AN APPLICATION OF PSEUDO-BOOLEAN PROGRAMMING TO THE STATE ASSIGNMENT OF SEQUENTIAL CIRCUITS WITH HIGH RELIABILITY

YASUYOSHI INAGAKI and KAZUE SUGINO

Department of Electronics Engineering

(Received October 30, 1965)

1. Introduction

The state assignment is an important problem in the practice of synthesizing sequential circuits, because it has a marked influence on the complexity and reliability of the circuits. It has been already shown in our previous paper[1] that the state assignment problems of reliable sequential circuits can be solved by the use of the assignment-iteration method. In this paper, it is first demonstrated that the assignment-improving routine of this method can be solved by the pseudo-Boolean programming, an algorithm recently found by P. L Ivănescu[2]. As shown in the succeeding section, the estimation of the elements of the fundamental matrix allows us to reach the optimal state assignment after only one iteration of the assignment-iteration method when the error probabilities of memory cells are small enough. Then, in these cases, it follows that the optimal state assignment becomes practically attainable even by the hand computations of the pseudo-Boolean programming for the problems of moderate size.

2. Assignment-Iteration Method

In order to consider the state assignment problem, the following assumptions are made;

(1) The sequential circuit under consideration consists of one combinational logical circuit g and r memory cells $\sigma^{(1)}, \sigma^{(2)}, \sigma^{(3)}, \ldots, \sigma^{(r)}$, as shown in Fig. 1.

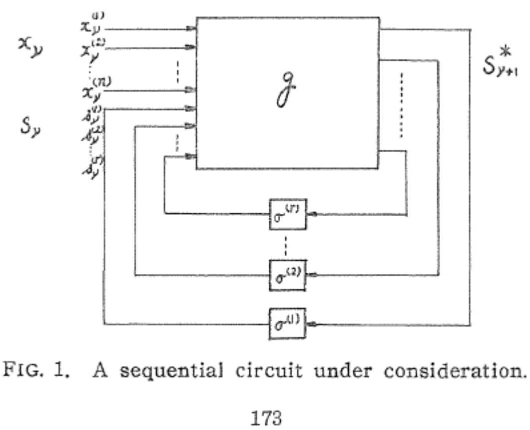

FIG. 1. A sequential circuit under consideration.

(2) The error probability of the combinational circuit is small enough to be disregarded.

(3) Errors occur only when the state of memory cells changes from 0 to 1 and 1 to 0. Let these error probabilities be α and β, respectively, where their second order terms are so small as to be negligible.

(4) Inputs are statistically mutually independent.

Under these conditions, the transition process of the internal state can be represented by the absorbing Markov chain with an absorbing state ⓪ which is the error state[1]. Let p_{ij} be the transition probability $P\{S_{\nu+1}=ⓙ \mid S_\nu=ⓘ\}$, then we get

$$\left.\begin{aligned} p_{ij} &= \rho_{ij}(1-\alpha b_{tu}-\beta\hat{b}_{tu})\ ;\ i,\ j=1,\ 2,\ \ldots,\ M \\ p_{io} &= \sum_{j=1}^{M}\rho_{ij}(\alpha b_{tu}+\beta\hat{b}_{tu})\ ;\ i=1,\ 2,\ \ldots,\ M \\ p_{oj} &= 0,\ p_{oo}=1 \qquad ;\ j=1,\ 2,\ \ldots,\ M \end{aligned}\right\} \tag{1}$$

where

$$\left.\begin{aligned} b_{tu} &= \bar{\sigma}_t \cdot \sigma_u = \sum_{i=1}^{r}\overline{\sigma^{(i)}}\,\sigma^{(i)'} \\ \hat{b}_{tu} &= \sigma_t \cdot \bar{\sigma}_u = \sum_{i=1}^{r}\sigma^{(i)}\overline{\sigma^{(i)'}} \end{aligned}\right\} \tag{2}$$

(the state ⓘ and ⓙ are associated with $\sigma_t=(\sigma^{(1)},\ \sigma^{(2)},\ \ldots,\ \sigma^{(r)})$, $\{t\}_{10}=\{\sigma^{(1)}\sigma^{(2)}\cdots\sigma^{(r)}\}_2$ and $\sigma_u=(\sigma^{(1)'},\ \sigma^{(2)'},\ \ldots,\ \sigma^{(r)'})$, $\{u\}_{10}=\{\sigma^{(1)'}\sigma^{(2)'}\cdots\sigma^{(r)'}\}$, respectively) and

$$\rho_{ij}=\sum_{k=0}^{2^n-1}P\{S^*_{\nu+1}=ⓙ \mid x_\nu=\xi_k,\ S_\nu=ⓘ\}\cdot P\{x_\nu=\xi_k\} \tag{3}$$

(ξ_k represents the k-th input state and $S^*_{\nu+1}$ the correct internal state at time $\nu+1$)

The arrangement of p_{ij}'s in a matrix form yields

$$P=\begin{matrix} ⓪ \\ ① \\ ② \\ \cdot \\ \cdot \\ \cdot \\ Ⓜ \end{matrix}\left(\begin{array}{c|cccc} 1 & 0 & 0 & \cdots & 0 \\ \hline p_{10} & p_{11} & p_{12} & \cdots & p_{1M} \\ p_{20} & p_{21} & p_{22} & \cdots & p_{2M} \\ \cdot & & & & \\ \cdot & & \cdots & p_{ij} & \cdots \\ \cdot & & & & \\ p_{M0} & p_{M1} & p_{M2} & \cdots & p_{MM} \end{array}\right)=\left(\begin{array}{c|c} I & 0 \\ \hline R & Q \end{array}\right) \tag{4}$$

Let us define T_i ($i=1,\ 2,\ \ldots,\ M$) to be the mean value of the time that the state of the sequential circuit stays in a set of the transient states ①, ②, ..., Ⓜ, when the operation of the circuit started in the state ⓘ. Then we obtain the following linear equations.

$$T_i=1+\sum_{j=1}^{M}p_{ij}T_j\ ;\ i=1,\ 2,\ \ldots,\ M. \tag{5}$$

Denoting the probability distribution of the initial state by π_i, we get the mean time to failure;

$$\tau = \sum_{i=1}^{M} \pi_i T_i \tag{6}$$

The state assignment maximizing τ can be obtained by "the assignment-iteration method" which is formulated as follows:

(I) Fundamental Matrix-Determination Operation

Use $p_{ij}(d)$ for an assignment d given by the routine (II) to determine the fundamental matrix

$$N = (N_{ij}) = (I - Q)^{-1} \tag{7}$$

and calculate

$$N_j = \sum_{i=1}^{M} \pi_i N_{ij} \tag{8}$$

and

$$T_i = \sum_{j=1}^{M} N_{ij} \tag{9}$$

(II) Assignment-Improving Routine

Find the assignment d that maximizes

$$y = \sum_{j=1}^{M} \sum_{k=1}^{M} N_j p_{jk}(d) \tag{10}$$

using N_j determined by the operation (I).

This iteration process terminates when the same assignment as the preceeding one appears, and this is the optimal assignment that gives the maximum mean value of the time to failure.[1]

3. Pseudo-Boolean Programming*

Let L_2 be the Boolean Algebra with two elements 0 and 1, its operations "$\vee$", "$\cdot$", "$-$" being defined by

$\vee$	0	1
0	0	1
1	1	1

$\cdot$	0	1
0	0	0
1	0	1

a	0	1
$\bar{a}$	1	0

We notice that

$$a \vee b = a + b - ab \tag{11}$$

and

$$\bar{a} = 1 - a$$

where $+$, $-$ and juxtaposition denote the usual addition, subtraction and multiplication.

A function

* For proofs and details, see reference (2).

$$F : L_2^n \to R$$

is called a pseudo-Boolean function; here L_2^n is the Cartesian product

$$\underbrace{L_2 \times L_2 \times \cdots \times L_2}_{n}$$

while R is the field of the real numbers. It has been proved in reference[2] that any pseudo-Boolean function may be written as a polynominal with real coefficients.

By a problem of pseudo-Boolean programming we mean the determination of all points $(x_1, \ldots, x_n) \in L_2^n$ minimizing a pseudo-Boolean function $F_1(x_1, \ldots, x_n)$.

The following procedure is given for minimizing a pseudo-Boolean function F_1. Let us put

$$F_1 = x_1F_{11} + \bar{x}_1F_{12} + F_{13} = x_1F_{11} + (1 - x_1)F_{12} + F_{13}$$

or,

$$F_1 = x_1g_1 + h_1,$$

where F_{11}, F_{12}, F_{13}, $g_1 = F_{11} - F_{12}$, $h_1 = F_{12} + F_{13}$ are pseudo-Boolean functions of $(x_2, x_3, \ldots, x_n)$.

Now putting

$$x_i^{\alpha_i} = \begin{cases} x_i & \text{if } \alpha_i = 1 \\ \bar{x}_i & \text{if } \alpha_i = 0 \end{cases} \tag{12}$$

and defining the sets of n-tuples

$$M_1 = \{(\alpha_2, \ldots, \alpha_n) \in L_2^{n-1} \mid g_1(\alpha_2, \ldots, \alpha_n) < 0\},$$
$$N_1 = \{(\beta_2, \ldots, \beta_n) \in L_2^{n-1} \mid g_1(\beta_2, \ldots, \beta_n) = 0\},$$

we provide the following expression of x_1;

$$x_1 = \vee_{(\alpha_2, \ldots, \alpha_n) \in M_1} x_2^{\alpha_2} x_3^{\alpha_3} \cdots x_n^{\alpha_n} \vee u_1[\vee_{(\beta_2, \ldots, \beta_n) \in N_1} x_2^{\beta_2} x_3^{\beta_3} \cdots x_n^{\beta_n}], \tag{13.1}$$

where u_1 is an arbitrary parameter in L_2 (*i.e.*, it may be equal to 0 or 1), and the operation

$$\vee_{j \in (k_1, \ldots, k_p)} y_j = y_{k_1} \vee y_{k_2} \vee \cdots \vee y_{k_p}$$

implies the repeated application of the operation defined by Eq. (11) (*e.g.*, $A \vee B = A + B - AB$, $A \vee B \vee C = (A \vee B) \vee C = A + B - AB + C - A(A + B - AB)C = A + B + C - AB - BC - CA + ABC$). In particular case,

$$\vee_{j \in \phi} y_j = 0.$$

From the general expression of x_1

$$x_1 = x_1(u_1, x_2, \ldots, x_n) \tag{13.1'}$$

we use the notation

$$x_1^0 = x_1(0,\ x_2, \ldots, x_n),$$

or, according to Eq. (13.1)

$$x_1^0 = \vee_{(\alpha_2,\cdots,\alpha_n)\in M_1} x_2^{\alpha_2} x_3^{\alpha_3} \cdots x_n^{\alpha_n} \tag{13.1^0}$$

By applying Eqs. (11) and (12) repeatedly to Eq. (13.1^0), the expression of x_1^0 can be transformed into another from, say x_1^+, containing only the usual arithmetical operations $+$, $-$, $\cdot$.

By $F_2(x_2, \ldots, x_n)$ we denote the function $F_1(x_1, x_2, \ldots, x_n)$ in which x_1 is replaced by

$$x_1 = x_1^+(x_2, \ldots, x_n),$$

that is,

$$F_2(x_2, \ldots, x_n) = F_1[x_1^+(x_2, \ldots, x_n),\ x_2, \ldots, x_n].$$

As F_2 is a pseudo-Boolean function we may write

$$F_2(x_2, \ldots, x_n) = x_2 g_2(x_3, \ldots, x_n) + h_2(x_3, \ldots, x_n).$$

Thus we define the sets

$$M_2 = \{(\alpha_3, \ldots, \alpha_n) \in L_2^{n-2} \mid g_2(\alpha_3, \ldots, \alpha_n) < 0\},$$
$$N_2 = \{(\beta_3, \ldots, \beta_n) \in L_2^{n-2} \mid g_2(\beta_3, \ldots, \beta_n) = 0\},$$

in the same way as described above and determine x_2 as

$$x_2 = \vee_{(\alpha_3,\ldots,\alpha_n)\in M_2} x_3^{\alpha_3} x_4^{\alpha_4} \cdots x_n^{\alpha_n} \vee u_2[\vee_{(\beta_3,\ldots,\beta_n)\in N_2} x_3^{\beta_3} x_4^{\beta_4} \cdots x_n^{\beta_n}], \tag{13.2}$$

where u_2 is an arbitrary parameter in L_2.

In this way we put generally

$$x_i = \vee_{(\alpha_{i+1},\ldots,\alpha_n)\in M_i} x_{i+1}^{\alpha_{i+1}} \cdots x_n^{\alpha_n} \vee u_i[\vee_{(\beta_{i+1},\cdots,\beta_n)\in N_i} x_{i+1}^{\beta_{i+1}} \cdots x_n^{\beta_n}], \tag{13.i}$$

and

$$x_n = \begin{cases} 1 & \text{if the constant } g_n < 0 \\ 0 & \text{if the constant } g_n > 0 \\ u_n & \text{if the constant } g_n = 0 \end{cases} \tag{13.n}$$

From Eq. (13.n) we see that

$$x_n = X_n(u_n), \tag{14.n}$$

where $X_n(u_n)$ is equal to 1 or 0 or the free parameter $u_n \in L_2$. From Eq. (13.n) and Eq. (13.n-1)

$$x_{n-1} = x_{n-1}(u_{n-1},\ x_n) = x_{n-1}(u_{n-1},\ X_n(u_n)) = X_{n-1}(u_{n-1},\ u_n) \tag{14.n-1}$$

and analogously

$$x_i = X_i(u_i,\ u_{i+1}, \ldots, u_n) \tag{14.i}$$

$$x_1 = X_1(u_1,\ u_2,\ .\ .\ .\ ,\ u_n) \tag{14.1}$$

It is proved in reference (2) that, for each system of values of parameters $u_1, u_2, \ldots, u_n$, the system (14. n), (14. n-1), ..., (14. i), ..., (14. 1) yields the minimum of F_1, and conversely. The computation procedure obtaining the minimum of any pseudo-Boolean function is clear in the above description.

4. The Solution for the Assignment-Improving Routine Using the Pseudo-Boolean Programming

The state assignment of the sequential circuit is to assign to each internal state the different state of memory cells. We introduce variables $\xi_{jt}(j=1, 2, \ldots, M;\ t=0, 1, \ldots, 2^r-1)$ whose value is 1 when the t-th of 2^r states of r memory cells $\sigma^{(1)}, \sigma^{(2)}, \ldots, \sigma^{(r)}$ is assigned to the j-th of M internal states of the circuit and 0 otherwise, that is,

$$\xi_{jt} = \begin{cases} 1; & \sigma_t \text{ is assigned to } Ⓙ, \\ 0; & \text{otherwise} \end{cases} \tag{15}$$

The set of ξ_{jt} uniquely determine the state assignment d;

$$d = \{\xi_{jt}\};\ \ j=1, 2, \ .\ .\ .\ , M,\ \ t=0, 1, \ .\ .\ .\ , 2^r-1. \tag{16}$$

ξ_{jt}'s must satisfy the following conditions in order to be a possible assignment.

i) To any internal state at least one state of memory cells must be assigned, *i.e.*,

$$\bigvee_{t=0}^{2^r-1} \xi_{jt} = 1;\ \ j=1, 2, \ .\ .\ .\ , M. \tag{17}$$

ii) Any state of memory cells can not be assigned to more than one internal state, *i.e.*,

$$S_{2/M}\langle \xi_{1t},\ \xi_{2t},\ .\ .\ .\ ,\ \xi_{Mt}\rangle = 0;\ \ t=0, 1, \ .\ .\ .\ , 2^r-1, \tag{18}$$

where $S_{2/M}\langle x_1, x_2, \ldots, x_M\rangle$ is a threshold function and equal to 1 only when two or more variables of M take value 1.

Now we represent y given in Eq. (10) in terms of ξ_{jt}. When the states of memory cells σ_t and σ_u are assigned to the internal states Ⓙ and Ⓚ respectively, we get the following equation,

$$p_{jk} = \rho_{jk}(1 - \alpha b_{tu} - \beta \hat{b}_{tu}). \tag{19}$$

Noting that $\xi_{jt}\cdot\xi_{ku}$ is equal to 1 only when Ⓙ and Ⓚ are assigned to σ_t and σ_u respectively and 0 otherwise, we obtain with the help of Eq. (19)

$$p_{jk}(d) = \sum_{t=0}^{2^r-1} \sum_{u=0}^{2^r-1} \rho_{jk}(1 - \alpha b_{tu} - \beta \hat{b}_{tu})\, \xi_{jt}\cdot \xi_{ku}. \tag{20}$$

Thus

$$y = \sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{t=0}^{2^r-1}\sum_{u=0}^{2^r-1} N_j\, \rho_{jk}(1 - \alpha b_{tu} - \beta \hat{b}_{tu})\, \xi_{jt}\cdot \xi_{ku} \tag{21}$$

Since this function y is a pseudo-Boolean function of ξ_{jt}, "the assignment-improving routine" in the assignment-iteration method can be formulated as follows;

"Find $d=\{\xi_{jt}\}$ $(j=1, 2, \ldots, M;\ t=0, 1, \ldots, 2^r-1)$ that maximizes the pseudo-Boolean function y shown in Eq. (21) under the Boolean constraints given by Eqs. (17) and (18)."

Rewriting y as

$$y=\sum_{j=1}^{M}\sum_{k=1}^{M}N_j\rho_{jk}-\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{t=0}^{2^r-1}\sum_{u=0}^{2^r-1}N_j\rho_{jk}(\alpha b_{tu}+\beta\hat{b}_{tu})\xi_{jt}\cdot\xi_{ku}$$

we obtain the following formulation, that is,

"Find d that minimize the pseudo-Boolean function y^*

$$y^*=\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{t=0}^{2^r-1}\sum_{u=0}^{2^r-1}N_j\rho_{jk}(\alpha b_{tu}+\beta\hat{b}_{tu})\xi_{jt}\cdot\xi_{ku} \tag{22}$$

under the Boolean constraints given by Eqs. (17) and (18)."

Application of Eq. (1) to the constraints of Eqs. (17) and (18) yields

$$\sum_{j=1}^{M}\prod_{t=0}^{2^r-1}(1-\xi_{jt})=0 \tag{23}$$

and

$$\sum_{t=0}^{2^r-1}[(\sum_{k=1}^{M}(-1)^{k-1}\varphi_k(\xi_{1t},\ldots,\xi_{Mt}))\{\prod_{i=1}^{M}(\sum_{k=1}^{M}(-1)^{k-1}\varphi_k(\xi_{1t},\ldots,\bar{\xi}_{it},\ldots,\xi_{Mt}))\}]=0 \tag{24}$$

where $\varphi_k(x_1, x_2, \ldots, x_n)=\sum_{1\leqq i_1<i_2<\cdots<i_k\leqq n}x_{i_1}x_{i_2}\cdots x_{i_k}$ and $\bar{\xi}_{it}=1-\xi_{it}$.

By these equations our problem can be reduced to the following one, that is, "Find d that minimize the pseudo-Boolean function

$$\begin{aligned} y^{**}=&\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{t=0}^{2^r-1}\sum_{u=0}^{2^r-1}N_j\rho_{jk}(\alpha b_{tu}-\beta\hat{b}_{tu})\\ &+\mathfrak{M}\sum_{j=1}^{M}\prod(1-\xi_{jt})\\ &+\mathfrak{M}\sum_{t=0}^{2^r-1}[(\sum_{k=1}^{M}(-1)^{k-1}\varphi_k(\xi_{1t},\ldots,\xi_{Mt}))\{\prod_{i=1}^{M}(\sum_{k=1}^{M}(-1)^{k-1}\varphi_k(\xi_{1t},\ldots,\bar{\xi}_{it},\\ &\ldots,\xi_{Mt}))\}], \end{aligned} \tag{25}$$

where $\mathfrak{M}$ is an arbitrary large positive number."
This problem can be solved by the pseude-Boolean programming.

5. Properties of the Fundamental Matrix*

In this section we consider some properties of the fundamental matrix $N=(N_{ij})=(I-Q)^{-1}$ which determines the coefficients of the objective function to be maximized later in the assignment-improving routine.

* For proofs and details, see reference (3).

Property 1. The variations $\pm\delta$ $(0<\delta\ll 1)$ of element of the matrix $(I-Q)$ change the minor determinant for the matrix $(I-Q)$ by the amount whose absolute value can not be more than $(M-1)^2\delta$, where M is the number of states.

Property 2. Let N_{ij} be the element of $N=(I-Q)^{-1}$, and

$$N_j = \sum_{i=1}^{M} N_{ij}, \qquad N_k = \sum_{i=1}^{M} N_{ik}, \tag{26}$$

and N'_j and N'_k be N_j and N_k, respectively, when the elements of $(I-Q)$ change by $\pm\delta (0<\delta\ll 1)$. If M states of the sequential circuit under consideration are strongly connected and M is not too large and error probabilities are so small that the condition $N_j \gg M^3 \dfrac{\delta}{|I-Q|}$ $(j=1, 2, \ldots, M)$ is satisfied, then

$$\left| \frac{N_j}{N_k} - \frac{N'_j}{N'_k} \right| < \frac{M(M-1)^2}{N_k} \frac{\delta}{|I-Q|} \quad \text{(for all } j \text{ and } k\text{)}. \tag{27}$$

6. The State Assignment by the Use of Pseudo-Boolean Programming

We can derive an important conclusion from the properties 1 and 2 in the previous chapter.

Theorem. If N_j/N_k changes to N'_j/N'_k when a state assignment d changes to d', then we obtain the inequality

$$\left| \frac{N_j}{N_k} - \frac{N'_j}{N'_k} \right| < \frac{M(M-1)^2}{N_j} \frac{r\alpha_0}{|I-Q|}, \tag{28}$$

where r is the integer which is equal to or just larger than $\log_2 M$ and $\alpha_0 = \max(\alpha, \beta)$.

This theorem states that the ratio N_j/N_k of the practically important sequential circuits satisfying above two conditions shows very small amount of variation inspite of change of the state assignment. Thus the ratio of coefficients $N_j \rho_{jk}(1-\alpha b_{tu}-\beta \hat{b}_{tu})$ between any two terms in the quadratic form of ξ_{jt} can be considered as almost constant for a given state assignment problem. Hence by this theorem and the fact that the iteration process terminates when the same assignment as the preceeding one appears, the state assignment problem, which initially necessitates two problem solving stages, can be reduced to the following single pseudo-Boolean program;

"Find $d=\{\xi_{jt}\}$ which minimizes the pseudo-Boolean function y^{**} given by Eq. (25)".

7. Illustrative Examples

Example 1. As a very simple example we present the state assignment problem of the sequential circuit illustrated in Fig. 2. Suppose the input distribution is uniform, *i.e.*,

$$P\{x_\nu = 0\} = P\{x_\nu = 1\} = \frac{1}{2}.$$

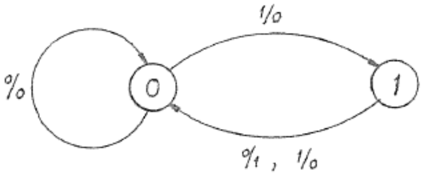

FIG. 2. Example 1.

and $\alpha = 5\times 10^{-7}(1/\mathrm{hr})$, $\beta = 1\times 10^{-7}(1/\mathrm{hr})$. Then we get ρ_{ij}, $B_{tu} = (\alpha b_{tu} + \beta \hat{b}_{tu})$ and N_0/N_1 as follows.

$$(\rho_{ij}) = \begin{array}{c} {}_{i}\,{}^{j} \\ ⓪ \\ ① \end{array} \begin{array}{cc} ⓪ & ① \\ \left(\begin{array}{cc} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{array}\right) \end{array},$$

$$(B_{tu}) = \begin{array}{c} t \quad u \\ 0\,(0) \\ 1\,(1) \end{array} \begin{array}{c} 0 \quad 1 \\ \left[\begin{array}{cc} 0 & \alpha \\ \beta & 0 \end{array}\right] \end{array} = \left[\begin{array}{cc} 0 & 5 \\ 1 & 0 \end{array}\right] \times 10^{-7},$$

$$N_0 : N_1 = 2 : 1.$$

The coefficient matrix is

$$\begin{array}{c} l \\ ⓚ \\ h\ ⓙ \quad t\ u \end{array} \quad \begin{array}{cccc} 0 & 1 & 2 & 3 \\ \overbrace{⓪\,0} & \overbrace{⓪\,1} & \overbrace{①\,0} & \overbrace{①\,1} \end{array}$$

$$(c_{hl}) = (N_j(\rho_{jk}B_{tu})) = \begin{array}{c} 0(⓪-0) \\ 1(⓪-1) \\ 2(①-0) \\ 3(①-1) \end{array} \left(\begin{array}{cccc} 2\cdot 0 & 2\cdot \frac{1}{2}\cdot\alpha & 2\cdot\frac{1}{2}\cdot 0 & 2\cdot\frac{1}{2}\cdot\alpha \\ 2\cdot\frac{1}{2}\beta & 2\cdot 0 & 2\cdot\frac{1}{2}\beta & 2\cdot\frac{1}{2}\cdot 0 \\ 1\cdot 1\cdot 0 & 1\cdot 1\cdot\alpha & 0 & 0 \\ 1\cdot 1\cdot\beta & 1\cdot 1\cdot 0 & 0 & 0 \end{array}\right)$$

$$= \left[\begin{array}{cccc} 0 & 10 & 0 & 5 \\ 2 & 0 & 1 & 0 \\ 0 & 5 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right] \times 10^{-7}.$$

Setting

$$z_0 = \xi_{00},\ z_1 = \xi_{01},\ z_2 = \xi_{10},\ z_3 = \xi_{11},$$

we obtain the pseudo-Boolean function

$$y = \sum_{h=0}^{3}\sum_{l=0}^{3} c_{hl} z_h z_l = (12\, z_0 z_1 + 6\, z_0 z_3 + 6\, z_1 z_2) \times 10^{-7} \tag{29}$$

and Boolean constraints

$$z_0 \vee z_1 = 1, \qquad z_2 \vee z_3 = 1, \tag{30}$$

$$z_0 z_2 = 0, \qquad z_1 z_3 = 0. \tag{31}$$

Therefore our problem is to find z_0, z_1, z_2, z_3 which minimize y shown in Eq. (29) under constraints Eqs. (30) and (31).

First solving Eqs. (30) and (31) we obtain

$$z_0 = \overline{u_0 u_1 u_2 u_3}, \;\; z_1 = u_0 u_1 u_2 u_3, \;\; z_2 = u_0 u_1 u_2 u_3, \;\; z_3 = \overline{u_0 u_1 u_2 u_3}.$$

Setting $t = u_0 u_1 u_2 u_3$

$$z_0 = z_3 = \bar{t}, \;\; z_1 = z_2 = t. \tag{32}$$

Substitution of Eq. (32) for Eq. (29) yields

$$y = \{6(1-t) + 6\,t\} \times 10^{-7} = 6 \times 10^{-7}. \tag{33}$$

Then

$$t = \text{arbitrary}, \qquad y_{\min} = 6 \times 10^{-7}, \tag{34}$$

and $t=0$ and $t=1$ give the assignment $z_0=0$, $z_1=1$, $z_2=1$, $z_3=0$ and $z_0=1$, $z_1=0$, $z_2=0$, $z_3=1$, respectively.

Thus two possible assignment (one (0), (1) to ⓪, ① and the other (0), (1) to ①, ⓪) give the same mean time to failure τ of which value is about 5×10^6 hours for the conditions of this problem.

Example 2. Consider the state assignment problem of the sequential circuit shown in Fig. 3. Suppose the conditions of probability distribution of inputs and probability of malfunction of memory cells are the same as the example 1 except that (0, 0) must be assigned to 0.

Setting

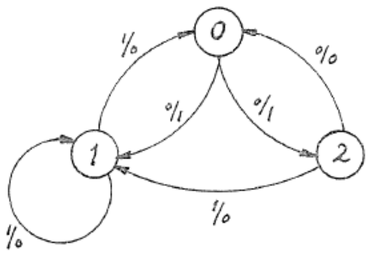

FIG. 3. Example 2.

$$\begin{aligned} &z_0 = \xi_{00}, \;\; z_1 = \xi_{01}, \;\; z_2 = \xi_{02}, \;\; z_3 = \xi_{03} \\ &z_4 = \xi_{10}, \;\; z_5 = \xi_{11}, \;\; z_6 = \xi_{12}, \;\; z_7 = \xi_{13} \\ &z_8 = \xi_{20}, \;\; z_9 = \xi_{21}, \;\; z_{10} = \xi_{22}, \;\; z_{11} = \xi_{23} \end{aligned} \tag{35}$$

the pseudo-Boolean function to be minimized is

$$y=(z_0\cdots z_{11})\begin{pmatrix} 0&0&0&0&0&10&10&20&0&10&10&20\\ 0&0&0&0&2&0&12&10&2&0&12&10\\ 0&0&0&0&2&12&0&10&2&12&0&10\\ 0&0&0&0&4&2&2&0&4&2&2&0\\ 0&15&15&30&0&15&15&30&0&0&0&0\\ 3&0&18&15&3&0&18&15&0&0&0&0\\ 3&18&0&15&3&18&0&15&0&0&0&0\\ 6&3&3&0&6&3&3&0&0&0&0&0\\ 0&5&5&10&0&5&5&10&0&0&0&0\\ 1&0&6&5&1&0&6&5&0&0&0&0\\ 1&6&0&5&1&6&0&5&0&0&0&0\\ 2&1&1&0&2&1&1&0&0&0&0&0 \end{pmatrix}\begin{pmatrix} z_0\\ z_1\\ \cdot\\ \cdot\\ \cdot\\ z_{11}\end{pmatrix} \tag{36}$$

and the Boolean constraints are

$$\begin{gathered} z_0\vee z_1\vee z_2\vee z_3=1,\ z_4\vee z_5\vee z_6\vee z_7=1,\ z_8\vee z_9\vee z_{10}\vee z_{11}=1\\ z_0z_4\vee z_4z_8\vee z_8z_0=0,\ z_1z_5\vee z_5z_9\vee z_9z_1=0,\ z_2z_6\vee z_6z_{10}\vee z_{10}z_2=0\\ z_3z_7\vee z_7z_{11}\vee z_{11}z_3=0 \text{ and } z_0=1 \end{gathered} \tag{37}$$

Solving Boolean equations (37) we obtain

$$\begin{aligned} &z_0=1,\ z_1=z_2=z_3=z_4=0,\ z_5=\bar{u}_6(\bar{u}_1\bar{u}_2\vee\bar{u}_5\vee\bar{u}_3)\vee\bar{u}_4\\ &z_6=u_6,\ \ z_7=\bar{u}_6u_2u_3u_5\vee u_1u_3u_4u_5,\ \ z_8=0\\ &z_9=u_6u_4u_1\vee u_1u_3u_4u_5,\ \ z_{10}=\bar{u}_6u_2\\ &z_{11}=u_6\bar{u}_4\vee u_6\bar{u}_1\vee\bar{u}_6\bar{u}_3\bar{u}_2\vee\bar{u}_4\bar{u}_2\vee\bar{u}_1\bar{u}_2\vee\bar{u}_5 \end{aligned} \tag{38}$$

where $u_1, u_2, \ldots, u_6$ are arbitrary parameter in L_2.
Substitution of Eqs. (38) and (36) yields

$$\begin{aligned} y=36&+17u_2+50u_6-23u_2u_5-17u_2u_6-50u_4u_6+17u_1u_4u_6+49u_2u_3u_5\\ &+23u_2u_5u_6+6u_1u_3u_4u_5-23u_1u_4u_5u_6-37u_2u_3u_4u_5-49u_2u_3u_5u_6\\ &+10u_1u_2u_3u_4u_5+28u_1u_3u_4u_5u_6+37u_2u_3u_4u_5u_6-10u_1u_2u_3u_4u_5u_6. \end{aligned} \tag{39}$$

The pseudo-Boolean algorithm described in Chap. 3 gives $u_1, u_2, \ldots, u_6$ which minimize y given by Eq. (39) and minimum value of y;

$$\begin{aligned} &u_1=t_5\vee t_1,\ \ u_2=\bar{t}_5\vee t_2,\ \ u_3=0,\ \ u_4=t_4\vee t_5,\ \ u_5=1,\ \ u_6=t_5,\\ &y_{\min}=30 \end{aligned} \tag{40}$$

where $t_1, t_2, \ldots, t_5$ are arbitrary parameter in L_2.

Substituting Eq. (40) to Eq. (35) we obtain

$$\begin{aligned} &z_0=1,\ z_1=z_2=z_3=z_4=0\\ &z_5=\bar{t}_5\\ &z_6=t_5\\ &z_7=z_8=0 \end{aligned} \tag{41}$$

$$z_9 = t_5$$

$$z_{10} = \bar{t}_5$$

$$z_{11} = 0$$

Therefore the optimal assignments are $z_0 = z_5 = z_{10} = 1$ (given by $t_5 = 0$) and $z_0 = z_6 = z_9 = 1$ (given by $t_5 = 1$), and the maximum mean time to failure τ_{max} is equal to about 4.0×10^6 hours.

Acknowledgement

The authors should like to express their deep gratitude to the late professor K. Udagawa for his guidance and encouragement during his life, and they also gratefully acknowledge their indebtedness to Assist. Professor T. Fukumura for his helpful suggestions leading them to this work.

References

1) K. Udagawa and Y. Inagaki: "State assignment method of reliable sequential circuits by means of dynamic programming", J. Inst. Elec. Com., Japan, **47**, 6, p. 40, 1964 (in Japanese).

2) P. Ivănescu, Ivo Rosenberg and S. Rudeanu: "Asupra determinarii minimelor functiilor Pseudo Booleene", Studii și Cercetari Matematice **14**, p. 359, 1963.

3) K. Udagawa, Y. Inagaki and M. Goto: "A state assignment of reliable sequential circuits using quadratic programming", J. Inst. Elec. Com., Japan, **47**, 11, p. 1731, 1964 (in Japanese).