Chapter 6: Sequential Circuits

时序逻辑电路

Dr Guohun Zhu

Email: g.zhu@uq.edu.au or guohun.zhu.phd@ieee.org

Presentation Outline

- Introduction to Sequential Circuits
- Synchronous versus Asynchronous (同步 vs 异步）
- Review Latches
- Review Flip-Flops
- Characteristic Tables and Equations

Reference

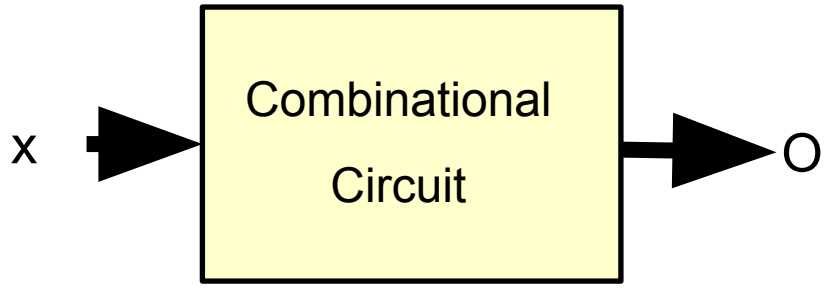
- Digital Electronics Principles and Applications by

  Roger L. Tokheim

- Sequential logic
  - https://github.com/computation-structures/course/blob/main/lectures/L05_Sequential_Logic.md
- Digital design source code
  - https://github.com/hneemann/Digital
  -

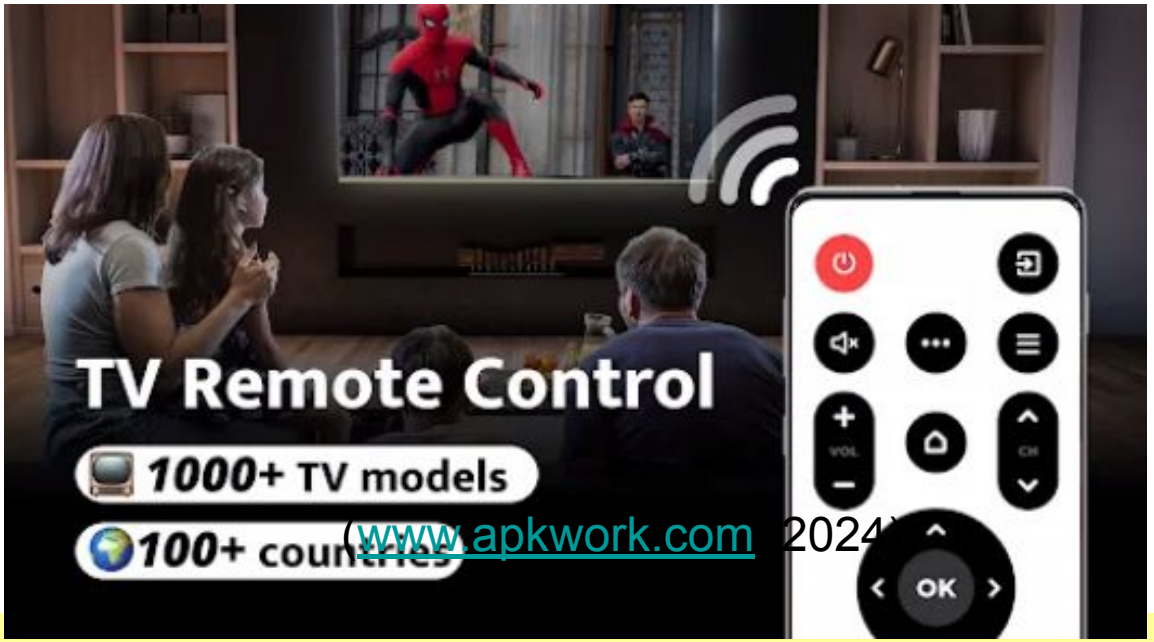
The Classes of Digital Circuits

- Two classes of digital circuits. (两类数字电路)
  - Combinational Circuits
  - Sequential Circuits
- Combinational Circuits （组合电路）
  - O = F(x)
  - Function of Inputs only
  - NO internal memory（没有内部存储）
- Sequential Circuits （时序电路）
  - O = F(x, ？)
  - Have memory ability（有记忆能力）



A Sequential Logical Example

- TV Remote Control with Channel Up and Channel Down Buttons
  - What is the next channel if press “Channel Up” Button?
  - If current channel number : 8, then next channel ?
  - If current channel number : 6, then next channel ?
  - The remote control's output is dependent on its previous state(O_0)
    - and **Buttons: x**?
  -
  - $O_1 = F(x, O_0)$

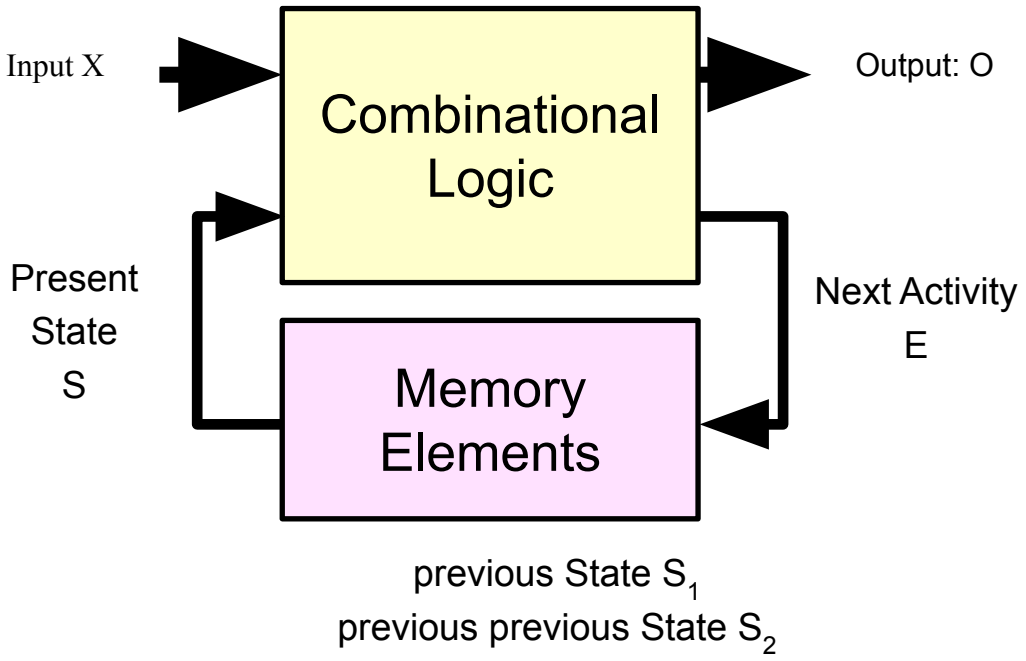


(www.apkwork.com 2024)

Introduction to Sequential Circuits

A Sequential circuit consists of: (Text books page 247)

1. Memory elements:
   - ◇ **Latches** or **Flip-Flops**
   - ◇ Store the **Present State: S**
2. Combinational Logic
   - ◇ Computes the **Outputs** :O
   - ◇ Computes the **Next Activity State** : E



Both O, E depends on the Inputs and the Present State Q

However, S could be affected by E and the memory (old S)

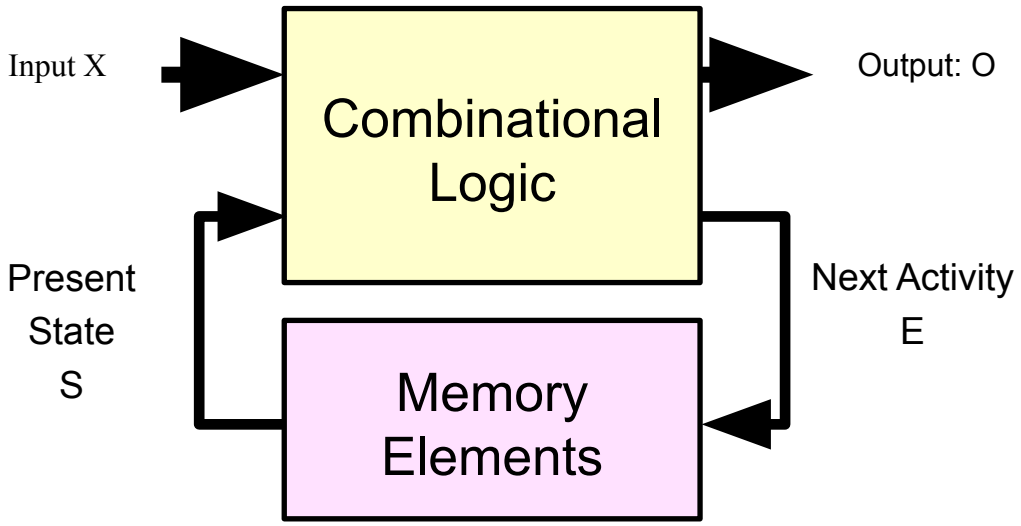
时序电路的描述方式：逻辑方程

Text book page 247

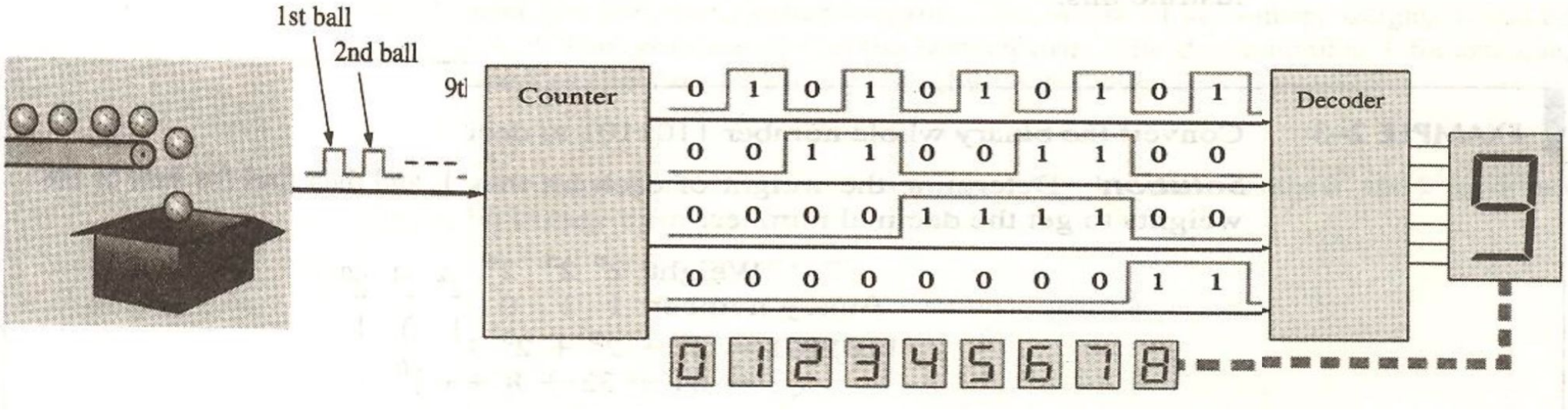
输出方程：$O=f_1(X,\ S)$

激励方程　　$E=f_2(X,\ S)$

状态方程　$S_{n+1}=f_3(E,\ S_n)$



A Counter Example



Counter increase one when a pulse signal was obtained. If there has no pulse, the state has no changed.

The pulse is called a clock

Assume that the output and all states change simultaneously: Synchronous（同步）

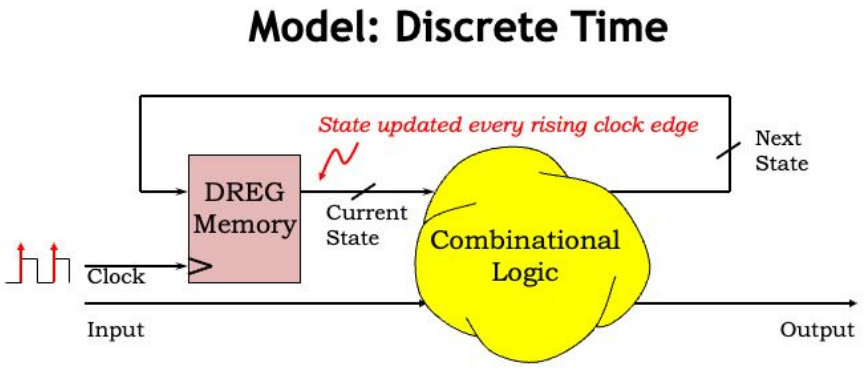
Output and States are changed at any time. Asynchronous（异步）

Two Types of Sequential Circuits

1. **Synchronous** Sequential Circuit 同步时序电路

- ◇ Uses clock signals as additional inputs
  - single-clock synchronous 单时钟同步
  - Multiple clock signals is possible, but analyzing (the timing) is quite tricky
- ◇ Changes in the memory elements are controlled by the clock
- ◇ Changes happen at discrete instances of time

变化发生在离散的时间点



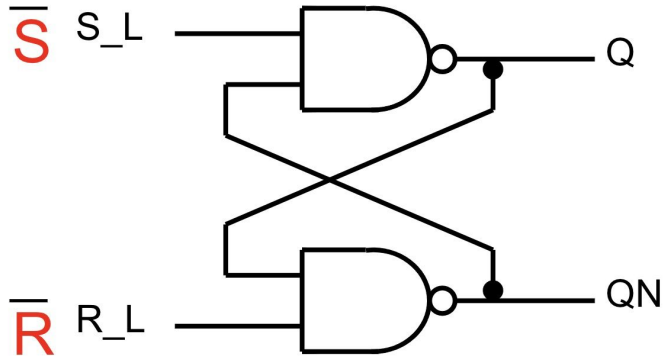
Active Clock Edges punctuate time ---

- Discrete Clock periods
- Sequences of states
- Simple rules – eg truth tables – relating outputs to inputs and the current state)
- ABSTRACTION: Finite State Machines (next lecture!)

抽象化：有限状态机

2. **Asynchronous** Sequential Circuit

- ◇ No clock signal
- ◇ Changes in the memory elements can happen at any instance of time
- ◇ Example: SR Latch（SR锁存器）



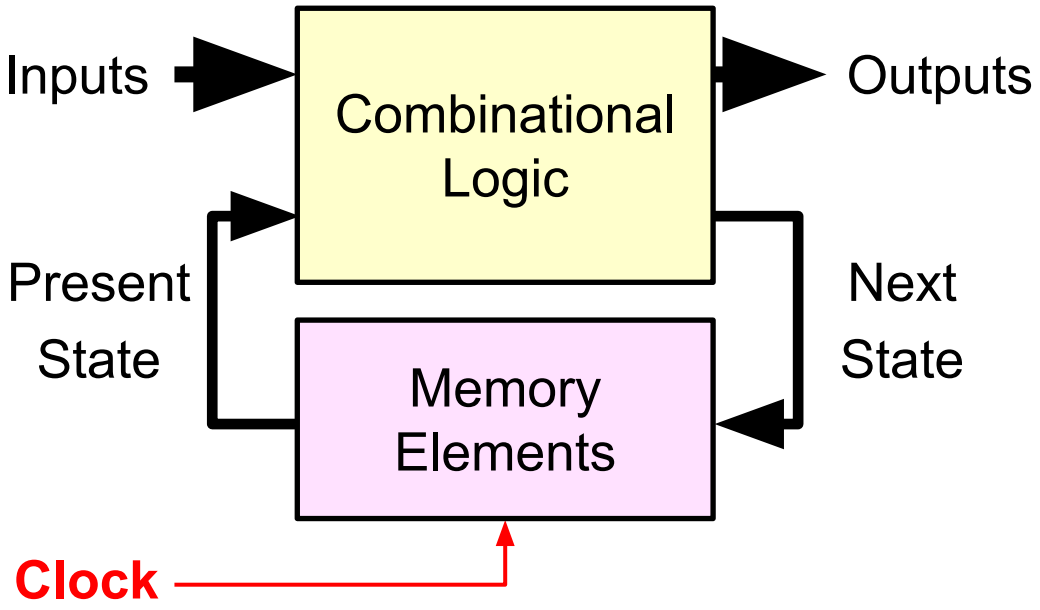
S-R Latch 功能表

S_L	R_L	Q	QN
1	1	维持现态	
1	0	0	1
0	1	1	0
0	0	1*	1*

❖ Advantage and Drawbacks

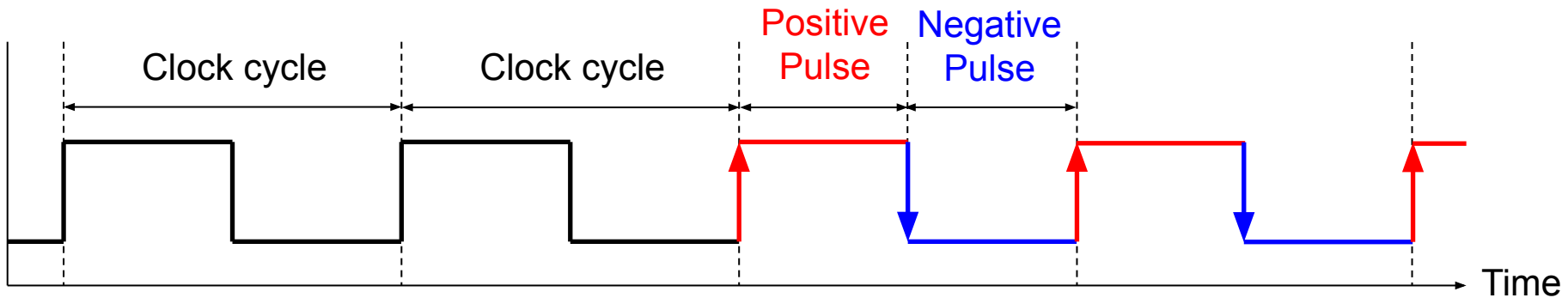
- ◇ Faster as there has no clock
- ◇ Difficult to design and analyze than synchronous sequential circuits

Synchronous Sequential Circuits



- Synchronous sequential circuits use a **clock signal**
- The clock signal is an input to the memory elements
- The clock determines **when** the memory should be updated
- The **present state** = output value of memory (stored)
- The **next state** = input value to memory (not stored yet)

The Clock



- Clock is a periodic signal 时钟是周期信号
- **Positive Pulse**: when the level of the clock is **1** **正脉冲**
- **Negative Pulse**: when the level of the clock is **0** **负脉冲**
- **Rising Edge**: when the clock goes **from 0 to 1** **上升沿**
- **Falling Edge**: when the clock goes **from 1 down to 0** **下降沿**

Memory Elements

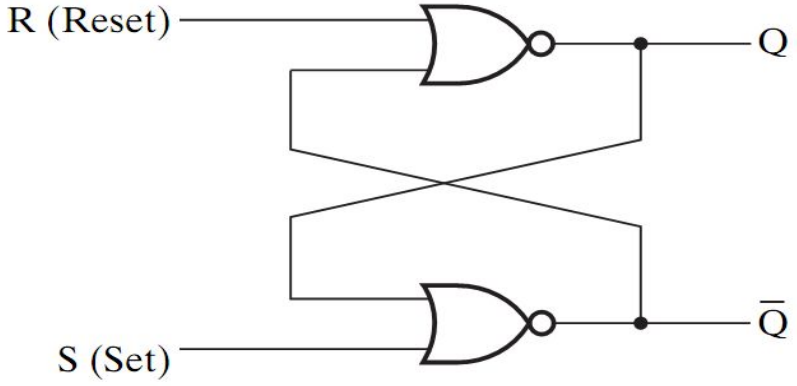
- Memory can store and maintain binary state in electronically
  - 1-bit data: 0 or 1
- Alternative memory styles（其他存储形式）
  - Magnetic（磁性), optical (光学), or acoustic（声学）
  - Quantum memory（量子存储）: qubits（量子位）
- Two main types:
  - Latches : level-sensitive (锁存器是电平敏感)
  - Flip-Flops : edge-sensitive (触发器是边沿敏感)
- Flip-Flips are used in synchronous sequential circuits
- Flip-Flops are built with latches

Next . . .

- Introduction to Sequential Circuits
- Synchronous versus Asynchronous
- **Review Latches**
- Review Flip-Flops
- Characteristic Tables and Equations

Review SR Latch

- A **latch** is a memory element that can store 0 or 1
- An **SR Latch** can be built using two **cross-coupled** NOR gates（或非门）
- 请问能否用 NAND gates 建成 SR Latch?
- Two inputs: S (Set) and R (Reset)
- Two outputs: Q and $\overline{Q}$

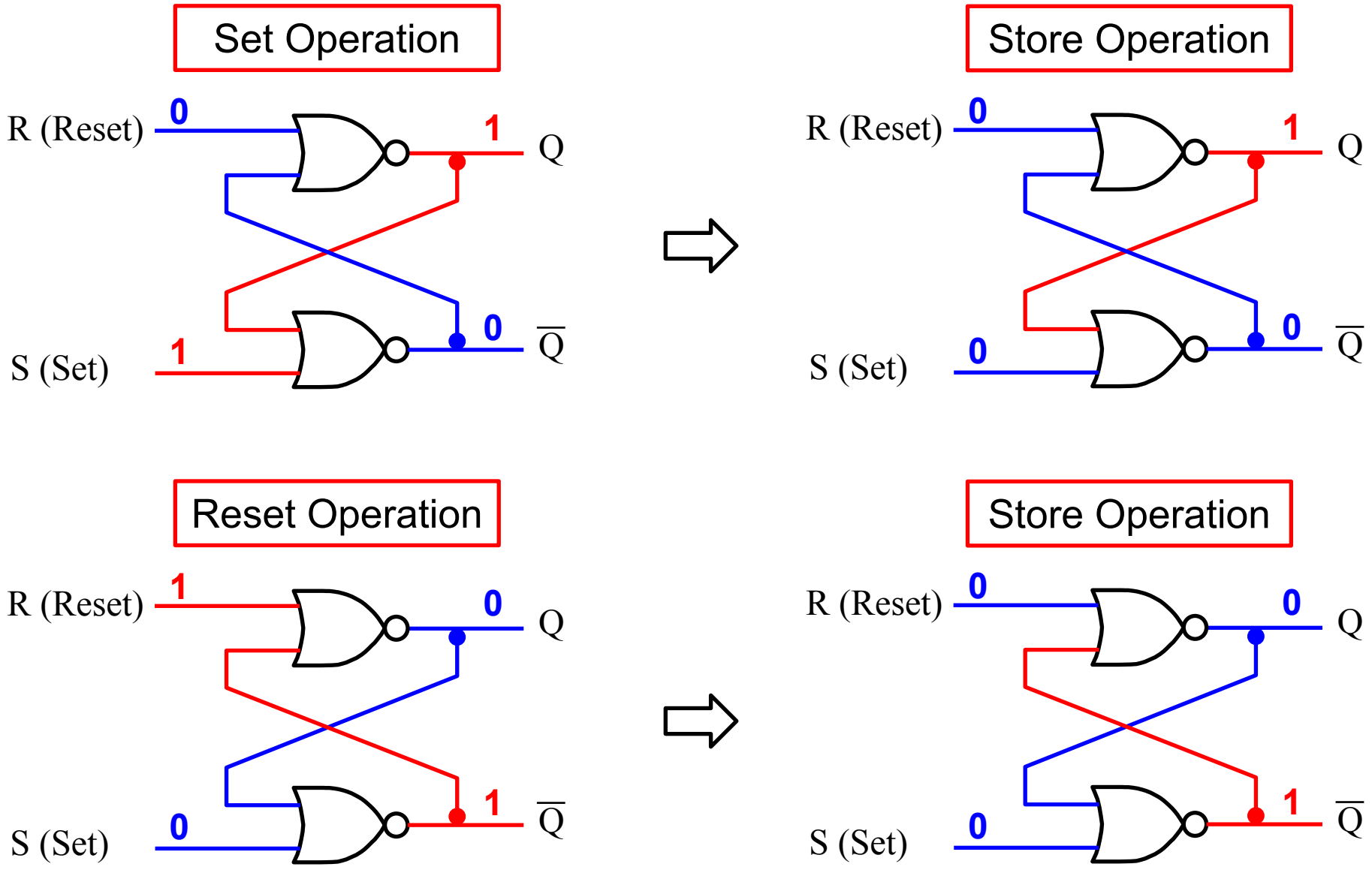


(a) Logic diagram

S	R	Q	$\overline{Q}$	
1	0	1	0	Set state
0	0	1	0	
0	1	0	1	Reset state
0	0	0	1	
1	1	0	0	Undefined

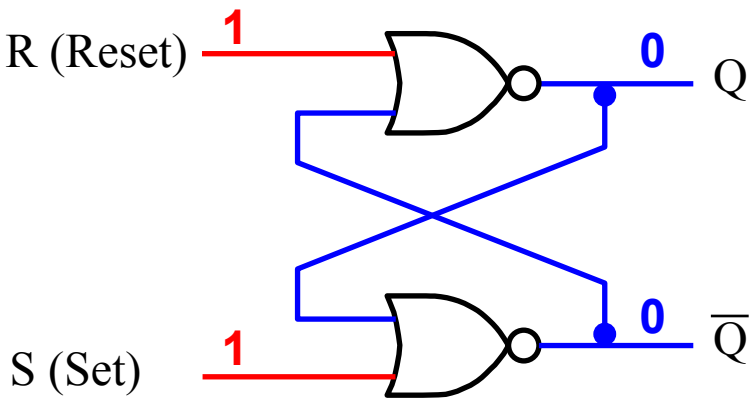
(b) Function table

SR Latch Operation

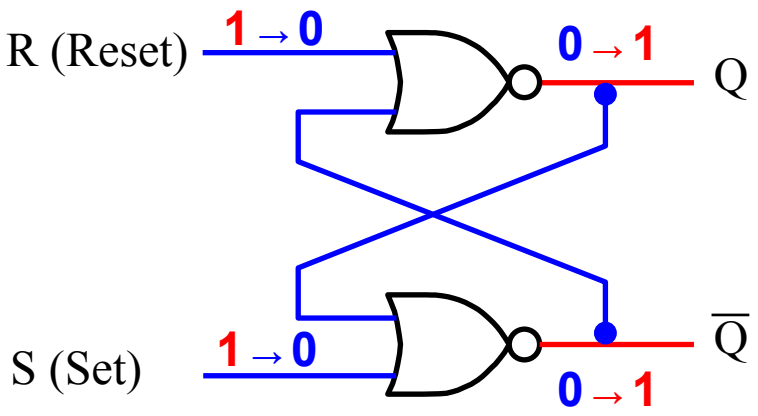


SR Latch Invalid Operation

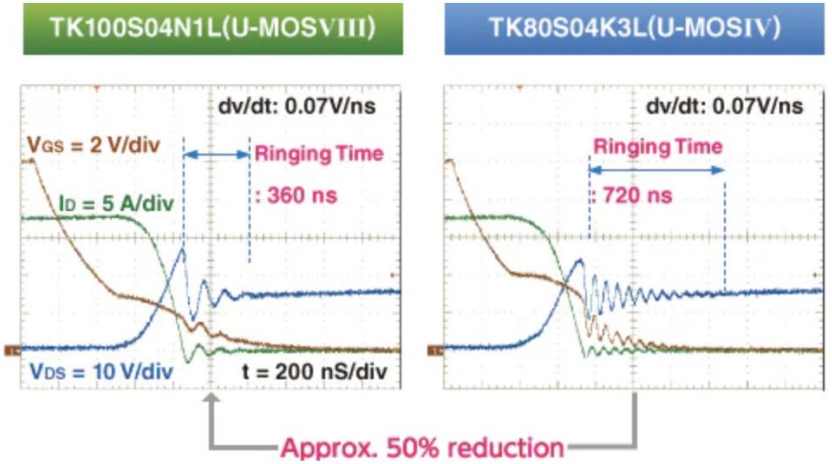
Invalid Operation



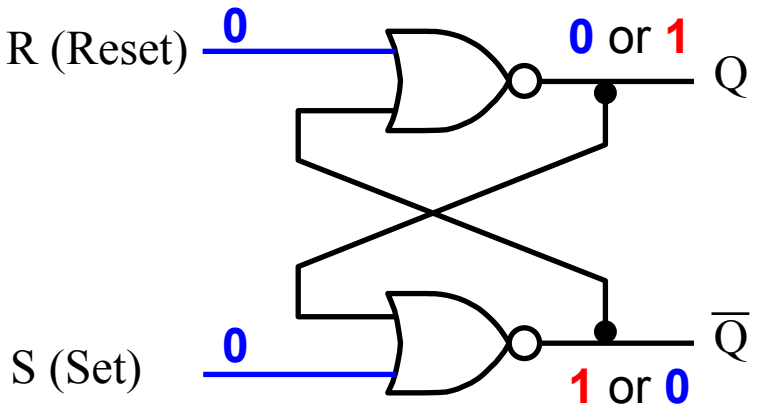
Race Condition



S = R = 1 should never be used

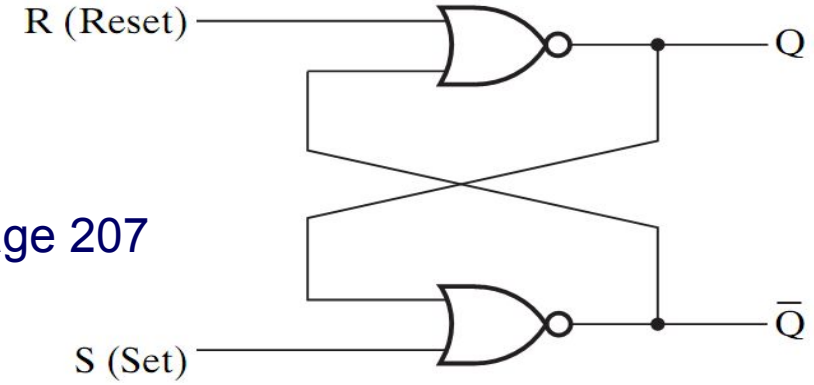


Unknown State



Timing Diagram of an SR Latch (时序图)

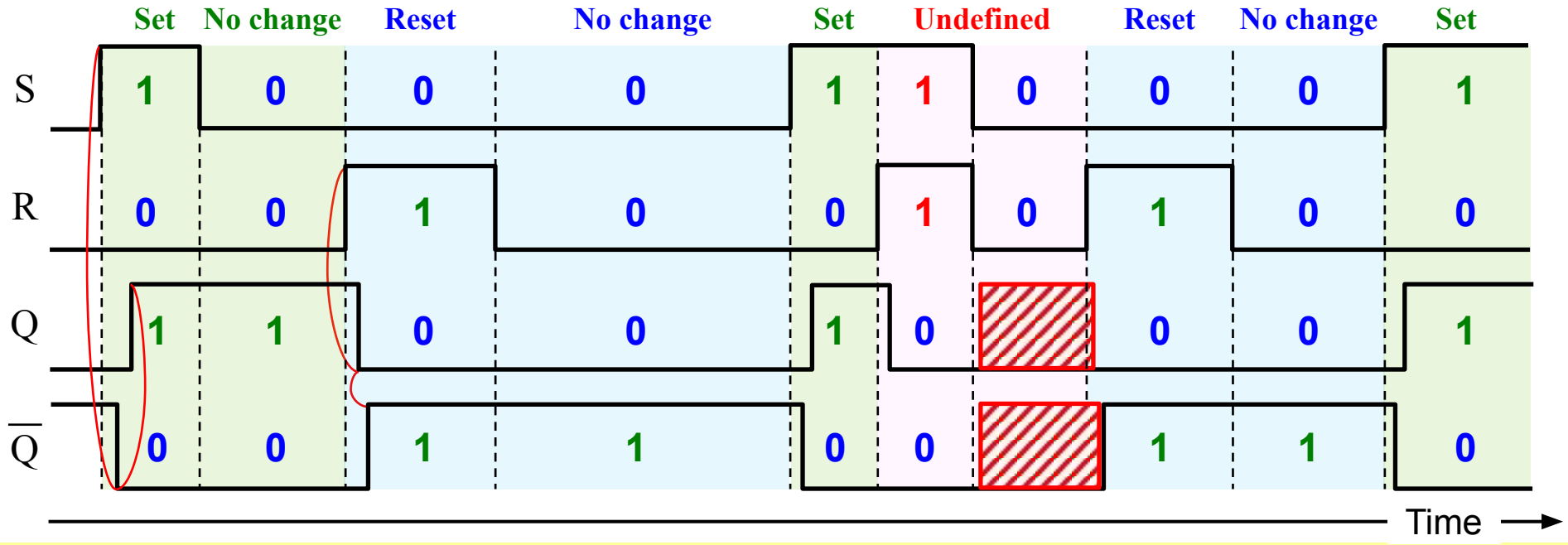
Text Books page 207



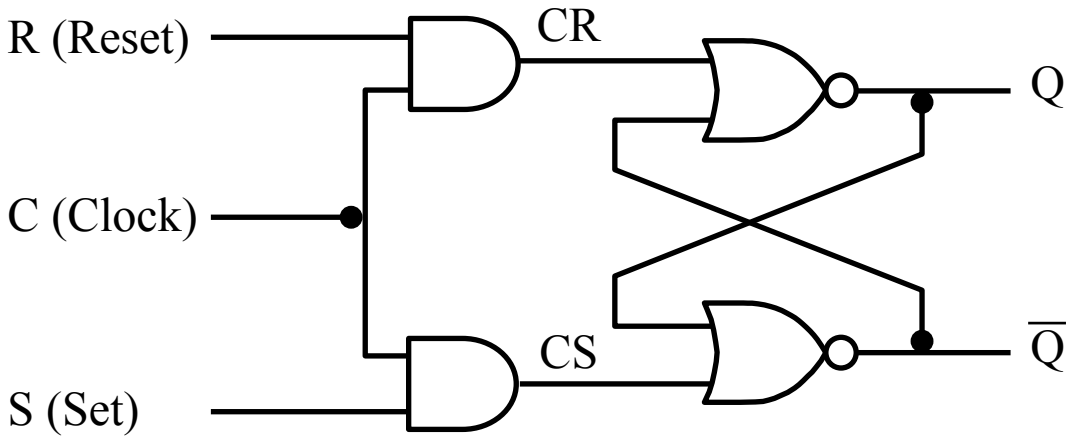
(a) Logic diagram

S	R	Q	$\overline{Q}$	
1	0	1	0	Set state
0	0	1	0	
0	1	0	1	Reset state
0	0	0	1	
1	1	0	0	Undefined

(b) Function table



Gated SR Latch with Clock Enable



C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; Set state
1	1	1	Undefined

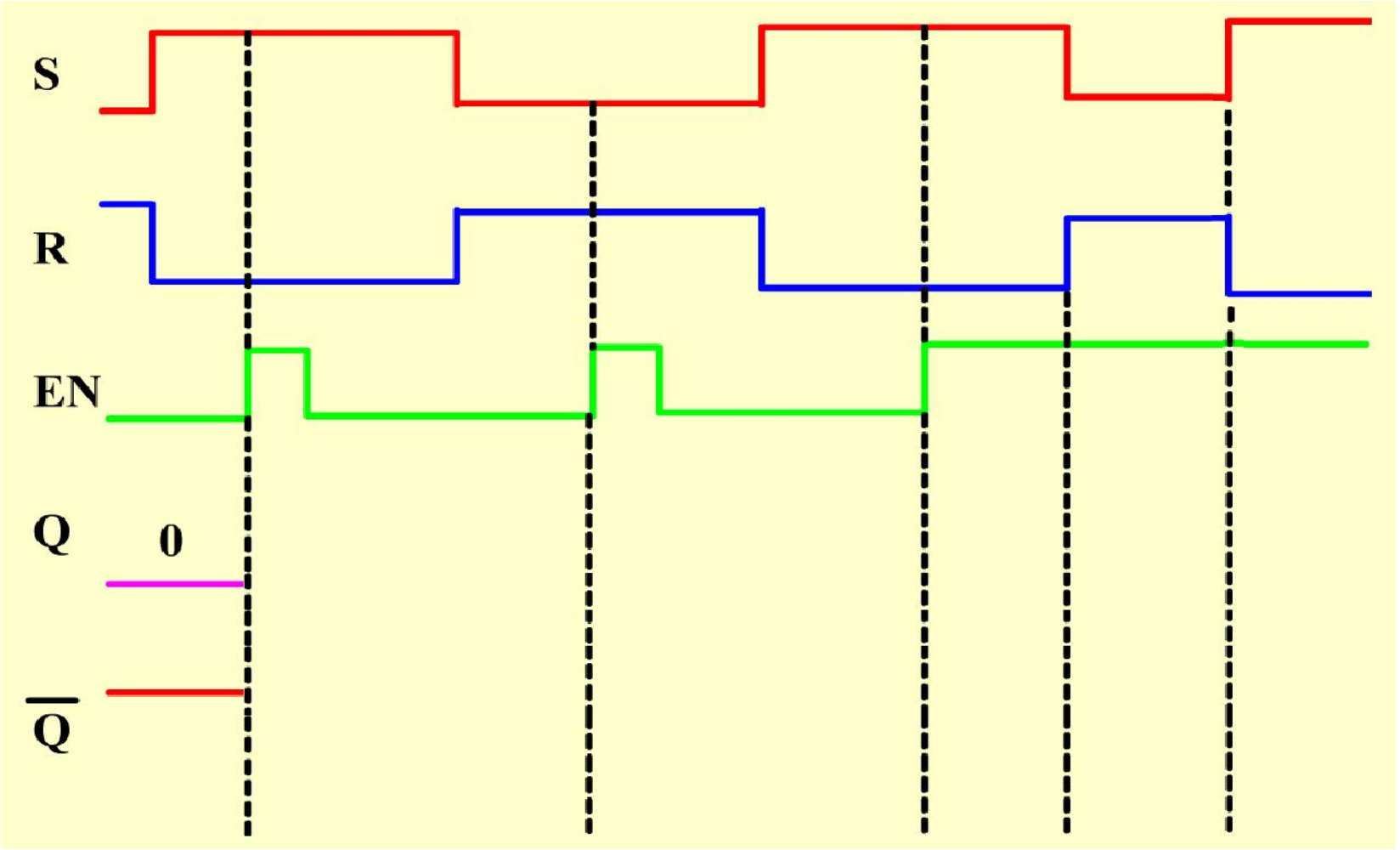
- An additional Clock (enable) input signal **C** is used
- Clock controls **when** the state of the latch can be changed
- When **C=0**, the S and R inputs have no effect on the latch

  The latch will remain in the same state, regardless of S and R
- When **C=1**, then normal SR latch operation

Text Books page 210

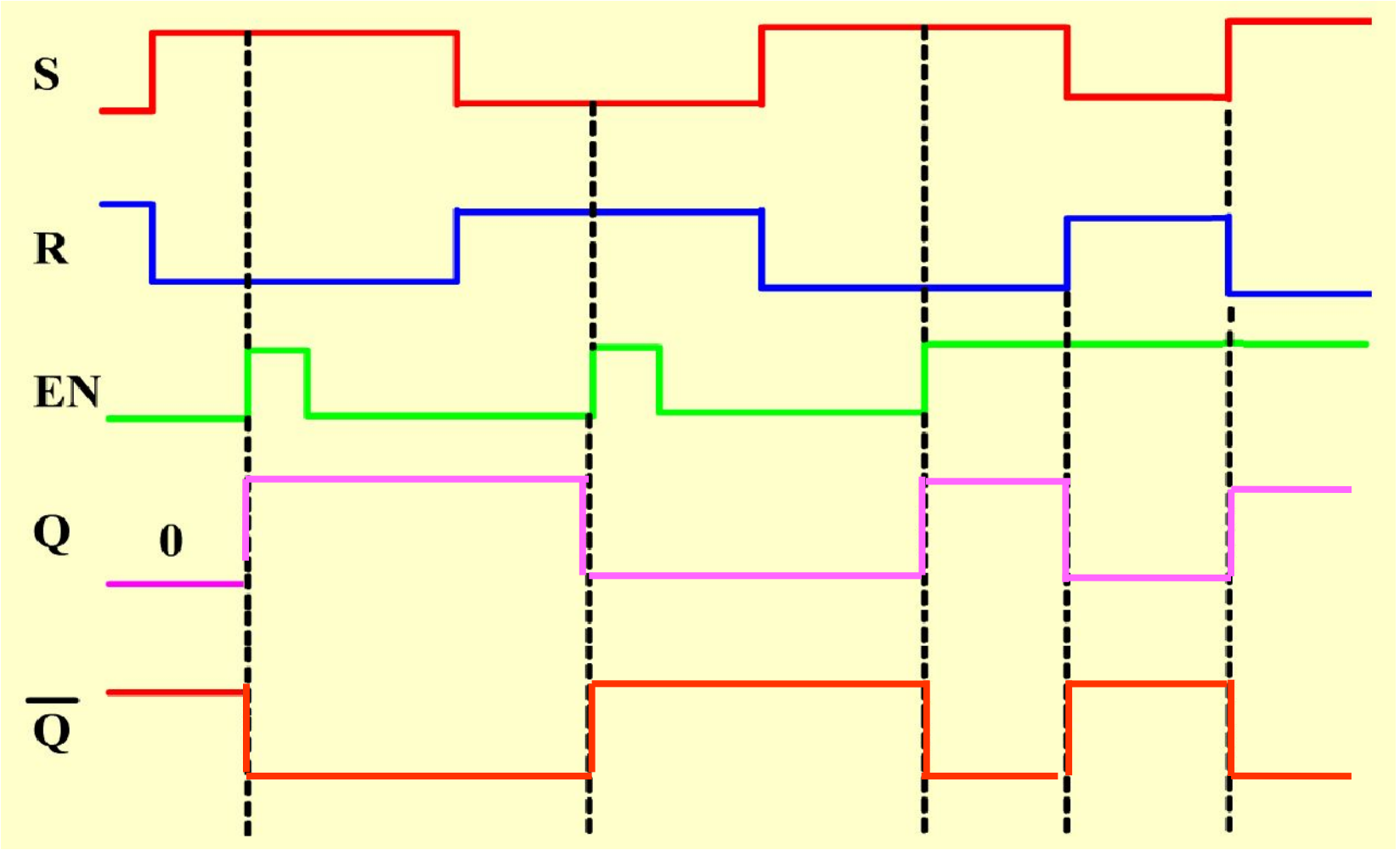
Timing Diagram of an SR Latch with Enable

请补充完整时序(波形)图

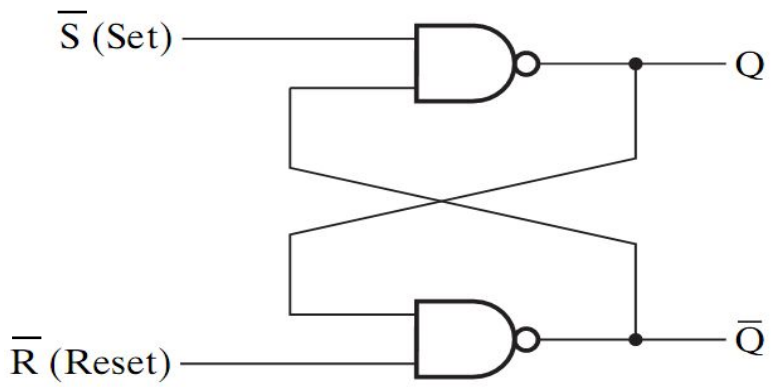


Timing Diagram of an SR Latch with Enable

答案:



$\overline{S}$ $\overline{R}$ Latch with NAND Gates



(a) Logic diagram

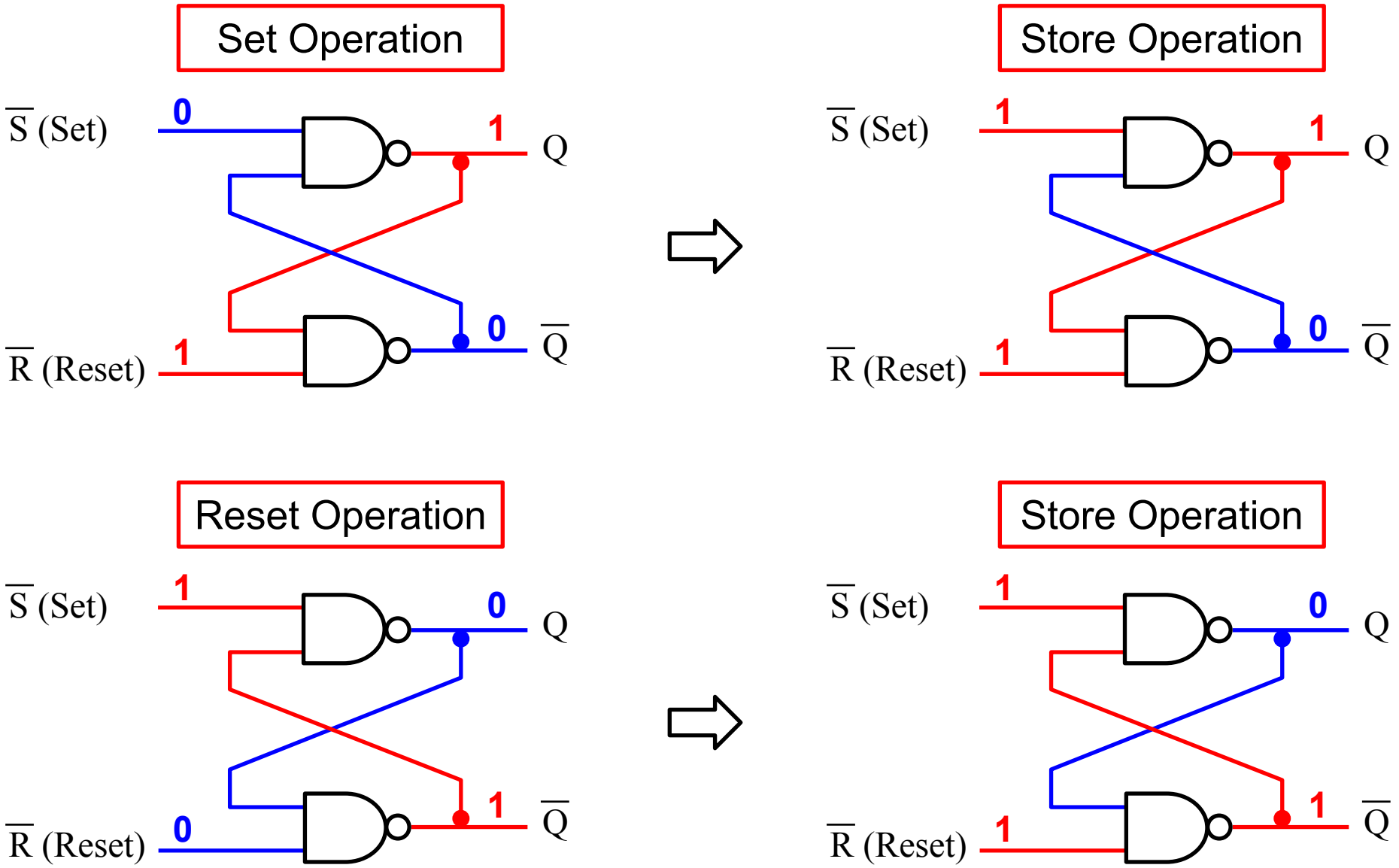
$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	
0	1	1	0	Set state
1	1	1	0	
1	0	0	1	Reset state
1	1	0	1	
0	0	1	1	Undefined

(b) Function table

Known as the Latch

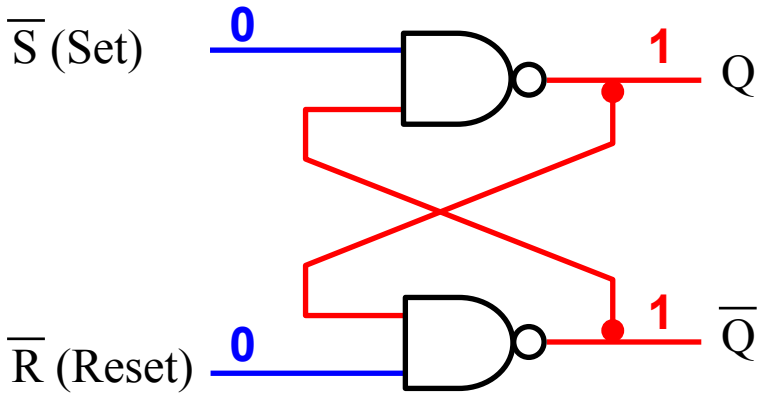
- If $\overline{S}$=? and $\overline{R}$=? then **Set** (Q=1)
- If $\overline{S}$=? and $\overline{R}$=? then **Reset** (Q=0)
- When $\overline{S}$=$\overline{R}$= ?, and Q is unchanged (remain the same)
- The latch stores its outputs Q and as long as $\overline{S}$=$\overline{R}$=?
- When $\overline{S}$=$\overline{R}$=? and Q is undefined (should never be used)

$\overline{S}\,\overline{R}$ Latch Operation

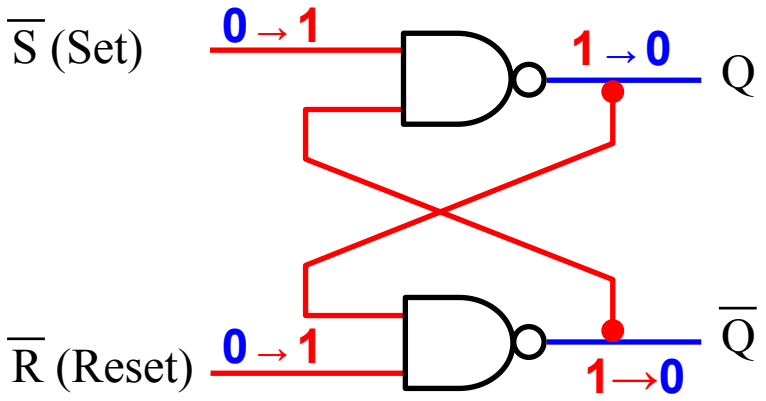


$\overline{S}\ \overline{R}$ Latch Invalid Operation

Invalid Operation



Race Condition

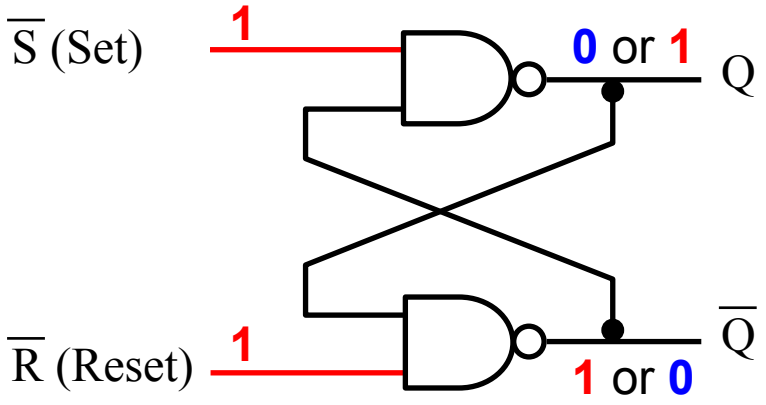


$\overline{S} = \overline{R} = 0$ should never be used

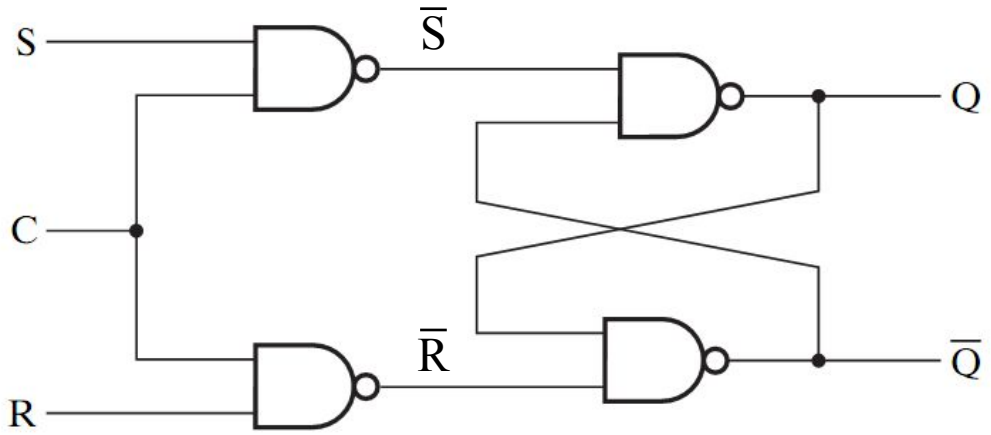
If $\overline{S}$ and $\overline{R}$ change from 0 → 1 simultaneously then race condition (oscillation) occurs

Final Q and $\overline{Q}$ are unknown

Unknown State



Gated SR Latch with Clock Enable



(a) Logic diagram

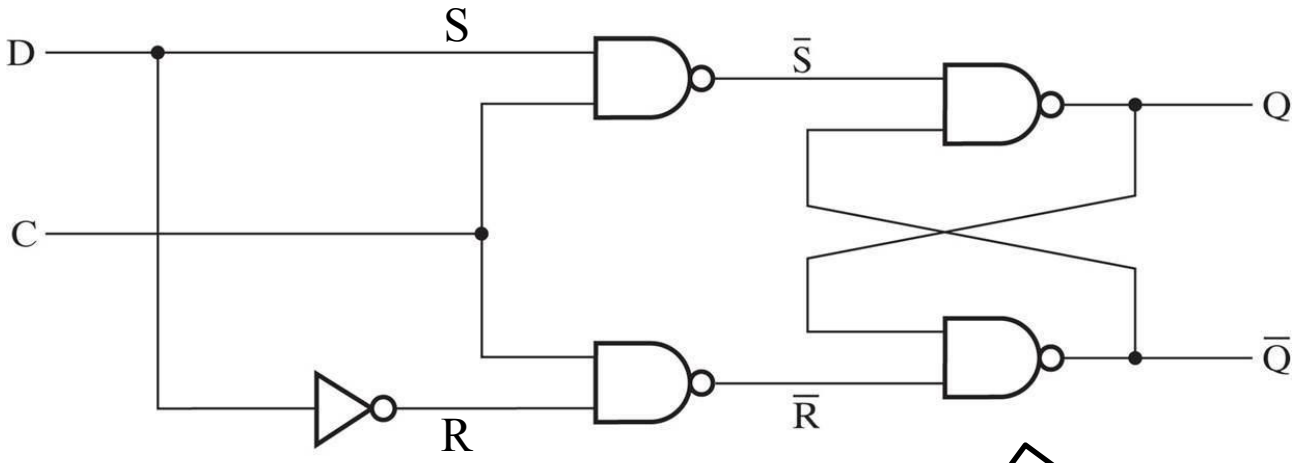
C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; Set state
1	1	1	Undefined

(b) Function table

- An additional Clock (enable) input signal **C** is used
- Clock controls **when** the state of the latch can be changed
- When **C=0**, the latch remains in the same state
- When **C=1**, then normal latch operation

  The NAND gates invert the **S** and **R** inputs when **C=1**

D-Latch with Clock Enable

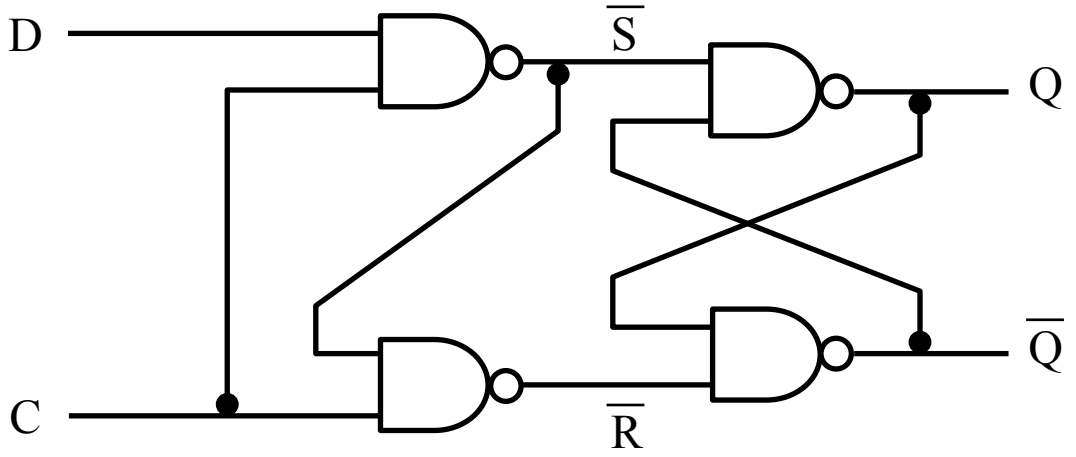


(a) Logic diagram

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

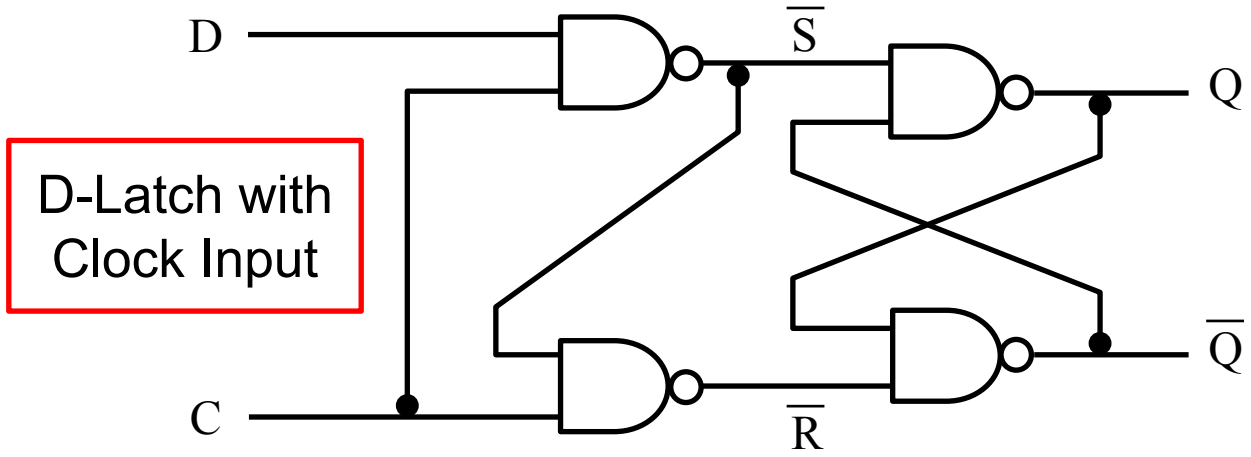
(b) Function table

C	D	S	$\overline{R}$	$\overline{S}$	R
1	0	0	1	1	0
1	1	1	0	0	1



Text Books page 212

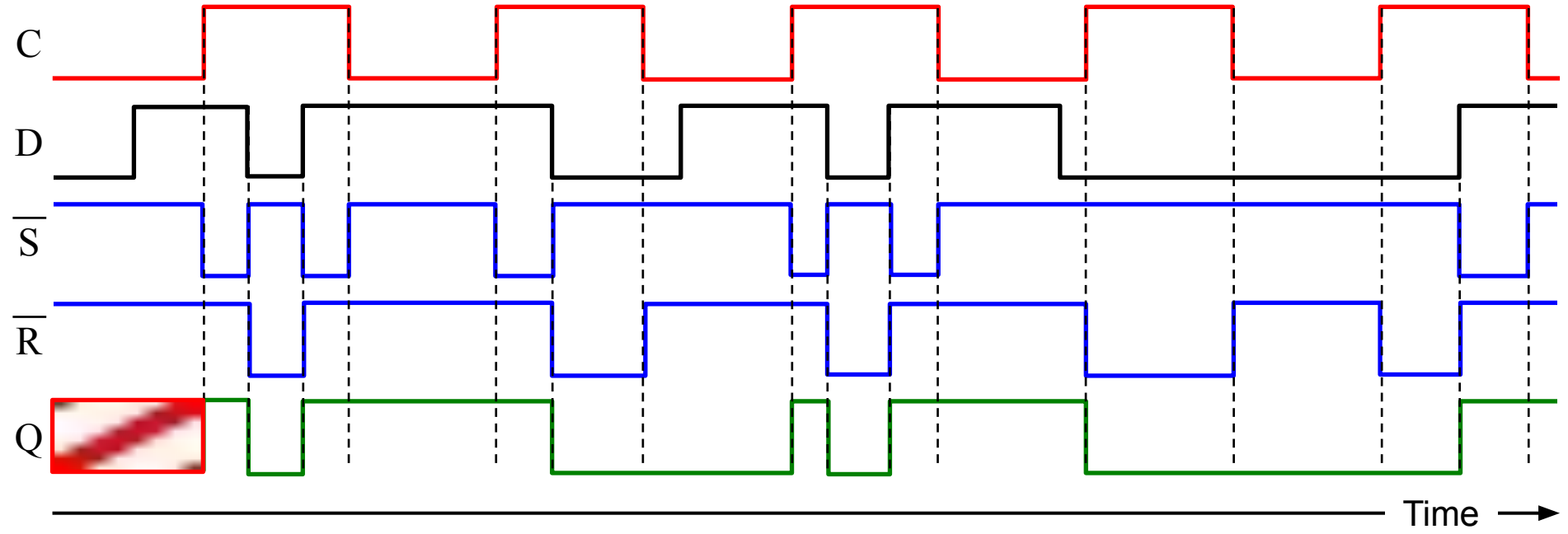
Timing of a D-Latch with Clock Enable



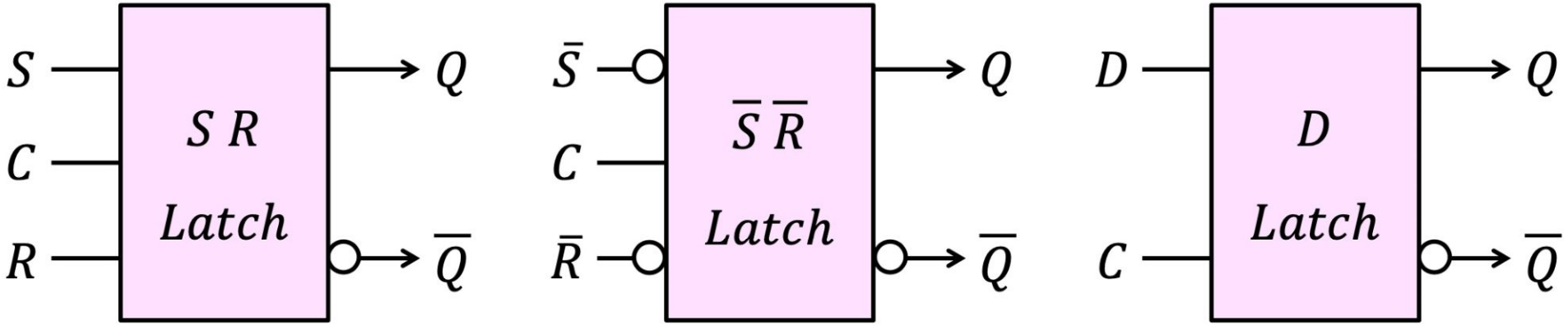
D-Latch with Clock Input

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

Function table



Graphic Symbols for Latches



- A bubble appears at the complemented output $\bar{Q}$

  Indicates that $\bar{Q}$ is the complement of Q

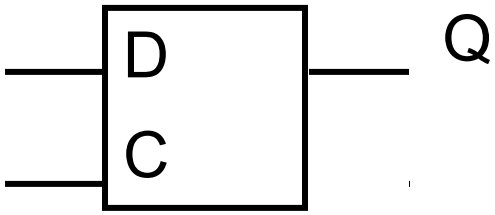
- A bubble also appears at the inputs of an $\bar{S}\,\bar{R}$ latch

  Indicates that **logic-0** is used (not logic-1) to set (or reset) the latch (as in the NAND latch implementation)

D-Latch Funtion

D-Latch：只有一个数据输入端D 称为D锁存器，也称为透明锁存器

逻辑符号



状态转移真值表

D	Q^{n+1}
0	0
1	1

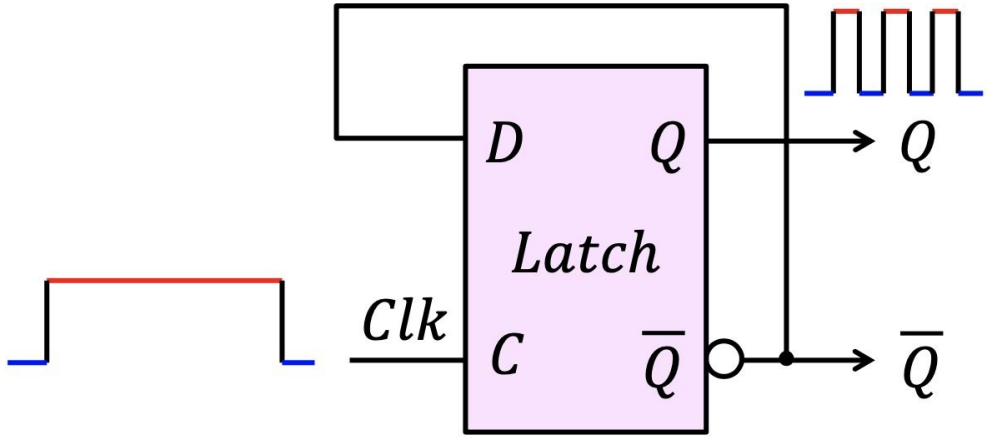
特征方程：$Q^{n+1} = D (C=1)$

Problem with Latches

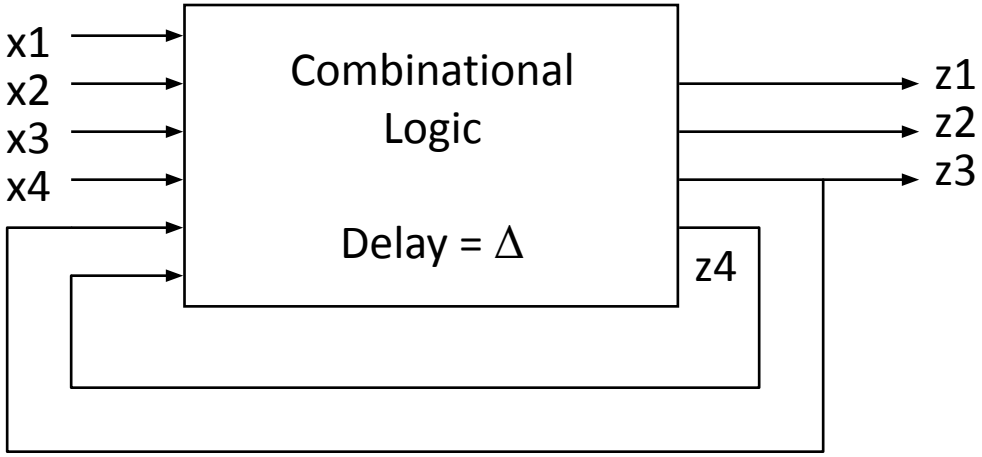
- A latch is **level-sensitive** (sensitive to the level of the clock)
- As long as the clock signal is **high** …

  Any change in the value of input D appears in the output Q
- Output Q keeps changing its value during a clock cycle
- Final value of output Q is uncertain

Due to this uncertainty, latches are NOT used as memory elements in synchronous circuits



Home work



列出表达式： z3 = ？

$z3(t+\Delta) = F(x1(t), \ldots, x4(t), z3(t), z4(t))$

Observations:

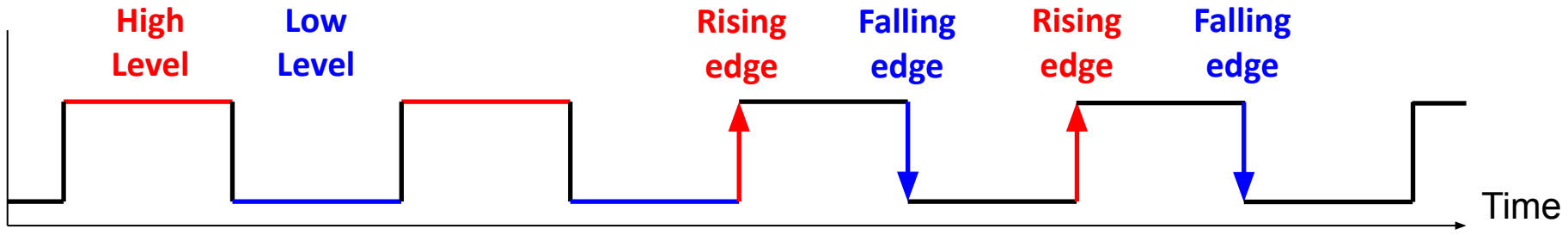
- z3 and z4 appear as both inputs and outputs.
- The "state" of variable z3 (or z4) at time $t+\Delta$ depends on its value at time t, i.e. $z3(t+\Delta) = F(z3(t))$, hence, circuit has memory.
- z3(t) and z4(t) are called state variables .

Next . . .

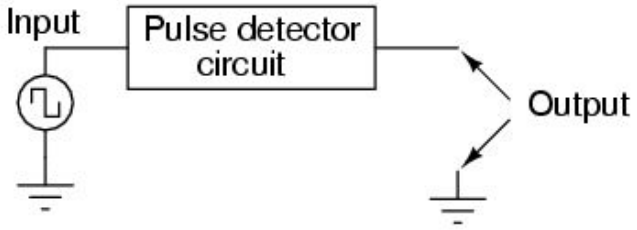
- Introduction to Sequential Circuits
- Synchronous versus Asynchronous
- Review Latches
- **Review Flip-Flops**
- **Characteristic Tables and Equations**

Flip-Flops（触发器）

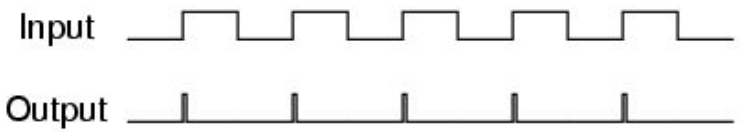
- A **Flip-Flop** is a better memory element for synchronous circuits
- Solves the problem of latches in synchronous sequential circuits
- A **latch** is sensitive to the **level** of the clock
- However, a **flip-flop** is sensitive to the **edge** of the clock
- A flip-flop is called an **edge-triggered** memory element
- It changes it output value at the **edge** of the clock

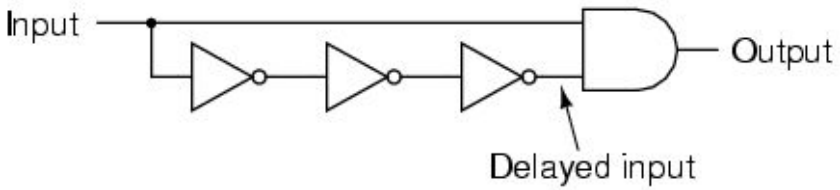


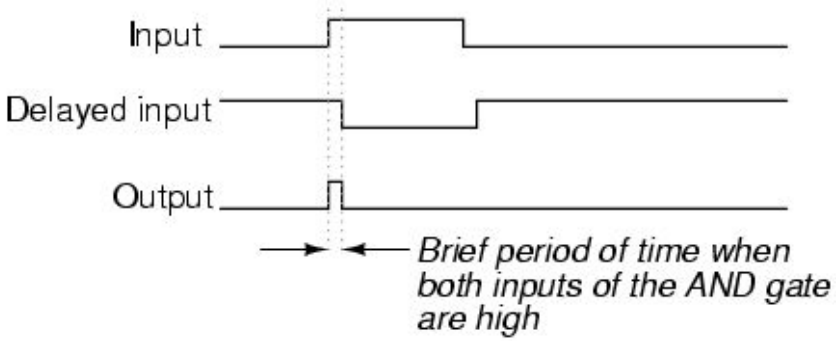
如何实现边沿触发？

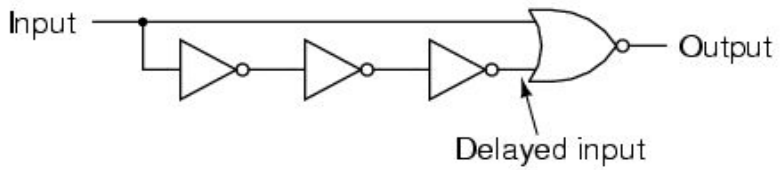


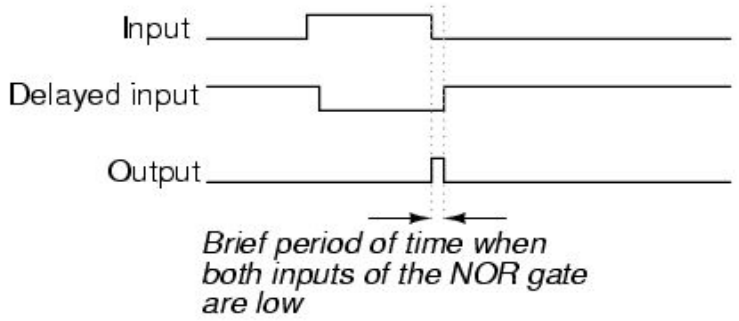
脉冲检测电路









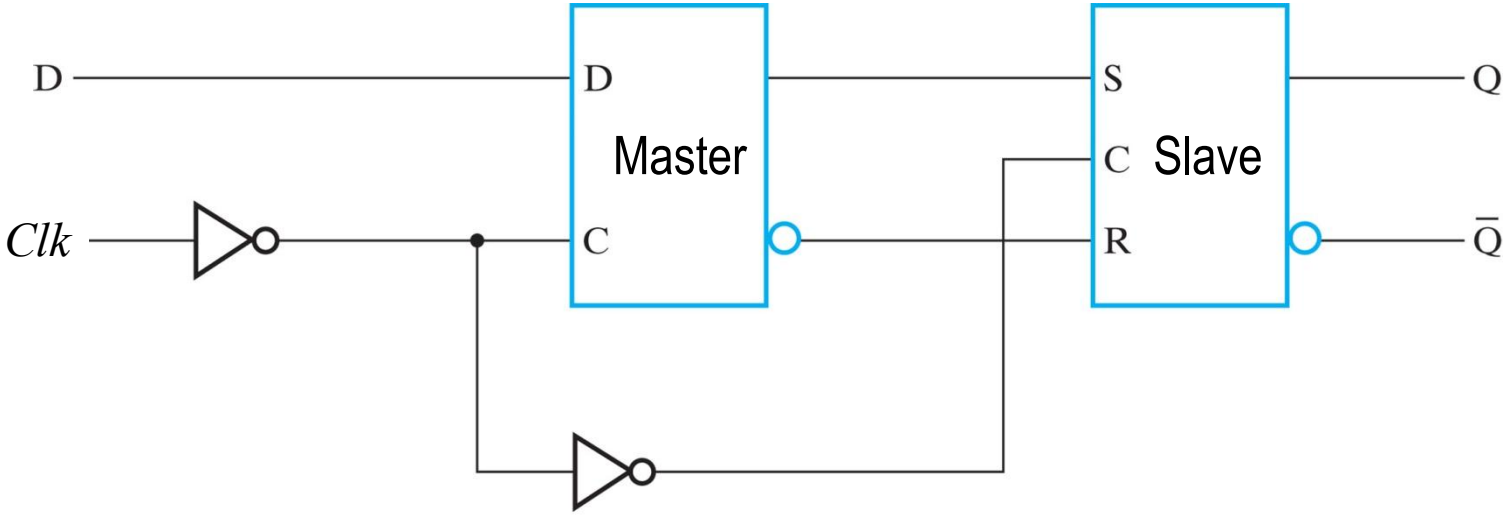


Edge-Triggered D Flip-Flop

- Built using two latches in a **master-slave** configuration
- A master latch (D-type) receives external inputs
- A slave latch (SR-type) receives inputs from the master latch
- Only one latch is enabled at any given time

When **Clk=0**, the master is enabled and the D input is latched (slave disabled)

When **Clk=1**, the slave is enabled to generate the outputs (master is disabled)



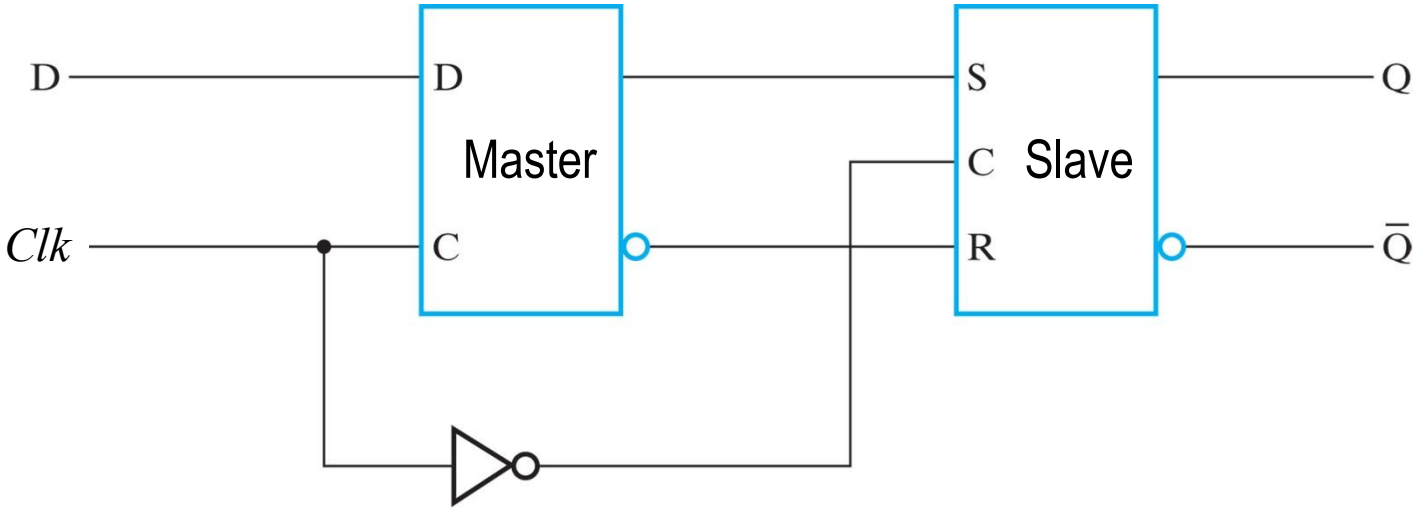
Outputs change when *Clk* changes **from 0 to 1**

Negative Edge-Triggered D Flip-Flop

- Similar to positive edge-triggered flip-flop
- The first inverter at the Master C input is removed
- Only one latch is enabled at any given time

When **Clk=1**, the master is enabled and the D input is latched (slave disabled)

When **Clk=0**, the slave is enabled to generate the outputs (master is disabled)



Outputs change when *Clk* changes **from 1 to 0**

Edge-Triggered D Flip-Flop

(1) Analyse the master-slave Flip-Flop
(2) Please create the Truth Table