北京交通大学 BEIJING JIAOTONG UNIVERSITY

Lancaster University

Topic 4: Building memory

Flip-flop 触发器

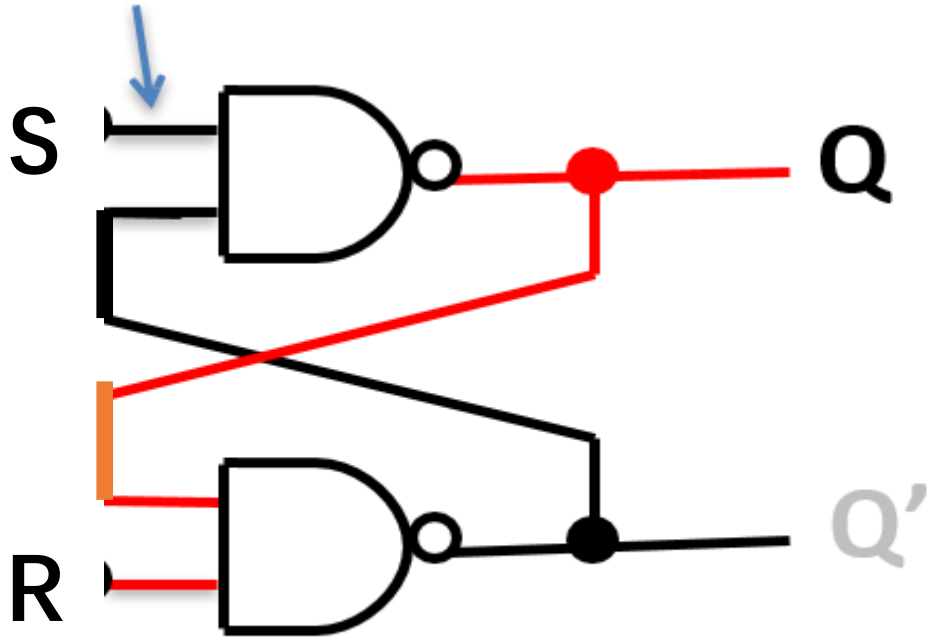
Register(寄存器) and Counter(计数器)

Combinational logic and sequential logic

- Combinatorial logic is that its outputs are purely a function of its inputs（ALU, voter, flags…..)

- Sequential logic is that its outputs are **a function of its inputs and of its current outputs(Register)**
  - This involves **feedback** of the outputs to the inputs
  - This feedback is the hook on which we hang memory(output remain)

S-R flip-flop



Q_0	S	R	Q	
0	1	1	0	
1	1	1	1	R=S=1, Unchanged
0	0	1	1	
1	0	1	1	S=0, R=1; Q=1 set
0	1	0	0	
1	1	0	0	S=1, R=0; Q=0 reset

R=S=0, not allowed

RS flip-flop

Feedback is introduced, Output Qn+1 is related to original Qn

R,S can set or reset output

R,S=1,the output remain (keep unchanged), It means this bit is reserved

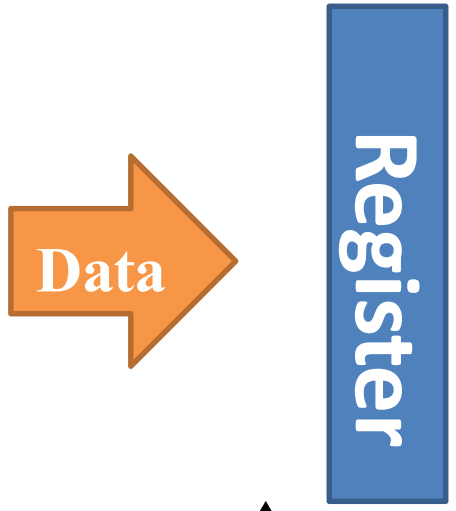
Basic logic gate for sequential circuit(state equation): Qn+1= S'+QR :

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

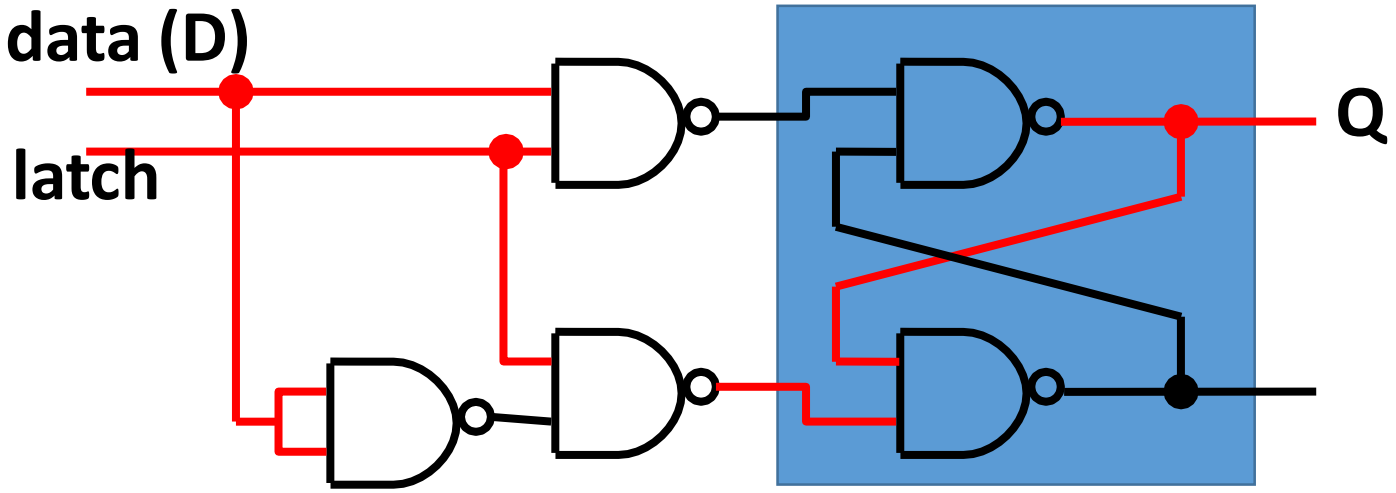
2. We have no way of telling the flip-flop exactly **when** it should store input data
   - We'd like a "**latch**" signal for this to work in a practical system under the supervision of a control unit



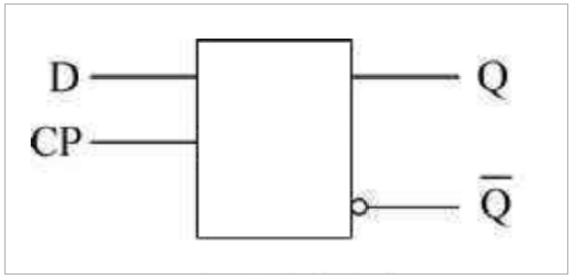
Add a latch as control to two gates , D is input data

latch(锁存)
(Store input data ***now***)

D-latch



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



D latch symbol

Difference from RS flip flop

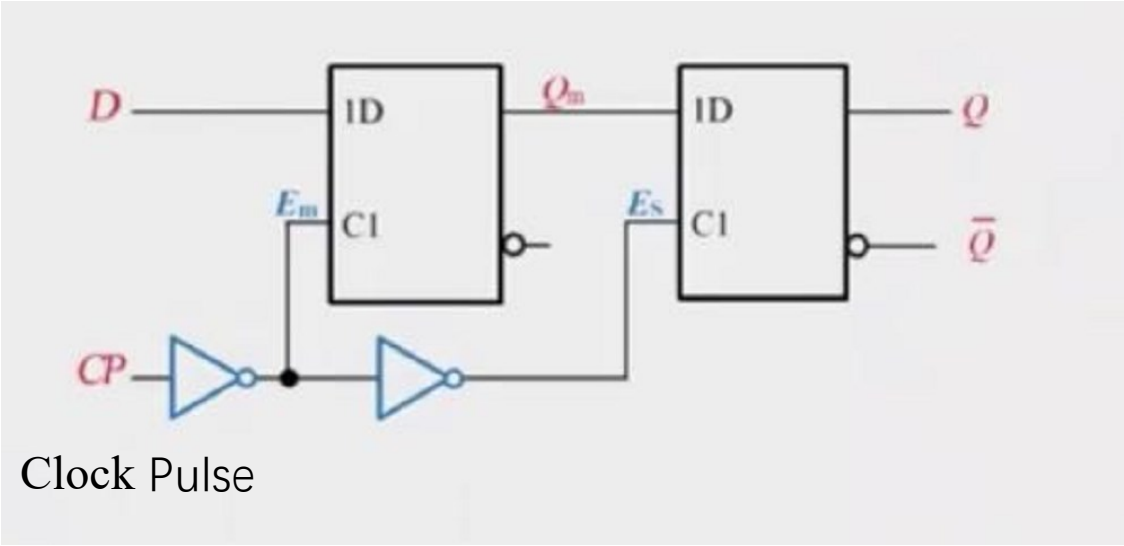
- One input
- One control (Latch)

$L = 1, Q_{(n+1)} = D$

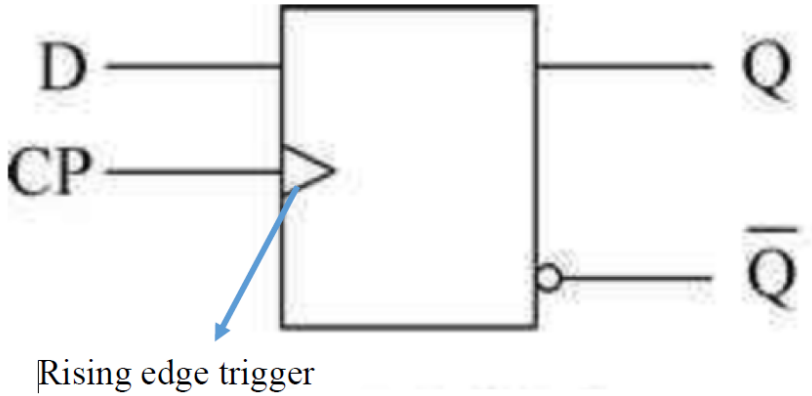
$L = 0, Q = Q_0$

D flip flop

Master latch Slave latch



Clock Pulse



In what case, Q=D? Benefit?

上升沿触发，脉冲触发，在瞬间触发，降低干扰

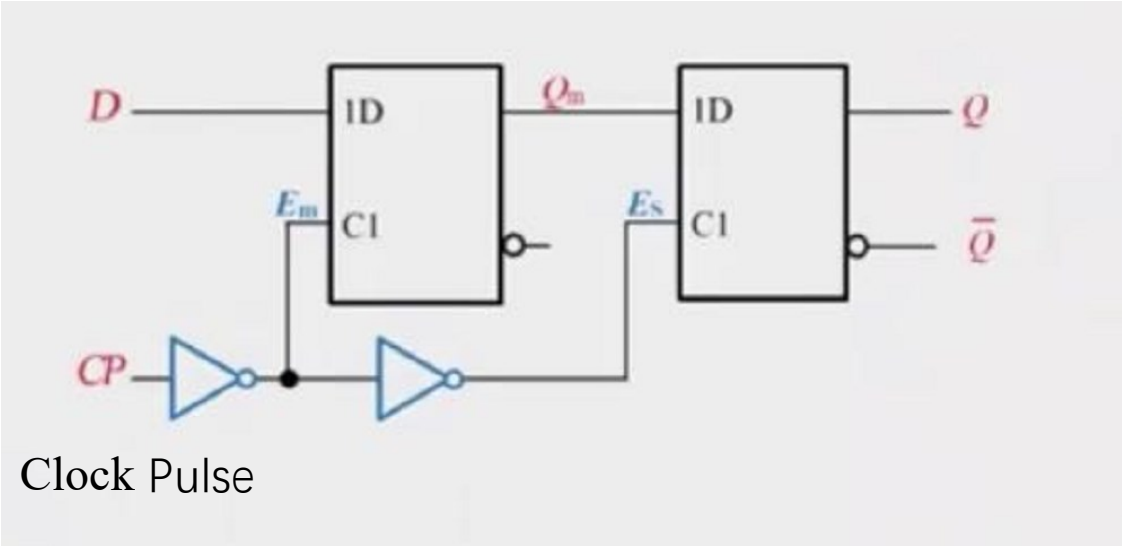
what's meaning of D, Q, CP, Q'

D: input data

Q: output

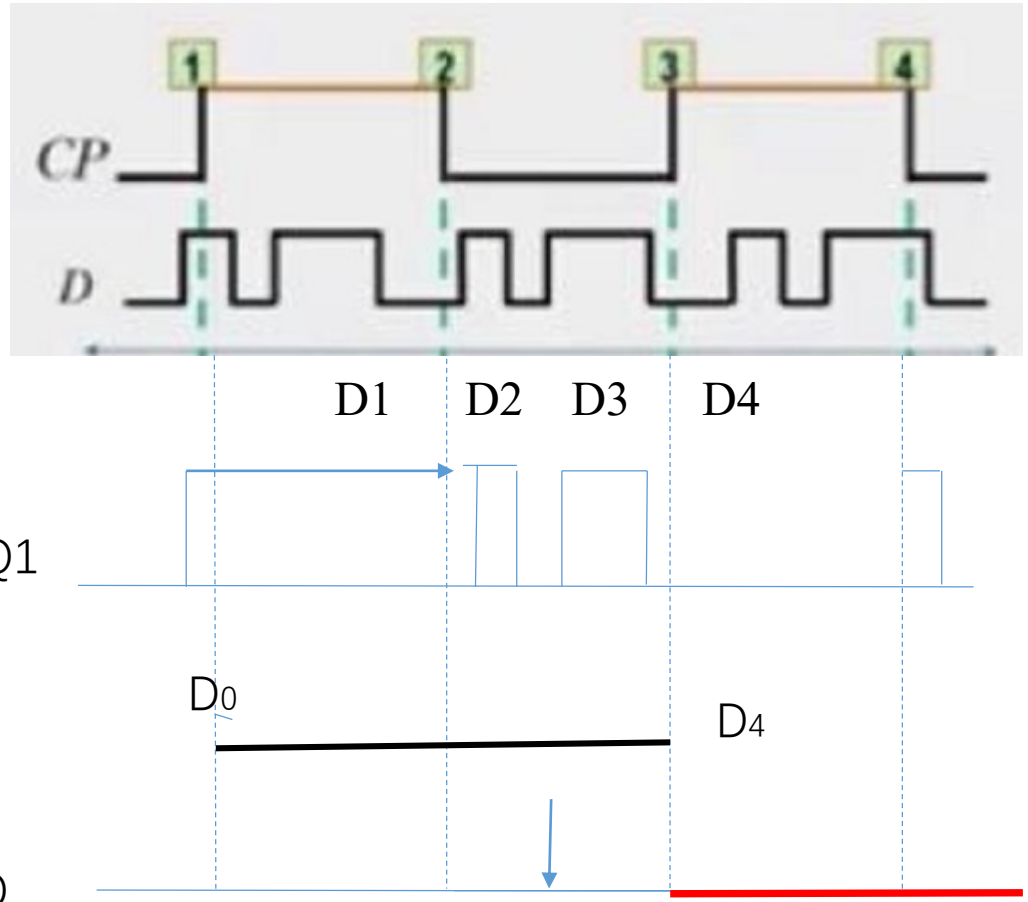
Q': the reverse of output

CP: clock pulse



Clock Pulse

Waveform 波形
Draw Q1, Q



D_0 is effective input
D1 ,D_2 D_3 are interference
D_4 is effective input
Only D_1 and D_4 are reserved

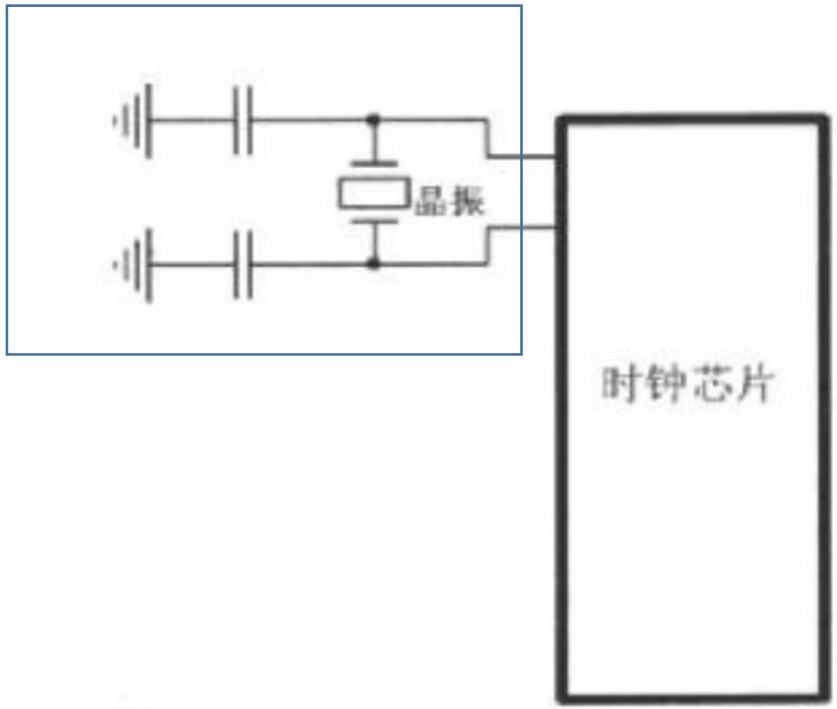
CP control to store

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

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

Where is CP from?

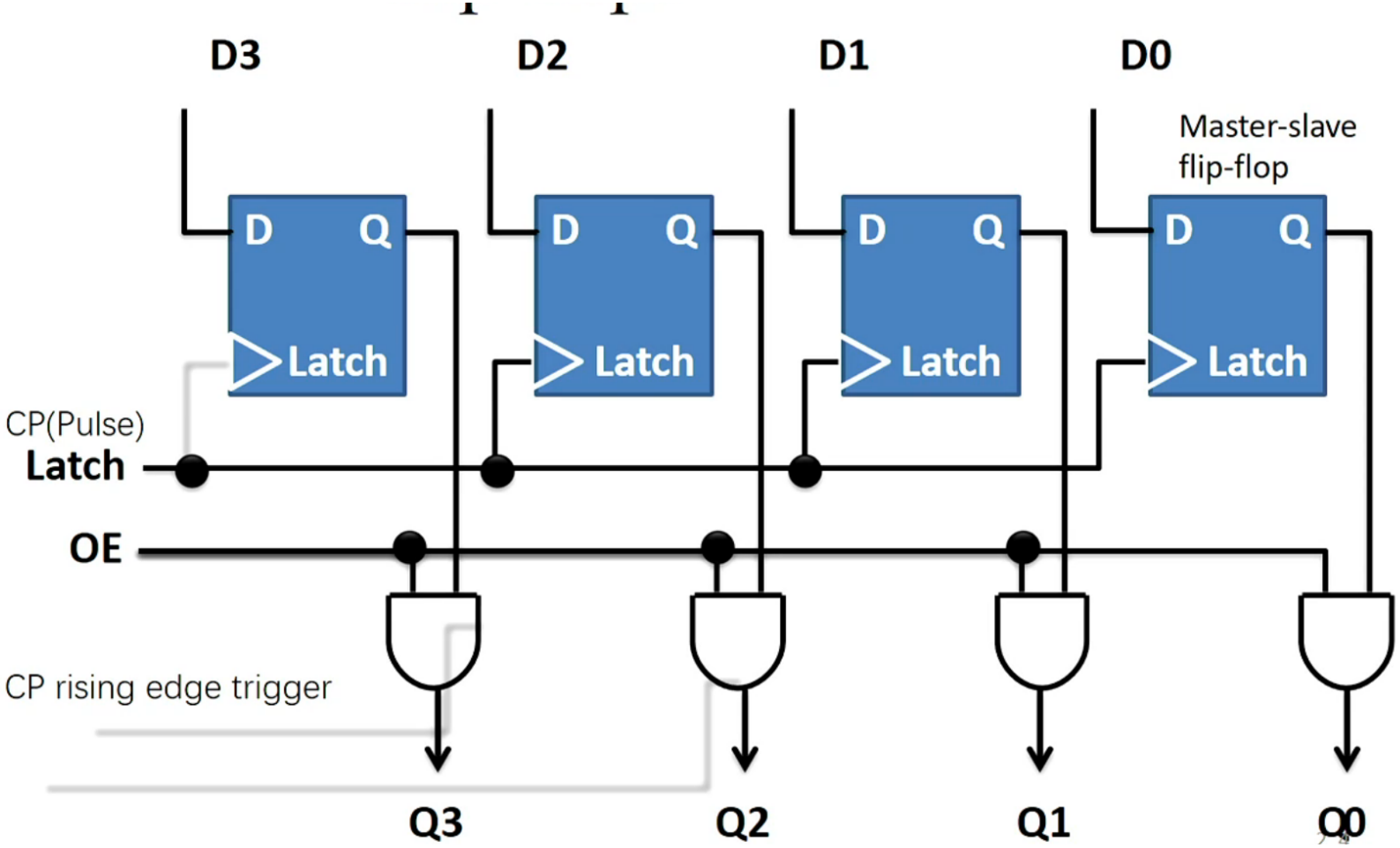


CP is control sigal, which input to control unit of each computer components

Clock pulse circuit

How to build register?

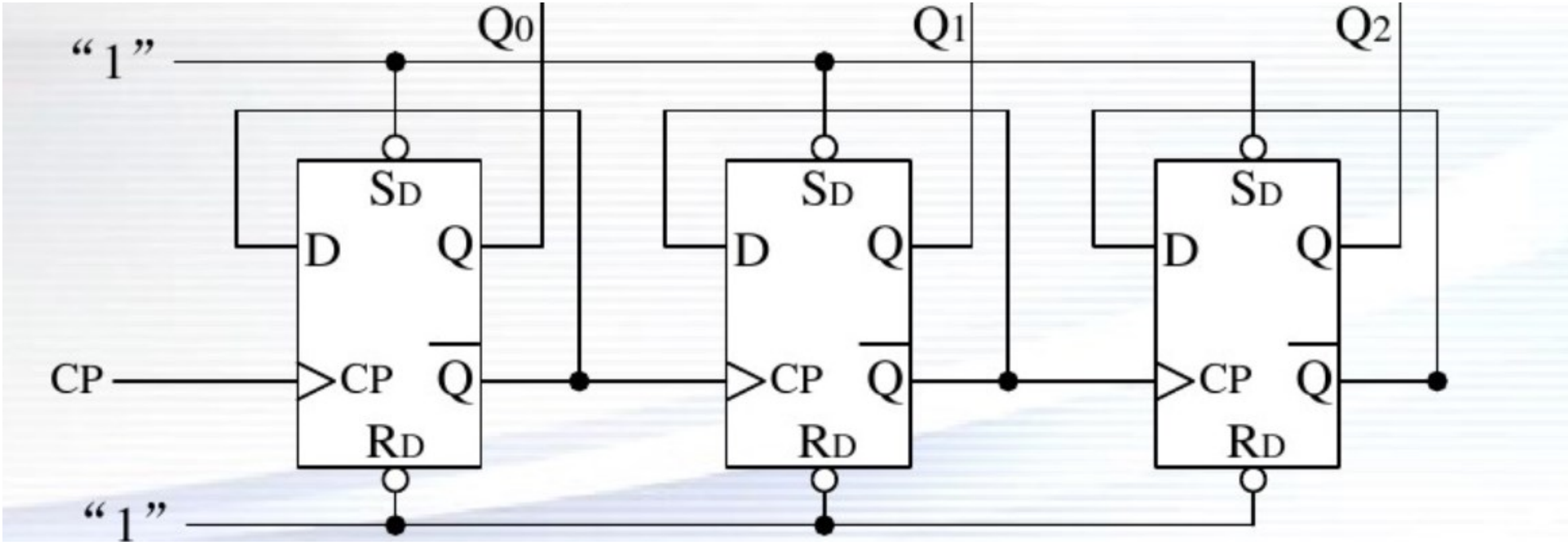
A register using multiple flip-flops



Output Enable

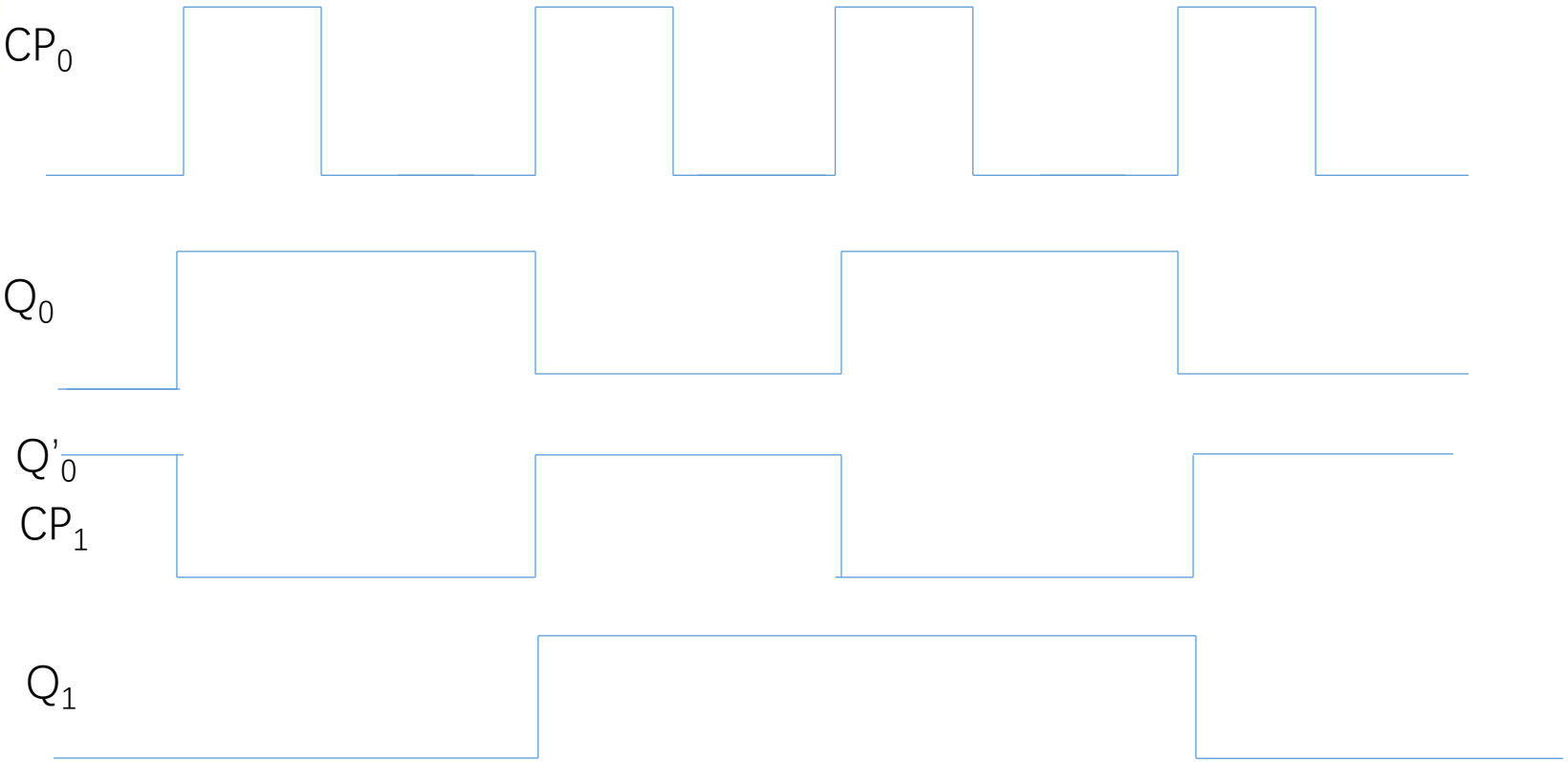
Draw the sequential diagram Analyze the logic gate function

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

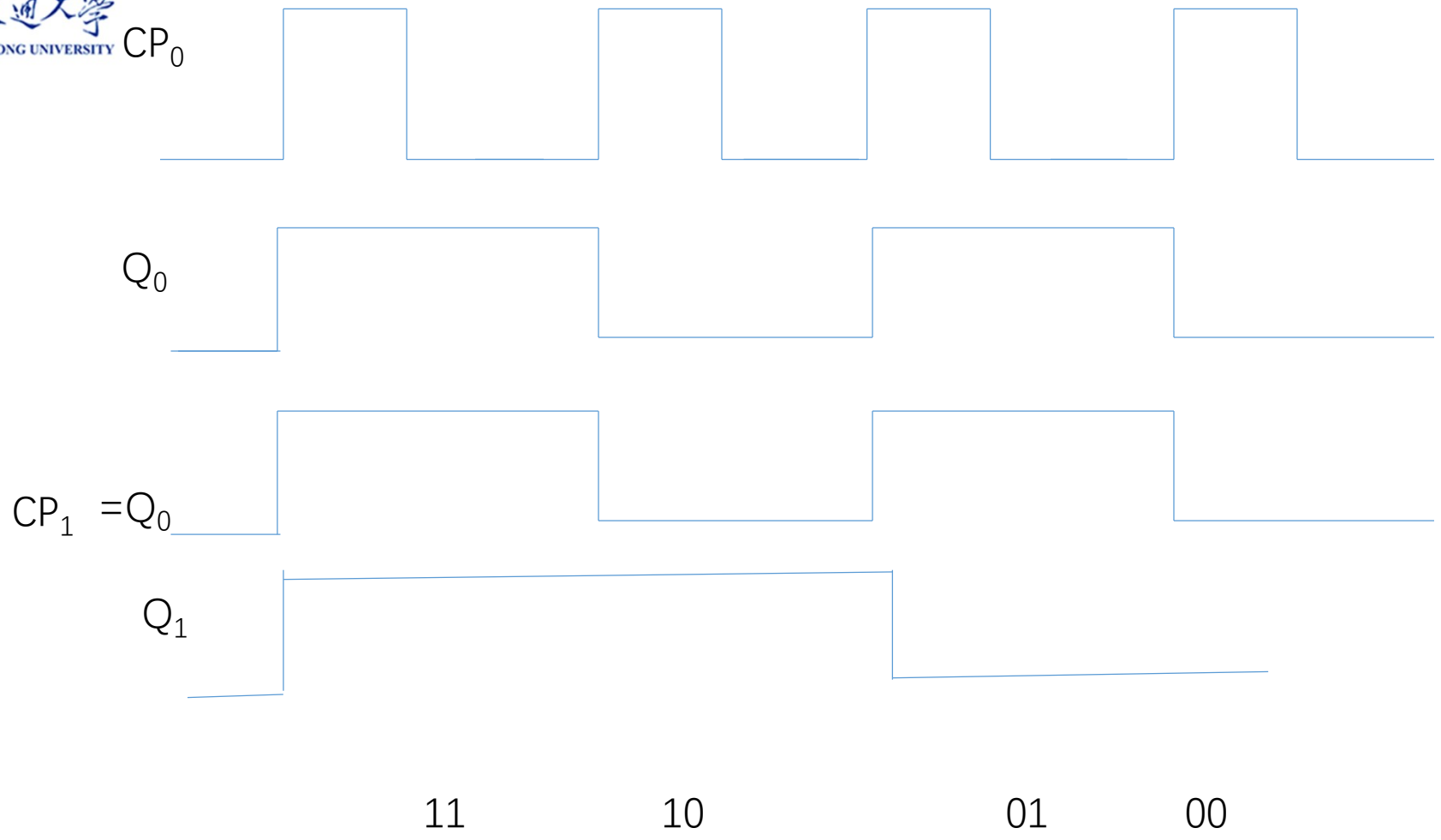


Write logic status equation

Draw sequential waveform

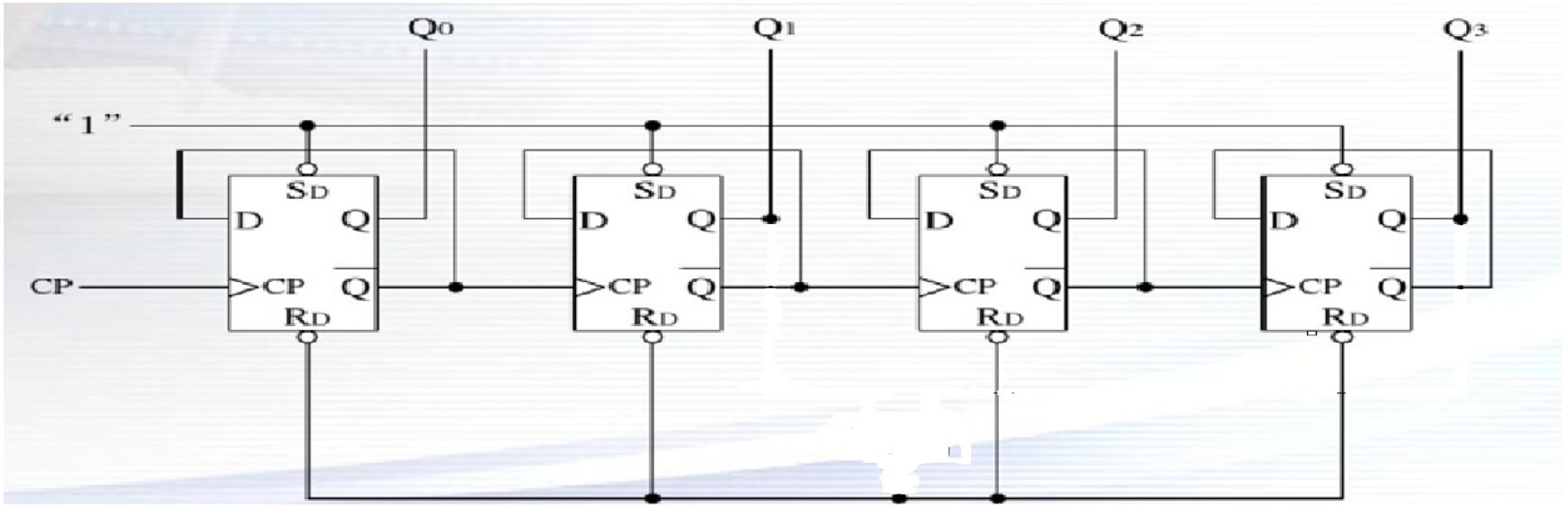


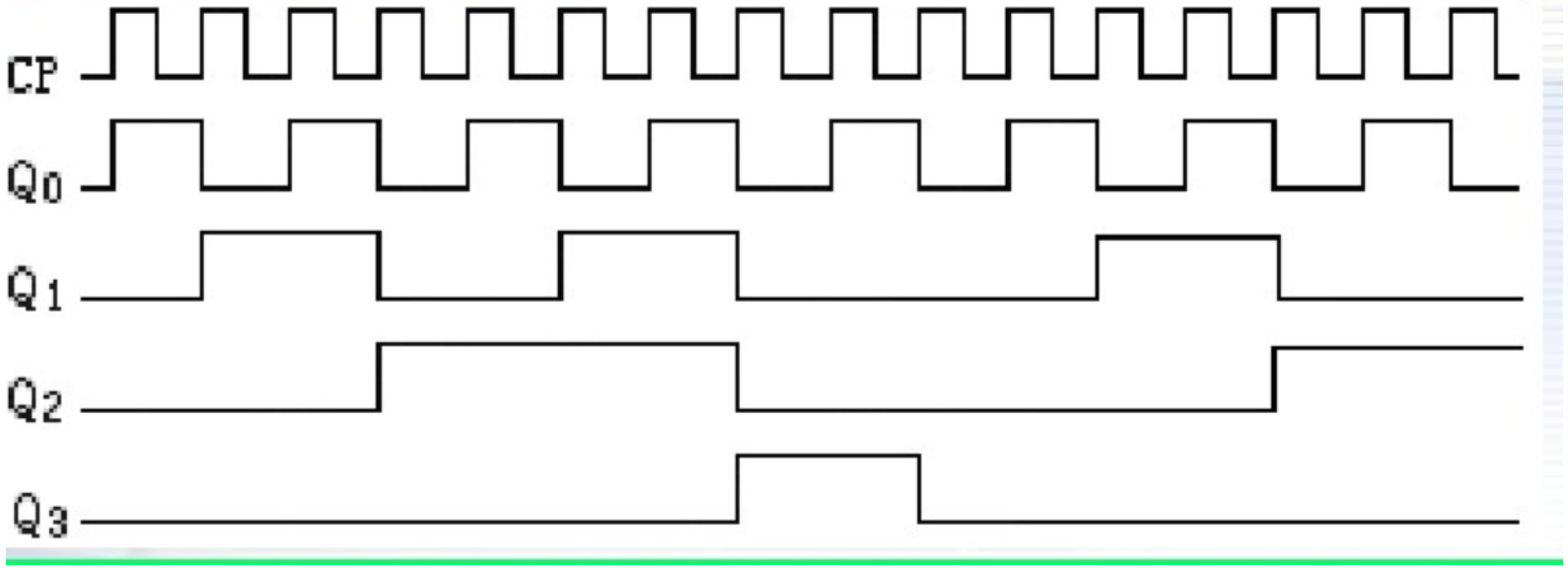
Base 4



Countdown counter

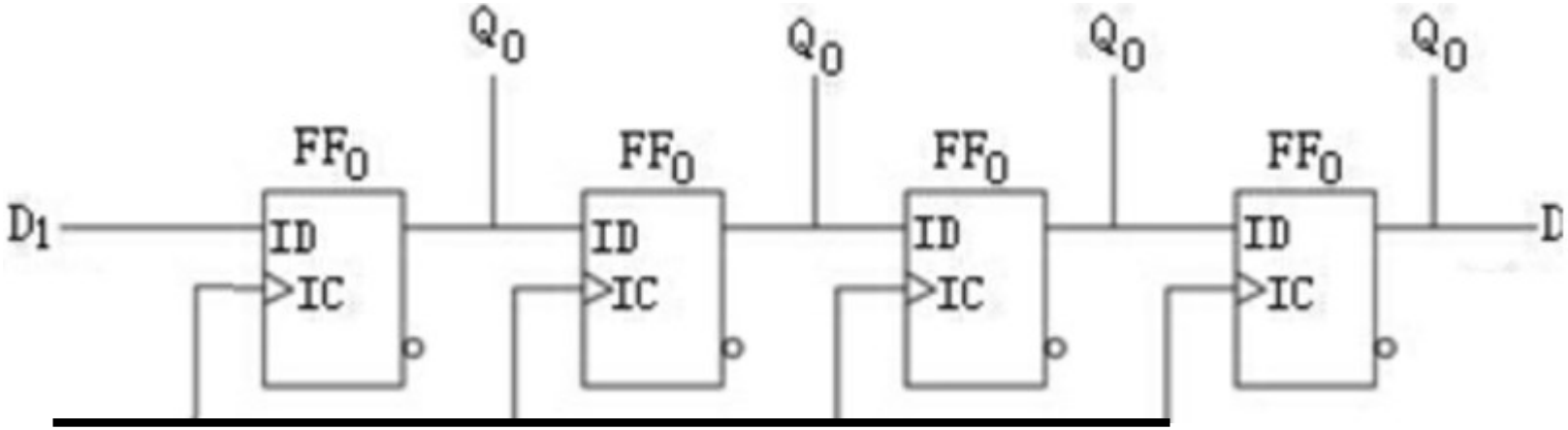
(S is set control, R is reset control, assume Q_0=0)





Counter (Timer)
Frequency divider

Shift right register

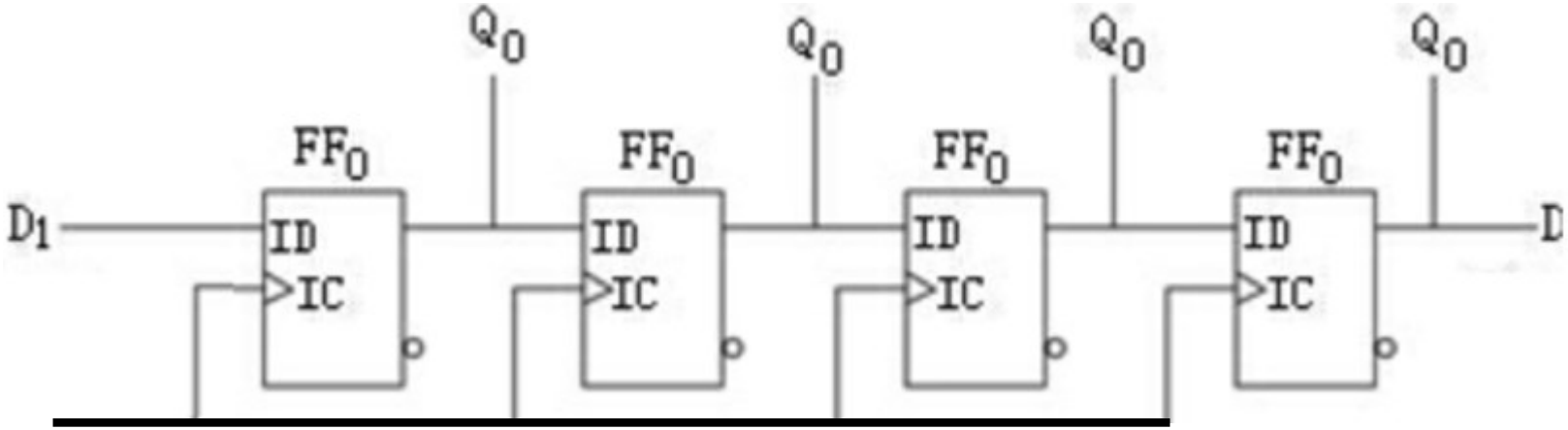


Serial input D1 and parallel out

shifting right

synchronous trigger

Shift right register

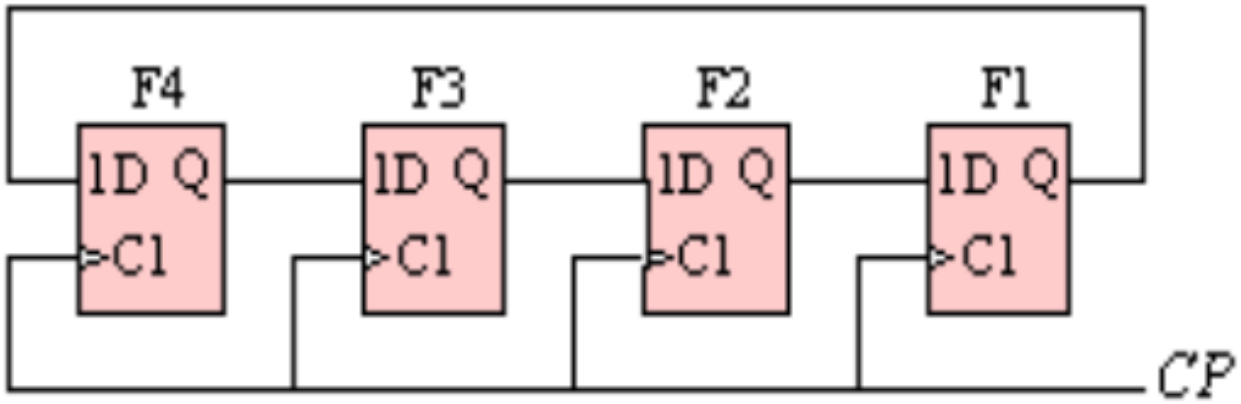


Serial input D1 and parallel out

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

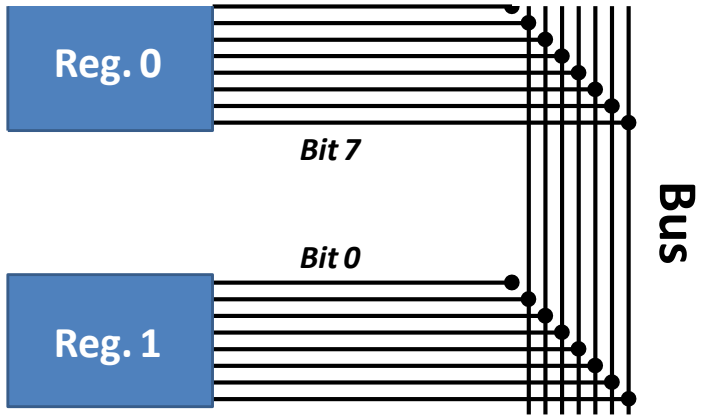
  - An *address bus* runs between the control unit and main memory
    - Used to tell main memory to access a specific address (whether for reading or writing)
    - The "width" of the address bus corresponds to the amount of addressable memory
      - Examples: 16 bits → 65536 bytes (2^{16}); 32 bits → 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)