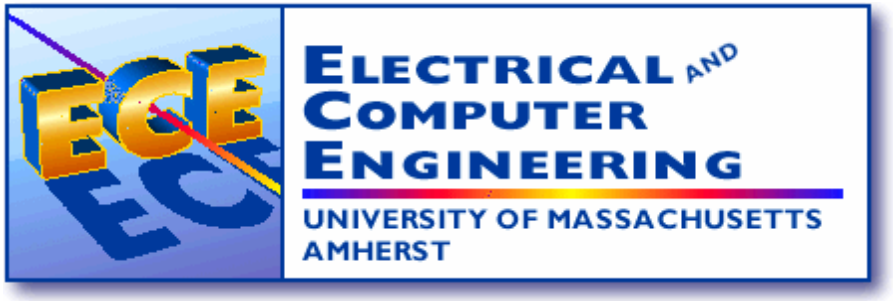
ENGIN 112

Intro to Electrical and Computer Engineering

Lecture 27

Counters



Overview

- Counters are important components in computers
  - The increment or decrement by one in response to input
- Two main types of counters
  - Ripple (asynchronous) counters
  - Synchronous counters
- Ripple counters
  - Flip flop output serves as a source for triggering other flip flops
- Synchronous counters
  - All flip flops triggered by a clock signal
- Synchronous counters are more widely used in industry.

Counters

- Counter: A register that goes through a prescribed series of states
- Binary counter
  - Counter that follows a binary sequence
  - N bit binary counter counts in binary from n to 2^{n-1}
- Ripple counters triggered by initial Count signal
- Applications:
  - Watches
  - Clocks
  - Alarms
  - Web browser refresh

Binary Ripple Counter

- **Reset** signal sets all outputs to 0
- Count signal toggles output of **low-order** flip flop
- Low-order flip flop provides trigger for adjacent flip flop
- Not all flops change value simultaneously
  - Lower-order flops change first
- Focus on D flip flop implementation

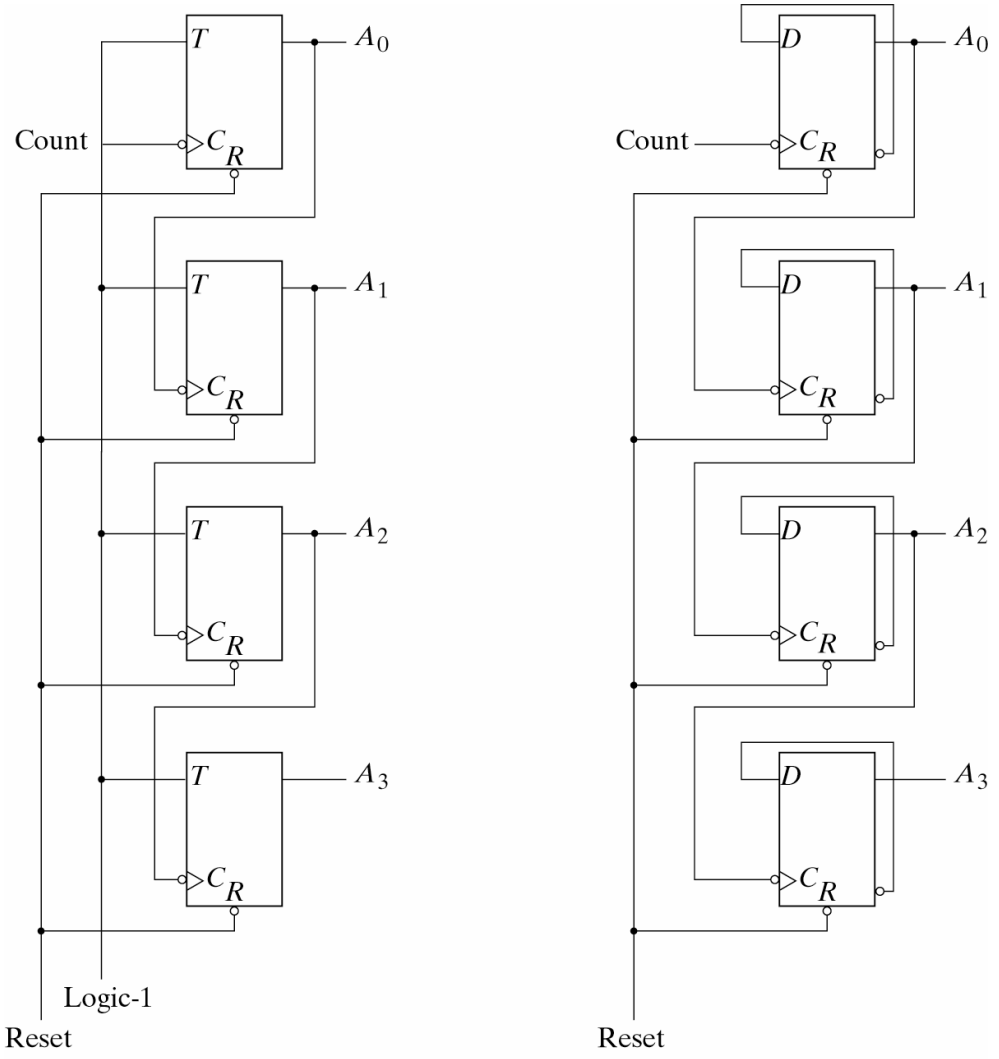
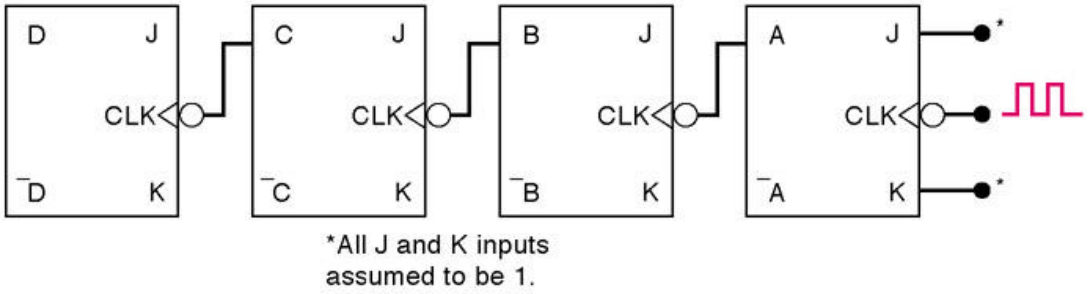
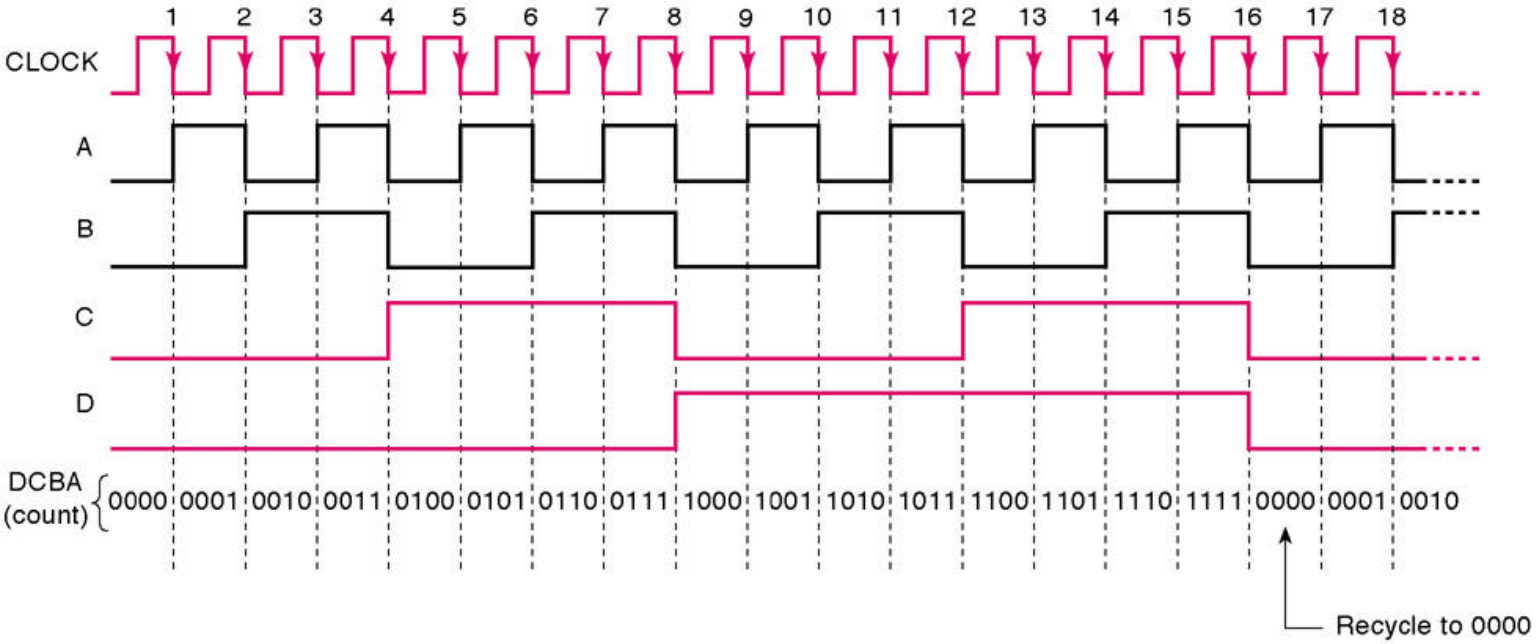


Fig. 6-8 4-Bit Binary Ripple Counter

Another Asynchronous Ripple Counter



*All J and K inputs assumed to be 1.



- **Similar to T flop example on previous slide**

Asynchronous Counters

- **Each FF output drives the CLK input of the next FF.**
- **FFs do not change states in exact synchronism with the applied clock pulses.**
- ***There is delay between the responses of successive FFs.***
- ***Ripple counter* due to the way the FFs respond one after another in a kind of rippling effect.**

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
1	0	0	0
1	0	0	1

Synchronous counters

- **Synchronous(parallel) counters**
  - **All of the FFs are triggered simultaneously by the clock input pulses.**
  - **All FFs change at same time**
- **Remember**
  - **If J=K=0, flop maintains value**
  - **If J=K=1, flop toggles**
- **Most counters are synchronous in computer systems.**
- **Can also be made from D flops**
- **Value increments on positive edge**

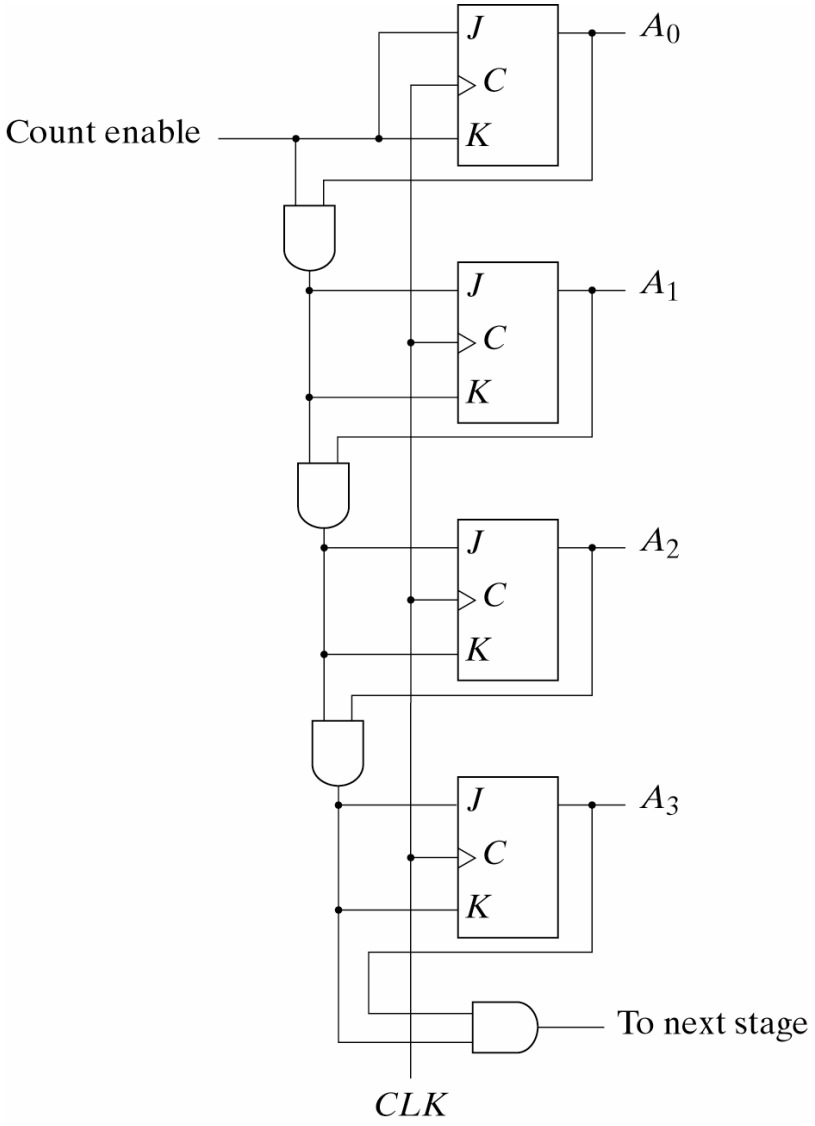
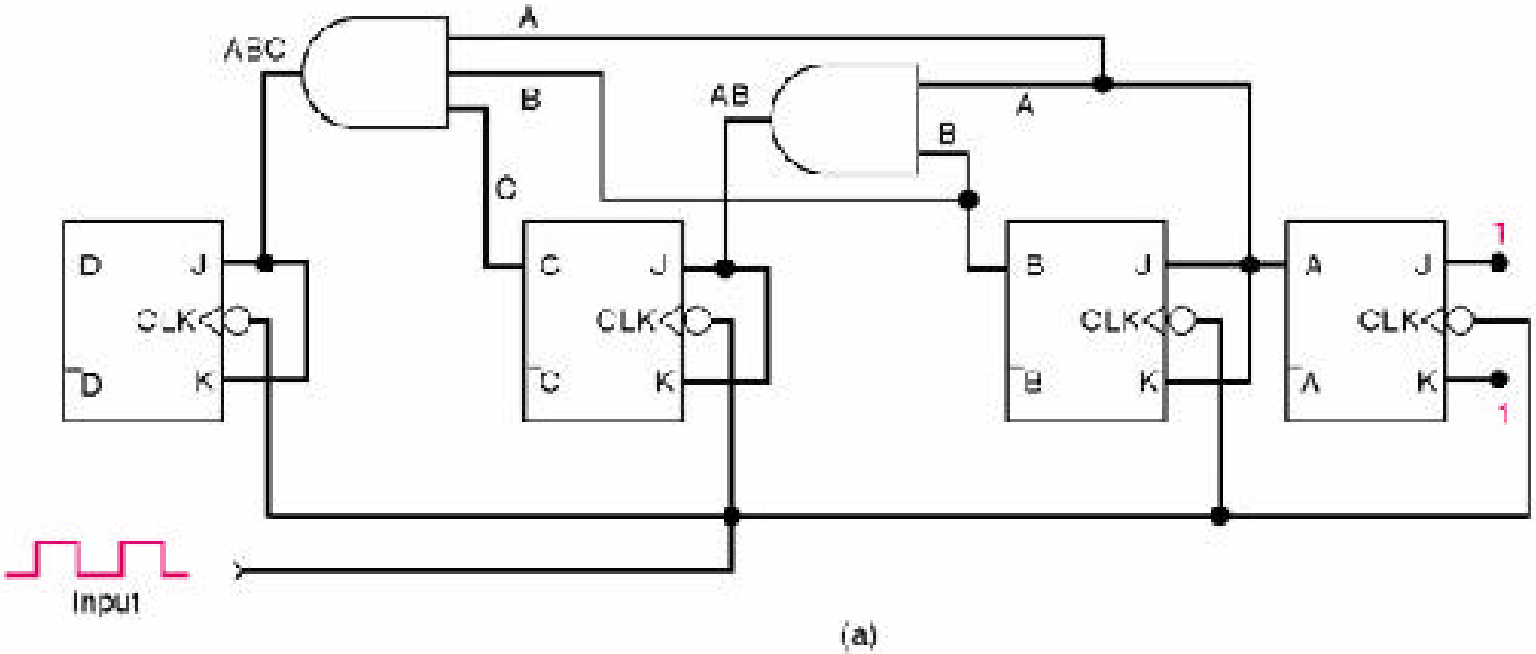


Fig. 6-12 4-Bit Synchronous Binary Counter

Synchronous counters

- **Synchronous counters**
  - **Same counter as previous slide except Count enable replaced by J=K=1**
  - **Note that clock signal is a square wave**
  - **Clock fans out to all clock inputs**



Circuit operation

Count	D	C	B	A
0	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	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0
.	.	.	.	.
.	.	.	.	.
.	.	etc.	.	.

(b)

- Count value increments on each negative edge
- Note that low-order bit (A) toggles on each clock cycle

Synchronous UP/Down counters

- Up/Down Counter can either count up or down on each clock cycle
- Up counter counts from 0000 to 1111 and then changes back to 0000
- Down counter counts from 1111 to 0000 and then back to 1111
- Counter counts up or down each clock cycle
- Output changes occur on clock rising edge

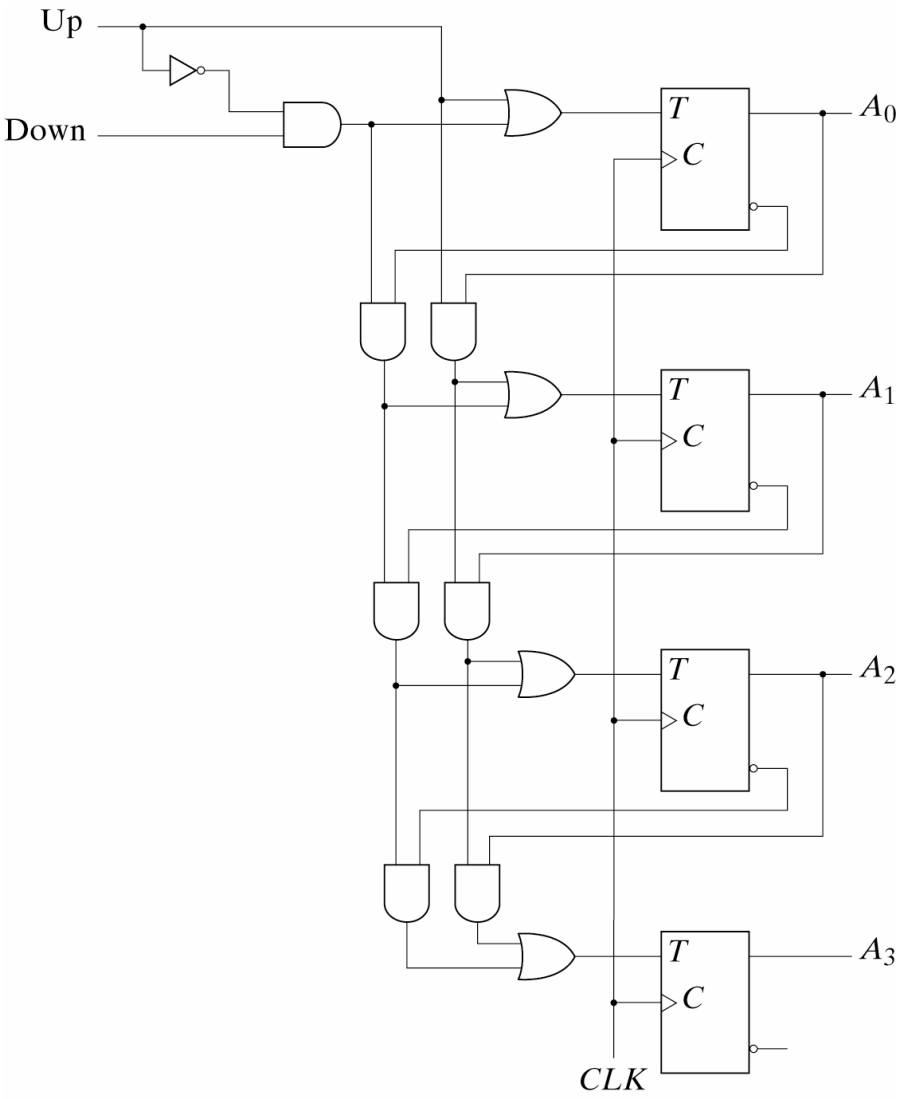


Fig. 6-13 4-Bit Up-Down Binary Counter

Counters with Parallel Load

- Counters with parallel load can have a preset value
- **Load** signal indicates that data ($I_3 \ldots I_0$) should be loaded into the counter
- **Clear** resets counter to all zeros
- **Carry output** could be used for higher-order bits

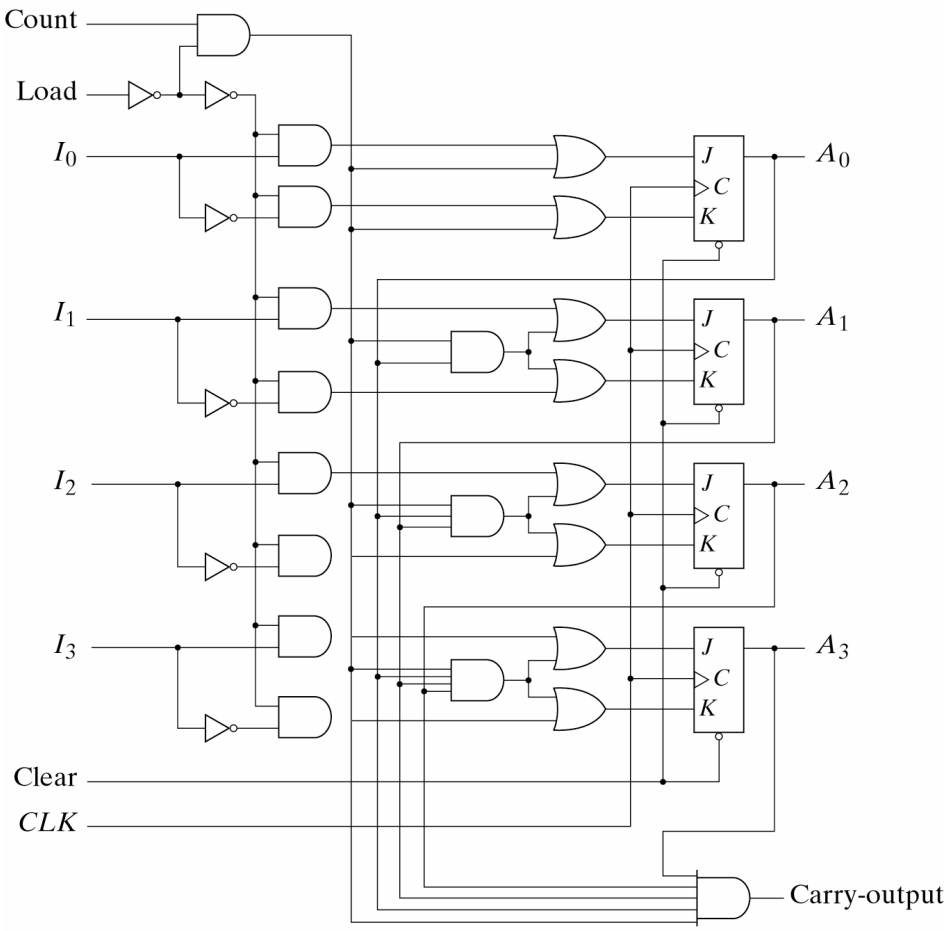


Fig. 6-14 4-Bit Binary Counter with Parallel Load

Counters with Parallel Load

Clear	Clk	Load	Count	Function
0	X	X	X	Clear to 0
1	?	1	X	Load inputs
1	?	0	1	Count
1	?	0	0	No Change

Function Table

- **If Clear is asserted (0), the counter is cleared**
- **If Load is asserted data inputs are loaded**
- **If Count asserted counter value is incremented**

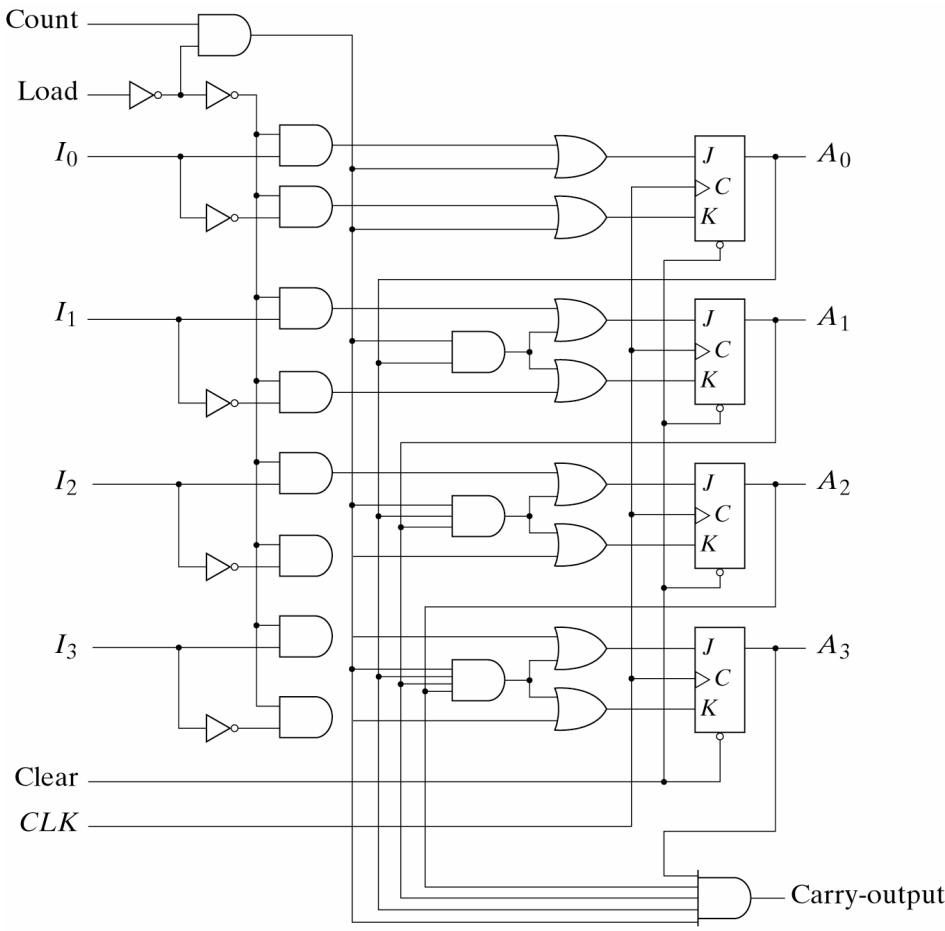
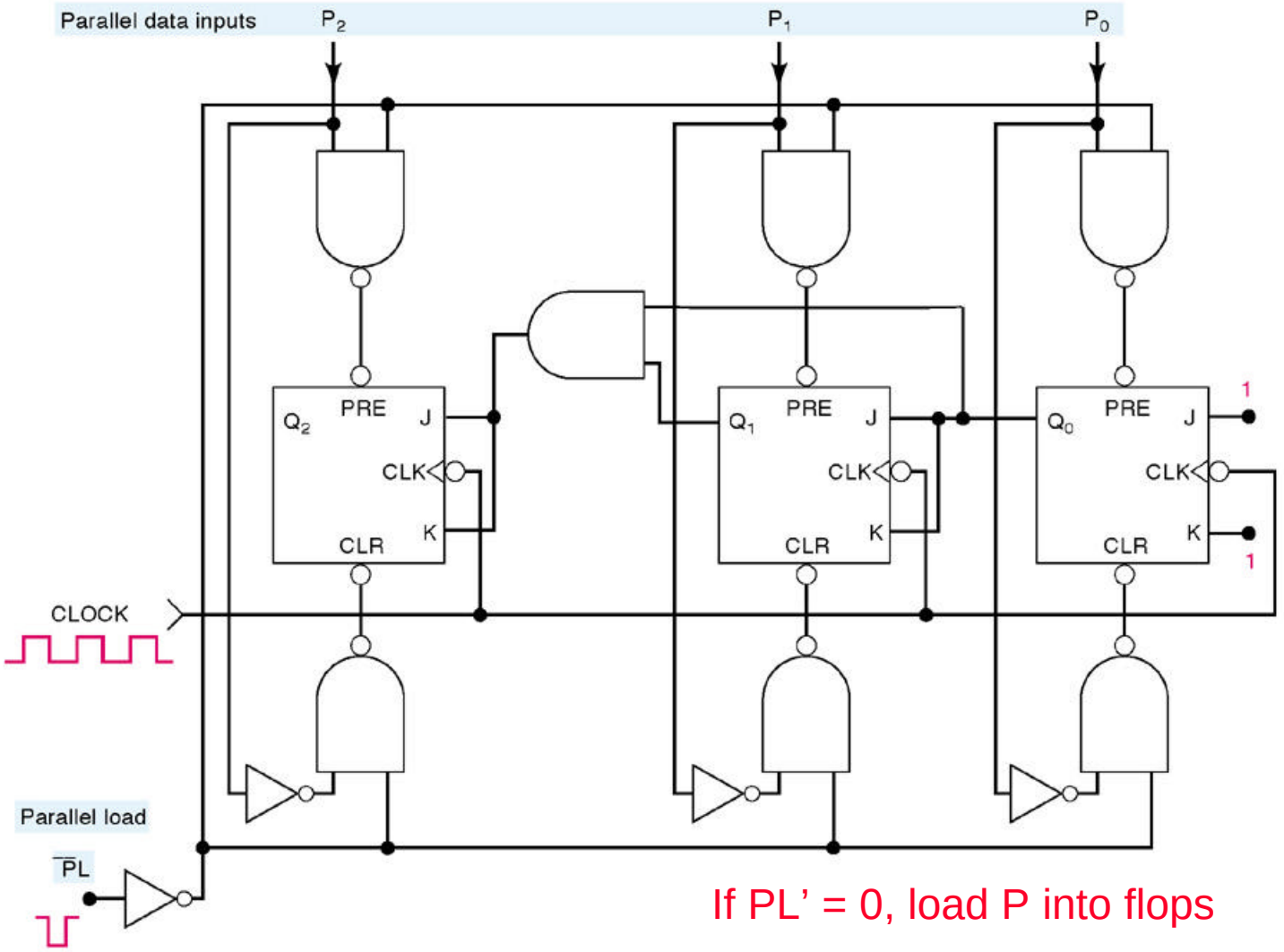


Fig. 6-14 4-Bit Binary Counter with Parallel Load

Binary Counter with Parallel Load and Preset

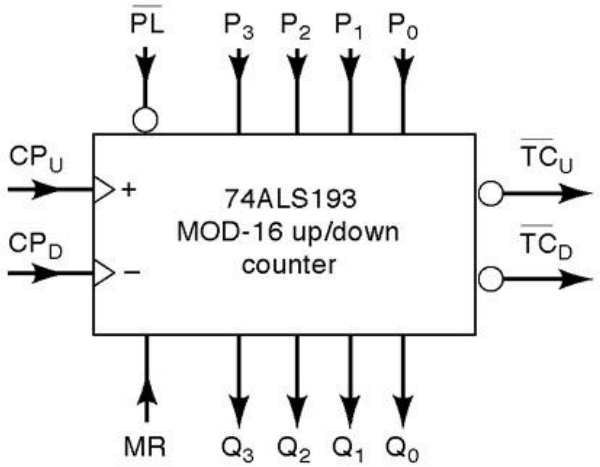
- **Presettable parallel counter with asynchronous preset.**



If PL' = 0, load P into flops

Binary Counter with Parallel Load and Preset

- **Commercial version of binary counter**



(a)

Mode Select

MR	$\overline{PL}$	CP_U	CP_D	Mode
H	X	X	X	Asynch. reset
L	L	X	X	Asynch. preset
L	H	H	H	No change
L	H	↑	H	Count up
L	H	H	↑	Count down

H = HIGH; L = LOW
X = Don't care; ↑ = PGT

(c)

Pin	Description
CP_U	Count-up clock input (active rising edge)
CP_D	Count-down clock input (active rising edge)
MR	Asynchronous master reset input (active HIGH)
$\overline{PL}$	Asynchronous parallel load input (active LOW)
P_0-P_3	Parallel data inputs
Q_0-Q_3	Flip-flop outputs
$\overline{TC}_D$	Terminal count-down (borrow) output (active LOW)
$\overline{TC}_U$	Terminal count-up (carry) output (active) LOW

(b)

Summary

- Binary counters can be ripple or synchronous
- Ripple counters use flip flop outputs as flop triggers
  - Some delay before all flops settle on a final value
  - Do no require a clock signal
- Synchronous counters are controlled by a clock
  - All flip flops change at the same time
- Up/Down counters can either increment or decrement a stored binary value
  - Control signal determines if counter counts up or down
- Counters with parallel load can be set to a known value before counting begins.