

論理回路(1)

工学基礎実験

この実験の構成

- 1 週目
 - 基本的論理素子の演習

- 2 週目
 - 応用的な論理回路の演習

- 3 週目
 - 実際のハードウェアによる論理回路の実習

• この教材のデータは一部wikipediaのデータ https://en.wikipedia.org/wiki/Logic_gate を使用しています

K5 論理回路の評価について

K5は下記の通り評価される。

- ・ **講義中の課題(10点) ※最初の1～2週目の課題**
- ・ レポートの評価(10点満点での評価、1～10点で評価) ※3週目の課題

レポートは必ず提出すること

(未提出、他の人のレポートを写した者(提供者も含む)は、0点とする)

レポートの内容については、3週目に説明を行う。

- これまで、皆さんが習った一般的な計算(算数や数学)
 - 一般的な計算(**一般的代数**)では、0～9までの数字が出てくる

- 論理回路では、入力・出力ともに「**論理**」になっている
 - 論理なので、1(YES)か0(NO)の2つの値しか、出てこない
 - これらを用いた論理代数 (**ブール代数**)

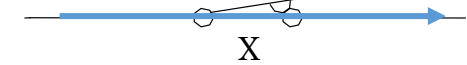
- この講義では、組み合わせ回路を取り扱う
 - 組み合わせ回路：入力が決まれば、出力が決まる回路
 - AND,OR,NOTで構成される
 - 特徴として**同じ入力に対しては、出力は同じ**になる

これから出てくる用語の補足 論理回路の説明の図で出てくる単語と定義

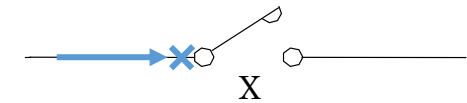
□ 図の中で出てくるものについての補足説明

□ スイッチ(入力)

- スイッチが導通状態(電気が流れる状態) → ON
- スイッチが開放状態(電気が流れない状態) → OFF



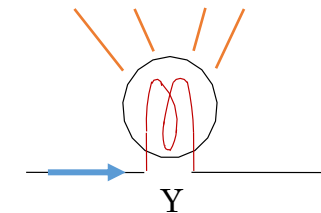
スイッチが導通状態



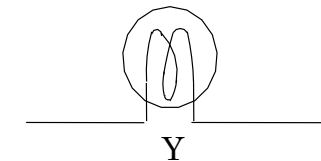
スイッチが開放状態

□ ランプ(出力)

- 電流が流れたとき、ランプが点灯 → ON
- 電流が流れないとき、ランプが消灯 → OFF



電流が流れた



電流が流れていない

- 点灯の有無で、出力の状態を示す

これから出てくる用語の補足 論理回路の説明の図で出てくる単語と定義

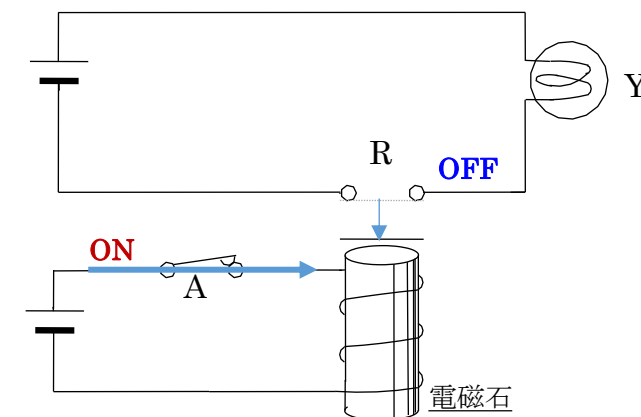
□ リレー

- 電磁石と、金属片のスイッチで構成
- 電磁石とスイッチRが連動

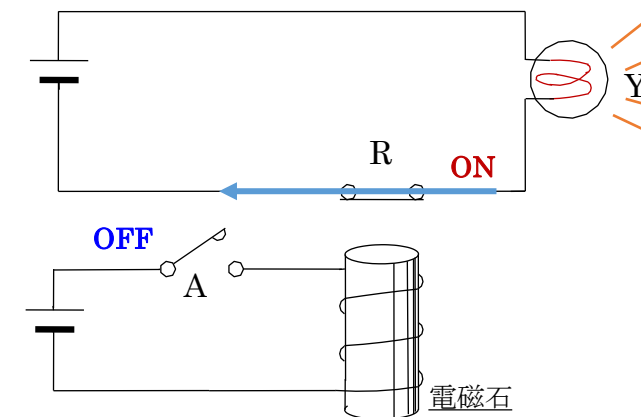
□ リレーの動作

- スイッチAがON(電磁石に電流が流れる)
→ スイッチRがOFF
- スイッチAがOFF(電磁石に電流が流れない)
→ スイッチRがON

□ NC(NormallyClose)のリレー



スイッチAが導通状態



スイッチAが開放状態

□ 論理回路

- 入力・出力がある
- 入力・出力ともに「論理」になっている
 - つまりYESかNOもしくは1 か 0 → これらを用いた論理代数（ブール代数）
 - それらの羅列（00101011 など、ビットパターン）で値(数値など)を表す
 - 回路なので、1, 0を電圧などで表現する
 - 一般的なもの: $0 = 0\text{ V} = \text{Low(No)}$, $1 = 5\text{ V} = \text{High(Yes)}$ など
 - 5 Vの代わりに 3.3 V, 1.8 V の場合も
 - 電圧が低い = ある電圧から別の電圧にすぐに変化出来る → 高速に動作
 - 最近は LVDS (Low Voltage Differential) などが超高速データ転送に使われる

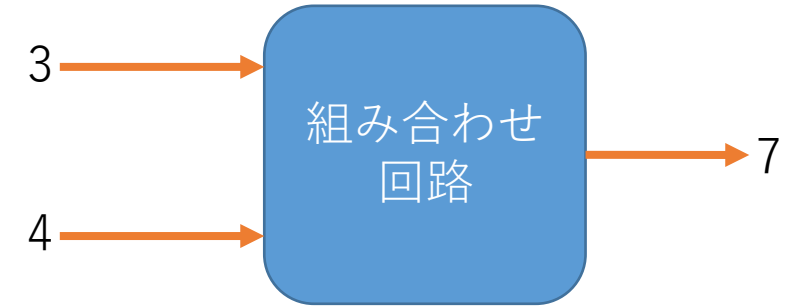
□ 大きく分けて二種類の回路パターン

- 組み合わせ回路(combinational circuit) ← この実験科目で取り扱う
- 順序回路(Sequential Logic) ← 3年の実験科目等で取り扱う(コースによる)

組み合わせ回路と順序回路

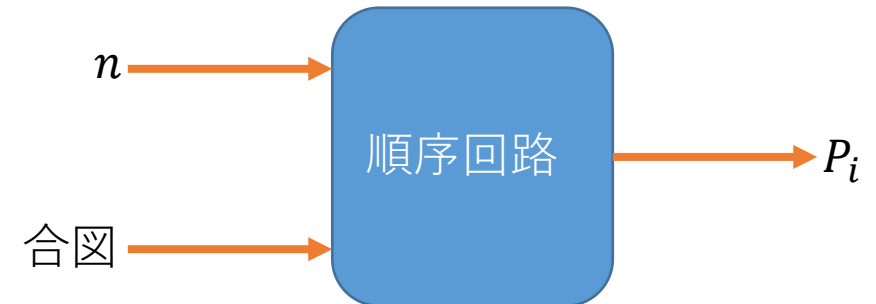
□ 組み合わせ回路

- 入力が決まれば出力が決まる
- 同じ入力なら常に同じ出力
- 例: ただの数学の演算 $3+4 = 7$
- **AND, OR, NOT, つまりブール代数**で構成



□ 順序回路

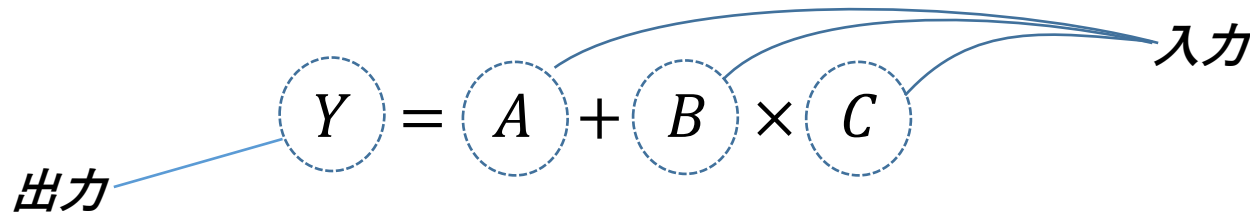
- 入力と、その時の内部の状態に応じて出力が変わる
- 同じ入力でも毎回出力が変わっても良い
- 例: 数列の計算 $P_{i+1} = P_i + n$
- 組み合わせ回路と **フリップフロップ**で構成



□ <https://kotobank.jp> より→

コンピューターの5大機能のひとつ。四則演算、数値の大小を比較する比較演算、論理演算などの計算処理のこと。また、パソコンで演算を行う装置を演算装置と呼ぶ。

□ 四則演算 … 和差積商で入力から出力を得る



- この場合、どの変数も一般的な値(整数・実数等) … 一般的代数
- 論理演算 … 整数や実数ではなくて論理を扱う(0, 1やYES, NO)
 - 和差積商に相当するものが別にある … ブール代数

論理積(AND)

□ スイッチA, B両方をオンにしたときのみ、ランプが点灯

□ この「両方を」→ AND(論理積)の意味

□ スイッチA, Bが論理入力、ランプYが論理出力

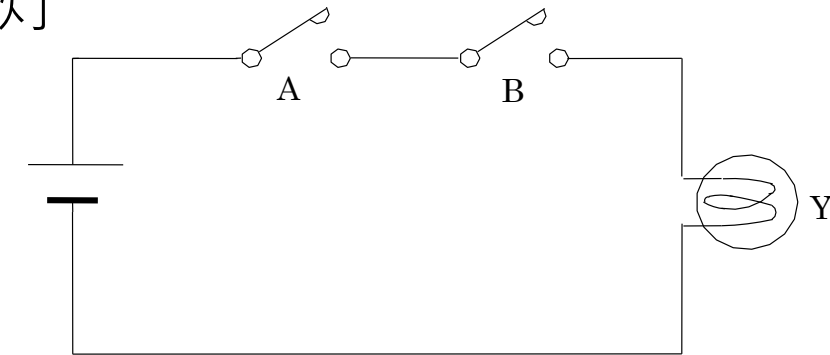
- スイッチオン=1
- スイッチオフ=0
- ランプが点灯=1
- 消灯している=0

□ A, B, Yの値(value) … 真理値(truth value)

□ 真理値をまとめたもの … 真理値表(truth table)

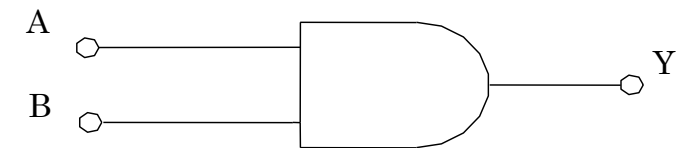
□ 論理式: $Y = A \cdot B$

□ 対応する論理回路: **AND回路**



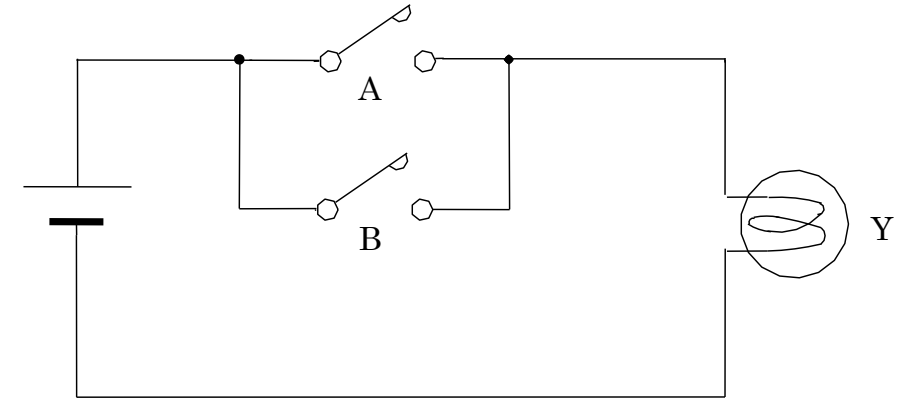
AND回路の真理値表

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

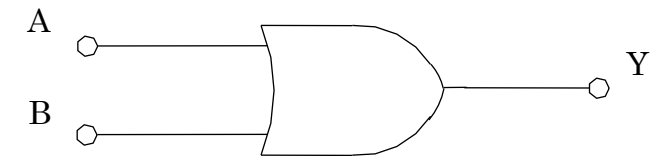


論理和(OR)

- スイッチA, Bが並列な場合
 - A, Bのいずれかが閉じればランプは点灯する
 - この「いずれかが」や「または」というのがOR(論理和)
- 真理値表は一般的な代数の和と一致しない
- 論理式: $Y = A + B$
- 対応する論理回路: **OR回路**

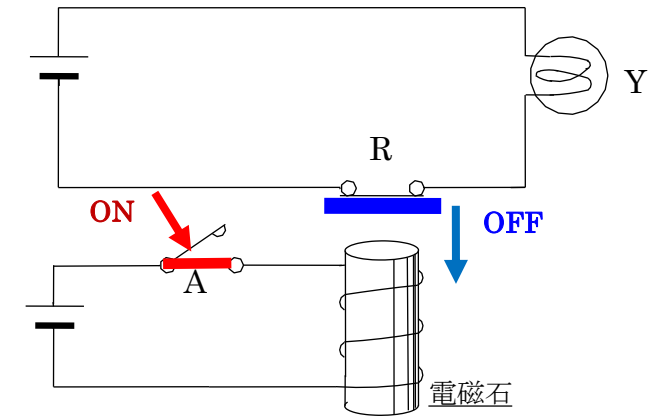


OR回路の真理値表		
A	B	Y
0	0	
1	0	
0	1	
1	1	



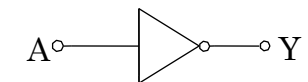
論理否定(NOT)

- リレーを用いた回路を考える
- スイッチAが開いているとき，リレーの接点Rが閉じており，ランプYは点灯している．スイッチを閉じると，リレーが働き，その接点Rは開き，ランプYは消灯
 - 入力: スイッチAの状態
 - 出力: ランプYの状態
- このような論理演算が論理否定(NOT)または否定
- 論理式: $Y = \bar{A}$
- 対応する論理回路: **NOT回路**



NOT回路の真理値表

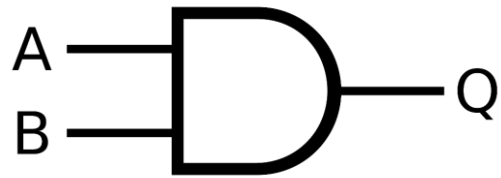
A	Y
0	1
1	0



組み合わせ回路の基本素子まとめ

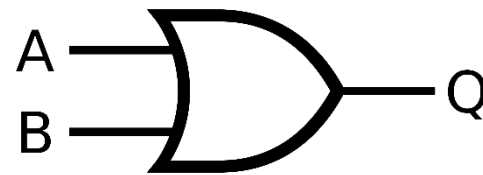
□ 全ての基本はAND, OR, NOT

AND



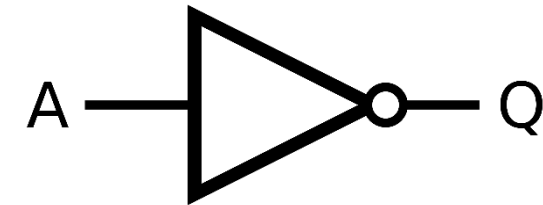
INPUT		OUTPUT
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

OR



INPUT		OUTPUT
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

NOT

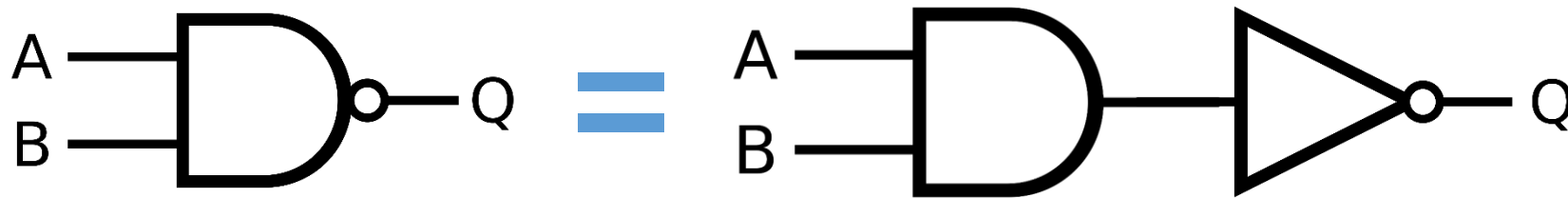


INPUT	OUTPUT
A	Q
0	1
1	0

応用素子(NANDとNOR)

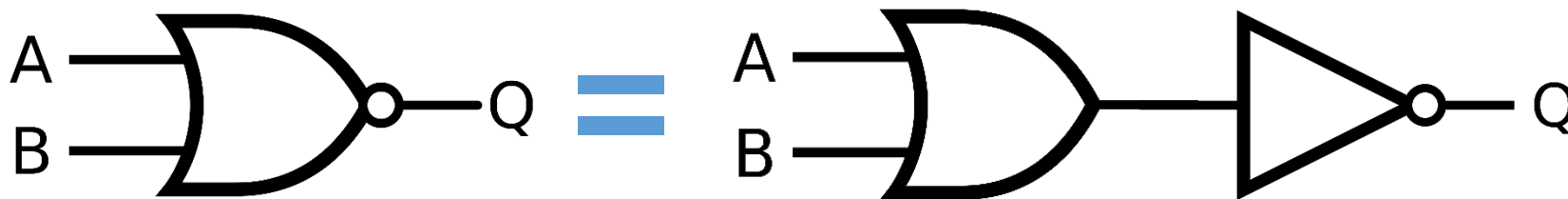
- 記号に○がつくとそこにNOTがあるのと同じ!
- AND, OR以外でも○がつくと常に0, 1が入れ替わると思えばよい

NAND



INPUT		OUTPUT
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

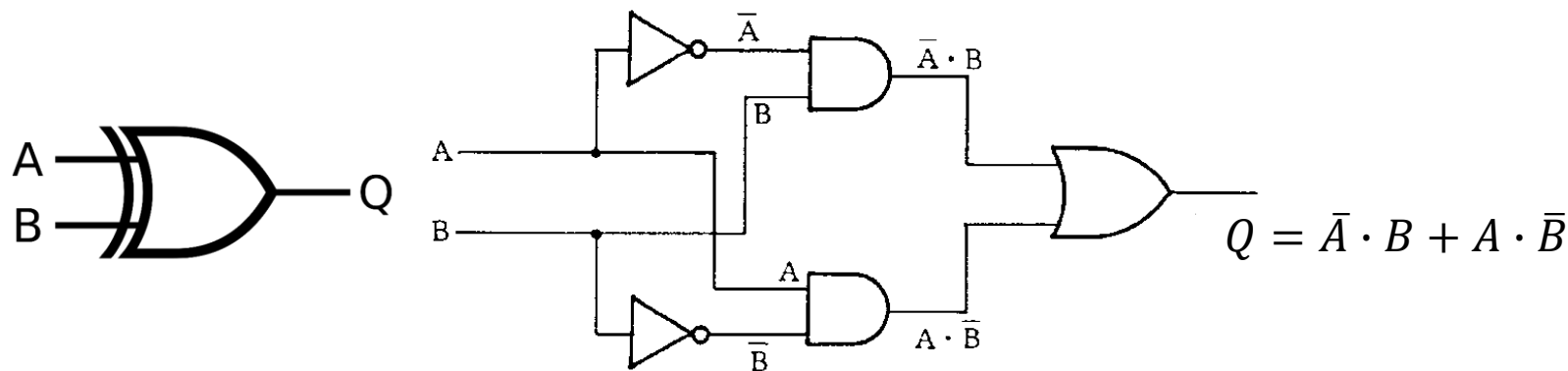
NOR



INPUT		OUTPUT
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

EXOR

- EXOR = Exclusive OR (排他的論理和)
- 排他的 = お互いに退ける、という意味
 - 排他的なOR → 同一の場合は除くという意味
- 入力A, Bのどちらかが1のときのみ出力が1となり、同じ場合は0
- $Q = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B$



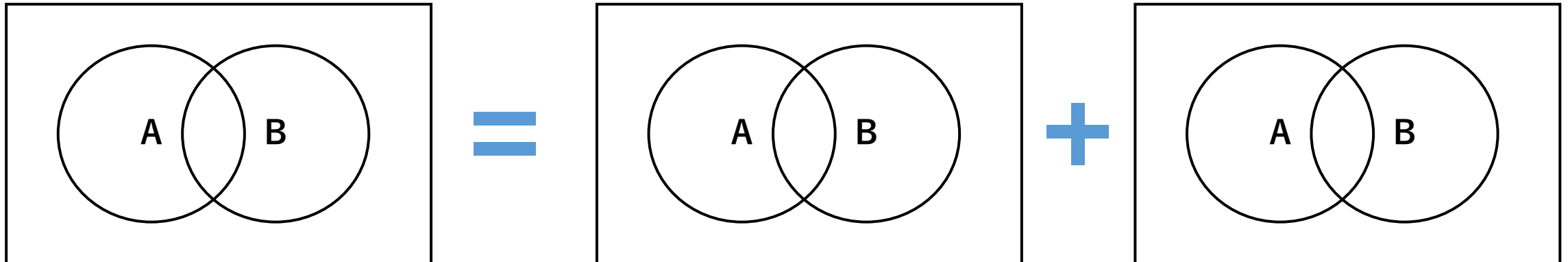
INPUT		OUTPUT
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

□ De Morgan's laws (ド・モルガンの法則) … 二つある

$\overline{A \cdot B} = \bar{A} + \bar{B}$ (1) … 「この動物は哺乳類(A)であり且つ尻尾(B)がある」の否定
= 「この動物は哺乳類(A)ではない、又は尻尾(B)は無い」

$\overline{A + B} = \bar{A} \cdot \bar{B}$ (2) … 「この猫は白猫(A)または黒猫(B)である」の否定
= 「この猫は白猫(A)でもなければ黒猫(B)でもない」

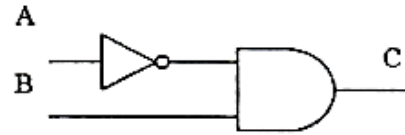
□ ベン図で証明してみよう(各自ノートに)



タイミングチャート

- デジタル回路を設計する場合、動作の時間的流れをつかむため「タイミングチャート(タイムチャート)」が利用される
- 横軸: 時間
- 縦軸: H または L レベル

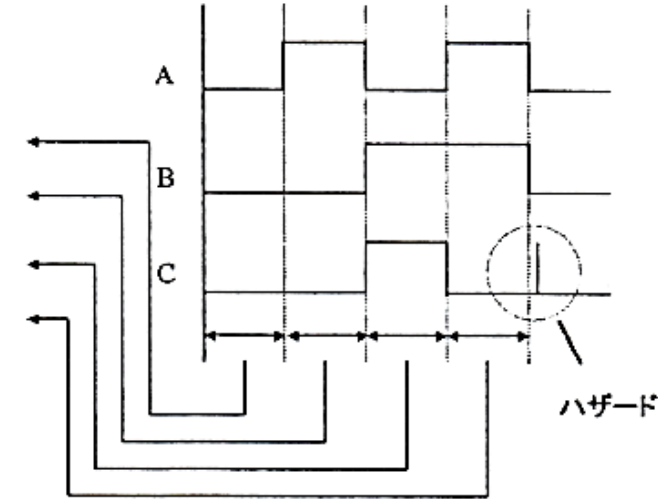
回路図



真理値表

A	B	C
L	L	L
H	L	L
L	H	H
H	H	L

タイムチャート



用方

- 信号の遅延(ディレイ)も考慮して誤動作が起きないかどうか確かめる
- デジタル回路のシミュレーション結果はこのタイムチャートで表示される
- 正確なタイムチャートの作成
→図のような不必要なヒゲ(ハザード)も発見できる。

- 0, 1しかないのにどうやって普通の数値 0, 1, 2, ... を表す?
- 複数の0, 1を組み合わせる
- “000”と並べたら数値の0, “001”なら1, という具合
- この場合、3つの0, 1を使って表すので、全部で8通りを表現できる ... この「3」をビット(bit)数といい、それぞれの桁ビットと言う → **bit は情報の最小単位**
 - 8 bit (8個のビット) → 256 通り
 - 16 bit → 65,536 通り
 - 32 bit → 4,294,967,296 通り
- これは0以上の整数の場合
- 正負の数・小数点などはまた別の表し方がある

ビット列 $b_2b_1b_0$	数値 D
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

ビット列から数値を求める

- それぞれのビットの重みを考えればよい
- ビット列を $b_7b_6b_5 \dots b_0$ のように書くと、 b_n は 2^n の重みをもつので、それを足せばよい（二進数から十進数への変換）

$$\begin{array}{c}
 b_7 \quad \quad \quad b_0 \\
 \swarrow \quad \quad \searrow \\
 10011011 \quad \text{二進数} \\
 \swarrow \quad \searrow \quad \downarrow \quad \swarrow \quad \searrow \\
 128 + 16 + 8 + 2 + 1 = 155 \quad \text{十進数}
 \end{array}$$

ビット列 $b_2b_1b_0$	数値 D
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

二進数同士の足し算（組み合わせ回路）

□ 一桁の二進数同士の足し算は下のパターンのみ

① $0 + 0 = 0$

② $0 + 1 = 1$

③ $1 + 0 = 1$

④ $1 + 1 = 10 \cdots$ 桁上げ(carry)が生じる

(重み)1の桁は、入力のどちらかが1の時のみ1
(重み)2の桁は、入力の両方が1なら1

二進数同士の足し算（複数桁）

- 十進数の場合の足し算(例えば $7+9=16$)を二進数であらわしてみる

$$\begin{array}{r} 7 \\ + 9 \\ \hline \end{array}$$

1 : carry
↓ 6 : sum
16 : Answer

$$\begin{array}{r} 0111 \\ + 1001 \\ \hline \end{array}$$

1111 : carry
↓ 0000 : sum
10000 : Answer

桁上げがよくわからなければ右の十進の計算を考えればよい

$$\begin{array}{r} 99 \\ + 99 \\ \hline \end{array}$$

11 : carry
↓ 98 : sum
198 : Answer

↓
sum=0000, carry=1と表現

- 二進数の場合は桁上げが大量に発生するので複雑だが、コンピュータの足し算演算回路は上記の sum と最上位の桁上げ状況(carry)を別々に出力するモジュールを組み合わせで構成されている

- 前頁の $7+9$ の場合で、二桁目の計算に注目
- ここだけ注目すると
 - 入力: 緑枠内 = A, Bと一つ下から来たC
 - 出力: Sと一つ上の桁へのCとなっている
- つまり、下のようなモジュール
(全加算器: full adder)があればよい

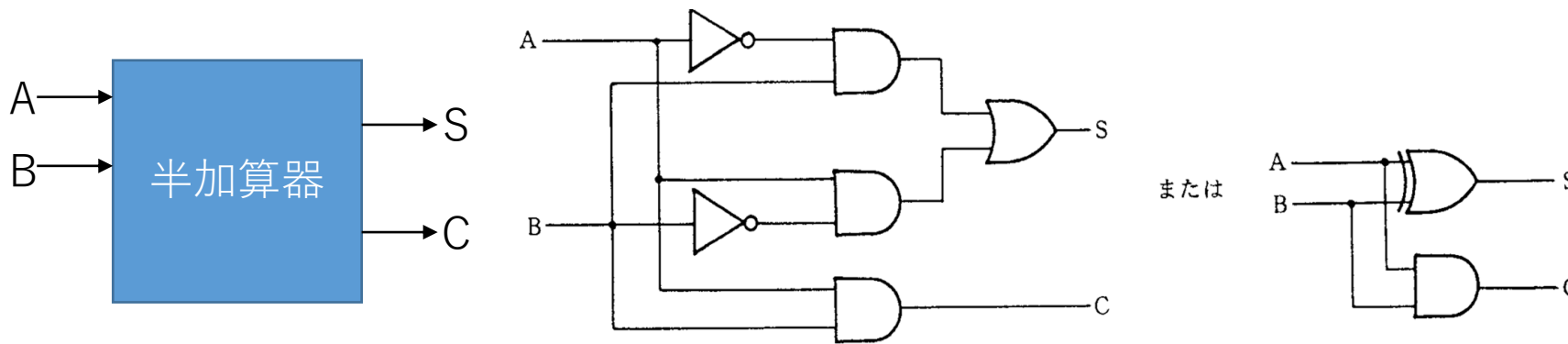


入力

A	0	1	1	
B+	1	0	0	1
C	1	1	1	: carry
S	0	0	0	: sum

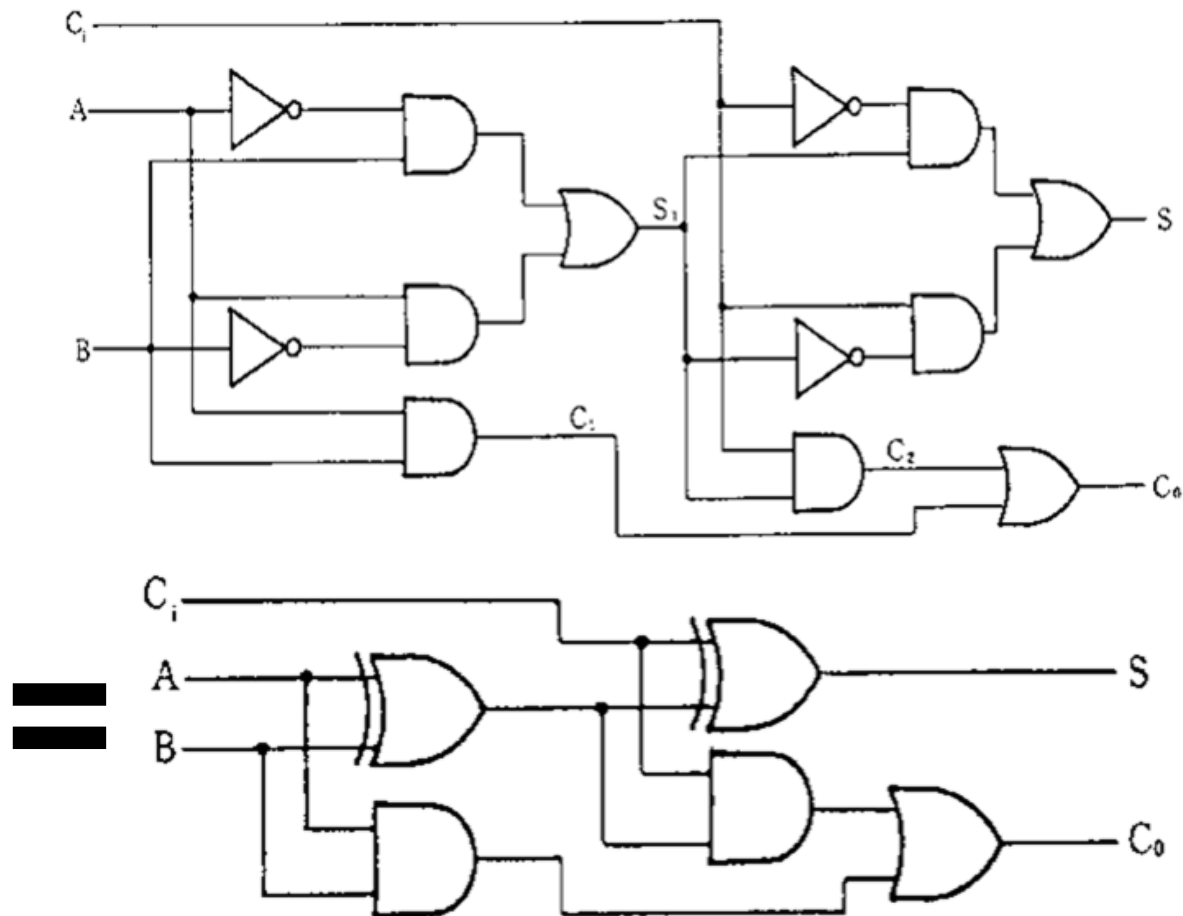
出力

- 下の桁からのCarryを考慮しないものを半加算器(half adder)と呼ぶ
 - 下からのCarryが無いので、出力と上へのCarryはAとBのみで決まり、
 - $S = A \oplus B$
 - $C = A \cdot B$
- (各自、下の真理値表を完成させ、上記式で良いことを確認)



INPUT		OUTPUT	
A	B	S	C
0	0		
0	1		
1	0		
1	1		

□ 下の桁からのcarryを考慮



INPUT			OUTPUT	
A	B	C	S	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		

各自回路図を元に真理値表を作り、確認すること

- 配布された **Exercise-Sheet 1** を全問解く
- 全部出来たらTAに採点してもらう
- 間違いがあったら再度トライ
- 全問正解に至ったらTAに確認印をもらい、終了
- 課題が終了した方から、回収BOXへ提出後は帰って構わない

終了後は、回収BOXへ提出すること。(採点のために回収します)

採点の終了は、16時です。時間内に終えられなかった方には、
クラスルーム経由で指示を出します。

論理回路(2)

工学基礎実験

□ 論理回路

- 入力・出力がある回路
- 入力・出力ともに「論理」になっている
 - つまりYESかNOもしくは1 か 0 → これらを用いた論理代数（ブール代数）
 - これまでの計算(一般的な代数)とは異なる

□ この講義では、組み合わせ回路を取り扱う

- 組み合わせ回路：入力が決まれば、出力が決まる回路
- AND,OR,NOTで構成される
- 特徴として同じ入力に対しては、出力は同じになる

組み合わせ回路の構成部品

- 全ての組み合わせ回路は 論理積(AND)、論理和(OR)、否定(NOT) の組み合わせで作られる
- つまり、組み合わせ回路は入力のビットパターンのみで出力が決まる
- ビットパターン: あるデータ（例えば数値）をビットの羅列であらわしたもの



入力Aのパターン	0010
入力Bのパターン	0101
出力Qのパターン	0111

組み合わせ回路では、このような感じで
A,Bが決まればQが決まる

真理値表から論理式を導く

- すべての組み合わせ論理回路について、真理値表が書ける
- 真理値表があれば論理式を導くことができる
- 論理式から回路を作れる

例:

INPUT		OUTPUT
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

ポイント: 1になる条件を探す

Sが1になる条件はここでは二つ

← $B = 0, B = 1$ が条件 $\rightarrow \bar{A} = 1$ 且つ $B = 1 \rightarrow \bar{A} \cdot B = 1$
 ← $B = 1, B = 0$ が条件 $\rightarrow A = 1$ 且つ $\bar{B} = 1 \rightarrow A \cdot \bar{B} = 1$



この二つの条件のうち、どちらかが満たされれば $S=1$
 $\rightarrow S = \bar{A} \cdot B + A \cdot \bar{B}$ が S を表す論理式となる

□ 1になる条件が真理値表中に n 個 $\rightarrow n$ 項の論理和になる

□ 右の場合4項となり

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

となる

□ 回路にするときに必要な素子:

- NOT5個
- 3入力AND5個
- 5入力OR1個

INPUT			OUTPUT
A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

カルノー図による最適化

□ カルノー図を使って、論理式を最適化できる

縦横方向にはビット一つずつ変化

INPUT			OUTPUT
A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



	$\bar{A} \cdot \bar{B}$	$\bar{A} \cdot B$	$A \cdot B$	$A \cdot \bar{B}$
$\bar{C}$			○	
C	○	○	○	○

ポイント
なるべく大きい
共通項を発見

[Red dashed box] $\rightarrow A, B$ によらず $C = 1$ なら出力は1

[Green dashed box] $\rightarrow C$ によらず $A = 1, B = 1$ が満たされれば出力は1

$\rightarrow S = A \cdot B + C$ だけで良い (AND, OR 1個ずつ)

カルノー図のパターン

- 4変数くらいまでなら最適化可能
- それ以上は人間には難しいのでコンピュータでクワイン・マクラスキー法等に頼るしかない

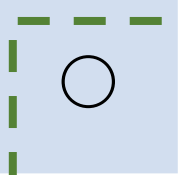
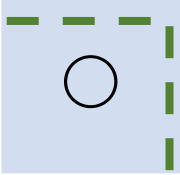
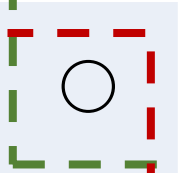
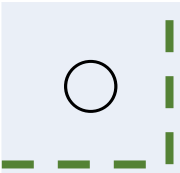
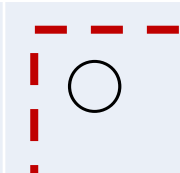
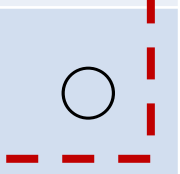
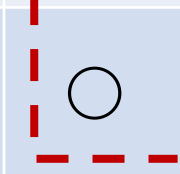
	$\bar{A} \cdot \bar{B}$	$\bar{A} \cdot B$	$A \cdot B$	$A \cdot \bar{B}$
$\bar{C}$		○	○	
C	○			○

$\boxed{} \rightarrow C=0$ で $B=1$ であれば A に依らず1 $\rightarrow B \cdot \bar{C}$
 $\boxed{} \rightarrow C=1$ で $B=0$ であれば A に依らず1 $\rightarrow \bar{B} \cdot C$

$$\therefore Y = B \cdot \bar{C} + \bar{B} \cdot C$$

まわりこみ可能!!

4変数のカルノー一図例

	$\bar{A} \cdot \bar{B}$	$\bar{A} \cdot B$	$A \cdot B$	$A \cdot \bar{B}$
$\bar{C} \cdot \bar{D}$				
$\bar{C} \cdot D$				
$C \cdot D$				
$C \cdot \bar{D}$				

ポイント
重複しても良いので
できるだけ大きな共通項を見つける

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

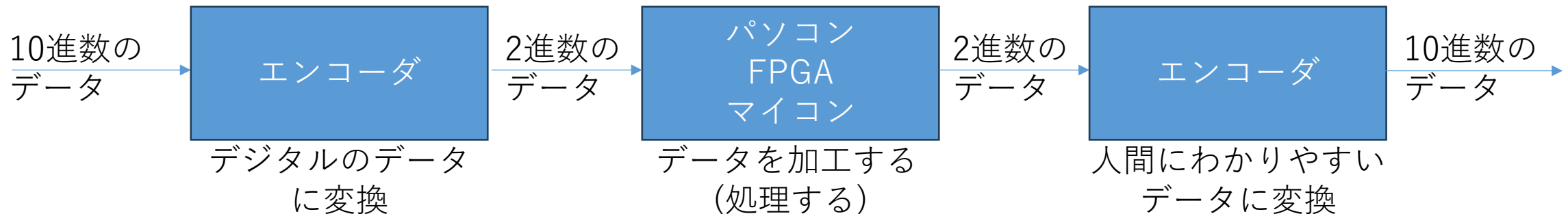
一般的に

- エンコーダ(encoder) … ある情報をコード化する (暗号コード化など) ための機械・回路・数式・プログラム
- デコーダ(decoder) … コード化されたものを元の情報に戻す

論理回路の場合(例)

- エンコーダ(encoder) … 十進数を二進数にする、グレイコードにする、等
- デコーダ(decoder) … 二進数を十進数に戻す、二進数の数値を 7 セグLEDのパターンにする、等

エンコーダ・デコーダの例



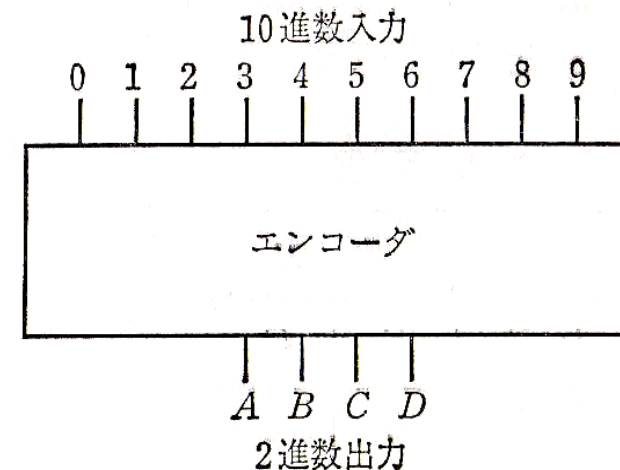
十進→二進変換エンコーダ

□ 情報を記述する文字や記号を何ビットかの2値(2進)コード(binary-code)に変換

- 2桁の2進数 → 0～3の $2^2 = 4$ 種類の組合せが表現可能
- 3桁の2進数 → 0～7の $2^3 = 8$ 種類の組合せが表現可能
- 0～9を表すには → 4桁を使う必要がある
 - 4桁なら0～15を表現できるが...

□ 2進化10進数(BCDコード)

- 0～9までの数字を入れると、4桁の2進数で出力
- 4桁の2進数を使用し、10進数のように変換
- 例：17をBCDコードで示すと 0001 0111 になる

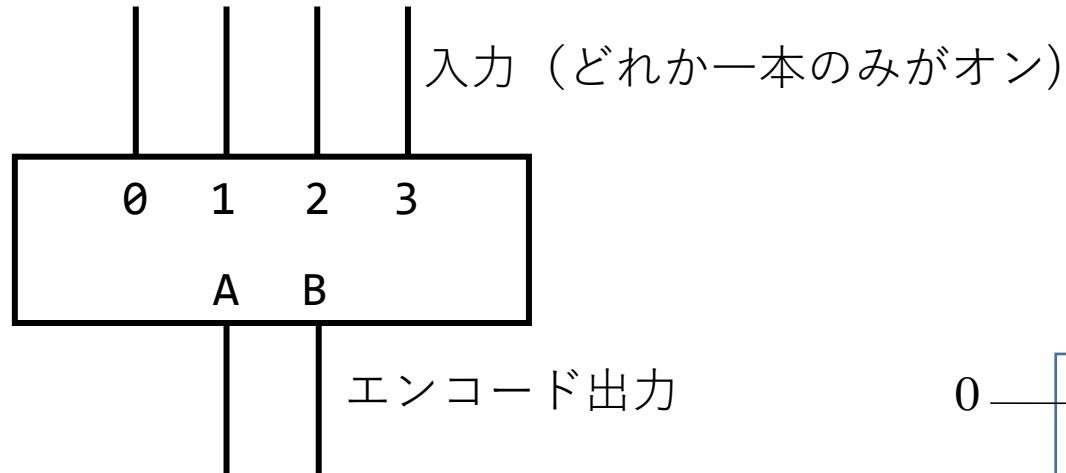


桁数とあらわせる数字

7	6	5	4	3	2	1	← 2進数での桁数n
1000000	100000	10000	1000	100	10	1	← n桁2進数の100...00
2^6	2^5	2^4	2^3	2^2	2^1	2^0	← 2^{n-1}
64	32	16	8	4	2	1	← 10進数

2ビットエンコーダ

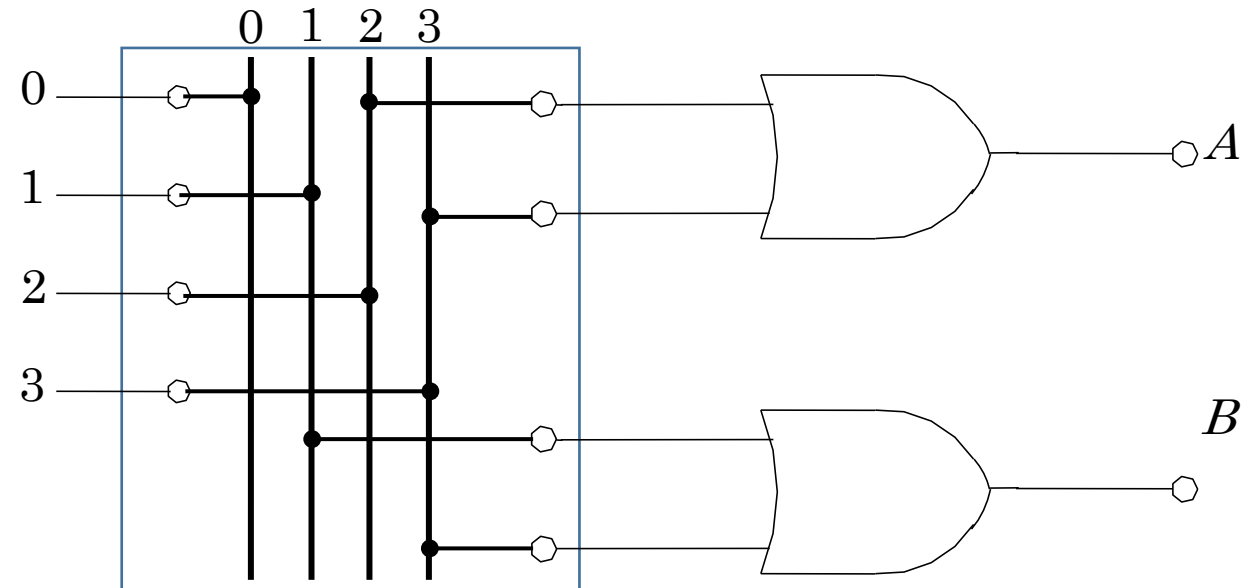
Step1: 真理値表を作る



10進数	2進数	A	B
0	00	0	0
1	01	0	1
2	10	1	0
3	11	1	1

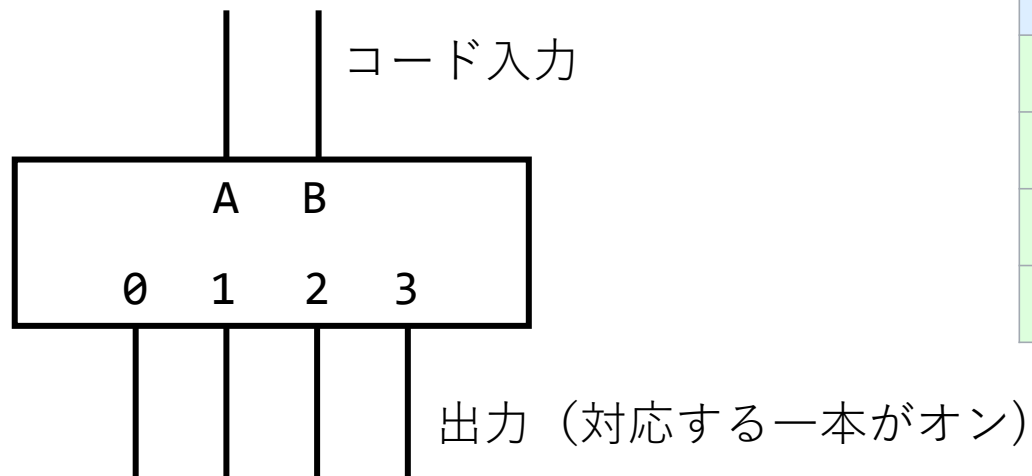
Step2: 回路を作る

- Aになる条件は入力2か3がオンのとき
- Bになる条件は入力1か3がオンのとき



2ビットデコーダ

Step1: 真理値表を作成



2進数入力		10進数出力			
A	B	0	1	2	3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

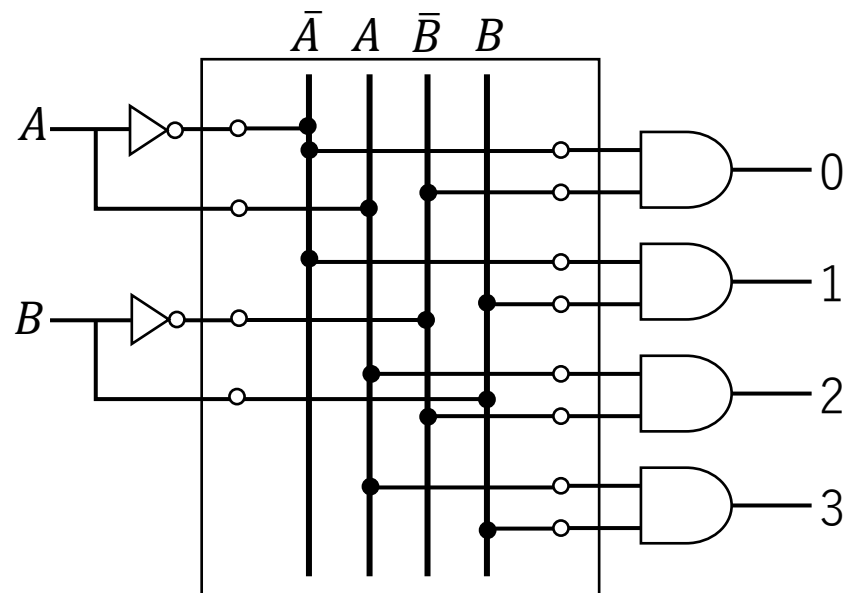
Step2: 論理回路と論理式の作成

0: $\bar{A} \cdot \bar{B}$

1: $\bar{A} \cdot B$

2: $A \cdot \bar{B}$

3: $A \cdot B$



- 配布された **Exercise-Sheet 2** を全問解く
- 全部出来たらTAに採点してもらう
- 間違いがあったら再度トライ
- 全問正解に至ったらTAに学籍番号・氏名を記録してもらい、終了

終了後は、回収BOXへ提出すること。(採点のために回収します)

採点の終了は、16時です。時間内に終えられなかった方には、
クラスルーム経由で指示を出します。