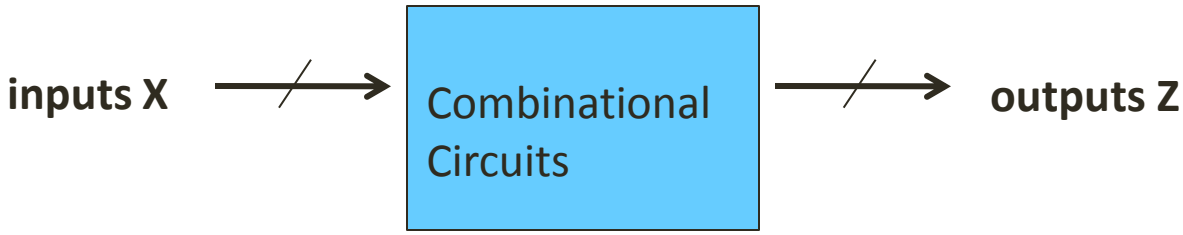
Contents

- Sequential Circuits
- Storage Elements
- Latches
- Flip-flops
- Analysis of Clocked Sequential Circuits
- State Reduction and Assignments
- Design Procedure
- Register
- Counters

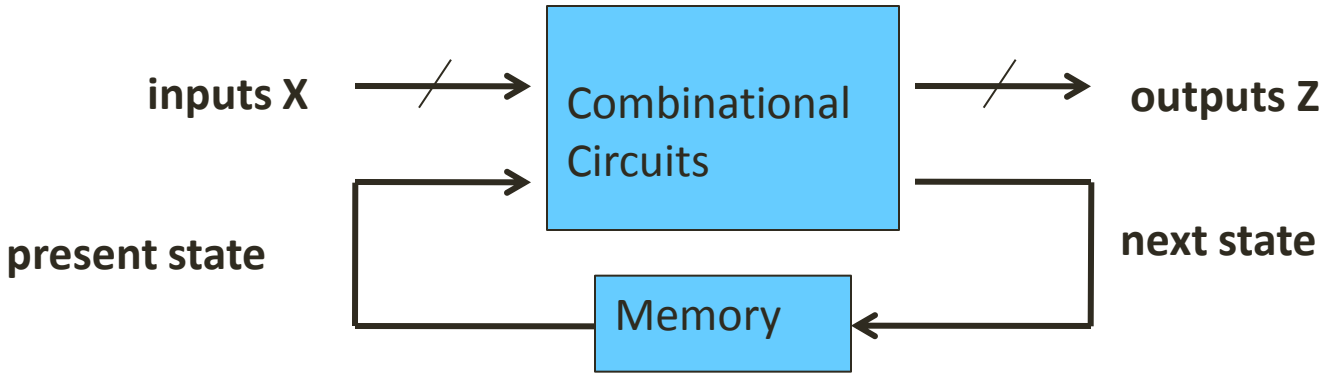
Combinational vs Sequential



In a **combinational** circuit:

- At any time, outputs depends only on inputs
  - Changing inputs cause changes in outputs
- No regard for previous inputs
  - No memory (history)
- Time is ignored !

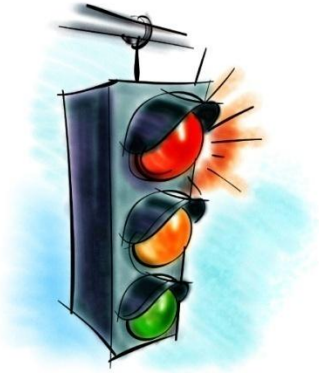
Combinational vs Sequential



A **sequential** circuit:

- A combinational circuit with **feedback** through **memory**
  - The stored information at any time defines a **state**
- Outputs depends on inputs and previous output
  - Previous outputs are stored as binary information into memory
- Next state depends on inputs and present state of output.

Examples of sequential systems



Traffic light

ATM

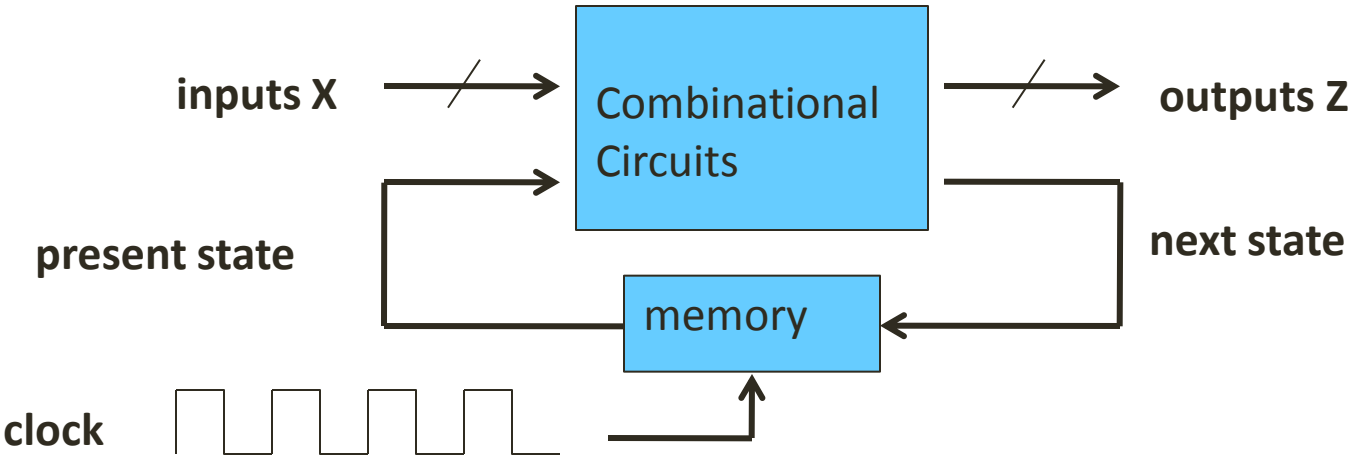
Vending machine

What is common between these systems?

Types of Sequential Circuits

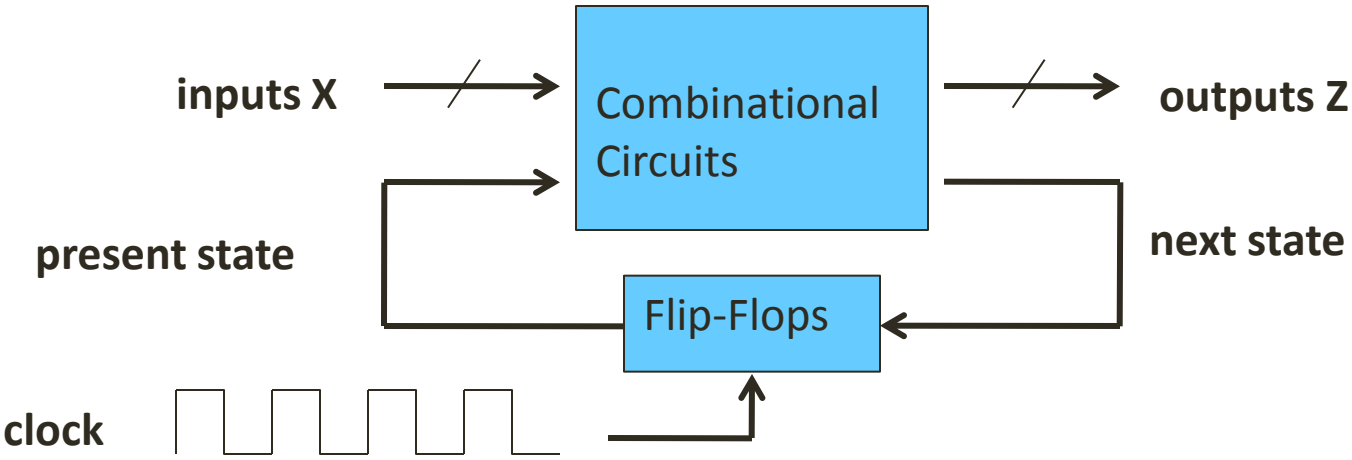
- Two types of sequential circuits:
  - **Synchronous**: The behavior of the circuit (output) changes at discrete instances of time on the change of the input values. Clock pulses are used.
  - **Asynchronous**: The behavior of the circuit (output) changes immediately on the changes of the input signals at any instance of time.

Synchronous Sequential Circuits



- Synchronous circuits employs a synchronizing signal called **clock** (a periodic train of pulses; 0s and 1s)
- A clock determines **when** computational activities occur
- Other signals determines **what** changes will occur

Synchronous Sequential Circuits



- The storage elements (memory) used in clocked sequential circuits are called **flip-flops**
  - Each flip-flop can store one bit of information 0,1
  - A circuit may use many flip-flops; together they define the circuit state
- Flip-Flops (memory/state) update **only** with the clock

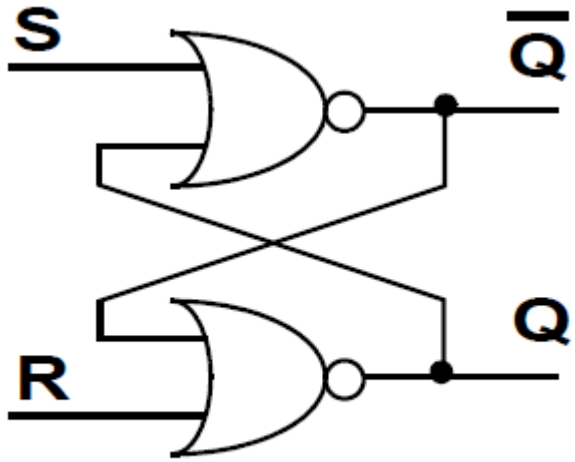
Storage Elements (Memory)

- A storage element can maintain a binary state (0,1) indefinitely, until directed by an input signal to switch state
- Main difference between storage elements:
  - Number of inputs they have
  - How the inputs affect the binary state
- Two main types:
  - **Latches** (level-sensitive)
  - **Flip-Flops** (edge-sensitive)
- Latches are useful in asynchronous sequential circuits
- Flip-Flops are built with latches

Latches

- A **latch** is binary storage element
- Can store a 0 or 1
- The most basic memory
- Easy to build
  - Built with gates (NORs, NANDs, NOT)

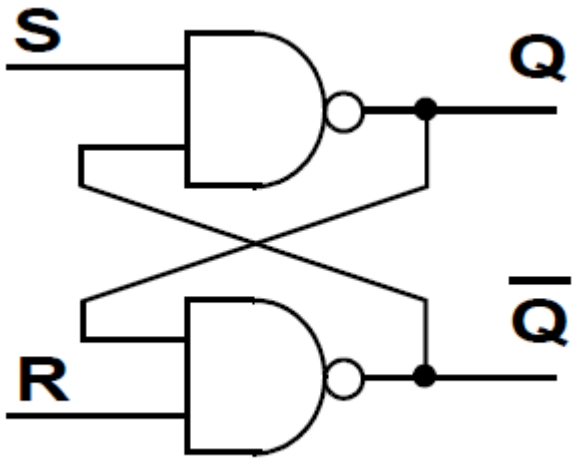
SR Latch using NOR gate (Active High)



Two states: Set (Q = 1) and Reset (Q = 0)
When S=R=0, Q remains the same,
S=R=1 is not allowed!
Normally, S=R=0 unless the state needs to be changed (memory?)
State of the circuit depends not only on the current inputs, but also on the recent history of the inputs or the current output.

S	R	Q^+	$\overline{Q}^+$	Function
0	0	Q	$\overline{Q}$	Storage State
0	1	0	1	Reset
1	0	1	0	Set
1	1	0-?	0-?	Indeterminate State

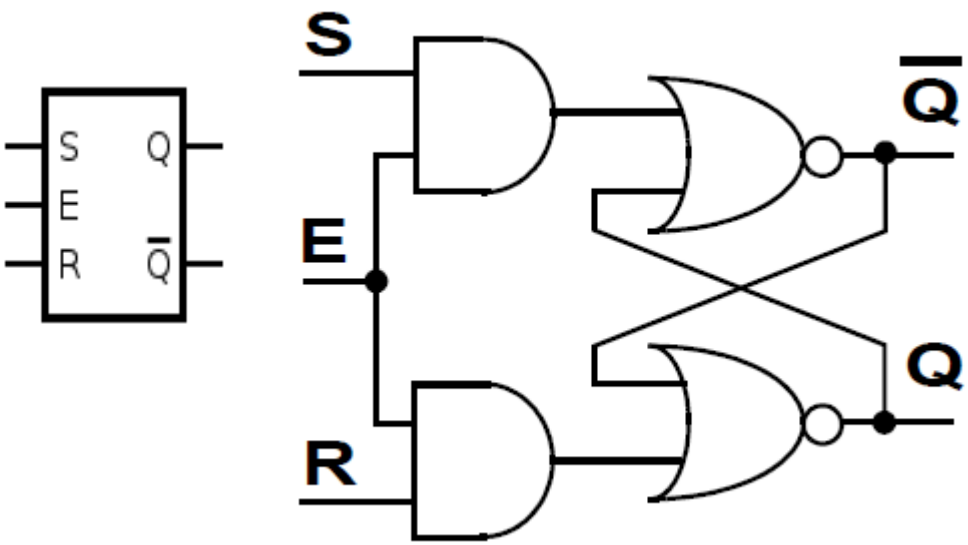
S R Latch using NAND gate (Active Low)



Similar to SR latch using NOR Gates.
Two states: Set (Q = 1) and Reset (Q = 0)
When S=R=1, Q remains the same
S=R=0 is not allowed!

S	R	Q^+	$\overline{Q}^+$	Function
0	0	1-?	1-?	Indeterminate State
0	1	1	0	Set
1	0	0	1	Reset
1	1	Q	$\overline{Q}$	Storage State

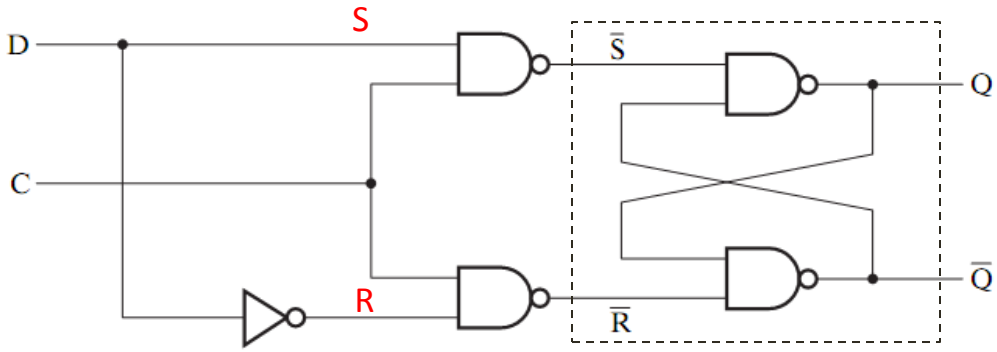
SR Latch with Enable



Asynchronous and level Sensitive
Cross Coupled NOR gates
S and R cannot be high same time.

E	S	R	Q^+	$\overline{Q}^+$	Function
0	x	x	Q	$\overline{Q}$	Storage State
1	0	0	Q	$\overline{Q}$	Storage State
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	0-?	0-?	Indeterminate State

D Latch



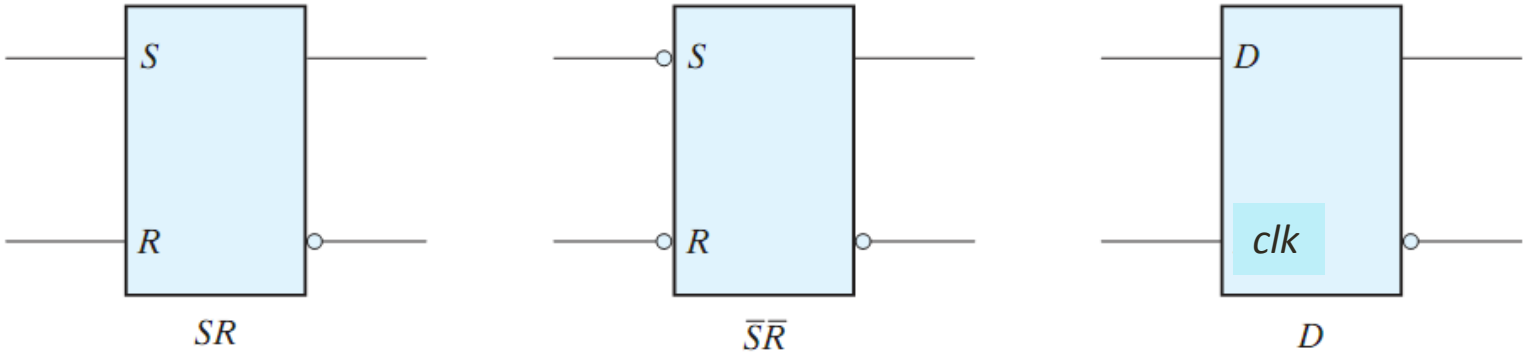
(a) Logic diagram

C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state

(b) Function table

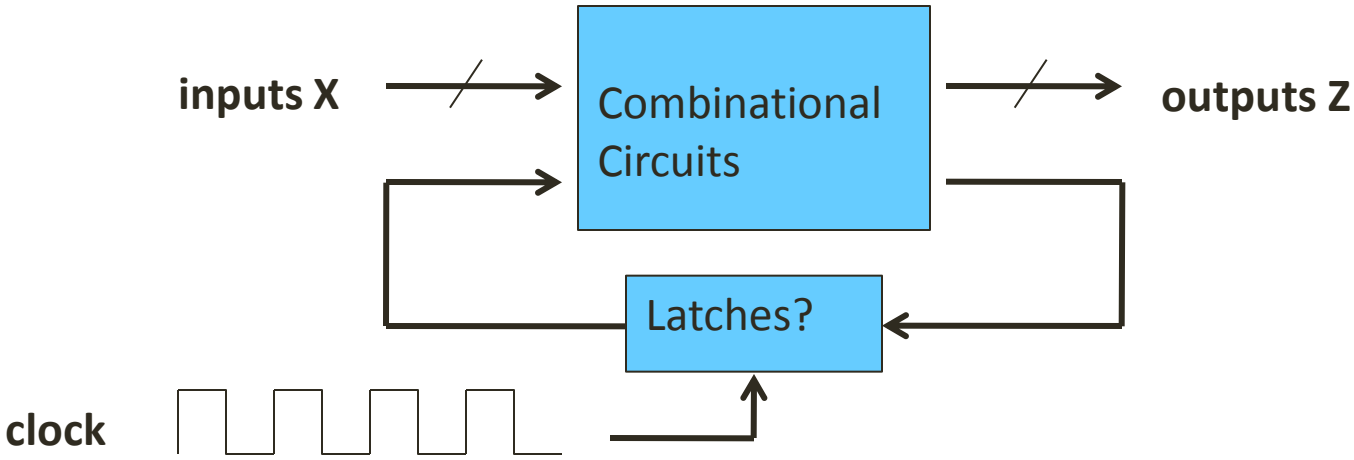
- Ensure S and R are never equal to 1 at the same time
- Add inverter
- Only one input (D)
  - D connects to S
  - D' connects to R
- D stands for data
- Output follows the input when C = 1
  - Transparent
- When C = 0, Q remains the same

Graphic Symbols for Latches



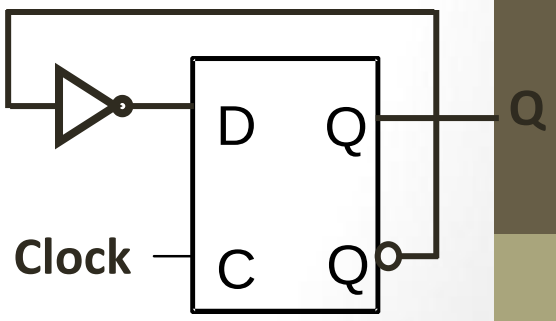
- A latch is designated by a rectangular block with inputs on the left and outputs on the right
- One output designates the normal output, the other (with the bubble) designates the complement
- For S'R' (SR built with NANDs), bubbles added to the input

Problem with Latches



- What happens if Clock=1? What will be the value of Q when Clock goes to 0?
- **Problem:** A latch is **transparent**; state keep changing as long as the clock remains active
- Due to this uncertainty, latches can not be reliably used as storage elements.

Example



Flip Flops

- A **flip-flop** is a one bit memory similar to latches
- Solves the issue of latch transparency
- Latches are **level** sensitive memory element (It has Enable input)
  - Active when the clock = 1 (whole duration)
- Flip-Flops are **edge-triggered** or **edge-sensitive** memory element (It has clock input)
  - Active only at transitions; i.e. either from 0 → 1 or 1 → 0