

# ハードウェア実験

## 組み込みシステム入門 第5回

2012年10月18日

## 順序論理回路の実験

このスライドの、ゲートの動作記述の部分は、藤井先生のスライドから多くをいただいています。  
藤井先生に慎んでお礼申し上げます。



# 今日の内容

- ▶ 以下の論理回路を動作させる。
  1. Dフリップフロップ回路
  2. 4進カウンタ回路(同期式)
  3. 10進カウンタ回路
- ▶ シフトレジスタを作成して、LEDの表示が流れる動作を見る。(発展課題)



# 順序論理回路とは

組み合わせ論理回路では

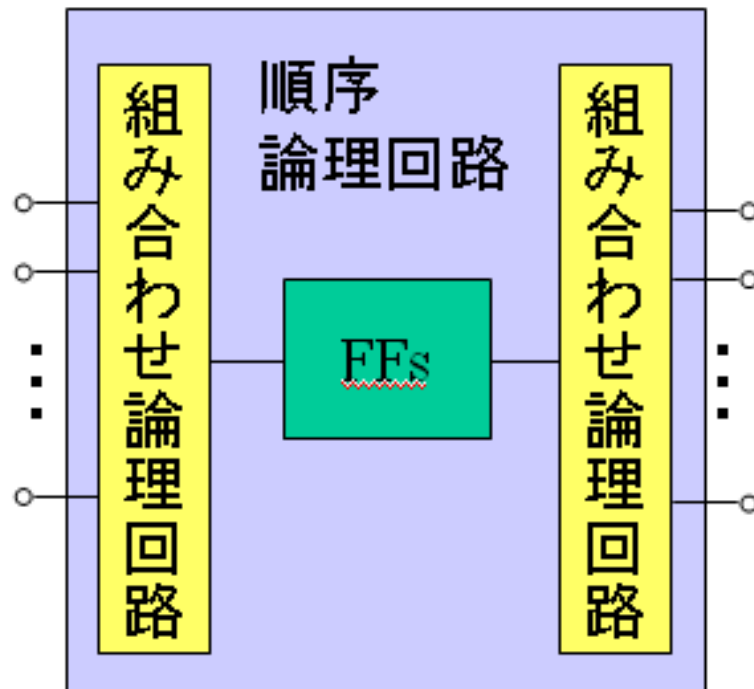
$$\text{出力} = f(\text{入力})$$

で記述するように、入力の組み合わせで出力が決まっていた。

順序論理回路では、

$$\text{出力} = f(\text{内部状態、入力})$$

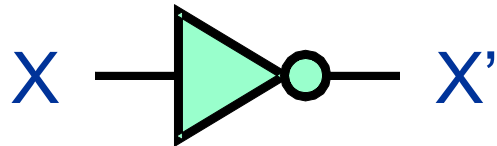
で記述されるように、入力の組み合わせだけでなく内部状態によっても出力が変化する。内部状態の記憶はFFによって行われる。



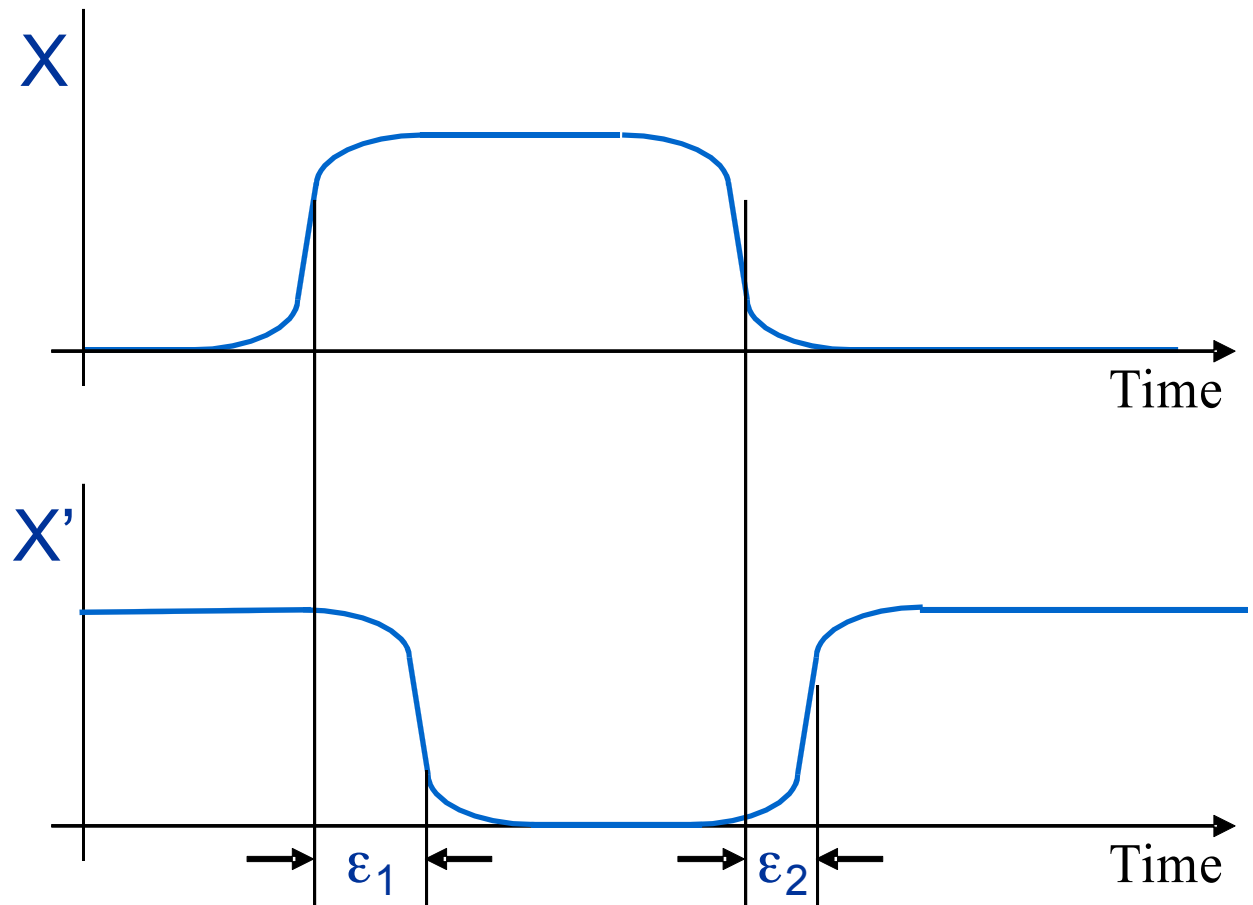
より具体的には、入力の組み合わせに応じて記憶素子であるFFの状態を変化させるのに必要なFFの入力を決定する。  
さらに、内部状態の変化に基づき必要な出力を組み上げる。



# 論理ゲートは遅延がある。



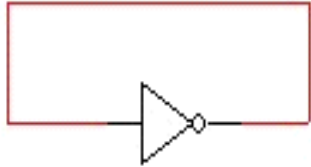
普通はゲートの遅延がドミナントだが、高速LSIでは配線/容量による遅延も無視できなくなる。



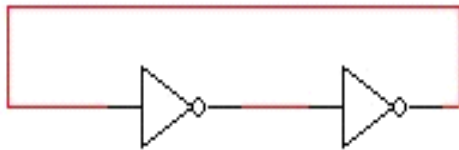
遅延 →



# 回路内にフィードバックループがあると

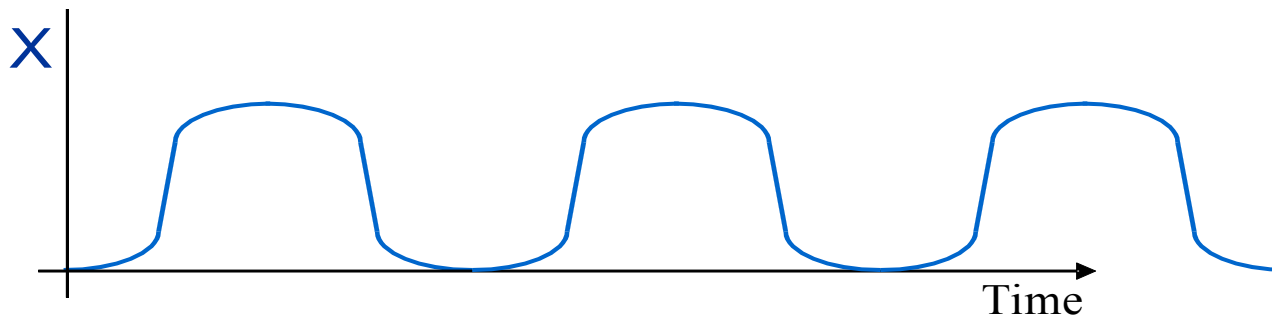
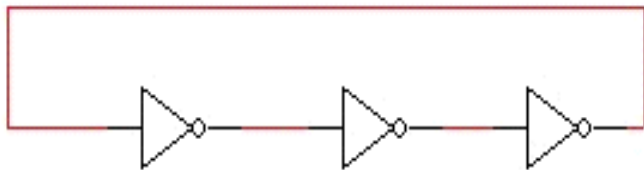


NOTゲートが奇数個のループは発振する。

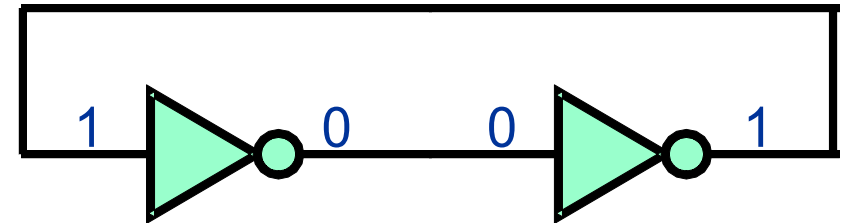
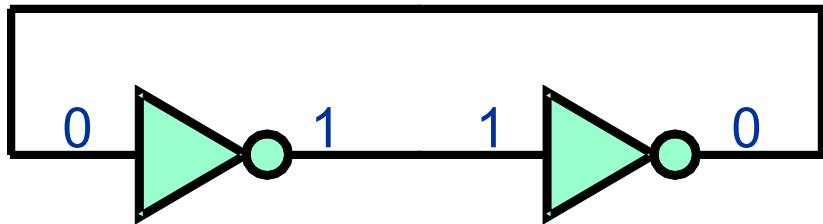
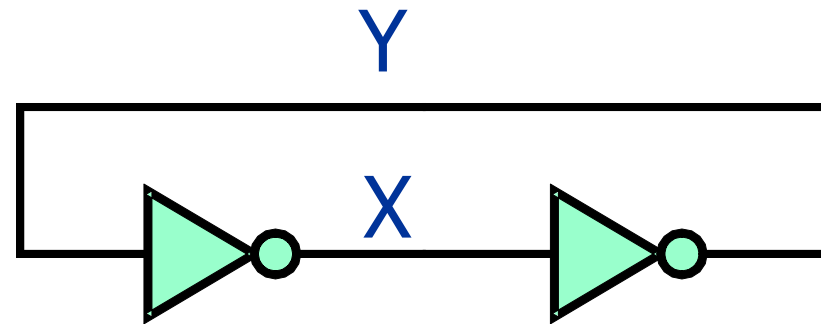


(一個だと通常は発振しない)

偶数個の場合は？



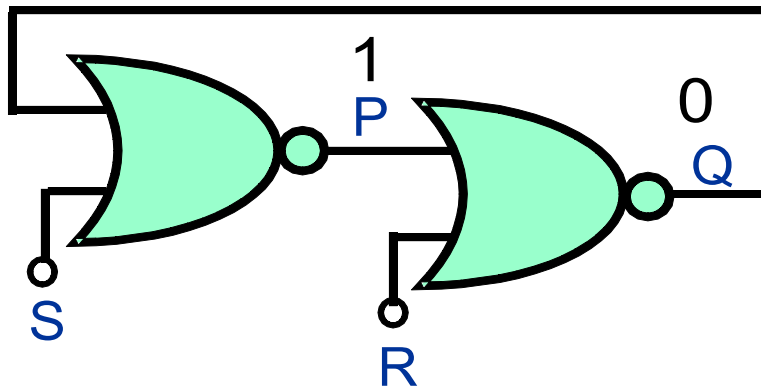
# 2個のNOTゲートのループ回路の振る舞い





# 2個のNORゲートのループ

S,Rの入力が“0”の時のゲート  
出力は？



(1)  $P=0$ とした時;

$P=0$  and  $Q=1$ で安定

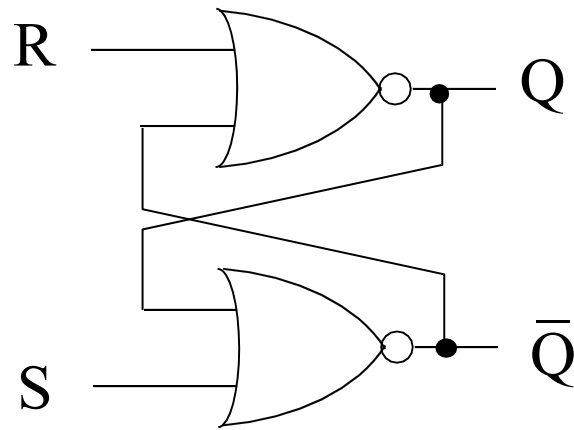
(2)  $P=1$ とした時;

$P=1$  and  $Q=0$ で安定

双安定回路  $\Rightarrow$  Set Reset Flip Flop



# NOR回路による RS-FF



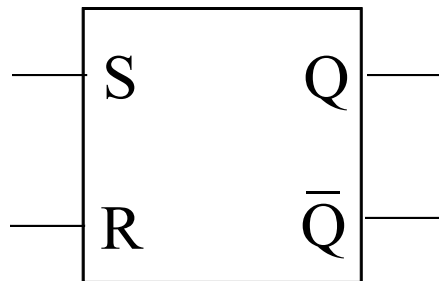
真理値表

| S | R | $Q_n$ | $Q_{n+1}$ |
|---|---|-------|-----------|
| 0 | 0 | 0     | 0         |
| 0 | 0 | 1     | 1         |
| 0 | 1 | 0     | 0         |
| 0 | 1 | 1     | 0         |
| 1 | 0 | 0     | 1         |
| 1 | 0 | 1     | 1         |
| 1 | 1 | 0     | -         |
| 1 | 1 | 1     | -         |

動作のまとめ

| S | R | $Q_{n+1}$ |           |
|---|---|-----------|-----------|
| 0 | 0 | $Q_n$     | : Keep    |
| 0 | 1 | 0         | : Reset   |
| 1 | 0 | 1         | : Set     |
| 1 | 1 | -         | : Inhibit |

}  $Q = \bar{Q} = 0$  となるので  
使用不能となる。



カルノー図

| SR \ $Q_n$ | 0      | 1      |
|------------|--------|--------|
| 00         |        | 1      |
| 01         |        | 1      |
| 11         | $\phi$ | $\phi$ |
| 10         | 1      | 1      |

特性方程式

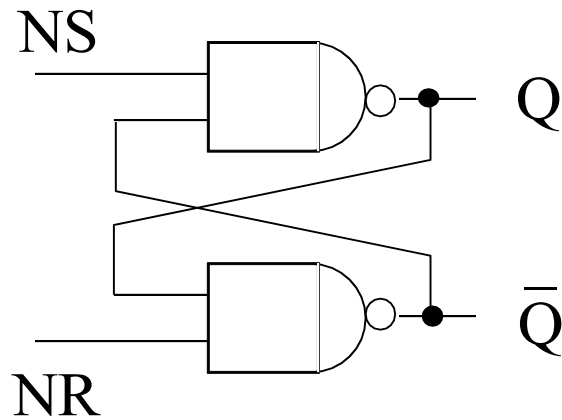
$$Q_{n+1} = S + \bar{R}Q_n$$

ただし、 $SR = 0$





# NAND回路による RS-FF



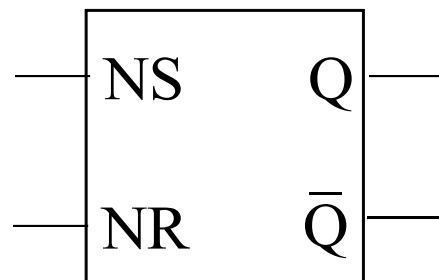
真理値表

| NS | NR | $Q_n$ | $Q_{n+1}$ |
|----|----|-------|-----------|
| 1  | 1  | 0     | 0         |
| 1  | 1  | 1     | 1         |
| 0  | 1  | 0     | 1         |
| 0  | 1  | 1     | 1         |
| 1  | 0  | 0     | 0         |
| 1  | 0  | 1     | 0         |
| 0  | 0  | 0     | -         |
| 0  | 0  | 1     | -         |

動作のまとめ

| NS | NR | $Q_{n+1}$ |           |
|----|----|-----------|-----------|
| 1  | 1  | $Q_n$     | : Keep    |
| 0  | 1  | 0         | : Reset   |
| 1  | 0  | 1         | : Set     |
| 0  | 0  | -         | : Inhibit |

}  $Q = \bar{Q} = 1$  となるので  
使用不能となる。



カルノー図

| NS \ NR \ $Q_n$ | 0      | 1      |
|-----------------|--------|--------|
| 00              | $\phi$ | $\phi$ |
| 01              |        |        |
| 11              |        | 1      |
| 10              | 1      | 1      |

特性方程式

$$Q_{n+1} = \overline{NR} + NS \cdot Q_n$$

ただし  $\overline{NS} \cdot \overline{NR} = 0$  は禁止

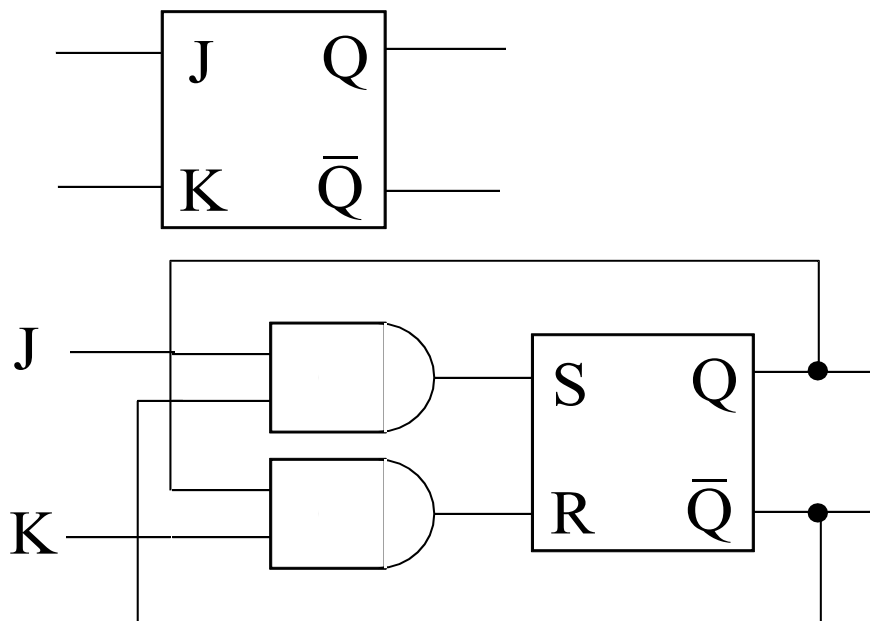


# FFの有用性と種類

- ▶ FFは順序回路を構成するための記憶回路を実現するために重要
- ▶ FFだけでも有用な(順序)回路が構成できる。
  - カウンタ、シフトレジスタ、ラッチ、
  - 分周期…
- ▶ FFの種類
  - RS-FF Reset/Setの入力・禁止状態あり
  - JK-FF RS-FFの禁止条件を緩和した回路
  - T-FF Trigger Flip Flop
  - D-FF Delay Flip Flop

# JK-FF

RS-FFの禁止条件を緩和したもの



真理値表

| J | K | $Q_n$ | $Q_{n+1}$ |
|---|---|-------|-----------|
| 0 | 0 | 0     | 0         |
| 0 | 0 | 1     | 1         |
| 0 | 1 | 0     | 0         |
| 0 | 1 | 1     | 0         |
| 1 | 0 | 0     | 1         |
| 1 | 0 | 1     | 1         |
| 1 | 1 | 0     | 1         |
| 1 | 1 | 1     | 0         |



カルノー図

| JK \ $Q_n$ | 0 | 1 |
|------------|---|---|
| 00         |   | 1 |
| 01         |   | 1 |
| 11         | 1 |   |
| 10         | 1 | 1 |



特性方程式

$$Q_{n+1} = \bar{K}Q_n + J\bar{Q}_n$$

動作のまとめ

| J | K | $Q_{n+1}$   |          |
|---|---|-------------|----------|
| 0 | 0 | $Q_n$       | : Keep   |
| 0 | 1 | 0           | : Reset  |
| 1 | 0 | 1           | : Set    |
| 1 | 1 | $\bar{Q}_n$ | : Invert |



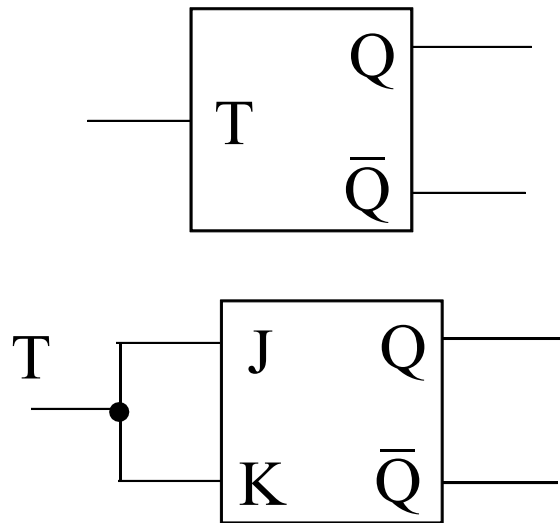
# JK-FFの名前の由来

- ▶ 後のノーベル物理学賞受賞者であるジャック・キルビー (Jack Kilby) がこの回路の開発に携わった際、セット用およびリセット用の入力端子の名前としてJとKを割り当てた。
  - <http://www.sophia-it.com/content/JK-FF>
- ▶ JK-FF の名前の由来は、Q (Queen) を J (Jack) と K (King) が奪い合うからだそうです。
  - <http://www.finetune.co.jp/~lyuka/technote/logic/logic-8.html>
- ▶ 名前はJack Knife (ジャックナイフ; 大型の折り畳み式ナイフ) のような反転動作を持つことに由来している。
  - <http://www.wdic.org/w/SCI/JKFF>
- ▶ 要するに語源が不明なのである。-- Wikipedia



# T-FF

Trigger-Flip Flop; 入力信号は一つ。(J=K)で、「トリガー」(=引き金)として、パルスが入るたびに、出力信号Qが反転する。



真理値表

| T | $Q_n$ | $Q_{n+1}$ |
|---|-------|-----------|
| 0 | 0     | 0         |
| 0 | 1     | 1         |
| 1 | 0     | 1         |
| 1 | 1     | 0         |



特性方程式

$$Q_{n+1} = \bar{T}Q_n + T\bar{Q}_n$$

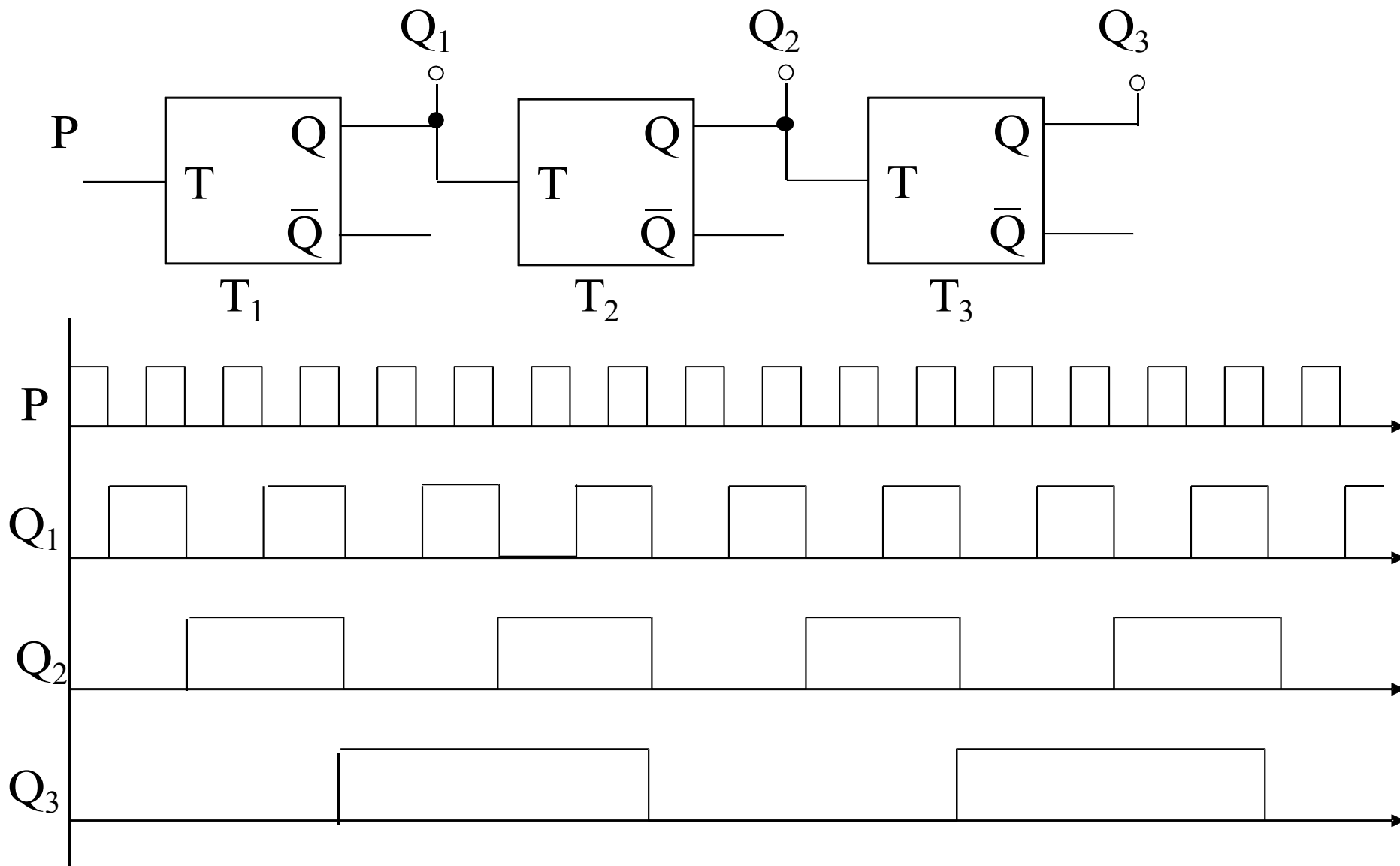
動作のまとめ

| T | $Q_{n+1}$   |          |
|---|-------------|----------|
| 0 | $Q_n$       | : Keep   |
| 1 | $\bar{Q}_n$ | : Invert |



# T-FF の応用: 分周器

注) 前段の変化が次々に影響が波及⇒ 直流駆動型、非同期式カウンタなど





# D-FF

Delay-Flip Flop; クロック信号が入るたびに、一つ前の入力信号値が出力される。(遅延をつくるフリップフロップ)

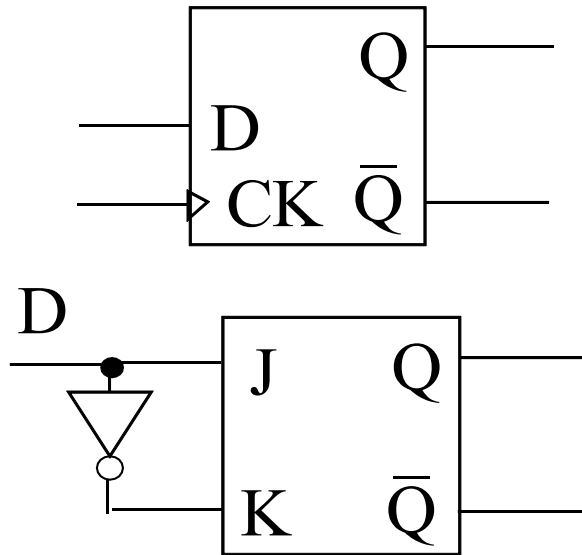
真理値表

| D | $Q_n$ | $Q_{n+1}$ |
|---|-------|-----------|
| 0 | 0     | 0         |
| 0 | 1     | 0         |
| 1 | 0     | 1         |
| 1 | 1     | 1         |



特性方程式

$$Q_{n+1} = D$$

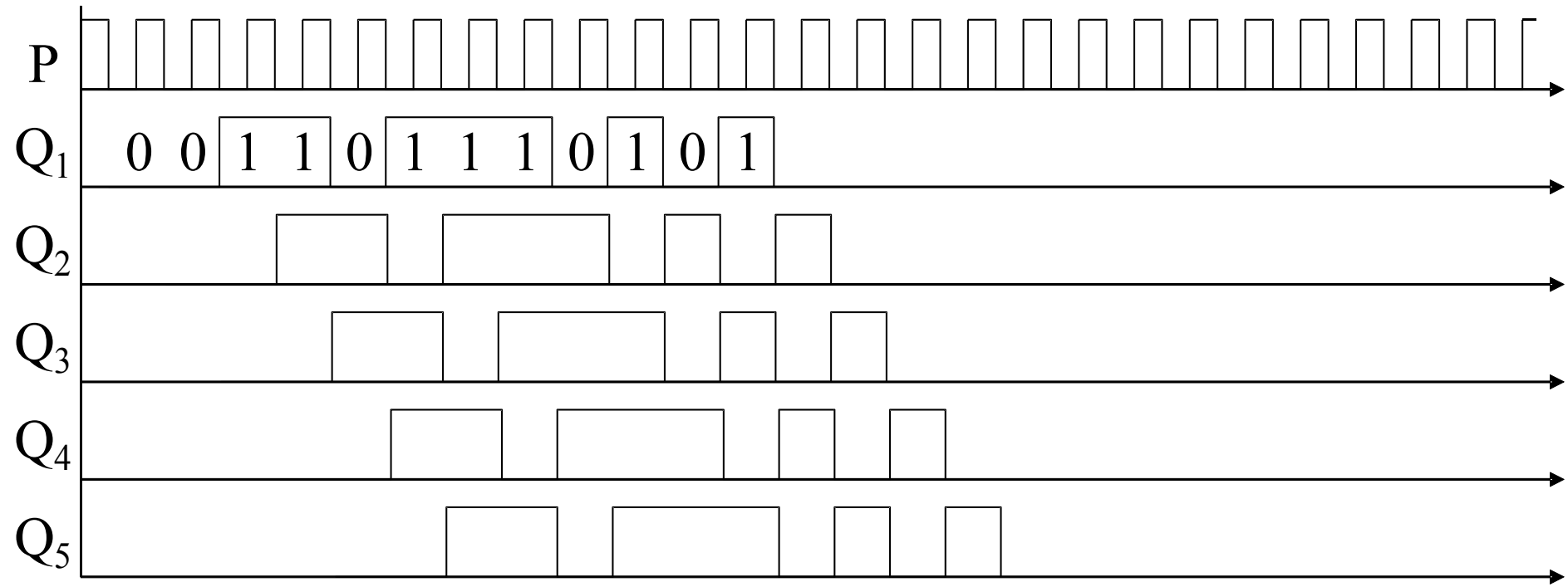
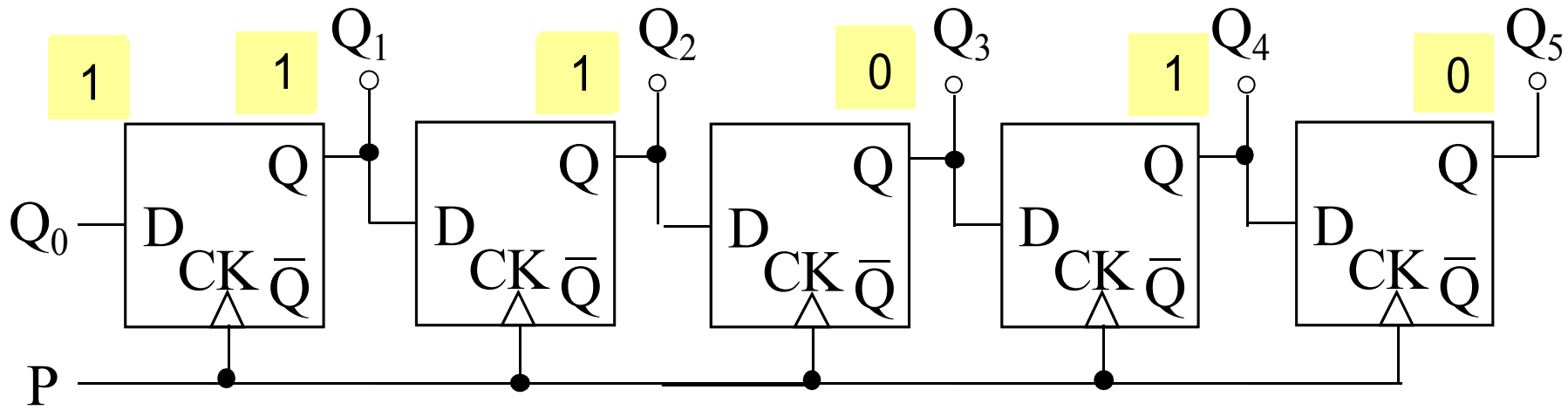


動作のまとめ

| D | $Q_{n+1}$ |         |
|---|-----------|---------|
| 0 | 0         | : Reset |
| 1 | 1         | : Set   |



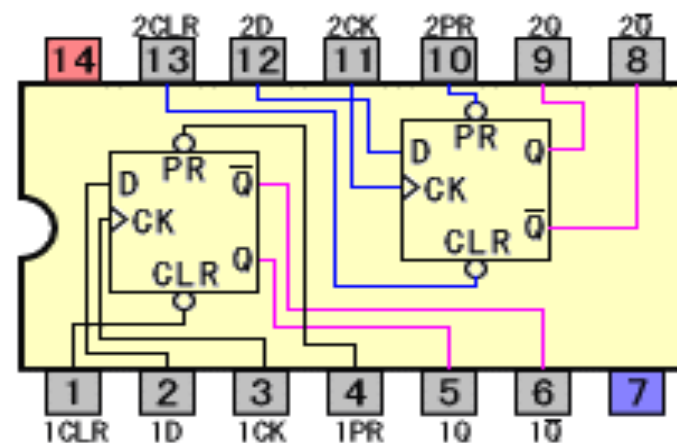
# D-FF の応用: シフトレジスタ





# 7474の使い方

- ▶ 14番(VCC)と7番(GND)は必須。
- ▶ 動作させる時は、CLRに"1"(H)を入力する。
  - CLRは負論理なので、0で CLR=true
  - CLR=trueの時y、出力は常に"0"(L)
- ▶ 初期値を設定(Q="1")する時は、PRに"0"(L)を入力する。
  - PRは負論理なので、0で PR=true
  - PR=trueの時、Qは"1"(H)
  - 初期値を設定しない時は、(H)に接続しておく



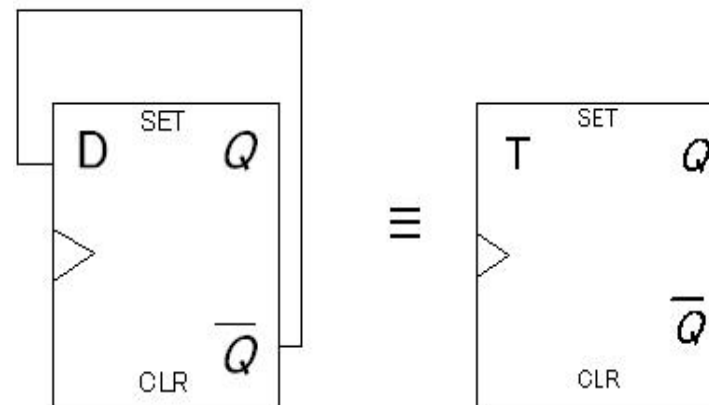
外側に○がついた端子は、「負論理」(Lでtrue)で動作する

PRはPReset  
CLRはCLear



# 7474をT-FFとして使う

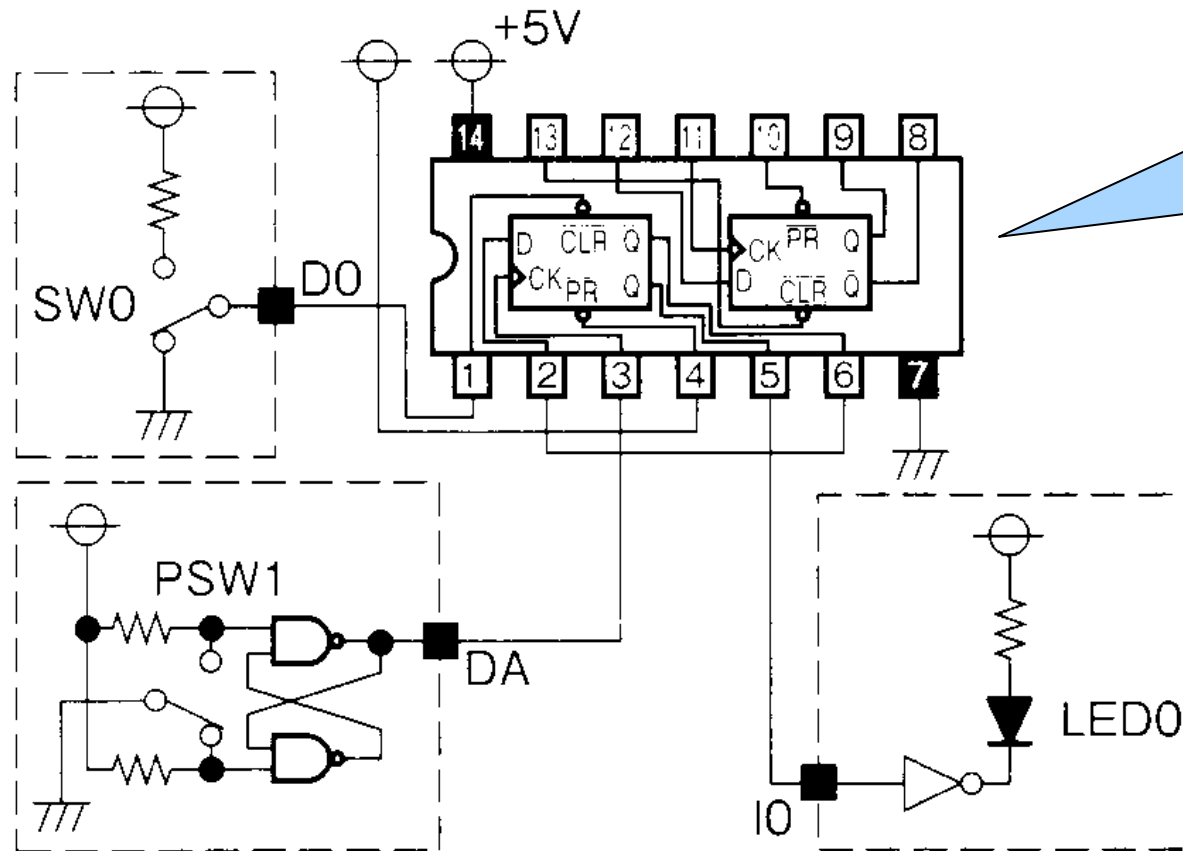
- ▶ not QをDに入れる。
- ▶ クロックが入るたびに、現在の値の反転 (not Q) がDelay信号に入るため、結果的に、クロックが入るたびに出力が反転する、T-FFとして使うことができる。





# 練習課題

## ▶ Dフリップフロップの実験



notQがDに入っているため、T-FFとして動作する。

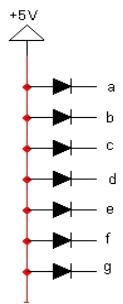
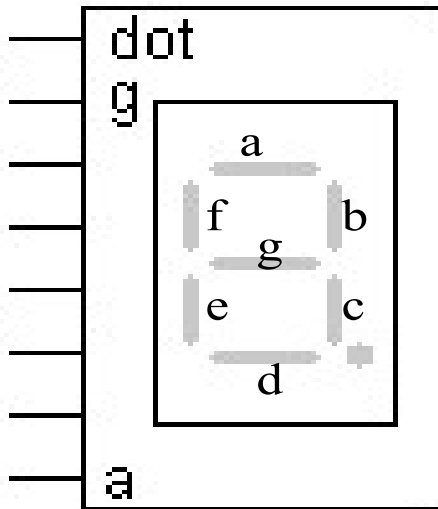
プッシュSWを押すたびに、Qが反転

図 3-19 D-FF 回路の接続図



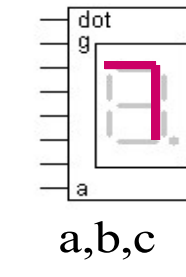
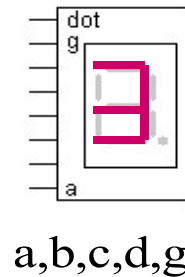
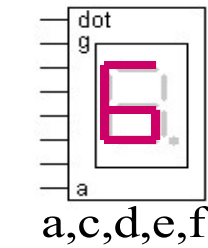
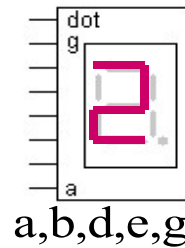
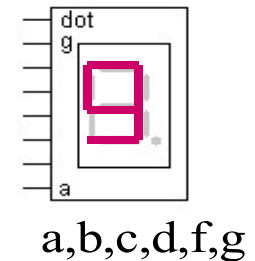
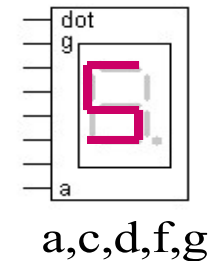
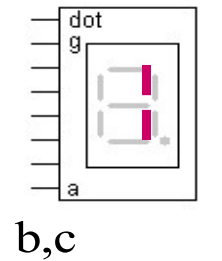
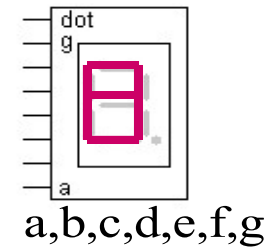
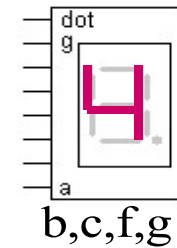
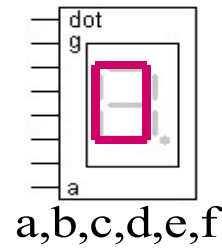
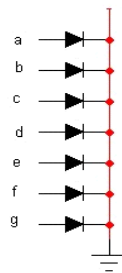
# 7 segment LED

## BCD-7セグメントデコーダ



アノードコモン型

カソードコモン型



カソードコモン型ではセグメント入力に“1”を入れると発光。

アノードコモン型では“0”とすると発光。



# 7447(7Segment Decoder)

## ▶ 「数値」を表示するための素子

- 7 Segment LED用のDecoder
- 7つのエレメント(セグメント)の組み合わせで、数字を表示する。
- 例えば、0を表示する時は、abcdefの6つのセグメントを発光させ、残りを消灯させる。
- 例えば、1を表示する時は、bcの二つを発光
- この素子を簡便に使うために、2進数の0が入力されるとabcdefが、1が入力されるとbcが、LOWとなるような、「デコーダ」がIC化されている。
- このデコーダに、信号を渡すことで、「数字」が表示できる。



# 報告課題(1)

- ▶ 4進カウンタ回路(同期式)回路を作る。
  - 7セグLEDに数値が表示されることを確認しよう。
- ▶ 考察事項
  - 回路図を元にして、初段と2段目の関係を明記する。
  - 非同期式の回路と、相違点を整理して報告する。

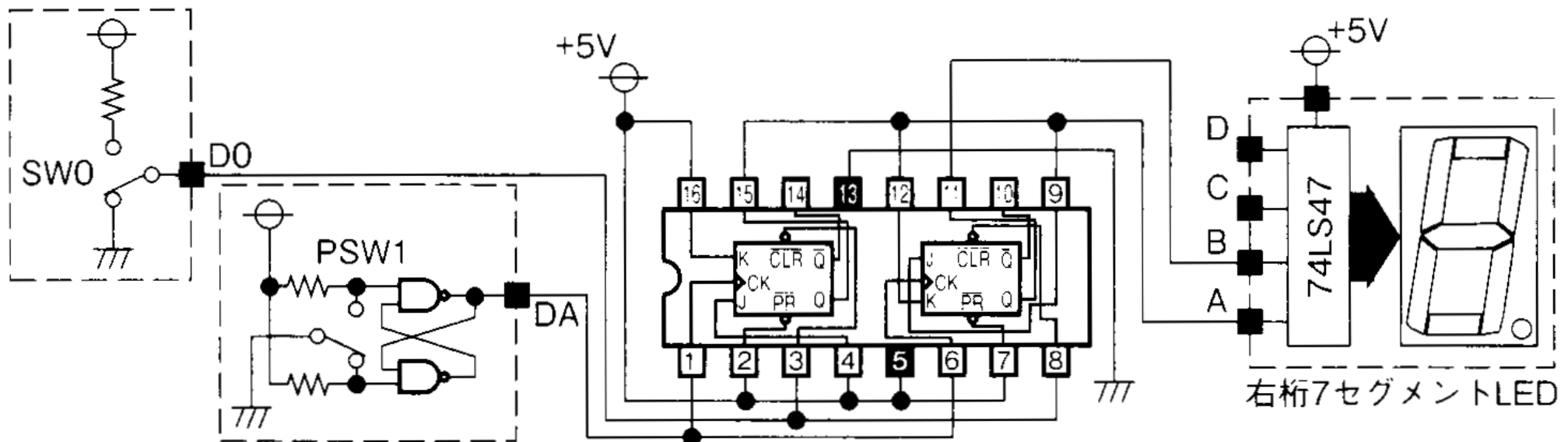


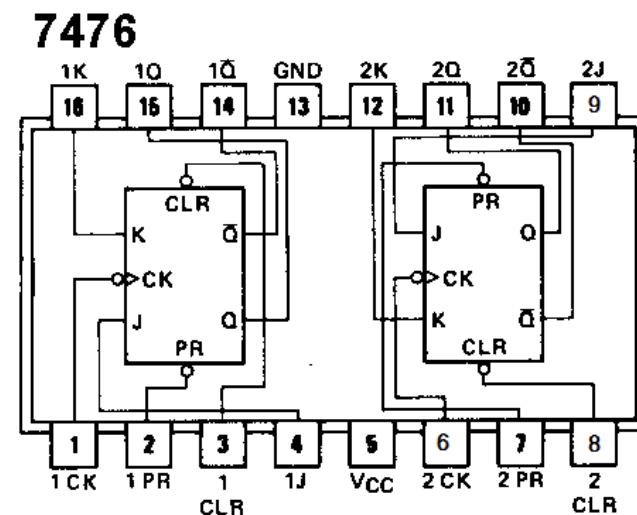
図 3-25 4進カウンタ回路(同期式)の接続図



# 同期式回路の特徴

- ▶ CLR入力は、データSW(回路動作のON/OFF)
- ▶ 初段のJとKは、"1"(H)に接続
  - CLK入力に同期して、出力を反転
- ▶ 2段目のJとKは、1段目のQを受け取る
  - 初段のQ=1の時、Clock入力で、反転する。
  - (分周される)

非同期式はどうなっている？  
(レポートのヒントです。)





# 報告課題(2)

## ▶ 10進カウンタを作る

- 7490について調べる。
- 0~9の数値を表示する動作を確認しよう。

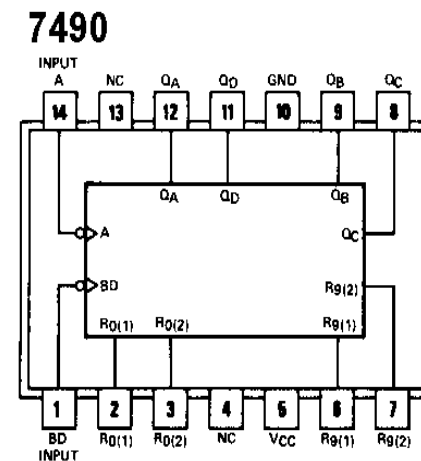
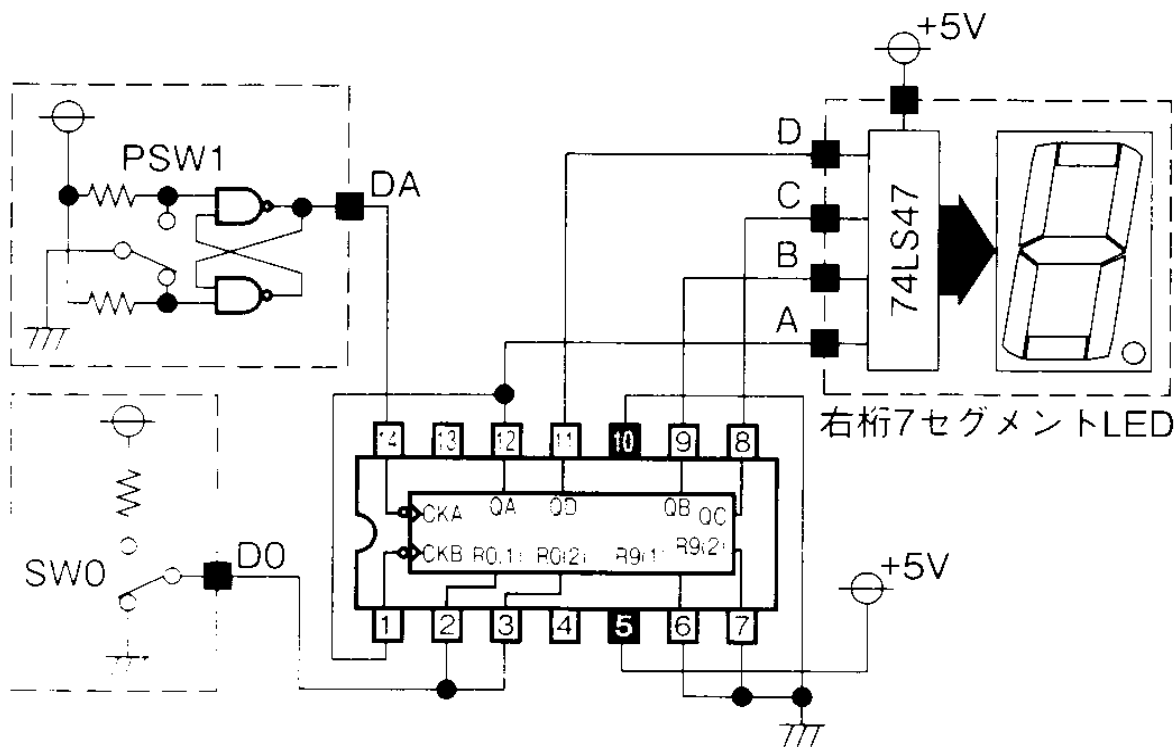


図 3-27 10進カウンタ回路の接続図





# ICトレーナの手引きの誤植

ICトレーナの手引き、P31、図3-28に誤植があります。

7490の配線の「配線例」の絵で、11番がGNDに接続され、10番がCN6のDに接続されていますが、逆です。

10番をGNDに接続し、11番をCN6のDに接続するのが正しいので、訂正してお読み下さい。



# 発展課題

## ▶ 実験室の7474を3個使う。(6段)

- 6段のシフトレジスタを作ろう。
- この教材の、P16の回路図を実装する。
  - notCLRの端子には、「動作SW」を接続する。
    - 6個のレジスタ全部に一つのSWの信号を入れる。
  - notPRの端子には、「初期値設定用」のSWを、それぞれのゲートに一つずつ(6個のSWから)接続する。
  - Qを連続するLEDに接続する。
- 使い方
  - notCLRをONにする。
  - 個々のPR(初期設定)でON/OFFを入力
  - 手動Clockを押す。
- 動作：
  - 電光掲示板サインのようにLEDが流れる。
- S評価

# フリーの回路図エディタについて

- ▶ 水魚堂から無料の回路図エディタが出ています。
- ▶ <http://www.suigyodo.com/online/schsoft.htm>
- ▶ レポートに回路図を添付する際に試してみてください。



# 課題レポートについて

## ▶ 報告課題(1) 4進カウンタ回路(同期式)

## ▶ 報告課題(2) 10進カウンタ回路

- 回路の動作について報告して下さい。
  - 写真、動画OK
- 4進カウンタでは、同期式と非同期式の違いについて、考察して下さい。
- 今回の実験を通じて理解した内容(フリップフロップについて、7セグメントLED、74系のIC、カウンタ回路について、など)について、報告して下さい。
- ここまでの内容が優れていれば、A評価とします。

## ▶ 発展課題:

- 作成したシフトレジスタの動作について、報告して下さい。(結果報告だけ、動けばOKで、この発展課題については考察は不要です。)



# 次回の予告

- ▶ ICトレーナのサンプル回路を最後まで試します。
- ▶ 次週に予定しているのは以下の回路です。
  - エンコーダ
  - デコーダ
  - 100進カウンタ
  - 電子サイコロ