



辦公室的故事 (Incremental Compilation)

我們總是在想，有沒有那麼一種可能，讓我可以少花力氣多賺錢，我們這裏就開始討論這樣一個問題。

好比你現在有一個很大的公司，你有一個辦公室，裏面空空的，你可以有很多方法把你的員工放在裏面。當然你可以一股腦兒的把所有人全部很隨意的放在裏面。但是導致的結果會是什麼樣的呢？你會發現兩個工作關係很精密的人，每天需要從這頭走到那頭的交流。這樣的工作效率不可能好的。所以我們會比較希望把他們整合在一起。至少可以讓一個部門坐在一起。然後我們再在這個相對比較小的區域裏面做調整來提高工作效率，這樣會容易很多了。每個部門當然就是公司的一部分(design partition)，我們需要把這個部門的人都放在一個限定的空間裏面，而給每個部門提供的空間，我們定義為 Logic Lock。空間是一個物理的概念，而部門是一個人為理念造成的一個概念。我們並不是一定要把他們倆結合起來。但是，如果結合起來會有這樣一個好處。

好比，我有五個部門 A,B,C,D,E。而不給他們限制空間，他們可能隨意的去做。也許相對來說部門內的人士坐在一起了，但是這中間留下的空間變得很沒規律。兩種情況下會有問題，一個是，需要加一個新的部門，那麼這個部門就需要拆開放，另外就是某些部門擴張了，新人就沒地方放了。所以，宏觀調控是非常重要的。所以我們需要把設計理念的分割和物理空間的分割緊密的結合在一起。

增量編譯的思想和這個故事是完全一致的。我們需要把設計分成一些相對比較大的部分，然後給他們安排相應的位置去放置。這樣有非常多的好處：

1. 關係相對精密的電路的位置比較接近，減少了連接線上的延遲。
2. 對一些已經表現很好的模組可以保留他們的編譯結果。這樣可以獲得最大的好處。每一次編譯都獲得一個比較好的結果加以保留，這樣整個設計就一點點按照增量的方式獲得了最好的那個結果。
3. 由於一些模組已經保留了，所以工具不會在那些模組上面耗費時間，所以編譯時間大大縮短。

我們來看看有這種思想可以實現的兩種編譯過程：

自上而下：

我是公司的總經理，我們去招了一堆人，然後把它們放在各個部門裏面。根據每個部門人數的多少，給他們劃分相應的資源。

自下而上：

我是公司的總經理，我手下有一些部門的負責人，我讓這些人去招人。我給他們一些人數的預算，然後給他們提供空間。那些部門經理自己去招人來填充自己的一畝三分地。最後把這個辦公室填滿。

和其他所有工具一樣，我們需要有一些行為準則來獲得最好的效果：

1. 儘量把工作關係精密的人放在同一個部門裏面。部門和部門之間是不能進行優化的。好比兩個人需要每小時都交流工作，但是你把他們放在兩個部門，即使是相互隔壁的部門，也不能保證他們正好牆對牆的做。工具只對部門內部進行調整優化，出了部門，他們就死人不管了。
2. 關門原則。在部門和部門之間都按上門。就是說對輸出和輸入的資料都用 register 打一下。這其實是和第一條是相關的。因為部門之間不進行任何優化，為怕避免麻煩，建議大家都按上門來提高效率
3. 部門和部門之間儘量少串門。儘可能的減少互動。這個其實是在設計的過程中進行調整挑選的。
4. 保證一個部門的人數，不是太少（大於 2,000LE）。這很好理解，就那麼幾個人，還優化什麼呀。
5. 不要放不用的介面。沒有結果的連接在編譯中是會被優化掉的。但是在增量編譯中，卻會被保留。所以如果已經知道一個介面是沒有用的，不如把它幹掉算了。
6. 大家知道，FPGA 內部是沒有雙向信號的，所以一定避免雙向信號。除非你這個型號是要直接連到晶片管腳上的。
7. 避免同一個信號重複輸出。
8. 不要把直接從輸入到輸出的信號放到部門內部。這是一種資源浪費
9. 公司有一些資源，好比印表機啊，傳真機啊之類的，在分配辦公區域的時候，儘量考慮資源的合理使用。由於資源位置是固定的，所以當你把區域分配給部門的時候，其實也把資源分配給他們使用了。而如果他們用不上，也就浪費了。
10. 多給部門與部門之間的交流留一些時間預算。

這麼看起來，如果你學會了增量編譯，好像順便把 HR 也個學了。