Original Paper

A Clock Partitioning Method Using Initialization Complexity for Sequential Testability

Haruhiko TAKASE, Terumine HAYASHI and Tsuyoshi SHINOGI
(Department of Engineering)

(Received September 16, 1998)

Abstract

This paper presents a new method for sequential testability based on clock controlling. First, we define the concept "initialization complexity" which measure the difficulty of initializing states of flip-flops. Next, we propose new criteria for determining clock control groups based on this concept. Finally, we show the effectiveness of our method in the viewpoint of fault detection with experimental results for benchmark circuits.

Key words : Sequential circuit, Design for testability, Fault detectability

1 Introduction

It is well known that the test generation for sequential circuits is more complex than one for combinational circuits. To improve the testability of sequential circuits, a number of papers have considered the application of clock controlling to sequential circuits [1–5]. The outline of clock controlling is as follows. First, we divide flip-flops (FFs) in the circuit into some clock control groups. Each group is controlled by its own clock. Next, during the test mode, we enable clocks of some groups. Otherwise, we enable clocks of all FFs.

Many criteria and algorithms are proposed to determine clock control groups. In [3], the FFs are partitioned to reduce the number of cycles and the path lengths in each partition. In [4], the components of strongly connected FFs are in the same clock control group. In [5], the FFs are partitioned to reduce *darn* state which is both difficult to reach and required to detect some faults.

In this paper, we propose a new method for FFs into clock control groups in sequential circuits to improve the testability. We shorten the minimum length of input sequences for initializing FFs. With this point of view, we determine clock control groups based on "initialization complexity" which represents a difficulty for initializing each FF. The value of initialization complexity for an FF is closely related to the minimum length of the input sequences for initializing the FF.

In the viewpoint of the purpose, our method is similar to [5]. The purpose is to reduce the state which is difficult to reach. In [5], it is evaluated by computing probabilities of FF states, approximately. In our method, it is evaluated by computing initialization complexity.

The rest of the paper is organized as follows. Our basic idea is shown in section 2. Section 3 shows an algorithm to compute initialization complexity. Section 4 shows an algorithm to divide FFs into clock control groups. Section 5 shows some experimental results and Section 6 concludes the paper.

Table 1: Initialization complexity

Gate	$C_0(l)$	$C_1(l)$
AND(l_A, l_B)	$\min\{C_0(l_A), C_0(l_B)\}$	$\max\{C_1(l_A), C_1(l_B)\}$
OR(l_A, l_B)	$\max\{C_0(l_A), C_0(l_B)\}$	$\min\{C_1(l_A), C_1(l_B)\}$
XOR(l_A, l_B)	$\min\{\max\{C_0(l_A), C_0(l_B)\},$ $\max\{C_1(l_A), C_1(l_B)\}\}$	$\min\{\max\{C_0(l_A), C_1(l_B)\},$ $\max\{C_1(l_A), C_0(l_B)\}\}$
NOT(l_A)	$C_1(l_A)$	$C_0(l_A)$

2 Clock Control Groups

We will begin with a simple example that clock controlling makes an initializing sequence short. For an 8-bit counter, the length of the input sequence to initialize bit#7 to 1 from the state (0, 0, $\cdots$, 0) is 2^7. Assume that clock control groups A =(bit#7, bit#6, bit#5, bit#4) and B =(bit#3, bit#2, bit#1, bit#0) are given. First, we supply clock signal only to FFs in the group B to initialize the state of the group B from (0, 0, 0, 0) to (1, 1, 1, 1). The length of the input sequence is $2^4 - 1$. Next, we supply clock signal only to FFs in the group A to initialize the state of the group A from (0, 0, 0, 0) to (1, 0, 0, 0), keeping the state of group B (1, 1, 1, 1). The length of the input sequence is 2^3. Therefore, the total length of the input sequence is $2^3 + (2^4 - 1)$. This example shows that the initialization sequence for the 8-bit counter is longer than one for the two 4-bit counters. In general, dividing a counter is useful to make an initializing sequence short.

Above discussions lead us to a news idea for partitioning FFs. We note the shortest length of input sequence which initialize each FF to arbitrary state. For FFs in a counter, FFs representing higher bit have longer one and FFs representing lower bit have shorter one. From above discussions, we may say that this length of a general circuit is substituted for the bit order of FFs in a counter. Here, we call this length as "initialization complexity".

According to initialization complexity, We partition FFs into some clock control groups. The outline of the procedure for partitioning FFs is described as follows. First, we calculate the initialization complexity for each FF in the circuit. Next, we divide FFs into clock control groups according to the initialization complexity of each FF. As a result, a large counter is divided into small counters. Therefore, the length of the input sequence which initializes the circuit becomes short.

3 Initialization complexity

The initialization complexity for a line l is constructed by $C_0(l)$ and $C_1(l)$. $C_v(l) + 1$ means the lower bound of the length of initializing sequences to make the state v of line l from initial state (X, X, $\cdots$, X). The initialization complexity is calculated as follows. For a primary input PI$_i$, $C_0(\mathrm{PI}_i)$ and $C_1(\mathrm{PI}_i)$ are equal to 0. For an AND-gate output l_A, $C_1(l_A)$ is equal to the maximum value of $C_1(l_k)$ for each input line l_k of the gate. Because, to make the logic value 1 on the output line, we have to make the logic value 1 on all input lines of the gate. $C_0(l_A)$ is equal to the minimum value of $C_0(l_k)$ for each input line l_k of the gate. Because, to make the logic value 0 of the output line, we have to make the logic value 0 on one of its input lines. Table 1 shows calculation of the initialization complexity for some kind of gates whose inputs are l_A and l_B. To calculate $C_1(l_F)$ and $C_0(l_F)$ for output l_F of FFs, we add 1 to the value of its input respectively. Here, we define the initialization complexity $C_{\max}(l)$ as $\max\{C_0(l), C_1(l)\}$.

The initialization complexity for all FFs in the circuits are calculated by following procedures.

1. Initialize $C_v(l)$ for all signal lines.

$$C_0(l), C_1(l) \leftarrow \begin{cases} 0 & \text{(if } l \text{ is primary input)} \\ \infty & \text{(otherwise)} \end{cases}$$

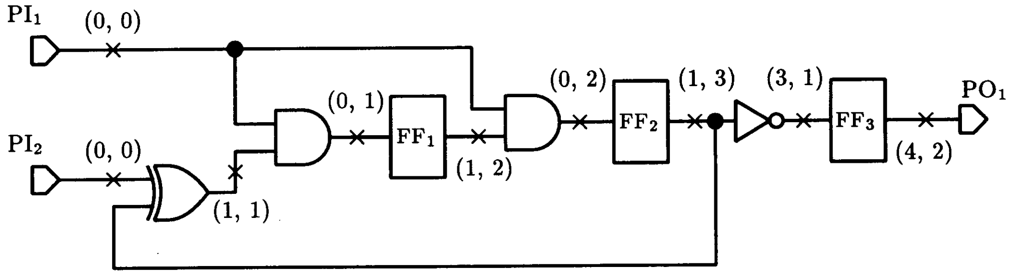

Figure 1: Example of Initialization complexity

2. Calculate $C_v(l)$ for the outputs of all gates. (Table 1)
3. Calculate $C_v(l)$ for the outputs of all FFs.

$$\begin{aligned} C_0(l) &\leftarrow C_0(l_A)+1 \\ C_1(l) &\leftarrow C_1(l_A)+1 \\ & \quad l_A \text{ is the input of the FF.} \end{aligned}$$

4. Repeat step 2 and 3, until all $C_v(l)$ are converged

Figure 1 shows example of calculation of $C_{\max}(l)$ for the actual circuit. In the figure, (a,b) means $C_0(l)=a$ and $C_1(l)=b$ at each cross mark. The initialization complexity for FF$_1$, FF$_2$ and FF$_3$ are 2, 3 and 4, respectively.

$C_0(l)$ and $C_1(l)$ is calculated by the similar way to the sequential testability measure $SC_0(l)$ and $SC_1(l)$ proposed by Goldstein [6], but there is a little difference due to the objective. The difference lies in calculation rules for gates. Namely, $C_0(l)$ and $C_1(l)$ is based on the minimum and maximum operation of the values of the inputs. For example, for an AND gate, $C_1(l)$ is equal to maximum among the value $C_1(l_k)$ of its inputs l_k, while $SC_1(l)$ is equal to the sum of the value $SC_1(l_k)$ of its inputs l_k.

4 Partitioning

In this section, we show the criteria to determine clock control groups using the initialization complexity $C_{\max}(l)$ for each FF. Our basic idea is as follows. With clock controlling, dividing large counter into small counters as clock control groups makes the length of initialization sequence short. We extend this idea for general circuits.

The criterion is described as following cases. They are represented by the initialization complexity of each FFs.

(1) $2 \le C_{\max}(Q) < \infty$.
Put each FF into group k ($k = 1, 2, \cdots$). k is satisfying the following equation for given step width M.

$$M \cdot (k-1) + 2 \le C_{\max}(Q) \le M \cdot k + 1.$$

(2) $C_{\max}(Q) = 1$.
Any FF does not belong to any groups. It is able to be initialized directly from only primary inputs. Therefore, it is not necessary to control its clock.

Table 2: The initialization complexity and clock control groups

Circuit name	Number of FFs				Number of clock control groups ($M = 4$)
	Total	$C_{\max} = 1$	$C_{\max} \geq 2$ $C_{\max} < \infty$	$C_{\max} = \infty$	
s208	8	0	8	0	2
s420	16	0	16	0	4
s838	32	0	32	0	8
s5378	179	33	146	0	7
s9234	228	20	36	172	3
s15850	597	4	225	368	10

Table 3: Fault detectability

Circuit name	Number of faults				
		Single clock		Clock control	
	Total	$3v$-untestable	Undetectable	$3v$-untestable	Undetectable
s280	215	78	65	14	0
s420	430	251	226	28	0

(3) $C_{\max}(Q) = \infty$.

Any FF does not belong to any groups. It is not able to be initialized with 3-values(0, 1 and X) simulation. Therefore, it is meaningless to control its clock.

For FFs which do not belong to any groups, their clocks are always supplied as in the original circuit.

It is difficult to decide the suitable M in criterion 1. Using small M, we get much clock control groups. As a result, the clock controlling requires much circuits. Using large M, we get fewer clock control groups. As a result, the length of the initialization sequence is not shorten. Deciding suitable M is an unsettled question.

We show a simple example. For a binary counter, $C_{\max}(Q_i)$ is equal to $i + 2$, where Q_i is the FF's output of i-th bit ($i = 0, 1, \cdots$). In the case $M = 8$, 16-bit binary counter is divided into two clock control groups as $\mathrm{FF}_{15} \sim \mathrm{FF}_8$ and $\mathrm{FF}_7 \sim \mathrm{FF}_0$.

5 Simulation

In this section, we investigate the improvement of sequential testability with clock control.

We show some experimental results for ISCAS'89 benchmark circuits. In Table 2, we show the initialization complexity and the number of clock control groups for $M = 4$. The circuit is divided into a small number of clock control groups. For other circuits in the ISCAS '89, we also get $0 \sim 10$ groups. These results show that the number of groups does not grow as growing the scale of circuits.

Table 3 shows fault detectability. The column "$3v$-untestable" in the table means the number of faults which are impossible to be detected by 3-values (0,1,X) fault simulation from initial state $\{X, X, \cdots, X\}$. The column "Undetectable" in the table means the number of faults which is impossible to distinguish faulty and fault-free circuits by any input sequence. The result shows that clock controlling brings us high fault detectability. It must be noted that there is no undetectable fault in the result.

6 Conclusion

We conclude this paper as follows. First, we have presented the new concept "initialization complexity" measuring the difficulty of initializing states of FFs. Next, we have proposed new criteria determining clock control groups based on this concept. Finally, we have shown the effectiveness of our proposal in the viewpoint of fault detection with experimental results for benchmark circuits.

References

[1] M. Abramovici, K. B. Rajan and D. T. Miller, "FREEZE ! : A New approach for testing sequential circuits," Proc. 29th Design Automation Conf., pp.22–25 (1992).

[2] V. D. Agrawal, S. C. Seth and J. S. Deogun, "Design for testability and test generation with two clocks," Proc. 4th Int. Symp. VLSI Design, pp.112–117 (1991).

[3] K. L. Einspahr, S. C. Seth and V. D. Agrawal, "Clock partitioning for testability," Proc. Third Great Lakes Symp. on VLSI, pp.42–46 (1993).

[4] S. Baeg and W. A. Rogers, "Hybrid design for testability combining scan and clock line control and method for test generation," Proc. Int. Test Conf. 1994, pp.340–349 (1994).

[5] K. B. Rajan, D. E. Long and M. Abramovici, "Increasing testability by clock transformation (Getting rid of those darn states)," Proc. 14th VLSI Test Symp., pp.224–230 (1996).

[6] L. H. Goldstein, "Controllability / observability analysis of digital circuits," IEEE Trans. on Circuits and Systems, vol. CAS-26, no.9, pp.1066–1077 (1993).