01 Jan 1970

State Assignment Selection In Asynchronous Sequential Circuits

Gary K. Maki

James H. Tracey
Missouri University of Science and Technology

Short Notes

State Assignment Selection in Asynchronous Sequential Circuits

GARY K. MAKI, MEMBER, IEEE, AND
JAMES H. TRACEY, MEMBER, IEEE

Abstract—Methods already exist for the construction of critical race-free assignments for asynchronous sequential circuits. Some of these methods permit the construction of many assignments for the same flow table. The algorithm presented here consists of two easy to apply tests which select that critical race-free assignment most likely to produce a set of simple next-state equations. The algorithm has been programmed.

Index Terms—Asynchronous sequential circuits, sequential circuit synthesis, sequential machines, state assignment.

Introduction

The problem of generating internal-state assignments for the realization of asynchronous sequential circuits operating in normal fundamental mode has been solved by Huffman [1], [2], Liu [3], and Tracey [4]. A sequential circuit is said to be operating in normal fundamental mode if all internal state transitions are direct and the input state is never changed unless the circuit is stable internally. In a direct transition, all internal-state variables that change state are excited at the beginning of the transition. Often, the algorithms developed produce several internal-state assignments with the same number of internal-state variables, all of which permit critical race-free realizations of a given flow table. It has been observed that the next-state expressions that result from some of these assignments are simpler than others. The purpose of this note is to present a means of predicting which of the internal-state assignments will yield a relatively simple set of next-state expressions.

State Assignment Selection

The development in this note will not be concerned with optimum coding in the input states. Therefore, the problem considered will be one of finding the next-state expressions on a per-column basis. If there are n internal-state variables and a flow table of m columns (m input states), the general form for the next-state expressions will be

$$
\begin{aligned}
Y_1 &= f_{11}(y_1, y_2, \cdots, y_n)I_1 \\
&\quad + f_{12}(y_1, y_2, \cdots, y_n)I_2 + \cdots \\
&\quad + f_{1m}(y_1, y_2, \cdots, y_n)I_m \\
Y_2 &= f_{21}(y_1, y_2, \cdots, y_n)I_1 \\
&\quad + f_{22}(y_1, y_2, \cdots, y_n)I_2 + \cdots \\
&\quad + f_{2m}(y_1, y_2, \cdots, y_n)I_m \qquad (1)\\
&\vdots \\
Y_n &= f_{n1}(y_1, y_2, \cdots, y_n)I_1 \\
&\quad + f_{n2}(y_1, y_2, \cdots, y_n)I_2 + \cdots \\
&\quad + f_{nm}(y_1, y_2, \cdots, y_n)I_m
\end{aligned}
$$

where $y_1, y_2, \cdots, y_n$ are the present-state variables; $Y_1, Y_2, \cdots, Y_n$ are the next-state variables; $I_1, I_2, \cdots, I_m$ are the input states; and $f_{11}, f_{12}, \cdots, f_{nm}$ are functions of the internal-state variables alone.

The intent of this development is to obtain a figure of merit that will predict which internal-state assignment for a normal fundamental-mode asynchronous sequential machine will yield simpler next-state expressions than other assignments for the same machine. In this note the assignment-selection process is considered to be the selection of that assignment which will tend to minimize the functions $f_{11}, f_{12}, \cdots, f_{nm}$ to simplest sum-of-products expressions. The assignment which tends to minimize the complete set of these functions will be referred to as a "good" assignment. It is realized that the coding of the input states will affect the complexity of the next-state expressions, but significant simplification will result from choosing the assignment which produces relatively simple f_{ij} coefficients in (1). As stated previously, optimum coding of the input states will not be part of this study.

There are two desirable characteristics in critical race-free assignments that are easy to test for. One of these characteristics permits certain f_{ij}'s in (1) to be equal to 1 or 0. The other permits some f_{ij}'s to be y_i.

The following discussion describes exactly how the preceding characteristics can be determined. First, some useful definitions are given.

Definition 1: A partition Π on a set S is a collection of subsets of S such that their pairwise intersection is the null set. The disjoint subsets are called the blocks of Π. If the set union of these subsets is S, the partition is completely specified; otherwise, the partition is incompletely specified. Elements of S that do not appear in Π are called unspecified or optional elements with respect to that partition.

Definition 2: The two-block partitions $\tau_1, \tau_2, \cdots, \tau_n$ induced by the internal-state variables $y_1, y_2, \cdots, y_n$, respectively, are called the set of *τ-partitions* of that assignment. The elements of each block are internal states with one block of each τ_i consisting of those states encoded with a 0 by y_i and the other block consisting of those states encoded with a 1 by y_i.

Manuscript received December 5, 1967. The research reported here was supported in part by the National Science Foundation under Grants GK-820 and GK-2017.

G. K. Maki is with the Department of Electrical Engineering, University of Idaho, Moscow, Idaho.
J. H. Tracey is with the Department of Electrical Engineering, University of Missouri at Rolla, Rolla, Mo.

Internal states		Assignment y_1	y_2	y_3
a	-	0	0	0
b	-	0	1	0
c	-	1	1	0
d	-	1	0	0
e	-	0	1	1
f	-	1	0	1

$\tau_1 = \{\overline{a,b,e}; \ \overline{c,d,f}\}$

$\tau_2 = \{\overline{a,d,f}; \ \overline{b,c,e}\}$

$\tau_3 = \{\overline{a,b,c,d}; \ \overline{e,f}\}$

Fig. 1. Internal-state assignment and corresponding τ-partitions.

	I_j
a-1	3
b-2	4
c-3	3
d-4	4
e-5	4
f-6	6
g-7	6
h-8	8
j-9	9

$\alpha_j = \{\overline{a,\underline{c}}; \ \overline{b,\underline{d},e}; \ \overline{\underline{f},g}; \ \overline{\underline{h}}; \ \overline{\underline{j}}\}$

Fig. 2. Partial flow table and corresponding column partition.

The following example in Fig. 1 will help illustrate the above definition. Here the first block in each τ-partition is a set of the internal states that have been coded with a 0 by each internal-state variable and the second block is a similar set of the states that have been coded with a 1 by each internal-state variable. It should be pointed out that the ordering of the blocks is unimportant.

Definition 3: A *k-set* of a single column of a flow table consists of all $k-1$ unstable entries leading to the same stable state, together with that stable state.

Definition 4: A *column partition* α_j is a collection of the k-sets of the column of a flow table with input state I_j, where each k-set constitutes a single block as shown in Fig. 2. Elements corresponding to stable states in the column are underlined for easy identification.

Theorem 1: If all the stable states of a column partition α_j are in the same block of a τ-partition τ_i, the next-state coefficient f_{ij} will be 1 or 0.

Proof: Each of the unstable states of the k-set k_r corresponds to a transition to the stable state in k_r. The next-state entry for all internal states involved in transitions to the stable state of k_r will have to be the same as the code assigned to the stable state. The direct transition between each unstable state and the stable state within k-set k_r may involve more than these two internal states in normal mode operation. However, the next-state entry for all the internal states involved in the transition between each unstable state and the corresponding stable state of k_r will have to be the same as the code assigned to the stable state. This will insure that a transition from the unstable states to the stable state will be independent of the order in which the excited state variables change. In other words, the next-state entries for all transitions of k-set k_r will be determined completely by the code assigned to the stable state of k_r. If the internal-state variable y_i in the code assigned to the stable state of k_r in a certain column of a flow table is 1 (or 0), then all the internal states involved in transitions to the stable state of k_r will have a next-state entry of 1 (0) for the next-state variable Y_i. It follows that if y_i has the same value in the code for all stable states in a column, then Y_i has the same next-state entry in the entire column, wherever specified. This corresponds to all the stable states of the column partition α_j appearing in the same block of τ-partition τ_i.

Constant coefficients f_{ij} can be determined from the α-partitions and τ-partitions. The stable states of each column are identified in the corresponding α-partition. If all the stable states of an α-partition α_j are in the same block of the τ-partition τ_i, then the internal-state variable y_i is 1 or 0 in the codes for all the stable states in the column with input state I_j. It follows that the next-state variable Y_i will have a next-state entry of 1 (0) in all the specified states of the column with input I_j. The partial next-state expression in this case is

$$Y_i = 1\ (0)\ I_j.$$

A method to determine constant f_{ij}'s is as follows.

1) List the stable states associated with each column partition α_j.
2) Compare each list of stable states from (1) with each τ-partition. If the stable states from α_j are in the same block of τ_i, f_{ij} will be 1 or 0, depending on the coding of the blocks of τ_i.

To demonstrate this method, the column partition of Fig. 2,

$$\alpha_j = \{\overline{a, \boldsymbol{c}}; \overline{b, \boldsymbol{d}, e}; \overline{\boldsymbol{f}, g}; \overline{\boldsymbol{h}}; \overline{\boldsymbol{j}}\},$$

will be compared to the τ-partitions

$$\tau_1 = \{\overline{a, c, d, f, h, j}; \overline{b, e, g}\}$$

and

$$\tau_2 = \{\overline{a, c, f, g, h}; \overline{b, d, e}\}.$$

The stable states in the column partition are c, d, f, h, and j.

By inspection, one can see all the stable states of α_j are included in the same block of τ_1, but not in τ_2. Therefore, it can be concluded from Theorem 1 that f_{2j} will not be constant but f_{1j} will be 1 or 0. If the first block of τ_1 is coded with 0 by the internal-state variable y_1, then the partial next-state expression will be $Y_1 = 0 \cdot I_j$.

Clearly, it is desirable to have a maximum number of constant f_{ij} coefficients in (1). The number of such constant coefficients in an assignment t will be designated A_t.

As stated earlier, the second desirable characteristic in an assignment is that it produce a number of f_{ij}'s equal to y_i.

Definition 5: Partition Π_2 is less than or equal to Π_1 ($\Pi_2 \leq \Pi_1$) where Π_1 and Π_2 may be incompletely specified, if and only if all elements specified in Π_2 are also specified in Π_1 and each block of Π_2 appears in a block of Π_1.

Theorem 2: A coefficient f_{ij} in (1) will equal y_i if $\alpha_j \leq \tau_i$.

Proof: The transitions in the column of a normal flow table with input state I_j will occur only between internal states of the same k-set. Transitions between an unstable and stable state (a transition pair) of a k-set may involve more than two internal states, but can never involve an internal state that is a member of another k-set. This follows from the characteristics of a critical race-free assignment. If the column partition α_j is less than or equal to a τ-partition τ_i, each block of the column partition α_j, which is a k-set of the column partition, is included in one of the blocks of the partition τ_i. Therefore, transition pairs always appear within a block of τ_i. Since all states within a block of τ_i are coded with the same value of y_i, y_i will never be excited to change state under input I_j. The result is that the next state of y_i is equal to the present state of y_i and therefore $f_{ij} = y_i$.

To demonstrate the use of Theorem 2, the τ-partitions

$$\tau_1 = \{\overline{a, c, f, g, h}; \overline{b, d, e, j}\}$$

$$\tau_2 = \{\overline{a, b, d, e}; \overline{c, f, g, h, j}\}$$

will be compared to the column partition of Fig. 2:

$$\alpha_j = \{\overline{a, \boldsymbol{c}}; \overline{b, \boldsymbol{d}, e}; \overline{\boldsymbol{f}, g}; \bar{\boldsymbol{h}}; \bar{\boldsymbol{j}}\}.$$

Each block of α_j is contained in a block of τ_1 or ($\alpha_j \leq \tau_1$). This means that in all the transitions of this column, internal-state variable y_1 will not change state, or the next state will always be equal to the present state for any transition in the column with input I_j. However, τ_2 does not satisfy Theorem 2 in that the block $\overline{a, c}$ of α_j does not appear in a block of τ_2. The internal-state variable y_2 will therefore undergo a change of state during the transition from state a to state c.

It might be noted that in making the test for $\alpha_j \leq \tau_i$ for all i and j, one only has to be able to show that each block of the column partition α_j is contained in a block of the τ-partition τ_i. Furthermore, only the blocks of α_j which contain two or more elements need be considered.

Let B_t be the total number of terms $f_{ij} = y_i$ in assignment t. Following is an algorithm that can be used to obtain B_t.

1) Form the partitions α_j and τ_i for all values of i and j.
2) Determine the number of occurrences under input I_1 where $f_{i1} = y_i$. Repeat for $I_2 \cdots I_m$. B_t will be the total number of occurrences.

Two characteristics of a "good" internal-state assignment have been discussed. The relative weight to attach to A_t and B_t for internal-state assignment t may vary with each type of implementation. The weight for an internal-state assignment t will be defined as

$$W_t = pA_t + B_t \tag{2}$$

where W_t is the weight attached to internal-state assignment t and p is a variable that allows one to adjust the weights of A_t and B_t with respect to each other. It seems safe to conclude that a constant coefficient would require a lesser amount of combinational logic for synthesis than a literal coefficient. However, the designer of a sequential circuit will in general decide on a value for p by determining how much easier it is to implement a constant coefficient as opposed to a literal coefficient. This could be done by obtaining a cost figure to compare the coefficients. This cost figure would also depend on the number of literals associated with each input state.

In general, then, p will vary with each type of implementation. For purposes of illustration in this note, p will be assigned the value of 2, which is somewhat arbitrary, to demonstrate the assignment-selection procedure. The weight for an internal-state assignment t will be defined in the examples shown in this note as

$$W_t = 2A_t + B_t. \tag{3}$$

The internal-state assignment with the largest weight associated with it will be predicted to yield relatively simple next-state expressions.

Sequential machine A in Fig. 3 can be coded with either of the two internal-state assignments shown. The criteria developed above will be used to predict which assignment will produce simpler next-state expressions. First B_t will be obtained by following the procedure developed in this note.

Step 1: The column partitions are

$$\alpha_1 = \{\overline{\boldsymbol{a}, d}; \overline{b, \boldsymbol{c}}; \overline{e, \boldsymbol{f}}\}$$

$$\alpha_2 = \{\overline{a, \boldsymbol{f}}; \overline{b, \boldsymbol{d}}; \bar{\boldsymbol{c}}; \bar{\boldsymbol{e}}\}$$

$$\alpha_3 = \{\overline{a, \boldsymbol{d}}; \overline{\boldsymbol{b}, c}; \overline{\boldsymbol{e}, f}\}.$$

The τ-partitions for Assignment 1 are

$$\tau_1 = \{\overline{a, d, e, f}; \overline{b, c}\}$$

$$\tau_2 = \{\overline{a, b, c, d}; \overline{e, f}\}$$

$$\tau_3 = \{\overline{a, c, f}; \overline{b, d, e}\}.$$

	I_1	I_2	I_3
a-1	1	6	4
b-2	3	4	2
c-3	3	3	2
d-4	1	4	4
e-5	6	5	5
f-6	6	6	5

	Assignment 1			Assignment 2		
	y_1	y_2	y_3	y_1	y_2	y_3
a	0	0	0	0	0	0
b	1	0	1	1	0	1
c	1	0	0	1	1	1
d	0	0	1	0	0	1
e	0	1	1	1	1	0
f	0	1	0	0	1	0

Fig. 3. Machine A with two assignments.

and for Assignment 2 are

$$\tau_1 = \{\overline{a, d, f}; \overline{b, c, e}\}$$
$$\tau_2 = \{\overline{a, b, d}; \overline{c, e, f}\}$$
$$\tau_3 = \{\overline{a, e, f}; \overline{b, c, d}\}.$$

Step 2: The τ-partitions for Assignment 1 that meet the conditions of Theorem 2 are

$$\alpha_1 \leq \tau_1$$
$$\alpha_1 \leq \tau_2$$
$$\alpha_2 \leq \tau_3$$
$$\alpha_3 \leq \tau_1$$
$$\alpha_3 \leq \tau_2,$$

and for Assignment 2

$$\alpha_2 \leq \tau_3.$$

Therefore,

$$B_1 = 2 + 1 + 2 = 5$$
$$B_2 = 0 + 1 + 0 = 1.$$

A_t is obtained by following the procedure given earlier.

Step 1: The stable states in each α-partition are as follows:

$$\alpha_1 : a, c, \text{ and } f$$
$$\alpha_2 : f, d, c, \text{ and } e$$
$$\alpha_3 : d, b, \text{ and } e.$$

Step 2: Comparing these stable states with the τ-partitions of Assignment 1, one finds a, c, and f from α_1 are in the first block of τ_3 and b, d, and e from α_3 are in the second block of τ_3.

Performing an identical comparison with the τ-partition of Assignment 2, one finds that none of the τ-partitions contain all the stable states of any of the column partitions in a single block. A_1 for Assignment 1 is 2 and A_2 for Assignment 2 is 0.

The weight for each assignment is

$$W_1 = 2(2) + 5 = 9,$$

and

$$W_2 = 2(0) + 1 = 1.$$

From the above information, Assignment 1 would be predicted to have the simpler next-state expressions.

The next-state expressions for assignment 1 are

$$\begin{aligned} Y_1 &= y_1 I_1 + y_1 y_3' I_2 + y_1 I_3 \\ Y_2 &= y_2 I_1 + (y_2 + y_1' y_3') I_2 + y_2 I_3 \\ Y_3 &= y_3 I_2 + I_3 \end{aligned}$$

and for Assignment 2 are

$$\begin{aligned} Y_1 &= y_1 y_3 I_1 + y_1 y_2 I_2 + (y_1 + y_2) I_3 \\ Y_2 &= (y_1 + y_2) I_1 + (y_2 + y_3') I_2 + y_2 y_3' I_3 \\ Y_3 &= y_1 y_3 I_1 + y_3 I_2 + (y_3 + y_2') I_3. \end{aligned}$$

Clearly, Assignment 1 yields the simpler next-state expressions.

The assignment selection algorithm presented above has been implemented in the form of a computer program. The program is written in PL/1 and is part of a larger synthesis program which automatically generates design equations from flow table specifications. For more details concerning the total program and/or programming considerations, the reader is referred to [5] and [6].

References

[1] D. A. Huffman, "The synthesis of sequential switching circuits," *J. Franklin Institute*, vol. 257, pp. 161–190 and 257–303, March and April 1954.

[2] ——, "A study of the memory requirements of sequential switching circuits," M.I.T., Cambridge, Mass., Tech. Rept. 293, March 1955.

[3] C. N. Liu, "A state variable assignment for asynchronous sequential circuits," *J. ACM*, vol. 10, pp. 209–216, April 1963.

[4] J. H. Tracey, "The internal state assignment for asynchronous sequential machines," *IEEE Trans. Electronic Computers*, vol. EC-15, pp. 551–560, August 1966.

[5] G. K. Maki, J. H. Tracey, and R. J. Smith, II, "Generation of design equations in asynchronous sequential circuits," *IEEE Trans. Computers*, vol. C-18, pp. 467–472, May 1969.

[6] R. J. Smith, II, J. H. Tracey, W. L. Schoeffel, and G. K. Maki, "Automation in the design of asynchronous sequential circuits," *1968 Spring Joint Computer Conf., AFIPS Proc.*, vol. 32. Washington, D. C.: Thompson, 1968, pp. 53–60.