ハードウエア実験
第１回－２
論理回路の基礎と設計

・電子回路素子

・各種ゲート回路

・ブール代数

・カルノー図

・組合せ論理回路の設計手順

・組合せ論理回路の例

・組合せ論理回路設計

1. 目的

・簡単な電子回路素子の知識を身に付ける
・論理回路の基礎を学ぶ
・組合せ論理回路を設計する

2. 論理式

・ブール代数を用いた論理式の簡単化、カルノー図による簡単化

3. 組合せ論理回路の設計

・組合せ論理回路の設計手順について学ぶ
・真理値表から論理回路を導く手法を学ぶ
・組合せ論理回路の実例について学ぶ

電子回路素子の基礎

1.電子回路素子の説明

R (抵抗)　　カーボン皮膜、ソリッド、金属皮膜、酸化金属皮膜

C (キャパシタ、コンデンサ)　　セラミック、フィルム、電解、

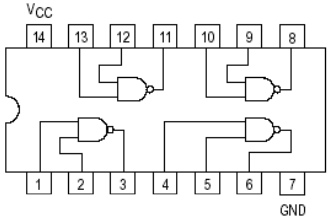
L (インダクタ、コイル)

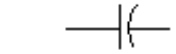
Di (ダイオード)

Tr (トランジスタ)

IC (集積回路)

IC (集積回路)





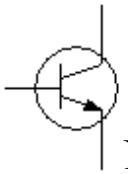
C (コンデンサ)

R (抵抗)

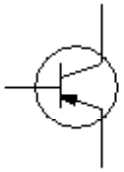
L (インダクタ)

Di (ダイオード)

Biporer Tr
(トランジスタ)

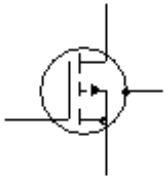


NPN型

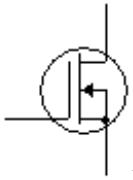


PNP型

MOS Tr
(トランジスタ)



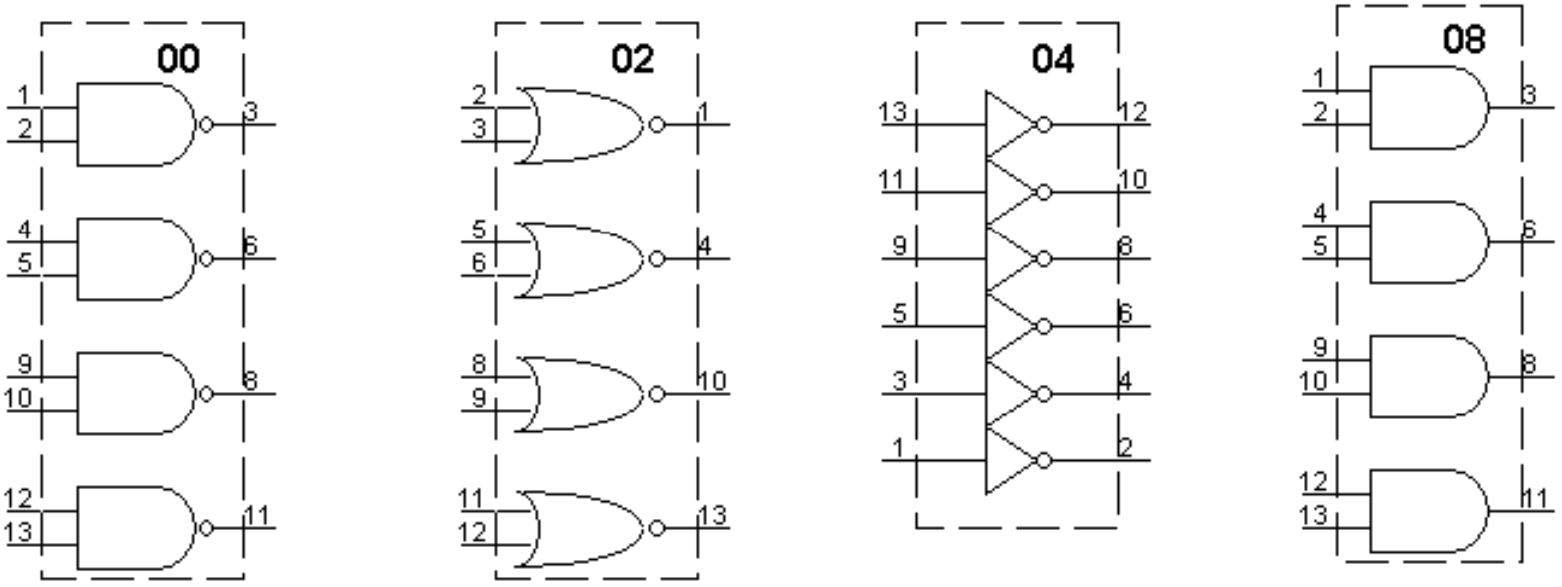
エンハンスP型



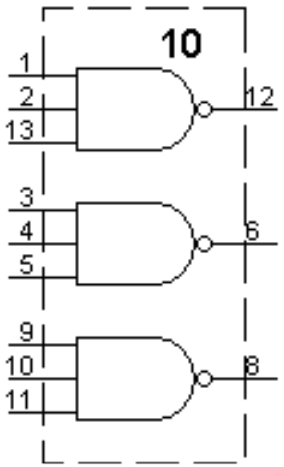
DIP　N型

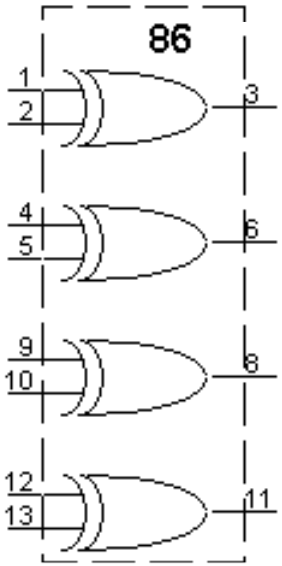
グラウンド、アース
通常電源グラウンド
は回路図上、
線で結ばない

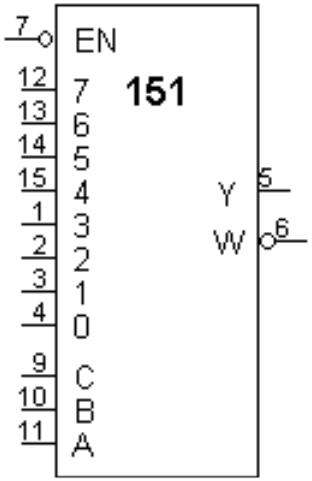
集積回路（組合せ回路系、74XXシリーズ）1/3

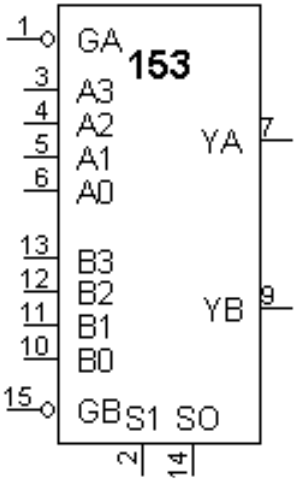


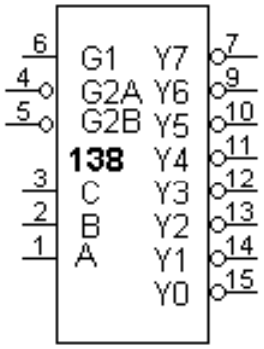
集積回路（組合せ回路系、74XXシリーズ）2/3

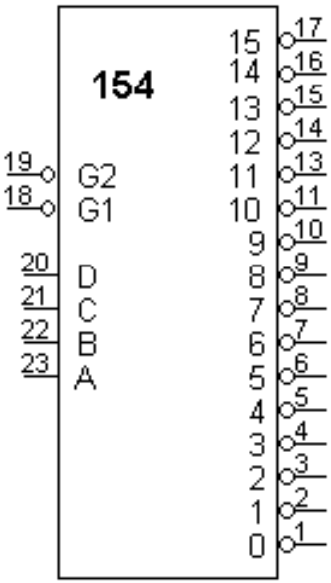


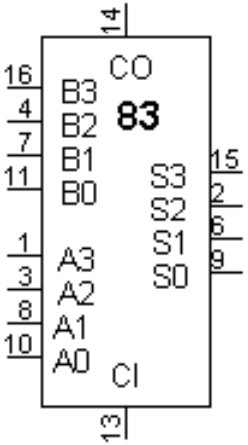


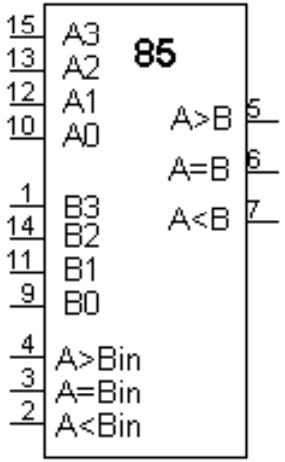




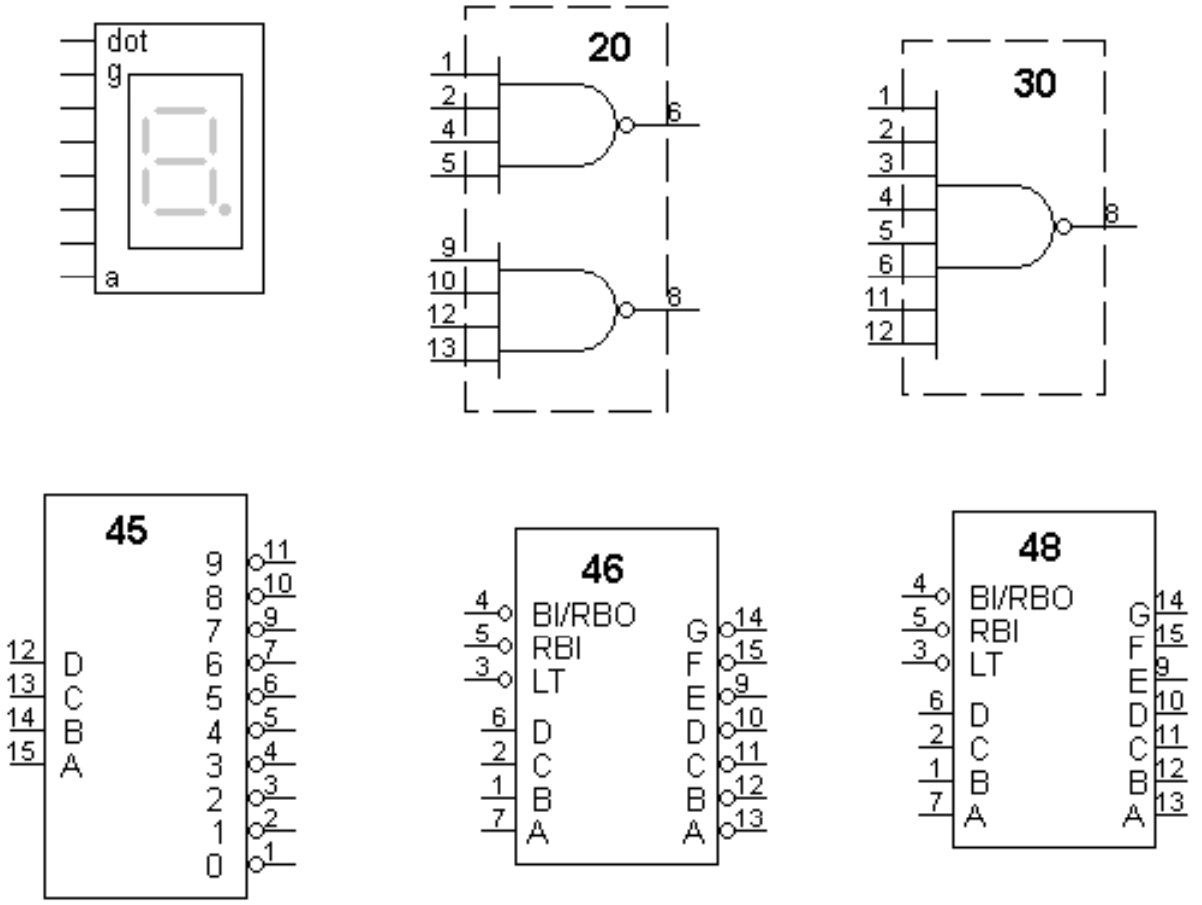




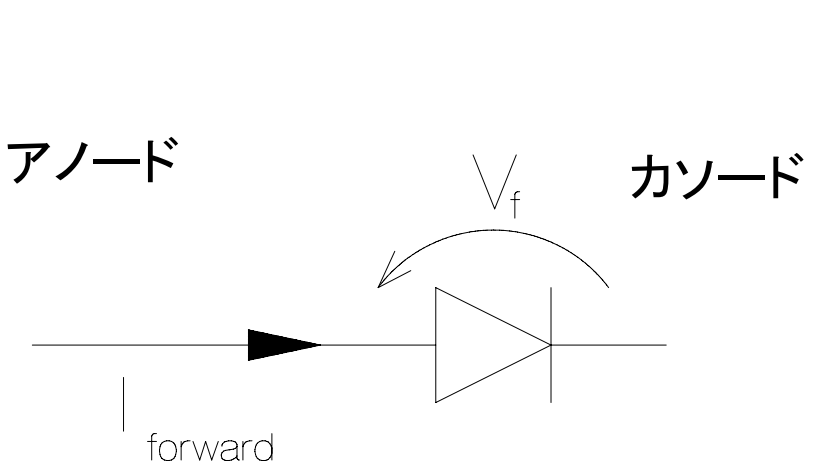


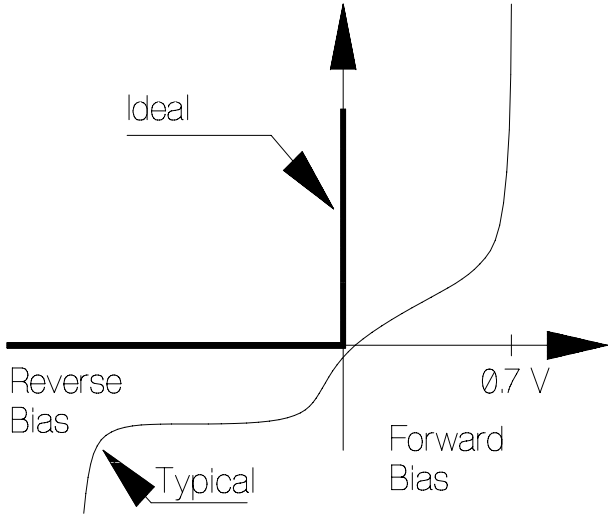


集積回路（組合せ回路系、74XXシリーズ）3/3

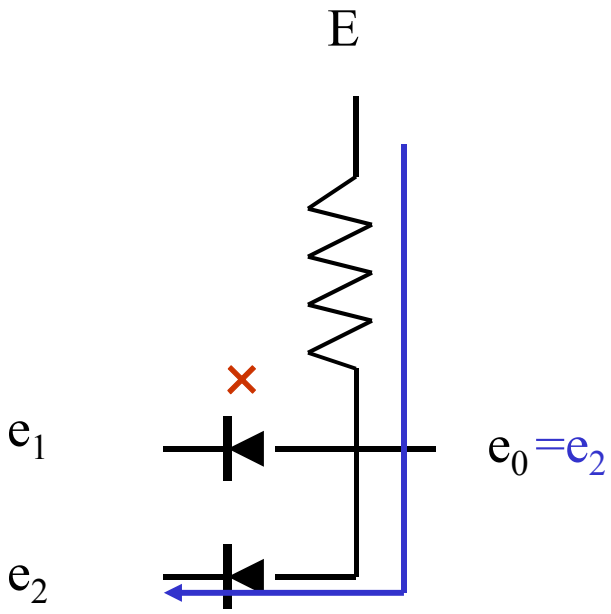


ダイオード

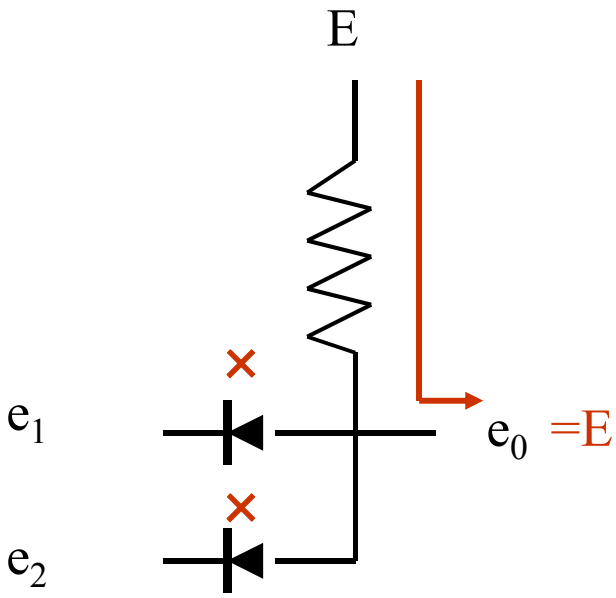

アノード
カソード
V_f
forward


Ideal
Reverse
Bias
0.7 V
Forward
Bias
Typical

A)　$e_2<e_1$,Eの時



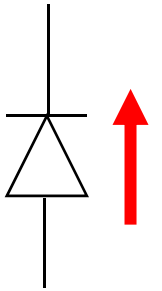
B)　E≲e_2, e_1の時



× : **OFF**

e_1かe_2の電圧が低い方に出力が従う⇒AND回路

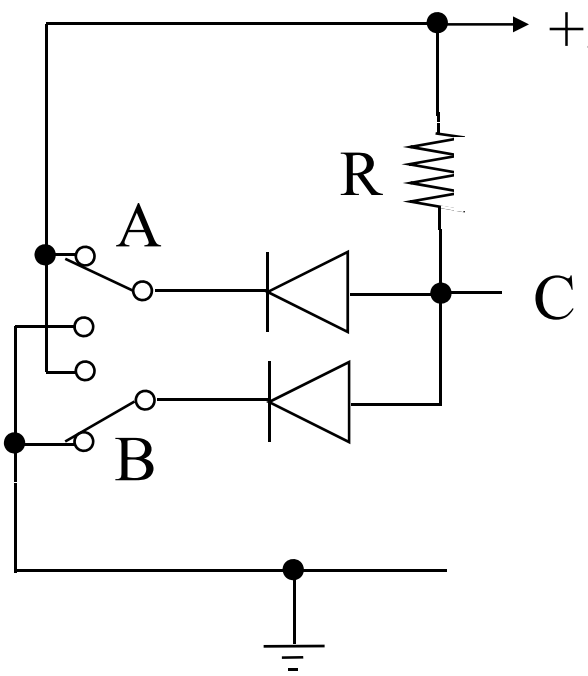
論理回路の構成－正論理と負論理－



ダイオード（スイッチング素子ともよばれる）
・一方向にしか電流が流れない

電流の流れる方向

A	B	C
0V	0V	0V
0V	5V	0V
5V	0V	0V
5V	5V	5V



A、Bどちらかが0Vであれば、Cも0Vとなる

次に、0Vと5Vとを数値の0，1に対応付ける

0Vと5Vとの数値0，1への対応付け

正論理

0V → 0

5V → 1

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

A	B	C
0V	0V	0V
0V	5V	0V
5V	0V	0V
5V	5V	5V

負論理

0V → 1

5V → 0

A	B	C
1	1	1
1	0	1
0	1	1
0	0	0

Bool 代数によればAND

$C = A \cdot B$

通常は正論理を使う

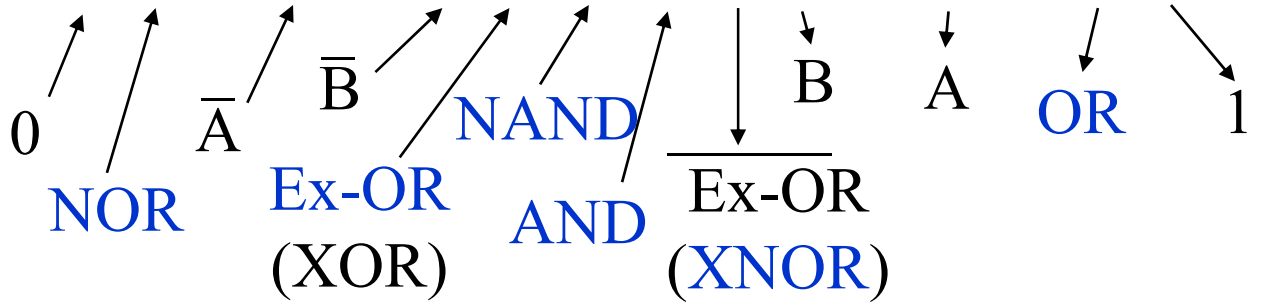
Bool 代数によればOR

$C = A + B$

正論理

２入力回路においては、入力の組合せが 2^2 の4通りある。
これらに対する出力は、4^2 の16通りが考えられる。

A	B	C															
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1



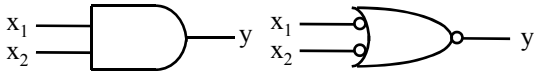
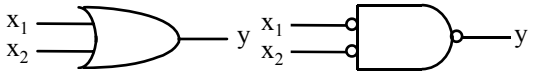
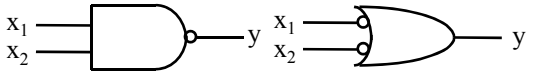
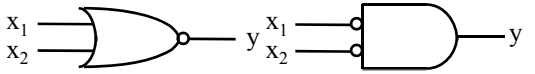
このうち、いくつかは意味を持たないが、多くは有用な論理を与える。

論理回路の基礎

基本論理ゲート

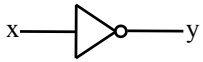
主なゲートを上げる

Buffer, NOT, AND, OR, NAND, NOR, Ex-OR(XOR), Ex-NOR(XNOR)

Buffer		バッファ
NOT		否定
AND		論理積
OR		論理和
NAND		論理積否定
NOR		論理和否定
Ex-OR(XOR)		排他的論理和
Ex-NOR(XNOR)		排他的論理和否定

NOT　：変数が唯一の関数

（変数 x が1なら関数 y は0、x が0なら y は1）



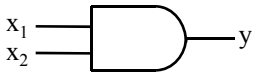
$y=\overline{x}$

x	y
0	1
1	0

AND　：変数の中の最小の値が関数の値となる演算

（変数の中に0が一つでもあれば関数は0、入力が全て1の時のみ出力は1）

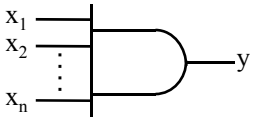
2入力AND演算



$y=x_1x_2$

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

多入力AND演算



$y=x_1x_2\cdots x_{n-1}x_n$

x_1	x_2	·····	x_{n-1}	x_n	y
0	0	·····	0	0	0
0	0	·····	0	1	0
		⋮			
1	1	·····	1	0	0
1	1	·····	1	1	1

OR　：変数の中の最大の値が関数の値となる演算

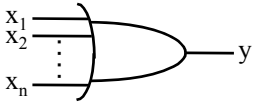
（変数の中に1が一つでもあれば関数は1、入力が全て0の時のみ出力は0）

2入力OR演算

$y=x_1+x_2$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

多入力OR演算



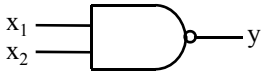
$y=x_1+x_2+\cdots+x_{n-1}+x_n$

x_1	x_2	·····	x_{n-1}	x_n	y
0	0	·····	0	0	0
0	0	·····	0	1	1
		⋮			
1	1	·····	1	0	1
1	1	·····	1	1	1

これらの3つの演算の組み合わせであらゆる論理関数が表現できる。→完全系

NAND　：変数の中の最小の値を否定した演算(ANDの否定)
（変数の中に0が一つでもあれば関数は1、入力が全て1の時のみ出力は0）

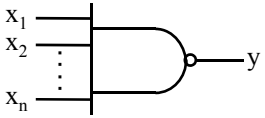
2入力NAND演算



$y=\overline{x_1 x_2}$

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

多入力NAND演算



$y=\overline{x_1 x_2 \cdots x_{n-1} x_n}$

x_1	x_2	·····	x_{n-1}	x_n	y
0	0	·····	0	0	1
0	0	·····	0	1	1
		⋮			
1	1	·····	1	0	1
1	1	·····	1	1	0

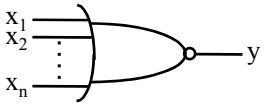
NOR　：変数の中の最大の値を否定した演算(ORの否定)
（変数の中に1が一つでもあれば関数は0、入力が全て0の時のみ出力は1）

2入力NOR演算

$y=\overline{x_1 + x_2}$

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

多入力NOR演算



$y=\overline{x_1 + x_2 + \cdots + x_{n-1} + x_n}$

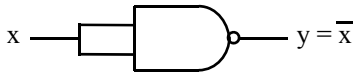
x_1	x_2	·····	x_{n-1}	x_n	y
0	0	·····	0	0	1
0	0	·····	0	1	0
		⋮			
1	1	·····	1	0	0
1	1	·····	1	1	0

NANDはこれ一つのみで完全系を形成している。→あらゆるGATEが実現できる。
NORも同様に、これ一つのみで完全系を形成している。

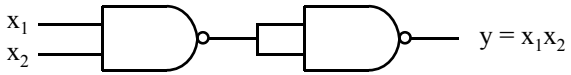
NANDはこれのみで完全系を形成している。→あらゆるGATEが実現できる。

NANDを繰り返し使うと、これのみで希望の論理回路が実現できる。

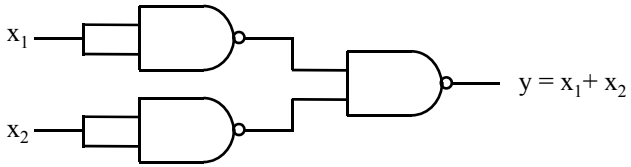
NOT演算



AND演算



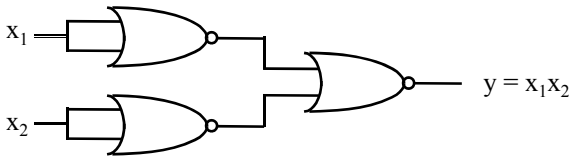
OR演算



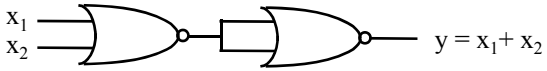
NORもこれのみで完全系を形成している。→あらゆるGATEが実現できる。

NOT演算

AND演算



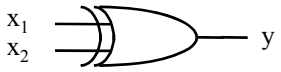
OR演算



Ex-OR(Exclusive OR, XOR)

：全ての入力が゜゛“1”の時、および全ての入力が“0”の時のみ出力は“0”で、
その他の場合は“1”を出力する。

2入力Ex-OR演算



$y=x_1 \oplus x_2$

$y=x_1 \cdot \overline{x_2} + \overline{x_1} \cdot x_2$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Ex-NOR(Exclusive NOR, XNOR)

：全ての入力が“1”の時、および全ての入力が“0”の時のみ出力は“1”で、
その他の場合は“0”を出力する。

2入力Ex-NOR演算

$y=x_1 \odot x_2$

$y=x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2}$

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	1

EXORは完全系でないが、加算回路、パリティ回路で有用

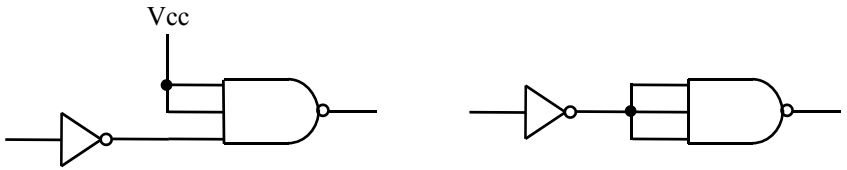
ファンアウト(fan out)、ファンイン(fan in)

ファンアウト

論理回路ICを使って論理を構成するとき、1個の論理素子の出力が駆動できる論理素子の数をファンアウト(fan out)という。論理素子に接続された論理素子の各入力には電流が流れ、この電流は1個の論理素子から供給される。接続された論理素子の数が多くなればなるほど、出力の電流も増加する。しかし素子の出力から供給できる電流には限界があるため、それ以上電流を流すと出力電圧が下がってしまい、論理素子の動作下限値より低くなってしまう。このようにならないよう、論理素子にはファンアウトが定められている。

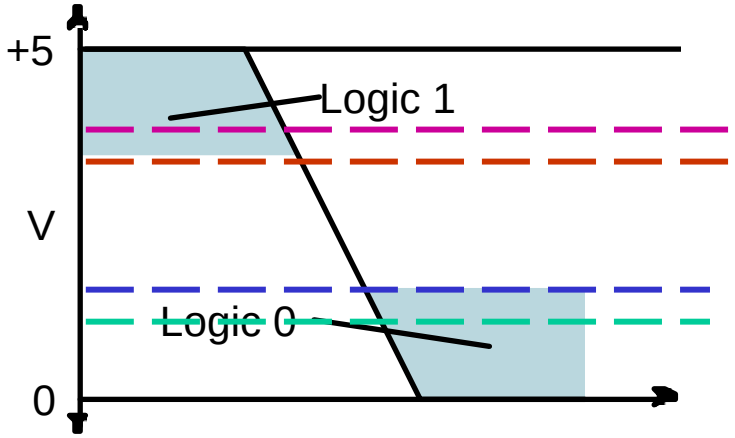
ファンイン

多入力の論理回路ICを一つだけ入力として使うとき、入力用の1つを除き他をまとめて電源(あるいはグランド)に接続するか、あるいは全入力端をまとめて一つの入力として接続する場合がある。後者の場合、負荷は入力端子の数だけ増えることになる。この入力の等価的な負荷をファンインという。



ファンイン＝1　　ファンイン＝3

入出力電圧と雑音余裕、ファンアウト



V_{OH}：出力側でHと出力する最低電圧
V_{IH}：入力側でHと判定する最低電圧

V_{IL},：入力側でLと判定する最高電圧
V_{OL},：出力側でLを出力する最高電圧

$$V_{OL} < V_{IL} < V_{IH} < V_{OH}$$ の関係が必要

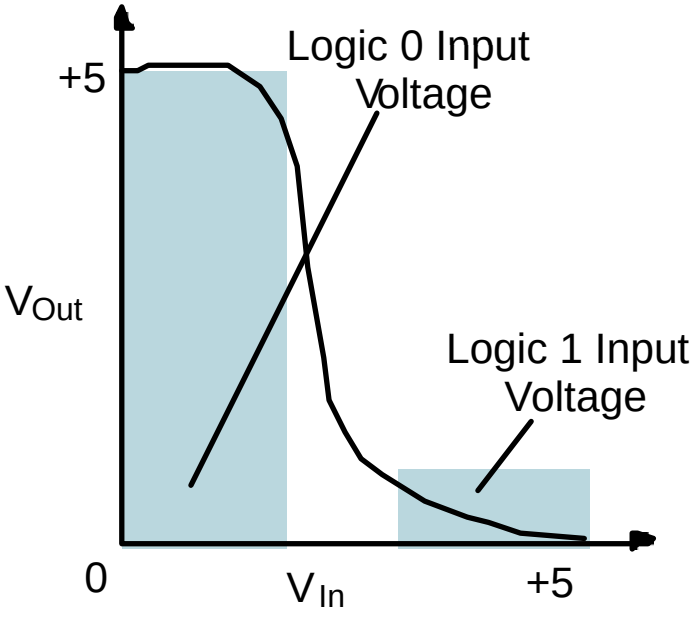
実際には供給電圧の変動、負荷の変動により出力電圧が変化する

余裕を持たせるためTTLでは雑音余裕として

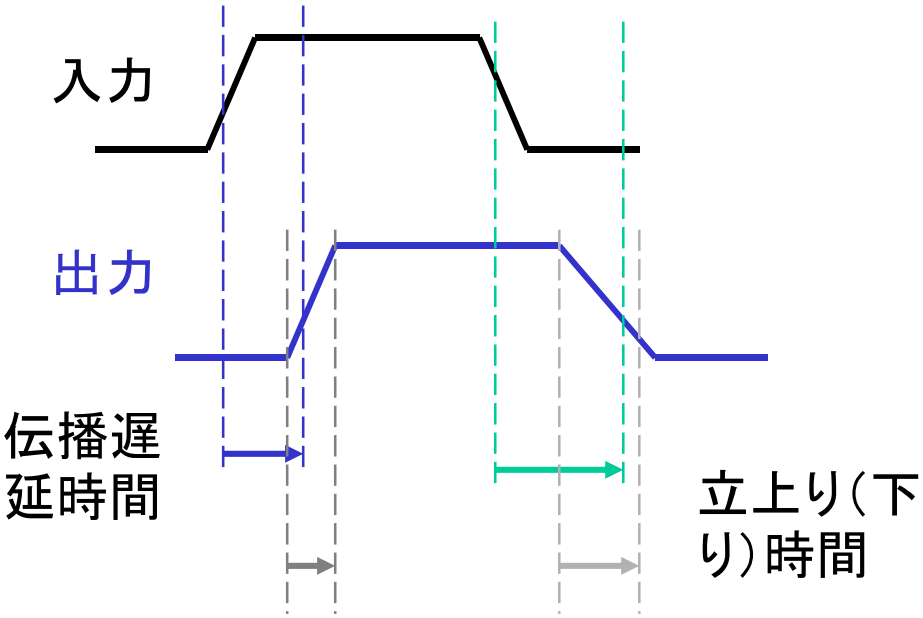
$$V_{OH} - V_{IH} = 0.4[V], V_{IL} - V_{OL} = 0.4[V]$$

出力段に後続の入力数を増やすと駆動するためレベルが低下(上昇)し余裕が無くなってくる。⇒最大入力数をファンアウト数と言う。TTLでは8程度が普通。CMOSでは電流が少ないので数はあまり問題にならない。

NOT回路の入出力特性



論理回路ICのスイッチング特性



一般的に立上り/下がり時間は異なる。

また、スイッチング時にオーバーシュート/アンダーシュートなどの過渡応答があり、誤動作することも。きちんとした配線、終端、整合が必要。

ブール代数(Boolean algebra)

基本公式

べき等律 (Idempotency low)
(同一則)

$$\begin{cases} x \cdot x = x \\ x + x = x \end{cases}$$

交換律 (Commutative low)
(交換則)

$$\begin{cases} x \cdot y = y \cdot x \\ x + y = y + x \end{cases}$$

結合律 (Associative low)
(結合則)

$$\begin{cases} x \cdot (y \cdot z) = (x \cdot y) \cdot z \\ x + (y + z) = (x + y) + z \end{cases}$$

分配律 (Distributive low)
(分配則)

$$\begin{cases} (x + y) \cdot z = (x \cdot z) + (y \cdot z) \\ (x \cdot y) + z = (x + z) \cdot (y + z) \end{cases} \qquad \left[x + y \cdot \overline{y} = (x + y) \cdot (x + \overline{y}) \right.$$

吸収律 (Absorption low)
(吸収則)

$$\begin{cases} x + 1 = 1 \\ x \cdot 0 = 0 \end{cases} \quad \begin{cases} x + 0 = x \\ x \cdot 1 = x \end{cases} \quad \begin{cases} x + \overline{x} \cdot y = x + y \\ x \cdot (\overline{x} + y) = x \cdot y \end{cases}$$

否定律 (Complement low)
(補元則)

$$\begin{cases} x + \overline{x} = 1 \\ x \cdot \overline{x} = 0 \end{cases}$$

双対の理 (Principle of duality)

$$+ \rightleftarrows \cdot \qquad 0 \rightleftarrows 1$$

ド･モルガンの定理 (De Morgan's Law)

1変数 $\quad \overline{\overline{x}} = \overline{(\overline{x})} = x$

2変数 $\quad \begin{cases} \overline{x \cdot y} = \overline{x} + \overline{y} \\ \overline{x + y} = \overline{x} \cdot \overline{y} \end{cases}$

n変数 $\quad \begin{cases} \overline{x_1 + x_2 + \cdots + x_n} = \overline{x_1} \cdot \overline{x_2} \cdot \cdots \cdot \overline{x_n} \\ \overline{x_1 \cdot x_2 \cdot \cdots \cdot x_n} = \overline{x_1} + \overline{x_2} + \cdots + \overline{x_n} \end{cases}$

関数系 $\quad \overline{f(x_1, x_2, \cdots, x_n, 1, 0, +, \cdot)} = f(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n}, 0, 1, \cdot, +)$

最小項と最大項

最小項：全ての変数を含み、論理積で表されている項を最小項(minterm)という。

最大項：全ての変数を含み、論理和で表されている項を最大項(maxterm)という。

	最小項		最大項	
x y z	項	記号	項	記号
0 0 0	$\bar{x}\,\bar{y}\,\bar{z}$	m_0	$x+y+z$	M_0
0 0 1	$\bar{x}\,\bar{y}\,z$	m_1	$x+y+\bar{z}$	M_1
0 1 0	$\bar{x}\,y\,\bar{z}$	m_2	$x+\bar{y}+z$	M_2
0 1 1	$\bar{x}\,y\,z$	m_3	$x+\bar{y}+\bar{z}$	M_3
1 0 0	$x\,\bar{y}\,\bar{z}$	m_4	$\bar{x}+y+z$	M_4
1 0 1	$x\,\bar{y}\,z$	m_5	$\bar{x}+y+\bar{z}$	M_5
1 1 0	$x\,y\,\bar{z}$	m_6	$\bar{x}+\bar{y}+z$	M_6
1 1 1	$x\,y\,z$	m_7	$\bar{x}+\bar{y}+\bar{z}$	M_7

x y z	f
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	1
1 1 1	1

最小項と最大項を用いた論理式の導出

右上の真理値表から、fが1の項に着目すると、次式のように表される。

$$f(x,y,z)=\bar{x}\,\bar{y}\,z+x\,\bar{y}\,\bar{z}+x\,y\,\bar{z}+x\,y\,z$$
$$=m_1+m_4+m_6+m_7$$
$$=\sum(1,4,6,7)$$

これを加法標準形(disjunctive canonical form)といい、最小項の論理和で表され、次式のように表される。

$$f=\sum(\text{最小項})$$

また、f が0の項に着目すると、次式のように表される。

$$\overline{f(x,y,z)}=\bar{x}\,\bar{y}\,\bar{z}+\bar{x}\,y\,\bar{z}+\bar{x}\,y\,z+x\,\bar{y}\,z$$

ここで、$f(x,y,z)=\overline{\overline{f(x,y,z)}}$であるので、De Morganの定理を用いて、

$$\begin{aligned}f(x,y,z)&=\overline{\overline{f(x,y,z)}}\\&=\overline{\bar{x}\,\bar{y}\,\bar{z}+\bar{x}\,y\,\bar{z}+\bar{x}\,y\,z+x\,\bar{y}\,z}\\&=(x+y+z)\,(x+\bar{y}+z)\,(x+\bar{y}+\bar{z})\,(\bar{x}+y+\bar{z})\\&=M_0\cdot M_2\cdot M_3\cdot M_5\\&=\Pi(0,2,3,5)\end{aligned}$$

これを乗法標準形(conjunctive canonical form)といい、最大項の論理積で表され、次式のように表される。

$$f=\Pi(\text{最大項})$$

加法標準形と乗法標準形は、一方が決定すると他方は容易に決定できる。

（例） $f(x,y,z)=\sum(0,1,3,4,6)$

とすると、3変数の場合の0,1,2,・・・,7のうち、( )の中の欠けた数から

$$f(x,y,z)=\Pi(2,5,7)$$

が得られる。したがって、

$$\begin{aligned}f(x,y,z)&=\bar{x}\,\bar{y}\,\bar{z}+\bar{x}\,\bar{y}\,z+\bar{x}\,y\,z+x\,\bar{y}\,\bar{z}+x\,y\,\bar{z}\\&=(x+\bar{y}+z)\,(\bar{x}+y+\bar{z})\,(\bar{x}+\bar{y}+\bar{z})\end{aligned}$$

が直ちに得られる。

加法標準形の作り方

（例） $f(x,y,z)=\bar{x}y+x\bar{y}z$

が与えられたとすると、最小項でない項において、含まない変数を$(x+\bar{x}=1)$の形で乗じて、

$$\begin{aligned}f(x,y,z)&=\bar{x}y(z+\bar{z})+x\bar{y}z\\&=\bar{x}yz+\bar{x}y\bar{z}+x\bar{y}z\end{aligned}$$

論理関数の簡単化

論理関数を簡単化することにより、実際の回路素子数が少なくなり、その結果、(1)信頼性が向上する、(2)小型化される、(3)経済的である、(4)保守性がよい、などの効果が期待される。

カルノー図(Karnaugh map)による簡単化

2変数の場合

2変数の場合、各最小項を下図のように配置する。

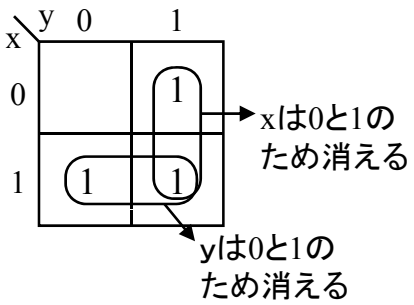
m_0	m_1
m_2	m_3

0	1
2	3

x \ y	0	1
0	$\bar{x}\bar{y}$	$\bar{x}y$
1	$x\bar{y}$	xy

（例） $f(x,y)=\bar{x}y+x\bar{y}+xy$

が与えられたとすると、カルノー図は下図のようになる。よって、$f=x+y$となる。



3変数の場合

3変数の場合、各最小項を下図のように配置する。

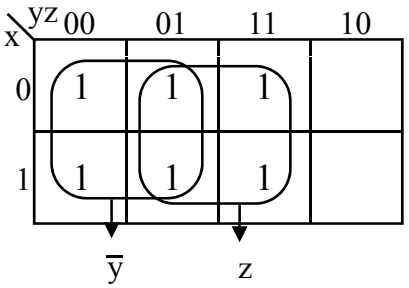
m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

0	1	3	2
4	5	7	6

x \ yz	00	01	11	10
0	$\bar{x}\bar{y}\bar{z}$	$\bar{x}\bar{y}z$	$\bar{x}yz$	$\bar{x}y\bar{z}$
1	$x\bar{y}\bar{z}$	$x\bar{y}z$	xyz	$xy\bar{z}$

（例） $f(x,y,z) = \sum(0,1,3,4,5,7)$ が与えられたとすると、カルノー図は下図のようになる。

よって、$f=\bar{y}+z$となる。



4変数の場合

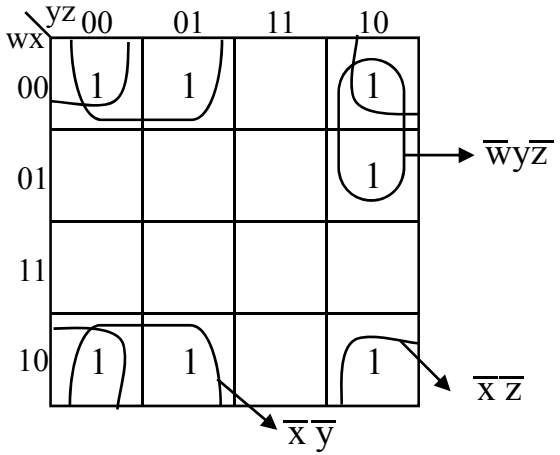
4変数の場合、各最小項を下図のように配置する。

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

wx \ yz	00	01	11	10
00	$\overline{w}\overline{x}yz$	$\overline{w}\overline{x}y\overline{z}$	$\overline{w}\overline{x}yz$	$\overline{w}\overline{x}y\overline{z}$
01	$\overline{w}x\overline{y}z$	$\overline{w}x\overline{y}\overline{z}$	$\overline{w}xyz$	$\overline{w}xy\overline{z}$
11	$wx\overline{y}z$	$wx\overline{y}\overline{z}$	$wxyz$	$wxy\overline{z}$
10	$w\overline{x}\overline{y}z$	$w\overline{x}y\overline{z}$	$w\overline{x}yz$	$w\overline{x}y\overline{z}$

（例） $f(w,x,y,z) = \sum(0,1,2,6,8,9,10)$ のとき、カルノー図は下図のようになる。



よって、$f = \overline{x}\,\overline{y} + \overline{x}\,\overline{z} + \overline{w}y\overline{z}$ となる。

組合せ論理回路

組み合わせ論理回路設計手順

手順1　設計システムを与える

手順2　そのシステムの真理値表を作成する

手順3　その真理値表から論理式を導く

手順4　論理式を簡略化する

手順5　AND, OR, NOTゲートを用いて論理回路を作成する

手順6　NAND, またはNORゲートに統一するかを決定する

手順7　最終的な論理回路のタイムチャートを作成する

ここでは、上記の手順1および手順2の真理値表(右表)
が与えられたものとする。

x_1	x_2	x_3	y
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

真理値表

真理値表(Truth table)から論理回路(Logic diagram)への変換

手順3：真理値表から論理式を導く

3個の変数と1個の関数の論理回路を例にとる。真理値表から論理回路への変換

$$y = f(x_1, x_2, x_3)$$

関数$f(x_1,x_2,x_3)$が右の真理値表のように与えられたとき、論理回路に変換する。

x_1	x_2	x_3	y
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

真理値表

○加法標準形論理式導出（ANDをとりORで結んだ形）

x_1,x_2,x_3それぞれについて、“1”と“0”の部分に分けると、8個の項の次式となる。

$$y = x_1x_2x_3\, f(1,1,1) + x_1x_2\overline{x_3}\, f(1,1,0) + x_1\overline{x_2}x_3\, f(1,0,1) + x_1\overline{x_2}\,\overline{x_3}\, f(1,0,0)$$
$$\overline{x_1}x_2x_3\, f(0,1,1) + \overline{x_1}x_2\overline{x_3}\, f(0,1,0) + \overline{x_1}\,\overline{x_2}x_3\, f(0,0,1) + \overline{x_1}\,\overline{x_2}\,\overline{x_3}\, f(0,0,0)$$

関数$f(x_1,x_2,x_3)$が右の真理値表のように与えられたとすると、$f(1,1,1)$から$f(0,0,0)$までの値を上式に代入し、yを求めると次式となる。

$$y = x_1x_2x_3 + x_1x_2\overline{x_3} + \overline{x_1}x_2x_3 + \overline{x_1}x_2\overline{x_3} + \overline{x_1}\,\overline{x_2}x_3$$

○乗法標準形論理式導出（ORをとりANDで結んだ形、yが0の時のx_1,x_2,x_3 のNOTをとる）

$$y = (\overline{x_1} + x_2 + \overline{x_3}) \cdot (\overline{x_1} + x_2 + x_3) \cdot (x_1 + x_2 + x_3)$$

一目で乗法標準形が有利と判断できる場合(例えば一項のみなど)を除いて、一般に加法標準形が用いられる。したがってここでも加法標準形を用い、下の論理式が得られたとする。

$$y = x_1x_2x_3 + x_1x_2\overline{x_3} + \overline{x_1}x_2x_3 + \overline{x_1}x_2\overline{x_3} + \overline{x_1}\,\overline{x_2}x_3$$

手順4：論理式を簡略化する

上記真理値表から求められたyは次式となる。

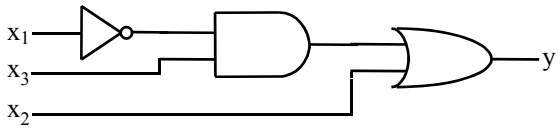
$$y = x_1x_2x_3 + x_1x_2\overline{x_3} + \overline{x_1}x_2x_3 + \overline{x_1}x_2\overline{x_3} + \overline{x_1}\,\overline{x_2}x_3$$

ブール代数(Boolean algebra)を用いて上式を簡略化する。

$$\begin{aligned} y &= x_1x_2x_3 + x_1x_2\overline{x_3} + \overline{x_1}x_2x_3 + \overline{x_1}x_2\overline{x_3} + \overline{x_1}\,\overline{x_2}x_3 \\ &= x_1x_2(x_3+\overline{x_3}) + \overline{x_1}x_2(x_3+\overline{x_3}) + \overline{x_1}x_3(x_2+\overline{x_2}) \\ &= x_1x_2 + \overline{x_1}x_2 + \overline{x_1}x_3 \\ &= x_2(x_1+\overline{x_1}) + \overline{x_1}x_3 \\ &= x_2 + \overline{x_1}x_3 \end{aligned}$$

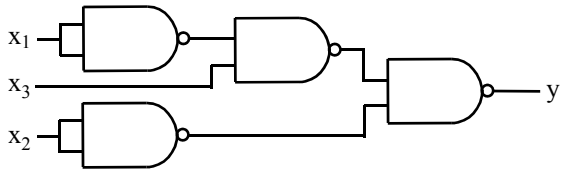
手順5：AND, OR, NOTゲートを用いて論理回路を作成する（AND-OR-NOT形式）

上式 $y=x_2+\overline{x_1}x_3$ を論理回路図で示す。



手順6：NAND, またはNORゲートに統一するかを決定する（テクノロジマッピング）

NANDで統一すると、次の論理回路図となる。

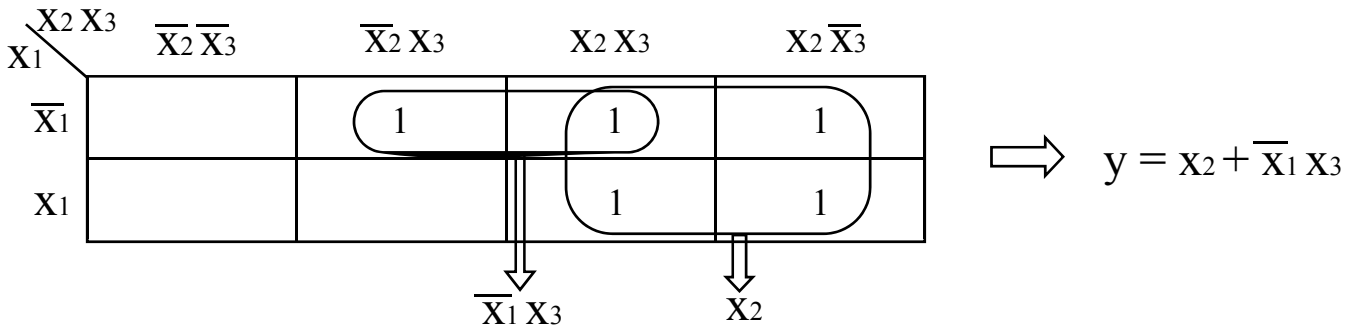


カルノー図(Karnaugh graph)による論理関数の簡略化

AとA̅の組み合わせを探す作業を機械的に行う。

導出された加法標準形論理式（次式）をカルノー図により簡略化し、検証する。

$$y = x_1x_2x_3 + x_1x_2\overline{x_3} + \overline{x_1}x_2x_3 + \overline{x_1}x_2\overline{x_3} + \overline{x_1}\,\overline{x_2}x_3$$



$$\Rightarrow\quad y = x_2 + \overline{x_1}\,x_3$$

したがって、カルノー図から得られた論理式は代数により得られた式(前記手順4の箇所)と一致し、上記論理回路が得られる。

組合せ論理回路の例

(1) 1ビットの足し算用半加算器 (half adder)

$$\begin{array}{r} 1 \cdots\cdots x \\ +\ 1 \cdots\cdots y \\ \hline 10 \end{array}$$

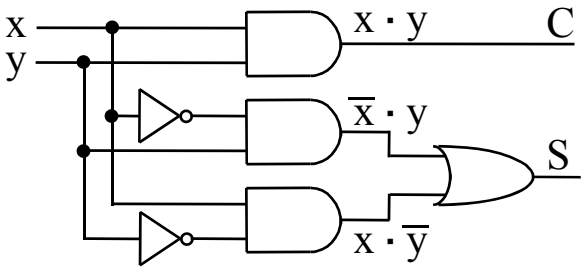
足し算結果の1ビット目：S
桁上がり：C

x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

2つの1ビットの2進数x, yの足し算は、上の式のように行われ、その真理値表は上の表となる。
この真理値表より、SおよびCの論理式を求めると、次のようになる。

$$S = \overline{x} \cdot y + x \cdot \overline{y}$$
$$C = x \cdot y$$

したがって、これらSおよびCを示す論理回路は次のようになる。これを半加算器という。



上図のように、CはANDであり、SはEx-OR(XOR)である。

(2) 2ビット以上の足し算用全加算器 (full adder)

$$
\begin{array}{rl}
 & 1 \quad \cdots\cdots C_{-1}：1ビット目の足し算により生じた桁上がり \\
 & 01 \quad \cdots\cdots x \\
+ & 11 \quad \cdots\cdots y \\
\hline
 & 100
\end{array}
$$

1ビット目の結果：S_1
2ビット目の結果：S_2
桁上がり：C

C_{-1}	x	y	S	C
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

半加算器の組合せで多数ビットの足し算回路が構成できる。上の例の場合の真理値表は上の表となる。
ただしC_{-1}は、xおよびyの1ビット目の足し算の結果生じた桁上がりである。
この真理値表から加法標準形の論理式を求めると次式となる。

$$S = \overline{x} \cdot y \cdot \overline{C_{-1}} + x \cdot \overline{y} \cdot \overline{C_{-1}} + \overline{x} \cdot \overline{y} \cdot C_{-1} + x \cdot y \cdot C_{-1}$$
$$C = x \cdot y \cdot \overline{C_{-1}} + \overline{x} \cdot y \cdot C_{-1} + x \cdot \overline{y} \cdot C_{-1} + x \cdot y \cdot C_{-1}$$

簡単化のため、カルノー図を書く。

	$\overline{x}\overline{y}$	$\overline{x}y$	xy	$x\overline{y}$
C_{-1}	1		1	
$\overline{C_{-1}}$		1		1

Sに対するカルノー図

	$\overline{x}\overline{y}$	$\overline{x}y$	xy	$x\overline{y}$
C_{-1}		1	1	1
$\overline{C_{-1}}$			1	

Cに対するカルノー図

Sに対しては、隣接項がないため簡単化できない。Cに対しては次式となる。

$$C = x \cdot y + y \cdot C_{-1} + x \cdot C_{-1}$$

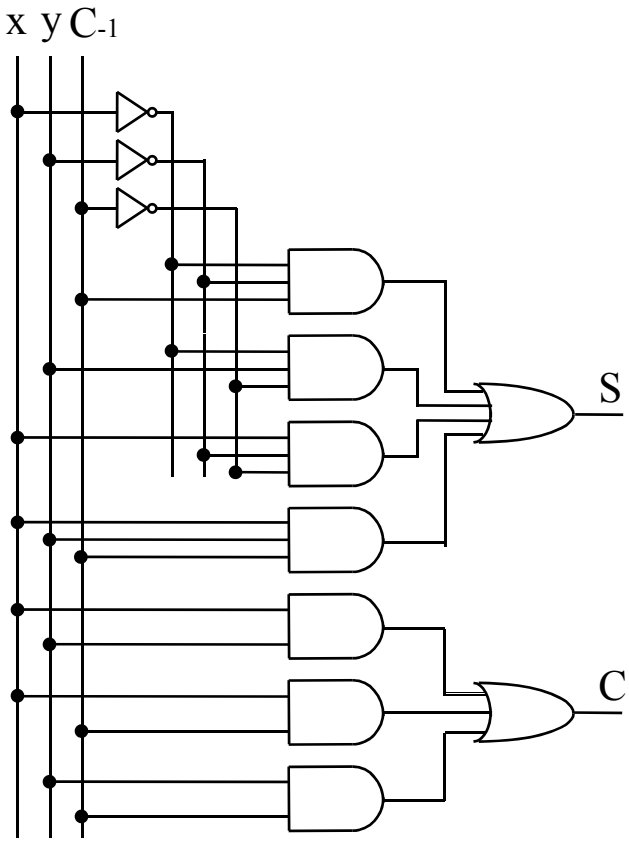
これらの結果から、論理式および論理図は次のように得られる。

論理式

$$S = \overline{x} \cdot y \cdot \overline{C_{-1}} + x \cdot \overline{y} \cdot \overline{C_{-1}} + \overline{x} \cdot \overline{y} \cdot C_{-1} + x \cdot y \cdot C_{-1}$$

$$C = x \cdot y + y \cdot C_{-1} + x \cdot C_{-1}$$

論理回路図



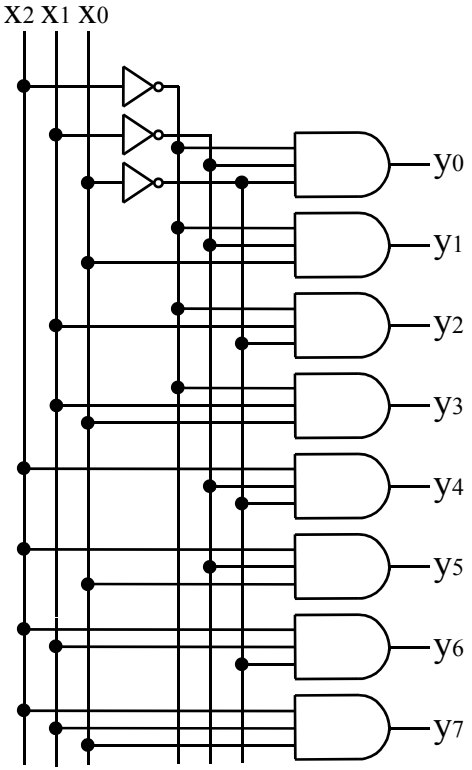
実際の全加算器は、4個用いた7480などがある。

(3) デコーダ (decoder)

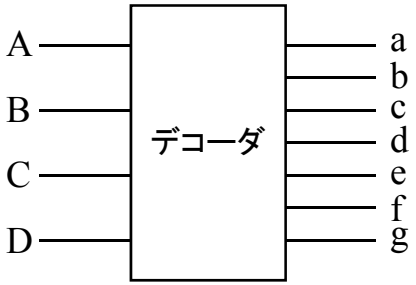
デコーダは入力符号を解読する論理回路で、入力に対応した出力線を選択する論理回路。
下の真理値表の時、出力の論理式および論理回路を以下に示す。

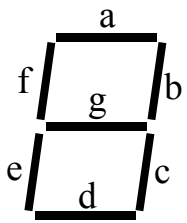
x_2 x_1 x_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0 0 0	1	0	0	0	0	0	0	0
0 0 1	0	1	0	0	0	0	0	0
0 1 0	0	0	1	0	0	0	0	0
0 1 1	0	0	0	1	0	0	0	0
1 0 0	0	0	0	0	1	0	0	0
1 0 1	0	0	0	0	0	1	0	0
1 1 0	0	0	0	0	0	0	1	0
1 1 1	0	0	0	0	0	0	0	1

$$y_0 = \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0}$$
$$y_1 = \overline{x_2} \cdot \overline{x_1} \cdot x_0$$
$$y_2 = \overline{x_2} \cdot x_1 \cdot \overline{x_0}$$
$$y_3 = \overline{x_2} \cdot x_1 \cdot x_0$$
$$y_4 = x_2 \cdot \overline{x_1} \cdot \overline{x_0}$$
$$y_5 = x_2 \cdot \overline{x_1} \cdot x_0$$
$$y_6 = x_2 \cdot x_1 \cdot \overline{x_0}$$
$$y_7 = x_2 \cdot x_1 \cdot x_0$$



下図は、4ビットの入力信号を解読して、7区分された数字表示のLEDを発光させる論理回路を示す。
真理値表を右図に示す。出力は0で点灯、1で停止する。





10進数	入力 D	C	B	A	出力 a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

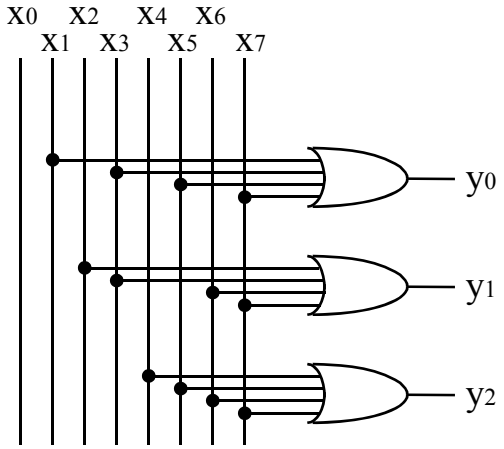
実際の論理回路は、74LS247等がある。

(4) エンコーダ (encoder)

エンコーダはデコーダの逆作用の論理回路で、複数の入力を持ち、選択された入力に対応する符号を出力する。
下の真理値表の時、出力の論理式および論理図は以下のようになる。

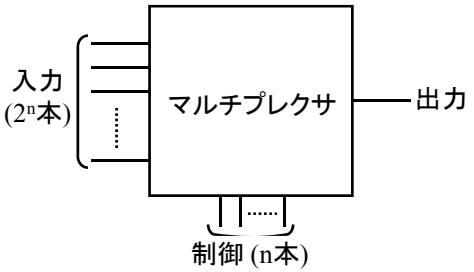
x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_2	y_1	y_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$y_2 = x_4 + x_5 + x_6 + x_7$$
$$y_1 = x_2 + x_3 + x_6 + x_7$$
$$y_0 = x_1 + x_3 + x_5 + x_7$$



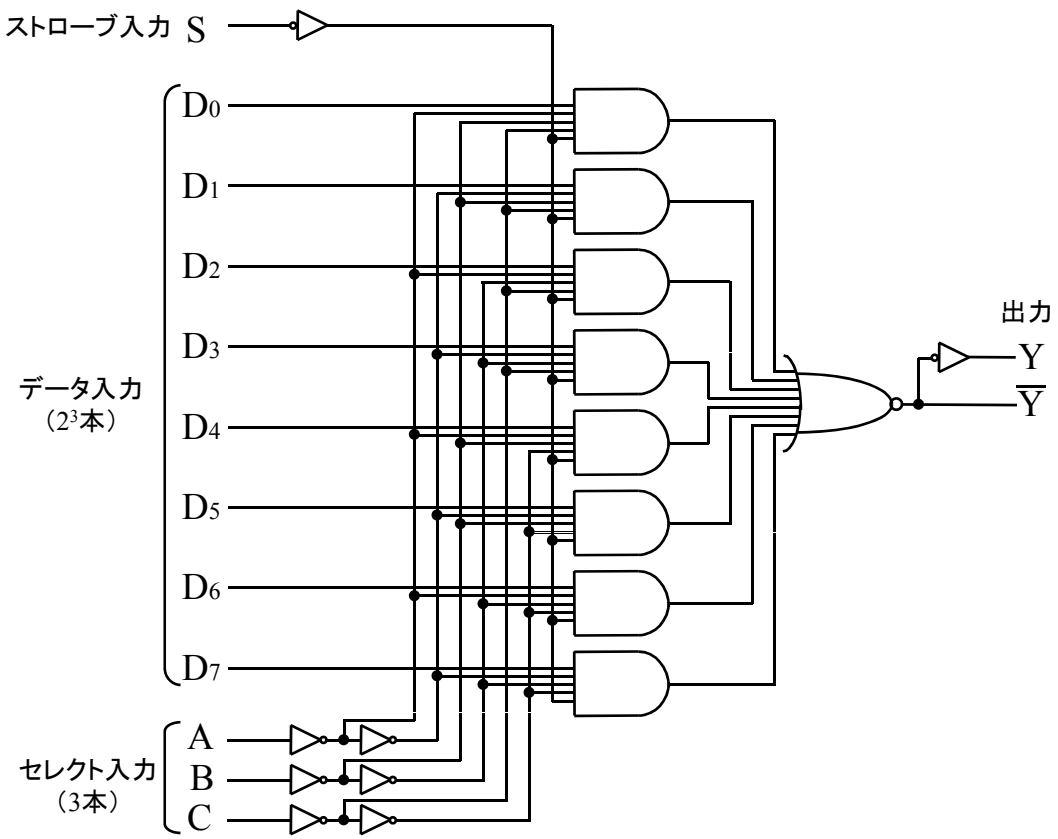
(5) マルチプレクサ (multiplexer: MUX)

マルチプレクサは、多数の情報線の中から制御信号によって1本の情報線を取り出す選択回路。
下図に原理図を示す。n本の制御線で、最大2^n個の入力線を選択できる。
右図に実例として74LS151の論理回路を示す。



入力			出力	
C	B	A	S	Y	$\overline{Y}$
x	x	x	1	0	1
0	0	0	0	D_0	$\overline{D_0}$
0	0	1	0	D_1	$\overline{D_1}$
0	1	0	0	D_2	$\overline{D_2}$
0	1	1	0	D_3	$\overline{D_3}$
1	0	0	0	D_4	$\overline{D_4}$
1	0	1	0	D_5	$\overline{D_5}$
1	1	0	0	D_6	$\overline{D_6}$
1	1	1	0	D_7	$\overline{D_7}$

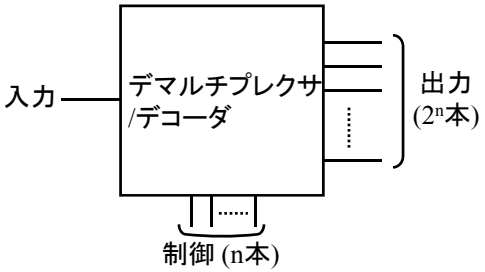
74LS151の真理値表



74LS151の論理回路

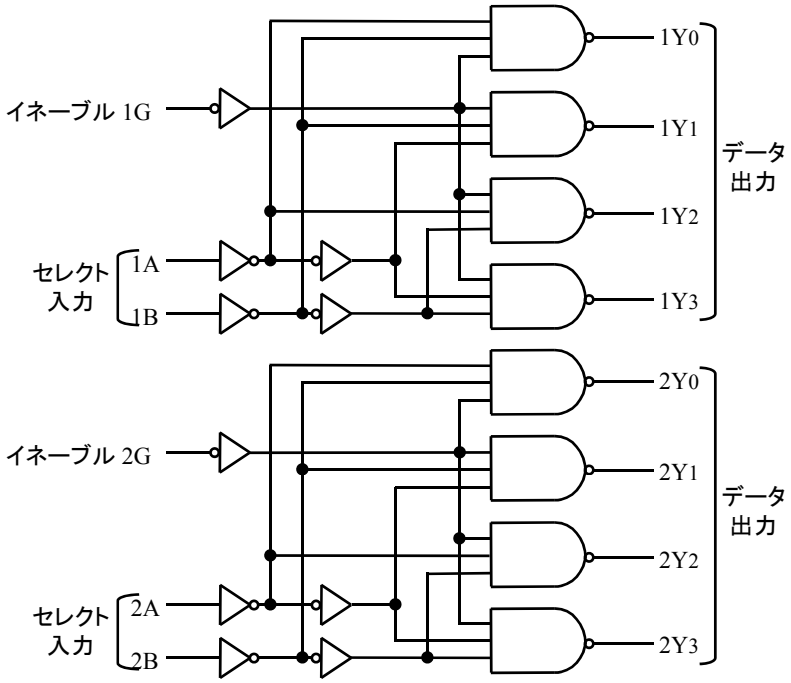
(6) デマルチプレクサ (demultiplexer: DEMUX)

デマルチプレクサは、マルチプレクサの逆の働きをするもので、1本の情報線の情報を制御信号によって、多数の情報線の中から1本を選択して送り出す選択回路。なおデマルチプレクサは、制御線を入力、入力を制御として用いるとデコーダとなる。
下図に原理図を示す。n本の制御線で、最大2^n個の出力線を選択できる。
右図に実例として74LS139の論理回路を示す。



入 力			出 力			
イネーブル	セレクト					
G	B	A	Y_0	Y_1	Y_2	Y_3
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

74LS139の真理値表



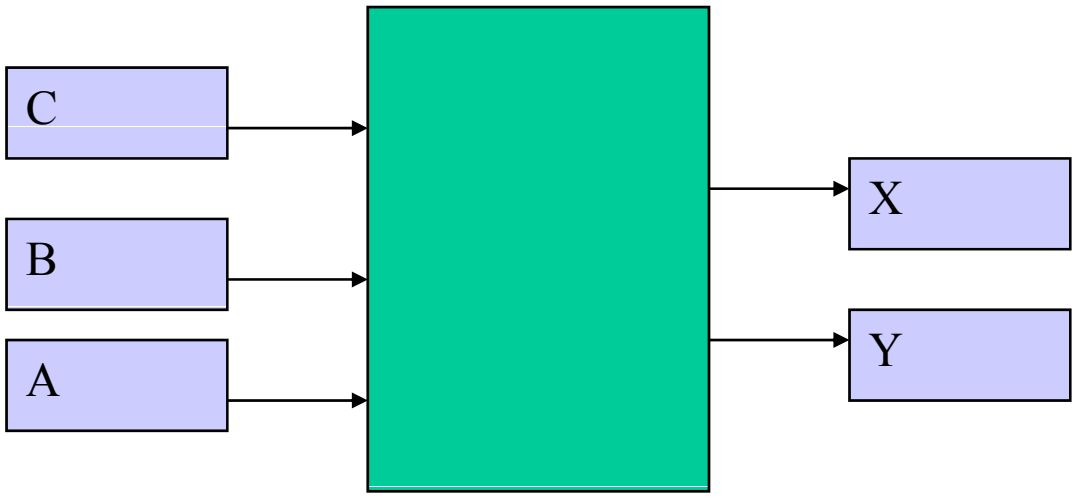
74LS139の論理回路

実験課題
プライオリティエンコーダの設計

・回路図面入力を用いて以下の回路を作成せよ。

・回路設計にあたり真理値表を作成し、回路動作が真理値表と同じ内容となるかを実験（シミュレーション）によって確認せよ。

・レポートには真理値表、回路図、シミュレーション結果それらの説明の記述が必修である。



$$X = C + B$$

$$Y = C + \bar{B}A$$

考察課題

レポートを作成するにあたって以下の問の解答を含めること。

問1．組合せ論理回路とはなにか。

問2．半加算器、全加算器とはどういう加算器か。違いがわかるように述べよ

問3．デコーダ、エンコーダとはなにか。

問4．マルチプレクサ、デマルチプレクサとはなにか。

問5．次の組合せ回路を(多入力)NAND素子だけを用いて構成せよ。
Maxplus2を使用して回路を設計し、回路図とシミュレーション結果を
レポートに添付せよ

(1) 4入力×1出力マルチプレクサ
(2) 1入力×4出力デマルチプレクサ
(3) 2ビットデコーダ（出力は4本）

実験レポートの最後の項に考察課題として記述すること。

課題（レポート補足を含む）

- 各自のラップトップPC にMaxplusのインストール、ライセンスの取得と設定をおこなう（次回実験までに）
- 各自のラップトップPC上のMaxplusを使ってプライオリティ・エンコーダ回路を作成し、そのシミュレーション結果をレポートに添付する。
- 次回実験に各自のノートPCを持参すること

レポート必須事項（その１）

- 設計した回路図（スイッチ、抵抗、Vcc、GNDも）
- 部品表
- 回路図の説明
- 真理値表、論理式、状態遷移図、カルノー図など
- 実験結果とその説明（何を変えて実験したか）
- 考察と感想は別項目でかく
- 回路図はオリジナルに限る
  （他人の情報をレポートにコピーするのは禁止）

レポート必須事項（その２）

- 実験の使用機器は製品名、メーカー名、モデル名、製造番号をかく
- オシロスコープの波形はFDに保存したデータをエクセルでグラフ化してレポートに載せる。スケール（Time/div、Volt/div）を記述すること。
- レポートの体裁

  ①ページ番号、図表のCaption（表は上部、図は下部へ）

  ②参考文献は考察の項の次に書く。