

東北大学 工学部 機械知能・航空工学科  
2015年度 5セメスター・クラスD

# 計算機工学

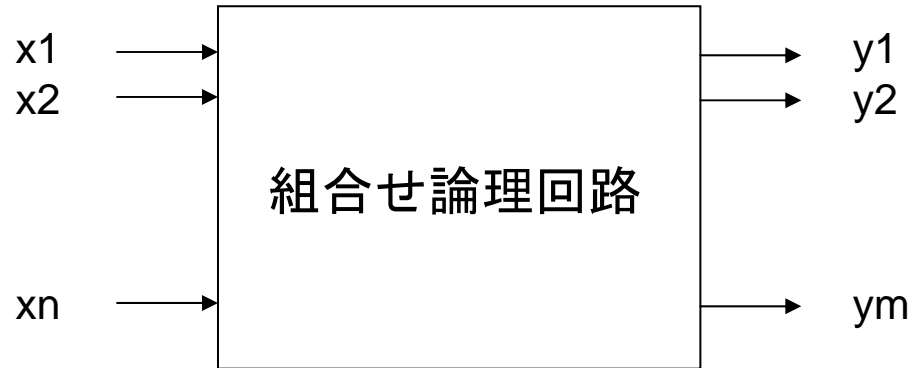
## 10. 組合せ回路 (教科書3.4～3.5節)

大学院情報科学研究科

鏡 慎吾

<http://www.ic.is.tohoku.ac.jp/~swk/lecture/>

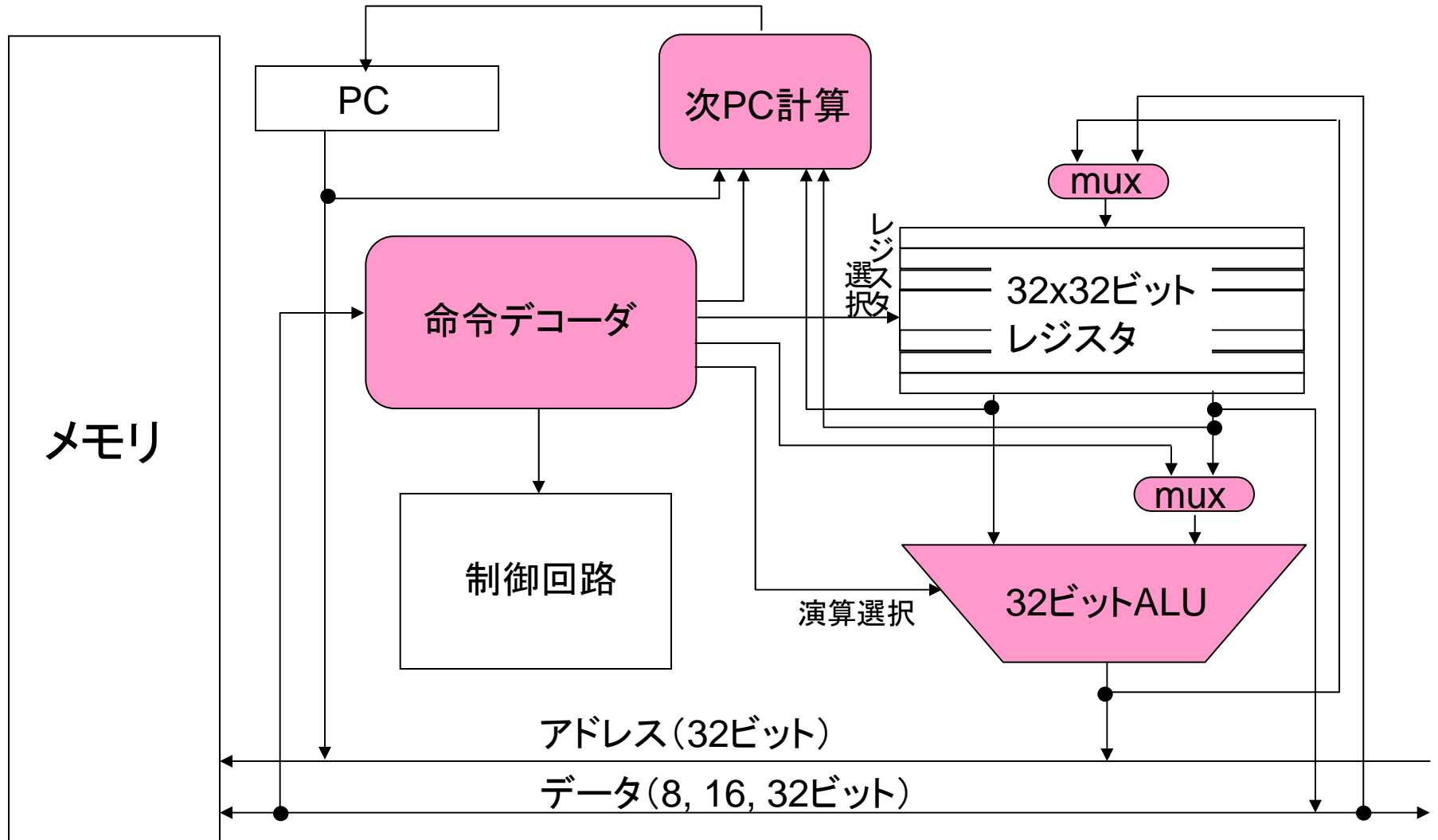
# 組合せ論理回路



$$y_i = f_i(x_1, x_2, \dots, x_n), i = 1, 2, \dots, m$$

- ある時点での出力が, その時点の入力のみで決まる (記憶を持たない) 回路
  - フィードバックが存在しない (入力→出力の方向にだけゲートが接続されている)
- 原理的には,  $n$  入力の論理関数が  $m$  個並んでいるものだと考えればよい

# 復習: MIPSの構造

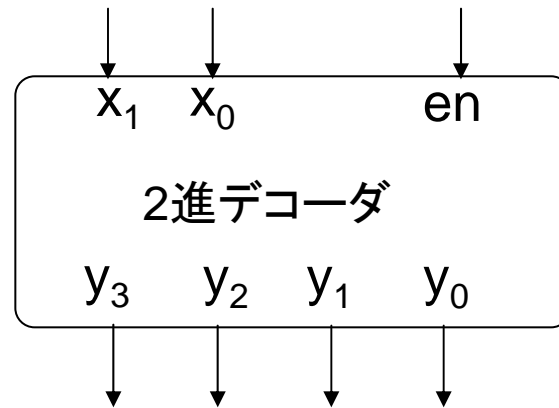


色つきの部分が組合せ回路

# 組合せ論理回路の構成方法

- 原理上は、必ず積和形回路で表すことができる, しかし
  - $n$ が大きい場合, 簡単化の計算に膨大なコストがかかる
  - それが最適とは限らない
- 算術論理演算のように入出力関係の規則性が高い場合は, その規則性に注目して回路を組み立てる方がよい
  - 複数の回路を接続するための部品
    - 2進デコーダ, マルチプレクサ
  - 演算回路
    - 算術演算, 論理演算
  - 複数回路の接続例
    - ALU

# (2進)デコーダ



- $en = 0$  のときは全出力が0
- $en = 1$  のときは, 入力を2進数  $k$  と見なして, 出力  $y_k$  を1, 他を0とする

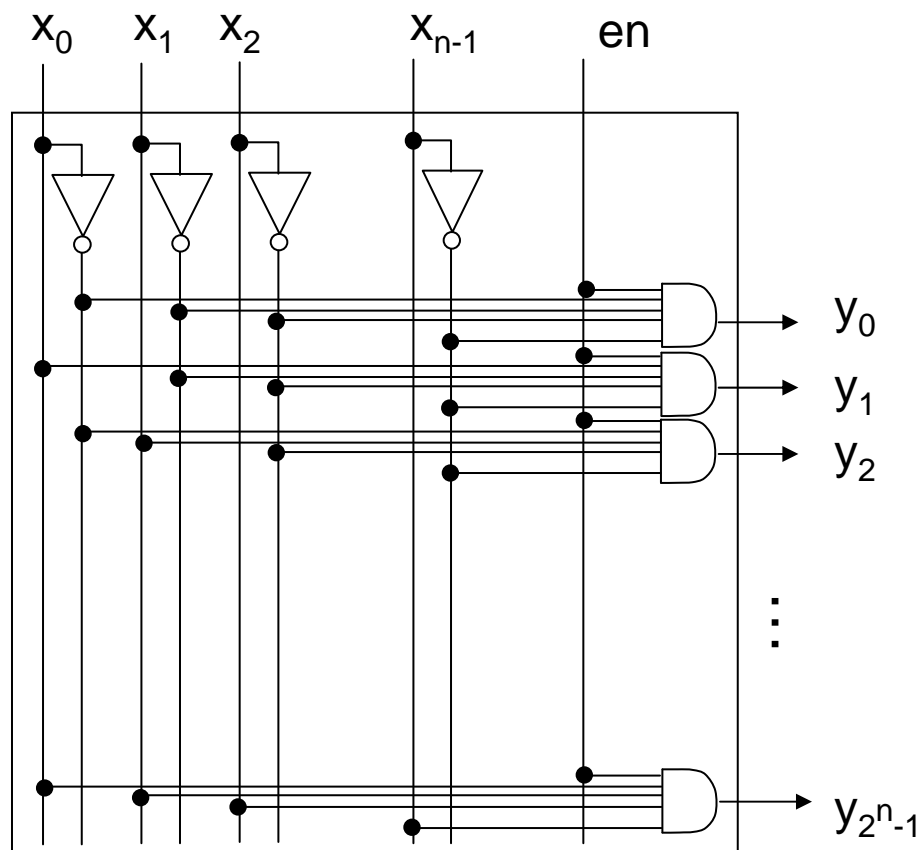
例:  $x_1 = 1, x_0 = 0$  のとき, 入力は2進数で「2」を表すので,  $y_2$  のみが1となる

- エンコード: 一般に, 注目している量に適当な数値(符号)を与えること
- デコード: エンコードの逆
- この例では, 「何番目の信号線か?」を2進数として符号化している
- $en$  は enable の略で, 活性化信号などと呼ばれる

# デコーダの真理値表と回路図

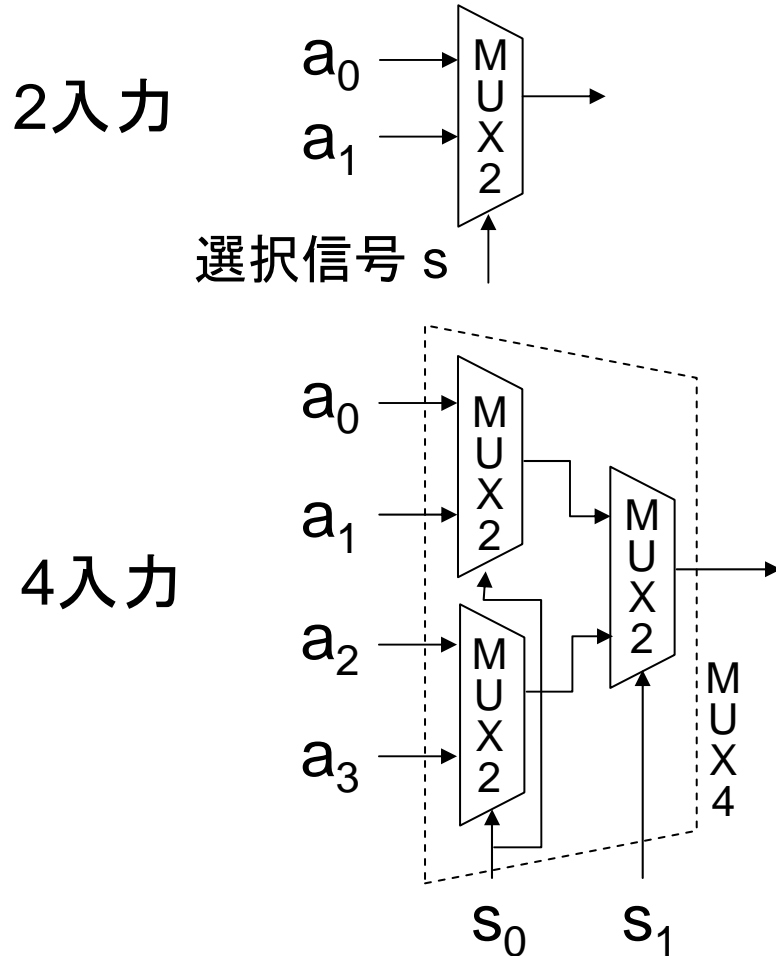
各  $y_i$  について真理値表を書くと, 1行だけ出力が1になるような表となる

$x_2$	$x_1$	$x_0$	$y_7y_6 \cdots y_1y_0$
0	0	0	0000 0001
0	0	1	0000 0010
0	1	0	0000 0100
0	1	1	0000 1000
1	0	0	0001 0000
1	0	1	0010 0000
1	1	0	0100 0000
1	1	1	1000 0000



# マルチプレクサ (セレクタ)

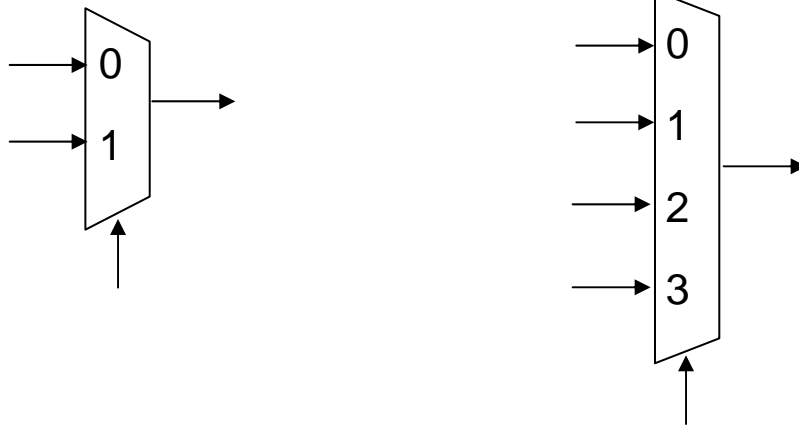
$s = i$  なら  $a_i$  を選ぶ



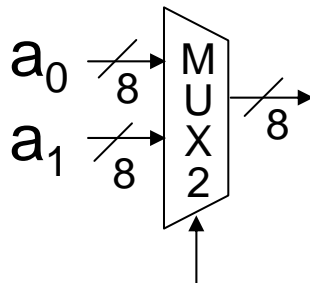
$a_0$	$a_1$	$s$	mux2
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

# マルチプレクサのバリエーション

どの制御信号によってどの入力を選択されるかを明示したい場合



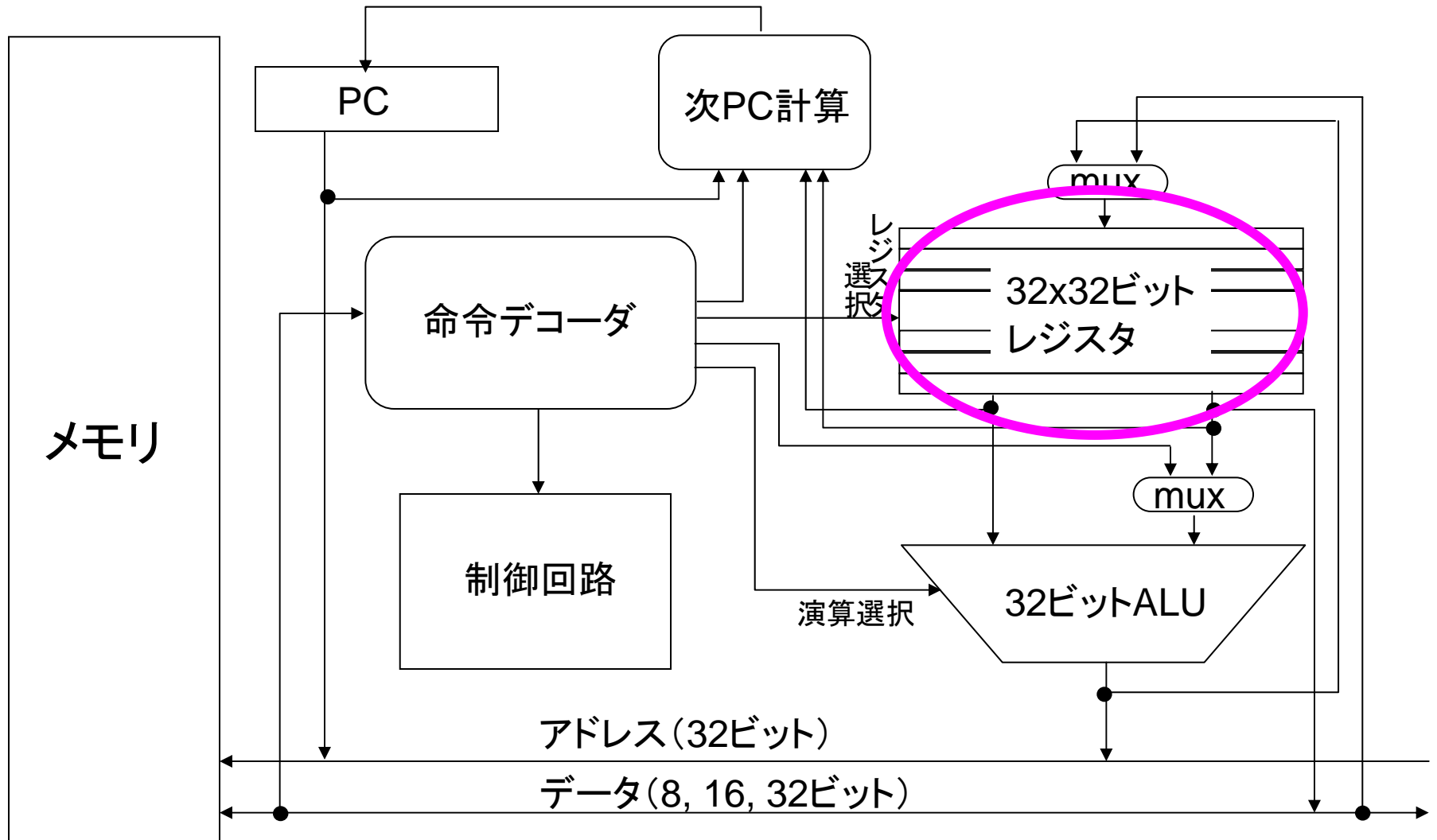
多ビットをまとめて選択したい場合



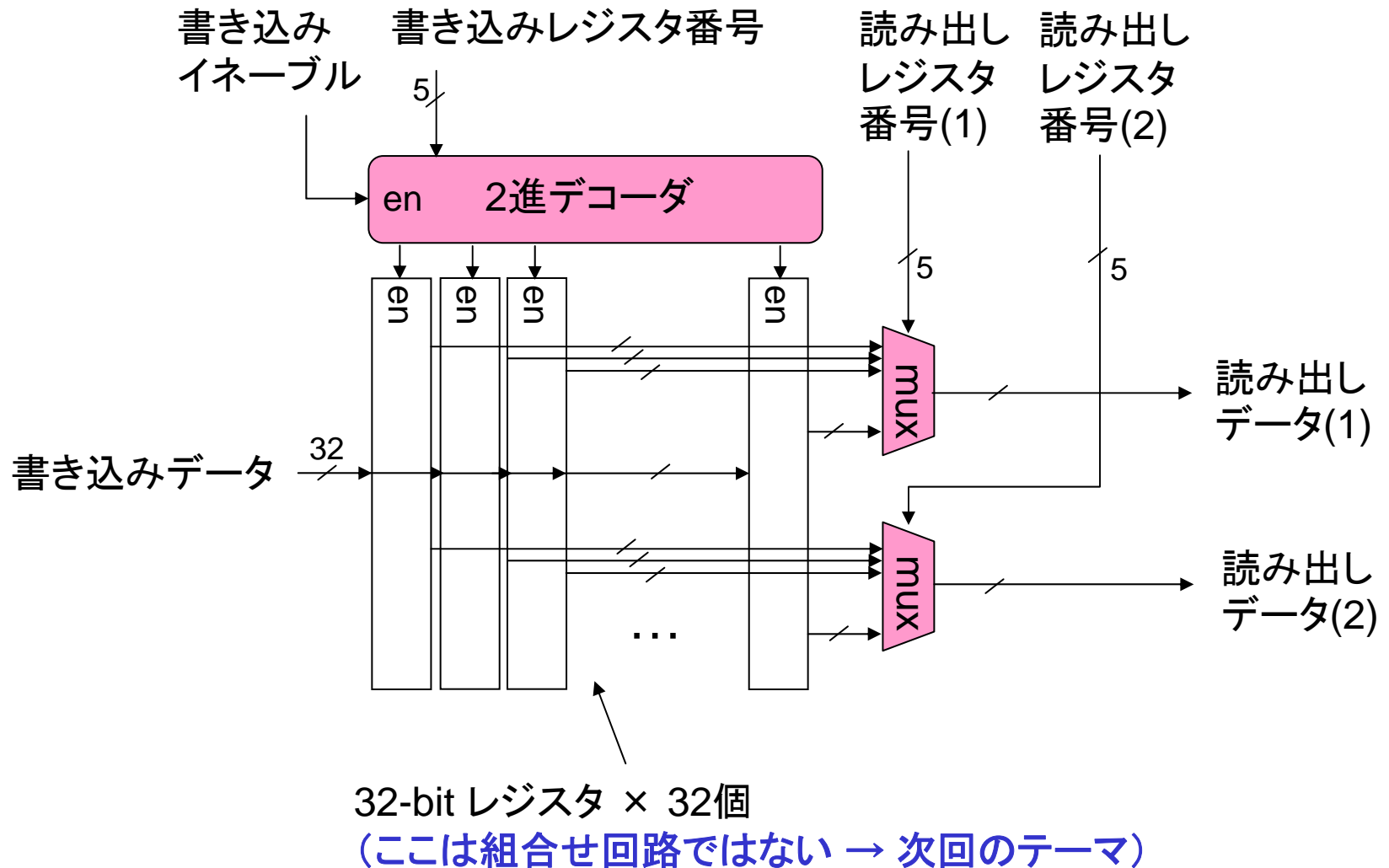
- 短い斜線と数字は、複数ビットをまとめたことを表示している  
(自明な場合, 興味のない場合は適宜省略)
- 記号の形状は, 台形だったり楕円だったりといろいろな流儀がある



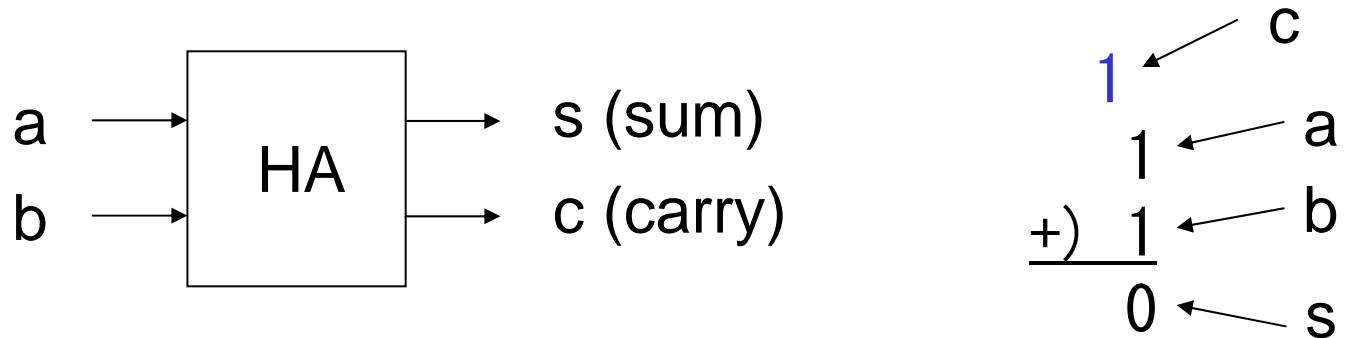
# 復習: MIPSの構造



# 32 × 32ビットレジスタ (1入力2出力)



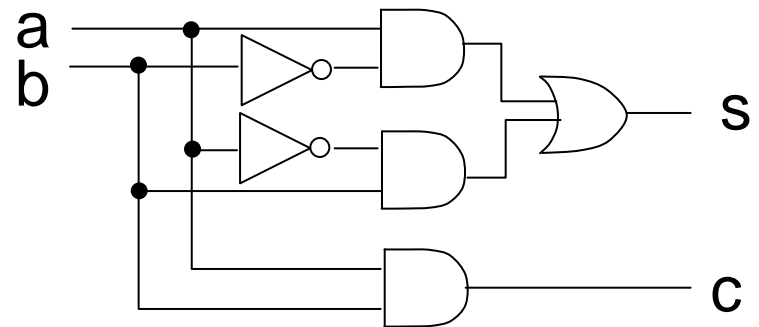
# 半加算器 (half adder)



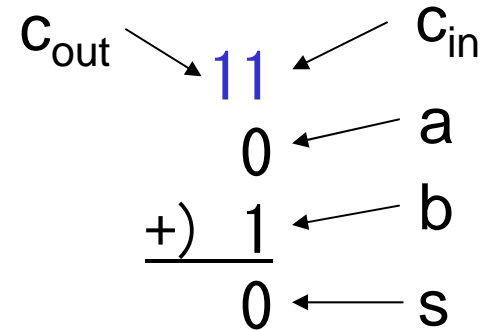
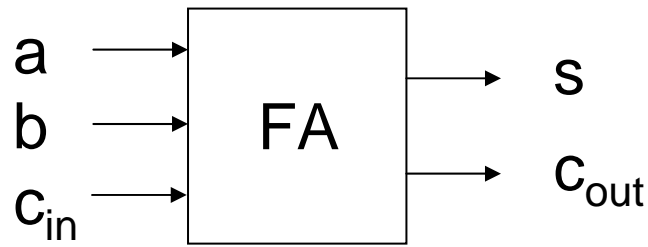
a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$s = \bar{a}b + a\bar{b} (= a \oplus b)$$

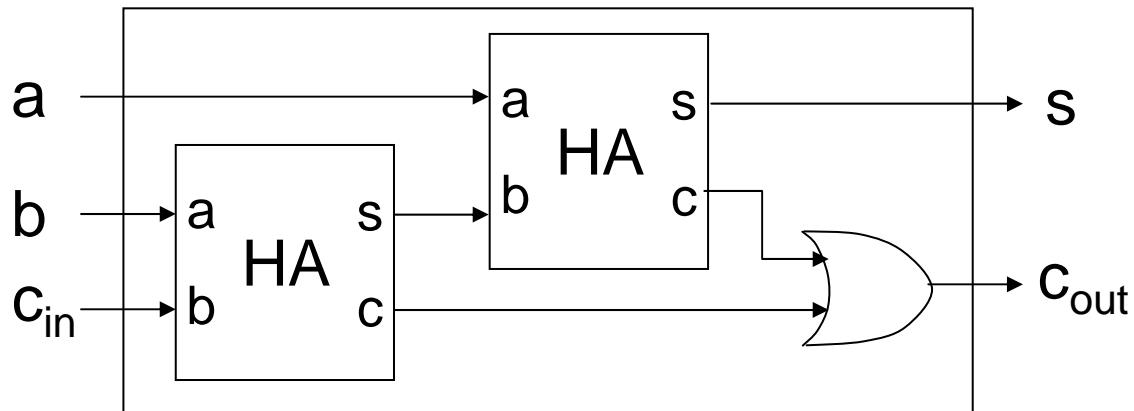
$$c = ab$$



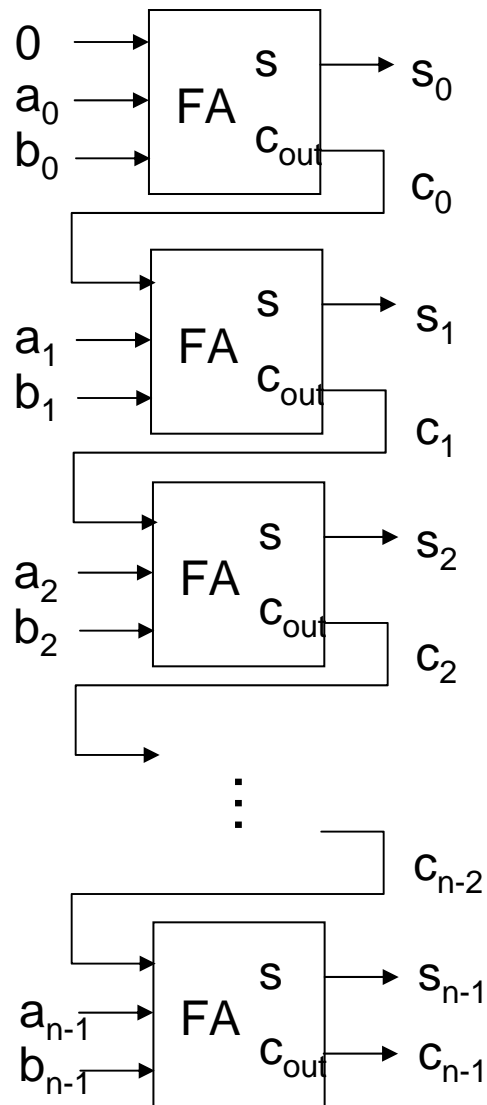
# 全加算器 (full adder)



前の位からの繰り上がりを考慮する. 半加算器が2つ必要



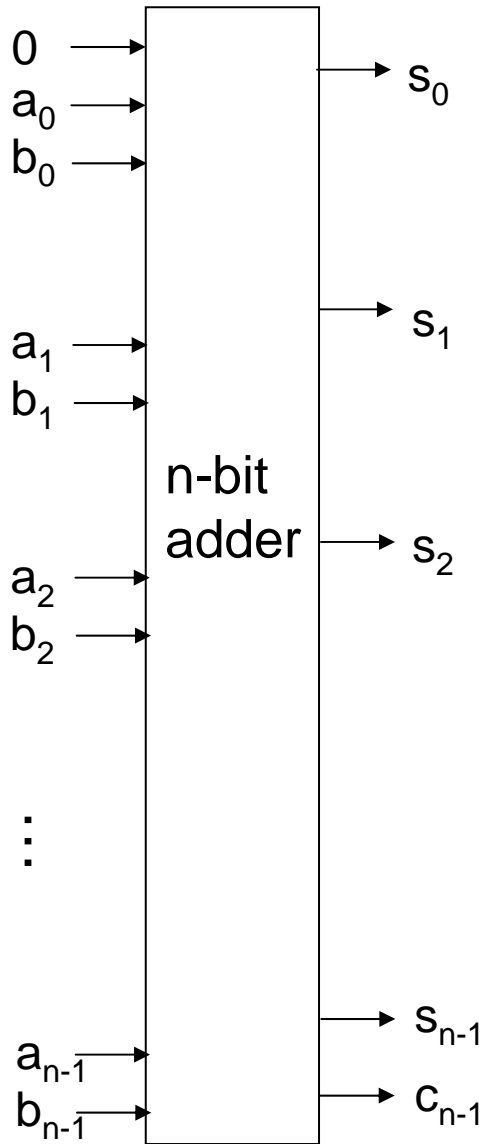
# n-ビット加算器



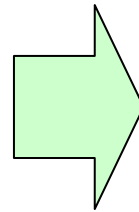
リプルキャリー型加算器と呼ばれる

- $n$ に比例して遅延が蓄積するため、決して速い回路ではない
- より高速な(しかし回路規模の大きい)加算回路も広く用いられている (e.g. キャリー先読み型加算器)

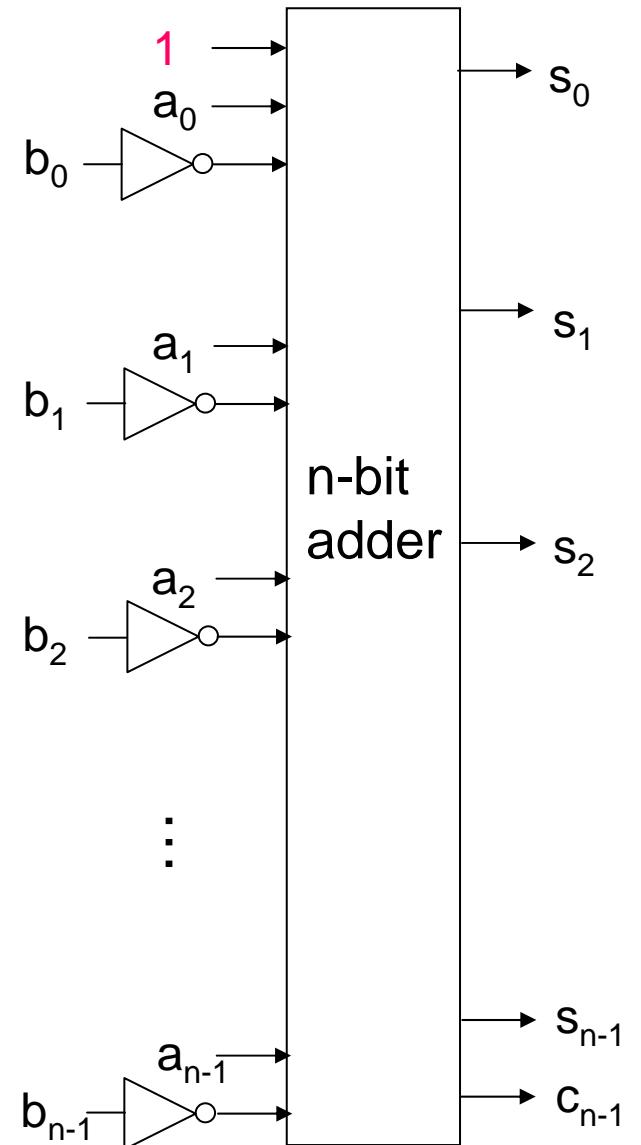
# n-ビット減算器



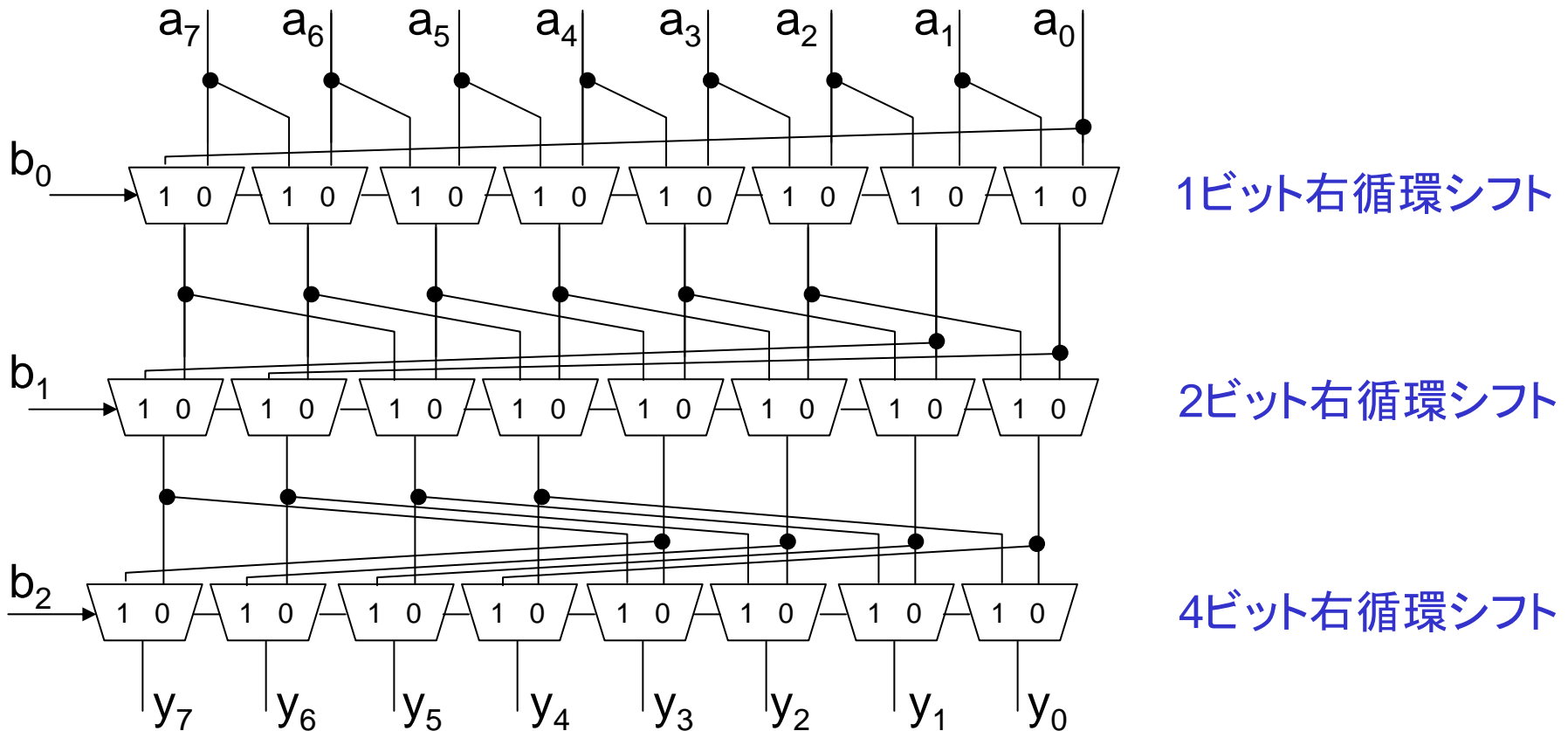
$$a - b = a + (-b)$$



入力を変えるだけで  
減算器になる



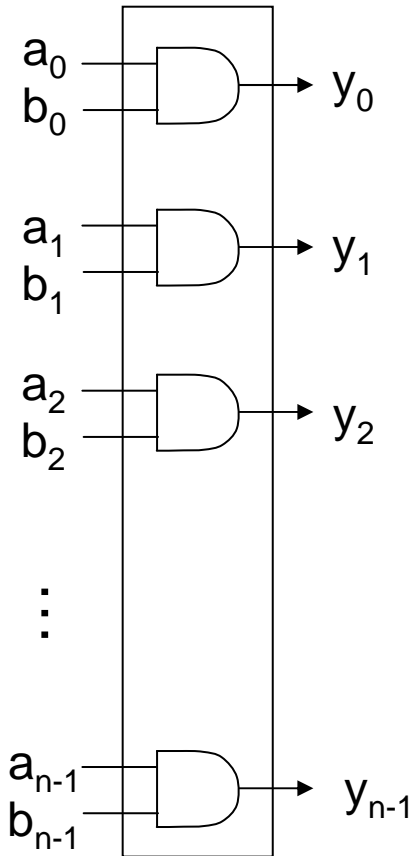
# バレルシフタ(ローテータ)



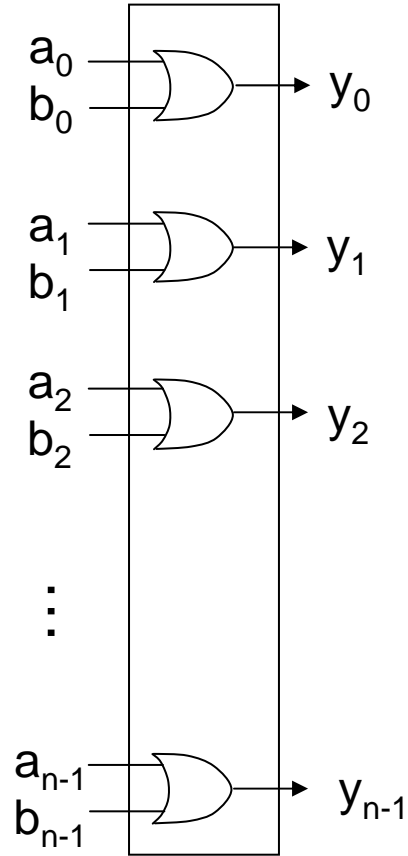
**Nビット値  $a$  を  $b$  ビット右循環シフトしたものを  $y$  として出力する組合せ回路**

- $b$  ビット左循環シフトは,  $N - b$  ビット右循環シフトと等価
- $b$  ビット左シフトは, 左循環シフト出力のLSB側  $b$  ビットを 0 にする
- $b$  ビット右シフトは, 右循環シフト出力のMSB側  $b$  ビットを 0 にする

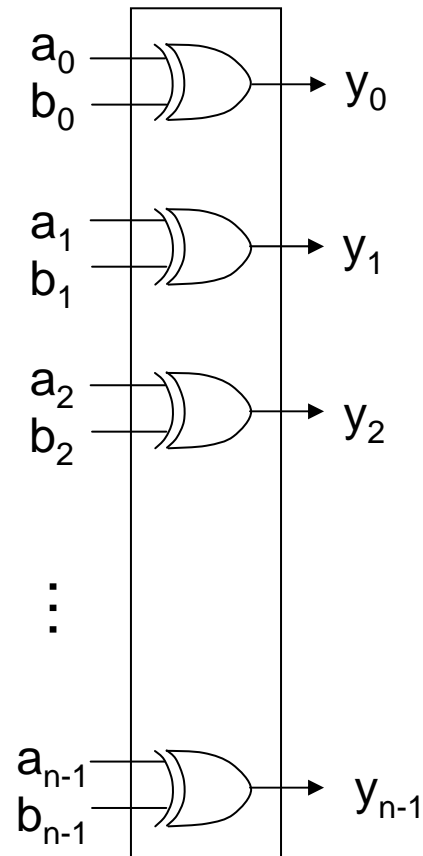
# ビットごと論理演算器



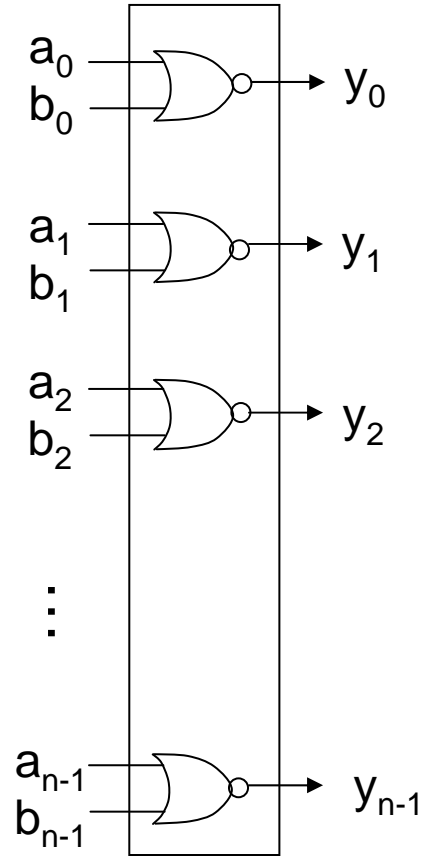
$$y = a \& b$$



$$y = a | b$$



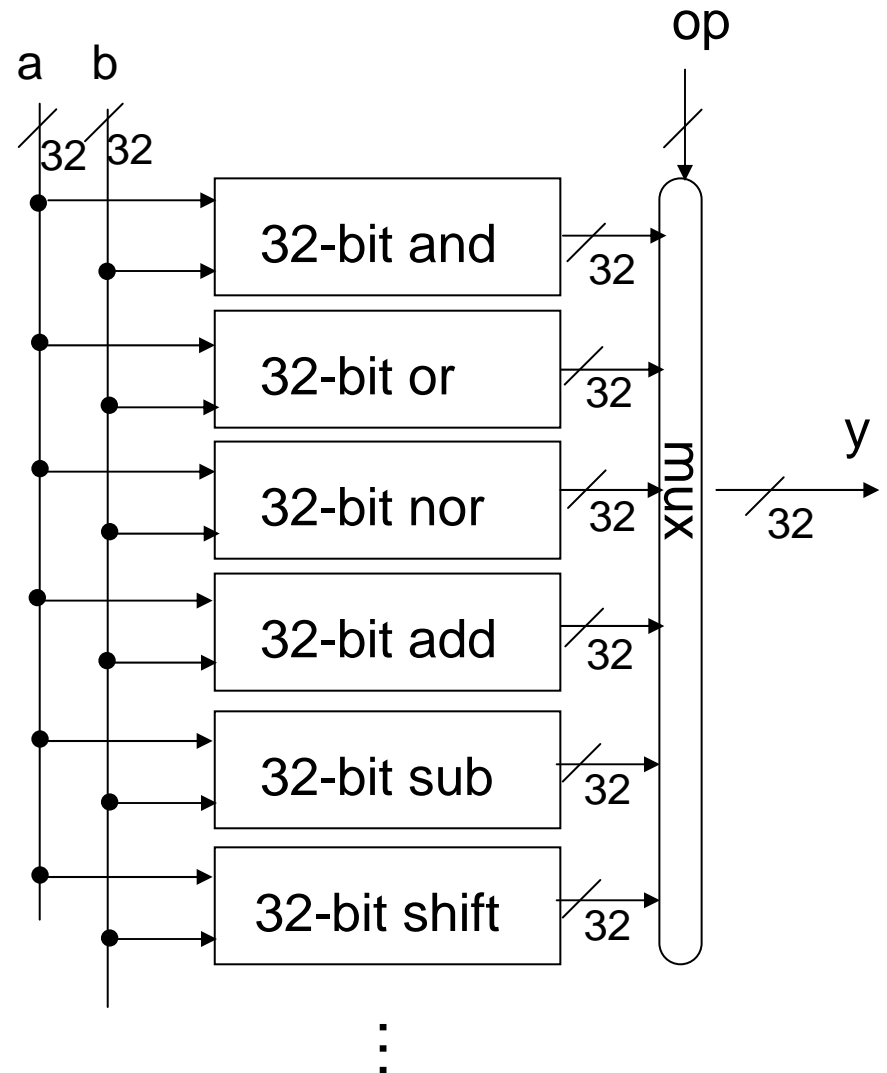
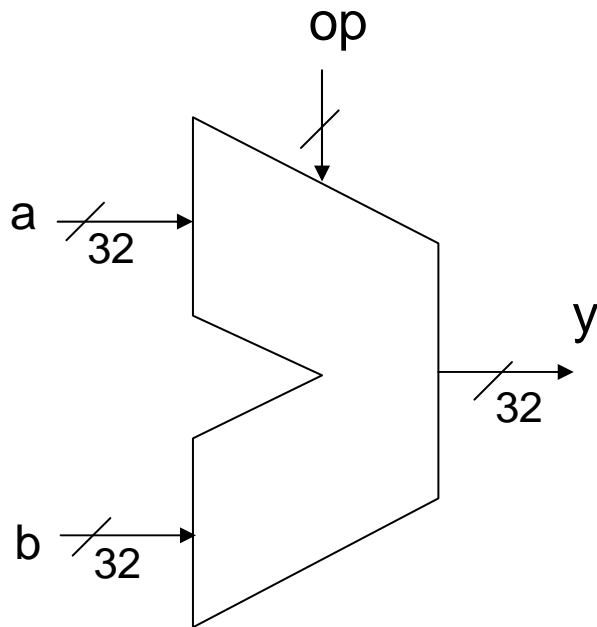
$$y = a \wedge b$$



$$y = \sim(a | b)$$



# ALU



# 参考

- 命令デコーダは, 32ビットの命令を入力として, 命令の解釈結果を出力する組合せ回路である. 出力信号は例えば:
  - 命令種別(レジスタ演算, 即値演算, ロード, ストア, 分岐)
  - レジスタ番号 rs
  - レジスタ番号 rt
  - レジスタ番号 rd
  - 即値・オフセット
  - オペコード
  - ...
- 「次PC計算」部は, 現在のPC値と2つのレジスタ値を入力として, 次のPCの値を出力する組合せ回路である. 内部では
  - 分岐条件の判定と
  - 次のPCの計算を行う
- 構成例: 教科書付録E章

# 練習問題

1. 2入力マルチプレクサ  $m(a_0, a_1, s)$  を主加法標準形の論理式で表せ.
2.  $m(a_0, a_1, s)$  のカルノー図をかき, できるだけ簡単な積和型の論理式で表せ. またその論理回路図を示せ.
3. 全加算器の両出力  $s(a, b, c_{in}), c_{out}(a, b, c_{in})$  のカルノー図をかき, それぞれをできるだけ簡単な積和型の論理式で表せ.

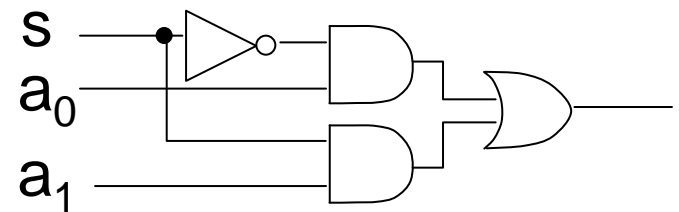
# 解答例

1.  $m = \bar{a}_0 a_1 s + a_0 \bar{a}_1 \bar{s} + a_0 a_1 \bar{s} + a_0 a_1 s$

2.

m		a0 a1			
		00	01	11	10
s	0			1	1
	1		1	1	

$$m = a_0 \bar{s} + a_1 s$$



3.

s		a b			
		00	01	11	10
cin	0		1		1
	1	1		1	

cout		a b			
		00	01	11	10
cin	0			1	
	1		1	1	1

$$s = \bar{a} \bar{b} c_{in} + \bar{a} b \bar{c}_{in} + a b c_{in} + a \bar{b} \bar{c}_{in} \quad (3\text{入力 XOR})$$

$$c_{out} = ab + bc_{in} + c_{in}a \quad (3\text{入力多数決関数})$$