XMUT 202
Digital Electronics

A/Prof. Pawel Dmochowski

School of Engineering and Computer Science
Victoria University of Wellington

Review 2 – Sequential Logic

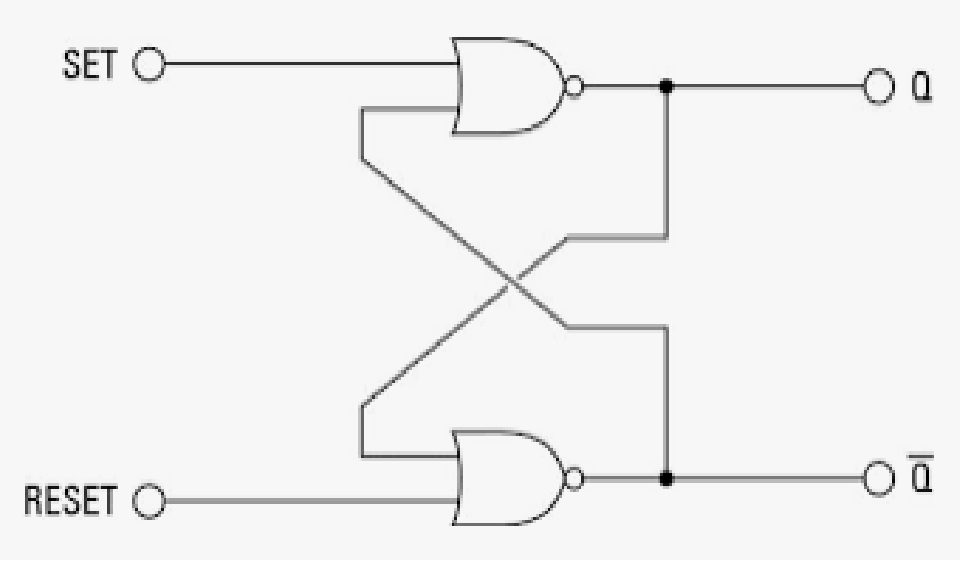
- Sequential Logic
  - Flip flops and Latches
  - Asynchronous and Synchronous Counters
    - Mod Number, Frequency Division
  - 555 Timer
  - General Counter Design
    - JK and D FF excitation Tables

Latches, Flip-Flops…

Latch is:

Asynchronous

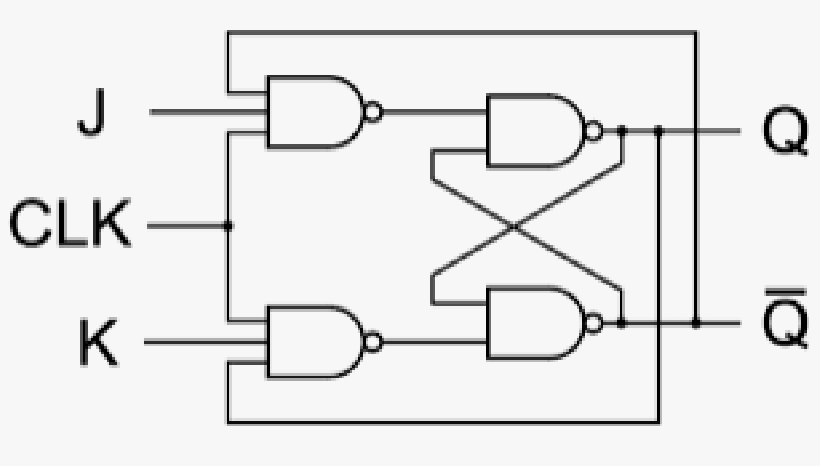
Output is triggered by input signals



Flip-flop:

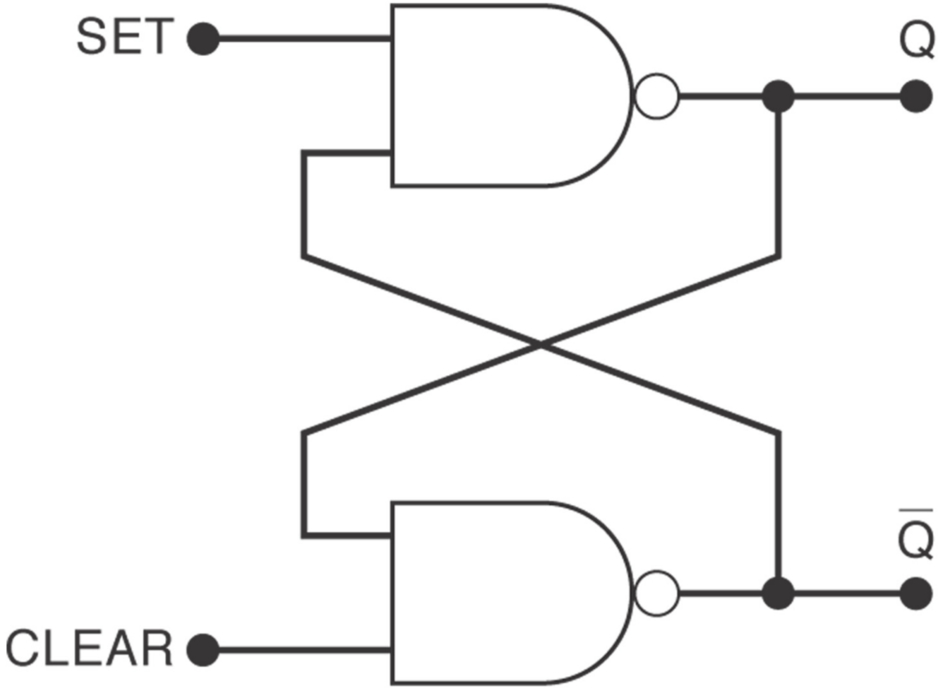
Synchronous

Edge triggered by clock signal



The NAND gate latch and truth table.

Active LOW



(a)

Set	Clear	Output
1	1	No change
0	1	Q = 1
1	0	Q = 0
0	0	Invalid*

*Produces $Q = \overline{Q} = 1$.

(b)

Exercise: Latches

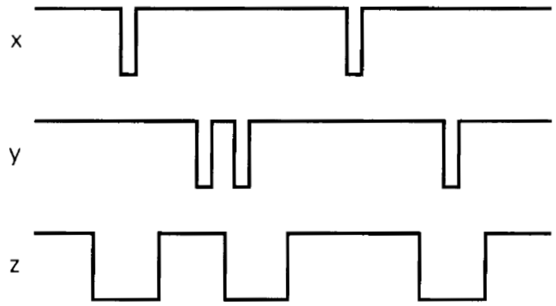


FIGURE 5-61

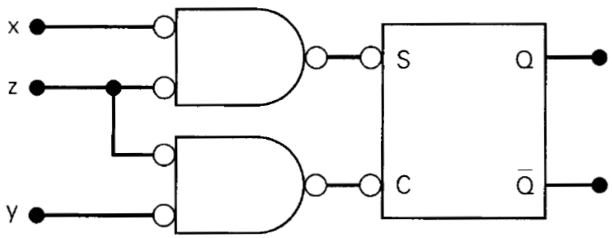
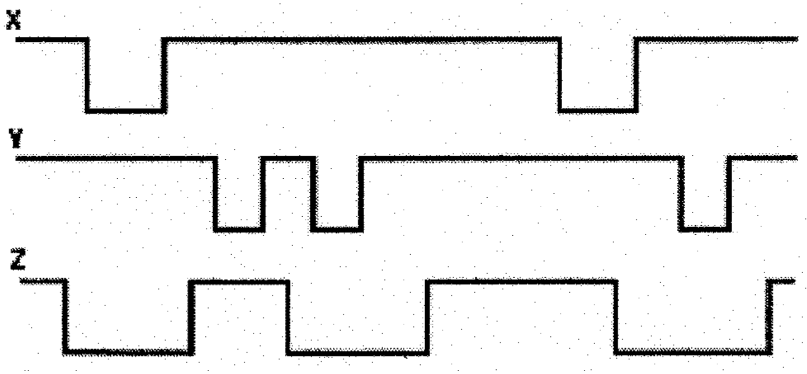


FIGURE 5-62

5-3. The waveforms of Figure 5-61 are connected to the circuit of Figure 5-62. Assume that $Q = 0$ initially, and determine the Q waveform.



Exercise: Latches

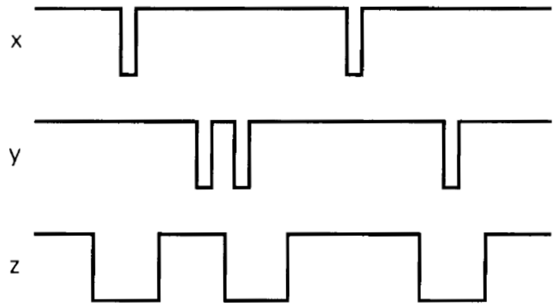


FIGURE 5-61

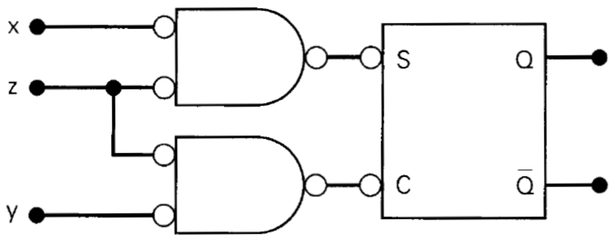
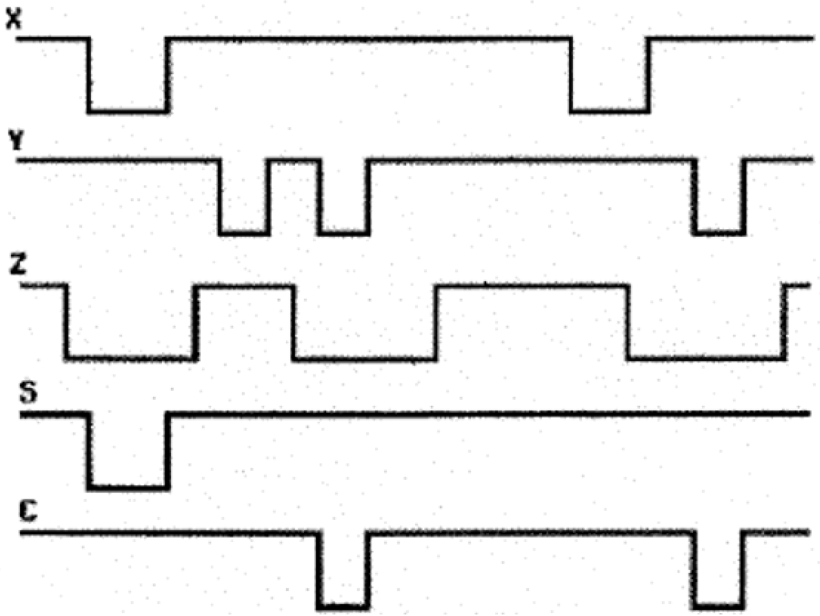


FIGURE 5-62

5-3. The waveforms of Figure 5-61 are connected to the circuit of Figure 5-62. Assume that $Q = 0$ initially, and determine the Q waveform.

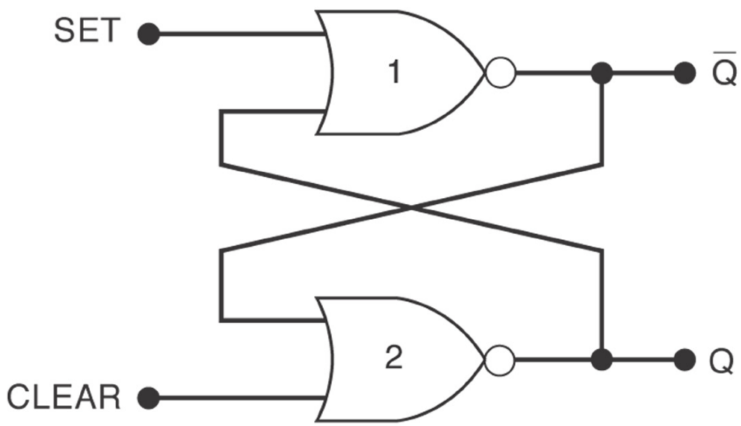


The NOR gate latch

NOR Gate

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Active High

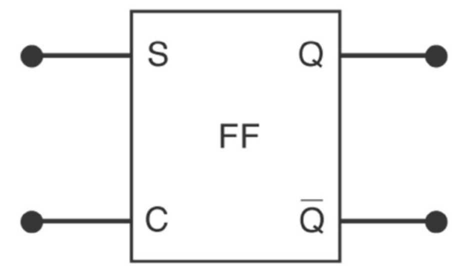


(a)

Set	Clear	Output
0	0	No change
1	0	Q = 1
0	1	Q = 0
1	1	Invalid*

*Produces Q = $\overline{Q}$ = 0.

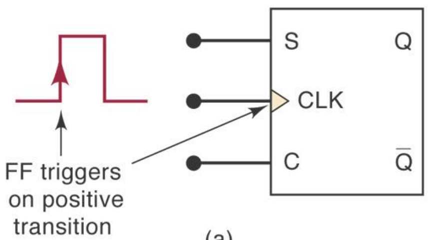
(b)



(c)

- SET = CLEAR = 0: Normal resting state – no effect on O/P state.
- SET = 1, CLEAR = 0: gets Q = 1, will remain when SET goes LO.
- SET = 0, CLEAR = 1: Clear to Q = 0, remain when CLEAR goes LO.
- SET = 1 = CLEAR, Tries to set and clear simultaneously, not allowed.

Clocked SC Flip Flop

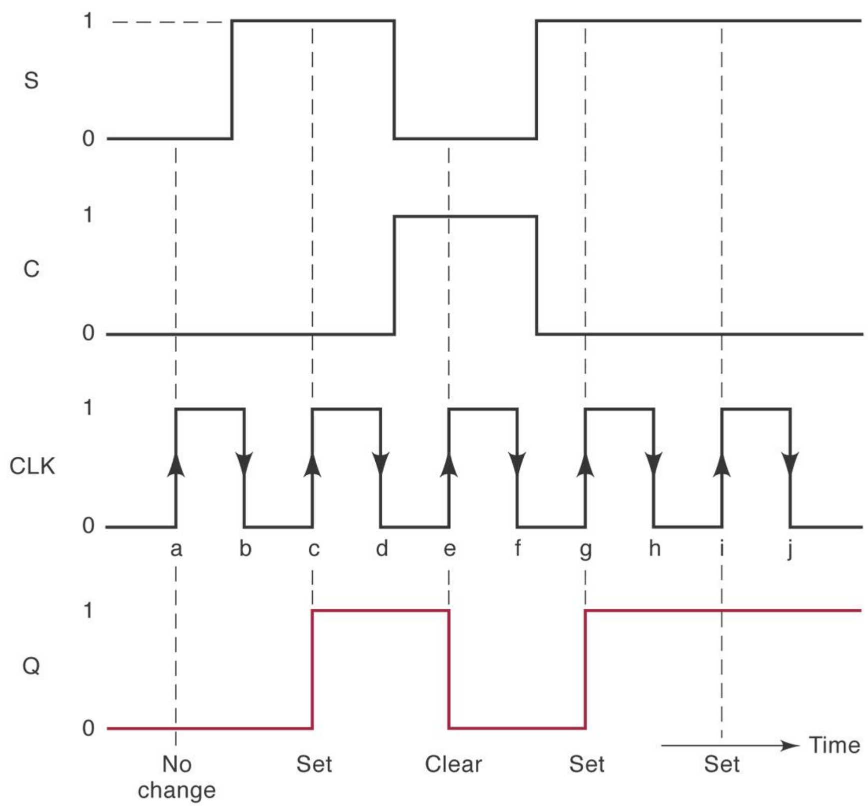


(a)

Inputs			Output
S	C	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

Q_0 is output level prior to ↑ of CLK.
↓ of CLK produces no change in Q.

(b)



(c)

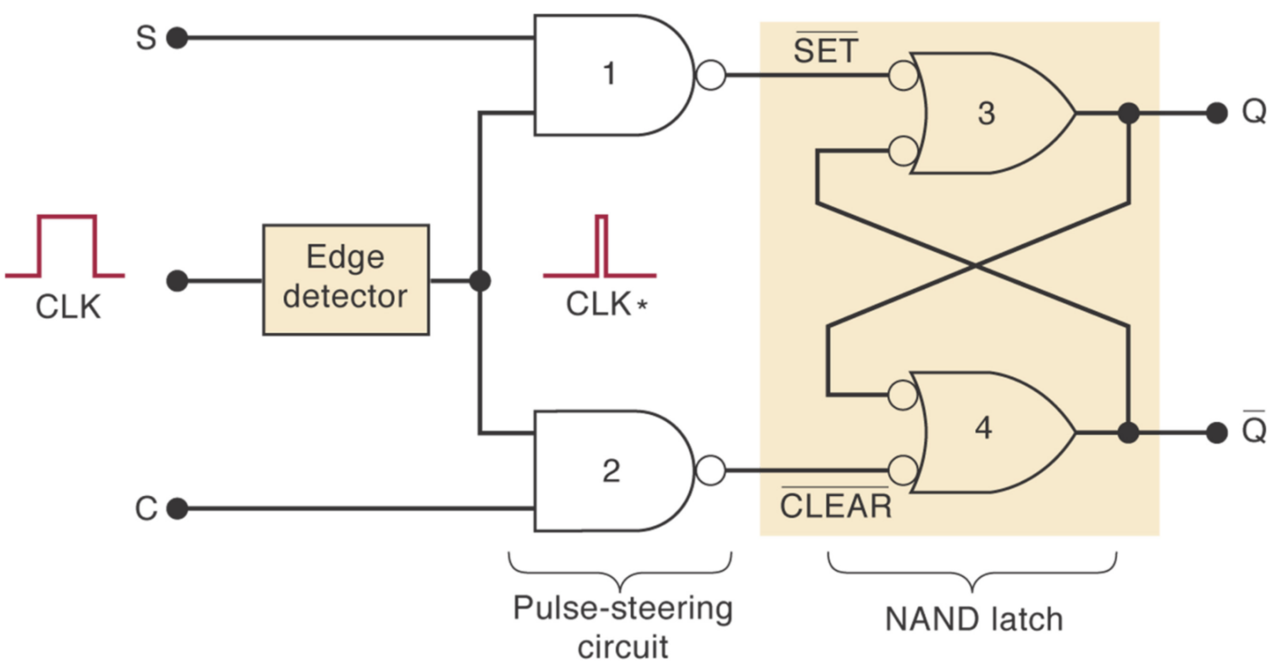
Internal circuitry of the edge triggered S-C FF

Consists of three different sections:

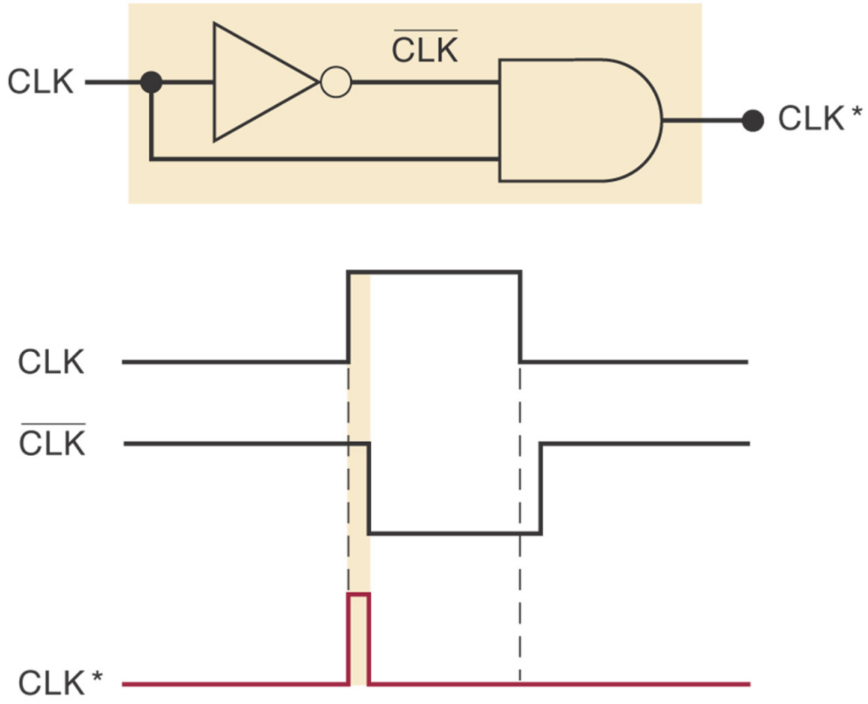
1. A basic NAND gate latch (NAND 3 and NAND 4)

2. Pulse steering circuit (NAND 1 and NAND 2)

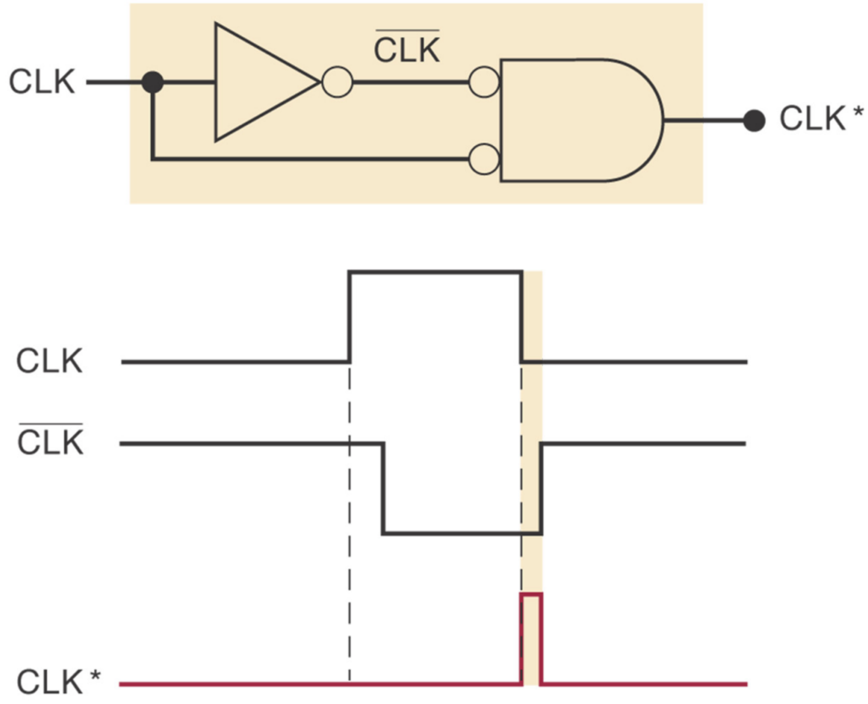
3. Edge detector circuit



Edge detector circuits.



(a)



(b)

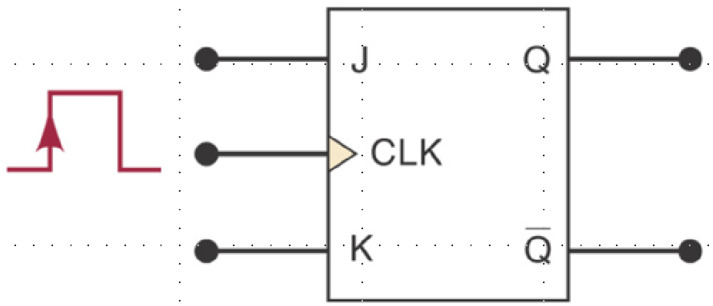
PGT

Positive Going Transition

NGT

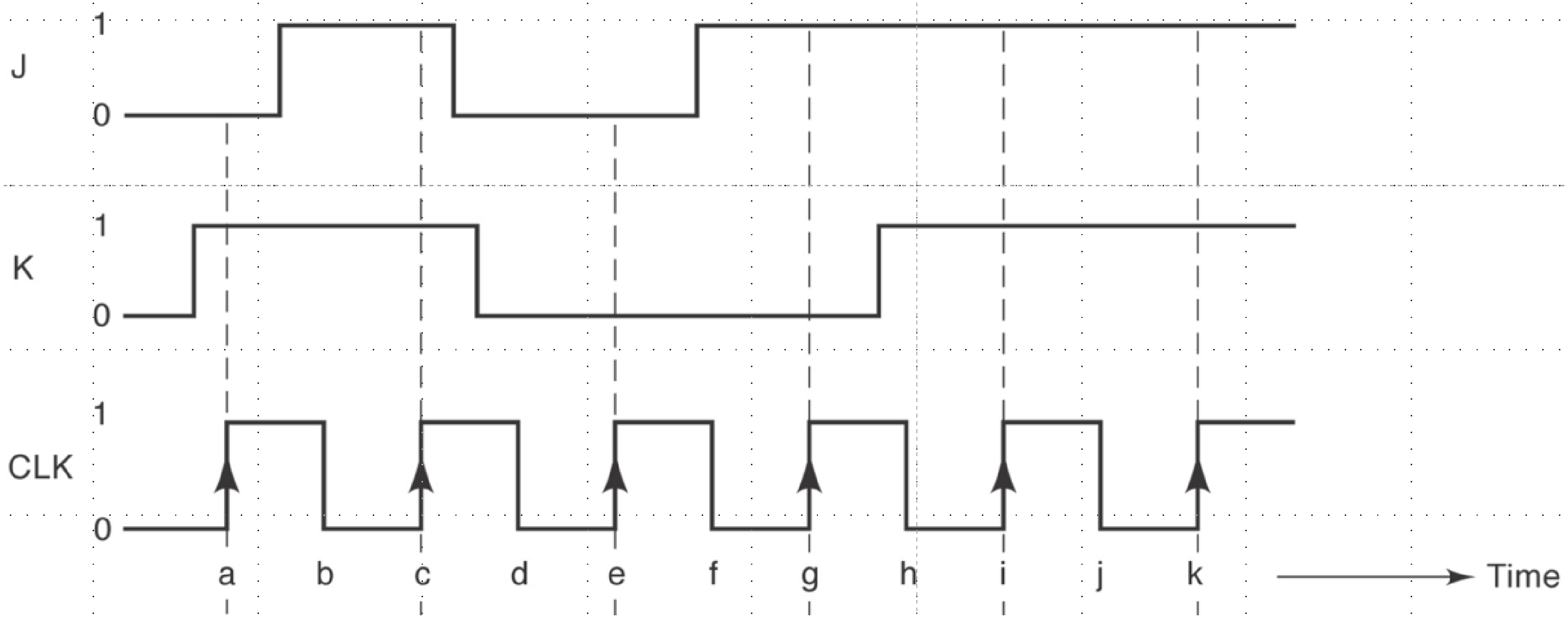
Negative Going Transition

Clocked J-K Flip Flop (triggers on PGT)

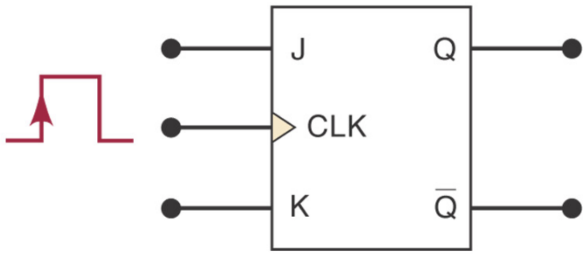


J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q_0}$ (toggles)

(a)

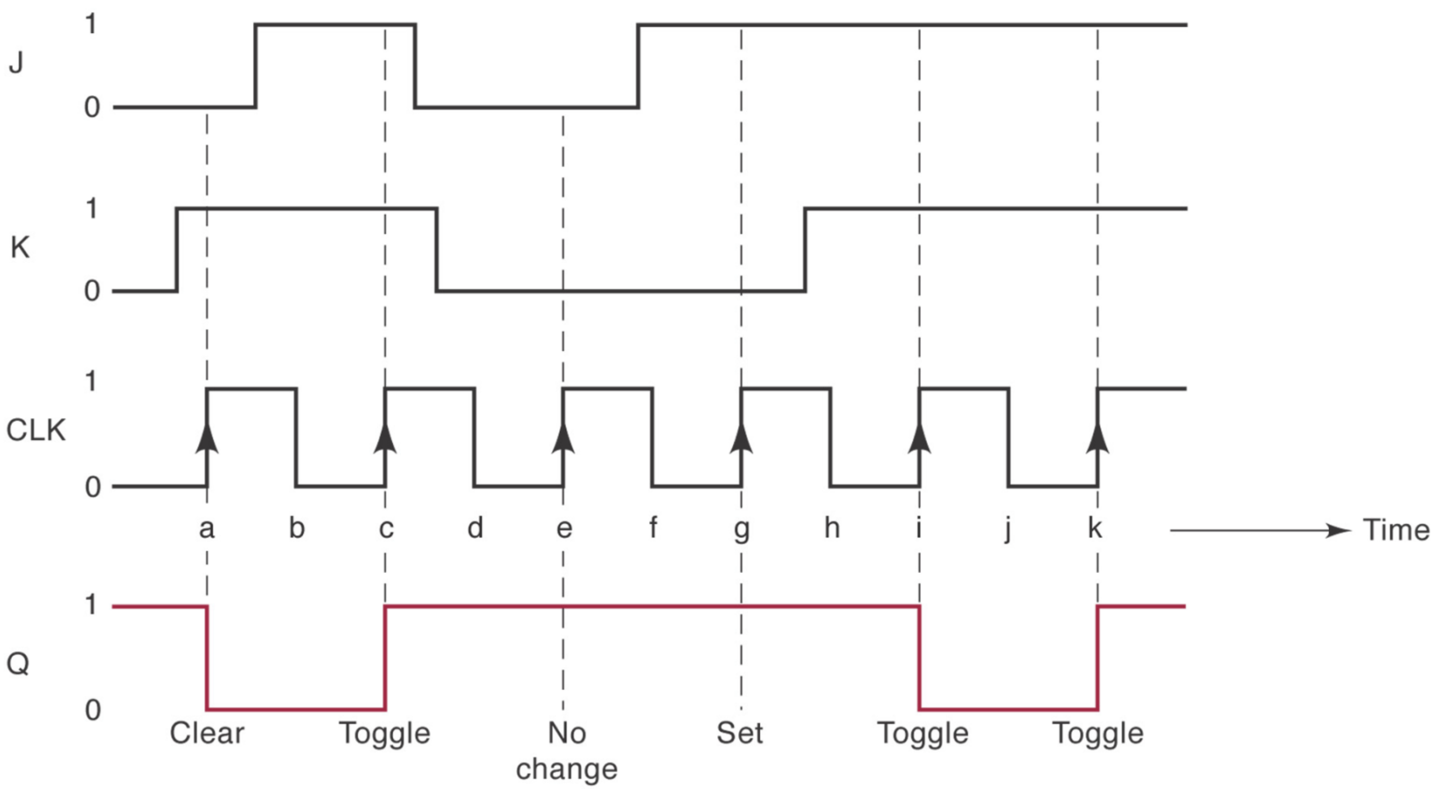


Clocked J-K Flip Flop (triggers on PGT)



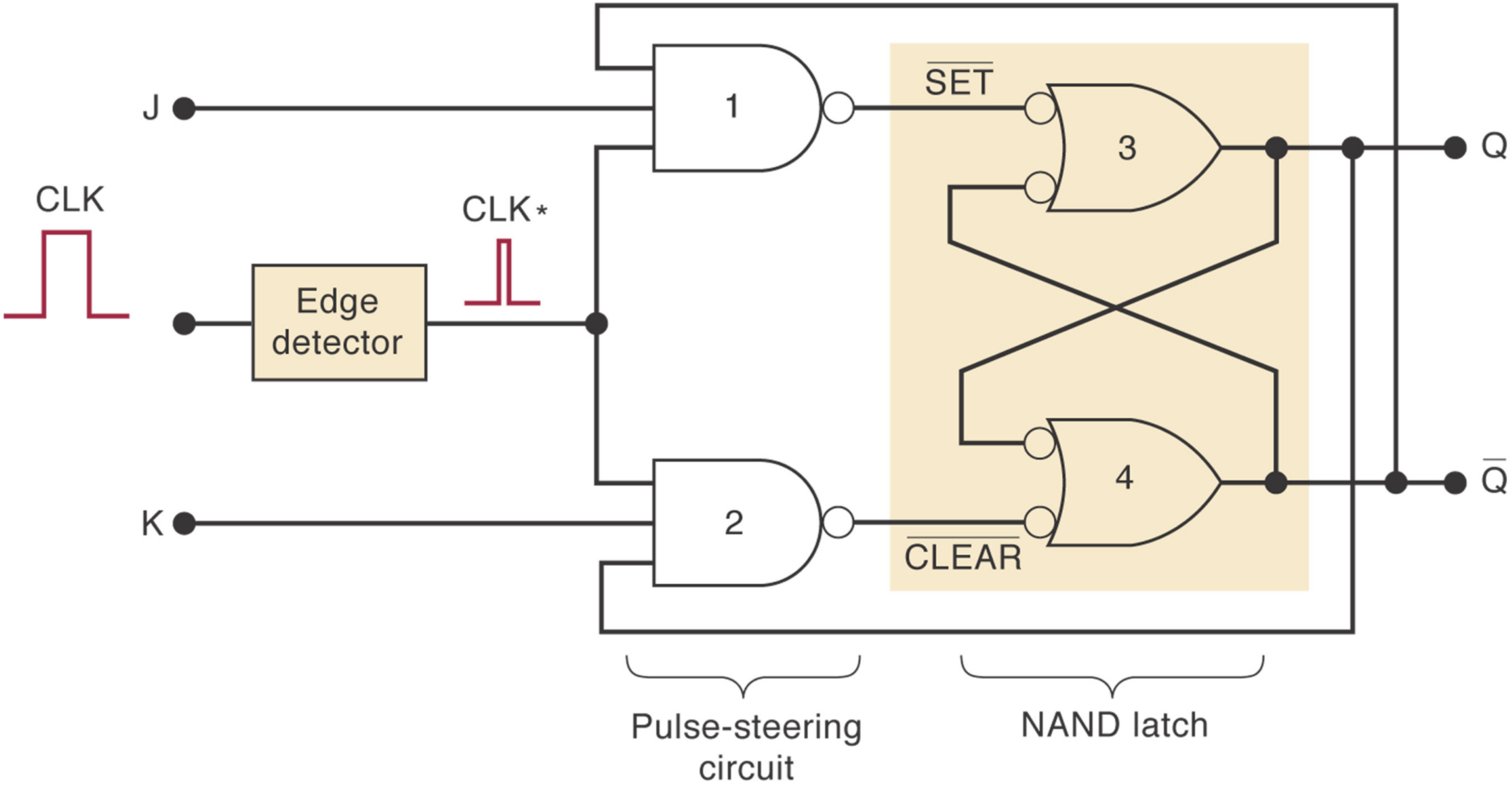
J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q_0}$ (toggles)

(a)



(b)

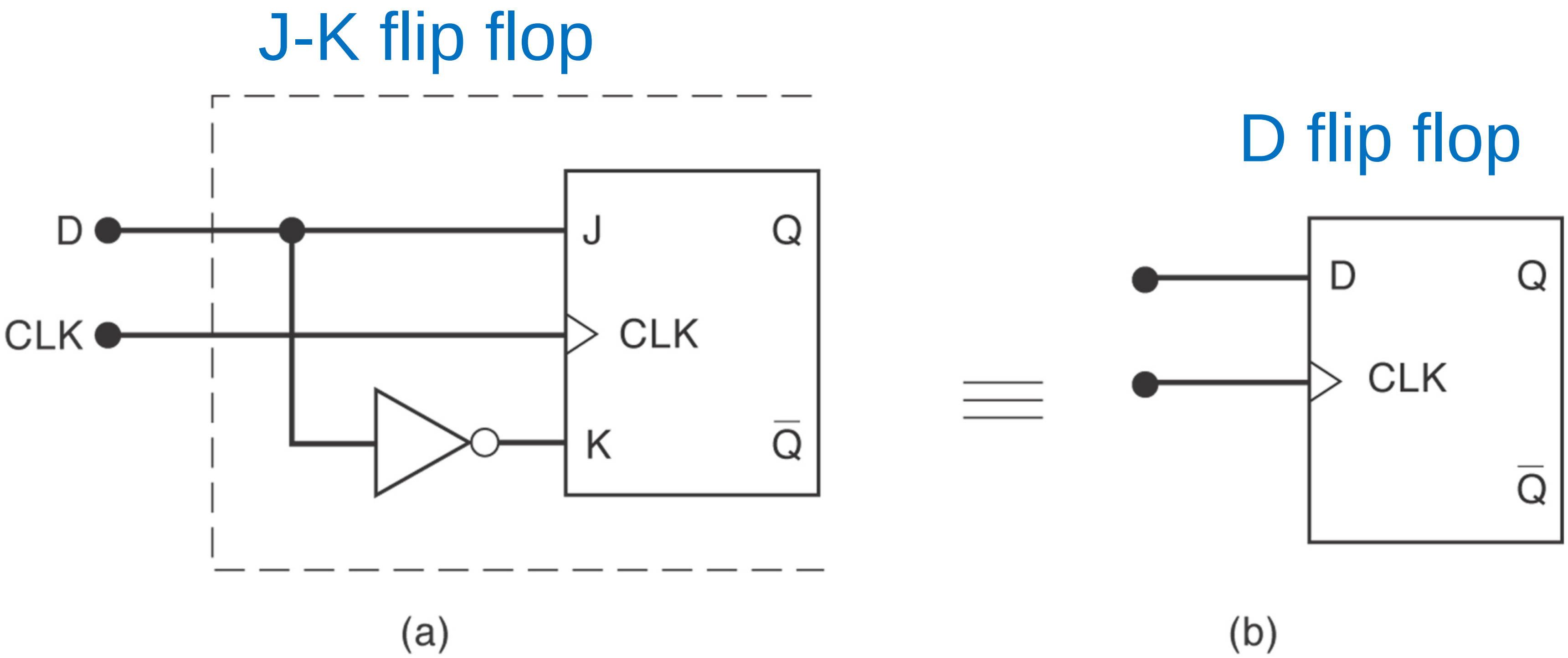
Internal circuitry of edge triggered J-K flip flop.



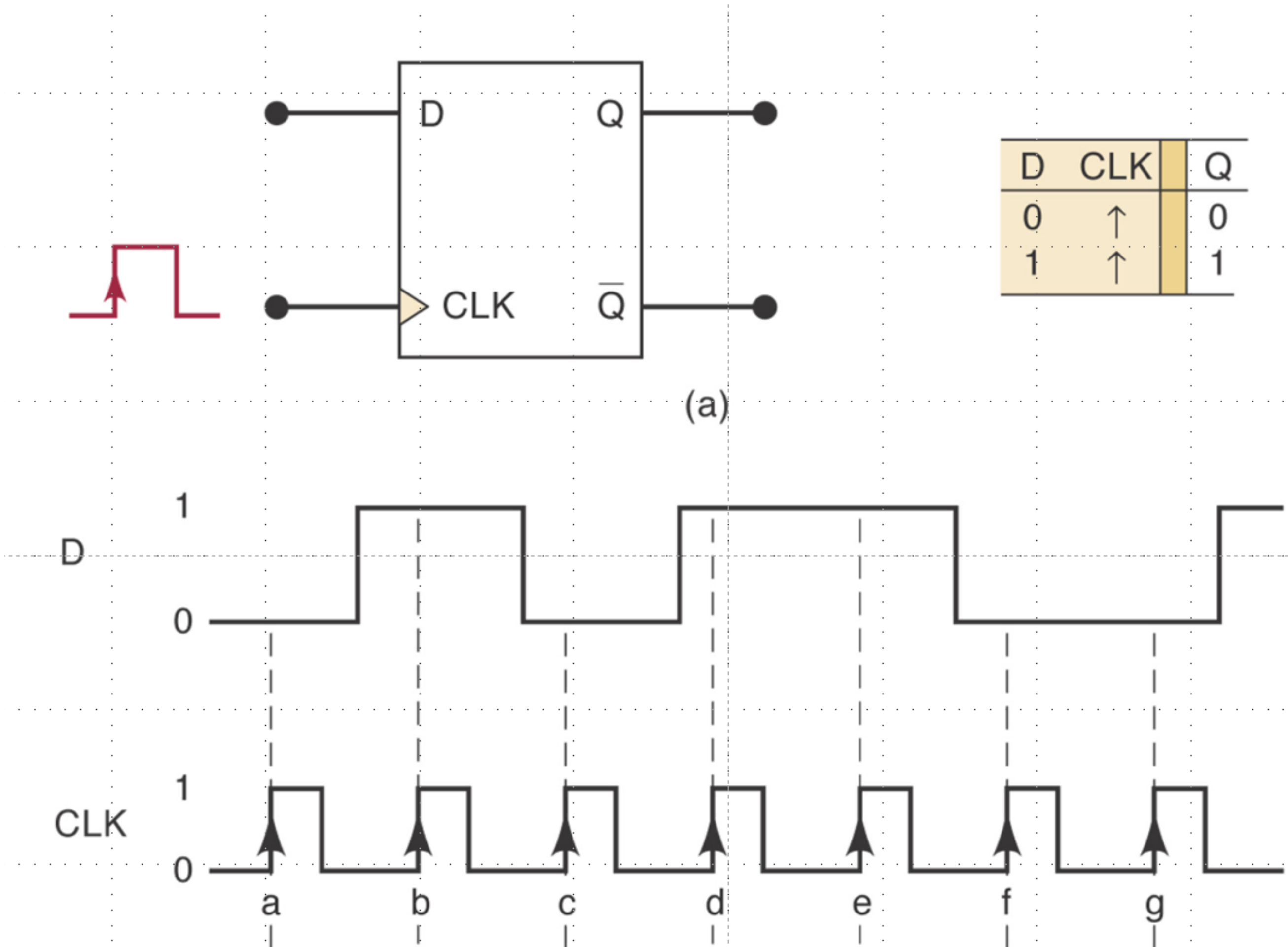
Clocked D Flip-Flop

- One data input.
- The output changes to the value of the input at either the positive going or negative going clock trigger (device dependent) .
- May be implemented with a J-K FF by tying the J input to the K input through an inverter.
- Useful for parallel data transfer.

D FF implementation from a J-K FF



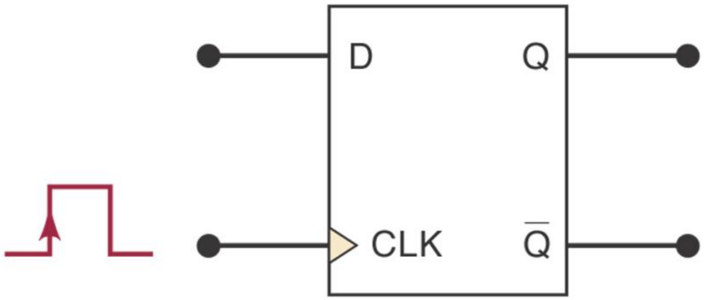
D flip flop triggering on PGT's



D	CLK	Q
0	↑	0
1	↑	1

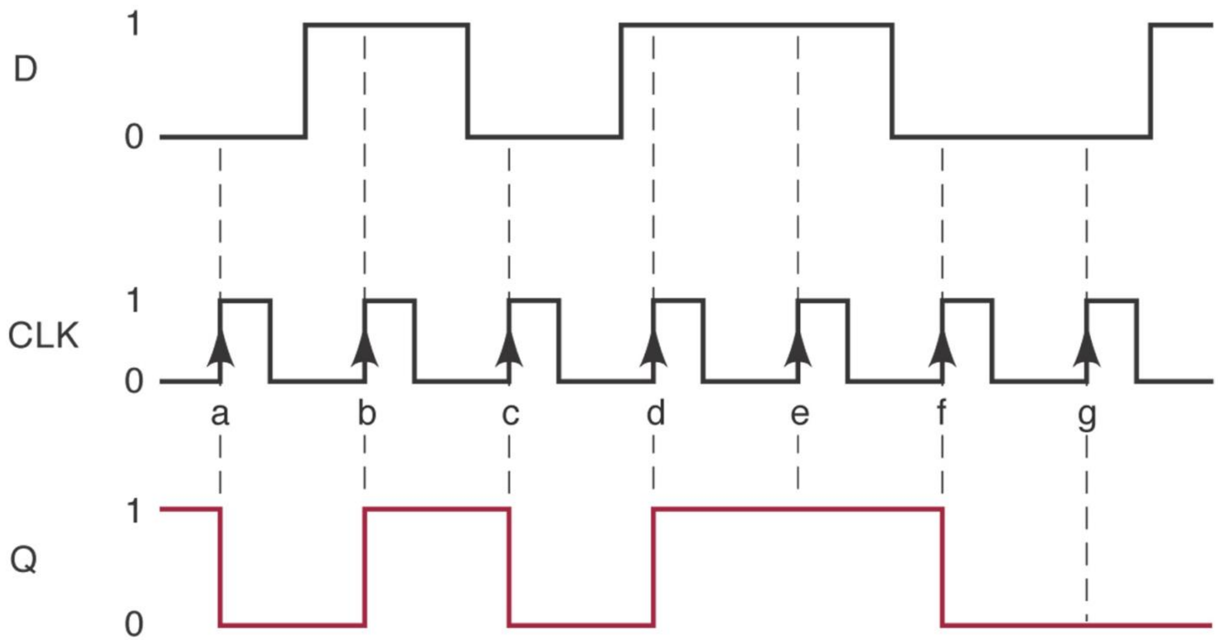
D flip flop triggering on PGT’s

Q is always the same as D
but updates on PGT only



D	CLK	Q
0	↑	0
1	↑	1

(a)

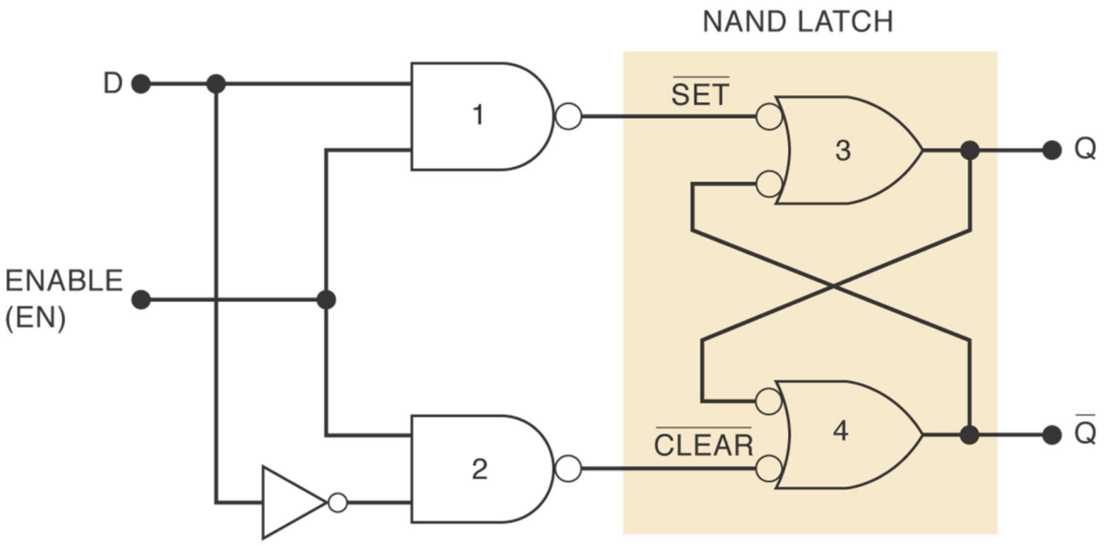


(b)

D Latch (Transparent Latch)

- One data input.
- The clock has been replaced by an enable (EN) line.
- The device is NOT edge triggered.
- The output follows the input only when EN is active.

Structure of the D latch

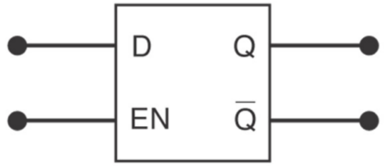


(a)

Inputs		Output
EN	D	Q
0	X	Q_0 (no change)
1	0	0
1	1	1

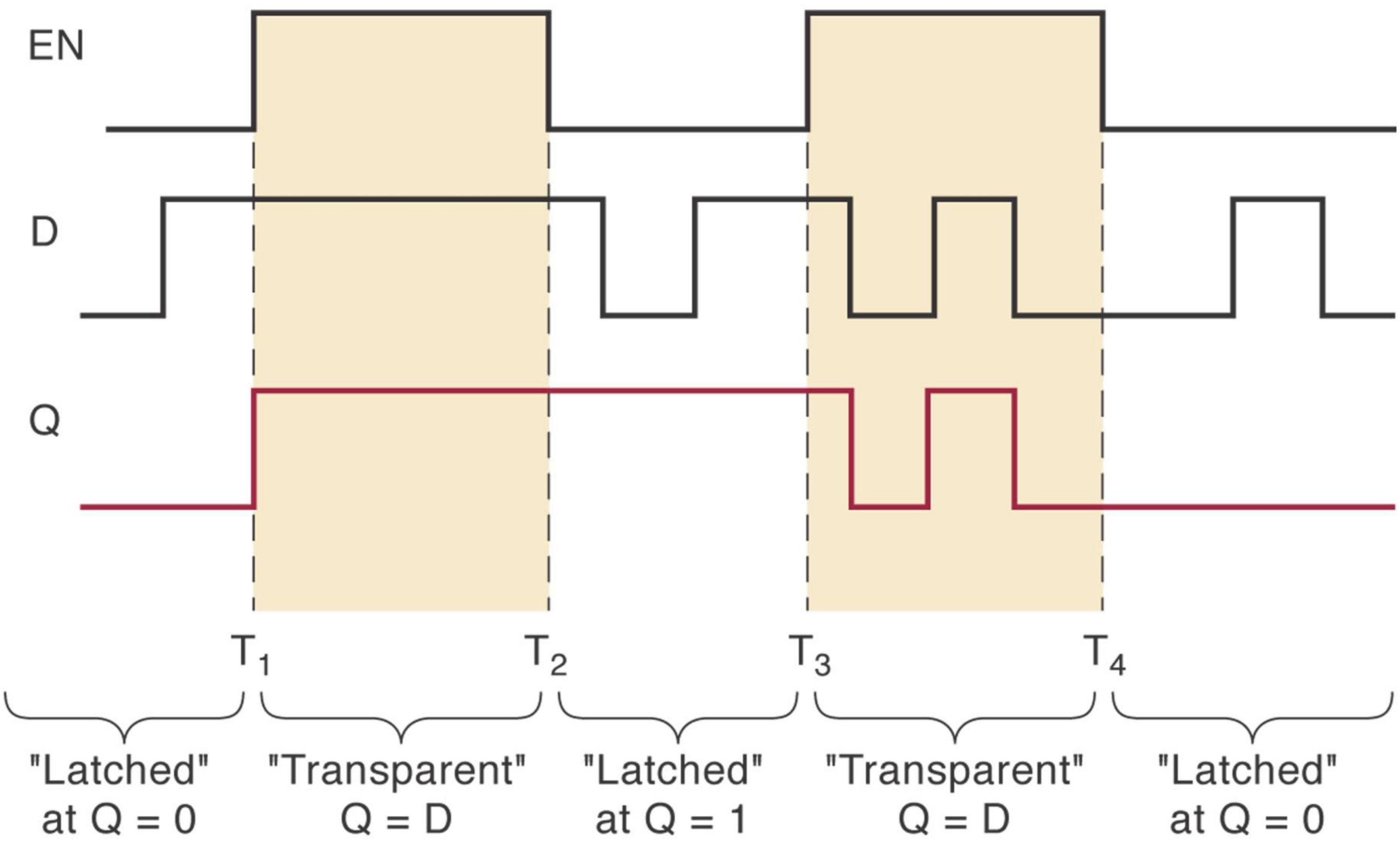
"X" indicates "don't care."
Q_0 is state Q just
prior to EN going LOW.

(b)

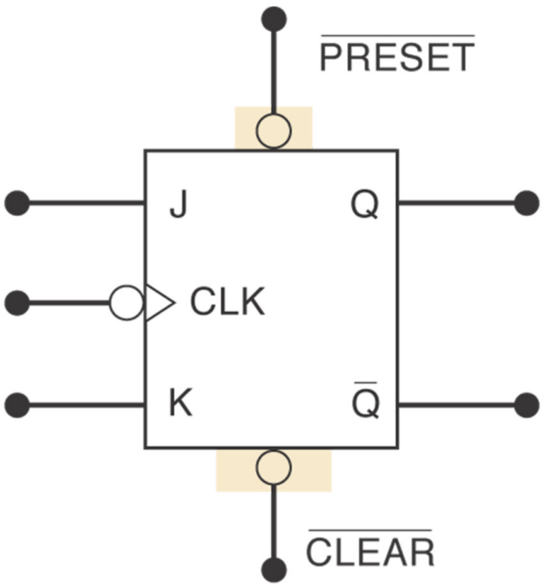


(c)

Operation of the D latch



Clocked J-K FF with asynchronous inputs.

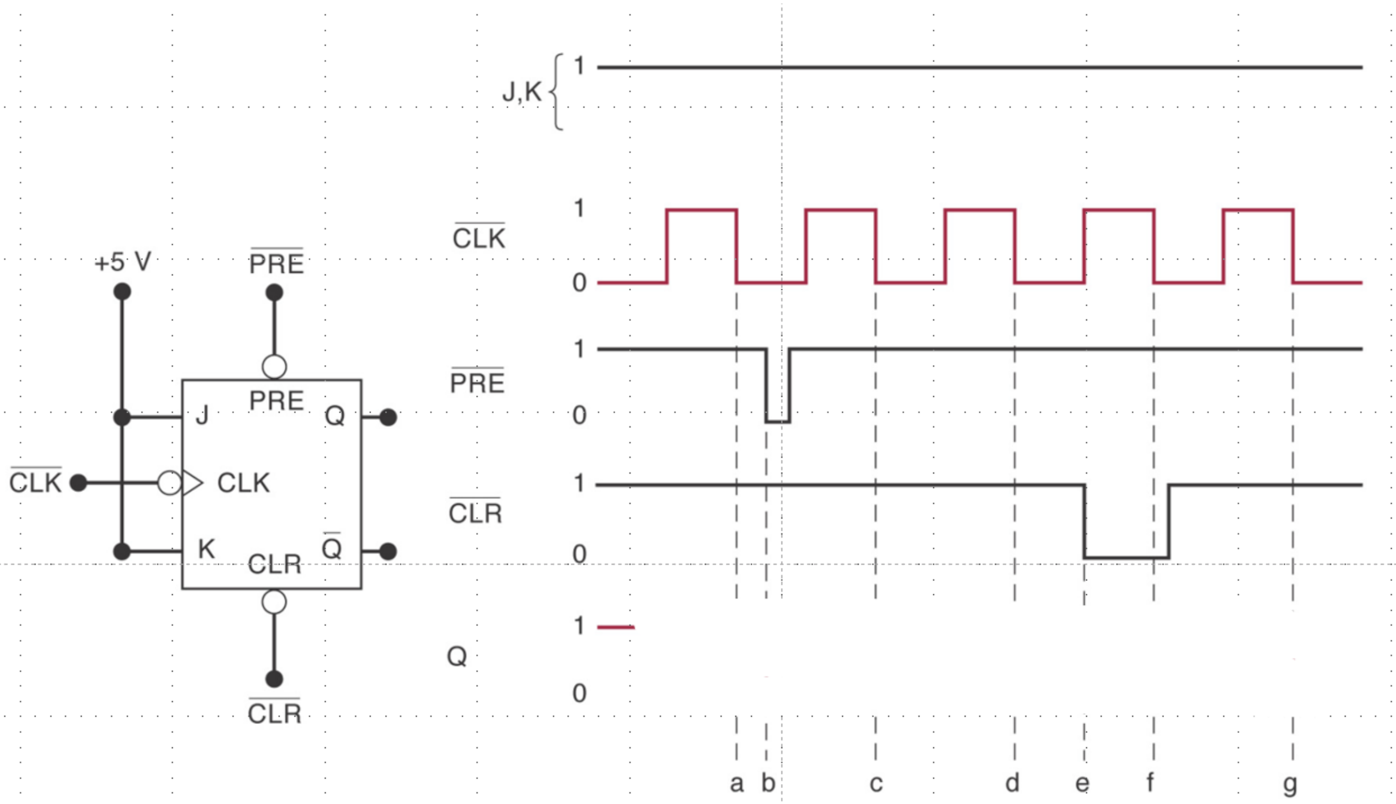


$\overline{\text{PRESET}}$	$\overline{\text{CLEAR}}$	FF response
1	1	Clocked operation*
0	1	Q = 1 (regardless of CLK)
1	0	Q = 0 (regardless of CLK)
0	0	Not used

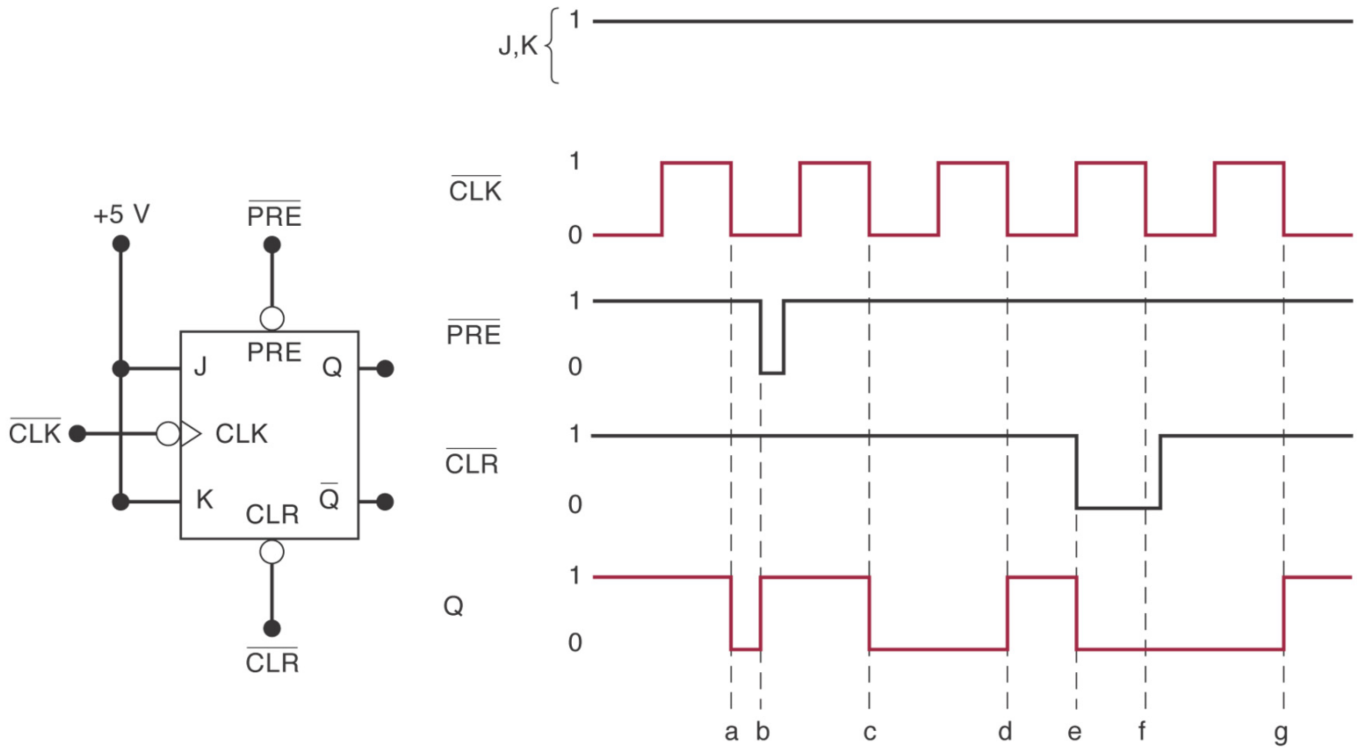
*Q will respond to J, K, and CLK.

- The labels PRE and CLR are used for asynchronous inputs.
- Active low asynchronous inputs will have a bar over the labels and inversion bubbles.

Clocked FF responding to asynchronous inputs



Clocked FF responding to asynchronous inputs



(a)

Point	Operation
a	Synchronous toggle on NGT of $\overline{CLK}$
b	Asynchronous set on $\overline{PRE}$ = 0
c	Synchronous toggle
d	Synchronous toggle
e	Asynchronous clear on $\overline{CLR}$ = 0
f	$\overline{CLR}$ overrides the NGT of $\overline{CLK}$
g	Synchronous toggle

(b)

FF Practical Timing Considerations

		TTL		CMOS	
		7474	74LS112	74C74	74HC112
t_S		20	20	60	25
t_H		5	0	0	0
t_{PHL}	from CLK to Q	40	24	200	31
t_{PLH}	from CLK to Q	25	16	200	31
t_{PHL}	from $\overline{CLR}$ to Q	40	24	225	41
t_{PLH}	from $\overline{PRE}$ to Q	25	16	225	41
$t_W(L)$	CLK LOW time	37	15	100	25
$t_W(H)$	CLK HIGH time	30	20	100	25
$t_W(L)$	at $\overline{PRE}$ or $\overline{CLR}$	30	15	60	25
f_{MAX}	in MHz	15	30	5	20

(a) Assume that $Q = 0$. How long can it take for Q to go HIGH when a PGT occurs at the *CLK* input of a 7474?

(b) Assume that $Q = 1$. How long can it take for Q to go LOW in response to the $\overline{CLR}$ input of a 74HC112?

(c) What is the narrowest pulse that should be applied to the $\overline{CLR}$ input of the 74LS112 FF to clear Q reliably?

(d) Which FF in Table 5-2 requires that the control inputs remain stable *after* the occurrence of the active clock transition?

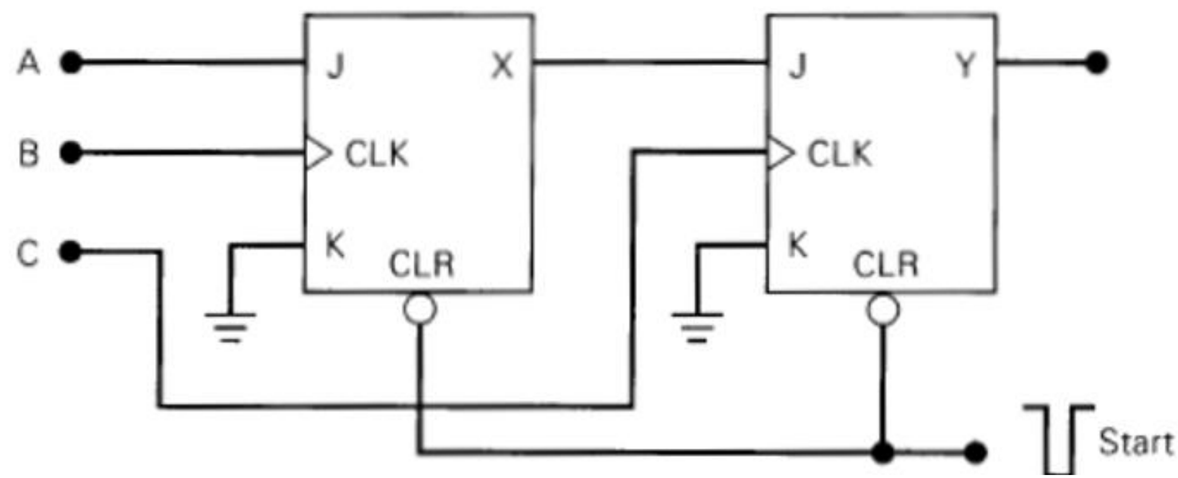
(e) For which FFs must the control inputs be held stable for a minimum time prior to the active clock transition?

		TTL		CMOS	
		7474	74LS112	74C74	74HC112
t_S		20	20	60	25
t_H		5	0	0	0
t_{PHL}	from *CLK* to Q	40	24	200	31
t_{PLH}	from *CLK* to Q	25	16	200	31
t_{PHL}	from $\overline{CLR}$ to Q	40	24	225	41
t_{PLH}	from $\overline{PRE}$ to Q	25	16	225	41
$t_W(L)$	*CLK* LOW time	37	15	100	25
$t_W(H)$	*CLK* HIGH time	30	20	100	25
$t_W(L)$	at $\overline{PRE}$ or $\overline{CLR}$	30	15	60	25
f_{MAX}	in MHz	15	30	5	20

In the circuit of Figure 5-71, inputs *A*, *B*, and *C* are all initially LOW. Output *Y* is supposed to go HIGH only when *A*, *B*, and *C* go HIGH in a certain sequence.

(a) Determine the sequence that will make *Y* go HIGH.

(b) Explain why the START pulse is needed.



Multivibrators

Latches and flip-flops are considered up to now can be described as **bistable multivibrators** – they have two equally stable O/P states. The device will be put in one of these states depending on the status of the input at the time of the clock pulse

However, we can define two more types of multivibrators:

1. Monostable multivibrator

2. Astable multivibrators

A monostable multivibrator (also called a one-shot) has only one stable state – the default or resting state.

The device will be in this stable state until triggered – then reverts to unstable state for a specific length of time.

The time in this quasi-stable state is determined by external components (usually resistors and capacitors).

Such monostable is normally used for simple timing applications, as capacitors and resistors often have poor stability.

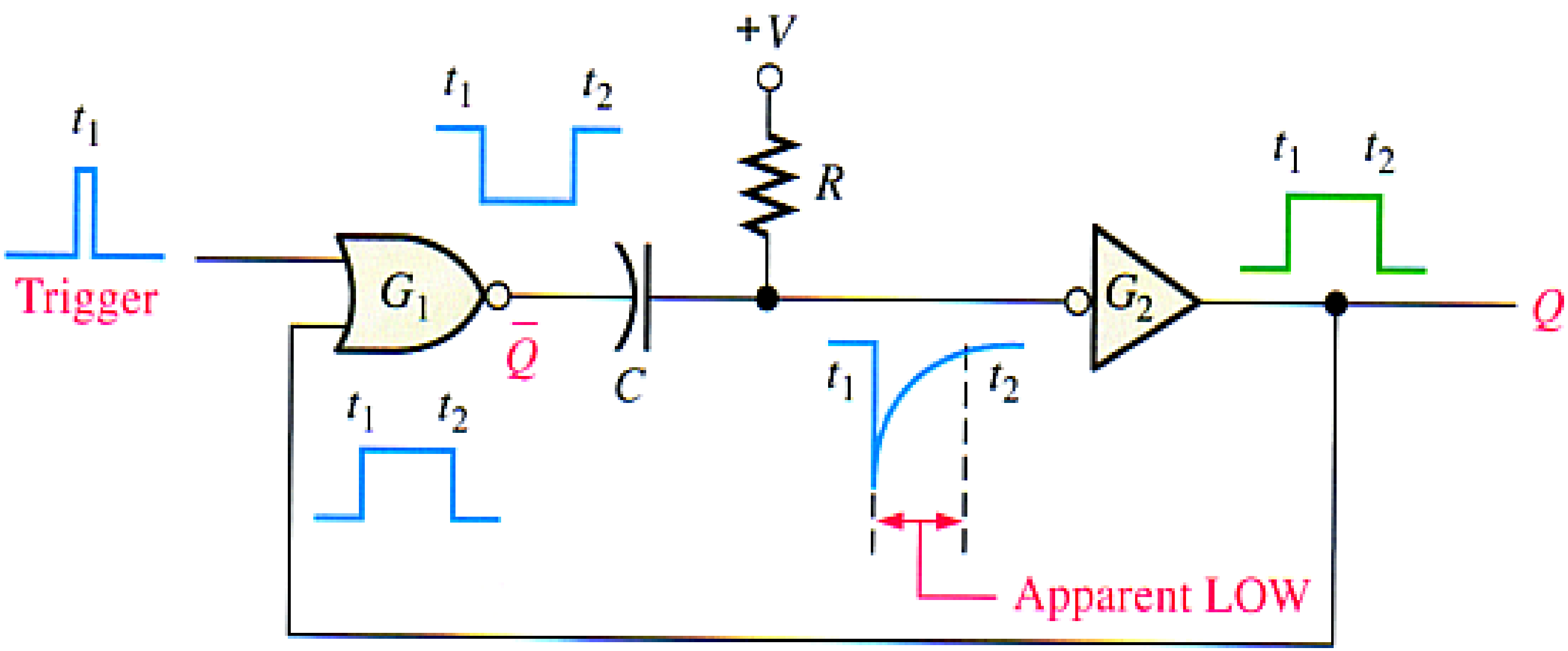
An astable multivibrator has two equally stable output states and will constantly flip between these states while the device is running.

This continuous HI-LO-HI-LO output thus makes it ideal for use as a clock signal. The frequency of output will again be determined by external resistors and capacitors that can be connected to the device.

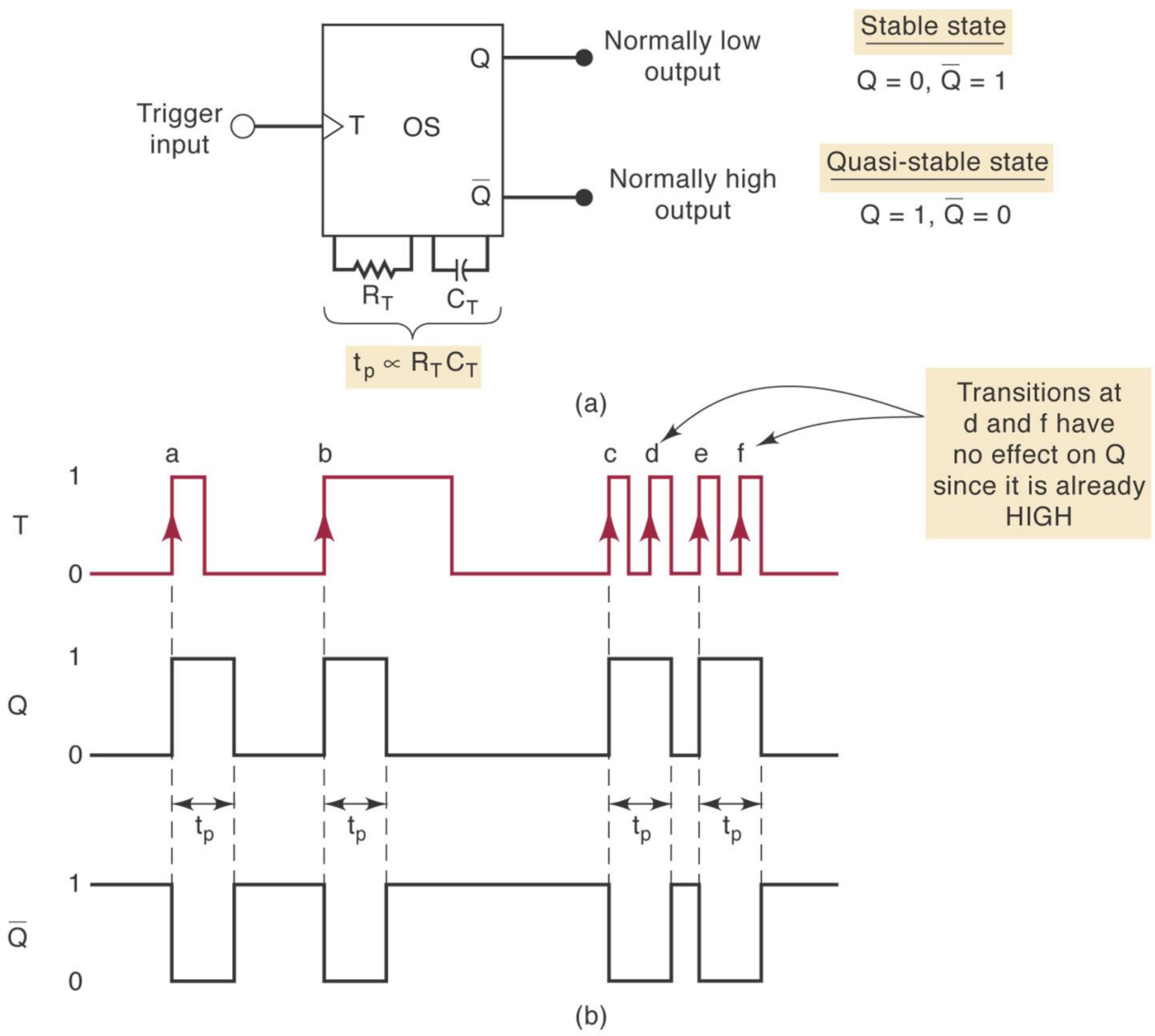
However, the precision and stability of these external components are again generally poor, so that such a clock will not be used in high precision applications.

Setting the time constant in monostables and astables.

We will normally use external components (resistors and capacitors) to set the timing of astable and monostables. The figure below shows how the RC time constant will determine the time in the unstable state (HI) for a monostable.

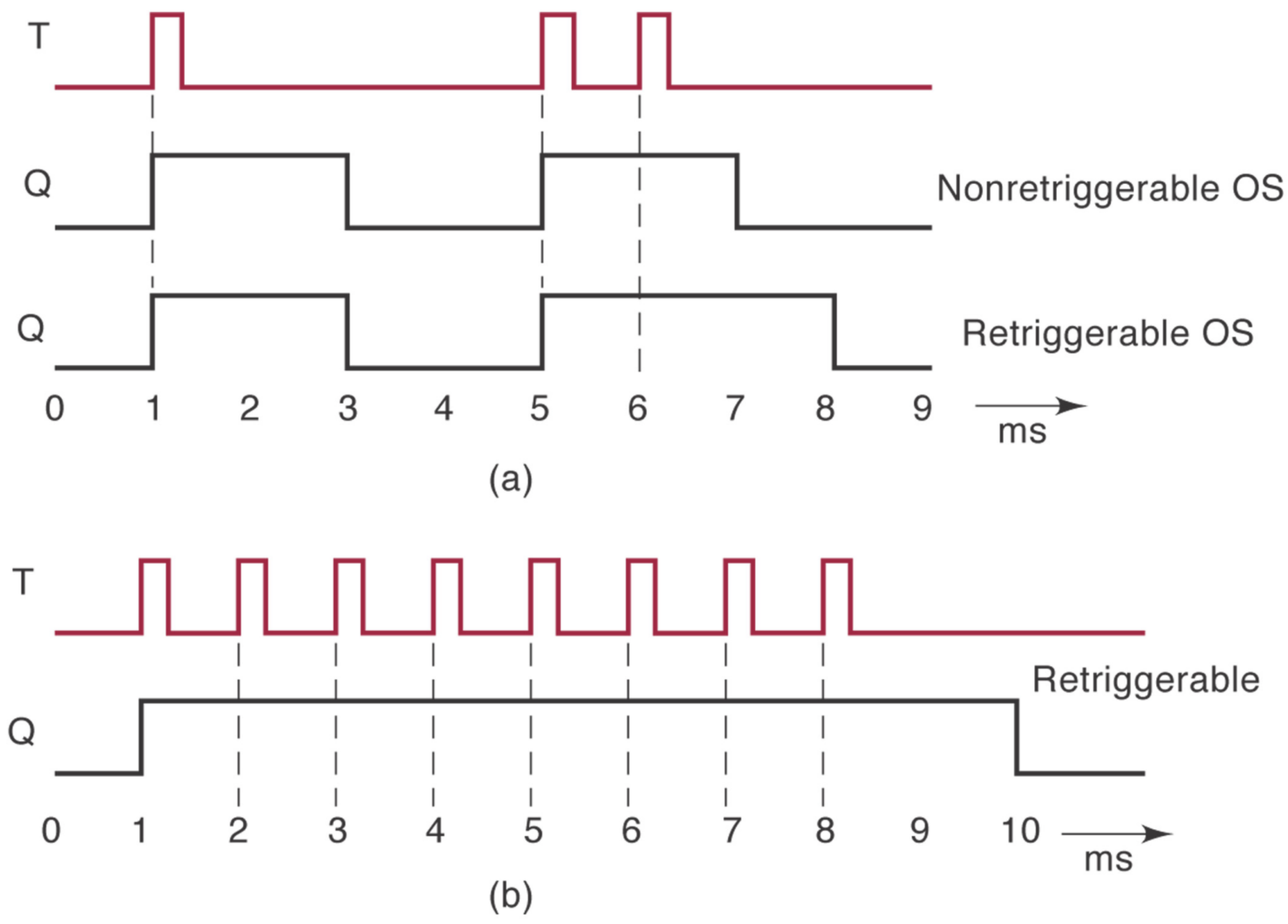


Operation of a One-Shot



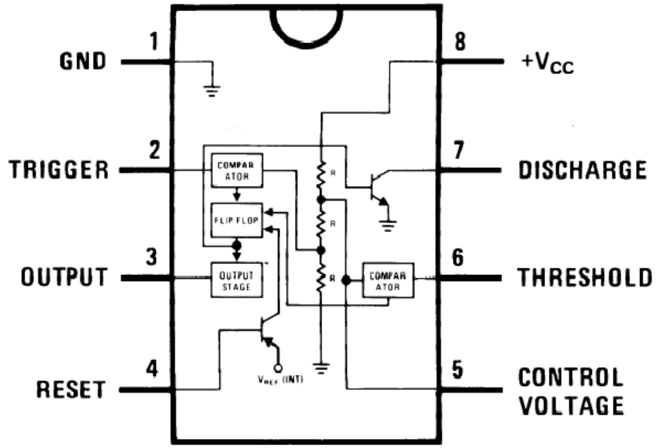
Will be in the stable state until triggered – then in the unstable state for a time determined by the external components

Comparison of a non-retriggerable and retriggerable OS



The 555 Timer

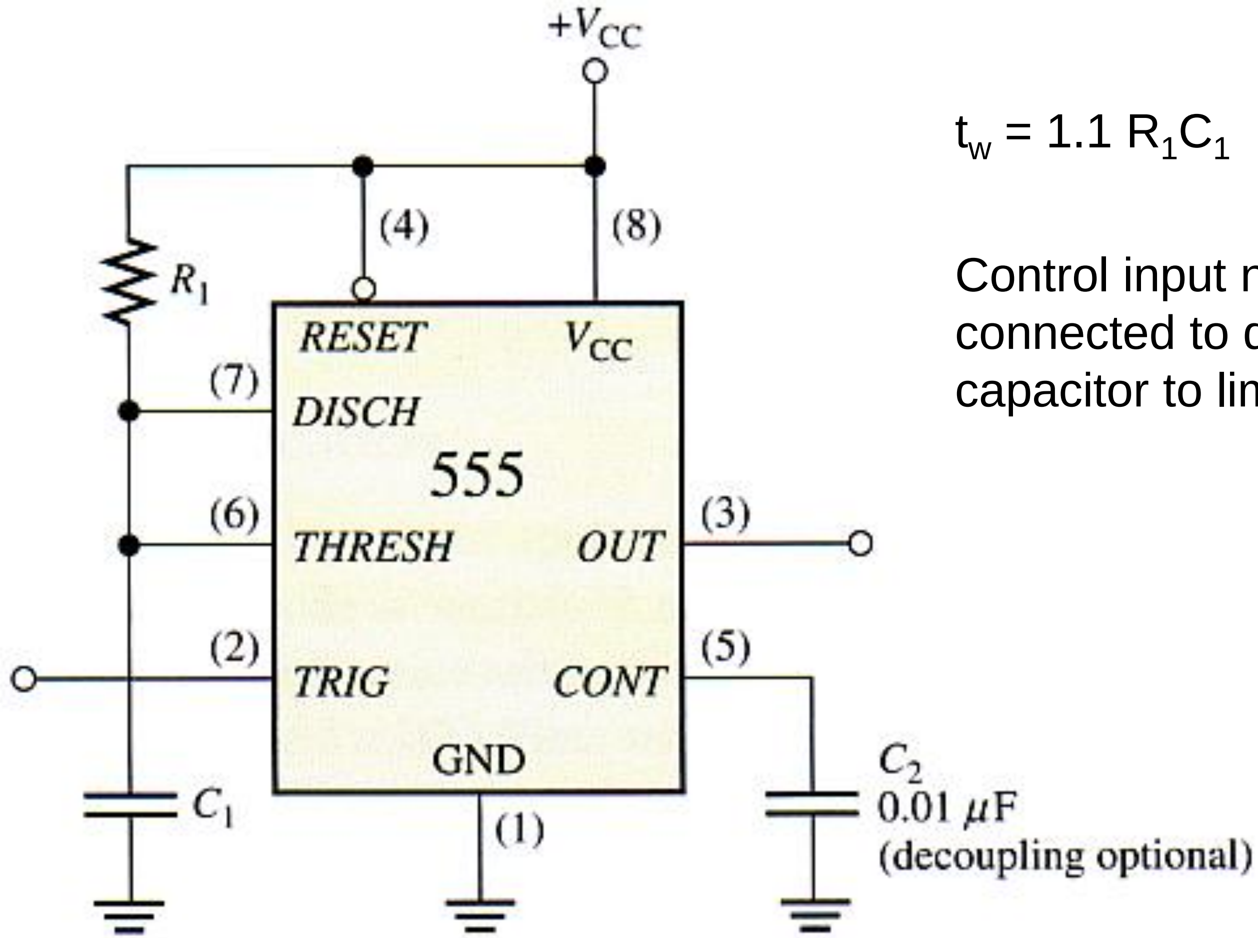
The 555 timer is a very useful IC as it can be configured as either a monostable or astable multivibrator.



LM555 Timer

555 Timer connected as a one-shot (monostable)

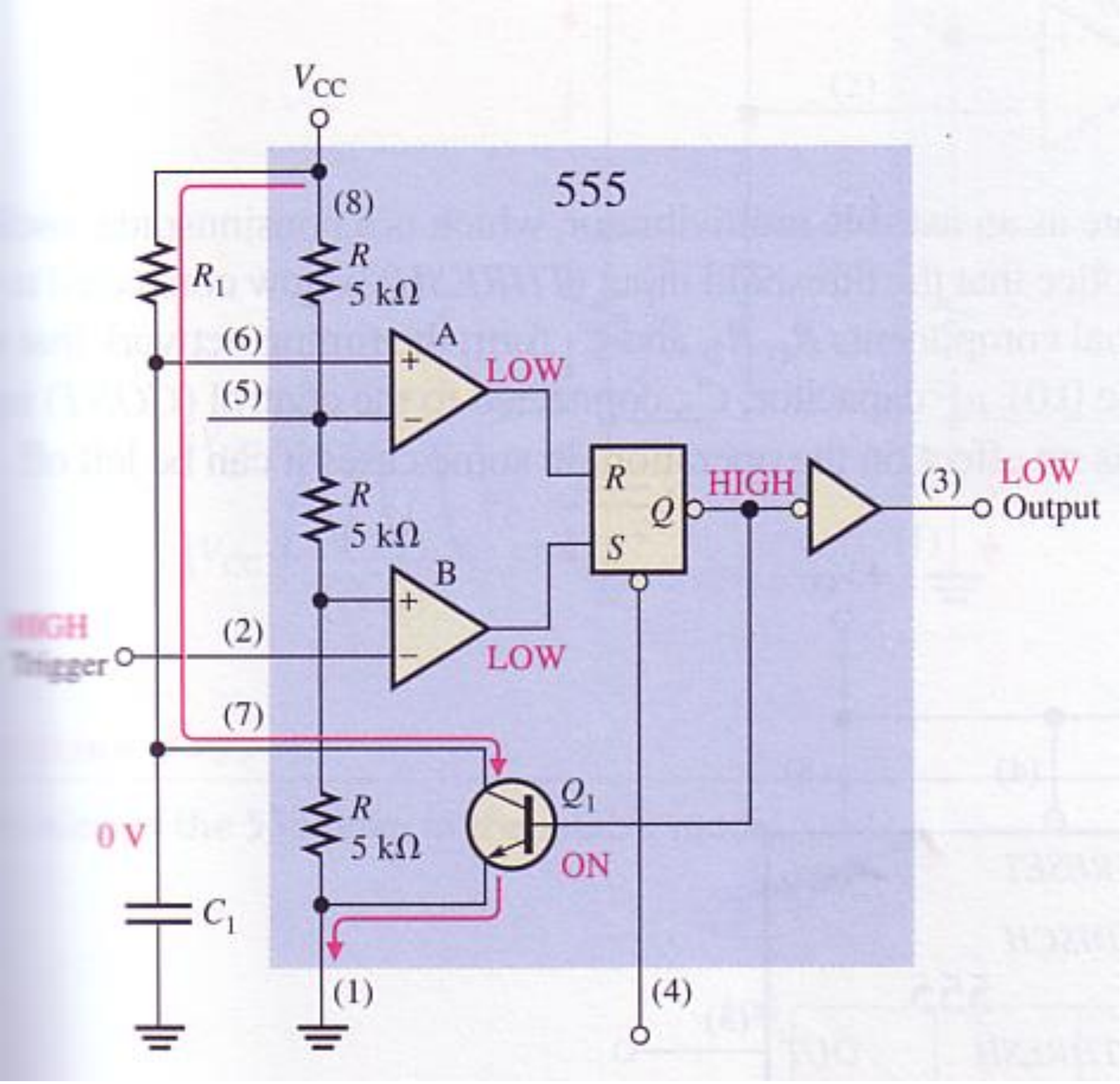
External R_1 and C_1 connected as shown below.



$t_w = 1.1\ R_1C_1$

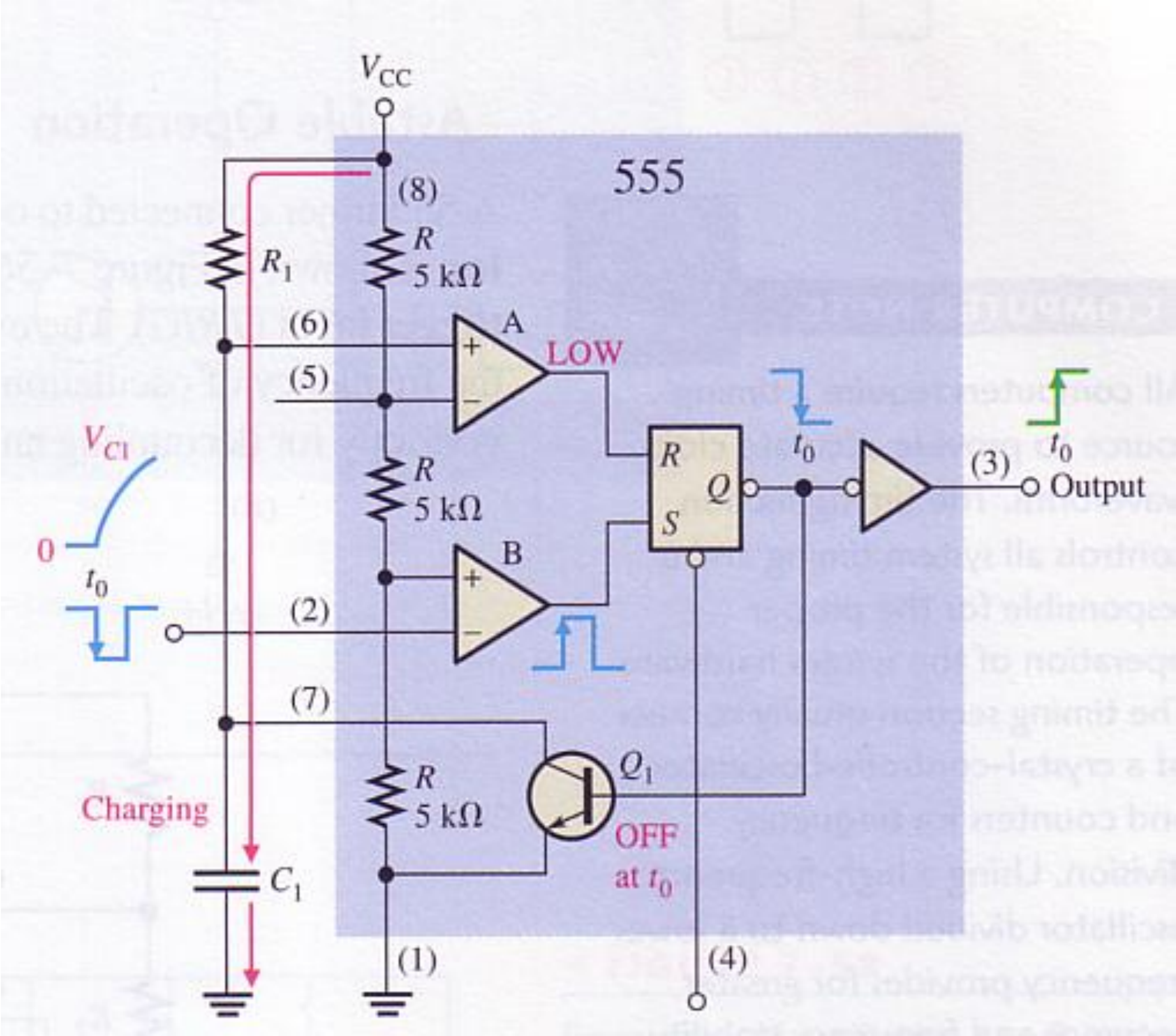
Control input not used – connected to decoupling capacitor to limit noise.

Prior to triggering



O/P is LO and Q1 is ON – keeps C1 discharged.

When triggered

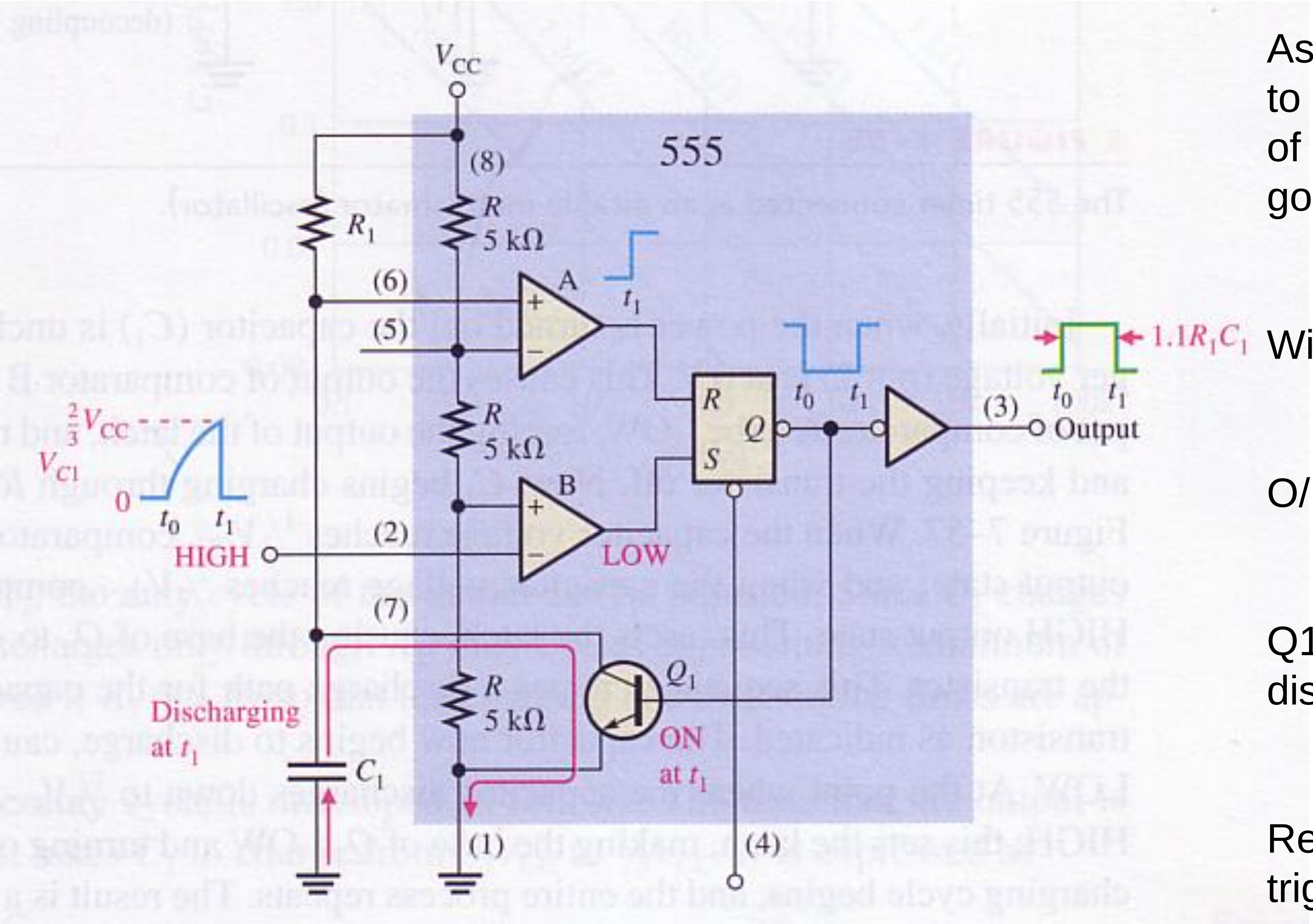


NG trigger pulse is applied – Comparator B switches to HI and the O/P switches to HI

Q1 turns off

Capacitor C1 starts to charge through R1

At end of charging interval



As C1 charges up to 2/3 Vcc, the O/P of comparator A will go HI at t1

Will reset the latch

O/P goes LO at t1

Q1 will turn ON and discharge C1

Ready for next trigger

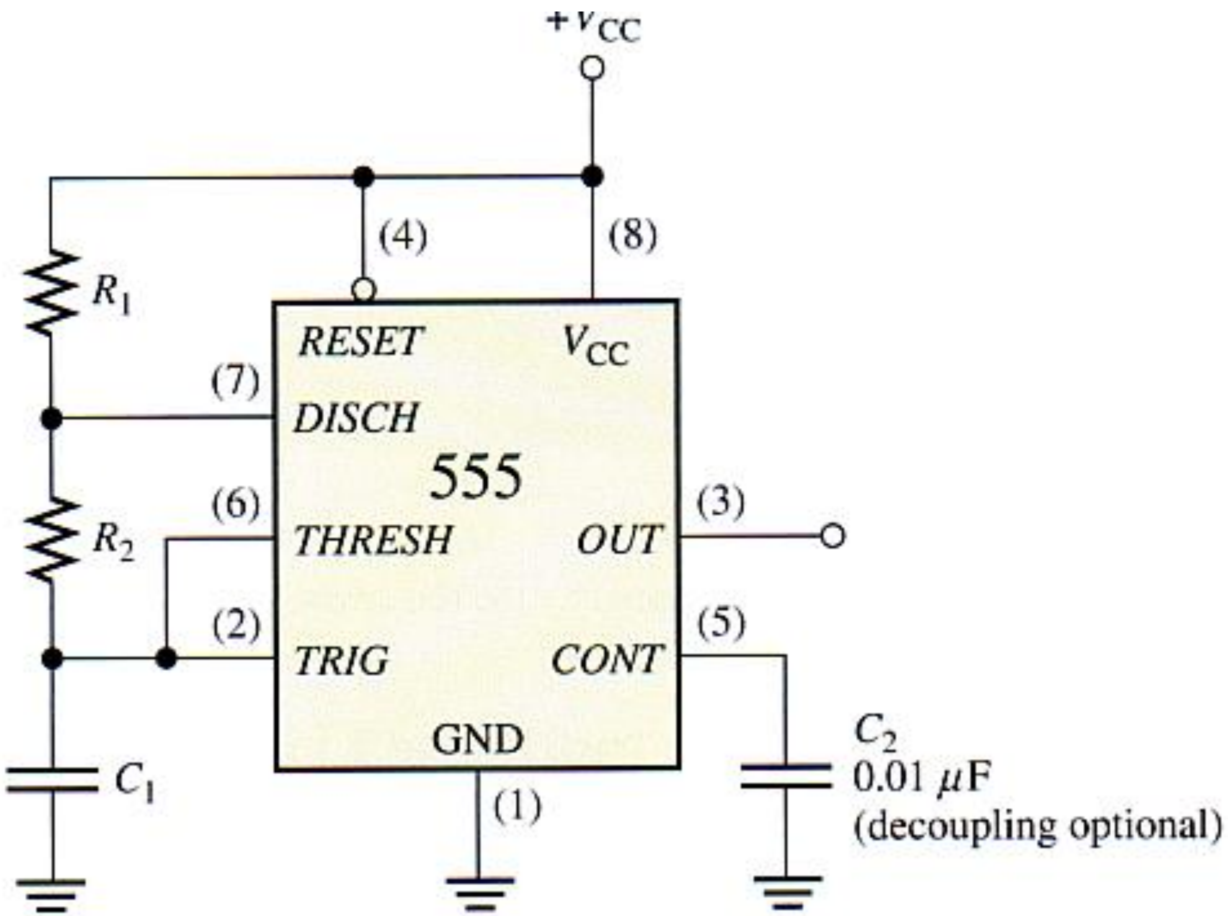
Astable Multivibrators = Clocks

Astable or **free-running multivibrators** switch back and forth between two unstable states. This makes it useful for generating clock signals for synchronous circuits.

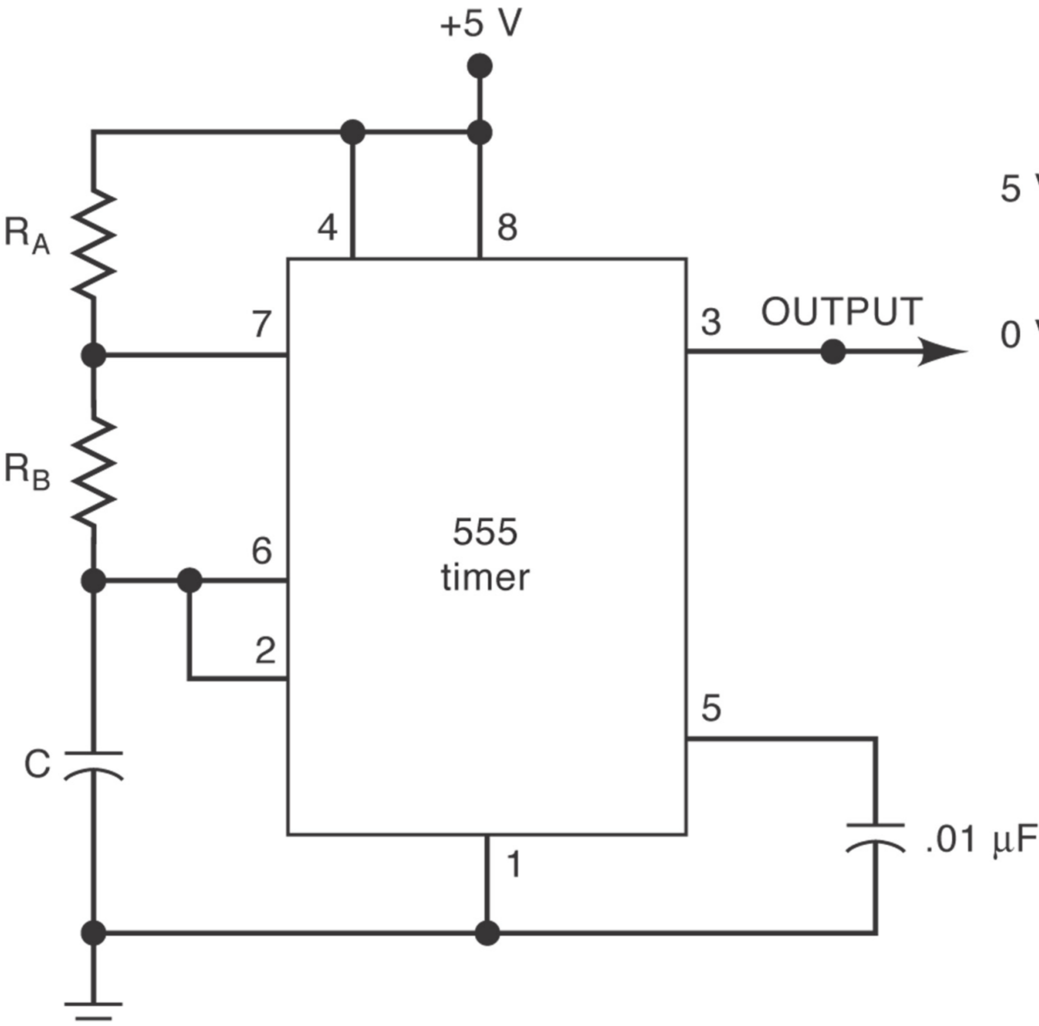
Different applications place varying demand on clock circuits in terms of precision and long term stability

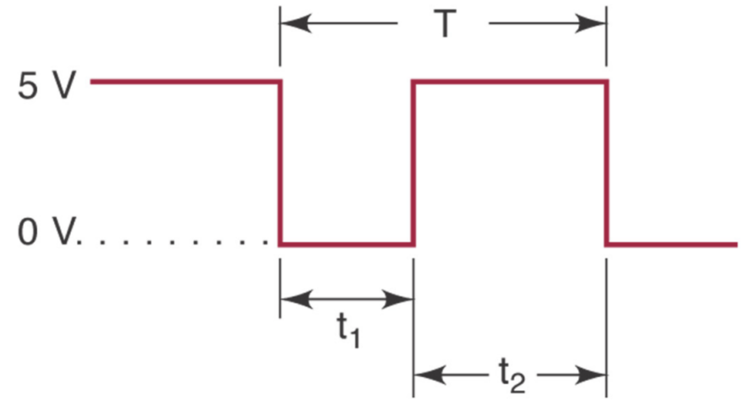
- Low end – Digital IC based clocks
- Medium – Crystal oscillator circuits
- Temperature controlled crystal oscillators, atomic clocks

The 555 connected as an astable (free running) multivibrator

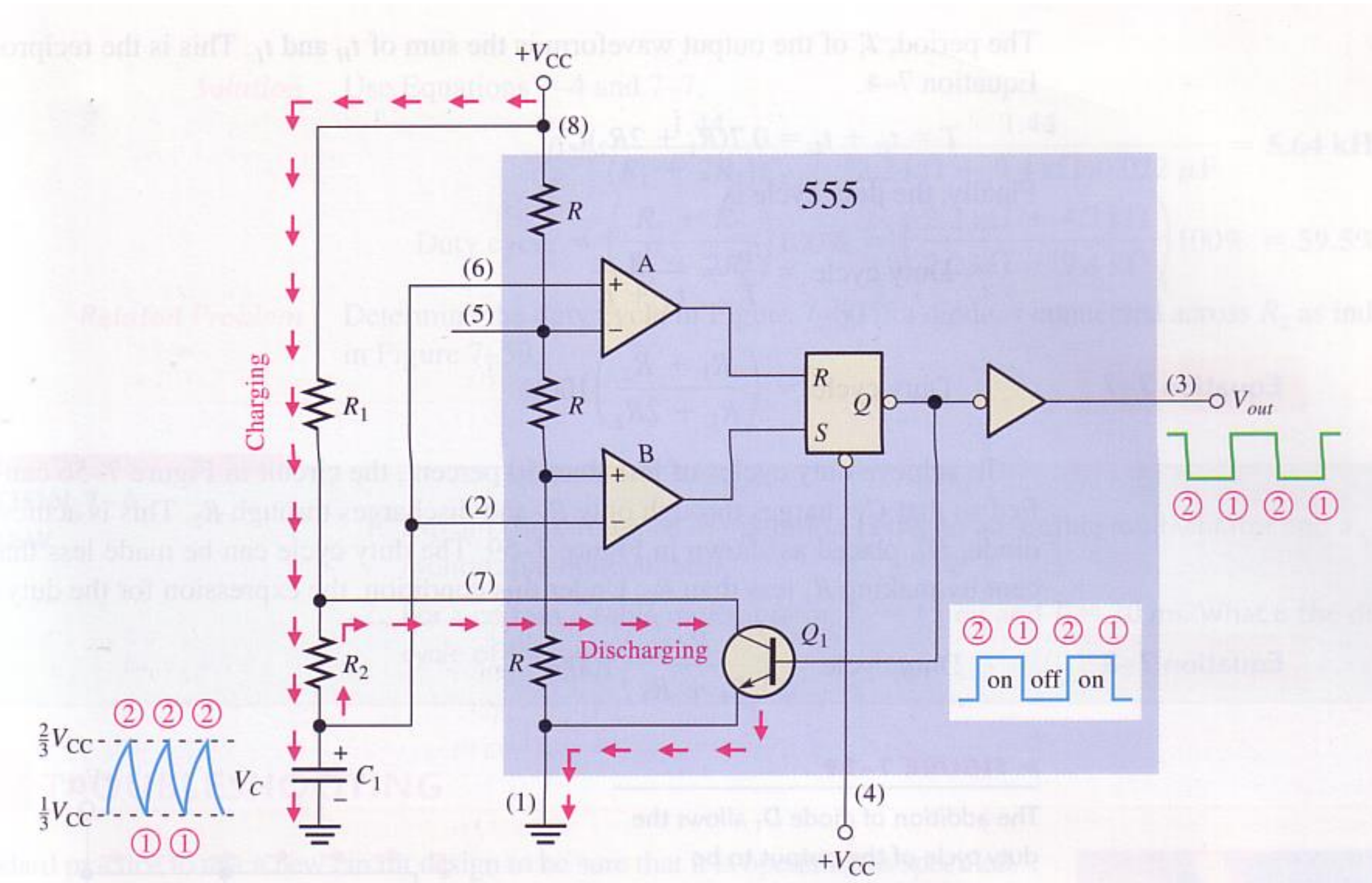


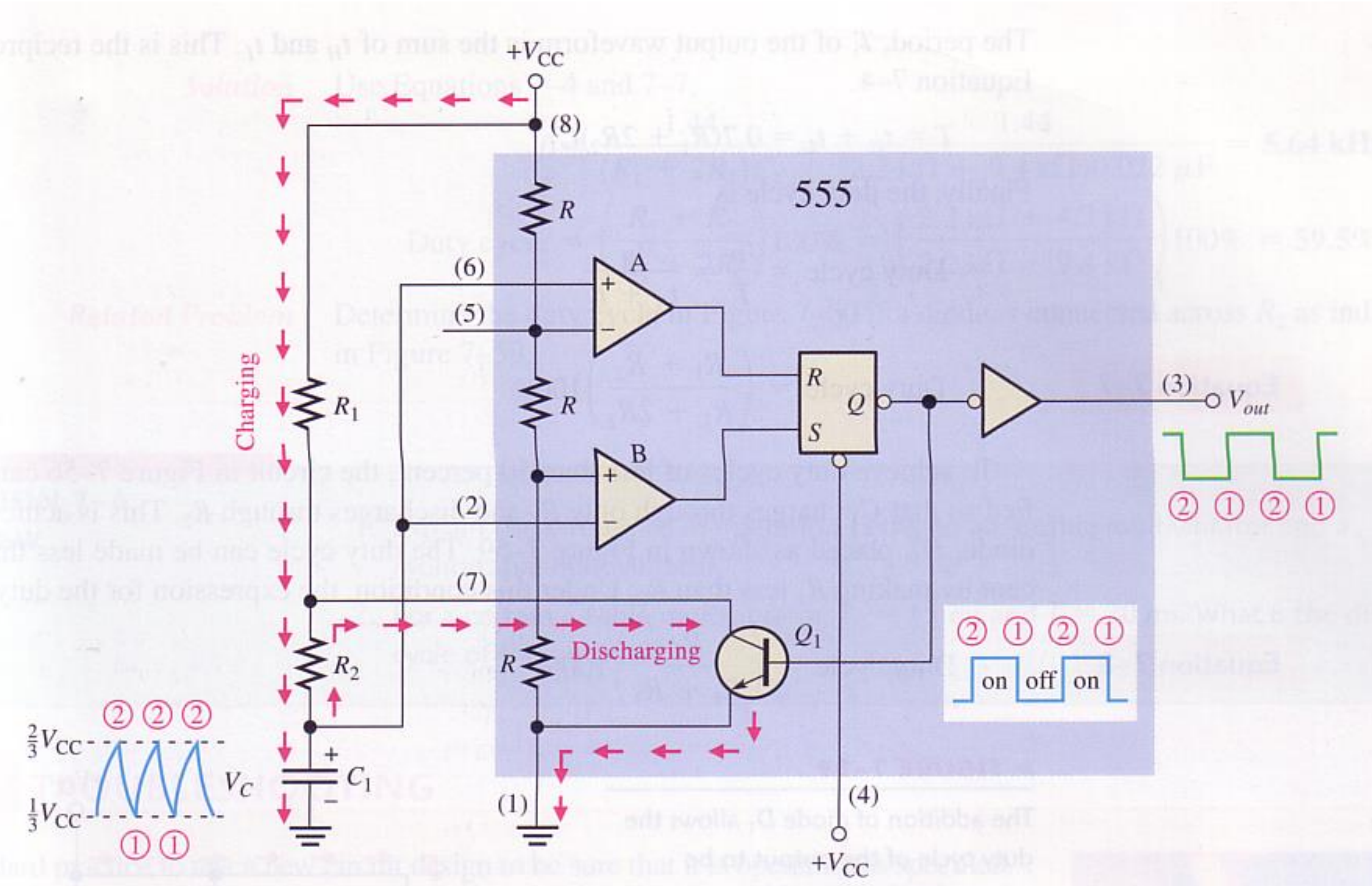
The 555 Timer as an Astable Multivibrator.





$t_1 = 0.693\ R_BC$
$t_2 = 0.693\ (R_A + R_B)C$
$T = t_1 + t_2$
frequency = 1/T
duty cycle = $t_2/T \times 100\%$
R_A] 1 kV
$R_A + R_B$ * 6.6 MV
C] 500 pF


+V_{CC}
(8)
555
R
A
(6)
(5)
R
B
(2)
(7)
R
R
S
Q
(3)
V_{out}
② ① ② ①
Charging
R_1
R_2
Discharging
Q_1
② ① ② ①
on off on
② ② ②
$\frac{2}{3}V_{CC}$
V_C
$\frac{1}{3}V_{CC}$
① ①
C_1
(1)
(4)
+V_{CC}



Astable operation

Threshold (6) now connected to the trigger (2) input

Two external resistors and a capacitor

2nd Capacitor (on Control) for decoupling only – can be left off

Initially capacitor is uncharged – trigger pin (2) is at 0V

This causes O/P of Comp B = HI and Comp A = LO This will keep Q1 OFF

Capacitor will now start to charge through R1 and R2

When it reaches 1/3 Vcc Comp B will switch to LO, when it reached 2/3 Vcc Comp A will switch to HI – Will now reset the latch.

This switches on Q1 and discharges capacitor through R2.

When capacitor discharges to 1/3Vcc Comp B goes HI and sets the latch again.

This turns off Q1 – Another charge cycle of the capacitor starts

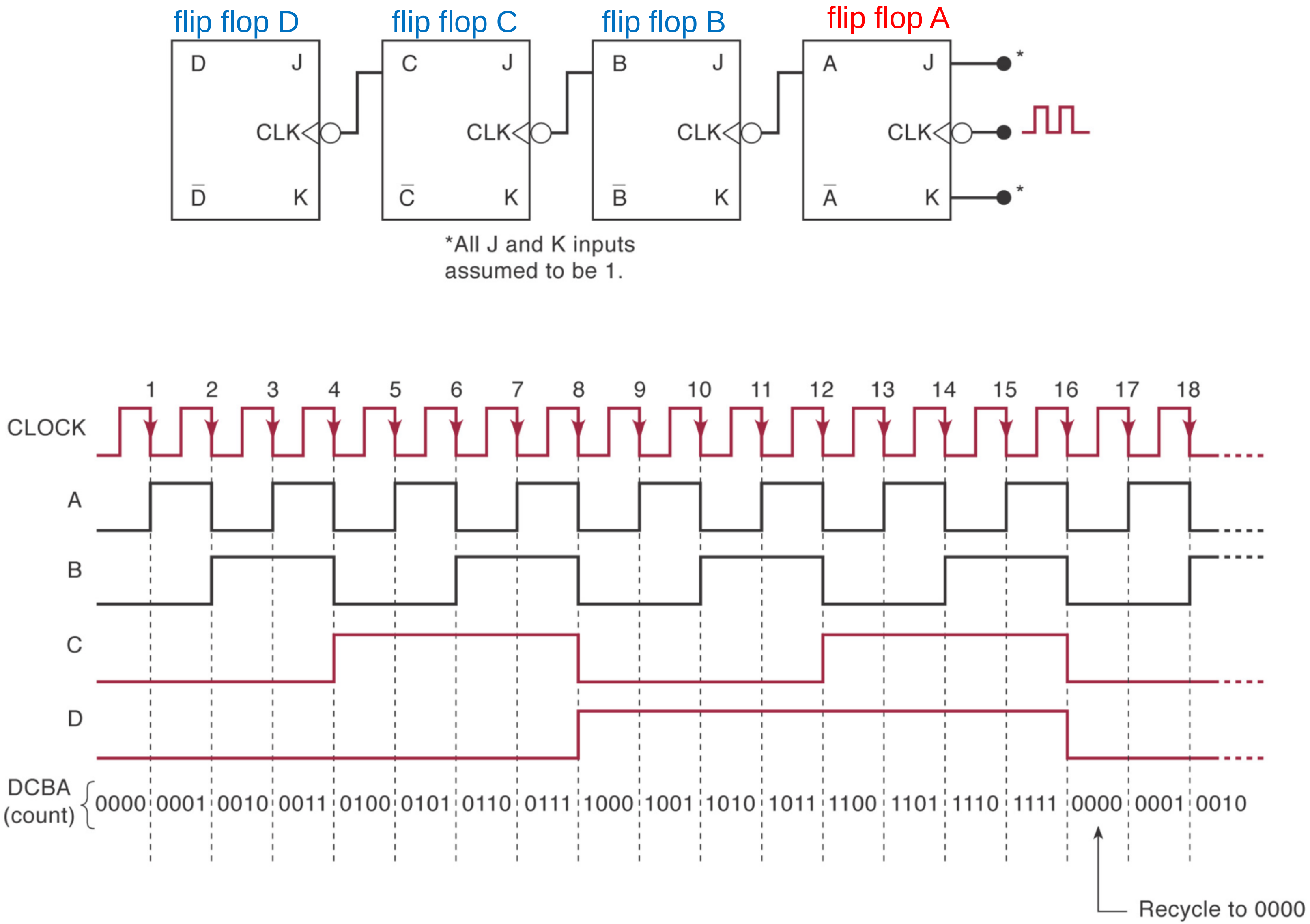
Calculate the frequency and the duty cycle of the 555 astable multivibrator output for $C = 0.001\ \mu F$, $R_A = 2.2\ k\Omega$, and $R_B = 100\ k\Omega$.

Design a 555 free-running oscillator to produce an approximate square wave at 40 kHz. *C* should be kept at 500 pF or greater.

Synchronous vs Asynchronous Counters

- Combining flip-flops and logic gates to form various counters and registers.
- Asynchronous (Ripple) Counters
- Synchronous (Parallel) Counters

Asynchronous (Ripple) Counters



Asynchronous (Ripple) Counters

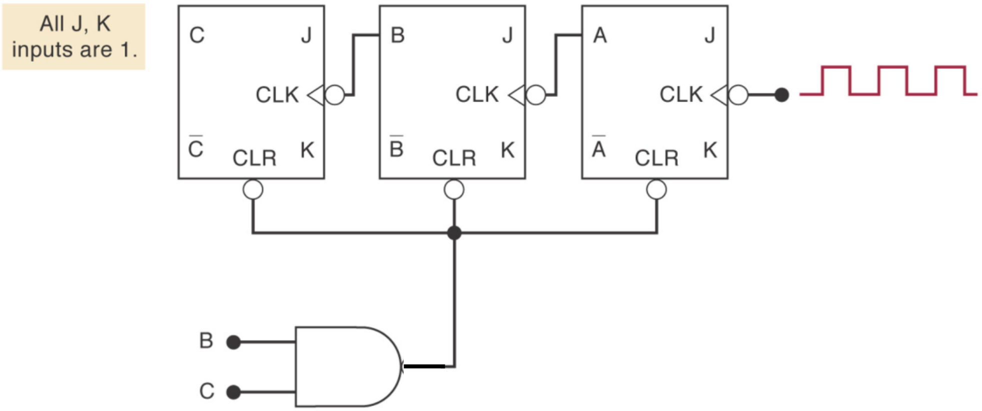
- **<u>MOD number is equal to the number of states that the counter goes through before recycling.</u>** Adding flip flops will increase the MOD number.

- MOD number = 2^n, where n = number of flip-flops.

- The output of the last flip flop (MSB) divides the input clock frequency by the MOD number.

Counters with MOD Number <2^N

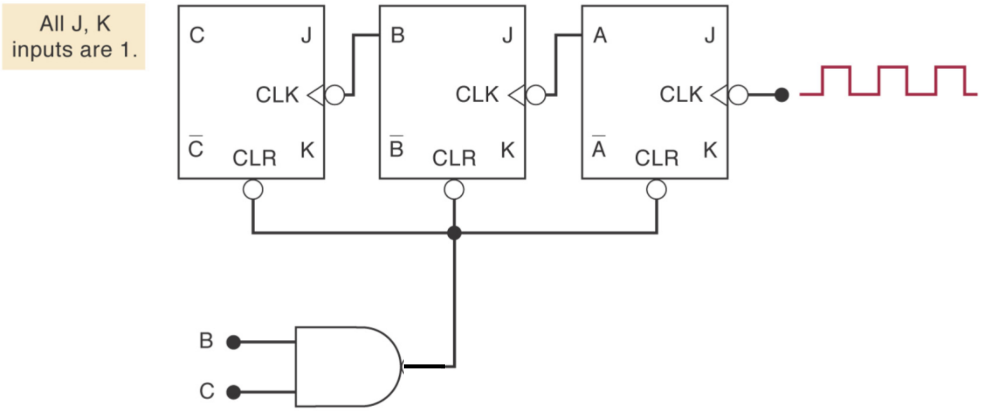
The MOD of a counter can be changed by designing the counter to reset on a specific counter state.

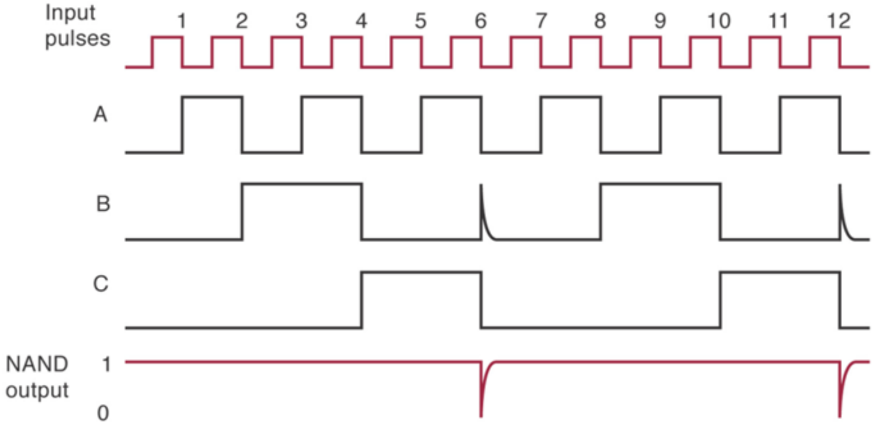


Turns MOD 8 counter into a MOD 6 counter

Counters with MOD Number <2^N

The MOD of a counter can be changed by designing the counter to reset on a specific counter state.





Turns MOD 8 counter into a MOD 6 counter

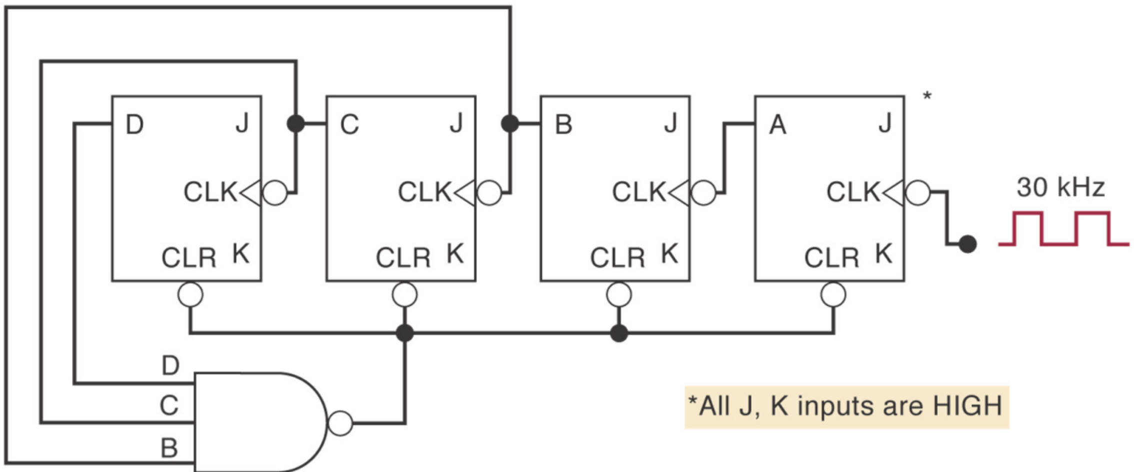
MOD number and frequency @ D for counters below?

Reset on 1110 (14_{10})

Thus MOD 14 cntr*

F = 30kHz/14

= 2.14 kHz



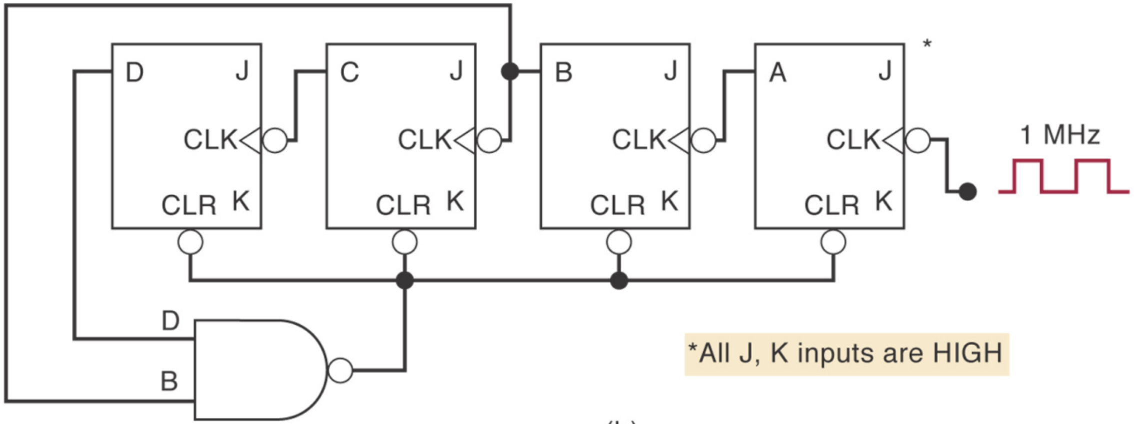
(a)

Reset on 1010 (10_{10})

Thus MOD 10 cntr*

F = 1 MHz/10

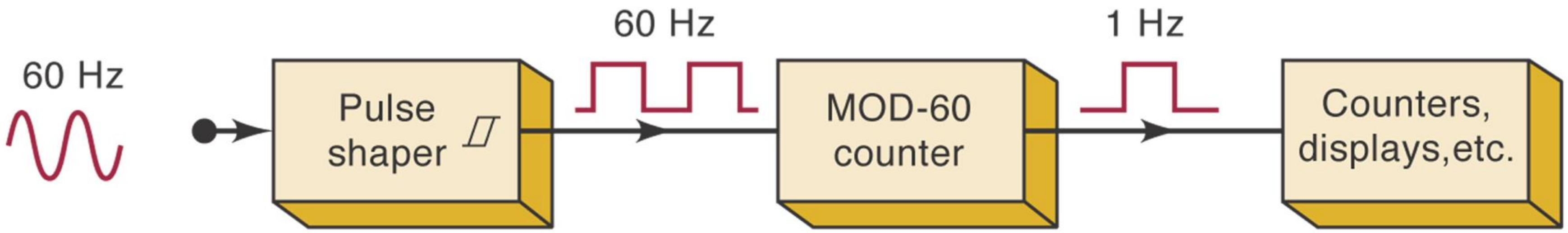
= 100 kHz

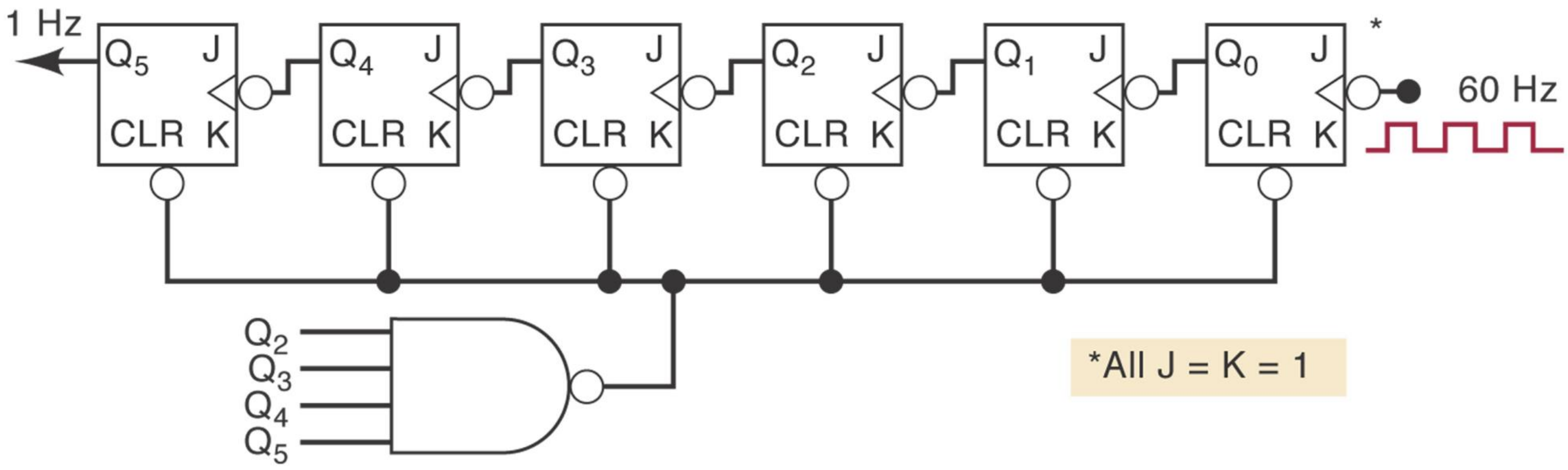


(b)

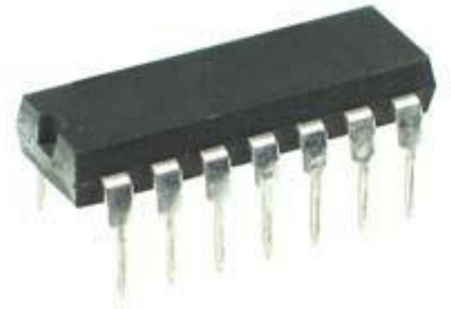
* The reset state is not a counting state

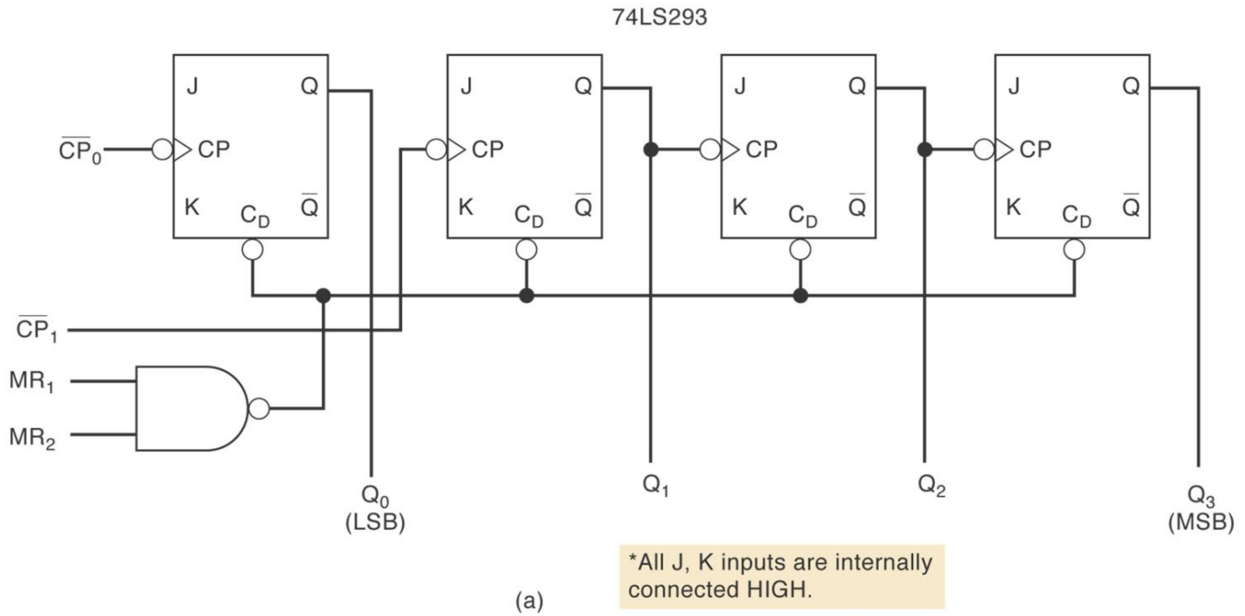
2. Designing a MOD 60 Counter



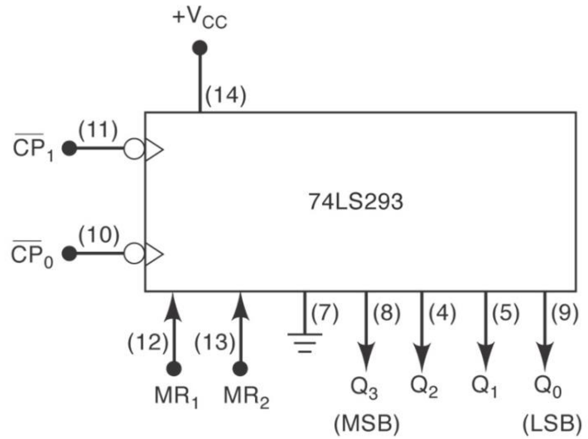


An IC asynchronous counter – the 74LS293



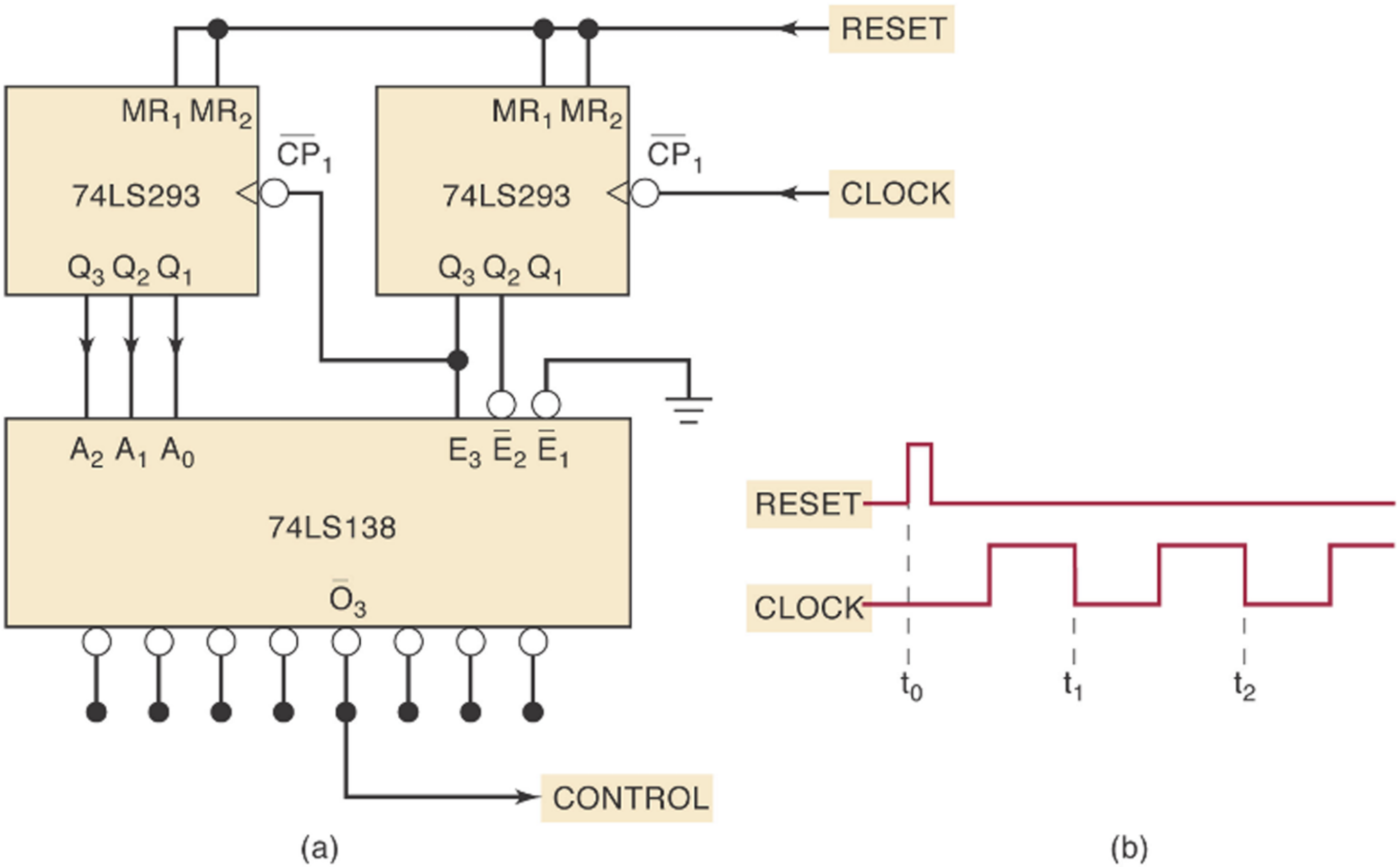


(a)



(b)

7. Examine the circuit below, identify each of the ICs used and explain the operation of the circuit. Now modify the circuit so that it will produce a control signal that will remain LO from CLK t_{20} to t_{24}.



An IC asynchronous counter – the 74LS293

MOTOROLA

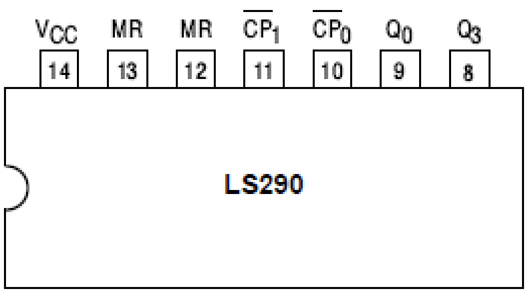
SN54/74LS290
SN54/74LS293

DECADE COUNTER; 4-BIT BINARY COUNTER

The SN54/74LS290 and SN54/74LS293 are high-speed 4-bit ripple type counters partitioned into two sections. Each counter has a divide-by-two section and either a divide-by-five (LS290) or divide-by-eight (LS293) section which are triggered by a HIGH-to-LOW transition on the clock inputs. Each section can be used separately or tied together (Q to CP)to form BCD, Bi-quinary, or Modulo-16 counters. Both of the counters have a 2-input gated Master Reset (Clear), and the LS290 also has a 2-input gated Master Set (Preset 9).

- Corner Power Pin Versions of the LS90 and LS93
- Low Power Consumption . . . Typically 45 mW
- High Count Rates . . . Typically 42 MHz
- Choice of Counting Modes . . . BCD, Bi-Quinary, Binary
- Input Clamp Diodes Limit High Speed Termination Effects

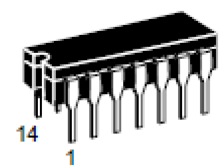
CONNECTION DIAGRAM DIP (TOP VIEW)



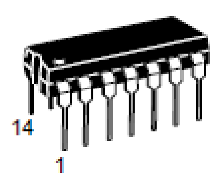
NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

DECADE COUNTER;
4-BIT BINARY COUNTER

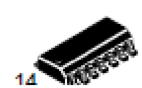
LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 632-08

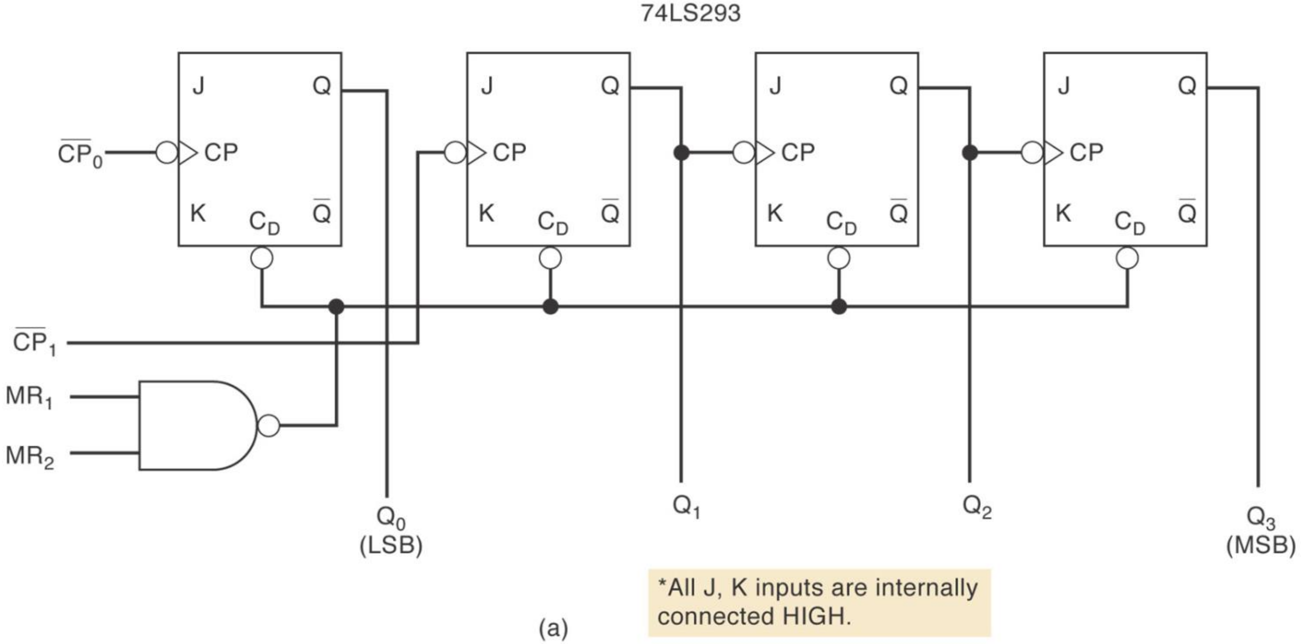


N SUFFIX
PLASTIC
CASE 646-06

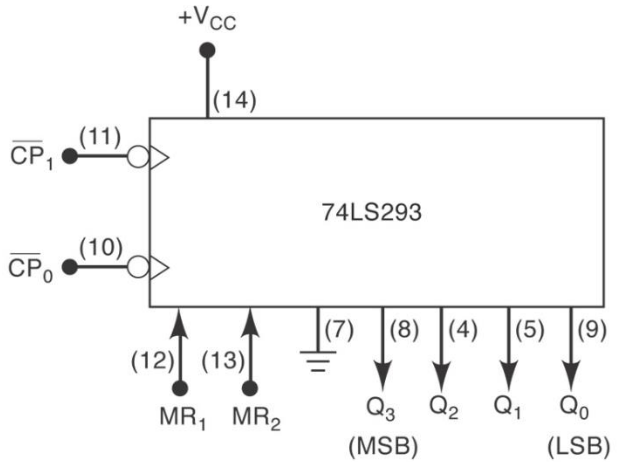


D SUFFIX
SOIC

An IC asynchronous counter – the 74LS293



(a)



(b)

An IC asynchronous counter – the 74LS293

- 4 JK flip flop (FFs)

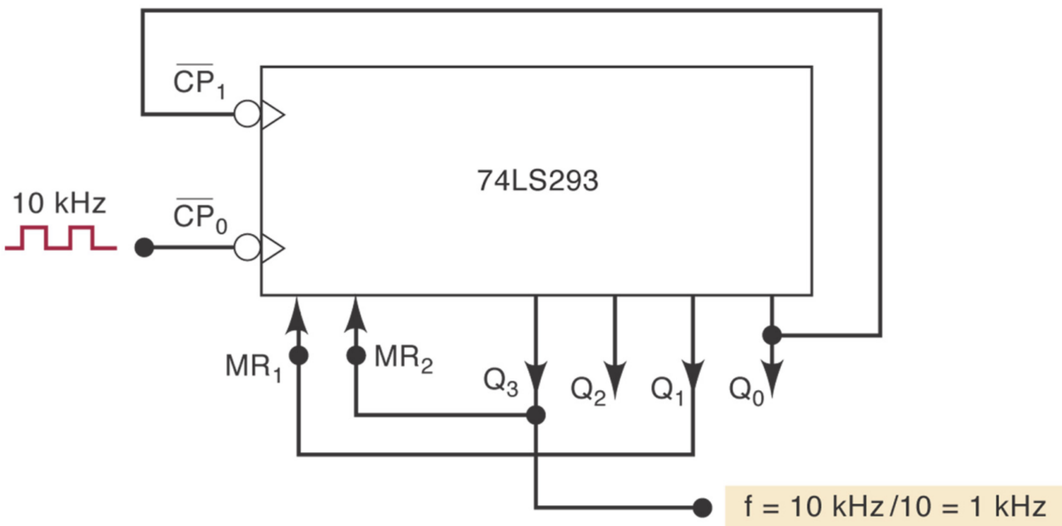
– Each FF has NGT CP (CLK) input

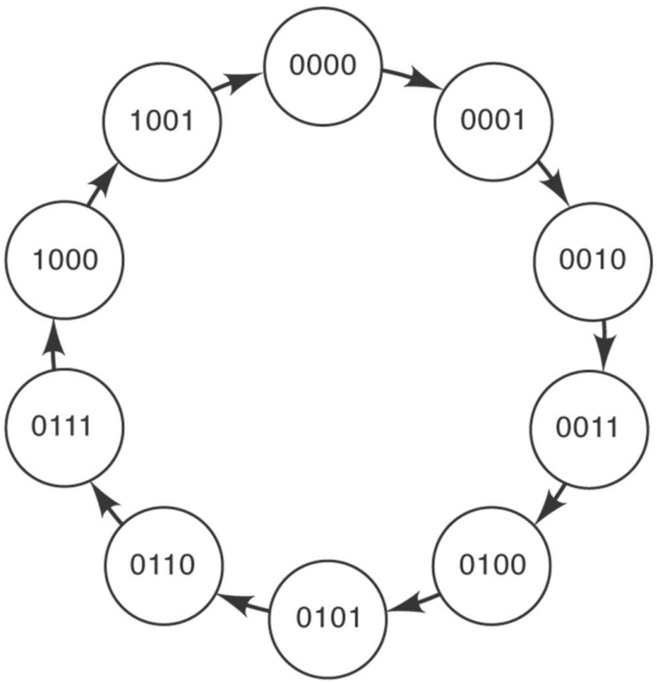
– Each FF has active low asynchronous CLEAR inputs (MR_1 and MR_2 through NAND gate). Both MR I/P's must be HI to clear the counter to 0000.

– FF's Q_1, Q_2, and Q_3 are connected as a 3 bit ripple counter. Q_0 is not connected internally, which allows flexibility in design.

74LS293 as a MOD 10 counter

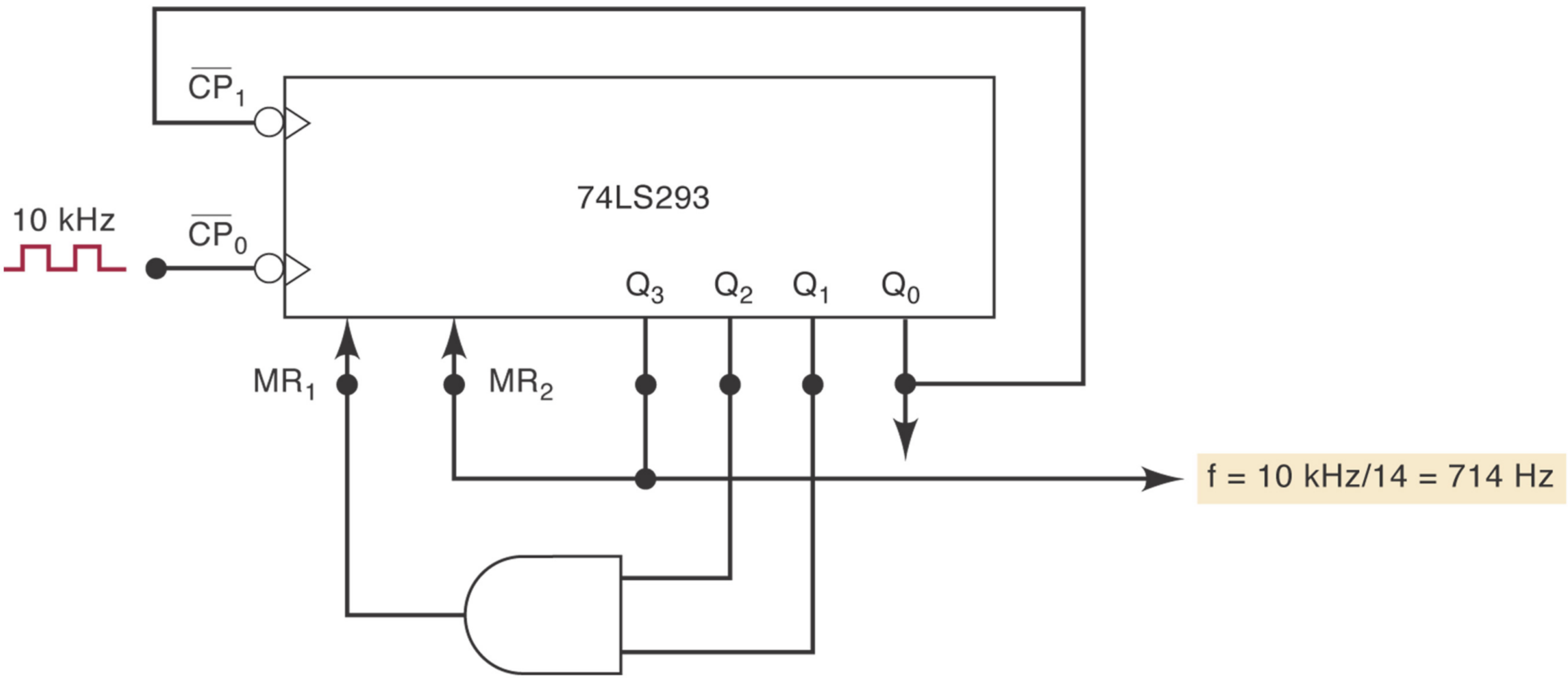
Should reset on 1010



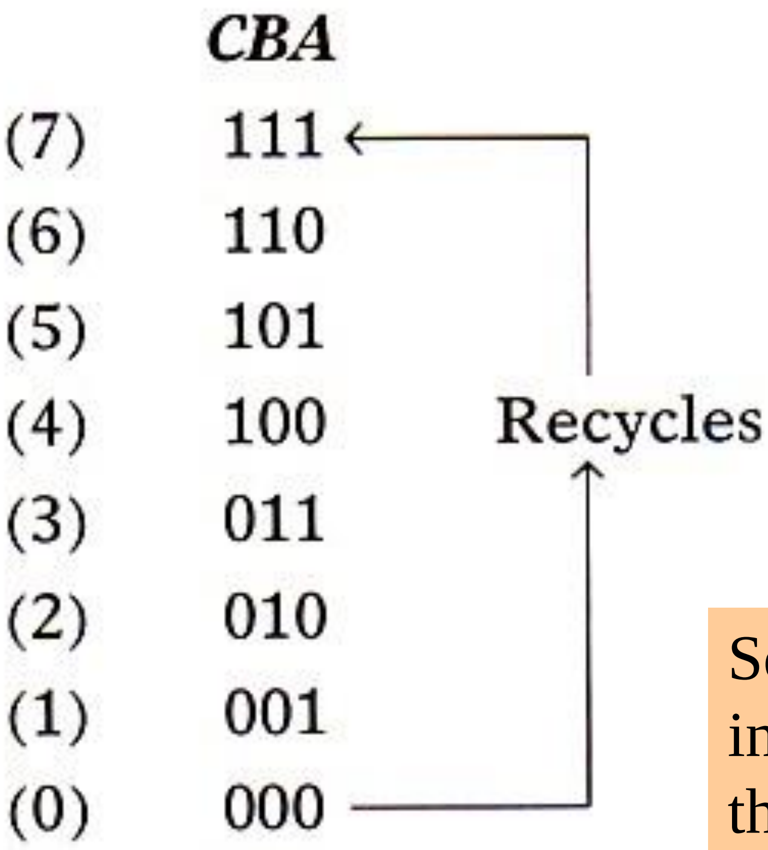


74LS293 as a MOD 14 counter

Should reset on 1110



Asynchronous Down Counter



A: Toggles as normal

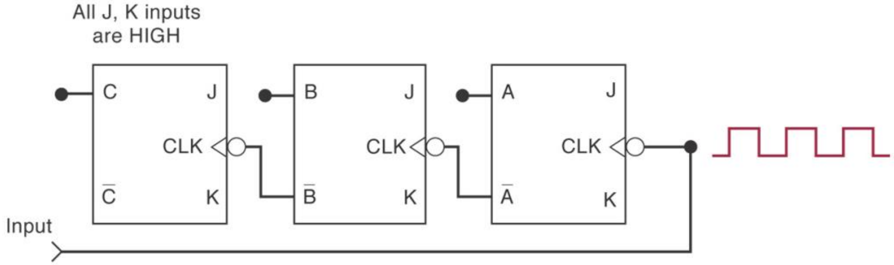
B: Toggles when A goes LO to HI

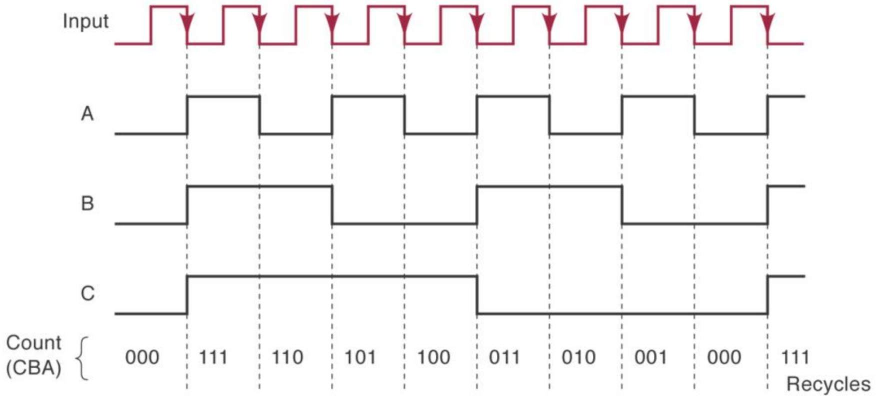
C: Toggles when B goes LO to HI

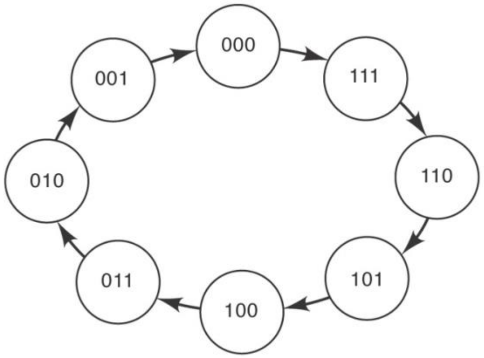
Solution: Drive each clock input from the inverted input of the preceding FF

Asynchronous Down Counter

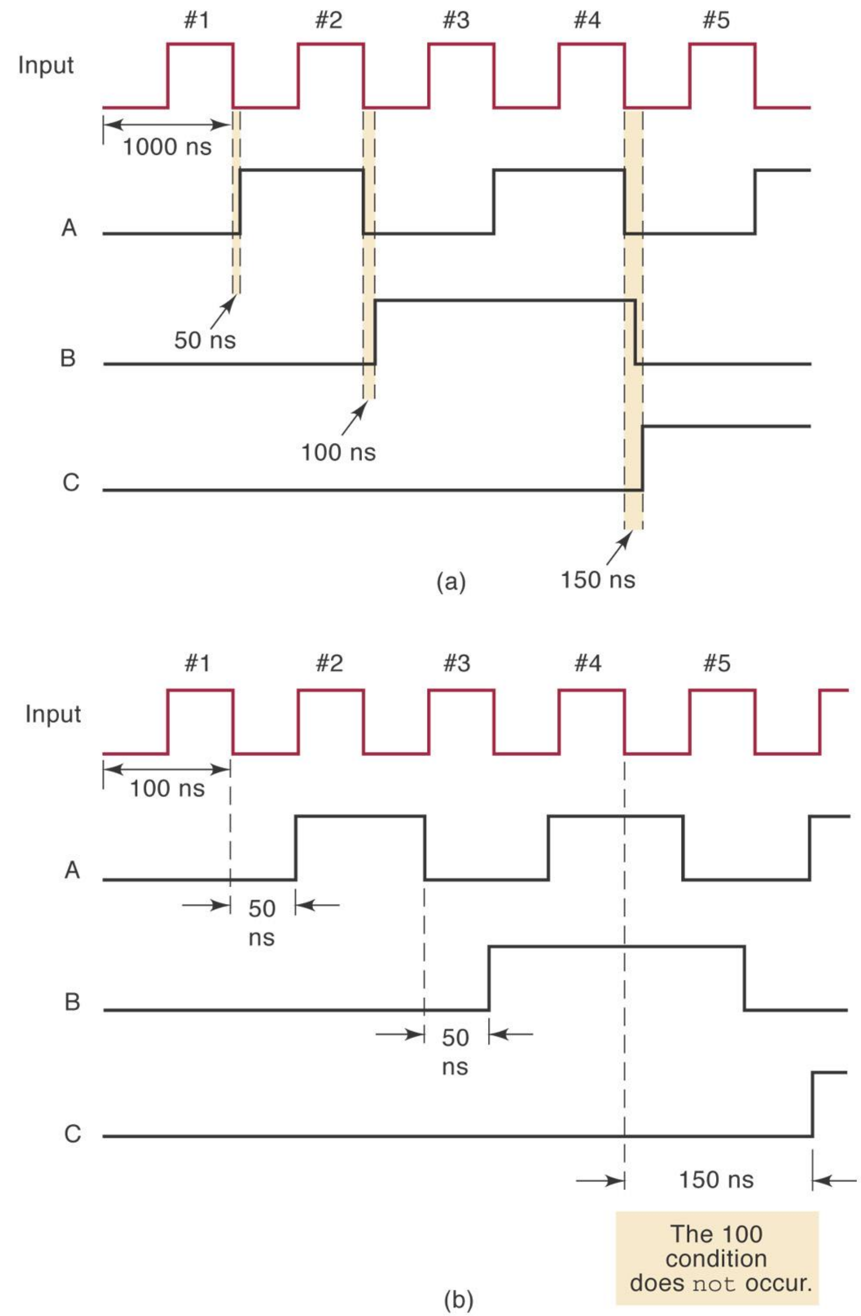
MOD 8 down counter







The Problem with Ripple Counters: Propagation Delay



For proper operation:

$$T_{\text{clock}} \geq N \text{ x } t_{pd}$$

$$F_{max} = 1/(N \text{ x } t_{pd})$$

where N = number of FFs.

Example: Four bit ripple counter constructed from a 74LS112 J-K FF.

t_{PLH} = 16 ns and t_{PHL} = 24 ns

What is highest frequency that can be used ?

f_{max} = 1/(4 x 24 ns) = 10.4 MHz

<u>This frequency limitation inhibits the use of asynchronous counters at moderately high frequencies</u>

Example: What is the highest frequency that a ripple counter with a MOD number 60 constructed from a 74LS112 J-K flip flop can operate at?

t_{PLH} = 16 ns and t_{PHL} = 24 ns

Example: What is the highest frequency that a ripple counter with a MOD number 60 constructed from a 74LS112 J-K flip flop can operate at?

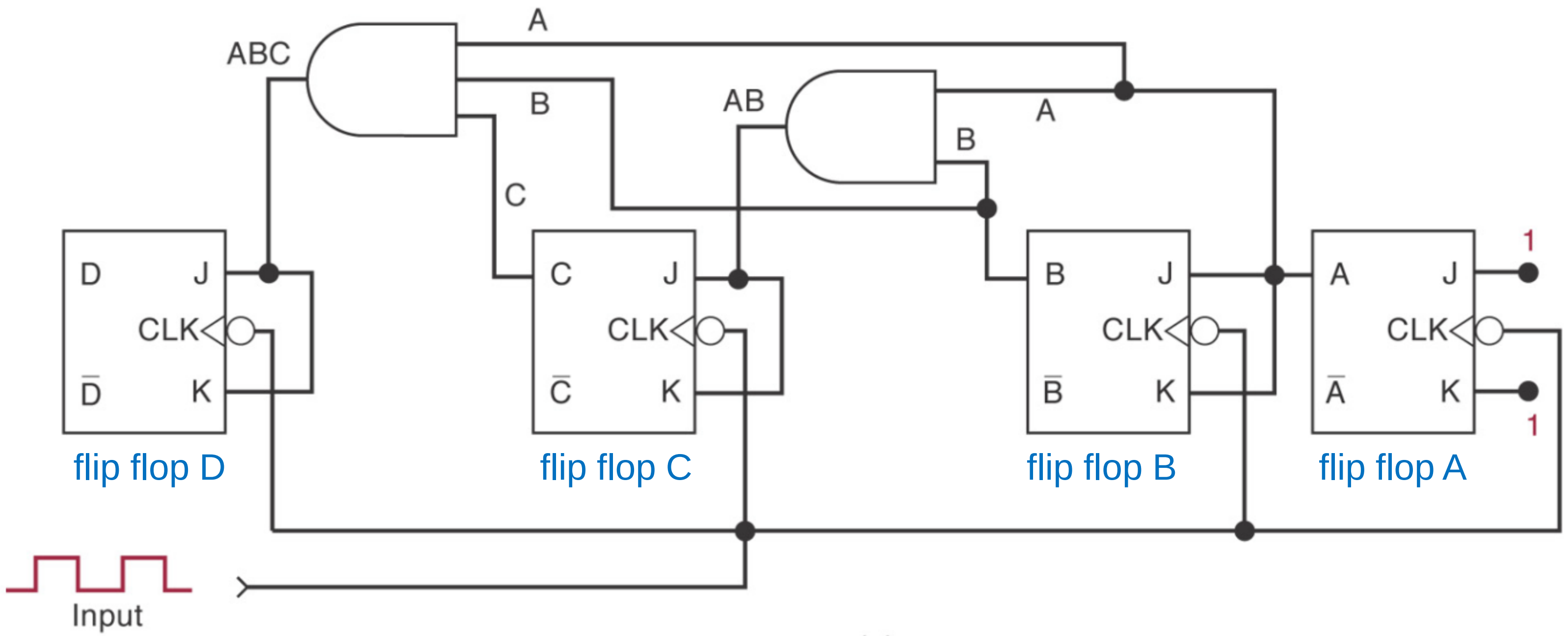
t_{PLH} = 16 ns and t_{PHL} = 24 ns

f_{max} = 1/(6 x 24 ns) = 6.9 MHz

Synchronous (Parallel) Counters

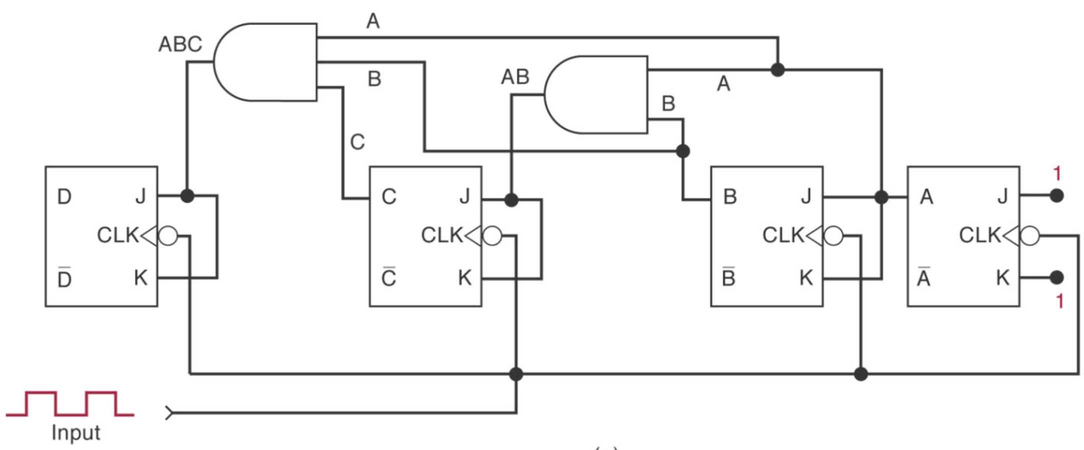
- Solution to the previous problem - All FFs are triggered by CPs simultaneously
- Each FF has J and K inputs connected so they are HIGH only when the outputs of all lower-order FFs are HIGH.
- The total propagation delay will be the same for any number of FFs.
- Synchronous counters can operate at much higher frequencies than asynchronous counters.

Synchronous MOD 16 counter



(a)

Synchronous MOD 16 counter



(a)

Truth table

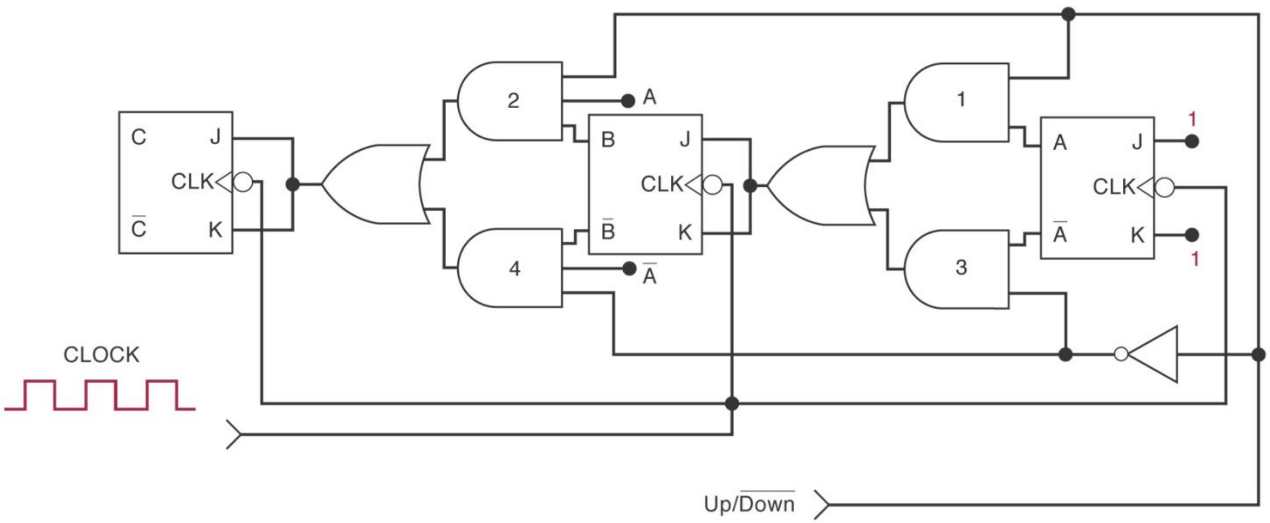
Count	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0
.	.	.	.	.
.	.	.	.	.
.	.	etc.	.	.

(b)

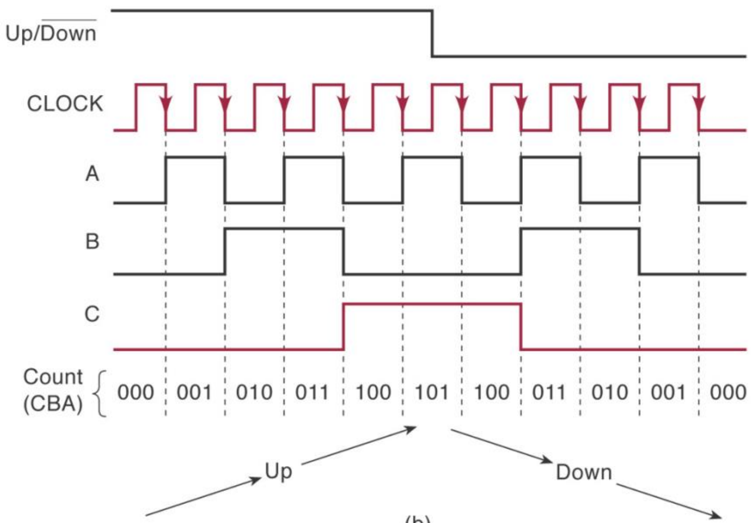
Synchronous Down and Up Counters

- The synchronous counter can be converted to a down counter by using the inverted FF outputs to drive the JK inputs.
- A synchronous counter can be made an up/down counter by suitable connections.
- Available in IC form with Up/<u>Down</u> control input.

MOD 8 Up/Down Counter



(a)



(b)

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.
- The next figure illustrates an asynchronous preset.
- There are several TTL and CMOS devices that provide both synchronous and asynchronous presetting.

General Counter Design

In the case of the previous two binary counters we find that the counters consist of:

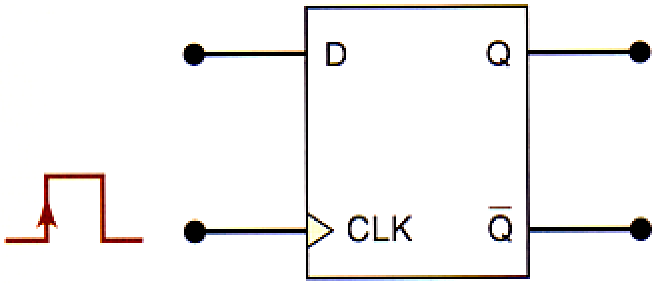
- Sequential element (the flip flops) that provides an output (Q) based on the state of their J, K inputs when the synchronous clock pulse occurs.
- The combination logic elements that ensures the J,K inputs are as desired to achieve the correct transitions in the FF.
- We could design the 4 bit up counter rather intuitively and with a little bit more thinking provide the up-down facility for the 3-bit counter.
- <u>Will now look at a general procedure for designing synchronous counters, taking into account that the states may not be the normal binary order.</u>

In short, the excitation table for J-K flip flops

Present Q	Next Q	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Similarly for D FF's

Truth Table



D	CLK	Q
0	↑	0
1	↑	1

Excitation table

Present State Q(n)	Next State Q(n+1)	Input D
0	0	0
0	1	1
1	0	0
1	1	1

Example from class: regular up counter

Present State Q(n)				Next State Q(n+1)			FF States Needed					
C	B	A		C	B	A	Jc	Kc	Jb	Kb	Ja	Ka
0	0	0		0	0	1	0	x	0	x	1	x
0	0	1		0	1	0	0	x	1	x	x	1
0	1	0		0	1	1	0	x	x	0	1	x
0	1	1		1	0	0	1	x	x	1	x	1
1	0	0		1	0	1	x	0	0	x	1	x
1	0	1		1	1	0	x	0	1	x	x	1
1	1	0		1	1	1	x	0	x	0	1	x
1	1	1		0	0	0	x	1	x	1	x	1

Step 5: Design the logic circuits needed to generate the levels required at each J and K input.

Method 1: Use K-maps for simplification and implement the combinational logic circuit for each of the J,K inputs.

For Jc

	/A	A
/C/B	0	0
/C.B	0	1
C.B	x	x
C./B	x	x

Jc=AB

For Kc

	/A	A
/C/B	x	x
/C.B	x	x
C.B	0	1
C./B	0	x

Kc=AB

For Jb

	/A	A
/C/B	0	1
/C.B	x	x
C.B	x	x
C./B	0	1

Jb=A

For Kb

	/A	A
/C/B	x	x
/C.B	0	1
C.B	0	1
C./B	x	x

Kb=A

For Ja

	/A	A
/C/B	1	x
/C.B	1	x
C.B	1	x
C./B	1	x

Ja=1

For Ka

	/A	A
/C/B	x	1
/C.B	x	1
C.B	x	1
C./B	x	1

Ka=1

We thus have logic requirements:

Ja = Ka = 1

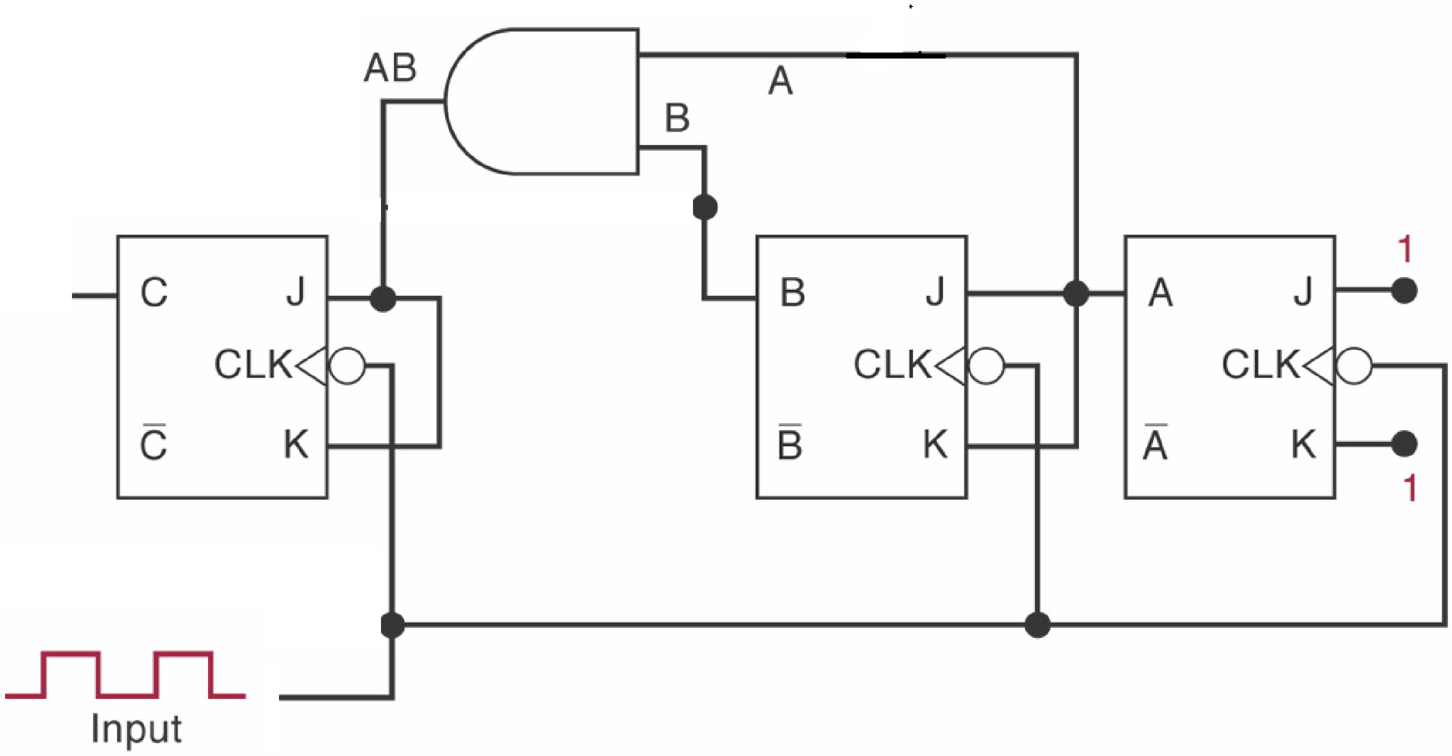
Jb = Kb = A

Jc = Kc = AB

SIMILAR TO OUR RATHER INTUITIVE DESIGN OF EARLIER !

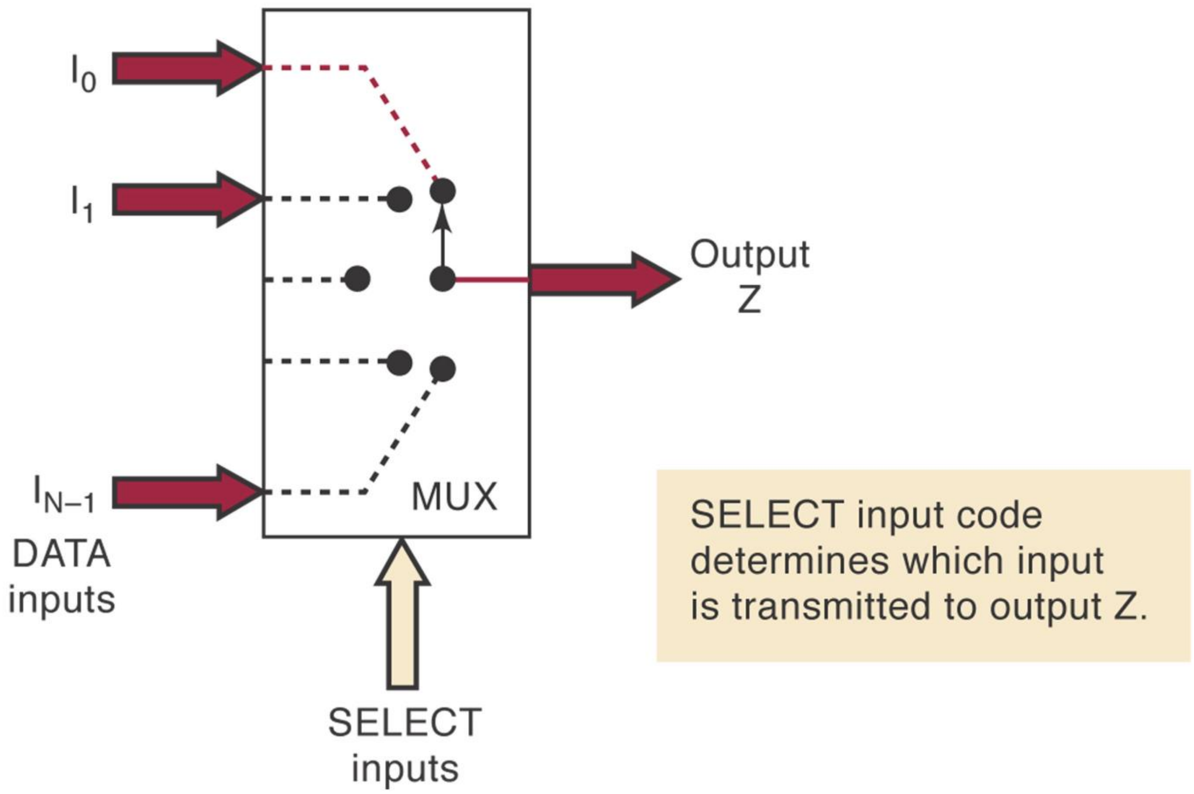
This logic can be implemented either using logic gates or using MUXes

Logic implemented with logic gates for a 4-bit synchronous UP counter – same what we have intuitively done

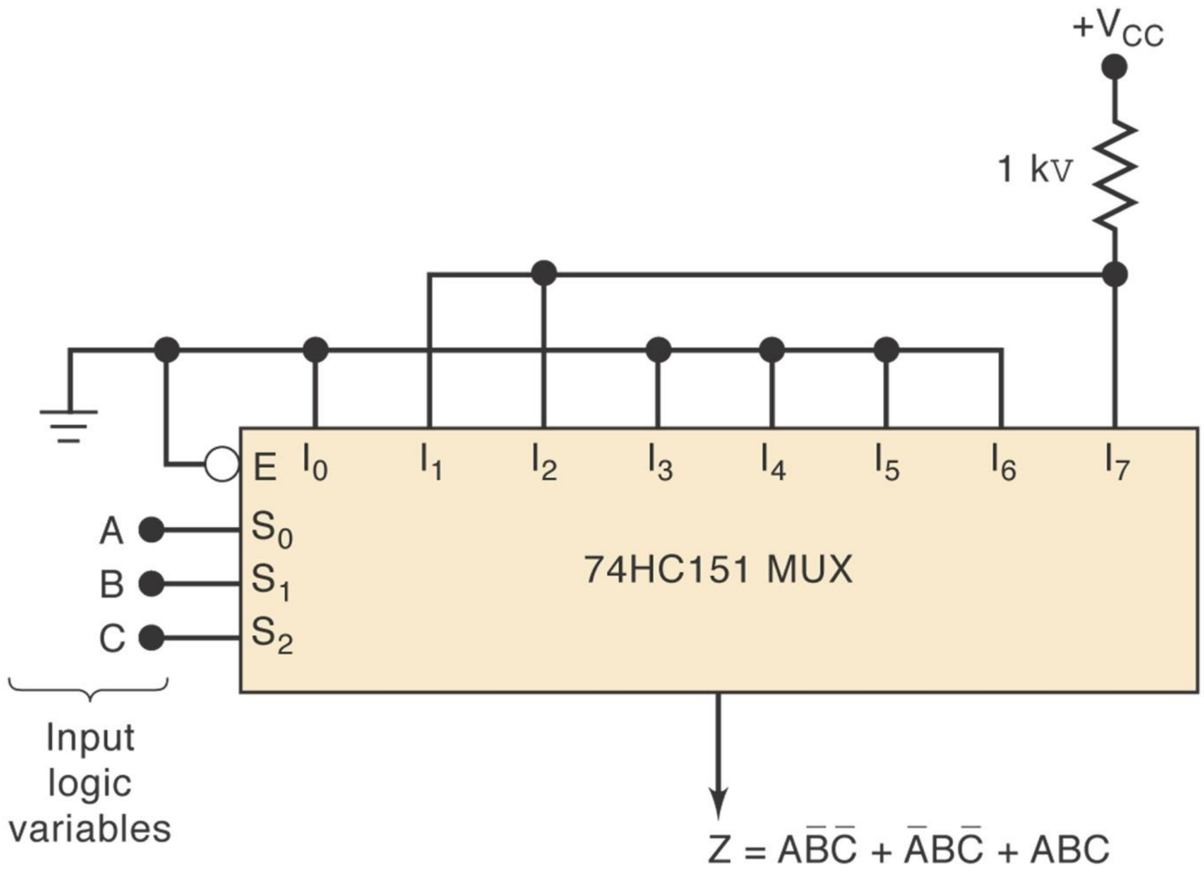


We can also construct our logic circuit using multiplexers

Recall from earlier: A multiplexer (MUX) selects one of multiple input signals and passes it to the output.



<u>Example:</u> Multiplexer used to implement a logic function described by the truth table.



(a)

C	B	A	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(b)

Exercises:

Design a Mod 6 **asynchronous** counter using JK FFs

If the input frequency is 240 kHz, what is the output frequency?

Repeat using a **synchronous** design. Use JK Flipflops.

Implement the circuit using logic gates and also MUXes.