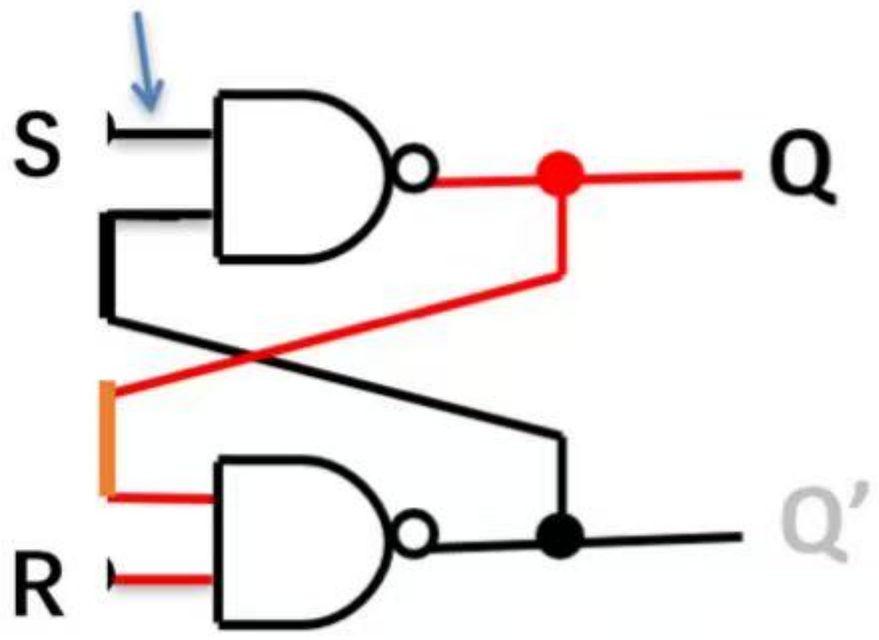
Topic 4: Building memory

Flip-flop 触发器
Register(寄存器) and Counter(计数器)

Setting an S-R flip-flop

S=0, R=1; Q=1 置位



S=1,R=0,Q=0复位

Q_0	S	R	Q
0	1	1	0
1	1	1	1
0	0	1	1
1	0	1	1
0	1	0	0
1	1	0	0

R=S=1, Unchanged
R=S=0, not allowed

A simple sequential logic

S-R flip-flop（触发器）

- This "remembers" one of 2 possible *states*
  - Two outputs Q and Q' and feedback is introduced
  - Q_n is the 'present' state; Q_{n+1} is the 'next' state(状态)
  - Q_n means without input, Q_{n+1} means with inputs

- Q depends on the S and R inputs:
  - A low pulse on **S** always make the Q=1 state (set)
    A low pulse on **R** always make the Q=0 state (reset)
  - When S or R returns to high after the low pulse, Q **stays where it is**
    - and so the flip-flop "remembers" if it is in state Q=1 or state Q=0
  - We're "not allowed" to have S and R both low simultaneously

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

Combinational logic and sequential logic
组合逻辑 时序逻辑

- The defining feature of combinatorial logic is that its outputs are **purely a function of its inputs（ALU, voter, flags…..）**

- The defining feature of sequential logic is that its outputs are **a function of its inputs and of its current outputs(Register)**
  - This involves **feedback(反馈)** of the outputs to the inputs
  - This feedback is the hook on which we hang memory

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

RS flip-flop

- Feedback is introduced
- Output Q_{n+1} is related to original Q_n
- R,S can set or reset output
- R,S=1,the output remain (keep unchanged)
- It means this bit is reserved
- Basic logic gate for sequential circuit
- Qn+1= S'+QR： state equation

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

Limitations of the S-R flip-flop

- There are two problems in using an S-R flip-flop to implement a bit in a register

1. It has distinct SET and RESET inputs
   - We'd ideally like just a **single input** that "sets" the state if it's 1 and "resets" the state if it's 0

2. We have no way of telling the flip-flop exactly **when** it should store input data
   - We'd like a "**latch**" signal for this to work in a practical system under the supervision of a control unit

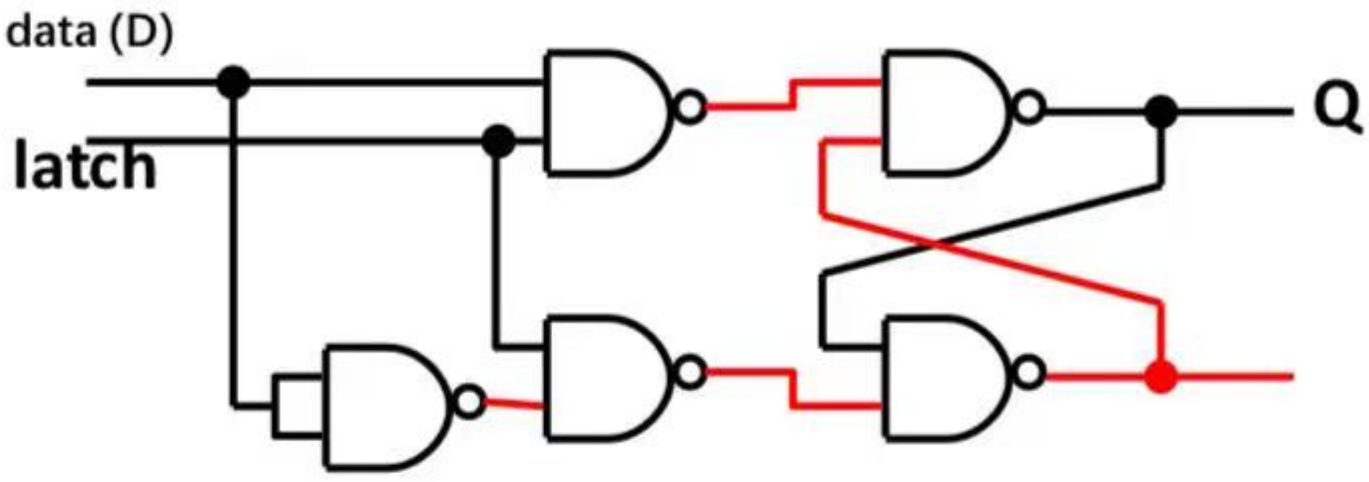
北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

D-type flip-flop

RS→D 触发器

- Add a latch as control to two gates
- D is input data
- Add a NAND Gates



Latch?

锁存

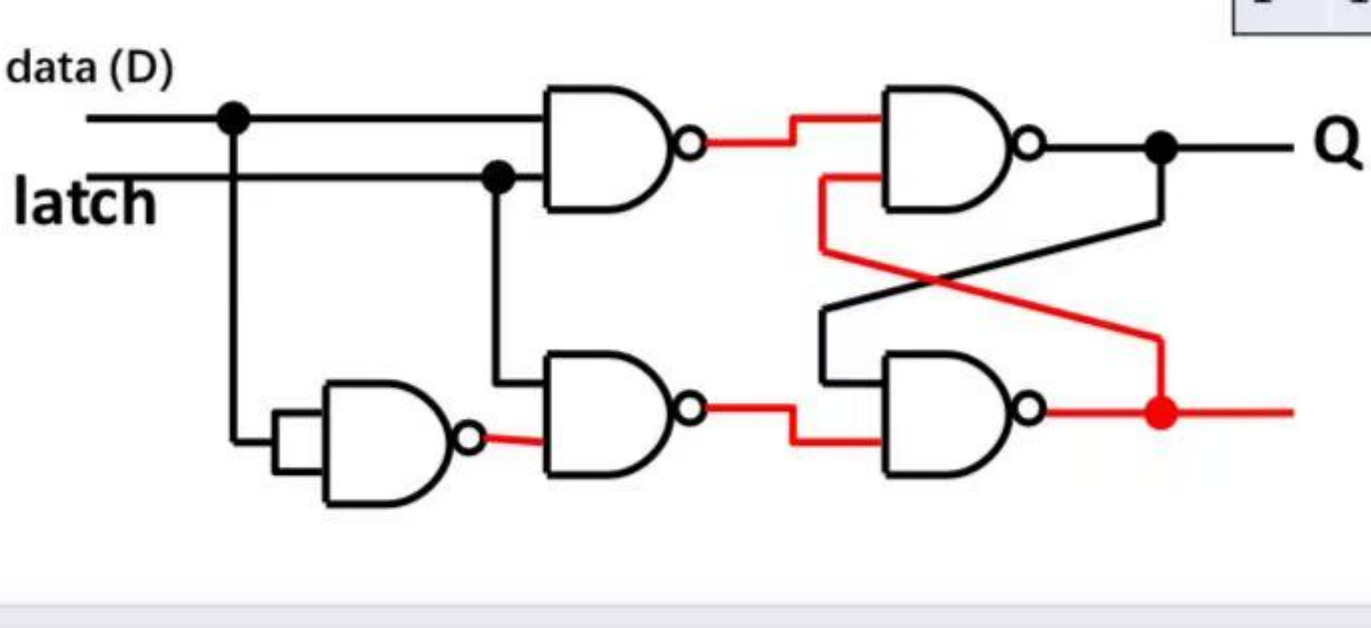
北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

D-type latch

D锁存器

Q_0	D	L	Q	
0	0	0		
1	0	0		
0	0	1		} "latch" a 0
1	0	1		
0	1	0		
1	1	0		
0	1	1		} "latch" a 1
1	1	1		



Latch?

锁存

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

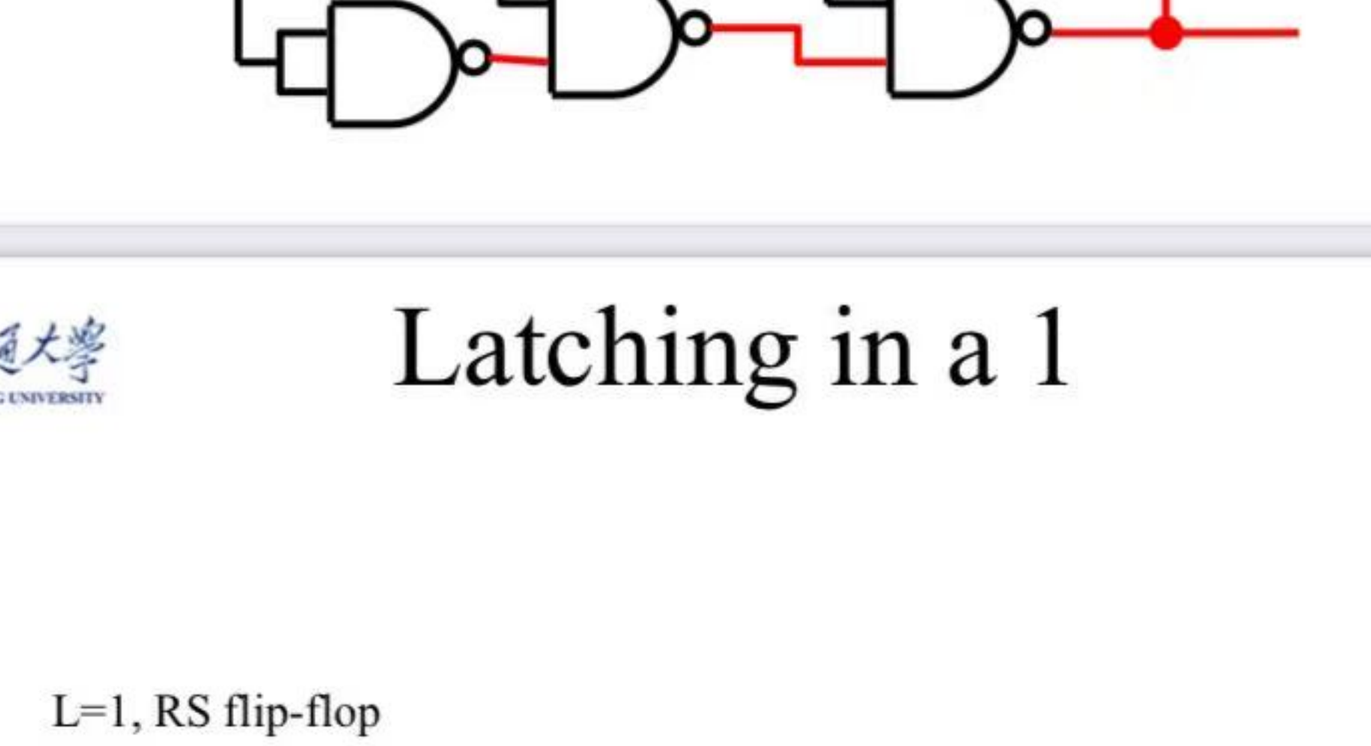
Latching in a 1

L=1, RS flip-flop

L=0 reserved

Truth table

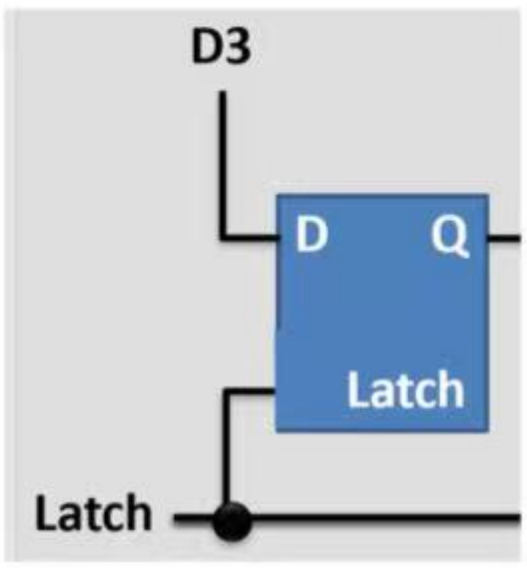
Q_n	D	L	Q_{n+1}	
0	0	0	0	} "latch" a 0
1	0	0	1	
0	0	1	0	
1	0	1	0	
0	1	1	1	} "latch" a 1
1	1	1	1	



Difference from RS flip flop

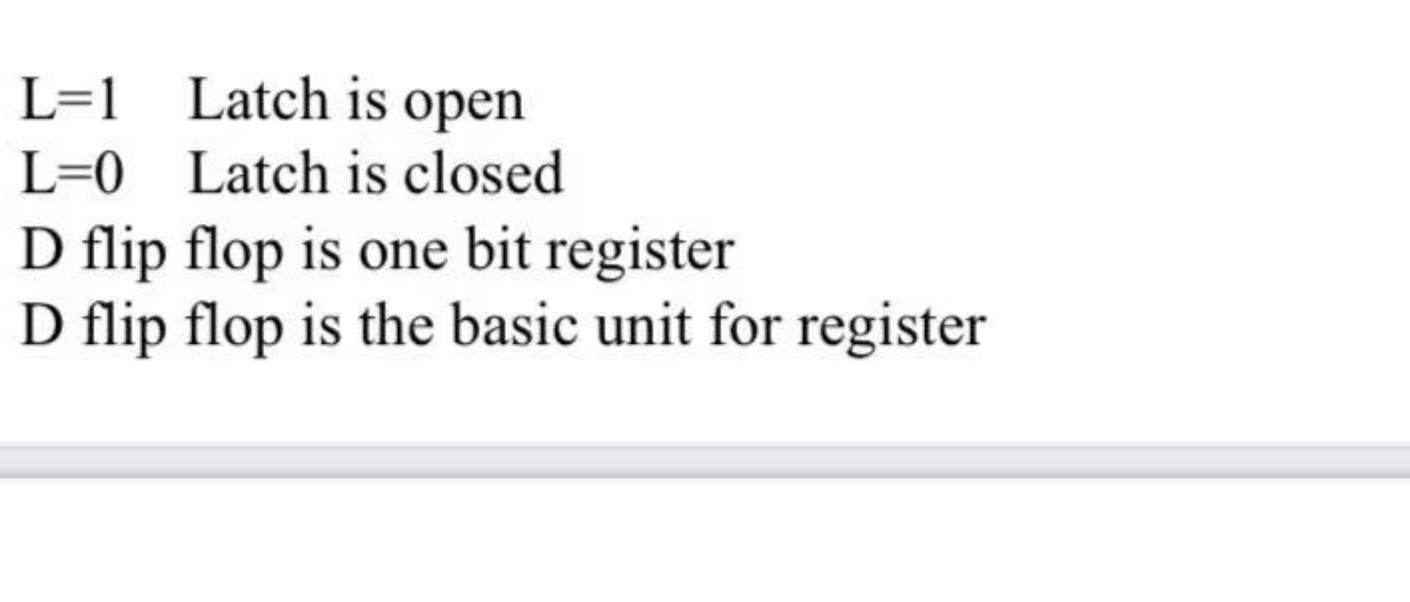
• One input	$Q_{(n+1)}=D$	L=1, Q=D
• One control	$Q=Q_0$	L=0: keep D
• Latch one bit		
• Hold the one bit in latch high		

D latch symbol



L=1 Latch is open
L=0 Latch is closed
D flip flop is one bit register
D flip flop is the basic unit for register

D flip flop D触发器

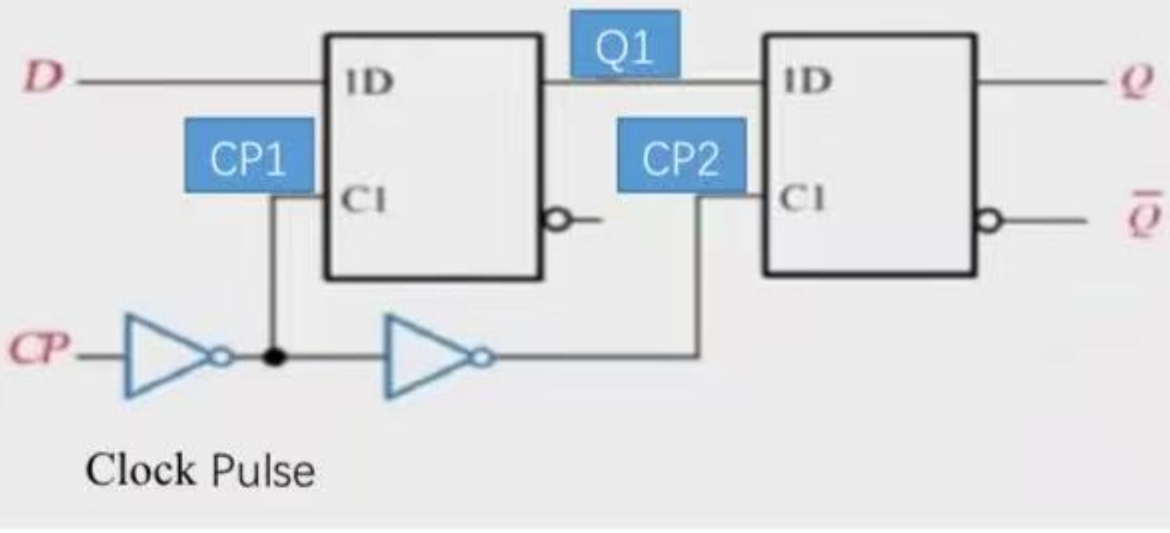


In what case, Q=D?
Benefit?

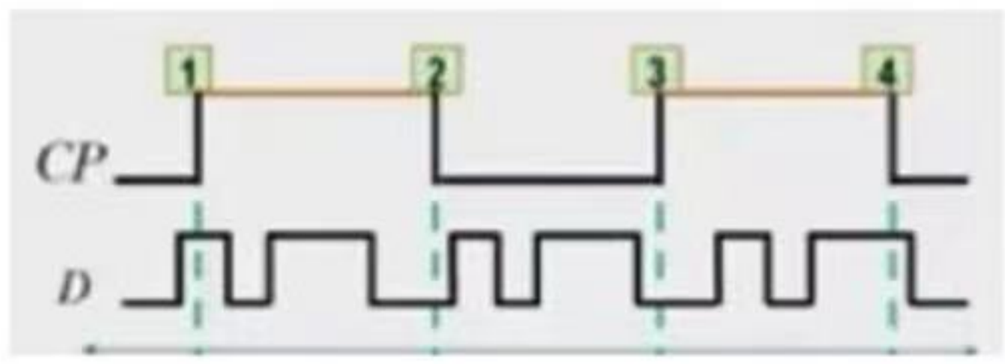
北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

Master latch Slave latch



Waveform 波形
Draw Q1, Q



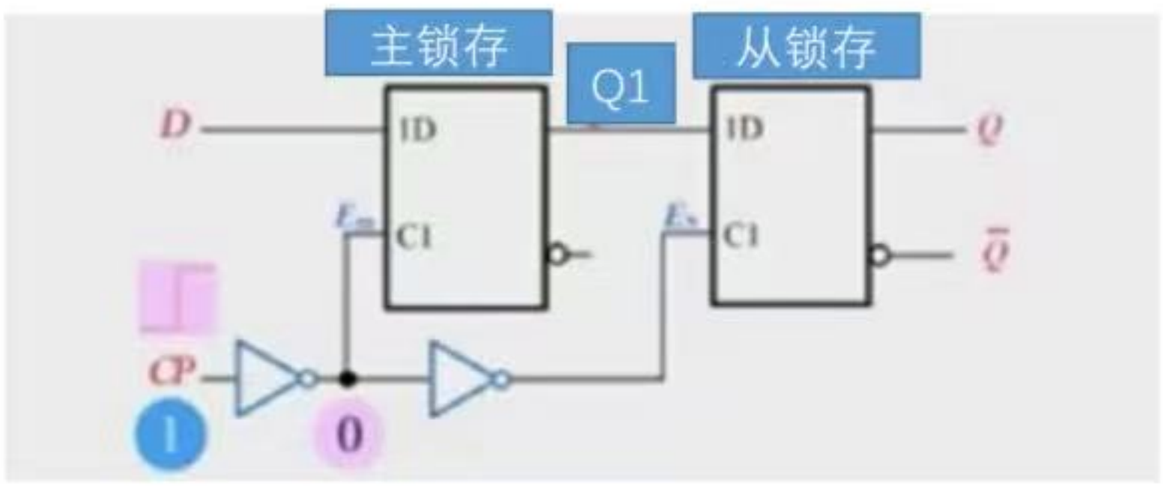
CP control to store

下降沿触发，脉冲触发

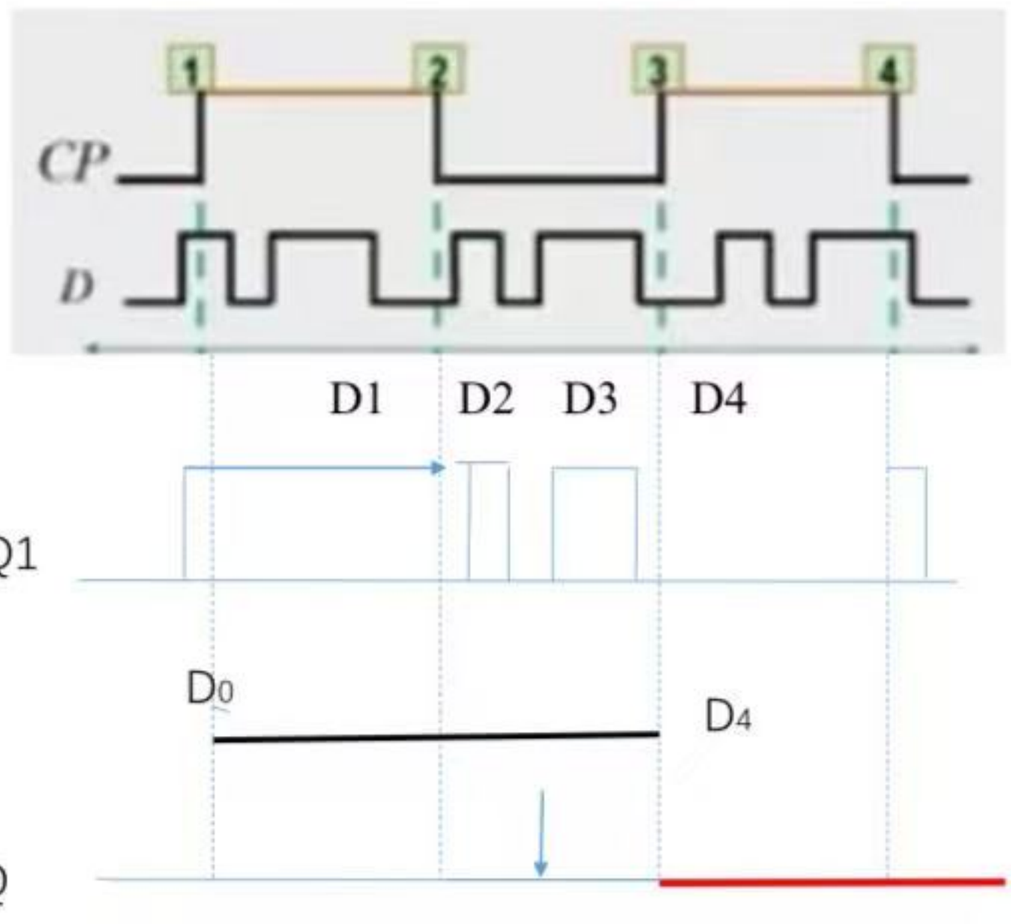
在瞬间触发，降低干扰

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University



Waveform 波形
Draw Q1, Q



D_0 is effective input
D_1 ,D_2 D_3 are interference
D_4 is effective input
Only D_1 and D_4 are reserved

Compare Q1, Q,
Q1: interference
Q: No interference
In CP edge, Q=D

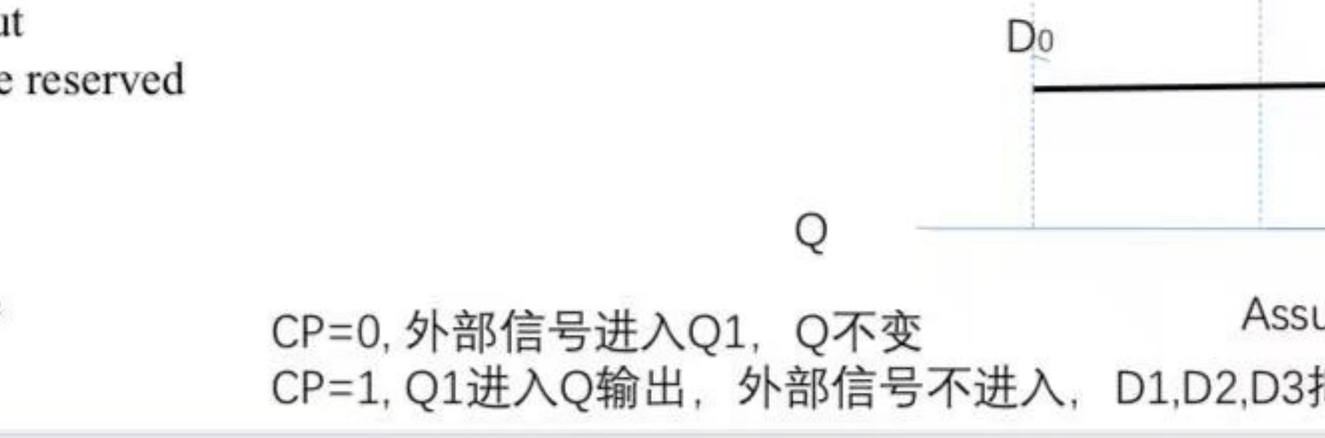
CP control to store

CP=0, 外部信号进入Q1，Q不变
CP=1, Q1进入Q输出，外部信号不进入，D1,D2,D3抑制

Assume: interference

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

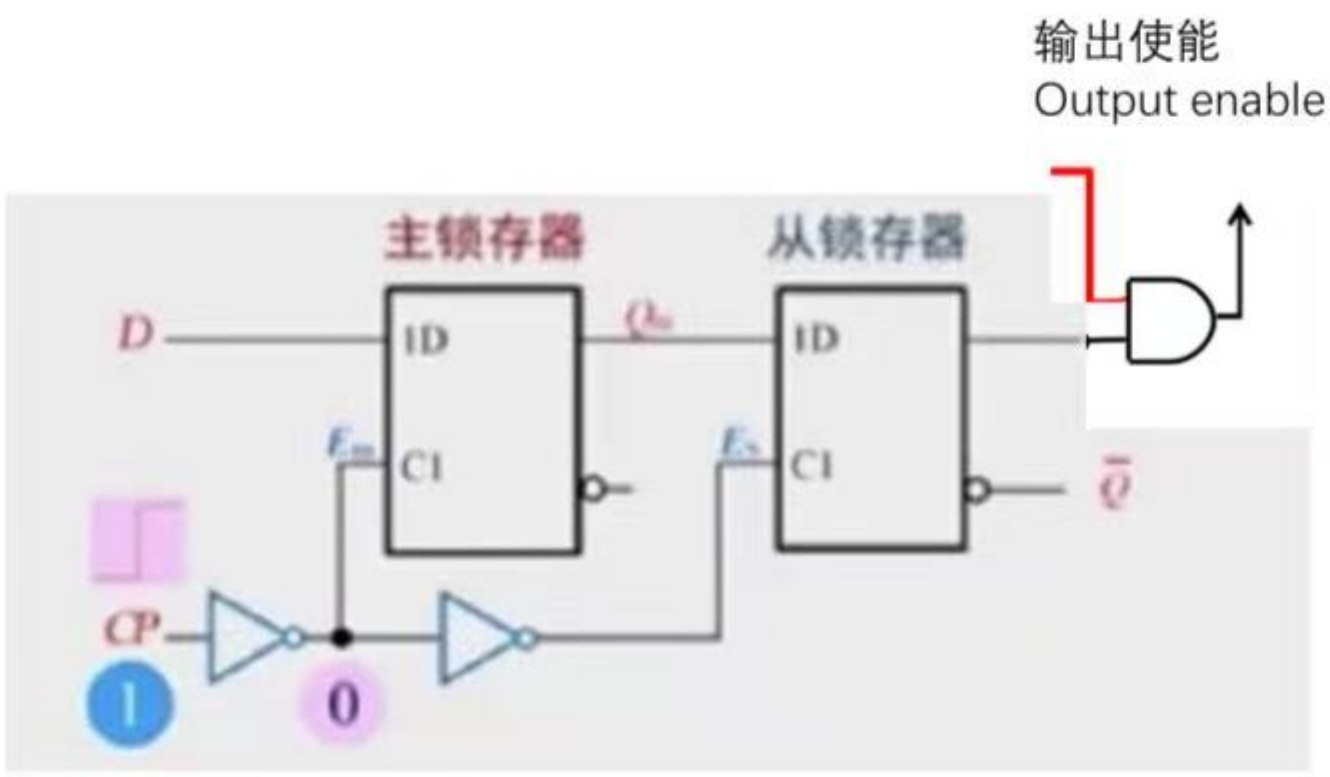


CP rising edge triggering

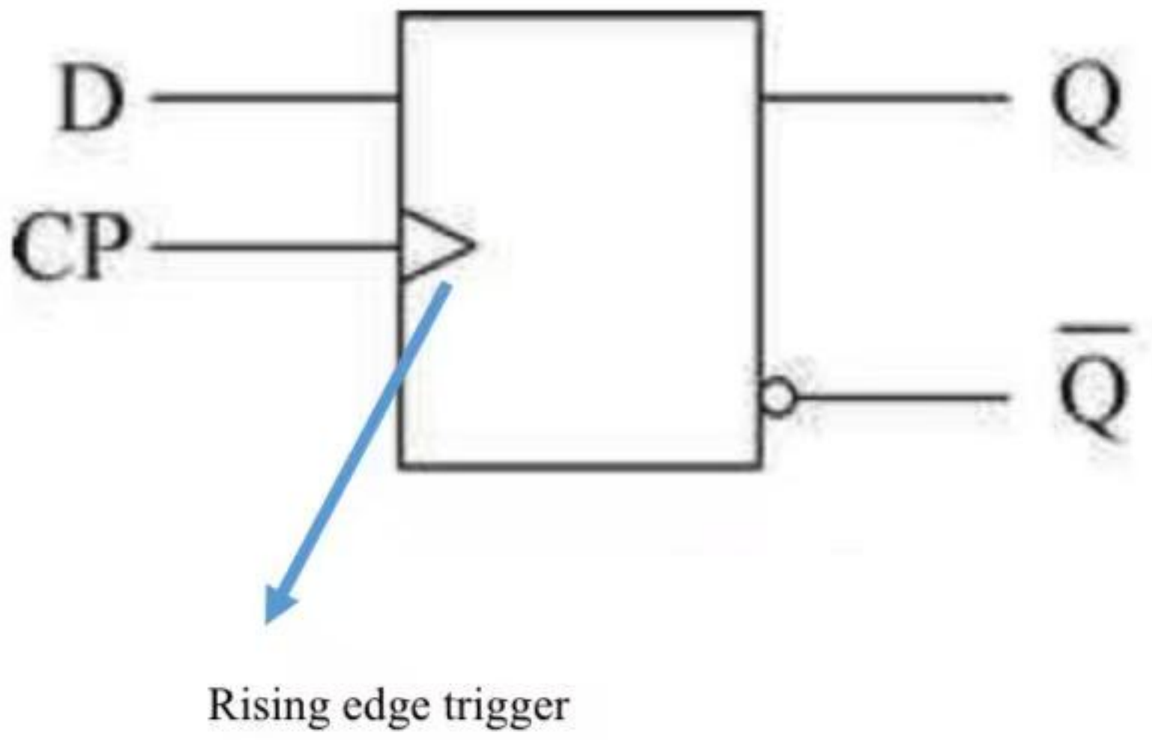
CP=0, Q1=D, Q remain; CP=1, Q1 remain, Q=D and remain

Q =D at rising edge

Advantage: anti-interference

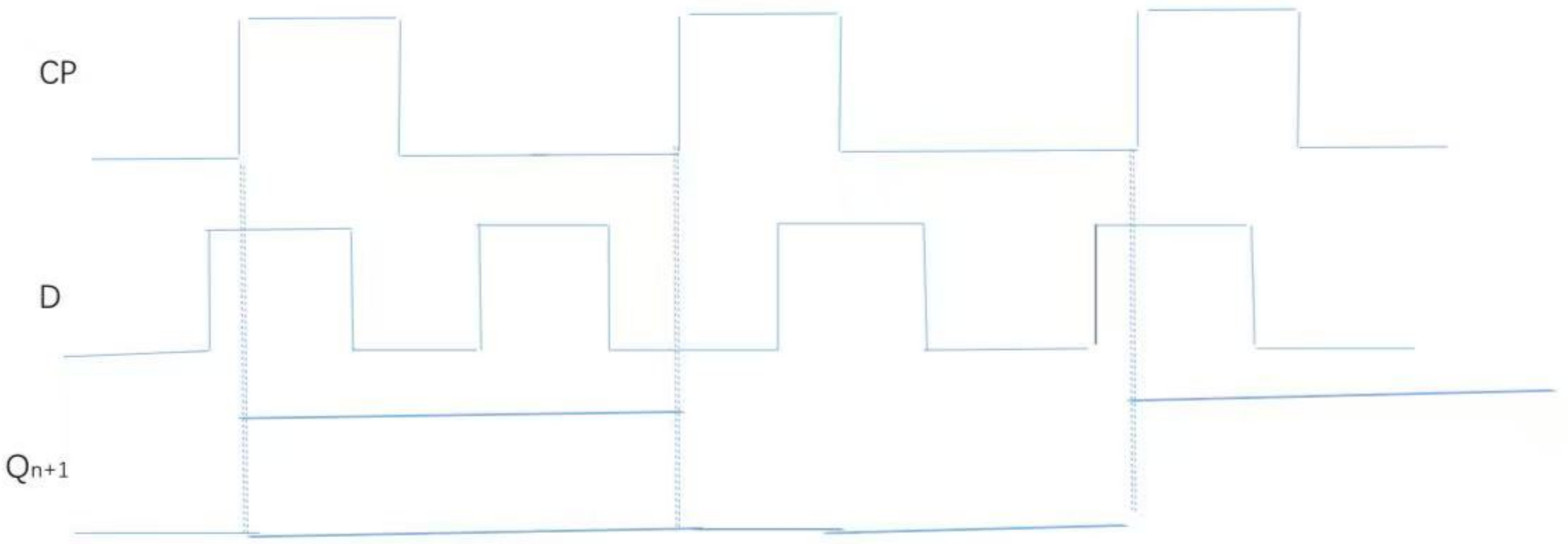


D flip flop



Rising edge trigger

Waveform for edge triggering D flip-flop



(Q_n=0)

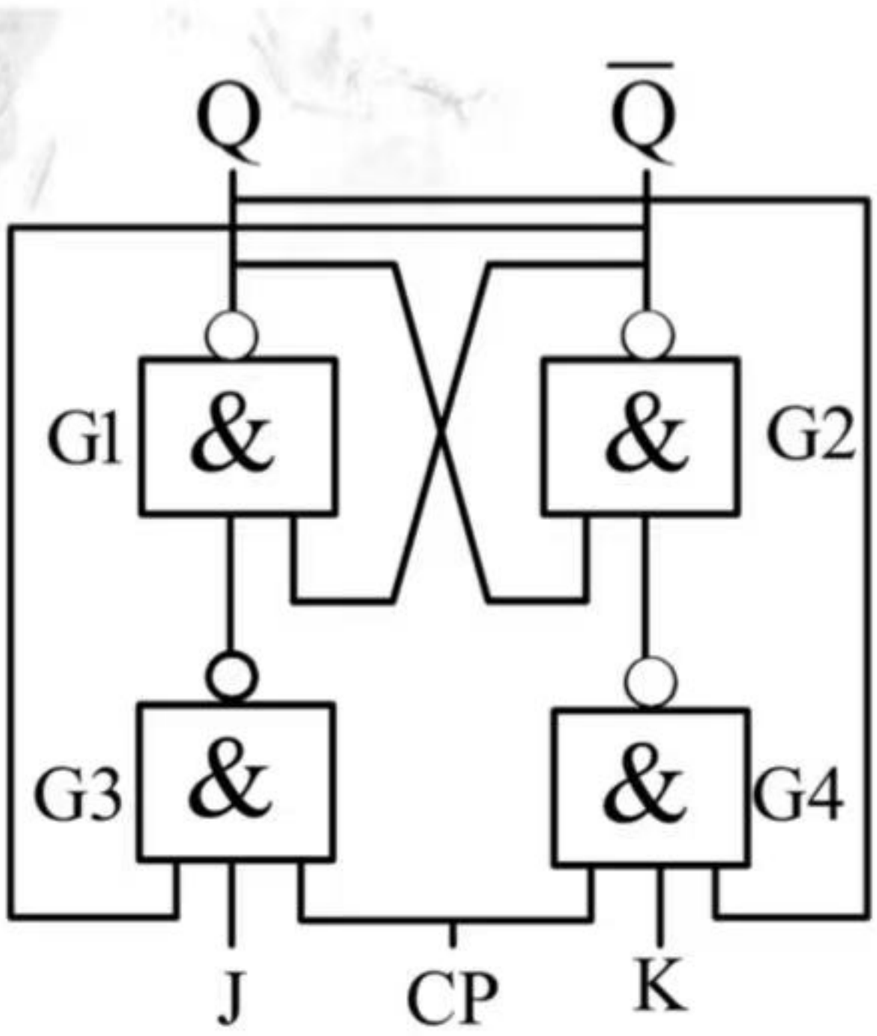
Draw Q_{n+1}

What tools are used?

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

JK flip-flop

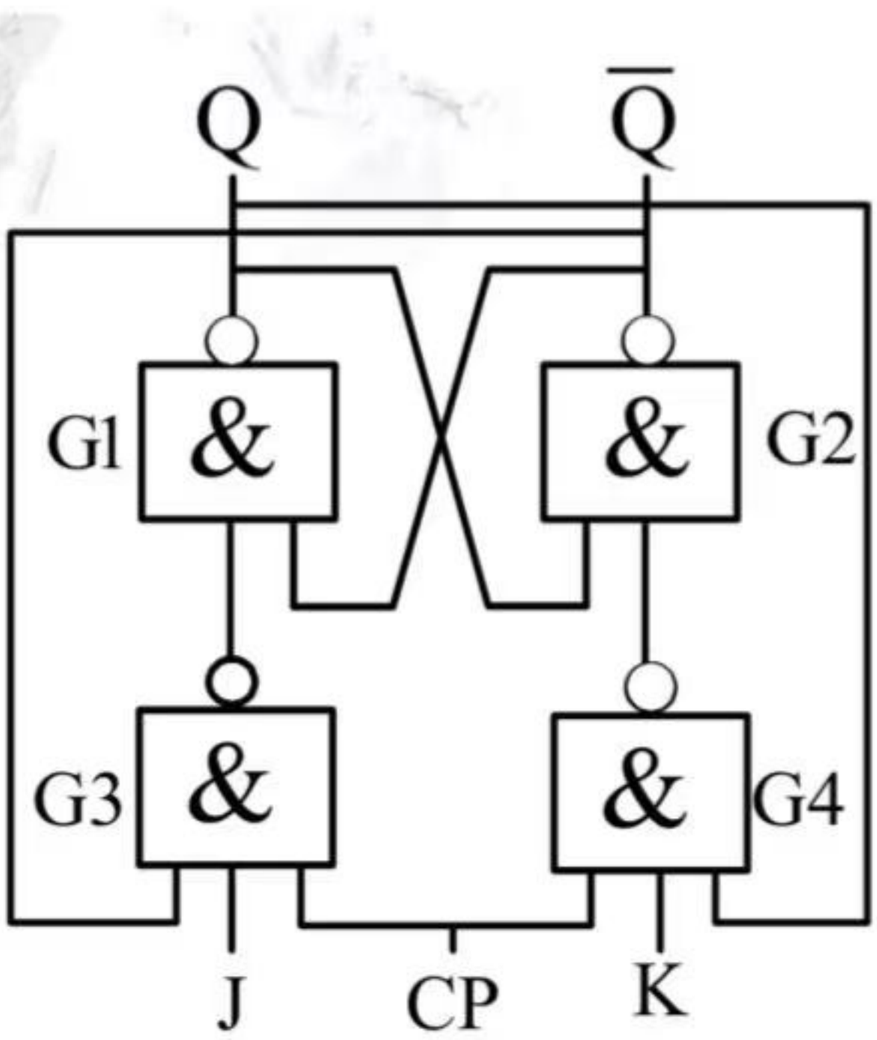


CP=1,J=K=0,	Q remain	Q' remain
CP=1, J=0,K=1	Q=0,	Q'=1
CP=1, J=1,K=0	Q=1,	Q'=0
CP=1, J=1,K=1	Q'	Q

北京交通大學 BEIJING JIAOTONG UNIVERSITY

Lancaster University

JK flip-flop

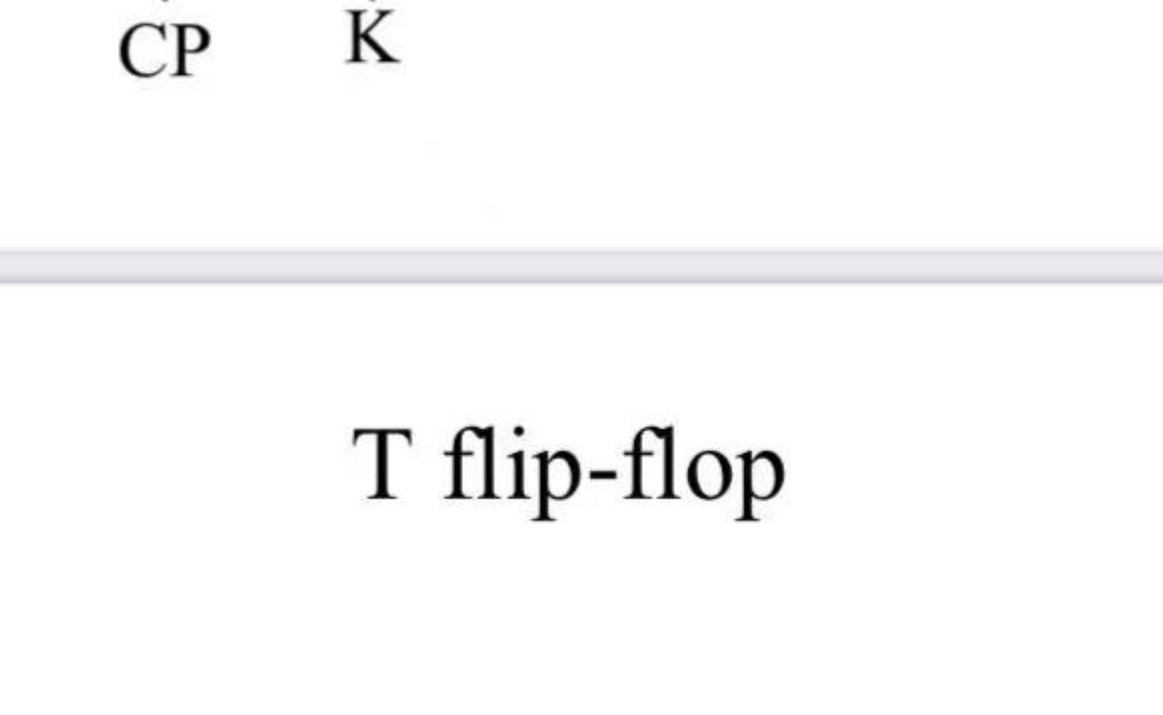


CP=1,J=K=0,	Q	Q'
CP=1, J=0,K=1	Q	Q'
CP=1, J=1,K=0	Q	Q'
CP=1, J=1,K=1	Q	Q'

北京交通大學 BEIJING JIAOTONG UNIVERSITY

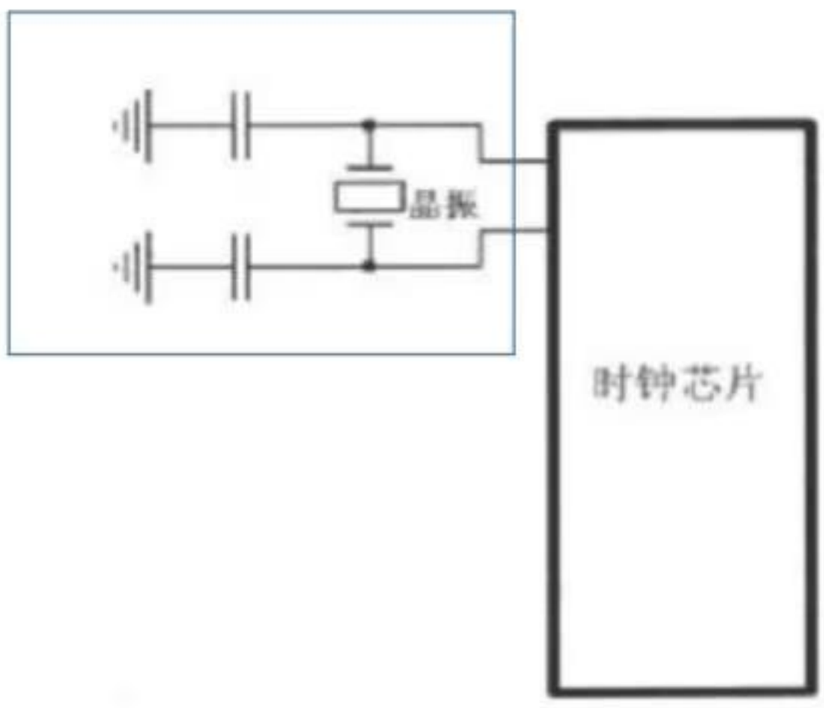
Lancaster University

T flip-flop



T	Qn+1
0	Qn
1	~Qn

Where is CP from?



Clock pulse circuit

Where is CP to?

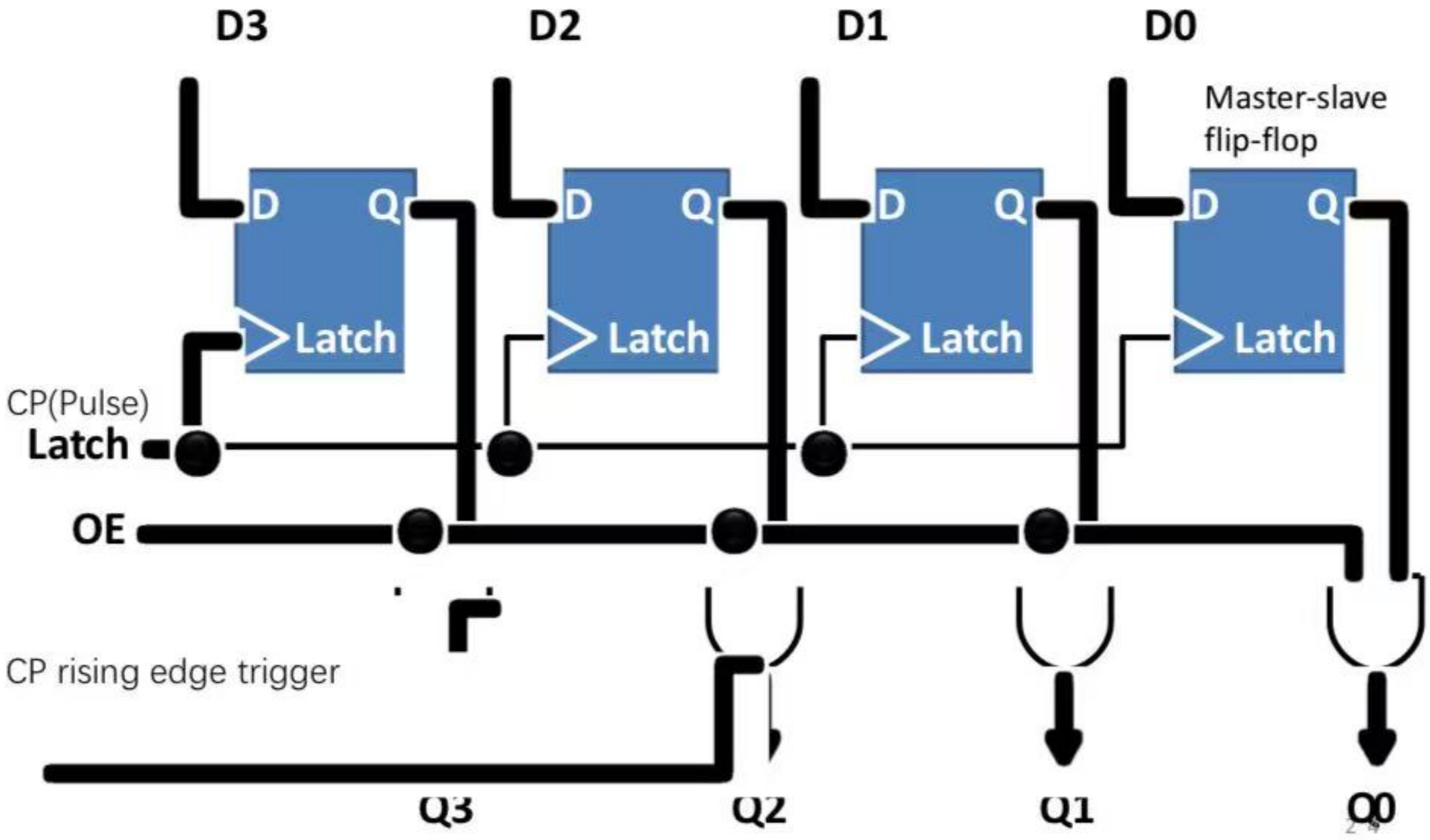
- CP is control signal
- CP is input to control unit of each computer components, such as CPU CU, memory CU, other controller.
- CU is the core of CPU

北京交通大學 BEIJING JIAOTONG UNIVERSITY
Lancaster University

How to build register?

北京交通大學 BEIJING JIAOTONG UNIVERSITY
Lancaster University

Implementing a register using multiple flip-flops



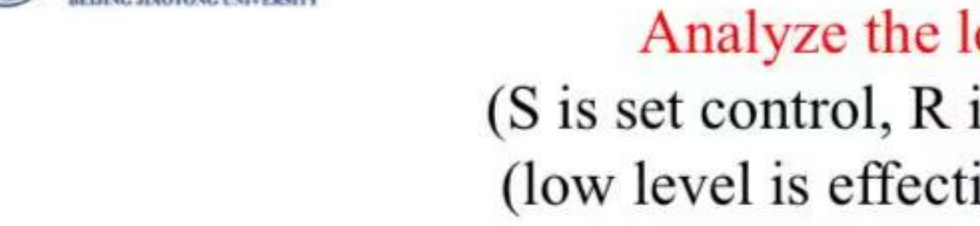
北京交通大學 BEIJING JIAOTONG UNIVERSITY
Lancaster University

Draw the sequential diagram

Analyze the logic gate function

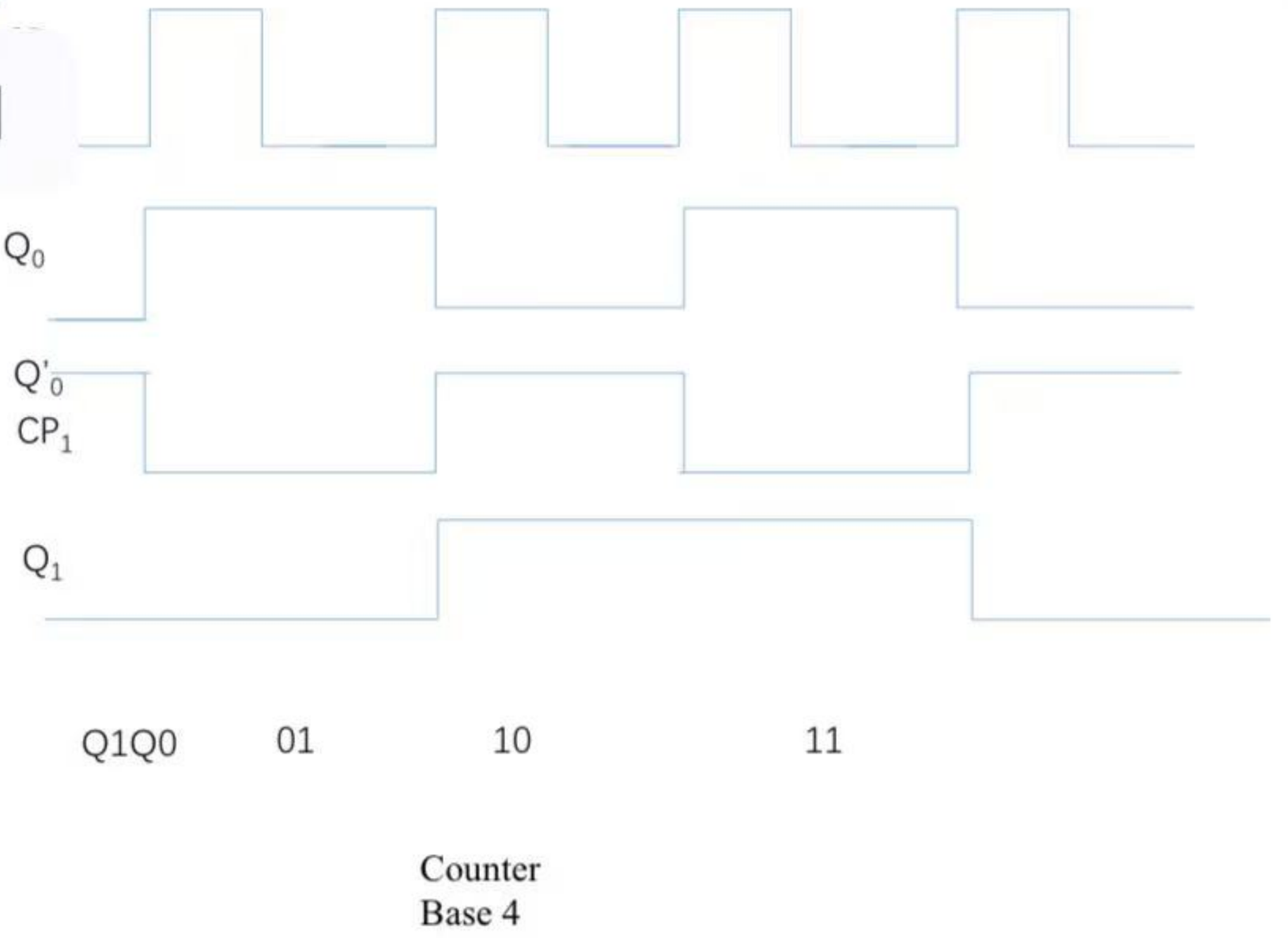
(S is set control, R is reset control, assume $Q_0=0$, $Q'_0=1$)

(low level is effective)



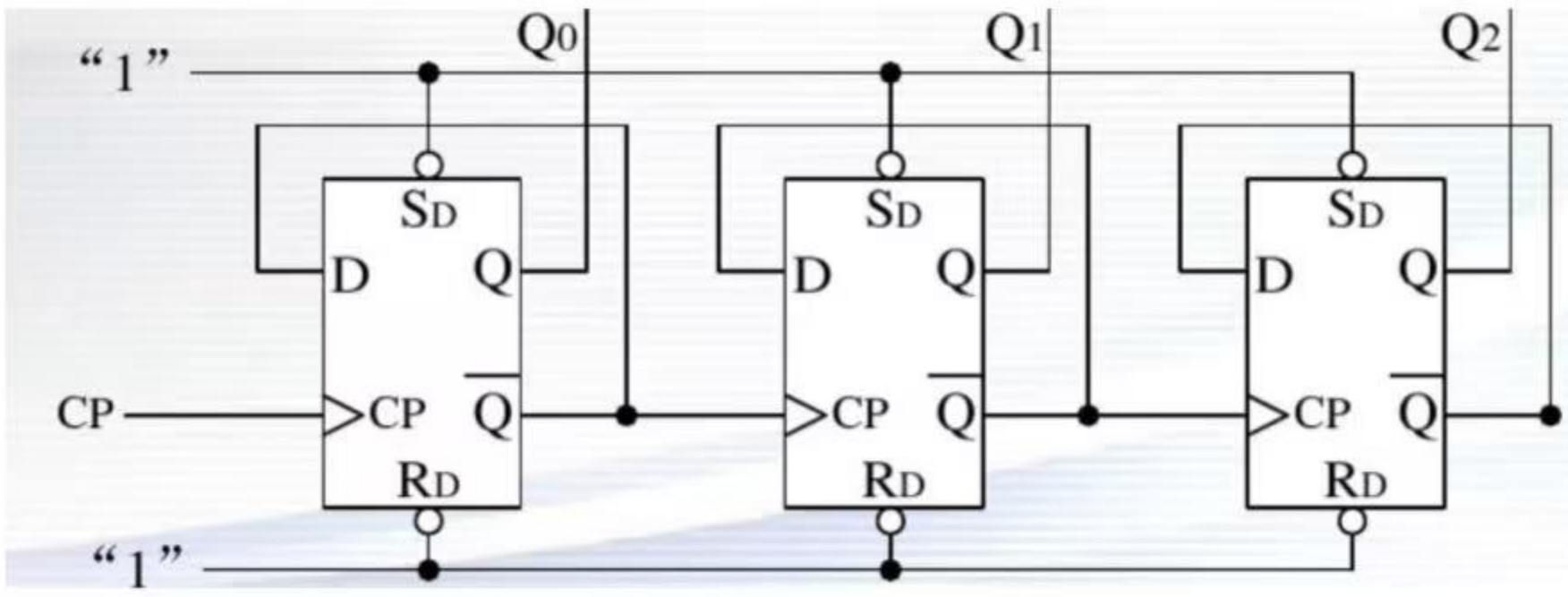
Write logic status equation

Draw sequential waveform



Counter
Base 4

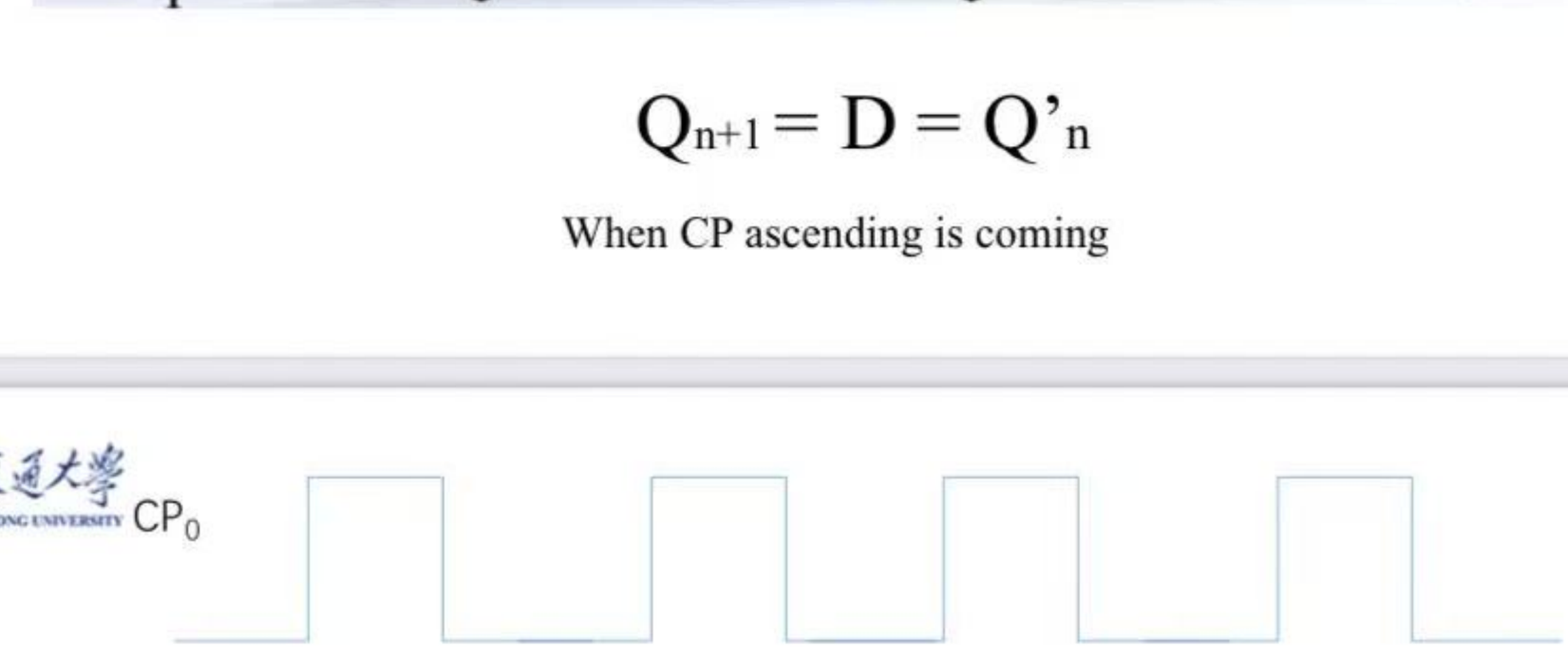
Assume $Q_{0,1,2}=0$, $Q'_{0,1,2}=1$



Register?
base?

$$Q_{n+1} = D = Q'_n$$

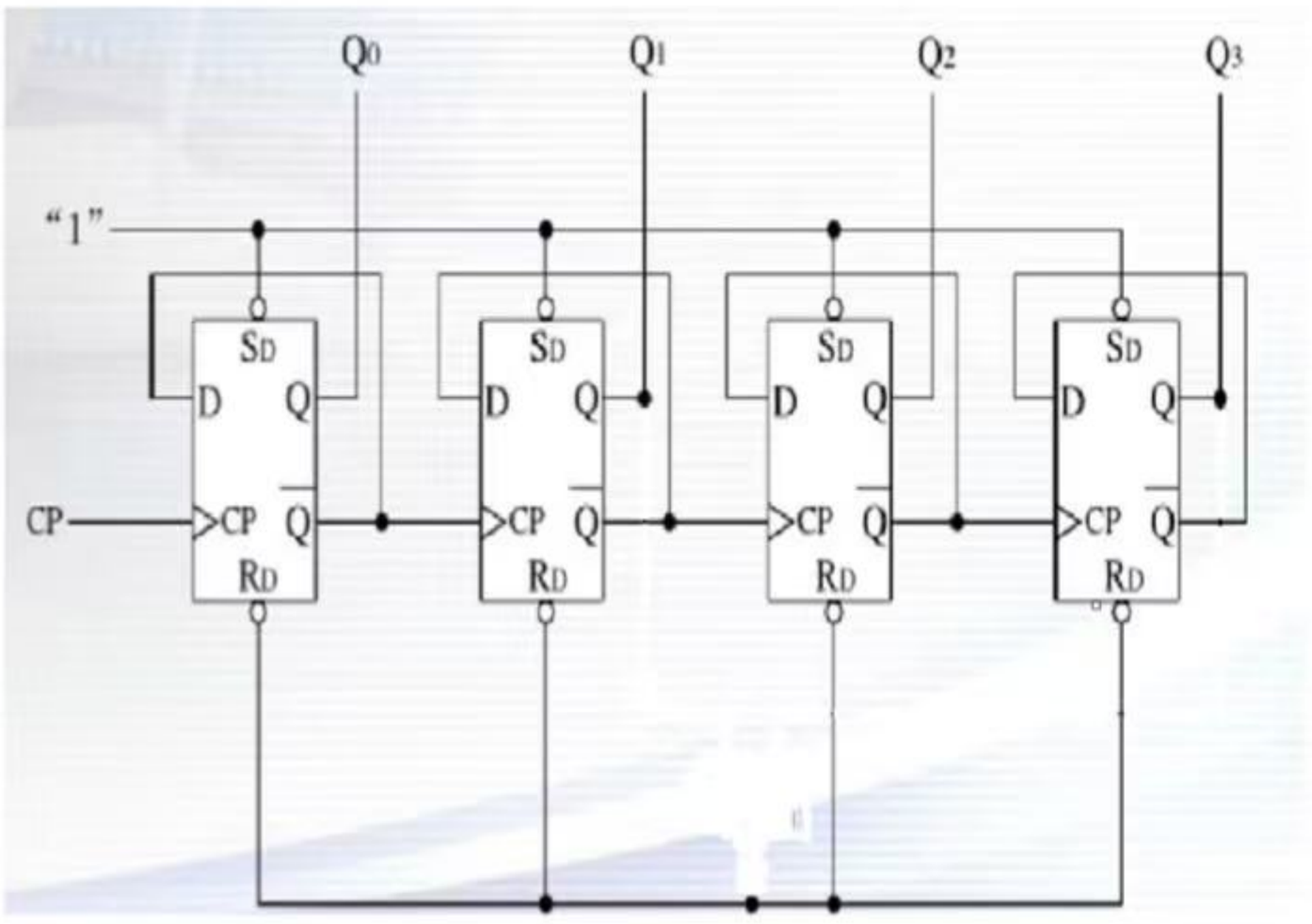
When CP ascending is coming



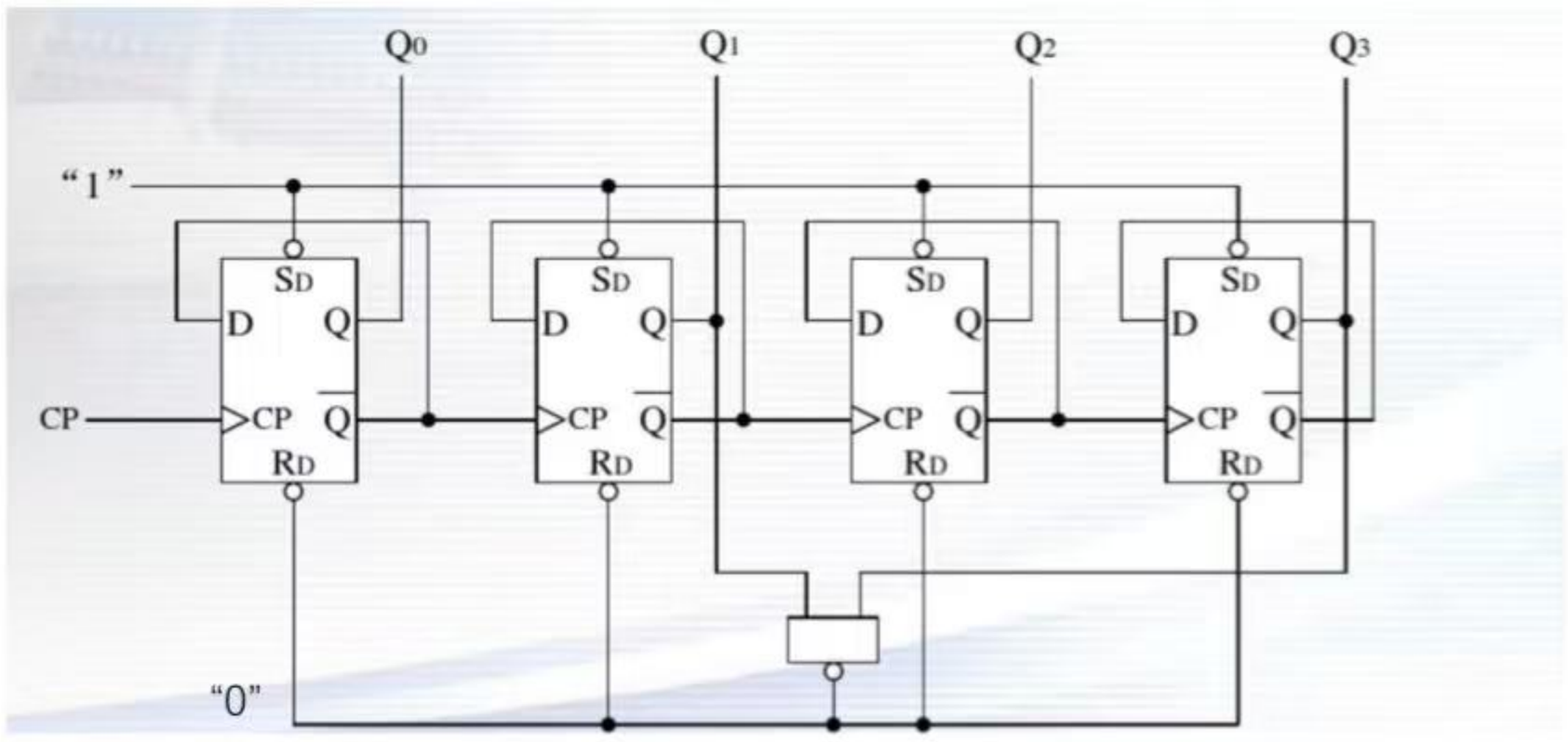
开始Q1Q0为0
Q0为CP1

Countdown counter

Base?

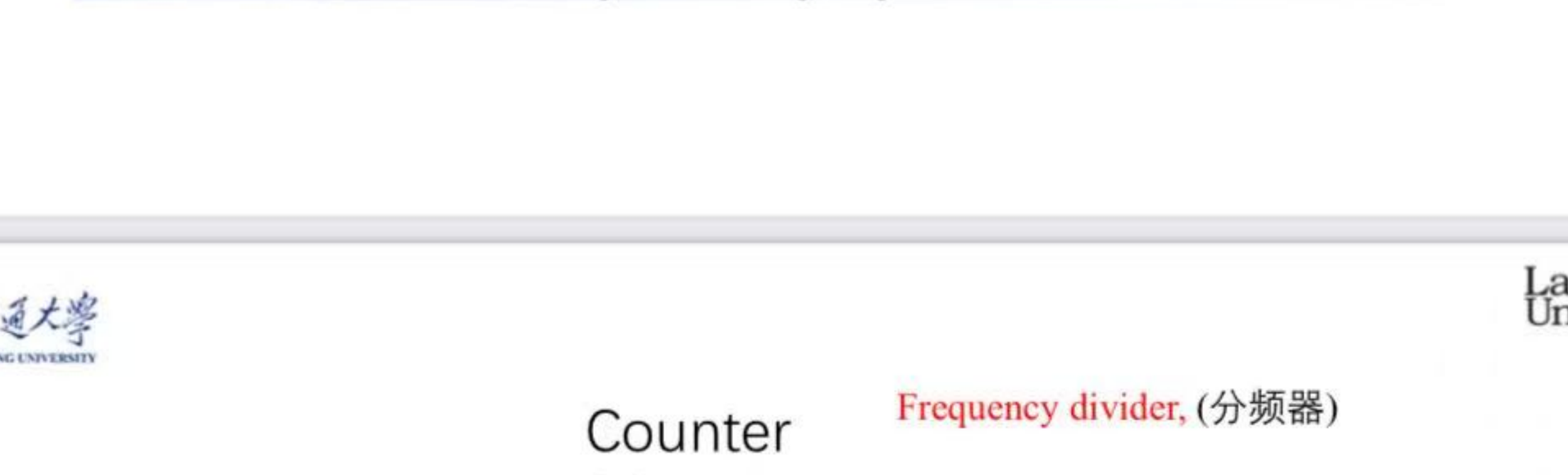


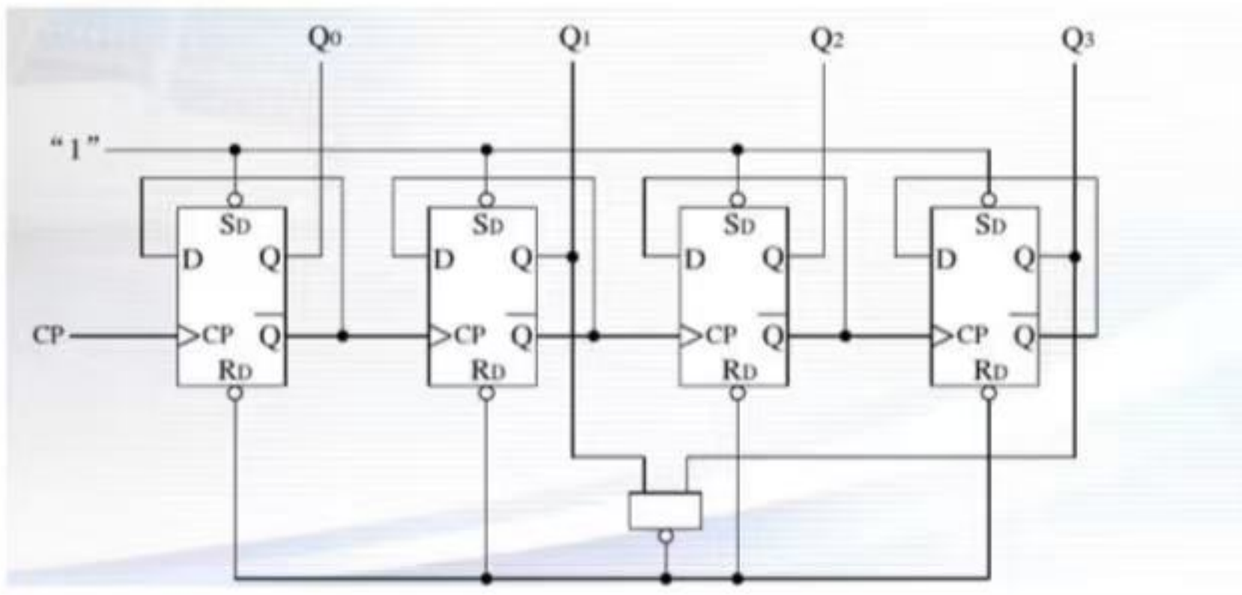
Draw the sequential diagram
Analyze the logic gate function
(S is set control, R is reset control, assume Q_0=0)



Counter
(Timer)

Frequency divider, (分频器)





Frequency divider
(分频器)

Generate different frequency signal

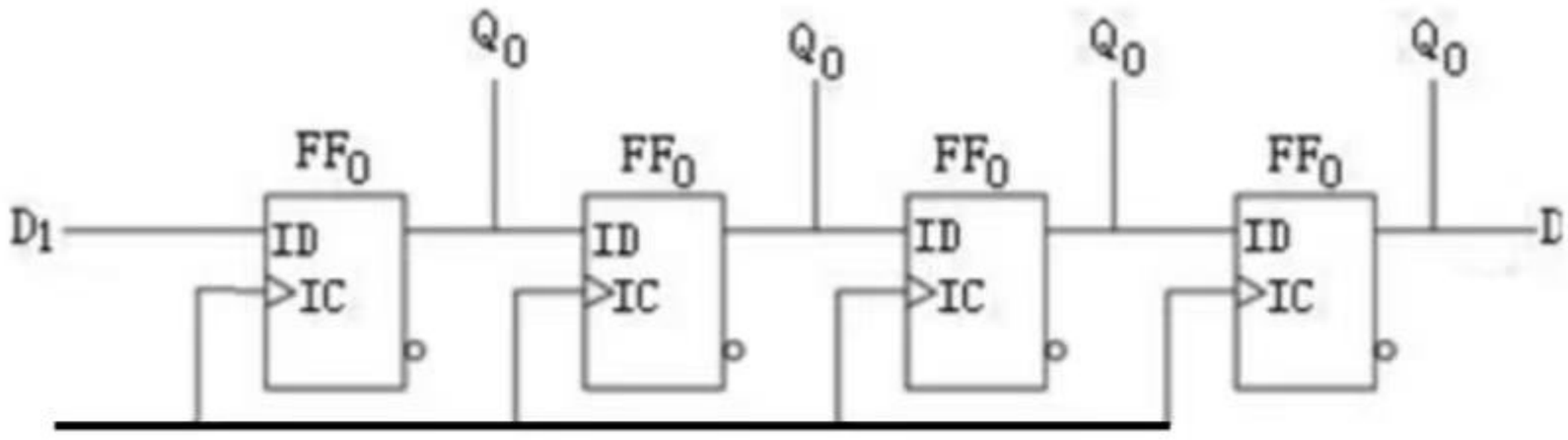
Used for different CU control

Counter
(计数器)

Cp1:Q0=1
Cp2: Q1=1
Cp3: 11
Cp4: Carry out

Frequency meter

Shift right register

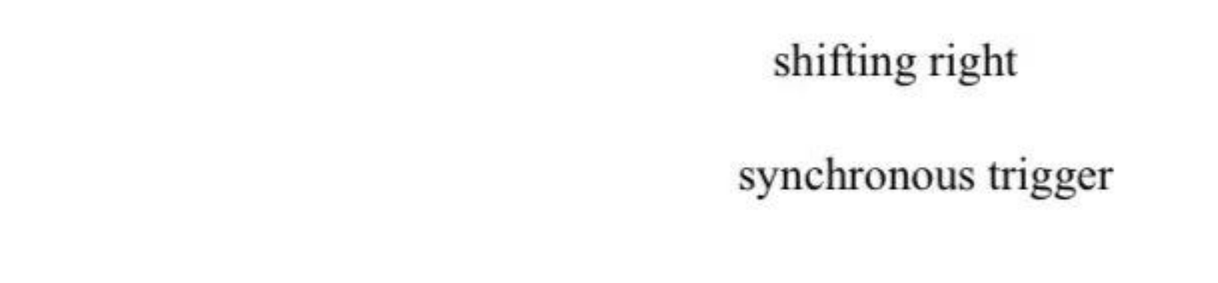


Serial input D1 and parallel out

shifting right

synchronous trigger

Loop shift register



Loop

	F4	F3	F2	F1
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

Initial state: F4F3F2F1:1000

Buses: connecting the blocks

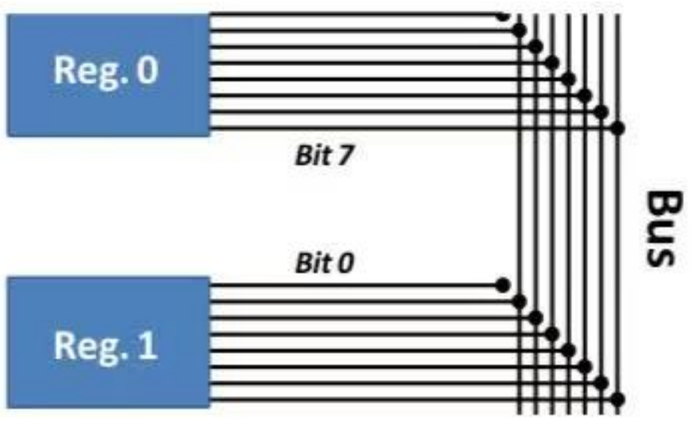
- Buses are the "glue" of the computer architecture
  - They connect the elements of the von Neumann architecture: the ALU, the control unit, memory chips, and I/O devices
  - They are essentially bundles of **wires**, one for each bit

- There are three types of bus

  - An *address bus* runs between the control unit and main memory
    - Used to tell main memory to access a specific address (whether for reading or writing)
    - The "width" of the address bus corresponds to the amount of addressable memory
      - Examples: 16 bits → 65536 bytes (2^{16}); 32 bits → 4 GB (2^{32})
  - *Data buses* carry data around the computer
    - An "n-bit" processor will have a data bus *n* bits wide (e.g. n=8 or 32 or 64)
      - This means we can read/ write *n* bits from/ to memory in one go
    - Some processors have distinct internal and external data buses
      - External bus may be narrower to reduce the number of external connections/ pins, and thus cost
  - *-- Control bus*

Data buses: connecting to registers Explanation

- Bus wires are *shared*



… so we must ensure that there is only one active output at a given time?

A job for **output enable**…

Summary

- We know the difference between combinatorial and sequential logic
- We know how logic components can be used to build *one bit* of static storage (static memory)flip flop

  we know how these bits can then be combined into multi-bit Registers and counter
- We understand how the elements of the computer architecture can be glued together using buses

Exercise

1： Draw out the D flip-flop symbol, explain the meaning of CP, D, Sd, Rd, Q, Enable.

2: Draw base-6 counter logic circuit

Draw the waveform of base_6 counter

3. Design any number system counter(任意进制，如54进制，前50个10进制，后4个位5进制)

Exercise after class

- Design 60 base counter

- Design a timer(counter based on subtraction)