

# デジタル回路入門 Part 2

---

2010年7月28日

内田智久 (e-sys, IPNS, KEK)

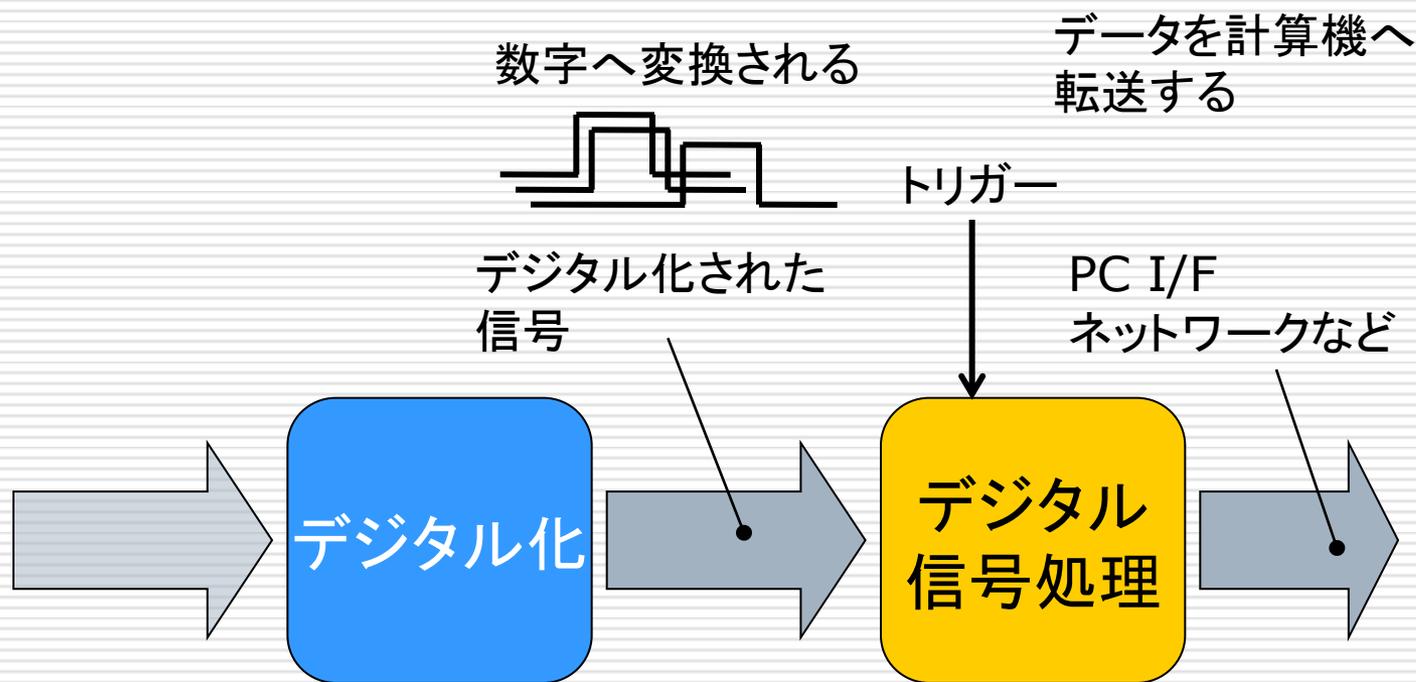
先端エレクトロニクスDAQセミナー'10

---

# 実験で使用されるデジタル回路は どのようなものか理解する

そして、これを作る！

# 検出器システム内でのデジタル回路



もっとも単純な回路を考えてみましょう

# フロントエンド回路の役目

---

## 検出データを計算機へ送り届ける

やるべき事は

- ①アナログデータを計算機が処理できるデジタルデータへ変換
- ②デジタルデータを計算機へ転送

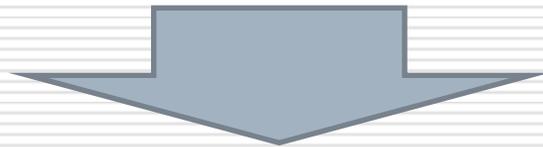
しかし、これだけではない。もうひとつ重要な役目がある

# データ量の削減

---

全ての検出データを処理（解析）する事は出来ない

多くの場合、全て処理しても意味が無い  
イベント以外は不要なノイズ



イベントらしいデータのみ取得して解析  
(トリガー機能)

前処理としてのデータ量削減

# フロントエンドの デジタル回路がすべき事

---

- アナログ・インターフェース
    - ADC, DAC
    - ASICやアナログ・フロントエンドに搭載される事も
  - 計算機とのインタフェース
  - データ量削減
    - トリガー機構
    - データバッファリング
    - 前処理
- 
- この部分を設計する

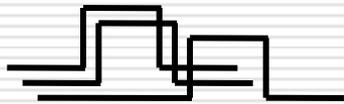
# 検出器デジタル回路 基本機能

シグナルらしいイベントのデータのみ  
計算機へ転送する

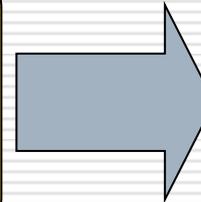
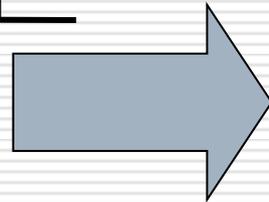
トリガーは判定時間が  
必要なので遅れてやってくる

トリガー

数字へ変換される



デジタル化された  
信号



データを計算機へ  
転送する

PC I/F  
ネットワークなど

検出器からの信号は  
絶え間なくやってくる

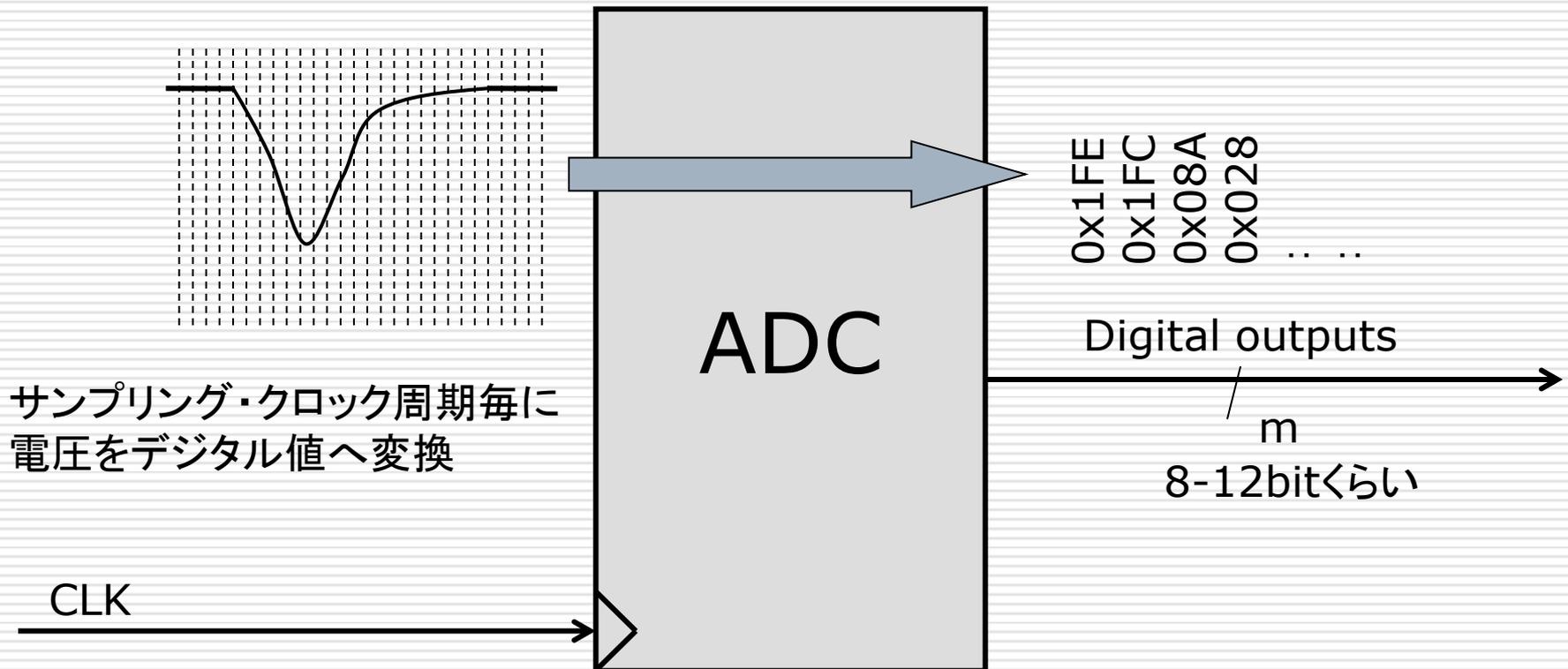
データ解析する為  
データ転送

# アナログ・インターフェース

---

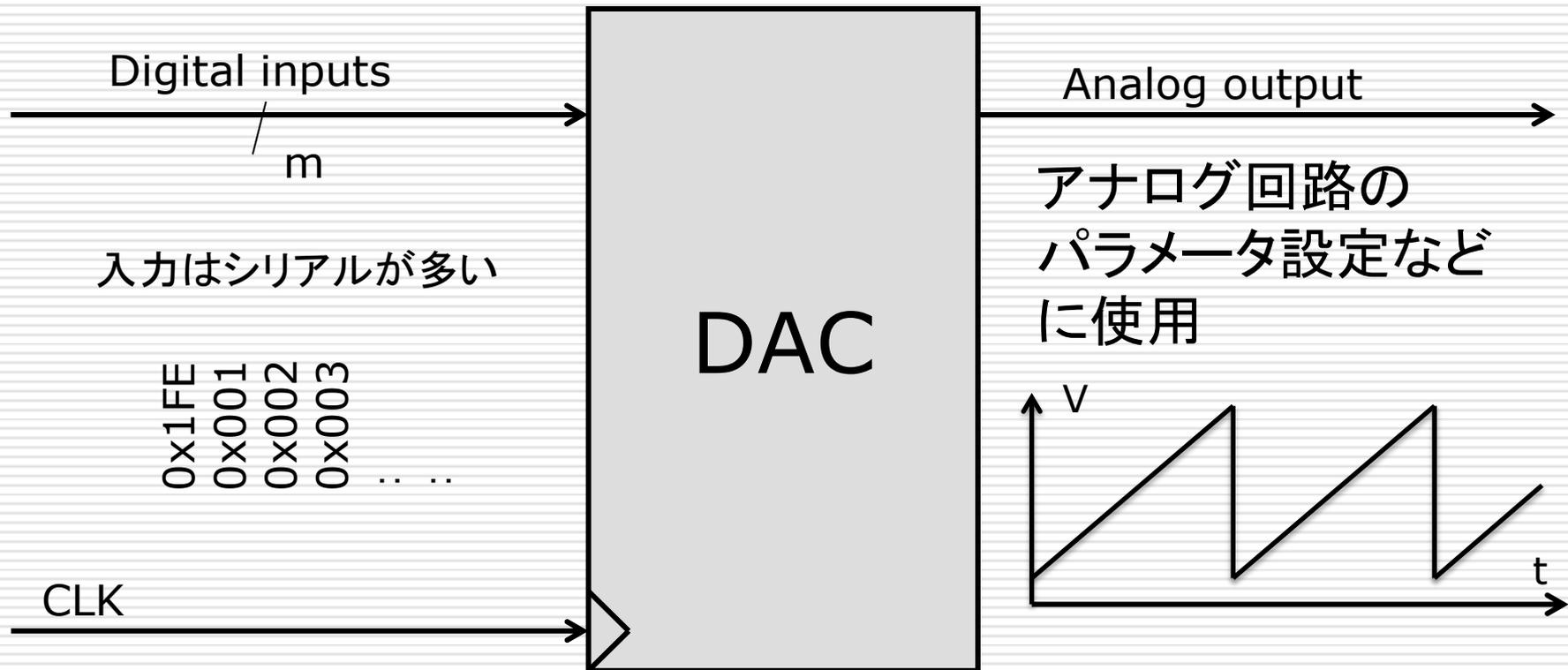
- 市販部品またはASICに内蔵
- Analog-to-digital converter (ADC)
  - 周期的にアナログ値をデジタル値へ変換する回路
    - サンプルングと言う
  - サンプルング周波数 < 数Gbps
  - 分解能8-12bitくらいを良く使う
  - 回路開発として熱い領域のひとつ
- Digital-to-analog converter (DAC)
  - デジタル値に応じたアナログ信号へ変換する回路
  - 分解能8-12bitくらいを良く使う
  - アナログ回路のパラメータ調整に使用する事が多い
  - 例えば、ディスクリの閾値電圧の設定など

# ADC



# DAC

デジタル値をアナログ値へ変換する



# トリガー機構

---

- トリガーが発行されるまで時間がかかる
  - イベントらしいシグナルを判断する時間
  
- バッファリング機能が必須
  - シグナル発生してからトリガー受信まで検出器データを保持する為
  - メモリーを使用する
    - リングバッファ、FIFOメモリ

# データ転送機能

---

最終的に必ず計算機へデータを送らなければいけない

- ネットワーク技術の利用
  - イーサネット
- バスシステムの利用
  - VME, CAMAC, PCI, ATCAなど
- コンピュータ周辺技術の利用
  - RS232C, USB, PCIe, Fiber channelなど
- 実験独自システムの利用
  - COPPERシステムなど

# 再度、 基本機能

シグナルらしいイベントのデータのみ  
計算機へ転送する

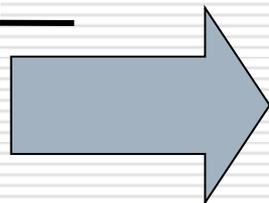
トリガーは判定時間が  
必要なので遅れてやってくる

トリガー

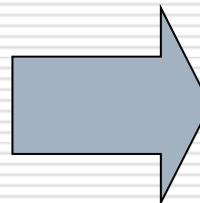
数字へ変換される



デジタル化された  
信号



データを計算機へ  
転送する



PC I/F  
ネットワークなど

検出器からの信号は  
絶え間なくやってくる

データ解析する為  
データ転送

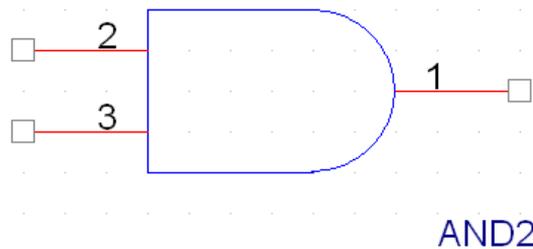
---

# デジタル回路

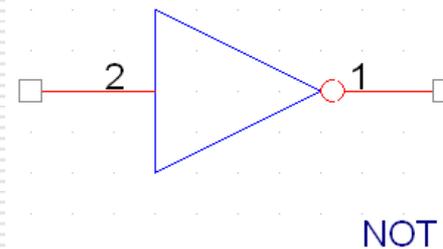
目的: デジタル回路になれる

# デジタル回路の構成要素

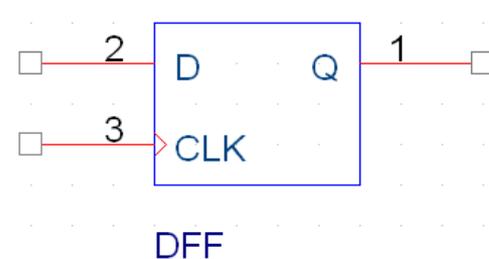
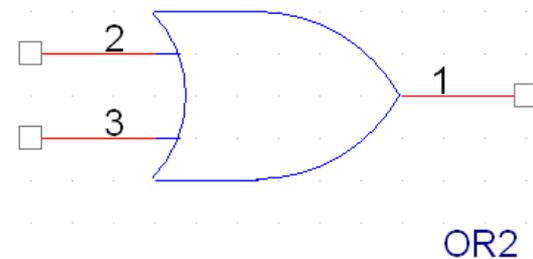
$$O = A \& B;$$



$$O = \sim I;$$



$$O = A | B;$$



全てのデジタル回路は

この4つの要素(回路)のみで構成されている

CPU, DSPなど一見複雑な回路もこの4つの組み合わせで設計可能

# 機能ブロック

---

- アナログインターフェース
  - ADC
  - DAC
- データバッファ
  - メモリ

# 組み合わせ回路

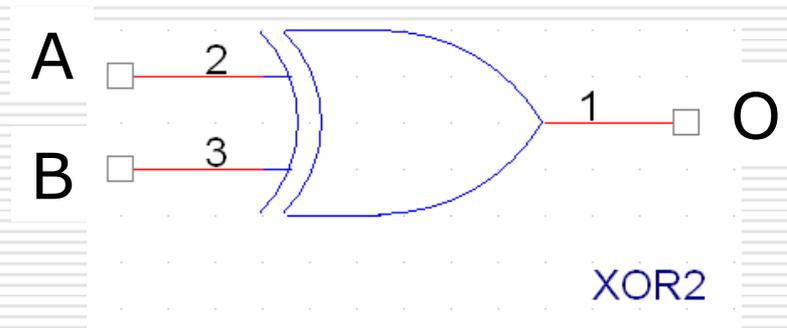
---

- 記憶素子を使わない回路
  - AND, OR, INVのみで表現可能
- 入出力の関係を表で表すことができる
  - 入力信号のみで出力が一意に決定する

希望する状態を検出する

信号のコインシデンスを取ったりする

# 問題1: XOR



A	B	O
L	L	L
H	L	H
L	H	H
H	H	L

基本要素を使って設計して(考えてみて)ください **5分間**  
AND, OR, INV

# 基本的な考え方

---

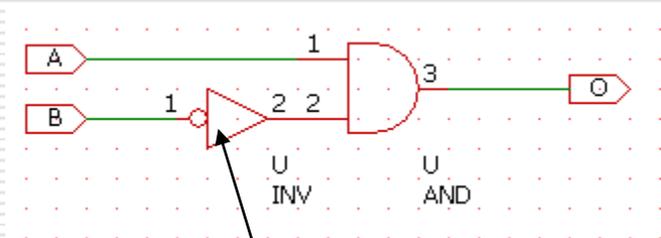
- いろいろな考え方がありますが...
- 今回は、Hを出力する条件に注目してみましょう

A	B	O
L	L	L
H	L	H
L	H	H
H	H	L

ここに注目！！

# ANDの気持ち

- 入力信号から条件を抽出する
- INVと組み合わせる事で特定入力パターンの時のみ出力する回路を設計できる
- 例
  - Aかつ~Bを抽出(~はINVの意味)
  - 気持ちとしては、Aかつ~Bの時のみ出力(H)

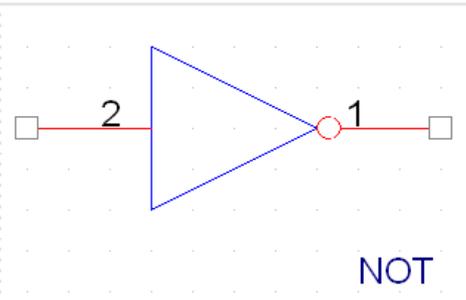


丸の位置が移動している  
丸の気持ちは？

A	B	O
L	L	L
H	L	H
L	H	H
H	H	L

この条件を抽出

# INVの気持ち



I	O
L	H
H	L

- 出力にある丸の意味は？
  - Lの時に有効
  - 有効になる極性が反転している
    - 負論理と言う
    - 電圧が低い時に有効になる信号 (NIMなど)
- Lの条件抽出

# ORの気持ち

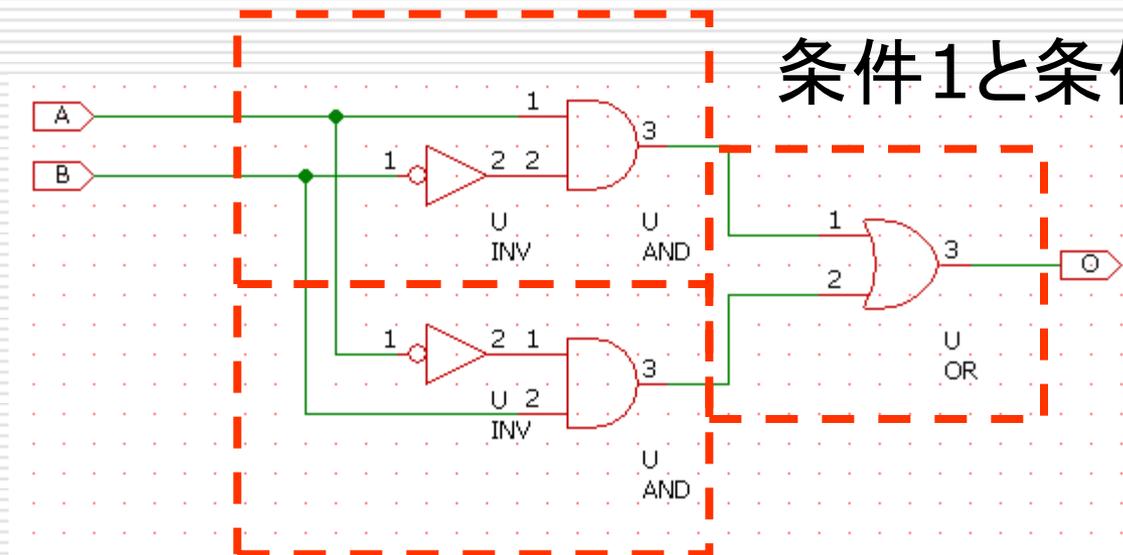
- 入力信号を足す
- 条件Aの時または条件Bの時に出力
- ANDと共に使用する事で初めて実用的な条件抽出ができる

	A	B	O
	L	L	L
条件A	H	L	H
条件B	L	H	H
	H	H	L

と言う事で、次ページのようになります

# 設計したXOR

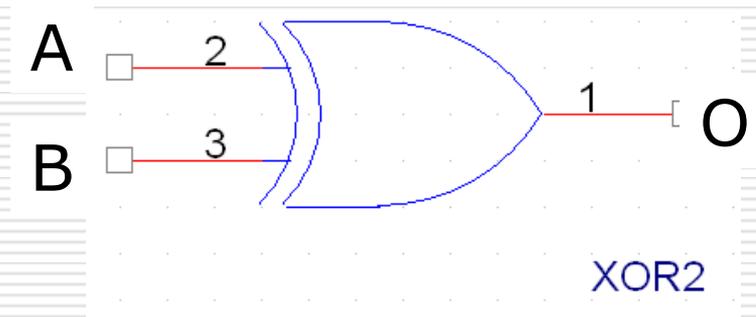
条件1:  $A=H$  &  $B=L$ を抽出



条件2:  $A=L$  &  $B=H$ を抽出

# XORの気持ち

- 入力AとBが異なる時にH
- A=HにするとBが反転する



A	B	O
L	L	L
H	L	H
L	H	H
H	H	L

# 負論理

---

- 負論理の2入力AND, ORをAND, OR, INVで書いてください
  
- Wired-OR, Wired-ANDとは？

# 代表的な組み合わせ回路

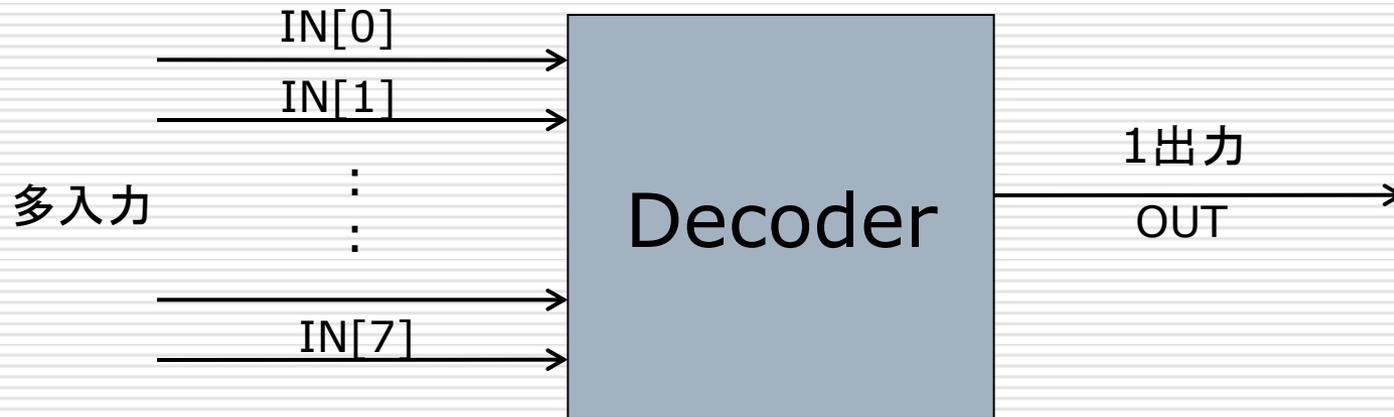
---

- Decoder
  - 符号(複数信号)から特定条件を検出する回路
- Encoder
  - 複数信号から特定の符号を生成する回路
- Multiplexer / Selector
  - 複数の信号から一つの信号を選択する回路
- Adder
  - 2つの符号を加算する回路
- Multiplier
  - 2つの符号を乗算する回路

# Decoder

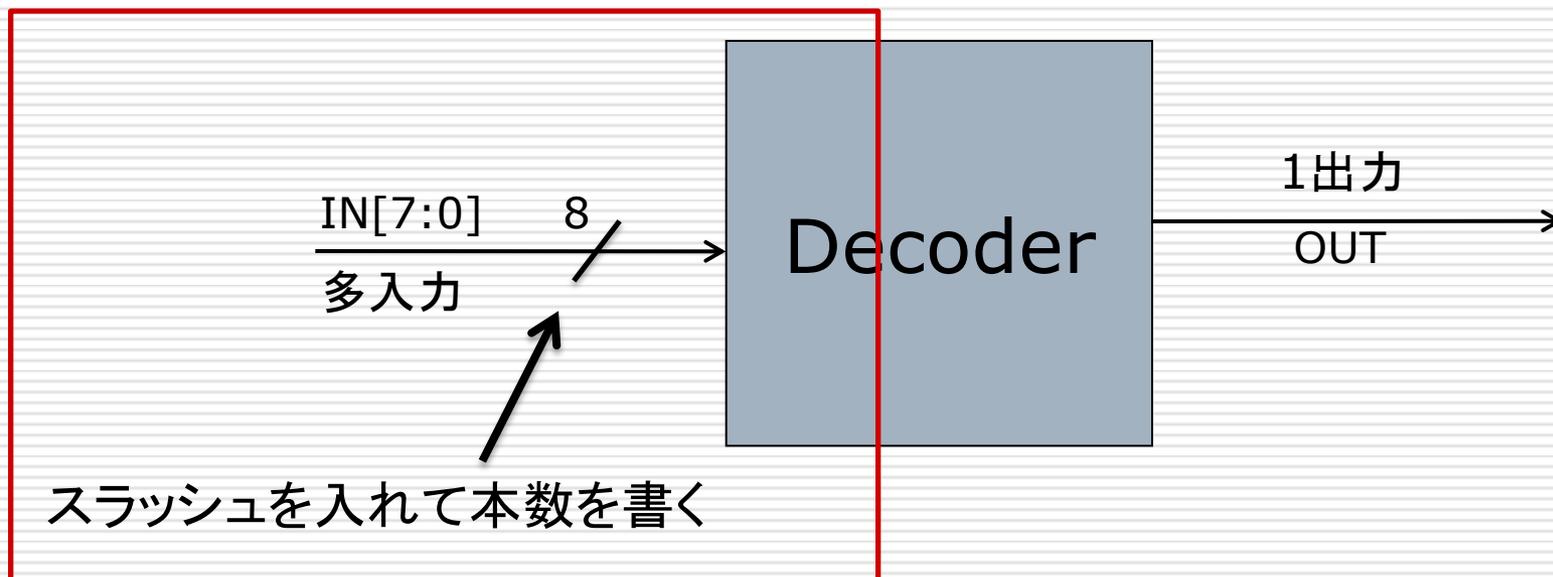
---

## 特定の条件を検出する回路



例えば、32bit値の範囲を演出  
0xFFFF\_0000から0xFFFF\_FFFFまでの値を検出  
アドレスデコーダなどに用いる

# 補足：複数本信号の表記



# デコーダー、課題2-7

---

- 8bit入力信号 $I[7:0]$ があるとき下の条件を検出する回路を設計してください
  - [問題2]  $I=5$ を検出
  - [問題3]  $I>0$ を検出
  - [問題4]  $I>1$ を検出
  - [問題5]  $I>3$ を検出
  - [問題6]  $I>2$ を検出
  - [問題7]  $I<4$ を検出

# デコーダー（解説）

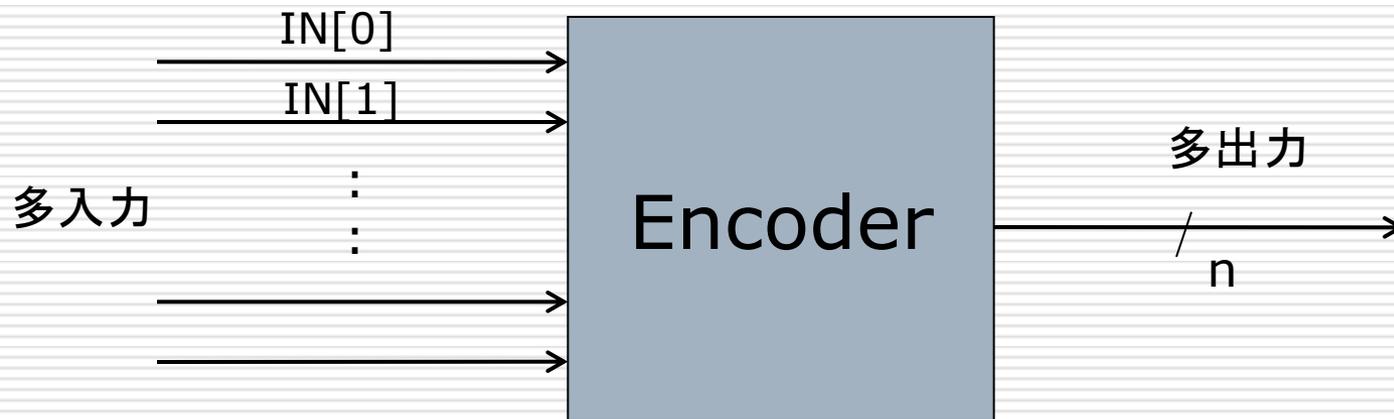
---

- 解答例、少し工夫すると簡単になる事を示したい
- これは解答例です。他の書き方もあります。
  - 5を検出:これは素直にANDで検出
  - $I > 0$ を検出:
    - $I \neq 0$ の時、これは $I[0]-I[7]$ の何れかがHの時
  - $I > 1$ を検出:  $I \neq 0$ かつ $I \neq 1$ の時
    - $I \neq 0$ かつ $I \neq 1$ の時、これは $I[1]-I[7]$ の何れかがHの時
  - $I > 3$ を検出:
    - $I \neq 0, \dots, 3$ の時、これは $I[2]-I[7]$ の何れかがHの時
  - $I > 2$ を検出:
    - $I > 3$ を検出+( $I = 3$ を検出)
  - $I < 4$ を検出: 上の反転
    - $I > 3$ でない時
- ちなみに $IA > IB$ を検出するにはどうしたらよいでしょうか？

# Encoder

---

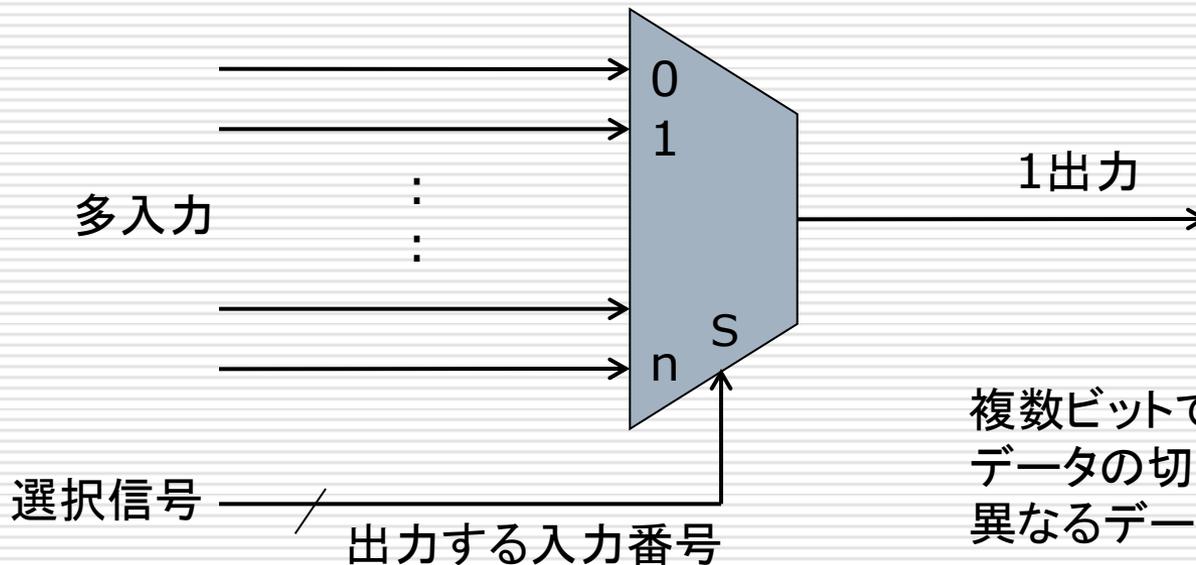
## 入力に応じて符号を生成する回路



例えば、最も高い優先順位の信号の番号を出力  
割り込みの優先順位を決めるときなどに使用

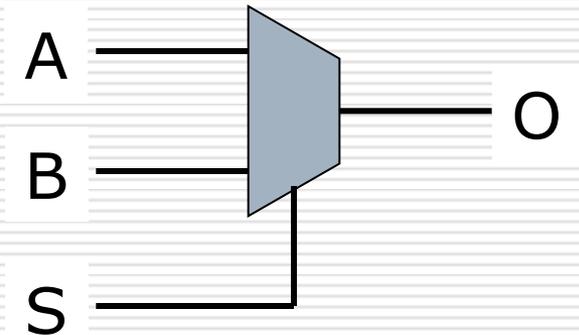
# Multiplexer / Selector

## 複数の入力から一つを選択する回路



複数ビットで使用されることが多い  
データの切り替え、  
異なるデータの時間分割転送など

# 問題8



気持ち：  
Oへ出力する入力信号A,Bを  
Sにより切り替える

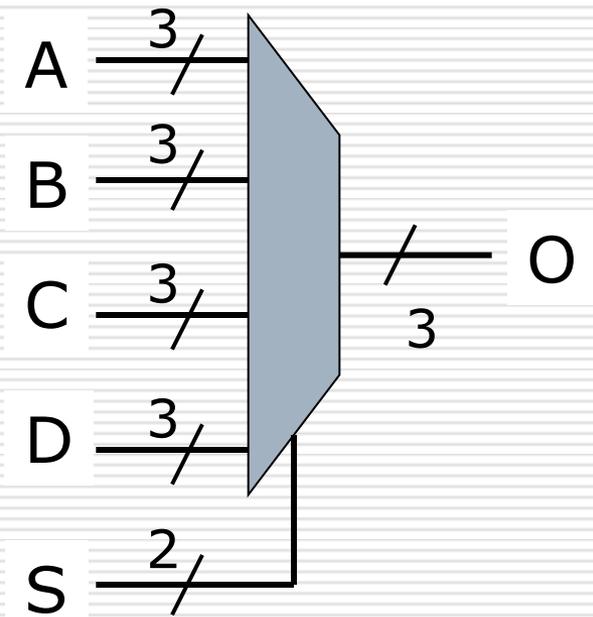
入出力の関係

A	B	S	O
L	X	L	L
H	X	L	H
X	L	H	L
X	H	H	H

XはDon't careで何でも良いと言う意味

基本要素を使って設計してください

# 問題9

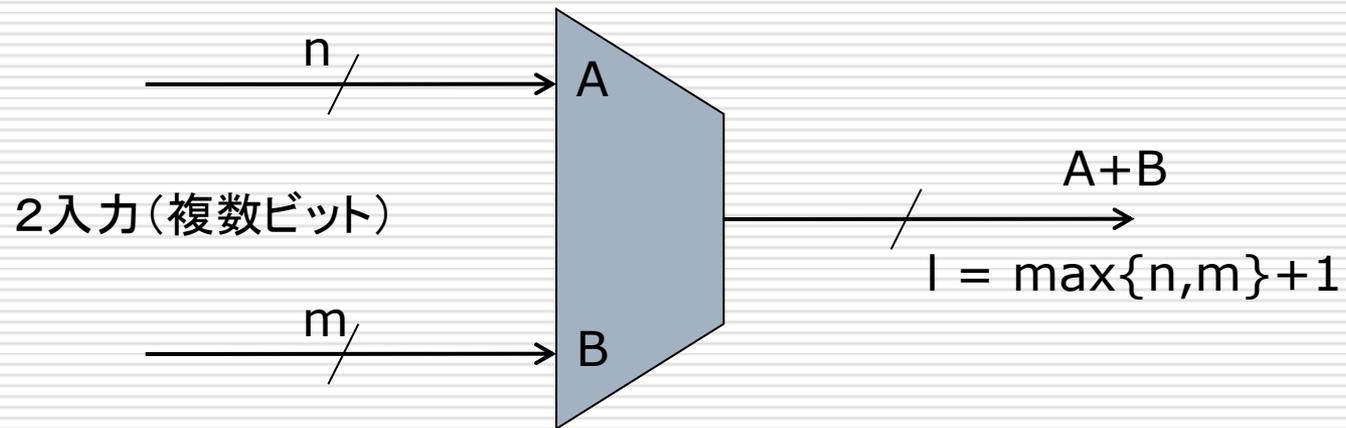


S=0の時にAを選択  
S=1の時にBを選択  
S=2の時にCを選択  
S=3の時にDを選択

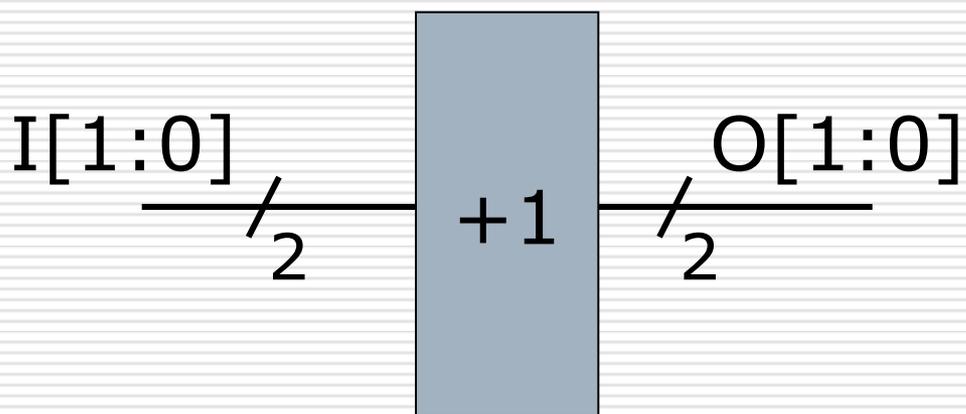
基本要素を使って設計してください

# Adder

---



# 課題10: +1演算器

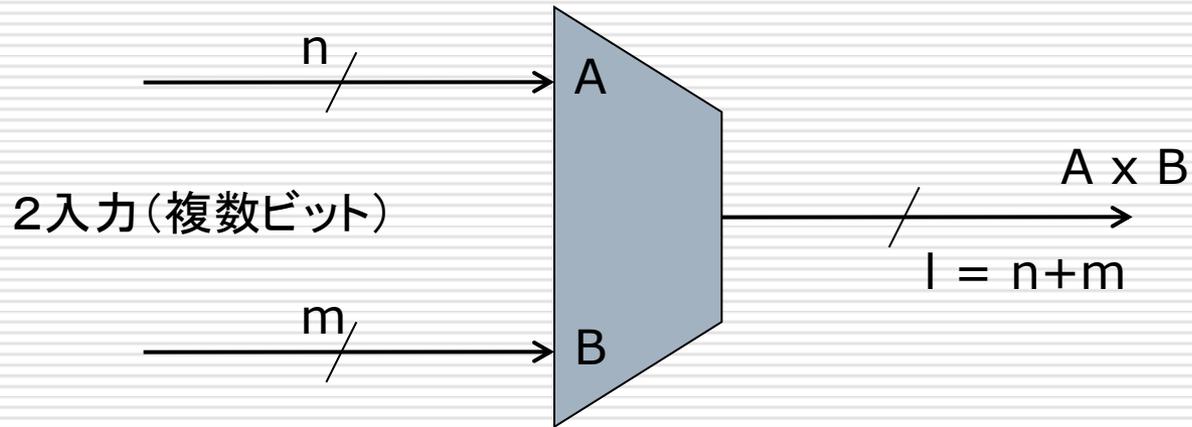


I	O
0 (00)	1 (01)
1 (01)	2 (10)
2 (10)	3 (11)
3 (11)	0 (00)

基本要素を使って設計してください

# Multiplier

---



掛け算器はあまり使用しません  
回路規模が大きく、高速動作させるには高度な技が必要です  
掛け算器は奥が深いです。色々な回路を考える事ができます