

# EDA cloud Cell-base Flow 使用說明

在 EDA Cloud 執行 Cell-base Flow 其實與傳統流程相似，但仍有一些操作細項略微不同，因此本文將針對這部分以 Lab 範例型式作解說。

本範例說明，是以 TSMC90UTM 製程為例，設計內容包含 Standard Cells、Memory Cells(含 Memory BIST 電路)、IO Pads。

## 1 Simulation Tool – NC-verilog/VCS/Modelsim

### 1.1 模擬前準備

為方便模擬，一般會先將模擬之相關檔案編輯成一個檔案 vlog.f 如下。

```
../tb_dprf.v
../../../../design/t90_mem.v
../../../../design/dpsram_4096x16/bist/dpsram_4096x16_top.v
../../../../design/dpsram_4096x16/bist/dpsram_4096x16_wrapper.v
../../../../design/dpsram_4096x16/bist/dpsram_4096x16_rb.v
../../../../design/dpsram_4096x16/bist/dpsram_4096x16.v
../../../../design/dprf_512x32/bist/dprf_512x32_top.v
../../../../design/dprf_512x32/bist/dprf_512x32_wrapper.v
../../../../design/dprf_512x32/bist/dprf_512x32_rb.v
../../../../design/dprf_512x32/bist/dprf_512x32.v
../../../../design/sprf_512x32/bist/sprf_512x32_top.v
../../../../design/sprf_512x32/bist/sprf_512x32_wrapper.v
../../../../design/sprf_512x32/bist/sprf_512x32_rb.v
../../../../design/sprf_512x32/bist/sprf_512x32.v
../../../../design/rom_2048x32/bist/rom_2048x32_top.v
../../../../design/rom_2048x32/bist/rom_2048x32_wrapper.v
../../../../design/rom_2048x32/bist/rom_2048x32_rb.v
../../../../design/rom_2048x32/bist/rom_2048x32.v
-v /cad/CBDK/CBDK_TSMC90GUTM_Arm_v1.2/CIC/Verilog/tpzn90gv3.v
+notimingcheck
+access+r
```

### 解說:

1. 除了 tpzn90gv3.v 是 CBDK 的 IO Pad Verilog 檔案, 其餘皆為自己編輯的 Verilog, 因此那些檔案, 直接給"相對路徑"即可。
2. 由於 EDA cloud 的所有 CBDK 都擺在/cad 目錄裡, 使用任何一個 CBDK 檔案前可先到/cad/CBDK 裡查詢, 在本例皆以 TN90GUTM 製程為例, 故使用時請先切換到/cad/CBDK/CBDK\_TSMC90GUTM\_Arm\_v1.2/CIC 目錄, 整個 Cell-base Flow 會用到相關檔案皆在此目錄可找到。
3. RTL 模擬, 在此只會用到 tpzn90gv3.v, 故可在/cad/CBDK/CBDK\_TSMC90GUTM\_Arm\_v1.2/CIC/Verilog 目錄裡找到該檔案, 其餘 CBDK 使用與查詢方式皆與此範例相似, 也與以往操作模式一樣。

## 1.2 使用 NC-Verilog 作模擬

```
UNIX% Rncverilog -f vlog.f
```

```
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
           0  Initialize Circuits
           26  Testing Bist Operation
          184399  Finish SRAM BIST simulation. Total BIST simulation cycles : 15365 .
           184423  Bist Operation succeeded.
Simulation complete via $finish(1) at time 184435 NS + 0
../tb_dprf.v:276  $finish;
ncsim> exit
Runtime = 5 sec
Connection to Ccb closed.
```

## 1.3 使用 VCS 作模擬

```
UNIX% Rvcs -R -f vlog.f
```

```
           0  Initialize Circuits
           26  Testing Bist Operation
          184399  Finish SRAM BIST simulation. Total BIST simulation cycles : 15365 .
           184423  Bist Operation succeeded.
$finish called from file "../tb_dprf.v", line 276.
$finish at simulation time 184435000
VCS Simulation Report
Time: 184435000 ps
CPU Time: 1.150 seconds; Data structure size: 0.1Mb
Wed Dec 11 17:08:58 2013
CPU time: 1.581 seconds to compile + .160 seconds to elab + .389 seconds to link + 1.334 seconds in simulation
Runtime = 15 sec
Connection to Ccb closed.
```

## 1.4 使用 Modelsim 作模擬

UNIX% Rvsim

註 1：Modelsim 後續使用方式與以往操作方式相同。

註 2：在 EDA Cloud 執行任何程式請勿加上&(背景執行符號)。

## 2 HDL Debug Tool – Verdi/nLint/nWave

### 2.1 Verdi/nLint/nWave 之開啟方式

a. 開啟 Verdi:

UNIX% Rverdi

b. 開啟 Verdi，同時 Import 整個 CHIP 會用到的相關檔案:

UNIX% Rdc\_shell -f vlog.f

註：vlog.f 內容與前一節所述之 vlog.f 內容完全相同。

c. 開啟 nLint，含 GUI 介面及指定 IPQ 之 Rule:

UNIX% RnLint -udr . -rs IPQ.rs -drm -gui

d. 開啟 nWave：

UNIX% RnWave

e. 開啟 nWave，同時 Restore 需要觀察之訊號

UNIX% RnWave -sswr wave.rc

## 3 Synthesis Tool – Design Compiler

### 3.1 合成前準備

編輯.synopsys\_dc.setup

```
set company "CIC"
set designer "andy"
set search_path ". ./lib /cad/CBDK/CBDK_TSMC90GUTM_Arm_v1.2/CIC/SynopsysDC/db $search_path"
set target_library "
slow.db \
fast_leakage.db \
slow_hvt.db \
fast_leakage_hvt.db \
dpsram_4096x16_ss_0.9_125.0_syn.db \
dpsram_4096x16_ff_1.1_-40.0_syn.db \
dprf_512x32_ss_0.9_125.0_syn.db \
dprf_512x32_ff_1.1_-40.0_syn.db \
sprf_512x32_ss_0.9_125.0_syn.db \
sprf_512x32_ff_1.1_-40.0_syn.db \
rom_2048x32_ss_0.9_125.0_syn.db \
rom_2048x32_ff_1.1_-40.0_syn.db \
tpzn90gv3wc.db \
tpzn90gv3lt.db \
"
set link_library "* $target_library dw_foundation.sldb"
set symbol_library " generic.sdb"
set synthetic_library "dw_foundation.sldb"

set verilogout_no_tri true
set sh_enable_line_editing true
set sh_line_editing_mode emacs
history keep 100
alias h history
```

Memory DB File

### 解說:

1. 上圖方框處為 Memory DB Files，請依 Design 需求自行產生。接著將這些檔案之路徑自行填上於"search\_path"，在此範例為"../lib"。
2. 上圖方框以外的 DB Files，皆為合成時會使用到的 DB 檔案。要如何知道 TN90GUTM 有哪些 DB 檔案才能加入這些 DB 檔案，可自行到/cad/CBDK/CBDK\_TSMC90GUTM\_Arm\_v1.2/CIC/SynopsysDC/db 裡查詢。接著透過"search\_path"的設定，自行加上/cad/CBDK/CBDK\_TSMC90GUTM\_Arm\_v1.2/CIC/SynopsysDC/db，合成軟體便可在這裡找到這些 DB 檔案。

## 3.2 Design Compiler 開啟方式

- a. 開啟 DC 需要有 GUI 介面:

```
UNIX% Rdv
```

- b. 開啟 DC 不要有 GUI 介面:

```
UNIX% Rdc_shell
```

- c. 想要查詢 EDA Cloud 有提供那些 DC 版本:

```
UNIX% Rdc_shell @ver
```

```
[andy@st102 run]$ Rdc_shell @ver
Available versions:
(0) 2010.03-sp5
(1) 2013.03-sp4
cur = 2013.03-sp4
```

d. DC 的 cur 為預設開啟版本 2013.03-sp4，如何切換成 2010.03-sp5:

```
UNIX% Rdc_shell @ver 0
```

e. 開啟 DC 不要有 GUI 介面，使用 2013.03-sp4 版本，同時執行 Script 檔案：

```
UNIX% Rdc_shell -f ../script/00_run.tcl
```

註： 00\_run.tcl 假設為自己編輯給 DC 合成之 Script 檔案。

f. 開啟 DC 不要有 GUI 介面，使用 2010.03-sp5 版本，同時執行 Script 檔案：

```
UNIX% Rdc_shell @ver 0 -f ../script/00_run.tcl
```

g. 開啟 DC 需要 GUI 介面，使用 2010.03-sp5 版本，同時執行 Script 檔案：

```
UNIX% Rdv @ver 0 -f ../script/00_run.tcl
```

## 4 Timing & Power Analysis – PrimeTime

### 4.1 PrimeTime 使用前之準備

編輯.synopsys\_pt.setup，內容與前一節.synopsys\_dc.setup 相同。

### 4.2 PrimeTime 之開啟方式

a. 開啟 PrimeTime，同時開啟 GUI 介面:

```
UNIX% Rprimetime
```

b. 開啟 PrimeTime，不要開啟 GUI 介面：

```
UNIX% Rpt_shell
```

c. 開啟 PrimeTime，不要開啟 GUI 介面，同時執行 script：

```
UNIX% Rpt_shell -f pt_script.tcl
```

## 5 Formal Verification – Formality

### 5.1 Formality 之開啟方式

a. 開啟 Formality，同時開啟 GUI 介面:

UNIX% Rformality

b. 開啟 Formality，不要開啟 GUI 介面：

UNIX% Rfm\_shell

c. 開啟 Formality，不要開啟 GUI 介面，同時執行 script：

UNIX% Rfm\_shell -f fm\_script.tcl

## 6 ATPG Tool – TetraMAX

### 6.1 TetraMAX 之開啟方式

a. 開啟 TetraMAX，同時開啟 GUI 介面:

UNIX% Rtmax

b. 開啟 TetraMAX，同時執行 script:

UNIX% Rtmax -shell tmax\_script.tcl

## 7 Memory Compiler Tool

### 7.1 Memory Compiler 工具查詢方式

Memory Compiler 在 TN90GUTM 製程提供了多種 Memory 類型，要如何知道提供那些 Memory，首先先 cd 進入到/cad/CBDK/CBDK\_TSMC90GUTM\_Arm\_v1.2/CIC/Memory 目錄，可以見得 TN90GUTM 共提供五種記憶體，包括 sram\_sp\_adv、sram\_dp\_adv、rf\_sp\_adv、rf\_2p\_adv、rodsd 等，但這些檔案不可以在前台直接執行，故執行 Memory 執行檔與以往不同，其操作新方式如下說明。

### 7.2 Memory Compiler 開啟方式

a. 開啟 Single Port SRAM 工具:

```
UNIX% /cad/CBDK/CBDK_TSMC90GUTM_Arm_v1.2/CIC/Memory/sram_sp_adv/  
bin/sram_sp_adv
```

b. 開啟 Dual Port SRAM 工具:

```
UNIX% /cad/CBDK/CBDK_TSMC90GUTM_Arm_v1.2/CIC/Memory/sram_dp_adv/  
bin/sram_dp_adv
```

c. 開啟 Single Port Register File 工具:

```
UNIX% /cad/CBDK/CBDK_TSMC90GUTM_Arm_v1.2/CIC/Memory/rf_sp_adv/  
bin/rf_sp_adv
```

d. 開啟 Two Port SRAM 工具:

```
UNIX% /cad/CBDK/CBDK_TSMC90GUTM_Arm_v1.2/CIC/Memory/rf_2p_adv/  
bin/rf_2p_adv
```

e. 開啟 ROM 工具:

```
UNIX% /cad/CBDK/CBDK_TSMC90GUTM_Arm_v1.2/CIC/Memory/rodsd/bin/rodsd
```

f. 開啟 Single Port SRAM 工具，同時給予已經設定好的 SPEC 檔:

```
UNIX% /cad/CBDK/CBDK_TSMC90GUTM_Arm_v1.2/CIC/Memory/sram_sp_adv/  
bin/sram_sp_adv -spec sram_1024x8.spec
```

註： sram\_1024x8.spec 只是個隨意舉例之 spec 檔。

## 8 Generate Memory FRAM View - Milkyway

### 8.1 FRAM View 產生方式

晶片佈局時若選擇使用 IC Compiler 作實現，Memory 的部份必須先將 LEF 轉換 FRAM View，其轉換方式如下說明。

step1: 準備 lef2fram.scm 檔案，此檔案可以從/cad/CBDK/CBDK\_TSMC90GUTM\_Arm\_v1.2/CIC/ICC/lef2fram.scm 複製過來。

step2: 修改 lef2fram.scm 部分內容，如下舉例。(1) lib\_name：請給實際 Memory 的名稱，(2)tech\_file：TSMC90UTM 必須設定為/cad/CBDK/CBDK\_TSMC90GUTM\_Arm\_v1.2/CIC/ICC/tsmc090\_9lm\_2thick\_cic.tf，(3)data\_path：請給 Memory 之 VCLEF 檔案的相對位子。

```
define lib_name "dpsram_4096x16"  
define tech_file "/cad/CBDK/CBDK_TSMC90GUTM_Arm_v1.2/CIC/ICC/tsmc090_9lm_2thick_cic.tf"  
define data_path "../"
```

step3: 開啟 Synopsys Milkyway Tool，並執行該 script 檔案，即可產生 FRAM View

```
UNIX% RMilkyway -galaxy -nogui -load lef2fram.scm
```

## 9 Memory BIST Circuit – Syntest TurboBIST

### 9.1 使用 TurboBIST 產生 BIST Circuit 方式

```
UNIX% Rsrambist dprf_512x32 -bcf_file test.bcf -algorithm MARCH_CM
```

註：dprf\_512x32 只是隨意舉例的 Memory，其 mdf 檔案請自行編輯。

## 10 APR Tool Part1 – IC Compiler

### 10.1 Layout 前準備

編輯.synopsys\_dc.setup 與 Design Compiler 完全相同即可。

### 10.2 開啟 IC Compiler(ICC)

a. 開啟 ICC 需要有 GUI 介面:

```
UNIX% Ricc_shell -gui
```

b. 開啟 ICC 不需要有 GUI 介面:

```
UNIX% Ricc_shell
```

c. 開啟 ICC 需要有 GUI 介面，同時執行 script 檔案:

```
UNIX% Ricc_shell -gui -f icc_script.tcl
```

### 10.3 ICC 使用過程之注意事項

因 EDA cloud 關係，使用 ICC 雖然與以往相同，但是 TSMC 相關機密檔案，會看得到檔名或目錄卻無法直接讀取，但 IC Compiler 執行時卻可以使用。例如下圖中，當執行 ls 指令，卻看到像 tpzn90gv3 等目錄或檔案會呈現紅色，表示該檔案或目錄與台積電有關，使用者會無法直接讀取，但 IC Compiler 卻可以正常使用與讀取，所以請使用者遇到這問題時，安心地造以往流程操作即可!

```
[andy@st102 ICC]$ ls
add_io_text.tcl      bond_pads          stout.map          tpzn90gv3          tsmc090nvt_fram
add_tie.tcl         lef2fram.scm      tluplus           tsmc090_9lm_2thick_cic.tf
antenna_9lm_CIC.clf macro.map          tpbn90v           tsmc090hvt_fram
```



## 11 APR Tool Part2 – Innovus

### 11.1 開啟 Innovus

a. 開啟 Innovus 需要有 GUI 介面:

```
UNIX% Rinnovus
```

### 11.2 Innovus 使用過程之注意事項

因 EDA cloud 關係，使用 Innovus 雖然與以往相同，但是 TSMC 相關機密檔案，會看得到檔名或目錄卻無法直接讀取，但 Innovus 執行時卻可以使用。例如下圖中，當執行 ls 指令，卻看到像 antenna\_9lm.lef 等檔案會呈現紅色，表示該檔案與台積電有關，使用者會無法直接讀取，但 Innovus 卻可以正常使用與讀取，所以請使用者遇到這問題時，安心地按照以往流程操作即可!

```
[andy@st102 CIC]$ ls /cad/CBDK/CBDK TSMC90GUTM_Arm/CIC/SOCE/lef/  
antenna_9lm.lef      tpzn90gv3_9lm_cic.lef  tsmc090lk_9lm_2thick_tech_cic.lef  
tpbn90v_9lm_cic.lef  tsmc090hvt_macros.lef  tsmc090nvt_macros.lef
```

## 12 DRC Verification – Qentry DRC/Qcalibre DRC

在 EDA cloud 系統裡，真實的 DRC command file 並不存在於 CBDK 目錄裡，因為這會牽扯到 TSMC 機密資料外洩問題，所以無法直接用 Calibre 軟體作 off-line DRC 驗證，為解決這問題，TSRI 在此提供兩程式—Qentry DRC、Qcalibre DRC，可作 off-line DRC 驗證，使用說明如下。

### 12.1 Qentry DRC 概念

即以往 TSRI 於 Queue server 系統提供的程式，它可將 SOCE 或 ICC 晶片佈局完的 GDS，直接置換成真 Layout，包含 Standard Cell、IO Pad、Bonding Pad、on-chip Memory 等，所以使用本程式為真 Layout 版的 DRC 驗證。

### 12.2 Qentry DRC 使用方法

Step1: 完成晶片佈局，產生假 Layout GDS 檔。

Step2: 如果有 on-chip Memory Cell，請準備 spec 檔。如果有 ROM，請多準備 DAT 檔案。

Step3: 開始執行 Qentry DRC 程式，本範例之語法如下：

```
UNIX% Qentry -M DRC -tech TSMC90GUTM -f CHIP_icc.gds -T CHIP
-s dpsram_4096x16.spec -t t90utm_sram_dp_adv
-s dprf_512x32.spec -t t90utm_rf_2p_adv
-s sprf_512x32.spec -t t90utm_rf_sp_adv
-s rom_2048x32.spec -t t90utm_rodscd
-rom rom_2048x32.dat
-c TSMC90GUTM_rvt -c TSMC90GUTM_hvt
-i TSMC90GUTM -addTagCell -addDummyCell
-o CHIP_netlist
```

註：查詢 Qentry DRC 語法與參數，可以直接打“Qentry”即可！

Step4: 驗證過程中，隨時可以打 showq，觀察 Qentry DRC 執行狀況。

```
[andy@st102 Qentry_drc]$ showq
queueName          qtype resv/used/tot. load_avg arch          states
-----
all.q@Cbat         BIP   0/0/14          0.00    lx24-amd64
#####
- PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
#####
233 0.00000 andyD   quser      qw   12/16/2013 17:28:28 1
```

### 12.3 Qentry DRC 驗證結果之觀察方法

當執行“showq”指令觀察 Qentry 執行狀況的 Log 訊息，假若 quser 那行消失，表示 Qentry DRC 已經驗證完畢。驗證完畢後會產生類似“result\_13-12-16\_andy\_DRC\_st102\_9223”的目錄，可以先觀察 DRC.rep 檔案，觀察 DRC 有無錯誤，或使用 Calibre RVE 直接觀察 DRC 錯誤狀況。注意：Calibre -rve 在 EDA cloud 系統無法直接開啟，必須透過 Laker 軟體間接呼叫 Calibre RVE 工具，方法如下。

Step1: 先將 Qentry 自動產生的 result\_13-12-16\_andy\_DRC\_st102\_9223 目錄內含 DRC\_RES.db 檔案，請先複製到自己可以存取的目錄裡。

Step2: 開啟 Laker 軟體

UNIX% Rlaker

Step3: 因為 Qentry DRC 驗證後，並無產生真 Layout 的 GDS，所以請先自行產生假 Layout 的 GDS 檔案，這樣就可以用真 Layout 的 DRC 驗證結果卻用假 Layout 觀察 DRC 錯誤發生的方位。

Step4: 用 Laker 先開啟 Layout 圖。

Step5: 開啟 Calibre RVE。方法：點選 Verify -> Calibre -> Start RVE ....

Step6: 點選 DRC\_RES.db 檔案，即可開啟 Qentry DRC 驗證結果。

## 12.4 Qcalibre DRC 概念

Qentry DRC 屬於真 Layout 版的 DRC 驗證，Qcalibre DRC 屬於假 Layout 版的 DRC 驗證，倘若設計者已有所有 Cell 的真 Layout GDS 檔，也可以用 Qcalibre DRC 做驗證，其驗證方法與以往操作 Calibre DRC 方式完全一樣，唯一不同是，Qcalibre DRC 由於資源有限，最多只能用到四顆 CPU 的資源，但總比 Qentry DRC 只能用一顆 CPU 更是快速許多。

## 12.5 Qcalibre DRC 使用方法

Step1: 完成晶片佈局，產生 GDS 檔。

Step2: 請複製” /cad/CBDK/CBDK\_TSMC90GUTM\_Arm/CIC/Calibre/drc/TN90GUTM\_DRC.rule”到執行 Qcalibre DRC 的目錄，並修改 TN90GUTM\_DRC.rule 內容如下。

```
LAYOUT PATH "CHIP_icc.gds"  
LAYOUT PRIMARY "CHIP"  
  
// Do not modify next line  
INCLUDE TN90GUTM_DRC_RULE
```

註 1： LAYOUT PATH => 填寫 GDS 檔案

註 2： LAYOUT PRIMARY => 填寫最上層 Cell 名稱

Step3: 開始執行 Qcalibre DRC 程式

```
UNIX% Qcalibre -drc -hier -turbo_all TN90GUTM_DRC.rule
```

## 12.6 Qcalibre DRC 驗證結果之觀察方法

當執行“showq”指令觀察 Qentry 執行狀況的 Log 訊息，假若 quser 那行消失，表示 Qcalibre DRC 已經驗證完畢。驗證完畢後會產生“CALIBRE\_result”的目錄，可以先觀察 DRC.rep 檔案，觀察 DRC 有無錯誤，或使用 Calibre RVE 直接觀察 DRC 錯誤狀況。注意：Calibre -rve 在 EDA cloud 系統無法直接開啟，必須透過 Laker 軟體間接呼叫 Calibre RVE 工具，方法如下。

Step1: 先將 Qcalibre DRC 自動產生的 CALIBRE\_result 目錄內含 DRC\_RES.db 檔案，請先複製到自己可以存取的目錄裡。

Step2: 開啟 Laker 軟體

```
UNIX% Rlaker
```

Step3: 用 Laker 先開啟 Layout 圖。

Step4: 開啟 Calibre RVE。方法：點選 Verify -> Calibre -> Start RVE ....

Step5: 點選 DRC\_RES.db 檔案，即可開啟 Qcalibre DRC 驗證結果。

## 13 LVS Verification – Qcalibre LVS

在 EDA cloud 系統裡，真實的 LVS command file 並不存在於 CBDK 目錄裡，因為這會牽扯到 TSMC 機密資料外洩問題，所以無法直接用 Calibre 軟體作 off-line LVS 驗證，為解決這問題，TSRI 在此提供 Qcalibre LVS 程式，可作 off-line LVS 驗證，使用說明如下。

### 13.1 Qcalibre LVS 概念

Qcalibre LVS 屬於假 Layout 版的 LVS 驗證，主要驗證 Layout 時有無訊號被 Floating 或 Power Open、Short 等，所以假 Layout 的 LVS 驗證足以，在此 TSRI 不會提供 Qentry LVS 程式，Qcalibre LVS 與 Qcalibre DRC 相似，最多可用到四顆 CPU 的資源。

## 13.2 Qcalibre LVS 使用方法

Step1: 完成晶片佈局，產生 GDS 檔。

Step2: 產生原始電路的 SPICE 檔 source.spi，示範語法如下

```
UNIX% Rv2lvs -64 -v CHIP_layout.v
-l dpsram_4096x16.v -l dprf_512x32.v
-l sprf_512x32.v -l rom_2048x32.v
-l tsmc090nvt_fram_lvs.v
-l tsmc090hvt_fram_lvs.v
-l tpzn90gv3_lvs.v
-s tsmc090nvt_fram_lvs.spi
-s tsmc090hvt_fram_lvs.spi
-s tpzn90gv3_lvs.spi
-s dpsram_4096x16.spi
-s dprf_512x32.spi
-s sprf_512x32.spi
-s rom_2048x32.spi
-s1 VDD
-s0 VSS
-o source.spi
```

Step3: 請複製” /cad/CBDK/CBDK\_TSMC90GUTM\_Arm/CIC/Calibre/lvs/TN90GUTM\_LVS\_CB.rule”到執行 Qcalibre LVS 的目錄，並修改 TN90GUTM\_LVS\_CB.rule 內容如下。

```
LAYOUT PRIMARY "CHIP"
LAYOUT PATH "CHIP_laker.gds"
LAYOUT SYSTEM GDSII
LAYOUT CASE YES

SOURCE PRIMARY "CHIP"
SOURCE PATH "source.spi"
SOURCE SYSTEM SPICE
SOURCE CASE YES

TEXT DEPTH PRIMARY
PORT DEPTH PRIMARY
```

註 1：LAYOUT PATH => 填寫 GDS 檔案(記得打上 VDD, VSS, IOVDD, IOVSS TEXT)

註 2：SOURCE PATH => 填寫剛剛用 Rv2lvs 轉換後的 SPICE 檔案 source.spi

Step4: 假若有記憶體或 Hard Macro 電路，請自行在 TN90GUTM\_LVS\_CB.rule 檔案最後一行加上 LVSBOX 的描述如下。

```
LVS BOX dpsram_4096x16  
LVS BOX sprf_512x32  
LVS BOX dprf_512x32  
LVS BOX rom_2048x32
```

Step5: 開始執行 Qcalibre LVS 程式

```
UNIX% Qcalibre -lvs -hier -turbo_all TN90GUTM_LVS_CB.rule
```

### 13.3 Qcalibre LVS 驗證結果之觀察方法

當執行“showq”指令觀察 Qentry 執行狀況的 Log 訊息，假若 quser 那行消失，表示 Qcalibre LVS 已經驗證完畢。驗證完畢後會產生“CALIBRE\_result”的目錄，請觀察 lvs.rep 檔案，觀察 LVS 有無出現驗證成功的笑臉即可。