THE REALIZATION OF ASYNCHRONOUS CIRCUITS USING DIRECTIONAL TRANSITION LOGIC

by

YUN-CHUNK KWAN

Submitted to the Department of Electrical Engineering in partial fulfilment of the requirements for the degree of Master of Science.

Department of Electrical Engineering
Faculty of Pure and Applied Science,
The University of Ottawa,
Ottawa, Canada
1971

ABSTRACT

A Directional Transition Logic Circuit (D. T. L. C.) is a circuit built with directional transition level-multipliers, transition or-gates, and ordinary logic gates.

A directional transition level-multiplier, named, 'K-element', is a circuit with two level inputs of which one is regarded as a transition input while the other is treated as ordinary level input, and two outputs. One of the outputs responds to the upward level change of the transition input, but the other output depends on the downward level change of the transition input. During any level change of the transition input, the ordinary level input must maintain a constant voltage level.

A transition or-gate, named 'M-element' is a circuit with at least two transition inputs and only one transition output, which produces a transition signal whenever a transition signal occurs on an input terminal.

This thesis considers the problem of realization of the K-element, and applies the concept of directional transition logic in the design of asynchronous circuits. Straightforward methods of realizing asynchronous circuits by directional transition logic are introduced.

The use of K-elements and the concept of directional transition logic avoid complicated race and hazard problem, result in circuit with simpler structure and bring closer the theories of synchronous and asynchronous sequential machines.

ACKNOWLEDGEMENTS

The author wishes to express his deep and sincere sense of gratitude to his supervisor Dr. C. L. SHENG for his guidance and providing stimulating criticisms and discussions throughout the course of this work.

Thanks are also due to Dr. Who-Kee CHUNG, Dr. S. R. DAS, Dr. N. GEORGANAS and Mr. H. MAR for their willingness to listen and offer suggestions.

Finally, the skillful and conscientious work done by my colleague Mr. S. K. SO who did the typing is also acknowledged.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Survey of the major problems

The concepts of transition signal and transition logic are first introduced by Chuang(1). In his paper, he regards a transition logic circuit as a circuit constructed from two basic transition logic elements, called G-element (a transition level-multiplier) and M-element (a transition or-gate), together with ordinary gates. He points out that the synthesis method for asynchronous machine, by applying the idea of transition logic, is simpler than ordinary level sequential synthesis method, mainly because there is no need to consider race and hazard difficulties. This approach also avoids the construction of a level flow table which is generally much larger than the transition flow table describing the circuit.

Chuang considers a transition signal as a change of binary level, either from 0 to 1, or from 1 to 0, regardless of the direction, and he handles the asynchronous problem by using G-elements. This kind of elements can change the input-output relationship of any level curcuit from asynchronous form to transition form.

We are going one step further by considering the transition signal as a directed change of binary level. A directional transition signal is a transition signal with its level changes classified into upward (from low level to high level) and downward (from high level to low level) transitions. Neither two upward transitions, nor two downward transitions can occur consecutively due to the fact that when a signal changes from low level to high level at the first time, it must return from high level to low level at the second time, or vice versa. This property of a directional transition signal is fundamental for the realization of directional transition logic circuits.

Principally, directional transition logic circuits are level sequential circuits. Therefore, the conventional synthesis method for level logic circuits can be applied. Unfortunately, the level logic circuits synthesis becomes laborious as the number of transition variables increases.

For the purpose of realization of directional transition logic, a new element, named 'K-element', is developed, which is able to change an asynchronous problem into transition form, and to distinguish between the direction of level changes of the transition input affecting the outputs of the circuit as well.

The synthesis method used here is similar to the one used by Chuang except that K-elements are used instead of G-elements. But, in some cases a simpler circuit realization with relatively few interconnections between elements can be obtained, such that, the circuit would respond to directional change of inputs rather than to simple change of inputs.

In Chapter 1, an effort is made to orient the reader and to survey the major problems to be discussed.

In Chapter 2, first, we introduce the structure of flow table of the directional transition logic circuits. Then we provide a comprehensive studies of describing, analyzing, and synthesizing the K-element. Its level logic circuit, together with the transition logic representation, is also shown. Next, we discuss the characteristics of the M-element and its application in general. Finally, we present the circuit configuration of directional transition logic circuit, the functional expressions of its state variables and output variables and the methods of dealing with hazard and race difficulties.

The synthesis procedures for directional transition circuits are outlined in Chapter 3. Also, some examples are presented to demonstrate these synthesis procedures.

Chapter 4 is the conclusion of this work, and suggestions for further study will also be discussed.

An Appendix is included at the end of this thesis with some related topics, such as 'Relation between G-element and K-element', 'Synthesis of G-element with K-elements', 'Synthesis of K-element with G-elements', and 'Circuit realization with G-elements' for completeness.

1.2 Transition logic circuits (1)

Before we introduce the directional transition logic, the transition logic will be discussed in this section.
We will not study the characteristics of this logic in detail at this moment, however, certain items will be brought out to describe the operation of a transition circuit in the followings.

A transition logic circuit is a circuit built with transition level- multipliers (G-elements), transition or-gates (M-elements), and ordinary logic gates, which is shown in Figure 1.1.

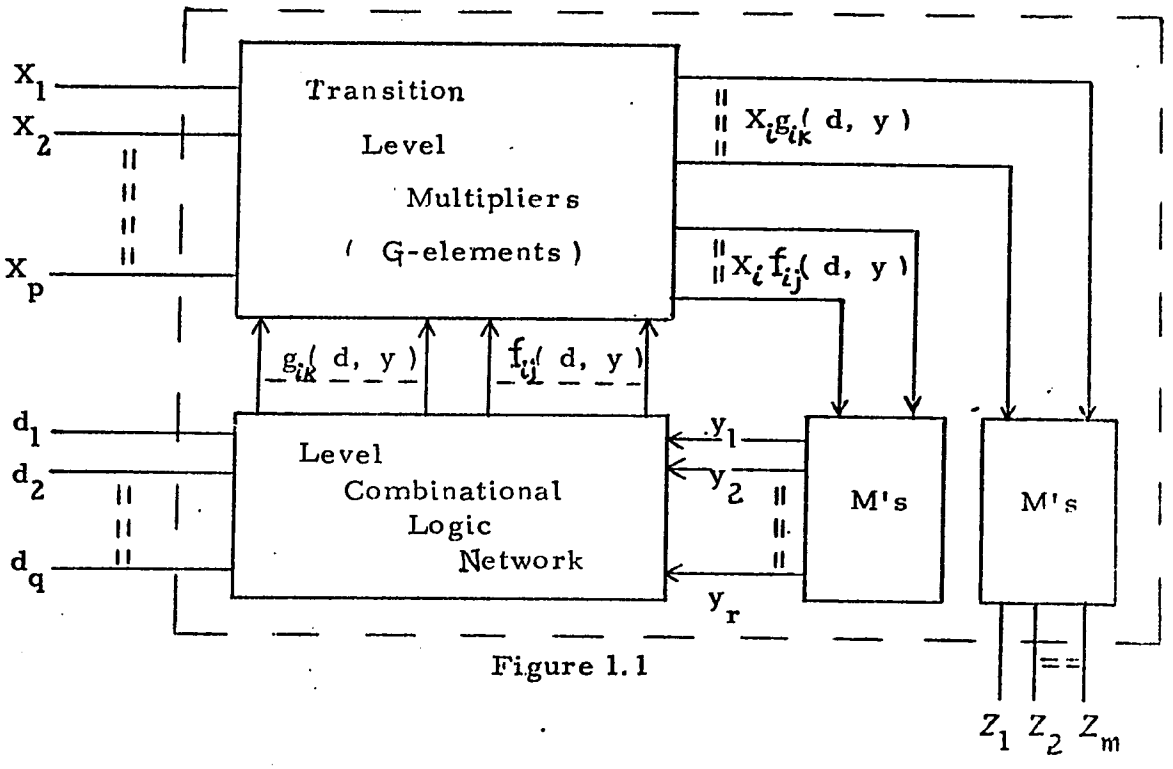


Figure 1.1

Here X's are the transition inputs of the circuit

d's are the level inputs of the circuit

Z's are the transition outputs of the circuit

i.e.

If X = 1, it implies the level signal x has a level change.

If X = 0, it implies the level signal x has no level change.

According to Chuang's result, a G-element is a transition level-multiplier with two outputs and two level inputs, of which one is regarded as a transition input while the other is treated as ordinary level input. Its black box representation in transition form can be shown as Figure 1.2.

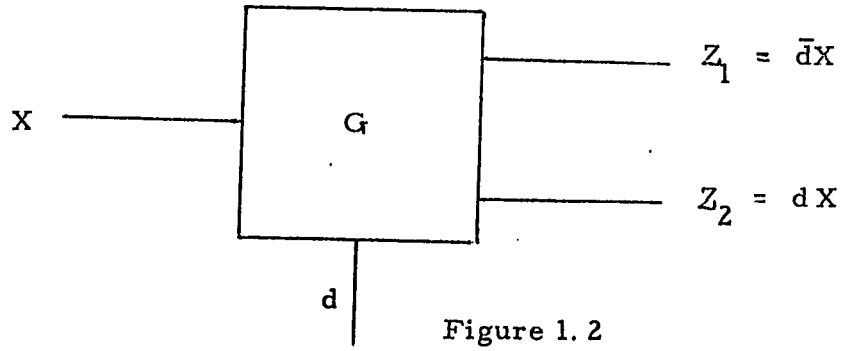


Figure 1.2

A G-element is characterized by the operation that an occurrence of a transition signal at input X (with no distinction between the directions of its level change) of the G-element produces a transition signal at output Z_1 or output Z_2 depending on whether the input level at the level input d is 0 or 1.

In order to form the logical sum of product terms representing transition signals, another element, named 'M-element' was also developed by Chuang, which produces a transition signal whenever a transition signal occurs on one of its input terminals. This element will be studied in Sec. 2.3 in more detail.

From the transition flow table (1,2) of a transition circuit, which is similar to a flow table of a Mealy machine, we can derive its

excitation functions and output functions in transition level form as follows:

$$Y_j = \sum_{i=1}^{P} X_i f_{ij}(d,y), \quad j = 1, 2, \ldots, r.$$

$$Z_k = \sum_{i=1}^{P} X_i g_{ik}(d,y), \quad k = 1, 2, \ldots, m.$$

where ' Σ ' represents Boolean sum X_i, Y_j and Z_k represent the transition signals of input variables, state variables, and output variables of the circuit respectively. Also, $f_{ij}(d,y)$ and $g_{ik}(d,y)$ are level signals and generated from the level combinational logic network in which 'y' is the level signal representing the internal state variable and 'd' is the level input of the transition circuit.

The flow table of an asynchronous circuit which satisfies the operation conditions for a transition circuit can always be transformed into transition flow table, and synthesized with transition logic. In conclusion, this kind of synthesis avoids race and hazard difficulties usually found in asynchronous circuits as well as avoiding the construction of a large size level flow table. Also, the circuit realizations of this kind of circuit will be given in Appendix for reference.

1.3 Definitions and operation conditions of Directional Transition Circuits.

Definition 1 A level signal change from 0 to 1 or from low level to high level is called an upward transition of the signal as shown in Figure 1.3 (a).

Definition 2 A level signal change from 1 to 0 or from high level to low level is called a downward transition of the signal as shown in Figure 1.3 (b).

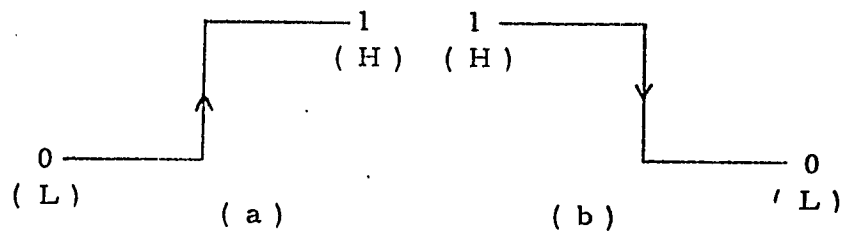


Figure 1. 3 (a, b)

Definition 3 A transition input of a transition circuit is a level input signal, such that the output behavior of the circuit depends on the level change of this level input signal.

Definition 4 A directional transition input is a transition input with the directions of its level change (upward or downward) specified.

A 'Directional Transition Signal' is a change of binary level which is either an upward signal transition or a downward signal transition. In a broad sense, a 'Directional Transition Logic Circuit' is a switching circuit with at least one of its input variables represented as a directional transition signal, such that the outputs of the curcuit respond to different types of level change of the transition inputs. This kind of circuit appears to be quite useful in asynchronous system realizations, because of the great convenience resulted from the avoidance of consideration of race and hazard difficulties.

Transition logic and level sequential logic are but two different views of the same object, since a circuit specified in terms of transition variables can be described in terms of level variables by treating each of the transition signals as a binary change of a level signal. Similarly, a level sequential circuit is always describable in terms of transition variables.

We can synthesize a directional transition logic circuit by considering it as a level sequential circuit and applying Huffman's conventional method (6). This kind of synthesis method becomes very laborious as the number of transition variables is increased, because the size of level flow table would increase rapidly as the number of transition variables increases. In order to by pass this difficulty, a simpler synthesis method by using the transition concept directly is desired.

The black box representation of a general directional transition logic circuit, with two types of inputs and two types of outputs, is given in Figure 1.4, where (X_1, X_2, , , , X_s) represents 's' transition inputs, while (l_1, l_2, , , , l_t) represents 't' level inputs. (Z_{11}, Z_{21}, , , , , Z_{m1}) represents one set of transition outputs which respond to the upward transition of the transition inputs, while (Z_{12}, Z_{22}, , , , Z_{m2}) is another set of transition outputs which depend only on the downward transition of the transition inputs.

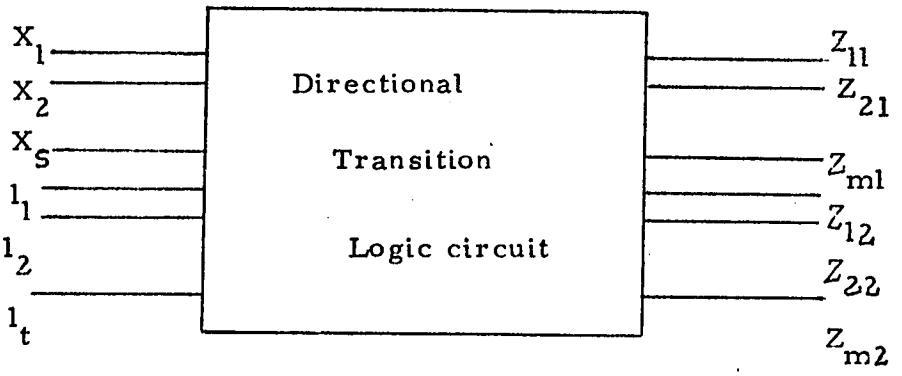


Figure 1.4

The transition logic circuits which can be realized by the model of Figure 1.4, are characterized by the following operating conditions.

(1) The input level can change only when the circuit is internally stable (fundamental mode).

(2) (a) No more than one level change can occur at the transition input terminals at each time.

(b) Level input change is not permitted, unless the level change of the transition input is completed.

(c) While only one level change is permitted to happen at the transition input terminals at each time, simultaneous changes at level input terminals are allowed.

(3) Changes of internal states occur merely in response to level changes at transition input terminals.

(4) The nature of the transition inputs is known. That is, the upward or downward transition of the inputs must be well defined.

CHAPTER 2

DIRECTIONAL TRANSITION LOGIC CIRCUIT

2.1 Flow Table of the Directional Transition Logic Circuit

A general form of flow table for the directional transition logic circuits which satisfy above operating conditions is shown in Table 2.1. It is a complete description of the circuit to be realized. This table has a column on the left hand side representing the set of present states, and columns with headings, X_1, X_2, , , , X_s. Each of these columns has two subcolumns which are called the subtable of the flow table and labeled X_{i1} and X_{i2}, so that, we can easily find out under what kind of level change of a transition input (downward or upward), the state transition would occur. Furthermore, for each subtable there is a row of minterms representation of the set of level variables (1's) to indicate the variation of the level inputs during the transition. Each non-empty entry of the table is a pair whose first element is a next state, and the second element is an m-bit binary number denoting output transitions, '1' representing a transition, which can be either an upward transition or a downward transition, and '0' representing no transition.

Referring to this Mealy type finite state machine flow table, we can do the circuit realization by following the familiar pulse sequential circuit synthesis method. That is we treat the transition signals as pulses and the result is a circuit with K-elements, M-elements and ordinary gates connected together, which has the ability of distinguishing the effects of the directional transition of the inputs upon the outputs. Because, there are many asynchronous problems that can be conveniently described in terms of level transition, the transition logic point of view is advantageous.

TABLE 2.1 Directional Transition Flow Table of A Directional Transition Circuit

I / S	X_1		X_2		...	X_s	
	X_{11}	X_{12}	X_{21}	X_{22}	...	X_{s1}	X_{s2}
	$1(1) - - - 1(2^t)$	$1(1) - - - 1(2^t)$	$1(1) - - - 1(2^t)$	$1(1) - - - 1(2^t)$		$1(1) - - - 1(2^t)$	$1(1) - - - 1(2^t)$
S_1							
S_2							
S_3							
-							
-							
-							
-							
-							
S_n							

Each entry is a pair whose first element is a next state, and the second element is an m-bit binary number denoting output transitions. '1' represents a transition, and '0' represents no transition

$1(1), 1(2), 1(3), , , , 1(2^t)$ are minterms of t variables $1_1, 1_2, 1_3, , , , , 1_t$

Any asynchronous circuit described by a primitive level flow table can be realized with directional transition logic, and every primitive level flow table can be transformed into a directional transition flow table. The algorithm as developed below can be used to carry out this transformation.

Algorithm A

Step 1 Consider each of the level outputs as a transition output and each of the level inputs as a directional transition input.

Step 2 Use the different directional transition input variables as the column headings of the directional transition flow table.

Step 3 Use the upward transition variable and downward transition variable of the corresponding directional transition variable as headings of the two subtables included in each of the columns.

Step 4 Transform all state transitions and output transitions as shown in the primitive flow table in the directional transition flow table by inserting state numbers and binary numbers representing state transitions and output transitions in the proper entries of the directional transition table.

Step 5 Treat those entries in the directional transition table as 'don't care' terms corresponding to which the state transition or output transition does not exist in the primitive flow table. The resulting table is the directional transition table derived from the corresponding primitive flow table.

Similarly, any directional transition table can be transformed into a primitive level flow table as well. The transformation is carried out by applying the following algorithm.

Algorithm B

Step 1 Consider each of the transition outputs as a level output and each of the transition inputs as a level input.

Step 2 Use the different minterm representations of the level input variables as the input states and the level output variables as the column headings of the output vectors of the primitive flow table.

Step 3 Consider the entry of the first row and first column (column containing all zeros as its heading) of the primitive flow table as the starting stable state, then, transform all state transitions and output transitions as shown in the directional transition flow table in the primitive flow table by inserting state numbers and binary numbers in the proper entries of the primitive level flow table, respectively. These state numbers and binary numbers represent the stable states and level outputs of the circuit.

Step 4 Treat those entries in the primitive level flow table as 'don't care' terms which correspond to the state transitions or output transitions that do not exist in the directional transition table, and insert a state number in the appropriate entry representing an unstable state which corresponds to a final stable state for each state transition. The resulting table is the primitive flow table obtained from the corresponding directional transition table.

The following is an example for illustration, where a level flow table is first derived from the given verbal description of a counter to be designed as shown in Table 2.2. If we treat both inputs as transition inputs X_1 and X_2 (no input is treated as level input in this case), and choose the symbols X_{11}, X_{21} and X_{12}, X_{22} as the upward transition and downward transition representations, respectively, of the inputs (X_1, X_2), then Table 2.2 can be changed to the form of Table 2.3. It is important that those empty entries in the table can be treated as ' don't care terms ', because of the property of the directional transitions of the input signals.

Example 1

Modulo-Four-Counter. The device to be designed has two outputs z_1 and z_2, which can represent the decimal numbers 0, 1, 2 and 3 in binary notation. There are two push button switches (x_1 and x_2) used as inputs. When x_1 is depressed, $x_1x_2 = 10$, the output advances by one, that is 0 - 1 - 2 - 3 - 0. When x_2 is depressed, $x_1x_2 = 01$, the output is decreased by one, that is, 0 - 3 - 2 - 1 - 0. Only one of the push button switches can be activated at a given time; thus between any two activations of the inputs, the input combination becomes $x_1x_2 = 00$, and the output remains the same as determined by the previous input combination.

In Table 2.3, columns N_1 and N_2 represent the transition of the outputs under the influence of different kind of level change of the transiton inputs, 1 denotes the occurrence of a transition of the output, 0 denotes no transition.

Example 1 has given us a general idea of transforming an ordinary level flow table into a directional transition flow table. Details of circuit realization of the D.T. L. C., and the transformation of a primitive level flow table into a directional transition flow table will be given after the synthesis procedure is discussed.

TABLE 2.2 Level Flow Table of the Modulo-Four-Counter

	00	01	11	10	z_1	z_2
1	(1)	9	-	2	0	0
2	3	-	-	(2)	0	1
3	(3)	12	-	4	0	1
4	5	-	-	(4)	1	0
5	(5)	11	-	6	1	0
6	7	-	-	(6)	1	1
7	(7)	10	-	8	1	1
8	1	-	-	(8)	0	0
9	7	(9)	-	-	1	1
10	5	(10)	-	-	1	0
11	3	(11)	-	-	0	1
12	1	(12)	-	-	0	0

TABLE 2.3 Directional Transition Flow Table of The Modulo-Four-Counter

	X_1		X_2	
	X_{11}	X_{12}	X_{21}	X_{22}
1	2, 01	-	9, 11	-
2	-	3, 00	-	-
3	4, 11	-	12, 01	-
4	-	5, 00	-	-
5	6, 01	-	11, 11	-
6	-	7, 00	-	-
7	8, 11	-	10, 01	-
8	-	1, 00	-	-
9	-	-	-	7, 00
10	-	-	-	5, 00
11	-	-	-	3, 00
12	-	-	-	1, 00
	N_1N_2	N_1N_2	N_1N_2	N_1N_2

2.2 K-element -- A Directional Transition Level-Multiplier

(A) Analysis

In order to change the input-output relationship of a level sequential circuit from asynchronous form (fundamental mode flow table) into transition form (directional transition flow table), as mentioned before, a K-element is developed for this purpose. A K-element is a level logic circuit with one transition input (X), which is an input variable, one level input (d) which is a level function of the present internal state variables (ys) and/or the level input variables (ls) of the D. T. L. C., and two different outputs (W_1 and W_2) depending on the nature of the level change of X. An occurrence of an upward transition signal at input terminal X of the K-element produces a transition signal at output terminal W_1, an occurrence of a downward transition signal at input terminal X produces a transition signal at output terminal W_2, while in both cases, the level input (d) should maintain at a constant level (d = 1).

It is noted (refer to Table 2.4) that for each time, only one output transition can occur. Whenever W_1 or W_2 has a transition occurred at its terminal, the other must remain at its previous value (1 or 0).

The K-element satisfies the above set of operating conditions, and the initial levels at input X and d as well as both outputs, are '0'. So far as transition concept is concerned, the transition logic representation of a K-element can be shown as Figure 2.1, and an example of the wave forms of its transition signals is given in Figure 2.2, together with a table (Table 2.4) illustrating the input-output relationships of the K-element.

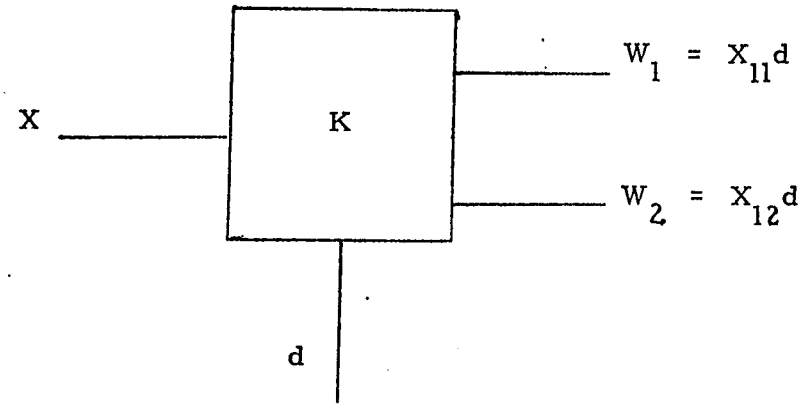


Figure 2 1

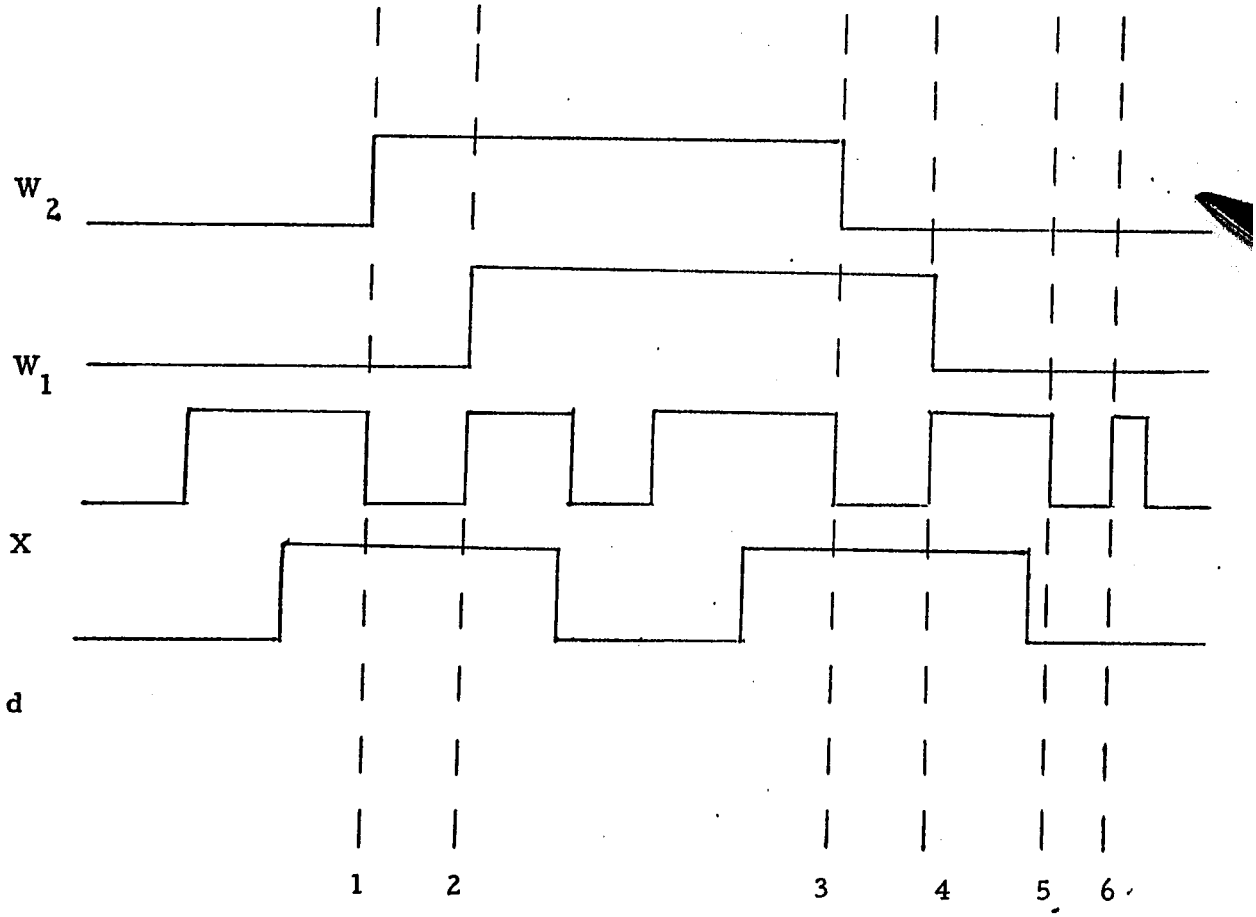


Figure 2.2

Referring to the table shown above, we realize that there are six cases to be considered:

Case 1

When d-input is at high level, and X-input changes from high to low (a downward transition), then, an upward transition will occur at the output terminal W_2, and W_2 will stay at that state until another downward transition occurs at X-terminal again. At the same time, W_1 has no change and stays at low level.

Case 2

When d-input is at high level, and X-input changes from low to high (an upward transition), then, an upward transition occurs at the output terminal W_1, and W_1 keeps on staying at that value, until another upward transition occurs at the X-input terminal, but W_2 maintains its previous state after its latest transition (in this cases, it maintains a high level, because it had an upward transition in case 1).

Case 3

When d-input is at high level, and X-input has a downward transition, then W_2 changes from high to low and W_1 stays at its previous state after its latest transition (in this case, it maintains a high level, because W_1 had an upward transition in case 2).

Case 4

When d-input is at high level, and X-input has an upward transition, then W_1 changes again (from high to low) and W_2 stays at its previous state after its latest transition (in this case, it maintains a low level, because W_2 had a downward transition in case 3).

Case 5 and Case 6

When d-input is at low level, neither W_1 nor W_2 would change, even X has a downward or upward transition, on the contrary both outputs remain at their previous state.

It is noted that, for all cases, only one output can be influenced and changed because of the transition of input X, and d-input has always to be kept at high level. In order to obtain the output transitions when K-element is in case 5 or case 6, an inverter has to be connected to the level input lead of the K-element which is shown in the following circuit diagram.

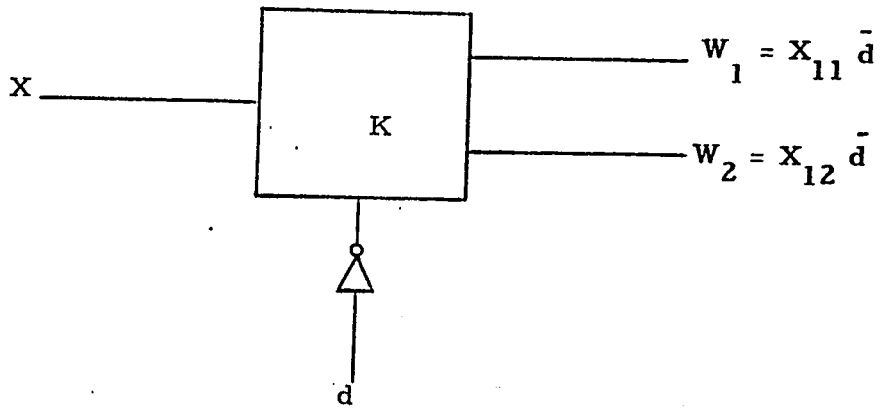


Figure 2.2 (a)

In conclusion, a K-element is actually a logic circuit with level inputs and level outputs in which one of its inputs (X) is treated as a transition input, for the reason that only the influence of this transition to the outputs is considered.

(B) Synthesis

(a) Synthesis of K-element with ordinary logic gates

As mentioned above, a transition circuit can be synthesized by Huffman's method and, therefore, it is natural that the K-element is also realizable with level logic synthesis method. The result is a level sequential circuit with level variables instead of transition variables as used in Figure 2.1.

Table 2.5 is a primitive level flow table, which describes the actual operating behaviour of the K-element considered as a level logic circuit. Checking the primitive level flow table, we discover it can be reduced further by state merging. Then, a reduced flow table (Table 2.6) with eight rows results. Carrying on the synthesis procedure of secondary assignment for asynchronous circuits (9), we can easily find a set of state assignments such that the transition between any two stable states is one shot and the Y-Matrix and Z vectors are obtained as in Table 2.7.

From Table 2.7, the minimal-sum-of-products representations of the excitation and output functions of K-element are derived, together with the level circuit diagrams as shown in Figure 2.3 (a).

TABLE 2.5 Primitive Level Flow Table of K-element

S \ dx	00	01	11	10	w_1	w_2
1	①	2	-	3	0	0
2	1	②	4	-	0	0
3	1	-	13	③	0	0
4	-	2	④	5	0	0
5	6	-	10	⑤	0	1
6	⑥	8	-	5	0	1
7	-	8	⑦	3	0	1
8	6	⑧	7	-	0	1
9	11	-	7	⑨	1	1
10	-	12	⑩	16	1	1
11	⑪	12	-	9	1	1
12	11	⑫	10	-	1	1
13	-	14	⑬	9	1	0
14	15	⑭	13	-	1	0
15	⑮	14	-	16	1	0
16	15	-	4	⑯	1	0

TABLE 2.6 Reduced Level Flow Table (A)

	00	01	11	10	w_1	w_2
A (1, 3)	①	2	13	③	0	0
B (2, 4)	1	②	④	5	0	0
C (5, 6)	⑥	8	10	⑤	0	1
D (7, 8)	6	⑧	⑦	3	0	1
E (9, 11)	⑪	12	7	⑨	1	1
F (10, 12)	11	⑫	⑩	16	1	1
G (13, 14)	15	⑭	⑬	9	1	0
H (15, 16)	⑮	14	4	⑯	1	0

TABLE 2.7 Reduced Level Flow Table (B)

		00	01	11	10	w_1	w_2
000	A (1, 3)	000	100	010	000	0	0
100	B (2, 4)	000	100	100	101	0	0
101	C (5, 6)	101	001	111	101	0	1
001	D (7, 8)	101	001	001	000	0	1
011	E (9, 11)	011	111	001	011	1	1
111	F (10, 12)	011	111	111	110	1	1
010	G (13, 14)	110	010	010	011	1	0
110	H (15, 16)	110	010	100	110	1	0

The excitation and output functions are :

$$Y_1 = \bar{d}\left[x \oplus (y_2 \oplus y_3)\right] + dy_1$$

$$= \bar{d}\left[x \oplus (y_2 \oplus y_3)\right] + dy_1 + \underline{y_1\left[x \oplus (y_2 \oplus y_3)\right]}$$

$$Y_2 = y_2\overline{(dx)} + (dx)\overline{(y_1 \oplus y_3)}$$

$$= y_2\overline{(dx)} + (dx)\overline{(y_1 \oplus y_3)} + \underline{y_2\overline{(y_1 \oplus y_3)}}$$

$$Y_3 = y_3\overline{(d\bar{x})} + (d\bar{x})\overline{(y_1 \oplus y_2)}$$

$$= y_3\overline{(d\bar{x})} + (d\bar{x})\overline{(y_1 \oplus y_2)} + \underline{y_3\overline{(y_1 \oplus y_2)}}$$

$$w_1 = y_2$$

$$w_2 = y_3$$

A preset signal has been inserted in the circuit of Figure 2.3 (a) in order to achieve the initial conditions which the K-element has to satisfy at the beginning of operation, that is, before the inputs (x or d) start to change, the K-element must be in the state of $x = 0$, $d = 0$, $w_1 = 0$, $w_2 = 0$.

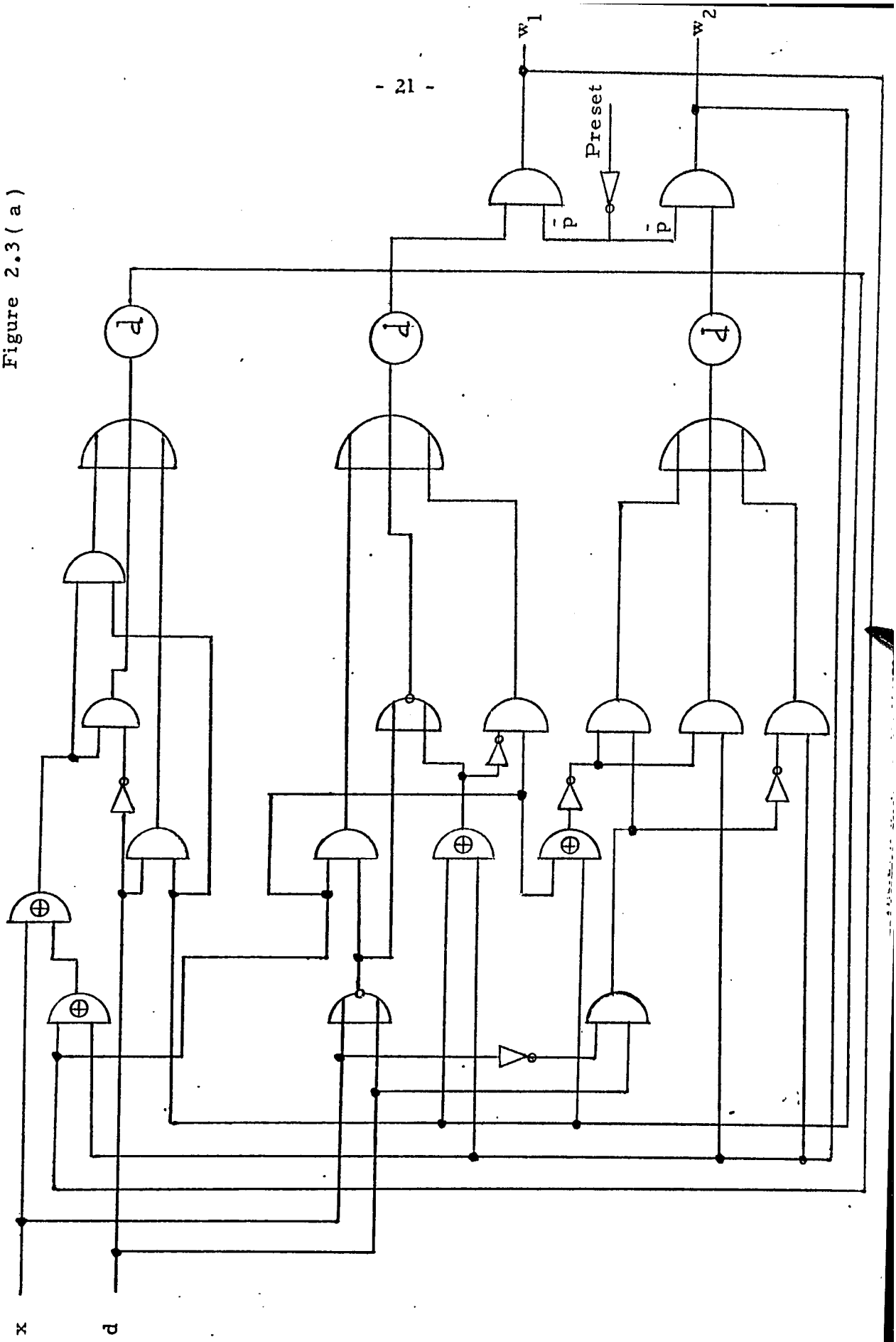


Figure 2.3 (a)

(b) Synthesis of K-element with wired-or gates

If we set $w_1 = y_2$, $w_2 = y_3$, $w_1' = Y_2$,

$w_2' = Y_3$, $y = y_1$, $Y = Y_1$,

and substitute them into those sets of excitation and output functions displayed above, we can get a set of expressions of slightly different form, from which a much more symmetric circuit configuration can be achieved.

The circuit is shown in Figure 2.3 (b), and the excitation and output functions are :

$$w_1' = dx\,\overline{(y \oplus w_2)} + w_1\left[\overline{(dx)} + \underline{\overline{(y \oplus w_2)}}\right]$$

$$w_2' = d\bar{x}\,(y \oplus w_1) + w_2\left[\overline{(d\bar{x})} + \underline{(y \oplus w_1)}\right]$$

$$Y = \bar{d}\left[x \oplus (w_1 \oplus w_2)\right] + y\left(d + \underline{\left[x \oplus (w_1 \oplus w_2)\right]}\right)$$

Referring to Figure 2.3 (b), the actual expressions for the outputs can be written as below, when a preset signal is used.

$$w_1' = \overline{\left(\left[dx\,\overline{(y \oplus w_2)}\right]\left[w_1\,(d\bar{x})\right]\left[w_1\,\overline{(y \oplus w_2)}\right]\right) + p}$$

$$= \bar{p}\left(dx\,\overline{(y \oplus w_2)} + w_1\,\overline{(dx)} + w_1\,\overline{(y \oplus w_2)}\right)$$

$$w_2' = \overline{\left(\left[d\bar{x}\,(y \oplus w_1)\right]\left[w_2\,\overline{(d\bar{x})}\right]\left[w_2\,(y \oplus w_1)\right]\right) + p}$$

$$= \bar{p}\left(d\bar{x}\,(y \oplus w_1) + w_2\,\overline{(d\bar{x})} + w_2\,(y \oplus w_1)\right)$$

The purpose of applying a preset signal P into the circuit is same as that explained before.

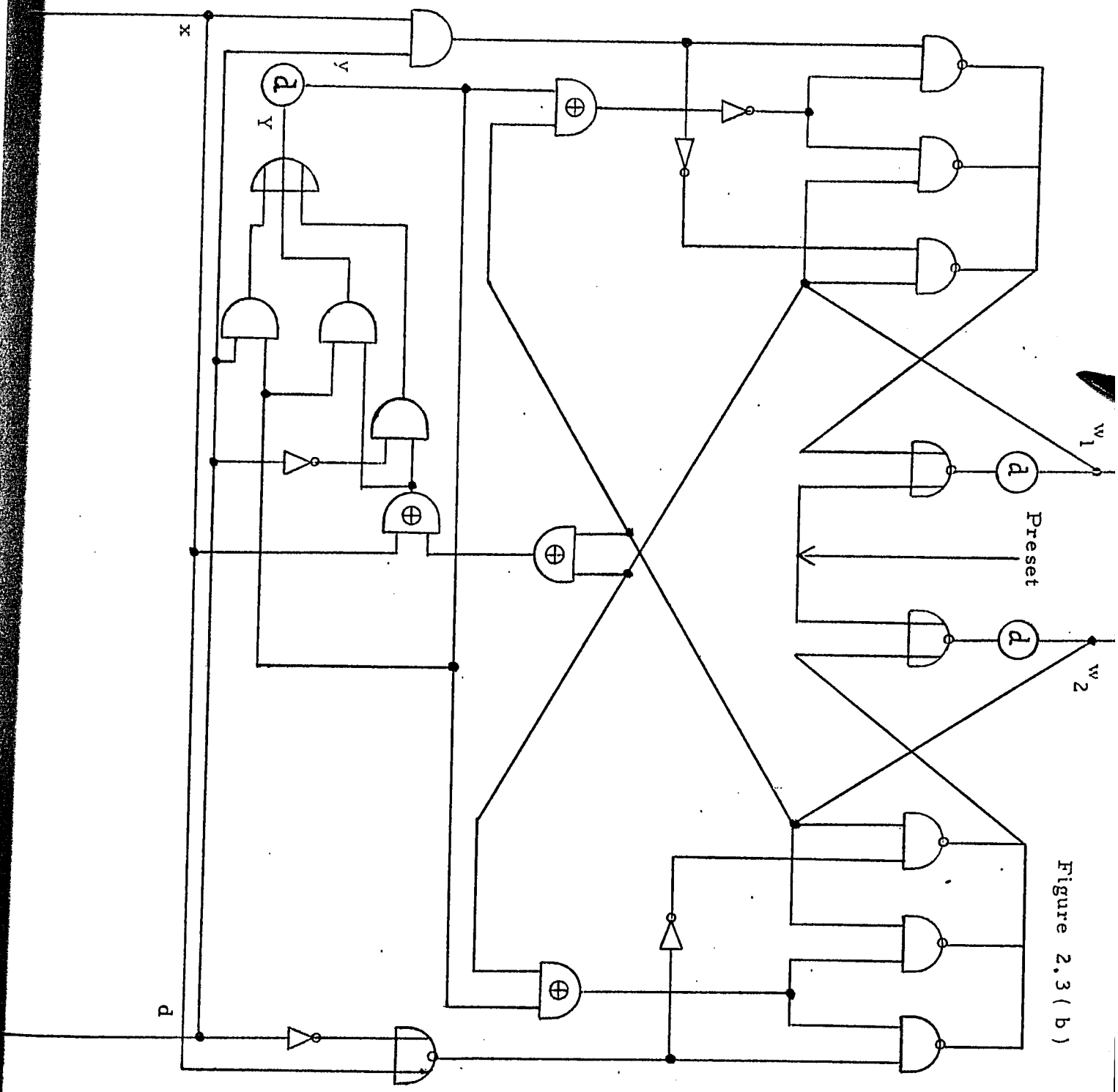


Figure 2.3 (b)

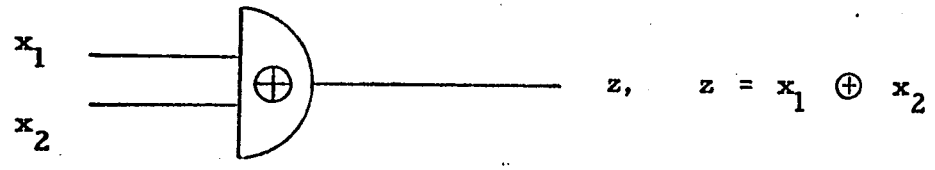
After knowing the circuit realization of the K-element and by inspecting the structure of the flow table for these two kinds of logic circuits, it is not hard to show that the flow table for transition logic circuit (1) is capable of being transformed into the flow table for directional transition logic circuit, by splitting each of the transition inputs of the transition flow table into its upward and downward transition signal form or vice versa. This shows that the concept of directional transition logic can always be applied to the problems where the concept of transition logic is applicable. There will be many 'don't care terms' in the directional transition flow tables because of this kind of transformation of tables. With the application of these don't care entries , a simpler circuit realization with relatively few interconnections between elements can be obtained, if the circuit is realized by K-elements instead of G-elements.

2.3 M-element (Transition Or-gate)

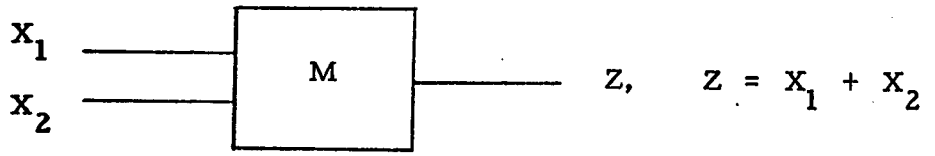
For the purpose of forming the Boolean sum of two or more transition signals, an 'M-element' was developed by Chuang (2). To show how to synthesize this element, let us take an M-element with two transition inputs and one transition output as shown in Figure 2.4 (b) as an example, and its characteristics can be stated as the following : 'The output (Z) of the M-element will produce a transition signal whenever there is a transition at either one of the two transition inputs'.

According to this verbal description, we can derive a level flow table for this M-element, and synthesize it by Huffman's method(6). The result of the circuit realization in this case is eventually an exclusive-or gate as shown in Figure 2.4 (a), and the output function of this M-element in level logic form is ' $z = x_1 \oplus x_2$ '. Since only one transition input change is permitted at each time, the output function of this M-element can be rewritten in transition level form as ' $Z = X_1 + X_2$ ', where the operation of Boolean addition is applicable. Referring to this equation, a black box representing this element in transition logic can be derived as in Figure 2.4 (b).

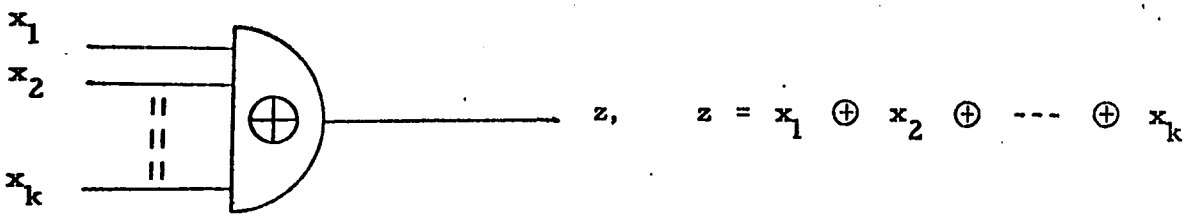
For the case of more than two transition inputs, the synthesis method of an M-element will be similar to the one used for the two inputs case described above. Because a modulo-2 summer (2) shown as in Figure 2.4 (c) satisfies the operating conditions of the M-element in this case, it can be used as the M-element to be designed. The level output function of the modulo-2 summer is . $z = x_1 \oplus x_2 \cdots \oplus x_k$ ', where k is the total number of transition inputs of the M-element. Similarly, the black box representing the M-element in transition logic can be shown as in Figure 2.4 (d) with the output function written in transition level form as ' $Z = \sum_{i=1}^{k} X_i$ '.



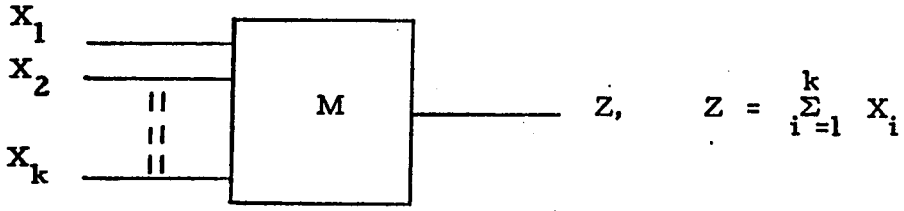
(a)



(b)



(c)



(d)

Figure 2.4 (a, b, c, d)

Referring to Figure 2.5, it should be noted that inputs of an M-element is non-directional transition output signals from the K-element, and the output of an M-element is either a transition signal representing an internal state variable y_i (level variable) when the M-element is used in the realization of excitation function, or a transition output Z_i when it is used in the realization of the output function, both of the directional transition circuit.

We have synthesized the M-element with restriction that at most one transition input change is permitted at each time. When the case of simultaneous transition input changes has to be considered, the M-element can be synthesized by applying the same technique. The method of solving this problem will be discussed briefly in Chapter 4.

2.4 General Circuit Configuration of Directional Transition Logic Circuit.

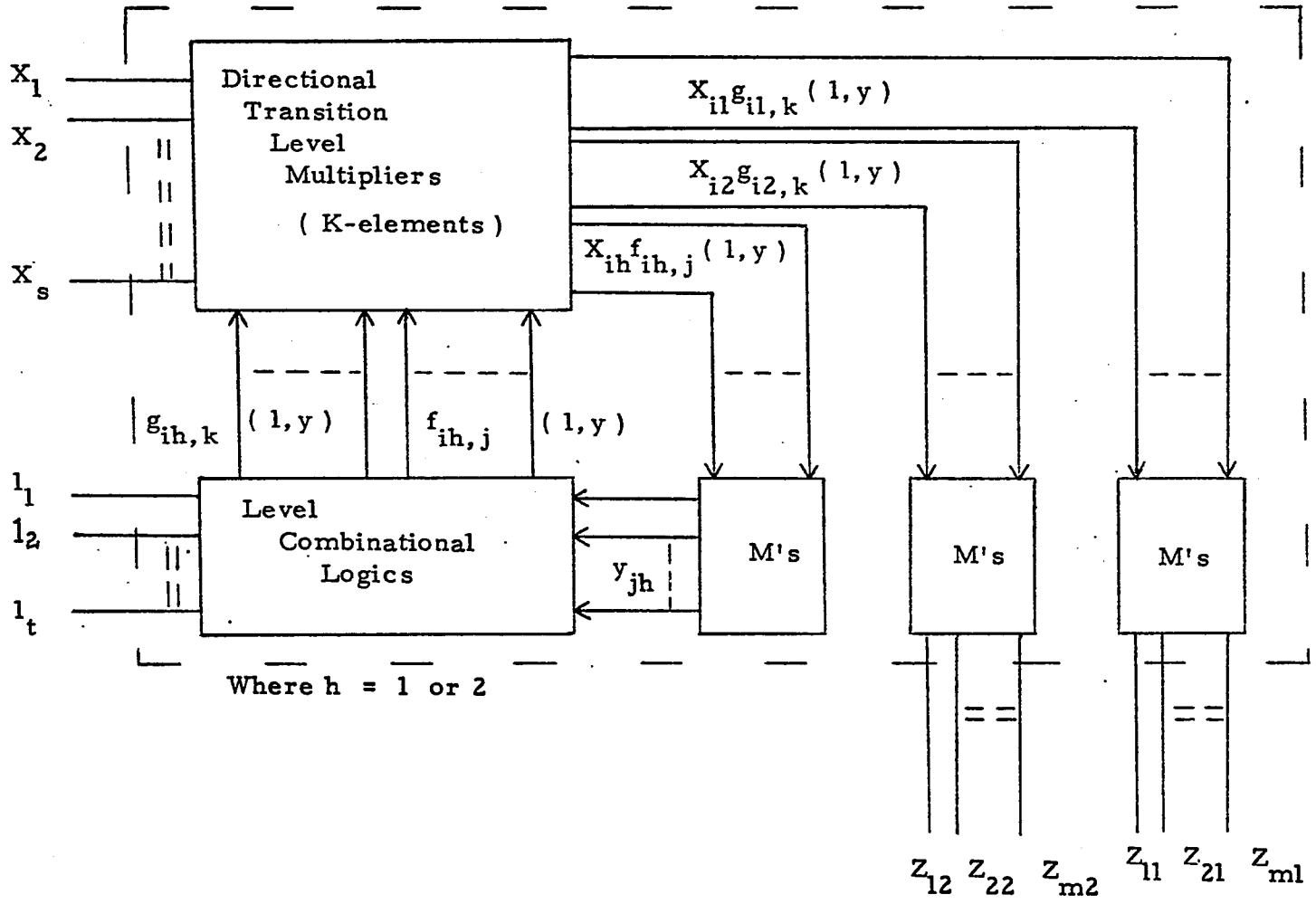
We have given a black box representation of the directional transition circuit in Figure 1.4 in Chapter 1. A more detailed description of this circuit model is given in this section.

The circuit configuration of D. T. L. C. in block diagram form is shown in Figure 2.5 . The block marked K-elements or the block marked M-elements is generally a network of K-elements or M-elements in parallel connections. The level functions $g_{i1,k}(1,y)$, $g_{i2,k}(1,y)$, $f_{i1,j}(1,y)$, and $f_{i2,j}(1,y)$ are generated by the block of level combinational logics.

In some situations where the effect does depend on the direction of transition (upward or downward), one block of M-elements in parallel configuration with the outputs from K s as its inputs, shown in Figure 2.5, can be saved. Therefore, only one set of outputs is presented in this case, this is Z_{k1} or Z_{k2} (k = 1. 2. . . , m.).

Also, it is noted that the set of level inputs of a transition circuit shown in Figure 1.1 or Figure 2.5 will not be used in the realization of an asynchronous circuit when the concept of transition logic or directional transition logic is applied.

Figure 2.5


X_1
X_2
X_s
Directional Transition Level Multipliers (K-elements)
$X_{il} g_{il,k}$ (1, y)
$X_{i2} g_{i2,k}$ (1, y)
$X_{ih} f_{ih,j}$ (1, y)
$g_{ih,k}$ (1, y)
$f_{ih,j}$ (1, y)
l_1
l_2
l_t
Level Combinational Logics
y_{jh}
M's
M's
M's
Where h = 1 or 2
Z_{12} Z_{22} Z_{m2} Z_{11} Z_{21} Z_{m1}

2.5 Hazards and Races.

Hazards and races are the causes of incorrect circuit operation encountered in asynchronous circuits. A transient hazard (7,14) is said to exist, if momentary false outputs appear at the output terminals of the circuit. An essential hazard (10,14) is said to occur in a fundamental-mode flow table, if and only if the circuit is initially in a total state S, the state reached by one change of a certain input (x) is different from the state reached after three consecutive changes in (x). Race (9) occurs in a flow table, if more than one state variable must change in the course of a transition.

If the final stable state reached as a result of the race depends on the order of change of the variable, the race is called critical. In order to eliminate these difficulties, many methods have been introduced recently (7,10,14,15).

The directional transition circuit (refer to Figure 2. 3 (a), 2. 3 (b) and Figure 2. 5) is free from hazard and race difficulties :

First , the outputs of each K-element are free from transient hazards (false outputs) because of adding the hazard removing terms (10), (those terms underlined) in its level logic circuit equations. In order to avoid the malfunctions due to conditions corresponding to essential hazards, three delay elements (pure or inertial delay) are inserted in the state branches of the K-element of Figure 2. 3 (a) and in the state branch (y and Y) and the feedback paths of w_1 and w_2 of the K-element of Figure 2. 3 (b). The only requirement these delays must meet is that they should be greater than the maximum response time of the K-element to its input signals. In other words, by using these delay units, the state variable change does not actually occur until the input change has been seen throughout the circuit, and then a proper operation is ensured. For avoiding race troubles, we have chosen such state assignment for the K-element

that only one state variable changes during each transiton (9,10). Also, because of the operation condition that a level change would occur at the transition input terminals only when the level signals from the combinational logics as the level inputs to each of the K-elements have been in stable state, then single level change is obtained and malfunctions due to races will not happen.

Second the outputs of the directional transition circuit are hazard free because the M-elements are hazard free. This is due to the fact that outputs of the M-elements are the transition outputs of the directional transition circuit (refer to Figure 2.5) and each of the M-elements never has a level change on more than one of its inputs at a time, and no adjacent input states produce the same output level. It is easy to show then (e.g. by Theorem 7.5-1 of McCluskey (10)), that each of the M-element is hazard free. Outputs of the D.T.L.C. are, therefore, hazard free.

Finally, only the stable level signals ($f_{ih,j}(l,y)$, $g_{ih,k}(l,y)$) generated by the combinational logics are considered as the level input signals to the K-elements. Therefore, momentary hazards appearing in $f_{ih,j}(l,y)$ and $g_{ih,k}(l,y)$ would not affect the operation of the D.T.L.C. and this concludes the hazard and race free properties of a directional transition circuit.

CHAPTER 3

REALIZATION OF DIRECTIONAL TRANSITION LOGIC CIRCUITS

3.1 The Synthesis Procedures

Asynchronous circuits can be realized by applying the concept of directional transition logic. Knowing the characteristics of the K-element, M-element, and the derivation of the flow table of D. T. L. C., the synthesis procedure for this kind of circuits which is similar to pulse sequential circuits synthesis method can be stated as follows.

Step 1. Transform the level flow table into directional transition flow table and reduce it with applicable merging techniques.

Step 2. Assign binary code to each of the present states, and next states to obtain the transition table and the output table.

Step 3. Perform the modulo-2 addition between the coded present state and all coded next states for each row of the transition table to obtain the excitation table.

Step 4. Derive excitation functions and output functions from the excitation table and the output table respectively.

These functions expressed in transition level form are ;

$$
\begin{aligned}
Z_{k1} &= \sum_{i=1}^{s} X_{i1}\, g_{i1,k}(1,y) \\
&\qquad\qquad , \quad k = 1, 2, 3, \ldots\ldots, m. \qquad (A) \\
Z_{k2} &= \sum_{i=1}^{s} X_{i2}\, g_{i2,k}(1,y)
\end{aligned}
$$

$$
\begin{aligned}
Y_{j1} &= \sum_{i=1}^{s} X_{i1}\, f_{i1,j}(1,y) \\
&\qquad\qquad , \quad j = 1, 2, 3, \ldots\ldots\ldots, r. \qquad (B) \\
Y_{j2} &= \sum_{i=1}^{s} X_{i2}\, f_{i2,j}(1,y)
\end{aligned}
$$

where ' Σ ' means Boolean sum

' s ', ' r ', and ' m ' are the total number of transition inputs, internal state variables and outputs respectively.

In above equations, X_{i1} and X_{i2} are the upward and downward transitions of X_i , Z_{k1} or Z_{k2} is the k^{th} output variable (Z_k) which responds to X_{i1} or X_{i2} respectively. The j^{th} excitation variable Y_{j1} or Y_{j2} (a transition variable) represents the change of the j^{th} internal state variable y_{j1} or y_{j2} (a level variable). $Y_{j1} = 1$ or $Y_{j2} = 1$ implies a change of y_{j1} or y_{j2}.

$f_{i1,j}(1,y)$ or $f_{i2,j}(1,y)$ and $g_{i1,k}(1,y)$ or $g_{i2,k}(1,y)$ represent level signals and are Boolean functions of level input variables (1's) and internal states variables (y's). $f_{i1,j}(1,y)$ is obtained from the j^{th} component of entries in the X_{i1} subtable, $f_{i2,j}(1,y)$ is obtained from the j^{th} component of entries in the X_{i2} subtable, both of the excitation table.

Similarily, $g_{i1,k}(1,y)$ and $g_{i2,k}(1,y)$ are obtained from k^{th} component of entries in the X_{i1} and X_{i2} subtable of the output table.

If $Z_{k1} = 0$ or $Z_{k2} = 0$, then the result will be a transition

circuit which does depend on one direction of transition of the input signals.

If $Z_{k1} \neq 0$ and $Z_{k2} \neq 0$, then the outputs of this circuit depend on both kinds of transition of transition inputs and, in this case, an M-element can be used to combine Z_{k1} and Z_{k2} together, such that equation (A) becomes $Z_k = Z_{k1} + Z_{k2}$.

Step 5. Construct a circuit according to the excitation functions and output functions.

3.2 Synthesis Example

Example 1 (cont'd) Let us return to example 1 of the modulo-four-counter problem; the counter starts to operate from stable state ① (Table 2.2), when X_1 changes from 0 to 1. The counter will go to stable state ② ; therefore, an entry of row 1 in subtable X_{11} of Table 2.3 is filled with the state number ' 2 ' indicating the state transition. Also during this state transition the outputs change from 00 to 01. This indicates that during this state change, only output Z_2 has transition. So, beside the state number ' 2 ' in row 1 of Table 2.3, we write down 01 representing the outputs transition when this state change happens. 0 indicates that Z_1 has no transition, 1 denotes that Z_2 has transition.

On the other hand, if X_2 changes first from 0 to 1 , when the counter is in state ① then the counter will go to state ⑨ , and during this state change, the outputs transition become 11, because both outputs would change from 0 to 1 , according to Table 2.2. With the same reasoning for other state transitions, we obtain the transition Table 2.3.

Step 1. Checking with Table 2.3, we can see that this table can be reduced further by applying the implication graph technique and state merging. Thus a transition table with four rows

instead of twelve is obtained as shown (Table 3.1 and Table 3.2).

Step 2. Since Table 3.2 has four rows exactly, two state variables have to be assigned. The result is shown in Table 3.3.

Step 3. Performing the modulo-2 addition among each coded present state with all coded next states in the same row, an excitation and output table is obtained (Table 3.4).

Step 4. Referring to Table 3.4, we obtain the excitation functions and output functions of the counter.

Step 5. According to the excitation functions and output functions, we get the circuit realization with K-elements, shown in Figure 3.1.

Table 3.1 Reduced Directional Transition Flow Table (A)

	X_1		X_2	
	X_{11}	X_{12}	X_{21}	X_{22}
A (1,5)	B, 01	-	E, 11	-
B (2,6)	-	C, 00	-	-
C (3,7)	D, 11	-	F, 01	-
D (4,8)	-	A, 00	-	-
E (9,11)	-	-	-	C, 00
F (10,12)	-	-	-	A, 00

Table 3.2 Reduced Directional Transition Flow Table (B)

	X_1		X_2	
	X_{11}	X_{12}	X_{21}	X_{22}
I (A, B)	I, 01	II, 00	III, 11	-
II (C, D)	II, 11	I, 00	IV, 01	-
III (E)	-	-	-	II, 00
IV (F)	-	-	-	I, 00

Table 3.3 Secondary Assignment

		X_1		X_2	
		X_{11}	X_{12}	X_{21}	X_{22}
00	I (A, B)	00	01	11	-
01	II (C, D)	01	00	10	-
11	III (E)	-	-	-	01
10	IV (F)	-	-	-	00

Table 3.4 Excitation and Output Table

		X_1		X_2	
		X_{11}	X_{12}	X_{21}	X_{22}
00	I (A, B)	00, 01	01, 00	11, 11	-
01	II (C, D)	00, 11	01, 00	11, 01	-
11	III (E)	-	-	-	10, 00
10	IV (F)	-	-	-	10, 00
		Y_1Y_2, Z_1Z_2	Y_1Y_2, Z_1Z_2	Y_1Y_2, Z_1Z_2	Y_1Y_2, Z_1Z_2

Excitation Functions and Output functions of the Modulo-Four Counter

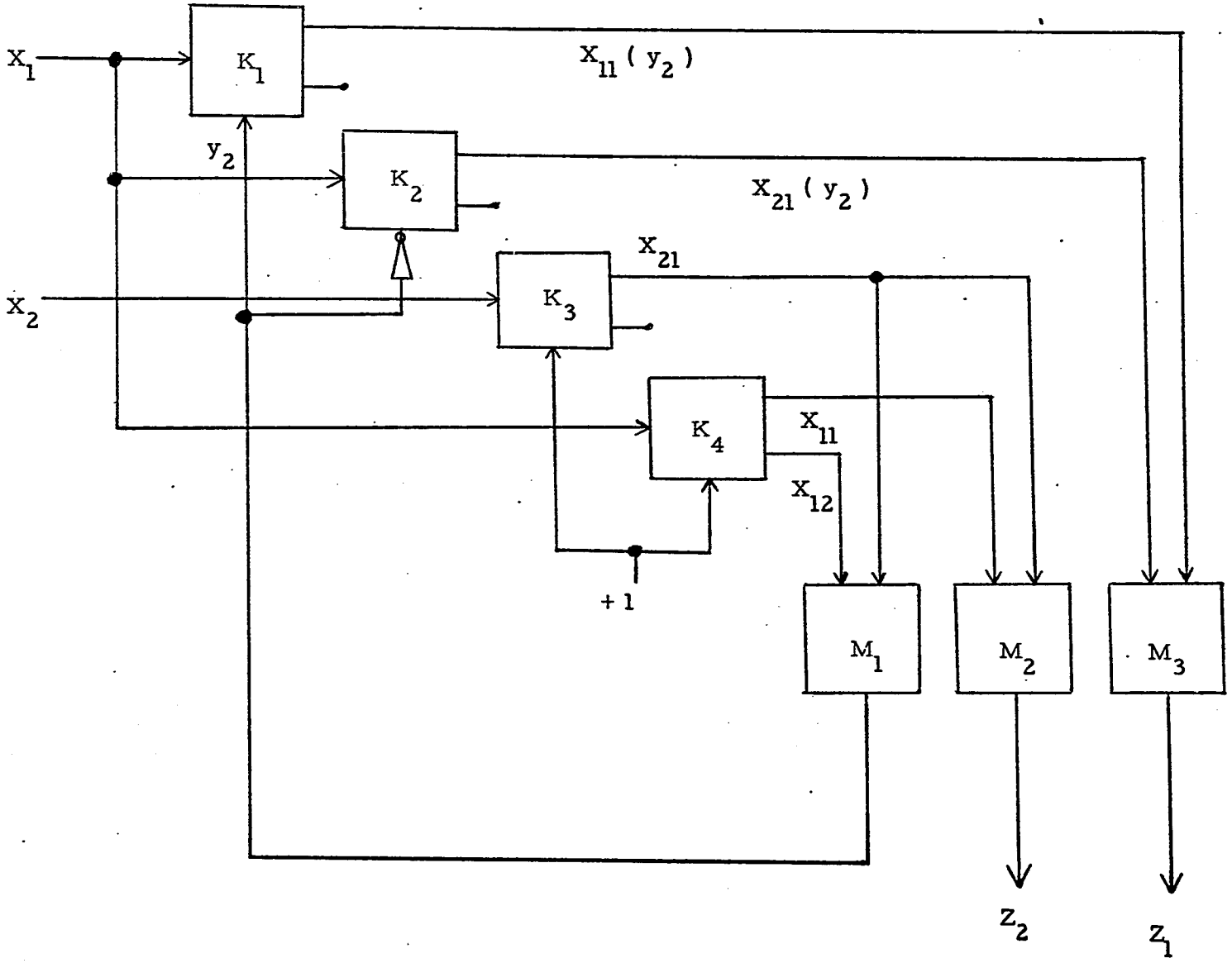
$$Y_1 = X_{21} + X_{22} = X_2$$

$$Y_2 = X_{12} + X_{21}$$

$$Z_1 = X_{11}(y_2) + X_{21}(\bar{y}_2)$$

$$Z_2 = X_{11} + X_{21}$$

Figure 3.1 Circuit Realization of the Modulo-four binary counter



Example 2. A level flow table is given as follows.

	00	01	11	10	z
1	①	2	-	5	0
2	-	②	3	-	0
3	-	4	③	8	0
4	1	④	-	-	1
5	-	-	6	⑤	0
6	-	7	⑥	8	0
7	1	⑦	-	-	0
8	1	-	-	⑧	0

Step 1. Take the transition inputs X_1 and X_2 of both x_1 and x_2 and the transition output Z of z.

	X_1		X_2	
	X_{11}	X_{12}	X_{21}	X_{22}
1	5, 0	-	2, 0	-
2	3, 0	-	-	-
3	-	4, 1	-	8, 0
4	-	-	-	1, 1
5	-	-	6, 0	-
6	-	7, 0	-	8, 0
7	-	-	-	1, 0
8	-	1, 0	-	-

Step 2. Secondary Assignment.

		X_1		X_2	
		X_{11}	X_{12}	X_{21}	X_{22}
000	1	010	-	110	-
110	2	101	-	-	-
101	3	-	001	-	100
001	4	-	-	-	000
010	5	-	-	111	-
111	6	-	011	-	100
011	7	-	-	-	000
100	8	-	000	-	-

Step 3. Perform the Modulo-2-addition.

		X_1		X_2	
		X_{11}	X_{12}	X_{21}	X_{22}
000	1	010, 0	1-- -	110, 0	--1 1
110	2	011, 0	1-- 1	1-1 -	- 11 -
101	3	-11 -	100, 1	1-- -	001, 0
001	4	-11 -	1-- 1	11- -	001 1
010	5	-11 -	1-- 1	101, 0	-11 -
111	6	-1- -	100, 0	1-1 -	011 0
011	7	-1- -	1-- -	1-1 -	011 0
100	8	-1- -	100, 0	1-- -	--1 -

Step 4. The excitation and output functions can be expressed as below.

$$Y_1 = X_{12} + X_{21}$$

$$Y_2 = X_{11} + X_{21}f_{21,2}(1,y) + X_{22}f_{22,2}(1,y)$$

$$Y_3 = X_{11}f_{11,3}(1,y) + X_{21}f_{21,3}(1,y) + X_{22}$$

$$Z_1 = X_{12}g_{12,1}(1,y) + X_{22}g_{22,1}(1,y)$$

where $f_{21,2}(1,y) = \bar{y}_1\bar{y}_2$

$$f_{22,2}(1,y) = y_2$$

$$f_{11,3}(1,y) = \bar{y}_2y_3 + y_2\bar{y}_3$$

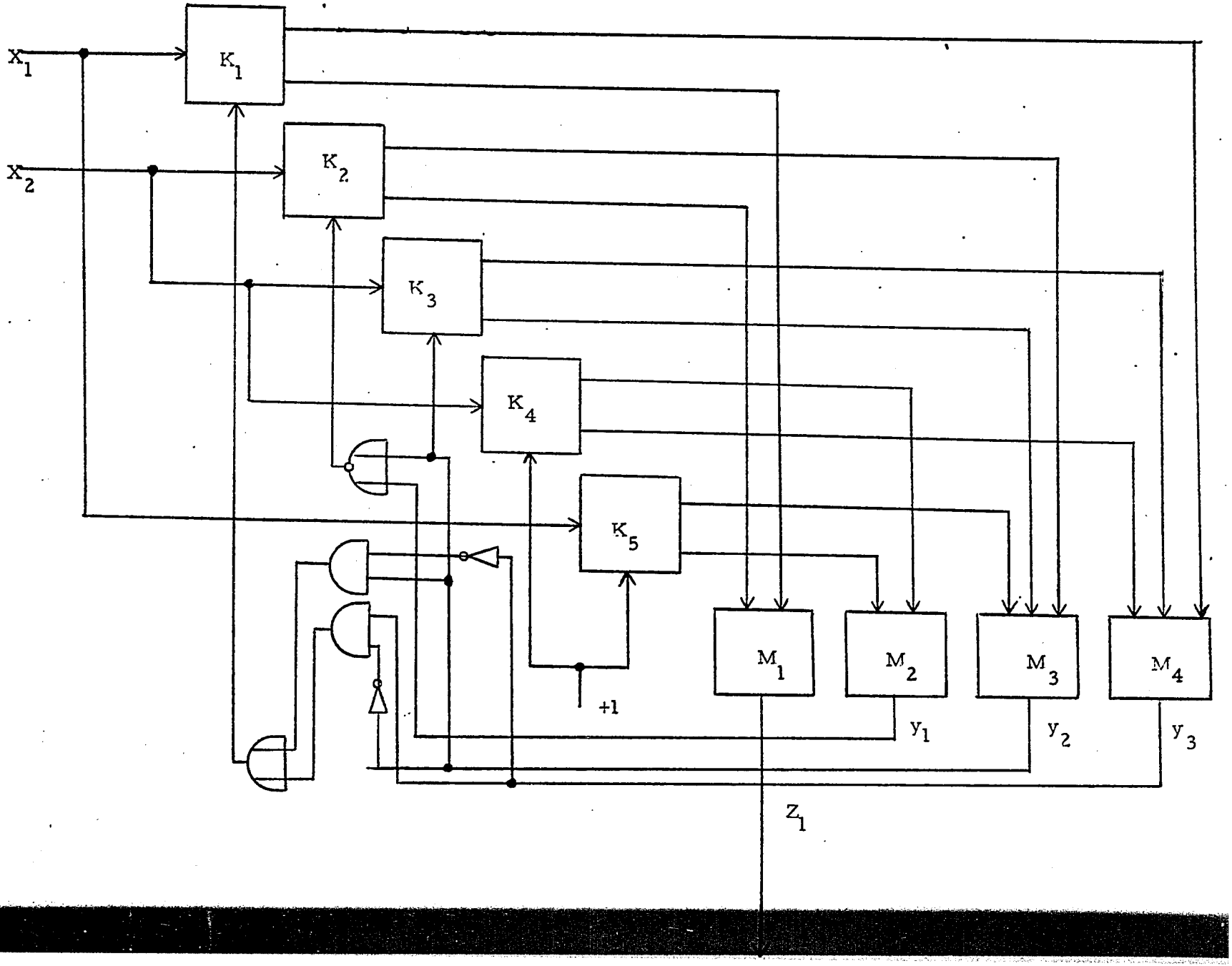
$$f_{21,3}(1,y) = y_2$$

$$g_{12,1}(1,y) = \bar{y}_2y_3 + y_2\bar{y}_3$$

$$g_{22,1}(1,y) = \bar{y}_1\bar{y}_2$$

In this example the $f_{ih,j}(1,y)$ and $g_{ih,k}(1,y)$ functions are Boolean functions of state variables alone, because no input signals at all have been selected as level inputs. Also, it should be noted that in this example the merging technique for flow table reduction is no longer applied, because it is not always guaranteed that a simpler circuit realization can be achieved by applying the state merging method.

Step 5. Circuit Realization



CHAPTER 4

CONCLUSIONS

In this thesis, we have presented an approach to the design of asynchronous circuits by applying the concept of directional transition logic, in an attempt to eliminate race and hazard difficulties and bring closer the synchronous and asynchronous switching theories.

A directional transition circuit has been shown as a circuit constructed from directional transition level multipliers, transition or-gates, and ordinary logic gates. With the application of K-elements, a directional transition circuit whose output changes responding to different level input changes is obtained.

Both G-element and K-element can be used as a module to realize an asynchronous problem and because of the present integrated circuit technology, it is believable that these elements can be made as a single logic element.

Although, a K-element is more complicated than a G-element due to its response to directional change of inputs rather than to simple change of input but a simple circuit realization with fewer interconnections between elements is always possible as a result of the using of K-elements.

We have also shown that race and hazard problems are considered and solved at the time of synthesizing the K-element, and only stable signals representing $g_{ih,k}(l,y)$ and $f_{ih,j}(l,y)$ functions would exist, whenever a transition input change happens at the input terminal. Therefore, simultaneous change of level inputs of the circuit are permissible and would not cause incorrect circuit operation.

A table (Table A-1) has been given in the Appendix as a summary to point out the different results of the same problem by using the two methods, one employing K-elements, and the other employing G. elements.

From that table, we can see that for some cases, the directional transition logic synthesis method is better than the transition logic synthesis method in that the number of elements used is reduced.

The use of 'Directional Transition Logic Method' for the synthesis of asynchronous circuit has the following advantages.

(1) The circuit realization is simple and straightforward.

(2) The race and hazard problems need not be considered, because these difficulties have been eliminated in the design of the K-element.

(3) It is able to distinguish the different transition input changes which affect the transition output changes of a circuit by applying this method.

(4) We have shown in Appendix that $X_1 d = X_{11} d + X_{12} d$ and $X_1 \bar{d} = X_{11} \bar{d} + X_{12} \bar{d}$, where the terms representing directional transition signals on the right hand side of these equations are independent of each other, and can be combined together by an M-element to form a nondirectional transition signal shown as the term on the left hand side of the equations. This indicates that all kinds of transition signal can be obtained, if K-elements are used. Therefore, the method suggested in this thesis is applicable to transition problems with situation where either the effect does depend on the direction of transition or the effect does not depend on the direction of transition.

(5) The approach introduced in this thesis can be extended to the case of multiple transition input changes. The way of solving this problem is to design the M-element by first obtaining its primitive level flow table for multiple input changes case, and then applying the synthesis method which is applicable to those realizations permitting multiple input changes (15). The circuit realization of this M-element will be an asynchronous circuit, whose circuit complexity is increased as the number of inputs increases. It is important to note that the circuit realization of the K-element would not alter, because the operation of a K-element is not affected when multiple input changes occur.

APPENDIX

A- 1 Relation between G-element and K-element

From the definitions of G-element and K-element, we know that two K-elements in parallel connection, incorporated with an inverter and two M-elements (Figure A. 2) would function as good as a G-element (Figure A. 1), and the outputs of this network respond to different types of level change of X, and the value of d or $\bar{d}$.

Since K_1 and K_2 are directional transition multiplier, their parallel connection which realizes a G-element can be regarded as a directional G-element as well (refer to A- 2).

Similarly, a K-element can also be realized by using two G-elements and an M-element as well as logic gates. This will be shown in Sec A-2 and Sec. A-2'.

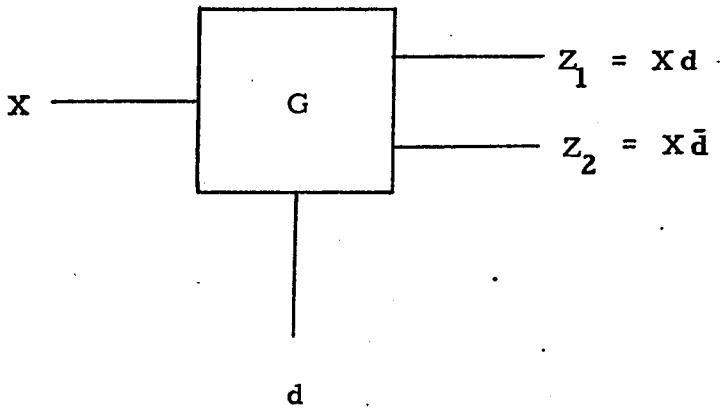


Figure A. 1

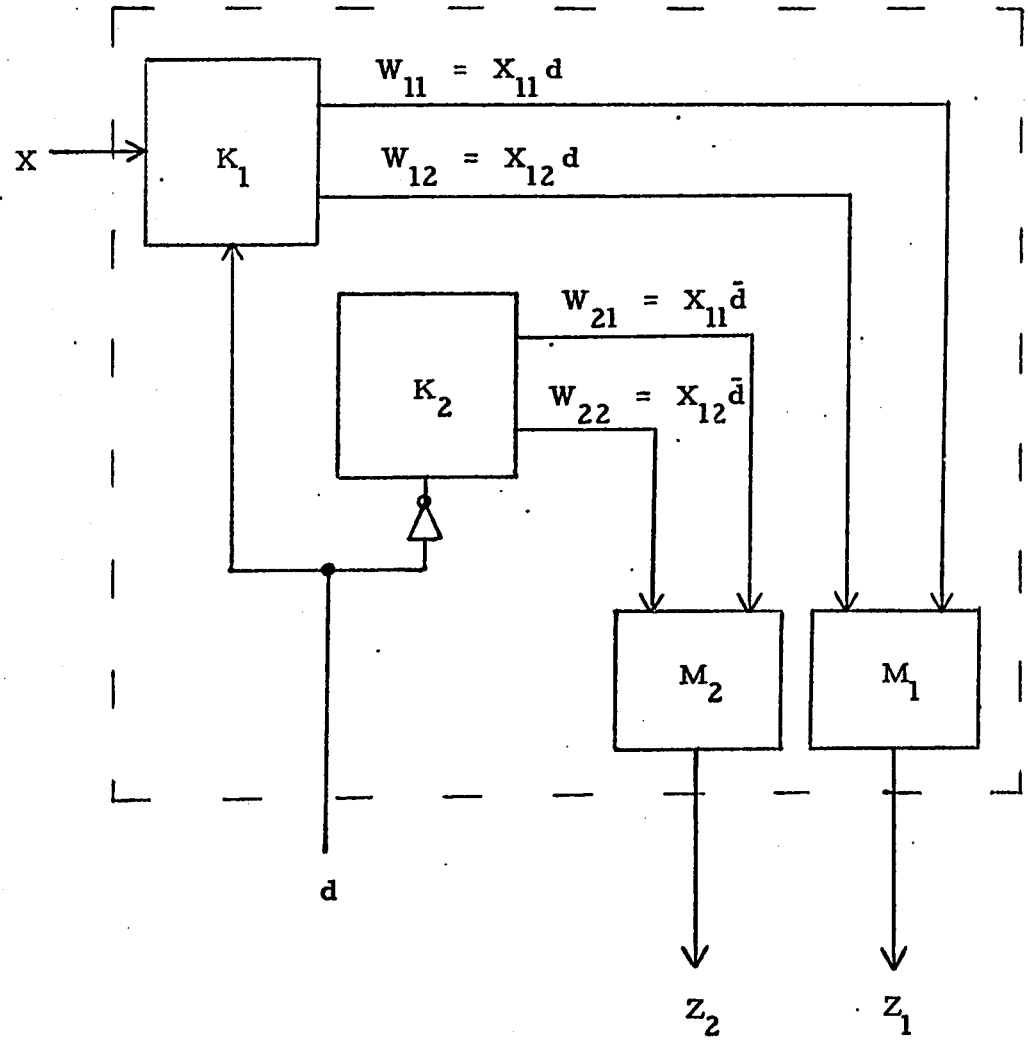


Figure A. 2

A- 2. Synthesis of G-element with K-elements.

The level flow table of the G-element (1) is given below.

	d = 0		d = 1			
	00	01	11	10	z_2	z_1
1	①	2	-	3	0	0
2	1	②	4	-	0	1
3	1	-	5	③	0	0
4	-	2	④	6	0	1
5	-	7	⑤	3	1	0
6	8	-	4	⑥	1	1
7	8	⑦	5	-	1	0
8	⑧	7	-	6	1	1

Step 1. Treat x as transition input X, and d as a level input

*	X			
	X_{11}		X_{12}	
	$\bar{d}$	d	$\bar{d}$	d
1	2, 01	-	-	-
2	-	-	1, 01	-
3	-	5, 10	-	-
4	-	-	-	6, 10
5	-	-	-	3, 10
6	-	4, 10	-	-
7	-	-	8, 01	-
8	7, 01	-	-	-

* Because only those state changes caused by the level change of transition input are considered, the state change caused by level change of level input (i.e. d) in the original level flow table is not shown in the directional transition flow table and those entries in the directional transition flow table for this type of state changes are treated as 'don. t care' terms.

Reduced Transition Flow Table

	X			
	X_{11}		X_{12}	
	$\bar{d}$	d	$\bar{d}$	d
A (1, 8)	B, 01	-	-	-
B (2, 7)	-	-	A, 01	-
C (3, 6)	-	D, 10	-	-
D (4, 5)	-	-	-	C, 10

Step 2 Secondary Assignment

		X			
		X_{11}		X_{12}	
		$\bar{d}$	d	$\bar{d}$	d
00	A (1, 8)	01	-	-	-
01	B (2, 7)	-	-	00	-
11	C (3, 6)	-	10	-	-
10	D (4, 5)	-	-	-	11

Step 3 Excitation and Output Table

		X			
		X_{11}		X_{12}	
		$\bar{d}$	d	$\bar{d}$	d
00	A (1, 8)	01, 01	-	-	-
01	B (2, 7)	-	-	01, 01	-
11	C (3, 6)	-	01, 10	-	-
10	D (4, 5)	-	-	-	01, 10

Step 4. Excitation and Output Functions

$$Y_1 = 0$$

$$Y_2 = \bar{d}\, X_{11} + d\, X_{11} + \bar{d}\, X_{12} + d\, X_{12}$$

$$= (\bar{d} + d)\, X_{11} + (\bar{d} + d)\, X_{12}$$

$$= X$$

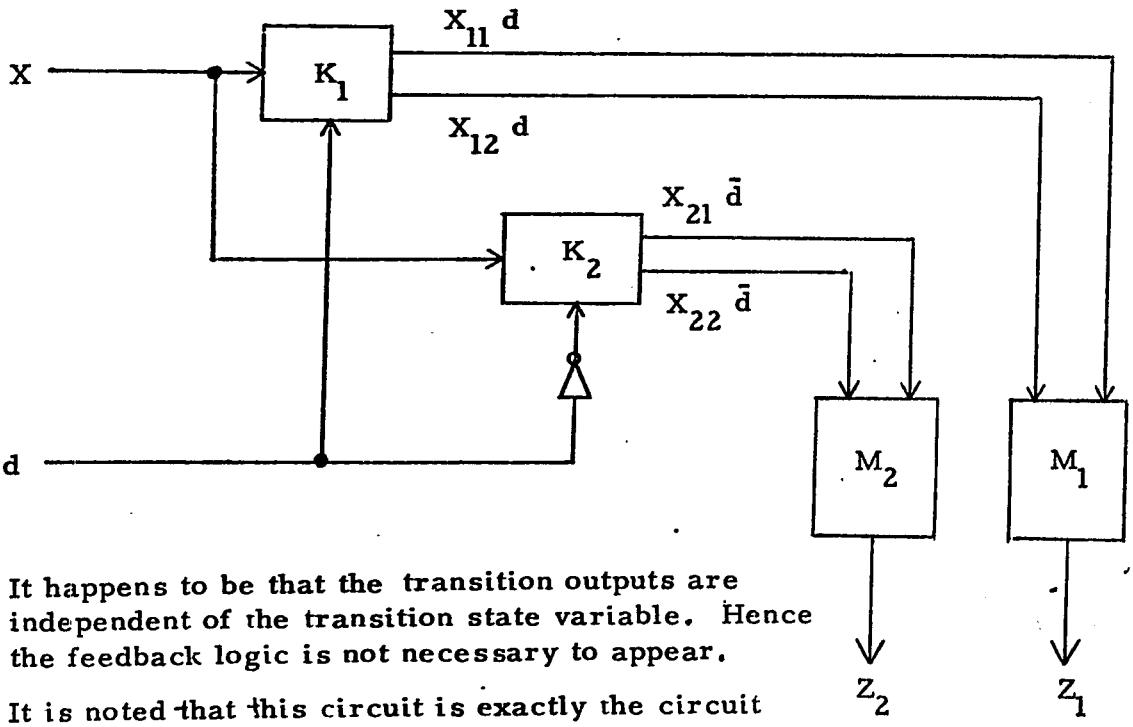
$$Z_1 = d\, X_{11} + d\, X_{12}$$

$$= X\, d$$

$$Z_2 = \bar{d}\, X_{11} + \bar{d}\, X_{12}$$

$$= X\, \bar{d}$$

Step 5. Circuit Realization



It happens to be that the transition outputs are independent of the transition state variable. Hence the feedback logic is not necessary to appear.

It is noted that this circuit is exactly the circuit shown in Figure A.2. Therefore the relation of these two elements is proved.

A-2' Synthesis of K-element with G-elements

The level flow table of a K-element is given below.

	d = 0		d = 1			
	00	01	11	10	w_1	w_2
1	(1)	2	-	3	0	0
2	1	(2)	4	-	0	0
3	1	-	13	(3)	0	0
4	-	2	(4)	5	0	0
5	6	-	10	(5)	0	1
6	(6)	8	-	5	0	1
7	-	8	(7)	3	0	1
8	6	(8)	7	-	0	1
9	11	-	7	(9)	1	1
10	-	12	(10)	16	1	1
11	(11)	12	-	9	1	1
12	11	(12)	10	-	1	1
13	-	14	(13)	9	1	0
14	15	(14)	13	-	1	0
15	(15)	14	-	16	1	0
16	15	-	4	(16)	1	0

Step 1. (a) Treat both inputs and outputs as transition inputs and transition outputs then form a transition flow table (1) of the K-element and as shown below.

Transition flow table of a K-element

	D		X	
1	3,	00	2,	00
2	4,	00	1,	00
3	1,	00	13,	10
4	2,	00	5,	01
5	6,	00	10,	10
6	5,	00	8,	00
7	8,	00	3,	01
8	7,	00	6,	00
9	11,	00	7,	10
10	12,	00	16,	01
11	9,	00	12,	00
12	10,	00	11,	00
13	14,	00	9,	01
14	13,	00	15,	00
15	16,	00	14,	00
16	15,	00	4,	10

(b) Reduce the transition flow table by finding a minimal closed collection of compatibles (15) that cover all the states of the transition flow table.

A complete, minimal, closed collection of compatibles is given below. The reduced transition flow table of a K-element corresponding to the collection is shown in Table A-2'.

I = (1,6,11,15)
II = (2,8,12,14)
III= (3,5,9,16)
IV= (4,7,10,13)

Table A-2' Reduced transition flow table of the K-element.

	D		X	
I	III,	00	II,	00
II	IV,	00	I,	00
III	I,	00	IV,	10
IV	II,	00	III,	01

Step 2 Secondary Assignment

		D	X
00	I	11	01
01	II	10	00
11	III	00	10
10	IV	01	11

Step 3 Excitation and output table

		D		X	
00	I	11,	00	01,	00
01	II	11,	00	01,	00
11	III	11,	00	01,	00
10	IV	11,	00	01,	00
		Y_1Y_2	Z_1Z_2	Y_1Y_2	Z_1Z_2

Step 4 Excitation functions and output functions of the K-element.

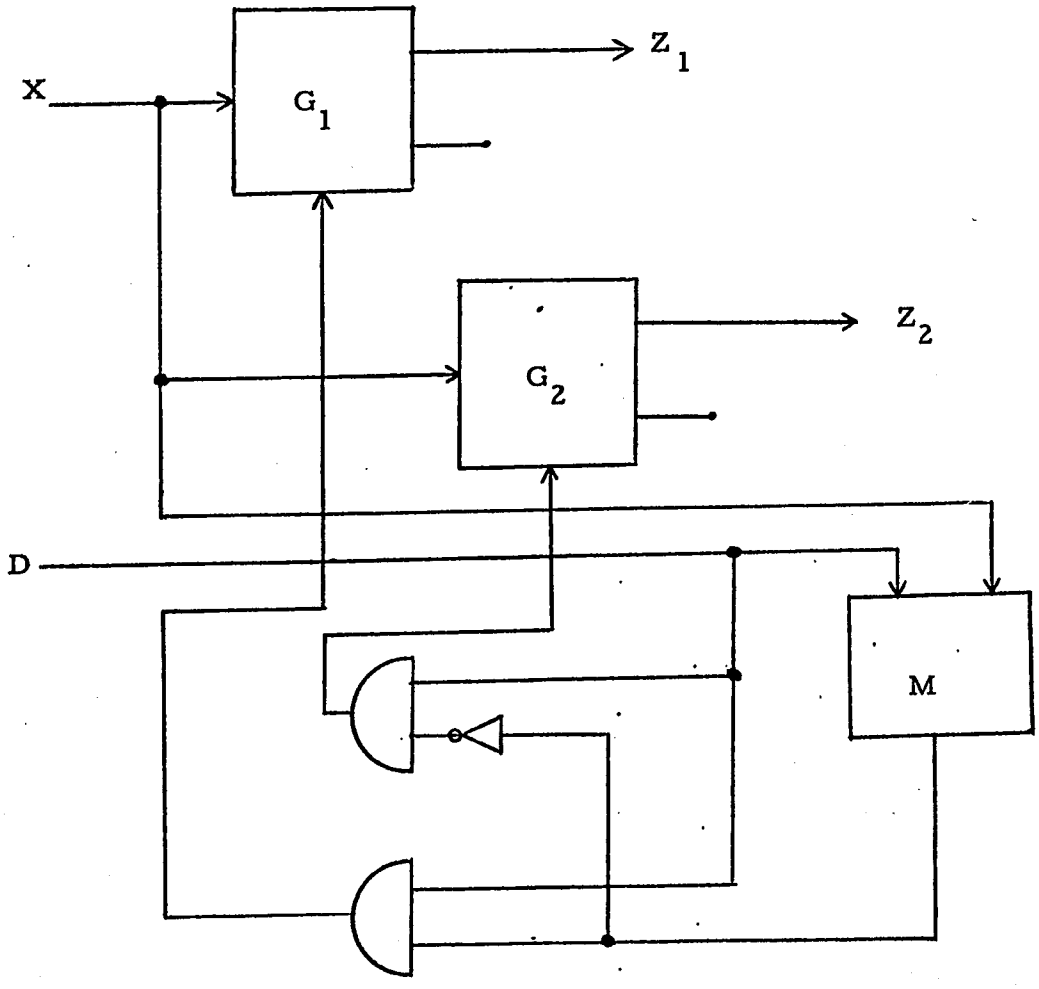
$$Y_1 = D$$

$$Y_2 = D + X$$

$$Z_1 = y_1 y_2 X$$

$$Z_2 = y_1 \bar{y}_2 X$$

Step 5 Circuit realization of the K-element with G-elements.



A - 3 Circuit Realizations with G-elements

In order to illustrate the differences between the application of G-element and K-element, we repeat the synthesis of each of those examples shown in Sec.3.2. First we obtain the transition flow table from the given original level flow table. Next, we apply the same state assignment which had been used for K-element realization in 3.2 to the problem. Then, we do the modulo-two addition to get the excitation and output table. Finally, we derive the excitation and output functions in transition-level form from the excitation and output table, together with the circuit realized with G-elements, M-elements and ordinary logic gates. At the end of this section, a table will be given to show the different results of circuit realization with G-elements and K-elements of these examples.

Example 1' (repeat the synthesis of example 1).

Level Flow Table

	00	01	11	10	z_1	z_2
1	①	9	-	2	0	0
2	3	-	-	②	0	1
3	③	12	-	4	0	1
4	5	-	-	④	1	0
5	⑤	11	-	6	1	0
6	7	-	-	⑥	1	1
7	⑦	10	-	8	1	1
8	1	-	-	⑧	0	0
9	7	⑨	-	-	1	1
10	5	⑩	-	-	1	0
11	3	⑪	-	-	0	1
12	1	⑫	-	-	0	0

Transition Flow Table

	X_1	X_2
1	2, 01	9, 11
2	3, 00	- -
3	4, 11	12, 01
4	5, 00	- -
5	6, 01	11, 11
6	7, 00	- -
7	8, 11	10, 01
8	1, 00	- -
9	- -	7, 00
10	- -	5, 00
11	- -	3, 00
12	- -	1, 00

Step 1 Simplified Transition Flow Table A (By combining equivalent state)

	X_1		X_2	
A (1, 5)	B,	01	E,	11
B (2, 6)	C,	00	-	-
C (3, 7)	D,	11	F,	01
D (4, 8)	A,	00	-	-
E (9, 11)	-	-	C,	00
F (10, 12)	-	-	A,	00

Simplified Transition Flow Table B (By state merging)

	X_1		X_2	
A (1, 5)	B	01	E	11
B (2, 6) E (9, 11)	C	00	C	00
C (3, 6)	D	11	F	01
D (4, 8) F (10, 12)	A	00	A	00

Step 2. Secondary Assignment

		X_1	X_2
00	I (A)	01	01
01	II (B F)	11	11
11	III (C)	10	10
10	IV (D F)	00	00

Step 3 Excitation and Output Table

		X_1		X_2	
00	I (A)	01	01	01	11
01	II (B,E)	10	00	10	00
11	III (C)	01	11	01	01
10	IV (D,F)	10	00	10	00
		Y_1Y_2	Z_1Z_2	Y_1Y_2	Z_1Z_2

Step 4. Excitation and Output Functions :

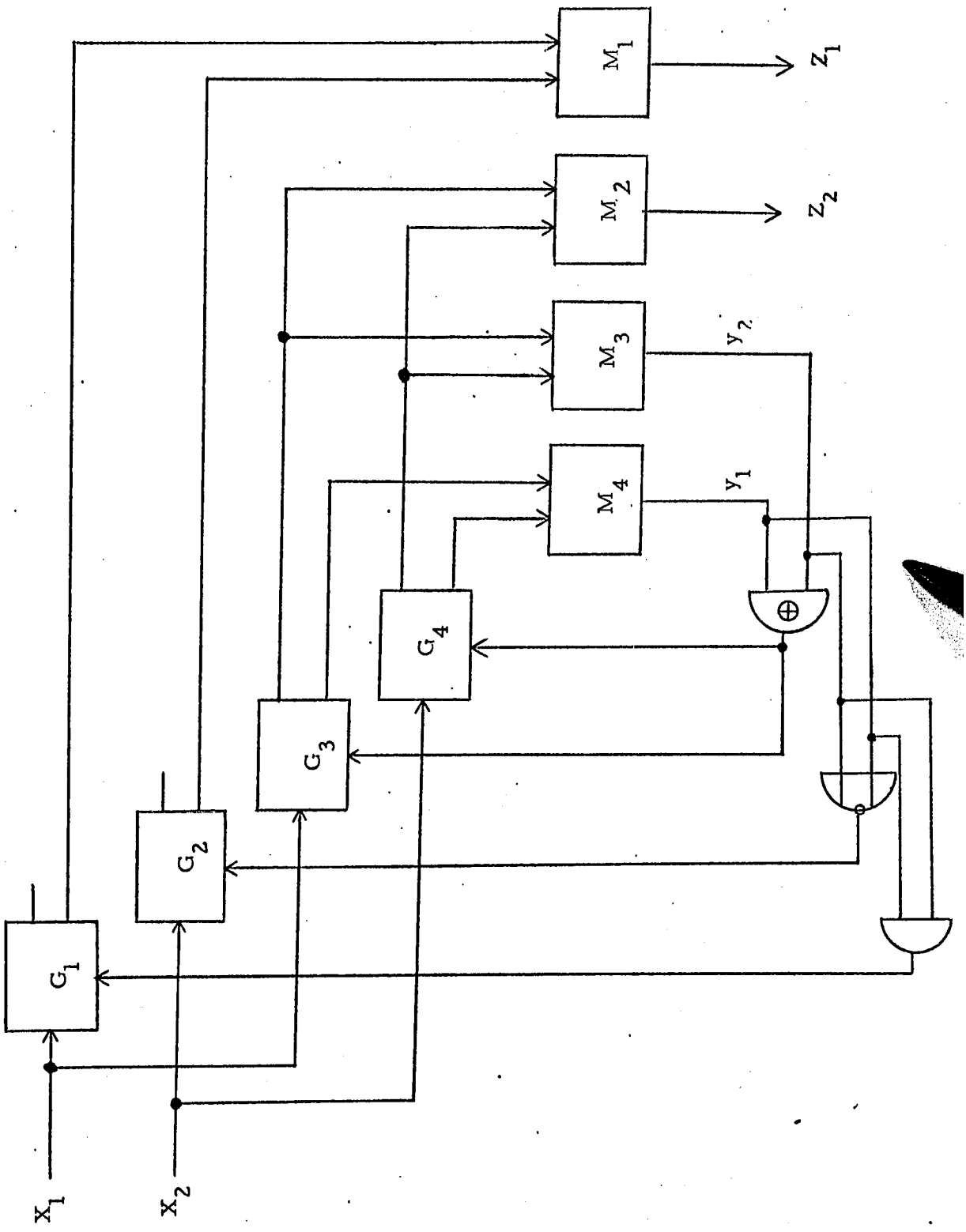
$$Y_1 = X_1(\bar{y}_1y_2 + y_1\bar{y}_2) + X_2(\bar{y}_1y_2 + y_1\bar{y}_2)$$

$$Y_2 = X_1(\bar{y}_1\bar{y}_2 + y_1y_2) + X_2(\bar{y}_1\bar{y}_2 + y_1y_2)$$

$$Z_1 = X_1(y_1y_2) + X_2(\bar{y}_1\bar{y}_2)$$

$$Z_2 = X_1(\bar{y}_1\bar{y}_2 + y_1y_2) + X_2(\bar{y}_1\bar{y}_2 + y_1y_2)$$

Step 5. Circuit Realization


M1
M2
M3
M4
G1
G2
G3
G4
z1
z2
y2
y1
x1
x2

Example 2' (Repeat the Synthesis of Example 2)

Level Flow Table

	00	01	11	10	z
1	①	2	-	5	0
2	-	②	3	-	0
3	-	4	③	8	0
4	1	④	-	-	1
5	-	-	6	⑤	0
6	-	7	⑥	8	0
7	1	⑦	-	-	0
8	1	-	-	⑧	0

Step 1. Transition Flow Table

	X_1	X_2
1	5, 0	2, 0
2	3, 0	- -
3	4, 1	8, 0
4	- -	1, 1
5	- -	6, 0
6	7, 0	8, 0
7	- -	1, 0
8	1, 0	- -

Step 2. Secondary Assignment

		X_1	X_2
000	1	010	110
110	2	101	-
101	3	001	100
001	4	-	000
010	5	-	111
111	6	011	100
011	7	-	000
100	8	000	-

Step 3. Excitation and Output Table

		X_1	X_2
000	1	010, 0	110, 0
110	2	011, 0	- -
101	3	100, 1	001, 0
001	4	- -	001, 1
010	5	- -	101, 0
111	6	100, 0	011, 0
011	7	- -	011, 0
100	8	100, 0	- -
		$Y_1Y_2Y_3$ Z	$Y_1Y_2Y_3$ Z

Step 4 Excitation and Output Functions

$$Y_1 = X_1(y_3 + y_1\bar{y}_2) + X_2(\bar{y}_3)$$

$$Y_2 = X_1(\bar{y}_1 + y_2\bar{y}_3) + X_2(\bar{y}_2\bar{y}_3 + y_2y_3)$$

$$Y_3 = X_1(y_2\bar{y}_3) + X_2(y_3 + \bar{y}_1y_2)$$

$$Z = X_1(\bar{y}_2y_3) + X_2(\bar{y}_1\bar{y}_2y_3)$$

Step 5. Circuit Realization

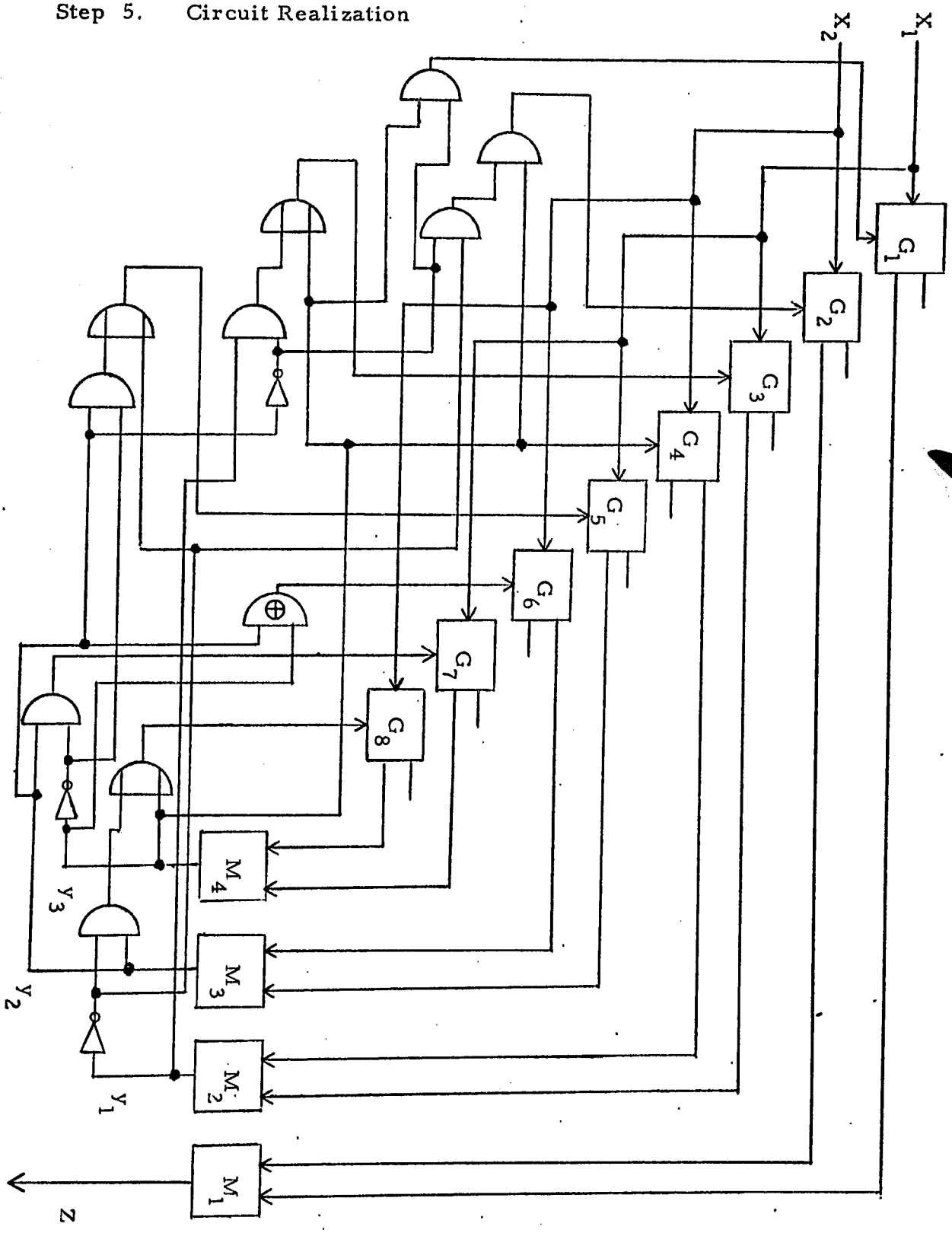


TABLE A -1

TRANSITION LOGIC SYNTHESIS

	Total no. of G-elements Used	Total no. of M-elements Used	Total no. of Logic gates Used	Comparisons
Example 1				
Example 1'	4	4	1 NOR 1 AND 1 Ex-OR 3	More M-elements Used, More Complicated Logic
Example 2				
Example 2'	8	4	3 Inverter 7 AND 3 OR 1 Ex-OR 14	More Complicated Logic And 8 G-elements Used

TABLE A -1 (Cont'd)

DIRECTIONAL TRANSITION LOGIC SYNTHESIS

	Total no. of K-elements Used	Total no. of M-elements Used	Total no, of Logic gates Used	Comparisons
Example 1	4	3	1 Inverter	Less M-elements Used less Complicated Logic
Example 1'				
Example 2	5	4	2 Inverter 2 AND 1 OR 1 NOR 6	Less Logic Gates And Only 5 K-elements Used
Example 2'				

REFERENCES

(1) Chuang, Y. H. 'Transition Logic Circuits and A Synthesis Method', IEEE Trans. on Computer, Vol. C-18, No. 2 ,February, 1968, pp. 154-168.

(2) Chuang, Y.H. 'A Synthesis Technique for A Class of Transition Logic Circuits', Computer Systems Lab., Washington U., St. Louis, Mo., Tech, Memo, 16, February 10, 1967.

(3) Caldwell, S.H. 'Switching Circuits and Logical Design', N.Y. Wiley, 1958.

(4) Batra, V.K. 'Design of Asynchronous Unit Delays', IEEE Trans. on Computer, Vol. C-19, No 10, October, 1970, pp.896-902.

(5) Danielson, P.E. 'A Notes on Wired Gates', IEEE Trans. on Computer, Vol. C-9, No. 9, September, 1970.

(6) Huffman, D.A. 'The Synthesis of Sequential Switching Circuits', J. Franklin Inst., Vol. 257, pp. 161-190 and 257-303, March April, 1954.

(7) Huffman, D.A. 'Design of Hazard-free Switching Circuit', JACM., Vol. 4, pp. 47-62, January, 1957.

(8) Hartmanis, J. and Stearns, R.E. 'Algebraic Structure Theory of Sequential Machines', Engleword Cliffs, N.J. Prentice Hall, 1966.

(9) Krieger, M. 'Basic Switching Circuit Theory', The MacMillan Co. N.Y. 1967.

(10) McCluskey, E.J. 'Introduction to the Theory of Switching Circuits', N.Y. McGraw-Hill, 1965.

(11) Su, Y.H. 'Shift Register Realizations of Asynchronous Sequential Machines', Inf. and Cont. 15, pp. 226-234, 1969.

(12) Sheng, C.L. 'Lectures Notes on Logic Circuits I and II', E.E. Dept., University of Ottawa, Ontario, Canada 1968-69.

(13) Singh, S. ' Design of Asynchronous Sequential Circuits with Asynchronous Unit Delays ', Ph. D. thesis, E. E. Dept., University of Ottawa, Ontario, Canada, November, 1967.

(14) Unger, S. H. ' Hazards and Delays in Asynchronous Sequential Switching Circuit ', IRE Trans. on Circuit Theory, Vol. CT-6, No. 1, pp. 12-25, March, 1959.

(15) Unger, S. H. ' Asynchronous Sequential Switching Circuits ', N. Y. John Wiley and Sons, Inc., 1969.

(16) Wann, D. F. and Ellis, R. A. ' Implementation of Time Dependent Algorithms in Asynchronous Computer Systems ', Computer Systems Lab., Washington University, St. Louis, Mo., Tech. Memo, 32, July 1967.

VITA

Name : Yun-Chunk KWAN

Born : 20 May, 1944, Hong Kong

Educated :

Primary : Hong Kong

Secondary : Tak Ming Middle School, Hong Kong

University : National Taiwan University,
Taipei, Taiwan, China,
B. Sc. in Electrical Engineering.