

Easily Testable Sequential Machines with Extra Inputs

HIDEO FUJIWARA, MEMBER, IEEE, YOICHI NAGAO, TSUTOMU SASAO, STUDENT MEMBER, IEEE, AND KOZO KINOSHITA, MEMBER, IEEE

Abstract—In this paper, an easily testable machine is defined as one which possesses: 1) a distinguishing sequence of length $\lceil \log_2 n \rceil$ which forces the machine into a specific state S_1 , and 2) transfer sequences of length at most $\lceil \log_2 n \rceil$ to carry the machine from state S_1 to state S_i for all i . A design procedure is presented in which an arbitrary machine is augmented to an easily testable machine by adding two special input symbols to the original machine. An efficient procedure is also described for designing checking experiments for the easily testable machines. For an n -state, m -input symbol machine, this procedure gives a bound on the length of the checking experiment that is approximately $mn \lceil \log_2 n \rceil$. Furthermore, the total checking experiments are preset.

Index Terms—Checking experiments, distinguishing sequences, easily testable machines, fault detection, sequential machines, shift register, transition checking.

I. INTRODUCTION

FOR sequential machines several authors [1]–[6] have considered the fault detection problem as an identification problem of sequential machines, that is, finding an input-output sequence which describes a given machine uniquely. A number of these papers are based on a method given by Hennie [2] for designing checking experiments, called the *transition checking approach*. His method yields good results for machines that possess a distinguishing sequence, and for machines that are reduced, strongly connected, and such that the actual machine has no more states than the correctly operating machine. However, for machines which do not have any distinguishing sequences, Hennie's procedure yields very long experiments, which makes it impractical. Therefore, several methods have been proposed for modifying a given sequential machine into a new one for which a short checking experiment can easily be found [3], [7]–[13]. These include: 1) a method of adding extra outputs [7], [8], and 2) a method of adding extra inputs [9]–[11]. For an n -state m -input symbol machine, the former gives a bound on the length of checking experiments that is approximately mn^3 , and the latter gives a bound of mn^2 .

This paper describes a method to augment an arbitrary machine to an easily testable machine by adding two special input symbols, and gives an efficient procedure to construct a checking experiment for it. For an n -state, m -input symbol machine, this procedure gives a good bound on the length of checking experiments that is ap-

proximately $mn \lceil \log_2 n \rceil$, where the square brackets denote "the smallest integer greater than or equal to the number inside the brackets." Furthermore, the total checking experiment is *preset* and thus requires no *adaptive* initializing sequence that adaptively brings the machine under test to the starting state.

II. NOTATION AND BASIC DEFINITIONS

The sequential machines considered in this paper are assumed to be finite state, synchronous, and deterministic Mealy machines, and are not required to be reduced, strongly connected, or completely specified. The machine M will be represented by a quintuple $M = (S, I, 0, \delta, \lambda)$ where $S = \{S_1, S_2, \dots, S_n\}$ is a finite set of states, $I = \{I_1, I_2, \dots, I_m\}$ is a finite set of input symbols, $0 = \{0_1, 0_2, \dots, 0_i\}$ is a finite set of output symbols, $\delta: S \times I \rightarrow S$ is called the next state function, and $\lambda: S \times I \rightarrow 0$ is called the output function.

A *checking experiment* is an input-output sequence which when the input sequence is applied to the tested machine, an output sequence is produced which establishes whether or not the tested machine is equivalent to the correctly operating machine, subject to some fault assumptions.

An experiment is said to be *adaptive* or *preset* depending on whether the next input signal to apply is or is not based upon the output signals previously produced by the machine.

A *synchronizing sequence* for a sequential machine is an input sequence whose application is guaranteed to leave the machine in a certain final state, regardless of the particular initial state of the machine.

A *homing sequence* for a sequential machine is an input sequence whose application makes it possible to determine the final state of the machine by observing the corresponding output sequence that the machine produces. A *distinguishing sequence* is an input sequence whose application makes it possible to determine the initial state of the machine by observing the corresponding output sequence that the machine produces. A *transfer sequence from state S_i to state S_j* is an input sequence which transfers the machine from state S_i to state S_j .

A machine $M' = (S', I', 0', \delta', \lambda')$ is a *submachine* of the machine $M = (S, I, 0, \delta, \lambda)$ if and only if $S' \subseteq S, I' \subseteq I, 0' \subseteq 0, \delta' = \delta$ restricted to $S' \times I'$, and $\lambda' = \lambda$ restricted to $S' \times I'$.

An easily testable machine is one for which a short preset checking experiment can be found with a simple algorithm. In order to obtain a short preset checking experiment, it is desirable for the machine to have a short distinguishing sequence, a short synchronizing sequence,

Manuscript received February 13, 1974; revised January 17, 1975.
H. Fujiwara, T. Sasao, and K. Kinoshita are with the Department of Electronic Engineering, Osaka University, Osaka, Japan.
Y. Nagao is with the Technical Laboratory, Kawasaki Heavy Industries, Ltd., Hyogo, Japan.

and short transfer sequences. Therefore, we make the following definition. An *easily testable machine* is a reduced and strongly connected machine which possesses 1) a distinguishing sequence X_d of length $\lceil \log_2 n \rceil$ which forces the machine into a specific final state S_1 , i.e., X_d is also a synchronizing sequence, and 2) transfer sequences $T(i)$ with a length that is at most $\lceil \log_2 n \rceil$ to move the machine from state S_1 to state S_i for all i , where n is the number of states of the machine and where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x .

Example: Consider the p -stage binary shift register shown in Fig. 1. The p -stage binary shift register is a serial connection of p -unit delays interconnected so that at the occurrence of a shift signal the contents of the i th delay is *shifted* into the $(i + 1)$ st delay. Let Y_1, Y_2, \dots, Y_p be the state variables, let X be the input variable, and let Z be the output variable. For the p -stage binary shift register, a p -tuple state assignment $Y_1 Y_2 \dots Y_p$ can be found for each state such that

$$1) Y_i(t + 1) = Y_{i-1}(t), \quad i = 2, 3, \dots, p$$

$$2) Y_1(t + 1) = X(t)$$

and

$$3) Z(t) = Y_p(t)$$

where $Y_1(t), Y_2(t), \dots, Y_p(t)$, $X(t)$, and $Z(t)$ are the values of $Y_1, Y_2, \dots, Y_p, X, Z$ at time t , respectively. Then it is easily seen that any input sequence of length p will be both a distinguishing sequence and a synchronizing sequence, and that $Y_p Y_{p-1} \dots Y_1$ is a transfer sequence of length p which carries the p -stage binary shift register to state S_i with state assignment $Y_1 Y_2 \dots Y_p$. Therefore, the p -stage binary shift register is an easily testable machine.

III. AUGMENTATION OF THE GIVEN MACHINE

In this section we present a procedure to augment a given machine by adding two extra input symbols so that the augmented machine is an *easily testable machine*.

In the next section we will discuss the design of efficient checking experiments under the assumption that faults do not increase the number of states. For a machine realized by binary devices, the probability of the occurrence of faults that can increase the number of states is indeed rather small when the number of states n is an integral power of 2, since physical creation of new state variables would have been implied. However, when n is not an integral power of 2, and/or more than $\lceil \log_2 n \rceil$ state variables are used in the realization, such faults are very likely to occur. From this point of view, it is desirable to augment the original n -state machine so that the augmented machine has $2^p = n'$ states where $p = \lceil \log_2 n \rceil$.

Let $M = (S, I, 0, \delta, \lambda)$ be a given machine, where $S = \{S_1, S_2, \dots, S_n\}$, $I = \{I_1, I_2, \dots, I_m\}$, and $0 = \{0_1, 0_2, \dots, 0_i\}$. Then we can give a procedure for augmenting the given

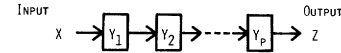


Fig. 1. The p -stage binary shift register.

machine M so that the augmented machine M^* is easily testable.

Augmentation Procedure

- 1) Add new states $S_{n+1}, S_{n+2}, \dots, S_{n'}$ to M if n is not an integral power of 2, where $n' = 2^p$ and $p = \lceil \log_2 n \rceil$.
- 2) Assign a p -bit binary code to all states such that each state has only one assignment.
- 3) Add new input symbols ϵ_0, ϵ_1 to M . The next state function δ and the output function λ for the new input symbols ϵ_0, ϵ_1 are defined as follows.

For each state S_i , with state assignment $Y_1 Y_2 \dots Y_p$, $\delta(S_i, \epsilon_0) = S_j$, and $\delta(S_i, \epsilon_1) = S_k$;

$$\lambda(S_i, \epsilon_0) = \lambda(S_i, \epsilon_1) = 0_1, \quad \text{if } Y_p = 0 \\ = 0_2, \quad \text{if } Y_p = 1$$

where S_j and S_k have state assignments $0Y_1 Y_2 \dots Y_{p-1}$ and $1Y_1 Y_2 \dots Y_{p-1}$, respectively.

The effect of this state transition is to shift the state assignment one digit to the right and introduce a zero or a one as new left most digit according to input ϵ_0 or ϵ_1 , respectively. Thus, this 2-column submachine restricted to inputs ϵ_0, ϵ_1 is isomorphic to the p -stage binary shift register. Since the p -stage binary shift register is an easily testable machine, this 2-column submachine is also easily testable, and hence the augmented machine M^* is too. Indeed, in the augmented machine M^* obtained above, any input sequence of length $p = \lceil \log_2 n \rceil$ consisting of ϵ_0 and ϵ_1 is both a distinguishing sequence and a synchronizing sequence, and $\epsilon_{Y_p} \epsilon_{Y_{p-1}} \dots \epsilon_{Y_2} \epsilon_{Y_1}$ is a transfer sequence of length p which transfers M^* from an arbitrary state to state S_i with state assignment $Y_1 Y_2 \dots Y_p$. The augmented machine M^* has n' states and $(m + 2)$ input symbols, where $\log_2 n' = \lceil \log_2 n \rceil$.

Example: Consider machine A given by Table I. Machine A is not strongly connected and does not have a distinguishing sequence. By applying the above procedure, we obtain the augmented machine A^* shown in Table II. A^* has a distinguishing sequence $\epsilon_0 \epsilon_0$ which is also a synchronizing sequence whose final state is S_1 . Transfer sequences are shown in Table III. Hence the augmented machine A^* is an easily testable machine.

IV. CHECKING EXPERIMENTS FOR THE EASILY TESTABLE MACHINES

In this section we consider checking experiments for the *easily testable machines*. The principle idea of our method is based mainly on those of Hennie [2] and Hsieh [5], and we assume that readers are familiar with the principle of those methods. Assume that the class of allowable failures satisfies the following conditions.

TABLE I
MACHINE A

input state		input	
		0	1
state	S_1	$S_2 (1)$	$S_1 (1)$
	S_2	—	$S_3 (0)$
	S_3	$S_2 (0)$	— (1)

The dash means "DON'T CARE."

TABLE II
AUGMENTED MACHINE A*

input state		input			
		0	1	ϵ_0	ϵ_1
state	00 S_1	$S_2 (1)$	$S_1 (1)$	$S_1 (0)$	$S_3 (0)$
	01 S_2	—	$S_3 (0)$	$S_1 (1)$	$S_3 (1)$
	10 S_3	$S_2 (0)$	— (1)	$S_2 (0)$	$S_4 (0)$
	11 S_4	—	—	$S_2 (1)$	$S_4 (1)$

TABLE III
TRANSFER SEQUENCES $T(i)$ FOR MACHINE A*

$T(1)$	$T(2)$	$T(3)$	$T(4)$
Λ	$\epsilon_1 \epsilon_0$	ϵ_1	$\epsilon_1 \epsilon_1$

" Λ " means the null sequence.

- 1) Any failure which occurs is assumed to occur throughout the test.
- 2) Failures do not increase the number of states.

Let $M = (S, I, 0, \delta, \lambda)$ be an n -state m -input easily testable machine. Let X_d be an input sequence of length $\lceil \log_2 n \rceil$ which is both a distinguishing sequence and a synchronizing sequence. Let S_1 be the final state resulting from the application of X_d . The transfer sequence with a length that is at most $\lceil \log_2 n \rceil$ to move M from state S_1 to state S_i is denoted by $T(i)$.

The checking experiment consists of five parts. The first part of the checking experiment is the *initializing part* which brings the machine under test to the starting state S_1 for the experiment. This can be done by a synchronizing sequence X_d . Hence, the first part of the experiment is preset and has the form:

$$\begin{array}{ll}
 \text{Input:} & X_d \\
 \text{State:} & - \quad S_1 \\
 \text{Output:} & -
 \end{array} \quad (1)$$

where the dash means "DON'T CARE."

The second part of the checking experiment carries the

correctly operating machine through all its states, displays all the different responses to X_d , and thus verifies that X_d is a distinguishing sequence. Thus, the second part of the checking experiment has the form:

$$\begin{array}{ll}
 \text{Input:} & X_d \\
 \text{State:} & S_i \quad S_1 \\
 \text{Output:} & Z_i
 \end{array} \quad (2)$$

for all states S_i of M , where $Z_i = \lambda(S_i, X_d)$.

The third part of the experiment verifies, by using a distinguished sequence X_d validated by the second part of the experiment, that X_d is a synchronizing sequence used to force the correctly operating machine into state S_1 . Thus, this part has the form:

$$\begin{array}{ll}
 \text{Input:} & X_d \quad X_d \\
 \text{State:} & S_i \quad S_1 \quad S_1 \\
 \text{Output:} & Z_i \quad Z_1
 \end{array} \quad (3)$$

for all states S_i of M .

The fourth part of the checking experiment verifies that $T(i)$ transfers the correctly operating machine from state

S_1 to state S_i . This can be done by using a distinguishing sequence X_d as follows:

Input:	X_d	$T(i)$	X_d	
State:	-	S_1	S_i	S_1 (4)
Output:	-	Z_{1i}	Z_i	

for all states S_i , where $Z_{1i} = \lambda(S_1, T(i))$.

The fifth part of the checking experiment is to be designed to check all the transitions and has the form

Input:	X_d	$T(i)$	I_j	X_d	
State:	-	S_1	S_i	$S_{ij} = \delta(S_i, I_j)$	S_1 (5)
Output:	-	Z_{1i}	$0_{ij} = \lambda(S_i, I_j)$	Z_{ij}	

for all states S_i and inputs I_j .

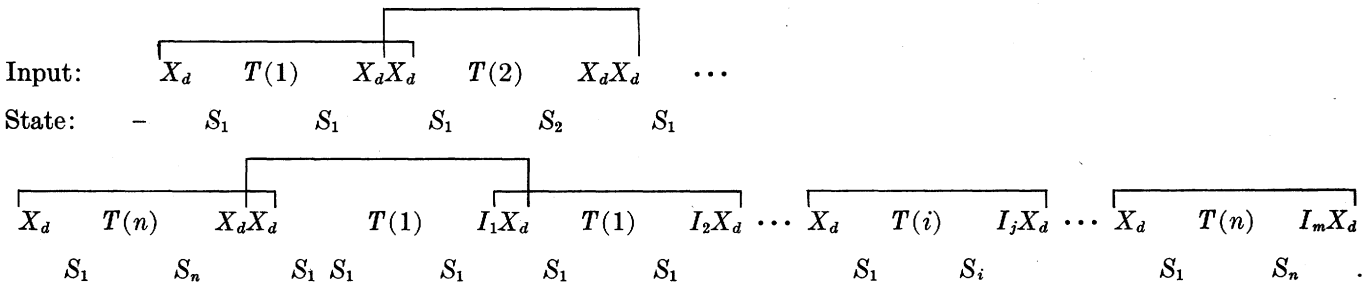
Since the distinguishing sequence X_d and the transfer sequence $T(i)$ have been validated by the previous parts of the checking experiment, S_i is uniquely determined by $T(i)$ and S_{ij} is recognized by X_d . Note that if both $\delta(S_i, I_j)$ and $\lambda(S_i, I_j)$ are unspecified, then such a transition from state S_i under input I_j need not be checked.

Although the checking experiment is functionally subdivided into five parts, these parts need not be physically separated from each other. Parts 1-4 can be completely contained in the following sequences:

Input:	X_d	$T(i)$	X_d	X_d
State:	-	S_1	S_i	S_1 (6)
Output:	-	Z_{1i}	Z_i	Z_1

for all states S_i .

Thus, the total checking experiment is to be organized from the subexperiments (5) and (6). Then we have the following checking experiment:



In this checking experiment, the initializing part is *preset*, hence the total checking experiment is preset, and thus is easy to be applied to the tested machine.

Let us derive the bound on the length of the checking experiment. Since the machine M is assumed to be an easily testable machine, $|X_d| = \lceil \log_2 n \rceil$ and $|T(i)| \leq$

$\lceil \log_2 n \rceil$ for $i = 1, 2, \dots, n$, where $|X|$ is the length of X .

From the organization of the checking experiment, it can be seen that the total length of the checking experiment is at most

$$|X_d| + \sum_{i=1}^n (|T(i)| + 2|X_d|) + \sum_{j=1}^m \sum_{i=1}^n (|T(i)| + |I_j| + |X_d|)$$

$$= (2n + 1)|X_d| + \sum_{i=1}^n |T(i)| + mn(|X_d| + 1) + m \sum_{i=1}^n |T(i)| \leq (2n + 1)\lceil \log_2 n \rceil + n\lceil \log_2 n \rceil + mn(\lceil \log_2 n \rceil + 1) + mn\lceil \log_2 n \rceil = (3n + 1)\lceil \log_2 n \rceil + mn(2\lceil \log_2 n \rceil + 1).$$

Thus, the upper bound on the length of the checking experiment is

$$(3n + 1)\lceil \log_2 n \rceil + mn(2\lceil \log_2 n \rceil + 1).$$

For large m and n , this bound is smaller than the bound

$$(n + \lceil \log_2 n \rceil)(1 + mn)$$

reported by Holborow [11], which is the best bound in the previous methods [7]-[11].

Example: Let us construct a checking experiment for machine A^* given by Table II. $X_d = \epsilon_0 \epsilon_0$ is both a distinguishing sequence and a synchronizing sequence whose final state is S_1 . Transfer sequences $T(i)$ from state S_1 to each state S_i are shown in Table III.

The total checking experiment is:

Input:	$\epsilon_0\epsilon_0$	$\overbrace{T(1)}^{(a)}$	$\epsilon_0\epsilon_0$	$\epsilon_0\epsilon_0$	$\overbrace{T(2)}^{(b)}$	$\epsilon_0\epsilon_0$	$\epsilon_0\epsilon_0$	$\overbrace{T(3)}^{(d)}$	$\epsilon_0\epsilon_0$	$\epsilon_0\epsilon_0$			
State:	-	S_1	S_1	S_1	S_1	S_2	S_1	S_1	S_3	S_1	S_1		
Output:	--	Λ	00	00	00	10	00	0	01	00			
		$\overbrace{T(4)}^{(f)}$	$\epsilon_0\epsilon_0$	$\epsilon_0\epsilon_0$	$T(1)$	$\epsilon_0\epsilon_0$	$T(1)$	1	$\epsilon_0\epsilon_0$	$\overbrace{T(1)}^{(a')}$	ϵ_0	$\epsilon_0\epsilon_0$	
		S_1	S_4	S_1	S_1	S_1S_2	S_1	S_1S_1	S_1	S_1S_1	S_1		
		00	11	00	Λ	1	10	Λ	1	00	Λ	0	00
		$\overbrace{T(1)}^{(b')}$	ϵ_1	$\epsilon_0\epsilon_0$	$T(2)$	1	$\epsilon_0\epsilon_0$	$\overbrace{T(2)}^{(c')}$	ϵ_0	$\epsilon_0\epsilon_0$	$T(2)$	ϵ_1	$\epsilon_0\epsilon_0$
		S_1	S_1S_3	S_1	S_2S_3	S_1	S_2S_1	S_1	S_2S_3	S_1	S_2S_3	S_1	
		Λ	0	01	00	0	01	00	1	00	00	1	01
		$T(3)$	0	$\epsilon_0\epsilon_0$	$T(3)$	1	$\epsilon_0\epsilon_0$	$\overbrace{T(3)}^{(d')}$	ϵ_0	$\epsilon_0\epsilon_0$	$\overbrace{T(3)}^{(e')}$	ϵ_1	$\epsilon_0\epsilon_0$
		S_1	S_3S_2	S_1	S_3-	S_1	S_3S_2	S_1	S_3S_4	S_1			
		0	0	10	0	1	--	0	0	10	0	0	11
		$\overbrace{T(4)}^{(f')}$	ϵ_0	$\epsilon_0\epsilon_0$	$T(4)$	ϵ_1	$\epsilon_0\epsilon_0$						
		S_1	S_4S_2	S_1	S_4S_4	S_1							
		00	1	10	00	1	11						

In the above experiment, subexperiments (a)–(f) are equivalent to (a')–(f'), respectively, and thus subexperiments (a')–(f') can be deleted. Then we can obtain the reduced checking experiment as follows:

procedure of designing checking experiments for such easily testable machines. For an n -state m -input symbol machine, this procedure gives a bound on the length of checking experiments that is approximately $mn[\log_2 n]$,

Input:	$\epsilon_0\epsilon_0$	$T(1)$	$\epsilon_0\epsilon_0\epsilon_0\epsilon_0$	$T(2)$	$\epsilon_0\epsilon_0\epsilon_0\epsilon_0$	$T(3)$	$\epsilon_0\epsilon_0\epsilon_0\epsilon_0$	$T(4)$	$\epsilon_0\epsilon_0\epsilon_0\epsilon_0$							
Output:	--	Λ	0000	00	1000	0	0100	00	1100							
		$T(1)$	$0\epsilon_0\epsilon_0$	$T(1)$	$1\epsilon_0\epsilon_0$	$T(2)$	$1\epsilon_0\epsilon_0$	$T(2)$	$\epsilon_1\epsilon_0\epsilon_0$	$T(3)$	$0\epsilon_0\epsilon_0$	$T(3)$	$1\epsilon_0\epsilon_0$	$T(4)$	$\epsilon_1\epsilon_0\epsilon_0$	
		Λ	110	Λ	100	00	001	00	101	0	010	0	1	--	00	111

V. CONCLUSION

In this paper, we have considered a method to construct easily testable machines and have considered checking experiments for such machines. An n -state easily testable machine considered in this paper is one which possesses 1) a distinguishing sequence of length $[\log_2 n]$ which forces the machine into a specific state S_1 and 2) transfer sequences of length at most $[\log_2 n]$ to carry the machine from state S_1 to state S_i for all i . We have shown that if an original machine is not easily testable, then it can be modified to an easily testable machine by adding two extra input symbols. We have also presented an efficient

which is smaller than the best bound mn^2 obtained in the previous methods [7]–[11]. Furthermore, the total checking experiment is preset.

ACKNOWLEDGMENT

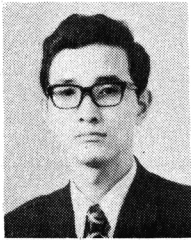
The authors wish to acknowledge the support and encouragement of Prof. H. Ozaki of Osaka University, Osaka, Japan. The authors also wish to acknowledge the useful comments of Dr. C. R. Kime of the University of Wisconsin.

REFERENCES

- [1] A. Gill, *Introduction to the Theory of Finite-State Machines*. New York: McGraw-Hill, 1962.
- [2] F. C. Hennie, "Fault detecting experiments for sequential

circuits," in *Proc. 5th Annu. Symp. Switching Circuit Theory and Logical Design*, Princeton, N. J., Nov. 1964, pp. 95-110.

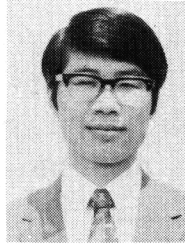
- [3] C. R. Kime, "A failure detection method for sequential circuits," *Dep. Elec. Eng., Univ. Iowa, Iowa City, Iowa, Tech. Rep. 66-13*, 1966.
- [4] G. Gönenc, "A method for the design of fault detection experiments," *IEEE Trans. Comput.* (Short Notes), vol. C-19, pp. 551-558, June 1970.
- [5] E. P. Hsieh, "Checking experiments for sequential machines," *IEEE Trans. Comput.*, vol. C-20, pp. 1152-1166, Oct. 1971.
- [6] D. E. Farmer, "Algorithms for designing fault-detection experiments for sequential machines," *IEEE Trans. Comput.*, vol. C-22, pp. 159-167, Feb. 1973.
- [7] Z. Kohavi and P. Lavallee, "Design of sequential machines with fault-detection capabilities," *IEEE Trans. Electron. Comput.*, vol. EC-16, pp. 473-484, Aug. 1967.
- [8] R. L. Martin, "The design of diagnosable sequential machines," in *Proc. Hawaii Int. Conf. Syst. Sci.*, 1968.
- [9] S.-I. Murakami, K. Kinoshita, and H. Ozaki, "Sequential machines capable of fault diagnosis," *IEEE Trans. Comput.*, vol. C-19, pp. 1079-1085, Nov. 1970.
- [10] J. R. Kane and S. S. Yau, "On the design of easily testable sequential machines," in *Conf. Rec. 12th Annu. Symp. Switching and Automata Theory*, pp. 38-42, Oct. 1971.
- [11] C. E. Holborow, "An improved bound on the length of checking experiments for sequential machines with counter cycles," *IEEE Trans. Comput.* (Short Notes), vol. C-21, pp. 597-598, June 1972.
- [12] M. J. Y. Williams and J. B. Angell, "Enhancing testability of large-scale integrated circuits via test points and additional logic," *IEEE Trans. Comput.*, vol. C-22, pp. 46-60, Jan. 1973.
- [13] H. Fujiwara and K. Kinoshita, "Design of diagnosable sequential machines utilizing extra outputs," *IEEE Trans. Comput.*, vol. C-23, pp. 138-145, Feb. 1974.



Hideo Fujiwara (S'70-M'74) was born in Nara, Japan, on February 9, 1946. He received the B.E., M.E., and Ph.D. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1969, 1971, and 1974, respectively.

He is currently a Research Assistant in the Department of Electronic Engineering, Osaka University. His research interests are switching theory and automata theory, and he specializes in the development of fault diagnosis of logical circuits.

Dr. Fujiwara is a member of the Institute of Electronics and Communication Engineers of Japan.

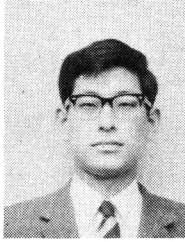


of Japan.

Yoich Nagao was born in Hyogo, Japan, on July 17, 1949. He received the B.E. and M.E. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1972 and 1974, respectively.

Since 1974 he has been with the Technical Laboratory, Kawasaki Heavy Industries, Ltd., Hyogo, Japan. His research interests are switching theory and automata theory.

Mr. Nagao is a member of the Institute of Electronics and Communication Engineers



Japan.

Tsutomu Sasao (S'72) was born in Osaka, Japan, on January 26, 1950. He received the B.E. and M.E. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1972 and 1974, respectively. He is currently a Ph.D. student in the Department of Electronic Engineering, Osaka University.

His research interests are switching theory and automata theory.

Mr. Sasao is a member of the Institute of Electronics and Communication Engineers of



Kozo Kinoshita (S'58-M'64) was born in Osaka, Japan, on June 21, 1936. He received the B.E., M.E., and Ph.D. degrees in communication engineering from Osaka University, Osaka, Japan, in 1959, 1961, and 1964, respectively.

Since 1964 he has been with Osaka University, where he is now an Associate Professor of Electronic Engineering. His fields of interest are switching theory, system and logical design, and fault diagnosis of information processing systems.

Dr. Kinoshita is a member of the Institute of Electronics and Communication Engineers of Japan and the Information Processing Society of Japan.