# SystemC促使SoC設計走向更高抽象層

當[SystemC](http://www.eettaiwan.com/search/keyword.php?keywords=SystemC&ACTION_TYPE=SEARCH&operation=PHRASE&search_mod=advanced&section=ALL&encode=1&sub_form=quickform)語言以一種新的[開放原始碼](http://www.eettaiwan.com/search/keyword.php?keywords=%B6%7D%A9%F1%AD%EC%A9l%BDX&ACTION_TYPE=SEARCH&operation=PHRASE&search_mod=advanced&section=ALL&encode=1&sub_form=quickform" \t "_blank)語言在1999年問世時，給設計工程師中帶來了不小的困惑。什麼是SystemC？一種硬體設計語言？如果是的話，怎麼能是以C++為基礎的呢？一種行為級語言？那麼它為什麼又這麼像[RTL](http://www.eettaiwan.com/search/keyword.php?keywords=RTL&ACTION_TYPE=SEARCH&operation=PHRASE&search_mod=advanced&section=ALL&encode=1&sub_form=quickform" \t "_blank)？它會不會取代Verilog和VHDL？其關鍵問題就在於它到底是做什麼的？

五年過去了，以上的所有問題都有答案了。不僅SystemC存活下來了，而且它在[SoC](http://www.eettaiwan.com/search/keyword.php?keywords=SoC&ACTION_TYPE=SEARCH&operation=PHRASE&search_mod=advanced&section=ALL&encode=1&sub_form=quickform" \t "_blank)設計流程中的優點也被認可了。從最初的只被歐洲和日本的少數架構設計者所採用到現在廣泛應用於北美的設計者，許多成功應用SystemC進行設計公司和組織已經將它融入到他們的設計流程中。

OSCI(Open SystemC Initiative)組織網站的點選率和人們對SystemC語言的興趣迅速上升，目前SystemC的專利數量已超過22,000而且還在增加中。很明顯SystemC已經成功定位，再不被認為是HDL的替代者，而是連接建構系統行為級模型的系統架構師和編寫RTL建置程式碼的工程師之間的橋樑。正是因為它從新的2.1版本開始加強了事務級的建模能力，SystemC才能跨越這兩個世界而且幫助進行軟硬體協同模擬。更可喜的是，OSCI和OCPIP(Open Core Protocol International Partnership)正為建置一種能共享的建模結構而繼續合作，如果成功將會使第三方的IP更容易整合在SystemC環境中。OCPIP是一個產業的聯盟，致力於製作一種通用的IP介面。現在已經有不少SystemC的電子系統級(ESL)工具和方法問世，使人們夢寐已久的系統級設計流程得以實現。

## 為什麼是SystemC？

直到幾年以前，設計工程師還能用RTL來描述IC設計。但是現在的SoC設計越來越複雜，並且伴隨著大量的IP重用，這兩個趨勢促使系統設計向更抽象的方向發展。SystemC使架構設計和分析成為可能，但它的能力不止這些，它還能讓SoC設計者在同時執行軟體的條件下以較高的速度驗證其設計並得到有用的分析結果來幫助改進設計。此外，應用SystemC還能在RTL驗證時重用在系統級驗證時所開發的測試平台。

一張含有 桌 的圖片

自動產生的描述

圖1: OSCI和OCP-IP的合作促成了一種層次化結構，下至SystemC核心語言，上至事務級傳輸庫,SystemC驗證庫直到產業標準匯流排協議

SystemC的核心語言建立在標準C++之上，包括事件驅動的模擬核心、結構化的基本元素(模組、埠、介面和訊息通道)、數據類型(整型、定點與向量等等)以及基本的訊息通道(Signal、FIFO與Mutex等等)。

在核心語言之上的是一個簡單的事務級傳輸庫，它建立了事務級的介面模型。同層的還有用於生成測試平台的SystemC驗證庫。在它們之上的則是開放核心協議(Open Core Protocol)的應用程式介面，即為方便IP重用的晶片通訊標準。另外還有一些用SystemC描述的產業標準匯流排協議，例如ARM的AMBA，見圖1。

這種層次化的結構隨著SystemC2.1版本的演變而變得越來越明顯，這一變化也符合OSCI確定的SystemC語言的發展方向，即加強IP的模組化，對事務級設計和驗證提供更好的支援，以及解決以前版本遺留的不易用問題。

## 事務級建模(TLM)的重要性

SystemC 2.1最受人關注的是加強了對事務級建模的支援。無論是嚴格應用於架構驗證而不關心與RTL驗證，還是為RTL驗證服務的行為級參考模型，TLM都是非常重要的。

“TLM是一種抽象的數據串流方式，這種方式更接近於應用設計者或架構設計者的思維方式。”Sonics公司的CTO Drew Wingard說，“它減少了模擬事件的數量，因此提高了模擬的速度。”CoWare公司的專家Pascal Chauvet也說：“TLM的關鍵就是只關注通訊中真正想看到的資訊，這樣就只需要對發起者和接受者交換的資訊進行建模，而不像RTL一樣需要在訊號級建模。TLM可以是非常高層次的抽象，例如簡單的讀和寫，也可是較細節的，例如週期精確性(Cycle Accurate)，但仍然執行得比RTL快很多。”

但是隱含在TLM概念中的彈性，正像Chauvet提到的也在設計者中產生出許多困惑。其中之一就是標準化問題，如今OSCI的事務級建模工作組正進行這一工作。他們對TLM的定義是幾種不同[抽象層](http://www.eettaiwan.com/search/keyword.php?keywords=%A9%E2%B6H%BCh&ACTION_TYPE=SEARCH&operation=PHRASE&search_mod=advanced&section=ALL&encode=1&sub_form=quickform" \t "_blank)次的建模方式，含蓋了所有RTL之上的東西。這些不同的方式應用於不同的設計領域，包括從純演算法模型到RTL模型等(圖2)。

同時也有一種以OCPIP為代表的不同TLM觀點，這種觀點只關心一件事：系統各模組間的介面，同時保持介面在不同建置階段的一致性。雖然OSCI和OCPIP的看法並不互相排斥，但它們的合作還是有利於創造一個可共享的SystemC上的TLM標準。根據雙方的合作協議，OCPIP選擇SystemC作為發佈基於其自身通訊訊息通道的TLM主語言，而OSCI的TLM工作組則正定義一個TLM的應用程式介面，並希望在雙方的努力下將OCPIP的TLM建立在該應用程式介面之上。它們的合作消除了不同標準競爭的可能性，給用戶帶來了福音。

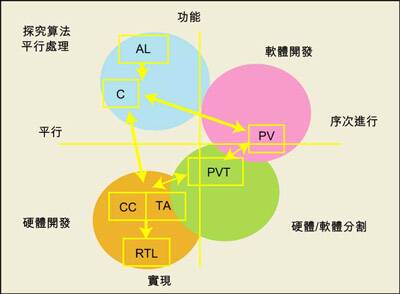


圖2: OSCI的事務級建模工作組定義的幾種TLM抽象級

## 聲明(Assertion)在SystemC中的應用

許多設計者現在已經廣泛使用聲明做為一種驗證方法，用它來建立其設計意圖的規則，然後在驗證過程中用這些規則來檢驗設計。聲明與所使用的設計語言無關的，用Cadence公司功能驗證組的資深架構者Stuart Swan的話來說就是“PSL(Property Specification Language，一種聲明描述語言)也可以應用在SystemC中，而且Cadence已經開始這麼做了。”PSL已經被Accellera認可，而且將要進入IEEE的正式標準化過程，目前支援Verilog和VHDL語言，對SystemC的支援勢在必行。

OSCI已經為PSL在SystemC標準化做了許多工作以保證與已有的標準相容。PSL和其他聲明描述語言相較的最大優勢在於它和底層的設計語言無關，因此避免了不同語言的模組或IP帶來的相容性問題。不過現在還有許多工作要做，例如建立PSL和不同的設計語言共同使用時的標準，目前在這方面還較混亂和任意，幸好IEEE已經意識到這一點，而且已將其提到了議事日程上。

## IP的問題

設計者早期對SystemC的一些擔憂已經消除不少，不過人們還是擔心缺少SystemC的IP。不過業界知名的IP供應商，比如ARM已經開始行動起來，而其合作者Synopsys開發出了DesignWare AMBA SystemC庫，包括AMBA AHB和APB事務級匯流排模型，符合ARM的AMBA2傳輸層SystemC規格，因而確保了完全與ARM的SystemC處理器模型RealView Model庫相容。

而其它的顧慮則與現在的IP有關，如果用戶已經有了RTL模組，他們從哪裡得到相應的SystemC模組呢？ 就像東芝SoC設計總監Bob Uvacek所面臨的一樣，他們是OCP的忠實追隨者，在設計中感受到了OCP的好處，但是他們在沒有一個可完全自動化地完成SystemC和RTL相互轉換的工具出現以前對在設計中採用SystemC還是很猶豫。實際上已經有一些工具可以進行從RTL到C的反向轉換工作，但其實如果遵循自上而下的設計流程完全可以減少對這類型的工具的需求，不過前提是SystemC的IP要夠多且廣泛。

本文由Cadence供稿