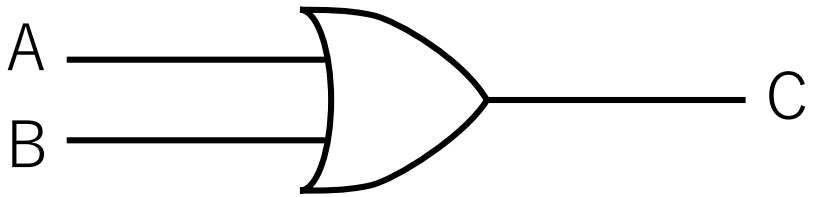
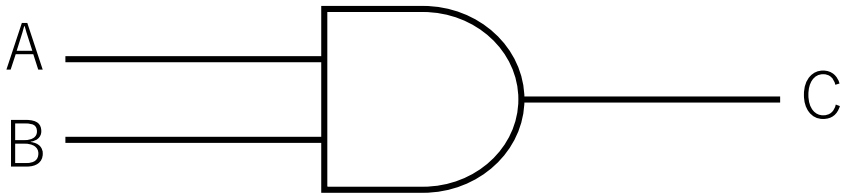
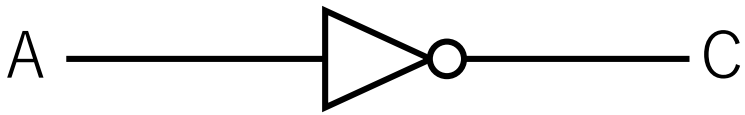
論理回路





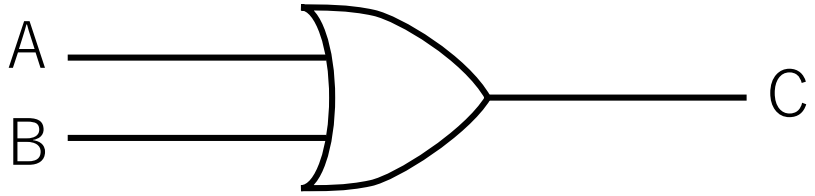


平野拓一

論理ゲート

- 2進法の論理演算(AND, OR, NOTなど)を電子回路で実現。
- 高電圧(H)で1を、低電圧(L)で0を表現するなど。
- 2進法の論理演算を行う電子回路をゲートと呼ぶ。
- 2進法に変換すると論理演算を使って算術演算も可能（ブール代数）。

論理和(OR)



$$C = A + B$$

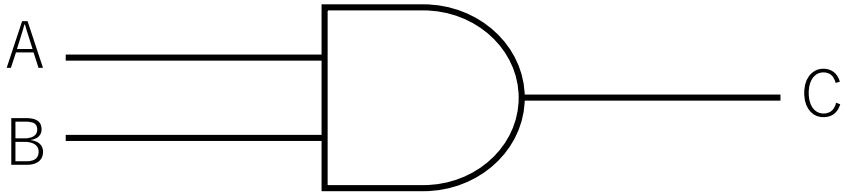
機能表

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

どちらかの入力が1ならば出力は1

C=A+B

論理積(AND)

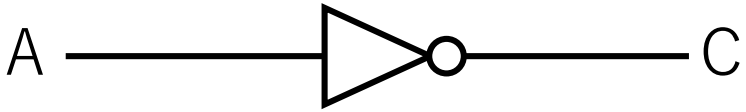


$$C = A \cdot B = AB$$

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

両方の入力が1のときのみ出力は1

否定(NOT)

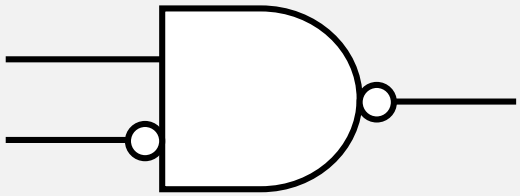


$$C = \overline{A}$$

A	C
0	1
1	0

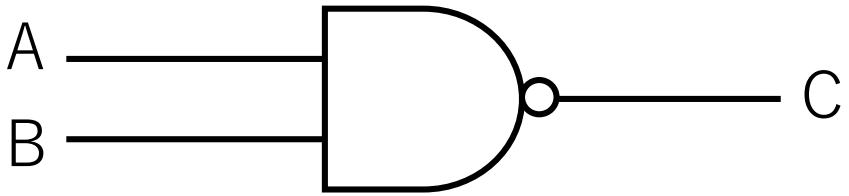
入力を反転（否定）

入力や出力でNOTをとるときはこのような記法も使う。



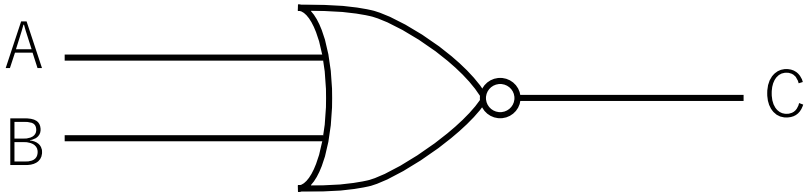
NANDとNOR

NAND



$$C = \overline{AB}$$

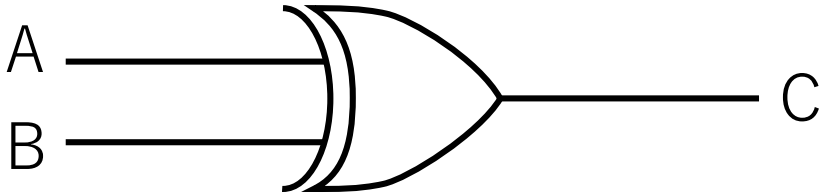
NOR



$$C = \overline{A + B}$$

回路実現の容易さから
現実的によく使われる。

排他的論理和(XOR)



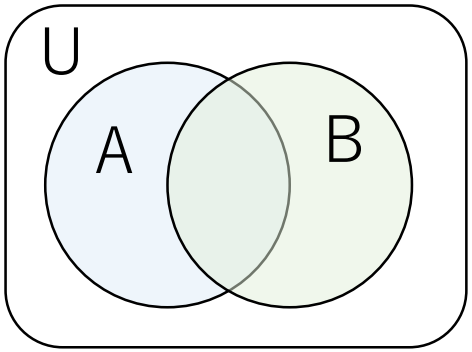
Exclusive **OR**

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

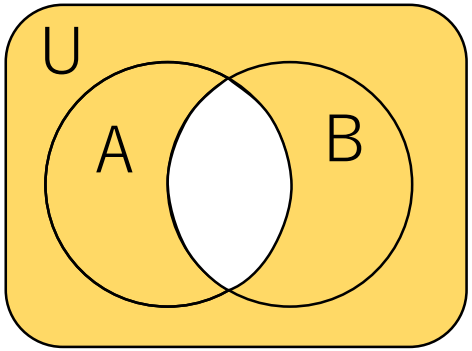
$$C = \overline{A}B + A\overline{B}$$

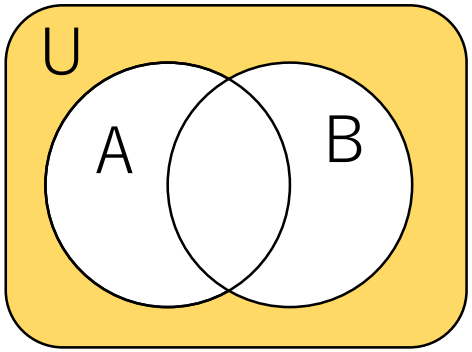
片方の入力が1のときのみ出力は1
２進法としてA+Bを行い、一番下の位を読めばよい。
（→つまり、繰り上がりを排除(Exclude)している）

ド･モルガンの法則



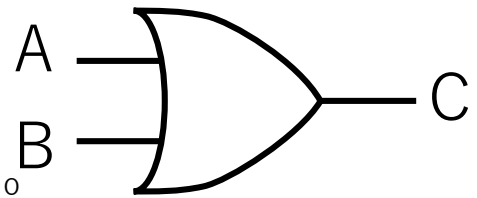
$$\begin{cases} \overline{AB} = \overline{A} + \overline{B} \\ \overline{A + B} = \overline{A} \cdot \overline{B} \end{cases}$$

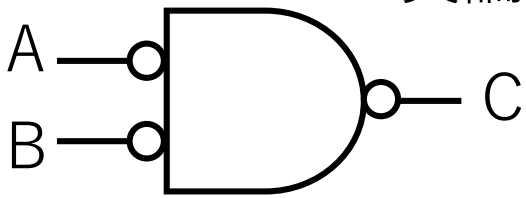




$$A + B = \overline{\overline{A} \cdot \overline{B}}$$

NANDと負論理で実現できる

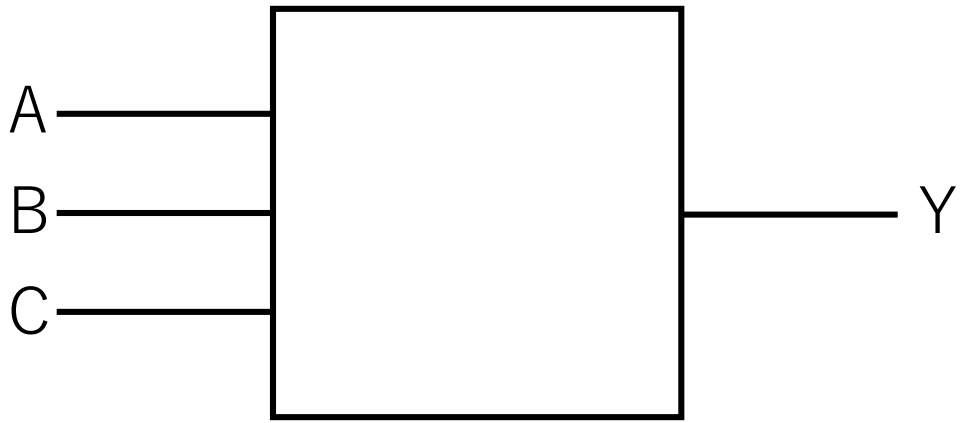




論理回路の設計と簡単化

- 論理回路：論理ゲートを組み合わせて作られた回路。
- 設計法:
  (1) 機能表で動作を記述
  (2) カルノー図を描いて数式表現を簡単化
  (3) 論理ゲートで表現
  (4) 必要に応じて、NAND回路のみで表現
  (5) 論理ゲートで組む。

例）3入力，1出力



機能表

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

カルノー図による簡単化

機能表

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

カルノー図

この並びに注意
隣同士の列は1カ所しか変化しない
（ハミング距離=1）

Y

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

Aによらない
⇒BC

B,Cによらない
⇒A

隣り合って1がかたまっている場所を囲む。
上下左右は2つ単位でグループ化できる。

$$Y = A\bar{B}\bar{C} + AB\bar{C} + \bar{A}BC + A\bar{B}C + ABC$$

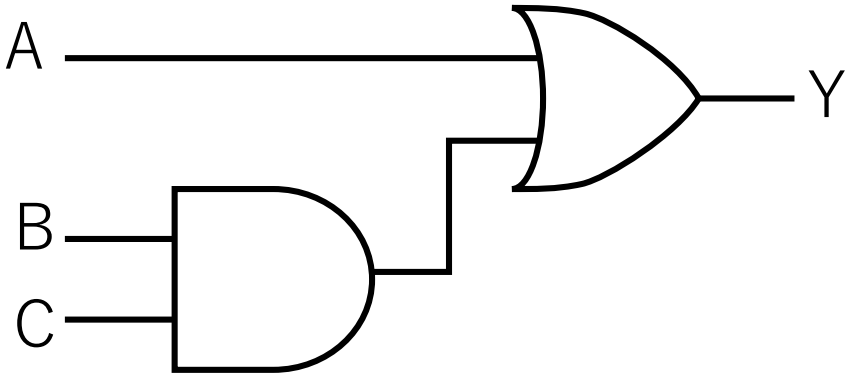
$$Y = A + BC$$

簡単化された！

コンピュータを用いた自動処理には**クワイン・マクラスキー法**などが使われる。

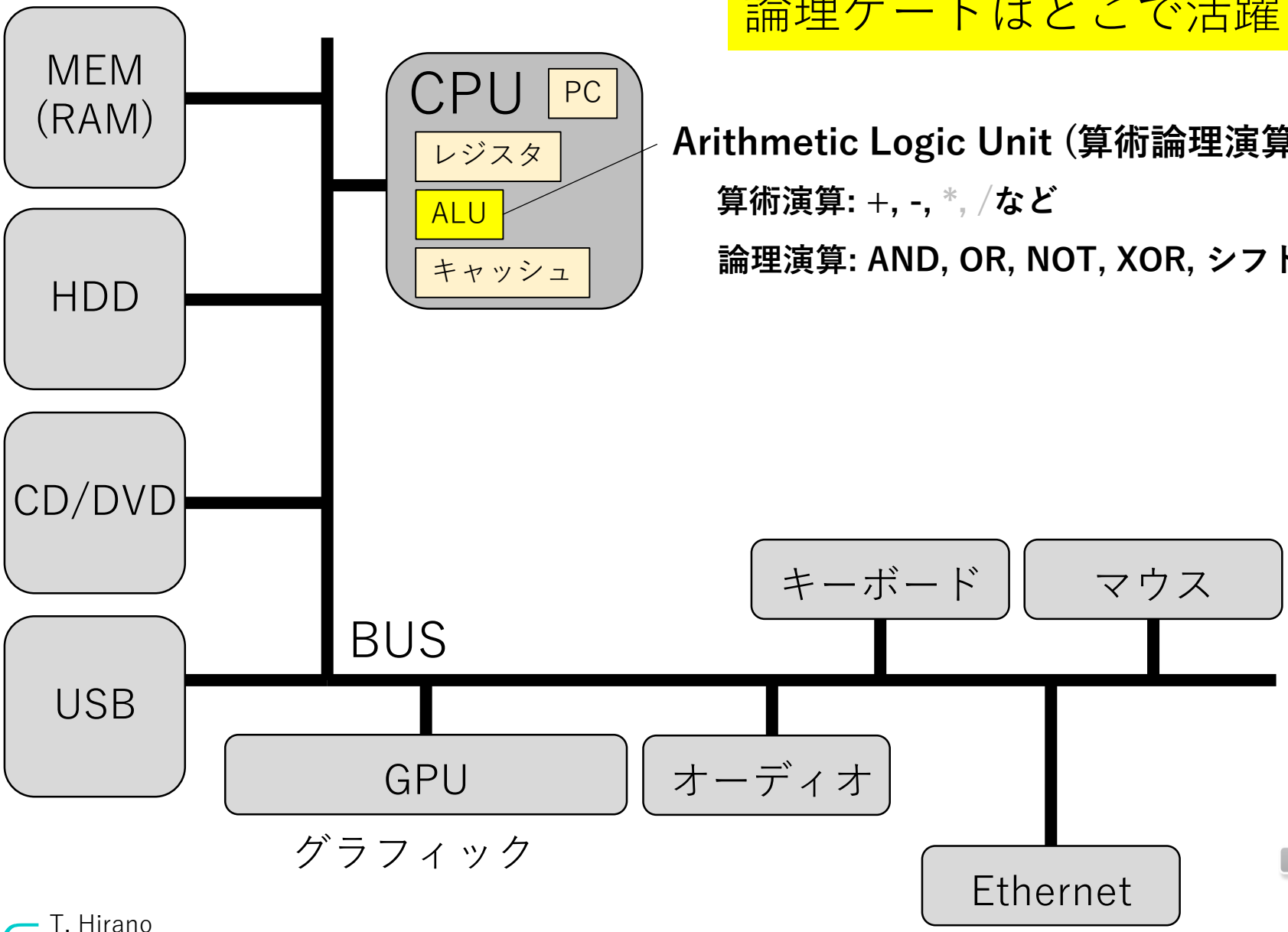
論理ゲートで実現

$$Y = A + BC$$



PCの構成

論理ゲートはどこで活躍？



Arithmetic Logic Unit (算術論理演算装置)

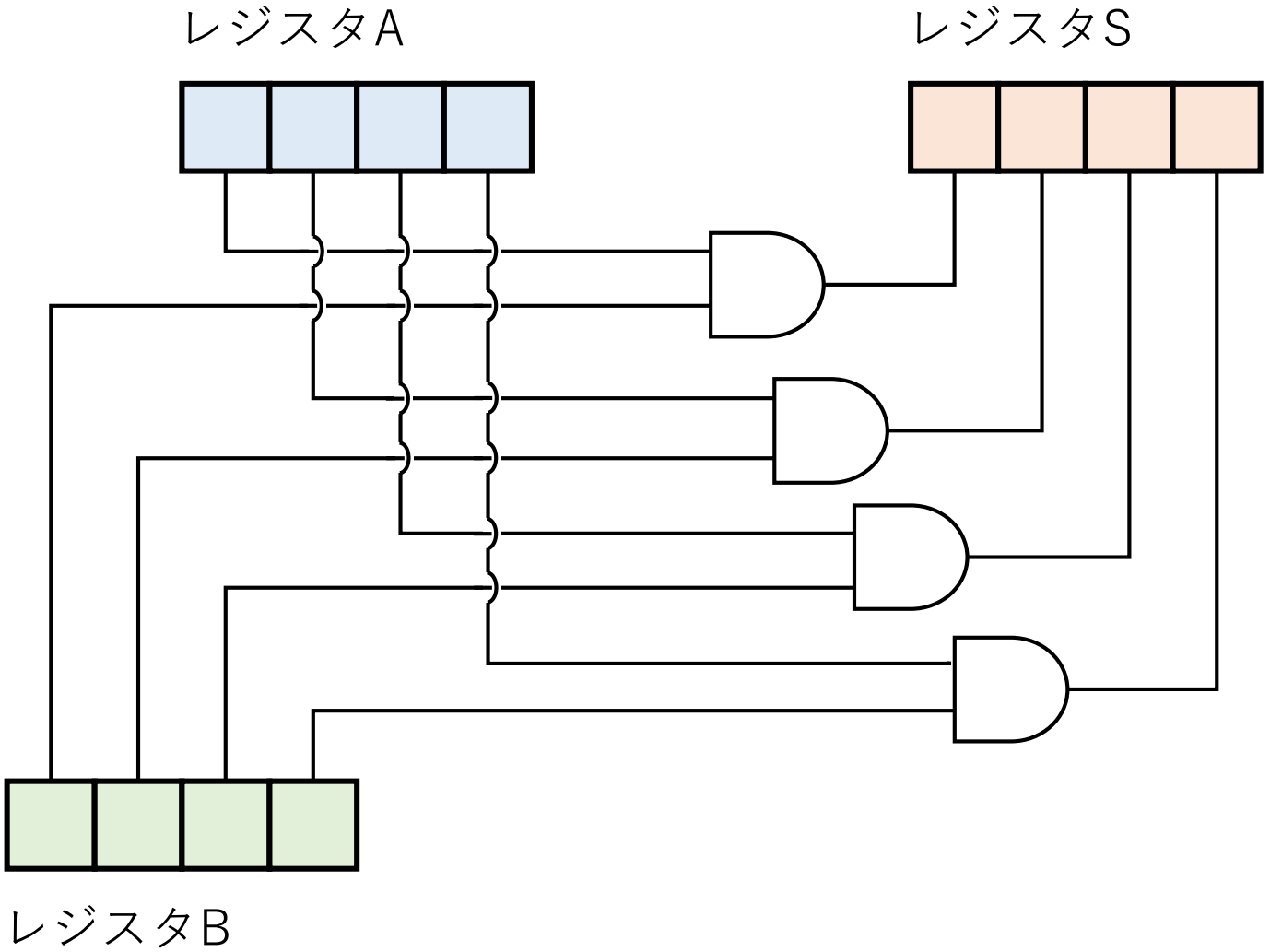
算術演算: +, -, *, /など

論理演算: AND, OR, NOT, XOR, シフトなど

論理演算（AND）

複数ビット

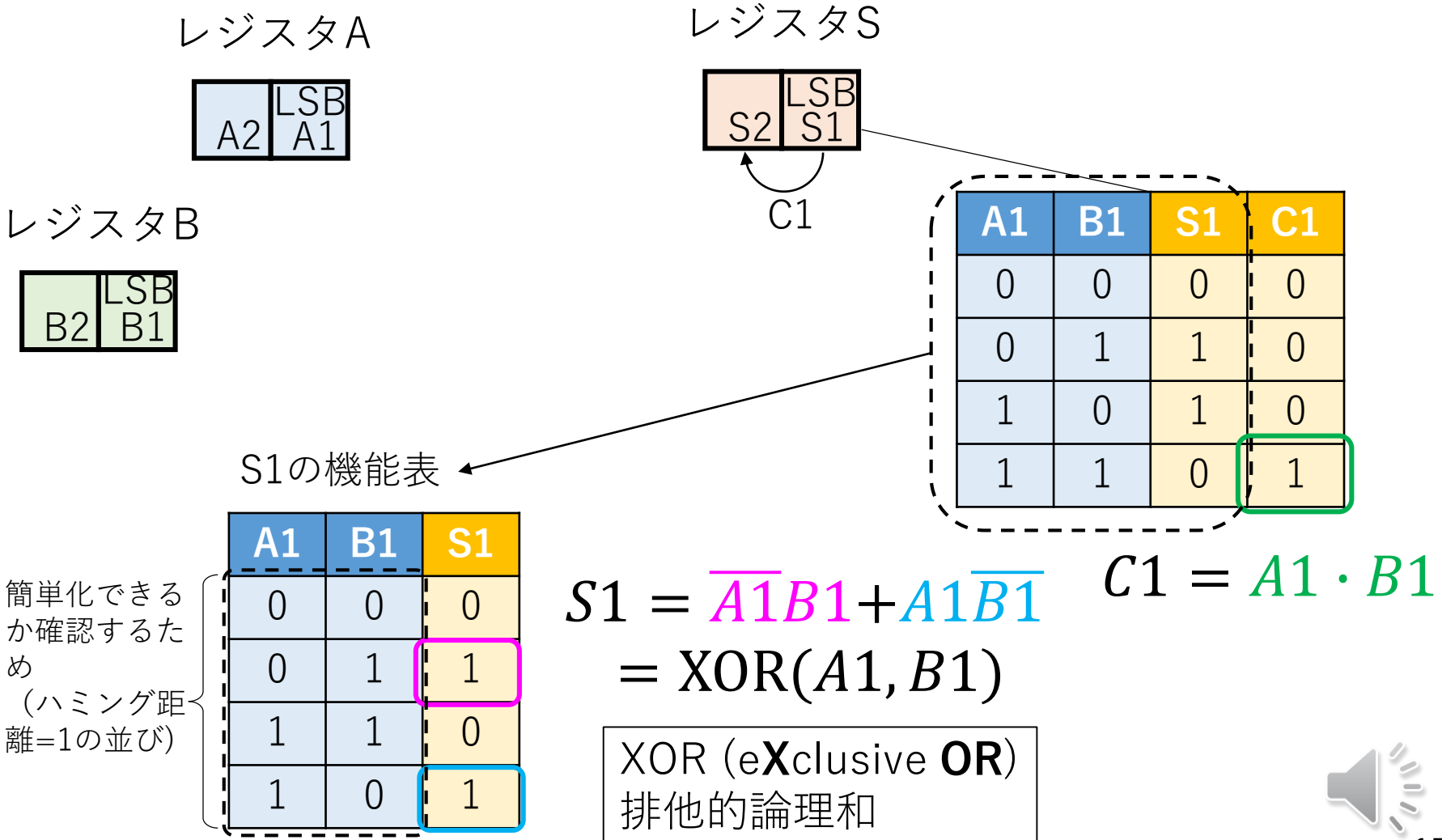
$$S = A \text{ and } B$$



算術演算 (+) : 加算器

1桁目(LSB)の計算

$$S = A + B$$

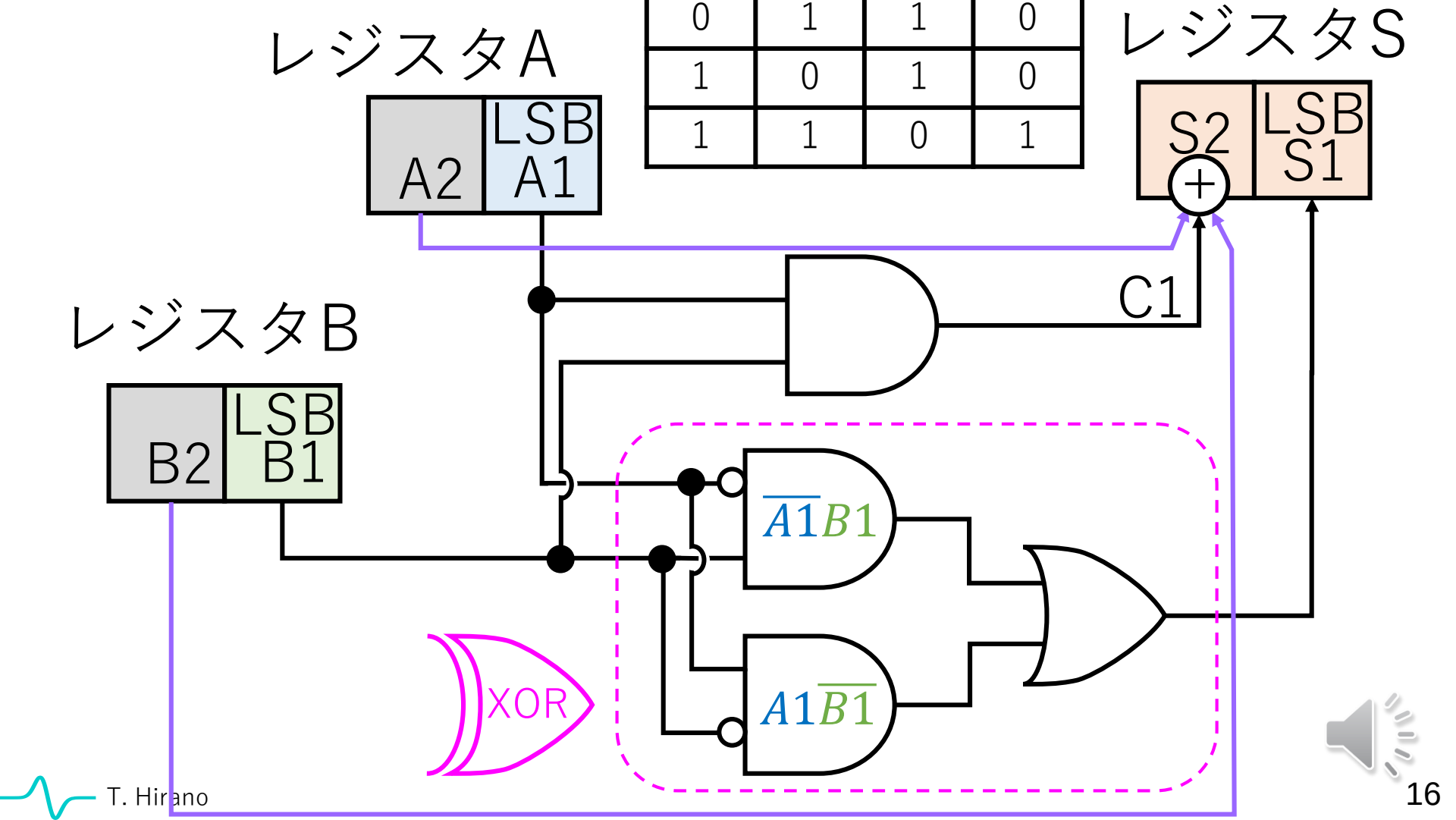


算術演算 (+) : 加算器

2 桁　　1桁　　1桁

$$S = A + B$$

A1	B1	S1	C1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

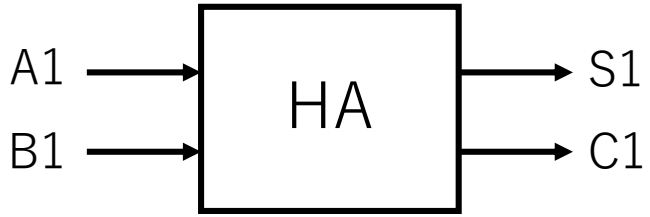


半加算器と全加算器

ある位の和と繰り上がり(Carry)に着目する。

半加算器(Half Adder)

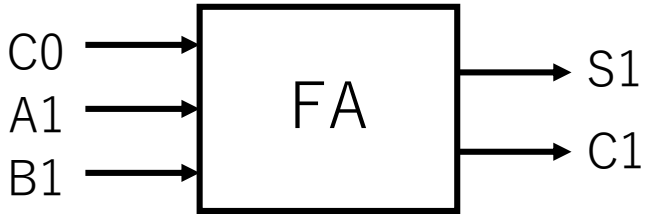
下の位からの繰り上がりを考えない。



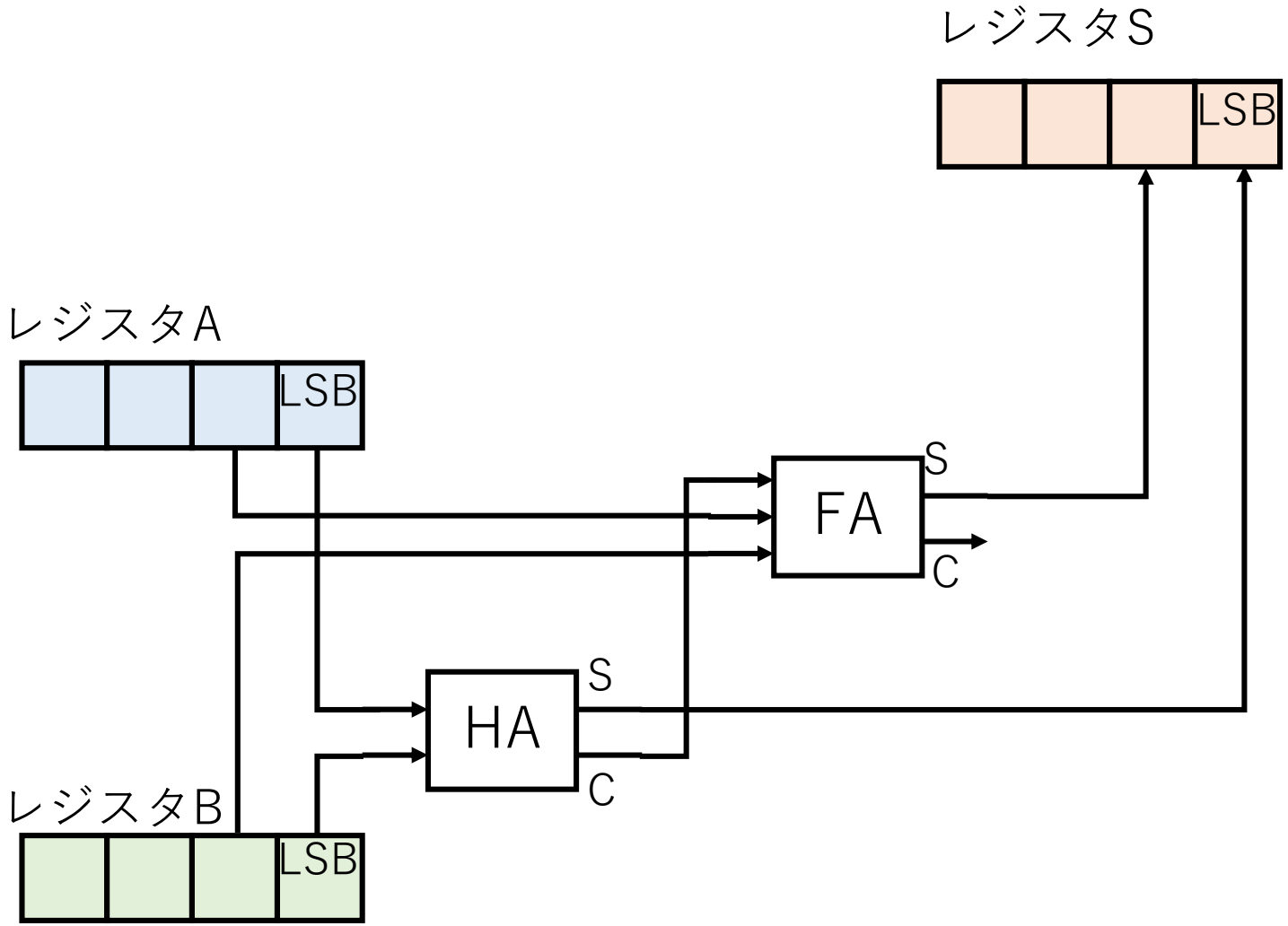
- ブラックボックスで描くとこのようになる。
- 論理ゲートでの実現法は前のスライドの通り。

全加算器(Full Adder)

下の位からの繰り上がりを考慮。



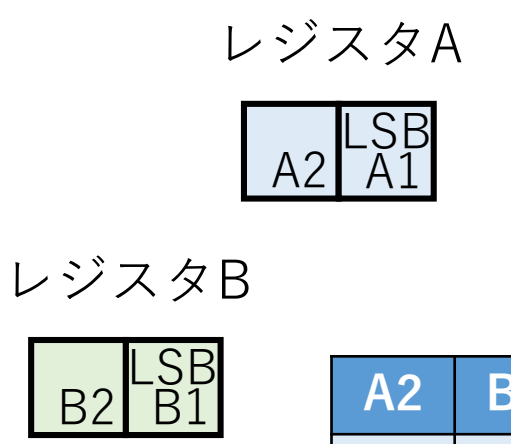
複数桁の加算

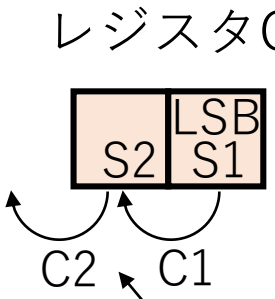


算術演算 (+): 加算器(全加算器)

$$S = A + B$$

2桁目の計算





A2	B2	C1	S2	C2
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

ハミング距離=1
に並べ替え

A2	B2	C1	S2	C2
⋮				
0	1	1	0	1
1	1	1	1	1
1	0	1	0	1

$B2 \cdot C1$

$A2 \cdot C1$

$$C2 = A2 \cdot B2 \cdot \overline{C1} + B2 \cdot C1 + A2 \cdot C1$$

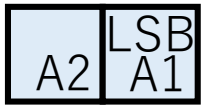
算術演算 (+): 加算器(全加算器)

繰り上がり(Carry)予測

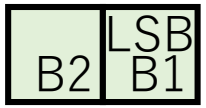
$$S = A + B$$

2桁目の計算

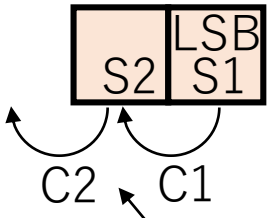
レジスタA



レジスタB



レジスタC



A2	B2	C1	S2	C2
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

ハミング距離=1
に並べ替え

A2	B2	C1	S2	C2
⋮				
0	1	1	0	1
1	1	1	1	1
1	0	1	0	1

$B2 \cdot C1$

$A2 \cdot C1$

$$C2 = A2 \cdot B2 \cdot \overline{C1} + B2 \cdot C1 + A2 \cdot C1$$
$$= A2 \cdot B2 \cdot \overline{C1} + A1 \cdot B1 \cdot B2 + A1 \cdot B1 \cdot A2$$