國　立　清　華　大　學

碩　士　論　文

題目：考慮單元位置之 X 填值以減輕在即時性測試下的電流電阻壓降效應

A Physical-Location-Aware X-filling Method for IR-Drop Reduction in At-Speed Scan Test

系 所 別：資訊工程學系碩士班

學號姓名：9562537 林翌聖（I-Sheng Lin）

指導教授：黃婷婷　博士　（TingTing Hwang）

中華民國九十七年八月

致謝

在研究所這兩年的學習裡，除了在專業知識的精進上，更學習到許多的社會、人生經驗。這對於未來的我，相信會有十足的幫助。在這篇論文的正文之前，我要先感謝大家以表達我的謝意。

首先是我的父母，在求學的過程中，一路默默地支持我。在我忙的昏天黑地，喘不過氣之時，總是有適時的關心，使我的心情能永達保持愉快，才有動力能夠完成碩士學業。

其次要感謝我的指導教授黃婷婷老師。在每次的 meeting 時，老師總是能提出有趣的觀念、想法與問題。對於我這位剛踏入研究領域的菜鳥來說，總是大開眼界並受益無窮。爾後，在進行研究時，每當陷入瓶頸時，只要與老師討論，總是能讓問題迎刃而解。還有要感謝實驗室的博士班學長姊，不厭其煩地在課業上、人生經驗給予指導。在這兩年間，聽世梁學長敘述他在求學時代與工作時代的有趣經驗，是一大樂事。柏元、文雯、名詔、昂志等學長姊，各有自己的擅長領域，讓我能夠多方地吸收 CAD 領域的知識。

最後要向哲宇、介俊和鈞澤這些同窗致謝。與你們一起修課、趕 project、打球、吃喝玩樂，為我這兩年的碩士生涯更添色彩。另外要感謝 CAD 壘球隊的各位同學，系運壘球的冠軍讓我這壘球菜鳥有了最美好的回憶。

中文摘要

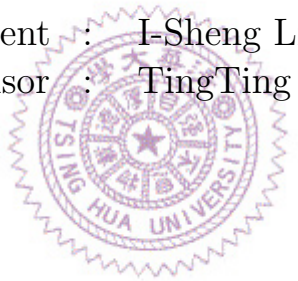
隨著製程的進步、單位元件尺寸的縮小，為了保證電路上的時序限制能夠被滿足，即時性測試被大量地運用在找出電路上的延遲缺陷。然而，在即時性測試的當下，電路會有極為可觀數量的元件同時發生轉換，使得電路上的供電開始不穩，產生所謂的電流電阻壓降效應。這會使電路的效能降低，甚至功能發生錯誤，使電路本身被誤判為錯誤。故我們需要設法降低在即時性測試中的元件轉換率。

為了解決前段所敘述的問題，X 填值的方法經常為人所使用。在即時性測試的測試樣本中，有著高比例的 X 值，而這 X 值可以隨意地填入 0 或 1 。經由將 X 填入較好的值，新產生的測試樣本可以有效地降低元件的轉換率，以防止在即時性測試中，電路上產生嚴重的電流電阻壓降效應。

在這篇論文中，我們提出了兩個方法來幫助我們執行 X 填值的動作。分別是考慮電路裡各元件的實際位置以及向後傳遞來 X 值來決定填值。首先當在某一小區域內的元件大量地發生轉換，可以確定會產生很嚴重的電流電阻壓降效應。這時我們選出一些對於電流電阻壓降效應影響力很大的元件，使用向後傳遞 X 值的方法，以確保這些元件在 X 填值後可以有效降低這小區域內的元件轉換率。

A Physical-Location-Aware X-filling Method for IR-Drop Reduction in At-Speed Scan Test

Student : I-Sheng Lin
Advisor : TingTing Hwang



Department of Computer Science
National Tsing Hua University
HsinChu, Taiwan 30043

Contents

List of Figures

List of Tables

Abstract

In order to ensure that a circuit meets timing requirements, at-speed scan test is widely used to detect delay defects. However, at-speed scan test suffers from the test-induced yield loss. Because the switching activity of whole circuit during test mode is much higher than that during normal mode, the large portion of gates simultaneously switching contributes to serious IR-drop delay. Thus, propagation delay does not meet the timing constraint only at test mode. This IR-drop problem during test mode exacerbates delay defects and results in false failures. In this thesis, we take the *X-filling* approach to reducing the IR-drop effect during at-speed test. The main difference between our approach and the previous *X-filling* methods lies in two aspects. The first one is that we take the spatial information into consideration in our approach. The second one is how *X-filling* is performed. In previous work [7, 8, 9], a forward-propagation approach is taken, while a backward-propagation approach is proposed in this thesis. Compared with the previous work [9], the experimental result shows that we have 26% reduction in the worst IR-

drop and 28% reduction in the average IR-drop. The IR-drop reduction also improves the IR-drop delay. We have 2.4% additional IR-drop delay in the critical paths as compared with the optimal path delay without considering IR-drop effect, while the previous work [9] has 3.4% additional IR-drop delay in the critical paths.

Chapter 1

Introduction

With the progress of fabrication processes and the growing complexity of chip design, delay-fault testing has become more and more important. The main reason comes from the fact that the defect characteristics of processes at 0.13-micron and below are timing-related.

In order to ensure that a circuit meets timing requirements, at-speed scan test is widely used to detect delay defects. For example, [1] reports that the defects per million (DPM) rates are reduced by 30 to 70 percent when at-speed testing is added to the traditional stuck-at tests. Moreover, [2] shows that the escape rate goes up nearly 3 percent if at-speed scan tests are removed from the test program with 0.18-micron feature size.

However, at-speed scan test suffers from the test-induced yield loss. Because the switching activity of whole circuit during test mode is much higher than that during normal mode, the large portion of gates simultaneously switching contributes to serious IR-drop delay. Thus, propagation delay does not meet the timing constraint only at test mode. This IR-drop problem during test mode exacerbates delay defects and results in false failures.

The analysis from [3] presents that IR-drop effect increases up to 16 percent during at-speed test compared to normal mode. In addition, both [4] and [5] addressed IR-drop issue and emphasize the importance of avoiding false delay test failures caused by IR-drop.

Recently many researchers have worked on generating test patterns to reduce IR-drop effect during at-speed testing so as to avoid false failures of delay test. The *preferred fill* proposed in [6] focuses on reducing the Hamming distance between flip-flips outputs. Wen et al. propose a series of work [7, 8, 9] by using *X-filling* technique to reduce the IR-drop effect. In [7], only flip-flops transition activity is considered while in [8], both flip-flops and gates transition activity are studied. In [9], the authors focus on the critical paths related activities so that the delay malfunction is reduced, and *ifill* proposed in [10] is to reduce both shift power and capture power during at-speed testing by improving the *X-filling* technique. Besides *X-filling* technique, in [11], a framework for test pattern generation is proposed. The process variation information, power grid topology and regional constraints on switching activity are taken into consideration to generate power-safe scan test patterns. The main idea of the above approaches except [11] utilizes switching probability of gates to predict the behavior of circuit and assign values to the unspecified bits accordingly in a test cube so that the fully-specified test vectors is expected to reduce IR-drop when applied in test mode. However, the target assigned value may not propagate through a long path. Therefore, the assignment may result in no effect on reducing IR-drop. In addition, except [11], the location information such as power grid topology and the adjacency of gates that switching at the same time are not taken into consideration.

Although the local switching activities is considered in [11], the extra test patterns have to be inserted and therefore increases test time.

The method of filling unspecified bits in test cube has the advantages of requiring only minimal changes to the existing ATPGs. In addition, it would not affect the test data volume and the test time. Therefore, in this thesis, we take the *X-filling* approach to reducing the IR-drop effect during at-speed test. The main difference between our approach and the previous *X-filling* methods lies in two aspects. The first one is that we take the spatial information into consideration in our approach. This is because the dynamic IR-drop occurs while a large number of gates switch simultaneously in the small region. Therefore, physical location should be taken into consideration to efficiently reduce the dynamic IR-drop. The second one is how *X-filling* is performed. In previous work [7, 8, 9], a forward-propagation approach is taken. Scan flip-flop is assigned value first. Then, the value is propagated to a target gate. In many cases, side-inputs along a path will block the propagation and thus target gate will not be assigned an expected value. Instead, we propose a backward-propagation approach. For a target gate, a target value is assigned to reduce IR-drop in the target region. Then the value is backward-propagated to scan flip-flop by solving *Pseudo Boolean* (PB) constraints [12]. Our method can assure that a target gate is assigned its expected value.

The rest of this thesis is organized as follows. Chapter 2 describes the at-speed testing model we used. Chapter 3 gives our motivation. Chapter 4 presents our main algorithm to reduce IR-drop by taking physical location of gates into consideration. Chapter 5 shows the simulation framework and

experimental results. Finally, Chapter 6 concludes this thesis.

Chapter 2

At-speed Testing Model

Before describing the motivation of our work, at-speed testing model is introduced first. In our work, we adopt the *launch-off-capture* schemes for at-speed scan-based delay test. Figure 2.1 shows an example waveform.

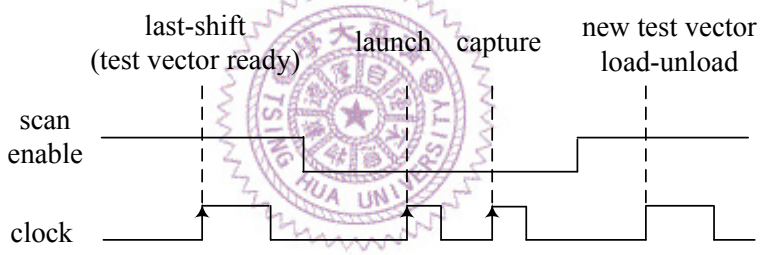


Figure 2.1: The waveform of launch-off-capture scheme.

In this approach, after the test vector is loaded into the scan chain (at *last-shift* pulse), two clock pulses are applied in capture mode. The first pulse *launches* the transition at target terminal, which is always the flip-flop output, and the second pulse *captures* the response from the flip-flop input at a scan cell. It can be regarded as two timing frames. The input vector of the first timing frame comes from the primary input and loaded into the

scan chain. Then, the output vector of the first timing frame become the input vector of the second timing frame as shown in Figure 2.2.

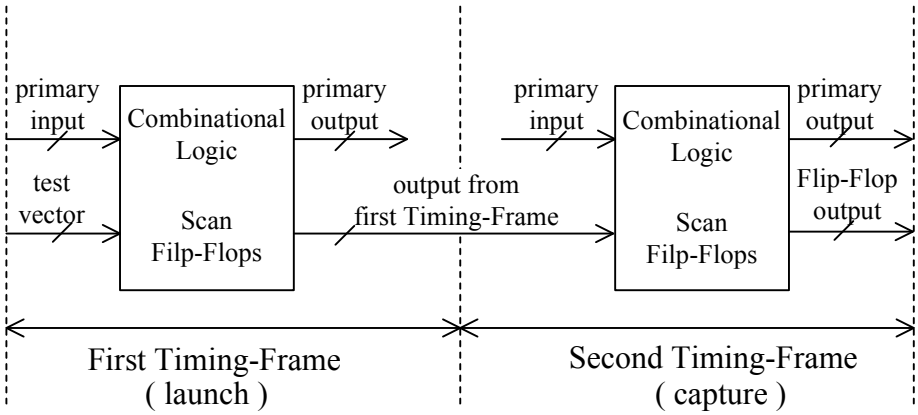


Figure 2.2: The timing frame form of launch-off-capture scheme.

Since the false delay path occurs between the launch pulse and the capture pulse, our goal is to minimize the impact of IR-drop effect during this test cycle. In other words, our goal is to minimize the switching activity of gates between the last-shift cycle (before launch cycle) and the launch cycle.

Chapter 3

Motivation

In this section, we use an example to show why a target gate may not be assigned a target value by using forward-propagation approach taken by [7, 8, 9].

Two approaches, *X-selection* and *value-selection* are used iteratively in [7, 8, 9]. *X-selection* is to select a specific X-value (specific flip-flop) to fill and *value-selection* is to assign value to the selected X based on some priority function calculated by switching probability to predict transitions.

Figure 3.1 shows an iteration of *X-selection* and *value-selection* proposed in [7, 8, 9]. This figure shows the circuit state at the last shift cycle and the launch cycle after applying a test vector $(P, FF\_1, FF\_2, FF\_3, FF\_4) = (1, X, X, X, 0)$, where X-bits represent unspecified bits.

Let G_2, G_3 and G_5 be *critical gates* defined in [9]. Critical gate means that G_2, G_3 and G_5 have high impact on the total IR-drop, and we assume that $G_5 > G_3 > G_2$ in their impact. Since G_2, G_3 and G_5 have the value *'X'* at the last shift cycle and *'1'* at launch cycle, in order to reduce the switching activity of G_2, G_3 and G_5, it is preferable to have the value of G_2,

G_3 and G_5 to be *'1'* at the last shift cycle.

First, *X-priority* values of all candidate *X-bits* including flip-flop *FF_1*, *FF_2* and *FF_3* are calculated. This value represents the priority order of these *X-bits* to be filled. For example, the *X-priority* value of *FF_1* reflects how likely the switching activity of G_2, G_3 and G_5 can be reduced by filling *FF_1*. In this example, *FF_2* has highest *X-priority* and is selected as the first unspecified bit to be filled.

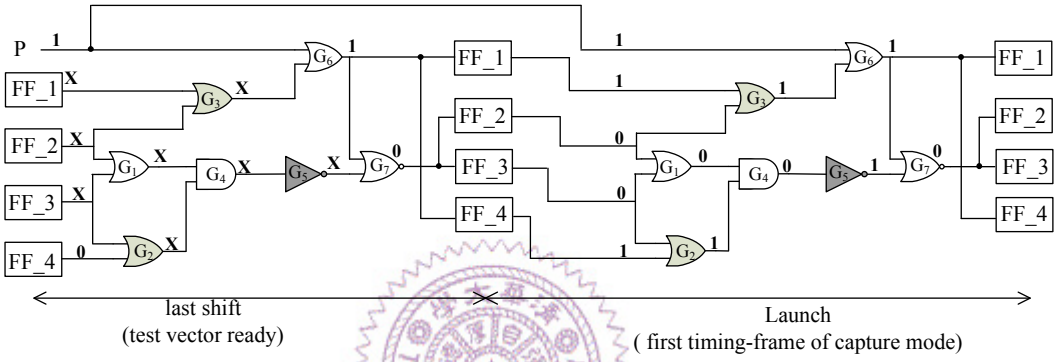


Figure 3.1: The example circuit with test cube.

After the target unspecified bit *FF_2* is selected, the value *'0'* and *'1'* are applied to *FF_2*. The value that can contribute lower transition of G_2, G_3 and G_5 by probability calculation is selected. In this example, value *'1'* is assigned to *FF_2*, and the corresponding circuit state at the last shift cycle is shown in Figure 3.2.

As shown in Figure 3.2, the value of *FF_2* is blocked at G_4 since *'1'* is not a controlling value of *AND* gate. Therefore, no switching of G_5 is not guaranteed after this assignment.

In the next iteration, G_2 and G_5 are the remaining *critical gates*. To

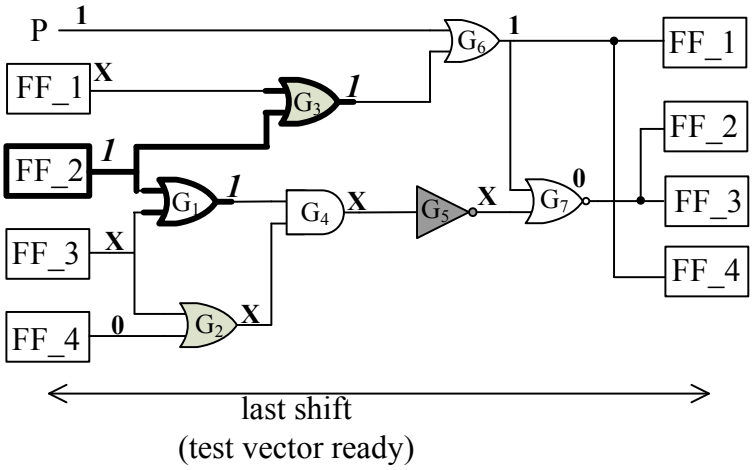


Figure 3.2: The example circuit after apply value '1' to *FF_2*.

reduce the switching of G_2 and G_5, *FF_3* is assigned value *'1'*. Figure 3.3 shows the circuit state after the second iteration of *X-selection* and *Value-selection* is applied. Now, both G_2 and G_3 are set to be *'1'*, and G_5 has the value *'0'* after implication.

In this case, we notice that G_2 and G_3 have no transition, but G_5 has to switch at the last shift cycle and the launch cycle. However, G_5 has highest impact on IR-drop, and to prevent G_5 from switching should have higher priority than G_2 and G_3. The reason why the assignment fails to reduce transition of G_5 is that *X-priority* is measured by using probability to predict all possible transitions. If a *critical gate* G is far away from an input and there are many unspecified X-bits, it is nearly impossible to make a transition path from an input to G. Therefore, the value of G may be set to unexpected value during the iterations.

At the beginning of *X-filling* process, the number of unspecified bits is

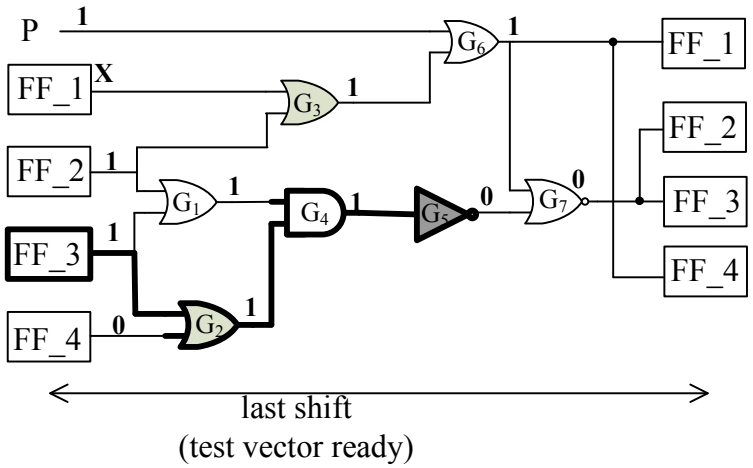


Figure 3.3: The example circuit after apply value '1' to *FF_3*.

about 80 to 90 percent of the total number of test cube. Hence, the value assigned to test vector will be always blocked and cannot propagate through a long path to a target gate. According to our experiment, *X-filling* step [7, 8, 9] at the beginning of the whole iteration process is similar to random fill, and the filled value hardly propagates to *critical gate* to reduce transition.

Based on the observation above, we propose a backward-propagation approach. First, an appropriate value for a target gate is selected to minimize transition. Then, the value is backward propagated to primary input or flip-flop. Figure 3.4 illustrates an example.

Let us consider the same example shown in Figure 3.1 and the value of G_5 be assigned *'1'* to reduce its switching. Then, the value is first backward propagated to G_4. Since G_4 is set to *'0'*, the value of G_1 or G_2 is set to *'0'*. Suppose G_2 is set to *'0'*, the value of *FF_3* must be set to *'0'*. Next, the value of G_3 is assigned *'1'* to reduce its switching. Hence, the value of *FF_2*

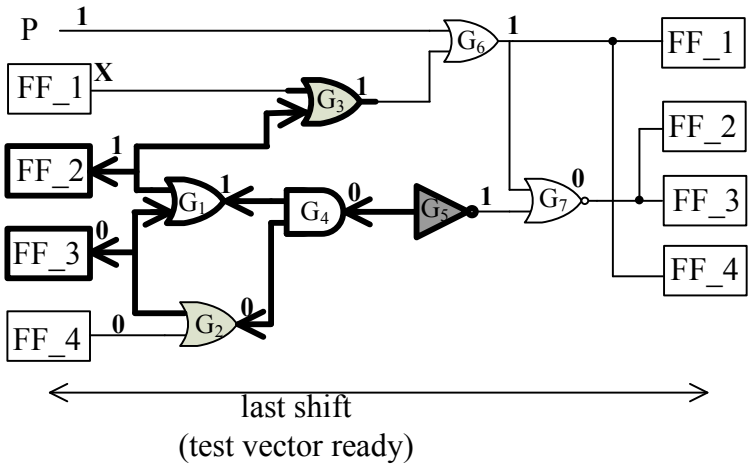


Figure 3.4: The example circuit after apply backward propagation.

is set to *'1'*. After implication process, G_1 is set to '1'. In this backward propagation process, more than one unspecified bits in test cube can be set in one iteration, and G_3 and G_5 are assured to have an output *'1'*.

Chapter 4

The Proposed Method

4.1 The Power Grid Architecture

The power/ground distribution network proposed in [3] is adopted. Figure 4.1 shows the power grid architecture. The power rings are created to carry power around the core chip area. Four VDD and VSS pads are inserted to the respective rings. The stripes directly connected to the power/ground ring and the rails draw power/ground from the nearest power stripe. By using stripes and rails, the power and ground is routed to the standard cells.

4.2 Overview of Our Proposed Method

Figure 4.2 illustrates the overall algorithm flow. First, a given circuit netlist is placed and routed. After placement and routing, physical information of the circuit is utilized in the following steps. Next, *region_partition* is performed. Since the power grid topology is taken into consideration, the chip is first divided into several regions according to the physical layout. Each cell in the design is then assigned to a region by using its physical co-ordinates. Based

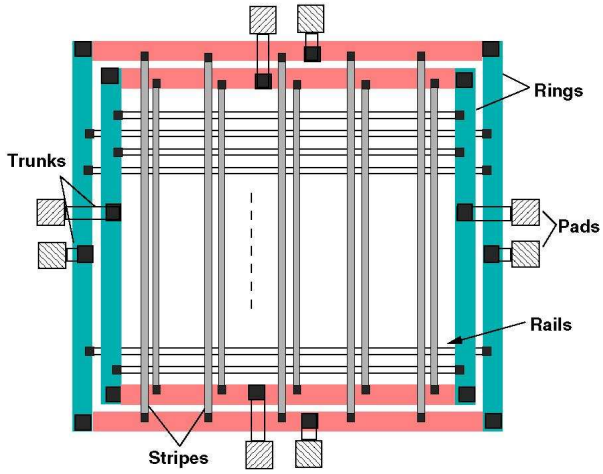


Figure 4.1: The power/ground distribution network.

on the characteristic of the power grid architecture described in Section 4.1, each region is bounded by the stripes and limited number of rails. If a large number of cells switch simultaneously within the same region, the IR-drop effect in this region is regarded as very serious and may increase the propagation delay.

Our main idea is to minimize switch activities of regions. In order to achieve this goal, appropriate values (0 or 1) are assigned to unspecified bits within each region. The next step is *target_region_selection*. With a set of given critical paths and the test cubes with unspecified bits, the switching activity of each region is first calculated. The region with highest switching activities is selected as *target region* for assigning values to unspecified bits. Finally, in the *target region*, *target bit* is selected and its value is set one by one. In the selection of *target bit*, sub-circuit in *target region* is modeled as a *Psuedo Boolean* constraint problem (PB-problem). The solution to the

PB-problem is used to assist us to select *target bit* and its value. After each selection, the value of *target bit* is propagated to primary input and primary output. The selection continues till the switching activity is smaller than a threshold value. Once a *target region* is processed completely, the switch activities of all regions are re-calculated. The procedure repeats till no more region to be processed.

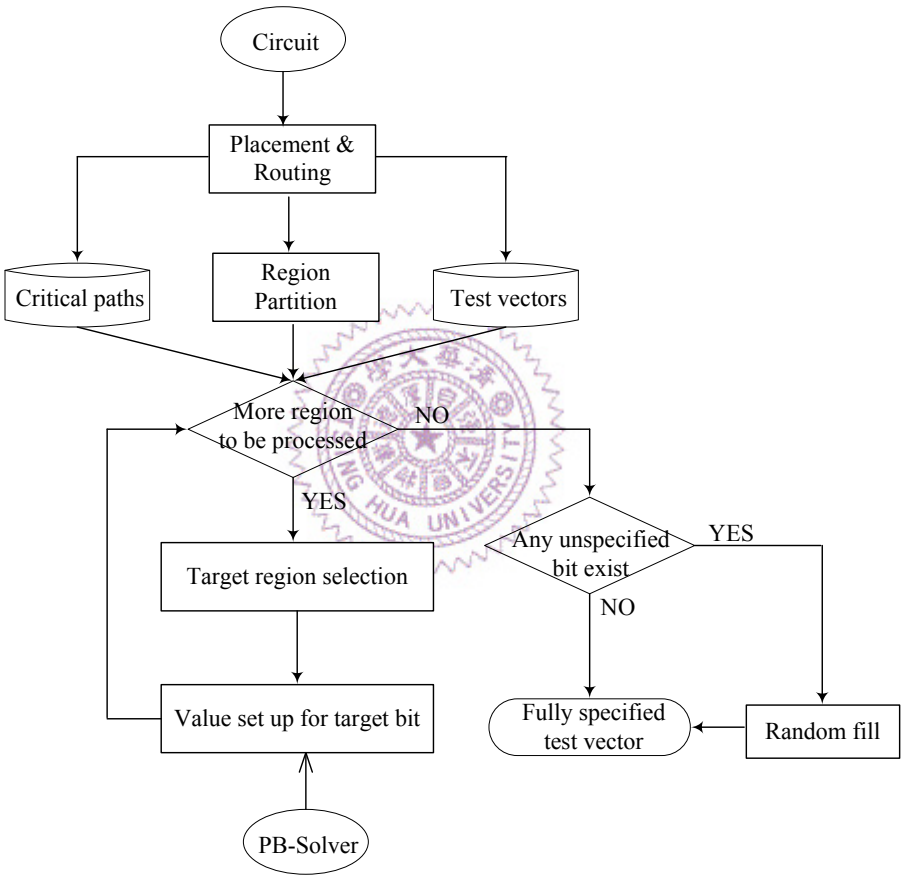


Figure 4.2: The overall flow of the proposed method.

The details of *target_region_selection* and the *value_set_up_for_target_bit*

steps are described in the following sections.

4.3 Target Region Selection

In this subsection, we propose a cost function to guide the selection of a region with serious IR-drop effect. We first apply test vectors to the circuit to observe the switching activities of each gate during at-speed test cycle. Our cost function, *weighted switching activities* (WSA), is defined to represent the IR-drop impact of each region caused by the test vectors. Give a region, *region j*, its $WSA_{regionj}$ is defined as

$$WSA_{region\ j} = \sum_{gate\ i}^{\forall gate\ i \in\ region\ j} (toggle(type_i) \times switching\ weight_i \times \sum_{k \in fanout\ of\ gate\ i} capacitance_k) \tag{4.1}$$

where $type_i$ is the type of toggles of *gate i*, and $toggle(type_i)$ represents the toggle count of $type_i$. Here, the toggle type includes all kinds of transitions, i.e. 1→0, 0→1, X→0, X→1, 1→X, 0→X and X→X. Type 4 has toggle count '0', and all other types '1' as shown in Table 4.1. For each type of toggle, we give its *switching weight* as shown in the fourth column of Table 4.1, where type 1 has the highest weight and type 4 has zero weight. The last term of WSA is to calculate the fanout capacitance of *gate i*.

WSA is defined to estimate the IR-drop of each region because it can represent the supply current demand and the power consumption. Based on the cost function, the region with the highest WSA is selected as *target region* and is the next region to be processed.

Table 4.1: The switching weight of every kind of transition

type	transition	toggle	switching weight
1	0 → 1 1 → 0	1	2
2	0 → X 1 → X X → 0 X → 1	1	1.5
3	X → X	1	1
4	0 → 0 1 → 1	0	0

4.4 Value Set Up for Target Bit

After the *target region* is selected, the *X-filling* technique will be applied to reduce IR-drop in this *target region*. Since values assigned in the region are outputs of internal gates, it is required to justify this assignment in the primary input. In addition, the correlations of internal gates between the last shift cycle and the launch cycle has to be satisfied. Therefore, we model this assignment problem as a *Pseudo Boolean* (PB) constraint problem and solve it by a PB-solver.

The PB-constraint are linear inequalities, and its normal form is expressed as:

$$C_1X_1 + C_2X_2 + ...C_{n-1}X_{n-1} \geq C_n \tag{4.2}$$

where $C_i \in Z$, and $X_i \in \{1, 0\}$. PB-constraint problem can be regarded as a specialized ILP (integer linear programming) problem where variables are

restricted to be boolean value, $X_i \in \{1, 0\}$. Moreover, if all coefficients C_i in equation (4.2) are 1, then the PB-constraint is equivalent to CNF clause in SAT problem. For example, a CNF clause $(X_1 \vee X_2)$ is equivalent to a constraint, $X_1 + X_2 \geq 1$ in PB formulation. The difference between SAT problem and PB problem is that PB problem has an objective function.

Since a circuit can be represented easily by CNF formulas, in our method, we use PB-constraint to express the correlation between internal gates at two timing frames. In addition, an objective function is defined to minimize IR-drop effect within a *target region*.

Before we describe the modeling, some variables are defined first. Given a sub-circuit with n gates and flip-flops, where $\{GF_i \mid i = 1, 2, ..., n\}$ represents a gate or flip-flop in the sub-circuit. The variables are:

S_gfi is the value of GF_i at the last shift cycle.

L_gfi is the value of GF_i at the launch cycle.

T_GF_i is the toggle of GF_i and defined as

$$T\_GF_i = \begin{cases} 1, & \text{if } S\_gfi \neq L\_gfi \\ 0, & \text{if } S\_gfi = L\_gfi \end{cases}$$

C_GF_i represents if GF_i is on critical paths and defined as

$$C\_GF_i = \begin{cases} 1, & \text{if } GF_i \text{ is on critical paths} \\ 0, & \text{otherwise.} \end{cases}$$

Since our goal is to reduce the total WSA within the *target region*, the *objective function* to be minimized is defined as following:

$$objective\ function = \sum_{i=1}^{n} (Impact_i \cdot T\_GF_i) \tag{4.3}$$

where $Impact_i$ represents how serious IR-drop can be affected if GF_i toggles and $Impact_i$ is defined as

$$Impact_i = (1+\alpha \cdot C\_GF_i) \times \sum_{K \in fanout\ of\ GF_i} capacitance_K \qquad 0 \leq \alpha \leq 1 \quad (4.4)$$

where α is the weight used to emphasize the importance of T_GF_i if GF_i is on the critical path. Since gates on critical paths switching simultaneously would cause serious IR-drop delay and result in delay defects, it is preferable that gates on critical paths will not switch. Similarly, gate with large loading will induce larger current flow, it has more impact on IR-drop.

Let us demonstrate the modeling of the objective function using circuit shown in Figure 4.3. Let fanout number represents fanout capacitance and the path, $FF\_2 \rightarrow G_1 \rightarrow G_4 \rightarrow G_5 \rightarrow G_7 \rightarrow FF\_3$, is a critical path in this example.

The objective function for this circuit is to minimize:
$((1+\alpha)\times 1)\cdot T\_G_1 + (1\times 1)\cdot T\_G_2 + (1\times 1)\cdot T\_G_3 + ((1+\alpha)\times 1)\cdot T\_G_4 +$
$((1+\alpha)\times 1)\cdot T\_G_5 + (1\times 3)\cdot T\_G_6 + (1\times 2)\cdot T\_G_7 + (1\times 1)\cdot T\_FF_1 +$
$((1+\alpha)\times 2)\cdot T\_FF_2 + ((1+\alpha)\times 2)\cdot T\_FF_3 + (1\times 1)\cdot T\_FF_4$

Next, three kinds of constraints are set to model the switching of circuit. They are *output constraint*, *consistency constraint* and *value constraint*.

- *Output constraint*: This constraint is used to represent the relationship between the switching activity and the value between the last shift cycle and the launch cycle. Therefore, it is modeled as XOR S_gfi and L_gfi.

- *Consistency constraint*: This constraint is used to maintain the function consistency at both last shift cycle and launch cycle.

- *Value constraint*: This constraint is used to indicate the circuit state after the test cube is applied to the circuit in the *target region*.

We use the same circuit shown in Figure 4.3 to demonstrate how these constraints are modeled:

Output constraint:

$T\_G1 = \text{XOR}(S\_g1, L\_g1)$ $T\_G2 = \text{XOR}(S\_g2, L\_g2)$

$T\_G3 = \text{XOR}(S\_g3, L\_g3)$ $T\_G4 = \text{XOR}(S\_g4, L\_g4)$

$T\_G5 = \text{XOR}(S\_g5, L\_g5)$ $T\_G6 = \text{XOR}(S\_g6, L\_g6)$

$T\_G7 = \text{XOR}(S\_g7, L\_g7)$

$T\_FF1 = \text{XOR}(S\_ff1, L\_ff1)$ $T\_FF2 = \text{XOR}(S\_ff2, L\_ff2)$

$T\_FF3 = \text{XOR}(S\_ff3, L\_ff3)$ $T\_FF4 = \text{XOR}(S\_ff4, L\_ff4)$

Consistency constraint:

$S\_g1 = \text{OR}(S\_ff2, S\_ff3)$ $S\_g2 = \text{OR}(S\_ff3, S\_ff4)$

$S\_g3 = \text{OR}(S\_ff1, S\_ff2)$ $S\_g4 = \text{AND}(S\_g1, S\_g2)$

$S\_g5 = \text{INV}(S\_g4)$ $S\_g6 = \text{OR}(P, S\_g3)$ $S\_g7 = \text{NOR}(S\_g5, S\_g6)$

$L\_g1 = \text{OR}(L\_ff2, L\_ff3)$ $L\_g2 = \text{OR}(L\_ff3, L\_ff4)$

$L\_g3 = \text{OR}(L\_ff1, L\_ff2)$ $L\_g4 = \text{AND}(L\_g1, L\_g2)$

$L\_g5 = \text{INV}(L\_g4)$ $L\_g6 = \text{OR}(P, L\_g3)$ $L\_g7 = \text{NOR}(L\_g5, L\_g6)$

Value constraint:

$P = 1$ $S\_ff4 = 0$ $S\_g6 = 1$ $S\_g7 = 0$

$L\_ff1 = 1$ $L\_ff2 = 0$ $L\_ff3 = 0$ $L\_ff4 = 1$

$L\_g1 = 0$ $L\_g2 = 1$ $L\_g3 = 1$ $L\_g4 = 0$

$L\_g5 = 1 \qquad L\_g6 = 1 \qquad L\_g7 = 0$

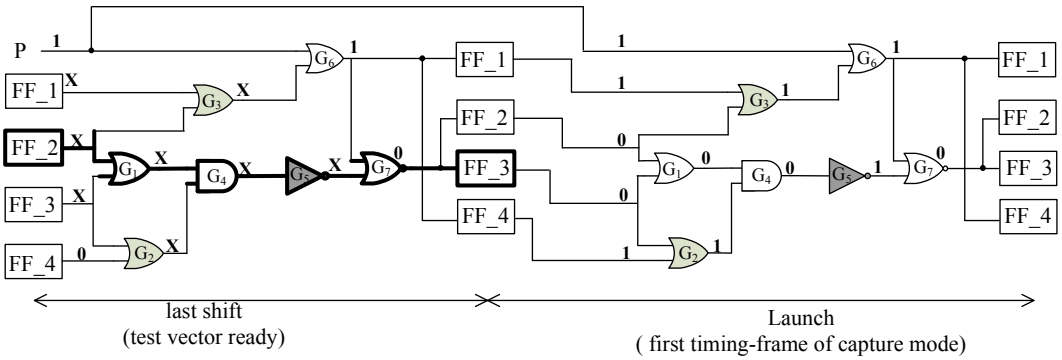


Figure 4.3: The example circuit with test cube.

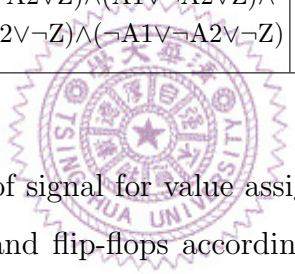
Finally, these constraints mentioned above are translated to the corresponding PB-constraint expression. Table 4.2 shows the type of gates we used and the corresponding CNF expression and PB-constraint. With the objective function, a PB-solver [12, 13] is called to solve the problem.

After we solve the above modeled PB problem, we obtained a set of value assignments of the gates in the *target region*. This assignment is to minimize the object function and, in terms, to minimize IR-drop in the region.

Nevertheless, we will not use all the assignments. Instead, we will just select a set of assignments and let un-selected gates be un-set. The reasons are twofold. First, more assignments are set in this region, less signals in other regions can be set after signals implication. It may result in the case where the IR-drop in *target regions* processed at the beginning of the iterations are maximally reduced and the rest of regions remains the same due to no unspecified bit. Second, a region with a controllable IR-drop is acceptable.

Table 4.2: The CNF expression of gates

Gate type	Input	Output	CNF expression	PB constraint
BUF	A	Z	$(A \vee \neg Z) \wedge (\neg A \vee Z)$	$(A+\overline{Z} \geq 1)$, $(\overline{A}+Z \geq 1)$, $(A+\overline{A} = 1)$, $(Z+\overline{Z} = 1)$
INV	A	Z	$(A \vee Z) \wedge (\neg A \vee \neg Z)$	$(A+Z \geq 1)$, $(\overline{A}+\overline{Z} \geq 1)$, $(A+\overline{A} = 1)$, $(Z+\overline{Z} = 1)$
NAND	A1,A2	Z	$(A1 \vee Z) \wedge (A2 \vee Z) \wedge (\neg A1 \vee \neg A2 \vee \neg Z)$	$(A1+Z \geq 1)$, $(A2+Z \geq 1)$, $(\overline{A1}+\overline{A2}+\overline{Z} \geq 1)$ $(A1+\overline{A1} = 1)$, $(A2+\overline{A2} = 1)$, $(Z+\overline{Z} = 1)$
NOR	A1,A2	Z	$(\neg A1 \vee \neg Z) \wedge (\neg A2 \vee \neg Z) \wedge (A1 \vee A2 \vee Z)$	$(\overline{A1}+\overline{Z} \geq 1)$, $(\overline{A2}+\overline{Z} \geq 1)$, $(A1+A2+Z \geq 1)$ $(A1+\overline{A1} = 1)$, $(A2+\overline{A2} = 1)$, $(Z+\overline{Z} = 1)$
AND	A1,A2	Z	$(A1 \vee \neg Z) \wedge (A2 \vee \neg Z) \wedge (\neg A1 \vee \neg A2 \vee Z)$	$(A1+\overline{Z} \geq 1)$, $(A2+\overline{Z} \geq 1)$, $(\overline{A1}+\overline{A2}+Z \geq 1)$ $(A1+\overline{A1} = 1)$, $(A2+\overline{A2} = 1)$, $(Z+\overline{Z} = 1)$
OR	A1,A2	Z	$(\neg A1 \vee Z) \wedge (\neg A2 \vee Z) \wedge (A1 \vee A2 \vee \neg Z)$	$(\overline{A1}+Z \geq 1)$, $(\overline{A2}+Z \geq 1)$, $(A1+A2+\overline{Z} \geq 1)$ $(A1+\overline{A1} = 1)$, $(A2+\overline{A2} = 1)$, $(Z+\overline{Z} = 1)$
XOR	A1,A2	Z	$(\neg A1 \vee A2 \vee Z) \wedge (A1 \vee \neg A2 \vee Z) \wedge (A1 \vee A2 \vee \neg Z) \wedge (\neg A1 \vee \neg A2 \vee \neg Z)$	$(\overline{A1}+A2+Z \geq 1)$, $(A1+\overline{A2}+Z \geq 1)$ $(A1+A2+\overline{Z} \geq 1)$, $(\overline{A1}+\overline{A2}+\overline{Z} \geq 1)$ $(A1+\overline{A1} = 1)$, $(A2+\overline{A2} = 1)$, $(Z+\overline{Z} = 1)$



Therefore, the selection of signal for value assignment is proceed as follows. First, we sort all gates and flip-flops according to their *Impact* defined in equation (4.4). We select the next gate or flip-flop with the highest impact for process. If its assignment toggles from last shift cycle to launch cycle, this gate is not selected and the values are not assigned. If the assignment cause no toggle, the gate is selected as *target bit* and the value *target value*. Next the selected value is now propagated backward and forward to justify this assignment. If conflict occurs during the propagation, then the assignment to this *target bit* has to be dropped. The selection of *target bit* and *target value* continues until the IR-drop is smaller than a threshold value or no more gate to be selected. Figure 4.4 shows the flow of *Value_Set_Up_for_Target_Bit*.

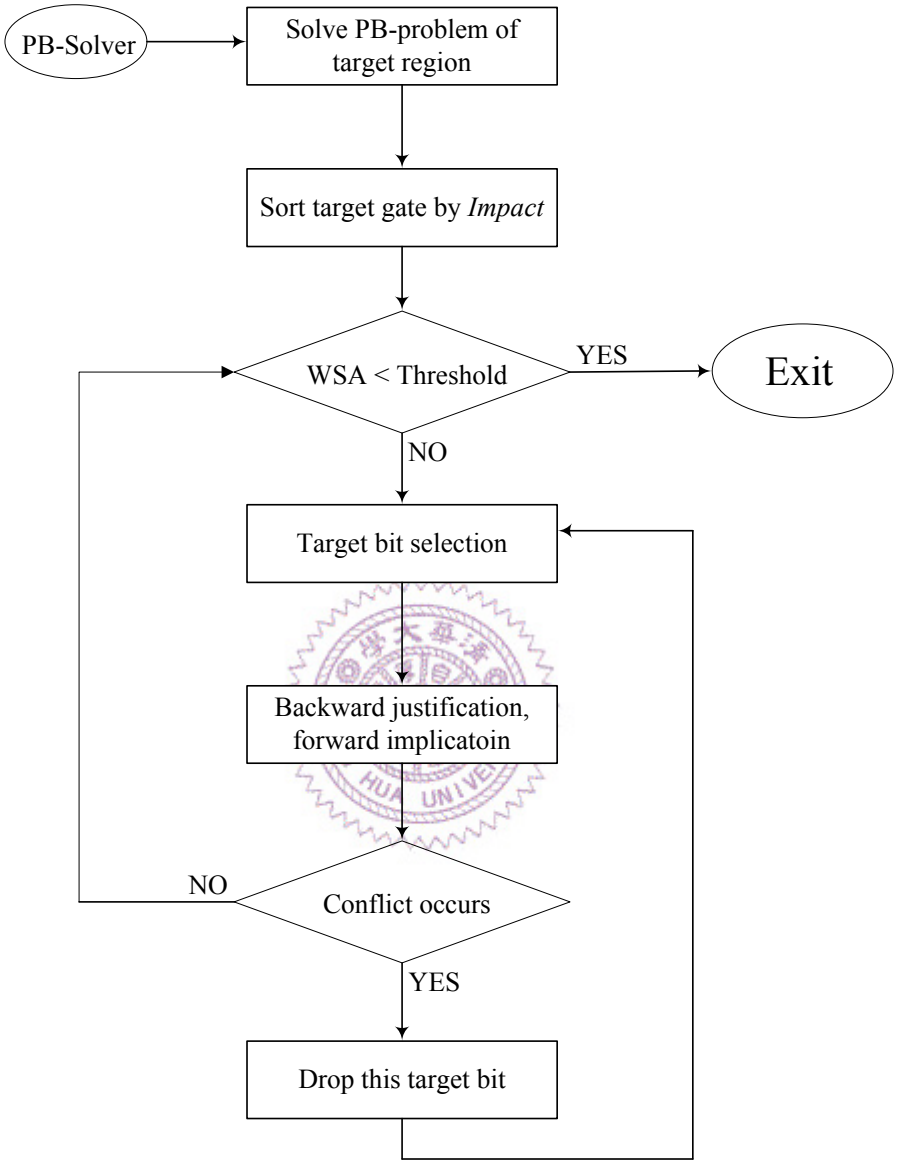


Figure 4.4: The overall flow of *Value_Set_Up_for_Target_Bit*.

Chapter 5

Experimental Result

To demonstrate the effectiveness and efficiency of our method, experiments are performed. Figure 5.1 illustrates our experimental flow.

The experiments are performed on ITC'99 benchmark [14]. The circuits are described in VHDL and synthesized by Synopsys Design Compiler with TSMC 90nm cell library. Six kinds of gates including *BUFFER*, *INVERTER*, *NAND*, *NOR*, *AND* and *OR* are used for synthesis. The gate-level netlist and the test protocol in STIL format are generated. The netlist is input to SoC Encounter (SOCE) and the output netlist of layout design with detailed physical location and the timing information after floorplaning, placement, and routing is produced by SOCE. Then the netlist and the test protocol are fed into TetraMax to generate test patterns with unspecified bits. Meanwhile, the timing information is used as input file for PrimeTime to produce critical paths information. Our algorithm takes the netlist of layout design, the test pattern with unspecified bits and the critical paths information as input files, and generates IR-drop-aware fully specified test patterns by utilizing MINISAT+ [13] as the PB-solver.

To produce precise IR-drop analysis by using RedHawk, the VCD (value change dump) format file which records the accurate transition is generated by simulating the fully specified test patterns. Then, RedHawk reports IR-drop information according to the VCD file and layout netlist. In addition, the modified timing information by taking IR-drop effect into consideration is also reported. Therefore, IR-drop-aware critical paths and path delay are generated by PrimeTime.

Table 5.1 shows the information of benchmark circuits and the generated test cube. The circuits are listed in the first column. The columns labeled *#PI*, *#FF*, and *#Gate* represent the number of primary input, the number of scan-chain flip-flop and the total number of gate count, respectively. The number of vertical stripes and the number of rows used to divide a layout into regions are listed in columns 5 and 6. Column 7 labeled as *#Region* is the number of regions in each design. The column labeled *Fault Coverage (%)* represents the fault coverage by running 100 test patterns and the last column labeled as *X-bits (%)* is the *X-bit* ratio.

Table 5.1: Descriptions of circuits and test cube

	Circuit Information						Test Cube Information	
circuit	#PI	#FF	#Gate	#Stripe	#Row	#Region	Fault (%) coverage.	X-bits (%)
b11	11	31	824	3	16	35	92	62
b12	9	121	935	3	16	41	95	79
b14	36	215	20328	3	99	202	76	54
b15	40	417	10289	3	115	147	80	77
b17	42	1317	30537	5	150	324	76	76
b21	36	430	25619	5	59	280	69	52

Table 5.2 shows the voltage drop reported by RedHawk. We compare our PB-based X-filling method to random fill, 0-fill, and CPA-based X-filling [9]. The results are labeled as *r-fill*, *0-fill*, *CPA-based*, and *PB-based*, respectively. The results of *r-fill* is our baseline result reference. The columns labeled *(%)* are the reduction ratio as compared to *r-fill* method and the row labeled *Imp(%)* shows the average reduction ratio. The columns labeled *MAX* and *Avg.* list the maximum voltage drop and the average voltage drop of each benchmark circuit.

Table 5.2: Comparisons of maximum and average voltage drop

circuit	r-fill		0-fill				CPA-based				PB-based			
	MAX	Avg.	MAX		Avg.		MAX		Avg.		MAX		Avg.	
	(mV)	(mV)	(mV)	(%)	(mV)	(%)	(mV)	(%)	(mV)	(%)	(mV)	(%)	(mV)	(%)
b11	15.4	1.7	6.9	55	1.0	41	10.5	32	1.0	41	5.0	68	0.9	47
b12	17.1	1.8	9.0	47	0.7	61	9.0	47	0.9	50	9.0	47	0.6	67
b14	280.5	115.5	270.2	4	64.3	44	273.1	3	96.2	17	275.7	2	66.9	42
b15	149.4	25.9	124.3	17	10.9	58	110.9	26	17.4	33	59.6	60	9.9	62
b17	209.3	53.2	77.5	63	16.6	69	122.3	42	30.3	43	77.5	63	14.8	72
b21	114.6	21.8	80.4	30	19.2	12	86.5	25	21.3	2	65.9	42	20.7	5
Imp(%)				36		48		29		31		47		49

Table 5.3 shows the results of worst IR-drop reduction. The numbers list in this table are reported by RedHawk. The cells with their worst IR-drop larger than 6%, 5% and 4% are reported. The columns labeled # are the number of cells, and the columns labeled *(%)* are the reduction ratio as compared to r-fill method and the row labeled *Imp(%)* shows the average reduction ratio. b11 and b12 will not be compared since they are small

circuits so that they have no cells with IR-drop larger than 4%.

Table 5.3: The number of cells with worst IR-drop

circuit	r-fill			0-fill						CPA-based						PB-based					
	6%	5%	4%	6%		5%		4%		6%		5%		4%		6%		5%		4%	
	#	#	#	#	(%)	#	(%)	#	(%)	#	(%)	#	(%)	#	(%)	#	(%)	#	(%)	#	(%)
b14	15481	15757	16149	12598	19	13098	17	13539	16	16645	-8	16869	-7	17080	-6	13631	12	14286	9	14848	8
b15	4563	4626	4743	2	100	5	100	37	99	866	81	2937	37	3403	28	0	100	0	100	12	100
b17	10864	11675	12386	6	100	19	100	162	99	5864	46	6407	45	7696	38	5	100	17	100	140	99
b21	256	422	1985	3	99	76	82	1056	47	103	60	265	37	1582	20	0	100	67	84	1389	30
Imp(%)					80		75		65		45		28		20		78		73		59

To study how much the maximum path delay can be improved by reducing IR-drop effect, the optimal path delay without considering IR-drop effect is first reported by PrimeTime and compared with those by random fill, 0-fill, CPA-based X-filling, and PB-based X-filling methods. Table 5.4 shows the results. The columns labeled *delay* list the maximum delay of each design, and the columns with label *r(%)* present the increased path delay ratio as compared to the optimal path delay. The row labeled *Inc(%)* shows the average ratio of the increased path delay.

Table 5.4: Comparisons of maximum path delay

circuit	optimal	r-fill		0-fill		CPA-based		PB-based	
	delay(ns)	delay(ns)	r(%)	delay(ns)	r(%)	delay(ns)	r(%)	delay(ns)	r(%)
b11	4.07	4.08	0.2	4.08	0.2	4.08	0.2	4.08	0.2
b12	3.17	3.17	0.0	3.17	0.0	3.17	0.0	3.17	0.0
b14	3.06	3.53	15.4	3.26	6.5	3.37	10.1	3.27	6.9
b15	6.19	6.37	2.9	6.32	2.1	6.35	2.6	6.30	1.8
b17	9.69	10.49	8.3	9.92	2.4	10.07	3.9	9.89	2.1
b21	10.13	10.62	4.8	10.40	2.7	10.46	3.3	10.46	3.3
Inc(%)			5.27		2.32		3.35		2.38

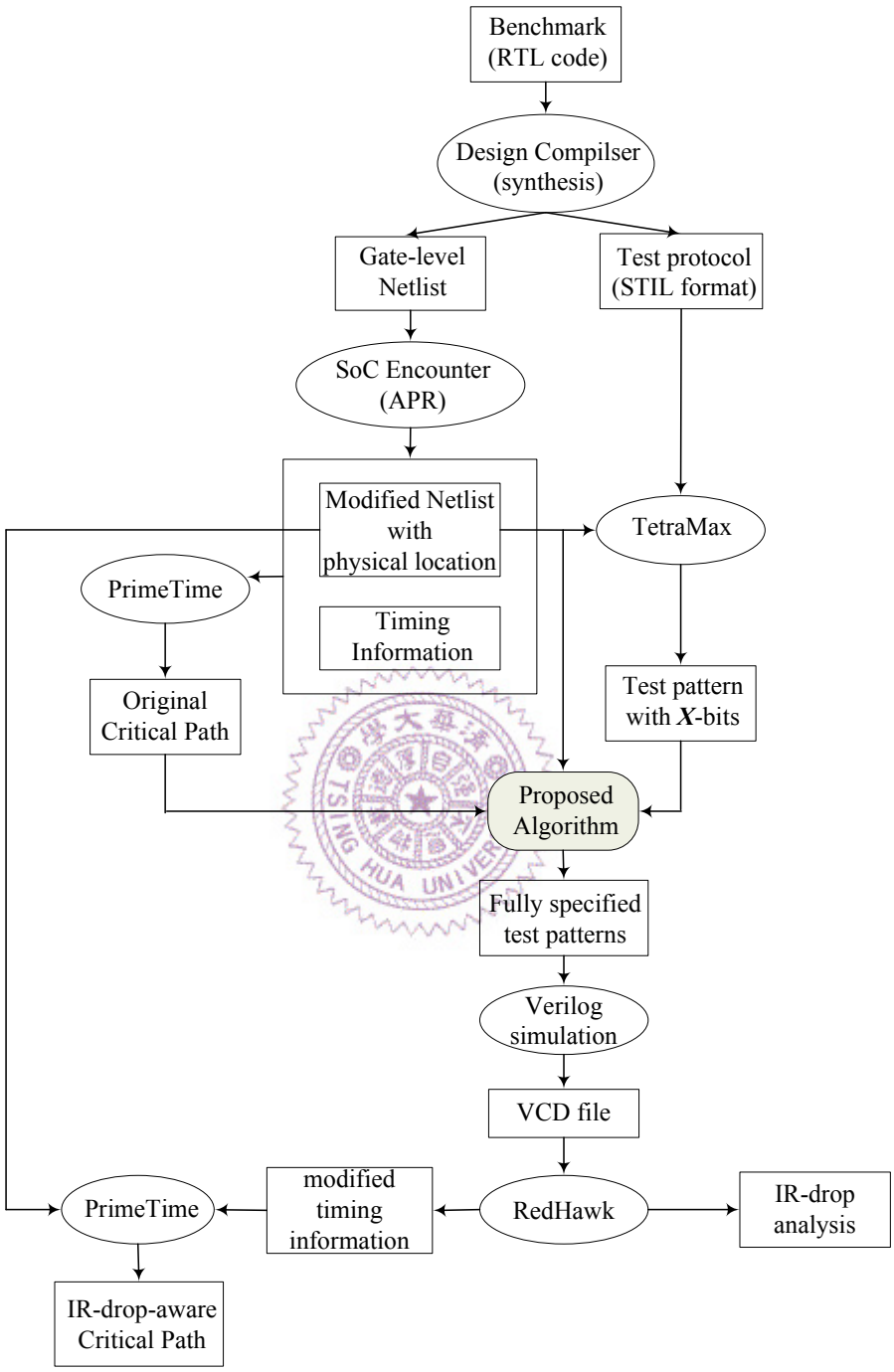


Figure 5.1: The overall flow of the experiment.

Chapter 6

Conclusions

In this thesis, we have proposed an *X-filling* method to reduce the IR-drop effect during at-speed scan test. First, the circuit was divided into several regions according to the physical layout. Second, we estimated the IR-drop effect for each region. Third, *X-filling* method was used to reduce the switching activity of the region. The procedure repeated until the IR-drop effect of all regions is below an user-specified threshold. Compared with the previous work [9], the experimental result has showed that in the worst and average IR-drop, we have 26% and 28% reduction, respectively. The IR-drop reduction also improved the IR-drop delay. We have 2.4% additional IR-drop delay in the critical paths as compared with the optimal path delay without considering IR-drop effect, while the previous work [9] has 3.4% additional IR-drop delay in the critical paths.

Bibliography

[1] B. Swanson and M. Lange, "At-speed testing made easy," *EE Times*, 3, Jun. 2004.

[2] J. Gatej, L. Song, C. Pyron, R. Raina, and T. Munns, "Evaluating ATE features in terms of test escape rates and other cost of test culprits," *Proceedings of the International Test Conference*, Oct. 2002, pp. 1040-1048.

[3] Nisar Ahmed, Mohammad Tehranipoor, Vinay Jayaram, "A novel framework for faster-than-at-speed delay test considering IR-drop effects," *ICCAD*, Nov, 2006, pp.198-203

[4] J. Saxena, K. M. Butler, V. B. Jayaram, et.al., "A Case Study of IR-drop in Structured At-Speed Testing", *Proc ITC 2003*, pp. 1098 V 1104.

[5] Srivaths Ravi, "Power-aware Test: Challenges and Solutions," *Proc ITC 2007*

[6] Santiago Remersaro, Xijiang Lin, Zhuo Zhang, Sudhakar M. Reddy, Irith Pomeranz and Janusz Rajski, "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs,"*Proc ITC 2006*

[7] X. Wen, Y. Yamashita, S. Morishima, S. Kajihara, L.-T. Wang, K. K. Saluja, and K. Kinoshita, "Low-Capture-Power Test Generation for Scan-Based At- Speed Testing," *Proc. ITC* Paper 39.2, 2005

[8] X. Wen, K. Miyase, T. Suzuki, Y. Yamato, S. Kajihara, L.-T. Wang, and K. K. Saluja, "A Highly-Guided X-Filling Method for Effective Low-Capture-Power Scan Test Generation," *Proc. ICCD*, pp. 251-258, 2006.

[9] X. Wen, Kohei Miyase, Tatsuya Suzuki, Seiji Kajihara, Yuji Ohsumi, and Kewal K. Saluja, "Critical-Path-Aware X-Filling for Effective IR-Drop Reduction in At-Speed Scan Testing," *DAC 2007*.

[10] Jia Li, Q. XU, Y. XU and X. Li, "iFill: an Impact-Oriented X-filling Method for shift-and capture-Power Reduction in At-Speed Scan-Based Testing,"*Proc DATE 2008*.

[11] V.R. Devanathan, C.P. Ravikumar and V. Kamakoti, "A Stochastic Pattern Generation and Optimization Framework for Variation-Tolerant, Power-Safe Scan Test,"*Proc ITC 2007*.

[12] Fadi A. Aloul, Arathi Ramani, Igor Markov, Karem Sakallah. "Generic ILP versus Specialized 0-1 ILP: an Update". *ICCAD*, November, 2002

[13] N. Een and N. Sorensson, "Translating Pseudo-Boolean Constraints into SAT," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 2, pp. 1-26, Mar. 2006.

[14] "http://www.cerc.utexas.edu/itc99-benchmarks/bench.html"