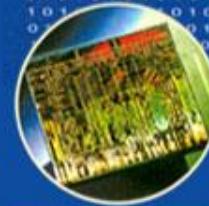


# Verilog

## 硬體描述語言

# Verilog HDL

A Guide  
to Digital  
Design  
and  
Synthesis



IEEE  
1364-2001  
Compliant

# 第1章 導論: 運用Verilog HDL作邏輯設計

1.1 計算機輔助數位設計的發展

1.2 硬體描述語言(HDL)出場

## 1.3 標準設計流程

## 1.4 硬體描述語言的重要性

1.5 Verilog HDL的普及

1.6 硬體描述語言未來的趨勢





## 1.1 計算機輔助數位設計的發展

- 在過去短短25年間，由於數位電路設計的快速發展，從第一代積體電路晶片是**小型積體電路(SSI)**，僅可容納少量邏輯閘。經過到**中型積體電路(MSI)**，可容納數以百計的邏輯閘，到可容納數以千計邏輯閘的**大型積體電路(LSI)**。
- 由於設計電路的工作開始變得非常複雜，電路設計自動化的需求也更迫切，因而促成**計算機輔助設計(Computer-Aided Design)**技術的發展。
- 到了**超大型積體電路(VLSI)**時代，單一晶片已可容納數以萬計的邏輯閘，計算機輔助設計技術成為驗證及設計VLSI電路的唯一方法。

1.1

1.2

1.3

1.4

1.5

1.6



## 1.1 計算機輔助數位設計的發展

- 設計者可以在有圖形介面的終端機上做電晶體層次(Gate-Level)的設計。這時的軟體已經有了階層化的架構，設計者可以使用先前設計的小區塊電路來組成大區塊電路，到合成整個設計為止。
- 邏輯模擬器(Logic Simulator)的出現則負擔了在實際製造晶片前，驗證區塊電路是否能正常工作的任務。

1.1

1.2

1.3

1.4

1.5

1.6



## 1.2 硬體描述語言(HDL)出場

- 電腦程式語言如FORTRAN、Pascal和C都是用來設計電腦程式，且都具有循序執行的性質。
- 在數位電路設計的領域，我們需要一個可以描述數位電路的標準化語言，稱做硬體描述語言(Hardware Description Language(HDL))。
- 硬體描述語言可以用來描述在硬體電路同時執行的情形，較普及的語言有兩種: Verilog HDL 與 VHDL。
- Verilog HDL 發源自Gateway Design Automation; VHDL則是由美國軍方的DARPA所發展出來的。

1.1

1.2

1.3

1.4

1.5

1.6



## 1.2 硬體描述語言(HDL)出場

- 在1980年末期，由於邏輯合成工具的出現，數位電路變成能用硬體描述語言，在**暫存器轉移層次(Register Transfer Level)**來描述，在此層次只需指明資料在暫存器中如何傳送，以及設計上如何處理這些資料。
- 其餘細部的工作如邏輯閘如何連結等，則由**邏輯合成工具**從暫存器轉移層次的描述來轉換而得。
- 邏輯合成工具讓設計者只需將**電路功能及資料傳送情形**，使用硬體描述語言描述出來，邏輯合成工具就可以將所指定的功能用邏輯閘來實現整個電路。
- 因此硬體描述語言也開始應用在系統層次的設計，如系統板(System Boards)、連結匯流排(Interconnect Buses)、FPGAs(Field Programmable Gate Arrays)的設計工作。

1.1

1.2

1.3

1.4

1.5

1.6



## 1.2 硬體描述語言(HDL)出場

- 自1995年開始，美國國家電機工程學會已經針對Verilog HDL訂定了編號1364-1995的標準，在2001年更進一步加入更多功能，並訂定編號1364-2001的標準。

1.1

1.2

1.3

1.4

1.5

1.6



## 1.3 標準設計流程

- 圖1-1是一個標準的超大型積體電路晶片的設計流程圖。
- 無陰影區塊: 代表**設計的層次**
- 有陰影區塊: 設計過程中**處理的程序**
- 在任何設計中，首先是對所設計晶片的要求，指定其功能、介面與整體線路架構，暫不需考慮完成之細部問題。**(行為敘述)**
- 接著用人工方式，將行為的描述轉換成**暫存器轉移層次(RTL)**的敘述，主要是詳細描述資料傳送情形。
- 邏輯合成工具**則將RTL的描述轉換成**邏輯閘層次(Gate-Level)**的線路圖，此線路圖接著可產生佈局圖(Layout)及實做IC。

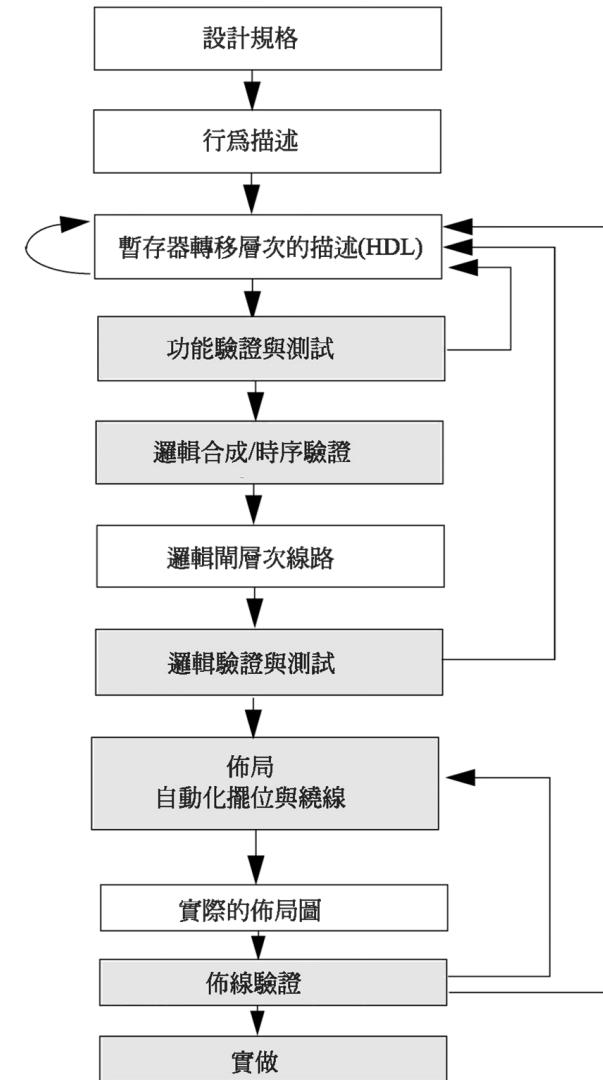


圖1-1 標準設計流程



## 1.4 硬體描述語言的重要性

- 設計者可不需考慮實際的晶片製程技術，邏輯合成工具可將設計相對於不同製程做不同的轉換，並做最佳化。
- 在硬體描述語言的設計中，功能的驗證可以在較高層次就執行，例如現在的暫存器轉移層次。設計者只需要在此層次做修正以符合要求，往下的電晶體層次與實際的佈局圖會發生錯誤的機率相當小。
- 使用硬體描述語言的設計設計電路如同撰寫電腦程式一樣，可以在程式中加入註解，來說明電路的功能，有利發展電路與除錯。

1.1

1.2

1.3

1.4

1.5

1.6



## 1.5 Verilog HDL的普及

- Verilog HDL關於硬體設計的有用特性：
  - Verilog HDL為一般性的硬體描述語言，易學又好用，語法與C語言相似。
  - Verilog HDL允許在同一模組，共同存在不同層次的表示法。設計者可同時使用電晶體、邏輯閘、暫存器轉移及行為模式等不同層次的表示法，來描述所設計的電路。
  - 一般的邏輯合成工具普遍都支援Verilog HDL。
  - 許多製造商都有提供Verilog HDL對應的元件資料庫，使用Verilog HDL設計晶片有較多廠商可選擇。
  - 程式語言介面(PLI)允許使用者可以用C語言撰寫屬於自己的Verilog HDL模擬器。

1.1

1.2

1.3

1.4

1.5

1.6



## 1.6 硬體描述語言未來的趨勢

- 隨著數位電路的複雜性與速度日益劇增，設計者必須在高層次來設計電路才能符合市場的快速需求。
- 藉由**計算機輔助設計工具**，設計者只要考慮電路的功能，細部工作就交給設計工具來最佳化與執行。
- 邏輯合成可做暫存器轉移層次到實際電路的轉換，所以現今趨勢大部份是在**暫存器轉移層次**進行設計的工作。
- 行為層次**的合成工具逐漸出現，等工具成熟後，設計者可以只在電路功能與行為上設計即可，其餘層次工作就交給計算機輔助設計工具執行與最佳化。
- 對於速度要求嚴格的電路，設計者常混合**邏輯閘層次**與**RTL層次**在同一模組中，以達到最佳化的結果。

1.1

1.2

1.3

1.4

1.5

1.6