한국정보전자통신기술학회논문지(jkiiect)'15-12, Vol.8 No.6
논문 15-08-06-470 http://dx.doi.org/10.17661/jkiiect.2015.8.6.470

A Heuristic Technique for Generating the Synchronizable and Optimized Conformance Test Sequences

Chul Kim*

최적화된 동기적 적합성시험 항목의 발견적 생성 방법

김철*

Abstract This paper presents a new technique for generating an optimum synchronizable test sequence that can be applied in the distributed test architecture where both external synchronization and input/output operation costs are taken into consideration. The method defines a set of phases that constructs a tester-related digraph from a given finite state machine representation of a protocol specification such that a minimum cost tour of the digraph with intrinsically synchronizable transfer sequences can be used to generate an optimum synchronizable test sequence using synchronizable state identification sequences as the state recognition sequence for each state of the given finite state machine. This hybrid approach with a heuristic and optimization technique provides a simple and elegant solution to the synchronization problem that arises during the application of a predetermined test sequence in some protocol test architectures that utilize remote testers.

요 약 본 논문은 통신 프로토콜의 적합성 시험시 하위 시험기와 상위 시험기간의 직접적인 상호 작용이 불가능한 분산 시험 환경에서 외부 동기화와 입출력 운용비용을 고려해야 할 경우 적용될 수 있는 최적화된 동기적 시험 항목 생성을 위한 새로운 방법을 제시한다. 이 방법은 세 가지 단계의 절차로 구성되어 있는데, 첫 번째 단계는 시험 구현물의 각 상태로부터 나오는 모든 입출력 짝들의 유일성을 기술하는 상태인식 표를 생성한다. 둘 째 단계에서는 이 상태인식 표로부터 각 상태의 동기화된 상태인식 열을 구성한다. 마지막 단계에서는 시험 구현물의 시험에지들과 각 상태의 동기화된 상태인식 열들을 이용하여 최적화된 동기적 시험열들을 생성한다. 본 논문의 방법은 원거리 시험기들을 사용하는 프로토콜 시험 구조에서 기 결정된 시험 항목을 적용할 때 시험 대상과 이들 시험기들간에 발생할 수 있는 동기화의 유용한 해결 기법으로 사용될 수 있다.

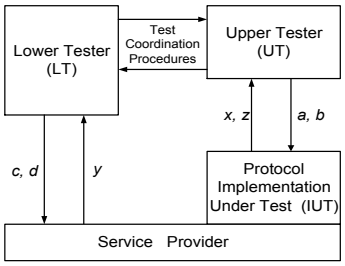
Key Words : Intrinsically synchronizable transfer sequence, optimum synchronizable test sequence, conformance testing, synchronization problem, synchronizable state identification sequence

1. Introduction

When a conformance testing is performed, a protocol implementation under test(IUT) is viewed as a black box. Most of test sequence generation methods[1] are based on the finite state machine(FSM) model[2, 3]. Synchronization between the upper tester(UT) and the lower tester(LT) can be achieved by constructing a synchronizable test sequence such that the corresponding sequence of transitions causes no synchronization problem[4, 5, 6]. This paper presents basically a new method for generating an

*Corresponding Author : Department of Computer Science, Yongin University (chulkim@yongin.ac.kr)
Received November 23, 2015 Revised December 02, 2015 Accepted December 10, 2015

optimally synchronizable test sequence. Section 2 describes an FSM model testing, and introduces the synchronization problem and the related concepts of synchronization between the testers. Section 3 proposes a new technique for generating optimum conformance test sequence that do not encounter a synchronization problem. This method consists of a set of phases that constructs a tester-induced digraph related to the LT and the UT, derives intrinsically synchronizable transfer sequences(ISTSs) and synchronizable state identification sequences(SSISs) to bridge between the testers, and generates an optimally synchronizable test sequence for protocol testing. Finally, conclusions are given in Section 4.



a, b: Inputs from UT to IUT *c, d*: Input from LT to IUT
x, z: Outputs from IUT to UT *y*: Output from IUT to LT

Fig. 1. The distributed test architecture for testing a protocol implementation

2. FSM Model Testing and Definitions Related to Synchronization Problem

A protocol entity can be specified by a deterministic FSM M with a quintuple (S, I, O, f, g), where: S = the set of states of M, including a special state s1 called the initial state; I = the set of inputs, written as ip in the following, ip ∈ I ; O = the set of outputs, written as oq in the following, including the null output(nu), oq ∈ O ; f = the next-state(transition) function, S x I → S ; g = the output function, S x I → O.

An FSM M is represented as a directed graph, G = (V , E), where the set of vertices V = {v1 , . . . , v n} represents the set of specified states S = {s1 , . . . , sn} of M and a directed edge (Tm ; vj , vk ; ip / oq) ∈ E represents a transition from state sj to state sk in M.

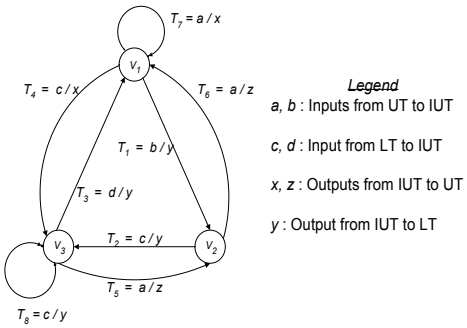


Fig. 2. A graph representation of a finite state machine M

Table 1. Transition table for M in Fig. 2

	Output				Next-State			
Input / State	*a*	*b*	*c*	*d*	*a*	*b*	*c*	*d*
V_1	*x*	*y*	*x*	*nu*	V_1	V_2	V_3	V_1
V_2	*z*	*nu*	*y*	*nu*	V_1	V_2	V_3	V_2
V_3	*z*	*nu*	*y*	*y*	V_2	V_3	V_3	V_1

(Legend) The symbol “nu” means null output.

Definition 1: Synchronization Problem

A pair of consecutive transitions whose labels form a sequence of input/output operations [i1 / o1 , i2 / o2] encounters a LT to UT synchronization problem if both i1 and o1 are related to LT(and not to UT), where i2 is related to UT; it encounters a UT to LT synchronization problem if both i1 and o1 are related to UT(and not to LT), where i2 is related to LT.

Definition 2: External Synchronization

Operations

LT to UT : LT informs UT that it is now a right time to send the next message.

UT to LT : UT informs LT that it is now a right time to send the next message.

As shown in Fig. 3, the LT to UT(or UT to LT) external synchronization operation is introduced each time the LT to UT(or UT to LT) synchronization problem is encountered.

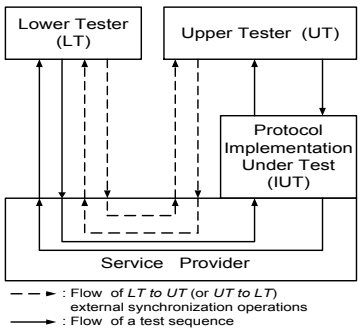


Fig. 3. The flows of external synchronization operations and a test sequence

Definition 3: Intrinsically Synchronizable Transfer Sequence

In the intrinsically synchronizable test sequence [i1 / o1 , X1 , i2 / o2 , X2 ,...,ik / ok , X1k , ik+1 / ok+1 ,..., Xn-1 , in / on] of the above Definition 4, [ik+1 / ok+1] relating to the sequence [ik / ok] is an intrinsically synchronizable transfer sequence(ISTS) if, for 1 ≤ k ≤ n-1, either

1) ok and ik+1 are related to the LT, and ok+1 is related to the UT, or

2) ok and ik+1 are related to the UT, and ok+1 is related to the LT.

3. The Proposed Method for Generating an Optimum Synchronizable Test Sequence

We introduce a new technique for generating optimum conformance test sequences that do not encounter the synchronization problem using intrinsically synchronizable transfer sequences(ISTSs) and synchronizable state identification sequences(SSISs) to bridge and synchronize the pairs of input/output interactions between the LT(or UT) and the UT(or LT). First, we construct a tester-induced digraph which is a directed graph of duplicated vertices generated according to the characteristics of input/output interactions between the testers, and then generate intrinsically synchronizable transfer sequences(ISTSs) from the tester-induced digraph. Second, we present a method for constructing synchronizable state identification sequences(SSISs) which can be used to generate an optimally synchronizable test sequence. Finally, we generate and optimize a test sequence which is synchronizable between the test subsequences.

3.1 Tester-induced Digraph Construction

Algorithm 1: *The Construction of Tester-induced Digraph*

1) For each vertex $v_j \in V$, create a pair of vertices LT_j, UT_j.

2) For each edge $(T_m ; v_j, v_k ; i_p/o_q) \in E$, create fine edges as follows: $(UT_j , UT_k ; i_p/o_q)$, if both i_p and o_q are related to UT, or $(UT_j , LT_k ; i_p/o_q)$, if i_p is related to UT and o_q is related to LT, or $(LT_j , LT_k ; i_p/o_q)$, if both i_p and o_q are related to LT, or $(LT_j , UT_k ; i_p/o_q)$, if i_p is related to LT and o_q is related to UT.

3) For each pair of the consecutive transitions $[i_k/o_k, i_{k+1}/o_{k+1}]$, find intrinsically synchronizable transfer sequences(ISTSs) and substitute every found edge $(T_m ; v_j, v_k ; i_p/o_q) \in E$ into bold edge as follows: $(LT_j , UT_k ; i_p/o_q)$, if i_p is related to LT and o_q is related to UT, or $(UT_j , LT_k ; i_p/o_q)$, if i_p is related to UT and o_q is related to LT.

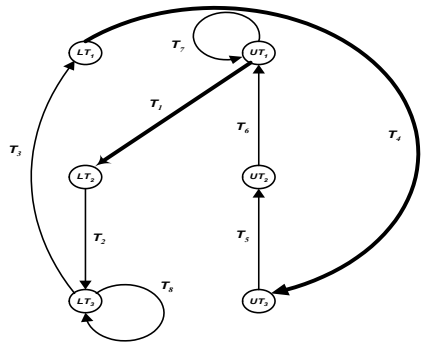


Fig. 4. The tester-induced digraph GT(VT, ET) converted from the graph G(V, E) of M in Fig. 2

3.2 Construction of Synchronizable State Identification Sequences

Algorithm 2: *The Derivation of State Identification Table (SIT)*:

1) Initialize the SIT with a chain of start states related to the LT_k (or UT_k) of the tester-induced digraph, $G^T(V^T, E^T)$, of the FSM M and a chain of input/output pairs outgoing from the corresponding start states.

2) Compare every outgoing edges for each start state until the uniqueness of each input/output pair can be found. If found, this pair for the state is marked as *True* in the SIT; otherwise it is marked as *False*. For each state, the resulting status may be marked as *Found* even if only one of the corresponding outgoing edges for the state is identified. Each iteration for finding the uniqueness can be completed whenever every outgoing edges for the corresponding start states are tested.

3) If there exist an unidentified states of input/output pairs that were marked as *Not Found* in the previous iteration, insert each final state of the transition into the last of the chain of start states in the SIT and add separately new outgoing edges for the state marked as *Not Found* into the last of the chain of input/output pairs and repeat Step 2) for the last state of the chain; otherwise this algorithm ends.

In the next procedure, a set of LT- and UT-synchronizable state identification sequences for each state of M, denoted as LT- (UT-) SSIS(vk), is constructed directly from the SIT of Table 2, as shown in Table 3. In addition, each SSIS of the table can be minimally optimized in length according to the above steps of Algorithm 2.

Table 2 The state identification table(SIT) derived from the tester-induced digraph GT(VT, ET) in Fig. 4

Iteration	Chain of Start States	Chain of Input/Output Pairs	Final State of Transition	Uniqueness Mark of Input/Output Pairs	Status
1	LT_1	c/x	UT_3	*True*	*Found*
	UT_1	a/x	UT_1	*True*	*Found*
		b/y	LT_2	*True*	
	LT_2	c/y	LT_3	*True*	*Found*
	UT_2	a/z	UT_1	*True*	*Found*
	LT_3	c/y	LT_3	*False*	*Found*
		d/y	LT_1	*True*	
	UT_3	a/z	UT_2	*False*	*Not Found*
2	UT_3, UT_2	a/z, a/z	UT_1	*True*	*Found*

Table 3. A set of LT- or UT-synchronizable state identification sequences, LT- or UT-SSIS(vk), of a FSM M constructed from the SIT in Table 2

Start State	LT- or UT-Synchronizable State Identification Sequence, *LT-* or *UT-SSIS*(v_k)	Final State
v_1	*LT-SSIS*(v_1) = [c/x] = [T_4]	v_3
	UT-SSIS(v_1) = [b/y] = [T_1]	v_2
v_2	*LT-SSIS*(v_2) = [c/y] = [T_2]	v_3
	UT-SSIS(v_2) = [a/z] = [T_8]	v_1
v_3	*LT-SSIS*(v_3) = [d/y] = [T_3]	v_1
	UT-SSIS(v_3) = [a/z, a/z] = [T_5, T_6]	v_1

3.3 Optimum Synchronizable Test Sequence Generation

Finally, we present that an optimum synchronizable test sequence can be generated using a tester-induced digraph and a set of minimum-length SSIS sequences.

Definition 4: Optimum Synchronizable Test Sequence

A test sequence of an FSM M is called an optimum synchronizable test sequence if there exists no synchronization problem between the transitions of test subsequences of M and if the total tour of the test sequence which tests each transition of M is possibly computed and

minimally costed.

For example, the graph G′ shown in Fig. 5 is derived from the graph G of Fig. 2 and the graph GT of Fig. 4, and the test subsequences of Table 4.

Table 4. A set of synchronizable test subsequence TSS(Tm) for each testing edge of FSM M in Fig. 2

Testing Edge T_m	Start State of T_m Related to LT_j (or UT_j)	Final State of T_m Related to LT_j (or UT_j)	Synchronizable Test Subsequence $TSS(T_m)$	Tail(LT-(UT-) $SS(S(v_k))$
T_1	UT_1	LT_2	$TSS(T_1) = [b/y\ @\ c/y] = [T_1,\ T_2]$	LT_3
T_2	LT_2	LT_3	$TSS(T_2) = [c/y\ @\ d/y] = [T_2,\ T_3]$	LT_1
T_3	LT_3	LT_1	$TSS(T_3) = [d/y\ @\ c/x] = [T_3,\ T_4]$	UT_3
T_4	LT_1	UT_3	$TSS(T_4) = [c/x\ @\ a/z,\ a/z] = [T_4,\ T_5,\ T_6]$	UT_1
T_5	UT_3	UT_2	$TSS(T_5) = [a/z\ @\ a/z] = [T_5,\ T_6]$	UT_1
T_6	UT_2	UT_1	$TSS(T_6) = [a/z\ @\ b/y] = [T_6,\ T_7]$	LT_2
T_7	UT_1	UT_1	$TSS(T_7) = [a/x\ @\ b/y] = [T_7,\ T_1]$	LT_2
T_8	LT_3	LT_3	$TSS(T_8) = [c/y\ @\ d/y] = [T_8,\ T_3]$	LT_1

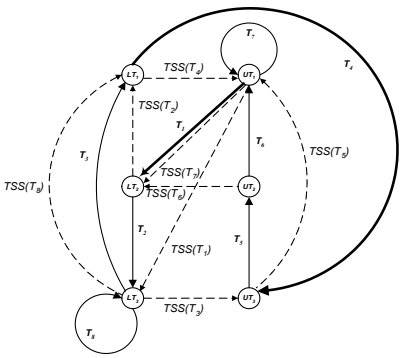


Fig. 5. The graph G' for the graph G of M shown in Fig. 2

Algorithm 3: *The mRCPT with Synchronization Property:*

1) Construct a directed graph $G^* = (V^*, E^*)$, the so-called a *tester-induced symmetric augmentation* of G', where $V^* \equiv V' \equiv LT_j \cup UT_j$, each edge of E in $E' \equiv E \cup E_{tss}$ is included in E^* zero or more times, and each edge of synchronizable $TSS(T_m)$ in E_{tss} is included in E^* at least once, such that the total cost of edges in E^* is minimum and the in-degree of each tester-induced vertex LT_j (or UT_j) $\in V^*$ is equal to its out-degree.

2) Find an Euler tour of the resulting tester-induced symmetric graph G^*, that is, a tour of G^* which traverses each edge of synchronizable $TSS(T_m)$ in E_{tss} exactly once.

The problem of determining a symmetric augmentation of a graph G′ can be reduced to a minimum-cost maximum flow[7] on a graph GF = (VF, EF) constructed from G′. The graph GF for the graph G′ of Fig. 5 is shown in Fig. 6 and depicts the minimum-cost maximum flow for the graph G′ that a total of 9 edge replications in E. Thus, the tester-induced symmetric augmentation graph G* for the graph G′ is constructed from the graph GF and is shown in Fig. 7 with the minimal replications of edge (vj , vk ; ip / oq) ∈ E′.

Finally, we can generate an optimally synchronizable test sequence. For an example as shown in Fig. 7, the minimum-cost tour over the dotted edges is as follows: [TSS(T1), TSS(T3), TSS(T5), TSS(T7), T2 , TSS(T8), T4 , T5 , TSS(T6), TSS(T2), TSS(T4)]. As given in Table 5, this tour that begins at UT1 and return to UT1 is used to generate the test sequence [b/y, c/y, d/y, c/x, a/z, a/z, a/x, b/y, c/y, c/y, d/y, c/x, a/z, a/z, b/y, c/y, d/y, c/x, a/z, a/z] with the total cost of 20 input/output operations, where no synchronization problem occurs between every pair of transitions.

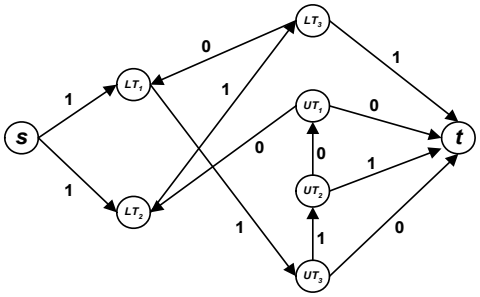


Fig. 6. The graph GF of the graph G' shown in Fig. 5 and a minimum-cost maximum flow

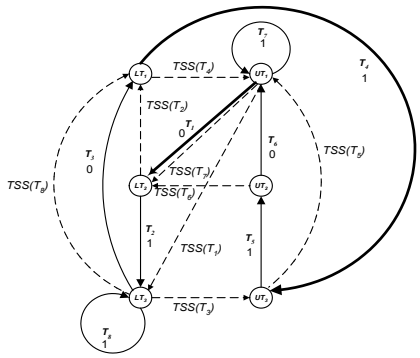


Fig. 7. The symmetric augmentation G* of the graph G' in Fig. 5

Table 5. An optimally synchronizable test sequence for an FSM M in Fig. 2

The sequence of synchronizable test subsequences with T_m: [$TSS(T_1)$, $TSS(T_3)$, $TSS(T_3)$, $TSS(T_7)$, T_2 , $TSS(T_8)$, T_4 , T_5 , $TSS(T_6)$, $TSS(T_2)$, $TSS(T_4)$]
The test sequence of input/output pairs: [*b/y, c/y, d/y, c/x, a/z, a/z, a/x, b/y, c/y, c/y, d/y, c/x, a/z, a/z, b/y, c/y, d/y, c/x, a/z, a/z*]
Total Cost = 20 input/output pairs

3.4 Complexity of Our Method and Performance Evaluation

Assume that n and |E| are the number of vertices and edges in G. The overall complexity of our method (in this paper, we do not show it in detail because of the limited space) is

$$\min(O(2(n-1) + \frac{|E|(5|E|-2)}{2}), O(8n2|E|)).$$

Table 6 shows a comparative test sequence length of major test generation methods[1] and Our Method. Length of the test sequence, in terms of number of input/output pairs, will determine the execution time for the test.

Table 6. Performance evaluation of Our Method

Test Generation Methods	Test Sequence Length of FSM M in [1]
DS-method	67 input/output pairs
W-method	109 input/output pairs
UIO-method	48 input/output pairs
Our Method	42 input/output pairs

3.5 The Experimental Result of an Application to the B-ISDN Q.2931

A finite state machine of the Q.2931 connection control procedures can be simplified and modeled into a transition diagram of each message and the corresponding state as shown in Fig. 8. As shown in Table 7, the total cost of an optimum synchronizable test sequence is 62 input/output operations, and the tour begins at v1 and returns to v1.

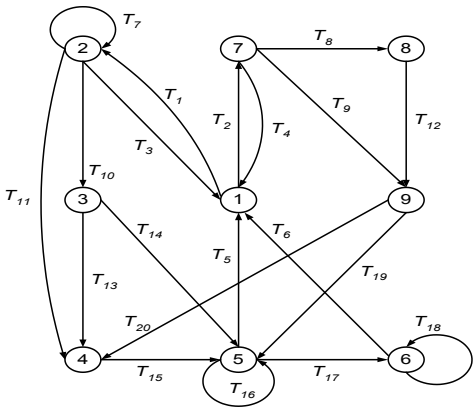


Fig. 8. The simplified FSM of B-ISDN Q.2931

Table 7. An optimum synchronizable test sequence for the simplified FSM of Q.2931

The sequence of synchronizable test subsequences with T_m: [$TSS(T_2)$, $TSS(T_{11})$, T_4, $TSS(T_3)$, $TSS(T_{10})$, $TSS(T_5)$, $TSS(T_{11})$, $TSS(T_{16})$, T_2 , $TSS(T_4)$, T_{11} , $TSS(T_{13})$, T_7 , $TSS(T_7)$, T_{15} , $TSS(T_{17})$, $TSS(T_{18})$, $TSS(T_9)$, T_{10} , $TSS(T_{13})$, T_5 , T_1 , T_{10} , $TSS(T_{14})$, T_2 , $TSS(T_9)$, $TSS(T_{20})$, T_5 , T_2 , T_9 , T_{12} , $TSS(T_{12})$, T_{15} , T_5 , T_2 , $TSS(T_9)$, T_{15} , T_5 , T_2 , T_9 , $TSS(T_{19})$]
The test sequence of input/output pairs: [*SETUP/setup.ind, release.resp/RELEASE COMPLETE, setup.req/SETUP, 2nd.T303/release.ind, setup.req/SETUP, 2nd.T303/release.ind, setup.req/SETUP, CALL PROCEEDING/proceeding.ind, T310/RELEASE & release.ind, RELEASE COMPLETE (or RELEASE)/release.conf, setup.req/SETUP, CONNECT/setup.conf & CONNECT ACKNOWLEDGE, release.req/RELEASE, 1st.T308/RELEASE, RELEASE COMPLETE (or RELEASE)/release.conf, SETUP/setup.ind, release.resp/RELEASE COMPLETE, setup.req/SETUP, CONNECT/setup.conf & CONNECT ACKNOWLEDGE, release.req/RELEASE, RELEASE COMPLETE (or RELEASE)/release.conf, setup.req/SETUP, 1st.T303/SETUP, 1st.T303/SETUP, CONNECT/setup.conf & CONNECT ACKNOWLEDGE, release.req/RELEASE, 2nd.T308/RESTART, 1st.T316 & 2nd.T316/RESTART, 1st.T316 & 2nd.T316/RESTART, 1st.T316 & 2nd.T316/RESTART, RESTART ACKNOWLEDGE/nu, setup.req/SETUP, CALL PROCEEDING/proceeding.ind, CONNECT/setup.conf & CONNECT ACKNOWLEDGE, release.req/RELEASE, RELEASE COMPLETE (or RELEASE)/release.conf, setup.req/SETUP, CALL PROCEEDING/proceeding.ind, T310/RELEASE & release.ind, RELEASE COMPLETE (or RELEASE)/release.conf, SETUP/setup.ind, proceeding.req/CALL PROCEEDING, setup.resp/CONNECT, CONNECT ACKNOWLEDGE/setup.complete.ind, release.req/RELEASE, RELEASE COMPLETE (or RELEASE)/release.conf, SETUP/setup.ind, setup.resp/CONNECT, setup.resp/CONNECT, setup.resp/CONNECT, CONNECT ACKNOWLEDGE/setup.complete.ind, release.req/RELEASE, RELEASE COMPLETE (or RELEASE)/release.conf, SETUP/setup.ind, setup.resp/CONNECT, CONNECT ACKNOWLEDGE/setup.complete.ind, release.req/RELEASE, RELEASE COMPLETE (or RELEASE)/release.conf, SETUP/setup.ind, setup.resp/CONNECT, T313/RELEASE & release.ind, RELEASE COMPLETE (or RELEASE)/release.conf*]
Total Cost: *62 input/output pairs of transitions*

4. Conclusions

In this paper, we proposed a new technique for generating an optimally synchronizable test sequence such that do not encounter a synchronization problem and provide an optimum length. This method which is a hybrid of a heuristic and optimization technique consists of a set of phases that constructs a tester-induced digraph related to the testers, derives intrinsically synchronizable transfer sequences(ISTSs) and synchronizable state identification sequences(SSISs) to rendezvous between the testers, and finally generates an optimally synchronizable test sequence. In summary, this test sequence can be applied in a testing system where the cost of both external synchronization operations and input/output operations are taken into consideration.

REFERENCES

[1] Chul Kim and J. S. Song, "Test Sequence Generation Methods for Protocol Conformance Testing", Proc. IEEE COMPSAC 94, pp. 169-174, 1994.

[2] A. T. Dahbura, K. K. Sabnani, and M. U. Uyar, "Formal Methods for Generating Protocol Conformance Test Sequences", Proceedings of the IEEE, vol. 78, no. 8, pp. 1317-1326, 1990.

[3] D. P. Sidhu and T. K. Leung, "Formal Methods for Protocol Testing: A Detailed Study", IEEE Transactions on Software Engineering, vol. 15, no. 4, pp. 413-426, 1989.

[4] S. Boyd and H. Ural, "The Synchronization Problem in Protocol Testing and Its Complexity", Information Processing Letters, vol. 40, pp. 131-136, 1991.

[5] Chul Kim and J. S. Song, "Generating Synchronizable Conformance Test Sequence Based on Distinguishing Sequence", IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E80-A, no. 6, pp. 1080-1082, 1997.

[6] B. Sarikaya and G. v. Bochmann, "Synchronization and Specification Issues in Protocol Testing", IEEE Transactions on Communications, vol. 32, no. 4, pp. 389-395, 1984.

[7] J. Edmonds and E. L. Johnson, "Mathing,

Euler Tours, and the Chinese Postman", Mathematical programming, vol. 5, pp. 88-124, 1973.

Author Biography

Chul Kim **[Regular member]**

- Feb. 1977 : Yonsei Univ., Electronics Engineering, B.S.
- Aug. 2000 : Yonsei Univ., Computer Science, Ph.D.
- Jul. 1981 ~ Jan. 1984 : Samsung LtD, Researcher
- Feb. 1984 ~ Jan. 1993 : IBM Korea, Engineer
- Mar. 1994 ~ current : Yongin Univ., Dept. of Computer Science, Professor

<Research Interests> Protocol Engineering, Computer Security