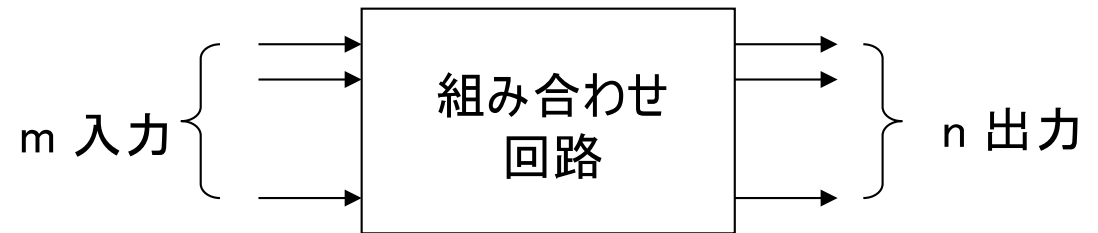

第5章「組み合わせ回路」

- ・ 組み合わせ回路
- ・ 回路の簡単化(カルノーマップ)
- ・ 加算回路(半加算器, 全加算器)
- ・ 比較回路

組み合わせ回路とは

- ・ 論理関数を実現する論理回路のこと
 - その時点の入力だけで出力がきまる静的なもの（ \Leftrightarrow 順序回路）
 - 入出力の数は一般に複数

$$\begin{aligned} z_1 &= f_1(x_1, \dots, x_m) \\ &\vdots \\ z_n &= f_n(x_1, \dots, x_m) \end{aligned}$$



- ・ 論理関数の簡単化
 - 1) 公式の適用による式変形
 - 2) カルノーマップ

カルノーマップ(ベイチ図)

	A	\bar{A}
B	AB	$\bar{A}B$
\bar{B}	A \bar{B}	$\bar{A}\bar{B}$

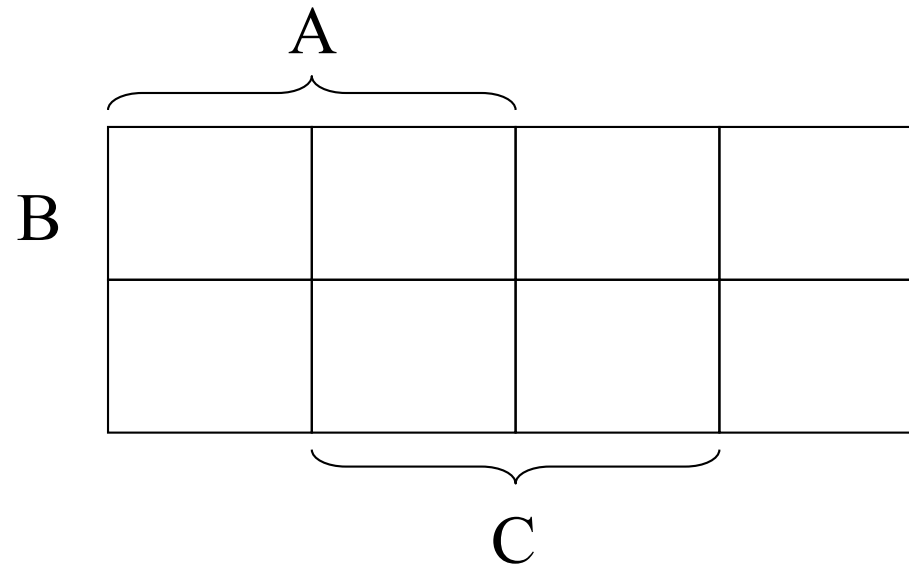
	A			
B	AB \bar{C}	ABC		
	C			

	A			
B			$\bar{A}BC\bar{D}$	
		ABCD	$\bar{A}BCD$	
	C			
				D

欄1つが1つの最小項
に対応

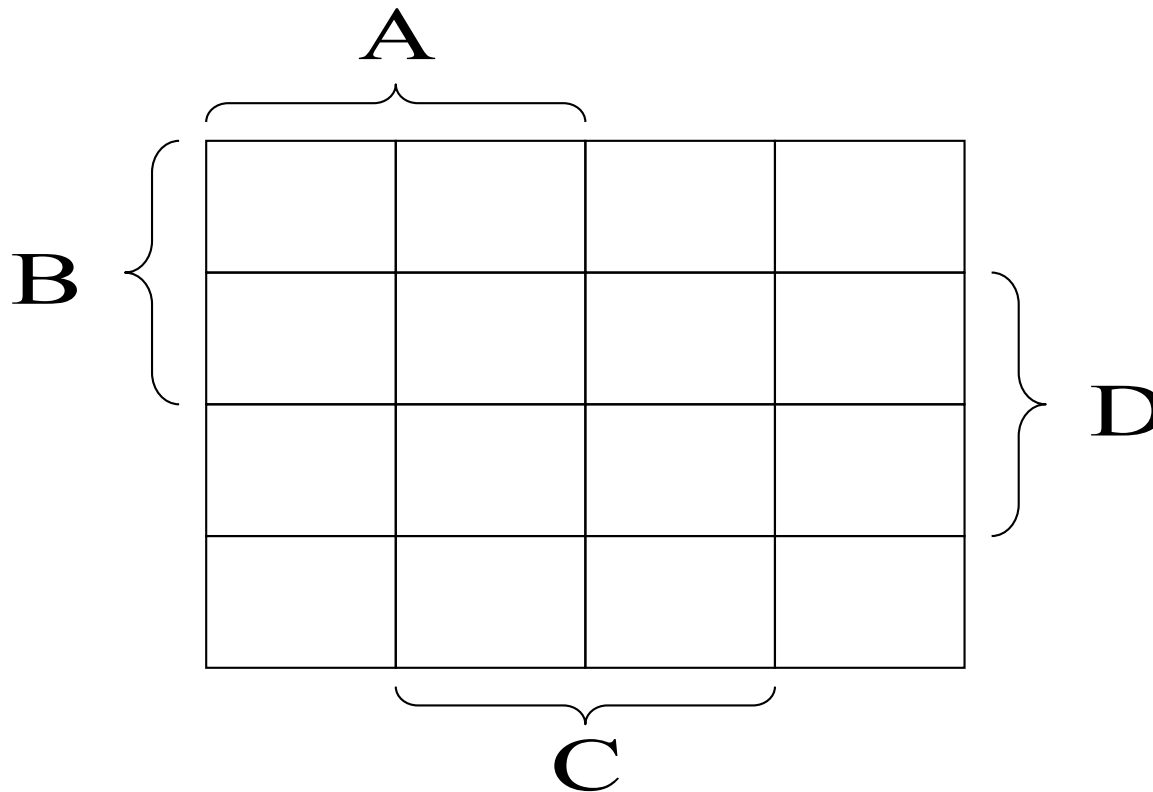
例(3変数)

$$f(A, B, C) = ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$$

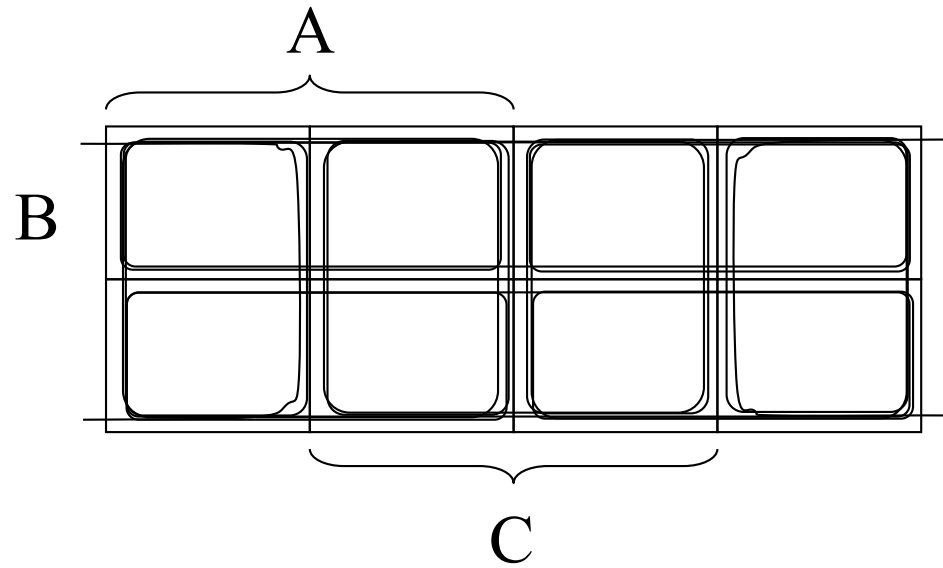


例(4変数)

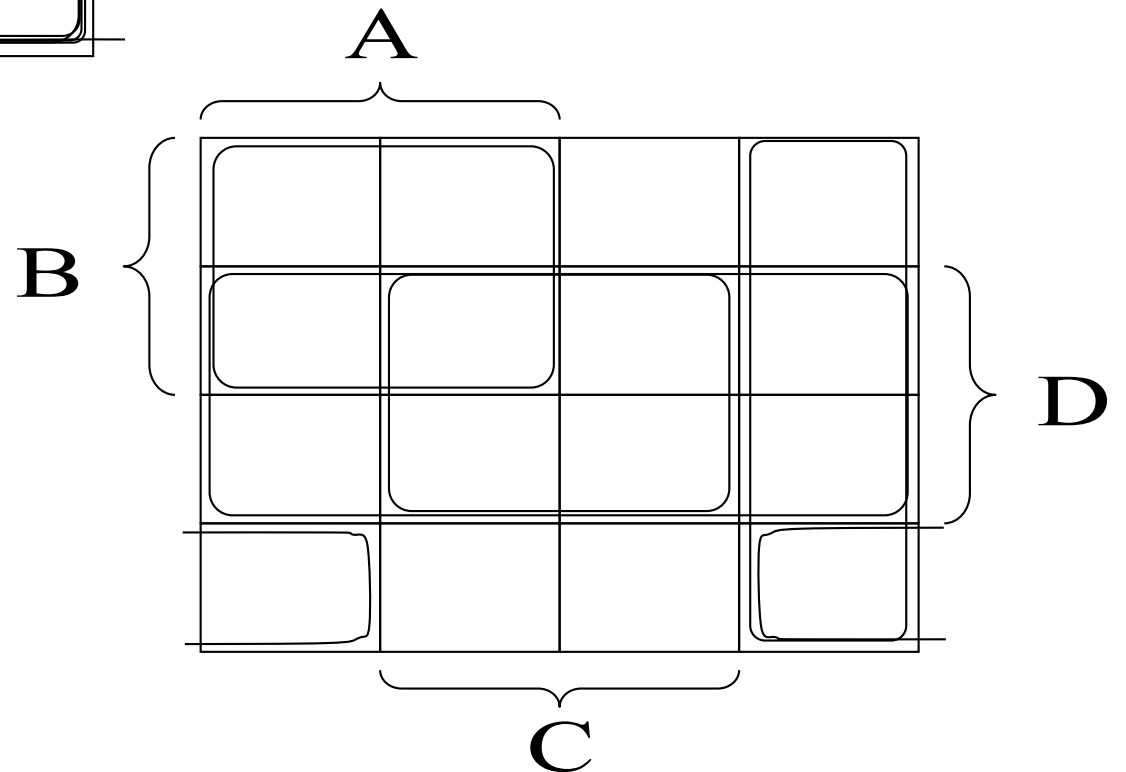
$$f(A, B, C, D) = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}BC\bar{D} + ABC\bar{D}$$



ループ



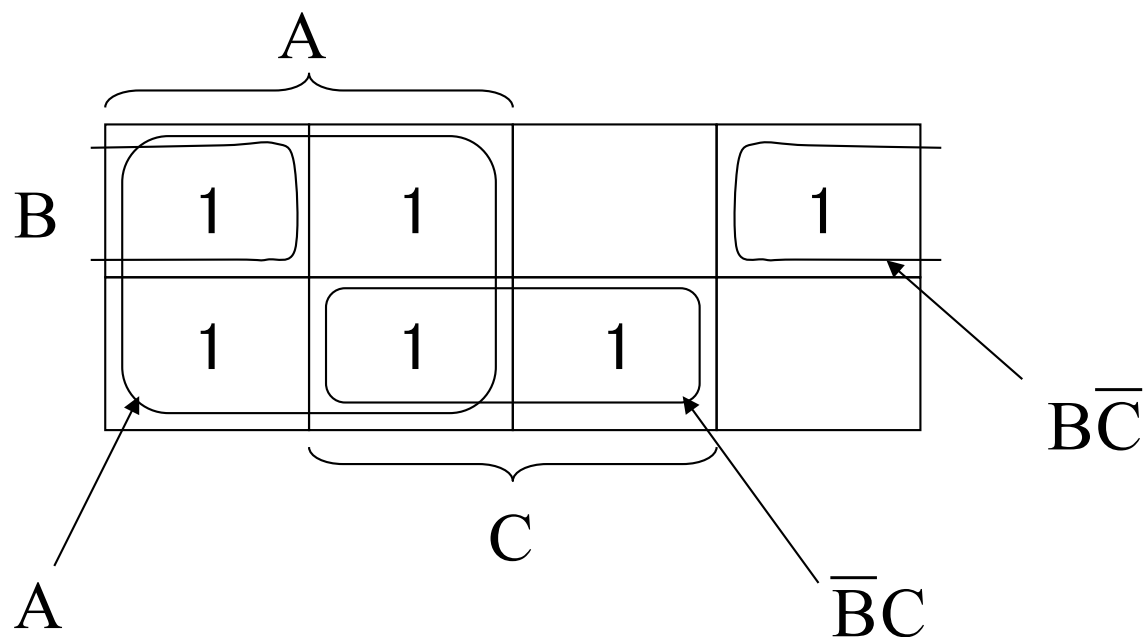
各ループと論理積の項が対応



カルノーマップ(ベイチ図)

- 1のみを内部に含む出来るだけ大きなループの組み合わせを見つける→簡単化

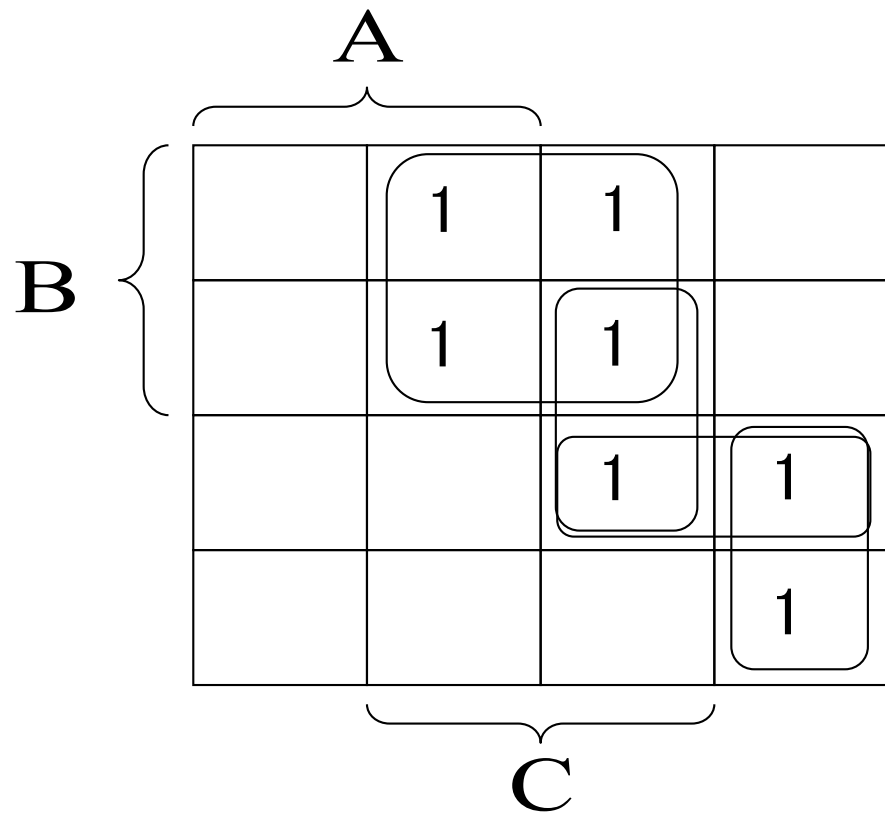
$$f(A, B, C) = ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$$



$$f(A, B, C) = A + \bar{B}C + B\bar{C}$$

例(4変数)

$$f(A, B, C, D) = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}BC\bar{D} + \bar{A}BCD + ABC\bar{D} + ABCD$$



$$D = BC + \bar{A}\bar{B}\bar{C} + \bar{A}CD$$

あるいは

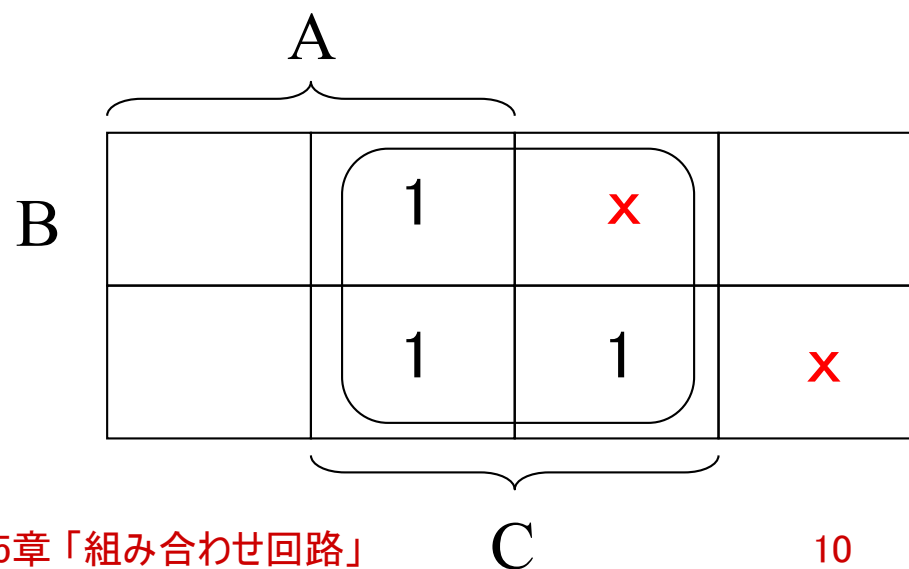
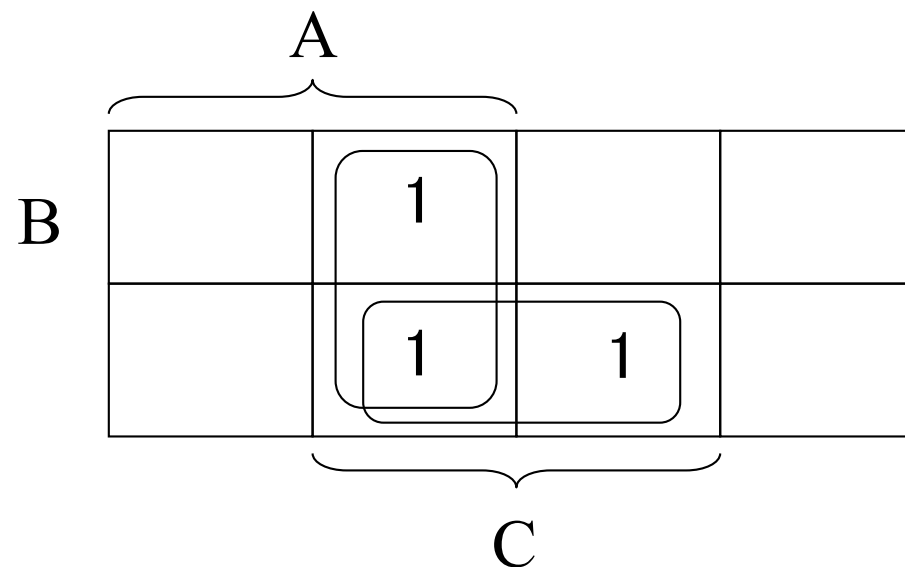
$$= BC + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}D$$

組み合わせ禁止

- ・ ある変数の組が決して生じない場合がある
- ・ そのような組を、組み合わせ禁止あるいは”don't care”と呼ぶ
- ・ その組の値は 0 でも 1 でもよいことになり、簡略化の際に利用できる

組み合わせ禁止のある場合の簡略化

A	B	C	f(A,B,C)
0	0	0	X
0	0	1	1
0	1	0	0
0	1	1	X
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



2進nビットの加減算

各ビット i における...

$$\begin{array}{r} 00101 \\ + 10110 \\ \hline \end{array}$$

$$\begin{array}{r} 011010 \\ + 011100 \\ \hline \end{array}$$

被加算数: X_i
加算数: Y_i
和: S_i
桁上がり: C_i

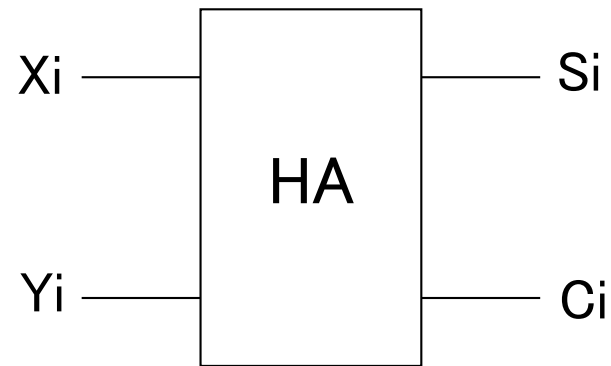
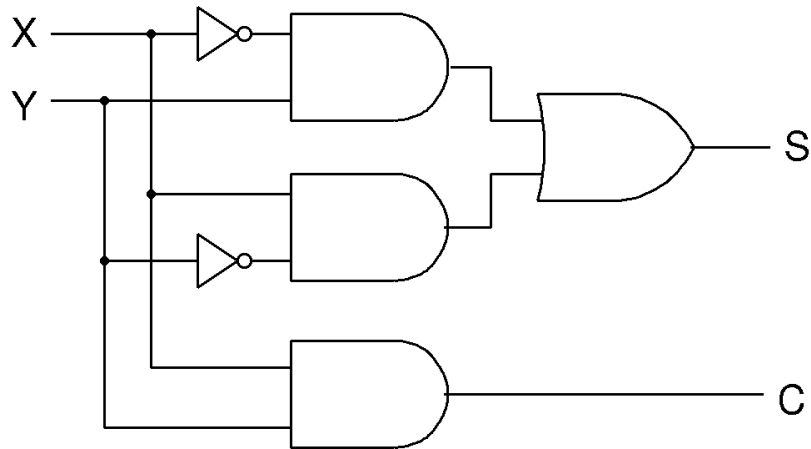
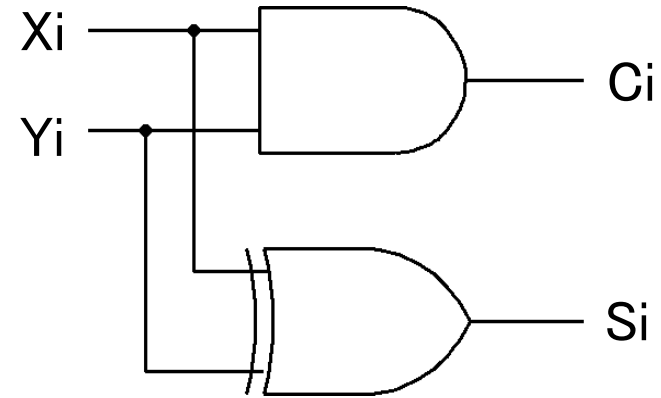
$$\begin{array}{r} 11011 \\ - 10110 \\ \hline \end{array}$$

$$\begin{array}{r} 110110 \\ - 011100 \\ \hline \end{array}$$

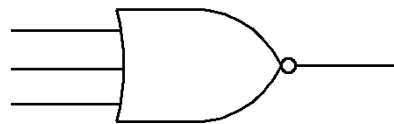
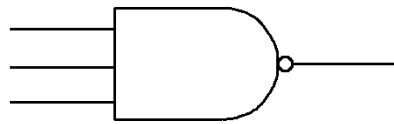
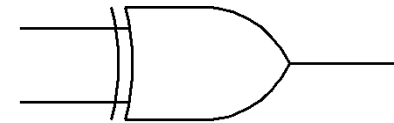
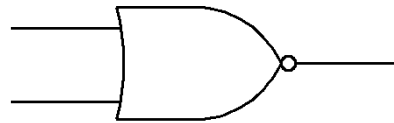
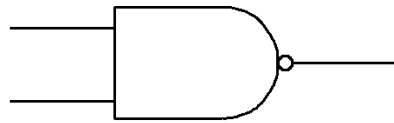
被減算数: X_i
減算数: Y_i
差: D_i
桁借り: B_i

半加算器 (half adder)

X_i	Y_i	S_i	C_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



NAND, NOR, 排他的論理和 (XOR) の回路記号



XOR

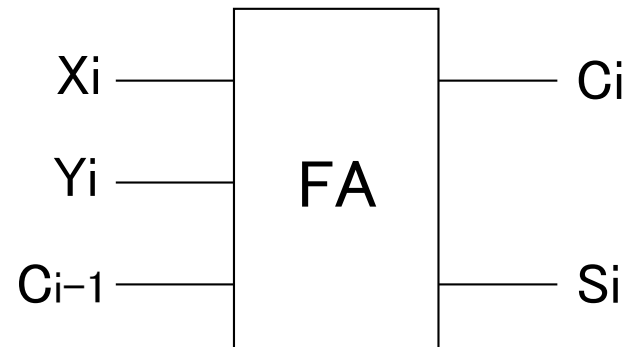
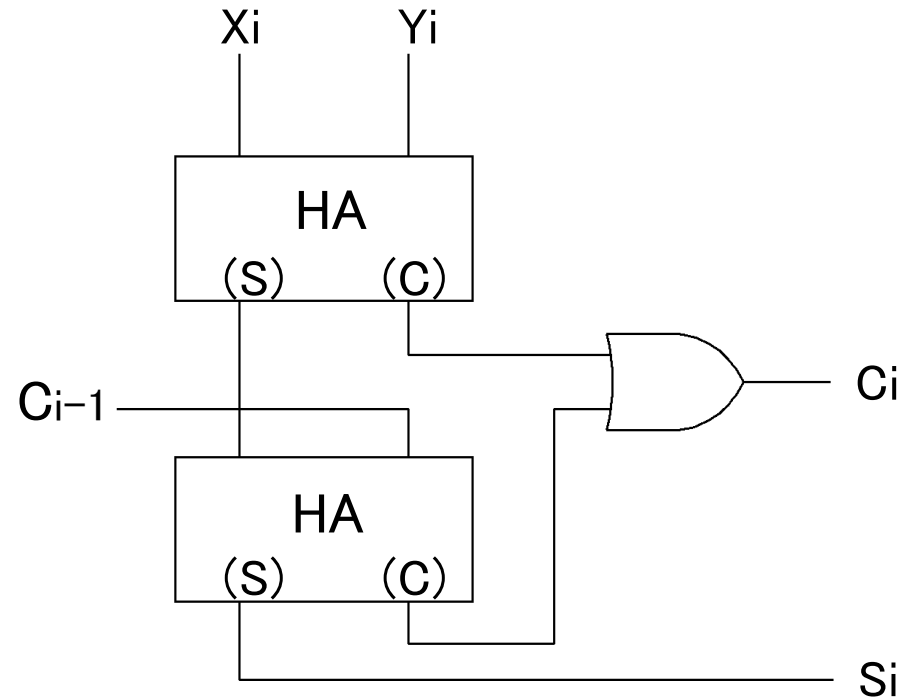
2入力NAND
3入力NAND

2入力NOR
3入力NOR

A	B	NOR	NAND	XOR
0	0	1	1	0
0	1	0	1	1
1	0	0	1	1
1	1	0	0	0

全加算器 (full adder)

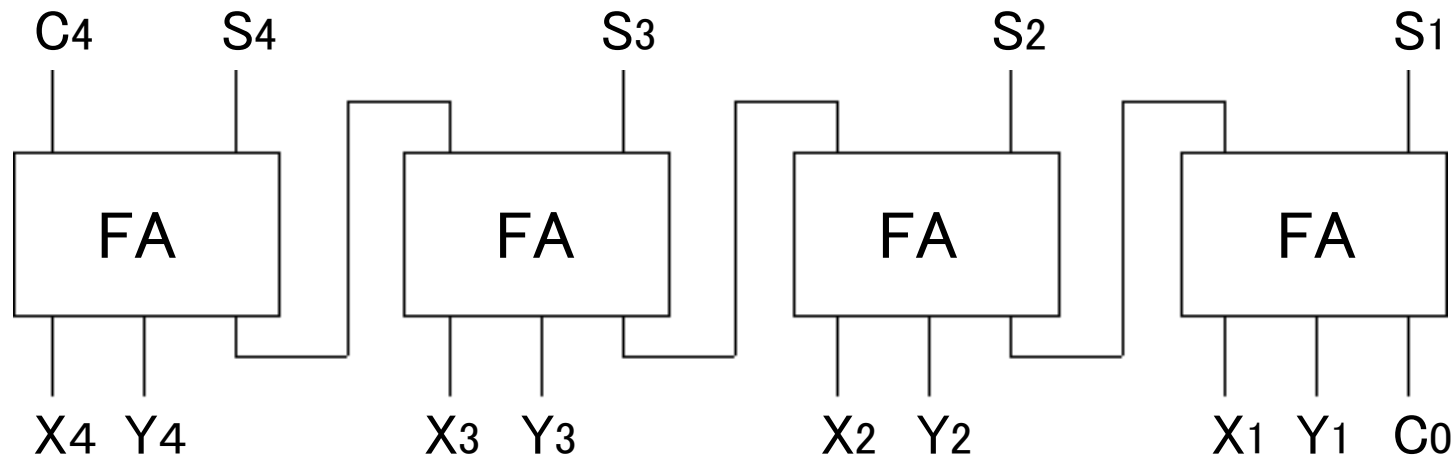
X_i	Y_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



2進nビット加減算器

- 全加算器のカスケード接続によって実現できる

例) 4ビット加算器



減算に関する注

- 負の数を2の補数で表現していれば加算で実現できる
- 1の補数はビット反転で, 2の補数は1を加算すれば良い
(→C₀入力を使える)

2進比較器

- ・ n ビット2進数の大小を比較 $X < Y$ なら 1, $X \geq Y$ なら 0

X_i	Y_i	F_{i-1}	F_i
0	0	0	0
0	1	0	1
1	0	0	0
1	1	0	0
0	0	1	1
0	1	1	1
1	0	1	0
1	1	1	1

例) X 010101
 Y 011011

 F 111010

