

## Cadence On-Line Document

---

- 1 目的: 使用 Cadence 線上查詢系統來幫助查詢 SOC Encounter 相關資料。
- 2 Cadence On-Line Document  
Cadence 公司的線上文件查詢系統，SOC Encounter 為 Cadence 公司之軟體，可用此線上查詢系統來 LEF, DEF 的語法以及 SOC Encounter 的指令以及語法。
  - 2.1 開啟線上查詢  
% /home/raid1\_1/linux/cadence/SOC/cur/tools/bin/cdnshelp &
  - 2.2 觀看 LEF/DEF Syntax 語法  
Products → Languages → LEF/DEF 5.7 Language Reference → LEF Syntax
  - 2.3 觀看 SOC Encounter 的 command line 指令  
Products → SoC Encounter → ...  
使用 search 的功能可協助我們更快查到需要的指令(search topic, search manual, ...)，例如 search “addstripe” (用來做 Powerplan 之一的指令)。

Source: These Labs are in the CIC standard flow (modified by Meng-Kai Hsu, 2011.11).

## SOCE Lab (1/2): Circuit Placement & Power Planning

---

- 1 Lab 目的: Design import, Floorplan, Powerplan (Power Ring, Power Stripe)
- 2 Lab materials are available at “~cvsd/CUR/SOCE/SOCE\_Lab.tar.gz”.  
Please install “MobaXterm” for graphical interface before lab (available from internet or “~cvsd/CUR/SOCE/MobaXterm\_v2.2.zip”).
- 3 可以先使用 vi , joe , 或是 cat 等指令查看 design\_data 的 CHIP.v 、CHIP.ioc 以及 CHIP.sdc 檔案,CHIP.v 檔主要是在原本合成完的 Design 上面架上 CHIP 之 module,此 module 包含 input/output pad 以及原本 Design 之 Top module (此範例使用的是 DCT 電路);而 CHIP.ioc 主要則是規畫各個 input/output pad 是擺在 CHIP 四周的哪些地方,此外還包含 Core power pad, I/O power pad 以及 Corner pad ; CHIP.sdc 則是包含 timing constraint ,之後做 timing analysis 與 timing driven PR (Place and Route)會用到。
  - 3.1 Core power pad 主要是用來做晶片內部的供電所需。
  - 3.2 I/O power pad 主要是用來做 input/output pad 本身的供電所需。
  - 3.3 Corner pad 是用來做 pad 與 pad power 之連接用。
- 4 開啟 SOC Encounter
  - 4.1 % source /usr/cadence/CIC/soc.csh
  - 4.2 % encounter  
注意不可以加上 & 指令(背景執行),因為 SOC Encounter 開啟之後會使用到兩種不同的介面,一者為目前開啟 SOC Encounter 的 terminal (command line 輸入),另一個則是開啟軟體之後會出現的 GUI 介面。
- 5 Design Import
  - 5.1 Design → Import Design...
  - 5.2 Verilog Netlist → Files 選擇旁邊的 “...” 按鈕,出現 Netlist Files 之後按下之後選取 design\_data/CHIP.v,接下按下 Add,即可將 CHIP.v 檔輸入至 Verilog Netlist → Files 中。
  - 5.3 Verilog Netlist → Top Cell 填入 CHIP。
  - 5.4 Timing Libraries 下面的幾種 Libraries 皆可按照 4.2 的方法依序輸入  
Max Timing Libraries:  
library/lib/RF2SH64x16\_slow\_syn.lib  
library/lib/slow.lib  
library/lib/tpz013g3wc.lib  
Min Timing Libraries:  
library/lib/RF2SH64x16\_fast@0C\_syn.lib  
library/lib/fast.lib  
library/lib/tpz013g3lt.lib

Max Timing Libraries 主要是用來計算 setup time，而 Min Timing Libraries 則是用來計算 hold time，此外也可以讓 SOC Encounter 在做 Timing optimization 時可以用哪些 inverter 或是 buffer。

### 5.5 LEF Files

library/lef/tsmc13fsg\_8lm\_cic.lef

library/lef/tpz013g3\_8lm\_cic.lef

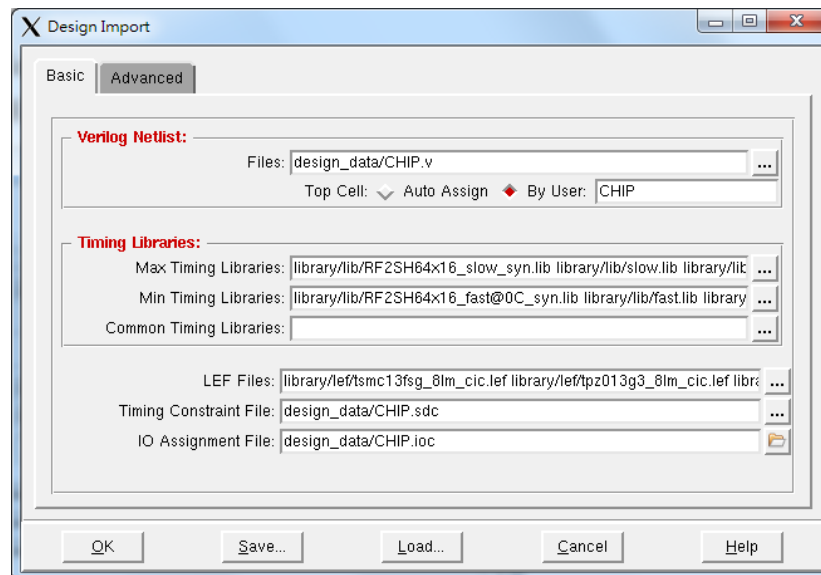
library/lef/RF2SH64x16.vclef

library/lef/antenna\_8.lef

**注意：**由於 tsmc13fsg\_8lm\_cic.lef 包含了所有製程資訊，所以一定要擺在第一個輸入。

### 5.6 Timing Constraint File 填入 design\_data/CHIP.sdc。

### 5.7 IO Assignment File 填入 design\_data/CHIP.ioc。



### 5.8 切換到 Advanced。

### 5.9 Power

5.9.1 Power Nets 填入 VDD；Ground Nets 填入 VSS (TSMC 製成)。

### 5.10 RC Extraction

5.10.1 RC Extraction → Typical/Best/Worst Capacitance Table File 三者皆填入 library/tsmc013.capTbl。

5.10.2 QX → Tech File 填入 library/tsmc13\_8lm.cl/icecaps\_8lm.tch。

5.10.3 QX → Library Directory 填入 library/tsmc13\_8lm.cl。

### 5.11 SI Analysis → CeltIC Libraries

5.11.1 Max cdB File 填入 library/celtic/slow.cdB。

5.11.2 Min cdB File 填入 library/celtic/fast.cdB。

5.11.3 Common cdB File 填入 library/celtic/typical.cdB。

5.12 由於每次在 Design Import 這邊都需要輸入很多設定，我們可以在填完設定之後按“Save...”的按鈕將我們設定好的檔案存出去，方便以後可以

直接使用“Load...”的按鈕快速將我們的設定載入。

5.13 此外此 Lab 已經有存好一個設定檔 CHIP.conf，可以直接載入。

5.14 按下 OK

## 6 Global Net Connect

此步驟主要是把所有的 Standard cell 的 power/ground pin 連接到 VDD/VSS。

### 6.1 Floorplan → Connect Global Nets

6.1.1 Pin Name(s)填入 VDD，To Global Net 填入 VDD，按 Add to List。

6.1.2 Pin Name(s)填入 VSS，To Global Net 填入 VSS，按 Add to List。

6.1.3 將 1'b1/1'b0 連接至 VDD/VSS。

6.1.3.1 選擇 Tie High，To Global Net 填入 VDD，按下 Add to List。

6.1.3.2 選擇 Tie Low，To Global Net 填入 VSS，按下 Add to List。

6.1.4 按下 Apply，再按 Check，再按 Close。

注意：除了將連接 1' b1/1' b0 至 VDD/VSS，我們亦可於之後再插入 Tie high/Tie low cell (跳過步驟 5.1.3)，其缺點在於會增加 chip utilization。另外須注意的是，如果 1'b1/1'b0 還沒有連接，Check 時可能會出現一些 Warning，在現階段是可以忽略的。(會在之後講解如何插入這些 cell。)

## 7 Specify Scan Chain

由於 Design 中已經有插入 Scan chain，所以必須跟 SOC Encounter 講 Scan chain 的位置 (scan in, scan out)，此步驟須由 terminal 輸入 command：

```
7.1 encounter> specifyScanChain scan1 -start ipad_SCAN_IN/C -stop opad_SCAN_OUT/I
```

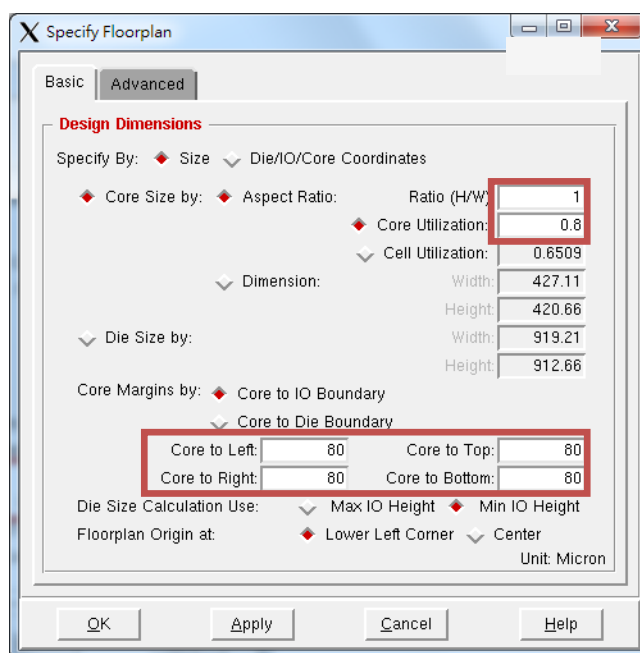
```
7.2 encounter> scantrace
```

注意：在這個範例中，scan out 為 primary outputs，理論上必須 specify 所有 scan in 到 primary output 的 scan chain，在此我們儘考慮唯一的 scan path。

**Q1: Scan trace summary 顯示這個 scan train 共有多少個 bits? \_\_\_\_\_**

## 8 Floorplan

8.1 Floorplan → Specify Floorplan ... (根據不同 Design，可給予適當設定。)




8.2 可以看到 Layout 上面旁邊粉紅色區塊是我們的 Design (DCT)，而另一邊的綠色區塊(SRAM\_i0)則是 Design 中使用到的 Memory block。

8.3 在此可以先讓 SOC Encounter 幫我們初步的擺入所有的元件  
Place → Standard Cells ...

8.3.1 選擇 Run Placement In Floorplan Mode (Optimization Options 的兩個選項 Include Pre-Place Optimization 及 Include In-Place Optimization 都取消不選)，按 OK。

8.4 執行完之後可以選擇不同的 design view (Amoeba view, ...)，會看到 tool 已經幫我們把所有的 cell 以及 block 擺進去 core 裡面，在此我們可以任意更改 Memory 的擺放方式，例如以下步驟：

8.4.1 用滑鼠點一下 Memory，選擇 Floorplan → Edit Floorplan → Flip/Rotate Instances...，選擇 R90，按 OK，此時會發現我們的 Memory 已經被轉了 90 度角。

8.4.2 切換到 Floorplan view，選擇 ，此時再去點選 Memory 會發現我們們可以任意移動 Memory 的位置，通常我們會將 Memory 擺到 Core 的四周 (ex: 往右上角擺)。

8.5 替 Memory 加上 Halo

要做這個步驟的原因是因為 Halo 區塊之下不會擺放任何的 Standard cell，這使得之後 Memory 在做繞線 (Routing) 時可以有比較大的空間，不至於讓 Standard cell 擠在 Memory 周圍。

8.5.1 Memory → Floorplan → Edit Floorplan → Edit Halo...

8.5.2 在 Edit Halo form 中，選擇 Selected Blocks/Pads (確定現在的 Memory 是在選到的狀態)，現在要加的是 Placement Halo，在

Top/Bottom/Left/Right 四個欄位都填入指定的寬度 (30um)，按 **OK**，會看到 Memory 周圍多了一圈紅的區域。

## 8.6 Timing analysis

8.6.1 Timing → Analysis Timing ...

8.6.2 Design Stage 選擇 Pre-CTS，按下 OK。

此時 Tool 會開始分析 trial route 與 RC Extraction，計算出各點的 delay 後再使用 STA (Static timing analysis) 分析 Data path。

8.6.3 Tool 跑完之後可以在 Terminal 看到分析之後的結果，主要要看的就是 WNS (Worst Negative Slack)，如果這個值是負的，則表示目前的 Placement 結果無法達到 CHIP.sdc 裡面的 Timing constraint。

**Q2: 此時 WNS 是多少? \_\_\_\_\_ ; TNS 是多少? \_\_\_\_\_。**

8.6.4 如果 WNS 是負的，SOC Encounter 有一補救的方法，就是執行 Timing Optimization (後面會講解步驟)。

8.7 在做完初步的 Placement (Run Placement In Floorplan Mode) 之後，我們緊接著跑 Full mode 的 Place。

8.7.1 Place → Standard Cells And Blocks ...

8.7.2 選擇 Run Full Placement，取消 Include Pre-Place Optimization，勾選 In-Place Optimization。

8.7.3 按 Mode，加選 Enable Clock Gating Cell Awareness，按 OK。

8.7.4 按 OK 開始跑 Placement。

8.8 執行 Place → Refine Placement 來讓 cell 擺置的方向正確。

8.9 再分析一次 Timing (步驟 7.6)。

8.10 如果 WNS 為負，則進行 Timing Optimization。

8.10.1 Timing → Optimize...

8.10.2 使用內定值，按 OK。

## 9 存出檔案

9.1 Design → Save Design as → SoCE ...

9.2 File name 取名為 placed.enc，按 Save。

## 10 Create Power-ring

主要是打在 core 周圍，讓晶片內部的供電電壓均勻，避免 IR drop。

10.1 Place → Refine Placement ...，直接按 OK。

此步驟主要是先拿掉 timing analysis 產生的 trial route 的結果，以免影響 Powerplan，做完這步之後會發現 Physical view 裡面原本有的繞線都被移除了。

10.2 Power → Power Planning → Add Rings ...

10.2.1 Net(s) 填入 VDD VSS。

10.2.2 Ring Configuration

- 10.2.2.1 Top/Bottom Layer 改成 METAL7 H。
- 10.2.2.2 Left/Right Layer 改成 METAL6 V。
- 10.2.2.3 Width 都改成 2。
- 10.2.2.4 填完之後按一下 Spacing 旁邊的 Update。

### 10.2.3 切換到 Advanced

- 10.2.3.1 選擇 Use wire group, Interleaving。
- 10.2.3.2 Number of bits 填入 15。
- 10.2.3.3 按 OK。

做完之後可以看到 Core 跟 I/O Pad 之間多了一圈 Power ring。

## 11 連接 Power pad

### 11.1 Route → Special Route ...

- 11.1.1 Net(s)填入 VDD VSS。
- 11.1.2 Route 只選取 Pad pins 其他皆不選取，按 OK。

做完之後會發現有四條 Power pad 已經連接到 Power ring 上面。

## 12 Create Power-stripe

這個做法主要也是用來使 core 內部的供電電壓均勻。

### 12.1 Power → Power Planning → Add Stripes ...

- 12.1.1 Net(s)填入 VDD VSS。
- 12.1.2 Layer 選擇 METAL6。  
由於我們想要 create 的是直的 stripe，所以在這裏我們選擇 METAL6，若是要打上橫的 stripe，則選擇 METAL7。
- 12.1.3 Width 設定為 1，按下 Update。
- 12.1.4 選擇 Set-to-set distance，並設定為 100。
- 12.1.5 X from left 設定為 150，X from right 設定為 100。
- 12.1.6 切換到 Advanced。
- 12.1.7 選擇 Use wire group 與 Interleaving，Number of bits 設定為 5。
- 12.1.8 加選
  - 12.1.8.1 Omit stripes inside block rings
  - 12.1.8.2 Switch layer over obstructions
  - 12.1.8.3 Pad/Core ring connection → Allow jogging
  - 12.1.8.4 Block ring connection → Allow jogging
- 12.1.9 切換到 Via Generation，並且加選
  - 12.1.9.1 Use exact overlap area on partially intersection wires
  - 12.1.9.2 Spilt vias while encountering Obs and different net Wires/Pins
  - 12.1.9.3 Generate same-sized stack vias while encountering macro Pins/Obs
- 12.1.10 按 OK

**Q3: 一共加入了幾組(group) power stripes? \_\_\_\_\_**

## 13 DRC Check

13.1 Verify → Verify Geometry ...

13.2 檢查是否有任何的 Violations，在這裡出現的 Violation 最好要解決掉(可以重新做 placement 或是手動移動 block)，否則留到後面就很難解決了

13.3 在 layout 上面如果有出現 X 的圖案，則表示那邊有 DRC error。

## 14 存出檔案

14.1 Design → Save Design as → SoCE ...

14.2 File name 取名為 powerplan.enc，按 Save。