

# デジタルに強くなる講座

コンピューター入門

福野泰介 @taisukef



一日一創

**DX、デジタル技術、デジタル庁  
デジタルと言えば、コンピューター！**

# コンピューターのつくりかた

デジタルに強いコンピューターがほしい



ゲームも動画もデジタル (=数) にできる  
すべての数は、0か1の二進法にできる  
二進法は論理式で計算できる  
論理式と記憶は電子回路で表せる  
電子回路は小さくできる



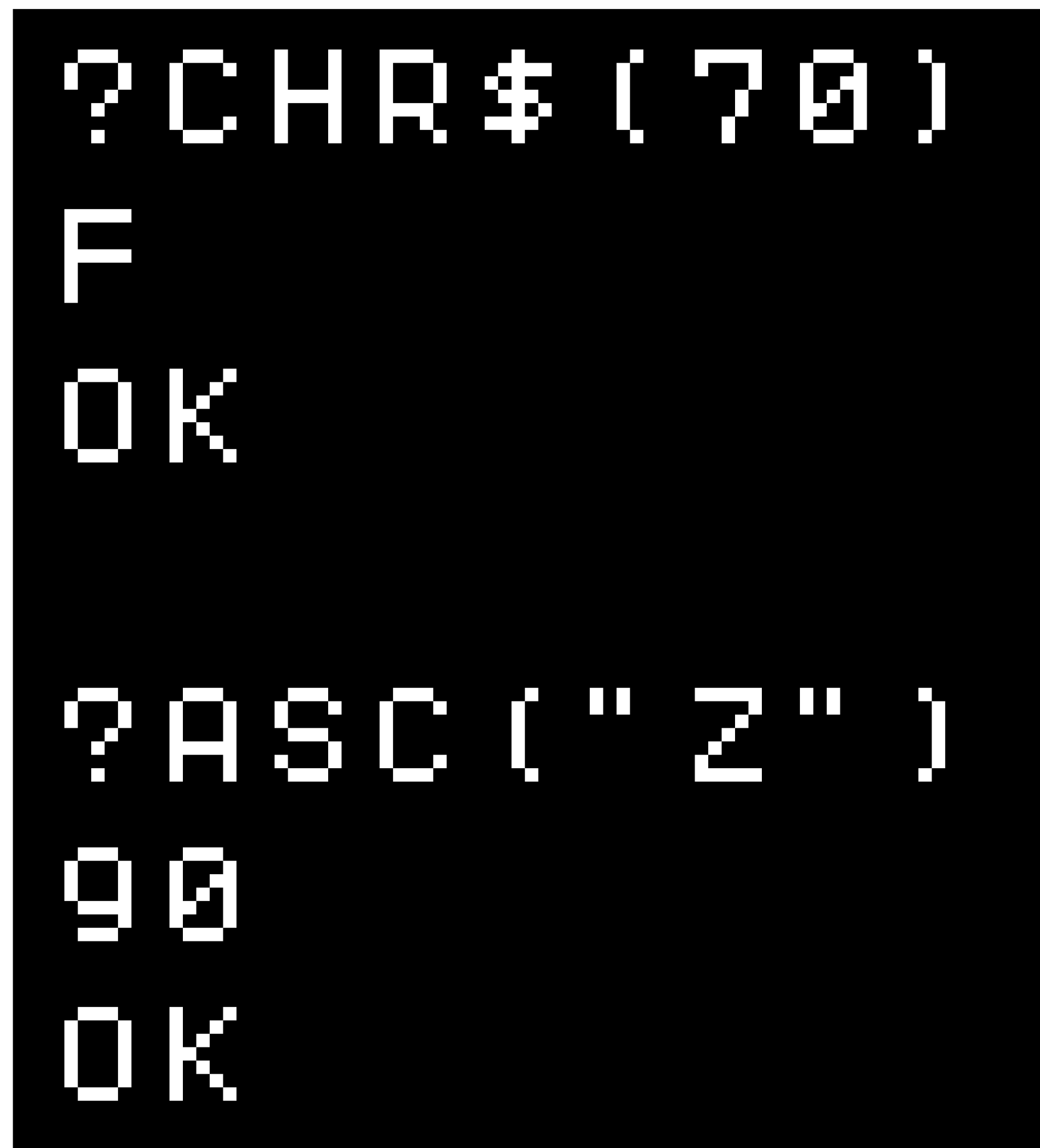
つまり、電子回路を詰めこめばコンピューターができる

デジタル  
ゲームは数？

# 文字を数で表せる

IchigoJam キャラクター 10 シンズウ 1.4~

\	0	1	2	3	4	5	6	7	8	9
0		■	▒	░	▤	✱	▣	✂	←	--
10	+	腎	◌	■	□	▣	▣	□		--
20	_	○	◎		=	/	\	✂	←	→
30	↑	↓		!	"	#	\$	%	&	/
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~	⇒		■
130	■	▒	░	▤	▥	▦	■	▤	▥	▦
140	▒	▤	▥	■	・	—		+	+	+
150	▤	▥	▦	▧	▨	▩	▪	▫	▬	▭
160	¥	。	「	」	、	・	ヲ	ァ	ィ	ゥ
170	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス
180	セ	ソ	タ	チ	ツ	テ	ト	ナ	ニ	ヌ
200	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ	ミ	ム
210	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ
220	ワ	ン	ゝ	ゞ	←	→	↑	↓	♣	♥
230	♣	♦	○	●	10	🏠	🕒	🕒	🎵	🕒
240	♣	♣	ゝ	ゝ	ゝ	0	🏠	🏠	🏠	🏠
250	🏠	🏠	[	🏠	]	🏠				



コードから文字へ  
CHR\$, キャラ

文字からコードへ  
ASC、アスキー

漢字などの表  
ユニコードもあるよ

## かわくだりゲーム（カーソル左右でよける！）

```
10 CLS : X=16
20 LC X,5:?"0"
30 LC RND(32),23:?"*"
35 WAIT 3
36 X=X-BTN(28)+BTN(29)
37 IF SCR(X,5) END
40 GOTO 20
```

プログラムは文字で表せる、つまりデジタル！

デジタル  
動画は数？

# 絵は数で表そう



←1画素を0か1で表して

8x8、64画素で表現した「ネコ (ALT+C)」

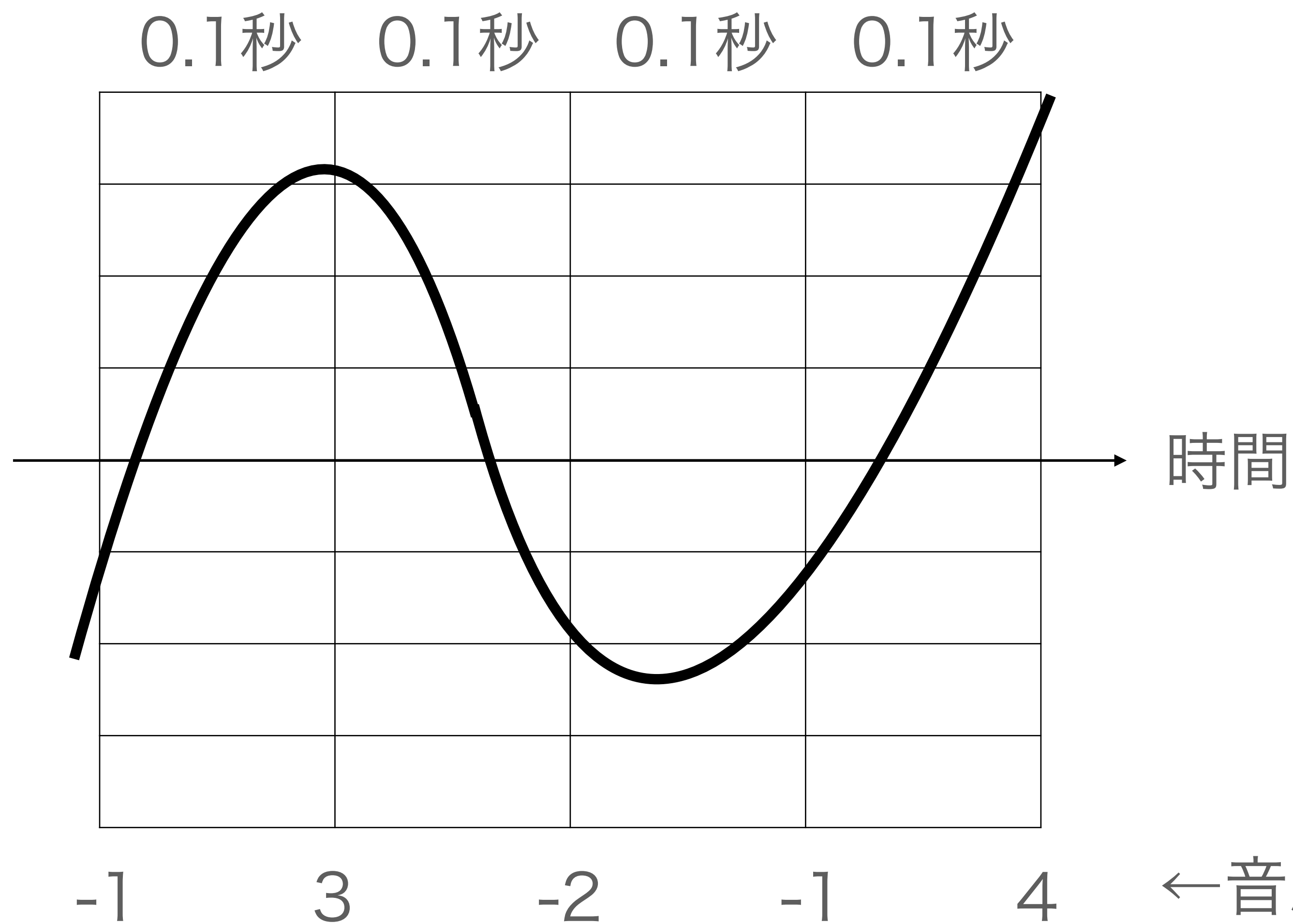
```
FOR I=0 TO 7: ?BIN$(PEEK(236*8+I),8):NEXT
```

写真は1画素を赤青緑の光の量を  
それぞれ256段階で表し、

1000x1000の100万画素ほどある数の集合



# 音も数で表そう



細かく区切るほど、高音質！  
(ただし、データ量は増える)

←音圧を0.1秒毎に計測して  
プラスマイナス4段階の数にしたもの

動画＝画像たくさん＋音  
つまり、数！

# コンピュータのつくりかた

デジタルに強いコンピュータがほしい



ゲームも動画もデジタル (=数) にできる

すべての数は、0か1の二進法にできる

二進法は論理式で計算できる

論理式と記憶は電子回路で表せる

電子回路は小さくできる



つまり、電子回路を詰めこめばコンピュータができる

片手でいくつまで数えられる？



# 指二進法





# N進法のヒミツ

	使う文字	1桁で表せる数の種類	2桁で表せる数の種類	5桁で表せる数
10進法	0から9までの10種類を使う	10	100 (=10 x 10)	100000 (=10の5乗)
2進法	0から1までの2種類を使う	2	4 (= 2 x 2)	32 (=2の5乗)
N進法	N種類を使う	N	N x N	Nの5乗

**なぜ三進法ではなく二進法か？**

チャレンジ、指三進法！



0か1か

これが一番ノイズに強い

問題、10進法の10  
2進法では？

# IchigoJam で確かめよう

```
?BIN$(10)
```

```
1010
```

```
OK
```

```
?`1010
```

```
10
```

```
OK
```

数を2進法で表示する表示コマンド BIN\$

(バック・クォート)

2進法表記で数を表す ` 表記

# コンピューターのつくりかた

デジタルに強いコンピューターがほしい



ゲームも動画もデジタル（=数）にできる

すべての数は、0か1の二進法にできる

二進法は論理式で計算できる

論理式と記憶は電子回路で表せる

電子回路は小さくできる



つまり、電子回路を詰めこめばコンピューターができる

# 二進法の足し算

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

$$A + B = C$$

# 二進法の足し算

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

←くりあがり

$$A + B = CD$$

# 2進法世界の九九は簡単！

	0	1
0	0	0
1	0	1

$$1 \times 1 = 1$$

# とつぜん、数理論理学入門

	偽	真
偽	偽	偽
真	偽	真

真と偽の数学  
1と0と言い換えると

AND 論理積  
(アンド)



# とつぜん、数理論理学入門

	0	1
0	0	0
1	0	1

AND 論理積  
(アンド)

2進法の  
掛け算とANDは  
一緒

掛け算は  
論理積で実現できる！

# 数理論理学の基本

	0	1
0	0	1
1	1	1

OR 論理和  
(オア)

	0	1
0	0	0
1	0	1

AND 論理積  
(アンド)

0	1
1	0

NOT 論理否定  
(ノット)

# 二進法の足し算を論理式で表現

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

$$A + B = CD$$

$$C = A \text{ AND } B$$

※AとBが両方1の時だけCは1になる

$$D = (A \text{ OR } B) \text{ AND } (\text{NOT } C)$$

※AとBがどちらから1で、  
かつ、Cが1でない時にDは1になる

→ 足し算も論理式で実現できる！

# コンピューターのつくりかた

デジタルに強いコンピューターがほしい



ゲームも動画もデジタル (=数) にできる  
すべての数は、0か1の二進法にできる

**二進法は論理式で計算できる**

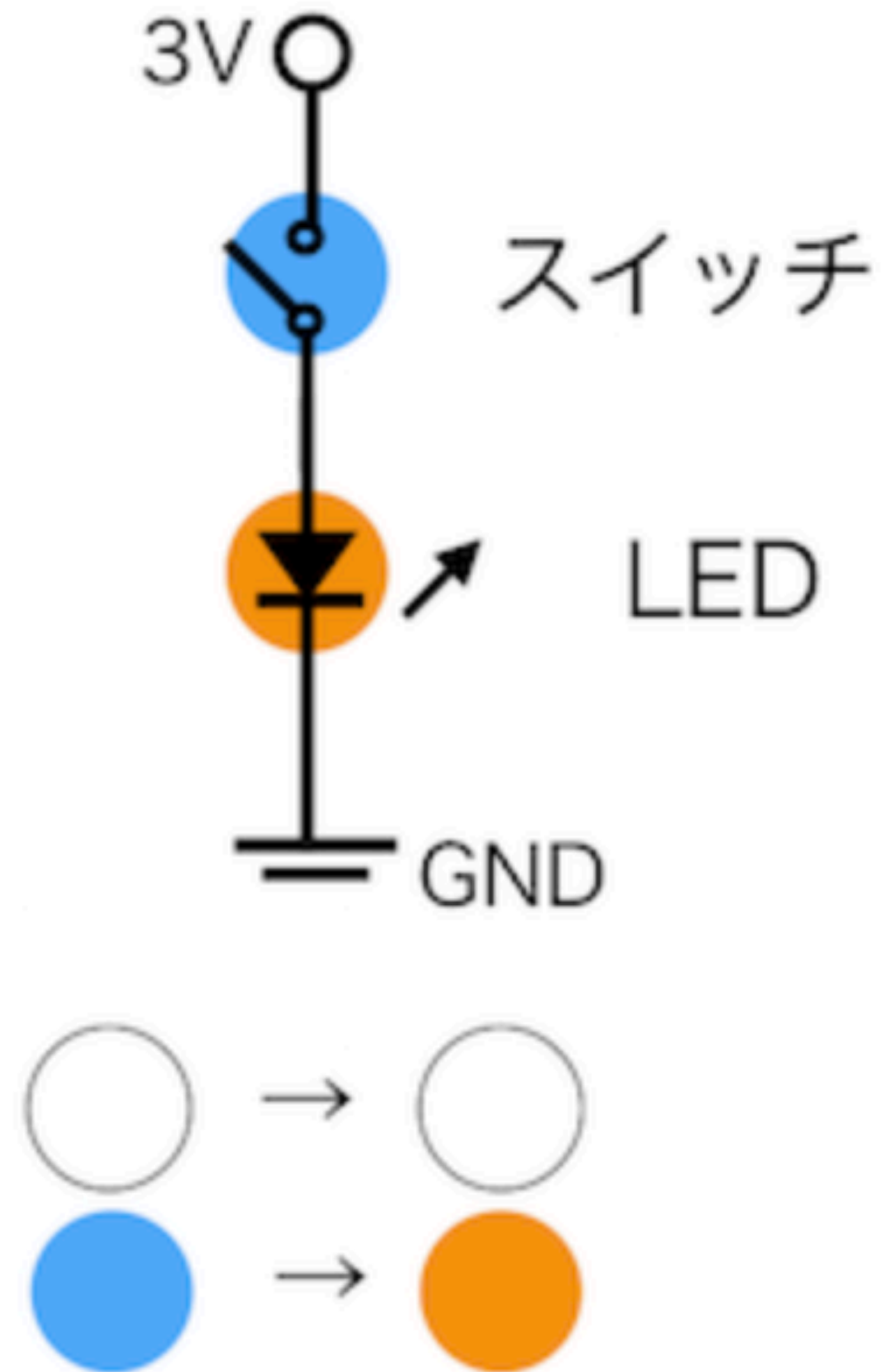
論理式と記憶は電子回路で表せる

電子回路は小さくできる



つまり、電子回路を詰めこめばコンピューターができる

# とつぜん、電気回路

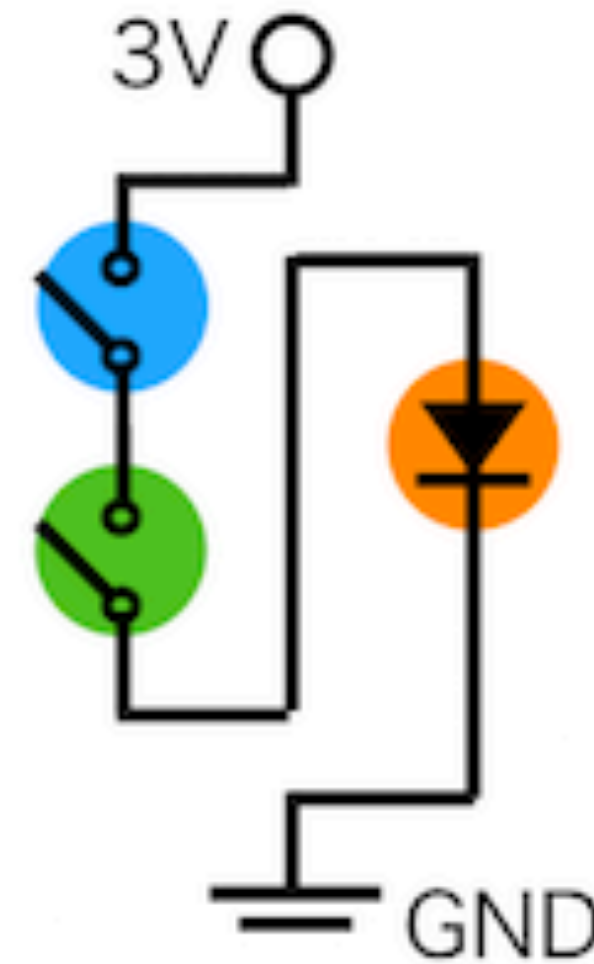


スイッチ  
押すと光る回路

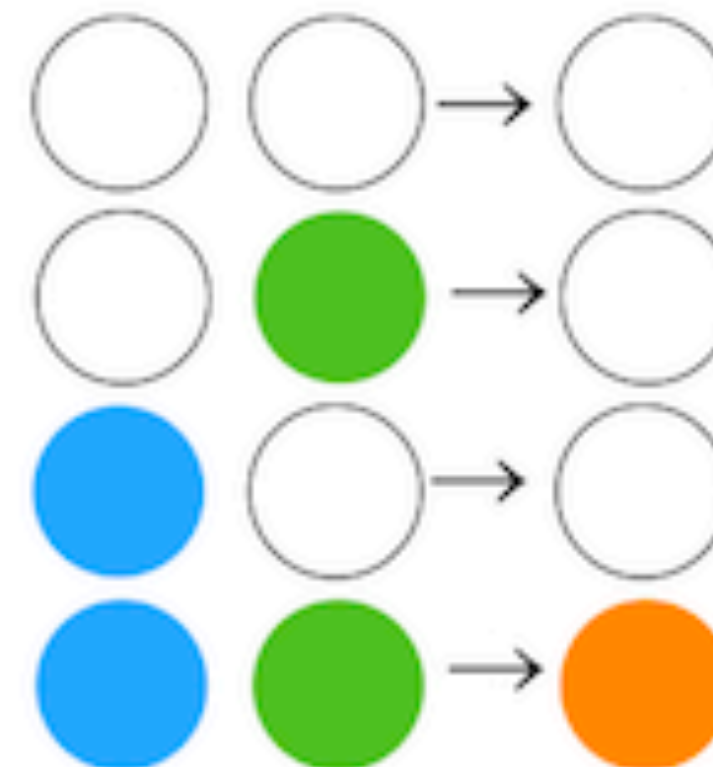
# 電気回路でANDを作ろう

論理積(AND)

2つのスイッチ  
両方押すと光る回路

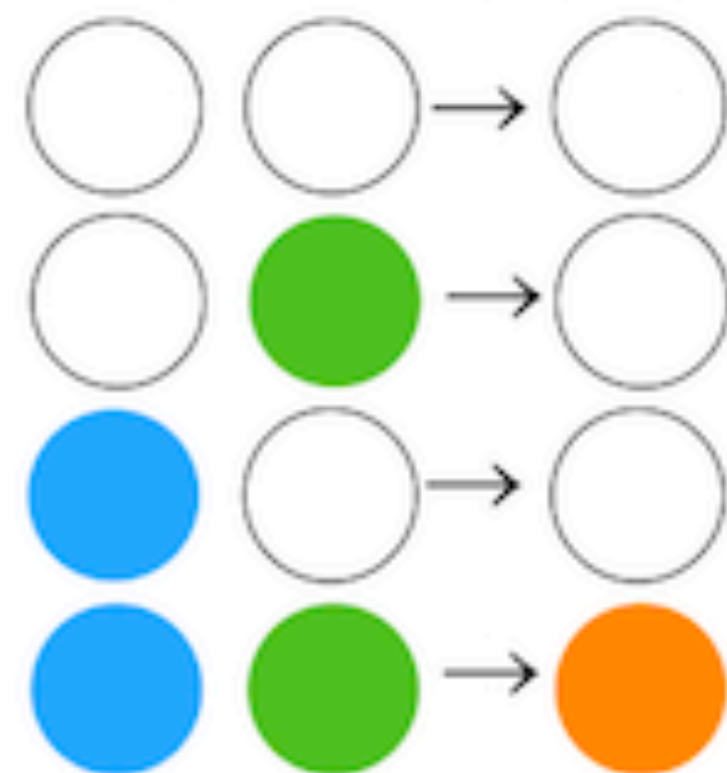
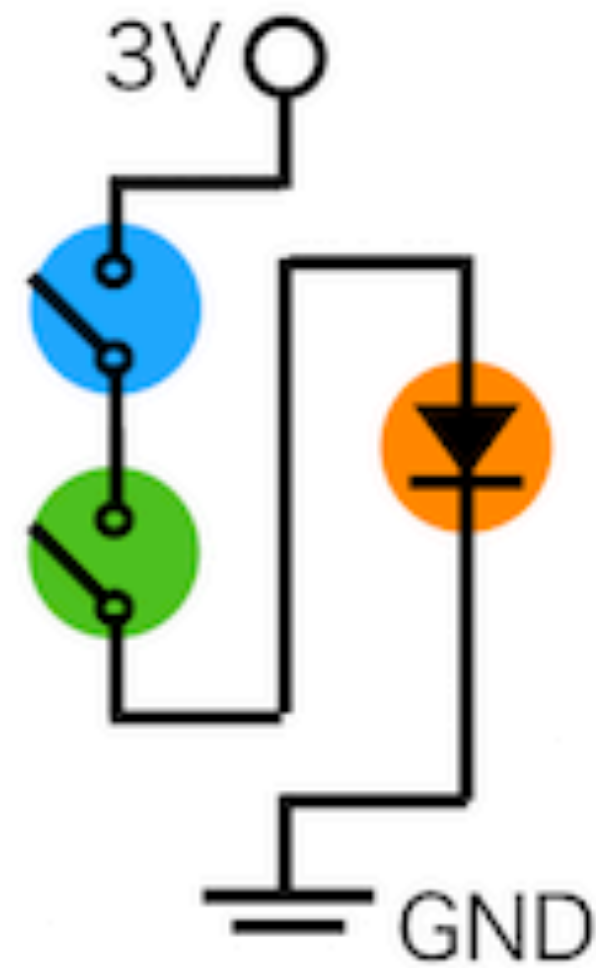


つまり、AND！

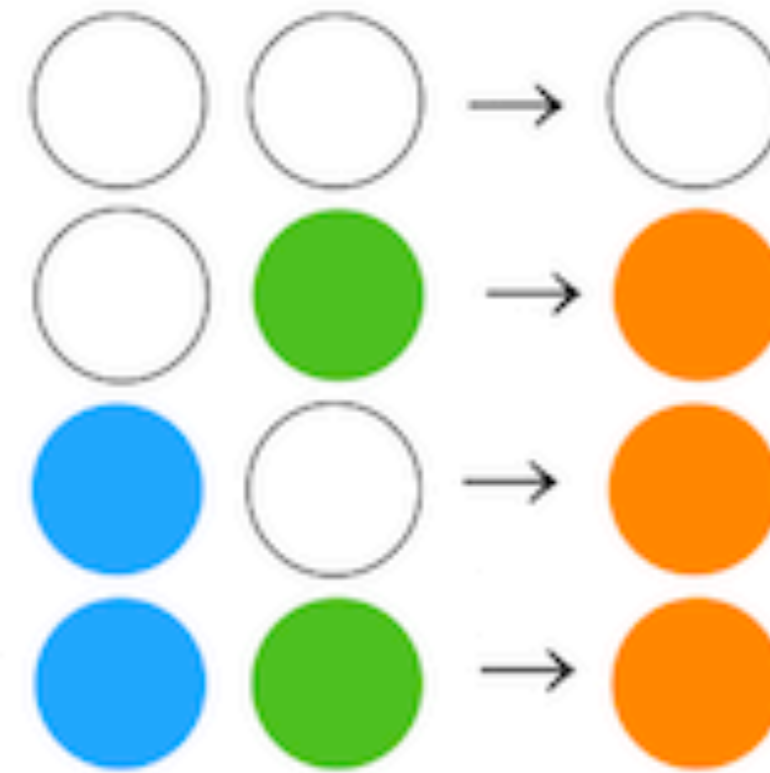
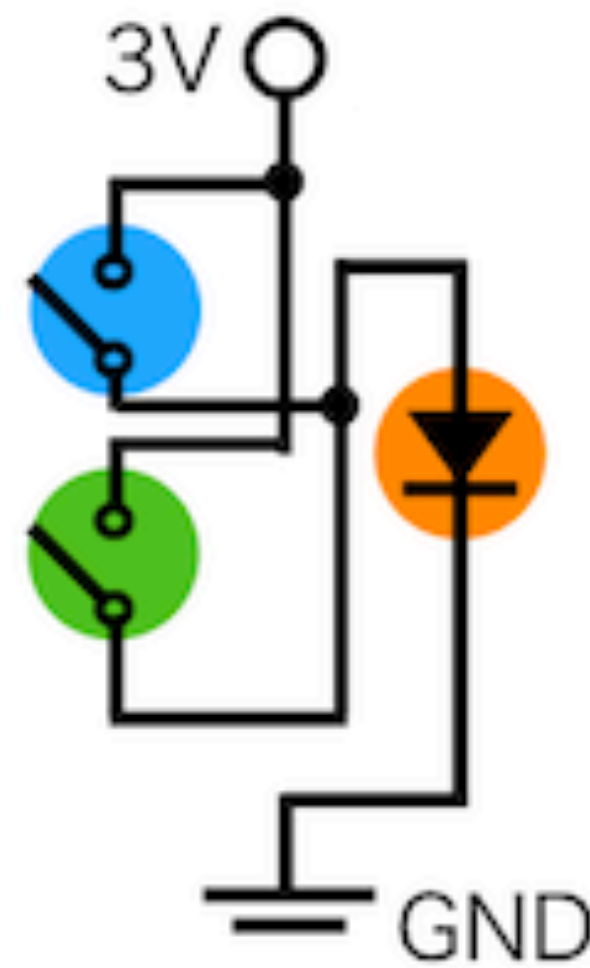


# 電気回路で論理式ができる！

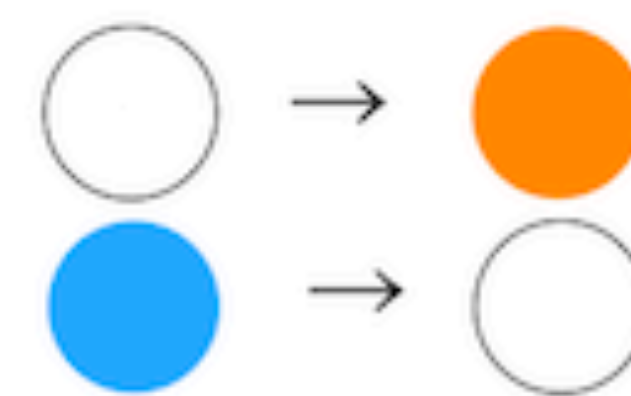
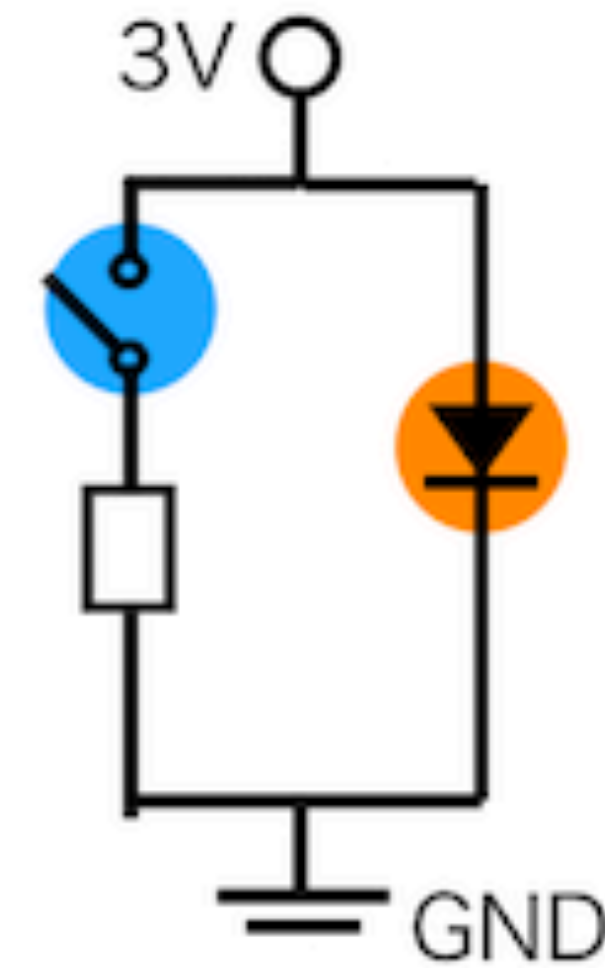
論理積(AND)



論理和(OR)

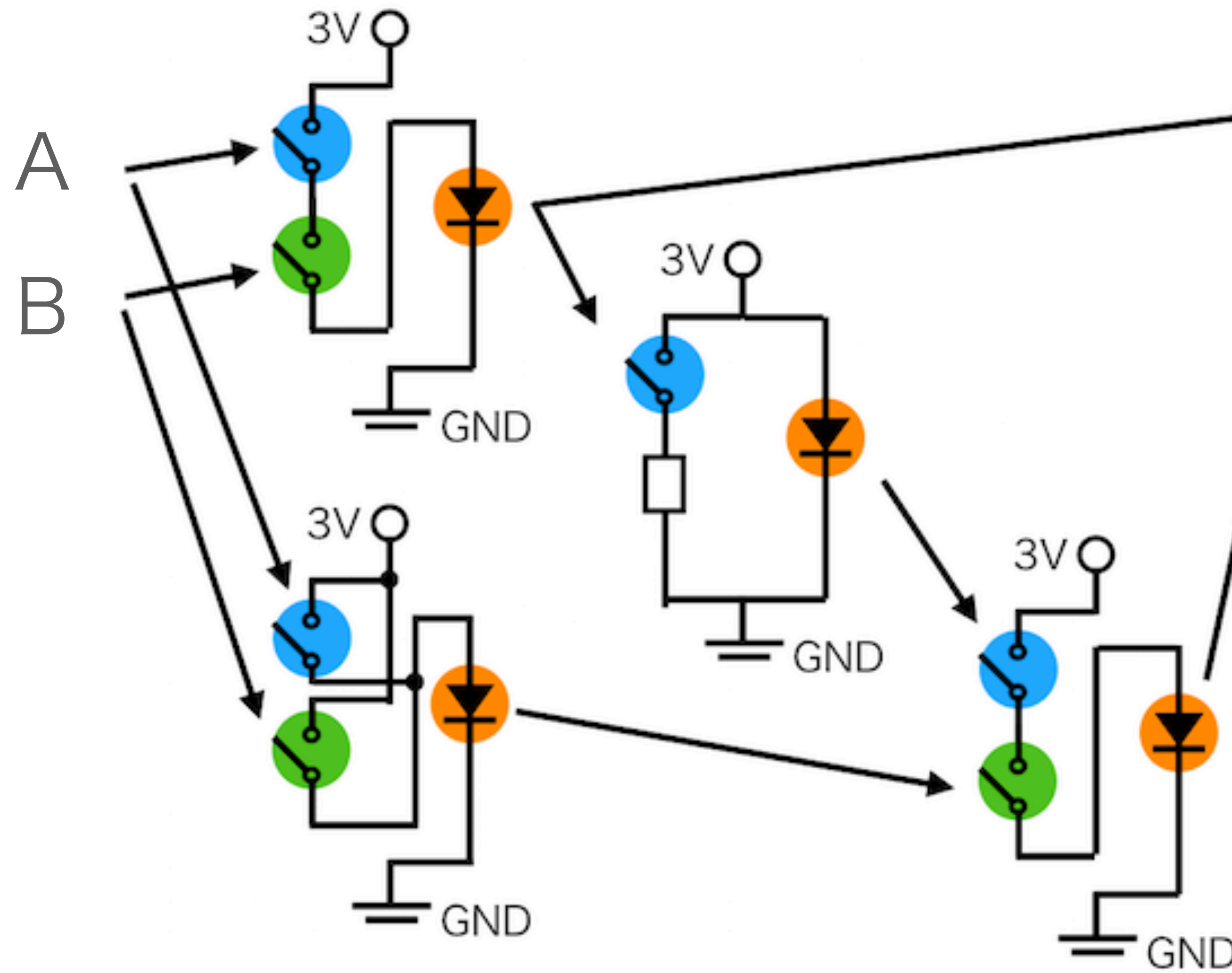


論理否定(NOT)





# 電気回路で2進法の足し算



$$C = A \text{ AND } B$$

※AとBが両方1の時だけCは1になる

$$D = (A \text{ OR } B) \text{ AND } (\text{NOT } C)$$

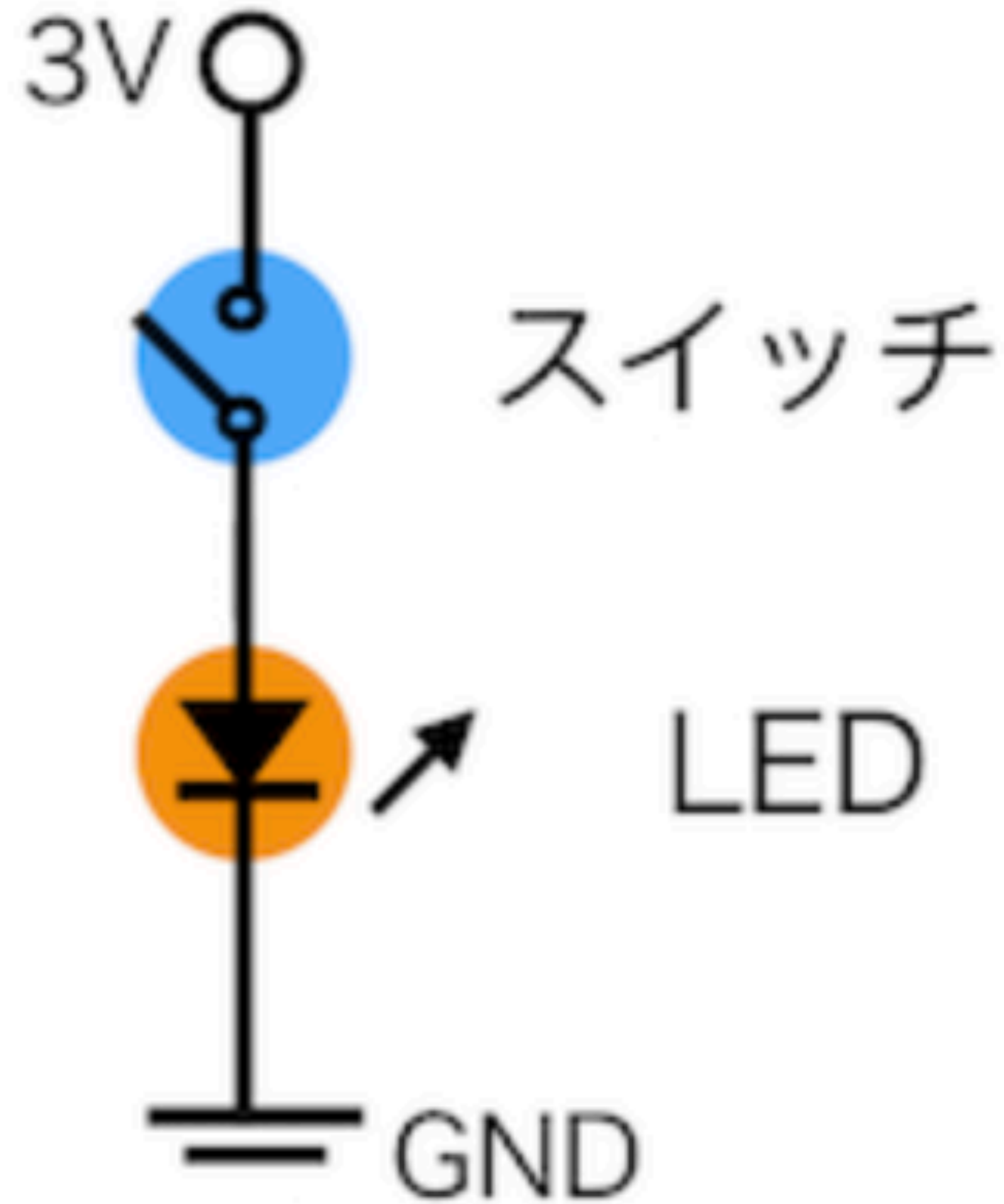
※AとBがどちらから1で、  
かつ、Cが1でない時にDは1になる

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



# 電気で動くスイッチ＝トランジスタ

トランジスタ→



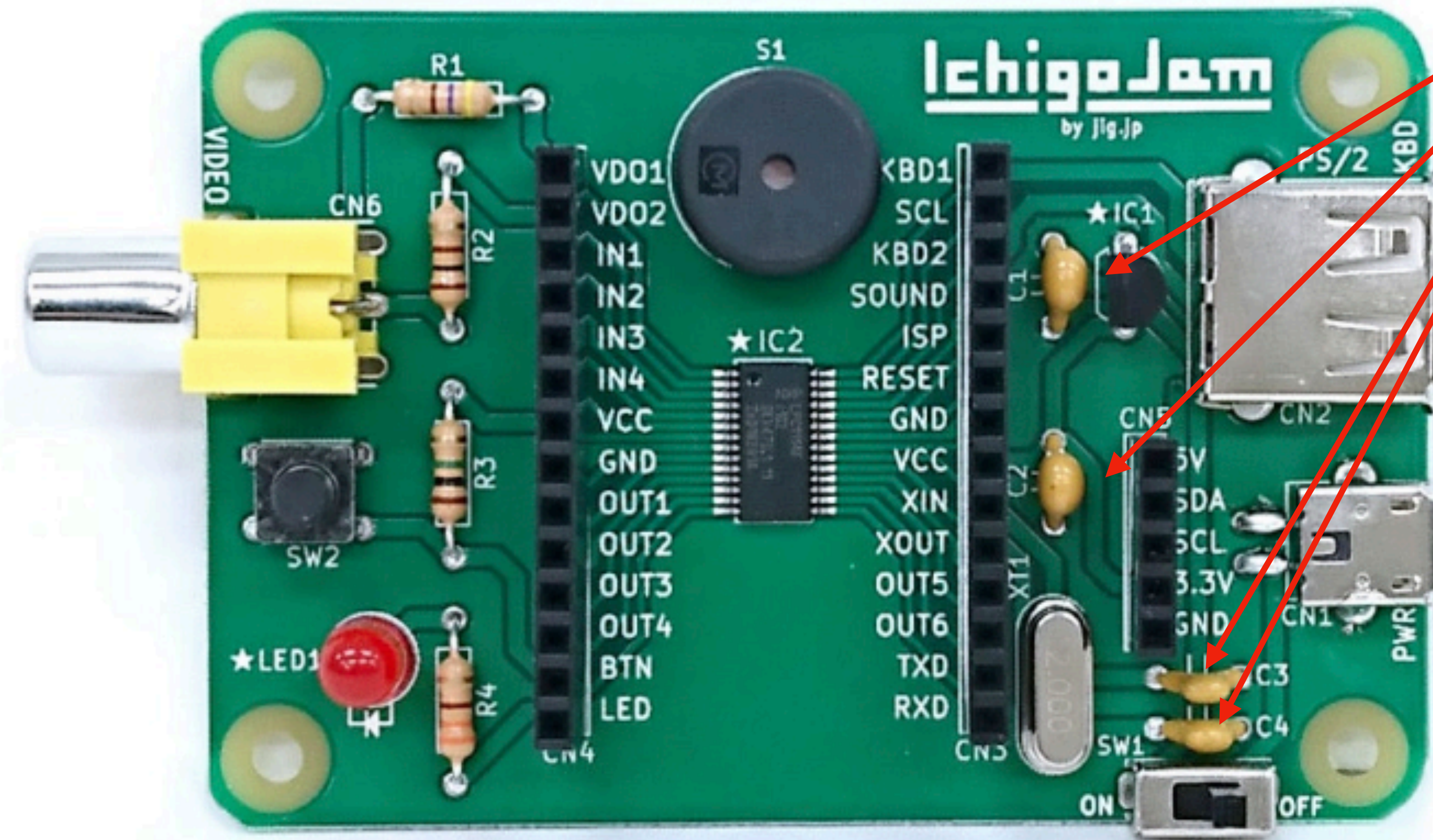
**電気をいれた瞬間、計算完了！**

**限界は？**

**限界は？ → 回路の長さ / 光の速さ**

計算はできた、記憶は？





電気を貯める  
電子部品  
コンデンサ

電気が溜まっている  
=1

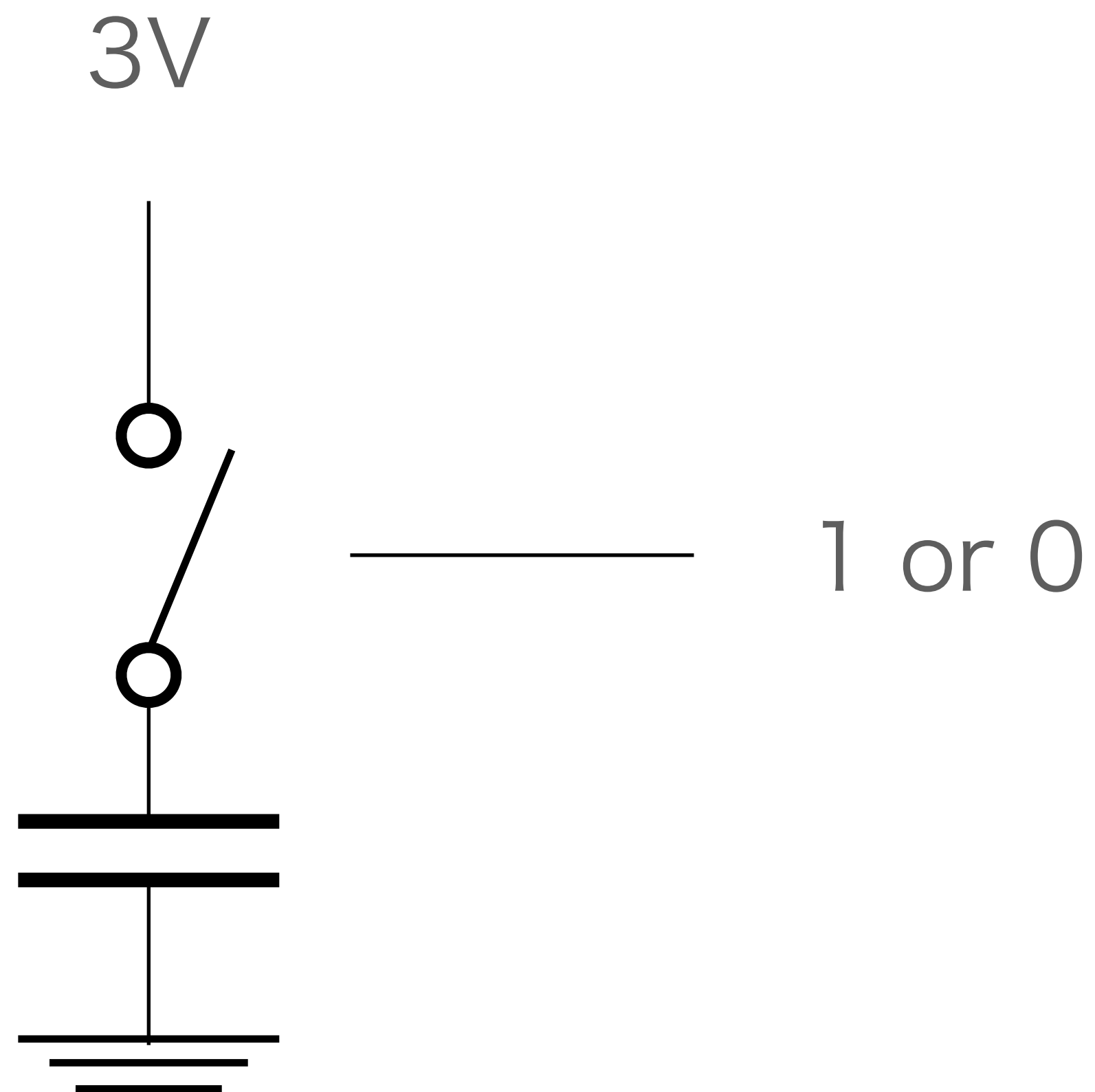
電気が溜まっていない  
=0



# 記憶への書き込み (1bit)

トランジスタ→

コンデンサ→

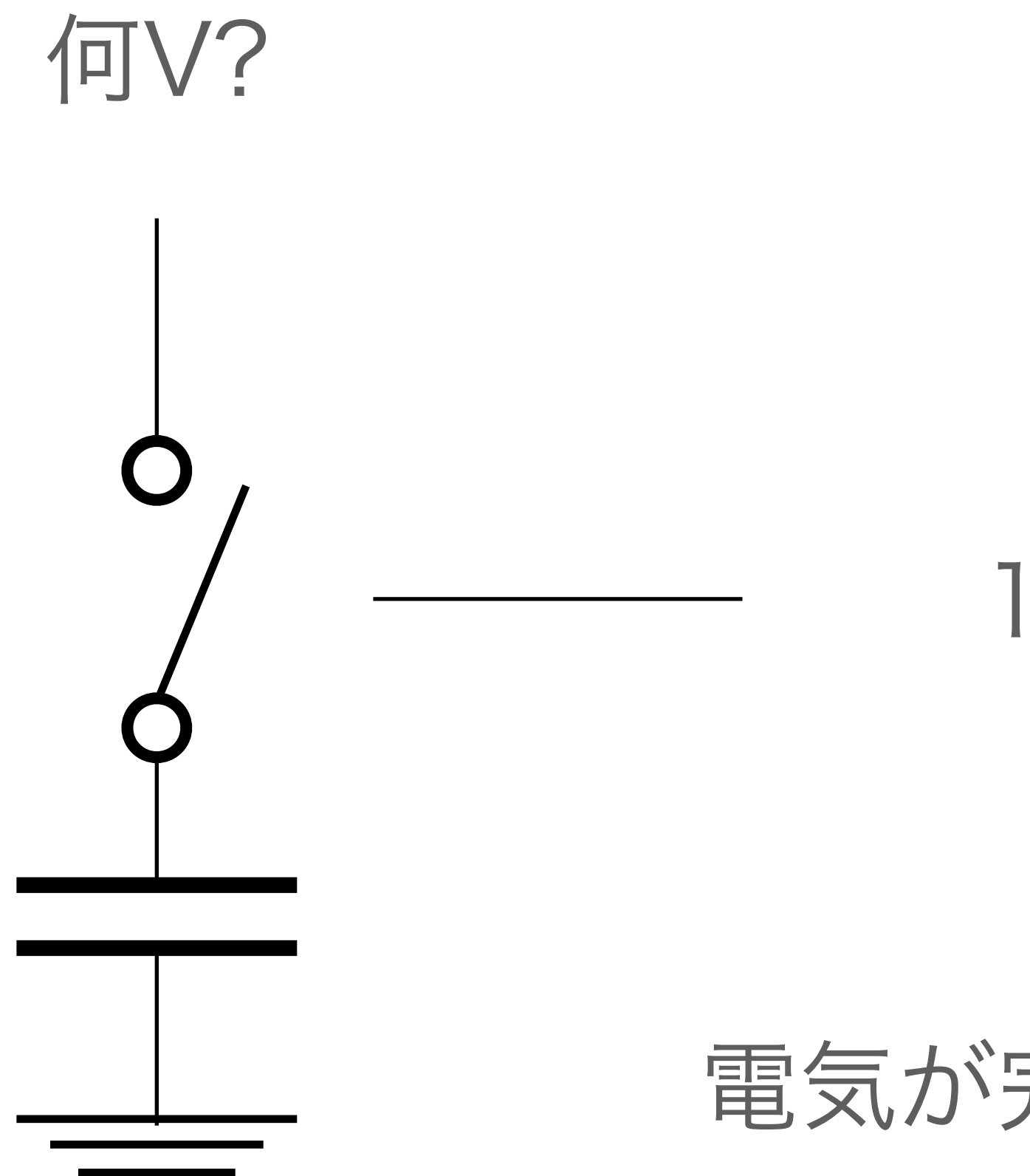


電気が貯まるまで待ちが必要  
(速度限界)

# 記憶から読み込み (1bit)

トランジスタ→

コンデンサ→



電気が完全に抜けないように  
定期的に上書き必要  
※CDやHDDなどは別の手法

# 8bit を 1byte と呼ぶ

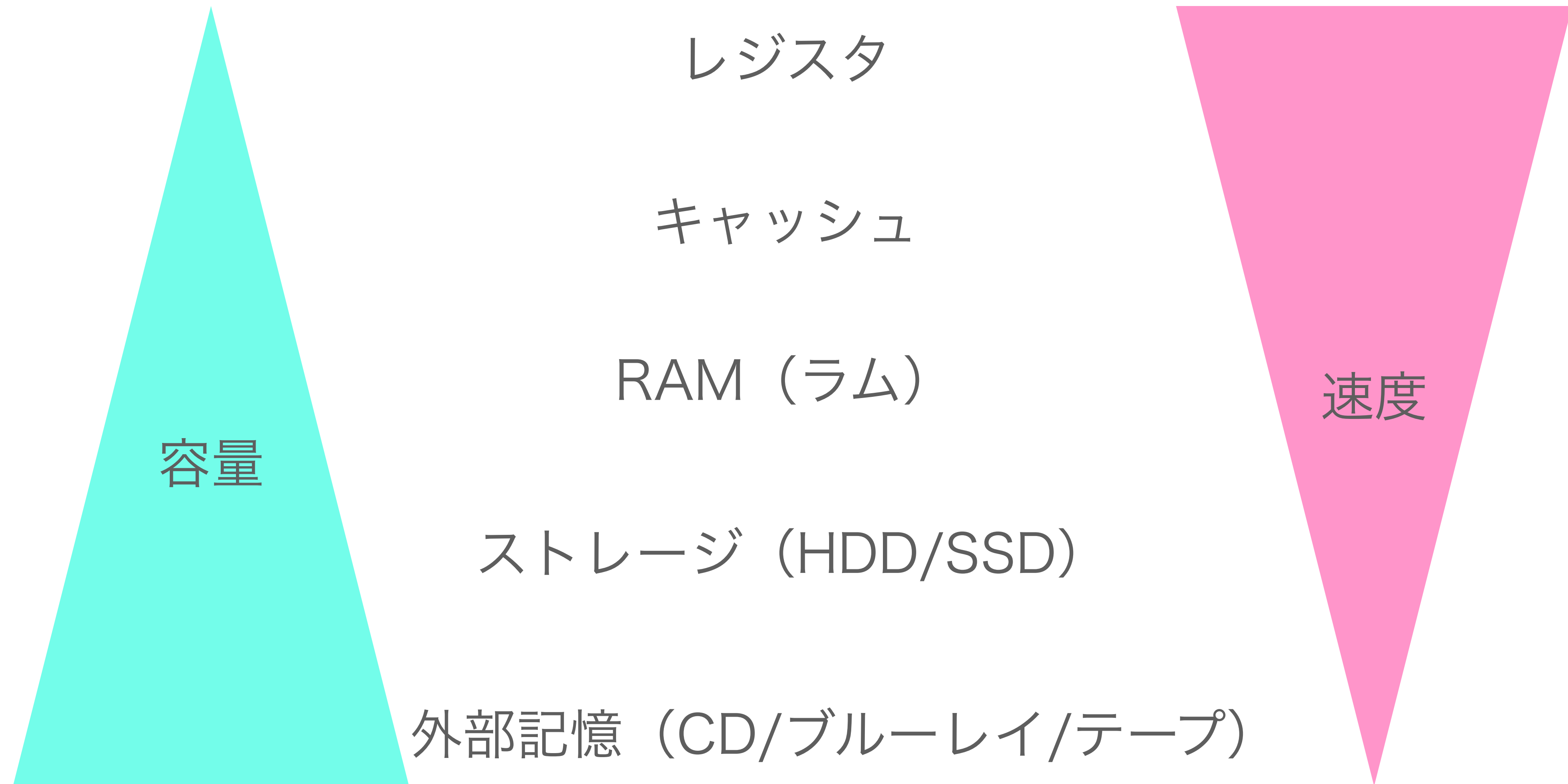
0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

## 90 = Zのキャラクターコード

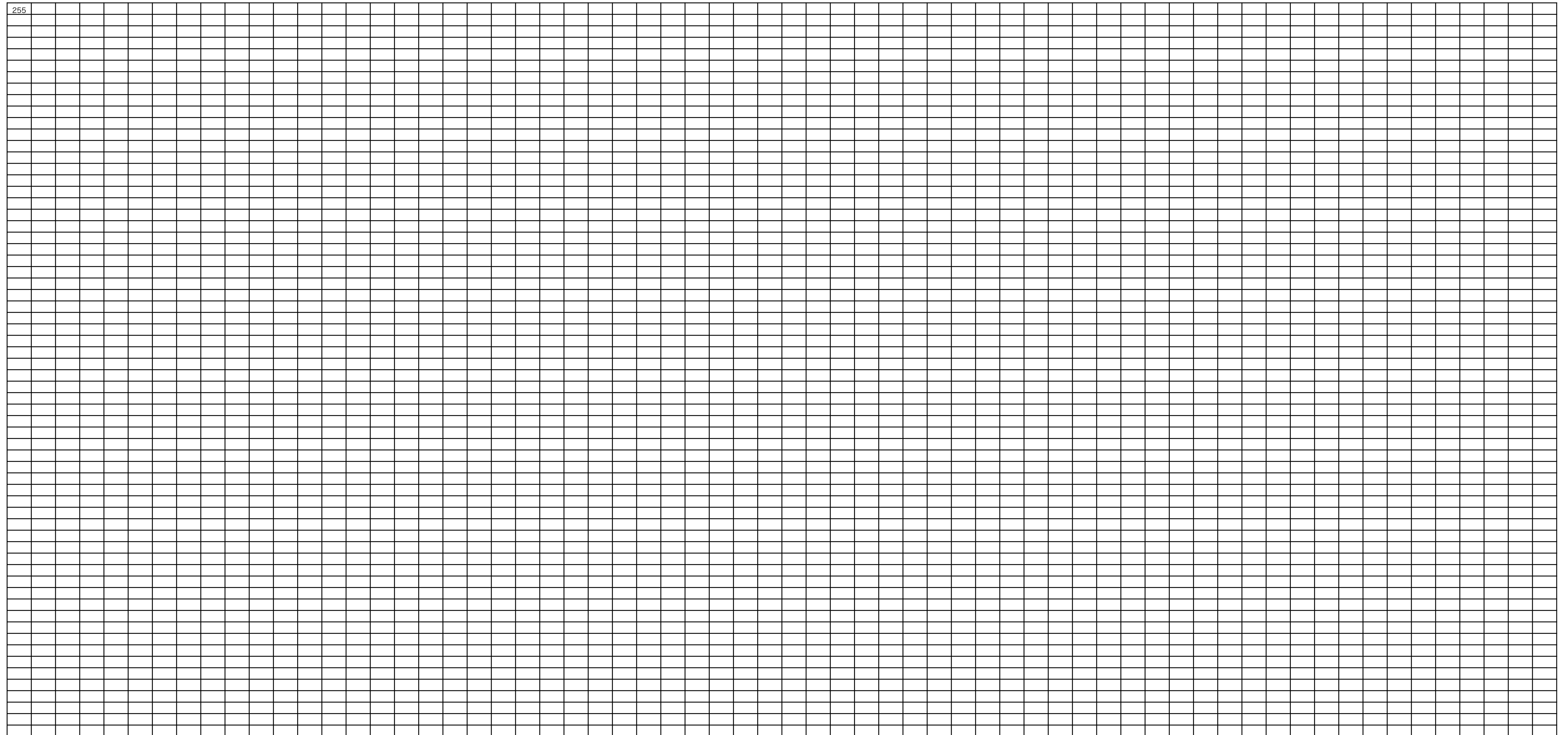
1024byte で、1kbyte (1キロバイト) 約1000バイト  
1024kbyte で、1Mbyte (1メガバイト) 約100万バイト  
1024Mbyte で、1Gbyte (1ギガバイト) 約10億バイト  
1024Gbyte で、1Tbyte (1テラバイト) 約1兆バイト



# コンピュータの記憶の種類



IchigoJam のメモリ、RAMは4KB=4096byte



# コンピューターのつくりかた

デジタルに強いコンピューターがほしい



ゲームも動画もデジタル (=数) にできる

すべての数は、0か1の二進法にできる

二進法は論理式で計算できる

論理式と記憶は電子回路で表せる

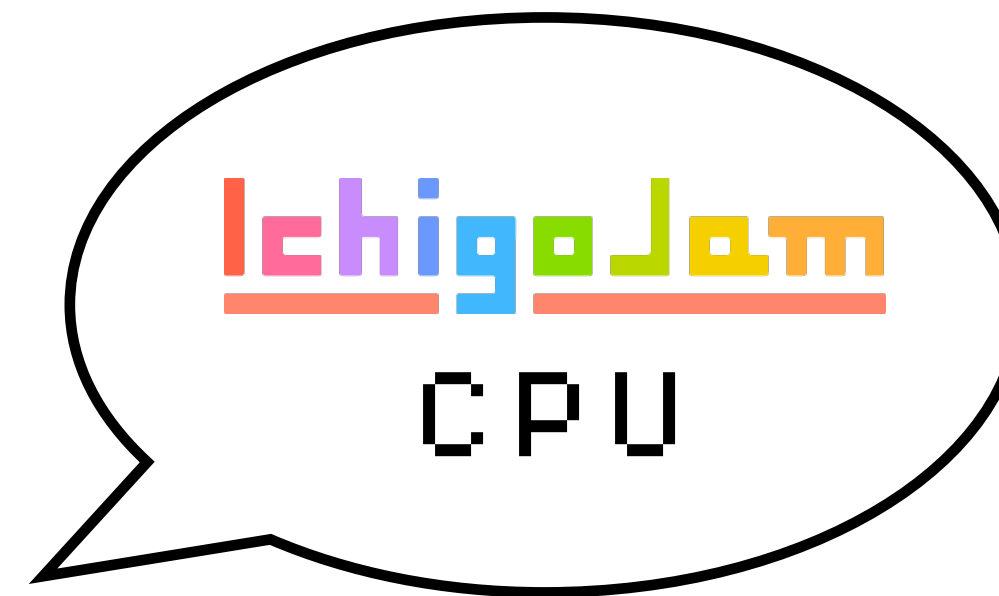
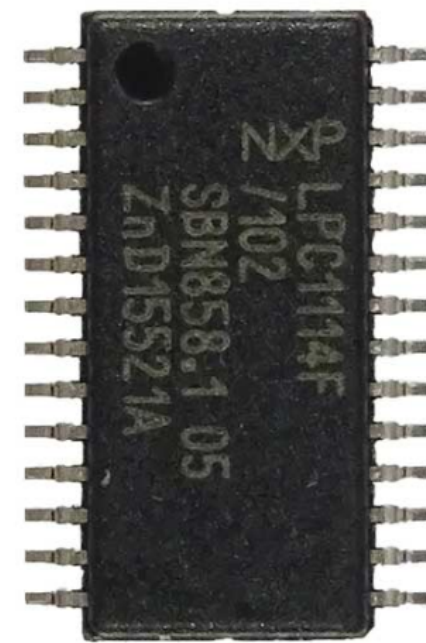
電子回路は小さくできる



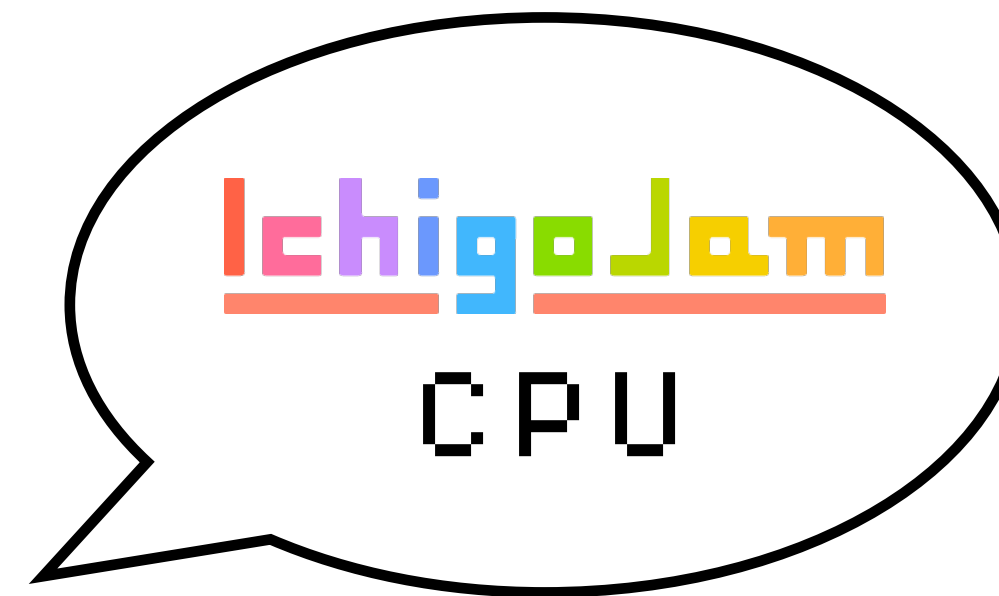
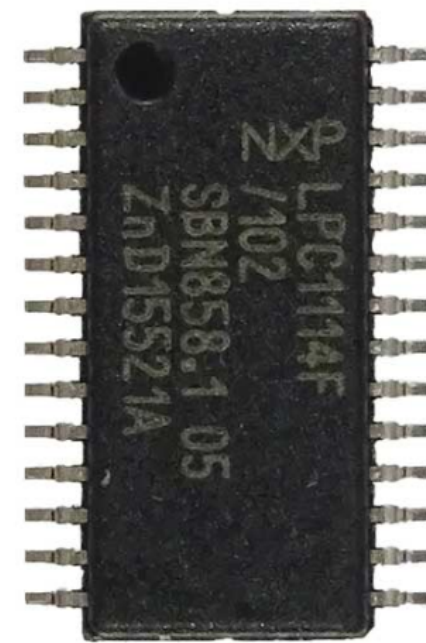
つまり、電子回路を詰めこめばコンピューターができる

IchigoJamのCPUには  
何個のスイッチが入っているでしょう？

IchigoJamのCPUには  
何個のスイッチが入っているでしょう？



# IchigoJamのCPUには 何個のスイッチが入っているでしょう？



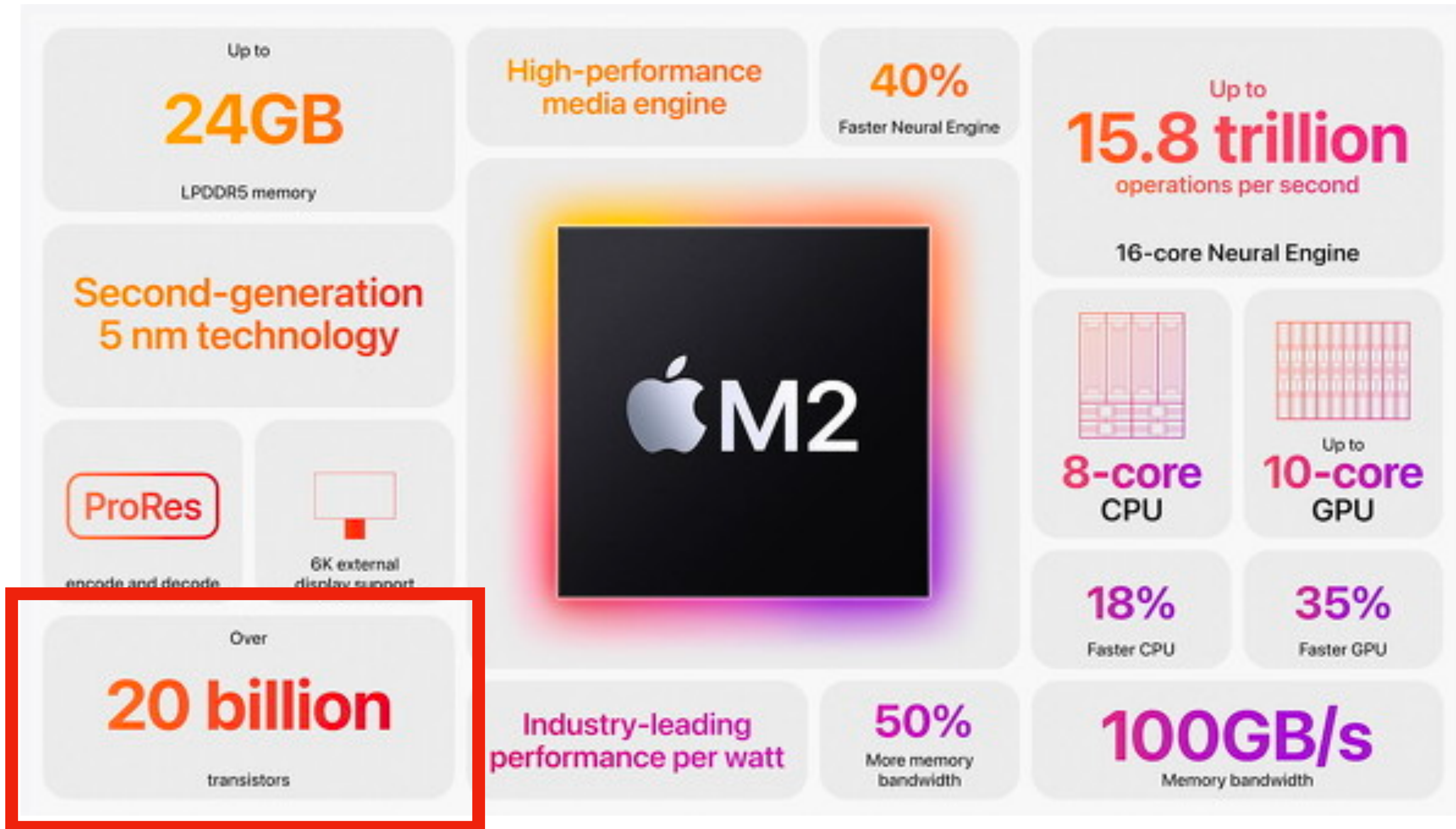
## 1万2千コ！

※Arm Cortex-M0 1.2万ゲート

<https://www.cqpub.co.jp/toragi/2008-2020/Portals/0/backnumber/2011/02/p055-060.pdf>



スマホにも収まるCPU、Apple M2 には、200億コのスイッチ！



# コンピューターのつくりかた

デジタルに強いコンピューターがほしい



ゲームも動画もデジタル (=数) にできる

すべての数は、0か1の二進法にできる

二進法は論理式で計算できる

論理式と記憶は電子回路で表せる

電子回路は小さくできる



つまり、電子回路を詰めこめばコンピューターができる



プログラミングとは？

プログラミング

=

やってほしいことを  
コンピューターが分かるレベルに  
分解したもの

# コンピューターが分かること 記憶と計算

次実行する命令の場所（数）を記憶するレジスタ

画面表示に相当する場所（数）

キーボードから入力されたキャラクターコード（数）を記憶する場所（数）

・・・など

いろんな数を1秒間に5000万回、計算し続けている  
(IchigoJamの場合)

**プログラミング言語**

**=**

**記憶と計算に翻訳してくれるもの  
(抽象的に書けて便利！)**