論理式の簡単化

休講について

- ５月２０日（金）

前回までのまとめ（１）

- データ（数、文字、画像など）は論理値(0,1)の集まりとして表現
- データの処理は複数のn変数論理関数として表現

$$f: B^n \to B$$

前回までのまとめ（2）

- 論理関数の表現法
  - 真理値表

a b c	C	S	十進数
0 0 0	0	0	0
0 1 0	0	1	1
1 0 0	0	1	1
1 1 0	1	0	2
0 0 1	0	1	1
0 1 1	1	0	2
1 0 1	1	0	2
1 1 1	1	1	3

前回までのまとめ（3）

- 論理関数の表現法
  - 論理式
    - 基本論理関数（**NOT, AND, OR**等）の組み合わせで表現

      ~f:　　否定(NOT)

      f･g:　論理積(AND)

      f+g:　論理和(OR)

      例：

      C(a,b,c) = ab~c+a~bc+~abc + abc

任意の論理関数は**NOT, AND, OR**の組み合わせで表現可能（**{NOT,AND,OR}**は完全系**)**

⇒**NOT, AND, OR**を計算する回路が作ることができれば任意の論理回路を関数として実現可能

前回までのまとめ（4）

- 論理式の標準形

$$f(x1,\ldots,xn) = \Sigma \quad f(e1,\ldots,en)\ x1^{e1}\ldots xn^{en}$$

$$\boxed{= \Sigma_{f(e1,\ldots,en)=1} \quad x1^{e1}\ldots xn^{en}}$$

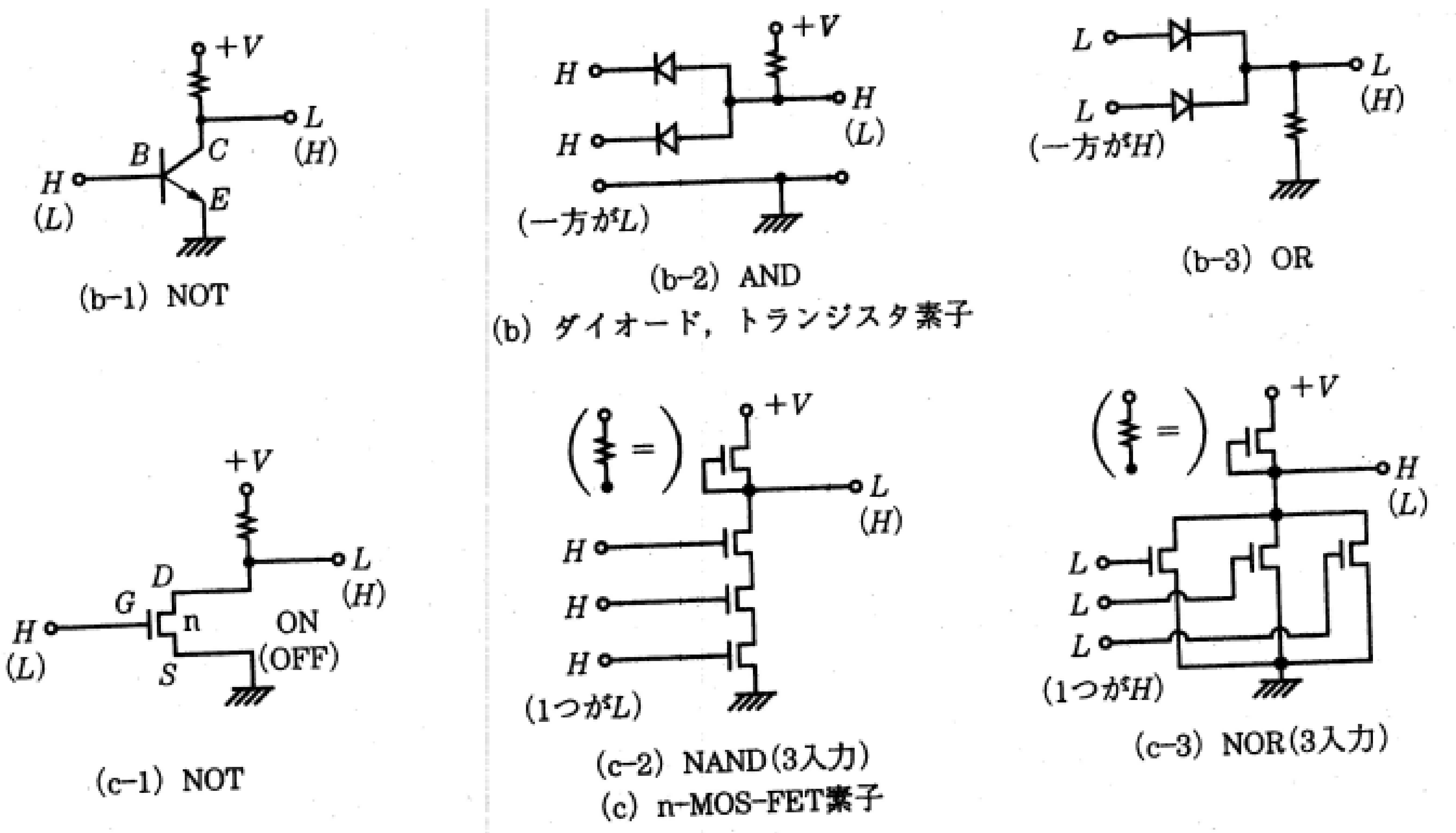
積和標準形
最小項展開

最小項

例：**full adder のキャリー部分の論理関数の最小項展開**

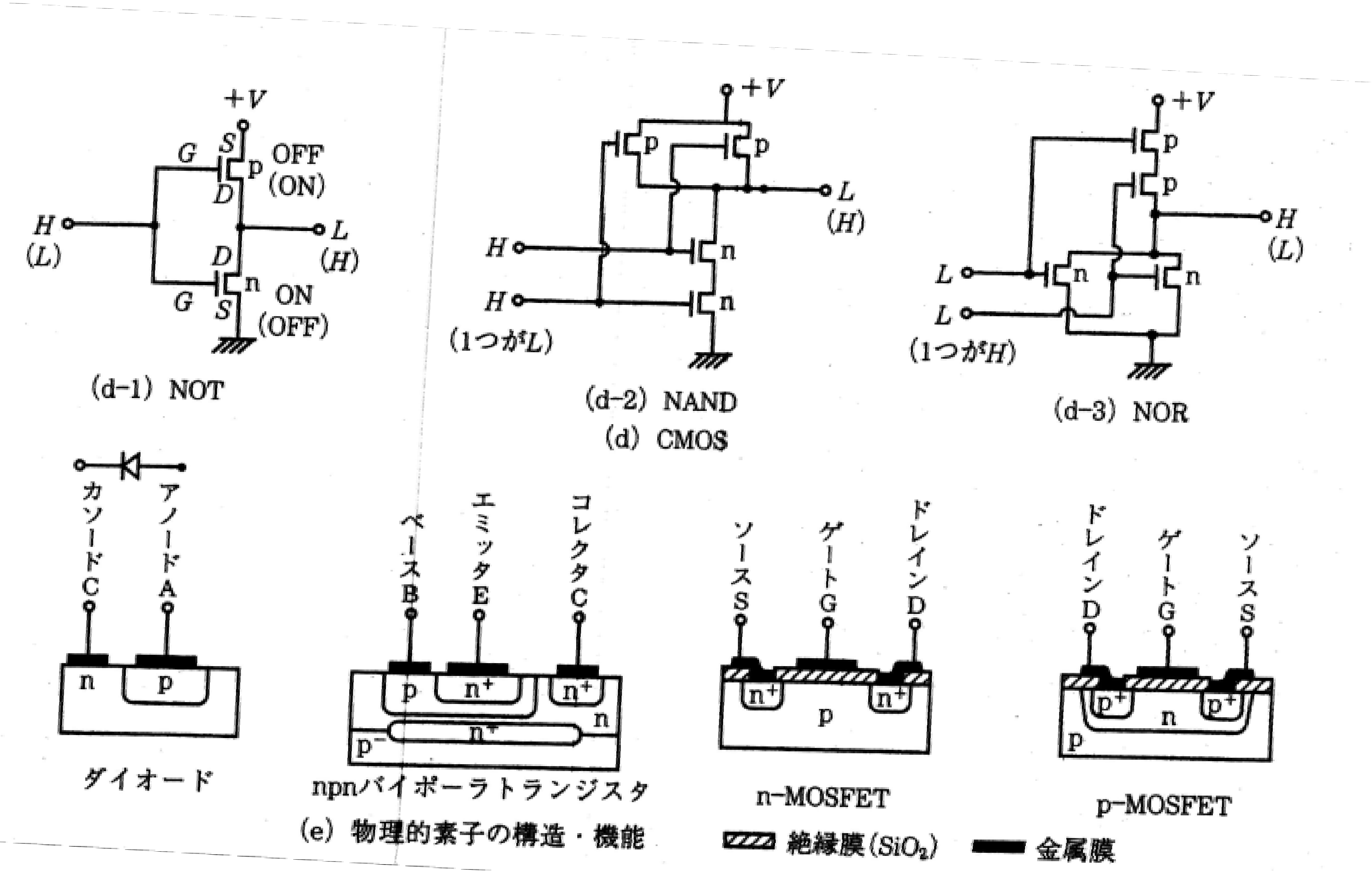
C(a,b,c) = C(0,0,0)~a~b~c + C(0,0,1)~a~bc + C(0,1,0)~ab~c
+C(0,1,1)~abc + C(1,0,0)a~b~c+ C(1,0,1)a~bc
+C(1,1,0)ab~c + C(1,1,1)abc
= ~abc + a~bc + ab~c + abc

論理素子の構成



+V
H
(L)
B
C
E
L
(H)
(b-1) NOT
H
H
(一方がL)
+V
H
(L)
(b-2) AND
L
L
(一方がH)
L
(H)
(b-3) OR
(b) ダイオード，トランジスタ素子
+V
H
(L)
G
D
S
n
ON
(OFF)
L
(H)
(c-1) NOT
+V
L
(H)
H
H
H
(1つがL)
(c-2) NAND(3入力)
+V
H
(L)
L
L
L
(1つがH)
(c-3) NOR(3入力)
(c) n-MOS-FET素子

論理素子の構成（2）



+V
G
S
D
p
n
OFF
(ON)
ON
(OFF)
H
(L)
L
(H)
(d-1) NOT
(1つがL)
(d-2) NAND
(1つがH)
(d-3) NOR
(d) CMOS
カソードC
アノードA
ダイオード
ベースB
エミッタE
コレクタC
npnバイポーラトランジスタ
ソースS
ゲートG
ドレインD
n-MOSFET
p-MOSFET
(e) 物理的素子の構造・機能
絶縁膜(SiO_2)
金属膜

今日の講義内容

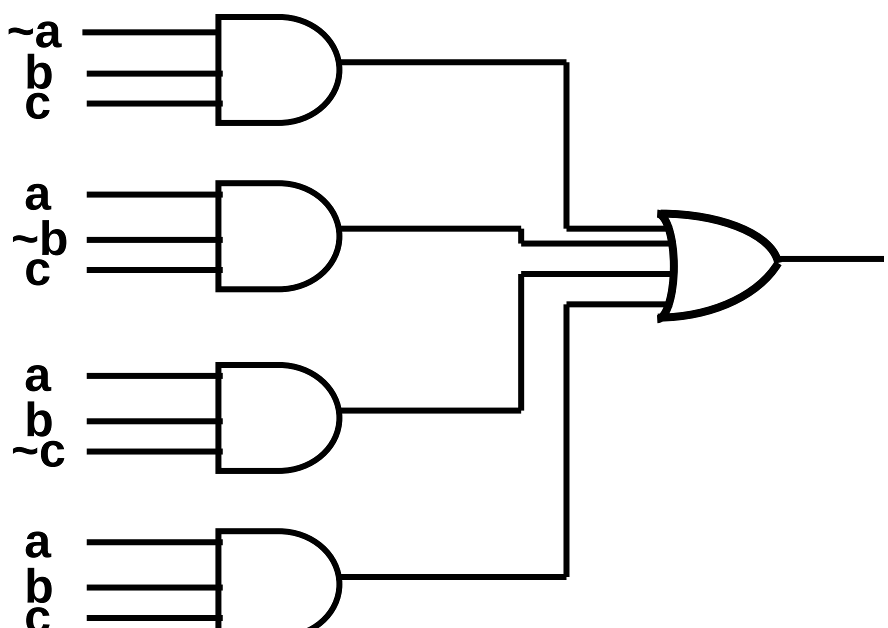
- 論理式の論理回路への変換
- 論理式の簡単化とその意義
- 論理式の簡単化の準備
  - 論理式に関する基本法則
  - 用語の定義（部分積項、主項、論理式の順序関係）
- カルノー図による論理式の簡単化

今日の講義内容

- 論理式の論理回路への変換
- 論理式の簡単化とその意義
- 論理式の簡単化の準備
  - 論理式に関する基本法則
  - 用語の定義（部分積項、主項、論理式の順序関係）
- カルノー図による論理式の簡単化

論理式の論理回路への変換

- 前回学んだNOT, AND, OR, NOR, NAND素子を使用
  – 例：C(a,b,c) = ~abc + a~bc + ab~c + abc



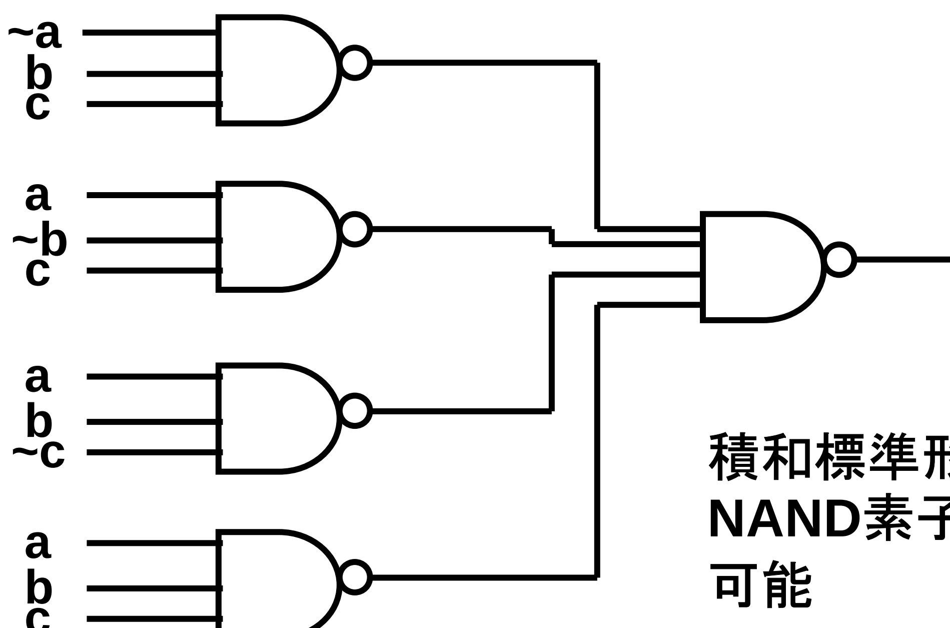
論理式の論理回路への変換

- C(a,b,c) のNAND素子による実現

~abc + a~bc + ab~c + abc

= ~~(~abc + a~bc + ab~c + abc)

= ~(~(abc) ･~(a~bc)･~(ab~c)･~(abc))



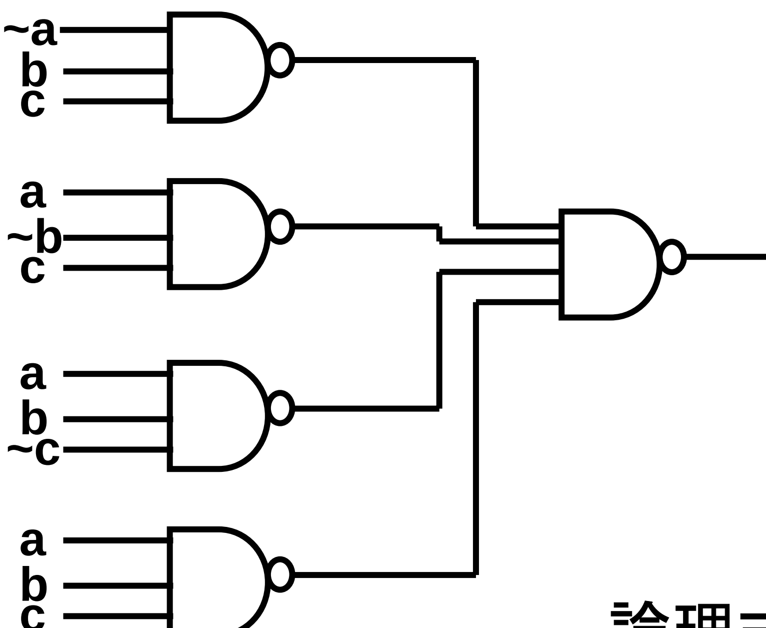
積和標準形の論理式は同様にNAND素子による回路に変換可能

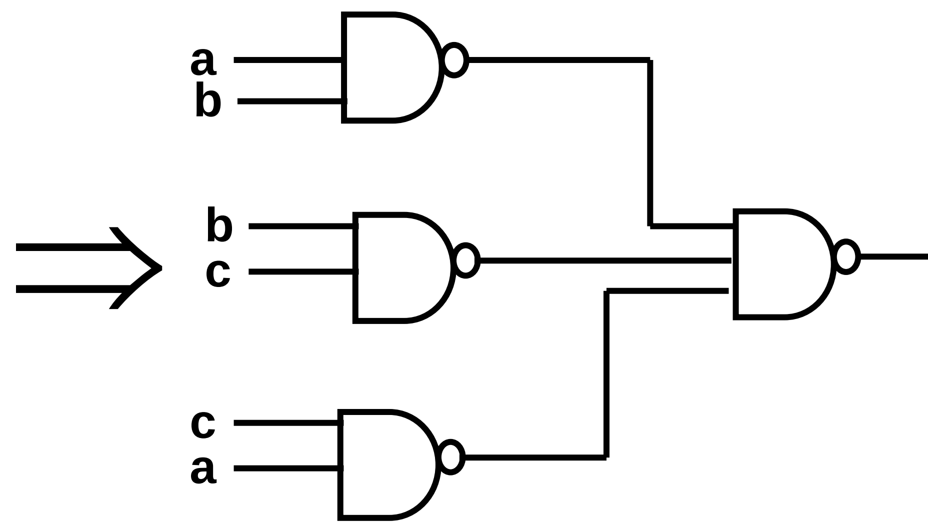
今日の講義内容

- 論理式の論理回路への変換
- 論理式の簡単化とその意義
- 論理式の簡単化の準備
  - 論理式に関する基本法則
  - 用語の定義（部分積項、主項、論理式の順序関係）
- カルノー図による論理式の簡単化

論理式の簡単化

- C(a,b,c) = ~abc + a~bc + ab~c + abcは、
  ab + bc + caと等価





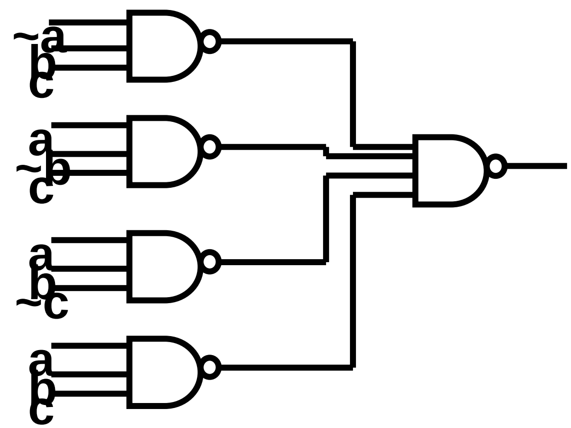
論理式の簡単化が論理回路の簡単化に

論理式の簡単化の効果

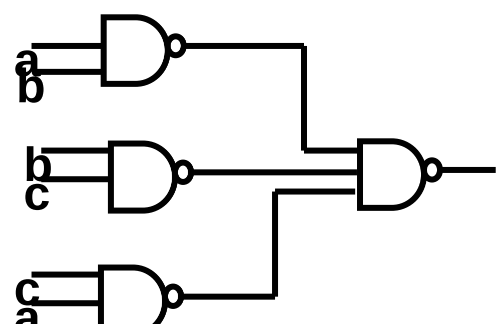
- 対応する論理回路中の論理素子数および配線数の減少
  - 集積効果：一つのチップにより多くの機能を集積
  - 電力消費量
  - 発熱量の減少
  - 演算の高速化

論理式の簡単化の基準その1

- 積和標準形で表したときの積項の数(=論理回路で表した時の一段目の論理素子数）が少ない
  - **~abc + a~bc + ab~c + abc**：４つ

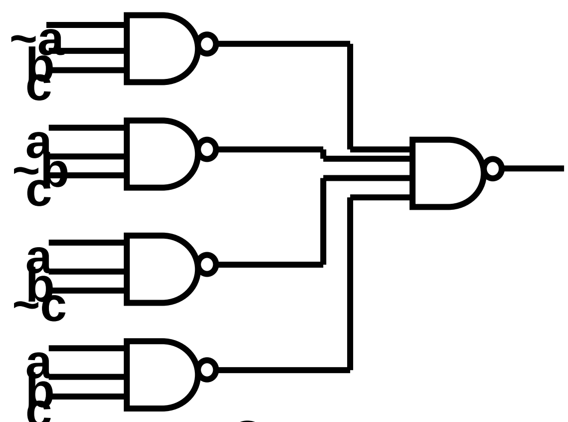


  - **ab + bc + ca :** ３つ

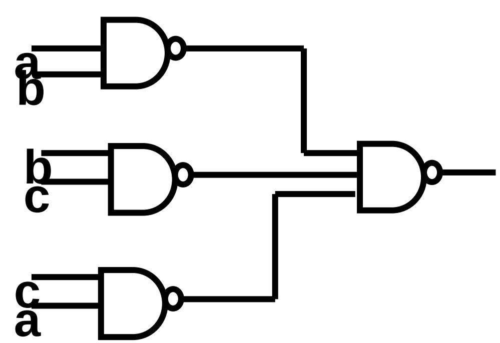


論理式の簡単化の基準その2

- 積和標準形で表したときのリテラル（変数またはその否定）の数(=論理回路の一段目の論理素子数へ配線数）が少ない
  - **~abc + a~bc + ab~c + abc：12**

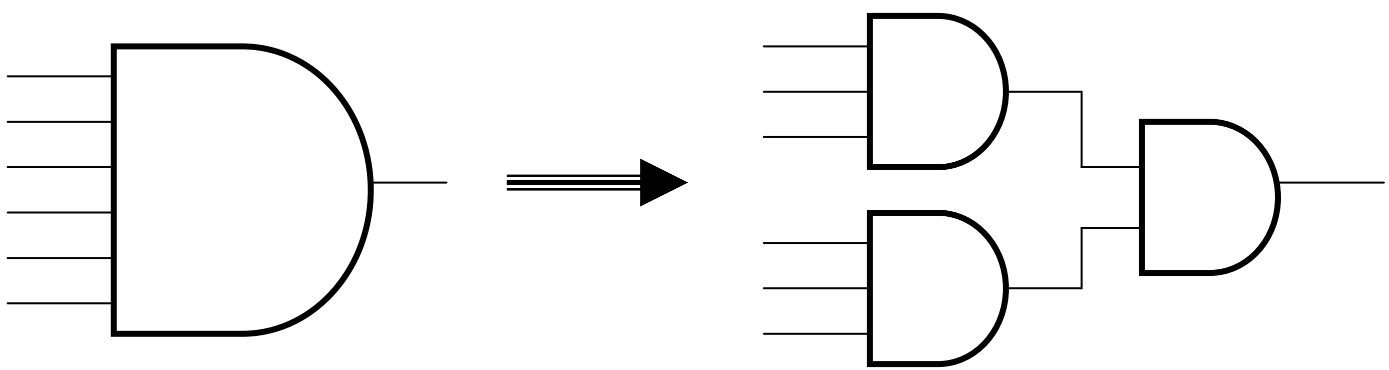


  - **ab + bc + ca : 6**



ファンインの制限と論理式の簡単化の効果

- ファンインの制限：AND, OR等の各論理素子には電気的理由から入力数に制限
  - →入力数が多いと一つの論理積・論理和を複数の論理素子で実現する必要あり



今日の講義内容

- 論理式の論理回路への変換
- 論理式の簡単化とその意義
- 論理式の簡単化の準備
  - 論理演算の性質
  - 用語の定義（部分積項、主項、論理式の順序関係）
- カルノー図による論理式の簡単化

論理演算の性質（教科書p.39)

- 1に関する法則

  x･1= 1･x = x　　　　x+1 = 1+x = 1
- 0に関する法則

  x･0= 0･x = 0　　　　x+0 = 0+x = x
- べき等律　xx = x+x = x
- 交換律　xy=yx　x+y = y+x
- 結合律　(xy)z = x(yz)　(x+y)+z =x+(y+z)
- 分配律　x(y+z) = xy+xz　x+yz = (x+y)(x+z)
- 相補律　x~x = 0　x+~x = 1
- 二重否定　~~x = x
- ド･モルガン律　~(xy) = ~x+~y　~(x+y) = ~x~y

論理演算の性質

- 例題 (x+y)(x+~y)(~x+y)の簡単化

$$
\begin{aligned}
& (x+y)(x+\sim y)(\sim x+y) \\
= \ & (xx + x\sim y + yx + y\sim y)(\sim x+y) && \text{（分配律）} \\
= \ & (x+x\sim y + yx)(\sim x+y) && \text{（べき等律、相補律、0）} \\
= \ & x\sim x+xy+x\sim y\sim x + x\sim yy+yx\sim x+yxy && \text{（分配律）} \\
= \ & xy + yxy && \text{（相補律、交換律、0）} \\
= \ & xy && \text{（べき等律、交換律）}
\end{aligned}
$$

今日の講義内容

- 論理式の論理回路への変換
- 論理式の簡単化とその意義
- 論理式の簡単化の準備
  - 論理演算の性質
  - 用語の定義（部分積項、主項、論理式の順序関係）
- カルノー図による論理式の簡単化

積和標準形、和積標準形

- リテラル：変数またはその否定**(x, ~x)**
- 積和標準形：リテラルの積の和の形をした論理式（例：　**ab~c + a~bc + ｃ + ab**）
  - **AND-OR, NAND-NAND**の２段階**(+NOT)**で実現可能
- 和積標準形：リテラルの和の積の形をした論理式（例：　**(a+b)(b+~c)(a+~b+c))**
  - **OR-AND, NOR-NOR**の２段階（**+NOT**）で実現可能

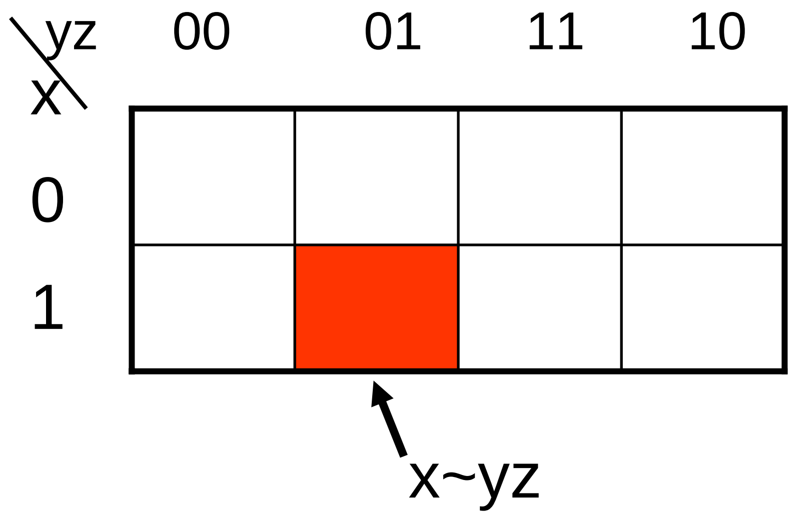
論理回路の実現には論理関数を簡単な和積標準形または積和標準形に変形すればよい

最小項、最大項

- 最小項：すべての入力変数をちょうど一回含むリテラルの積

  例（入力変数がx,y,zの場合）：

  ○xyz　　○ x~yz　　× xy

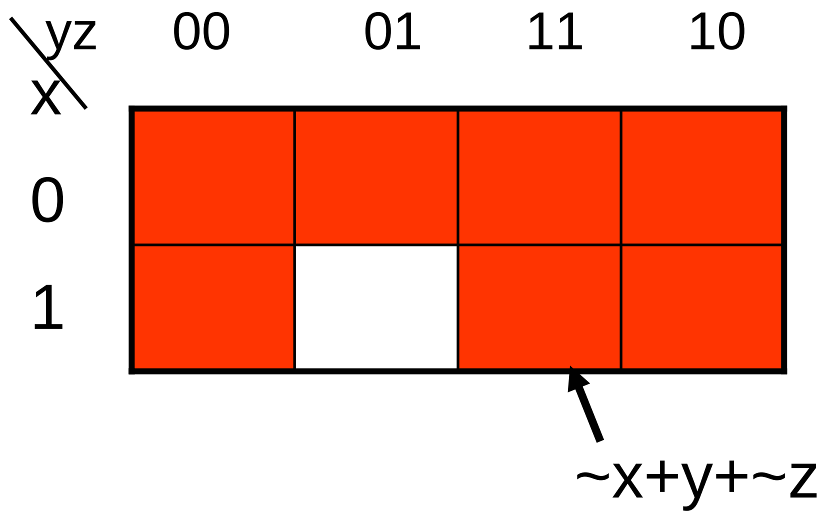


最小項、最大項

- 最大項：すべての入力変数をちょうど一回含むリテラルの和

  例（入力変数がx,y,zの場合）：

  ○x+y+z　　○ ~x+y+~z　　× x+y



最小項展開と最大項展開

- 最小項展開：最小項の和の形で表したもの
- 最大項展開：最大項の積の形で表したもの

例：C(a,b,c) = ab + bc + ca

最小項展開：abc+ab~c+~abc+a~bc

最大項展開: (a+b+c)(~a+b+c)(a+~b+c)(a+b+~c)

カルノー図による方法とクワインーマクラスキー法に基づく論理式の簡単化ではまず最小項展開を求める

最小項展開と最大項展開の求め方

- 最小項展開:シャノンの展開定理

  $f(x_1,\dots,x_{n-1},x_n) = x_n\, f(x_1,\dots,x_{n-1},1)+\sim x_n\, f(x_1,\dots,x_{n-1},0)$

  より

$$\mathbf{f(x_1,\dots,x_n) = \Sigma\ \ f(e_1,\dots,e_n)\, x_1^{e1}\dots x_n^{en}}$$

$$\mathbf{= \sum_{f(e_1,\dots,e_n)=1} x_1^{e1}\dots x_n^{en}}$$

- 最大項展開:シャノンの展開定理の双対形より

$$\mathbf{f(x_1,\dots,x_n) = \prod\ (\ f(e_1,\dots,e_n)+ x_1^{\sim e1}+\dots+x_n^{\sim en})}$$

$$\mathbf{= \prod_{f(e_1,\dots,e_n)=0} (x_1^{\sim e1}+\dots+x_n^{\sim en})}$$

論理式の順序

- f(x1,…,xn)≤g(x1,…,xn)

  ⇔∀x1,…,xn.(f(x1,⋯,xn)=1 ⇒ g(x1,⋯,xn)=1)

  「gはfを覆う」「fはgに覆われる」

  例： $xy+x{}^{\sim}yz \leq x$

x \ yz	00	01	11	10
0				
1		1	1	1

≤

x \ yz	00	01	11	10
0				
1	1	1	1	1

積和標準形においてなるべく「大きな」積項を用いれば、和の数が少なくてすむ→ 主項

主項

- 積項pが論理関数fの主項:

  (1)pはfに覆われる(p≤f)

  (2)fに覆われ、かつpよりも真に大きな積項が存在しない

  すなわち、「fに覆われる極大の積項」

例:C(a,b,c) = ab~c+a~bc+~abc+abcの主項

〇ab, 〇bc, 〇ca, ×ab~c, ×a

論理式を簡単化するには、主項の和で表せばよい

例:C(a,b,c) = ab + bc + ca

部分積項、主項

- 積項$x_1^{e1}\ldots x_n^{en}$の部分積項: $x_1^{e1},\ldots,x_n^{en}$の一部からなる積

  例:xy~zの部分積項

$$\underbrace{1,\ x,\ y,\ \sim z,\ xy,\ y\sim z,\ x\sim z}_{\text{真の部分積項}},\ xy\sim z$$

主項

- 積項pが論理関数fの主項：

  (1)pはfに覆われる(p≤f)

  (2)fに覆われ、かつpよりも真に大きな積項が存在しない

  すなわち、「fに覆われる極大の積項」

例：C(a,b,c) = ab~c+a~bc+~abc+abcの主項

〇ab, 〇bc, 〇ca, ×ab~c, ×a

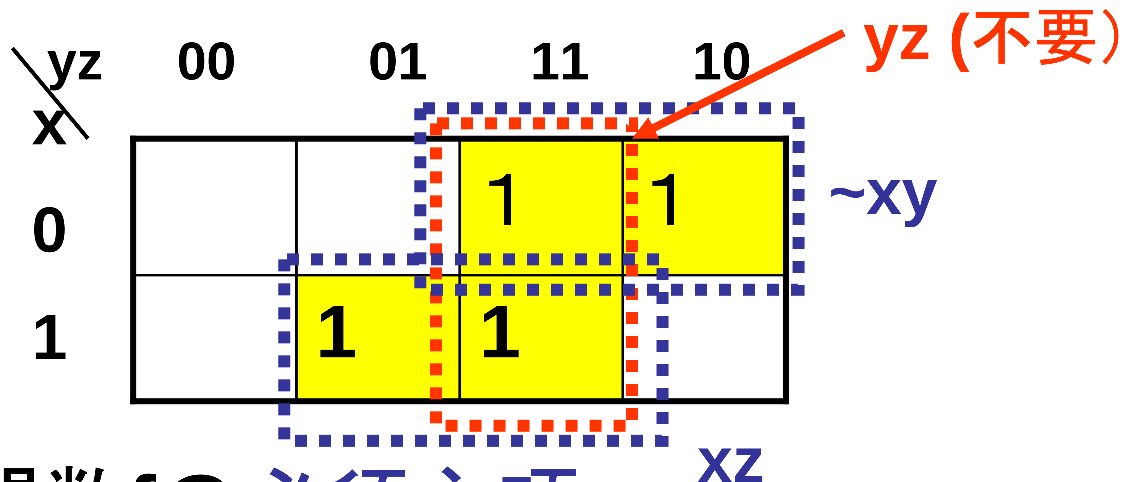
論理式を簡単化するには、主項の和で表せばよい

例：C(a,b,c) = ab + bc + ca

必須主項

- 論理式を積和標準形で表すとき、すべての主項が必要とはかぎらない。

  例：$f(x,y,z) = xz + \sim xy + yz = xz + \sim xy$



- 論理関数fの必須主項
  - fのある最小項を覆う唯一の主項

論理式を簡単化するにはまず必須主項を探し、残った最小項を覆う主項を選べばよい

論理式の簡単化の方針のまとめ

（１）簡単化したい論理関数fのすべての主項を求める

（２）fを覆う最小の主項の部分集合（最小被覆集合）を求める。

　　最小被覆集合を求めるにはまず

　　必須主項を探せばよい

（３）最小被覆集合が複数ある場合には変数の出現回数が最小のものを選ぶ

今日の講義内容

- 論理式の論理回路への変換
- 論理式の簡単化とその意義
- 論理式の簡単化の準備
  - 論理演算の性質
  - 用語の定義（部分積項、主項、論理式の順序関係）
- カルノー図による論理式の簡単化

論理式の簡単化手法

- カルノー図による方法
  - ３～６変数論理関数に適用可能
  - 図を使うので人間にはわかりやすい
  - 多変数の関数、計算機による自動化には向かない
- クワイン-マクラスキ法
- コンセンサス法

次回以降

カルノー図

- 真理値表を2次元の図で表したもの
- 積項が長方形に対応するように配置

真理値表

x	y	z	C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

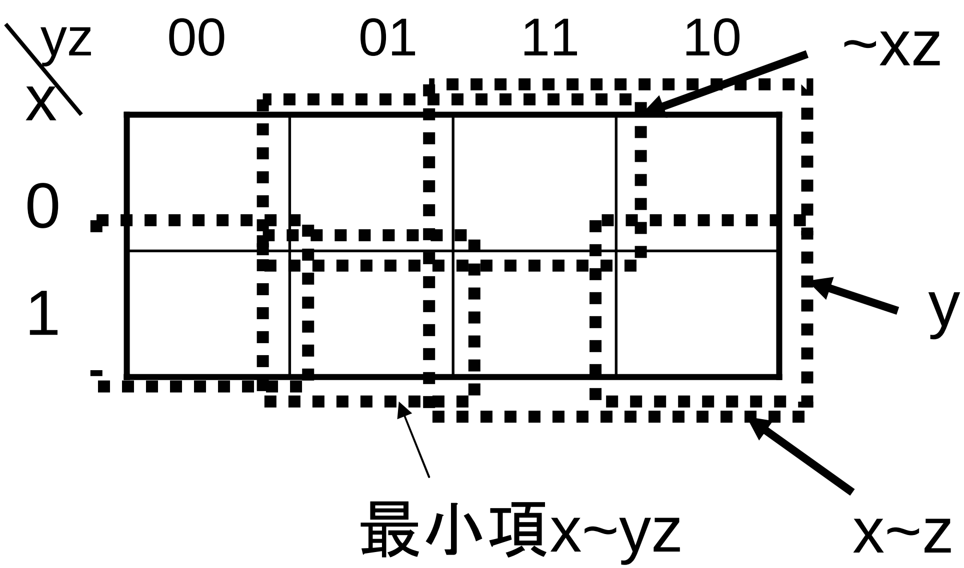
カルノー図による表現

x \ yz	00	01	11	10
0			1	
1		1	1	1

カルノー図

- 真理値表を2次元（または3次元）の図で表したもの
- 積項が長方形に対応するように配置

３変数のカルノー図

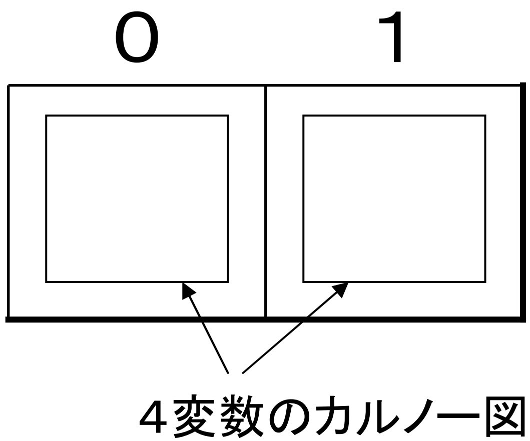


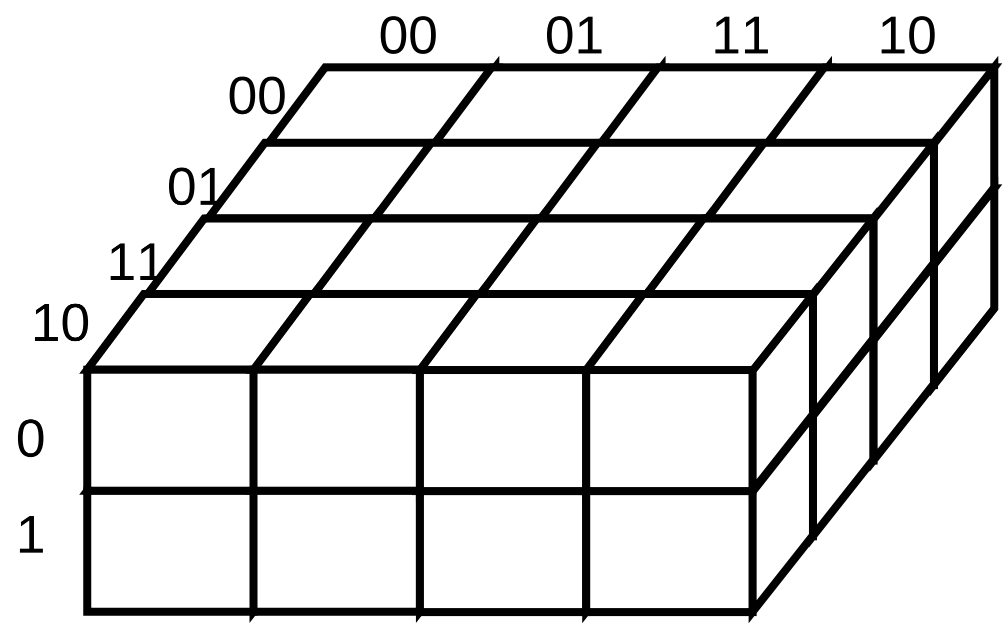
4変数のカルノー図

xy \ zw	00	01	11	10
00				
01				
11				
10				

カルノー図

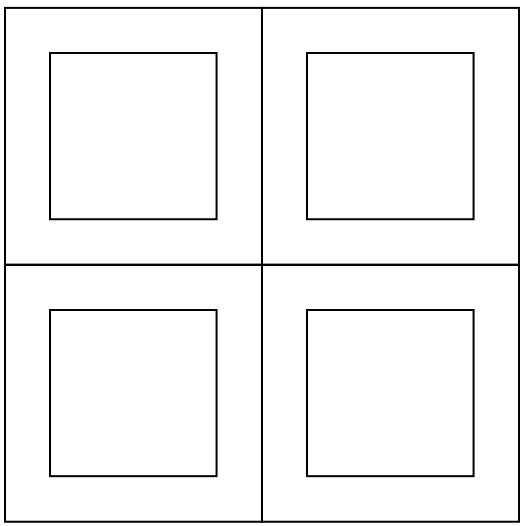
５変数のカルノー図

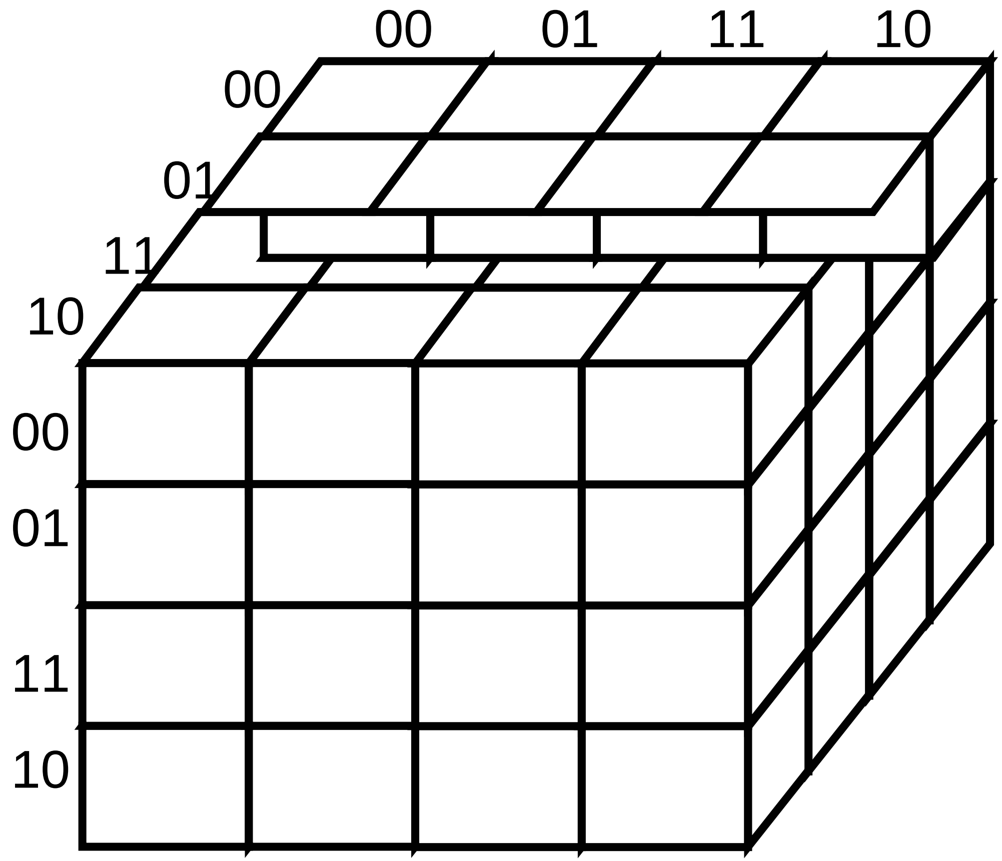




カルノー図

6変数のカルノー図





カルノー図による簡単化

(1)積和標準形または真理値表に基づきカルノー図を作成

x \ yz	00	01	11	10
0			1	
1		1	1	1

（２）積項に対応する長方形ですべて１からなるものを大きなものから順に探す

カルノー図による簡単化

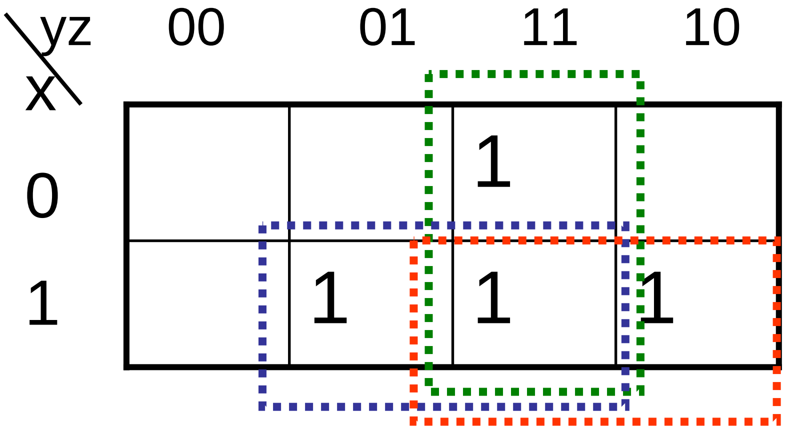
（2）積項に対応する長方形ですべて1からなるものを大きなものから順に探す

右の例:

最大のもの(1)　なし

面積4のもの　　なし

面積2のもの　右の3つ



カルノー図による簡単化

（３）（２）で見つけた長方形のなるべく少ない組み合わせですべてを覆うものを探す

右の例:

右の3つすべて

x \ yz	00	01	11	10
0			1	
1		1	1	1

（４）（３）で選んだ長方形の組み合わせを積和標準形の形で表す（各長方形を表す積項の和）

xy + yz + zx

第2回レポート課題

- 2つの2桁の2進数（x2x1とy2y1)の掛け算の演算結果を$z_4z_3z_2z_1$とする。

（1）z_3を表すカルノー図を作れ

（2）（1）を用いてz3をなるべく簡単な積和標準形で表せ。