

行政院國家科學委員會專題研究計畫 成果報告

針對晶片系統連接網路之驗證與自動合成之研究(3/3)

計畫類別：個別型計畫

計畫編號：NSC93-2215-E-009-011-

執行期間：93年08月01日至94年07月31日

執行單位：國立交通大學電子工程學系暨電子研究所

計畫主持人：周景揚

報告類型：完整報告

報告附件：國際合作計畫研究心得報告

處理方式：本計畫可公開查詢

中 華 民 國 94 年 9 月 26 日

針對晶片統連接網路之驗證與自動合成之研究 (3/3)

“The Study of Verification and Synthesis for Interconnect Networks on System-on-a-Chip”

計畫編號：NSC 93-2215-E-009-011

執行期間：93 年 8 月 1 日 至 94 年 7 月 31 日

主持人：周景揚 交通大學電子工程系教授

一、中文摘要

由於半導體技術的突飛猛進，單晶片系統（SoC）已經變成了二十一世紀裡的主流科技，藉由單晶片系統的技術，可以加速促成三 C（電腦、通訊、民生家電）之間的整合。這不管對企業或是科技來說，都是一股不可避免的潮流。對整個人類社會來說，三 C 的整合應用更是一個非常重要的進步。

要支援單晶片系統技術，除了晶片設計的方法必需改變之外，計算機輔助設計（CAD）的軟體也必須加緊發展。由於單晶片系統設計的複雜度極高，驗證（verification）成為整個晶片發展流程中十分重要的一環。為了加速設計流程，時下單晶片系統多採取平台式(platform-based)的設計，亦即矽智產元件(Intellectual Properties, IPs)只要針對特定界面協定(protocol)來設計，就能很快地整合到平台式的單晶片系統中。所以對於矽智產元件是否吻合特定的通訊協定則成為驗證流程中的首要目標。另一方面，為了驗證單晶片系統整合後的功能是否正常，我們使用抽象化的連接埠順序障礙模型。連接埠順序障礙模型假設由晶片設計者所提供的矽智產元件是已經設計好且驗證好的。因此，在整合的過程中我們針對矽智產連接介面進行驗證工作，以達成完整的驗證成果。

在本計畫中，我們針對了單晶片系統中，介面連接時所可能發生的錯誤，提供了完整的驗證機制，一是針對驗證單一矽智產元件與所對應界面協定間的相容性，我們提出了一個正規驗證的演算法，得以有效率地來完成此項任務。而針對接線介面驗證，我們則提出了用以偵查並修正單晶片系統設計核心間接線介面連接錯誤的自動驗證向量產生器（AVPG）及自動介面接線修正器（AIR）。藉以達到完整且有效率的介面協定與連接上的驗證與快速除錯。

關鍵字

單晶片 矽智產 連接埠順序障礙模型 連接介面驗證 自動驗證向量產生器

英文摘要

With the advent of semiconductor technology, the System-on-a-chip (SoC) becomes a mainstream and focus of technologies on coming 21st century. The integration of Computing, Communication and Consumer Electronics will be speeded up through SoC, which causes the inevitable trend of new era for technology and enterprises. Especially, the application of 3C integration is very important to the progress of human being society.

To support SoC, the design methodologies have to be refined and several new CAD tools have to be developed. Due to the high complexity of SoC designs, verification has become the major bottleneck of the entire design process. There is an emerging need for a practical solution to verify large SoC designs. First, we need to verify the interface compliance of the IP before it is integrated into an SOC design. Next, to verify the functionality of a SoC, the port order fault model (POF) is used for abstraction. The POF model assumes that the Intellectual Properties (IPs) provided from the core providers are pre-designed and pre-verified. During this process, system integrators should focus on the interconnections and interface protocols verification instead.

In this project, we provide a complete solution for interface and interconnection verification. First, we propose a formal algorithm checking if the IP is designed correctly with target interface protocol. This algorithm can efficiently finish the interface compliance verification task. Then, for the interconnection verification issue, we develop an automatic verification pattern generation and automatic interconnection rectification generating the verification patterns to detect the port misplacements among the cores in a SoC and then to correct these misplacements quickly.

Keywords:

SOC, IP, POF, port order fault model, interconnection verification, AVPG, automatic verification pattern generation

二、計畫的緣由與目的

不斷進步的矽製程技術快速地縮小了可以實現於積體電路(integrated circuit)上的矽結構的實體尺寸，這個矽結構實體尺寸的縮小帶來了電路容量的增加及電路效能的重大改善。儘管先進的製程技術讓晶片的複雜度(complexity)增加，然而晶片設計者的產能成長率卻遠遠落後於晶片複雜度的成長率。此二種成長率之間的差距將隨著製程技術的進步而日趨擴大。因此單晶片系統(System-on-a-chip)的概念 - 將一個完整系統整合進單一晶片中，就被提出來以使此差距可以縮短。

單晶片系統時代的主要設計方法可以大約區分為以方塊為基礎的設計方法(block-based design)，以及以平台為基礎的設計方法(platform-based design)，這二種設計方法分別有著不同的設計理念。以方塊為基礎的設計方法著重在功能方塊的重複使用，而以平台為基礎的設計方法則著重在平台結構的重複使用，然而這兩種方法都同樣利用了設計重複使用(design reuse)的技巧來回應當今高複雜設計的挑戰，即是藉著利用已經有的功能方塊或設計平台，來實現一個新的設計。如此一來，所需花費的設計工作將可以大量減少。然而現在的設計方法尚無法快速的處理來自於不同群體間的設計核心，其彼此間的匹配互動問題，尤其是驗證整個設計是否完全滿足規格是最困難的問題之一，也是整個單晶片系統設計過程的瓶頸，我們急需實際的解決方案來驗證大型的單晶片系統設計。

我們稱那些在單晶片系統設計中被用到的，且已事先設計好及驗證好的元件或功能方塊為矽智產。從對矽智產的使用觀點來看，積體電路設計群體可以被分為兩個小群體，一個是矽智產供應者群體，一個是矽智產使用者或稱為系統整合者群體。一般而言，單晶片系統設計中被用到的矽智產可能是來自於自己公司內部或是由矽智產供應者所授權使用的。

單晶片系統的系統整合者從供應者手中拿到的元件不再是積體電路元件，而是矽智產，利用各個擊破技巧(divide and conquer)，單晶片系統設計的驗證工作可以被簡化成兩個步驟，分別是基本功能方塊驗證及整合驗證(integration verification)。當每一個步驟的驗證工作都順利完成後，整個單晶片系統設計的驗證工作也同時完成了。因此我們假設在單晶片系統設計中所使用的矽智產都是已經被驗證過，同時無設計錯誤存在的，於是在整合過程中，它們是不需要再被驗證的。如此一來單晶片系統設計的驗證工作就只剩下整合驗證，也唯有如此才有可能在合理的時間成本內讓單晶片系統設計順利地進入市場。

隨著矽智產的日趨複雜 - 動輒上百或甚至上千個輸入輸出埠的矽智產已經愈來愈普遍了，於是單晶片系統內矽智產間的接線驗證工作的重要性正與日俱增。系統整合者由於缺乏對來自於不同設計群體的矽智產間的接線關係有足夠的認識及資訊，所以在整合時發生矽智產彼此間接線錯誤的機會是很高的，因此在單晶片系統整合過程中，矽智產彼此間的接線驗證是必須的，同時此接線驗證也是矽智產彼此間互動驗證的第一步，因為如果發生接線錯誤而未察覺到，就直接做矽智產彼此間

的互動驗證，那麼此互動驗證將很有可能是白忙一場的。因為當察覺出有錯誤時，我們會以為是矽智產彼此間的互動出錯了，而一直專注在其上之除錯，而這很明顯的是找錯了錯誤真正發生的原因。所以我們應該先作矽智產間的接線驗證，待其完全無誤時，再作矽智產間的互動驗證。

本計畫的目標主要針對單一矽智產的界面協定驗證，及整個單晶片系統的接線驗證。我們首先提出了一個以有限狀態機(FSM)為基礎的正規驗證方式來檢查界面協定的功能是否正確，在整合階段，我們發展了自動產生驗證向量的演算法用於矽智產之間的接線驗證，並研發可以在單晶片系統的整合過程中，對以驗證向量所偵測得的錯誤介面接線連接，提供自動介面接線修正器 (AIR)，進以加速設計與驗證的速度。

三、 研究方法及成果

在本報告中，我們針對前一節所提到的界面驗證問題提出我們的解法，並簡單地說明我們的研究成果。

目前的矽智產為了可被重覆使用(reusability)及加速單晶片系統的建立，多半會針對特定的界面協定來設計。因此，驗證矽智產是否吻合所期待的界面協定便成為整合前的重要課題，為了避免傳統模擬為基礎(simulation-based)方法所可能發生的誤正判(false positive)問題，我們提出了一個正規驗證的方式來進行相容性檢查。我們的方法主要是先將界面協定的規格有系統地使用有限狀態機模型來描述，用此模型來表示設計時應具有的所有特性。另一方面，我們則將矽智產設計中與界面協定相關的部份萃取出來，這一部份的設計也是一個有限狀態機。我們將界面協定的驗證轉化為兩個有限狀態機之間的相容性問題。因為此兩個有限狀態機通常具有不等個數的狀態(state)，彼此間也往往不是一個完全包含另一個的關係，因此，不屬於傳統等同性檢查(equivalence checking)的範圍。我們針對這個問題提出了一個分支與邊界(branch-and-bound)的演算法，得以有效地找出兩有限狀態機中所有相對應的狀態，進而達到相容性驗證的目的。當驗證過程中，發生了不相容的對應關係時，我們的演算法也能提供給使用者設計錯誤的反例(counter-example)來進行除錯的工作。經由實驗，我們證明了我們的演算法應用在真實設計時的驗證是十分地具有效率，此方法也已發表於國際會議中[Yang05]。

在完成了單一矽智產的相容性驗證後，我們便可將接下來，我們則針對整合時所可能發生的接線問題及解決方法做一完整陳述。一般而言，針對功能性錯誤(functional error)，例如轉換故障(transition fault)，或針對製造故障，例如固定在某一值的故障(stuck-at-fault)，的自動向量產生器，均會事先清楚地建立起故障集合以決定有多少個故障需要被偵測，接著再產生隨機向量(random pattern)或確定向量(deterministic pattern)來偵測這個故障集合內的每個故障。然而對於針對接線錯誤的自動驗證向量產生器而言，其故障集合並不能明顯地列舉出來，這是因為對一個有 N 個輸入的矽智產來說，總共有 $N!$ 個排列，可以代表其接線的情形，我們將一個接線情形稱為一個連接埠序列(port sequence)，這 $N!$ 個連接埠序列之中，只有一個是對的，其餘 $N!-1$ 個都是錯的，我們稱它們為錯誤的連接埠序列(faulty port sequence)。這些錯誤的連接埠序列都被放入故障集合內，於是故障集合內的錯誤的連接埠序列數目將會隨著 N 的增加而急劇的成長。舉例來說，當 $N=20$ 時， $N!-1 \approx 2.4 \times 10^{18}$ ，或當 $N=69$ 時， $N!-1 \approx 1.7 \times 10^{98}$ ，所以對於針對接線錯誤的自動驗證向量產生器而言，其首要必須解決的問題是如何順利且有效的表示出在故障集合內，剩餘的未被偵測到的錯誤的連接埠序列。自動驗證向量產生器藉由一種內隱式的(implicit)未偵測連接埠序列(undetected port sequences)表示式來解決這個問題，這個內隱式的未偵測連接埠序列表示式可以在產生驗證向量的過程中指出剩餘的未被偵測連接埠序列，當此剩餘的未被偵測連接埠序列變成空的時候，就代表全部的接線錯誤均已被偵測到。從另一方面來說，自動驗證向量產生器與傳統向量產生器的不同點還有其產生向量的方法，它不用隨機向量或確定向量來當作驗證向量，而是採用啟發式(heuristic)向量，換句話說，自動驗證向量產生器是以一種有系統的方式來針對接線錯誤產生有效的向量。

我們以一個例子來說明內隱式的未偵測連接埠序列表示式。已知一個有八個輸入埠的矽智產，其輸入埠被標示成 1~8 的數字，每個數字代表一個輸入埠，未偵測連接埠序列表示式(12345678)表示在括弧中的輸入埠 1~8，其任意的接錯情形，均未被偵測到，未偵測連接埠序列的總數為 $8!-1$ ，其中的 1 代表的是正確的連接埠序列。未偵測連接埠序列表示式(125)(4)(3678)表示輸入埠 1, 2, 5 彼此之間以及輸入埠 3, 6, 7, 8 彼此之間，其任意的接錯情形，均未被偵測到，未偵測連接埠序列的總數剩下 $3! \times 1! \times 4!-1$ 。未偵測連接埠序列表示式的括弧先後排列順序以及括弧內輸入埠的先後排列順序是沒有影響的，舉例來說，未偵測連接埠序列表示式(125)(4)(3678)也可以表示成(4)(215)(8763)。未偵測連接埠序列表示式(12)(3)(4)(5)(6)(78)含有 4 個連接埠序列，它們分別為 12345678, 21345678, 12345687 以及 21345687。未偵測連接埠序列表示式(1)(2)(3)(4)(5)(6)(7)(8)有八個括弧，每個括弧內只有一個輸入埠，因此每個括弧內不會再有輸入埠接錯的情形，未偵測連接埠序列的總數為 $1! \times 1! \times 1! \times 1! \times 1! \times 1! \times 1! \times 1!-1=0$ ，因此未偵測連接埠序列表示式(1)(2)(3)(4)(5)(6)(7)(8)代表 $8!-1$ 未偵測連接埠序列都被偵測到了，於是我們可以發現，只要未偵測連接埠序列表示式從(12345678)演變為(1)(2)(3)(4)(5)(6)(7)(8)，那麼全部可能的錯誤接線都被偵測到了。

我們除了提出內隱式的未偵測連接埠序列表示式之外，還將問題轉換成求圖形自同構(graph automorphism) [Agr96][Cha95] 的問題，藉由這樣的轉換方式，我們很順利地找出一組有效的驗證向量，可以用來偵測所有可能錯誤的接線，結果相當令人滿意，已發表於國際性期刊中[Wan02]。而針對被驗證向量所偵測到的接線錯誤，我們提出自動介面接線修正的演算法(AIR Algorithm)。利用把原驗證向量中的某些位元有技巧的交換來模擬因接線不同可能導致的錯誤輸入向量，並不斷地觀察模擬結果以得知到底是在哪一組向量輸入時，模擬的結果跟正確值相同。由此來獲知正確的連線關係，然後加以修正介面接線錯誤。由這樣一個基本觀念所發展的自動介面連接修正器，初步結果十分成功，且已發表於國際性期刊中[Wan03]。此外，我們也將此向量產生方法應用在計算電路中最大對稱輸入集合(maximal set of symmetric inputs)，透過此一應用，我們可以很快地確認電路中非對稱的輸入部分，用以加強處理在邏輯驗證(logic verification)中常需要處理的輸入配對(input matching)及元件庫對應(library binding)等問題。關於此成果也已順利發表於國際性會議[Chou04]。

四、結論與討論

在本報告中，我們主要是以目前的單晶片系統設計驗證流程所需面對的界面問題，提出適當的解決方法，其一，針對整合過程前，界面協定的相容性驗證，我們提出了一套以有限狀態機為基礎的正規驗證演算法，同時，針對整合過程中可能發生的接線錯誤，我們也提出了一個結合了驗證向量生成、偵測接線錯誤、診斷錯誤出處、及修正接線錯誤的自動介面接線修正器，以上的研究成果皆已在國際會議及期刊上順利發表。

表一簡列近年本研究群的相關研究成果。89 年發表會議論文 5 篇，期刊論文 6 篇，並有 4 篇論文於 IEEE 期刊發表，90 年發表會議論文 10 篇，期刊論文 3 篇，並有 3 篇論文於 IEEE 期刊發表，91 年發表會議論文 5 篇，期刊論文 3 篇，並於 IEEE 期刊發表 2 篇論文。92 年發表會議論文 12 篇，期刊論文 4 篇，並有 3 篇論文於 IEEE 期刊發表。93 年發表會議論文 12 篇，期刊論文 3 篇，並有 3 篇論於 IEEE 期刊會議論文發表。

| Year | Number of Papers | | | | SCI |
|------|------------------|---------|---------------|----------------|-----|
| | Domestic | | International | | |
| | Conference | Journal | Conference | Journal | |
| 2000 | 0 | 0 | 5 | 6 (IEEE:4) | 6 |
| 2001 | 2 | 0 | 8 | 3 (IEEE: 3) | 3 |
| 2002 | 3 | 0 | 2 | 3 (IEEE:2) | 3 |
| 2003 | 4 | 0 | 8 | 4 (IEEE: 3) | 4 |
| 2004 | 3 | 0 | 9 | 3 (IEEE: 3) | 3 |

表一、本研究群近年相關研究成果

五、參考文獻

- [Agr96] M. Agrawal, and V. Arvind, "A Note on Decision Versus Search for Graph Automorphism," *Eleventh Annual IEEE Conference on Computational Complexity*, pp.272-277, 1996.
- [Cha95] R. Chang, W. Gasarch, and J. Toran, "On Finding the Number of Graph Automorphism," *IEEE Structure in Complexity Theory Conference*, pp.288-298, 1995.
- [Wan02] C.-Y. Wang, S.-W. Tung, and J.-Y. Jou, "An Automorphic Approach to Verification Pattern Generation for SoC Design Verification using Port-Order Fault Model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp.1225-1232, vol. 21, no. 10, Oct. 2002.
- [Wan03] C.-Y. Wang, S.-W. Tung, and J.-Y. Jou, "Automatic Interconnection Rectification for SoC Design Verification Based on the Port Order Fault Model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp.104-114, vol. 22, no. 1, Jan. 2003.
- [Chou04] C.-L. Chou, C.-Y. Wang, G.-W. Lee, and J.-Y. Jou, "Graph Automorphism-based Algorithm for Determining Symmetric Inputs," *International Conference on Computer Design ICCD'04*, October 2004.
- [Yang05] Ya-Ching Yang, Juinn-Dar Huang, Chia-Chih Yen, Che-Hua Shih, and Jing-Yang Jou, "Formal Compliance Verification of Interface Protocols," *Proc. of IEEE Int'l Symposium on VLSI Design, Automation, and Test*, April 2005.