

行政院國家科學委員會專題研究計畫 期末報告

結合可調整延遲時間緩衝器及兼顧負載平衡與最小時序偏移之時脈樹合成方法應用於動態式多重供應電壓佈局環境
(II)

計畫類別：個別型
計畫編號：NSC 101-2221-E-343-004-
執行期間：101年08月01日至102年07月31日
執行單位：南華大學資訊工程學系

計畫主持人：蔡加春

計畫參與人員：碩士班研究生-兼任助理人員：范家禎
大專生-兼任助理人員：楊仁傑
大專生-兼任助理人員：賴宏輝
大專生-兼任助理人員：歐庭好
大專生-兼任助理人員：張晏翔

報告附件：出席國際會議研究心得報告及發表論文

公開資訊：本計畫涉及專利或其他智慧財產權，1年後可公開查詢

中華民國 102年08月27日

中文摘要：對目前的奈米積體電路設計來說，功率消耗是個很重要的影響。對於這個問題，動態式的多重供應電壓(multiple dynamic supply voltage, MDSV)設計方法是個有效的解決方法。然而，在功率模式切換的時候，會增加時序差異的改變，進而增加修正時序差異的困難度。在此，我們提出了一個負載平衡式的時脈樹合成器，並加入可調整延遲時間的緩衝器(adjustable delay buffer, ADB)，讓使用 MDSV 設計方法的晶片能有最小的時序差異。一旦初步的時脈樹已建立完成，我們會逐層的加入 ADB 並指派它本身的延遲時間，去減少 MDSV 時脈樹的時序差異。由於 ADB 可以被用來產生額外的延遲時間，因此能調整時脈樹的延遲時間與時序差異。最後，對於 MDSV 的時脈樹，我們提出了一個有效的演算法，以最少的執行時間加入 ADB 去改善時序差異，並讓 ADB 消耗最少的面積。相較於目前最好的加入 ADB 演算法，實驗結果顯示了我們的方法能減少 42.40% 的 ADB 面積以及加快 117 倍的執行時間。同時再以整數線性規劃方法來評估我們的演算法，MILP-0 以對每一個 ADB 延遲之最小化為目標，其所有 ADB 數的總延遲較功率模式導向時脈樹最佳化結果平均少 5%，MILP-1 以對插入 ADB 總數之最小化為目標，其所有插入 ADB 總數較功率模式導向時脈樹最佳化結果平均少 8%，但時脈延遲仍然與功率模式導向時脈樹最佳化結果相同。實驗結果證明了我們的方法確實達到最佳化。

中文關鍵詞：功率消耗、時脈樹合成器、時序差異、動態式的多重供應電壓、可調整延遲時間的緩衝器、整數線性規劃。

英文摘要：

英文關鍵詞：

結合可調整延遲時間緩衝器及兼顧負載平衡與最小時序偏移之時脈樹合成方法應用於動態式多重供應電壓佈局環境(II)

Clock Tree Synthesis Associated with Adjustable Delay Buffers for Load Balancing and Skew Minimization in Multiple Dynamic Supply Voltages (II)

計畫類別：個別型計畫

國科會計畫編號：NSC 101-2221-E-343-004

執行期限：101年8月1日至102年7月31日

主持人：南華大學資工系 蔡加春教授

摘要--- 對目前的奈米積體電路設計來說，功率消耗是個很重要的影響。對於這個問題，動態式多重供應電壓(multiple dynamic supply voltage, MDSV)設計方法是個有效的解決方法。然而，在功率模式切換的時候，會增加時序差異的改變，進而增加修正時序差異的困難度。在此，我們提出了一個負載平衡式的時脈樹合成器，並加入可調整延遲時間的緩衝器(adjustable delay buffer, ADB)，讓使用MDSV設計方法的晶片能有最小的時序差異。一旦初步的時脈樹已建立完成，我們會逐層的加入ADB並指派它本身的延遲時間，去減少MDSV時脈樹的時序差異。由於ADB可以被用來產生額外的延遲時間，因此能調整時脈樹的延遲時間與時序差異。最後，對於MDSV的時脈樹，我們提出了一個有效的演算法，以最少的執行時間加入ADB去改善時序差異，並讓ADB消耗最少的面積。相較於目前最好的加入ADB演算法，實驗結果顯示了我們的方法能減少42.40%的ADB面積以及加快117倍的執行時間。同時再以整數線性規劃方法來評估我們的演算法，MILP-0以對每一個ADB延遲之最小化為目標，其所有ADB數的總延遲較功率模式導向時脈樹最佳化結果平均少5%，MILP-1以對插入ADB總數之最小化為目標，其所有插入ADB總數較功率模式導向時脈樹最佳化結果平均少8%，但時脈延遲仍然與功率模式導向時脈樹最佳化結果相同。實驗結果證明了我們的方法確實達到最佳化。

關鍵詞：功率消耗、時脈樹合成器、時序差異、動態式多重供應電壓、可調整延遲時間的緩衝器、整數線性規劃。

Abstract--- Power consumption is known to be a crucial issue in current nanometer-based chip designs. To tackle this problem, multiple dynamic supply voltage (MDSV) designs are proposed as an efficient solution in modern chip designs. However, the increasing variability of clock skew during the switching of power modes leads to an increase in the complication of clock skew reduction in MDSV designs. In this project, we propose a tunable clock tree structure with considering loading balance by adopting the adjustable delay buffers (ADB). The ADBs can be used to produce additional delays, hence the clock latencies and skew become tunable in a clock tree. Importing a buffered clock tree, the ADBs with delay value assignments are inserted to reduce clock skew in MDSV designs. An efficient algorithm of ADB insertion for the minimization of clock skew, area, and runtime in MDSV designs has been presented. Comparing with the state-of-the-art algorithm, experimental results show maximum 42% area overhead improvement and 117 \times runtime speedup. Finally, we applied the integer linear programming to the above problem to evaluate our approach. The simulated results approve that our approach can get the optimal solution.

Keywords: power consumption, clock tree synthesis, clock skew, multiple dynamic supply voltage, adjustable delay buffer, integer linear programming.

一、簡介

隨著積體電路製程的進步，功率消耗對於現今的奈米設計來說是個重要的考量因素。近年來，多重供應電壓(multiple supply voltage, MSV)的設計方式可解決功率消耗的問題；它和傳統單一供應電壓的設計方法不同之處在於，MSV方法將原本已經設計好的電路分割成數個電壓島(voltage island)，並將相同供應電壓的單元區塊(cell block)擺放在同一個電壓島。而著重效能的區塊則施以較高的供應電壓。利用不同的供應電壓設計，降低整體的晶片功率消耗。然而，MSV設計方法算是靜態的概念，因為此方法是在電路的設計階段即指定各電壓島的供應電壓。

為了較符合晶片在實際操作過程下，各功能區塊的非定性動態的改變供應電壓運作情況，動態式多重供應電壓(multiple dynamic supply voltage, MDSV)設計方式被提出來了，並藉此提供動態地節省能量。當電壓島處理非著重效能的工作時，可以使用較低的供應電壓。相對的，當處理大量計算工作的時候，電壓島就使用較高的供應電壓。因此，能夠以動態的方式調整每個電壓島的供應電壓，更進一步地降低整體的晶

片功率消耗，且能保持原本的晶片效能。圖 1 顯示了當系統晶片(system-on-chip)處理不同的工作時，各電壓島所使用的供應電壓情況。

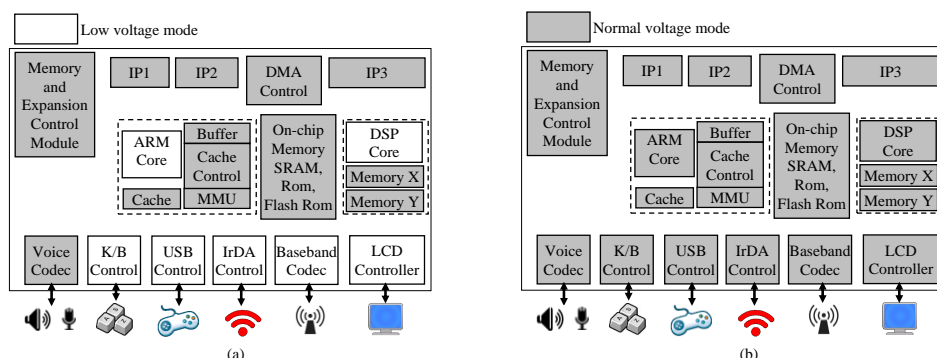


圖 1 系統晶片操作在不同模式下的電壓分配情況，(a)當操作在低計算量的工作，如播放音樂時；(b)當操作在全速，如播放高畫質影片時。

圖 2 是一個使用動態多重供應電壓(multiple dynamic supply voltage, MDSV)設計方法的例子，它包含了 38 個被切割出的電壓島。在全速模式(full speed mode)動作下，如圖 2(a)所示，所有電壓島均操作在標準電壓。對於非全速或不需要高效能的情況，如圖 2(b)之動作模式一(active mode 1)和圖 2(c)之動作模式二(active mode 2)所示，就可將部分的電壓島操作在低電壓，進而節省功率，並同時在所規定的時間內完成計算工作。

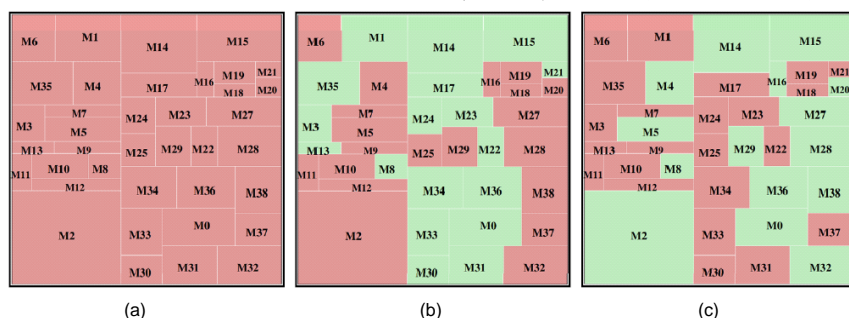


圖 2 一個低功率設計的例子。將晶片分割成數個電壓島(模組)，並讓他們在標準電壓(以紅色表示)與低電壓(以綠色表示)的操作模式下切換，進而達到節省功率消耗且能維持晶片效能的目標。(a)操作在全速模式(full speed mode)，(b)動作模式一(active 1 mode)，(c)動作模式二(active 2 mode)

當系統被分割成數個電壓島的時候，必須建立一個時脈樹(clock tree)去連接這些電壓島的時脈端點(clock sink)，然後傳送時脈信號，同步這些電壓島的動作。並在時脈樹擺放緩衝器(buffer)，進一步地縮短時脈延遲(clock delay)及控制在合理時序差異(clock skew)下，進一步改善時脈樹的效能。

在過去數年裡，對於高效能的積體電路設計來說，建立一個具有最少時序差異之時脈樹的技術仍舊佔據了相當重要的一部分。在 90 年代初期，Tsay [13] 首先提出了零時序差異(exact-zero skew)時脈繞線演算法。使用 Elmore 延遲時間模型，以由下而上的方式開始反覆地建立零時序差異子樹，最後完成零時序差異時脈樹的建立。但 Tsay 的方式並沒辦法保證時脈樹有最短的線段總長。Chao et al. [1] 等作者提出了延後合併嵌入(deferred-merge embedding, DME)演算法。此演算法在時脈樹的建置過程中考量了零時序差異與最短時脈樹之線段總長。以由下而上的方式(bottom-up)計算合併之線段長度，之後再以由上而下的方式(top-down)整合分支節點，完成具有最短線長之零時序差異時脈樹。由於 DME 方法使用分割式的方式建立時脈樹，所以仍舊無法保證時脈樹有最短的線段總長。Eda [3, 4] 利用叢集的方式(clustering)改良 DME 方法，進一步縮短線長，然後合成出一個非平衡架構(non-balanced)的時脈樹。除了最佳化原始的時脈樹，仍有其他學者提出以 DME 方式為架構的研究，以加入緩衝器(buffer insertion)或調整線寬(wire sizing)的方式去進一步改善時脈樹的時脈延遲與時序差異。

隨著階層式時脈樹(hierarchical-based clock tree)設計方法被提出後，對於在時脈樹上加入緩衝器而言，此方法能夠提供一個簡單的繞線結果。Mehta et al. [9] 等學者提出了將時脈樹劃分為數層之後，再逐一建立的方法，他們先將晶片面積切割為數個區塊，讓總區塊數的負載電容值能夠最小，進而建立時脈樹，並加入緩衝器於每個區塊的 root 端。學者 Shelar [11] 將時脈樹的目標由低時脈延遲改變為低功率消耗，他以最小擴張樹(minimum spanning tree)的方式去估計各時脈端點之間的連線距離，再反覆地以叢集的方式將時脈端點們互相連接，使其之間的連線距離最短，然後完成具最小電容值的時脈樹。

近年來，由於動態的多重供應電壓(multiple dynamic supply voltage, MDSV)設計方式逐漸地被應用於要求低功率消耗的設計上，在不同電壓模式下進行操作時，對於時脈樹的時序差異有決定性的影響。圖 3(a)

是一個使用 MDSV 方法的例子，因為此方法是利用功率模式(power mode)的概念，也就是依據各電壓島的操作模式給定不同的電壓，例如：全速(full speed)時使用電壓模式二(active 2)，保持高效能。而在其他動作或等待狀態時使用較低電壓的模式一(active 1)，藉此節省功率消耗，如圖 3(b)所示。當晶片操作在全速模式下，如圖 3(c)所示，時脈信號從最上層的時脈源(clock source)經過兩層的緩衝器之後，到達 A、B 及 C 三個電壓島的時脈延遲均為 2 個單位時間，因此，時序差異為 0。當晶片切換到 Active 2 模式時，從圖 3(b)得到，A 與 C 電壓島的操作電壓由原本的模式二(1.2v)降低為模式一(1.0v)，所以緩衝器的自身延遲時間也由 1 個單位時間增加到 3 個單位時間，更讓整體的時脈延遲改變成最大的 4 個單位時間與最小的 2 個單位時間，形成了 2 個單位時間的時序差異，如圖 3(d)所示。

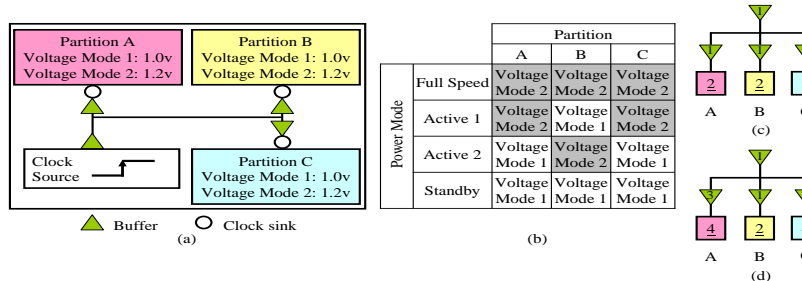


圖 3 (a)以 MDSV 方法設計之晶片，(b)各電壓島之電壓模式，(c)操作在全速模式時的時脈樹，(d)操作在 Active 2 模式時的時脈樹

為了解決使用 MDSV 方法設計晶片所造成的時序差異，Su et al. [12]等學者首先採用具可調整延遲時間式的緩衝器(adjustable delay buffer, ADB) [6, 10]加入到時脈樹裡面去調整時序差異。當傳送信號時，ADB 被用於產生額外的延遲時間，因此可以使用 ADB 去調整時脈樹的延遲時間與時序差異，在不同的電壓操作模式下，都可以讓修正過的時脈樹符合所需的時序差異條件。他們的方法是先讀取一個有加入緩衝器的時脈樹，接著再針對不同的功率模式加入 ADB 去產生額外的延遲時間，讓隨著功率模式而改變的時序差異能夠被減小。對於單功率模式與多功率模式，也提出了一個隨機啟發式的逐步調整 ADB 延遲時間方法，讓修正過的時脈樹能有允許的結果。圖 4(a)為一個操作在兩種電壓模式下的時脈樹，以模式一(Mode 1)來說，最長與最短的時脈延遲時間分別是 16 和 3 的單位時間，所以模式一的時序差異是 13 個單位時間 (i.e., 16-3)；而模式二(Mode 2)最長與最短的時脈延遲時間分別是 14 和 2 的單位時間，時序差異是 12 個單位時間 (i.e., 14-2)。為了最小化時序差異，他們利用廣度優先搜尋(breadth-first search)的方式，如圖 4(b)，找出可調整延遲時間式的緩衝器 ADB 的延遲時間，並將 ADB 擺放到時脈樹中。最後，可以將模式一與模式二的時序差異都減少至 4 個單位時間，如圖 4(c)所示。

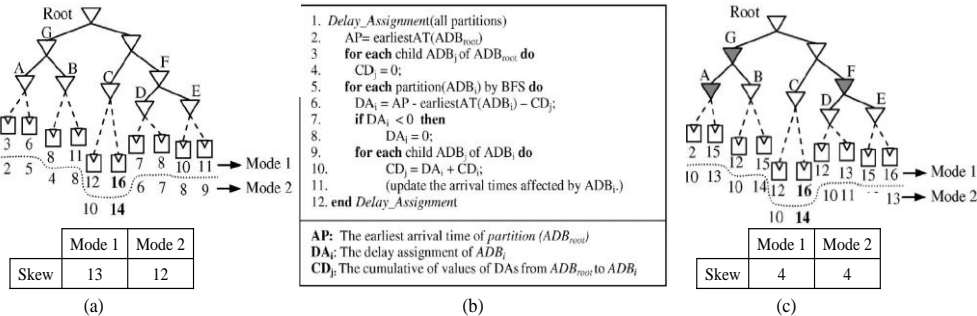


圖 4 (a)未加入 ADB 的兩種模式之時序差異，(b)指定 ADB 之延遲時間的演算法，(c) 加入 ADB 後，兩種模式之時序差異變小了

而 Lung et al. [8]等作者提出了考慮功率模式的時脈樹最佳化(power-mode-aware clock tree optimization, CTO)方法。此方法同樣利用 ADB 且搭配一組多工器去選擇加入 ADB 之後的時脈路徑，讓使用 MDSV 設計之電路的時序差異獲得改善。CTO 的方法包含了兩個部份，第一部份是晶片層的時脈樹最佳化(chip-level CTO)，第二部份是模組層的時脈樹最佳化(module-level CTO)。在所有可能發生的功率模式下，chip-level CTO 用於減少其全域時序差異；而 module-level CTO 著重在各電壓島操作在不同電壓下的時序差異控制。對 chip-level CTO 來說，他們設計了考慮功率模式的緩衝器(power-mode-aware buffer, PMAB)。

二、 研究背景

A. 動態式的多重供應電壓設計 (multiple dynamic supply voltage, MDSV)

隨著手持式電子設備愈來愈普及，為了延長電池的使用時間，MDSV 設計方法已用於這些設備的設計上。圖 5 為一個使用 MDSV 設計方法的例子，此方法可以解決使用傳統多重供應電壓(multiple supply voltage, MSV)設計方式的功率過大問題。對於效能需求不高的工作，功率模式可以被切換到待機模式，讓功率降低。

對於需要大量計算的工作，可以將功率模式切換到全速模式，保持高效能。

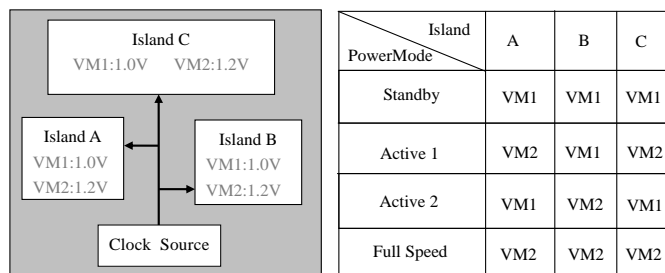


圖 5 MDSV 設計方法可以指派每個電壓島工作在多個電壓模式(voltage mode, VM)。以設計不同的功率模式(power mode, PM)和控制功率模式的切換時間，可以讓整體的功率降低且保時系統的效能。

B. MDSV 設計下的時脈延遲與時序差異

對於以 MDSV 方法設計的晶片來說，在不同的電壓模式下進行操作時，要如何建立其時脈樹，並同步地傳送時脈信號給各個電壓島，避免時序差異(clock skew)發生，這將是個重要的課題。圖 6(a)為一個加入了緩衝器(buffer)的 MDSV 時脈樹。其中，紅色三角形與其包覆的數字分別表示緩衝器與緩衝器的延遲時間；而圓圈與其包覆的數字分別表示正反器(flip-flop)與每個分支的延遲時間總和。當操作在全速模式的時候，最長與最短的路徑延遲時間分別是 5 個(黑色實線)和 3 個(黑色虛線)單位時間，所以時序差異是 2 個單位時間。當操作在動作模式一的時候，如圖 6(b)所示，時序差異增加到 7 個單位時間。原因是某些緩衝器操作在較低的工作電壓，使得緩衝器的延遲時間增加，進而使時脈樹違反了原先設定的時序差異條件。

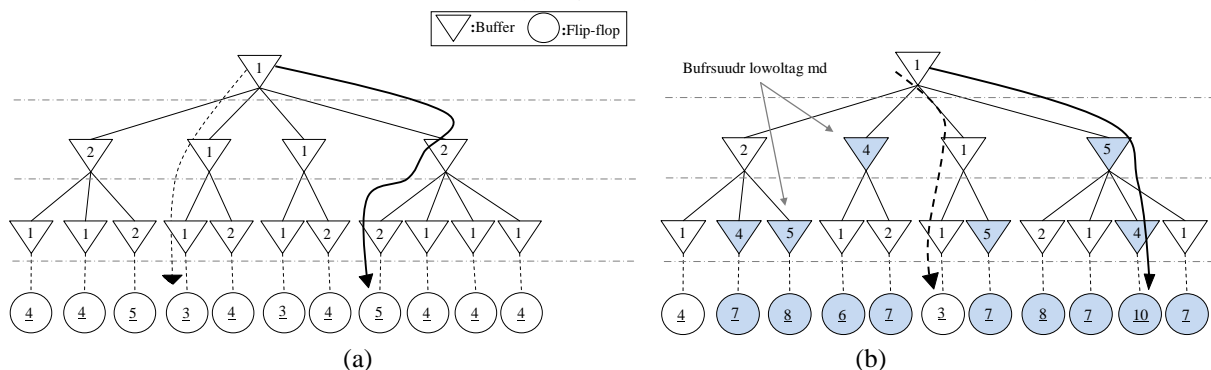


圖 6 當時脈樹在不同功率模式下的時序差異 (a)全速模式的時序差異為 2 個單位(符合時序差異的約定值) (b) 動作模式一的時序差異為 7 個單位(違反原先設定的時序差異條件)

C. 可調整延遲時間式的緩衝器(ADB)的模型

圖 7 是具可調整延遲時間式緩衝器(adjustable delay buffer, ADB)的模型，是將兩個反向器並聯後再加上一個選擇(SELECT)接腳去控制驅動模式，藉此讓 ADB 提供兩種延遲時間。亦可並聯多個 ADB 與增加選擇接腳，去達成具有更多種延遲時間的緩衝器。但受限於積體電路的實體設計情況，ADB 的延遲時間無法連續地調整。因此，對於有違反時序差異條件的 MDSV 時脈樹，可以將原本的緩衝器替換為 ADB，去修正時序差異，進而讓全部的電壓島都能同步動作。

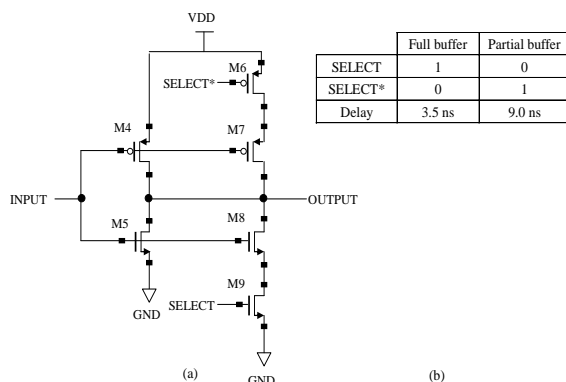


圖 7 ADB 的模型與其所能提供的延遲時間

三、主要貢獻

對於具有緩衝器的時脈樹建立，目前最佳的方法為 Mehta et al. [9]，可以減少建立時的負載不平衡問

題，但以切割方式去叢集時脈端點會增加繞線時的長度。不同於[9]，我們提出了以圖形理論式的叢集方式，確保負載能夠平衡且能夠減少繞線長度。此外，我們更提出了以功率模式導向設計之叢集方式，將類似的切換功率模式做預先叢集，進而減少 ADB 所需的數量或電晶體數目。

對於 MDSV 設計的時序差異，目前最佳的方法 Su et al. [12] 可以減少時序差異，但需要花費較多的執行時間去選擇加入 ADB 的位置。不同於[12]，我們提出了一個有效的演算法去加入 ADB，並讓 MDSV 設計的時序差異可以達到最小。無論是單功率模式或多功率模式，在加入 ADB 的架構下，他們的時脈樹時序差異問題都能被有效的解決。我們將此架構分為兩個主要步驟，分別是在單功率模式下的兩階段加入 ADB 演算法與多功率模式下的整合型加入 ADB。

兩階段的加入 ADB 演算法包含了由上而下的加入 ADB 與由下而上的改善加入結果兩部份。所以這個演算法能在兩次穿越時脈樹的過程中完成 ADB 的加入。然後，將此結果交給多功率模式下的加入 ADB 演算法。此外，我們也考慮了 ADB 的資源消耗問題。由於 ADB 的自身延遲時間無法連續地調整，僅允許固定的延遲時間。如果為了讓 ADB 有較多的延遲時間選擇，將會增加所需要的面積，因此，我們也對這個所需面積的問題提出了相關的演算法。

在整個架構上，我們提出了一個負載平衡式的時脈樹合成器，讓 MDSV 設計有最小的時序差異。此架構分為兩個主要步驟，第一步驟是建立初步的負載平衡式時脈樹並加入緩衝器，而第二步驟是對 MDSV 設計的時序差異做後端最佳化，第三步驟是以整數線性規劃(ILP optimization)對插入 ADBs 與緩衝器作指定時間延遲之最佳化數學方法等兩者獲得結果作一驗證及比較。我們的主要貢獻如下：

- 對於單功率模式，我們開發了一個兩階段的加入 ADB 演算法，並能指派 ADB 的自身延遲時間。此演算法能夠決定加入 ADB 的位置並減少時脈樹的時序差異。
- 對於多功率模式，我們提出了一個有效的整合型加入 ADB 演算法，同樣能指派 ADB 的自身延遲時間。此也算法也一樣能決定加入 ADB 的位置並減少時脈樹的時序差異。
- 為了考量與減少節省晶片資源，我們提出了一個減少延遲時間的演算法。此演算法能減少 MDSV 設計中 ADB 的面積。
- 整數線性規劃個別對所有 ADB 數的總延遲與對插入 ADB 總數之最小化為目標，我們提出的功率模式導向時脈樹結果是最佳化的。

表 1 顯示了 Su [12] 等學者和我們的方法在架構上的比較。這兩個方法都使用了線性時間的最佳化演算法去指定 ADB 的延遲時間。於單功率模式與多種功率模式下，我們的方法在加入 ADB 時能展現出較快的速度優勢。此外，我們的方法在 ADB 的資源上會消耗的比較少。

表 1 Su [12] 等學者和我們的方法在架構上的比較。

	Su [12]	我們的方法
指定 ADB 的延遲時間	於線性時間的最佳化指定	於線性時間的最佳化指定
在單功率模式下加入 ADB	隨機啟發式與逐步調整	分為兩階段
在多種功率模式下加入 ADB	增量啟發式	和單功率模式下的結果相結合
減少 ADB 的延遲時間值	無	合併延遲時間值

四、問題定義

對於一個尚未具有時脈樹的動態多重供應電壓(multiple dynamic supply voltage, MDSV)設計，而此設計已包含了數個電壓島。假設每個電壓島有兩種電壓模式：1.0v 模式與 1.2v 模式。我們先建立 MDSV 設計的時脈樹並加入緩衝器。接著，加入可調整延遲時間式緩衝器(adjustable delay buffer, ADB)，並指派其自身延遲時間去進一步縮小時脈樹的時序差異(clock skew)。我們參考了 Synopsys 的業界元件庫(industry cell library)取得每個元件的時間資訊，而延遲時間是參考每個分支點上的緩衝器時間總和。所以，時脈樹的延遲時間與時序差異就能被計算出來。此時脈樹建置的問題定義如下：

輸入：給定一個 MDSV 設計，其具有一組時脈端點(如正反器)、已被定義好的端點電容量、數個電壓島及各種功率模式。

輸出：在 MDSV 設計下，合成出一個具有緩衝器的時脈樹建置，且不會違反時序差異。

五、演算法

我們的演算法流程可以分為主要步驟(main step)和後端最佳化步驟(post-optimization step)兩部份。主要步驟是建立初步的時脈樹並加入緩衝器。首先，讀取各電壓島的時脈端點位置，並建立其最小擴張樹(minimum spanning tree, MST)表示鄰近的時脈端點與估計線段負載。為了減少連接各電壓島之間的連線延遲與時序差異，我們將叢集(clustering)方法分為本地階層叢集(local-leveled clustering)與全域階層叢集(global-leveled clustering)。前者限制了叢集的範圍為每個電壓島，而後者並沒有此限制。在叢集完成之後，

這些時脈端點所構成的數個電壓島已達到負載平衡的情況，因此可以執行階層式的時脈繞線並逐層加入緩衝器。建立完時脈樹之後，我們的後端最佳化步驟可以減少 MDSV 設計的時序差異。此步驟包含了四個部份，分別是：由上而下地加入 ADB 並指派延遲時間、由下而上地估計 ADB 並擴展延遲時間、在多種功率模式下的加入 ADB 並指派延遲時間，以及延遲時間的改善。

A. 負載平衡式的時脈端點叢集法 (Load-balanced sink clustering)

對於時脈樹的拓撲與實體架構，負載平衡式的時脈端點叢集法都可以完成。首先，讀取時脈端點的位置，然後建立其最小擴張樹(MST)用來表示鄰近的時脈端點與估計線段負載。接著，為了減少連接各電壓島之間的連線延遲與時序差異，我們將叢集(clustering)方法分為本地階層叢集(local-leveled clustering)與全域階層叢集(global-leveled clustering)。前者限制了叢集的範圍為每個電壓島，而後者並沒有此限制。在叢集完成之後，這些時脈端點所構成的數個電壓島已達到負載平衡的情況，因此可以執行階層式的時脈繞線並逐層加入緩衝器。

為了執行負載平衡式的時脈端點叢集法，我們採用圖形理論的叢集法[5]。透過反覆地搜尋 MST 數的最小成本線段然後一一移除，最後可以完成負載平衡式的時脈端點群。而每個時脈端點群都能保持住他們的相干特性，也就是比較短的繞線長度。在初步的叢集步驟裡，我們參考元件庫並設定緩衝器能夠驅動最大的電容值為每個叢集的電容臨界值，然後決定目標叢集的數量 $N = \alpha C_T / C_{buf}$ ，其中， C_T 表示 MST 的時脈端點與線段等效電容之總和，而 C_{buf} 表示元件庫中緩衝器能夠驅動最大的電容值。相較於以 DME 方法所完成的繞線長度，使用 MST 所估計的繞線長度會低於實際的繞線長度，所以我們提供一個使用者定義的參數 α 作為錯誤修正。

關於負載平衡式的端點叢集方法，我們使用圖 8 來做說明。對於給定的時脈端點，先建立其最小擴張樹(MST)，如圖 8(a)與(b)所示。根據 MST 的時脈端點總電容量加上線段總電容量與緩衝器所能推動的最大電容量之比值(ratio)，來決定要將 MST 切割為多少個區塊，此方法稱為比值切割技術(ratio-cut technique)。圖 8(c)為第一次使用比值切割技術後所得到的左右兩區塊。然後，將右邊已切割區塊表示為一個栓點(tapping point)，再對左邊的 MST 進行比值切割技術，的到第二次的切割結果，如圖 8(d)所示。反覆地切割並得到最終的結果，如圖 8(e)所示。

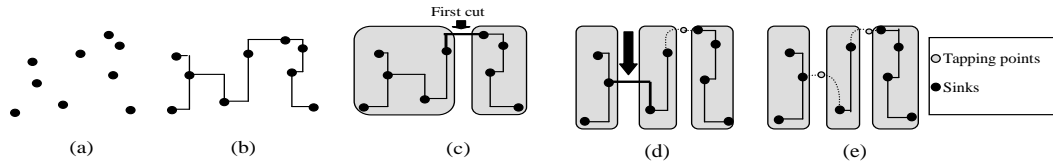


圖 8 考慮負載平衡的端點叢集方法

B. 逐層地建立時脈樹並加入緩衝器

在完成單一階層的叢集之後，我們使用以叢集為基礎的 DME(deferred-merge embedding)演算法，對每個叢集進行時脈樹的建立，如圖 9 所示。之後，加入緩衝器於每個叢集的 root 端，再與其他階層的時脈樹進行合併，直到完成。

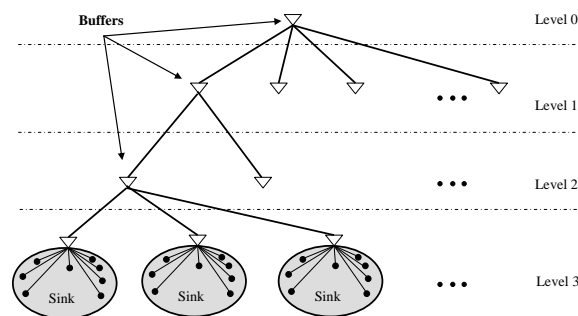


圖 9 逐層地建立時脈樹並加入緩衝器

C. 加入 ADB 並指派其延遲時間

為了解決複雜的功率模式下之時序差異問題，傳統的方式是以加入緩衝器並調整其尺寸或是以調整線寬的方式去解決。即使在單功率模式下的時序差異問題已被成功地解決，但在其他的功率模式下也許會失敗。所以，我們提出了一個有效的演算法去解決 MDSV 設計的時序差異問題。參考了[12]的最佳化指派 ADB 延遲時間與在單功率模式下加入 ADB 的方法，我們改善了前述的策略並將其延伸到 MDSV 設計的時序差異問題上。我們的方法包含了四個部份，分別是：由上而下地加入 ADB 並指派延遲時間、由下而上地估計 ADB 並擴展延遲時間、在多種功率模式下的加入 ADB 並指派延遲時間，以及延遲時間的改善。

1) 由上而下地加入 ADB 並指派延遲時間

為了避免使用額外的 ADB，我們以由上而下的方式，在較上層的時脈樹部份加入 ADB 去影響大部分

的時脈子樹。此種策略能讓較高層的時脈樹有較高的優先權去加入 ADB。

在實現這個由上而下的演算法時，我們將時序差異的約束條件設定為零。首先，分析整個時脈樹，並於單功率模式下找出時脈樹的最大延遲時間。由於改變時脈樹 root 端的延遲時間並無法改善時序差異，所以我們從 root 端以下的階層用由上而下的方式掃描每個時脈樹節點。然後將整個時脈樹定義為全域時脈樹，而每個節點下的時脈樹定義為本地時脈樹。若本地時脈樹的最大延遲時間不等於全域時脈樹的最大延遲時間，我們會加入一個 ADB 到時脈樹的節點並指派兩個最大延遲時間之差值當作是 ADB 的延遲時間。

圖 10 是我們所提出的演算法，此演算法以由上而下的方式掃描整個時脈樹(第 2 行)。若本地時脈樹的最大延遲時間(L_{lt})小於全域時脈樹的最大延遲時間(L_{gt})(第 3 行)，則此演算法會加入一個 ADB 到時脈樹的節點，並指派一個延遲時間(ADB_d)給它(第 4-5 行)。若兩者的延遲時間相等，則此演算法將繼續掃描下一個時脈樹節點(第 7-8 行)。

```

Algorithm: Top-Down ADB Insertion and Delay Value Assignments
Input: A buffered clock tree
Output: The buffered clock tree with ADB insertion
1 {
2   for each clock tree node (top-down order) do
3     { if ( $L_{gt} > L_{lt}$ ) then
4       {   Insert ADB;
5          $ADB_d = L_{gt} - L_{lt}$ ;
6       }
7     } if ( $L_{gt} = L_{lt}$ ) then
8       {   Keep original buffer;
9     }
10  }
11 }
    
```

圖 10 由上而下地加入 ADB 並指派延遲時間之演算法

圖 11(a)和(b)顯示了這個演算法的例子。從 Level 1 由上而下掃描之後，得到時脈樹的本地最大延遲時間(L_{lt})為 10 個單位，且不等於全域時脈樹的最大延遲時間(L_{gt}) 11 個單位(如虛線所示)。所以我們在節點 A 和 B 加入分別加入 ADB，並指派他的延遲時間為兩者的延遲時間差(11-10)。之後，繼續以同樣的方法掃描整個時脈樹。

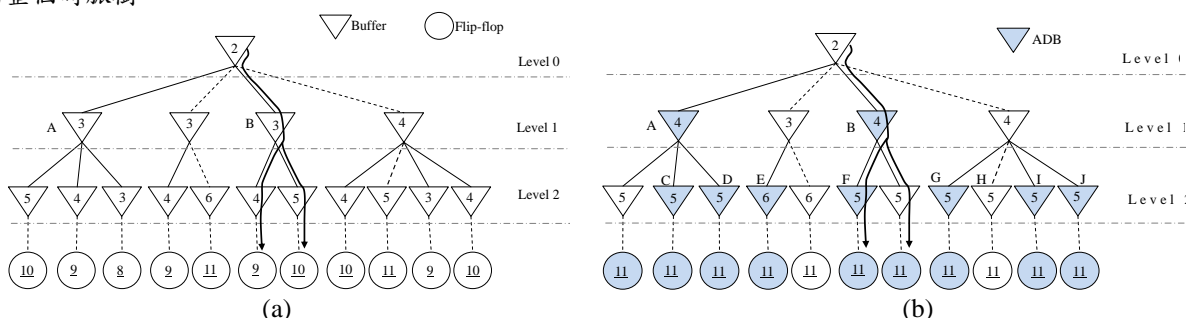


圖 11 由上而下地加入 ADB 並指派延遲時間之演算法，(a)初始的時脈樹與(b)修正後的時脈樹

2) 由下而上地估計 ADB 並擴展延遲時間

在前面的步驟中，單功率模式下的時序差異已經被解決了。但在零時序差異的約束條件下，這個結果對於 ADB 的數量和所需面積來說會有過於浪費的問題產生。在實際的晶片設計上，有限的時序差異(非零時序差異)在時脈樹的建立過程中是被允許的。根據這點，我們觀察初步加入 ADB 的結果後發現，位於較上層的 ADB 改善了大部分的時序差異問題。所以我們在由下而上的步驟裡，著重在減少驅動著小型本地時脈樹的 ADB 數量。

在這個步驟裡，於使用者定義的時序差異約束條件下，我們提出了由下而上的減少 ADB 數量並加長延遲時間的演算法，去減少 ADB 的面積消耗。首先，以由下而上的方式掃描整個時脈樹，一旦 ADB 的延遲時間小於或等於時序差異約束條件，這個 ADB 將被定義為冗餘(redundant)ADB，因而能夠被移除掉。反之，若 ADB 的延遲時間大於時序差異約束條件，原本的延遲時間會被延長為一個範圍，且任何位於此範圍的延遲時間值，都能符合時序差異約束條件。所以就能利用此範圍去減少 ADB 的電晶體數量。

圖 12(a)是我們所提出的演算法，此演算法以由下而上的方式掃描整個時脈樹(第 2 行)。如果 ADB 的延遲時間(ADB_d)小於或等於使用者定義的時序差異(user-defined skew, $Skew_u$)(第 3 行)，則這個演算法會把這個 ADB 從時脈樹的節點上移除掉(第四行)。反之，這個演算法會保留這個 ADB，並指派 ADB_d 與 $Skew_u$ 的差值給它(第 7-8 行)。我們定義這個差值為 ADB 的延長時間(ADB_{ed})，它可以將固定的延遲時間延長到某個範圍的延遲時間，進而提供更多的延遲時間候選者給多功率模式下的時脈樹。

```

Algorithm: Bottom-Up ADB Elimination and Delay Value Extensions
Input: A clock tree with ADB insertion
Output: The refined clock tree with ADB insertion
1 {
2   for each clock tree node (bottom-up order) do
3     { if ( $ABD_d \leq Skew_u$ ) then
4       { Delete ADB;
5       }
6     else
7       { Keep ADB;
8          $ADB_{ed} = ABD_d - Skew_u$ ;
9       }
10    }
11 }

```

圖 12 由下而上地估計 ADB 並擴展延遲時間之演算法，(a)修正前的時脈樹與(b)修正後的時脈樹

圖 13(a)和(b)顯示了這個演算法的例子。此演算法從 Level 2 開始，以由下而上的方式掃描整個時脈樹。我們假設 $Skew_u$ 為 1 個單位時間，而加在節點 C、F、G 和 J 的 ADB 被定義為冗餘(redundant)ADB。加在節點 D、E 和 I 的 ADB 被無法被移除，但其延遲時間可以被延長。以節點 D 的 ADB 來說，他的 ADB_{ed} 值為 $4(5-1)$ 個單位時間。以同樣的方法可以求得節點 E 和 I 的 ADB_{ed} 值分別為 $5(6-1)$ 個單位時間和 $4(5-1)$ 個單位時間。

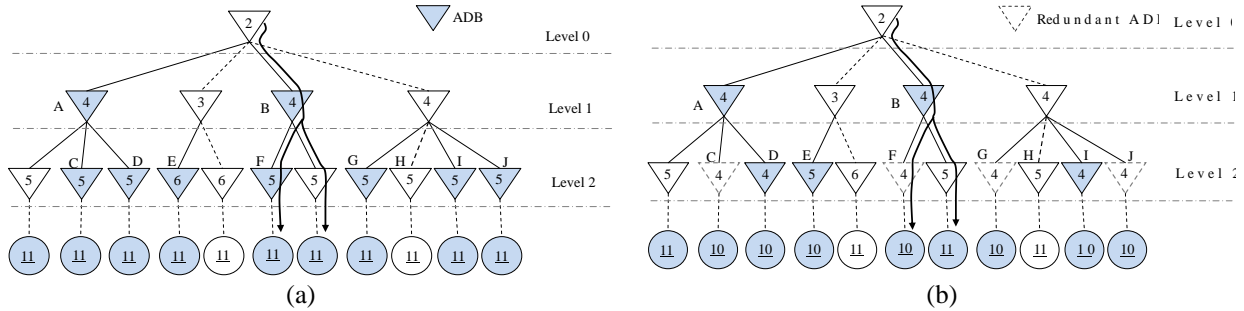


圖 13 由下而上地估計 ADB 並擴展延遲時間之演算法，(a)修正前的時脈樹與(b)修正後的時脈樹

3) 在多種功率模式下的加入 ADB 並指派延遲時間

為了在多種功率模式下的時脈樹加入 ADB 並指派他們的延遲時間，對每個功率模式都使用一個二階段的演算法，去確認其時序差異條件不會在所有的功率模式下有違反的情況發生。然後將這些時脈樹的結果合併起來，之後，每個時脈樹的節點會有 0 個到和功率模式數(PM_{num})相同多個的延遲時間值被指派。

圖 14 是我們所提出的演算法。首先，掃描被合併之時脈樹的所有時脈子樹節點(第二行)。若節點的延遲時間總數(DV_{count})為零(第三行)，這個演算法會保留原先的緩衝器。若 DV_{count} 值不為零(第六、十行)，我們會加入 ADB 到時脈樹的節點(第七、十一行)。當節點的 DV_{count} 值小於 PM_{num} ，意味著 ADB 在每個功率模式下都不能被啟動。因此，我們會指派延遲時間給初步的緩衝器以及額外的延遲時間給 ADB。所以，最後被指派給 ADB 的延遲時間種類(ADB_{count})是 $DV_{count} + 1$ (第八行)。反之，就是指派 PM_{num} 。

```

Algorithm: ADB Insertion and Delay Value Assignments in Multiple Power Modes
Input: The refined clock tree with ADB insertion
Output: The final clock tree with ADB insertion
1 {
2   for each clock tree node do
3     { if ( $DV_{count} = 0$ ) then
4       { Keep original buffer;
5       }
6     if ( $DV_{count} > 0$  and  $DV_{count} < PM_{num}$ ) then
7       { Insert ADB;
8          $ADB_{count} = DV_{count} + 1$ ;
9       }
10    if ( $DV_{count} = PM_{num}$ ) then
11      { Insert ADB;
12         $ADB_{count} = PM_{num}$ ;
13      }
14    }
15 }

```

圖 14 在多種功率模式下的加入 ADB 並指派延遲時間之演算法

4) 延遲時間的改善

完成第三步驟之後，所有功率模式下的 ADB 位置與其延遲時間都已經決定好了。由於每個 ADB 的面積臨界值可以利用 ADB 的延遲時間種類數來估計，為了將面積臨界值最小化，ADB 的延遲時間種類 (ADB_{count}) 也應該要減至最少。所以，我們提出了合併延遲時間的演算法，對 ADB_{count} 做最佳化。在第二步驟中，我們已經確定了 ADB 的延長時間 (ADB_{ed}) 範圍，所以這個合併演算法能利用此區間去調整延遲時間，且不會有違反的情況發生。最後，找出延長時間範圍的交集，就能合併與分享延遲時間。

圖 15 是用來解釋合併延遲時間演算法的例子，而 ADB 的六種延長時間 (ADB_{ed}) 範圍已經在第二步驟決定了，如圖 15(a) 所示。接著，我們找出 ADB_{ed} 範圍的交集，如圖 15(b) 的藍色區塊所示。對於 (PM0, PM1, PM5) 和 (PM2, PM3) 來說，可以合併成兩個延遲時間，而最終會有三個延遲時間，如圖 15(b) 的藍色粗線所示。最後，得到 ADB 有三個延遲時間，分別為 6.5、9.5 和 11.5 個單位時間。

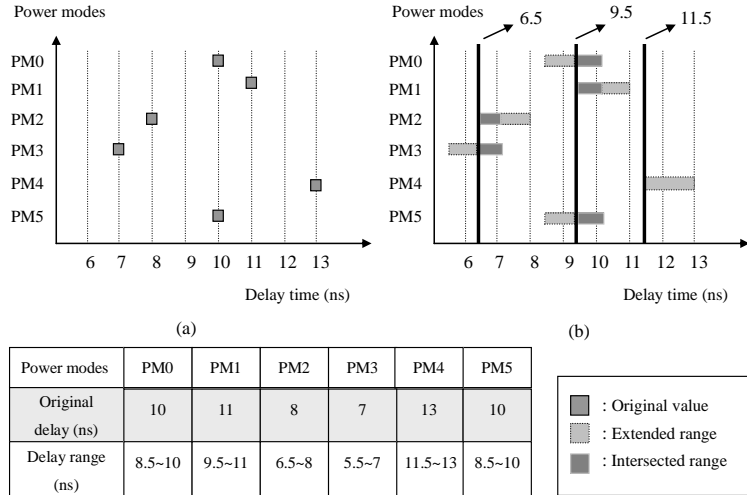


圖 15 合併延遲時間的演算法

5) 時間複雜度分析

在所提出的加入 ADB 演算法中，假設初步時脈樹中已被加入了 n 個緩衝器。兩階段的加入 ADB 演算法，時脈樹的合併與延遲時間合併過程都使用了寬度優先 (breadth-first traversal) 方式去遍歷時脈樹，所以他們的時間複雜度都是 $O(n)$ 。在最差的情況下，ADB 的指派延遲時間與延遲時間的合併都需要 $O(n)$ 的時間去計算結果，所以整體的時間複雜度是 $O(n^2)$ 。

六、實驗結果

我們透過 C/C++ 語言實現了我們的演算法，以標準測試例子作為評估。在每個標準測試例子裡設計了九到三十八個電壓島去模擬 MDSV 的操作環境。也參考了 Synopsys 的元件庫去定義實驗中所需的兩種電壓模式，分別為 1.0v 和 1.2v。在功率模式部份，除了全速模式與等待模式是讓全部的電壓島分別操作在 1.2v 與 1.0v，剩餘的功率模式都採隨機方式去指派各電壓島的操作電壓。為了估計整個時脈樹所使用的面積，我們以所需的電晶體數量作為參考。基本的非反向型緩衝器需要四個電晶體。此外，我們參考了 [10] 的設計去定義非反向型 ADB 所需的電晶體數量。當 ADB 的延遲時間增加一個單位，ADB 就需要增加四個電晶體。

表 2 列出了我們所使用的標準測試例子 (benchmark) 統計數據，並以他們的晶片規模由小至大排列。在所有的標準測試例子裡，我們設定 ADB 最多能有三倍的延遲時間調整範圍。由於 ADB 的延遲時間不能連續性的調整，所以我們必須指派固定的延遲時間。當 ADB 的延遲時間數量增加時，其所需的晶片面積也會同步增加，並影響到電晶體的數量。

表 2 標準測試例子的統計數據

Circuits	#Flip-flops	#Voltage Islands	#Power Modes
design 1	384	10	10
design 2	992	29	15
design 3	1536	39	25
design 4	3360	46	30
design 5	6144	38	40

表 3 列出了標準測試例子的延遲時間與時序差異之實驗結果，並實現 [12] 方法作為比較。分析了時脈樹的延遲時間之後，我們發現兩個方法都能修正時序差異，且不會有延遲時間過多的問題。這個結果確定在 MDSV 時脈樹加入 ADB 能夠有效的修正時序差異。

表 3 延遲時間與時序差異的實驗結果

Circuits	Skew Constraint (ps)	Worst Clock Skew (ps)			Average Clock Skew (ps)			Worst Clock Latency (ps)		
		LB clustering	[12]	Ours	LB clustering	[12]	Ours	LB clustering	[12]	Ours
design1	200	476	200	200	383	184	200	1316	1316	1316
	300	476	293	300	383	262	277	1316	1316	1316
design2	200	463	200	200	388	197	198	1560	1560	1560
	300	463	300	300	388	272	292	1560	1560	1560
design3	200	630	200	200	506	195	200	1667	1667	1667
	300	630	298	300	506	290	300	1667	1667	1667
design4	200	1018	199	200	785	196	200	2888	2888	2888
	300	1018	299	300	785	292	300	2888	2888	2888
design5	200	1167	198	200	796	196	200	3069	3069	3069
	300	1167	297	300	796	294	300	3069	3069	3069

表 4 同樣列出了我們和[12]的方法於標準測試例子的 ADB 數量、電晶體數量與執行時間之實驗結果。其中，面積使用率與執行時間對於實際的晶片設計是很重要的。我們的方法與[12]都能解決每個標準測試例子的時序差異問題。此外，我們的方法可以減少 7.59%–42.40%的電晶體數量，且最多能加快 117.84 倍的執行時間。

表 4 ADB 數量、電晶體數量與執行時間的實驗結果

Circuits	#Buffers	Skew Constraint (ps)	#ADBs			#Transistors			Runtime(s)	
			[12]	Ours	Improvement (%)	[12]	Ours	Improvement (%)	[12]	Ours
design1	33	200	19	19	0.00%	336	260	22.62%	0.01	0.01
		300	8	7	12.50%	216	168	22.22%	<0.01	<0.01
design2	79	200	61	61	0.00%	896	828	7.59%	0.17	0.01
		300	18	17	5.55%	644	516	19.88%	0.09	0.01
design3	127	200	100	100	0.00%	2192	1724	21.35%	1.05	0.04
		300	37	32	13.51%	1764	1016	42.40%	0.55	0.03
design4	337	200	199	196	1.51%	5172	4156	19.64%	12.84	0.13
		300	113	99	12.38%	4688	2980	36.43%	10.03	0.14
design5	519	200	289	286	1.04%	7680	6112	20.42%	50.67	0.43
		300	120	89	25.83%	6756	3996	40.85%	35.50	0.43

表 5 列出了標準測試例子以功率模式導向設計之時脈樹最佳化的延遲時間與時序差異實驗結果。從表中可以知道，此設計方法並不會導致過多的時序差異與延遲時間。

表 5 以功率模式導向之時脈樹最佳化的延遲時間與時序差異實驗結果

Circuits	Skew Constraint (ps)	Worst Clock Skew (ps)			Average Clock Skew (ps)			Worst Clock Latency (ps)		
		PM clustering	[12]	Ours	PM clustering	[12]	Ours	PM clustering	[12]	Ours
design1	200	476	200	200	383	184	200	1316	1316	1316
	300	476	293	300	383	262	277	1316	1316	1316
design2	200	506	200	200	414	196	199	1676	1676	1676
	300	506	294	300	414	275	294	1676	1676	1676
design3	200	605	200	200	457	194	200	1658	1658	1658
	300	605	295	300	457	287	298	1658	1658	1658
design4	200	1074	200	200	818	196	200	2881	2881	2881
	300	1074	298	300	818	291	300	2881	2881	2881
design5	200	1080	200	200	753	198	200	2792	2792	2792
	300	1080	299	300	753	291	300	2792	2792	2792

當 design3.def 在 MDSV 的設計方法下，圖 16(a)是未加入 ADB，僅有一般緩衝器的初步時脈樹。而圖 16(b)是加入共 32 個 ADB 之後的時脈樹。

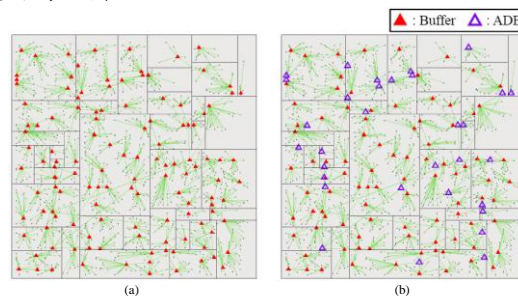


圖 16 design3.def 之(a)初步時脈樹與(b)加入 ADB 後之時脈樹

七、以整數線性規劃最佳化做驗證

最後，為了進一步驗證與評估所提出一個具有插入緩衝器之時脈樹建置與再具有插入 ADBs 修正之緩衝器時脈樹建置的後端最佳化等兩步驟之結果，我們將嘗試以此結果當已知資訊，重新設計以數學規劃方法(mathematical programming method)對此結果作最佳化，即對 ADB 插入總數量及每個 ADB 延遲時間的數量作最小化，此舉可將演算法與數學方法等兩者獲得結果作一比較，勢必可得到一些解析資料，了解所提出的演算法與數學方法最佳化結果的真實差異。

已知上述之後端最佳化演算法之結果，可得到緩衝器插入位置與數量、ADBs 插入位置與數量及邊界時序偏移等時脈樹的結構，因此，我們制定這些目標嵌入至混合整數線性規劃(Mixed Integer Linear Programming, MILP)；然而，為了對每個 ADB 延遲時間的數量作最小化，則將有大量的二元變數(binary variables)之需求，此舉將對 MILP 造成大量而複雜的制定函數，亦使 MILP 的執行時間冗長而欠實際性。因此，一個替代解決方法，改為對所有 ADBs 的延遲時間之總合作最小化，來取代對每個 ADB 延遲時間數量的最小化，此仍具有潛在的同等效用。綜合以上陳述，我們所設計的混合整數線性規劃(MILP)為：

已知我們所提出之演算法而所得到緩衝器與 ADB 插入時脈樹的結果，其具有 k 個節點(nodes, $i=1$ 至 k)、 p 種電源模式(power modes, $j=1$ 至 p)、時序偏移邊界 $Skew_u$ 及每個緩衝器與 ADB 的插入位置；其目標是在 p 種功率模式及時序偏移邊界 $Skew_u$ 的限制下，對 ADB 插入總數量及所有 ADB 延遲時間之總合作最小化。

以下我們定義一些符號而適用於 MILP：

C_{ij} ：在功率模式 j 下，於節點 i 的緩衝器已指定之固有延遲時間。

D_{ij} ：在功率模式 j 下，於節點 i 的 ADB 延遲時間之變數。

S_j ：在功率模式 j 下，表示最小時脈延遲(smallest clock latency)之變數。

L_j ：在功率模式 j 下，表示最大時脈延遲(largest clock latency)之變數。

B_i ：表示節點 i 是否插入 ADB 之 0-1 二元變數，1 與 0 分別代表插入 ADB 與否。

因此 MILP 設計之目標(objective)可定義如下：

Objective：

$$\text{Minimize } \alpha \sum_{i=1-k} B_i + (1 - \alpha) \sum_{j=1-p} \sum_{i=1-k} D_{ij}$$

其中 α 為使用者所定義，介於 0 與 1 之間，可對 ADB 插入的總數量與所有 ADB 延遲時間的總和等兩者作一權衡或折衷 (tradeoff)。

而相關之限制條件(related constraints)如下：

Constraints：

(1) 對每一個功率模式 j ，均應該滿足時序偏移邊界(Boundary skew)的限制，即

$$0 \leq L_j - S_j \leq Skew_u, \quad 1 \leq j \leq p$$

(2) 在每一個功率模式 j 下，從時脈樹的根(root)到任何樹葉(leaf)的每一條路徑 Q 的延遲時間，都應該維持大於等於最小時脈延遲 S_j 及小於等於最大時脈延遲 L_j 之間，即

$$S_j \leq \sum_{k_i \in Q} (C_{ij} + D_{ij}) \leq L_j, \quad 1 \leq j \leq p$$

(3) 在每一個功率模式 j 下，在節點 i 之 ADB 能被指定的額外延遲時間(additional delay value)，須被限制在我們事先設定的範圍內，如 $X-1$ 倍的延遲時間(如 $X=3$)，即

$$0 \leq D_{ij} \leq (X-1)C_{ij}, \quad 1 \leq j \leq p, \quad 1 \leq i \leq k$$

(4) 為了精確地計算 ADB 插入的總數量，當節點 i 在任一功率模式 j 下有被指定額外的延遲時間，表示該節點已插入 ADB，也就是 B_i 須設定為 1。其限制條件為所有功率模式之 ADB 延遲時間的總和須小於等於已插入 ADB (即 $B_i=1$) 延遲時間之總和 T ，即

$$\sum_{j=1-p} D_{ij} \leq TB_i, \quad 1 \leq i \leq k$$

以上 MILP 的做法可簡易分為兩種情況執行模式，即以 $\alpha=0$ 之 ADB 延遲時間總和之最小化，簡稱 MILP-0；與以 $\alpha=1$ 之 ADB 總數量之最小化，簡稱 MILP-1。這兩種情況之執行結果，再與我們演算法這結果作一比較，在相同的時序偏移邊界的限制下，以插入 ADB 總數量(#ADBs)、ADB 延遲時間之總和、最差的時序偏移值、最差的時脈延遲及執行時間等項目作比較。我們預期 MILP-0 的執行結果在 ADB 延遲時間總和將小於 MILP-1 者，即優於 MILP-1；而 MILP-1 的執行結果在插入 ADB 總數量將少於 MILP-0 者，即優於 MILP-0。但 MILP-0 與 MILP-1 兩者均可符合時序偏移邊界之限制，但最差的時脈延遲較難作預期觀察，須待實際執行出來之後，再作分析與比較。但仍有跡可循，依觀察現有發表的相關論文，插入較多

量的 ADBs，其最差的時脈延遲會有縮減的趨勢，因此我們預期 MILP-0 執行結果之最差的脈延遲將少於 MILP-1 者，即優於 MILP-1。表 6 所示為 MILP-0、MILP-1 與功率模式導向時脈樹最佳化結果之比較，由表中可知，MILP-0 以對每一個 ADB 延遲之最小化為目標，其所有 ADB 數的總延遲較功率模式導向時脈樹最佳化結果平均少 5%，但時脈延遲仍然與功率模式導向時脈樹最佳化結果相同；MILP-1 以對插入 ADB 總數之最小化為目標，其所有插入 ADB 總數較功率模式導向時脈樹最佳化結果平均少 8%，但時脈延遲仍然與功率模式導向時脈樹最佳化結果相同。因此，從表中可得到，除了整數線性規劃 MILP-0 與 MILP-1 個別的最小化目標之外，功率模式導向時脈樹最佳化結果如時脈延遲、電晶體總數與執行時間等，均得到最佳化的結果。

表 6 MILP-0、MILP-1 與功率模式導向時脈樹最佳化結果之比較
(MILP-0: minimize sum of delay values of each ADB, MILP-1: minimize number of ADBs)

Circuits	#Buf.	Skew Cons. (ps)	#ADBs			#Transistors			Sum of ADB delays (ps)			Worst clock latency (ps)			Runtime (s)		
			MILP-0	MILP-1	Ours	MILP-0	MILP-1	Ours	MILP-0	MILP-1	Ours	MILP-0	MILP-1	Ours	MILP-0	MILP-0	Ours
design1	33	200	22	18	19	232	268	260	4601	6949	5283	1316	1316	1316	0.01	0.26	0.01
		300	12	7	7	160	156	168	1164	2490	1362	1316	1316	1316	0.01	0.18	0.01
design2	79	200	63	56	57	944	960	812	23139	37664	24396	1676	1676	1676	0.03	1.22	0.01
		300	41	17	17	560	600	512	6566	6883	7225	1676	1676	1676	0.03	1.04	0.01
design3	127	200	107	92	101	2152	3644	1704	56616	151863	63804	1658	1658	1658	0.09	3.13	0.03
		300	56	27	30	1188	1504	932	15657	42997	17531	1658	1658	1658	0.08	2.78	0.04
design4	336	200	219	169	182	4956	8108	4000	252256	408669	260451	2881	2916	2881	0.38	32.42	0.14
		300	156	80	88	3692	5212	2824	142105	248454	153667	2881	2905	2881	0.32	32.93	0.14
design5	518	200	309	241	262	7132	11120	5864	344504	495795	360058	2792	2792	2792	0.97	120.7	0.41
		300	168	80	87	4892	7376	3840	163610	285437	170360	2792	2864	2792	0.81	183.3	0.41
Ave. & ratio	222	200	144	115	124	3083	4820	2528	136223ps	220188ps	142798ps	2065ps	2072ps	2065ps	0.30s	31.55s	0.12s
			1.16	0.93	1	1.22	1.91	1	0.95	1.54	1	1.00	1.00	1	2.50	262.92	1
		300	87	42	46	2098	2970	1655	65820ps	117252ps	70029ps	2065ps	2084ps	2065ps	0.25s	44.05s	0.12s
			1.89	0.91	1	1.27	1.79	1	0.94	1.67	1	1.00	1.01	1	2.08	367.08	1

八、結論與未來展望

我們提出了一個負載平衡式的時脈樹合成器，並搭配後端最佳化方式。包含了數個新的技術去解決 MDSV 設計的時脈樹合成問題。我們的負載平衡式時脈樹建立方法可以建立一個負載平衡的時脈樹，並在時脈樹裡加入緩衝器。而後端最佳化方式會加入 ADB 並指派有效的延遲時間，藉此減少 MDSV 設計的時序差異與面積使用問題。對於實際的 MDSV 設計來說，實驗結果顯示了在時序差異、面積，以及演算法的執行時間方面，我們的方法是相當有效果且有效率。

參考文獻

- [1] T. H. Chao, Y. C. Hsu, J. M. Ho, K. D. Boese, and A. B. Kahng, "Zero skew clock routing with minimum wirelength," *IEEE Trans. on Circuits and Systems*, vol. 39, no. 11, pp. 799-814, 1992.
- [2] Y. P. Chen and D. F. Wong, "An algorithm for zero-skew clock tree routing with buffer insertion," in Proc. European Design and Test Conference, pp. 230-236, 1996.
- [3] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," in Proc. ACM/IEEE Design Automation Conference, pp. 612-616, 1993.
- [4] M. Edahiro, "An efficient zero-skew routing algorithm," in Proc. ACM/IEEE Design Automation Conference, pp. 375-380, 1994.
- [5] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.
- [6] T. K. Johnston, "Clock tree adjustable buffer," U. S. Patent, no. 7571406B2, 2009
- [7] I. M. Liu, T. L. Chou, A. Aziz, and D. F. Wong, "Zero-skew clock tree construction by simultaneous routing, wire sizing and buffer insertion," in Proc. Intl. Symposium on Physical Design, pp. 33-38, 2000.
- [8] C. L. Lung, Z. Y. Zeng, C. H. Chou, and S. C. Chang, "Clock skew optimization considering complicated power modes," Proc. Design Automation and Test in Europe, pp. 1474-1480, Nov. 2010.
- [9] A. D. Mehta, Y. P. Chen, N. Menezes, D. F. Wong, and L. T. Pileggi, "Clustering and load balancing for buffered clock tree synthesis," Proc. IEEE International Conference on Computer Design, pp. 217-223, Oct. 1997.
- [10] G. N. Roberts, "Adjustable buffer driver," U. S. Patent, no. 5361003, 1994.
- [11] R. S. Shelar, "An efficient clustering algorithm for low power clock tree synthesis," Proc. International Symposium on Physical Design, pp. 181-188, March 2007.
- [12] Y. S. Su, W. K. Hon, C. C. Yang, S. C. Chang, and Y. J. Chang, "Value assignment of adjustable delay buffers

for clock skew minimization,” Proc. International Conference on Computer Aided Design, pp. 535-538, Nov. 2009.

- [13] R. S. Tsay, “An exact zero-skew clock routing algorithm,” IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 12, no. 2, pp. 242-249, Feb. 1993.
- [14] J. L. Tsai, T. H. Chen, and C. C. P. Chen, “Zero skew clock tree optimization with buffer insertion/sizing and wire sizing,” IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 23, no. 4, pp. 565-572, 2004.

研究成果與相關論文發表

1. Kuan-Yu Lin, Hong-Ting Lin, and Tsung-Yi Ho, and Chia-Chun Tsai, “Load-Balanced Clock Tree Synthesis with Adjustable Delay Buffer Insertion for Clock Skew Reduction in Multiple Dynamic Supply Voltage Designs,” *ACM Trans. on Design Automation of Electronic Systems*, Vol. 17, No. 3, Article 34., June 2012
2. Chia-Chun Tsai and Trong-Yen Lee, “Power Awareness for Multi-voltage Island X-Clock Tree Construction with Double-via Insertion,” *The 4th Asia Symposium on Quality Electronic Design*, July 10-11, 2012.
3. Chia-Chun Tsai, Chung-Chieh Kuo, Feng-Tzu Hsu, and Trong-Yen Lee, “Discharge-Path-Based Antenna Effect Detection and Fixing for X-Architecture Clock Tree,” *Integration, the VLSI Journal*, Vol. 45, No. 1, pp. 76-90, Jan. 2012.
4. Chia-Chun Tsai, Chung-Chieh Kuo, and Trong-Yen Lee “Post-Routing Double-Via Insertion for X-Architecture Clock Tree Yield Improvement,” *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, Paper No. 2010EAP1166, 2011. (SCI)
5. Chia-Chun Tsai, Chung-Chieh Kuo, and Trong-Yen Lee “High Performance Buffered X-Architecture Zero-Skew Clock Tree Construction with Via Delay Consideration,” *International Journal of Innovative Computing, Information and Control*, Vol. 7, No. 9, pp. 5145-5161, Sept. 2011. (SCI)
6. Chia-Chun Tsai, Sheng-Bin Dai, and Trong-Yen Lee, “The RF Circuit Design of Power and Data Contactless Transmission for ISO/IEC 14443-2 Type B,” *Journal of Circuits, Systems, and Computers*, Vol. 20, No. 8, pp. 1637-1658, Aug. 2011. (SCI)
7. Chung-Chieh Kuo, Chia-Chun Tsai, and Trong-Yen Lee, “Pattern-matching-based X-Architecture Zero-skew Clock Tree Construction with X-Flip Technique and Via Delay Consideration,” *Integration, the VLSI Journal*, Vol. 44, No. 1, pp. 87-101, Jan. 2011. (SCI)
8. Chia-Chun Tsai, Chung-Chieh Kuo, and Trong-Yen Lee, “Jumper Insertion for Antenna Avoidance in X-clock Routing,” *Far East Journal of Electronics and Communications*, Vol. 4, No. 2, pp. 123-132, June 2010. (EI)
9. Chia-Chun Tsai and Trong-Yen Lee, “Power Awareness for Multi-voltage Island X-Clock Tree Construction with Double-via Insertion,” *The 4th Asia Symposium on Quality Electronic Design*, pp. 187-192, July 10-11, 2012, Penang, Malaysia. (EI)
10. Chia-Chun Tsai, Tsung-Ming Liu, and Trong-Yen Lee, “Micro Fuel Cell Power Management Circuit Design for Portable Devices,” *International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 2506-2509, May 29-31, 2012, Chongqing, Mainland China. (EI)
11. Chia-Chun Tsai, Chung-Chieh Kuo, and Trong-Yen Lee, “Voltage-Island Aware X-Clock Tree Construction for Power Minimization,” *International Conference on Computer Science and Service System*, pp. 4132-4135, June 27-29, 2011, Nanjing, Mainland China. (EI)
12. Chia-Chun Tsai, Chung-Chieh Kuo, Feng-Tzu Hsu, Lin-Jeng Gu, and Trong-Yen Lee, “X-Architecture Zero-Skew Clock Tree Construction with Performance and DFM Considerations,” *International SOC Design Conference*, pp. 294-297, Nov. 22-23, 2010, Incheon, Korea. (EI)
13. Chia-Chun Tsai, Chung-Chieh Kuo, Lin-Jeng Gu, and Trong-Yen Lee, “Double-Via Insertion for Improving the Reliability of X-Architecture Clock Tree,” *The 21st VLSI Design/CAD Symposium*, August 3-6, 2010, Kaohsiung, Taiwan. (Best Paper Nominee)

出席國際會議研討心得報告

南華大學資工系 蔡加春 教授

國科會專題計畫補助：NSC 101-2221-E-343-004，2012/8/1~2013/7/31

結合可調整延遲時間緩衝器及兼顧負載平衡與最小時序偏移之時脈樹合成方法
應用於動態式多重供應電壓佈局環境(II)

Clock Tree Synthesis Associated with Adjustable Delay Buffers for Load Balancing
and Skew Minimization in Multiple Dynamic Supply Voltages (II)

● 2012 年亞洲版品質電子設計國際研討會 ASQED

品質電子設計國際研討會 (ISQED---International Symposium on Quality Electronic Design) 在美國已舉辦十三屆了，它是一個高品質又專業的技術研討會，提供給此領域的業界、學術界及資訊技術系統應用者等經驗交流的機會，且引起熱烈的迴響與好評。2009 年起第一次建立亞洲版 ISQED，即 ASQED---Asia Symposium on Quality Electronic Design，並於當年 7 月 15-16 日在馬來西亞吉隆坡 [Crowne Plaza Mutiara Kuala Lumpur](#) 舉行第一屆 ASQED；而後每年在馬來西亞檳城(Penang)與吉隆坡(Kuala Lumpur)兩地輪流舉辦，今年 2012 年又輪到在檳城(Penang)舉辦，於 Queensbay Estin Hotel 舉行。第四屆 [亞洲版品質電子設計國際研討會\(ASQED 2012\)](#) 是在橋接 electronic design tools and processes, integrated circuit technologies, processes & manufacturing 之間間隙去完成高設計品質。ASQED'12 舉辦單位為 the International Society for Quality Electronic Design with technical sponsorship from several IEEE Societies. 主要共同承辦與管理單位為 SHRDC. 次要共同承辦單位為 Malaysian Industry-Government Group for High Technology (MIGHT) and Malaysian Institute of Microsystems (MIMs)，含來自 from Malaysian Industrial Development Authority (MIDA)的支持及工業界的參與如 Synopsys, MIMOS, Altera, Cadence, etc.. ASQED'12 特別強調創新與最近研發，尤其在 System and IC Design, MEMS & NEMS, Semiconductor Technology & Manufacturing, IC 封裝 & PCB Technology, Test, and Bio & Nano Electronics.

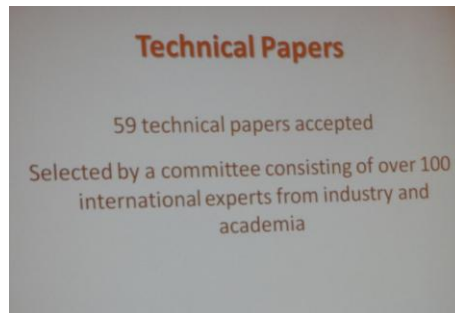
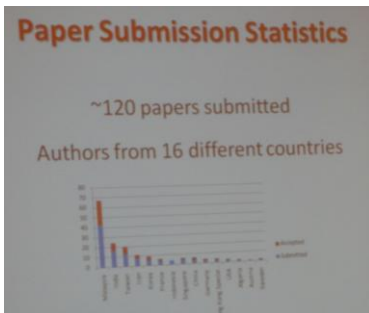


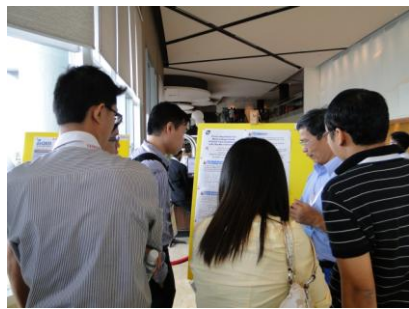
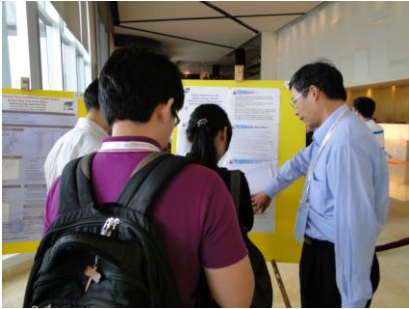
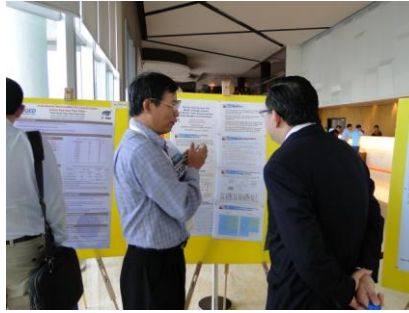
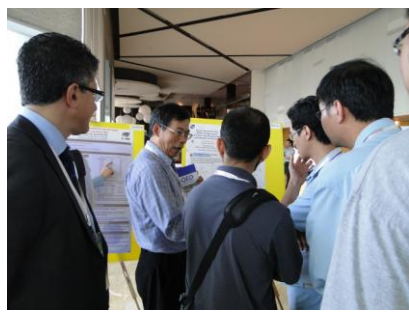
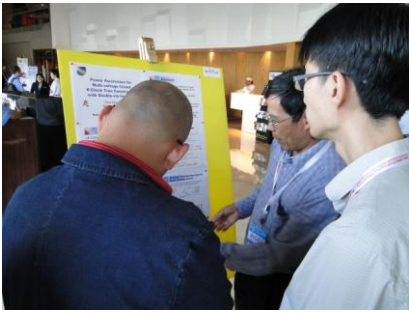
ASQED'12 舉辦兩天，2012/7/10-11, in 3 parallel tracks, hosting over 44 technical presentations, one Poster for two days, 4 Keynote speeches, and 5 Tutorials. 研討會所有論文由 IEEE 出版 and 置於 the digital library.

本研討會為來自 16 個國家投稿論文為 120 篇，僅接受 59 篇為口頭報告與海報，接受率約為 48%。國內來自各公私立大學在此會議中總計發表論文八篇，其中 [Session 2A Device Modeling and Automation for Physical Design](#)，四篇口頭報告全由台灣包辦：[2A.1 Clock Tree Construction Using Gated Clock Cloning](#), [Wun-Han Chen¹](#), [Hsin-Hung Chang¹](#), [Jui-Hung Hung²](#), [Tsai-Ming Hsieh¹](#) ¹[Department of Information and Computer Engineering](#), ²[Department of Electronic Engineering](#), 中原大學。 [2A.2 A Bias-Driven Approach to Improve the Efficiency of Automatic Design Optimization for CMOS OP-Amps](#), [Ya-Fang Cheng](#), [Li-Yu Chan](#), [Yen-Lung Chen](#), [Yu-Ching Liao](#), [Chien-Nan Jimmy Liu](#) 國立中央大學。 [2A.3 A Robust Incremental Power Grid Analyzer by Macromodeling Approach and Orthogonal Matching Pursuit](#), [Yi-Hsuan Lee¹](#), [Yu-Min Lee¹](#), [Liang-Chia Cheng²](#), [Yen-Tang Chang³](#) ¹[Department of Electrical and Computer Engineering](#), 國立交通大學, ²[Information and Communications Research Laboratories](#), 工業技術研究院, ³[Bureau of Standards, Metrology and Inspection, M.O.E.A., Taiwan](#)。 [2A.4 Analyzing BTI Effects on Retention Registers](#), [Yao-Te Wang](#) and [Ing-Chao Lin](#) 國立成功大學。

本人有一篇論文以海報方式在會中發表，隸屬於 EDA 領域，藉此與國內外學者交流，有助於增加南華大學的曝光率，來參與面對面討論者非常踴躍 (如圖片所示)。

[Chia-Chun Tsai](#) and [Trong-Yen Lee](#), “Power Awareness for Multi-voltage Island X-Clock Tree Construction with Double-via Insertion,” *The 4th Asia Symposium on Quality Electronic Design*, pp. 187-192, July 10-11, 2012, Penang, Malaysia.





檳城(Penang)的喬治亞市區(George Downtown)已被聯合國列為世界遺產，藉此機會參訪檳城一些古蹟，如 Chew Jetty (姓周橋)、Snake Temple (蛇廟)、Floating Masjid (水上清真寺)、Penang Hill(升旗山)、Little India (小印度區)、Tailand 寺 Sleeping Buddha (臥佛)、緬甸寺大佛及百年菩提樹。





此行參加研討會，與來自世界各地之國際學者及業界相互交流，藉此了解他們研究方向與成果，並帶回大會相關資料及論文光碟片，學術之旅收穫豐碩。有關大會註冊費、機票與生活費方面等，感謝國科會專題計畫補助及南華大學支援部分補助，共同鼎力支持學術論文發表。

● 2013 年自然計算及模糊系統與知識發現國際研討會

ICNC & FSKD'13, July 23-25, 2013, Shenyang, Liaoning, Mainland

自然計算國際研討會(2013 9th International Conference on Natural Computation, ICNC)及模糊系統與知識發現國際研討會(2013 10th International Conference on Fuzzy Systems and Knowledge Discovery, ICFSKD)相結合成為一個高品質又多元專業的交流平台技術研討會，它提供給此領域的業界、學術界、研究人員、工程師及服務系統技術應用者等經驗交流的機會。ICNC2013 & ICFSKD2013 的舉辦單位為 IEEE Circuits and Systems Society，承辦單位為大陸遼寧工程技術大學、東北大學等。

Organizer:			Technical Co-Sponsor :	
				

今年於 2013 年 7 月 23 日至 25 日在遼寧省瀋陽市遼寧政協會館(Liaoning Zhengxie Hotel，皇姑區崇山東路 73 号，距地鐵 2 號線中醫藥大學站 800 米) 國際會議廳舉行，會議包含下列幾個主題為 Neural Computation、Enabling of Systems Biology Concepts as Software Intensive Complex Systems、Complex Systems and Dependability、Usability of Complex Systems、Evolutionary Computation、Fuzzy Theory and Algorithms、Knowledge Discovery Foundations、Information Technology & Computer applications 等。



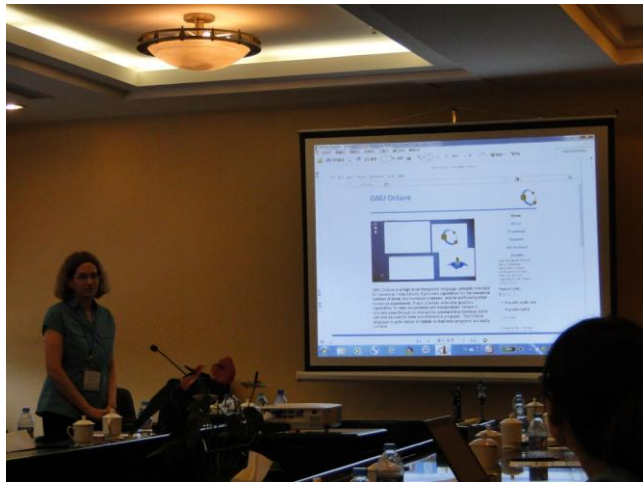
所有錄取的 papers 都可在 IEEE Xplore 被檢索 in Ei Compendex. 被選擇較好 papers 擴充版可發表在的 ICNC-FSKD special issue of International Journal of Intelligent Systems, an SCI-indexed journal with an Impact Factor 1.653 。

ICNC2013 & ICFSKD2013 國際研討會主要包括三場 Keynote speeches 及 16 場 Oral Sessions 的發表，及另有四場的 Poster 的展示等。

Title	Keynote Speaker	Affiliation
Fuzzy Multi-Attribute Decision Making Algorithm for Vehicle Routing in Personalized Intelligent Vehicle Assistance Environment	Prof. Philip Chen 	University of Macau IEEE Fellow; President, IEEE Systems, Man, Cybernetics Society
Processing the Imprecision and Ambiguity in Medical Image Analysis	Prof. Pau-Choo (Julia) Chung 	National Cheng Kung University IEEE Fellow, Distinguished Professor
Development of Fuzzy Cerebellar Model Neural Network	Prof. Chih-Min Lin 	Yuan Ze University IEEE Fellow, Chair Professor



在會場相關的活動包含 Coffee break 及 7/23 晚上的 Banquet 等，可多與一些人互動機會，含青島中國海洋大學信息科學與工程學院的馮晨博士、廣東中山大學信息科學與技術學院的李中華博士、及馬來西亞 Taylor University 的 Hong Jer Lang 博士與其一位大四專題生等。

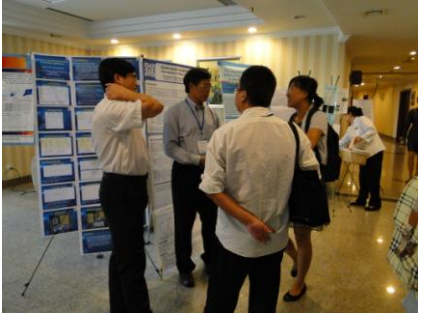
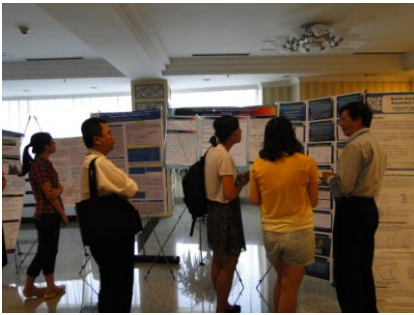
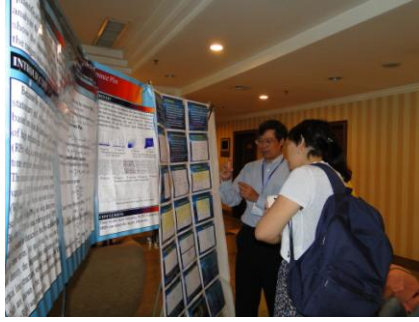
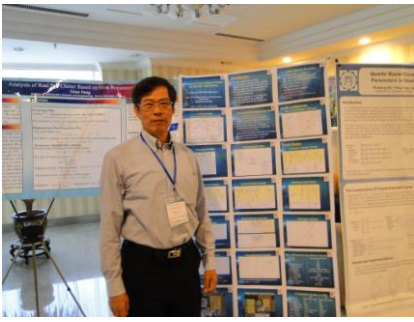


參與聆聽 Session 的報告

本人有一篇論文，並選擇海報展示有更多得互動與討論，同時也聽了幾場 sessions，其中一位學者研究 paper review 的 mistake rate，因專業背景的不同而 mistake rate 有逐年提升的趨勢(如上圖左下角圖片所示)，這是一個很有趣的問題。



晚宴中與各地學者及研究生合影



論文發表的情況

其中來自馬來西亞 Taylor University 的很年輕博士 Hong Jer Lang (只會講英文) 發表六篇與其一位大四專題生發表三篇，Hong 博士告訴我：「他每天只睡 6 小時，大部分時間都在很認真工作，他現在主持六個 Projects、帶 24 個學生，含 6 個博士生」，Hong 博士再告訴我：「他累積做了很多實務的小計畫與組成的模組與元件(並 show 電腦內的畫面讓我看)，可幫政府單位做很多事，因此可順利接政府一些 Projects」，令我佩服他的累積實務實力與一股拼勁及 Work very hard。

與青島中國海洋大學信息科學與工程學院及與遼寧大學等兩位年輕博士(青島那位為馮晨博士)，她們告訴我：「一星期只上課四小時及協助一些行政事務，其他時間均運用來做研究」，年新接近於十萬元人民幣，這是給年輕博士衝刺研究的機會。另東北大學女研究生，異口同聲：「父母親權力給她們資源，盡力在培養她們」，目的就是盡力在培育她以後有好工作與好未來，她們也都自信滿滿，絕不讓父母失望。

此行參加研討會，也參觀了一些瀋陽一些歷史景點，經歷了歷史文化之旅。瀋陽是遼寧省會，著名的工業重鎮和歷史文化名城，幅員面積近 1.3 萬平方公里，建城區面積 430 平方公里，轄 9 區、3 縣、1 市和 3 個國家級開發區，常住人口 822 萬。瀋陽因地處渾河（古稱沈水）北岸而得名。從西元前 300 年燕國大將秦開在此建立侯城算起，瀋陽已有 2300 餘年建城史，曾有沈州、瀋陽中衛等名稱。瀋陽素有「一朝發祥地，兩代帝王都」之美譽。1625 年，清皇太祖努爾哈齊遷都於此，更名為盛京。1636 年，皇太極在此改國號為「清」，建立清王朝。1644 年，清軍入關定都北京後，確定盛京為陪都，1651 年，在此設立奉天

府。1945 年，抗日戰爭勝利後，復稱瀋陽市。近年來，瀋陽市牢牢抓住國家振興東北地區等老工業基地戰略，深入實施，瀋陽經濟區被確定為國家新型工業化綜合配套改革試驗區、第十二屆全運會主會場設在瀋陽的重大機遇，緊緊圍繞做優發展空間、做大中心城市、做強縣域經濟、加強生態建設和著力改善民生的「五大任務」，建設國家中心城市、先進裝備製造業基地、生態宜居之都「三大目標」，大力推動經濟社會又好又快發展。2012 年，全市地區生產總值 6606.8 億元，公共財政預算收入 715 億元，城市居民人均可支配收入 26431 元，農民人均純收入 13045 元，分別比上年增長 10%、15.3%、13.3%、12.7%。

瀋陽是中國東北地區最大的中心城市和交通樞紐，素有「東方魯爾」美譽。全市工業門類已達 165 個，形成了以機械加工為主，包括汽車、石化、航空、製藥、建材、冶金、輕工、紡織、電子、煤炭等行業在內的門類齊全的工業體系。瀋陽擁有國內最大的鐵路編組站和東北地區密度最大的公路網，與國際 16 個城市、國內 59 個城市（地區）通航，擁有 130 條航線。2006 年，瀋陽被批准為臺胞口岸簽注點。2009 年，瀋陽至臺北實現了直航常態化，目前有四家航空公司每週往返 22 個航班班次，每天都有航班在瀋臺兩地間往返，航程僅兩個多小時，每年來瀋陽臺胞達 5 萬多人次。瀋陽旅遊資源豐富，既有清故宮、福陵、昭陵三處世界文化遺產，也有張氏帥府博物館、九一八歷史博物館、怪坡和棋盤山冰雪大世界等獨具特色的著名景區。



清太祖努爾哈赤福陵 遺址



清太極昭陵 遺址



瀋陽 東北虎園



九一八博物館



瀋陽故宮博物館



張大帥府 (張學良將軍)



瀋陽金融博物館

此行參加研討會，與來自世界各地之國際學者及業界相互交流，藉此了解他們研究方向與成果，並帶回大會相關資料及論文光碟片，學術之旅收穫豐碩。有關大會註冊費、機票與生活費方面等，感謝國科會專題計畫補助及南華大學在行政上的支援，共同鼎力支持學術論文發表。

國科會補助計畫衍生研發成果推廣資料表

日期:2013/08/27

國科會補助計畫	計畫名稱: 結合可調整延遲時間緩衝器及兼顧負載平衡與最小時序偏移之時脈樹合成方法應用於動態式多重供應電壓佈局環境 (II)
	計畫主持人: 蔡加春
	計畫編號: 101-2221-E-343-004- 學門領域: 積體電路及系統設計
無研發成果推廣資料	

101 年度專題研究計畫研究成果彙整表

成果項目		量化			單位	備註 (質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等)
		實際已達成數 (被接受或已發表)	預期總達成數 (含實際已達成數)	本計畫實際貢獻百分比		
國內	論文著作	期刊論文	0	0	100%	篇
		研究報告/技術報告	0	0	100%	
		研討會論文	1	1	100%	
		專書	0	0	100%	
	專利	申請中件數	0	0	100%	件
		已獲得件數	0	0	100%	
	技術移轉	件數	0	0	100%	件
		權利金	0	0	100%	千元
	參與計畫人力 (本國籍)	碩士生	0	0	100%	人次
		博士生	0	0	100%	
		博士後研究員	0	0	100%	
		專任助理	0	0	100%	
國外	論文著作	期刊論文	2	2	100%	篇
		研究報告/技術報告	0	0	100%	
		研討會論文	2	2	100%	
		專書	0	0	100%	章/本
	專利	申請中件數	0	0	100%	件
		已獲得件數	0	0	100%	
	技術移轉	件數	0	0	100%	件
		權利金	0	0	100%	千元
	參與計畫人力 (外國籍)	碩士生	1	1	100%	人次
		博士生	0	0	100%	
		博士後研究員	0	0	100%	
		專任助理	0	0	100%	

計畫主持人：蔡加春

計畫編號：101-2221-E-343-004-

計畫名稱：結合可調整延遲時間緩衝器及兼顧負載平衡與最小時序偏移之時脈樹合成方法應用於動態式多重供應電壓佈局環境 (II)

<p>其他成果 (無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。)</p>	<p>1. 擔任第八屆 2013 國際健康資訊管理研討會副主席。 2. 協助國科會大專學生參與專題研究線上審查七件。</p>
--	--

	成果項目	量化	名稱或內容性質簡述
科 教 處 計 畫 加 填 項 目	測驗工具(含質性與量性)	0	
	課程/模組	0	
	電腦及網路系統或工具	0	
	教材	0	
	舉辦之活動/競賽	0	
	研討會/工作坊	0	
	電子報、網站	0	
	計畫成果推廣之參與(閱聽)人數	0	

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

達成目標

未達成目標（請說明，以 100 字為限）

實驗失敗

因故實驗中斷

其他原因

說明：

2. 研究成果在學術期刊發表或申請專利等情形：

論文： 已發表 未發表之文稿 撰寫中 無

專利： 已獲得 申請中 無

技轉： 已技轉 洽談中 無

其他：（以 100 字為限）

重要成果: Kuan-Yu Lin, Hong-Ting Lin, and Tsung-Yi Ho, and Chia-Chun Tsai, 'Load-Balanced Clock Tree Synthesis with Adjustable Delay Buffer Insertion for Clock Skew Reduction in Multiple Dynamic Supply Voltage Designs,' ACM Trans. on Design Automation of Electronic Systems, Vol. 17, No. 3, Article 34,, June 2012

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）（以 500 字為限）

對目前的奈米積體電路設計來說，功率消耗是個很重要的影響。對於這個問題，動態式的多重供應電壓(multiple dynamic supply voltage, MDSV)設計方法是個有效的解決方法。然而，在功率模式切換的時候，會增加時序差異的改變，進而增加修正時序差異的困難度。在此，我們提出了一個負載平衡式的時脈樹合成器，並加入可調整延遲時間的緩衝器(adjustable delay buffer, ADB)，讓使用 MDSV 設計方法的晶片能有最小的時序差異。一旦初步的時脈樹已建立完成，我們會逐層的加入 ADB 並指派它本身的延遲時間，去減少 MDSV 時脈樹的時序差異。由於 ADB 可以被用來產生額外的延遲時間，因此能調整時脈樹的延遲時間與時序差異。最後，對於 MDSV 的時脈樹，我們提出了一個有效的演算法，以最少的執行時間加入 ADB 去改善時序差異，並讓 ADB 消耗最少的面積。相較於目前最好的加入 ADB 演算法，實驗結果顯示了我們的方法能減少 42.40%的 ADB 面積以及加快 117 倍的執行時間。同時再以整數線性規劃方法來評估我們的演算法，MILP-0 以對每一個 ADB 延遲之最小化為目標，其所有 ADB 數的總延遲較功率模式導向時脈樹最佳化結果平均少 5%，但時脈延遲仍然與功率模式導向時脈樹最佳化結果相同；MILP-1 以對插入 ADB 總數之最小化為目標，其所有插入 ADB 總數較功率模式導向時脈樹最佳化結果平均少 8%，但時脈延遲仍然與功率模式導向時脈樹最佳化結果相同。實驗結果證明了我們的方法確實達到最佳化。

