コンピュータの世界の計算を体験してみよう！

http://www.ee.t-kougei.ac.jp/Logic

2026年9月26日　オープンキャンパス

東京工芸大学　電気電子コース

行谷時男

はじめに

今や生活に深く溶け込んでいるコンピュータやスマートフォンなどの仕組みって気になったことはありませんか？

コンピュータが出始めの頃から、そして今も変わらないものがあります。

世界中で使用されているコンピュータは論理積、論理和、否定といった論理演算で成りったっています。

コンピュータの中の計算(大学1年)

コンピュータの中では情報を1と0で表記します。

論理和　1or0=1　1or1=1　0or1=1　0or0=0

論理積　1・0=0　1・1=1　0・1=0　0・0=0

否定　$\overline{1}=0$　$\overline{0}=1$

否定を意味する記号

論理和

A君が研究室の鍵を持っている

B君が研究室の鍵を持っている	ある(1)	なし(0)
ある(1)	入れる(1)	入れる(1)
なし(0)	入れる(1)	入れない(0)

どっちかが鍵を持っていれば研究室に入れる

論理積

本屋の在庫

手持ちのお金	ある(1)	なし(0)
ある(1)	買える(1)	買えない(0)
なし(0)	買えない(0)	買えない(0)

お金と本屋に在庫があることで本が買える

0と1で数値を表現するには(大学1年)

10進数で11を2進数で表記すると1011になります。

$$(11)_{10} = (1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$
$$= 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 8 + 2 + 1 = (11)_{10}$$

位取り記数法（くらいどりきすうほう）

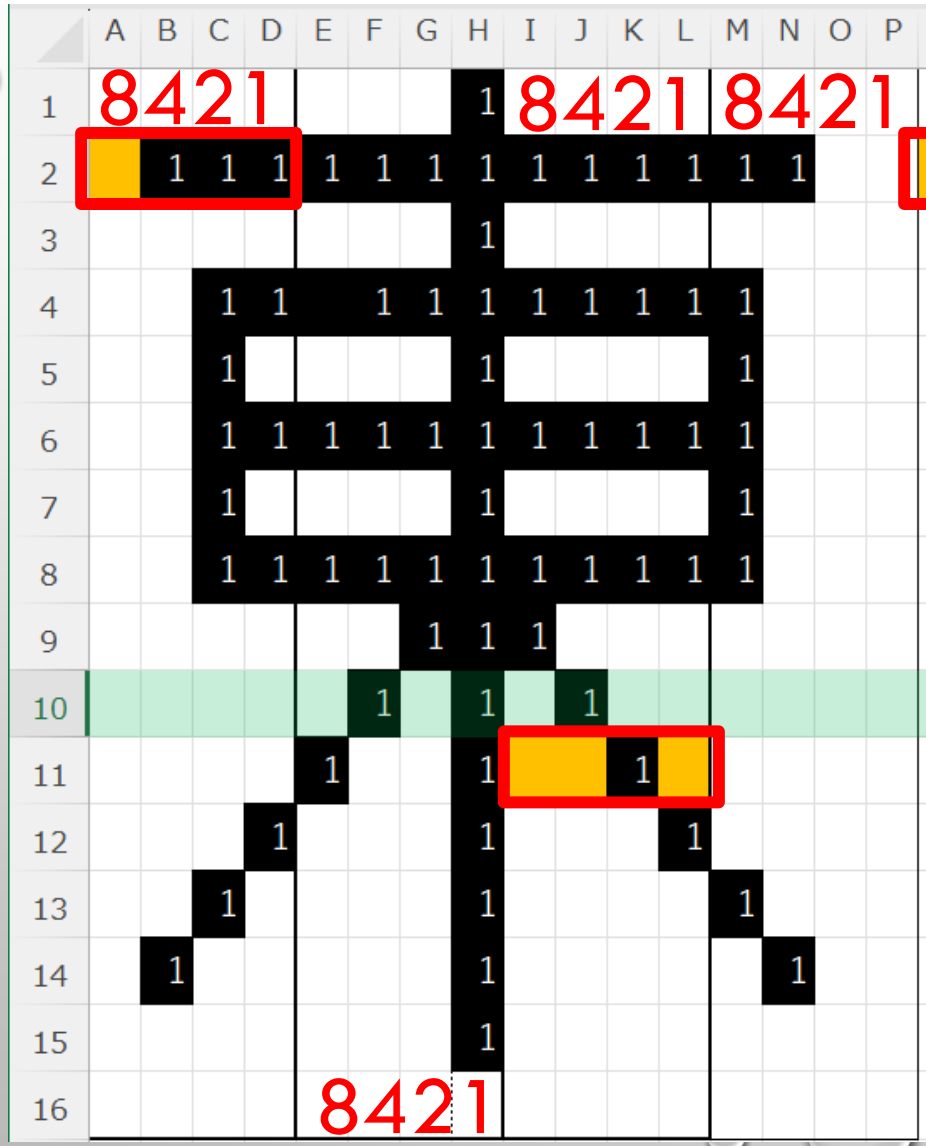
$$4 \times 10^5 + 2 \times 10^4 + 1 \times 10^3 + 5 \times 10^2 + 8 \times 10^1 + 6 \times 10^0 = 421586$$

5本の指で幾つまでカウントできるかというクイズがよくありました。

0から31まで（32個の数）を数えることができます

では2進数で計算する足し算は？

昔のコンピュータでの文字の表現はドット絵です。
2進数を使って文字を表現していました。



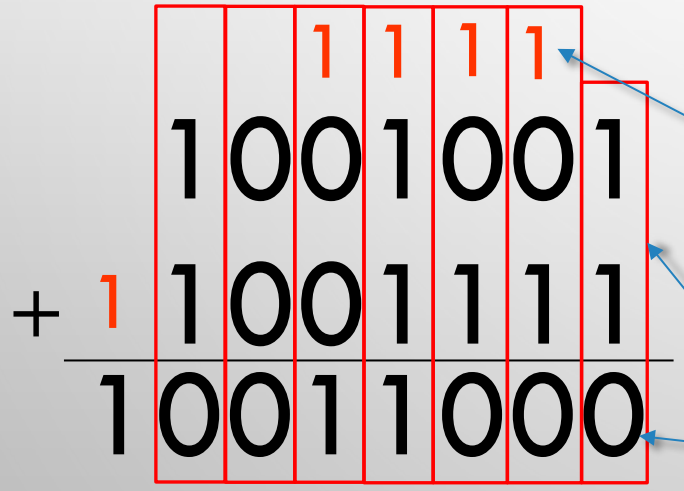
	Q	R	S	T
1	0000	0001	0000	0000
2	0111	1111	1111	1100
3	0000	0001	0000	0000
4	0011	0111	1111	1000
5	0010	0001	0000	1000
6	0011	1111	1111	1000
7	0010	0001	0000	1000
8	0011	1111	1111	1000
9	0000	0011	1000	0000
10	0000	0101	0100	0000
11	0000	1001	0010	0000
12	0001	0001	0001	0000
13	0010	0001	0000	1000
14	0100	0001	0000	0100
15	0000	0001	0000	0000
16	0000	0000	0000	0000

空白の部分は0

2進数の足し算(大学2年)

加算(2入力)　0+1=1, 1+0=1, 0+0=0　1+1=10

加算(3入力)　0+0+1=1, 0+1+0=1, 1+0+0=1, 0+0+0=0
0+1+1=10, 1+0+1=10, 1+1+0=10
1+1+1=11



A	B	C	桁上がり2	和1
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

1の数が3個

真理値表

論理変数は0か1の値を持つ

	A	B	C	桁上がり2	和 S 1
	0	0	0	0	0
$\overline{A}\overline{B}C$	0	0	1	0	1
$\overline{A}B\overline{C}$	0	1	0	0	1
	0	1	1	1	0
$A\overline{B}\overline{C}$	1	0	0	0	1
	1	0	1	1	0
	1	1	0	1	0
ABC	1	1	1	1	1

- 演算結果が1のところに注目！
- 論理変数が0ならば否定とし、1ならばそのまま記述する。
- これを論理積でまとめる。

001

$\overline{A}\overline{B}C$ $\overline{A}$と$\overline{B}$とCの論理積

最後に1の部分の論理変数を論理積でまとめたものを、論理和で結合する。

$$S = \overset{001}{\overline{A}\overline{B}C} + \overset{110}{A\overline{B}\overline{C}} + \overset{100}{A\overline{B}\overline{C}} + \overset{111}{ABC}$$

論理和

真理値表ができると機械的に複雑な論理式を作ることができます。

2進数、論理演算、簡単に思えたものが複雑になってきました。(大学2年)

真理値表から機械的に論理式ができます。

加法標準形といいます。

A	B	C	桁上がり Co		和 S	
0	0	0	0		0	
0	0	1	0		1	$\bar{A}\bar{B}C$
0	1	0	0		1	$\bar{A}B\bar{C}$
0	1	1	1	$\bar{A}BC$	0	
1	0	0	0		1	$A\bar{B}\bar{C}$
1	0	1	1	$A\bar{B}C$	0	
1	1	0	1	$AB\bar{C}$	0	
1	1	1	1	ABC	1	ABC

演算結果が1の所だけに注目して作ります。
AやBやCが0の所は否定形で、1の所はそのまま書きます。

$$Co = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$
$$= \bar{A}BC + ABC + A\bar{B}C + ABC$$
$$+AB\bar{C} + ABC = BC(\bar{A} + A)$$
$$+AC(\bar{B} + B) + AB(\bar{C} + C)$$
$$= BC + AC + AB$$

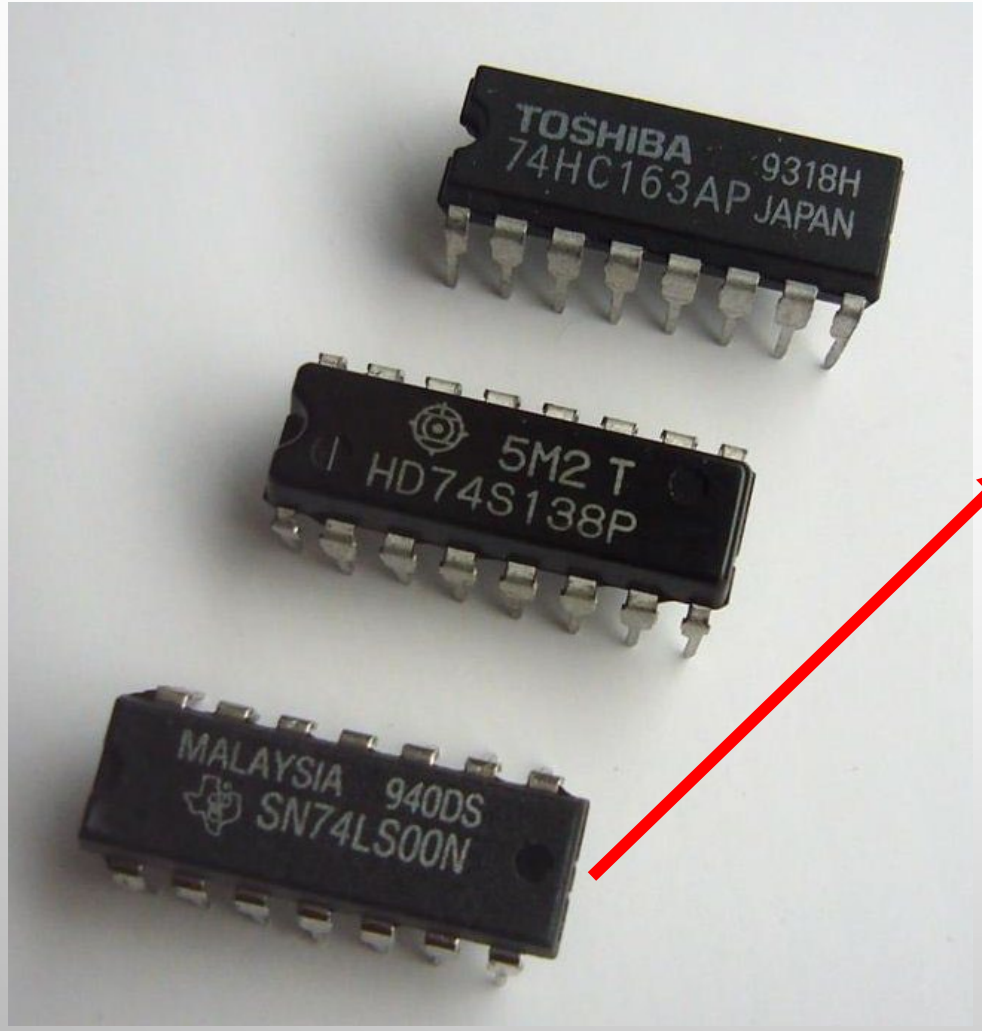
$$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= (\bar{A}B\bar{C} + A\bar{B}\bar{C}) + (\bar{A}\bar{B}C + ABC)$$
$$= \bar{C}(\bar{A}B + A\bar{B}) + C(\bar{A}\bar{B} + AB)$$
$$= \bar{C}(A \oplus B) + C(\overline{A \oplus B})$$
$$= (A \oplus B) \oplus C$$

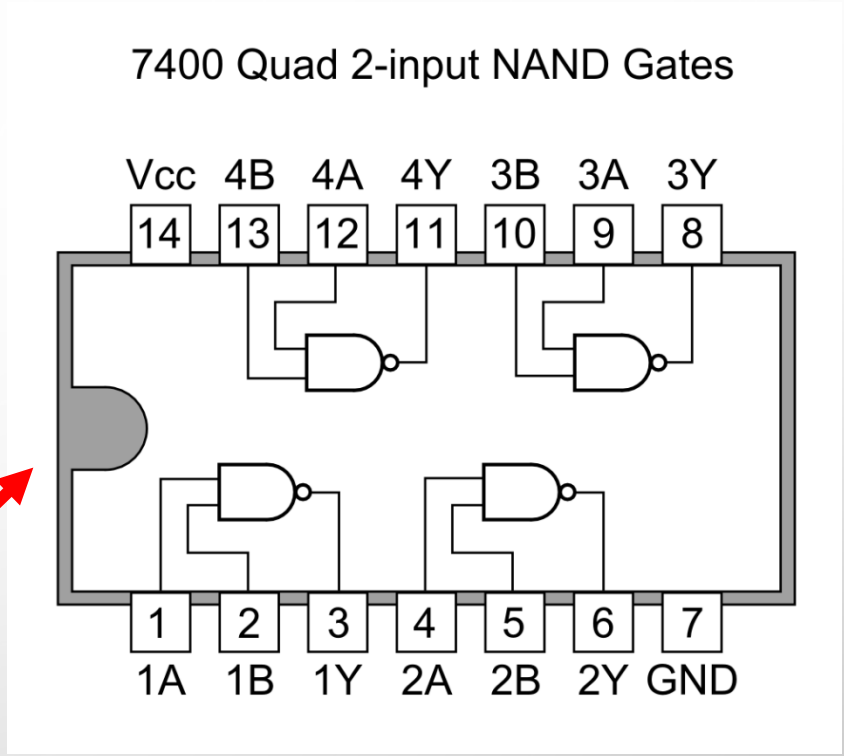
排他的論理和　$\bar{A}B + A\bar{B} = A \oplus B$

排他的論理和の否定　$\bar{A}\bar{B} + AB = \overline{A \oplus B}$

大学2年の論理回路で登場

論理ゲート



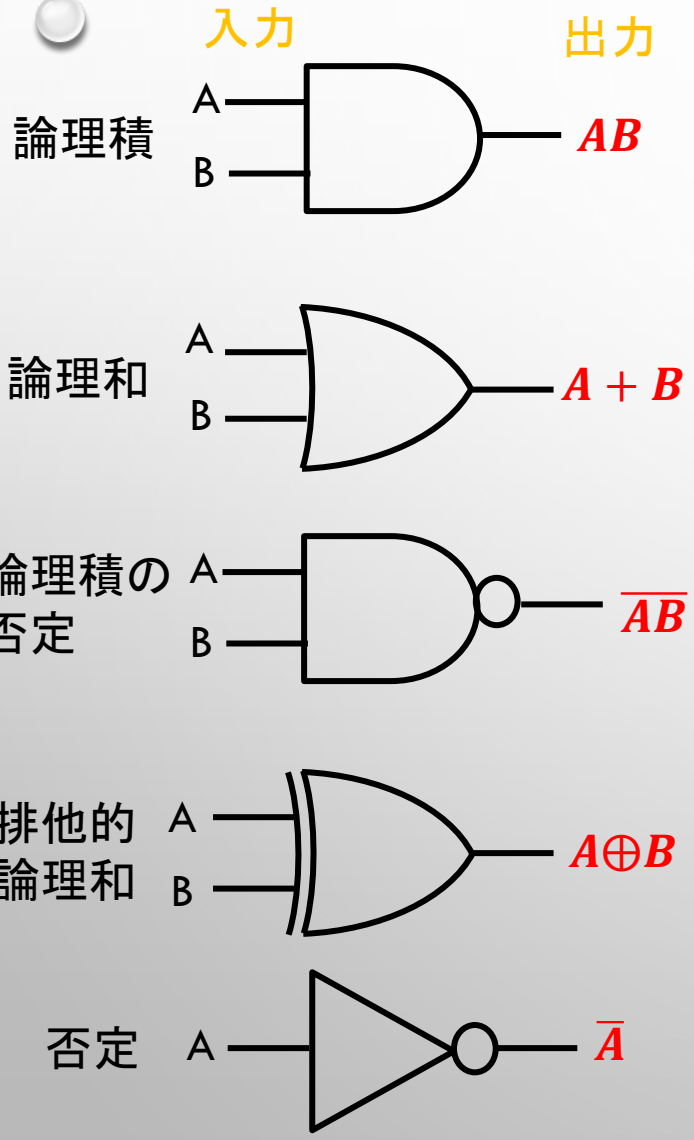


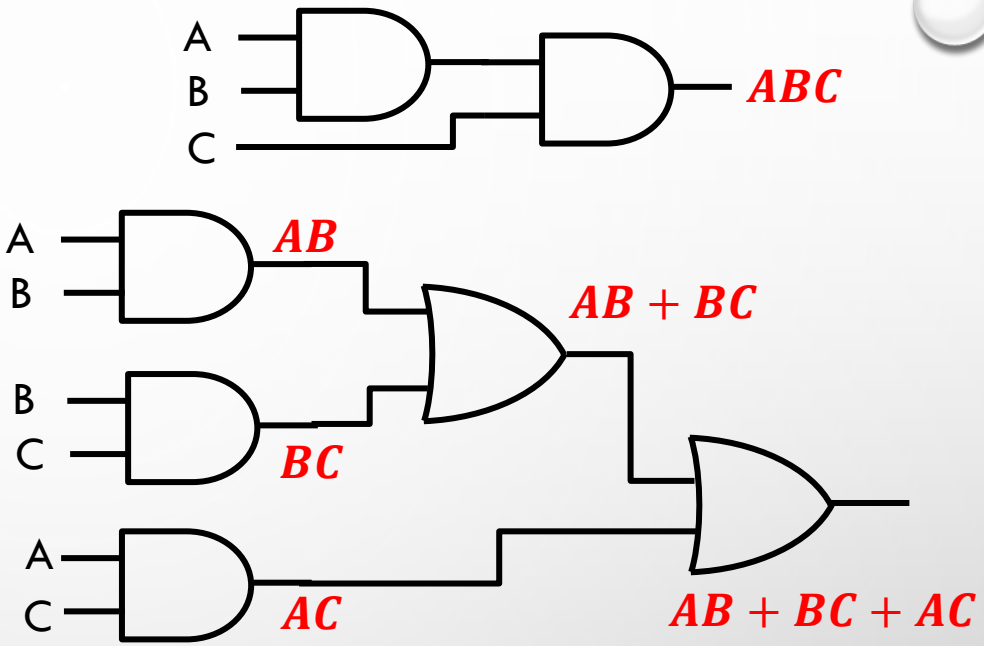
Tosaka - Author's original, CC 表示 3.0, https://commons.wikimedia.org/w/index.php?curid=6518690による

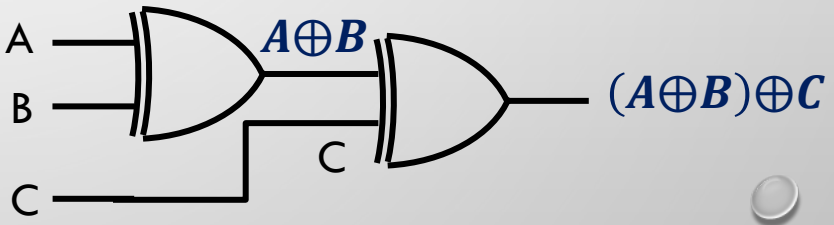
S. Kaba - 投稿者自身による著作物, パブリック・ドメイン, https://commons.wikimedia.org/w/index.php?curid=3432096による

論理式を論理回路へ(大学2年)

ゲート記号



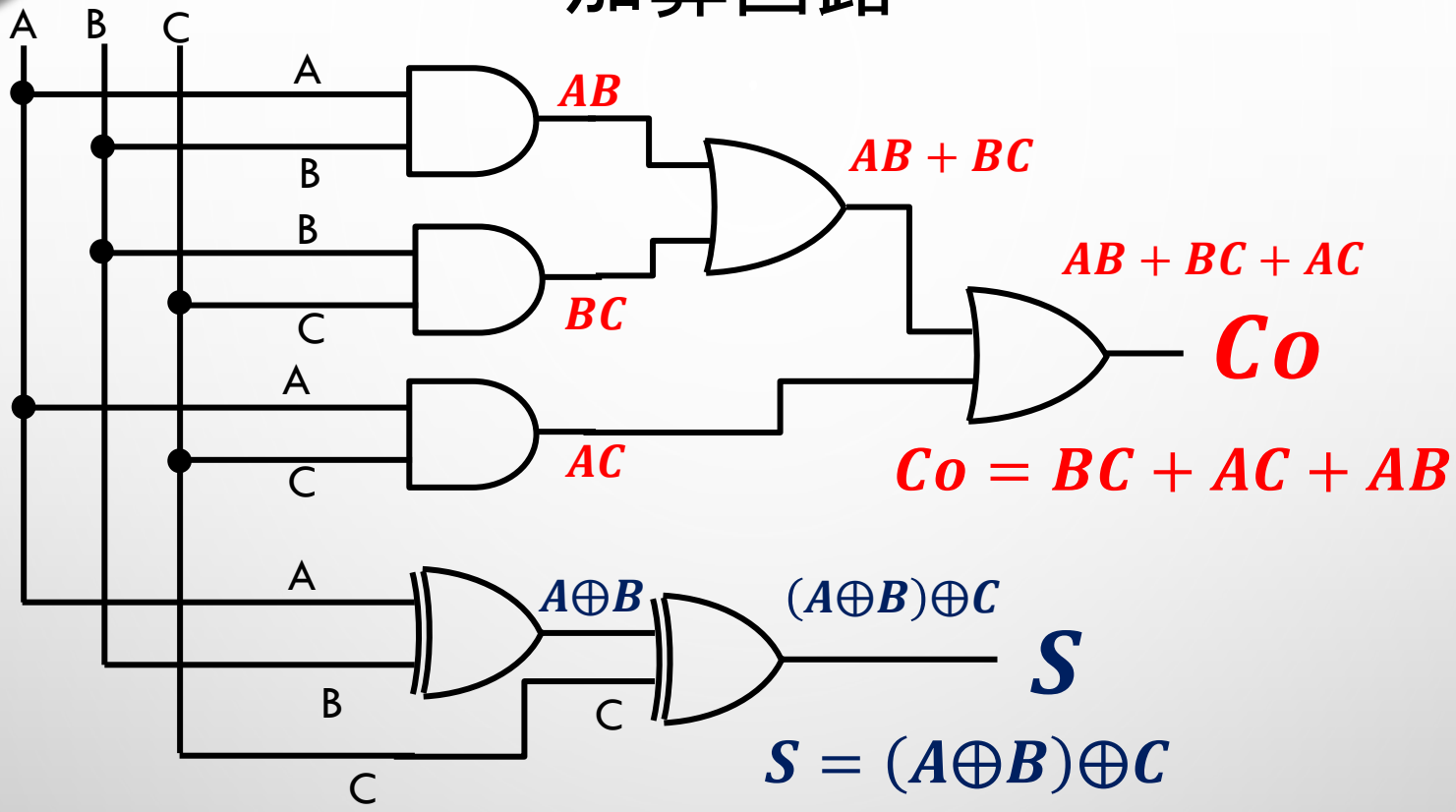




$$Co = BC + AC + AB$$

$$S = (A \oplus B) \oplus C$$

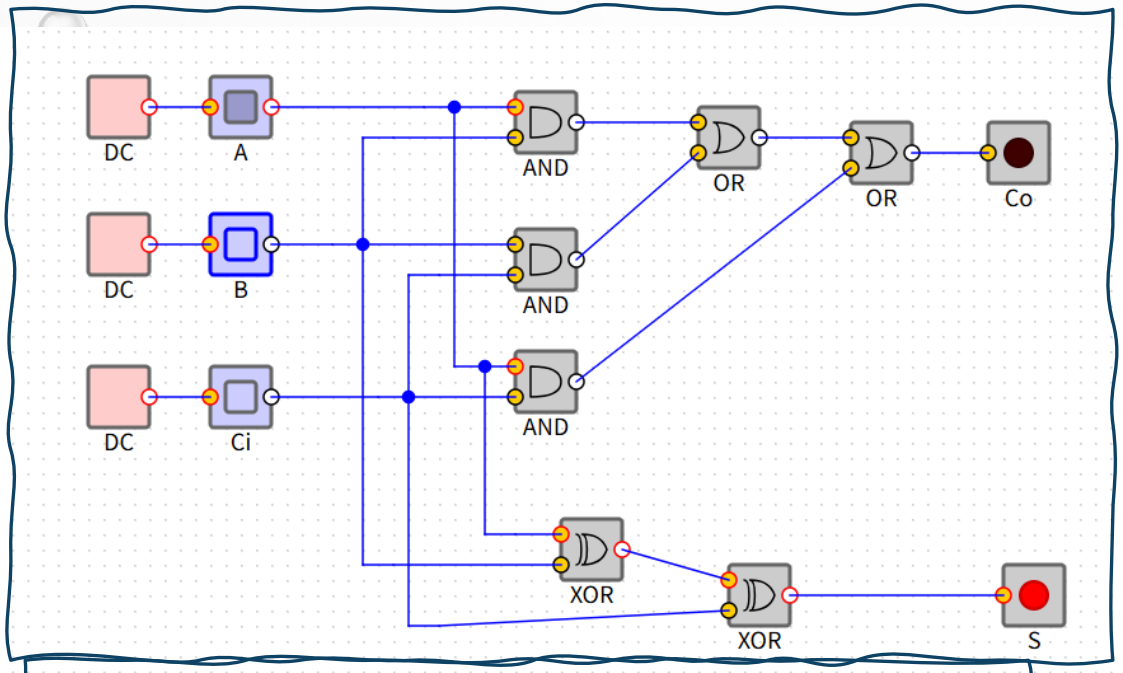
加算回路



ジョージ・ロバート・スティビッツ　1937年11月にModel Kをベル研究所で開発した。これが2進数を計算機に使った最初の例と言われている。

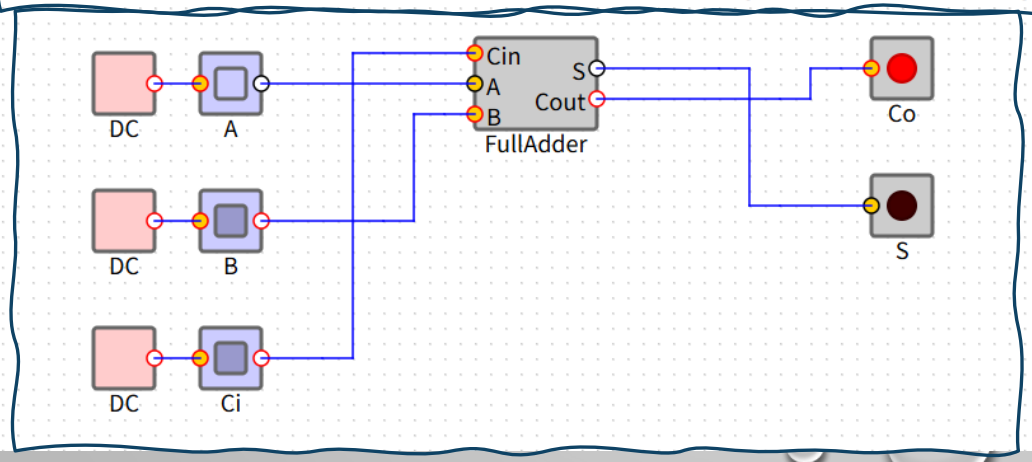
クロード・エルウッド・シャノン　1937年に修士論文（A Symbolic Analysis of Relay and Switching Circuits）で2進法で論理演算が計算機で実現可能であることを示す。

論理回路シミュレータで確認



ANDやOR、XORで構成した加算回路

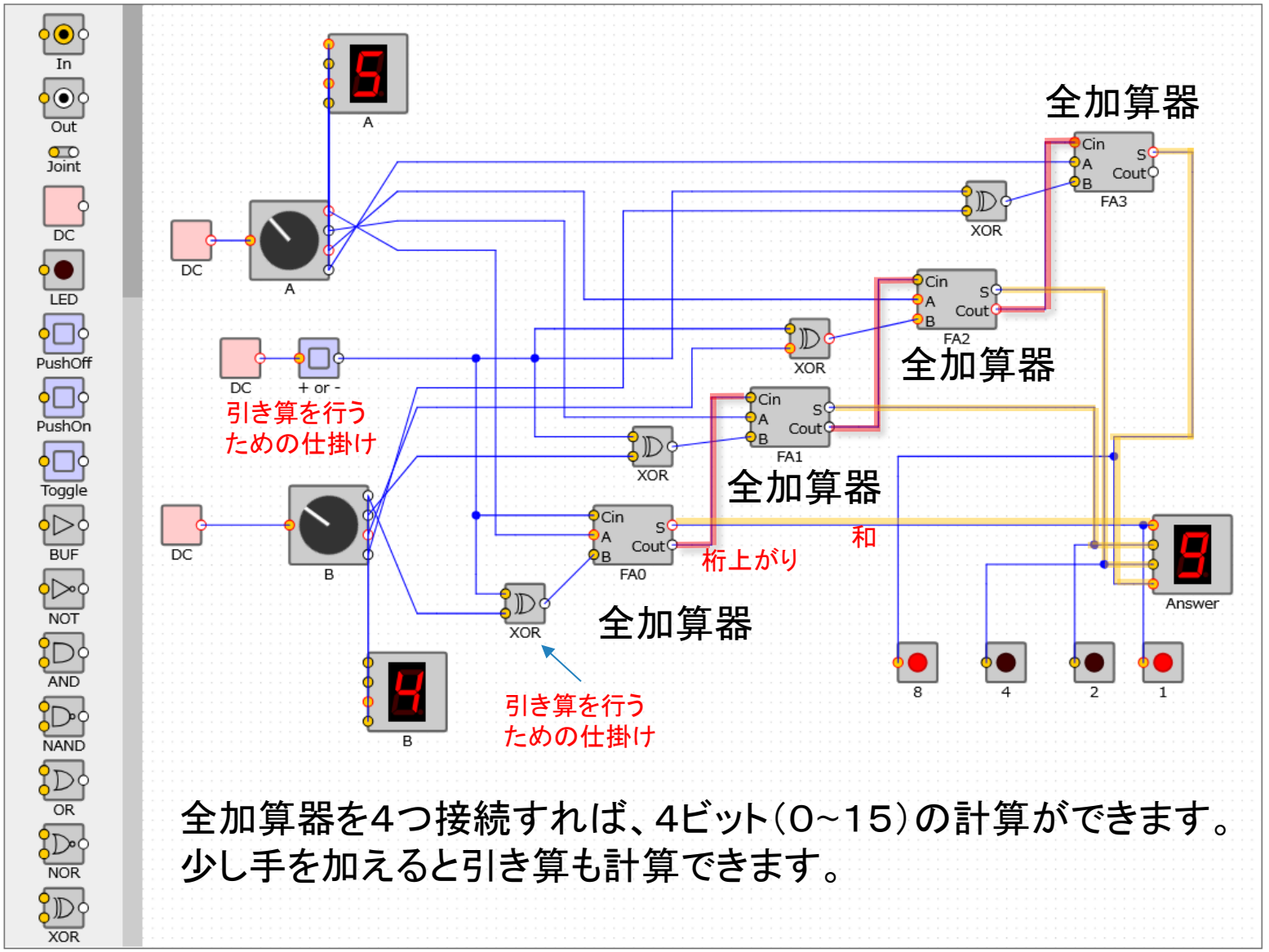
Aが1、BとCiが0のとき、和は1となり、桁上がりは発生しません。



シミュレータの加算回路を使用

Aが0、BとCiが1のとき、和は0となり、桁上がりが発生します。

桁数を増やすためには（大学2年生）

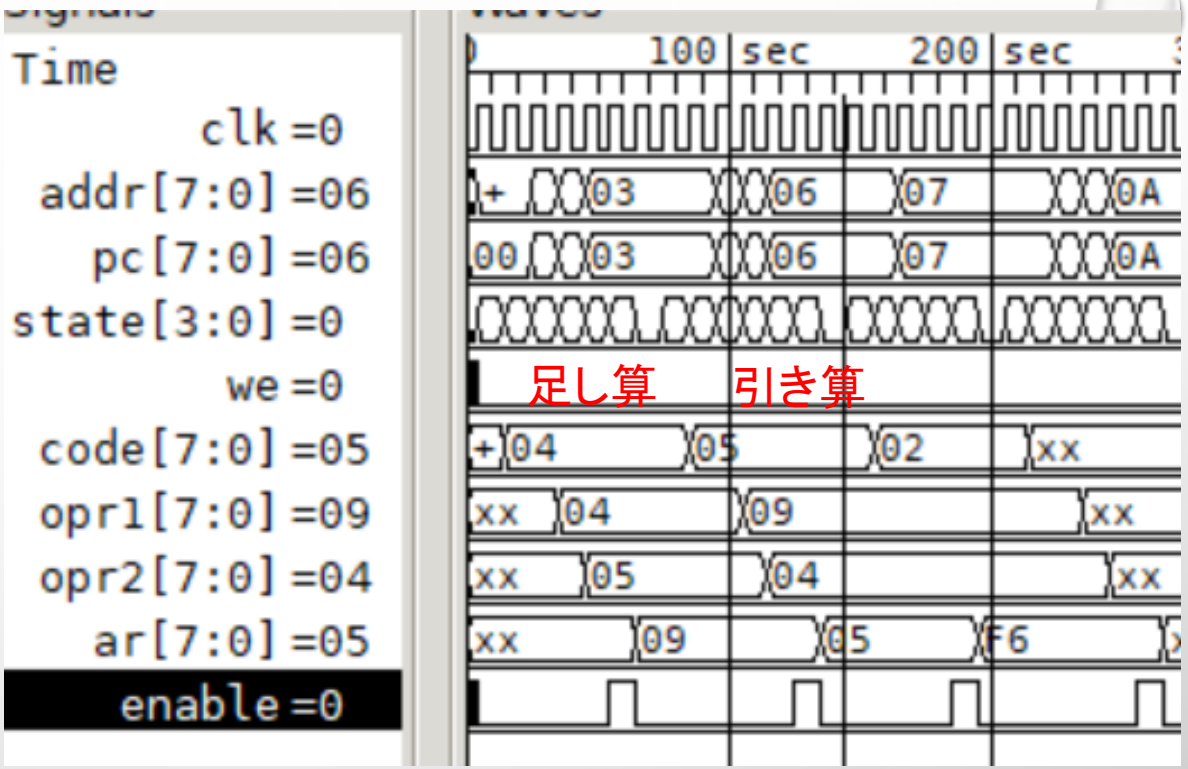


全加算器を4つ接続すれば、4ビット（0~15）の計算ができます。
少し手を加えると引き算も計算できます。

では現在のロジック回路の設計方法は？

```
module alu (a,b,opcode,x,enable);
input enable;
input [7:0] a,b;
input [7:0] opcode;
output [7:0] x;
reg    [7:0] x;

always @(posedge enable) begin
  case (opcode)
        `AND: x <= a&b;
        `OR : x <= a|b;
        `NOT: x <= ~a;
        `XOR: x <= a^b;
        `ADD: x <= a+b;
        `SUB: x <= a-b;
        default: x <= 8'bxxxxxxxx;
  endcase
end
endmodule
```




```
`define AND 8'd0
`define OR  8'd1
`define NOT 8'd2
`define XOR 8'd3
`define ADD 8'd4
`define SUB 8'd5
```

4+5　　9-4

授業で使用しているCPUのコードより

まとめ

- 紹介しました加算回路は、これを多段にすることにより、より多くの桁数を計算することができます。しかしながらそのままでは演算に時間がかかります。演算速度を早くするために、桁上がりを先に検知する等工夫がされています。

- コンピュータを2進法で動作させることは今日でも変わっていません。

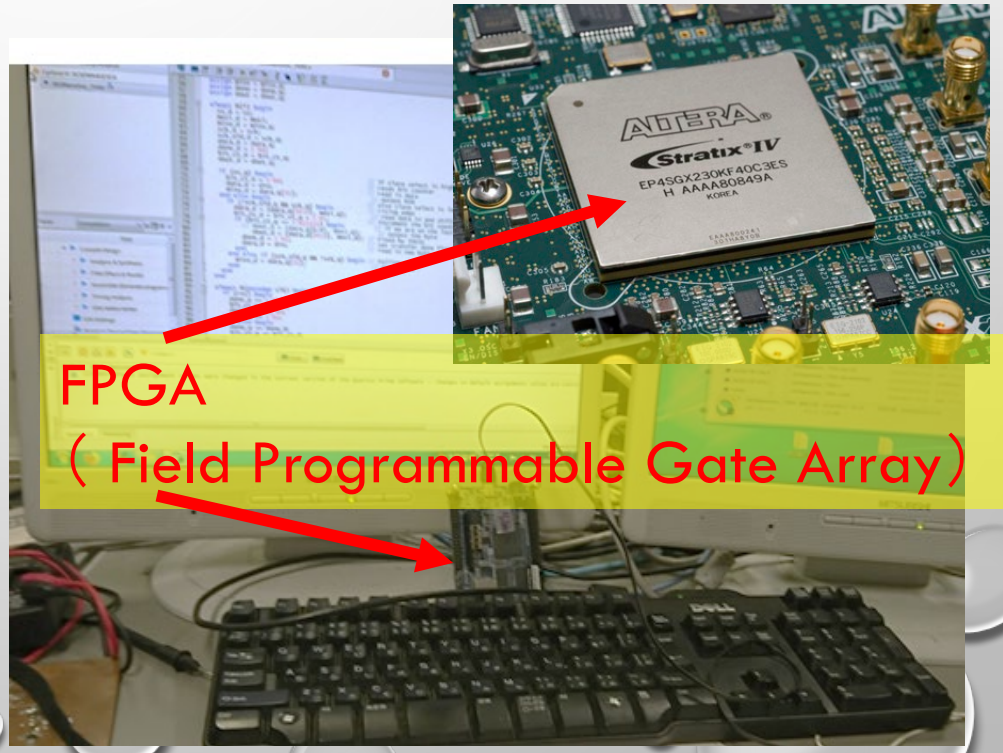
- ディジタル回路のエンジニアは求人が大量に発生しています。理由はボタンを押して動作させるものなど電気仕掛けで動作するものすべてが対象になるからです。

最近は卓上で複雑なロジックを構成することができ、個人で数万ゲート程度でも開発することができます。

研究室レベルで集積回路を開発することができる時代になったのです。

検索「FPGA　AI」

検索「検索エンジン　FPGA」



論理回路シミュレータを動かしてみよう

http://www.ee.t-kougei.ac.jp/Logic

シミュレータのページに移動

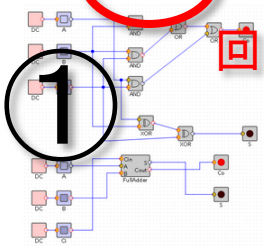
論理回路シミュレータ

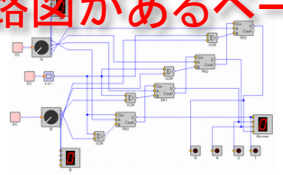
論理回路シミュレータ　SimcirJS（配布元）

⑤

以下のリンク先にあるコードを[Copy text]ボタンを押すことでクリップボードにコピーし、
論理回路シミュレータ　SimcirJS テキスト入力フィールドにペーストすることで書き込み、
[set data]を押すと加算回路が現れます。

全加算器　　4ビット全加算器

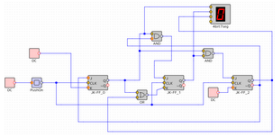


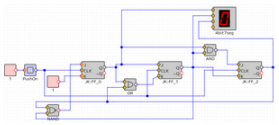


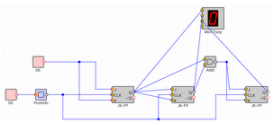
①

回路図があるページに移動

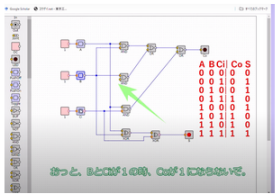
5進カウンタ　　7進カウンタ　　8進カウンタ

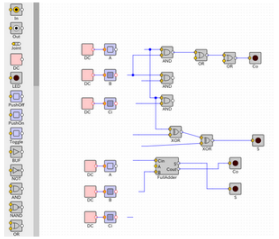






SimCirJSの使い方　　全加算器(練習用)　　おまけ





1ビット全加算器 (動画)

押すとウィンドウが現れます。

②

全加算器

Copy text

Back

```
{
  "width":1024,
  "height":768,
  "showToolbox":true,
  "toolbox":[
    {"type":"In"},
    {"type":"Out"},
```

④

前のページに戻ります

www.ee.t-kougei.ac.jp

コピーできました！：

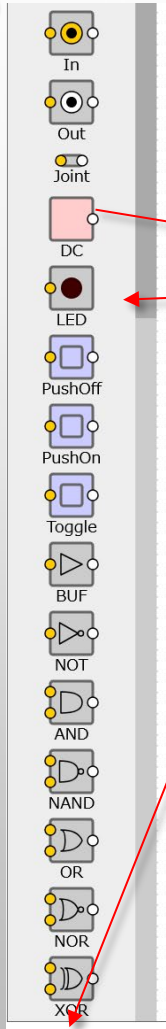

```
{
  "width":1024,
  "height":768,
  "showToolbox":true,
  "toolbox":[
    {"type":"In"},
    {"type":"Out"},
    {"type":"Joint"},
    {"type":"DC"},
    {"type":"LED"},
    {"type":"PushOff"},
    {"type":"PushOn"},
    {"type":"Toggle"},
    {"type":"BUF"},
    {"type":"NOT"},
    {"type":"AND"},
    {"type":"NAND"},
    {"type":"OR"},
    {"type":"NOR"},
    {"type":"XOR"},
    {"type":"XNOR"},
    {"type":"OSC"},
    {"type":"7seg"},
    {"type":"16seg"},
    {"type":"4bit7seg"},
    {"type":"RotaryEncoder"},
    {"type":"BusIn"},
    {"type":"BusOut"},
    {"type":"RS-FF"},
    {"type":"JK-FF"},
    {"type":"T-FF"},
    {"type":"D-FF"},
    {"type":"8bitCounter"},
    {"type":"HalfAdder"},
    {"type":"FullAdder"},
    {"type":"4bitAdder"},
    {"type":"2to4BinaryDecoder"},
    {"type":"3to8BinaryDecoder"},
    {"type":"4to16BinaryDecoder"}
  ],
  "devices":[
    {"type":"DC","id":"dev0","x":120,"y":272,"label":"DC"},
    {"type":"DC","id":"dev1","x":120,"y":192,"label":"DC"},
    {"type":"DC","id":"dev2","x":120,"y":120,"label":"DC"},
```


③

OK

コピーします。

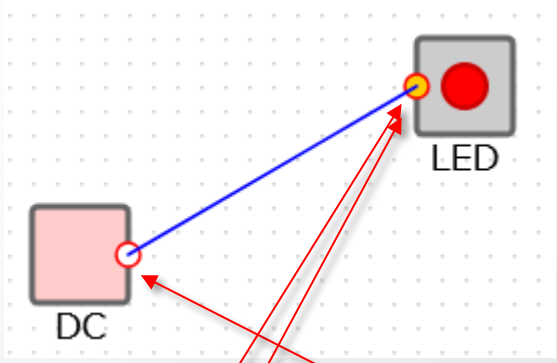
論理回路シミュレータを動かしてみよう



部品はマウスカーソルを対象の部品の上に移動させ、マウスの左ボタンを押したまま移動させると配置することができます

部品の消去はマウスカーソルを対象の部品上に移動させ、マウスの左ボタンを押したまま、部品リストに戻すと消えます。

②get dataをマウスの左ボタンでクリックすると、回路図のデータがテキストエリアに現れます。これをメモ帳等でテキストデータをコピーペーストして保存します。



配線はマウスのカーソルを〇の上に移動させ、マウスの左ボタンを押したまま、接続先の〇までマウスのカーソルを移動させます。

配線の消去はマウスのカーソルをオレンジ色の〇の上に移動させ、マウスの左ボタンを押すと消えます。

②set dataをマウスの左ボタンでクリックすると、回路図が現れます。

①コピーした回路データはここにCtrl+vでペーストする。

論理回路シミュレータを動かしてみよう

SimCirJSによる全加算器のシミュレーション解説動画

https://youtu.be/BFxZFptK71M