

D-7

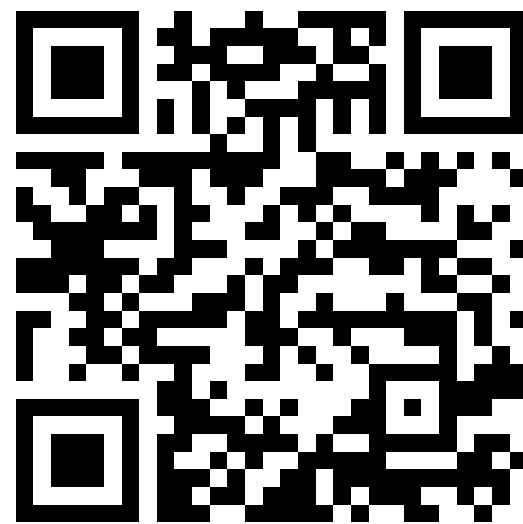
論理回路の設計を体験できる ブラウザ学習教材の開発と実践

名古屋市立緑高等学校

小林 裕司

タブレットやスマートフォンでも動作します。
よければ動かしながらご聴講ください⇒

Webページ動作



自己紹介

- 名前: 小林 裕司 (こばやし ゆうじ)
- 経歴
 - 2007/3 三重大学 工学部 情報工学科 卒業
 - 2009/3 三重大学大学院 工学研究科 情報工学専攻 修了
 - 2009/4 株式会社 日立製作所 公共システム事業部 入社
 - 2012/3 佛教大学 通信教育課程 教育学部 (数学免許)
 - 2015/3 佛教大学 科目履修コース (情報免許)
 - 2017/4 名古屋市立桜台高等学校
 - 2025/4 名古屋市立緑高等学校
- 高校時代の夢: ゲームクリエイター

SE

教員

良ければ、やってみてください

Webページ動作



https://nagoya-kobayashi.github.io/logic_circuit/

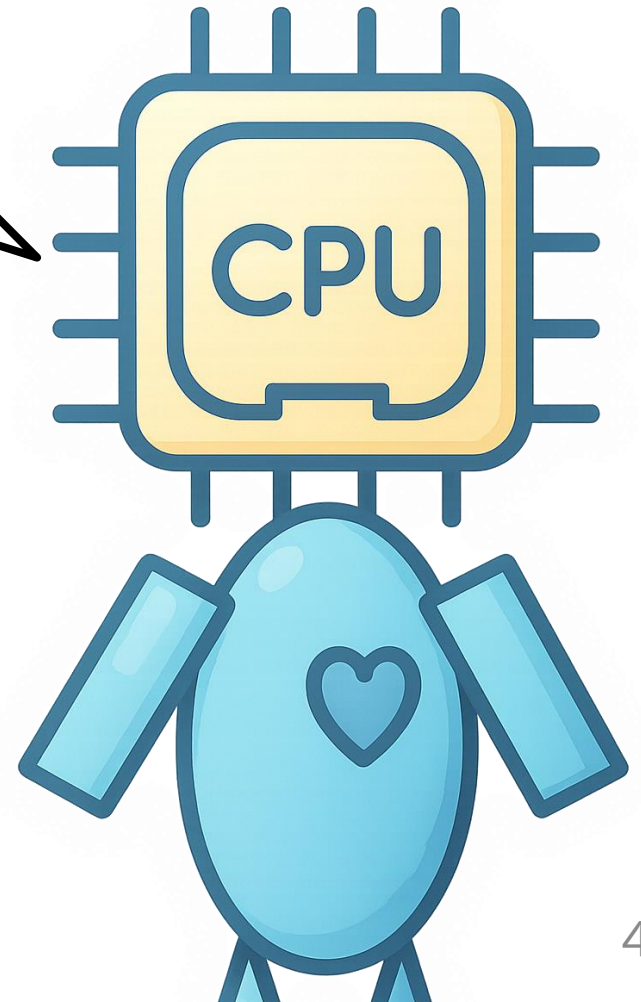
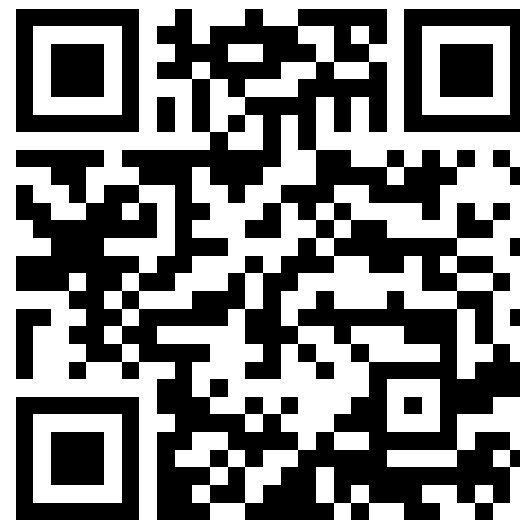
タブレットやスマートフォンでも動作します。
よければ動かしながらご聴講ください。

論理回路の設計を体験できる ブラウザ学習教材の開発と実践

- 背景・目的
- 授業の内容
- ブラウザ学習教材
- 授業実践の結果
- まとめ

よろしく
お願いいたします

Webページ動作

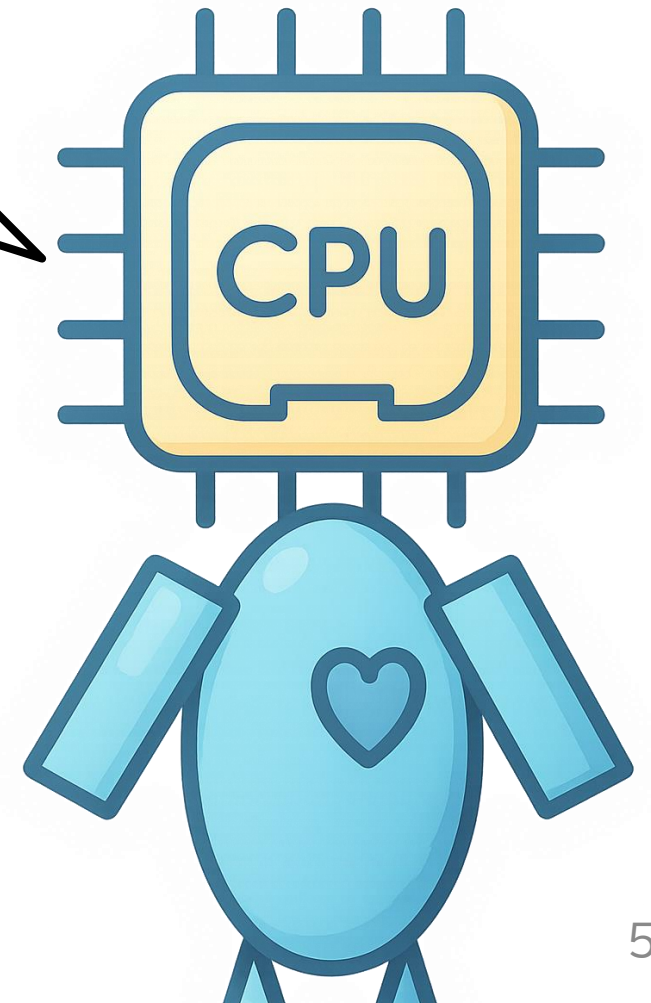
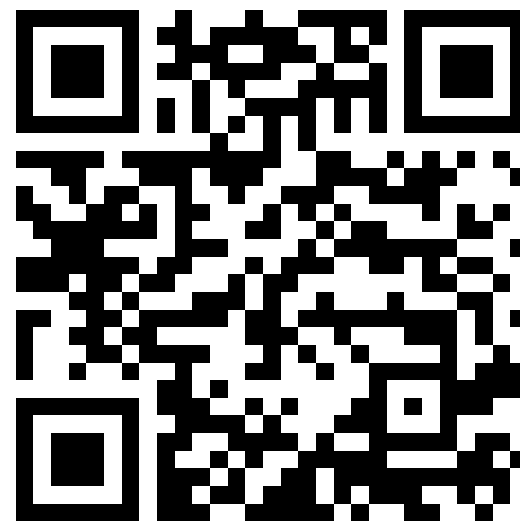


論理回路の設計を体験できる ブラウザ学習教材の開発と実践

- 背景・目的
- 授業の内容
- ブラウザ学習教材
- 授業実践の結果
- まとめ

よろしく
お願いいたします

Webページ動作



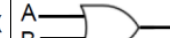
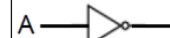
背景・目的

・入試でも扱われる「論理回路」の学習

問3 次の文章を読み、空欄 **カ** ～ **ク** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

基本的な論理回路には、論理積回路（AND 回路）、論理和回路（OR 回路）、否定回路（NOT 回路）の三つがあげられる。これらの図記号と真理値表は次の表1で示される。真理値表とは、入力と出力の関係を示した表である。

表1 図記号と真理値表

回路名	論理積回路	論理和回路	否定回路																																												
図記号																																															
真理値表	<table><tr><th colspan="2">入力</th><th>出力</th></tr><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	入力		出力	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th colspan="2">入力</th><th>出力</th></tr><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	入力		出力	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><th>入力</th><th>出力</th></tr><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	入力	出力	A	X	0	1	1	0
	入力		出力																																												
	A	B	X																																												
	0	0	0																																												
	0	1	0																																												
1	0	0																																													
1	1	1																																													
入力		出力																																													
A	B	X																																													
0	0	0																																													
0	1	1																																													
1	0	1																																													
1	1	1																																													
入力	出力																																														
A	X																																														
0	1																																														
1	0																																														

(1) S航空会社が所有する旅客機の後方には、トイレが二つ（ $A \cdot B$ ）ある。トイレAとトイレBの両方が同時に使用中になると乗客の座席前にあるパネルのランプが点灯し、乗客にトイレが満室であることを知らせる。入力Aは、トイレAが使用中の場合には1、空いている場合には0とする。Bについても同様である。出力Xはランプが点灯する場合に1、点灯しない場合に0となる。これを実現する論理回路は次の図2である。



(2) S航空会社では新しい旅客機を購入することにした。この旅客機では、トイレを三つ（ $A \cdot B \cdot C$ ）に増やし、三つのうちどれか二つ以上が使用中になったら混雑を知らせるランプを点灯させる。入力や出力は(1)と同様とする。この場合の真理値表は **キ** で、これを実現する論理回路は図3である。

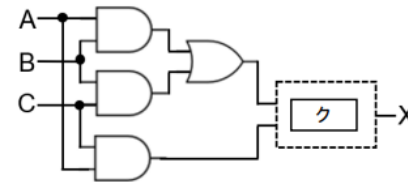
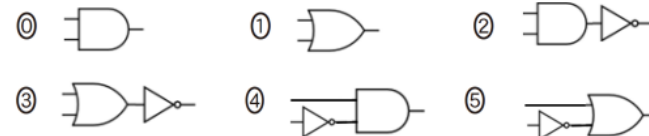


図3 (2)の論理回路

カ , **ク** の解答群



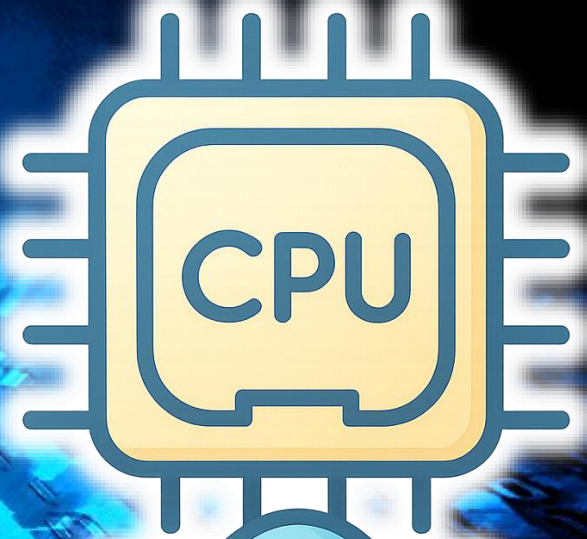
キ の解答群

0	1	2	3
入力	入力	入力	入力
A B C X	A B C X	A B C X	A B C X
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 1 0	0 0 1 1	0 0 1 0	0 0 1 1
0 1 0 0	0 1 0 1	0 1 0 0	0 1 0 1
0 1 1 0	0 1 1 1	0 1 1 1	0 1 1 1
1 0 0 0	1 0 0 1	1 0 0 0	1 0 0 1
1 0 1 0	1 0 1 0	1 0 1 1	1 0 1 1
1 1 0 0	1 1 0 0	1 1 0 1	1 1 0 1
1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1

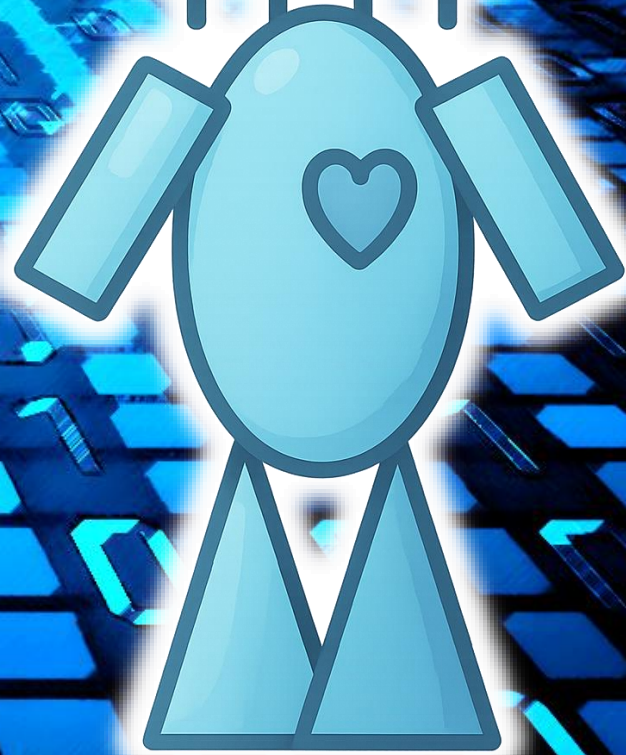
背景・目的

- 入試でも扱われる「論理回路」の学習
 - コンピュータの仕組みとして理解を深める
 - 実用的応用力や設計思考の醸成
 - 授業時間が足りなく、最低限の入試対策
 - 真理値表の読み方、回路上の信号をたどる演習などで終わってしまいがち
- ⇒ **ブラウザ型の体験教材**を開発し、
短時間で深い理解を得られる授業を目指す！

談話



CPUくんと 友達になろう



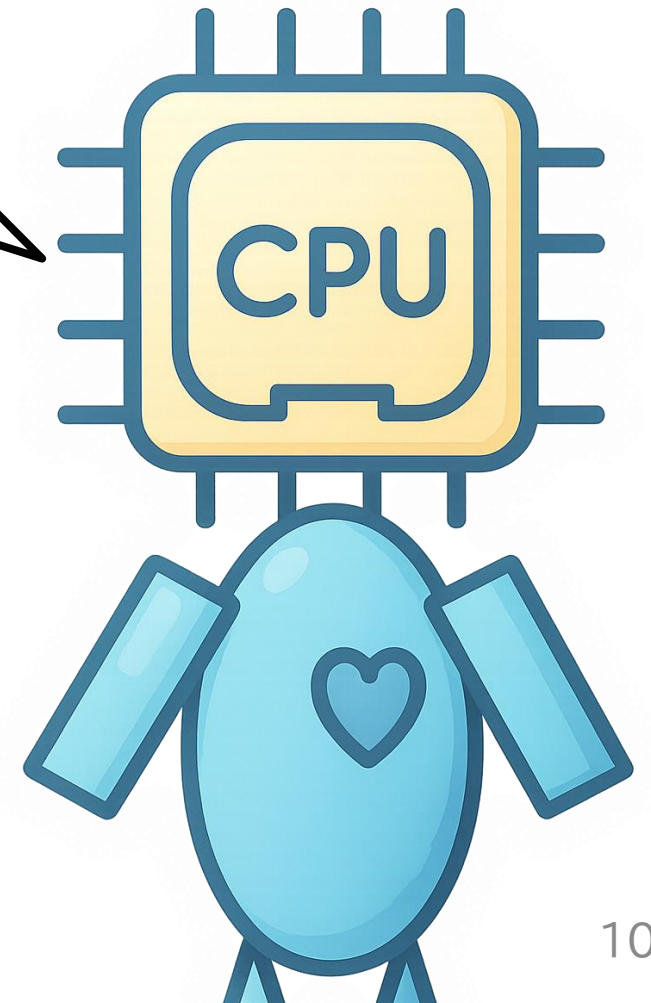
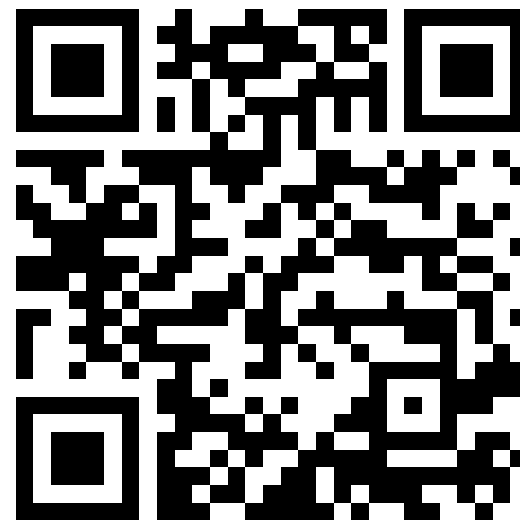
という授業を
やってみました。

論理回路の設計を体験できる ブラウザ学習教材の開発と実践

- 背景・目的
- 授業の内容
- ブラウザ学習教材
- 授業実践の結果
- まとめ

よろしく
お願いいたします

Webページ動作



論理回路

- 電子回路とは？
- CPUの正体



授業計画

時間	内容
0- 5分	二進数筆算の復習
5 - 10分	ステップ1: チュートリアル
10 - 17分	ステップ2: 半加算機設計【Level 1】
17 - 20分	ステップ3: チュートリアル(XOR)
20 - 27分	ステップ4: 半加算機設計(XOR有)【Level 1】
27 - 34分	ステップ5: 全加算機設計【Level 2】
34 - 41分	ステップ6: 2ビット加算回路実装【Level 3】
41 - 48分	ステップ7: 4ビット加算回路実装【Level Max】
49 - 50分	まとめ & 振り返り

2進数の計算

$$10001001_{(2)} + 01101000_{(2)}$$

	1	0	0	0	1	0	0	1
+	0	1	1	0	1	0	0	0
	1	1	1	1	0	0	0	1

Carry: 1

2進数の計算

$$10001001_{(2)} + 01101000_{(2)}$$

	128	64	32	16	8	4	2	1
	1	0	0	0	1	0	0	1
+	0	1	1	0	1	0	0	0
	1	1	1	1	0	0	0	1

$$137_{(10)}$$

+

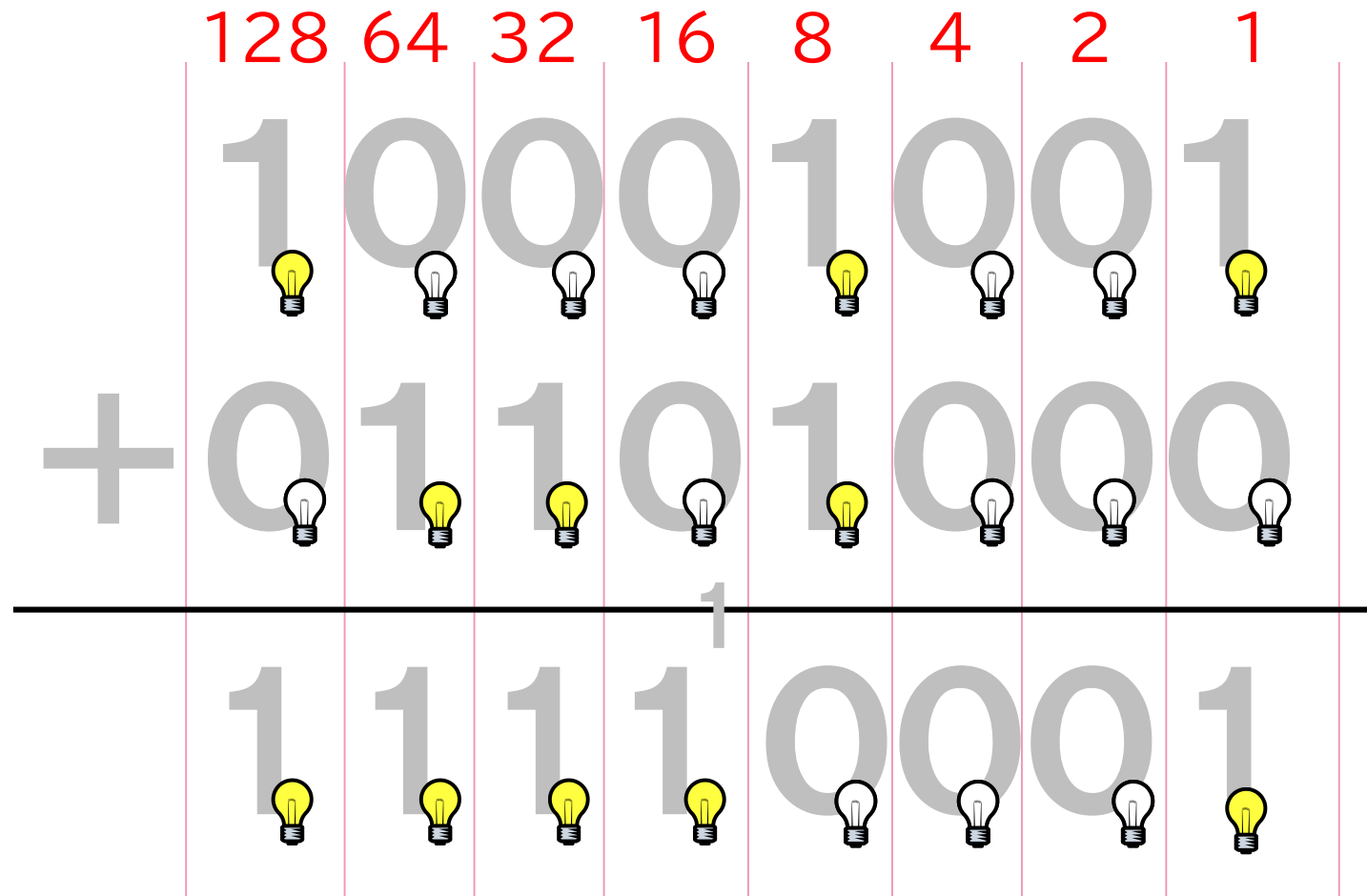
$$104_{(10)}$$

=

$$241_{(10)}$$

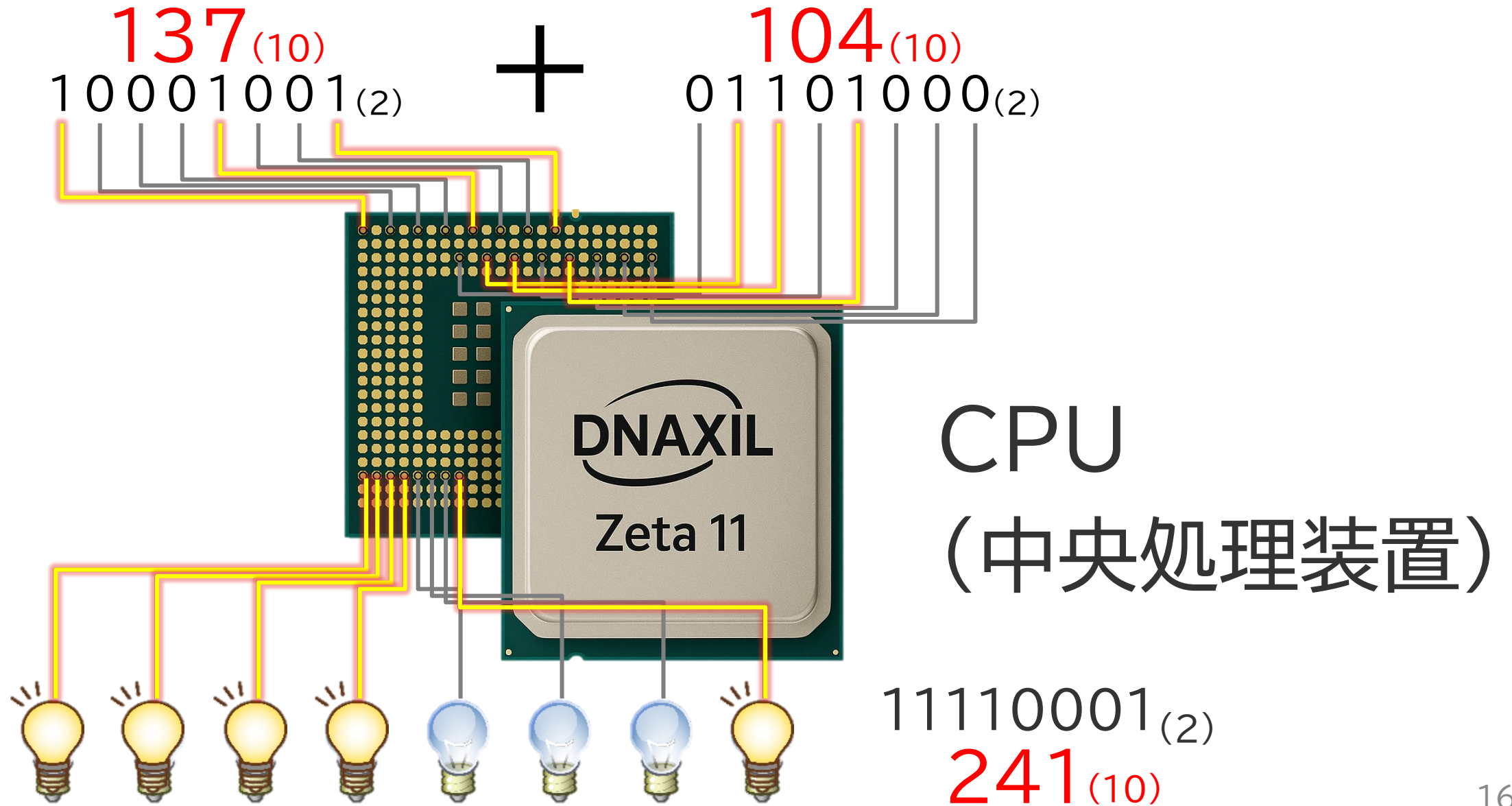
2進数の計算

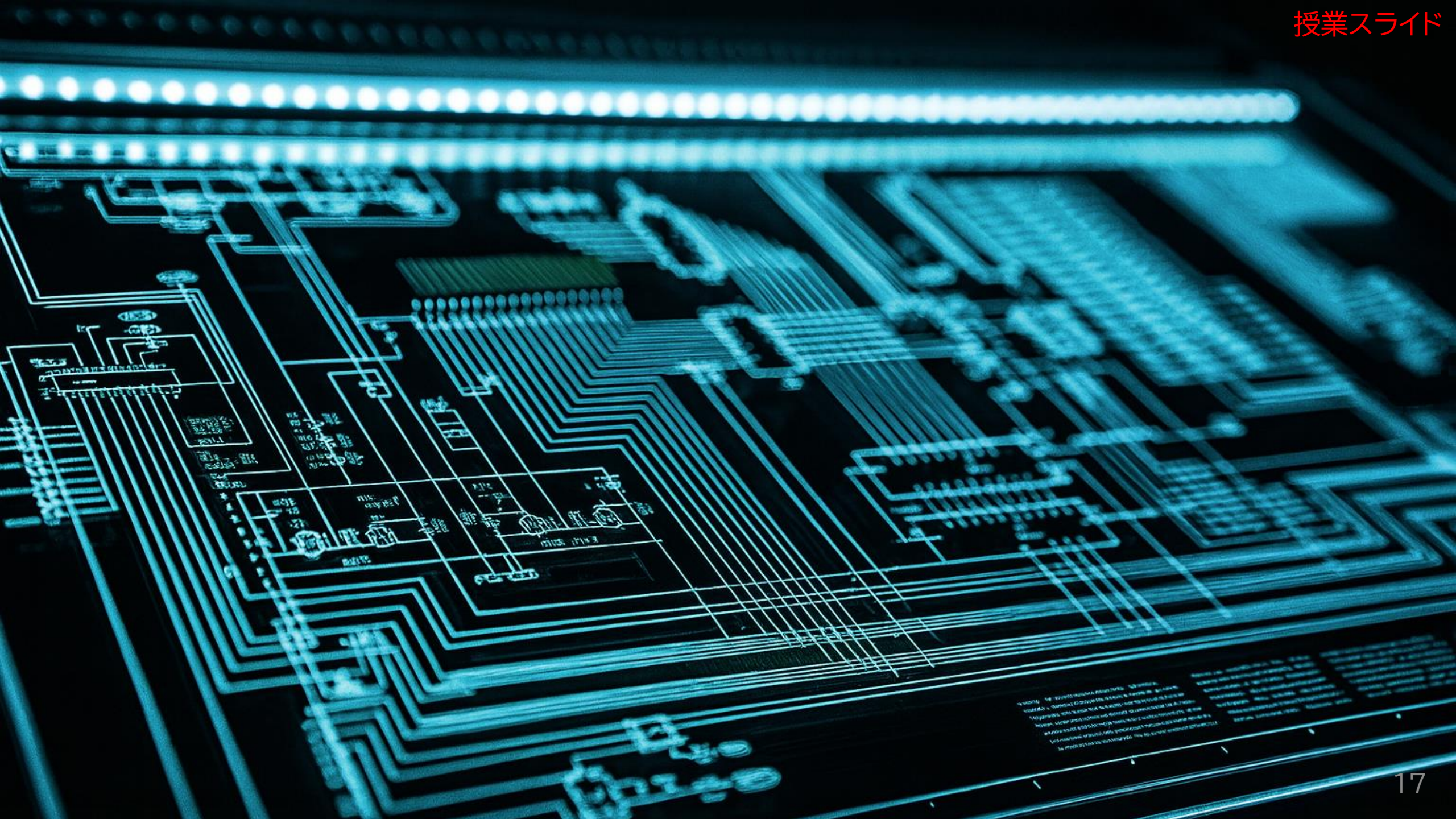
$$10001001_{(2)} + 01101000_{(2)}$$

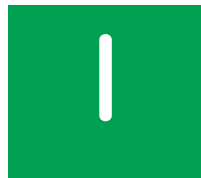


$$\begin{aligned} &137_{(10)} \\ &+ \\ &104_{(10)} \\ &= \\ &241_{(10)} \end{aligned}$$

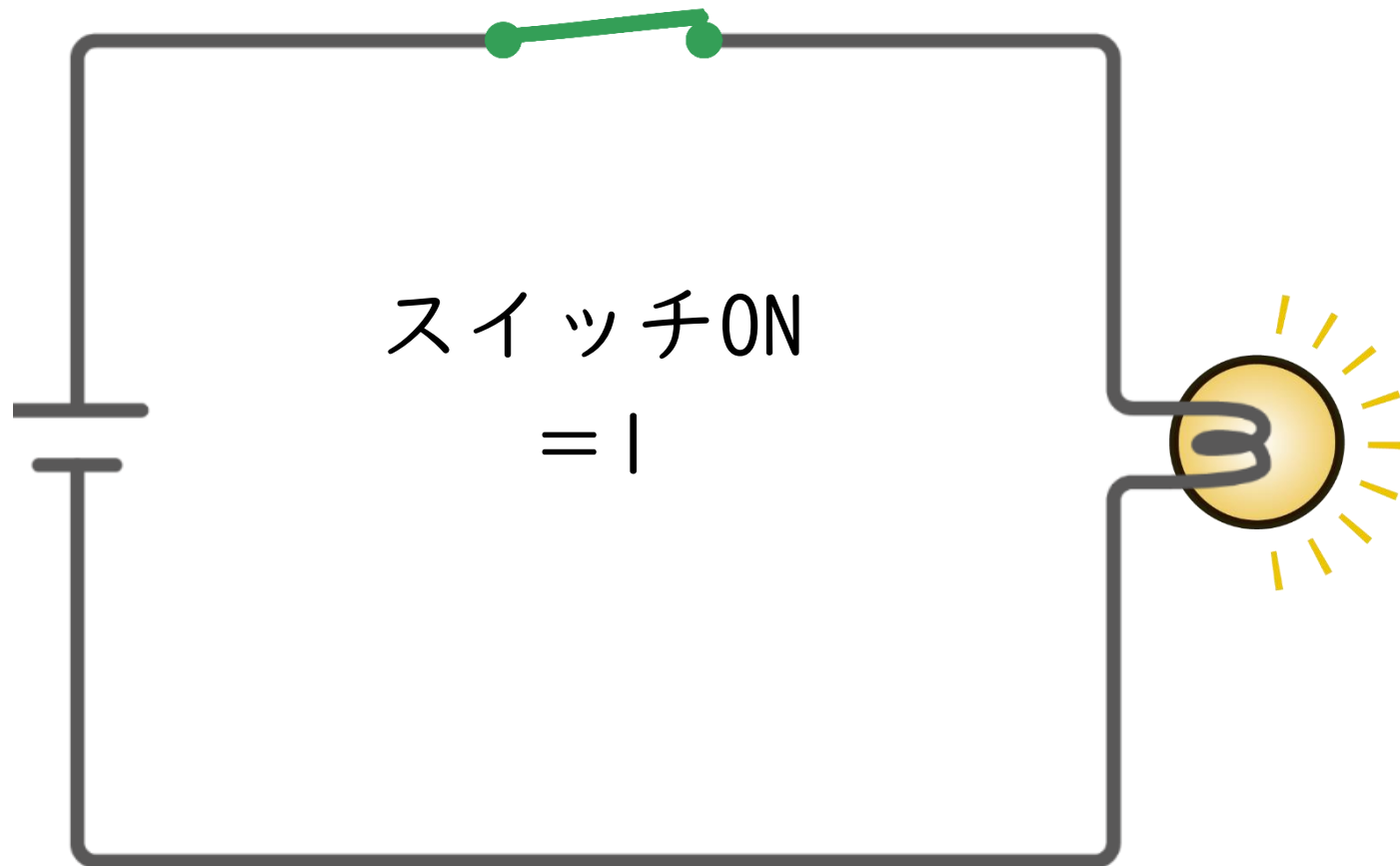
2進数の計算







基本論理回路

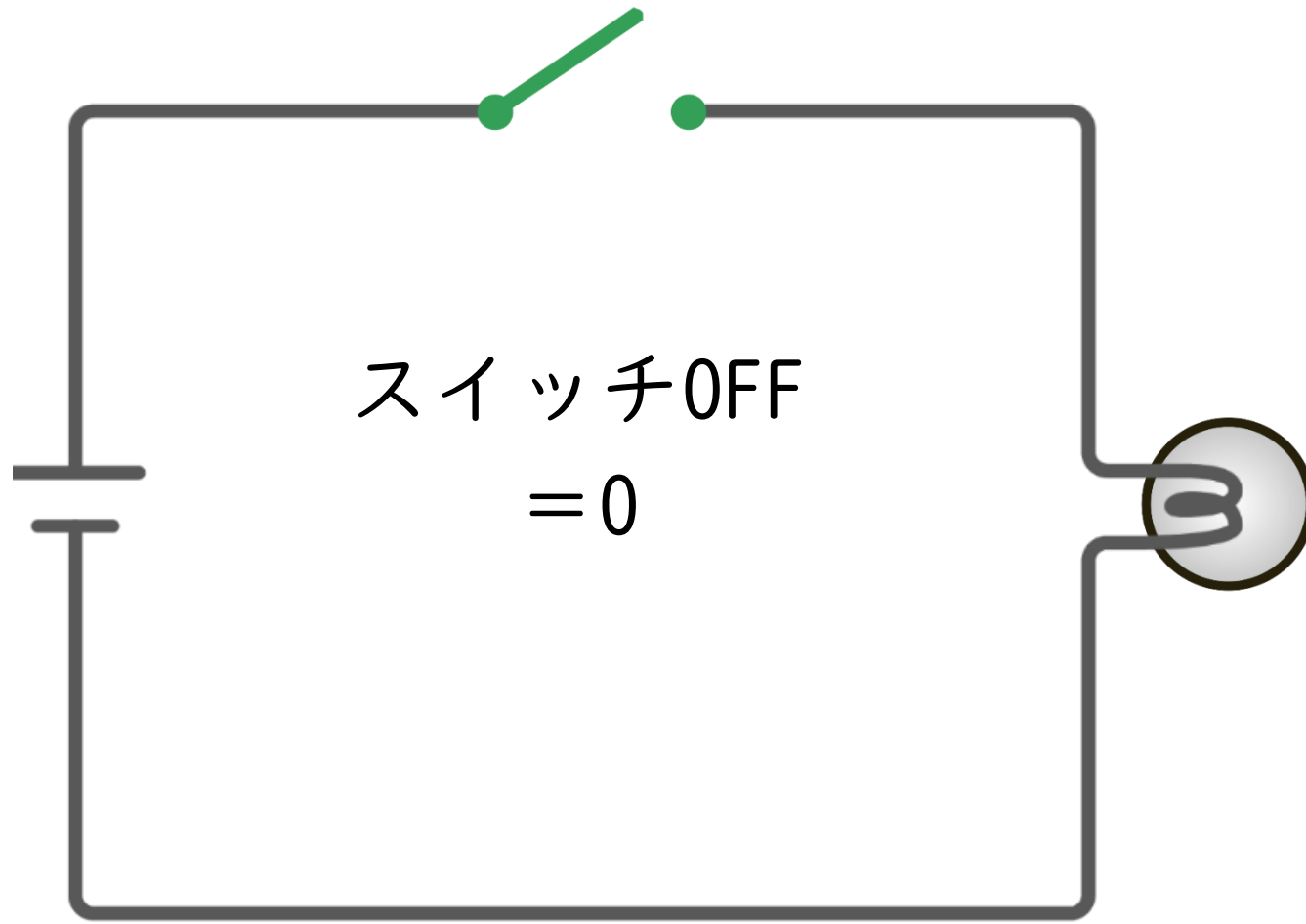


スイッチがON
なら1, OFFな
ら0と考えられ
ます





基本論理回路



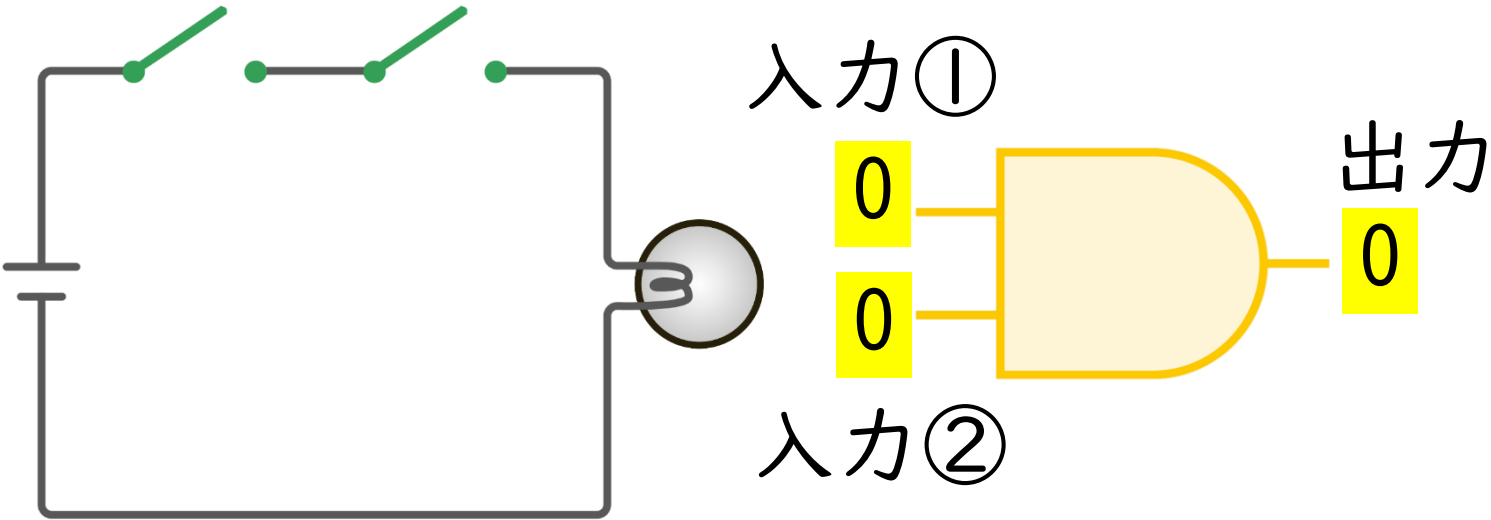
スイッチがON
なら1, OFFな
ら0と考えられ
ます





基本論理回路

■AND回路 入力が全て1のときだけ出力が1となる回路。

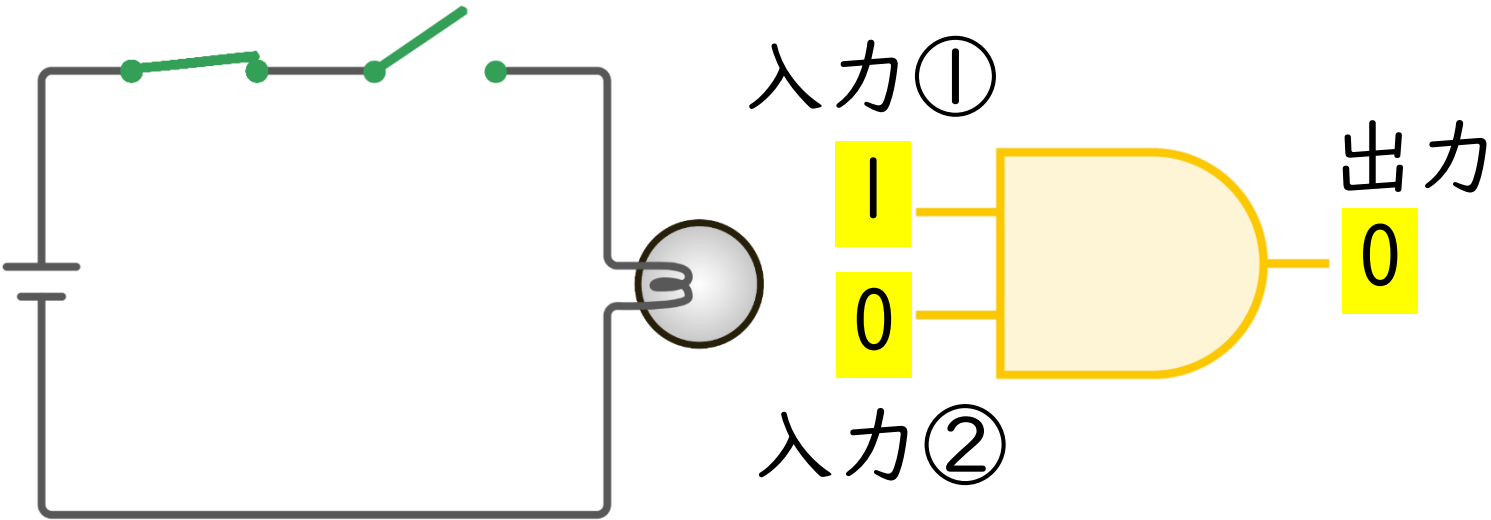


①	②	出力
0	0	0
1	0	0
0	1	0
1	1	1



基本論理回路

■AND回路 入力が全て1のときだけ出力が1となる回路。

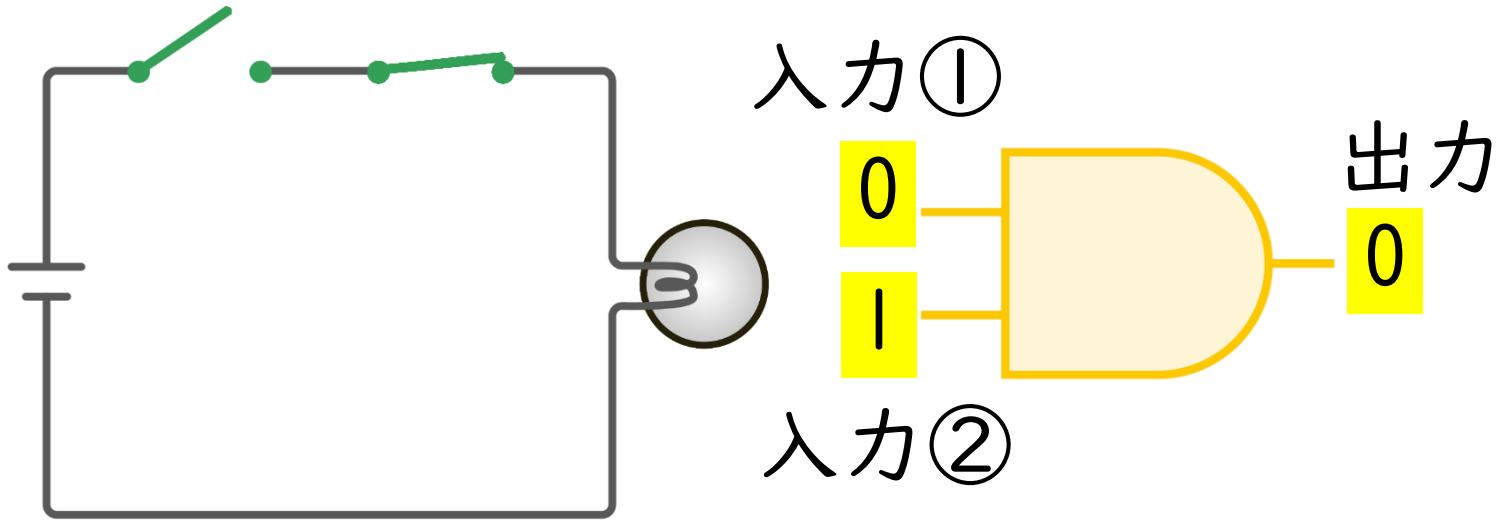


①	②	出力
0	0	0
1	0	0
0	1	0
1	1	1



基本論理回路

■AND回路 入力が全て1のときだけ出力が1となる回路。

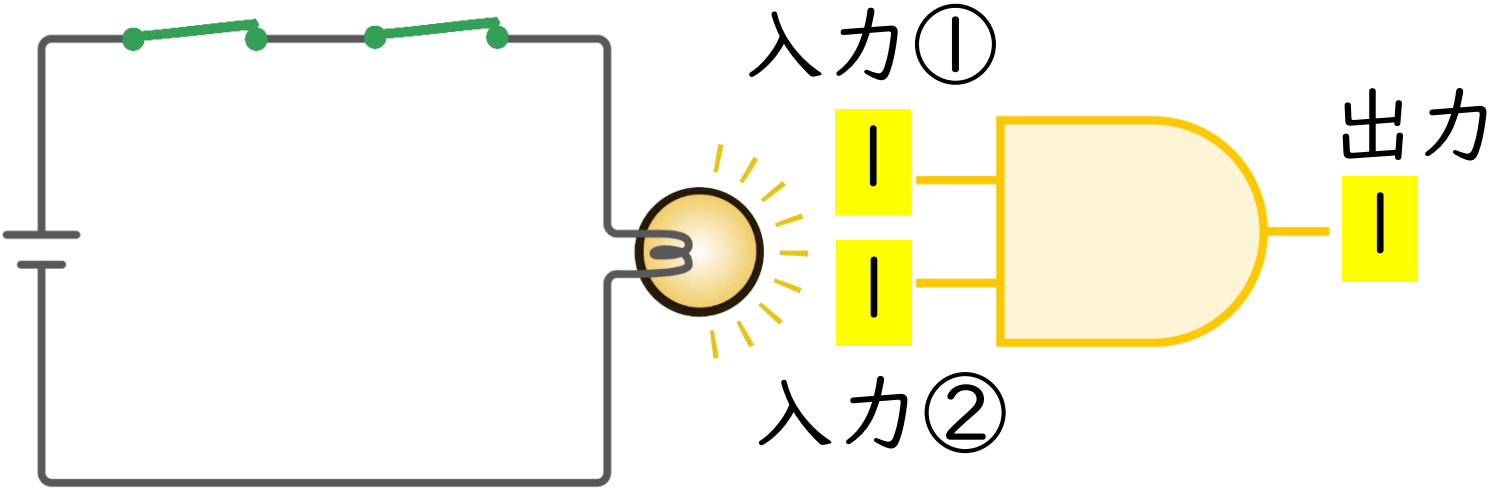


①	②	出力
0	0	0
1	0	0
0	1	0
1	1	1



基本論理回路

■AND回路 入力が全て1のときだけ出力が1となる回路。

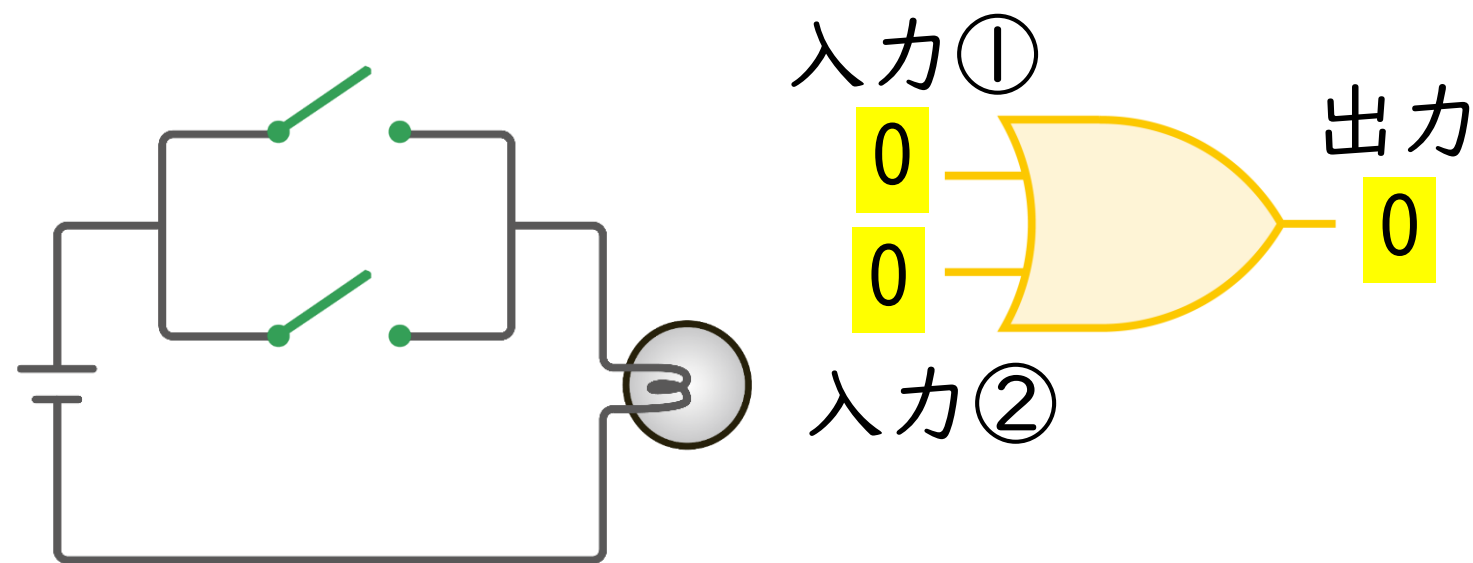


①	②	出力
0	0	0
1	0	0
0	1	0
1	1	1



基本論理回路

■OR回路 入力がどれか1つでも1なら出力が1になる回路。

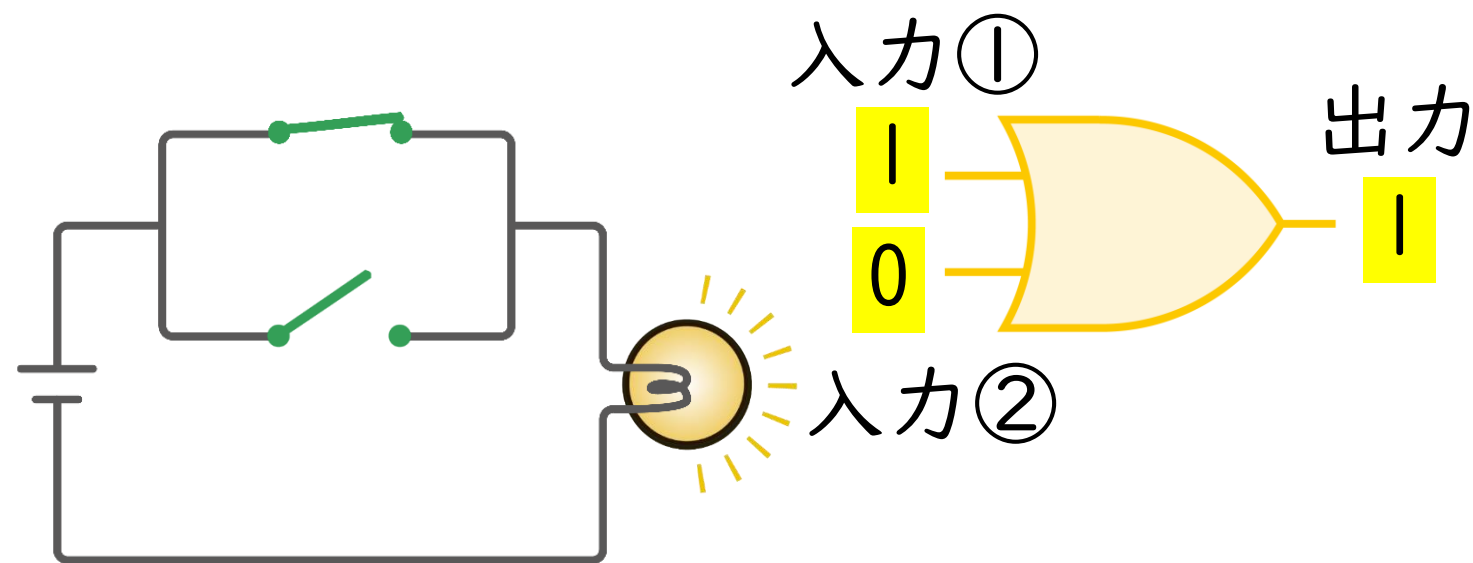


①	②	出力
0	0	0
1	0	1
0	1	1
1	1	1



基本論理回路

■OR回路 入力がどれか1つでも1なら出力が1になる回路。

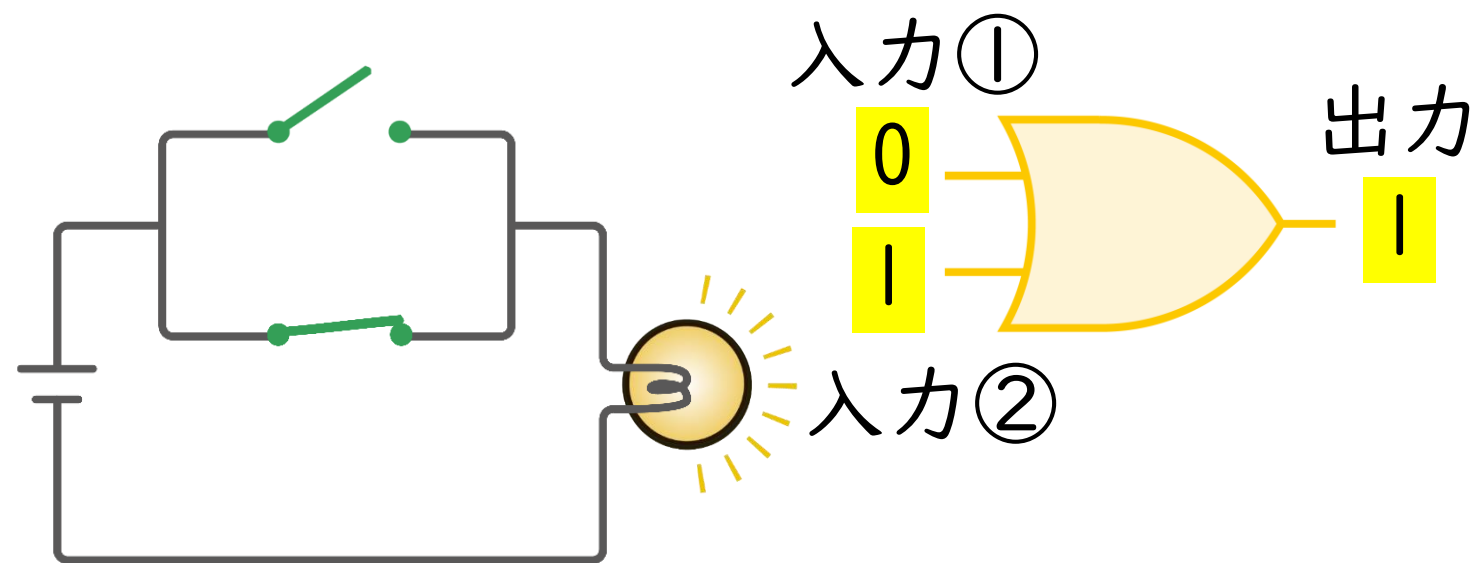


①	②	出力
0	0	0
1	0	1
0	1	1
1	1	1



基本論理回路

■OR回路 入力がどれか1つでも1なら出力が1になる回路。

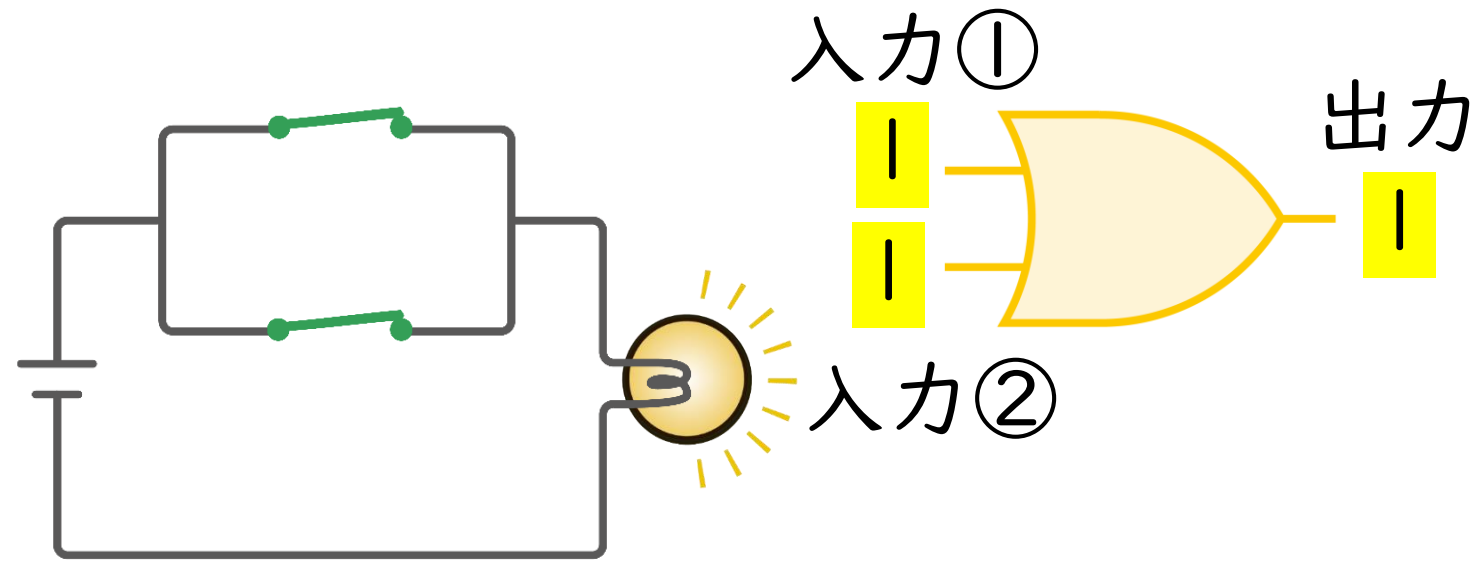


①	②	出力
0	0	0
1	0	1
0	1	1
1	1	1



基本論理回路

- OR回路 入力がどれか1つでも1なら出力が1になる回路。

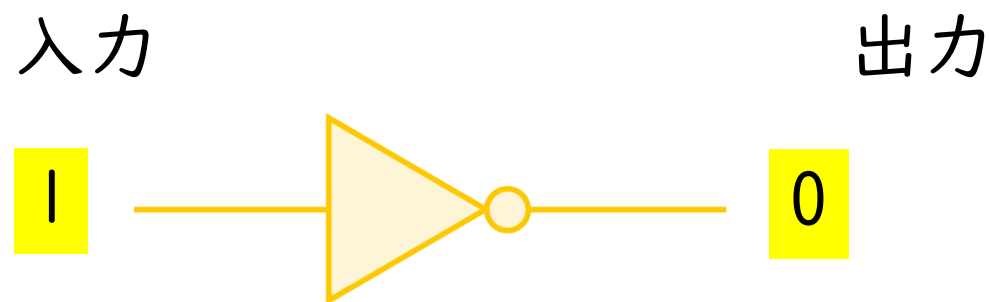


①	②	出力
0	0	0
1	0	1
0	1	1
1	1	1



基本論理回路

■NOT回路 入力と反対の結果を出力する回路。



入力	出力
0	1
1	0

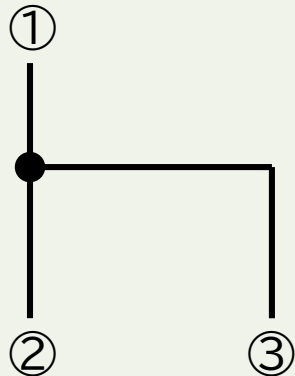
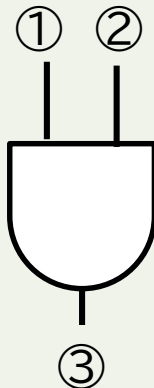
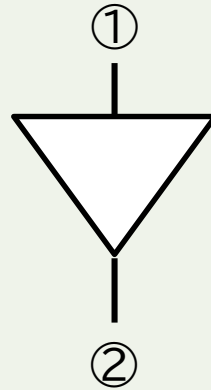
これらの回路は，入力された電気信号に対応する信号を出力する。

論理回路の組み合わせにより，さまざまな処理が実現されている。



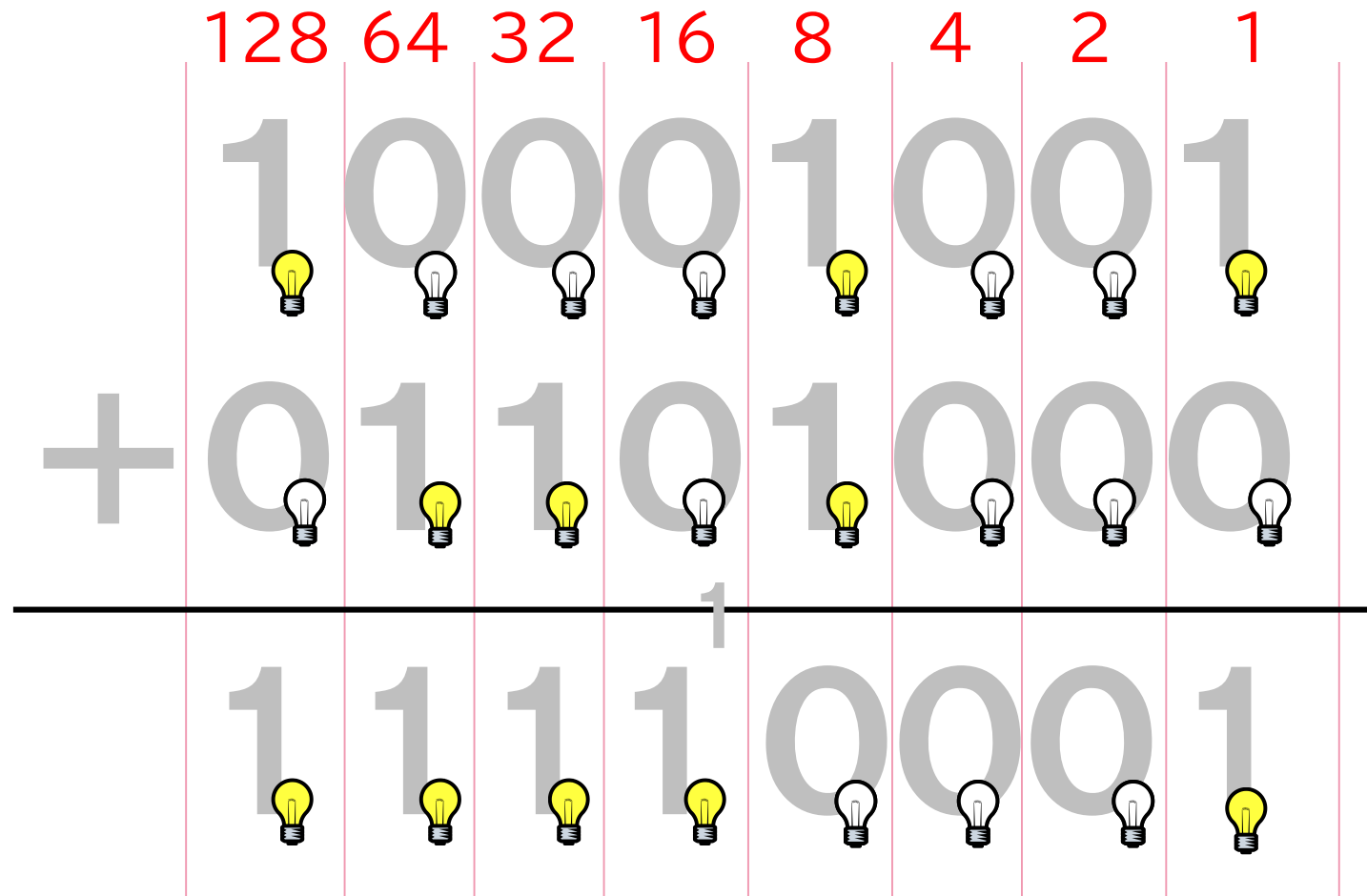
MEMO

AND 回路, OR 回路, NOT回路をまとめて, 基本論理回路という。

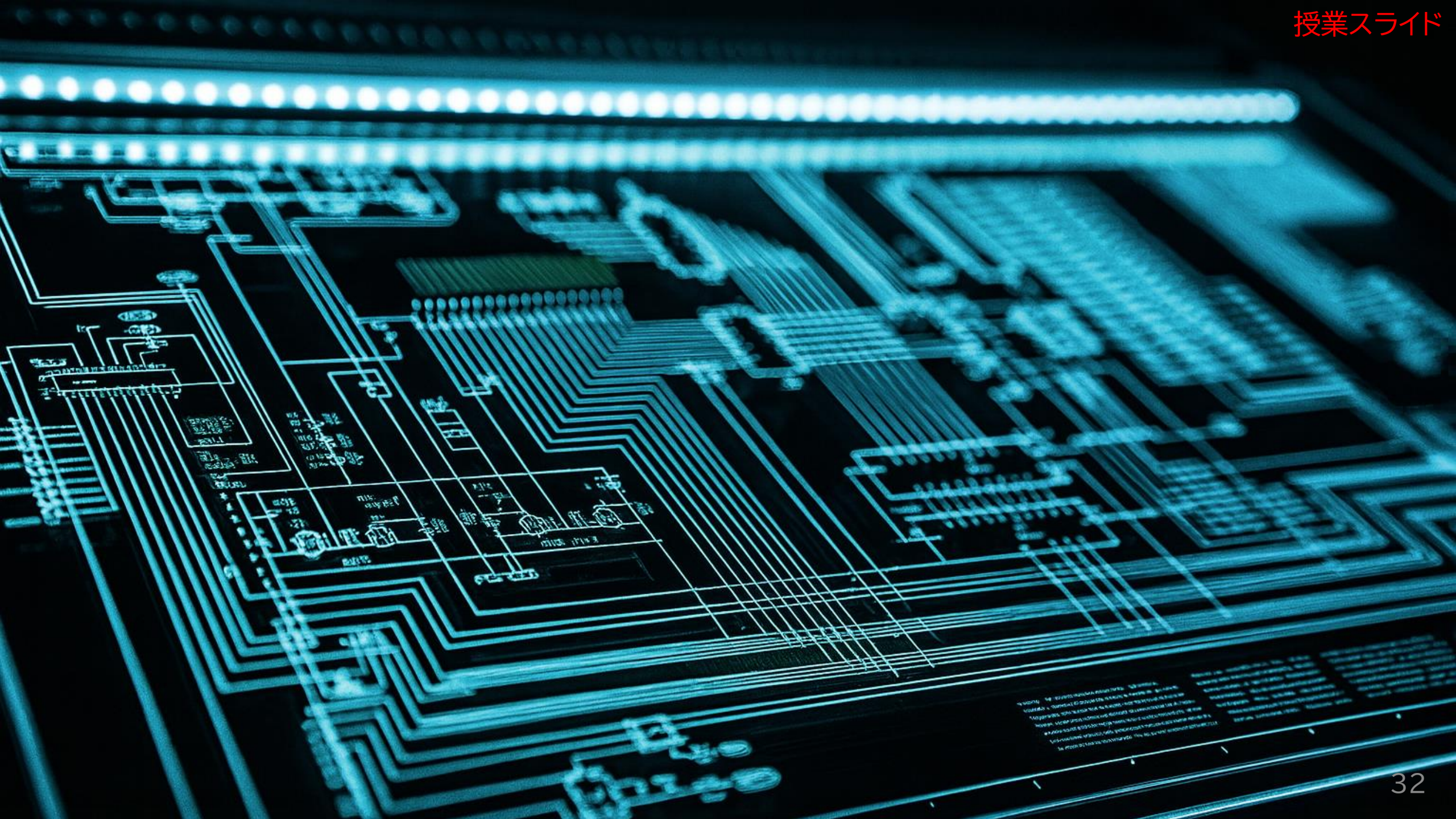
導線	AND	OR	NOT																																													
電気が流れる。 つなぐことで、分岐 もできる。	2本の導線から電 気が流れてきたら、 後の1本の導線に 電気が流れる。	2本の導線のうち、 一方から電気が流 れてきたら、後の1 本の導線に電気が 流れる。	導線から電気が流 れてこなかったら、 後の1本の導線に 電気が流れる。																																													
																																																
<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	①	②	③	0	0	0	1	1	1	<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	①	②	③	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	①	②	③	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><th>①</th><th>②</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	①	②	0	1	1	0
①	②	③																																														
0	0	0																																														
1	1	1																																														
①	②	③																																														
0	0	0																																														
0	1	0																																														
1	0	0																																														
1	1	1																																														
①	②	③																																														
0	0	0																																														
0	1	1																																														
1	0	1																																														
1	1	1																																														
①	②																																															
0	1																																															
1	0																																															

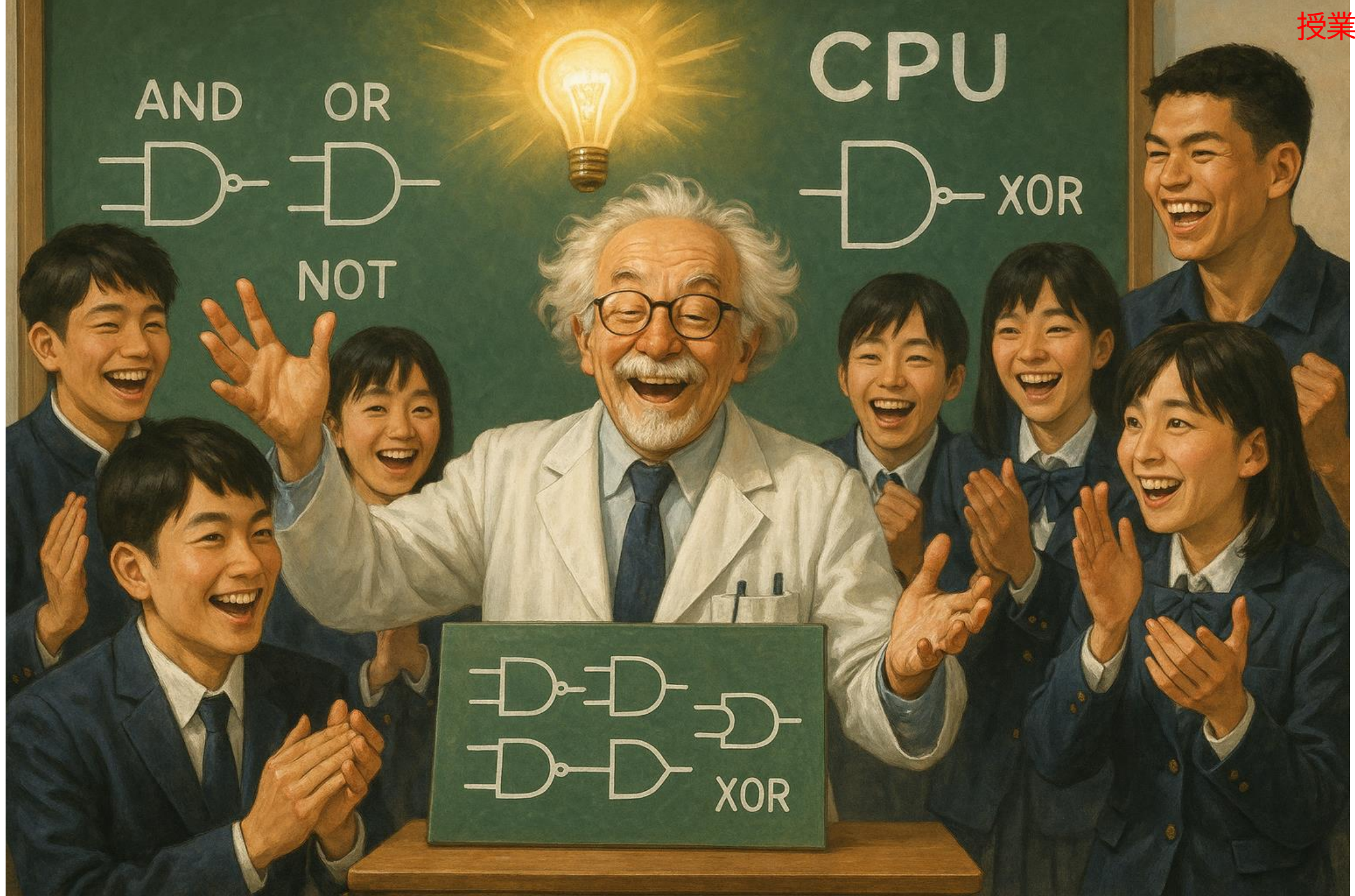
2進数の計算

$$10001001_{(2)} + 01101000_{(2)}$$



$$\begin{aligned} &137_{(10)} \\ &+ \\ &104_{(10)} \\ &= \\ &241_{(10)} \end{aligned}$$





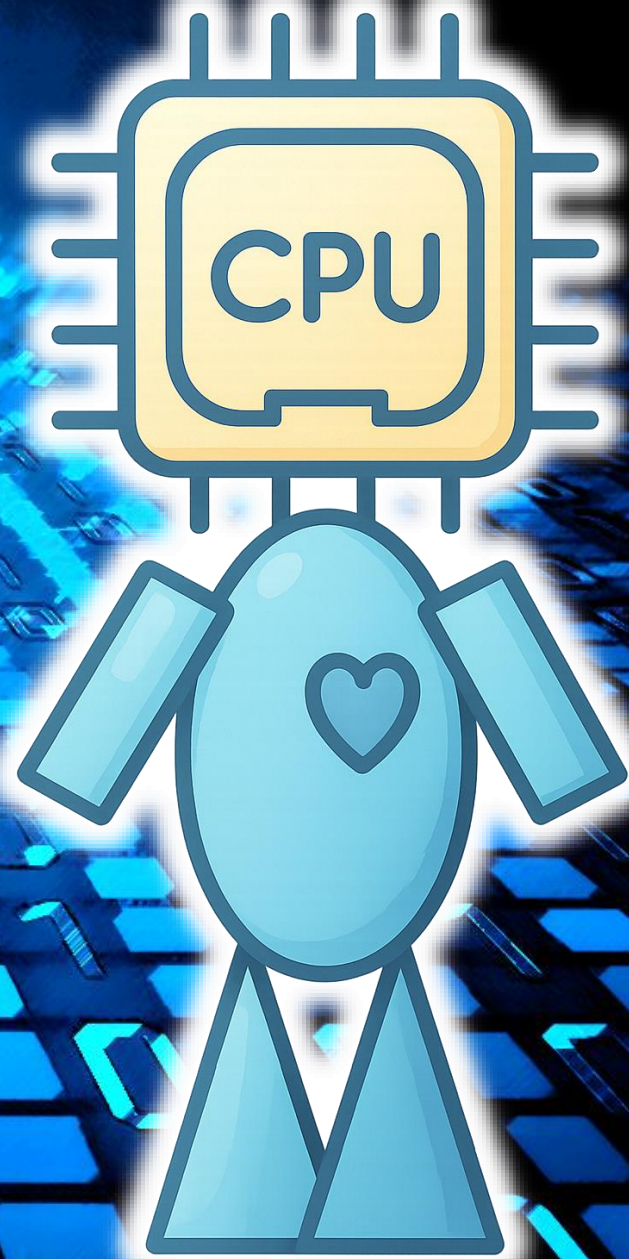
AND OR
NOT

CPU
XOR

XOR

談話

CPUくんと 友達になろう



授業計画

時間	内容
0- 5分	二進数筆算の復習
5 - 10分	ステップ1: チュートリアル
10 - 17分	ステップ2: 半加算機設計【Level 1】
17 - 20分	ステップ3: チュートリアル(XOR)
20 - 27分	ステップ4: 半加算機設計(XOR有)【Level 1】
27 - 34分	ステップ5: 全加算機設計【Level 2】
34 - 41分	ステップ6: 2ビット加算回路実装【Level 3】
41 - 48分	ステップ7: 4ビット加算回路実装【Level Max】
49 - 50分	まとめ & 振り返り

2進数の計算

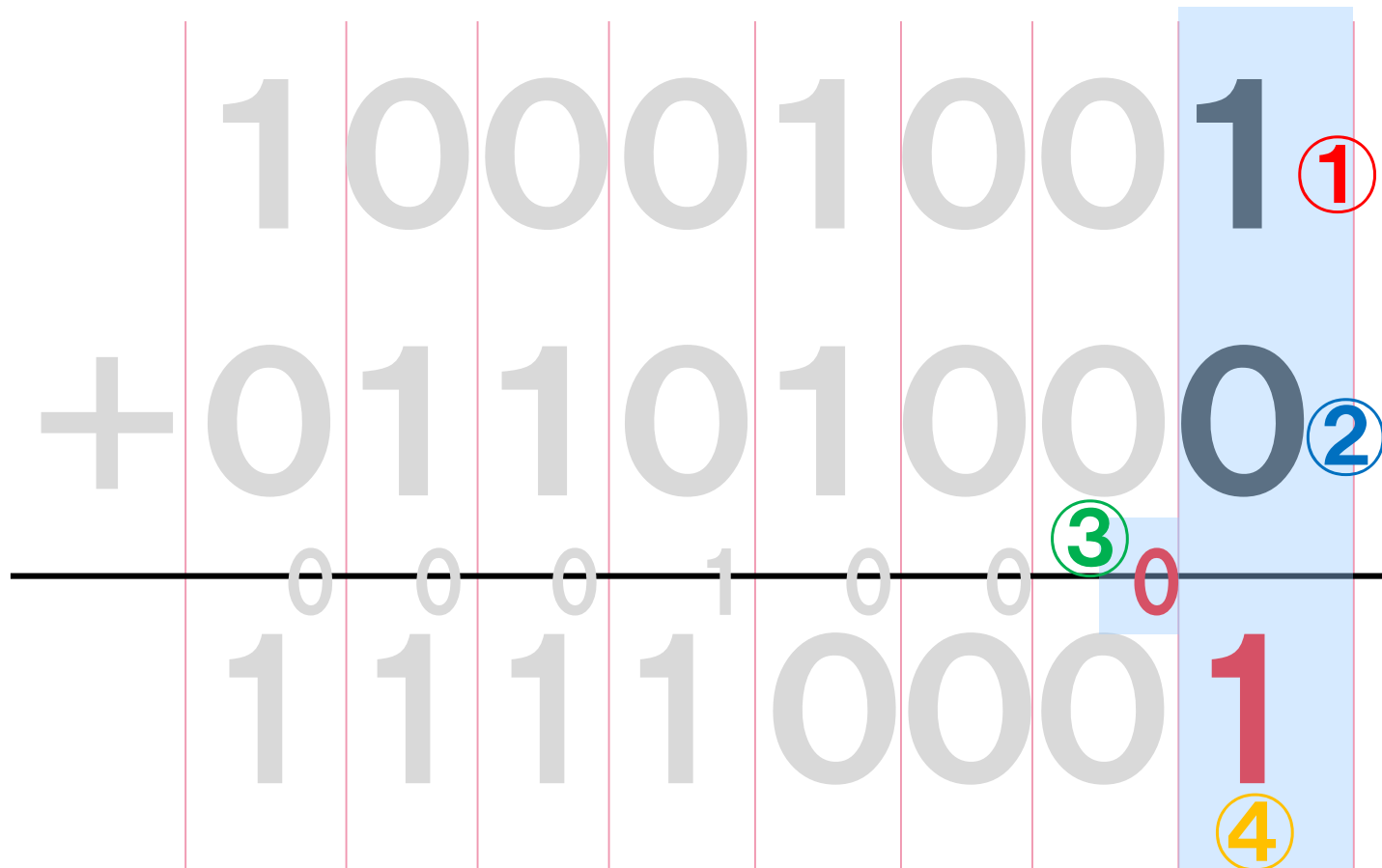
$$10001001_{(2)} + 01101000_{(2)}$$

The diagram illustrates the binary addition of $10001001_{(2)}$ and $01101000_{(2)}$. The numbers are aligned by their least significant bits. A horizontal line separates the addends from the sum. Red digits below the line represent the sum of each column, with a carry of 1 to the next column on the left. A blue shaded area highlights the final result and the carry-out.

Column	1	2	3	4	5	6	7	8
Top Number	1	0	0	0	1	0	0	1
Bottom Number	0	1	1	0	1	0	0	0
Sum	1	1	1	1	0	0	0	1

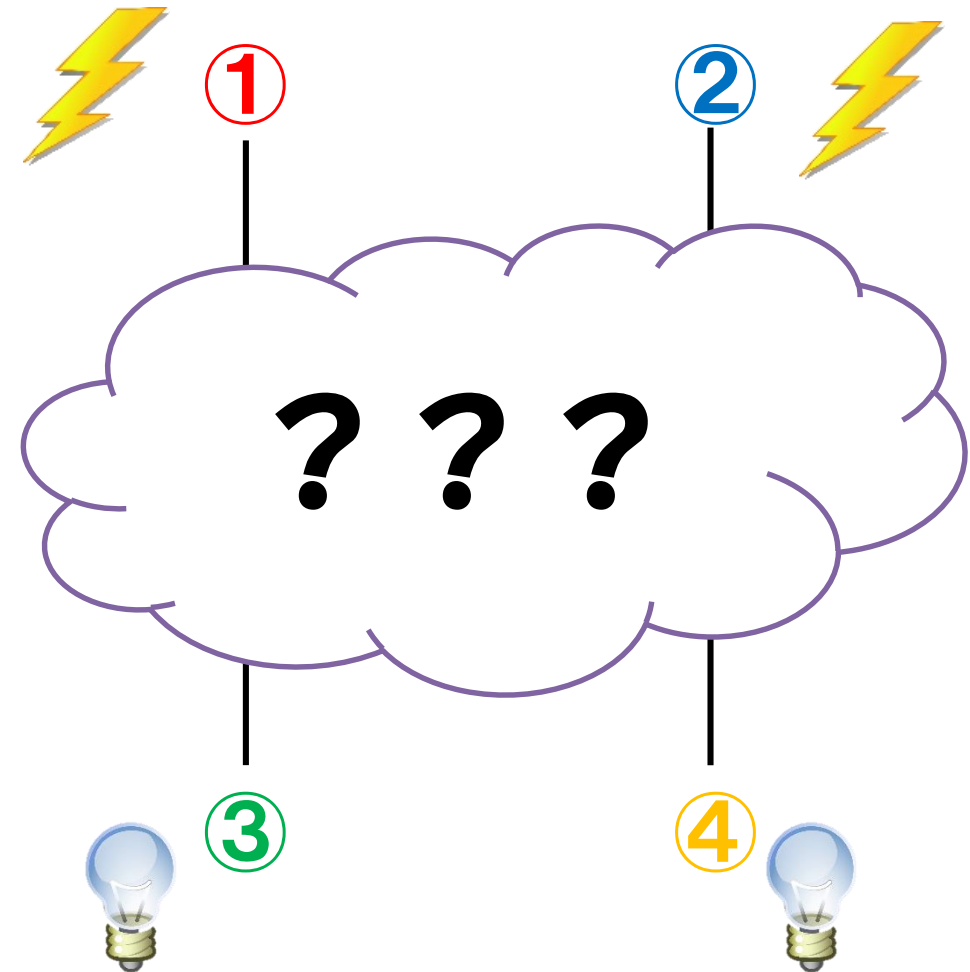
2進数の計算

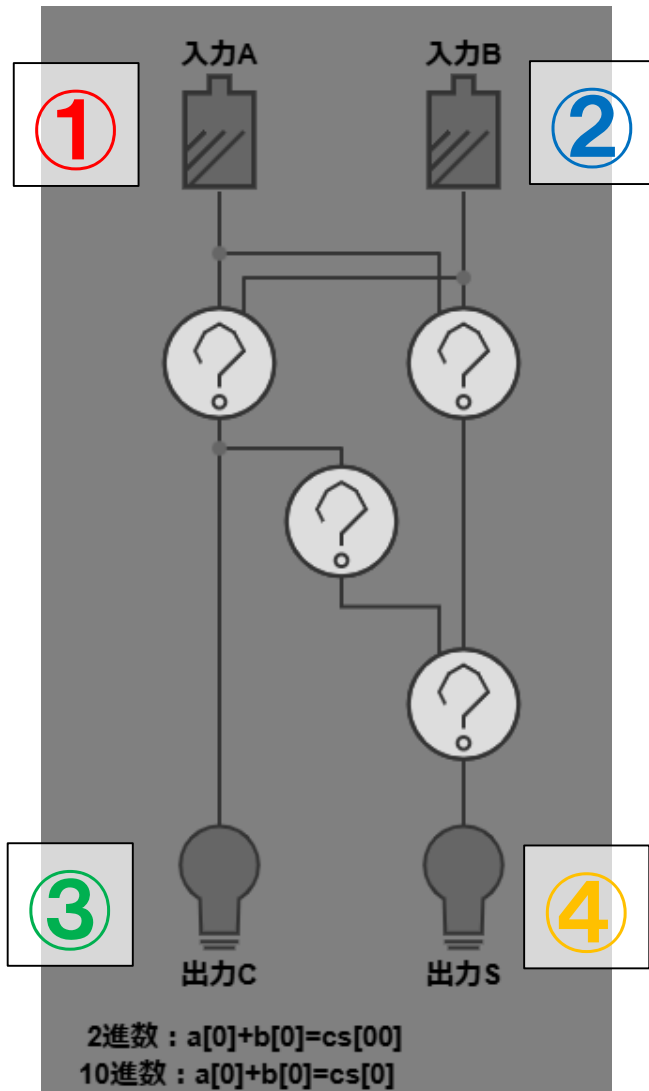
$$10001001_{(2)} + 01101000_{(2)}$$



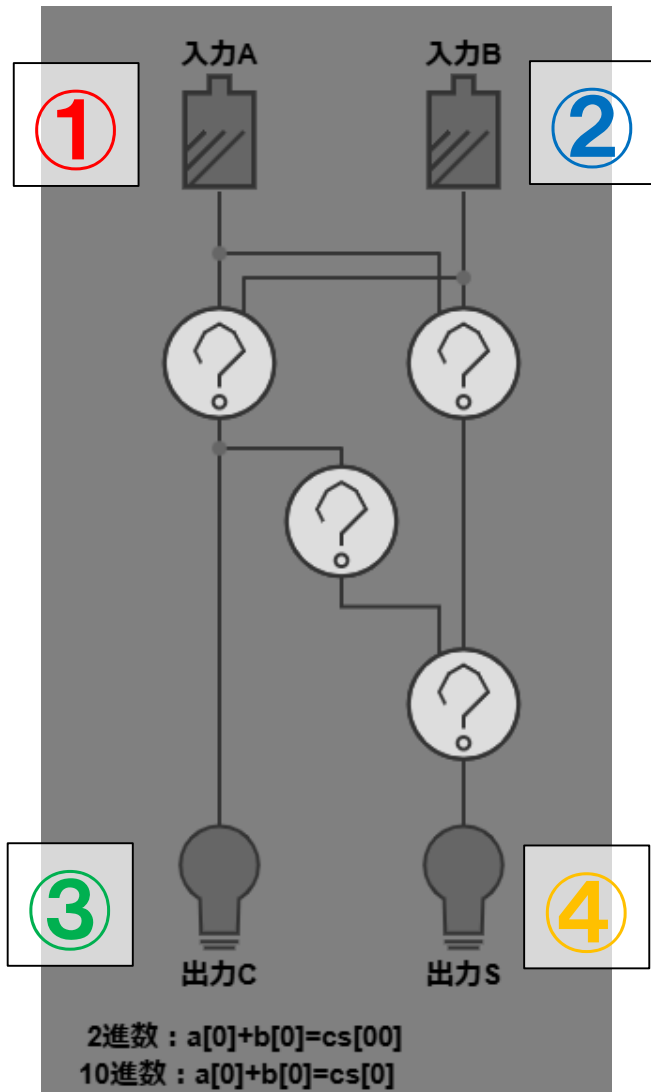
2進数の足し算を行う回路を作ってみよう。【Lv1】

①		②		③	④
0 ₍₂₎	+	0 ₍₂₎	=	0	0 ₍₂₎
0 ₍₂₎	+	1 ₍₂₎	=	0	1 ₍₂₎
1 ₍₂₎	+	0 ₍₂₎	=	0	1 ₍₂₎
1 ₍₂₎	+	1 ₍₂₎	=	1	0 ₍₂₎

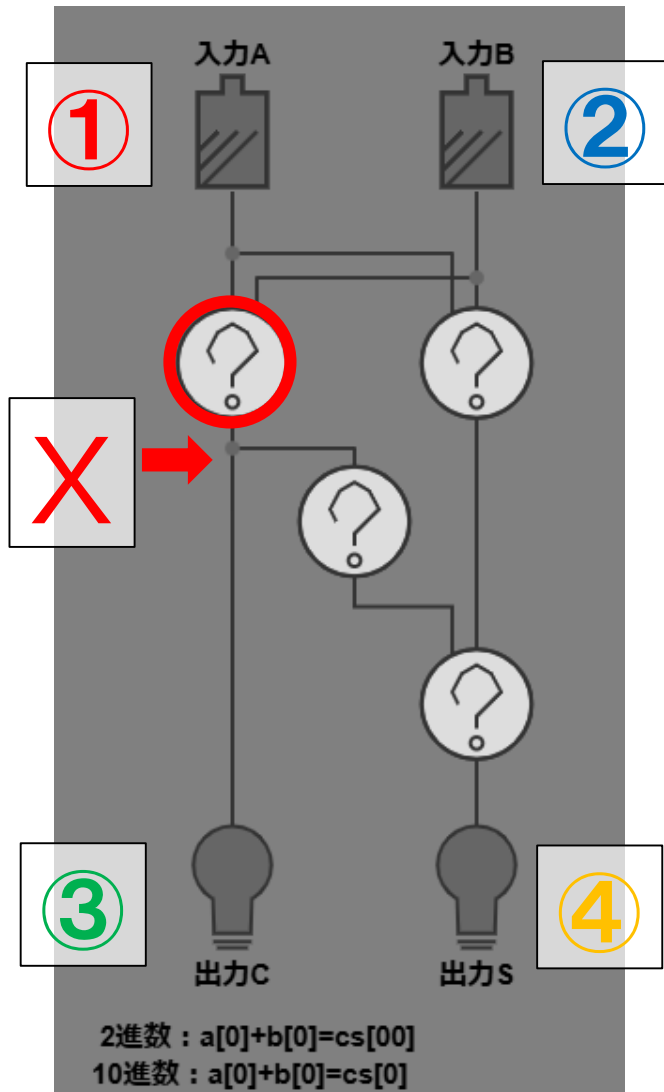




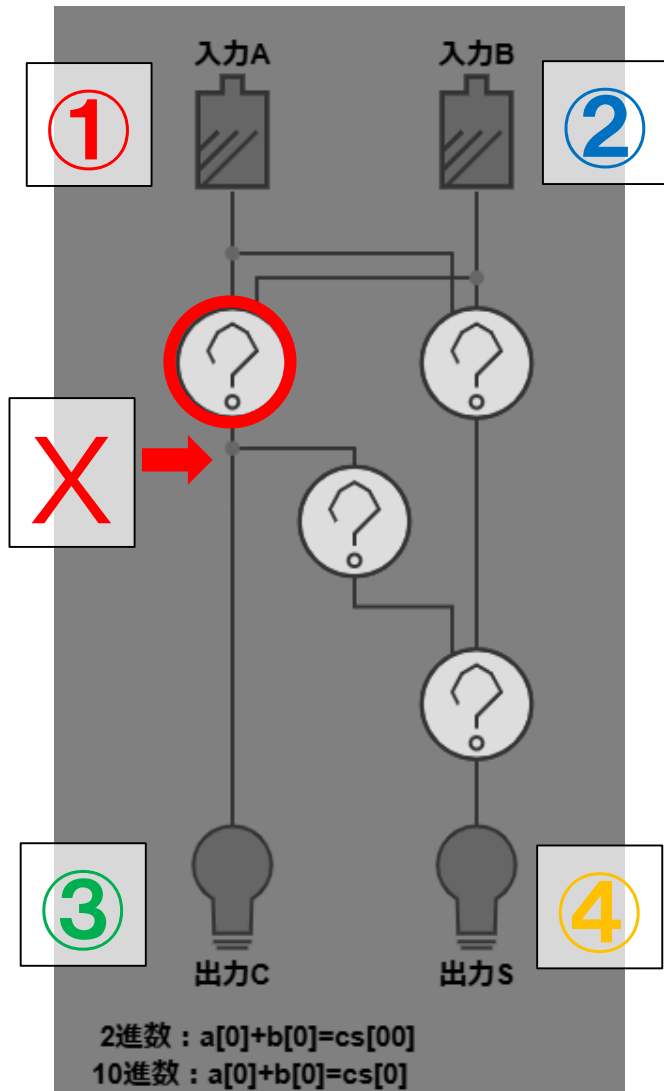
①		②		③	④
0	+	0	=	0	0
0	+	1	=	0	1
1	+	0	=	0	1
1	+	1	=	1	0



①	②				③	④
0	0				0	0
0	1				0	1
1	0				0	1
1	1				1	0

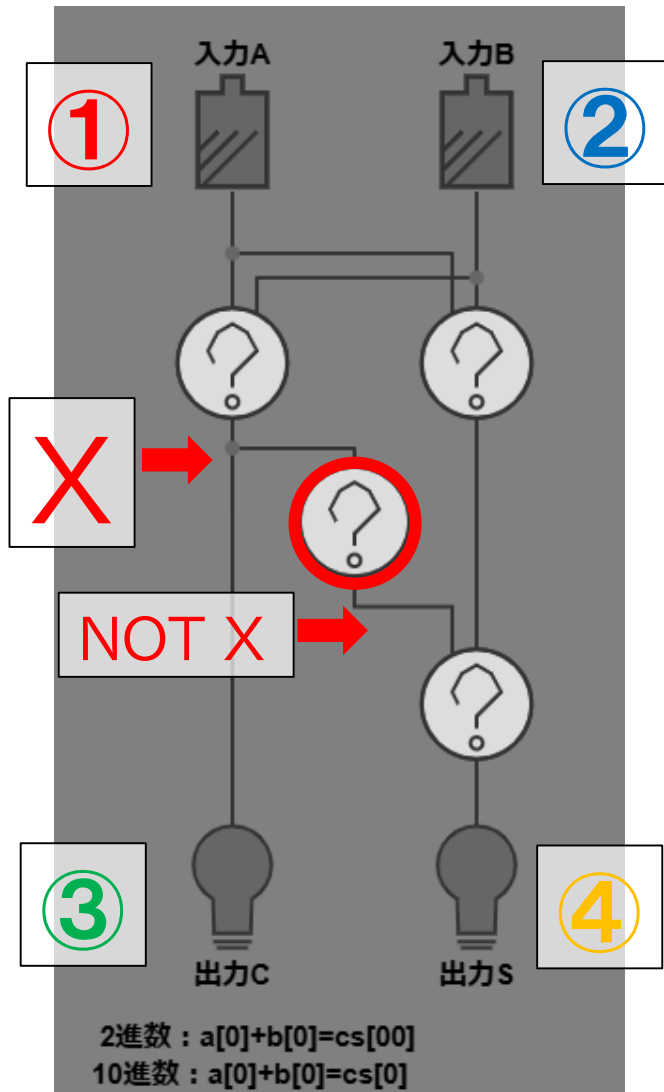


①	②	X			③	④
0	0				0	0
0	1				0	1
1	0				0	1
1	1				1	0

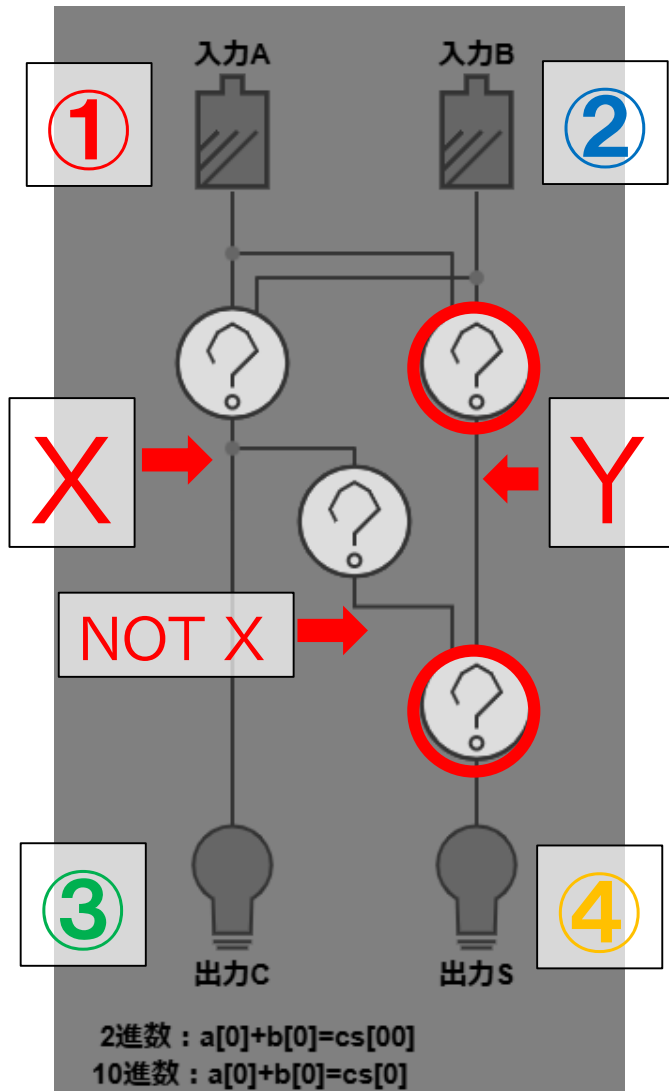


①	②	X			③	④
0	0	0			0	0
0	1	0			0	1
1	0	0			0	1
1	1	1			1	0

Arrows indicate the flow of data: from input ② to the XOR gate, from the XOR gate output to the sum output ④, and from the AND gate output to the carry output ③.



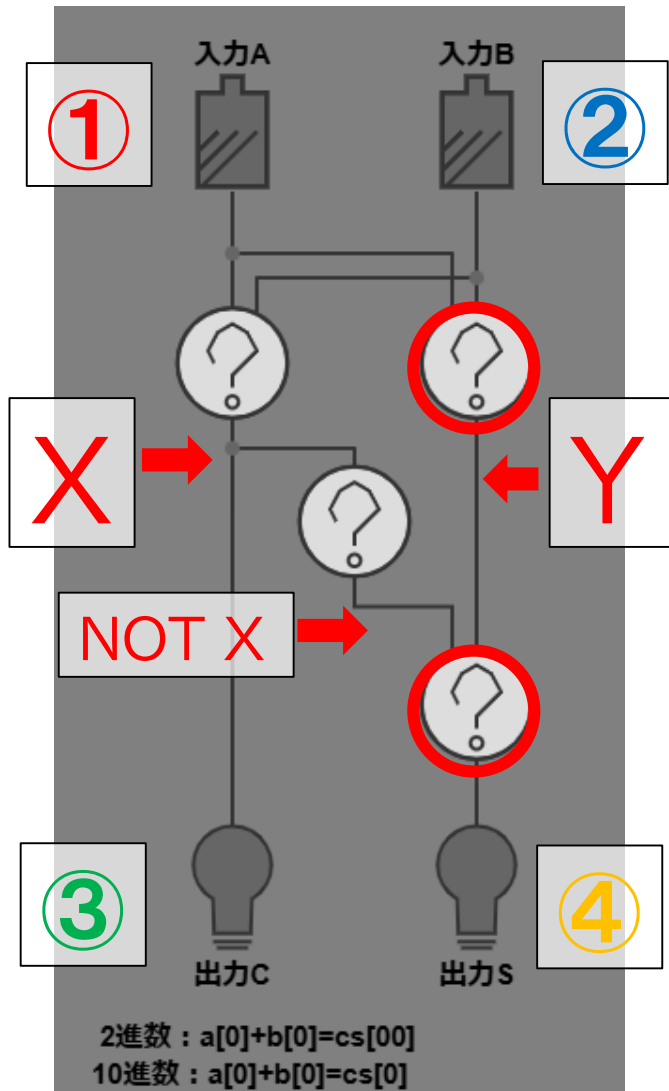
①	②	X	NOT X		③	④
0	0	0	1		0	0
0	1	0	1		0	1
1	0	0	1		0	1
1	1	1	0		1	0



①	②	X	NOT X	Y	③	④
0	0	0	1	0	0	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	1	0

Arrows indicate the flow of data from the circuit diagram to the table:

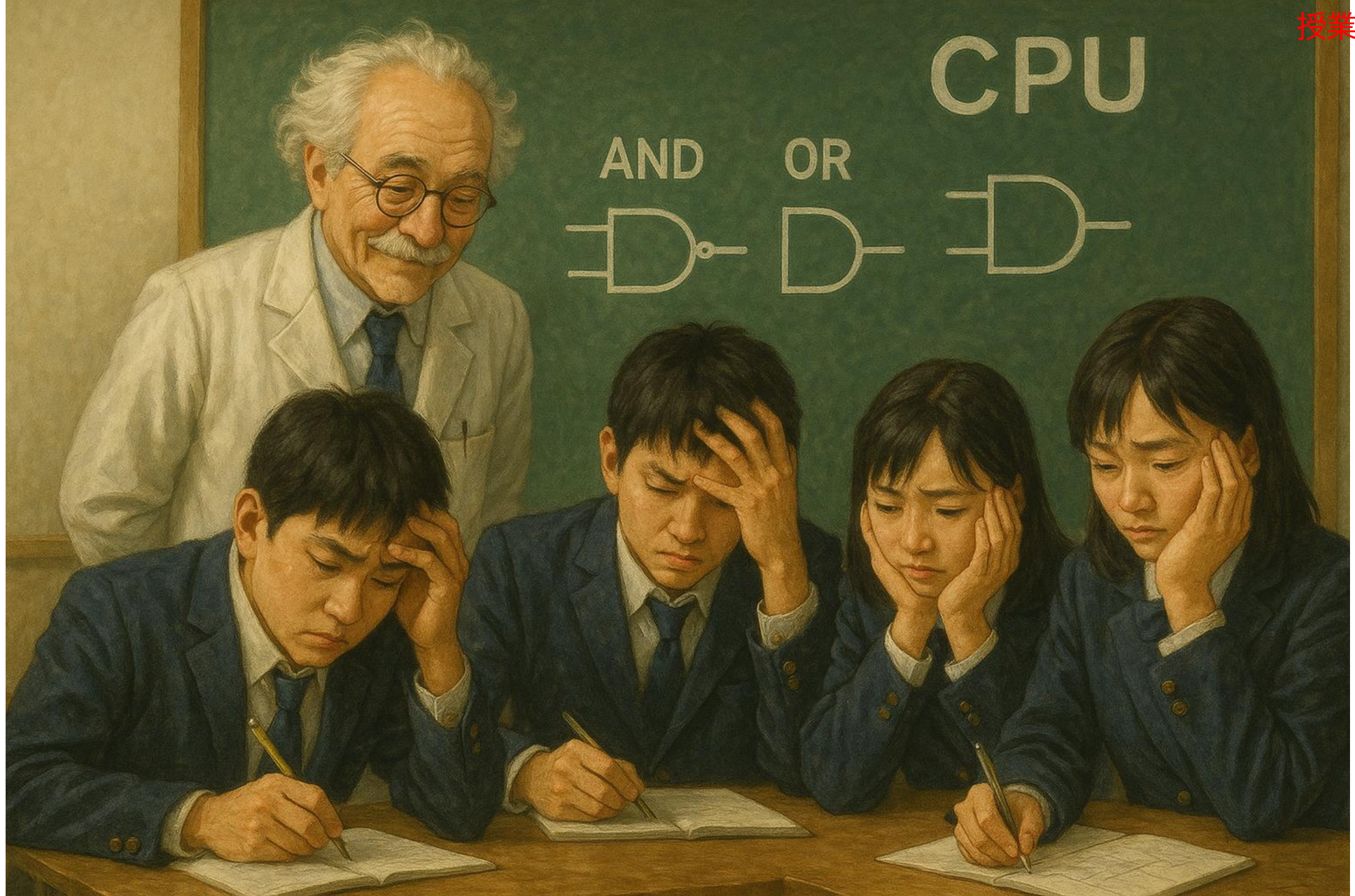
- From the XOR gate output (X) to the third column.
- From the AND gate output (NOT X) to the fourth column.
- From the OR gate output (Y) to the fifth column.
- From the output C (出力C) to the sixth column.
- From the output S (出力S) to the seventh column.



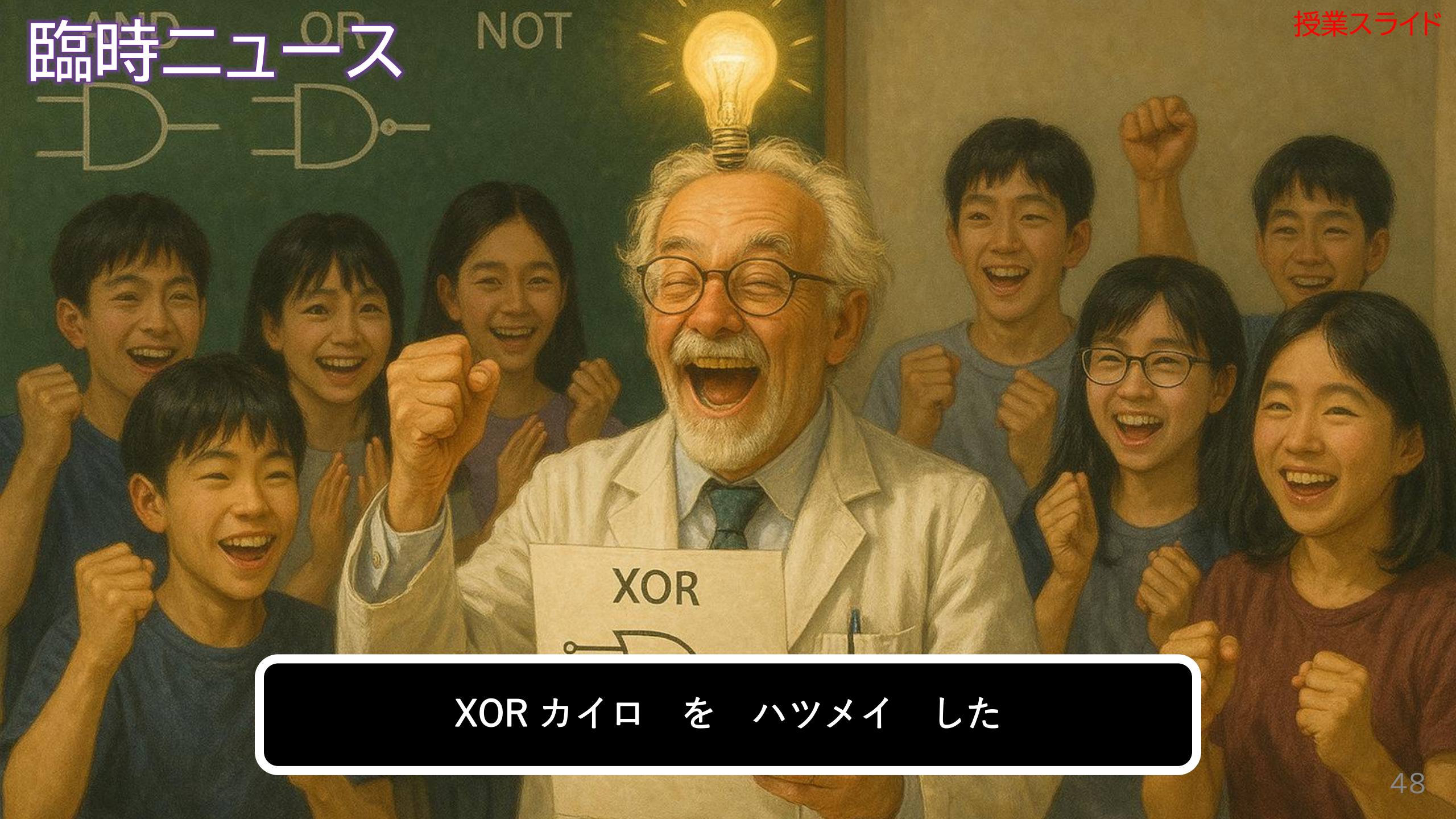
①	②	X	NOT X	Y	③	④
0	0	0	1	0	0	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	1	0

AND

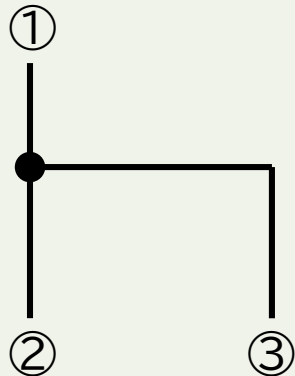
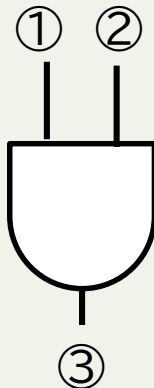
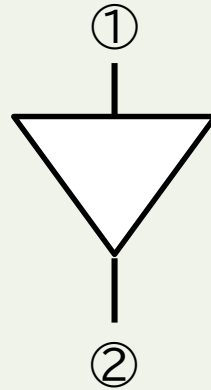
OR

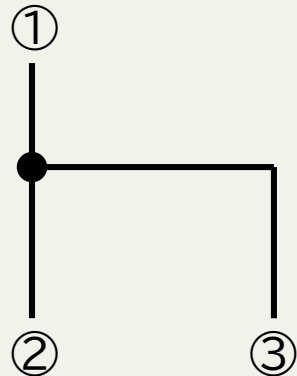
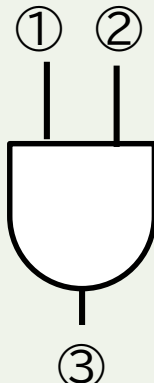
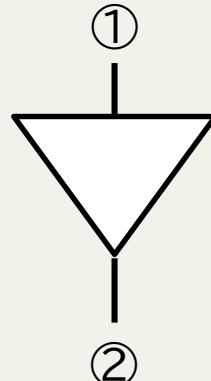


臨時ニュース



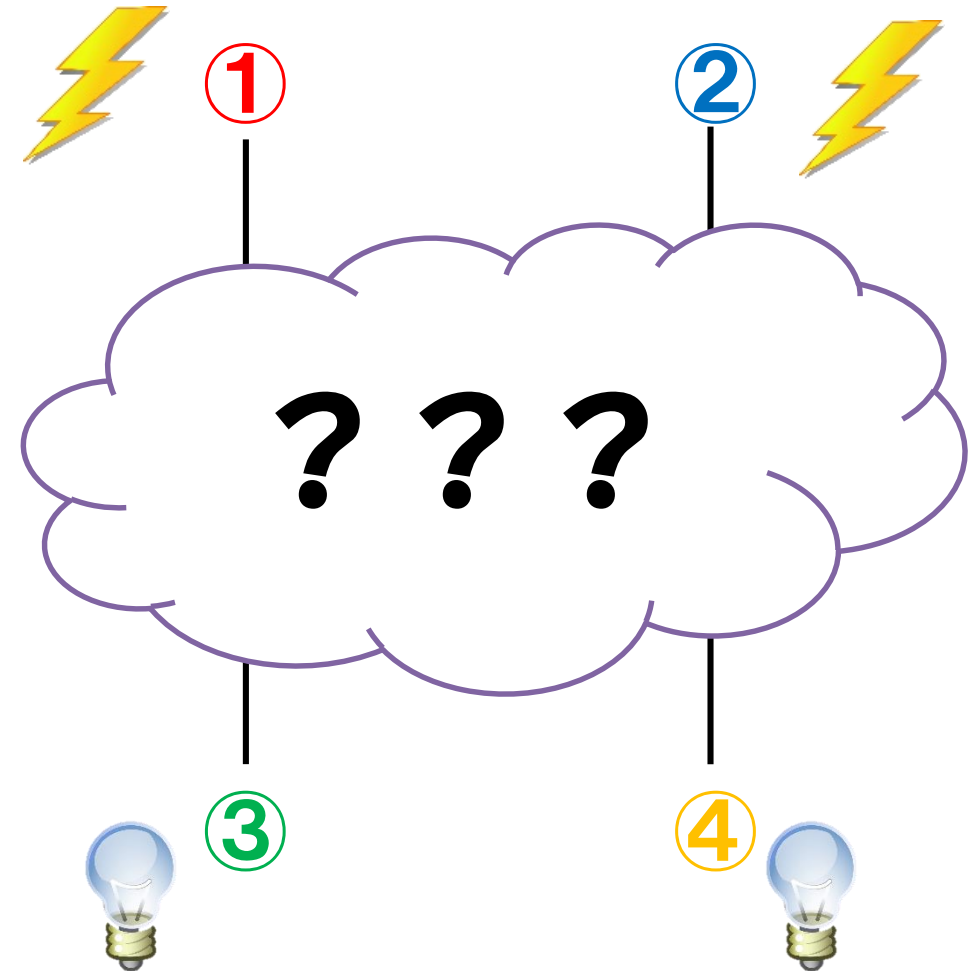
XOR カイロ を ハツメイ した

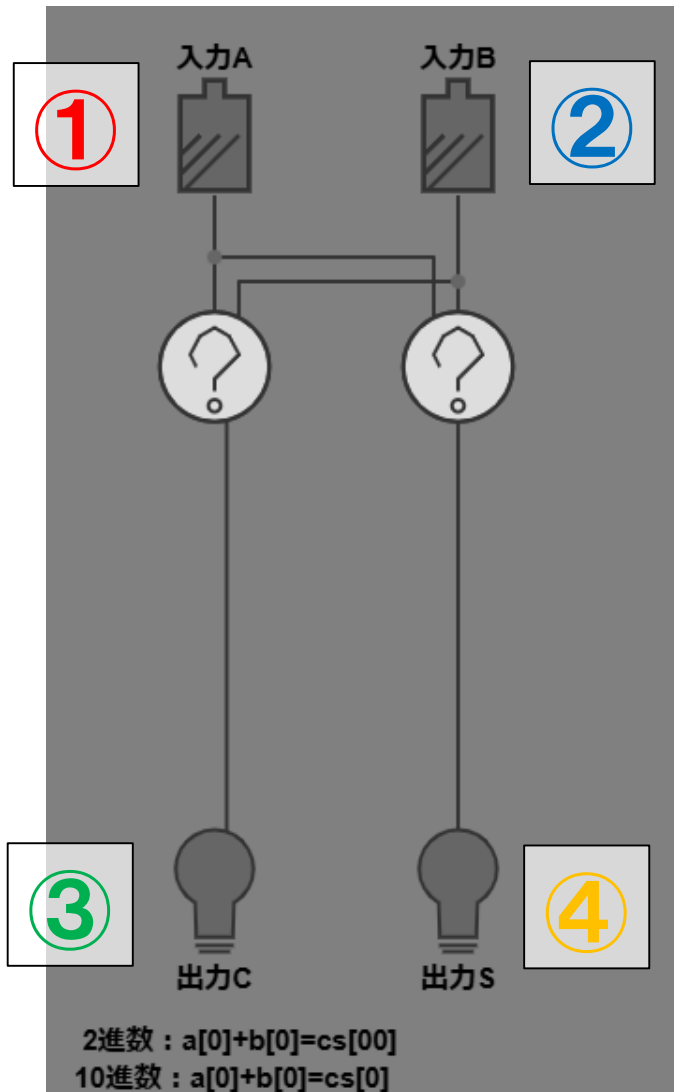
導線	AND	OR	NOT																																													
電気が流れる。 つなぐことで、分岐 もできる。	2本の導線から電 気が流れてきたら、 後の1本の導線に 電気が流れる。	2本の導線のうち、 一方から電気が流 れてきたら、後の1 本の導線に電気が 流れる。	導線から電気が流 れてこなかったら、 後の1本の導線に 電気が流れる。																																													
																																																
<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	①	②	③	0	0	0	1	1	1	<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	①	②	③	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	①	②	③	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><th>①</th><th>②</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	①	②	0	1	1	0
①	②	③																																														
0	0	0																																														
1	1	1																																														
①	②	③																																														
0	0	0																																														
0	1	0																																														
1	0	0																																														
1	1	1																																														
①	②	③																																														
0	0	0																																														
0	1	1																																														
1	0	1																																														
1	1	1																																														
①	②																																															
0	1																																															
1	0																																															

導線	AND	OR	NOT	XOR																																																												
電気が流れる。 つなぐことで、分岐 もできる。	2本の導線から電 気が流れてきたら、 後の1本の導線に 電気が流れる。	2本の導線のうち、 一方から電気が流 れてきたら、後の1 本の導線に電気が 流れる。	導線から電気が流 れてこなかったら、 後の1本の導線に 電気が流れる。	2本の導線のうち、 1本だけから電気が 流れてきたら、後の 1本の導線に電気 が流れる。																																																												
																																																																
<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	①	②	③	0	0	0	1	1	1	<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	①	②	③	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	①	②	③	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><th>①</th><th>②</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	①	②	0	1	1	0	<table><tr><th>①</th><th>②</th><th>③</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	①	②	③	0	0	0	0	1	1	1	0	1	1	1	0
①	②	③																																																														
0	0	0																																																														
1	1	1																																																														
①	②	③																																																														
0	0	0																																																														
0	1	0																																																														
1	0	0																																																														
1	1	1																																																														
①	②	③																																																														
0	0	0																																																														
0	1	1																																																														
1	0	1																																																														
1	1	1																																																														
①	②																																																															
0	1																																																															
1	0																																																															
①	②	③																																																														
0	0	0																																																														
0	1	1																																																														
1	0	1																																																														
1	1	0																																																														

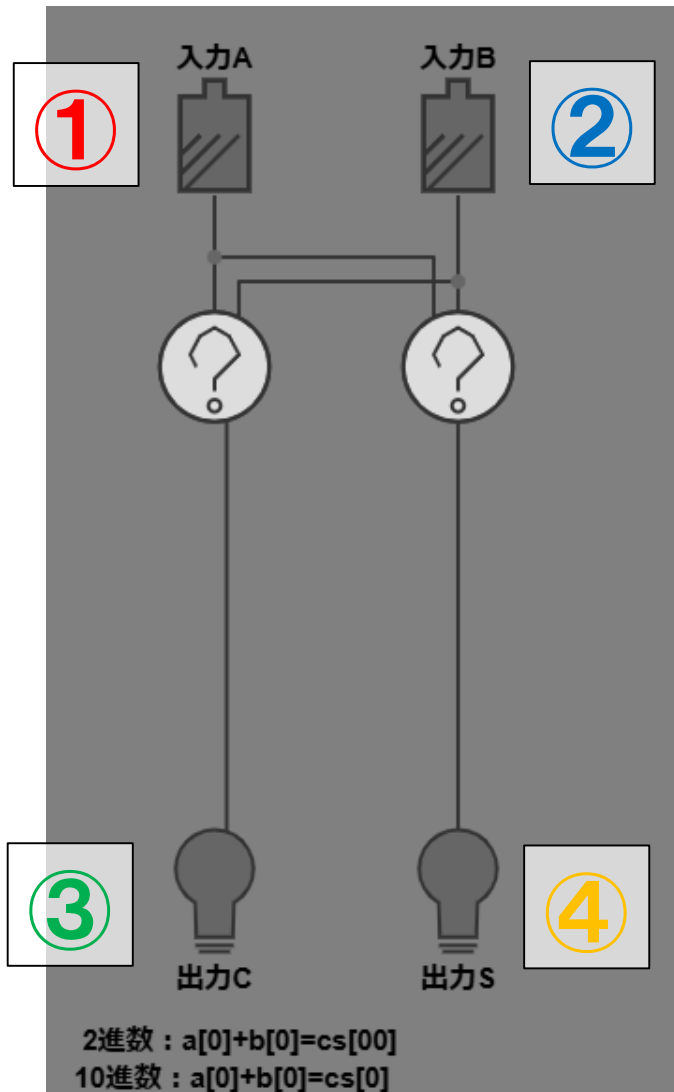
2進数の足し算を行う回路を作ってみよう。【Lv1】

①		②		③	④
0 ₍₂₎	+	0 ₍₂₎	=	0	0 ₍₂₎
0 ₍₂₎	+	1 ₍₂₎	=	0	1 ₍₂₎
1 ₍₂₎	+	0 ₍₂₎	=	0	1 ₍₂₎
1 ₍₂₎	+	1 ₍₂₎	=	1	0 ₍₂₎

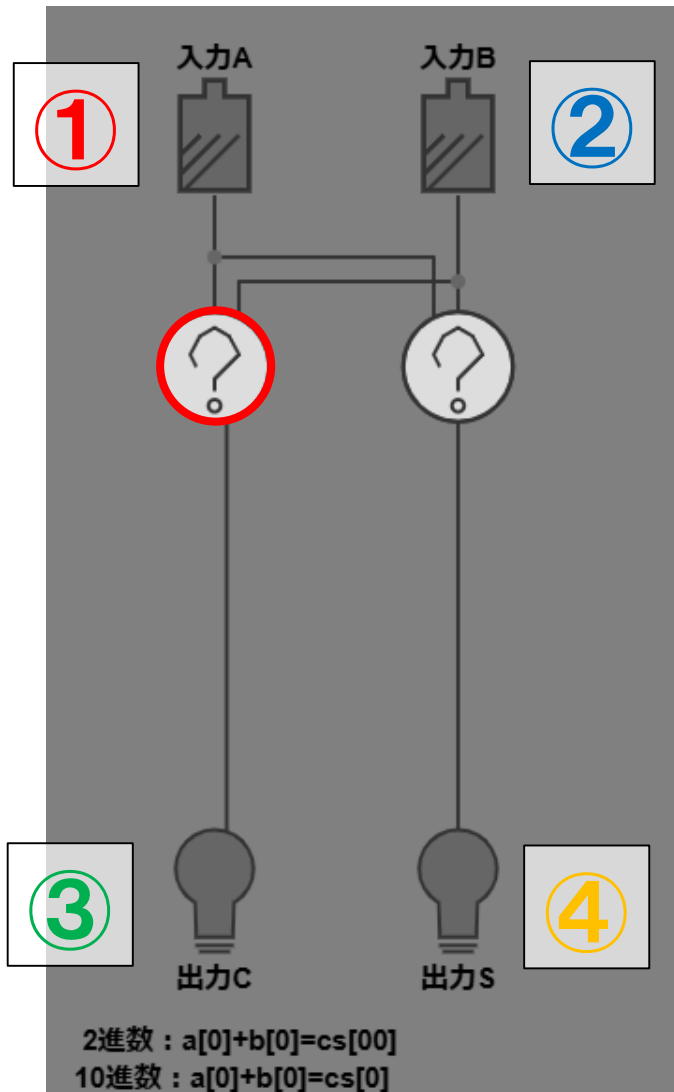




①		②	=	③	④
0	+	0	=	0	0
0	+	1	=	0	1
1	+	0	=	0	1
1	+	1	=	1	0

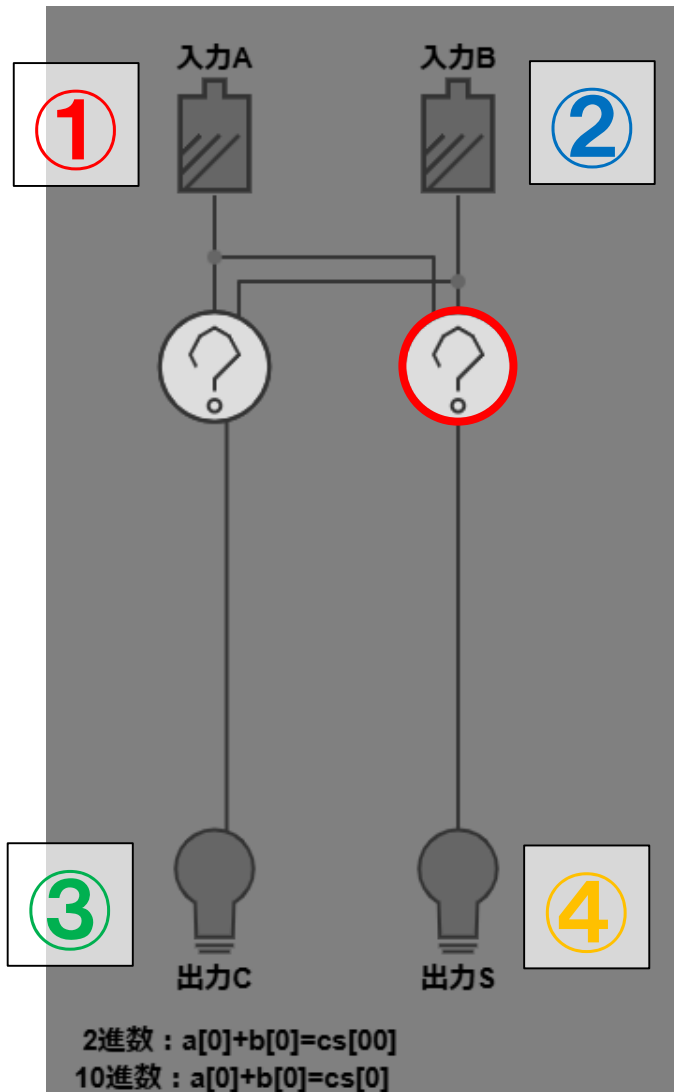


①	②	③	④
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



???

①	②	③	④
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

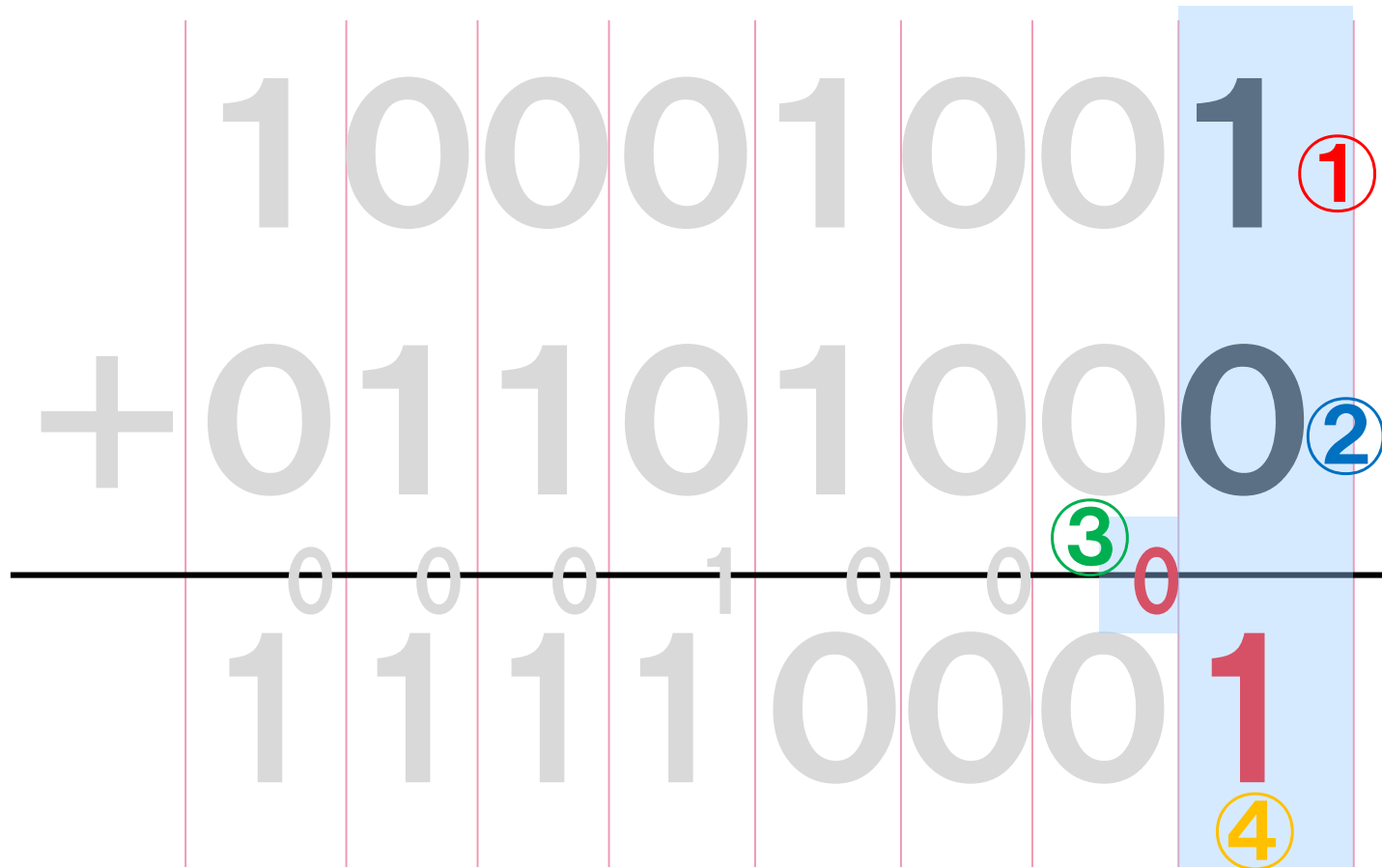


???

①	②	③	④
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

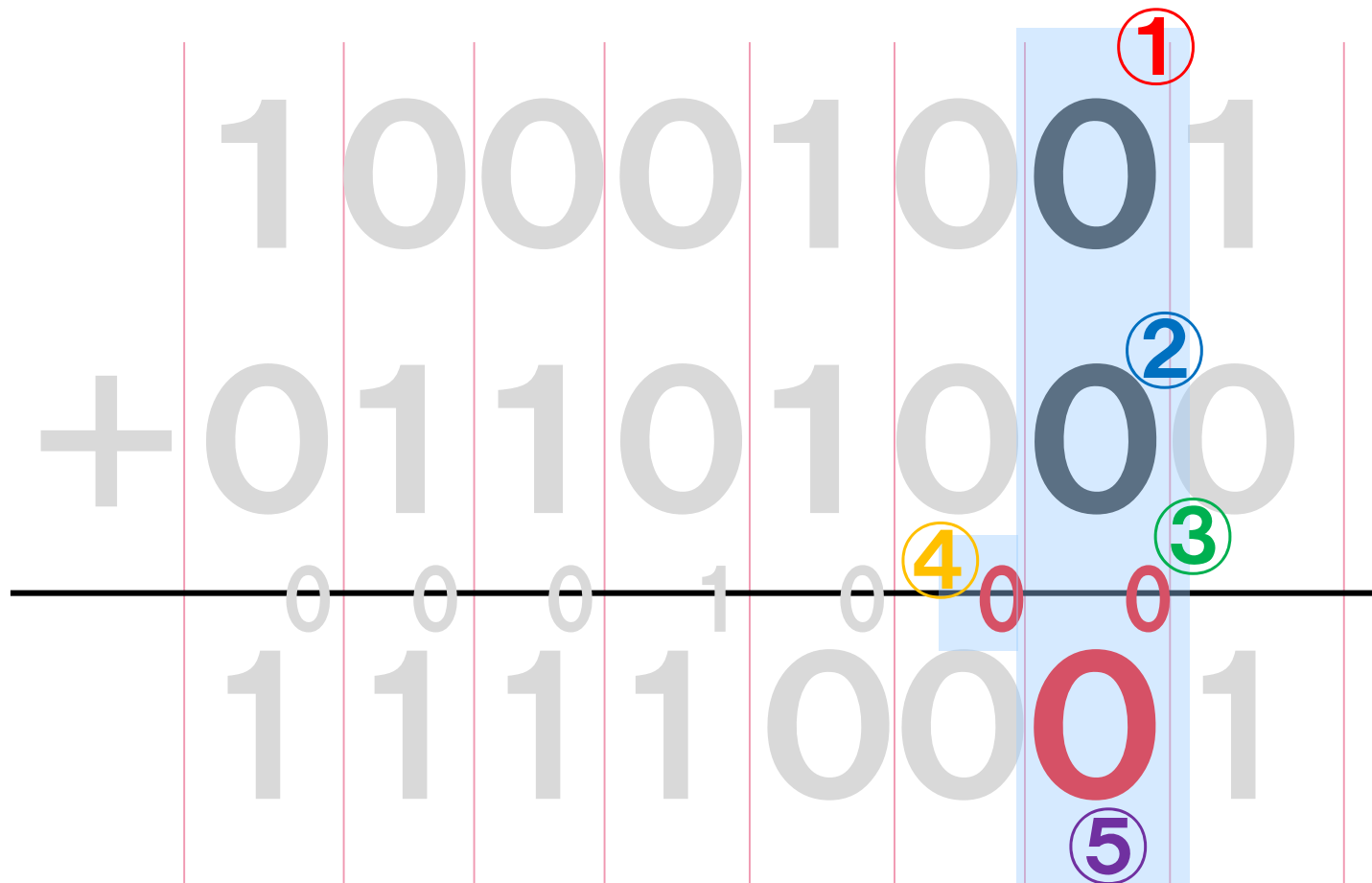
2進数の計算

$$10001001_{(2)} + 01101000_{(2)}$$



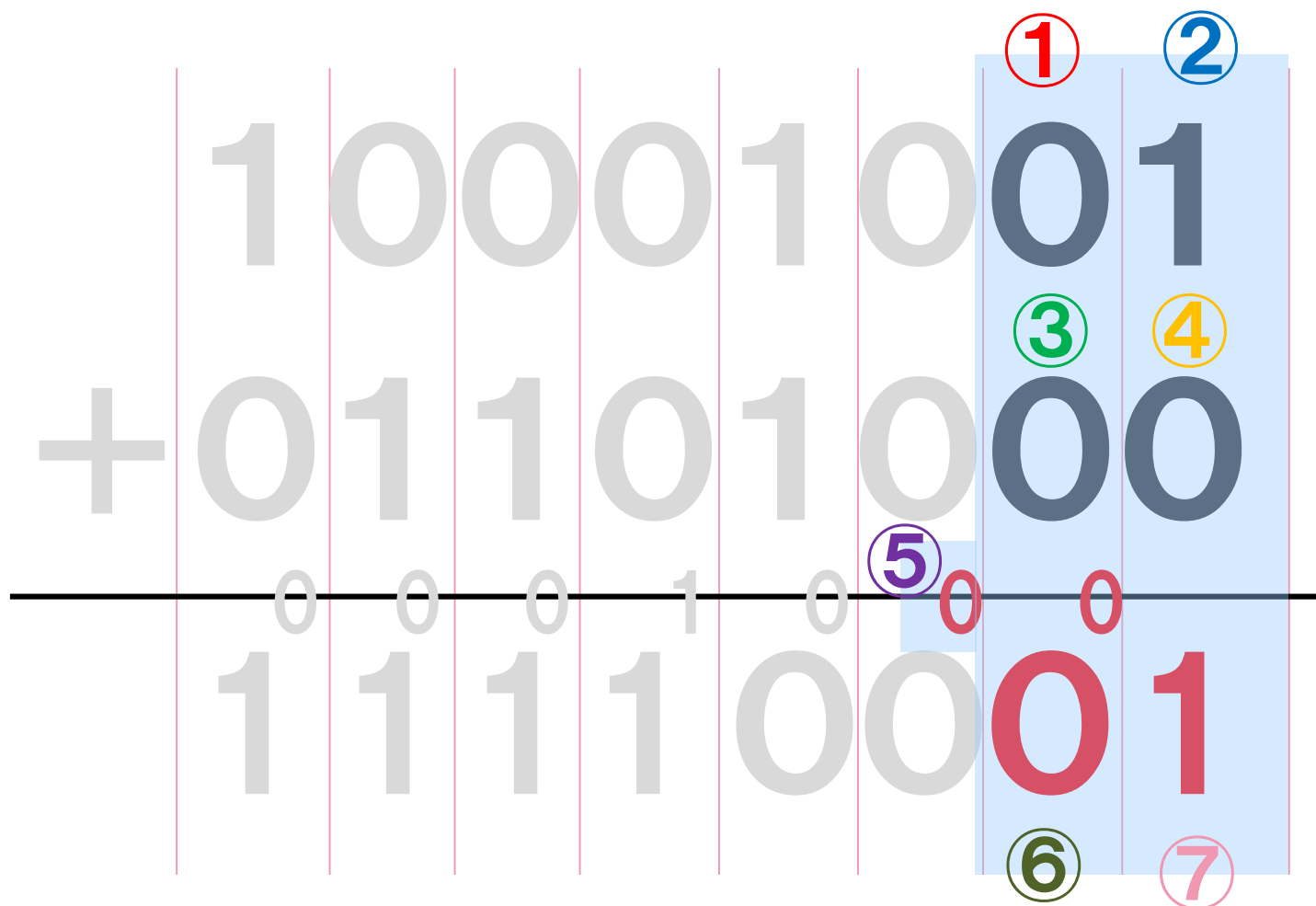
2進数の計算

$$10001001_{(2)} + 01101000_{(2)}$$



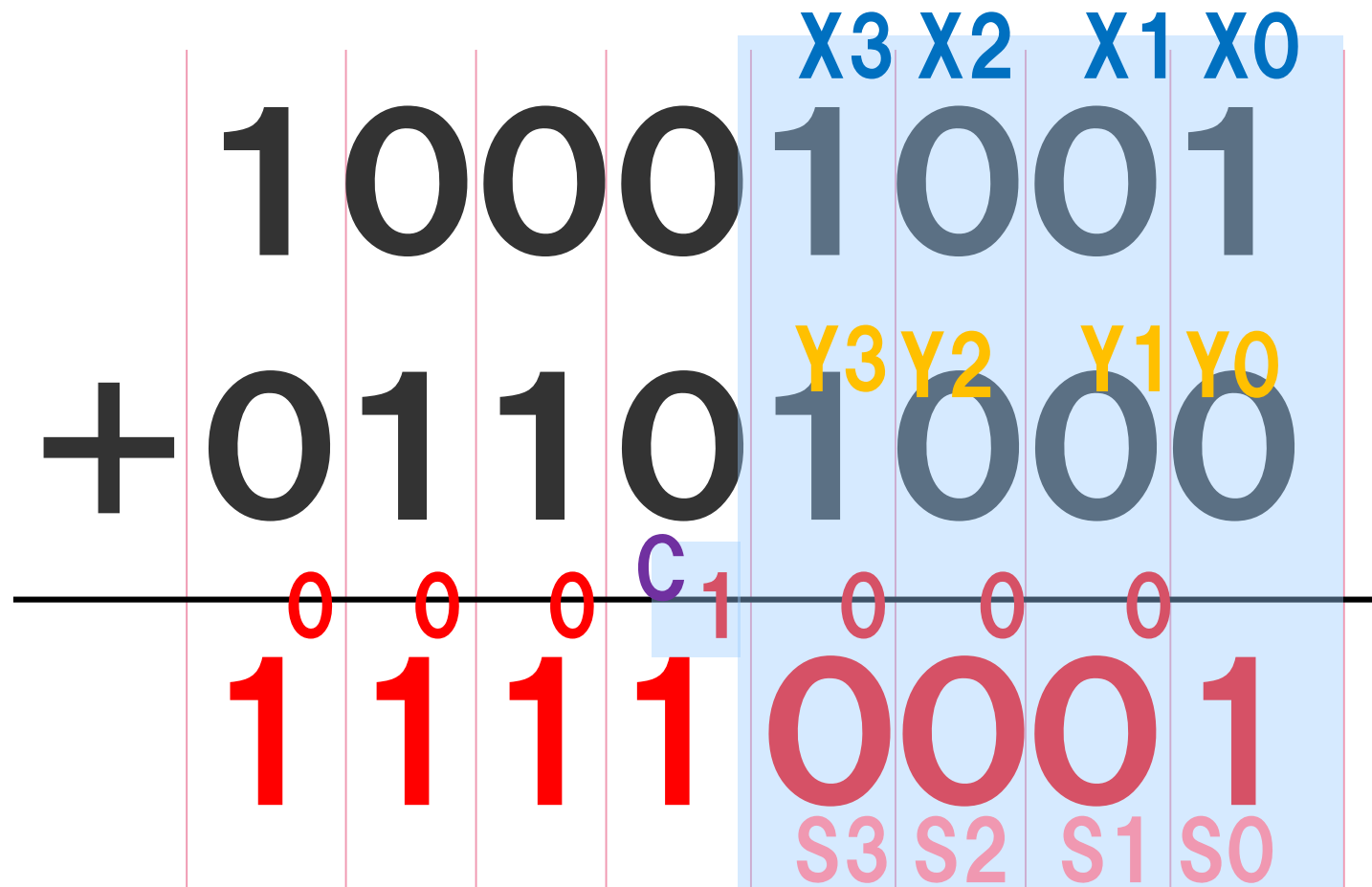
2進数の計算

$$10001001_{(2)} + 01101000_{(2)}$$



2進数の計算

$$10001001_{(2)} + 01101000_{(2)}$$



2進数の計算

$$10001001_{(2)} + 01101000_{(2)}$$

	X7	X6	X5	X4	X3	X2	X1	X0
	1	0	0	0	1	0	0	1
+	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
	0	1	1	0	1	0	0	0
C	0	0	0	0	1	0	0	0
	1	1	1	1	0	0	0	1
	S7	S6	S5	S4	S3	S2	S1	S0

授業計画

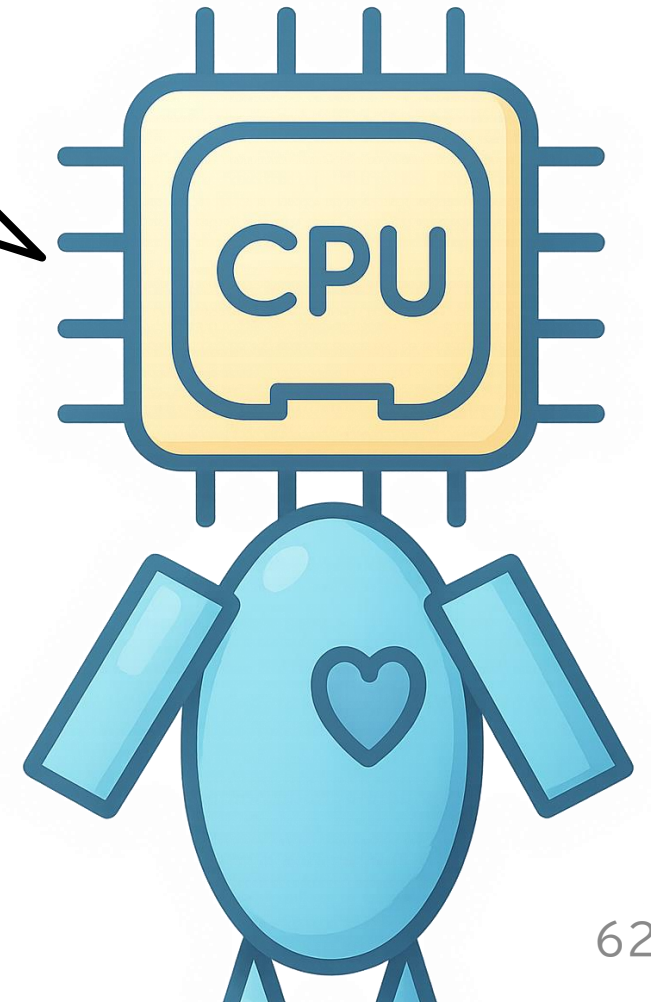
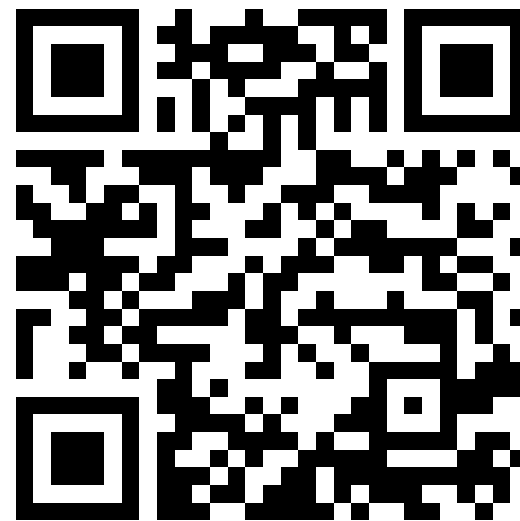
時間	内容
0 - 5分	二進数筆算の復習
5 - 10分	ステップ1: チュートリアル
10 - 17分	ステップ2: 半加算機設計【Level 1】
17 - 20分	ステップ3: チュートリアル(XOR)
20 - 27分	ステップ4: 半加算機設計(XOR有)【Level 1】
27 - 34分	ステップ5: 全加算機設計【Level 2】
34 - 41分	ステップ6: 2ビット加算回路実装【Level 3】
41 - 48分	ステップ7: 4ビット加算回路実装【Level Max】
49 - 50分	まとめ & 振り返り

論理回路の設計を体験できる ブラウザ学習教材の開発と実践

- 背景・目的
- 授業の内容
- ブラウザ学習教材
- 授業実践の結果
- まとめ

よろしく
お願いいたします

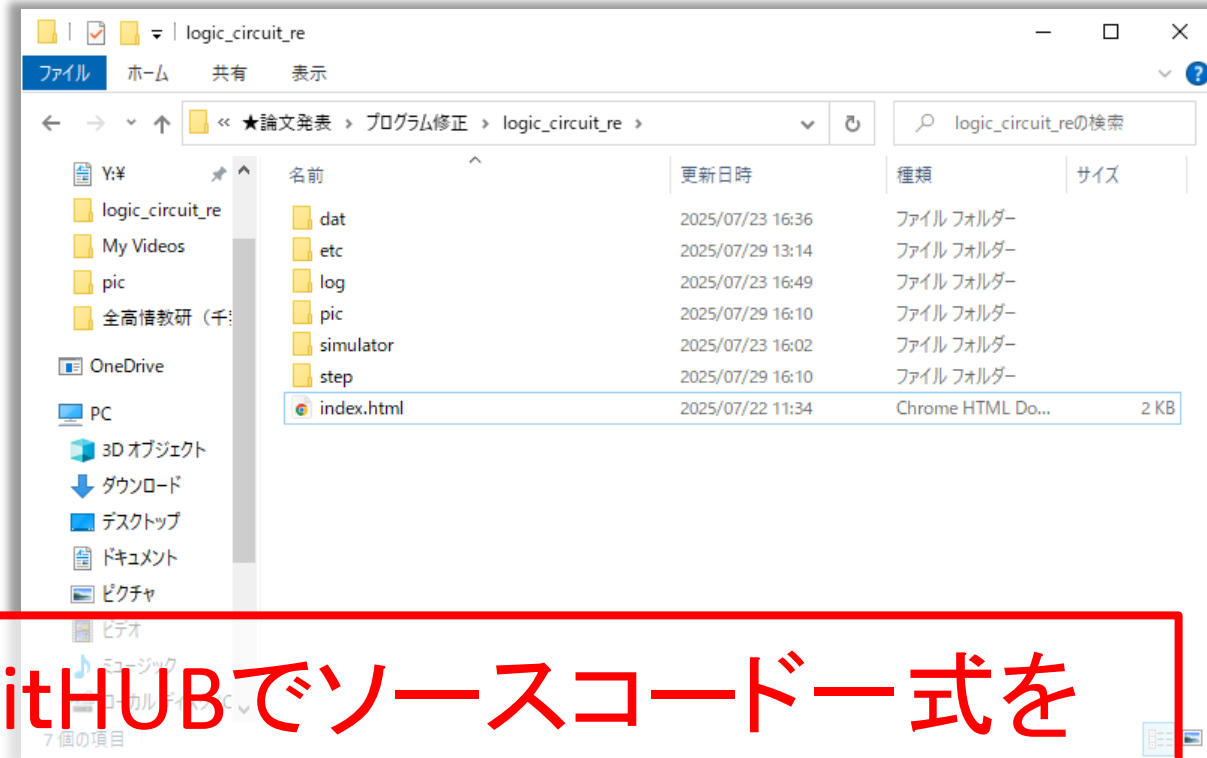
Webページ動作



ウェブブラウザで動作



ウェブブラウザで動作



GitHubでソースコード一式を
ダウンロードして配置するだけ
で、インターネット接続環境が
無くても動作します。



ソースコード(GitHUB)

https://github.com/nagoya-kobayashi/logic_circuit



Webページ動作

https://nagoya-kobayashi.github.io/logic_circuit/

JavaScriptの論理回路シミュレータ

STEP1:チュートリアル①

192.168.30.1/教師間共有/50_NW研修/★論文発表/プログラム修正/logic_circuit_re/step/step01/index....

おすすめサイト IE ブックマーク solar-monitor Google Yahoo!ニュース Active! mail google edu管理

チュートリアル① : AND、OR、NOTの論理ゲートを試してみよう！

入力1 入力2 入力3

出力1 出力2

導線	AND	OR	NOT																																													
電気が流れる。つなぐことで、分岐もできる。	2本の導線から電気が流れてきたら、後の1本の導線に電気が流れる。	2本の導線のうち、一方から電気が流れてきたら、後の1本の導線に電気が流れる。	導線から電気が流れてこなかったら、後の1本の導線に電気が流れる。																																													
<table><thead><tr><th>①</th><th>②</th><th>③</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	①	②	③	0	0	0	1	1	1	<table><thead><tr><th>①</th><th>②</th><th>③</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	①	②	③	0	0	0	0	1	0	1	0	0	1	1	1	<table><thead><tr><th>①</th><th>②</th><th>③</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	①	②	③	0	0	0	0	1	1	1	0	1	1	1	1	<table><thead><tr><th>①</th><th>②</th></tr></thead><tbody><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></tbody></table>	①	②	0	1	1	0
①	②	③																																														
0	0	0																																														
1	1	1																																														
①	②	③																																														
0	0	0																																														
0	1	0																																														
1	0	0																																														
1	1	1																																														
①	②	③																																														
0	0	0																																														
0	1	1																																														
1	0	1																																														
1	1	1																																														
①	②																																															
0	1																																															
1	0																																															

[つぎへ](#)

シンプルな操作(ルール)とゴール

STEP2:半加算器(XOR無し)

192.168.30.1/教師間共有/50_NW研修/★論文発表/プログラム修正/logic_circuit_re/step/step02/index....

おすすめサイト IE ブックマーク solar-monitor Google Yahoo!ニュース Active! mail google edu管理

すべてのブックマーク

【Lv.1】 2進数の足し算を行う回路を作ってみよう。

$0_{(2)} + 0_{(2)}$ $=00_{(2)}$

$0_{(2)} + 1_{(2)}$ $=01_{(2)}$

$1_{(2)} + 0_{(2)}$ $=01_{(2)}$

$1_{(2)} + 1_{(2)}$ $=10_{(2)}$

チェック

[まえへ](#) [つぎへ](#)

XORで難易度を下げ「できる！」を体験

STEP3:チュートリアル②(XOR)

192.168.30.1/教師間共有/50_NW研修/★論文発表/プログラム修正/logic_circuit_re/step/step03/index....

おすすめサイト IE ブックマーク solar-monitor Google Yahoo!ニュース Active! mail google edu管理

すべてのブックマーク

チュートリアル②：XORの論理ゲートを試してみよう！

- 「入力1」～「入力3」の電池をクリックすると電気信号が流れるよ。
- 「？」の部分をクリックすると論理ゲートが変わるよ。

入力1 入力2 入力3

出力1 出力2

導線	AND	OR	NOT	XOR																																																												
電気が流れる。つなぐことで、分岐もできる。	2本の導線から電気が流れてきたら、後の1本の導線に電気が流れる。	2本の導線のうち、一方から電気が流れてきたら、後の1本の導線に電気が流れる。	導線から電気が流れてこなかったら、後の1本の導線に電気が流れる。	2本の導線のうち、1本だけから電気が流れてきたら、後の1本の導線に電気が流れる。																																																												
<table><thead><tr><th>①</th><th>②</th><th>③</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	①	②	③	0	0	0	1	1	1	<table><thead><tr><th>①</th><th>②</th><th>③</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	①	②	③	0	0	0	0	1	0	1	0	0	1	1	1	<table><thead><tr><th>①</th><th>②</th><th>③</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	①	②	③	0	0	0	0	1	1	1	0	1	1	1	1	<table><thead><tr><th>①</th><th>②</th></tr></thead><tbody><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></tbody></table>	①	②	0	1	1	0	<table><thead><tr><th>①</th><th>②</th><th>③</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></tbody></table>	①	②	③	0	0	0	0	1	1	1	0	1	1	1	0
①	②	③																																																														
0	0	0																																																														
1	1	1																																																														
①	②	③																																																														
0	0	0																																																														
0	1	0																																																														
1	0	0																																																														
1	1	1																																																														
①	②	③																																																														
0	0	0																																																														
0	1	1																																																														
1	0	1																																																														
1	1	1																																																														
①	②																																																															
0	1																																																															
1	0																																																															
①	②	③																																																														
0	0	0																																																														
0	1	1																																																														
1	0	1																																																														
1	1	0																																																														

[まえへ](#) [つぎへ](#)

XORで難易度を下げ「できる！」を体験

STEP4:半加算器

192.168.30.1/教師間共有/50_NW研修/★論文発表/プログラム修正/logic_circuit_re/step/step04/index.html

【Lv.1】 (version2) 2進数 1桁の足し算を行う回路を作ってみよう。

- ・右側のように、入力に応じて2進数の足し算の結果が出力されるように論理ゲートの組み合わせを切り替えてみよう。
- ・すべての入力パターンで、意図した出力が出るようにできたら成功！次のレベルに進もう！

2進数: $a[0]+b[0]=cs[00]$
10進数: $a[0]+b[0]=cs[0]$

チェック

入力A	入力B	出力C	出力S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

チェック機能と正解画像で誤りを確認

STEP4:半加算器

192.168.30.1/教師間共有/50_NW研修/★論文発表/プログラム修正/logic_circuit_re/step/step04/index.html

【Lv.1】 (version2) 2進数 1桁の足し算を行う回路を作ってみよう。

- ・右側のように、入力に応じて2進数の足し算の結果が出力されるように論理ゲートの組み合わせを切り替えてみよう。
- ・すべての入力パターンで、意図した出力が出るようにできたら成功！次のレベルに進もう！

2進数: $a[0]+b[1]=cs[00]$
10進数: $a[0]+b[1]=cs[0]$

入力A 入力B

出力C 出力S

0₍₂₎+0₍₂₎ 0₍₂₎+1₍₂₎ 1₍₂₎+0₍₂₎ 1₍₂₎+1₍₂₎

=00₍₂₎ =01₍₂₎ =01₍₂₎ =10₍₂₎

チェック

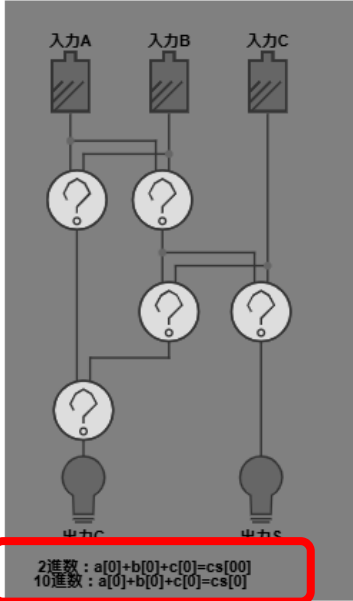
加算器としての意味を画面上で確認

STEP5:全加算器

192.168.30.1/教師間共有/50_NW研修/★論文発表/プログラム修正/logic_circuit_re/step/step05/index.html

【Lv.2】 3つの2進数の足し算を行う回路を作ってみよう。

- ・右側のように、入力に応じて2進数の足し算の結果が出力されるように論理ゲートの組み合わせを切り替えてみよう。
- ・すべての入力パターンで、意図した出力が出るようにできれば成功！次のレベルに進もう！



2進数: $a[0] + b[0] + c[0] = cs[00]$
10進数: $a[0] + b[0] + c[0] = cs[0]$

入力A	入力B	入力C	出力C	出力S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
0	0	1	0	1
1	1	0	1	0
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

チェック

ヒント

- ①【入力A】 + 【入力B】 の"足し算"をする。
- ②その【結果の1桁目のビット】と【入力C】を"足し算"をする。

※"足し算"はLv1で組んだように、ANDとXORで作れる。

③桁上がりとなる【出力C】は"どちらか"の足し算で発生していれば1になる。

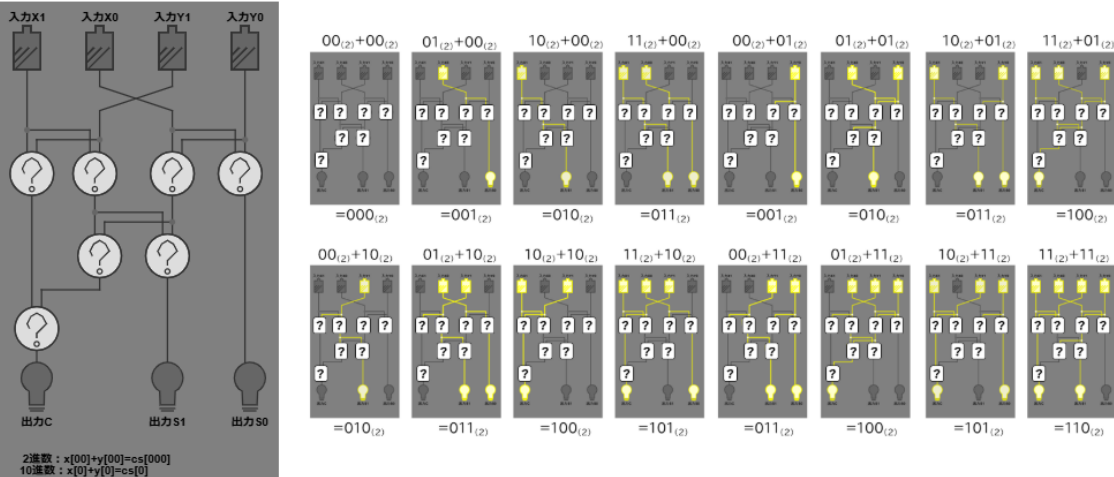
ヒントを確認して設計の意味をつかむ

STEP6:加算器(2桁)

192.168.30.1/教師間共有/50_NW研修/★論文発表/プログラム修正/logic_circuit_re/step/step06/index.html

【Lv.3】 2進数2桁の足し算を行う回路を作ってみよう。

・右側のように、入力に応じて2進数の足し算の結果が出力されるように論理ゲートの組み合わせを切り替えてみよう。
・すべての入力パターンで、意図した出力が出るようにできたら成功！CPU発明家の称号をGETだ！



2進数: $x[0] + y[0] = s[0]$
10進数: $x[0] + y[0] = cs[0]$

チェック

ヒント

(1)出力S0の計算
まずは【入力X0】 + 【入力Y0】の"足し算"をする部分を作ろう。
※2ビットの"足し算"はLv1で組んだように、ANDとXORで作れるね。

(2)出力S1の計算
その【結果の桁上りのビット】と【入力X1】と【入力Y1】を"足し算"をする部分を作ろう。
※3ビットの"足し算"はLv2で組んだように作れるね。
①まずは【入力X1】 + 【入力Y1】の"足し算"をする部分を作ろう。
②その【結果の1桁目のビット】と【(1)の桁上がり】を"足し算"をする部分を作ろう。
※"足し算"はLv1で組んだように、ANDとXORで作れるね。
③桁上がりとなる【出力C】は"どちらか"の足し算で発生していれば1になるから、残りの部分は・・・)

最終ステップはノーヒント！

STEP7:加算器(4桁)

192.168.30.1/教師間共有/50_NW研修/★論文発表/プログラム修正/logic_circuit_re/step/step07/index.html

おすすめサイト IE ブックマーク solar-monitor Google Yahoo!ニュース Active! mail google edu管理

すべてのブックマーク

【Lv.Max】 2進数4桁の足し算をする回路を作ってみよう。

2進数 : $x[0000] + y[0000] = cs[00000]$
10進数 : $x[0] + y[0] = cs[0]$

チェック

良ければ、やってみてください

Webページ動作



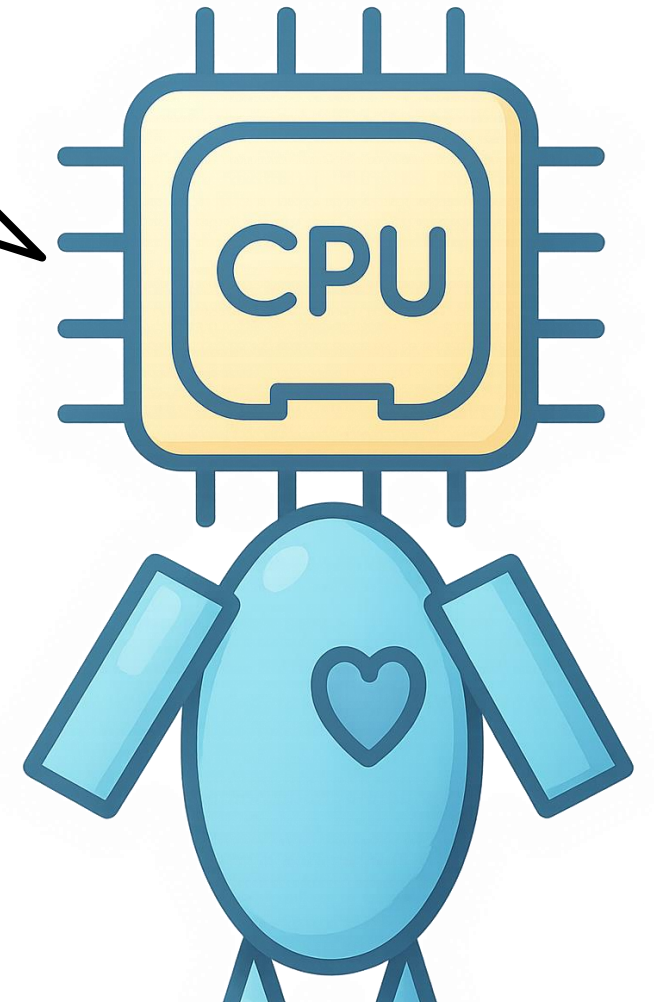
https://nagoya-kobayashi.github.io/logic_circuit/

タブレットやスマートフォンでも動作します。
よければ動かしながらご聴講ください。

論理回路の設計を体験できる ブラウザ学習教材の開発と実践

- 背景・目的
- 授業の内容
- ブラウザ学習教材
- 授業実践の結果
- まとめ

よろしく
お願いいたします



授業実践

- 対象：M高等学校1年生6クラス(233名)
- 期間：2025/7/9(水)～7/14(月)
(※1学期期末考査後～夏休み前)
- 授業場所：一般的なPC教室
- 授業時間：45分
- 評価方法：①操作ログの分析
②授業後アンケート

操作ログ収集

- 以下のタイミングで操作ログを出力

入力の切替

論理ゲートの切替

チェック押下/クリア時

【操作ログファイル (CSV) 仕様】
操作者ID、ステップ番号、タイムスタンプ、イベント種別、対象オブジェクトID、前状態、後状態

操作ログの分析結果

- ステップ毎の達成率
- ステップ毎の平均クリック頻度
- ステップ毎の論理ゲート辺りの誤切替回数
- ステップ達成までの論理ゲート辺りの切替回数
- ステップ達成までの所要時間
- ステップ達成までのチェック回数

ブラウザを途中で閉じる等でステップに手戻りが生じた生徒のログは除外し、223 名を対象に分析を実施した。

ステップ毎の達成率

課題	入力数	ゲート数	達成者数	達成率
ステップ1: チュートリアル	2	2	-	-
ステップ2: 半加算機設計	2	4	134	60.1%
ステップ3: チュートリアル(XOR)	2	2	-	-
ステップ4: 半加算機設計(XOR有)	2	2	185	83.0%
ステップ5: 全加算機設計	3	5	171	76.7%
ステップ6: 2ビット加算回路実装	4	7	173	77.6%
ステップ7: 4ビット加算回路実装	8	17	77	34.5%

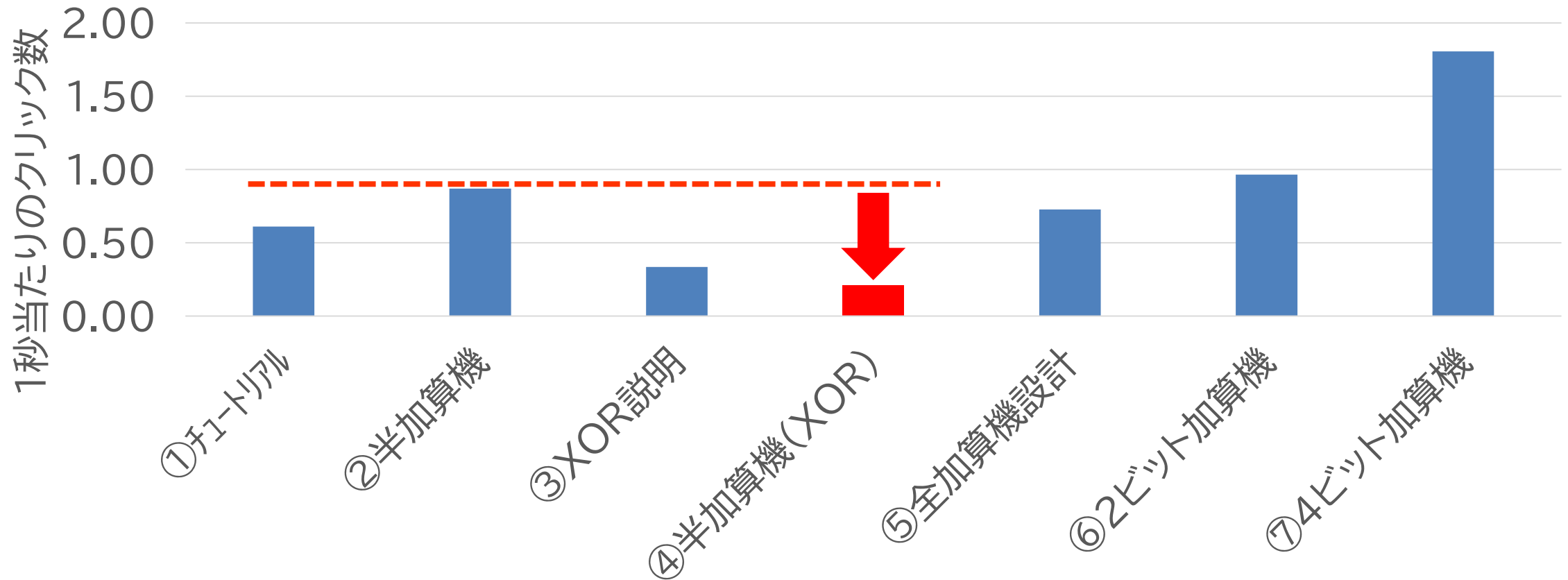
- ステップ6の正解までは授業進捗に合わせ正解も示しながら実施したが、理解が追いつかなかった生徒が22.4%(100%-77.6%)いたことが分かる。⇒授業スピードをもう少しゆっくりできると良い
- ステップ7はノーヒントで実施し34.5%がクリアしている。一定数は論理回路の設計能力を醸成できたといえる。

ステップ毎の達成率

課題	入力数	ゲート数	達成者数	達成率
ステップ1: チュートリアル	2	2	-	-
ステップ2: 半加算機設計	2	4	134	60.1%
ステップ3: チュートリアル(XOR)	2	2	-	-
ステップ4: 半加算機設計(XOR有)	2	2	185	83.0%
ステップ5: 全加算機設計	3	5	171	76.7%
ステップ6: 2ビット加算回路実装	4	7	173	77.6%
ステップ7: 4ビット加算回路実装	8	17	77	34.5%

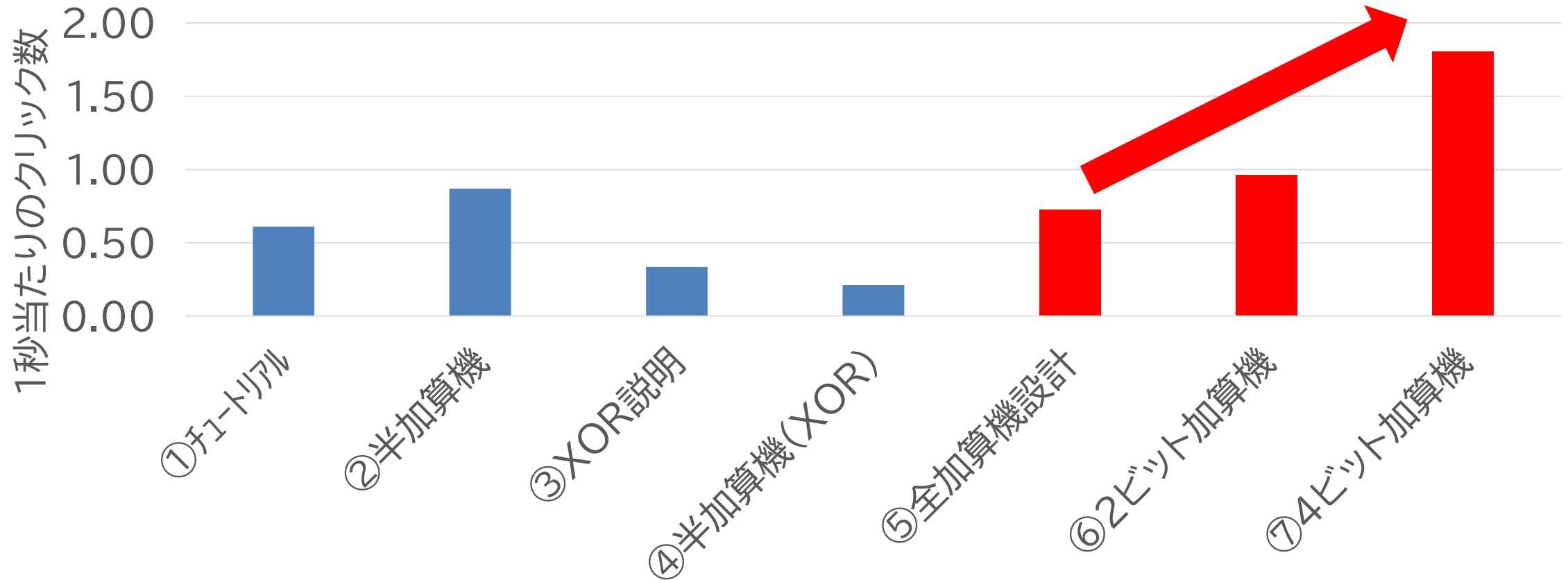
- ステップ6の正解までは授業進捗に合わせ正解も示しながら実施したが、理解が追いつかなかった生徒が22.4%(100%-77.6%)いたことが分かる。⇒授業スピードをもう少しゆっくりできると良い
- ステップ7はノーヒントで実施し34.5%がクリアしている。一定数は論理回路の設計能力を醸成できたといえる。

ステップ毎の平均クリック頻度



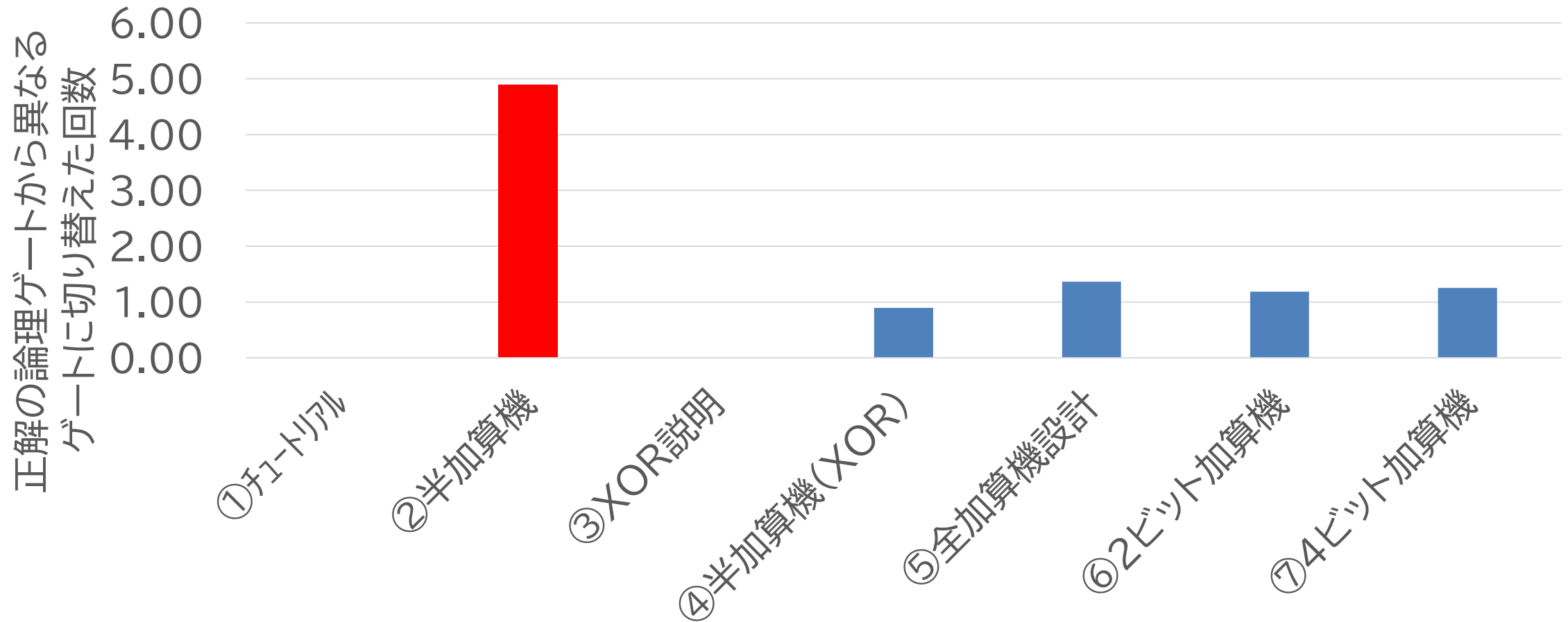
- ステップ1と2よりステップ4は下がっている。正解を導くためにどうすればどう操作すれば良いかを検討し始めたことが示唆される。
- ステップ5以降では徐々にクリック頻度は高くなっている。回路設計課題の取り組み要領が理解できていったことを示唆している。

ステップ毎の平均クリック頻度



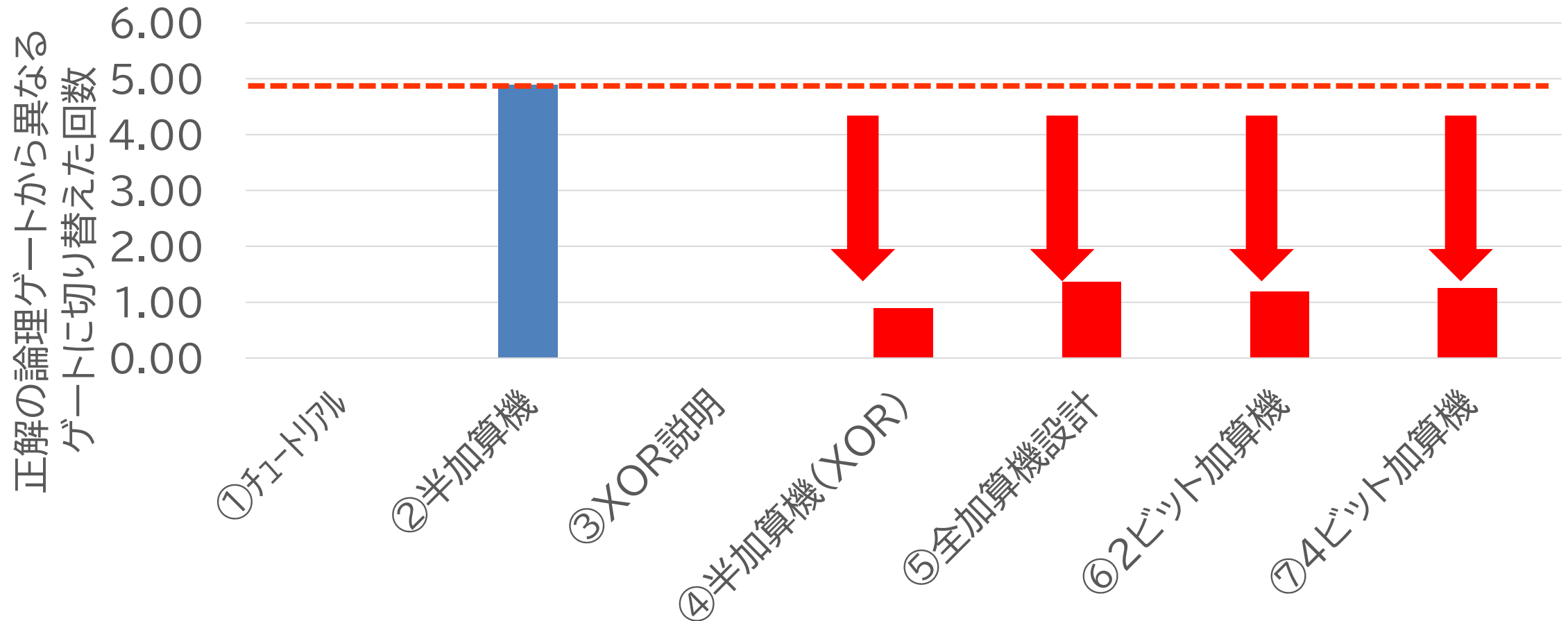
- ステップ1と2よりステップ4は下がっている。正解を導くためにどうすればどう操作すれば良いかを検討し始めたことが示唆される。
- ステップ5以降では徐々にクリック頻度は高くなっている。回路設計課題の取り組み要領が理解できていったことを示唆している。

ステップ毎の論理ゲートあたりの誤切替回数



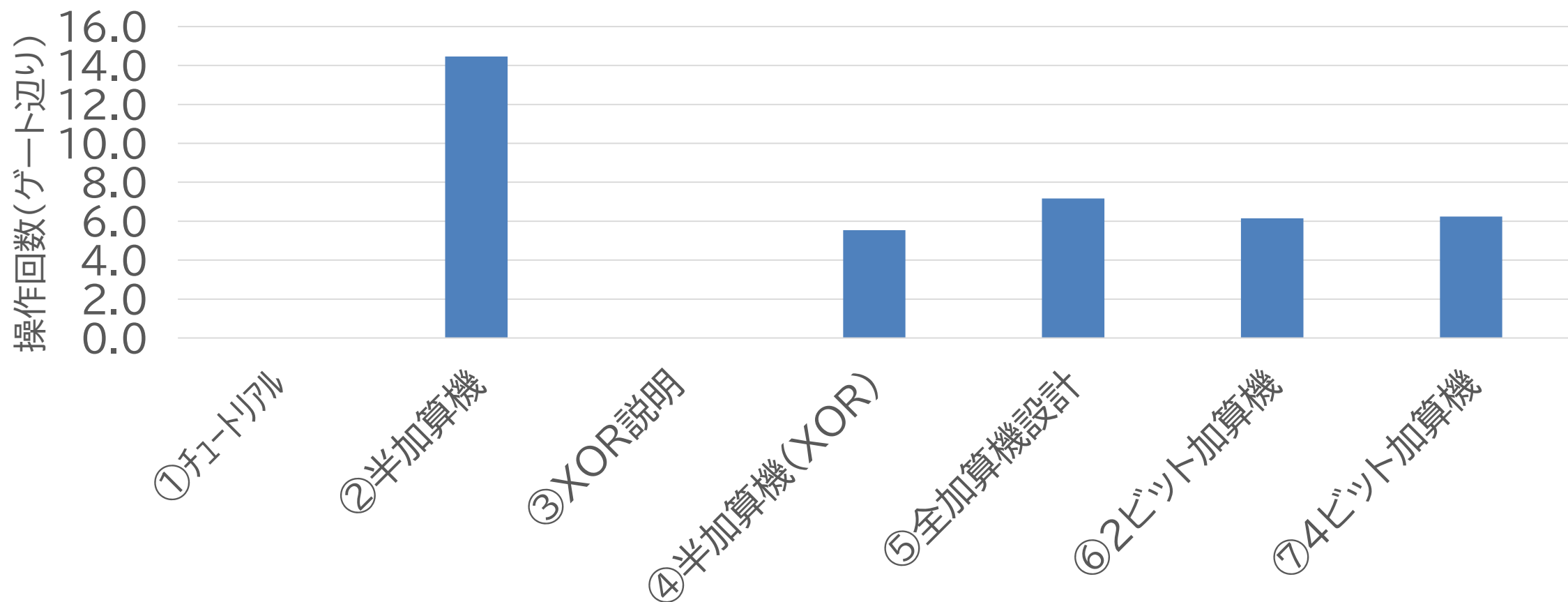
- ステップ2では4.89回⇒無作為な操作により正解を導こうとした生徒が一定数居たことが推察できる。
- ステップ4以降では誤切替回数が減少⇒回路構造が複雑になり、無作為な試行でなく、加算器の特性をしっかりと考えて操作を行うようになったと考えられる。

ステップ毎の論理ゲートあたりの誤切替回数



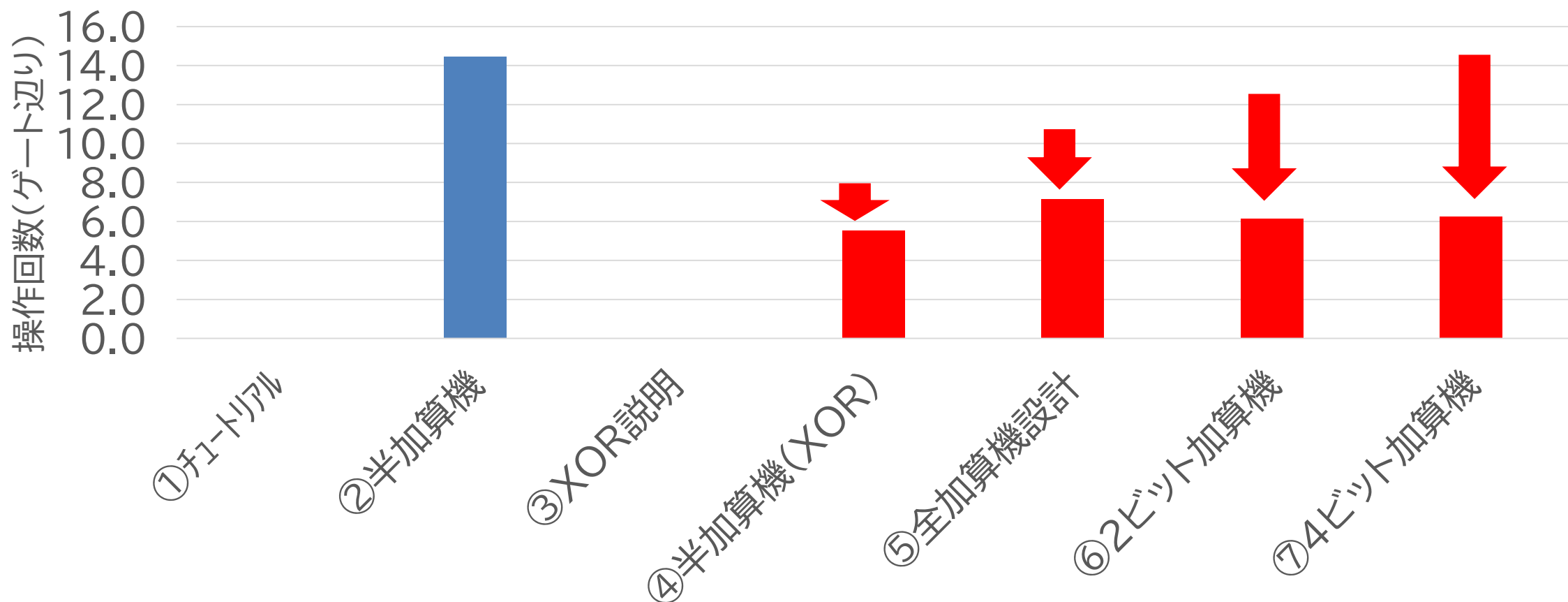
- ステップ2では4.89回⇒無作為な操作により正解を導こうとした生徒が一定数居たことが推察できる。
- ステップ4以降では誤切替回数が減少⇒回路構造が複雑になり、**無作為な試行**でなく、**加算器の特性をしっかりと考えて操作**を行うようになったと考えられる。

ステップ達成までの論理ゲートあたりの切替回数



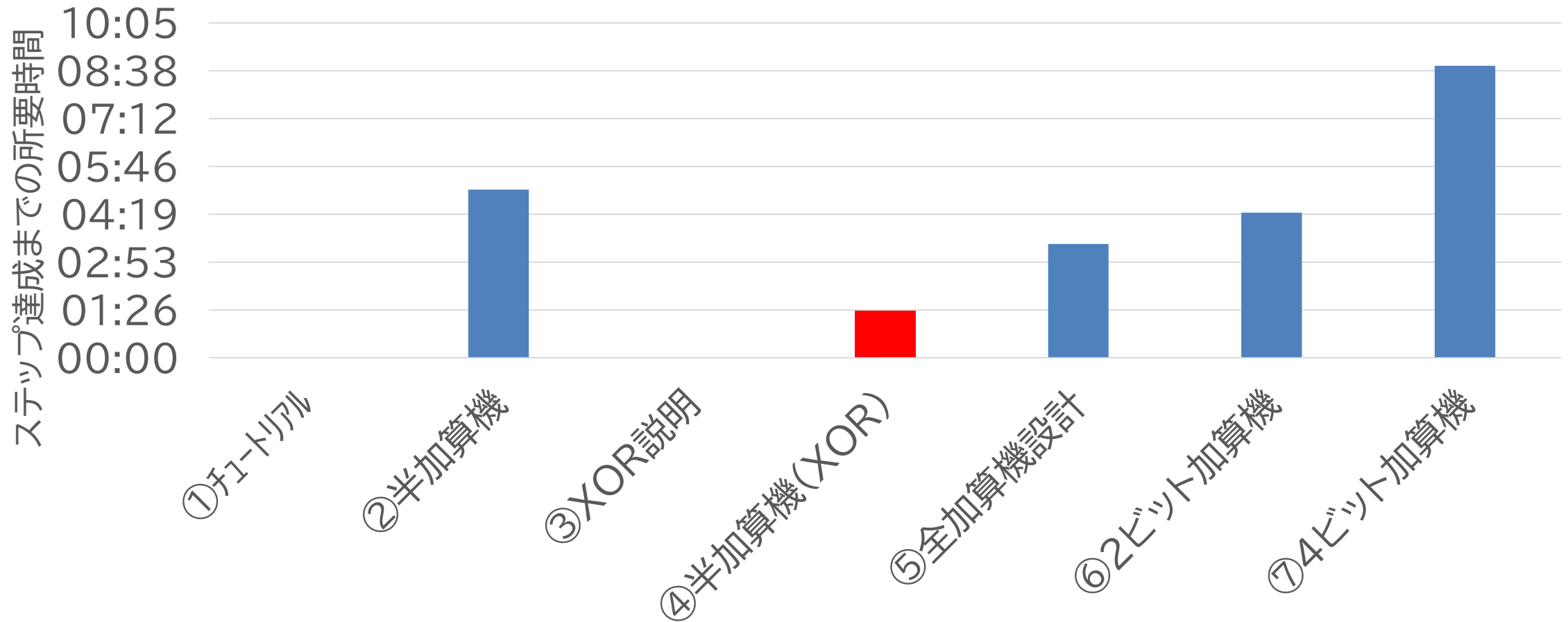
- ステップ課題の達成者が、ステップ達成までに行ったゲート1つあたりの切替回数
- 回路構成が複雑化することに伴い増加傾向が見られると想定していたが、実際には後半のステップでも大きな変動は無かった。
⇒ステップを進めた生徒は加算回路の構造を理解しながら必要最小限の切替に留めていたと考えられる。

ステップ達成までの論理ゲートあたりの切替回数



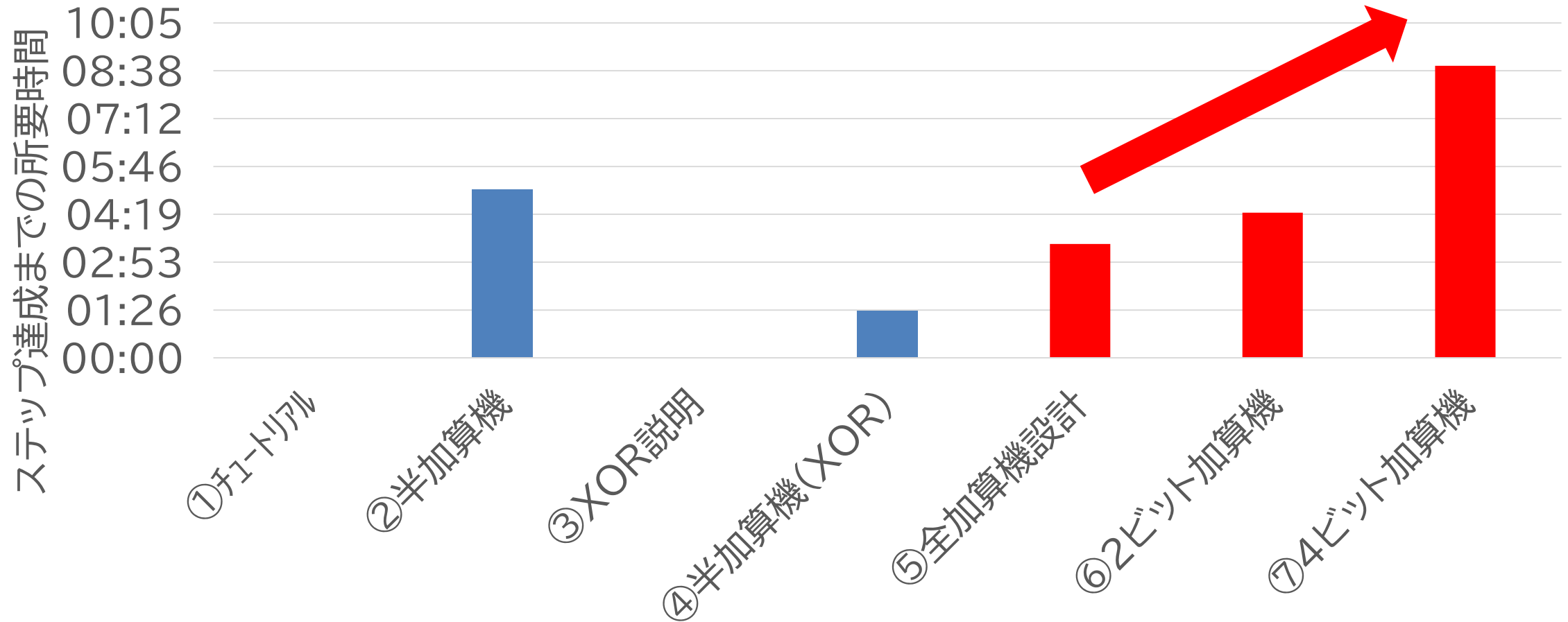
- ステップ課題の達成者が、ステップ達成までに行ったゲート1つあたりの切替回数
- 回路構成が複雑化することに伴い増加傾向が見られると想定していたが、実際には後半のステップでも大きな変動は無かった。
⇒ステップを進めた生徒は加算回路の構造を理解しながら必要最小限の切替に留めていたと考えられる。

ステップ達成までの所要時間



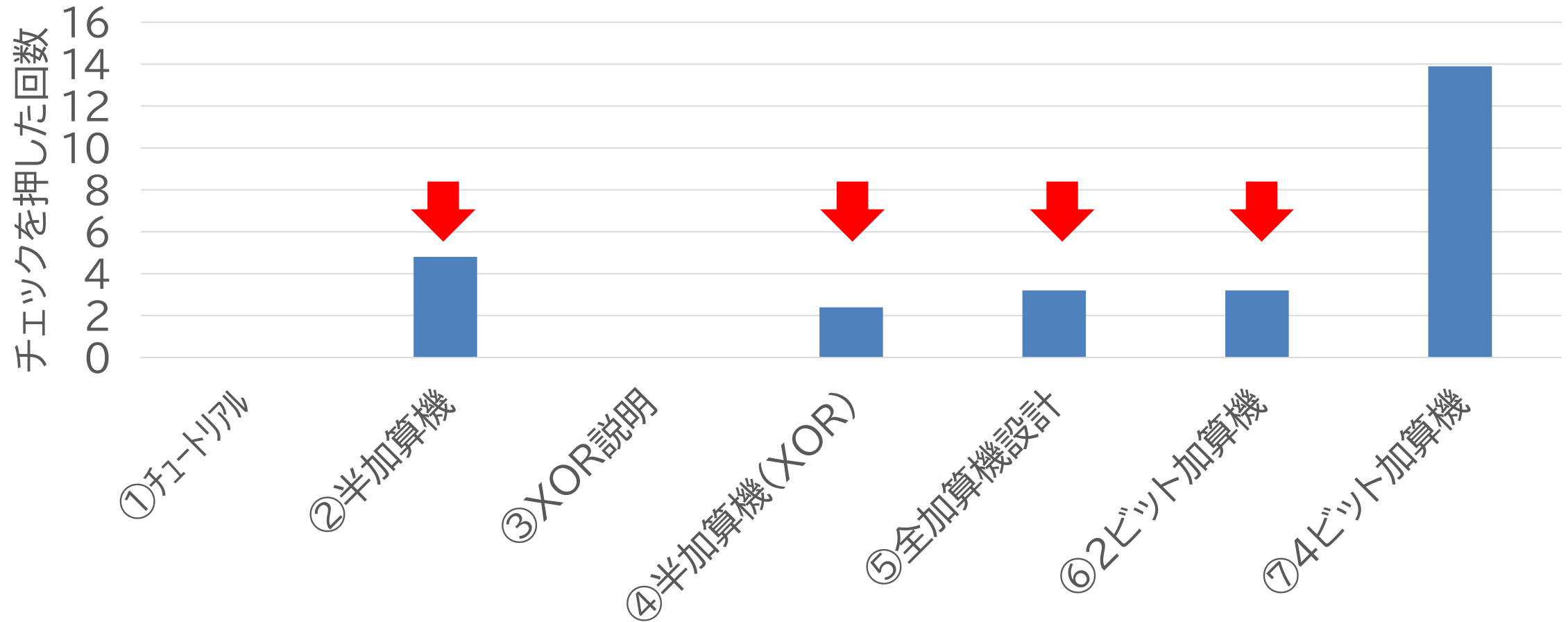
- ステップ4で最も短い⇒半加算機の実装経験があり、かつ、XORを追加したことで回路構成が簡略化が要因と考えられる。
- ステップ5以降では回路規模と複雑度の増加に伴って、所要時間の増加傾向が見られた。

ステップ達成までの所要時間



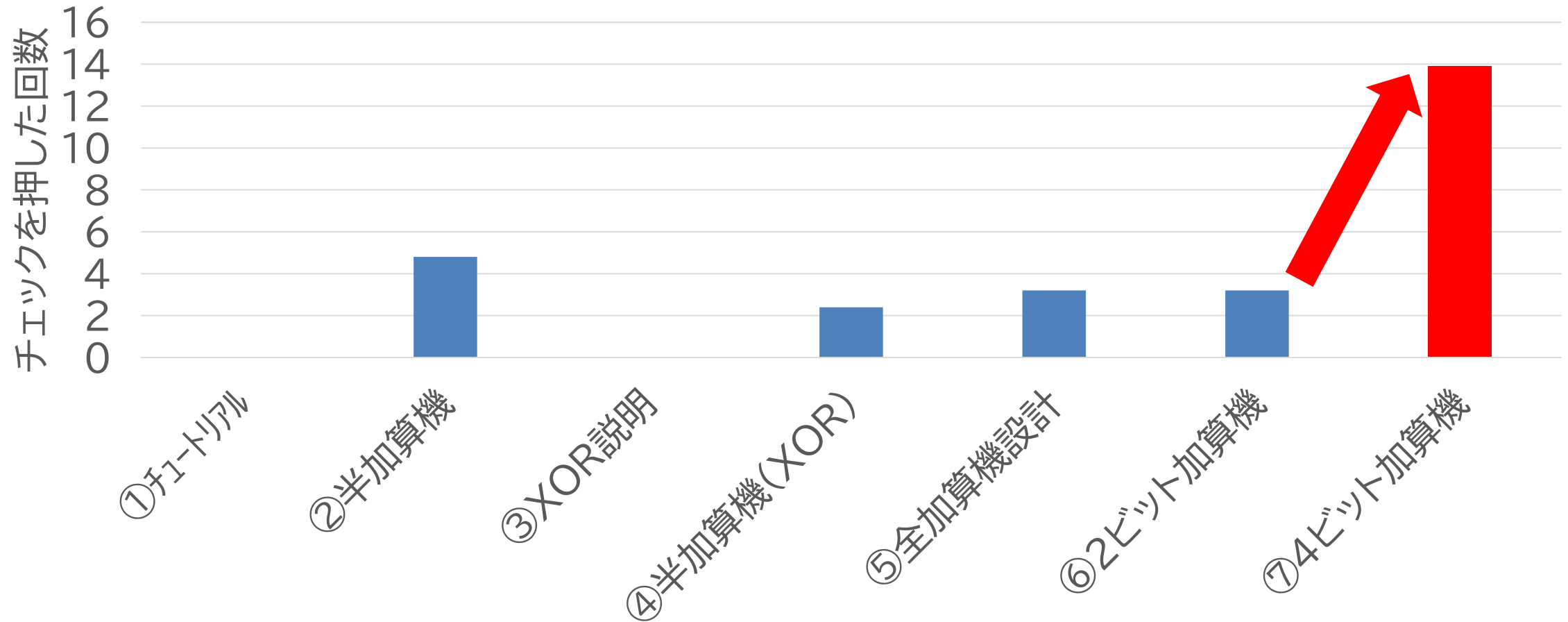
- ステップ4で最も短い⇒半加算機の実装経験があり、かつ、XORを追加したことで回路構成が簡略化が要因と考えられる。
- ステップ5以降では回路規模と複雑度の増加に伴って、所要時間の増加傾向が見られた。

ステップ達成までのチェック回数



- ステップ6の正解までは少ない。⇒授業進捗に合わせ正解も示しながら実施したため、**自信をもって完成させてからチェックしていた**と推測される。
- ステップ7ではチェック回数が大幅に増加している。授業でノーヒントとなり、生徒自身で試行錯誤する過程で、こまめにチェックを活用したと思われる。

ステップ達成までのチェック回数



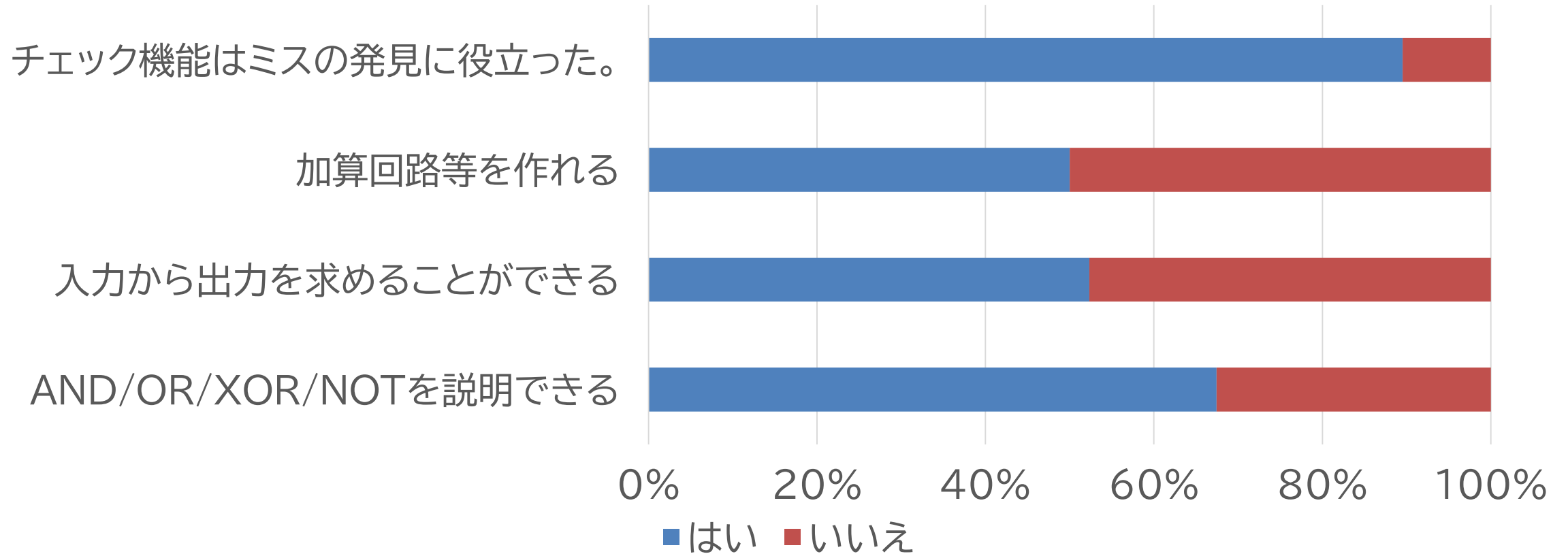
- ステップ6の正解までは少ない。⇒授業進捗に合わせ正解も示しながら実施したため、**自信をもって完成させてからチェックしていた**と推測される。
- ステップ7ではチェック回数が大幅に増加している。授業でノーヒントとなり、**生徒自身で試行錯誤する過程で、こまめにチェックを活用した**と思われる。

操作ログの分析結果

- ステップ毎の達成率
- ステップ毎の平均クリック頻度
- ステップ毎の論理ゲート辺りの誤切替回数
- ステップ達成までの論理ゲート辺りの切替回数
- ステップ達成までの所要時間
- ステップ達成までのチェック回数

⇒生徒が試行錯誤しながら操作し、自分の力で理解し解こうと頑張っていた形跡が確認できた。

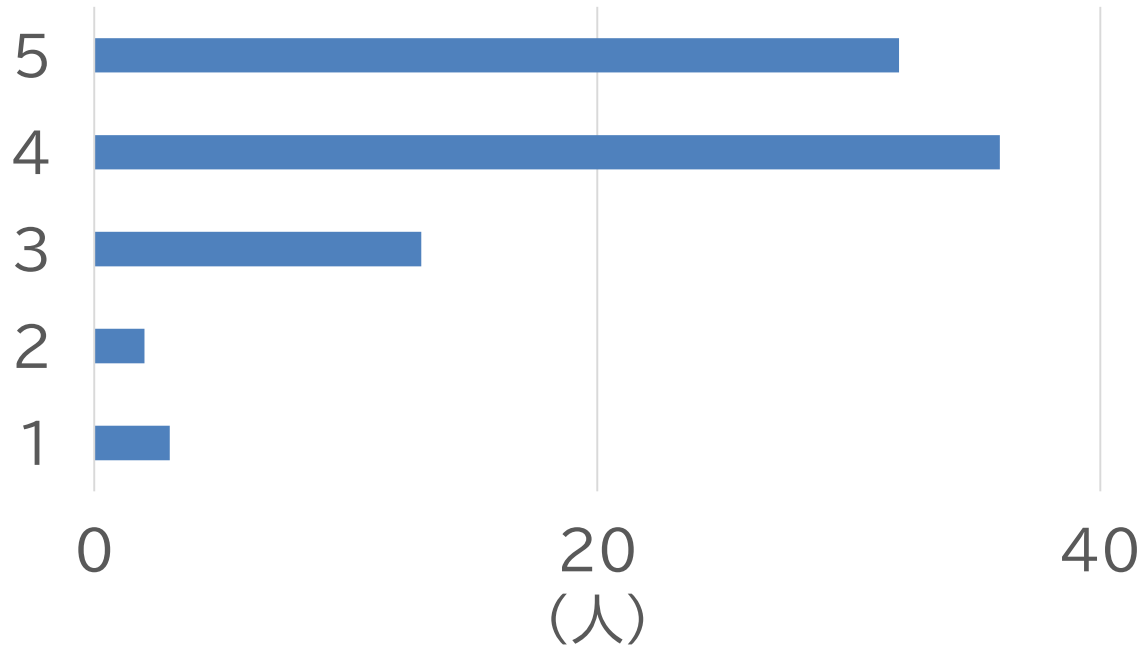
授業後アンケート(理解度)



⇒論理回路をしっかり理解できた生徒は半分程度

授業後アンケート(教材)

操作画面の使いやすさ評価

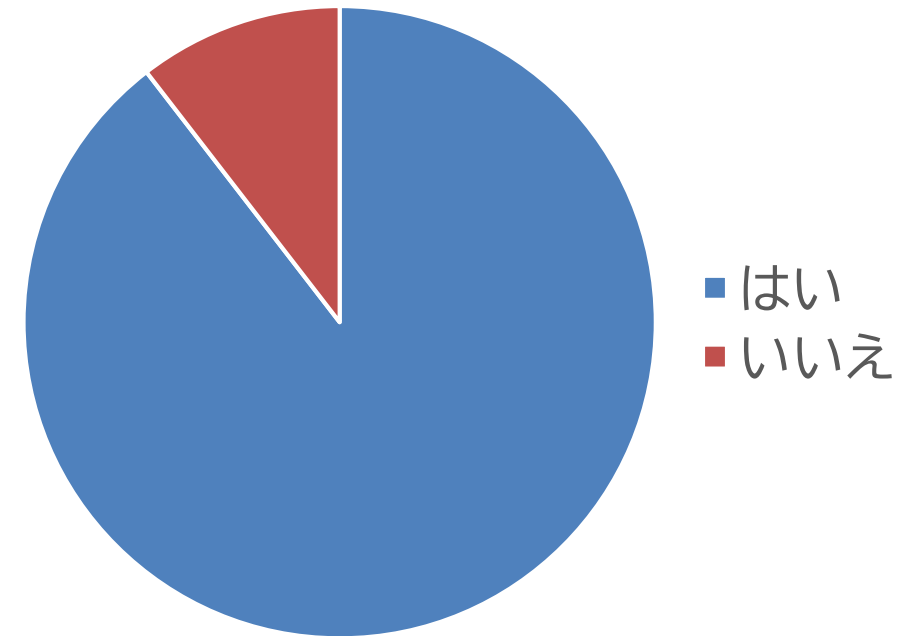


4.10

平均評価

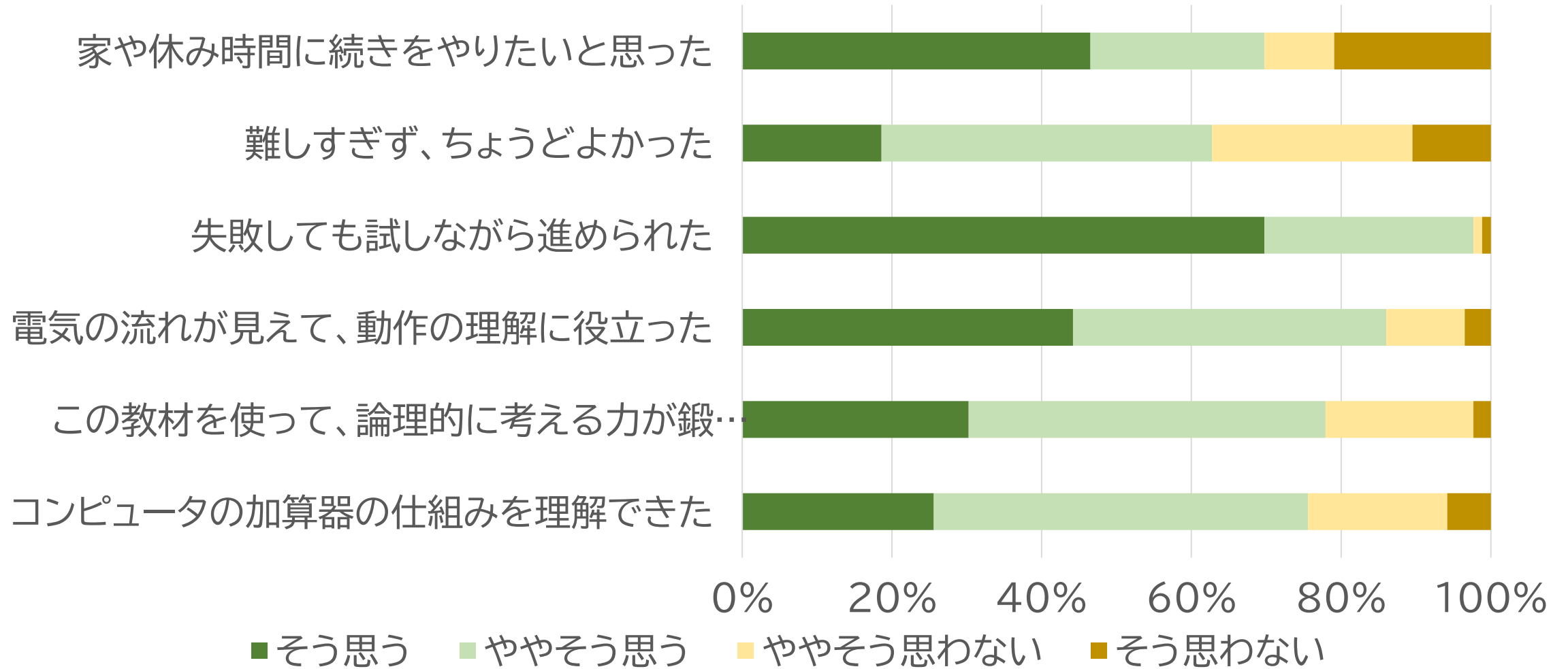


チェック機能はミスの発見に役立った



⇒概ね、好評

授業後アンケート(他)



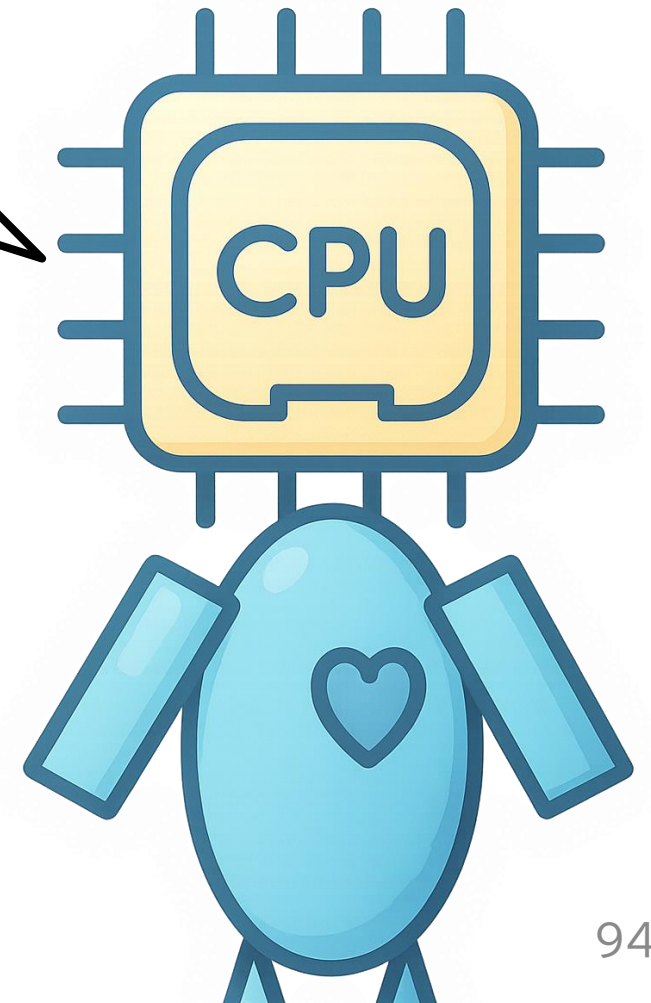
⇒一定数の生徒の興味を引き出すことに成功！

論理回路の設計を体験できる ブラウザ学習教材の開発と実践

- 背景・目的
- 授業の内容
- ブラウザ学習教材
- 授業実践の結果
- まとめ

よろしく
お願いいたします

Webページ動作



まとめ

- 「論理回路」で活用する目的として、HTML+JavaScriptのみで動作する体験教材を開発し、授業で実践した。
 - 操作ログや授業後アンケートで評価を実施した。
 - 教材の難易度や操作性などは好評を得た。
 - 7割程度が基本的な理解を得て、半分程度の生徒が実用的応用力や設計思考を醸成できた。
 - 7割程度の生徒が継続学習に意欲を持った。
- ⇒短時間(1コマ)で論理回路の基本的な理解を達成し、また、応用的な設計への興味を引き出すことに成功した。

今後について

- 授業改善

- 今回の授業で理解が追いつかなかった3割程度の生徒へのフォローも含めた授業改善

- 教材の改良

- 課題エディタの開発

- 新たなステップ課題の追加

- 通信回路、フリップフロップ、ALU全体など