

# カルノー図による論理関数の簡単化

## カルノー図による論理関数の簡単化

OR の真理値表		
入力 A	入力 B	出力 f(A, B)
0	0	0
0	1	1
1	0	1
1	1	1

- 上記 OR の真理値表から加法標準形を作ると、

$$\begin{aligned}
 f &= A' \cdot B + A \cdot B' + A \cdot B \\
 &= A \cdot (B+B') + A' \cdot B \\
 &= A + A' \cdot B \\
 &\text{うまく OR の形にならない。}
 \end{aligned}$$

$$\begin{aligned}
 &\text{再度挑戦} \\
 &\text{べき等則} \\
 f &= A' \cdot B + A \cdot B' + A \cdot B \\
 &= A' \cdot B + A \cdot B' + A \cdot B + A \cdot B \\
 &= (A \cdot B + A' \cdot B) + (A \cdot B + A \cdot B') \\
 &\text{分配則で因数分解すると、} \\
 &= (A+A') \cdot B + A \cdot (B+B') \\
 &= B+A
 \end{aligned}$$

- 以上のように、ブール代数による変形は熟練がないと、行き詰まることもある。

そこでカルノー図を用いて式の簡単化をする。

- カルノー図とは真理値表をマスメを用いて表したものである。

2変数のカルノー図	2入力 OR のカルノー図																
<p style="text-align: center;"><b>2入力カルノー図</b></p> <p>出力 f      入力 A                   0    1</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 5px;">入力 B</td> <td style="padding-right: 5px;">0</td> <td style="padding: 5px;">a</td> <td style="padding: 5px;">c</td> </tr> <tr> <td></td> <td style="padding-right: 5px;">1</td> <td style="padding: 5px;">b</td> <td style="padding: 5px;">c</td> </tr> </table> <p>a は入力A=0, B=0に対応する出力 b は入力A=0, B=1に対応する出力 c は入力A=1, B=0に対応する出力 d は入力A=1, B=1に対応する出力</p>	入力 B	0	a	c		1	b	c	<p>出力 f      入力 A                   0    1</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 5px;">入力 B</td> <td style="padding-right: 5px;">0</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> <tr> <td></td> <td style="padding-right: 5px;">1</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> </tr> </table>	入力 B	0	0	1		1	1	1
入力 B	0	a	c														
	1	b	c														
入力 B	0	0	1														
	1	1	1														

- カルノー図の隣接するマスメは1つの入力信号の変化に対応する。

2入力 OR のカルノー図	2入力 OR のカルノー図																
<p>出力 f      入力 A                   0    1</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 5px;">入力 B</td> <td style="padding-right: 5px;">0</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> <tr> <td></td> <td style="padding-right: 5px;">1</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> </tr> </table> <p>矢印方向の変化は、B=1の時に、Aを0から1に変化させることに対応する。 この時、出力は1のままであるので、B=1ならば、Aに関係なく出力は1。 したがって、この2マスをも <math>B \cdot A' + B \cdot A</math> でなく <math>B</math> で表せる。</p>	入力 B	0	0	1		1	1	1	<p>出力 f      入力 A                   0    1</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 5px;">入力 B</td> <td style="padding-right: 5px;">0</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> <tr> <td></td> <td style="padding-right: 5px;">1</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> </tr> </table> <p>同じく、点線で囲まれた2マスも <math>A \cdot B' + A \cdot B</math> ではなく、<math>A</math> とすることができる。 結局、<math>f = A + B</math> となる。 隣接する1をグループ化することで、ブール式を簡単化することができる。</p>	入力 B	0	0	1		1	1	1
入力 B	0	0	1														
	1	1	1														
入力 B	0	0	1														
	1	1	1														

### 3入力のカルノー図

3変数のカルノー図

**3入力カルノー図**

出力 f                      入力 A,B

		00	01	11	10
入力 C	0				
	1				

### 4入力のカルノー図

4変数のカルノー図

**4入力カルノー図**

出力 f                      入力 A,B

		00	01	11	10
入力 C,D	00				
	01				
	11				
	10				

### 例題

$$f(A,B,C,D) = A \cdot B \cdot C' \cdot D' + A' \cdot B' \cdot C' \cdot D + A \cdot B' \cdot C' \cdot D + A' \cdot B' \cdot C \cdot D + A' \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A' \cdot B' \cdot C \cdot D' + A' \cdot B \cdot C \cdot D'$$

を簡単化する。以前やった、ブール式の性質だけでは難しそうだが、カルノー図でやると以下のようにになります。

4変数のカルノー図	隣接する $2^n$ 個 (1,2,4,...) の '1' をグループ化する
<b>4入力カルノー図</b> 出力 f 入力 A,B 00 01 11 10 00 01 入力 C,D 11 10	<b>4入力カルノー図</b> 出力 f 入力 A,B 00 01 11 10 00 01 入力 C,D 11 10

- 四角で囲った部分は AND ゲートとなり、全体を OR することになる。
- オーバーラップしてでも、四角で多くの1を囲む方が良い、1の数が1つでは4入力 AND ゲート、2つでは3入力 AND ゲート、4つでは2入力 AND ゲートとゲートが単純(トランジスタ数が少ない)になる。
- 点線の四角の囲みはなくても、すべての1を囲めているので、不要である。

ということで、  $f(A,B,C,D) = A \cdot B \cdot C' \cdot D' + B' \cdot C' \cdot D + A' \cdot C + B \cdot C \cdot D$

と簡単化される。回路図は以下のようなになる。

簡単化したあとの回路図	NAND,NOR,NOT を使ってトランジスタ数を減らすと、
トランジスタ数は $2 \cdot 4 + 10 + 2 \cdot 8 + 6 + 10 = 50$	トランジスタ数は $2 \cdot 4 + 8 + 2 \cdot 6 + 4 + 8 = 40$

## 禁止項を利用した簡単化

- まず、禁止項とはなにか？  
→ そのような入力の組み合わせが発生せず、出力がどうでもよい。
- 英語では DONT CARE (ドントケア) と言い、「気にしない、どうでも良い」の意味

### 例題

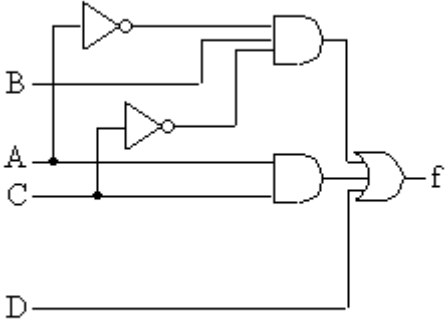
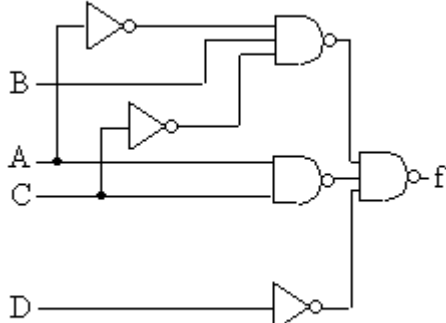
4変数のカルノー図	隣接する $2^n$ 個 (1,2,4,...) の '1' をグループ化する																																																																								
<p>ドントケアを <math>\phi</math> で示す。</p> <p>出力 f</p> <table border="1"> <tr> <td></td> <td></td> <td colspan="4">入力 A,B</td> </tr> <tr> <td></td> <td></td> <td>00</td> <td>01</td> <td>11</td> <td>10</td> </tr> <tr> <td>00</td> <td></td> <td></td> <td>1</td> <td></td> <td></td> </tr> <tr> <td>01</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>11</td> <td><math>\phi</math></td> <td><math>\phi</math></td> <td><math>\phi</math></td> <td><math>\phi</math></td> <td></td> </tr> <tr> <td>10</td> <td></td> <td></td> <td></td> <td>1</td> <td>1</td> </tr> </table> <p>入力 C,D</p>			入力 A,B						00	01	11	10	00			1			01	1	1	1	1	1	11	$\phi$	$\phi$	$\phi$	$\phi$		10				1	1	<p>ドントケアを利用して、すべての1をなるべく大きいな四角で囲む。</p> <p>出力 f</p> <table border="1"> <tr> <td></td> <td></td> <td colspan="4">入力 A,B</td> </tr> <tr> <td></td> <td></td> <td>00</td> <td>01</td> <td>11</td> <td>10</td> </tr> <tr> <td>00</td> <td></td> <td></td> <td>1</td> <td></td> <td></td> </tr> <tr> <td>01</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>11</td> <td><math>\phi</math></td> <td><math>\phi</math></td> <td><math>\phi</math></td> <td><math>\phi</math></td> <td></td> </tr> <tr> <td>10</td> <td></td> <td></td> <td></td> <td>1</td> <td>1</td> </tr> </table> <p>入力 C,D</p>			入力 A,B						00	01	11	10	00			1			01	1	1	1	1	1	11	$\phi$	$\phi$	$\phi$	$\phi$		10				1	1
		入力 A,B																																																																							
		00	01	11	10																																																																				
00			1																																																																						
01	1	1	1	1	1																																																																				
11	$\phi$	$\phi$	$\phi$	$\phi$																																																																					
10				1	1																																																																				
		入力 A,B																																																																							
		00	01	11	10																																																																				
00			1																																																																						
01	1	1	1	1	1																																																																				
11	$\phi$	$\phi$	$\phi$	$\phi$																																																																					
10				1	1																																																																				

- ドントケア ( $\phi$  の部分) はそのような入力はされなくて、そのときの出力はいつでもよいので、必要であればそれを1と考えることで、四角で囲まれたグループを大きくする。
- 説明したように、囲みが大きいほど、対応する AND ゲートの入力数は減り、トランジスタ数が減る。
- 一般的に、トランジスタ数が少ない方が、LSI 上の面積が小さく、かつ動作スピードも速い。
- もちろん、すべてのドントケアを囲む必要はない。

ということで、 $f(A,B,C,D) = A' \cdot B \cdot C' + D + A \cdot C$

と簡単化される。

回路図は以下のようになる。

<p>単純化したあとの回路図</p>	<p>NAND,NOR,NOT を使って トランジスタ数を減らすと、</p>
	
<p>トランジスタ数は <math>2*2 + 8 + 6 + 8 = 26</math></p>	<p>トランジスタ数は <math>3*2 + 6 + 4 + 6 = 22</math></p>

宿題4 学籍番号 名前 日付 を書いて 提出すること。

(注意:回路を設計する場合、特に明記しないが、なるべく少ないゲート数で実現せよ。)

- 1)  $A \cdot (A' + B) = A \cdot B$  をブール代数の性質をもちいて証明せよ。
- 2)  $(A \cdot B + A' \cdot B) = A \cdot B' + A' \cdot B$  をブール代数の性質をもちいて証明せよ。
- 3) 以下の真理値表をカルノー図で表し、隣接する1があれば○で囲み、その結果より出力関数fのブール式を求めよ。

<p>??の真理値表</p>		
<p>入力 A</p>	<p>入力 B</p>	<p>出力f(A, B)</p>
<p>0</p>	<p>0</p>	<p>0</p>
<p>0</p>	<p>1</p>	<p>1</p>
<p>1</p>	<p>0</p>	<p>0</p>
<p>1</p>	<p>1</p>	<p>1</p>

4) 以下をカルノー図を用いてブール式を簡単化せよ。

$$f = A' \cdot B' \cdot C' \cdot D' + A' \cdot B \cdot C' \cdot D + A \cdot B \cdot C \cdot D + A \cdot B' \cdot C' \cdot D \\ + A' \cdot B' \cdot C \cdot D' + A \cdot B \cdot C' \cdot D + A' \cdot B \cdot C \cdot D + A \cdot B' \cdot C \cdot D$$

5) 前問で  $A' \cdot D$  が禁止項の場合、 $f$  はどのように簡単化されるか示せ。

6) 上記2)の論理関数 $f$ を回路図を実現せよ。(なるべく少ないトランジスタ数で実現せよ。)

また、そのときのトランジスタ数はいくつか？

7) 二つの2ビットの2進数  $A$ 、 $B$  があるとき、

(i)  $A > B$  のとき、出力が1となる論理関数 $f$ の真理値表を描け

(ii) (i)で得られた真理値表より加法標準形で論理関数を表せ

(iii) (ii)の論理関数について、カルノー図を描き簡単化を行え。

(iv) 簡単化された論理関数を実現せよ。

この時、2進数  $A$  の MSB を  $A_1$ 、LSB を  $A_0$ 、 $B$  の MSB を  $B_1$ 、LSB を  $B_0$  とせよ。

8) 上記 4)の論理関数の回路図をなるべく少ないトランジスタ数で実現せよ。

また、そのときのトランジスタ数はいくつか？

9) 上記 7)で  $A=B$  の時だけ、出力が1となる回路を同様の手順で設計せよ。

この時、2進数  $A$  の MSB を  $A_1$ 、LSB を  $A_0$ 、 $B$  の MSB を  $B_1$ 、LSB を  $B_0$  とせよ。

以上