Ch5: Digital Circuits – Part 2 (Sequential Logic)

Contents:

SR Flip-Flop

D Flip-Flop

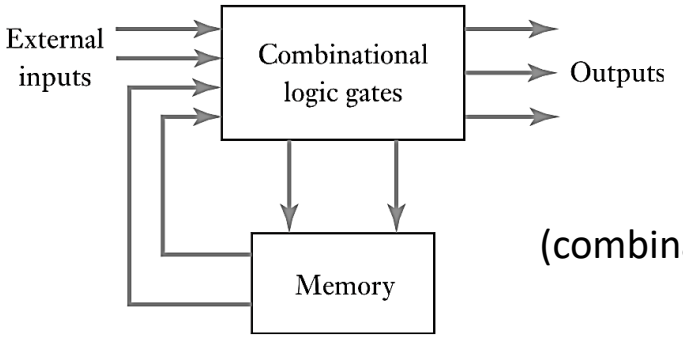
Digital Logic Families and ICs

Oscillators and Timers

SR Flip-Flop

Sequential Logic

When input data is applied to a combinational circuit, the output responds almost immediately, and they cannot store information. To provide "memory" to circuits we need **Sequential Logic** devices.



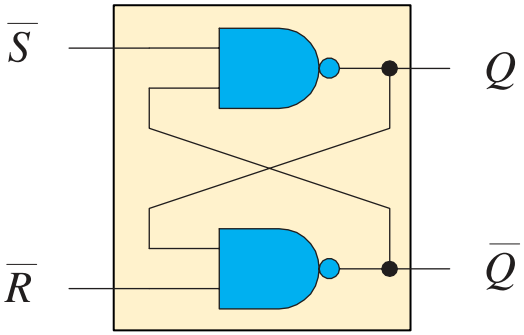
(combinational logic circuit + memory)

- Devices in this class include flip-flops, counters, monostables, latches, and microprocessors.
- These devices usually respond to inputs when a separate **trigger** signal (or **clock** (CK) signal) transitions from one level to another.
- The clock signal can be a periodic square wave or an aperiodic collection of pulses.

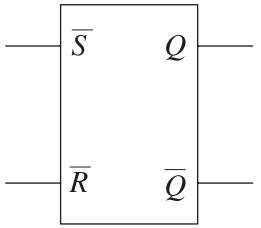
Set-Reset (SR) Flip-Flop

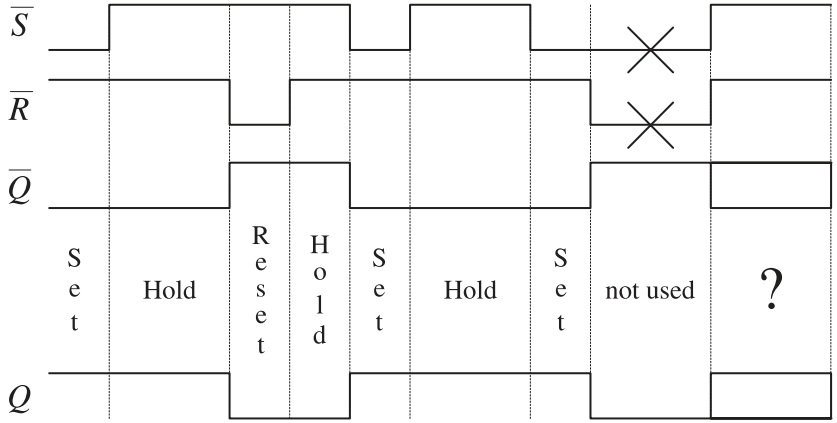
Set-Reset (SR) Flip-Flop is the most elementary data-storage circuit. There are two basic kinds of SR flip-flops: **Cross-NAND SR Flip-Flop** and **Cross-NOR SR Flip-Flop**.

Cross-NAND SR Flip-Flop:



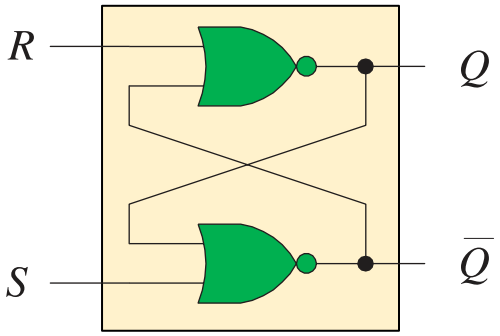
S	R	Q	$\overline{Q}$	condition
0	0	Q	$\overline{Q}$	Hold (no change)
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	not used (race)



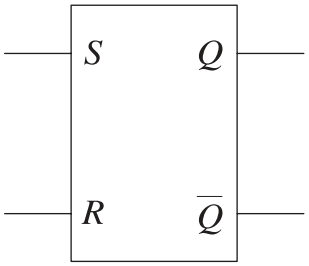


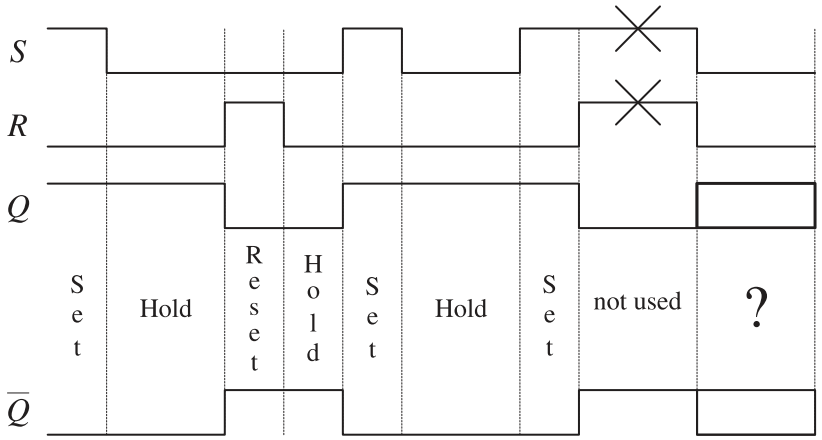
Set-Reset (SR) Flip-Flop

Cross-NOR SR Flip-Flop:



S	R	Q	$\overline{Q}$	condition
0	0	Q	$\overline{Q}$	Hold (no change)
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	not used (race)



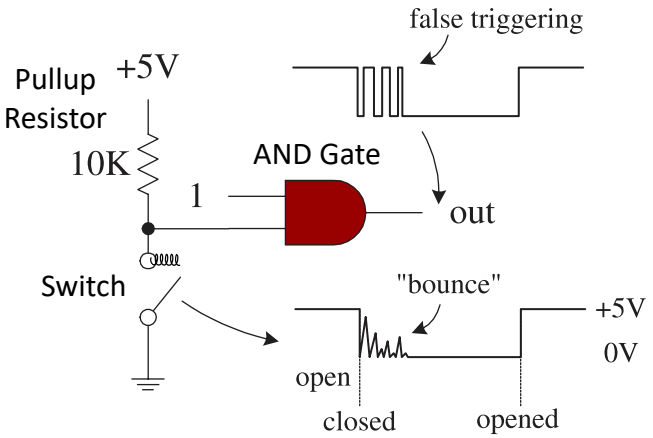


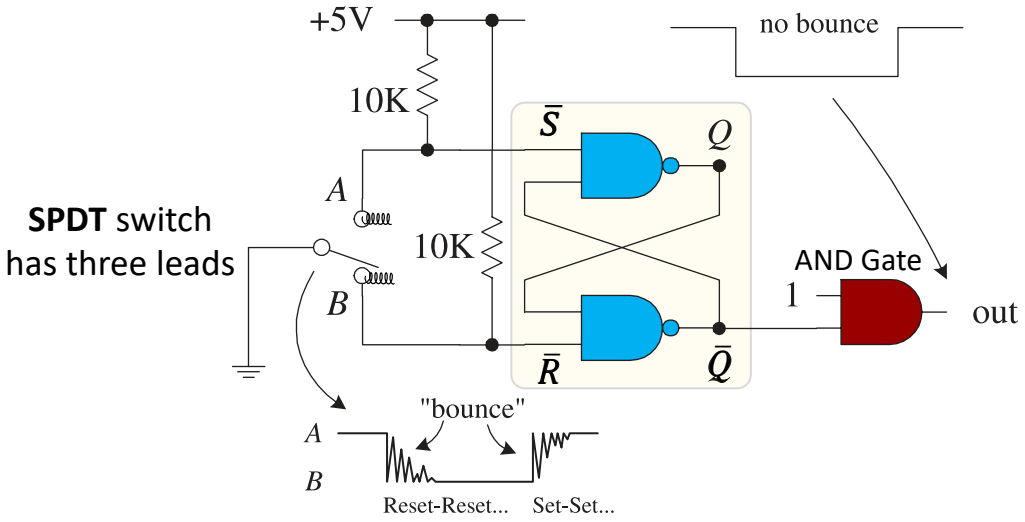
- Flip-Flop has the capability to **remain** in a particular output state (i.e., storing a bit) until input signals cause it to change state. This is the basis of all semiconductor information storage (e.g., computer RAM) and processing in digital computers.

Application: Switch Debouncer

Switch Bounce: When a switch is closed/opened, the metal contacts bounce several times before coming to rest due to inherent **springlike characteristics of the contacts** or **electrical arcing** that causes voltage oscillations. Though the bouncing typically lasts no more than 50 ms, the results can lead to unwanted false triggering. Each of the contacts during this bouncing time can register as a separate contact.

- A simple hardware solution for debouncing an SPDT switch is using an SR flip-flop to store the initial switch contact voltage while ignoring all trailing bounces.

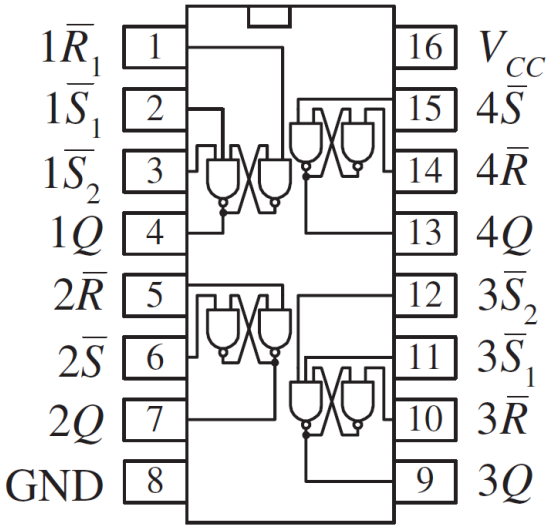




Application: Switch Debouncer

The 74LS279A is commonly used in switch debouncers that contains four independent SR latches.

74LS279A Quad SR Flip-Flop



- This method cannot be used to debounce signals from an SPST switch, which only has two leads (e.g., pushbutton); instead, a **Schmitt trigger** is used.

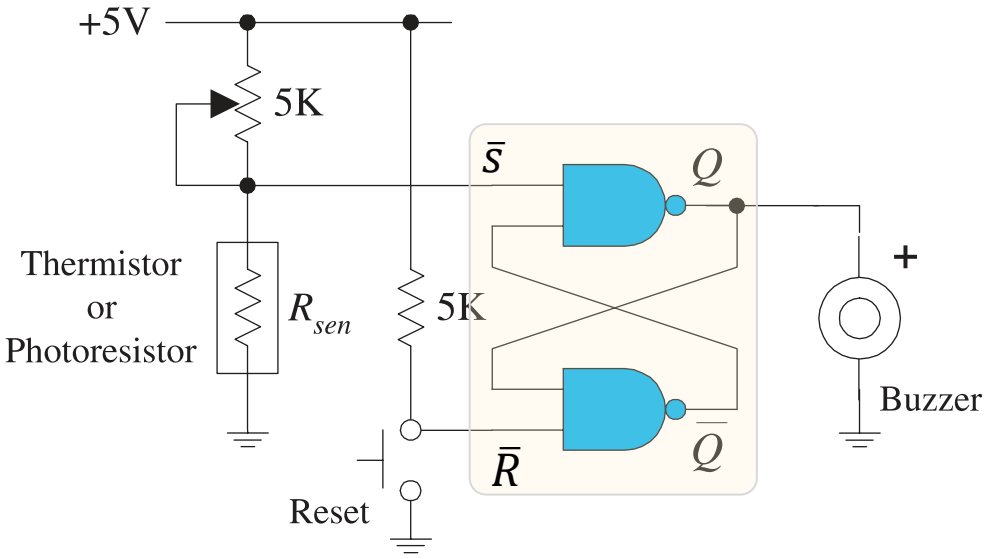
- Always remember that switch debouncing with **hardware** logic costs money to make, but **software** switch debouncing for a switch connected to a microcontroller is free. It's just an extra line or two of program code.

Application: Temperature or Light Alarm

In this simple circuit, an SR flip-flop is used to sound a buzzer alarm when the temperature or the light intensity reaches a critical level.

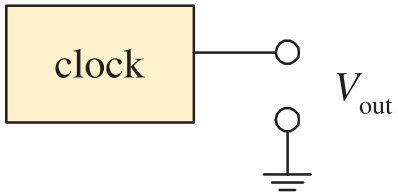
- When the temperature or light increases, the resistance of the thermistor or photoresistor decreases, and the $\bar{S}$ input voltage goes down.
- When the $\bar{S}$ input voltage goes below the high threshold level of the NAND gate, the flip-flop is set, and the alarm is sounded.

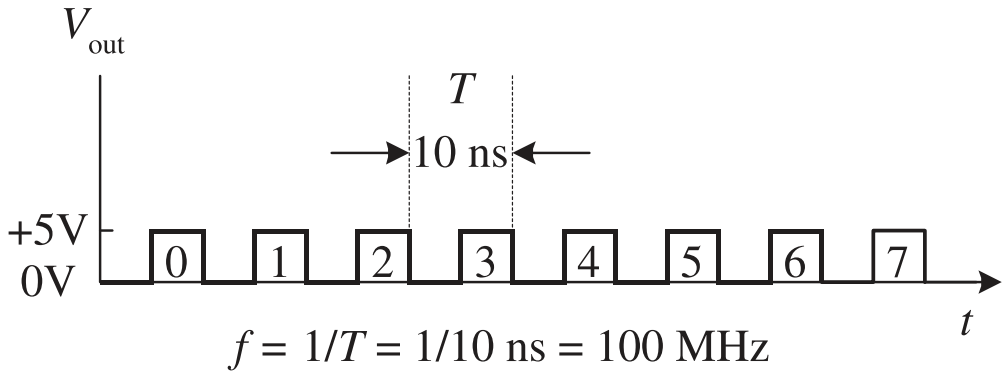
- The alarm will continue to sound until the RESET switch is pressed and the temperature or light level has gone below the critical triggering level.
- The pot is used to adjust this level.



Clock Timing

Most digital circuits require **precise timing to function properly**. Usually, a clock circuit that generates a series of high and low pulses at a fixed frequency is used as a **reference** on which to base all critical actions executed within a system.
The clock is also used to push bits of data through the digital circuitry.

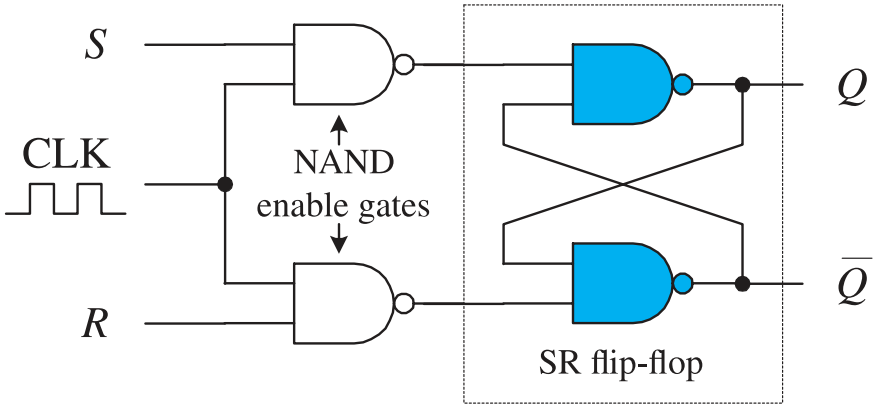


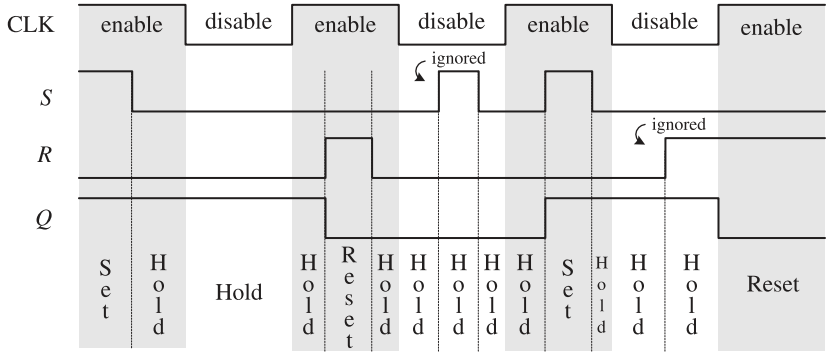


Level-Triggered SR Flip-Flop

When the S and R inputs of an SR flip-flop is either enabled or disabled by a **clock**, it is called **Level-Triggered**, **Synchronous**, or **Clocked SR Flip-Flop**.

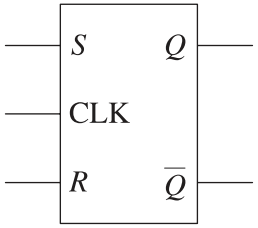
Level-triggered NAND SR flip-flop:





Only when the clock is high, S and R inputs are enabled. When the clock is low, the inputs are disabled, and the flip-flop is placed in hold mode.

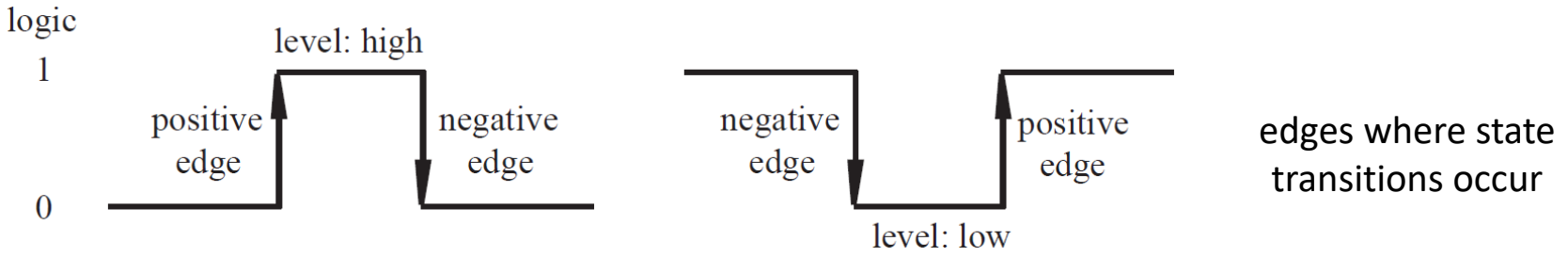
CLK	S	R	Q	$\overline{Q}$	Mode	
0	0	0	Q	$\overline{Q}$	hold	
0	0	1	Q	$\overline{Q}$	hold	SR-inputs
0	1	0	Q	$\overline{Q}$	hold	disabled
0	1	1	Q	$\overline{Q}$	hold	
1	0	0	Q	$\overline{Q}$	hold	
1	0	1	0	1	RESET	SR-inputs
1	1	0	1	0	SET	enabled
1	1	1	1	1	indeterminate	



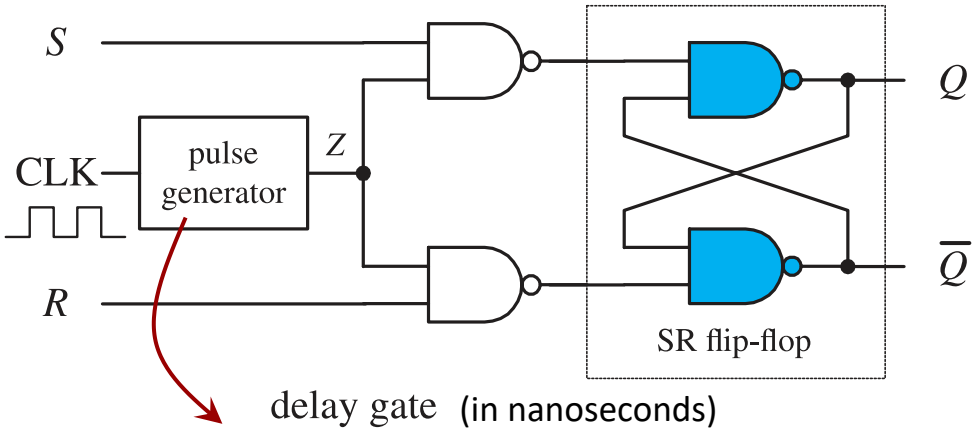
Edge-Triggered SR Flip-Flops

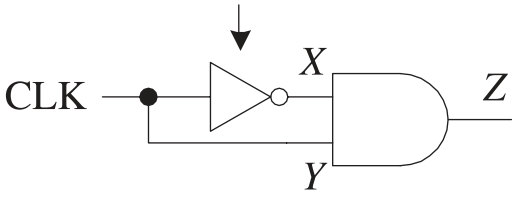
The level-triggered SR flip-flop samples S and R inputs during the entire time that the clock signal is enabling the flip-flop. An **Edge-Triggered Flip-Flop** samples the inputs only during either a positive clock edge (↑) or negative clock edge (↓). Any changes that occur before or after the clock edge are ignored and the flip-flop will be placed in hold mode.

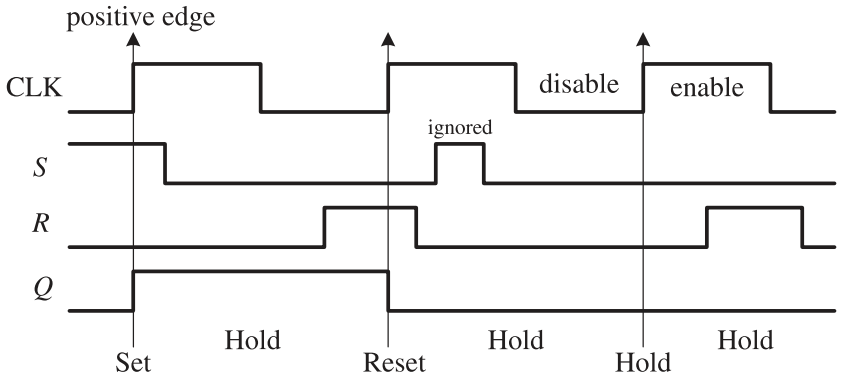
- **Positive Edge-Triggered** devices respond to a low-to-high (0 to 1) transition.
- **Negative Edge-Triggered** devices respond to a high-to-low (1 to 0) transition.



Positive Edge-Triggered

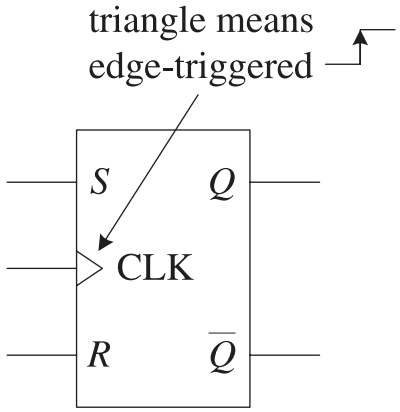




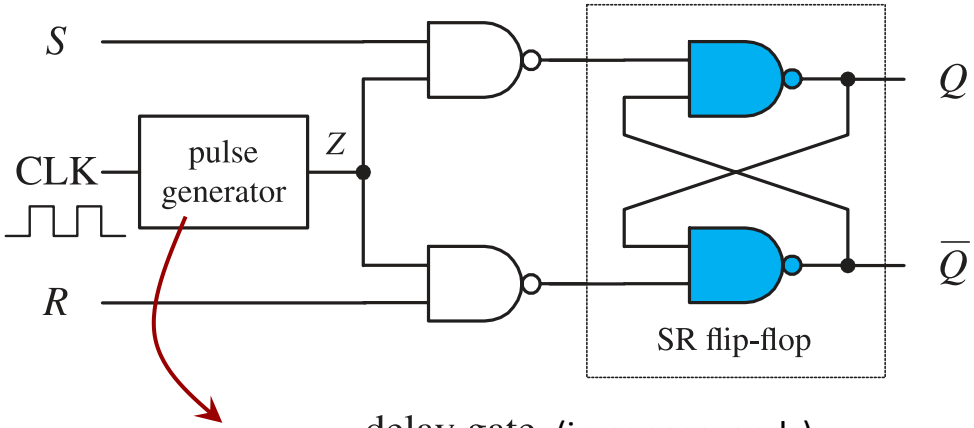


CLK	S	R	Q	$\overline{Q}$	Mode
0	X	X	Q	$\overline{Q}$	hold
1	X	X	Q	$\overline{Q}$	hold
↓	X	X	Q	$\overline{Q}$	hold
↑	0	0	Q	$\overline{Q}$	hold
↑	0	1	0	1	RESET
↑	1	0	1	0	SET
↑	1	1	1	1	indeterm.

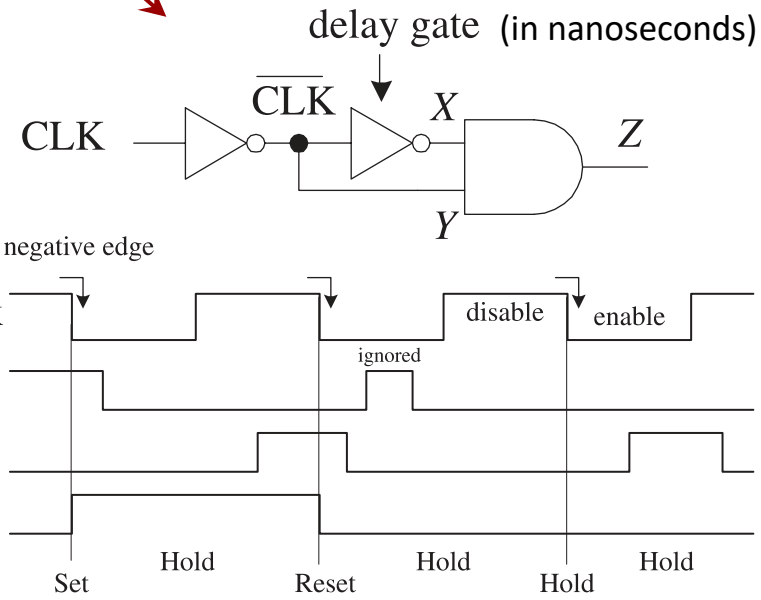
X = don't care

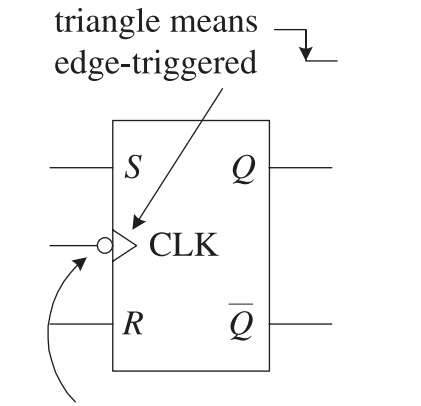


Negative Edge-Triggered



CLK	S	R	Q	$\overline{Q}$	Mode
0	X	X	Q	$\overline{Q}$	hold
1	X	X	Q	$\overline{Q}$	hold
↑	X	X	Q	$\overline{Q}$	hold
↓	0	0	Q	$\overline{Q}$	hold
↓	0	1	0	1	RESET
↓	1	0	1	0	SET
↓	1	1	1	1	indeterm.

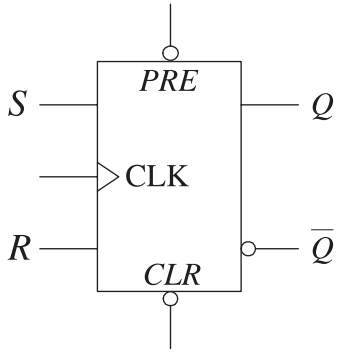




bubble next to triangle means
negative edge-triggered input

Asynchronous Inputs Preset & Clear

All types of flip-flops may have two additional inputs **Preset** (PRE) and **Clear** (CLR), which are called **Asynchronous Inputs**. Unlike the synchronous inputs (S and R), the asynchronous inputs **instantaneously override any other inputs**, i.e., S and R.



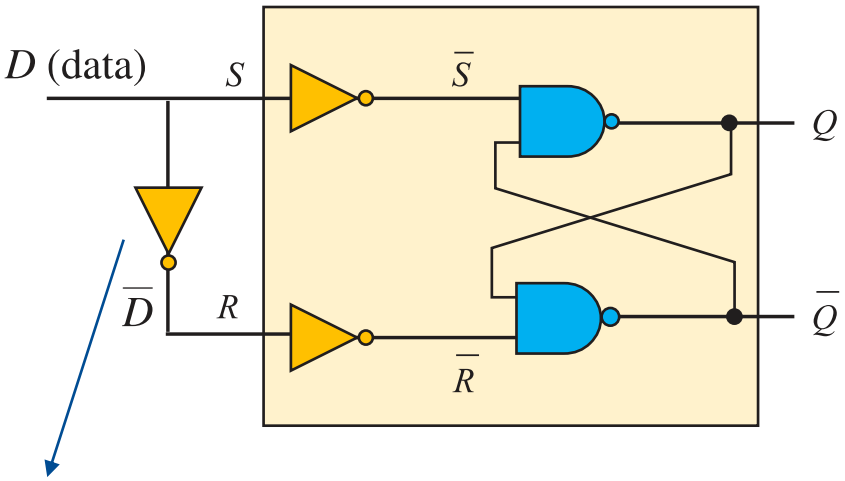
$\overline{PRE}$	$\overline{CLR}$	CLK	S	R	Mode
0	1	X	X	X	Preset
1	0	X	X	X	Clear
0	0	X	X	X	not used (race)

- Either of these inputs can be used to define the state of a flip-flop **after power-up**; otherwise, **at power-up** the output of a flip-flop is **uncertain**.
- The Preset input is used to set or initialize the output Q of the flip-flop to high (1).
- The Clear input is used to clear or reset the output Q of the flip-flop to low (0).
- The small inversion symbols imply that the function is asserted when the asynchronous input signal is low (active-low input).
- Both preset and clear should not be asserted simultaneously.

D Flip-Flop

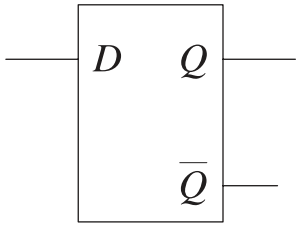
Level-Triggered D Flip-Flop or D Latch

A **D (Data) Flip-Flop** is a single input device that is basically an SR flip-flop, where S is replaced with D and R is replaced with $\overline{D}$.

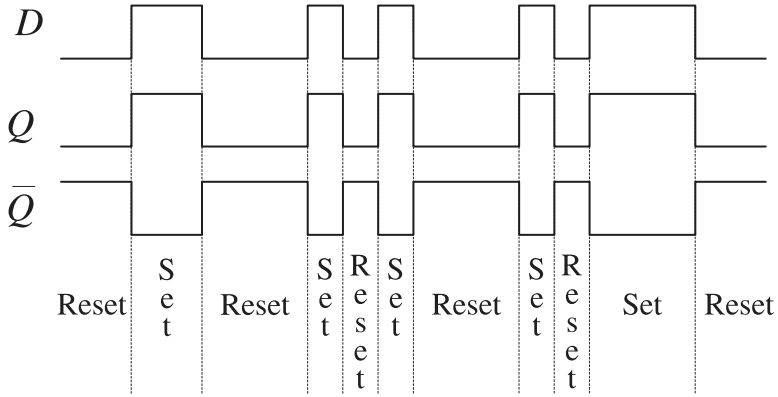


The inverter eliminates the hold condition!

logic symbol

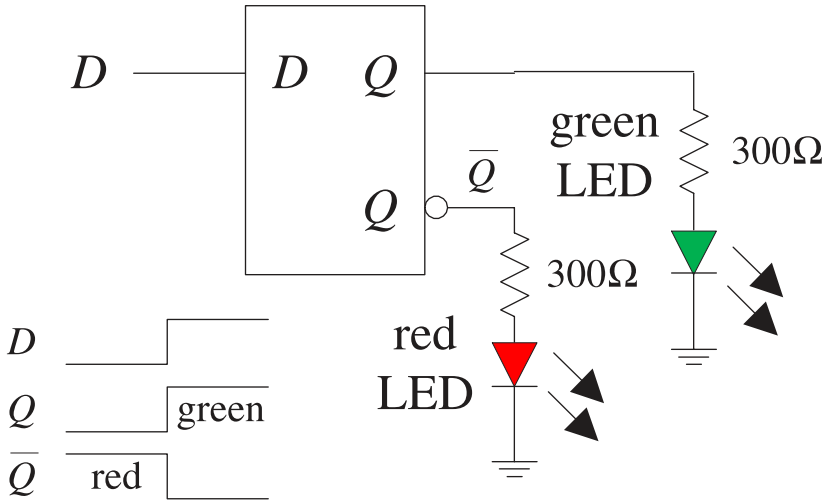


D	Q	$\overline{Q}$	Mode
0	0	1	Reset
1	1	0	Set

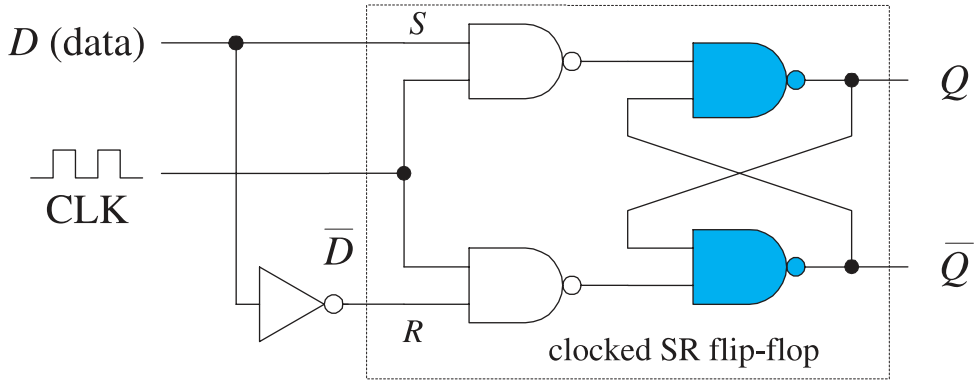


Example: Stop-Go Indicator

A simple level-triggered D flip-flop is used to turn on a red LED when its D input is low (reset) and turn on a green LED when the D input is high (set). Only one LED can be turned on at a time.

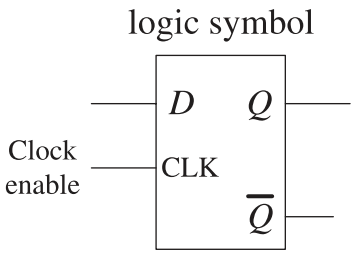


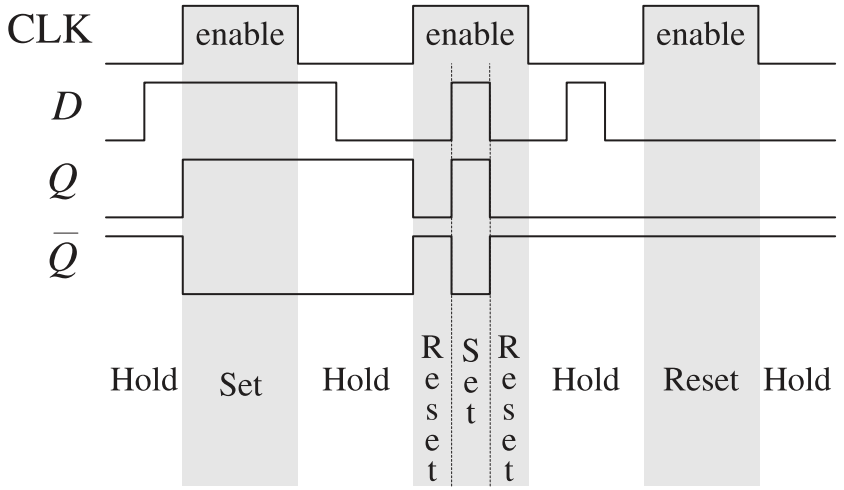
Clocked or Level-Triggered D Flip-Flop



CLK	D	Q	$\overline{Q}$	Mode
0	X	Q	$\overline{Q}$	Hold
1	0	0	1	Reset
1	1	1	0	Set

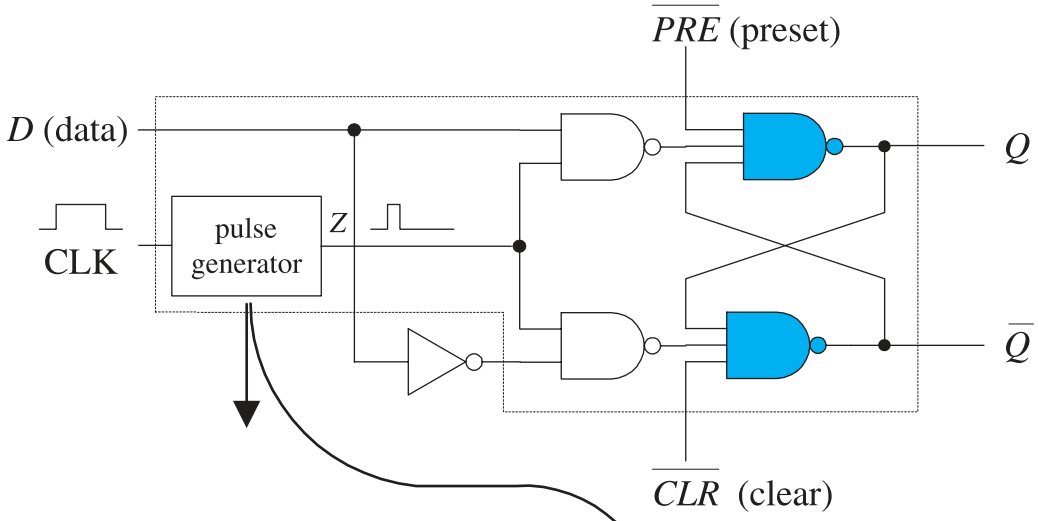
X: don't care

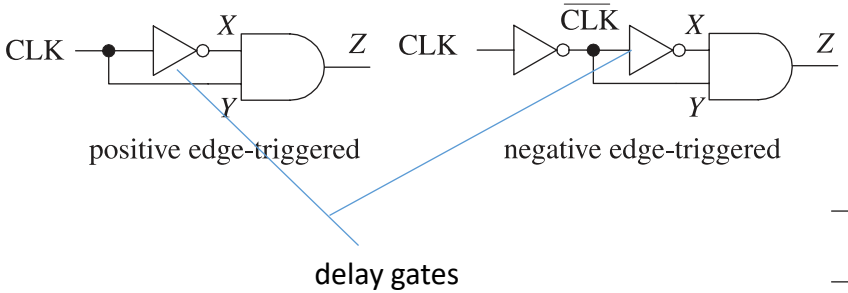




7475 is a popular level-triggered D flip-flop IC.

Edge-Triggered D Flip-Flop

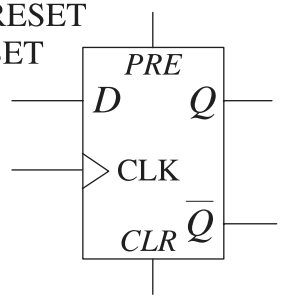




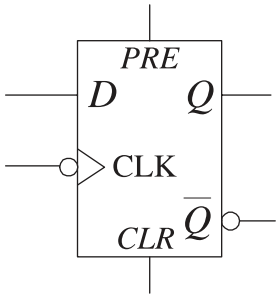
Positive edge-triggered:

$\overline{PRE}$	$\overline{CLR}$	CLK	D	Q	Mode
0	1	X	X	1	Preset
1	0	X	X	0	Clear
0	0	X	X	Q	not used (race)
1	1	0, 1, ↓	X	Q	Hold
1	1	↑	0	0	RESET
1	1	↑	1	1	SET

X: don't care



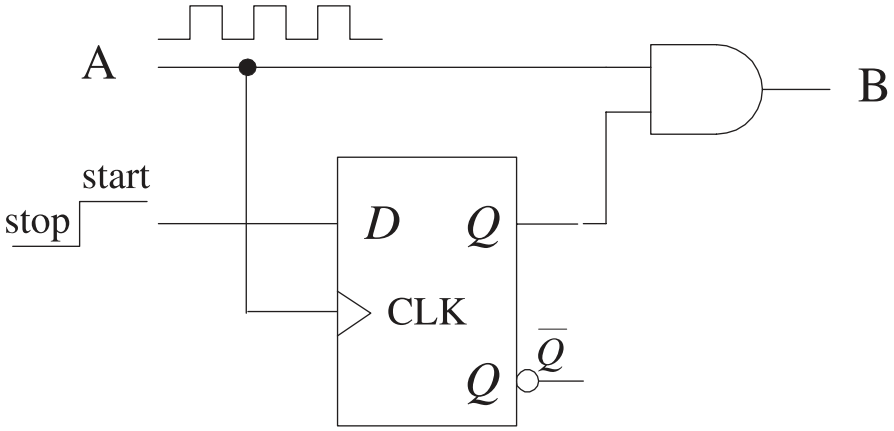
Negative edge-triggered:

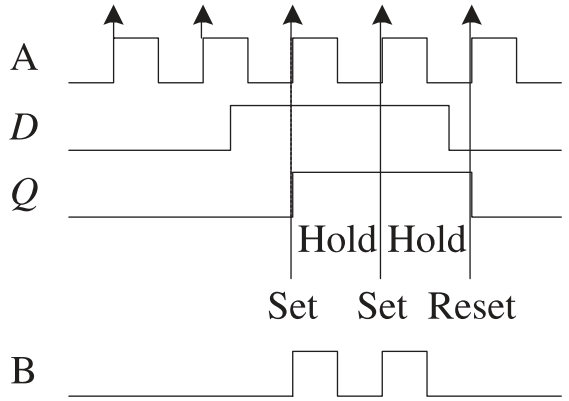


$\overline{PRE}$	$\overline{CLR}$	CLK	D	Q	Mode
0	1	X	X	1	Preset
1	0	X	X	0	Clear
0	0	X	X	Q	not used (race)
1	1	0, 1, ↑	X	Q	Hold
1	1	↓	0	0	RESET
1	1	↓	1	1	SET

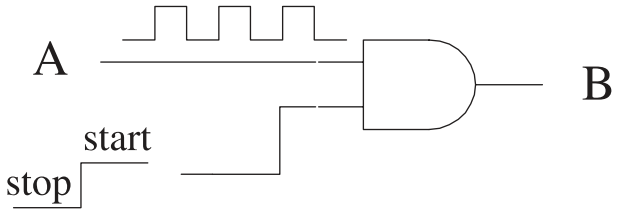
7474 is a popular edge-triggered D flip-flop IC.

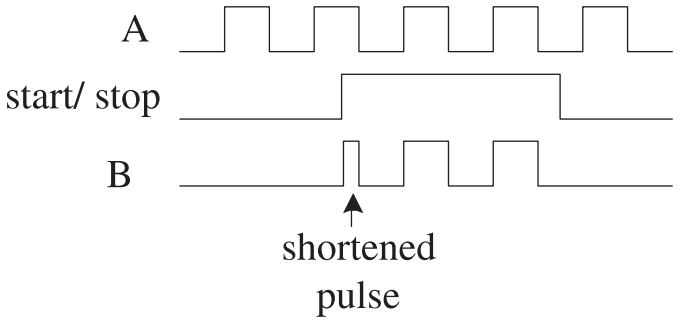
Application: Synchronizer





What would happen if D flip-flop wasn't in place



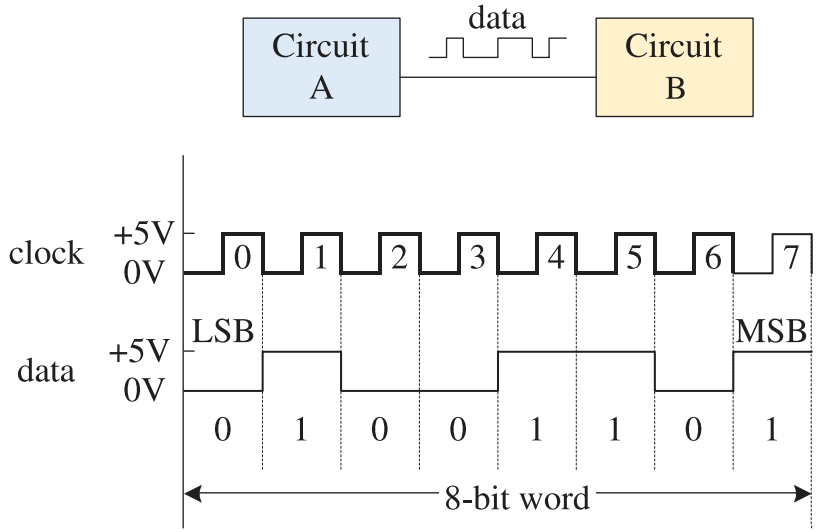


Serial vs. Parallel Representation

Binary information can be transmitted from one location to another in either a **Serial** or **Parallel** manner.

Serial transmission uses a **single** electrical conductor (and a common ground) for data transfer. **Each bit** from the binary number **occupies a separate clock period**, with the change from one bit to another occurring at each falling or leading clock edge; the type of edge depends on the circuitry used.

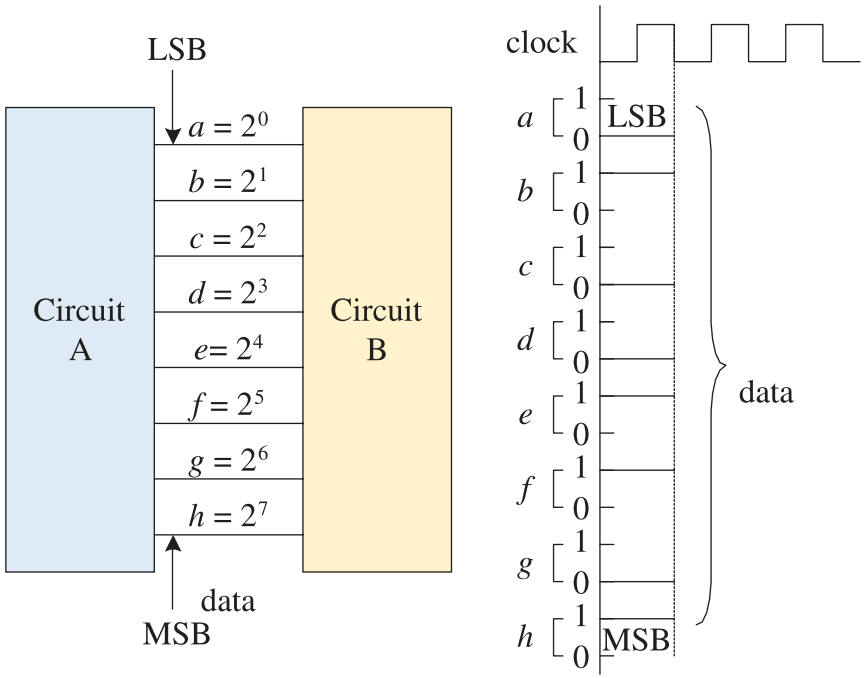
Serial transmission of an 8-bit word (10110010):



Serial vs. Parallel Representation

Parallel transmission uses separate electrical conductors for each bit (and a common ground). Unlike serial transmission, the **entire word is transmitted simultaneously in only one clock cycle**, not eight clock cycles. In other words, it is **eight times faster**.

Parallel transmission of an 8-bit word (10110010):

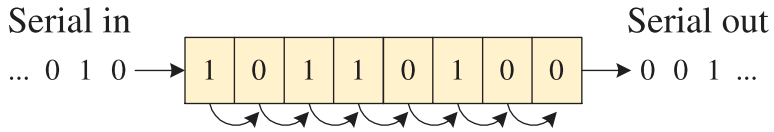


Shift Registers

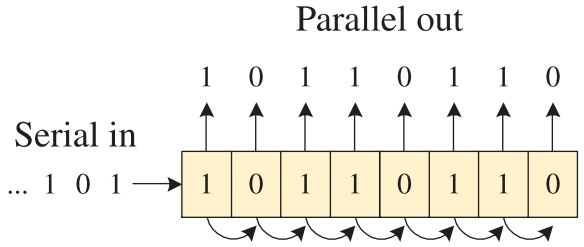
Data words traveling through a digital system frequently must be temporarily held, copied, and bit-shifted to the left or to the right. A device that can be used for such applications is the **Shift Register**.

A shift register is constructed from a **row of flip-flops** connected so that digital data can be shifted down the row either in a left or right direction.

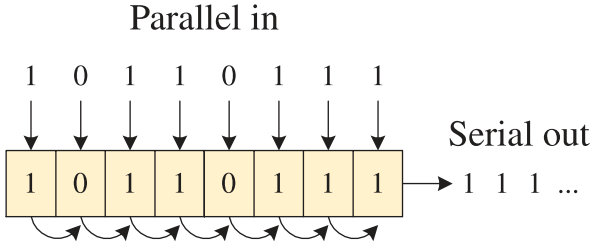
- **Serial-in/Serial-out**:



- **Serial-in/Parallel-out**:

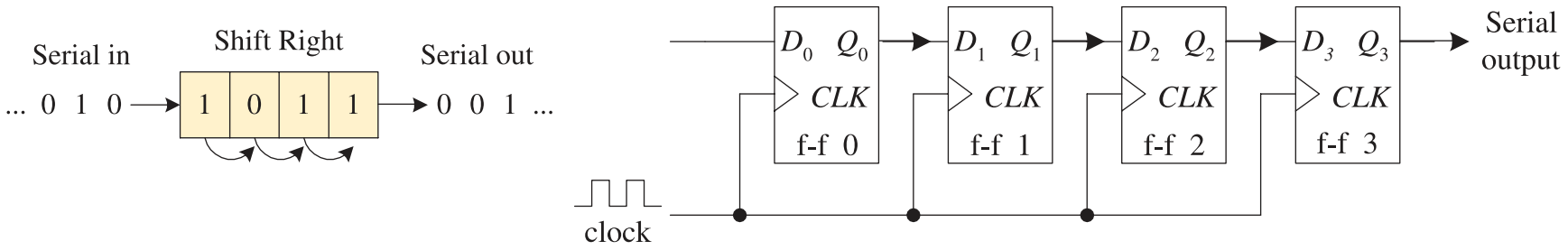


- **Parallel-in/Serial-out**:



Serial-In/Serial-Out Shift Register

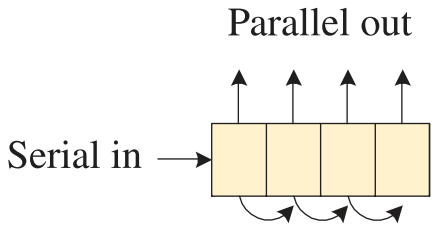
A simple 4-bit serial-in/serial-out shift register made from **D flipflops**:

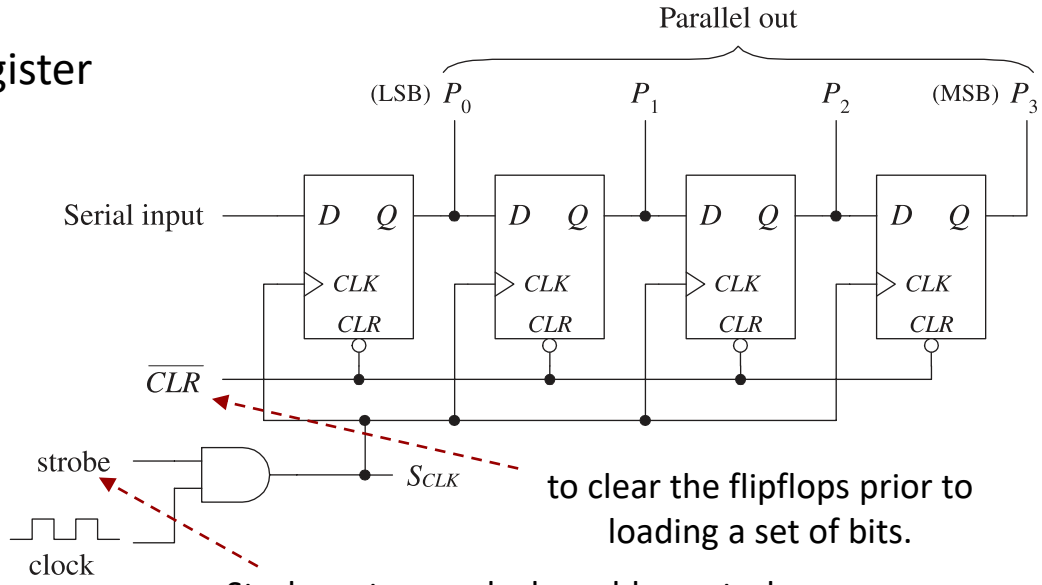


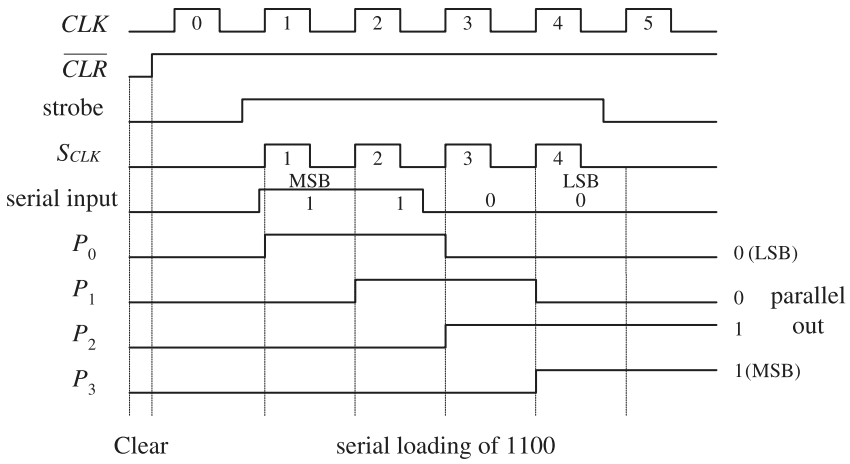
- Serial data is applied to the D input of flip-flop 0. When the clock line receives a positive clock edge, the serial data is shifted to the right from flip-flop 0 to flip-flop 1. Whatever bits of data were present at flip-flop 2's and 3's outputs are shifted to the right during the same clock pulse.
- To store a **4-bit word** into this register requires **four clock pulses**.

Serial-In/Parallel-Out Shift Register

A 4-bit serial-in/parallel-out shift register constructed from D flip-flops:



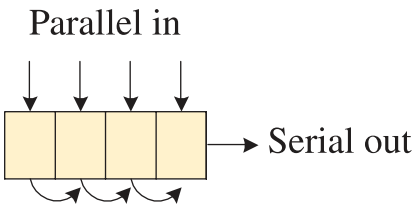


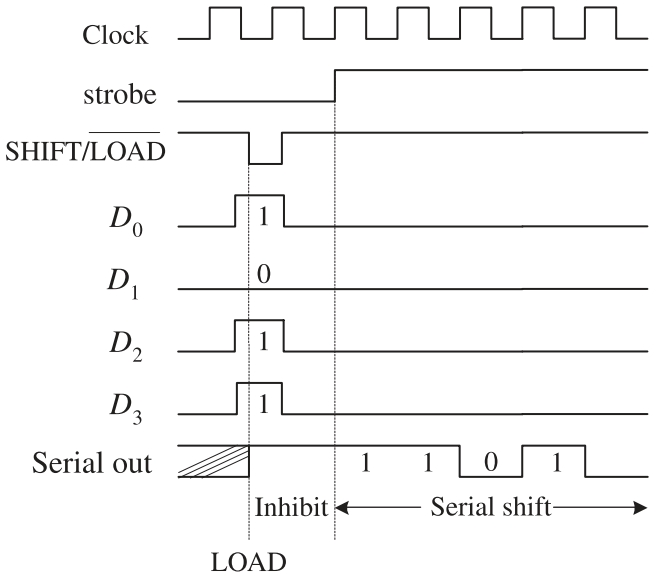


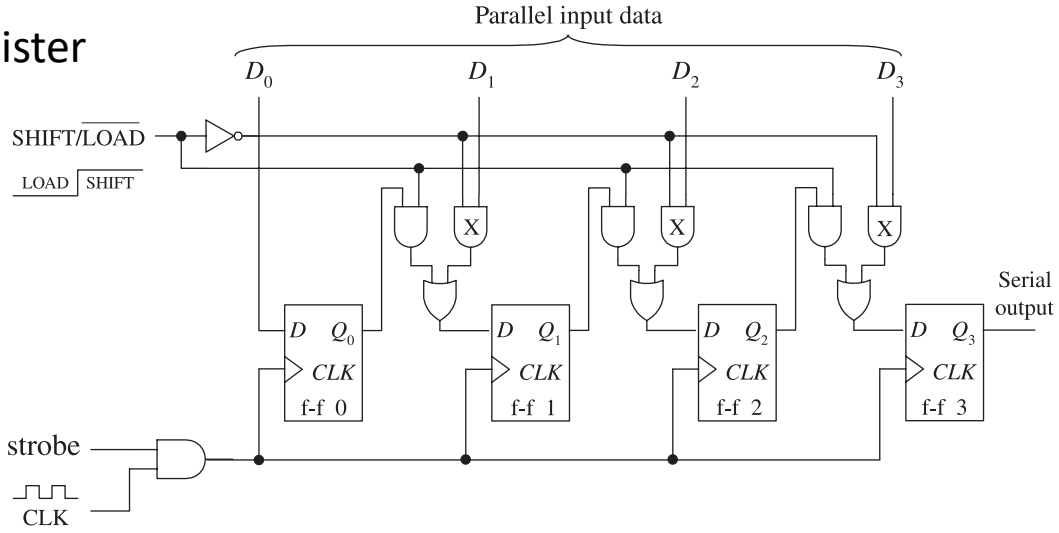
- It is common to use a serial to parallel shift register IC (e.g., 74595 or 74164) with a microcontroller to provide it with more outputs.
- The output is synchronized by a clock signal.
- The shift register ICs can be cascaded for a larger number of bits.

Parallel-In/Serial-Out Shift Register

A 4-bit parallel-in/serial-out shift register constructed from D flip-flops:







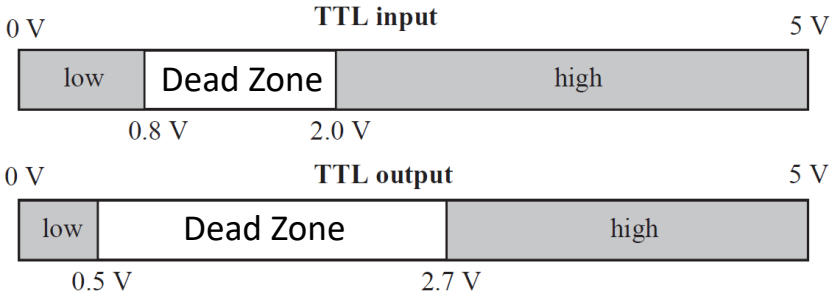
- Sometimes, a serial to parallel shift register IC (e.g., 74165) is used with a microcontroller to provide it with more inputs.
- The output is synchronized by a clock signal.
- The shift register ICs can be cascaded for a larger number of bits.

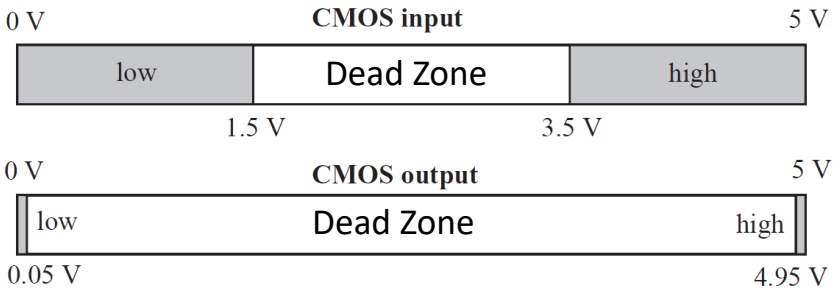
Digital Logic Families and ICs

Logic Families and Integrated Circuits

There are two families of logic devices called **TTL** and **CMOS**. **TTL** stands for **Transistor-Transistor Logic** devices and **CMOS** for **Complementary Metal-Oxide Semiconductor** devices. **BJTs** are the building blocks for **TTL** logic, and **MOSFETs** are the building blocks for **CMOS** logic. **TTL and CMOS input and output levels:**

Logic 0 ≡ Low, Logic 1 ≡ High





When interfacing and mixing different types of digital devices, both **voltage levels** and **input/output current capabilities** of the devices are important.

These capabilities are the amount of current a device can **source** (produce) when the output is **high** [I_{OH}: "high-level output current"] and the amount of current the device can **sink** (draw) when the output voltage is **low** [I_{OL}: "low-level output current"].

TTL vs. CMOS

❖ Advantages of CMOS:

- Because the MOSFET gates are insulated, CMOS devices consume power only when switching between logic states or when there is a load attached. Therefore, CMOS is **useful in battery-operated applications** where power is limited (TTL devices dissipates power continuously regardless of whether the output is high or low).
- The **wide power supply range** of CMOS (3–18 V) provides more design flexibility and allows use of less tightly regulated power supplies (TTL devices are powered by a 5 V DC supply).

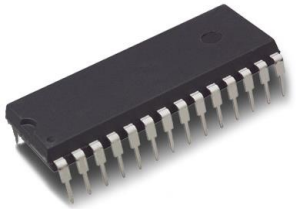
❖ Disadvantages of CMOS:

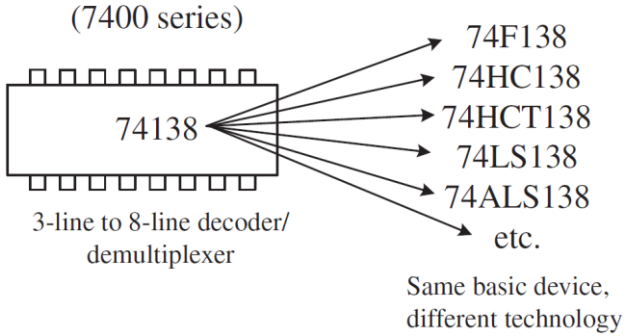
- CMOS is **sensitive to static discharge** even with internal protective diodes. Protective packaging and static discharge during handling and assembly are necessary; otherwise, the devices are easily damaged.
- CMOS requires negligible input current, but its output current is also small compared to TTL. This **limits the ability of CMOS to drive large TTL fan-out** or other high current devices.
- CMOS is generally about **5 to 10 times slower** than TTL. Typically, TTL have delay times of 2 to 40 ns

Digital Logic ICs

TTL ICs are available in the form **AAxxyzz**:

- **AA**: Manufacturer's prefix (SN for TI and others; DM for National Semiconductor).
- **xx**: 54 for military quality, 74 for industrial quality.
- **y**: Different internal designs (no letter: standard TTL; L: low-power dissipation; H: high-power dissipation; S: Schottky type; AS: advanced Schottky, LS: low-power Schottky; ALS: advanced low-power Schottky, …).
- **zz**: two- or three-digit Device number.





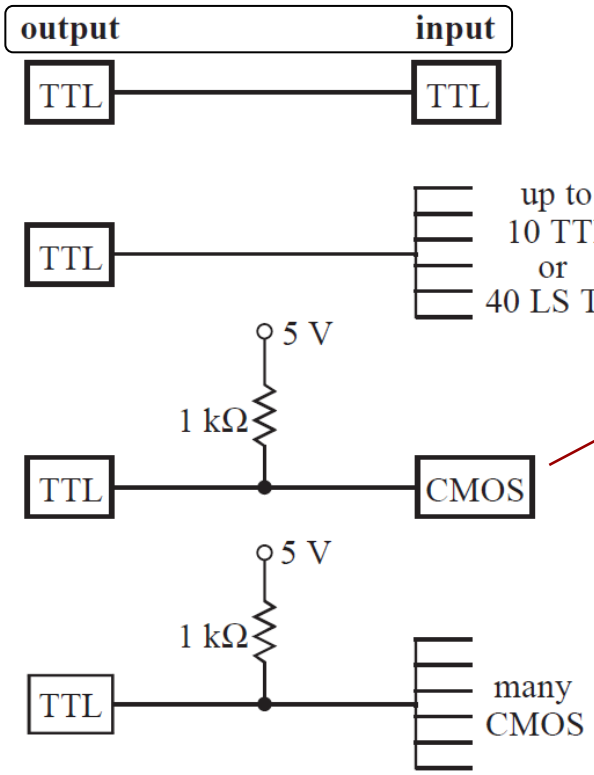
CMOS ICs are available in the 40XXB series and the 74CXX series. There are also different varieties of the 74CXX family that provide different speed and power characteristics.

pin-compatible with the TTL family

- 74**HC**XX (high-speed CMOS) [very popular today]
- 74**AC**XX (advanced CMOS)
- 74**HCT**XX and 74**ACT**XX (high-speed CMOS with TTL threshold)

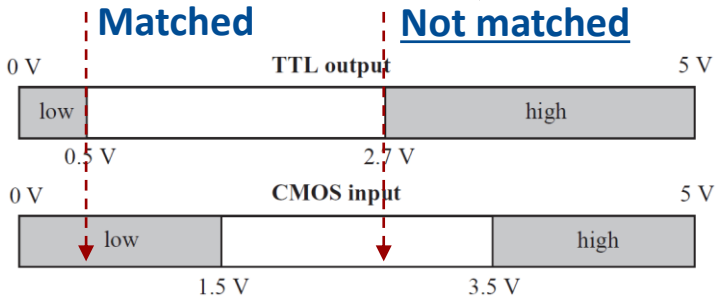
Interfacing TTL and CMOS Devices

When designing digital systems, it is recommended to use only one family of devices (TTL or CMOS), but there may be situations where we need to connect devices from the same and/or different families.

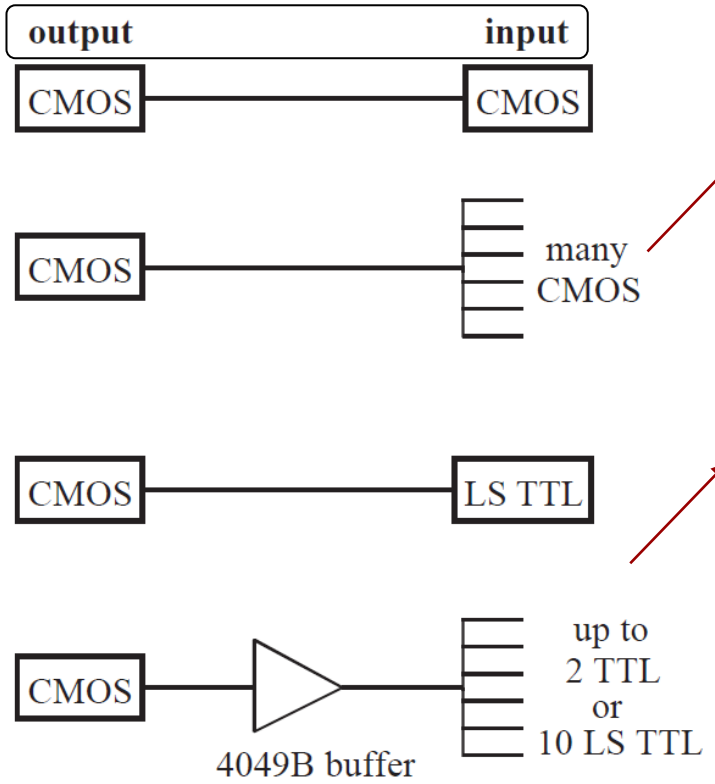


- The output of a TTL device sinks current when it is low and sources current when it is high. TTL low sink current (I_{OL}) is the limiting factor

- Using a proper pull-up resistor on the TTL output will raise the output voltage above the 3.5 V required by a CMOS input.



Interfacing TTL and CMOS Devices



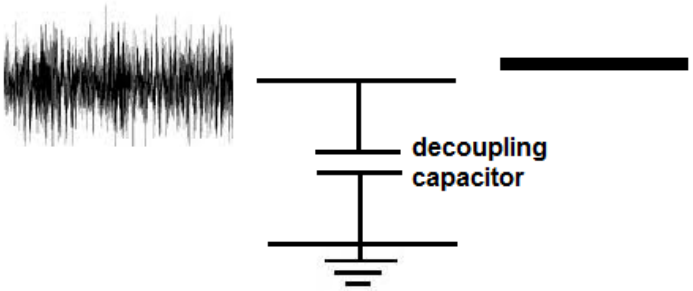
- A CMOS device neither sources nor sinks current at its input due to the insulating gate at the input.

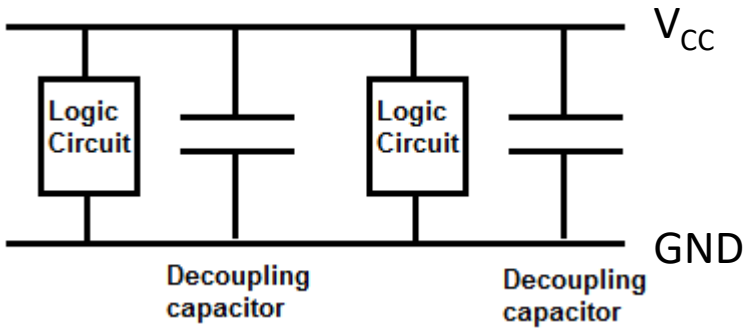
- If CMOS is powered by a 5 V supply voltage (V_{DD}) and is used to drive TTL devices, a CMOS high is no problem because the CMOS device can source enough current to drive the TTL input. However, a CMOS low can sink only enough current to drive one LS TTL input. A CMOS 4049 buffer can be used to provide adequate fan-out for up to two standard TTL inputs or approximately 10 LS TTL inputs.

A Standard Practice for Powering ICs

During the interval that a logic device makes a low-to-high or a high-to-low level transition, a drastic change in power supply current occurs, which results in a sharp, high-frequency **current spike** within the supply line. This unwanted spike can cause false triggering of other devices in the circuit and also can generate unwanted electromagnetic radiation.

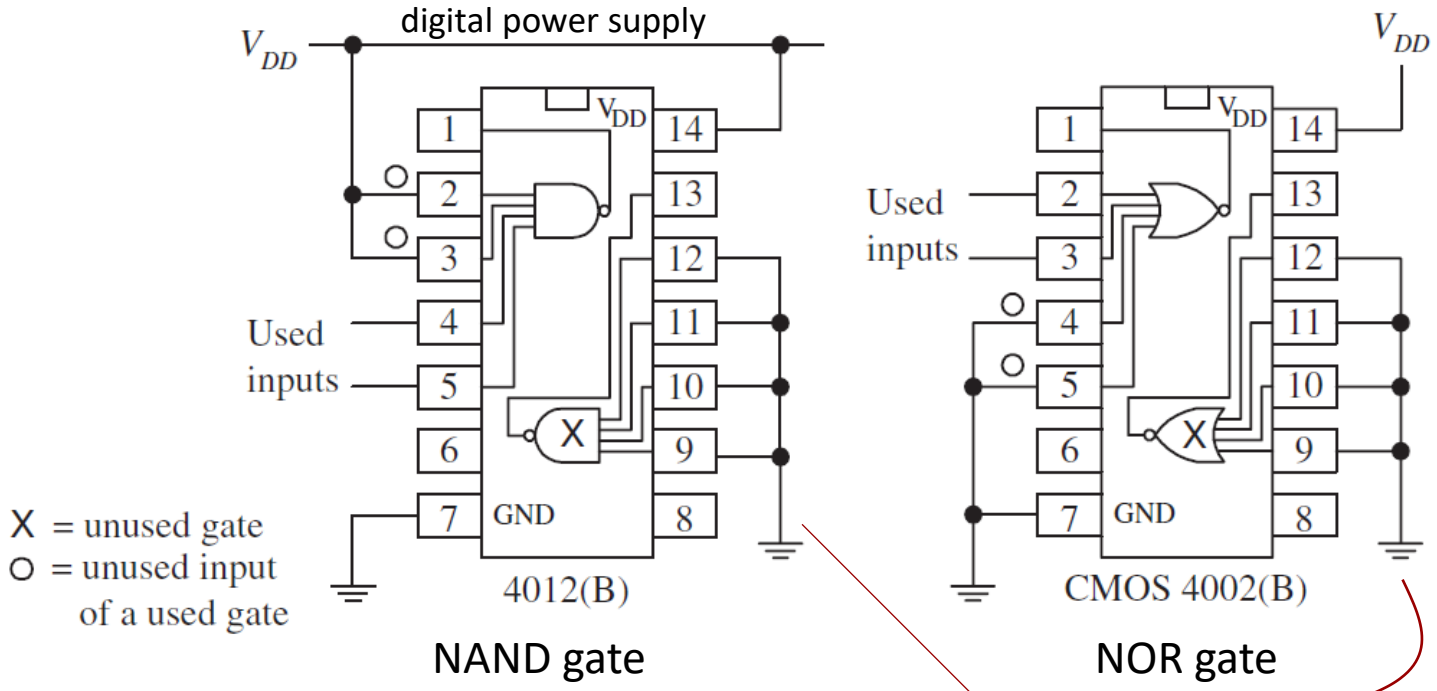
- To avoid unwanted spikes within logic systems, **decoupling capacitors**, typically ceramic capacitor from 0.01 to 0.1 µF (>5 V), can be used. Decoupling capacitors is placed directly across the V_{CC}-to-ground pins of each IC in the circuit to absorb the spikes and keep the V_{CC} level at each IC constant.
- Decoupling capacitors should be placed as close to the ICs as possible to keep current spikes local.





A Standard Practice for Unused Inputs

Unused inputs of ICs (e.g., CMOS IC) that affect the logical state of a chip should not be allowed to float. Floating inputs are liable to pick up external electrical noise, which leads to erratic output behavior. Instead, they should be tied high or low, as necessary.

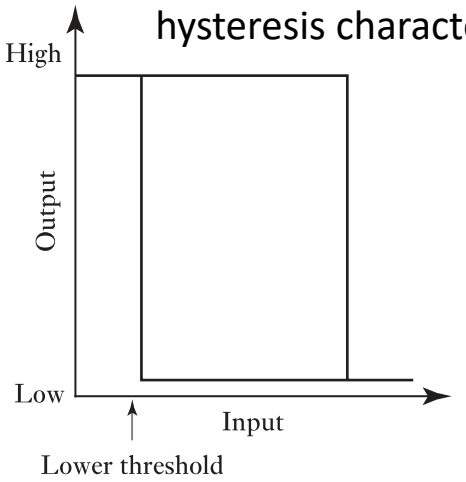


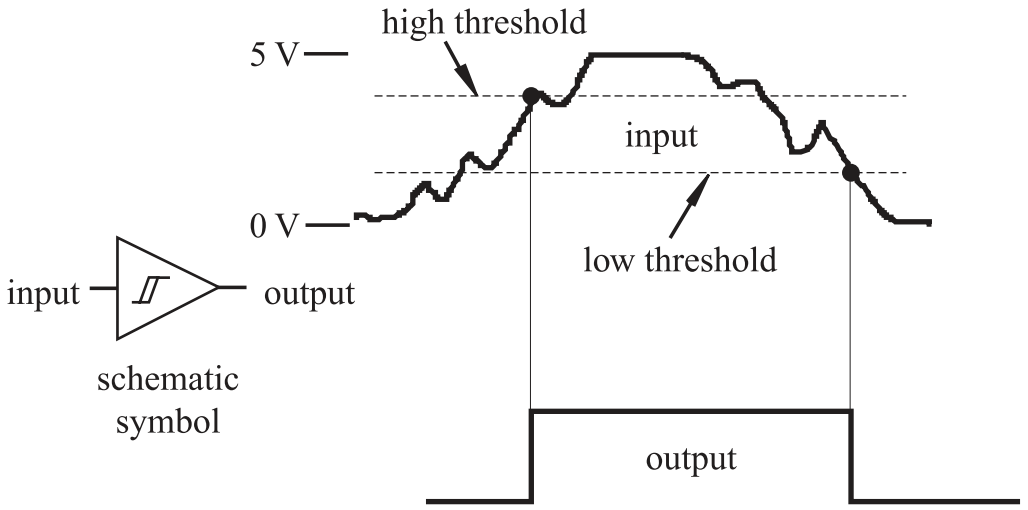
Inputs of unused CMOS gates should be grounded.

Oscillators and Timers

Schmitt Trigger

In some applications, digital pulses may not exhibit **sharp edges**; instead, the signal may ramp from 0 to 5V over a finite time period, and it may do so in a "noisy" (jumpy) fashion. **Schmitt Trigger** is a device that can convert such a slowly changing signal into a sharp pulse using the threshold hysteresis effect.



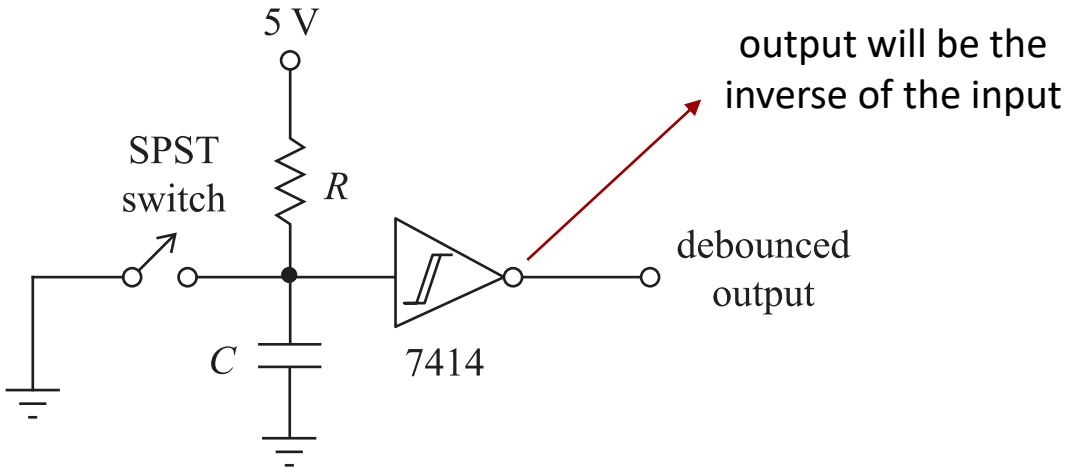


The output goes high when the input exceeds the high threshold and remains high until the input falls below the low threshold. The hysteresis between the low and high thresholds results in the distinct edges in the output.

Application: Switch Debouncer

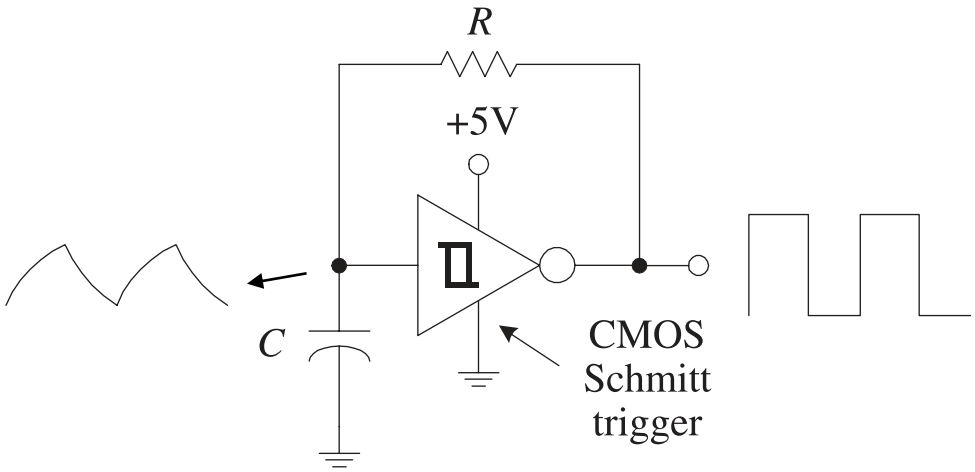
To debounce signals from an SPST switch, which only has two leads (e.g., pushbutton) Schmitt trigger is used.

(7414 is an IC with six Schmitt Trigger Inverters)



With the switch open, the capacitor is fully charged to 5 V, and the output of the 7414 is low. When the switch is closed, the capacitor is shorted to ground, and the 7414's output goes high. Depending on the capacitor value, the capacitor voltage might discharge below the low threshold of the 7414 on first contact. If not, the capacitor will eventually discharge as any bouncing occurs and continual contact is established. When the switch is opened, the capacitor again begins to charge. When the charge on the capacitor exceeds the high threshold of the 7414, the output again goes low. The result is a clean, bounce-free, output pulse.

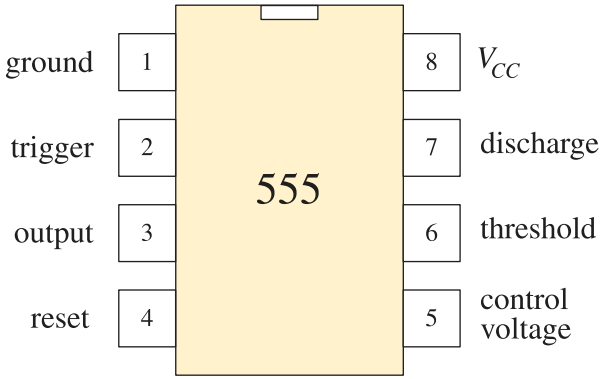
Application: Digital Oscillator

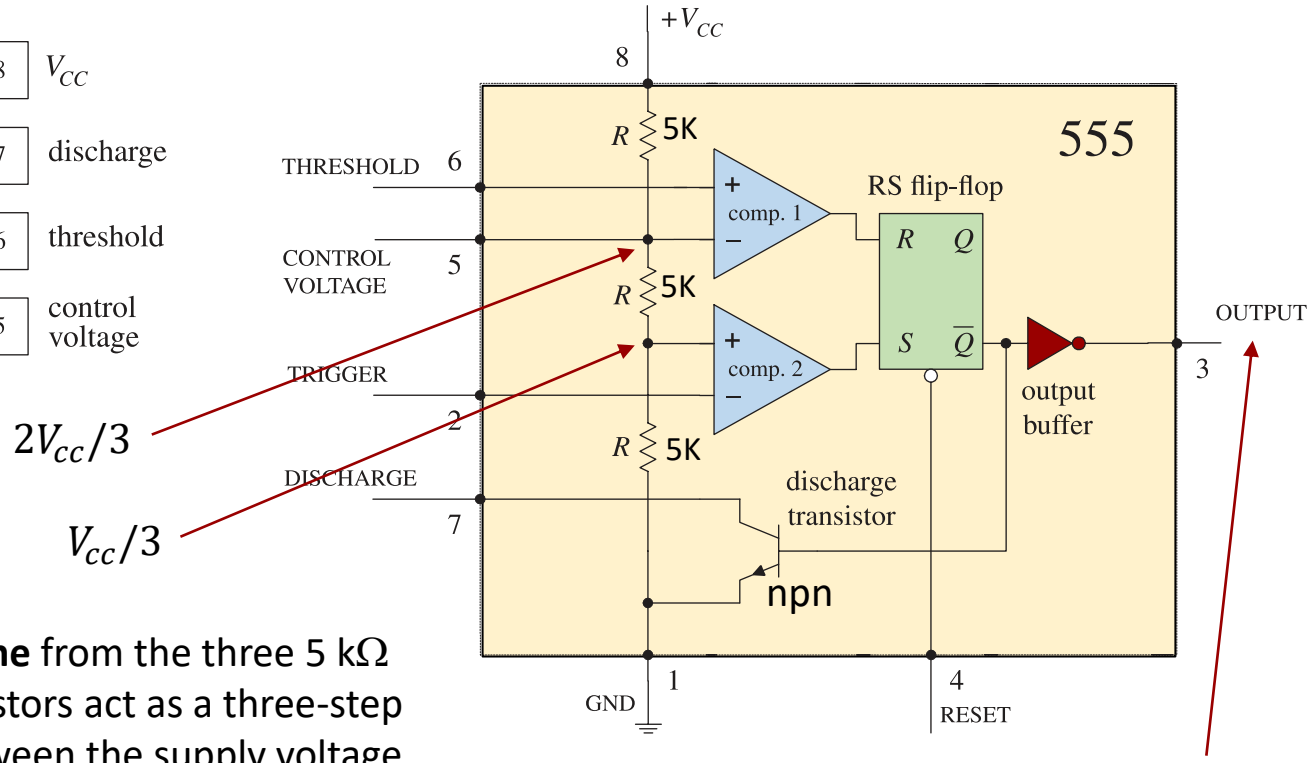


When power is first applied to the circuit, the voltage across C is zero, and the output of the inverter is high (+5 V). The capacitor starts charging up toward the output voltage via R. When the capacitor voltage reaches the positive- going threshold of the inverter (e.g., 1.7 V), the output of the inverter goes low (≈0 V). With the output low, C discharges toward 0 V. When the capacitor voltage drops below the negative- going threshold voltage of the inverter (e.g., 0.9 V), the output of the inverter goes high. The cycle repeats. The on/off times are determined by the positive- and negative- going **threshold voltages** of the Schmitt trigger and the $\boldsymbol{RC}$ **time constant**.

555 Timer

The 555 timer IC is known as the "time machine" that can act as either a **Timer** (**Monostable Mode**) or an **Oscillator** (**Astable Mode**). 555 IC is a combination of digital and analog circuits.



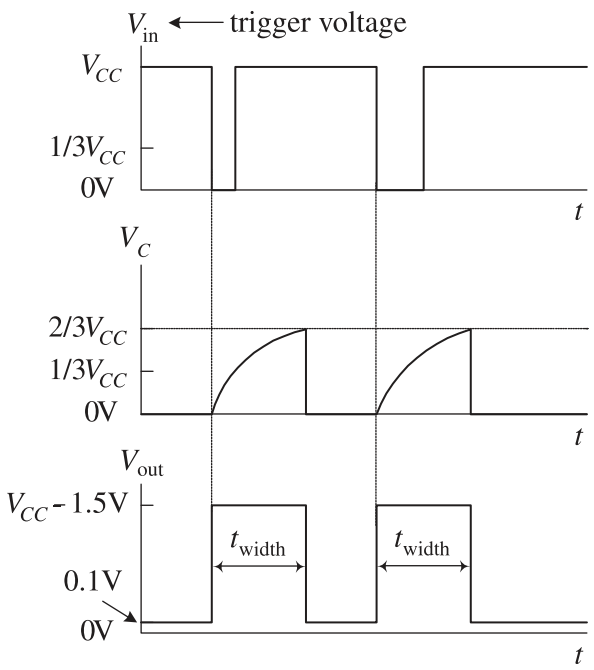


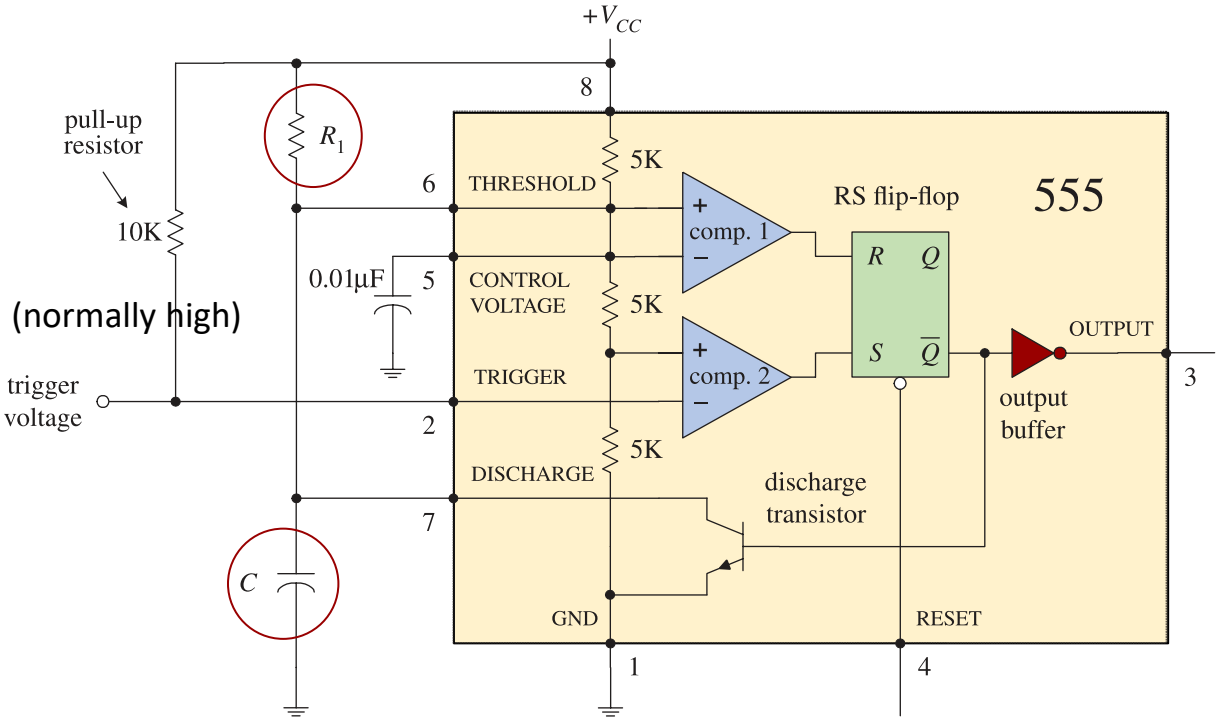
- The 555 gets its **name** from the three 5 kΩ resistors. These resistors act as a three-step **voltage divider** between the supply voltage (V_{CC}) and ground.

Capable of sinking or sourcing around 200 mA ($V_{\text{out}}(\text{high}) \cong V_{CC} - 1.5$ V and $V_{\text{out}}(\text{low}) \cong 0.1\ V$).

Monostable Mode (One-Shot)

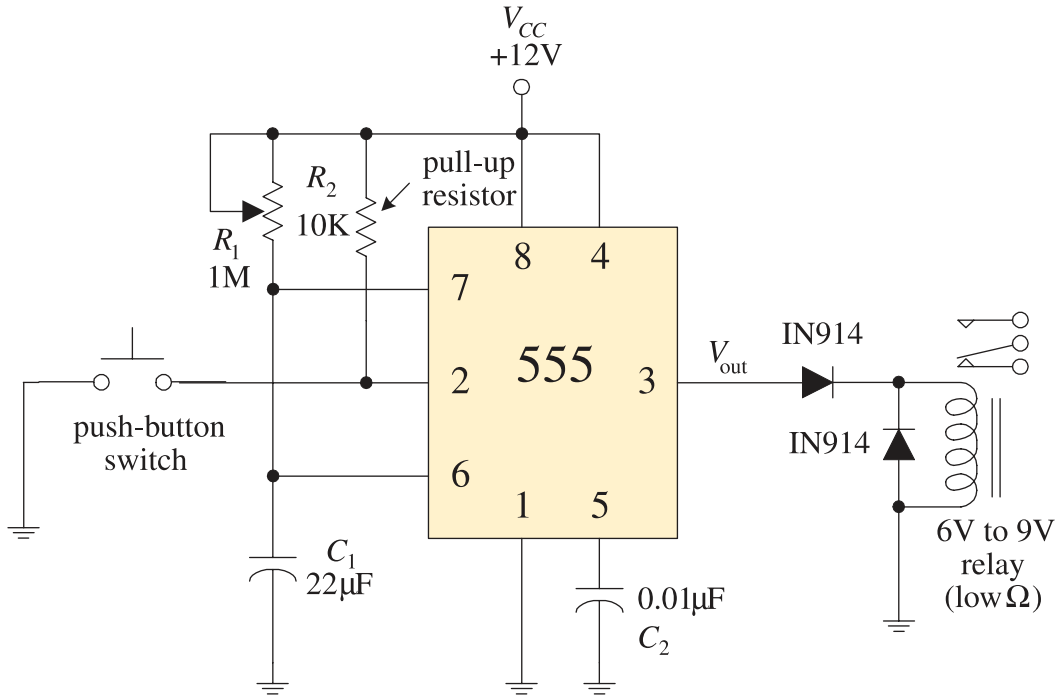
Monostable Mode (**One-Shot**) is constructed by adding an external capacitor and resistor to a 555. It is also called **monostable multivibrator**. It generates a single pulse of desired duration (based on the time constant of the resistor-capacitor) when it receives a trigger signal.





Pulse Length: $t_{\text{width}} = 1.1R_1C$

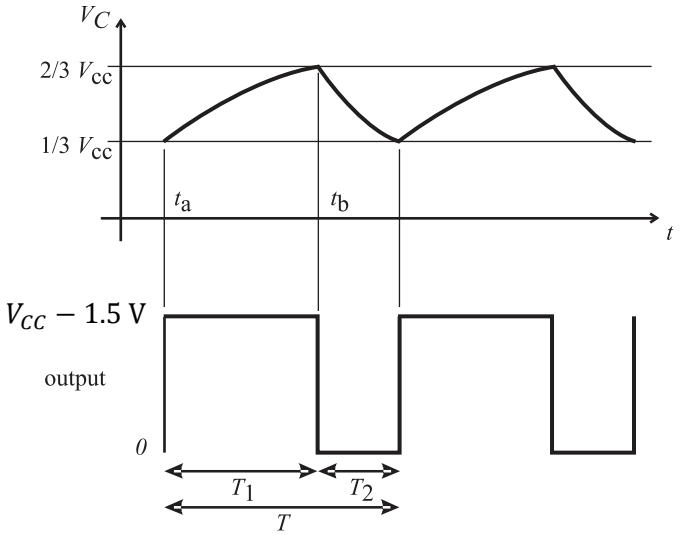
Application: Relay Driver

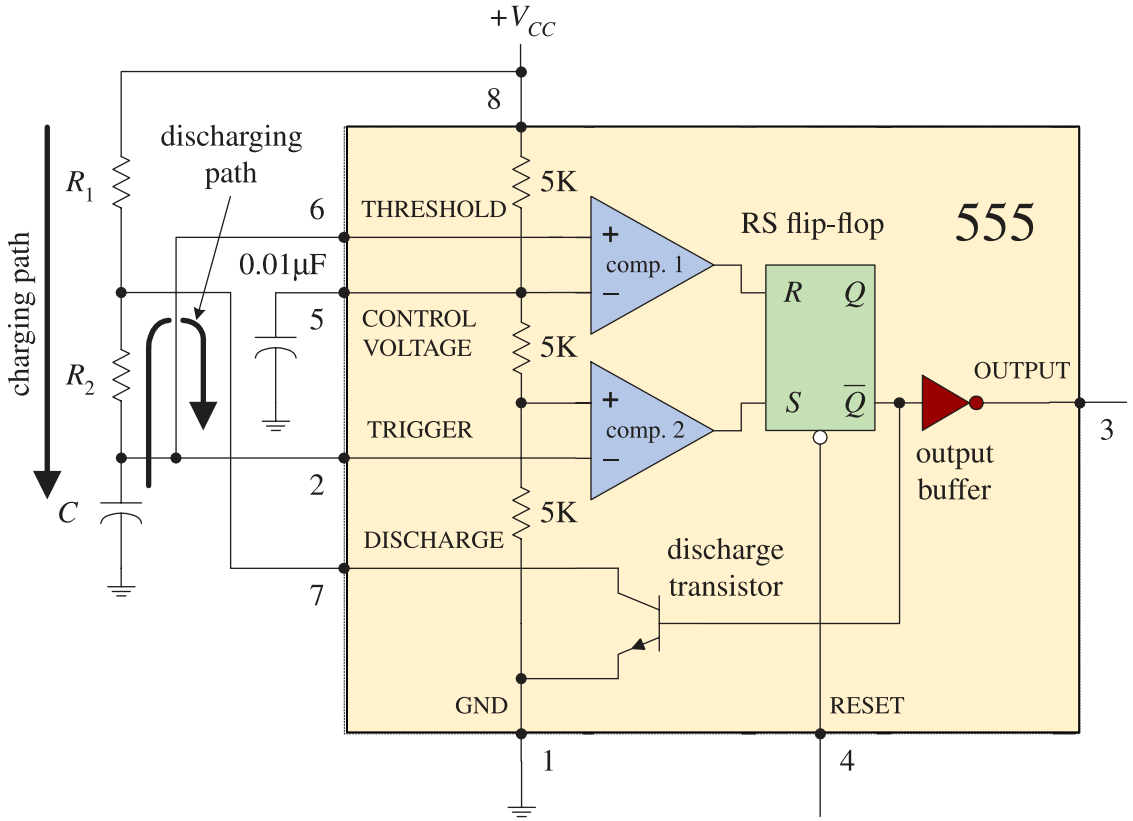


The circuit is used to actuate a relay for a given duration. With the pushbutton switch open, the output is low (around 0.1 V), and the relay is at rest. However, when the switch is momentarily closed, the 555 begins its timing cycle; the output goes high (12-1.5=10.5 V) for a duration equal to $1.1R_1C_1$. The relay will be actuated for the same time duration.

Astable Mode

Astable Mode is constructed by adding an external capacitor and two resistors to a 555. It is also called **astable multivibrator** or **square-wave generator**.



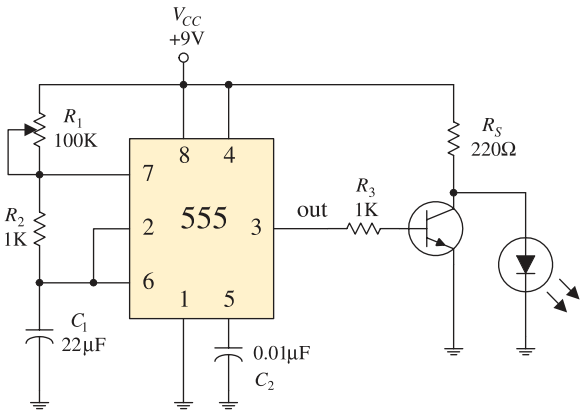


$$T_1 = \ln(2)\,(R_1 + R_2)C$$

$$T_2 = \ln(2)\,R_2 C$$

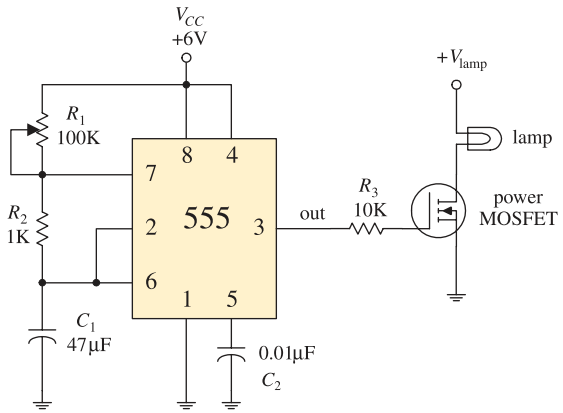
Application: LED and Lamp Flasher & Metronome

LED Flasher



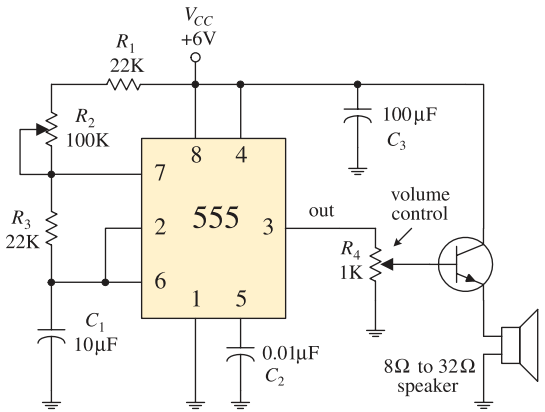
A transistor is used to amplify the 555's output to provide sufficient current to drive the LED, while R_S is used to prevent excessive current from damaging the LED.

Lamp Flasher



In the lamp-flasher circuit, a power MOSFET amplifier is used to control current flow through the lamp.

Metronome



The metronome circuit produces a series of clicks at a rate determined by R_2. To control the volume of the clicks, R_4 can be adjusted.

All these circuits are oscillator circuits (astable multivibrators).