

電気電子システム学科

# コンピュータ工学 I

Part 2

Rev. 2019.12.23

# 論理回路

---

## ❖ 内容

- ① 論理演算 (ブール代数)
- ② ブール代数の公理
- ③ 論理式の簡単化
- ④ 組み合わせ論理回路
- ⑤ 順序回路

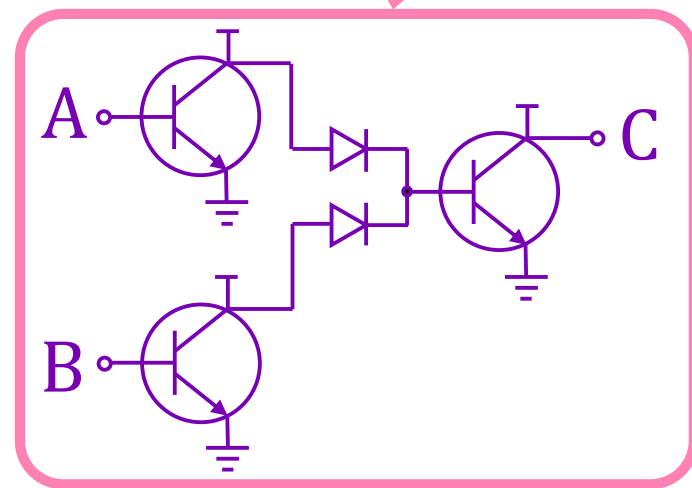
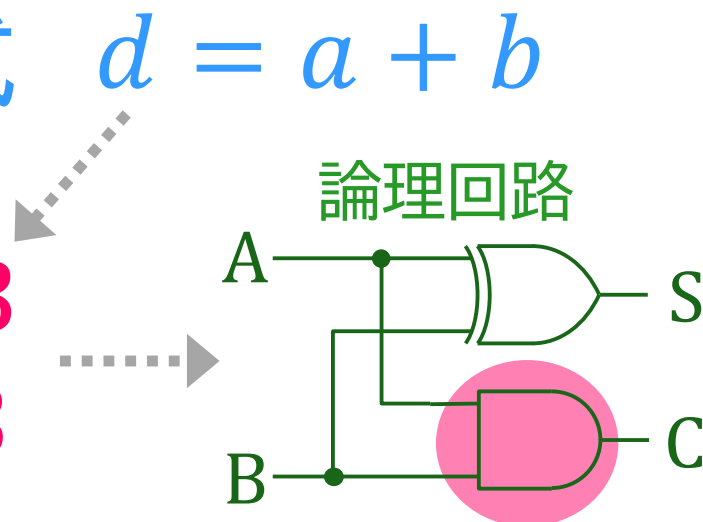
# 演算回路の設計

① 回路化する計算式  $d = a + b$

② 論理式  $S = A \oplus B$   
 $C = A \cdot B$

③ 論理回路

④ デジタル回路



# 論理演算(ブール代数)

重要

真(true)と偽(false)の2状態を扱う演算

論理値

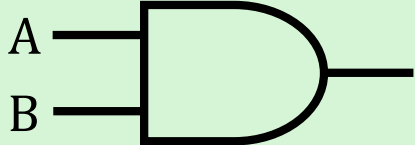
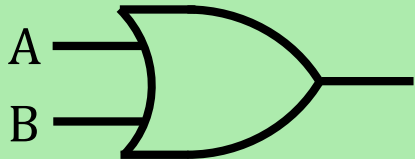
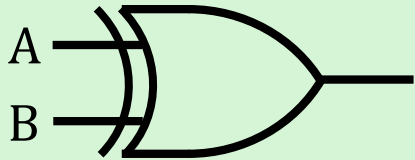
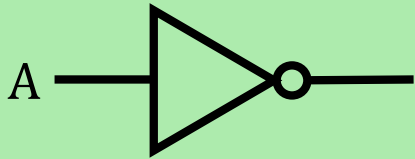
真 ... 1

偽 ... 0

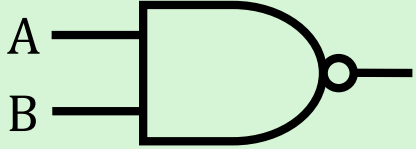
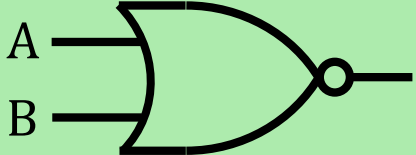
2進数(0, 1)の演算の実現に適している。

# 基本論理演算

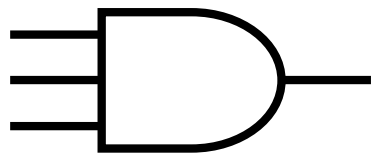
重要

	演算記号	回路記号
論理積 AND	$A \cdot B$	 The diagram shows a standard AND gate symbol with two input lines labeled 'A' and 'B' on the left and one output line on the right.
論理和 OR	$A + B$	 The diagram shows a standard OR gate symbol with two input lines labeled 'A' and 'B' on the left and one output line on the right.
排他的論理和 Exclusive OR (XOR)	$A \oplus B$	 The diagram shows a standard XOR gate symbol with two input lines labeled 'A' and 'B' on the left and one output line on the right.
否定 NOT	$\bar{A}$	 The diagram shows a standard NOT gate symbol with one input line labeled 'A' on the left and one output line on the right.

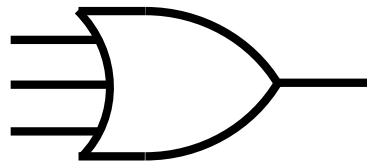
# 基本論理演算

	式の表記	回路記号
否定論理積 NAND	$\overline{A \cdot B}$	
否定論理和 NOR	$\overline{A + B}$	

## 3入力の場合



$$A \cdot B \cdot C$$



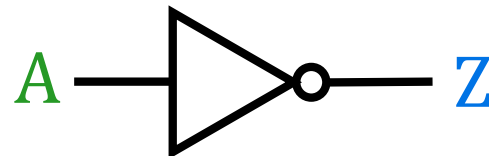
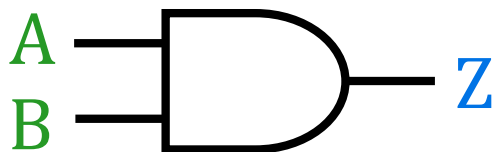
$$A + B + C$$

# 基本論理演算の真理値表

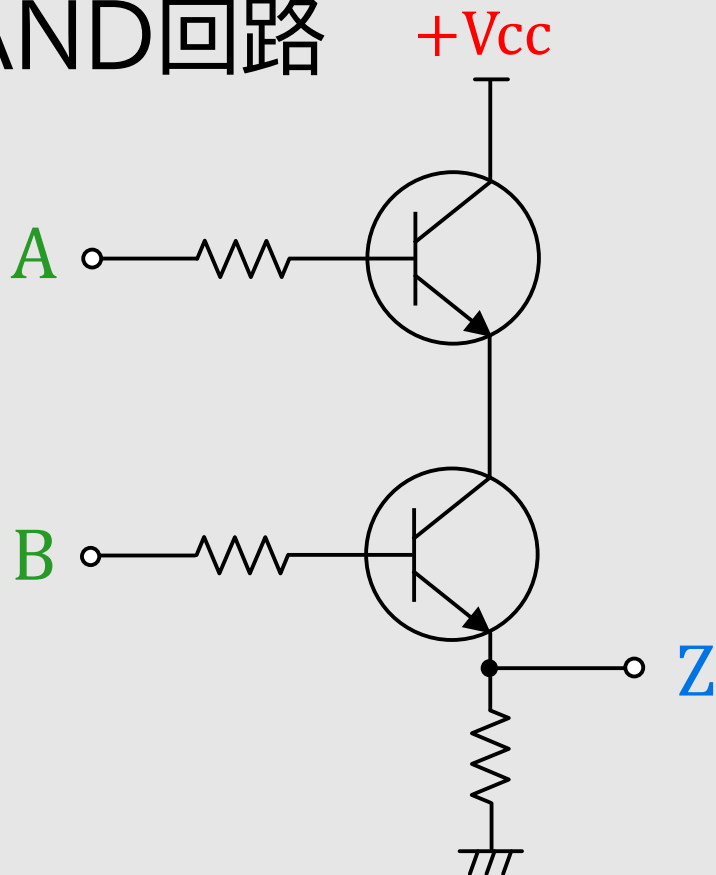
重要

入力値		出力値			
A	B	$A \cdot B$	$A + B$	$A \oplus B$	$\bar{A}$
0	0	0	0	0	1
0	1	0	1	1	
1	0	0	1	1	0
1	1	1	1	0	

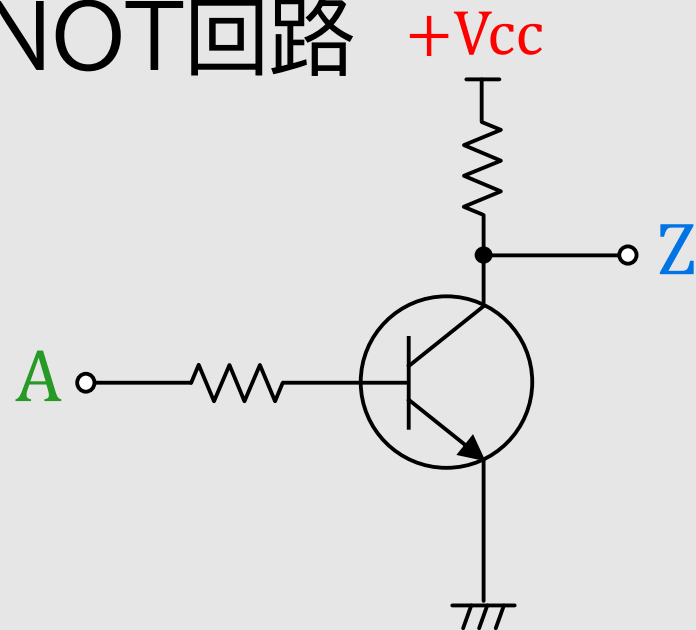
# トランジスタによる論理回路の構成



AND回路



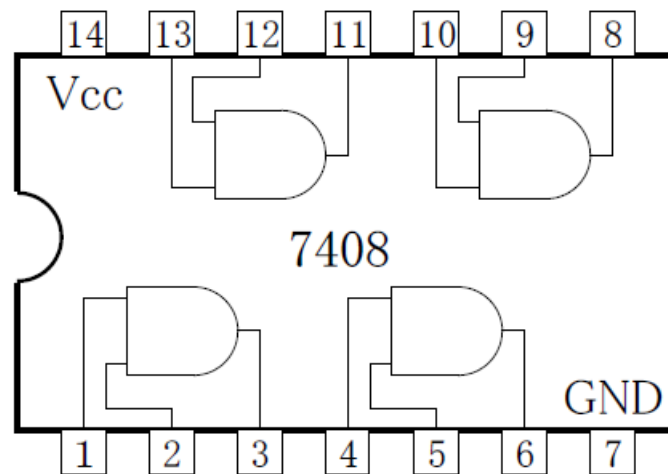
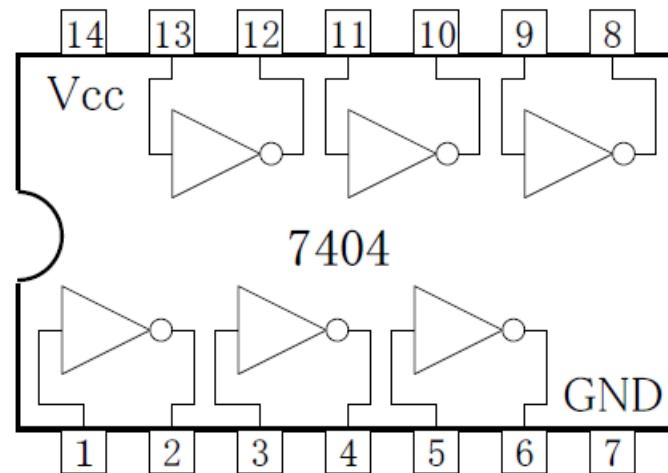
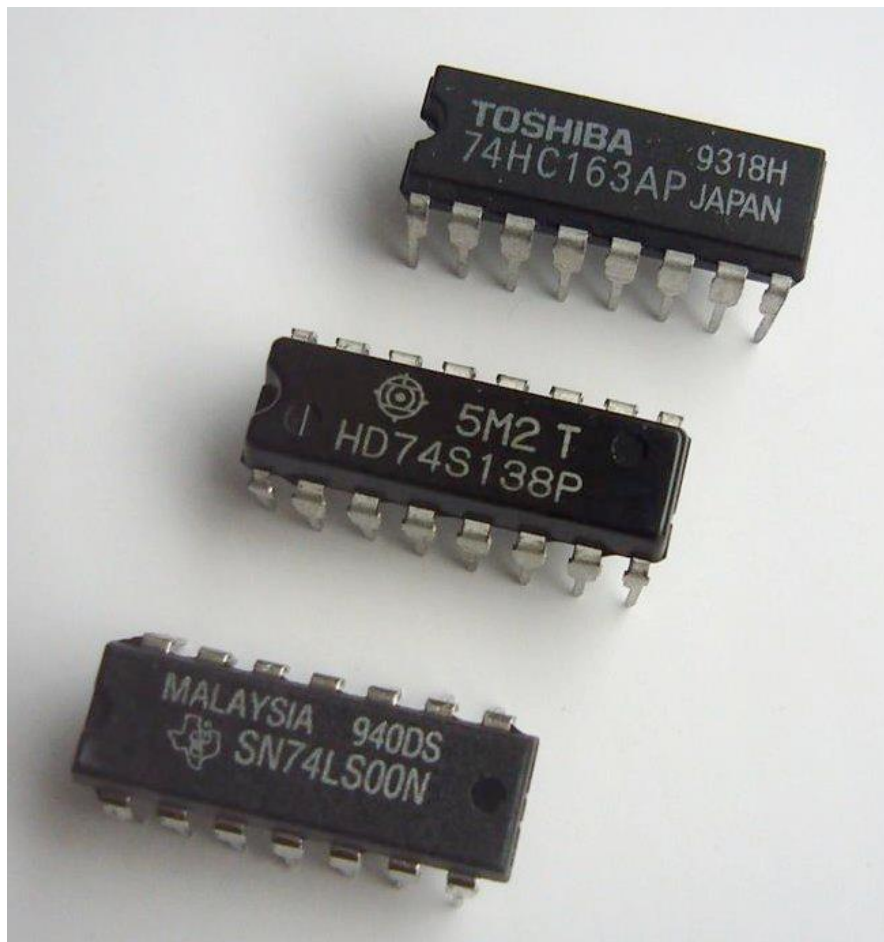
NOT回路



※ 回路構成の一例



# ロジックIC (集積回路)



引用: [jp.wikipedia.org/wiki/汎用ロジックIC](http://jp.wikipedia.org/wiki/汎用ロジックIC)

# ビット演算

重要

```
    1001
AND 1010
-----
    1000
```

```
    1001
OR  1010
-----
    1011
```

```
    1001
XOR 1010
-----
    0011
```

```
    NOT 1001
-----
    0110
```

# C言語の論理演算子

参考

## ビット演算子

演算名	記号	例
AND	&	a & b
OR		a   b
XOR	^	a ^ b
NOT	~	~a

1bitごとに論理演算を行う。

## 結合演算子

演算名	記号	例
AND	&&	a>0 && b>0
OR		a>0    b>0
XOR		なし
NOT	!	!(a>0)

非0を真、0を偽とみなして、論理演算を行う。

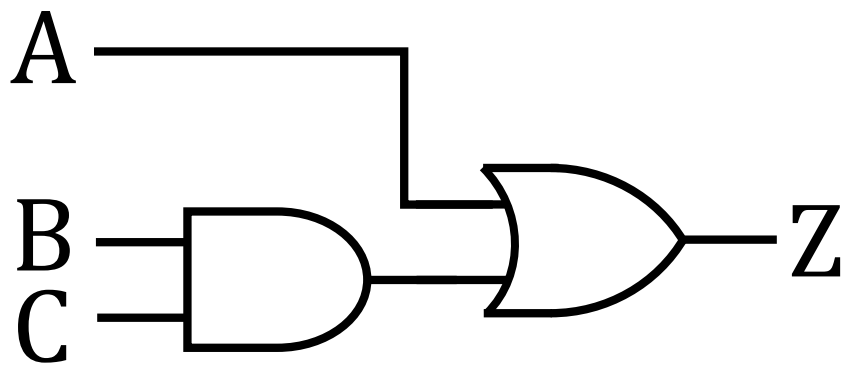
# 論理式の計算

## 論理式

$$Z = A + \underline{B \cdot C}$$

AND演算は、OR演算よりも演算順位が高い。

## 論理回路図



$A = 0, B = 1, C = 1$   
の場合

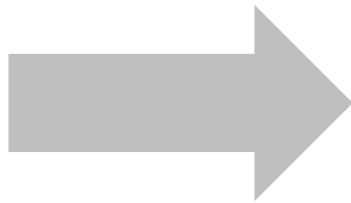
$$\begin{aligned} Z &= A + (B \cdot C) \\ &= 0 + (1 \cdot 1) \\ &= 0 + 1 \\ &= 1 \end{aligned}$$

# 真理値表の作成(2入力)

## $\bar{A} + B$ の真理値表

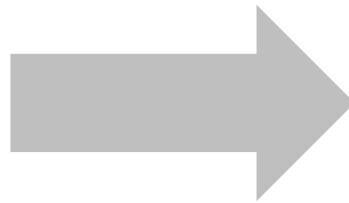
入力値のすべての組み合わせを列挙する。

A	B
0	0
0	1
1	0
1	1



Aから $\bar{A}$ を  
求める

$\bar{A}$
1
1
0
0



$\bar{A}$ とBから  
 $\bar{A} + B$ を  
求める

A	B	$\bar{A} + B$
0	0	1
0	1	1
1	0	0
1	1	1

# 真理値表の作成(3入力)

## A + B · C の真理値表

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



B · C
0
0
0
1
0
0
0
1



A	B	C	A + B · C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

# ブール代数の公理・定理

重要

✦ ベキ等則

✦ 交換則

✦ 結合則

✦ 吸収則

✦ 分配則

✦ 二重否定

✦ ド・モルガン則

✦ 単位元

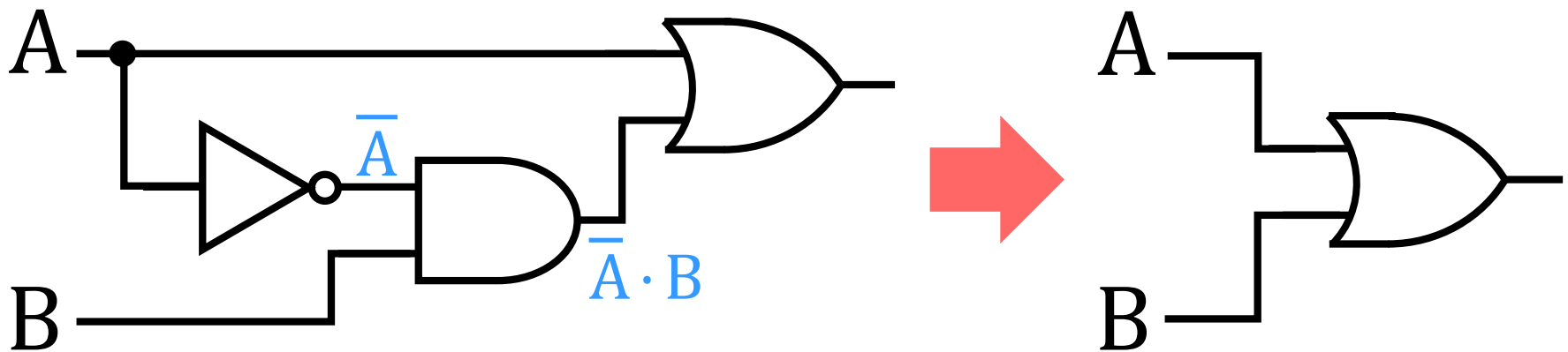
✦ 零元

✦ 補元

公式は別紙資料を参照せよ。

# 論理式の簡単化

$$\begin{aligned} & A + \bar{A} \cdot B \\ &= (A + \bar{A}) \cdot (A + B) \\ &= 1 \cdot (A + B) \\ &= A + B \end{aligned}$$



論理回路の構成部品を削減できる。



# 演算回路の作成手順

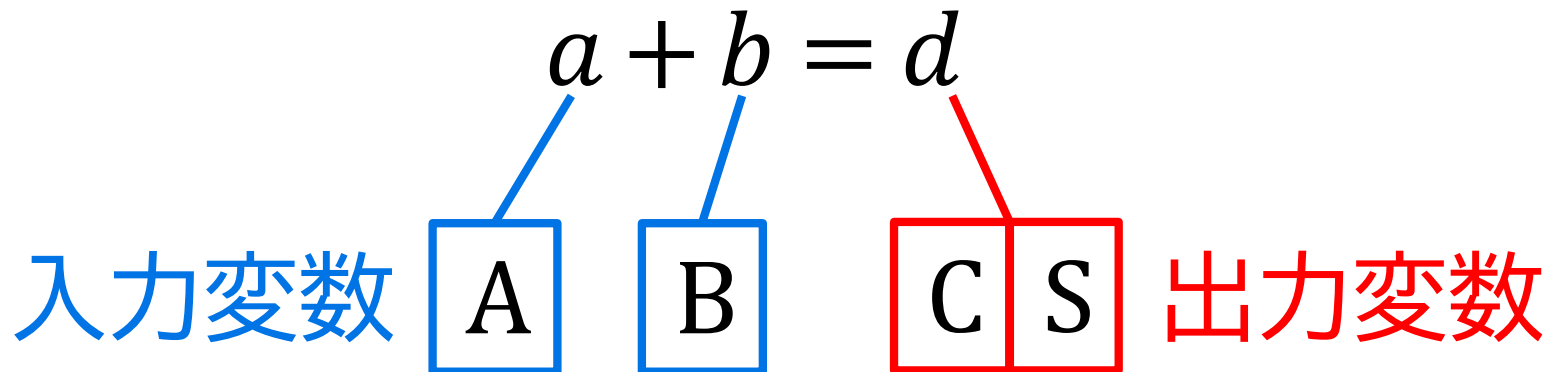
---

- ① 真理値表の作成
- ② 論理式の組み立て
- ③ 論理式の簡単化
- ④ 論理回路への置き換え

# 半加算器

1 bitの2つの値の加算を行う論理回路。  
計算結果は2 bitで表す。

## 回路化する加算の式



1bitごとに論理変数を割り当てる。

# 半加算器の真理値表

重要

入力値のすべての組み合わせを列挙する。

$$a + b = d$$

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

真理値表に  
まとめる。



A

B

C S

真理値表

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

# 半加算器の論理式と論理回路

重要

真理値表

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

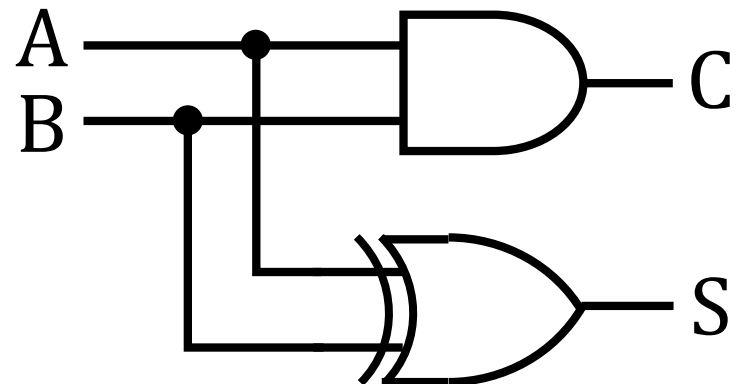
真理値表から論理式を導出する。

論理式

$$C = A \cdot B$$

$$S = A \oplus B$$

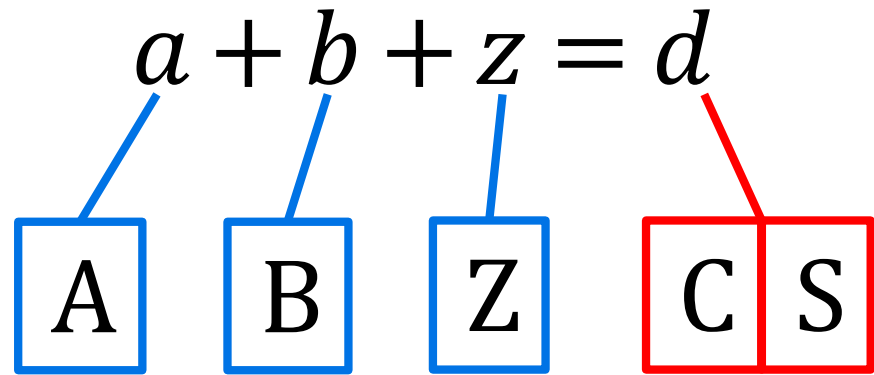
論理回路図



# 全加算器

1 bitの3つの値の加算を行う論理回路。

3つ目の値は、下位bitからの桁上げに相当する。



真理値表

A	B	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

# 論理式の加法標準形

重要

A	B	Z	C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>
0	0	0	0
0	0	0	0
0	0	0	0
1	0	0	0
0	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

左のように C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub>, C<sub>4</sub> を割り当てると、

$$C = C_1 + C_2 + C_3 + C_4$$

$\bar{A} = 1, B = 1, Z = 1$  のとき C<sub>1</sub> = 1 なので、

$$C_1 = \bar{A} \cdot B \cdot Z$$

論理積項

加法標準形

$$C = \bar{A} \cdot B \cdot Z + A \cdot \bar{B} \cdot Z + A \cdot B \cdot \bar{Z} + A \cdot B \cdot Z$$

# 論理式の簡単化

$$C = \bar{A} \cdot B \cdot Z + A \cdot \bar{B} \cdot Z + A \cdot B \cdot \bar{Z} + A \cdot B \cdot Z$$

↓ べき等則

$$= \bar{A} \cdot B \cdot Z + A \cdot B \cdot Z + A \cdot \bar{B} \cdot Z + A \cdot B \cdot Z + A \cdot B \cdot \bar{Z} + A \cdot B \cdot Z$$

↓ 分配則

$$= B \cdot Z \cdot (\bar{A} + A) + A \cdot Z \cdot (\bar{B} + B) + A \cdot B \cdot (\bar{Z} + Z)$$

↓ 補元

$$= B \cdot Z \cdot 1 + A \cdot Z \cdot 1 + A \cdot B \cdot 1$$

↓ 単位元

$$= B \cdot Z + A \cdot Z + A \cdot B$$

交換則

$$= A \cdot B + B \cdot Z + A \cdot Z$$

# 全加算器の論理式

---

$$C = A \cdot B + B \cdot Z + A \cdot Z$$

$$S = \bar{A} \cdot \bar{B} \cdot Z + \bar{A} \cdot B \cdot \bar{Z} + A \cdot \bar{B} \cdot \bar{Z} + A \cdot B \cdot Z$$

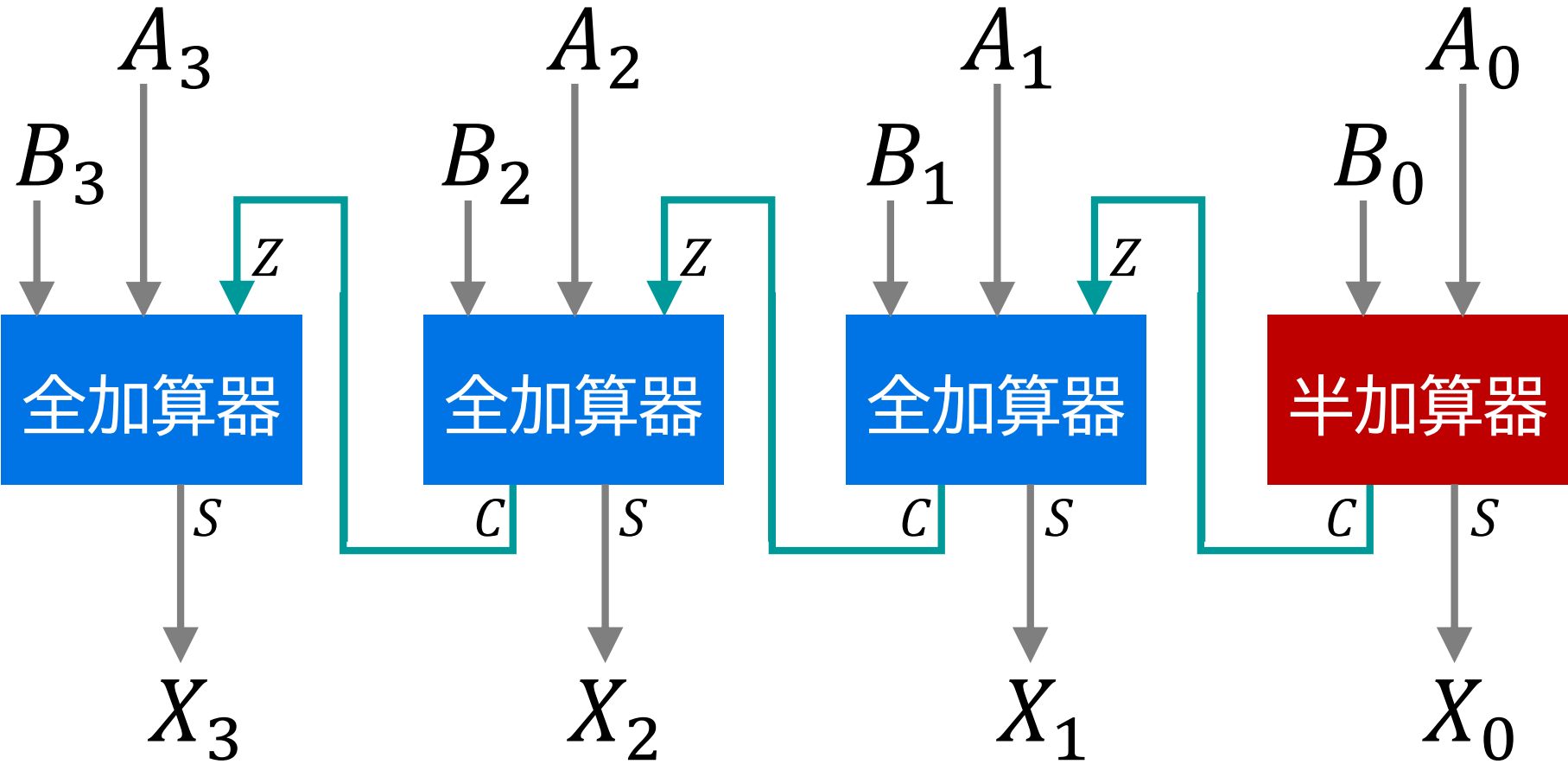
論理回路図

教科書 図3.17 を参照



# 多bit加算回路の構成

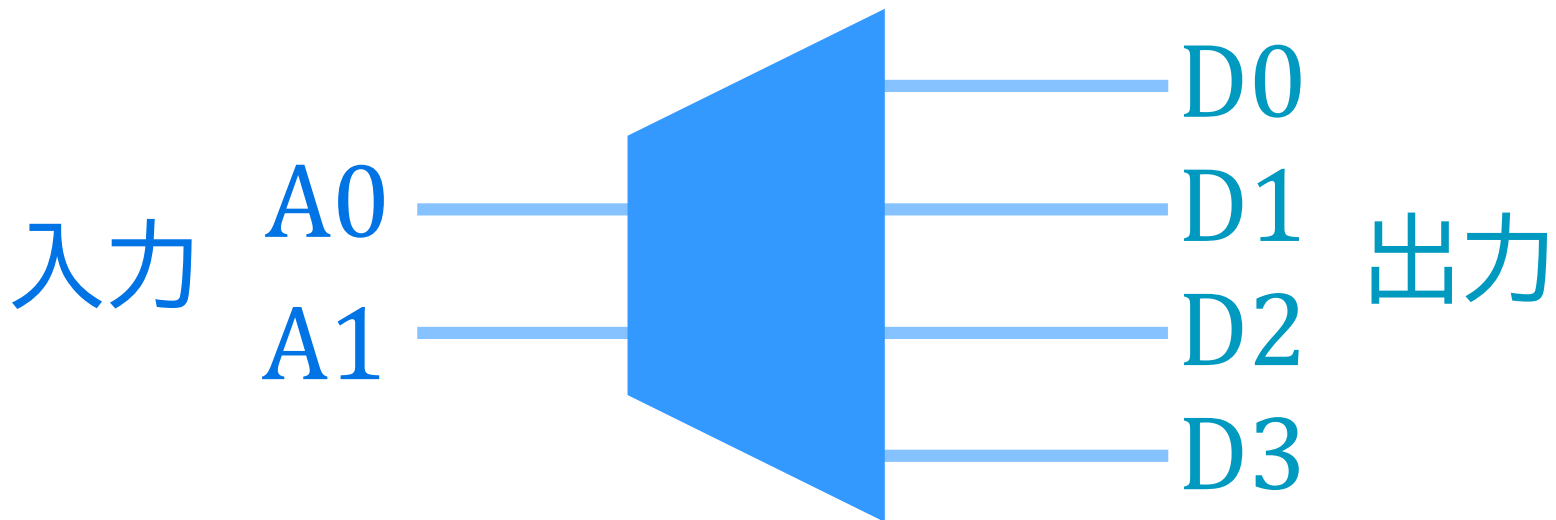
4bit 2進数  $A_3A_2A_1A_0$  と  $B_3B_2B_1B_0$  の加算



# デコーダ

重要

複数の入力信号からなる数値を解読して、出力信号パターンを生成する回路。



符号化された  
データ

復号



元のデータ

# 2入力4出力デコーダ

入力値 0 のとき、出力  $D0 = 1$

入力値 1 のとき、出力  $D1 = 1$

入力値 2 のとき、出力  $D2 = 1$

入力値 3 のとき、出力  $D3 = 1$

真理値表

A0	A1	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

論理式

$$D0 = \overline{A0} \cdot \overline{A1}$$

$$D1 = \overline{A0} \cdot A1$$

$$D2 = A0 \cdot \overline{A1}$$

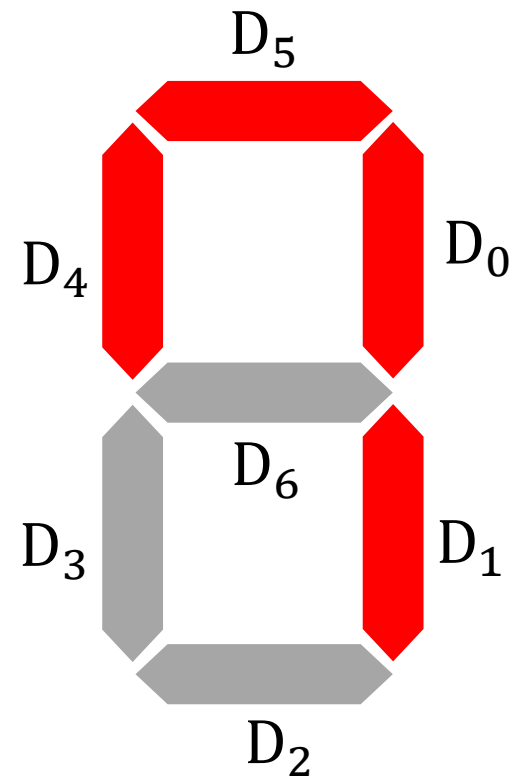
$$D3 = A0 \cdot A1$$

# 7セグメントデコーダ

参考

7セグメントLEDに数字を表示するために、2進数の入力値からLEDの点灯パターンを生成する回路。

A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1

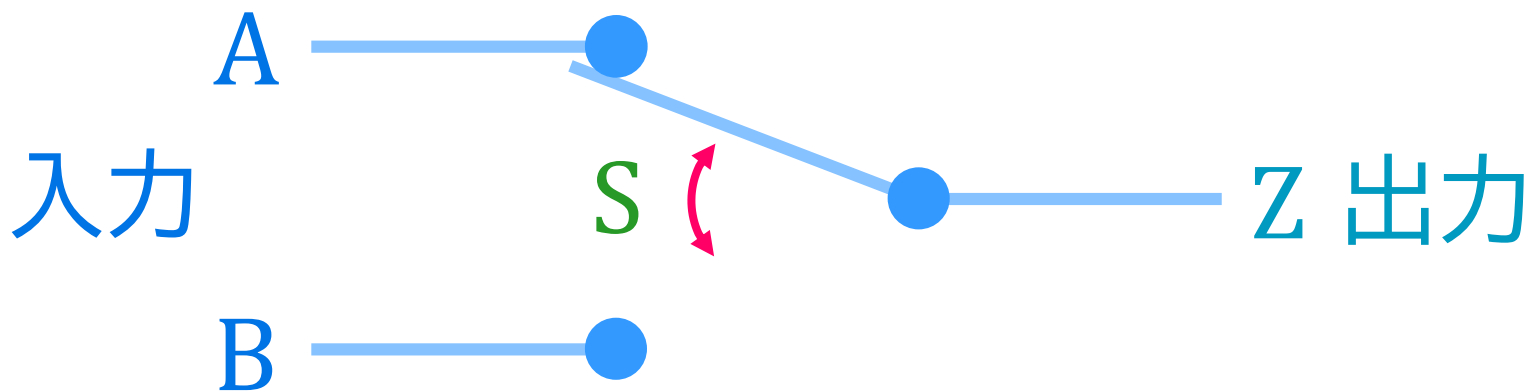


# マルチプレクサ

重要

複数の入力信号から1つを選んで出力する回路

動作のイメージ図(2入力の場合)



$S = 0$  のとき  $Z = A$

$S = 1$  のとき  $Z = B$

# 2入力1出力マルチプレクサ

真理値表

S	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

論理式

加法標準形

$$Z = \bar{S} \cdot A \cdot \bar{B} + \bar{S} \cdot A \cdot B + S \cdot \bar{A} \cdot B + S \cdot A \cdot B$$

簡単化した式

$$Z = \bar{S} \cdot A + S \cdot B$$

# カルノー図

---

論理式の簡単化を行うためのツール。

ブール代数の定理を使わずに、簡単化した論理式を導き出すことができる。

# 真理値表→カルノー図の変換

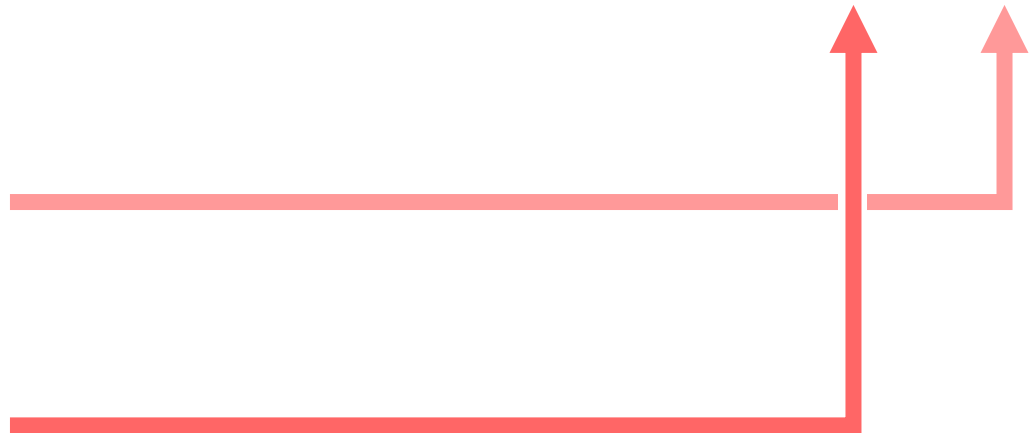
重要

真理値表

A	B	C	Z
0	0	0	①
0	0	1	②
0	1	0	③
0	1	1	④
1	0	0	⑤
1	0	1	⑥
1	1	0	⑦
1	1	1	⑧

カルノー図

		AB			
		00	01	11	10
C	0	①	③	⑦	⑤
	1	②	④	⑧	⑥





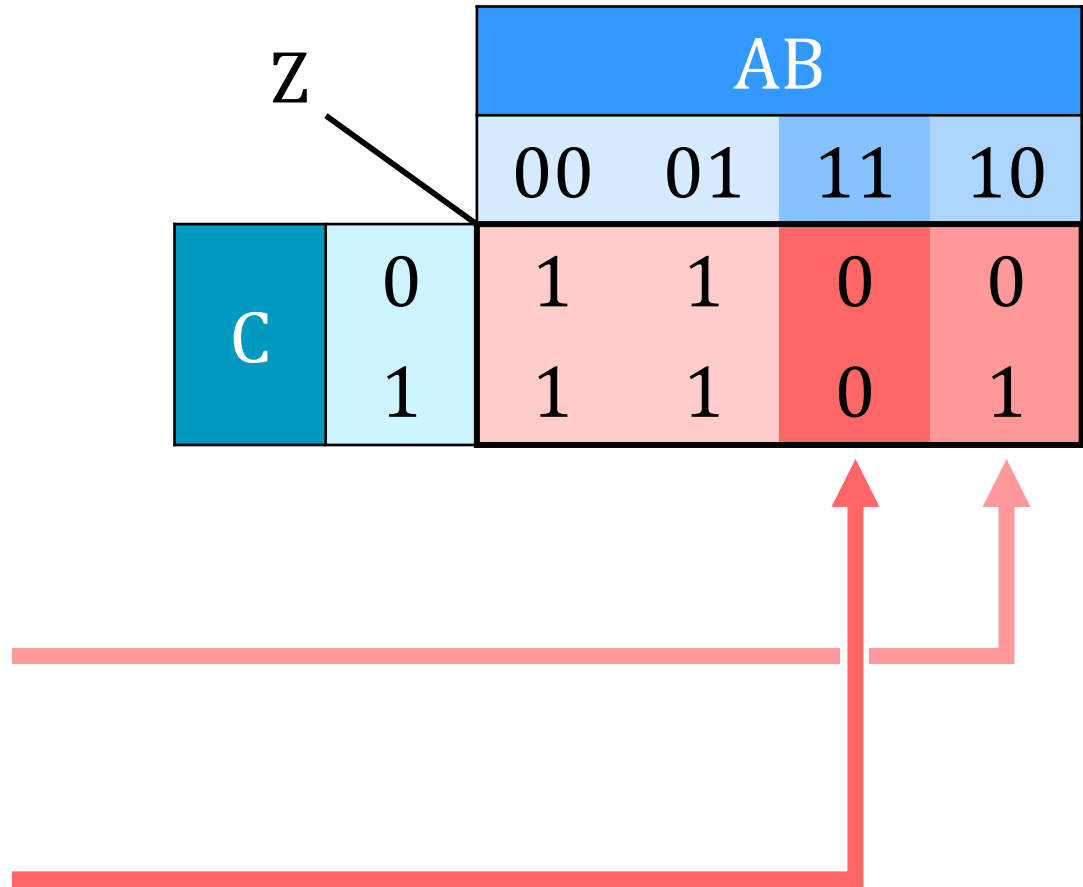
# 真理値表→カルノー図の変換

重要

真理値表

A	B	C	Z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

カルノー図



# カルノー図による簡単化①

重要

- ❖ 出力値の 1 だけを四角形で囲む。
- ❖ 四角形の幅は、1、2、4のいずれかにする。
- ❖ 可能限り大きな四角形で囲む。
- ❖ 四角形の数を最小にする。

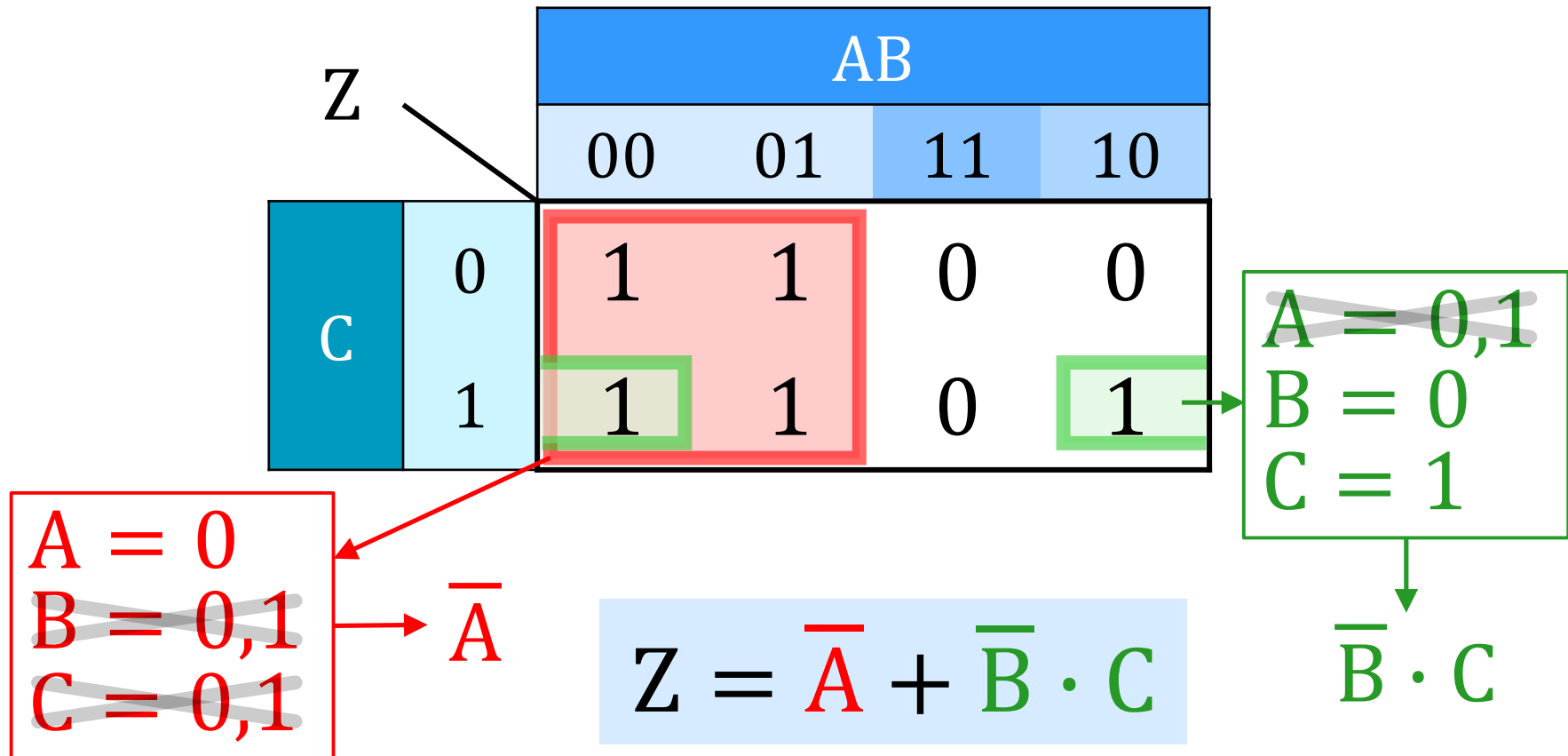
Z		AB			
		00	01	11	10
C	0	1	1	0	0
	1	1	1	0	1

上下・左右は繋がって四角形が重なっても良い。いると考える。

# カルノー図による簡単化②

**重要**

- ① 各四角形において、入力値が 0 または 1 の片方しか含まれていない変数で、論理積項を作る。
- ② すべての論理積項を論理和でつなげる。



# 全加算器のカルノー図

		AB			
		00	01	11	10
Z	0	0	0	1	0
	1	0	1	1	1

$$B \cdot Z$$

~~$$A = 0,1$$~~

$$B = 1$$

$$Z = 1$$

$$A \cdot B$$

$$A = 1$$

$$B = 1$$

~~$$Z = 0,1$$~~

$$A \cdot Z$$

$$A = 1$$

~~$$B = 0,1$$~~

$$Z = 1$$

		AB			
		00	01	11	10
S	0	0	1	0	1
	1	1	0	1	0

$$\bar{A} \cdot \bar{B} \cdot Z$$

$$\bar{A} \cdot B \cdot \bar{Z}$$

$$A \cdot B \cdot Z$$

$$A \cdot \bar{B} \cdot \bar{Z}$$

# カルノー図の囲み方の例

		AB			
		00	01	11	10
C	0	1	1	1	1
	1	0	$\bar{C}$	0	1

Diagram showing a Karnaugh map with a blue horizontal group covering the top row (C=0) and a red vertical group covering the right two columns (AB=11 and 10).

		AB			
		00	01	11	10
C	0	1	0	1	1
	1	0	$\bar{B} \cdot \bar{C}$	0	1

Diagram showing a Karnaugh map with a blue horizontal group covering the top row (C=0), a red vertical group covering the right two columns (AB=11 and 10), and a blue vertical group covering the first column (AB=00).

		AB			
		00	01	11	10
C	0	0	1	1	1
	1	0	$B \cdot \bar{C}$	0	1

Diagram showing a Karnaugh map with a blue horizontal group covering the top row (C=0), a red vertical group covering the right two columns (AB=11 and 10), and a blue horizontal group covering the middle two cells of the top row (AB=01 and 11).

		AB			
		00	01	11	10
C	0	1	$\bar{B}$	0	1
	1	1	1	1	1

Diagram showing a Karnaugh map with a blue horizontal group covering the top row (C=0), a red vertical group covering the right two columns (AB=11 and 10), a blue vertical group covering the first column (AB=00), and a green horizontal group covering the bottom row (C=1).

# 式の変形とカルノー図の操作の関係

$$Z = A \cdot B \cdot \bar{C} + A \cdot B \cdot C + \bar{A} \cdot B \cdot C$$

$$= A \cdot B \cdot \bar{C} + A \cdot B \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot C$$

$$= A \cdot B \cdot (\bar{C} + C) + (\bar{A} + A) \cdot B \cdot C$$

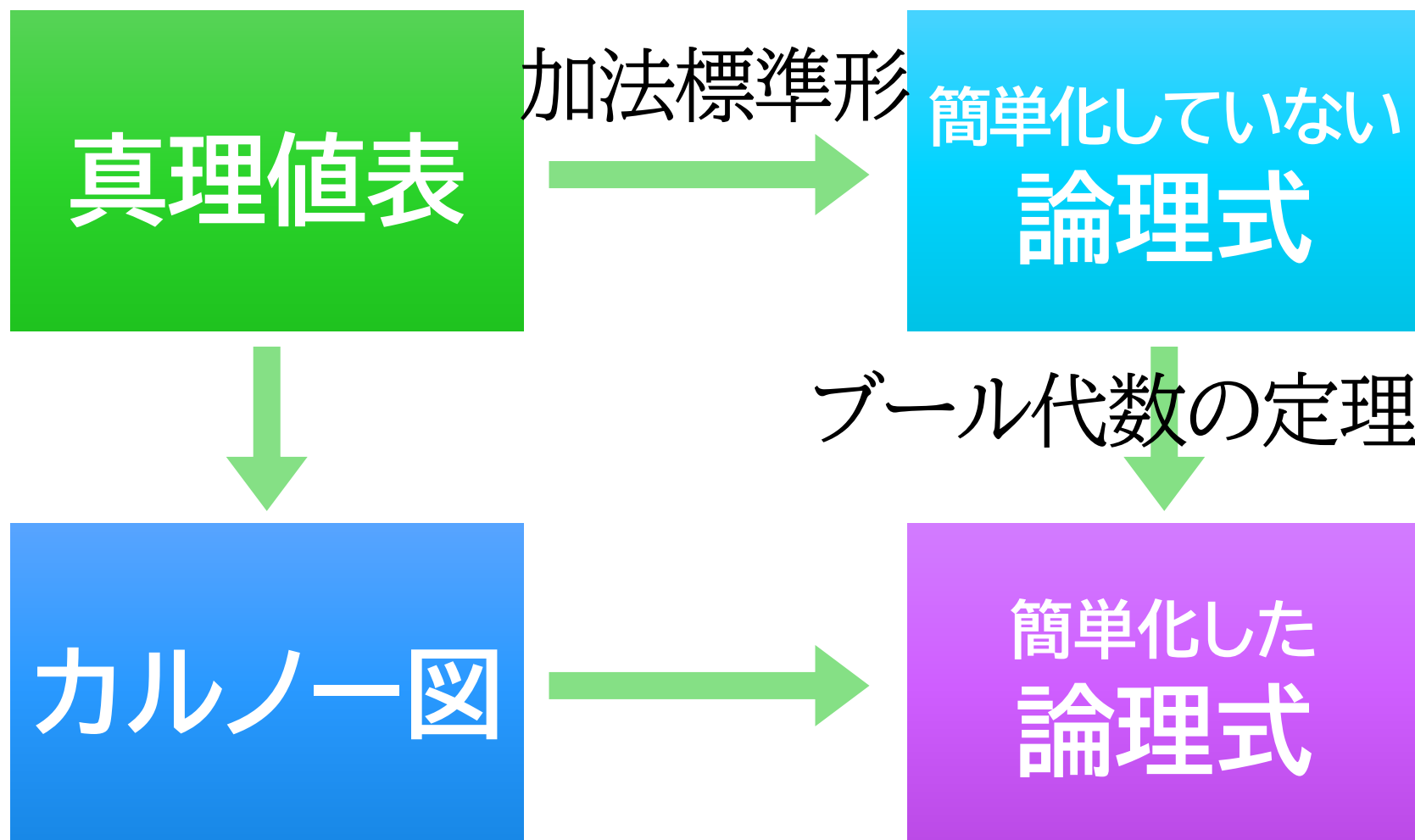
$$= A \cdot B + C \cdot B$$

Z		AB			
		00	01	11	10
C	0	0	0	1	0
	1	0	1	1	0

分配則による統合が簡単にできる。

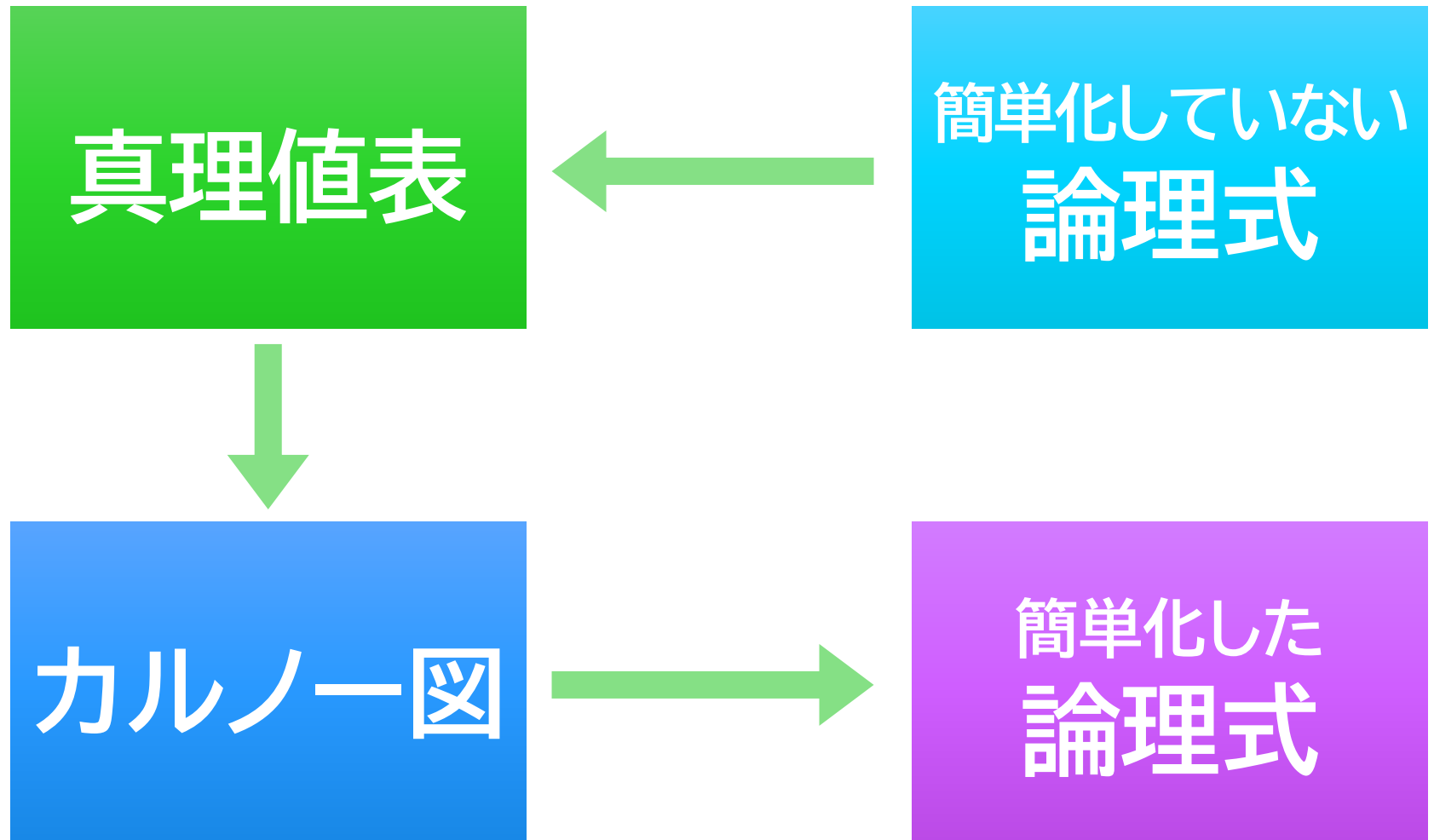
共通の論理積項が簡単に見つかる。

# 真理値表から簡単化した論理式をつくる



# カルノー図による論理式の簡単化

---





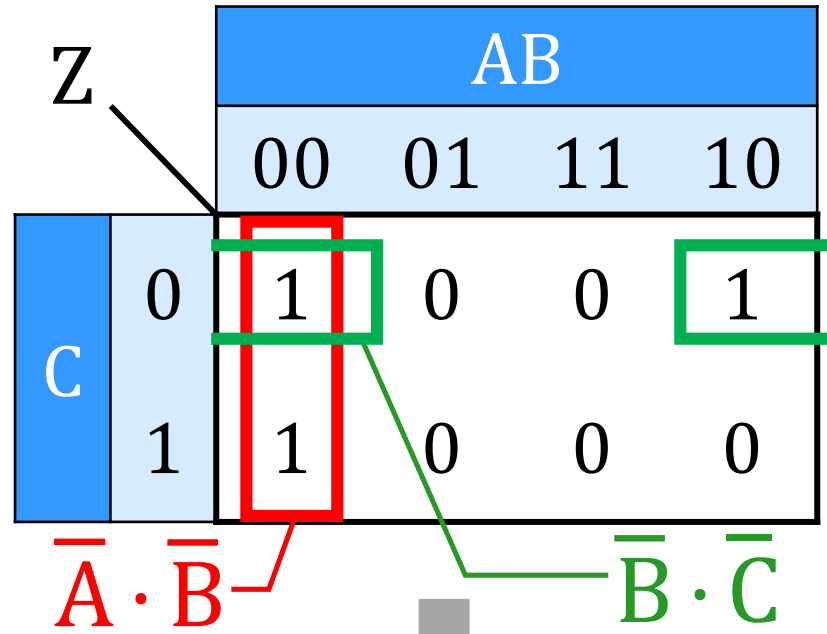
# カルノー図による論理式の簡単化

重要

$$Z = \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C}$$

A,B,C = 0 0 0    0 0 1    1 1 0    0 0

A	B	C	Z
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0



$$Z = \bar{A} \cdot \bar{B} + \bar{B} \cdot \bar{C}$$

# 課題 2

---

❖ 教科書 3章末の演習問題

1,2,3,5,6

❖ これまでの講義についての感想

どのような内容(感想・要望)でも良い。

提出日時 1月6日(月) 講義開始前

中間評価試験 ~~1月6日(月)~~

1月9日(木)

# レポートの作成について

---

- ❖ 岡山理大学専用のレポート用紙に書く。
- ❖ ホッチキスまたは糊で綴じる。
- ❖ 学生番号、氏名、講義名、提出日を書く。
- ❖ 途中の計算過程を書く。
- ❖ 解答した後、教科書の演習問題解答を見て、赤ペンで○×をつける。
- ❖ 間違えた問題、解けなかった問題は赤ペンで計算過程と正答を書く。

# 論理回路の種類

---

## ❖ 組合せ論理回路

現在の入力値によって出力値が定まる。

全加算器、デコーダ、マルチプレクサなど

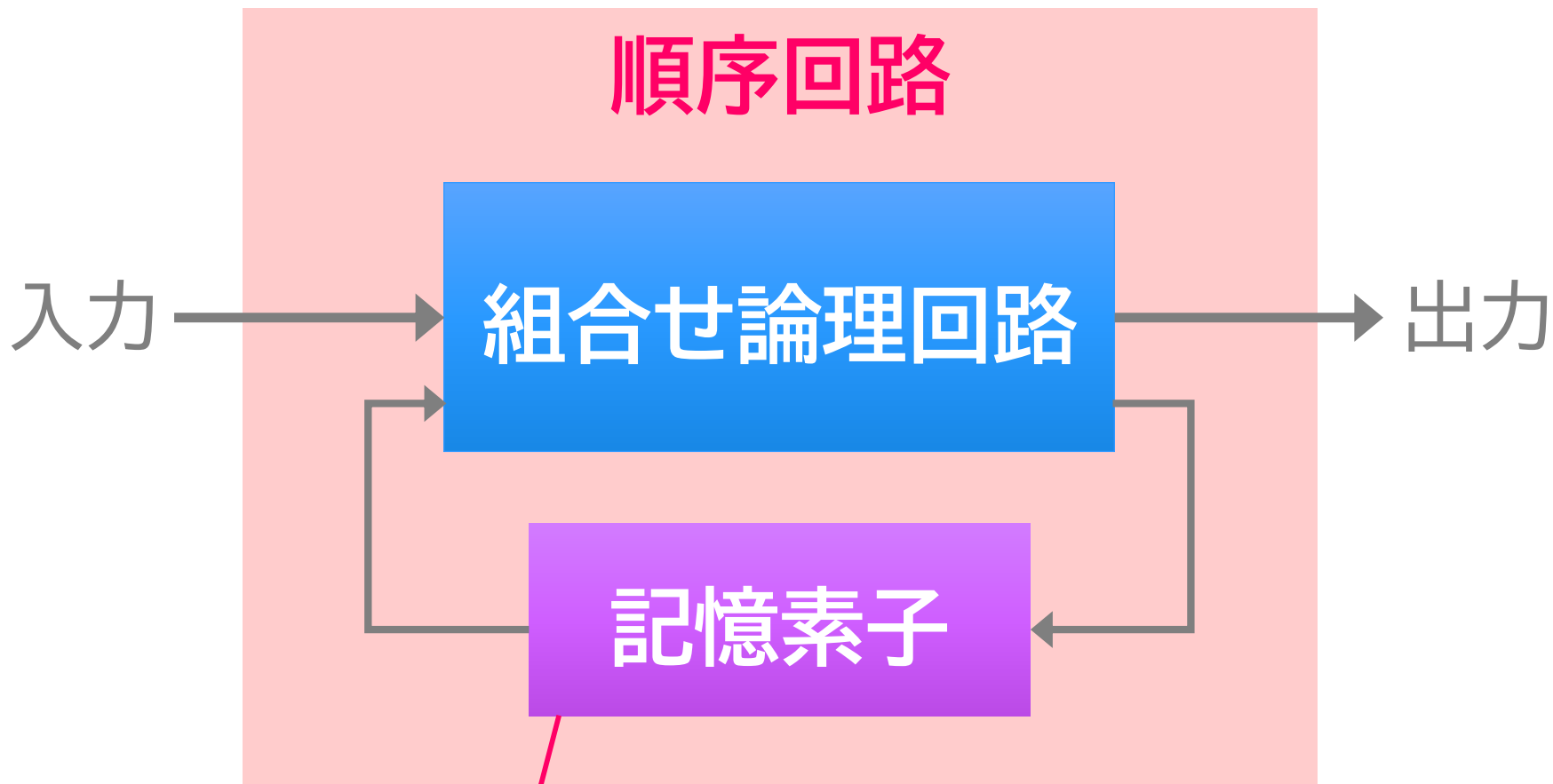
## ❖ 順序回路

現在の入力値と過去の入力値によって出力値が定まる。

レジスタ、シフトレジスタ、カウンタなど

# 順序回路

---



フリップフロップ回路

1 bitの値を記憶する回路

# RSフリップフロップ

重要

## 入力信号

R リセット S セット

## 出力信号

$Q(n)$  時刻  $n$  のときの出力

$R=0, S=1$  のとき、1を記憶。

$R=1, S=0$  のとき、0を記憶。

$R=S=0$  のとき、記憶は変化しない。

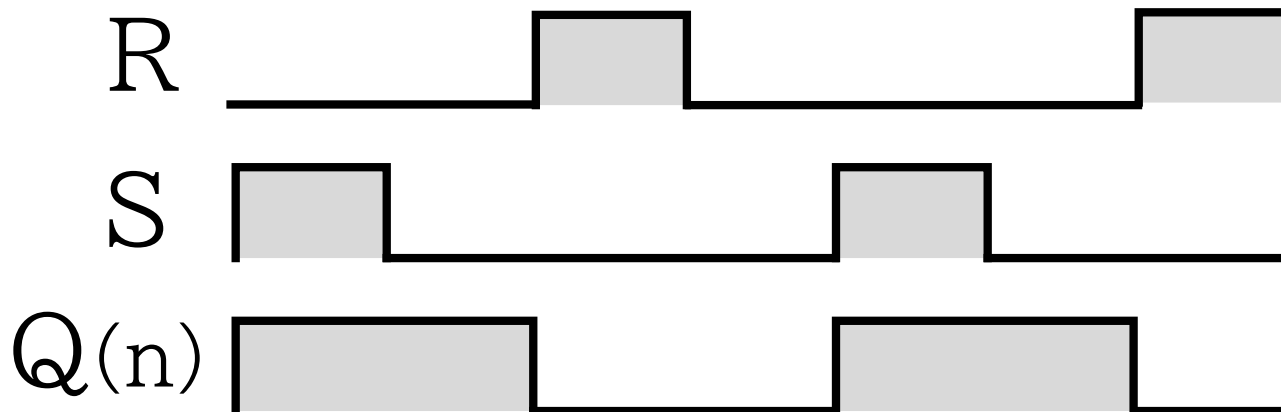
## 特性表

R	S	$Q(n)$
0	0	$Q(n-1)$
0	1	1
1	0	0
1	1	不定

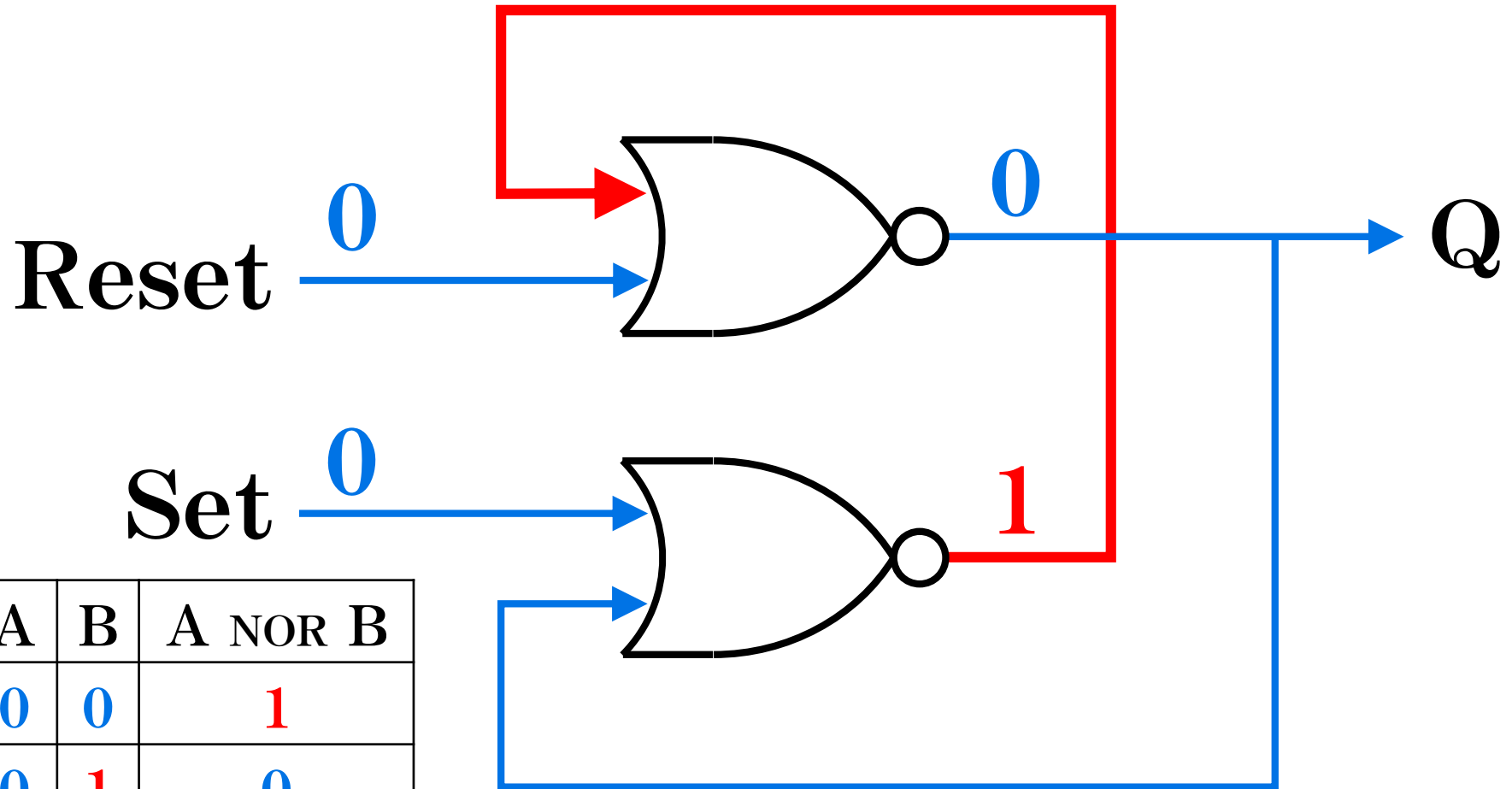
# RSフリップフロップの動作

時刻 n	1	2	3	4	5	6	7
入力信号 R	0	0	1	0	0	0	1
入力信号 S	1	0	0	0	1	0	0
出力信号 Q(n)	1	1	0	0	1	1	0

## タイミングチャート



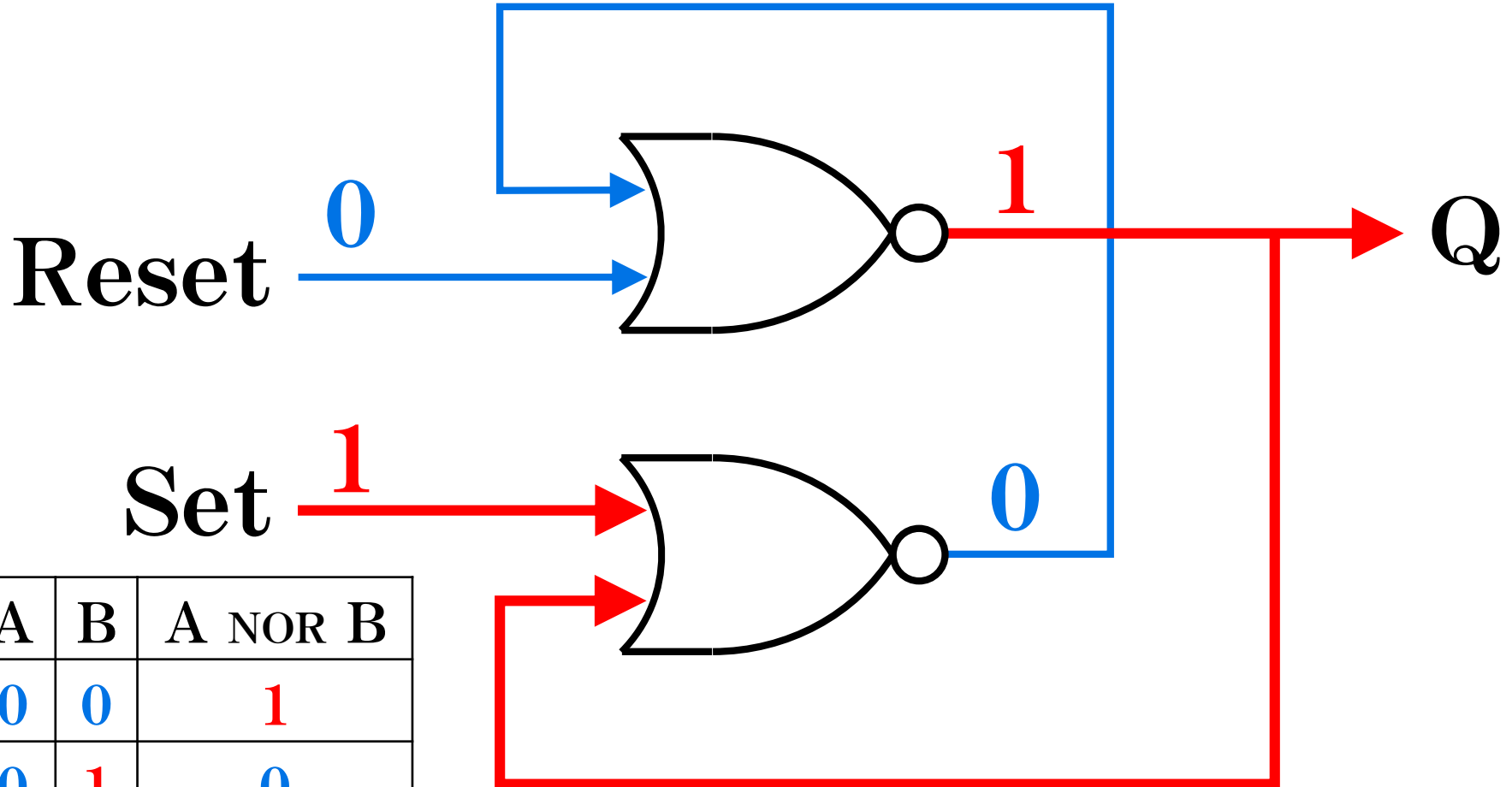
# フリップフロップの仕組み



A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

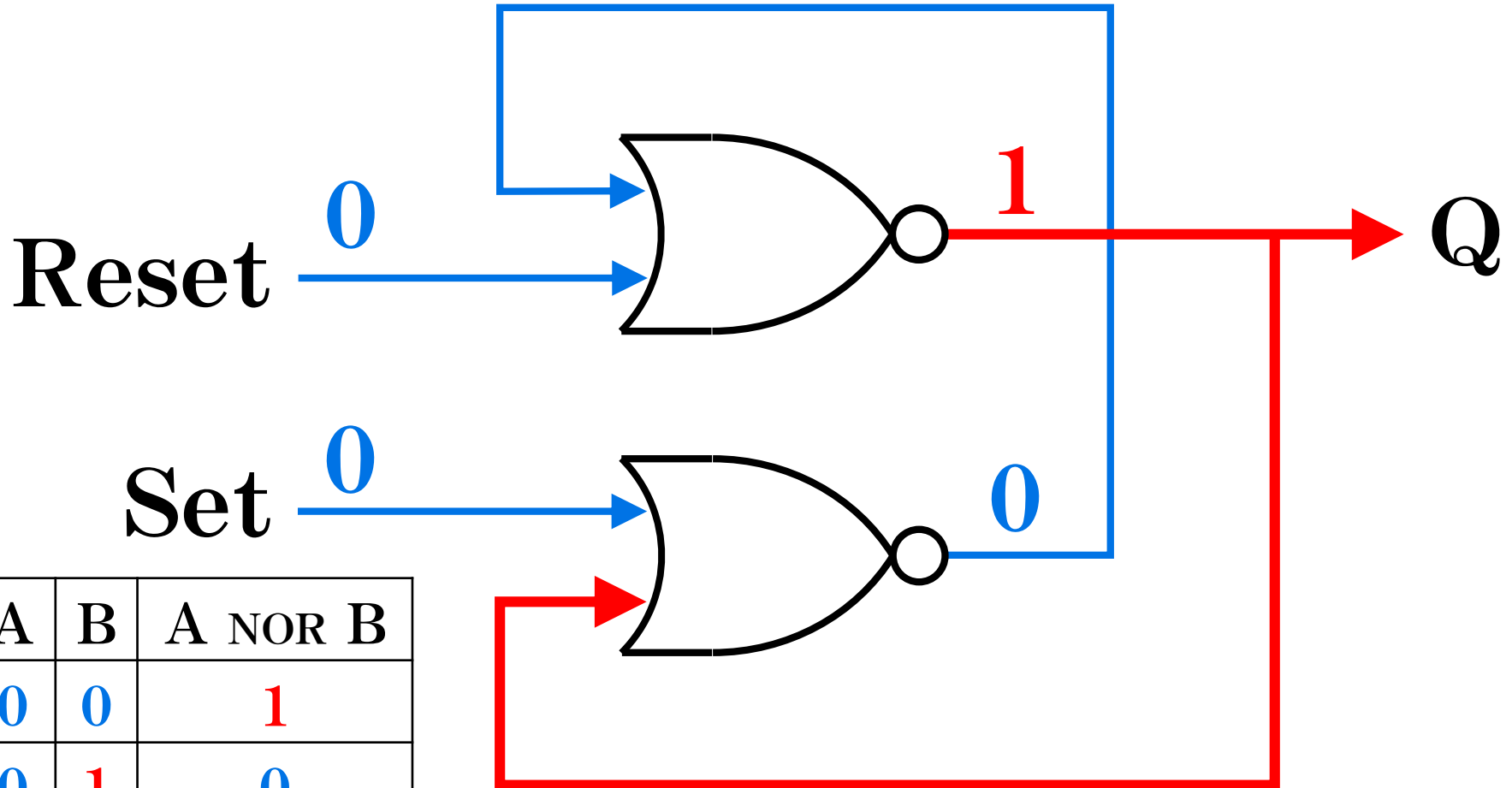


# フリップフロップの仕組み



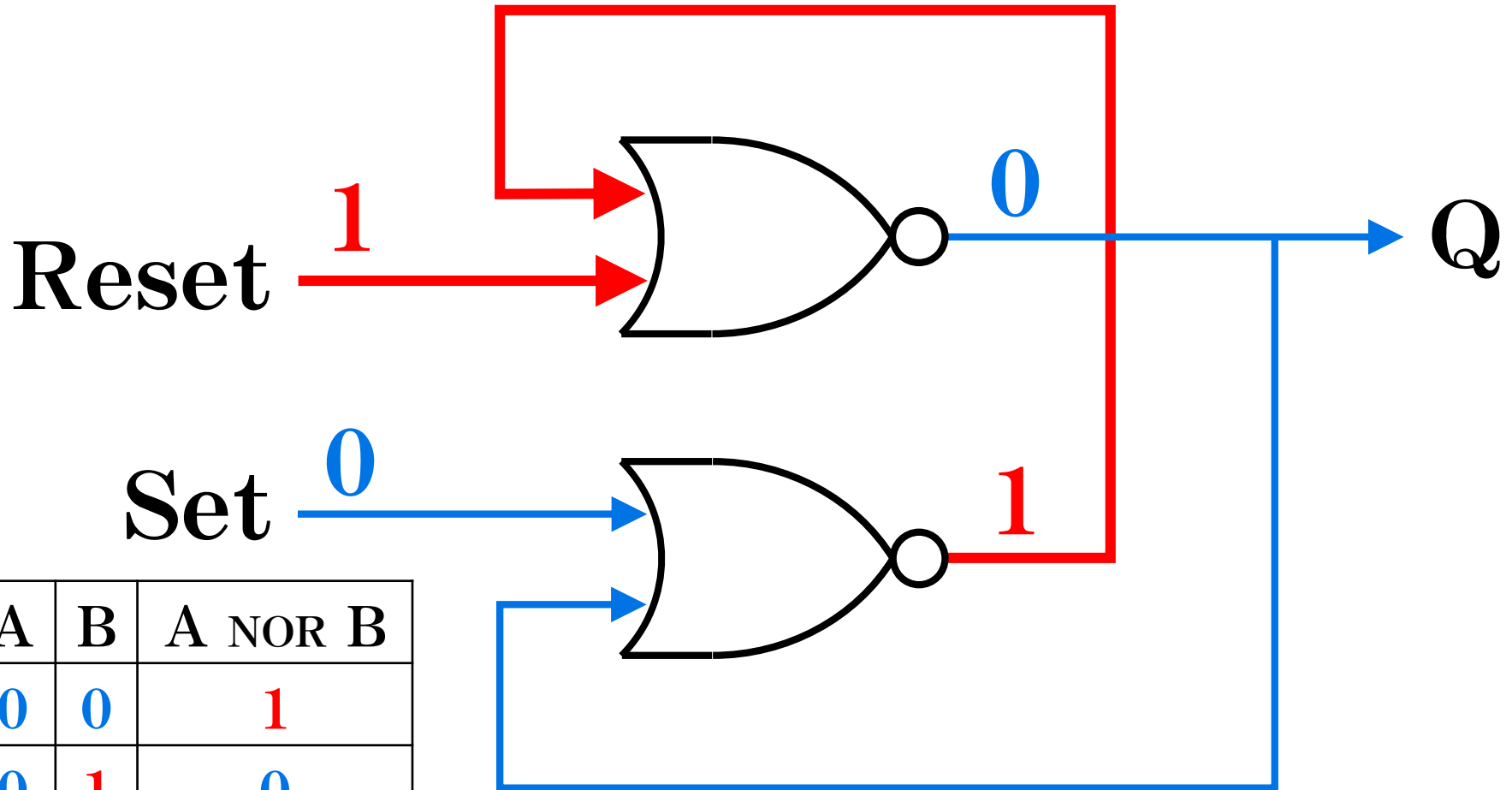
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

# フリップフロップの仕組み



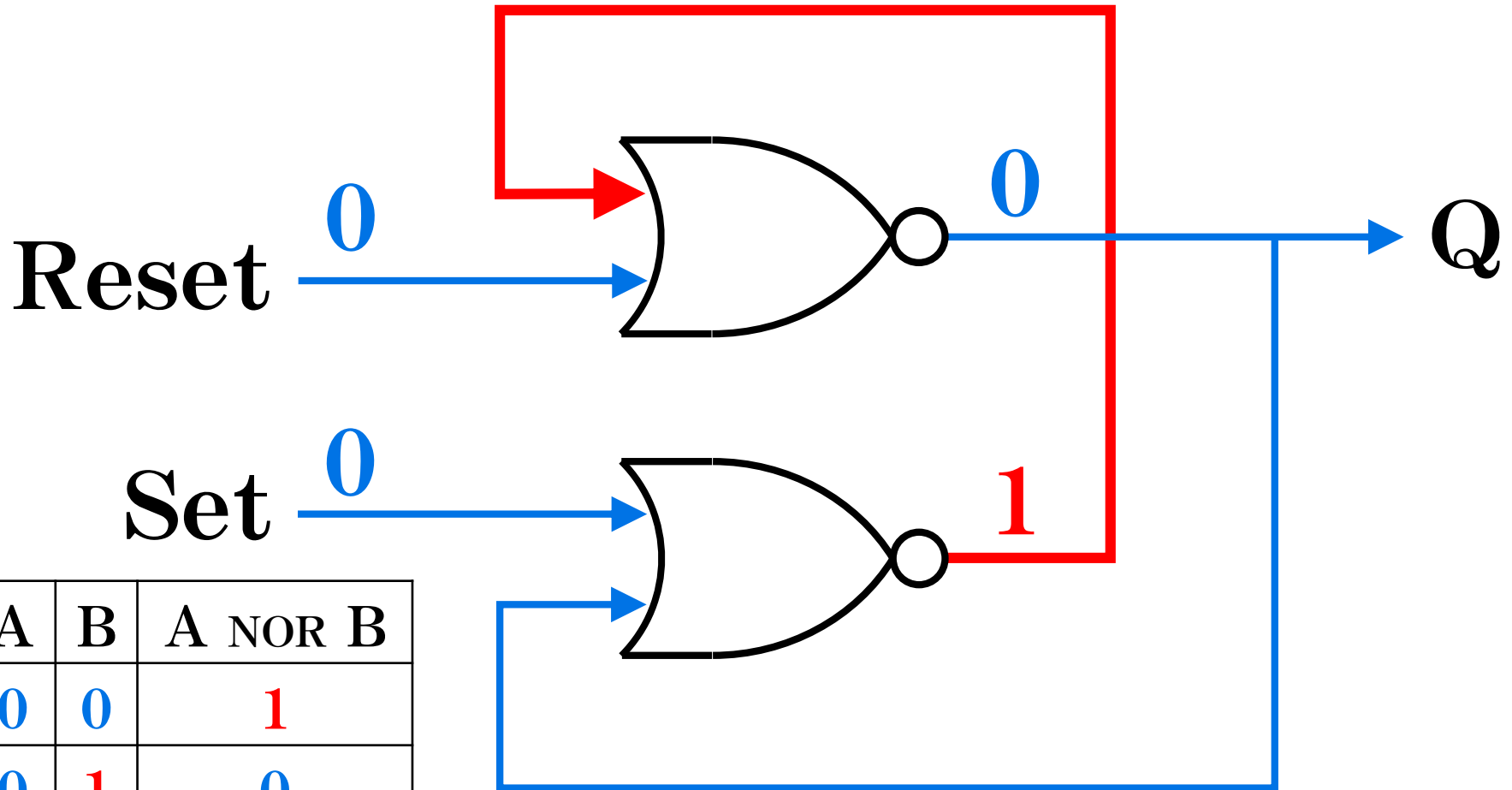
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

# フリップフロップの仕組み



A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

# フリップフロップの仕組み



A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

# JKフリップフロップ

重要

## 入力信号

J セット    K リセット

J=1, K=0のとき、1を記憶。  
J=0, K=1のとき、0を記憶。  
J=K=0のとき、記憶は変化しない。  
J=K=1のとき、記憶している値を反転する。

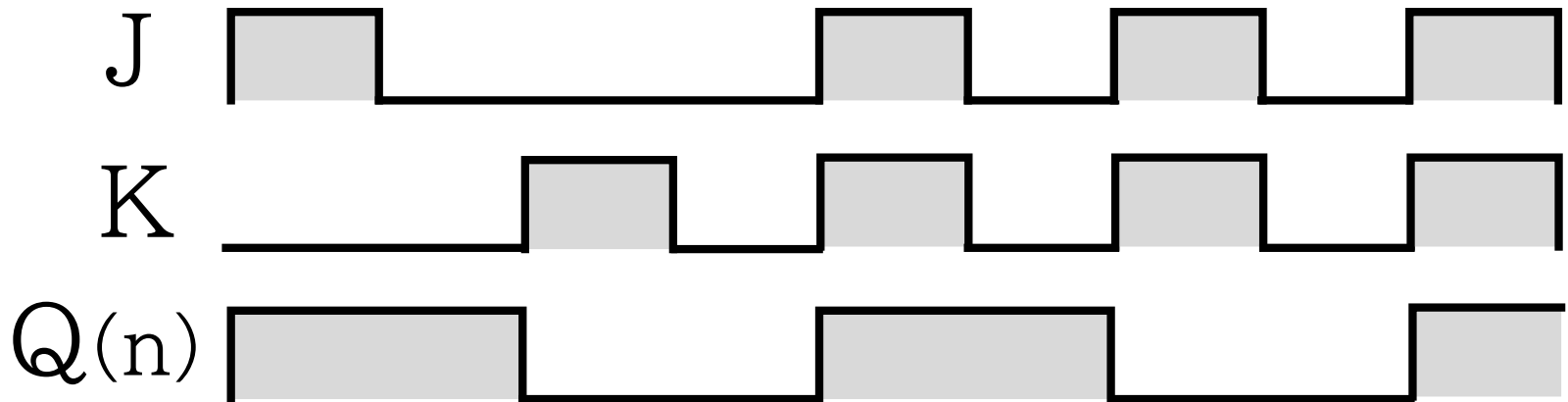
## 特性表

J	K	Q(n)
0	0	Q(n-1)
0	1	0
1	0	1
1	1	$\overline{Q(n-1)}$

# JKフリップフロップの動作

時刻 n	1	2	3	4	5	6	7	8	9
入力信号 J	1	0	0	0	1	0	1	0	1
入力信号 K	0	0	1	0	1	0	1	0	1
出力信号 Q(n)	1	1	0	0	1	1	0	0	1

## タイミングチャート



# 同期式フリップフロップ

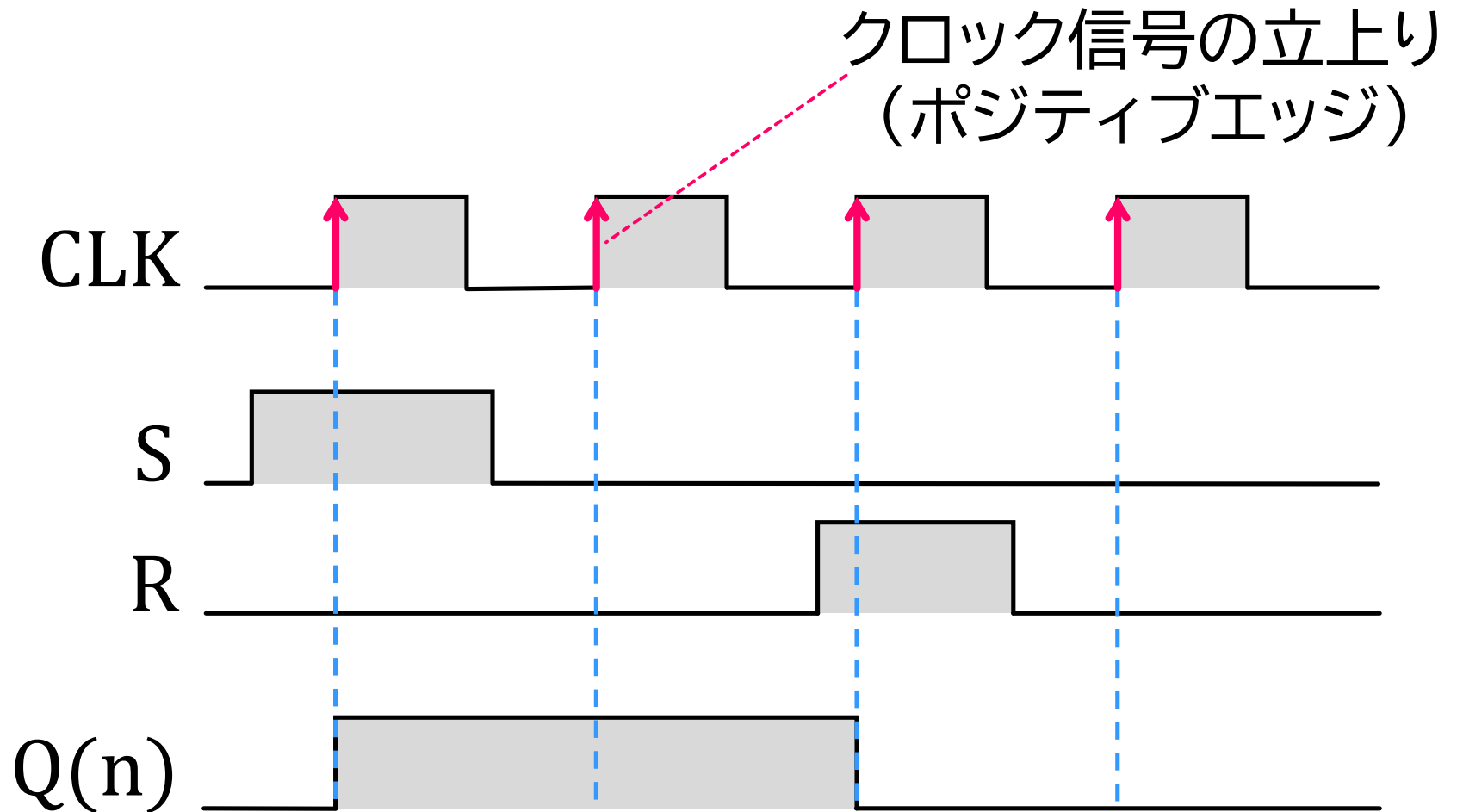
## ❖ クロック信号

複数の電子回路の間で信号の送受信のタイミングを合わせる(同期をとる)ために用いる一定周期の信号

## ❖ 同期式フリップフロップ

クロック信号の立上り(または立下り)のときの入力値で出力が決定するフリップフロップ

# 同期式フリップフロップのタイミングチャート





# その他のフリップフロップ

## ♣ Dフリップフロップ

入力信号 D を1クロック周期保持する。

D	Q(n)
0	0
1	1

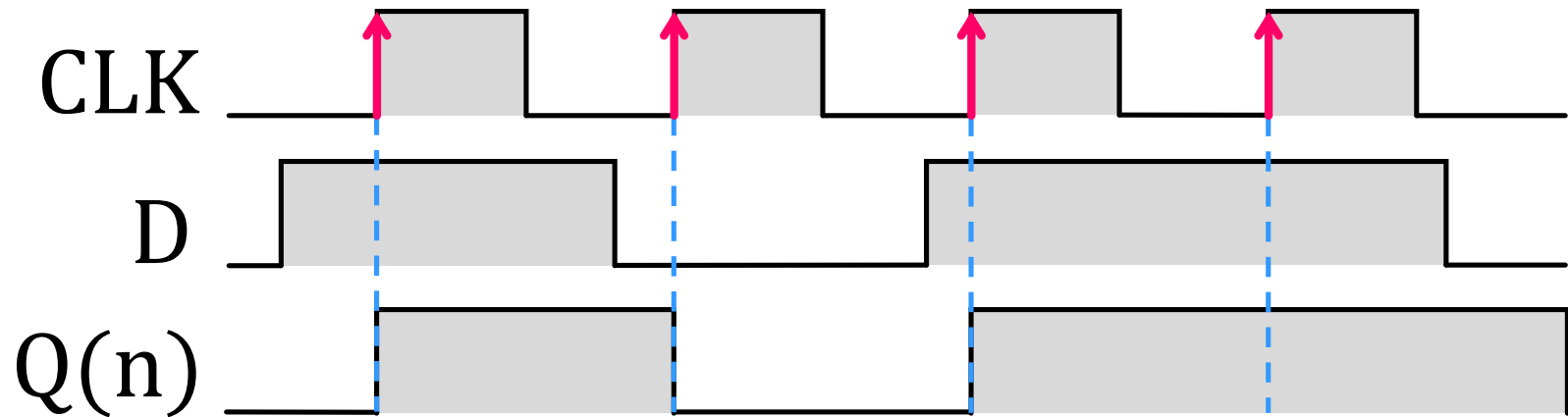
## ♣ Tフリップフロップ

トリガ信号 T が入力されるたびに出力を反転する。

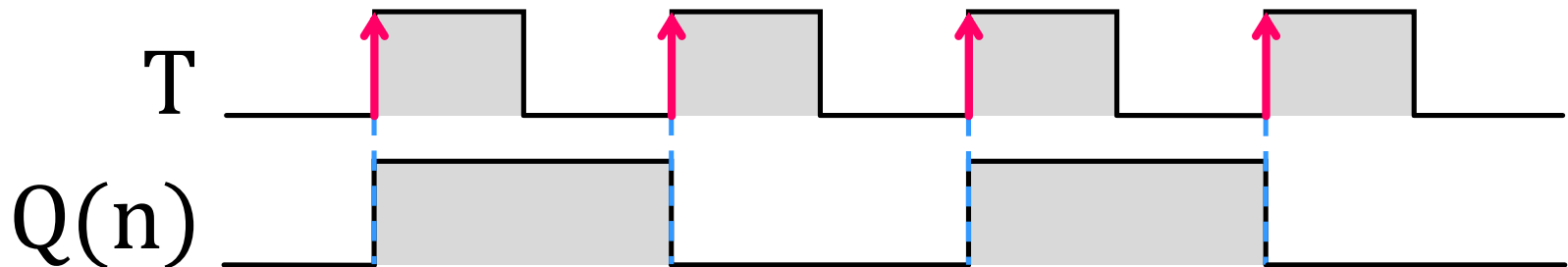
T	Q(n)
0	Q(n-1)
1	$\overline{Q(n-1)}$

# D-FF/T-FFのタイミングチャート

## Dフリップフロップ



## Tフリップフロップ



# フリップフロップの応用

---

## CPUの中の記憶装置

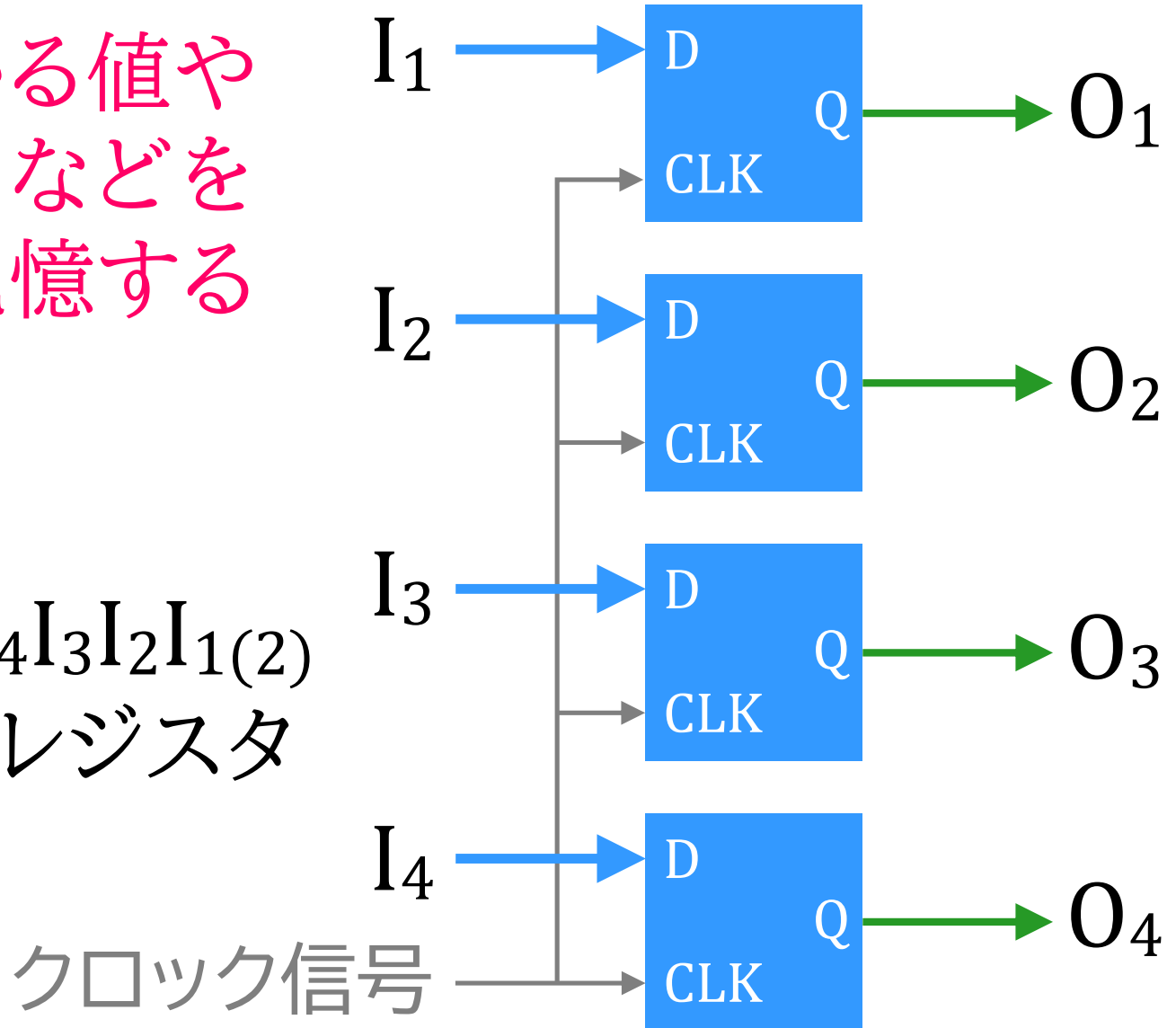
- ♣ レジスタ
- ♣ シフトレジスタ
- ♣ カウンタ
- ♣ 基本記憶素子(キャッシュメモリ)

重要

# レジスタ

演算に用いる値や命令コードなどを一時的に記憶する装置

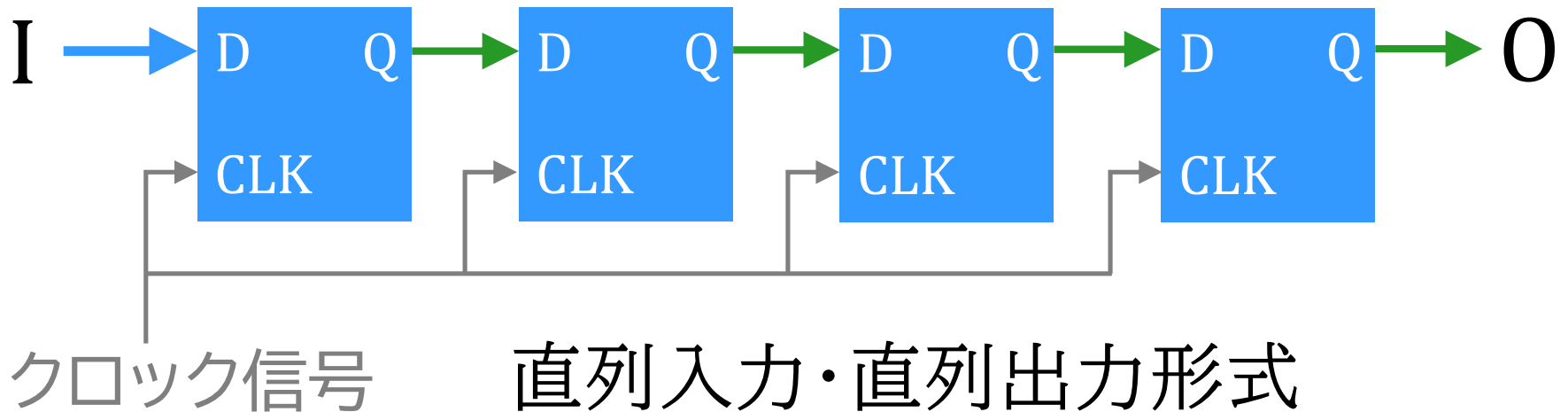
4bitの値  $I_4I_3I_2I_1(2)$  を記憶するレジスタ



# シフトレジスタ

クロック信号が入力されるたびに、数値が  
1 bitシフトするレジスタ

データを1bitずつ送信する。

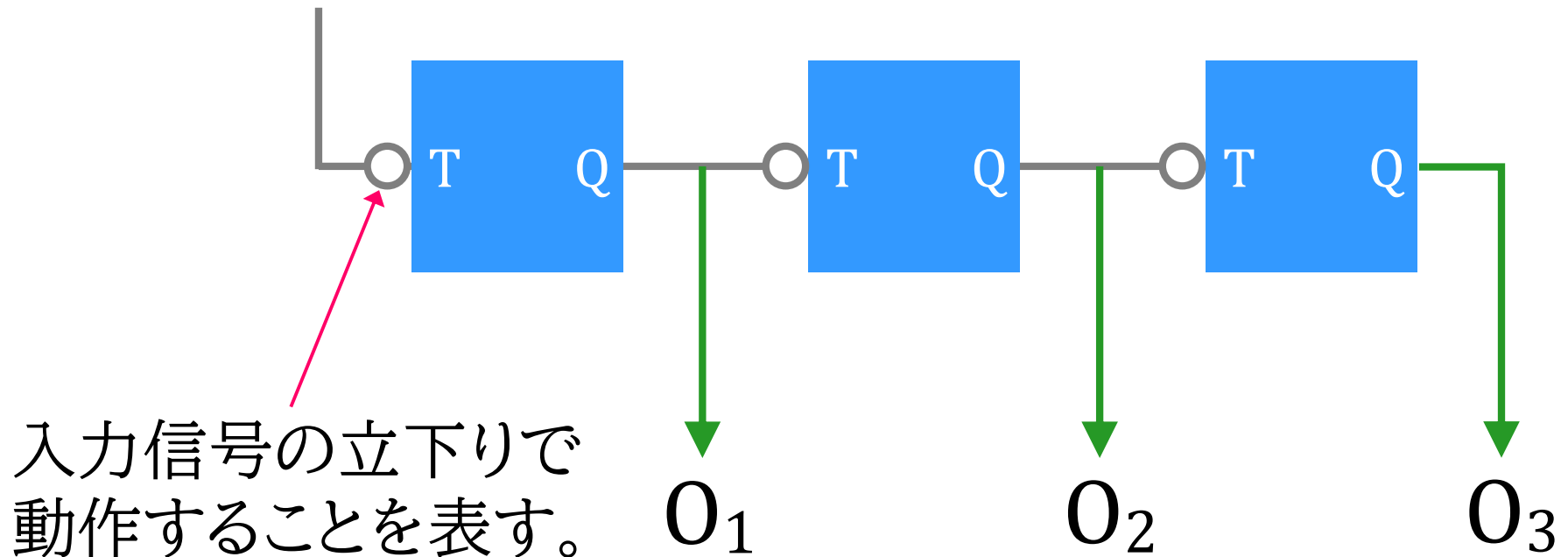


# カウンタ

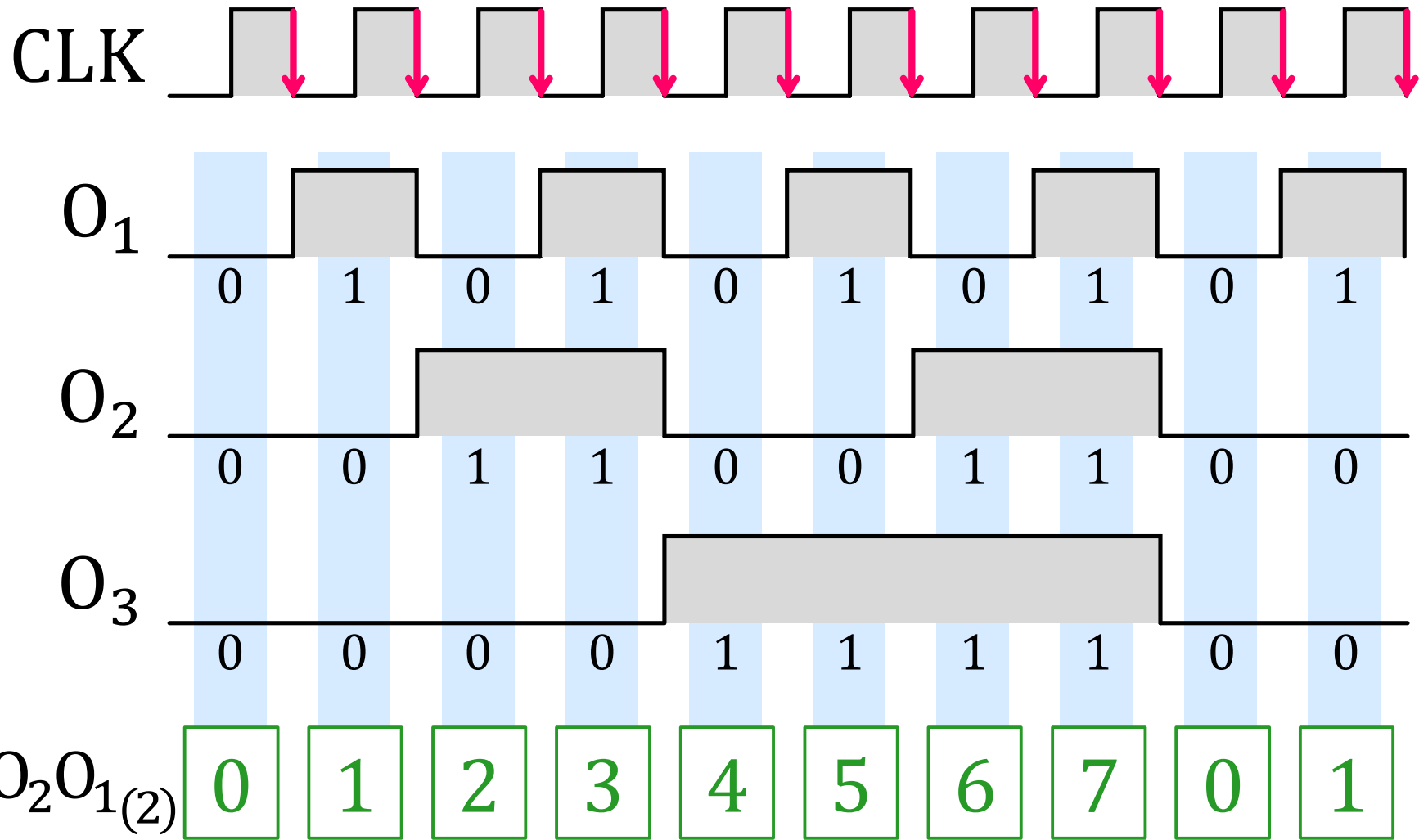
重要

入力信号のパルス数をかぞえる機能を持つレジスタ

クロック信号

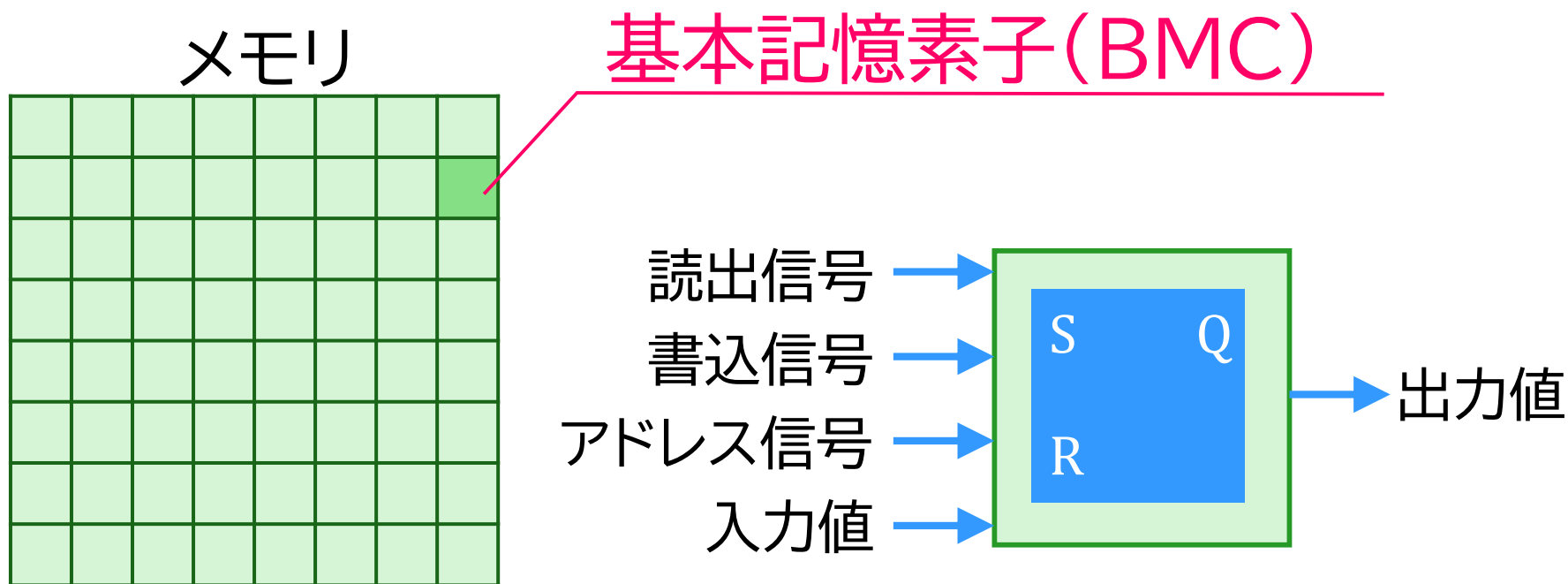


# カウンタのタイミングチャート



# 基本記憶素子

アレイ状のメモリを構成する1bitの記憶素子



CPUのキャッシュメモリのBMCは、フリップフロップを用いている。