

---

# 実習で学ぶ論理回路

2011.5.9 改訂  
弘前大学 小山智史

---

氏 名： ( [@stu.hirosaki-u.ac.jp](mailto:@stu.hirosaki-u.ac.jp) )



# 第1章 ブール代数と論理回路

実習装置を用いて動作の検証を行うことにより、論理回路に対する理解を深め、応用力を養います。

## 1.1 ブール代数と命題論理

ブール代数では、以下のように2値のブール変数と3つの論理演算が定義されます。

(公理1) 論理値 あるブール変数  $x$  の値 (論理値) は「0」か「1」のいずれかである。

(公理2) NOT 「0」のNOTは「1」、 「1」のNOTは「0」である。

(公理3) AND 「1」どうしのANDだけが「1」であり、他の3通りの組み合わせのANDは「0」である。

(公理4) OR 「0」どうしのORだけが「0」であり、他の3通りの組み合わせのORは「1」である。

このように、ブール代数は0と1しかない世界の中で構築された数学です。ここで、 $x$  のNOTを  $\bar{x}$ 、 $x$  と  $y$  のANDを  $x \cdot y$ 、 $x$  と  $y$  のORを  $x + y$  という演算記号を用いて表します<sup>1</sup>。

さて、

「本州の最北端の県は青森県である」

「弘前は青森県の県庁所在地である」

「青森の天気は晴である」

「弘前の天気は雨である」

「このリンゴの糖度は十分である」

「このリンゴにはキズがある」

のように真または偽である言明は命題と呼ばれます。

ここで、論理値「1」を真、「0」を偽に対応させると、ブール代数を用いて命題論理を扱うことができます。

例えば、ブール変数  $x$  が「このリンゴにはキズがある」という命題を表すものとします。この命題が真の時、 $x$  は1、 $\bar{x}$  は0となる。また、この命題が偽の時  $x$  は0、 $\bar{x}$  は1となります。

## 1.2 ブール関数と真理値表

$n$  個のブール変数 (入力) に対し、1個のブール関数 (出力) を決めたいとします。一般に  $n$  個のブール変数  $x_1, x_2, \dots, x_n \in \{0, 1\}$  のそれぞれに対し論理値を与えると、ブール関数  $y = f(x_1, x_2, \dots, x_n) \in$

<sup>1</sup>ただし、AND 演算の演算記号「 $\cdot$ 」を省略して「 $xy$ 」と表すこともあります。

0, 1 が決まります。ここで、 $2^n$  通りあるブール変数の取り得るすべての組み合わせに対応するブール関数の値を表にまとめた真理値表によりブール関数を記述することができます。

表 1.1 は、ブール代数で定義される 3 つの論理演算を真理値表で示したものです。

表 1.1: 3 つの論理演算の真理値表

(a) NOT		(b) AND			(c) OR		
$x$	$y = \bar{x}$	$x_1$	$x_2$	$y = x_1 \cdot x_2$	$x_1$	$x_2$	$y = x_1 + x_2$
0	1	0	0	0	0	0	0
0	1	0	1	0	0	1	1
1	0	1	0	0	1	0	1
1	0	1	1	1	1	1	1

ブール関数の例として、 $y = x_1 \cdot \bar{x}_2$  を考えてみましょう。この真理値表は以下のようになります。

$x_1$	$x_2$	$y = x_1 \cdot \bar{x}_2$
0	0	0
0	1	0
1	0	1
1	1	0

今、 $x_1$  を「このリンゴの糖度は十分である」、 $x_2$  を「このリンゴにはキズがある」という命題とすると、「このリンゴの糖度は十分である」が真でかつ「このリンゴにはキズがある」が偽の時に  $y$  は真となります。選果場のベルトコンベアにやってくるリンゴひとつひとつについてセンサから  $x_1, x_2$  の値が得られたとしたら、 $y$  は選別の指標としての意味を持つかもしれません。

### 1.3 論理素子と論理回路

ブール代数で定義される NOT, AND, OR の論理演算を論理素子で表し、ブール関数とその結線として表したものが論理回路です。

図 1.1 に論理素子の回路記号を示します。

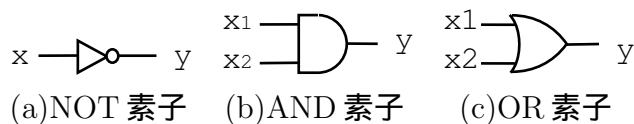


図 1.1: 論理素子の回路記号

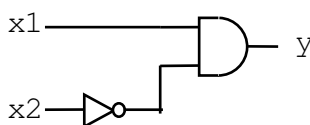


図 1.2: 選果判定機械の論理回路

先に示したリングの選果の例で、 $y = x_1 \cdot \overline{x_2}$  に対応する論理回路は図 1.2 のように表すことができます。

(実習) 実習装置を用いて NOT 素子の入出力の関係を調べ、以下の真理値表を完成させなさい。

$x$	$y = \overline{x}$
0	
1	

(実習) 実習装置を用いて AND 素子の入出力の関係を調べ、以下の真理値表を完成させなさい。

$x_1$	$x_2$	$y = x_1 \cdot x_2$
0	0	
0	1	
1	0	
1	1	

(実習) 実習装置を用いて OR 素子の入出力の関係を調べ、以下の真理値表を完成させなさい。

$x_1$	$x_2$	$y = x_1 + x_2$
0	0	
0	1	
1	0	
1	1	

(実習) 実習装置を用いて図 1.2 の回路の入出力の関係を調べ、以下の真理値表を完成させなさい。

$x_1$	$x_2$	$y = x_1 \cdot \overline{x_2}$
0	0	
0	1	
1	0	
1	1	

## 1.4 その他の論理素子

NOT, AND, OR の 3 つの演算子に対応する論理素子があれば、任意の論理回路を実現できますが、これらの論理素子を組み合わせた論理素子があると好都合な場合があります。

3 入力の AND 素子 ( $y = x_1 \cdot x_2 \cdot x_3 = (x_1 \cdot x_2) \cdot x_3$ ) は、2 入力の AND 素子を組み合わせることにより実現できます (図 1.3)。

NAND、NOR はそれぞれ図 1.3(b)(c) の回路記号で表されます。また、eXclusive OR ( $y = \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}$ ) は (d) の回路記号で表されます。

1.7 で示されるように、NAND 素子を用いて NOT、AND、OR のいずれをも構成することができます。同様に、NOR 素子を用いて NOT、AND、OR のいずれをも構成することができます。従って、NAND 素子あるいは NOR 素子 1 種類を用いてあらゆる論理回路を構成することができます。

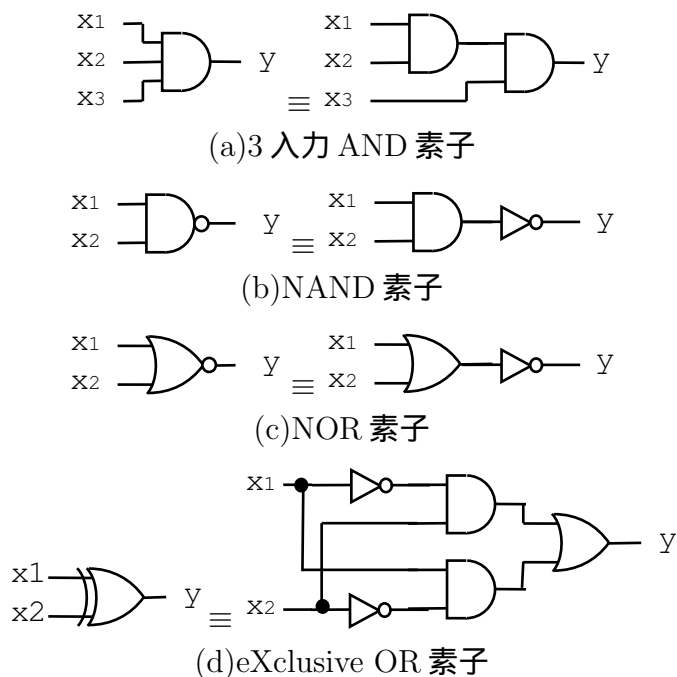


図 1.3: その他の論理素子

(実習) 実習装置を用いて、3 入力 AND 素子について図 1.3(a) 左右の回路の入出力の関係を調べ、以下の真理値表を完成させなさい。

$x_1$	$x_2$	$x_3$	$y = x_1 \cdot x_2 \cdot x_3$	$y = (x_1 \cdot x_2) \cdot x_3$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

(実習) 実習装置を用いて、NAND 素子について図 1.3(b) 左右の回路の入出力の関係を調べ、真理値表で表しなさい。

(実習) 実習装置を用いて、NOR 素子について図 1.3(c) 左右の回路の入出力の関係を調べ、真理値表で表しなさい。

(実習) 実習装置を用いて、eXclusive OR 素子について図 1.3(d) 左右の回路の入出力の関係を調べ、真理値表で表しなさい。

## 1.5 論理素子の内部回路

論理素子の内部は電子回路で構成されています。図 1.4 は、NOT 素子、AND 素子、OR 素子の各々について、内部回路の例を示したものです。ここで、論理値 1 を High(高い電圧)、0 を Low(低い電圧) に対応させています。

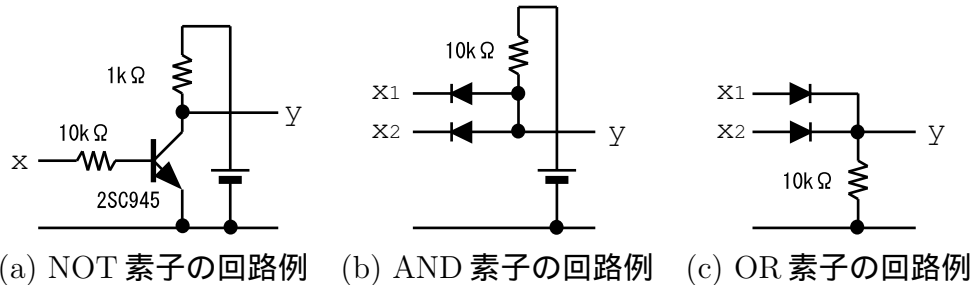


図 1.4: 論理素子の内部回路

実際に用いられている論理素子の内部回路はもっと複雑で、入出力の電圧-電流特性が保証されています<sup>2</sup>。具体的な NOT 素子 (TI 社 SN7404) の入出力の特性を図 1.5 に示します。実習装置で用いている SN74LS04 もほぼ同様です。SN74LS04 は、ある条件下で、出力の High は 2.4V 以上、Low は 0.4V 以下を保証し、また入力は 2.0V 以上の時に High、0.8V 以下の時に Low とみなすように設計されています (表 1.2)。これから、素子を互いに接続した時の雑音に対するマージンが確保されていることがわかります。

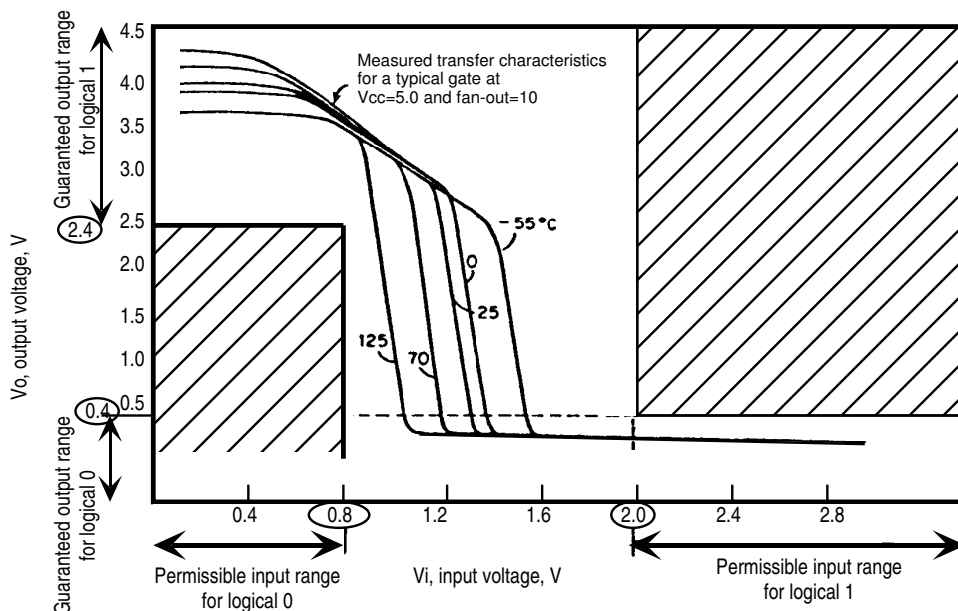


図 1.5: NOT 素子 SN7404 の入出力特性

<sup>2</sup>図 1.4 の回路例は模式的なものであり、組み合わせて接続すると不都合を来すことがあります。

表 1.2: NOT 素子 SN74LS04 の規格 (抜粋)

	MIN	NOM	MAX	UNIT	TEST CONDITIONS
$V_{CC}$ , Supply voltage	4.5	5	5.5	V	
$I_{OH}$ , High-level output current			-400	$\mu$ A	
$I_{OL}$ , Low-level output current			8	mA	
$V_{IH}$ , High-level input voltage	2			V	
$V_{IL}$ , Low-level input voltage			0.8	V	
$V_{OH}$ , High-level output voltage	2.7	3.4		V	$V_{CC} = \text{MIN}$ , $V_{IL} = V_{IL} \text{ Max}$ , $I_{OH} = \text{MAX}$
$V_{OL}$ , Low-level output voltage			0.4	V	$V_{CC} = \text{MIN}$ , $V_{IH} = 2\text{V}$ , $I_{OL} = 4\text{mA}$
$I_{IH}$ , High-level input current			20	$\mu$ A	$V_{CC} = \text{MAX}$ , $V_{IH} = 2.7\text{V}$
$I_{IL}$ , Low-level input current			-0.4	mA	$V_{CC} = \text{MAX}$ , $V_{IL} = 0.4\text{V}$

(実習) 実習装置の NOT 素子 (SN74LS04) について、入力を 0 および 1 にした時の、入力と出力の電圧を測定しなさい。

(実習) 図 1.4(a) ~ (c) の回路を組み立て、入出力の関係を調べ、真理値表で表しなさい。その際、真理値表には、0,1 だけでなく、電圧値も測定して書き込みなさい。

(実習) 図 1.4(a) の NOT 回路を組み立て、図 1.5 に相当する入出力の特性を調べなさい。また、この回路の交流小信号増幅回路としての動作について考察しなさい。

## 1.6 論理 IC(Integrated Circuit: 集積回路) の規模

実習装置で用いている論理素子は、6 個の NOT 素子、4 個の AND 素子、4 個の OR 素子が、各々 1 個の IC にパッケージされています。この規模の IC は SSI(Small Scale Integrated circuit) と呼ばれます。より集積度の高い IC は、MSI, LSI, 超 LSI などと呼ばれます (表 1.3)。素子数とトランジスタ数の数え方の違いはありますが、Pentium III は 950 万個、Pentium IV は 4200 万個のトランジスタからなる超 LSI です。

表 1.3: IC の規模

IC の規模	1 個の IC 中の素子数
SSI	~ 30 個
MSI	30 ~ 1000 個
LSI	1000 ~ 10 万個
超 LSI	10 万個 ~

## 1.7 ブール代数の基本公式

ブール代数の公理から表 1.4 に示す種々の基本公式が導かれます。どれも、真理値表を用いて容易に証明することができます。

表 1.4: 論理代数の基本公式

吸収則	$1 + x = 1$ $0 + x = x$ $x + x \cdot y = x$	$1 \cdot x = x$ $0 \cdot x = 0$ $x \cdot (x + y) = x$
交換則	$x + y = y + x$	$x \cdot y = y \cdot x$
結合則	$x + (y + z) = (x + y) + z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
分配則	$x \cdot (y + z) = x \cdot y + x \cdot z$	$x + y \cdot z = (x + y) \cdot (x + z)$
べき等則	$x + x = x$	$x \cdot x = x$
相補則	$x + \bar{x} = 1$	$x \cdot \bar{x} = 0$
二重否定	$\overline{\bar{x}} = x$	
ド・モルガンの定理	$\overline{x + y} = \bar{x} \cdot \bar{y}$	$\overline{x \cdot y} = \bar{x} + \bar{y}$

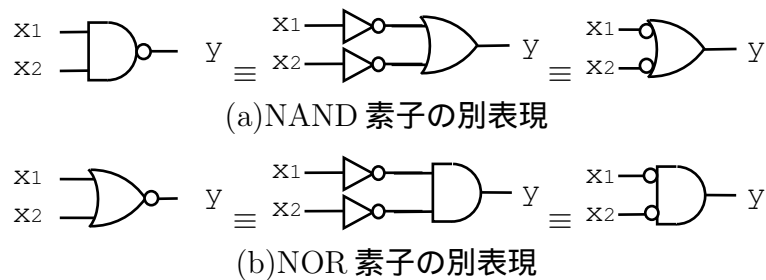


図 1.6: ド・モルガンの定理と論理素子の別表現

ド・モルガンの定理は、NAND 素子が図 1.6(a) のように別表現することができることを表しています。同様に NOR 素子は図 1.6(b) のように別表現することができます。

また、図 1.7 に示すように、べき等則により、NAND 素子や NOR 素子を用いて NOT 素子を実現できます。ド・モルガンの定理と合わせて考えれば、NAND 素子のみを用いて NOT, AND, OR と等価な回路を構成できることがわかります。同様に NOR 素子のみを用いて NOT, AND, OR と等価な回路を構成できます。

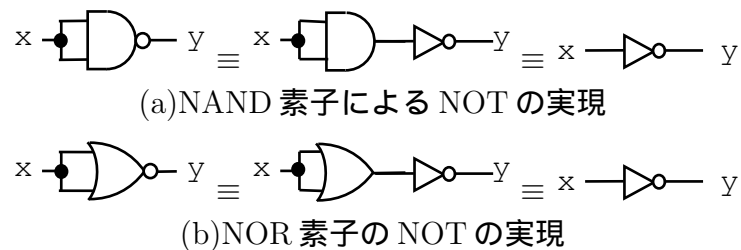


図 1.7: NAND 素子や NOR 素子による NOT 素子の実現

さて、これまでは真理値表を用いて論理演算や論理素子の機能を表示してきましたが、集合で用いられるベン図を用いると理解しやすいでしょう。図 1.8 は NOT, AND, OR の各演算の機能をベン図で示したものです。

また、図 1.9 は、ド・モルガンの定理をベン図を用いて説明したものです。

(練習) 表 1.4 の結合則と分配則をベン図を用いて説明しなさい。

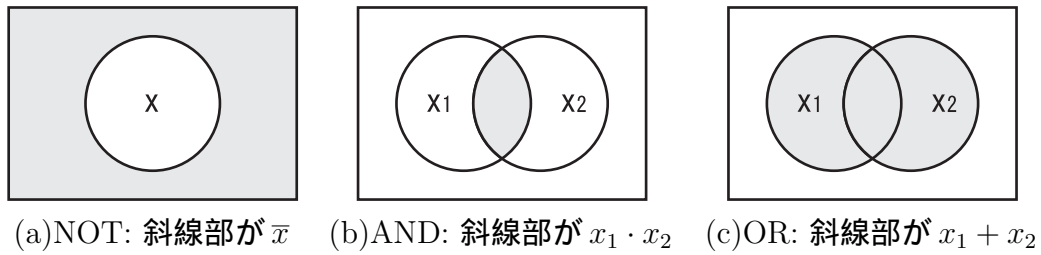


図 1.8: ベン図による論理演算の表現

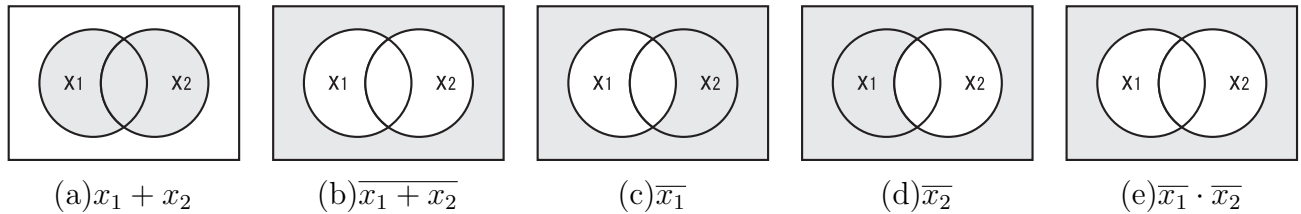


図 1.9: ベン図によるド・モルガンの定理の説明

(練習) NAND、NOR、eXclusive OR の各々の素子の機能をベン図を用いて示しなさい。

(実習) ド・モルガンの定理について、図 1.6(a) 左と中央の回路が等価であることを、実習装置を用いて検証しなさい。同様に、(b) 左と中央の回路が等価であることを検証しなさい。

(実習) NAND 素子を用いて NOT, AND, OR の各素子と等価な回路を作り、真理値表で表し、動作を検証しなさい。

(実習) NOR 素子を用いて NOT, AND, OR の各素子と等価な回路を作り、真理値表で表し、動作を検証しなさい。

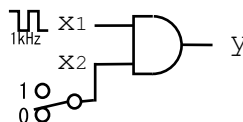
(実習) ド・モルガンの定理の左右両辺の否定をとると

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

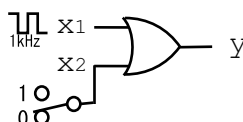
$$x \cdot y = \overline{\overline{x} + \overline{y}}$$

となる。これに基づき、図 1.6 を参考に、AND および OR の回路記号の別表現を示し、実習装置を用いて動作を検証しなさい。

(実習) 下記の回路について、スイッチを切り替えながら入出力の様子をオシロスコープで観測し、タイムチャートで示しなさい。



(実習) 下記の回路について、スイッチを切り替えながら入出力の様子をオシロスコープで観測し、タイムチャートで示しなさい。



## 第2章 組み合わせ論理回路

組み合わせ論理回路は、過去の入力履歴には依存せず、現在の入力の状態のみで出力が決まります (図 2.1)。その設計は概ね次の手順で行います。

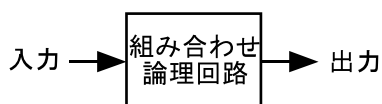


図 2.1: 組み合わせ論理回路

- (1) 与えられた問題を真理値表として表現する。
- (2) 真理値表からブール関数を求める。
- (3) ブール関数を簡単化する。
- (4) 簡単化したブール関数を論理回路として表す。

### 2.1 組み合わせ論理回路の設計

#### 主加法標準形

ブール関数  $y = f(x_1, x_2, \dots, x_n)$  を  $x_1 = 0$  を含む項と  $x_1 = 1$  を含む項に分けると

$$f(x_1, x_2, \dots, x_n) = \bar{x}_1 \cdot f(0, x_2, \dots, x_n) + x_1 \cdot f(1, x_2, \dots, x_n)$$

となります。1変数なら

$$\begin{aligned} f(x) &= \bar{x} \cdot f(0) + x \cdot f(1) \\ &= \bar{x} \cdot f(0) + x \cdot f(1) \end{aligned}$$

2変数なら

$$\begin{aligned} f(x_1, x_2) &= \bar{x}_1 \cdot f(0, x_2) + x_1 \cdot f(1, x_2) \\ &= \bar{x}_1 \cdot (\bar{x}_2 \cdot f(0, 0) + x_2 \cdot f(0, 1)) + x_1 \cdot (\bar{x}_2 \cdot f(1, 0) + x_2 \cdot f(1, 1)) \\ &= \bar{x}_1 \cdot \bar{x}_2 \cdot f(0, 0) + \bar{x}_1 \cdot x_2 \cdot f(0, 1) + x_1 \cdot \bar{x}_2 \cdot f(1, 0) + x_1 \cdot x_2 \cdot f(1, 1) \end{aligned}$$

一般には

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= \overline{x_1} \cdot \overline{x_2} \cdot \dots \cdot \overline{x_n} \cdot f(0, 0, \dots, 0) \\ &\quad + \overline{x_1} \cdot \overline{x_2} \cdot \dots \cdot x_n \cdot f(0, 0, \dots, 1) \\ &\quad \dots \\ &\quad + x_1 \cdot x_2 \cdot \dots \cdot x_n \cdot f(1, 1, \dots, 1) \end{aligned}$$

と合計  $2^n$  通りの項に展開されます。ブール関数をこのような形に展開したものを主加法標準形といいます。各論理積の部分を最小項とよびます。

ここで

$$f(0, 0, \dots, 0), f(0, 0, \dots, 1), f(1, 1, \dots, 1) \in \{0, 1\}$$

は真理値表で与えられ、 $f(\dots) = 0$ の項は除去できます。

この結果、主加法標準形よりブール関数は次のようにして求められます。

1. 真理値表より、ブール関数値が1をとる変数の組み合わせだけを取り出す。
2. ついで、それらのブール変数が0には $\overline{x_i}$ 、1には $x_i$ を対応させ、その論理積を作る。
3. それらの論理和をとったものを論理式とする。

以下、具体例で考えてみましょう。

図 2.2(a) の真理値表を満たす論理回路を実現したいとします。同図 (b) はこれをベン図で表したものです。

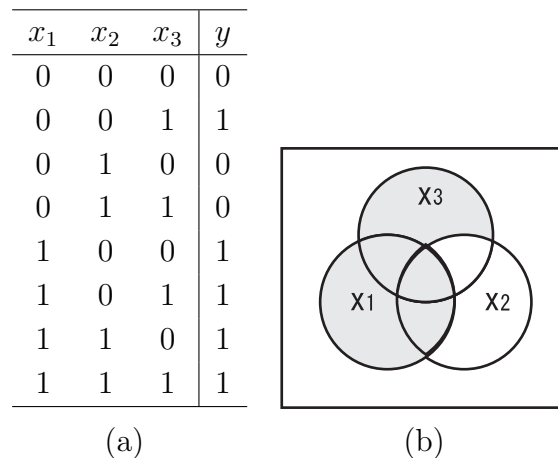


図 2.2: 設計したい論理回路の真理値表の例とベン図による表現

ここで、この真理値表は1.2節で例として取り上げたリンゴの選果システムの論理判断を表すものとしてとします。 $x_1$ を「このリンゴの糖度は十分である」、 $x_2$ を「このリンゴにはキズがある」、 $x_3$ を「このリンゴは色が良い」という命題に対応させ、適当なセンサを用いることにより、それぞれについて1(真)または0(偽)の信号が得られているものとしてとします。 $y = 1$ となる $x_1 = 0, x_2 = 0, x_3 = 1$ は、「このリンゴの糖度は十分でなく」、「このリンゴにはキズがなく」、「このリンゴは色が良い」ことを表して、この時に $y = 1$ つまり「出荷OK」を出力する。 $y = 1$ とする他の条件をも加え、この信号を用いてアクチュエータを制御すれば、リンゴを物理的に振り分けることができます。

上に示した1~3の手順に従えば、次のように論理式を求めることができます。

1. 真理値表より、 $y = 1$  をとる変数の組み合わせを取り出すと、

$$x_1 = 0, x_2 = 0, x_3 = 1$$

$$x_1 = 1, x_2 = 0, x_3 = 0$$

$$x_1 = 1, x_2 = 0, x_3 = 1$$

$$x_1 = 1, x_2 = 1, x_3 = 0$$

$$x_1 = 1, x_2 = 1, x_3 = 1$$

である。

2. それらの  $x_i = 0$  には  $\bar{x}_i$ 、 $x_i = 1$  には  $x_i$  を対応させ、その論理積を作ると、

$$\bar{x}_1 \cdot \bar{x}_2 \cdot x_3, x_1 \cdot \bar{x}_2 \cdot \bar{x}_3, x_1 \cdot \bar{x}_2 \cdot x_3, x_1 \cdot x_2 \cdot \bar{x}_3, x_1 \cdot x_2 \cdot x_3$$

となる。

3. それらの論理和をとったものを  $y$  とすると、

$$y = \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3$$

となる (最小項)。

これを回路図として描いたものが図 2.3 です。

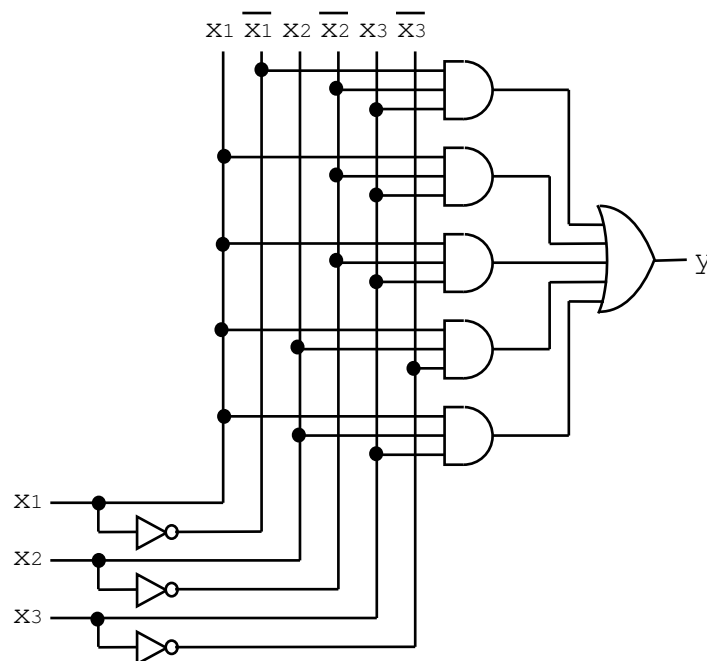


図 2.3: 主加法標準形で導いた論理回路

(実習) 次の真理値表を満たす論理回路を実現したい。

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

(1) 主加法標準形で式を導きなさい。

(2) 導いた式を回路図として描きなさい。また、実習装置を用いてその回路を組み立て、上の真理値表を満たしていることを確認しなさい。

### 主乗法標準形

ブール関数  $y = f(x_1, x_2, \dots, x_n)$  を  $x_1 = 0$  を含む項と  $x_1 = 1$  を含む項に分けると

$$f(x_1, x_2, \dots, x_n) = (x_1 + f(0, x_2, \dots, x_n)) \cdot (\bar{x}_1 + f(1, x_2, \dots, x_n))$$

となります。1変数なら

$$\begin{aligned} f(x) &= x \cdot \bar{x} + f(x) = (x + f(x)) \cdot (\bar{x} + f(x)) \\ &= (x + f(0)) \cdot (\bar{x} + f(1)) \end{aligned}$$

2変数なら

$$\begin{aligned} f(x_1, x_2) &= (x_1 + f(0, x_2)) \cdot (\bar{x}_1 + f(1, x_2)) \\ &= (x_1 + (x_2 + f(0, 0)) \cdot (\bar{x}_2 + f(0, 1))) \cdot (\bar{x}_1 + (x_2 + f(1, 0)) \cdot (\bar{x}_2 + f(1, 1))) \\ &= (x_1 + x_2 + f(0, 0)) \cdot (x_1 + \bar{x}_2 + f(0, 1)) \cdot (\bar{x}_1 + x_2 + f(1, 0)) \cdot (\bar{x}_1 + \bar{x}_2 + f(1, 1)) \end{aligned}$$

一般には

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= (x_1 + x_2 + \dots + x_n + f(0, 0, \dots, 0)) \\ &\quad \cdot (x_1 + x_2 + \dots + \bar{x}_n + f(0, 0, \dots, 1)) \\ &\quad \dots \\ &\quad \cdot (\bar{x}_1 + \bar{x}_2 + \dots + \bar{x}_n + f(1, 1, \dots, 1)) \end{aligned}$$

と合計  $2^n$  通りの項に展開されます。ブール関数をこのような形に展開したものを主乗法標準形といいます。各論理和の部分の最大項とよびます。

上の式からブール関数は次のようにして求められます。

1. 真理値表よりブール関数値が0をとる変数の組み合わせだけを取り出す。
2. ついで、それらのブール変数が0には  $x_i$ 、1には  $\bar{x}_i$  を対応させ、その論理和を作る。
3. それらの論理積をとったものを論理式とする。

前節と同様、次の真理値表を満たす論理回路を作ること考えましょう。

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

上に示した1~3の手順に従えば、次のように論理式を求めることができます。

1. 真理値表より、 $y = 0$ をとる変数の組み合わせを取り出すと、

$$x_1 = 0, x_2 = 0, x_3 = 0$$

$$x_1 = 0, x_2 = 1, x_3 = 0$$

$$x_1 = 0, x_2 = 1, x_3 = 1$$

である。

2. それらの  $x_i = 0$  には  $x_i$ 、 $x_i = 1$  には  $\overline{x_i}$  を対応させ、その論理和を作ると、

$$x_1 + x_2 + x_3, x_1 + \overline{x_2} + x_3, x_1 + \overline{x_2} + \overline{x_3}$$

となる(最大項)。

3. それらの論理積をとったものを  $y$  とすると、

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

となる。

これを回路図として描いたものが図2.4である。

以上のように、真理値表が与えられた時に、主加法標準形や主乗法標準形で論理式を導き、回路を構成すれば目的は達せられます。しかし、このままでは無駄が多く、次節に示す「簡単化」を行うことにより、もっと簡単な回路で実現することができます。

(実習) 次の真理値表を満たす論理回路を実現したい。

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

(1) 主乗法標準形で式を導きなさい。

(2) 導いた式を回路図として描きなさい。また、実習装置を用いてその回路を組み立て、上の真理値表を満たしていることを確認しなさい。

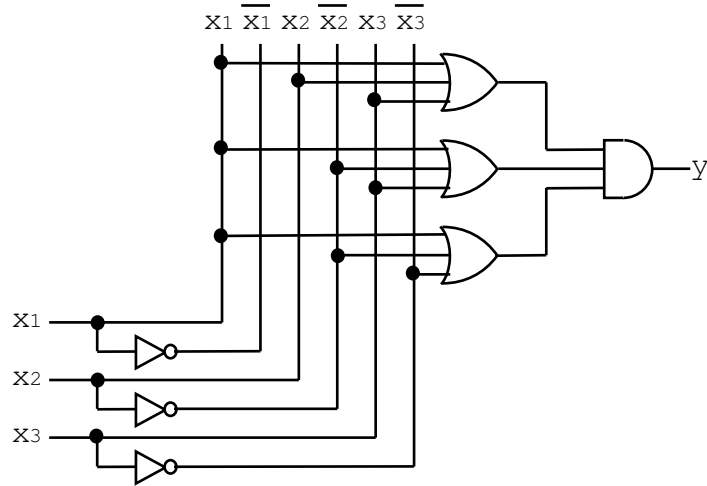


図 2.4: 主乗法標準形で導いた論理回路

## 2.2 組み合わせ論理回路の簡単化

前節のように、主加法標準形や主乗法標準形を用いて真理値表を満たす論理回路を設計することができました。しかし、このままでは多くの論理素子を用いることとなり、無駄が多いのです。そこで、同じ真理値表を満たす「より簡単な」論理回路を求める方法について示します。

ここでは、図 2.2(a) の真理値表を例に考えよう (以下に再掲)。

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

ここで、 $y = 1$  とするのは、 $x_2 = 0, x_3 = 1$  または  $x_1 = 1$  の場合ですが、図 2.5 のようなカルノー (Karnaugh) 図を用いると、このような入力の組み合わせを容易に見つけることができます。

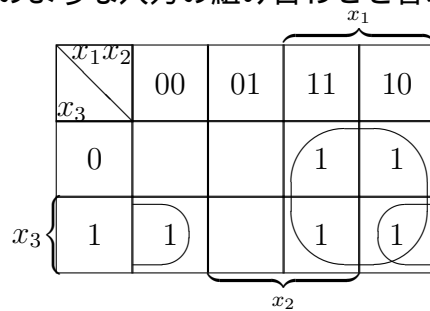


図 2.5: 図 2.2(a) の真理値表のカルノー図による表現と簡単化

カルノー図を用いた簡単化の手順は以下のとおりです。

- (1) 真理値表をカルノー図で表現する。

- (2) 表の 1 が記載された箇所を、隣接関係にある  $2^n$  個のなるべく大きな矩形で囲む。
- (3) 囲まれていない「1」の箇所が無くなるまで、(2) を繰り返す。囲みはオーバーラップしてもよい。
- (4) 囲みに相当する式を論理和で結合して式を得る。

図 2.5 の例では、まず隣接する 4 個の「1」を選んで囲み、次に隣接する 2 個の「1」を選んで囲みます。囲みは、それぞれ  $x_1$  と  $\overline{x_2} \cdot x_3$  に相当します。これらを論理和で結合し、次式を得ます。

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

回路図で表せば図 2.6 となります。この回路は図 2.2(a) の真理値表を満たし、同じ真理値表を実現する図 2.3 や図 2.4 の回路よりもはるかに簡単な回路となっていることがわかります。

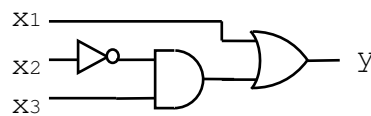


図 2.6: 図 2.2(a) の真理値を満たす簡単化した回路

4 変数の場合の簡単化の例は 3.11 にあります。また、これを並べて描くことにより、6 変数程度までは手作業による簡単化が可能です。

論理回路の簡単化の方法としては、カルノー図による方法の他に、計算機による自動化に適した Quine-McCluskey 法があります。

(実習) 実習装置を用いて図 2.6 の「りんご出荷判定回路」を組み立て、図 2.2(a) との関係を確認しなさい。

## 2.3 組み合わせ論理回路の例

### 加算器

加算器には下位からの桁上げを考慮しない半加算器 (Half Adder) と下位からの桁上げを考慮する全加算器 (Full Adder) があります。

半加算器では、図 2.7(a) の真理値表のような動作が求められます。

主加法標準形により論理式を求めると、

$$S = \overline{A} \cdot B + A \cdot \overline{B}$$

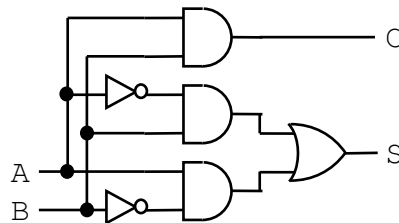
$$C = A \cdot B$$

で、これ以上簡単化することはできません。回路図で示せば図 2.7(b) のようになります。XOR(exclusive OR) 素子を用いると図 2.7(c) のように実現できます。

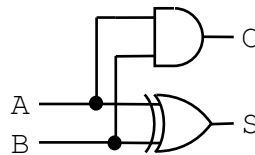
全加算器では、図 2.8(a) の真理値表のような動作が求められます。

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

(a) 半加算器の真理値表



(b) NOT AND OR 素子で実現した半加算器



(c) eXclusive OR を用いて実現した半加算器

図 2.7: 半加算器

主加法標準形により論理式を求めると、

$$S = \bar{A} \cdot \bar{B} \cdot C_i + \bar{A} \cdot B + \bar{C}_i + A \cdot \bar{B} \cdot \bar{C}_i + A \cdot B \cdot C_i$$

$$C_o = \bar{A} \cdot B \cdot C_i + A \cdot \bar{B} \cdot C_i + A \cdot B \cdot \bar{C}_i + A \cdot B \cdot C_i$$

となります。カルノー図による簡単化を試みた場合、 $S$  はこれ以上簡単化できず、 $C_o$  は

$$C_o = A \cdot B + B \cdot C_i + C_i \cdot A$$

のように簡単化できます。図 2.8(b) には、eXclusive OR 素子を用いた回路図を示します。

8 ビットの加算回路を構成するには 1 個の半加算器と 7 個の全加算器を図 2.9 のように接続します。2 つの 8 ビットの数  $A_7 \dots A_1 A_0$  と  $B_7 \dots B_1 B_0$  を入力し、加算結果が  $S_7 \dots S_1 S_0$  に得られます。

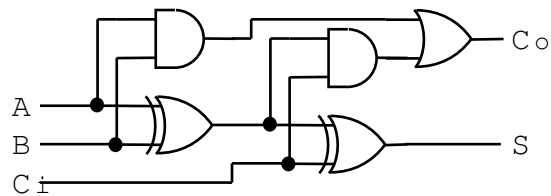
(実習) 実習装置を用いて図 2.7(c) の半加算器を組み立て、1 ビットの加算演算の動作を確認しなさい。

(実習) 実習装置を用いて図 2.8(b) の全加算器を組み立て、1 ビットの加算演算の動作を確認しなさい。

(実習) 図 2.9 を参考に、2 ビットの加算回路の図を書きなさい。また、この回路を組み立て、2 進数の加算が行われる様子を確認しなさい。

A	B	C <sub>i</sub>	S	C <sub>o</sub>
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

(a) 全加算器の真理値表



(b) 全加算器の回路図

図 2.8: 全加算器

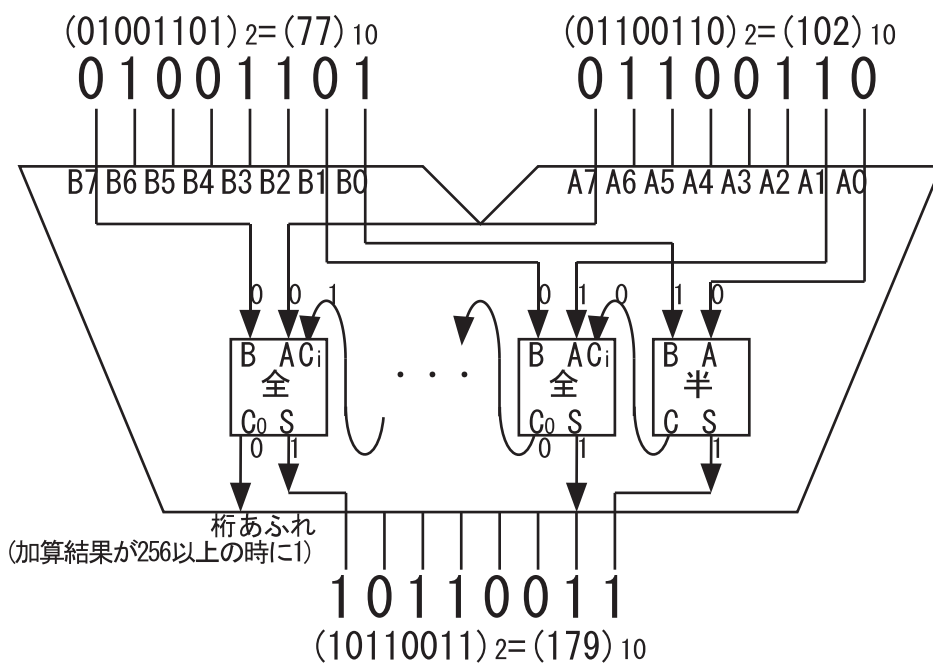


図 2.9: 8 ビットの加算回路

### エンコーダ

8本の信号の中のいずれかひとつが1で他は0であるとします。この信号を入力し、3ビットで符号化したいとします。この真理値表および、回路図は図 2.10 のようになります。

### デコーダ

3ビットの符号化された信号を入力し、8個の中のいずれが選択されているか復号したいとします。この真理値表および、回路図は図 2.11 のようになります。

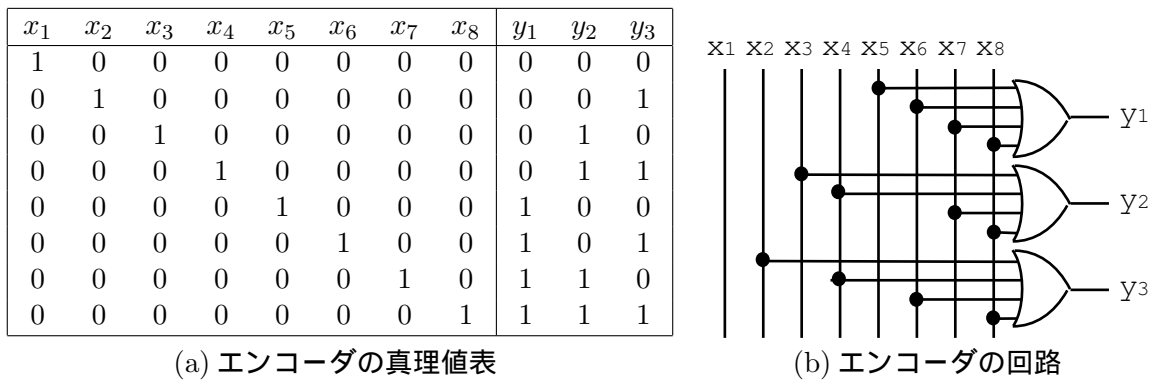


図 2.10: エンコーダ

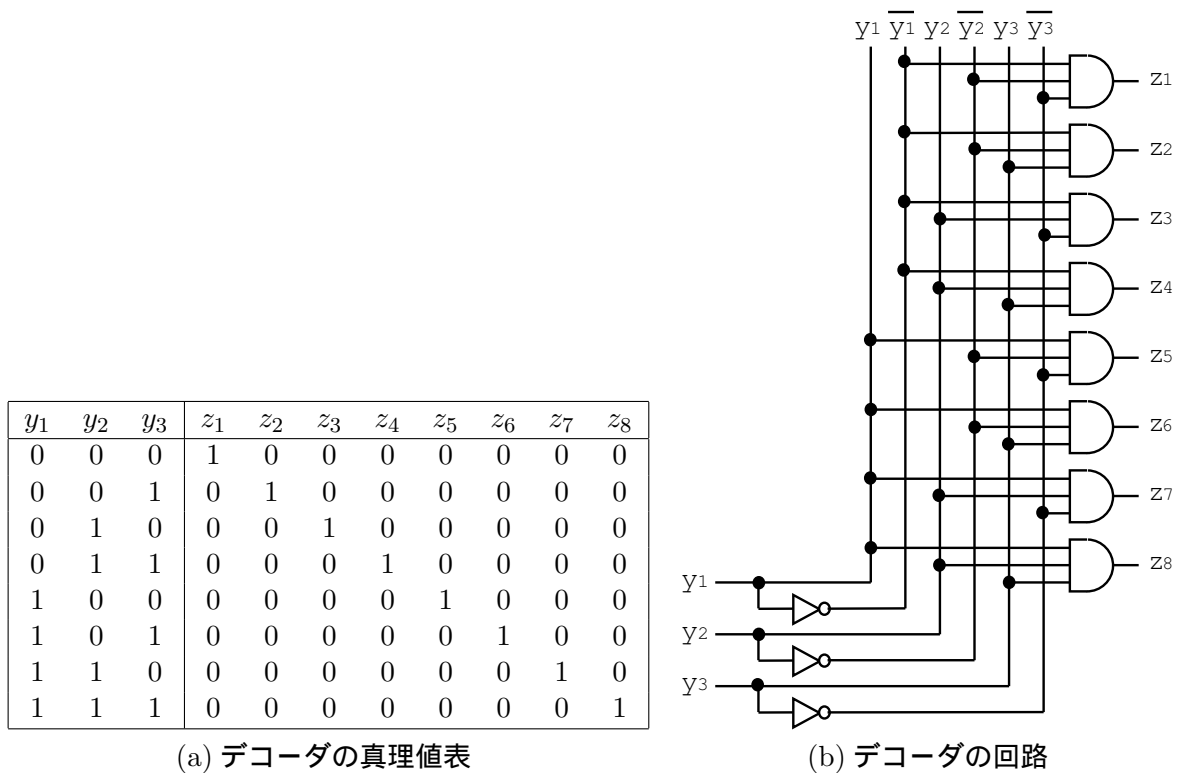


図 2.11: デコーダ

### 7セグメントデコーダ

7セグメントLEDは、7個のセグメントのどこを発光させるかで0~9の数字を表現する素子で、電卓などの数字表示に用いられます(図2.12)。



図 2.12: 7セグメントLED

ここでは、3ビットで表された数を数字で表示する回路(7セグメントデコーダ)を実現したいと

します。この真理値表は以下のようになります。

$y_1$	$y_2$	$y_3$	$a$	$b$	$c$	$d$	$e$	$f$	$g$	表示
0	0	0	1	1	1	1	1	1	0	0
0	0	1	0	1	1	0	0	0	0	1
0	1	0	1	1	0	1	1	0	1	2
0	1	1	1	1	1	1	0	0	1	3
1	0	0	0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	0	1	1	5
1	1	0	1	0	1	1	1	1	1	6
1	1	1	1	1	1	0	0	1	0	7

a~dの各々についてカルノー図を描き、簡単化した式を導くと図 2.13 のようになります。

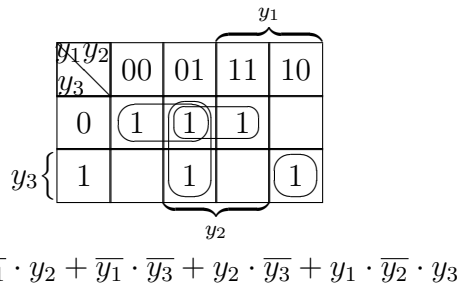
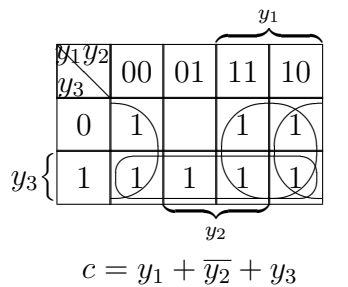
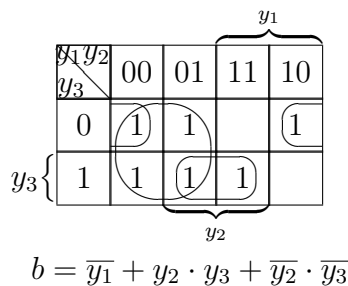
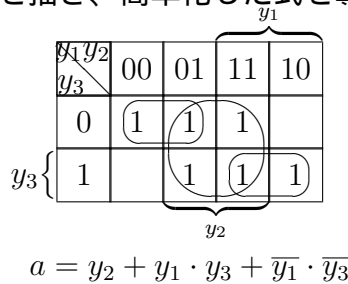


図 2.13: 7セグメントデコーダの簡単化 (a~d)

(練習) 図 2.13 を参考に、e~g について簡単化した式を求めなさい。また、e~g について図 2.14(a) を補いなさい。

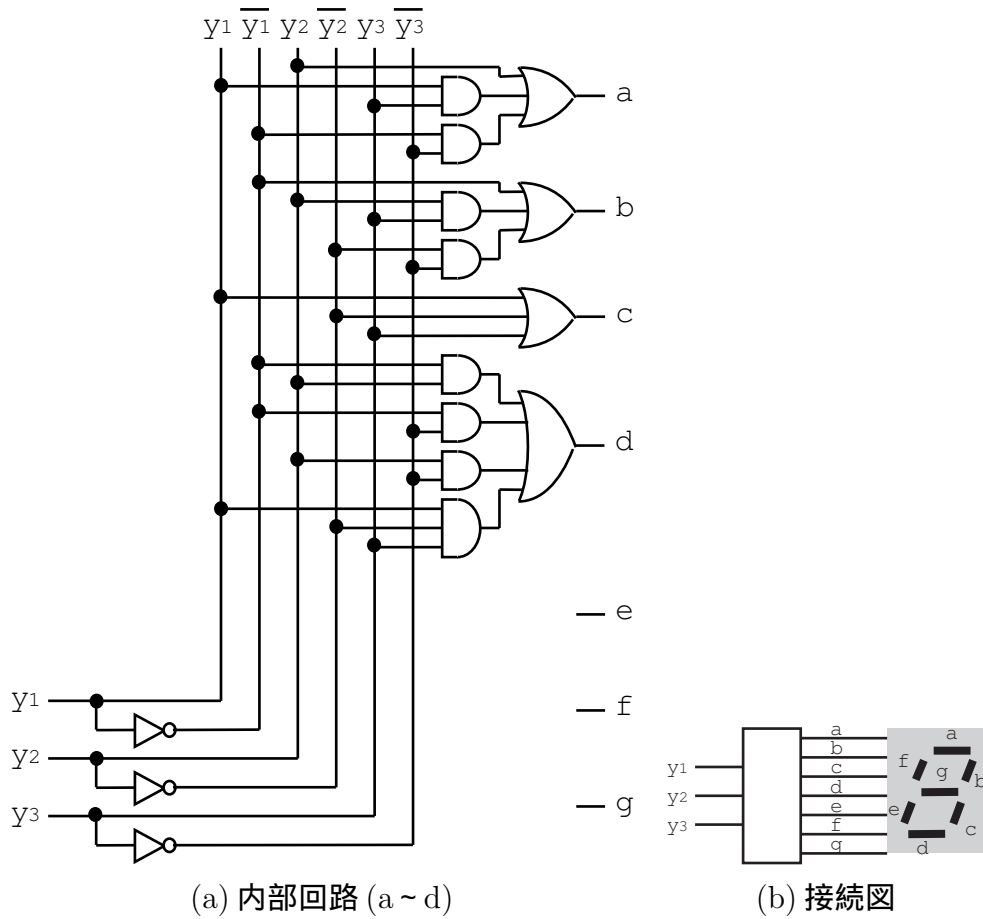


図 2.14: 3 ビット-7 セグメントデコーダ

(実習) 2 ビットで表された数を数字で表示する回路 (7 セグメントデコーダ) を設計し、動作を確認しなさい。

- (1) 入力  $y_1, y_2$  と出力 a ~ g の関係を真理値表で表す。
- (2) a ~ g を  $y_1, y_2$  の式として表す。
- (3) 回路図を描く。
- (4) 実習装置で動作を確認する。

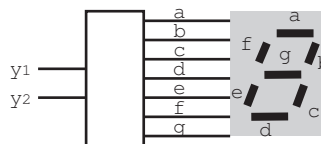


図 2.15: 2 ビット-7 セグメントデコーダ

## 第3章 順序論理回路

組み合わせ論理回路は、入力の状態で出力が決まりました。順序論理回路は、過去の入力履歴を記憶する内部メモリと現在の入力の状態で出力が決まります。この章では、状態を記憶するフリップフロップ、それをを用いた順序論理回路の例、一般的な順序論理回路の設計方法について学びます。

### 3.1 フリップフロップ

1ビットのメモリ(記憶素子)がフリップフロップです。 $n$ ビットのメモリを用いると $2^n$ 個の状態を表すことができます。

以下に示すように、フリップフロップにはいくつかの種類があり、動作に特徴があります。

#### RS フリップフロップ

図3.1(a)はRSフリップフロップで、(b)のように実現できます。(c)は典型的な動作を示すタイムチャート、(d)は真理値表です。表中の $R_n = 1, S_n = 1$ の欄は、このように用いることはないことを表しています。

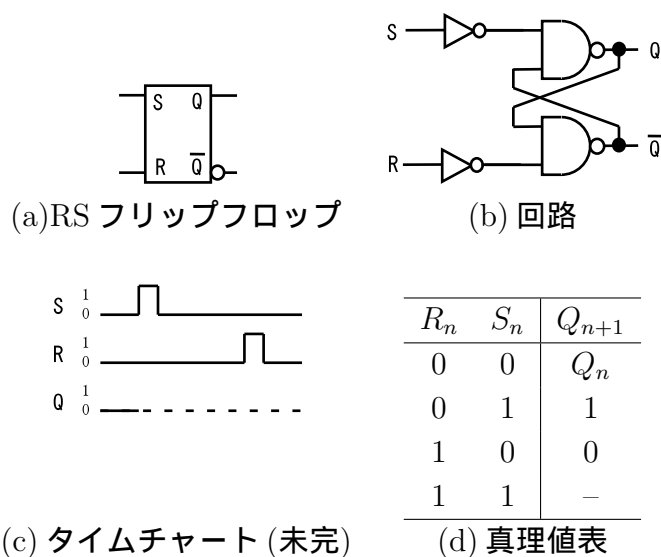
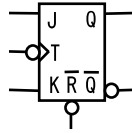


図 3.1: RS フリップフロップ

(実習) 実習装置を用いて図3.1(b)のRSフリップフロップを組み立て、動作を確認しなさい。また、同図(c)のタイムチャートを調べ、「記憶」の特徴を生かした用途を考えてみなさい。

### JK フリップフロップ

図3.2(a)はJK フリップフロップで、 $Q$ はクロック信号  $T$  に同期して変化します。どのように変化するかは、 $J$ と $K$ の状態により異なります。真理値表で表せば図3.2(b)のようになります。

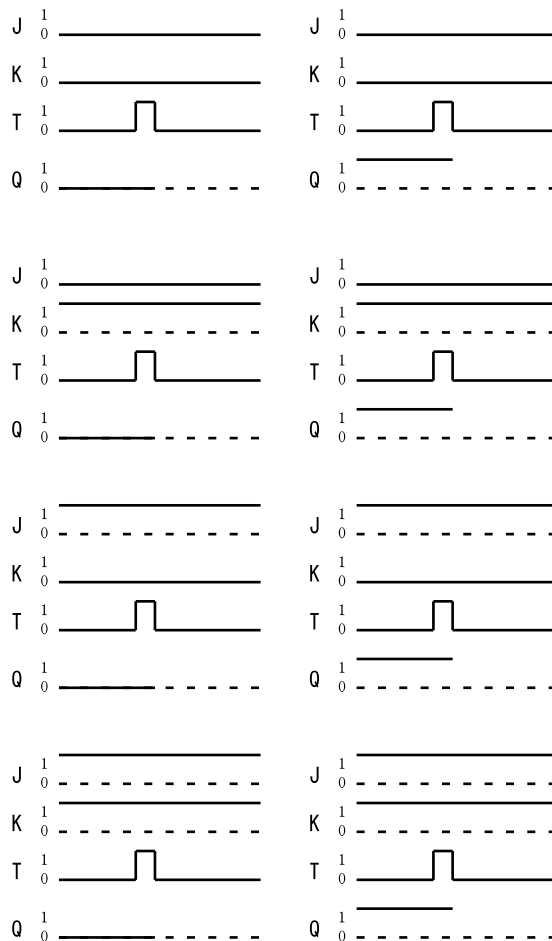


J	K	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$\overline{Q_n}$

(a)JK フリップフロップ (b) 真理値表

図 3.2: JK フリップフロップ

(実習) 実習装置を用い、以下の方法で JK フリップフロップの動作を調べなさい。



J	K	$Q_n$	$Q_{n+1}$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

この真理値表が図 3.2(b) と同じ内容であることを確かめなさい。

### T フリップフロップ

図 3.3(a) は T フリップフロップで、(b) のように JK フリップフロップを用いて実現できます。(c) は典型的な動作を示すタイムチャート、(d) は真理値表です。

クロックに同期して出力 Q が反転する。周期信号 (矩形波) を入力した場合、周期は 2 倍 (周波数は 2 分の 1) になります。このため、T フリップフロップは分周器とも呼ばれます。

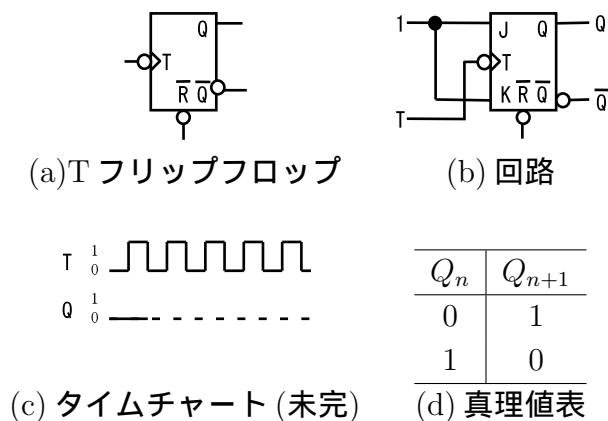


図 3.3: T フリップフロップ

(実習) 実習装置を用いて図 3.3(b) の T フリップフロップを組み立て、動作を確認しなさい。また、同図 (c) のタイムチャートを調べ、「記憶」の特徴を生かした用途を考えてみなさい。

### D フリップフロップ

図 3.4(a) は D フリップフロップで、(b) のように JK フリップフロップを用いて実現できる。(c) は典型的な動作を示すタイムチャート、(d) は真理値表である。

クロックに同期して入力 D の状態を保持 (記憶) する。

(実習) 実習装置を用いて図 3.4(b) の D フリップフロップを組み立て、動作を確認しなさい。また、同図 (c) のタイムチャートを調べ、「記憶」の特徴を生かした用途を考えてみなさい。

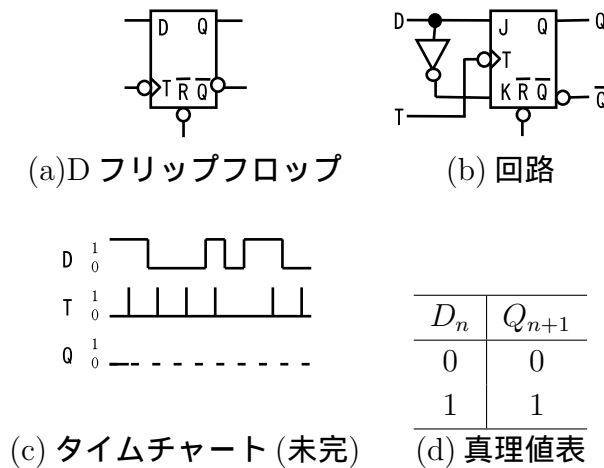


図 3.4: D フリップフロップ

## 3.2 順序論理回路の例

### カウンタ

T フリップフロップを  $n$  個縦続接続すれば、 $2^n$  進カウンタを構成できる。10 進カウンタのように  $2^n$  進以外のカウンタの場合は、所定の数に達した時にカウントをリセットするなどの回路を追加する。

(実習) 実習装置を用いて図 3.5(a) の 4 進カウンタを組み立て、動作を確認しなさい。入力に 10Hz の周期信号 (矩形波) を加え、入力 T と出力  $Q_A$ 、 $Q_B$  の関係をオシロスコープで観測し、タイムチャートで表しなさい。また、このカウンタに図 2.15 の 2 ビット-7 セグメントデコーダを接続し、LED の表示を確かめなさい。

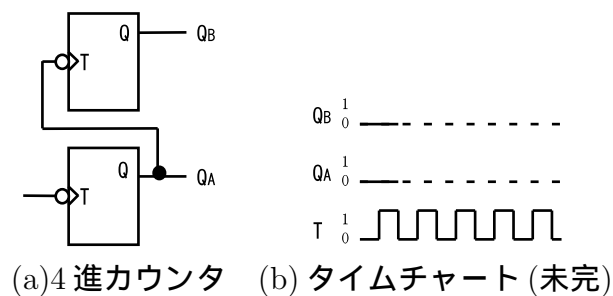


図 3.5: カウンタ

### データラッチとバス構造

電卓のように、多数桁の 7 セグメント LED に別々の数字を表示させたいとしましょう。

図 3.6 は、データラッチとバス構造を用いて複数桁 (図は 2 桁) の LED に任意の数字を表示させる回路の構成例です。各桁の LED に対応させて、表示内容を記憶するデータラッチを用意します。この場合、データラッチは a ~ g の 7 ビットのデータを記憶するために、7 個の D フリップフロップを並べたものです。データラッチはレジスタとして用いられます。

制御回路は、まず LED1 に表示すべきデータを  $a \sim g$  に与え、T1 にタイミングパルス信号を与えます。次に、LED2 に表示すべきデータを  $a \sim g$  に与え、T2 にタイミングパルス信号を与えます。D フリップフロップの性質から、T にタイミングパルス信号が与えられない限り Q が変化することはありません。 $a \sim g$  の信号線は異なる目的で時分割で共同利用され、データバスと呼ばれます。

LED の数が 10 個の場合を考えてみましょう。この構造では、データバス 7 本とタイミング信号線 10 本の計 17 本の信号線を接続すれば、個々の LED に任意の数字を表示させることができます。一方、もしも各 LED に個別に表示データを伝えるとなれば、 $7 \times 10 = 70$  本の信号線を接続する必要があります。このことからバス構造が合理的であることがわかります。

コンピュータの内部にも、データやアドレスにバス構造が用いられています。

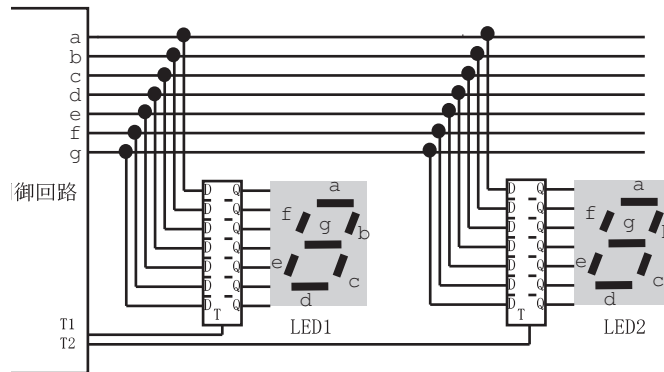


図 3.6: データラッチとバス

## シフトレジスタ

パソコンのキーボードには 100 個以上のキー (スイッチ) があり、どのキーが押されたかがパソコンに伝えられます。そのための最も単純な方法は、個々のキーからパソコンに 100 数十本の信号線を配線する方法です。しかし、実際のキーボードの信号線は数本しかありません。信号線の数減らすために、エンコード (符号化) とシリアル信号 (時系列信号) 通信が用いられています。

まず、図 2.10 のようなエンコーダを用いると、 $2^n$  個の中のどのキーが押されたかを  $n$  ビットで表すことができます。128 個の中のどのキーが押されたかは 7 ビットで表すことができます。

符号化された信号はシフトレジスタを用いてパラレル-シリアル変換され、時系列のシリアル信号としてパソコンに伝えられます。パソコンの中では、この信号を受け取り、シフトレジスタを用いてシリアル-パラレル変換してパラレル信号を復元します。

図 3.7 はこの様子を示したもので、特にパソコン中のシリアルパラレル変換の部分を詳しく描いています。

(実習) 実習装置を用いて図 3.8(a) の 4 ビットシフトレジスタを組み立て、動作を確認しなさい。(b) のタイムチャートを完成させ、シリアル-パラレル変換の仕組みについて考えなさい。

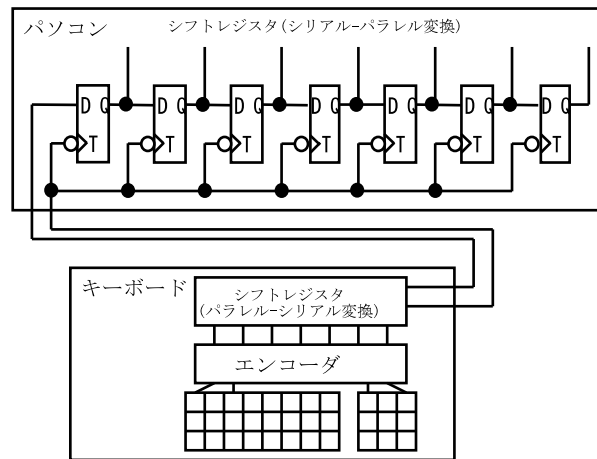
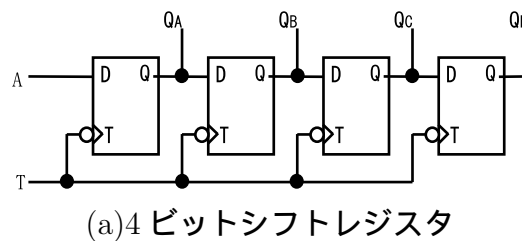
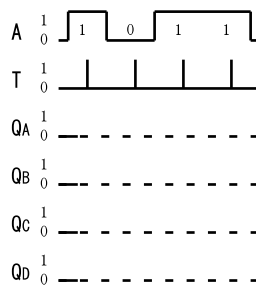


図 3.7: シフトレジスタを用いたシリアル-パラレル変換とキーボードへの適用例



(a) 4 ビットシフトレジスタ



(b) タイムチャート (未完)

図 3.8: シフトレジスタによるシリアル-パラレル変換

### 3.3 順序論理回路の設計

#### 順序論理回路の構成

図 3.9(a) は順序論理回路の構成を示したものです。組み合わせ論理回路では入力のみで出力が決まるが、順序論理回路では入力と内部状態を記憶する状態メモリにより出力が決まります。また、次の内部状態も、入力と現在の内部状態で決まります。 $n$  ビットの状態メモリを用いると  $2^n$  の状態を表すことができます。

図 3.9(a) を少し詳しく描いたものが (b)(c) です。(c) のムーア型では出力は状態メモリのみで決まるため、(b) のミーリ型に比べて出力は安定しますが、回路は複雑になります。

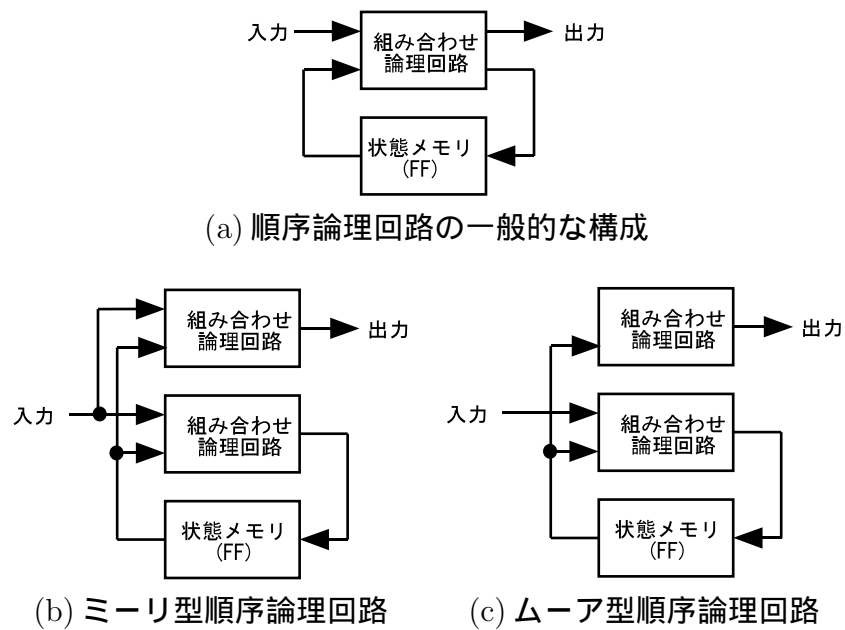


図 3.9: 順序論理回路

### 自動販売機の設計例

以下の仕様の自動販売機をミーリ型順序論理回路で設計してみましょう。

- 商品の値段...150 円
- 使える硬貨...50 円玉と 100 円玉 (ただし同時投入はなし)
- おつりを考慮すること

この仕様に従い、自動販売機の状態遷移図を描くと図 3.10 のようになります。

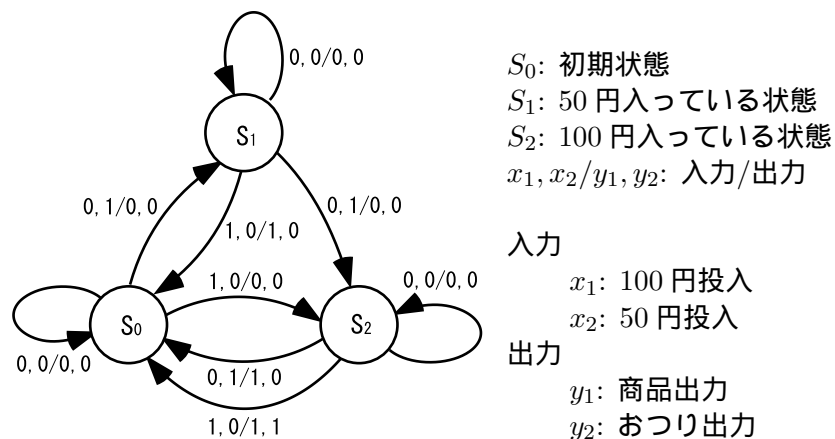


図 3.10: 自動販売機の状態遷移図

図 3.10 を表で表したものが表 3.1(a) である (状態遷移表)。

状態の数は 3 つなので、2bit で表現することができます。ここでは、状態メモリとして 2 個の D フリップフロップを用いることにします。

フリップフロップの出力  $Q_1, Q_2$  の状態割り当ては、ここでは

$$S_0 = 00, S_1 = 01, S_2 = 10(\text{ただし } S_i = Q_1Q_2)$$

と決めることにします。あとは、図 3.9 において表 3.1(a) の要件を満たす組み合わせ論理回路を設計する問題を考えればよいこととなります。組み合わせ論理回路を構成しなければならない要素としては、出力  $y_1, y_2$  および次の状態  $D_1, D_2$  の 4 つありますが、これらが明示されるように表 3.1(a) を書き直すと (b) のようになります。ただし、表中は「次の状態  $D_1D_2$ /出力  $y_1y_2$ 」を示しています。

表 3.1: 自動販売機の状態遷移表

(a) 状態遷移表					(b) 状態割り当てをした状態遷移表				
現在の状態	入力 $x_1x_2$				$Q_1Q_2$	$x_1x_2$			
	00	01	10	11		00	01	10	11
$S_0$	$S_0/0,0$	$S_1/0,0$	$S_2/0,0$	—	00	00/00	01/00	10/00	—
$S_1$	$S_1/0,0$	$S_2/0,0$	$S_0/1,0$	—	01	01/00	10/00	00/10	—
$S_2$	$S_2/0,0$	$S_0/1,0$	$S_0/1,1$	—	10	10/00	00/10	00/11	—

これを、真理値表として書き直したものが表 3.2 です。

ここで、 $Q_1Q_2 = 11$  になることはありませんし (この状態は存在しない)、 $x_1x_2 = 11$  という入力加わることもありません (お金は同時に投入されない)。従って、現れることがない入力に対しては出力は 0 になっても 1 になってもよいこととなります。これが、表中の「×印 (Don't Care Event)」です。都合が良いように 0 または 1 を仮定して構いません。

この真理値表から、各々の出力について主加法標準形を用いて論理式を得ると次のようになります。×印の箇所はすべて 0 を仮定しています。

$$D_1 = \overline{Q_1} \cdot \overline{Q_2} \cdot x_1 \cdot \overline{x_2} + \overline{Q_1} \cdot Q_2 \cdot \overline{x_1} \cdot x_2 + Q_1 \cdot \overline{Q_2} \cdot \overline{x_1} \cdot \overline{x_2}$$

$$D_2 = \overline{Q_1} \cdot \overline{Q_2} \cdot \overline{x_1} \cdot x_2 + \overline{Q_1} \cdot Q_2 \cdot \overline{x_1} \cdot \overline{x_2}$$

$$y_1 = \overline{Q_1} \cdot Q_2 \cdot x_1 \cdot \overline{x_2} + Q_1 \cdot \overline{Q_2} \cdot \overline{x_1} \cdot x_2 + Q_1 \cdot \overline{Q_2} \cdot x_1 \cdot \overline{x_2}$$

$$y_2 = Q_1 \cdot \overline{Q_2} \cdot x_1 \cdot \overline{x_2}$$

この式に従って論理回路を構成しても目的とする機能は得られますが、3.11 に示すように回路の簡単化を行うことができます。×印の箇所のいくつかは、簡単化の都合に合わせて 1 が仮定されています。

このように以下の簡単化された式を得ることができます。

$$D_1 = Q_2 \cdot x_2 + \overline{Q_1} \cdot \overline{Q_2} \cdot x_1 + Q_1 \cdot \overline{x_1} \cdot \overline{x_2}$$

$$D_2 = \overline{Q_1} \cdot \overline{Q_2} \cdot x_2 + Q_2 \cdot \overline{x_1} \cdot \overline{x_2}$$

$$y_1 = Q_1 \cdot x_2 + Q_2 \cdot x_1 + Q_1 \cdot x_1$$

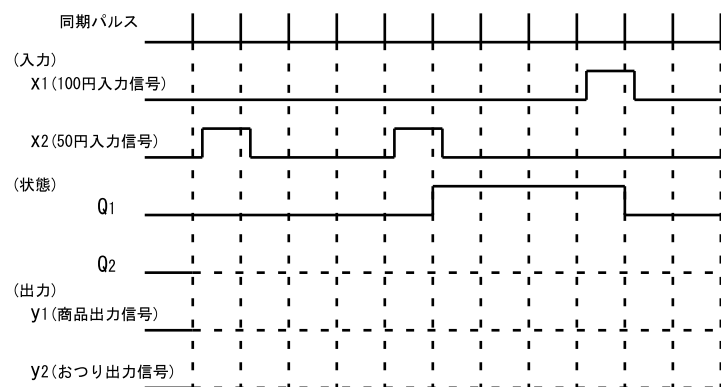
$$y_2 = Q_1 \cdot x_1$$

このように得られた組み合わせ論理回路、および状態メモリを回路図で示したものが図 3.12 です。

表 3.2: 自動販売機の真理値表

$Q_1$	$Q_2$	$x_1$	$x_2$	$D_1$	$D_2$	$y_1$	$y_2$
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	×	×	×	×
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	0	1	0
0	1	1	1	×	×	×	×
1	0	0	0	1	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	0	1	1
1	0	1	1	×	×	×	×
1	1	0	0	×	×	×	×
1	1	0	1	×	×	×	×
1	1	1	0	×	×	×	×
1	1	1	1	×	×	×	×

(実習) 実習装置を用いて図 3.12 の自動販売機を組み立て、動作を確認しなさい。以下のタイムチャートを完成させなさい。なお、可能ならば表 3.1 のとは異なる状態割り当て(つまり  $S_i$  に割り当てる  $Q_1Q_2$  を変える)にして以降の設計をしておし、実習装置で動作を確認しなさい。



(練習) 図 3.10 の自動販売機の状態遷移図はミューア型順序論理回路で設計する場合の図だが、ムーア型順序論理回路で設計する場合の状態遷移図はどのようになるか考えなさい。

$x_1x_2$ $Q_1Q_2$	00	01	11	10
00	0	0	×	1
01	0	1	×	0
11	×	×	×	×
10	1	0	×	0

$$D_1 = Q_2 \cdot x_2 + \overline{Q_1} \cdot \overline{Q_2} \cdot x_1 + Q_1 \cdot \overline{x_1} \cdot \overline{x_2}$$

(1)  $D_1$  の簡単化

$x_1x_2$ $Q_1Q_2$	00	01	11	10
00	0	1	×	0
01	1	0	×	0
11	×	×	×	×
10	0	0	×	0

$$D_2 = \overline{Q_1} \cdot \overline{Q_2} \cdot x_2 + Q_2 \cdot \overline{x_1} \cdot \overline{x_2}$$

(2)  $D_2$  の簡単化

$x_1x_2$ $Q_1Q_2$	00	01	11	10
00	0	0	×	0
01	0	0	×	1
11	×	×	×	×
10	0	1	×	1

$$y_1 = Q_1 \cdot x_2 + Q_2 \cdot x_1 + Q_1 \cdot x_1$$

(3)  $y_1$  の簡単化

$x_1x_2$ $Q_1Q_2$	00	01	11	10
00	0	0	×	0
01	0	0	×	0
11	×	×	×	×
10	0	0	×	1

$$y_2 = Q_1 \cdot x_1$$

(4)  $y_2$  の簡単化

図 3.11: 自動販売機のカルノー図による簡単化

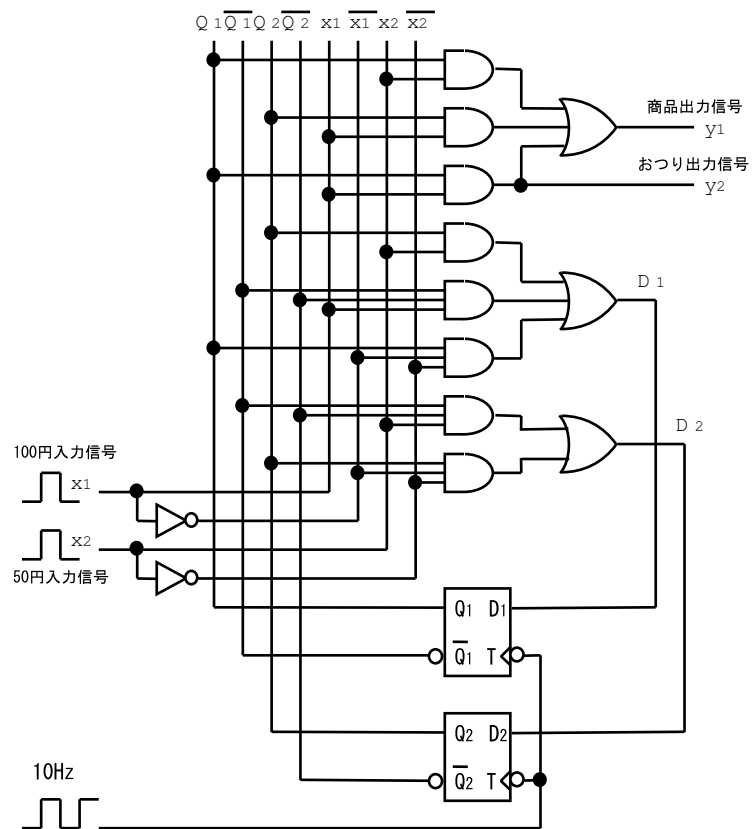


図 3.12: 自動販売機の回路図



# 付録 実習装置

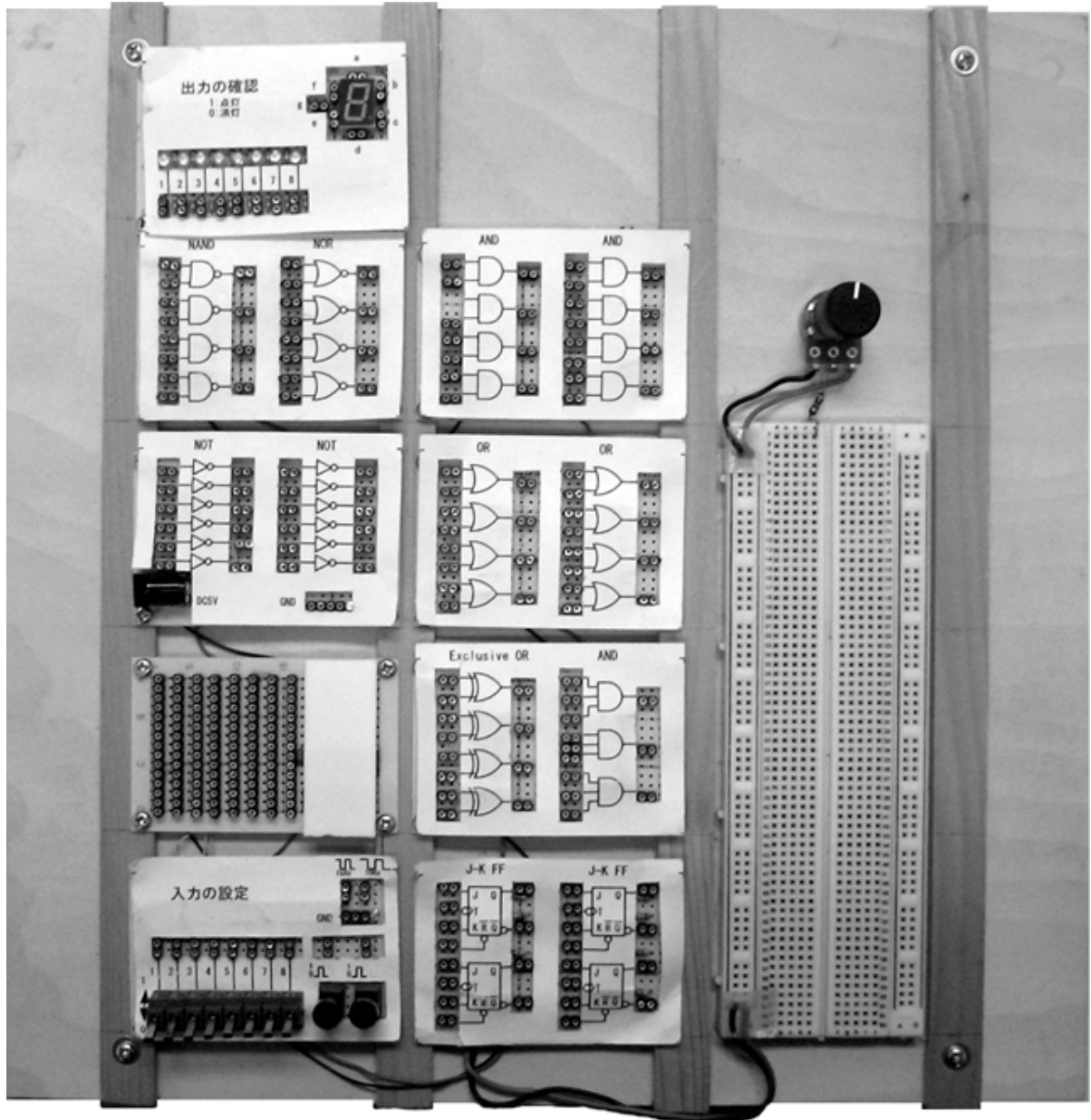


図 A1 実習装置の外観

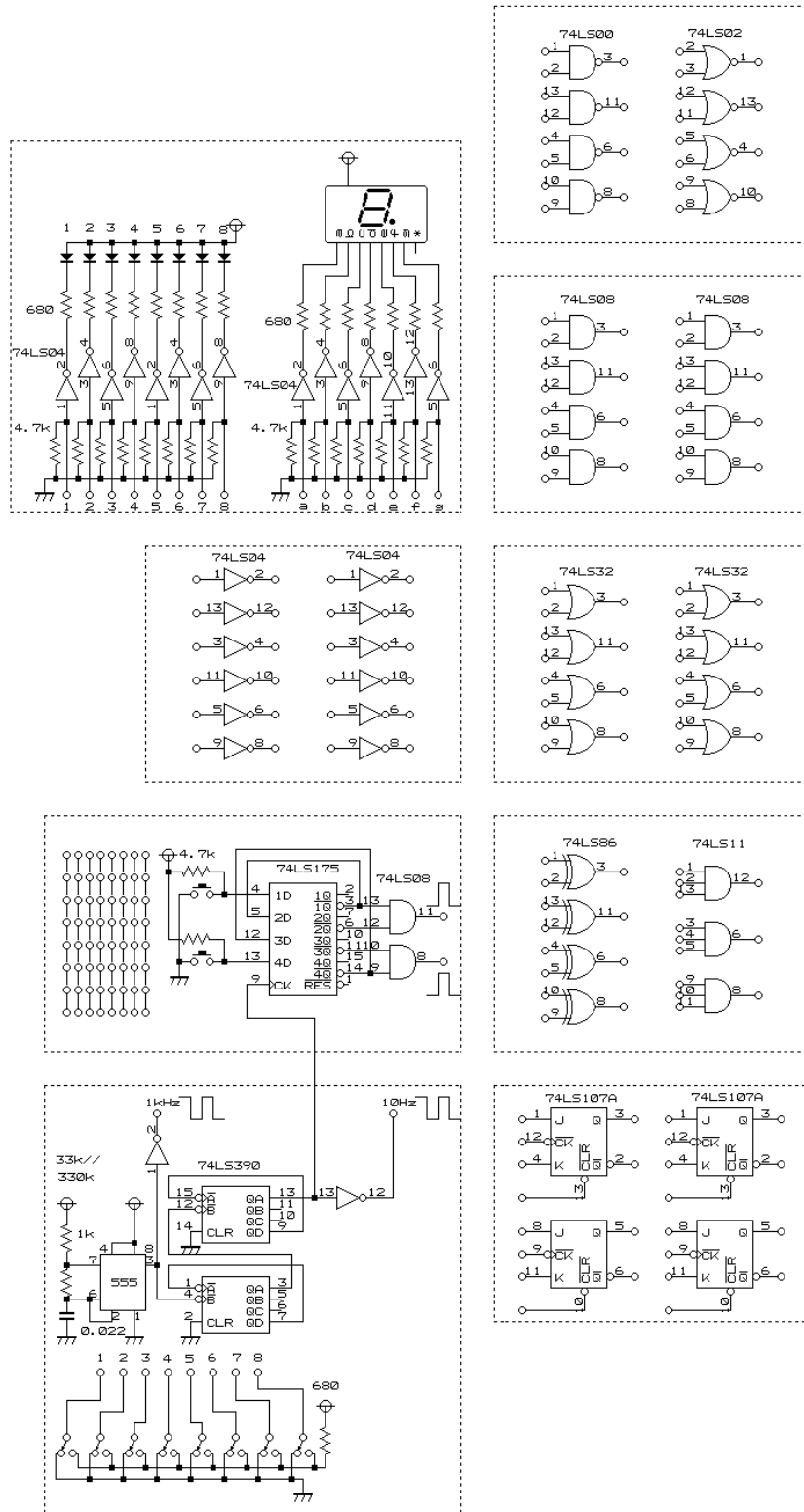


図 A2 実習装置の回路図