5치 논리 시뮬레이션에서 효율적인 헤저드 분석을 위한 TRF 지연 모델

강 민 섭†

요 약

본 논문은 5치 논리 시뮬레이션 환경에서 효율적인 헤저드 분석을 위한 새로운 TRF (Transition Rise/Fall) 지연모델을 제안한다. 주어진 회로에 있어서 헤저드 분석을 수행하기 위하여 우선 응답시간 지연과 천이시간 지연을 허용하는 5치 논리 시뮬레이터에 의해서 타이밍 분석이 수행되고, 이미 수행된 타이밍 관계를 조사하므로 써 헤저드를 검출할 수 있다.

몇 개의 예제회로에 대해서 시뮬레이션을 수행한 결과를 통하여 제안한 방법의 실용성을 입증하였다.

A New TRF Delay Model for the Efficient Hazard Analysis in a 5-valued Logic Simulation

Min-sup Kang†

ABSTRACT

This paper proposes a new TRF (Transition Rise/Fall) delay model for the efficient hazard analysis in a 5-valued logic simulation environment.

For the hazard for a given logic circuit, the timing analysis is first performed by means of a 5-valued logic simulator which uses the TRF delay model which incorporates the response delay for a response state with the transition delay for a transition state of an element, and then hazards are detected through investigating timing relations.

Simulation examples and experimental results are also given to demonstrate the practicability of the proposed methods.

1. 서 론

With more and more increasing complexities and performances of Very Large Scale Integration (VLSI) circuits, the use of Computer-Aided Design (CAD) tools have become indispensable for both design and verification of VLSI's[1]. Among these tools, a logic simulator is used for verifying design errors in a logic design process, and its accuracy can be determined by means of the number of signal values and the type of delays used for primitives[2].

In logic simulation, the choice of particular delay parameters is closely related to the number of signal values used in the simulation. In general, either a 2-valued or a 3-valued logic set is used in zero, unit, or rise/fall (R/F) delay model and a 5-valued one can be used in a Min/Max (ambiguity) delay model or a

† 종신회원:안양대학교 컴퓨터학과
논문접수:1996년 1월 27일, 심사완료:1996년 7월 11일

precise delay model.

A hazard, such as a transient error or spike, may occur in a combinational circuit due to inherent delays associated with logic gates, and a race may occur in an asynchronous sequential circuit due to concurrent changes of several signals. These abnormal functions have been regarded as a very serious problem in logic design[2].

Various methods for detecting hazards which may exist in the circuit have been reported by a number of researchers[2-6, 8].

A 2-valued or a 3-valued simulation system[2-4] with the R/F delay is not sufficient to detect hazards (races) by means of an accurate timing analysis because the system can not use transition states such as upward and downward transitions.

A method for handling a problem of multiple-input transitions has been introduced by Eichelberger[3]. However, not only it is restricted to a 3-valued logic set, but also it can not predict existence of dynamic hazards in combinational circuits. To solve these problems, multi-valued logic systems[4, 5] have been proposed. A 9-valued simulation system which is adopted for detecting static and dynamic hazards require a considerable computation cost due to increasing the number of events to be analyzed.

A Min/Max (ambiguity) delay model is proposed for obtaining accurate timing information, which can often lead to overly pessimistic results in handling large circuits with reconvergent fanout[2]. Time symbolic simulation can not be applied to a large circuit [6]. Thus effective logic simulator must take care of not only predicting logic behaviors of combinational and sequential circuits, but also detecting timing errors such as potential hazards or race conditions.

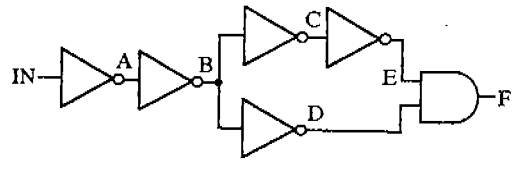
This paper proposes a new TRF (Transition Rise/Fall) delay model for the efficient hazard analysis in a 5-valued logic simulation environment. For the hazard for a given logic circuit, the timing analysis is first performed by means of a 5-valued logic simulator which uses the TRF delay model which incorporates the response delay for a response state with the transition delay for a transition state of an element, and then hazards are detected through investigating timing relations. Simulation examples and experimental results are also given to demonstrate the practicability of the proposed methods.

The organization of this paper is as follows. Section 2 presents a problem of the conventional Min/Max delay model. Section 3 describes a new TRF delay model used in the simulator. In section 4, an efficient hazard detection algorithm is given, which is based on a new TRF delay model. Section 5 gives a number of experimental results, and conclusion is given in Section 6.

2. Problem of Min/Max Delay Model

A Min/Max delay model is a refinement of the R/F delay model, and takes into account the delay time of a logic gate which is specified as an interval ⟨m, M⟩, where m(M) denotes minimum (maximum) delay time. Here m-M (i.e. the duration of the signal transition) is known as an ambiguity region. This model can be used for the worst case analysis of a circuit performance, but it is well known that simulation results are overly pessimistic even for correct design when considering reconvergent circuits[2].

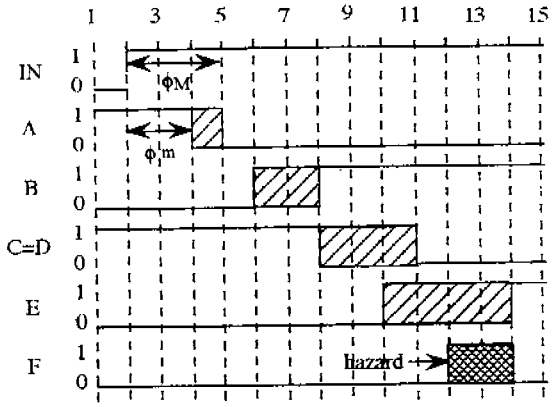
As an example, consider a simple circuit with a reconvergent fanout[2] shown in (Fig. 1), and assume that a delay time of each gate is in ⟨2, 3⟩.



(Fig. 1) Reconvergent circuit

If an input is applied to this circuit with the Min/Max delay, which changes from 0 to 1 at time 2, then

we can get the simulation result as shown in (Fig. 2). As can be seen from this result, the use of the Min/Max delay produces a potential hazard that does not actually occur between times 12 and 14 because of the overlap of two ambiguity regions (between times 10 and 11) at nodes C=D and E. Thus the Min/Max delay model is ineffective in detecting timing errors of large circuits with reconvergent fanouts.



(Fig. 2) Min/Max delay simulation

3. A New Transition Rise/Fall Delay Model

An actual logic gate has a specific logic function associated with an inherent propagation delay. There are two ways to set up models of the propagation delay[7]. A front-end delay model initially computes the delays assigned to all inputs before a gate is evaluated. A back-end delay model which is proceeded in the reverse order assumes that the assigned delay is associated with outputs of logic gates. Although both models may be applied to digital logic simulation, we adopt the latter for the sake of execution efficiency.

In a digital simulation system, actual signals propagated through a gate is normally represented by a set of k discrete values, often referred to as a k-valued logic set.

In our simulation system, a signal model uses a 5-valued logic set V5={0, 1, X, U, D}, where 0, 1, and X indicate logical zero, logical one, and unknown value, respectively. Also U and D are used to represent states of signal rising (upward transition) and signal falling (downward transition), respectively.

Let Rs and Ts be sets of response states and transition states of a signal, respectively. Then, in a 5-valued simulation, Rs and Ts are expressed as follows;

$$Rs=\{0, 1\}$$

and

$$Ts=\{U, D, X\}.$$

Note that, in a 3-valued simulation which does not use U and D, a transition state is represented by value X.

First, consider an n-input gate having a R/F delay, and assume that an input signal Si changes from logic value a to logic value b at time t. Let signal Si be represented by

$$Si=(a, t, b), \quad (1)$$

where a, b∈Rs. In this case, let output signal So be represented by

$$So=(c, t', d), \quad (2)$$

where c, d∈Rs. Then the delay t′ of the gate is defined as

$$t'=t+\Delta R(\Delta F), \quad (3)$$

where ΔR(ΔF) is called a response delay for the response state, which represents a rise delay (fall delay) of the gate.

Next, consider an n-input gate with a TRF(Transition Rise/Fall) delay, and let St be an input signal of the gate. In this case, assume that St is given as

$$St=(a, t1, t2, b), \quad (4)$$

where a, b∈Rs, and t1 (t2) denotes the start time (end time) for an upward transition or a downward transition of the signal. Then output So can be defined as

$$So = (c, t1', t2', d), \tag{5}$$

where c, d∈Rs. Then delays t1′ and t2′ of the gate are given by

$$t1' = t1 + \Delta R(\Delta F) \tag{6a}$$

and

$$t2' = t2 + \Delta R(\Delta F). \tag{6b}$$

Let Td be a transition delay for the transition state, given by $Td = t2' - t1'$, where 0 Td M. In order to obtain correct timing result, we restrict maximum value M, where M = R (F)/2. Thus the TRF delay model based on a 5-valued signal set can allows the use of both response and transition delays on the output. In the present version, a distinct initial value for the transition delay is assigned to each logic gate having a primary input.

For example, consider a 2-input AND gate with two input signals St1 and St2, represented as

$$St1 = (0, t1, t2, 1) \tag{7}$$

and

$$St2 = (0, t3, t4, 1). \tag{8}$$

If the gate is simulated using the proposed delay model on the assumption that $t1 < t3 < t4$, $t2 = t4$, then its output So will be given by

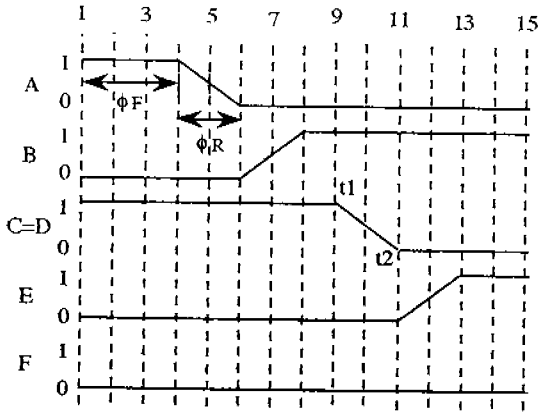
$$So = (0, t3 + \Delta R, t2 + \Delta R, 1). \tag{9}$$

This example shows that the proposed delay model can specify the region of transition delay for a signal propagating through a logic gate. The same process can be extended to multiple-input gates.

In the TRF delay simulation, if Td is not defined (i.e. Td = 0), then the output of a gate is the same as that of the R/F delay simulation except for the hazard detection capability.

To evaluate the performance of this delay model, reconsider a reconvergent circuit shown in (Fig. 1).

If the circuit is simulated using the TRF delay of 2 time units, then there is no output pulse (an output F has logic value 0) for given input signals (Fig. 3) on the assumption that ΔR/ΔF of each gate is 2/3 time unit.



(Fig. 3) Proposed TRF delay simulation

On the other hand, there does not exist a potential hazard which occurs in the Min/Max delay. Therefore the proposed approach can be used as an accurate delay model which performs timing analysis and hazard detection.

4. Hazard Analysis

Hazards in a combinational circuit can be classified into static and dynamic hazards depending upon initial and final values of an output, respectively[3]. Static hazards involve signals which should remain

stable. But, dynamic hazards are associated with signals which are supposed to change values.

In this chapter, modified logic simulation technique is briefly introduced to implement the proposed hazard detection algorithm. Also a new hazard detection based on TRF delay model is proposed to realize hazard-free combinational and sequential circuits.

4.1 Simulation technique

A selective trace, event-driven simulation[8] is devised for reducing the simulation time in such a way that if an event does not occur at any input of an element, then no event is produced at the output. A hazard detection procedure to be discussed here is based on an event-driven simulation algorithm. Assume that this simulation algorithm uses a time wheel with event lists implemented as a circular list[9].

Logic simulation is to be run until the current time does not exceed the maximum time and is repeated for a set of events to be scheduled at the same time. Gate evaluation is performed in this repeat statement, and an actual hazard analysis routine is called whenever an old output value is not equal to a new output value after evaluating a target gate and H[net_no] is not equal to 0. H[net_no] is used for store the schedule times, where H[] denotes one-dimensional array, and net_no represents an identifier(so called net number) for the output signal line of each gate. Finally the simulation time is shifted by increasing the index into the time wheel.

4.2 Hazard detection algorithm

For a hazard analysis, timing analysis is first performed by means of a 5-valued logic simulator which allows the use of both response and transition delays on the output of an element. and then hazards are detected through investigating timing relations.

Before performing the actual hazard analysis, some steps are proceeded during the logic simulation. Initially, the start time of simulation is also placed in H[net_no] for handling a scheduling time of each gate. If H[net_no] is equal to the current time, then H[net_no] is reset to 0. This means that there is no longer any event pending for a signal. It is used to check timing relations by comparing the time H[net_no] with the current time.

Now, we consider input signals to a specific gate which has a TRF delay are sampled at some instant of the time in which each event exists. In order to define each event used for timing analysis, consider again a 2-input AND gate with input signals St1 and St2 and the output signal So, given by equations (7), (8), and (9), respectively.

Now, assume that a current event occurs at time t2 of signal St1. On input signal St1, designate time t2 as the current_time,and time t1-1 (a previous time of t1) as the previous_time. We also call time t4 the base_time for input signal St2, and time (t2 + R) the slot_time for output signal So.

```
Procedure hazard_analysis();
begin
(1) h_flag = No;
(2) if an event on any input line is of
        either U or D then
    begin
(3)   if new output value = X then
        a static hazard exists at
        a scheduled time;
(4)   else
      begin
(5)     calculate previous_time and
        base_time for input signals;
(6)     if previous_time > base_time then
(7)       a spike occurs at
          the scheduled time already;
        else
        begin
(8)       if current_time < slot_time and
              h_flag = No then
(9)         a static hazard exists at
            the scheduled time;
```

```
            h_flag = Yes;
        else
          begin
(10)        a dynamic hazard exists at
            the scheduled time;
            h_flag = No;
          end
        end
      end
    end
(11) perform normal scheduling;
(12) if h_flag = Yes then
       read a next input vector;
(13) put H[net_no] equal to slot_time;
end
```

(Fig. 4) Hazard detection algorithm

(Fig. 4) represents a procedure for detecting dynamic-and static hazards. In this algorithm, those times associated with an event such, as current_time and slot_time, are used to check the timing relation of input and output signals of each gate.

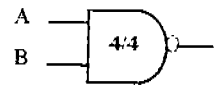
As described the above, hazard analysis routine (Procedure hazard_analysis) is called whenever an old output value is not equal to a new output value after evaluating a target gate and H[net_no] is not equal to 0. Procedure hazard_analysis is initially hazard flag called h_flag is to set No in line (1). Then it is started with checking transition state of the signal if condition described in lines (2) does not met, normal scheduling process is performed in line (11).

If condition of lines(3) met, its output line has a static hazard at a scheduled time; otherwise next condition is checked(4). In line (5), previous_time and base_time for input signals is checked, which are placed in timing wheel, and then the magnitude of those two values are compared(6). If this condition met, a spike is detected at the scheduled time already (7). Otherwise line (8) is scanned.

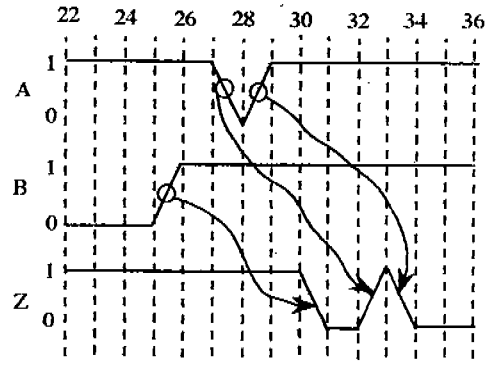
In line (8), both conditions are checked. If this condition met, a static hazard is detected at the scheduled time(9); otherwise a dynamic hazard occurs in line (10).

A dynamic 0-hazard is a presence of a sequence 1010 in a signal which is supposed to change from 1 to 0 while a dynamic 1-hazard is a presence of a sequence 0101 when a signal changes from 0 to 1.

(Fig. 5) shows an example for the creation of dynamic 0 hazard when we apply the procedure shown in (Fig. 4) to a 2-input NAND gate which has inputs A and B, and output Z. Where we assume that this NAND gate is a part of the large circuit, and R/F delays of the gate are 4/4 time units, and a transition time is 1 time unit.



(a) 2-input NAND gate



(b) Timing chart

(Fig. 5) An example for the generation of a potential hazard

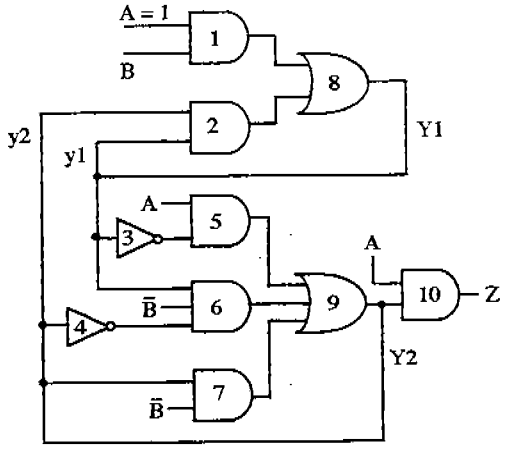
If the gate is simulated by using the TRF delay with given input signals of (Fig. 5(b)), then the simulation result is obtained as shown in (Fig. 5(b)). It is assumed that an input a contains a static hazard (spike) at time 28.

As can be seen from this simulation result, a poten-

tial hazard (dynamic 0-hazard) is generated at an output F between times 30 and 34. In this case our logic simulator reports warning massage such as Hazard occurred time:33, node:z to user. In this work, a race is treated in an asynchronous sequential circuit as a hazard when the race is created by simultaneous changes of several inputs.

5. Experimental Results

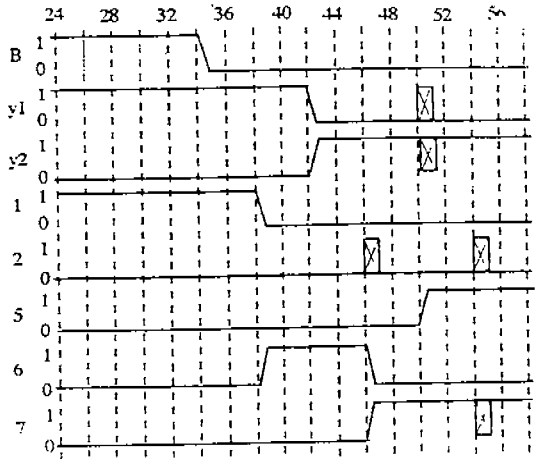
We have implemented the digital logic simulator based on the TRF delay model in the C language on a Sparc-II Workstation. The following example shows how a hazard (race) analysis of an asynchronous sequential circuit shown in (Fig. 6) can be performed.



(Fig. 6) Asynchronous sequential circuit

This circuit is simulated by using a transition delay of 1 time unit on the assumption that R/F delays for NOT gates have 1/1, and the rest are 4/4 time units. Now, we consider an input B is changed from logic 1 to 0 (the circuit is in stable state 10, with AB=11). In this case, a simulation result is shown in (Fig. 7), and transition and state tables are illustrated in ⟨Table 1⟩.

From ⟨Table 1⟩, it can be observed that the circuit makes a transition from a stable state D to an unstable state B* when only input B changes from 1 to 0 (i.e. AB=10). In this case, outputs y1 (Y1) and y2 (Y2) is simultaneously changed, and the output of Gate 2 produces a potential error (denoted as X) between times 46 and 47. Actually, this error (a



(Fig. 7) Simulation results for (Fig. 6)

⟨Table 1⟩ Transition and state tables for ⟨Fig. 6⟩

y1y2 \ AB	00	01	11	10
00	(00)	(00)	11	01
01	(01)	00	11	(01)
11	(11)	10	10	(11)
10	—	00	(10)	01*

(a) A transition table

	00	01	11	10
A	A	A	C	B
B	B	A	C	B
C	C	D	D	C
D	—	A	D	B*

(b) A state table

nomentary 1 pulse) may or may not occur at the output of Gate 2 according to delay times assigned to each gate in the circuit. As can be seen from (Fig. 7), the TRF delay simulation can accurately analyze the timing problem.

〈Table 2〉 shows an experimental results for various combinational and sequential circuits. CKT5 represents the combinational circuit with the function of 4-bit ALU(Arithmetic Logic Unit) while the rest are sequential circuits which are composed of synchronous decade counters(SN 56LS160 A).

〈Table 2〉 Experimental results

Circuit Name	logic type	no. of gates	no. of events	CPU time (min)
CKT1	Seq.	576	108562	1.35
CKT2	Seq.	1152	99853	1.52
CKT3	Seq.	2304	93422	2.31
CKT4	Seq.	4608	65498	6.53
CKT5	Com.	666	93344	0.54

6. Conclusion

In a 5-valued logic simulation environment, the efficient hazard analysis algorithm based on a new TRF delay model. has been described, which can be used as a tool for the analysis of combinational and sequential circuits. In hazard analysis process, the evaluation of the gates is performed by using lookup table approach which specifies the operation of the logic circuits. Thus the simulation time is fast although the number of events(both transition and response states) are increased during analyzing hazards.

Conventional logic simulation system[2-4] with the R/F delay cannot perform the timing analysis accurately because the system cannot allows the use of the transition delays for upward and downward transitions.

Since the proposed delay model allows the use of both response and transition delays, it is effective not only for verifying timing but also for predicting the probability of a race and hazard condition. A 5-valued logic simulator generates a log file related to the existing the static and dynamic hazards during simulation of a given digital circuit, and the realization of the hazard-free digital logic circuits is possible from the synthesis of the simulation result which contains hazard information.

There still remains a problem of how to find accurate delay information for each logic gate in the circuit.

References

[1] A. R. Newton and A. L. Sangiovanni-Vincentelli, "Computer-Aided Design for VLSI Circuits," IEEE Computer, Vol. 19, No. 4, pp. 38-60, Apr. 1986.

[2] M. A. Breuer and A. D. Friedman, Diagnosis and Reliable Design of Digital System, Computer Systems Press, Woodland Hills, CA, 1976.

[3] E. B. Eichelberger, "Hazard Detection in Combinational and Sequential Switching Circuits," IBM J. Res. & Dev., Vol. 9, pp. 92-99, Mar. 1965.

[4] M. A. Breuer, and L. Harrison, "Procedures for Eliminating Static and Dynamic Hazards in Test Generation," IEEE Trans. on Comput., Vol. C-23, No. 10, pp. 1069-1078, Oct 1974.

[5] G. Fantauzzi, "An Algebraic Model for the Analysis of Logical Circuits," IEEE Trans. on Comput., Vol. C-23, No. 6, pp. 176-181, 1974.

[6] N. Ishiura and S. Yajima, "On Time Symbolic Simulation," IEICE Technical Report, VLD87-112, 1987 (in Japanese).

[7] Y. H. Jea and S. A. Szygenda, "Mapping and Algorithm for Gate Modeling in a Digital Simulation Environment", IEEE Trans. on Circuits and Systems, Vol. CAS-26, No. 5, May 1979.

[8] P. Agrawal and W. Dally, "A Hardware Logic Simulation System," IEEE Trans. on Computer-Aided Design, Vol. CAD-9, No. 1, pp. 19- 29,

Jan. 1990.

[9] M. S. Kang, C. D. Lee, and Y. U. Yu, "LOSIM: Logic Simulation Program for VLSI," KITE Trans., Vol. 26, No. 5, pp. 108-116, May 1989 (in Korean).

강 민 섭

1979년 광운대학교 전자통신공학과 졸업(공학사)
1984년 한양대학교 전자공학과 졸업(공학석사)
1992년 일본 오사카대학교 전자공학과 졸업(공학박사)
1984년~1993년 한국전자통신연구소 선임연구원
1986년~1987 일본 오사카대학교 연구원
1993년~현재 안양대학교 컴퓨터학과 조교수
관심분야: ASIC 설계 및 CAD, VLSI 테스트, 컴퓨터 구조