DESIGN OF ASYNCHRONOUS UNIT DELAYS

by

Vinod K. Batra, B. Tech.

Submitted in partial fulfillment of the requirements

for the degree of Master of Science

Department of Electrical Engineering

Faculty of Pure and Applied Science

University of Ottawa

Ottawa, Canada.

May, 1967.

DESIGN OF ASYNCHRONOUS UNIT DELAYS

ABSTRACT

The problem considered is the design of an asynchronous unit delay. An asynchronous unit delay is an n-input, n-output, asynchronous sequential circuit such that the present value of the output n-tuple is equal to the value of the input n-tuple prior to the last input change. The delay is of significance as a building block for shift register realizations of asynchronous circuits.

The logical design of a 2-input asynchronous unit delay is first investigated. Several different circuit realizations are developed and compared. The logical design of a 3-input delay and of a 2-input, 2-unit delay is also examined. A different design for an n-input delay is obtained by using an internally generated pulse to control the state changes. It is shown that this technique can be generalized to apply to an arbitrary fundamental mode sequential circuit: The method utilises a change detecting circuit to produce a control pulse and the remainder of the design can be done on pulse mode basis. This allows unrestricted assignments for the main part of the circuit which now operates in pulse mode.

The 2-input asynchronous delay is analysed and performance criteria are developed. The operation of two delays connected in series is also studied. Several circuit realizations are then experimentally tested for the important design parameters and a comparison of the realizations is made.

ACKNOWLEDGMENTS

The author wishes to thank Dr. Janusz A. Brzozowski of the Electrical Engineering Department, for his contributions to the development of ideas expressed in this thesis. The assistance is acknowledged of Professor P.M. Thompson of the Electrical Engineering Department, who provided the ideas for the pulse-controlled design.

Thanks are due to my colleagues the graduate students in the Electrical Engineering Department, and in particular: Shanker Singh and Godfrey Muehle. The auther is grateful to Mrs. G. Chalmer who typed the final manuscript.

This work was supported by the National Research Council of Canada under grant No. A-1617.

TABLE OF CONTENTS

TABLE OF CONTENTS

CHAPTER 1

FUNDAMENTALS

1.1 Introduction

Switching circuits are normally classified as being either synchronous or asynchronous. In synchronous circuits, clock pulses regulate the operation of the circuit and a knowledge of logical characteristics is sufficient to predict the state of the circuit at any clock pulse. Each operation in a system, such as a computer, designed in the synchronous mode is allotted a specific amount of time. As a result of using specific delay times, it is possible that systems of this type are operating at a fraction of their potential speed. On the other hand, circuits may be designed without external timing. There is no source of clock pulses to synchronize the operations and, in effect, the resulting system may take full advantage of the basic device speed. Thus a system designed in the asynchronous mode will operate as fast as switching can be accomplished. In order to achieve this each building block of the system must be designed to emit an output signal when the input signals have been processed. However, the absence of clock pulses introduces the problem of insuring that the circuit functions according to specifications, independent of variations in speeds of some of the logical elements which comprise the system.

This thesis considers the design of an Asynchronous Unit Delay (AUD), which can be used as a basic building block for realizations of

asynchronous circuits [1]. An asynchronous unit delay is an n-input, n-output asynchronous sequential circuit in which the present value of the output n-tuple is equal to the value of the input n-tuple prior to the last input change; thus for such a device

$$z_i(t) = x_i(t-\Delta t) \quad \text{for} \quad i = 1,2,\ldots,n$$

where $(t-\Delta t)$ is the instant of time when the input n-tuple was last changed.

Realizations using such functional blocks have the advantage of leading to simple and often economical assignments for the internal states of the machine [1]. Another advantage of the AUD is that it is the only memory device required for such realizations. For $n \geq 2$, it is a larger building block than a flip-flop and its use leads to simpler structures of identical elements.

This thesis is divided into 4 sections; the first summarizes the basic logic concepts used in later developments. The second section considers the design of an Asynchronous Unit Delay under fundamental mode and under pulse-controlled operation. A method is proposed for realizations of asynchronous circuits, in general, by an equivalent clocked synchronous circuit. Further, various circuit models are described and the performance under the two schemes is compared. In the third section we describe some basic circuit concepts and the tests to determine whether or not the circuit is operating properly, and finally, in the last section, the different circuit models are compared and experimental results are summarized.

1.2 Sequential Circuits

An asynchronous circuit can be defined as an r-input, p-output

device. The signals at the input and output terminals are binary and take the values 0 and 1. There is no numerical significance attached to these symbols. In an electronic switching circuit, the signals are usually interpreted as voltages and frequently the higher voltage is assigned the symbol 1. However, there are times when the other assignment is more useful.

A signal is called a level signal if it can assume the value 0 or 1 for any length of time, greater than some minimum duration t_0 which is the response time of the circuit. An ideal binary, level signal is sketched in Fig. 1.1. A pulse, on the other hand, is a signal which is in the 1 state for much shorter period of time than it is in 0 state.

The set of signals present at the input terminals are referred to collectively as the input state (for brevity, the input), and the term output state (or output) refers to the set of output signals. If the output is always uniquely determined by the present input state, the system is called a combinational circuit. A more general situation occurs in the case of a sequential circuit in which the output state is a function of present input state and a past history of the inputs. A sequential circuit can be described by a system of the form shown in Fig. 1.2, which consists of combinational circuits with some of the output signals fed back to some of the input terminals. The feedback branches are called state-branches. The signal at the output end of each state-branch is defined as a state-variable and will be represented by y_i, for the i-th state-branch. For proper sequential operation, the gain in the feedback loops should be greater than unity in order to have two stable states possible in the loop.

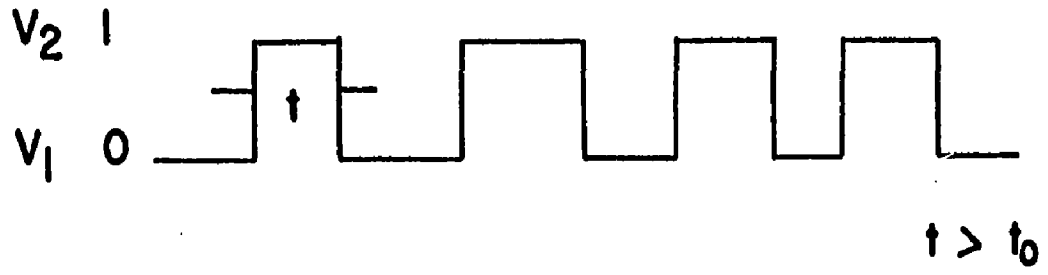


Fig. 1.1

Ideal Binary Level Signal

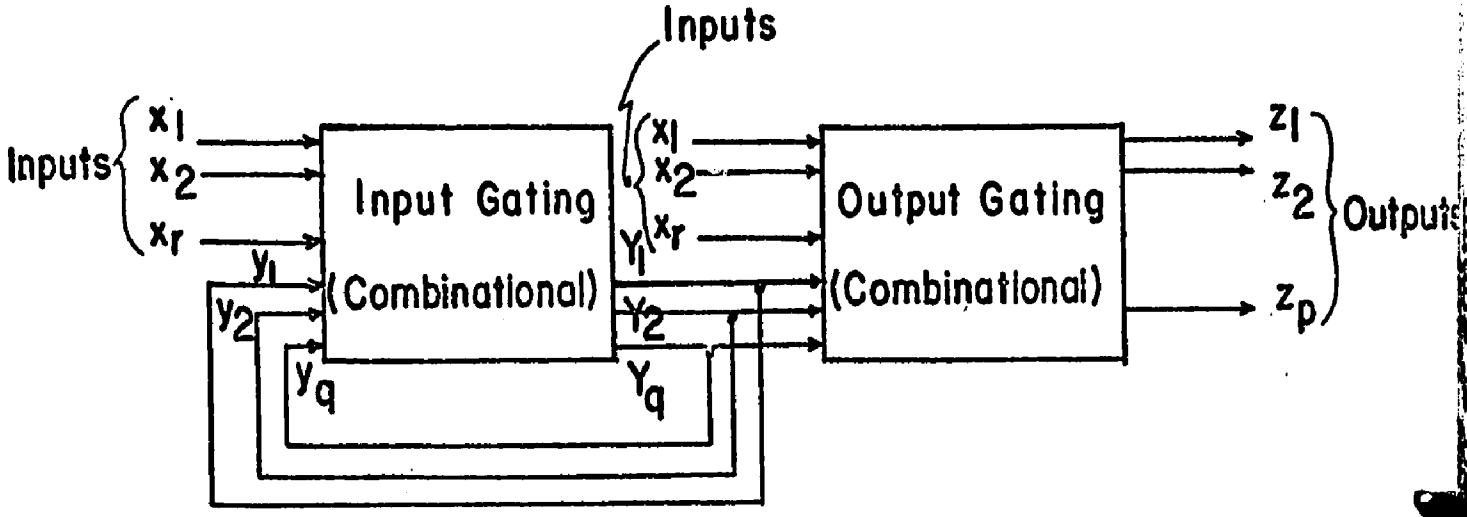


Fig. 1.2

General Form of an Asynchronous Sequential Circuit

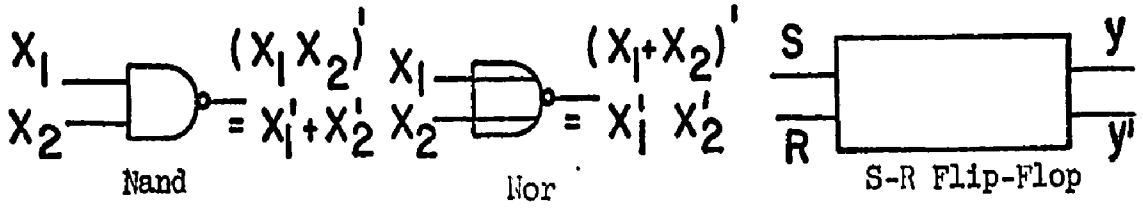


Fig. 1.3

The combinational circuits can be constructed with a wide variety of devices such as electronic gates, relays, cryotrons, magnetic cores, etc. In our discussion we shall be using electronic gate-type logic. A symbolic representation of Nand, Nor gates and Set-Reset (S-R) flip-flop is given in Fig. 1.3. The operation of an S-R flip-flop is shown in Table 1.1.

SR / y	00	01	11	10
0	0	0	-	1
1	1	0	-	1

Y

Table 1.1

S-R Flip-Flop

1.3 Flow Tables

A commonly used method for describing the terminal characteristics of a sequential circuit is the flow table [6]. The columns of the table correspond to input states, the rows represent the present internal states, and the table entries are the next internal states and the output states. For example, if the circuit of Table 1.2 is in state y_1 under input X_1, the next internal state is also Y_1 and the circuit is stable.

	X_1	X_2	X_3	X_4
y_1	(1),10	Y_4,10	Y_3,00	(1),00
y_2	Y_4,--	---	Y_1,10	(2),10
y_3	(3),01	Y_5,--	Y_4,--	Y_2,--
y_4	(4),01	(4),00	Y_5,00	Y_2,11
y_5	Y_3,--	(5),00	Y_2,--	Y_1,00

Table 1.2

Flow Table Description of a General Sequential Circuit

If now the input is changed to X_2, the next internal state will change to Y_4 and stay there as long as X_2 is present. There is another type of transition possible: If the circuit was presently in state y_5 under input X_2 and the input was changed to X_3, the circuit would cycle through all the states. Such transitions will not be considered here. A transition for which the next internal state is '--' (a don't care) means that further circuit action is of no interest. A total state, which is an input state X_i and an internal state Y_j is called stable, if the next state entry is equal to the present internal state. This fact is denoted by circling the next state entry. When the circuit represented by a flow table is in a stable total state, the internal state cannot change until the input state is changed. Whenever the circuit inputs are controlled in such a way that no input is changed unless the present total state is stable, the circuit is said to be operating in a fundamental mode [8]. If no more than one change occurs in the internal state for each input change, the circuit is said to operate in normal fundamental mode [5].

A different mode of operation is possible if pulses are present at the input. A sequential circuit is said to be operating in pulse-mode [8] if the following conditions are satisfied.

(i) At least one of the inputs is a pulse signal.

(ii) Changes in internal state occur only in response to the occurence of a pulse.

(iii) Each input change causes at most one change in each internal state but any number of internal state variables are allowed to change at the same time.

A fundamental mode flow table with at most one stable state in each row of the table, is called a primitive flow table [8]. Such a table can be reduced by any of the several methods proposed in [6, 8, 10]

1.4 Serial and Parallel Decompositions

An important step in the synthesis procedure is the binary coding of the internal states of a flow table. It is the construction of this code, called a state assignment, that must be made in such a fashion that the circuit will function according to flow table specifications independent of variations of delays within the circuit. A useful tool in the determination of suitable codes for the internal states of a machine is the partition theory [7].

Definition 1.1: A partition [7] on the set of states Y of a flow table is a collection of disjoint subsets of Y whose set union is Y i.e.

$$\pi = \{a_i\}$$

such that $a_i \cap a_j = \phi$ for $i \neq j$ and

$$\cup(a_i) = Y.$$

We refer to the a_i as blocks of π .

Definition 1.2: A partition π on the set of states of a flow table is said to have the substitution property [7] (referred to as S.P.) if and only if each input maps blocks of π into blocks of π . For example if y_i and y_j are contained in the same block of π then the next states of y_i and y_j under any input X_i are also in the same block of π .

We shall require later, under what conditions a sequential circuit described by a flow table can be realized as a parallel or series connection of two smaller circuits. The pertinent results are summarized in theorems 1.1 and 1.2 [7]. In these theorems only state behaviour realizations will be considered which corresponds to assigning a single n-tuple of secondary variables to each state.

Theorem 1.1: A sequential circuit S has a nontrivial parallel decomposition of its state behaviour if and only if there exist two nontrivial S.P. partitions π_1 and π_2 in S such that

$\pi_1 \cdot \pi_2 = 0$, where 0 is the trivial partition with one state per block.

Theorem 1.2: A sequential circuit S has a nontrivial serial decomposition of its state behaviour if and only if there exists a nontrivial S.P. partition π on the set of states Y of S.

It has been shown [7] that often state decompositions lead to realizations in which the complexity of the next-state function is reduced. However, there are flow tables for which the most economical realization is not the state behaviour realization. Also in reducing a flow table before analysing its structure one may obscure some simple realizations. In such

cases it may be better to expand the flow table through state splitting. For further details the reader is referred to [7].

1.5 Timing Problems in Sequential Circuits

The various logical elements comprising a sequential circuit do not operate instantaneously. Instead, there are delays caused by the wiring and the gates. Further, the gate delays may vary. For example, individual transistors as opposed to wiring delay have switching delays that vary not only with time as components age but more importantly, they vary individually depending on their loading, whether turning on or off, the slope of the input signals, fluctuations in the power supply, change in their gain, temperature, their switching threshold etc.

In any circuit there can be different paths for an input to affect the output and the delays associated with each path are, in general, different. These unequal delays have the tendency to cause spurious transients which may affect the overall operation of the circuit. For synchronous circuits, such transients during a change of state variables are ignored, and several state variables are allowed to change simultaneously. But for asynchronous circuits, the state variables are allowed to change simultaneously (races) only if the resulting state does not depend upon the order of change of these variables (Non-critical races). If a race condition exists and there is a possibility that unequal delays may cause the circuit to reach a stable state other than the one intended, the race is called critical.

Beside races, spurious momentary outputs can occur in a specific network during a change of input. A transition between a pair of input states which produces no change in the steady state output will be considered

first. If it is possible for a spurious output to be produced during such a transition, this condition is said to correspond to a static hazard [8]. Spurious outputs can also be produced during input changes which do affect the steady state output. Such a condition is called dynamic hazard [8]

Critical race conditions can be avoided by making appropriate row assignments [6] and hazards, by inserting delay elements in all feedback loops and using hazard-free combinational circuits [11] or by any of the several methods proposed in [3,4,8,11].

There are also restrictions on the way the inputs can change. For multi-input circuits we have to assume that inputs do not change simultaneously. This is because if simultaneous changes were allowed, they may propagate at different speeds through the circuit and changes will not be felt simultaneously at the outputs. Due to this restriction on the input changes, the input rate becomes a function of the number of inputs. We shall derive the exact relationship in Chapter 3.

CHAPTER 2

Asynchronous Unit Delay

2.1 Definition:

In this chapter we shall consider the logical design of an Asynchronous Unit Delay (AUD). Realizations using cascade connections of unit delays (shift registers) are quite common for synchronous circuits. A unit delay for synchronous circuits can be a clocked flip-flop. A similar module for asynchronous circuits is defined as follows.

Definition 2.1: [1, 8] An ideal n x n asynchronous unit delay (n x n AUD) is an asynchronous sequential circuit with n-binary level inputs x_1 x_2,..., x_n and n-binary level outputs z_1, z_2,..., z_n, such that at any time t the value of the output n-tuple is equal to the value of the input n-tuple at time $(t - \Delta t)$, where $(t - \Delta t)$ is the instant of time when the input n-tuple was changed to its present value; thus for such a device

$$z_i(t) = x_i(t - \Delta t), \quad \text{for } i = 1, 2, \ldots, n.$$

The input-output behaviour of an n x n AUD is described in Table 2.1.

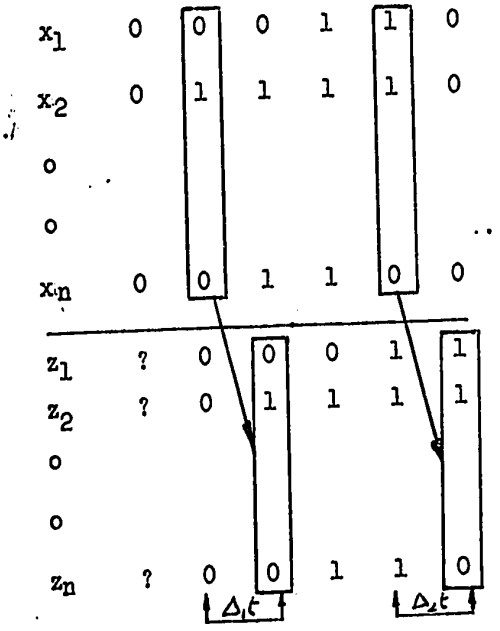


Table 2.1

A 1 x 1 binary input AUD is a trivial case of n x n AUD. For example, an inverter and a normally closed relay contact are 1 x 1 AUD's.

In the following section we shall give general methods for the design of an n x n AUD and various design schemes for a 2 x 2 AUD will be considered.

2.2 Design Consideration

To insure proper operation of the circuits we make the following assumptions:

(i) The machine specified is an asynchronous machine.

(ii) No critical race conditions occur during a state change.

(iii) The combinational networks are hazard-free.

(iv) The delay in feedback lines is sufficiently large to allow the associated combinational network to become stable before the state changes.

(v) The input rate is sufficiently slow to allow the network to become stable before the inputs are changed.

(vi) In the general case all input changes are allowed, but for our interest we restrict ourselves to adjacent input combinations, i.e. only single input changes are allowed.

We shall describe two methods for the design of an AUD.

(i) Fundamental Mode Operation

(ii) Pulse-Controlled Operation.

Fundamental Mode Operation [1]

Case 1, All input changes allowed:

Assume that the AUD is stable in some input column. The previous input can be any one of the 2^n-1 possible inputs; hence there must be at

least 2^n-1 states in each column, each state having the output corresponding to the previous input. In a normal primitive flow table there must be at least $2^n(2^n-1)$ stable states or rows. This number of states is also sufficient, for suppose the present total state is the stable state (Y_i,X_j). If the input changes to X_k then the next total state (Y_i,X_k) is a stable state Y_p in column X_k and the output changes to X_j. Thus in the normal primitive flow table all transitions have the form shown in Table 2.2.

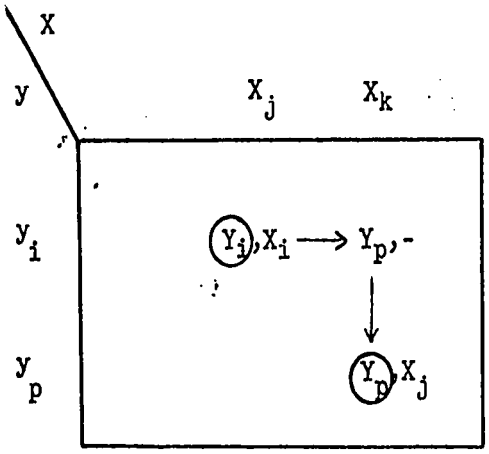


Table 2.2

Typical Transition In Fundamental Mode Operation

There are no unspecified next state entries and it can be verified that this itself is the reduced table for $n \geq 2$.

Case 2, Single input changes only:

Assume only single input changes are allowed. A flow table describing such a circuit must have n stable states per column, corresponding to n allowed previous inputs. In a normal primitive flow table, there must be at least 2^nn rows. Since there would be don't care entries in such a flow table, it is possible to merge some rows and get a reduced table, as will be illustrated in the next section.

2.3 Fundamental Mode Realizations of the 2 x 2 AUD

The normal primitive flow table under the restriction of single input changes is given in Table 2.3. The table contains 8 states and can be further reduced to table 2.4. For the case of all input changes, the reduced flow table [1] contains 12 states. This case is mainly of academic

x_1x_2	00	01	11	10
1	(1),10	2,-	--	3,-
2	4,-	(2),00	5,-	-
3	1,-	-	6,-	(3),00
4	(4),01	2,-	-	3,-
5	-	8,-	(5),01	7,-
6	-	8,-	(6),10	7,-
7	1,-	-	6,-	(7),11
8	4,-	(8),11	5,-	-

Table 2.3

2 x 2 AUD-Single Input Changes Only

x_1x_2	00	01	11	10
1	(1),01	(1),00	3,-	2,-
2	(2),10	1,-	4,-	(2),00
3	1,-	(3),11	(3),01	4,-
4	2,--	3,-	(4),10	(4),11

Table 2.4

Reduced Table for 2 x 2 AUD

interest because in practice even if simultaneous changes were allowed in the inputs, the effects of input changes will not be simultaneous throughout the circuit.

The physical realization of the flow table (Table 2.4) depends upon the state assignment. Since every distinct 1-1 race-free assignment leads to a different realization, to choose a "best" realization we impose the following requirements. In the model of Fig. 1.2 we have seen that there are different delays involved in each path. So the desired realization is to have minimal logic (cost function in terms of gates), and least transmission delays, which should be as equal as possible.

Often it is possible to reduce the logic requirements by reducing the functional dependence among the state variables. The transmission delays can be reduced by reducing the reaction time i.e. making one step transitions.

The flow table of Table 2.4 has two non-trivial partitions with substitution property:

$$\pi_1 = \{\overline{12},\ \overline{34}\} \quad \text{and} \quad \pi_2 = \{\overline{13},\ \overline{24}\}$$

such that $\pi_1 \cdot \pi_2 = 0$. Hence the circuit can be realized as a parallel combination of two subcircuits s_1 and s_2. The state assignment according to π_1 and π_2 leads to the following next state and output equations, where $y_1 = o$ is assigned to block $\overline{12}$ of π_1 and $y_2 = o$ to block $\overline{13}$ of π_2 :

$$Y_1 = x_1x_2 + y_1(x_1 + x_2) \qquad \text{(i)}$$

$$Y_2 = x_1x_2' + y_2(x_1 + x_2') \qquad \text{(ii)}$$

$$z_1 = y_1y_2 + x_1'(y_1 + y_2) \qquad \text{(iii)}$$

$$z_2 = y_1y_2' + x_2'(y_1 + y_2') \qquad \text{(iv)}$$

Assuming double-rail inputs (both x and x' available) and using Nand gates, the circuit requires 16 gates and one inverter (Total 17 gates). It can be verified that this is the only race-free, 2 variable, 1-1 assignment for this flow table, not counting trivial modifications by complementing and/or interchanging variables.

Note that such a modular realization of the flow table has lead to quite a symmetrical circuit (Fig. 2.1) with respect to the outputs, and we can expect approximately equal transmission delays. One can also verify that neither output could be equal to a state variable in the realization using two circuits in parallel because the reduced flow table

$Y_1 = x_1x_2 + y_1(x_1 + x_2)$ $\qquad Z_1 = y_1y_2 + x_1' (y_1 + y_2)$

$Y_2 = x_1x_2' + y_2(x_1 + x_2')$ $\qquad Z_2 = y_1y_2' + x_2' (y_1 + y_2')$

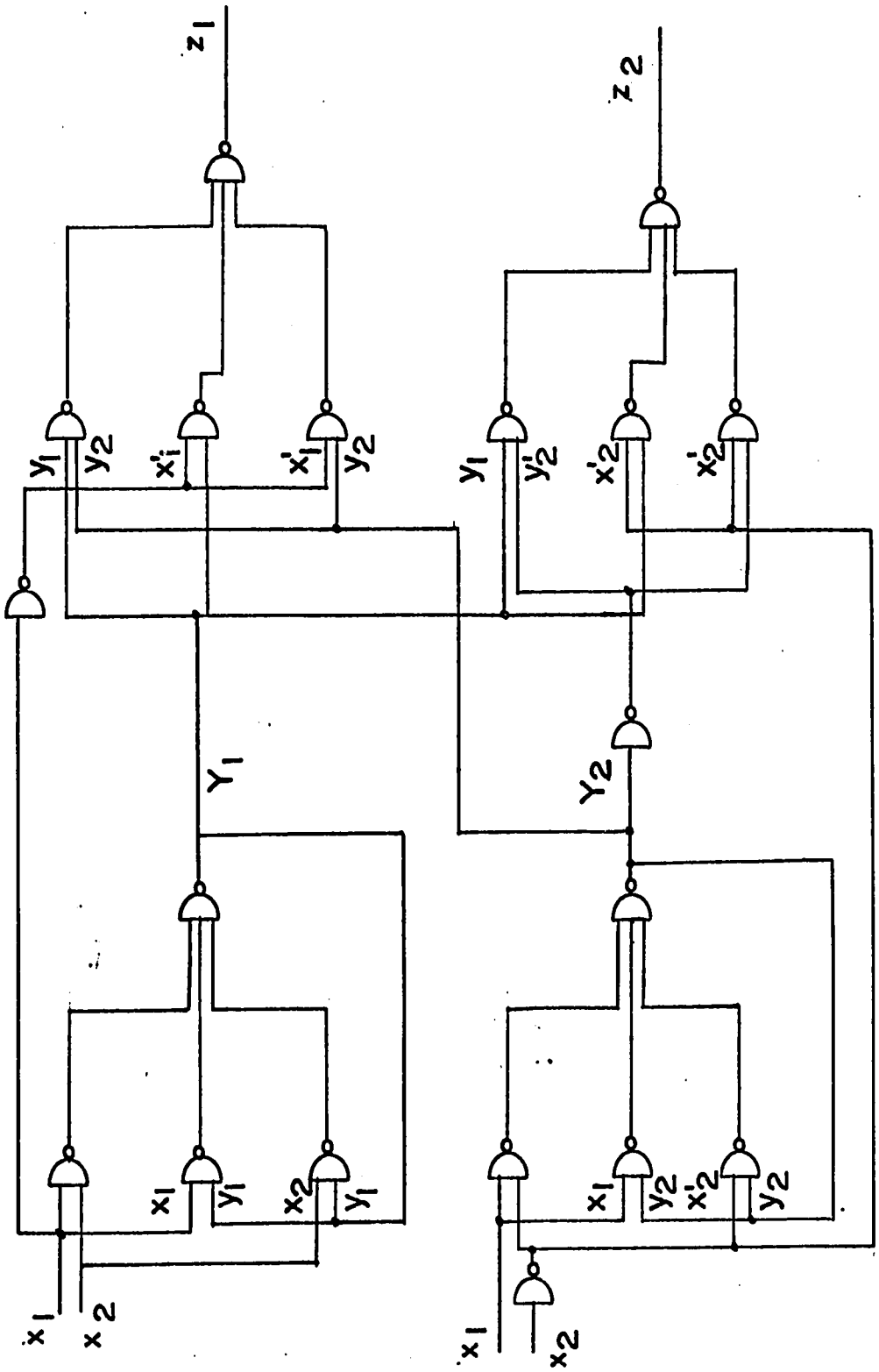


Fig. 2.1

of Table 2.4 does not have an output consistent S.P. partition. (A partition is output consistent if the output is the same for all the states in any block.) It can be shown that for the flow table of Table 2.3, there exists only one race-free, 3 variable, 1-1 assignment which uses 21 gates. By expanding the machine through state splitting we find that there does not exist a 3 variable assignment which is free of all races and is more economical than the parallel realization considered above.

Some 4 variable assignments were examined but none was found to be better than the parallel 2 variable realization considered above.

Consider next the possibility of obtaining an economical realization by reducing the feedback loops as proposed by Eichelberger [4] and Friedman [5]. Eichelberger's approach using hazards and delays shows that any double-rail input flow table can be realized by a delayed-input circuit (Fig. 2.2) containing S_o state branches, where S_o is the smallest integer that is at least equal to the base two logarithm of the number n of the states of the flow table. This is denoted by $S_o = [\log_2 n]$. If it is desired to reduce the number of feedback loops in the circuit, it has been shown that any normal fundamental mode flow table containing a maximum of S stable states per column can be realized with feedback index $Q = [\log_2 S]$.

This approach was used for the 2 x 2 AUD and it was found that such realizations involve at least the same cost as the 2-variable parallel realization. Moreover there is an increase in the transmission delays.

Friedman's method of assignment using non-critical races leads to a state assignment similar to the 2-variable parallel realization. However the feedback and output functions require more logical hardware as shown below.

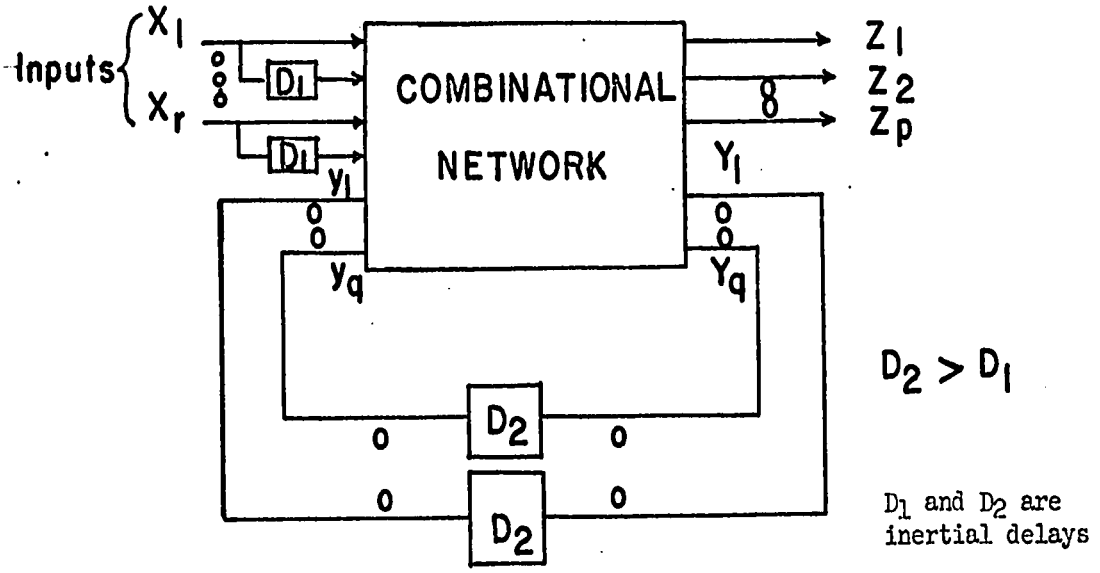


Fig. 2.2

Eichelberger's Delayed Input Model

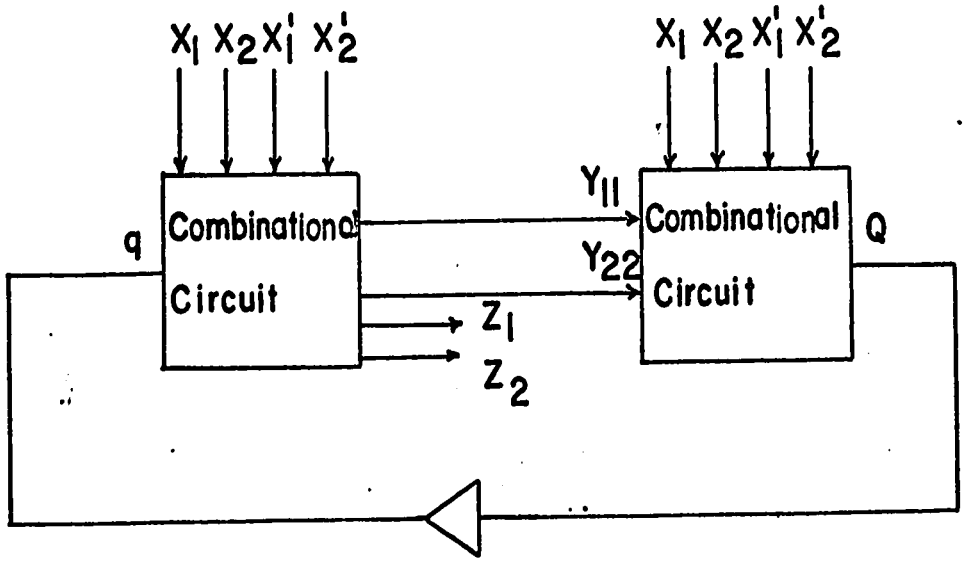


Fig. 2.3

Single Feedback loop Realization of 2 x 2 AUD

$$Y_{11} = x_1x_2' + q(x_1 + x_2')$$

$$Y_{22} = x_1x_2 + q(x_1 + x_2)$$

q is the feedback function given by

$$Q = (x_1'x_2' + x_1x_2)\ Y_{11} + (x_1'x_2 + x_1x_2')\ Y_{22}$$

and $z_1 = q$

$$z_2 = (x_1'x_2' + x_1x_2)\ q' + (x_1'x_2 + x_1x_2')\ q$$

The schematic representation of the AUD with feedback index one is shown in Fig. 2.3.

A comparison of various assignment schemes shows that the parallel realization of Table 2.4 requires the minimal logic and every transition is a one step transition (i.e. every input change leads to a stable state). A circuit diagram using Nand gates is shown in Fig. 2.1 and will be used later.

2.4 3 x 3 AUD

The reduced flow table for a 3 x 3 AUD is given in Table 2.5. The flow table has 4 partitions $\pi_1,\ldots,\pi_4$ with substitution property and $\pi_1 \cdot \pi_2 \cdot \pi_3 \cdot \pi_4 = 0$; thus it can be realized as a parallel connection of 4 subcircuits. The partitions are;

$$\pi_1 = \{\overline{13451012}\ ,\ \overline{2678911}\}$$

$$\pi_2 = \{\overline{124568}\ ,\ \overline{379101112}\}$$

$$\pi_3 = \{\overline{12351112}\ ,\ \overline{4678910}\}$$

$$\pi_4 = \{\overline{123467}\ ,\ \overline{589101112}\}$$

The assignment based on these partitions leads to the following next state and output equations.

	$x_1x_2x_3$ 000	001	011	010	110	111	101	100
1	(1),001	(1),000	4,-	2,-	--	--	5,-	3,-
2	(2),010	1,-	6,-	(2),000	8,-	--	--	3,-
3	(3),100	1,-	--	2,-	10,-	--	12,-	(3),000
4	1,-	(4),011	(4),001	6,-	--	7,-	5,-	--
5	1,-	(5),101	4,-	--	--	11,-	(5),001	12,-
6	2,-	4,-	(6),010	(6),011	8,-	7,-	--	--
7	--	4,-	(7),111	6,-	9,-	(7),011	11,-	--
8	2,-	--	6,-	(8),110	(8),010	9,-	--	10,-
9	--	--	7,-	8,-	(9),111	(9),110	11,-	10,-
10	3,-	--	--	8,-	(10),100	9,-	12,-	(10),110
11	--	5,-	7,-	--	9,-	(11),101	(11),111	12,-
12	3,-	5,-	--	--	10,-	11,-	(12),100	(12),101

Table 2.5

Reduced Table for 3 x 3 AUD - Single Input Changes Only

$$Y_1 = x_1y_1 + x_3'(x_1x_2 + x_2y_1) + x_3(x_1x_2' + x_2'y_1)$$

$$Y_2 = x_2y_2 + x_3'(x_1x_2 + x_1'y_2) + x_3(x_1'x_2 + x_1'y_2)$$

$$Y_3 = x_1y_3 + x_3'(x_1x_2' + x_2'y_3) + x_3(x_1x_2 + x_2y_3)$$

$$Y_4 = x_2y_4 + x_3'(x_1'x_2 + x_1'y_4) + x_3(x_1x_2 + x_1y_4)$$

$$z_1 = y_1y_3 + x_1'(y_1 + y_3)$$

$$z_2 = y_2y_4 + x_2'(y_2 + y_4)$$

$$z_3 = y_1y_2'y_3' + y_1'y_2y_3 + y_2y_3'y_4' + y_2'y_3y_4 + x_3'(y_2'y_3' + y_3y_4 + y_1y_2' + x_2y_2)$$

Note that a parallel realization of this flow table leads to a very unsymmetrical circuit with respect to the outputs and our assumption of all path lengths being the same is no more valid. An attempt to equalize the delays leads to about twice the hardware. On the other hand, if we consider the worst case design (maximum delay in each path), the input rate must be reduced appreciably. Thus this design for the 3 x 3 AUD does not appear to be very attractive.

2.5 2-Unit Asynchronous Delay

So far we have been able to remember the input n-tuple prior to the last input change. The input history prior to that can be remembered by a cascade connection of AUD's.

Definition 2.2: An n x n Asynchronous Shift Register (n x n ASR) of length k is a structure of k, n x n AUD's connected in cascade, without feedback; the outputs of the ith AUD being the inputs to the (i+1)st.

In the cascade connection of AUD's the stray wiring and transistor delays become a deciding factor on the length of the shift register. For every additional stage, the delays keep on accumulating and this may require the reduction of the input rate for proper operation. There is a possibility of reducing the accumulation of the stray delays by designing a bigger

building block i.e. instead of just remembering the input prior to the last change, we shall consider the design of a circuit which also remembers the input prior to the last two changes; thus for such a circuit

$$[z_1(t),\ z_2(t)] = [x_1(t-\Delta_1 t),\ x_2(t-\Delta_1 t)]$$

$$[z_3(t),\ z_4(t)] = [x_1(t-\Delta_1 t-\Delta_2 t),\ x_2(t-\Delta_1 t-\Delta_2 t)]$$

where $(t-\Delta_1 t)$ and $(t-\Delta_1 t-\Delta_2 t)$ are the instants of time when the input was changed.

A characterization of this type leads to the following reduced table (Table 2.6).

A comparison of this flow table with the table obtained by a cascade connection of two AUD's (Table 2.7) shows that a subtable of Table 2.7 is isomorphic to Table 2.6. Table 2.7 is fully specified but the states numbered 3, 8, 9, and 14 are transient states and can never be entered.

A state assignment for Table 2.6 based on unit step transitions would be most acceptable to reduce delays. This technique requires $2^{S_o}-1=15$ internal variables ($S_o=[\log_2 S]$)[6] and an assignment of this magnitude becomes quite complicated. Thus reduction of accumulated delays by this method appears to be very costly.

$x_1 x_2$	00	01	11	10
1	(1),01,00	(1),00,01	3,--	2,--
2	6,--	--	4,--	(2),00,01
3	--	5,--	(3),01,00	7,--
4	--	9,--	(4),10,00	8,--
5	10,--	(5),11,01	(5),01,11	7,--
6	(6),10,00	11,--	4,--	(6),00,10
7	12,--	--	8,--	(7),11,01
8	12,--	9,--	(8),10,11	(8),11,10
9	10,--	(9),11,10	5,--	--
10	(10),01,11	1,--	--	2,--
11	1,--	(11),00,10	3,--	--
12	(12),10,11	11,--	--	6,--

Table 2.6

Reduced Table for 2-Unit Delay - Single Input Changes Only

x_1x_2	00	01	11	10
1	(1),01,00	(1),00,01	13,-	5,-
2	1,-	(2),00,10	14,-	5,-
3	4,-	2,-	14,-	8,-
4	(4),01,11	1,-	13,-	8,-
5	6,-	2,-	9,-	(5),00,01
6	(6),10,00	2,-	10,-	(6),00,10
7	(7),10,11	3,-	10,-	6,-
8	7,-	3,-	9,-	5,-
9	8,-	14,-	10,-	12,-
10	7,-	14,-	(10),10,00	11,-
11	7,-	15,-	(11),10,11	(11),11,10
12	8,-	15,-	11,-	(12),11,01
13	4,-	16,-	(13),01,00	9,-
14	3,-	15,-	13,-	9,-
15	3,-	(15),11,10	16,-	12,-
16	4,-	(16),11,01	(16),01,11	12,-

Table 2.7

Two 2 x 2 AUD's In Cascade - Single Input Changes Only

2.6 Pulse-Controlled Operation

A different method of designing an n x n AUD has been proposed by P.M. Thompson *. In an AUD, the present value of the output n-tuple is equal to the input n-tuple prior to the last input change. This behaviour can be achieved as follows

(i) Store the present value of the input till an input change is detected

(ii) Detect any input change and generate a "change" pulse.

(iii) Shift the stored value of the input to the output under the control of the generated pulse.

A system of this type can be described by a circuit model of Fig. 2.4.

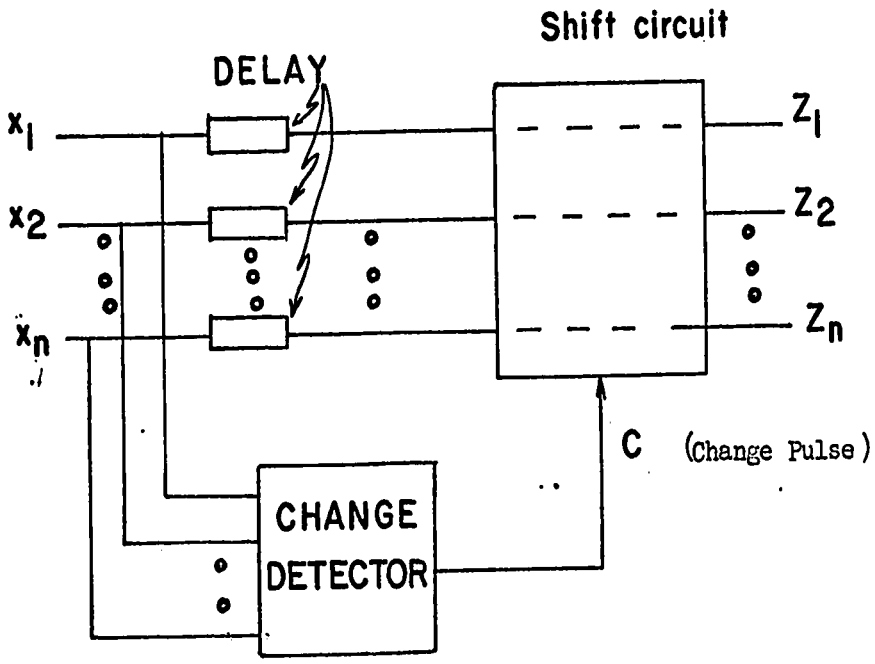


Fig. 2.4

Block Diagram of n x n AUD - Pulse-Controlled Operation

The circuit has n-level inputs and n-level outputs and one internally generated pulse. The desired performance is as follows

(i) The change pulse is present only during an input change.

(ii) The delay in input lines should be greater than the sum of the delay through change detector and of the duration of the change pulse.

(iii) The output change occurs only in the presence of the change pulse.

(iv) All input changes are allowed.

Before considering some specific realizations for the AUD under this scheme we note that by introducing a change detector circuit, an asynchronous sequential behaviour can be achieved by an equivalent clocked synchronous circuit. A change pulse is obtained for every input transition and this pulse acts like a clock. Thus given an asynchronous flow table, it should be possible to obtain an equivalent synchronous table. Such a transformation is described below.

2.6.1 Transformation of an Asynchronous Flow Table to an Equivalent Synchronous Flow Table

Definition 2.3 [2] : Two state tables are said to be equivalent if for each stable (Y_i, X_j) in one state table, there is at least one stable (Y_i, X_j) in the other state table such that taking these as the initial conditions, the same output sequence is obtained for any given input sequence. (Only the outputs associated with the stable states are of interest.)

The form of state table adopted for asynchronous tables has an ordered pair of numbers (next state and output) as an entry [Table 1.2]. The first step in transforming a Level input-Level output [LL Table] to an

equivalent Pulse input-Level output [PL Table] is to number arbitrarily the different ordered pairs corresponding to stable states in the LL flow table. Don't care entries are not to be numbered. A row in the PL table is then assigned for each differently numbered ordered pair. When the change pulse is absent, all next states are same as the present state. There may be don't care entries due to restriction on input changes. The next state entries for C = 1 are filled in according to the transitions specified in the LL flow table. All stable entries which appear alone in a row for C = 1 are replaced by don't care entries. Making all possible row mergers gives the final PL equivalent state table.

The validity of this procedure stems from the fact that in a PL Table, the output is associated with the internal state rather than the total state. Thus the first step is to recognize all the different internal states. This is also equivalent to obtaining the primitive form for the LL flow table. An unstable entry leading to a stable entry in the LL flow table furnishes the next state information for the PL flow table. Since the state changes take place only under the control of the change pulse, all states are stable for C = 0.

The output is only associated with the stable state. Thus appending an output column gives us all the information for the PL flow table. The stable entries under C = 1 are superfluous because like successive inputs cannot occur and they can be replaced by don't care entries.

We shall illustrate the transformation described above for a 2 x 2 AUD operating in fundamental mode operation. Since the primitive

flow table is already available [Table 2.3] there is no need to number the ordered pairs. The transformed table is given in Table 2.10.

	Input x_1x_2								Output z_1z_2
	C = 0				C = 1				
	00	01	11	10	00	01	11	10	
1	①	①	--	①	--	2	--	3	10
2	②	②	②	--	4	--	5	--	00
3	③	--	③	③	1	--	6	--	00
4	④	④	--	④	--	2	--	3	01
5	--	⑤	⑤	⑤	--	8	--	7	01
6	--	⑥	⑥	⑥	--	8	--	7	10
7	⑦	--	⑦	⑦	1	--	6	--	11
8	⑧	⑧	⑧	--	4	--	5	--	11

Table 2.10

Equivalent PL Table for 2 x 2 AUD

None of the rows can be merged but all columns under C = 0 can be merged. The reduced table is shown in Table 2.11.

Input x_1x_2	C = 1				
C = 0	00	01	11	10	z_1z_2
①	--	2	--	3	10
②	4	--	5	--	00
③	1	--	6	--	00
④	--	2	--	3	01
⑤	--	8	--	7	01
⑥	--	8	--	7	10
⑦	1	--	6	--	11
⑧	4	--	5	--	11

Table 2.11

Reduced PL Table for 2 x 2 AUD

There are 8 states and an assignment with the S.P. partitions

$$\pi_1 = \{\overline{1268}, \overline{3456}\}$$
$$\pi_2 = \{\overline{1478}, \overline{2356}\}$$
$$\pi_3 = \{\overline{1456}, \overline{2378}\}$$

leads to the following state and output equations.

$$Y_1 = y_1C' + [x_1x_2' + (x_1 + x_2')\, y_1']\, C$$
$$Y_2 = y_2C' + [x_1x_2 + (x_1 + x_2)\, y_2']C$$
$$Y_3 = y_3C' + y_3C$$
$$z_1 = y_2' y_3 + y_1' y_3'$$
$$z_2 = y_1y_3' + y_2' y_3$$

The assignment leads to similar state equations as for fundamental mode operation. If the don't care entries are filled in according to the entries of the primitive flow table, one of the assignments will be always the same as for fundamental mode operation. The state changes now take place under the control of the change pulse.

To realize the outputs at no cost, corresponding to Fig. 2.4, we require two output consistent partitions π_3 and π_4. Further to obtain the series-parallel decomposition, we need two S.P. partitions π_1 and π_2 such that

$$\pi_1 \cdot \pi_3 = \pi_A$$
$$\pi_2 \cdot \pi_4 = \pi_B$$

and $\pi_A \cdot \pi_B = 0$ [π_A and π_B are S.P. partitions]

For a fully specified table unless π_3 and π_4 have S.P., π_A and π_B will not have S.P. But the table 2.10 is incompletely specified and we can obtain π_A and π_B with S.P. For Table 2.10, the following partitions are obtained.

$$\pi_3 = \{\overline{1678}, \overline{2345}\}$$
$$\pi_4 = \{\overline{1236}, \overline{4578}\}$$
$$\pi_1 = \{\overline{1248}, \overline{3567}\}$$
$$\pi_2 = \{\overline{1347}, \overline{2568}\}$$
$$\pi_A = \{\overline{18}, \overline{24}, \overline{35}, \overline{67}\}$$
$$\pi_B = \{\overline{13}, \overline{26}, \overline{37}, \overline{58}\}$$

and $\pi_A \cdot \pi_B = 0$

The next state and output equations are

$$Y_1 = x_1; \quad Y_2 = x_2$$
$$Y_3 = y_1; \quad Y_4 = y_2$$
$$z_1 = y_3; \quad z_2 = y_4$$

Thus a very simple realization, with only delays, is obtained for the 2 x 2 AUD.

2.6.2 Pulse-Controlled Realizations of 2 x 2 AUD

We shall consider some specific realizations using different methods of change detection, storing the inputs and shifting the stored inputs to outputs. Each realization aims to reduce the delay involved in detecting a change. The methods are of very general nature and can be extended to n x n delays.

Change Detection:

Method 1 [Fig. 2.5] Assuming double-rail inputs, the circuit requires two differentiating networks per input. The time constant RC should be very small compared to the pulse width t_p i.e. $RC << t_p$ but the resulting pulse should have a duration which is at least equal to the response time of the shift circuit. There is an amplitude attenuation due to the RC network. Whenever the amplitude falls below the decision level of the circuits

following it, we require amplifiers. If the pulse input has sufficient amplitude, the OR-gate acts both as a pulse shaper and as an amplifier. The delay involved in detecting any change is due to OR-gate and the differentiating network.

Method 2 [Fig. 2.6]: In method 2, a monostable multivibrator is used in detecting an input change. A monostable multivibrator is triggered by a positive going change so that the output changes level for a fixed but adjustable period of time. Assuming double-rail inputs, the circuit requires two monostable multivibrators per input. The pulse duration can be controlled and in general the pulses are well defined. There is no amplitude attenuation and the delay involved in detecting a change is due to OR-gate and turning on time of a transistor.

Method 3 [Fig. 2.7]: This method utilises the inherent delay of devices for change detection. For level changes from 0 to 1 X_1 and X_1' (X_1 complemented and delayed) are fed to an AND gate. The output of the AND gate is 1 whenever X_1 and X_1' are 1. Similarly changes from 1 to 0 are detected by feeding X_1' and X_1'' (X_1' complemented and delayed) to another AND gate. The two outputs are taken through an OR-gate. This method has a disadvantage of unequal delays for levels going high and low and more than one inverter is required to obtain a sufficient pulse width.

Method 4 [Fig. 2.8]: This method uses the current-voltage characteristics of a tunnel-diode for change detection. The output depends upon I_L which changes as shown in Fig. 2.8. It is governed by the characteristics of the transistor emitter and tunnel diode in series. The input voltage is arranged to take the circuit from A to B or vice versa. If the input logic levels are

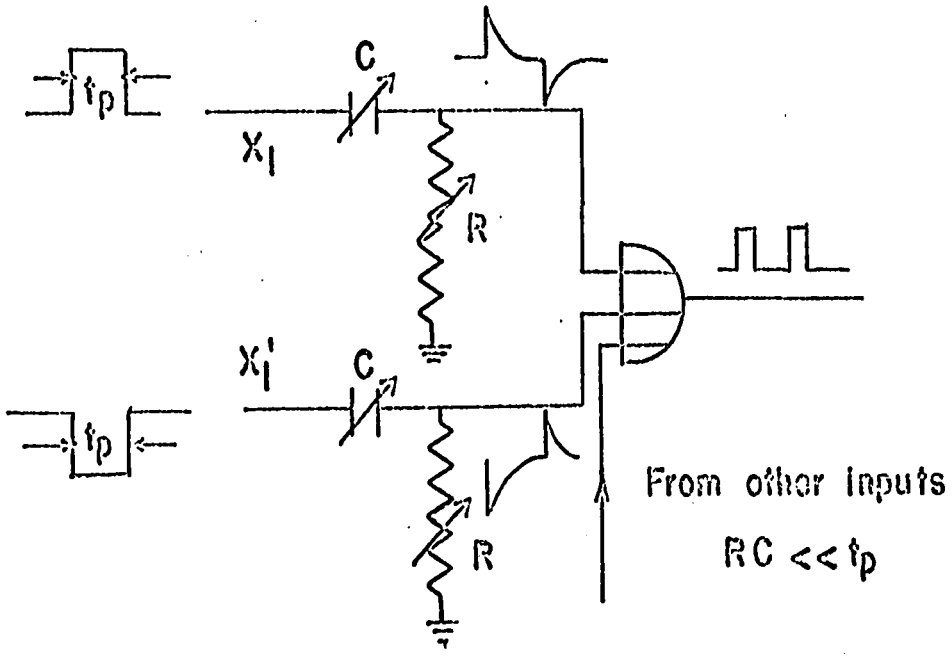


Fig. 2.5

Differentiating Network For Change Detection

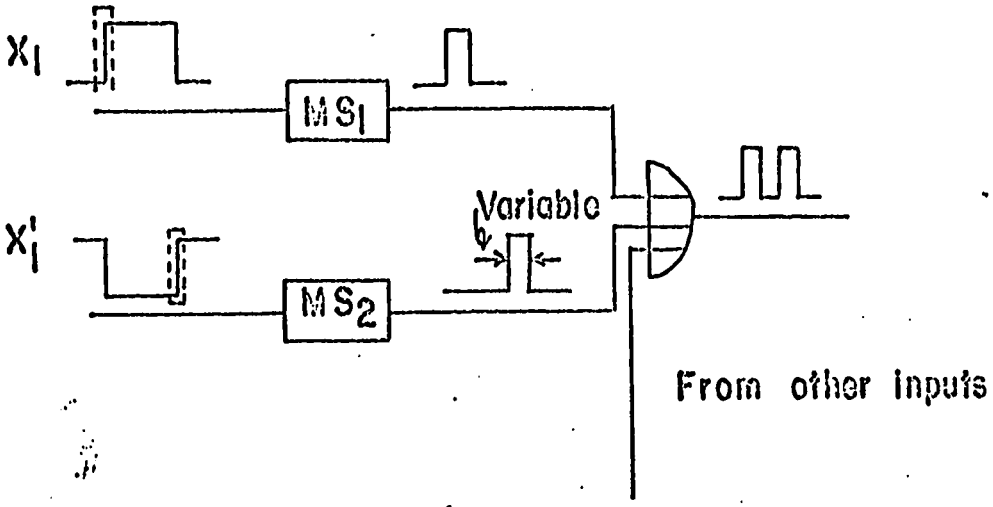


Fig. 2.6

Monostable Multivibrators for Change Detection

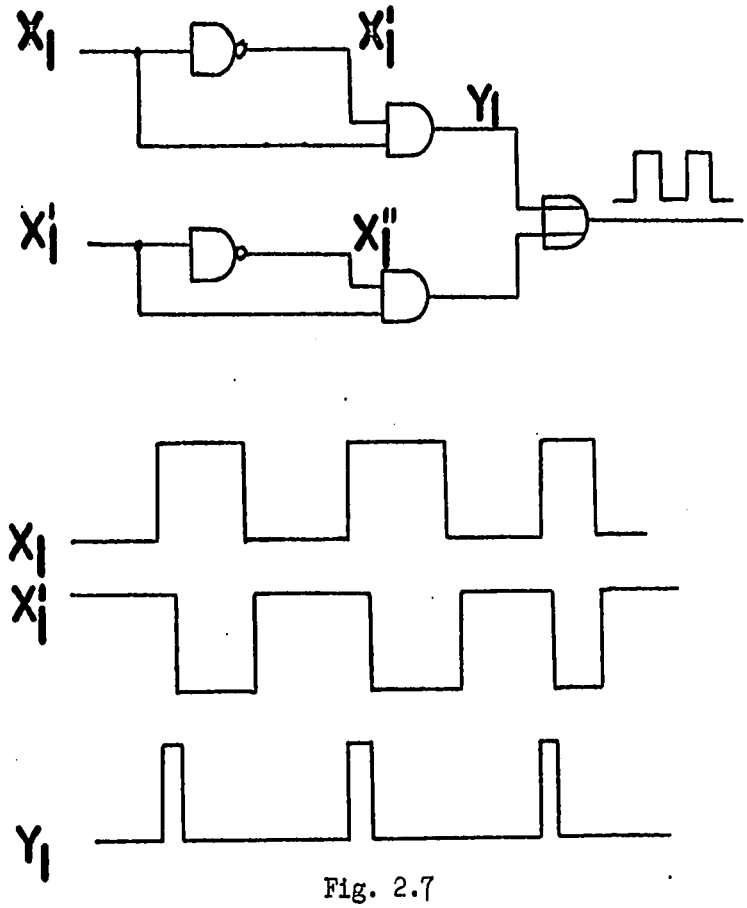


Fig. 2.7
Inverters For Change Detection

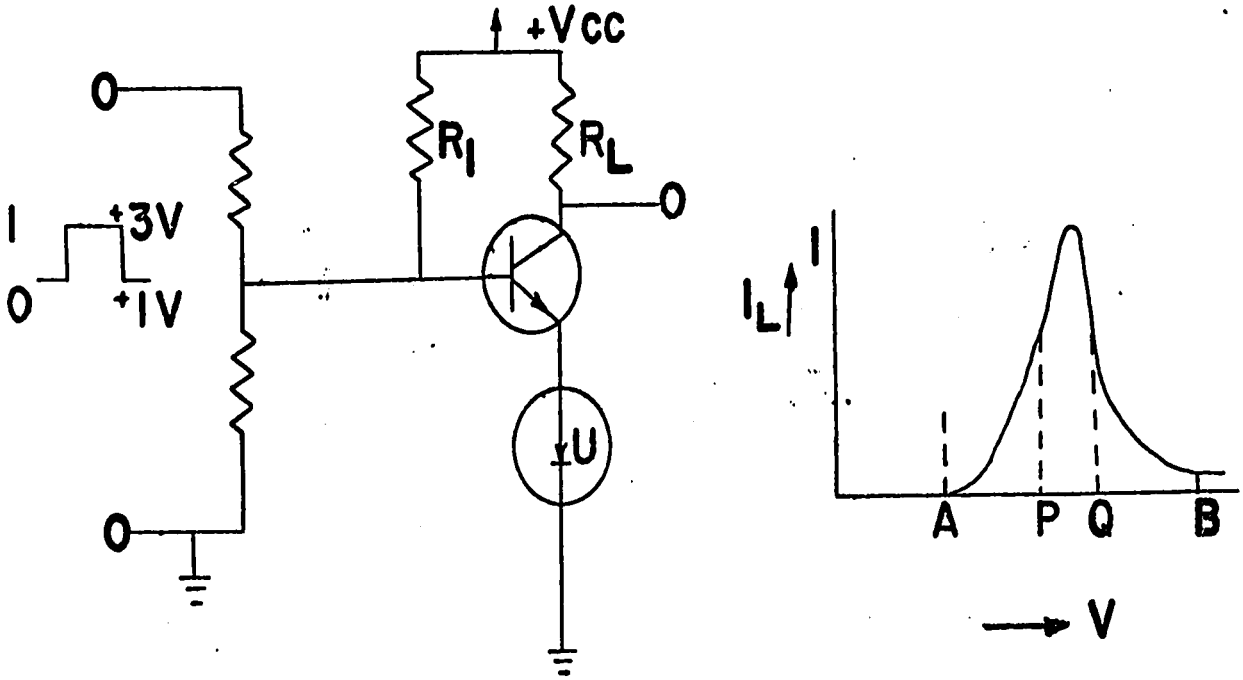


Fig. 2.8
Tunnel Diode For Change Detection

too large, they may be reduced by a simple potential divider arrangement Thus for every level change a pulse output corresponding to PQ is obtained. The delay involved in detecting a change corresponds to the interval AP or BQ. The pulse output may require further reshaping.

Shift Operation The output of change detector determines the time when the stored input must be transferred to the output. A flow table description for this circuit is as follows.

Input	C = 0		C = 1	
	X^d		X^d	
	0	1	0	1
1	1	1	1	2
2	2	2	1	2

(a)

Input	C = 0		C = 1	
	X^d		X^d	
y_1	0	1	0	1
0	0	0	0	1
1	1	1	0	1

Y_1

(b)

Input	C = 0		C = 1	
	X^d		X^d	
	0	1	0	1
0	00	00	01	10
1	00	00	01	10

S_1R_1

(c)

Table 2.13

Transition and Excitation Tables

The excitation and next state functions are

$$S_1 = x_1^d C$$

$$R_1 = x_1^{\prime d} C$$

$$Y_1 = x_1^d C' + y_1 C'$$

Realizations using S-R flip-flop and a feedback loop as bistable devices are shown in Figs.. 2.9 and 2.10.

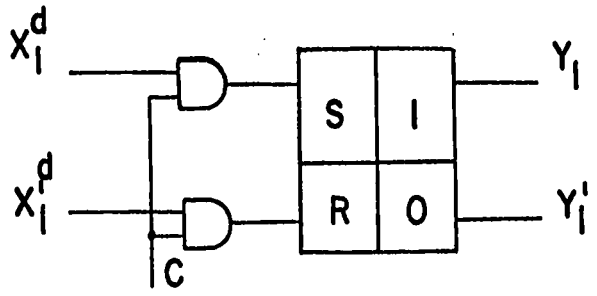


Fig. 2.9

Shift Circuit Using S-R Flip-Flop

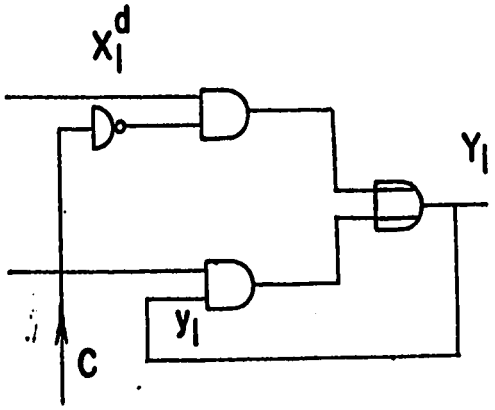


Figure 2.10

Shift Circuit Using Feedback Loop

Delaying the Input: There are many methods of delaying any signal e. g. an RC network, a delay line, a chain of inverters, etc. The choice of a particular device depends upon the frequency of operation, usage and the cost criteria. The delay can also be obtained by using a shift circuit similar to the one described above. The circuit will shift the input to memory under the control of the change pulse. Care has to be taken, however, to insure that the delay through shift circuit is greater than the duration of the change pulse. In a practical circuit the required tolerances may be difficult to achieve and additional gating must be used.

A few circuits using some of these techniques have been tried [Figs. 2.11, 2.12 and 2.13] and the results are presented in Chapter 4.

2.7 Comparison of Fundamental Mode and Pulse-Controlled Realization of n x n AUD

A comparison of 2 x 2 and 3 x 3 AUD designed under fundamental mode operation shows that whereas a 2 x 2 AUD requires 17 gates, a 3 x 3 AUD requires 46 gates. A state assignment has been obtained which realizes the circuits as a parallel combination of sub-circuits but the output logic is interdependent. Thus there are different paths from an input to output and it is difficult to insure equal path lengths for every change. For the realizations of asynchronous circuits using a cascade connection of such units, the transmission delays will keep on accumulating and the maximum delay of every stage will have to be considered to determine the input rate. The design procedure cannot be generalized to n x n AUD and for $n \geq 4$, the state tables become extremely complex.

The operation of AUD under pulse-controlled operation is very much akin to a clocked system; the difference being that the change pulses

are derived from the input rather than an external clock. Even though the flow tables contain more states than for fundamental mode operation, the state assignment is almost trivial. The only delay in any path is due to the change detector and the shift circuit. Also because there is only one path from any input to any output, all path lengths are equal and hence there are equal transmission delays.

There is no accumulation of delays in cascading such units and the input rate is not a function of the number of units in cascade. The pulse-controlled operation can be very easily extended to n x n AUD and the cost for every additional input is about the same. The circuit of Fig. 2.12 has the advantage of utilising identical circuits for shift operations and change detection which would simplify fault diagonosis and replacement.

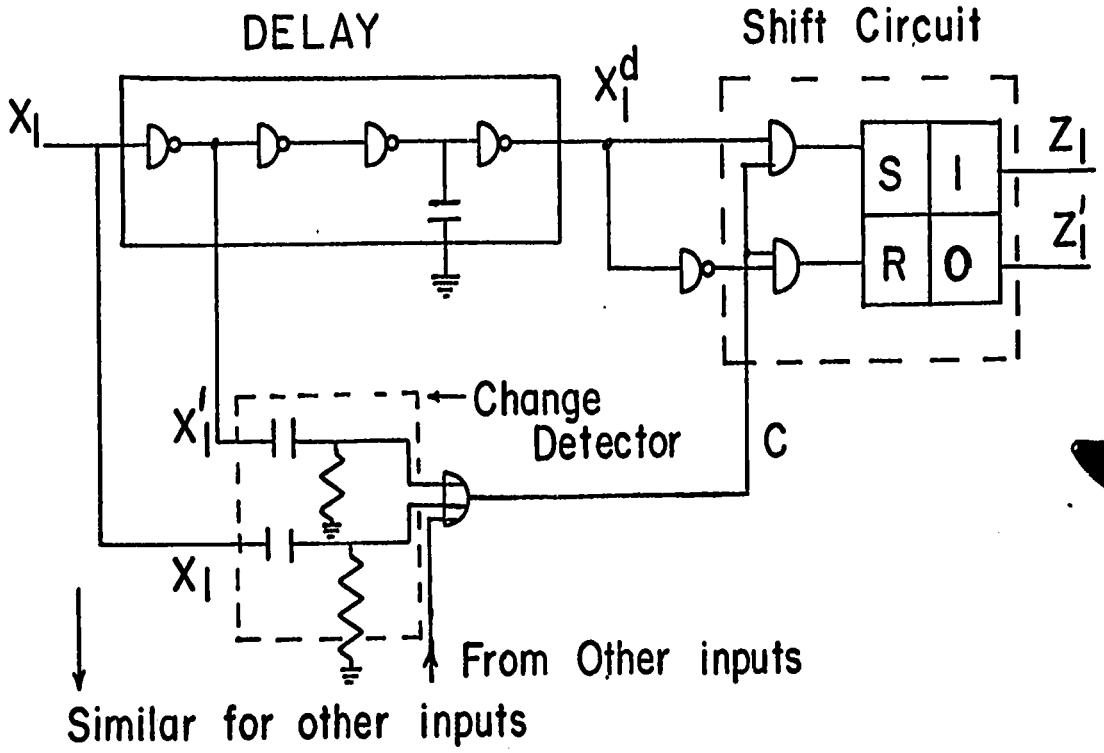

DELAY
Shift Circuit
X_I
X_I^d
S
I
R
O
Z_I
Z_I'
X_I'
Change
Detector
C
X_I
From Other inputs
Similar for other inputs

Fig. 2.11

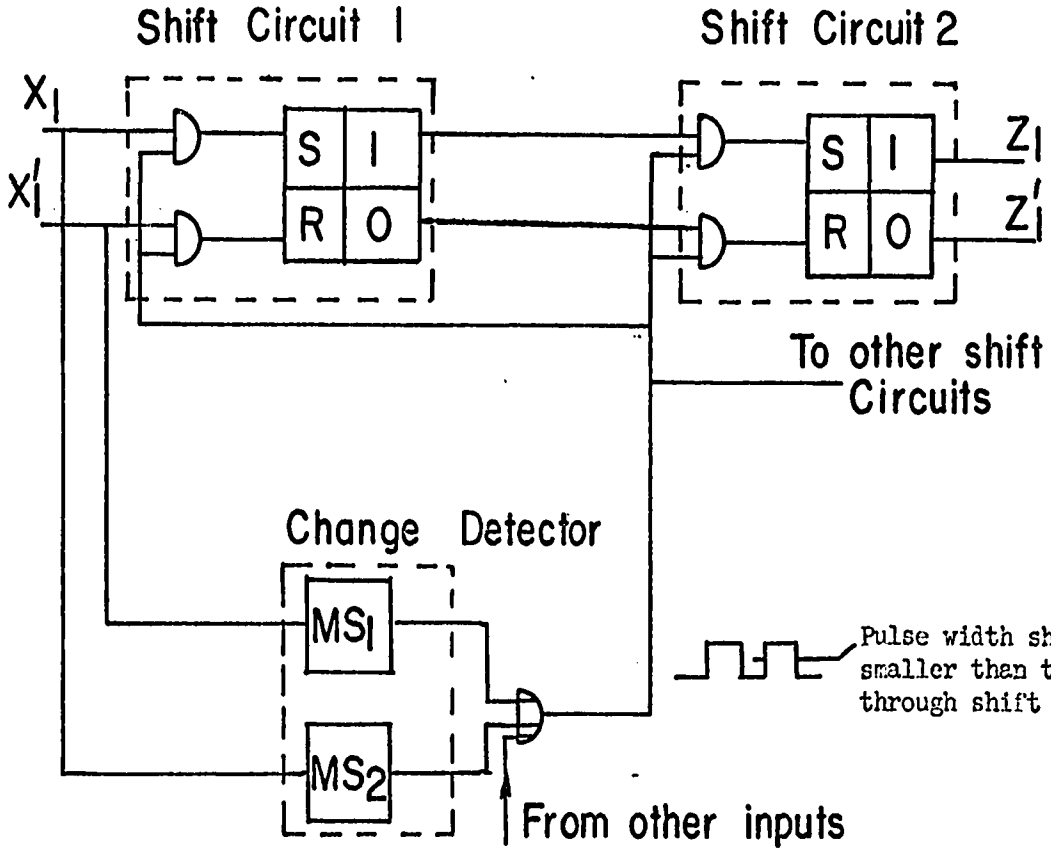


Fig. 2.12

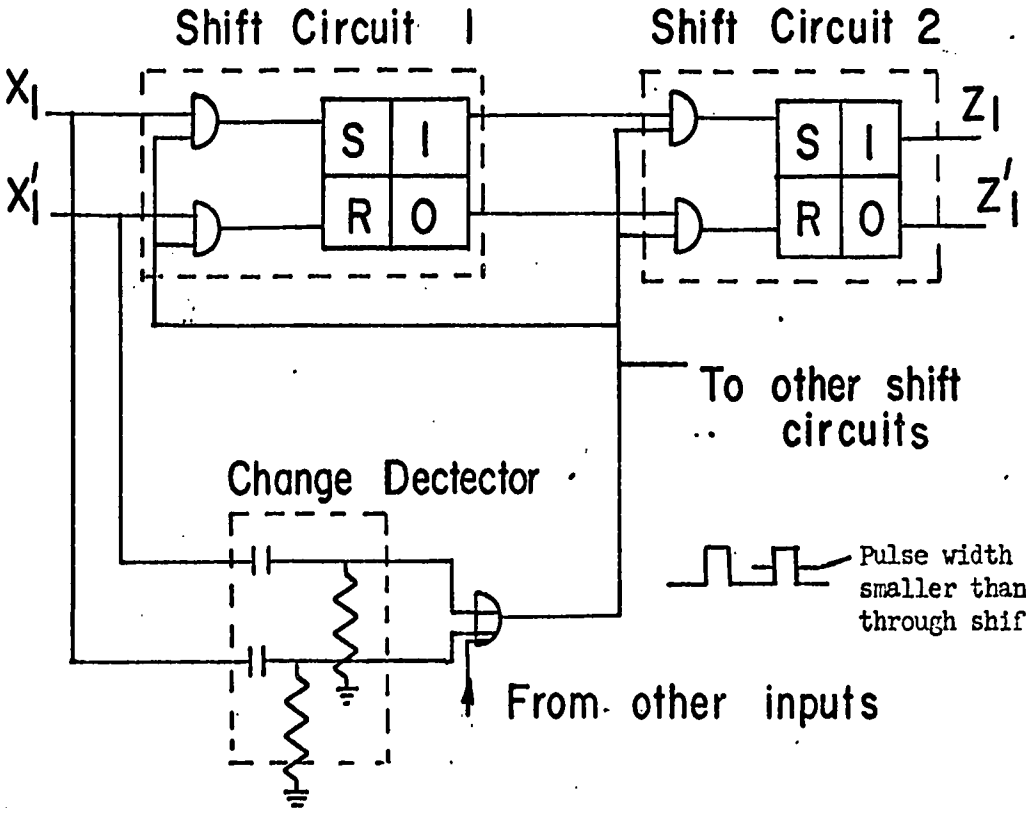


Fig. 2.13

CHAPTER 3

Circuit Analysis of Asynchronous Unit Delays

In this chapter we shall consider some of the fundamental concepts involved in the circuit design of an asynchronous unit delay and the performance tests on the different models operating both in the fundamental mode and under pulse-control. The tests are oriented towards the measurements of

(i) The proximity of input changes (i.e. the resolution time and the response times).

(ii) Maximum transmission delay in any path.

(iii) Different path lengths for input transitions.

(iv) Accumulation of delays in cascading two stages of an AUD.

(v) Distortion of the waveforms, etc.

The above tests have been performed using DEC (Digital Equipment Corporation system modules) and micro-circuits (Motorola Corporation). The experimental results and a comparison of the different models are summarized in chapter 4.

3.1 Switching Characteristics

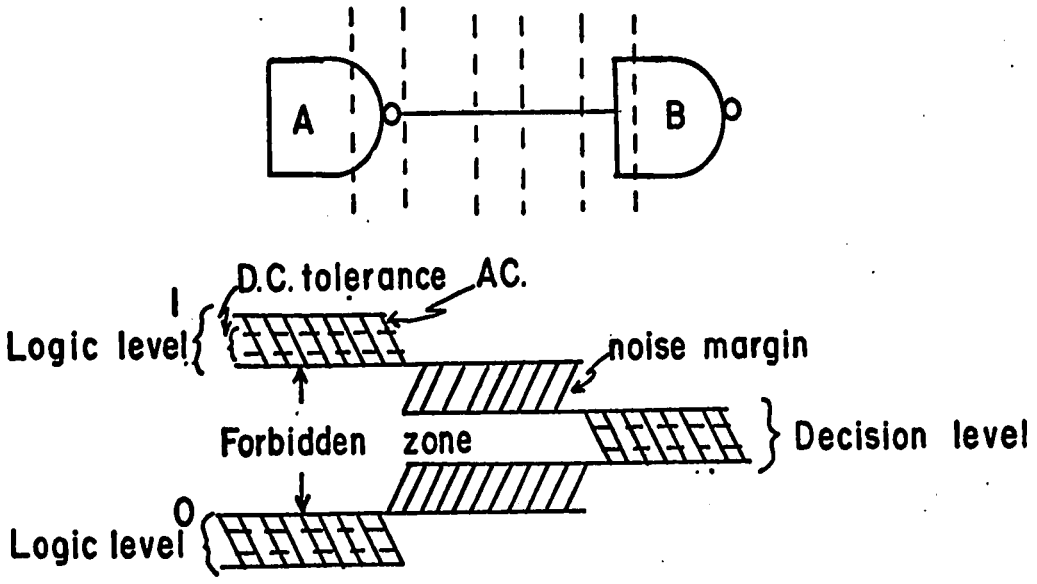
Logic Levels: The distinguishing feature of the circuits to be discussed here is the use of two-valued or binary signals. The statement that the signals are two-valued implies that under quiescent conditions, the value of each signal is within one of the two (non-overlapping) continuous ranges. To understand the switching characteristics of such circuits, we shall

consider in detail the

(i) connection between logic blocks and

(ii) behaviour of devices as they make transitions from one state to the other.

The discussion will be kept general in order to include all different circuit techniques (e.g. RTL, DTL, TTL, logic*) employed in the implementation of Boolean functions and operations. Referring to Fig. 3.1, consider the logic block 'A' feeding another logic block 'B'. The regions of interest are (i) The output of logic block A (ii) The connection between logic block A and B (iii) The input to logic block B. The output of logic block A must be constrained between the two band of voltages (or currents) i.e. the logic level 1 and 0. The region between these two tolerance zones will be called the forbidden zone through which the output passes during transitions. The input to logic block B differentiates between level 1 and level 0 by detecting whether the signal value



Output Transition Input

Fig. 3.1

Tolerance Zones For Logic Gates

* RTL - Resistor Transistor Logic DTL - Diode Transistor Logic
TTL - Transistor Transistor Logic

is above or below a certain threshold. This threshold is sometimes called the decision level and is not a discrete level but a range of levels for most practical circuits.

Noise and Noise Margins: The regions between the upper tolerance of decision level and lower tolerance of logic level 1 and the lower tolerance of decision level and upper tolerance of logic level 0 represent the noise margins. Thus, if the decision level were not in the centre of the forbidden zone, the noise margins for 0 and 1 would be unequal. The major sources of noise signals in practical systems are (i) signal currents flowing in common nodes (ii) ground noise and (iii) induced coupling from other signal lines. The noise generation characteristics must be considered in conjunction with the noise margins of the logical networks being used. If the voltage levels corresponding to 1 and 0 are widely separated, the signals will have a large noise margin, but low speed.

The tolerances for the different levels specified above consist of two components (i) Static (d.c.) tolerances. (ii) Dynamic (a.c.) tolerances. The d.c. tolerances depend upon the impedance (resistive part) of the diodes, transistors and other components and the currents that pass through them. The a.c. tolerances depend upon transient phenomena such as charging currents for circuit capacitances, (Fig. 3.2.) e.g. an output may approach its final value asymptotically or overshoot and ring before settling down to its steady state value. Further, the a.c. tolerances have a considerable effect on the transitions times. There are other factors affecting the d.c. and a.c. tolerances e.g. the drift in the power supply, loading effects, change in gain, the aging of components, etc.

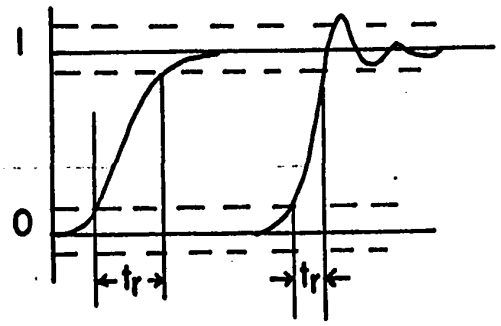


Fig. 3.2

Effect of Charging Currents on Transition Time

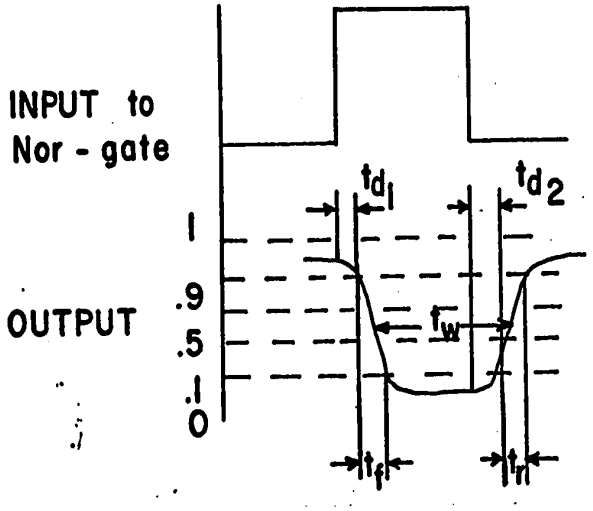


Fig. 3.3

Switching Times of a Nor-Gate

Switching Time of Gates

Consider a Nor-gate driven by a pulse waveform as shown in Fig. 3.3. It is assumed that the input pulse changes level in negligible amount of time. The output however, does not immediately respond to the input and there is a delay. The delay time (t_d) is defined as the time interval between the input level passing its decision level and the output signal entering the forbidden zone. Note that the value of the delay time is dependent upon the quiescent value of the output signal as related to its d.c. tolerances. The time for transition between higher voltage level and lower voltage level is called the fall time (t_f) and is measured from the lower tolerance of higher voltage to upper tolerance for lower voltage. The output during this time is indefinite. In practice, the a.c. tolerances on either level are taken as $\pm$10 percent. Thus for most practical systems the fall time is defined as the time interval from 90 percent of maximum amplitude on the trailing edge to 10 percent of maximum amplitude. Similarly rise time (t_r) is defined as the time interval from 10 percent of the maximum amplitude on the leading edge to 90 percent of the maximum amplitude.

On the input side, the duration for which the pulse is in the 0 state or 1 state depends upon the decision level, e.g. if the noise margins for both states are equal, the average pulse duration (t_w) is the time interval from 50 percent of maximum amplitude on the trailing edge to 50 percent of maximum amplitude on the leading edge.

A transition having been introduced by an input change, a certain minimum time must elapse before a succeeding input change will be able to introduce another transition reliably. This smallest allowable interval between changes is called the resolution time (t_p) for the circuit. To differentiate between successive changes on the same input line and

different input lines, the former will be called the response time (t_o) and the latter, resolution time. The resolution time consists of the transition time, defined earlier, and settling time (t_s) or recovery time, defined as the time for overshoots to decay to a value within the overall tolerances (Fig. 3.4). However, there is no clear cut separation between the transition phase and the settling phase.

The resolution time determines the maximum input rate for a device

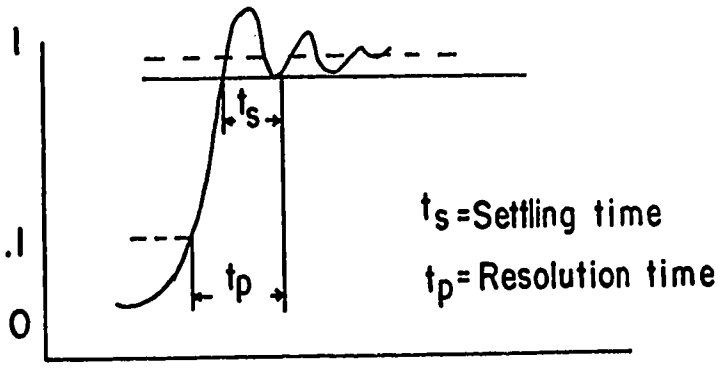


Fig. 3.4

under consideration. For a worst case design and no noise margins allowed, the maximum average input rate for a device is given by $\frac{t_r + t_f}{2}$. There are many methods of increasing the speed of a device and these are discussed extensively in the literature on Pulse, Digital and Switching waveforms [9].

3.2 Performance Tests

In this section we consider the procedure adopted in testing the parameters of an AUD.

(i) Pulse vs. Level Input: In design techniques discussed above, it was assumed that the binary input signals are levels. A level input, as defined earlier, assumes the value 0 or 1 for any length of time greater than some minimum duration t_o, which is the response time of the circuit. This minimum duration t_o can be discovered by decreasing the pulse width of an input waveform till the circuit operation is no longer satisfactory.

(ii) Resolution Time: For the fundamental mode operation of the AUD, it is assumed that only single input changes are allowed. Due to different speeds through the circuit, any two changes occuring simultaneously at the input will not affect the output at the same instant. Thus for rest of the circuit they appear as having occured at different instants of time. To determine the resolution time for an AUD operating in fundamental mode, consider the following two cases.

(i) Only one input is allowed to change while others are kept fixed at a constant D.C. value, during this period.

(ii) All inputs are allowed to change.

Case (i) is concerned with finding the response time of a circuit. Unless some care is exercised in choosing components, the response time varies depending on what input is changed. Thus to insure proper operation, the minimum duration for any pulse should be greater than the maximum response time. Case (ii) determines the shortest time 't_p' between input changes not necessarily on the same line. If these changes affect the output through different paths, t_p also depends on the order in which inputs are changed (Fig. 3.5).

The circuit models designed to operate in the pulse-control mode do not have the restrictions against simultaneous changes. In case, two inputs change at the same instant, the pulses generated, superpose and are recognized as a single pulse for the shift circuit. However, if the changes do not appear at the same time, the minimum interval between them is the sum of delay through change detector and the duration of the change pulse. In between, the circuit output is ambiguous and mainly depends on the delay in the input lines.

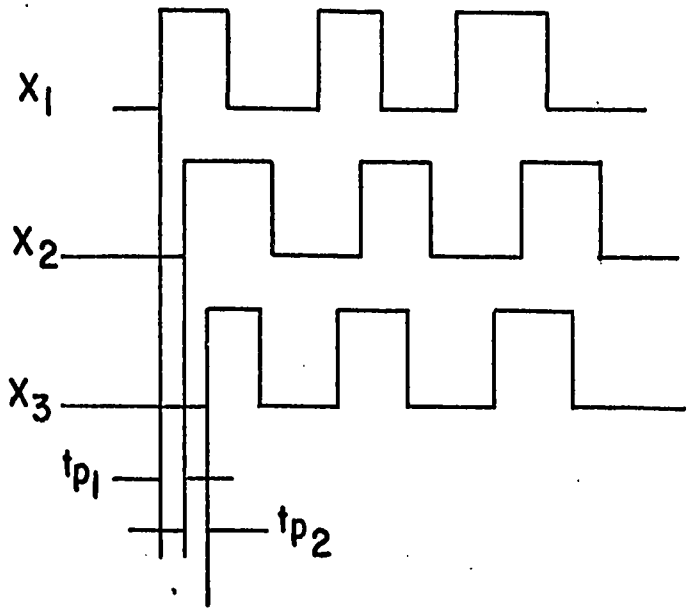


Fig. 3.5

Showing the Proximity of Input Changes

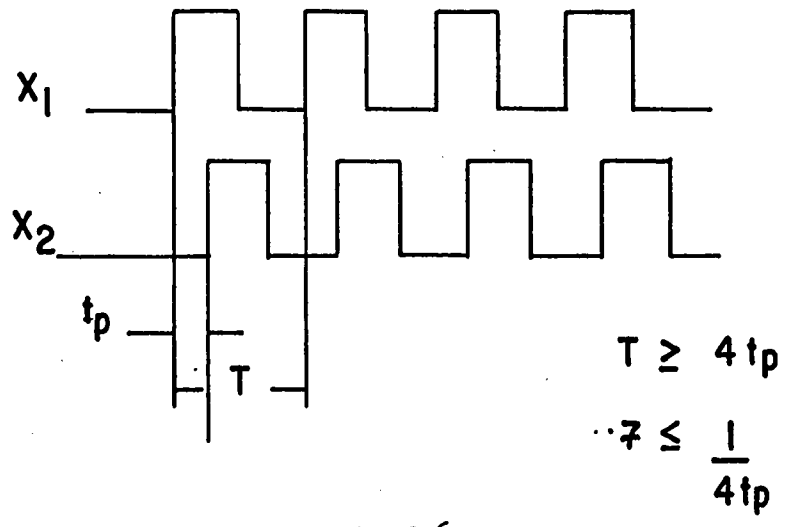


Fig. 3.6

Frequency as a Function of Number of Inputs

(iii) Speed of Operation: The restriction of single input changes puts an upper limit on the maximum input rate. The overall frequency for the AUD is defined as the maximum frequency for which the circuit responds to all input changes. For the fundamental mode operation, this is a function of the response time t_o and since response time varies for different inputs, the overall frequency has only an upper bound. Line frequency, defined as the maximum frequency at which any input can be changed in presence of any other input, is inversely proportional to the number of inputs (Fig. 3.6). [under the restriction of single input changes]. For n inputs, the line frequency $f \leq \frac{1}{2nt_p}$ and decreases rapidly with increase of number of inputs.

For pulse-controlled operation, under the restriction of single input changes, the speed of operation does not change with the number of inputs. It depends only on the delay through the change detector and the duration of the change pulse.

(iv) Transmission Delays: The transmission delays are a convenient measure of the speed of operation of a circuit. The delay between two successive waveforms is measured at a specified level, generally the decision level. For the fundamental mode operation of the AUD, the transmission delays vary depending upon the input transition. Table 3.1 lists all the possible transitions from any input to any output.

Following notations are introduced.

↑ implies the particular line is changing level from 0 to 1.

↓ implies the particular line is changing level from 1 to 0.

✓ Transition possible.

X Transition not possible.

Table 3.1 shows that there are 12 different transitions possible. The transmission delay for any transition will depend upon the path through the circuit. For an n x n AUD, there are 2n(2n-1) possible transitions and possible delay values.

Inputs / Outputs	X_1 ↑	X_1 ↓	X_2 ↑	X_2 ↓
Z_1 ↑	X	✓	✓	✓
Z_1 ↓	✓	X	✓	✓
Z_2 ↑	✓	✓	X	✓
Z_2 ↓	✓	✓	✓	X

Table 3.1

Possible Transitions for a 2 x 2 AUD

Under pulse-controlled operation, there are similar paths from any input to any output and the delay for any transition is constant.

(v) Distortion: Distortion and external disturbances can be tolerated until they start affecting the proper operation of a circuit. In the case where many similar units are connected in series, the distortion may accumulate and after a number of stages, pulse shaping networks need to be used.

The waveform distortion can be due to amplitude attenuation, different delays for the rising and falling edges, noise, interaction between circuit components, loading etc. In general, for transistor sequential circuits, there is no amplitude attenuation because there is amplification and pulse shaping at each stage. But for circuit models described in Figs. 2.11, 2.12 and 2.13, the signals at any point should not fall below the decision level of the circuit following them.

3.3 Cascading of 2 AUD's

Consider two identical AUD's (operating in the fundamental mode) in cascade. Due to differences in circuit delays, the resolution time can alter. So long as the interval between two input changes is greater than or equal to the resolution time of the following stage, the circuit will respond to both changes equally well. If, however, after a number of stages this interval is less than t_p, the circuit may give a false output. (Fig. 3.7). Hence for the circuit model of Fig. 2.1, in the fundamental mode operation, it is not sufficient for changes to occur t_p apart but the period t must be given by

$$t \geq t_p + (K-1)(\eta_1 - \eta_2)$$

$(\eta_1 - \eta_2)$ is the time difference in delays of two successive changes.

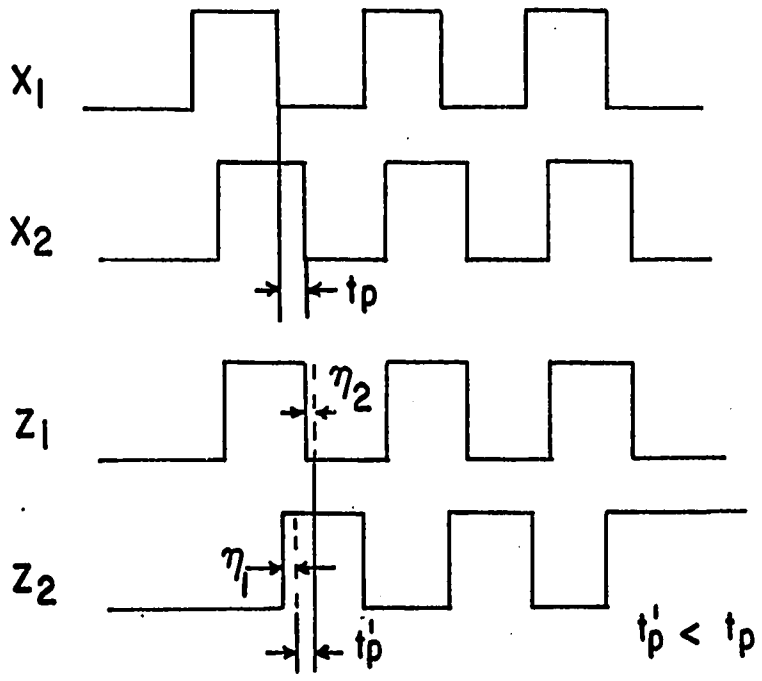
K = Number of stages in cascade.

For $(\eta_1 - \eta_2) > o$, the minimum interval t increases with the number of AUD's in cascade and the overall frequency of operation will be reduced.

For circuits operating under pulse-control, the transmission delay is a constant and the overall frequency is not affected by the number of units in cascade.

3.4 Calculation of Delays for a Single Stage and for Two Stages of a Shift Register Under Fundamental Mode Operation.

The circuit of Fig. 2.1 has been analysed for the maximum delay in each path. Table 3.2 lists all the values of delays that can occur and a comparison with the experimental results shows good agreement.



t_p Minimum interval between changes for first stage

t_p' Minimum interval between changes for second stage

Fig. 3.7

Effect of Transmission Delays on Resolution Time

Outputs \ Inputs	$X_1\uparrow$	$X_1\downarrow$	$X_2\uparrow$	$X_2\downarrow$
$Z_1\uparrow$	X	$2\downarrow$ $1\uparrow$	$2\uparrow$ $2\downarrow$	$3\uparrow$ $2\downarrow$
$Z_1\downarrow$	$2\uparrow$ $1\downarrow$	X	$3\uparrow$ $2\downarrow$	$2\uparrow$ $2\downarrow$
$Z_2\uparrow$	$2\uparrow$ $2\downarrow$	$2\downarrow$ $3\uparrow$	X	$2\downarrow$ $1\uparrow$
$Z_2\downarrow$	$3\uparrow$ $2\downarrow$	$2\uparrow$ $2\downarrow$	$2\uparrow$ $1\downarrow$	X

$\uparrow$ Level 0 to 1
$\downarrow$ Level 1 to 0

Table 3.2

Maximum Transmission Delay for Different Transitions in 2 x 2 AUD.

Outputs \ Inputs	$X_1\uparrow$	$X_1\downarrow$	$X_2\uparrow$	$X_2\downarrow$
$Z_3\uparrow$	3	2	2	2
$Z_3\downarrow$	2	3	2	2
$Z_4\uparrow$	2	2	3	2
$Z_4\downarrow$	2	2	2	3

Table 3.3

Delay Paths for 2 Stages of 2 x 2 AUD

If the delay for levels going high and low is assumed to be the same, the minimum transmission delay involved in any path is 3 units and the maximum transmission delay is 5 units. For the circuits under pulse-control, the minimum and maximum transmission delay is 5 units and is constant for any number of stages in cascade. Table 3.3 lists the number of delay paths involved in cascading two stages of the circuit model of Fig. 2.1. For two stages there are as many as 36 different paths and for 3 stages in cascade there are 87 different paths. Thus the interdependence of outputs leads to a complex structure of identical modules. On the other hand, the design considered under pulse-control leads to relatively simple realizations for nxn AUD.

CHAPTER 4

Experimental Results and Discussion

4.1 Discussion of Results

The following is a discussion of results obtained from the various experiments described above. The experiments were carried in fair detail to establish the validity and existence of the AUD which is a basic building block of asynchronous circuits. The experiments provide estimates of parameters rather than an exact measure of the circuit operation. The criterion for selecting a particular model will depend on the application, speed of operation required and technology available. For instance, circuits using micro-logic were found to be cheap and fast. However, DEC modules were found to be versatile and suitable for experimentation.

Table 4.1 lists the results of the various experiments. An AUD constructed out of the micro-circuits is about 10 times faster than the DEC modules used. The transmission delays of the micrologic are about half that of the DEC modules but the maximum transmission delay for the micrologic AUD is about twice the minimum delay due to widely differing rise time and fall times. Two stages of AUD operating in the fundamental mode, were cascaded and the total delay from an input to output was found to depend upon the path through the individual units. For worst case design (maximum speed), the maximum delay of each unit has to be considered as the circuit delay.

Circuits 1, 2 and 3 (Figs. 2.11, 2.12 and 2.13), operating under pulse-control were found to have comparatively larger delays than fundamental mode circuits but the delays were constant. So long as the criteria for level signals is met, simultaneous changes can be allowed. The major advantage of pulse-control operation lies in the independence of inputs and the reduction in logical hardware required for extra inputs. The circuit of Fig. 2.12 was the best scheme analysed.

4.2 Description of the Equipment

DEC Equipment (System Modules): DEC system modules are packaged electronic circuits for use in the design and construction of digital circuits. Most logical operations with digital modules are performed with saturating PNP transistors. All DEC modules use dual polarity level logic and logical voltage levels are -3 volts and ground, corresponding to logic state 1 and 0. A positive diode nor gate (nand gate for negative logic) has a typical total transition time, for output rise as 0.13 usecs and, for output fall, 0.07 usecs. The maximum frequency of operation is 1Mc/s. The input pulse duration is of the order of 0.04 usecs. The modules use -15 volts/ 88mA power supply. Fan in and fan out varies with the individual modules but up to 8 units of load can be driven by each output.

Nand gates were found to have the least transmission delay (0.05 usecs average for both rising and falling edges) and were used in constructing the circuit of AUD.

Input Sequence Generation: A clock pulse (DEC module no. 1404) is used to trigger a monostable multivibrator (module no. 1304) which switches from its normal ground level to -3 volts for a predetermined but adjustable period of time (Fig. 4.1). The positive pulse output is made to trigger

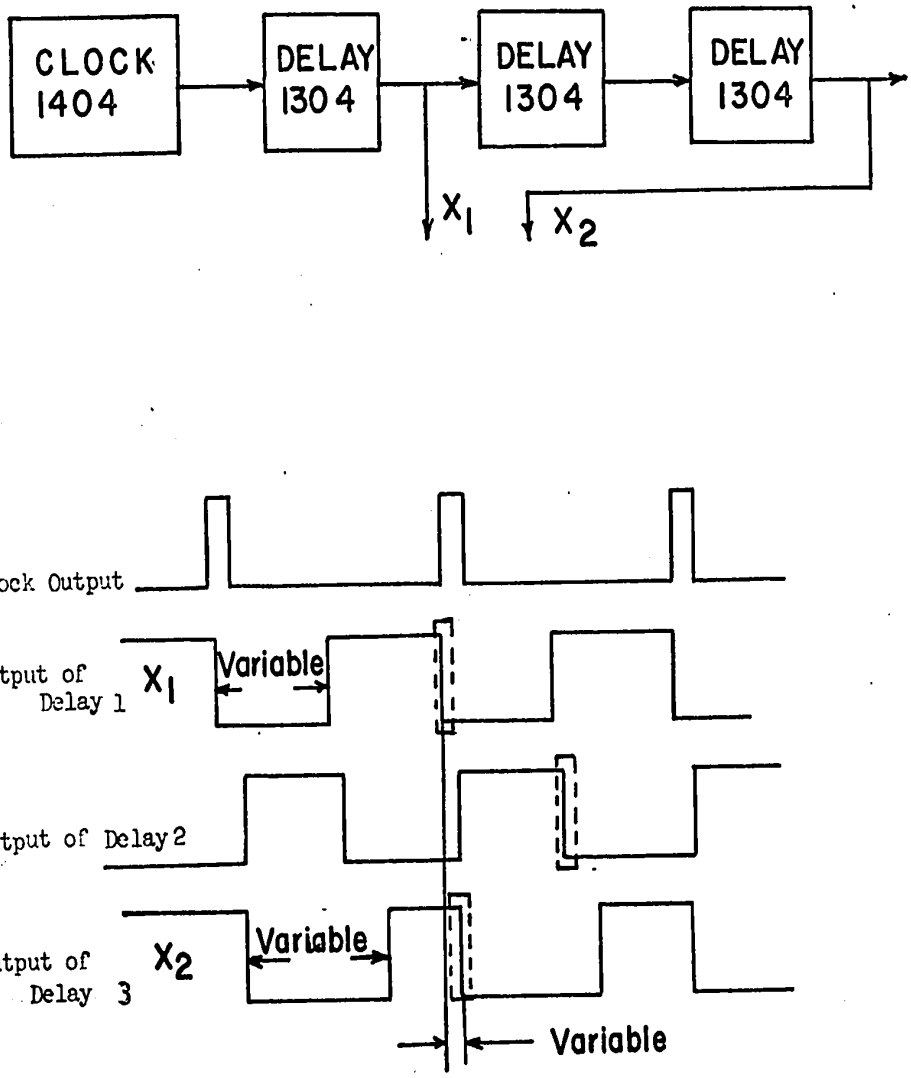


Fig. 4.1

Input Sequence Generation

another similar multivibrator with adjustable pulse width. The positive pulse output triggers a third monostable whose output can be delayed with respect to the output of the first multivibrator for a predetermined period of time. The output of the first and third multivibrators are used as the inputs to the AUD.

Output Display: The outputs of the modules were fed to a dual beam oscilloscope (Type 555 Tektronix Inc.) with the provision for synchronization of the two inputs.

Micro-Circuits: The monolithic integrated diode transistor logic (DTL) circuits are designed for low power high speed applications. Typical propagation delay is 80 nsecs and they are about 10 times faster than the DEC system modules. The integrated circuits use a power supply of +5.0 volts and use dual polarity level logic like the system modules. The logical level voltages are 0 volts and +3 to +5 volts. The nand gates have a fan out of 8 and fan in of 3.

Because of their small size and high speed, the circuits were made on a printed circuit board. For a comparison between the various circuit parameters, the same input was used for all experiments but to drive micro-circuits, a clamping circuit had to be used. The loading of the modules becomes evident when two stages are cascaded and there is waveform distortion. The modules were found to have widely different transmission delays for rising (25 nsecs) edge and for falling edge (80 nsecs).

The output display for all the circuits is similar to one described above.

Experimental Results for 2 x 2 AUD (Table 4.1)

Type of Test	Fundamental Mode Operation		Pulse-Controlled Operation			Remarks
	DEC Fig. 2.1	Micro-Logic Fig. 2.1	Circuit 1 Fig. 2.11	Circuit 2 Fig. 2.12	Circuit 3 Fig. 2.13	
. Minimum Pulse Width For which tested (available) (Response Times)	0.25 usecs	0.25 usecs	Does not operate below 0.30 usecs	0.25 usecs	Does not operate below 0.30 usecs	The response times for circuit 1 and 3 are limited by the change detector. For others, it cannot be determined due to limitations on input circuitry.
. Proximity of Input Changes (Resolution Times)	0.14 usecs	0.04 usecs	0.18 usecs	0.16 usecs	0.18 usecs	The performance of circuits 1, 2 and 3 is satisfactory for simultaneous changes also.
. Speed of Operation						
i) Line Frequency	1.8 Mc/s	6.25 Mc/s	1.4 Mc/s	1.5 Mc/s	1.4 Mc/s	These values are theoretical estimates. The values are approximated because of finite rise and fall times of the waveforms, no exact measurements were possible.
ii) Overall Frequency	3.6 Mc/s	12.5 Mc/s	1.6 Mc/s	3 Mc/s	1.6 Mc/s	
. Transmission Delays						
$X_1 \uparrow Z_1 \downarrow$	0.15 usecs	0.08 usecs	0.18 usecs	0.18 usecs	0.18 usecs	Max Delay
$X_1 \downarrow Z_1 \uparrow$	0.14 usecs	0.014 usecs	0.24 usecs	0.18 usecs	0.22 usecs	DEC 0.24 usecs
$X_2 \uparrow Z_2 \downarrow$	0.14 usecs	0.08 usecs	0.18 usecs	0.18 usecs	0.18 usecs	Micrologic 0.14 usecs

Table 4.1 (Contd)

Type of Test	DEC	Micro-logic	Circuit 1	Circuit 2	Circuit 3	Remarks
$X_2 \downarrow$ $Z_2 \uparrow$	0.15 usecs	0.14 usecs	0.24 usecs	0.18 usecs	0.22 usecs	Circuit 1 0.24 usecs
$X_1 \uparrow$ $Z_2 \uparrow$	0.20 usecs	0.14 usecs				Circuit 2 0.18 usecs
$X_1 \uparrow$ $Z_2 \downarrow$	0.22 usecs	0.08 usecs				Circuit 3 0.22 usecs
$X_1 \downarrow$ $Z_2 \uparrow$	0.18 usecs	0.08 usecs				
$X_1 \downarrow$ $Z_2 \downarrow$	0.18 usecs	0.14 usecs	do	do	do	
$X_2 \uparrow$ $Z_1 \uparrow$	0.20 usecs	0.14 usecs				
$X_2 \uparrow$ $Z_1 \downarrow$	0.24 usecs	0.08 usecs				
$X_2 \downarrow$ $Z_1 \uparrow$	0.22 usecs	0.08 usecs				
$X_2 \downarrow$ $Z_1 \downarrow$	0.18 usecs	0.14 usecs				
Distortion	Negligible	Waveform distortion due to widely different values of rise (25 nsecs) and fall (80 nsecs) times	Negligible	Negligible	Negligible	Loading effects are specially prominent for micro-logic
2 Units of AUD in cascade						
(i) Proximity of Input Change (Resolution Times)	---	Varies from 0.04 to 0.1 usecs	---	0.16 usecs	---	The resolution times for circuits operating in fundamental mode were found to depend upon the input transitions and the values of η_1 and η_2 (Section 3.3). But for circuit 2 under pulse-control, the resolution time is a constant.
(ii) Speed of Operation						
(i) Line Frequency	---	2.5 Mc/s	---	1.5 Mc/s	---	

Table 4.1 (Contd)

Type of Test	DEC	Micro-logic	Circuit 1	Circuit 2	Circuit 3	Remarks
(ii) Overall Frequency	---	5 Mc/s	---	3 Mc/s	---	Theoretical estimates
(iii) Transmission Delays	---	Minimum Delay for any change = 0.22 usecs Maximum Delay for any change = 0.28 usecs	---	Minimum Delay = 0.18 usecs Maximum Delay = 0.18 usecs	---	The delay values are found to vary between the two extremes of 0.22 usecs and 0.28 usecs
7. (i) Cost for 2 terminals	19 gates	19 gates	24 gates	22 gates	22 gates	Change detector is approximated to 2 gates
(ii) Cost for 3 terminals	42 gates	42 gates	36 gates	33 gates	33 gates	
(iii) Cost for Additional Terminals	Depends upon the state assignment used and has to be optimized for every assignment		12 gates	11 gates	11 gates	

CHAPTER 5

Final Remarks - Conclusions

In conclusion, the design of asynchronous unit delays has been studied. Two methods of design are proposed, namely, the fundamental mode operation and the pulse-controlled operation. In the fundamental mode the circuit action is completely asynchronous and the circuit will operate as fast as switching can be accomplished. There are problems of races and hazards. The input rate depends upon the number of inputs and the number of units in cascade. The pulse-controlled operation is partially synchronous. An internally derived change pulse performs the clock operation. Thus all races can be neglected and any assignment can be used. A transformation to obtain an equivalent synchronous flow table from an asynchronous flow table is proposed and applied to the design of the AUD. A few circuit models (under both designs) have been described and experimentally tried. From this, estimates of certain parameters were obtained which were used to compare the two designs. It was demonstrated that, when the AUD was constructed using circuits designed for the fundamental mode of operation, it was slightly faster in the fundamental than in the pulse mode. Thus in order to gain a speed improvement from the pulse mode, special pulse mode circuits must be designed. However, the pulse-controlled operation has all the advantages of an orderly design. Also, the design procedure can be extended to n x n case and the cost for extra inputs is a linear function of the number of inputs.

REFERENCES

(1) Brzozowski, J.A. and S. Singh. Definite Asynchronous Sequential Circuits. Electrical Engineering Department, University of Ottawa. Technical Report No. 67-3. February, 1967.

(2) Cadden, W.J. Equivalent Sequential Circuits. IRE Trans On Circuit Theory, Vol. CT-6, No. 1, pp. 30-34. March, 1959.

(3) Eichelberger, E.B. The Synthesis of Combinational Circuits Containing Hazards. Digital Systems Laboratory, Electrical Engineering Department, Princeton University. Technical Report No. 17. March, 1962.

(4) Eichelberger, E.B. Sequential Circuit Synthesis Using Hazards And Delays. Digital Systems Laboratory, Electrical Engineering Department, Princeton University. Technical Report No. 19. June, 1962.

(5) Friedman, A.D. Feedback in Asynchronous Sequential Circuits 1965 IEEE Conference Record on Switching Circuit Theory and Logical Design, pp. 94-104. October, 1965.

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

(7) Hartmanis, J. and R.E. Stearns. Algebraic Structure Theory of Sequential Machines. Prentice-Hall 1966.

(8) McCluskey, E.J. Introduction to the Theory of Switching Circuits. McGraw Hill Inc. 1965. pp. 180-184.

(9) Millman, J. and H. Taub. Pulse, Digital and Switching Waveforms McGraw Hill Inc. 1965. pp. 745-791.

(10) Paull, M.C. and S.H. Unger. Minimizing the Number of States in Incompletely Specified Sequential Switching Functions. IRE Trans on Electric Computers, Vol. EC-8, No. 3, pp. 356-367, September, 1959.

(11) Unger, S.H. Hazards and Delays in Asynchronous Sequential Switching Circuits. IRE Trans On Circuit Theory, Vol. CT-6, No. 1, March, 1959.

(12) Unger, S.H. A Study of Asynchronous Logical Feedback Networks. Research Laboratory of Electronics, M.I.T. Cambridge (Mass.). Technical Report 320 (1957).

VITA

Name: Vinod K. Batra

Born: October 1, 1943. Lahore (W. Pakistan)

Educated:

Intermediate - Punjab University, India
1960, F.Sc (Non-Medical)

Engineering - Indian Institute of Technology
Madras, India
Feb., 1965 Bachelor of Technology
(Option - Light Current)