1	0	0	0	0	1	1	0	0	1

$TQ_1 = 1$, $TQ_2 = Q'_8Q_1$, $TQ_4 = Q_2Q_1$, $TQ_8 = Q_8Q_1 + Q_4Q_2Q_1$, $y = Q_8Q_1$

4-bit Synchronous Counter with Parallel Load

Clear	Clk	Load	Count	Function
0	x	x	x	Clear
1	↑	1	x	Load
1	↑	0	1	Count
1	↑	0	0	No Change

Table 6-6

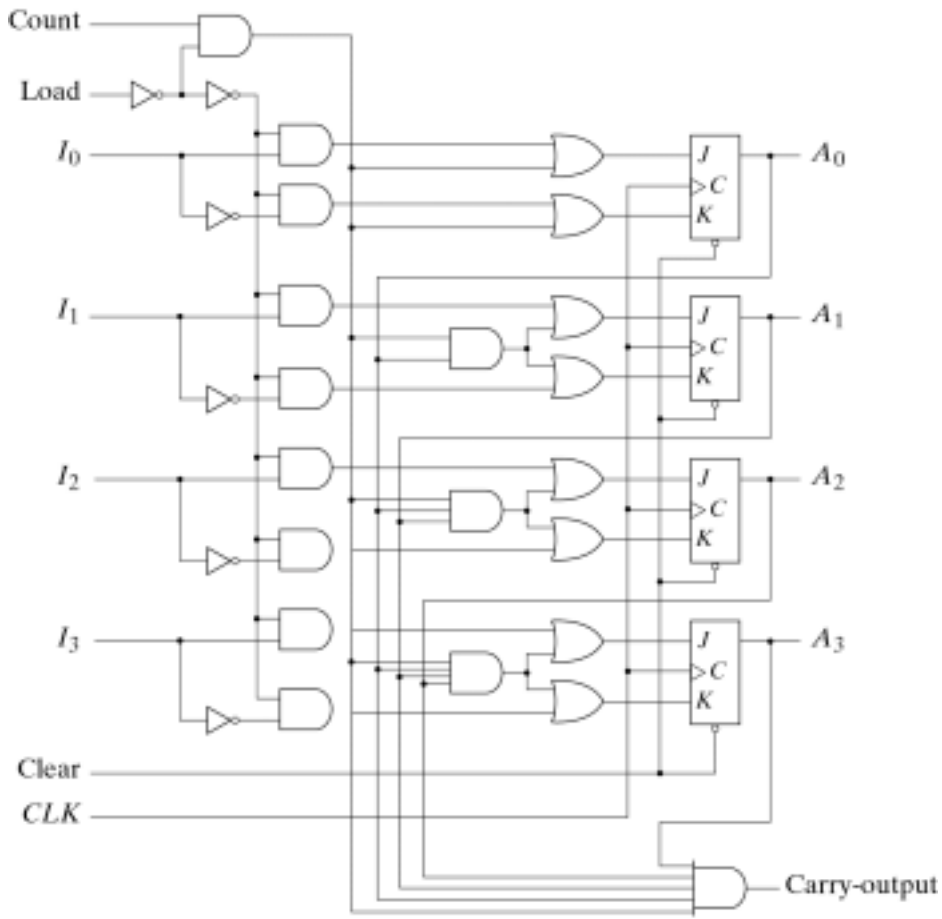


Fig. 6-14 4-Bit Binary Counter with Parallel Load

More BCD Counters with Parallel Load

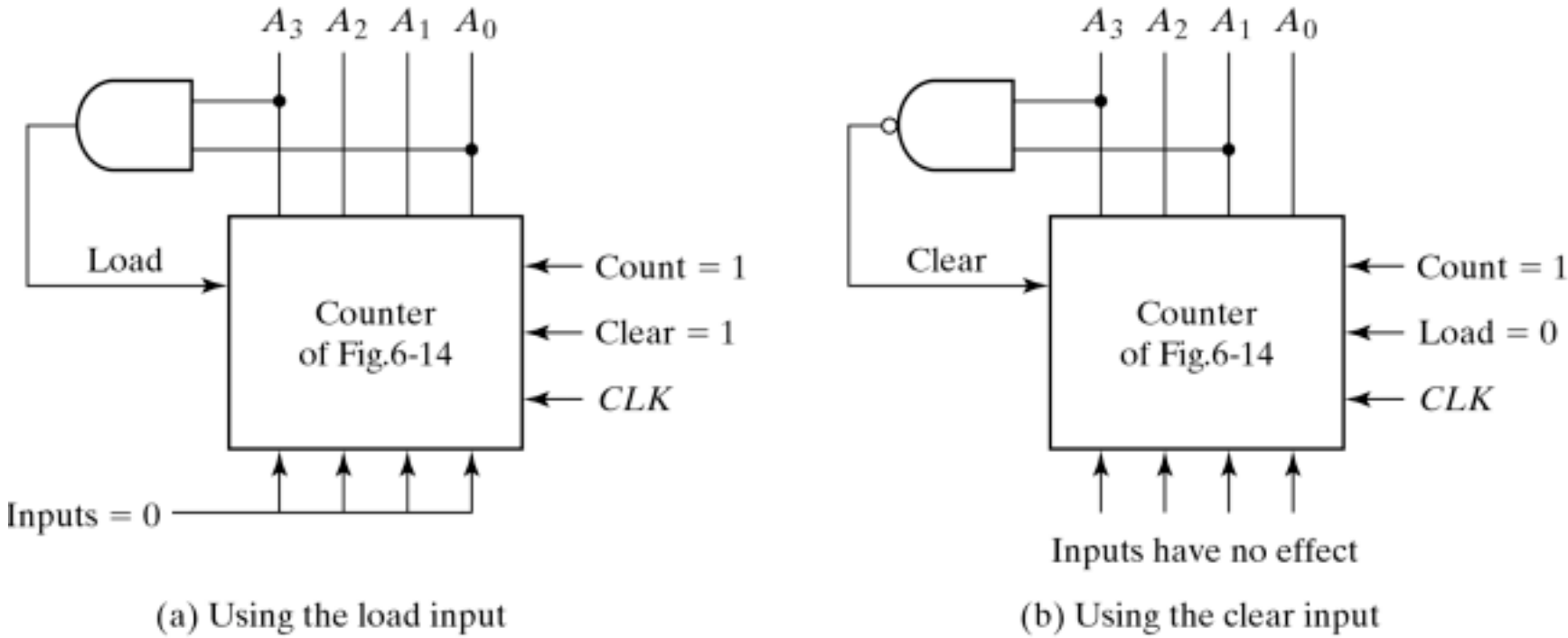
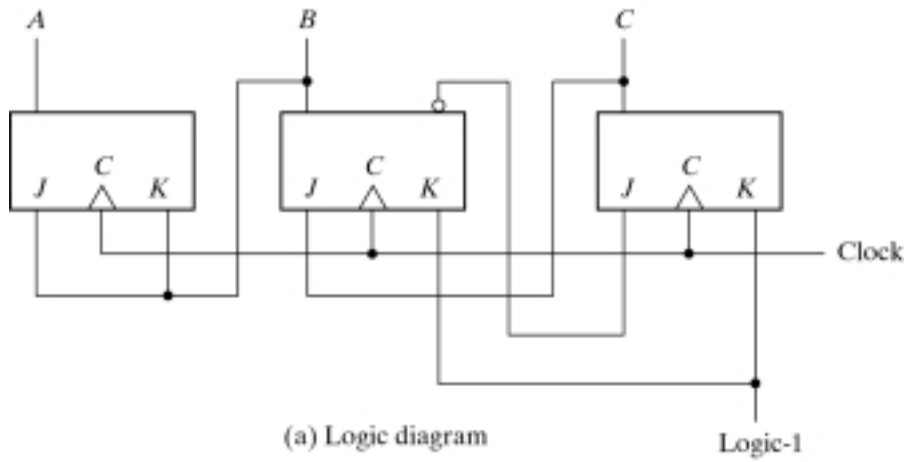


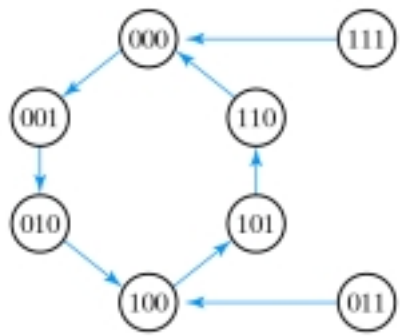
Fig. 6-15 Two ways to Achieve a BCD Counter Using a Counter with Parallel Load

Counters with Unused States

- If in unused state
  - Ensure automatic return to a valid state
  - "Self-correcting"



(a) Logic diagram

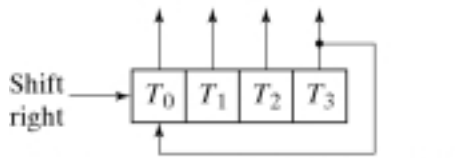


(b) State diagram

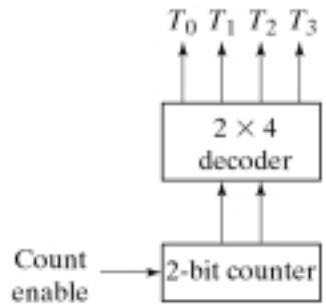
Fig. 6-16 Counter with Unused States

Generating Timing Pulses

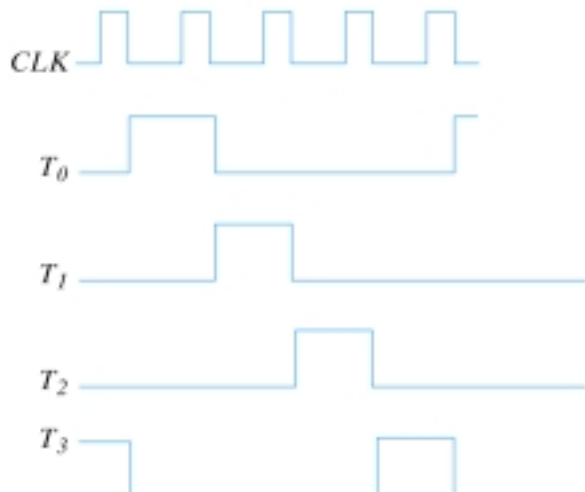
- Shift Register
  - Walk a 1
- Counter and decoder



(a) Ring-counter (initial value = 1000)



(b) Counter and decoder

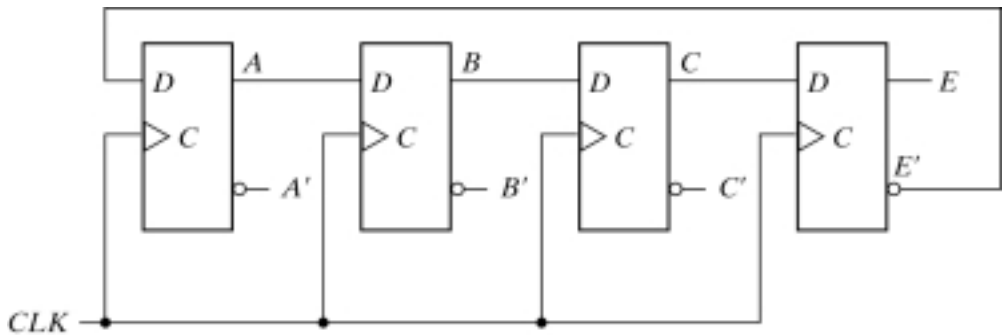


(c) Sequence of four timing signals

Fig. 6-17 Generation of Timing Signals

Johnson Counter

- Shift register
- 2*k States
  - Walk 1s
  - Then walk 0s
- Only needs simple output logic for timing pulses (Mealy model)



(a) Four-stage switch-tail ring counter

Sequence number	Flip-flop outputs A	B	C	E	AND gate required for output
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

(b) Count sequence and required decoding

Fig. 6-18 Construction of a Johnson Counter