

# 電気工学通論II (5)

坂井 修一

東京大学大学院 情報理工学系研究科 電子情報学専攻

東京大学 工学部 電子情報工学科

---

- ・ スケジュール
- ・ フリップフロップ

# はじめに

---

## ■ 本講義の目的

- デジタル回路とコンピュータアーキテクチャの基礎を学習する

## ■ 時間・場所

- 火曜日 8:30 – 10:15、工学部2号館213

## ■ ホームページ（ダウンロード可能）

- url: <http://www.mtl.t.u-tokyo.ac.jp/~sakai/tsuron2/>

## ■ 教科書

- 坂井修一 『論理回路入門』（培風館）
- 坂井修一 『コンピュータアーキテクチャ』（コロナ社）

# 講義の概要と予定 (1 / 2)

---

## 1. デジタル回路入門

デジタルとアナログ、2進数、補数表現、四則演算

## 2. 論理演算

論理演算とは、3つの基本論理演算、NORとNAND、完備性、デジタル回路の表現、ブール代数、標準形

## 3. 組み合わせ回路の構成法

真理値表と組合せ回路、組合せ回路の簡単化

## 4. 組合せ回路の実例

加算器、補数、減算器、ALU、デコーダ、セレクトタなど

## 5. フリップフロップ

FF、SRラッチ、Dラッチ、非同期と同期、SR-FF、D-FF、マスタスレーブ形とエッジトリガ形、JK-FF、レジスタ

## 6. 基本的な順序回路

## 7. 一般的な順序回路の作り方

# 講義の概要と予定 (2 / 2)

## 8. コンピュータアーキテクチャ入門

計算のサイクル、主記憶装置、メモリの構成と分類、レジスタファイル、命令、命令実行の仕組み、実行サイクル、算術論理演算命令、シーケンサ、条件分岐命令

## 9. 命令セットアーキテクチャ

操作とオペランド、命令の表現形式、アセンブリ言語、命令セット、算術論理演算命令、データ移動命令、分岐命令、アドレッシング、サブルーチン、RISCとCISC

## 10. パイプライン

## 11. キャッシュと仮想記憶

## 12. 入出力と周辺装置

周辺装置、ディスプレイ、二次記憶装置、ハードウェアインタフェース、割り込みとポーリング、アービタ、DMA、例外処理

休講: X月Y日      試験: Z月W日

成績: **出席(小テスト) + 試験**      **4年生: 追試・レポート無し**

# 論理回路教科書について



- 出版社品切れで迷惑をおかけしました
  - 出版社と生協書籍部には早くに連絡してあったのですが。。
- 先週、重版ができ、生協書籍部に入りました

# 試験について

---

■ 1月14日(火) 8:30～10:00

を予定しています

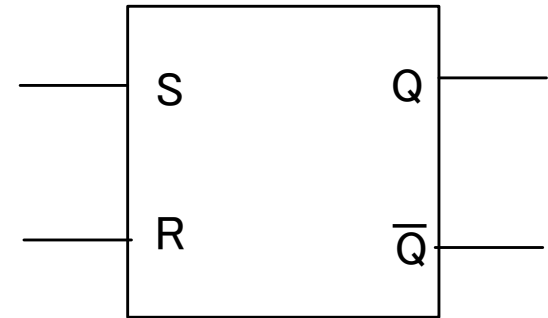
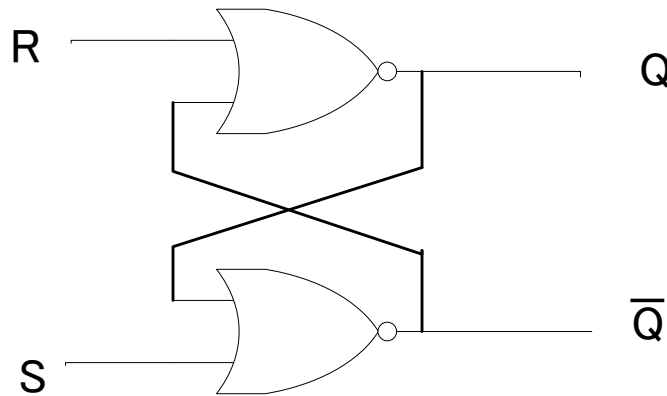
# 5. フリップフロップ (Flip Flop, FF)

---

- フリップフロップ = 1 bitの状態をもつ回路
- フリップフロップの分類
  - 非同期型 (ラッチ、latch)
    - 入力を与えられるとただちに出力が変化する
      - ・ SRラッチ (非同期SRフリップフロップ)
      - ・ Dラッチ (非同期Dフリップフロップ)
  - 同期型 (狭義のフリップフロップ)
    - クロック (矩形波) が生じた時点の入力にだけ出力が影響を受ける
      - ・ SRフリップフロップ
      - ・ Dフリップフロップ
      - ・ JKフリップフロップ
      - ・ Tフリップフロップ

# 5.1 SRラッチ (非同期SRフリップフロップ)

## ■ SR: Set - Reset

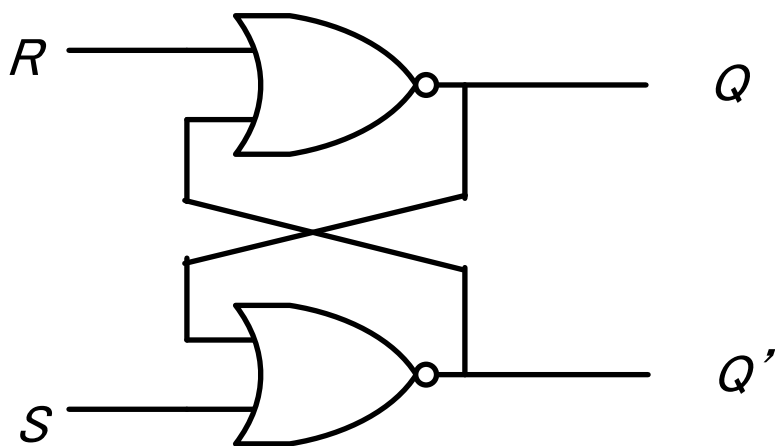


S	R	Next Q	Next $\bar{Q}$	意味
0	0	Q	$\bar{Q}$	前の状態
0	1	0	1	リセット
1	0	1	0	セット
1	1	0	0	禁止入力

左のような表を  
**特性表**という



# SRラッチ：動作の理解



- $S = 0, R = 0$  のとき

$$\text{Next } Q = \text{not}(Q')$$

$$\text{Next } Q' = \text{not}(Q)$$

- $S = 0, R = 1$  のとき

$$\text{Next } Q = \text{not}(1) = 0$$

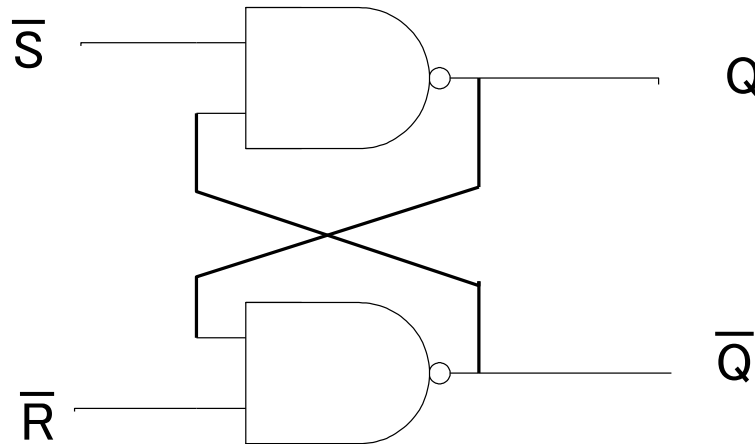
$$\text{Next } Q' = \text{not}(Q) = 1$$

- $S = 1, R = 0$  のとき

$$\text{Next } Q = \text{not}(Q') = 1$$

$$\text{Next } Q' = \text{not}(1) = 0$$

# NANDゲートによるSRラッチ



## 問題

1. 上の回路の状態遷移表を書け
2. NORゲートによる状態遷移表と差があるか？

今後、NANDゲートによるSRラッチを基本回路と考える

# 回答

---

S	R	Next Q	Next $\overline{Q}$	意味
0	0	Q	$\overline{Q}$	前の状態
0	1	0	1	リセット
1	0	1	0	セット
1	1	1	1	禁止入力

禁止入力に対する出力が違う ⇒ 実用的な意味はない

# 特性表と励起表

## SRラッチの動作

(a) 特性表

S	R	Q <sub>next</sub>
0	0	Q
0	1	0
1	0	1
1	1	禁止入力

(b) 励起表

Q	Q <sub>next</sub>	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

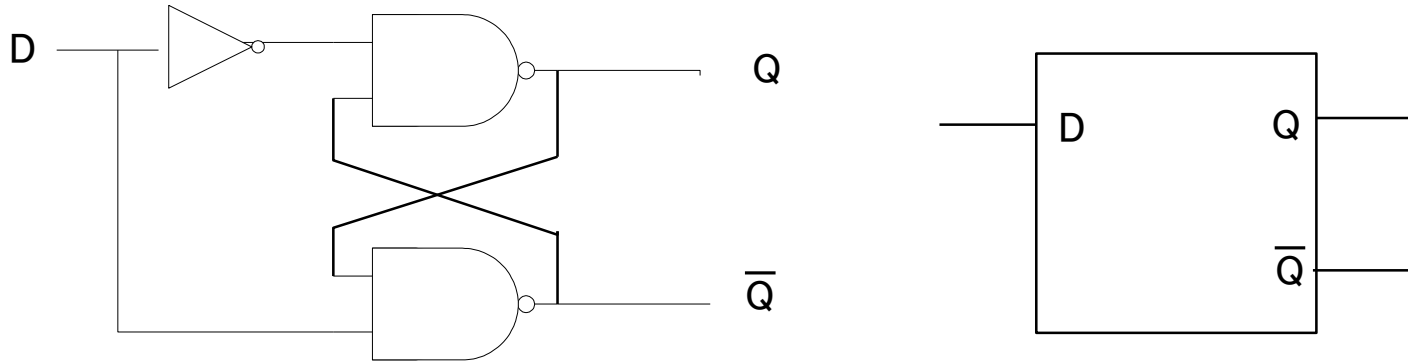
Xは、0でも1でもよい

「入力と状態に対する次状態」を表す

「求める状態変化に対する入力」を表す

# 5.2 Dラッチ (非同期Dフリップフロップ)

## ■ D: Delay



Dラッチの動作

(a) 特性表

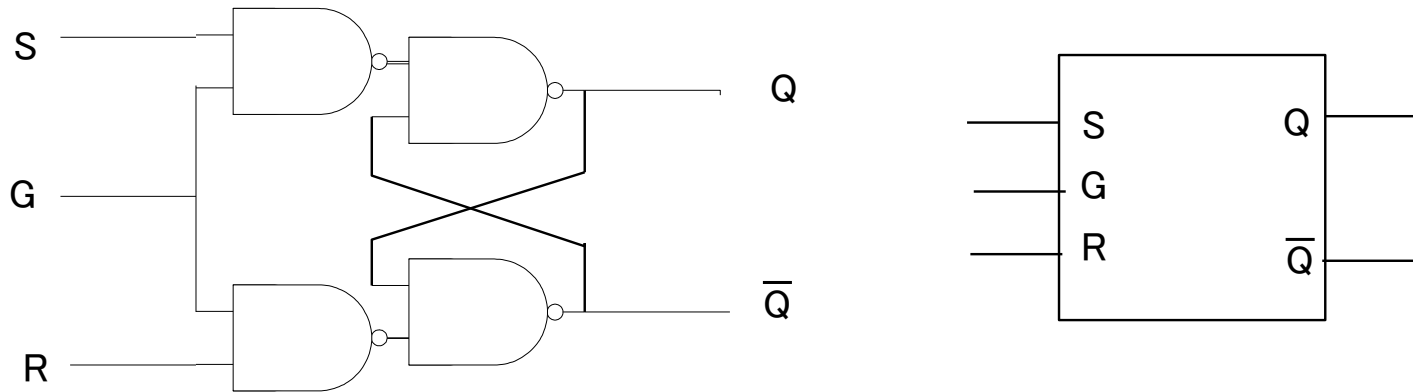
D	Q <sub>next</sub>
0	0
1	1

(b) 励起表

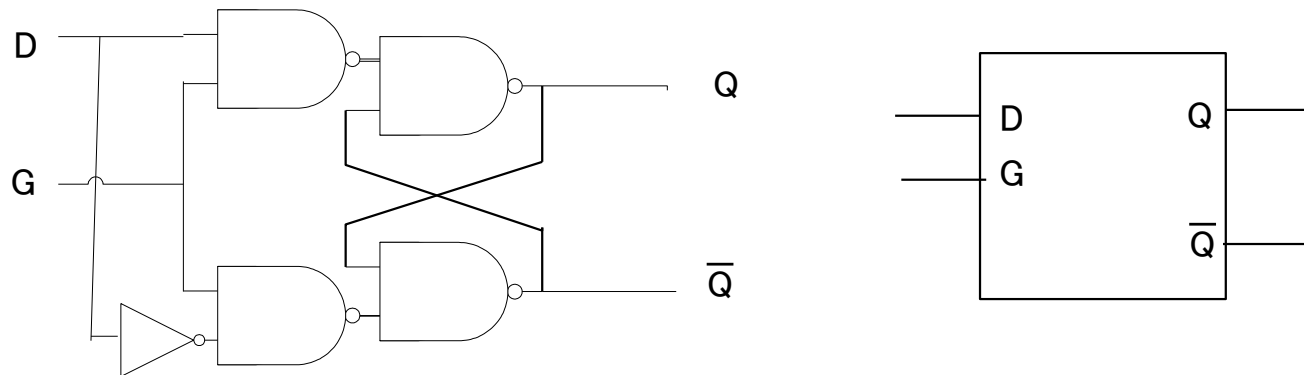
Q	Q <sub>next</sub>	D
0	0	0
0	1	1
1	0	0
1	1	1

# 5.3 ゲートつきラッチ

- ゲート： 入力が有効であるかどうかを制御する信号



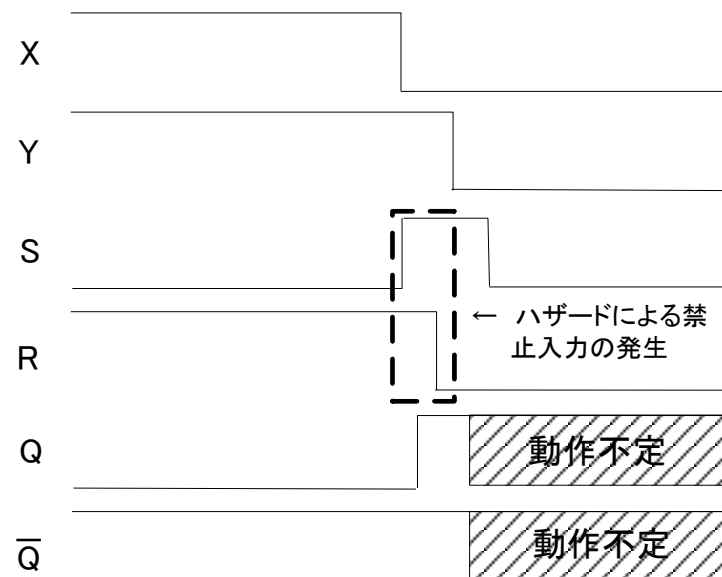
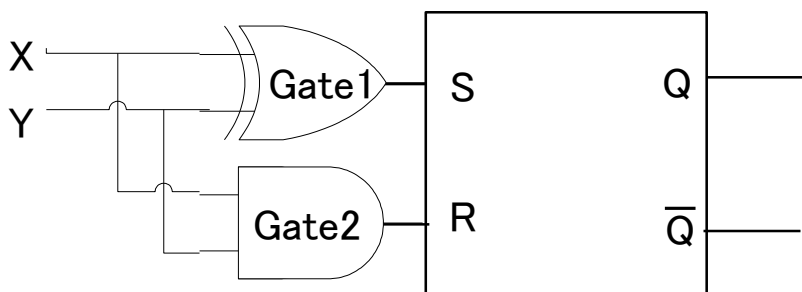
ゲートつきSRラッチ



ゲートつきDラッチ

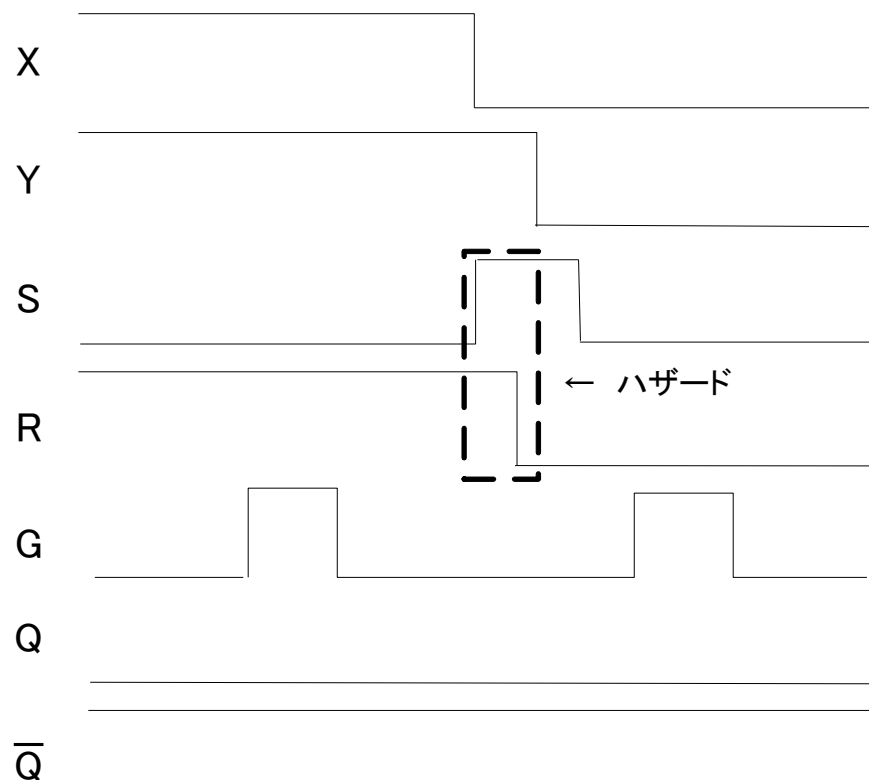
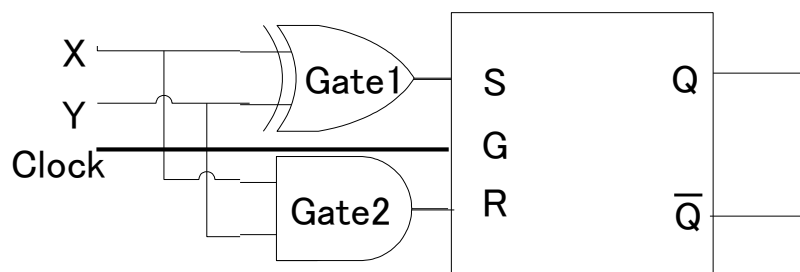
# 5.4 非同期と同期

- 非同期回路: 入力を与えられるとただちに出力が変化する回路
- 非同期の問題点: ハザード(Hazard)**
- 2つの入力SとRに時間的誤差があるとき、一時的に期待していない値が現れる現象



# 非同期と同期（続き）

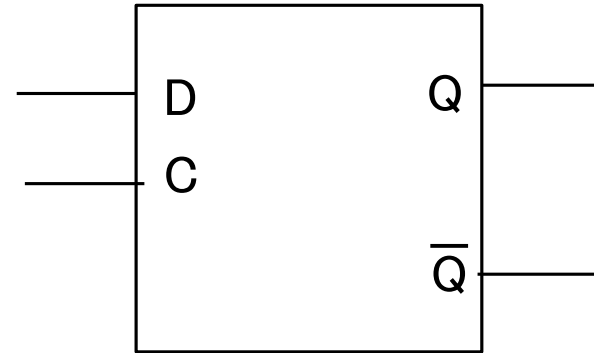
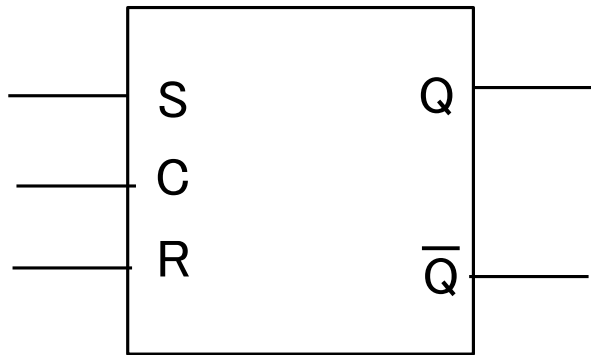
- 同期回路：クロックが与えられた時点における入力信号のみに応答する回路
  - 同期回路では**ハザードが生じない**ようにクロックを与えることができる





## 5.5 (同期式)フリップフロップ

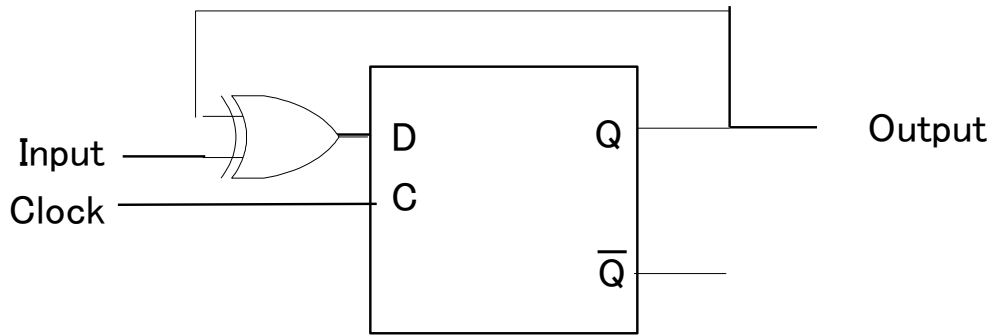
- CLK = 1 のときだけ値を取り込む



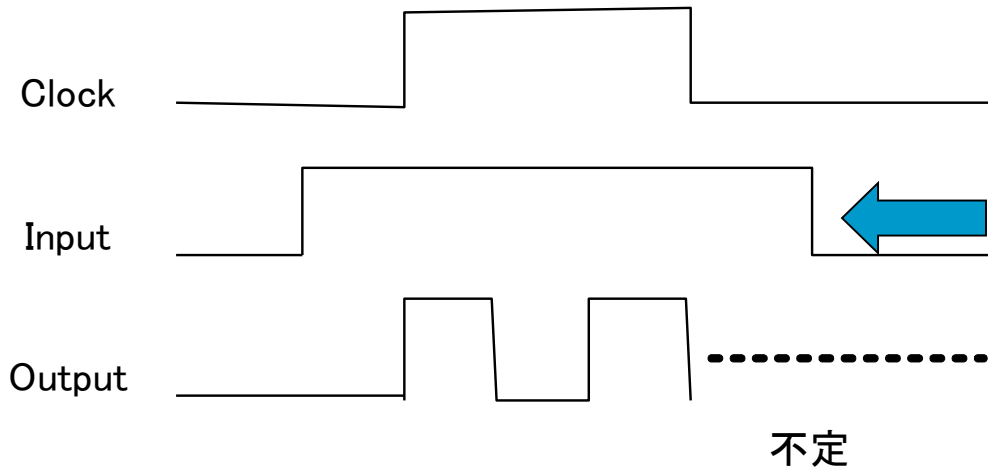
CLKをゲートに入れることで同期式を実現

# フリップフロップの問題点

## ■ 発振



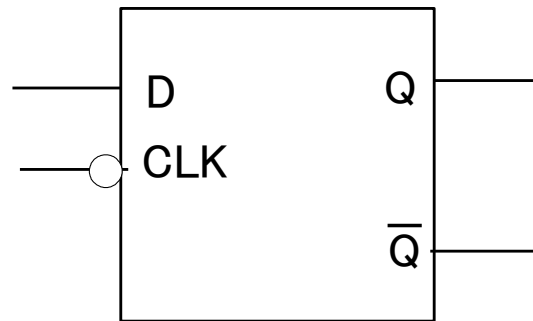
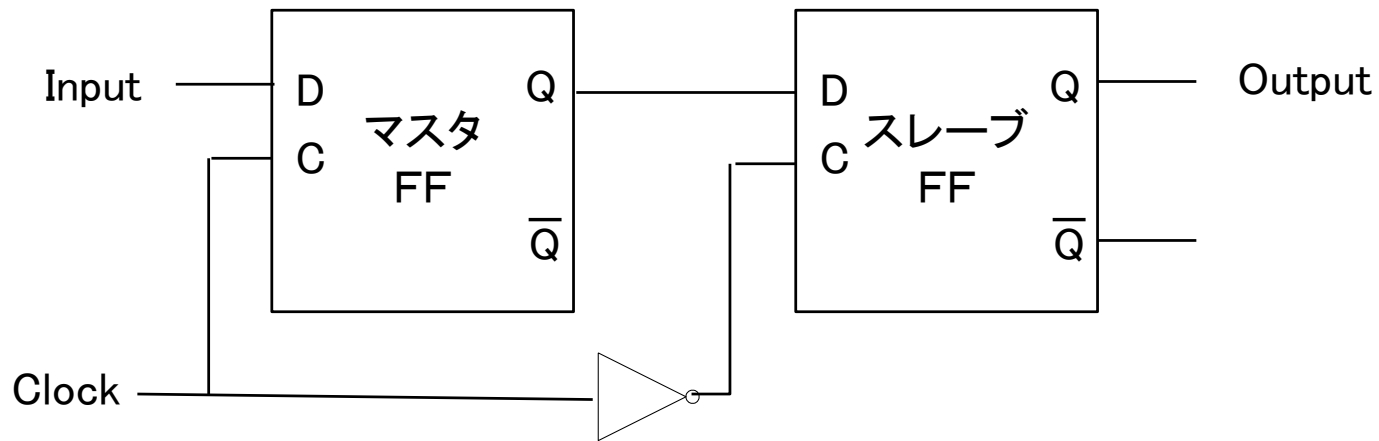
回路にフィードバックがある場合、1クロックの間に何度も出力が変化することがある



← 入力はクロック = 1 の間安定している

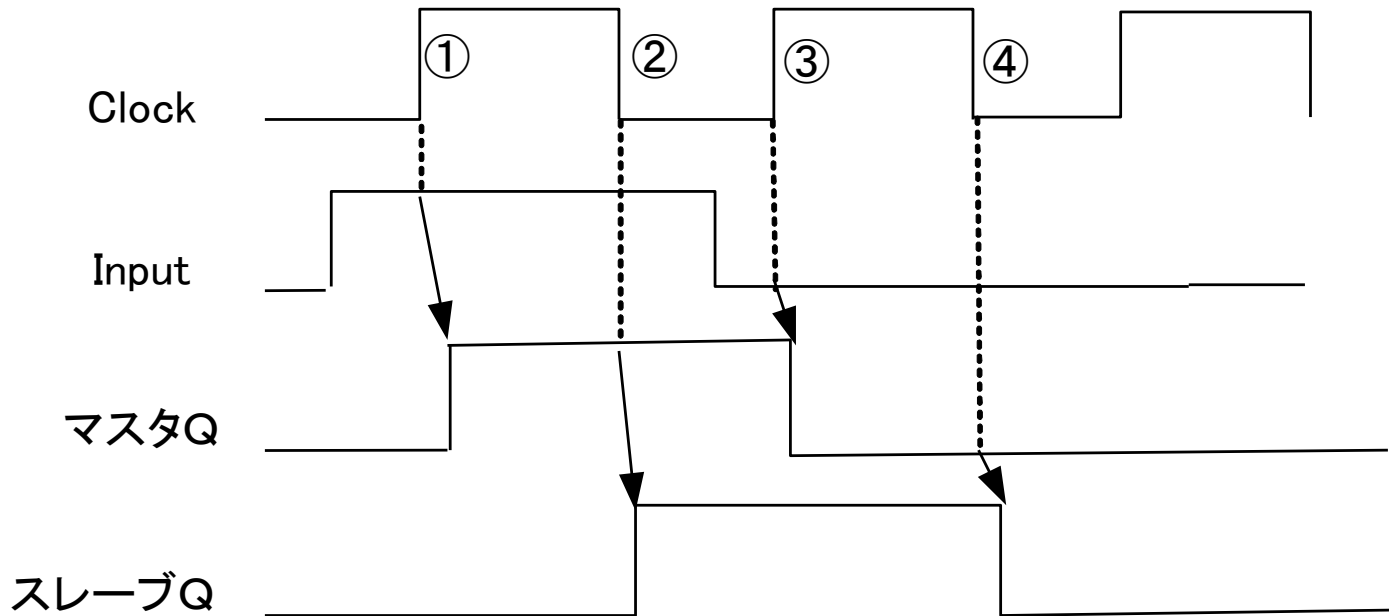
# 5.6 マスタスレーブ方式

1クロックの間に起こる状態遷移と出力の変化を1度だけにして発振を防ぐ



マスタスレーブ型Dフリップフロップ

# マスタスレーブ形FFの動作



①③スレーブゲートが閉じる

マスタゲートが開き、マスタFFに入力値が取り込まれる

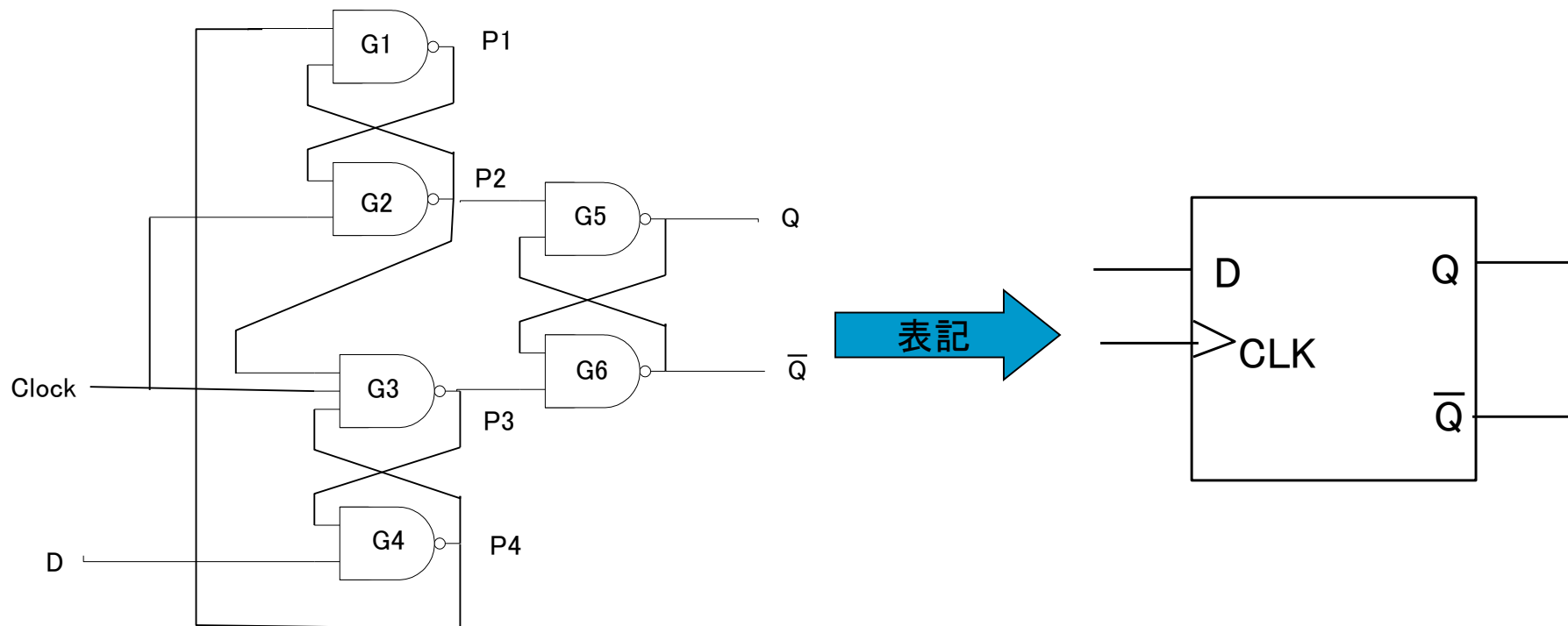
②④マスタゲートが閉じ、FFの内容がこのときの値に決まる

スレーブゲートが開き、マスタFFの内容がスレーブFFに移る

注意. 通常この形のFFは、CLKの立ち上げ～立ち下げまで入力値を変えないようにしている

# 5.7 エッジトリガ(edge trigger)方式

- マスタスレーブ形の問題点: クロックの「山の幅」分の遅延がFFの中で生じる



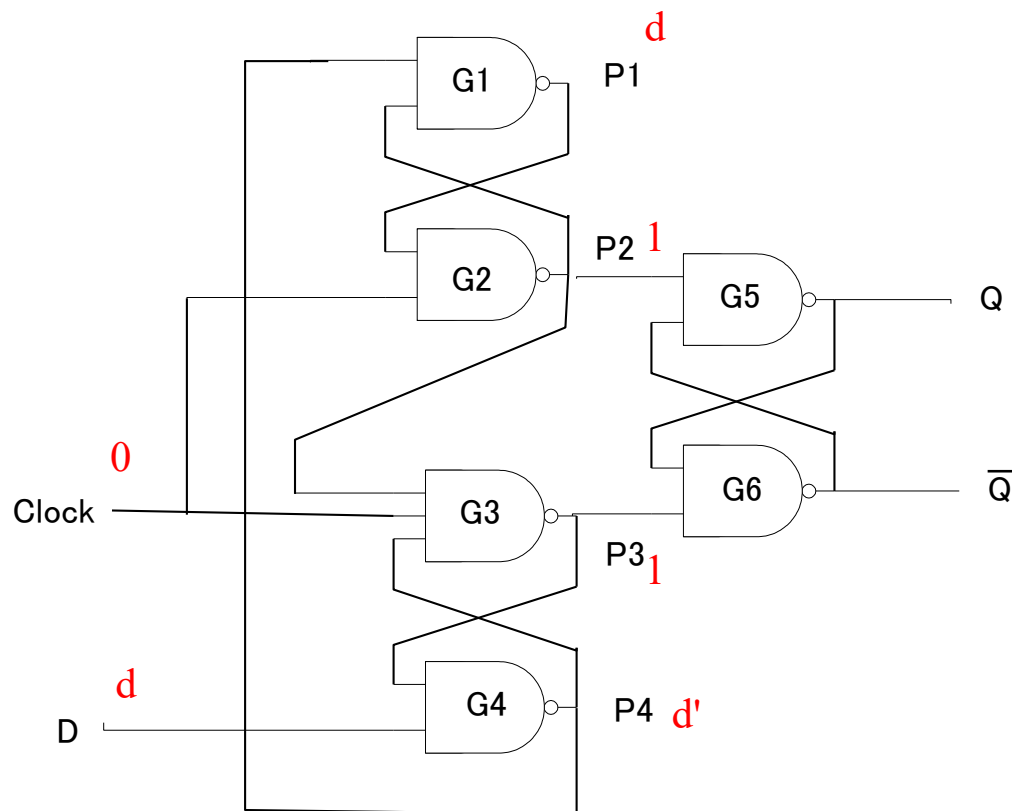
エッジトリガ型Dフリップフロップ

原理： クロックの立ち上（下）がりで値が確定するFF

遅延は信号の変化時間だけで、クロックの山の幅には関係しない

# エッジトリガD-FFの動作 (1)

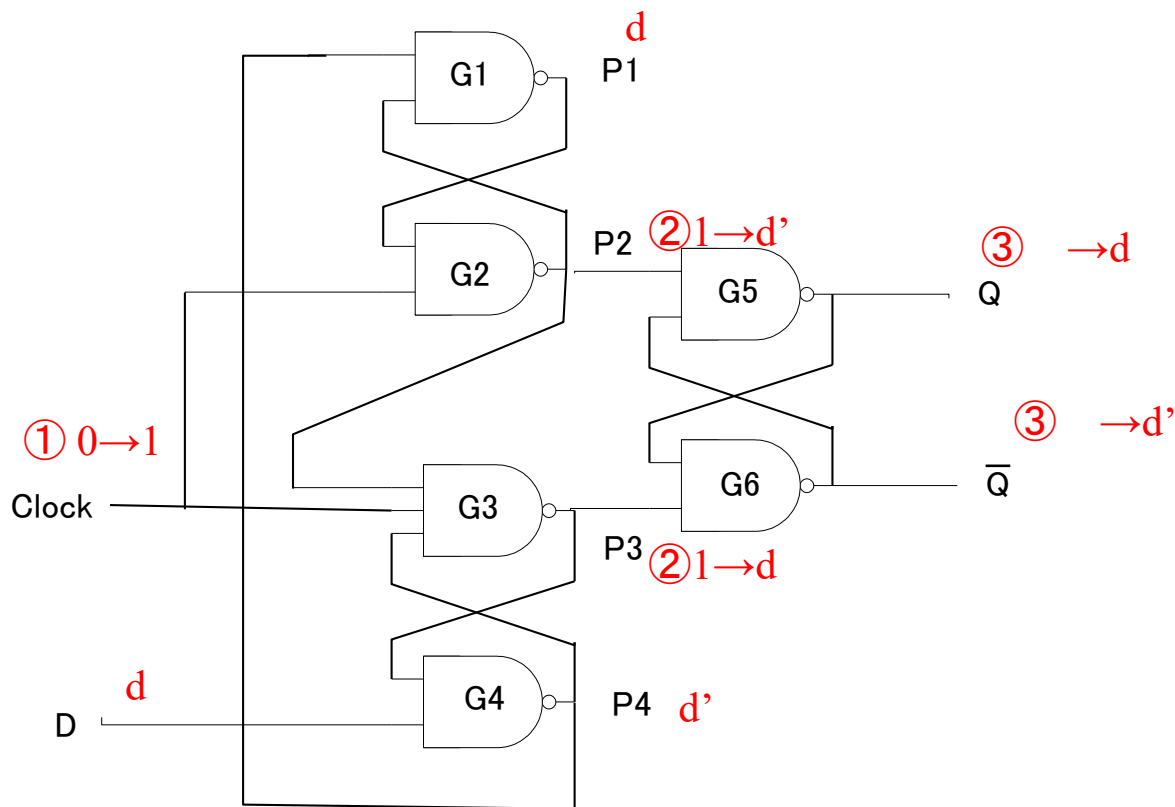
## ■ $Clock=0$



値は保持される

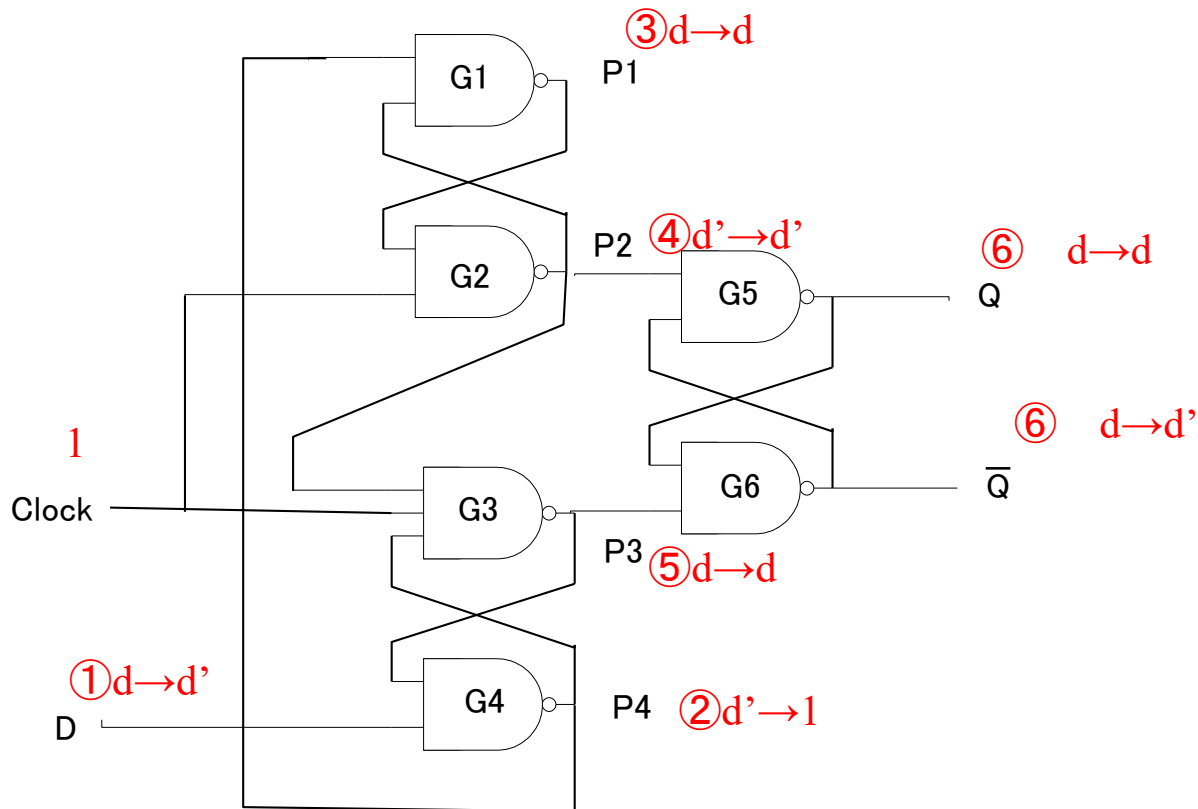
# エッジトリガD-FFの動作 (2)

- *Clock*: 0 → 1



# エッジトリガD-FFの動作 (3)

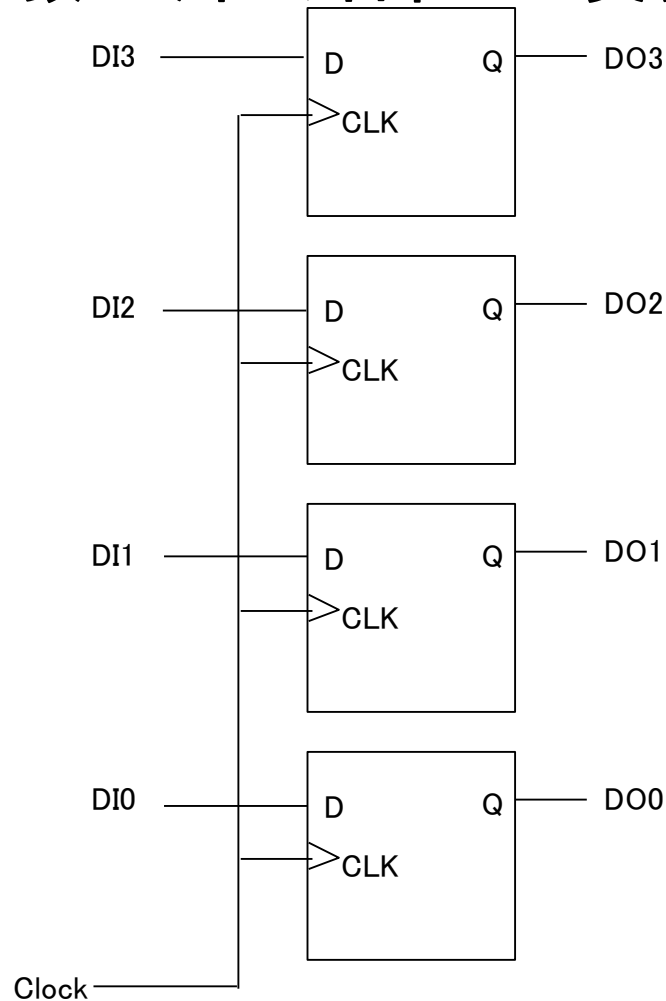
- $Clock=1$ : 入力データが反転しても値は保持される





## 5.8 レジスタ(置数器、register)

- 情報を一時蓄える回路。フリップフロップを必要なビット数だけ組み合わせて実現する

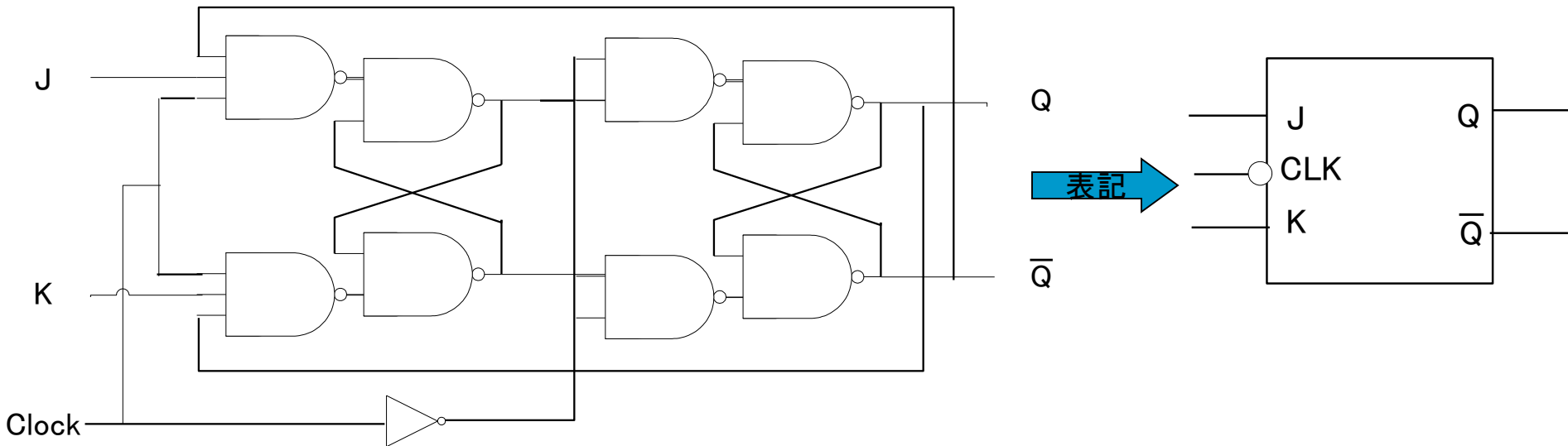


CLKの立ち上がりで、Nビットのデータを取り込み、これを出力する

# 5.9 JKフリップフロップ

マスタFF

スレーブFF



(a) 特性表

J	K	$Q_{next}$
0	0	Q
0	1	0
1	0	1
1	1	$\bar{Q}$

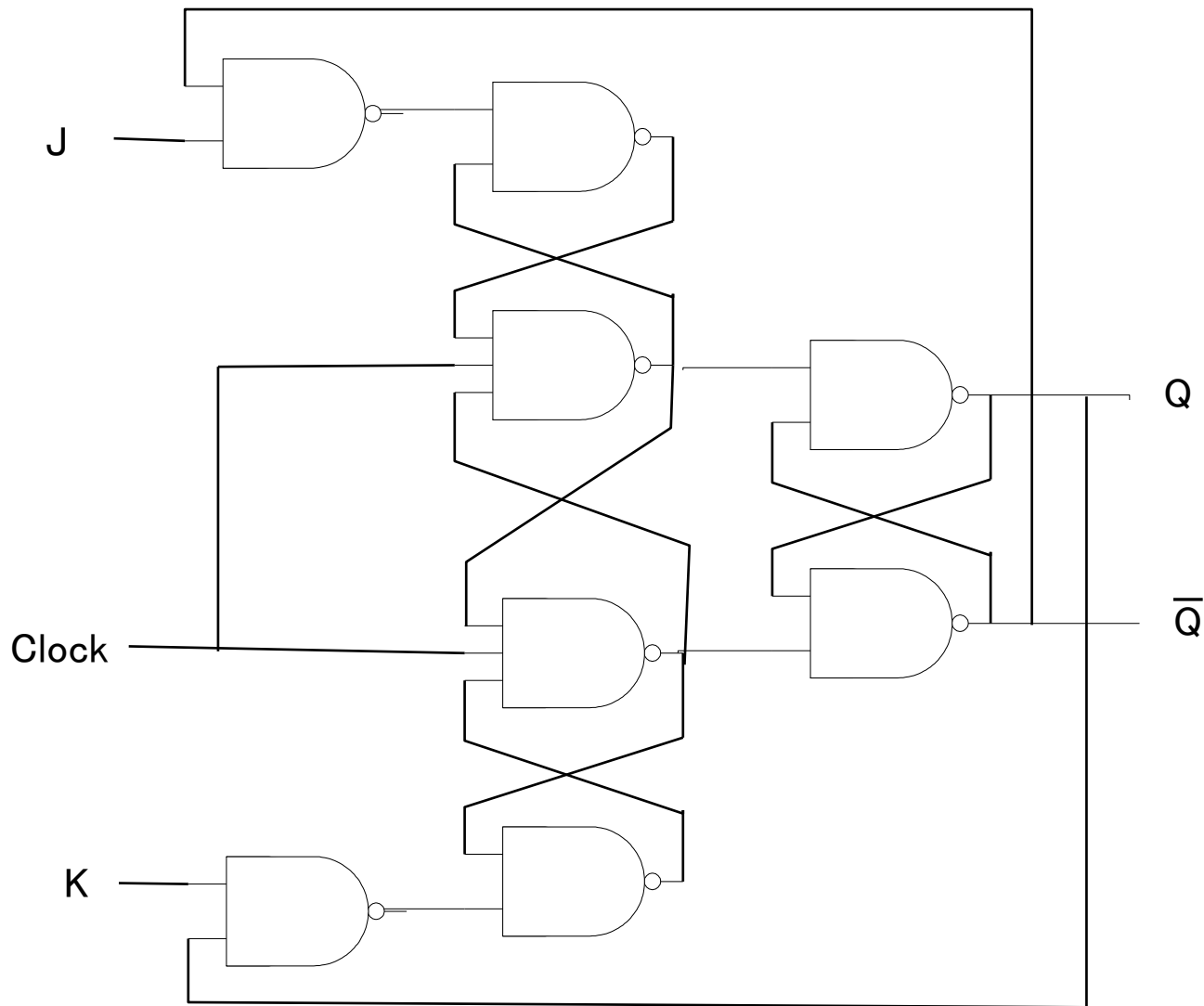
(b) 励起表

Q	$Q_{next}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Xは、0でも1でもよい

- ・ SR-FFとほぼ同じだが、入力が(1,1)のとき、**状態が反転**する
- ・ マスタスレーブ型またはエッジトリガ型となる

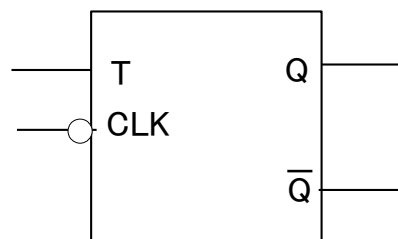
# エッジトリガ型JKフリップフロップ



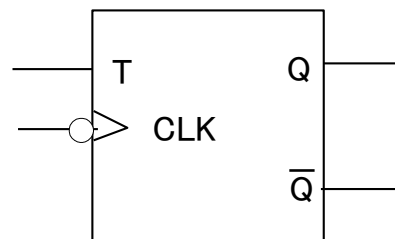
# 5.10 Tフリップフロップ

## ■ T: toggle (trigger)

- JKフリップフロップの入力JとKとともにTを入力したもの



マスタスレーブ型



エッジトリガ型

(a) 特性表

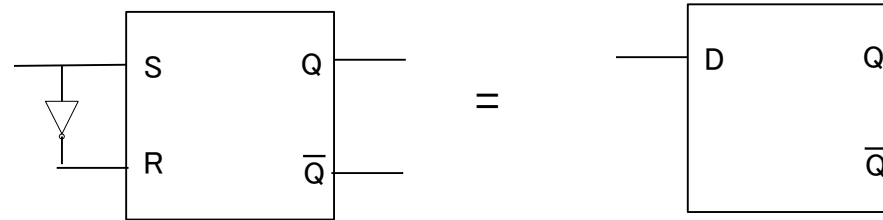
T	$Q_{next}$
0	Q
1	$\bar{Q}$

(b) 励起表

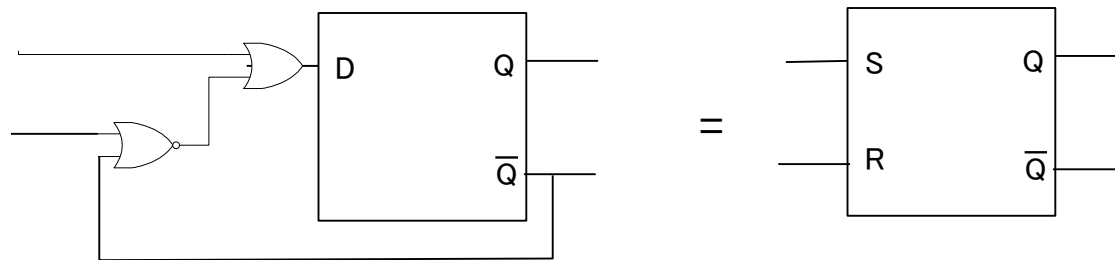
Q	$Q_{next}$	T
0	0	0
0	1	1
1	0	1
1	1	0

- ・ Tが1のとき、状態が反転する
- ・ マスタスレーブ形T-FF、エッジトリガ形T-FFがある

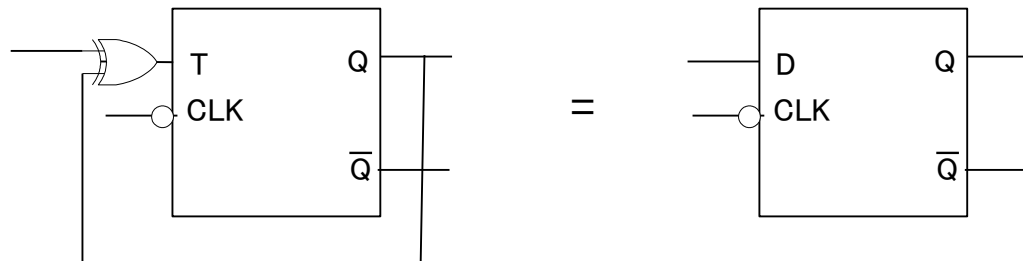
# 5.11 フリップフロップの変換



(a) SRラッチとDラッチ



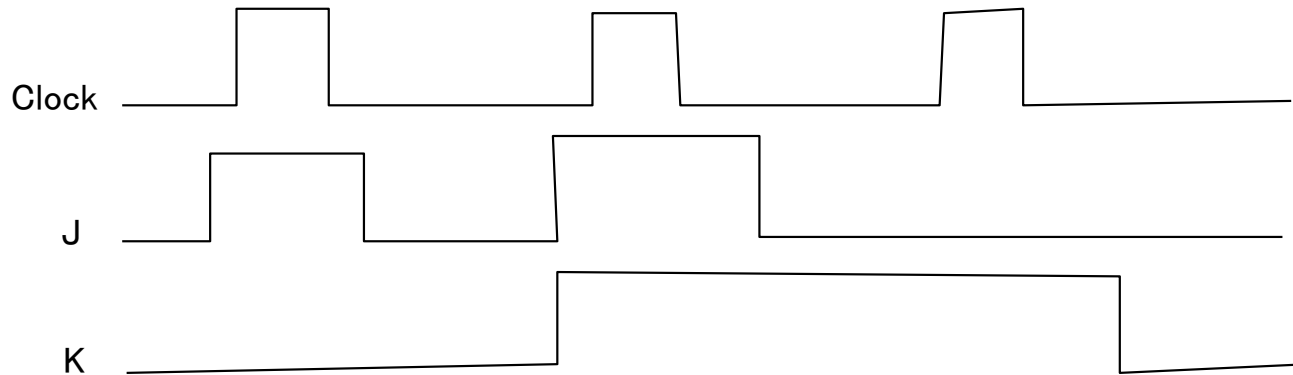
(b) DラッチとSRラッチ



(c) TフリップフロップとDフリップフロップ  
(マスタスレーブ型)

# 問題

1. エッジトリガ型D-FF、エッジトリガ型JK-FFの動作を確認せよ
2. マスタスレーブ型JKフリップフロップの動作を、下の入力に対して確定せよ



3. 次のフリップフロップの変換を、図を書いて行え。
  - (1) SRフリップフロップを使ってJKフリップフロップを実現 (マスタスレーブ型、以下同様)
  - (2) Tフリップを使ってJKフリップフロップを実現
  - (3) JKフリップフロップを使ってSRフリップフロップを実現)
  - (4) JKフリップフロップを使ってDフリップフロップを実現
  - (5) Dフリップフロップを使ってJKフリップフロップを実現
  - (6) Dフリップフロップを使ってTフリップフロップを実現