

# IC Design Lab

## Place & Route : Innovus

### Innovus Lab PartI

- 1 可以先使用 vi, joe, 或是 cat 等指令查看 design\_data 的 CHIP\_syn.v、CHIP.ioc 以及 CHIP.sdc 檔案，CHIP\_syn.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；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。
  - 1.1 Core power pad 主要是用來做晶片內部的供電所需。
  - 1.2 I/O power pad 主要是用來做 input/output pad 本身的供電所需。
  - 1.3 Corner pad 是用來做 pad 與 pad power 之連接用。
- 2 開啟 Innovus
  - 2.1 % source /usr/cad/cadence/CIC/license.cshrc
  - 2.2 % source /usr/cad/cadence/CIC/innovus.cshrc
  - 2.3 % innovus

注意不可以加上 & 指令(背景執行)，因為 Innovus 開啟之後會使用到兩種不同的介面，一者為目前開啟 Innovus 的 terminal (command line 輸入)，另一個則是開啟軟體之後會出現的 GUI 介面。

注意面積大小不能超過  $1.5 \times 1.5 \text{mm}^2$  (包含 bonding pad)
- 3 Design Import
  - 3.1 File → Import Design...
  - 3.2 Verilog → Files 選擇旁邊的 “...” 按鈕，出現 Netlist Files 之後按下之後選取 design\_data/CHIP\_syn.v，接下按下 Add，即可將 CHIP\_syn.v 檔輸入至 Verilog Netlist → Files 中。
  - 3.3 Verilog → Top Cell 填入 CHIP。
  - 3.4 LEF Files (檔案順序要和下列一樣)

將底下的 Filters 改為 All Files(\*)後，再到 lef 資料夾點選下列的檔案

header6\_V55\_20ka\_cic.lef

fsa0m\_a\_generic\_core.lef

FSA0M\_A\_GENERIC\_CORE\_ANT\_V55.lef

fsa0m\_a\_t33\_generic\_io.lef

FSA0M\_A\_T33\_GENERIC\_IO\_ANT\_V55.lef

BONDPAD.lef

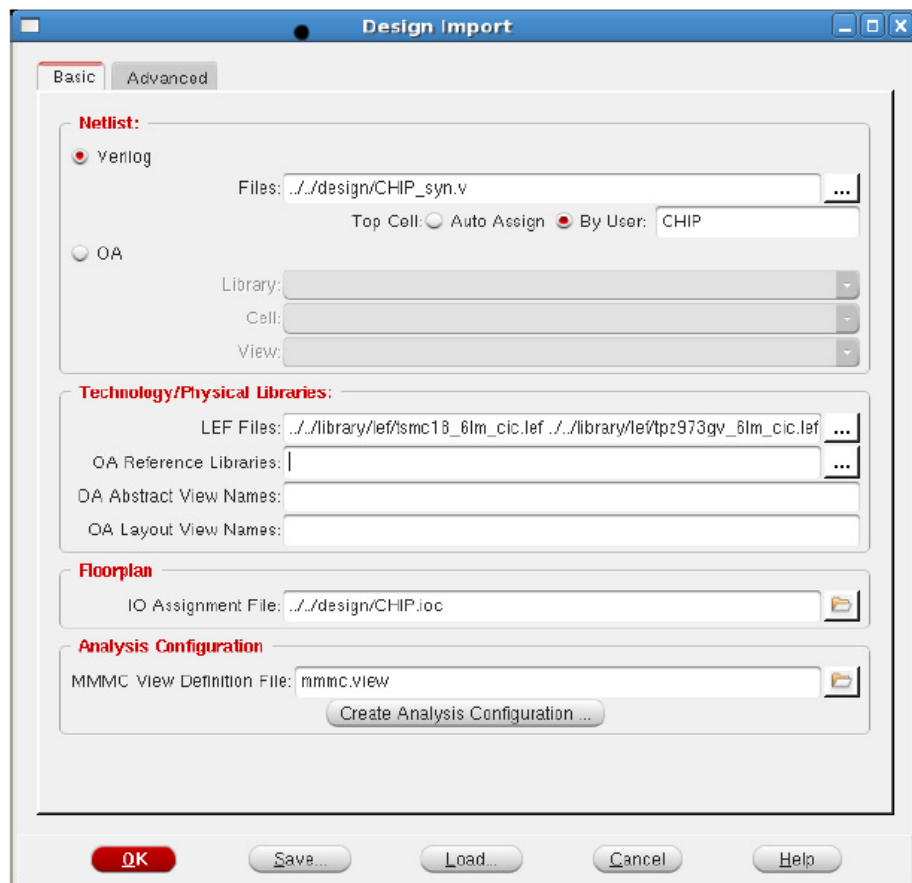
注意：由於 header6\_V55\_20ka\_cic.lef 包含了所有製程資訊，所以一定要擺在第一個輸入。

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

### 3.6 MMMC Objects

MMMC 的設定相當多，為了節省時間，我們直接 load 已經設好的存檔。

按旁邊的 Load，選擇 mmmc.view



3.7 切換到 Advanced。

### 3.8 Power

3.8.1 Power Nets 填入 VCC；Ground Nets 填入 GND。

3.9 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。

3.10 按下 OK

## 4 Global Net Connect

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

### 4.1 Power → Connect Global Nets

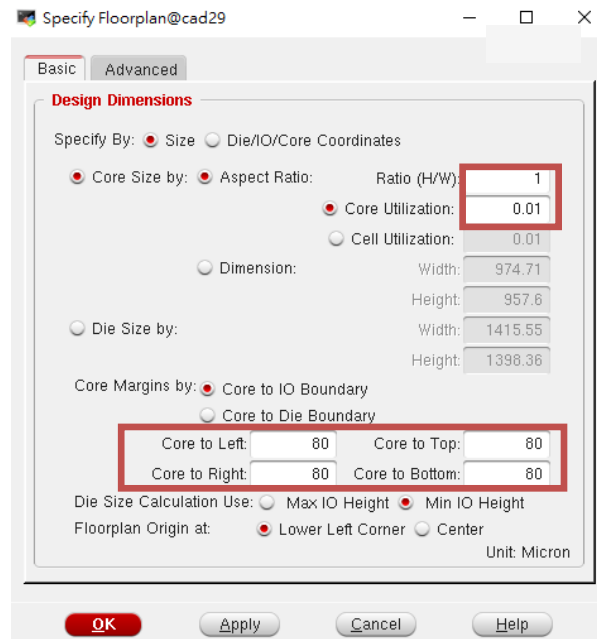
4.1.1 Pin Name(s)填入 VCC，Scope 選 Apply All，To Global Net 填入 VCC，按 Add to List。

4.1.2 Pin Name(s)填入 GND，Scope 選 Apply All，To Global Net 填入 GND，按 Add to List。

4.1.3 按下 Apply，再按 Check，再按 Close。

## 5 Floorplan

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



5.2 可以看到 Layout 上面旁邊粉紅色區塊是我們的 Design。

## 6 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。

### 6.1 Power → Power Planning → Add Rings ...

6.1.1 Net(s)填入 VCC GND。

6.1.2 Ring Configuration

6.1.2.1 Top/Bottom Layer 改成 METAL5H。

6.1.2.2 Left/Right Layer 改成 METAL4V。

6.1.2.3 Width 都改成 2。

6.1.2.4 填完之後按一下 Spacing 旁邊的 Update。

6.1.2.5 Offset 選 Center in channel

6.1.3 切換到 Advanced

6.1.3.1 選擇 Use wire group, Interleaving。

6.1.3.2 Number of bits 填入 15。

6.1.3.3 按 OK。

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

## 7 連接 Power pad

### 7.1 Route → Special Route ...

7.1.1 Net(s)填入 VCC GND。

7.1.2 Route 只選取 Pad pins 其他皆不選取

7.1.3 按 OK(core power 顯示 XX 可以先不理)。

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

## 8 Create Power-stripe

這個做法主要也是用來使 core 內部的供電電壓均勻。

### 8.1 Power → Power Planning → Add Stripes ...

8.1.1 Net(s)填入 VCC GND。

8.1.2 Layer 選擇 METAL4。

由於我們想要 create 的是直的 stripe，所以在這裏我們選擇 METAL4，若是要打上橫的 stripe，則選擇 METAL5，也可打完直的再打橫的。

8.1.3 Width 設定為 1，按下 Update。

8.1.4 選擇 Set-to-set distance，並設定為 50。

8.1.5 X from left 設定為 20，X from right 設定為 20。

8.1.6 按 OK

## 9 存出檔案

### 9.1 File → Save Design

File name 取名為 powerplan.enc，按 OK。

# Innovus Lab PartII

---

## Clock Tree Synthesis

### 1 Add IO Filler

1.1 在 Innovus 的 terminal 執行 addIoFiller.cmd

```
Innovus> source addIoFiller.cmd
```

1.2 可以看到 IO pad 的空隙被填滿了

### Route Power

### 2 Connect Powerpin

2.1 Route → Special Route...

2.2 Basic → Net(s) → ... → 加入 GND VCC

SRoute 只留下 Follow pins → OK

將 Color control 的 Net 設為不可見，就可以看到 core cell 的 power 都連到左右的 power ring 上

### 3 我們接著跑 Full mode 的 Place。

3.1.1 Place → Place Standard Cells ...

3.1.2 選擇 Run Full Placement，勾選 Include Pre-Place Optimization 及 In-Place Optimization。

3.1.3 按 Mode，Specify Maximum Density 設 0.6，按 OK。

3.1.4 按 OK 開始跑 Placement。

3.2 分析 Timing，Design Stage 選擇 Pre-CTS。

3.3 如果 WNS 為負，則進行 Timing Optimization。

3.3.1 Optimize → Optimize Design...

3.3.2 使用內定值，按 OK。

### 4 執行 Clock → Synthesis Clock Tree

在 Synthesis Clock Tree 裡按 **Gen Spec...**

4.1 在 Generate Clock Spec Form 裡，把 \*CK 等 cell 加入到 Selected Cells 中 Output Specification Files 取名為 Clock.ctstch，按 **OK**

4.2 回到 Synthesize Clock Tree，Clock Specification File 保留 Clock.ctstch，按 **OK**

4.3 CTS 結束後，可在 log 裡看到各個 clock timing 的結果，或是可以在 clock\_report/clock.report 裡看到結果，看看有沒有符合 constraint

### 5 執行 Clock → Display → Display Clock Tree...

5.1 在 Display Clock Tree form 裡的 Display Selection 部份選 Display Clock Tree 及 All Level，**OK**

5.2 切到 physical view，把 net display disable，就可以看到 clock tree 的分布

5.3 Clock → Display → Clear Clock Tree Display

(註：Edit 有一些選項可以讓我們手動改 clock tree)

6 File → Save Design，存成 cts.enc

7 再做一次 Timing analysis，Timing → Report Timing，Design Stage 選 Post-CTS，**OK**

7.1 若 WNS 是負的，做 Optimize → Optimize Design...，選 Post-CTS，**OK**，

7.2 重複 7.1 直到 WNS 為正，再存檔成 cts.enc

## Routing

8 Route

8.1 Route → NanoRoute → Route... 加選 Optimize Via, Optimize Wire, Timing drive 和 SI Driven

選 Insert Diodes，Diode Cell Name 輸入 ANTENNA

8.2 NanoRoute form 按 OK，開始 routing

看看有沒有 violation? (IO filler 上會出現 overlap violation，這些可以忽略: Tools → Violation Browser... → Clear Violation)

9 Timing Analysis, Post route Optimization

9.1 Timing → Report Timing

Design Stage 選 Post-Route

Analysis Type: Setup

**OK**

9.2 若 WNS 為負的則做 9.3 不然就跳到 9.4

9.3 Optimize → Optimize Design...

Design Stage: Post-route

加選 Max Fanout

**OK**

一直重覆 9.3 到 WNS 為正的為止

9.4 同 9.1，但將 Analysis Type 改為 Hold 來分析看看 hold time

9.5 先存檔，File → Save Design ... 存成 routed.enc

10 Add Core Filler

10.1 Place → Physical Cells → Add Filler...，在 form 裡的 Cell Name(s) 按 Select，選右邊 Cells List 裡全部的 filler(有 C 的先載入，fillerCC 不用加)按 Add 加入左邊 Selectable Cells Lists 裡

10.2 在 Add Filler form 裡按 OK，Innovus 自動從最大的 filler 加到最小的 filler

10.3 Verify Geometry...，OK (pad 中間的 xx 不用管)

10.4 Verify Process Antenna...，OK

10.5 Verify Connectivity，選 All，選 unrouted net...，OK

10.6 File → Save Design, 存成 corefiller.enc

11 Finish

11.1 File → Save → Netlist..., Netlist File 填 CHIP.v, **OK**

11.2 Timing →

```
% setAnalysisMode -analysisType bcwc
% write_sdf -max_view av_func_mode_max -min_view
av_func_mode_min -edges noedge -splitsetuphold -remashold
-splitcrem -min_period_edges none CHIP.sdf
```

11.3 File → Save → DEF..., 點選 Save Scan, File Name 填 CHIP.def, **OK**

12 加上 bonding pad 我們使用 CORNERD 的 PAD

```
% perl addbonding_v3.8D.pl CHIP.def
% source addbond.cmd
```

13 為了在 LVS 驗證與 posim extraction 時可以找到 IO power 的位置, 要在 export GDS 之間在 IO power pad 外加上 power label, 預計輸入的位置在右邊的 IO power pad 及 IO ground pad 的靠外邊上 (注意: label 一定要打在 pad 上面才有效)

13.1 Edit → Custom Object Editor, 在 Layer 裡輸入 METAL5 按 enter, 選 Text, text 輸入 IOVDD, Origin X,Y 輸入座標(滑鼠移到 pad 上再參考右下角座標), Height 輸入 10, 按 Add to List

13.2 再 Text 輸入 IOVSS

Origin X: x 座標  
Y: y 座標  
Add to List

13.3 Apply, Close, 就會看到 IOVDD 及 IOVSS 被加到我們預定的位置  
File → Save Design, 存成 finish.enc

14 Stream out GDS

```
%setStreamOutMode -specifyViaName default -SEvianames false
-virtualConnection false -uniquifyCellNamesPrefix false -snapToMGrid false
-textSize 1 -version 3
%streamOut CHIP.gds -mapFile streamOut.map -merge
{ ./Phantom/fsa0m_a_generic_core_cic.gds ./Phantom/fsa0m_a_t33_generic_io_
cic.gds ./Phantom/BONDPAD.gds } -stripes 1 -units 1000 -mode ALL
Design → Exit, Yes
```

15 Post-layout simulation

將 CHIP.V 及 CHIP.sdf 複製到 Verilog 資料夾, 更改 TB 中的 module name 及 SDF 的檔名, 執行

```
nverilog HW3_alu_tb.v CHIP.v -v fsa0m_a_generic_core_21.lib.src
fsa0m_a_t33_generic_io_21.lib.src +define+SDF
```

fsa0m\_a\_generic\_core\_21.lib.src 及 fsa0m\_a\_t33\_generic\_io\_21.lib.src 兩個檔案放在 Verilog 的目錄下