

### 7.1

長さ  $k$  の識別系列で求めた分割を  $\Pi_k$  とする.  $\Pi_0$  は入力を加えない場合で, 全ての状態を含む一つのブロックからなる分割である. 即ち,

$$\Pi_0 = \{[Q_0, Q_1, Q_2, Q_3, Q_4, Q_5]\}$$

次に, 入力  $x = 0$  を加えると,

$$\Pi_1 = \{[Q_0, Q_1, Q_2, Q_4, Q_5], [Q_3]\}$$

となる. 次に,  $x = 10$  と入力を加えた場合,

$$\Pi_2 = \{[Q_0, Q_2, Q_4], [Q_1, Q_5], [Q_3]\}$$

となる. さらに  $\Pi_2$  において入力を加えるが,  $\Pi_3 = \Pi_2$  となる. よって, 状態数を最小化すると表 1 のようになる.

表 1: 最小化した状態表

	$x = 0$	$x = 1$
$Q_0$	$Q_0, 0$	$Q_1, 1$
$Q_1$	$Q_0, 0$	$Q_3, 1$
$Q_3$	$Q_0, 1$	$Q_0, 1$

### 7.2

$\Pi_0 = \{[Q_0, Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7]\}$  に入力  $x = 0, 1$  を加えた場合,  $Pi_0$  を出力に応じて以下の様に分割できる.

$$\Pi_1 = \{[Q_0, Q_4, Q_7], [Q_1, Q_2], [Q_3, Q_5, Q_6]\}$$

次に, 状態  $Q_0, Q_4, Q_7$  に入力  $x = 01$  を加えた場合, 出力系列は  $01, 01, 00$  となるので,  $[Q_0, Q_4]$  と  $[Q_7]$  は識別可能である. しかし, 他の入力  $x = 00, 10, 11$  を加えても, 状態  $Q_0, Q_4$  を細分できない. 次に, 状態  $Q_1, Q_2$  に入力  $x = 01$  を加えると, 出力系列はどれも  $00, 00$  となるが,  $[Q_4]$  と  $[Q_7]$  が識別可能なので,  $[Q_1]$  と  $[Q_2]$  も識別可能である. また, 状態  $Q_3, Q_5, Q_6$  に入力  $x = 11$  を加えた場合, 出力系列はそれぞれ  $01, 00, 01$  となるので,  $[Q_3, Q_6]$  と  $[Q_5]$  は識別可能である. 他の入力  $x = 00, 01, 10$  を加えても, 状態  $Q_3, Q_6$  を細分できない. したがって,

$$\Pi_2 = \{[Q_0, Q_4], [Q_1], [Q_2], [Q_3, Q_6], [Q_5], [Q_7]\}$$

となる. したがって一例として  $\{[Q_0], [Q_1], [Q_2], [Q_3], [Q_5], [Q_7]\}$  と状態数を最小化できる.

### 7.3

#### (割当 1)

割当 1 の遷移表を表 2 に示す. 割当 1 をもとにした状態遷移関数のカルノー図を図 1 に示す. これより,

$$D_1 = y_1 x \vee \bar{y}_1 \bar{x}$$

$$D_2 = y_2 \bar{x} \vee \bar{y}_2 x$$

これより, 回路を図 2 に示す.

#### (割当 2)

割当 2 の遷移表を表 3 に示す. 割当 2 をもとにした状態遷移関数のカルノー図を図 3 に示す. これ

表 2: 割当 1 の遷移表

	$x = 0$	$x = 1$
$y_1 y_2$	$y_1' y_2', z$	$y_1' y_2', z$
00	10,0	01,1
01	11,1	00,0
10	00,0	11,1
11	01,1	10,0

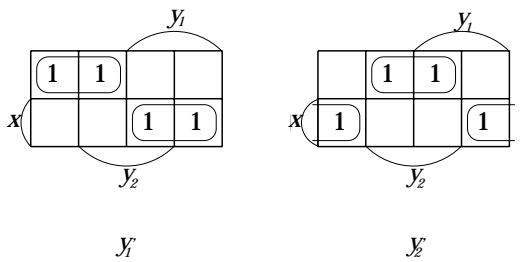


図 1: 割当 1 の状態遷移関数のカルノー図

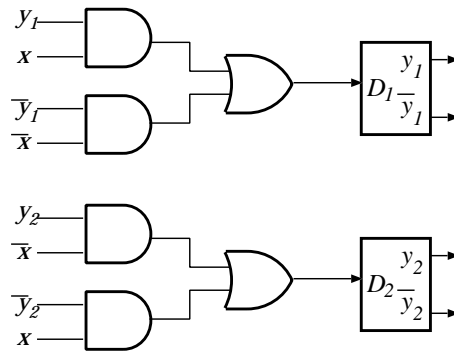


図 2: 割当 1 の回路

より,

$$D_1 = y_1 x \vee \bar{y}_1 \bar{x}$$

$$D_2 = \bar{y}_2 x$$

を得る. これより, 回路を図 4 に示す.

これより, 割当 2 を用いたほうが回路が簡単になる.

#### 7.4

$n$  個の状態変数で  $2^n$  個の状態を表現できる. よって  $R$  個の状態を表現するために,  $n$  個の状態変数を用いると, 異なる状態割当は

$$\frac{2^n!}{(2^n - R)!}$$

表 3: 割当 2 の遷移表

	$x = 0$	$x = 1$
$y_1 y_2$	$y_1' y_2', z$	$y_1' y_2', z$
00	11,0	01,1
01	10,1	00,0
11	00,0	10,1
10	01,1	11,0

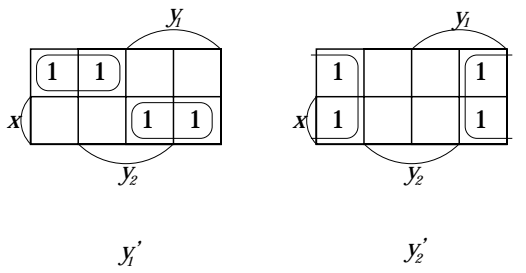


図 3: 割当 2 の状態遷移関数のカルノー図

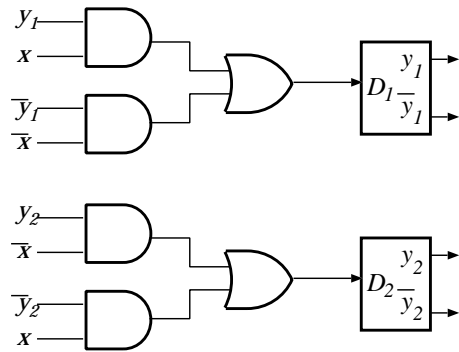


図 4: 割当 2 の回路

通り存在する. ここで, 各  $D$ -FF の出力は AND-OR 二段回路の入力へと接続されるが, これらの出力のラベル付けを入れ換えても AND-OR 二段回路の最小ゲート数は変わらないので

$$\frac{2^n!}{(2^n - R)!n!}$$

個の異なる状態割当を考えればよい. また, FF の出力の否定・入力変数の否定を行っても AND-OR 二段回路の最小ゲート数は変わらないので

$$\frac{2^n!}{(2^n - R)!n!2^n} = \frac{(2^n - 1)!}{(2^n - R)!n!}$$

個の異なる状態割当を考えれば十分である.

### 7.5

$m$  個の肯定リテラルを持つ最小項  $c_m \cdot c_{\bar{m}}$  を考える. 但し,  $c_m$  は肯定リテラルの積であり,  $c_{\bar{m}}$  は否定リテラルの積である.

m-out-of-n 符号の場合, 入力の 1 のビット数が  $m$  個より少ない時はドント・ケアになる. よって最小項  $c_m \cdot c_{\bar{m}}$  の否定リテラル, すなわち  $c_{\bar{m}}$  のリテラルのうち一つ以上を肯定リテラルに変えたものに対する関数値はドント・ケアとなる. ここで,  $c_m \cdot c_{\bar{m}}$  の否定リテラルのうち一つ以上を肯定リテラルに変えることで得られる全ての最小項の論理和は  $c_m$  となる. つまり,  $c_m \cdot c_{\bar{m}}$  はドント・ケアを用いることで積項  $c_m$  に包含させることができる. よって, 否定リテラルが消え, インバータが不要になる.