

## Lab 8: DRC and LVS

---

- 1 These Labs are modified from the CIC standard flow
- 2 在今天的 Lab 中，我們會練習到
  - 2.1 Stream In GDS with Virtuoso
  - 2.2 Design Rule Check (DRC)
  - 2.3 Layout Versus Schematic (LVS)
- 3 先複製並解壓縮 Calibre Lab 的檔案。  
Lab files are at `~cvsd/CUR/Calibre` .
- 4 同學請將自己於前面實驗完成的 `CHIP.v` 和 `CHIP.gds` 替換到 `design_data` 目錄裡，繼續完成下面的驗證。但如果前面實驗還沒有完成，或是自己的檔案有驗證錯誤，可以使用我們提供的 `CHIP.v` 和 `CHIP.gds` 檔案。

## Lab 8-1: Virtuoso

---

1 目的: 使用 Virtuoso Layout Editor 檢查自己的設計, 熟悉 Virtuoso Layout Editor 的使用者介面

2 進入 Lab8-1

*% cd Lab8-1*

3 檢查相關的檔案是否齊全

3.1 GDS File : CHIP.gds (前次 Lab 自己產生, 或本次 Lab 所提供的)

3.2 Virtuoso Technology File : Virtuoso4.4\_0.13um\_Ver2.1a.1.tf 和 display.drf

3.3 Initial File : .cdsinit

3.4 上述檔案中的 GDS File 是由 SOCEncounter 產生的, Virtuoso Technology File 會隨著所使用的製程不同而改變, Initial File 則是 Virtuoso 的初始環境設定檔

3.5 檢查.cdsinit 的內容

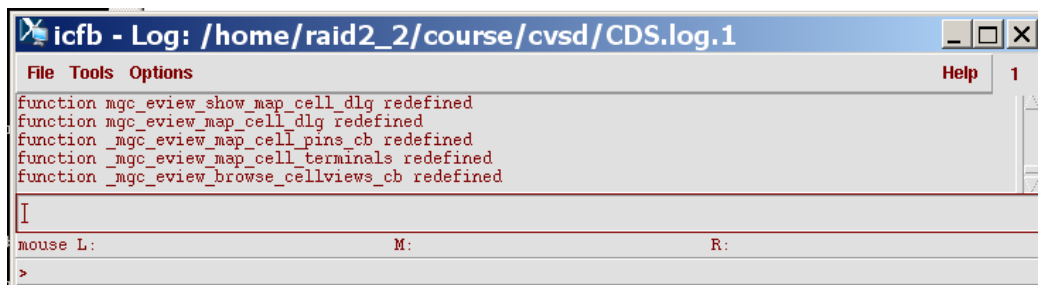
*% vi .cdsinit*

裡面 skill file 的路徑會隨著使用環境不同而改變, 如果要在 Virtuoso 執行 Calibre 的話要注意這點, CAD 工作站底下正確路徑應為:

`/usr/mentor/calibre/cur/shared/pkg/icv/tools/queryskl`

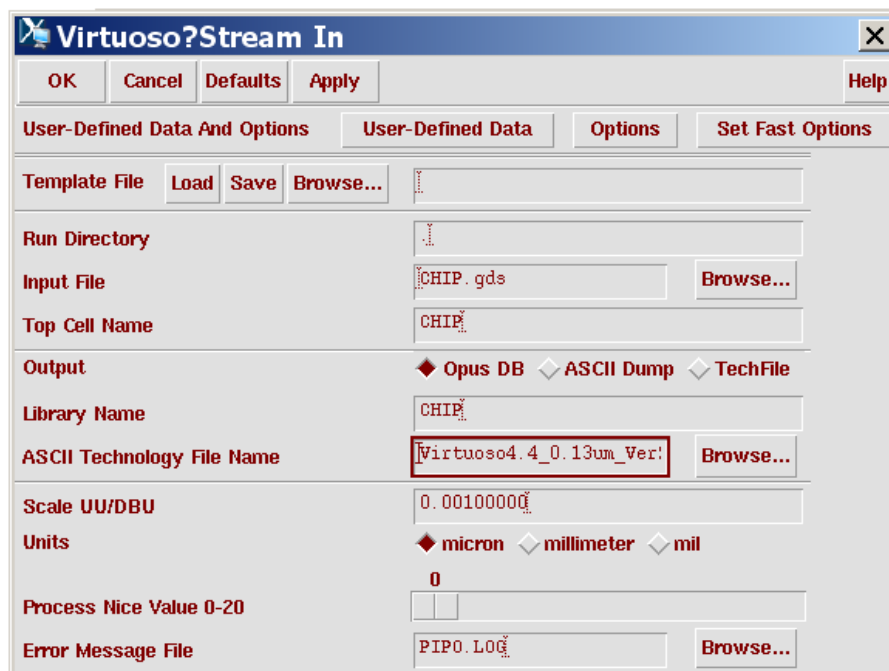
4 執行 icfb

*% icfb &*

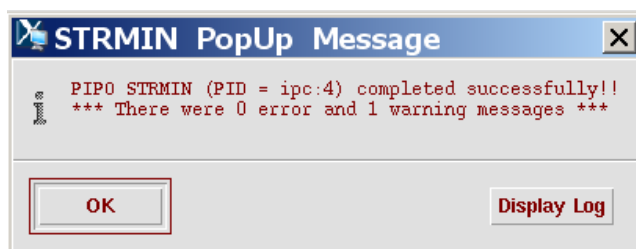


5 Stream In Design

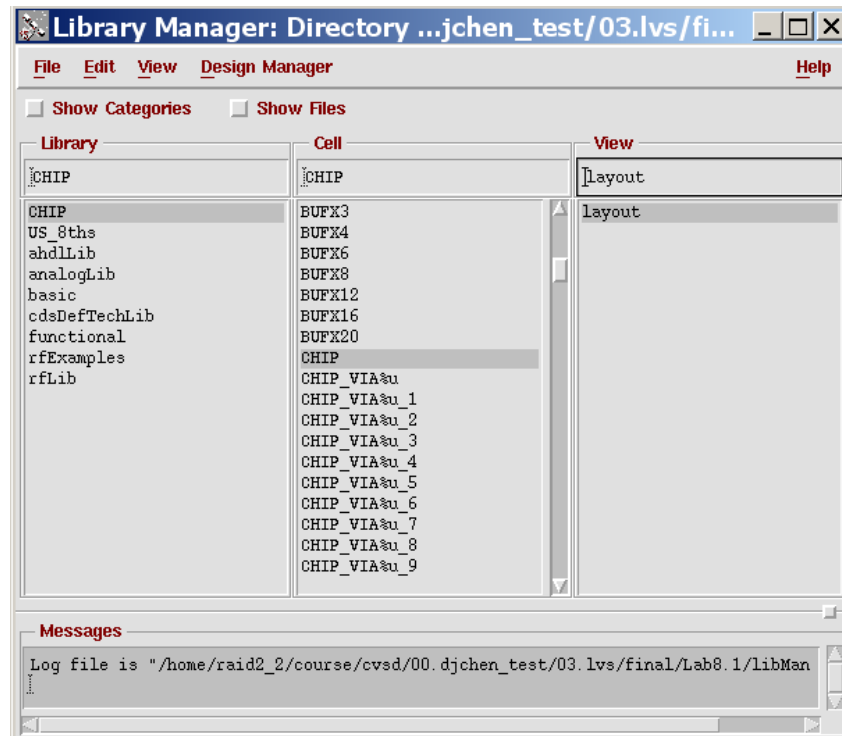
5.1 在 icfb 視窗裡, 執行 File->Import->Stream... 會跳出一個新視窗, 在 Input File 欄位輸入 CHIP.gds, Top Cell Name 欄位輸入 CHIP, Library Name 欄位輸入 CHIP, ASCII Technology File Name 欄位輸入 Virtuoso4.4\_0.13um\_Ver2.1a.1.tf, 最後選 OK。



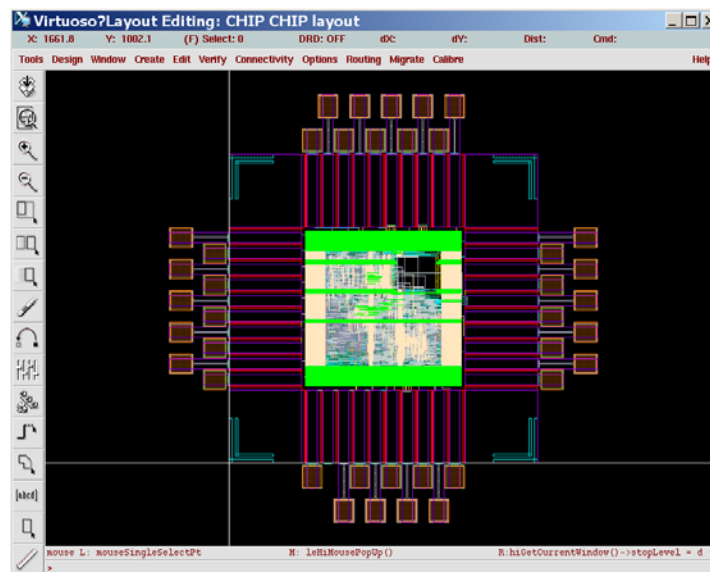
成功的話會出現如下訊息表示沒有 Error 發生。



- 5.2 在 icfb 視窗裡，執行 Tools->Library Manager...會跳出 Library Manager 視窗，先執行 View->Refresh 更新欄位內容，然後在 Library 欄位選擇 CHIP，Cell 欄位選擇 CHIP，View 欄位選擇 layout，並在其上面按右鍵，選 Open 來打開 CHIP/CHIP/layout 這個 cell view:

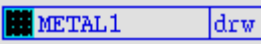


打開後按下 shift+f 展開所有 cell，如下圖：



## 6 熟悉 Virtuoso Layout Editor 的使用者介面

6.1 這時候操作的視窗主要有兩個，一個是顯示 layout 的 Virtuoso Layout Editor，另外一個是長條型的 LSW。

6.2 LSW 主要用來控制各 layer 的顯示與否，例如  是指是否顯示 Metal1 的 layer，在上面按滑鼠中鍵圖示會變成



，回去 Layout Editor 那邊按 ctrl+r 把畫面 redraw，

Metal1 layer 便沒有再顯示在畫面中。

- 6.3 Virtuoso Layout Editor 是用來對 layout 進行一些操作，所以用的功能如視窗左邊的圖案所示，例如改變 Metal 的位置 ，延伸或縮短 Metal 長度  等，主要是用來解決 DRC/LVS Error 用。下面的快捷鍵可以幫助使用者觀察 layout。

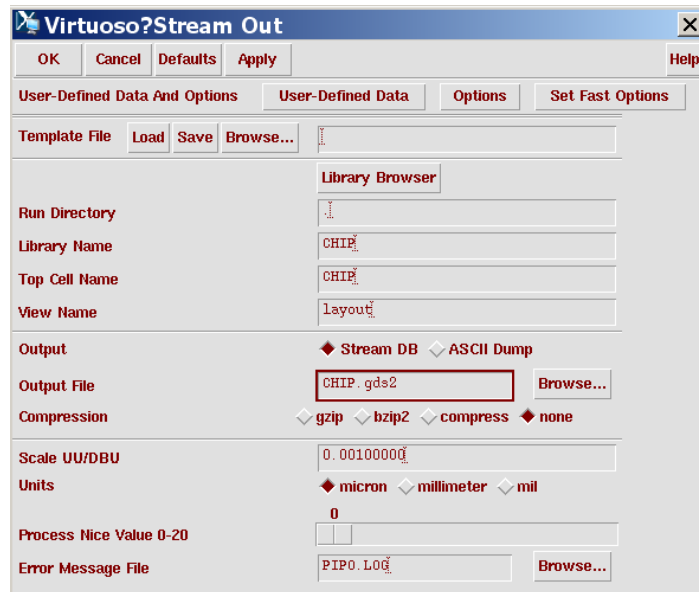
Zoom In	ctrl + z
Zoom Out	shift + z
Fit	f
Display All Hierarchy	shift + f
Display Top Hierarchy	ctrl + f
Edit Instance Property	q

## 7 檢查自己的設計

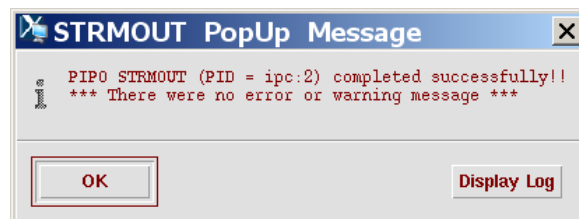
- 7.1 在 Display All Hierarchy 時，在四個 corner pad 的外側會看到兩條 L 的形狀，可以用來確定 corner 擺放的方向有沒有問題。
- 7.2 在之前用 SOCEncounter 進行 P&R 時不用為 memory 加入 block ring，是因為它本身就已經有包含，在這邊只有 Zoom In 到 memory 的位置便可以看到。
- 7.3 到之前加 IOVDD 和 IOVSS 的位置把 label 找出來，按 q 可以檢查 label 所在的 metal layer 是否跟之前設定的一樣。

## 8 Stream Out

- 8.1 在 icfb 視窗執行 File->Export->Stream...，在 Stream Out Form 中的 Library Name 欄位填入 CHIP，Top Cell Name 欄位填入 CHIP，在 View Name 欄位填入 layout，Output File 欄位填入 CHIP.gds2，最後按 OK 開始進行 stream out 的動作：



成功的話會出現如下訊息表示沒有 Error 發生:



## 9 關閉 icfb

9.1 在 icfb 視窗選擇 File->Exit...，按 YES 即可離開。

## 10 Lab8-1 結束

**Check: 完成 Virtuoso Layout Editor 的開啟並 output stream CHIP.gds2**

## Lab 8-2: Design Rule Check

- 1 Lab 目的: 學習如何進行 Design Rule Check (DRC)。
- 2 進入 Lab8-2
  - % cd Lab8-2*
- 3 複製相關的檔案到本資料夾
  - 10.1 GDS 檔案: CHIP.gds (前次 Lab 自己產生, 或本次 Lab 所提供的)
- 4 編輯 Calibre DRC runset 檔案
  - 4.1 對 CL13S\_8M.21d 修改以下幾個項目 (約在第 345 行), 如紅字部份所示:
    - LAYOUT PATH "CHIP.gds"
    - LAYOUT PRIMARY "CHIP"
  - 4.2 存檔並離開。
- 5 執行 Calibre DRC
  - 5.1 *% source /usr/mentor/CIC/calibre.cshrc*
  - 5.2 *% calibre -drc CL13S\_8M.21d*
- 6 觀看 DRC 結果
  - 6.1 打開檔案 DRC.rep, 檢查 RULECHECK RESULT STATISTIC 中 Result Count 不等於 0 的部份即為 DRC Error
  - 6.2 *% grep ^RULECHECK DRC.rep | grep -v 0\$*
  - 6.3 如果用預設的 GDS 檔應該會看到幾個 DRC Error, 像 OD.DN.x, PO.DN.x, Mx.DN.1L, DOD.R.1, DPO.R.1 等, 其中 Mx.Dn.1L 指體的 metal density 不夠, 這是因為設計最外圈的 bounding pad 佔的比例太大, 而 dummy metal 沒有加到 core 外面的區域, 所以整體 density 不夠。其餘錯誤指的是 poly 和 oxide 的 density 不足, 這是因為我們所使用的 standard cell layout 並不是完整的, 裡面沒有 poly 和 oxide, 所以 density 不夠是正常的。這幾個 DRC Error 都不用理會。

```

cad32:/home/raid2_2/course/cvsvd/00.djchen_test/03.lvs/final/Lab8.1% grep ^RULECHECK DRC.rep | grep -v
0$
RULECHECK OD.DN.1 ..... TOTAL Result Count = 1
RULECHECK OD.DN.3 ..... TOTAL Result Count = 1
RULECHECK PO.DN.1 ..... TOTAL Result Count = 1
RULECHECK M1.DN.1L ..... TOTAL Result Count = 1
RULECHECK M2.DN.1L ..... TOTAL Result Count = 1
RULECHECK M3.DN.1L ..... TOTAL Result Count = 1
RULECHECK M4.DN.1L ..... TOTAL Result Count = 1
RULECHECK M5.DN.1L ..... TOTAL Result Count = 1
RULECHECK M6.DN.1L ..... TOTAL Result Count = 1
RULECHECK M7.DN.1L ..... TOTAL Result Count = 1
RULECHECK M8.DN.1L ..... TOTAL Result Count = 1
RULECHECK DOD.R.1 ..... TOTAL Result Count = 1
RULECHECK DPO.R.1 ..... TOTAL Result Count = 1
cad32:/home/raid2_2/course/cvsvd/00.djchen_test/03.lvs/final/Lab8.1%
  
```

- 7 Lab8-2 結束
  - Check: 完成 DRC, 截取 count 不等於 0 的部份**

## Lab 8-2(b): Layout vs. Schematic

---

- 1 Lab 目的: 學習如何進行 Layout Versus Schematic (LVS)。
- 2 進入 Lab8-3  
`% cd Lab8-3`
- 3 複製相關的檔案
  - 10.2 GDS 檔案: CHIP.gds (前次 Lab 自己產生, 或本次 Lab 所提供的)
  - 10.3 Netlist 檔案: CHIP.v (前次 Lab 自己產生, 或本次 Lab 所提供的)
- 4 編輯 Calibre LVS runset 檔案
  - 4.1 對 CL013G\_1P8M.lvs 修改以下幾個項目 (約在第 153 行), 如紅字部份所示:  
LAYOUT PRIMARY "CHIP"  
LAYOUT PATH "CHIP.gds"  
LAYOUT CASE YES  
SOURCE PRIMARY "CHIP"  
SOURCE PATH "source.spi"  
SOURCE CASE YES  
其中加入的兩個 CASE YES 表示 layout netlist 和 source netlist 都是 case sensitivity。
  - 4.2 在最後一行加入 LVS BOX RF2SH64x16, 意思是要把這顆 memory 在 LVS 時當成是黑盒子不作內部的比較。未來有使用到其他的 Memory 或是類比模組, 在做連接性的驗證時也可以這樣使用。
  - 4.3 最後還要把 LVS BOX 中的 power pad, corner pad, bonding pad 都註解掉, 如:  
//LVS BOX FILL1  
//LVS BOX FILL16  
//LVS BOX FILL2  
//LVS BOX FILL32  
//LVS BOX FILL4  
//LVS BOX FILL64  
//LVS BOX FILL8  
//LVS BOX PADIZ40  
//LVS BOX PADOZ40  
//LVS BOX PCORNERDG  
//LVS BOX PCORNERDG\_L  
//LVS BOX PFEED01  
//LVS BOX PFEED10



```
//LVS BOX PFEED1
//LVS BOX PFEED20
//LVS BOX PFEED2
//LVS BOX PFEED22
//LVS BOX PFEED35
//LVS BOX PFEED50
//LVS BOX PFEED5
//LVS BOX PVDD1DGZ
//LVS BOX PVDD2DGZ
//LVS BOX PVSS1DGZ
//LVS BOX PVSS2DGZ
//LVS BOX PVSS3DGZ
```

原因是因為在 verilog netlist 中這些 cell 都不存在，可是 layout 有，所以希望不要把這些 cell 當成是黑盒子。

#### 4.4 存檔並離開。

### 5 檢查 LVS 的目錄，我們發現目前還沒有 source.spi，因此必須先把 CHIP.v 轉換成 SPICE 格式輸入：

```
5.1 % source /usr/mentor/CIC/calibre.cshrc
5.2 % v2lvs -v RF2SH64x16.v -o RF2SH64x16.spi
5.3 % v2lvs -v tsmc13gfsg_fram_lvs.v -o tsmc13gfsg_fram_lvs.spi
5.4 % v2lvs -v tpz013g3_lvs.v -o tpz013g3_lvs.spi
5.5 % v2lvs -v CHIP.v -l RF2SH64x16.v -l tsmc13gfsg_fram_lvs.v -l
    tpz013g3_lvs.v -o source.spi -s RF2SH64x16.spi -s tsmc13gfsg_fram_lvs.spi
    -s tpz013g3_lvs.spi -s1 VDD -s0 VSS
```

#### 5.6 成功的話會產生source.spi

### 6 執行 Calibre LVS

```
6.1 % calibre -lvs -spice layout.spi -hier -auto CL013G_1P8M.lvs
```

### 7 觀看 LVS 結果

7.1 打開檔案 lvs.rep，檢查 OVERALL COMPARISON RESULT 的部份是不是出現笑臉與大勾，如果是的話便表示設計有通過 LVS 了。

```
OVERALL COMPARISON RESULTS

#####
#          #          #          #          #          #          #          #
#          #          #          #          #          #          #          #
#          #          #          #          #          #          #          #
#          #          #          #          #          #          #          #
#####

Warning: Components with non-identical power or ground pins.

*****
CELL SUMMARY
*****

Result      Layout      Source
-----
CORRECT     CHIP        CHIP
```

8 Lab8-2(b)結束

Check: 完成 LVS 並出現笑臉