

需求、規格與 IC 設計

晶片在進晶圓廠之前，要先在電腦上整個設計好、檢查很多遍：大的晶片有上百億個電晶體，畫不完也查不完，每一步都靠設計軟體，有些區塊還是買來的。從「要做什麼」到交出設計圖，每一步都在避免下線之後才發現錯。

半導體入門：跟著晶片的一生

課堂與非營利可以免費使用，請註明 fab24hr ；不能販售，也不能改成自己的作品。

學到什麼

- 1 說得出數位晶片從規格、寫出行為、驗證、合成、佈局繞線到簽核下線的順序，以及每一步用哪一類設計軟體（EDA）。
- 2 說明驗證怎麼做：照規格出題、看覆蓋率、用數學證明，以及為什麼下線之後才發現錯特別貴。
- 3 知道晶片裡哪些區塊常是買來的矽智財（IP）、軟的和硬的差在哪，以及晶圓廠的規則怎麼透過 PDK 接進設計。



下線後還想改，就得重做光罩：又貴、又要多等好幾個月

晶片先在電腦上設計好、檢查很多遍，才下線，交出去做光罩。下線之前常回頭改好幾次；下線之後再改，就得重做光罩。

一步一步

1 定規格：要做哪些事、跑多快、最多用多少電、面積多大。

為什麼：後面每一步都照它：軟體照它的目標找做法，驗證照它出考題。

不這樣會怎樣：沒有規格，就說不出設計對不對，也說不出驗完了沒有。

2 選好晶圓廠和製程之後，挑要買的 IP 區塊，確認有這一代製程的版本。

為什麼：很少有人每一塊都從零畫，而一顆晶片只能用同一套製程做。

不這樣會怎樣：挑到對不上的，就得等人重做一份，或換晶圓廠。

3 用硬體描述語言（HDL），把自己那幾塊電路的行為寫成文字。

為什麼：「每一拍把這兩個數加起來」一行就寫完，畫成電晶體卻是一大片。

不這樣會怎樣：寫錯了，後面的軟體只會照著錯的做。

一步一步

4 驗證：照規格出題模擬、看覆蓋率，買來的 IP 也接上去一起考。

為什麼：在電腦上改一行字就能再試；真的做一次晶片，要好幾個月。

不這樣會怎樣：錯一路帶到下線，就得改設計、重做光罩，而且做出來的每一顆都一樣錯。

5 合成：用元件庫把寫好的行為，翻成邏輯閘接成的電路。

為什麼：同一件事有很多種接法，軟體查表比較速度、面積、耗電，挑合乎目標的。

不這樣會怎樣：人一個一個接，接不完，也比不出哪一種好。

6 佈局繞線：擺好邏輯閘、送好時脈、在好幾層導線裡拉線，做出版圖。

為什麼：擺得好不好很重要：線越長，訊號越慢、越耗電。

不這樣會怎樣：擺不好，線繞不過去或訊號來不及，只能回頭重擺。

一步一步

7 簽核：查規則、查版圖跟電路一不一樣、查電晶體做出來偏快或偏慢（角落）時，訊號還來不來得及。

為什麼：全部過了才下線；光罩是照這份檔案做的，之後就不能隨手改。

不這樣會怎樣：漏掉一個錯，要等好幾個月晶片回來才發現，只能重做光罩。

規格是需求單，軟體照著它找做法

設計從規格開始：這顆晶片要做哪些事、跑多快、最多用多少電、面積多大。這幾樣常互相拉扯：要快，常得多放電路、多用電；晶片越大，一片晶圓排得下的越少，每一顆就越貴；靠一顆小電池撐很久的東西，省電排第一。一顆手機晶片有上百億個電晶體，就算只算一百億個，一秒畫一個、不吃不睡也要三百多年，所以每一步都靠一整套設計軟體，叫 EDA（電子設計自動化）。軟體不會自己設計：工程師把目標告訴它，例如每一拍最多花多久（晶片跟著一個固定的節拍動作，叫時脈），它在限制裡找做法、交出報告；哪裡不合格，人要看懂原因，改了再跑。

從寫出行為，到一層一層的圖形

數位電路先用硬體描述語言（HDL）把行為寫成文字：「每一拍把這兩個數加起來」一行就寫完，畫成電晶體卻是一大片。合成工具再用元件庫（照這一代製程事先做好的小積木，有各種邏輯閘，也有記住一個 0 或 1 的小電路，每一塊多快、多耗電都先算好）把它翻成邏輯閘接成的電路；同一件事有很多種接法，軟體查表挑合乎目標的那一種。佈局繞線再把邏輯閘擺進晶片、把時脈送到每一處、在好幾層導線裡拉線，做出每一層的圖形，叫版圖；線越長，訊號越慢、越耗電。這些軟體都照晶圓廠給的一套檔案算，叫製程設計套件（PDK）：設計規則、電晶體的模型、每一層的資料；模型不準，算得再細，做出來也不一樣。

驗證：照規格出題，找還沒考到的地方

驗證是在晶片做出來之前，確認設計做的事跟規格一樣，一般說法是常佔掉設計工作很大一塊。考題要照規格出，不能照設計的人寫的電路出，不然寫錯的地方也會被當成對的。模擬讓電腦假裝是這顆晶片，送進輸入、跟標準答案比，也讓電腦隨機出很多題；設計每改一次，舊考題全部重跑。考題全對不代表沒錯，可能只是錯的地方還沒考到，所以要看覆蓋率，找出還缺哪些題。題目也試不完：兩個 32 位元的數相加，輸入就有 2 的 64 次方種，一秒試十億種也要五百多年；特別要緊的小部分，就用形式驗證，拿數學證明某件事一定成立，例如兩個訊號絕不會同時打開。

很多區塊是買來的：矽智財（IP）

大的晶片有上百億個電晶體，很少每一塊都從零畫：處理器核心、記憶體介面、把聲音這類訊號變成數字的類比電路，常是買來的矽智財（IP），也就是事先設計好、驗證過，授權給別人放進自己晶片的一塊電路設計；賣的是設計，不是晶片。軟的 IP 是文字寫的電路描述，換一家晶圓廠可以重做一遍；硬的 IP 已經排好圖形，跟某一家的某一代製程綁在一起。類比電路多半是硬的：它對每個電晶體多大、怎麼擺很敏感，交給別人重排就不一定對。類比電路做小，好處不大，有時反而更難做，所以很多類比晶片用好幾代以前的作法做。買來的 IP 單獨驗證過，接上去之後，時脈、電源、重新開機的方式兩邊不一定一樣，所以整顆要一起再驗證。

常見誤解

常聽到：設計晶片就是寫程式。

其實：硬體描述語言寫起來像寫程式，差別是：程式照順序一行一行做，它寫的是電路，很多部分同時在動，而且大多跟著一個固定的節拍（時脈）動作。寫完也還沒結束：要驗證、合成、排出位置和連線、通過簽核，才交得出去。

常聽到：驗證、晶圓廠的量測、晶片測試都是在檢查，做一種就夠了。

其實：三件事查的東西不一樣。驗證在下線之前、大多在電腦上，查設計圖對不對，錯了每一顆都一樣錯；量測與檢測在晶圓廠做到一半，查重要的步驟做得對不對，例如膜多厚、線多寬；晶片測試通常在晶圓做完、包好之後各一次，每一顆都測，查做出來的這一顆有沒有壞。像蓋房子：開工前檢查設計圖，蓋的時候量牆直不直，交屋前每一間開水開電試一次。

玩

**從第 1 章「12 片怎麼分」開始：把設計時
用模型算過的角落真的做出來，看設計撐不撐
得住**

用手機掃，從這裡開始玩

fab24hr.com/?story=1&utm_source=slides



想一想

- 1 一家做助聽器的小公司第一次自己設計晶片：把麥克風的聲音變成數字的類比電路、給整顆晶片打節拍的時脈電路，可以自己畫，也可以買現成的 IP。你會怎麼決定哪些自己做、哪些用買的？買之前會先問賣的人哪些事？
- 2 驗證要花很多人力和時間，可是少做一點，又怕下線之後才發現錯。如果時程很趕，你會怎麼決定「檢查到這裡就夠了」？

必讀

晶片是怎麼設計出來的？ IC 設計在做什麼

fab24hr.com/blog/chip-design/

矽智財（IP）是什麼？晶片裡為什麼有買來的電路

fab24hr.com/blog/silicon-ip/

EDA 是什麼？設計晶片的軟體，每一步在做什麼

fab24hr.com/blog/what-is-eda/

晶片驗證是什麼？下線前怎麼確定設計是對的

fab24hr.com/blog/chip-verification/

延伸

SoC 是什麼？手機裡的「系統單晶片」白話解釋

類比晶片和數位晶片差在哪？

IC 設計（VLSI）和晶圓廠的工作差在哪？

半導體入門：跟著晶片的一生

<https://fab24hr.com/learn/design/>

內容整理自公開的教科書、產業標準、大學課程講義與論文；遊戲裡的公司、人物、機台編號和數字是虛構的。

課堂與非營利可以免費使用，請註明 fab24hr ；不能販售，也不能改成自己的作品。

通用示意：依公開資料、由 AI 協助製作，可能有錯；不代表任何一家公司的機台