

加速無線及動態影片處理系統 IP 演算法的驗證

Ken Karnofsky, MathWorks 公司訊號處理應用資深策略長

演算法密集系統的驗證是一個長期且昂貴的過程，許多研究結果顯示，多數嵌入式系統的缺陷和漏洞主要肇因在規格(specification)階段，但往往沒被檢測出來，直到後期開發階段才出現；這些缺陷是開發專案延遲的主要病因，而且為工程成本的耗費貢獻最多。

不過情況是可以改觀的，現在很多演算法密集系統的開發者已經知道有工具可協助他們在控制之下進行驗證，工程師們可以使用相同的工具來建立系統模型，同時又能在整個開發過程中提早發現正確的問題，如此不僅可以減少驗證時間，又能提高其設計性能。在這篇文章中，我們將介紹三個實際可行的方法來進行早期驗證，使這一切成為可能。

首先，讓我們來看看為什麼目前的演算法 IP 驗證過程效率低且容易出錯。演算法的 IP 可以區分通訊、聲音、動態影片、影像和導航功能。和現成既有的 IP 組件不同的是，演算法 IP 可以說是一個公司的“秘密武器”，本質上是為特定應用而開發的，因此需要不同的方法來驗證。

在一個典型的工作流程中，設計開始於演算法的開發者，他們將演算法做成規格文件，再將設計交付給硬體和軟體團隊進行下階段開發設計（圖 1）。

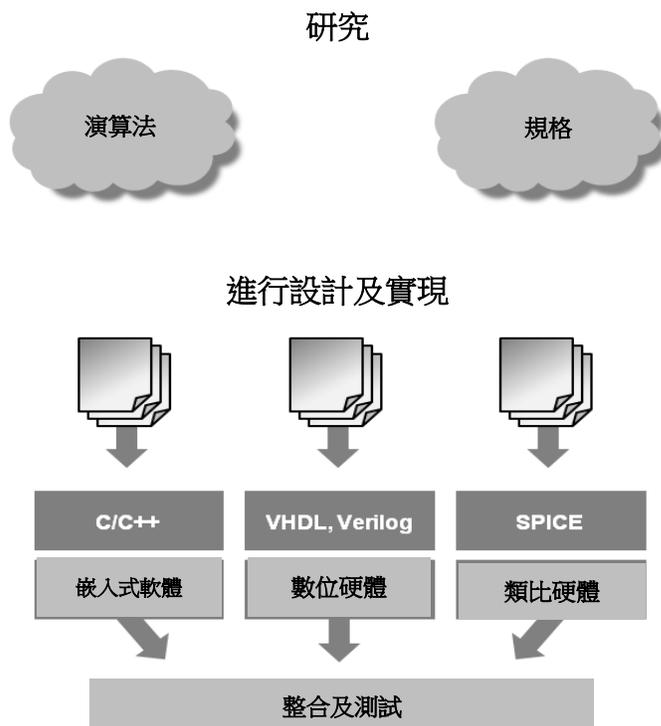


圖 1：在典型的工作流程中，設計開始於演算法的開發者，他們將演算法做成靜態規格文件，再將設計交付給硬體和軟體團隊進行下階段開發設計。

通常每個小組會有自己的測試程序以確定該設計實現後的功能是否正確，這些測試程序往往以其特設的方式架構，主要仰賴規格工程師的解釋，問題由此產生了，因為硬體和軟體工程師往往缺乏與規格應用領域知識或工具來正確地解釋和執行規格。反之亦然，演算法開發人員也可能缺乏相關的工具和專業，以確保他們已經界定出“真實世界”的要求，所以往往在開發後期，才發現演算法無法如預期在所設定的目標環境中運作。

加重 IP 演算法驗證低效率的原因，還在於軟體、數位和射頻/類比硬體元件開發都各自使用獨立的工具和工作流程，這會抑制跨領域的系統行為驗證，而導致意想不到的軟硬體交互作用。其結果是，直到工作流程結束才進行系統驗證，而在系統整合階段的設計變更往往是最困難和最昂貴的。

因為大多數的錯誤是在規格階段就已產生，傳統的硬體、軟體和 ESL 工具解決不了這些問題，傳統的工具假定設計需求(requirements)都已經充分而準確的納入規格裡。所以驗證問題必須在設計流程的一開始就加以解決，這可以透過將演算法和行為建模以及與下游的工作流程加以連結來進行。

早期驗證與模型化基礎設計(Model-Based Design)

模型化基礎設計提供了一個更好的方法，它提供了一套演算法設計、系統模擬、原型化，以及嚴謹的分析工具，使用這些工具，演算法和測試可以設計為行為系統模型的一部分，把這系統模型當成是一個可執行的規格，所有的設計團隊可以以它作為設計參考(design reference)和測試平台(test bench)。這種方法提供所有的設計團隊，包括演算法、系統架構，以及各種元件的開發團隊，對於設計需求有一個共享的、明確的瞭解（圖2）。

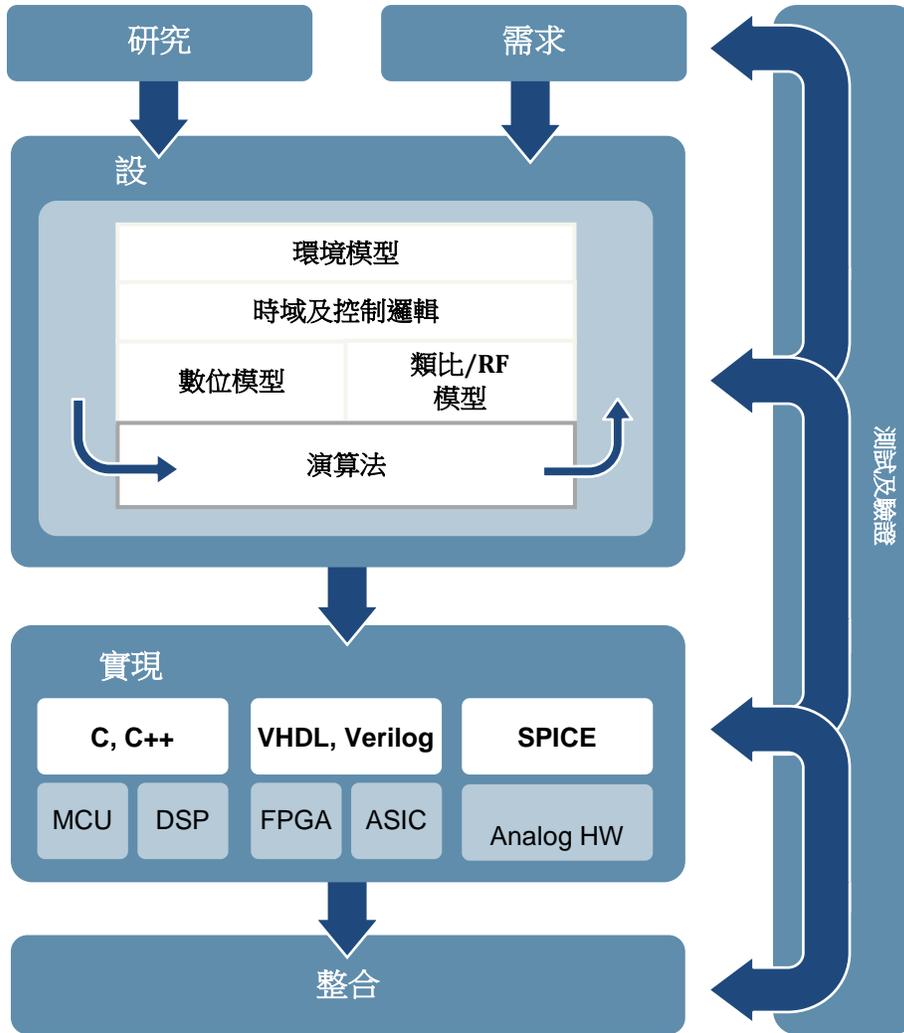


圖 2: 模型化基礎設計工作流程可以進行早期驗證

使用模型化基礎設計驗證系統和元件的行為，具有以下幾個優點：

- 設計和整合問題可以透過模擬在早期發現，此時的錯誤很容易被修正。
- 測試可以和開發與設計同時進行，以確保規格可被執行以符合設計需求規範。
- 進行元件實現時，可重覆使用模型做為測試平台，可除去人工測試和減少詮釋上的錯誤。
- 設計師可以快速地評估設計權衡(tradeoffs)、元件間的交互作用和系統層級的各種可能。

這些早期驗證功能，使工程師們在規格階段能即早發現並糾正設計缺陷，這解決了造成產品延遲上市的一大黑洞；可靠獨立的研究指出，這樣可以減少產品開發時間和減少一半的成本，也能帶來卓越的產品設計。

自動驗證和測試平台(Test Bench)可重覆使用

模型化基礎設計的起步出奇的簡單，首先，演算法設計和實現團隊可以利用他們現在使用的工具讓現有的測試程序自動化。從演算法設計的角度來看，這種自動化改變思維遠多於工作量的變化，因為演算法的設計師本就已經開發出測試平台，以檢查自己的工作，只不過在傳統的設計流程中，這些測試台只是在演算法設計組內使用，但透過模型化基礎設計的流程，這些測試平台可以被重覆使用，而不會在設計實現時因團隊的切換而遺失。

該重覆使用的過程可以透過演算法和系統建模工具內的自動化介面來進行，可以與許多被廣泛使用的硬體模擬器和開發板進行協同模擬(co-simulation)，此協同模擬可代替人工撰寫以及以 script 程式為基礎的技術，不然的話，這些技術則需要驗證 C 程式碼、HDL 和類比電路的實現結果以符合系統層級的各種可能(metrics)。測試平台的重覆使用可顯著地減少驗證工作，而且能讓每個團隊更有效地利用現有的工具和工作流程。

多領域建模

另一個傳統設計流程的問題是：軟體、數位和類比硬體團隊各自使用不同的工具和工作流程，這些不同的工具無法進行跨領域的設計和驗證，因而留下一些暗藏的錯誤，直到系統整合階段才被發現。這個問題可以透過把驗證提高到更高的設計流程來解決，模型化基礎設計(Model-Based Design)支援這種早期驗證方法，透過“虛擬整合”，將模擬演算法、軟體、數位硬體和類比整合在一個環境中，這種多領域建模方法可讓設計者在設計早期評估設計權衡(tradeoffs)、各組件的可能產生的交互作用，以及系統層級的各種可能。

多領域建模和模擬，能把數位元件離散時間(discrete-time)建模和類比元件連續時間建模匯集在一起；它也可以將時序(timing)和控制邏輯、有限狀態機(finite state

machines)、事件導向模擬(event-driven simulation)和定點模擬(fixed-point simulation)整合。設計者可以從較抽象的演算法模型來捕捉行為和驗證的需求，之後隨著工作進展，直到達成時序和位元真(bit-true)準確時再闡述模型。

通過使用模型的虛擬整合，工程師們可以看到元件的設計決策如何影響整個系統的行為，而不必成為不同領域的專家。因此，他們可以更快速地找到滿足或超過設計需求的解決方案，並能解決一般未能發現，直到後期在系統整合階段才出現的問題。

這種方法為快速變動的市場諸如無線通訊市場提供立竿見影的好處，無線系統設計者關心的是不只是基頻演算法，也包含而且射頻鏈(RF chain)、接收器同步，與更高層(higher layer)的通訊協定堆疊(stack)，如網路延遲(network latency)和吞吐量(throughput)等等整合。

舉例來說，一個常見的挑戰是使用較低功率的功率放大器來降低基地台的成本和功耗，這些放大器需要採用數位預失真(digital predistortion, DPD)以及可補償設備非線性失真(nonlinearities)的數位訊號處理(DSP)技術，多領域的建模可以讓 DSP 工程師，藉由使用相關的射頻減損(RF impairments)模型，去驗證這些演算法是否有作用。

快速原型化(Rapid Prototyping)

驗證 IP 演算法的第三個挑戰是，演算法無法如預期在目標環境中作用，在傳統的工作流程中，演算法的問題不易被發現直到設計流程結束，這會造成整個設計出現大量的重覆工作，顯然不可取。

透過模型化基礎設計，演算法開發和系統模擬都使用相同的工具，也能雛形化設計將之實現在處理器和 FPGA 上，無需進行低層的程式編撰，這種早期的驗證技術使設計人員能夠迅速證明新思維的可行性以及分析其在現實條件下的效能。

透過快速原型化，以往需要花費數周的設計測試週期，可以在不到一天的時間內即可完成，此功能對於那些需要快速、徹底驗證一些新想法及未經測試設計的工程師而言，特別具有價值。

量化結果

許多通訊、電子和半導體的領導者目前已採用模型化基礎設計進行早期驗證，因開發成本減少而獲得競爭優勢，如圖 3 所示，獨立研究顯示，採用此種方法的公司可以降低一半的開發成本，這些結果是在現存設計介面到 HDL 模擬器之前估量的，由於導入了新的作法，客戶獲得更好的成果，例如，根據一間國際通訊和資訊技術公司的報告指出，其功能驗證的時間更減少了 85%。

一般規格 到 RTL 成本 (美 元) 之前	一般規格 到 RTL 成本 (美元) 之後	淨直接省下費 用(美元)	省下 百分比
\$ 310 萬	\$ 130 萬	\$ 180 萬	56%

圖 3: 使用模型化基礎設計前後的設計成本

來源：Simulink 進行電子系統設計投資報酬率，©2005 國際商業策略

或許更重要的是創造更好的產品設計和發展更強大的系統模型，以及更容易地開發衍生設計，以及能夠快速適應市場新的規格需求。採取早期驗證技術的公司，也改善了其團隊之間的溝通與協同作業，除了降低他們的測試和驗證成本之外，其快速開發能力也加強了新產品推出的創新能力！