

# 行政院國家科學委員會專題研究計畫 期中進度報告

## 後次微米時代新興電子設計自動化技術之研究--子計畫 三：角落錯誤之矽除錯(2/3) 期中進度報告(完整版)

計畫類別：整合型  
計畫編號：NSC 98-2220-E-009-023-  
執行期間：98年08月01日至99年07月31日  
執行單位：國立交通大學電子工程學系及電子研究所

計畫主持人：周景揚  
共同主持人：趙家佐

處理方式：本計畫可公開查詢

中 華 民 國 99 年 05 月 28 日

行政院國家科學委員會補助專題研究計畫 ☐ 成果報告  
☒ 期中進度報告

計畫名稱：後次微米時代新興電子設計自動化技術之  
研究 - 子計畫三：角落錯誤之矽除錯 (2/3)

計畫類別：☐ 個別型計畫 ☒ 整合型計畫

計畫編號：NSC 98-2220-E-009-023-

執行期間：98年8月1日至99年7月31日

計畫主持人：周景揚

共同主持人：趙家佐

計畫參與人員：林步青、韓秉勳、石銘恩、鄭安哲、張瀚元、陳詣航、  
張智為、曾遵銘、林政偉、楊皓宇、陳弘昕、涂偉勝、陳擴安、郭淳  
仁、黃欽達、張玟翔、徐浩文、王易民

成果報告類型(依經費核定清單規定繳交)：☒ 精簡報告 ☐ 完整報告

本成果報告包括以下應繳交之附件：

- ☐ 赴國外出差或研習心得報告一份
- ☐ 赴大陸地區出差或研習心得報告一份
- ☐ 出席國際學術會議心得報告及發表之論文各一份
- ☐ 國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫  
及下列情形者外，得立即公開查詢

☒ 涉及專利或其他智慧財產權，☐ 一年 ☒ 二年後可公開查詢

執行單位：

中 華 民 國 99 年 5 月 27 日

後次微米時代新興電子設計自動化技術之研究 – 子計畫三：  
角落錯誤之矽除錯(2/3)

Emerging EDA Technologies beyond DSM Era – Sub-Project 3 :

Silicon Debug for Hard-Corner Design Errors

計畫編號: NSC97-2220-E-009-034

執行期間: 98 年 8 月 1 日 至 99 年 7 月 31 日

主持人: 周景揚 交通大學電子工程系教授

共同主持人: 趙家佐 交通大學電子工程系助理教授

一、 中文摘要

由於越來越高的設計複雜度，在目前的晶片製作中第一次製作的晶片通常都是失敗的或是僅有非常低的良率。我們可由這批失敗的晶片中，收集晶片的錯誤行為，並且經由這些錯誤行為找到並且更正我們設計中錯誤的部分。但是，由於不停增加的設計複雜度以及製成的不穩定性，錯誤分析已經變的越來越困難且比以前需要更多的時間。

在第一次晶片製作之後，對於給定的一個錯誤設計，除錯最重要的第一步就是找出第一個錯誤的時間。只要能得到第一個錯誤時間的範圍，就能將該時段的電路展開成數個組合電路連續串接的狀態，並根據該時段所擁有的信號重建當時的電路狀態 使用既有的診斷錯誤工具進而找到真正錯誤的地方。

在這個子計畫中，我們提出了在矽除錯中追蹤緩衝區的訊號選擇問題。追蹤緩衝區是一塊小部分的記憶體可在執行時同時儲存系統的資訊。針對如何選擇要記錄的訊號，我們提出了一條件式的選擇方法，選擇一些正反器作為儲存入追蹤緩衝區的信號。我們將此問題套用至 minimum feedback vertex set 問題，此問題已經是大家熟知的 NP-complete 問題。在本計畫設定之下，在測試電路 s9234 中，可直接觀察或推論出超過 97% 正反器的狀態而只需要選取 32 個正反器。即使在最差的測試電路 s13207 中，亦可得到超過 50% 的正反器狀態。這些實驗數據並沒有考慮可直接由輸出端觀察的情形，若配合上直接觀察輸出端的部分，可得到的正反器狀態會再增加。執行時間是可以接受的，並不會因為選擇的正反器數量不同而有所影響。

## 二、 英文摘要

Due to the high design complexity, the first silicons of today's ICs usually fail or have very low yield. Based on these failed chips, we can collect faulty behaviors, identify and correct the failure design. However, with the nonstop increasing design complexity and uncertainty of process variation, the failure analysis becomes more and more difficult and needs much time than before.

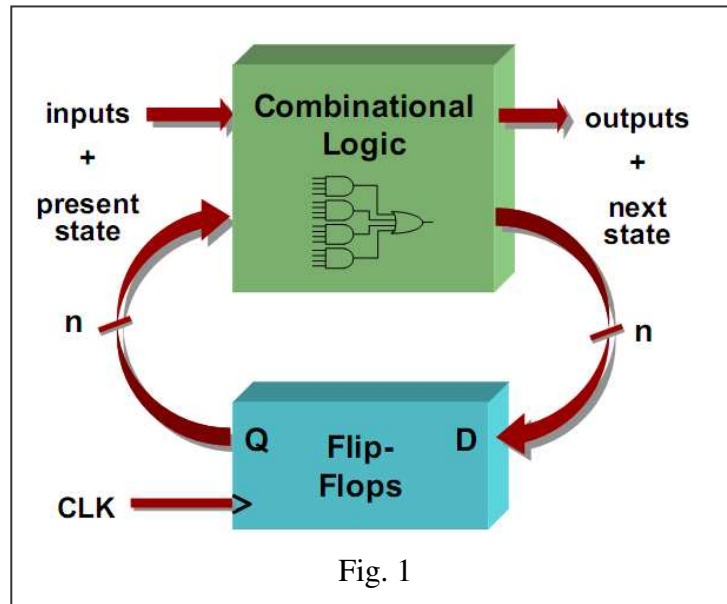
For debugging a given faulty design after first silicon, the most important thing is to find the first error cycle. Once we have the range of first faulty cycle, we can flatten the circuit into several combinational circuits and use existing techniques to diagnosis the combinational circuit of the faulty design.

In this project, we propose the problem of trace-buffer selection for silicon debug and give a heuristic methodology to choose the observation flip flops for trace buffer. Trace buffer is a small set of memories on chip and could record the desired information at run time. We model this problem into minimum feedback vertex set (MFVS) problem which is a well-known NP-complete problem. With our assumptions, we can observe or infer the status of flip flops about 97% in benchmark s9234 with 32 observation flip flops. Even the worst case of our benchmark, we still could observe or infer more than 50% flip flops. These experimental results are not combined with the output information, which is trivially information we can get. The status of flip flops could be inferred more than the experimental results if we consider with the output information. The runtime is all acceptable and do not disturb with different size of observation flip flops.

### 三、 研究方法及成果

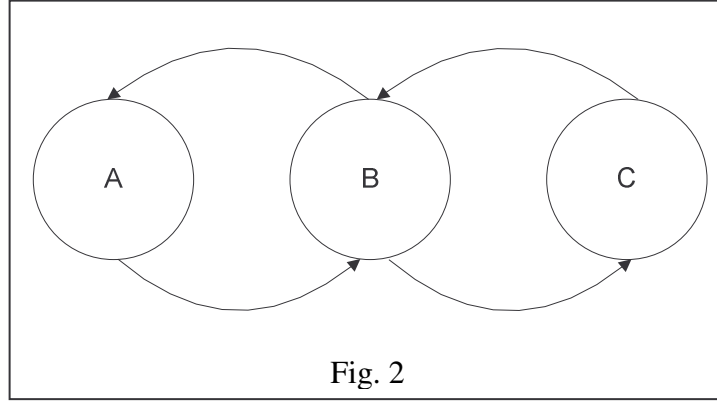
#### Trace-buffer selection for silicon debug

To observe the faulty signal as soon as possible, we have to not only choose the correct flip flop but also minimize the number of flip flop to reduce the cost of trace buffers. Every design could be represented as following:



If the faulty signals propagate to the output, it is obviously the output could be observed at each clock cycle. But if the faulty signals only propagate to the flip flops, these faulty signals could not be observed at the first faulty cycle, and these faulty signals even loop for ten or hundred cycles before it can be observed at outputs.

The trace buffer is a set of memories on the chip. With a small part of the combination driven circuit, the trace buffer can save the status of internal signals until it is full. The status of internal signals has much help for failure analysis. According the information, we can identify the faulty cycle more accurately and reduce the faulty candidates and the time of debugging. Thus, the signals we choose are very important. In this project, we only choose the signals from flip flops, which is called observation flip flops, because the number of flip flop is much less than the combinational circuit and we can infer more signals by forward or backward implications from these observation flip flops.



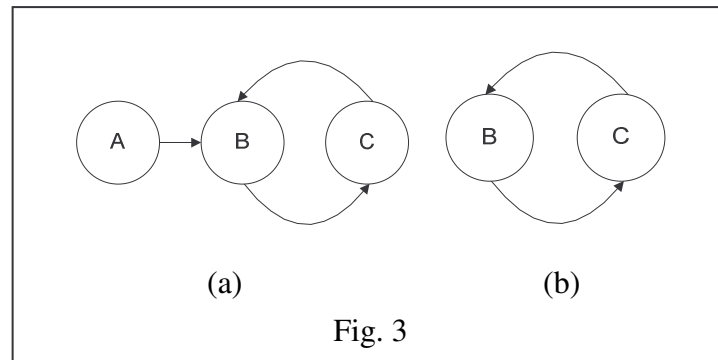
Here, we use an example to illustrate our main idea. In Fig. 2, we use the node to represent the flip flops and the arc to demonstrate the connections between flip flops and ignore the combination circuit. The arc from node A to node B means that there is a directly path from flip flop A to flip flop B and the signal can reach flip flop B in one cycle.

The easiest way to observe fault at flip flops is to choose all flip flops but this method will cost too much cost. If there is only one flip flop we can choose, the situation will be different according to the flip flop. For example, if we choose flip flop A to be our observation flip flop, we can observe the faulty signal in one cycle if the faulty signal happens at flip flop A or B. If the faulty signal happens at flip flop C, we cannot guarantee how many cycles we need to observe the faulty signal at observation flip flop A. The reason is that there is a loop between flip flop B and C. If the faulty signal happens at flip flop C, the faulty signal could be propagated by the path  $(BC)_nA$ , and we cannot decide what the value  $n$  is. The best choose in this example is flip flop B. If the flip flop is set to the observation flip flop, we can observe the faulty signal in one cycle whether the faulty signal happens at flip flop A or C.

From the above example, we have the solution for the problem of trace buffer selection. If we can choose the minimum number of flip flops to break all the loops between any two flip flop, we can observe the faulty signal in the acceptable cycles. For efficiently breaking all the loops, we model our problem to Minimum Feedback Vertex Set problem (MFVS problem).

The minimum feedback vertex set problem is a well-known problem. In the mathematical discipline of graph theory, a feedback vertex set of a graph is a set of vertices whose removal leaves a graph without cycles. The minimum feedback vertex set problem is an NP-complete problem in computational complexity theory. If we model the flip flops to be the node and each direct path to be the arc in a graph, the trace-buffer selection problem is the same and can be model to the MFVS problem. For our trace-buffer selection problem, if we can remove all the cycles formed by flip flops in the design, the design could be decomposed into acyclic combinational circuit, and the maximum cycle we observe a faulty signal is the longest path of the new circuit.

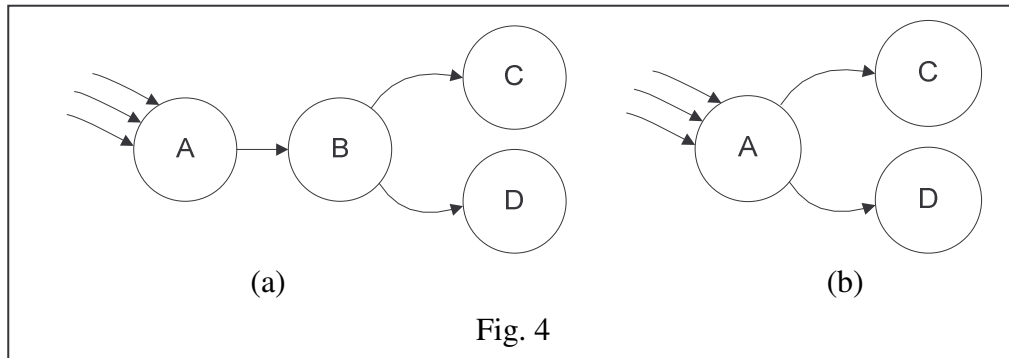
Because MFVS problem is NP-complete, we propose some heuristic method to break the loops.



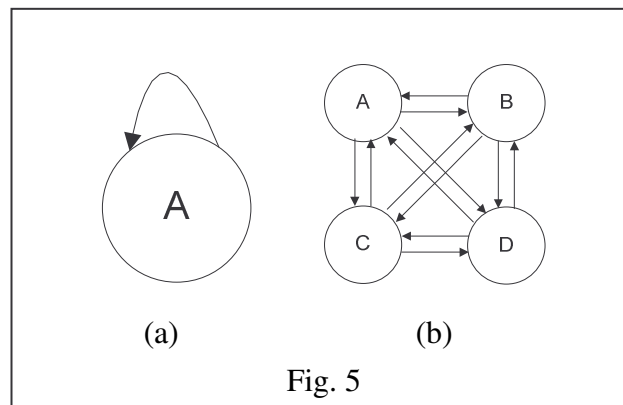
The first assumption is that once the node has no input or output as the node A in Fig. 3(a), the node A could be eliminated and the new figure is shown as Fig. 3(b). The assumption is trivial because the node with no input or output cannot break any loop and can be removed from the candidate set of observation flip flops.

The second assumption is the extension of first one. If a node with only one input/output as the node B in Fig. 4(a), the node could be merged into its input/output to reduce the number of nodes for further selection. From Fig. 4(a), we can observe that node B has only one input from node A. It means whether we choose node A or node B to be observation flip flop, we can all observe the faulty signal in cycle N or cycle N+1. For this

situation, we merge these two nodes to reduce the candidate set.



The last assumption is the node we must choose to be the observation flip flop. The first example is the node with self-loop, as shown in the Fig. 5(a). If we do not choose node with self-loop, the faulty signal could be loop at that node and could not predict what the real error cycle is. The second example is the nodes with clique. We have to choose the N-1 node for N-clique to guarantee we can observe faulty signal in 1 cycle. Once we do not choose N-1 nodes, the situation of Fig. 5(a) could be happened and form the self-loop.





### Experimental results:

We test our method on ISCAS89 circuits and show three results as following:

Circuit	FFs	FF coverage (%)	Size of SW	Run time (sec)
s9234	8	77.73	5.1	0.22
	16	87.68	4.9	0.23
	32	97.16	4.8	0.26
s13207	8	45.44	13.82	1.21
	16	55.46	12.91	1.45
	32	64.28	12.52	2.39
s15804	8	58.46	6.33	112.04
	16	70.35	6.54	111.98
	32	75.71	6.20	114.98

Table 1

The first column of Table 1 is the name of circuits. The second column is the number of flip flop we pick up. The third column is the flip flop coverage of total flip flop. The forth column is the size of SW (suspect window), and it means that once we find an error signal at cycle N, the real fault must can be found between cycle N-SW and cycle N. The last column is the run time of our program.

According to Table 1, we can find that the more FFs we pick up, the more coverage we have, and the size of suspect window will be less. The number of FFs and the coverage rate will be different dependent on designs. For example, we can get 97.16% coverage with 32 FFs at s9234, but we only get 64.28% coverage with the same number of FFs at s13207. The size of suspect windows shows that, after picking the target FFs, we can observe the faulty signals at target FFs or outputs in 5 cycles for s9234, and the run time is acceptable for all these circuits.

#### 四、 結論與討論

We propose the problem of trace-buffer selection for silicon debug and model this problem into minimum feedback vertex problem, a well-known NP-complete problem. With our assumptions, we can observe or infer the status of flip flops about 97% in benchmark s9234 with 32 observation flip flops. Even the worst case of our benchmark, we still could observe or infer more than 50% flip flops. These experimental results are not combined with the output information, which are trivially information we can get. The status of flip flops could be inferred more than the experimental results if we consider with the output information. The runtime is all acceptable and do not disturb with different size of observation flip flops.

#### 五、 參考文獻

- [1] Junpei Nonaka, Toshio Ishiyama and Kazuki Shigeta, “Design for Failure Analysis inserting Replacement-type Observation Points for LVP,” International Test Conference, pp. 1-10, 2009.
- [2] Sandesh Prabhakar and Michael Hsiao, “Using Non-trivial Logic Implications for Trace Buffer-Based Silicon Debug,” Asian Test Symposium, pp 131-136, 2009.
- [3] Xiao Liu and Qiang Xu, “Trace Signal Selection for Visibility Enhancement in Post-Silicon Validation,” Design, Automation & Test in Europe Conference, pp 1338-1343, 2009.
- [4] Ho Fai Ko and Nicola Nicolici, “Algorithms for State Restoration and Trace-Signal Selection for Data Acquisition in Silicon Debug,” IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems, Vol. 28, No. 2, pp 285-297, Feb. 2009.
- [5] Yong Liu, Timwah Luk, and Scoot Irving, “Parameter Modeling for Wafer Probe Test,” IEEE Transactions on Electronics Packing Manufacturing, Vol. 32, No. 2,

81-88, Apr. 2009.

- [6] Kuang Chao Fan, Yejin Chen, and Weili Wang, "Probe Technologies for Micro/Nano Measurements," International Conference on Nanotechnology, pp. 989-993, 2007.
- [7] Yu-Chin Hsu; Furshing Tsai; Wells Jong; Ying-Tsai Chang, "Visibility enhancement for silicon debug," Design Automation Conference, pp. 13-18, July 2006.
- [8] Savir, J; Ditlow, G S; Bardell, P H, "Random Pattern Testability," IEEE TRANS. COMP. Vol. C-33, no. 1, pp. 79-90. 1984.
- [9] Goldstein, L.H.; Thigpen, E.L., "SCOAP: Sandia Controllability/Observability Analysis Program," Design Automation Conference, pp. 190-196, June 1980.
- [10] Reddy, P.V.C.V., "Testability measure and analysis," ASIC Conference and Exhibit, 1992., Proceedings of Fifth Annual IEEE International, pp.129-138, Sep 1992.

## 六、計畫成果自評

This year, we have developed a framework to solve the trace-buffer selection problem for silicon debug. By inserting the trace buffers, we have more chance to observe the faulty signals before they propagate to the outputs. We can observe or infer more than 50% flip flops in our benchmarks within 32 observation flip flops and acceptable run time.